Appendix A. Requirement and Verification Table

Table 5: Verification and Testing

Req.	Verification	Expected results	Tested
EIC.1	Send a data packet from the computer to the Ethernet controller using LabVIEW and observe the resulting output with a network analyzer from the Ethernet controller on the SPI bus. I. If the expected result does not match, then write a VI in LabVIEW that acts as the receiver of the Ethernet frame and decodes the data.	The output should exactly match with the bytes of data sent from the computer in the payload portion of the Ethernet Frame using LabVIEW. I. The decoded data should match the sent data to verify that the original VI functionally works fine.	
EIC.2	Send an analog write command with a defined waveform from the computer to the microcontroller and then probe the analog output pin from the microcontroller.	The output waveform should match the defined waveform sent by the computer.	
MC.1a	Request data sampling from the router, and then remove the computer from the network, replacing it with a different computer (or changing the MAC address)	Data should not be received by the computer. The network card on the computer will receive data, but will not acknowledge it because it was sent to an incorrect destination. A network analyzer will show the returning data, and the lack of acknowledge back	
MC.1a	Use a second computer, or change the MAC address on the main computer, communicating again	The hardware router may acknowledge requests made by the "new" MAC address (e.g. ping), but will not take any actions commanded by the computer, as shown by probing the network with an analyzer while reading the router output signals on a scope	
MC.2a	1. Feed a signal from the signal generator to the analog input, sample the data, send it to the computer through the Ethernet, plot the data, and compare versus the original provided signal. 2. Feed any nonzero value to the analog input, sample the data, and program some debugging code that sets another pin to digital out and hi if any values are actually read. Verify by reading this pin	1. The graphed data should match the signal generated, and the sample times should adhere to the sampling rate provided in the instruction to the microcontroller. 2. The voltage read by the voltage probe should be hi whenever a signal is applied to the analog input pin.	1. 2. ✓

	with a voltage probe.		
MC.2b	1. Feed a binary sequence to the digital input, collect the data, send it to the computer through the Ethernet, and compare the gathered sequence to the original. 2. Code a simple program such that another pin is set to digital out and have it match the digital input pin's measured value. Then connect an LED to the digital out pin and apply a voltage bias by hand to the digital input pin. 3. Code a simple program such that a sinusoidal PWM and a flickering digital output either change frequency or hold at the current setting when the digital input is high.	1. The recorded data sequence should match the original sequence, provided the sampling rate was high enough that all of the sequence could be caught. 2. The LED should light up during contact between the voltage bias and the pin. 3. The change in the frequency or a holding of the value should be observable with an oscilloscope and/or LED.	1. 2. ✓ 3. ✓
MC.2c	1. Using the computer and over the Ethernet, send a command to the microcontroller to generate a PWM providing a constant analog output. Then read the output with the oscilloscope to see if the PWM is the same as our set waveform and then also use the voltmeter to see if the constant analog output is as expected. 2. Repeat verification 1 but excluding the command from the computer over the Ethernet, and instead with the microcontroller preloaded with the PWM command. 3. Implement a PWM sinusoidal wave using a timer and a changing duty cycle.	1. The PWM output from the microcontroller should match that which we tried detailed in the instruction sent to the microcontroller. 2. The PWM output from the microcontroller should match that which we tried to implement in the program. 3. Observe using an oscilloscope and/or LED the sinusoidal changing intensity of the voltage outputted at the analog out. Verify that it's frequency matches the desired frequency.	1. 2. ✓ 3. ✓
MC.2d	1. Using the computer and over the Ethernet, send a binary sequence to the microcontroller to be outputted through the digital out. An LED should then be attached to the digital out. 2. Repeat verification 1 but excluding the command from the	1. The LED lighting pattern should match the binary sequence detailed in the instruction sent from the computer. 2. The LED lighting pattern should match the binary sequence detailed in the program. 3. The voltmeter should indicate that	1. 2. ✓ 3. ✓ 4. ✓ 5. ✓

	computer over the Ethernet 3. Program the digital output pin to a constant high output and test it with the voltmeter. 4. Program the digital output pin to a constant low output and test it with the voltmeter. 5. Program the digital output pin to a square wave, where by using a timer, the output switches from low to high and high to low at set intervals.	the digital output pin is set to high. 4. The voltmeter should indicate that the digital output pin is set to low. 5. Observe with both LED and oscilloscope that a square wave "flicker" signal is generated.	
MC.2e	1. Using the computer and over the Ethernet, the switches will be: I. Both turned off II. One turned on III. Both turned on After which, the output voltages from both the 5V logic and the external power supply will be read using a voltmeter. 2. Repeat verification 1 but preload the microcontroller with the command, such that the computer and Ethernet aren't involved.	1. I. The voltage read from both should be zero. II. The voltage read at the switch set to off should be zero, while the other should be the original voltage. III. Both should be the original voltage. 2. Same results expected.	1. 2. ✓
MC.2f	When given a reset instruction from the computer over the Ethernet, the microcontroller returns to its starting state.	All peripheral outputs should be low and the device power switches should switched off. The microcontroller should then go back to waiting for instructions.	
MC.2g	When a status update instruction is sent from the computer over the Ethernet, the microcontroller should respond.	The returned data from the microcontroller should properly indicate the current outputs from all pins.	
MC.3a	Command the microcontroller to set a pin to "high", and probe at the output from the hardware router	Required to be at least 3.75V, though at least 3.85V should be measured due to the buffer's specifications [6]	✓
MC.3b	Command the microcontroller to set a pin to "low", and probe at the output from the hardware router	Required to be at most 1.25V, though the output should be measured at approximately 1.045V, according to the buffer's specifications, when running at 5V.	✓
PS.1	Using a scope, probe the output of the relays for varying applied loads and switch states, for constant input voltages.	When the relays are closed, the voltage and current readouts shall be 0. When the relays are open, the voltage shall be at most the specified	✓

		voltages (5V for the logic, 60V for the external power), and shall continue to function for loads up to 500mA for the 5V logic, and up to 10A for the external power (at 12V)	
PH.1	Using a scope, check all the outputs of the power hub while varying the loads applied to the power hub.	All the voltages should be at least 4.891V, at up to 1.76A.	✓

Appendix B. Payload Protocol Definitions

Table 6: Payload Protocol Definition

Category	Instruction Command	Data
Microcontroller Status	1. State of the Microcontroller ➢ Get the output (Hi/Low) of all the pins of the microcontroller. 2. Initialize the Microcontroller ➢ Set the output value of all the pins of the microcontroller to desired value. 3. Set $V_{logic}(5V)/V_{ext}$ State ➢ Instruct the microcontroller whether to power the device through the 5V logic source or the external power supply. 4. Reset ➢ Set all of the microcontroller pins output to low.	1. • When this command is sent from the computer to the microcontroller, this portion of the payload will mean nothing to the microcontroller. • When the computer receives this command from the microcontroller, this portion will contain the status of each pin. 2. Bits representing the state of each pin to be set by the microcontroller. 3. This portion mentions whether the microcontroller powers the device through $V_{logic}(5V)$ or $V_{ext.}$ 4. This portion will not contain any useful information.
Analog Read	1. Status ➢ Is the device receiving an analog signal input from the microcontroller? ➢ If the microcontroller is outputting an Analog signal to the device input, sample the output and continuously send this output to the computer in real-time.	
Analog Write	If the device is currently receiving an input from the microcontroller, STOP immediately and output an analog signal to the device input with the following properties. 1. Functionality ➢ DC signal output value ➢ Duration (How long to maintain the output)	
Digital Read	1. Status ➢ Is the device receiving a digital signal input from the microcontroller? ➢ If the microcontroller is outputting a digital signal to the device input, continuously send this output to the computer in real-time.	
Digital Write	If the device is currently receiving an input from the microcontroller, STOP immediately and output the received data from the computer as the digital input to the device. ➢ Digital data sent to the device.	

Appendix C. Eagle Schematic Layouts

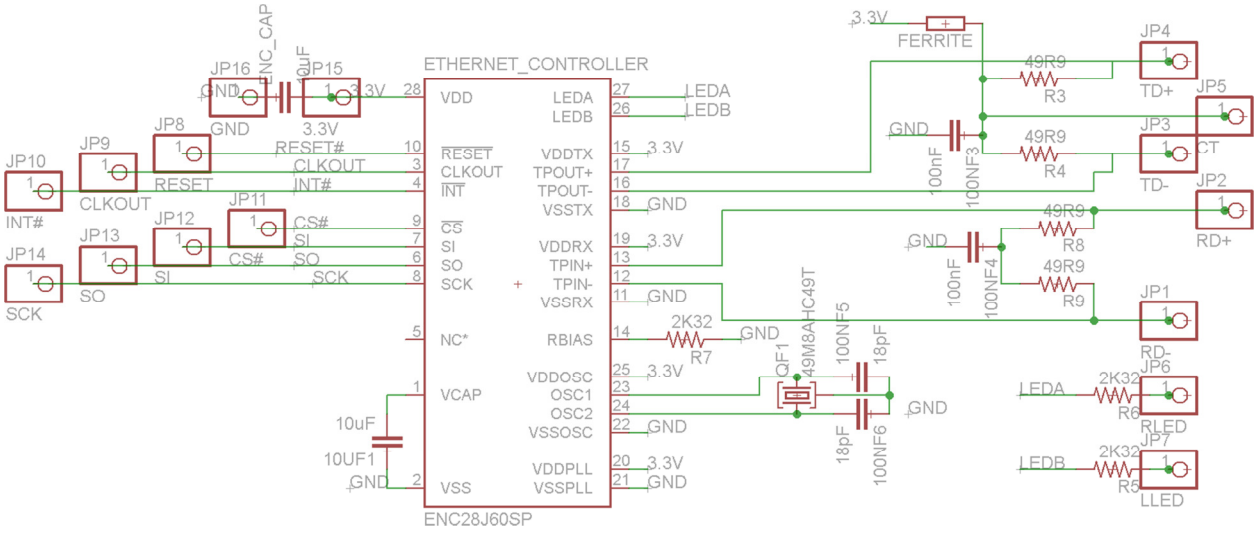

Figure 4: Ethernet Board Layout

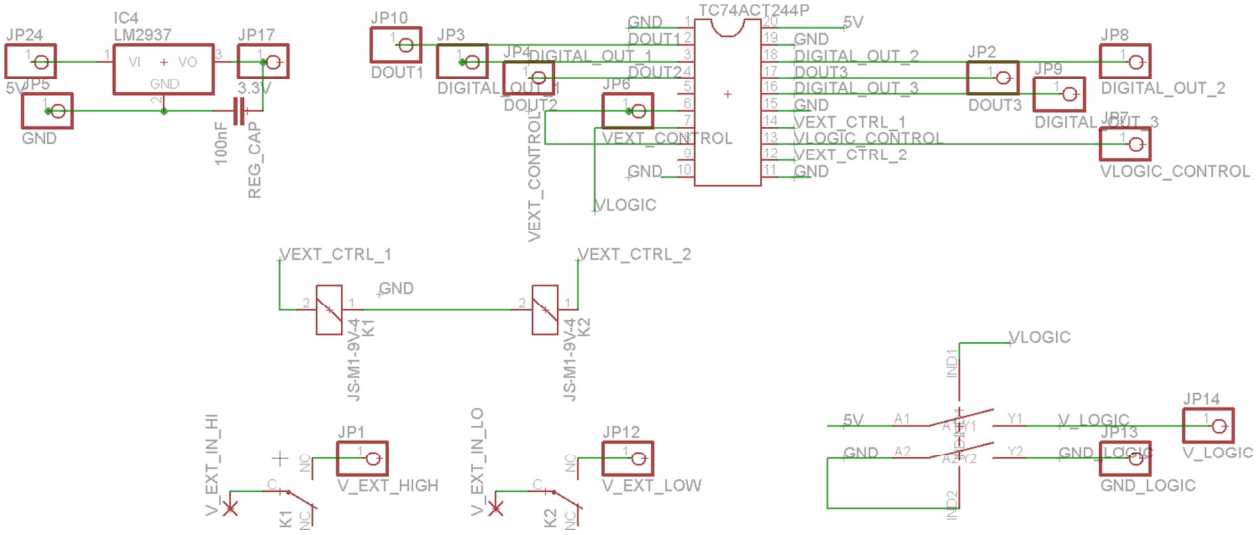

Figure 5: Power Board Layout

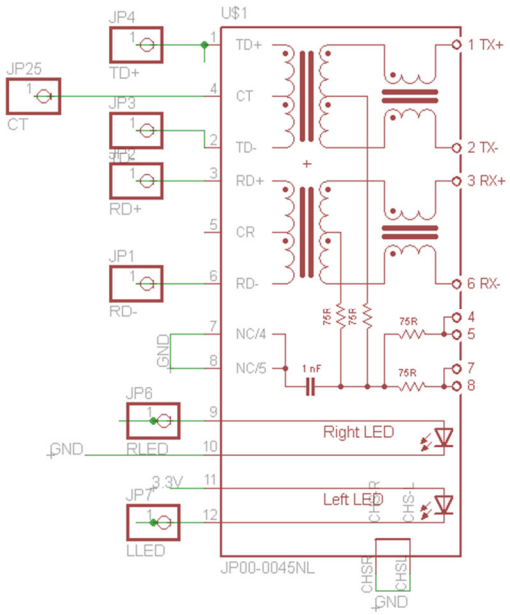

Figure 6: Ethernet Jack Layout

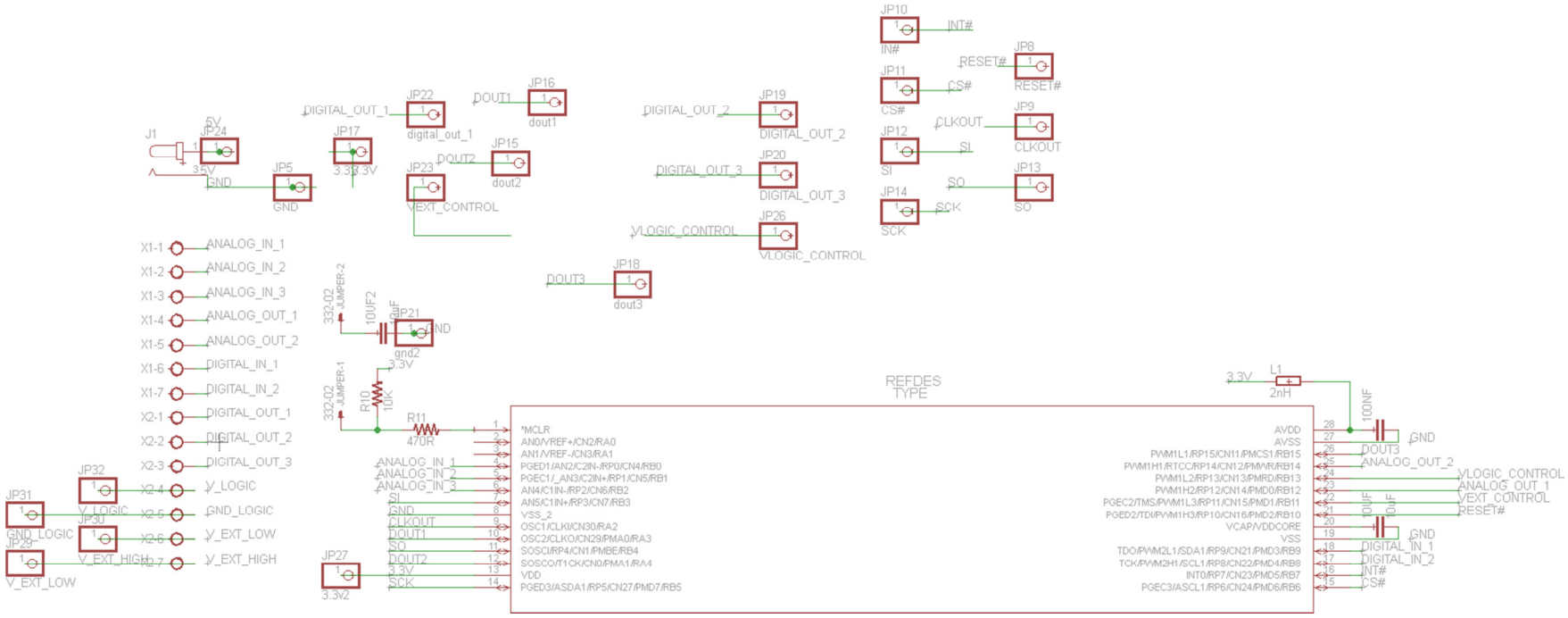

Figure 7: Main Board Layout

Appendix D. LabVIEW Program

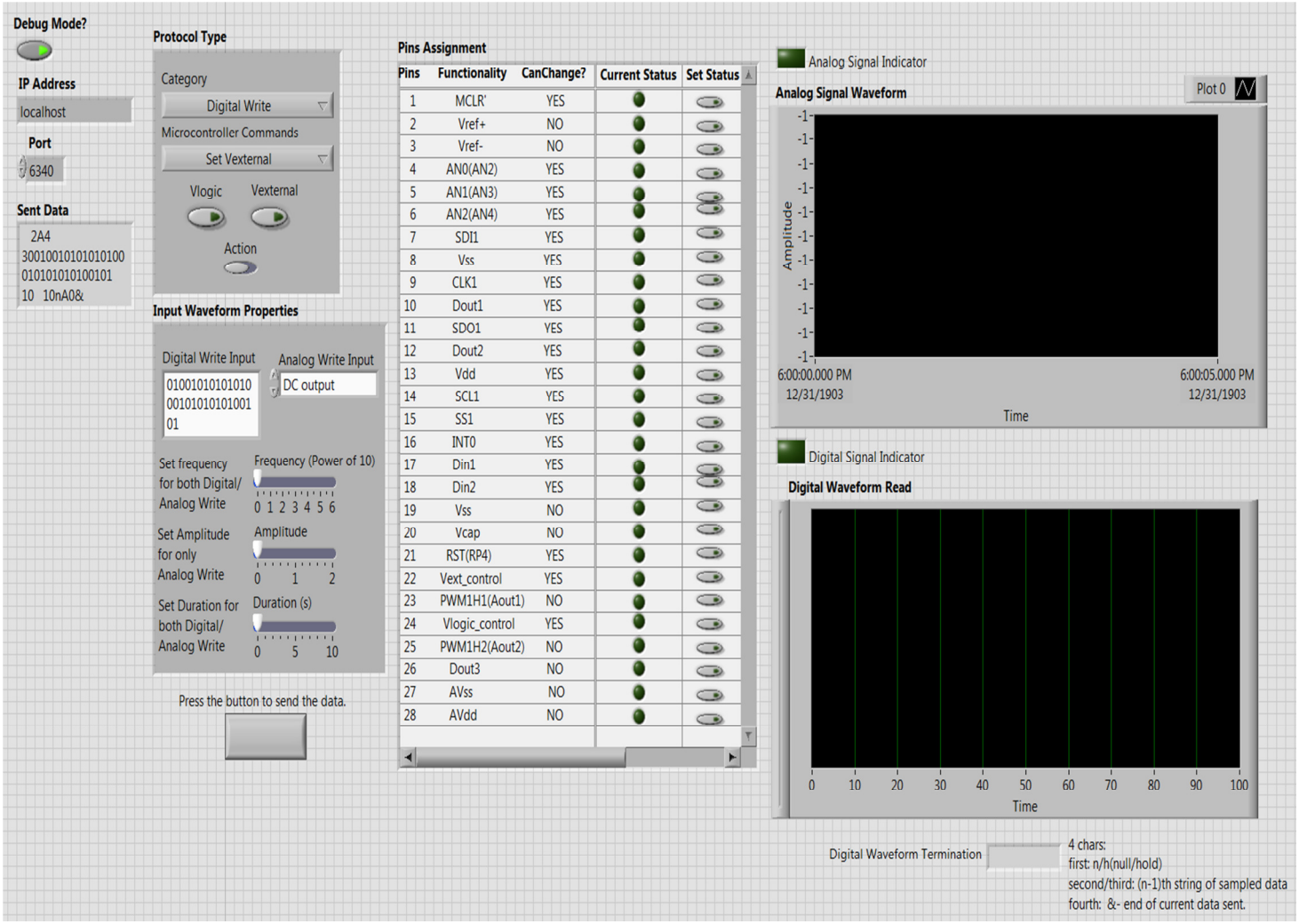

Figure 10: LabVIEW Program VI [12]

Appendix E. Final Hardware

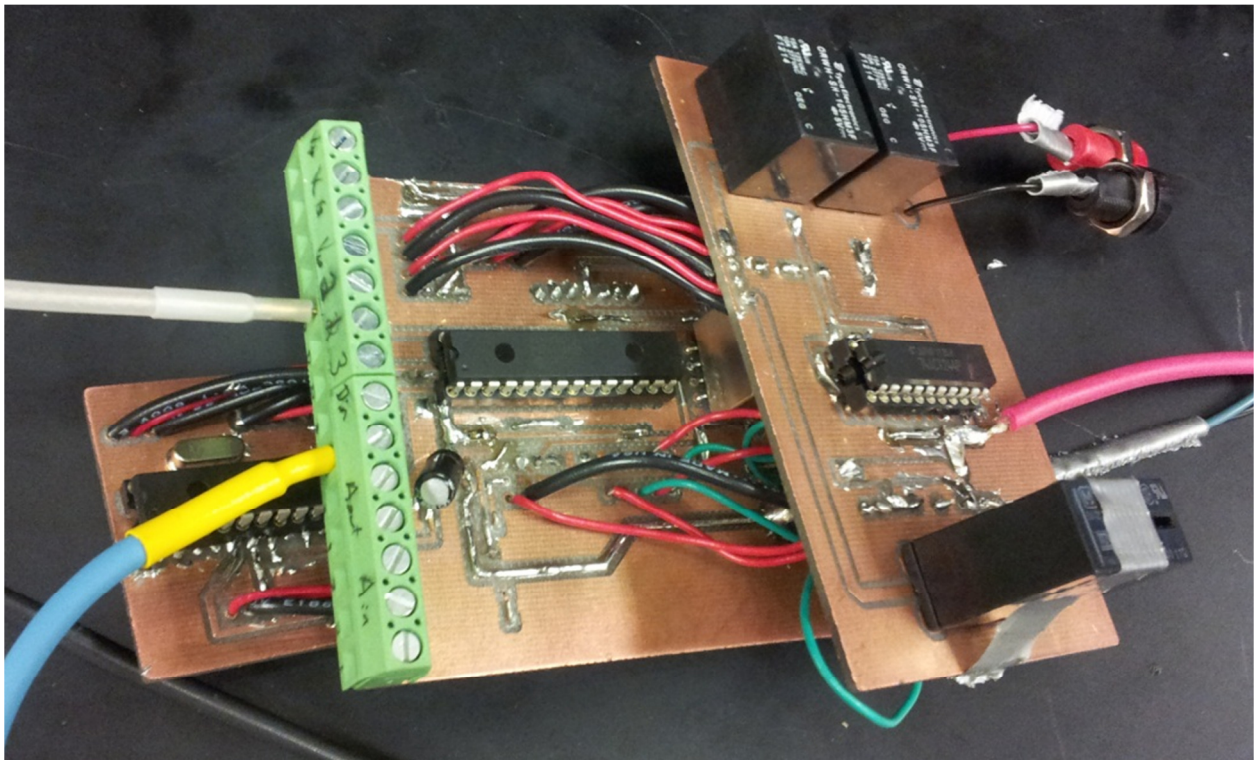

Figure 8: Finished Product

Appendix F. PWM Frequency Testing Results

Program Set PWM Frequency (KHz)	Measured Resulting PWM Frequency (KHz)	Measured Duty Cycle when Duty Cycle was Supposed to be 70.1% (%)	Measured Duty Cycle when Duty Cycle was Supposed to be 82.3% (%)
39.1	41.49	70	80
41.49	41.49	70	80.1
50	51.88	66.6	
60	62.21	70	
62.21	62.21	70	80.1
70	77.82		
77.82	77.82	68.9	81.3
80	88.89	64.4	
100	103.9	66.6	
500	623.1	50	
623	623.1	50	
700	623.1	50	
1000	623.1	50	

During implementation of the PWM module, some testing was done to find an appropriate range of frequency for sufficient resolution. Shown in the table BELOW? are the results of sweeping a small range, from which we decided to stick primarily to 41.49KHz and below for testing and demonstration of our router.