Modbus extension and test program

This extension is built on my extension SerialOTG and uses the same library for serial communication. There are a few add on to enable single 16 bit register read and write to a device using ModbusRTU and ModbusTCP protocol. It can be used for ModbusRTU over serial, ModbusRTU bridged over WiFi or Bluetooth, and ModbusTCP over WiFi.

Methods

Init, Open, Close: see SerialOTG documentation

ret = ReadSingleInputRegister (slave, addr)   using Function code 0x04
slave slave number 1-255
addr (0..65535)

ret = ReadSingleHoldingRegister (slave, addr)   using Function code 0x03
slave slave number 1-255
addr (0..65535)

ret = ReadDoubleHoldingRegister (slave, addr)   using Function code 0x03
Read two 16bit registers as one 32bit register.
slave slave number 1-255
addr (0..65535)

ret = WriteSingleHoldingRegister (slave, addr, data)   using Function code 0x06
slave slave number 1-255
addr (0..65535)
data (0..65535)
ret <0 error

ret = ResponseSingle()
(get response from single register read or write)
ret = response. <0 error, 0..65535 response from read

Error codes:
-1 No serial open
-2 TCP format error or no response
-3 RTU format error or no response
-4 RTU crc error
-5 Unknown function code or nr of bytes error

Protocol(prot)
prot=0 RTU, prot=1 TCP

To build other messages:
ReadByte() WriteByte(n)   see SerialOTG documentation.

b=HiByte(n)   Return hi byte of an int

b=LoByte(n)   Return low byte of an int

crc=CRC16Seed() Return start value for CRC calculation

crc=NewCRC(byte,crc)   Add byte to crc, return new crc. Do this for every byte in message.
Note: Add Lobyte(crc), HiByte(crc) to end of RTU message.

How to use
Init, open, select baud parity etc if not the default values.

Send a request to read or write a register.
Wait for the response. (A fixed time is easiest to implement or wait until you get a message. Can be tested with Available())
Get ResponseSingle()

Test program Modbus
Simple application to write, read and read with address increment.
The test program uses decimal notation.

Uses ReadSingleHoldingReg, ReadSingleInputReg, WriteSingleHoldingReg and ReadDoubleHoldingReg.
InputReg and HoldingReg are often the same data area.

Some PLC:s number the registers from 0 and some from 1. Ex: Instead of dec 4097 you should use dec 4096 (Hex 1000).

44097 means read input register 4097 using function code 4 (16bit word).
The test program does not accept this format. Use address 4097 and FC4 instead.

Note:
FC03 and FC04 can specify read multiple registers but are used with nr registers =1 (or =2 for ReadDouble) in this extension.
The use of transaction identifier is not implemented for Modbus TCP. You have to wait for the response or timeout, before sending a new request.