WIRELESS KEYPAD WITH LCD DISPLAY

By

Nguyen Phan

Wei-tang Wang

Yihan Liu

Final Report for ECE 445, Senior Design, Fall 2017

TA: Xinrui Zhu

13 December 2017

Project No. 19

Abstract

This document discusses the design and implementation of a wireless programmable keypad with LCD display. The main components that are designed are the keypad matrix and the microcontroller board. The keypad communicates with computer through Bluetooth Low Energy (BLE). The LCD screen displays the key shown in the corresponding key which helps the user to know what key they type. This paper provides details for both the hardware and software design of the system and verification of its main modules.

Contents

1. Introduction

1.1 Objective

Programmable Keypad is handy for people who use hot-key heavy software like Photoshop. However, the common programmable keypad is very expensive, one can cost up to $100 or more [1], and they are not really user friendly for people who are new to the product. The software and hardware manual can be inadequate for the user to get the full grasp of keypad in a short time [2]. Therefore, the goal of this project is to create a user friendly programmable keyboard with LCD screen that can show the users the functionality they install in the key. In addition, the keyboard must be inexpensive, so it can be more accessible to a wide variety of users (under $70).

1.2 High Level Requirement

- Wireless connection to computer via Bluetooth
- Provide a number keypad for 13’ inch laptop
- Provide a gaming keypad. The keypad incorporates basic key for playing video games such as WASD. In addition, allowing user to remap the key in the keypad using add-on software

2 Design

2.1 Block diagram

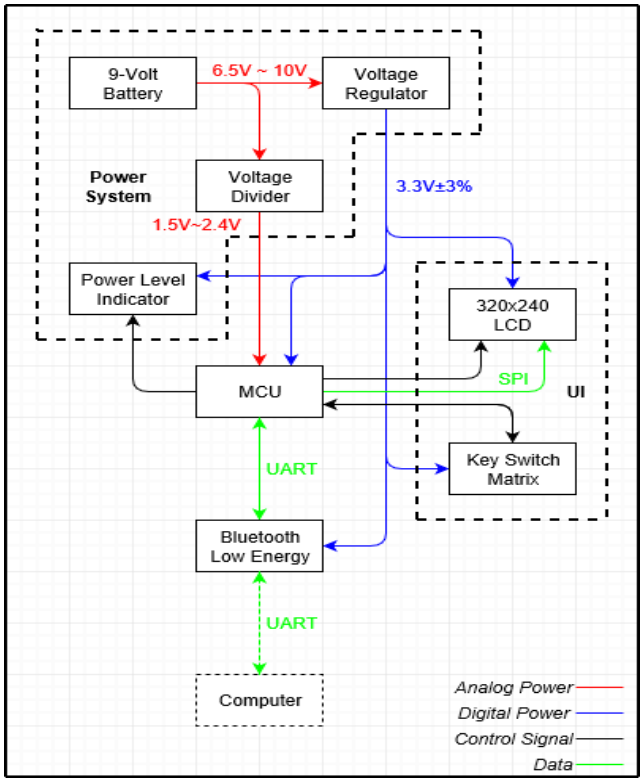

Figure 1: High Level Block diagram

The whole project can be split into several building blocks. Each of them has limited interacts with the other blocks so that they can be developed and tested separately. In case one block does not work properly, the others can still rely on some external devices which yields well-defined power and signals. Our design, in the spirit of modularity, contains the following building blocks:

Power System: A battery will be used to power the rest of the circuit. Its voltage first is regulated by a voltage regulator and then feed to microchips, pull-up circuits et cetera. A power level indicator keeps monitoring the remaining power of the battery, alerting the user when the power is insufficient.

Keypad Switches: 17 key switches, which are connected by a 4x5 grid, form a key matrix, with its layout like those of common number keypad on the market. It should not merely interpret the human behaviors like clicking, holding, or releasing a key, but handle the situation when multiple keys are pressed, as well. Furthermore, this design seeks a hardware approach to solve the bouncing problem by attaching the keys to RC filters.

Microcontroller: A MSP430F5529 is chosen to be the microcontroller due to its immense memory, both volatile and non-volatile, abundant general purpose I/O, and ability handle both digital and analog signals [3]. This

building block has little hardware design, but extensive programming is required for its interaction with its peripherals.

Bluetooth Module: It construct a bridge the device and the host, like a computer. We use a HM-10 Bluetooth Low Energy as it features minimum power consumption. The module communicates with the microcontroller using the UART terminals and can transfer data at a rate of 9600 bit/s.

LCD Display: We use a 320x240 ILI9341 as our output device of the whole design, which not merely show the functionality of the current key set but is able to highlight the active key, as well. The module's display could be set on and off by a single bit, a feature we can exploit to reduce the power consumption. [4]

2.2 Hardware Design

2.2.1 Power System

2.2.1.1 Voltage regulator

Input: 9V power from the battery

Output: 3.3V±5% power for the Bluetooth module, the microcontroller, pull-up resistors, and the LCD screen

The circuit consists of a LM317T adjustable voltage regulator, whose functionality is to step down the varying input voltage and generate stable outputs for the rest of the device. The regulator features a wide range of input voltage and a maximum output current of 1.5A, both of which exceed our requirement significantly. Aside from the two resistors (**R1, R2**) necessary for adjusting the output voltage, extra components are added to protect the device and improve the performance. The decoupling capacitor $\mathbf{C_1}$ filters out unwanted AC noise which is induced for various reason, while $\mathbf{C_2}$ improves ripple rejection of about 15dB. A third capacitor $\mathbf{C_3}$ is connected between output and ground to improve transient response. The diode $\mathbf{D_1}$ protects the device against potential input short circuit, and $\mathbf{D_2}$ is against output short circuit for discharging the capacitors.

The output voltage of LM317T is given by its datasheet as

$$V_{out} = V_{ref}(1 + \frac{R_2}{R_1}) + I_{adj}R_2.$$

(2.4.1.1)

V_{ref} is the voltage between the output and adjustment terminals and is fixed at $1.25V$, while I_{adj} is designed to be minimized and to maintain constant with line and load changes? Because this current is relatively small ($100\mu A$ max), the error term $I_{adj}R_2$ can be neglected for simplifying calculation. Furthermore, the datasheet also suggests a resistance of 240Ω for R_1. As a result, to acquire the desired voltage, the value of R_2 can be determined by the equation

$$V_{out} = 1.25(1 + \frac{R_2}{240}).$$

(2.4.1.2)

Rearranging the equation and then plugging V_{out} in, we get

$$R_2 = 240(\frac{3.3}{1.25} - 1) = 393.6\Omega$$

(2.4.1.3)

for output voltage of $3.3V$

By Ohm's law we have

$$I = \frac{V}{R}$$

The current flowing through R1 and R2, connected in series, is 5.238mA, which is not a trivial value. A plausible solution to reduce the current is to increase resistance of the two resistors by ratio. However, it is unfortunate that the magnitude of the error term would also boost, make itself unneglectable if R2 was increased by, say, an order, increasing the uncertainty of an arbitrary LM317 chip.

The external capacitors are for bypassing purpose and improve the performance of the circuit, and the diodes are necessary to prevent circuit from damage caused by the discharging capacitors. The values of those components are given by the datasheet. Since the circuit performs well when it is tested on the breadboard, no further adjustment is necessary.

2.2.1.2 Voltage Divider

The voltage divider consists of two resistors in series, giving an analog signal proportional to the voltage of battery, which itself is too high for the ADC port of the microcontroller (Comp_B).

The ratio of output to input voltage satisfies the straightforward voltage division law

$$\frac{V_{out}}{V_{in}} = \frac{R_{down}}{R_{up} + R_{down}}$$

(2.4.2.1)

The ratio between the two resistors is therefore

$$\frac{R_{up}}{R_{down}} = \frac{V_{in} - V_{out}}{V_{out}}$$

(2.4.2.2)

Here V_{in} is the critical voltage level of $7V$, under which the battery drains out rapidly, and V_{out} is half of

the V_{cc} of the microcontroller, which is close to $1.65V$. Plugging the two values into the equation yields

$$\frac{R_{up}}{R_{down}} = \frac{7 - 1.65}{1.65} = 3.242$$

(2.4.2.3)

Let R_{up} and R_{down} be $36k\Omega$ and 11 $k\Omega$, respectively. The maximum current flowing through is therefore

$$I = \frac{V}{R_{tot}} = \frac{9.0}{(36 + 11) \times 10^3} = 191.5\mu A$$

(2.4.2.4)

This is an acceptable value in terms of power consumption, and should be a few orders greater than any potential noise.

2.2.1.3 Power Level Indicator

Each of the green and red LEDs are connected to a pull-up/down resistor, respectively. Because one of them is active-low and the other is active-high, a single logic signal is enough to control both LEDs, provided that one and only one is turned on in an arbitrary moment.

The magnitude of the resistors determines the current through the LEDs, and therefore their lightness. With the Kirchoff's voltage law applied two equations are derived:

$$V_{cc} - R_{pull-up}I_f - V_f - V_{Lmax} = 0$$

(2.4.3.1)

$$V_{Hmn} - R_{pull-down}I_f - V_f = 0$$

where V_{Lmax} and V_{Hmin} is the maximum logic low (0.25V) and minimum logic high (3.05V) the microcontroller might produce, and the V_{foward} is the voltage drop, which is slightly higher than 2V, when the LED is activated. In order to induce a current of 5mA, the value of resistor should be selected around 150Ω.

2.2.2 Keypad matrix

Input: User press a keystroke, 3.3V±5% power supply

Output: Signal to the microcontroller

We intend to build a 5x4 keypad, with traditional functions such as registering numerical values with the receiver. On top of it, we will add new functions such as play/pause music/video, or even customizable by the user.

The keypad matrix is controlled by the microcontroller. For a 17-button 5x4 matrix with 5 rows and 4 columns, 9 pins of the microcontroller will be connected to the matrix. The advantage of such connection, compared with the 1-to-1 mapping, is that significant number of pins are saved, as only nine (four outputs and five inputs) GPIO of MCU are needed.

To implement our design, we will be using button switches as our keys, because they are the most cost-efficient option for us. In addition, we also add a Schottky diode to each key switch to prevent the current from flowing in reverse direction. Thus, it also provides a solution to the ghosting/masking problem. The Schottky diodes also feature a much lower forward drop voltage compared with the common ones at a given forward current, and generate a lower logic low output.

2.3 Software design

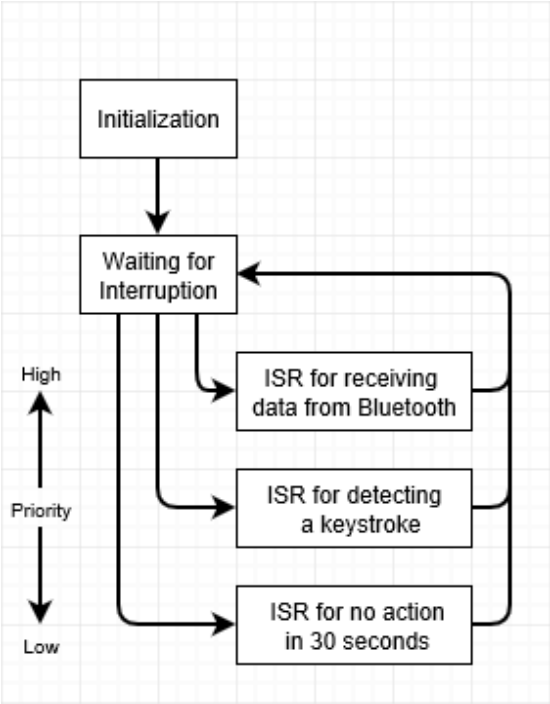

Figure 2: Flow chart for ISR (interrupt service routine)

By default, the ISR is waiting for interruption. This can be triggered when the users remap a key which send data from the computer to the microcontroller via Bluetooth module. Another occasion is when the user presses a key. In this case, ISR is used for detecting the pressed keystroke. The last case when ISR is initiated is when there is no action for 30 seconds, ISR is used to trigger the LCD screen to turn off. Receiving Bluetooth data has the highest priority when initiating ISR, next is the detecting keystroke and idle keypad (no action) has the lowest priority.

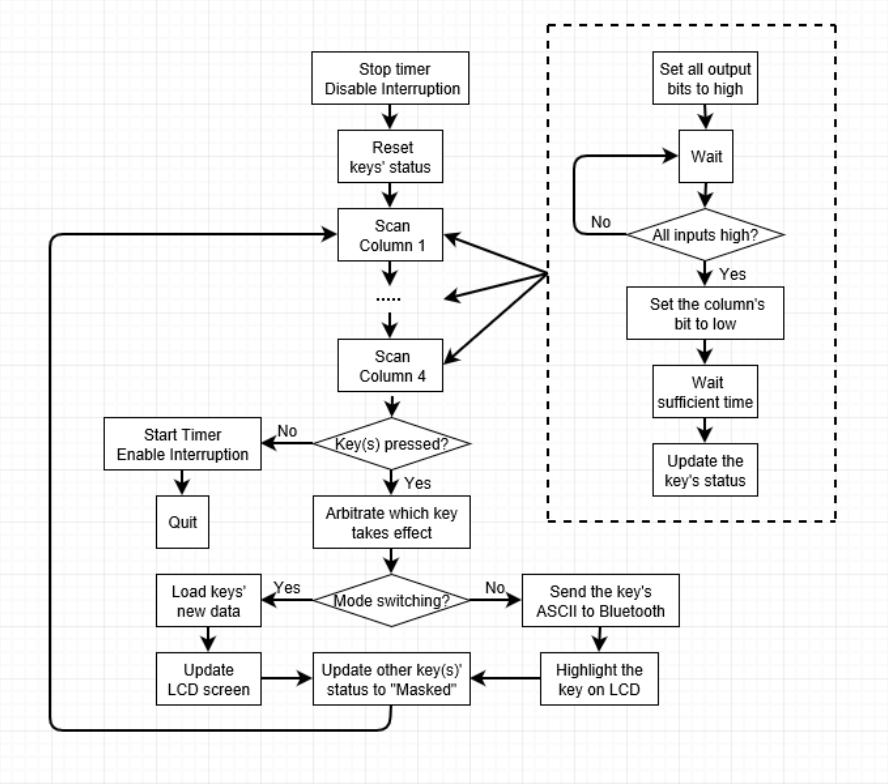

Figure 3: Key switch detection flow chart

The default mode is numpad mode. For our project, we intend to use the row-interrupt approach. 4 pins will be INPUTS of the microcontroller and connected to the COLUMN wires, while the other 5 pins will be OUTPUTS of the microcontroller (MCU) connected to the ROW wires. When the key is not pressed, the INPUT is low, so the microcontroller does not take any action. When a key is pressed, the OUTPUT of the microcontroller become HIGH, and when the signal come to the INPUT, and turn it to HIGH. The microcontroller scan through the INPUT pins and detect the specific OUTPUT that is HIGH. The MCU then detect whether mode switching key is pressed or not. If the mode switch is not press, then the MCU send the pressed key ASCII value to the Bluetooth and send signal to highlight the key on LCD. If the mode switching key is pressed, the MCU load the second mode key pattern and update the LCD screen. The MCU will then update other key status to ‘Masked’, and then go back to the scanning column again.

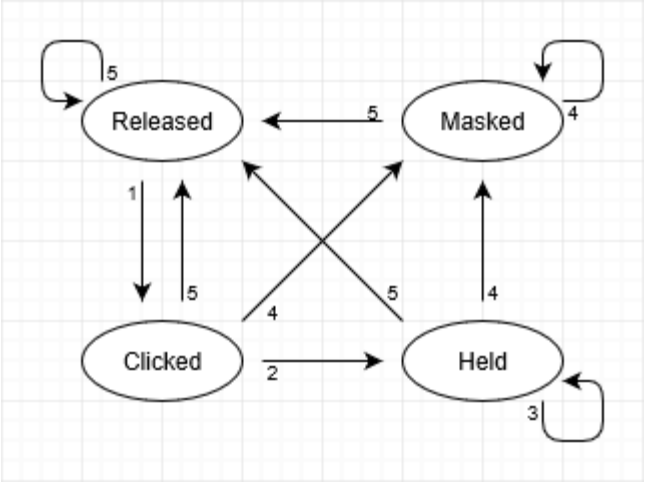

Figure 4 State diagram of the keypad

Nothing will happen if the user does not press any key. If the user releases a click/held/masked key, the key will go to released state. If multiple keys are clicked/held, the old keys (can be clicked or held keys) will be masked out. When users holds the clicked key, the keystroke moves to held state, and remains so if the user keep holding it.

3. Design Verification

3.1 Power system

3.1.1 Output voltage of the voltage regulator

To verify the voltage regulator output, we connected the input and the output of the circuit to a DC voltage generator and a multimeter, respectively. We then swept the output of the power supply between 5V and 9.5V, and read the measurement shown on the multimeter. The result is show in the graph below.

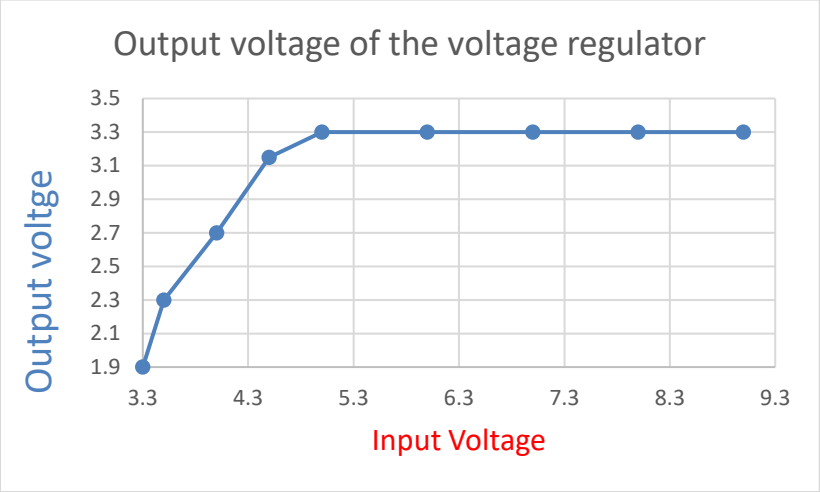

Figure 5: The measured output voltage for 3.3V Regulator (R_2 = 393.6 Ω)

From **Figure 5**, the voltage regulator consistently produces 3.3V when the input voltage is greater than 5V

3.1.2 Energy Consumption

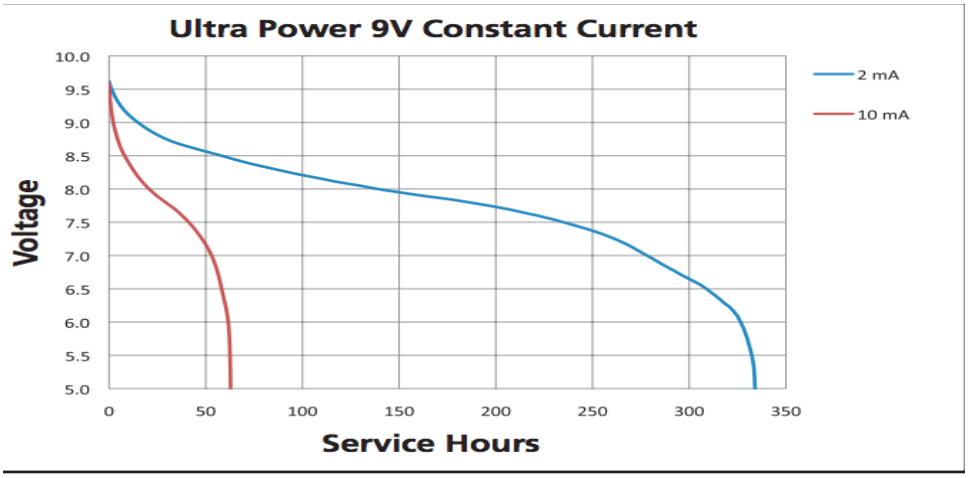

Figure 6: Output Voltage over Time for a standard Duracell Ultra Power 9V Battery [5]

Table 1: Total power consumption of the keypad

Component	Voltage	Current	Unit Number	Power
Microcontroller	3.3V	0.29 mA	1	0.957 mW
LCD screen	3.3V	90.5 mA*	1	298.65 mW
Key switch	3.3V	0.01 mA	16	0.48mW
Bluetooth module	3.3V	14 mA	1	46.2mW
Total		104.8 mA		346.281 mW

(*: this is the maximum supply current value for the LCD screen)

The estimated milliamp-hours capacity of a 9V battery is about 400mAh, and if the current discharge of our keypad is 104.8 mA. The estimate hours of operation of the keypad is

$$Hours\ of\ operation = \frac{\text{Miliamp} - \text{hours capacity}}{\text{Discharged current}} = \frac{400\ mAh}{104.8\ mA} = 3.82\ hours$$

The result indicates that the keypad will last for about 3.82 hours. This assume that the LCD screen supply current is on all the time. Often time, the LCD screen will be off when the keypad is idle. When the screen is off the total current is about 1/8 of the current shown in **Table1,** so the keypad can operate for longer time than the calculated value.

3.2 Keypad matrix

3.2.1 Bounce time

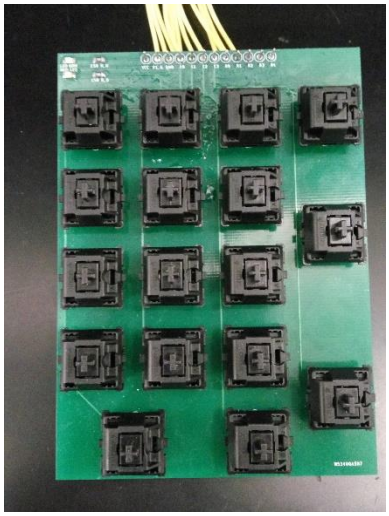

Figure 7: Actual implementation of the keypad

There is one issue we must deal with when working with mechanical key switch, that is the bounce time. When a switch is toggled, contacts move changes their positions. As the components of the switch settle into their new positions, they mechanically bounce, causing the circuit to be opened and closed several times.

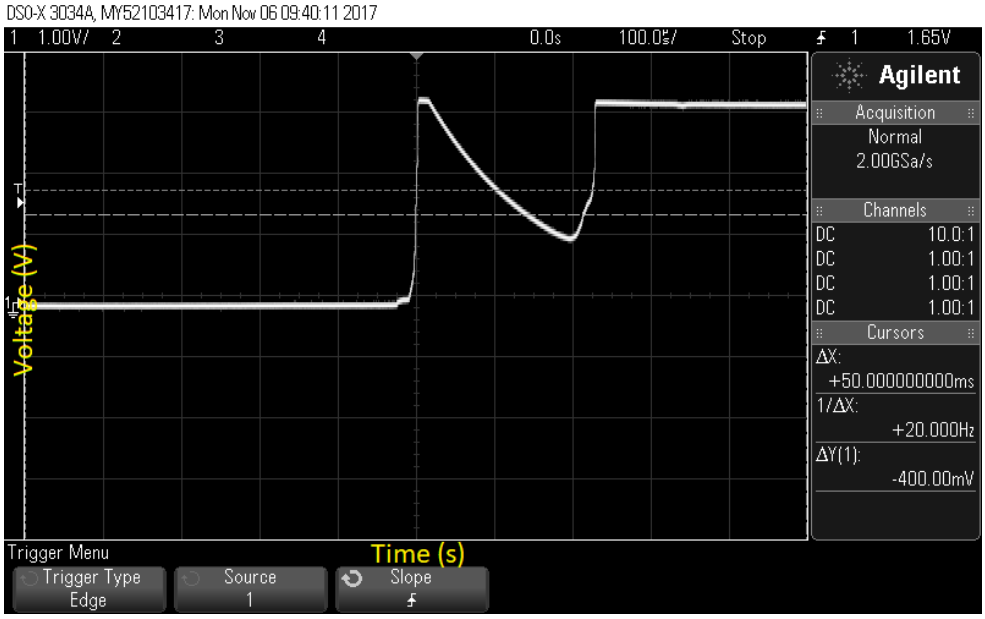

Figure 8: The transition curve of the key switch without using filter

As can be seen from **Figure 8**, voltage is jumping between the middle point when the key is pressed or released. This causes unstable performance of the keypad.

To solve this problem, we added filters to the keypad, which consists of 12nF capacitors. This greatly improve the performance of the keypad since it slows the bounce time down.

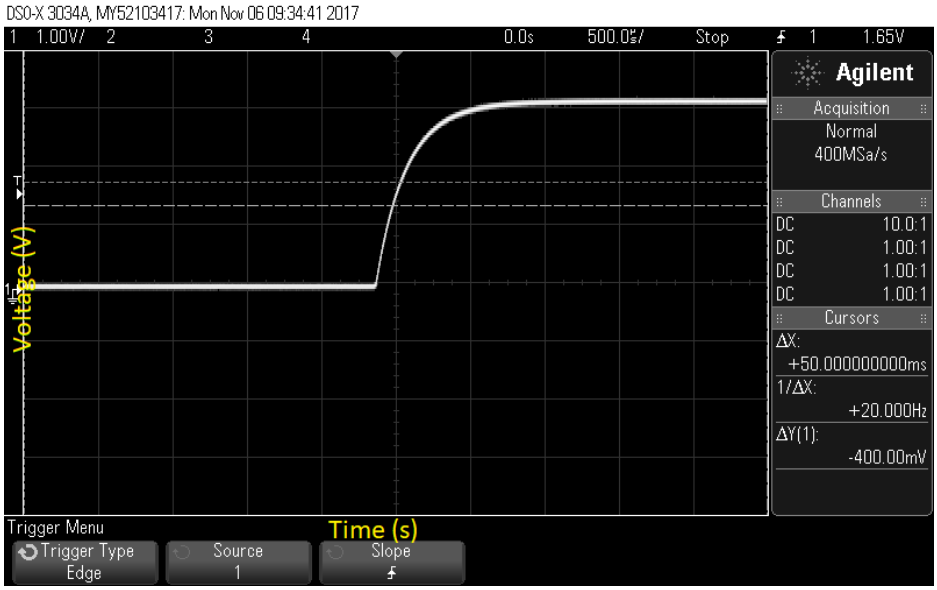

Figure 9: The transition curve of the key switch using filter

From **Figure 9**, we can see that the curve is smoother which ensure stable performance of the keypad.

3.3 LCD screen

3.3.1 Display different modes

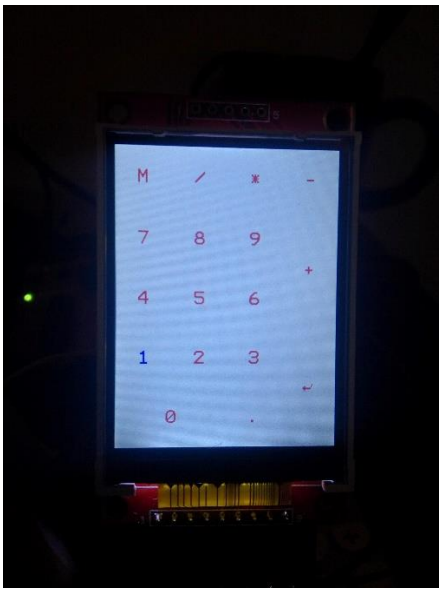

Figure 10: The numpad display

We tested the LCD screen when the system was running. We sent the pattern from the microcontroller to the LCD and checked whether we could get the desire pattern. From the figure, the LCD can display all of the character of the regular numpad. This is also the default mode of the keypad. The only different is that we replaced the **NumLock** on the top left corner with **M**. This button allows the user to switch to a different mode. When the user press a button, the corresponding character will be highlighted by blue. In **Figure 10**, the user pressed **1,** so **1** highlighted in blue.

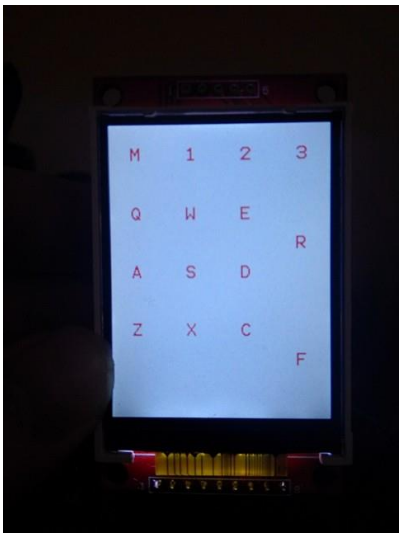

Figure 11: The gaming keypad display

For the gaming keypad, the LCD screen display all the common keys for most video game such as WASD. These keys position is like their position in regular keyboard. When using this mode, users also expect similar performance as numpad mode.

Figure 12: The remapping mode display

For remapping mode, we left the screen mostly blank except for **M** key on top left corner. When the user remaps a key in the keypad with new character, the LCD will display that character on the screen with correct location. The location of each keystroke in the LCD is shown in the figure below.

	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P
Q	R	S	T

Figure 13: Location of each keystroke in both LCD screen and keypad (the top left corner is **M**)

In this example in **Figure 12**, the user mapped character **k** to the key next to **M.** To do so, the user typed **Bk** in the computer. The first letter indicates the location of the keystroke in both the LCD and keypad, the second letter is the character the user wants to map to that keystroke.

3.4 Bluetooth module

3.4.1 Transmission

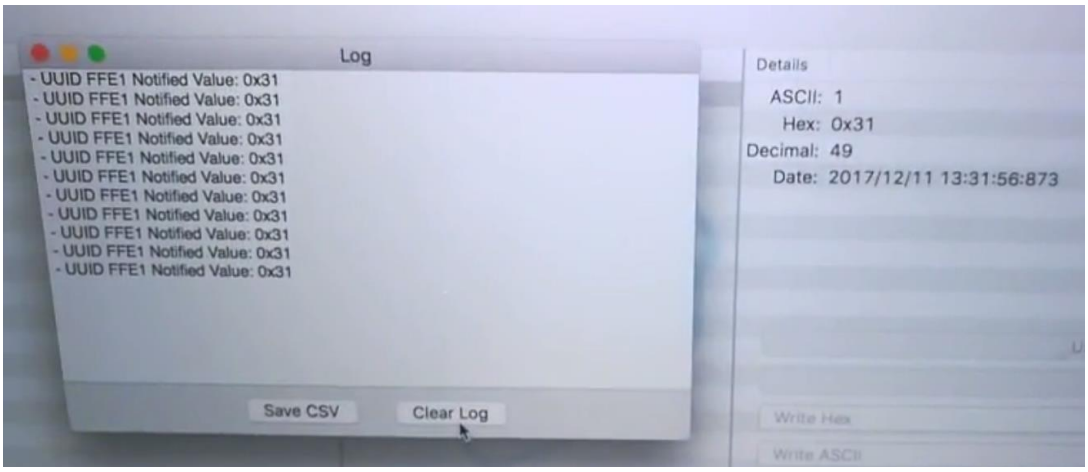

Figure 14: Data transmit to the computer when user press a key

We tested the Bluetooth transmission when the system was running. When the user pressed a key on the keypad, the corresponding character of that key will be show on the screen. From **Figure 14**, we can see the computer shows ASCII and Hex value of the entered character.

3.4.2 Receiving

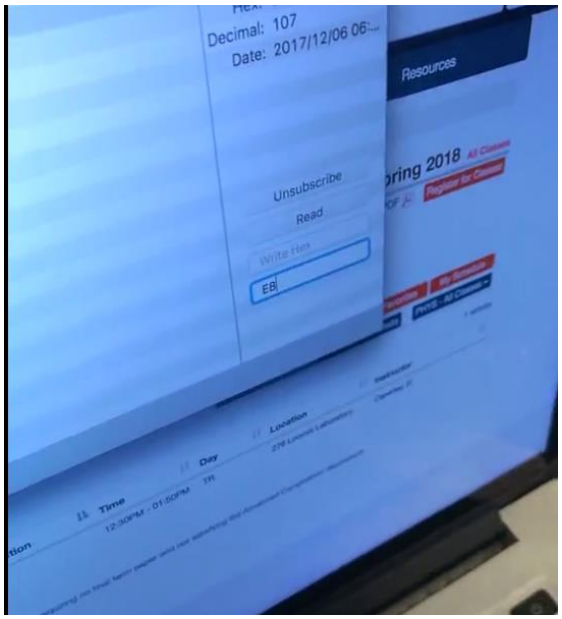

Figure 15: Data transmit from the computer when user remap a key

As can be seen from **Figure 15,** when the user wants to remap a key they simply enter two characters. The first character for the location of the remapped key, and the second character is the ASCII value of the remapped key. In this example, the user enters E8 which means the user want to map number **8** to the key right below **M**

Figure 15: The remapped key show in the LCD screen

4. Costs

4.1 Parts

Table 2: Components Costs

Part Name	Part Number	Unit Cost	Quantity	Total
Microcontroller	TI MSP430F5529IPNR	$7.50	1	$7.50
Bluetooth module	HM-10	$7.78	1	$7.78
Battery		$1.88	1	$1.88
Voltage Regulator	LM317EMP/NOPB	$1.28	1	$1.28
2.2” 320x240 Pixel LCD	IL9431	$10.99	1	$10.99
Key switch	MX1A-11NW	$0.79	20	$15.80
Schottky Diode	SD08058020S1 RO	$0.42	19	$7.98
PCB		$36.00	1	$36.00
Passive Components		$5.6		$5.6
Total				$94.81

4.2 Labor

Table 3: Labor Costs

Name	Hours Invested	Hourly Rate	Total Cost = Rate * Hours*2.5
Nguyen Phan	200	$25	$12500
Yihan Liu	200	$25	$12500
Wei-Tang Wang	200	$25	$12500
Total	600		$37500

4.3 Grand Total

Table 4: Total Costs (Component Costs + Labor Costs)

Parts	Labor	Grand Total
$94.81	$37500	$37594.81

5. Conclusion

5.1 Accomplishment

Our team successfully developed a working keypad that could communicate with computer via Bluetooth. The keypad could send correct ASCII character to the computer when the user presses the corresponding keystroke. The user was able to remap desired key using the computer. The ASCII value is assigned to correct keystroke. The LCD displayed correct pattern of the keypad in both numpad and gaming mode. It also showed correct character and location in remapping mode.

5.2 Uncertainties

One issue we had was our voltage regulator. We could produce the correct 3.3V output voltage when testing the voltage regulator individually. However, when integrating the voltage regulator into the circuit, it did not work properly which damaged our microcontroller. We suspected that the voltage regulator we chose is was cheap linear regulator, so it did not have stable performance.

5.3 Ethical considerations

During our design and research process, we keep sticking to the codes of ethics promulgated by the IEEE community. Particularly, we are honest in using the data collected during the experiments and tests, not trying to misinterpret some phenomenon even if the data are against them. We are also eager to accept advice and criticism from our instructors and TAs, as they can point out the improper or unrealistic aspects in our works, helping us return to the right track. Moreover, we realize that there are techniques and solutions, some of them even patented, related to our design, and it is our responsibility to credit them properly can avoid any potential plagiarism.

5.4 Future work

Other than fixing the aforementioned problem, we are considering to optimize our device in the following ways: First we are going to replace our Bluetooth module with the one that support the HID mode, which is for the computer-keyboard communication. Although our current module is able to transmit and receive ASCII code properly, the host does not recognize it as a keyboard device. We will also attach an oscillator, whose frequency matches the baud rate of the new Bluetooth module, to the microcontroller in order to reduce the error rate to minimum. Finally, a software for reassignment of keys should be developed. Currently a user is able to send arbitrary strings to the device, potentially causing security problems. A well-built software should regulate the behavior of the user and encapsulate the detailed implementation.

References

[1]Komar's techblog, "How to make a keyboard-The matrix", 2014. [Online]. Available: blog.komar.be/how-to-make-a-keyboard-the-matrix/. Accessed 3 Oct. 2017.

[2] IEEE Xplore, "Design of Multi-function and Programmable Man-Machine Interface",2014.[Online]. Available: http://ieeexplore.ieee.org/document/6821583/. Accessed 4 Oct. 2017.

[3]Chron.com, "Advantages & Disadvantages of Using the QWERTY Keyboard", 2015.[Online]. Available: http://smallbusiness.chron.com/advantages-disadvantages-using-qwerty-keyboard-66874.html Accessed 4 Oct. 2017

[4]Torpal.com, "From the Ground Up: How I Built the Developer's Dream Keyboard", 2015[Online]. Available: https://www.toptal.com/embedded/from-the-ground-up-how-i-built-the-developers-dream-keybooard. Accessed 4 Oct. 2017

[5] Batterysales, "Duracell MX1604 Alkaline-Manganese Dioxide Battery", 2017. [Online]. Available: http://www.batterysales.nl/files/downloads/Battery_Sales_Europe_Ultra_Power_9V_MX1604.pdf . Accessed 4 Oct. 2017.

Appendix A Requirement and Verification Table

Table 1: The requirement and verification for Power System

Requirement	Verification	Verification status(Y/N)
Voltage Regulator		
The voltage across the output terminal and the ground stays within the range of 3.3±0.2V, provided that the input voltage is from 5V to 9.5V.	1a. Connect the input and the output of the circuit to a DC voltage generator and a multimeter, respectively. 1b. Sweep the output of the power supply between 5V and 9.5V, and read the measurement shown on the multimeter	Y
Battery		
3. 9V alkaline battery can power up the whole circuit	3a. Attach the battery to the circuit and see if the system works properly.	Y
Voltage Level Indicator		
4. The green and red LEDs are turned on and off respectively with a logic low signal, and the opposite vice versa.	4a. Connect the corresponding terminals to Vcc, the ground, and a DC voltage generator. 4b. Sweep the output of power supply within the low-level output voltage of MSP430 (0-0.25V) and check if the green LED is on and the red is off. 4c. Redo the step b. for the red LED	Y

Table 2: The requirement and verification for Keypad matrix

Requirement	Verification	Verification status(Y/N)
Keypad matrix		
1. The keypad should be free of ghosting and masking problem.	1a. Build a 2x2 mini key pad using the same schematic. 1b. Name the column A and B, and row 1 and 2. 1c. Connect the corresponding output of, say, A1 to an active-low LED. 1d. Connect the column of A1, which is A in this case, to logic low, and the other column, B, to logic high. 1e. Hold the other three keys, A2, B1 and B2, and verify that the LED remains off. 1f. While keep holding the three keys, press and then hold the fourth key, A1, and verify that the LED set on. 1g. While keep holding the three keys, release the fourth key, A1, and verify that the LED set off.	Y
2. The bounce time of any key is less than 5ms.	2a. Connect the power supply to the Vcc and tune its output voltage to 3.3V. 2b. Connect HDO4054-MS Oscilloscope to the output of the circuit. 2c. Click a key and measure the output voltage against time.	Y
3. There is a distinguish between pressing a key and holding a key	3a. Link the keypad with a Mac via Bluetooth 3b. Press a key and check the amount of time the ASCII character appear in the Mac log 3c. Hold a key and check the amount of time the ASCII character appear in the Mac log	Y

4.The software can find out which key or keys are pressed. It should also respond differently based on the key or keys being pressed.	4a. Write a debug function: If the value of the variable curr_key in the program matches a certain key, the LED on the launchpad is then triggered. 4b. Replace the sending_key_index function with the debug function so that instead of telling the system which key is generate an effective signal, the LED is flipped only if that certain key is taking the charge. 4c. Download the test program into the launchpad. 4d. Power the launchpad and connect the input and output terminals of the Keypad Matrix and the Pull-up circuit when the soldering work is finished. 4e. Test the behavior of that certain key.	Y

Table 3: The requirement and verification for Microcontroller

Requirement	**Verification**	**Verification status(Y/N)**
Microcontroller		
1. The microcontroller can send correct bit pattern to the LCD screen	1a. Use a predefined bit pattern and send it to the microcontroller 1b. Determine whether the LCD can display the correct bit pattern	Y
2. The microcontroller can send data to the Bluetooth module	2a. Use a predefined bit pattern and send it to the Bluetooth via microcontroller 2b. Mac receives the desired data and show it on the screen	Y

Table 4: The requirement and verification for Bluetooth Module

Requirement	**Verification**	**Verification status(Y/N)**
Bluetooth Module		
1. The Bluetooth module is able receive data from the host device	1a. Send a predefined bit pattern from the computer to the keypad via Bluetooth 1b. The bit pattern is displayed on the LCD screen	Y
2. The Bluetooth module can transmit data	2. The ASCII characters sent from the keypad when a key is pressed can be showed in the host computer	Y

Table 5: The requirement and verification for LCD screen

Requirement	**Verification**	**Verification status(Y/N)**
LCD Screen		
1. The pattern shown on the LCD screen can change based on the current mode.	1a. Press the mode-switching key. 1b. The pattern on the LCD display changes to the corresponding mode with each key position reflected on the screen.	Y
2.The LCD can display the key pattern when that key is pressed	2a. Press any key except the mode-switching key. 2b. The pattern changes to a different color when the corresponded key is pressed	Y

Appendix B Circuit Schematic

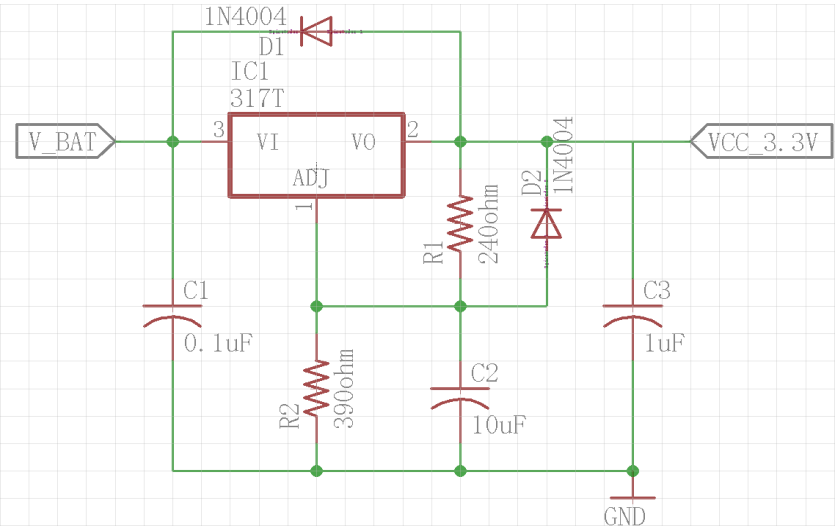

Figure 1: Schematic for 3.3V regulator

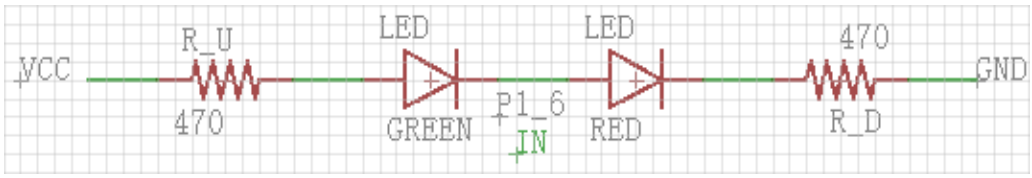

Figure 2: Schematic for the power level indicator

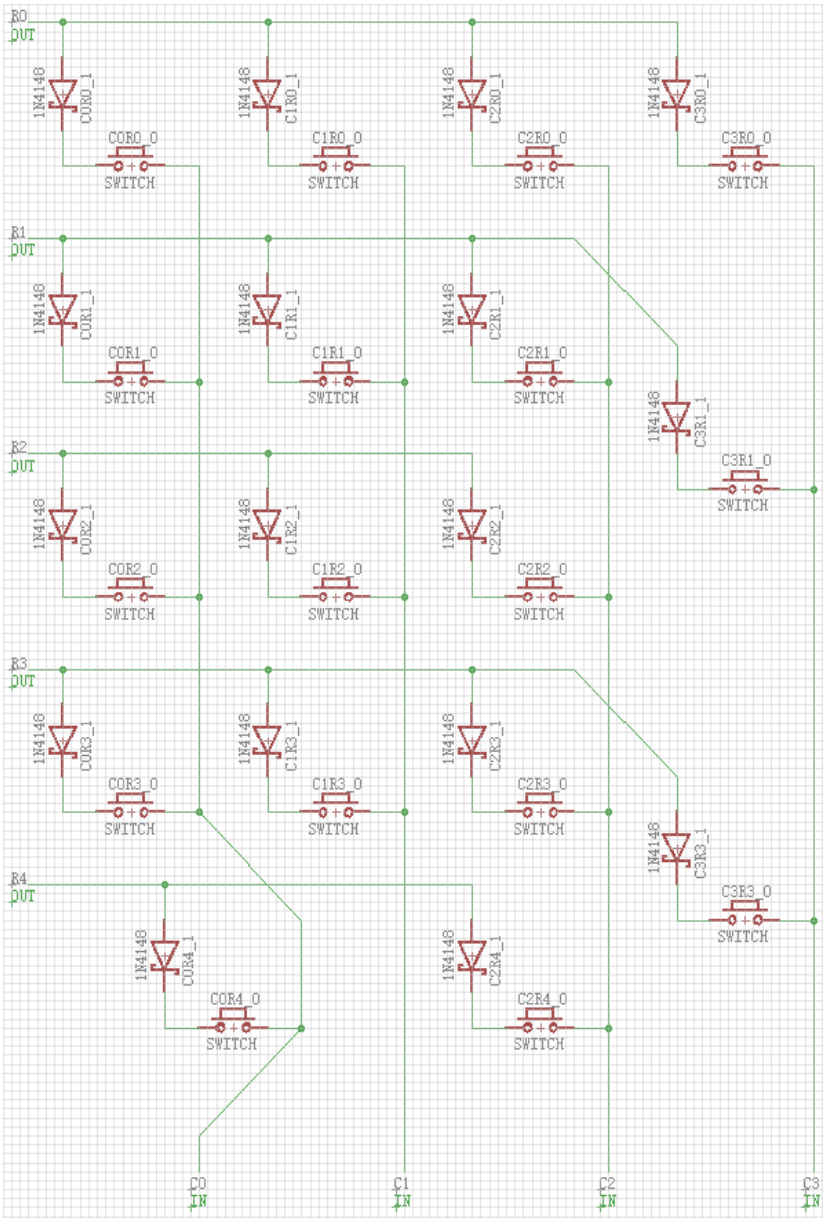

Figure 3: Schematic for the keypad matrix

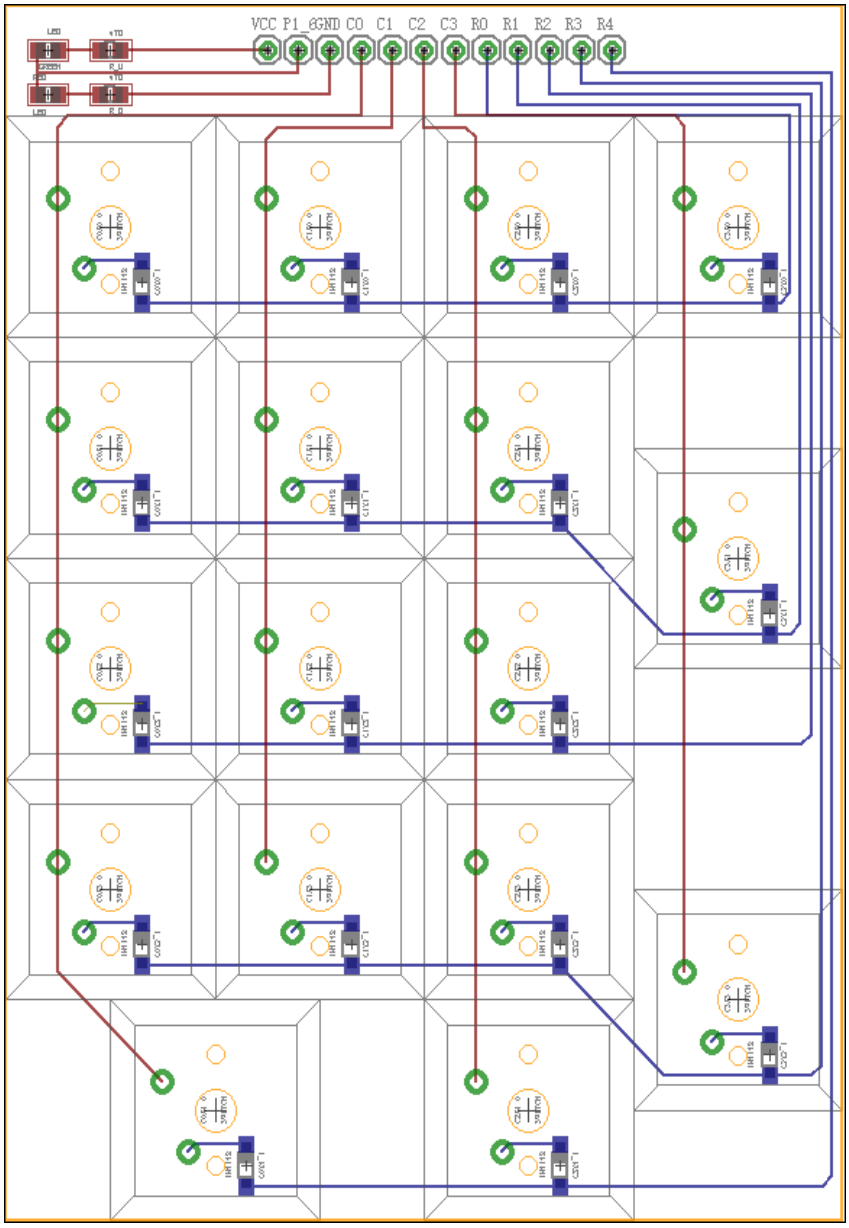

Figure 4: The PCB layout of Keypad Matrix and Power Level Indicator

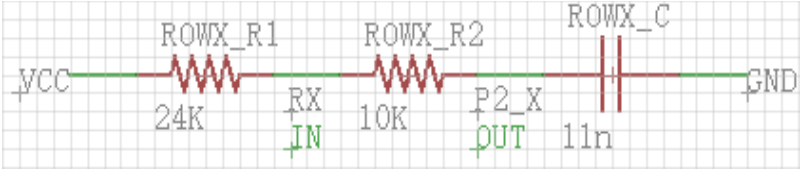

Figure 5: Schematic for the pull-up circuit with RC filter

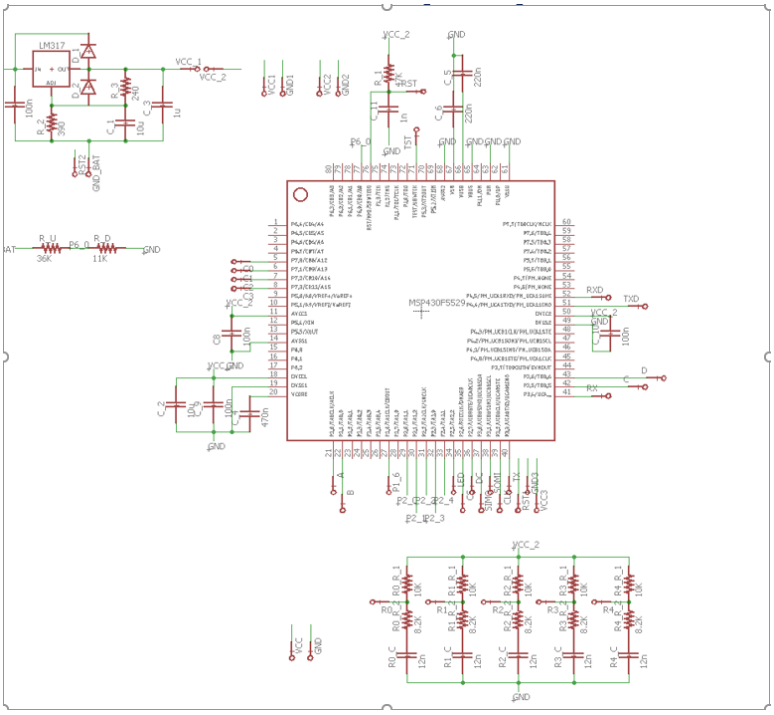

Figure 6: Schematic for the microcontroller board

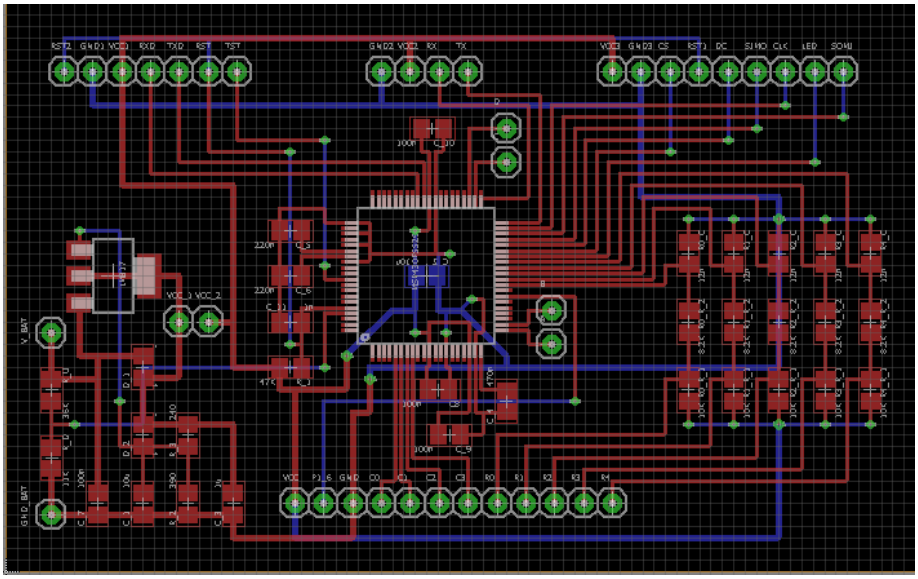

Figure 7: PCB layout the microcontroller board

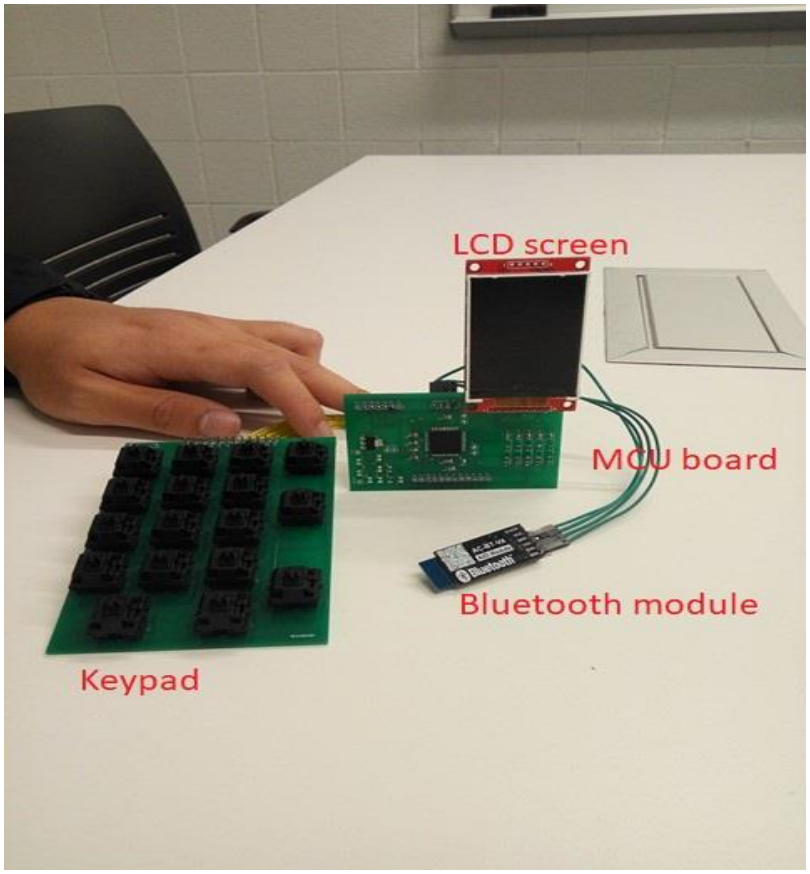

Figure 8: The actual implementation of the whole system