Miniature Programmable Power Supply

EE 344: ELECTRONIC DESIGN LAB

Project Report

by

Abhishek M, Gouri S Dev, G Srilasya Sasidhar

(Roll Numbers: 200070003, 200070019, 200070022)

Group : TUES-33

Faculty Mentors

Prof. Joseph John, Prof. Shiladri Chakraborty

Department of Electrical Engineering

Indian Institute of Technology Bombay

April 2023

Contents

1 Introduction **3**

1.1 Project Goals . . . 3

2 Block Diagram **4**

2.1 Block Diagram explained: . . . 5

3 Schematic **6**

3.1 Component Selection Details . . . 7

4 PCB layout **8**

4.1 Top Layer . . . 9

4.2 Bottom Layer . . . 10

5 CAD Model **11**

6 Subsystem Test Results **12**

6.1 Boost Circuit . . . 12

6.1.1 Test Setup . . . 12

6.1.2 Methodology . . . 12

6.1.3 Result . . . 13

6.2 Digipot . . . 13

6.2.1 Test Circuit . . . 13

6.2.2 Methodology . . . 13

6.2.3 Results . . . 13

6.2.4 Code . . . 14

6.3 ADC . . . 15

6.3.1 Test Circuit . . . 15

6.3.2 Methodology . . . 15

6.3.3 Results . . . 15

6.3.4 Code . . . 15

7 Final Testing **17**

7.1 Code for the full system . . . 17

8 Components BOM **21**

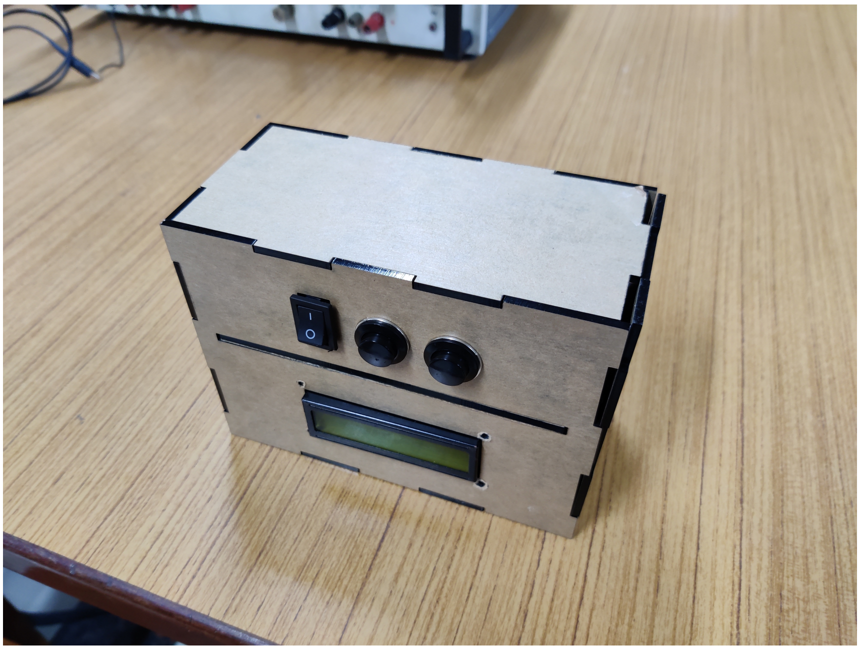
9 Major Problems Faced **21**

10 Future Work **22**

1 Introduction

It is quite difficult to transfer the benchtop power supply that is provided in the WEL lab when using it for different experiments. We lack a dependable power source that is compact, adaptable, and able to generate a range of voltages from a single input supply for many portable projects.

A power supply that can draw power from the AC mains using the standard C-type mobile charger (5V), and generate a stepped-up range of voltages using the charger's DC voltage could be quite useful in this scenario, and the aim of this project is to build one such miniature programmable power supply.



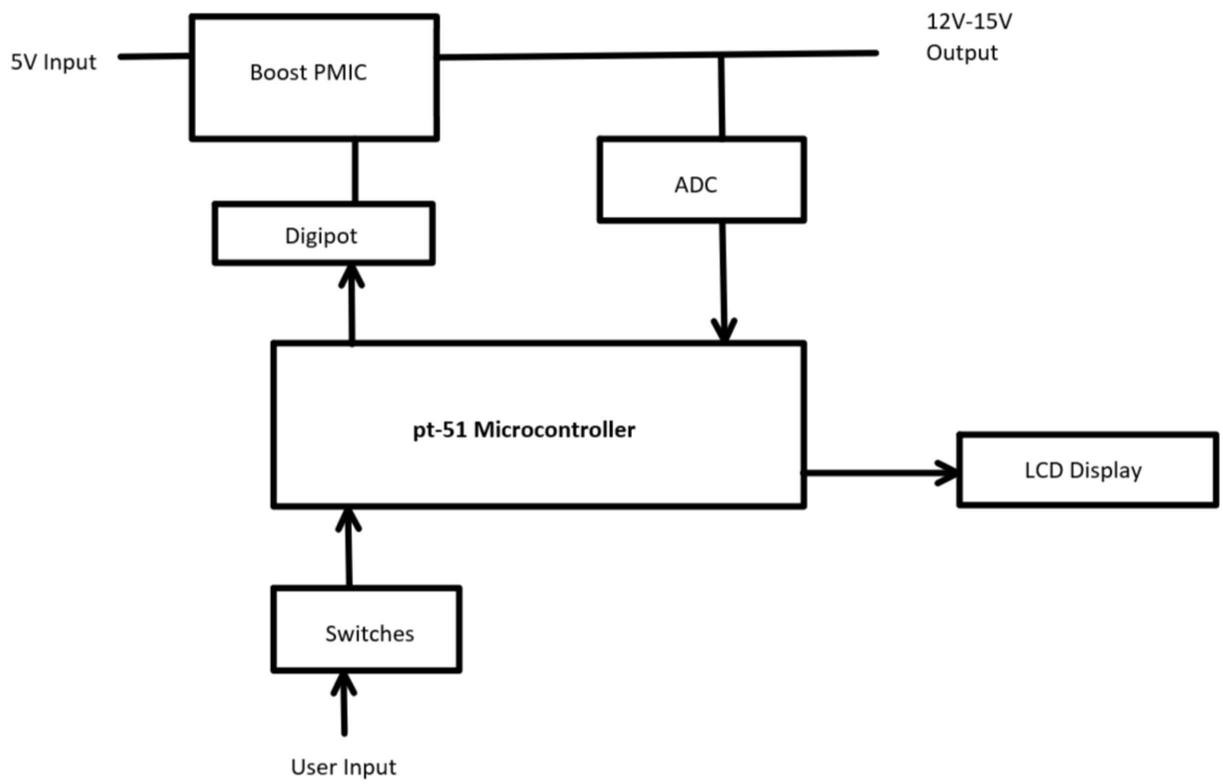
Miniature Programmable Power Supply

1.1 Project Goals

In this project, we have made a programmable compact power supply with the following specifications for input and output:
Input: 5V via USB-C of a regular mobile phone charger.
Output: 12V to 15V

Users can easily set the desired output voltage with the help of push buttons and a toggle switch. The output voltage can be conveniently plugged into a breadboard and will be immediately displayed on an LCD display

2 Block Diagram



Block Diagram

The miniature portable power supply is composed of the following:

- Input port for a 5V supply
- Buttons: Means to take input from the user.
- Boost Converter: The DC-DC converter will step up the input 5V to a voltage in the range of 12V to 15V depending upon the input given by the user through the buttons.
- Analog-to-Digital Converter: Converts the analog output voltage signal to a 10-bit digital value.
- LCD Display: A display to show the user the present voltage output.
- Microcontroller: It takes the input from the user (the required voltage value) and uses it to change the output voltage accordingly. It also takes the output of the ADC and displays the output voltage on the LCD.

2.1 Block Diagram explained:

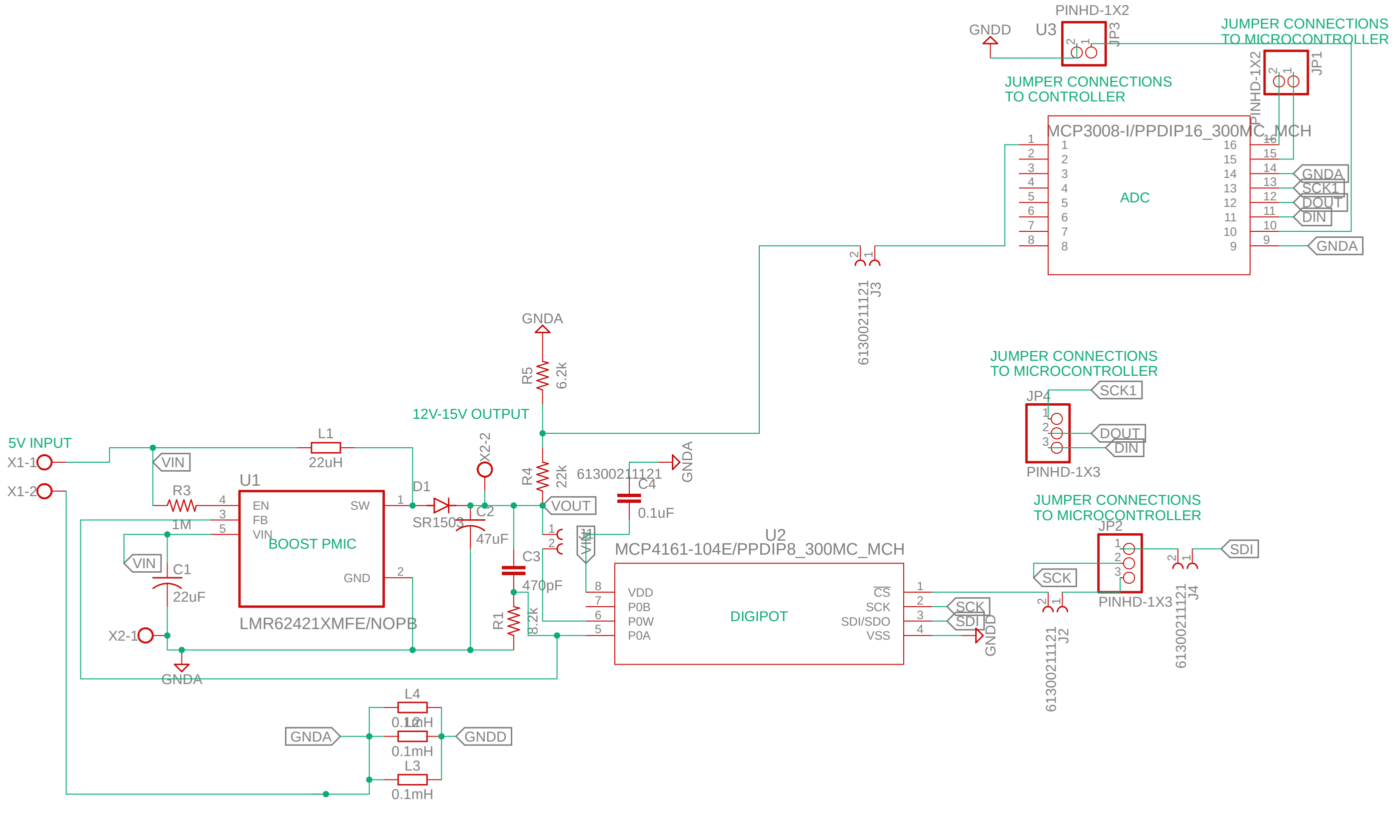
- The user sets the output voltage required using two push buttons and a toggle switch. The toggle switch is used to switch between coarse and fine voltage adjustment, while the push buttons are used to increase and decrease the output voltage.

- The boost PMIC receives a 5V supply from the USB-C charger. It then boosts the voltage and its output voltage is controlled by its feedback resistors. Since we need a variable V_{out}, we use a digipot as a variable resistor in place of one of the feedback resistors and thereby vary the value of V_{out} as per the input given by the user through the switches. $V_{out} = (\frac{R_{digipot}}{R_1} + 1)1.255$, where R_1 is the non-varying feedback resistance.

- To control the resistance of the digipot we have used the pt-51 microcontroller. Once the user inputs a value through the rocker switch and push buttons, the pt-51 microcontroller adjusts the input passed to the digipot, thereby altering its resistance accordingly.

- The resulting output voltage is then directed to a 10-bit ADC which converts the voltage to a digital value. This 10-bit digital value is then passed to the microcontroller, which displays it on the LCD in real time.

3 Schematic



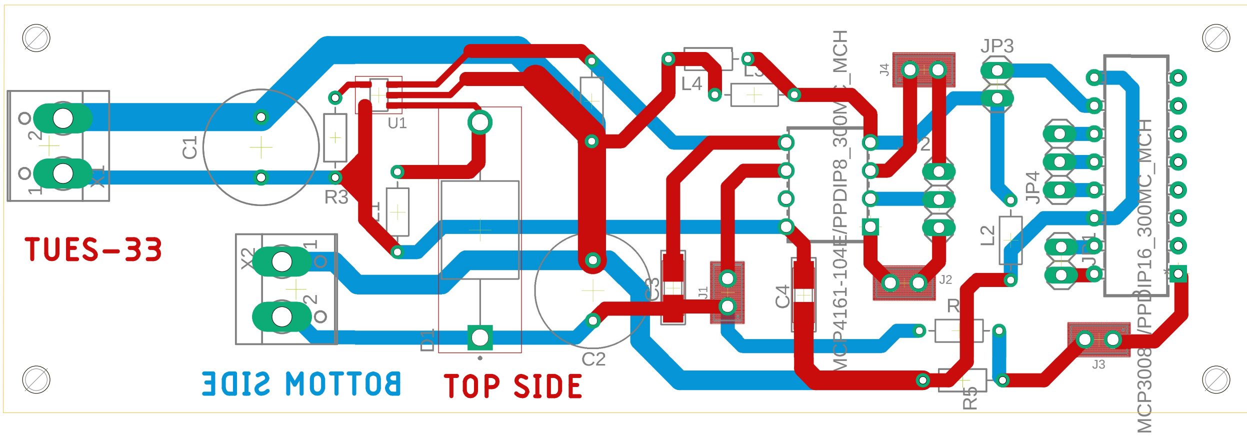
4/29/2023 12:30 AM C:\Users\USER\AppData\Local\Temp\Temp1_Gouri_S_Dev (1).zip\Gouri S Dev\PCB_v59_07_APR_2023.sch (Sheet: 1/1)

3.1 Component Selection Details

Following are the values of the passives in the boost circuit, chosen as per the datasheet:
$R_1 = 12k\Omega$
$R_3 = 1M\Omega$
$C_1 = 22uF$
$C_2 = 4.7uF$
$C_3 = 470pF$

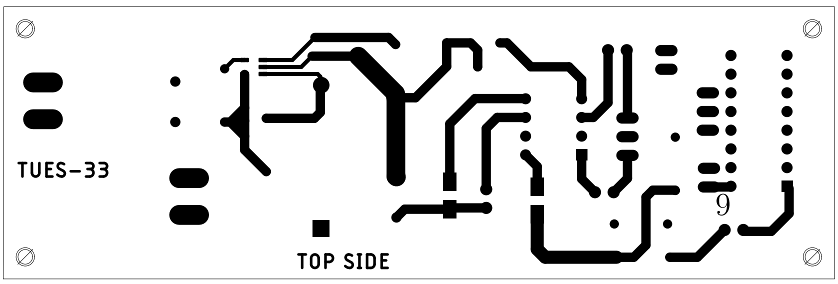
Calculation of Inductance for the boost circuit:
$f_{\mathrm{sw}} = 1.6MHz$
$\Delta i_L = 0.2 * i_L$ (limiting ripple current to 20 percentage)
$L_1 = \frac{V_{in} * D * T_s}{(2\Delta i_L)} = 16\ \mu$ H

4 PCB layout

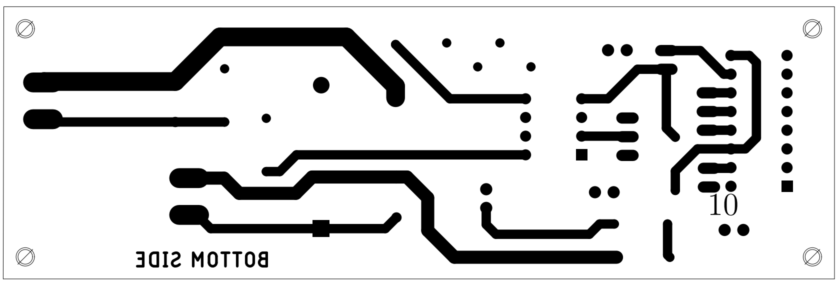


TUES-33
BOTTOM SIDE
TOP SIDE
U1
R3
C1
C2
L4
JP3
JP4
L2
R5
J4
J2
J3
J1
C4
MCP4161-104E/PPDIP8_300MC_MCH
MCP3008-I/PPDIP16_300MC_MCH

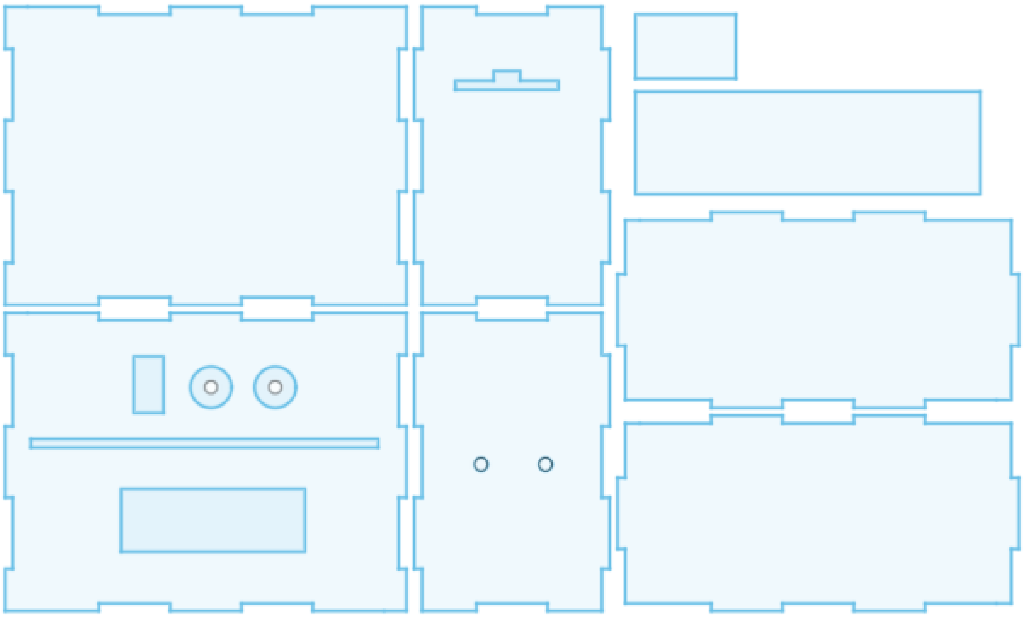
4.1 Top Layer



4.2 Bottom Layer



5 CAD Model

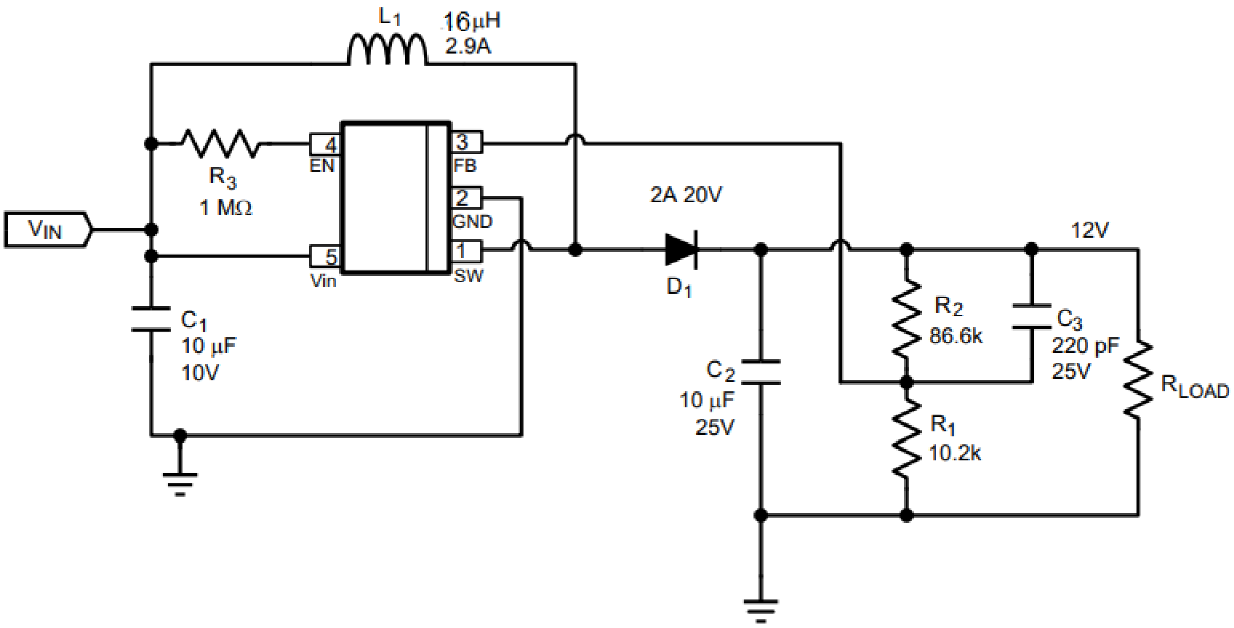


- The miniature programmable power supply will be enclosed in the box. The left bottom face has the cavities for the LCD display (larger rectangular cavity), and the switches. The extrusion will act as a ledge on which the PCB will be placed.
- The top middle face has an extrusion on which the USB-C connector will be placed and the USB-C charger will be connected to it through the cavity right above it.
- The middle bottom face has two holes through which the output wires can be taken and connected to the breadboard for use.

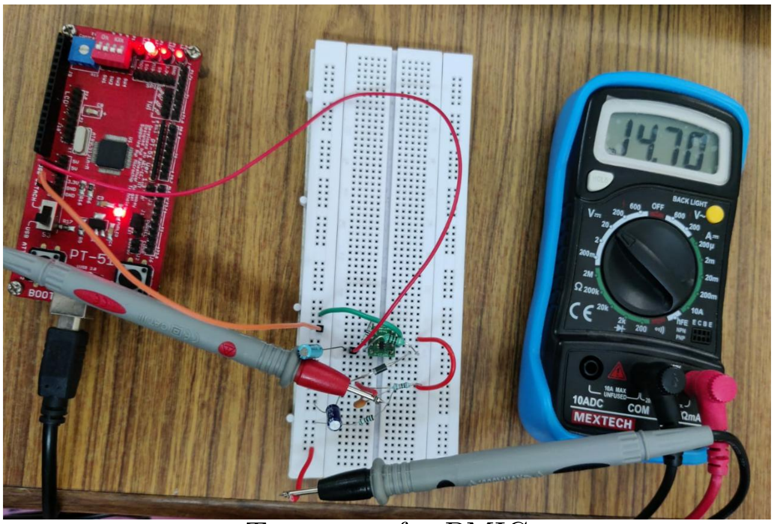
6 Subsystem Test Results

6.1 Boost Circuit

6.1.1 Test Setup



Boost PMIC test setup circuit diagram



Test setup for PMIC

6.1.2 Methodology

The boost circuit is made as in the diagram above where the value of V_{out} and R_2 are related as $R_2 = ((\frac{Vout}{1.255}) - 1)R_1$. To test its functioning we provided a supply of 5V and used a potentiometer in place of R_2 (shown in image). The potentiometer resistance was varied continuously and we observed the change in V_{out} to check whether it was varying according to the relation given above.

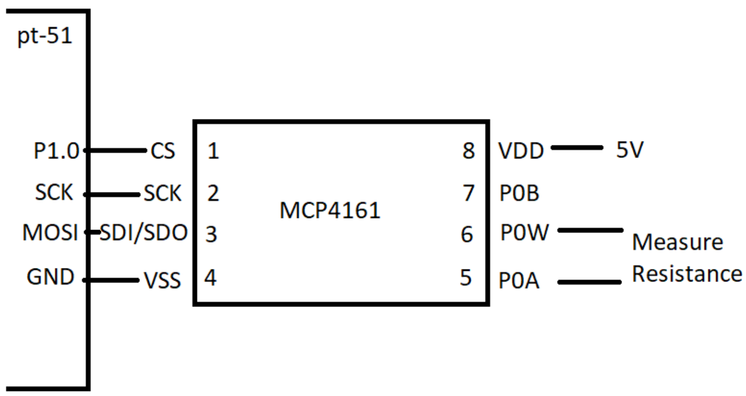
6.1.3 Result

The output voltage was varying as expected with change in R_2 and therefore we concluded that the PMIC testing is successful and that the boost subsystem is working.

Link: Click for Boost subsystem testing video.

6.2 Digipot

6.2.1 Test Circuit



Test connections for Digipot

6.2.2 Methodology

For testing the digipot, we make the connections as shown in the above circuit diagram. Here, the resistance between the wiper pin and one of the end terminals of the potentiometer has to be used as the feedback resistance in the boost subsystem to control its output voltage.
For testing, we have varied the input parameter given to the digipot through the toggle switch and push buttons to change its resistance and measured its resistance with a multimeter to see if it is changing as per our expectations.

6.2.3 Results

Through this code, we confirmed that the digipot was able to continuously update itself for a given range with the help of SPI communication with pt-51.

The digipot is working successfully as a variable resistor and can be used as a feedback resistor for the boost PMIC.

Link: Click for Digipot subsystem testing video.

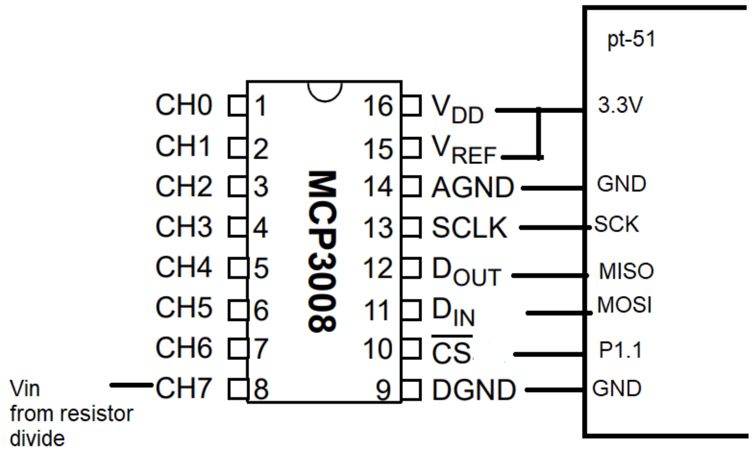
The change in the resistance between the wiper pin and one of the end terminals of the digipot according to the input given through switches can be seen here.

6.2.4 Code

```

/*********************************************************************
This is the test for the digipot MCP4161 using  the SPI already given
*********************************************************************/

#include <at89c5131.h>
#include "digipot.h"
#include "spi.h"
#include "lcd.h"

unsigned int resis;
unsigned int i;

void main(void)
{

  spi_init();        // initialize SPI interface
  digipot_init();    // initialize digipot

  for (i = 1; i <150; i = i+4){
    resis = i;
    digipot_write(resis);
    msdelay(100);
  }

}
```

6.3 ADC

6.3.1 Test Circuit



Test connections for ADC

6.3.2 Methodology

Since the microcontroller has to display the output voltage on LCD, we have to down convert the 12-15V output of the boost circuit to 0-3.3V using a resistor divider so as to give it as input to the ADC.
To test the ADC, we connect its CH7 channel to the output voltage signal. We then interface the ADC with the microcontroller using SPI. This code takes in the value of the CH7 channel and accurately converts it to its corresponding 10-bit representation. The microcontroller then displays this value on the LED.

6.3.3 Results

After running the code, we confirm that the test was successful and that the ADC is functioning as intended.

Link: Click for ADC subsystem testing video

6.3.4 Code

```
/*********************************************************
This is the code to interface the adc MCP3008 using SPI
*********************************************************/

#include <at89c5131.h>
```

```c
#include "lcd.h"   //Driver for interfacing lcd
#include "adc.h" //Driver for interfacing ADC ic MCP3008
#include "spi.h"

char adc_ip_data_ascii[6] = {0, 0, 0, 0, 0, '\0'};
code unsigned char display_msg1[] = "Volt.: ";
code unsigned char display_msg2[] = " mV";

void main(void)
{
  int j = 0;
  unsigned int adc_data = 0;

  spi_init();
  adc_init();
  lcd_init();

  while (1)
  {
    unsigned int x;
    // Read analog value from 7th channel of ADC Ic MCP3008
    // Converting received 10 bit value to mV (3.3*1000*i/p /1023)
    x = adc(7);
    adc_data = (unsigned int)(x * 3.2258)*5.05;

    // Display "Volt: " on first line of lcd
    lcd_cmd(0x80);
    lcd_write_string(display_msg1);

    // Converting integer to string of ascii
    int_to_string(adc_data, adc_ip_data_ascii);

    // Print analog sampled input on lcd in the format "XXXXX mV"
    lcd_write_string(adc_ip_data_ascii);
    lcd_write_string(display_msg2);

  }
}
```

7 Final Testing

All three subsystems were then integrated on a breadboard and tested together. The testing worked and voltages in the range of 12V-15V were obtained from a 5V supply.

We have performed load testing for 1.2k Ω and 10k Ω resistances and observed the appropriate current values. It was observed that the output voltage did not change with the change in the load resistance.

Link: Click for final open circuit testing video
Link: Click for load testing video

7.1 Code for the full system

```
/*********
This is the final code that integrates all the subsystems (digipot,
   adc, lcd, and switches to the PMIC)
********/

#include <at89c5131.h>
#include "digipot.h"
#include "spi.h"
#include "lcd.h"
#include "adc.h"

int resis;
int min_bit;
int max_bit;

// Setting the default, min and max resistance respectively
unsigned long int def_resistance = 500;
unsigned long int low_res = 100;
unsigned long int high_res = 35000;

int j = 0;
unsigned int adc_data = 0;

code unsigned char display_msg1[] = "Fine inc ";
code unsigned char display_msg2[] = "Fine dec ";
code unsigned char display_msg3[] = "Course inc ";
code unsigned char display_msg4[] = "Course dec ";
code unsigned char display_msg5[] = "Hit minimum ";
code unsigned char display_msg6[] = "Hit maximum ";

code unsigned char display_txt1[] = "Volt.: ";
code unsigned char display_txt2[] = " mV";

char data_ascii[6] = {0, 0, 0, 0, 0, '\0'};
char bit_ascii[6] = {0, 0, 0, 0, 0, '\0'};

//Specifying the pins for the input switches
```

```
sbit inc = P3^4;
sbit dec = P3^5;
sbit mode = P3^6;

void main(void){

  digipot_init();
  spi_init();
  lcd_init();
  adc_init();

  mode = 0;
  inc = 0;
  dec = 0;

  resis = def_resistance*256/100000;
  min_bit = low_res*256/100000;
  max_bit = high_res*256/100000;

  // Loop to update the value of the digipot and obtain the required
    voltage according to the inputs given
  while(1){
      //collecting the value from adc
      unsigned int x;
      x = adc(7);
      adc_data = (unsigned int)((x * 3.2258 * 4.95) - 0.13);

      msdelay(100);
      int_to_string(resis,bit_ascii);

      lcd_cmd(0x80) ;

    //Now checking for updates
    //first line prints the voltage
    //second line prints the resis bit

      if(mode == 1){//fine control

        if(inc == 1 && dec == 0){//fine increase
          resis += 1;
          lcd_cmd(0x01);
          lcd_write_string(display_txt1);
          int_to_string(adc_data, data_ascii);
          lcd_write_string(data_ascii);
          lcd_write_string(display_txt2);
          lcd_cmd(0xC0);
          lcd_write_string(display_msg1);
          lcd_write_string(bit_ascii);
          msdelay(100);
        }
        else if(inc == 0 && dec == 1){//fine decrease
          resis -= 1;
          lcd_cmd(0x01);
          lcd_write_string(display_txt1);
```

```
          int_to_string(adc_data, data_ascii);
          lcd_write_string(data_ascii);
          lcd_write_string(display_txt2);
          lcd_cmd(0xC0);
          lcd_write_string(display_msg2);
          lcd_write_string(bit_ascii);
          msdelay(100);
        }
      }
      else if(mode == 0) {//course control

        if(inc == 1 && dec == 0){//course increase
          resis += 5;
          lcd_cmd(0x01);
          lcd_write_string(display_txt1);
          int_to_string(adc_data, data_ascii);
          lcd_write_string(data_ascii);
          lcd_write_string(display_txt2);
          lcd_cmd(0xC0);
          lcd_write_string(display_msg3);
          lcd_write_string(bit_ascii);
          msdelay(100);
        }
        else if(inc == 0 && dec == 1){//course decrease
          resis -= 5;
          lcd_cmd(0x01);
          lcd_write_string(display_txt1);
          int_to_string(adc_data, data_ascii);
          lcd_write_string(data_ascii);
          lcd_write_string(display_txt2);
          lcd_cmd(0xC0);
          lcd_write_string(display_msg4);
          lcd_write_string(bit_ascii);
          msdelay(100);
        }
      }

      if(resis < min_bit){ // Checking for the min limit
        resis = min_bit;
        lcd_cmd(0x01);
        lcd_write_string(display_txt1);
        int_to_string(adc_data, data_ascii);
        lcd_write_string(data_ascii);
        lcd_write_string(display_txt2);
        lcd_cmd(0xC0);
        lcd_write_string(display_msg5);
        lcd_write_string(bit_ascii);
        msdelay(100);
      }
      else if(resis > max_bit){ // Checking for the max limit
        resis = max_bit;
        lcd_cmd(0x01);
        lcd_write_string(display_txt1);
        int_to_string(adc_data, data_ascii);
```

```
        lcd_write_string(data_ascii);
        lcd_write_string(display_txt2);
        lcd_cmd(0xC0);
        lcd_write_string(display_msg6);
        lcd_write_string(bit_ascii);
        msdelay(100);
      }
    digipot_write(resis); // Writing the value of resis to digipot
  }
}
```

8 Components BOM

Table 1: Components Used

S. No	Component	Specification	Quantity	Price (INR p.u.)
1	Microcontroller	pt-51 (WEL Lab)	1	250
2	Digipot	MCP4161	1	120
3	PMIC (boost)	LMR62421	1	130
4	PMIC (boost)	TPS61040DBVR (backup)	1	134
5	Schottky Diode	SR150	1	2.4
6	USB Type C	Breakout PcB Board Pins	1	100
7	Push Button Switch	R13-507	2	21
8	Rocker Switch	SPST	1	16.4
9	Resistance	8.2 kΩ, 1 MΩ, 6.2 kΩ, 22 kΩ	4	WEL
10	Inductor	16uH	1	WEL
11	Capacitor	22 uF, 4.7 uF, 470 pF, 0.1 uF	4	WEL
12	ADC	MCP3008	1	198

9 Major Problems Faced

S No.	Problems	Solution
1	The previous ADC, MAX1118 used 3 wire SPI and hence required additional pins for implementation. We weren't able to achieve the exact timing diagram for CNVST, clock, data input which lead to improper transmission of data.	We switched to MCP3008, a 10-bit ADC available in WEL, which supported direct SPI connection
2	The previous digipot, MAX5401 had a similar SPI interfacing issue. Hence went into wiper default mode and the digipot no longer updated the wiper positions.	We switched to MCP4161, 100k digipot available in lab and were able to successfully integrate that.
3	For the current digipot MCP4161, the maximum voltage allowed across its pins was 3.6 V; otherwise it would go into WiperLock protection mode and disable the SPI communication.	To ensure this did not happen in our range of output voltages, we added a resistance in series with the digipot to keep the digipot resistance smaller and therefore, reduce the voltage across it.

10 Future Work

Currently, we have successfully implemented the project on a breadboard. The efficiency would be higher when implemented on a PCB because of lesser wiring, loose connections and other such errors. The power supply could be improved by adding voltage regulators based on the end application, to avoid any abrupt increase or decrease in voltage which could damage components. Fuses can also be used as a mitigation strategy. More extensive load testing can also be performed to see if the miniature programmable power supply is capable of giving load currents as high as 0.2 A.