Thermal Regulation in a PCR Chamber (June 2021)

Dimas Bakhitasyarf and Dave Danadiva Agusta Peerera

***Abstract*—Polymerase chain reaction (PCR) is a well-known DNA duplication method that has many uses. Amidst the COVID-19 outbreak, diagnosing methods to detect viral infection had experienced a tremendous increase in demand. Various methods are available but the PCR method is one of the most accurate in diagnosing the COVID-19 virus. This paper will report the progress of making thermal regulation in a PCR chamber, which is the main functioning system of a PCR machine in executing the three important processes of DNA duplication.**

***Index Terms*—PCR, DNA, thermocycle, chamber**

I. Introduction

Polymerase chain reaction (PCR) is a laboratory technique that is use to produce copies of many particular regions of DNA or Deoxyribonucleic acid of our interest by generating a large amount of copies from a small initial number of samples. The technique depends on a thermostable DNA polymerase, Taq polymerase and DNA primers are needed to specifically design the region of DNA interest. The cycle reaction involves a series of temperature changes which is done repeatedly, this allows the target copies to be made. The goal of PCR is to produce enough target DNA so that it could proceed to be used or analyzed. The technique is usually used in the biology field (Khan Academy, 2021).

Polymerase itself is an enzyme that makes polymers of any other molecule, which in this case is DNA, while chain reaction is a type of chemical reaction which progresses in an exponential way.

The key ingredients in PCR reactions are polymerase, primers, DNA template and nucleotides. Polymerase is the enzyme that makes new strands of DNA using existing strands as templates. Primers are short sequences of nucleotides which provide a starting point and select the exact point of the DNA of which to be amplified. DNA template is the template of the DNA which we want to amplify. Nucleotides are the basic building blocks for DNA synthesis.

PCR undergoes the temperature change cycle. It is called the thermocycler process which consists of denaturation, annealing and extension. The cycle involves increasing and decreasing the temperature. Denaturation happens at 95 degree celsius which separates the two strands of DNA. Annealing happens between 55 to 65℃. The sample is made cooler so the primers can bind to the target sequence on the single stranded DNA as the starting point. The last process is called extension. Extension happens at 72℃. The polymerase will extend the primers, synthesizing new strands of DNA by adding the complementary DNA of the original strand. The cycle is then repeated (Khan Academy, 2021).

II. Material

- Sample: The sample that will be used in this project is water.
- Chamber: Workspace for the PCR process to prevent contamination.
- Arduino Uno: Supply voltage and act as microcontroller.
- Breadboard MB-102: Workplace to connect all the components.
- Jumper wire: Connect components to the breadboard and for the current to flow.
- Heater 220V AC: Bring the temperature up.
- Fan 12V 0.5A: Bring the temperature down.
- 5V Relay (STM32 PIC AVR ARM uC) (x2): Used for the heater and fan to control them as a switch.
- Thermistor (NTC 3950)-10k +- 1% (-20 - 105 dgC): React to the temperature and send the resistance value.
- Heat sink: spread heat flow away from the device.
- LCD (16X2): Display the result.
- 330 Ohm Resistor: Reduce current flow to LCD.
- 100k Ohm Resistor: Act as fixed resistor value for thermistor resistance reading.
- Adaptor 12V 10A DC: Power source for fan.
- AC Cable: To connect to power source.
- Alumunium foil: To make the container and keep the heat inside.
- Cardboard: To keep the heat inside.
- Fuse: Break circuit when there’s exceeding current flow.

III. Methodology

The sample will first be taken where in this experiment water will be used as the sample. The sample is then placed inside the PCR Chamber. After that, the thermocycle loop process will start. To be able to make sure everything runs properly, the first stage goal is to be able to reach 10 cycles with a satisfying result. If this goal can be achieved, the thermocycle will be tested to run for 20 to 30 cycles.

A. Algorithm

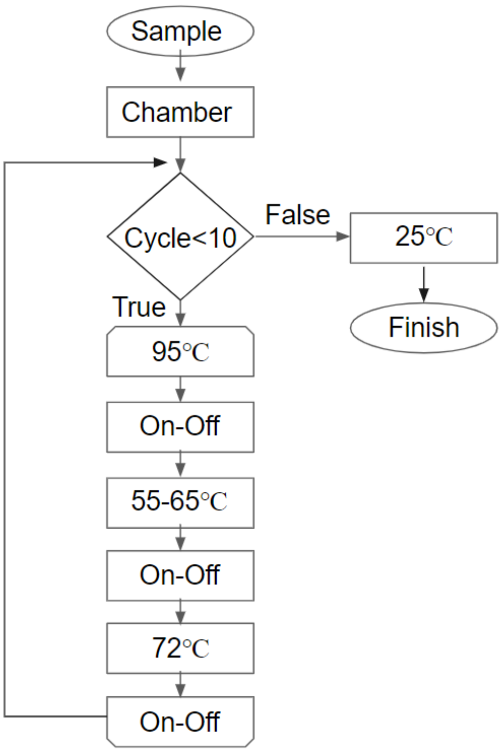

Fig. 3.1 Algorithm flowchart

As shown in figure 3.1, the count of the cycle will start from zero and will enter the thermocycle loop with the goal of 10 cycles. During the thermocycle loop, the temperature is taken up to 95℃ to undergo the denaturation process. The temperature is increased using the heater. Once the temperature reaches 95℃, the temperature needs to be sustained. The temperature is sustained by turning the heater on and off repeatedly. After some specific minutes, the temperature is then dropped to 55-65℃ to undergo annealing. The temperature is dropped by using fan which will cool it down. The temperature is then sustained by turning the heater on and off repeatedly. After some specific minute the temperature is increased again using the heater. The temperature is increased to 72℃ to undergo extension. The temperature is sustained for some specific time again. The cycle is then repeated 10 times. Using while loop, where the loop will keep continuing to run again and again and after reaching the desired cycle prompted, program will break from the loop and cool down to room temperature (25℃).

B. Mechanism of System

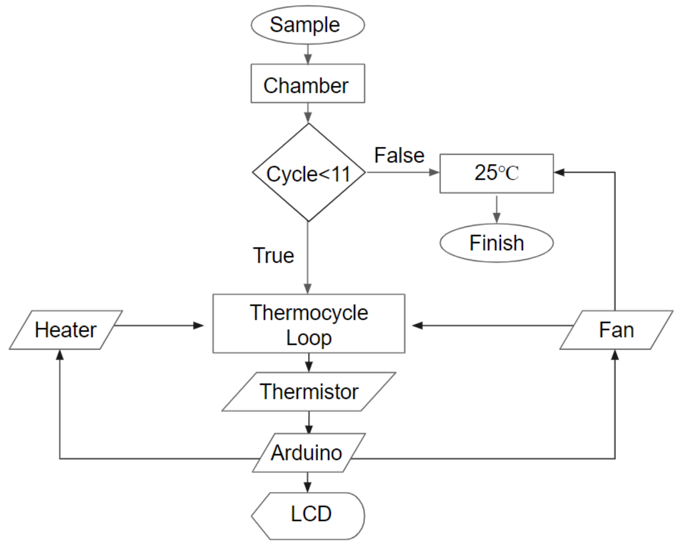

Fig. 3.2 Components relationship to the applied algorithm diagram

The thermistor reads the sample temperature in the chamber. The resistance value will be sent to the arduino microcontroller to be converted into temperature. The reading is then shown using an LCD. Based on the temperature reading, the arduino will control the relay switch in a way to produce and sustain the desired temperature for each process.

C. Exterior Design and Components Utilization

Fig 3.3 Hardware Design Front, Upper, and Side View

In designing the hardware, the first main priority is to make an effective heat distribution. To able to undergo the denaturation process, an ability to reach temperature with maximum of 95℃ is required; hence material with a small surface area for the heat to spread were used in the trial, particularly an alumunium foil container. The container was placed above the heater plate; hence heat source come from below the container. This design makes conduction and convection heat transfer spread efficiently.

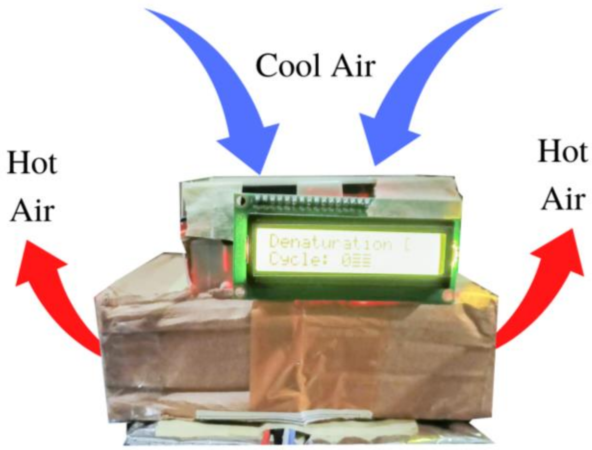

Fig 3.4 Device Air Flow Mechanism

Second main priority is to provide an open gateway for the heat to be release and for the fan cooling system to be able to work properly. This is done by making a stand from a cardboard which will enable the fan cooler to be placed above the sample and facing down into it; thus able blast outer air into the interior of the system. Two ventilation was also provided at the side surface of the exterior to fasten the cooling process when attaining the annealing temperature from denaturation; hence giving paths for the air flow to throw the outer air blasted by the fan and sweeps the hot air from the interior (figure 3.4). This ventilation was tested to work efficiently since it didn't hinder the heating process but rather stabilize both time needed for heating and cooling.

The consideration of using only fan for the cooling system instead of also using peltier is due to the effectiveness and power consumption. In the initial trial, peltier cooler with heatsink attached to fan was used. However, as experimented, the peltier need to be attached to the first cover (shown in figure 3.5) to work properly. The problem is since the peltier will be needed after the denaturation process, the peltier will need to cool down itself first (from 95ºC) before being able to cool the environment temperature. This is of course time consuming and will only require the device more power (need at least 72 Watt to power). On the other hand, with only using fan, it will directly insert and blast outer air to the interior environment and sweep hot air out through the ventilation and require only 6 Watt. Lastly, the size of the overall device is small compared to the real commercial PCR machine. This gives advantageous in porting, installing, and storing the device.

D. Interior Design

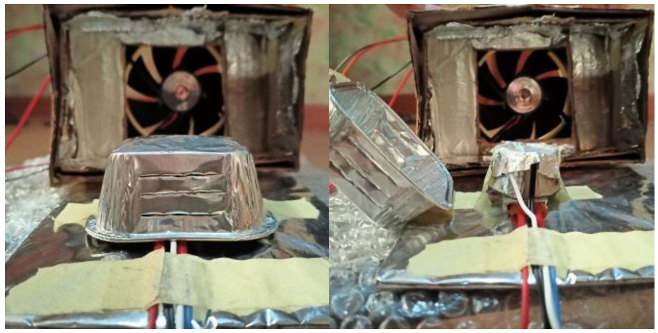

Fig 3.5 First and Second Heat Isolating Cover

Similar with the exterior design, the interior design of the environment was design to make heating and cooling more efficient in term of functionality and time consumption. To support heat isolation for faster heating, the inner layer of the cardboard is layered with alumunium foil and the container was covered with dual alumunium cover. In addition, to support faster cooling, 8 ventilation lines were made at the alumunium cover surface (3 front, 3 at the back, and 1 at each side).

IV. Calculation

NTC thermistor type was used. Meaning, higher in temperature will decrease the resistance.

A. Analog to Digital Signal Conversion

Before working on the resistor to temperature conversion, analog input signal from the thermistor should be converted from voltage analog scaling to resistance value. The fixed resistor used in the circuit is a 100kΩ resistor and the 5V voltage pin will be used to supply, so to find the voltage of the thermistor, voltage divider is used.

$$Vo = \frac{R}{R + 100k} \times Vcc \ ; Vcc = 5V \ (\mathbf{1})$$

Then analog to digital conversion will be done, where the varef is the maximum voltage input (5V), which scaled to 1023 in term of analog read; hence:

$$ADC = Vi \times \frac{1023}{Varef} \ ; Vi = Vo \ (\mathbf{2})$$

Substituting Vi to Vo and solve for resistance value using algebra

$$ADC = \frac{R}{R + 100k} \times Vcc \times \frac{1023}{Varef} \ ; Vcc = Varef$$

$$ADC = \frac{R}{R + 100k} \times 1023$$

$$ADC \times (R + 100k) = R \times 1023$$

$$ADC \times R + ADC \times 100k = R \times 1023$$

$$R = \frac{100k \times ADC}{1023 - ADC} \ (\mathbf{3})$$

B. Resistance to Temperature Conversion

After obtaining the resistance value reading, it can be then converted to temperature using the Steinhart-Hart equation.

$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3 \ \ (\mathbf{4})$$

However, a simplified version of the equation which only consist of 1 constant will be used.

$$\frac{1}{T}=\frac{1}{T0}+\frac{1}{\beta}\ln\left(\frac{R}{R0}\right)\quad \textbf{(5)}$$

Where β can be calculated by:

$$\beta=\frac{\ln\left(\frac{R0}{R}\right)}{\left(\frac{1}{T0}-\frac{1}{T}\right)}\quad \textbf{(6)}$$

Since the calculation uses SI units, the temperature is in Kelvin, while the general usage in daily life temperature reading is in degree Celsius; thus conversion is needed by:

$$°C=K-273.15\quad \textbf{(7)}$$

To be able to make the conversion, first the β constant should be calibrated by equation (6), using the resistance value at room temperature 25°C (298.15 K) which is 10kΩ and another T value with its R value, which are obtained from the product datasheet (attached in the appendix). In this case, Resistance value of 0.7816kΩ with its temperature value of 95°C (368.15 K) are used.

$$\beta=\frac{\ln\left(\frac{10k}{0.7816k}\right)}{\left(\frac{1}{298.15}-\frac{1}{368.15}\right)}$$

$$\beta=3997$$

For temperature calculation, the same T0 and R0 value are used and for demonstration, the same data can be proof using equation (5).

$$\frac{1}{T}=\frac{1}{298.15}+\frac{1}{3997}\ln\left(\frac{0.7816k}{10k}\right)$$

$$T=\left(\frac{1}{T}\right)^{-1}$$

$$T=368.14\ K$$

$$T=368.14-273.15=94.99\ °C$$

V. Discussion

A. Circuit Schematic Design

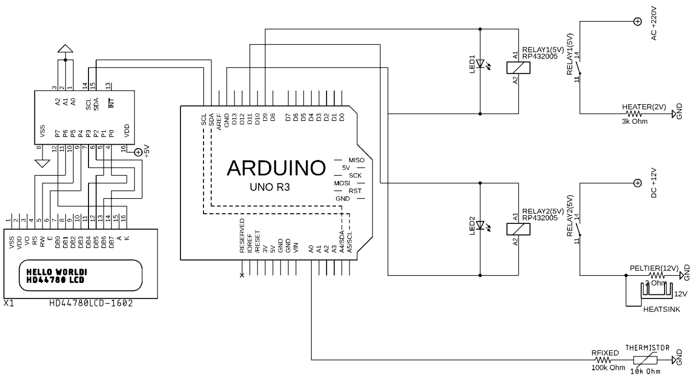

Fig 5.1 Circuit Schematic

The circuit schematic was designed using Eagle Autodesk software. The circuit can be divided into 4 parts (As shown in figure 5.1), which are upper, middle, lower, and LCD parts. At the upper part, the emphasis is for the heating system, where the arduino uno microcontroller from digital pin 9 will control the on and off of the heater plate using a relay, magnetically. The middle part focuses on the cooling system, which connects digital pin 11 with the relay that will also control the switching of the fan cooler. Third, the bottom part, which is characterized by the presence of the 100k Ohm thermistor, which will react to temperature changes by decreasing and increasing the resistance value (in this case is NTC). The input signal will be transferred to the input analog pin 0 to be converted from resistance to temperature. This is the most crucial part as it will be the one that determines what output signal will be sent to the relays controlling the heater and the fan. Lastly, the LCD part that will receive and show the converted temperature reading from the arduino, which is connected by SDA and SCL pin (Analog Input, top left corner of the arduino).

B. Program Parts and Features

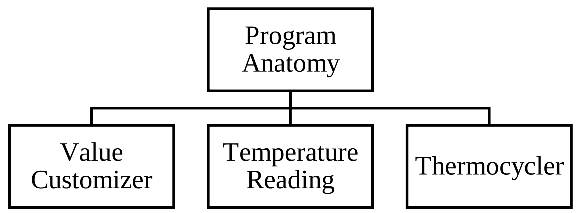

Fig 5.2 Program Anatomy

The program was made into 3 parts. In the value customizer, user can easily change all the controlled variables. For example, the beta coefficient for the temperature reading, the fixed resistor, output and input pins, the desired number of cycle, and the time of how long each cycle will undergo. This makes interaction to the program easier and also debugging process faster.

Second part is the temperature reading, where all live calculation result of the temperature reading (using the calculation method stated in calculation section) will be updated every second. For the reading itself, voltage analog reading can be noisy; hence 50 samples of ADC value will be taken and averaged to produce better reading results. Conversion from Kelvin to Celsius is also undergo in this part of the program.

Third, the thermocycler program which coexist with the other 2 parts of the program. This part regulates the temperature control by using combination of for loop, while loop, and conditional statement. The big loop that control the cycle was made using a cycle counter and regulate by using while loop which will automatically break from the loop when the desired cycles have been reached. For each process in the thermocycle (denaturation, annealing, and extension), for the transition state, for loop and while loop was used to reach the particular temperature. Afterward, to sustain the temperature for each process, conditional statements are used to detect whether the temperature is too high or too low. Then inside it, while loop is used to bring back the temperature to the optimum temperature for each process.

Each processes are also packed with a counter; hence enabling the user to customize how long each process will occur. After each cycle finished and after the desired number of cycles reached, a text printed in the serial monitor will also appear to notify the user. This will help the user to know when will the sample are ready to be analyzed after the process finish.

C. Temperature Regulation Flow

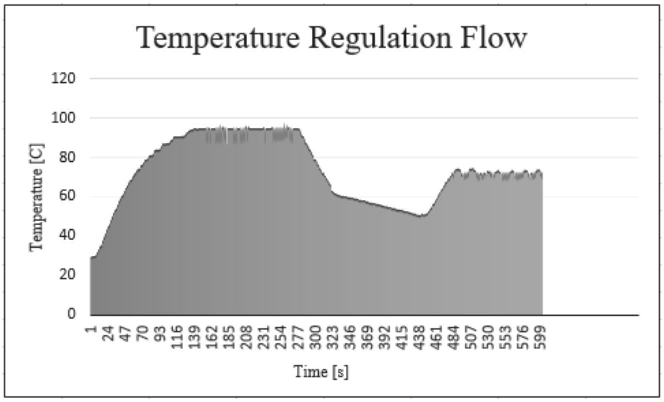

Fig 5.3 One Cycle Temperature Flow

The temperature flow shown in figure 5.3 was tested using the sustaining time of 2 minutes for each process. The timing feature of the program works nicely in regulating the time for each process where each process was able to sustain for 2 minutes as what setted and 1 cycle can be achieved for 10 minutes. As for the process itself, seen in the graph, the heating process to attain the denaturation temperature (95ºC) is not linear compared with the other transition process. This is a good thing because the logarithmic temperature increase form tells that the temperature rising is quite rapid, which exceed linear increase. Then at denaturation (95), annealing (55-65), and extension (72), the temperature can be sustained pretty well at their specific or ranged temperature(s).

D. Power Consumption

Load	Quantity	Voltage	Current	Power
Heater Plate	1	220 V	68 mA	15 Watt
Fan	1	12 V	0.5 A	6 Watt
Relay	2	5 V	90 mA	0.9 Watt
16x2 LCD	1	5V	1 mA	5 mWatt
Total				21.905 Watt

Fig 5.4 Power Consumption Table

Seen from the table in figure 5.4 power needed by the device is 21.905 Watt power. Although it seems that the most affecting component is the heater, it is one of the most energy efficient and best suited heater choice for the device because of its heating area. Overall, this PCR thermocycler device is more energy efficient compared to the commercial PCR device which may need 48 – 60 watt to function.

E. Limitation and Future Improvement

The temperature reading won't be 100% precise due to tolerance value of the resistor, the noise in the analog input, and calculation rounding for each conversion. An effort to make better temperature reading were done by taking the average of multiple samples of ADC value for each second. In the initial trial only 15 samples were taken and the result was not satisfying; hence 50 samples of ADC value was taken and able to give better result for the temperature reading. However, as seen on figure 5.3 noises can still be seen during the sustaining process. Despite the noises, it can be guaranteed that the sustaining process work smoothly as the temperature value was also tested using real thermometer which is not affected by noise in the reading. Another effort to make better reading was also done by using the resistance value of the fixed resistor based on multimeter reading (not the specification).

This device can also still be improved with more functionality. For examples adding small keyboard to be able to customize the number of cycle to be achieved without connecting it to a PC, adding notification to phone; hence user can multitask without monitoring the device's LCD status, and many more. Since this project have lots of potential, future improvement options itself are widely open to be implemented for making a better and functional system.

VI. CONCLUSION

PCR is a DNA duplication method that is used in many fields, mainly for the use of DNA duplication. It involves the thermocycle loop which utilizes temperature. By undergoing all the steps, the result could be obtained. If this project is developed, it could amplify the simplification of the current PCR machine as it is more simple and inexpensive in term of material and power consumption. Since PCR is important as it is now widely used, the simplification of it could fulfill and satisfy the demand of its usage.

APPENDIX

- NTC 3950 Thermistor Temperature to Resistance Data

Resistance @ 25°C = 10KΩ			$B_{25/50}$ =3950K								
T(°C)	R(KΩ)	T(°C)	R(KΩ)	T(°C)	R(KΩ)	T(°C)	R(KΩ)	T(°C)	R(KΩ)	T(°C)	R(KΩ)
-40	277.2	1	30.25	42	4.915	83	1.128	124	0.3434	165	0.1265
-39	263.6	2	28.82	43	4.723	84	1.093	125	0.3341	166	0.1239
-38	250.1	3	27.45	44	4.539	85	1.059	126	0.3253	167	0.1213
-37	236.8	4	26.16	45	4.363	86	1.027	127	0.3167	168	0.1187
-36	224.0	5	24.94	46	4.195	87	0.9955	128	0.3083	169	0.1163
-35	211.5	6	23.77	47	4.034	88	0.9654	129	0.3002	170	0.1139
-34	199.6	7	22.67	48	3.880	89	0.9363	130	0.2924	171	0.1115
-33	188.1	8	21.62	49	3.733	90	0.9083	131	0.2848	172	0.1092
-32	177.3	9	20.63	50	3.592	91	0.8812	132	0.2774	173	0.1070
-31	167.0	10	19.68	51	3.457	92	0.8550	133	0.2702	174	0.1048
-30	157.2	11	18.78	52	3.328	93	0.8297	134	0.2633	175	0.1027
-29	148.1	12	17.93	53	3.204	94	0.8052	135	0.2565	176	0.1006
-28	139.4	13	17.12	54	3.086	95	0.7816	136	0.2500	177	0.0986
-27	131.3	14	16.35	55	2.972	96	0.7587	137	0.2437	178	0.0966
-26	123.7	15	15.62	56	2.863	97	0.7366	138	0.2375	179	0.0947
-25	116.6	16	14.93	57	2.759	98	0.7152	139	0.2316	180	0.0928
-24	110.0	17	14.26	58	2.659	99	0.6945	140	0.2258	181	0.0909
-23	103.7	18	13.63	59	2.564	100	0.6744	141	0.2202	182	0.0891
-22	97.9	19	13.04	60	2.472	101	0.6558	142	0.2148	183	0.0873
-21	92.50	20	12.47	61	2.384	102	0.6376	143	0.2095	184	0.0856
-20	87.43	21	11.92	62	2.299	103	0.6199	144	0.2044	185	0.0839
-19	82.79	22	11.41	63	2.218	104	0.6026	145	0.1994	186	0.0822
-18	78.44	23	10.91	64	2.141	105	0.5858	146	0.1946	187	0.0806
-17	74.36	24	10.45	65	2.066	106	0.5694	147	0.1900	188	0.0790
-16	70.53	25	10.00	66	1.994	107	0.5535	148	0.1855	189	0.0774
-15	66.92	26	9.575	67	1.926	108	0.5380	149	0.1811	190	0.0759
-14	63.54	27	9.170	68	1.860	109	0.5229	150	0.1769	191	0.0743
-13	60.34	28	8.784	69	1.796	110	0.5083	151	0.1728	192	0.0729
-12	57.33	29	8.416	70	1.735	111	0.4941	152	0.1688	193	0.0714
-11	54.50	30	8.064	71	1.677	112	0.4803	153	0.1650	194	0.0700
-10	51.82	31	7.730	72	1.621	113	0.4669	154	0.1612	195	0.0686
-9	49.28	32	7.410	73	1.567	114	0.4539	155	0.1576	196	0.0672
-8	46.89	33	7.106	74	1.515	115	0.4412	156	0.1541	197	0.0658
-7	44.62	34	6.815	75	1.465	116	0.4290	157	0.1507	198	0.0645
-6	42.48	35	6.538	76	1.417	117	0.4171	158	0.1474	199	0.0631
-5	40.45	36	6.273	77	1.371	118	0.4055	159	0.1441	200	0.0619
-4	38.53	37	6.020	78	1.326	119	0.3944	160	0.1410		
-3	36.70	38	5.778	79	1.284	120	0.3835	161	0.1379		
-2	34.97	39	5.548	80	1.243	121	0.3730	162	0.1350		
-1	33.33	40	5.327	81	1.203	122	0.3628	163	0.1321		
0	31.77	41	5.117	82	1.165	123	0.3530	164	0.1293		

- PCR Code

```
//LCD Library
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//LCD Address
LiquidCrystal_I2C lcd(0x27,16,2);

//Temperature Reading
const int    SAMPLE_NUMBER      = 15;
const double BALANCE_RESISTOR   = 100000.0;
const double MAX_ADC            = 1023.0;
const double BETA               = 3997.0;
const double ROOM_TEMP          = 298.15;
const double RESISTOR_ROOM_TEMP = 10000.0;

//Save Current Temperature
double currentTemperature1 = 0;
double currentTemperature2 = 0;
double currentTemperature3 = 0;
double currentTemperature4 = 0;
double currentTemperature5 = 0;
double currentTemperature6 = 0;
double currentTemperature7 = 0;
double currentTemperature8 = 0;
double currentTemperature9 = 0;
double currentTemperature10 = 0;

//Inputs:
int thermistorPin = 0;

//Outputs:
int RelayHeaterPin = 9;
int RelayPeltierPin = 11;

//Number of Desired Cycle:
int cycle = 2;

//Thermocycle Process Time:
int Dentime = 120;
int Anntime = 120;
int Extentime = 120;

void setup()
{
  // Set the port speed for serial window messages
  Serial.begin(19200);
  pinMode(RelayHeaterPin, OUTPUT);
  pinMode(RelayFanPin, OUTPUT);

  // initialize the lcd
  lcd.init();
  lcd.backlight();
}
```

```
//Read Thermistor
double readThermistor()
{
  // variables that live in this function
  double rThermistor = 0;            // Holds thermistor resistance value
  double tKelvin     = 0;            // Holds calculated temperature
  double tcelsius    = 0;            // Hold temperature in celsius
  double adcAverage  = 0;            // Holds the average voltage measurement
  int    adcSamples[SAMPLE_NUMBER];  // Array to hold each voltage measurement

  //Calculate thermistore average resistance
  for (int i = 0; i < SAMPLE_NUMBER; i++)
  {
    adcSamples[i] = analogRead(thermistorPin);  // read from pin and store
    delay(10);        // wait 10 milliseconds
  }

  //average the sample
  for (int i = 0; i < SAMPLE_NUMBER; i++)
  {
    adcAverage += adcSamples[i];      // add all samples up
  }
  adcAverage /= SAMPLE_NUMBER;        // average it w/ divide

  // Thermistor resistance
  rThermistor = BALANCE_RESISTOR * ( (MAX_ADC / adcAverage) - 1);

  //Temperature reading (conversion)[Kelvin]
  tKelvin = (BETA * ROOM_TEMP) /
          (BETA + (ROOM_TEMP * log(rThermistor / RESISTOR_ROOM_TEMP)));

  //Convert Temperature to Celsius
  tcelsius = tKelvin - 273.15;  // convert kelvin to celsius

  return tcelsius;    // Return the temperature in Celsius
}


void loop()
{

 //BIG THERMO CYCLER LOOP

  //Will automatically break after desired cycle achieved

  for (int cyclecount=0 ;cyclecount < cycle; cyclecount += 1){
    int DentimeCount = 0;
    int AnntimeCount = 0;
    int ExtentimeCount = 0;
    readThermistor();
    currentTemperature1 = readThermistor();
    Serial.print("We start at ");
    Serial.print(currentTemperature1);
    Serial.println("C.");
```

```
//DENATURATION
  // OTW to Denaturation Temp
  Serial.println("Attaining Denaturation");
   while(currentTemperature1 < 95.0){
       digitalWrite(RelayHeaterPin, HIGH);
       readThermistor();
       currentTemperature1 = readThermistor();
         Serial.print("it's ");
         Serial.print(currentTemperature1);
         Serial.println("C.");
         lcd.clear();
         lcd.setCursor(0,0);
         lcd.print("Denaturation [^]");
         lcd.setCursor(0,1);
         lcd.print("Cycle: ");
         lcd.println(cyclecount);
         delay(2000);
    // On until temp >= 95.2
    }

 digitalWrite(RelayHeaterPin, LOW);
 digitalWrite(RelayFanPin, LOW);

 // Sustain Denaturation Temp
Serial.println("Sustaining Denaturation");
  while (DentimeCount < Dentime){
   readThermistor();
   currentTemperature3 = readThermistor();

    if (currentTemperature3 < 95.0){

         digitalWrite(RelayHeaterPin, HIGH);
         digitalWrite(RelayFanPin, LOW);

    }

    else if (currentTemperature3 >= 95.0 && currentTemperature3 < 9

         digitalWrite(RelayHeaterPin, LOW);
         digitalWrite(RelayFanPin, LOW);

    }

    else if (currentTemperature3 >= 95.7){

         digitalWrite(RelayHeaterPin, LOW);
         digitalWrite(RelayFanPin, HIGH);

    }

     Serial.print("it's ");
     Serial.print(currentTemperature3);
     Serial.println("C.");
     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("Denaturation [-]");
     lcd.setCursor(0,1);
     lcd.print("Cycle: ");
     lcd.println(cyclecount);
     DentimeCount += 1;
     delay(1000);
     //Break when reach Dentime
   }

 Serial.println("Denaturation Complete");
 digitalWrite(RelayHeaterPin, LOW);
 digitalWrite(RelayFanPin, LOW);
```

```
//Annealing
  // OTW to Denaturation Temp
  Serial.println("Attaining Annealing");
   readThermistor();
   currentTemperature5 = readThermistor();
   while(currentTemperature5 > 66){
       digitalWrite(RelayFanPin, HIGH);
       readThermistor();
       currentTemperature5 = readThermistor();
       Serial.print("it's ");
       Serial.print(currentTemperature5);
       Serial.println("C.");
       lcd.clear();
       lcd.setCursor(0,0);
       lcd.print("Annealing [^]");
       lcd.setCursor(0,1);
       lcd.print("Cycle: ");
       lcd.println(cyclecount);
       delay(2000);
    // On until temp < 66
    }

 digitalWrite(RelayHeaterPin, LOW);
 digitalWrite(RelayFanPin, LOW);

 // Sustain Annealing Temp
  Serial.println("Sustaining Annealing");
  while (AnntimeCount < Anntime){
     readThermistor();
     currentTemperature6 = readThermistor();
     if (currentTemperature6 < 64){

          digitalWrite(RelayHeaterPin, HIGH);

       }

     else if (currentTemperature6 >= 64);{

          digitalWrite(RelayHeaterPin, LOW);

       }

    readThermistor();
    currentTemperature7 = readThermistor();
    Serial.print("it's ");
    Serial.print(currentTemperature7);
    Serial.println("C.");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Annealing [-]");
    lcd.setCursor(0,1);
    lcd.print("Cycle: ");
    lcd.println(cyclecount);
    AnntimeCount += 1;
    delay(1000);
    //Break when reach Anntime
  }

Serial.println("Annealing Complete");
digitalWrite(RelayHeaterPin, LOW);
digitalWrite(RelayFanPin, LOW);
```

```
//EXTENSION
  // OTW to Extension Temp
   Serial.println("Attaining Extension");
   readThermistor();
   currentTemperature8 = readThermistor();
   while(currentTemperature8 < 72.2){
       digitalWrite(RelayHeaterPin, HIGH);
       readThermistor();
       currentTemperature8 = readThermistor();
       Serial.print("it's ");
       Serial.print(currentTemperature8);
       Serial.println("C.");
       lcd.clear();
       lcd.setCursor(0,0);
       lcd.print("Extension [^]");
       lcd.setCursor(0,1);
       lcd.print("Cycle: ");
       lcd.println(cyclecount);
       delay(2000);
    // On until temp < 72.2
    }

  digitalWrite(RelayHeaterPin, LOW);
  digitalWrite(RelayFanPin, LOW);
 // Sustain Extension Temp
Serial.println("Sustaining Extension");
  while (ExtentimeCount < Extentime){
   readThermistor();
   currentTemperature9 = readThermistor();

    if (currentTemperature9 < 72.2){

         digitalWrite(RelayHeaterPin, HIGH);
         digitalWrite(RelayFanPin, LOW);

    }

    else if (currentTemperature9 >= 72.2 && currentTemperature9 < 72.7){

         digitalWrite(RelayHeaterPin, LOW);
         digitalWrite(RelayFanPin, LOW);

    }

    else if (currentTemperature9 >= 72.7){

         digitalWrite(RelayHeaterPin, LOW);
         digitalWrite(RelayFanPin, HIGH);

    }
         readThermistor();
         currentTemperature10 = readThermistor();
         Serial.print("it's ");
         Serial.print(currentTemperature10);
         Serial.println("C.");
         lcd.clear();
         lcd.setCursor(0,0);
         lcd.print("Extension [-]");
         lcd.setCursor(0,1);
         lcd.print("Cycle: ");
         lcd.println(cyclecount);
         ExtentimeCount += 1;
         delay(1000);
         //Break when reach Extentime
       }

   Serial.println("Extension Complete");
   digitalWrite(RelayHeaterPin, LOW);
   digitalWrite(RelayFanPin, LOW);

 //AFTER EACH CYCLE PROCESS ===================

  digitalWrite(RelayHeaterPin, LOW);
  digitalWrite(RelayFanPin, LOW);

  //cycle count notification
  Serial.print(cyclecount + 1);
  Serial.println(" CYCLE HAVE BEEN ACHIEVED");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(cyclecount + 1);
  lcd.println(" CYCLE HAVE");
  lcd.setCursor(0,1);
  lcd.print("BEEN ACHIEVED");
  delay(3000);

 }
 //After desired cycle had been achieved
 Serial.println("YOUR SAMPLE IS READY");
 delay(3000);
 digitalWrite(RelayHeaterPin, LOW);
 digitalWrite(RelayFanPin, LOW);

 for (;;){}

}
```

ACKNOWLEDGMENT

Thank you to Swiss German University electrical engineering lab lecturers for all guidance and feedback also NN-Digital shop for providing complete materials specification.

REFERENCES

[1] BYJUS. (n.d.). Difference Between Analog And Digital Signal in Tabular Form | BYJU'S. [online] Available at: https://byjus.com/physics/difference-between-analog-and-digital/.

[2] Khan Academy (2021). Polymerase chain reaction (PCR) (article). [online] Khan Academy. Available at: https://www.khanacademy.org/science/ap-biology/gene-

expression-and-regulation/biotechnology/a/polymerase-chain-reaction-pcr.

[3] ncd.io. (2017). Relay Logic - How to Connect Relays for Logical Switching Applications. [online] Available at: https://ncd.io/relay-logic/.

[4] Public Health Ontario. (n.d.). *Explained: COVID-19 PCR Testing and Cycle Thresholds*. [online] Available at: https://www.publichealthontario.ca/en/about/blog/2021/explained-covid19-pcr-testing-and-cycle-thresholds.

[5] Sciencedirect. (2017). Reverse Transcription Polymerase Chain Reaction - an overview | ScienceDirect Topics. [online] Available at: https://www.sciencedirect.com/topics/neuroscience/reverse-transcription-polymerase-chain-reaction.