Testing VAMP relay as Modbus slave

1 Test system configuration

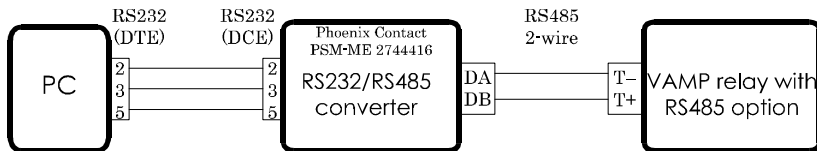The testing was carried out with the configuration outlined in.

Figure 1. Test system. A PC connected with an RS232 cable to a Phoenix Contact 232/485 converter. The converter is connected with a twisted pair to a VAMP relay.

PC:

• At least one serial port
• ModScan32 Modbus master program or similar

Modbus master PC program configuration:

• Bit rate 19200, even parity
• Address of the slave: 5
• Scanned register area: Holding registers
• First register number to be scanned: 2001
• Number of register to be scanned: 18

RS232/RS485 converter:

• Manufacturer: Phoenix Contact.
• Type PSM-ME-RS232/RS485-P Ord.-No.: 2744416
• Supply: 24 VDC
• Termination ON (There are only two devices in the system. Thus both can have their termination on)
• RTS self-controlled
• Two-wire mode

VAMP relay:

• Protocol: Modbus slave
• Bit rate: 19200, even parity
• Address: 5
• Two-wire mode

- Termination ON (There are only two devices in the system. Thus both can have their termination on)
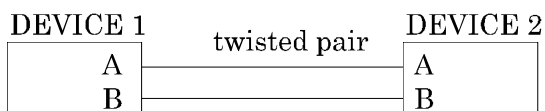
2 RS485 polarity

In RS485 a bit value is equal to the voltage polarity between two wires, A and B. E.g. when the voltage from A to B is positive the bit is "1" and when the voltage is negative the bit is "0".
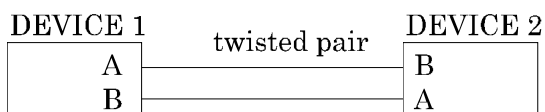
This differs from RS232 where the voltage in a signal line is defined against a common ground level. In RS485 the both lines can have a voltage against ground but the bit value depends only of the polarity of voltage difference between the two signal lines A and B.

Unfortunately the names A and B (or often + and -) and the voltage polarity for bit values "1" and "0" varies depending of the manufacturer of the RS 485 device.

Devices with the same naming convention are connected together like this:

```
DEVICE 1                           DEVICE 2
          ┌──────twisted pair──────┐
   A ──────                        ────── A
   B ──────                        ────── B
```

And devices with different naming convention are connected together like this:

```
DEVICE 1                           DEVICE 2
          ┌──────twisted pair──────┐
   A ──────                        ────── B
   B ──────                        ────── A
```

3 Checking the RS485 polarity

A simple way to check the polarity is to measure the voltage between A and B (often called as + and -) when the RS485 transmitter is in idle state i.e. ready to send asynchronous data i.e. sending continuously bit value "1".

Measurement results:

VAMP relay: Voltage from T+ to T- was +0.58 V (termination on). Thus a positive voltage corresponds to bit value "1".

Phoenix Contact RS232/RS485 converter: Voltage from D(A) to D(B) was –0.94 V (termination on). Thus a negative voltage corresponds to bit value "1".

Conclusion:

The two measured voltages had different signs. Thus these two devices have different naming convention and the differential wires must be cross-coupled between these two device types. This is just what has been done in figure 1: A is connected to – and B is connected to +.

Examples of other RS485 converters:

Entrelec ILPH RS232/RS485-422 24 VDC converter has similar polarity than VAMP relays.

Westermo MD-42 DC RS232/RS485 converter has opposite polarity than VAMP relays.

4 Modbus communication

As Modbus master I used a program called ModScan. It is a simple Windows program, which is able continuously read data from at least one slave. Figure 2 is a snap shot of the program running. The comments on the right are not part of ModScan. They were added afterwards to explain the item.
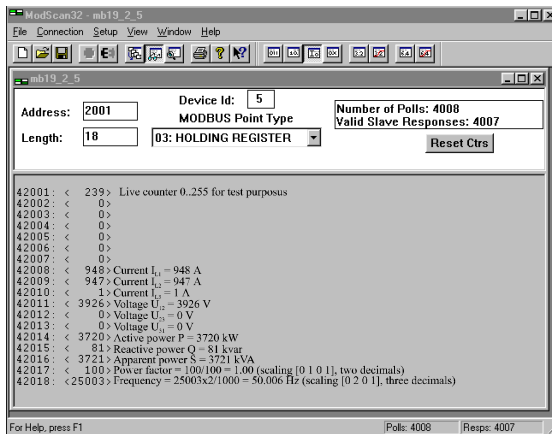

Figure 2. Screen shot of a Modbus master program scanning a VAMP relay as slave 5, holding registers 2001 ... 2018.

5 Modbus message in detail

Figure 3 is screen shot of the actual bytes of Modbus messages when reading just one register: 2001 from holding area.

The Modbus holding register 2001 of VAMP devices is an eight bit counter (0, 1, 2, .. 255, 0, 1, 2, ..) which is incremented once a second. It can be used to test the Modbus communication.
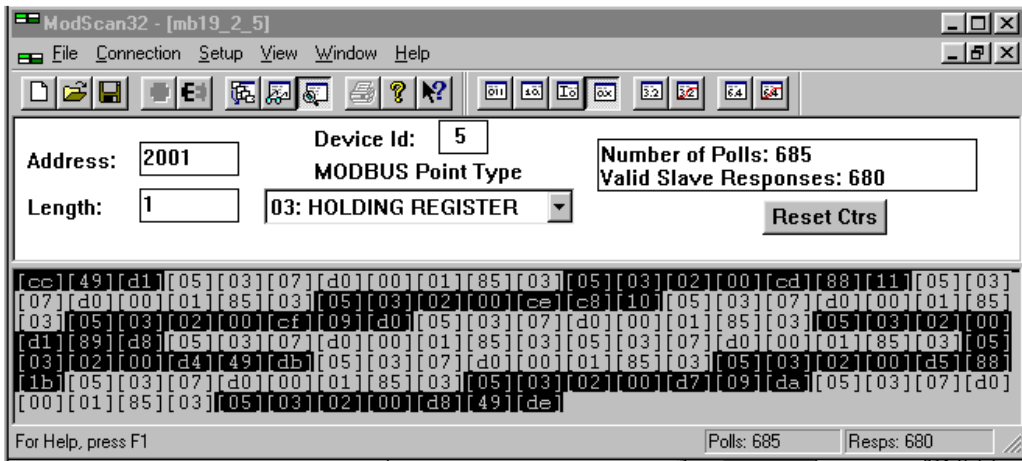
Figure 3. Screen shot of a Modbus master program scanning a VAMP relay live counter at holding register 2001. The first complete dialog on the screen starts from the 4th byte. Master's enquiry is black on white and eight bytes long: [05] [03] [07] [0d] [00] [01] [85] [03]. The response from slave is white on black and seven bytes long [05] [03] [02] [00][cd][88][11].

Let's analyze the three first complete Modbus dialogs in Figure 3 in detail:

CC

49D1   Last three bytes of a previous
        slave response.

Master's enquiry 1

05   Slave address i.e. device
        identifier (Device Id).

03   Function code: Read holding
        registers

07DO   First register (Address) to be
        read 07D016 = 200110 – 1.

        (The offset "–1" is Modbus
        feature.)

0001   Number of registers to be
        read

8503   Cyclic redundancy check
        (CRC).

Slave's response 1

05	Slave address i.e. device identifier (Device Id).
03	Function code: Read holding registers
02	Number of data bytes
00CD	Data bytes: Value of holding register 2001 = $00CD_{16}$ = 205_{10}.
8811	Cyclic redundancy check (CRC).

Master's enquiry 2

05	Slave address i.e. device identifier (Device Id).
03	Function code: Read holding registers
07DO	First register (Address) to be read $07D0_{16} = 2001_{10} - 1$. (The offset "–1" is Modbus feature.)
0001	Number of registers to be read
8503	Cyclic redundancy check (CRC).

Slave's response 2

05	Slave address i.e. device identifier (Device Id).
03	Function code: Read holding registers
02	Number of data bytes

00CE Data bytes: Value of holding register 2001 =

$00CE_{16} = 206_{10}$.

C810 Cyclic redundancy check (CRC).

Master's enquiry 3

05 Slave address i.e. device identifier (Device Id).

03 Function code: Read holding registers

07DO First register (Address) to be read $07D0_{16} = 2001_{10} - 1$.

(The offset "–1" is Modbus feature.)

0001 Number of registers to be read

8503 Cyclic redundancy check (CRC).

Slave's response 3

05 Slave address i.e. device identifier (Device Id).

03 Function code: Read holding registers

02 Number of data bytes

00CF Data bytes : Value of holding register 2001 = $00CF_{16} = 207_{10}$.

09D0 Cyclic redundancy check (CRC).

The scanning interval was one second. That is why the responses from the slave are 206, 207, 208, ... i.e. a value which increments once a second. This is just what the holding register 2001 is; a counter 0,1,2, ... 254,255,0,1,2, ... updated once in every second.