

Lab 4 Write Up

Schematic

Please find the schematic for the lab 4 project as follows. Schematics shows LCD_ENABLE connected from SPLD to LCD. Figure also shows connections for EEPROM and LCD.

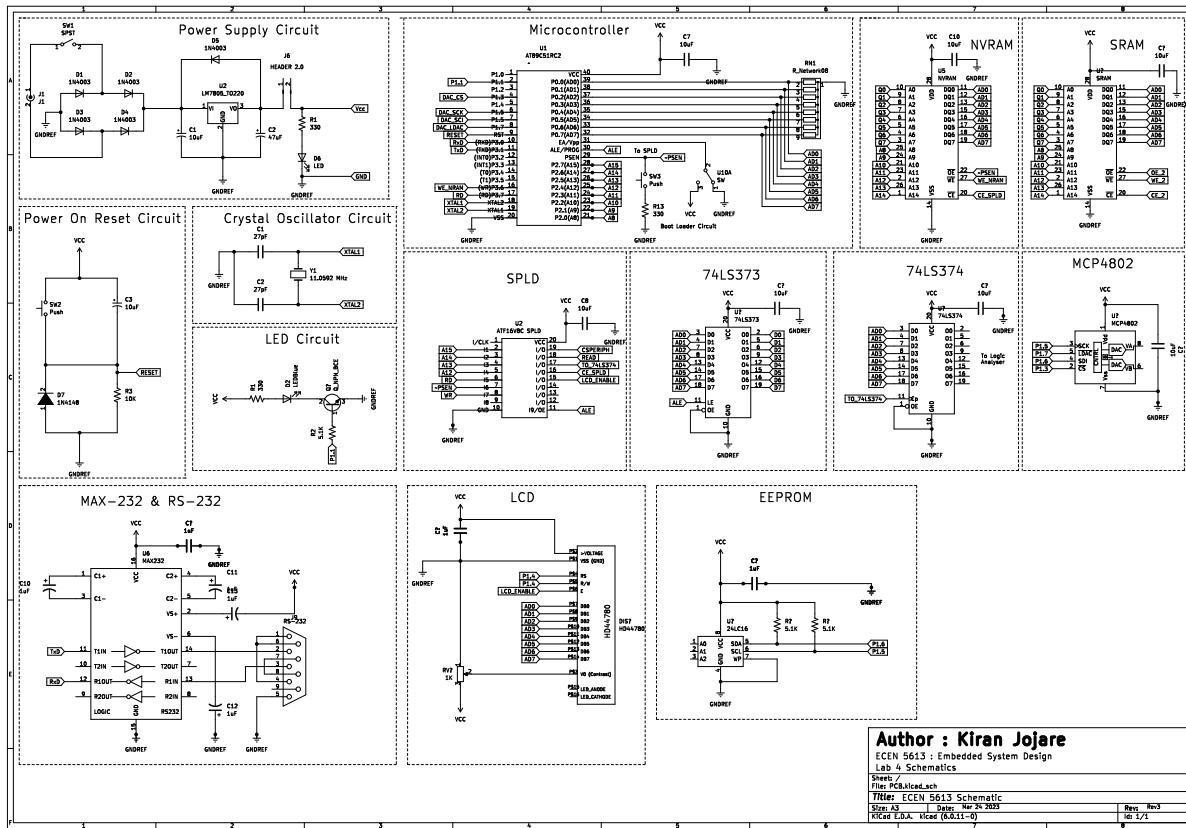


Figure 1: Board Schematic for Lab 4

Signoff Sheet

Front Sheet :

(Handwritten notes: "Test sheet" and "Lab 4 signoff sheet")

ECEN 5613	Lab #4 Signoff Sheet	Spring 2023			
You will need to obtain the signature of your TA on the following items in order to receive credit for your lab assignment. Print your name below, sign the honor code pledge, and then demonstrate your working hardware & firmware in order to obtain the necessary signatures.					
Student Name: <u>Kiran Navendrea Jojare</u>					
Honor Code Pledge: "On my honor, as a University of Colorado student, I have neither given nor received unauthorized assistance on this work. I have clearly acknowledged work that is not my own."					
Student Signature: <u>[Signature]</u>					
<u>Masae MD</u> 04/24/23					
Signoff Checklist					
Part 1 Elements					
<input checked="" type="checkbox"/> Pins and signals labeled and decoupling capacitors present on board <input checked="" type="checkbox"/> C code for EEPROM functional, contents present after power cycle <input checked="" type="checkbox"/> I ² C diagram/timing analysis					
Part 2 Elements					
<input checked="" type="checkbox"/> LCD functional, C code for basic LCD routines functional <input checked="" type="checkbox"/> LCD control signal timing meets specifications (logic analyzer trace/diagram, analysis) <input checked="" type="checkbox"/> Elapsed time stop, restart, reset to "00:00:0" <input checked="" type="checkbox"/> Good integration with previous code, all functions work, no irregularities					
Part 3 Required and Supplemental Elements					
<input checked="" type="checkbox"/> LCD Hex/DDRAM/CGRAM dumps, custom LCD characters, fun logo <input type="checkbox"/> SPI interface, logic analyzer trace, compare with I ² C. <input type="checkbox"/> ARM code development, 2 new features, ISR <input type="checkbox"/> PCF8574 I ² C I/O Expander, input, output, ISR					
<u>Masae MD</u> 04/24/23					
FOR TA/INSTRUCTOR USE ONLY					
Part 1 Elements					
Schematics, SPLD code Hardware physical implementation Required Elements functionality Sign-off done without excessive retries Student understanding and skills Overall Demo Quality (Part 1 elements)	Not Applicable <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Poor/Not Complete <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Meets Requirements <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Exceeds Requirements <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Outstanding <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Part 2 Elements					
Schematics, SPLD code Hardware physical implementation Required Elements functionality Sign-off done without excessive retries Student understanding and skills Overall Demo Quality (Part 2 elements)	Not Applicable <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Poor/Not Complete <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Meets Requirements <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Exceeds Requirements <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Outstanding <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Part 3 Elements					
Schematics, SPLD code Hardware physical implementation Required Elements functionality Supplemental Elements functionality Sign-off done without excessive retries Student understanding and skills Overall Demo Quality (Part 3 elements)	Not Applicable <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Below Expectation <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Meets Requirements <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Exceeds Requirements <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Outstanding <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TA/Instructor Comments <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>					

Figure 2 : Front side of sign off sheet

Back Sheet :

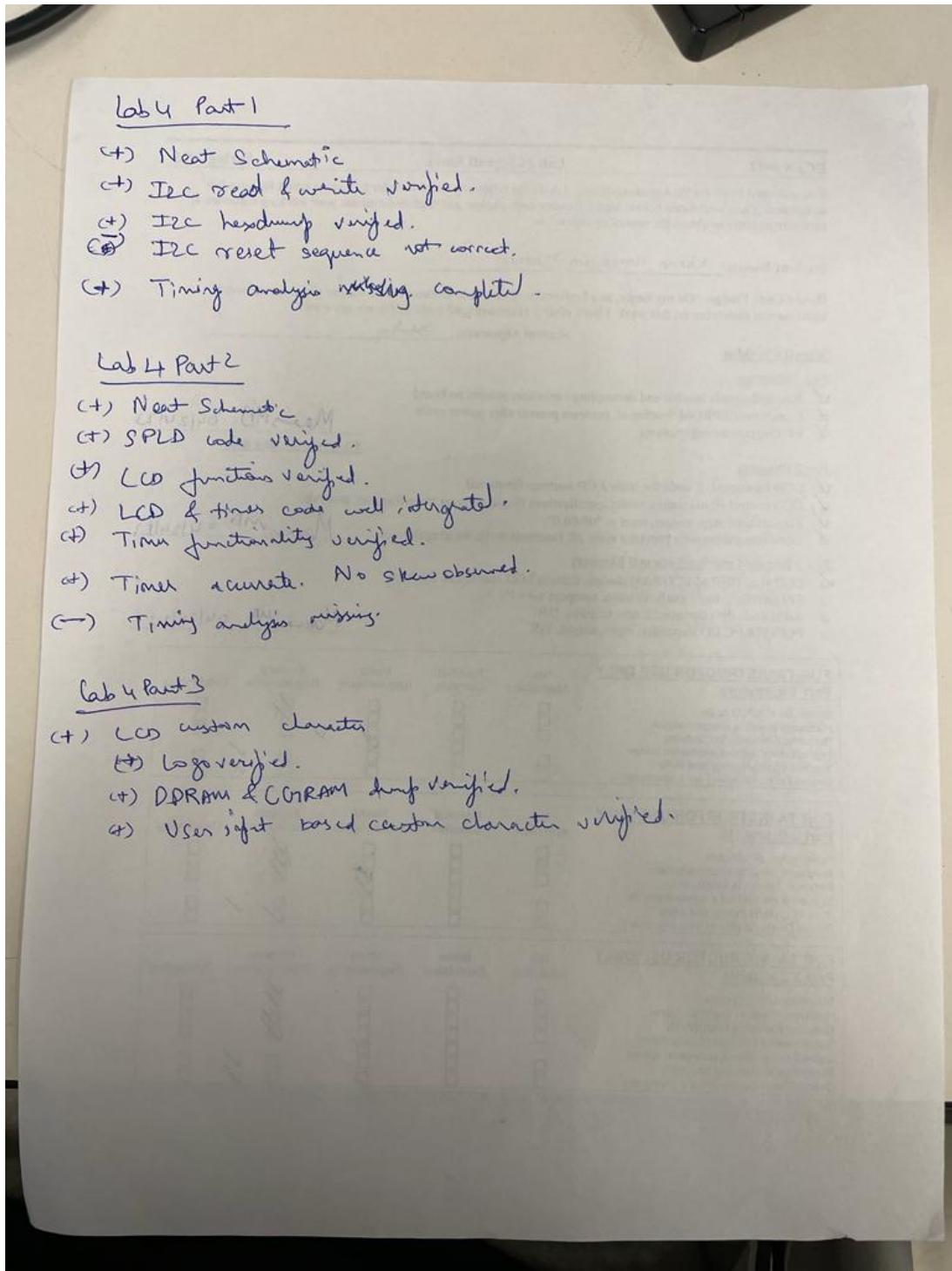


Figure 3 : Back side of sign-off sheet

Board Top

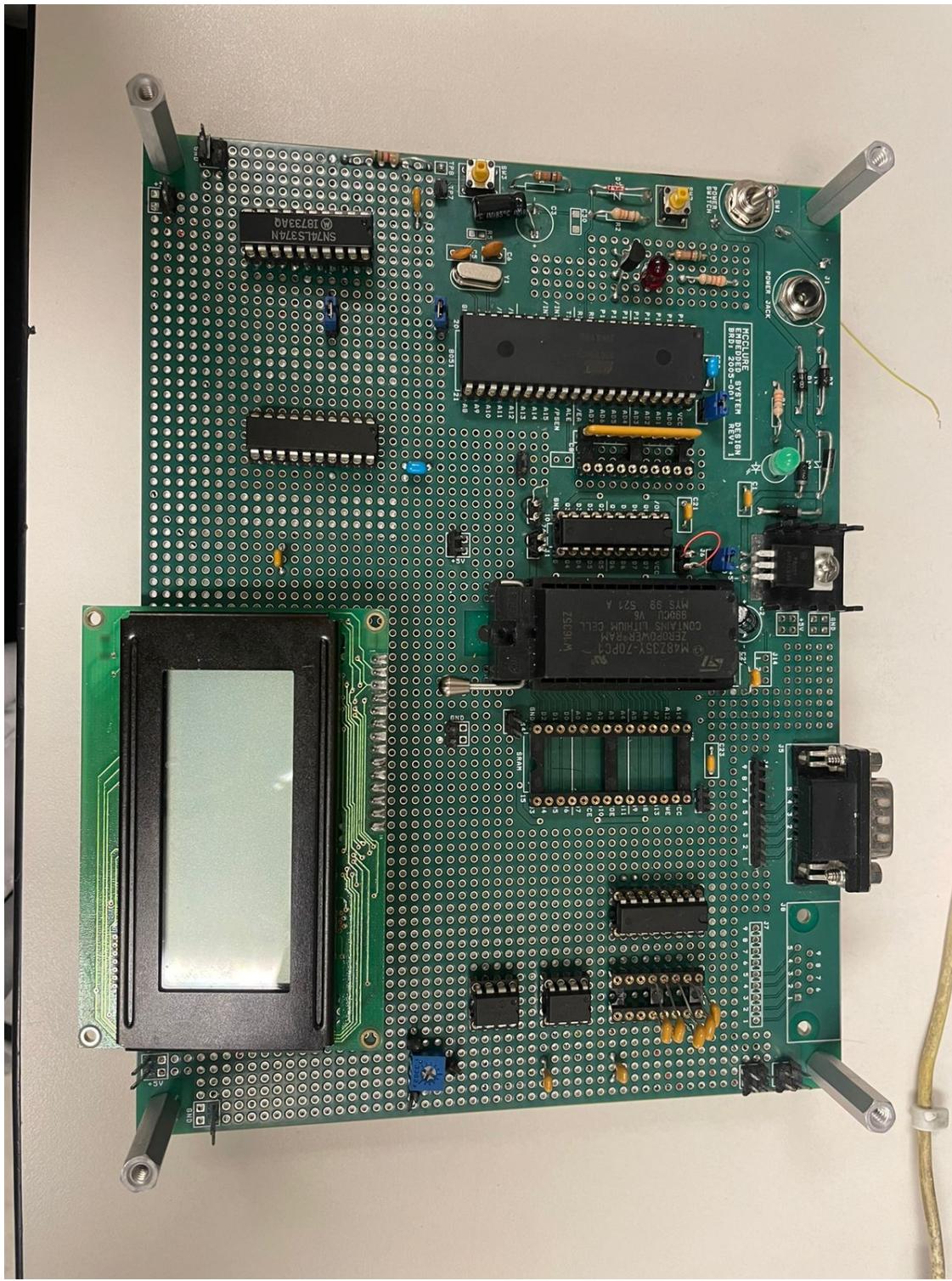


Figure 4 : Board Top Layout

Board Bottom

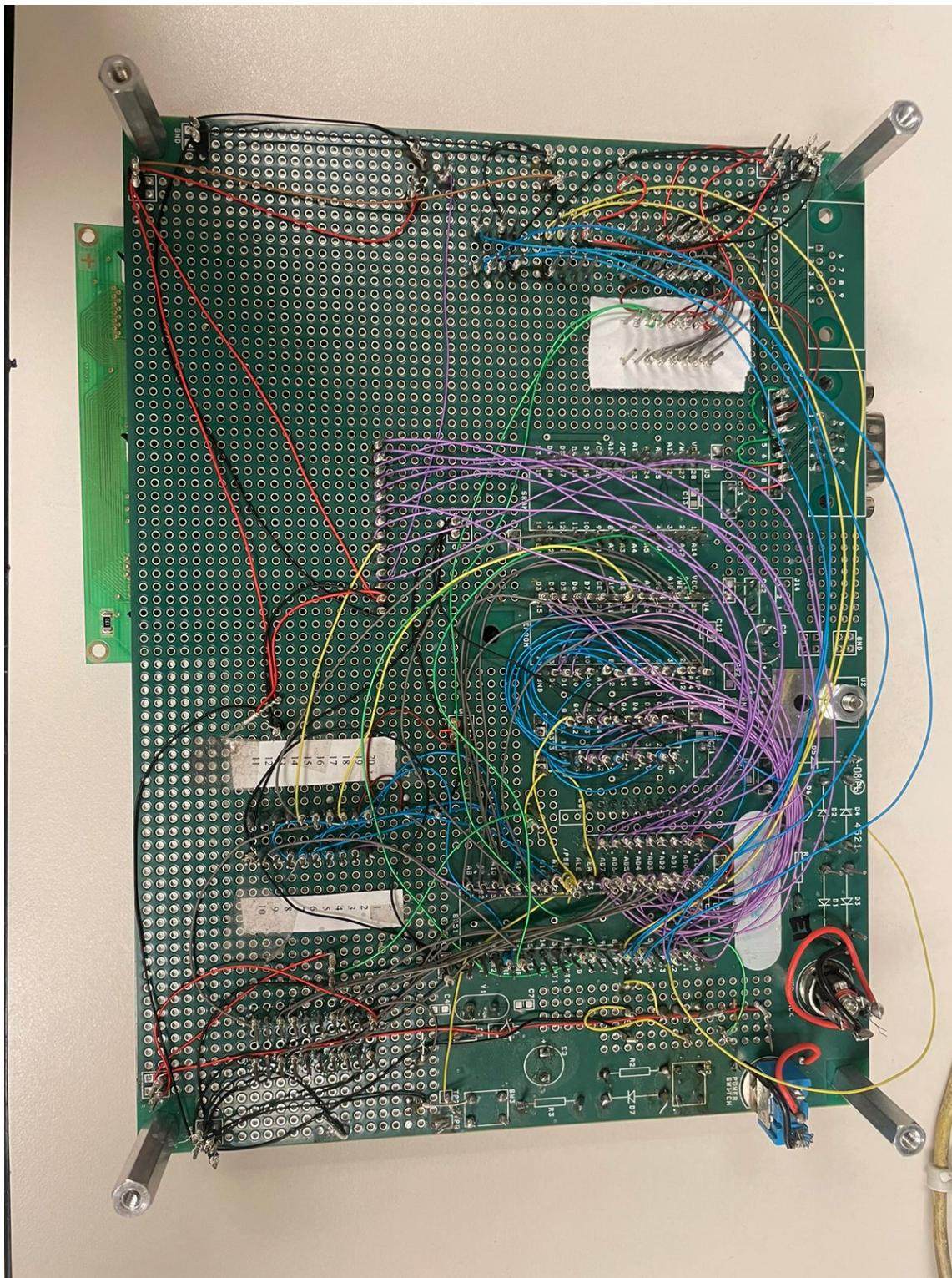


Figure 5 : Board Top Layout

Lab 4 Part 1

I2C EEPROM

Implemented an EEPROM I2C device driver with the ability to bit-bang write and read a byte at any EEPROM I2C address using function calls from C. Please see the UI used for the same as follows:



Figure 6 : UART Console for EEPROM Part 1

Figure below shows the Logic Analyzer for EEPROM write when data 10 is loaded in memory address 10H. Figure shows the write cycle operation as expected. We have a slave address followed by ACK. Data byte is followed by ACK bit.

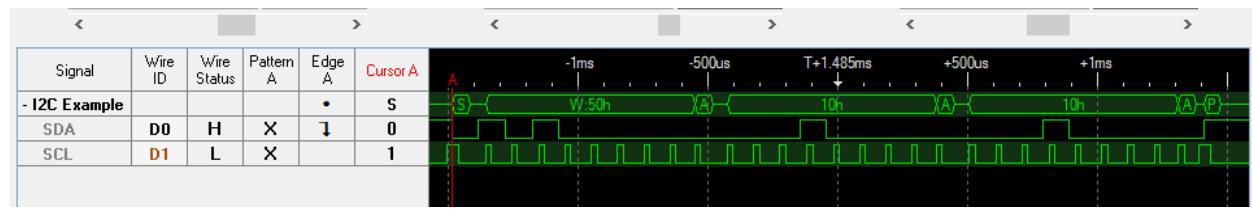


Figure 7: EEPROM Write

Figure below shows the cycle for EEPROM read which is behaving as expected when trying to read value stored at address 010H. The highlighted part shows the binary representation of a data read from address which is nothing but a 10. Read operations contains slave address followed by data byte.

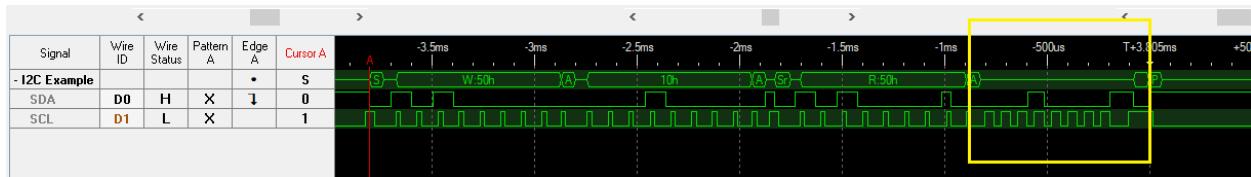


Figure 8 : EEPROM Read

Figure below shows the sequence of bit pattern when function EEPROM rest is called. The reset sequence involves sending a start condition on the I²C bus, followed by the device address with the R/W bit set to write (0), and then sending the bit pattern 8 SDA high, followed by an I²C clock, and one stop bit. The 8 SDA high bits serve as a command to reset the device.

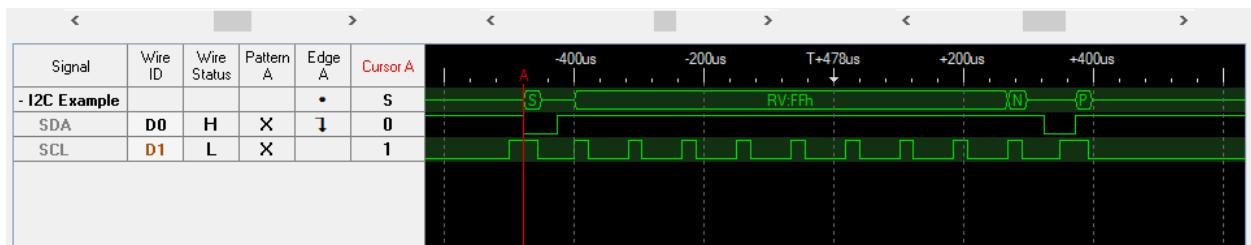


Figure 9 : EEPROM Reset

Timing calculation is performed on LCD and it perfectly shows that the rise time and fall time calculation perfectly satisfies the limits mentioned in I²C specifications and EEPROM data sheet. Observed fall time is 55 ns and observed rise time is 76 ns, which is well under the requirements of 300 ns for each of them

DSO-X 1102G, CN57276278: Sat Apr 22 07:59:01 2023

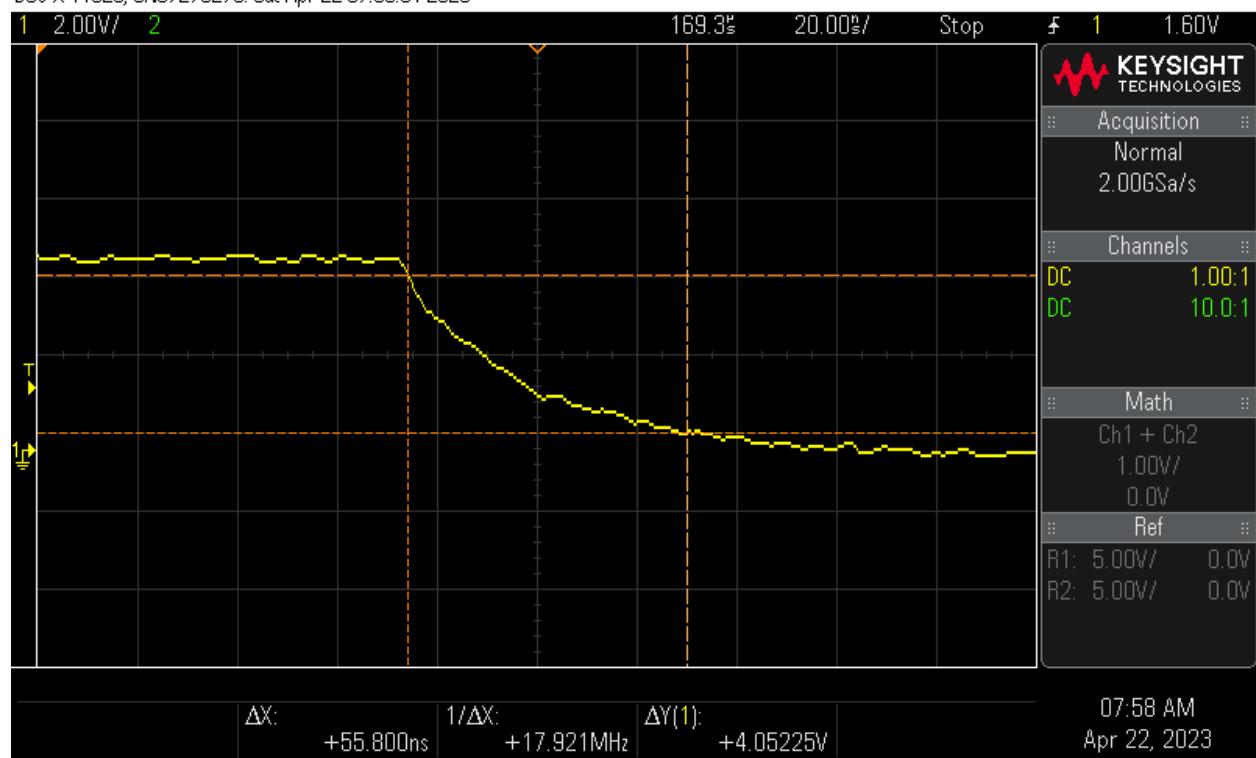


Figure 10 : Fall Time below 300 ns

DSO-X 1102G, CN57276278: Sat Apr 22 07:57:18 2023

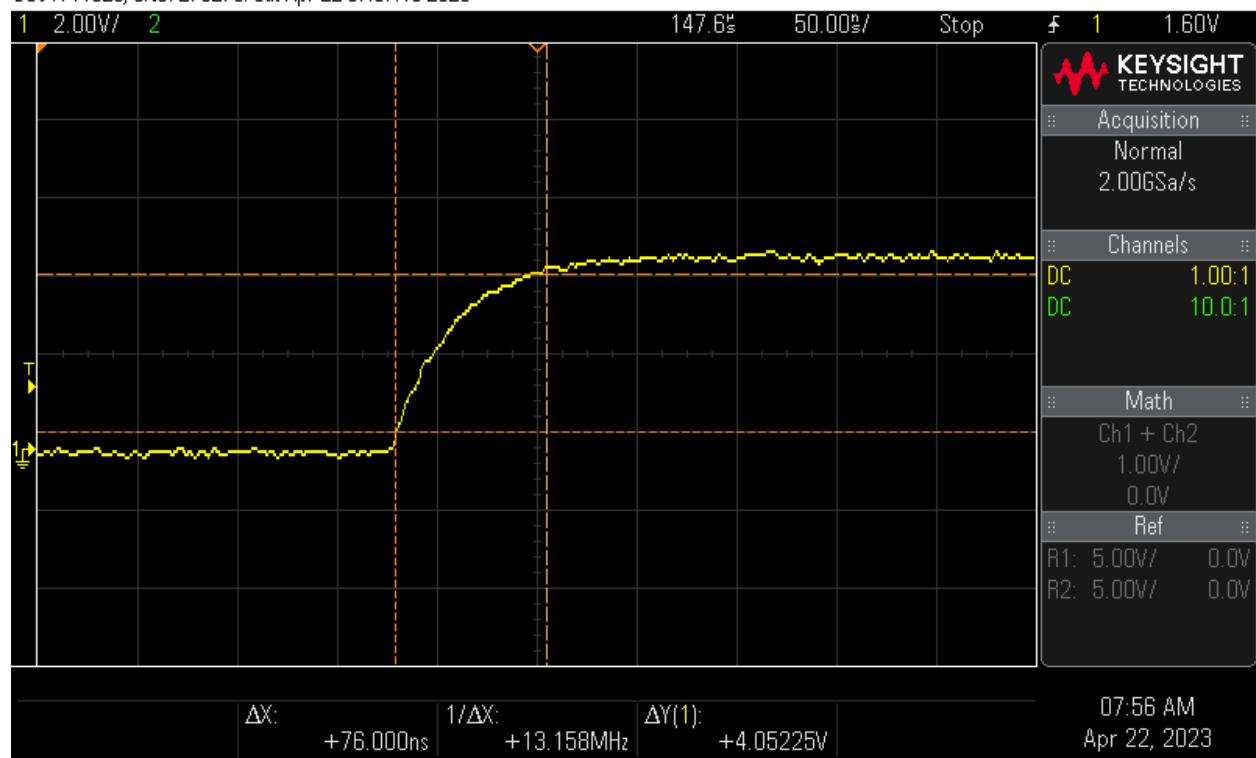


Figure 11 : Rise time below 300 ns

Figure below verifies that HEXDUMP works for 000 to 7FF.

Lab 2 & Lab 3 Part 3 Supplemental 1 :

The lab 2 deals with using LCD and implementing a full-scale timer with a real time clock implementing using Timer interrupt. Ideal output with author info and clock running is as follows .

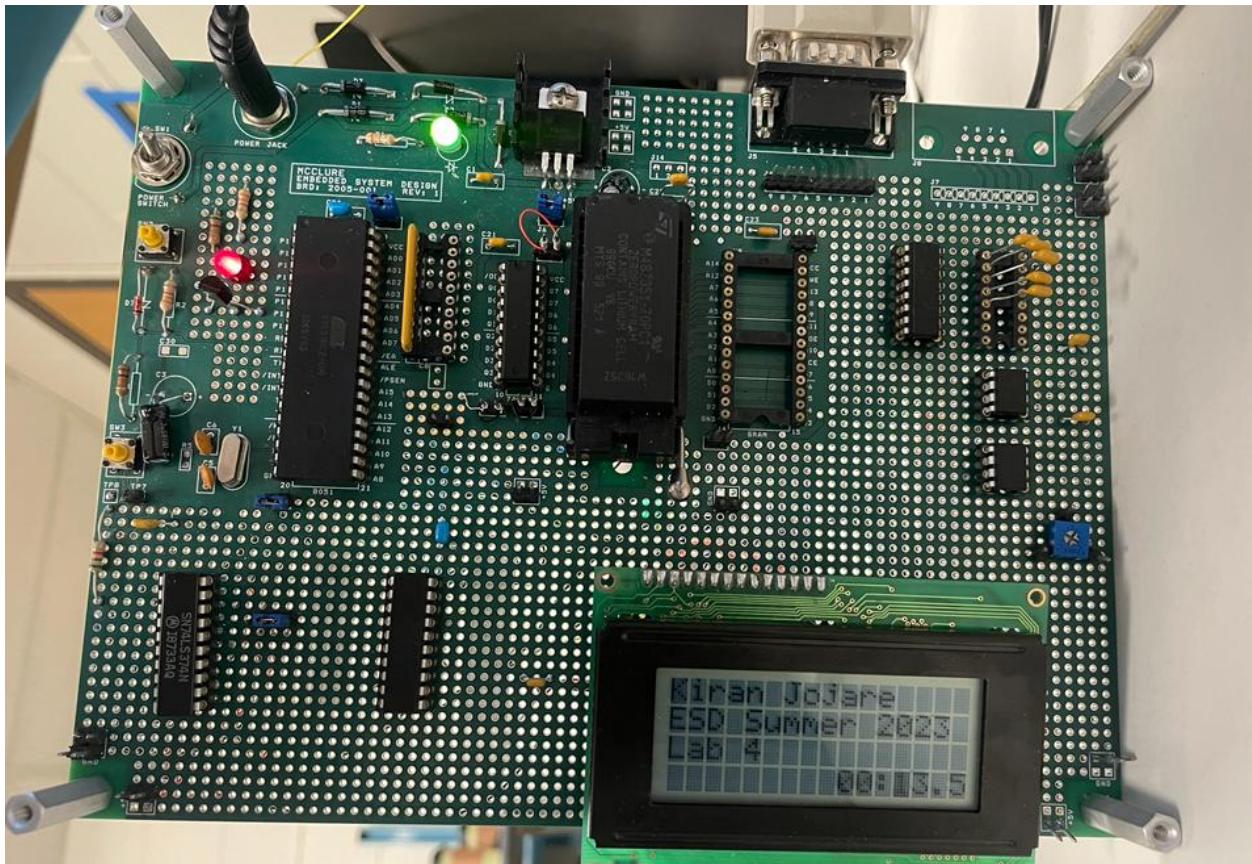


Figure 12 : LCD with author info and real-time clock

The ideal UART interface looks like this as shown in below picture. Which takes inputs to print character, string, got to address and (x,y). Also it includes options ot printf a custom character based on user inputs as well two built in customer characters as seen in photograph below .

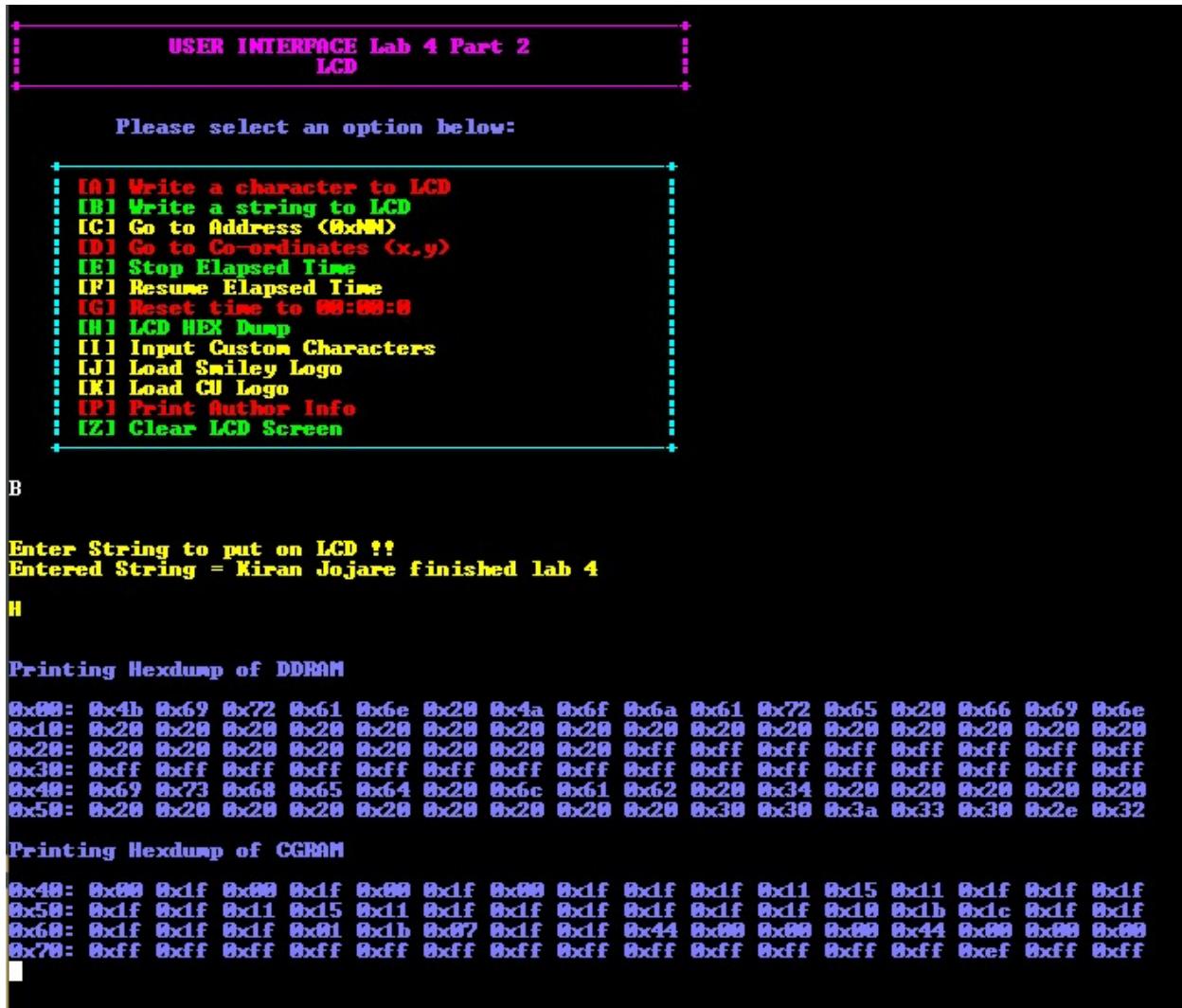


Figure 13 : UART interface showing HEX dump of CGRAM AND DDRAM

Above interface also shows the LCD hex dump after printing string “Kiran Jojare Finished Lab 4” as seen and verified using HEXDUMP of CGRAM and DDRAM.

Figure options to pause the elapsed time and reset the elapsed time has also been tested and verified as seen in sign off sheet.

Timing analysis of LCD's using logic analyser on E, R/W and RS signal has been proven to be under specified range as seen in the image below. Setup time is observed to be 1.085 usec and hold time is observed to be 1.095 usec as seen in the screenshot below.

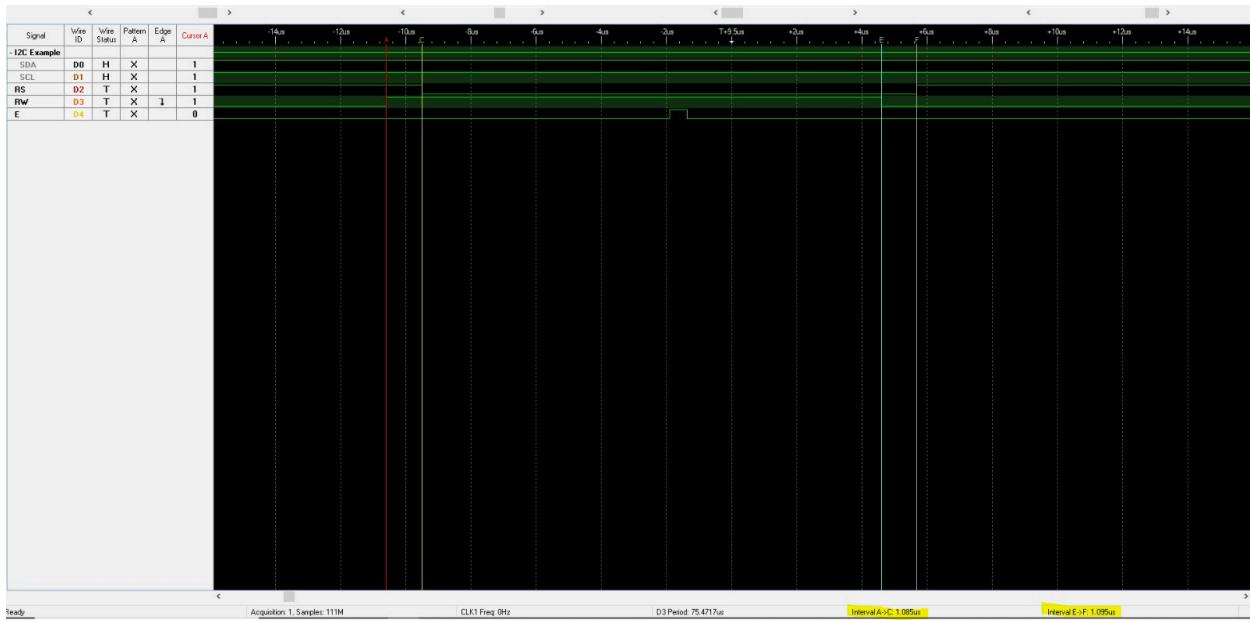


Figure 14 : Setup and Hold Time Timing Analysis

Custom characters has been implemented using all 8 characters codes and is visible as follows:

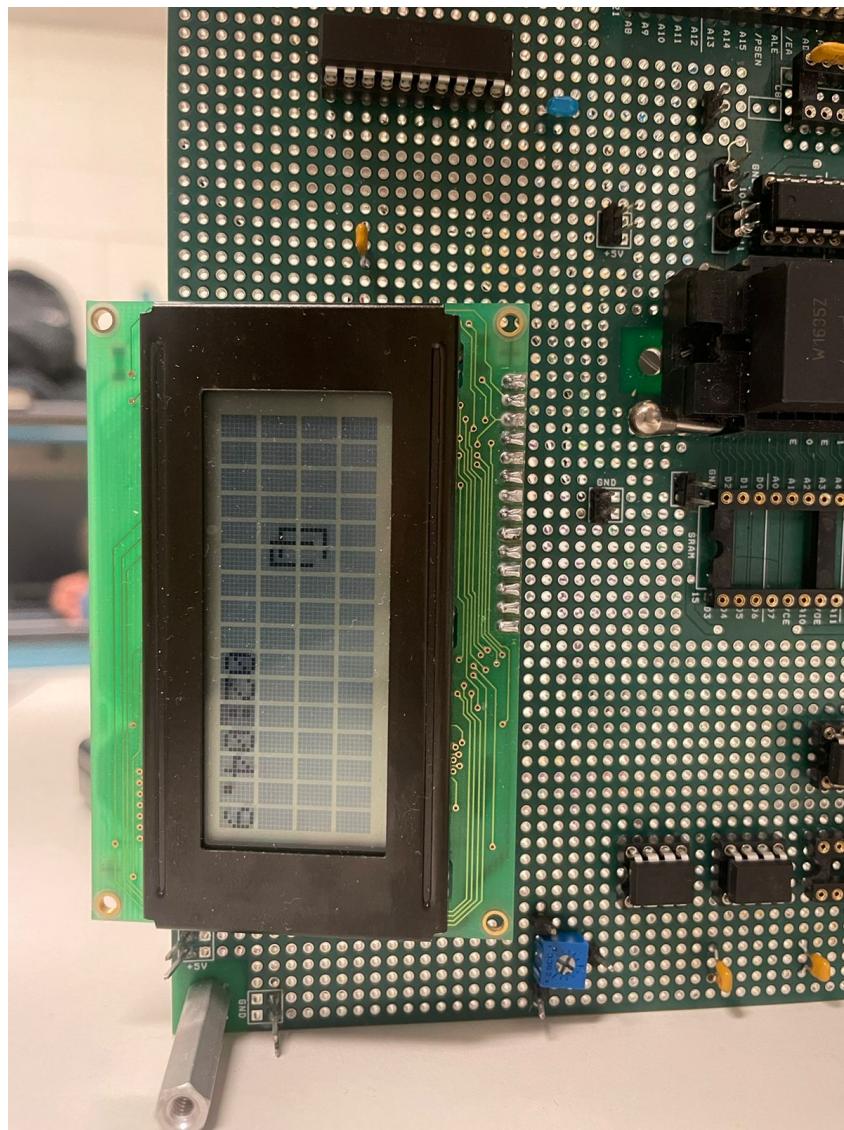


Figure 15 : Customer Character CU Logo

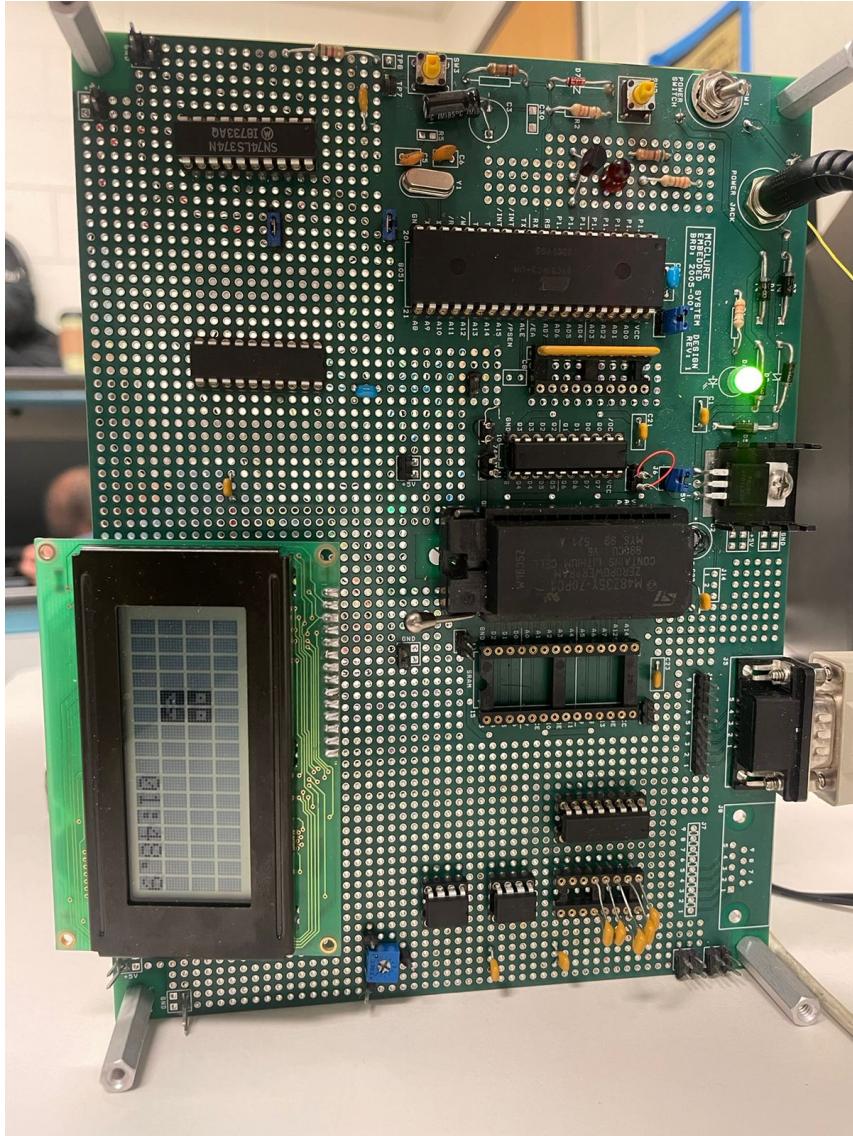


Figure 16 : Custom Character with Smileys Logo

The code also implements the logic where input arguments are taken from the user for each row value as well as customer character code and respective customer character is then printed on the desired X and Y co-ordinates.