EE447 Term Project

PROJECT REPORT

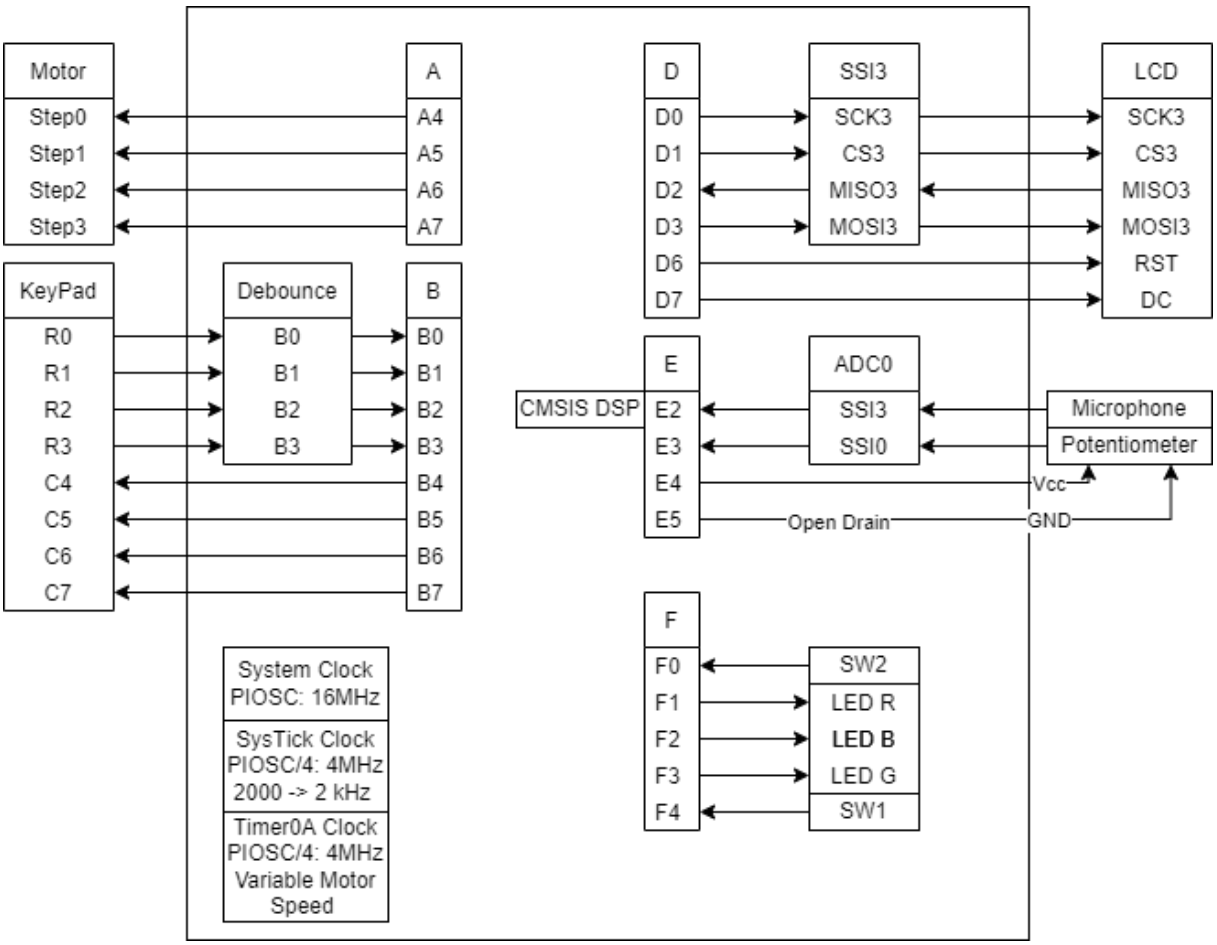
Ataberk ÖKLÜ
METU EEE | 2305142

Table of Contents

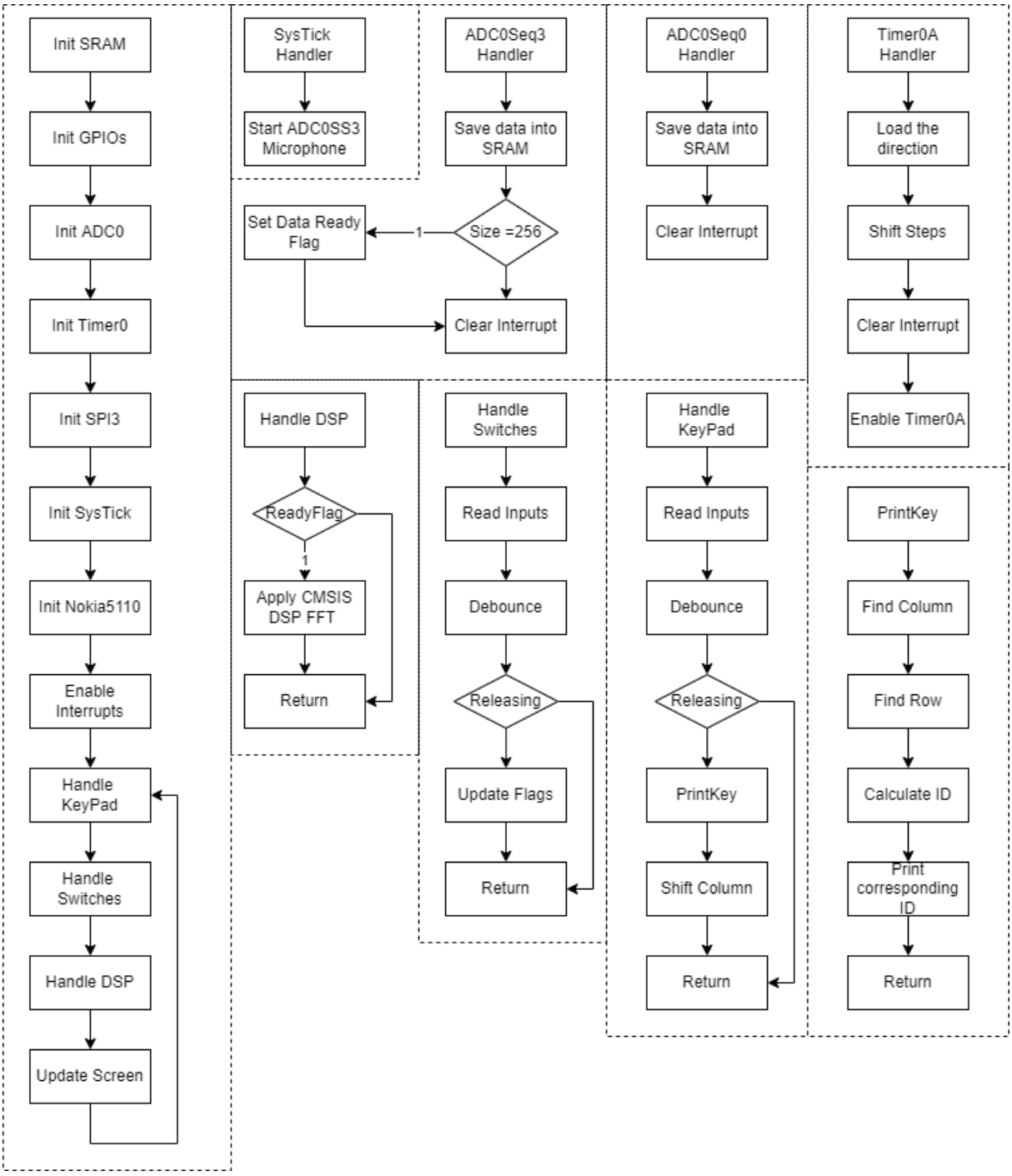
Requirements

- Microphone output should be sampled at 2kHz by ADC
- 256 Samples must be stored in SRAM
- The samples must be converted regarding the DC offset of 1.25V
- Converted samples must be converted using CMSIS DSP FFT Implementation.
- The controller must handle a 4x4 Keypad, such that only releasing the button triggers the action.
- Low and High Thresholds can be set using Keypad.
- The controller must drive Stepper Motor regarding the input frequency.
- A potentiometer should be sampled in order to set the amplitude threshold.
- The controller must handle Nokia 5110 LCD.
- The onboard switches control the motor directions.
- The color and the brightness of the onboard LED should be managed according to the input frequency and amplitude.

System Overview



Flowchart


Init SRAM
Init GPIOs
Init ADC0
Init Timer0
Init SPI3
Init SysTick
Init Nokia5110
Enable Interrupts
Handle KeyPad
Handle Switches
Handle DSP
Update Screen
SysTick Handler
Start ADC0SS3 Microphone
ADC0Seq3 Handler
Save data into SRAM
Size =256
1
Set Data Ready Flag
Clear Interrupt
ADC0Seq0 Handler
Save data into SRAM
Clear Interrupt
Timer0A Handler
Load the direction
Shift Steps
Clear Interrupt
Enable Timer0A
Handle DSP
ReadyFlag
1
Apply CMSIS DSP FFT
Return
Handle Switches
Read Inputs
Debounce
Releasing
Update Flags
Return
Handle KeyPad
Read Inputs
Debounce
Releasing
PrintKey
Shift Column
Return
PrintKey
Find Column
Find Row
Calculate ID
Print corresponding ID
Return

Modules

Initiations

This part consists of GPIO, ADC0, Timer0, SSI3, SysTick, SRAM, and Nokia 5110 initiations. GPIO initiations include the Alternative Functions as well. The Summary of Peripheral Initiations:

- ADC0
  - SS0: Potentiometer, one-shot, interrupt enabled, Pin AIN1 (E2), Priority 2
  - SS3: Microphone, one-shot, interrupt enabled, Pin AIN0 (E3), Priority 1
- Timer0:
  - Timer0A: Motor Driving, 4 MHz Clock, One-shot, Counting Down, Timeout Interrupt Enabled, Priority 2, Slow Reload Value.
- SysTick:
  - Generating 2 kHz Sampling Rate for ADC0SS3 and Enabling 1-Sec Update Routine. Enable Interrupt, Priority 1.
- SSI3:
  - 4 MHz CLK, 8-bit Frame, FreeScale Mode.

IRQs / ISRs

ADC0SS0 Handler

Responsible for setting the Amplitude Threshold by measuring Potentiometer Voltage and updating the Threshold value presented on the Screen.

ADC0SS3 Handler

Responsible for reading the microphone input with a 2 kHz sampling rate. The read data is scaled up to 16-bit and stored as interleaved, real, and imaginary values. When 256 samples are collected, Data Ready Flag is set in order to process them in the Handle_DSP Routine.

$$\{\text{real}[0], \text{imag}[0], \text{real}[1], \text{imag}[1], \dots\}$$

SysTick Handler

Each interrupt serving triggers the ADC0SS3 to read one data with a 2 kHz Sampling Rate and increment a counter which tracks the 1-sec intervals. When 1 second is elapsed, Update Screen Flag is set.

Timer0A Handler

This handler manages the stepper motor steps, and the reload value (Motor Speed) is updated according to the dominant frequency resulting from FFT. The motor speed is only updated when amplitude of the dominant frequency is higher than the Amplitude Threshold set by Potentiometer, else it is kept as current speed.

Main Handlers

Handle KeyPad

The keypad input is software debounced, and the releasing button triggers the Key Function. Each key is printed to the UART0 (38400 bps | 8 Data Bits | 1 Stop Bit | No Parity). “A” and “B” keys have a particular function, respectively, setting Low-Frequency Threshold and High-Frequency Threshold. When “A” or “B” is activated, the user is expected to input maximum of 3-digit threshold in Hz. The Threshold setting process is also be terminated or cancelled by pressing the same function button. The Serial Terminal can also be actively used during this process. There is no delay or stuck during the input process.

One important point to emphasize is that PD0 – PB6 and PD1 – PB7 pairs are interconnected by the 0Ω shunt resistors R9 and R10 onboard as stated in [1]. This results in a conflict between Port B GPIO and Port D SSI3 modules and Port D SSI3 module overwrites it. Therefore, keypad usage is distorted. Henceforth, I removed these R9 and R10 shunt resistors to use both modules together.

Handle Switches

The Switches are used to change the direction of motor rotation by manipulating the GPIO B outputs. This function has its own debounce Routine for switches.

Handle ADC0SS3 DSP

If the Data Ready Flag is not set, it skips this Routine. When the 256 samples are collected and ready to be processed, the first FFT is applied using arm_cfft_q15 Routine. Then, magnitude is calculated, and found the bin with the maximum amplitude, and its ID represents the frequency.

$\frac{2000\,Hz}{256} = 7.8125$ is the Hz Step Size for the frequency bins. Hence, the resulting frequency in Hz is calculated using $Freq = ID * 7.8125$, which is practically calculated in the code by $Freq = ID * \frac{7813}{1000}$.

This Routine updated the Amplitude and Frequency ID variables and corresponding areas of the Screen.

Update Screen

This Routine is responsible for a 1-second update for Screen. If Update Screen Flag is not set, this Routine is skipped. First, LED statuses are set and updated screen memory when it is set. Then, the whole Screen Memory is read, and corresponding characters are written via SPI3 into the Nokia 5110 RAM (Total 504 Bytes).

Screen Memory

There are two distinct memory areas for 6x8 Font and the character IDs for each row. This design enables flexibility of changing the Font Style and size of each character without the need to change the memory holding the texts on the Screen. I have chosen a 6x8 Font Size since all columns of each row is 8bit height, and each row has 84 columns, which corresponds to $\frac{84}{6} = 14$ characters can be written to each row. There are six rows. The Screen Design is shown below.

```
; 14 Chars (6bit*14 = 84) in each row.
; 48 / 8 = 6 row;
;  _________________
; |THRS1: ---     |  ; Amplitude      (+)
; |THRS2: --- Hz  |  ; Low    - 300   (+)
; |THRS3: --- Hz  |  ; High   - 600   (+)
; |FREQ : --- Hz  |                   (+)
; |AMPLT: ---     |                   (+)
; |LED  : -       |  ; 0: OFF (1,2,3) (+)
; ----------------
```

There are two variables in memory:

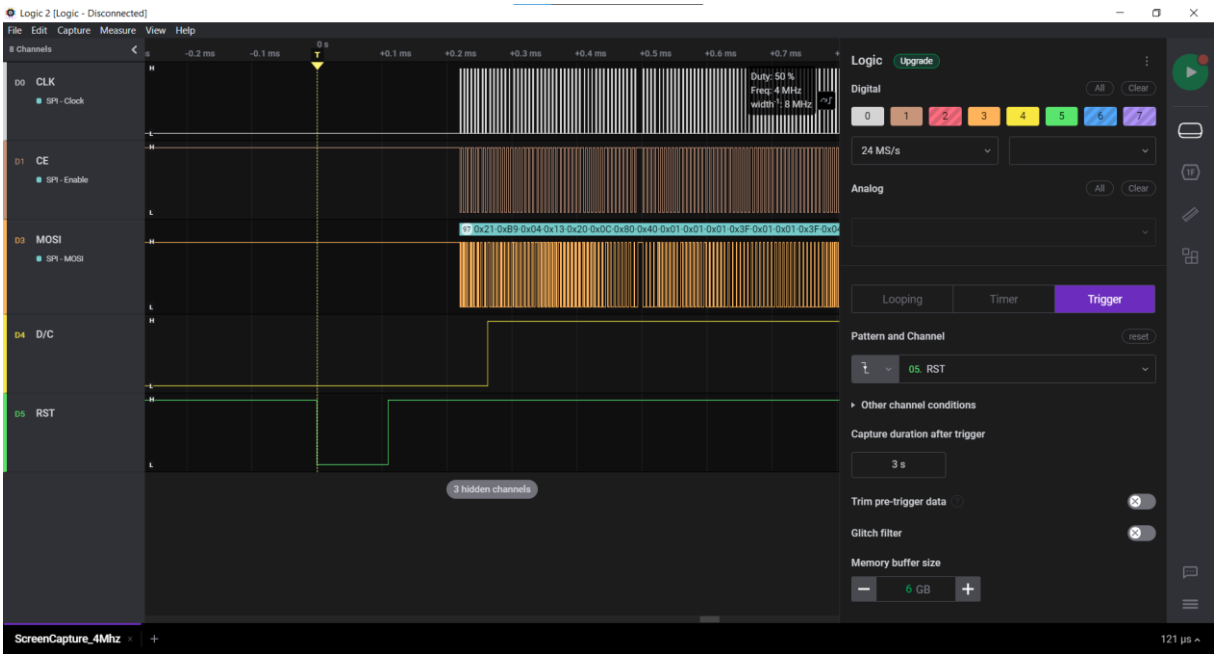
- Screen Data: Holds the character IDs defined in the Font6x8 variable.
- Font6x8: Holds the 6 bytes for each character. Retrieved from the source given [2]. There are different styled and sized fonts, and I can easily switch between them.

The Screen Data Variable is shown below.

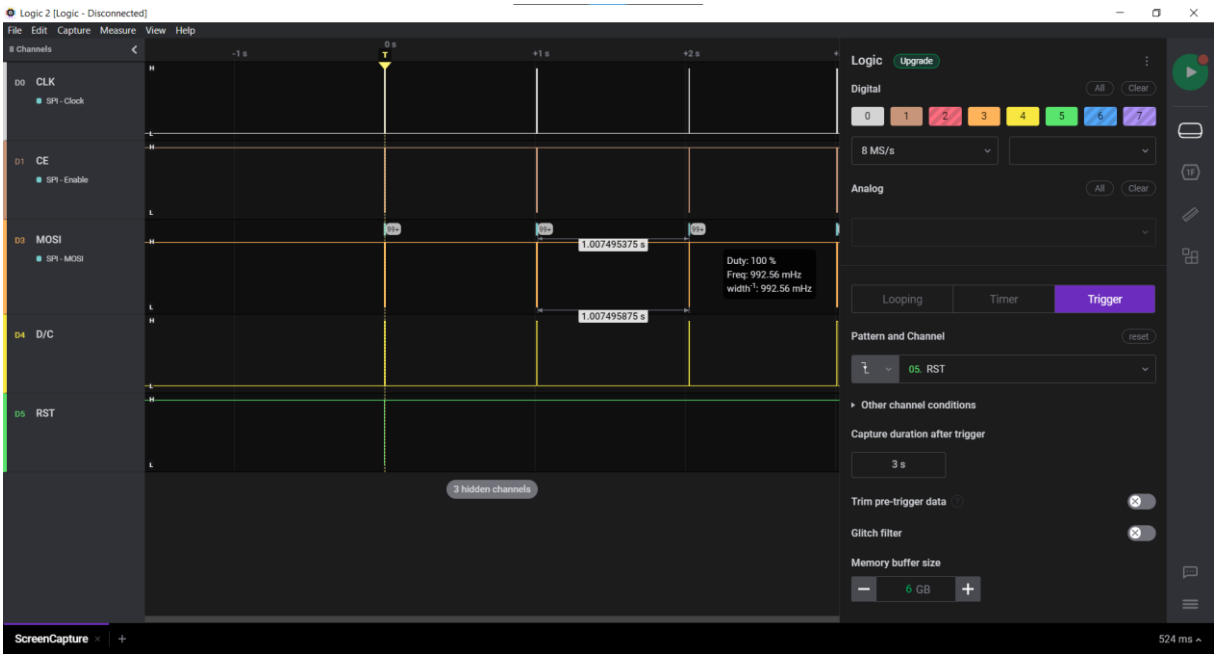
```
SCREEN_DATA     DCB 52, 40, 50, 51, 17, 26, 0x00, 13, 13, 13, 0x00, 0x00, 0x00, 0x00,\
                    52, 40, 50, 51, 18, 26, 0x00, 19, 16, 16, 0x00, 40, 90, 0x00,\
                    52, 40, 50, 51, 19, 26, 0x00, 22, 16, 16, 0x00, 40, 90, 0x00,\
                    38, 50, 37, 49, 00, 26, 0x00, 13, 13, 13, 0x00, 40, 90, 0x00,\
                    33, 45, 48, 44, 52, 26, 0x00, 13, 13, 13, 0x00, 0x00, 0x00, 0x00,\
                    44, 37, 36, 0, 0,   26, 0x00, 13, 0 , 0 , 0x00, 0x00, 0x00, 0x00
```

Each byte corresponds to the ID in Font6x8, which contains 96 characters.

Logic Analyzer Results



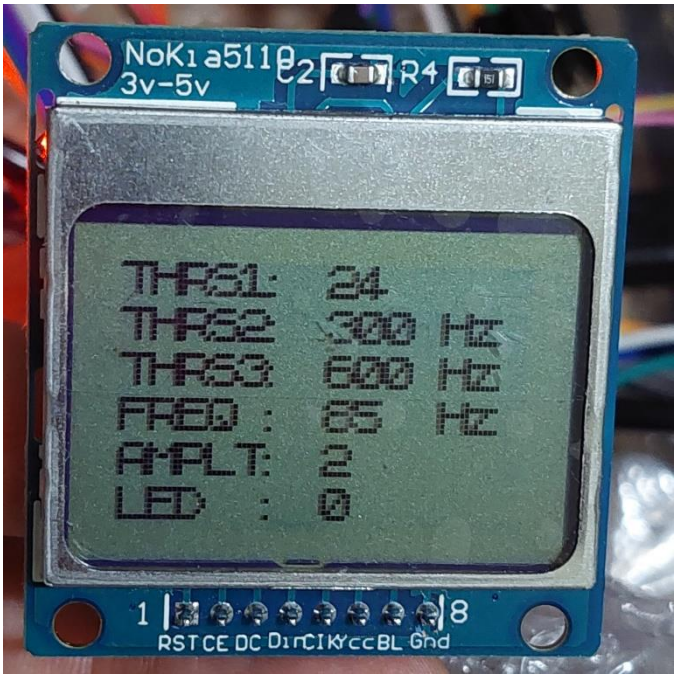
As Can be seen, SPI CLK is 4 Mhz. There is an active-low reset pulse before initializing. Then initializing commands are sent, followed by the screen data.



Here, the 1-second update can be seen.

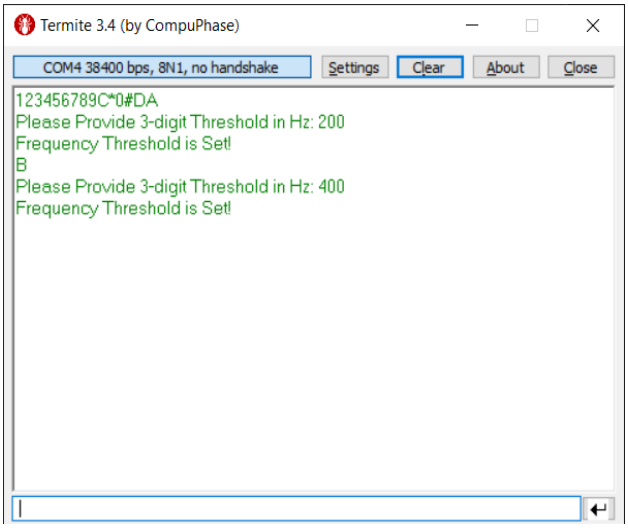
Implementation Results

LCD Screen



- THRS1: Amplitude Threshold
- THRS2: Low-Frequency Threshold
- THRS3: High-Frequency Threshold
- FREQ : Dominant Frequency
- AMPLT: Amplitude of Dominant Frequency
- LED : LED Status
  - 0 : LEDs OFF
  - 1 : 1st Level – RED LED
  - 2 : 2nd Level – GREEN LED
  - 3 : 3rd Level – BLUE LED

Keypad Interface



Before activating “A” and “B”, each key is printed as it is. After “A” is pressed, the user is informed to input 3-digit, each digit is printed to the serial terminal. After 3-digit is entered, the corresponding threshold is updated in the Screen.

The LEDs and Amplitude Threshold Potentiometer are implemented and verified that they are working correctly.

References

[1] Texas Instrument, "Tiva C Series TM4C123G LaunchPad Evaluation Kit User's Manual," Texas Instrument, [Online]. Available: https://www.ti.com/lit/ug/spmu296/spmu296.pdf#page=10.

[2] "Custom Fonts for Microcontrollers," [Online]. Available: https://jared.geek.nz/2014/jan/custom-fonts-for-microcontrollers.