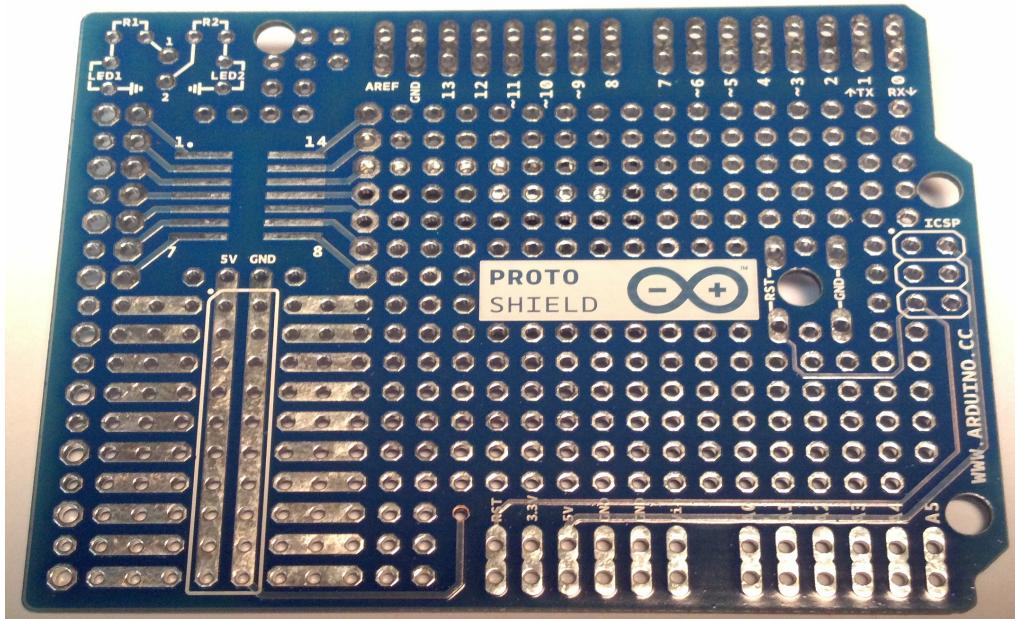


Using an Arduino protoshield for a CAN interface for OpenLCB

by Tim Hatch

As stated in the OpenLCB web site, there are several versions of CAN hardware available or under development for the Arduino. All of these use the Microchip MCP2515/MCP2551 chip set, so are compatible with the standard CAN library. This article will explain how to use the Arduino protoshield # A000024. It lends itself to an easy CAN shield for the Arduino development boards. The tutorial will lead you step by step to completion of the board for just the CAN interface. There is more to come for the Button LED consumer, producer with Blue/Gold LED, and buttons.

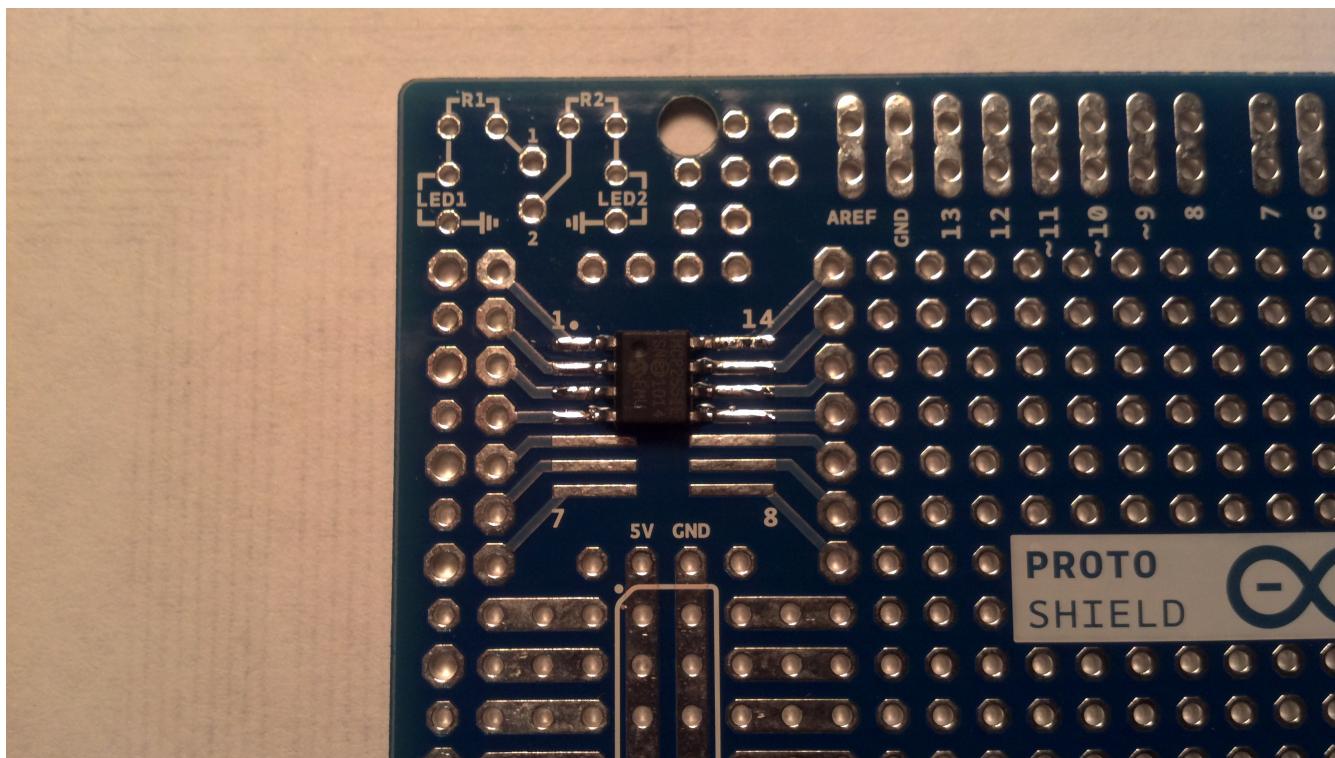


Parts list

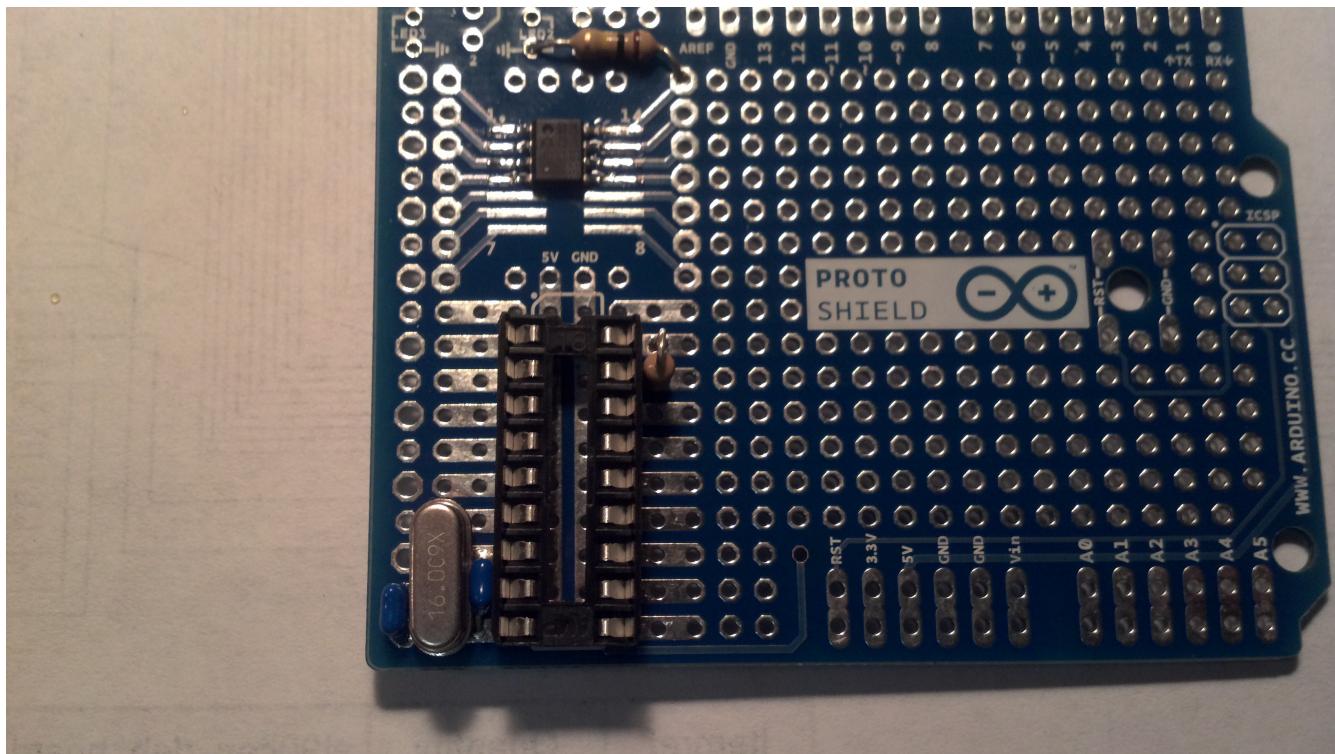
Parts for the OpenLCB Arduino CAN Shield

Quantity	Description	Mouser Part #	Digi-key Part#
2	18 pf capacitor	594-K180J15C0GF5TH5	
1	Crystal	695-HC49US-16-U	
1	MCP2551 transceiver	579-MCP2551-I/SN	
1	MCP2515 CAN	579-MCP2515-I/P	
1	100k Resistor		
1	10k Resistor		
inches	#30 black wire		K394-ND
inches	#30 red wire		K395-ND
inches	#30 blue wire		K325-ND
1	Arduino Proto Shield	782-A000024	1050-1009-ND
1	18 pin IC socket	571-1-390261-5	

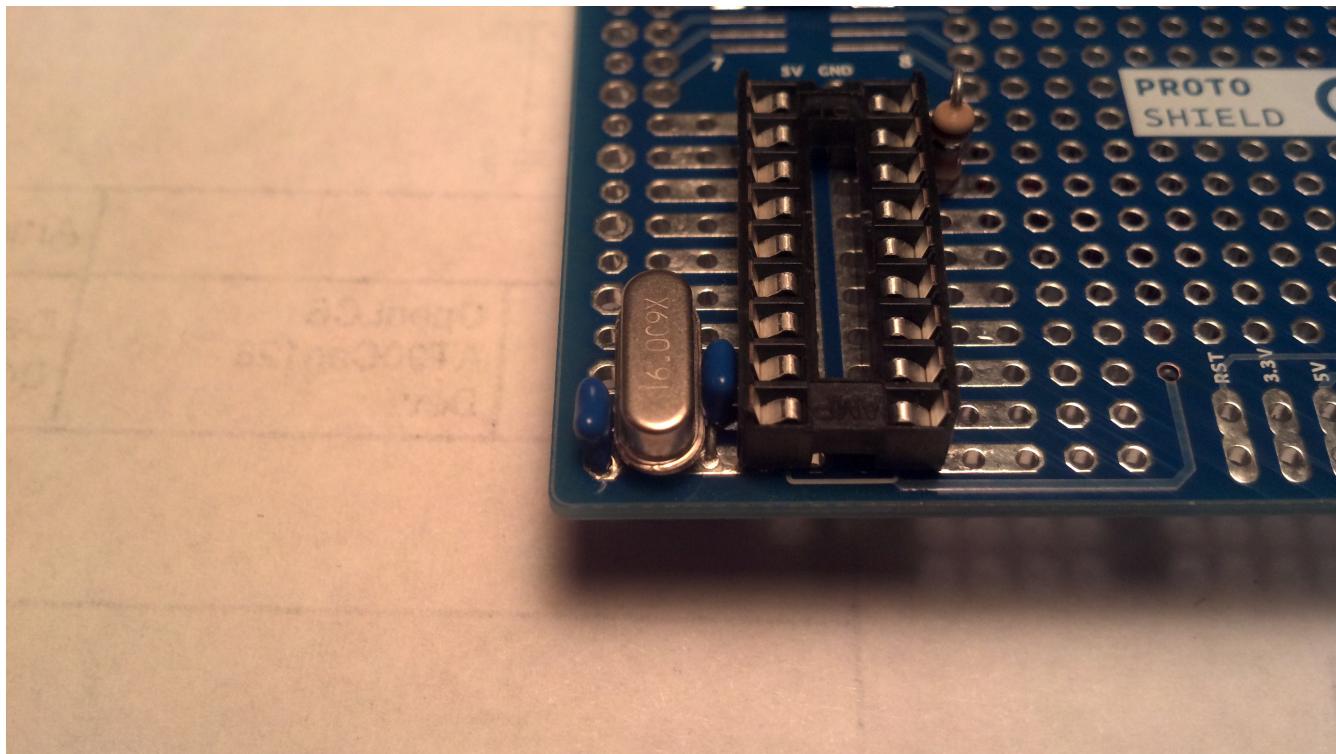
Begin with soldering the MCP2551 SOIC surface mount IC, observing the picture below.



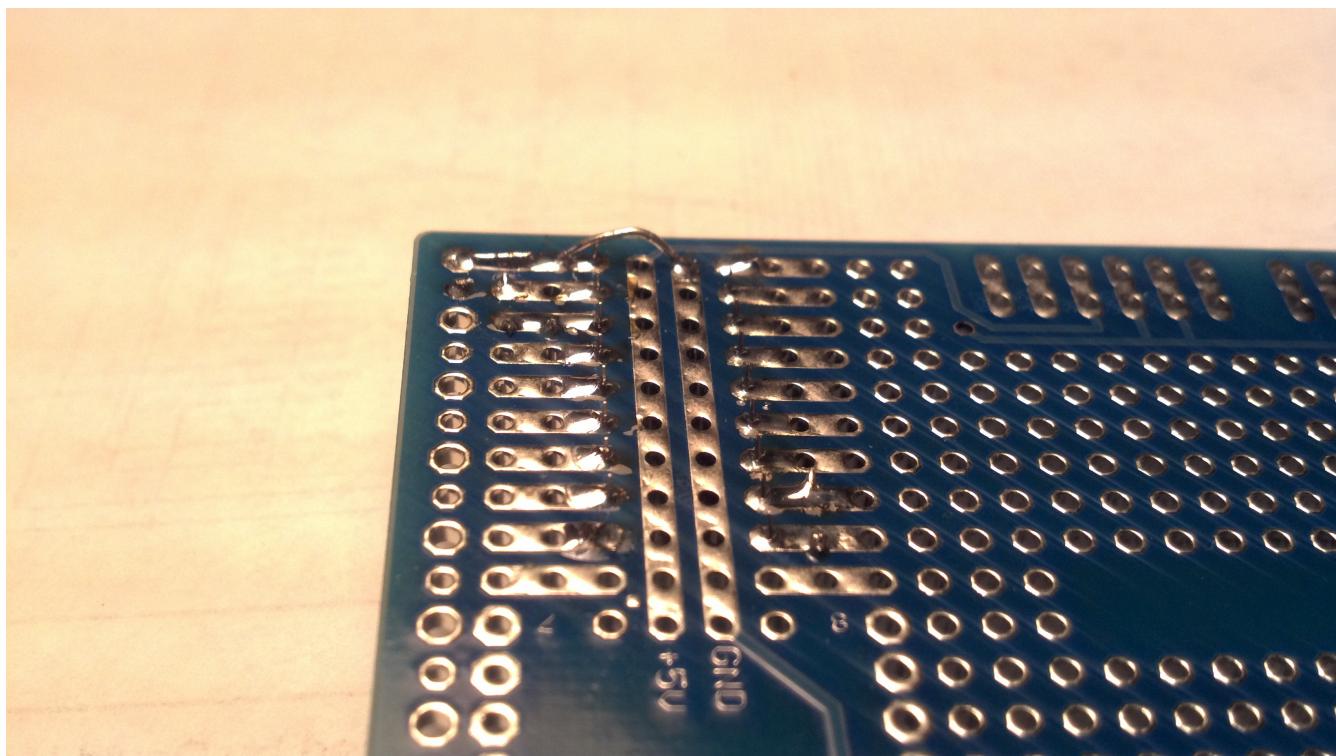
Next install the 18 pin socket for the MCP2551 IC and the 10k resistor between Pins 18 and 17. Also install the 100k pin 8 and ground for the MCP2515 IC.



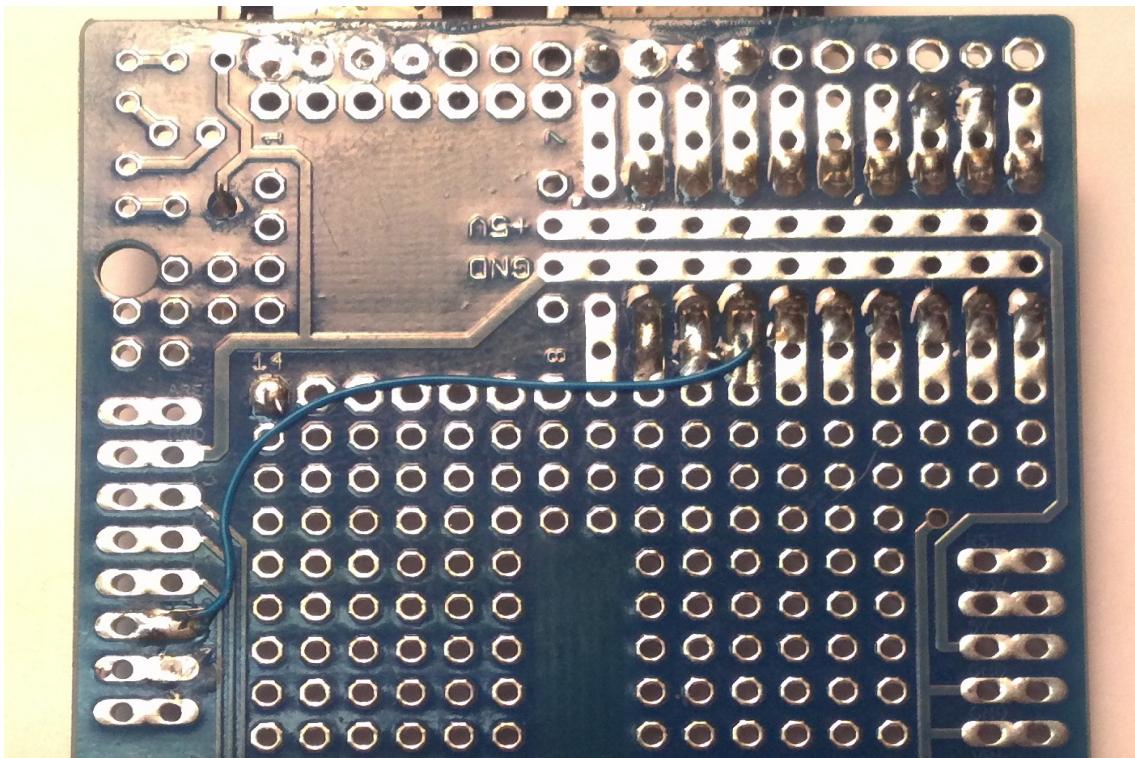
Installing the crystal and the 18 pf capacitors is a little tricky. The crystal is installed to Pins 7 and 8. The two 18 pf capacitors go between Pins 7 and 9 and Pins 8 and 9.



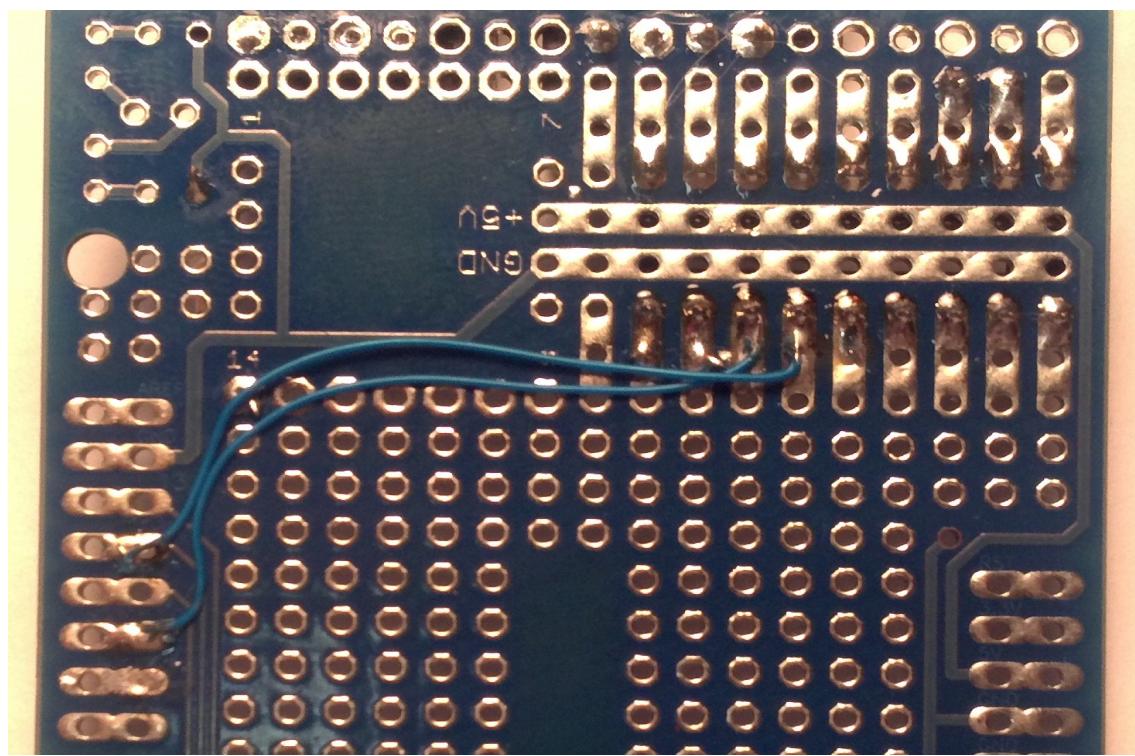
On the bottom of the board, solder a jumper wire between the ground pad and Pin 10.



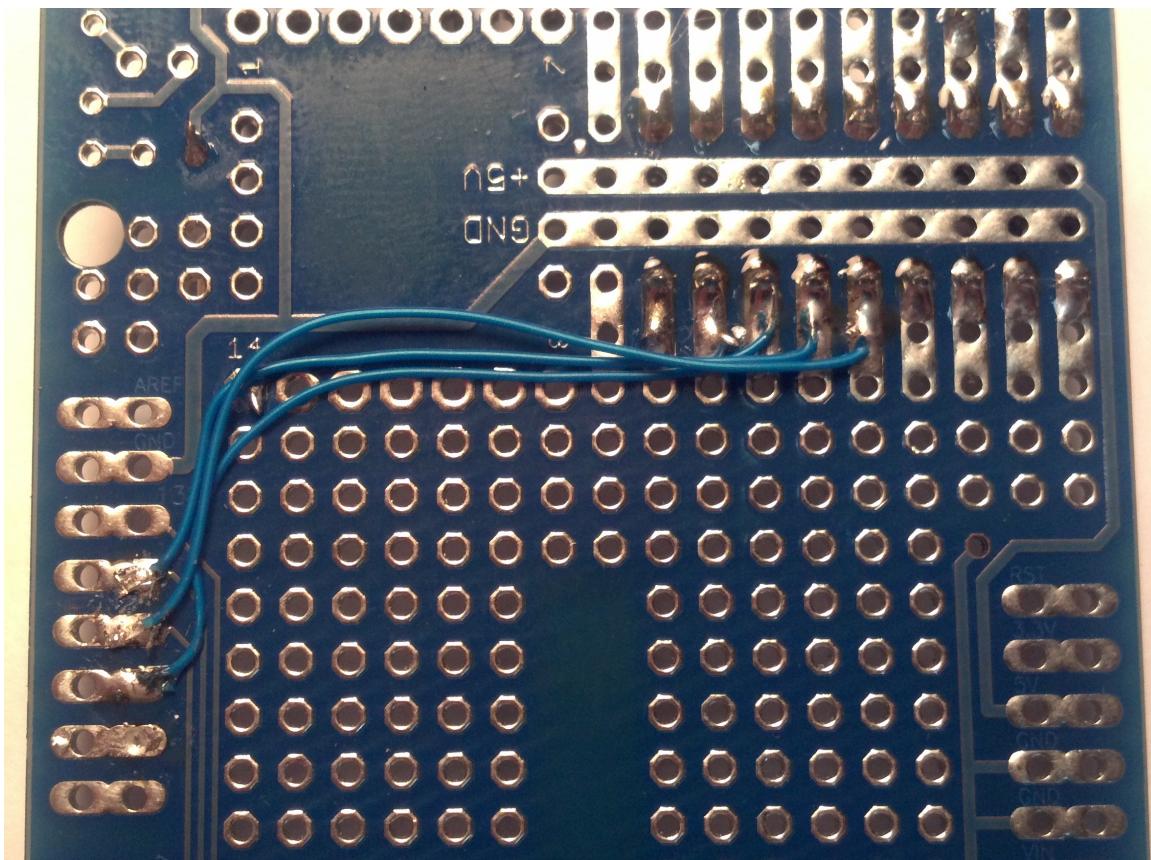
Now it is time to use the #30 wire wrap. Solder a piece of blue wire from Pin 16 of the MCP2515 to the Arduino shield Pin #10.



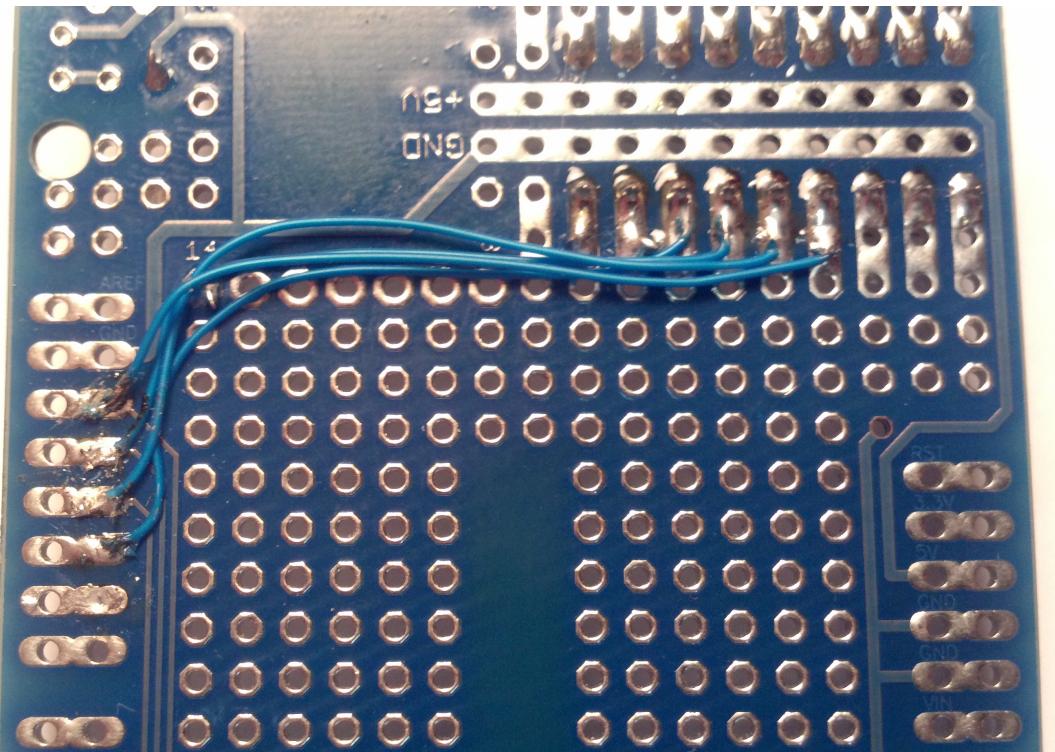
Solder a blue wire from Pin15 of the MCP2515 to the Arduino shield #12.



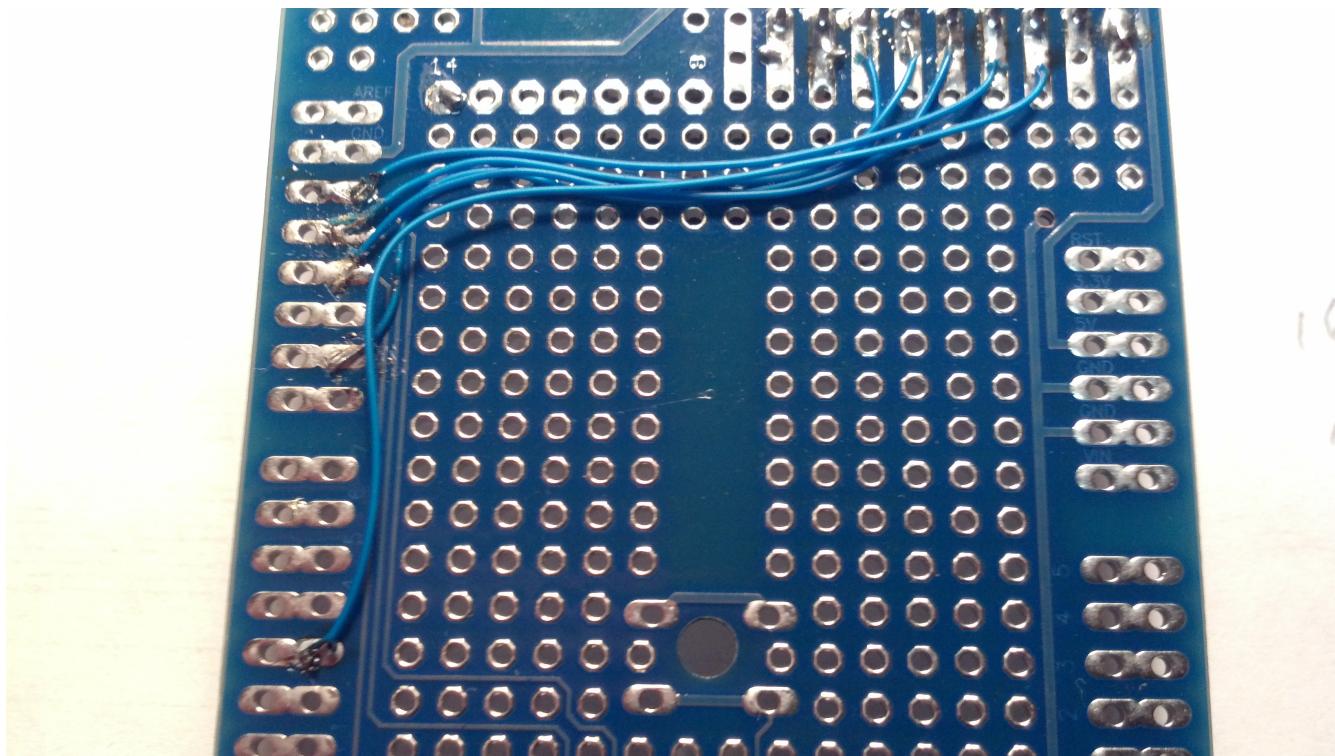
Solder Pin 14 of the MCP2515 to Arduino shield #11.



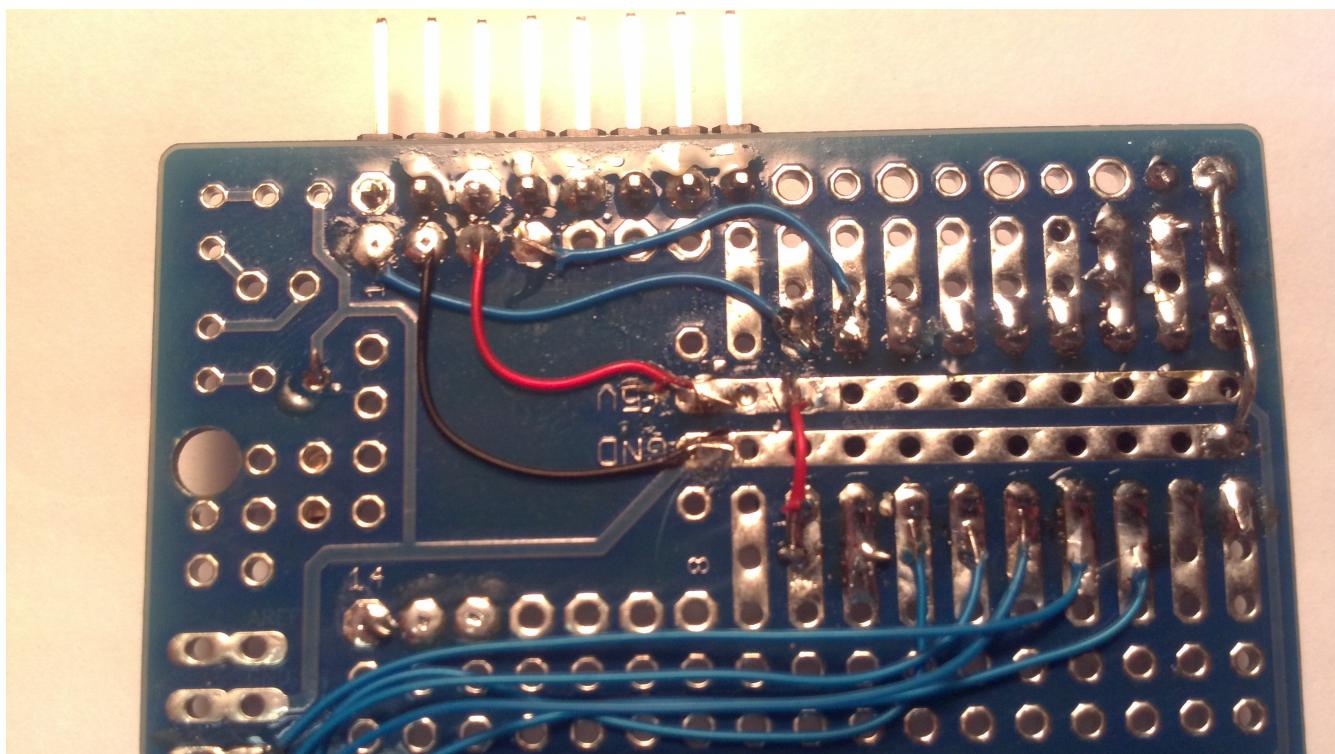
Solder Pin 13 of the MCP2515 to Arduino shield #13.



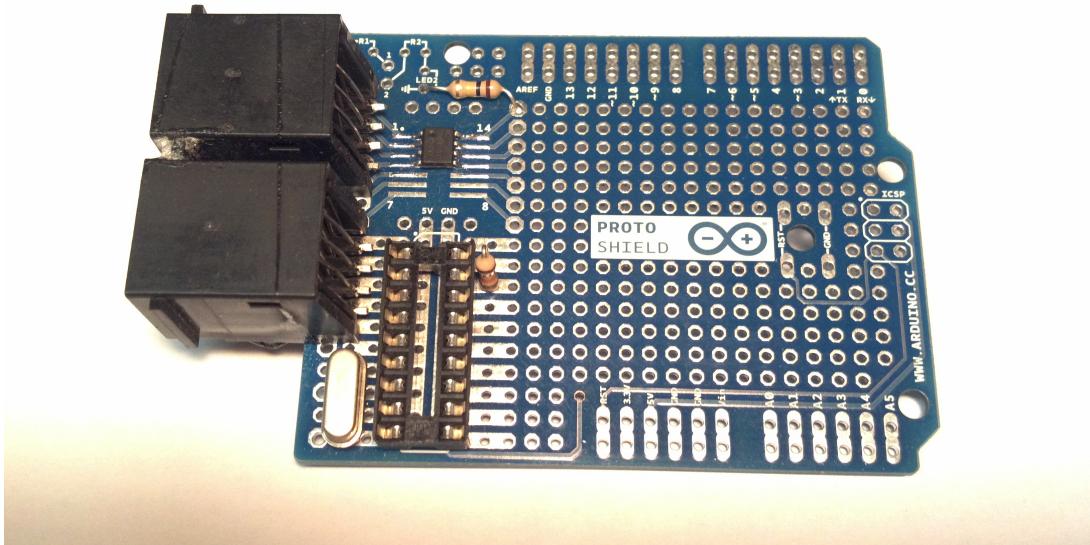
Solder Pin 12 of the MCP2515 to Arduino shield #3.



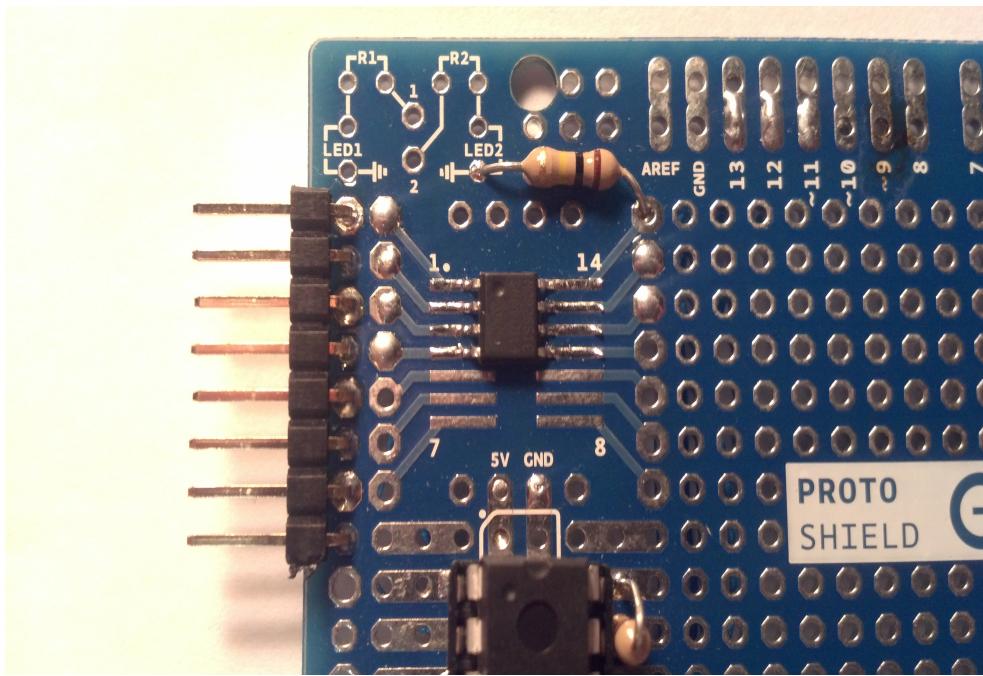
Solder blue wire from Pin 1 on the MCP2515 DIP to Pin 1 on the MCP2551 SOIC. Then solder a blue wire from Pin 2 on the MCP2515 DIP to Pin 4 on the MCP2551 SOIC. Solder a red wire on +5 v Pad to Pin 3 on the MCP2551 SOIC. Solder a red wire from Pin 1 of the MCP2515 to +5 v Pad. Solder a black wire from Pin 2 on the MCP2515 to the ground pad.



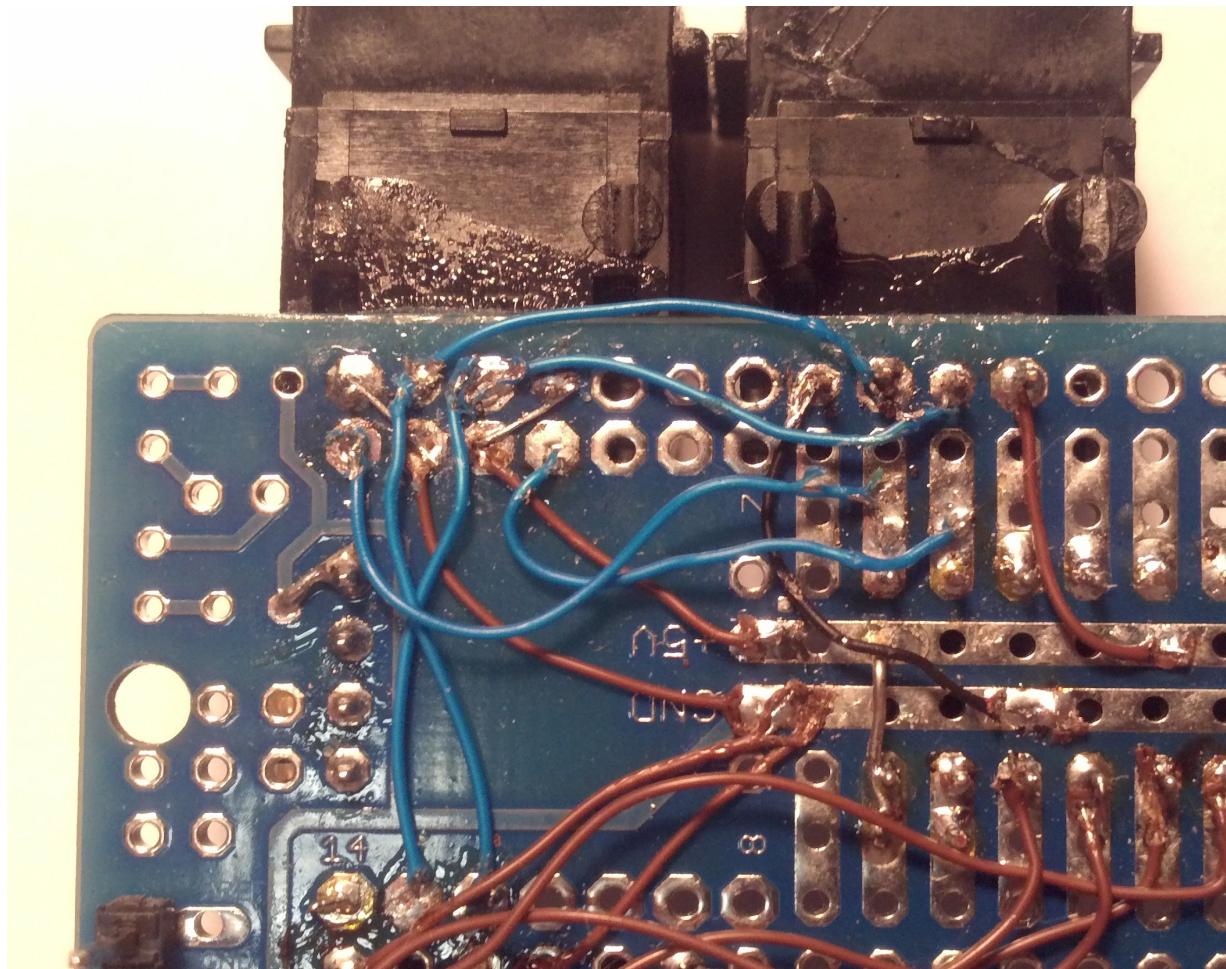
There are two ways used to connect the MCP2551 CAN transceiver to the bus. One way would be to use two RJ-45 jacks with the front 4 pins bent up and cut off, using Pins 1,3,5 and 7 for connections.



The other way is to use a right angle .1 inch header.



Connections to the jacks can be just two wires to other OpenLCB nodes just like this one. Other connections can be +5 volts and ground. The author has chosen to use the four wire approach with the CAN high, CAN low, +5 volts and ground connected to the RJ-45 jacks. This is one of the first the author constructed. A little bit of super glue has been applied to keep the jacks stable because the front half of the pins on the jack have been removed and the back four are used to connect the CAN high, CAN low to the MCP2551 SOIC and power. The tabs on the RJ-45 to the left have been removed to clear the USB port on the Arduino.

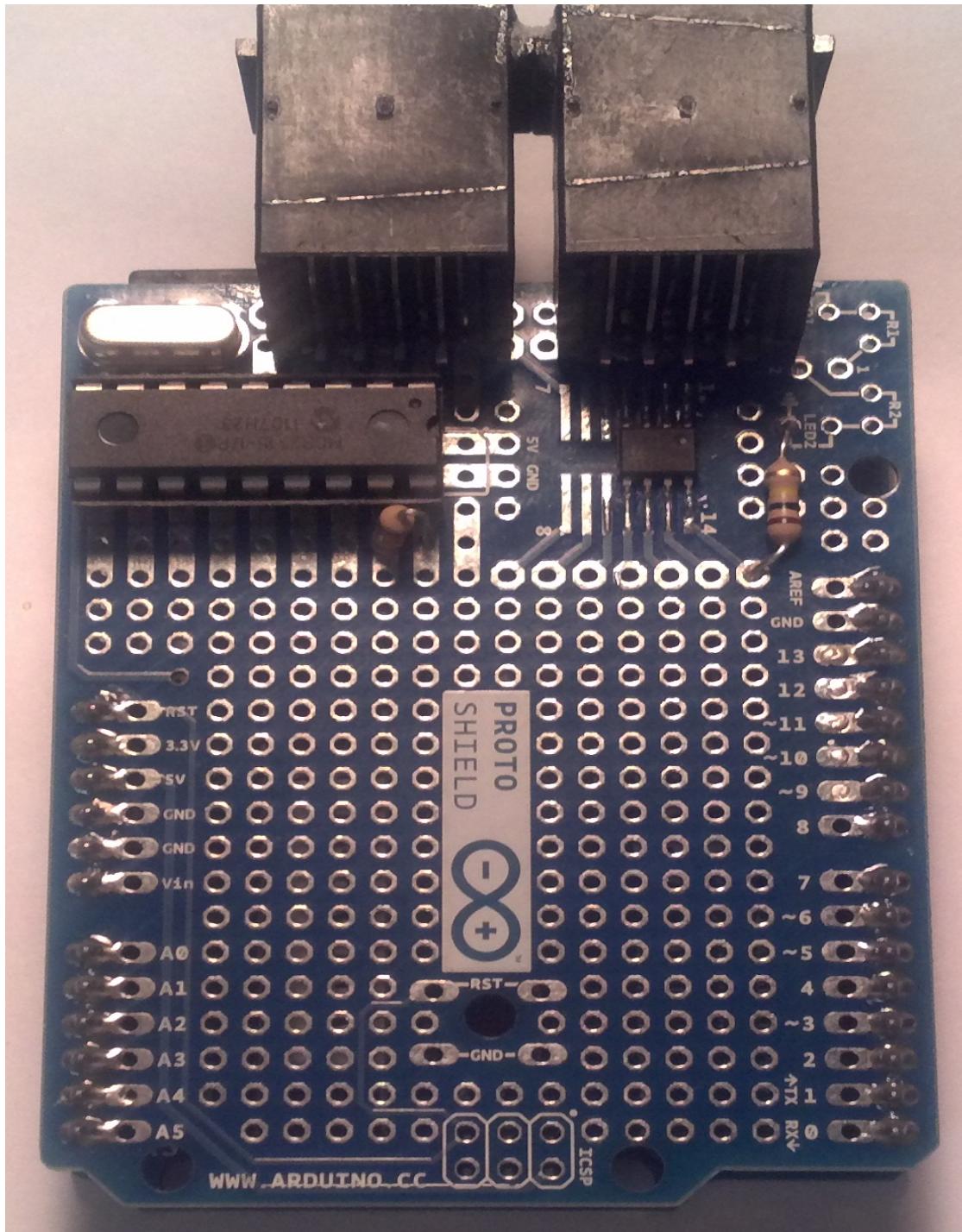


Pins used on the RJ-45 are 1,3,5,7. Two RJ-45 jacks are connected in parallel.

RJ-45		MCP2551 SOIC
1	=	Ground Pin 2
3	=	CAN High Pin 7
5	=	CAN Low Pin 6
7	=	+5 volts Pin 3

The completed protoshield. Plenty of room to add buttons, LEDs, and other components.

More to come.



Thanks for reading.

