Arduino library MODBUS slave

History

MODBUS Slave driver and supporting classes for implementation on the Arduino AVR platform allowing it to communicate with a wide variety of HMI programs and other SCADA devices.

The library contains various register type containers, a device container, and a MODBUS slave driver.

Example

```
#include <modbus.h>
#include <modbusDevice.h>
#include <modbusRegBank.h>
#include <modbusSlave.h>

/*
This example code shows a quick and dirty way to get an
arduino to talk to a modbus master device with a
device ID of 1 at 9600 baud.
*/

//Setup the brewtrollers register bank
//All of the data accumulated will be stored here
modbusDevice regBank;
//Create the modbus slave protocol handler
modbusSlave slave;

void setup()
{

//Assign the modbus device ID.
  regBank.setId(1);

/*
```

```
modbus registers follow the following format
00001-09999  Digital Outputs, A master device can read and write to these
registers
10001-19999  Digital Inputs, A master device can only read the values from
these registers
30001-39999  Analog Inputs, A master device can only read the values from
these registers
40001-49999  Analog Outputs, A master device can read and write to these
registers

Analog values are 16 bit unsigned words stored with a range of 0-32767
Digital values are stored as bytes, a zero value is OFF and any nonzer
value is ON

It is best to configure registers of like type into contiguous blocks.
this
allows for more efficient register lookup and and reduces the number of
messages
required by the master to retrieve the data
*/

//Add Digital Output registers 00001-00016 to the register bank
  regBank.add(1);
  regBank.add(2);
  regBank.add(3);
  regBank.add(4);
  regBank.add(5);
  regBank.add(6);
  regBank.add(7);
  regBank.add(8);
  regBank.add(9);
  regBank.add(10);
  regBank.add(11);
  regBank.add(12);
  regBank.add(13);
  regBank.add(14);
  regBank.add(15);
  regBank.add(16);

//Add Digital Input registers 10001-10008 to the register bank
  regBank.add(10001);
  regBank.add(10002);
  regBank.add(10003);
  regBank.add(10004);
  regBank.add(10005);
  regBank.add(10006);
  regBank.add(10007);
  regBank.add(10008);

//Add Analog Input registers 30001-10010 to the register bank
  regBank.add(30001);
  regBank.add(30002);
  regBank.add(30003);
  regBank.add(30004);
  regBank.add(30005);
  regBank.add(30006);
  regBank.add(30007);
  regBank.add(30008);
  regBank.add(30009);
  regBank.add(30010);

//Add Analog Output registers 40001-40020 to the register bank
  regBank.add(40001);
  regBank.add(40002);
  regBank.add(40003);
  regBank.add(40004);
  regBank.add(40005);
  regBank.add(40006);
  regBank.add(40007);
  regBank.add(40008);
  regBank.add(40009);
  regBank.add(40010);
  regBank.add(40011);
  regBank.add(40012);
  regBank.add(40013);
  regBank.add(40014);
  regBank.add(40015);
  regBank.add(40016);
  regBank.add(40017);
  regBank.add(40018);
  regBank.add(40019);
  regBank.add(40020);

/*
Assign the modbus device object to the protocol handler
This is where the protocol handler will look to read and write
register data.  Currently, a modbus slave protocol handler may
only have one device assigned to it.
*/
  slave._device = &regBank;
```

```
// Initialize the serial port for coms at 9600 baud
  slave.setBaud(9600);
}

void loop()
{
//put some data into the registers
  regBank.set(1, 1);
  regBank.set(2, 1);
  regBank.set(3, 0);
  regBank.set(4, 1);
  regBank.set(5, 1);
  regBank.set(6, 0);
  regBank.set(7, 1);
  regBank.set(8, 0);

  regBank.set(10001, 1);
  regBank.set(10002, 1);
  regBank.set(10003, 1);
  regBank.set(10004, 1);
  regBank.set(10005, 0);
  regBank.set(10006, 0);
  regBank.set(10007, 0);
  regBank.set(10008, 0);


  regBank.set(30001,1);
  regBank.set(30002,2);
  regBank.set(30003,3);
  regBank.set(30004,4);
  regBank.set(30005,5);
  regBank.set(30006,6);
  regBank.set(30007,7);
  regBank.set(30008,8);
  regBank.set(30009,9);
  regBank.set(30010,10);

  regBank.set(40001,1);
  regBank.set(40002,2);
  regBank.set(40003,2);
  regBank.set(40004,4);
  regBank.set(40005,5);
  regBank.set(40006,6);
  regBank.set(40007,7);
  regBank.set(40008,8);
  regBank.set(40009,9);
  regBank.set(40010,10);

 while(1)
  {
    //put a random number into registers 1, 10001, 30001 and 40001
    regBank.set(1, (byte) random(0, 2));
    regBank.set(10001, (byte) random(0, 2));
    regBank.set(30001, (word) random(0, 32767));
    regBank.set(40001, (word) random(0, 32767));

     slave.run();
  }
}
```