Ethernet Interface for Hardware Data Routing

Design Review

Team:
John Alaimo
Hendrik Dewald
Satyam Shah

Group: 9

TA: Justine Fortier

Prepared for:

ECE 445
Department of Electrical Engineering
University of Illinois at Urbana-Champaign

October 4th, 2012

TABLE OF CONTENTS

NOMENCLATURE

Acronym	Definition
A/D	Analog to Digital Converter
ADC	Analog to Digital Converter
API	Application Programming Interface
CRC	Cyclic Redundancy Check
D/A	Digital to Analog Converter
DAC	Digital to Analog Converter
DC	Direct Current
GPIB	General Purpose Interface Bus
GUI	Graphical User Interface
IC	Integrated Circuit
IDE	Integrated Development Environment
IFG	Interframe Gap
LAN	Local Area Network
LED	Light Emitting Diode
MAC	Media Access Control Address
NI	National Instruments
NI-VISA	National Instruments-Virtual Instrument Software Architecture
PC	Personal Computer
PCB	Programmable Circuit Board
PWM	Pulse-width Modulation
RJ-45	Registered Jack-45
SAP	Service Access Point
SOF	Start of Frame
SPI	Serial Peripheral Interface Bus
USB	Universal Serial Bus
VI	Virtual Instrument
VISA	Virtual Instrument Software Architecture

INTRODUCTION

Our project hopes to address the issue of multiple and varied device control from a single controller. Using our routing blocks, a range of useful devices like servos, motors, sensors, and LEDs can be controlled over a network. This ability proves itself useful in many robotics applications, providing a modular and compact method for distributed control of many parts and sensors. The advantages of the hardware routers are its modularity, power-management support, and device adaptability.

Existing Technology

Currently, a number of devices are required to use data acquisition units, which are either internal cards in a PC or a USB device, in order to interface them to computers. Some devices include built in USB or Ethernet functionality. However, if a device is not compatible, expensive real time computers can be required to monitor them, or at least some sort of interface card, which can either be dependent on hardware, or dependent on the computer platform by requiring particular hardware interfaces or software packages.

Benefits

- Control multiple devices of varying types from one computer
- Access through the Ethernet
- Device power control reduces power consumption
- Universal computer support – works with any Ethernet-capable computer

Features

- Analog and digital inputs
- Analog and digital outputs
- Fused power hub
- Interface with LabVIEW for verification of functionality

DESIGN

Network Block Diagram

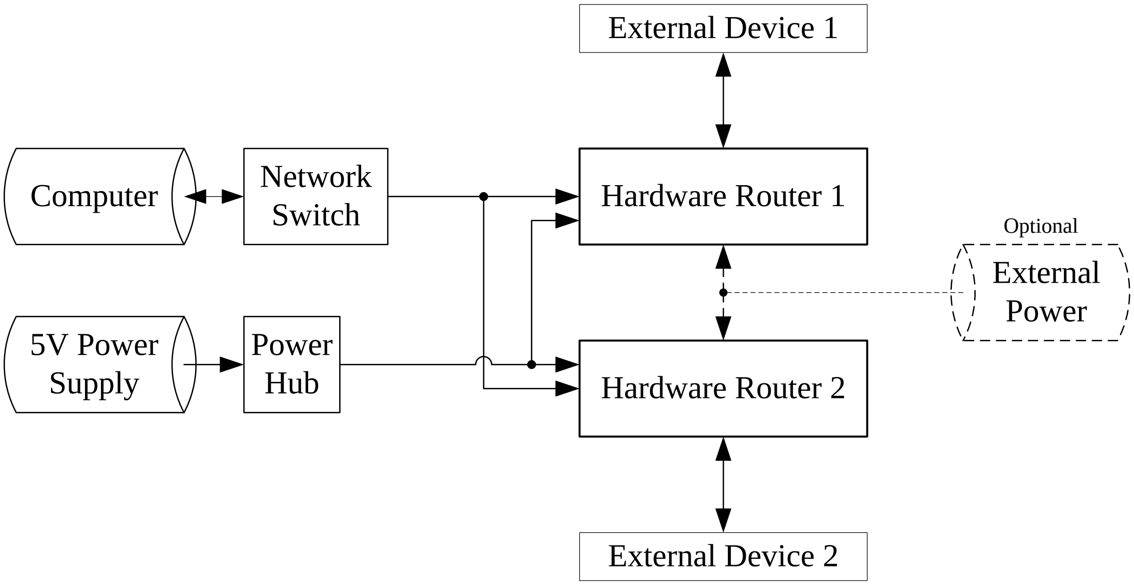

Figure 1: Network Overview

Computer: This represents any computer with Ethernet connectivity. The computer will generate commands, transmitted as a data packet called Ethernet Frame to the router through Ethernet interface using the source and destination MAC addresses. The computer will use the MAC address of the Ethernet Controller as the destination MAC address when sending data to the microcontroller. The computer will process the data sent from the routers collected from the computer network hardware. To test functionality of the final product, we will use LabVIEW to generate data commands to be sent to the router and process the data sent from the router to the computer. In general, a user can use any program that supports network communication, as well as write his/her own script to communicate with the device through Ethernet protocol.

Network Switch: A network switch connects multiple networked stations through Ethernet interface, transferring data between stations using their assigned MAC address as address. It allows multiple hardware router devices to be controlled by a single computer, thus, maintaining device modularity.

5V Power Supply: This is a power source, either a 5V battery or wall plug, which supplies power to the microcontroller through the Power Hub.

Power Hub: The power hub is the central location where the 5V logic power is distributed to the various routers, providing the routers overcurrent protection.

External Device: The external device is any device that does not already have a computer-ready interface, and is being controlled by a computer using our hardware router. Examples of such devices include motor controllers, sensors, lighting and indicators. These devices can be near the computer, or as part of a distributed system.

External Power: External Power represents optional power to be supplied to the external device, for devices requiring more power than can be sourced by the 5V logic supply, or for higher voltage requirements.

Internal Block Diagram

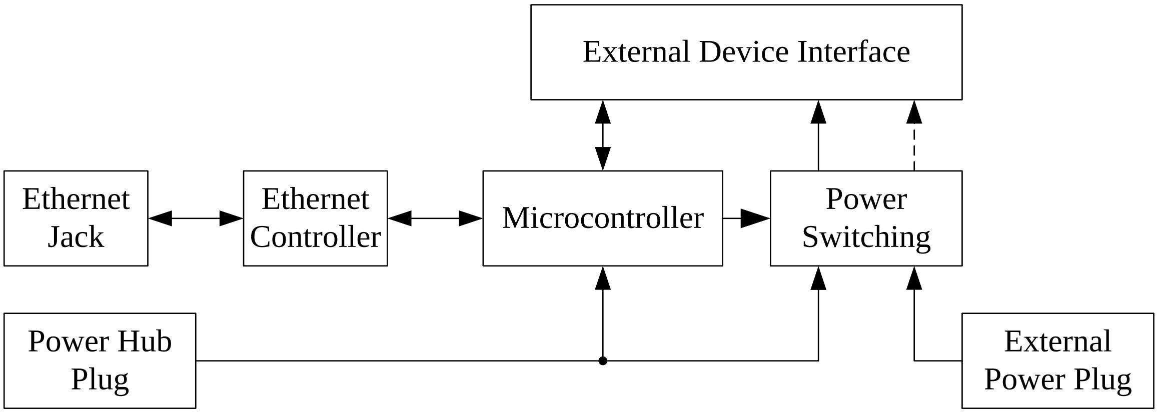

Figure 2: Hardware Router Internal Details

Ethernet Jack: This is an RJ-45 jack which will carry the Ethernet Frame to be transferred from/to the computer. It will have end stations as a Network Switch and an Ethernet Controller.

Ethernet Controller: When data is sent from the computer to the microcontroller as an Ethernet Frame, it will first pass through this Ethernet controller. This controller will decode the frame and remove the Header and all the other preliminaries and only transfer the payload portion of the Ethernet frame to the microcontroller. When data is sent from the external device or the microcontroller to the computer, this data will first be processed by the Ethernet controller to transform it into an Ethernet Frame before transferring to the proper destination.

Microcontroller: The microcontroller will send and receive data from the computer through an SPI interface with the Ethernet Controller. When started, it will wait for a first command from the computer, which will contain the MAC address of the computer. After which, it will only accept commands from and send information to the computer. Based on instructions received, it will perform a multitude of tasks and is responsible for the majority of the functionality of the system. It will directly handle the routing of power to the device, both the 5V logic power and the higher voltage external power. It will also handle the analog and digital input sampling, the digital output generation, and the PWM signal outputs. Using the microcontroller, all the device management and data gathering will be run according to the user commands through a universal instruction protocol. It will also be responsible for properly disabling unused functionality, especially when just turned on, as well as responding to network pinging and status requests from the computer.

Power Switching: Solid-state switches that control the external power as well as the 5V logic power to the external device. The microcontroller sets these switches, disabling the external device initially until the device is initialized.

Power Plug: The power plugs are the physical jacks on each router to connect the various external power sources (from the external power and 5V logic power).

Schematics

The schematics are included in the appendix, for the power hub and the hardware router.

Software Flow Block Diagram

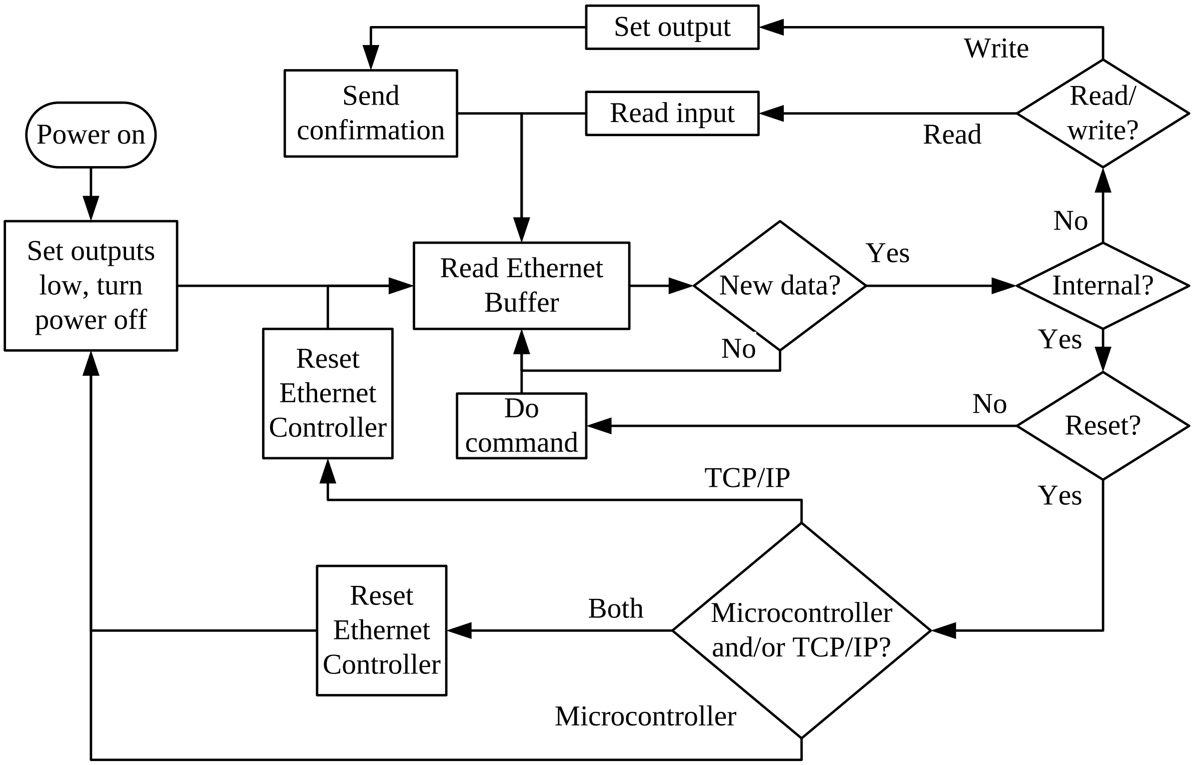

Figure 3. Software Flow

Interface between Computer and the Microcontroller

Since the external device interface with the computer will be physically wired, the device must be locally stationed. Hence, we will use Ethernet protocol to communicate with the device from the computer. Ethernet is a LAN technology with end stations assigned unique 48-bit addresses, called media access control or MAC address.

Ethernet stations communicate with each other by sending data packets called Ethernet frame. When data is to be sent from one station to another, the data is arranged in frames. There are multiple Ethernet frame types and the most common type in use today is Ethernet II frame, whose protocol is shown in Table 1.

Table 1: An Ethernet II Frame Protocol

Ethernet Frame						
	Header					
64 bits (8 bytes)	48 bits (6 bytes)	48 bits (6 bytes)	16 bits (2 bytes)	368 to 12000 bits (46 to 1500 bytes)	32 bits (4 bytes)	96 bits (12 bytes)
Preamble + SOF	Destination MAC Address	Source MAC Address	Ethertype	Payload	Cyclic Redundancy Check (CRC)	Interframe Gap

A typical Ethernet frame is minimum of 64 bytes and maximum of 1518 bytes, as shown in equations (1) and (3), not including the Preamble + SOF and the Interframe gap as a part of the frame.

$$Full\ Frame\ Min\ Size\ =\ Header + Payload + CRC \tag{1}$$

$$Full\ Frame\ Min\ Size\ = 14\ +\ 46\ +\ 4\ =\ 64\ bytes \tag{2}$$

$$Full\ Frame\ Max\ Size\ =\ Header + Payload + CRC \tag{3}$$

$$Full\ Frame\ Max\ Size\ =\ 14\ +\ 1500\ +\ 4\ =\ 1518\ bytes \tag{4}$$

The header features the destination and source MAC addresses and the Ethertype. MAC addresses are included for the intended station to know that it is to receive the package and who sent it. Ethertype provides a Service Access Point (SAP) to identify the protocol being carried in the payload of the frame and the payload is the actual data to be transmitted. The preamble alerts the receiving station that a frame is coming by allowing the receiver's data clock to synchronize with the sender's data clock. The last 2 bits of the preamble are called SOF, which are set to 11. The purpose of SOF is to notify the receiving station that the frame bits come in next. The CRC allows detection of corrupted data in the entire frame. Any frame with an invalid CRC is discarded by the receiver. Interframe gap, typically 96 bit times, is the idle period between frames required to be sent by the transmitter before transmitting the next frame to allow receiving station to prepare for the reception of the next frame. [1]

Ethernet frame is generated by the computer's operating system, but the data has to be provided by the application. Thus, the user has to structure only the payload portion of the entire frame. The MAC address of the destination station, Ethernet Controller for this project, will be supplied the user. Standard Ethernet protocol transmits bytes in big-endian format (left-most byte is sent first) and bits of each byte in little-endian format (rightmost bit of the byte is sent first). The data protocol scheme that we will be using in our project is defined in the Table 2, which will be ordered in the correct endian format before transmission after its conversion to binary. Each row represents all the information to be transmitted in the data section of each frame. Each row is divided into columns and each column occupies a fixed number of data bytes meeting the minimum of 46 bytes payload standards, which represent the variety of information to be sent to the microcontroller.

Ethernet Frame Payload Protocol

Table 2: Ethernet Frame 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.	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. Initialize the Microcontroller ➢ Set the output value of all the pins of the microcontroller to desired value.	2. Bits representing the state of each pin to be set by the microcontroller.
	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.	3. This portion mentions whether the microcontroller powers the device through V_{logic}(5V) or $V_{ext.}$
	4. Reset ➢ Set all of the microcontroller pins output to low.	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.	

When Ethernet frame is transmitted from one station to another, the receiving station (typically a computer), has the frame decoded with help of a network interface card. Since the receiving station will not necessarily be a computer in our project, but a discrete device component, the received frame will contain all the sections from

Ethernet Frame Payload Protocol

Table 2. We need an intermediate device that decodes the frame and sends only the data to the microcontroller. Therefore, an Ethernet controller has been selected, the ENC28J60 from Microchip, which has an SPI bus to connect to the microcontroller. An Ethernet board interfaces directly with the cable, and breaks out the necessary signals to go to the Ethernet controller. The pins assignment is as shown in the Figure 4. The SPI interface pins of the Ethernet controller, which carry the information, will be some of the inputs to the microcontroller and vice-versa for data transfer to be bi-directional.

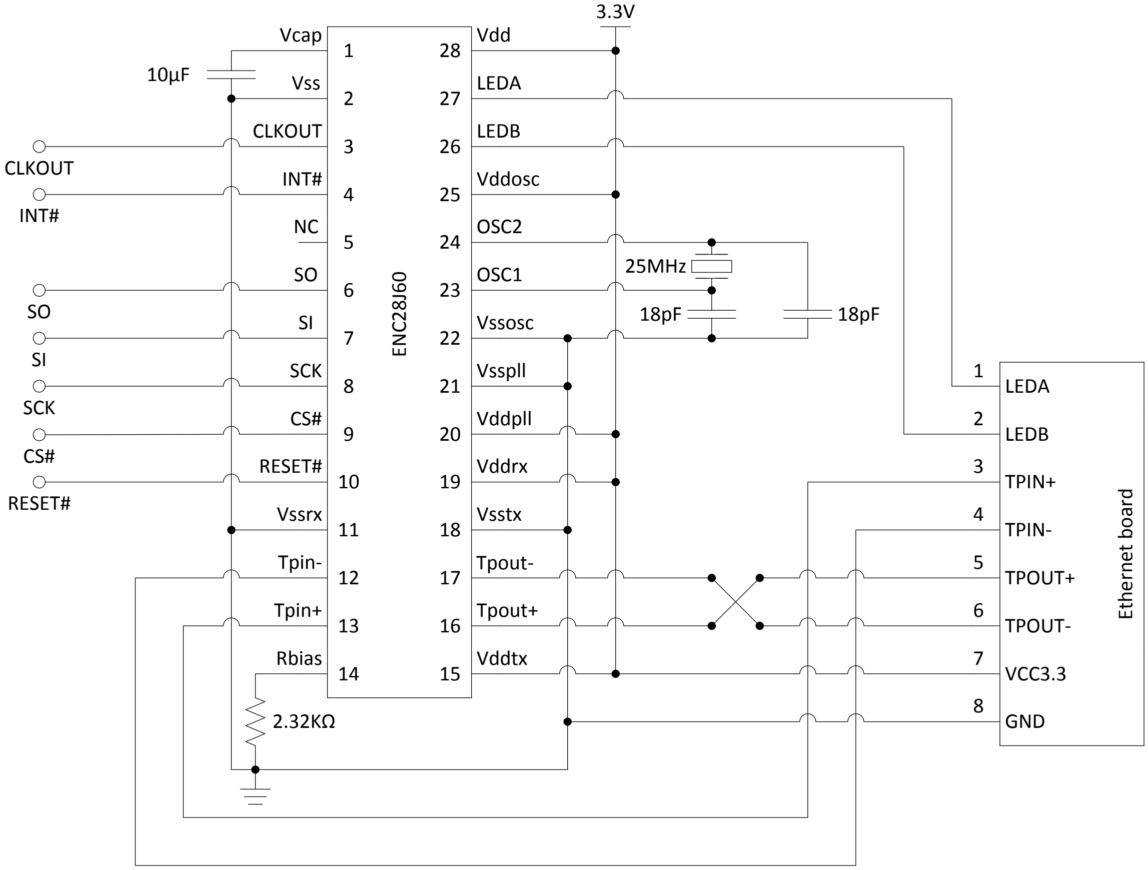

Figure 4. The Ethernet Controller pins assignment.

In our final design implementation, we will have multiple external devices interacting with the computer simultaneously. Therefore, the computer cannot support a direct Ethernet connection to each device. To maintain device modularity, we will use an n-port Ethernet switch, a 4-port switch in our case. A port on an Ethernet switch appears as an end station to a station except that it does not consume a MAC address. As an example, consider two stations; station A connected to port 1 and station B to port 2 of a 4-port switch. When station A sends a message to station B, the switch port 1 reads the entire frame into its internal input buffer. If the switch does not know the port of the destination MAC address station, it floods the entire frame to output buffer of all ports. When station B eventually replies, the switch learns that station B is on port 2. [2]

To test our final product functionality, we will use LabVIEW, a GUI program from NI. LabVIEW uses NI-VISA, a high-level API to communicate with instrumental buses over a variety of protocols such as Ethernet, USB, GPIB, etc. [3] VISA provides the programming interface between the hardware and development environments. We will use LabVIEW to generate the payload portion of the Ethernet frame, which will be transmitted over the Ethernet port once the entire frame is formed by the operating system.

Calculations

According to the dsPIC33FJ datasheet [4], an inductor can be placed, from AV$_{dd}$ to AV$_{ss}$, to eliminate noise and needs to be designed with a capacitor from the master clear pin (MCLEAR#) to ground. Generally, the frequency is half of the ADC sampling rate, such that

$$f = {}^{1}/_{2}\,(ADC\ rate) \tag{5}$$

The datasheet also specifies that the impedance of the inductor should be less than one Ohm

$$2\pi f L < 1\Omega \tag{6}$$

$$L < \frac{1\Omega}{2\pi f} \tag{7}$$

The selected microcontroller has a maximum sampling rate of 1.1 million samples per second,

$$L < \frac{1\Omega}{2\pi(1.1x10^{6}\text{ Hz})} \tag{8}$$

$$L < 289\text{ nH} \tag{9}$$

resulting in an inductance maximum value of 289 nH. To find the capacitance value, the datasheet provides

$$L = \left(\frac{1}{2\pi f\sqrt{C}}\right)^{2} \tag{10}$$

Rearranging and applying the above inequality, (9), yields

$$C > \left(\frac{1}{2\pi f\sqrt{L}}\right)^{2} \tag{11}$$

$$C > \left(\frac{1}{2\pi f\sqrt{289nH}}\right)^{2} \tag{12}$$

$$C > \ 72\,nF \tag{13}$$

That imposes a lower limit on the capacitance, and includes both types of capacitors used on the microcontroller, 10 µF and 100 nF. The lower capacitance is relatively close, almost within 25%, so the 10 µF capacitor value is selected, in order to keep parts standardized. Substituting back into (6) yields

$$L = \left(\frac{1}{2\pi f\sqrt{10\mu F}}\right)^2 \tag{14}$$

$$L = 2.093\,nH \tag{15}$$

Given the result of (15), an inductor value of 2 nH was selected.

The Ethernet controller sinks, according to the datasheet [5], I_{DD} = 180 mA. The LED pins, clock out, data out, and interrupt pins can also sink or source up to 24 mA, resulting in a total power use of

$$P = I * V \tag{16}$$

$$P = (180 + 24)mA * 3.3V \tag{17}$$

$$P = 673\,mW \tag{18}$$

The microcontroller itself uses 300 mA max, for a maximum power consumption of

$$P = 300mA * 3.3V \tag{19}$$

$$P = 990\,mW \tag{20}$$

Thus, these two controllers consume approximately 1.66 W. Allowing for a safety factor of 25% for temperature effects, the total power being provided by the linear regulator should be approximately 2.08 W. The total power into the regulator is

$$P_{in} = \frac{V_{in}}{V_{out}} P_{out} \tag{21}$$

$$P_{in} = \frac{5V}{3.3V} 2.08W \tag{22}$$

$$P_{in} = 3.15W \tag{23}$$

which results in 630 mA on the regulator. The power hub should be able to supply at least that much to each router, but a budget of double, 1260 mA, should be allocated, to include all other internal logic and support circuitry. Additionally, a limit of 500 mA is imposed on the external logic, which is consistent with the USB specifications, creating a router maximum current of 1.76 A. Again, this is at full load, and happens briefly as the transistors change states. For two routers, the power hub should be able to support approximately 3.5 A. Considering the doubled current assumption, 3.5 A is an absolute maximum, and would not be continuous. Each device would use a maximum of 1.76 A, but the best solution for overcurrent protection is fuse, and the nearest common fuse value is 2 A, which will be used on each line in the power hub.

REQUIREMENTS AND VERIFICATION

Requirements

Ethernet Integrated Circuit (EIC)

EIC.1: Receive data packets from the computer successfully.

EIC.2: Communicates with the microcontroller using an SPI interface.

Microcontroller (MC)

MC.1: Sends, receives, and interprets information from the Ethernet IC, through the use of a set instruction protocol and the SPI interface.

a) Will only send data to the computer, as the microcontroller will save the MAC address of the first command it receives which will be sent from the computer after start up
b) Will only accept commands from the computer, again, identified by the MAC address set at start up

MC.2: Can perform multiple tasks in response to computer instruction and according to supplied parameters.

a) Samples analog inputs
b) Samples digital inputs
c) Produces a PWM output
d) Produces a digital output
e) Switch on or off both the 5V logic and the higher voltage external power supply to the device
f) Can be reset to start conditions
g) Can send a status update on all pin's current settings and functionality

MC.3: Logic output voltages shall be 5V signals

a) V_{OH} > 75% Vcc
b) V_{OL} < 25% Vcc

Power Switching (PS)

PS.1: The relays and fuses of the 5V logic and the higher voltage external power supply ensure the safety of the device use.

Power Hub (PH)

PH.1: Nominally supplies 5V to each of the hardware routers, with a minimum of 4.891V for proper operation of the linear regulator.

Network Switch (NS)

NS.1: The Network Switch will

Computer (CO)

CO.1: LabVIEW interface provides a means to access and utilize the microcontroller output, input, power routing, and status functions.

CO.2: LabVIEW presents acquired data in table, graph, or scope form.

Verification

Table 3: Verification and Testing

Req.	Verification	Expected results
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	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.

	reading this pin 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.	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.
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.	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.
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 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.	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 the digital output pin is set to high. 4. The voltmeter should indicate that the digital output pin is set to low.
MC.2e	1. Using the computer and over the Ethernet, the switches will be: I. Both turned off II. One turned on	1. I. The voltage read from both should be zero. II. The voltage read at the switch set

	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.	to off should be zero, while the other should be the original voltage. III. Both should be the original voltage. 2. Same results expected.
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.

Tolerance Analysis

The regulated voltage to the microcontroller needs to be 3.3V ± 0.3V in order for the microcontroller to function. For the selected voltage regulator, it provides a load regulation typically of ±7 mV, and a maximum of ±25 mV; this is an order of magnitude less than the required tolerance of ±300mV, even at full load [7]. The output variation, varying from zero to full load and 4.9V to 30V, is +66/-33mV, according to manufacturer's tests, which still maintains the microcontroller voltage inside it's acceptable operating envelope.

For the regulator to supply enough current, the dropout on it needs to be maintained high enough, which imposes a lower limit on the voltage from the power hub. The dropout is typically 1.3V, up to 1.5V maximum at full load. Thus, the power hub output needs to be at least

$$3.3V + .066V + .025V + 1.5V = 4.891V \tag{24}$$

This can be tested on the power hub by using voltage and current probes on an oscilloscope to measure the hub output, when the power hub is connected to an electronic load, which can be varied to get a maximum current rating of 1.76A. The power hub should be able to provide up to 1.76A at no less than 4.891V for it to be up to requirements, since each device is fused at 2A, but may draw up to 1.76A as shown in previous calculations.

Additionally, a similar test can be performed on the 5V power supply going into the power hub, using an electronic load to vary current up to the device's maximum rating of 3A. If the voltage drops below 4.891V, then the power supply will need to be replaced, but the primary assumption for the power supply is that both routers will not be operating at their maximum current draws continuously. Further tests will need to be run to record the router current draw during normal operating conditions, since these conditions cannot be readily simulated.

COST

Table 4: Budget

Part	Part No	Supplier	Price	Qty	SubTotal
Microcontroller	dsPIC33FJ64GP802-DIP	Digikey	$6.50	2	$13.00
Network Switch			$10.00	1	$10.00
Ethernet controller	ENC28J60	Digikey	$3.86	2	$7.72
2 nH inductor (smd)	MLG1005S2N0S	Digikey	$0.06	2	$0.12
25 Mhz crystal oscillator	ABLS-25.000MHZ-B4-F-T	Digikey	$0.41	2	$0.82
Linear regulator	LD1086D2M33TR	Digikey	$1.02	2	$2.04
100nF Capacitor	C1608X7R1E104K	Digikey	$0.10	4	$0.40
10 µF Capacitor	EMK212BJ106KG-T	Digikey	$0.25	7	$1.75
18 pF Capacitor	C0603C0G1E180G	Digikey	$0.10	4	$0.40
Power Supply (5V, 3A)	SW-53W	Amazon	$19.99	1	$19.99
Fuse (2A)	5MF 2-R	Digikey	$0.16	4	$0.64
Fuse holder	64900001039	Digikey	$0.39	2	$0.78
Switch (SPDT slide)	GF-624-6014	Digikey	$0.77	1	$0.77
PCB		Parts shop	$10.00	3	$30.00
7 Pin Terminal Block	20020316-H071B01LF	Digikey	$1.73	4	$6.92
470Ω Resistor			$0.05	2	$0.10
10KΩ Resistor			$0.05	2	$0.10
2.32KΩ Resistor			$0.05	3	$0.15
Banana Jack (red)	108-0902-001	Digikey	$0.70	2	$1.40
Banana Jack (black)	108-0903-001	Digikey	$0.70	2	$1.40
28 pin DIP socket	4828-3004-CP	Digikey	$0.72	4	$2.88
Power Relay (SPST)	ORWH-SH-105HM3F,000	Digikey	$1.63	4	$6.52
Signal Relay (DPST)	ALA2F05	Digikey	$1.96	2	$3.92
Power Jack	PJ-037A	Digikey	$0.91	5	$4.55
Hex buffer	SN74HC365N	Digikey	$0.62	2	$1.24
Ethernet connector board		MikroElectronika	$9.90	2	$19.80
Shipping and handling			*20%*		*$27.48*
Total					**$164.89**

Table 5: Labor Cost

Worker	**Hourly Rate**	**Hours/Week**	**Weeks**	**Salary**
John Alaimo	$ 60.00	12	12	$ 8,640
Hendrik Dewald	$ 40.00	18	12	$ 8,640
Satyam Shah	$ 30.00	24	12	$ 8,640
Subtotal				*$ 25,920*
Total	**2.5x overhead**			**$ 64,800**

Estimated project grand total: $64,964.89

SCHEDULE

Table 6: Schedule

Week	Description of Task	Group Member
9/24/2012	**Design Review Sign-up, Basic Software Flow**	**Group**
	Preliminary Hardware Layout, Preliminary Power Hub Design	John Alamo
	Set-up and Learn PIC MPLAB IDE	Hendrik Dewald
	Learn LabVIEW Usage	Satyam Shah
10/1/2012	**Design Review**	**Group**
	Finish Hardware Selection	John Alamo
	Order Parts from Suppliers	Hendrik Dewald
	Develop Power-on Protocols	Satyam Shah
10/8/2012		**Group**
	PCB Schematic Design	John Alamo
	Create Skeleton code for PIC, Run Basic Simulation	Hendrik Dewald
	Develop Data Protocols	Satyam Shah
10/15/2012		**Group**
	Breadboard Verification of Layout	John Alamo
	Test Uploading to PIC, Run Simulation	Hendrik Dewald
	Run Protocols in Simulation	Satyam Shah
10/22/2012	**Individual Progress Report**	**Group**
	PCB Schematic and Layout, Submit for Milling	John Alamo
	Verify Data Transmission with Network Analyzer	Hendrik Dewald
	Develop LabVIEW Interface	Satyam Shah
10/29/2012	General Troubleshooting	**Group**
	Further Test Breadboard Layout, Refine connections	John Alamo
	Test and Refine Protocols	Hendrik Dewald
	Test and Refine Protocols	Satyam Shah
11/5/2012	**Mock Demos, Mock Presentation Sign-up**	**Group**
	Mock Presentation Sign-up	John Alamo
	Sign-up,Test Protocols	Hendrik Dewald
	Refine LabVIEW Interface	Satyam Shah
11/12/2012	**Mock-up Presentations, First Revision PCB Fabrication**	**Group**
	Revise PCB Design for Additional Router(s)	John Alamo
	Test Routers using Different Devices	Hendrik Dewald
	Test Routers using Different Devices	Satyam Shah
11/19/2012	**Final Revision PCB Fabrication, Thanksgiving Break**	**Group**
11/26/2012	**Demos and Presentation Sign-up, Fine-tune and Repair Testing Damages**	**Group**
	Schedule Practice Room and Possible Audience	John Alaimo
	Final Code Adjustments	Hendrik Dewald
	Final Code Adjustments	Satyam Shah
12/3/2012	**Demos, Practice Presentation**	**Group**
12/10/2012	**Presentations, Final Paper, Lab Checkout, Lab Notebook**	**Group**
	Compile All Process Documents	John Alamo
	Inventory Check on Loaned Design Lab Supplies, Find Any Missing	Hendrik Dewald
		Satyam Shah

RELEVANT IEEE ETHICAL ISSUES

We will adhere to all of IEEE Code of Ethics, especially those that pertain most to our project.

> 1. To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment

We will take into account all the safety measures possible to make sure the end product is safe to use; especially, any safety concerns related to the control of the external power, where our specified device ratings will have at least a 25% safety factor included with respect to the manufacturer’s listed ratings.

> 3. To be honest and realistic in stating claims or estimates based on available data

We will not fabricate any of our results or attempt to hide any shortcomings of our final product. All results and subsequent conclusions will be supported by collected data and simulations.

> 7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others

We are trying to make a cost-effective and the best possible product design using all the available resources. We are open to any constructive criticisms made that would help us in improving the product design, performance, and its applicability. Since the product is made using parts and programs already available in the market, we will do our best to give full credit to their manufacturers in any means possible.

REFERENCES

[1] *Ethernet II Frame*. (2012, October). *Ethernet Frame* [Online]. Available: http://www.infocellar.com/networks/ethernet/frame.htm

[2] *Ethernet Switch.* (2011, April). *Understanding Ethernet Switches and Routers* (Volume 3, Issue 1) [Online]. Available: http://www.ccontrols.com/pdf/Essentials0411.pdf

[3] *NI-VISA.* (n.d.). *What is VISA?* [Online]. Available: http://www.ni.com/visa/

[4] Microchip Technology, “16-bit Digital Signal Controllers (up to 128 KB Flash and 16K SRAM) with Advanced Analog,” dsPIC33FJ64GPX02/X04 datasheet, Sept. 2007 [Revised May 2009].

[5] Microchip Technology, “Stand-Alone Ethernet Controller with SPI Interface,” ENC28J60 datasheet, Jan. 2006 [Revised July 2012].

[6] Toshiba Corporation, “TC74ACT244P/F/FW/FS,” TC74ACT244 datasheet, April 1996.

[7] STMicroelectronics, “1.5 A adjustable and fixed low drop positive voltage regulator,” LD1086D2M33TR datasheet, May 2006 [Revised Mar. 2012].

APPENDIX

Power Hub

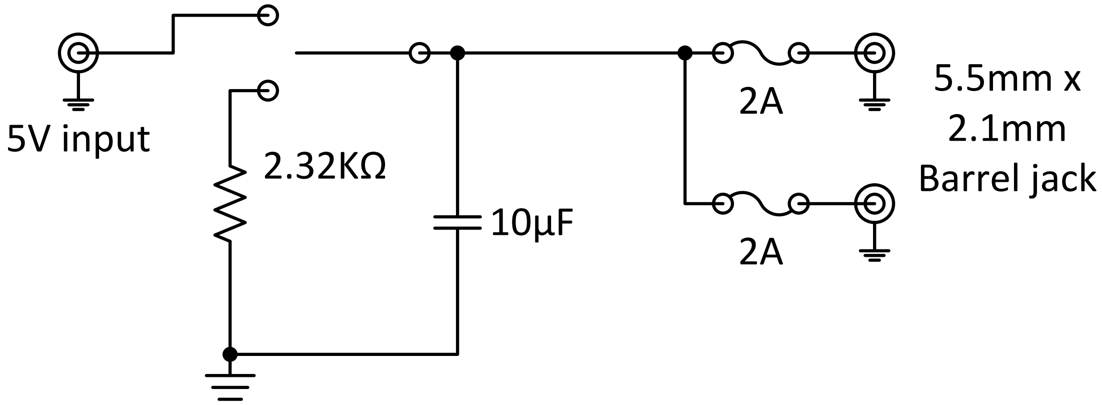

Figure 4: Power Hub Schematic

Microcontroller and Buffer

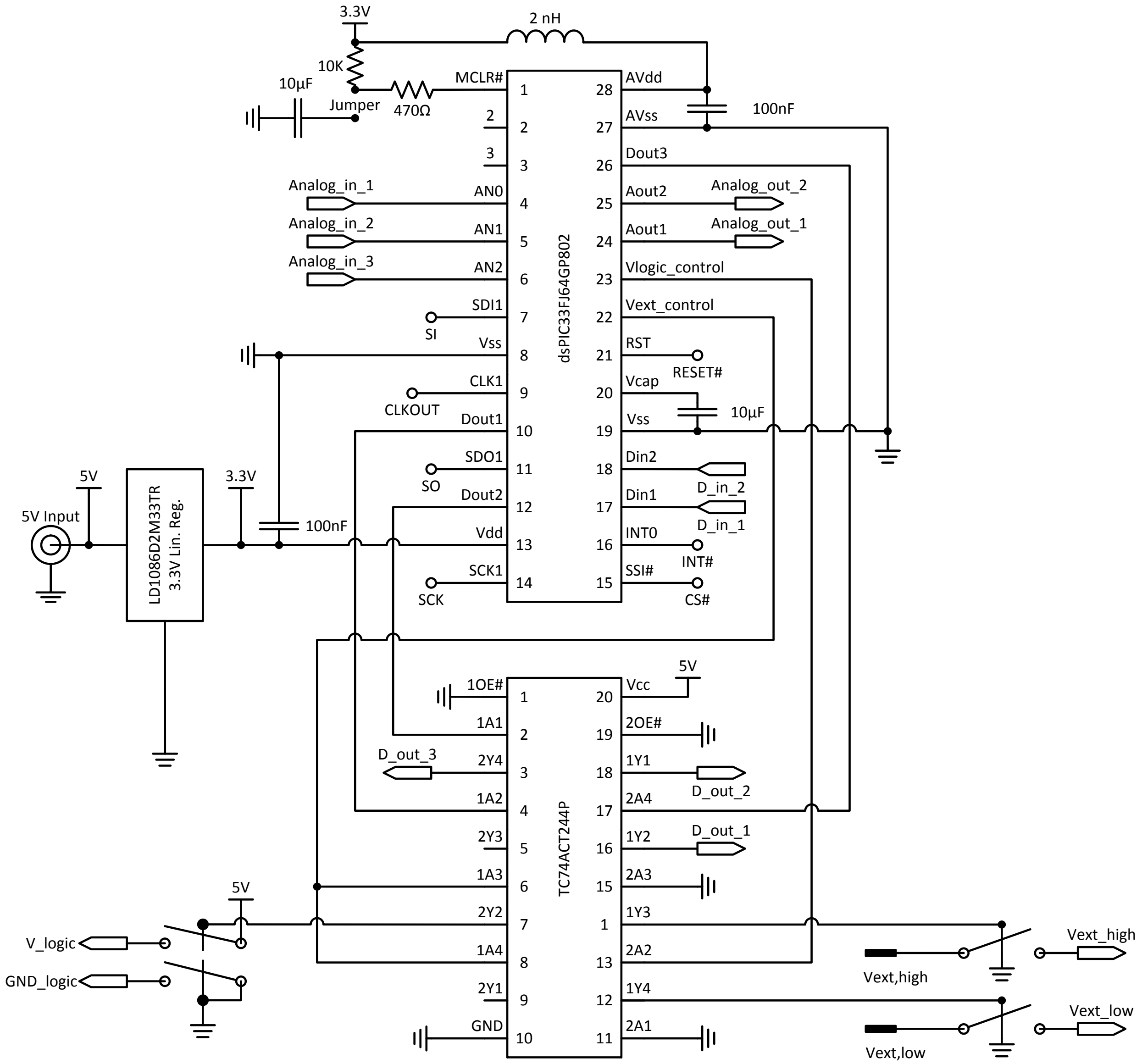

Figure 5: Microcontroller, Buffer, and Relays

Ethernet Controller

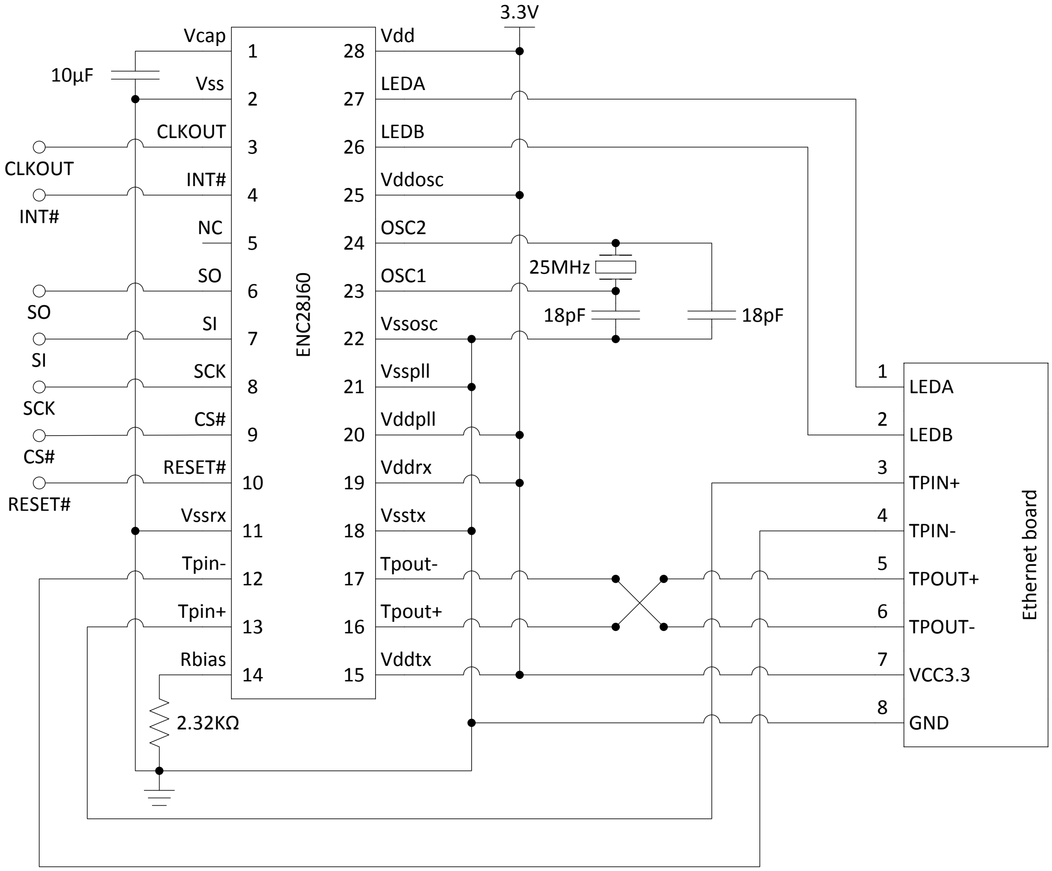

Figure 6: Ethernet Controller, Data Lines, and Ethernet Interface Board