Computer Systems Design

22 April 2016

Tyler Sriver | Brad Browning

Table of Contents

Abstract

In this report we cover and discuss the computer system we built over the course of a semester based on the M68HC11 microprocessor. This system was designed and built to interact with a liquid crystal display (LCD) with a keypad to control laser. The keypad would is mapped to alphanumeric characters that are able to be displayed by the LCD and drawn with a laser. This report aims to show the overall design of the system as well as results from final testing. Herein are discussions on the process of building and programming the system, as well as where we encountered trouble and what we learned from the entire process.

Introduction

In this project we were tasked with building a system that would interface with an LCD and keypad to control a laser and program it to accomplish this task. The end goal was to control the laser to draw both alphanumeric characters and an animation by the way of four “modes.” We spent the first semester of this course understanding the M68HC11 microprocessor; how to program it, and how it interacted with the memory. The second semester we spent the majority of our time building the system and making sure it was operational so we could then implement a functional code. This report is aimed at giving someone with general electrical and computer engineering knowledge to recreate our system in its entirety. We will use diagrams and flowcharts to represent several hardware and software aspects of the design as well as include our source code.

Needs for the System

This system has several requirements for what the software has to accomplish and how it is to be built. The requirements state what the general design of the hardware should be and what duties the software should be able to accomplish to utilize the LCD, keypad, and laser. The LCD should be able to display all the alphanumeric characters mapped to the keypad, and the laser should be able to display the last four characters pushed on the keypad. The software will operate in the following modes:

Mode 0: LCD will only display the characters from the keypad, see Appendix D
Mode 1: Laser will display a test pattern, see. Appendix B
Mode 2: Laser will display up to four consecutive characters
Mode 3: Laser will display an animation, see Appendix C

Subsystems

For visual see Appendix A

- Microprocessor
  - Motorola 68HC11 microcontroller
  - Using EEPROM chip for memory
- Chip Control and Control Logic
  - GAL programmable logic chip
- Laser System
  - Peripheral Interface Adapter (PIA) hardware and A/D converters

Specifications

- Voltage inputs for the system from power supply
  - +5 VDC
    - Used for most of the system components
  - $^{+}$/- 15 VDC
    - Used for the PIA subsystem
- Processor operates at 2 MHz
  - Controlled with an 8MHz crystal
- Variable Delay
  - Controlled with a potentiometer

System Design

Hardware Design

Hardware Components

- Processor: Motorola 68HC11 class processor:
  - Model: 52-Pin PLCC MC68HC11A8 with 256 Bytes RAM
  - Use: used for the overall processing for the system
- Programmable Logic Device:
  - Model: ispGAL22v10C - Electronically Erasable Floating gate technology to allow us to re-program it to achieve desired logic
  - Use: used to control the reset of the system, and when subsystem IC's needed to be activated
- Keypad:
  - Each key sends a value in hex $00 - $13 across memory bus

    - Use: general purpose 20-key keypad
- LCD :
    - Model: PC 1604-A
    - Use: The liquid crystal display with 4 rows
- Output Power Supply:
    - Model: Agilent E3631A Triple output
    - Use: Supply power to the entire system
- Memory:
    - Model: AT28C64B
    - Use: EEPROM used for storing program
- Driver Chips:
    - Model: 74HC245
    - Use: Increase the current of the output pins used for the Addresses and Data
- PIA:
    - Model: Motorola MC6821P
    - Use: Interface between processor and laser driving the X and Y coordinates
- D/A Converter:
    - Model: Atmel AVR400
    - Use: Convert 8 bit data to +/-10 V X/Y points
- Operational Amplifier (Op Amp):
    - Model: LM741
    - Use: Amplify voltage to +/-10V
- Potentiometer:
    - Model: General 10k ohm potentiometer
    - Use: variable delay for laser
- Flip-Flop:
    - Model: Motorola 74HC74
    - Use: detect rising edge of keypad input
- Latch:
    - Model: 74HC00 nand chip
    - Use: reset and laser on/off logic

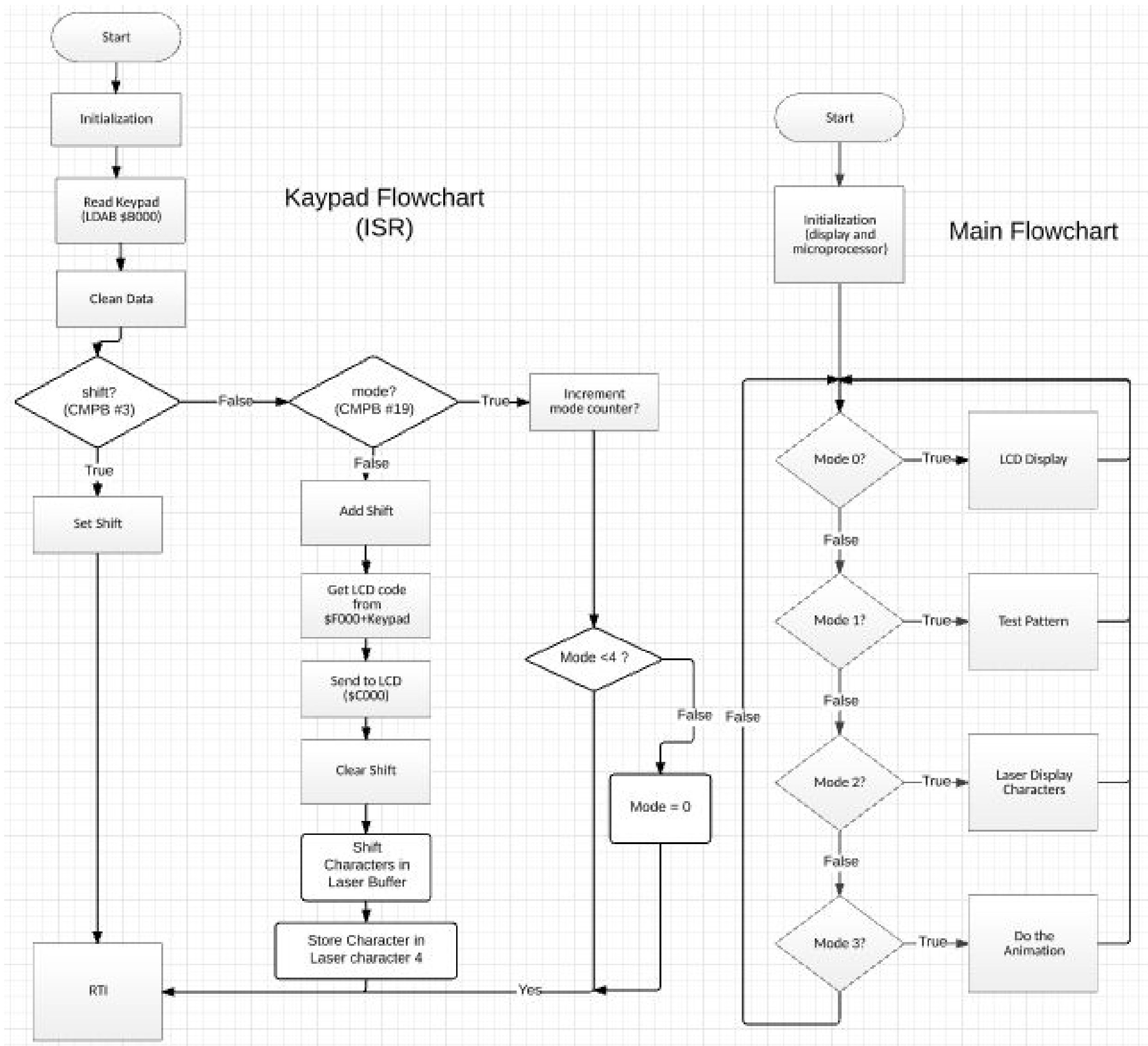
Debug Hardware

- Oscilloscope
    - Model: Tektronix TDS 2022 Two Channel Digital Storage Oscilloscope
    - Use: Testing the output of the PIA
- Multimeter:
    - Model: Agilent 34401A
    - Use: Test voltage of pins

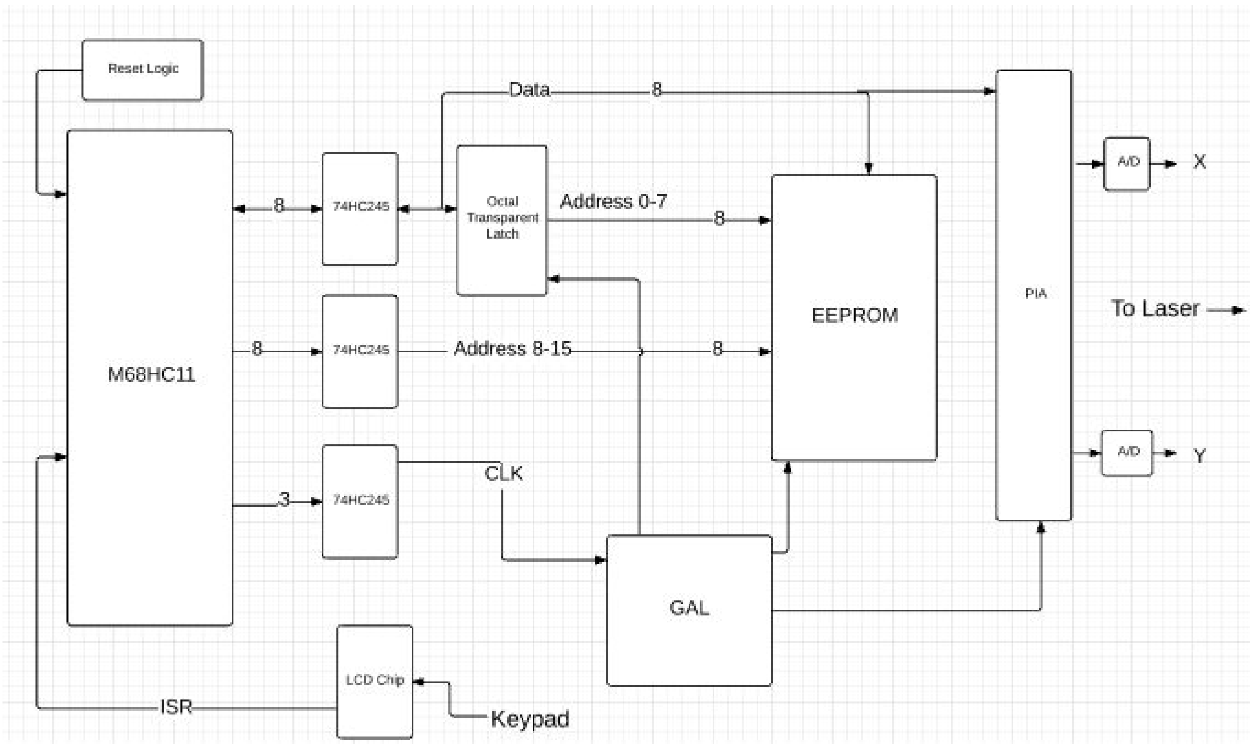
- Logic Analyzer:
    - Model: Agilent 16802A Logic Analyzer
    - Use: View contents of Address and Data bus

Software Design

High Level Flowcharts



High Level Block Diagram



Detailed Design

Detailed Memory Design

We have the the first 256 bytes (0x0 to 0xFF) of memory being used as random access memory (RAM) in the processor. We have 0x100 to 0xDFFF allocated for communication with external devices. Table 1 shows the memory-mapped locations for each device. For our project, we haved used 0x1030, 0x1031, and 0x1039 for use of the processor's internal A/D converter. We used 0xE000 through 0xFFF for read-only memory (ROM) , in which we hold the program and constants that is being used to run our system.

Table 1. Memory-Mapped Locations for External Devices

	LCD	Keypad	PIA	Laser
Memory-mapped location	0xC000	$B000	0xD000 - X	0xA001 - off 0xA000 - on

			0xD002 - Y	

Detailed Display Design

The LCD displays alphanumeric characters that have been chosen by pressing keys on the keypad. The characters are displayed from left to right in order of rows 1,3,2,4. After the fourth row has been filled, the cursor will jump back to row one and restart the cycle. A blinking cursor is displayed in the place for the first character that is not being read yet.

Detailed Keypad Design

Whenever a key is pressed it triggers the interrupt on the microprocessor and stores the key that has been pressed into address location $B000

Detailed Laser Design

The supplied laser takes three inputs: an analog x-location, an analog y-location, and a digital laser-on or laser-off signal. These three inputs are used to control the laser in order to display the desired alphanumeric characters and animation. Using the PIA and the A/D converters we can take data from the 8-bit data bus and relate it to a voltage between -10V and +10V.

Testing the System

Subsystem Testing

To test the system, we needed to first take into consideration each subsystem, checking it off as functional before moving on. This ensures that each particular group of parts work as expected. Given the cumbersome nature of this project, each of the following tests include the hardware of the previously listed one to include as little electronics as possible. This was all in efforts to simplify the system and reduce errors in the long run.

Clock

We aimed to test the basic functionality of the microprocessor in this test. This did not require any software. After the 8 MHz crystal and power and ground were connected, we could hook up the output clock pin to the oscilloscope (scope). This test was successful once one cycle (2 MHz) occurred every 0.5 μs.

Reset Button

This piece of the system allowed the microprocessor to reset to the beginning of the instruction set of our code. Even though this part was unnecessary, it proved extremely useful for debugging the system. By tieing the output of the reset to the multimeter in Vdc mode, we could verify this test. This Test was successful if the the output was high without action and low when the button was pressed.

Drivers & Demultiplexer

These electronics were tested by connecting the output of the address bus to the logic analyzer. This subsystem test was successful when the reset button was pressed, the address was set to 0xFFFE, and then the reset button was released, to show the address on the logic analyzer to be 0xFFFF.

EEPROM

The EEPROM is the main memory for our system, it is used to store the program and all associated data. To test this we loaded an extremely short program that would loop and continually cycle through loading 0 then 1. This was helpful because we were able to use the logic analyzer and see the bit change from 0 to 1 so we knew that the memory was working.

GAL

The GAL replaced the 7400-series of chips, which were performing NAND and OR tasks on short processes for certain data. The test for the GAL comprised of first testing the drivers, demultiplexer, and the memory chip. The GAL test was successful once the system was able to perform the prior tests successfully.

LCD

We then attempted to display the word, “Hello.” Once this was successful, we proved the LCD was working properly by getting all alphanumeric characters to display on the LCD.

Keypad

This test required the implementation of the D flip-flop and keypad converter. We displayed the letter, “A” on the LCD when that letter was pressed on the keypad. Once this was successful, we made sure we were able to again display the phrase, “Hello” on the LCD when these sequence of characters were pressed on the keypad.

PIA

The PIA is used to control the laser and the easiest way to test it is to load a program that displays a simple shape that demonstrates using the full range of the laser. Once we knew we had the full voltage range from the D/A converters we loaded a program that draws a box and we tied the laser control to high so the laser remains on.

D/A converters & Op-Amps

Before we are able to test the laser functionality we tied all the bits into the PIA high to see that the X and Y output were a +10V and then tie them all low to see that they were a -10V. This simple test demonstrates achieving the full voltage range we expect to receive from the Op-Amps.

Laser On/Off switch

This small subsystem designed to control when the laser is turned on uses a 74HC00 to build a latch that through the control of a signal from the GAL can turn the laser on and off. The laser control was a simple high or low output so to easily test if it was working is to trigger the latch and see if the output was either high or low.

Variable Delay

This addition to the system allowed us to control the delay between actions on the laser therefore being able to manipulate the laser so it more accurately displays our desired character or animation. In order to test this first you have to have a potentiometer connected to a pin on the microcontroller (Shown in Appendix A) and then to power and ground. After an addition to the code is made by altering the position of the potentiometer you can visually see the change in the picture displayed by the laser

Complete System Testing

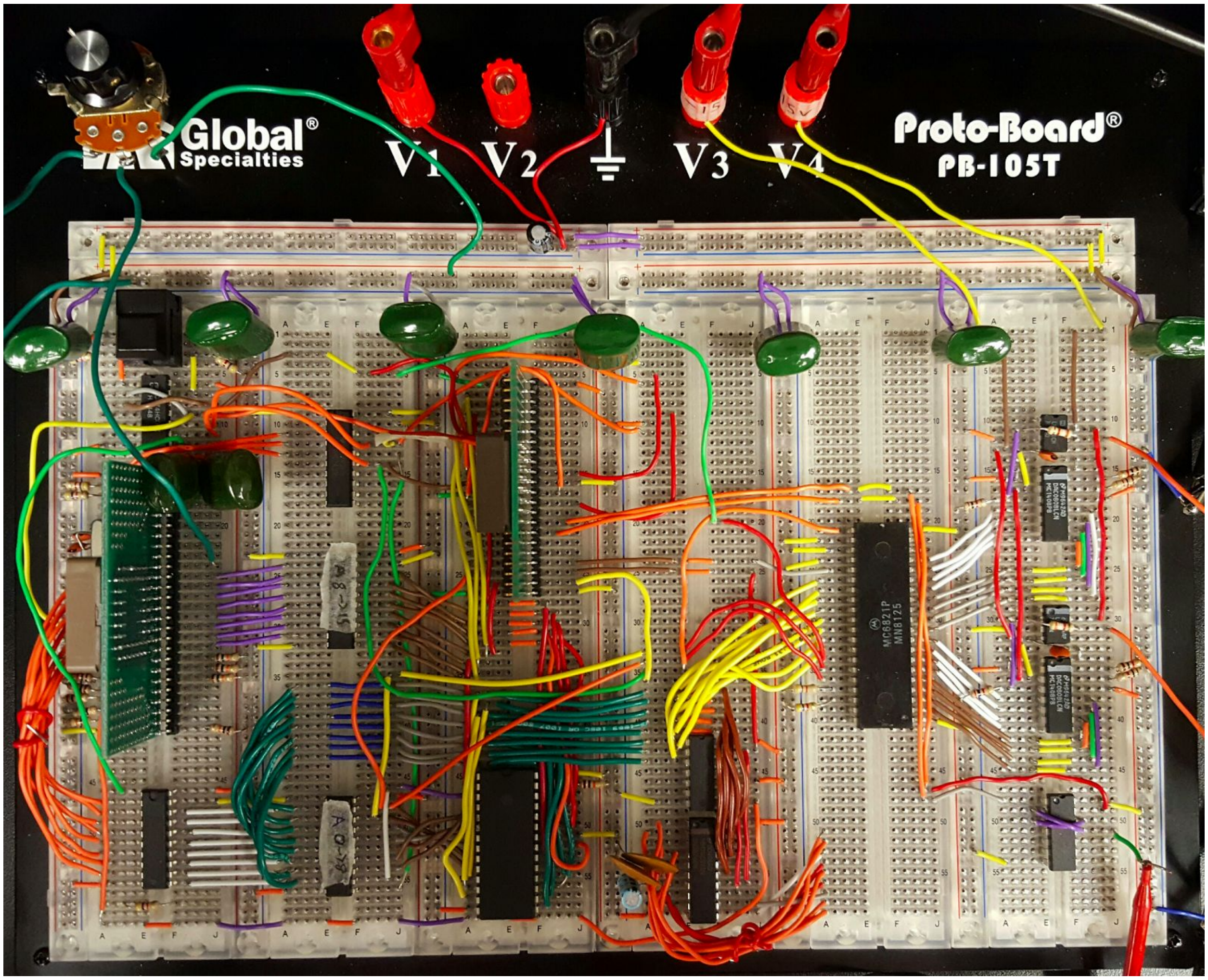
After we have all of the individual subsystems working separately we have to make them all work together to form our complete system. A complete test of the system would be to have all of the modes described in the requirements met, this was done with our main program which is including in the appendices E, F and G.The keypad is shown in Appendix D; we have the mode button on the keypad working which switches between all of the different display/control modes. The shift key also works and is used to output a different character from each of the keypad buttons.

Conclusion

At the end of the semester we we able to complete the requirements of the project, we successfully were able to demonstrate all of the modes both using the keyboard, the laser and the variable delay. We had a couple of problems displaying a few of the characters on the laser, we have about five characters that do not work on the laser. Over the course of the semester we kept neat wiring which allowed us to keep a steady pace and debug any issues quickly and effectively. With a working system and the end of the semester this project was a success.

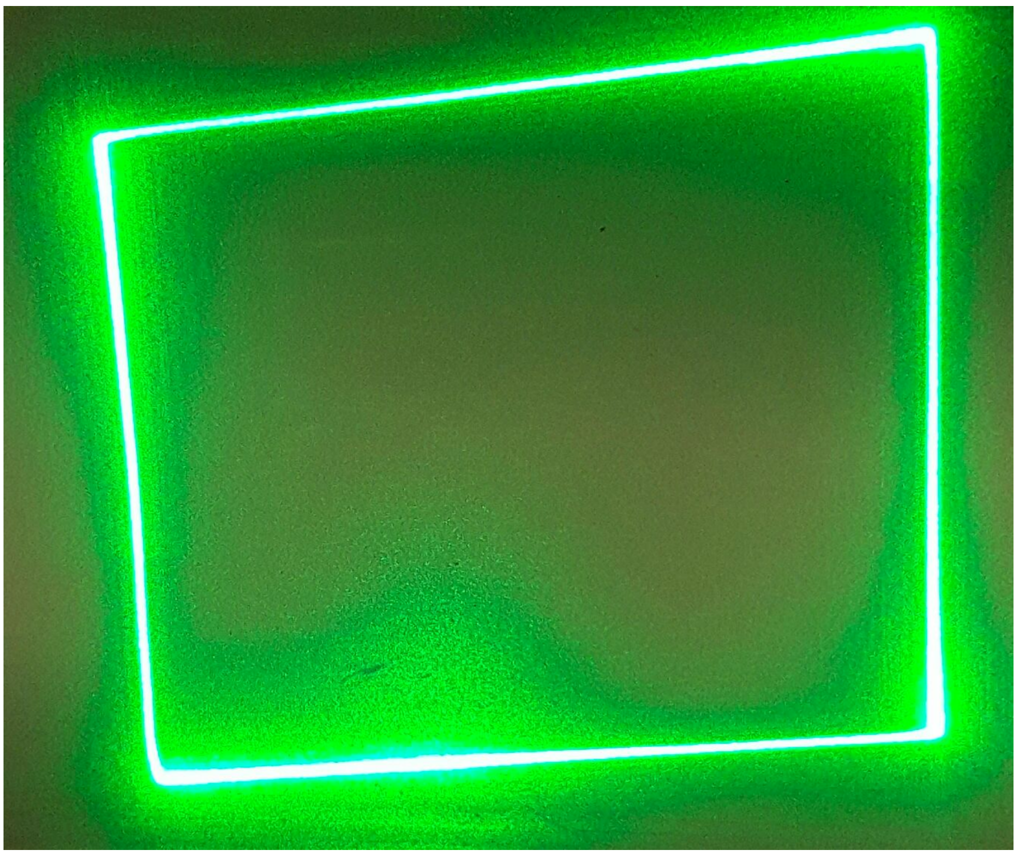
Appendices

Appendix A



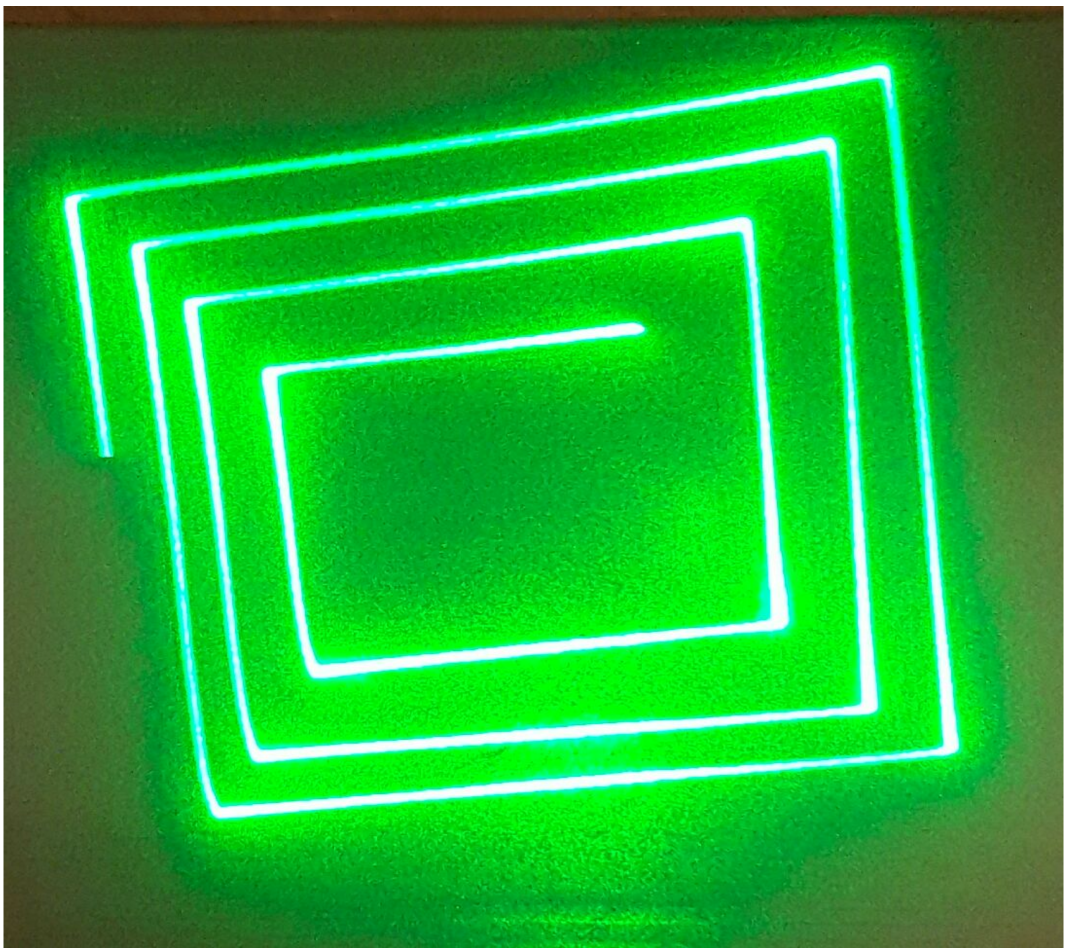
This figure shows the full hardware design on the Proto-Board

Appendix B



This figure shows our test pattern form the laser, a Box

Appendix C



This figure shows our test pattern fro the laser

Appendix D

This figure shows the LCD and keypad

∆

Appendix E

*This is the **first file attached** to this document, **compsysv4.txt**, which is our main assembly program*

Appendix F

*This is the **second file attached** to this document, **compsysv4.lst**, which is the lst file associated with our main program*

Appendix G

*This is the **third file attached** to this document, **compsysv.s19,** which is the s19 file associated with our main program and is used for programming the EEPROM*