EDUCATIONAL SMART BREADBOARD

By

Chinnies Chibuko

Mostafa Elkabir

Minseong Kim

Final Report for ECE 445, Senior Design, Spring 2018

TA: Kexin Hui

02 May 2018

Project 68

Abstract

This report discusses the design of an Educational Smart Breadboard. In this report, the design, trade-offs and the cost analysis of this Smart Breadboard is discussed. This report will show how such a breadboard can be used as both a chip-tester and a voltage-line reader. It also discusses the advantages of such a product and the motivation behind the project

Contents

1. Introduction

The first time any kid or college student learns circuits, he/she will encounter breadboards. A breadboard is a board for making an experimental model of an electric circuit. It is essentially a learning tool for students to gain more knowledge about circuits. Breadboards can be used for prototyping which is the process of testing out an idea by creating a preliminary model from which other forms are developed or copied. However, one of the big problems any student will face with breadboards is the issue of debugging. Debugging on the breadboard is particularly difficult. This is because of the relative small spacing between holes in the breadboard. As a result, wires tend to become clustered and make it a bit difficult to debug. Furthermore, there may be too many wires in a TTL circuit such that using a voltmeter can be such a headache. Thus, the need for the Educational Smart Breadboard.

The goal of this project is to help solve the problem of debugging on the breadboard, with educational emphasis. This is so the students can focus on actual debugging skills and not on the mess of clustered and intertwined wires. There are two main ways in which we intend to help make debugging easier for the student. One is checking the voltage of each row by just inputting row location. The second is by allowing a chip test. It will improve the quality of learning in introductory electronics classes such as ECE 110.

1.1 Background

Debugging on the breadboard could get a little bit difficult. The small spacing between the holes causes the wires to be clustered and messy. Sometimes the chips could be faulty without the student knowing and the student might end up taking out all the wires without knowing it is just a faulty chip. Our smart breadboard for that purpose functions as a quick chip tester, when a chip is placed in the chip testing area. The main idea is running all possible configurations to input pins and generate truth table then the processor will compare it to a lookup table. Also, the smart breadboard will function as the voltage and logical value checker when the user wishes to check what is going around for his/her specified pin. Surprisingly, there has not been really a simple debugging-purpose-extended breadboard. This may be because breadboard is not actually used for industry purposes, and we usually learn debugging by connecting it to multimeters, voltmeters and oscilloscope. But while these natural debugging methods are versatile, using them can be waste of time for digital circuitry purposes. For example, we do not wish to manually check all pins to check whether chips are functioning correctly. This project is particularly important because it will help the learning experience and would help improve the standard of teaching electronic circuits. It will make it easier for students to debug and concentrate on the functionality of the circuit

2 Design

As our end goal was to make a debugging Breadboard. Our design can be broken into its hardware components;
First for hardware, we have 5 main main units:

1. Voltage reading + supplying unit
2. Chip testing unit
3. Processor unit
4. Power supply unit
5. Display Unit

2.1 Block Diagram

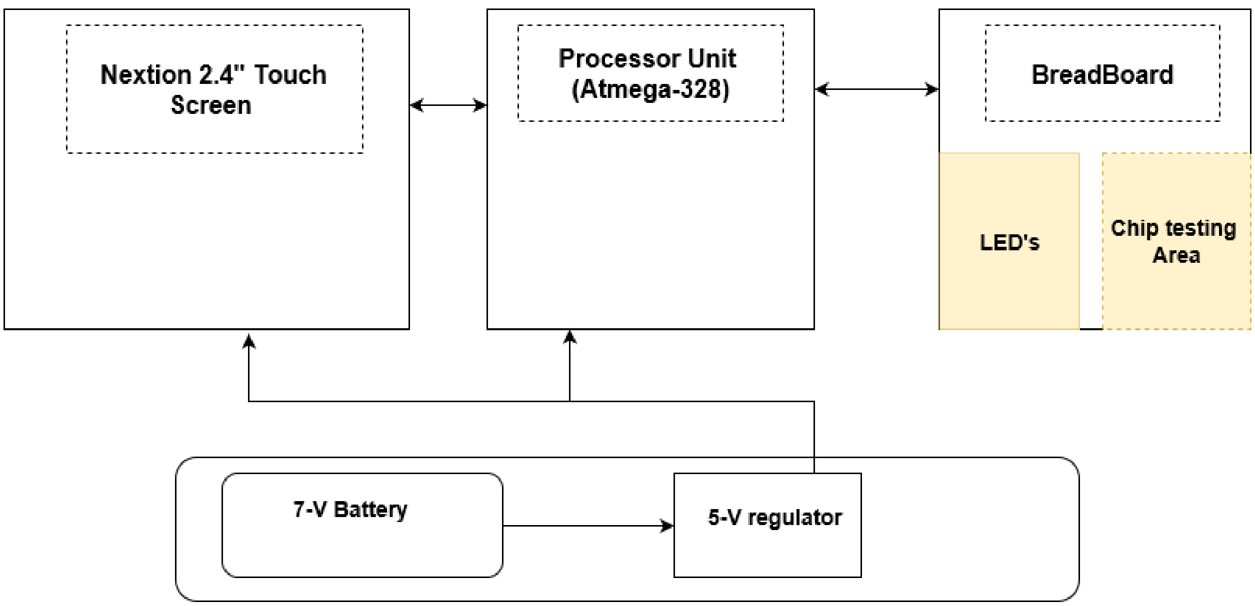

Figure.1 Block Diagram

2.1.1 Power Supply

Power supply unit is mainly comprised of 7V battery, 5V regulator circuit, where the circuit is constructed using Ls7805 and 2 capacitors connected for input and output of 7805 chip as illustrated in the below schematic

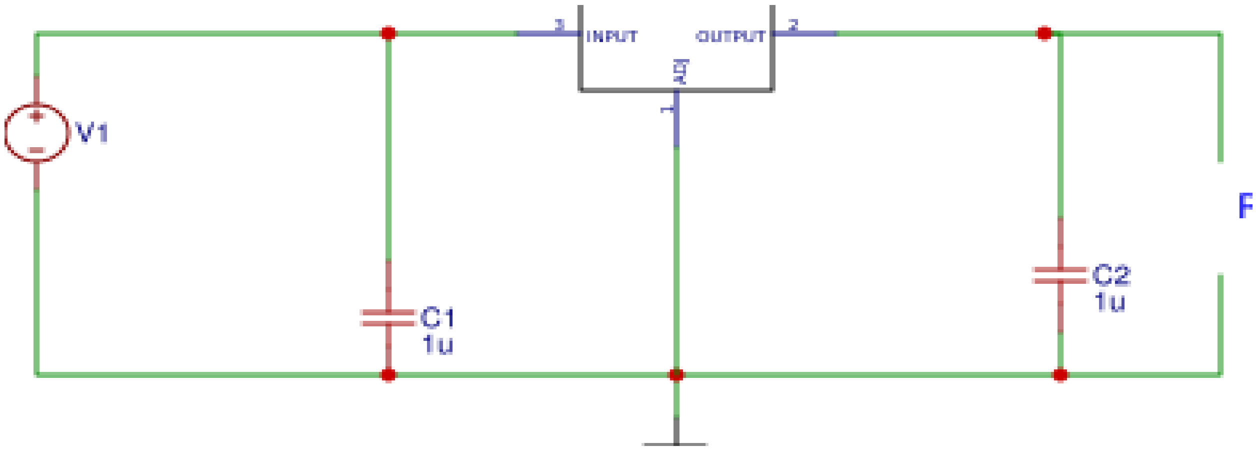

Figure.2. 5V Voltage Regulator Circuit

2.1.2 Processor Unit

The processor unit is constructed of Atmega328, capacitors, 16 MHz crystals, Low Pass filter circuit constructed using a resistor and a capacitor in parallel. The clock was made by connecting a 16MHz to pins 9,10 and connecting both legs to 20 pF capacitors, that helped run the whole system with the same clock. Something to note is that this is the only clock that can drive the system, if we use any other crystal value, the processor unit cannot communicate with the screen. Circuit describing this unit is shown below

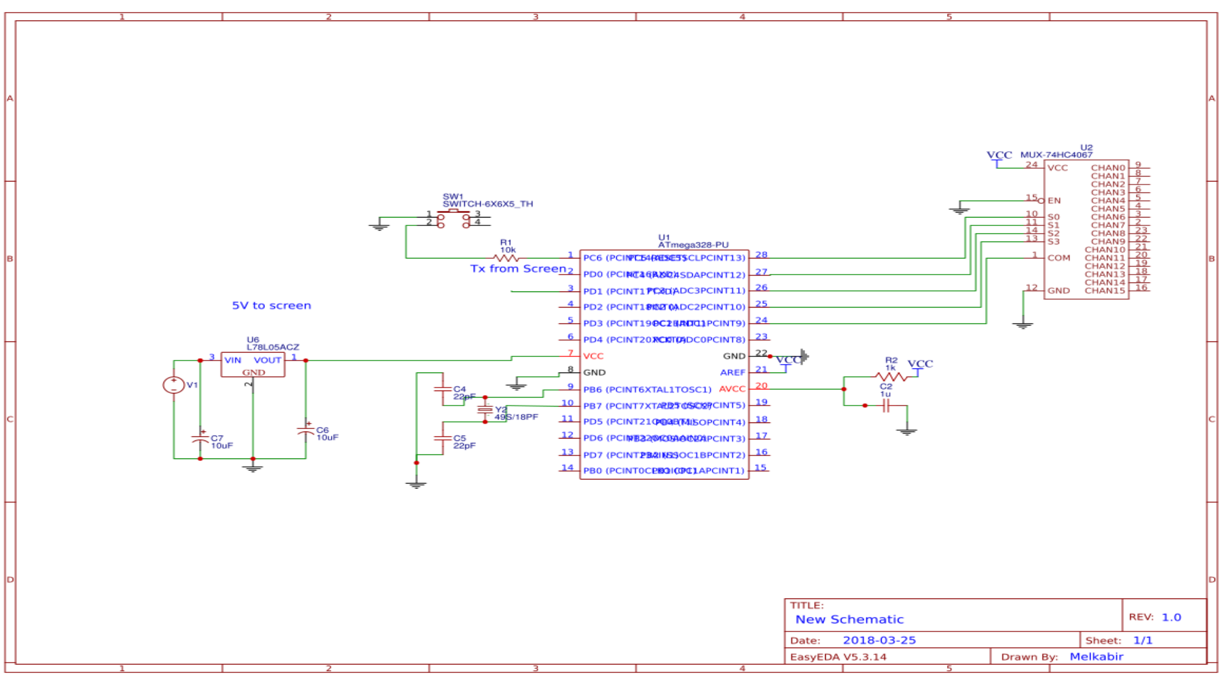

Figure 3: Processor Unit

As we can see in the schematic, pins 2 and 3 are used to communicate with the screen, pin 20 is connected to a low pass filter.

2.1.3 Chip Testing Unit

The chip testing unit is where the user can place any logic TTL chip that is already saved in the library and then the processor will run values to this unit and process the output.
This unit ran through two different challenges:

2.1.3.1 Original Design

The original design was comprised of 8 counters, 20 D Flip-Flops, 20 tristate buffers; because we want to send bits to 20 different pins with different unique values, where we need to supply Vcc, GND, Input and process the output. Where we used 8- 4 bit counters, 5 - 4 tristate buffers chips, and 3- 8 D Flip-Flop chips.

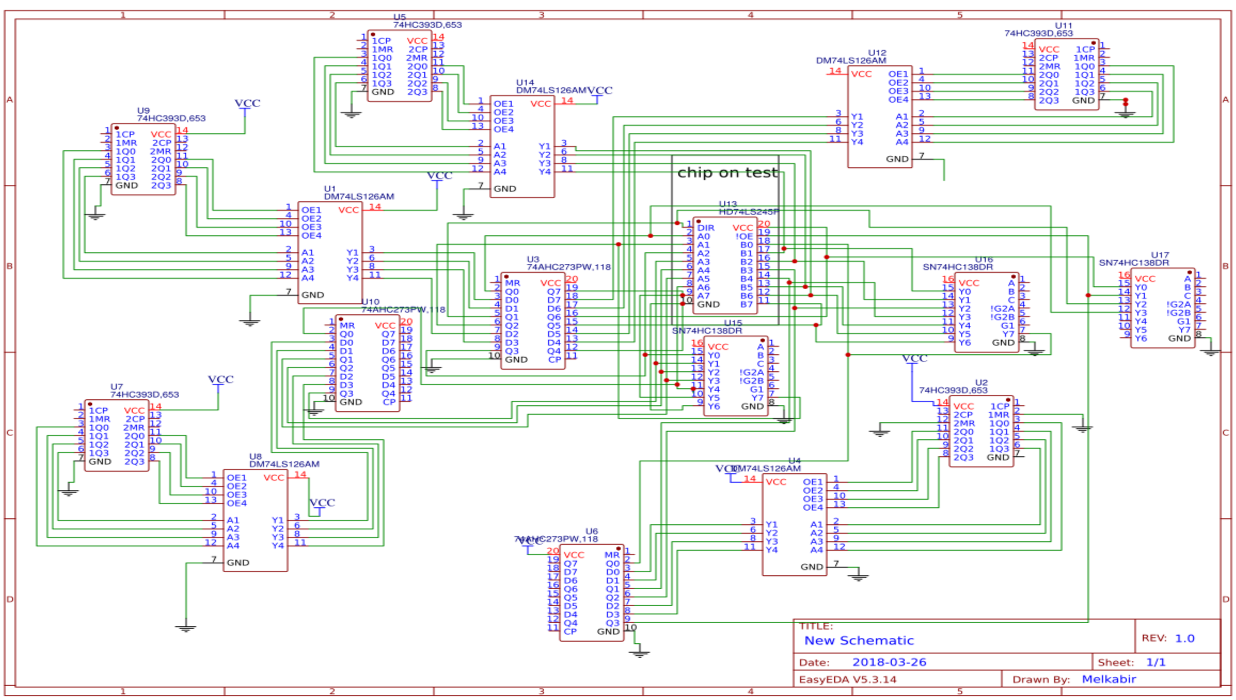

Figure 4. Original Chip Testing Unit

The original design can test any stable I/O chips with pins up to 20. The reason why we didn't manage to demo this for our project is simply because of soldering. We did not manage to construct a fully functional PCB, because of that a change in design had to happen.

2.1.3.2 Final Chip Testing Unit Design

In the updated design we connected the chip being tested to the processor immediately. This design has pros and cons. Well the positive side of this design is computation time is decreased dramatically from before where it went from 148 sec to approximately 1.5 sec to test 16 unique input pins lie a 16:1 mux. Now also we can supply Vcc and GND to the chip safely by simply applying 1's and 0's to the chip being tested.

2.1.3.3 Original Design Calculation

Approximately 10000 instructions, where each counter can reach 16 as maximum, then be reseted each time and have clock inputted to D Flip-Flops also taking care of the screen I/O operations; 60 nanosecond delay for each tri-state buffer chip, 6 nanosecond for each D Flip-Flop; 20 nanosecond for each 4 bit counter.
Given that the processor functions with a speed of 16 million operation per second;
10000/ (16*10^6) + (60 nsec*20)*10^-6 + (20 nsec*8)*10^-16 + (6nsec*20)*10^-6 = 2.105 msec for each input pin. Then to have 2^16 input configuration we just raise what we have to the power of 16 which gives 148605 msec that is 148.6 seconds.

2.1.3.4 Final Design Calculation

Where inputs are inputted to the chip being tested immediately, where we have 2^16 lines of code for input configuration then we process the output which is one pin that is being read for each different configuration -> 2^16*16*10^-6=1.5sec

As we can clearly see using this method decreased computation time dramatically, but what is the cons of this design:
First this design is limited to only 16 pins, second by changing this design, we had to change the voltage supplying and reading design as well, that is because the limited number of pins that can be used by the processor.

2.1.4 Voltage Reading and Supplying Unit

In this unit we want the user to be able to view all the voltage lines values just by a press of a button.

2.1.4.1 Original Design

As can be noticed from figure 5, we are using 4 16:1 analog muxes to represent all the 60 voltage lines in the breadboard where 16*4 = 64, which is more than enough. All those muxes get the same 4 select bits from the processor, then the output of those different muxes are inputted to an analog 4:1 mux where the 2 select bit are coming from the processor as well then, the output is fed to the processor.

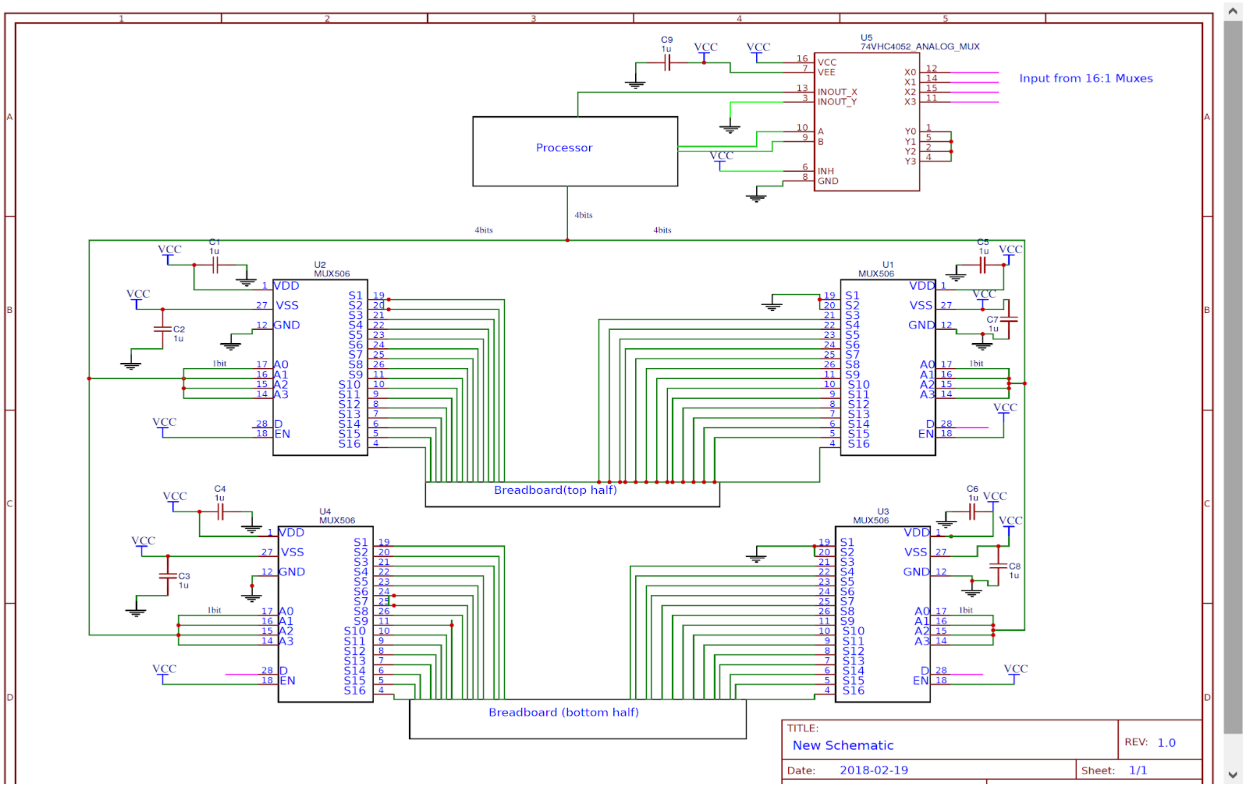

Figure 5. Voltage Line Reading

Original design cannot be tweaked to add voltage supplying as well, and because we changed the chip testing design, we had to change this design as well.

2.1.4.2 Final Design

In the final design we managed to supply voltages and show whether any voltage line is active or not. We managed to do this by connecting all the voltage lines to LEDs to show whether a line is active or not, and for supplying voltage we are using sets of 4-bit counters, and 2 tri-state buffers for each voltage lines. The first tri-state buffer gets the input from Vcc line, and the select bit is a unique value from the counter for each line, which is controlled by the processor, where the user can specify whether the voltage line is active or not, If active, then the tri-state buffer will get value of 1 inputted to that buffer, and the output of that buffer is fed into a second buffer, the idea of having a second buffer is to only pass the values after the counters reach its final state and have the correct output, then a common bit for all the second buffers are fed by the processor, where the processor sends 1 if the counter reach the number it wants it to. Figure 6 below have the schematic for the final design, where it contains 14 LEDs and can supply voltage for 8 lines.

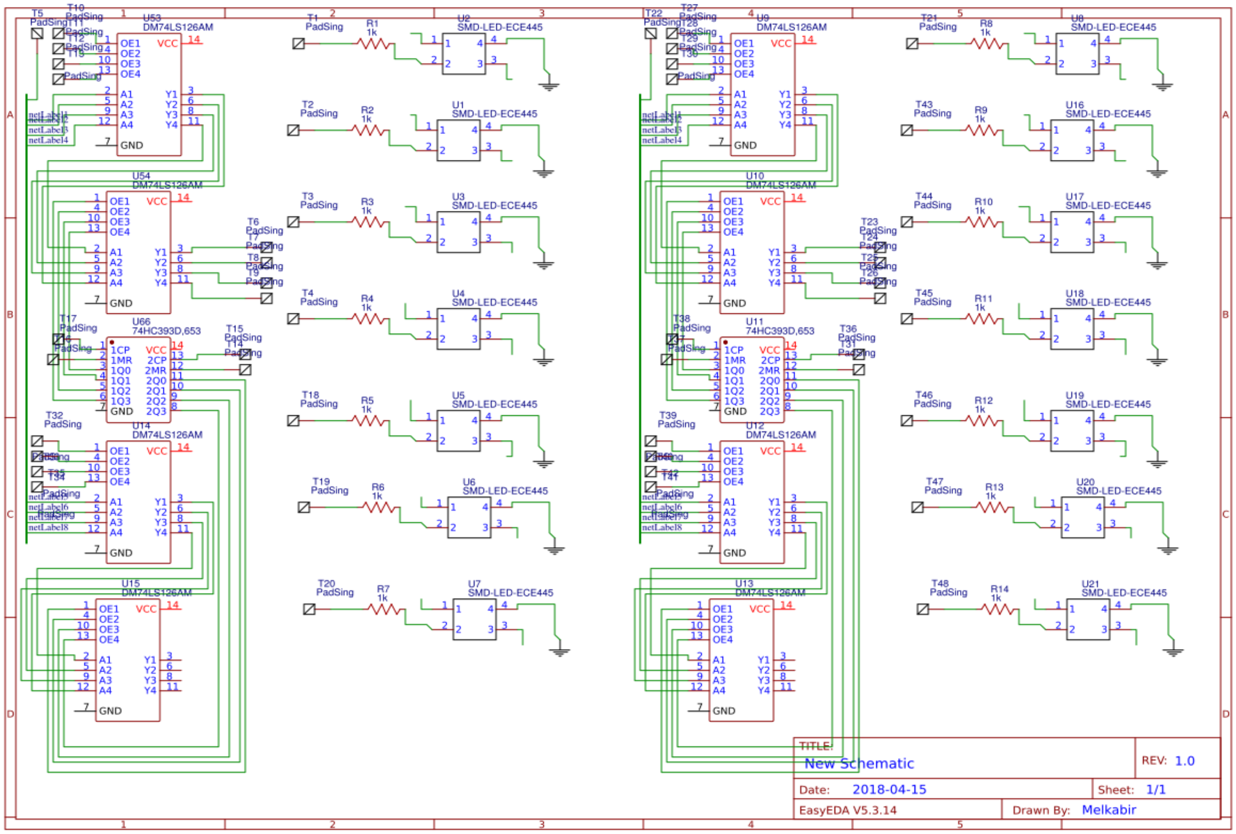

Figure 6 Voltage Supplying and reading

2.1.5 Price vs Power Consumption

- Original Voltage Reading Design:
  - 5_16:1 analog muxes
  - Price: 5*3.34$ = 16.7$;
  - power (based on data sheet) 5 * 150 = 750mW

- Updated Voltage Reading and Supplying design
  - 10 LED's, 8 tri-state buffers, 8 counters
  - Price: 10*0.3 + 8 *0.40 + 8*0.78= 12.44$
  - Power (based on Data sheet) = 8 * 90 mW + 8* 45 mW = 1080 mW

Note that the LED's power consumption is not added because it gets its Vcc from the user's power supply, where it consumes 0.2W for each LED, that's the reason why we fed Vcc to the main tri-state buffer for all voltage lines, and it runs the LED as well as because the processor has limitation of max current of 1A.The battery cannot drive the power to run all the LEDs, As a result we used the user Vcc line to feed to the processor.

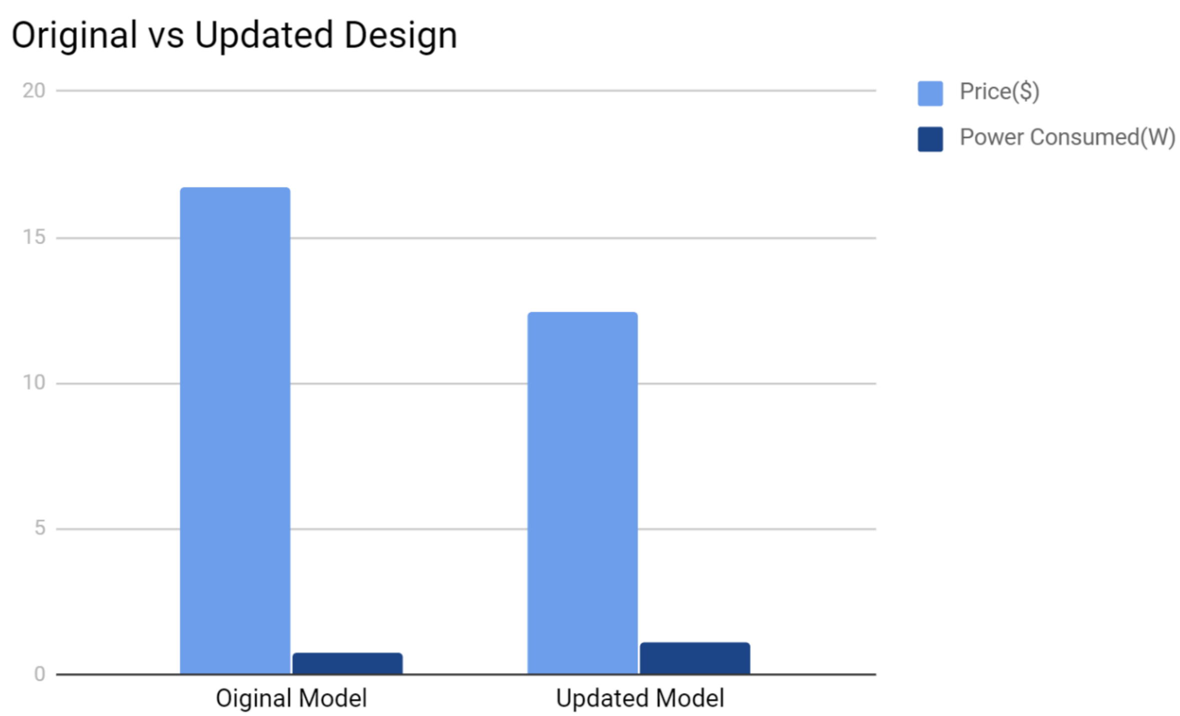

Figure 7: Chart showing power consumption & Price for both designs

As we see adding a feature of voltage supplying is a reasonable idea, as the power is reasonable with max of 1080 mW and price is actually dropped as we are not using analog muxes anymore.

2.1.5.1 Overall Design Power Consumption (Based on Datasheet)

Voltage line Reading + Supplying: Counters + tristate buffers+ Processor = 1080 mW

Chip Testing: Processor + tristate buffers + counters =

232.5 mW + 4*120 mW 4*45mW = 892.5 mW

Sleep Mode: 172mW

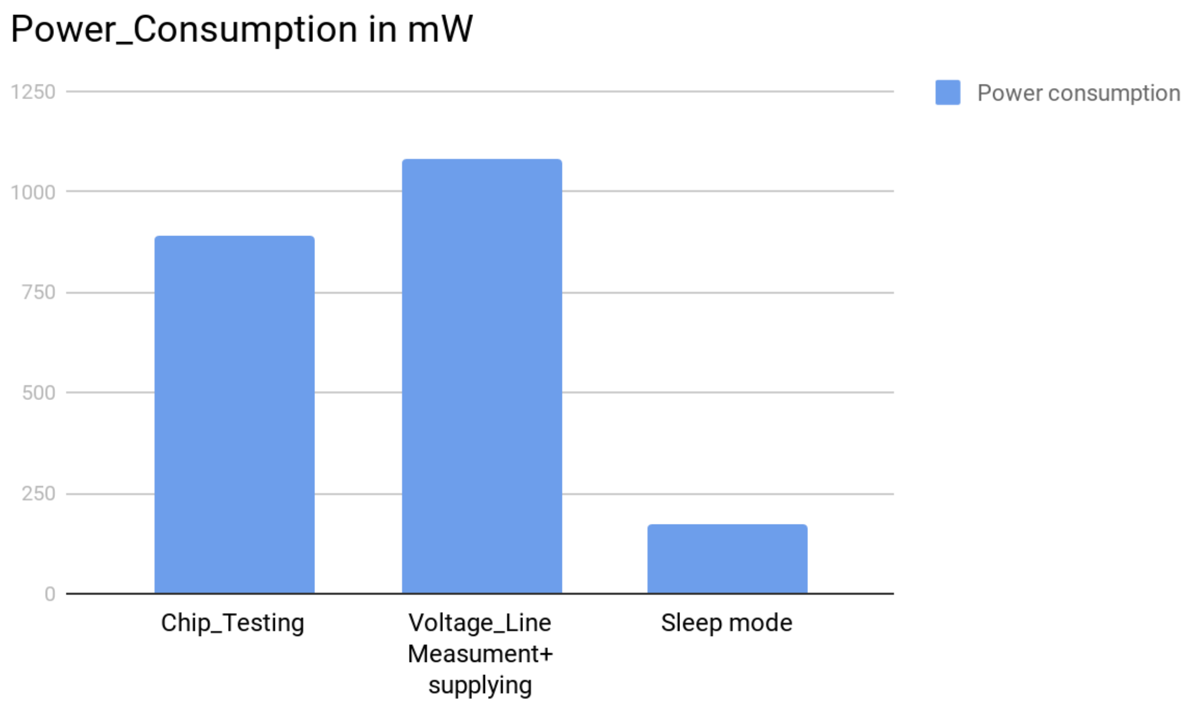

Figure 8. Power Consumption for different modes in mW chart

Figure 8 shows that voltage line measurement and supplying is consuming the most power. Also this chat is a motivation of adding a sleeping mode where the screen is off, and only the processor unit is functioning in a sleeping mode.

3 Design Verification

To re-state design principles necessary for explanations of verification, our circuit consists of the processor, screen and breadboard module. The screen module is connected only to the processor module, with both modules receiving and sending data. The processor module is connected also to the breadboard module, with both modules receiving and sending data. The breadboard module consists of the tri-state buffer, LED and chip testing area sub-module. The chip testing area sub-module is a dedicated separated from the typical breadboard area solely dedicated to the chip testing area. One LED is connected to every row of the breadboard, forming the LED sub-module. The tri-state buffer sub-module is used to properly switch between chip testing mode, voltage setting mode and ordinary breadboard operations. Ordinary breadboard operations should be possible in the breadboard area not dedicated to chip testing even when voltage is being set to some rows of the breadboard, as specified by the user interfacing through the screen.

All rows of the ordinary breadboard area excluding the chip testing area should be allowed to set voltage through the screen interface.

3.1 Breadboard: LEDs and Ordinary Breadboard Area Operations

As with other subsections to Section 3, full details will be in Appendix A.

The LED circuit is a fairly straightforward standalone LED circuit, and thus the only detail that would really matter would be the correct use of resistors – we used 220 ohm resistors for LEDs, and verified that LEDs work to scaling of 0-1.5V (low) and 4-5.5V (high).

When completed, our design allows users to set their breadboard into three different modes: chip testing, voltage setting, and ordinary operations without interference of the processor. Switching between these modes or ensuring that latter two modes are executed properly creates additional requirements, mostly with interactions with tri-state buffers. These tri-state buffers, along with counters, are necessary because number of processor pins is limited. Counters allow us to limit down number of processor pins used to control tri-state buffers from the processor, trading off time with number of pins. One input tri-state buffer and one output tri-state buffer is connected to each row of the breadboard, ensuring that outputs from the processor are ignored or accepted when appropriate, and outputs from the breadboard area are blocked from interfering with data pins of the processor when not appropriate and vice versa. This means that our verification focuses mainly on tri-state buffer control, input and output bits.

3.2 Breadboard: Chip testing

The chip testing area breadboard also has LEDs and the same tri-state buffers, and thus share the same requirements as other areas of the breadboard. Chip testing gains additional requirements in the processor program side and processor-screen interaction side. Furthermore, after screen/processor modules are verified, we need to verify whether our coded program in the processor implements our desired chip testing routines. Chip testing routine simply sends every possible input configuration to input pins of a test chip one at a time, and checks off expected output values against actual output values. Our verification mainly proceeds by checking input/output voltage sent to/from the test chip.

3.3 Processor

The processor can only do its job once it is programmed with something. Thus, verification of the standalone processor module proceeds by burning a program into the processor, configure the processor appropriately with a necessary power supply and an oscillator, and check whether output voltage is as expected. Then we configure the processor module with the Nextion screen module to verify.

3.4 Screen

The Nextion screen requires the processor to operate correctly, and thus the test necessarily proceeds by connecting with the processor. We first burn the program that is designed to interact with the screen through RX/TX serial ports and see if screen display transitions work as expected.

4 Cost

This section is going to discuss the fixed costs (costs of the parts) and the variable costs(cost of labor).The total costs of all this components is going to be outlined later in this section.First,the costs of the parts will be discussed

4.1 Parts

Below is a table which shows the cost of various parts that were used in this project.

Table 1:Table showing costs of various parts

Part	Manufacturer	Part#	Quantity	Cost
16:1 Analog mux	Texas Instruments	MUX506IPWR	5	$2.00
Tri-state buffer	Texas Instruments	SN74AHC126N	8	$0.78
Voltage regulator	ST microelectronics	L7805CV	10	$9.50
Breadboard			1	$5.69
AtMeg328	Microchip Technology	74HC4514	1	$4.00
LCD screen	Nextion	NX3224T024	1	$23.88
8 4 bit Counters		74HC393D	8	$0.40
D Flip-Flop		74HC273	6	$0.40
LEDs	Cree Electronics	CLM2D-CPC-CYbA03123	60	$0.14

4.2 Labor

The labor which comprised of three workers is estimated at $40/hour at 15 hours per week.This project took about 8 weeks to complete.In other words,it took about 120 hours to complete and at $40/hr for 3 people.It brings the total cost for labor to $40 * 3*120 which equals to $14400. Shown below is the table for Labor Schedule:

Table 2: Table showing the schedule of Labor

Week	Chinnies	Mostafa	Minseong
3/5/18	Ordering of parts	Making the processor circuit	Taking care of tri-state buffer and demultiplexer circuit
3/12/18	Assemble parts	Taking care of the muxes circuit	Write programs inside the processor
3/19/18	Spring Break	Designing PCB	Spring Break
3/26/18	Wiring of the socket	Designing PCB + testing	Writing test scripts for programs
4/2/18	Making the design layout of the breadboard	debugging + measurements	With inputs from Mostafa: check connections of the mux module with the rest of the logic module. And test the assembled design. Pass the program to Chinnies for final testing of the program in the processor
4/9/18	Program the processor	Making look-up tables + screen commands	User interface checks. Physical casing checks.
4/16/18	Finalizing our project		
4/24/18	Begin final report	Prepare final presentation	Prepare Final presentation

5 Conclusion

5.1 Accomplishments

We accomplished our main goal for the project, which is to allow the breadboard function as a chip tester and a voltage supplier and reader. The design processes for these functions are outlined earlier in this report. We are proud of the accomplishment that we made. We are at most proud of our ability to test up to 5 chips.

5.2 Uncertainties

When the processor unit was bought, we thought all we needed to do was to place it in the Arduino and we can load the code into it, but apparently, we needed to burn the bootloader inside it, then place it into the Arduino.

Another issue that we faced is that we used a 20 MHz crystal for the processor unit, but the screen did not work, we thought that the screen only works with the Arduino unit for some unclear reasons, we later changed the crystal to be 16MHz and it worked instantly.
Another thing is that we thought all GND is the same, we did not know that GND is only a reference, that is because when we connected the power supply to screen and USB to Arduino the screen got burned and printed white page, while when we used power supply for the processor and the same Vcc and GND to screen it worked immediately.

5.3 Challenges

One of the main challenges that we faced is Soldering. Our inability to solder properly and efficiently is a huge problem we faced. It caused our inability to use PCB, we then resorted to the use of perf board which led our project to look a little bit messy. Another challenge we had is the cost of parts, we did not estimate the costs of parts to be a little bit high. We projected to have a cheap project but that wasn't the case for our project.

5.4 Ethical considerations

Chip tester and voltage setters create the following ethical problem, related to a safety problem to be discussed in the next subsection. The reason why we have the dedicated chip tester and voltage setter are to allow users to easily debug and operate a TTL circuit. However, by advertising our product this way, this creates illusion that our circuit does not harm their chips.

Suppose you have an inverter chip instead of a NOR chip and ask our smart breadboard to test whether a chip is a proper NOR chip. Our tester will simply test all input configurations by sending bits to purported input pins that may not be input pins. Thus, we risk damaging chips this way. Our voltage setter risks the same, as users can specify rows of the breadboard connected to output pins of some chip to be set voltage.

5.5 Safety Statements

There is a number of safety issues, firstly the battery could be faulty. If there is a huge current drain by the circuit of faulty connections by the user smoke might rise and the user might inhale which would lead to a variety of health issues. As the user will supply the power for his own circuit, a number of safety issue might occur, one example is having a short circuit or applying too much voltage to the circuit. This might damage the chips they are using or any other electric tool such as applying too much voltage to a BJT which could burn the metal. The ethical codes of IEEE [6] and ACM [4] will be followed strictly when implementing this project. However, code 1.2 of ACM code of ethics states "Avoid Harm to others ", harm might occur in wrong usage of the circuit or applying too much power of some sort. In the other hand the capability for students to learn from this tool is beyond that. Also the safety issues rising come from the wrong usage of outside tools. We should avoid plagiarism because it is immoral and academic theft.All ideas and written documents should be original and not copied from any external source

Our design is considerably safe as it uses a safe range of voltage and current. Our projects run on a 5V, processor, and 6.5 V screen. One ethical issue might rise is that students become dependent on this project, and do not follow other methods of testing. Students may use our project as the main testing tool without being supervised by a professional. This might cause harm for young children and new users as well. In the times of huge amount of computation is done, the processor, or any of the parts might be damaged, or even burned in the process. That will result in sometime a chemical reaction and smoke might rise. The voltage regulator have a high rate of watts being dissipated, which it needs a heat sink. That much power could be dangerous for young children when heat sink fails.

5.5 Future work

For our future work, we plan on designing the board to debug if there is a shortage in a line. Also, we have a plan to design to be able to test more chips including the complex ones like the sequential chips. Due to our poor soldering efforts, we couldn’t use PCBs for this project. Thus, for our future work we plan to make our whole design in one compact PCB

References

1. Amazon.com. (2018). HiLetgo 2.4 Inch TFT LCD Display Shield Touch Panel ILI9341 240X320 for Arduino UNO MEGA. [online] Available at: https://www.amazon.com/HiLetgo-Display-ILI9341-240X320-Arduino/dp/B0722DPHN6/ [Accessed 2 May. 2018].

2. Arduino. (2018). Arduino Uno Rev3. [online] Available at: https://store.arduino.cc/usa/arduino-uno-rev3 [Accessed 2. May, 2018].

3. Staples, G. (2018). Arduino Power, Current, and Voltage Limitations. [online] Electricrcaircraftguy.com. Available at: http://www.electricrcaircraftguy.com/2014/02/arduino-power-current-and-voltage.html [Accessed 2 May 2018].

4. Acm.org. (2018). ACM Code of Ethics and Professional Conduct. [online] Available at: https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct [Accessed 2 May. 2018].

5.Ee-classes.usc.edu. (2018). SN74LS150 data sheet. [online] Available at: http://ee-classes.usc.edu/ee459/library/datasheets/sn74150.pdf [Accessed 2 May. 2018].

6. Ieee.org. (2016). IEEE Code of Ethics. [online]. Available at: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 2 May 2018].

Appendix A Requirement and Verification Table

2

Table 2: Requirement and verification for LEDs and ordinary breadboard area operations

Part	Requirement	Verification
LEDs	Do not noticeably light up in case voltage of row is in 0-1.5V and light up in case voltage of row is 3.5V or upper in any circumstance	- A. The LED circuit is simply LED-to-resistor-to-ground, and thus can be tested separately for functionality. This test was done with external power supply. This was verified. B. When our circuit is built entirely, we can set voltage of each row using screen. We verified operation of our LEDs this way. C. During chip testing, it really does not matter what LEDs show. Thus, A and B are sufficient.
Ordinary bread-board area	Users should be able to do any operation they like on the breadboard without any interference when they are not testing a chip.	A. Each row of the breadboard is designed to be connected to one “input” tri-state buffer and “output” tri-state buffer. Each tri-state buffer controls whether input should be taken from the processor or output should be sent to the processor. We check on tristate buffer inputs and outputs that ensure switching between chip testing, setting voltage on each row and ordinary breadboard operations are done smoothly. We also did live-testing by adding chips and wiring them together in the ordinary breadboard area to see if they operate without any trace of additional modules. This verification was also a success.

Table 3: Continuation of requirement and verification for ordinary breadboard area operations and voltage setter

Part	Requirement	Verification	Verification Status
Ordinary breadboard area	Users should be able to do any operation they like on the breadboard without any interference when they are not testing a chip.	(Continued) When chip testing feature is enabled, rows of the breadboard that are not chip testing area should have tri-state buffers that are disabled. When the breadboard module is not on the chip testing mode, if some row is specified by a user using the screen to have a certain voltage, then the input tri-state buffer of that row should be enabled, with the output tri-state buffer disabled. Otherwise, both tristate buffers should be disabled. This is tested with a hard-coded program not depending on the screen. For extra debugging, we checked counter inputs and outputs connected to tri-state buffers used to set control bits of tri-state buffers to reduce the number of pins of the processor occupied for this switching operation. These verifications were necessarily done with the screen interaction, as the screen and the processor is the main interface users use to specify modes of the breadboard: whether testing chip or setting voltage and/or running ordinary circuit operations on the ordinary breadboard area. When this interaction misbehaved, we tested directly by simulating processor outputs and checking whether our breadboard module operates as it should. Then	Y

		we tested the processor with a hardcoded program to see whether the processor/program is properly taking inputs and sending outputs to input tristate buffers. Then we verified the processor and screen interaction As said before, these were all successfully done.	
Voltage setter	Users should be able to set voltage successfully to each row specified through the screen module.	This is verified through LEDs connected to each row of the ordinary breadboard area and directly checking voltage through a voltmeter. As in above. We first tested through the screen, and if anything goes wrong, we directly simulated the processor outputs sent to input tri-state buffers. And then we checked back screen-processor interactions. All these was	Y

		verified. There was no gray area that is neither low or high in digital voltage for LEDs.	
Tri-state buffers	Power voltage of tri-state buffers should be regulated to 7V instead of 5V for other components. 4	This was done through a voltage regulator set to provide 7V within 0.5% tolerance. This was directly verified through voltmeter and there was no operational problems within this tolerance level.	Y

Table 4: Requirement and verification for the processor and screen module

Part	Requirement	Verification	Verification Status
Processor	Processor should operate independently of other modules, along with voltage regulators.	We hard-coded our processor with some programs written in Arduino programming language and checked voltage of output pins of the processor. This verification was successful.	Y
Screen	Screen sends outputs to processor, receives inputs from the processor successfully and displays pages as desired, commanded by the processor.	This could be tested successfully separately from other modules. Since there are four pins of the processor that the screen is connected to, and the screen is connected to nowhere else, verification process was a simple one. Hardcode a processor with a program that interacts with the screen, and check whether screens display things properly. For chip testing, one can provide voltage coming from the	Y

		breadboard module to the processor instead with external power supplies by slowing down running speed of the program. This verification was done successfully.	
Processor	Processor and screen voltage should be regulated to 5V within 0.5% tolerance level by voltage regulator. Plus, current needs should be under operational limits of power supply.	This was verified by running the processor-screen module integrated with voltage regulators and checking voltage with voltmeter plus checking whether the screen operates correctly. Both screen and processor the same voltage supply, along with tri-state buffers, and this raises the current consumption questions, but maximum theoretical current needs were so negligible relative to any imaginable current upper bound of either batteries or power supplies. These calculations were verified with multimeter.	Y

Table 5: Requirement and verification for chip testing area

Part	Requirement	Verification	Verification Status
Current needs	When testing a chip in the chip testing area, voltage and currents are provided by the processor to the chip. This means that we have to check whether maximum current that can be provided by the processor is not being reached.	Theoretical calculations showed that current needs are negligible. The processor provides voltage to clocks of counters and to input and power pins of a chip being tested. We tested theoretical calculations by using multimeter, and multimeter value came under maximum theoretical current needs.	Y
Tri-state buffers	Input tri-state buffers should be enabled for input pins of a chip being tested with output tri-state buffer turned off. Output tri-state buffers should be enabled for output pins of a chip being tested in the dedicated chip testing area, with input tri-state buffers turned off. When chip testing is done, all tri-state buffers connected to rows of the dedicated chip testing area should be turned off.	This was verified first with the processor module not attached to the screen module that is hard-coded with a program that runs chip testing routines. We tested voltage of control bits, inputs and outputs of tri-state buffers. Verification was successfully done. And then we connected the screen to the processor and hard-coded a program to the processor so that it interacts with the screen and executes chip testing routines when some buttons were pressed in the screen. This was also verified successfully.	Y
Program	The above assumes that the program is written correctly. Thus one first needs to verify whether the program is running as intended.	The chip testing routine simply sends one input configuration at a time, testing all input configurations and getting back outputs to check with theoretical values. We could verify this program operation by measuring voltage in pins of both the	Y

		chip being tested and the processor. This verification was successfully.	