

/*

MgsModbus.h - an Arduino library for a Modbus TCP master and slave.
V-0.1.1 Copyright (C) 2013 Marco Gerritse
written and tested with Arduino 1.0

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

For this library the following library is used as start point:

- [1] Mudbus.h - an Arduino library for a Modbus TCP slave.
Copyright (C) 2011 Dee Wykoff
- [2] Function codes 15 & 16 by Martin Pettersson

The following references are used to write this library:

- [3] Open Modbus/Tcp specification, Release 1.0, 29 March 1999
By Andy Swales, Schneider Electric
- [4] Modbus application protocol specification V1.1b3, 26 april 202
From <http://www.modbus.org>

External software used for testing:

- [5] modpoll - www.modbusdriver.com/modpoll.html
- [6] ananas - www.tuomio.fi/ananas
- [7] mod_rssim - www.plcsimulator.org
- [8] modbus master - www.cableone.net/mblansett/

This library use a single block of memory for all modbus data (mbData[] array). ↗

The
same data can be reached via several modbus functions, either via a 16 bit access or via an access bit. The length of MbData must at least 1.

For the master the following modbus functions are implemented: 1, 2, 3, 4, 5, 6, ↗
15, 16

For the slave the following modbus functions are implemented: 1, 2, 3, 4, 5, 6, ↗
15, 16

The internal and external addresses are 0 (zero) based

```

V-0.1.1 2013-06-02
bugfix

V-0.1.0 2013-03-02
initinal version
*/

#include "Arduino.h"

#include <SPI.h>
#include <Ethernet2.h>

#ifndef MgsModbus_h
#define MgsModbus_h

#define MbDataLen 30 // length of the MdData array
#define MB_PORT 502

enum MB_FC {
    MB_FC_NONE                = 0,
    MB_FC_READ_COILS          = 1,
    MB_FC_READ_DISCRETE_INPUT = 2,
    MB_FC_READ_REGISTERS      = 3,
    MB_FC_READ_INPUT_REGISTER = 4,
    MB_FC_WRITE_COIL          = 5,
    MB_FC_WRITE_REGISTER      = 6,
    MB_FC_WRITE_MULTIPLE_COILS = 15,
    MB_FC_WRITE_MULTIPLE_REGISTERS = 16
};

class MgsModbus
{
public:
    // general
    MgsModbus();
    word MbData[MbDataLen]; // memory block that holds all the modbus user data
    boolean GetBit(word Number);
    boolean SetBit(word Number,boolean Data); // returns true when the number is in
        the MbData range
    // modbus master
    bool Req(MB_FC FC, word Ref, word Count, word Pos, byte IP[4]);
    void MbmRun();
    IPAddress remSlaveIP;
    // modbus slave
    void MbsRun();
    word GetDataLen();

```

```
private:
    // general
    MB_FC SetFC(int fc);
    // modbus master
    uint8_t MbmByteArray[260]; // send and recieve buffer
    MB_FC MbmFC;
    int MbmCounter;
    void MbmProcess();
    word MbmPos;
    word MbmBitCount;
    //modbus slave
    uint8_t MbsByteArray[260]; // send and recieve buffer
    MB_FC MbsFC;
};

#endif
```