Good afternoon dear examiner. Today, I would like to introduce our temperature control system. Our system contains of 3 main components: user interface, current amplifier, signal conditioner and the processing unit. The main functionality of our system is to apply highly controlled temperature change on a small surface. Ranging from 5 to 45 Celsius degree. During practical experiment, our system can reach its maximum temperature within 40 seconds from room temperature, and cool down from that to 5 degrees within 2 mins. The accuracy of the temperature controller is within 0.01 degree. The system’s accuracy and efficiency makes it extremely suitable to be applied to polymerase chain reaction, a biomedical reaction for generating DNA strand replica, which requires multiple heating and cooling cycle. Now, please allow me to get into the detail of our solution, starting from the user interaction module.

The user interface of our system is consisted of local input, local output, and cloud output. The local input is an 10K potential meter nob connected to an analog pin of the Arduino, the center processing unit. The user can adjust the desired temperature of the system by using this nob. A liquid crystal display is used to display the current temperature of the system and the desired temperature set by the user. For more information of the system’s status, we used to an Arduino Wifi Shield to obtain http connectivity, which allow us to post data onto the internet. The software implementation of that is pretty straightforward, a lib called <HttpClient.h> is used to push data to Dweet.io.

A thermistor is used for temperature sensing of the system. Knowing that the resistance of the thermistor will change according to the temperature, we adopted the method of using a NE555 timer in a stable mode to detect the change. The change in resistance along with the temperature can be recorded by using Voltage Division method by applying a constant voltage across the thermistor in series with a known resistor as well, however the relationship between the voltage and temperature is highly depend on the resistance of the known resistor, which is very inaccurate because the tolerance of that could be ranging from 2% to 20%. We exploit the fact that NE555’s output frequency is changing sensitively according to its configuration of resistor combination, given by this equation from the official specification [], therefore, the change in temperature around the thermistor will result in the change of resistance, and further result in the change of frequency output of NE555. The relationship between temperature and frequency can be obtained in the collaboration by record a curve contain temperature data from AD590 and frequency of the NE555 output by using an Arduino digital pin. An Arduino digital pin is capable of reading frequency input ranging from 0 – 10K Hz in an accurate manner (proven by website). After obtaining the curve, temperature value can be regenerated easily using the curve. Nevertheless, using Arduino PulseIn function to calculate the frequency usually result in fluctuated reading. Unstable reading of temperature is one of the limitations of our system.