



Electronic Design Lab

Milestone -2

Schematic and preliminary analysis review

Mohit Jindal
20D070052

Vinayak Goyal
20D070088

Vrinda Goel
20D070090

March 27, 2023

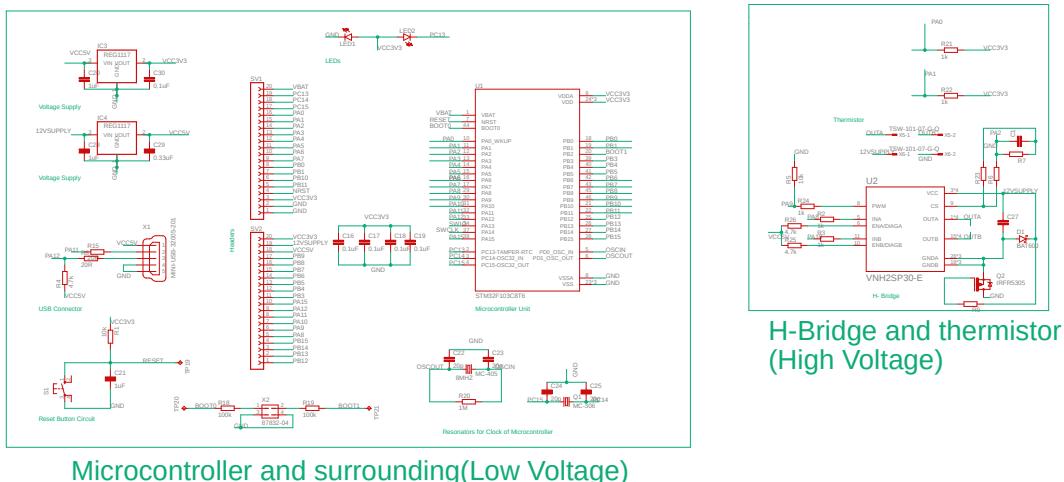
1 PCB Schematics

Given Below are the PCB Schematics for our PCR Machine. These include the Schematic for Micro-controllers, H-Bridge, Voltage Regulator, Power Supply, Sensors Connection, etc.

Given Below is the pin allotment that we have made:

- **ADCs:** We have used 2 inbuilt ADCs for 2 thermistors, one for the temperature of peltier and other for the temperature of heated lid. Pins PA0 and PA1 are used as input pins for these ADCs. We could have used the pins PA0-PA7 and PB0-PB1 as analog pins for this purpose.
- **LCD:** We have used GPIO Pins(General Purpose Input Output) for this. There are a total of 37 GPIO pins in our micro-controller. We have allotted pins A6-A7, B0-B1 and B10-B11 as the output pins for LCD Display. This will basically be the temperature display of ADC output.
- **Fan:** A PWM Pin is allocated to control whether the Fan will be ON/OFF. It will be ON when we want to cool the system otherwise it will be OFF. The Pin allocated for this is PB6 which is PWM signal.
- **H-Bridge:** We have used PWM Pins for this. There are 15 PWM Pins in our micro-controller which are PA0-PA3, PA6-PA10, PB0-PB1 and PB6-PB9. In our schematic we have used PA8-PA10 PWM Pins as input to the H-Bridge. The output of H-Bridge is given to the peltier through OUTA-OUTB and a CS(Current Sensor) signal is sent to micro-controller through the pin PA2.

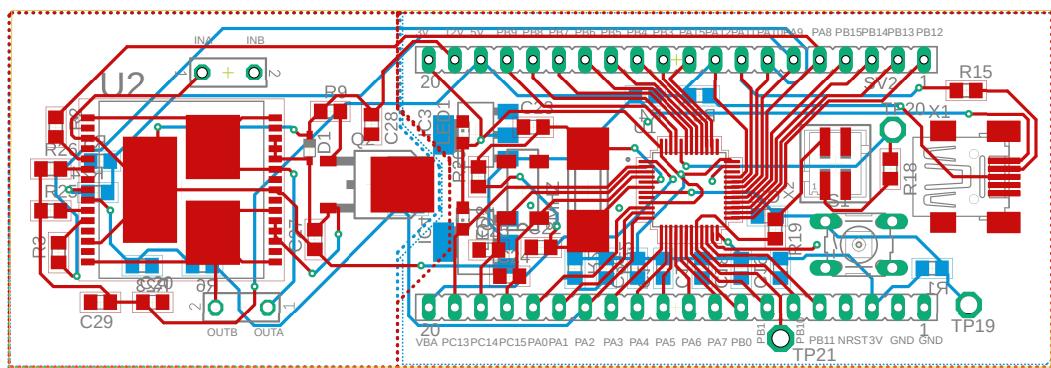
We have made some changes in our design since the last checkpoint. We decided to integrate the PCBs of the microcontroller and H-bridge to form a single PCB. The modified PCB and the first completed layout of the PCB are shown below:



Microcontroller and surrounding(Low Voltage)

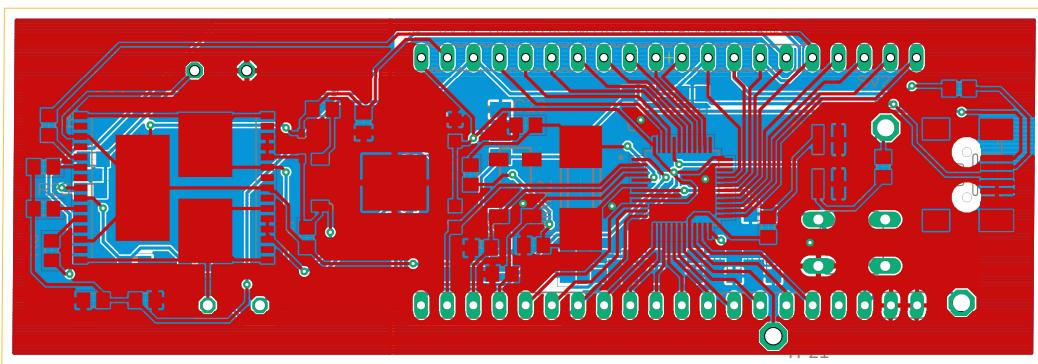
27-03-2023 02:32 f=0.65 C:\Users\Vinay\OneDrive\Documents\EAGLE\projects\EDL\LQFP48-7x7mm.pac.sch (Sheet: 1/1)

Figure 1: Schematic of our PCB



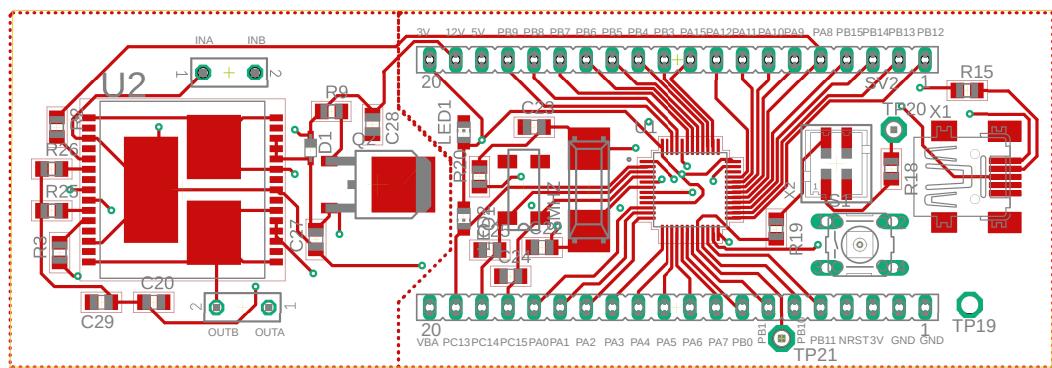
27-03-2023 03:08 f=2.60 C:\Users\Vinay\OneDrive\Documents\EAGLE\projects\EDL\LQFP48-7x7mm.pac.brd

Figure 2: PCB layout consisting of both layer components



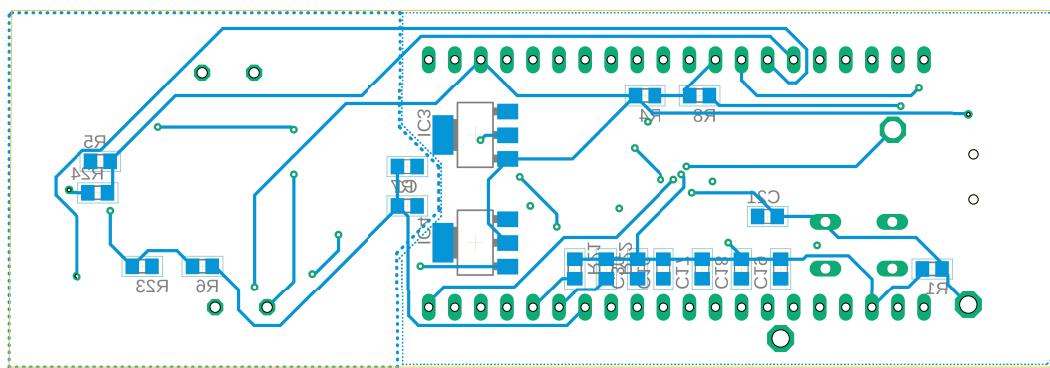
27-03-2023 03:08 f=2.60 C:\Users\Vinay\OneDrive\Documents\EAGLE\projects\EDL\lQFP48-7x7mm.pac.brd

Figure 3: PCB layout consisting of both layer components along with ground plane



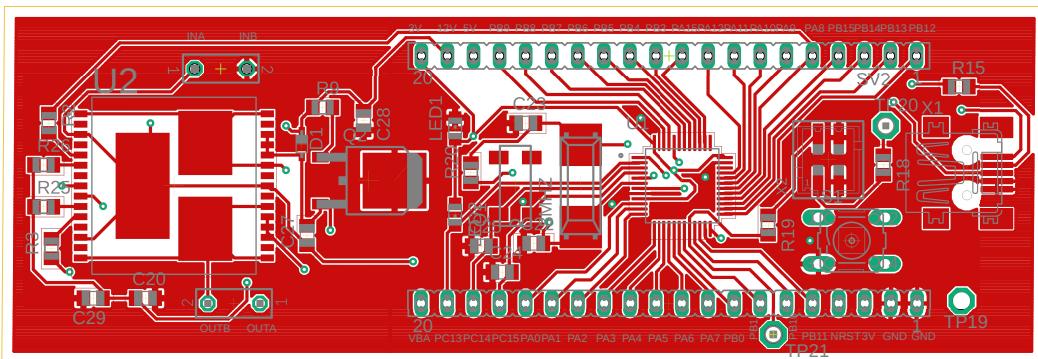
27-03-2023 03:07 f=2.60 C:\Users\Vinay\OneDrive\Documents\EAGLE\projects\EDL\LQFP48-7x7mm.pac.brd

Figure 4: PCB layout of the top layer



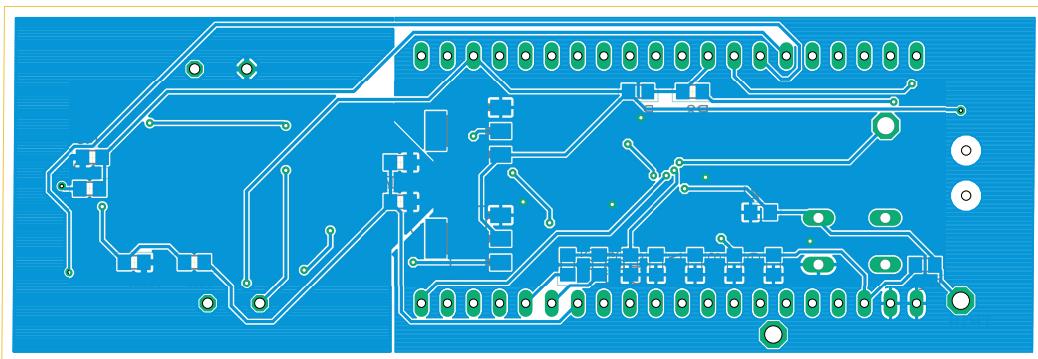
27-03-2023 03:03 f=2.60 C:\Users\Vinay\OneDrive\Documents\EAGLE\projects\EDL\LQFP48-7x7mm.pac.brd

Figure 5: PCB layout of the bottom layer



27-03-2023 03:09 f=2.60 C:\Users\Vinay\OneDrive\Documents\EAGLE\projects\EDL\LQFP48-7x7mm.pac.brd

Figure 6: PCB layout of top layer with ground plane



27-03-2023 02:05 f=2.60 C:\Users\Vinay\OneDrive\Documents\EAGLE\projects\EDL\LQFP48-7x7mm.pac.brd

Figure 7: PCB layout of bottom layer with ground plane

2 Justification for component selection

2.1 Peltier Cell

Name- TEC1-12712.

Specifications - Qmax = 144W, Imax=12A

Peltier with the above specifications was chosen because the one used in WELPCR had Imax=12A and Qmax=120W, so we decided to use Peltier with Qmax=144W so that samples can heat fast as it will have a high ramp rate. Other options are available, but their power rating is around 180W, and our supply rating is also 180W, so they might not be compatible as there are other components where this power will be distributed.

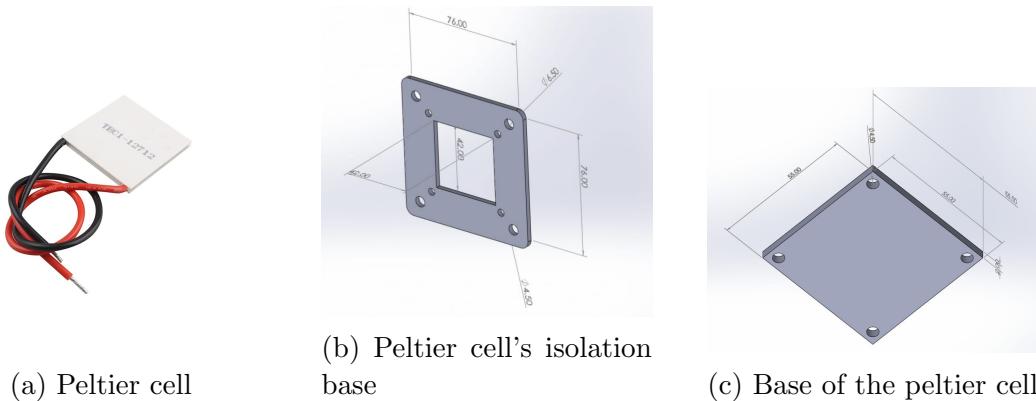


Figure 8: The Peltier cell: used for heating and cooling of the block

2.2 Microcontroller

Name- STM32F103C8T6

This microcontroller satisfies our I/O requirements and was used in WELPCR also. It has 2 inbuilt 12 bit ADC's which will be helpful in reading the thermistor. It has 37 GPIO (General Purpose Input Output Pins) which will be sufficient for our requirement. Also the STM32F103C8T6 is widely used in various embedded systems applications such as robotics, industrial control, and home automation, among others.

2.3 H-Bridge

VNH2SP30 Motor Driver

It has an inbuilt H bridge to drive Peltier. The voltage range is between 5.5 to 16 Volts which satisfies our requirement. It is compatible with our microcontroller,

and it can be controlled with PWM signals to control the speed and direction of the motor. It also has thermal shutdown and overcurrent protection.

Reference: <https://www.instructables.com/Monster-Motor-Shield-VNH2SP30/>

2.4 Power Supply

Specifications - Qmax = 180W, I_{max} = 15A

The Power Supply is appropriate for our component selection that is to drive the Peltier of 12A. Another option was to select a power supply of 12V 10A, but it won't be able to drive our Peltier. SMPS power supplies can provide precise regulation of the output voltage and current, which makes them suitable for use in sensitive electronic devices that require stable and clean power like our PCR Machine.

2.5 Fan

Name- Silverstone KR02

We plan to use the Silverstone KR02 fan as it has an rpm in the range of 800-2800, air flow rate 15.12 56.1 CFM which is better compared to Antec A30 which was used in WELPCR. Also, the dimensions are 97mm (W) x 125mm (H) x 51mm (D) compared to 125mm x 73mm x 86mm of Antec A30. This fan will help in the fast cooling to make the ramp rate better.



Figure 9: Silverstone KR02

2.6 Main Heating Block

Copper

Copper is used for the main heating block instead of aluminium because it has a lower specific heat capacity (0.385) as compared to aluminium (0.89) thus will result in a faster heat transfer from the peltier to the wells which will give us a higher ramp rate. We have also reduced the size of the block to reduce the thermal

mass of copper. The original block was of size 40x40 which we have reduced to make it 24x24.

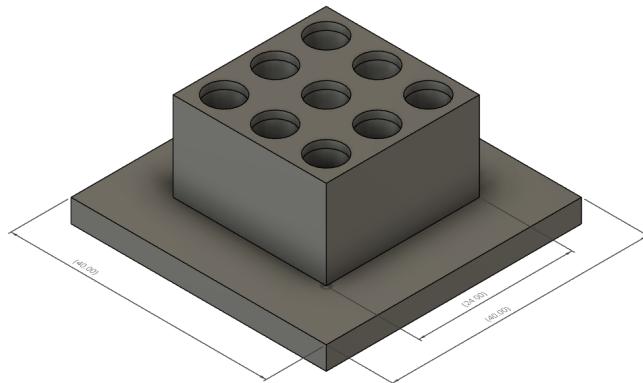


Figure 10: The main heating block created on autodesk fusion 360

3 Principle of operation of each subsystem

3.1 Heating Block

The heating block is the component that holds the DNA sample and is responsible for heating the sample to the desired temperature. It is typically made of a thermally conductive material, such as aluminum, to ensure efficient heat transfer. But we use copper because it has a relatively higher thermal conductivity than aluminium. The heat is provided by the Peltier cell to the heating block.

3.2 Cooling System

The cooling system cools the DNA sample to the desired temperature. It can be a fan-based or refrigeration-based cooling system. We are using a fan-based system that will cool the DNA sample after the denaturation process in which the temperature rises to around 96. The peltier cell can act as a heat absorber if the direction of current is reversed in it. Thus, peltier is used for cooling of the main heating block.

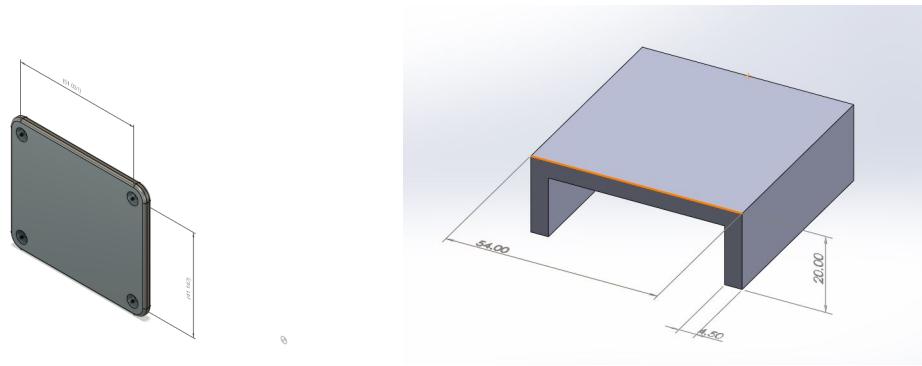
3.3 Temperature Controller

The temperature controller measures the DNA sample's temperature and adjust the heating and cooling elements accordingly. It typically uses thermistors or

thermocouples to measure the temperature and can be programmed to control the temperature profile for the PCR process. Here we will be using a thermistor (Model : 20D-9 NTC) to measure the temperature and then use the inbuilt ADC on the Micro-controller to control it. The sensors provide feedback to the heating and cooling system to ensure accurate temperature control during the PCR process.

3.4 Heated Lid

We would have a heated lid to cover the heating block to minimize condensation of the samples at the top of the sample tube and ensure that there is no condensation of the sample on the upper side of the eppendorf tube. This heated lid is kept at a constant temperature of 100. Nichrome wire will be used in the heated lid for this purpose. We plan to follow the design of WELPCR for heated lid.



(a) Heating lid

(b) Cover for heated lid

Figure 11: The heated lid model made on fusion 360

3.5 Control Unit

The thermal cycler is controlled by a computer interface or control unit, which is used to program the temperature cycles and monitor the progress of the reaction. The control system is responsible for programming the thermal cycler, controlling the heating and cooling system, and monitoring the temperature of the reaction mixture. The control system uses a micro-controller to execute the PCR protocol and provide feedback to the user.

3.6 Power Supply

A power supply is used to power the PCR machine's various components, including the micro-controller, heating element (Peltier Cell), and cooling element (Fan). It would be connected to the micro-controller for controlling the voltage and current. H-Bridge is used to reverse the direction of current in order to decide whether we want to heat or cool the sample.

3.7 H-Bridge

The H-Bridge is used to switch the current through the peltier that is in thermal contact with the sample. By switching the current through the heating element on and off in a precise manner, the temperature of the sample can be raised or lowered rapidly and accurately to the desired temperature for the various steps of the PCR process.

3.8 LCD Display

The results of the PCR Machine will be shown on an LCD screen. The results include the running time along with the controllability of time of reaction. The graphs for the current ongoing PCR machine will also be displayed and we can input the number of cycles for which we want to perform the PCR reaction.

4 Preliminary analysis

The Copper Block has been given for manufacturing post which the following experiment will be conducted:

- Aim: To check the copper block prepared.
- Method: The copper block prepared by us will be installed on the already existing WELPCR. Then the PCR process will be started on the machine to check the ramp rate obtained. The ramp rate will be observed from the LCD.
- Results: After seeing the ramp rate, it will be decided whether to use the formed copper block for further use or not.
- Next Step: If the desired ramp rate is not observed then other design of blocks like trapezium shaped block to have max heat transfer.

5 Mechanical Designs

One of the major components of mechanical designing was to make an appropriate heating block so as to minimize the thermal mass and maximize the heat transfer. So here we greatly reduced the size of the thermal block and reduced the gap between samples for faster heat transfer and lower loss of heat.

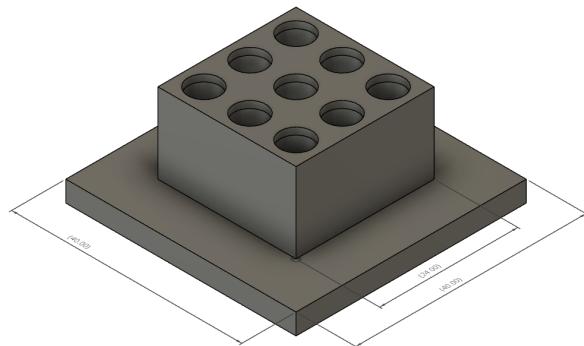


Figure 12: The main heating block created on autodesk fusion 360

The depth of the samples were calculated using the tube dimensions and it's mechanical design. Help was taken from the RA and TA for this design and dimension calculation.

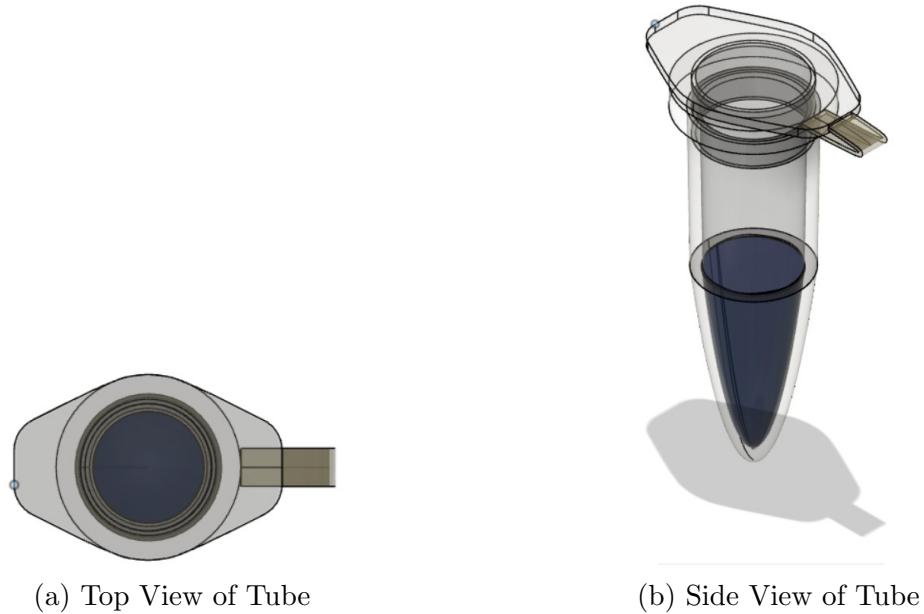
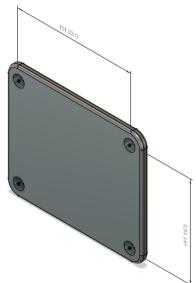


Figure 13: Tube for putting samples

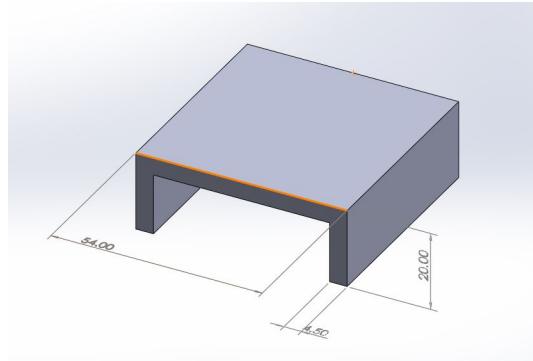
Another component that we have is the heated lid at the top of the Peltier cell which will be maintained at a constant temperature of around 100 to prevent condensation and this will be made of aluminium as given below :

We also need to design a **base** for Peltier on which it can rest. It's a 3-layer system which will be as shown below:

Below this plate, we have a **Peltier cover** which is around the same size and below which, again, a **metal plate** of almost same size as Peltier is used the design for which is given as :

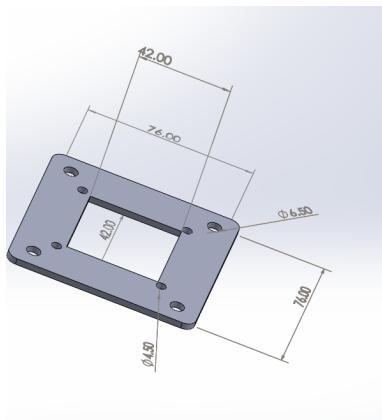


(a) Heating lid

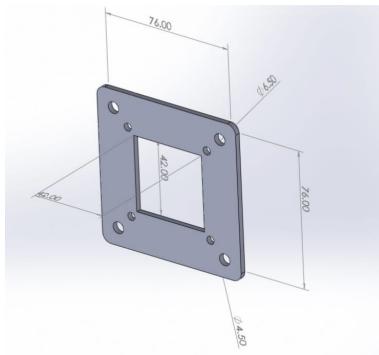


(b) Cover for heated lid

Figure 14: The heated lid model made on fusion 360



(a) Base Top View



(b) Base Side View

Figure 15: The heated lid model made on fusion 360

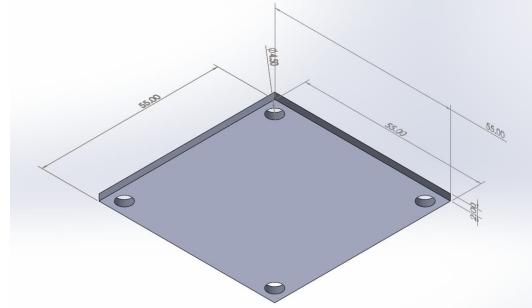


Figure 16: The main heating block created on autodesk fusion 360

6 CAD Design

Following are the CAD designs for the laser cutting of the outer casing of PCR Machine.

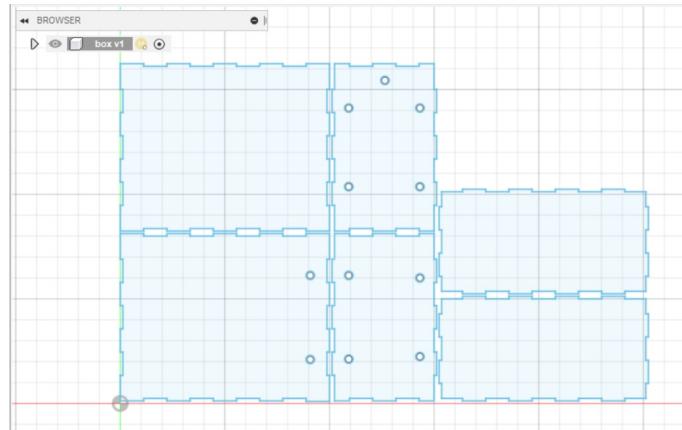


Figure 17: 2D design of outer casing

7 Testing of Sub-Circuits

7.0.1 Voltage Regulator(12V to 5V)

We did the testing for the Voltage Regulator(LM7805C), which will convert the input voltage of 12V to 5V to be supplied to the USB, and it seemed to work fine. Here is the snapshot of the circuit.

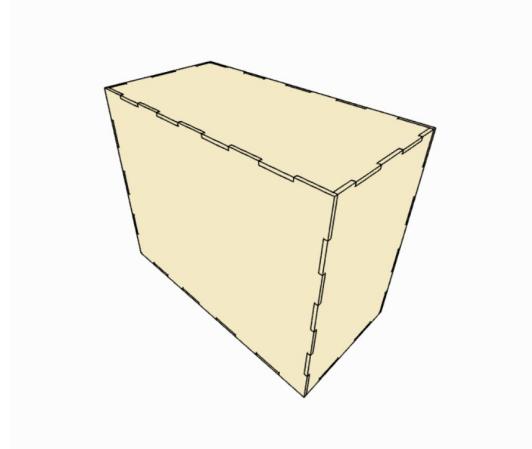


Figure 18: 3D design of outer casing

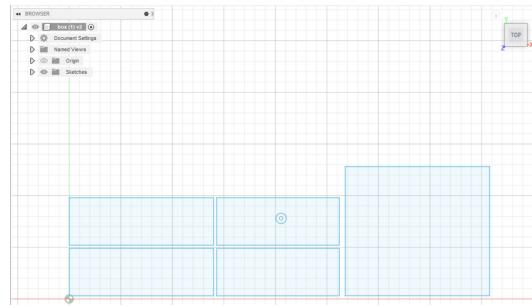


Figure 19: 2D design of lid cover

7.0.2 Voltage Regulator 5V to 3.3V

Below is the circuit for the Voltage regulator that takes 5V input and gives 3.3V output to be used by the microcontroller.

8 Programming

8.0.1 Code for interfacing of Thermistor with STM32F103C8T6

We have written the code for interfacing the thermistor with the microcontroller to sense temperature from Peltier and send the signal to the microcontroller and display the temperature on LCD. Here is the screenshot of the above-mentioned Program

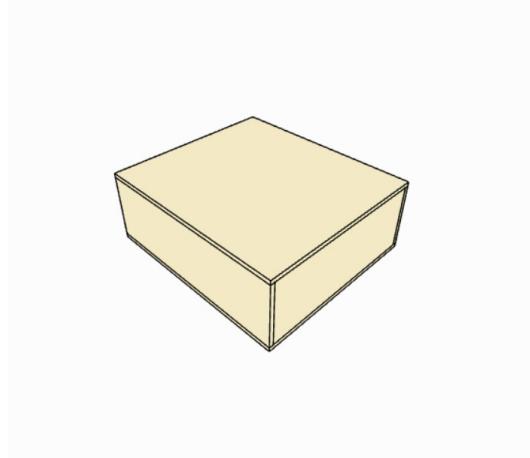


Figure 20: 3D design of lid casing

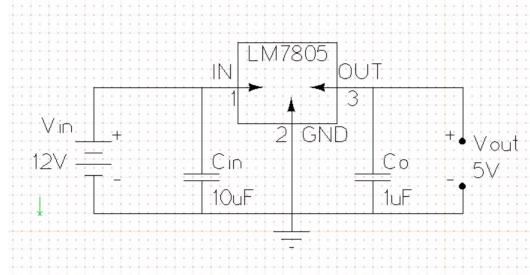


Figure 21: Circuit used for Testing Voltage Regulator(12V to 5V)

9 Risk mitigation/contingency

9.1 Ramp rate not achieved

- **Anticipated problem:** The copper block has better thermal conductivity than aluminum but has more thermal mass, so it may require more power to achieve the required ramp rate.
- **Likelihood:** High
- **Mitigation strategy:** We plan to test it on WELPCR, and if the ramp rate is not achieved, then we will try using an aluminium block or with another design for the copper block.

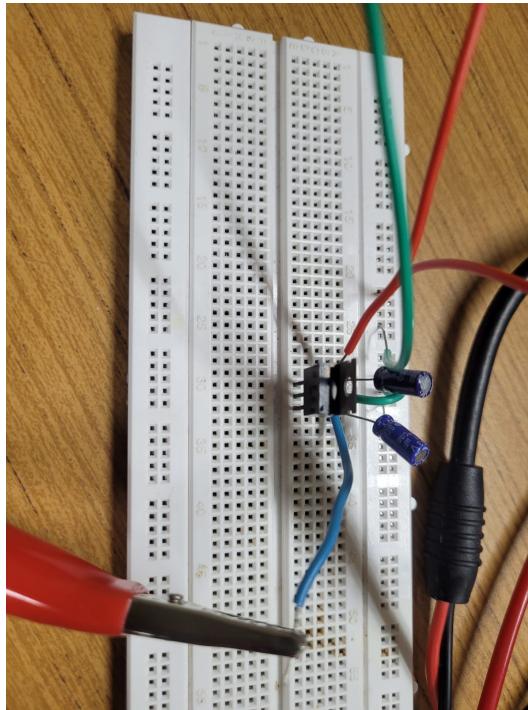


Figure 22: Circuit used for Testing Voltage Regulator

9.2 Possibility of short circuit and overcurrent

- **Anticipated problem:** Our circuit uses high currents and the microcontroller has a maximum current rating of 150mA. This may cause damage to components.
- **Likelihood:** High
- **Mitigation strategy:** Creating overcurrent and overvoltage detection circuits and applying proper spacing in PCB.

9.3 Heating of conducting wires

- **Anticipated problem:** High current and heating block may cause the melting of wires or damage to components.
- **Likelihood:** Medium
- **Mitigation strategy:** We plan to cover this during the packaging of the circuit.

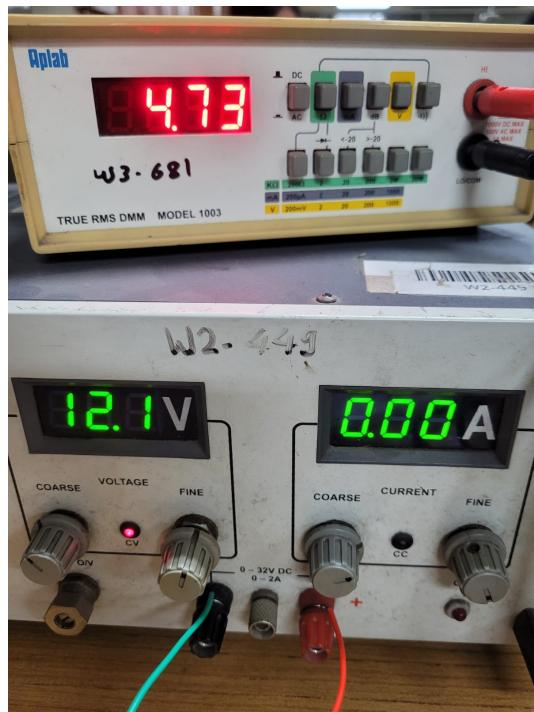


Figure 23: Output Voltage from Voltage Regulator

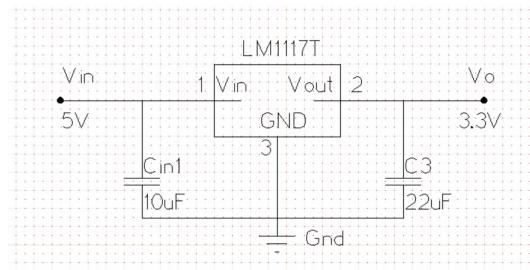


Figure 24: Voltage Regulator(5V to3.3V

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer** (left sidebar):
 - workspace_1.12.0 - thermistor2/Core/Src/main.c - STM32CubeIDE
 - File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
 - Project Explorer X
 - Drivers, Debug, STM32F103C8T6_FLASH.ld, Thermistor1.loc, Includes, Core, Inc, AVR, binary.h, core_build_options.h, core_pins.h, LiquidCrystal_I2C.h, main.h, pins_arduino.h, stm32f1xx_hal_conf.h, stm32f1xx_it.h, wiring.h, WProgram.h, Src, main.c, stm32f1xx_hal.msp.c, stm32f1xx_it.c, Stream.h, syscalls.c, systemc, system_stm32f1xx.c, Wire.h, Startup.
- Code Editor (right panel)**: The code editor displays the main.c file content, which includes header files, defines, and a setup() function. The code is as follows:

```
#include <stm32f1xx_hal.h>
#include <main.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

#define VSupply 3.3 //power supply voltage (3.3 V rail) -STM32 ADC pin is NOT 5 V tolerant
#define RNTC 9840 //10K resistor measured resistance in Ohms
#define B_param 3700 //B-coefficient of the thermistor
#define T0 298.15; //25°C in Kelvin
#define Temp_K; //Temperature measured by the thermistor (Kelvin)
#define Temp_C; //Temperature measured by the thermistor (Celsius)
const int VoutPin = PA0; //ADC pin of STM32
const int I2C_SDA = 10; //I2C SDA pin
const int I2C_SCL = 11; //I2C SCL pin

void setup()
{
    Serial.begin(115200); //Starting serial
    //-----LCD-----
    lcd.begin();
    lcd.setCursor(0,0); //Defining positon to write from first row,first column .
}
```

Below the code editor, there are tabs for Problems, Tasks, Console, Properties, and CDT Build Console [thermistor2].

Figure 25: Code for Thermistor

```
workspace_1.12.0 - thermistor2/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer X
  workspace_1.12.0
    +-- main.c
    +-- main.h
    +-- LiquidCrystal_I2C.h
    +-- wiring.h
    +-- pgmspace.h
    +-- Wire.h
    +-- Stream.h
    +-- main.c
      +-- main()
        +-- lcd.setCursor(0,0); //Defining position to write from first row,first column .
        +-- lcd.print("NTC Thermometer");
        +-- lcd.print("Hello,World!");
        +-- lcd.print("Demonstration"); //You can write 16 Characters per line .
        +-- delay(3000); //wait 3 sec
        +-- PrintCDC();
        +-- -----
        +-- pinMode(VoutPin, INPUT_ANALOG);
        +-- timeNow_1 = millis(); //noting down start time
        +-- void loop()
        +-- {
        +--   //time condition
        +--   timeNow_2 = millis(); //update time
        +--   if(timeNow_2 - timeNow_1 > UpdateInterval) //500 millis elapsed
        +--   {
        +--     UpdateLCD();
        +--     timeNow_1 = millis(); //update time. this resets the update interval
        +--   }
        +--   ConvertToTemperature();
        +-- }
        +-- void ConvertToTemperature()
        +-- {
        +--   Vout = analogRead(VoutPin)* (3.3/4095); //4095 is 12 bit resolution of the blue pill
        +--   //For Arduino users: (5.0 / 1023)
        +--   Serial.print("Vout: ");
        +--   Serial.println(Vout);
        +-- }
    +-- main.h
    +-- LiquidCrystal_I2C.h
    +-- wiring.h
    +-- pgmspace.h
    +-- Wire.h
    +-- Stream.h
    +-- syscalls.c
    +-- system.c
    +-- system.h
    +-- system_stm32f103c.h
    +-- STM32F103C8T6_FLASH.ld
    +-- Thermistor1.ioc
  STM32F103C8T6_FLASH.ld
  Thermistor1.ioc
  thermistor2
    +-- Binaries
    +-- Includes
    +-- Core
      +-- Inc
        +-- avr
        +-- binary.h
        +-- core_build_options.h
        +-- core_pins.h
        +-- LiquidCrystal_I2C.h
        +-- main.h
        +-- pins_arduino.h
        +-- stm32f1xx_hal_conf.h
        +-- stm32f1xx_it.h
        +-- wiring.h
        +-- WProgram.h
    +-- Src
      +-- main.c
      +-- main.h
      +-- stm32f1xx_hal_msp.c
      +-- stm32f1xx_it.c
      +-- Stream.h
      +-- syscalls.c
      +-- system.c
      +-- system.h
      +-- system_stm32f103c.h
      +-- Wire.h
  Startup

Problems Tasks Console Properties
CDT Build Console [thermistor2]
```

Figure 26: Code for Thermistor

The screenshot shows the STM32CubeIDE interface. The Project Explorer on the left lists the project structure, including drivers, debug configurations, and source files like main.c, LCD.h, and Stream.h. The main window displays the code for main.c, which includes definitions for Temp_K and Temp_C, and functions for printing to the LCD and updating the display. The code editor has syntax highlighting and various toolbars at the top.

```
#include "main.h"
#include "LCD.h"
#include "Stream.h"

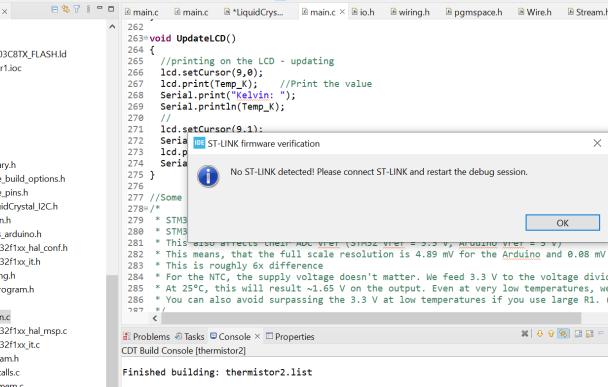
main()
{
    // Initialize LCD and Stream
    LCD_Init();
    Stream_Init();

    // Main loop
    while (1)
    {
        // Print current temperature in Celsius
        LCD_PrintTempC();
        // Update LCD display
        LCD_Update();
    }
}

void LCD_PrintTempC()
{
    // Print "Temperature:" followed by a carriage return
    LCD_Print("Temperature:\r");
    // Set cursor position for value
    LCD_SetCursor(9, 0);
    // Print the temperature value
    LCD_Print(Temp_C);
}

void LCD_Update()
{
    // Set cursor position for unit
    LCD_SetCursor(9, 1);
    // Print "Celsius:" followed by a carriage return
    LCD_Print("Celsius:\r");
    // Set cursor position for value
    LCD_SetCursor(9, 0);
    // Print the temperature value
    LCD_Print(Temp_C);
}
```

Figure 27: Code for Thermistor



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure:
 - workspace_1.12.0 - thermistor2/Core/Src/main.c - STM32CubeIDE
 - File Edit Source Refactor Navigate Search Project Run Window Help
 - Project Explorer (selected)
 - Drivers
 - Debug
 - STM32F103C8T6_FLASH.hd
 - thermistor1.ioc
 - thermistor2
 - Binaries
 - Includes
 - Core
 - Inc
 - build_options.h
 - binary.h
 - core_pins.h
 - LiquidCrystal_I2C.h
 - main.h
 - pins_arduino.h
 - stm32f10x_hal.conf
 - stm32f10x_it.h
 - wiring.h
 - WProgram.h
 - Src
 - main.c
 - stm32f10x_hal_msp.c
 - stm32f10x_it.c
 - Stream.h
 - syscalls.c
 - systems.c
 - system_stm32f1xx.c
 - Wire.h
 - Startup
- Code Editor:** Displays the main.c file with the following code snippet and a tooltip:

```
    262
    263 void UpdateLCD()
    264 {
    265     // Clear the LCD - updating
    266     lcd.setCursor(9,0);
    267     lcd.print(Temp_K); //Print the value
    268     Serial.print("Kelvin ");
    269     Serial.println(Temp_K);
    270
    271     lcd.setCursor(9,1);
    272     Serial ST-LINK firmware verification
    273     lcd.p
    274     Serial
    275 }
```

A tooltip window is open at the bottom of the code editor with the message: "No ST-LINK detected! Please connect ST-LINK and restart the debug session." with an "OK" button.
- Bottom Status Bar:** Shows the message "Finished building: thermistor2.list" and the timestamp "09:10:49 Build Finished. 0 errors, 0 warnings. (took 1s.56ms)".
- Bottom Icons:** Includes icons for Problems, Tasks, Console, Properties, and a Build Console tab for "thermistor2".

Figure 28: Code compiled with 0 errors and is ready to be used for interfacing