

Bilaga C

Skriptkod

“MODBUS.LUA”

```
require("socket")
require("string")
function modbusInit()

    sendStringW = ""
    length = 16

    Buffer[1] = string.char( 0 )           -- Transaction identifier
    Buffer[2] = string.char( 0 )
    Buffer[3] = string.char( 0 )           -- Protocol identifier
    Buffer[4] = string.char( 0 )
    Buffer[5] = string.char( 0 )           -- Message length
    Buffer[6] = string.char( 7+length*2 )
    Buffer[7] = string.char( 0 )           -- Unit identifier
    Buffer[8] = string.char( 16 )          -- Function code
    Buffer[9] = string.char( 4 )           -- Address1
    Buffer[10] = string.char( 0 )          -- Address2
    Buffer[11] = string.char( 0 )          -- Word count: Length1
    Buffer[12] = string.char( length )     -- Length2
    Buffer[13] = string.char( length*2 )   -- Byte count(Length * 2)

    Buffer0 = 0                            -- Transaction identifier
    Buffer1 = 0
    Buffer2 = 0                            -- Protocol identifier
    Buffer3 = 0
    Buffer4 = 0                            -- Message length
    Buffer5 = 6
    Buffer6 = 0                            -- Unit identifier
    Buffer7 = 3                            -- Function code
    Buffer8 = 0                            -- Address1
    Buffer9 = 0                            -- Address2
    Buffer10 = 0                           -- Word count: Length1
    Buffer11 = 16                          -- Length2
    sendString = string.char( Buffer0, Buffer1, Buffer2, Buffer3, Buffer4, Buffer5, Buffer6,
    Buffer7, Buffer8, Buffer9, Buffer10, Buffer11 )

end

function modbusReadMultiple()

sock:send(sendString)    -- send the command read multiple registers to the modbus slave

    Str, ErrStr = sock:receive('*b',41)   -- reads the respond from the modbus slave
    if Str then
        Str = string.sub(Str, 12)         -- take out the modbus slave register data.
    else
        print("Error, no receive...")
    end
end
```

```

    Lua_Write_PdBuff(Str)          -- writes the modbus slave data to the ABCC
end

function modbusWriteMultiple()

    receivebuff = Lua_Read_PdBuff() -- get 32 byte data from ABCC read buffer
    for i=1,32,1 do
        Buffer[13+i] = string.char( receivebuff[i])    -- adding the received value to the
                                                       modbus message
    end

    sendStringW = table.concat( Buffer )                -- transform from table to string
    sock:send( sendStringW )                          -- sending message to modbus slave

    receivebuff = nil
    Str, ErrStr = sock:receive('*b',12)               -- reads the respond from the modbus slave
    if Str then
        Str = string.sub(Str, 10, 10)                -- check if ok
        if not string.byte(Str) == 16 then
            print("write error")
        end
    else
        print("Error, no receive...")
    end

end

end

Buffer = {}
local Addr = "10.10.20.75"                          --adres to the Modbus slave we want to connect to
local Port = 502                                     --modbus port
local SckHnd, ErrStr, Str
modbusInit()
sock, err = socket.tcp()                             --creates a TCP socket
SckHnd, ErrStr = sock:connect(Addr, Port, 0, 0)      --attempts to connect to the modbus
slave

if SckHnd then
    while 1 do
        modbusReadMultiple()
        modbusWriteMultiple()
    end
end
end

```