Stroud EnviroDIY Mayfly Data Logger and Modbus RS-485 Sensors

Peter Wise[1,2], Steve Hicks[2], Anthony Aufdenkampe[2]

1. Goshen College, Goshen Indiana 2. Stroud Water Research Center, Avondale Pennsylvania

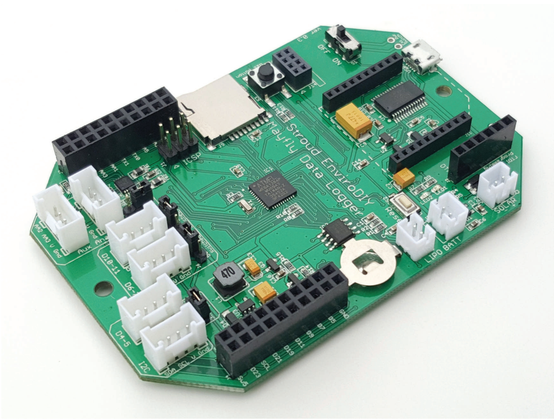
Introduction

Scientists and educators around the world have begun to use the the Arduino open source electronics platform to build a variety of do-it-yourself (DIY) devices. Our new Arduino-compatible EnviroDIY Mayfly datalogger board (http://envirodiy.org/mayfly/) expands the potential for DIY environmental monitoring by connecting to a wide variety of analog and digital sensors, streaming data live from remote locations, and not requiring significant modification or additional components in order to function.

Here we present a set of new low-cost software and hardware approaches to expand the EnviroDIY Mayfly logger's digital sensor capabilities to the wide variety of industrial sensors that operate using Modbus RS-485 communication protocol. The Mayfly's RS-485 compatibility will make environmental monitoring more feasible, allowing more researchers and scientists from a broad range of backgrounds to collect high quality, high resolution data.

The EnviroDIY webpage (http://envirodiy.org/) has more information, tutorials, and forums for people to share ideas, get help, and present their gadgets.

Mayfly Data Logger

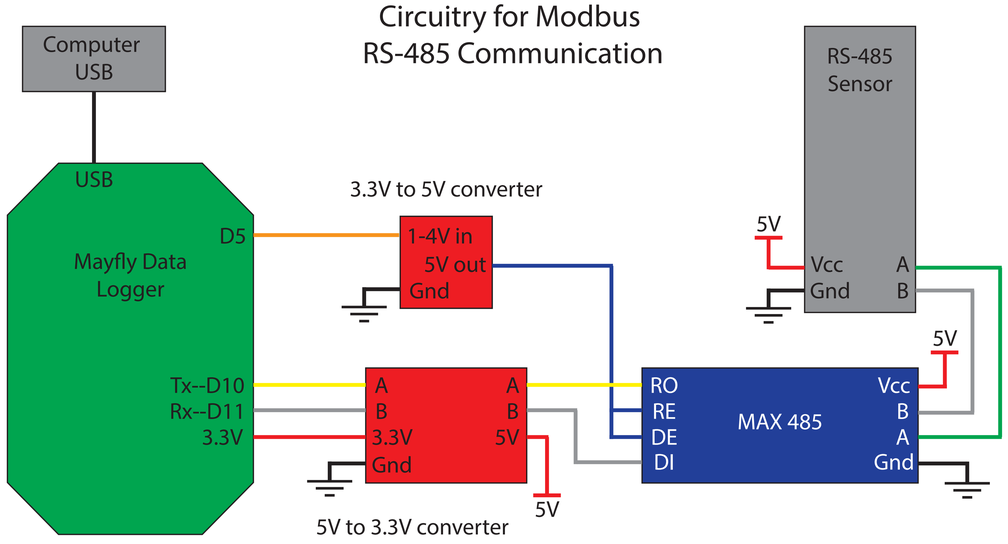


Yosemitech Modbus RS-485 Sensors

4 Electrode Conductivity

Self-Cleaning Optical Turbidity

Circuitry for Modbus RS-485 Communication

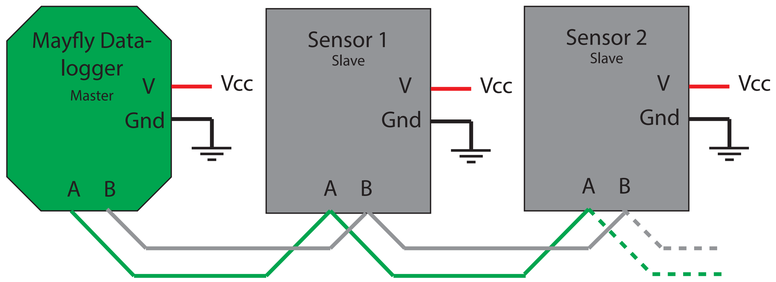


Compatibilty

Incompatibility is caused by a difference in how the Mayfly communicates and how the RS-485 sensors communicate. The Mayfly expects to receive messages in a format specified by transistor to transistor logic (TTL). Modbus RS-485 sensors expect to receive messages in RS-485 format. The Mayfly send messages in TTL format and the sensors send messages in RS-485 format. While both devices may be sending perfectly valid information in their messages, the other device is incapable of properly reading the other's message. This results in a message received that is complete nonsense.

Both TTL and Modbus RS-485 are physical layers. Physical layers are ways of specifying how bits will be transmitted, grouped, encoded, and decoded. Physical layers also define limitations of the network such as cable length, number of devices capable of connecting, baud rate, and network topology. The Modbus RS-485 physical layer restricts maximum cable length to 1200 meters and 32 devices. The devices must be connected in line in a sequence (daisy chaining). RS-485 communicates data over two wires, A and B via a differential voltage.

Modbus RS-485 Network



A Modbus RS-485 network has a master and 1 or more slaves. Each conversation is started by the master. The master sends out a message to one of the slaves. Each slave will check the message address against its own. If the addresses do not match, the slave ignores the message. If the address does match, the slave hears the rest of the incoming message and responds accordingly.

Circuitry

The Mayfly board operates at 3.3V, but the RS-485 sensors require a 5V input. This 5V requirement is met by the 3.3V to 5V converters in the circuit. The D5 output gives 3.3V, but the 5V booster is needed in order to confidently write to DE and RE (data enable and receive enable). A jumper is used to provide a constant 5V supply off the Grove connector next to D10 and D11. Lastly, the incoming 5V signals from the sensor need to be stepped down to 3.3V with another converter so that the Mayfly can read incoming messages.

The MAX485 allows only 1-way communication at a time. If the Mayfly data logger wants to send a message to a sensor, digital pin 5 needs to be turned high. Turning D5 high turns DE (data enable) off and RE (receive enable) on, allowing the Mayfly to transmit a message to the sensor. Writing D5 low turns RE off and DE on, allowing the sensor to send messages to the Mayfly data logger.

Pins D10 and D11 are software serial pins. They are not hardware dedicated serial pins, but using the SoftwareSerial library, pins D10 and D11 are written to be the Tx and Rx pin respectively.

Messaging

Every Modbus message has the same 4 basic elements. Each of these elements is communicated in sequential order. After each message is sent, Modbus requires a 3.5 character length delay. This silence signals the end of the message.

Modbus RS-485 specifies many function codes, but Yosemitech sensors use only two function codes. Read registers code 03 and write registers code 10 (0x03 and 0x10 in hexadecimal). The read registers command will be used to get data from the sensors. The write registers command will be used to tell the sensor to collect data.

Request Frame

Definition	Device Address	Function Code	Start Address		Number of Registers		CRC	
Byte	0	1	2	3	4	5	6	7
Value	0x01	0x03	0x26	0x00	0x00	0x05	0x8E	0x81

Response Frame

Definition	Device Address	Function Code	Number of Bytes	Register Value				CRC	
Byte	0	1	2	3-6	7-10	11	12	13	14
Value	0x01	0x03	0x05	17.625	17.625	0x00	0x00	0xc7	0x33

Register Values for Temperature and Conductivity

Temperature (3-6)				Conductivity (7-10)				Error Flag (11)	Reserved (12)
0x00	0x00	0x8D	0x41	0x00	0x00	0x8D	0x41	0x00	0x00

Goals

While we have not yet achieved our goal of creating a working network of RS-485 sensors and Mayfly dataloggers, we have laid the foundation and gathered information necessary to design a working solution. Currently, we have designed a solution for the hardware, but we lack the software capabilities needed to communicate between the Mayfly and the RS-485 sensors. Our future efforts will be focused in creating the software required to facilitate successful communication.