```
/*
    Modbus slave simple example

    Control and Read Arduino I/Os using Modbus serial connection.

    This sketch show how to use the callback vector for reading and
    controleing Arduino I/Os.

    * Controls digital output pins as modbus coils.
    * Reads digital inputs state as discreet inputs.
    * Reads analog inputs as input registers.

    The circuit: ( see: ./extras/ModbusSetch.pdf )
    * An Arduino.
    * 2 x LEDs, with 220 ohm resistors in series.
    * A switch connected to a digital input pin.
    * A potentiometer connected to an analog input pin.
    * A RS485 module (Optional) connected to RX/TX and a digital control pin.

    Created 8 12 2015
    By Yaacov Zamir

    https://github.com/yaacov/ArduinoModbusSlave

*/

#include <ModbusSlave.h>

/* slave id = 1, rs485 control-pin = 8, baud = 9600
 */
#define SLAVE_ID 1
#define CTRL_PIN 8
#define BAUDRATE 9600

/**
 *  Modbus object declaration
 */
Modbus slave(SLAVE_ID, CTRL_PIN);

void setup() {
    /* set some pins for output
     */
    pinMode(10, INPUT);
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);

    /* register handler functions
     * into the modbus slave callback vector.
     */
    slave.cbVector[CB_WRITE_COILS] = writeDigitalOut;
    slave.cbVector[CB_READ_DISCRETE_INPUTS] = readDigitalIn;
```

```
53      slave.cbVector[CB_READ_INPUT_REGISTERS] = readAnalogIn;
54
55      // set Serial and slave at baud 9600.
56      Serial.begin( BAUDRATE );
57      slave.begin( BAUDRATE );
58  }
59
60  void loop() {
61      /* listen for modbus commands con serial port
62       *
63       * on a request, handle the request.
64       * if the request has a user handler function registered in cbVector
65       * call the user handler function.
66       */
67      slave.poll();
68  }
69
70  /**
71   * Handle Force Single Coil (FC=05) and Force Multiple Coils (FC=15)
72   * set digital output pins (coils).
73   */
74  uint8_t writeDigitalOut(uint8_t fc, uint16_t address, uint16_t length) {
75      // set digital pin state(s).
76      for (int i = 0; i < length; i++) {
77          digitalWrite(address + i, slave.readCoilFromBuffer(i));
78      }
79
80      return STATUS_OK;
81  }
82
83  /**
84   * Handel Read Input Status (FC=02)
85   * write back the values from digital in pins (input status).
86   *
87   * handler functions must return void and take:
88   *      uint8_t  fc - function code
89   *      uint16_t address - first register/coil address
90   *      uint16_t length/status - length of data / coil status
91   */
92  uint8_t readDigitalIn(uint8_t fc, uint16_t address, uint16_t length) {
93      // read digital input
94      for (int i = 0; i < length; i++) {
95          slave.writeCoilToBuffer(i, digitalRead(address + i));
96      }
97
98      return STATUS_OK;
99  }
100
101 /**
102  * Handel Read Input Registers (FC=04)
103  * write back the values from analog in pins (input registers).
104  */
```

```
105 uint8_t readAnalogIn(uint8_t fc, uint16_t address, uint16_t length) {
106     // read analog input
107     for (int i = 0; i < length; i++) {
108         slave.writeRegisterToBuffer(i, analogRead(address + i));
109     }
110 
111     return STATUS_OK;
112 }
113 
114 
```