Wireless programmable keypad with LCD display

ECE 445 Design Review

Team 19:

Nguyen Phan

Yihan Liu

Wei-Tang Wang

TA: Xinrui Zhu

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 Background

QWERTY keyboard is considered the world standard. However, one major problem with QWERTY keyboard in this day and age is that its arrangement is inflexible as it can only support the antiquated key configuration [3]. For people who use software that require a lot of hot keys like Photoshop or playing video games, so using QWERTY keyboard can be a bit cumbersome. This is where programmable keypads are handy because they offer shortcuts for any area where keyboard shortcut is useful. However, programmable keypad tends to be expensive. For instance, an XK-24 USB Programmable Keypad, which has 24 keys, is about $145[4]. This is the reason why programmable keyboard are not accessible to a wide variety of customer. Aside from that, using these keyboard require some understanding about keyboard and how they works[5], so they are definitely not easy to use for people who are new to programmable keypad.

1.3 High Level Requirement

- Wireless connection to computer via Bluetooth
- Provide a number keypad for 13’ inch laptop
- Provide a media controller. The keystrokes allow users to play or pause the current music or video on the host computer, change the brightness of the screen.

In addition, allowing user to set up additional functionality for the keypad using add-on software

2. Design

2.1 Block diagram

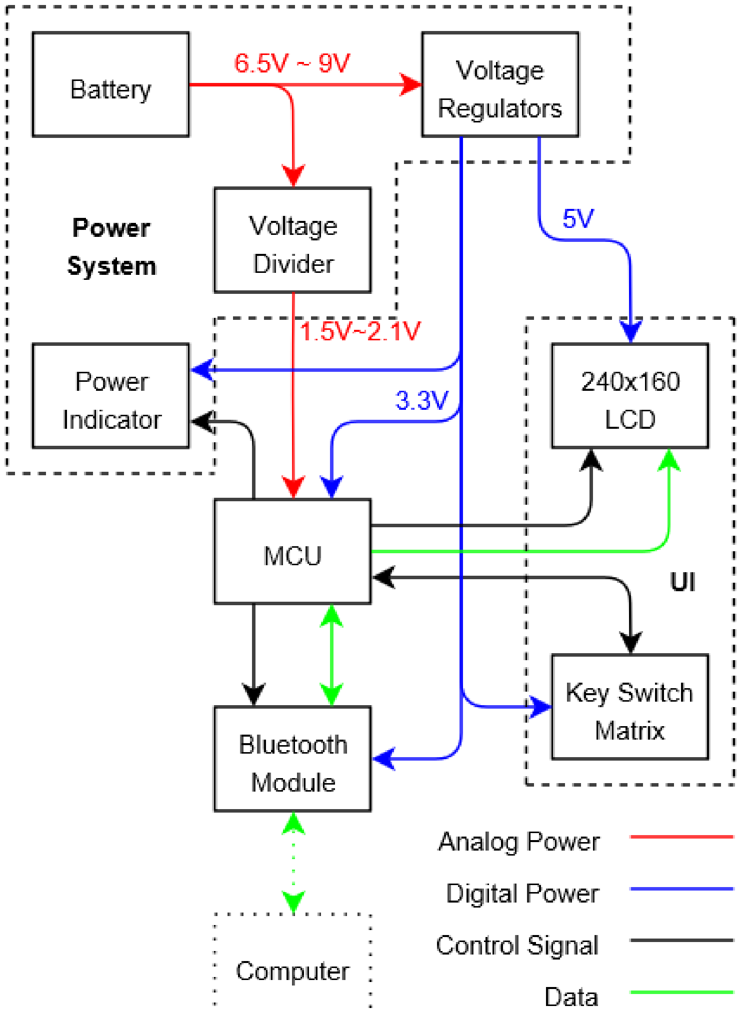

Figure 1: High Level Block diagram

2.2 Physical Design

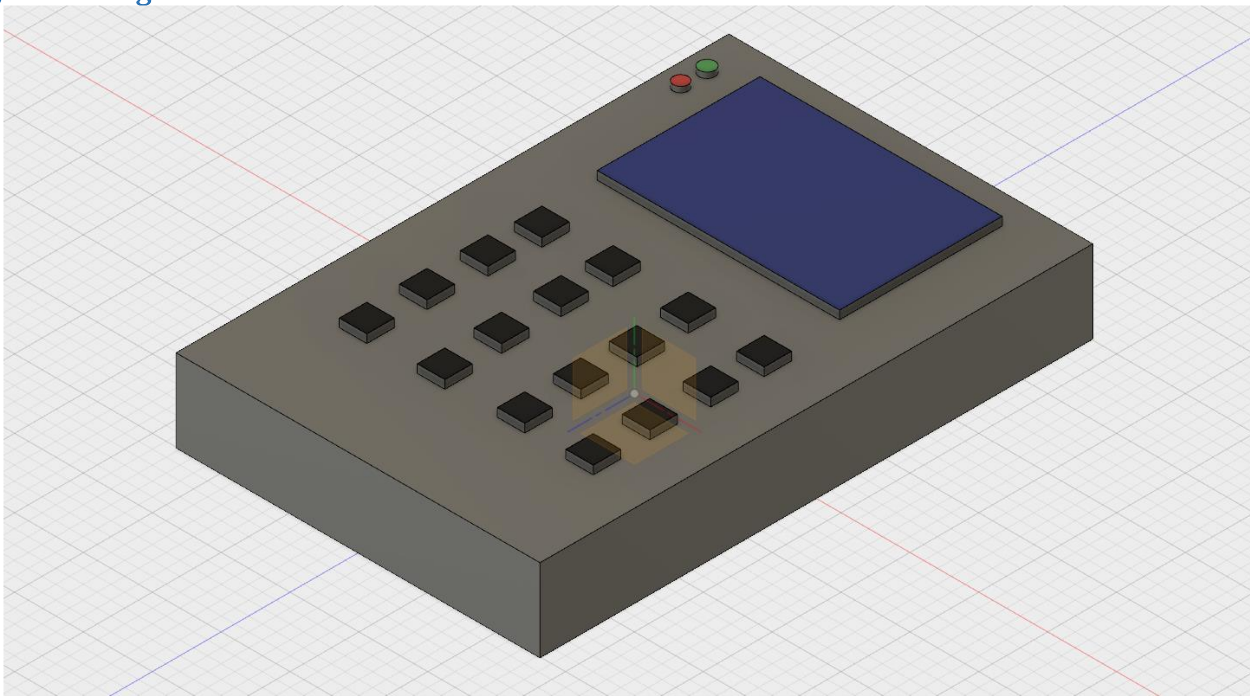

Figure 2: Physical design of the keyboard

2.3 Hardware Design

2.3.1 Power System

A consistent power supply is necessary in order to power all of the components in our circuit. We choose to use a standard 9V battery, which benefits the user in the sense that they are easily replaceable and are not hard to find. In order to maintain a stable power supply, we will also build 2 voltage regulator/divider circuits to satisfy the input voltage requirements of our circuit components.

2.3.1.1 Battery

A 9-volt alkaline battery is select to power the whole device, as it benefits the user in the sense that they are easily replaceable and are not hard to find. The battery is capable of providing a regulated 5V to the LCD screen, much less the 3.3V required for powering microcontroller, BLE module, and miscellaneous LEDs.

2.3.1.2 Voltage regulator

Input: 9V power from the battery

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

The circuit consists of two 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 **C$_1$** filters out unwanted AC noise which is induced for various reason, while **C$_2$** improves ripple rejection of about 15Db. A third capacitor **C$_3$** is connected between output and ground to improve transient response. The diode **D$_1$** protects the device against potential input short circuit, and **D$_2$** is against output short circuit for discharging the capacitors.

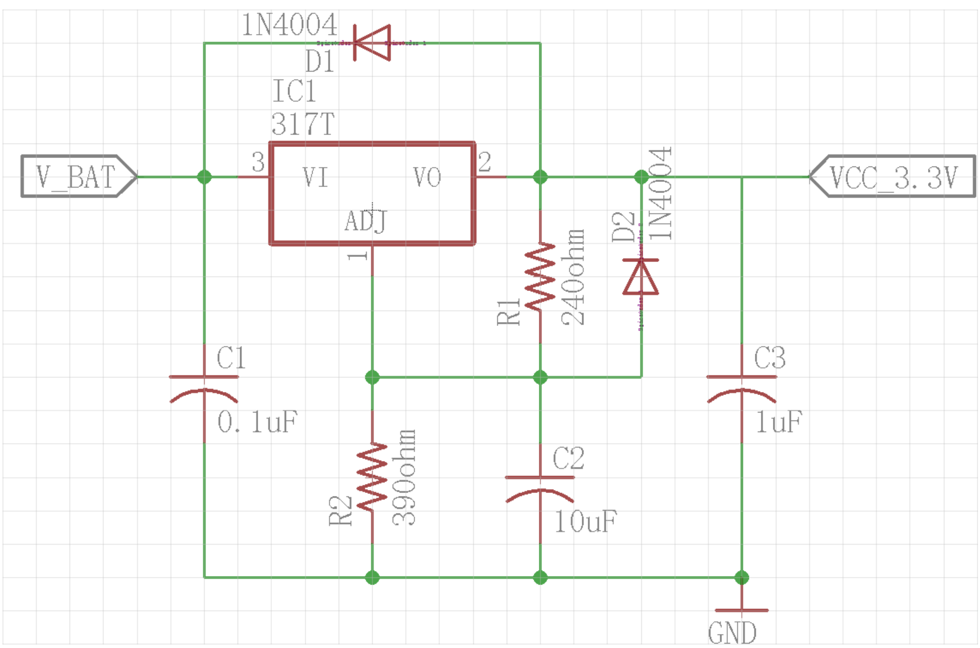

Figure 3: The schematic for 3.3V regulator

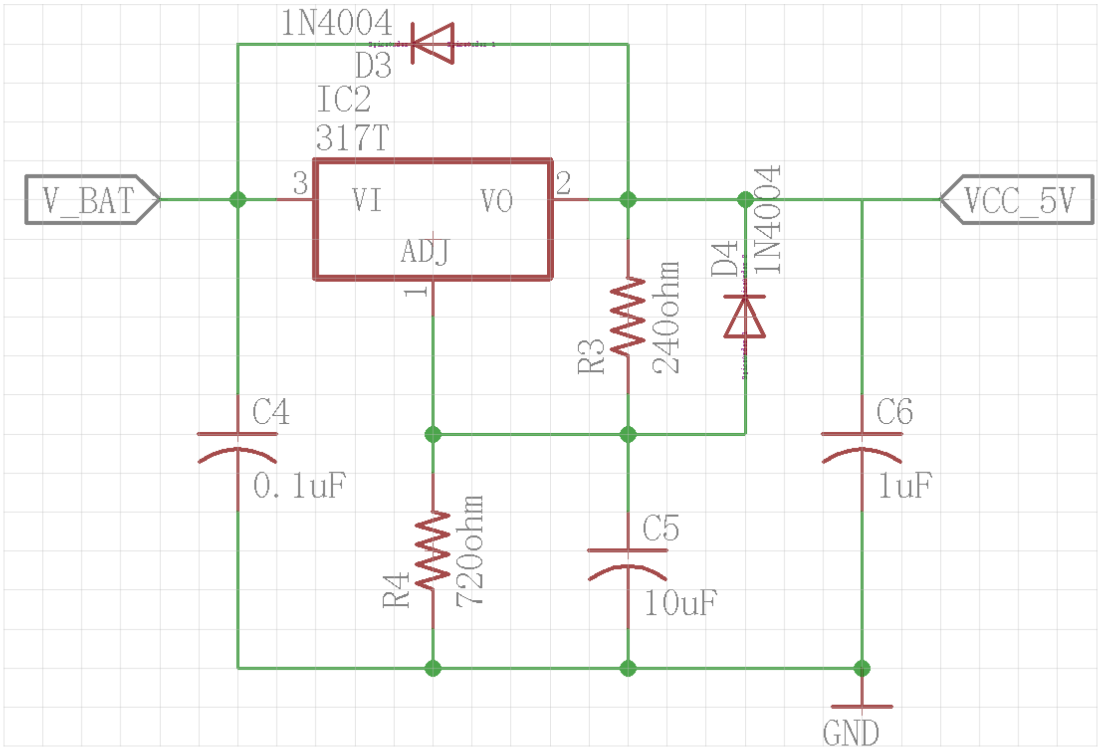

Figure 4: The schematic for 5V voltage regulator

2.3.1.3 Voltage Divider

This simple resistive voltage divider is essentially two resistors connected in series, producing the analog output fractional to the battery voltage, which is otherwise too high for the ADC port of the microcontroller.

2.3.1.4 Power Level Indicator

One green LED and one red LED are used to indicate the power level of the battery. Only the green LED will be lighted if the power is turned on and battery voltage is higher than 7 volts. Once the voltage drops below the level, the red LED instead is set on, reminding the user to change battery.

2.3.2 Microcontroller (TI MSP430F5324)

Input: Keypad signal, 3.3V±5% power supply

Output: Bluetooth control signal, keystrokes pattern signal for LCD screen

This unit serves as the “brain” of the whole keypad. For normal mode, when a keystroke is pressed, the microcontroller will go through the process of identify the keystroke (its column and row), and pull out the ASCII code for that key from the look-up table. Then it transmits that processed ASCII code signal to LCD screen to display it. In addition, it also sends appropriate signal to the host (HM-10 Bluetooth transceiver). For keypad program mode, when the user wants to change or add new function for a keystroke, the microcontroller receive data from the computer via Bluetooth transceiver, and reassign or add new ASCII code to the corresponding key.

The TI MSP430 will store the keystrokes functionalities, which are basically the ASCII value for each symbol in the character set, such as letters, digits, punctuation marks, special characters, and control characters. It will be powered by a regulated 3.3V source from 9V battery, and integrated with the LCD screen, Bluetooth module, keypad matrix, on a PCB.

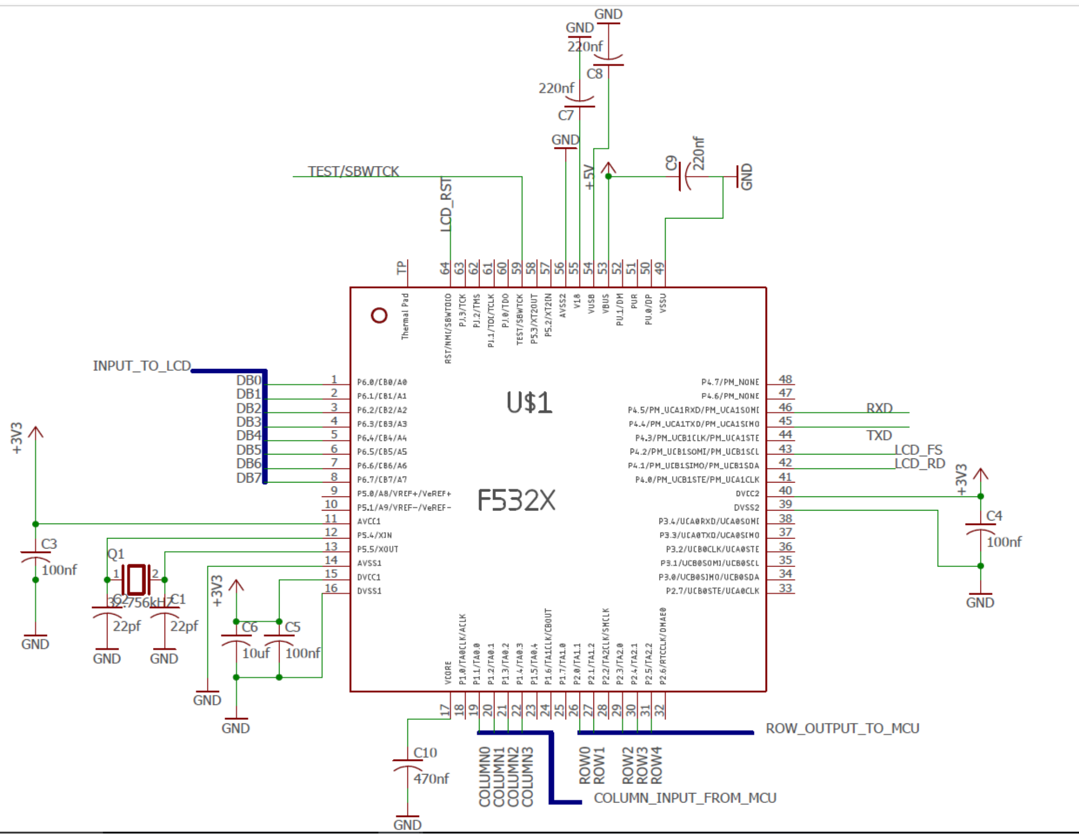

Figure 5: Schematic for the microcontroller

2.3.3 Bluetooth module

Input: Command from the microcontroller, 3.3V±5% power supply

Output: Data transmitted to the host device (Bluetooth protocol)

Our Bluetooth module will act as the device to transfer data wirelessly between the intended host device and the microcontroller. A HM-10 Bluetooth transceiver is used and by taking advantage of the existing firmware we can communicate with our MCU effectively to attain our desired functionality. The HM-10 is a small 3.3V SMD Bluetooth 4.0 BLE module based on the TI CC2541 Bluetooth SoC (System on Chip).

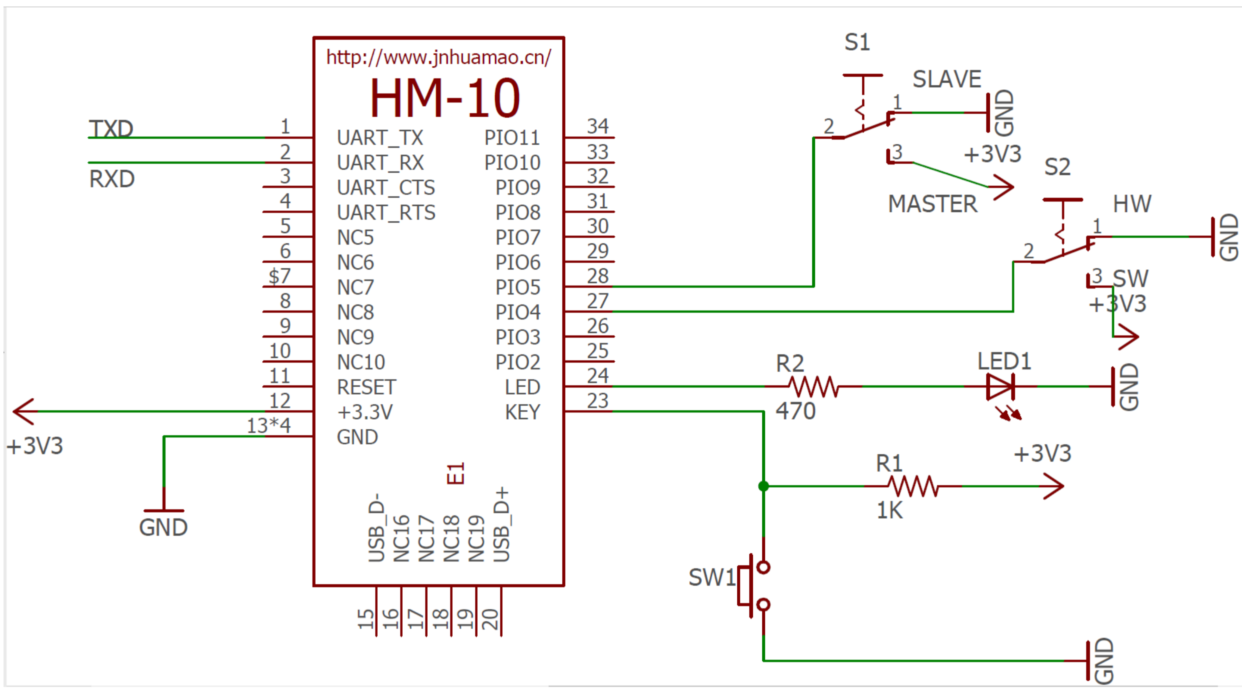

Figure 6: Schematic for the HM-10 Bluetooth module

2.3.4 User Interface

2.3.4.1 LCD Screen

Input: Keystrokes pattern signal from the microcontroller, 5V±5% power supply

Output: Keystrokes pattern display on the LCD screen

We intend to use a 240x128 pixels screen. The LCD screen is used to show the pattern of each keystrokes. This way the user can easily know what functionality or pattern they assign to the keypad.

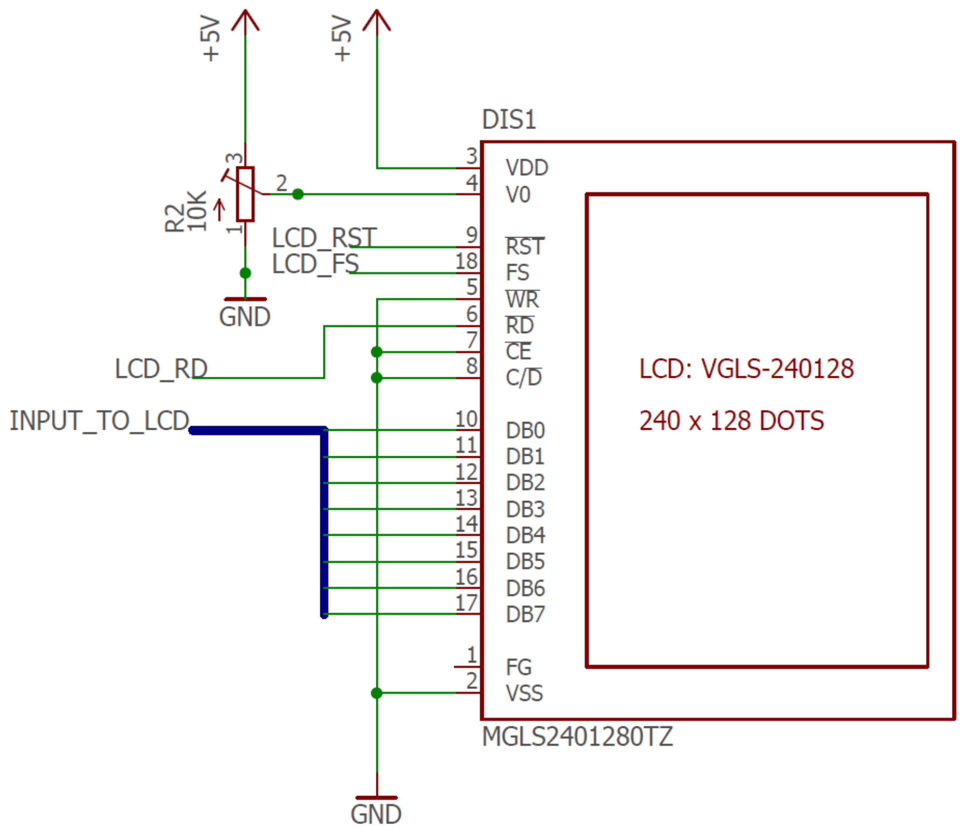

Figure 7: Schematic for the 240x128 pixel LCD

2.3.4.2 Keypad matrix

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

Output: Signal to the microcontroller

We intend to build a 4x4 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. To achieve this, all the keystroke inputs will go towards our

MCU through our data bus, and making use of the processing unit we can manipulate the output signal that's intended to be sent towards the receiver.

The keypad matrix is controlled by the microcontroller. For a 16-button 6x4 matrix with 4 rows and 4 columns, 9 pins of the microcontroller will be connected to the matrix. 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 4 pin will be OUTPUTS of the microcontroller 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.

To implement our design, we will be using button switches as our keys, because they are the most cost-efficient option for us.

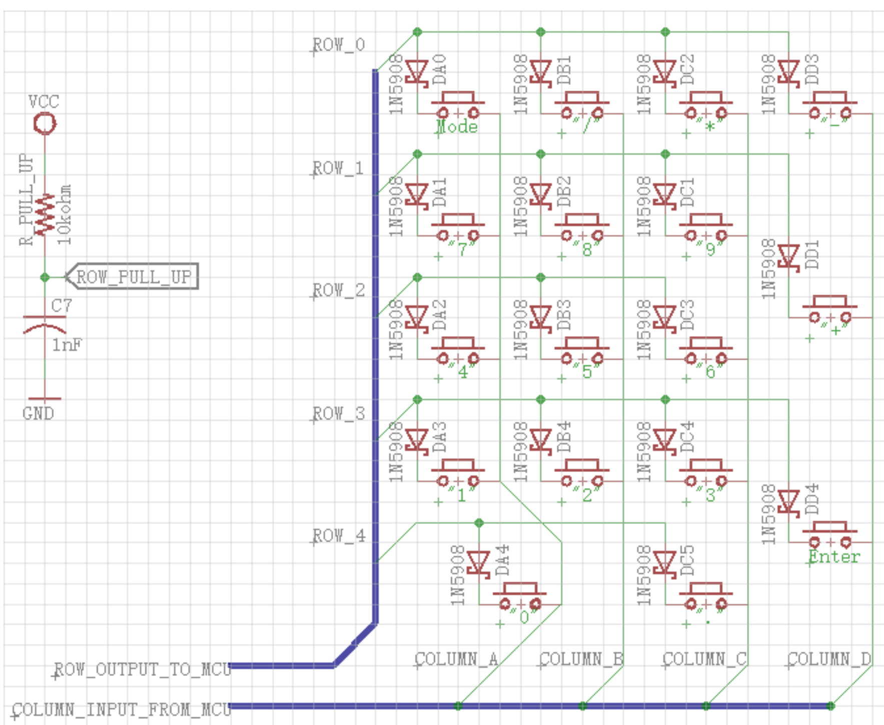

Figure 8: Schematic for the keypad matrix

2.4 Calculation and Simulation

2.4.1 Power Supply Resistors

The output voltage of LM317T is given by its datasheet as

$$V_{out} = V_{ref}(1 + \frac{R_2}{R_1}) + I_{adj}R_2. \tag{2.4.1.1}$$

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, in order to acquire the desired voltage, the value of R_2 can be determined by the equation

$$V_{out} = 1.25(1 + \frac{R_2}{240}). \tag{2.4.1.2}$$

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

$$R_2 = 240(\frac{3.3}{1.25} - 1) = 393.6\Omega \tag{2.4.1.3}$$

$$R_2 = 240(\frac{5}{1.25} - 1) = 720\Omega \tag{2.4.1.4}$$

for output voltage of $3.3V$ and $5V$, respectively.

2.4.2 Voltage Divider Resistors

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.

2.4.3 Pull-up Resistors for LED

The voltage drop of a single LED branch is

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

(2.4.3.1)

According to datasheet, a typical surface mount LED has a forward current of $20mA$, which causes a voltage drop of $0.2V$. The LED is connected active-low, and V_{ol}, in the worst situation, is $0.3V$ above the ground level. The pull-up resistor is therefore

$$R_{pull-up} = \frac{V_{cc} - V4e_{ol} - V_f}{I_f} = \frac{3.3 - 0.3 - 0.2}{20 \times 10^{-3}} = 140\Omega$$

(2.4.3.2)

However, the current of $20mA$ in the above calculation seems too large, and be an overkill to light up a wee LED. As a result, the magnitude of the current might be lowered in practice and the resistance of the pull-up resistor is subject to change.

2.4.4 Energy Consumption

Table 1: Total power consumption of the keypad

Component	Voltage	Current	Unit Number	Power
Microcontroller	3.3 V	0.29 Ma	1	0.957 mW
LCD screen	5 V	126.2 mA*	1	631 mW
Key switch	3 V	0.01 mA	16	0.48mW
Bluetooth module	3.3 V	50 mA	1	165 mW
Total		176.5 mA		797.437 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 assuming that the current discharge of our keypad is 176.5 mA. The estimate hours of operation of the keypad is

$$Hours\ of\ operation = \frac{Milliamp - hours\ capacity}{Discharged\ current} = \frac{400mAh}{176.5mA} = 2.26\ hours$$

The result indicates that the keypad will last for about 2.26 hours. This assume that the LCD screen supply current is at its maximum value. Often time, the LCD screen supply current will not reach its maximum value, so the keypad can operate for longer time than the calculated value.

2.4.5 Simulation

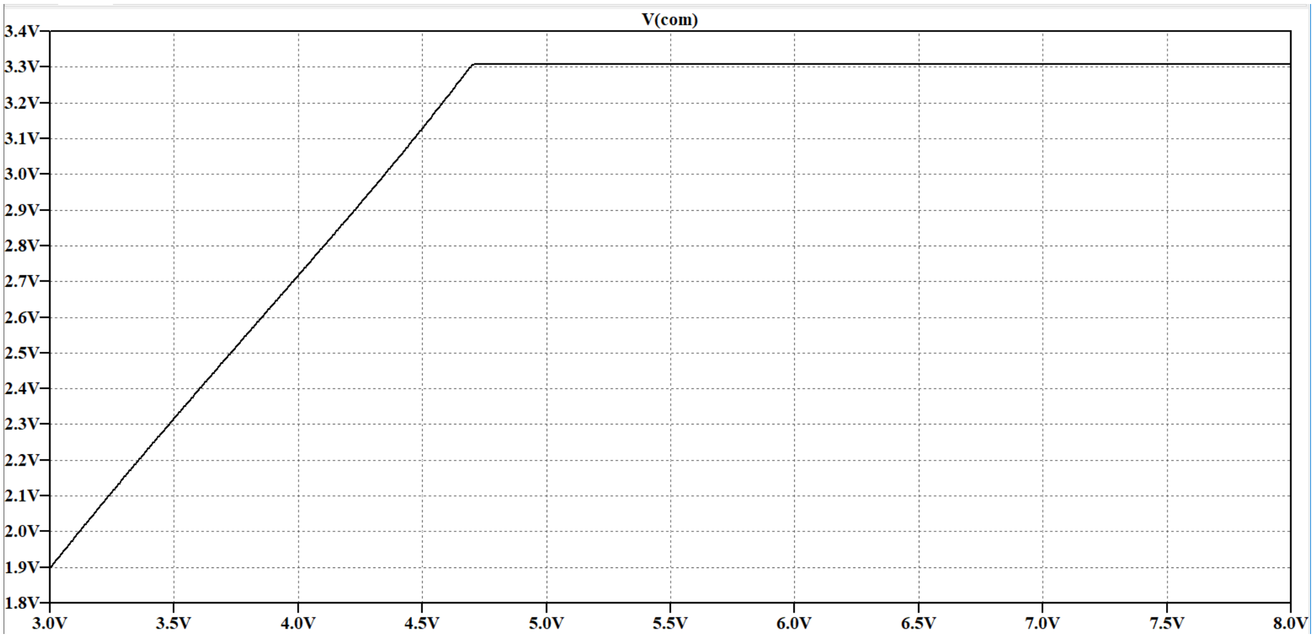

Figure 9: The output voltage for 3.3V Regulator ($R_2 = 393.6\ \Omega$)

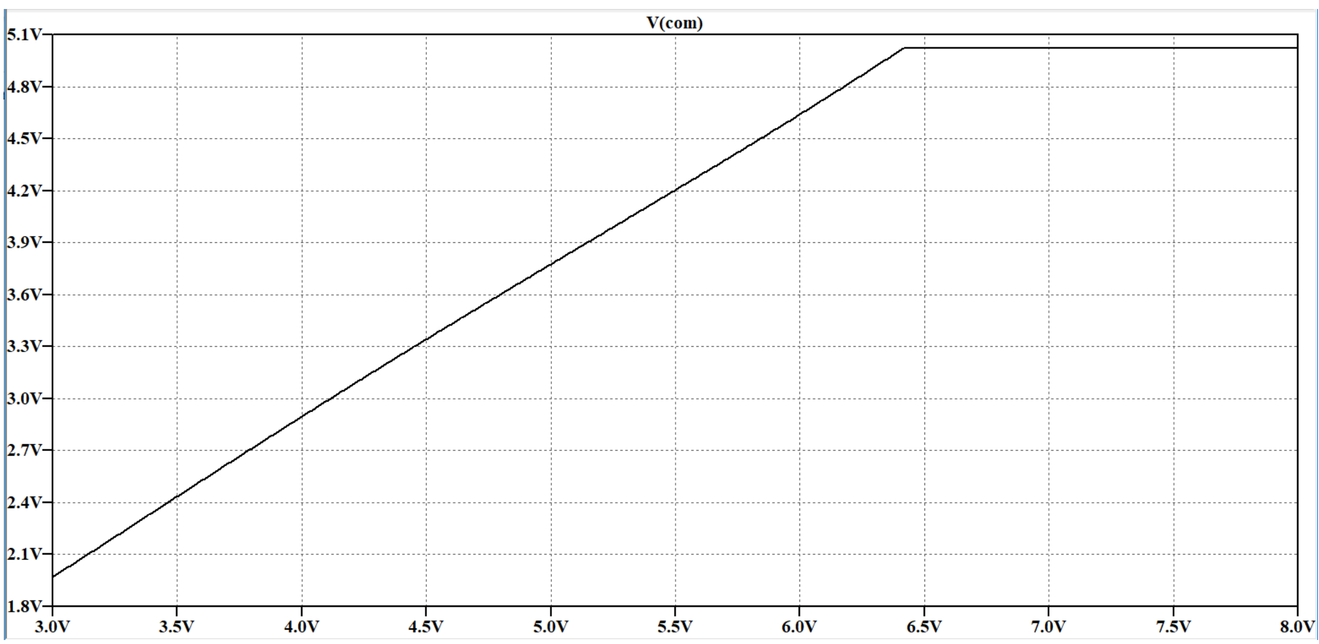

Figure 10: The output voltage for 5V Regulator ($R_2 = 720\ \Omega$)

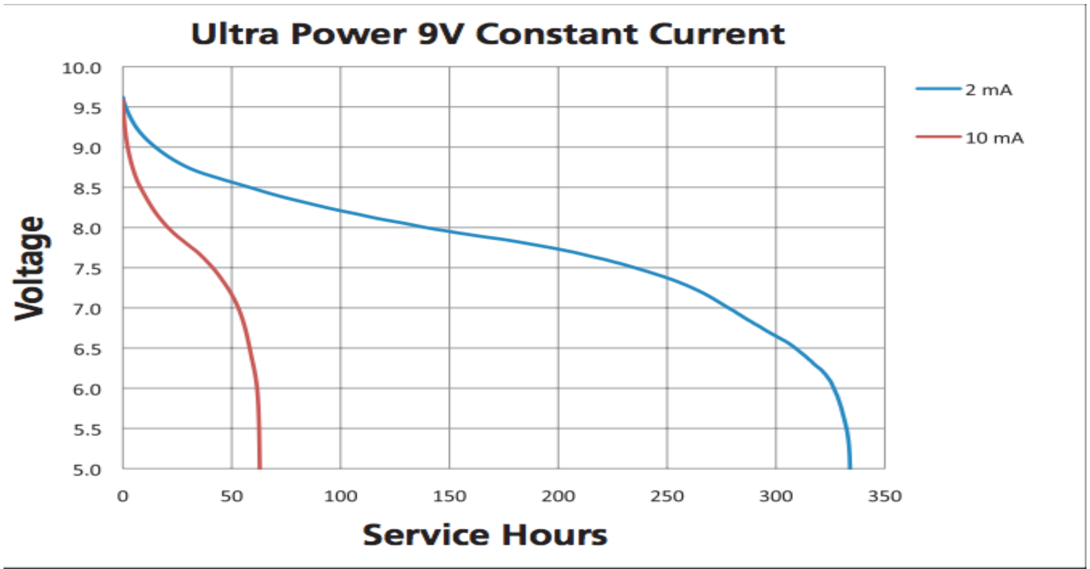

Figure 11: Output Voltage over Time for a standard Duracell Ultra Power 9V Battery [6]

2.5 Software Design

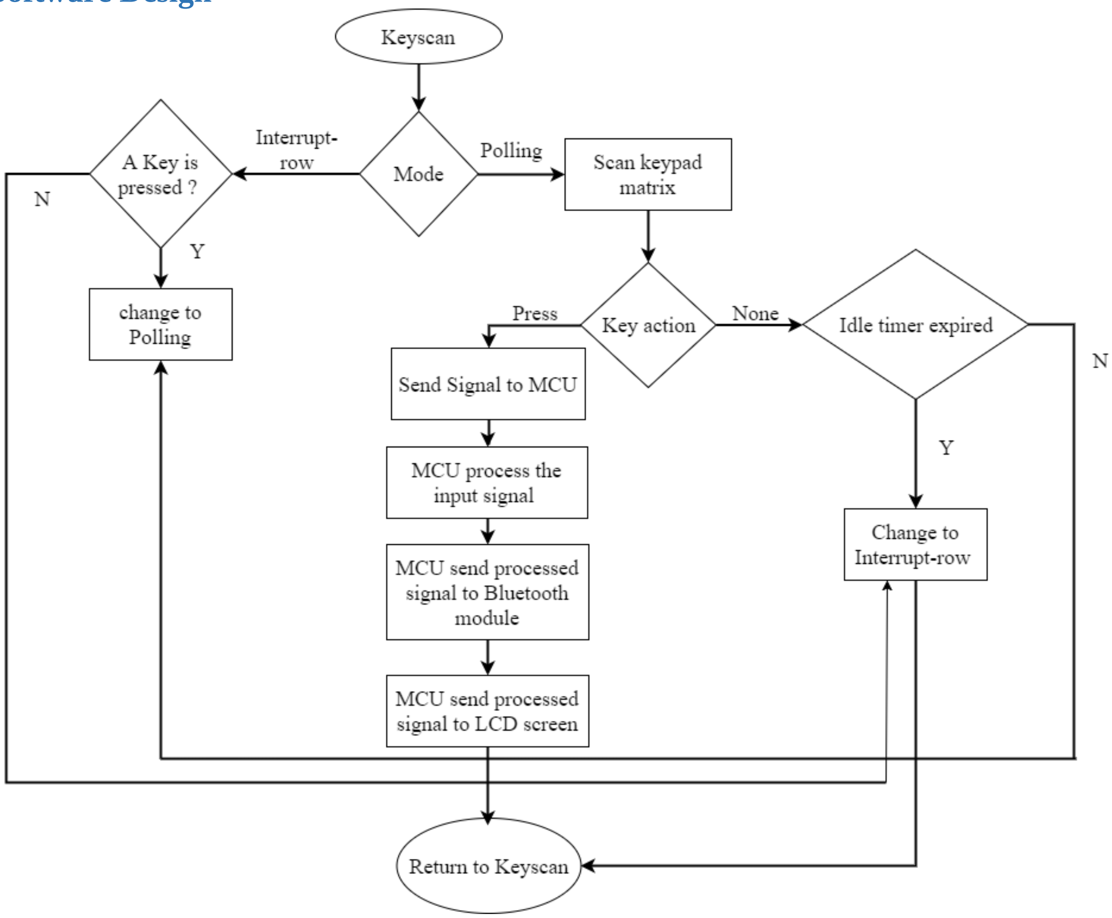

Figure 12: Key switch detection flow chart

The keypad has two modes. The default mode is interrupt-row, and polling mode is only triggered when a key is pressed. In both mode, the MCU will scan the key board for pressed key. If no key is pressed (no input signal) then the MCU will not do anything. When the key is pressed the keypad change to polling mode. In this mode, the MCU receives an input signal, and process that signal by going through the look-up table and assign the corresponding ASCII code for the pressed key stroke. Then it sends the ASCII code to the bluetooth module via TXD connection, so it can send that data to the computer. After that, it will send the ASCII code to the LCD screen to display the corresponding pattern. In the polling mode, there is an idle timer, which is 30s. When no key is pressed during polling mode, and the idle timer expires, the keypad will change back to row-interrupt mode to save energy.

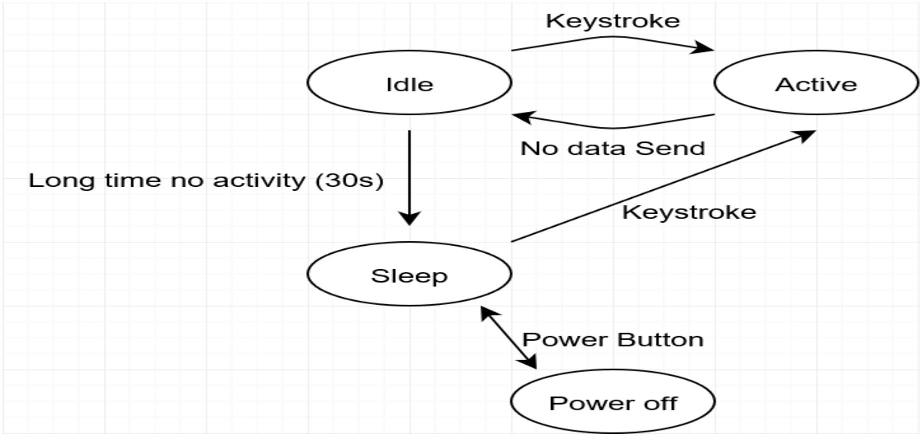

Figure 13 State diagram of the keypad

For the programmable function, we intend to write a script (for simplicity) in the computer that allow the user to change the ASCII code of each corresponding key stroke. Once the user finishes the reassignment, the computer will send that data to the microcontroller via bluetooth. The microcontroller will reassign the ASCII code to the correct key stroke based on the receive data.

3. Requirement and Verification

Table 2: The requirement and verification for Power System [10 points]

Requirement	Verification	Point
Voltage Regulator		4
1. The voltage across the output terminal and ground must be 3.3/5V with a maximum error of ±5% when the battery is fully charged.	1a. Connect *E3631A Triple Output DC Power Supply* to the input of the regulator circuit and tune its output voltage to 9.5V 1b. Connect *34461A 6$^{1}/_{2}$ Digit Multimeter* to the output of the circuit and measure the voltage.	2
2. The voltage must remain stable within the error range as the battery is discharged to 6.5V	2a. Sweep down the output of power supply slowly and read the measurement shown on the multimeter.	2
(continue on next page)		

Requirement	Verification	Point
Voltage Divider		2
3. The output voltage is 1.65V when the battery is discharged to 7V	1a. Repeat 1a and 1b with the tuned output voltage of 7V	2
Voltage Level Indicator		4
4. The red and green LEDs must be active low , turning on with logic low and off with logic high	2a. Repeat 1a. with the tuned output voltage of 3.3V 2b. Connect another power supply to the input of the circuit and tune it to 0.3V. This is the highest value logic LOW, both LEDs are expected to be on 2c. Repeat step 4b setup with the tuned output voltage of 3V. This is the lowest value for logic HIGH, both LEDs are expected to be off	4

Table 3: The requirement and verification for Keypad matrix [20 points]

Requirement	Verification	Point
Key Switch matrix		20
1. The keypad should be free of ghosting problem.	1a. Build a 2x2 mini key pad using the same schematic. 1b. Connect the mini key pad to the microcontroller with LEDs showing the input signals. 1c. Keep the microcontroller from performing key scanning. 1d. Hold arbitrary three keys and the corresponding LEDs should light up (blink). 1e. Hold the 4^{th} key and check if the LED is on.	10
2. The keypad should be free of masking problem	2a. Repeat steps 1a-1d and check if the LED is off.	5
3. The bounce time of any key is less than 10ms.	3a. Connect the power supply to the Vcc and tune its output voltage to 3.3V. 3b. Connect HDO4054-MS Oscilloscope to the output of the circuit. 3c. Click a key and measure the output voltage against time.	5

Table 4: The requirement and verification for Microcontroller [10 points]

Requirement	Verification	Point
Microcontroller		10
1. The microcontroller is able to send correct bit pattern to the LCD screen	1a. Use a predefined bit pattern and send it to microcontroller 1b. Use multimeter to measure the output on the LCD connection end	5
2. The microcontroller is able to send data to the bluetooth module	2a. Use a predefined bit pattern and send it to the bluetooth via microcontroller 2b. The computer receives the desired data	5

Table 5: The requirement and verification for Bluetooth Module [5 points]

Requirement	Verification	Point
Bluetooth module		5
1. The bluetooth module is able receive data from the host device	1a. Send a predefine bit pattern from the computer to the keypad via bluetooth 1b. The bit pattern is displayed on the LCD screen	3
2. The bluetooth module is recognized by the host device	2a. Set the bluetooth module to advertising mode. 2b. Check on the computer if it can recognize it	2

Table 6: The requirement and verification for LCD screen [5 points]

Requirement	Verification	Point
LCD Screen		5
1. Display a hard coded test string of characters	1a. Use predefined bit pattern and send it to the screen via microcontroller 1b. The LCD displays the correct bit pattern	5

4. Tolerance Analysis

The key switches, a fundamental component in the design, involves mechanical engineering and we cannot neglect a phenomenon called switch bounce. The problem is that the voltage across a real switch in a circuit with a pull-up resistor does not go instantly and cleanly between V_{cc} and the ground. It is possible that the switch makes multiple transitions or partial transitions with pauses on the way, or just changes slowly on the time scale of a microcontroller. Our device should be capable of accommodating these deficiencies rather than act as if the switch has been pressed multiple times when the user clicked only once.

There are a multitude of methods for debouncing in software, and the simplest one is to detect a transition from the switch, wait for a fixed delay, and test the input again. The input is accepted as a valid value only if remain the same. We could also accomplish the debouncing task with a hardware approach, given that each input of MSP430 is provided with a Schmitt trigger, which, unlike the conventional buffer, never yield an output with an undefined logic value.

Among the many parameters, V_{IT+} and V_{IT-} are of the two most significance for a Schmitt trigger, which determine the positive and negative-going input threshold voltage. The third parameter V_{hys} is the difference of the aforementioned voltages. The minimum hysteresis sets the noise margin of the input, guaranteeing that as long as the noise is kept below this critical level, the device will not trigger an unwanted transition.

We can therefore exploit the feature of Schmitt trigger, adding an RC filter to the pull-up circuit. The resistor and capacitor slow down the signal seen by the microcontroller and smooth out rapid changes caused by bouncing.

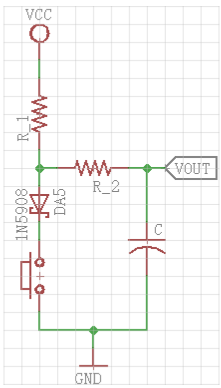

Figure 13: RC filter and pull-up resistor schematic

Suppose the capacitor is first charge to V_{cc}, then switch is clicked and the resistor and the capacitor are connected in a short circuit at t = 0. The capacitor discharges and the voltage of output is given by

$$V_{out}(t) = V_{discharge}(t) = V_{cc}e^{-\frac{t}{R_2C}} \tag{4.1}$$

Similarly, if the capacitor is discharged to zero and the switch is released, the voltage of output is given by

$$V_{out}(t) = V_{charge}(t) = V_{cc}(1 - e^{-\frac{t}{(R_1+R_2)C}}) \tag{4.2}$$

The value of R_1, R_2, and C should be chosen properly to handle the worst situation, in which hysteresis V_{hys} is at its minimum of $0.4V$. We further assume for simplicity that the thresholds are symmetric between $V_{cc}$and ground, yielding $V_{IT+} = 1.85V$ and $V_{IT-} = 1.45V$. We then set the scenario in which the key switch has been hold for a long time and is then released.

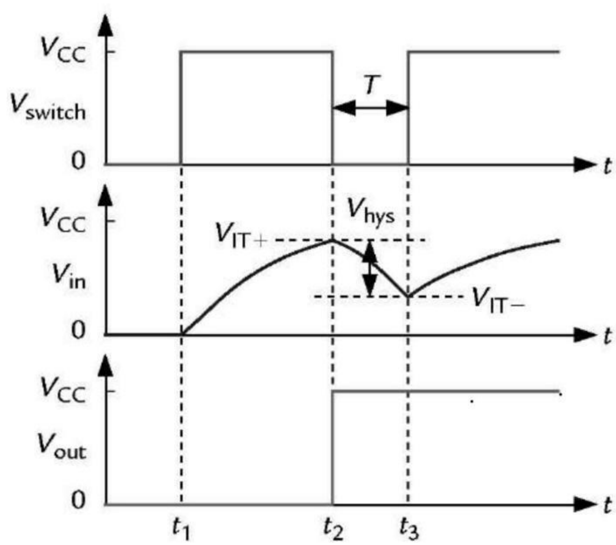

Figure 14: Capacitor charging plot for the worst case scenario when the key switch has been hold for a long time and is then released

The switch is opened at t_1 and the capacitor begins to charge through R_1 and R_2 with time constant $\tau_1 = (R_1 + R_2)C$. At t_2 the input to the Schmitt trigger crosses the positive-going threshold voltage V_{IT+} and the output goes from logic low to high. Immediately afterward, the switch bounces back and the capacitor C begins to discharge through R_2 with time constant $\tau_2 = R_2C$. The switch then bounces closed again at time t_3, right before the voltage reaches the negative-going threshold voltage V_{IT-}. A false transition caused by switch bounce is therefore avoided and the output voltage raises as C charges again.

The critical part of this scenario is from between t_2 and t_3, during which C discharges with time constant τ_2. If the time interval $T = t_3 - t_2$ was slightly longer, the debouce circuit would fail since the output had been triggered low. According to the equation (4.1), the voltage at t_2 and t_3 is

$$V_{out} = V_{cc}e^{-\frac{t_2}{R_2C}} = V_{IT+} \tag{4.3}$$

$$V_{out} = V_{cc}e^{-\frac{t_3}{R_2C}} = V_{IT-} \tag{4.4}$$

Dividing the two equations we get

$$\frac{V_{IT+}}{V_{IT-}} = e^{\frac{t_3-t_2}{R_2C}} = e^{\frac{T}{R_2C}} \tag{4.5}$$

Eventually, we can take the natural log of both sides to yield

$$T = R_2Clog_e(\frac{V_{IT+}}{V_{IT-}}) = 0.2436R_2C \tag{4.6}$$

The bounce time of our MX series key switch is less than $5ms$, and with a C of $0.1\mu F$ the resistance of R_2 is

$$R_2 = \frac{T}{0.2436C} = \frac{5 \times 10^{-3}}{0.2436 \times 0.1 \times 10^{-6}} = 205k\Omega \tag{4.7}$$

We could also analysis for the opposite scenario for switch having been release and then being clicked, but the result will be the same because the threshold voltages are symmetric. The only thing changed is the time constant, which is $\tau_1 = (R_1 + R_2)C$. As a result,

$$R_1 = \frac{T}{0.2436C} - R_2 = \frac{5 \times 10^{-3}}{0.2436 \times 0.1 \times 10^{-6}} - 205 = 0 \tag{4.8}$$

The result indicates that R_1 can be as small as possible if only the debouncing is concerned. However, we do not want to waste current, so a resistance of $10k\Omega$ should be reasonable.

5. Ethics and Safety

We will put our best effort to keep our ethics in accordance with IEEE Code of Ethics #1,3,5 and 7[7].

The most important concern of our lies in the power supply side. Though we only making a keypad, a harmless device, we realize that there could be unexpected dangers of using the power, and we will do our best to minimize the potential power and circuit failure with in accordance with IEEE code of ethics #1 by optimizing the both hardware and software component. Our power supply is an Alkaline battery, which contains both cathode and fuel in a container. If these two parts react, it can trigger explosion or fire. In addition, an explosion can also be caused by overheating the battery. To maintain safety, we will closely monitor the temperature of the battery to make sure that our battery stays within acceptable range when the keypad is used, and disconnect the battery from the entire circuit if the temperature is abnormal to avoid potential fire hazard or explosion.

Our project involves in complex hardware and software design therefore to fulfill IEEE Code Ethics #5, we need sufficient background knowledge to ensure we not only successfully implement the system but also produce a quality product.

Under IEEE code of ethics #7, we will openly listen to TA's criticism and advice since they are useful in making sure we are on the right track. As we're aware of, there are existing products on the market that offers similar functionalities to our project, and very likely patented for their design. Although we have no intentions thus far to market this project commercially, we have to credit sources properly and avoid plagiarism to the best of our abilities.

We are aware that during our design process, there will be time when our tests and experiments went wrong and we will not yield good result. But we know that is not an excuse to violate IEEE code of ethics #3 by forging fake data to meet the class requirement. We will honest with what we data we have even though it may not be the ideal result.

6. Cost Analysis and Schedule

6.1. Cost Analysis

6.1.1. Parts

Table 7: Components Costs

Part Name	**Part Number**	**Unit Cost**	**Quantity**	**Total**
Microcontroller	TI MSP430F5324	$5.10	1	$5.10
Bluetooth module	HM-10	$12.99	1	$12.99
Battery		$1.88	1	$1.88
Voltage Regulator	LM317LCPK	$0.69	2	$1.38
240x160 Pixel LCD	ERC240160SBS-2	$9.19	1	$9.19
Key switch	MX1A-11NW	$0.79	20	$15.80
Passive Components		$7.78		$7.78
Total				$54.12

6.1.2. Labor

Table 8: Labor Costs

Name	**Hours Invested**	**Hourly Rate**	**Total Cost = Rate * Hours*2.5**
Nguyen Phan	200	$30	$15000
Yihan Liu	200	$30	$15000
Wei-Tang Wang	200	$30	$15000
Total	600		$45000

6.1.3. Grand Total

Table 9: Total Costs (Component Costs + Labor Costs)

Parts	**Labor**	**Grand Total**
$54.12	$45000	$45054.12

6.2. Schedule

Table 10: Project Schedule and Task Allocation

Week	**Task**	**Delegation**
9/18/2017	Finalize project proposal	Nguyen Phan
	Design Keypad matrix with control circuit	Yihan Liu
	Come up with the physical design	Wei-Tang Wang
9/25/2017	Research and select microcontroller	Nguyen Phan
	Research and select bluetooth module	Yihan Liu
		(Continue on next page)

Week	**Task**	**Delegation**
9/25/2017	Research and select LCD screen	Wei-Tang Wang
10/2/2017	Finalize the design document	Nguyen Phan
	Prepare for design review	Yihan Liu
	Figure out bluetooth protocol	Wei-Tang Wang
10/9/2017	Study datasheet for microcontroller, bluetooth module	Nguyen Phan
	Program microcontroller	Yihan Liu
	Purchase hardware & all parts	Wei-Tang Wang
10/16/2017	Testing the microcontroller	Nguyen Phan
	Design PCB	Yihan Liu
	Testing the bluetooth module	Wei-Tang Wang
10/23/2017	Interfacing microcontroller with the keypad	Nguyen Phan
	Finalize/ Order the PCB with all components included	Yihan Liu
	Interfacing microcontroller with LCD	Wei-Tang Wang
10/30/2017	Assemble the PCB	Nguyen Phan
	Writing software for programming the keyboard	Yihan Liu
	Run tests on final project as a whole	Wei-Tang Wang

(Continue on next page)

Week	**Task**	**Delegation**
11/6/2017	Test/Debug System for microcontroller	Nguyen Phan
	Take power measurements for tolerance analysis with full system	Yihan Liu
	Test/Debug System for wireless communication	Wei-Tang Wang
11/13/2017	Optimization for microcontroller	Nguyen Phan
	Prepare for mock demo	Yihan Liu
	Optimization for wireless communication	Wei-Tang Wang
11/20/2017	Sign-up for demo	Nguyen Phan
	Prepare demo	Yihan Liu
	Prepare presentation	Wei-Tang Wang
11/27/2017	Prepare final paper	Nguyen Phan
	Sign-up for presentation	Yihan Liu
	Prepare presentation	Wei-Tang Wang
12/4/2017	Turn in Final Paper	Nguyen Phan
	Proofread Final Paper	Yihan Liu
	Lab Checkout	Wei-Tang Wang

7. Citations

[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]Amazon.com, “XK-24 USB Programmable Keypad for Windows or Mac”. [Online].Available:

https://www.amazon.com/XK-24-USB-Programmable-Keypad-Windows/dp/B003MB780E. Accessed 4 Oct. 2017

[5]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

[6] Batterysales, “Duracell MX1604 Alkaline-Manganese Dioxide Battery”, 2017. [Online].

Available:

http://www.batterysales.nl/files/downloads/Battery_Sales_Europe_Ultra_Power_9V_MX1604.p

df . Accessed 4 Oct. 2017.

[7] IEEE.org , "IEEE IEEE Code of Ethics", 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. Accessed 4 Oct. 2017.