Modicon
Bibliothèque de blocs de schéma à contacts
Guide utilisateur
Tome 4

840 USE 101 01

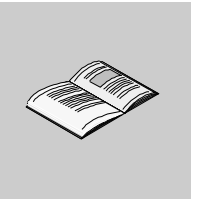
4/2006

31004674.02

a brand of Schneider Electric

Telemecanique

Table des matières



Consignes de sécurité xxix

A propos de ce manuel xxxi

Partie I Informations générales 1

Chapitre 1 Vue d'ensemble du schéma à contacts 3

Segments et réseaux du schéma à contacts 4
Exécution du schéma à contacts par un automate 7
Eléments et instructions de schéma à contacts 9

Chapitre 2 Affectation de mémoire dans un automate 15

Mémoire utilisateur 16
Valeurs de la mémoire d'état. 18
Structure de la mémoire d'état 20
Table de configuration. 22
Table d'affectation des E/S 27

Chapitre 3 Codes opérandes de schéma à contacts 29

Conversion des éléments de schéma à contacts dans la base de données de la mémoire système. 30
Conversion d'instructions DX dans la base de données de la mémoire système 33
Paramètres par défaut des codes opérandes des instructions chargeables 36

Chapitre 4 Instructions 37

Chapitre 5 Groupes d'instructions 39
Instruction du groupe ASCII Functions 41
Instructions du groupe Counters/Timers 42
Instructions du groupe Fast I/O Instructions 43
Instructions du groupe Loadable DX 44
Instructions du groupe Math 45
Instructions du groupe Matrix. 47
Instructions du groupe Miscellaneous 48
Instructions du groupe Move 49
Instructions du groupe Skips/Specials 50
Instructions du groupe Special. 51
Instructions du groupe Coils, Contacts and Interconnects. 52

Chapitre 6 Réseaux d'équation 53
Structure d'un réseau d'équations 54
Equations mathématiques dans les réseaux d'équation 58
Opérations mathématiques dans les réseaux d'équation 62
Fonctions mathématiques dans les réseaux d'équations 68
Conversions de données dans un réseau d'équation 70
Différences d'arrondi dans les automates sans coprocesseur mathématique 72
Performance de référence 73

Chapitre 7 Régulation en boucle fermée/Valeurs analogiques 75
Régulation en boucle fermée/Valeurs analogiques 76
Sous-fonctions PCFL. 77
Exemple PID 81
Exemple PID2 de régulation de niveau 84

Chapitre 8 Mise en forme de messages pour les opérations ASCII READ/WRIT. 89
Mise en forme de messages pour les opérations ASCII READ/WRIT. 90
Identificateurs de format 91
Considérations d'installation spéciales du format des signaux de contrôle/commande 94

Chapitre 9 Bobines, contacts et interconnexions 97
Bobines 98
Contacts. 100
Interconnexions (liaisons) 102

Chapitre 10 Traitement des interruptions 103

Chapitre 11 Traitement des sous-programmes 105

Chapitre 12 Installation des instructions chargeables DX 107

Partie II Description des instructions (de A à D) 109

Chapitre 13 1X3X : Simulation d'entrée. 111
Description sommaire : 1X3X : Simulation d'entrée 112
Représentation : 1X3X : Simulation d'entrée 113

Chapitre 14 AD16 : Addition de valeurs 16 bits 115
Description sommaire 116
Représentation : AD16 : Addition de valeurs 16 bits. 117

Chapitre 15 ADD : Addition 119
Description sommaire 120
Représentation : ADD : Addition simple précision. 121

Chapitre 16 AND : Et Logique 123
Description sommaire 124
Représentation : AND : Et logique 125
Description des paramètres 127

Chapitre 17 BCD : valeur binaire en valeur binaire codée 129
Description sommaire 130
Représentation : BCD : Conversion binaire codée décimale 131

Chapitre 18 BLKM : Copie de bloc. 133
Description sommaire 134
Représentation : BLKM : Copie de bloc 135

Chapitre 19 BLKT : Bloc vers table 137
Description sommaire 138
Représentation : BLKT : Copie de bloc vers table 139
Description des paramètres 140

Chapitre 20 BMDI : Copie de bloc avec interruptions invalidées 141
Description sommaire : BMDI : interruptions de copie de bloc désactivées . . . 142
Représentation : BMDI : interruptions de copie de bloc désactivées 143

Chapitre 21 BROT : Rotation de bit 145
Description sommaire 146
Représentation : BROT : Rotation de bit. 147
Description des paramètres 149

Chapitre 22 CALL : Activation d'une fonction DX immédiate ou différée 151
Description sommaire : CALL : Activation d'une fonction DX immédiate ou différée 152
Représentation : CALL : Activation d'une fonction DX immédiate 153
Représentation : CALL : Activation de la fonction DX différée. 156

Chapitre 23 CANT : Interprétation des bobines, contacts, temporisateurs, compteurs et du bloc SUB 159
Description sommaire : CANT : Interprétation des bobines, contacts, temporisateurs, compteurs et du bloc SUB 160
Représentation : CANT : Interprétation des bobines, contacts, temporisateurs, compteurs et du bloc SUB 161
Description des paramètres : CANT : Interprétation des bobines, contacts, temporisateurs, compteurs et du bloc SUB 163

Chapitre 24 CHS : Configuration de redondance d'UC. 167
Description sommaire 168
Représentation : CHS : Configuration de redondance d'UC 169
Description détaillée. 171

Chapitre 25 CKSM : Checksum. 175
Description sommaire 176
Représentation : CKSM : Checksum 177
Description des paramètres. 179

Chapitre 26 CMPR : Registre de comparaison 181
Description sommaire 182
Représentation : CMPR : Comparaison logique 183
Description des paramètres. 185

Chapitre 27 Bobines. 187
Description sommaire : bobines. 188
Recommandations liées à l'utilisation de bobines 189

Chapitre 28 COMM : Fonction de communication ASCII 191
Description sommaire : COMM : Bloc de communication ASCII 192
Représentation : COMM : Fonction de communication ASCII. 193

Chapitre 29 COMP : Complément d'une matrice. 195
Description sommaire 196
Représentation : COMP : Complément logique. 197
Description des paramètres. 199

Chapitre 30 Contacts 201
Description sommaire : Contacts 202
Représentation : contacts 203

Chapitre 31 CONV : Conversion de données 205
Description sommaire : CONV : Conversion de données 206
Représentation : CONV : Conversion de données 207

Chapitre 32 CTIF : Fonction compteur, temporisateur et interruption. . . 209
Description sommaire : CTIF : Fonction compteur, temporisateur et interruption 210
Représentation : CTIF : Fonction compteur, temporisateur et interruption 211
Description des paramètres : CTIF : Table d'utilisation des registres (partie haute) 212

Chapitre 33 DCTR : Décompteur 217
Description sommaire 218
Représentation : DCTR : Décompteur 219

Chapitre 34 DIOH : Santé des E/S distribuées 221
Description sommaire 222
Représentation : DIOH : Santé des E/S distribuées 223
Description des paramètres 225

Chapitre 35 DISA : Moniteur de bis invalidés 227
Description sommaire : DISA : Moniteur de bits invalidés. 228
Représentation : DISA : Moniteur de bits invalidés 229

Chapitre 36 DIV : Division. 231
Description sommaire 232
Représentation : DIV : Division simple précision. 233
Exemple 235

Chapitre 37 DLOG : Consignation de données pour support de lecture/écriture PCMCIA 237
Description sommaire 238
Représentation : DLOG. 239
Description des paramètres 241
Traitement des erreurs d'exécution. 243

Chapitre 38 DMTH : Fonctions mathématiques en double précision . . . 245
Description sommaire : DMTH : Fonctions mathématiques en double précision : Addition, soustraction, multiplication et division 246
Représentation : DMTH : Fonctions mathématiques en double précision : Addition, soustraction, multiplication et division 247

Chapitre 39 DRUM : Séquenceur à tambour 253
Description sommaire 254
Représentation : DRUM. 255
Description des paramètres. 257

Chapitre 40 DV16 : Division de valeurs 16 bits 259
Description sommaire 260
Représentation : DV16 : Division de valeurs 16 bits 261
Exemple. 262

Partie III Description des instructions (E) 263

Chapitre 41 EARS : Système d'enregistrement d'alarme événementielle. 265
Description sommaire : EARS : Système d'enregistrement d'alarme événementielle 266
Représentation : EARS : Système d'enregistrement d'alarme événementielle. 267
Description des paramètres : EARS : Système d'enregistrement d'alarme événementielle 269

Chapitre 42 EMTH : Fonctions mathématiques étendues 273
Description sommaire 274
Représentation : EMTH : Fonctions mathématiques étendues 275
Description des paramètres. 277
Fonctions EMTH en virgule flottante 279

Chapitre 43 EMTH-ADDDP : Addition en double précision 281
Description sommaire 282
Représentation : EMTH : ADDDP : Fonctions mathématiques en double précision : Addition. 283
Description des paramètres. 285

Chapitre 44 EMTH-ADDFP : Addition en virgule flottante 287
Description sommaire 288
Représentation : EMTH : ADDFP : Fonctions mathématiques en virgule flottante : Addition 289
Description des paramètres. 290

Chapitre 45 EMTH-ADDIF : Addition entier + virgule flottante 291
Description sommaire 292
Représentation : EMTH-ADDIF : Addition entier + virgule flottante 293
Description des paramètres 294

Chapitre 46 EMTH-ANLOG : Antilogarithme de base 10 295
Description sommaire 296
Représentation : EMTH : ANLOG : Antilogarithme de base 10 entier 297
Description des paramètres 299

Chapitre 47 EMTH-ARCOS : Arccosinus en virgule flottante d'un angle (en radians) 301
Description sommaire 302
Représentation : EMTH : ARCOS : Fonctions mathématiques en virgule flottante : Arccosinus d'un angle (en radians). 303
Description des paramètres 305

Chapitre 48 EMTH-ARSIN : Arcsinus en virgule flottante d'un angle (en radians). 307
Description sommaire 308
Représentation : EMTH : ARSIN : Arcsinus d'un angle (en radians) 309
Description des paramètres 311

Chapitre 49 EMTH-ARTAN : Arctangente en virgule flottante d'un angle (en radians) 313
Description sommaire 314
Représentation : Fonctions mathématiques en virgule flottante : Arctangente d'un angle (en radians) 315
Description des paramètres 317

Chapitre 50 EMTH-CHSIN : Changement du signe d'un nombre en virgule flottante 319
Description sommaire 320
Représentation : EMTH – CHSIN : Changement du signe d'un nombre en virgule flottante 321
Description des paramètres 323

Chapitre 51 EMTH-CMPFP : Comparaison en virgule flottante 325
Description sommaire 326
Représentation : EMTH - CMFPF : Comparaison mathématique en virgule flottante. 327
Description des paramètres 329

Chapitre 52 EMTH-CMPIF : Comparaison entier-virgule flottante 331
Description sommaire 332
Représentation : EMTH - CMPIF : Fonctions mathématiques en virgule flottante : Comparaison entier/virgule flottante 333
Description des paramètres 335

Chapitre 53 EMTH-CNVDR : Conversion en virgule flottante de degrés en radians 337
Description sommaire 338
Représentation : EMTH - CNVDR : Conversion de degrés en radians 339
Description des paramètres 341

Chapitre 54 EMTH-CNVFI : Conversion virgule flottante en entier 343
Description sommaire 344
Représentation : EMTH – CNVFI : Conversion virgule flottante en entier 345
Description des paramètres 347
Gestion des erreurs d'exécution 348

Chapitre 55 EMTH-CNVIF : Conversion d'entier en virgule flottante 349
Description sommaire 350
Représentation : EMTH - CNVIF : Conversion entier en virgule flottante 351
Description des paramètres 353
Gestion des erreurs d'exécution 354

Chapitre 56 EMTH-CNVRD : Conversion en virgule flottante de radians en degrés 355
Description sommaire 356
Représentation : EMTH - CNVRD : Conversion de radians en degrés 357
Description des paramètres 359

Chapitre 57 EMTH-COS : Cosinus en virgule flottante d'un angle (en radians) 361
Description sommaire 362
Représentation : EMTH - COS - Cosinus d'un angle (en radians) 363
Description des paramètres 365

Chapitre 58 EMTH-DIVDP : Division en double précision 367
Description sommaire 368
Représentation : EMTH - DIVDP : Fonctions mathématiques en double précision : Division 369
Description des paramètres 371
Gestion des erreurs d'exécution 372

Chapitre 59 EMTH-DIVFI: Nombre à virgule flottante divisée par entier 373
Description sommaire 374
Représentation : EMTH - DIVFI : Nombre en virgule flottante divisé par entier 375
Description des paramètres 377

Chapitre 60 EMTH-DIVFP: Division en virgule flottante 379
Description sommaire 380
Représentation : EMTH - DIVFP : Division en virgule flottante 381
Description des paramètres 383

Chapitre 61 EMTH-DIVIF: Entier divisé par nombre à virgule flottante . . 385
Description sommaire 386
Représentation : EMTH - DIVIF : Entier divisé par nombre en virgule flottante 387
Description des paramètres 388

Chapitre 62 EMTH-ERLOG : Consignation d'erreurs en virgule flottante 389
Description sommaire 390
Représentation : EMTH - ERLOG : Fonctions mathématiques en virgule flottante : Consignation d'erreurs 391
Description des paramètres 393

Chapitre 63 EMTH-EXP : Fonction exponentielle en virgule flottante . . . 395
Description sommaire 396
Représentation : EMTH - EXP : Fonctions mathématiques en virgule flottante : Fonction exponentielle 397
Description des paramètres 399

Chapitre 64 EMTH-LNFP : Logarithme népérien en virgule flottante 401
Description sommaire 402
Représentation : EMTH - LNFP : Logarithme népérien 403
Description des paramètres 405

Chapitre 65 EMTH-LOG : Logarithme de base 10 407
Description sommaire 408
Représentation : EMTH - LOG : Fonctions mathématiques sur entiers : Logarithme de base 10 409
Description des paramètres 411

Chapitre 66 EMTH-LOGFP : Logarithme à base 10 en virgule flottante . . 413
Description sommaire 414
Représentation : EMTH - LOGFP : Logarithme à base 10 415
Description des paramètres 417

Chapitre 67 EMTH-MULDP : Multiplication en double précision 419
Description sommaire 420
Représentation : EMTH - MULDP : Fonctions mathématiques en double précision : Multiplication 421
Description des paramètres 423

Chapitre 68 EMTH-MULFP : Multiplication en virgule flottante 425
Description sommaire 426
Représentation : EMTH - MULFP : Virgule flottante : Multiplication 427
Description des paramètres 428

Chapitre 69 EMTH-MULIF: Multiplication Entier x Nombre en virgule flottante 429
Description sommaire 430
Représentation : EMTH - MULIF : Multiplication Entier x Nombre en virgule flottante 431
Description des paramètres 433

Chapitre 70 EMTH-PI : Chargement de la valeur en virgule flottante de "Pi" 435
Description sommaire 436
Représentation : EMTH - PI : Fonctions mathématiques en virgule flottante : Chargement de la valeur en virgule flottante de PI 437
Description des paramètres 439

Chapitre 71 EMTH-POW : Elévation d'un nombre en virgule flottante à une puissance entière 441
Description sommaire 442
Représentation : EMTH - POW : Elévation d'un nombre en virgule flottante à une puissance entière 443
Description des paramètres 445

Chapitre 72 EMTH-SINE : Sinus en virgule flottante d'un angle (en radians) 447
Description sommaire 448
Représentation : EMTH - SINE : Fonctions mathématiques en virgule flottante : Sinus d'un angle (en radians) 449
Description des paramètres 451

Chapitre 73 EMTH-SQRFP : Racine carrée en virgule flottante 453
Description sommaire 454
Représentation : EMTH - SQRFP : Racine carrée 455
Description des paramètres 457

Chapitre 74 EMTH-SQRT : Racine carrée 459
Description sommaire 460
Représentation : EMTH - SQRT : Racine carrée 461
Description des paramètres 463

Chapitre 75 EMTH-SQRTP : Racine carrée procédé 465
Description sommaire 466
Représentation : EMTH - SQRTP : Fonctions mathématiques en double précision : Racine carrée procédé 467
Description des paramètres 469
Exemple 470

Chapitre 76 EMTH-SUBDP : Soustraction double précision 471
Description sommaire 472
Représentation : EMTH - SUBDP : Fonctions mathématiques en double précision : Soustraction 473
Description des paramètres 475

Chapitre 77 EMTH-SUBFI : Soustraction virgule flottante – entier. 477
Description sommaire 478
Représentation : EMTH - SUBFI : Virgule flottante moins entier 479
Description des paramètres 480

Chapitre 78 EMTH-SUBFP : Soustraction en virgule flottante 481
Description sommaire 482
Représentation : EMTH - SUBFP : Soustraction en virgule flottante 483
Description des paramètres 485

Chapitre 79 EMTH-SUBIF : Soustraction entier – nombre en virgule flottante 487
Description sommaire 488
Représentation : EMTH - SUBIF : Entier moins virgule flottante. 489
Description des paramètres 491

Chapitre 80 EMTH-TAN : Tangente en virgule flottante d'un angle (en radians). 493
Description sommaire 494
Représentation : EMTH - TAN : Tangente d'un angle (en radians) 495
Description des paramètres 497

Chapitre 81 ESI : Prise en charge du module ESI 499
Description sommaire 500
Représentation. 501
Description des paramètres 502
READ ASCII Message (sous-fonction 1) 505
WRITE ASCII Message (sous-fonction 2) 509
GET DATA (sous-fonction 3) 510
PUT DATA (sous-fonction 4) 512
ABORT (entrée médiane à l'état actif) 517
Erreurs d'exécution 518

Chapitre 82 EUCA : Conversion d'unités physiques et alarmes 519
Description sommaire 520
Représentation : EUCA : Unité physique et alarme. 521
Description des paramètres 522
Exemples 524

Partie IV Description des instructions (de F à N) 531

Chapitre 83 FIN : Pile premier entré 535
Description sommaire 536
Représentation : FIN : Pile premier entré 537
Description des paramètres 538

Chapitre 84 FOUT : Pile premier sorti. 539
Description sommaire 540
Représentation : FOUT : Pile premier sorti 541
Description des paramètres 543

Chapitre 85 FTOI : Conversion d'un nombre à virgule flottante en entier 545
Description sommaire 546
Représentation : FTOI : Conversion d'un nombre à virgule flottante en entier 547

Chapitre 86 GD92 : Bloc fonction flux gazeux 549
Description sommaire : GD92 : Bloc fonction flux gazeux 550
Représentation : GD92 : Bloc fonction flux gazeux 551
Description des paramètres : Entrées : GD92 : Bloc fonction flux gazeux 553
Description des paramètres : Sorties : GD92 : Bloc fonction flux gazeux 560
Description des paramètres : Sorties facultatives : GD92 : bloc fonction flux gazeux 561

Chapitre 87 Bloc fonction flux gazeux GFNX AGA GFNX AGA#3 ‘85 et NX19 ‘68. 563
Description sommaire : GFNX : Bloc fonction flux gazeux 564
Représentation : GFNX : Bloc fonction flux gazeux 565
Description des paramètres : Entrées : GFNX : Bloc fonction flux gazeux 567
Description des paramètres : Sorties : GFNX : Bloc fonction flux gazeux 574
Description des paramètres : Sorties facultatives : GFNX : Bloc fonction flux gazeux 575

Chapitre 88 Bloc fonction flux gazeux par la méthode brute GG92 AGA n° 3 1992. 577
Description sommaire : GG92 : Bloc fonction flux gazeux 578
Représentation : GG92 : Bloc fonction flux gazeux. 579
Description des paramètres : Entrées : GG92 : Bloc fonction flux gazeux 581
Description des paramètres : Sorties : GG92 : Bloc fonction flux gazeux. 587
Description des paramètres : Sorties facultatives : GG92 : Bloc fonction flux gazeux 588

Chapitre 89 Bloc fonction flux gazeux par la méthode détaillée GM92 AGA n° 3 et n° 8 1992. 589
Description sommaire : GM92 : Bloc fonction flux gazeux 590
Représentation : GM92 : Bloc fonction flux gazeux 591
Description des paramètres : Entrées : GM92 : Bloc fonction flux gazeux 593
Description des paramètres : Sorties : GM92 : Bloc fonction flux gazeux. 600
Description des paramètres : Sorties facultatives : GM92 : Bloc fonction flux gazeux 601

Chapitre 90 Bloc fonction flux gazeux G392 AGA #3 1992 603
Description sommaire : G392 : Bloc fonction flux gazeux. 604
Représentation : G392 : Bloc fonction flux gazeux 605
Description des paramètres : Entrées : G392 : Bloc fonction flux gazeux 607
Description des paramètres : Sorties : G392 : Bloc fonction flux gazeux 612
Description des paramètres : Sorties facultatives : G392 : Bloc fonction flux gazeux. 613

Chapitre 91 HLTH : Matrices des états et des historiques 615
Description sommaire 616
Représentation : HLTH : Fonctionnement du système 617
Description des paramètres 618
Description des paramètres de la partie haute (matrice de l'historique) 619
Description des paramètres de la partie médiane (matrice des états) 624
Description des paramètres de la partie basse (longueur) 629

Chapitre 92 HSBY : Redondance d'UC 631
Description sommaire : HSBY : Redondance d'UC 632
Représentation : HSBY : Redondance d'UC 633
Description des paramètres de la partie haute : HSBY : Redondance d'UC 635
Description des paramètres de la partie médiane : HSBY : Redondance d'UC 636

Chapitre 93 IBKR : Lecture indirecte de bloc 637
Description sommaire 638
Représentation : IBKR : Lecture indirecte de bloc 639

Chapitre 94 IBKW : Ecriture indirecte d'un bloc 641
Description sommaire 642
Représentation : IBKW : Ecriture indirecte d'un bloc 643

Chapitre 95 ICMP : Comparaison d'entrée 645
Description sommaire 646
Représentation : ICMP : Comparaison d'entrée 647
Description des paramètres 648
Blocs DRUM/ICMP en cascade 650

Chapitre 96 ID : Interruption désactivée 651
Description sommaire : ID - Interruption désactivée 652
Représentation : ID : Interruption désactivée 653
Description des paramètres : ID - Interruption désactivée 654

Chapitre 97 IE : Interruption activée 655
Description sommaire : IE - Interruption activée 656
Représentation : IE : Interruption activée 657
Description des paramètres : IE - Interruption activée 658

Chapitre 98 IMIO : E/S immédiate 659
Description sommaire : IMIO - E/S directe 660
Représentation : IMIO - E/S directe 661
Description des paramètres : IMIO - E/S directe 663
Gestion des erreurs d'exécution : IMIO - E/S directe 665

Chapitre 99 IMOD : Instruction du module d'interruption 667
Description sommaire : IMOD - Module d'interruption 668
Représentation : IMOD : Module d'interruption 669
Description des paramètres : IMOD - Module d'interruption 672

Chapitre 100 ITMR : Générateur d'intervalle de temps 677
Description sommaire : ITMR – Interruption de la temporisation d'intervalle. 678
Représentation : ITMR : Interruption à générateur d'intervalle de temps 679
Description des paramètres : ITMR – Interruption de la temporisation d'intervalle. 681

Chapitre 101 ITOF : Conversion entier en virgule flottante. 685
Description sommaire 686
Représentation : ITOF : Conversion entier en virgule flottante 687

Chapitre 102 JSR : Saut vers sous-programme 689
Description sommaire 690
Représentation : JSR : Saut vers sous-programme 691

Chapitre 103 LAB : Etiquette d'un sous–programme 693
Description sommaire 694
Représentation : LAB : Etiquette 695
Description des paramètres 696

Chapitre 104 LOAD : Chargement de la mémoire flash. 697
Description sommaire 698
Représentation : LOAD : Chargement 699
Description des paramètres 700

Chapitre 105 MAP 3 : Transmission MAP 701
Description sommaire 702
Représentation : MAP 3 : Transmission Map 703
Description des paramètres 704

Chapitre 106 MATH : Opérations sur entiers 709
Description sommaire : MATH : Opérations sur entiers : Racine carrée décimale, racine carrée procédé, logarithme (base 10) et antilogarithme (base 10) 710
Représentation : MATH : Opérations sur entiers : Racine carrée décimale, racine carrée procédé, logarithme (décimal) et antilogarithme (décimal) 711

Chapitre 107 MBIT : Modifier bit 719
Description sommaire 720
Représentation : MBIT : Modification d'un bit logique 721
Description des paramètres 723

Chapitre 108 MBUS : Transmission MBUS 725
Description sommaire 726
Représentation : MBUS : Transfert de Modbus II 727
Description des paramètres 728
La fonction MBUS Lire statistiques 730

Chapitre 109 MRTM : Module de transfert à registres multiples 735
Description sommaire 736
Représentation : MRTM : Module de transfert à registres multiples 737
Description des paramètres 739

Chapitre 110 MSPX (Seriplex) 741
Description sommaire : MSPX (Seriplex) 742
Représentation : MSPX (Seriplex) 743

Chapitre 111 MSTR : Fonctions maître 745
Description sommaire 747
Bloc fonction MSTR 984LL 748
Description des paramètres 751
Commande MSTR Ecrire 755
Commande MSTR LIRE 757
Commande MSTR Lire statistiques locales 759
Commande MSTR Supprimer statistiques locales 761
Commande MSTR Ecrire données globales 763
Commande MSTR Lire données globales 764
Commande MSTR Lire statistiques distantes 765
Commande MSTR Supprimer statistiques distantes 767
Commande MSTR Etat de diffusion des E/S 769
Commande MSTR Réinitialiser module optionnel 772
Commande MSTR Lire CTE (Table d'extension de configuration) 774
Commande MSTR Ecrire CTE (Table d'extension de configuration) 776
Statistiques du réseau Modbus Plus 778
Statistiques du réseau Ethernet TCP/IP 784
Erreurs d'exécution 785
Codes d'erreur Modbus Plus et Ethernet SY/MAX 786
Codes d'erreur spécifiques à SY/MAX 788
Codes d'erreur Ethernet TCP/IP 790
Codes d'erreur CTE pour Ethernet SY/MAX et Ethernet TCP/IP 793

Chapitre 112 MU16 : Multiplication de valeurs 16 bits 795
Description sommaire 796
Représentation : MU16 : Multiplication de valeurs 16 bits 797

Chapitre 113 MUL : Multiplication 799
Description sommaire 800
Représentation : MUL : Multiplication simple précision. 801
Exemple 803
Chapitre 114 NBIT : Contrôle de bit. 805
Description sommaire 806
Représentation : NBIT : Bit normal 807
Chapitre 115 NCBT : Bit NF 809
Description sommaire 810
Représentation : NCBT : Bit NF 811
Chapitre 116 NOBT : Bit NO 813
Description sommaire 814
Représentation : NOBT : Bit NO 815
Chapitre 117 NOL : Module d'option réseau pour Lonworks 817
Description sommaire 818
Représentation : NOL : Module d'option réseau pour Lonworks. 819
Description détaillée 821
Partie V Description des instructions (de O à Q). 823
Chapitre 118 OR : Ou Logique 825
Description sommaire 826
Représentation : OR : Ou logique 827
Description des paramètres 829
Chapitre 119 PCFL : Bibliothèque des fonctions de régulation 831
Description sommaire 832
Représentation : PCFL : Bibliothèque des fonctions de régulation du procédé 833
Description des paramètres 834
Chapitre 120 PCFL-AIN : Entrée analogique. 839
Description sommaire 840
Représentation : PCFL - AIN : Conversion des entrées en unités physiques à l'échelle. 841
Description des paramètres 842
Chapitre 121 PCFL-ALARM : Gestionnaire central d'alarme. 845
Description sommaire 846
Représentation : PCFL - ALRM : Gestionnaire central d'alarme pour une entrée P(v) 847
Description des paramètres 848

Chapitre 122 PCFL-AOUT : Sortie analogique 851
Description sommaire 852
Représentation : PCFL - AOUT : Conversion
des sorties en valeurs dans la plage de 0 à 4095 853
Description des paramètres 854

Chapitre 123 PCFL-AVER : Moyenne des entrées pondérées 855
Description sommaire 856
Représentation : PCFL - AVER : Moyenne des entrées pondérées 857
Description des paramètres 858

Chapitre 124 PCFL-CALC : Calcul d'une formule prédéfinie 861
Description sommaire 862
Représentation : PCFL - CALC : Calcul d'une formule prédéfinie 863
Description des paramètres 864

Chapitre 125 PCFL-DELAY : File d'attente de retard temporel 867
Description sommaire 868
Représentation : PCFL - DELY : File d'attente de retard temporel 869
Description des paramètres 870

Chapitre 126 PCFL-EQN : Calculateur d'équation formatée 873
Description sommaire 874
Représentation : PCFL - EQN : Calculateur d'équation formatée 875
Description des paramètres 876

Chapitre 127 PCFL-INTEG : Intégration d'entrées à intervalles définis . . 879
Description sommaire 880
Représentation : PCFL - INTG : Intégration d'entrées à intervalles définis 881
Description des paramètres 882

Chapitre 128 PCFL-KPID : PID ISA complet non interactif 883
Description sommaire 884
Représentation : PCFL - KPID : Proportionnelle–intégrale–
dérivée (PID) complète non interactive 885
Description des paramètres 886

Chapitre 129 PCFL-LIMIT : Limite de Vp 889
Description sommaire 890
Représentation : PCFL - LIMIT : Limite de Vp 891
Description des paramètres 892

Chapitre 130 PCFL-LIMV : Limite de la vitesse de variation de VP 893
Description sommaire 894
Représentation : PCFL-LIMV : Limite de la vitesse de variation de VP 895
Description des paramètres 896

Chapitre 131 PCFL-LKUP : Linéarisation par interpolation 897
Description sommaire 898
Représentation : PCFL - LKUP : Linéarisation par interpolation 899
Description des paramètres 900

Chapitre 132 PCFL-LLAG : Filtre avance/retard du premier ordre 903
Description sommaire 904
Représentation : PCFL - LLAG : Filtre avance/retard du premier ordre 905
Description des paramètres 906

Chapitre 133 PCFL-MODE : Configuration de l'entrée en mode manuel ou automatique 907
Description sommaire 908
Représentation : PCFL - MODE - Configuration de l'entrée en mode manuel ou automatique 909
Description des paramètres 910

Chapitre 134 PCFL-ONOFF : Valeurs ON/OFF de la plage neutre 911
Description sommaire 912
Représentation : PCFL - ONOFF : Valeurs ON/OFF de la plage neutre 913
Description des paramètres 914

Chapitre 135 PCFL-PI : PI ISA non interactif 917
Description sommaire 918
Représentation : PCFL - PI 919
Description des paramètres 920

Chapitre 136 PCFL-PID : Algorithmes PID 923
Description sommaire 924
Représentation : PCFL - PID : Algorithmes. 925
Description des paramètres 926

Chapitre 137 PCFL-RAMP : Rampe vers la consigne à pente constante. .929
Description sommaire 930
Représentation : PCFL - RAMP : Rampe vers la consigne à pente constante 931
Description des paramètres 932

Chapitre 138 PCFL-RATE : Calcul du taux dérivé sur une durée définie 935
Description sommaire 936
Représentation : PCFL - RATE : Calcul du taux dérivé sur une durée définie 937
Description des paramètres 938

Chapitre 139 PCFL-RATIO : Régulateur de rapport 4 positions 939
Description sommaire 940
Représentation : PCFL - RATIO : Régulateur de rapport 4 positions 941
Description des paramètres 942

Chapitre 140 PCFL-RMPLN : Rampe logarithmique vers consigne 945
Description sommaire 946
Représentation : PCFL - RMPLN : Rampe logarithmique vers consigne 947
Description des paramètres 948

Chapitre 141 PCFL-SEL : Sélection des entrées 949
Description sommaire 950
Représentation : PCFL - SEL : Sélection d'entrée haute/basse/moyenne 951
Description des paramètres 952

Chapitre 142 PCFL-TOTAL : Totalisateur de flux 955
Description sommaire 956
Représentation : PCFL - TOTAL - Totalisateur de flux 957
Description des paramètres 958

Chapitre 143 PEER : Transmission PEER 961
Description sommaire 962
Représentation : PEER : Transfert identique de Modbus II 963
Description des paramètres 964

Chapitre 144 PID2 : Proportionnelle–intégrale–dérivée 965
Description sommaire 966
Représentation : PID2 : Proportionnelle–intégrale–dérivée 967
Description détaillée 969
Description des paramètres 972
Erreurs d'exécution 977

Partie VI Description des instructions (de R à Z) 979

Chapitre 145 R --> T : Registre vers Table 981
Description sommaire 982
Représentation : R Æ T : Copie de registre vers table 983
Description des paramètres 985

Chapitre 146 RBIT : Mise à 0 de bit 987
Description sommaire 988
Représentation : RBIT : Mise à 0 de bit 989

Chapitre 147 READ : Lecture 991
Description sommaire 992
Représentation : READ : Lecture d'un port ASCII. 993
Description des paramètres 995

Chapitre 148 RET : Retour d'un sous–programme 997
Description sommaire 998
Représentation : RET : Retour à la logique ordonnancée. 999

Chapitre 149 RTTI : Table registre vers entrée. 1001
Description sommaire : RTTI : Table registre vers entrée. 1002
Représentation : RTTI : Table registre vers entrée 1003

Chapitre 150 RTTO : Table registre vers sortie 1005
Description sommaire : RTTO : Table registre vers sortie 1006
Représentation : RTTO : Table registre vers sortie. 1007

Chapitre 151 RTU : Terminal déporté 1009
Description sommaire : RTU : Terminal déporté 1010
Représentation : RTU : Terminal déporté 1011

Chapitre 152 SAVE : Sauvegarde mémoire flash 1015
Description sommaire 1016
Représentation : SAVE : Sauvegarde 1017
Description des paramètres 1018

Chapitre 153 SBIT : Mise à 1 de bit 1019
Description sommaire 1020
Représentation : SBIT : Mise à 1 de bit. 1021

Chapitre 154 SCIF : Interfaces de commande séquentielle 1023
Description sommaire 1024
Représentation : SCIF : Interface de commande séquentielle 1025
Description des paramètres 1027

Chapitre 155 SENS : Détection 1029
Description sommaire 1030
Représentation : SENS : Lecture de bit logique 1031
Description des paramètres 1033

Chapitre 156 Liaisons 1035
Description sommaire : Liaisons 1036
Représentation : Liaisons 1037

Chapitre 157 SKP : Omission de réseaux 1039
Description sommaire : SKP : Omission de réseaux. 1040
Représentation : SKP : Omission de réseaux 1041

Chapitre 158 SRCH : Recherche 1043
Description sommaire 1044
Représentation : SRCH : Recherche 1045
Description des paramètres 1047

Chapitre 159 STAT : Etat 1049
Description sommaire 1050
Représentation : STAT : Etat 1051
Description des paramètres 1052
Description de la table des états 1053
Mots 1 à 11 de l'état de l'automate pour Quantum et Momentum 1057
Mots 12 à 20 de l'état de fonctionnement des modules d'E/S pour Momentum 1062
Mots 12 à 171 de l'état de fonctionnement des modules d'E/S pour Quantum 1064
Mots 172 à 277 de l'état de communication pour Quantum 1066
Mots 1 à 11 de l'état de l'automate pour TSX Compact et Atrium 1071
Mots 12 à 15 de la santé des modules d'E/S pour TSX Compact 1074
Mots 182 à 184 de l'état du fonctionnement global et des nouvelles tentatives pour TSX Compact 1075

Chapitre 160 SU16 : Soustraction de valeurs 16 bits 1077
Description sommaire 1078
Représentation : SU16 : Soustraction de valeurs 16 bits 1079

Chapitre 161 SUB : Soustraction 1081
Description sommaire 1082
Représentation : SUB : Soustraction 1083

Chapitre 162 SWAP : Permutation de bits VME. 1085
Description sommaire : SWAP : Permutation de bits VME 1086
Représentation : SWAP : Permutation de bits VME 1087

Chapitre 163 TTR : Table vers registre. 1089
Description sommaire : TTR : Table vers registre 1090
Représentation : TTR : Table vers registre 1091

Chapitre 164 T --> R Table vers registre. 1093
Description sommaire 1094
Représentation : T Æ R : Copie de registre vers table 1095
Description des paramètres 1097

Chapitre 165 T --> T : Table vers table 1099
Description sommaire 1100
Représentation : T Æ T : Copie de table vers table. 1101
Description des paramètres 1103

Chapitre 166 Temporisation T.01 : Temporisation au centième de seconde 1105
Description sommaire 1106
Représentation : T.01 : Temporisation au centième de seconde 1107

Chapitre 167 Temporisation T0.1 : Temporisation au dixième de seconde 1109
Description sommaire 1110
Représentation : T0.1 : Temporisation au dixième de seconde 1111

Chapitre 168 Temporisation T1.0 : Temporisation à la seconde 1113
Description sommaire 1114
Représentation : T1.0 : Temporisation à la seconde. 1115

Chapitre 169 Temporisation T1MS : Temporisation à la milliseconde. . .1117
Description sommaire 1118
Représentation : T1MS : Temporisation à la milliseconde 1119
Exemple 1121

Chapitre 170 TBLK : Table vers Bloc 1123
Description sommaire 1124
Représentation : TBLK : Copie de table vers bloc 1125
Description des paramètres 1127

Chapitre 171 TEST : Comparaison de deux valeurs 1129
Description sommaire 1130
Représentation : TEST : Comparaison de deux valeurs..... 1131

Chapitre 172 UCTR : Compteur 1133
Description sommaire 1134
Représentation : UCTR : Compteur 1135

Chapitre 173 VMER : Lecture de VME 1137
Description sommaire : VMER : Lecture de VME 1138
Représentation : VMER : Lecture de VME 1139
Description des paramètres : VMER : Lecture de VME 1141

Chapitre 174 VMEW : Ecriture de VME 1143
Description sommaire : VMEW : Ecriture de VME 1144
Représentation : VMEW : Ecriture de VME. 1145
Description des paramètres : VMEW : Ecriture de VME..... 1147

Chapitre 175 WRIT : Ecrire 1149
Description sommaire 1150
Représentation : WRIT : Ecriture d'un port ASCII..... 1151
Description des paramètres 1153

Chapitre 176 XMIT : Emission 1155
Présentation générale : XMIT : Emission. 1156
Fonctions Modbus du XMIT 1157

Chapitre 177 Bloc de communication XMIT 1163
Description sommaire : Bloc de communication XMIT 1164
Représentation : Bloc de communication XMIT. 1165
Description des paramètres : Partie médiane :
Table de commande des communications 1167
Description des paramètres : Bloc de communication XMIT 1172
Description des paramètres : Bloc de communication XMIT 1174

Chapitre 178 Bloc d'état du port XMIT 1175
Description sommaire : Bloc d'état du port XMIT. 1176
Représentation : Bloc d'état du port XMIT 1177
Description des paramètres : Partie médiane : Bloc de conversion XMIT. . . . 1179

Chapitre 179 Bloc de conversion XMIT 1183
Description sommaire : Bloc de conversion XMIT 1184
Représentation : Bloc de conversion XMIT 1185
Description des paramètres : Bloc de conversion XMIT 1187

Chapitre 180 XMRD : Lecture de mémoire étendue 1191
Description sommaire 1192
Représentation : XMRD : Lecture de mémoire étendue 1193
Description des paramètres. 1194

Chapitre 181 XMRD : Ecriture en mémoire étendue 1197
Description sommaire 1198
Représentation : XMWT : Ecriture en mémoire étendue 1199
Description des paramètres. 1200

Chapitre 182 XOR : OU exclusif 1203
Description sommaire 1204
Représentation : XOR : OU exclusif booléen. 1205
Description des paramètres. 1207

Annexes **1209**

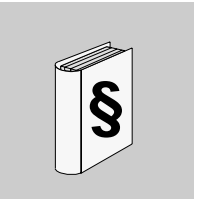
Annexe A **Annexe A** **1211**

Optimisation des performances RIO à l'aide de l'ordonnanceur de segments 1211
Temps de cycle 1212
Mesure du temps de cycle 1216
Optimisation du débit 1217
Ordre d'exécution 1220
Utiliser l'ordonnanceur de segments pour améliorer le débit des E/S critiques 1221
Utiliser l'ordonnanceur de segments pour améliorer les performances du système 1223
Utiliser l'ordonnanceur de segments pour améliorer le traitement des ports de communication 1224
Fonctions de cycle 1225

Glossaire . . . . . **xxxiii**

Index . . . . . **lix**

Consignes de sécurité



AVIS

Veuillez lire soigneusement ces consignes et examiner l'appareil afin de vous familiariser avec lui avant son installation, son fonctionnement ou son entretien. Les messages particuliers qui suivent peuvent apparaître dans la documentation ou sur l'appareil. Ils vous avertissent de dangers potentiels ou attirent votre attention sur des informations susceptibles de clarifier ou de simplifier une procédure.

L'apposition de ce symbole à un panneau de sécurité Danger ou Avertissement signale un risque électrique pouvant entraîner des lésions corporelles en cas de non-respect des consignes.

Ceci est le symbole d'une alerte de sécurité. Il vous avertit d'un risque de blessures corporelles. Respectez scrupuleusement les consignes de sécurité associées à ce symbole pour éviter de vous blesser ou de mettre votre vie en danger.

⚠ DANGER

DANGER indique une situation immédiatement dangereuse qui, si elle n'est pas évitée, **entraînera** la mort ou des blessures graves.

⚠ AVERTISSEMENT

AVERTISSEMENT indique une situation présentant des risques susceptibles de **provoquer** la mort, des blessures graves ou des dommages matériels.

⚠ ATTENTION

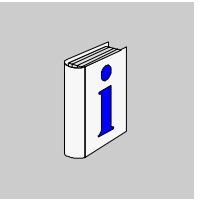
ATTENTION indique une situation potentiellement dangereuse et susceptible d'**entraîner** des lésions corporelles ou des dommages matériels.

REMARQUE IMPORTANTE

Les équipements électriques doivent être installés, exploités et entretenus par un personnel d'entretien qualifié. Schneider Electric n'assume aucune responsabilité des conséquences éventuelles découlant de l'utilisation de cette documentation.

© 2006 Schneider Electric. All rights reserved.

A propos de ce manuel



Présentation

Objectif du document

Cette documentation vous aidera à configurer les instructions LL 984 d'un automate utilisant ProWorx NxT, ProWorx 32 ou Modbus Plus. Les exemples fournis dans ce guide sont réalisés avec ProWorx 32. Les exemples réalisés avec LL 984 à l'aide du logiciel Concept, sont disponibles dans le document Bibliothèque de blocs Concept LL984 (840USE49600).

Champ d'application

Les données et illustrations fournies dans cette documentation ne sont pas contractuelles. Nous nous réservons le droit de modifier nos produits conformément à notre politique de développement permanent. Les informations présentes dans ce document peuvent faire l'objet de modifications sans préavis et ne doivent pas être interprétées comme un engagement de la part de Schneider Electric.

Document à consulter

Titre	Référence
Bibliothèque de blocs Concept LL 984	840 USE 496

Avertissements liés au(x) produit(s)

Schneider Electric ne saurait être tenu responsable des erreurs pouvant figurer dans le présent document. Si vous avez des suggestions, des améliorations ou des corrections à apporter à cette publication, veuillez nous en informer.

Aucune partie de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit, électronique, mécanique ou photocopie, sans autorisation préalable de Schneider Electric.

Toutes les réglementations de sécurité pertinentes locales doivent être observées lors de l'installation et de l'utilisation de ce produit. Pour des raisons de sécurité et pour garantir une conformité aux données système documentées, seul le fabricant est habilité à effectuer des réparations sur les composants.

Lorsque les automates sont utilisés pour des applications présentant des exigences de sécurité technique, suivez les instructions appropriées.

La non-utilisation du logiciel Schneider Electric ou du logiciel approuvé avec nos produits peut entraîner des blessures, des dommages ou un fonctionnement incorrect.

Le non-respect de cet avertissement relatif au produit peut entraîner des blessures ou des dommages matériels.

Commentaires utilisateur

Envoyez vos commentaires à l'adresse e-mail techpub@schneider-electric.com

Description des instructions (de O à Q)

V

Présentation

Introduction

Dans cette section, les descriptions des instructions sont triées par ordre alphabétique de O à Q.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
118	OR : Ou Logique	825
119	PCFL : Bibliothèque des fonctions de régulation	831
120	PCFL-AIN : Entrée analogique	839
121	PCFL-ALARM : Gestionnaire central d'alarme	845
122	PCFL-AOUT : Sortie analogique	851
123	PCFL-AVER : Moyenne des entrées pondérées	855
124	PCFL-CALC : Calcul d'une formule prédéfinie	861
125	PCFL-DELAY : File d'attente de retard temporel	867
126	PCFL-EQN : Calculateur d'équation formatée	873
127	PCFL-INTEG : Intégration d'entrées à intervalles définis	879
128	PCFL-KPID : PID ISA complet non interactif	883
129	PCFL-LIMIT : Limite de Vp	889
130	PCFL-LIMV : Limite de la vitesse de variation de VP	893
131	PCFL-LKUP : Linéarisation par interpolation	897
132	PCFL-LLAG : Filtre avance/retard du premier ordre	903
133	PCFL-MODE : Configuration de l'entrée en mode manuel ou automatique	907
134	PCFL-ONOFF : Valeurs ON/OFF de la plage neutre	911
135	PCFL-PI : PI ISA non interactif	917
136	PCFL-PID : Algorithmes PID	923
137	PCFL-RAMP : Rampe vers la consigne à pente constante	929
138	PCFL-RATE : Calcul du taux dérivé sur une durée définie	935
139	PCFL-RATIO : Régulateur de rapport 4 positions	939
140	PCFL-RMPLN : Rampe logarithmique vers consigne	945
141	PCFL-SEL : Sélection des entrées	949
142	PCFL-TOTAL : Totalisateur de flux	955
143	PEER : Transmission PEER	961
144	PID2 : Proportionnelle–intégrale–dérivée	965

OR : Ou Logique

118

Présentation

Introduction

Ce chapitre décrit l'instruction OR.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

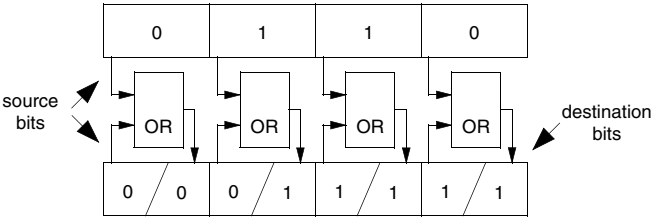
Sujet	Page
Description sommaire	826
Représentation : OR : Ou logique	827
Description des paramètres	829

Description sommaire

Description de la fonction

L'instruction OR effectue une opération OU logique sur les configurations binaires des matrices source et cible.

La configuration binaire liée par OR est ensuite placée dans la matrice cible, écrasant ainsi le contenu précédent.



⚠ AVERTISSEMENT

Ecrasement de toute bobine invalidée dans la matrice cible sans pour autant la valider.

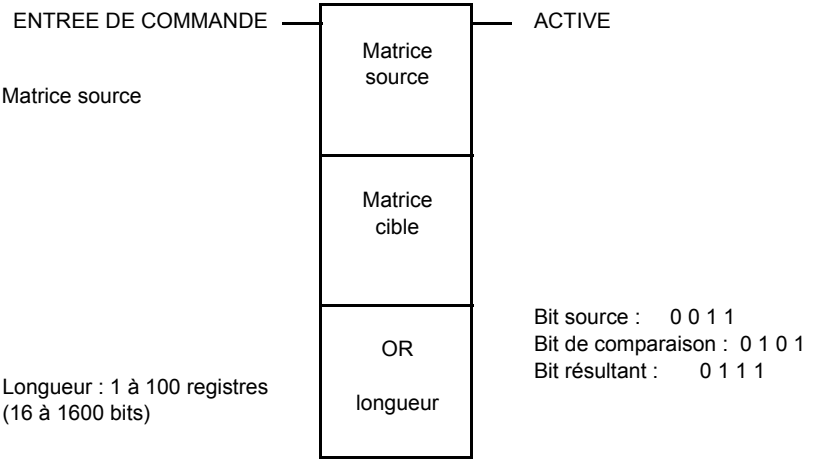
OR écrasera toute bobine invalidée dans la matrice cible sans pour autant la valider. Ceci peut provoquer des dommages si une bobine a invalidé une opération pour des travaux d'entretien ou de réparation puisque l'état de la bobine peut changer suite à l'instruction OR.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

Représentation : OR : Ou logique

Symbole

Représentation de l'instruction

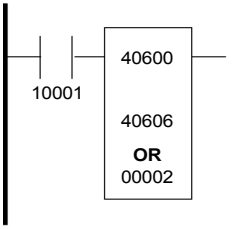


Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Déclenche le OR
Matrice source (partie haute)	0x, 1x, 3x, 4x	ANY_BIT	Première référence de la matrice source.
Matrice cible (partie médiane)	0x, 4x	ANY_BIT	Première référence de la matrice cible
Longueur (partie basse)		INT, UINT	Longueur de la matrice, plage : 1 à 100.
Sortie haute	0x	Aucun	Donne une image de l'état de l'entrée haute

Exemple OR

Chaque fois que le contact 10001 transmet du courant, la *matrice source* formée par la configuration binaire dans les registres 40600 et 40601 est liée par OR à la *matrice cible* formée par la configuration binaire dans les registres 40606 et 40607. La configuration binaire liée par OR est ensuite copiée dans les registres 40606 et 40607, écrasant la configuration binaire cible d'origine.



Matrice source
40600 = 1111111100000000 40601 = 1111111100000000

Matrice cible **d'origine**
40606 = 1111111111111111 40607 = 0000000000000000

Matrice cible **liée par OR**
40606 = 1111111111111111 40607 = 1111111100000000

⚠ ATTENTION

Les sorties et les bobines ne peuvent pas être désactivées à l'aide de l'instruction OR.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Note : Pour conserver la configuration binaire cible d'origine des registres 40606 et 40607, copiez les informations dans une autre table à l'aide de l'instruction BLKM avant d'exécuter l'opération OR.

Description des paramètres

Longueur de la matrice (partie basse)	Le nombre entier saisi en partie basse indique la longueur de la matrice, c'est-à-dire le nombre de registres ou de mots 16 bits des deux matrices. La longueur est comprise entre 1 et 100. La longueur 2 indique que 32 bits de chaque matrice seront liés par OR.

PCFL : Bibliothèque des fonctions de régulation

119

Présentation

Introduction

Ce chapitre décrit l'instruction PCFL.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	832
Représentation : PCFL : Bibliothèque des fonctions de régulation du procédé	833
Description des paramètres	834

Description sommaire

Description de la fonction

L'instruction PCFL vous permet d'accéder à une bibliothèque de fonctions de régulation de procédé utilisant des valeurs analogiques.

Les opérations PCFL relèvent de trois catégories principales.

- Calculs avancés
- Traitement du signal
- Régulation

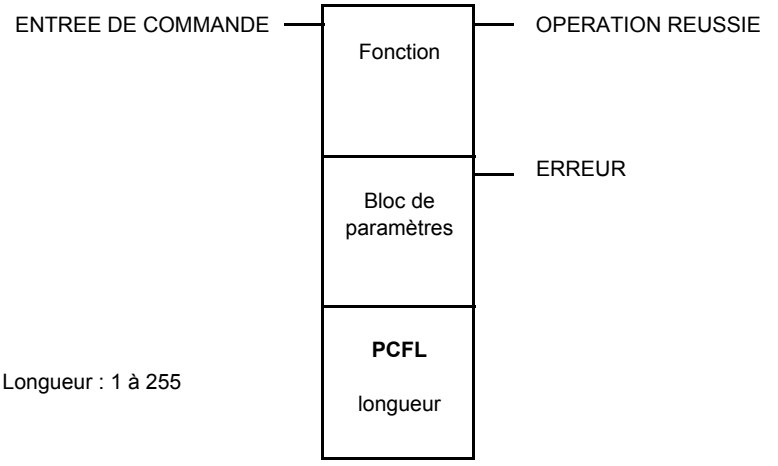
Une fonction PCFL est sélectionnée dans une liste de sous-fonctions alphabétiques dans un menu déroulant du logiciel de console, et la sous-fonction est affichée dans la partie haute de l'instruction (voir *p. 834* pour la liste des sous-fonctions et leurs descriptions).

PCFL utilise la même bibliothèque VF que EMTH. Si l'automate que vous utilisez pour PCFL n'est pas équipé du coprocesseur mathématique 80x87 intégré, l'exécution des calculs dure en comparaison plus longtemps. Les automates avec coprocesseur mathématique peuvent effectuer les calculs PCFL dix fois plus vite que les automates sans la puce. La vitesse ne doit cependant pas être un critère déterminant dans la plupart des applications de régulation de procédé pour lesquelles les temps sont mesurés en secondes et non en millisecondes.

Représentation : PCFL : Bibliothèque des fonctions de régulation du procédé

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie.
Fonction (partie haute)			Sélection de la fonction de régulation de procédé. Un indicateur de la fonction sélectionnée dans la bibliothèque PCFL est précisé dans la partie haute. Pour plus d'informations, reportez-vous *p. 834*.
Bloc de paramètres (partie médiane)	4x	INT, UINT, WORD	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie.
Longueur (partie basse)		INT, UINT	Longueur du bloc de paramètres (dépend de la sous-fonction choisie)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Fonction (partie haute)

Une sous-fonction de la fonction sélectionnée dans la bibliothèque PCFL (sous-fonction) est indiquée dans la partie haute.

Opération	Sous-fonction	Description	Fonctionnement dépendant du temps
Calculs avancés	AVER	Moyenne pondérée des entrées	Non
	CALC	Calcule la formule définie	Non
	EQN	Calcul de l'équation au format	Non
Traitement du signal	ALARM	Gestionnaire central d'alarme pour une entrée VP	Non
	AIN	Convertit les entrées en unités physiques à l'échelle	Non
	AOUT	Convertit les sorties en valeurs dans la plage de 0 à 4095	Non
	DELAY	File d'attente de retard temporel	Oui
	LKUP	Linéarisation par interpolation	Non
	INTEG	Intègre une entrée à intervalle défini	Oui
	LLAG	Filtre avance/retard du premier ordre	Oui
	LIMIT	Limite de VP (bas/bas, bas, haut, haut/haut)	Non
	LIMV	Limiteur de variation de la VP (bas, haut)	Oui
	MODE	Met l'entrée en mode auto ou manuel	Non
	RAMP	Rampe vers consigne à une pente constante	Oui
	RMPLN	Rampe logarithmique vers la consigne (rapprochement aux 2/3 de la consigne à chaque constante de temps)	Oui
	RATE	Calcul du taux dérivé sur une durée définie	Oui
	SEL	Sélection d'entrée haute/basse/ moyenne	Non

Opération	Sous-fonction	Description	Fonctionnement dépendant du temps
Régulation	KPID	Proportionnelle-intégrale-dérivée (PID) ISA complète non interactive	Oui
	ONOFF	Spécifie les valeurs de déclenchement de la plage neutre	Non
	PID	Algorithmes PID	Oui
	PI	PI ISA non interactif (avec modes de fonctionnement arrêt/manuel/auto)	Oui
	RATIO	Régulateur de rapport à quatre positions	Non
	TOTAL	Totalisateur de flux	Oui

Calculs avancés

Les calculs avancés sont utilisés à des fins mathématiques générales et ne sont pas limités aux applications de régulation de procédé. A l'aide des calculs avancés, vous pouvez créer des algorithmes personnalisés de traitement du signal, en déduire des états du procédé contrôlé, des mesures statistiques du procédé, etc.

Des routines mathématiques élémentaires sont déjà proposées dans l'instruction EMTH. La fonctionnalité de calcul intégrée dans PCFL est un calculateur d'équations textuelles permettant l'écriture d'équations personnalisées plutôt que de programmer une série d'opérations mathématiques une à une.

Traitement du signal

Les fonctions de traitement du signal sont utilisées pour traiter des signaux du procédé et dérivés du procédé. Elles peuvent le faire de différentes façons ; elles linéarisent, filtrent, retardent et d'une manière générale modifient un signal. Cette catégorie comporte des fonctions telles que les entrées/sorties analogiques, les limiteurs, Avance/retard et les générateurs de rampe.

Régulation

Les fonctions de régulation effectuent des régulations en boucle fermée pour des applications très diverses. Il s'agit en général d'une boucle de régulation PID (proportionnelle–intégrale–dérivée) à rétroaction. Les fonctions PID dans PCFL offrent différents niveaux de fonctionnalité. La fonction 75, PID, possède la même fonctionnalité générale que l'instruction PID2 mais utilise les opérations mathématiques en virgule flottante et représente certaines options différemment. PID est intéressante dans les cas où PID2 n'est pas adaptée du fait de considérations numériques telles que les arrondis.

vous *p. 77*.

Bloc de paramètres (partie médiane)

Le registre 4x saisi en partie médiane est le premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de l'opération PCFL définie.

Les façons dont les nombreuses opérations PCFL implémentent le bloc de paramètres sont décrites dans les diverses sous-fonctions (opérations PCFL).

Au sein du bloc de paramètre de chaque fonction PCFL, on utilise deux registres pour les états des entrées et sorties.

Drapeaux de sortie

Dans toutes les fonctions PCFL, les bits 12 à 16 du registre d'état des sorties caractérisent les drapeaux standard de sortie suivants :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 11	Inutilisé
12	1 = Erreur mathématique, virgule flottante ou sortie invalide
13	1 = Fonction PCFL inconnue
14	Inutilisé
15	1 = La taille de la table de registre affectée est trop petite
16	1 = Une erreur est survenue, activation de la sortie basse

Pour les fonctions PCFL dépendantes du temps, les bits 9 et 11 sont également utilisés comme suit :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Inutilisé
9	1 = Initialisation en cours
10	Inutilisé
11	1 = Période de traitement illicite
12	1 = Erreur mathématique, virgule flottante ou sortie invalide
13	1 = Fonction PCFL inconnue
14	Inutilisé
15	1 = La taille de la table de registre affectée est trop petite
16	1 = Une erreur est survenue, activation de la sortie basse

Drapeaux d'entrée

Dans toutes les fonctions PCFL, les bits 1 et 3 du registre d'état des entrées définissent les drapeaux suivants des entrées normales :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Initialisation de la fonction achevée ou en cours 0 = Initialise la fonction
2	Inutilisé
3	1 = Ecrasement de la temporisation
4 à 16	Inutilisé

Longueur (partie basse)

Le nombre entier saisi en partie basse indique la longueur, c'est-à-dire le nombre de registres, du bloc de paramètres PCFL. La longueur maximum possible pourra varier selon la fonction que vous définissez.

PCFL-AIN : Entrée analogique

120

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-AIN.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	840
Représentation : PCFL - AIN : Conversion des entrées en unités physiques à l'échelle	841
Description des paramètres	842

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction AIN met l'entrée brute générée par les modules d'entrées analogiques à l'échelle en valeurs physiques utilisables dans les calculs ultérieurs.

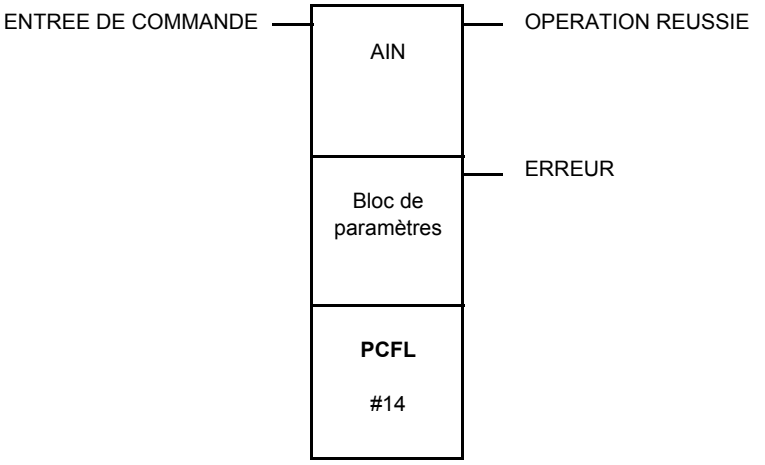
Il existe trois options de mise à l'échelle.

- mise à l'échelle automatique des entrées
- mise à l'échelle manuelle des entrées
- extraction de la racine carrée procédée sur l'entrée pour linéariser le signal avant la mise à l'échelle.

Représentation : PCFL - AIN : Conversion des entrées en unités physiques à l'échelle

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
AIN (partie haute)			Sélection de la sous-fonction AIN
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 842*.
14 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous-fonction AIN (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Mode de fonctionnement

AIN dispose de plages de résolution pour les types d'abonnés suivants :

Plages physiques Quantum

Résolution	Plage : Valide	Plage : En-dessous	Plage : Au-dessus
10 V	768 à 64 768	767	64 769
V	16 768 à 48 768	16 767	48 769
0 à 10 V	0 à 64 000	0	64 001
0 à 5 V	0 à 32 000	0	32 001
1 à 5 V	6 400 à 32 000	6 399	32 001

Thermocouple Quantum

Résolution	Plage : Valide
Degrés TC	-454 à +3 308
0,1 degré TC	-4 540 à +32 767
Unités brutes TC	0 à 65 535

Voltmètre Quantum

Résolution	Plage : Valide	Plage : En dessous	Plage : Au-dessus
10 V	-10 000 à +10 000	-10 001	+10 001
5 V	-5 000 à +5 000	-5 001	+5 001
0 à 10 V	0 à 10 000	0	10 001
0 à 5 V	0 à 5 000	0	5 001
1 à 5 V	1 000 à 5 000	999	5 001

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres AIN est de 14 registres.

Registre	Contenu
Affiché	Entrée depuis un registre 3x
Premier implicite	Réservé
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Échelle 100 % des unités physiques
Sixième et septième implicite	Échelle 0 % des unités physiques
Huitième et neuvième implicite	Entrée manuelle

Registre	Contenu
10ème et 11ème implicite	Entrée auto
12ème et 13ème implicite	Sortie

Etat de sortie

Bit	Fonction
1 à 5	Inutilisé
6	1 = avec racine carrée procédé TC, invalide ; en plage extrapolée, racine carrée procédé non utilisée
7	1 = entrée hors limites
8	1 = écho en dessous de la plage du module d'entrée
9	1 = écho au-dessus de la plage du module d'entrée
10	1 = choix du mode de sortie invalide
11	1 = unités physiques non admissibles
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

Bit	Fonction
1 à 3	Bits d'entrées normales (drapeaux)
4 à 8	Plages (voir les tableaux suivants)
9	1 = racine carrée procédé sur l'entrée brute
10	1 = mode mise à l'échelle manuelle 0 = mode mise à l'échelle auto
11	1 = extrapole le dépassement positif/négatif de plage en mode auto 0 = borne le dépassement pos/nég de plage en mode auto
12 à 16	Inutilisé

Plages physiques Quantum

Bit					
4	5	6	7	8	Plage
0	1	0	0	0	+/- 10 V
0	1	0	0	1	+/- 5 V
0	1	0	1	0	0 à 10 V
0	1	0	1	1	0 à 5 V
0	1	1	0	0	1 à 5 V

Thermocouple Quantum

Bit					
4	**5**	**6**	**7**	**8**	**Plage**
0	1	1	0	1	Degrés TC
0	1	1	1	0	0,1 degré TC
0	1	1	1	1	Unités brutes TC

Voltmètre Quantum

Bit					
4	**5**	**6**	**7**	**8**	**Plage**
1	0	0	0	0	+/- 10 V
1	0	0	1	0	+/- 5 V
1	0	1	0	0	0 à 10 V
1	0	1	1	0	0 à 5 V
1	1	0	0	0	1 à 5 V

Note : Le bit 4 de ce registre n'est pas d'utilisation normale.

PCFL-ALARM : Gestionnaire central d'alarme

121

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-Alarm.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	846
Représentation : PCFL - ALRM : Gestionnaire central d'alarme pour une entrée P(v)	847
Description des paramètres	848

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction ALARM vous offre un bloc central pour la gestion des alarmes dans lequel vous pouvez positionner des seuils haut (H), bas (B), haut max (HM), et bas min (BM) sur une variable de procédé.

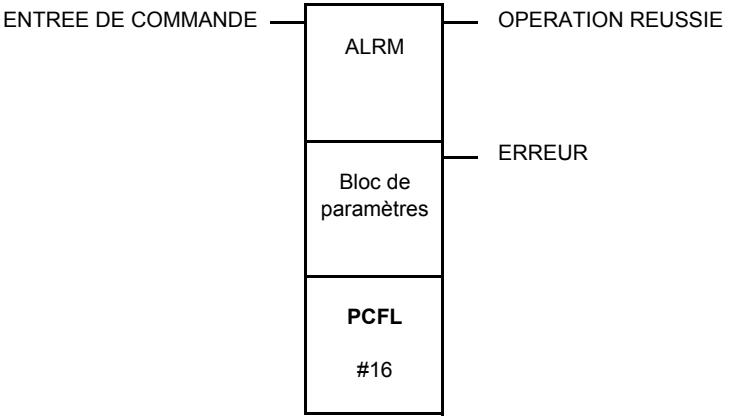
ALARM vous demande de préciser :

- le choix d'un mode de fonctionnement normal ou écart
- l'utilisation des seuils H/B ou les deux H/B et HM/BM
- l'utilisation de la plage neutre (PN) au voisinage des seuils.

Représentation : PCFL - ALRM : Gestionnaire central d'alarme pour une entrée P(v)

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
ALRM (partie haute)			Sélection de la sous-fonction ALARM
Bloc de paramètres (partie médiane)	4x	INT, UINT, WORD	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 848*.
16 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction ALARM (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Mode de fonctionnement

Les modes de fonctionnement suivants sont disponibles.

Mode	Signification
Mode de fonctionnement normal	ALARM fonctionne directement sur l'entrée. Le mode normal est celui par défaut.
Mode de fonctionnement écart	ALARM agit sur le changement entre l'entrée actuelle et l'entrée précédente.
Plage neutre	Lorsqu'elle est validée, l'option PN est incorporée aux seuils HM/H/BM/B. Ces seuils calculés comportent les limites les plus extrêmes ; par exemple si l'entrée a été dans la limite haute, la sortie reste haute et ne change pas lorsque l'entrée atteint le seuil H.
Fonctionnement	Un drapeau est activé lorsque l'entrée ou l'écart est égal ou dépasse le seuil correspondant. Si l'option PN est utilisée, les seuils HM, H, BM et B sont réglés en interne pour le contrôle de franchissement de seuil et l'hystérésis.

Note : ALARM suit automatiquement la dernière entrée même si vous définissez le mode normal pour faciliter un changement en douceur vers le mode écart.

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres ALARM est de 16 registres.

Registre	Contenu
Affiché et premier implicite	Registres d'entrée
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Valeur seuil HM
Sixième et septième implicite	Valeur seuil H
Huitième et neuvième implicite	Valeur seuil B
10ème et 11ème implicite	Valeur seuil BM
12ème et 13ème implicite	Plage neutre (PN) au voisinage du seuil
14ème et 15ème implicite	Dernière entrée

Etat de sortie

Bit	Fonction
1 à 4	Inutilisé
5	1 = PN mise à un nombre négatif
6	1 = mode écart choisi avec option PN
7	1 = BM franchi ($x \leq BM$)
8	1 = B franchi ($x \leq B$, ou $BM < x \leq B$) avec option HM/BM validée
9	1 = H franchi ($x \geq H$, ou $H \leq x < HM$) avec option HM/BM validée
10	1 = HM franchi ($x \geq HM$)
11	1 = seuils définis invalides
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	1 = mode écart 0 = mode normal
6	1 = les deux seuils H/B et HM/BM s'appliquent
7	1 = PN validée
8	1 = conserve le drapeau H/B lorsque les seuils HM/BM sont franchis
9 à 16	Inutilisé

PCFL-AOUT : Sortie analogique

122

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-AOUT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	852
Représentation : PCFL - AOUT : Conversion des sorties en valeurs dans la plage de 0 à 4095	853
Description des paramètres	854

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction AOUT est une interface des signaux calculés des modules de sortie. Elle convertit le signal en une valeur dans la plage 0 à 4 096.

Formules

Formule de la fonction AOUT :

$$\mathrm{OUT} = \frac{\mathrm{scale} \times (\mathrm{IN} - \mathrm{LEU})}{(\mathrm{HEU} - \mathrm{LEU})}$$

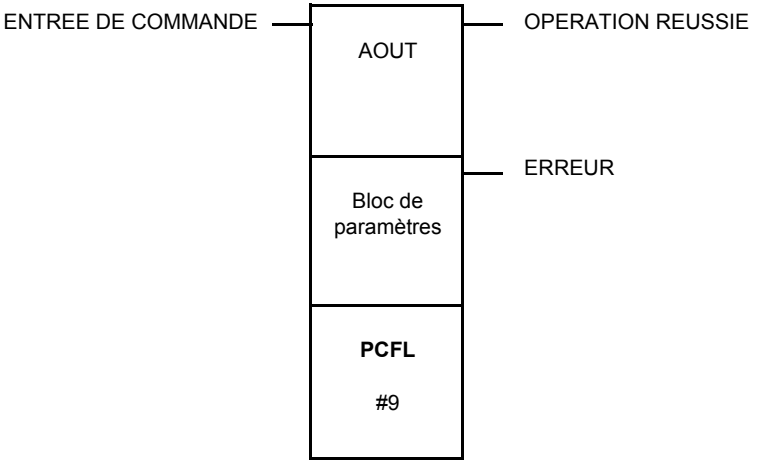
Signification des éléments :

Elément	Signification
UPH	Unité physique haute
IN	Entrée
UPB	Unité physique basse
OUT	Sortie
échelle	Echelle

Représentation : PCFL - AOUT : Conversion des sorties en valeurs dans la plage de 0 à 4095

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
AOUT (partie haute)			Sélection de la sous-fonction AOUT
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 854*.
9 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction AOUT (ne peut être modifiée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres de AOUT est de 9 registres.

Registre	Contenu
Affiché et premier implicite	Entrée en unités physiques
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Unités physiques hautes
Sixième et septième implicite	Unités physiques basses
Huitième et neuvième implicite	Sortie

Etat de sortie

Bit	Fonction
1 à 7	Inutilisé
8	1 = blocage bas
9	1 = blocage haut
10	Inutilisé
11	1 = seuils H/B invalides
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5 à 16	Inutilisé

PCFL-AVER : Moyenne des entrées pondérées

123

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-AVER.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	856
Représentation : PCFL - AVER : Moyenne des entrées pondérées	857
Description des paramètres	858

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Calculs avancés.

La fonction AVER calcule la moyenne de 4 entrées pondérées maximum.

Formules

Formule de la fonction AVER :

$$RES = \frac{(k + (w_1 \times In_1) + (w_2 \times In_2) + (w_3 \times In_3) + (w_4 \times In_4))}{1 + w_1 + w_2 + w_3 + w_4}$$

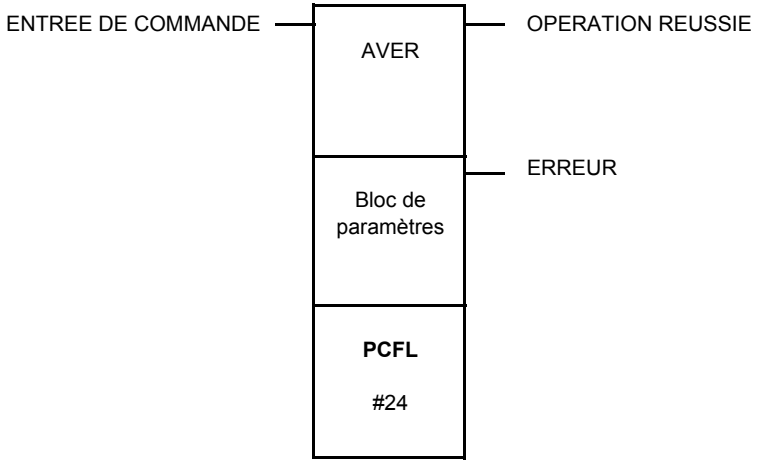
Signification des éléments :

Elément	Signification
In_1 à In_4	Entrées
k	Constante
RES	Résultat
w_1 à w_4	Poids

Représentation : PCFL - AVER : Moyenne des entrées pondérées

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie.
AVER (partie haute)			Sélection de la sous-fonction AVER
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 858*.
24 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous-fonction AVER (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres AVER est de 24 registres.

Registre	Contenu
Affiché et premier implicite	réservé
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Valeur de l'entrée 1 (In1)
Sixième et septième implicite	Valeur de l'entrée 2 (In2)
Huitième et neuvième implicite	Valeur de l'entrée 3 (In3)
10ème et 11ème implicite	Valeur de l'entrée 4 (In4)
12ème et 13ème implicite	Valeur de k
14ème et 15ème implicite	Valeur de wv1
16ème et 17ème implicite	Valeur de wv2
18ème et 19ème implicite	Valeur de wv3
20ème et 21ème implicite	Valeur de wv4
22ème et 23ème implicite	Valeur du résultat

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 9	Inutilisé
10	1 = aucune entrée activée
11	1 = résultat négatif 0 = résultat positif
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	1 = In4 et w4 utilisés
6	1 = In3 et w3 utilisés
7	1 = In2 et w2 utilisés
8	1 = In1 et w1 utilisés
9	1 = k est actif
10 à 16	Inutilisé

Un coefficient de pondération ne peut être utilisé que si son entrée correspondante est validée, c'est-à-dire que les 20ème et 21ème registres implicites (lesquels contiennent la valeur de w4) ne peuvent être utilisés que si les 10ème et 11ème registres implicites (lesquels contiennent la valeur de l'entrée 4) sont validés. Le I du dénominateur n'est utilisé que si la constante est validée.

PCFL-CALC : Calcul d'une formule prédéfinie

124

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-CALC.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	862
Représentation : PCFL - CALC : Calcul d'une formule prédéfinie	863
Description des paramètres	864

Description sommaire

Description de la fonction

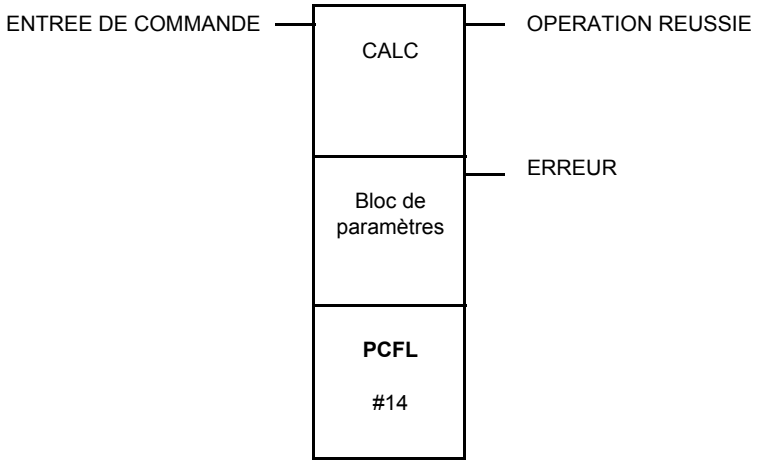
Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Calculs avancés.

La fonction CALC calcule une formule prédéfinie comportant quatre entrées maxi, chacune caractérisée dans un registre distinct du bloc de paramètres.

Représentation : PCFL - CALC : Calcul d'une formule prédéfinie

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie.
CALC (partie haute)			Sélection de la sous–fonction CALC
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie. Pour plus d'informations, reportez-vous *p. 864*.
14 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction CALC (ne peut être modifiée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres CALC est de 14 registres.

Registre	Contenu
Affiché et premier implicite	Réservé
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Valeur de l'entrée A
Sixième et septième implicite	Valeur de l'entrée B
Huitième et neuvième implicite	Valeur de l'entrée C
10ème et 11ème implicite	Valeur de l'entrée D
12ème et 13ème implicite	Valeur de la sortie

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 10	Inutilisé
11	1 = choix de code d'entrée incorrect
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5 à 6	Inutilisé
7 à 10	Code de formule
11 à 16	Inutilisé

Code de formule

Bit				Code de formule
7	8	9	10	
0	0	0	1	$(A \times B) - (C \times D)$
0	0	1	1	$(A \times B)/(C \times D)$
0	1	0	0	$A/(B \times C \times D)$
0	1	0	1	$(A \times B \times C)/D$
0	1	1	0	$A \times B \times C \times D$
0	1	1	1	$A + B + C + D$
1	0	0	0	$A \times B(C-D)$
1	0	0	1	$A[(B/C)^{D}]$
1	0	1	0	$A \times LN(B/C)$
1	0	1	1	$(A-B) - (C - D)/LN[(A - B)/(C - D)]$
1	1	0	0	$(A/B)^{(-C/D)}$
1	1	0	1	$(A-B)/(C - D)$

PCFL-DELAY : File d'attente de retard temporel

125

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-DELAY.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	868
Représentation : PCFL - DELY : File d'attente de retard temporel	869
Description des paramètres	870

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction DELAY peut être utilisée afin de constituer une série de lectures pour compensation de retards temporels dans le programme. Il est possible d'utiliser jusqu'à 10 durées de base pour retarder une entrée.

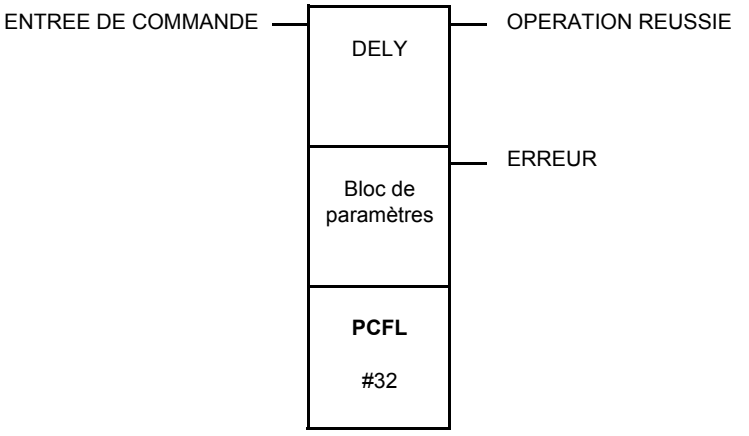
Toutes les valeurs sont placées en même temps dans des registres, le registre x[0] contenant l'entrée actuellement échantillonnée. La 10ème durée de retard n'a pas besoin d'être mémorisée. Lorsque la 10ème durée de la séquence prend effet, la valeur du registre x[9] peut être directement copiée sur la sortie.

Un message DXDONE est retourné lorsque le calcul est terminé. La fonction peut être réinitialisée en basculant le bit de premier cycle.

Représentation : PCFL - DELY : File d'attente de retard temporel

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
DELY (partie haute)			Sélection de la sous-fonction DELY
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 870*.
32 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous-fonction DELY (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres DELAY est de 32 registres.

Registre	Contenu
Affiché et premier implicite	Entrée au temps n
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième implicite	Registre de temps
Cinquième implicite	Réservé
Sixième et septième implicite	Δt (en ms) depuis la dernière exécution
Huitième et neuvième implicite	Intervalle de résolution (en ms)
10ème et 11ème implicite	Retard x[0]
12ème et 13ème implicite	Retard x[1]
14ème et 15ème implicite	Retard x[2]
...	...
28ème et 29ème implicite	Retard x[9]
30ème et 31ème implicite	Registres de sortie

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 3	Inutilisé
4	1 = k hors limites
5 à 8	Nombre de registres restant à initialiser
9 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5 à 8	Retard temporel ≤ 10
9 à 11	Nombre de registres restant à initialiser
12 à 16	Inutilisé

PCFL-EQN : Calculateur d'équation formatée

126

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-EQN.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	874
Représentation : PCFL - EQN : Calculateur d'équation formatée	875
Description des paramètres	876

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Calculs avancés.

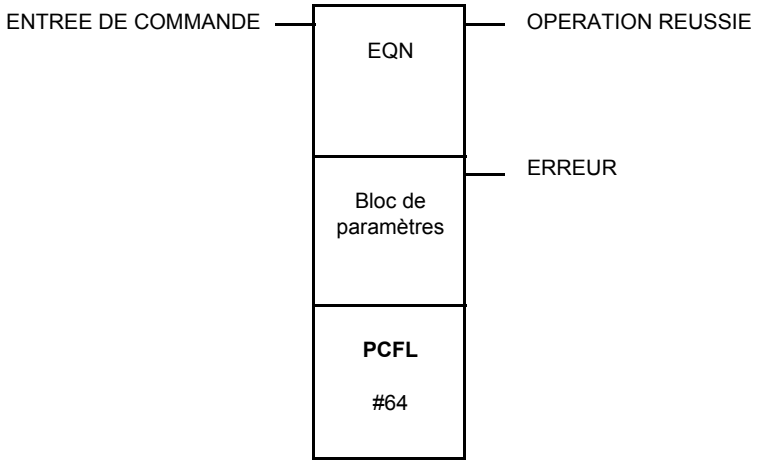
La fonction EQN est un calculateur d'équation formatée. Vous devez définir l'équation située dans le bloc de paramètres avec différents codes précisant les opérateurs, l'entrée choisie et les variables.

EQN sert aux équations ayant quatre variables ou moins, mais n'entrant pas dans le format CALC. Elle complète la fonction CALC en vous permettant de saisir une équation avec variables entières et virgule flottante ainsi que les opérateurs.

Représentation : PCFL - EQN : Calculateur d'équation formatée

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
EQN (partie haute)			Sélection de la sous–fonction EQN
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 876*.
15 à 64 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction EQN
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres EQN peut aller jusqu'à 64 registres.

Registre	Contenu
Affiché et premier implicite	Réservé
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Variable A
Sixième et septième implicite	Variable B
Huitième et neuvième implicite	Variable C
10ème et 11ème implicite	Variable D
12ème et 13ème implicite	Sortie
14ème implicite	Premier code de formule
15ème implicite	Deuxième code de formule possible
...	...
63ème implicite	Dernier code de formule possible

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	Erreur de pile
2 à 3	Inutilisé
4 à 8	Code de la dernière erreur mémorisée
9	1 = code de choix d'opérateur incorrect
10	1 = programmation EQN incomplète
11	1 = choix de code d'entrée incorrect
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	1 = option degré/radian pour les opérations trigonométriques
6 à 8	Inutilisé
9 à 16	Taille de l'équation pour affichage dans Concept

Code de formule

Chaque code de formule dans la fonction EQN définit soit un code de sélection d'entrée, soit un code de sélection d'opérateur.

Code de formule (bloc de paramètres)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Inutilisé
5 à 8	Définition de la sélection d'entrée
9 à 11	Inutilisé
12 à 16	Définition de la sélection d'opérateur

Sélection des entrées

Bit				Sélection des entrées
5	**6**	**7**	**8**	
0	0	0	0	Utiliser sélection d'opérateur
0	0	0	1	Entrée VF
0	0	1	1	Entier 16 bits
1	0	0	0	Variable A
1	0	0	1	Variable B
1	0	1	0	Variable C
1	0	1	1	Variable D

Sélection d'opérateur

Bit					Sélection d'opérateur
12	**13**	**14**	**15**	**16**	
0	0	0	0	0	Pas d'opération
0	0	0	0	1	Valeur absolue
0	0	0	1	0	Addition
0	0	0	1	1	Division
0	0	1	0	0	Exposant
0	0	1	1	1	LN (logarithme népérien)
0	1	0	0	0	G (logarithme)
0	1	0	0	1	Multiplication
0	1	0	1	0	Négation
0	1	0	1	1	Puissance
0	1	1	0	0	Racine carrée
0	1	1	0	1	Soustraction
0	1	1	1	0	Sinus
0	1	1	1	1	Cosinus
1	0	0	0	0	Tangente
1	0	0	0	1	Arcsinus
1	0	0	1	0	Arccosinus
1	0	0	1	1	Arctangente

PCFL-INTEG : Intégration d'entrées à intervalles définis

127

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-INTEG.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	880
Représentation : PCFL - INTG : Intégration d'entrées à intervalles définis	881
Description des paramètres	882

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction INTEG permet d'intégrer sur un intervalle de temps défini. Il n'existe pas de protection contre la saturation de l'intégrale dans cette fonction. INTEG est fonction du temps, par exemple si vous intégrez une valeur d'entrée de 1/sec, tout dépend si elle s'applique sur une seconde (auquel cas le résultat vaut 1) ou sur une minute (auquel cas le résultat est 60).

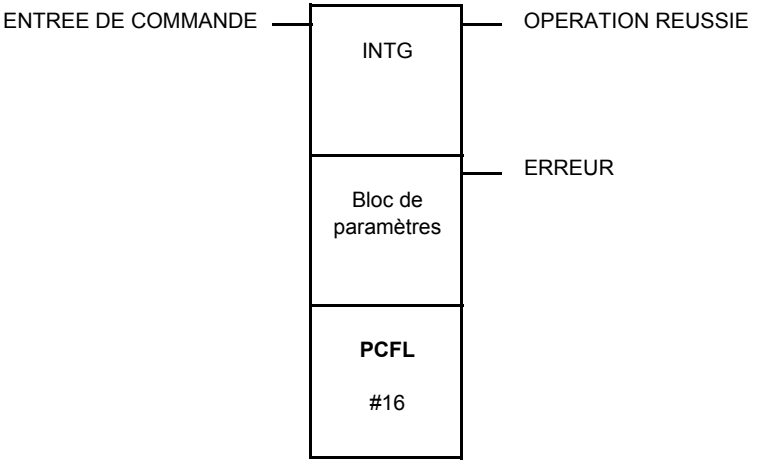
Vous pouvez définir des drapeaux soit pour initialiser soit pour redémarrer la fonction après un temps d'arrêt indéterminé, et vous pouvez remettre à zéro la somme de l'intégrale si vous le souhaitez. Si vous positionnez le drapeau initialisation, vous devez indiquer la valeur initiale (zéro ou la dernière sortie en cas de coupure secteur), et les calculs seront ignorés pour une itération.

La fonction retourne un message DXDONE lorsque l'opération est achevée.

Représentation : PCFL - INTG : Intégration d'entrées à intervalles définis

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
INTG (partie haute)			Sélection de la sous-fonction INTEG
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 882*.
16 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction INTEG (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres INTEG est de 16 registres.

Registre	Contenu
Affiché et premier implicite	Entrée courante
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième implicite	Registre de temps
Cinquième implicite	Réservé
Sixième et septième implicite	Δt (en ms) depuis la dernière exécution
Huitième et neuvième implicite	Intervalle de résolution (en ms)
10ème et 11ème implicite	Dernière entrée
12ème et 13ème implicite	Valeur d'initialisation
14ème et 15ème implicite	Résultat

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Inutilisé
9 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	Somme d'initialisation
6 à 16	Inutilisé

PCFL-KPID : PID ISA complet non interactif

128

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-KPID.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	884
Représentation : PCFL - KPID : Proportionnelle–intégrale–dérivée (PID) complète non interactive	885
Description des paramètres	886

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Régulation.

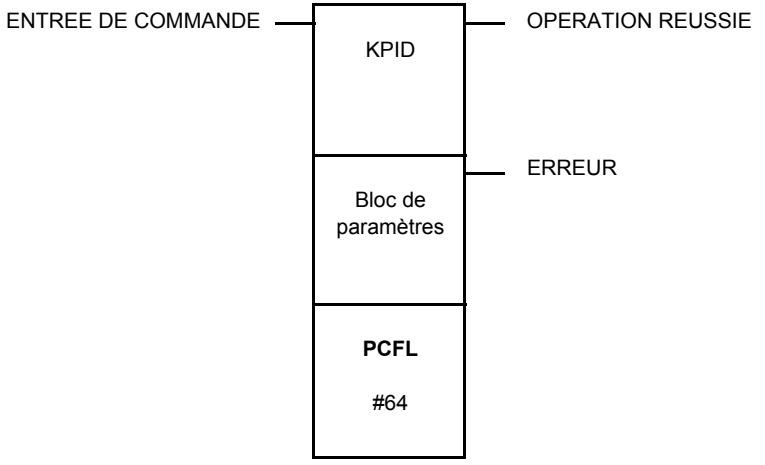
La fonction KPID offre une amélioration de la fonctionnalité de la fonction PID, grâce aux caractéristiques supplémentaires :

- zone de réduction de gain
- registre distinct pour les transferts sans mémoire annexe lorsque le terme intégral n'est pas utilisé
- mode d'initialisation
- consigne externe pour régulation en cascade
- limiteurs de vitesse intégrés pour les changements de consigne et changements en sortie manuelle
- constante ajustable de filtre dérivée
- extension en option des limites d'anti-saturation

Représentation : PCFL - KPID : Proportionnelle–intégrale–dérivée (PID) complète non interactive

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
KPID (partie haute)			Sélection de la sous–fonction KPID
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 886*.
64 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction KPID (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres KPID est de 64 registres.

	Registre	Contenu
Paramètres généraux	Affiché et premier implicite	Entrée réelle, x
	Deuxième implicite	Etat des sorties, registre 1
	Troisième implicite	Etat des sorties, registre 2
	Quatrième implicite	Réservé
	Cinquième implicite	Etat d'entrée
Paramètres d'entrée	Sixième et septième implicite	Facteur proportionnel, KP
	Huitième et neuvième implicite	Temps intégrale, TI
	10ème et 11ème implicite	Durée d'action dérivée, TD
	12ème et 13ème implicite	Constante de temps de retard, TD1
	14ème et 15ème implicite	Zone de réduction de gain, ZRG
	16ème et 17ème implicite	Réduction de gain dans ZRG, KZRG
	18ème et 19ème implicite	Accroissement limite de la valeur de consigne manuelle
	20ème et 21ème implicite	Accroissement limite de la sortie manuelle
	22ème et 23ème implicite	Limite haute de Y
	24ème et 25ème implicite	Limite basse de Y
	26ème et 27ème implicite	Extension des limites d'anti-saturation de l'intégrale
Entrées	28ème et 29ème implicite	Consigne externe pour cascade
	30ème et 31ème implicite	Consigne manuelle
	32ème et 33ème implicite	Entrée manuelle Y
	34ème et 35ème implicite	Initialisation de Y
	36ème et 37ème implicite	Pied (Bias)

	Registre	Contenu
Sorties	38ème et 39ème implicite	Registre de transfert sans mémoire, BT
	40ème et 41ème implicite	Différence de régulation calculée (terme d'erreur), XD
	42ème implicite	Mode de fonctionnement précédent
	43ème et 44ème implicite	Temps de retard (en ms) depuis la dernière exécution
	45ème et 46ème implicite	Précédent écart système, XD_1
	47ème et 48ème implicite	Entrée précédente, X_1
	49ème et 50ème implicite	Composante intégrale de Y, YI
	51ème et 52ème implicite	Composante différentielle de Y, YD
	53ème et 54ème implicite	Consigne, C
	55ème et 56ème implicite	Composante proportionnelle de Y, YP
	57ème implicite	Etat de fonctionnement précédent
Informations temporelles	58ème implicite	10 ms d'horloge à l'instant n
	59ème implicite	Réservé
	60ème et 61ème implicite	Intervalle de résolution (en ms)
Sortie	62ème et 63ème implicite	Sortie régulée, Y

Etat des sorties, registre 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	Erreur
2	1 = Limite basse franchie
3	1 = Limite haute franchie
4	1 = Sélection mode cascade
5	1 = Sélection mode auto
6	1 = Sélection mode arrêt
7	1 = Sélection mode manuel
8	1 = Sélection mode initialisation
9 à 16	Bits des sorties normales (drapeaux)

Etat des sorties, registre 2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Inutilisé
5	1 = Mode précédent D sélectionné
6	1 = Mode précédent I sélectionné
7	1 = Mode précédent P sélectionné
8	1 = Sélection mode précédent
9 à 16	Inutilisé

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	1 = Mode initialisation
6	1 = Mode manuel
7	1 = Mode arrêté
8	1 = Mode cascade
9	1 = Exécution algorithme proportionnelle
10	1 = Exécution algorithme intégrale
11	1 = Exécution algorithme dérivée
12	1 = Exécution algorithme dérivée sur base x 0 = Exécution algorithme dérivée sur base xd
13	1 = Anti-saturation seulement sur YI 0 = Anti-saturation normale intégrale
14	1 = Transfert sans mémoire invalidé 0 = Transfert sans mémoire
15	1 = Suivi Y manuel Y
16	1 = Action inverse de sortie de boucle 0 = Action directe de sortie de boucle

PCFL-LIMIT : Limite de Vp

129

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-LIMIT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	890
Représentation : PCFL - LIMIT : Limite de Vp	891
Description des paramètres	892

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

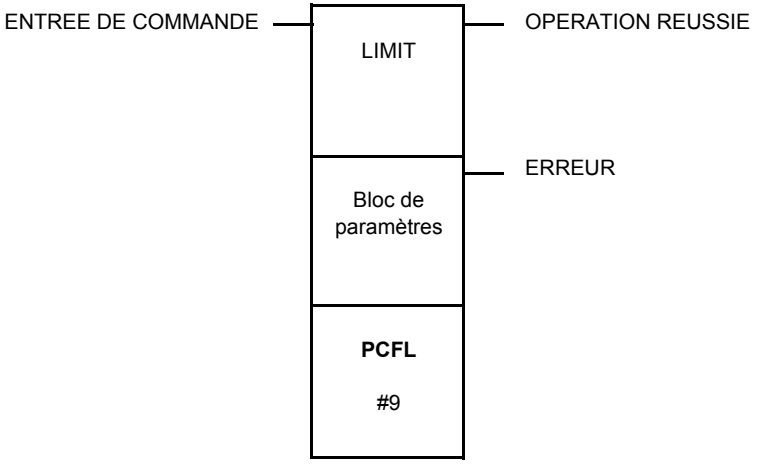
La fonction LIMIT limite l'entrée dans une plage entre des valeurs haute et basse définies. Si la limite haute ou basse est atteinte, la fonction positionne un drapeau H ou B et borne la sortie.

LIMIT retourne un message DXDONE lorsque l'opération est terminée.

Représentation : PCFL - LIMIT : Limite de Vp

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
LIMIT (partie haute)			Sélection de la sous–fonction LIMIT
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 892*.
9 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction LIMIT (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres LIMIT est de 9 registres.

Registre	Contenu
Affiché et premier implicite	Entrée courante
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Limite basse
Sixième et septième implicite	Limite haute
Huitième implicite	Registre de sortie

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Inutilisé
9	1 = entrée < limite basse
10	1 = entrée > limite haute
11	1 = limites haute/basse invalides (par exemple basse ≥ haute)
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5 à 16	Inutilisé

PCFL-LIMV : Limite de la vitesse de variation de VP

130

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-LIMV.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	894
Représentation : PCFL-LIMV : Limite de la vitesse de variation de VP	895
Description des paramètres	896

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

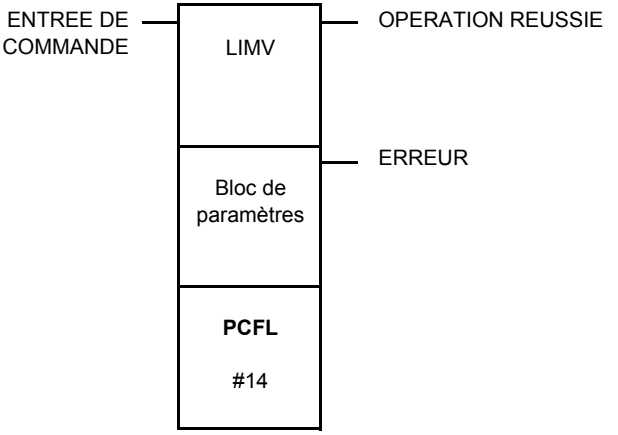
La fonction LIMV limite la vitesse de variation de la variable d'entrée entre des valeurs haute et basse définies. Si la limite haute ou basse est atteinte, la fonction positionne un drapeau H ou B et borne la sortie.

LIMV retourne un message DXDONE lorsque l'opération est terminée.

Représentation : PCFL-LIMV : Limite de la vitesse de variation de VP

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
LIMV (partie haute)			Sélection de la sous-fonction LIMV
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 896*.
14 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction LIMV (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres LIMV est de 14 registres.

Registre	Contenu
Affiché et premier implicite	Registre d'entrée
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième implicite	Registre de temps
Cinquième implicite	Réservé
Sixième et septième implicite	Δt (en ms) depuis la dernière exécution
Huitième et neuvième implicite	Intervalle de résolution (en ms)
10ème et 11ème implicite	Limite de vitesse / sec
12ème et 13ème implicite	Résultat

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 5	Inutilisé
6	1 = limite de vitesse négative
7	1 = entrée < limite basse
8	1 = entrée > limite haute
9 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	(voir *p. 837*)Bits d'entrées normales (drapeaux)
5 à 16	Inutilisé

PCFL-LKUP : Linéarisation par interpolation

131

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-LKUP.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	898
Représentation : PCFL - LKUP : Linéarisation par interpolation	899
Description des paramètres	900

Description sommaire

Description de la fonction

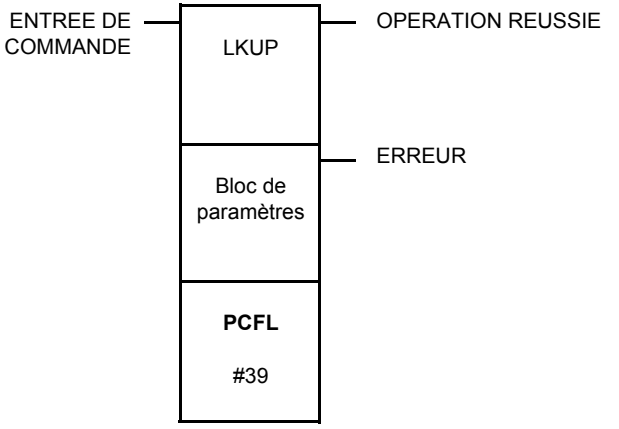
Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction LKUP établit une linéarisation par interpolation à l'aide d'un algorithme linéaire d'interpolation entre points. LKUP peut traiter des intervalles variables entre les points et des nombres variables de points.

Représentation : PCFL - LKUP : Linéarisation par interpolation

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
LKUP (partie haute)			Sélection de la sous–fonction LKUP
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 901*.
39 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction LKUP (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Mode de fonctionnement

La fonction LKUP établit une linéarisation par interpolation à l'aide d'un algorithme linéaire d'interpolation entre points. LKUP peut traiter des intervalles variables entre les points et des nombres variables de points.

Si l'entrée (x) se trouve en dehors de la plage de points définie, la sortie (y) est bornée à la sortie correspondante y0 ou yn. Si la longueur du bloc de paramètres définie est trop courte ou si le nombre de points est hors limites, la fonction ne contrôle pas xn car les informations de ce pointeur sont invalides.

Les points à interpoler sont déterminés par un algorithme binaire de recherche débutant près du centre des données x. La recherche est valide pour x1 < x < xn. La variable x peut apparaître plusieurs fois avec la même valeur ; la valeur choisie dans la linéarisation par interpolation est la première occurrence trouvée.

Par exemple, si la table est la suivante :

x	y
10,0	1,0
20,0	2,0
30,0	3,0
30,0	3,5
40,0	4,0

une entrée de 30,0 trouvera alors la première occurrence de 30,0 et affectera 3,0 en sortie. Une entrée de 31,0 affecterait la valeur 3,55 en sortie.

Aucun tri n'est effectué sur le contenu de la linéarisation par interpolation. Différentes valeurs de table indépendantes doivent être saisies dans l'ordre croissant afin d'éviter des intervalles inaccessibles dans la table.

La fonction retourne un message DXDONE lorsque l'opération est achevée.

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres LKUP est de 39 registres.

Registre	Contenu
Affiché et premier implicite	Entrée
Deuxième implicite	Etat de sortie
Troisième implicite	*Etat p. 901*Etat d'entrée
Quatrième implicite	Nombre de couples de points
Cinquième et sixième implicite	Point x1
Septième et huitième implicite	Point y1
Neuvième et dixième implicite	Point x2
11ème et 12ème implicite	Point y2
. . .	. . .
33ème et 34ème implicite	Point x8
35ème et 36ème implicite	Point y8
37ème et 38ème implicite	Sortie

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 9	Inutilisé
10	1 = entrée bornée, c'est-à-dire en dehors de la plage de la table
11	! = nombre de points invalide
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5 à 16	Inutilisé

PCFL-LLAG : Filtre avance/ retard du premier ordre

132

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-LLAG.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	904
Représentation : PCFL - LLAG : Filtre avance/retard du premier ordre	905
Description des paramètres	906

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction LLAG offre une compensation dynamique à une perturbation connue. Elle apparaît généralement dans un algorithme de commande anticipative ou comme filtre dynamique. LLAG fait passer l'entrée à travers un filtre comportant un terme d'avance (un numérateur) et un terme de retard (un dénominateur) dans le domaine de fréquence, puis la multiplie par un gain. Avance, retard, gain et intervalle de résolution doivent être définis par l'utilisateur.

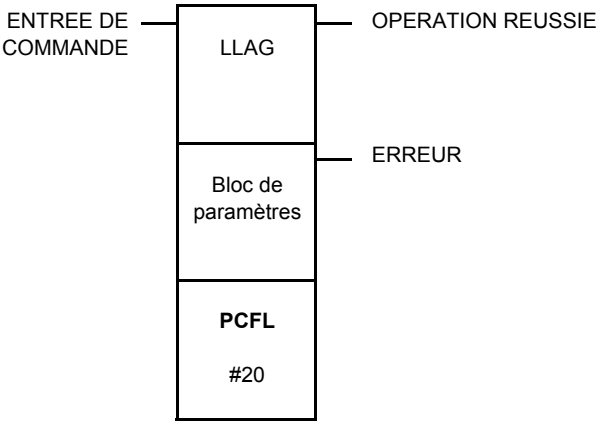
Pour de meilleurs résultats, donnez des valeurs d'avance et de retard $\geq 4 * \Delta t$. Ceci assurera une granularité suffisante de la réponse de sortie.

LLAG renvoie un message DXDONE lorsque l'opération est terminée.

Représentation : PCFL - LLAG : Filtre avance/retard du premier ordre

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
LLAG (partie haute)			Sélection de la sous–fonction LLAG
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 906*.
20 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction LLAG (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres LLAG est de 20 registres.

Registre	Contenu
Affiché et premier implicite	Entrée courante
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième implicite	Registre de temps
Cinquième implicite	Réservé
Sixième et septième implicite	Δt (en ms) depuis la dernière exécution
Huitième et neuvième implicite	Intervalle de résolution (en ms)
10ème et 11ème implicite	Dernière entrée
12ème et 13ème implicite	Terme d'avance
14ème et 15ème implicite	Terme de retard
16ème et 17ème implicite	Gain du filtre
18ème et 19ème implicite	Résultat

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Inutilisé
9 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5 à 16	Inutilisé

PCFL-MODE : Configuration de l'entrée en mode manuel ou automatique

133

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-MODE.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	908
Représentation : PCFL - MODE - Configuration de l'entrée en mode manuel ou automatique	909
Description des paramètres	910

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction MODE configure une station en manuel ou automatique pour valider ou invalider les transferts de données vers le bloc suivant. La fonction opère comme l'instruction BLKM, en copiant une valeur dans le registre de sortie.

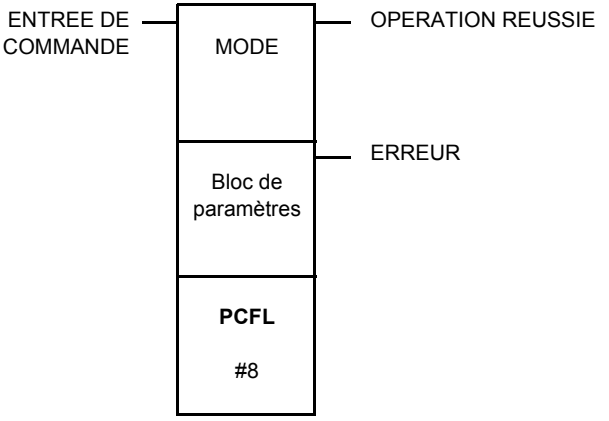
En mode auto, l'entrée est copiée sur la sortie. En mode manuel, la sortie est écrasée par une saisie de l'utilisateur.

MODE retourne un message DXDONE lorsque l'opération est terminée.

Représentation : PCFL - MODE - Configuration de l'entrée en mode manuel ou automatique

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
MODE (partie haute)			Sélection de la sous-fonction MODE
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 910*.
8 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction MODE (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres MODE est de huit registres.

Registre	Contenu
Affiché et premier implicite	Entrée
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Entrée manuelle
Sixième et septième implicite	Registre de sortie

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 10	Inutilisé
11	Mode Echo : 1 = mode manuel 0 = mode auto
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	1 = mode manuel 0 = mode auto
6 à 16	Inutilisé

PCFL-ONOFF : Valeurs ON/OFF de la plage neutre

134

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-ONOFF.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	912
Représentation : PCFL - ONOFF : Valeurs ON/OFF de la plage neutre	913
Description des paramètres	914

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Régulation.

La fonction ONOFF permet de commander le signal de sortie entre les niveaux complètement à 1 et complètement à 0, de sorte qu'un utilisateur puisse forcer manuellement la sortie à 1 ou à 0.

Vous pouvez commander la sortie par configuration soit directe, soit inverse :

Configuration	Si l'entrée...	Alors la sortie...
Directe	< (C - PN)	1
	> (C + PN)	0
Inverse	> (C + PN)	1
	< (C - PN)	0

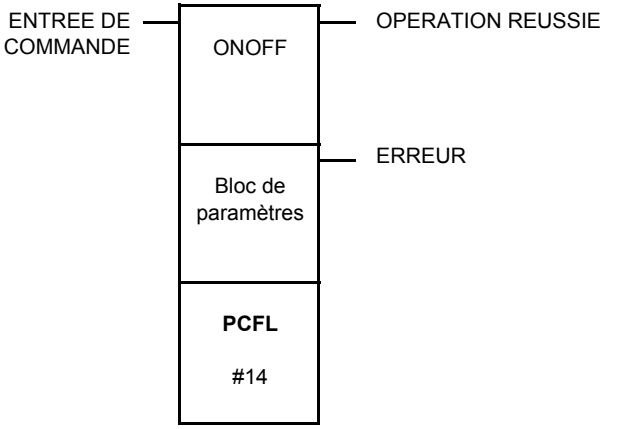
Forçage manuel

Deux bits dans le registre d'état des entrées (le troisième registre implicite du bloc de paramètres) sont utilisés pour le forçage manuel. Lorsque le bit 6 est mis à 1, le mode manuel est forcé. En mode manuel, un 0 dans le bit 7 force la sortie à 0, et un 1 dans le bit 7 force la sortie à 1. L'état du bit 7 n'a de signification qu'en mode manuel.

Représentation : PCFL - ONOFF : Valeurs ON/OFF de la plage neutre

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
ONOFF (partie haute)			Sélection de la sous-fonction ONOFF
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 914*.
14 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction ONOFF (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres ONOFF est de 14 registres.

Registre	Contenu
Affiché et premier implicite	Entrée courante
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Consigne, C
Sixième et septième implicite	Plage neutre (PN) autour de C
Huitième et neuvième implicite	Complètement à 1 (sortie maximum)
10ème et 11ème implicite	Complètement à 0 (sortie minimum)
12ème et 13ème implicite	Sortie, à 1 ou 0

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Inutilisé
9	1 = PN mise à un nombre négatif
10	Mode Echo : 1 = forçage manuel 0 = mode auto
11	1 = sortie mise à 1 0 = sortie mise à 0
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	1 = configuration inverse 0 = configuration directe
6	1 = forçage manuel 0 = mode auto
7	1 = force la sortie à 1 en mode manuel 0 = force la sortie à 0 en mode manuel
8 à 16	Inutilisé

PCFL-PI : PI ISA non interactif

135

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-PI.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	918
Représentation : PCFL - PI	919
Description des paramètres	920

Description sommaire

Description de la fonction

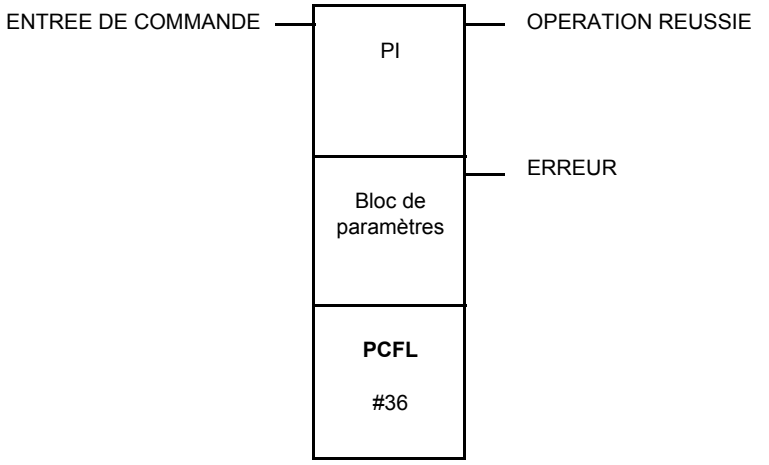
Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Régulation.

La fonction PI effectue des opérations proportionnel–intégrale élémentaires à l'aide des fonctions mathématiques en virgule flottante. Elle dispose des modes de fonctionnement arrêt/manuel/auto. Elle est similaire aux fonctions PID et KPID mais ne comporte pas autant d'options. Elle s'utilise pour une exécution plus rapide des boucles, ou pour des stratégies de boucles internes en cascade.

Représentation : PCFL - PI

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
PI (partie haute)			Sélection de la sous-fonction PI
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 920*.
36 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous-fonction PI (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres PI est de 36 registres.

	Registre	Contenu
Paramètres généraux	Affiché et premier implicite	Entrée réelle, x
	Deuxième implicite	Etat de sortie
	Troisième implicite	Mot d'erreur
	Quatrième implicite	Réservé
	Cinquième implicite	Etat d'entrée
Entrées	Sixième et septième implicite	Consigne, C
	Huitième et neuvième implicite	Sortie manuelle
	10ème et 11ème implicite	Ecart de régulation calculé (erreur), XD
Sorties	12ème implicite	Mode de fonctionnement précédent
	13ème et 14ème implicite	Temps de retard (en ms) depuis la dernière exécution
	15ème et 16ème implicite	Précédent écart système, XD_1
	17ème et 18ème implicite	Composante intégrale de la sortie Y
	19ème et 20ème implicite	Entrée précédente, X_1
	21ème implicite	Etat de fonctionnement précédent
Informations temporelles	22ème implicite	10 ms d'horloge à l'instant n
	23ème implicite	Réservé
	24ème et 25ème implicite	Intervalle de résolution (en ms)
Paramètres d'entrée	26ème et 27ème implicite	Facteur proportionnel, KP
	28ème et 29ème implicite	Temps intégrale, TI
	30ème et 31ème implicite	Limite haute sur la sortie Y
	32ème et 33ème implicite	Limite basse sur la sortie Y
Sortie	34ème et 35ème implicite	Sortie régulée, Y

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	Erreur
2	1 = Limite basse franchie
3	1 = Limite haute franchie
4 à 8	Inutilisé
9 à 16	Bits des sorties normales (drapeaux)

Mot d'erreur

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 11	Inutilisé
12 à 16	Description de l'erreur

Description de l'erreur

Bit					Signification
12	13	14	15	16	
1	0	1	1	0	Constante de temps intégrale négative
1	0	1	0	1	Erreur de limite haute/basse (basse ≥ haute)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	Inutilisé
6	1 = Mode manuel
7	1 = Mode arrêté
8 à 15	Inutilisé
16	1 = Action inverse de sortie de boucle 0 = Action directe de sortie de boucle

PCFL-PID : Algorithmes PID

136

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-PID.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	924
Représentation : PCFL - PID : Algorithmes	925
Description des paramètres	926

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Régulation..

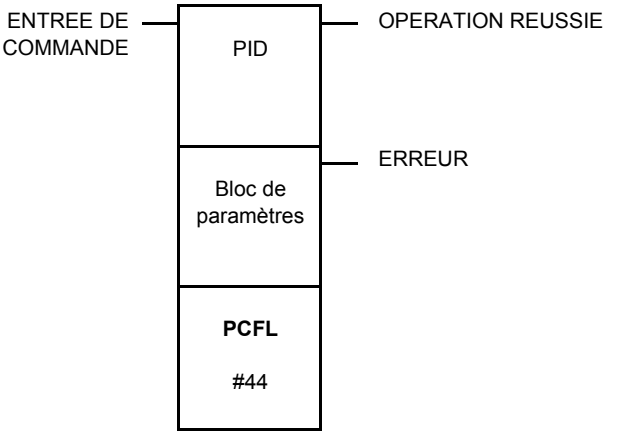
La fonction PID effectue des opérations proportionnelle-intégrale-dérivée (PID) ISA non interactives utilisant les calculs en virgule flottante. Du fait qu'elle utilise les opérations en VF (à l'inverse de PID2), les erreurs d'arrondi sont négligeables.

vous *p. 81* dans la section "Informations générales".

Représentation : PCFL - PID : Algorithmes

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
PID (partie haute)			Sélection de la sous-fonction PID
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 926*.
44 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous-fonction PID (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres KPID est de 44 registres.

	Registre	Contenu
Paramètres généraux	Affiché et premier implicite	Entrée réelle, x
	Deuxième implicite	Etat de sortie
	Troisième implicite	Mot d'erreur
	Quatrième implicite	Réservé
	Cinquième implicite	Etat d'entrée
Entrées	Sixième et septième implicite	Consigne, C
	Huitième et neuvième implicite	Sortie manuelle
	10ème et 11ème implicite	Jonction de somme, Pied (Bias)
Sorties	12ème et 13ème implicite	Erreur, XD
	14ème implicite	Mode de fonctionnement précédent
	15ème et 16ème implicite	Temps écoulé (en ms) depuis la dernière exécution
	17ème et 18ème implicite	Précédent écart système, XD_1
	19ème et 20ème implicite	Entrée précédente, X_1
	21ème et 22ème implicite	Terme intégral de la sortie Y, YI
	23ème et 24ème implicite	Terme différentiel de la sortie Y, YD
	25ème et 26ème implicite	Terme proportionnel de la sortie Y, YP
	27ème implicite	Etat de fonctionnement précédent
Informations temporelles	28ème implicite	Heure actuelle
	29ème implicite	Réservé
Entrées	30ème et 31ème implicite	Intervalle de résolution (en ms)
	34ème et 35ème implicite	Temps intégrale, TI
	36ème et 37ème implicite	Durée d'action dérivée, TD
	38ème et 39ème implicite	Limite haute sur la sortie Y
	40ème et 41ème implicite	Limite basse sur la sortie Y
	42ème et 43ème implicite	Sortie régulée, Y

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	Erreur
2	1 = Limite basse franchie
3	1 = Limite haute franchie
4 à 8	Inutilisé
9 à 16	Bits des sorties normales (drapeaux)

Mot d'erreur

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 11	Inutilisé
12 à 16	Description de l'erreur

Description de l'erreur

Bit					Signification
12	13	14	15	16	
1	0	1	1	1	Constante de temps dérivée négative
1	0	1	1	0	Constante de temps intégrale négative
1	0	1	0	1	Erreur de limite haute/basse (basse ≥ haute)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	Inutilisé
6	1 = Mode manuel
7	1 = Mode arrêté
8	Inutilisé
9	1 = Exécution algorithme proportionnelle
10	1 = Exécution algorithme intégrale
11	1 = Exécution algorithme dérivée
12	1 = Exécution algorithme dérivée sur base x 0 = Exécution algorithme dérivée sur base xd
13 à 15	Inutilisé
16	1 = Action inverse de sortie de boucle 0 = Action directe de sortie de boucle

PCFL-RAMP : Rampe vers la consigne à pente constante

137

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-RAMP.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	930
Représentation : PCFL - RAMP : Rampe vers la consigne à pente constante	931
Description des paramètres	932

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction RAMP vous permet d'effectuer une rampe linéaire vers une consigne cible à une pente d'approche définie.

Vous devez définir :
- la consigne cible, dans les mêmes unités que celles définies pour le contenu du registre d'entrée
- le taux d'échantillonnage
- une pente positive vers la consigne cible, les pentes négatives sont illégales

La direction de la rampe dépend de la relation entre la consigne à atteindre et l'entrée, c'est-à-dire si x < C, la rampe monte ; si x > C, la rampe descend.

Vous pouvez utiliser un drapeau pour l'initialisation après un certain temps d'arrêt. La fonction mémorisera un nouvel échantillon, puis attendra pendant un cycle pour récupérer le deuxième échantillon. Les calculs seront ignorés pendant un cycle et la sortie sera laissée tel quel, après quoi la rampe reprendra.

RAMP s'arrête lorsque l'opération de rampe complète est terminée (sur plusieurs cycles), et retourne un message DXDONE.

Démarrage de la rampe

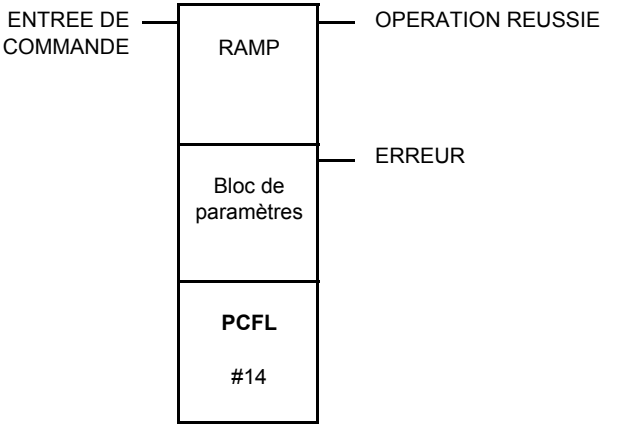
Les étapes suivantes doivent être effectuées lorsque vous démarrez la rampe (vers le haut ou le bas) et à chaque fois que vous devez démarrer ou redémarrer la rampe.

Etape	Action
1	Réglez le bit 1 des bits d'entrée standard sur "1" (troisième registre implicite du bloc de paramètres).
2	Rebasculez l'entrée haute (entrée active) sur l'instruction. La rampe va alors commencer à monter (descendre) à partir de la valeur initiale préalablement configurée au-dessus (en dessous) de la consigne configurée précédemment. Surveillez le 12 ème registre implicite du bloc de paramètres pour connaître la valeur de rampe courante en virgule flottante.

Représentation : PCFL - RAMP : Rampe vers la consigne à pente constante

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
RAMP (partie haute)			Sélection de la sous-fonction RAMP
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 932*.
14 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous-fonction RAMP (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres RAMP est de 14 registres.

Registre	Contenu
Affiché et premier implicite	Consigne (entrée)
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième implicite	Registre de temps
Cinquième implicite	Réservé
Sixième et septième implicite	Δt (en ms) depuis la dernière exécution
Huitième et neuvième implicite	Intervalle de résolution (en ms)
10ème et 11ème implicite	Taux de changement (par seconde) vers la consigne
12ème et 13ème implicite	Sortie

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Inutilisé
5	1 = la pente de la rampe est négative
6	1 = rampe terminée 0 = rampe en cours
7	1 = rampe descendante
8	1 = rampe montante
9 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5 à 16	Inutilisé

Sortie haute (commande réussie)

La sortie haute de la sous-fonction PCFL - RAMP passe à l'état actif à chaque incrémentation/décrémentation successive réussie de la rampe de bits. Cela se produit si rapidement que la sortie haute semble active en permanence. Cette sortie haute ne doit **PAS** être utilisée comme "bit d'exécution de rampe".

Surveillez le bit 6 de l'état de sortie (second registre implicite du bloc de paramètres) comme "bit d'exécution de rampe".

PCFL-RATE : Calcul du taux dérivé sur une durée définie

138

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-RATE.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	936
Représentation : PCFL - RATE : Calcul du taux dérivé sur une durée définie	937
Description des paramètres	938

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction RATE calcule le taux de variation sur les deux dernières valeurs d'entrée. Si vous positionnez un drapeau d'initialisation, la fonction enregistre un échantillon et active les drapeaux appropriés.

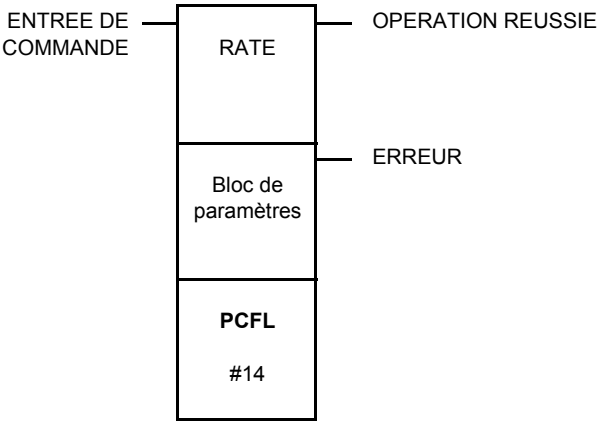
Si une division par zéro est tentée, la fonction retourne un message DXERROR.

Elle retourne un message DXDONE lorsque l'opération est terminée avec succès.

Représentation : PCFL - RATE : Calcul du taux dérivé sur une durée définie

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
RATE (partie haute)			Sélection de la sous-fonction RATE
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *Bloc de paramètres (partie médiane), p. 938*.
14 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous-fonction RATE (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres RATE est de 14 registres.

Registre	Contenu
Affiché et premier implicite	Entrée courante
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième implicite	Registre de temps
Cinquième implicite	Réservé
Sixième et septième implicite	Δt (en ms) depuis la dernière exécution
Huitième et neuvième implicite	Intervalle de résolution (en ms)
10ème et 11ème implicite	Dernière entrée
12ème et 13ème implicite	Résultat

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Inutilisé
9 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5 à 16	Inutilisé

PCFL-RATIO : Régulateur de rapport 4 positions

139

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-RATIO.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	940
Représentation : PCFL - RATIO : Régulateur de rapport 4 positions	941
Description des paramètres	942

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Régulation.

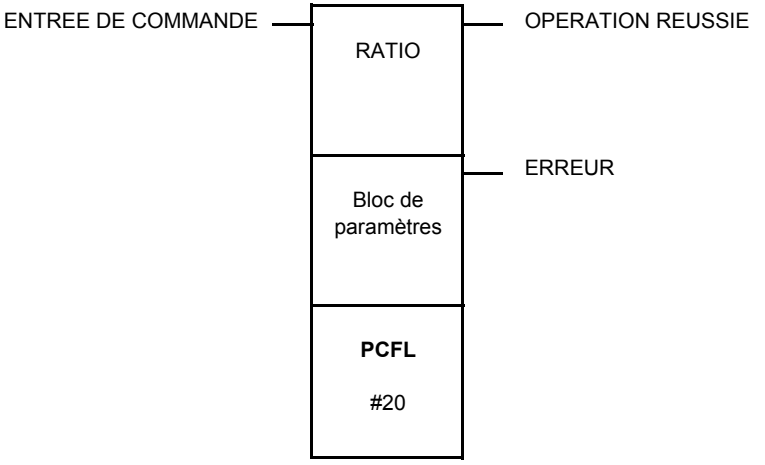
La fonction RATIO offre un régulateur de rapport 4 positions. La régulation de rapport s'utilise dans les applications où un ou plusieurs ingrédients bruts dépendent d'un ingrédient primaire. L'ingrédient primaire est mesuré, et la mesure est convertie en unités physiques via une fonction AIN. La valeur convertie sert à régler les cibles des autres entrées proportionnalisées.

Les sorties du régulateur de rapport peuvent générer des consignes pour d'autres automates. Elles peuvent également être utilisées dans une structure en boucle ouverte pour les applications dans lesquelles le retour n'est pas nécessaire.

Représentation : PCFL - RATIO : Régulateur de rapport 4 positions

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
RATIO (partie haute)			Sélection de la sous-fonction RATIO
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 942*.
20 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous-fonction RATIO (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres RATIO est de 20 registres.

Registre	Contenu
Affiché et premier implicite	Entrée réelle
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Rapport de l'entrée 1
Sixième et septième implicite	Rapport de l'entrée 2
Huitième et neuvième implicite	Rapport de l'entrée 3
10ème et 11ème implicite	Rapport de l'entrée 4
12ème et 13ème implicite	Sortie de l'entrée 1
14ème et 15ème implicite	Sortie de l'entrée 2
16ème et 17ème implicite	Sortie de l'entrée 3
18ème et 19ème implicite	Sortie de l'entrée 4

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 9	Inutilisé
10	1 = paramètre(s) hors limites
11	1 = aucune entrée activée
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	1 = entrée 4 active
6	1 = entrée 3 active
7	1 = entrée 2 active
8	1 = entrée 1 active
9 à 16	Inutilisé

PCFL-RMPLN : Rampe logarithmique vers consigne

140

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-RMPLN.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	946
Représentation : PCFL - RMPLN : Rampe logarithmique vers consigne	947
Description des paramètres	948

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

La fonction RMPLN vous permet d'évoluer en rampe logarithmique vers une consigne cible à une pente d'approche définie. A chaque appel successif, elle calcule la sortie jusqu'à ce qu'elle se trouve dans une plage neutre définie (PN). PN est nécessaire car la distance incrémentale effectuée par la rampe décroît à chaque exécution.

Vous devez définir :

- la consigne cible, dans les mêmes unités que celles définies pour le contenu du registre d'entrée
- le taux d'échantillonnage
- la constante de temps utilisée pour la rampe logarithmique, laquelle vaut la durée mise pour atteindre 63,2% de la nouvelle consigne

Pour de meilleurs résultats, utilisez un $t \geq 4 * \Delta t$. Cela assurera une granularité suffisante de la réponse de sortie.

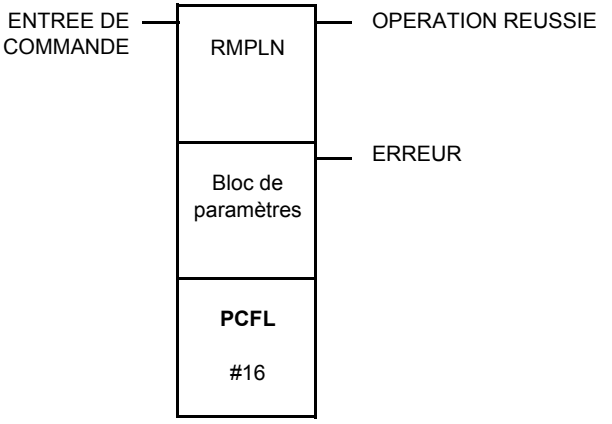
Vous pouvez utiliser un drapeau pour l'initialisation après un certain temps d'arrêt. La fonction mémorisera un nouvel échantillon, puis attendra pendant un cycle pour récupérer le deuxième échantillon. Les calculs seront ignorés pendant un cycle et la sortie sera laissée tel quel, après quoi la rampe reprendra.

RMPLN est terminée lorsque l'entrée atteint la consigne cible + la PN déterminée, et retourne un message DXDONE.

Représentation : PCFL - RMPLN : Rampe logarithmique vers consigne

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
RMPLN (partie haute)			Sélection de la sous-fonction RMPLN
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie. Pour plus d'informations, reportez-vous *p. 948*.
16 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous-fonction RMPLN (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres RMPLN est de 16 registres.

Registre	Contenu
Affiché et premier implicite	Consigne (entrée)
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième implicite	Registre de temps
Cinquième implicite	Réservé
Sixième et septième implicite	Δt (en ms) depuis la dernière exécution
Huitième et neuvième implicite	Intervalle de résolution (en ms)
10ème et 11ème implicite	Constante de temps, τ, (en seconde) de la rampe exponentielle vers la consigne visée
12ème et 13ème implicite	PN (en unités physiques)
14ème et 15ème implicite	Sortie

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Inutilisé
5	1 = PN ou τ configuré avec un nombre négatif
6	1 = rampe terminée 0 = rampe en cours
7	1 = rampe descendante
8	1 = rampe montante
9 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5 à 16	Inutilisé

PCFL-SEL : Sélection des entrées

141

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-SEL.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	950
Représentation : PCFL - SEL : Sélection d'entrée haute/basse/moyenne	951
Description des paramètres	952

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Traitement du signal.

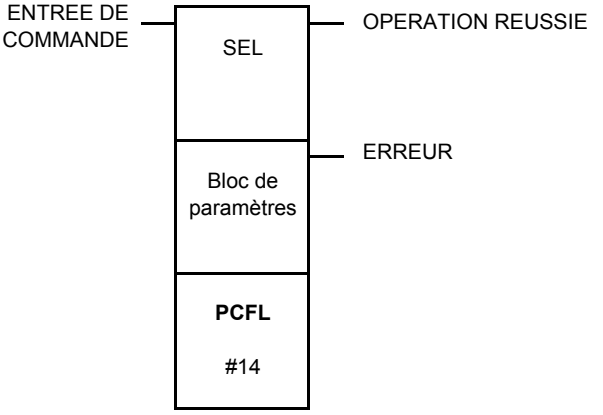
La fonction SEL compare jusqu'à quatre entrées et effectue un choix basé soit sur la valeur la plus haute, la plus basse ou moyenne. A vous de choisir les entrées à comparer et le critère de comparaison. La sortie est une copie de l'entrée sélectionnée.

SEL retourne un message DXDONE lorsque l'opération est terminée.

Représentation : PCFL - SEL : Sélection d'entrée haute/basse/moyenne

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
SEL (partie haute)			Sélection de la sous–fonction SEL
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 952*.
14 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction SEL (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres SEL est de 14 registres.

Registre	Contenu
Affiché et premier implicite	Réservé
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième et cinquième implicite	Entrée 1
Sixième et septième implicite	Entrée 2
Huitième et neuvième implicite	Entrée 3
10ème et 11ème implicite	Entrée 4
12ème et 13ème implicite	Sortie

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 9	Inutilisé
10	Modes de sélection invalides
11	Aucune entrée sélectionnée
12 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	1 = valide l'entrée 1 0 = invalide l'entrée 1
6	1 = valide l'entrée 2 0 = invalide l'entrée 2
7	1 = valide l'entrée 3 0 = invalide l'entrée 3
8	1 = valide l'entrée 4 0 = invalide l'entrée 4
9 à 10	Mode de sélection
11 à 16	Inutilisé

Mode de sélection

Bit		Signification
9	**10**	
0	0	Sélection de la valeur moyenne
0	1	Sélection de la valeur haute
1	0	Sélection de la valeur basse
1	1	réservé / invalide

PCFL-TOTAL : Totalisateur de flux

142

Présentation

Introduction

Ce chapitre décrit la sous-fonction PCFL-TOTAL.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	956
Représentation : PCFL - TOTAL - Totalisateur de flux	957
Description des paramètres	958

Description sommaire

Description de la fonction

Note : Cette instruction est une sous-fonction de l'instruction PCFL. Elle fait partie de la catégorie Régulation.

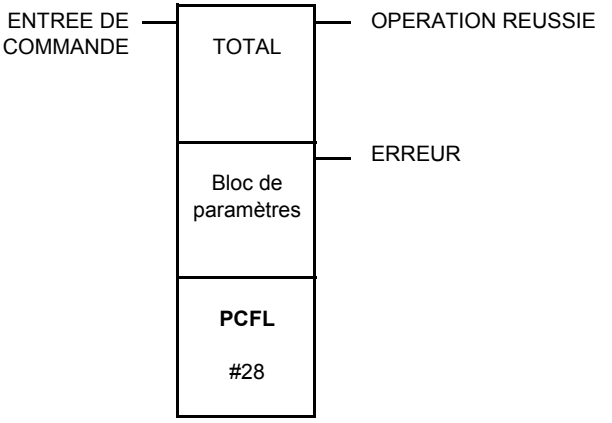
La fonction TOTAL est un totalisateur de flux pour les applications de traitements par lots (batch). Le signal d'entrée comporte les unités de poids ou de volume par unité de temps. Le totalisateur intègre l'entrée en fonction du temps.

L'algorithme délivre trois sorties :
- la somme de l'intégrale
- le reste à doser
- la sortie vanne (en unités physiques).

Représentation : PCFL - TOTAL - Totalisateur de flux

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = valide la fonction de régulation de procédé définie
TOTAL (partie haute)			Sélection de la sous–fonction TOTAL
Bloc de paramètres (partie médiane)	4x	INT, UINT	Premier d'un bloc de registres de sortie successifs dans lesquels sont mémorisés les paramètres de la sous–fonction définie Pour plus d'informations, reportez-vous *p. 959*.
28 (partie basse)		INT, UINT	Longueur du bloc de paramètres de la sous–fonction TOTAL (ne peut pas être changée)
Sortie haute	0x	Aucun	Etat actif = opération réussie
Sortie basse	0x	Aucun	Etat actif = erreur

Description des paramètres

Mode de fonctionnement

La fonction utilise jusqu'à trois consignes différentes :

- une consigne débit lent
- une consigne cible
- une consigne auxiliaire de débit lent

La consigne cible sert à la quantité totale à mesurer. Ici, la sortie est mise à zéro.

La consigne débit lent correspond au point de coupure lorsque la sortie doit décroître depuis le débit maximum jusqu'à un certain pourcentage du débit maximum, de sorte que la consigne cible soit obtenue avec une meilleure granularité.

La consigne débit lent auxiliaire est optionnelle. Elle sert à obtenir un autre niveau de granularité. Si cette consigne est validée, la sortie est encore réduite à 10 % de la sortie lente.

Le totalisateur prend zéro comme point de départ. La consigne doit être une valeur positive.

En fonctionnement normal, la sortie vanne est réglée à 100 % du débit lorsque la valeur intégrée se trouve en dessous de la consigne de débit lent. Lorsque la somme de l'intégrale dépasse la consigne de débit lent, le débit de la vanne devient un pourcentage programmable du débit maximal. Lorsque la somme atteint la consigne souhaitée, la sortie vanne est réglée à un débit de 0 %.

Les consignes peuvent être relatives ou absolues. Avec une consigne relative, on utilise l'écart entre la dernière somme et la consigne. Dans les autres cas, la somme est utilisée en comparaison absolue avec la consigne.

Il existe une option permettant de faire arrêter l'opération d'intégration par le système.

Lorsque l'opération est terminée, la somme de sortie est conservée pour une utilisation ultérieure. Vous avez la possibilité d'effacer cette somme. Dans certaines applications, il est important de la sauvegarder, par exemple si les mesures ou les récepteurs de charge ne peuvent pas traiter le lot en une seule charge, et que les mesures se font en plusieurs fois, s'il y a plusieurs cuves à remplir pour un lot et que vous voulez conserver un suivi des sommes du lot et de la production.

Bloc de paramètres (partie médiane)

La longueur du bloc de paramètres TOTAL est de 28 registres.

Registre	Contenu
Affiché et premier implicite	Entrée réelle
Deuxième implicite	Etat de sortie
Troisième implicite	Etat d'entrée
Quatrième implicite	Registre de temps
Cinquième implicite	Réservé
Sixième et septième implicite	Δt (en ms) depuis la dernière exécution
Huitième et neuvième implicite	Intervalle de résolution (en ms)
10ème et 11ème implicite	Dernière entrée, X_1
12ème et 13ème implicite	Valeur d'initialisation
14ème et 15ème implicite	Consigne, cible
16ème et 17ème implicite	Consigne, débit lent
18ème et 19ème implicite	% du débit maxi pour la consigne de débit lent
20ème et 21ème implicite	Débit maxi
22ème et 23ème implicite	Reste à faire de la consigne
24ème et 25ème implicite	Somme résultante
26ème et 27ème implicite	Sortie de l'organe final de régulation

Etat de sortie

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 2	Inutilisé
3 à 4	0 0 = état repos 0 1 = débit lent 1 0 = débit maxi
5	1 = opération effectuée
6	1 = totalisation en cours
7	1 = dépassement de la consigne de plus de 5%
8	1 = paramètre(s) hors limites
9 à 16	Bits des sorties normales (drapeaux)

Etat d'entrée

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Bits d'entrées normales (drapeaux)
5	1 = réinitialisation de la somme
6	1 = pause de l'intégration
7	1 = écart consigne 0 = consigne absolue
8	1 = utilisation de la consigne auxiliaire de débit lent
9 à 16	Inutilisé

PEER : Transmission PEER

143

Présentation

Introduction

Ce chapitre décrit l'instruction PEER.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	962
Représentation : PEER : Transfert identique de Modbus II	963
Description des paramètres	964

Description sommaire

Description de la fonction

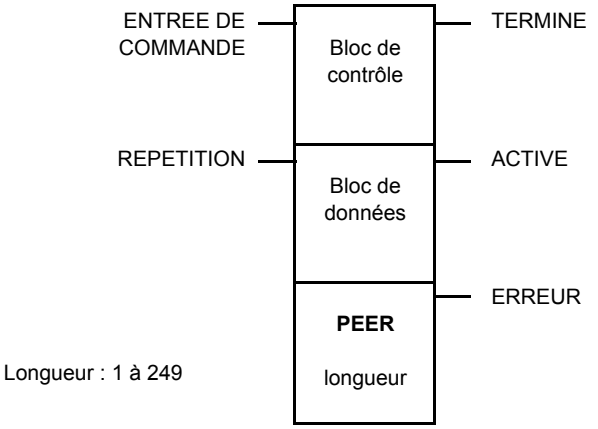
Note : Cette instruction n'est disponible que si vous avez décompacté et installé les instructions chargeables DX. Vous trouverez de plus amples informations dans *p. 107*.

Les modules optionnels de l'interface S975 Modbus II utilisent deux blocs fonction chargeables : MBUS et PEER. L'instruction PEER peut déclencher des transmissions de messages identiques avec 16 appareils à la fois sur Modbus II. Dans une transmission PEER, vous ne pouvez écrire que des données registre.

Représentation : PEER : Transfert identique de Modbus II

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Valide la transmission MBUS
Entrée médiane	0x, 1x	Aucun	Répète la transmission dans le même cycle
Bloc de contrôle (partie haute)	4x	INT, UINT, WORD	Premier de 19 registres successifs du bloc de contrôle PEER. Pour plus d'informations, reportez-vous *p. 964*.
Bloc de données (partie médiane)	4x	INT, UINT	Premier registre d'un bloc de données que la fonction PEER doit transmettre
Longueur (partie basse)		INT, UINT	Longueur, c'est-à-dire nombre de registres de sortie, du bloc de données ; dans la plage : 1 à 249.
Sortie haute	0x	Aucun	Transmission achevée
Sortie médiane	0x	Aucun	Transmission en cours ou démarrage d'une nouvelle transmission
Sortie basse	0x	Aucun	Erreur détectée dans la transmission

Description des paramètres

Bloc de contrôle (partie haute)

Le registre 4x mémorisé en partie haute est le premier de 19 registres successifs du bloc de contrôle PEER.

Registre	Fonction
Affiché	Indique l'état des transmissions de chaque équipement, le bit à l'extrémité gauche représentant l'état de l'équipement nº1 et le bit à l'extrémité droite l'état de l'équipement nº16 : 0 = OK, 1 = erreur de transmission
Premier implicite	Définit la référence du premier registre 4x à inscrire dans le récepteur ; la valeur 0 n'est pas admise dans ce champ et engendrera une erreur (la sortie basse s'activera)
Deuxième implicite	Temps alloué pour achever une transmission avant qu'une erreur ne soit déclarée ; exprimé en multiple de 10 ms, p. ex. 100 correspond à 1 000 ms. Le délai d'attente par défaut est de 250 ms.
Troisième implicite	L'adresse du port 3 Modbus du premier récepteur ; plage d'adresses : 1 à 255 (0 = aucune transmission demandée)
Quatrième implicite	L'adresse du port 3 Modbus du deuxième récepteur ; plage d'adresses : 1 à 255 (0 = aucune transmission demandée)
. . .	. . .
18ème implicite	L'adresse du port 3 Modbus du seizième récepteur (plage d'adresses : 1 à 255)

PID2 : Proportionnelle–intégrale–dérivée

144

Présentation

Introduction

Ce chapitre décrit l'instruction PID2.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	966
Représentation : PID2 : Proportionnelle–intégrale–dérivée	967
Description détaillée	969
Description des paramètres	972
Erreurs d'exécution	977

Description sommaire

Description de la fonction

L'instruction PID2 met en oeuvre un algorithme effectuant des fonctions proportionnelles–intégrales–dérivées. L'algorithme ajuste le fonctionnement en boucle fermée d'une manière similaire aux régulateurs pneumatiques traditionnels en boucle fermée et aux régulateurs à électronique analogique en boucle fermée. Il utilise un filtre dérivateur (RGL) sur la VP comme il est utilisé pour la composante dérivée, éliminant ainsi par filtrage les sources de bruit VP à fréquence élevée (aléatoires et générés par le procédé).

Formules

Régulation proportionnelle

$$M_V = K_1 E + \text{bias}$$

Régulation proportionnelle–intégrale

$$M_V = K_1 \left(E + K_2 \int_0^t E \Delta t \right)$$

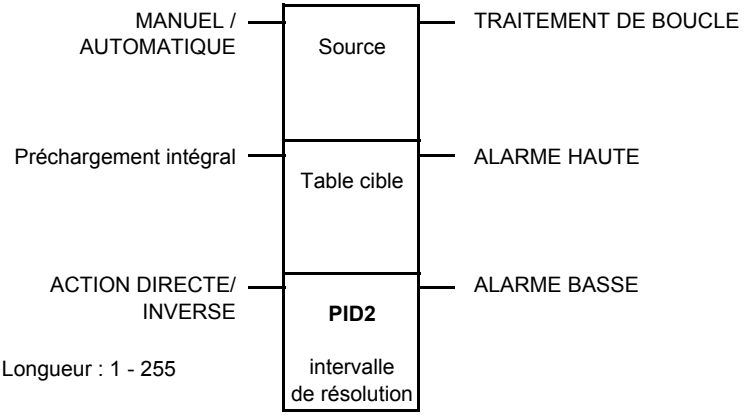
Régulation proportionnelle–intégrale–dérivée

$$M_V = K_1 \left(E + K_2 \int_0^t E \Delta t + K_3 \frac{\Delta PV}{\Delta t} \right)$$

Représentation : PID2 : Proportionnelle–intégrale–dérivée

Symbole

Représentation de l'instruction

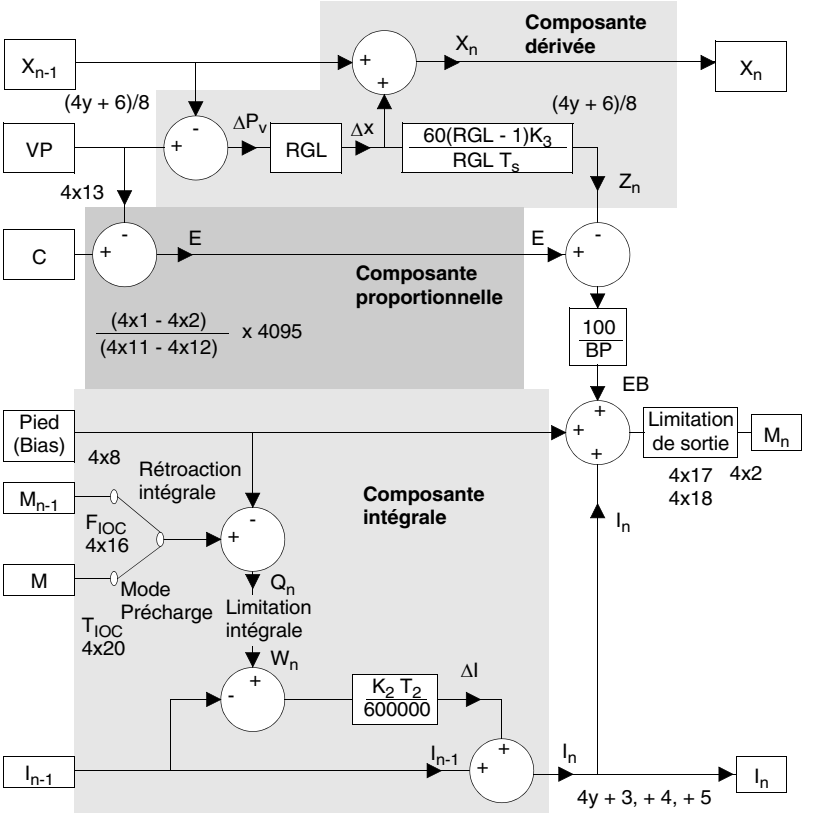


Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	0 = Mode manuel 1 = Mode auto
Entrée médiane	0x, 1x	Aucun	0 = Préchargement intégrale à l'état repos 1 = Préchargement intégrale à l'état actif
Entrée basse	0x, 1x	Aucun	0 = La sortie augmente avec E 1 = La sortie diminue avec E
Source (partie haute)	4x	INT, UINT	Premier de 21 registres de sortie successifs du bloc source. Pour plus d'informations, reportez-vous *p. 972*.
Cible (partie médiane)	4x	INT, UINT	Premier de neuf registres de sortie successifs utilisé pour le calcul PID2. Ne rien charger dans ces registres !.Pour plus d'informations, reportez-vous *p. 975*.
Intervalle de résolution (partie basse)		INT, UINT	Contient un nombre compris entre 1 et 255, indiquant le nombre de fois que la fonction doit être effectuée.
Sortie haute	0x	Aucun	1 = Paramètre utilisateur invalide ou boucle active mais non traitée
Sortie médiane	0x	Aucun	1 = VP ≥ limite supérieure de l'alarme
Sortie basse	0x	Aucun	1 = VP ≤ limite inférieure de l'alarme

Description détaillée

Schéma fonctionnel



Ces éléments du schéma fonctionnel ont la signification suivante :

Elément	Signification
E	Erreur, exprimée en unités analogiques brutes
C	Consigne, comprise entre 0 et 4095
VP	Variable procédé, comprise entre 0 et 4095
x	VP filtrée
K2	Constante de gain du mode intégrale, exprimée en 0.01 min^{-1}
K3	Constante de gain du mode dérivée, exprimée en centièmes de minute

Elément	Signification
RGL	Constante de filtrage de la dérivée, dans la plage de 2 à 30
Ts	Temps de résolution, exprimé en centièmes de seconde
BP	Bande proportionnelle, dans la plage 5 à 500 %
Pied (bias)	Facteur de pied (bias) de sortie de boucle, dans la plage 0 à 4095
M	Sortie de boucle
EB	Erreur brute, la composante proportionnelle–dérivée à la sortie de la boucle
Z	Composante de mode dérivée de l'EB
Qn	Sortie de boucle sans pied (bias)
F	Retour, dans la plage 0 à 4095
I	Composante de mode intégrale de la sortie de boucle
I_{bas}	C antisaturation basse, dans la plage 0 à 4095
I_{haut}	C antisaturation haute, dans la plage 0 à 4095
K1	100/BP

Note : Le calcul de la composante de mode intégrale intègre réellement la différence entre la sortie et la somme de l'intégrale, ce qui revient effectivement au même résultat que d'intégrer l'erreur.

Régulation proportionnelle

A l'aide de la régulation uniquement proportionnelle (P), vous pouvez calculer la variable manipulée en multipliant l'erreur par une constante proportionnelle, K1, et en ajoutant ensuite un pied (bias). voir *p. 966*.

Néanmoins, les conditions du procédé dans la plupart des applications sont modifiées par d'autres variables système de manière à ce que le pied (bias) ne reste pas invariable ; il en résulte une erreur de décalage dans laquelle VP est constamment décalée par rapport à C. Ce problème limite les capacités de la régulation uniquement proportionnelle.

Note : La valeur de la partie intégrale (dans les registres 4y + 3, 4y + 4 et 4y + 5) est toujours utilisée, même lorsque la composante intégrale n'est pas activée. L'utilisation de cette valeur est nécessaire pour préserver un transfert sans mémoire annexe entre les composantes. Si vous voulez invalider un transfert sans mémoire, ces trois registres doivent être mis à zéro. En mode manuel, les modifications apportées au niveau de la consigne ne prennent pas effet tant que les trois registres ci-dessus ne sont pas remis à zéro et que le mode n'est pas repassé en automatique. Le transfert n'est pas sans à-coup.

Régulation proportionnelle–intégrale

Pour éliminer cette erreur de décalage sans vous forcer à changer le pied (bias) manuellement, une fonction intégrale peut être ajoutée à l'équation de régulation. voir *Formules, p. 966*.

La régulation proportionnelle–intégrale (PI) élimine le décalage en intégrant E comme une fonction du temps. K1 est la constante d'intégration exprimée sous forme de rép/min. Tant que $E \neq 0$, la valeur de l'intégrateur augmente (ou diminue), ajustant ainsi S. Ceci continue jusqu'à ce que l'erreur de décalage soit éliminée.

Régulation proportionnelle–intégrale–dérivée

Si vous souhaitez ajouter une composante dérivée à l'équation de régulation pour réduire les effets des modifications fréquentes de charges ou pour remplacer la fonction intégrale afin d'arriver à la consigne C plus rapidement. voir *Formules, p. 966*.

La régulation proportionnelle–intégrale–dérivée (PID) peut être utilisée pour économiser de l'énergie au sein du procédé ou en tant que mesure de sécurité dans le cas d'une variation soudaine et inattendue dans le déroulement du procédé. K3 est la constante de temps dérivée exprimée en min. DVP est la variation de la variable procédé sur une période de temps de Δt.

Exemple

Pour trouver un exemple de régulation de niveau PID2, reportez-vous à la section Exemple PID2 de régulation de niveau.

Description des paramètres

Bloc source (partie haute)

Le registre 4x saisi dans la partie haute est le premier de 21 registres de sortie successifs d'un bloc source. Le contenu des registres implicites cinq à huit détermine si l'opération est en P, PI ou PID :

Opération	Cinquième implicite	Sixième implicite	Septième implicite	Huitième implicite
P	Etat actif			Etat actif
PI	Etat actif	Etat actif		
PID	Etat actif	Etat actif	Etat actif	

Le bloc source comprend les affectations de registre suivantes :

Registre	Nom	Contenu
Affiché	VP à l'échelle	Appelée par le bloc à chaque analyse ; une mise à l'échelle linéaire est effectuée dans le registre 4x + 13 à l'aide des limites hautes et basses des registres 4x + 11 et 4x + 12 : VP mise à l'échelle = (4x13 / 4095) * (4x11 - 4x12) + 4x12
Premier implicite	C	Vous devez définir la consigne en unités physiques ; la valeur doit être inférieure à la valeur du 11ème registre implicite et supérieure à la valeur du 12ème registre implicite
Deuxième implicite	S	Appelée par le bloc à chaque exécution de la boucle ; elle est limitée dans la plage de 0 à 4095, rendant la sortie compatible avec un module de sortie analogique ; le registre de la variable manipulée peut être utilisé pour d'autres calculs de processeur tels que les boucles en cascade.
Troisième implicite	Limite supérieure de l'alarme	Renseignez ce registre pour définir une alarme haute pour la VP (égale ou supérieure à C) ; saisissez la valeur en unités physiques dans la plage donnée dans les 11ème et 12ème registres implicites
Quatrième implicite	Limite inférieure de l'alarme	Renseignez ce registre pour définir une alarme basse pour la VP (égale ou inférieure à C); saisissez la valeur en unités physiques dans la plage donnée dans les 11ème et 12ème registres implicites
Cinquième implicite	Bande proportionnelle	Renseignez ce registre avec la constante proportionnelle souhaitée dans la plage 5 à 500 ; plus le nombre est petit, plus la composante proportionnelle est grande. Pour le fonctionnement de PID2, ce registre doit contenir une valeur admise.

Registre	Nom	Contenu
Sixième implicite	Constante de temps de l'intégrale	Renseignez ce registre pour ajouter une action intégrale au calcul. Saisissez une valeur comprise entre 0000 et 9999 pour définir une plage de 00,00 à 99,99 répétitions/min. Plus le nombre est grand, plus la composante intégrale est importante. Une valeur supérieure à 9999 arrête le calcul PID2.
Septième implicite	Constante de temps de la dérivée	Renseignez ce registre pour ajouter une action intégrale au calcul. Saisissez une valeur comprise entre 0000 et 9999 pour définir une plage de 00,00 à 99,99 répétitions/min. Plus le nombre est grand, plus la composante intégrale est importante. Une valeur supérieure à 9999 arrête le calcul PID2.
Huitième implicite	Pied (Bias)	Renseignez ce registre pour ajouter un pied (bias) à la sortie. La valeur doit être comprise entre 000 et 4095, et ajoutée directement à S, que la composante intégrale soit validée ou non.
Neuvième implicite	Seuil antisaturation haut de l'intégrale	Renseignez ce registre avec le seuil haut de la valeur de sortie (entre 0 et 4095) à partir duquel l'antisaturation prend effet. La mise à jour de l'intégrale est arrêtée si cette valeur est dépassée (elle vaut normalement 4095)
10ème implicite	Seuil antisaturation bas de l'intégrale	Renseignez ce registre avec le seuil bas de la valeur de sortie (entre 0 et 4095) à partir duquel l'antisaturation prend effet (il vaut normalement 0)
11ème implicite	Plage physique supérieure	Renseignez ce registre avec la plus grande valeur délivrée par l'instrument de mesure, par exemple si la plage d'une sonde résistive de température va de 0 à 500 degrés C, la valeur de la plage physique supérieure est 500. Cette valeur doit être un entier positif compris entre 0001 et 9999, correspondant à la mesure analogique brute 4095.
12ème implicite	Plage physique inférieure	Renseignez ce registre avec la plus petite valeur délivrée par l'instrument de mesure. La plage doit être un entier positif compris entre 0 et 9998, et doit être inférieure à la valeur du 11ème registre implicite. Elle correspond à la mesure analogique brute 0.
13ème implicite	Mesure de la valeur analogique brute	Le programme renseigne ce registre avec la VP ; la mesure doit être à l'échelle et linéaire dans la plage 0 à 4095

Registre	Nom	Contenu
14ème implicite	Pointeur du registre compteur de boucle	La valeur que vous chargez dans ce registre pointe vers le registre comptant le nombre de boucles traitées à chaque cycle. L'entrée est déterminée en enlevant le chiffre de poids fort du registre dans lequel l'automate compte les boucles traitées par cycle. Si par exemple l'automate effectue le comptage dans le registre 41236, saisissez 1236 dans le 14ème registre implicite. La même valeur doit être chargée dans le 14ème registre implicite à chaque bloc PID2 du programme.
15ème implicite	Nombre maximum de boucles	Traitées en un cycle : Si le 14ème registre implicite contient une valeur différente de zéro, vous pouvez saisir une valeur dans ce registre afin de limiter le nombre de boucles devant être traitées en un cycle.
16ème implicite	Pointeur de l'entrée retour intégrale :	La valeur que vous chargez dans ce registre pointe vers le registre de sortie contenant la valeur du retour (R) ; retirez le 4 du registre de retour et saisissez les quatre chiffres restants dans ce registre. Les calculs d'intégrales dépendent de la valeur R fonction de S, c'est-à-dire lorsque la sortie PID2 varie de 0 à 4095, R varie de 0 à 4095.
17ème implicite	Limite haute de sortie	La valeur saisie dans ce registre définit la limite supérieure de S (elle vaut normalement 4095)
18ème implicite	Limite basse de sortie	La valeur saisie dans ce registre définit la limite inférieure de S (elle vaut normalement 0)
19ème implicite	Constante de filtrage de la dérivée (RGL)	La valeur saisie dans ce registre définit le degré réel du filtrage dérivée ; la plage est de 2 à 30 ; plus la valeur est petite, plus le filtrage a d'effet.
20ème implicite	Pointeur du préchargement intégrale	La valeur saisie dans ce registre pointe vers le registre de sortie contenant la valeur de l'entrée asservissement (T) ; enlevez le 4 du registre d'asservissement et saisissez les quatre chiffres restants dans ce registre. La valeur du registre T dépend de l'entrée de la composante intégrale à condition que les bits auto et préchargement intégrale soient tous deux vrais.

Cible (partie médiane)

Le registre 4y saisi dans la partie médiane est le premier de neuf registres de sortie successifs utilisés dans les opérations PID2. Vous n'avez pas besoin de renseigner ces registres :

Registre	Nom	Contenu
Affiché	Registre d'état de boucle	Douze des 16 bits de ce registre sont utilisés pour définir l'état de la boucle.
Premier implicite	Bits d'état d'erreur (E)	Ce registre affiche les codes d'erreur PID2.
Deuxième implicite	Registre de temporisation de boucle	Ce registre mémorise la lecture de l'horloge temps réel sur l'horloge système à chaque fois que la boucle est traitée : la différence entre la valeur actuelle de l'horloge et la valeur mémorisée dans le registre correspond au temps écoulé ; si temps écoulé ≥ intervalle de résolution (10 fois la valeur indiquée en partie basse du bloc PID2), la boucle doit alors être traitée au cours de ce cycle
Troisième implicite	Usage interne	Intégrale (partie entière)
Quatrième implicite	Usage interne	Intégrale-fraction 1 (1/3 000)
Cinquième implicite	Usage interne	Intégrale-fraction 2 (1/600 000)
Sixième implicite	VP x 8 (filtré)	Ce registre mémorise le résultat de l'entrée analogique filtrée (du registre 4x14) multiplié par 8. Cette valeur est très utile dans les applications de régulation dérivée
Septième implicite	Valeur absolue de E	Ce registre, mis à jour après chaque traitement de boucle, contient la valeur absolue de (C –VP) ; le bit 8 du registre 4y + 1 donne le signe de E.
Huitième implicite	Usage interne	Intervalle de résolution actuel

Registre d'état de boucle

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	Etat de sortie supérieure (abonné hors tension ou erreur de paramètre)
2	Etat de sortie médiane (alarme haute)
3	Etat de sortie inférieure (alarme basse)
4	Boucle en mode AUTO et temps depuis la dernière résolution ≥ intervalle de résolution
5	Mode désaturation de l'intégrale (pour Rév. B ou ultérieure)
6	Boucle en mode AUTO mais non traitée
7	L'adressage du registre 4x14 par 4x15 est valide
8	Signe de E dans 4y + 7 : ● 0 = + (plus) ● 1 = - (moins)
9	Rév B ou ultérieure
10	Antisaturation haute de l'intégrale jamais à 1
11	Antisaturation intégrale saturée
12	Valeurs négatives dans l'équation
13	Etat entrée inférieure (action directe/inverse)
14	Etat entrée médiane (mode poursuite) : ● 1 = poursuite ● 0 = pas de poursuite
15	Etat entrée supérieure (MAN/AUTO)
16	Le bit 16 est mis à 1 après le démarrage initial ou l'installation de la boucle. Si vous le mettez à 0, les actions suivantes se dérouleront sur un cycle : ● Le registre d'état de la boucle 4y est remis à zéro ● La valeur actuelle de l'horloge temps réel est mémorisée dans le premier registre implicite (4y+1) ● Les valeurs des registres trois à cinq (4y + 2,3) sont remises à zéro ● La valeur du 13ème registre implicite (4x+13) x 8 est mémorisée dans le sixième registre implicite (4y + 6) ● Les registres implicites sept et huit (4y + 7,8) sont remis à 0

Intervalle de résolution (partie basse)

La partie basse indique qu'il s'agit d'une fonction PID2 et contient un nombre compris entre 1 et 255, indiquant combien de fois la fonction doit se dérouler. Le nombre représente une durée exprimée en dixièmes de seconde ; par exemple. le nombre 17 signifie que la fonction PID doit se dérouler toutes les 1,7 s.

Erreurs d'exécution

Bit d'état d'erreur Le premier registre implicite de la cible contient les bits d'état d'erreur :

Code	Explication	Contrôlez ces registres dans le bloc source (partie haute)
0000	Aucune erreur, toutes les validations sont correctes	Aucun
0001	C à l'échelle supérieure à 9999	Premier implicite
0002	Alarme haute supérieure à 9999	Troisième implicite
0003	Alarme basse supérieure à 9999	Quatrième implicite
0004	Bande proportionnelle inférieure à 5	Cinquième implicite
0005	Bande proportionnelle supérieure à 500	Cinquième implicite
0006	Intégrale supérieure à 99,99 r/min	Sixième implicite
0007	Dérivée supérieure à 99,99 min	Septième implicite
0008	Pied (bias) supérieur à 4095	Huitième implicite
0009	Limite supérieure de l'intégrale supérieure à 4095	Neuvième implicite
0010	Limite inférieure de l'intégrale supérieure à 4095	10ème implicite
0011	Borne haute de l'unité physique (U.P.) supérieure à 9999	11ème implicite
0012	Borne basse de l'U.P. supérieure à 9999	12ème implicite
0013	U.P. haute inférieure à U.P. basse	11ème et 12ème implicite
0014	C mise à l'échelle supérieure à U.P. haute	Premier et 11ème implicite
0015	C mise à l'échelle inférieure à U.P. basse	Premier et 12ème implicite
0016	Nb maximum de boucles/cycle > 9999 **Note :**Activé en boucle maximum, c'est-à-dire seulement si 4x15 n'est pas égal à zéro.	15ème implicite
0017	Pointeur de retour intégrale hors limites	16ème implicite
0018	Limite de la sortie haute supérieure à 4095	17ème implicite
0019	Limite de la sortie basse supérieure à 4095	18ème implicite
0020	Limite de la sortie basse supérieure à la limite de la sortie haute	17ème et 18ème implicite
0021	RGL inférieure à 2	19ème implicite
0022	RGL supérieure à 30	19ème implicite

Code	Explication	Contrôlez ces registres dans le bloc source (partie haute)
0023	Pointeur de poursuite R hors limites **Note** Activé seulement si le mode poursuite est enclenché, c'est-à-dire si l'entrée médiane du bloc PID2 reçoit du courant en étant dans le mode AUTO.	20ème implicite et entrée médiane à l'état actif
0024	Pointeur de poursuite R égal à zéro **Note** Activé seulement si le mode poursuite est enclenché, c'est-à-dire si l'entrée médiane du bloc PID2 reçoit du courant en étant dans le mode AUTO.	20ème implicite et entrée médiane à l'état actif
0025	Partie bloquée (cycle écourté) **Note :**Activé en boucle maximum, c'est-à-dire seulement si 4x15 n'est pas égal à zéro. **Note :** Si des blocages apparaissent souvent et que les paramètres sont tous valides, augmentez le nombre maximum de boucles/cycle. Des blocages peuvent aussi apparaître si les registres de comptage utilisés ne sont pas remis à zéro quand il faut.	Aucun
0026	Le pointeur de comptage de boucle est à zéro **Note :**Activé en boucle maximum, c'est-à-dire seulement si 4x15 n'est pas égal à zéro.	14ème et 15ème implicite
0027	Le pointeur de comptage de boucle est hors limites	14ème et 15ème implicite

Description des instructions (de R à Z)

VI

Présentation

Introduction

Dans cette section, les descriptions des instructions sont triées par ordre alphabétique de R à Z.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
145	R --> T : Registre vers Table	981
146	RBIT : Mise à 0 de bit	987
147	READ : Lecture	991
148	RET : Retour d'un sous–programme	997
149	RTTI : Table registre vers entrée	1001
150	RTTO : Table registre vers sortie	1005
151	RTU : Terminal déporté	1009
152	SAVE : Sauvegarde mémoire flash	1015
153	SBIT : Mise à 1 de bit	1019
154	SCIF : Interfaces de commande séquentielle	1023
155	SENS : Détection	1029
156	Liaisons	1035
157	SKP : Omission de réseaux	1039
158	SRCH : Recherche	1043
159	STAT : Etat	1049
160	SU16 : Soustraction de valeurs 16 bits	1077
161	SUB : Soustraction	1081
162	SWAP : Permutation de bits VME	1085
163	TTR : Table vers registre	1089
164	T --> R Table vers registre	1093

Chapitre	Titre du chapitre	Page
165	T --> T : Table vers table	1099
166	Temporisation T.01 : Temporisation au centième de seconde	1105
167	Temporisation T0.1 : Temporisation au dixième de seconde	1109
168	Temporisation T1.0 : Temporisation à la seconde	1113
169	Temporisation T1MS : Temporisation à la milliseconde	1117
170	TBLK : Table vers Bloc	1123
171	TEST : Comparaison de deux valeurs	1129
172	UCTR : Compteur	1133
173	VMER : Lecture de VME	1137
174	VMEW : Ecriture de VME	1143
175	WRIT : Ecrire	1149
176	XMIT : Emission	1155
177	Bloc de communication XMIT	1163
178	Bloc d'état du port XMIT	1175
179	Bloc de conversion XMIT	1183
180	XMRD : Lecture de mémoire étendue	1191
181	XMRD : Ecriture en mémoire étendue	1197
182	XOR : OU exclusif	1203

R --> T : Registre vers Table

145

Présentation

Introduction

Ce chapitre présente l'instruction R → T.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	982
Représentation : R → T : Copie de registre vers table	983
Description des paramètres	985

Description sommaire

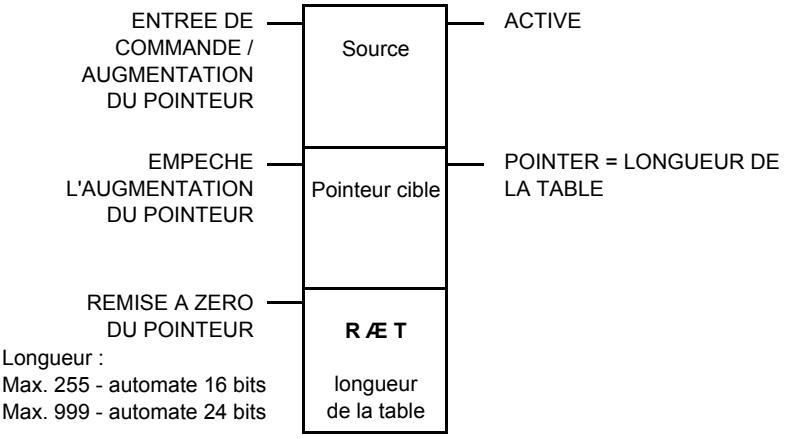
Description de la fonction

L'instruction R→T copie la configuration binaire d'un registre ou d'une chaîne de bits internes successifs mémorisés dans un mot en un registre spécifique situé dans une table. Elle peut gérer le transfert d'un registre/mot par cycle.

Représentation : R → T : Copie de registre vers table

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = copie les données source et incrémente la valeur du pointeur
Entrée médiane	0x, 1x	Aucun	Etat actif = fige la valeur du pointeur
Entrée basse	0x, 1x	Aucun	Etat actif = remet à zéro la valeur du pointeur
Source (partie haute)	0x, 1x, 3x, 4x	INT, UINT, WORD	Données source à copier pendant le cycle en cours
Pointeur cible (partie médiane)	4x	INT, UINT	Table cible dans laquelle les données source seront copiées pendant le cycle
Longueur de la table (partie basse)		INT, UINT	Nombre de registres de la table cible, dans la plage : 1 à 999 Longueur : Max. 255 - automate 16 bits Max. 999 - automate 24 bits
Sortie haute	0x	Aucun	Donne une image de l'entrée haute.
Sortie médiane	0x	Aucun	Etat actif = valeur du pointeur = longueur de la table (l'instruction ne peut pas incrémenter au–delà)

Description des paramètres

Entrée haute

L'entrée de la partie haute déclenche l'opération de transfert DX.

Entrée médiane

Lorsque l'entrée médiane est activée, la valeur se trouvant actuellement dans le pointeur cible est figée, tandis que l'opération DX se poursuit. De nouvelles données sont alors copiées vers la cible, qui écraseront les données copiées au cycle précédent.

Entrée basse

Lorsque l'entrée basse est activée, la valeur du registre du pointeur cible est remise à zéro. Ceci provoque la copie par l'opération de transfert DX suivante des données source dans le premier registre de la table cible.

Données source (partie haute)

Lorsque les registres de type 0x ou 1x sont utilisés :

- la première référence 0x d'une chaîne de 16 bobines ou sorties TOR successives
- la première référence 1x d'une chaîne de 16 entrées TOR.

Pointeur cible (partie médiane)

Le registre 4x mémorisé en partie médiane est un pointeur vers la table cible, dans laquelle les données source seront copiées pendant le cycle. Le premier registre de la table cible est le registre 4x successif suivant immédiatement le pointeur ; autrement dit si le registre du pointeur est 400027, la table cible commence au registre 400028.

La valeur mémorisée dans le registre du pointeur indique le registre de la table cible dans lequel les données source seront copiées. La valeur 0 indique que les données source seront copiées dans le premier registre de la table cible ; la valeur 1 indique que les données source seront copiées dans le deuxième registre de la table cible, etc.

Note : La valeur mémorisée dans le registre du pointeur cible ne peut pas dépasser l'entier de longueur de la table spécifié dans cette partie.

Sorties

R→T peut générer deux sorties possibles, sur les parties haute et médiane. L'état de la sortie de la partie haute est la copie de l'état de l'entrée haute. La sortie médiane est activée lorsque la valeur du registre de pointeur cible est identique à la longueur indiquée pour la table. L'instruction ne peut à cet instant être davantage incrémentée.

RBIT : Mise à 0 de bit

146

Présentation

Introduction

Ce chapitre décrit l'instruction RBIT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	988
Représentation : RBIT : Mise à 0 de bit	989

Description sommaire

Description de la fonction

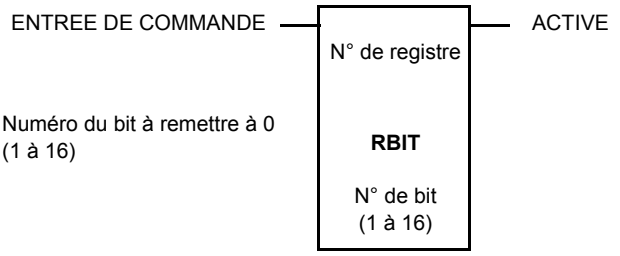
L'instruction de mise à 0 de bit (RBIT) vous permet de mettre à zéro un bit bloqué à l'état actif en activant l'entrée haute. Le bit reste à 0 lorsque l'entrée n'est plus activée. Cette instruction sert à la mise à zéro d'un bit défini à 1 par l'instruction SBIT.

Note : L'instruction RBIT ne suit pas les mêmes règles de placement de réseau que les bobines référencées 0x. Une instruction RBIT ne peut pas être placée en colonne 11 d'un réseau et peut être positionnée à gauche d'autres éléments logiques sur les mêmes échelons du schéma.

Représentation : RBIT : Mise à 0 de bit

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = met le bit indiqué à 0. Le bit reste à 0 lorsque l'entrée n'est plus activée.
N° de registre (partie haute)	4x	WORD	Registre de sortie dont la configuration binaire est contrôlée
N° de bit (partie basse)		INT, UINT	Indique lequel des 16 bits est mis à zéro
Sortie haute	0x	Aucun	Etat actif = le bit indiqué est mis à 0

READ : Lecture

147

Présentation

Introduction

Ce chapitre décrit l'instruction READ.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	992
Représentation : READ : Lecture d'un port ASCII	993
Description des paramètres	995

Description sommaire

Description de la fonction

L'instruction READ permet de lire des données depuis un périphérique d'entrée ASCII (clavier, lecteur code barre, etc.) dans la mémoire de l'automate en passant par son réseau RIO. La connexion au périphérique ASCII s'effectue sur une interface RIO.

Dans la procédure de traitement du message, l'instruction READ effectue les fonctions suivantes :

- vérifie la longueur des champs de données variables
- vérifie si les paramètres de communication ASCII sont corrects, par exemple le numéro de port et le numéro de message
- détecte les erreurs et les enregistre
- signale l'état de l'interface RIO.

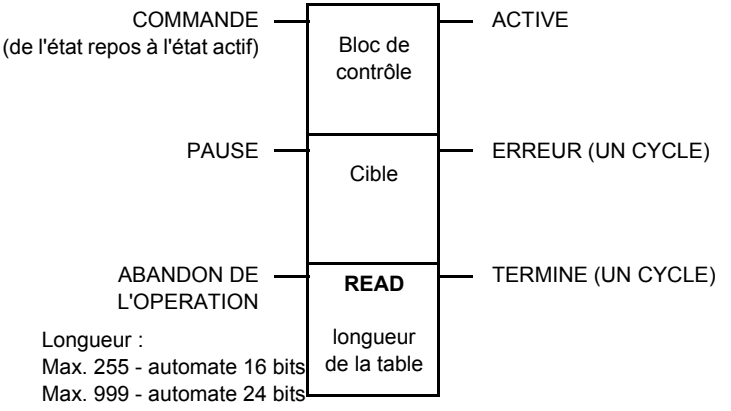
READ nécessite deux tables de registres : une table cible dans laquelle les données variables obtenues (le message) sont stockées et un bloc de contrôle dans lequel les paramètres du port de communication et les paramètres de message sont identifiés.

Vous trouverez de plus amples informations sur la mise en forme de messages à la section *p. 89*.

Représentation : READ : Lecture d'un port ASCII

Symbole

Représentation de l'instruction



Description des paramètres

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = déclenche une opération READ
Entrée médiane	0x, 1x	Aucun	Etat actif = suspend l'opération READ
Entrée basse	0x, 1x	Aucun	Etat actif = abandonne l'opération READ
Bloc de contrôle (partie haute)	4x	INT, UINT, WORD	Bloc de contrôle (premier de sept registres de sortie successifs)
Cible (partie médiane)	4x	INT, UINT, WORD	Table cible
Longueur de la table (partie basse)		INT, UINT	Longueur de la table cible (nombre de registres dans lesquels les données du message seront mémorisées), dans la plage : 1 à 999 Longueur : Max. 255 - automate 16 bits Max. 999 - automate 24 bits
Sortie haute	0x	Aucun	Donne une image de l'entrée haute
Sortie médiane	0x	Aucun	Etat actif = erreur de communication ou l'opération a dépassé les limites de temps (pour un cycle)
Sortie basse	0x	Aucun	Etat actif = opération READ achevée (pour un cycle)

Description des paramètres

Bloc de contrôle (partie haute)

Le registre 4x mémorisé en partie haute est le premier des sept registres de sortie successifs du bloc de contrôle.

Registre	Définition
Affiché	Numéro de port et code d'erreur
Premier implicite	Numéro de message
Deuxième implicite	Nombre de registres nécessaires pour respecter le format
Troisième implicite	Compte du nombre de registres transmis jusque–là
Quatrième implicite	Etat du cycle
Cinquième implicite	Réservé
Sixième implicite	Total de contrôle des registres 0 à 5

Numéro de port et code d'erreur

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Code d'erreur de l'automate
5	Inutilisé
6	Entrée de l'équipement ASCII incompatible avec le format
7	Débordement du tampon d'entrée, données reçues trop rapidement sur le RIOP
8	Erreur USART, RIOP a reçu un octet incorrect
9	Equipement ASCII hors–ligne, contrôle du câblage
10	Format illicite, mauvaise réception RIOP
11	Message ASCII terminé trop tôt (en mode clavier)
12 à 16	Numéro du port de communication (1 à 32)

Code d'erreur API

Bit				Signification
1	**2**	**3**	**4**	
0	0	0	1	Erreur sur l'entrée RIOP depuis le périphérique ASCII
0	0	1	0	Réponse d'anomalie depuis RIOP, données incorrectes
0	0	1	1	Nombre séquentiel de RIOP diffère de la valeur attendue
0	1	0	0	Erreur de total de contrôle du registre utilisateur, souvent due à une modification des registres READ lorsque le bloc est actif
0	1	0	1	Détection d'un numéro de port ou de message invalide
0	1	1	0	Abandon par l'utilisateur, partie basse activée
0	1	1	1	Aucune réponse de la station, erreur de communication
1	0	0	0	Partie abandonnée du fait de l'instruction SKP
1	0	0	1	Zone message brouillée, recharger la mémoire
1	0	1	0	Port non configuré dans l'affectation des E/S
1	0	1	2	Demande ASCII illicite
1	1	0	0	Réponse inconnue du port ASCII
1	1	0	1	Elément ASCII illicite détecté dans la logique utilisateur
1	1	1	1	RIOP de l'automate est arrêté

Cible (partie médiane)

La partie médiane contient le premier registre 4x d'une table cible. Les données variables d'un message READ sont enregistrées dans cette table. La longueur de la table est définie en partie basse.

Considérons ce message READ :

Veuillez saisir le mot de passe **:** **AAAAAAAAAA**

(Texte imbriqué) (Données variables)

Note : Un message READ ASCII peut contenir le texte imbriqué, placé entre guillemets, ainsi que les données variables au format défini, c'est-à-dire le message ASCII. Le champ ASCII sur 10 caractères **AAAAAAAAAA** est le champ de données variables ; les données variables doivent être saisies à l'aide d'un périphérique d'entrée ASCII.

RET : Retour d'un sous–programme

148

Présentation

Introduction

Ce chapitre décrit l'instruction RET.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	998
Représentation : RET : Retour à la logique ordonnancée	999

Description sommaire

Description de la fonction

L'instruction RET peut être utilisée pour retourner sur condition l'analyse logique vers la partie suivant immédiatement le dernier bloc JSR exécuté. Cette instruction ne peut être implémentée que depuis le segment du sous-programme, le dernier segment (non ordonné) du programme utilisateur.

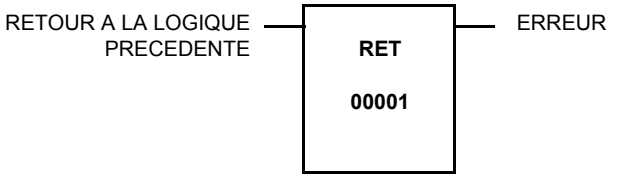
Note : Si un sous–programme ne comporte pas de bloc RET, soit un bloc LAB, soit la fin de logique (selon ce qui arrive en premier) sert de retour par défaut depuis le sous–programme.

vous *p. 105*.

Représentation : RET : Retour à la logique ordonnancée

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = retour à la logique précédente L'état actif renvoie le cycle logique vers la partie suivant immédiatement la dernière instruction JSR exécutée ou vers le point du cycle logique où l'interruption est survenue.
00001		INT, UINT	Valeur constante, ne peut pas être changée
Sortie haute	0x	Aucun	Etat actif = erreur du sous–programme indiqué

RTTI : Table registre vers entrée

149

Présentation

Introduction

Ce chapitre décrit l'instruction RTTI.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : RTTI : Table registre vers entrée	1002
Représentation : RTTI : Table registre vers entrée	1003

Description sommaire : RTTI : Table registre vers entrée

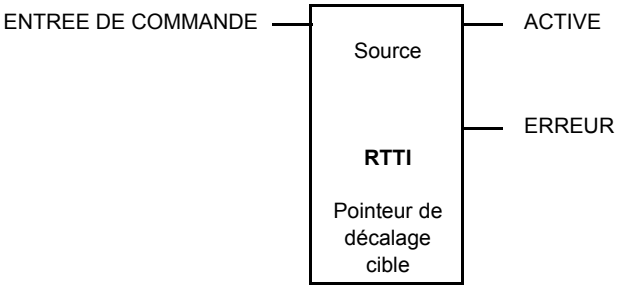
Description de la fonction

Le bloc Table registre vers entrée est l'une des quatre instructions de remplacement du 484. Il copie le contenu d'un registre d'entrée ou de sortie dans un autre registre d'entrée ou de sortie. Ce registre cible est désigné par le registre d'entrée déterminé par la constante en partie basse. Seule une telle opération peut être prise en charge par le système à chaque cycle.

Représentation : RTTI : Table registre vers entrée

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Source de commande
Source (partie haute)	3x, 4x	INT, UINT	La partie source (ou partie haute) contient l'adresse du registre source. Les données placées à l'adresse du registre source sont copiées à l'adresse cible, déterminée par le pointeur de décalage cible.
Pointeur (partie basse)	(1 à 254) (801 à 832)	INT, UINT	Le pointeur est un registre 3xxxx déterminé par une constante (00018 -> 30018), dont le contenu indique la destination. Une valeur comprise entre 1 et 254 indique un registre de sortie (40001 - 40254) tandis qu'une valeur comprise entre 801 et 832 indique un registre d'entrée (30001 - 30032). Si la valeur est en dehors de cette plage, l'opération n'est pas exécutée et le rail ERREUR est alimenté. Notez que la valeur du pointeur n'augmente pas automatiquement.
Sortie haute	0x	Aucun	Donne une image de la valeur de l'entrée haute
Sortie basse	0x	Aucun	Etat actif = erreur Valeur du pointeur hors limites

RTTO : Table registre vers sortie

150

Présentation

Introduction

Ce chapitre décrit l'instruction RTTO.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : RTTO : Table registre vers sortie	1006
Représentation : RTTO : Table registre vers sortie	1007

Description sommaire : RTTO : Table registre vers sortie

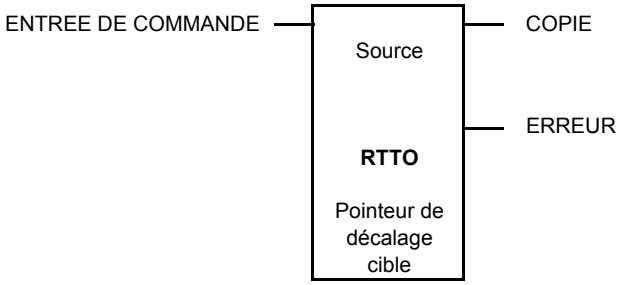
Description de la fonction

Le bloc Table registre vers sortie est l'une des quatre instructions de remplacement du 484. Il copie le contenu d'un registre d'entrée ou de sortie dans un autre registre d'entrée ou de sortie. Le registre de sortie déterminé par la constante en partie basse désigne ce registre cible. Seule une telle opération peut être prise en charge par le système à chaque cycle.

Représentation : RTTO : Table registre vers sortie

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Source de commande
Source (partie haute)	3x, 4x	INT, UINT	La partie source (ou partie haute) contient l'adresse du registre source. Les données placées à l'adresse du registre source sont copiées à l'adresse cible, déterminée par le pointeur de décalage cible.
Pointeur (partie basse)	(1 à 254) (801 à 824)	INT, UINT	Le pointeur est un registre 4xxxx déterminé par une constante (00018 -> 40018), dont le contenu indique la destination. Une valeur comprise entre 1 et 254 indique un registre de sortie (40001 - 40254) tandis qu'une valeur comprise entre 801 et 832 indique un registre d'entrée (30001 - 30032). Si la valeur est en dehors de cette plage, l'opération n'est pas exécutée et le rail ERREUR est alimenté. Notez que la valeur du pointeur n'augmente pas automatiquement.
Sortie haute	0x	Aucun	Donne une image de la valeur de l'entrée haute
Sortie basse	0x	Aucun	Etat actif = erreur Valeur du pointeur hors limites

RTU : Terminal déporté

151

Présentation

Introduction

Ce chapitre décrit l'instruction RTU.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : RTU : Terminal déporté	1010
Représentation : RTU : Terminal déporté	1011

Description sommaire : RTU : Terminal déporté

Description de la fonction

Le bloc MODBUS RTU (Remote Terminal Unit, terminal déporté) prend en charge les débits de données suivants :

- 1200
- 2400
- 4800
- 9600
- 19200

Représentation : RTU : Terminal déporté

Description des paramètres

Description des paramètres d'instruction

Registre	Fonction
4x	Numéro de révision du RTU (en lecture seule)
4x + 1	Champ d'état de défaut (en lecture seule)
4x + 2	Champ inutilisé
4x + 3	Définit le registre Débit des données Pour plus d'informations sur les entrées de registre correspondant aux débits, reportez-vous à la section ci-dessous : Entrées de registre correspondant aux débits.
4x + 4	Définit le registre Bits de données Pour plus d'informations sur les entrées de registre correspondant aux bits de données, reportez-vous à la section ci-dessous : Entrées de registre correspondant aux bits de données.
4x + 5	Registre de parité
4x + 6	Registre de bit d'arrêt
4x + 7	Champ inutilisé
4x + 8	Définit le registre Mot de commande Pour plus d'informations sur les entrées de registre correspondant aux mots de commande, reportez-vous à la section ci-dessous : Entrées de registre correspondant aux mots de commande

Entrées de registre correspondant aux débits

Le bloc MODBUS RTU (Remote Terminal Unit, terminal déporté) prend en charge les débits de données suivants :

- 1200
- 2400
- 4800
- 9600
- 19200

Les entrées de registre citées ci-dessous correspondent aux débits de données supportés. Pour configurer un débit, tapez le chiffre décimal approprié (par exemple 1200) dans le registre de débit de données.

Entrée de registre	Débit
1200	1200
2400	2400
4800	4800

Entrée de registre	Débit
9600	9600
19200	19200

Entrées de registre correspondant aux bits de données

Le bloc RTU prend en charge les bits de données 7 et 8. Vous trouverez ci-dessous les entrées de registre possibles pour le champ bits de données :

Entrée de registre	Champ Bit de données
7	7
8	8

Des messages Modbus peuvent être envoyés au format Modbus RTU ou Modbus ASCII.

- S'ils sont envoyés au format Modbus ASCII, tapez **7** dans ce champ.
- S'ils sont envoyés au format Modbus RTU, tapez **8**.

Si vous envoyez des messages ASCII, ce registre peut être réglé sur les bits de données 7 ou 8.

Entrées de registre correspondant aux mots de commande

Le bloc RTU interprète chaque bit du mot de commande comme fonction de mise en œuvre ou d'exécution. Vous trouverez ci-dessous la définition des bits des entrées de registre de mots de commande.

Entrée de registre	Définitions
1 (bit de poids fort)	Inutilisé
2	Validation du contrôle RTS/CTS
3	Inutilisé
4	Inutilisé
5	Inutilisé
6	Inutilisé
7	Validation des messages de type chaîne de caractères ASCII
8	Validation des messages Modbus
9	Inutilisé
10	Inutilisé
11	Inutilisé
12	Inutilisé
13	Inutilisé
14	Déconnexion du modem
15	Numérotation sur le modem
16 (bit de poids faible)	Initialisation du modem

Pour plus d'informations sur les bits 2, 7 et 8, reportez-vous à la section ci-dessous.

- Bit 2 : Validation du contrôle RTS/CTS (request-to-send/clear-to-send)
  Ce bit doit être activé lorsqu'un équipement DCE (Data Communication Equipment, matériel de transmission de données) connecté à l'automate nécessite un protocole matériel utilisant le contrôle RTS/CTS.
  Ce bit peut être utilisé conjointement avec les valeurs contenues dans le registre de retard de début de transmission (4xxxx ¸ 13) et dans le registre de retard de fin de transmission (4xxxx + 13). Le retard de début de transmission maintient le signal RTS activé pendant une durée pouvant varier entre 0 et 9999 ms avant que le bloc RTU n'envoie un message à partir du port de l'automate. Une fois que le bloc RTU a envoyé un message, le retard de fin de transmission maintient le signal RTS activé pendant une durée pouvant varier entre 0 et 9999 ms. Lorsque le retard de fin de transmission a expiré, le bloc RTU désactive le signal RTS.
- Bit 7 : Validation des messages de type chaîne de caractères ASCII
  Ce bit doit être activé pour envoyer des messages de type chaîne de caractères ASCII à partir du port de communication n°1 de l'automate.
  Le bloc RTU peut envoyer une chaîne ASCII dont la longueur maximum peut atteindre 512 caractères. Chaque message ASCII doit être programmé dans des registres 4x successifs de l'automate. Deux caractères sont autorisés par registre.
  Note : Cette chaîne de message ASCII ne doit pas être confondue avec un message Modbus envoyé au format ASCII.
- Bit 8 : Validation des messages Modbus
  Ce bit doit être activé pour envoyer des messages Modbus à partir du port de communication n°1 de l'automate.
  Les messages Modbus peuvent être envoyés au format RTU ou ASCII.
  - Si vous envoyez des messages Modbus au format RTU, réglez les bits de données du registre de bit de données (4xxxx + 4) sur 8.
  - Si vous envoyez un message Modbus au format ASCII, réglez les bits de données du registre de bit de données (4xxxx + 4) sur 7.

SAVE : Sauvegarde mémoire flash

152

Présentation

Introduction

Ce chapitre décrit l'instruction SAVE.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1016
Représentation : SAVE : Sauvegarde	1017
Description des paramètres	1018

Description sommaire

Description de la fonction

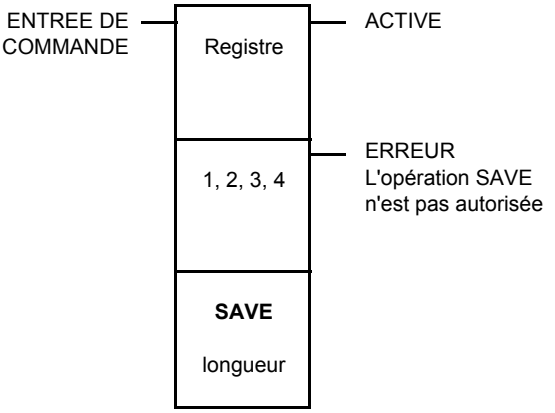
Note : Cette instruction est disponible avec la famille des automates TSX Compact, des processeurs Quantum 434 12/ 534 14 et des processeurs Momentum CCC 960 x0/ 980 x0.

L'instruction SAVE sauvegarde un bloc de registres 4x dans la mémoire d'état où ils sont protégés de toute modification non autorisée.

Représentation : SAVE : Sauvegarde

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Début de l'opération SAVE : elle doit rester activée jusqu'à ce qu'elle s'achève sans erreur ou jusqu'à ce qu'une erreur se produise.
Registre (partie haute)	4x	INT, UINT, WORD	Premier des 512 registres 4x successifs maximum à sauvegarder dans la mémoire d'état
1, 2, 3, 4 (voir *1, 2, 3, 4, p. 1018*) (partie médiane)		INT	Valeur entière qui définit le tampon spécifique où le bloc de données doit être sauvegardé
Longueur (partie basse)		INT	Nombre de mots à charger, compris entre 1 et 512
Sortie haute	0x	Aucun	Etat actif = SAVE est actif
Sortie médiane (voir *p. 1018*)	0x	Aucun	Etat actif = SAVE n'est pas autorisé

Description des paramètres

1, 2, 3, 4 (partie médiane)

La partie médiane définit le tampon spécifique, dans la mémoire d'état, où le bloc de données doit être sauvegardé. Quatre tampons de 512 mots sont autorisés. Chaque tampon se définit en plaçant la valeur correspondante en partie médiane ; autrement dit la valeur 1 représente le premier tampon, la valeur 2 le second tampon, etc. Les valeurs permises sont 1, 2, 3 et 4. Lorsque l'automate démarre, les quatre tampons sont à zéro. Par conséquent, vous ne pouvez pas sauvegarder les données dans le même tampon sans d'abord les charger avec l'instruction LOAD (voir *p. 697*). Si vous tentez de le faire, la sortie médiane passe à l'état actif. En d'autres termes, dès qu'un tampon est utilisé, il ne pourra l'être à nouveau qu'après suppression des données.

Sortie médiane

La sortie médiane passe à l'état actif lorsque les données précédemment sauvegardées n'ont pas été utilisées par l'instruction LOAD (voir *p. 697*). Vous évitez ainsi d'écraser malencontreusement les données du tampon SAVE.

SBIT : Mise à 1 de bit

153

Présentation

Introduction

Ce chapitre décrit l'instruction SBIT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1020
Représentation : SBIT : Mise à 1 de bit	1021

Description sommaire

Description de la fonction

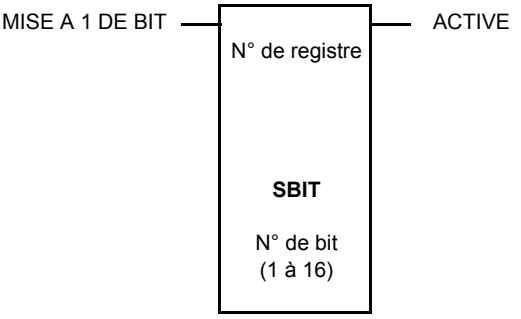
L'instruction Mise à 1 de bit (SBIT) vous permet de mettre un bit déterminé à l'état actif (1) en activant l'entrée supérieure.

Note : L'instruction SBIT ne suit pas les mêmes règles de placement de réseau que les bobines référencées 0x. Une instruction SBIT ne peut pas être placée en colonne 11 d'un réseau et peut être positionnée à gauche d'autres éléments logiques sur les mêmes échelons du schéma.

Représentation : SBIT : Mise à 1 de bit

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = met le bit indiqué à 1. Le bit reste à 1 lorsque l'entrée n'est plus activée.
N° de registre (partie haute)	4x	WORD	Registre de sortie dont la configuration binaire est contrôlée.
N° de bit (partie basse)		INT, UINT	Indique lequel des 16 bits est positionné à 1.
Sortie haute	0x	Aucun	Est activée lorsque le bit déterminé est mis à 1, et reste activée jusqu'à ce qu'il soit mis à zéro (au moyen de l'instruction RBIT (voir *p. 987*)).

SCIF : Interfaces de commande séquentielle

154

Présentation

Introduction

Ce chapitre décrit l'instruction SCIF.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1024
Représentation : SCIF : Interface de commande séquentielle	1025
Description des paramètres	1027

Description sommaire

Description de la fonction

L'instruction SCIF exécute soit une opération de séquenceur à tambour, soit une comparaison d'entrées (ICMP) avec les données définies dans la table de données d'étape.

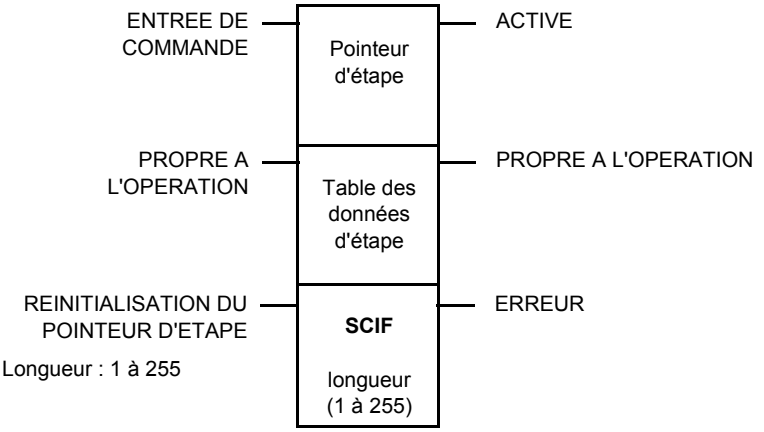
La sélection de l'opération est effectuée en définissant la valeur du premier registre de la table de données d'étape :

- **0 = mode tambour :**
  L'instruction commande les sorties d'une application de séquenceur à tambour.
- **1 = mode ICMP :**
  L'instruction vient lire les entrées pour s'assurer que les fins de course, les détecteurs de proximité, les boutons poussoirs, etc., sont correctement positionnés pour permettre aux sorties du tambour d'être déclenchées.

Représentation : SCIF : Interface de commande séquentielle

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = déclenche l'opération de commande séquentielle définie
Entrée médiane	0x, 1x	Aucun	**Mode tambour :** incrémente le pointeur d'étape à l'étape suivante **Mode ICMP :** l'état de la comparaison est transmis à la sortie médiane
Entrée basse	0x, 1x	Aucun	**Mode tambour :** Etat actif = remet le pointeur d'étape à 0 **Mode ICMP :** non utilisé
Pointeur d'étape (partie haute)	4x	INT, UINT	Numéro de l'étape en cours dans la table des données d'étape
Table des données d'étape (voir *p. 1027*) (partie médiane)	4x	INT, UINT	Premier registre de la table des données d'étape Pour plus d'informations, reportez-vous à la section *p. 1027*.
Longueur (voir *p. 1028*) (partie basse)		INT, UINT	Nombre de registres spécifiques à l'application utilisés dans la table des données d'étape
Sortie haute	0x	Aucun	Donne une image de l'état de l'entrée haute
Sortie médiane	0x	Aucun	Le mode tambour passe à l'état actif pour la dernière étape **Note :** Lors de l'utilisation de la sortie médiane, n'oubliez pas qu'en cas d'intégration avec une autre logique, si le pointeur d'étape est égal à zéro et si l'entrée médiane est à l'état actif, la sortie médiane est également à l'état actif. Dans cette condition, le pointeur d'étape est d'une étape hors de la séquence.
Sortie basse	0x	Aucun	Etat actif = erreur détectée

Description des paramètres

Table des données d'étape (partie médiane)

Le registre 4x mémorisé dans la partie médiane est le premier registre de la table de données d'étape. Les sept premiers registres de la table contiennent les données constantes et variables nécessaires à l'exécution de l'instruction :

Registre	Nom du registre	Description
Affiché	Type de sous–fonction	0 = mode tambour ; 1 = mode ICMP (la saisie d'une valeur différente dans ce registre entraînera la désactivation de toutes les sorties)
Premier implicite	Données de sortie masquée (en mode tambour)	Chargé par SCIF à chaque résolution du bloc ; le registre contient le registre des données actuelles de l'étape masqué par le registre de masque de sortie
	Données brutes d'entrée (en mode ICMP)	Chargé par l'utilisateur à partir d'un groupe d'entrées séquentielles devant être utilisées par le bloc de l'étape en cours
Deuxième implicite	Données actuelles de l'étape	Chargé par SCIF à chaque résolution du bloc ; le registre contient des données de l'étape en cours (indiquée par le pointeur d'étape).
Troisième implicite	Masque de sortie (en mode tambour)	Chargé par l'utilisateur avant d'utiliser le bloc, le contenu ne sera pas modifié lors de l'exécution du programme ; contient un masque à appliquer aux données à chaque étape du séquenceur.
	Masque d'entrée (en mode ICMP)	Chargé par l'utilisateur avant l'utilisation du bloc, il contient un masque à relier par un ET logique avec les données brutes d'entrées pour chaque étape ; les bits masqués ne sont pas comparés ; les données masquées sont placées dans le registre de données d'entrées masquées.
Quatrième implicite	Données d'entrée masquées (en mode ICMP)	Chargé par SCIF à chaque exécution du bloc, il contient le résultat du ET logique entre le masque d'entrée et les données brutes d'entrée.
	Non utilisé en mode tambour	

Registre	Nom du registre	Description
Cinquième implicite	Etat de comparaison (en mode ICMP)	Chargé par SCIF à chaque exécution du bloc, il contient le résultat du OU exclusif entre les données d'entrée masquées et les données d'étape courante ; les entrées non masquées n'étant pas dans l'état logique correct provoquent la mise à 1 du bit du registre associé ; les bits différents de zéro provoquent une comparaison non vérifiée et mettent à 1 la sortie médiane du bloc SCIF.
	Non utilisé en mode tambour	
Sixième implicite	Début de la table de données	Le premier de K registres de la table contenant les données de commande spécifiques à l'utilisateur. **Note :** Ce registre ainsi que les registres restants représentent les données d'étape spécifiques à l'application du procédé contrôlé.

Longueur de la table des données d'étape (partie basse)

Le nombre entier mémorisé dans la partie basse correspond à la longueur, c'est-à-dire au nombre de registres spécifiques à l'application utilisés dans la table de données d'étape. La longueur peut varier entre 1 et 255.

Le nombre total de registres nécessaires dans la table de données d'étape est égal à la longueur + 7. La longueur doit être ≥ à la valeur placée dans le registre des étapes utilisées dans la partie médiane.

SENS : Détection

155

Présentation

Introduction

Ce chapitre décrit l'instruction SENS.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1030
Représentation : SENS : Lecture de bit logique	1031
Description des paramètres	1033

Description sommaire

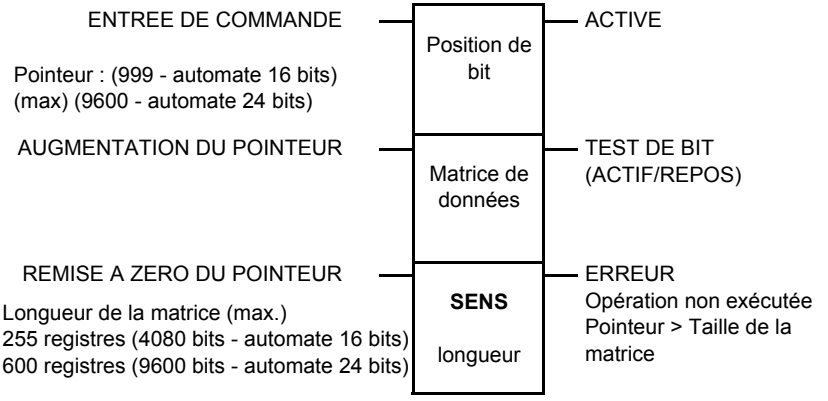
Description de la fonction

L'instruction SENS teste et communique l'état (1 ou 0) d'une position de bit spécifique dans la matrice de données. Une position de bit est détectée par cycle.

Représentation : SENS : Lecture de bit logique

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = lit la position du bit
Entrée médiane	0x, 1x	Aucun	Incrémente de 1 la position du bit au cycle suivant
Entrée basse	0x, 1x	Aucun	Remet la position du bit à 1
Position de bit (voir *p. 1033*) (partie haute)	3x, 4x	WORD	Position de bit spécifique à détecter dans la matrice de données, mémorisée explicitement comme entier ou enregistrée dans un registre, dans la plage : 1 à 9600 Pointeur : (999 - automate 16 bits) (max) (9900 - automate 24 bits)
Matrice de données (partie médiane)	0x, 4x	BOOL, WORD	Premier mot ou registre de la matrice de données.
Longueur (voir *, p. 1033*) (partie basse)		INT, UINT	Longueur max. de la matrice 255 registres (4080 bits - automate 16 bits) 600 registres (9600 bits - automate 24 bits)
Sortie haute	0x	Aucun	Donne une image de l'état de l'entrée haute
Sortie médiane	0x	Aucun	Etat actif = test de bit égal 1 Etat repos = test de bit égal 0
Sortie basse	0x	Aucun	Etat actif = erreur : position de bit > longueur de la matrice

Description des paramètres

Position de bit (partie haute)

Note : Si la position de bit est saisie comme un nombre entier ou dans un registre 3x, l'instruction ignorera l'état des entrées médiane et basse.

Longueur de la matrice (partie basse)

Le nombre entier saisi en partie basse indique la longueur de la matrice, c'est-à-dire le nombre de registres ou de mots 16 bits de la matrice de données. La longueur peut varier entre 1 et 600 dans une UC de 24 bits, autrement dit une longueur de matrice de 200 signifie 3200 positions de bits.

Liaisons

156

Présentation

Introduction

Ce chapitre décrit l'élément d'instruction Liaisons.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : Liaisons	1036
Représentation : Liaisons	1037

Description sommaire : Liaisons

Description de la fonction

Les liaisons sont de simples connexions par ligne droite entre contacts et/ou instructions d'un réseau de schéma à contacts. Les liaisons verticales (|) et horizontales (—) sont utilisées pour créer des connexions entre les lignes et les colonnes de logique. Pour annuler une liaison verticale, utilisez une ouverture verticale.

Représentation : Liaisons

Liaisons verticales	Elles relient verticalement des contacts ou des instructions dans une colonne de réseau, ou des entrées et sorties d'abonné pour créer des conditions OU. Lorsque deux contacts sont raccordés par des liaisons verticales, le courant passe si un ou les deux contacts sont activés.
Liaisons horizontales	Assurent des connexions horizontales le long d'un échelon dans un réseau de schémas à contacts.

SKP : Omission de réseaux

157

Présentation

Introduction

Ce chapitre décrit l'instruction SKP.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : SKP : Omission de réseaux	1040
Représentation : SKP : Omission de réseaux	1041

Description sommaire : SKP : Omission de réseaux

Description de la fonction

L'instruction SKP est une instruction de base présente dans tous les automates. Elle doit être utilisée avec précaution

L'instruction SKP permet de réduire la durée du cycle en laissant une partie de la logique non traitée. Le cycle logique omet alors les réseaux spécifiés dans le programme.

La fonction SKP permet de :
- ignorer des séquences de programmes rarement utilisées
- créer des sous-programmes

L'instruction SKP permet d'omettre un certain nombre de réseaux dans un programme schéma à contacts. Lorsqu'elle est activée, l'opération SKP est exécutée à chaque cycle. Le reste du réseau où l'instruction apparaît représente le premier du nombre spécifié de réseaux à ignorer. L'UC continue à ignorer des réseaux jusqu'à ce que le nombre total de réseaux ignorés soit égal au nombre spécifié dans le bloc d'instructions ou encore jusqu'à ce que la limite d'un segment soit atteinte. Une opération SKP ne peut pas franchir une limite de segment.

Une instruction SKP peut être activée seulement si vous avez précisé dans l'éditeur de la configuration automate que les sauts sont permis. SKP est une instruction nodale de niveau supérieur.

⚠ AVERTISSEMENT

Entrées et sorties omises

SKP est une instruction dangereuse qui doit être utilisée avec précaution. Si les entrées et sorties effectuant normalement des contrôles sont omises involontairement (ou non omises), le résultat peut engendrer des conditions dangereuses pour le personnel et les installations de l'application.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

⚠ ATTENTION

Lecture des valeurs pendant la modification

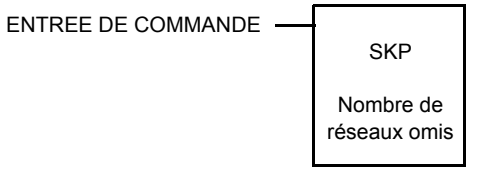
Utilisez les registres 3xxxx et 4xxxx avec précaution. Le processeur peut lire la valeur pendant la modification.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Représentation : SKP : Omission de réseaux

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	1x	Aucun	L'état actif déclenche une opération d'omission de réseau lorsque le courant est transmis. Une opération SKP est exécutée à chaque cycle tant que l'entrée est activée.
Nombre de réseaux omis (partie haute)	3x, 4x	INT, UINT WORD	La valeur saisie dans la partie détermine le nombre de réseaux à omettre. La valeur peut être • indiquée explicitement sous forme de valeur constante entière comprise entre 1 et 999 • mémorisée dans un registre d'entrée 3xxxx • mémorisée dans un registre de sortie 4xxxx La valeur de la partie inclut le réseau contenant l'instruction SKP. Les zones nodales du réseau où se trouve la SKP n'ayant pas encore été analysées seront omises ; ceci compte pour l'un des réseaux spécifiés comme devant être omis. Le processeur continue d'omettre des réseaux jusqu'à ce que le nombre total de réseaux ignorés soit égal à la valeur définie.

SRCH : Recherche

158

Présentation

Introduction

Ce chapitre décrit l'instruction SRCH.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1044
Représentation : SRCH : Recherche	1045
Description des paramètres	1047

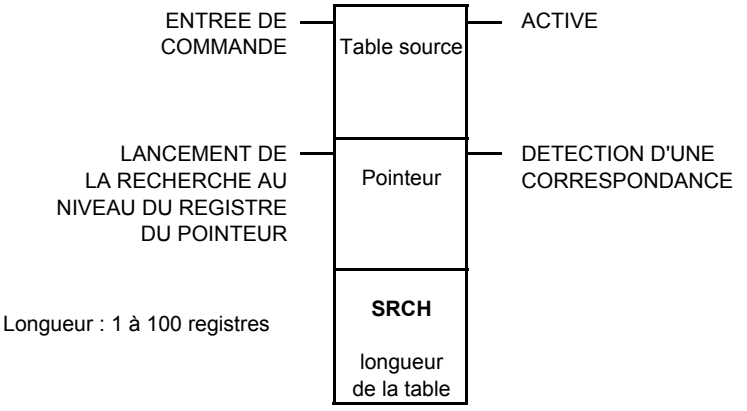
Description sommaire

Description de la fonction	L'instruction SRCH recherche une configuration binaire spécifique dans les registres d'une table source.

Représentation : SRCH : Recherche

Symbole

Représentation de l'instruction



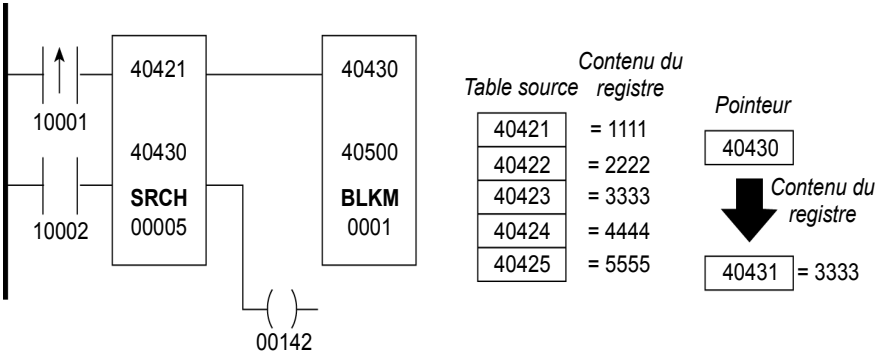
Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = déclenche la recherche
Entrée médiane	0x, 1x	Aucun	Etat repos = recherche depuis le début Etat actif = recherche depuis la dernière correspondance
Table source (partie haute)	3x, 4x	INT, UINT, WORD	Table source dans laquelle effectuer la recherche
Pointeur (voir *p. 1047*) (partie médiane)	4x	INT, UINT	Pointeur vers la table source
Longueur de la table (partie basse)		INT, UINT	Nombre de registres dans la table source, dans la plage : 1 à 100
Sortie haute	0x	Aucun	Donne une image de l'état de l'entrée haute
Sortie médiane	0x	Aucun	Etat actif = correspondance trouvée

Exemple SRCH

Dans l'exemple suivant, nous recherchons une *table source* contenant cinq registres (40421 à 40425) pour une configuration binaire spécifique. Le registre du pointeur (40430) indique que la configuration binaire souhaitée est mémorisée dans le registre 40431. Or, le registre contient une valeur de bit égale à 3333.



Dans chaque cycle où un contact F.M. 10001 passe de l'état repos à l'état actif, une configuration binaire équivalente à 3333 est recherchée dans la *table source*. Lorsque la correspondance est détectée, la sortie médiane transmet du courant vers la bobine 00142.

Si le contact N.O. 10002 est à l'état repos lorsque la correspondance est détectée dans le registre 40423, l'instruction SRCH active la bobine 00142 pour un cycle, puis relance la recherche dans le cycle suivant en haut de la *table source* (registre 40421). Si le contact 10002 est à l'état actif, l'instruction SRCH active la bobine 00142 pour un cycle, puis lance la recherche dans le registre 40424.

L'entrée haute étant un contact F.M., la valeur du pointeur est remise à zéro dans tous les cycles où l'entrée haute n'est pas alimentée. Nous utilisons une instruction BLKM pour régler la valeur du pointeur sur le registre 40500.

Description des paramètres

Pointeur (partie médiane)	Le registre 4x mémorisé en partie médiane est le pointeur vers la table source. Il pointe vers le registre source contenant la même valeur que celle mémorisée dans le registre suivant immédiatement le pointeur. Par exemple, si le registre pointeur est 400015, le registre 400016 contient une valeur pour laquelle l'instruction SRCH essaiera de trouver une correspondance dans la table source.

STAT : Etat

159

Présentation

Introduction

Ce chapitre décrit l'instruction STAT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1050
Représentation : STAT : Etat	1051
Description des paramètres	1052
Description de la table des états	1053
Mots 1 à 11 de l'état de l'automate pour Quantum et Momentum	1057
Mots 12 à 20 de l'état de fonctionnement des modules d'E/S pour Momentum	1062
Mots 12 à 171 de l'état de fonctionnement des modules d'E/S pour Quantum	1064
Mots 172 à 277 de l'état de communication pour Quantum	1066
Mots 1 à 11 de l'état de l'automate pour TSX Compact et Atrium	1071
Mots 12 à 15 de la santé des modules d'E/S pour TSX Compact	1074
Mots 182 à 184 de l'état du fonctionnement global et des nouvelles tentatives pour TSX Compact	1075

Description sommaire

Description de la fonction

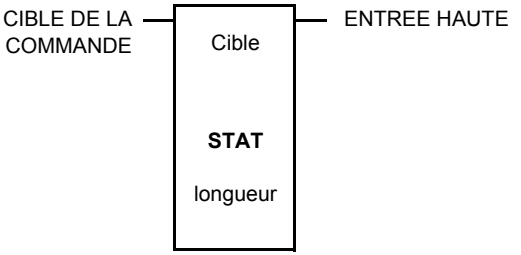
L'instruction STAT permet l'accès à un nombre de mots défini dans une table d'états (voir *p. 1053*) de la mémoire système de l'automate. Des données vitales de diagnostic concernant la santé de l'automate et de ses stations d'E/S déportées y sont placées.

Ces données comprennent :
- l'état de l'automate
- les états d'erreur possibles des modules d'E/S
- l'état de communication entrée–automate–sortie

Représentation : STAT : Etat

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = copie un nombre défini de mots depuis la table des états
Cible (voir *p. 1052*) (partie haute)	0x, 4x	INT, UINT, BOOL, WORD	Première position du bloc cible
Longueur (voir *p. 1052*) (partie basse)		INT, UINT	Nombre de registres ou de mots 16 bits du bloc cible Le nombre entier saisi en partie basse indique la longueur de la matrice, c'est-à-dire le nombre de registres ou de mots 16 bits de la matrice de données. La longueur peut varier entre 1 et 255 dans un UC 16 bits et entre 1 et 600 dans une UC 24 bits. Ainsi, une longueur de matrice de 200 signifie 3200 positions de bits. **Note :** Si des références 0xxxx sont utilisées comme cible, elles ne peuvent pas être programmées comme bobines, seulement comme contacts référençant ces numéros de bobines. Pour plus d'informations sur la longueur de table et les automates, reportez-vous à la section *p. 1052*.
Sortie haute	0x	Aucun	Etat actif = opération réussie

Description des paramètres

Mode de fonctionnement

L'instruction STAT vous permet de copier tout ou partie des mots d'état dans un bloc de registres ou un bloc de références TOR successives.

La copie vers le bloc STAT commence toujours par le premier mot de la table pour s'arrêter au dernier mot qui vous intéresse. Par exemple, si le tableau des états a une longueur de 277 mots et que vous vous intéressez uniquement à la statistique fournie par le mot 11, il vous suffit de copier les mots 1 à 11 en définissant une longueur de 11 dans l'instruction STAT.

Bloc cible (partie haute)

Le nombre saisi en partie haute est la première position du bloc cible, autrement dit le bloc dans lequel les mots actuellement intéressants de la table des états seront copiés.

Le nombre de registres de sortie ou de mots 16 bits du bloc cible est défini en partie basse (longueur).

Note : Nous vous recommandons de ne pas utiliser de données TOR dans la partie cible STAT en raison du trop grand nombre qui serait nécessaire pour contenir les informations d'état.

Longueur (partie basse)

Le nombre entier saisi en partie basse indique le nombre de registres ou de mots 16 bits du bloc cible dans lequel les informations d'état actuelles seront écrites.

La longueur maximum permise diffère en fonction du type d'automate utilisé et du type de protocole de communication d'E/S employé.

- Pour un automate à UC de type modulaire 984A, 984B ou 984X utilisant le protocole RIO *S901*, la plage disponible de la table d'état système varie entre 1 et 75 mots.
- Pour les automates à UC 16 bits utilisant le protocole RIO *S908* (par exemple les automates à UC de type carte 38x, 48x et 68x), la plage disponible de la table d'état système varie entre 1 et 255
- Pour les automates à UC 24 bits utilisant le protocole RIO *S908* (par exemple les automates à UC de type carte 38x, 48x et 68x), la plage disponible de la table d'état système varie entre 1 et 277
- Pour les automates Compaq-984, la plage disponible de la table d'état système varie entre 1 et 184
- Pour les automates Modicon Micro, la plage disponible de la table d'état système varie entre 1 et 56

Description de la table des états

Généralités

L'instruction STAT permet d'afficher l'état de l'automate et le système des E/S pour Quantum (voir *p. 1053*), Atrium (voir *p. 1056*), TSX Compact (voir *p. 1056*) et Momentum (voir *p. 1055*).

Les 11 premiers mots d'état sont utilisés pareillement par Quantum et Momentum et pareillement par TSX Compact et Atrium. Les mots suivants ont une signification différente pour Quantum, TSX Compact et Momentum.

Vue d'ensemble Quantum

Les 277 mots de la table des états s'organisent en trois sections :

- Etat de l'automate (mots 1 à 11) (voir *p. 1057*)
- Fonctionnement des modules d'E/S (mots 12 à 171) (voir *p. 1064*)
- Fonctionnement des communications d'E/S (mots 172 à 277) (voir *p. 1066*)

Mots de la table des états :

Mot décimal	Contenu du mot	Mot hex
1	Etat de l'automate	01
2	Etat de la redondance d'UC	02
3	Etat de l'automate	03
4	Etat du RIO	04
5	Etat d'arrêt de l'automate	06
6	Nombre de segments en schéma à contacts	06
7	Pointeur (EOL) de fin de logique	07
8	Redondance et timeout du RIO	08
9	Etat des messages ASCII	09
10	Etat Marche/chargement/optimisé	0A
11	Inutilisé	0B
12	Station 1, châssis 1	0C
13	Station 1, châssis 2	0D
. . .	. . . . .	. . .
16	Station 1, châssis 5	0F
17	Station 2, châssis 1	10
18	Station 2, châssis 2	11
. . .	. . . . .	. . .
171	Station 32, châssis 5	AB
172	Code d'erreur de démarrage S908	AC
173	Erreurs du câble A	AD

Mot décimal	Contenu du mot	Mot hex
174	Erreurs du câble A	AE
175	Erreurs du câble A	AF
176	Erreurs du câble B	B0
178	Erreurs du câble B	B1
178	Erreurs du câble B	B2
179	Erreurs de communication globale	B3
180	Erreurs de communication globale	B4
181	Erreurs de communication globale	B5
182	Erreurs, état de fonctionnement et compteurs des nouvelles tentatives de la station 1 des automates TSX Compact 984 (premier mot)	B6
183	Erreurs, état de fonctionnement et compteurs des nouvelles tentatives de la station 1 des automates TSX Compact 984 (second mot)	B7
184	Erreurs, état de fonctionnement et compteurs des nouvelles tentatives de la station 1 des automates TSX Compact 984 (troisième mot)	B8
185	Erreurs, état de fonctionnement et compteurs des nouvelles tentatives de la station 2 des automates TSX Compact 984 (premier mot)	B9
. . .	. . . . .	. . .
275	Erreurs, état de fonctionnement et compteurs des nouvelles tentatives de la station 32 des automates TSX Compact 984 (premier mot)	113
276	Erreurs, état de fonctionnement et compteurs des nouvelles tentatives de la station 32 des automates TSX Compact 984 (second mot)	114
277	Erreurs, état de fonctionnement et compteurs des nouvelles tentatives de la station 32 des automates TSX Compact 984 (troisième mot)	115

Présentation de Momentum

Les 20 mots de la table des états s'organisent en deux sections :

- Etat de l'automate (mots 1 à 11) (voir *p. 1057*)
- Fonctionnement des modules d'E/S (mots 12 à 20) (voir *p. 1062*)

Mots de la table des états :

Mot décimal	Contenu du mot	Mot hex
1	Etat de l'automate	01
2	Etat de la redondance d'UC	02
3	Etat de l'automate	03
4	Etat du RIO	04
5	Etat d'arrêt de l'automate	06
6	Nombre de segments en schéma à contacts	06
7	Pointeur (EOL) de fin de logique	07
8	Redondance et timeout du RIO	08
9	Etat des messages ASCII	09
10	Etat Marche/chargement/optimisé	0A
11	Inutilisé	0B
12	Santé du module d'E/S Momentum local	0C
13	Santé du module de bus d'E/S	0D
14	Santé du module de bus d'E/S	0E
15	Santé du module de bus d'E/S	0F
16	Santé du module de bus d'E/S	10
17	Santé du module de bus d'E/S	11
18	Santé du module de bus d'E/S	12
19	Santé du module de bus d'E/S	13
20	Santé du module de bus d'E/S	14

Présentation de TSX Compact et d'Atrium

Les 184 mots de la table des états s'organisent en trois sections :

- Etat de l'automate (mots 1 à 11) (voir *p. 1071*)
- Fonctionnement des modules d'E/S (mots 12 à 15) (voir *p. 1074*)
- Inutilisé (de 16 à 181)
- Etat de fonctionnement global et des nouvelles tentatives de communication (mots 182 à 184) (voir *p. 1075*)

Mots de la table des états :

Mot décimal	Contenu du mot	Mot hex
1	Etat de l'automate	01
2	Inutilisé	02
3	Etat de l'automate	03
4	Inutilisé	04
5	Etat d'arrêt de l'automate	06
6	Nombre de segments en schéma à contacts	06
7	Pointeur (EOL) de fin de logique	07
8	Inutilisé	08
9	Inutilisé	09
10	Etat Marche/chargement/optimisé	0A
11	Inutilisé	0B
12	Etat de fonctionnement des E/S Châssis 1	0C
13	Etat de fonctionnement des E/S Châssis 2	0D
14	Etat de fonctionnement des E/S Châssis 3	0E
15	Etat de fonctionnement des E/S Châssis 4	0F
16 à 181	Inutilisé	10 à B5
182	Etat de fonctionnement	B6
183	Compteur des erreurs d'E/S	B7
184	Compteur des nouvelles tentatives du bus PAB	B8

Mots 1 à 11 de l'état de l'automate pour Quantum et Momentum

Etat de l'automate (mot 1)

Le mot 1 affiche les aspects suivants de l'état de l'automate :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 - 5	Inutilisé
6	1 = autorisation cycle constant
7	1 = autorisation retard cycle unique
8	1 = logique utilisateur 16 bits 0 = logique utilisateur 24 bits
9	1 = alimentation CA activée
10	1 = voyant RUN éteint
11	1 = protection de la mémoire désactivée
12	1 = pile défectueuse
13 - 16	Inutilisé

Etat de la redondance d'UC (mot 2)

Le mot 2 affiche l'état de la redondance d'UC pour les automates 984 qui utilisent des modules d'UC redondante S911/R911 :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = S911/R911 présent et OK
2 - 10	Inutilisé
11	0 = automate basculé sur A 1 = automate basculé sur B
12	0 = les automates ont une logique cohérente 1 = les automates n'ont pas de logique cohérente
13, 14	Etat du système distant : 0 1 = déconnecté (1 dec) 1 0 = primaire (2 dec) 1 1 = redondant (3 dec)
15, 16	Etat du système local : 0 1 = déconnecté (1 dec) 1 0 = primaire (2 dec) 1 1 = redondant (3 dec)

Etat de l'automate (mot 3)

Le mot 3 affiche plus d'aspects de l'état de l'automate :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = premier cycle
2	1 = commande de démarrage en attente
3	1 = délai du cycle constant dépassé
4	1 = Configuration initiale existante
5 - 12	Inutilisé
13 - 16	Cycles uniques

Etat du RIO (mot 4)

Le mot 4 est utilisé pour des informations sur le processeur d'E/S :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = processeur d'E/S défaillant
2	1 = timeout du processeur d'E/S
3	1 = rebouclage du processeur d'E/S
4	1 = défaillance mémoire du processeur d'E/S
5 - 12	Inutilisé
13 - 16	Les E/S = 00 n'ont pas répondu 01 = pas de réponse reçue 02 = échec du rebouclage

Etat d'arrêt de l'automate (mot 5)

Le mot 5 affiche les conditions d'état d'arrêt de l'automate :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = arrêt du port de périphérique

Bit	Fonction
2	Erreur de parité de la mémoire étendue (pour les automates à UC de type modulaire) ou erreur d'affectation des E/S/S908 (pour les autres automates) Si le bit = 1 dans un **automate 984B**, une erreur a été détectée dans la mémoire étendue ; l'automate s'exécutera, mais l'erreur sera envoyée pour les fonctions XMRD/XMWT. Si le bit = 1 pour **tout autre automate que à UC de type modulaire**, soit une erreur d'affectation des E/S a été détectée, soit le S908 manque dans une configuration multi–stations.
3	1 = automate en Configuration initiale
4	1 = intervention du périphérique non valide
5	1 = ordonnanceur de segments invalide
6	1 = le lancement de l'abonné n'a pas démarré le segment
7	1 = échec du test de la mémoire d'état
8	1 = affectation des E/S invalide
9	1 = temporisation chien de garde expirée
10	1 = erreur d'horloge en temps réel
11	Le traitement de la logique de l'UC a échoué (pour les automates à UC de type modulaire) ou le tableau d'utilisation des bobines (pour les autres automates). Si le bit = 1 dans un automate à UC de type modulaire, les diagnostics internes ont détecté une défaillance de l'UC. Si le bit = 1 dans tout autre automate qu'à UC modulaire, le tableau d'utilisation des bobines ne correspond pas aux bobines de la logique utilisateur.
12	1 = défaillance du processeur d'E/S
13	1 = abonné invalide
14	1 = total de contrôle logique
15	1 = bobine désactivée en mode MARCHE (voir le **ATTENTION** ci-dessous)
16	1 = mauvaise config.

⚠ ATTENTION

Utilisation d'un automate Quantum ou 984-684E/785E

Si vous utilisez un automate Quantum ou 984–684E/785E, le bit 15 du mot 5 n'est jamais mis à 1. Ces automates peuvent démarrer et marcher en ayant leurs bobines désactivées en mode MARCHE (optimisé). De même, tous les bits du mot 5 doivent être mis à 0 lorsque l'un de ces automates est en marche.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Etat d'arrêt de l'automate (mot 6)

Le mot 6 affiche le nombre de segments en schéma à contacts ; un nombre binaire est affiché :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 - 16	Nombre de segments (exprimé en nombre décimal)

Etat d'arrêt de l'automate (mot 7)

Le mot 7 affiche l'adresse du pointeur de fin de logique (EOL) :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 - 16	Adresse du pointeur de fin de logique

Redondance et Timeout du RIO (mot 8)

Le mot 8 utilise ses quatre bits de poids faible pour afficher la constante du timeout des E/S décentralisées :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 - 12	Inutilisé
13 - 16	Constante timeout RIO

Etat des messages ASCII (mot 9)

Le mot 9 utilise ses quatre bits de poids faible pour afficher l'état des messages ASCII :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 12	Inutilisé
13	1 = incohérence entre le nombre de messages et les pointeurs
14	1 = pointeur de message invalide
15	1 = message invalide
16	1 = erreur de total de contrôle des messages

Etat Marche/ chargement/ optimisé (mot 10)

Le mot 10 utilise ses deux bits de poids faible pour afficher l'état Marche/ chargement/optimisé :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 14	Inutilisé
15, 15	0 0 = Optimisé (0 dec) 0 1 = Marche (1 dec) 1 0 = Chargement (2 dec)

Mot 11

Ce mot n'est pas utilisé.

Mots 12 à 20 de l'état de fonctionnement des modules d'E/S pour Momentum

Etat de fonctionnement des modules d'E/S

Les mots d'état de 12 à 20 affichent l'état de fonctionnement des modules d'E/S.

Un mot est réservé pour chacune des stations locales, huit mots sont utilisés pour représenter l'état de fonctionnement de 128 modules de bus d'E/S maximum

Etat de fonctionnement du module d'E/S Momentum local

Le mot 12 affiche l'état de fonctionnement du module d'E/S local Momentum :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Module local
2 - 16	Inutilisé

Etat de fonctionnement du module de bus d'E/S Momentum

Les mots 13 à 20 affichent l'état de fonctionnement des modules de bus d'E/S Momentum, comme suit :

Mot	Modules de bus d'E/S
13	1 à 16
14	17 à 32
15	33 à 48
16	49 à 64
17	65 à 80
18	81 à 96
19	97 à 112
20	113 à 128

Chaque mot affiche l'état de fonctionnement des modules de bus d'E/S Momentum, comme suit :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	**Fonction**
1	1 = Module 1
2	1 = Module 2
3	1 = Module 3
4	1 = Module 4
5	1 = Module 5
6	1 = Module 6
7	1 = Module 7
8	1 = Module 8
9	1 = Module 9
10	1 = Module 10
11	1 = Module 11
12	1 = Module 12
13	1 = Module 13
14	1 = Module 14
15	1 = Module 15
16	1 = Module 16

Mots 12 à 171 de l'état de fonctionnement des modules d'E/S pour Quantum

Mots d'état du RIO

Les mots d'état de 12 à 20 affichent l'état de fonctionnement des modules d'E/S.

Cinq mots sont réservés pour chacune des 32 stations maximum, un seul mot pour chacun des cinq châssis possibles (logements d'E/S) dans chaque station. Chaque châssis peut contenir jusqu'à 11 modules d'E/S ; les bits de 1 à 11 de chaque mot représentent l'état de fonctionnement du module d'E/S associé de chaque châssis.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Emplacement 1
2	1 = Emplacement 2
3	1 = Emplacement 3
4	1 = Emplacement 4
5	1 = Emplacement 5
6	1 = Emplacement 6
7	1 = Emplacement 7
8	1 = Emplacement 8
9	1 = Emplacement 9
10	1 = Emplacement 10
11	1 = Emplacement 11
12	1 = Emplacement 12
13	1 = Emplacement 13
14	1 = Emplacement 14
15	1 = Emplacement 15
16	1 = Emplacement 16

Quatre conditions doivent être respectées avant qu'un module d'E/S puisse indiquer un bon fonctionnement :

- L'emplacement doit être affecté en E/S
- L'emplacement doit contenir un module de type correct
- Les communications valides doivent exister entre le module et l'interface du RIO pour les stations distantes
- Les communications valides doivent exister entre l'interface du RIO de chaque station distante et le processeur d'E/S de l'automate

Mots d'état pour les commandes d'opérateur MMI

L'état des 32 commandes des boutons Eléments et des unités PanelMate sur un réseau RIO peut également être géré avec un mot d'état de fonctionnement des E/S. Les commandes de bouton occupent l'emplacement 4 dans un châssis d'E/S et peuvent être gérés au bit 4 du mot d'état approprié. Un PanelMate sur réseau RIO occupe l'emplacement 1 du châssis 1 de la station et peut être géré au bit 1 du premier mot d'état pour la station.

Note : L'état de communication du clavier ASCII peut être géré avec les codes d'erreur dans les blocs ASCII de lecture/écriture (READ/WRIT).

Mots 172 à 277 de l'état de communication pour Quantum

Etat du DIO

Les mots d'état 172 à 277 contiennent l'état de communication du système d'E/S. Les mots 172 à 181 sont des mots d'état globaux. Parmi les 96 mots restants, trois mots sont dédiés à chacune des 32 stations maximum, en fonction du type de l'automate.

Le mot 172 stocke le code d'erreur de démarrage Quantum. Ce mot est toujours 0 lorsque le système est en marche. En cas d'erreur, l'automate ne démarre pas, mais génère un code d'état d'arrêt de 10 (mot 5 (voir *p. 1059*)).

Codes d'erreur de démarrage Quantum

Code	Erreur	Signification (endroit où l'erreur s'est produite)
01	BADTCLEN	Longueur d'affectation des E/S
02	BADLNKNUM	Numéro de lien d'E/S décentralisée
03	BADNUMDPS	Nombre de stations dans l'affectation des E/S
04	BADTCSUM	Total de contrôle d'affectation des E/S
10	BADDDLEN	Longueur du descripteur de station
11	BADDRPNUM	Numéro de station d'E/S
12	BADHUPTIM	Temps d'autonomie de la station
13	BADASCNUM	Numéro de port ASCII
14	BADNUMODS	Nombre de modules de la station
15	PRECONDRP	Station déjà configurée
16	PRECONPRT	Port déjà configuré
17	TOOMNYOUT	Plus de 1024 points de sortie
18	TOOMNYINS	Plus de 1024 points d'entrée
20	BADSLTNUM	Adresse d'emplacement de module
21	BADRCKNUM	Adresse du châssis de module
22	BADOUTBC	Nombre d'octets de sortie
23	BADINBC	Nombre d'octets d'entrée
25	BADRF1MAP	Premier numéro de référence
26	BADRF2MAP	Second numéro de référence
27	NOBYTES	Pas d'octets en entrée ou sortie
28	BADDISMAP	Bits internes pas sur limite des 16 bits
30	BADODDOUT	Module de sortie impair dépareillé
31	BADODDIN	Module d'entrée impair dépareillé
32	BADODDREF	Référence de module impair dépareillé
33	BAD3X1XRF	Référence 1x après le registre 3x

Code	Erreur	Signification (endroit où l'erreur s'est produite)
34	BADDMYMOD	Référence du module factice déjà utilisé
35	NOT3XDMY	Le module 3x n'est pas factice
36	NOT4XDMY	Le module 4x n'est pas factice
40	DMYREAL1X	Module 1x factice puis réel
41	REALDMY1X	Module 1x réel puis factice
42	DMYREAL3X	Module 3x factice puis réel
43	REALDMY3X	Module 3x réel puis factice

Etat du câble A

Les mots 173 à 175 sont des mots d'erreur du câble A :

Mot 173

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Compte les erreurs de trame
9 à 16	Compte les dépassements du récepteur DMA

Mot 174

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Compte les erreurs de réception
9 à 16	Compte les réceptions de station incorrectes

Mot 175

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Trame trop courte
2	1 = Pas de fin de trame
3 à 12	Inutilisé
13	1 = Erreur CRC
14	1 = Erreur d'alignement
15	1 = Erreur de dépassement
16	Inutilisé

Etat du câble B

Les mots 176 à 178 sont des mots d'erreur du câble A :

Mot 176

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Compte les erreurs de trame
9 à 16	Compte les dépassements du récepteur DMA

Mot 177

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Compte les erreurs de réception
9 à 16	Compte les réceptions de station incorrectes

Mot 178

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Trame trop courte
2	1 = Pas de fin de trame
3 à 12	Inutilisé
13	1 = Erreur CRC
14	1 = Erreur d'alignement
15	1 = Erreur de dépassement
16	Inutilisé

Etat de communication globale (mots 179 à 181)

Le **mot 179** affiche l'état de communication globale :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Santé comm
2	1 = Etat du câble A
3	1 = Etat du câble B
4	Inutilisé
5 à 8	Compteur des communications perdues
9 à 16	Compteur totalisateur des nouvelles tentatives

Le **mot 180** est le compteur totalisateur global des erreurs pour le câble A :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Compte les erreurs détectées
9 à 162	Ne compte aucune réponse

Le **mot 181** est le compteur totalisateur global des erreurs pour le câble B :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Compte les erreurs détectées
9 à 162	Ne compte aucune réponse

Etat des E/S décentralisées (mots 182 à 277)

Les mots 182 à 277 permettent de décrire l'état de la station des E/S décentralisées ; trois mots d'état sont utilisés pour chaque station.

Le premier mot de chaque groupe de trois affiche l'état de communication pour la station appropriée :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Santé de la communication
2	1 = Etat du câble A
3	1 = Etat du câble B

Bit	Fonction
4	Inutilisé
5 à 8	Compteur des communications perdues
9 à 16	Compteur totalisateur des nouvelles tentatives

Le second mot de chaque groupe de trois est le compteur totalisateur des erreurs sur le câble A pour la station appropriée :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Au moins une erreur dans les mots 173 à 175
9 à 162	Ne compte aucune réponse

Le troisième mot de chaque groupe de trois est le compteur totalisateur des erreurs sur le câble B pour la station appropriée :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 8	Au moins une erreur dans les mots 176 à 178
9 à 162	Ne compte aucune réponse

Note : Pour les automates où la station 1 est réservée pour les E/S locales, les mots d'état de 182 à 184 sont utilisés comme suit :

Le **mot 182** affiche l'état de la station locale :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Tous les modules sont OK
2 à 8	Toujours 0
9 à 162	Nombre de fois où un module a été vu comme étant défectueux ; le compteur reboucle à 255

Le **mot 183** est un compteur de défaut 16 bits qui indique le nombre de fois où un module a été jugé défectueux lors de l'accès à ce module. Reboucle à 65535.

Le **mot 184** est un compteur de défaut 16 bits qui indique le nombre d'erreurs de communication survenues lors d'un accès à un module d'E/S. Reboucle à 65535.

Mots 1 à 11 de l'état de l'automate pour TSX Compact et Atrium

Etat de l'UC (mot 1)

Le mot 1 affiche les aspects suivants de l'état de l'automate :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 5	Inutilisé
6	1 = autorisation cycle constant
7	1 = autorisation retard cycle unique
8	1 = logique utilisateur 16 bits 0 = logique utilisateur 24 bits
9	1 = alimentation CA activée
10	1 = voyant RUN éteint
11	1 = protection de la mémoire désactivée
12	1 = pile défectueuse
13 à 16	Inutilisé

Mot 2

Ce mot n'est pas utilisé.

Etat de l'automate (mot 3)

Le mot 3 affiche les aspects de l'état de l'automate :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = premier cycle
2	1 = commande de démarrage en attente
3	1 = le temps de cycle a dépassé la cible du cycle constant
4	1 = configuration initiale existante
5 à 12	Inutilisé
13 à 16	Cycles uniques

Mot 4

Ce mot n'est pas utilisé.

Etat d'arrêt de l'UC (mot 5)

Le mot 5 affiche les conditions d'état d'arrêt de l'automate :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = arrêt du port de périphérique
2	1 = erreur de parité XMEM
3	1 = configuration initiale
4	1 = intervention du périphérique non valide
5	1 = ordonnanceur de segments invalide
6	1 = pas de départ de réseau (SON) au démarrage d'un segment
7	1 = échec du test de la mémoire d'état
8	1 = pas de fin de logique (EOL), (affectation des E/S incorrecte)
9	1 = la temporisation chien de garde a expiré
10	1 = erreur d'horloge en temps réel
11	1 = défaillance de l'UC
12	Inutilisé
13	1 = abonné invalide dans la logique du schéma à contacts
14	1 = erreur de total de contrôle logique
1	1 =bobine désactivée en mode MARCHE
16	1 = configuration automate défectueuse

Nombre de segments du programme (mot 6)

Le mot 6 affiche le nombre de segments en schéma à contacts ; un nombre binaire est affiché. Pendant la mise sous tension, ce mot est confirmé être le nombre d'éléments EOS (DOIO) plus 1 (pour les éléments de fin de logique) ; s'il est faux, un code d'arrêt est mis à 1, ce qui a pour effet d'éteindre le voyant run :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 16	Nombre de segments du programme en schéma à contacts en cours d'exécution (exprimé par un nombre décimal)

Adresse du pointeur de fin de logique (mot 7)

Le mot 7 affiche l'adresse du pointeur de fin de logique (EOL) :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 16	Adresse du pointeur de fin de logique

Mot 8, Mot 9

Ces mots ne sont pas utilisés.

Etat Marche/ chargement/ optimisé (mot 10)

Le mot 10 utilise ses deux bits de poids faible pour afficher l'état Marche/ chargement/optimisé :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 14	Inutilisé
15, 16	0 0 = Optimisé (0 dec) 0 1 = Marche (1 dec) 1 0 = Chargement (2 dec)

Mot 11

Ce mot n'est pas utilisé.

Mots 12 à 15 de la santé des modules d'E/S pour TSX Compact

Santé des modules d'E/S TSX Compact

Les mots 12 à 15 permettent d'afficher le fonctionnement des modules d'E/S A120 des quatre châssis :

Mot	Numéro de châssis
12	1
13	2
14	3
15	4

Chaque mot contient l'état de fonctionnement de cinq modules d'E/S A120 au maximum. Le bit de poids fort (le plus à gauche) représente le fonctionnement du module à l'emplacement 1 du châssis :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Emplacement 1
2	1 = Emplacement 2
3	1 = Emplacement 3
4	1 = Emplacement 4
5	1 = Emplacement 5
6 à 16	Inutilisé

Si un module est affecté en E/S et s'il est actif, le bit aura une valeur "1". Si un module est inactif ou n'est pas affecté en E/S, le bit aura une valeur "0".

Note : Les emplacements 1 et 2 du châssis 1 (mot 12) ne sont pas utilisés parce que l'automate lui–même utilise ces deux emplacements.

Mots 182 à 184 de l'état du fonctionnement global et des nouvelles tentatives pour TSX Compact

Introduction

Il existe trois mots contenant des informations sur le fonctionnement et la communication dans les modules d'E/S installés. S'ils sont gérés par le bloc Stat, on les trouve dans les mots 182 à 184. Cela nécessite une longueur de bloc Stat d'au moins 184 mots (les mots 16 à 181 ne sont pas utilisés).

Mots 16 à 181

Ces mots ne sont pas utilisés.

Etat de fonctionnement (mot 182)

Le mot 182 incrémente à chaque défectuosité d'un module. Après une défectuosité du module, ce compteur ne réincrémentera que lorsque ce module redeviendra OK, puis sera de nouveau défectueux.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = Tous les modules sont OK
2 à 9	Inutilisé
10 à 16	Compteur "Module devenu défectueux"

Compteur des erreurs d'E/S (mot 183)

Ce compteur est le même que celui cité plus haut, sauf que ce mot incrémente à chaque cycle où un module reste défectueux.

Compteur des nouvelles tentatives du bus PAB (mot 184)

Les diagnostics sont effectués sur les communications passant par ce bus. Ce mot doit normalement n'être formé que de zéros. Si au bout de 5 tentatives, une erreur de bus est toujours détectée, l'automate s'arrêtera et un code d'erreur 10 sera affiché. Une erreur peut se produire en cas de court–circuit de l'embase ou à cause d'un bruit. Le compteur reboucle en cours d'exécution. S'il y a moins de 5 tentatives, aucune erreur de bus n'est détectée.

SU16 : Soustraction de valeurs 16 bits

160

Présentation

Introduction

Ce chapitre décrit l'instruction SU16.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1078
Représentation : SU16 : Soustraction de valeurs 16 bits	1079

Description sommaire

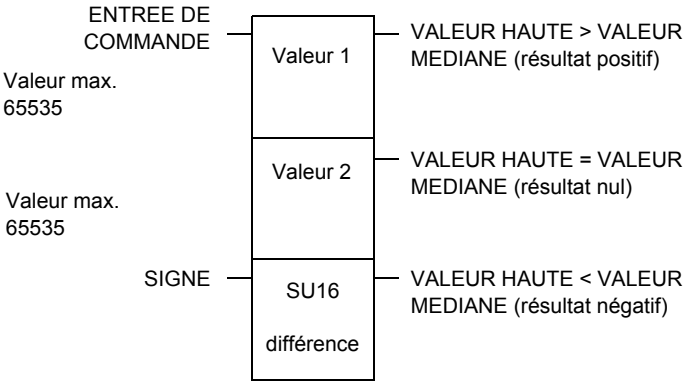
Description de la fonction

L'instruction SU16 effectue une soustraction signée ou non signée 16 bits (valeur 1 - valeur 2) entre les valeurs haute et médiane, et place la différence signée ou non dans un registre de sortie 4x de la partie basse.

Représentation : SU16 : Soustraction de valeurs 16 bits

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = déclenche l'opération valeur 1 - valeur 2.
Entrée basse	0x, 1x	Aucun	Etat actif = opération signée. Etat repos = opération non signée.
Valeur 1 (partie haute)	3x, 4x	INT, UINT	Diminuende, peut être affiché explicitement sous forme d'entier (compris entre 1 et 65.535) ou mémorisé dans un registre.
Valeur 2 (partie médiane)	3x, 4x	INT, UINT	Diminuteur, peut être affiché explicitement sous forme d'entier (compris entre 1 et 65.535) ou mémorisé dans un registre.
Différence (partie basse)	4x	INT, UINT	Différence
Sortie haute	0x	Aucun	Etat actif = valeur 1 > valeur 2.
Sortie médiane	0x	Aucun	Etat actif = valeur 1 = valeur 2.
Sortie basse	0x	Aucun	Etat actif = valeur 1 < valeur 2.

SUB : Soustraction

161

Présentation

Introduction

Ce chapitre décrit l'instruction SUB.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1082
Représentation : SUB : Soustraction	1083

Description sommaire

Description de la fonction

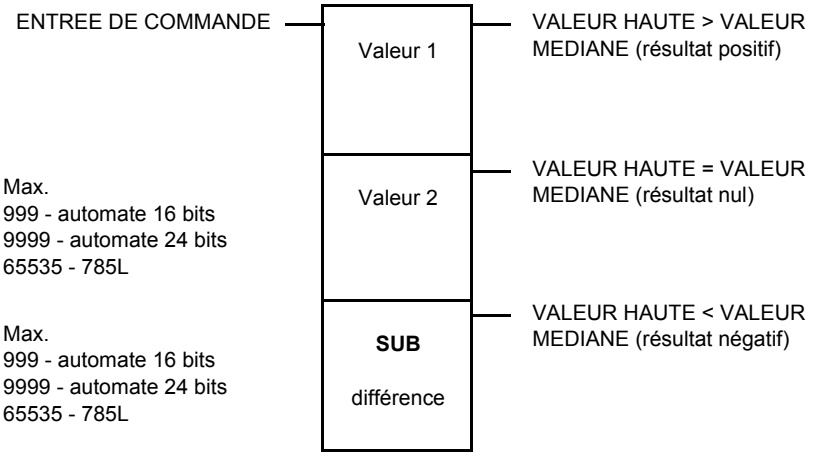
L'instruction SUB effectue une soustraction signée ou non signée 16 bits (valeur 1 - valeur 2) entre les valeurs haute et médiane, et place la différence signée ou non dans un registre de sortie 4x de la partie basse.

Note : SUB est souvent utilisée en tant que comparateur, l'état des sorties indiquant si la valeur 1 est supérieure, égale ou inférieure à la valeur 2.

Représentation : SUB : Soustraction

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = déclenche l'opération valeur 1 - valeur 2.
Valeur 1 (partie haute)	3x, 4x	INT, UINT	Diminuende, peut être affiché explicitement sous forme d'entier ou mémorisé dans un registre. Max. 255 - automate 16 bits Max. 999 - automate 24 bits Max. 65535-785L
Valeur 2 (partie médiane)	3x, 4x	INT, UINT	Diminuteur, peut être affiché explicitement sous forme d'entier ou mémorisé dans un registre. Max. 255 - automate 16 bits Max. 999 - automate 24 bits Max. 65535-785L
Différence (partie basse)	4x	INT, UINT	Différence
Sortie haute	0x	Aucun	Etat actif = valeur 1 > valeur 2.
Sortie médiane	0x	Aucun	Etat actif = valeur 1 = valeur 2.
Sortie basse	0x	Aucun	Etat actif = valeur 1 < valeur 2.

SWAP : Permutation de bits VME

162

Présentation

Introduction

Ce chapitre décrit l'instruction SWAP.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : SWAP : Permutation de bits VME	1086
Représentation : SWAP : Permutation de bits VME	1087

Description sommaire : SWAP : Permutation de bits VME

Description de la fonction

Le bloc SWAP permet à l'utilisateur de lancer l'une des trois commandes de permutation :

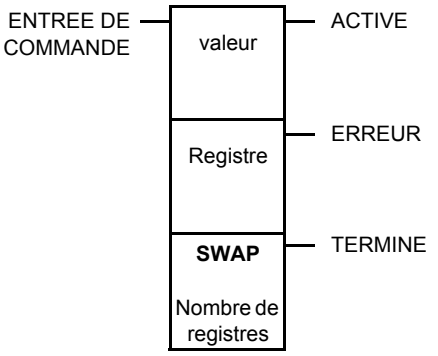
- permutation des bits de poids fort et des bits de poids faible dans un mot 16 bits
- permutation des bits de poids fort et des bits de poids faible dans un mot double 32 bits
- permutation (inverse) des bits dans l'octet de poids faible d'un registre

Note : Disponible uniquement sur l'automate Quantum VME-424/X.

Représentation : SWAP : Permutation de bits VME

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	L'état actif déclenche une opération SWAP
Valeur (partie haute)		INT, UINT, WORD	Contient une constante comprise entre 1 et 3, correspondant au type de permutation à exécuter : **1.** Permutation des bits de poids fort et des bits de poids faible dans un mot 16 bits. **2.** Permutation des bits de poids fort et des bits de poids faible dans un mot double 32 bits. **3.** Permutation (inverse) des bits dans l'octet de poids faible d'un registre.
Registre (partie médiane)	3x, 4x	INT, UINT, WORD	Contient le registre sur lequel la permutation doit être exécutée
Nombre de registres (partie basse)		INT, UINT, WORD	Contient une constante indiquant le nombre de registres à faire permuter, en commençant par le registre source.
Sortie haute	0x	Aucun	Donne une image de l'entrée haute
Sortie médiane	0x	Aucun	Erreur
Sortie basse	0x	Aucun	Permutation réussie

TTR : Table vers registre

163

Présentation

Introduction

Ce chapitre décrit l'instruction TTR.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : TTR : Table vers registre	1090
Représentation : TTR : Table vers registre	1091

Description sommaire : TTR : Table vers registre

Description de la fonction

Le bloc Table vers registre est l'une des quatre instructions de remplacement du 484.

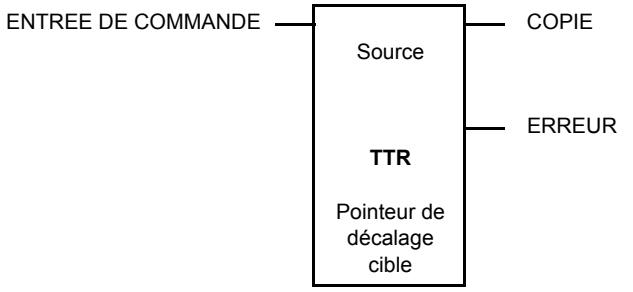
Elle copie le contenu d'un registre source (entrée ou sortie) dans un registre de sortie déterminé par la constante en partie basse. Ce registre source est désigné par le registre d'entrée ou de sortie indiqué en partie haute. Seule une telle opération peut être prise en charge par le système à chaque cycle.

Note : Disponible uniquement sur les 984-351 et 984-455.

Représentation : TTR : Table vers registre

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Source de commande
Source (partie haute)	3x, 4x	INT, UINT	La partie source (ou partie haute) contient l'adresse du registre source. Les données placées à l'adresse du registre source sont copiées à l'adresse cible, déterminée par le pointeur de décalage cible.
Cible (partie basse)	(1 à 254) (801 à 824)	INT, UINT	Le pointeur est un registre 3xxxx ou 4xxxx dont le contenu indique la source. Une valeur comprise entre 1 et 254 indique un registre de sortie (40001 - 40254) tandis qu'une valeur comprise entre 801 et 832 indique un registre d'entrée (30001 - 30032). Si la valeur est en dehors de cette plage, l'opération n'est pas exécutée et le rail ERREUR est alimenté.
Sortie haute	0x	Aucun	Transmet le courant lorsque l'entrée haute est alimentée
Sortie basse	0x	Aucun	Valeur du pointeur hors limites

T --> R Table vers registre

164

Présentation

Introduction

Ce chapitre décrit l'instruction T→R.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1094
Représentation : T → R : Copie de registre vers table	1095
Description des paramètres	1097

Description sommaire

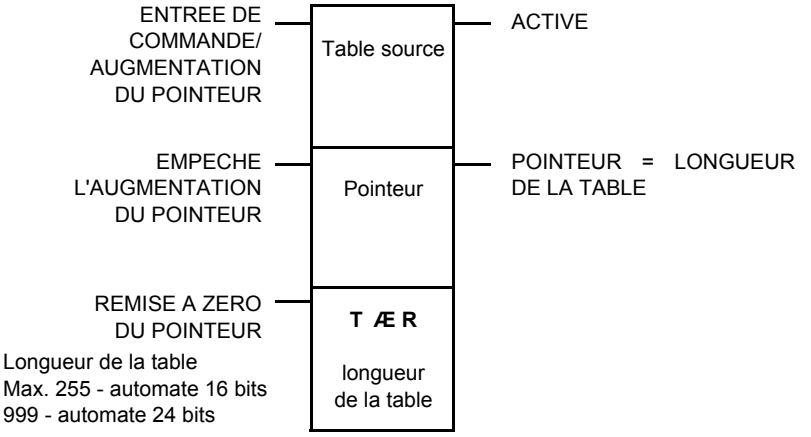
Description de la fonction

L'instruction T→R copie la configuration binaire d'un registre ou de 16 valeurs TOR successives d'une table vers un registre de sortie donné. Elle peut gérer le transfert d'un registre par cycle. Elle dispose de trois entrées de commande et génère deux sorties possibles.

Représentation : T → R : Copie de registre vers table

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = copie les données source et incrémente la valeur du pointeur
Entrée médiane (voir *p. 1097*)	0x, 1x	Aucun	Etat actif = fige la valeur du pointeur
Entrée basse (voir *p. 1097*)	0x, 1x	Aucun	Etat actif = remet à zéro la valeur du pointeur
Table source (partie haute)	0x, 1x, 3x, 4x	INT, UINT, WORD	Premier registre ou référence TOR de la table source. Un registre ou une chaîne de valeurs TOR successives de cette table sera copié en un cycle.
Pointeur (voir *p. 1097*) (partie médiane)	4x	INT, UINT	Pointeur vers la cible dans laquelle les données source seront copiées
Longueur de la table (partie basse)		INT, UINT	Longueur de la table source : nombre de registres à copier, dans la plage : 1 à 999 Longueur : Max. 255 - automate 16 bits Max. 999 - automate 24 bits
Sortie haute	0x	Aucun	Donne une image de l'entrée haute
Sortie médiane	0x	Aucun	Etat actif = valeur du pointeur = longueur de la table (l'instruction ne peut pas incrémenter au–delà)

Description des paramètres

Entrée médiane	Lorsque l'entrée médiane est à l'état actif, la valeur actuelle mémorisée dans le registre du pointeur est figée tandis que l'opération DX continue. Par conséquent, les mêmes données de table seront écrites dans le registre cible à chaque cycle.
Entrée basse	Lorsque l'entrée basse est à l'état actif, la valeur du pointeur est remise à zéro. L'opération de transfert DX suivante copie alors donc le premier registre cible dans la table.
Pointeur (partie médiane)	Le registre 4x mémorisé en partie médiane est un pointeur vers la cible dans laquelle les données source seront copiées. Le registre cible est le registre 4x successif suivant immédiatement le pointeur. Si par exemple la partie médiane affiche un pointeur de 400100, le registre cible pour la copie T→R est alors 400101. La valeur mémorisée dans le registre du pointeur indique le registre de la table source qui sera copié dans le registre cible lors du cycle actuel. La valeur 0 du pointeur indique que la configuration binaire du premier registre de la table source sera copiée vers la cible ; la valeur 1 du registre du pointeur indique que la configuration binaire du deuxième registre de la table source sera copiée dans le registre cible, etc.

T --> T : Table vers table

165

Présentation

Introduction

Ce chapitre décrit l'instruction T→ T.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1100
Représentation : T → T : Copie de table vers table	1101
Description des paramètres	1103

Description sommaire

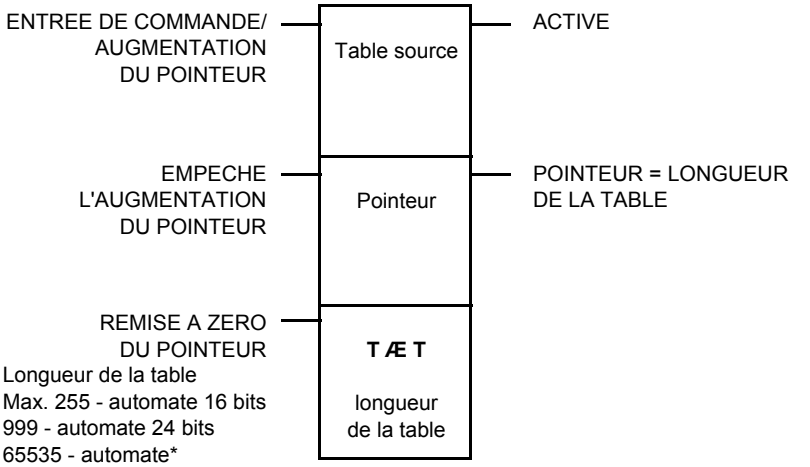
Description de la fonction

L'instruction T→T copie la configuration binaire d'un registre ou de 16 données TOR depuis une position dans une table vers une position équivalente dans une autre table de registres. Elle peut gérer le transfert d'un registre par cycle. Elle dispose de trois entrées de commande et génère deux sorties possibles.

Représentation : T → T : Copie de table vers table

Symbole

Représentation de l'instruction



*Disponible sur :

- les automates E685/785
- les automates L785
- les automates de la série Quantum

Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = copie les données source et incrémente la valeur du pointeur
Entrée médiane (voir *p. 1103*)	0x, 1x	Aucun	Etat actif = fige la valeur du pointeur
Entrée basse (voir *p. 1103*)	0x, 1x	Aucun	Etat actif = remet à zéro la valeur du pointeur
Table source (partie haute)	0x, 1x, 3x, 4x	INT, UINT, WORD	Premier registre ou référence TOR de la table source. Un registre ou une chaîne de valeurs TOR successives de cette table sera copié en un cycle.
Pointeur (voir *p. 1103*) (partie médiane)	4x	INT, UINT	Pointeur vers les tables source et cible
Longueur de la table (partie basse)		INT, UINT	Longueur des tables source et cible (doivent être égales en longueur) Plage : Max. 255 - automate 16 bits Max. 999 - automate 24 bits Max. 65535 - 785L
Sortie haute	0x	Aucun	Donne une image de l'entrée haute
Sortie médiane	0x	Aucun	Etat actif = valeur du pointeur = longueur de la table (l'instruction ne peut pas incrémenter au–delà)

Description des paramètres

Entrée médiane

Lorsque l'entrée dans la partie médiane est à l'état actif, la valeur actuelle mémorisée dans le registre du pointeur est figée tandis que l'opération DX se poursuit. De nouvelles données sont alors copiées vers la cible, qui écraseront les données copiées au cycle précédent.

Entrée basse

Lorsque l'entrée dans la partie basse est à l'état actif, la valeur du registre du pointeur est remise à zéro. Ceci provoque la copie par l'opération de transfert DX suivante des données source dans le premier registre de la table cible.

Pointeur (partie médiane)

Le registre 4x mémorisé en partie médiane est un pointeur vers les tables source et cible, indiquant d'où à où les données seront copiées pendant le cycle en cours. Le premier registre de la table cible est le registre 4x successif suivant immédiatement le pointeur. Si, par exemple, la partie médiane affiche une référence de pointeur de 400100, le premier registre de la table cible est 400101.

La valeur mémorisée dans le registre du pointeur indique le registre de la table source qui sera copié dans la table cible. Comme la longueur des deux tables est égale et que la copie T→T s'effectue vers le registre équivalent de la table cible, la valeur actuelle du registre du pointeur indique vers quel registre de la table cible les données source seront copiées.

La valeur 0 du registre du pointeur indique que la configuration binaire du premier registre de la table source sera copiée vers le premier registre de la table cible ; la valeur 1 du registre du pointeur indique que la configuration binaire du deuxième registre de la table source sera copiée vers le deuxième registre de la table cible, etc.

Temporisation T.01 : Temporisation au centième de seconde

166

Présentation

Introduction

Ce chapitre décrit l'instruction Temporisation T.01.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1106
Représentation : T.01 : Temporisation au centième de seconde	1107

Description sommaire

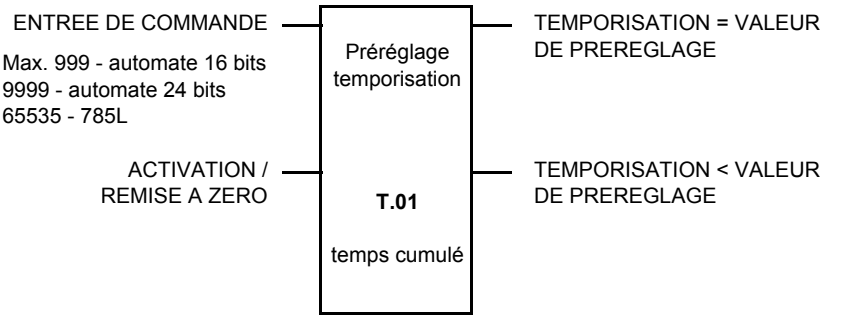
Description de la fonction

L'instruction T.01 mesure le temps par intervalles d'un centième de seconde. Elle peut être utilisée pour la temporisation d'un événement ou la création d'un retard. T.01 dispose de deux entrées de commande et peut générer l'une de deux sorties possibles.

Représentation : T.01 : Temporisation au centième de seconde

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat repos → Etat actif = déclenche l'opération de temporisation : le temps est incrémenté par centièmes de seconde lorsque les entrées haute et basse sont à l'état actif.
Entrée basse	0x, 1x	Aucun	Etat repos = temps cumulé remis à 0 Etat actif = incrémentation de la temporisation
Préréglage temporisation (partie haute)	3x, 4x	INT, UINT	Valeur préréglée (nombre d'incréments d'un centième de seconde), peut être affichée explicitement sous forme d'entier ou mémorisée dans un registre. Plage : Max. 255 - automate 16 bits Max. 999 - automate 24 bits Max. 65535 - 785L
Temps cumulé (partie basse)	4x	INT, UINT	Temps cumulé en incréments d'un centième de seconde.
Sortie haute	0x	Aucun	Etat actif = temps cumulé = préréglage de la temporisation.
Sortie basse	0x	Aucun	Etat actif = temps cumulé < préréglage de la temporisation.

Temporisation T0.1 : Temporisation au dixième de seconde

167

Présentation

Introduction

Ce chapitre décrit l'instruction Temporisation T0.1.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1110
Représentation : T0.1 : Temporisation au dixième de seconde	1111

Description sommaire

Description de la fonction

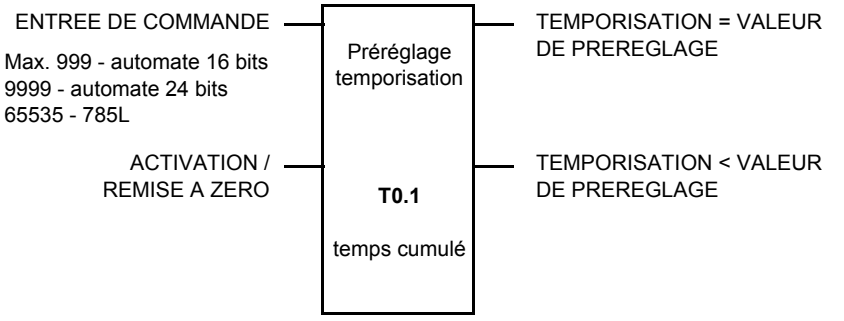
L'instruction T0.1 mesure le temps par intervalles d'un dixième de seconde. Elle peut être utilisée pour la temporisation d'un événement ou la création d'un retard. T0.1 dispose de deux entrées de commande et peut générer une sortie parmi deux possibles.

Note : Si vous mettez en cascade des temporisations T0.1 avec des valeurs préréglées de 1, les temporisations se termineront toutes ensemble ; pour éviter ce problème, mettez les valeurs préréglées à 10 et remplacez par une temporisation T.01 (voir *p. 1105*).

Représentation : T0.1 : Temporisation au dixième de seconde

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat repos → Etat actif = déclenche l'opération de temporisation : le temps est incrémenté par dixièmes de seconde lorsque les entrées haute et basse sont à l'état actif
Entrée basse	0x, 1x	Aucun	Etat repos = temps cumulé remis à 0 Etat actif = incrémentation de la temporisation
Préréglage temporisation (partie haute)	3x, 4x	INT, UINT	Valeur préréglée (nombre d'incréments d'un dixième de seconde), peut être affichée explicitement sous forme d'entier ou mémorisée dans un registre Plage : Max. 255 - automate 16 bits Max. 999 - automate 24 bits Max. 65535 - 785L
Temps cumulé (partie basse)	4x	INT, UINT	Temps cumulé en incréments d'un dixième de seconde.
Sortie haute	0x	Aucun	Etat actif = temps cumulé = préréglage de la temporisation.
Sortie basse	0x	Aucun	Etat actif = temps cumulé < préréglage de la temporisation

Temporisation T1.0 : Temporisation à la seconde

168

Présentation

Introduction

Ce chapitre décrit l'instruction Temporisation T1.0.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1114
Représentation : T1.0 : Temporisation à la seconde	1115

Description sommaire

Description de la fonction

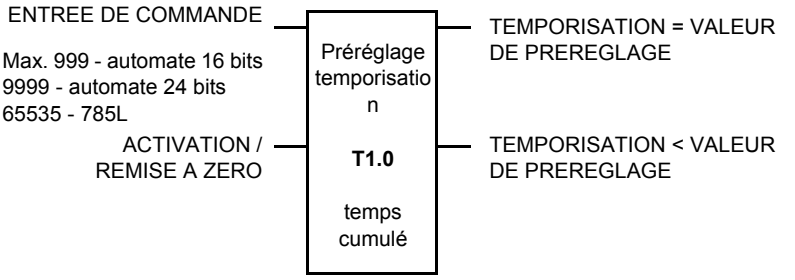
L'instruction Temporisation T1.0 mesure le temps par incréments d'une seconde. Elle peut être utilisée pour la temporisation d'un événement ou la création d'un retard. T1.0 dispose de deux entrées de commande et peut générer l'une de deux sorties possibles.

Note : Si vous mettez en cascade des temporisations T1.0 avec des valeurs préréglées de 1, les temporisations se termineront toutes ensemble ; pour éviter ce problème, mettez les valeurs préréglées à 10 et remplacez par une temporisation T0.1 (voir *p. 1109*).

Représentation : T1.0 : Temporisation à la seconde

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat repos → Etat actif = déclenche l'opération de temporisation : le temps est incrémenté en secondes lorsque les entrées haute et basse sont activées
Entrée basse	0x, 1x	Aucun	Etat repos = temps cumulé remis à 0 Etat actif = incrémentation de la temporisation
préréglage temporisation (partie haute)	3x, 4x	INT, UINT	Valeur préréglée (nombre d'incréments d'une seconde), peut être affichée explicitement sous forme d'entier ou mémorisée dans un registre Plage : Max. 255 - automate 16 bits Max. 999 - automate 24 bits Max. 65535 785L
Temps cumulé (partie basse)	4x	INT, UINT	Temps cumulé en incréments d'une seconde.
Sortie haute	0x	Aucun	Etat actif = temps cumulé = préréglage de la temporisation
Sortie basse	0x	Aucun	Etat actif = temps cumulé < préréglage de la temporisation

Temporisation T1MS : Temporisation à la milliseconde

169

Présentation

Introduction

Ce chapitre décrit l'instruction Temporisation T1MS.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1118
Représentation : T1MS : Temporisation à la milliseconde	1119
Exemple	1121

Description sommaire

Description de la fonction

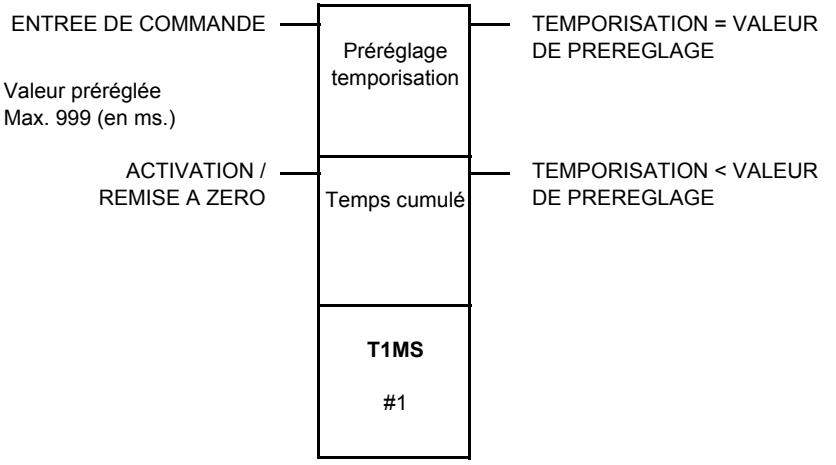
L'instruction de temporisation T1MS mesure le temps par incréments d'une milliseconde. Elle peut être utilisée pour la temporisation d'un événement ou la création d'un retard.

Note : L'instruction T1MS est disponible uniquement sur le B984-102, sur le Micro 311, 411, 512 et 612, ainsi que sur le Quantum 424 02.

Représentation : T1MS : Temporisation à la milliseconde

Symbole

Représentation de l'instruction



Description des paramètres

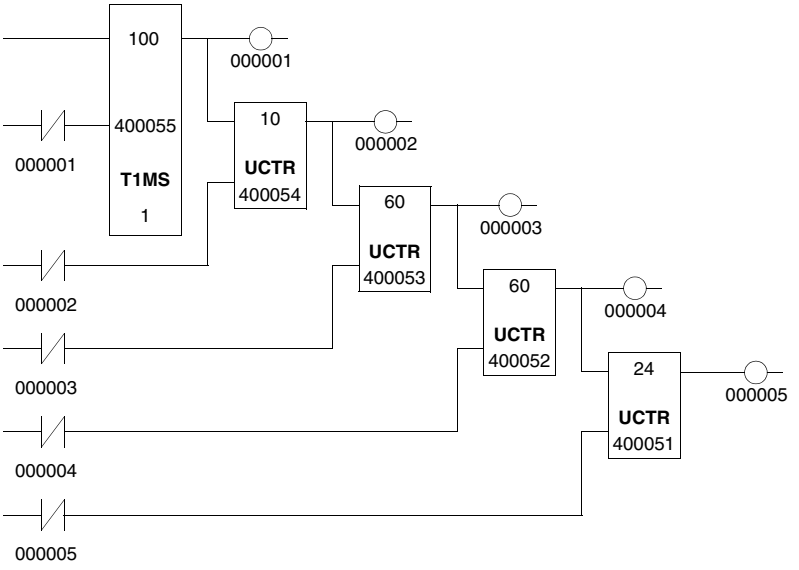
Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = lance la commande de temporisation : le temps incrémente en millisecondes lorsque les entrées haute et médiane sont activées.
Entrée médiane	0x, 1x	Aucun	Etat repos = temps cumulé remis à 0 Etat actif = incrémentation de la temporisation
Préréglage temporisation (partie haute)	3x, 4x	INT, UINT	La valeur préréglée (nombre d'incréments en millisecondes que la temporisation peut cumuler), peut être affichée explicitement comme un entier (compris entre 1 et 999) ou mémorisée dans un registre
Temps cumulé (partie médiane)	4x	INT, UINT	Temps cumulé par incréments d'une milliseconde.
#1 (partie basse)		INT, UINT	Valeur constante de 1
Sortie haute	0x	Aucun	Etat actif = temps cumulé = préréglage de la temporisation
Sortie médiane	0x	Aucun	Etat actif = temps cumulé < préréglage de la temporisation

Exemple

Exemple de temporisation à la milliseconde

Voici, sous la forme d'un schéma à contacts, une horloge en temps réel avec une précision à la milliseconde :



L'instruction T1MS est programmée pour transmettre du courant toutes les 100 ms ; elle est suivie de cinq compteurs (voir *p. 1133*) en cascade qui stockent le temps respectivement en centièmes de seconde, dixièmes de seconde, secondes, minutes et heures.

Au début du traitement de la logique, la valeur du temps cumulée commence à incrémenter dans le registre 40055 du bloc T1MS. Au bout de 100 incréments d'une milliseconde, la sortie haute transmet le courant et active la bobine 00001. A ce moment–là, la valeur du registre 40055 du temporisateur est remise à 0. La valeur des comptages cumulés dans le registre 40054 du premier bloc UCTR incrémente de 1 en 1, indiquant que 100 ms se sont écoulées. Comme le compte du temps cumulé dans le bloc T1MS n'est plus égal à la temporisation prédéfinie, la temporisation commence à réincrémenter le temps en millisecondes.

Lorsque le compte cumulé du registre 40054 de la première instruction UCTR passe à 10, la sortie haute de ce bloc d'instruction transmet le courant et active la bobine 00002. La valeur du registre 40054 est alors réinitialisée, et le compte cumulé du registre 40053 du second bloc UCTR incrémente de 1 unité.

Comme le temps est cumulé dans chaque compteur, l'heure du jour peut être lue dans cinq registres de sortie comme suit :

Registre	Unité de temps	Plage valide
40055	Millièmes de seconde	0 à 100
40054	Dixièmes de seconde	0 à 10
40053	Secondes	0 à 60
40052	Minutes	0 à 60
40051	Heures	0 à 24

TBLK : Table vers Bloc

170

Présentation

Introduction

Ce chapitre décrit l'instruction TBLK.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1124
Représentation : TBLK : Copie de table vers bloc	1125
Description des paramètres	1127

Description sommaire

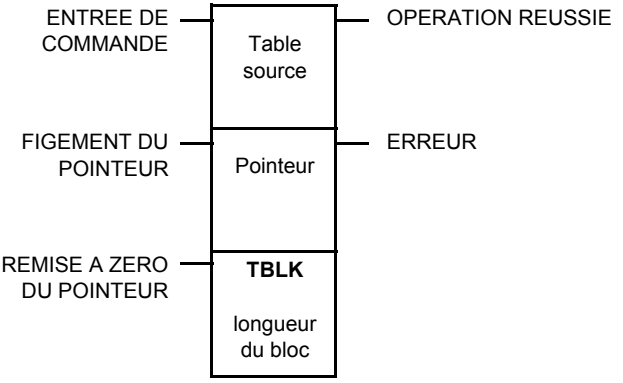
Description de la fonction

L'instruction TBLK (table-vers-bloc) combine en une même instruction les fonctions TÆR (voir *p. 1093*) et BLKM (voir *p. 133*). En un seul cycle, elle peut copier jusqu'à 100 registres 4x successifs d'une table vers un bloc cible. Le bloc cible a une longueur fixe. Le bloc de registres copiés depuis la table source a la même longueur, mais la longueur totale de la table source n'est limitée que par le nombre de registres de votre configuration système.

Représentation : TBLK : Copie de table vers bloc

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = déclenche l'opération de déplacement
Entrée médiane (voir *p. 1127*)	0x, 1x	Aucun	Etat actif = fige le pointeur Les entrées des parties médiane et basse peuvent être utilisées pour contrôler la valeur dans le pointeur afin que la taille de la table source puisse être contrôlée. **Important :** Vous devez utiliser une logique externe en combinaison avec l'entrée médiane ou l'entrée basse pour limiter la valeur du pointeur cible dans une plage sûre. Lorsque l'entrée de la partie médiane est à l'état actif, la valeur du registre du pointeur est figée tandis que l'opération TBLK se poursuit. Le même bloc de données source est alors copié vers la table cible à chaque cycle.
Entrée basse (voir *p. 1127*)	0x, 1x	Aucun	Etat actif = remet à zéro le pointeur

Paramètres	Référence de mémoire d'état	Type de données	Signification
Table source (voir *p. 1127*) (partie haute)	4x	INT, UINT, WORD	Premier registre de sortie de la table source Le registre 4xxxx mémorisé en partie haute est le premier registre de sortie de la table source. **Note :** La table source est segmentée en une série de blocs de registres, dont chacun a la même longueur que le bloc cible. De ce fait, la taille de la table source est un multiple de la longueur du bloc cible, mais elle n'est pas spécifiquement définie dans l'instruction. Si elle n'est pas contrôlée, la table source risque d'occuper tous les registres 4xxxx disponibles dans la configuration de l'automate.
Pointeur (voir *p. 1127*) (partie médiane)	4x	INT, UINT	Pointeur vers le bloc source, bloc cible
Longueur du bloc (partie basse)		INT, UINT	Nombre de registres du bloc cible et nombre de blocs de la table source ; dans la plage : 1 à 100
Sortie haute	0x	Aucun	Etat actif = transfert réussi
Sortie médiane	0x	Aucun	Etat actif = erreur / transfert impossible

Description des paramètres

Entrée médiane

Lorsque l'entrée médiane est à l'état actif, la valeur du registre du pointeur est figée tandis que l'opération TBLK se poursuit. Le même bloc de données source est alors copié vers la table cible à chaque cycle.

Entrée basse

Lorsque l'entrée basse est à l'état actif, la valeur du pointeur est remise à zéro. L'opération TBLK copie alors les données du premier bloc de registres vers la table source.

⚠ ATTENTION

Limiter la valeur du pointeur cible à une plage sûre.

Vous devez utiliser une logique externe en combinaison avec les entrées médiane et basse pour limiter la valeur du pointeur cible à une plage sûre.

Le non-respect de cette directive peut entraîner des lésions corporelles et/ou des dommages matériels.

Table source (partie haute)

Le registre 4x mémorisé en partie haute est le premier registre de sortie de la table source.

Note : La table source est segmentée en une série de blocs de registres, dont chacun a la même longueur que le bloc cible. De ce fait, la taille de la table source est un multiple de la longueur du bloc cible, mais elle n'est pas spécifiquement définie dans l'instruction. Si elle n'est pas contrôlée, la table source risque d'occuper tous les registres 4x disponibles dans la configuration de l'automate.

Pointeur (partie médiane)

Le registre 4x mémorisé en partie médiane est le pointeur vers le bloc source. Le premier registre du bloc cible est le registre successif suivant immédiatement le pointeur. Si, par exemple, le pointeur est le registre 400107, le premier registre du bloc cible est 400108.

La valeur mémorisée dans le pointeur indique le bloc de données de la table source qui sera copié vers le bloc cible. Cette valeur indique un numéro de bloc dans la table source.

TEST : Comparaison de deux valeurs

171

Présentation

Introduction

Ce chapitre décrit l'instruction TEST.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1130
Représentation : TEST : Comparaison de deux valeurs	1131

Description sommaire

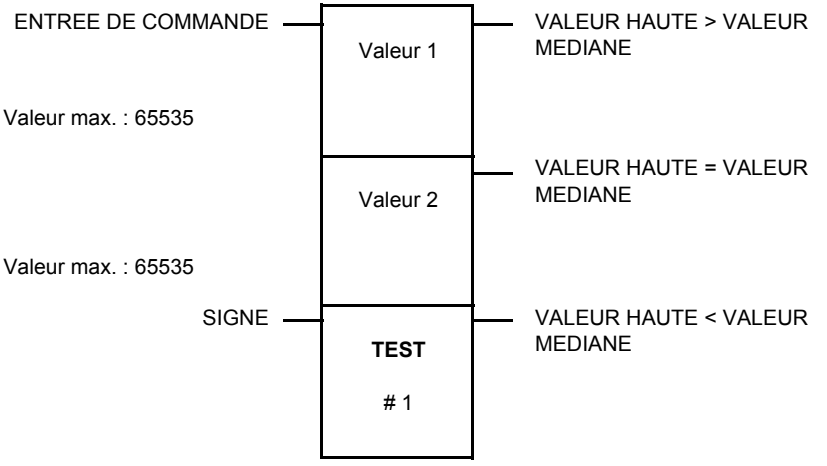
Description de la fonction

L'instruction TEST effectue la comparaison signée ou non signée des valeurs 16 bits des parties haute et médiane, et indique la relation entre elles via les sorties du bloc.

Représentation : TEST : Comparaison de deux valeurs

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = compare les valeurs 1 et 2
Entrée basse	0x, 1x	Aucun	Etat actif = opération signée Etat repos = opération non signée
Valeur 1 (partie haute)	3x, 4x	INT, UINT	Valeur 1, peut être affichée explicitement sous forme d'entier (compris entre 1 et 65 535) ou mémorisée dans un registre
Valeur 2 (partie médiane)	3x, 4x	INT, UINT	Valeur 2, peut être affichée explicitement sous forme d'entier (compris entre 1 et 65 535) ou mémorisée dans un registre
1 (partie basse)		INT, UINT	Valeur constante, ne peut pas être changée
Sortie haute	0x	Aucun	Etat actif = valeur 1 > valeur 2
Sortie médiane	0x	Aucun	Etat actif = valeur 1 = valeur 2
Sortie basse	0x	Aucun	Etat actif = valeur 1 < valeur 2

UCTR : Compteur

172

Présentation

Introduction

Ce chapitre décrit l'instruction UCTR.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1134
Représentation : UCTR : Compteur	1135

Description sommaire

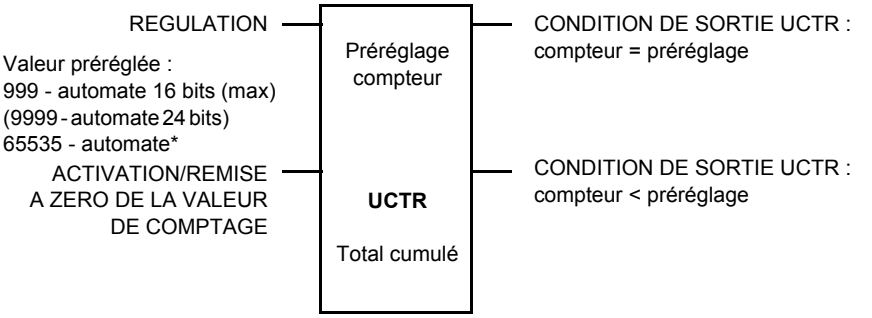
Description de la fonction

L'instruction UCTR compte les transitions entre l'état repos et l'état actif de l'entrée de commande, depuis zéro jusqu'à une valeur de compteur préréglée.

Représentation : UCTR : Compteur

Symbole

Représentation de l'instruction



*Disponible sur :

- les automates E685/785
- les automates L785
- les automates de la série Quantum

Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat repos → Etat actif = déclenche l'opération de comptage
Entrée basse	0x, 1x	Aucun	Etat repos = remet à 0 l'accumulateur Etat actif = compteur actif.
Préréglage compteur (partie haute)	3x, 4x	INT, UINT	Valeur préréglée, peut être affichée explicitement sous forme d'entier ou mémorisée dans un registre Valeur préréglée : Max. 255 - automate 16 bits Max. 999 - automate 24 bits Max. 65535 - 785L
Total cumulé (partie basse)	4x	INT, UINT	Valeur de comptage (valeur réelle), laquelle est incrémentée de 1 à chaque transition de l'état repos vers l'état actif de l'entrée haute jusqu'à ce qu'elle atteigne la valeur définie de préréglage du compteur.
Sortie haute	0x	Aucun	Etat actif = total cumulé = préréglage compteur
Sortie basse	0x	Aucun	Etat actif = total cumulé < préréglage compteur

VMER : Lecture de VME

173

Présentation

Introduction

Ce chapitre décrit l'instruction VMER.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : VMER : Lecture de VME	1138
Représentation : VMER : Lecture de VME	1139
Description des paramètres : VMER : Lecture de VME	1141

Description sommaire : VMER : Lecture de VME

Description de la fonction

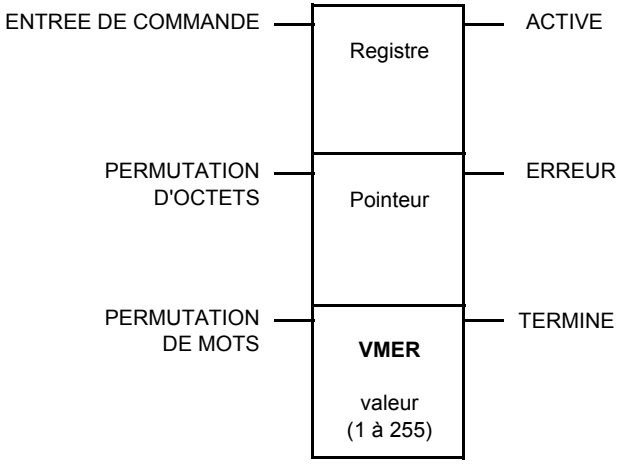
Le bloc Lecture de VME permet à l'utilisateur de lire des données à partir d'équipements situés sur le bus VME. Si la permutation des octets est active, l'octet de poids fort est échangé avec l'octet de poids faible d'un mot une fois qu'il a été lu à partir du bus VME. Si la permutation des mots est active, le mot supérieur est échangé avec le mot inférieur d'un mot double une fois qu'il a été lu. Une erreur se produit si les deux entrées sont activées simultanément.

Note : Disponible uniquement sur l'automate Quantum VME-424/X.

Représentation : VMER : Lecture de VME

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	L'état actif déclenche la lecture
Entrée médiane	0x, 1x	Aucun	Etat actif = permutation des octets
Entrée basse	0x, 1x	Aucun	Etat actif = permutation des mots
Registre (partie haute)	4x	INT, UINT, WORD	La parte haute compte cinq registres de contrôle. Ils sont affectés comme suit : 4x : Code du modificateur d'adresses VME (39, 3A, 3D, 3E, 29 ou 2D) 4x+1 à 4x+4 : Bloc de contrôle VME Pour plus d'informations, reportez-vous à la table nommée *p. 1141*.
Pointeur (partie médiane)	4x	INT, UINT WORD	Pointeur désignant le premier registre cible. Pour plus d'informations, reportez-vous à la table nommée *p. 1141*.
Valeur (partie basse)		INT, UINT WORD	Constante indiquant le nombre de registres cible vers lesquels les données sont transférées. Cette constante peut varier entre 1 et 255.
Sortie haute	0x	Aucun	Etat actif lorsque l'entrée haute est alimentée
Sortie médiane	0x	Aucun	Etat actif lorsqu'une erreur se produit
Sortie basse	0x	Aucun	Etat actif lorsque la lecture est terminée

Description des paramètres : VMER : Lecture de VME

Bloc de contrôle VME

Il s'agit du bloc de contrôle VME.

Registre	Description
Affiché	Code du modificateur d'adresses VME
Premier implicite	Etat des codes d'erreur Reportez-vous à la table Etat des codes d'erreur.
Deuxième implicite	Longueur des données devant être lues/écrites
Troisième implicite	Adresse de l'équipement VME (octet de poids faible)
Quatrième implicite	Adresse de l'équipement VME (octet de poids fort)

Etat des codes d'erreur

Il s'agit de la table Etat des codes d'erreur.

Erreur	Description
01	Décompte des mots incorrect. Le nombre de mots doit être pair.
02	Longueur incorrecte, supérieure à 255
03	Longueur des données incorrecte La longueur était nulle ou supérieure à 255
04	Modificateur d'adresses incorrect dans le premier bloc de contrôle
05	Commande incorrecte en partie haute du bloc SWAP
06	Interface de bus VME incorrecte
07	L'adresse du bus VME n'existe pas
08	Dépassement de délai du 486 VME
09	L'interface du bus VME n'a pas été configurée
10	Les entrées de permutation d'OCTETS et de MOTS ont été sélectionnées
11	Correspondance avec le type déterminé par le code AM (A16 ou A2)

VMEW : Ecriture de VME

174

Présentation

Introduction

Ce chapitre décrit l'instruction VMEW.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : VMEW : Ecriture de VME	1144
Représentation : VMEW : Ecriture de VME	1145
Description des paramètres : VMEW : Ecriture de VME	1147

Description sommaire : VMEW : Ecriture de VME

Description de la fonction

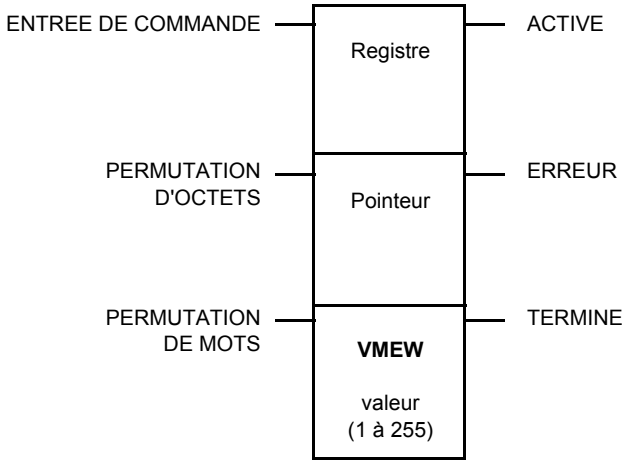
Le bloc Ecriture de VME permet à l'utilisateur d'écrire des données sur des équipements situés sur le bus VME. Si la permutation des octets est active, l'octet de poids fort est échangé avec l'octet de poids faible d'un mot avant qu'il soit écrit dans le bus VME. Si la permutation des mots est active, le mot supérieur est échangé avec le mot inférieur d'un mot double avant qu'il soit écrit. Une erreur se produit si les deux entrées sont activées simultanément.

Note : Disponible uniquement sur l'automate Quantum VME-424/X.

Représentation : VMEW : Ecriture de VME

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	L'état actif déclenche la lecture
Entrée médiane	0x, 1x	Aucun	Etat actif = permutation des octets
Entrée basse	0x, 1x	Aucun	Etat actif = permutation des mots
Registre (partie haute)	4x	INT, UINT WORD	La parte haute compte cinq registres de contrôle. Ils sont affectés comme suit : 4x : Octet de poids fort : code du modificateur d'adresses VME (39, 3A, 3D, 3E, 29 ou 2D) 4x : Octet de poids faible : taille des bus de données 4x + 1 à 4x + 4 : Bloc de contrôle VME. Pour plus d'informations, reportez-vous à la table nommée *p. 1147*.
Pointeur (partie médiane)	3x, 4x	INT, UINT WORD	Pointeur désignant le premier registre cible. Pour plus d'informations, reportez-vous à la table nommée *p. 1147*.
Valeur (partie basse)		INT, UINT WORD	Constante indiquant le nombre de registres cible vers lesquels les données sont transférées. Cette constante peut varier entre 1 et 255.
Sortie haute	0x	Aucun	Etat actif lorsque l'entrée haute est alimentée. Transmet le courant lorsque l'entrée haute est alimentée
Sortie médiane	0x	Aucun	Etat actif lorsqu'une erreur se produit
Sortie basse	0x	Aucun	Etat actif lorsque l'écriture est terminée

Description des paramètres : VMEW : Ecriture de VME

Bloc de contrôle VME

Il s'agit du bloc de contrôle VME.

Registre	Description
Affiché	Code du modificateur d'adresses VME
Premier implicite	Etat des codes d'erreur. Reportez-vous à la table Etat des codes d'erreur.
Deuxième implicite	Longueur des données devant être lues/écrites
Troisième implicite	Adresse de l'équipement VME (octet de poids faible)
Quatrième implicite	Adresse de l'équipement VME (octet de poids fort)

Etat des codes d'erreur

Il s'agit de la table Etat des codes d'erreur.

Erreur	Description
01	Décompte des mots incorrect. Le nombre de mots doit être pair.
02	Longueur incorrecte, supérieure à 255
03	Longueur des données incorrecte. La longueur était nulle ou supérieure à 255
04	Modificateur d'adresses incorrect dans le premier bloc de contrôle
05	Commande incorrecte en partie haute du bloc SWAP
06	Interface de bus VME incorrecte
07	L'adresse du bus VME n'existe pas
08	Dépassement de délai du 486 VME
09	L'interface du bus VME n'a pas été configurée
10	Les entrées de permutation d'OCTETS et de MOTS ont été sélectionnées
11	Correspondance avec le type déterminé par le code AM (A16 ou A2)

WRIT : Ecrire

175

Présentation

Introduction

Ce chapitre décrit l'instruction WRIT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1150
Représentation : WRIT : Ecriture d'un port ASCII	1151
Description des paramètres	1153

Description sommaire

Description de la fonction

L'instruction WRIT envoie un message depuis l'automate à un afficheur ASCII (écran, imprimante etc.) en passant par le lien de communication RIO.

Pendant l'envoi du message, WRIT effectue les fonctions suivantes :

- vérifie si les paramètres de communication ASCII sont corrects, par exemple le numéro de port et le numéro de message
- vérifie la longueur des champs de données variables
- détecte les erreurs et les enregistre
- signale l'état de l'interface RIO

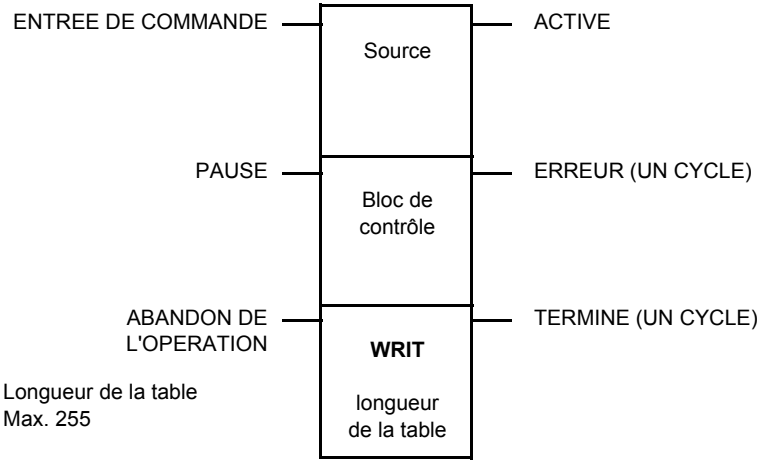
WRIT nécessite deux tables de registres : une table source dans laquelle les données variables (le message) sont copiées et un bloc de contrôle dans lequel les paramètres du port de communication et les paramètres des messages sont identifiés.

vous trouverez de plus amples informations sur la mise en forme de messages à la section *p. 89*.

Représentation : WRIT : Ecriture d'un port ASCII

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = lance WRIT
Entrée médiane	0x, 1x	Aucun	Etat actif = suspend l'opération WRIT
Entrée basse	0x, 1x	Aucun	Etat actif = abandonne l'opération WRIT
Source (voir *p. 1153*) (partie haute)	3x, 4x	INT, UINT, WORD	Table source
Bloc de contrôle (voir *p. 1153*) (partie médiane)	4x	INT, UINT, WORD	Bloc de contrôle ASCII (premier de sept registres de sortie successifs). Pour plus d'informations, reportez-vous à la section *p. 1153*.
Longueur de la table (partie basse)		INT, UINT	Longueur de la table source (nombre de registres dans lesquels les données du message seront mémorisées), dans la plage : 1 à 255
Sortie haute	0x	Aucun	Donne une image de l'entrée haute
Sortie médiane	0x	Aucun	Etat actif = erreur de communication ou l'opération a dépassé les limites de temps (pour un cycle)
Sortie basse	0x	Aucun	Etat actif = opération WRIT terminée (pour un cycle)

Description des paramètres

Table source (partie haute)

La partie haute contient le premier registre 3x ou 4x d'une table source dont la longueur est déterminée en partie basse. Cette table contient les données nécessaires pour remplir le champ des données variables d'un message.

Considérons le message WRIT suivant

vessel #1 temperature is**:** **III**

Le champ ASCII sur 3 caractères **`III`** est le champ de données variables ; les données variables sont chargées, généralement à l'aide de copies DX, vers une table de données de champ de variable.

Bloc de contrôle (partie médiane)

Le registre 4x mémorisé en partie médiane est le premier des sept registres de sortie successifs du bloc de contrôle.

Registre	Définition
Affiché	*Numéro de port et code d'erreur, p. 1154*
Premier implicite	Numéro de message
Deuxième implicite	Nombre de registres nécessaires pour respecter le format
Troisième implicite	Compte du nombre de registres transmis jusque–là
Quatrième implicite	Etat du cycle
Cinquième implicite	Réservé
Sixième implicite	Total de contrôle des registres 0 à 5

Numéro de port et code d'erreur

Numéro de port et code d'erreur

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1 à 4	Code d'erreur API (voir tableau ci-dessous)
5	Inutilisé
6	Entrée de l'équipement ASCII incompatible avec le format
7	Débordement du tampon d'entrée, données reçues trop rapidement sur le RIOP
8	Erreur USART, RIOP a reçu un octet incorrect
9	Format illicite, mauvaise réception RIOP
10	Equipement ASCII hors–ligne, contrôler le câblage
11	Message ASCII terminé trop tôt (en mode clavier)
12 à 16	Numéro du port de communication (1 à 32)

Code d'erreur API

Bit				Signification
1	**2**	**3**	**4**	
0	0	0	1	Erreur sur l'entrée RIOP depuis le périphérique ASCII
0	0	1	0	Réponse d'anomalie depuis RIOP, données incorrectes
0	0	1	1	Nombre séquentiel de RIOP diffère de la valeur attendue
0	1	0	0	Erreur de total de contrôle du registre utilisateur, souvent due à une modification des registres READ lorsque le bloc est actif
0	1	0	1	Détection d'un numéro de port ou de message invalide
0	1	1	0	Abandon par l'utilisateur, partie basse activée
0	1	1	1	Aucune réponse de la station, erreur de communication
1	0	0	0	Partie abandonnée du fait de l'instruction SKP
1	0	0	1	Zone message brouillée, recharger la mémoire
1	0	1	0	Port non configuré dans l'affectation des E/S
1	0	1	1	Demande ASCII illicite
1	1	0	0	Réponse inconnue du port ASCII
1	1	0	1	Elément ASCII illicite détecté dans la logique utilisateur
1	1	1	1	RIOP de l'automate est arrêté

XMIT : Emission

176

Présentation

Introduction

Ce chapitre décrit l'instruction XMIT : Emission.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation générale : XMIT : Emission	1156
Fonctions Modbus du XMIT	1157

Présentation générale : XMIT : Emission

Introduction

Le bloc fonction XMIT (Emission) envoie des messages Modbus d'un automate "maître" à plusieurs automates esclaves ou des chaînes de caractères ASCII du port 1 ou 2 de l'esclave Modbus de l'automate à des imprimantes et des terminaux ASCII. L'instruction envoie ces messages via des modems téléphoniques à accès commuté, des modems radio ou tout simplement des connexions directes.

Pour plus d'informations sur le bloc fonction XMIT, reportez-vous à la section intitulée *p. 1157*.

L'instruction XMIT peut fonctionner en trois modes : communication, état du port et conversion.

Ces modes sont décrits dans les sections suivantes.

- *Bloc de communication p. 1163*
- *Bloc d'état du port p. 1175*
- *Bloc de conversion p. 1183*

En mode de communication, le bloc XMIT exécute les fonctions d'entrée ASCII générales, y compris au format ASCII simple et ASCII terminé. Vous pouvez utiliser un bloc XMIT supplémentaire pour reporter les informations d'état des ports dans des registres, pendant qu'un autre bloc XMIT effectue la fonction de communication ASCII. Vous pouvez importer/exporter des données ASCII ou binaires vers/à partir de votre automate et les convertir en diverses données binaires ou ASCII pour les envoyer à des équipements DCE (Data Communication Equipment, matériel de transmission de données) en fonction des besoins de votre application.

Le bloc comprend des fonctions de diagnostic intégrées qui vérifient qu'aucun autre bloc XMIT n'est actif sur l'automate. Dans le bloc XMIT, une table de commande vous permet de contrôler la liaison de communication entre l'automate et les équipements DCE connectés au port 1 ou 2 Modbus de l'automate. Le bloc XMIT n'active pas le voyant du port lors de la transmission des données.

Note : Le protocole Modbus est un protocole "maître/esclave", conçu pour ne compter qu'un seul maître interrogeant plusieurs esclaves. Par conséquent, si vous utilisez le bloc XMIT dans un réseau avec plusieurs maîtres, vous êtes responsable de la résolution des conflits et de l'évitement des collisions. La programmation de schémas à contacts vous aidera à résoudre facilement ces problèmes.

Fonctions Modbus du XMIT

Présentation

Le bloc fonction XMIT prend en charge les codes fonction Modbus suivants :

- 01 ... 06
- 08
- 15 et 16
- 20 et 21

Pour les messages Modbus, la matrice MSG_OUT doit contenir le tableau de définition Modbus. Le code fonction 01, 02, 03, 04, 05, 06, 15 et 16 du tableau de définition Modbus comporte cinq chiffres. Vous devez définir XMIT_SET.MessageLen sur 5 pour que le XMIT fonctionne correctement. Le tableau de définition Modbus est présenté dans le tableau ci-dessous.

Codes fonction Modbus 01...06

Pour les messages Modbus, la matrice MSG_OUT doit contenir le tableau de définition Modbus. Le code fonction 01, 02, 03, 04, 05, 06, 15 et 16 du tableau de définition Modbus comporte cinq chiffres. Vous devez définir XMIT_SET.MessageLen sur 5 pour que le XMIT fonctionne correctement. Le tableau de définition Modbus est présenté dans le tableau ci-dessous.

Codes fonction du tableau de définition Modbus (01 ... 06, 15 et 16)

Contenu	Description
Code fonction Modbus (MSG_OUT[1])	Le XMIT prend en charge les codes fonction suivants : 01 = Lire plusieurs bobines (0x) 02 = Lire plusieurs entrées TOR (1x) 03 = Lire plusieurs registres de maintien (4x) 04 = Lire plusieurs registres d'entrée (3x) 05 =Ecrire une seule bobine (0x) 06 = Ecrire des registres de maintien uniques (4x) 15 = Ecrire plusieurs bobines (0x) 16 = Ecrire plusieurs registres de maintien (4x)
Quantité (MSG_OUT[2])	Entrez le volume de données dont vous souhaitez l'écriture sur l'automate esclave et le volume de données dont vous souhaitez la lecture à partir de l'automate esclave. Entrez 100 pour lire 100 registres de maintien à partir de l'automate esclave et entrez 32 pour écrire 32 bobines sur un automate esclave. La quantité autorisée dépend du modèle de l'automate. Pour plus d'informations sur les limitations, reportez-vous à l'annexe A.

Contenu	Description
Adresse de l'automate esclave (MSG_OUT[3])	Entrez l'adresse de l'automate Modbus esclave. La plage d'adresses est généralement comprise entre 1 et 247. Pour envoyer un message Modbus à plusieurs automates, indiquez 0 pour l'adresse de l'automate esclave. Il s'agit du mode de diffusion. Le mode de diffusion prend uniquement en charge les codes fonction Modbus qui permettent d'écrire des données sur des automates esclaves depuis l'automate maître. Le mode de diffusion NE prend PAS en charge les codes fonction Modbus qui permettent de lire des données à partir d'automates esclaves.
Zone de données de l'automate esclave (MSG_OUT[4])	Pour une commande de lecture, la zone de données de l'automate esclave correspond à la source des données. Pour une commande d'écriture, la zone de données de l'automate esclave correspond à la cible des données. Si par exemple vous souhaitez lire des bobines (00300 ... 00500) à partir d'un automate esclave, entrez 300 dans ce champ. Lorsque vous souhaitez écrire des données dans le registre (40100) d'un automate esclave à partir d'un automate maître, entrez 100 dans ce champ. En fonction du type de la commande Modbus (écriture ou lecture), les zones de données source et cible doivent être définies comme indiqué dans le tableau Zones de données source et cible ci-dessous.
Zone de données de l'automate maître (MSG_OUT[5])	Pour une commande de lecture, la zone de données de l'automate maître est la zone cible des données renvoyées par l'esclave. Pour une commande d'écriture, la zone de données de l'automate maître correspond à la source des données. Si par exemple vous souhaitez écrire des bobines (00016 ... 00032) situées dans l'automate maître sur un automate esclave, entrez 16 dans ce champ. Si vous souhaitez lire des registres d'entrée (30001 ... 30100) d'un automate esclave et placer les données dans la zone de données de l'automate maître (40100 .... 40199), entrez 100 dans ce champ. Selon le type de commande Modbus (lecture ou écriture), les zones de données source et cible doivent être conformes à celles du tableau ci-dessous.

Zones de données source et cible pour les codes de fonction (01 à 06, 15 et 16)

Code fonction	Zone de données de l'automate maître	Zone de données de l'automate esclave
03 (Lire plusieurs 4x)	4x (cible)	4x (source)
04 (Lire plusieurs 3x)	4x (cible)	3x (source)
01 (Lire plusieurs 0x)	0x (cible)	0x (source)
02 (Lire plusieurs 1x)	0x (cible)	1x (source)

Code fonction	Zone de données de l'automate maître	Zone de données de l'automate esclave
16 (Ecrire plusieurs 4x)	4x (source)	4x (cible)
15 (Ecrire plusieurs 0x)	0x (source)	0x (cible)
05 (Ecrire 0x unique)	0x (source)	0x (cible)
06 (Ecrire 4x unique)	4x (source)	4x (cible)

Pour envoyer 20 messages Modbus depuis l'automate, vous devez transférer 20 tables de définition Modbus l'une après l'autre dans MSG_OUT après chaque opération réussie de XMIT. Vous pouvez également programmer 20 blocs XMIT différents, puis les activer l'un après l'autre en utilisant la logique utilisateur.

Code fonction Modbus (08)

Le tableau de définition Modbus pour les codes fonction Modbus 08 comporte cinq chiffres. Vous devez définir XMIT_SET.MessageLen sur 5 pour que XMIT fonctionne correctement. Pour les messages Modbus, la matrice MSG_OUT doit contenir le tableau de définition Modbus. Le tableau de définition Modbus est présenté ci-dessous.

Codes fonction du tableau de définition Modbus (08)

Contenu	Description
Code fonction Modbus (MSG_OUT[1])	Le bloc XMIT prend en charge le code fonction 08 = Diagnostics.
Diagnostics (MSG_OUT[2])	Entrez la valeur décimale du code de sous-fonction du diagnostic dans ce champ pour exécuter la fonction de diagnostic voulue. Les sous-fonctions de diagnostic suivantes sont prises en charge : Code Description 00 Renvoyer les données de requête 01 Relancer l'option comm. 02 Renvoyer registre de diagnostic 03 Changer le séparateur d'entrée ASCII 04 Forcer le mode écoute seul 05 ... 09 Réservé 10 Réinitialiser les compteurs (et les registres de diagnostic dans 384, 484) 11 Renvoyer le compte de messages bus 12 Renvoyer le compte d'erreurs de comm. bus 13 Renvoyer le compte d'exceptions de bus 14 ... 15 Non pris en charge 16 Donner le compte de NAK esclave 17 Donner le compte d'esclaves occupés 18 Renvoyer le compte de dépassement de car. de bus 19 ... 21 Non pris en charge

Contenu	Description
Adresse de l'automate esclave (MSG_OUT[3])	Entrez l'adresse de l'automate Modbus esclave. La plage d'adresses Modbus est généralement comprise entre 1 et 247. Le code fonction 8 NE PREND PAS en charge le mode diffusion (adresse 0).
Contenu du champ de données de la fonction de diagnostic (MSG_OUT[4])	Vous devez entrer la valeur décimale nécessaire pour la zone de données de la sous-fonction de diagnostic particulière. Pour les sous-fonctions 02, 04, 10, 11, 12, 13, 16, 17 et 18, cette valeur est automatiquement réglée sur zéro. Pour les sous-fonctions 00, 01 et 03, vous devez entrer la valeur voulue dans le champ de données. Pour plus de détails, reportez-vous au document Modicon Modbus Protocol Reference Guide (PI-MBUS-300).
Zone de données de l'automate maître (MSG_OUT[5])	Pour toutes les sous-fonctions, la zone de données de l'automate maître est la destination des données renvoyées par l'esclave. Vous devez indiquer un registre 4x marquant le début de la zone de données dans laquelle les données renvoyées sont placées. Par exemple, pour placer les données dans la zone de données de l'automate maître commençant à (40100), entrez 100 dans ce champ. La sous-fonction 04 NE RENVOIE PAS de réponse. Pour plus de détails, reportez-vous au document Modicon Modbus Protocol Reference Guide (PI-MBUS-300).

Codes fonction Modbus (20, 21)

Pour les messages Modbus, la matrice MSG_OUT doit contenir le tableau de définition Modbus. Le tableau de définition Modbus pour les codes fonction Modbus 20 et 21 comporte six registres. Vous devez définir XMIT_SET.MessageLen sur 6 pour que le XMIT fonctionne correctement. Le tableau de définition Modbus est présenté ci-dessous.

Codes fonction du tableau de définition Modbus (20, 21)

Contenu	Description
Code fonction Modbus (MSG_OUT[1])	Le XMIT prend en charge les codes fonction suivants : 20 = Lire référence générale (6x) 21 = Ecrire référence générale (6x)
Quantité (MSG_OUT[2])	Entrez la quantité de données à écrire ou à lire dans l'automate esclave. Entrez 100 pour lire 100 registres de maintien à partir de l'automate esclave et entrez 32 pour écrire 32 bobines sur un automate esclave. Il existe une taille limite, qui dépend du modèle de l'automate. Reportez-vous à l'annexe A pour obtenir tous les détails sur les limites.
Adresse de l'automate esclave (MSG_OUT[3])	Entrez l'adresse de l'automate Modbus esclave. La plage des adresses Modbus va généralement de 1 à 247. Les codes fonction 20 et 21 ne prennent PAS en charge le mode diffusion (adresse 0).

Contenu	Description
Zone de données de l'automate esclave (MSG_OUT[4])	Pour une commande de lecture, la zone de données de l'automate esclave est la source des données. Pour une commande d'écriture, la zone de données de l'automate esclave est la destination des données. Par exemple, lorsque vous souhaitez lire des registres (de 600300 à 600399) depuis un automate esclave, entrez 300 dans ce champ. Lorsque vous souhaitez écrire des données dans le registre (600100) d'un automate esclave à partir d'un automate maître, entrez 100 dans ce champ. Selon le type de commande Modbus (lecture ou écriture), les zones de données source et cible doivent être conformes à celles du tableau ci-dessous. Le plus petit registre étendu correspond au registre "zéro" (600000). Le plus petit registre de maintien correspond au registre "un" (400001).
Zone de données de l'automate maître (MSG_OUT[5])	Pour une commande de lecture, la zone de données de l'automate maître est la destination des données renvoyées par l'esclave. Pour une commande d'écriture, la zone de données de l'automate maître est la source des données. Par exemple, lorsque vous souhaitez écrire des registres (de 40016 à 40032), situés dans l'automate maître, vers des registres 6x, situés dans un automate esclave, entrez 16 dans le champ. Lorsque vous souhaitez lire des registres 6x (de 600001 à 600100) depuis un automate esclave, et placer les données dans la zone de données de l'automate maître (de 40100 à 40199), entrez 100 dans ce champ. Selon le type de commande Modbus (lecture ou écriture), les zones de données source et cible doivent être conformes à celles du tableau ci-dessous. Le plus petit registre étendu correspond au registre "zéro" (600000). Le plus petit registre de maintien correspond au registre "un" (400001).
Nombre de fichiers (MSG_OUT[6])	Entrez le nombre de fichiers pour les registres 6x à écrire ou à lire. (1 ... 10) selon la taille de la zone de données du registre étendu. 600001 correspond à 60001 fichier 1 et 690001 correspond à 60001 fichier 10, comme considéré par l'éditeur de données de référence.

Zones de données source et cible des codes fonction (20, 21)

Code fonction	Zone de données de l'automate maître	Zone de données de l'automate esclave
20 (Lire référence générale 6x)	4x (cible)	6x (source)
21 (Ecrire référence générale 6x)	4x (source)	6x (cible)

Pour envoyer 20 messages Modbus depuis l'automate, vous devez transférer 20 tables de définition Modbus l'une après l'autre dans MSG_OUT après chaque opération réussie de XMIT. Vous pouvez également programmer 20 blocs XMIT différents, puis les activer l'un après l'autre en utilisant la logique utilisateur.

Bloc de communication XMIT

177

Présentation

Introduction

Ce chapitre décrit l'instruction Bloc de communication XMIT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : Bloc de communication XMIT	1164
Représentation : Bloc de communication XMIT	1165
Description des paramètres : Partie médiane : Table de commande des communications	1167
Description des paramètres : Bloc de communication XMIT	1172
Description des paramètres : Bloc de communication XMIT	1174

Description sommaire : Bloc de communication XMIT

Description de la fonction

L'objectif du bloc de communication XMIT est de recevoir et de transmettre des messages ASCII et des messages maîtres Modbus via les ports de l'automate.

Le bloc d'instructions XMIT ne fonctionnera pas correctement si :

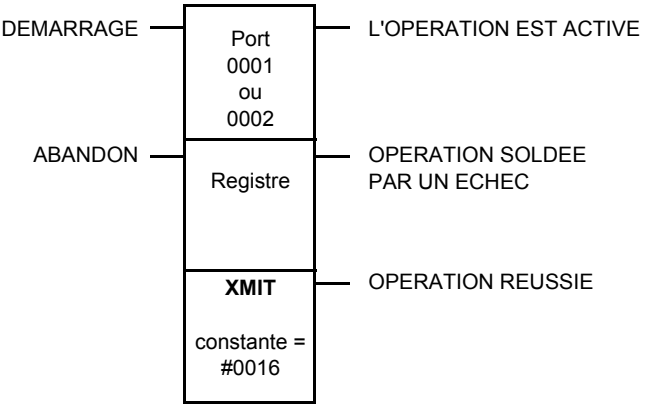
- Les instructions chargeables NSUP et XMIT ne sont pas installées
- L'instruction chargeable NSUP est installée après l'instruction chargeable XMIT
- Les instructions chargeables NSUP et XMIT sont installées dans un automate Quantum utilisant un exécutif obsolète (antérieure à la version 2.10 ou 2.12)

Pour une présentation de l'instruction XMIT, reportez-vous à la section *p. 1156*.

Représentation : Bloc de communication XMIT

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	L'état actif déclenche une opération XMIT. L'instruction START doit rester active jusqu'à ce que l'opération se soit déroulée avec succès ou qu'une erreur soit survenue.
Entrée médiane	0x, 1x	Aucun	L'état actif abandonne toutes les opérations XMIT actives et place de force le port en mode esclave. Le code d'abandon (121) est placé dans le registre d'état de défaut. Tant que cette entrée est à l'état actif, le port reste fermé. **Note :** Pour réinitialiser un défaut XMIT et effacer le contenu du registre de défauts, l'entrée haute doit passer à l'état repos pendant au moins un cycle API.

Paramètres	Référence de mémoire d'état	Type de données	Signification
Port #0001 ou #0002 (partie haute)	4x	INT, UINT, WORD	La partie haute doit contenir une des constantes suivantes : (#0001) pour sélectionner le port numéro 1 de l'automate, ou (#0002) pour sélectionner le port numéro 2 de l'automate. **Note :** La version chargeable accepte les registres 4xxxx en partie haute contrairement à la version intégrée.
Registre (partie médiane)	4x	INT, UINT, WORD	Le registre 4xxxx mémorisé en partie médiane est le premier d'un groupe de seize (16) registres de sortie successifs comprenant le bloc de contrôle, comme indiqué dans la Table de commande des communications. Pour plus d'informations sur cette partie, reportez-vous à la section *p. 1167* dans Description des paramètres de la partie médiane : Bloc de communication XMIT. **Important :** NE PAS modifier l'adresse située dans la partie médiane du bloc XMIT ou la supprimer du bloc tant que le programme est actif. Le port serait bloqué et les communications ne pourraient s'effectuer.
#0016 (partie basse)		INT, UINT, WORD	La partie basse doit contenir une constante égale à (#0016). Celle-ci représente le nombre de registres utilisés par l'instruction XMIT.
Sortie haute	0x	Aucun	Etat actif tant qu'une opération XMIT est en cours. Transmet du courant lorsqu'une opération XMIT est en cours.
Sortie médiane	0x	Aucun	Etat actif lorsque XMIT a détecté une erreur ou a envoyé une commande d'abandon. Transmet du courant lorsque XMIT a détecté une erreur ou qu'une opération XMIT a été abandonnée.
Sortie basse	0x	Aucun	Etat actif pour un seul cycle lorsqu'une opération XMIT a été exécutée avec succès. Transmet du courant lorsqu'une opération XMIT a été exécutée avec succès. Note : L'entrée START doit rester active jusqu'à la désactivation de l'affichage OPERATION REUSSIE.

Description des paramètres : Partie médiane : Table de commande des communications

Table de commande des communications

Cette table représente le premier d'un groupe de 16 registres de sortie successifs formant le bloc de commande.

Registre	Nom	Description	Aucune entrée valide
4xxxx	Numéro de révision	Affiche le numéro de révision courant du bloc XMIT. Ce numéro est chargé automatiquement par le bloc qui écrase tout autre numéro entré dans ce registre.	Lecture seule
4xxxx + 1	Etat d'erreur	Ce champ affiche un code d'erreur généré par le bloc d'état du port XMIT. Pour plus d'informations, reportez-vous *p. 1172* dans la section Description des paramètres : Bloc de communication XMIT.	Lecture seule
4xxxx + 2	Disponible pour l'utilisateur	Le bloc XMIT n'utilise pas ce registre. Il peut cependant être utilisé dans des schémas à contacts comme pointeur. Pour utiliser efficacement le bloc XMIT, placez la valeur pointeur d'une instruction TBLK dans ce registre.	Lire/Ecrire
4xxxx + 3	Vitesse de données	Le bloc XMIT prend en charge les vitesses de données suivantes : 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600 et 19200. Pour configurer une vitesse de données, entrez le nombre décimal correspondant dans ce champ. Lorsqu'une vitesse invalide est saisie, le bloc signale une erreur due à une configuration non valide (code d'erreur 127) dans le registre d'état de défaut (4xxxx + 1).	Lire/Ecrire

Registre	Nom	Description	Aucune entrée valide
4xxxx + 4	Bits de données	Le bloc XMIT prend en charge les bits de données suivants : 7 et 8. Pour configurer une taille de bit de données, entrez le nombre décimal correspondant dans ce registre. **Note :** Les messages Modbus peuvent être envoyés en mode ASCII ou RTU. Le mode ASCII nécessite 7 bits de données, tandis que le mode RTU en nécessite 8. Lorsque vous envoyez un message en caractères ASCII, vous pouvez utiliser 7 ou 8 bits de donnés. Lorsqu'un bit de données invalide est saisi, le bloc signale une erreur due à une configuration non valide (code d'erreur 127) dans le registre d'état de défaut (4xxxx + 1).	Lire/Ecrire
4xxxx + 5	Bits de parité	Le bloc XMIT prend en charge la parité suivante : aucune, impaire et paire. Entrez l'un des nombres décimaux suivants : 0 = aucune parité, 1 = parité impaire ou 2 = parité paire. Lorsqu'une parité invalide est saisie, le bloc signale une erreur due à une configuration non valide (code d'erreur 127) dans le registre d'état de défaut (4xxxx + 1).	Lire/Ecrire
4xxxx + 6	Bits d'arrêt	Le bloc XMIT prend en charge un ou deux bits d'arrêt. Entrez l'un des nombres décimaux suivants : 1 = un bit d'arrêt ou 2 = deux bits d'arrêt. Lorsqu'un bit d'arrêt invalide est saisi, le bloc signale une erreur due à une configuration non valide (code d'erreur 127) dans le registre d'état de défaut (4xxxx + 1).	Lire/Ecrire
4xxxx + 7	Disponible pour l'utilisateur	Le bloc XMIT n'utilise pas ce registre. Il peut cependant être utilisé dans des schémas à contacts comme pointeur. Pour utiliser efficacement le bloc XMIT, placez la valeur pointeur d'une instruction TBLK dans ce registre.	Lire/Ecrire

Registre	Nom	Description	Aucune entrée valide
4xxxx + 8	Mot de commande	(Nombre binaire à 16 chiffres) Le bloc XMIT interprète chaque bit du mot de commande comme fonction d'exécution. Si les bits 7 et 8 sont à 1 simultanément, si plusieurs bits parmi les bits 13, 14, 15 ou 16 sont à 1 simultanément ou si le bit 7 n'est pas à 1 alors que l'un des bits 13, 14, 15 ou 16 l'est, le système génère l'erreur 129. Pour plus d'informations, reportez-vous *p. 1174* dans la section Description des paramètres : Bloc de communication XMIT.	Lire/Ecrire
4xxxx + 9	Mot pointeur de message	(pointeur de message) Les valeurs sont limitées par la plage de registres 4x configurés. La table de messages comprend des éléments des catégories suivantes. • Caractères ASCII Pour les chaînes de caractères ASCII, le pointeur correspond au décalage vers le premier registre de la chaîne de caractères ASCII. Chaque registre peut contenir jusqu'à deux caractères ASCII. Chaque chaîne ASCII peut se composer de 1024 caractères au maximum. Par exemple, lorsque vous souhaitez envoyer dix messages ASCII depuis l'automate, vous devez programmer 10 chaînes de caractères ASCII dans des registres 4xxxx de l'automate puis, avec des schémas à contacts, définir le pointeur sur le début de chaque message après chaque exécution réussie de l'instruction XMIT. • Codes de fonctions Modbus Pour plus d'informations, reportez-vous à la section *p. 1157* Saisissez un pointeur qui indique le début de la table de messages.	Lire/Ecrire

Registre	Nom	Description	Aucune entrée valide
4xxxx + 10	Longueur du message	(0 à 512) Saisissez la longueur du message courant. Lorsque le bloc XMIT envoie des messages Modbus pour les codes de fonction 01, 02, 03, 04, 05, 06, 08, 15 et 16, la longueur du message est automatiquement mise sur cinq. Lorsqu'il reçoit une entrée au format ASCII terminé, la longueur du message doit être mise sur cinq. Sinon, une erreur survient. Lorsque le bloc XMIT envoie des messages Modbus pour les codes de fonction 20 et 21, la longueur du message est automatiquement mise sur six. Lorsqu'il envoie des messages ASCII, cette longueur peut être comprise entre 1 et 1024 caractères ASCII par message.	Lire/Ecrire
4xxxx + 11	Dépassement du délai de réponse (ms)	(0 à 65535 millisecondes) Saisissez une durée en millisecondes (ms) pour déterminer le temps d'attente, par le bloc XMIT, d'un message de réponse valide provenant d'un appareil esclave (automate, modem, etc.). De plus, la durée s'applique aux transmissions ASCII et aux commandes de contrôle de flux. Lorsque la réponse n'est pas entièrement structurée dans le délai imparti, le bloc XMIT génère une erreur. La durée peut être comprise entre 0 et 65535 ms. Le dépassement de délai commence après l'envoi du dernier caractère du message.	Lire/Ecrire
4xxxx + 12	Limite de nouvelles tentatives	(0 à 65535 millisecondes) Saisissez la quantité de nouvelles tentatives d'envoi d'un message effectuées par le bloc XMIT avant réception d'une réponse valide d'un appareil esclave (automate, modem, etc.). Lorsque la réponse n'est pas entièrement structurée dans le délai imparti, le bloc XMIT génère une erreur et un code d'erreur. Le nombre de nouvelles tentatives peut être compris entre 0 et 65535. Ce champ est utilisé conjointement avec le délai de réponse (4xxxx + 11).	Lire/Ecrire

Registre	Nom	Description	Aucune entrée valide
4xxxx + 13	Retard du début de transmission (ms)	(0 à 65535 millisecondes) Lorsque le contrôle RTS/CTS est activé, saisissez la durée en millisecondes (ms) déterminant le temps d'attente du bloc XMIT entre la réception du message CTS et la transmission d'un message par le port de l'automate n° 1. De plus, vous pouvez utiliser ce registre même lorsque le contrôle RTC/CTS n'est PAS activé. Dans ce cas, la durée entrée détermine le temps d'attente du bloc XMIT avant l'envoi d'un message à partir du port de l'automate n° 1. Vous pouvez l'utiliser comme un temporisateur avant message. Les valeurs peuvent être comprises entre 0 et 65535 ms.	Lire/Ecrire
4xxxx + 14	Retard de la fin de transmission (ms)	(0 à 65535 millisecondes) Pour déterminer la durée pendant laquelle le bloc XMIT garde un état logique vrai RTS après l'envoi du message à partir du port n° 1 de l'automate, entrez la valeur du délai en millisecondes (ms) lorsque le contrôle RTS/CTS est activé. A l'expiration de ce délai, le bloc XMIT met fin à l'état logique vrai. De plus, vous pouvez utiliser ce registre même lorsque le contrôle RTS/CTS n'est PAS activé. Dans ce cas, la valeur entrée détermine le temps d'attente du bloc XMIT après l'envoi d'un message à partir du port n° 1 de l'automate. Vous pouvez l'utiliser comme un temporisateur après message. Les valeurs peuvent être comprises entre 0 et 65535 ms.	Lire/Ecrire
4xxxx + 15	Nouvelles tentatives courantes	La valeur affichée indique le nombre courant de tentatives effectuées par le bloc XMIT.	Lecture seule

Description des paramètres : Bloc de communication XMIT

Table d'état des erreurs

La table ci-dessous dresse la liste des codes d'erreurs générés par le bloc d'état du port XMIT (4x + 1).

Code d'erreurs	Description de l'erreur
1	Exception Modbus, fonction non valide
2	Exception Modbus, adresse de données non valide
3	Exception Modbus, valeur de données non valide
4	Exception Modbus, échec du périphérique esclave
5	Exception Modbus, accusé de réception
6	Exception Modbus, périphérique esclave occupé
7	Exception Modbus, accusé de réception négatif
8	Exception Modbus, erreur de parité de la mémoire
9 à 99	Réservé
100	La zone de données automate esclave ne peut être égale à zéro
101	La zone de données automate maître ne peut être égale à zéro
102	Bobine (0x) non configurée
103	Registre de sortie (4xxxx) non configuré
104	La longueur des données ne peut être égale à zéro
105	Le pointeur vers la table de messages ne peut être égal à zéro
106	Le pointeur vers la table de messages est en dehors des limites des registres de sortie (4xxxx) configurés
107	Timeout de transmission du message Cette erreur est générée lorsque l'émetteur-récepteur asynchrone universel ne parvient pas à terminer une transmission en 10 secondes maximum. Elle contourne le compteur de tentatives et active la sortie d'erreur sur la première erreur.
108	Erreur non définie
109	Le modem a renvoyé le message ERREUR
110	Le modem a renvoyé le message PAS DE PORTEUSE
111	Le modem a renvoyé le message PAS DE TONALITE
112	Le modem a renvoyé le message OCCUPE
113	Total de contrôle LRC invalide en provenance de l'automate esclave
114	Total de contrôle CRC invalide en provenance de l'automate esclave
115	Code de fonction Modbus invalide
116	Dépassement de délai de message de réponse Modbus
117	Timeout de réponse du modem

Code d'erreurs	Description de l'erreur
118	XMIT n'a pas obtenu d'accès au port de communication de l'automate (port 1 ou 2)
119	XMIT ne peut libérer le récepteur de port de l'automate
120	XMIT UART automate non reconnue
121	L'utilisateur a envoyé une commande d'abandon
122	Partie haute de XMIT différente de 0, 1 ou 2
123	Partie basse de XMIT différente de 7, 8 ou 16
124	Etat interne non défini
125	Mode de diffusion non autorisé avec ce code de fonction Modbus
126	L'ETCD n'a pas activé le signal CTS
127	Configuration non valide (vitesse de données, bits de données, parité ou bits d'arrêt)
128	Réponse inattendue en provenance de l'esclave Modbus
129	Paramétrage du mot de commande non valide
130	Mot de commande modifié en cours d'activité
131	Nombre de caractères invalide
132	Bloc de registre invalide
133	Erreur de débordement de l'entrée FIFO ASCII
134	Nombre de caractères de début ou de fin invalide

Description des paramètres : Bloc de communication XMIT

Table de fonctions de communication du mot de commande

Cette table décrit les fonctions exécutées lorsque le bloc XMIT interprète chaque bit du mot de commande.

Fonction du mot de commande (4x + 8)	Bits du mot de commande qui doivent être mis sur 1	Bits du mot de commande qui doivent être mis sur 0
Entrée ASCII terminée (bit 5 = 1)	2,3,9,10,11,12	6,7,8,13,14,15,16
Entrée ASCII simple (bit 6 = 1)	2,3,9,10,11,12	5,7,8,13,14,15,16
Sortie ASCII simple (bit 7 = 1)	2,3,9,10,11,12	5,6,8,13,14,15,16
Sortie de modem (bit 7 = 1)	2,3,13,14,15,16	5,6,8,9,10,11,12 (et SEULEMENT un des bits suivants est mis sur 1 : 13,14,15 ou 16, les trois autres bits devant être mis sur 0)
Sortie de messagerie maître Modbus (Bit 8 = 1)	2,3	5,6,7,9,10,11,12,13,14, 15,16
Validation de l'entrée FIFO de réception ASCII uniquement (bit 9 = 1)	2,3,10,11,12	5,6,7,8,13,14,15,16

Bloc d'état du port XMIT

178

Présentation

Introduction

Ce chapitre décrit l'instruction Bloc d'état du port XMIT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : Bloc d'état du port XMIT	1176
Représentation : Bloc d'état du port XMIT	1177
Description des paramètres : Partie médiane : Bloc de conversion XMIT	1179

Description sommaire : Bloc d'état du port XMIT

Description de la fonction

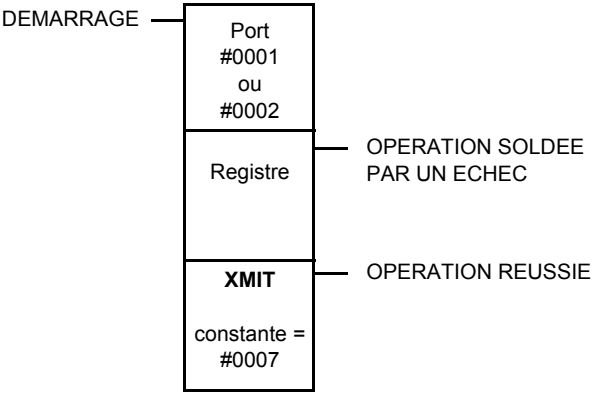
Le bloc d'état du port XMIT indique l'état courant du port, l'activité de l'esclave Modbus, l'entrée FIFO ASCII et l'information sur le contrôle de flux pouvant être utilisée dans des schémas à contacts pour certaines applications. Le bloc d'état du port XMIT est totalement passif. Il n'occupe pas, ne libère pas et ne contrôle pas le port de l'automate.

Pour une présentation de l'instruction XMIT, reportez-vous au *p. 1156*.

Représentation : Bloc d'état du port XMIT

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	L'état actif déclenche une opération XMIT. L'instruction doit rester active jusqu'à ce que la commande se soit déroulée avec succès ou qu'une erreur soit survenue.
Port #0001 ou #0002 (partie haute)	4x	INT, UINT WORD	Doit contenir une des constantes suivantes pour (#0001) sélectionner le port automate #1, ou (#0002) pour sélectionner le port automate #2. **Note :** La version chargeable ACCEPTE les registres 4xxxx dans la partie haute, tandis que la version intégrée ne les accepte PAS.
Registre (partie médiane)	4x	INT, UINT, WORD	Le registre 4xxxx mémorisé en partie médiane est le premier d'un groupe de sept registres de sortie successifs comprenant le bloc d'affichage de l'état du port, comme indiqué dans *p. 1179* de la section Description des paramètres de la partie médiane : Bloc de conversion XMIT. **Important :** NE PAS modifier l'adresse située dans la partie médiane du bloc XMIT ou la supprimer du bloc tant que celui-ci est actif. Le port serait bloqué et les communications ne pourraient s'effectuer.
Constante = #0007 (partie basse)		INT, UINT, WORD	Doit contenir une constante égale à (#0007). Celle-ci représente le nombre de registres utilisés par l'instruction d'état du port XMIT.
Sortie médiane	0x	Aucun	Etat actif lorsque XMIT a détecté une erreur ou a envoyé une commande d'abandon.
Sortie basse	0x	Aucun	Etat actif lorsqu'une commande XMIT a été exécutée avec succès.

Description des paramètres : Partie médiane : Bloc de conversion XMIT

Présentation

Cette section présente de manière détaillée des informations relatives à la partie médiane. La section se compose de six (6) unités.

- Table d'affichage d'état des ports
- Table de génération de codes d'erreur
- Table de génération d'états
- Table d'appartenance des ports
- Table d'état de l'entrée FIFO
- Table de longueur de l'entrée FIFO

Table d'affichage d'état des ports

Cette table représente le premier d'un groupe de sept (7) registres de sortie successifs comprenant le bloc d'état du port.

Registre	Nom	Description	Aucune entrée valide
4xxxx	Numéro de révision	Affiche le numéro de révision courant du bloc XMIT. Ce numéro est chargé automatiquement par le bloc qui écrase tout autre numéro mémorisé dans ce registre.	Lecture seule
4xxxx +1	Etat de défaut	Ce champ affiche un code d'erreur généré par le bloc d'état du port XMIT. Pour plus d'informations, reportez-vous à la Table de génération de codes d'erreur ci-dessous.	Lecture seule
4xxxx +2	Etat du login esclave/Etat actif du port esclave	Ce registre affiche l'état de deux articles générés par le block d'état du port XMIT. Il s'agit de l'état du login esclave et de l'état actif du port esclave. Les schémas à contacts peuvent éventuellement utiliser cette information pour réduire ou éviter les collisions dans un réseau Modbus multimaître. Pour plus d'informations, reportez-vous à la Table de génération d'états ci-dessous.	Lecture seule

Registre	Nom	Description	Aucune entrée valide
4xxxx +3	Compteur de transactions esclaves	Ce registre affiche le nombre de transactions esclaves générées par le block d'état du port XMIT. Le compteur incrémente chaque fois que le port esclave de l'automate Modbus reçoit une nouvelle commande du maître Modbus. Les schémas à contacts peuvent éventuellement utiliser cette information pour réduire ou éviter les collisions dans un réseau Modbus multimaître.	Lecture seule
4xxxx +4	Etat du port	Ce registre affiche l'appartenance du port ainsi que son état. Le registre est généré par le bloc d'état du port XMIT. Pour plus d'informations, reportez-vous à la Table d'appartenance des ports ci-dessous.	Lecture seule
4xxxx +5	Bits d'état de l'entrée FIFO	Le registre affiche l'état de sept articles liés à l'entrée FIFO. Le registre est généré par le bloc d'état du port XMIT. Pour plus d'informations, reportez-vous à la Table de l'entrée FIFO ci-dessous.	Lecture seule
4xxxx +6	Longueur de l'entrée FIFO	Ce registre affiche le nombre courant de caractères présents dans l'entrée FIFO ASCII. Le registre peut contenir d'autres valeurs selon l'état de l'entrée FIFO et selon la longueur (vide ou dépassement). Le registre est généré par le bloc d'état du port XMIT. Pour plus d'informations, reportez-vous à la Table de longueur de l'entrée FIFO ci-dessous.	Lecture seule

Table de génération de codes d'erreur

Cette table décrit les codes d'erreur générés par le bloc d'état du port XMIT dans le registre (4x + 1).

Code d'erreur	Description de l'erreur
118	XMIT n'a pas obtenu d'accès au port de communication de l'automate (port 1 ou 2).
122	Partie haute de XMIT différente de 0, 1 ou 2.
123	Partie basse de XMIT différente de 7, 8 ou 16.

Table de génération d'états

Cette table décrit l'état du login esclave et l'état actif du port esclave généré par le bloc d'état du port XMIT pour le registre (4x + 2).

(4x + 2, octet de poids fort) **Etat du login esclave**	**(4x + 2, octet de poids faible)** **Etat actif du port esclave**
Oui : Lorsqu'un périphérique de programmation est connecté au port esclave de l'automate.	**Oui** : Lorsque le port observé appartient à l'automate et est en train de recevoir une commande Modbus ou de transmettre une réponse Modbus.
Non : Lorsqu'un périphérique de programmation n'est pas connecté au port esclave de l'automate. **Note :** Un maître Modbus peut envoyer des commandes, mais ne peut être connecté.	**Non** : Lorsque le port observé n'appartient pas à l'automate et n'est pas en train de recevoir une commande Modbus ou de transmettre une réponse Modbus.

Table d'appartenance des ports

Cette table décrit l'appartenance et l'état des ports pour le registre (4x + 4).

Propriétaire du port	**Etat actif**	**Valeur**
Automate	Esclave Modbus de l'automate	0
XMIT	Modem à numérotation à tonalité	1
XMIT	Déconnexion du modem	2
XMIT	Messagerie Modbus	3
XMIT	Sortie ASCII simple	4
XMIT	Modem à numérotation par impulsions	5
XMIT	Initialisation du modem	6
XMIT	Entrée ASCII simple	7
XMIT	Entrée au format ASCII terminé	8
XMIT	L'entrée ASCII FIFO est active, mais aucune fonction XMIT ne l'est.	9

Table d'état de l'entrée FIFO

Cette table décrit les bits d'état liés à l'entrée FIFO pour le registre (4x + 5).

Numéro de bit	Définition	Oui / 1	Non / 0
1 à 3	Réservé		
4	Propriétaire du port ...	XMIT	Automate
5 à 7	Réservé		
8	Transmission de sortie ASCII	Bloqué par récepteur	Non bloqué par récepteur
9	Entrée ASCII reçue ...	Nouveau caractère	Pas de nouveau caractère
10	L'entrée FIFO ASCII est ...	Vide	Non vide
11	L'entrée FIFO ASCII est ...	Dépassement (erreur)	Pas de dépassement (erreur)
12	L'entrée FIFO ASCII est ...	Activée	Désactivée
13 à 15	Réservé		
16	Réception d'entrée ASCII ...	Emetteur bloqué XMIT	Emetteur non bloqué XMIT

Table de longueur de l'entrée FIFO

Cette table décrit le nombre courant de caractères présents dans l'entrée FIFO ASCII pour le registre (4x + 6).

LORSQUE l'entrée FIFO	ALORS Longueur
= OFF	= 0
= ON et Vide	= 0
= ON et Dépassement	= 512

Bloc de conversion XMIT

179

Présentation

Introduction

Ce chapitre décrit l'instruction Bloc de conversion XMIT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire : Bloc de conversion XMIT	1184
Représentation : Bloc de conversion XMIT	1185
Description des paramètres : Bloc de conversion XMIT	1187

Description sommaire : Bloc de conversion XMIT

Description de la fonction

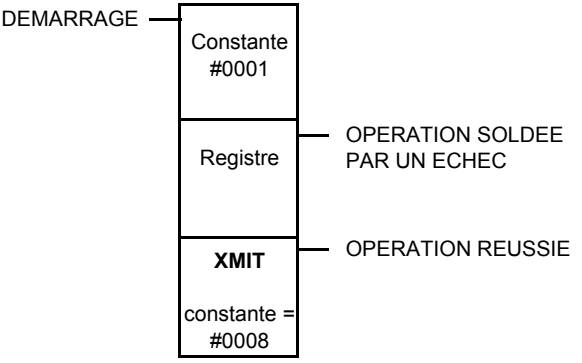
L'objectif du bloc de conversion XMIT est le traitement et la conversion de données en d'autres formes utilisables selon les besoins de vos applications. Le bloc de conversion exécute onze (11) fonctions ou options distinctes. Certaines fonctions incluent la conversion d'ASCII en binaire, d'entiers en ASCII, la permutation d'octets, la recherche dans des chaînes ASCII, etc. Ce bloc autorise les conversions internes au moyen de blocs source 4xxxx et de blocs cible 4xxxx.

Pour une présentation de l'instruction XMIT, reportez-vous *p. 1156*.

Représentation : Bloc de conversion XMIT

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	L'état actif déclenche une opération XMIT. L'instruction doit rester active jusqu'à ce que la commande se soit déroulée avec succès ou qu'une erreur soit survenue. **Note :** Pour réinitialiser un défaut XMIT et effacer le contenu du registre de défauts, l'entrée haute doit passer à l'état repos pendant au moins un cycle API.
Constante #0001 (partie haute)	4x	INT, UINT WORD	La partie haute doit contenir une constante (#0000) car les conversions ne concernent pas le port de l'automate. La version chargeable accepte les registres 4xxxx dans la partie haute, tandis que la version intégrée ne les accepte pas.

Paramètres	Référence de mémoire d'état	Type de données	Signification
Registre (partie médiane)	4x	INT, UINT, WORD	Le registre 4xxxx mémorisé en partie médiane est le premier d'un groupe de huit (8) registres de sortie successifs comprenant le bloc de contrôle, comme indiqué dans la *p. 1187* de la section Description des paramètres : Bloc de conversion XMIT **Important :** NE PAS modifier l'adresse située dans la partie médiane du bloc XMIT ou la supprimer du programme tant que le bloc est actif. Le port serait bloqué et les communications ne pourraient s'effectuer.
Constante = #0008 (partie basse)		INT, UINT, WORD	La partie basse doit contenir une constante égale à (#0008). Celle-ci représente le nombre de registres utilisés par l'instruction de conversion XMIT.
Sortie médiane	0x	Aucun	Etat actif lorsque XMIT a détecté une erreur ou envoyé une commande d'abandon.
Sortie basse	0x	Aucun	Etat actif lorsqu'une commande XMIT a été exécutée avec succès.

Description des paramètres : Bloc de conversion XMIT

Présentation

Cette section présente de manière détaillée des informations relatives à la partie médiane. La section se compose de quatre (4) unités.

- Table de contrôle du bloc de conversion
- Table de génération de codes d'erreur
- Table des bits de contrôle de la conversion de données
- Table de code opérande de conversion de données

Table de contrôle du bloc de conversion

Cette table représente le premier d'un groupe de huit (8) registres de sortie successifs comprenant le bloc d'état du port.

Registre	Nom	Description	Aucune entrée valide
4xxxx	Numéro de révision XMIT	Affiche le numéro de révision courant du bloc XMIT. Ce numéro est chargé automatiquement par le bloc qui écrase tout autre numéro mémorisé dans ce registre.	Lecture seule
4xxxx + 1	Etat d'erreur	Ce champ affiche un code d'erreur généré par le bloc d'état du port XMIT. Pour plus d'informations, reportez-vous à la Table de génération de codes d'erreur ci-dessous.	Lecture seule
4xxxx + 2	Disponible pour l'utilisateur	0 (Peut être utilisé comme pointeur pour des instructions telles que TBLK.) Le bloc de conversion XMIT n'utilise pas ce registre. Il peut cependant être utilisé dans des schémas à contacts comme pointeur. Pour utiliser efficacement le bloc XMIT, placez la valeur pointeur d'une instruction TBLK dans ce registre.	Lire/Ecrire
4xxxx + 3	Bits de contrôle de la conversion de données	Ce mot de 16 bits fait référence au mot (4xxxx + 3) de conversion de données. Ces bits offrent des options de contrôle supplémentaires selon la conversion choisie parmi les onze disponibles. Pour plus d'informations, reportez-vous à la Table des bits de contrôle de la conversion de données ci-dessous.	Lire/Ecrire

Registre	Nom	Description	Aucune entrée valide
4xxxx + 4	Code opérande de conversion de données	Sélectionnez le type de conversion que vous souhaitez effectuer dans la liste des onze options de la Table de codes opérandes de conversion de données ci-dessous. Après avoir choisi le type de conversion, reportez-vous aux bits de contrôle (4xxxx + 4) de conversion de données pour des options de contrôle supplémentaires liées au type de conversion sélectionné.	Lire/Ecrire
4xxxx + 5	Registre source	Entrez le registre 4xxxx désiré. C'est le premier registre lu du bloc source. N'oubliez pas de préciser l'octet (de poids fort ou de poids faible) de début de la lecture.	Lire/Ecrire
4xxxx + 6	Registre cible	Entrez le registre 4xxxx désiré. C'est le premier registre lu du bloc source. N'oubliez pas de préciser l'octet (de poids fort ou de poids faible) de début de la lecture. La sélection située en regard de ce registre dans le zoom DX est la même que le bit 16 dans (4xxxx + 3).	Lire/Ecrire
4xxxx + 7	Comptage des caractères d'une chaîne ASCII	Entrez la zone de recherche Ce registre définit la zone de recherche. Lorsque la source (bit13) ou la cible d'avance automatique (bit14) sont à l'état actif et qu'aucun caractère ASCII n'est détecté, le bloc ajuste automatiquement le comptage de caractères.	Lire/Ecrire

Table de génération de codes d'erreur

Cette table décrit les codes d'erreur générés par le bloc de conversion XMIT dans le registre (4x + 1).

Code d'erreur	Description de l'erreur
122	Partie haute de XMIT est différente de 0, 1 ou 2
123	Partie basse de XMIT différente de 7, 8 ou 16
131	Nombre de caractères invalide
135	Bloc de registre cible invalide
136	Bloc de registre source invalide
137	Pas de numéro ASCII
138	Existence de caractères à signes multiples
139	Dépassement numérique détecté
140	Erreur de correspondance de chaîne

Code d'erreur	Description de l'erreur
141	Chaîne non trouvée
142	Détection de vérification d'erreur invalide
143	Code opérande de conversion invalide

Table des bits de contrôle de la conversion de données

Cette table décrit les options de contrôle disponibles selon la conversion sélectionnée dans le registre (4x + 3).

Numéro de bit	Définition	1 =	0 =
2	Valeur de départ CRC 16	0x0000	0xFFFF
3	Type de contrôle d'erreur	LRC 8	CRC 16
4	Contrôle d'erreur	Valider	Ajouter
7	Conversion de la casse	Majuscule - minuscule	Minuscule - majuscule
8	Sensibilité à la casse	Non	Oui
9	Format des bits de tête	Zéros	Blancs
10	Format de sortie	Fixe	Variable
11	Type de conversion	Non signé	Signé
12	Mot de conversion	32 bits	16 bits
13	Pointeur source d'avance automatique (désigne le caractère suivant le dernier caractère purgé)	Oui	Non
14	Pointeur cible d'avance automatique (désigne le caractère suivant le dernier caractère purgé)	Oui	Non
15	Commencer la lecture des données ASCII à la source commençant par ...	Octet de poids faible	Octet de poids fort (normal)
16	Commencer la sauvegarde des données ASCII à la cible commençant par ...	Octet de poids faible	Octet de poids fort (normal)

Table de code opérande de conversion de données

Cette table décrit les onze (11) fonctions ou options pour l'exécution des conversions utilisant les codes opérandes de conversion de données dans le registre (4x + 4).

Code opérande	Action	Type de données (bloc 4xxxx)
Code opérande non valide	Affichage lors de la détection d'un code opérande non valide	Sans objet
(1 hexadécimal) Chaîne de caractères décimaux ASCII reçue	Conversion	Entier binaire signé ou non signé 16 ou 32 bits
(2 hexadécimal) Chaîne de caractères hexadécimaux ASCII reçue	Conversion	Entier binaire non signé 16 ou 32 bits
(3 hexadécimal) Chaîne de caractères hexadécimaux ASCII reçue	Conversion	Tableau non signé d'entiers binaires 16 bits
(4 hexadécimal) Entier signé ou non signé 16 ou 32 bits	Conversion	Chaîne de caractères décimaux ASCII pour transmission
(5 hexadécimal) Entier binaire non signé 16 ou 32 bits	Conversion	Chaîne de caractères hexadécimaux ASCII pour transmission
(6 hexadécimal) Tableau non signé d'entiers 16 bits	Conversion	Chaîne de caractères hexadécimaux ASCII pour transmission
(7 hexadécimal) Octets de poids fort et de poids faible du bloc de registre source ASCII enregistré.	Permutation vers	Bloc de registre cible ASCII
(8 hexadécimal) Chaîne ASCII d'un bloc de registre source	Copie dans	Bloc de registre cible ASCII avec ou sans conversion de la casse
(9 hexadécimal) Bloc de registre source ASCII	Comparaison avec	Chaîne de données ASCII définie dans le bloc de registre cible sensible ou non à la casse
(10 hexadécimal) Bloc de registre source ASCII	Recherche	Chaîne ASCII définie dans le bloc cible sensible ou non à la casse
(11 hexadécimal) Contrôle d'erreur 8 bits LRC ou 16 bits CRC	Validation ou Ajout à	Chaîne de données ASCII dans un bloc de registre source

XMRD : Lecture de mémoire étendue

180

Présentation

Introduction

Ce chapitre décrit l'instruction XMRD.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1192
Représentation : XMRD : Lecture de mémoire étendue	1193
Description des paramètres	1194

Description sommaire

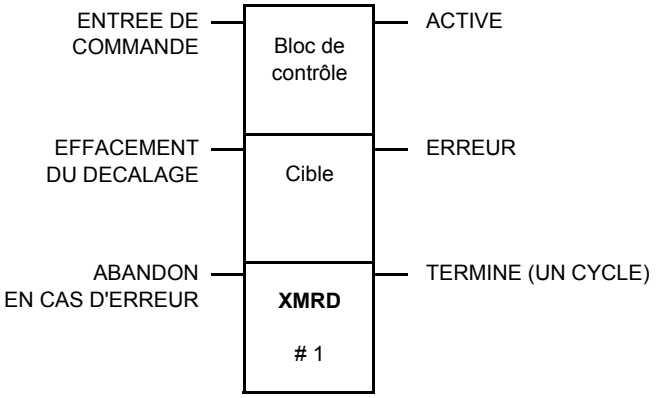
Description de la fonction

L'instruction XMRD est utilisée pour copier une table de registres de mémoire étendue 6x vers une table de registres de sortie 4x en mémoire d'état.

Représentation : XMRD : Lecture de mémoire étendue

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = active l'opération lecture
Entrée médiane	0x, 1x	Aucun	Etat repos = remet le décalage à 0 Etat actif = n'efface pas le décalage
Entrée basse	0x, 1x	Aucun	Etat repos = abandon si erreur Etat actif = n'abandonne pas si erreur
Bloc de contrôle (voir *p. 1194*) (partie haute)	4x	INT, UINT, WORD	Premier de six registres de sortie successifs de la mémoire étendue Pour plus d'informations, reportez-vous à la section *p. 1194*.
Cible (partie médiane)	4x	INT, UINT, WORD	Le premier registre de sortie 4x dans une table de registres recevant les données transmises depuis les registres de stockage de la mémoire étendue 6x.
1 (partie basse)		INT, UINT	Contient la valeur constante 1, ne peut être modifiée.
Sortie haute	0x	Aucun	Transfert lecture actif
Sortie médiane	0x	Aucun	Détection d'une condition d'erreur
Sortie basse	0x	Aucun	Etat actif = opération terminée

Description des paramètres

Bloc de contrôle (partie haute)

Le registre 4x saisi en partie haute est le premier de six registres de sortie successifs du bloc de contrôle de la mémoire étendue.

Référence	Nom du registre	Description
Affiché	Mot d'état	Contient les informations du diagnostic de la mémoire étendue (voir la section *p. 1195*)
Premier implicite	Numéro de fichier	Indique le fichier de mémoire étendue actuellement utilisé (compris entre 1 et 10)
Deuxième implicite	Adresse de départ	Indique quel registre de stockage 6x du fichier actuel est l'adresse de départ ; 0 = 60 000, 9 999 = 69 999
Troisième implicite	Compte	Indique le nombre de registres à lire ou écrire dans un cycle lorsque le bloc fonction correspondant est activé, compris entre 0 et 9999, sans dépasser le nombre indiqué dans Nombre maximum de registres (cinquième implicite)
Quatrième implicite	Décalage	Donne le nombre total actualisé de registres transférés jusque–là
Cinquième implicite	Nombre maximum de registres	Indique le nombre maximum de registres pouvant être transférés lorsque le bloc fonction est activé (compris entre 0 et 9999)

Si vous êtes en mode multicycle, ces six registres doivent être spécifiques à ce bloc fonction.

Mot d'état du bloc de contrôle

Mot d'état du bloc de contrôle

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = erreur diagnostic à la mise sous tension
2	1 = erreur de parité en mémoire étendue
3	1 = mémoire étendue inexistante
4	0 = transfert non activé 1 = occupé
5	0 = transfert en cours 1 = transfert terminé
6	1 = limite de fichier franchie
7	1 = paramètre de décalage trop grand
8 - 9	Inutilisé
10	1 = mémoire d'état inexistante
11	Inutilisé
12	1 = erreur de paramètre du nombre maximum de registres
13	1 = erreur de paramètre de décalage
14	1 = erreur de paramètre de compte
15	1 = erreur de paramétrage d'adresse de départ
16	1 = erreur de paramètre du numéro de fichier

XMRD : Ecriture en mémoire étendue

181

Présentation

Introduction

Ce chapitre décrit l'instruction XMWT.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description sommaire	1198
Représentation : XMWT : Ecriture en mémoire étendue	1199
Description des paramètres	1200

Description sommaire

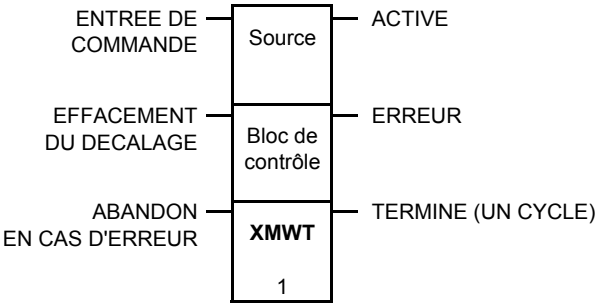
Description de la fonction

L'instruction XMWT est utilisée pour écrire des données d'un bloc de registres d'entrée ou de sortie de la mémoire d'état dans un bloc de registres 6x d'un fichier de mémoire étendue.

Représentation : XMWT : Ecriture en mémoire étendue

Symbole

Représentation de l'instruction



Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Etat actif = active l'opération d'écriture
Entrée médiane	0x, 1x	Aucun	Etat repos = remet le décalage à 0 Etat actif = n'efface pas le décalage
Entrée basse	0x, 1x	Aucun	Etat repos = abandon si erreur Etat actif = n'abandonne pas si erreur
Source (partie haute)	3x, 4x	INT, UINT, WORD	Le premier registre 3x ou 4x d'un bloc de registres source successifs, c'est-à–dire de registres d'entrée ou de sortie dont le contenu sera écrit dans des registres 6x de mémoire étendue.
Bloc de contrôle (voir *p. 1200*) (partie médiane)	4x	INT, UINT, WORD	Premier de six registres de sortie successifs de la mémoire étendue Pour plus d'informations, reportez-vous à la section *p. 1200*.
1 (partie basse)		INT, UINT	Contient la valeur constante 1, ne peut être modifiée.
Sortie haute	0x	Aucun	Procédure d'écriture active
Sortie médiane	0x	Aucun	Détection d'une condition d'erreur
Sortie basse	0x	Aucun	Etat actif = opération terminée

Description des paramètres

Bloc de contrôle (partie médiane)

Le registre 4x mémorisé en partie médiane est le premier de six registres de sortie successifs du bloc de contrôle de la mémoire étendue.

Référence	Nom du registre	Description
Affiché	Mot d'état	Contient les informations du diagnostic de la mémoire étendue (voir la section *p. 1201*)
Premier implicite	Numéro de fichier	Indique le fichier de mémoire étendue actuellement utilisé (compris entre 1 et 10)
Deuxième implicite	Adresse de départ	Indique quel registre de stockage 6x du fichier actuel est l'adresse de départ ; 0 = 60 000, 9 999 = 69 999
Troisième implicite	Compte	Indique le nombre de registres à lire ou écrire dans un cycle lorsque le bloc fonction correspondant est activé, compris entre 0 et 9999, sans dépasser le nombre indiqué dans Nombre maximum de registres (cinquième implicite)
Quatrième implicite	Décalage	Donne le nombre total actualisé de registres transférés jusque–là
Cinquième implicite	Nombre maximum de registres	Indique le nombre maximum de registres pouvant être transférés lorsque le bloc fonction est activé (compris entre 0 et 9999)

Si vous êtes en mode multicycle, ces six registres doivent être spécifiques à ce bloc fonction.

Mot d'état du bloc de contrôle

Mot d'état du bloc de contrôle

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Bit	Fonction
1	1 = erreur diagnostic à la mise sous tension
2	1 = erreur de parité en mémoire étendue
3	1 = mémoire étendue inexistante
4	0 = transfert non activé 1 = occupé
5	0 = transfert en cours 1 = transfert terminé
6	1 = limite de fichier franchie
7	1 = paramètre de décalage trop grand
8 à 9	Inutilisé
10	1 = mémoire d'état inexistante
11	Inutilisé
12	1 = erreur de paramètre du nombre maximum de registres
13	1 = erreur de paramètre de décalage
14	1 = erreur de paramètre de compte
15	1 = erreur de paramétrage d'adresse de départ
16	1 = erreur de paramètre du numéro de fichier

XOR : OU exclusif

182

Présentation

Introduction

Ce chapitre décrit l'instruction XOR.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

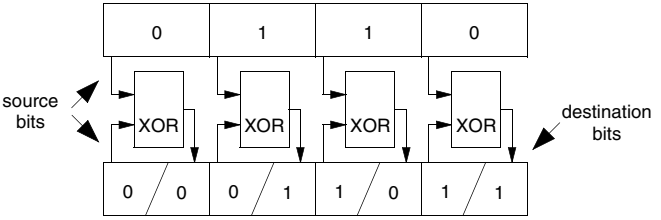
Sujet	Page
Description sommaire	1204
Représentation : XOR : OU exclusif booléen	1205
Description des paramètres	1207

Description sommaire

Description de la fonction

L'instruction XOR effectue une opération OU exclusive booléenne sur les configurations binaires des matrices source et cible.

La configuration binaire obtenue par XOR est ensuite placée dans la matrice cible, écrasant ainsi le contenu précédent :



⚠ AVERTISSEMENT

XOR écrasera toute bobine invalidée dans la matrice cible sans pour autant la valider.

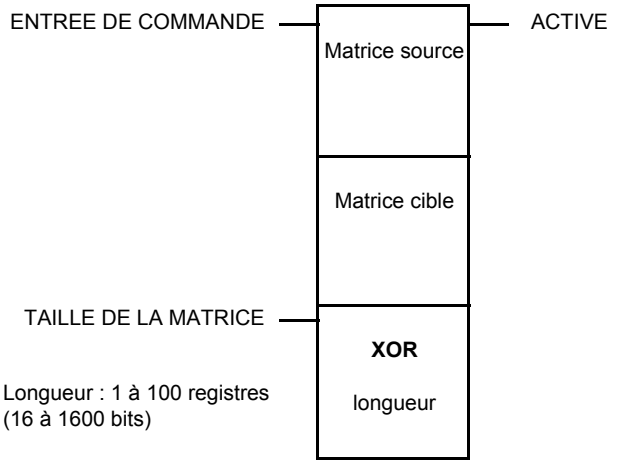
Ceci peut provoquer des dommages corporels si une bobine a invalidé une opération pour des travaux d'entretien ou de réparation puisque l'état de la bobine peut changer suite à l'opération XOR.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

Représentation : XOR : OU exclusif booléen

Symbole

Représentation de l'instruction



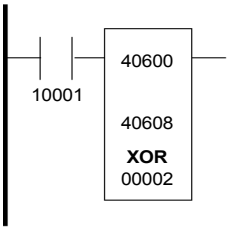
Description des paramètres

Description des paramètres d'instruction

Paramètres	Référence de mémoire d'état	Type de données	Signification
Entrée haute	0x, 1x	Aucun	Déclenche XOR
Matrice source (partie haute)	0x, 1x, 3x, 4x	BOOL, WORD	Première référence de la *matrice source*
Matrice cible (partie médiane)	0x, 4x	BOOL, WORD	Première référence de la *matrice cible*
Longueur (partie basse)		INT, UINT	Longueur de la matrice ; comprise entre 1 et 100 registres
Sortie haute	0x	Aucun	Donne une image de l'état de l'entrée haute

Exemple XOR

Lorsque le contact 10001 transmet du courant, la *matrice source* formée par la configuration binaire dans les registres 40600 et 40601 est liée par XOR à la *matrice cible* formée par la configuration binaire dans les registres 40608 et 40609.



Matrice source
40600 = 1111111100000000 40601 = 1111111100000000

Matrice cible **d'origine**
40608 = 1111111111111111 40609 = 0000000000000000

Matrice cible **obtenue par XOR**
40608 = 0000000011111111 40609 = 1111111100000000

Note : Pour conserver la configuration binaire cible d'origine des registres 40608 et 40609, copiez les informations dans une autre table à l'aide de l'instruction BLKIM avant d'exécuter l'opération XOR.

Description des paramètres

Longueur de la matrice (partie basse)	Le nombre entier saisi en partie basse indique la longueur de la matrice, c'est-à-dire le nombre de registres ou de mots 16 bits des deux matrices. La longueur est comprise entre 1 et 100. La longueur 2 indique que 32 bits de chaque matrice seront liés par XOR.

Annexes

Optimisation des performances RIO à l'aide de l'ordonnanceur de segments

Objectif

Cette section montre comment optimiser vos RIO à l'aide de l'ordonnanceur de segments.

Contenu de cette annexe

Cette annexe contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
A	Annexe A	1211

Annexe A

A

Optimisation des performances RIO à l'aide de l'ordonnanceur de segments

Objectif

Cette annexe montre comment optimiser les performances RIO à l'aide de l'ordonnanceur de segments.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Temps de cycle	1212
Mesure du temps de cycle	1216
Optimisation du débit	1217
Ordre d'exécution	1220
Utiliser l'ordonnanceur de segments pour améliorer le débit des E/S critiques	1221
Utiliser l'ordonnanceur de segments pour améliorer les performances du système	1223
Utiliser l'ordonnanceur de segments pour améliorer le traitement des ports de communication	1224
Fonctions de cycle	1225

Temps de cycle

Vue d'ensemble

Le temps nécessaire à l'automate pour exécuter le programme logique et mettre à jour le système physique est appelé temps de cycle. Il inclut le temps que prend l'automate pour :

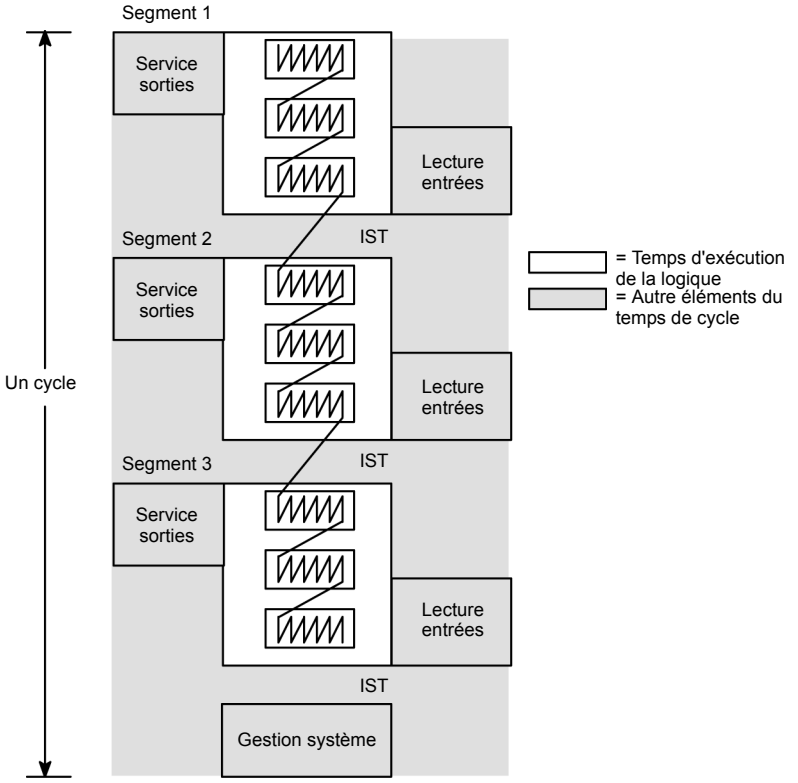
- exécuter toute la logique programmée, c'est-à-dire le temps d'exécution de la logique ;
- traiter les stations d'E/S ;
- prendre en charge les ports de communication et les processeurs d'option ;
- exécuter le transfert inter-segments (IST) et le diagnostic système.

Temps d'exécution de la logique

Le temps d'exécution de la logique est le temps nécessaire à l'UC pour exécuter les éléments et les instructions utilisés dans le programme logique. Il fait partie du temps de cycle total qui est indépendant du temps de service des E/S et du temps de gestion système. Le temps d'exécution de la logique se mesure en ms/K mots de logique utilisateur. Les différents modèles d'automates présentent des temps d'exécution de la logique différents, comme illustré ci-dessous :

Temps d'exécution de la logique	Modèles d'automates	Types d'automates
0,75 ms/Kmots	984A, 984B, 984X	Montage sur châssis
1,0 ms/Kmots	E984-685/-785, L984-785	Montage sur emplacement
	CPU11302, CPU11303, CPU21304	Série Quantum
1,5 ms/Kmots	AT-984, MC-984	Basé sur l'hôte
	0984-780/-785	Montage sur emplacement
2,0 ms/Kmots	Q984	Basé sur l'hôte
	0984-685	Montage sur emplacement
2,5 ms/Kmots	110CPU51x et 110CPU61x	Micro
3,0 ms/Kmots	984-385, 984-485, 984-680	Montage sur emplacement
4,25 ms/Kmots	984-A12x, 984-A13x, 984-A14x	Compact
	110CPU311 et 110CPU411	Micro
5,0 ms/Kmots	984-380/-381, 984-480	Montage sur emplacement

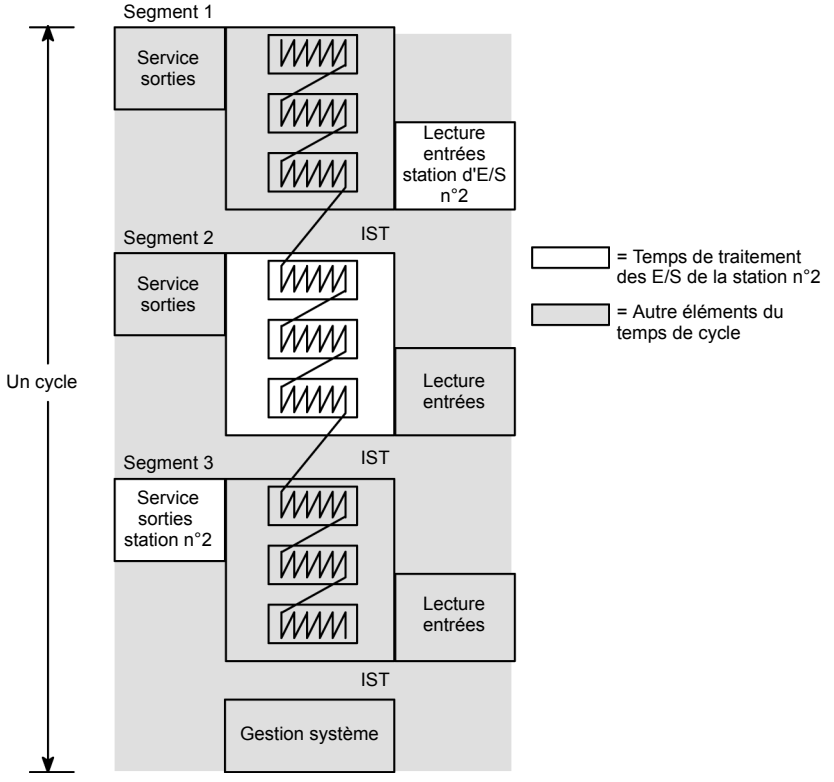
L'illustration suivante montre comment le temps d'exécution de la logique intervient dans le fonctionnement du temps de cycle total :



Traitement des E/S

Afin de gérer le débit du système efficacement, l'automate coordonne la solution des segments logiques via son UC et le traitement des stations d'E/S via son processeur d'E/S. En général, un segment logique est coordonné avec une station d'E/S particulière ; par exemple, les réseaux logiques du segment 2 correspondent aux points d'E/S monde réel de la station 2. Les entrées sont lues au segment précédent et les sorties sont écrites au segment suivant.

Cette méthode de traitement des E/S garantit que le dernier état de l'entrée est disponible pour l'exécution de la logique et que les sorties sont écrites dès que possible après l'exécution de la logique. Elle assure la cohérence entre l'automate et le procédé qu'il contrôle.

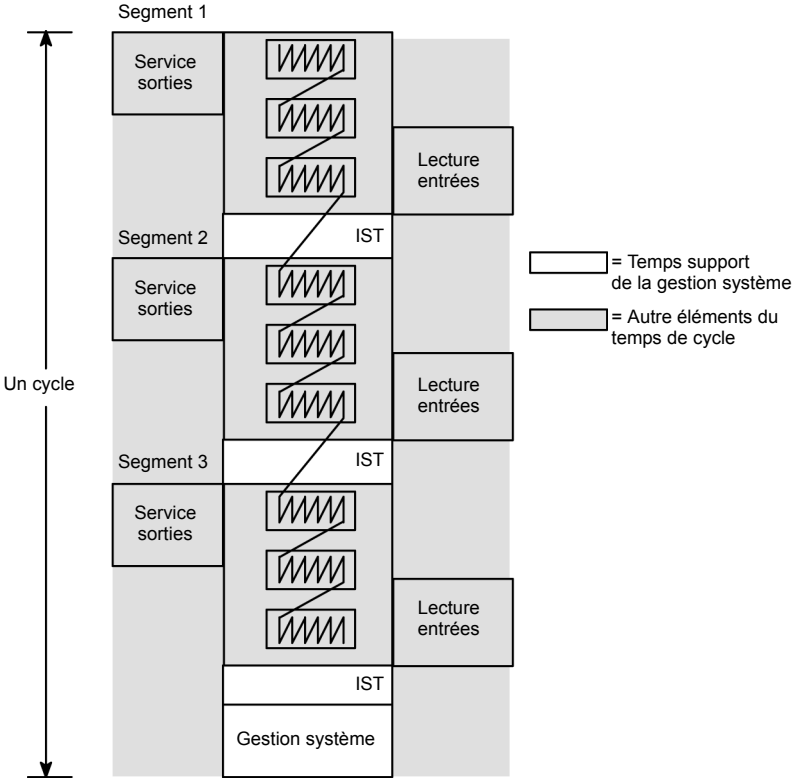


Gestion système

Un transfert inter-segments (IST) se produit entre chaque segment. A ce stade, le processeur d'E/S et la mémoire d'état échangent des données ; les entrées précédentes sont transférées vers la mémoire d'état et les sorties suivantes sont transférées vers le processeur d'E/S. Le cycle logique et le traitement des E/S de chaque segment sont coordonnés de cette manière. Utilisant l'accès mémoire direct (DMA), l'IST requiert en général moins de 1 ms/segment.

A la fin de chaque cycle, les messages d'entrée vers les ports de communication Modbus sont traités. Le temps maximum alloué au traitement des ports de communication est de 2,5 ms/cycle ; les temps de traitement typiques sont inférieurs à 1 ms/cycle. Si l'automate utilise des processeurs d'option (coprocesseurs C986 ou processeurs de communication distribuée D908), ils sont également traités à la fin de chaque cycle et requièrent en général moins de 1 ms/cycle.

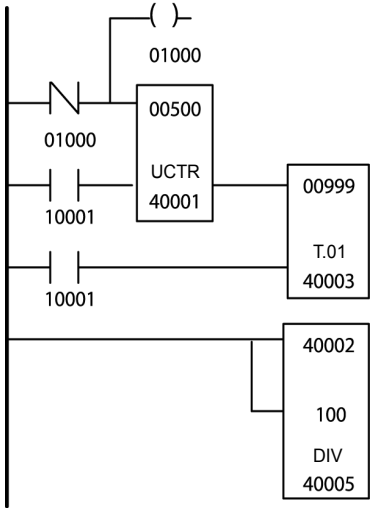
Les diagnostics système utilisent entre 1 et 2 ms/cycle pour s'exécuter, en fonction du type de l'automate.



Mesure du temps de cycle

Vue d'ensemble

Le circuit de schéma à contacts suivant peut être utilisé dans votre programme applicatif afin d'évaluer le temps de cycle du système :



Le compteur recense 1 000 cycles étant donné qu'il effectue 500 transitions. Lorsque le compteur a réalisé les 500 transitions, le temporisateur T.01 est désactivé et enregistre dans le registre 40003 le nombre de centièmes de secondes qui ont été nécessaires au compteur pour réaliser les 500 transitions (1 000 cycles).

La valeur enregistrée en 40002/40003 du bloc DIV est ensuite divisée par 100 et le résultat, qui représente le temps d'exécution de la logique en ms, est enregistré dans le registre 40005.

Note : 10001 est piloté via une commande DESACTIVER ou une entrée câblée ; si vous exécutez le programme en mode optimisé, une entrée câblée est requise pour basculer 10001.

Note : La durée maximum autorisée pour un cycle est de 250 ms ; si le cycle n'est pas terminé dans la durée impartie, une temporisation chien de garde de l'UC arrête l'application et transmet un message d'erreur de timeout à l'affichage de la console de programmation. La limite maximum du temps de cycle empêche l'automate d'entrer dans une boucle sans fin.

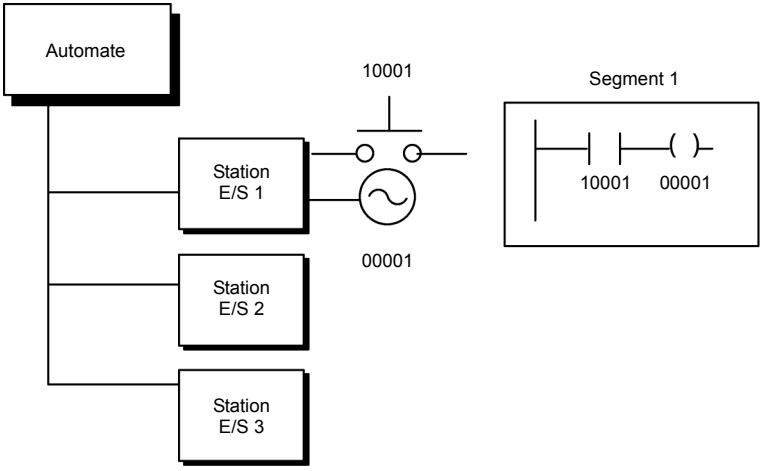
Optimisation du débit

Vue d'ensemble

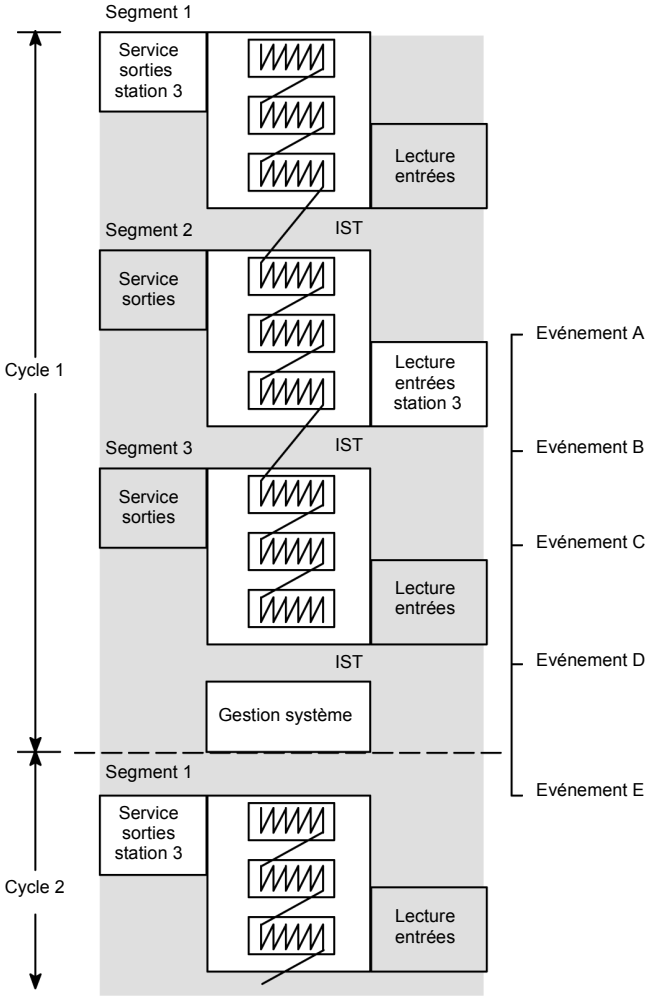
L'architecture de l'automate exécute la logique tout en traitant les stations d'E/S afin d'optimiser le débit du système. Le débit est le temps nécessaire pour transmettre à l'automate un signal reçu par un périphérique de lecture en unité sous forme d'entrée, le traiter dans le schéma à contacts et le retourner sous forme de signal de sortie à un périphérique en unité actif. Le temps de débit peut être plus long ou plus court qu'un simple cycle ; il vous donne une mesure réaliste des performances réelles du système.

La situation de débit idéal

Si l'ordonnanceur de segments par défaut est en place, le système exécute automatiquement la logique en commençant au segment 1 et en se déplaçant de manière séquentielle au segment n. Le débit est optimisé lorsque la logique se référant aux E/S monde réel est contenue dans le segment qui correspond à cette station d'E/S. Par exemple, si vous utilisez des E/S de la station 1 d'un système à trois stations pour contrôler un bouton de commande qui démarre un moteur, la condition idéale est que le segment logique 1 contienne toute la logique appropriée :



Lorsque tous les segments logiques sont coordonnés avec toutes les stations d'E/S physiques de cette manière, le débit d'un segment logique donné peut être inférieur à un cycle. Voici comment il peut être représenté dans notre modèle de temps de cycle :


Segment 1
Service sorties station 3
Lecture entrées
Segment 2
IST
Service sorties
Lecture entrées station 3
Segment 3
IST
Service sorties
Lecture entrées
IST
Gestion système
Segment 1
Service sorties station 3
Lecture entrées
Cycle 1
Cycle 2
Evénement A
Evénement B
Evénement C
Evénement D
Evénement E

Le modèle suit le débit de la station d'E/S 3. Dans cet exemple optimal, le débit représente près de 75 % du temps de cycle total. Cinq événements repères sont illustrés :

- l'événement A, où les entrées de la station 3 sont disponibles pour le processeur d'E/S ;
- l'événement B, où le processeur d'E/S transfère les données à la mémoire d'état ;
- l'événement C, où les réseaux logiques du segment 3 sont exécutés ;
- l'événement D, où les données sont transférées de la mémoire d'état au processeur d'E/S ;
- l'événement E, où les données de sortie sont écrites dans les modules d'entrée au niveau de la station d'E/S 3.

Ordre d'exécution

Vue d'ensemble

Vous spécifiez le numéro des segments et des stations d'E/S à l'aide de l'éditeur de configuration de votre ensemble logiciel de console. La condition d'ordre d'exécution par défaut va du segment 1 au segment n dans l'ordre et de manière continue, une fois par cycle ; les stations d'E/S correspondantes sont traitées dans le même ordre. Vous pouvez modifier l'ordre d'exécution à l'aide de l'éditeur de l'ordonnanceur de segments fourni dans votre ensemble logiciel de console.

Il est parfois possible de modifier cet ordre afin d'améliorer les performances globales du système. L'ordonnanceur de segments peut être utilisé pour :

- améliorer le débit des E/S critiques ;
- améliorer les performances globales du système ;
- optimiser le traitement des ports de communication.

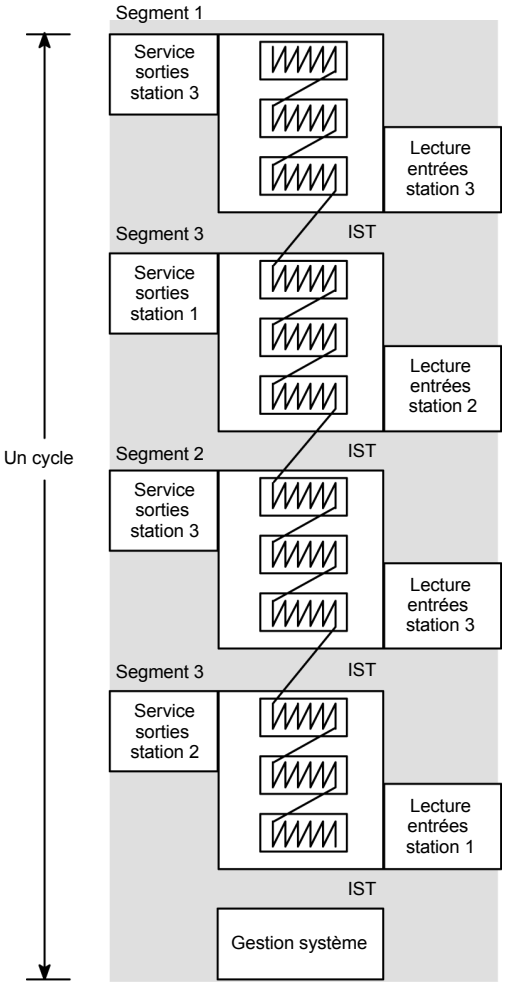
Utiliser l'ordonnanceur de segments pour améliorer le débit des E/S critiques

Vue d'ensemble

Supposons que votre programme logique soit long de trois segments et que le segment 3 contienne la logique critique pour votre application, surveiller un commutateur de proximité pour vérifier la présence de pièces, par exemple. Les segments 1 et 2 exécutent une logique non critique telle qu'une analyse de compte de pièces et la collecte de statistiques. Le programme fonctionne dans le mode standard ordre d'exécution et vous découvrez que l'automate n'est pas en mesure de lire des entrées critiques avec la fréquence souhaitée, entraînant par conséquent un retard système inacceptable.

A l'aide de l'éditeur de l'ordonnanceur de segments, vous pouvez améliorer le débit des E/S critiques à la station 3 en ordonnançant l'exécution du segment 3 deux fois (au minimum) au cours du même cycle.

Voici l'exemple d'un programme logique réordonnancé. Il reprend notre modèle de temps de cycle précédent :



En réordonnançant la table d'ordre d'exécution, vous augmentez effectivement le temps de cycle, mais plus important encore, vous améliorez le débit des E/S critiques géré par la logique du segment 3. Le débit constitue la meilleure mesure des performances du système.

Utiliser l'ordonnanceur de segments pour améliorer les performances du système

Vue d'ensemble

Cette méthode peut être utilisée lorsque certaines zones d'un programme de schéma à contacts n'ont pas besoin d'être exécutées en continu sur chaque cycle. Par exemple, un programme de gestion d'alarme, un programme d'analyse de données ou plusieurs programmes de télégrammes de diagnostic peuvent être désignés comme segments pilotés par l'éditeur de l'ordonnanceur de segments. Basé sur l'état d'une E/S ou d'une référence interne, un segment piloté peut être ordonnancé pour être sauté, et de ce fait, réduire le temps de cycle et améliorer le débit global du système.

Supposons, par exemple, que vous disposez d'une logique de gestion des alarmes dans le segment 2 d'un programme de logique à trois segments. Vous pouvez utiliser l'éditeur de l'ordonnanceur de segments pour piloter le segment 2 basé sur l'état d'une bobine 00056 — si la bobine est ON, la logique du segment 2 sera activée dans le cycle, si la bobine est OFF, le segment ne sera pas exécuté au cours du cycle.

Utiliser l'ordonnanceur de segments pour améliorer le traitement des ports de communication

Vue d'ensemble

Lorsque vous trouvez que la fréquence de traitement standard de fin de cycle des ports de communication, des processeurs d'option ou des diagnostics système est inappropriée aux exigences de votre application, vous pouvez accroître la fréquence de traitement en insérant un ou plusieurs programmes de réinitialisation de temporisation chien de garde dans la table d'ordre d'exécution. A chaque fois que l'UC est confrontée à ce programme, tous les ports de communication sont traités et le diagnostic système s'exécute.

Fonctions de cycle

Vue d'ensemble

Les fonctions de cycle permettent d'analyser un programme logique à intervalles fixes. L'automate n'exécute pas la logique plus rapidement pour autant et ne termine pas les cycles prématurément.

Cycle constant

Le cycle constant vous permet de paramétrer des temps de cycle compris entre 10 et 200 ms (en multiples de 10). Un temps de cycle est le temps entre le début d'un cycle et le début du suivant ; il ne s'agit pas du temps entre la fin d'un cycle et le début du prochain.

Le cycle constant est utile dans des applications où des données doivent être échantillonnées à intervalles de temps constants. Si un cycle constant est appelé avec un laps de temps inférieur au temps de cycle réel, ce temps sera ignoré et le système utilisera sa propre fréquence de cycle normale. Le temps de cycle constant intègre l'exécution de la logique, le traitement des E/S et du port Modbus, ainsi que le diagnostic système. Si vous réglez un temps de cycle sur 40 ms et que l'exécution de la logique, le service des E/S et les diagnostics ne requièrent que 30 ms, l'automate patientera 10 ms sur chaque cycle.

Cycle unique

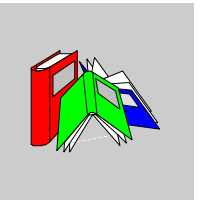
La fonction Cycle unique permet à l'automate d'exécuter un nombre fixe de cycles (1 à 15) et d'arrêter l'exécution de la logique tout en poursuivant le traitement des E/S. Cette fonction est utile pour les diagnostics, elle permet d'examiner la logique exécutée, les données déplacées et les calculs réalisés afin de détecter les erreurs.

⚠ AVERTISSEMENT

La fonction Cycle unique ne doit pas être utilisée pour déboguer des commandes sur des machines-outils, des procédés ou des systèmes de gestion de matériel lorsqu'ils sont actifs. Une fois qu'un nombre défini de cycles a été exécuté, toutes les sorties sont figées à leur dernier état. Etant donné qu'aucune exécution de logique ne prend le relais, l'automate ignore toutes les informations d'entrée. Il peut en résulter un fonctionnement peu sûr, dangereux et préjudiciable à la machine ou au procédé connecté à l'automate.

Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.

Glossaire



A

Abonné de réseau

Un abonné est un appareil avec une adresse (164) sur le réseau Modbus Plus.

Abonné local du réseau

L'abonné local est celui qui est projeté à l'instant.

Adresse abonné

L'adresse abonné sert à la désignation univoque d'un abonné du réseau dans l'itinéraire de routage. L'adresse est réglée directement sur l'abonné, par le bouton rotatif sur la face arrière du module, par exemple.

Adresses

Les adresses (directes) sont des zones de mémoire dans l'API. Celles-ci se trouvent dans la mémoire d'état et peuvent être affectées à des modules d'entrée/sortie. L'affichage/la saisie d'adresses directes est possible dans les formats suivants :

- Format standard (400001)
- Format séparateur (4:00001)
- Format compact (4:1)
- Format CEI (QW1)

Affectation des E/S

L'affectation des E/S est une liste d'équipements générée à partir de la liste d'équipements de l'utilisateur. L'affectation des E/S est gérée dans l'API et contient par exemple des informations d'état des stations et modules E/S, en plus de la liste d'équipements de l'utilisateur.

ANL_IN ANL_IN est le type de données de "entrée analogique" et est utilisé pour le traitement des valeurs analogiques. Lesréférences 3x du module d'entrée analogique configuré déterminées dans la liste d'affectation des E/S sont affectées automatiquement au type de données et doivent de ce fait être occupées uniquement par des variables non localisées.

ANL_OUT ANL_OUT est le type de données de "sortie analogique" et est utilisé pour le traitement des valeurs analogiques. Lesréférences 4x du module de sortie analogique configuré déterminées dans la liste d'affectation des E/S sont affectées automatiquement au type de données et doivent de ce fait être occupées uniquement par des variables non localisées.

ANY Dans la présente version, "ANY" comprend les types de données BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME et WORD ainsi que les types de données qui en sont dérivés.

ANY_BIT Dans la présente version, "ANY_BIT" comprend les types de données BOOL, BYTE et WORD.

ANY_ELEM Dans la présente version, "ANY_ELEM" comprend les types de données BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME et WORD.

ANY_INT Dans la présente version, "ANY_INT" comprend les types de données DINT, INT, UDINT et UINT.

ANY_NUM Dans la présente version, "ANY_NUM" comprend les types de données DINT, INT, REAL, UDINT et UINT.

ANY_REAL Dans la présente version, "ANY_REAL" correspond au type de données REAL.

API Automate programmable industriel

Appareil de programmation Le matériel et le logiciel gérant la programmation, l'élaboration, le test, la mise en service et la recherche de défauts dans les applications API ainsi que dans les applications système décentralisées, afin de rendre possible la documentation et l'archivage des sources. L'appareil de programmation peut, le cas échéant, également être utilisé pour la visualisation du procédé.

Appel La procédure par laquelle l'exécution d'une opération est lancée.

Argument Synonyme de paramètre réel.

Atrium L'automate basé sur PC est monté sur platine standard AT et s'utilise au sein d'un ordinateur hôte dans un emplacement de bus ISA. Ce module possède une carte mère (nécessite un pilote SA85) avec deux emplacements pour cartes filles PC104. L'une des cartes fille PC104 sert d'UC et l'autre à la commande Interbus S.

Avertissement Si un état critique est identifié lors de l'exécution d'un FFB ou d'une étape (par exemple des valeurs d'entrée critiques ou des limites temporelles dépassées), un avertissement survient. Celui-ci peut être visualisé à l'aide de la commande **En ligne** → **Affichage événements...** . Sur les FFB, la sortie ENO reste sur "1".

B

Base de données de projet La base de données de l'appareil de programmation, contenant les informations de configuration d'un projet.

Bibliothèque Ensemble d'objets logiciels prévus pour la réutilisation lors de la programmation de nouveaux projets ou bien même pour l'élaboration de nouvelles bibliothèques. Les exemples sont les bibliothèques des types de blocs fonction élémentaires.
Les bibliothèques EFBpeuvent être subdivisées en groupes.

Bits d'entrée (Références 1x) L'état 1/0 des bits d'entrée est commandé par les données du procédé arrivant depuis un périphérique d'entrée dans l'UC.

Note : Le x suivant le premier chiffre du type de référence représente un emplacement sur cinq positions en mémoire de données utilisateur, par exemple la référence 100201 signifie un bit d'entrée à l'adresse 201 de la mémoire d'état.

Bits d'état Il existe un bit d'état pour chaque abonné à entrée globale, entrée ou sortie spécifique de données de diffusion. Si un groupe de données défini a pu être transmis avec succès avant écoulement du timeout réglé, le bit d'état correspondant est mis à 1. Dans le cas contraire, ce bit est mis à 0 et toutes les données appartenant à ce groupe (à 0) sont effacées.

Bits de sortie/ bits internes (Références 0x) Un bit de sortie/bit interne peut être utilisé pour commander des données de sortie réelles via une unité de sortie du système de contrôle ou pour définir une ou plusieurs sorties discrètes en mémoire d'état. Indication : Le x suivant immédiatement le premier chiffre du type de référence, représente un emplacement mémoire sur 5 chiffres dans la mémoire de données utilisateur, par exemple la référence 000201 signifie un bit interne ou de sortie à l'adresse 201 de la mémoire d'état.

Bloc fonction (instance) (FB)

Un bloc fonction est une unité d'organisation de programme, qui, en fonction de sa fonctionnalité définie dans la description de type de bloc fonction, calcule des valeurs pour ses sorties et variable(s) interne(s), lorsqu'elle est appelée comme instance particulière. Toutes les valeurs des sorties et variables internes d'une instance particulière de bloc fonction sont conservées d'un appel du bloc fonction au suivant. Des appels répétés de la même instance de bloc fonction avec les mêmes arguments (valeurs des paramètres d'entrée) ne délivrent de ce fait pas forcément la (les) même valeur(s) de sortie.
Chaque instance de bloc fonction est représentée graphiquement par un symbole bloc rectangulaire. Le nom du type de bloc fonction est situé en haut au milieu, à l'intérieur du rectangle. Le nom de l'instance de bloc fonction est également en haut, bien qu'à l'extérieur du rectangle. Il est généré automatiquement à la création d'une instance mais peut si besoin, être modifié par l'utilisateur. Les entrées sont représentées à gauche, les sorties à droite du bloc. Les noms des paramètres formels d'entrée/sortie sont indiqués à l'intérieur du rectangle aux places correspondantes.
La description précédant la représentation graphique est valable de principe également pour lesappels de fonction et pour les appels DFB. Les différences sont décrites dans les définitions correspondantes.

Bobine

Une bobine est un élément LD transmettant, sans l'altérer, l'état de la liaison horizontale sur sa gauche à l'état de la liaison horizontale sur sa droite. L'état est alors mémorisé dans la variable/adresse directe.

BOOL

BOOL signifie type de données "booléen". La longueur des éléments de données est 1 bit (stocké en mémoire sur 1 octet). La plage de valeurs des variables de ce type de données est 0 (FALSE) et 1 (TRUE).

BYTE

BYTE est le type de données "cordon binaire 8". L'entrée peut se faire en libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 8 bits. Il n'est pas possible d'affecter à ce type de données de plage de valeur numérique.

C

CEI 611313 Norme internationale : Automates programmables - Partie 3 : Langages de programmation.

Configuration de transmission de données Paramètres déterminant comment les informations sont transmises depuis votre appareil de programmation vers l'API.

Connexion série En connexion série (COM), les informations sont transmises bit par bit.

Constantes Une constante est une variable non localisée, à laquelle est affectée une valeur qui ne peut être modifiée par la logique de programme (lecture seule).

Contact Un contact est un élément LD transmettant un état sur la liaison horizontale située à sa droite. Cet état est le résultat d'un ET booléen entre l'état de la liaison horizontale sur sa gauche et l'état de la variable/adresse directequi lui est affectée. Un contact ne modifie pas la valeur des variables/adresses directes attribuées.

Convention CEI sur les noms (Identificateur) Un identificateur est une suite de lettres, chiffres et caractères de soulignement devant commencer par une lettre ou un caractère de soulignement (par exemple nom d'un type de bloc fonction, d'une instance, d'une variable ou d'une section). Les lettres des polices de caractères nationales (par exemple : ö,ü, é, õ) peuvent être utilisées sauf dans les noms de projets et de DFB.
Les caractères de soulignement sont significatifs dans les identificateurs ; par exemple "A_BCD" et "AB_CD" sont interprétés comme identificateurs différents.
Plusieurs caractères de soulignement de tête ou de suite ne sont pas autorisés.
Les identificateurs ne doivent pas comporter d'espaces. Les majuscules/minuscules ne sont pas significatives ; par exemple "ABCD" et "abcd" sont interprétés comme identificateurs identiques.
Les identificateurs ne doivent pas être des mots clé.

Cordon de bits C'est un élément de données constitué d'un ou de plusieurs bits.

Cycle programme Un cycle programme consiste en la lecture des entrées, le traitement de la logique de programme et l'édition des sorties.

D

DDE (Echange de données en dynamique)

L'interface DDE permet à deux programmes sous Windows d'échanger des données en dynamique. L'utilisateur peut se servir de l'interface DDE en moniteur étendu afin d'appeler ses propres applications d'affichage. Avec cette interface, l'utilisateur (c.-à-d. le client DDE) non seulement peut lire des données du moniteur étendu (le serveur DDE), mais peut également écrire des données sur l'API via le serveur. L'utilisateur peut ainsi modifier directement des données dans l'API tout en surveillant et en analysant les résultats. Lors de l'utilisation de cette interface, l'utilisateur peut créer son propre "Outil graphique", "Face Plate" ou "Outil de réglage", et intégrer celui-ci dans le système. Ces outils peuvent être écrits dans n'importe quelle langue qui gère le DDE (Visual Basic, Visual-C++, par exemple). Ils sont appelés lorsque l'utilisateur actionne l'un des boutons de commande de la boîte de dialogue Moniteur avancé. Outil graphique Concept : Grâce au lien DDE entre Concept et l'outil graphique Concept, il est possible de représenter les signaux d'une configuration sous forme de chronogramme.

Déclaration

Le mécanisme que permet d'établir la définition d'un élément de langage. Normalement, une déclaration comprend le rattachement d'un identificateur à l'élément de langage et l'affectation d'attributs, tels que lestypes de données et les algorithmes, à l'objet de langage concerné..

Défaut

Si lors du traitement d'un FFB ou d'une étape une erreur est détectée (par exemple valeur d'entrée non autorisée ou erreur de durée), un message d'erreur est généré, lequel peut être visualisé à l'aide de la commande**En ligne → Affichage événements...** . Sur les FFB la sortie ENOest mise à "0".

Derived Function Block (DFB) (Bloc fonction dérivé)

Un bloc fonction dérivé représente l'appel d'un type de bloc fonction dérivé. Vous trouverez des détails de la forme graphique de l'appel dans la définition "Bloc fonction (instance)". A l'opposé de l'appel de types d'EFB, les appels de types DFB sont caractérisés par des lignes doubles verticales sur les côtés gauche et droit du symbole rectangulaire du bloc. Le corps d'un type de bloc fonction dérivé est projeté en langage langage FBD quoique seulement dans la version actuelle du système de programmation. D'autres langages CEI ne peuvent pas encore à ce jour être utilisés pour la définition de types DFB, même les fonctions dérivées ne peuvent dans la version actuelle être encore définies. On fait la distinction entre les DFB locaux et globaux.

DFB globaux

Les DFB globaux sont disponibles dans tout projet Concept et sont enregistrés dans le répertoire DFB directement situé sous le répertoire Concept.

DFB locaux Les DFB locaux ne sont disponibles que dans un seul projet Concept et sont enregistrés dans le répertoire DFB sous le répertoire de projet..

Diagramme fonctionnel en séquence (SFC) Leséléments de langage SFC permettent de subdiviser une Unité d'organisation de programme en certain nombre d'étapes et de transitions, reliées entre elles par des liaisons dirigées. A chaque étape correspond un nombre d'actions et à chaque transition est associée une condition de transition.

DINT DINT signifie type de données "entier double (double integer)". L'entrée s'effectue en libellé entier, libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 32 bits. La plage de valeurs pour les variables de ce type de données va de -2 exp (31) à 2 exp (31) -1.

DP (PROFIBUS) RP = Remote Peripheral (périphérie décentralisée)

DX Zoom Cette caractéristique vous permet de vous raccorder sur un objet de programmation afin d'en surveiller des valeurs et de les modifier, si nécessaire.

E

Elément de langage Chaque élément de base dans l'un des langages de programmation CEI, par exemple une étape dans SFC, une instance de bloc fonction dans FBD ou la valeur de départ d'une variable.

EN / ENO (autorisation / affichage d'erreur) Si la valeur de EN vaut "0", lorsque le FFB est lancé, les algorithmes définis par le FFB ne sont pas exécutés et toutes les sorties conservent leur valeur précédente. La valeur de ENO est dans ce cas mise automatiquement à "0". Si la valeur de EN est "1" lors de l'appel du FFB, les algorithmes définis par le FFB sont exécutés. Après l'exécution sans erreur de ces algorithmes, la valeur de ENO est mise automatiquement à "1". Si une erreur survient lors de l'exécution de ces algorithmes, ENO est mis automatiquement à "0". Le comportement de sortie des FFB est indépendant du fait que ceux-ci sont appelés sans EN/ENO ou avec EN=1. Si l'affichage de EN/ENO est activé, l'entrée EN doit absolument être câblée. Le FFB n'est sinon jamais exécuté. L'activation/la désactivation de EN et ENO se fait dans la boîte de dialogue des caractéristiques du bloc fonction. La boîte de dialogue est appelée via les commandes **Objets** → **Propriétés...** ou en cliquant deux fois sur le FFB.

Enoncé (ST) Les énoncés sont des "commandes" du langage de programmation ST. Les énoncés doivent être terminés par des points-virgules. Plusieurs énoncés (séparés par des points-virgules) peuvent se trouver sur une même ligne.

Erreur d'exécution Erreur, survenant lors du traitement du programme sur l'API, sur des objets SFC (les étapes, par exemple) ou des FFB. Il s'agit par exemple de dépassement de plage de valeurs sur les compteurs ou bien d'erreurs temporelles sur les étapes.

Etape Elément de langage SFC : Situation dans laquelle le comportement d'un programme suit, en fonction de ses entrées et sorties, les opérations définies par les actions correspondantes de l'étape.

Etape initiale (Etape de départ) L'étape de démarrage d'une séquence. Une étape initiale doit être définie dans chaque séquence. La séquence est démarrée à son premier appel par l'étape initiale.

Evaluation C'est le processus par lequel est déterminée une valeur d'une fonction ou des sorties d'un bloc fonction lors de l'exécution du programme.

Expression Les expressions sont constituées d'opérateurs et d'opérandes.

F

Fenêtre active Il s'agit de la fenêtre momentanément sélectionnée. Pour un instant donné, seule une fenêtre peut être active. Lorsqu'une fenêtre devient active, la couleur de sa barre de titre change afin de la distinguer des autres fenêtres. Les fenêtres non sélectionnées ne sont pas actives.

Fenêtre d'application Il s'agit de la fenêtre contenant l'espace de travail, la barre de menus et la barre d'outils du programme applicatif. Le nom du programme applicatif apparaît dans la barre de titre. Une fenêtre d'application peut contenir plusieurs fenêtres de document. Dans Concept, la fenêtre d'application correspond à un projet.

Fenêtre de document Une fenêtre contenue dans une fenêtre d'application. Plusieurs fenêtres de document peuvent être ouvertes simultanément dans une fenêtre d'application. Mais seule une fenêtre de document peut être active. Les fenêtres de document dans Concept sont, par exemple, les sections, la fenêtre des messages, l'éditeur de données de références et la configuration de l'API.

FFB (fonctions/ blocs fonction) Terme générique désignant le EFB (Fonction/Blocs fonction Elémentaires) et les DFB (Blocs fonction dérivés)

Fichier de code source (Concept EFB) Le fichier code source est un fichier source ordinaire en C++. Après exécution de la commande **Bibliothèque** → **Créer des fichiers**, ce fichier contient un cadre de code EFB dans lequel vous devez porter un code spécifique de l'EFB sélectionné. Pour ce faire, lancez la commande **Objets** → **Source**.

Fichier de sauvegarde (Concept EFB)

Le fichier de sauvegarde est une copie du dernier fichier de code source. Le nom de ce fichier de sauvegarde est "backup??.c" (on suppose ce faisant que vous n'avez jamais plus de 100 copies de votre fichier de sauvegarde). Le premier fichier de sauvegarde porte le nom "backup00.c". Si vous avez procédé à des modifications du fichier de définition n'appelant pas de modification d'interface pour l'EFB, vous pouvez vous dispenser de créer un fichier de sauvegarde en éditant son fichier de code source (**Objets** → **Source**). Si un fichier de sauvegarde est créé, vous pouvez lui donner le nom Fichier source.

Fichier définition (ConceptEFB)

Le fichier définition renferme des informations générales de description de l'EFB sélectionnés et ses paramètres formels.

Fichier factice

Il s'agit d'un fichier vide constitué d'une en-tête contenant diverses informations générales sur le fichier, comme l'auteur, la date de création, la désignation de l'EFB, etc. L'utilisateur doit procéder à la préparation de ce fichier factice à l'aide d'entrées supplémentaires.

Fichier prototype (Concept EFB)

Le fichier prototype contient tous les prototypes des fonctions affectées. On indique en outre, si elle existe, une définition type de la structure de la situation interne.

Fichier Template (Concept EFB)

Le fichier Template est un fichier ASCII contenant des informations de mise en page pour l'éditeur FBD de Concept, ainsi que des paramètres pour la génération de code.

Filtre RIF

(Filtre Finite Impulse Response) Filtre à réponse impulsionnelle finie

Filtre RII

(Filtre Infinite Impulse Response) Filtre à réponse impulsionnelle infinie

Fonction (FUNC)

Une unité d'organisation de programme, délivrant à l'exécution exactement un élément de donnée. Une fonction ne dispose pas d'information d'état interne. Les appels répétés de la même fonction avec les même paramètres d'entrée délivrent toujours les mêmes valeurs de sortie.
Vous trouverez des détails de la forme graphique des appels de fonction dans la définition "Bloc fonction (instance)". A l'opposé de l'appel de blocs fonction, les appels de fonction ne disposent que d'une unique sortie sans nom, son nom étant le nom de la fonction elle-même. En FBD, chaque appel est caractérisé par un numéro unique par le bloc graphique ; ce numéro est créé automatiquement et ne peut pas être modifié.

Fonctions/blocs fonction élémentaires (EFB)

Caractérisation des fonctions ou des blocs fonction, dont les définitions de type n'ont pas été formulées dans l'un des langagesCEI, c'est-à-dire dont les corps par exemple ne peuvent être modifiés à l'aide de l'éditeur DFB (Concept-DFB). Les types EFB sont programmés en "C" et sont mis à disposition en forme précompilée par les bibliothèques.

Format CEI (QW1)

Au début de l'adresse se trouve un identificateur conforme à CEI, suivi de l'adresse à cinq positions :

- %0x12345 = %Q12345
- %1x12345 = %I12345
- %3x12345 = %IW12345
- %4x12345 = %QW12345

Format compact (4:1)

Le premier chiffre (la référence) est séparé par deux points (:) de l'adresse suivante, les zéros de tête n'étant pas indiqués dans l'adresse.

Format séparateur (4:00001)

Le premier chiffre (la référence) est séparé de l'adresse suivante à cinq positions par deux points (:).

Format standard (400001)

L'adresse à cinq positions se situe juste après le premier chiffre (la référence).

G

Groupes (EFB)

Quelques bibliothèques EFB (par exemple la bibliothèque CEI) sont subdivisées en groupes. Cela simplifie, particulièrement dans les importantes bibliothèques, la recherche des EFB.

I

IHM Interface Homme-Machine

Instanciation La création d'une instance.

Instruction (IL) Les instructions sont des "commandes" du langage de programmation IL. Chaque instruction commence à une nouvelle ligne et est suivie d'un opérateur le cas échéant avec modificateur et, si nécessaire pour l'opération concernée, d'une ou de plusieurs opérandes. Si l'instruction utilise plusieurs opérandes, ceux-ci sont séparés par des virgules. Devant l'instruction peut se trouver une étiquette suivie de deux points. Le commentaire doit, s'il existe, être le dernier élément de la liste.

Instruction (LL984) La mission d'un utilisateur lors de la programmation d'automatismes électriques est de mettre en oeuvre des instructions codées de façon opérationnelle sous forme d'objets imagés classés selon les formes identifiables de contact. Les objets du programme ainsi conçu sont convertis au niveau utilisateur en codes opérande utilisables par l'ordinateur, et ce lors de la procédure de chargement. Les codes opérande sont décodés dans l'UC et traités par les fonctions micrologicielles de l'automate, de sorte que la commande désirée soit ainsi mise en oeuvre.

INT INT correspond au type de données "nombre entier (integer)". L'entrée s'effectue en libellé entier, libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 16 bits. La plage de valeurs pour les variables de ce type de données va de -2 exp (15) à 2 exp (15) -1.

Interbus S (PCP) Afin d'utiliser le canal PCP de l'Interbus S et du prétraitement de données de procédé Interbus S (PDV), le configurateur propose maintenant le nouveau type de station d'E/S Interbus S (PCP). A ce type de station d'E/S est affecté de manière fixe le module de connexion Interbus 180-CRP-660-01.
Le 180-CRP-660-01 se distingue du 140-CRP-660-00 seulement par une plage d'E/S sensiblement plus importante en mémoire d'état de l'automate.

J

Jeton

Le jeton du réseau régit la possession momentanée du droit de transmission d'un abonné individuel. Le jeton circule entre les abonnés dans un sens circulaire (croissant) des adresses. Tous les abonnés suivent la rotation du jeton et peuvent obtenir toute sorte de données qui y sont véhiculées.

L

Liaison

Une liaison de contrôle ou de données entre objets graphiques (par exemple étapes dans l'éditeur SFC, blocs fonction dans l'éditeur FBD) au sein d'une section, graphiquement représenté par une ligne.

Liaison locale (Local Link)

La liaison locale de réseau est le réseau reliant l'abonné local à d'autres abonnés, soit directement soit par l'amplificateur de bus.

Liaisons binaires

Il s'agit des liaisons en sorties et entrées de FFB de type de données BOOL.

Libellé de durée

Les unités permises pour les durées (TIME) sont les jours (J), les heures (H), les minutes (M), les secondes (S) et les millisecondes (MS) ou une combinaison de ces unités. La durée doit être caractérisée par le préfixe t#, T#, time# ou TIME#. Le "dépassement" de l'unité de plus grande valeur est admise ; par exemple, l'entrée T#25H15M est permise.

Exemple
t#14MS, T#14.7S, time#18M, TIME#19.9H, t#20.4D, T#25H15M, time#5D14H12M18S3.5MS

Libellé en base 16

Les libellés en base 16 servent à codifier les entiers dans le système hexadécimal. La base doit être repérée par le préfixe 16#.. Les valeurs doivent être non signées (+/-). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple
16#F_F ou 16#FF (décimal 255)
16#E_0 ou 16#E0 (décimal 224)

Libellé en base 2

Les libellés en base 2 servent à codifier les entiers dans le système de base 2. La base doit être repérée par le préfixe 0,91kg. Les valeurs doivent être non signées (+/). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple
2#1111_1111 ou 2#11111111 (décimal 255)
2#1110_1111 ou 2#11100000 (décimal 224)

Libellé en base 8

Les libellés en base 8 servent à codifier les entiers dans le système de base 8. La base doit être repérée par le préfixe 3,63kg. Les valeurs doivent être non signées (+/-). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple
8#3_1111 ou 8#377 (décimal 255)
8#34_1111 ou 8#340 (décimal 224)

Libellé entier

Les libellés entiers servent à indiquer des valeurs entières dans le système décimal. Les valeurs peuvent être signées (+/-). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple
-12, 0, 123_456, +986

Libellés

Les libellés servent à fournir des valeurs directement aux entrées des FFB, conditions de transition etc. Ces valeurs ne peuvent pas être écrasées par la logique du programme (lecture seule). Différenciez entre les libellés génériques et typisés. De plus, les libellés servent à affecter une valeur à une constante ou une valeur initiale à unevariable.
L'entrée se fait en libellé en base 2, libellé en base 8, libellé en base 16, libellé entier, libellé réel ou libellé réel avec exposant.

Libellés génériques

Si le type de données d'un libellé n'a pas d'importance pour vous, indiquez la valeur du libellé. Dans ce cas, Concept affecte automatiquement un type de données adéquat au libellé.

Libellés réels

Les libellés réels servent à indiquer les valeurs à virgule flottante dans le système décimal. Les libellés réels s'identifient au point décimal. Les valeurs peuvent être signées (+/-). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs.

Exemple
-12.0, 0.0, +0.456, 3.14159_26

Libellés réels avec exposant

Les libellés réels avec exposant servent à indiquer les valeurs à virgule flottante dans le système décimal. Les libellés réels avec exposant se caractérisent par le point décimal. L'exposant donne la puissance de dix avec lequel le chiffre de devant doit être multiplié pour obtenir la valeur à représenter. La base peut être précédée d'un signe moins (-). L'exposant peut être signé (+/-). Les caractères de soulignement individuels (_) entre les chiffres ne sont pas significatifs. (Uniquement entre les chiffres, et non avant ou après la virgule ni avant ou après "E", "E+" ou "E-")

Exemple
-1.34E-12 ou -1.34e-12
1.0E+6 ou 1.0e+6
1.234E6 ou 1.234e6

Libellés typisés

Si vous voulez déterminer le type de données d'un libellé, vous pouvez le faire avec la construction suivante : 'nomtypedonnée'#'Valeur du libellé'

Exemple
INT#15 (type de données : entier, valeur : 15),
BYTE#00001111 (type de données : octet, valeur : 00001111)
REAL#23.0 (type de données : réel, valeur : 23.0)

Pour l'affectation du type de données REAL, vous pouvez indiquer la valeur de la manière suivante : 23.0.
En indiquant ce point décimal, le type de données REAL est affecté automatiquement.

Liste d'affectation des E/S

Dans la liste d'affectation des E/S, on configure les modules d'E/S et modules experts des différentes unités centrales.

Liste d'instructions (IL)

IL est un langage littéral conforme à la CEI 1131, dans lequel les opérations, comme les appels sur ou sans condition de blocs fonction et de fonction, les sauts conditionnels ou sans condition etc., sont représentés par des instructions.

Littéral Structuré (ST)

ST est un langage littéral conforme à la CEI 1131, dans lequel les opérations telles que le lancement de blocs fonction et de fonctions, les exécutions conditionnelles d'énoncés, la réitération d'énoncés, entre autres, sont représentées par des énoncés.

M

Macro

Les macros sont créées à l'aide du logiciel Concept DFB.
Les macros servent à dupliquer des sections et des réseaux fréquemment utilisés (y compris leur logique, leurs variables et leur déclaration de variable).
On fait la distinction entre les macros locales et globales.

Les macros possèdent les caractéristiques suivantes :

- Les macros ne peuvent être créées qu'avec les langages FBD et LD.
- Les macros ne contiennent qu'une seule section.
- Elles peuvent contenir une section d'une complexité quelconque.
- D'un point de vue programme, une macro instanciée, c'est-à-dire une macro insérée dans une section, ne se distingue pas d'une section créée de manière conventionnelle.
- Appel de DFB dans une macro
- Déclaration de variables
- Utilisation de structures de données propres aux macros
- Validation automatique des variables déclarées dans la macro
- Valeurs initiales des variables
- Instanciation multiple d'une macro dans tout le programme avec différentes variables
- Le nom de la section, les noms des variables et le nom de la structure de données peuvent comporter jusqu'à 10 repères d'échange (@0 à @9) différents.

Macros globales

Les macros globales sont disponibles dans tout projet Concept et sont enregistrées dans le répertoire DFB directement situé sous le répertoire Concept.

Macros locales

Lesmacros locales ne sont disponibles que dans un seul projet Concept et sont enregistrées dans le répertoire DFB sous le répertoire de projet..

Mémoire d'état

La mémoire d'état est l'emplacement mémoire pour toutes les grandeurs sollicitées dans le programme utilisateur par des références (représentation directe). Par exemple, les bits d'entrée, les bits de sortie/bits internes, les mots d'entrée et mots de sortie/mots internes se trouvent en mémoire d'état.

Mode ASCII

American Standard Code for Information Interchange. Le mode ASCII est utilisé pour la communication avec différents équipements hôte. ASCII fonctionne sur 7 bits de données.

Mode RTU

Remote Terminal Unit
Le mode RTU est utilisé pour la communication entre l'API et un ordinateur personnel compatible IBM. RTU fonctionne sur 8 bits de données.

Module SA85 Le module SA85 est une carte Modbus Plus pour ordinateur IBM-AT ou compatible.

Mots clés Les mots clés sont des combinaisons uniques de caractères utilisés comme éléments spéciaux de syntaxe comme il est défini à l'annexe B de la CEI 1131-3. Tous les mots clés utilisés dans la CEI 1131-3 et donc dans Concept, sont listés en annexe C de la CEI 1131-3. Ces mots clés de la liste ne doivent être utilisés à aucune autre fin (pas comme nom de variable, nom de section, nom d'instance, par exemple).

Mots d'entrée (Références 3x) Un mot d'entrée contient des informations émanant d'une source externe et par lesquelles un nombre sur 16 bits est représenté. Un registre 3x peut également contenir 16 bits successifs lus dans le registre au format binaire ou BCD (binaire codé décimal). Indication : Le x suivant immédiatement le premier chiffre du type de référence, représente un emplacement mémoire à cinq chiffres dans la mémoire de données utilisateur, par exemple la référence 300201 signifie un mot d'entrée de 16 bits à l'adresse 201 de la mémoire d'état.

Mots de sortie/ mots internes (Références 4x) Un mot de sortie/mot interne peut être utilisé pour la mémorisation de données numériques (binaires ou décimales) en mémoire d'état ou bien pour envoyer des données depuis l'UC vers une unité de sortie du système de contrôle. Indication : Le x suivant immédiatement le premier chiffre du type de référence, représente un emplacement mémoire à cinq chiffres dans la mémoire de données utilisateur, par ex la référence 400201 signifie un mot de sortie/mot interne de 16 bits à l'adresse 201 de la mémoire d'état.

N

Nom d'étape

Le nom d'étape sert à la désignation unique d'une étape dans uneunité d'organisation de programme. Le nom d'étape est créé automatiquement, mais peut être édité. Il doit être unique dans toute l'unité d'organisation de programme, sinon unmessage d'erreur apparaît.
Le nom d'étape créé automatiquement a toujours la structure suivante : S_n_m

E = Etape
n = Numéro de la section (numéro courant)
m = Numéro de l'étape dans la section (numéro courant)

Nom d'instance

Un identificateur, associé à une instance spécifique de bloc fonction.. Le nom d'instance sert au repérage sans équivoque d'un bloc fonction au sein d'une unité d'organisation de programme. Le nom d'instance est créé automatiquement, mais peut être édité. Le nom d'instance doit être unique dans toute l'unité d'organisation de programme, la distinction Majuscule/Minuscule n'est pas faite. Si le nom saisi existe déjà, vous en êtes averti et vous devez choisir un autre nom. Le nom d'instance doit répondre aux conventions CEI sur les noms, sinon un message d'erreur survient. Le nom d'instance créé automatiquement a toujours la structure suivante : FBI_n_m

FBI = Instance de bloc fonction
n = Numéro de la section (numéro courant)
m = Numéro de l'objet FFB dans la section (numéro courant)

Numéro d'identification

Le numéro d'identification sert à caractériser de manière unique une fonction dans un programme ou DFB. Le numéro d'identification ne peut être édité et est attribué automatiquement. Il a toujours la structure : .n.m

n = Numéro de la section (numéro courant)
m = Numéro de l'objet FFB dans la section (numéro courant)

O

Opérande Un opérande est un libellé, une variable, une évaluation de fonction ou une expression.

Opérateur Un opérateur est un symbole d'une opération arithmétique ou booléenne à exécuter.

P

Paramètre d'entrée (Entrée) Transmet lors de l'appel d'unFFB l'argument s'y rapportant.

Paramètre de sortie (Sortie) Un paramètre avec lequel est (sont) retourné(s) le(s) résultat(s) de l'évaluation d'un FFB.

Paramètre réel Paramètre d'Entrée-/Sortie actuellement attribué.

Paramètres formels Paramètres d'entrée-/sortie, utilisés au sein de la logique d'un FFB et sortant du FFB en entrées ou en sorties.

Paysage Paysage signifie que la page, au regard du texte imprimé, est plus large que haute.

Pont Un pont est un dispositif permettant de relier des réseaux. Il permet la communication entre abonnés de deux réseaux. Chaque réseau possède sa propre séquence de rotation de jeton le jeton n'est pas transmis par les ponts.

Portrait Portrait signifie que la page, au regard du texte imprimé, est plus haute que large.

Presse-papiers Le presse-papiers est une mémoire temporaire pour les objets coupés ou copiés. Ces objets peuvent être collés dans des sections. A chaque nouveau coupé ou copier, l'ancien contenu du presse-papiers est écrasé.

Processeur de communication Le processeur de communication traite les passages de jeton et le flux de données entre le réseau Modbus Plus et la logique utilisateur de l'API.

Programmation de la redondance d'UC (Hot Standby)	Un système redondant est constitué de deux API configurés de manière identique, et qui communiquent entre eux par des processeurs redondants. En cas de panne de l'API primaire, l'API secondaire prend le contrôle de l'automatisme. Dans les conditions normales, l'API secondaire n'effectue aucune fonction de commande mais il vérifie les informations d'état afin de déceler les erreurs.
Programme	La plus haute unité d'organisation de programme. Un programme est chargé en entier sur un seul API.
Projet	Appellation générale du niveau le plus élevé d'une arborescence logicielle, qui définit le nom de projet supérieur d'une application d'API. Après avoir défini le nom du projet, vous pouvez sauvegarder votre configuration système et votre programme de commande sous ce nom. Toutes les données apparaissant lors de la création de la configuration et du programme font partie de ce projet supérieur pour cette tâche spéciale d'automatisation. Désignation générale du jeu complet d'informations de programmation et de configuration dans la base de données de projet, laquelle représente le code source décrivant l'automatisation d'une installation.

R

REAL

REAL correspond au type de données "nombre à virgule flottante". L'entrée se fait en libellé réel ou en libellé réel avec exposant. La longueur des éléments de données est de 32 bits. La plage de valeurs pour les variables de ce type de données va de 8.43E-37 à 3.36E+38.

Note : En fonction du type de processeur mathématique de l'UC, différentes zones de cette plage de valeur permise ne peuvent pas être affichées. Cela s'applique aux valeurs tendant vers ZERO et aux valeurs tendant vers l'INFINI. Dans ces cas, une valeur NAN (**N**ot **A N**umber) ou INF (**INF**inite) est affichée en mode Animation.

Référence

Toute adresse directe est une référence commençant par un code indiquant s'il s'agit d'une entrée ou d'une sortie et s'il s'agit d'un bit ou d'un mot Les références commençant par le chiffre 6 représentent des registres de la mémoire étendue de la mémoire d'état.
Plage 0x = bits internes/de sortie
Plage 1x = bits d'entrée
Plage 3x = mots d'entrée
Plage 4x = mots internes/de sortie
Plage 6x = registres en mémoire étendue

Note : Le x suivant immédiatement le premier chiffre de chaque type de référence, représente un emplacement mémoire à cinq chiffres dans la mémoire de données utilisateur, par exemple la référence 400201 signifie un mot de sortie/mot interne de 16 bits à l'adresse 201 de la mémoire d'état.

Registres en mémoire étendue (référence 6x)

Les références 6x sont des mots indicateurs en mémoire étendue de l'API. Ils ne peuvent être utilisés que pour les programmes utilisateur LL984 et seulement sur les UC CPU 213 04 ou CPU 424 02.

Représentation directe

Une méthode pour représenter une variable dans un programme d'API, à partir de laquelle peut être déterminée directement une correspondance avec un emplacement logique et indirectement avec l'emplacement physique.

Réseau

Un réseau est une connexion commune d'appareils sur une voie de données commune, et qui communiquent entre eux à l'aide d'un protocole commun.

Réseau décentralisé (DIO)

Une programmation décentralisée dans le réseau Modbus Plus permet une performance maximale de l'échange de donnés et n'a aucune exigence particulière sur les liaisons. La programmation d'un réseau décentralisé est simple. La configuration du réseau ne nécessite pas de logique de schéma à contact supplémentaire. Toutes les conditions du transfert de donnés sont remplies en renseignant les paramètres correspondants du processeur de communication.

RIO (E/S décentralisée)

L'E/S décentralisée indique un emplacement physique des appareils E/S à commande par point par rapport au processeur qui les gère. Les entrées/sorties décentralisées sont reliées avec l'appareil de commande via un câble de communication.

S

Saut

Elément du langage SFC. Les sauts sont utilisés pour éviter des zones de la séquence.

Schéma à contacts (LD)

Le schéma à contacts est un langage de programmation graphique conforme à la norme CEI1131, dont l'aspect visuel suit les "échelons" d'un schéma à relayage.

Schéma à contacts 984 (LL)

Comme leur nom l'indique, les schémas à contacts comportent des contacts. Contrairement à un schéma électrique, les électrotechniciens se servent d'un schéma à contacts pour dessiner un circuit (à l'aide de symboles électriques). Celui-ci doit montrer l'évolution d'événements, et non les fils en présence qui relient les différentes parties entre elles. Une interface de schéma à contacts permet de réaliser une interface utilisateur traditionnelle pour commander les actions des constituants d'automatisme, afin que les électrotechniciens ne soient pas obligés d'apprendre un langage de programmation avec lequel ils ne sont pas familiers. La construction d'un schéma à contacts effectif permet de relier des éléments électriques de manière à créer une sortie de commande. Celle-ci dépend d'un flux d'énergie logique passant par les objets électriques utilisés, lesquels représentent la condition préalable nécessaire d'un appareil électrique physique.
Sous une forme simple, l'interface utilisateur est un écran vidéo élaboré par l'application de programmation d'API, organisant un quadrillage vertical et horizontal dans lequel sont rangés des objets de programmation. Le schéma reçoit de l'énergie par le coté gauche du quadrillage, et par connexion à des objets activés, l'énergie s'écoule de gauche à droite.

Schéma en bloc fonction (FBD)

Une ou plusieurs sections contenant des réseaux représentés graphiquement composés de fonctions, blocs fonction et liaisons.

Section

Une section peut par exemple être utilisée pour décrire le principe de fonctionnement d'une unité technologique telle qu'un moteur.
Un programme ou un DFB est constitué d'une ou de plusieurs sections. Des sections peuvent être programmées à l'aide des langages de programmation CEI FBD et SFC. Au sein d'une même section, seul un des langages de programmation mentionnés peut être utilisé.
Dans Concept, chaque section a sa propre fenêtre de document. Cependant pour des raisons de vue d'ensemble, il est utile de subdiviser une grande section en plusieurs petites. La barre de défilement sert à se déplacer au sein d'une section.

Station d'E/S DCP

A l'aide d'un processeur de contrôle distribué (D908), vous pouvez configurer un réseau décentralisé piloté par un API. Lorsque l'on utilise un D908 avec API décentralisé, l'APIpilote considère l'API décentralisé comme une station d'E/S décentralisée. Le D908 et l'API décentralisé communiquent par le bus système, ce qui permet une grande performance pour un effet minimal sur le temps de cycle. L'échange de données entre le D908 et l'API pilote s'effectue par le bus d'E/S décentralisé à 1,5 Mégabits par seconde. Un API pilote peut gérer jusqu'à 31 processeurs D908 (adresse 2-32).

SY/MAX

Dans les automates Quantum, Concept gère la mise à disposition des modules d'E/S SY/MAX sur l'affectation des E/S pour la commande RIO par l'API Quantum. L'embase distante SY/MAX dispose d'une carte d'E/S décentralisée à l'emplacement 1, laquelle communique par un système d'E/S décentralisées Modicon S908 R. Les modules d'E/S SY/MAX vous sont listés pour la sélection et la prise en compte dans l'affectation des E/S de la configuration Concept.

Symbole (Icône)

Représentation graphique de différents objets sous Windows, par exemple unité de lecteur, programmes utilisateur et fenêtre de document.

T

TIME

TIME est le type de données "temps". L'entrée se fait sous forme de libellé de temps. La longueur des éléments de données est de 32 bits. La plage de valeurs des variables de ce type de données va de 0 à 2 exp (32)-1. L'unité du type de données TIME est 1 ms.

Transition

La condition par laquelle la commande d'une ou de plusieurs étapes précédentes passe à une ou plusieurs étapes suivantes le long d'une liaison.

Type de bloc fonction

Un élément de langage constitué de : 1. la définition d'une structure de données, subdivisée en variables d'entrée, de sortie et internes ; 2. un jeu d'opérations exécutées avec les éléments de la structure de données, lorsqu'une instance du type de bloc fonction est appelée. Ce jeu d'opérations peut être formulé soit dans l'un des langages CEI (type DFB) ou en "C" (type EFB). Un type de bloc fonction peut être instancié (appelé) plusieurs fois.

Type de données dérivé

Les types de données dérivés sont des types de données qui ont été dérivés des types de données élémentaires et/ou d'autres types de données dérivés. La définition des types de données dérivés s'effectue dans l'éditeur de type de données de Concept.
On fait la distinction entre les types de données globaux et les types de données locaux.

Type de données générique

Un type de données représentant plusieurs autres types de données.

Types de données

La vue d'ensemble montre la hiérarchie des types de données génériques, et comment ils sont utilisés à l'entrée et à la sortie des fonctions et de blocs fonction. Les types de données génériques sont caractérisés par le préfixe "ANY".

- ANY_ELEM
  - ANY_NUM
    ANY_REAL (REAL)
    ANY_INT (DINT, INT, UDINT, UINT)
  - ANY_BIT (BOOL, BYTE, WORD)
  - TIME
- Types de données système (Extension CEI)
- dérivé (des types de données 'ANY')

Types de données dérivés globaux

Les types de données dérivés globaux sont disponibles dans tout projet Concept et sont enregistrés dans le répertoire DFB directement situé sous le répertoire Concept.

Types de données dérivés locaux

Les types de données dérivés locaux ne sont disponibles que dans un seul projet Concept et ses DFB locaux et sont enregistrés dans le répertoire DFB sous le répertoire de projet..

U

UDEFB

Fonctions/Blocs fonction élémentaires défini(e)s par l'utilisateur
Fonctions ou blocs fonction créés en langage de programmation C et que Concept met à votre disposition dans desbibliothèques.

UDINT

UDINT représente le type de données "entier double non signé (unsigned double integer)". L'entrée s'effectue en libellé entier, libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 32 bits. La plage de valeurs des variables de ce type de données va de 0 à 2 exp (32)-1.

UINT

UINT représente le type de données "entier non signé (unsigned integer)". L'entrée s'effectue en libellé entier, libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 16 bits. La plage des valeurs des variables de ce type de données va de 0 à (2 exp 16)-1.

Unité d'organisation de programme

Une fonction, un bloc fonction ou un programme. Ce terme peut se rapporter soit à un type soit à une instance.

V

Valeur initiale

La valeur affectée à une variable lors du lancement du programme. L'affectation de la valeur s'effectue sous forme d'un libellé.

Variable localisée

Une adresse de mémoire d'état (adresses de références 0x, 1x, 3x,4x) est affectée aux variables localisées. La valeur de ces variables est enregistrée dans la mémoire d'état et peut être modifiée en ligne avec un éditeur de données de référence. Ces variables peuvent être adressées avec leur nom symbolique ou avec leur adresse de référence.

Toutes les entrées et les sorties de l'API sont reliées à la mémoire d'état. L'accès du programme aux signaux des périphériques connectés à l'API ne se fait que via des variables localisées. Les accès de l'extérieur via les interfaces Modbus ou Modbus Plus de l'API, des systèmes de visualisation par exemple, sont également possibles via des variables localisées.

Variable non localisée

Aucune adresse de mémoire d'état n'est affectée aux variables non localisées. Elles n'occupent donc pas non plus d'adresse de mémoire d'état. La valeur de ces variables est enregistrée dans le système et peut être modifiée en ligne avec un éditeur de données de référence. Ces variables ne dont adressées que par leur nom symbolique.

Les signaux ne disposant pas d'accès à la périphérie (résultats intermédiaires, repères systèmes, par exemple) doivent être de préférence déclarés comme variable non localisée.

Variables

Les variables servent à l'échange de données au sein de sections, entre plusieurs sections et entre leprogramme et l'API.
Les variables consistent au moins en un nom de variables et un type de données. Si une adresse directe (référence) est affectée à une variable, on parle alors de variable localisée. Si aucune adresse directe n'est affectée à une variable, on parle alors de variable non localisée. Si un type de données dérivé est affecté à une variable, on parle alors d'une variable multi-éléments.
Il existe en outre des constantes et des libellés.

Variables de champs

Variables auxquelles sont affectées untype de données dérivé défini à l'aide du mot clé ARRAY (tableau). Un tableau est un ensemble d'éléments de données appartenant au même type.

Variables multi-éléments

Variables, auxquelles est affecté un type de données dérivé défini avec STRUCT ou ARRAY.
On fait ici la distinction entre variables de champs et variables structurées.

Variables structurées

Variables auxquelles est affecté un type de données dérivé défini avec STRUCT (structure).
Une structure est un ensemble d'éléments de données avec en général différents types de données (types de données élémentaires et/ou types de données dérivés).

W

WORD

WORD correspond au type de données "Profil binaire 16". L'entrée peut se faire en libellé en base 2, libellé en base 8 ou libellé en base 16. La longueur des éléments de données est de 16 bits. Il n'est pas possible d'affecter à ce type de données de plage de valeur numérique.

Index

A

ABS, 68
AD16, 115
ADD, 119
Addition, 119
  AD16, 115
  ADD, 119
Addition de valeurs 16 bits, 115
Addition en double précision, 281
Addition en virgule flottante, 287
Addition entier + virgule flottante, 291
Algorithmes PID, 923
AND, 123
Antilogarithme de base 10, 295
ARCCOS, 68
Arccosinus en virgule flottante d'un angle (en radians), 301
ARCSIN, 68
Arcsinus en virgule flottante d'un angle (en radians), 307
ARCTAN, 68
Arctangente en virgule flottante d'un angle (en radians), 313
Argument
  limites, 69
  réseau d'équations, 68
ASCII Functions
  READ, 991
  WRIT, 1149
Automates
  différences d'arrondi, 72
  temps de cycle, 73
Automates Quantum
  différences d'arrondi, 72

B

BCD, 129
Bibliothèque des fonctions de régulation, 831
Bit de sortie
  messages d'erreur, 56
  réseau d'équations, 55
Bit NF, 809
Bit NO, 813
BLKM, 133
BLKT, 137
Bloc vers table, 137
BMDI, 141
Bobines, 97
Booléen, 59
BROT, 145

C

Calcul d'une formule prédéfinie, 861
Calcul du taux dérivé sur une durée définie, 935
Calculateur d'équation formatée, 873
Calculs avancés, 832
Changement du signe d'un nombre en virgule flottante, 319
Chargement de la mémoire flash, 697
Chargement de la valeur en virgule flottante

de "Pi", 435
Checksum, 175
CHS, 167
CKSM, 175
CMPR, 181
Combinaisons d'opérateurs
    réseau d'équation, 70
Communication
    MSTR, 745
COMP, 195
Comparaison d'entrée, 645
Comparaison de configuration binaire
    CMPR, 181
Comparaison de deux valeurs, 1129
Comparaison en virgule flottante, 325
Comparaison entier-virgule flottante, 331
Complément d'une matrice, 195
Compteur, 1133
Configuration de l'entrée en mode manuel ou automatique, 907
Configuration de redondance d'UC, 167
Consignation d'erreurs en virgule flottante, 389
Consignation de données pour support de lecture/écriture PCMCIA, 237
Constante
    réseau d'équation, 53
Contact de validation
    liaison horizontale, 56
    normalement fermé, 56
    normalement ouvert, 56
    ouvert horizontal, 56
Contact normalement fermé
    réseau d'équations, 56
Contact normalement ouvert
    réseau d'équations, 56
Contacts, 97
Contrôle de bit, 805
Conversion
    BCD
        binaire, 129
    binaire
        BCD, 129
Conversion d'entier en virgule flottante, 349
Conversion d'un nombre à virgule flottante en entier, 545
Conversion d'unités physiques et alarmes, 519
Conversion en virgule flottante de degrés en radians, 337
Conversion en virgule flottante de radians en degrés, 355
Conversion entier en virgule flottante, 685
Conversion virgule flottante en entier, 343
Conversions de données
    réseau d'équation, 70
Copie
    BLKM, 133
    BLKT, 137
Copie de bloc, 133
Copie de bloc avec interruptions invalidées, 141
Coprocesseur mathématique
    différences d'arrondi, 72
COS, 68
COSD, 68
Cosinus en virgule flottante d'un angle (en radians), 361
Couches imbriquées
    parenthèses, 53
Counters/Timers
    DCTR, 217
    Temporisation T.01, 1105
    Temporisation T0.1, 1109
    Temporisation T1.0, 1113
    Temporisation T1MS, 1117
    UCTR, 1133
Création d'un réseau d'équations, 54

D

DCTR, 217
Décompteur, 217
Détection, 1029
Différences d'arrondi
    réseau d'équation, 72
DIOH, 221
DIV, 231
Division, 231
Division de valeurs 16 bits, 259
Division en double précision, 367
Division en virgule flottante, 379

DLOG, 237
Données
    équation mathématique, 59
Données de constante
    équation mathématique, 60
    long (32 bits), 60
    LSB (octet de poids faible), 60
    virgule flottante, 60
Données de variable
    équation mathématique, 59
DRUM, 253
DV16, 259

E

E/S immédiate, 659
Ecrire, 1149
    MSTR, 755
Ecriture en mémoire étendue, 1197
Ecriture indirecte d'un bloc, 641
Editeur logique
    réseau d'équation, 53
    réseau d'équations, 54
Elévation d'un nombre en virgule flottante à une puissance entière, 441
EMTH, 273
EMTH-ADDDP, 281
EMTH-ADDFP, 287
EMTH-ADDIF, 291
EMTH-ANLOG, 295
EMTH-ARCOS, 301
EMTH-ARSIN, 307
EMTH-ARTAN, 313
EMTH-CHSIN, 319
EMTH-CMPFP, 325
EMTH-CMPIF, 331
EMTH-CNVDR, 337
EMTH-CNVFI, 343
EMTH-CNVIF, 349
EMTH-CNVRD, 355
EMTH-COS, 361
EMTH-DIVDP, 367
EMTH-DIVFI, 373
EMTH-DIVFP, 379
EMTH-DIVIF, 385
EMTH-ERLOG, 389
EMTH-EXP, 395
EMTH-LNFP, 401
EMTH-LOG, 407
EMTH-LOGFP, 413
EMTH-MULDP, 419
EMTH-MULFP, 425
EMTH-MULIF, 429
EMTH-PI, 435
EMTH-POW, 441
EMTH-SINE, 447
EMTH-SQRFP, 453
EMTH-SQRT, 459
EMTH-SQRTP, 465
EMTH-SUBDP, 471
EMTH-SUBFI, 477
EMTH-SUBFP, 481
EMTH-SUBIF, 487
EMTH-TAN, 493
Entier divisé par nombre à virgule flottante, 385
Entrée analogique, 839
Equation mathématique
    données de constante, 60
    notation exponentielle, 61
    types de valeurs et de données, 58
ESI, 499
Et Logique, 123
Etat, 1049
Etiquette d'un sous–programme, 693
EUCA, 519
Exemple PID, 81
Exemple PID2 de régulation de niveau, 84
EXP, 68
Expression
    réseau d'équations, 64
Expression algébrique
    réseau d'équations, 56
Expression conditionnelle
    réseau d'équation, 53
    réseau d'équations, 64
Expression logique
    réseau d'équation, 53
Expression unique
    réseau d'équations, 64
Expressions groupées dans des couches de

parenthèses imbriquées
réseau d'équation, 53

F

Fast I/O Instructions
BMDI, 141
ID, 651
IE, 655
IMIO, 659
IMOD, 667
ITMR, 677
File d'attente de retard temporel, 867
Filtre avance/retard du premier ordre, 903
FIN, 535
FIX, 68
FLOAT, 68
Fonction
ABS, 68
ARCCOS, 68
ARCSIN, 68
ARCTAN, 68
argument, 68
COS, 68
COSD, 68
EXP, 68
FIX, 68
FLOAT, 68
limites d'argument, 69
LN, 68
LOG, 68
saisie dans un réseau d'équations, 68
SIN, 68
SIND, 68
SQRT, 68
TAN, 68
TAND, 68
Fonction exponentielle en virgule flottante, 395
Fonction mathématique
ABS, 68
ARCCOS, 68
ARCSIN, 68
ARCTAN, 68
argument, 68
COS, 68
COSD, 68
EXP, 68
FIX, 68
FLOAT, 68
limites d'argument, 69
LN, 68
LOG, 68
réseau d'équations, 68
saisie dans un réseau d'équations, 68
SIN, 68
SIND, 68
SQRT, 68
TAN, 68
TAND, 68
Fonctions maître, 745
Fonctions mathématiques étendues, 273
Fonctions Modbus, 1157
FOUT, 539
FTOI, 545

G

Générateur d'intervalle de temps, 677
Gestionnaire central d'alarme, 845
Groupes d'instructions, 39
ASCII Functions, 41
Coils, Contacts and Interconnects, 52
Counters/Timers, 42
Fast I/O Instructions, 43
Loadable DX, 44
Math, 45
Matrix, 47
Miscellaneous, 48
Move, 49
présentation, 40
Skips/Specials, 50
Special, 51

H

HLTH, 615

I

IBKR, 637
IBKW, 641
ICMP, 645
ID, 651
IE, 655
IMIO, 659
IMOD, 667
Installation des instructions chargeables DX, 107
Instruction
    bobines, contacts et interconnexions, 97
Instruction du module d'interruption, 667
Intégration d'entrées à intervalles définis, 879
Interconnexions, 97
Interfaces de commande séquentielle, 1023
Interruption activée, 655
Interruption désactivée, 651
ITMR, 677
ITOF, 685

J

JSR, 689

L

LAB, 693
Lecture, 991
Lecture de mémoire étendue, 1191
Lecture indirecte de bloc, 637
Liaison horizontale
    réseau d'équations, 56
Limite de la vitesse de variation de VP, 893
Limite de Vp, 889
Linéarisation par interpolation, 897
LIRE
    MSTR, 757
LL984
    AD16, 115
    ADD, 119
    AND, 123
    BCD, 129
    BLKM, 133
    BLKT, 137
    BMDI, 141
    Bobines, contacts et interconnexions, 97
    BROT, 145
    CHS, 167
    CKSM, 175
    CMPR, 181
    COMP, 195
    DCTR, 217
    DIOH, 221
    DIV, 231
    DLOG, 237
    DRUM, 253
    DV16, 259
    EMTH, 273
    EMTH-ADDDP, 281
    EMTH-ADDFP, 287
    EMTH-ADDIF, 291
    EMTH-ANLOG, 295
    EMTH-ARCOS, 301
    EMTH-ARSIN, 307
    EMTH-ARTAN, 313
    EMTH-CHSIN, 319
    EMTH-CMPFP, 325
    EMTH-CMPIF, 331
    EMTH-CNVDR, 337
    EMTH-CNVFI, 343
    EMTH-CNVIF, 349
    EMTH-CNVRD, 355
    EMTH-COS, 361
    EMTH-DIVDP, 367
    EMTH-DIVFI, 373
    EMTH-DIVFP, 379
    EMTH-DIVIF, 385
    EMTH-ERLOG, 389
    EMTH-EXP, 395
    EMTH-LNFP, 401
    EMTH-LOG, 407
    EMTH-LOGFP, 413
    EMTH-MULDP, 419

EMTH-MULFP, 425
EMTH-MULIF, 429
EMTH-PI, 435
EMTH-POW, 441
EMTH-SINE, 447
EMTH-SQRFP, 453
EMTH-SQRT, 459
EMTH-SQRTP, 465
EMTH-SUBDP, 471
EMTH-SUBFI, 477
EMTH-SUBFP, 481
EMTH-SUBIF, 487
EMTH-TAN, 493
ESI, 499
EUCA, 519
FIN, 535
FOUT, 539
FTOI, 545
HLTH, 615
IBKR, 637
IBKW, 641
ICMP, 645
ID, 651
IE, 655
IMIO, 659
IMOD, 667
ITMR, 677
ITOF, 685
JSR, 689
LAB, 693
LOAD, 697
MAP 3, 701
MBIT, 719
MBUS, 725
mise en forme de messages pour les opérations ASCII READ/WRIT, 89
MRTM, 735
MSTR, 745
MU16, 795
MUL, 799
NBIT, 805
NCBT, 809
NOBT, 813
NOL, 817
OR, 825
PCFL, 831
PCFL-AIN, 839
PCFL-ALARM, 845
PCFL-AOUT, 851
PCFL-AVER, 855
PCFL-CALC, 861
PCFL-DELAY, 867
PCFL-EQN, 873
PCFL-INTEG, 879
PCFL-KPID, 883
PCFL-LIMIT, 889
PCFL-LIMV, 893
PCFL-LKUP, 897
PCFL-LLAG, 903
PCFL-MODE, 907
PCFL-ONOFF, 911
PCFL-PI, 917
PCFL-PID, 923
PCFL-RAMP, 929
PCFL-RATE, 935
PCFL-RATIO, 939
PCFL-RMPLN, 945
PCFL-SEL, 949
PCFL-TOTAL, 955
PEER, 961
PID2, 965
R --> T, 981
RBIT, 987
READ, 991
Régulation en boucle fermée/Valeurs

analogiques, 75
RET, 997
SAVE, 1015
SBIT, 1019
SCIF, 1023
SENS, 1029
SRCH, 1043
STAT, 1049
SU16, 1077
SUB, 1081
T-->R, 1093
T-->T, 1099
TBLK, 1123
Temporisation T.01, 1105
Temporisation T0.1, 1109
Temporisation T1.0, 1113
Temporisation T1MS, 1117
TEST, 1129
traitement des interruptions, 103
traitement des sous-programmes, 105
UCTR, 1133
WRIT, 1149
XMRD, 1191, 1197
XOR, 1203
LN, 68
LOAD, 697
Loadable DX
CHS, 167
DRUM, 253
ESI, 499
EUCA, 519
HLTH, 615
ICMP, 645
installation, 107
MAP 3, 701
MBUS, 725
MRTM, 735
NOL, 817
PEER, 961
LOG, 68
Logarithme à base 10 en virgule flottante, 413
Logarithme de base 10, 407
Logarithme népérien en virgule flottante, 401
LSB (octet de poids faible)
données de constante, 60

M

MAP 3, 701
Math
AD16, 115
ADD, 119
BCD, 129
DIV, 231
DV16, 259
FTOI, 545
ITOF, 685
MU16, 795
MUL, 799
SU16, 1077
SUB, 1081
TEST, 1129
Matrices des états et des historiques, 615
Matrix
AND, 123
BROT, 145
CMPR, 181
COMP, 195
MBIT, 719
NBIT, 805
NCBT, 809, 813
OR, 825
RBIT, 987
SBIT, 1019
SENS, 1029
XOR, 1203
MBIT, 719
MBUS, 725

Miscellaneous
CKSM, 175
DLOG, 237
EMTH, 273
EMTH-ADDDP, 281
EMTH-ADDFP, 287
EMTH-ADDIF, 291
EMTH-ANLOG, 295
EMTH-ARCOS, 301
EMTH-ARSIN, 307
EMTH-ARTAN, 313
EMTH-CHSIN, 319
EMTH-CMPFP, 325
EMTH-CMPIF, 331
EMTH-CNVDR, 337
EMTH-CNVFI, 343
EMTH-CNVIF, 349
EMTH-CNVRD, 355
EMTH-COS, 361
EMTH-DIVDP, 367
EMTH-DIVFI, 373
EMTH-DIVFP, 379
EMTH-DIVIF, 385
EMTH-ERLOG, 389
EMTH-EXP, 395
EMTH-LNFP, 401
EMTH-LOG, 407
EMTH-LOGFP, 413
EMTH-MULDP, 419
EMTH-MULFP, 425
EMTH-MULIF, 429
EMTH-PI, 435
EMTH-POW, 441
EMTH-SINE, 447
EMTH-SQRFP, 453
EMTH-SQRT, 459
EMTH-SQRTP, 465
EMTH-SUBDP, 471
EMTH-SUBFI, 477
EMTH-SUBFP, 481
EMTH-SUBIF, 487
EMTH-TAN, 493
LOAD, 697
MSTR, 745
SAVE, 1015
SCIF, 1023
XMRD, 1191, 1197
Mise à 0 de bit, 987
Mise à 1 de bit, 1019
Mise en forme de messages, 89
Modbus Plus
MSTR, 745
Modifier bit, 719
Module d'option réseau pour Lonworks, 817
Module de transfert à registres multiples, 735
Mot
nombre maximum dans un réseau d'équations, 57
Mots utilisés
données de constante, 60
équation mathématique, 59
Move
FIN, 535
FOUT, 539
IBKR, 637
IBKW, 641
R --> T, 981
SRCH, 1043
T-->R, 1093
T-->T, 1099
TBLK, 1123
Moyenne des entrées pondérées, 855
MRTM, 735
MSTR, 745
Codes d'erreur CTE pour Ethernet SY/MAX et Ethernet TCP/IP, 793
codes d'erreur Ethernet TCP/IP, 790
Codes d'erreur Modbus Plus et Ethernet SY/MAX, 786
Codes d'erreur spécifiques à SY/MAX, 788
écrire CTE (Table d'extension de configuration), 776
Ecrire données globales, 763
Etat de diffusion des E/S, 769
Lire CTE (Table d'extension de

configuration), 774
Lire données globales, 764
Lire statistiques distantes, 765
Lire statistiques locales, 759
Réinitialiser module optionnel, 772
Statistiques du réseau Ethernet TCP/IP, 784
statistiques du réseau Modbus Plus, 778
Supprimer statistiques distantes, 767
Supprimer statistiques locales, 761
MU16, 795
MUL, 799
Multiplication, 799
Multiplication de valeurs 16 bits, 795
Multiplication en double précision, 419
Multiplication en virgule flottante, 425
Multiplication Entier x Nombre en virgule flottante, 429

N

NBIT, 805
NCBT, 809
NOBT, 813
NOL, 817
Nombre à virgule flottante divisée par entier, 373
Notation algébrique
réseau d'équation, 53
Notation exponentielle
équation mathématique, 61
Notation infixée
réseau d'équations, 54

O

Opérateur arithmétique, 62
Opérateur conditionnel, 62
Opérateur d'affectation, 62
Opérateur d'élévation à une puissance, 62
Opérateur de manipulation de bits, 62
Opérateur de relation, 62
Opérateur mathématique
réseau d'équation, 53
Opérateur monadique, 62
Opération mathématique
opérateur arithmétique, 62
opérateur conditionnel, 62
opérateur d'affectation, 62
opérateur d'élévation à une puissance, 62
opérateur de manipulation de bits, 62
opérateur de relation, 62
opérateur monadique, 62
parenthèses, 62
réseau d'équations, 62
Opérations READ/WRIT, 89
OR, 825
OU exclusif, 1203
Ou Logique, 825
Ouvert horizontal
réseau d'équations, 56

P

Parenthèses
couches imbriquées, 53
imbriquées, 66
réseau d'équation, 53
saisie dans un réseau d'équations, 67
utilisation dans un réseau d'équations, 66
Parenthèses imbriquées
réseau d'équations, 66
PCFL, 831
PCFL-AIN, 839
PCFL-ALARM, 845
PCFL-AOUT, 851
PCFL-AVER, 855
PCFL-CALC, 861
PCFL-DELAY, 867
PCFL-EQN, 873
PCFL-INTEG, 879
PCFL-KPID, 883
PCFL-LIMIT, 889
PCFL-LIMV, 893
PCFL-LKUP, 897
PCFL-LLAG, 903
PCFL-MODE, 907
PCFL-ONOFF, 911
PCFL-PI, 917

PCFL-PID, 923
PCFL-RAMP, 929
PCFL-RATE, 935
PCFL-RATIO, 939
PCFL-RMPLN, 945
PCFL-SEL, 949
PCFL-TOTAL, 955
PEER, 961
Performance de référence
    réseau d'équation, 73
PI ISA non interactif, 917
PID ISA complet non interactif, 883
PID2, 965
Pile premier entré, 535
Pile premier sorti, 539
Priorité
    réseau d'équations, 65
Priorité des opérateurs
    réseau d'équations, 65
Prise en charge du module ESI, 499
Proportionnelle–intégrale–dérivée, 965

R

R --> T, 981
Racine carrée, 459
Racine carrée en virgule flottante, 453
Racine carrée procédé, 465
Rampe logarithmique vers consigne, 945
Rampe vers la consigne à pente constante, 929
RBIT, 987
READ, 991
Recherche, 1043
Redondance d'UC
    CHS, 167
Référence TOR
    équation mathématique, 59
Registre 3x ou 4x
    saisie dans une équation mathématique, 60
Registre de comparaison, 181
Registre vers table, 981
Registres utilisés
    équation mathématique, 59
Régulateur de rapport 4 positions, 939
Régulation, 832
Régulation en boucle fermée, 75
Réseau d'équation
    constante, 53
    conversions de données, 70
    différences d'arrondi, 72
    éditeur logique, 53
    expression conditionnelle, 53
    expression logique, 53
    expressions groupées dans des couches de parenthèses imbriquées, 53
    notation algébrique, 53
    opérateur mathématique, 53
    performance de référence, 73
    variable, 53
    vue d'ensemble, 53
Réseau d'équations
    ABS, 68
    ARCCOS, 68
    ARCSIN, 68
    ARCTAN, 68
    argument, 68
    bit de sortie, 55
    contact de validation, 56
    contenu, 56
    COS, 68
    COSD, 68
    création, 54
    EXP, 68
    expression algébrique, 56
    expression conditionnelle, 64
    expression unique, 64
    FIX, 68
    FLOAT, 68
    fonction mathématique, 68
    limites de l'argument, 69
    LN, 68
    LOG, 68
    mathématique, 58
    mots utilisés, 57
    opérateur arithmétique, 62
    opérateur conditionnel, 62
    opérateur d'affectation, 62
    opérateur d'élévation à une puissance,

62
opérateur de manipulation de bits, 62
opérateur de relation, 62
opérateur monadique, 62
opération mathématique, 62
parenthèses, 62, 66
parenthèses imbriquées, 66
priorité des opérateurs, 65
résultat, 56
saisie d'une fonction, 68
saisie de parenthèses, 67
SIN, 68
SIND, 68
SQRT, 68
taille, 57
TAN, 68
TAND, 68
Résultat
réseau d'équations, 56
RET, 997
Retour d'un sous–programme, 997
Rotation de bit, 145

S

Santé des E/S distribuées, 221
Saut vers sous-programme, 689
Sauvegarde mémoire flash, 1015
SAVE, 1015
SBIT, 1019
SCIF, 1023
Sélection des entrées, 949
SENS, 1029
Séquenceur à tambour, 253
SIN, 68
SIND, 68
Sinus en virgule flottante d'un angle (en radians), 447
Skips/Specials
JSR, 689
LAB, 693
RET, 997
Sortie analogique, 851
Sous-fonction EMTH
EMTH-ADDDP, 281
EMTH-ADDFP, 287, 291
EMTH-ANLOG, 295
EMTH-ARCOS, 301
EMTH-ARSIN, 307
EMTH-ARTAN, 313
EMTH-CHSIN, 319
EMTH-CMPFP, 325
EMTH-CMPIF, 331
EMTH-CNVDR, 337
EMTH-CNVFI, 343
EMTH-CNVIF, 349
EMTH-CNVRD, 355
EMTH-COS, 361
EMTH-DIVDP, 367
EMTH-DIVFI, 373
EMTH-DIVFP, 379
EMTH-DIVIF, 385
EMTH-ERLOG, 389
EMTH-EXP, 395
EMTH-LNFP, 401
EMTH-LOG, 407
EMTH-LOGFP, 413
EMTH-MULDP, 419
EMTH-MULFP, 425
EMTH-MULIF, 429
EMTH-PI, 435
EMTH-POW, 441
EMTH-SINE, 447
EMTH-SQRFP, 453
EMTH-SQRT, 459
EMTH-SQRTP, 465
EMTH-SUBDP, 471
EMTH-SUBFI, 477
EMTH-SUBFP, 481
EMTH-SUBIF, 487
EMTH-TAN, 493

Sous-fonction PCFL
PCFL-AIN, 839
PCFL-ALARM, 845
PCFL-AOUT, 851
PCFL-AVER, 855
PCFL-CALC, 861
PCFL-DELAY, 867
PCFL-EQN, 873
PCFL-INTEG, 879
PCFL-KPID, 883
PCFL-LIMIT, 889
PCFL-LIMV, 893
PCFL-LKUP, 897
PCFL-LLAG, 903
PCFL-MODE, 907
PCFL-ONOFF, 911
PCFL-PI, 917
PCFL-PID, 923
PCFL-RAMP, 929
PCFL-RATE, 935
PCFL-RATIO, 939
PCFL-RMPLN, 945
PCFL-SEL, 949
PCFL-TOTAL, 955
Sous-fonctions PCFL
généralités, 77
Soustraction, 1081
Soustraction de valeurs 16 bits, 1077
Soustraction double précision, 471
Soustraction en virgule flottante, 481
Soustraction entier – nombre en virgule flottante, 487
Soustraction virgule flottante – entier, 477
Special
DIOH, 221
PCFL, 831
PCFL-, 851
PCFL-AIN, 839
PCFL-ALARM, 845
PCFL-AVER, 855
PCFL-CALC, 861
PCFL-DELAY, 867
PCFL-EQN, 873
PCFL-KPID, 883
PCFL-LIMIT, 889
PCFL-LIMV, 893
PCFL-LKUP, 897
PCFL-LLAG, 903
PCFL-MODE, 907
PCFL-ONOFF, 911
PCFL-PI, 917
PCFL-PID, 923
PCFL-RAMP, 929
PCFL-RATE, 935
PCFL-RATIO, 939
PCFL-RMPLN, 945
PCFL-SEL, 949
PCFL-TOTAL, 955
PCPCFL-INTEGFL, 879
PID2, 965
STAT, 1049
SQRT, 68
SRCH, 1043
STAT, 1049
Statistiques du réseau Ethernet TCP/IP
MSTR, 784
Statistiques du réseau Modbus Plus
MSTR, 778
SU16, 1077
SUB, 1081

T

T-->R, 1093
T-->T, 1099
Table vers Bloc, 1123
Table vers registre, 1093
Table vers table, 1099
TAN, 68

TAND, 68
Tangente en virgule flottante d'un angle (en radians), 493
TBLK, 1123
Temporisation à la milliseconde, 1117
Temporisation à la seconde, 1113
Temporisation au centième de seconde, 1105
Temporisation au dixième de seconde, 1109
Temporisation T.01, 1105
Temporisation T0.1, 1109
Temporisation T1.0, 1113
Temporisation T1MS, 1117
TEST, 1129
Totalisateur de flux, 955
Traitement des interruptions, 103
Traitement des sous-programmes, 105
Transmission MAP, 701
Transmission MBUS, 725
Transmission PEER, 961
Types de données mixtes
    réseau d'équation, 70
Types de valeurs et de données
    équation mathématique, 58

U

UCTR, 1133

V

Valeur binaire en valeur binaire codée, 129
Valeurs analogiques, 75
Valeurs ON/OFF de la plage neutre, 911
Variable
    réseau d'équation, 53
Variable à virgule flottante, 59
Variable de consigne, 76
Variable de procédé, 76
Variable long non signée (32 bits), 59
Variable long signée (32 bits), 59
Variable non signée 16 bits, 59
Variable signée 16 bits, 59

W

WRIT, 1149

X

XMRD, 1191, 1197
XOR, 1203