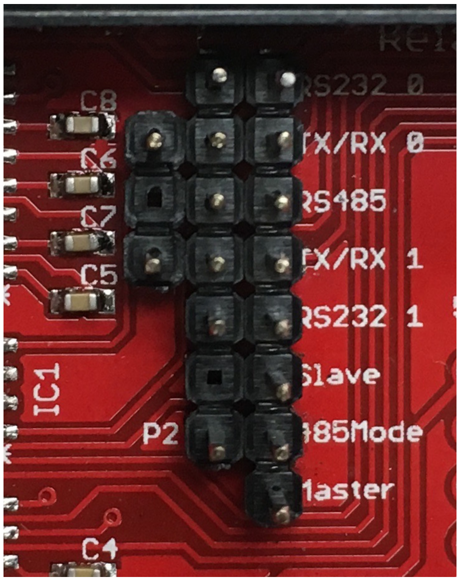
MercuryShield Quick Start Guide

The MercuryShield is designed to be a swiss army knife of communication. It gives the ability to use RS232, RS485, or both. You can also easily connect a 40pin header to transfer up to 16 pins and 16 ground connections using a layout compatible with a typical pinout used in I/O relay boards such as Opto22 or Grayhill. There are also 3 LEDs that can be connected via solder jumper for extra visual feedback. Here is the full breakdown of connection options:



- Serial Port 0: USB(No Connections), RS232, RS485

- Serial Port 1: RS232, RS485 (Assuming you didn't connect Port 0 to RS485)
- Serial Ports 2 & 3: RS232

- Serial Ports 0 & 1 also have a Ground pin available nearby to allow easy 3-wire connections for TTL communications.

- Be aware that if you are using an Ardunio Uno, You will only have hardware Serial Port 0. However, you could create wire jumpers from other pins to create connections for Software Serial ports.

- All Serial port connections go to two rows of .1" headers, spaced .3" apart. These are a perfect fit for Miniature Spring Terminal Blocks.

- RS485 Directional Control on Pin 2 (Or direct Master/Slave only hookup)

- 40 pin, Opto-22 or Relay Header compatible header utilizing Pins 3-12 and A0-A5

- Red, Green, and Blue LEDs on pins 9, 10, and 11, respectively. (PWM pins on both Uno and Mega)

There's also a great location to put some basic prototype circuits linked by wire jumpers to the rest of your board – there's enough room to fit a couple tactile switches and LEDs, a voltage divider, or even a small IC if you've got something else in mind.

Assembly:

Step 1: Solder Shield Headers On

If you are using an Arduino Mega, all headers can be filled. If you are using an Uno, you don't need to put in the header for pins 14-21, since they don't exist on your board. You may also decide to cut the connections to A7 and A8, as the Uno will not have them.

Step 2: Solder in Pins for Jumpers

If you'd like to use this board for lots of different projects, it's best to solder some pin headers in, pulling out the extra pins where there are no holes. You can then use standard jumpers to select which communication type to use with which port.

Note that the RS485 Control has the choice to be a Master, Slave, or bidirectional. To do bidirectional communication, jumper to the pin labeled "P2" – this is Pin 2 of the Arduino. If you write the pin High, it sets the RS485 chip to write data out. Setting the pin low will set it to receive data.

Notes on Port 0:

The Port 0 connection is shared with the USB connection on the Uno and Mega 2560. Leaving jumpers across the RS232 or 485 pins from Port 0 may cause issues with uploading your sketch via USB, so remember to take the jumpers off if you have issues uploading.

Step 3: Solder Jumpering the LEDs

On the bottom side of the Board, there are 3 solder jumpers available. These are used to tie Pin 9, 10, and 11 to the Red, Green, and Blue LEDs, respectively.

Getting Started

If you bought this board, you probably have at least a rough idea of what RS232 and RS485 are.

Example Sketches - https://github.com/McCallisterMicro/MercuryShield
There are three included examples for testing the MercuryShield.
MultiSerialMegaTest.ino – Allows you to send and receive RS232 messages from any serial port to another.
RS485Test.ino – this is a simple send/receive RS485 messages using pin 2 for control.
RelayHeaderTest.ino – steps though the relay header pins from 1-16.

Matthias Hertel's DMX Serial Library – https://github.com/mathertel/DMXSerial
Most of the time when I've used RS485, it's been in a DMX application, which is unidirectional. I use this library extensively. Matt's got some great info on RS485 in general, and good notes on this library.

If you are using an Arduino Mega, you should edit the header file so that Serial Port 1 is used instead of 0, thus freeing up the USB port for debugging. Open DMXSerial.h, and uncomment code line 50, which should read #define DMX_USE_PORT1.

Nick Gammon's RS485 Libraries - http://www.gammon.com.au/forum/?id=11428
Nick's got some Wikipedia links for RS485, but also gives some quick explanations of the setup for RS485, and has some handy libraries for if you're doing bidirectional RS485 communications.