qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdTheGamehjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopaMattisTheBestsdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| Final Project Report - HAL  Matthew Lamoureux-Durr  5/7/2012  Final Project II |

Executive Summary:

This project is called HAL which stands for Home Automation Logic. What the project is is a home automation system. What that means is that you can remotely and by user defined schedule that tells the system to turn on or off 120V outlets that can be hooked up to lights or appliances. This is all done over either a computer that the system is connected to or over any web enabled device.

How the system works it that is has an ATMEGA32u4 to control it, and a ENC29J60 that enables it to be accessed over a network connection. It’s main outputs are the relay boards that control the lights/appliances. The user can see what’s going on with the system in person with the graphical LCD that is attached to the system.

The circuit in ISIS is broken out into the 5 main section of the system: the LCD, the Ethernet section, the MCU section, the Inputs, the Outputs and the Power Supply. The LCD section shows how the LCD is hooked up to the micro. The Ethernet part shows how it is setup and how it connects to the micro. The MCU section that shows how the ATMEGA32u4 is hooked up. The Inputs and Outputs pages show example circuits for each. And the Power Supply section shows the 5 and 3 volt regulator.

Each of the circuit design pages are shown in the report.

I list the import maximum ratings of components like the AVR, the ENC28J60, the relays, and voltage regulators.

Next I give a short user guide on how to interact with the device. Through the web server, through the GUI and though the LCD.

A bill of materials of the explained circuit is shown. This part shows what devices are used and how many of them as well as what their values are if applicable.

A timeline is presented to show when I expect to accomplish each task in the process of making this project for both this semester and the next.

In conclusion of the report I say that I had a good experience with my project, but it did not meet my expectation. My biggest problem was with my Ethernet device and if I had more time, I’d fix my big bugs, then add more featuers.

TABLE OF CONTENTS:

* Executive Summary – Page 1
* Table Of Contents – Page 2
* Introduction – Page 3
* Functional Description – Page 4
* Circuit Diagrams – Page 10
* Maximum Ratings – Page 16
* User Guide – Page 17
* Bill of materials – Page 19
* Timeline – Page 21
* Conclusion – Page 23
* Resources – Page 24

Introduction:

This project at it’s core is a home automation system. It’s purpose is to allow easy and automated control over AC appliances and lights. My motivation for this project was to allow myself to easily control and automate appliances in my household. The idea is to be able to change the state of lights/appliances from either a set schedule or live form either USB connected to a computer or from a network connection. It’s practical uses besides practicality could be to remotely turn off appliances if they were left on.

This project has an ATmega32u4 microcontroller as the MCU, an ENC29J60 Ethernet controller and a ST7565 graphical LCD. The ATMEGA will be controlled either over Ethernet or USB to set it’s schedules or its current state of it’s outputs. Basically I want this system to automate my household and making it accessible over both USB and Ethernet gives me/the user more options and when used over Ethernet it means that you don’t have to have a host machine connected to it to give it commands, instead you could use a cell phone or other web enabled devices, or you could use the LCD and buttons on the device to control it. I plan on leaving room in the design for future add-ons, such as ADC ports and extra IO pins. The relays will be located on external daughter boards to reduce the size of the main system and that way the device is not limited on how many appliances you can plug in.

There are similar commercial and homemade systems available to mine. But the difference between my design and the ones that are online/available is that theirs are specialized (ie. sprinkler system) or are not designed to be modular or have as much direct feed back as mine. What my system shares in common with many of the home-made systems is that they use the ENC29J60 chip to control the Ethernet component.

Part of this project consists of several parts, one is the Circuit Design. This was where in the ISIS program I had to create my circuit, including the ATMEGA chip, the relays, ENC29J60 and the LCD.

The next part was to get approval of this circuit by our professor Dr. Hussam Al-Hertani, and correct and mistakes or adjust the circuit based on input from him.

The next steps will be to get a running prototype going and get the PCB designed in ARES.

Functional Description:

My project is broken into five main areas:

 First let's start with the graphical LCD part. In my project I am using a ST7565 graphical LCD. It has a resolution of 128x64, it has a RGB backlight which I plan on using as a fading screen-saver effect for when the system is idle and for giving the user visual feedback in addition to what is being displayed on the screen. The LCD interfaces with the micro using SPI interface, which transmits data using a serial based protocol that allows more than one device to share the same data lines in addition to each having their own chip select so that it knows when to listen to the data on the shared lines.

As seen in Figure 1, the LCD is using 3V logic, so a 5V to 3V buffer chip is used to convert the 5V logic of the micro down to the 3V logic of the LCD. The LCD it'self is controlled using 5 pins, SID for serial input data, SCLK for the SPI clock, A0 for telling the LCD if the data is a command of data for screen, RST for resetting the LCD and CS for chip select.

Figure

The RGB LED is controlled using the pins B-, G-, A+ and R-. The anode is connected directly to 5V, and each of the cathodes are connected to a PWM pin on the micro using a NPN transistor to connect them to ground.

It's operating temperature range is between -10 deg C and 60 deg C. It takes about 1mA for the LCD plus 20mA for each LED.

The second major part of my project is the Ethernet component. The magic is controlled using the cheap ENC29J60. Using this in my project will allow my system to receive commands and send status updates over a LAN or WAN connection to any web enabled device at a very cost effective range. It allows TCP/IP connections to be made to the system for showing information to a client or for taking commands from a client.

 This device also uses an SPI interface to connect with the micro. But in addition since connections from the device could come at any time there is also a dedicated interrupt pin used to tell the micro that there is data coming in.

Figure

The chip is rated from 10Mb/s connections and runs at 3.3V. Its power consumption is about 180mA, so this with the LCD make about 200mA (rounding) of 3.3V the whole system needs to supply. The chip runs at 25MHz, so it has its own crystal, and one feature this chip has is that it can take it’s 25MHz clock and divide it down to give other clock that you can use in the rest of your system. I will not be using this clock. It has an operating temperature Range of between -40°C to +85°C.

For the micro itself, I’m using an ATMEGA32u4 breakout board from adafruit.com . It has a pre-populated reset button, 16 MHz crystal status LEDs and resistors and capacitors for the USB circuitry.

The chip runs at 5V and by default, draws power from the USB but I will be cutting that connection and be powering the board only by the systems 5V regulator.

As seen below in fig 3, that is an approximate circuit diagram of the MCU board.



Figure

The power supply for the system consists of a MC7805CT 5V regulator and a L4931 3.3 V regulator. They are rated for 1A and 250mA respectively. Their operating temperature range is -40 °C to 125 °C.

My system will need minimum about 100mA on the 5V rail, but that will increase based on how many relays are connected and if other sensors are added. And I will be using about 200mA on the 3.3V rail, most of that goes to the ENC29J60 module.

See below for the configuration of the power regulators in my circuit.

The only inputs I have designed into my circuit so far is are a set of selection buttons. They will be used to highlight options in the menu that will be displayed on the LCD. But I will be leaving all unused pins exposed on the circuit board to that extra sensors could be added in the future.

Figure

See below in fig 5 to see that the configuration will be.



Figure

For outputs, I have the relays connections. The relays will be mounted on separate external relays control boards. This allows the user to select only what they need to control their appliances and allows for simple upgrades if they need a bigger rated relay.

 The output goes to the base of a transistor located on the external relay board. When the output is activated the transistor will allow current to flow through the transistor and activate the relay. The average current consumption of a relay is about 40mA.

Figure



Figure



Figure



Figure



Figure

Figure





Figure

Maximum Ratings:

LCD:

* Logic level voltage max = 3.3V

ATMEGA32U4:

* 16MHz @ 5V
* 32KB of flash
* 2.5KB of RAM
* 1KB of EEPROM
* 5.5V Maximum

ENC28J60:

* Max SPI speed of 20MHz

30A Relay:

* Max coil power 1W
* Max on to off / off to on time 15ms

15A Relay:

* Max operating power 1250VA

MC7805CT (5V Regulator):

* Input voltage 30V
* Power dissipation 15W

AP1117(3V Regulator):

* Max input voltage 18V

User Guide:

With my project, it has three main ways to interface with the HAL unit. Website, Windows GUI and onboard LCD.

Website:

The website offers two usable pages currently. The first is the landing page that shows the systems current time and date, as well as a section of that page that is for monitoring and changing the state of the systems relays. The page renders dynamically, on render (HAL side), the system will copy the systems time and date values into variables to serve to the client browser. The system also checks the state of each relay and will either render an “Off” or “On”. And when one of the toggle buttons is pressed, the client browser will request the normal URL with additional variables “?RU&RLX=TGL” where the X would be replace with the relay number to receive a command and TGL means that it should toggle. When the system gets a URL with these commands it will parse them out and apply the requested command to the relay specified.

The other usable page of the webserver is the time page, where the webpage will take the system time at render and place it into a textbox that can be sent to the server. After setting the time, the now synced time can be viewed on the homepage.

GUI:

The GUI has two main functions, one is that it syncs the system time to the HAL unit over COM port on initialization, and the other is to view and control the state of the relays.

LCD:

The LCD lets you access two system menus using the buttons located underneath the LCD. Pressing the RED button when the system is in its idle state brings up main menu at the top of the LCD. It shows you two options you can choose using the RIGHT and LEFT buttons on the keypad, either the LED menu or the TIME menu. To close this main menu, press the UP button. Your currently selected menu item will appear inverted so that it is surrounded by a white box.

In the LED menu, you have the option to change the contrast of the LCD, and the brightness of the three colour channels of the backlight LED (RED, GREEN, BLUE). At the bottom of this screen you will see a SAVE and EXIT button, the SAVE button exits this menu while saving your current selection, and the EXIT button exits the menu while change back to the setting you had when you started this menu.

In the time menu, you can see the current time and date that the system thinks it is. This is a read-only menu.

Bill of materials:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Total Parts In Design** | : | 73 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 21 Resistors | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | Quantity: |  | References |  | Value |  | Order Code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 5 |  | R1-R5 |  | 330 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | R6, R11 |  | 1k |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 5 |  | R7, R12, R17, R18, R21 |  | 10k |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | R8 |  | 2.3k |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | R9, R10 |  | 230 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 3 |  | R13, R15, R16 |  | 50 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | R14 |  | 50. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | R19, R20 |  | 22 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 Capacitors | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | Quantity: |  | References |  | Value |  | Order Code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | C1, C2 |  | 18pF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | C3 |  | 0.1uF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | C4, C17 |  | 10uF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 3 |  | C5-C7 |  | 100nF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | C8, C9 |  | 20pF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | C11, C12 |  | 1uF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | C13 |  | 100uF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | C14, C16 |  | 22uF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | C15 |  | 10nF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 Integrated Circuits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | Quantity: |  | References |  | Value |  | Order Code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | U1 |  | 7805 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | U2 |  | ENC28J60 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 Transistors | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | Quantity: |  | References |  | Value |  | Order Code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 7 |  | Q1-Q7 |  | NPN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 Diodes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | Quantity: |  | References |  | Value |  | Order Code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 4 |  | D1-D4 |  | DIODE-SC |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | D5 |  | LED-BLUE |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22 Miscellaneous | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | Quantity: |  | References |  | Value |  | Order Code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 5 |  | DOWN, LEFT, RIGHT, SELECT, UP |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | J1 |  | AU-Y1007-R |  | Digikey AE9925-ND |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 4 |  | J2, J4-J6 |  | TBLOCK-I4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | J3 |  | CONN-H2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | L1 |  | 1mH |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | LCD1 |  | ST7565 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | MCU1 |  | ATMEGA32U4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | RJ1 |  | RJ45 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 4 |  | RL1-RL4 |  | OMI-SH-105D |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 |  | VR1 |  | L4931 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  | X1, X2 |  | CRYSTAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Timeline:

|  |  |  |
| --- | --- | --- |
| **Semester 1** |  |  |
| **MileStone** | **Start Date** | **End Date** |
| **Complete Project** | 8/22/2011 | 12/7/2011 |
| **Designing** |  |  |
| Design Phase I - Brainstorm | 8/22/2011 | 9/5/2011 |
| Design Phase II - Project Proposal | 9/5/2011 | 9/12/2011 |
| Design Phase III - ISIS | 9/12/2011 | 9/26/2011 |
| Design Phase IV - ARES | 10/20/2011 | 11/10/2011 |
| Design Phase V - Project Report | 9/26/2011 | 10/19/2011 |
| **Implementation** |  |  |
| Implementation Phase I - Order Parts | 9/26/2011 | 10/20/2011 |
| Implementation Phase II - Bread-boarding | 9/26/2011 | 11/15/2011 |
| Implementation Phase III - PCB Assembly | 11/15/2011 | 12/1/2011 |
| **Testing** |  |  |
| Testing Phase I - Power Supply | 10/1/2011 | 10/7/2011 |
| Testing Phase II - MCU | 9/15/2011 | 9/25/2011 |
| Testing Phase III - LCD | 10/1/2011 | 10/25/2011 |
| Testing Phase III - Ethernet | 10/15/2011 | 10/31/2011 |
| Testing Phase III - Relays | 11/1/2011 | 11/7/2011 |
| Testing Phase III - PCB | 11/15/2011 | 12/1/2011 |
|  |  |  |
| **Semester 2** |  |  |
| **MileStone** | **Start Date** | **End Date** |
| **Complete Project** | 1/16/2012 | 5/8/2012 |
| **Designing** |  |  |
| Design Phase IA - Brinstrom Firmware | 1/16/2012 | 2/13/2012 |
| Design Phase 2A - Prototype Firmware | 2/13/2012 | 4/1/2012 |
| Design Phase 3A - Polish Firmware | 4/1/2012 | 5/5/2012 |
| Design Phase IB - Brinstrom GUI | 3/13/2012 | 3/20/2012 |
| Design Phase 2B - Prototype GUI | 3/20/2012 | 4/1/2012 |
| Design Phase 3B - Polish GUI | 4/1/2012 | 5/5/2012 |
| **Implementation** |  |  |
| Implementation Phase I - Get Hardware to communicate with PC | 3/10/2012 | 5/5/2012 |
| **Testing** |  |  |
| Testing Phase I - Troubleshoot | 3/10/2012 | 5/5/2012 |
| **Semester 2** |  |  |
| **MileStone** | **Start Date** | **End Date** |
| **Complete Project** | 1/16/2012 | 5/8/2012 |

Conclusion:

My experience with my final project was pretty good. I can say I learned a lot about the software parts of my project, like how to hunt down broken functions that are given to me, how to modify 3rd party libraries to better suite my application like the GLCD library from Adafruit. I had a few annoying drawback towards the end when I was cramming in more features, like my micro randomly crashes when the user is interacting with it, and the incompatibility of my GUI on any other system then the one it was developed on.

I think this project fell a little short of what I was hoping to have, but only because development of the software took longer than I hoped, or at least I was not able to put in enough hours into it. It’s not that it was difficult, it just took long to figure out how each new piece of the puzzle needed to fit into my project.

The biggest challenge of my project was getting the Ethernet device working. When I was building it I still was not sure that I had the software to support it, and after I built it I was encouraged to use an “Arduino” to test it, since I did have code to test it with that, but it as not working, this either mean that my hardware was not designed correctly or that the software was crap. Turns out that the software was crap because as soon as I ditched the “Arduino” platform and turned a C library that I found I had it working in minutes. The only reason I wasted time with the Arduino system is that I was afraid that the real C library would be too difficult for me to understand, turns out that it was about 400% easier than trying to understand the stupid Arduino code structure, when you get down to the bare bone of the Arduino code, everything is virtualized, so it is very complex and inefficient. Whereas the C code, was very straight forward and made sense based on what we were tough in class.

If I had more time at the end of this class, I would first of all ~~kill~~ eliminate the bug I have that will hard-lock the entire system randomly when using either the LCD or the Web interface. After that I would fix the main bugs I have with my GUI program, mainly the issues I was having with the serial communication. After those were solved I would finish implementing my scheduling system, and then add options for it in all three interfaces to read and create new events. After that I want to have a manual override system on the HAL LCD to control the relays. After that, I need to implement options for monitoring remote IO on the relay boards, like manual overrides. On the hardware side, I would like to have used latching relays to save power on the state of the relays, instead of having to give power whenever I want them on. Also creating a proof-of-concept daughter board would be really cool for the system, as I left the hardware UART pins of the micro untouched I could easily add something like a Speech-To-Text or more IO.

Resources:

**Similar** **Project** :

OpenSprinkler : <http://rayshobby.net/blog/?page_id=160>

HomeSeer : <http://www.homeseer.com/>

**Parts:**

Atmega32u4 : <http://www.adafruit.com/products/296>

LCD : <http://www.adafruit.com/products/438>

Ethercard: <http://shop.moderndevice.com/products/jeelabs-ether-card>

**Code Libraries Used:**

ENC28J60 : <http://tuxgraphics.org/common/src2/article09051/>

Adafruit GLCD Library : <https://github.com/adafruit/ST7565-LCD>

Tuxgraphics Time Conversion Library : [http://tuxgraphics.org/common/src2/article11022/](http://tuxgraphics.org/common/src2/article11022/%20)