# Building a Precision Fridge

Azid Harun

November 8, 2018

**Abstract**

In this project, a precision refrigerator named PyFridge is built and designed together with a Peltier thermoelectric device to precisely control the temperature of water in the beaker. The temperature of water and heat sink are monitored using 1-wire temperature sensors. PyFridge uses the pulse-width modulation signal provided by Raspberry Pi to cool down and maintain the temperature water. The best duty cycle coefficient $\alpha$ is experimentally determined to be 150. With this chosen value, PyFridge exhibits a good performance of cooling within 10 minutes with a small hysteresis loop after reaching the desired temperature of the refrigerator.

## Declaration

I declare that this project and report is my own work.

Signature:  Date: 29/11/2018

# Contents

# 1 Introduction

Most of the laboratories nowadays require a precision refrigerator for preservation and storage for their laboratory products. Almost every reagent in chemistry experiments require precise refrigeration to remain stable. For medical laboratory, it is also used to store cerebral-spinal fluid, post-tested blood products, and urine. In this project, a precision refrigerator is built using electronic devices such as Raspberry Pi, temperature sensor, power transistor, Peltier device, heat sinks, DC power supply and some laboratory apparatus. [1]

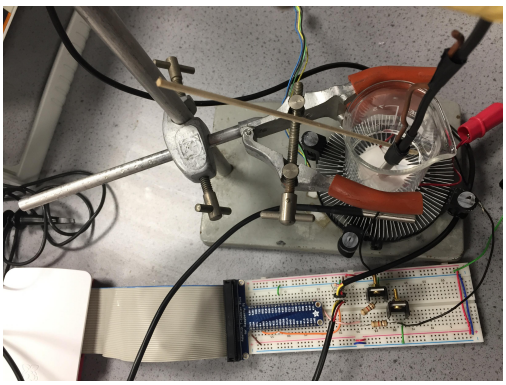


**Figure 1:** Project arrangement listing all equipment and apparatus required.

# 2 Theory

## 2.1 Newton's Law of Cooling

This project utilised one of the famous law stated by Sir Isaac Newton in 1701 which is known as Newton's Law of Cooling. This law states that the rate of heat loss of a body is directly proportional to the difference in the temperatures between the body and its surroundings provided the temperature difference is small and the nature of radiating surface remains same. This law can be further interpreted as shown in Equation 1.

$$\frac{dT}{dt} \propto (T - T_{\mathrm{a}}) \tag{1}$$

where T and $T_a$ are the temperature of object and ambient, respectively. Since this refrigerator cools the water through convection, it can be assumed to be described with Newton's Law of Cooling, with a certain modification on the equation.

## 2.2 Temperature Sensor DS18B20

The waterproof temperature sensor DS18B20 monitor the temperature of the fluid. The communication protocol of this temperature sensor is known as 'One-Wire', which uses only one wire to transmit the temperature measurements to the Raspberry Pi. Normally, DS18B20 requires three wire for its operation which are the power supply line $V_{cc}$, ground line $G_{nd}$ and the data wire DQ. It can also be operated in parasite power mode where only the data and ground lines are used, with power being supplied through the data line. The DS18B20 can measure the temperature within range of -55°C to 125°C.

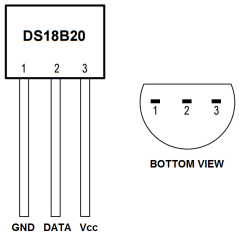



**Figure 2:** Illustration of the pin out of DS18B20 temperature sensor. [2]

## 2.3 Peltier thermoelectric device

In order to cool down the fluid, a thermoelectric cooler which operate based on Peltier effect is used. This effect is discovered by a French Physicist Jean Peltier. It transfers heat across between two electric junctions, resulting a difference in temperature. As a voltage is applied across the junctions, an electric current is formed and then flows in between them, causing heat to be deposited at one junction. Hence, the cooling process occurs on the other junction. [3]

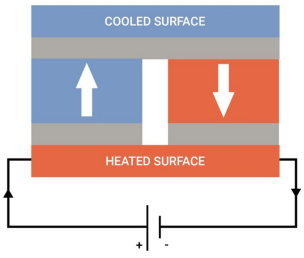



**Figure 3:** Visualisation of Peltier effect. [4]

## 2.4 Power transistor Darlington TO-220

Transistors are semiconductor electronic devices with three terminals which are Base, Collector and Emitter pins. These transistors take small current into the Base pin and then allow a larger current to flow between collector and emitter pins. Power transistor usually has a really low output resistance in order for a large load current to flow. Therefore, it needs to be in contact with heat sink so that it can dissipate heat in order to prevent overheating. In this project, power transistor is used as a switch in Peltier circuit. [5]

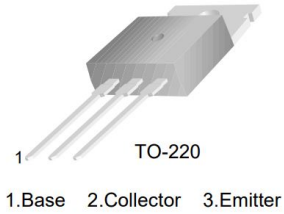
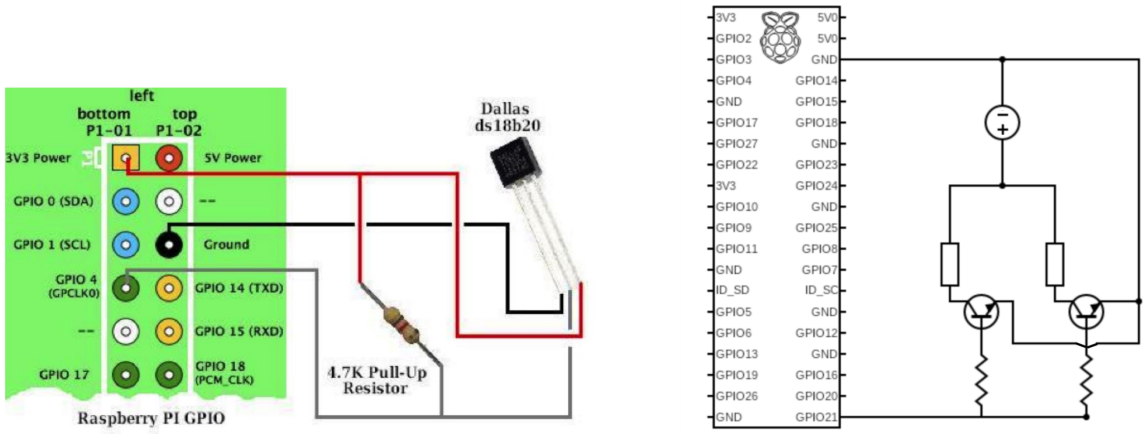



**Figure 4:** Illustration of the pin out of Power Transistor Darlington TO-220 BDX33 . [6]

# 3 Procedures

## 3.1 Equipment and apparatus required

- Raspberry-Pi
- DC power supply
- 2 Peltier thermoelectric device
- 2 Waterproof temperature sensor S18B20
- 4.7 kΩ and 1.0kΩ resistor
- 2 Power transistor Darlington TO-220
- Heat sinks
- Beaker
- Retort stand with clamps
- Breadboard and wires




**(a)** Temperature Sensor Circuit. [7] **(b)** Peltier Control Circuit. [8]

**Figure 5:** Electronic circuit diagram

## 3.2 Building the refrigerator

This project requires two electronic circuit working together, which are temperature sensors circuit and Peltier control circuit.

For the temperature sensors circuit, the temperature sensors are connected to the Raspberry Pi as shown in the Figure 5a. The sensor output is located by identifying the serial number of DS12B20 temperature sensors in the file as shown in code below. The existence of the serial number in the above directory means that the circuit for the temperature sensor worked perfectly fine.

```
student21@dahpi14 ~ $ cd /sys/bus/w1/devices
student21@dahpi14 /sys/bus/w1/devices $ ls
28-000005eb9151 28-000005ead278 w1_bus_master1
```

In the case for Peltier control circuit, a voltage is produced using DC power supply and then controlled using power transistors that connected to a GPIO pin of Raspberry Pi with 1kΩ resistors. This electronic circuit is shown in Figure 5b. The power supply is set up such that 3.8V is supplied to the Peltier circuit, which is the maximum value of voltage that the Peltier can withstand. As the Peltier device is turned on, one of its side became hot as the other side gets cold. Therefore, it is required to be in contact with a heat sink to prevent overheating, otherwise the Raspberry Pi will freezes or the Peltier device will be damaged. A thermal conducting paste is placed in between its hot side with the heat sink to maximise the heat transfer. [8]

The heat sink is placed on the base of the retort stand. The Peltier device is placed under the clamped beaker containing tap water, such that its cold side facing the bottom part of the beaker and the hot side is in contact with the heat sink. The first DS18B20 temperature sensor is immersed in the beaker without touching the glass so that the sensor will be measuring the water temperature, instead of the glass temperature. The second

temperature sensor is placed such that it is in contact with the heat sink to measure its temperature. This is to ensure that there is a heat exchange between the hot side of Peltier with the heat sink when the Peltier circuit is turned on. The project arrangement is shown in Figure 1.

## 3.3 Regulating the temperature

A Python script is written to plot a graph that is able to update itself as the temperature measurements for both temperature sensors are made, as shown in Appendix A. In this script, this refrigerator settings use the pulse-width modulation(PWM) signal to vary the cooling power of the Peltier element.

Pulse-width modulation creates digital signal to generate analog signal. The behaviour of these PWM signals are defined by two main properties which are the Duty Cycle and frequency. Duty cycle is a percentage of the total time taken to complete a cycle where the digital signal is in the high state. The frequency determines how fast these PWM signal complete a cycle. By setting up a PWM signal with varying value of duty cycle with high frequency, the cooling power of Peltier module can be varied. For example, a duty cycle of 50% and a frequency of 1000 output a half Peltier cooling power. The current flow in Peltier circuit will vary depending on the amount of duty cycle at the time. [9]

In Appendix A, all the required packages are imported at the beginning of the code. The Raspberry Pi is set to be programmed using Broadcom(BCM) pin numbers. The pin for peltier circuit is then declared and initialised as output. The code first prompted the user to input the desired temperature of the refrigerator, $T_o$. With the temperature of water, T being measured by using one of the temperature sensor DS18B20, the duty cycle of the cooling process is varied such that it reduces as the difference between the temperature of water at the time and the refrigerator temperature decreases. The cooling rate is then maintained to be constant as water temperature reaches the refrigerator temperature. In improving the accuracy of water temperature getting closer to the refrigerator temperature, the duty cycle coefficient, $\alpha$ is introduced as shown in Equation 2. This coefficient $\alpha$ is then determined experimentally until the best accuracy in this case is achieved.

$$\text{Duty Cycle} = \alpha(T - T_o) \tag{2}$$

Equation 2 have the same form as Newton's Law of Cooling. However, in this case, the duty cycle will always be positive and this is equivalent to the rate of change of temperature of object, $\frac{dT}{dt}$ in Equation 1 being negative due to cooling process.

In order to obtain a better temperature measurement of water, the water is stirred using an insulated thin rod. This is to ensure that the temperature gradient across the water in the beaker is minimised. Without stirring, regions of hotter and colder spots in the water will be present and this could become a factor that contributes to a lesser accuracy in measurement of water temperature.

In order to determine whether our refrigerator works or not before making further analysis, the project is ran by measuring the peltier directly with one of the temperature sensor. The temperature sensor is placed such that its top surface is directly contact with the cold side of peltier. The result is shown in Figure 6 below as the desired refrigerator temperature is chosen to be 21.0°C.

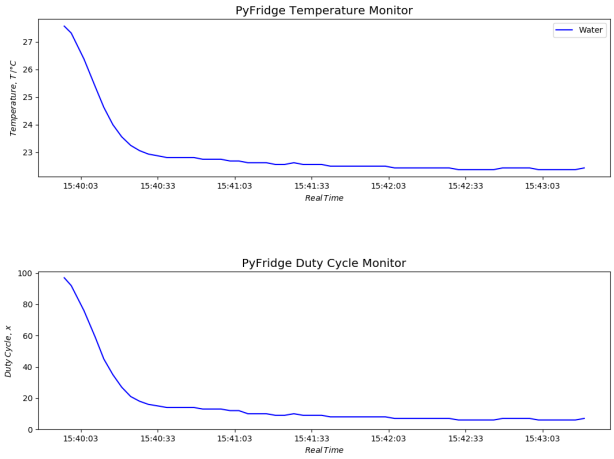



**Figure 6:** Result for Project Testing.

# 4 Analysis

## 4.1 Determination of best duty cycle coefficient, $\alpha$

Analysis of Equation 2 tells that if the chosen value of duty cycle is too small, it will not provide enough cooling power as the difference between the temperature of water and refrigerator decreases. The temperature of water in this case decreases gradually with time. This causes the performance of the refrigerator in term of accuracy to drop. In other case, an overlarge value of duty cycle cause the refrigerator to perform a hysteresis loop in the cooling process. The project is re-ran with a refrigerator temperature of 21.0°C and different values of duty cycle coefficient. The results for this project run are shown in Appendix B. After taking consideration of all the analysis above, the chosen duty cycle coefficient $\alpha$ is determined to be 150.

## 4.2 PyFridge performance

The project is then re-ran with duty cycle coefficient $\alpha$ of 150 and multiple desired temperatures to see how good is the refrigerator work in term of accuracy as well as to see how cold can the refrigerator make the liquid. The results are shown in Appendix C. The rise in temperature of heat sink indicates that the Peltier in in the cooling state.

| $T_o$ (°C) | $T_i$ (°C) | $T_{10min}$ (°C) | $\Delta T$ (°C) |
|---|---|---|---|
| 20.00 | 20.50 | 20.06 | 0.06 |
| 19.00 | 21.50 | 19.63 | 0.63 |
| 18.00 | 20.38 | 18.60 | 0.60 |

**Table 1:** Analysis of water temperature after 10 minutes.

Based on the graphs in Appendix C, PyFridge shows a good performance in cooling the water where the refrigerator managed to cool down the water to all three chosen desired temperature of the refrigerator which are $T_o$ = 18.0°C, 19.0°C and 20.0°C. However, the small hysteresis loop is seems to be forming in Figure 11 and 12. According to the analysis of Equation 2, this means that the duty cycle coefficient is still larger than the best value.

On the contrary, a problem found while running the project where it is hard for the refrigerator to cool down the water to a temperature lower than 19.0 °C as the temperature of heat sink exceeds 40.0 °C. This is probably due to the heat radiated by the heat sink to the surrounding. As a consequence, this affects the temperature of the water in the beaker. In order to minimise this issue, the current heat sink should be replaced with the one in which its material have a larger value of heat capacity. This issue also indirectly contributes to a decrease in performance of the refrigerator in terms of accuracy for a lower chosen duty cycle coefficient. In that case, the temperature of water would probably take a longer time in reaching the desired temperature of refrigerator.

Therefore, the current chosen duty cycle coefficient can be considered as the best value even though a small hysteresis loop is formed after the temperature of water reaches the desired temperature of refrigerator.

# 5 Conclusions

Based on the analysis of the project, PyFridge shows a good cooling performance in terms of accuracy and precision when the duty cycle coefficient, $\alpha$ used is 150. The current heat sink should be replaced with the other one in which its material have a larger heat capacity in order to prevent the heat sink from overheating quickly. If the project is to be repeated again, the project can be done with a slightly lower duty cycle coefficient from the current chosen value so that the formation of hysteresis loop can be minimised.

# References


[1] A. K. Long. 2009. *The Many Uses of the Laboratory Refrigerator*. Available at *http://ezinearticles.com/?The-Many-Uses-of-the-Laboratory-Refrigerator&id=3425642*. [Accessed 8 November 2018].

[2] Maxim Integrated Products, Inc. 2018. *DS18B20 - Programmable Resolution 1-Wire Digital Thermometer*. Available at *https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf*. [Accessed 8 November 2018].

[3] II-VI Marlow. 2018. *Thermoelectric Technology*. Available at *https://www.marlow.com/resources/thermoelectric-technology-guide*. [Accessed 10 November 2018].

[4] Instructables. 2018. *How to Set Up a Peltier Module*. Available at *https://www.instructables.com/id/How-to-Set-Up-a-Peltier-Module* [Accessed 10 November 2018].

[5] Farnell. 2018. *Power Transistor Definition*. Available at *https://uk.farnell.com/power-transistor-definition*. [Accessed 15 November 2018].

[6] DigChip. 2000. *Fairchild Semiconductor BDX33 datasheet*. Available at *https://www.digchip.com/datasheets/parts/datasheet/161/BDX33-pdf.php*. [Accessed 15 November 2018].

[7] *Checkpoint 5 Temperature Sensors, Chapter 2 DAH Checkpoints*. Available at *https://www.learn.ed.ac.uk/bbcswebdav/pid-3430212-dt-content-rid-6725057_1/courses/PHYS101002018-9SS1SEM1/dah-checkpoints.pdf*.

[8] *Project E: Building a Precision Refrigerator, Chapter 3 DAH Projects*. Available at *https://www.learn.ed.ac.uk/bbcswebdav/pid-3496840-dt-content-rid-6997984_1/courses/PHYS101002018-9SS1SEM1/dah-projects%281%29.pdf*.

[9] National Instruments. 2018. *What is a Pulse Width Modulation (PWM) Signal and What is it Used For?* Available at *https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019OkFSAU&l=en-GB*. [Accessed 15 November 2018].

# Appendix A PyFridge Project Script

```
# Import required packages
from webiopi.devices.sensor.onewiretemp import DS18B20
import pylab
import matplotlib.animation as animation
import datetime
import time
import RPi.GPIO as GPIO

# The function to call each time the plot is updated
def updatePlot(i):

    # Declare the temperature sensors
    tmp1 = DS18B20(slave="28-000005eb9151")
    tmp2 = DS18B20(slave="28-000005ead278")

    # Readout temperature sensor
    temp_water, temp_heat_sink = tmp1.getCelsius(), tmp2.getCelsius()
    elapsed_time = time.time() - start_time

    # Obtain measured temperature after 10 mins ( 600s ).
    if elapsed_time >= 600.0:
        print('Time elapsed : {}'.format(elapsed_time))
        print('Measured temperature after 10min : {}'.format(temp_water))

    # Calculate the current duty cycle
    exp_coefficient = 150
    duty_cycle = int(exp_coefficient*(temp_water - temp_set))

    # Regulate the duty cycle
    if duty_cycle > 100:
        duty_cycle = 100
    elif duty_cycle < 0:
        duty_cycle = 0

    cool.ChangeDutyCycle(duty_cycle)
    time.sleep(1)

    # Store the current time and measurements
    timeValues.append( datetime.datetime.now() )
    measurements_water.append( temp_water )
    measurements_heat_sink.append( temp_heat_sink )
    measurements_duty_cycle.append( duty_cycle )

    # Clear the old plot
    plotFigure.clear()

    # Make the new plot
    pylab.subplot(2, 1, 1)
    pylab.xlabel(r'$Real\/Time$')
    pylab.ylabel(r'$Temperature,\/T\//\degree C$')
    pylab.title('PyFridge Temperature Monitor', fontsize='x-large')
```

```
    pylab.plot( timeValues, measurements_water, 'b-', label='Water' )
    pylab.plot( timeValues, measurements_heat_sink, 'r-', label='Heat Sink' )
    pylab.legend()

    pylab.subplot(2, 1, 2)
    pylab.xlabel(r'$Real\/Time$')
    pylab.ylabel(r'$Duty\/Cycle,\/x$')
    pylab.title('PyFridge Duty Cycle Monitor', fontsize='x-large')
    pylab.plot( timeValues, measurements_duty_cycle, 'b-', label='Water' )
    pylab.ylim((0,101))

    pylab.subplots_adjust(hspace=0.6)

# Hide any warnings
GPIO.setwarnings(False)

# GPIO programming by BCM pin numbers
GPIO.setmode(GPIO.BCM)

# Declare the pin
peltier = 21

# Initialises the pin as output
GPIO.setup(peltier, GPIO.OUT)

# Set up the plot object
plotFigure = pylab.figure()

# Prompt required data for refrigeration
temp_set = float(input('Input the fridge temperature, Tf in centigrade : '))

# Initialise the Pulse Width Modulation process
cool = GPIO.PWM(peltier, 100)
cool.start(0)

# Empty arrays of time and measurement values to plot
timeValues, measurements_water, measurements_heat_sink, measurements_duty_cycle = [ ], [
    ], [ ], [ ]

# Set initial time
start_time = time.time()

# Create a live plot
ani = animation.FuncAnimation(plotFigure, updatePlot, interval=200)
pylab.show()
```

# Appendix B PyFridge Project Run with $T_o = 21.0°C$

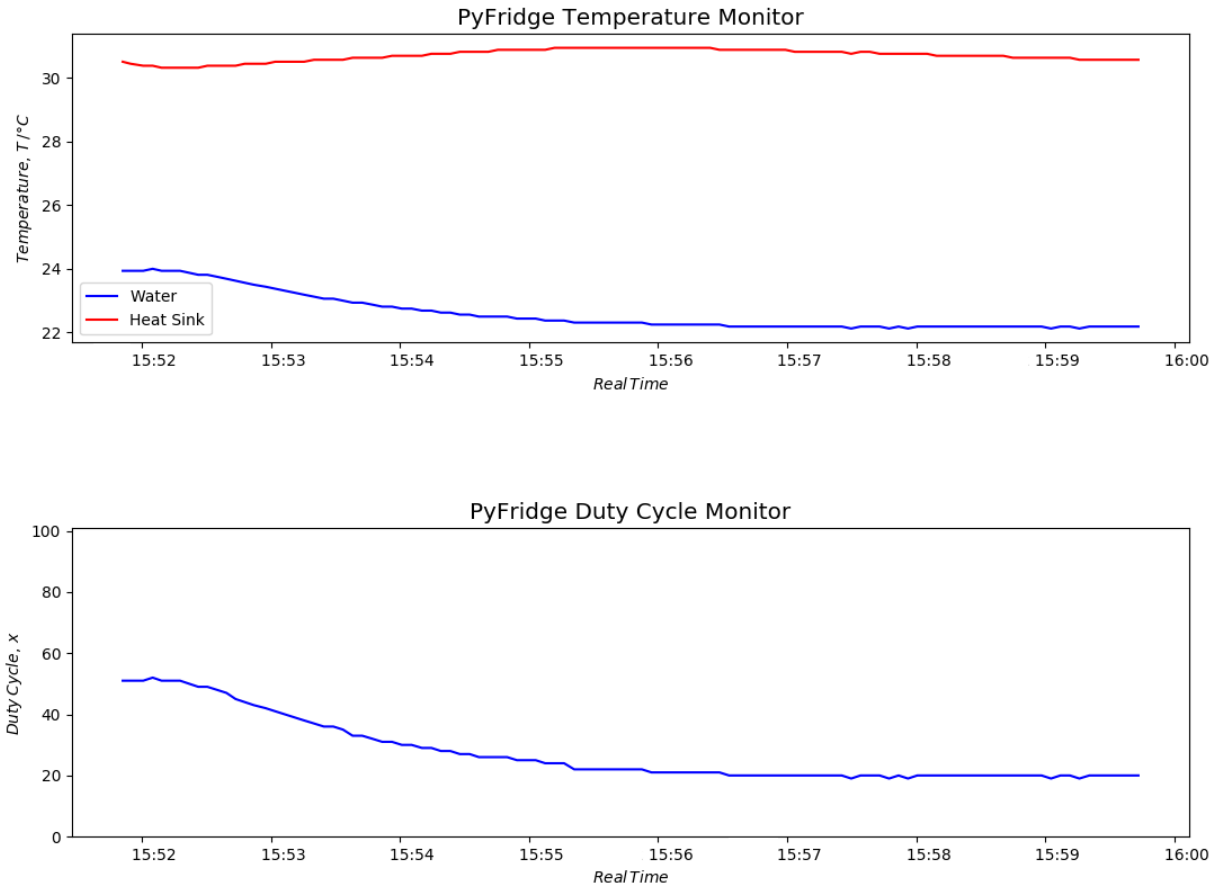



**Figure 7:** Coefficient : 17.5

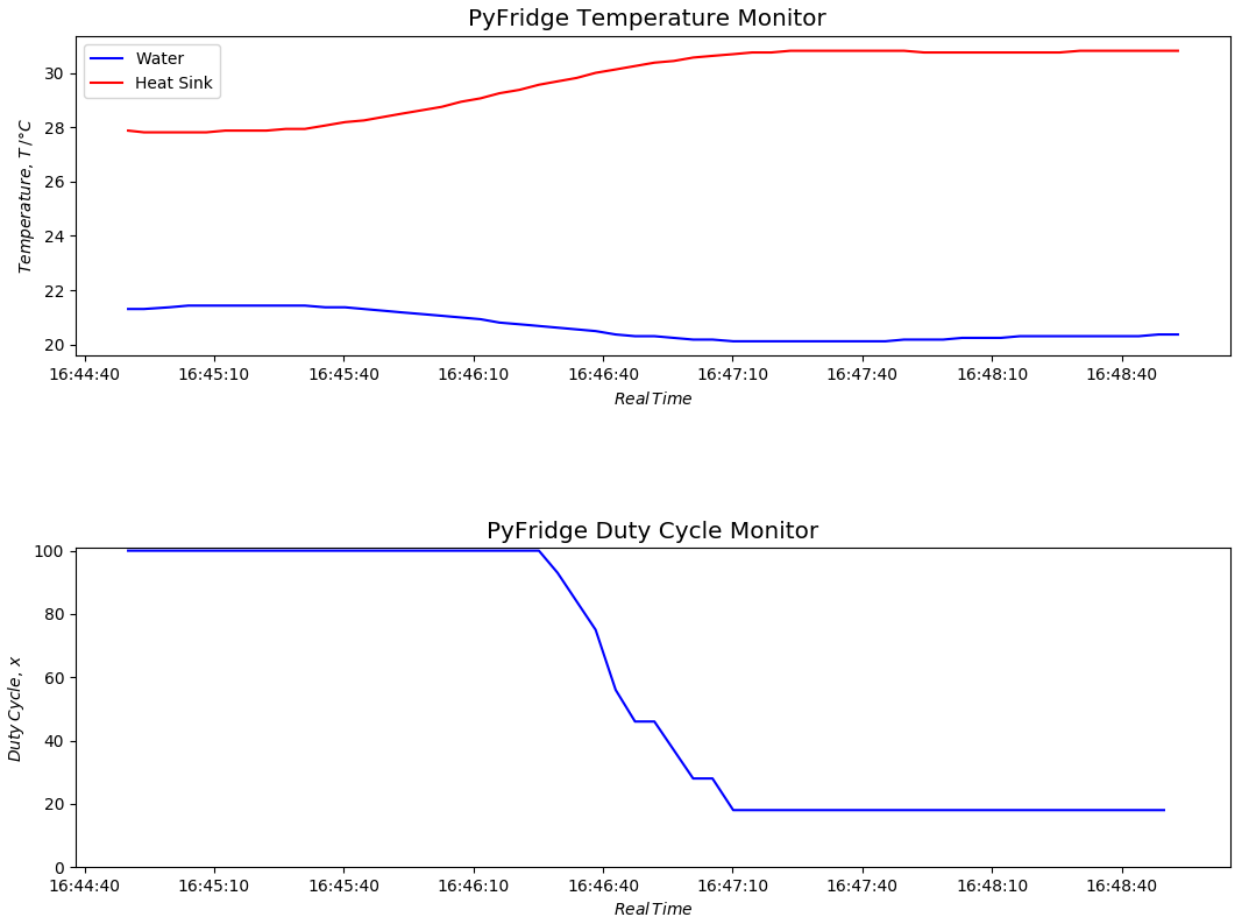



**Figure 8:** Coefficient : 150

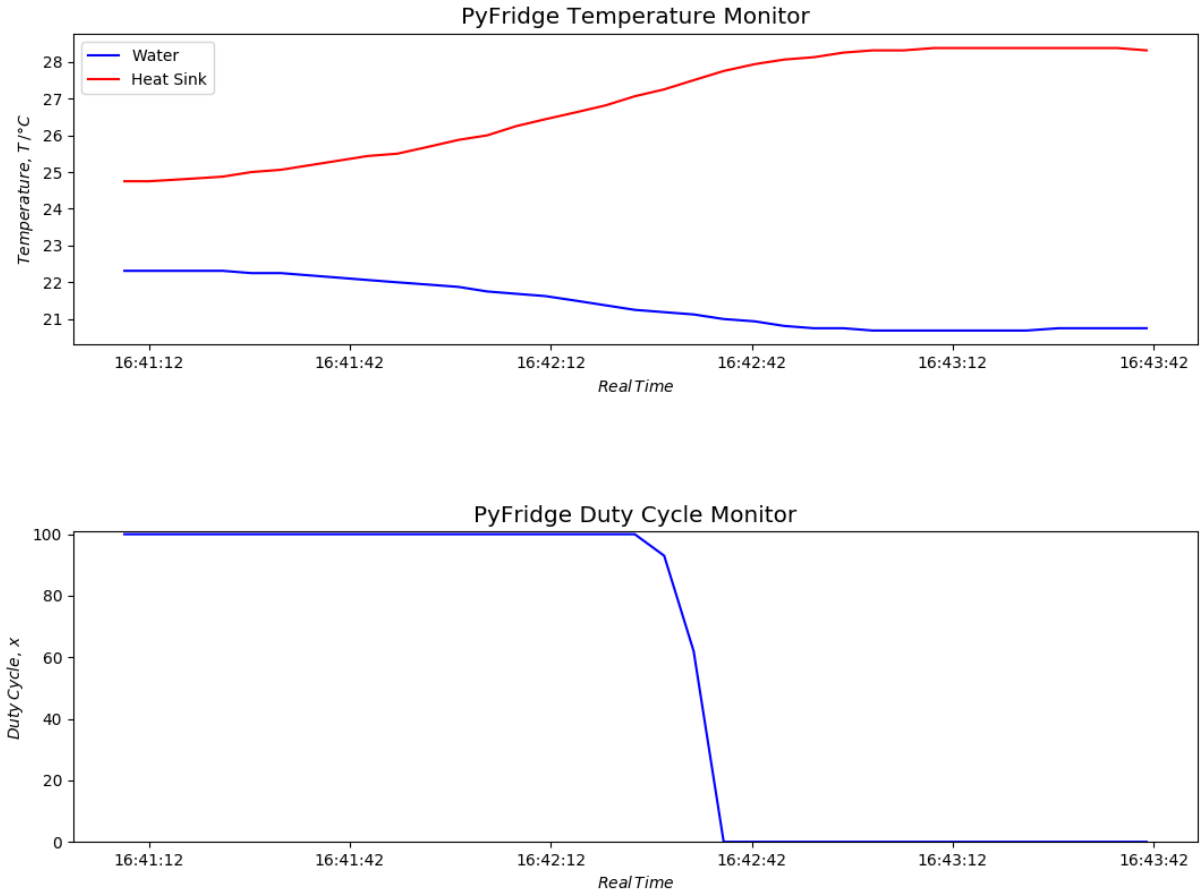



**Figure 9:** Coefficient : 500

# Appendix C PyFridge Project Run with $\alpha = 150$

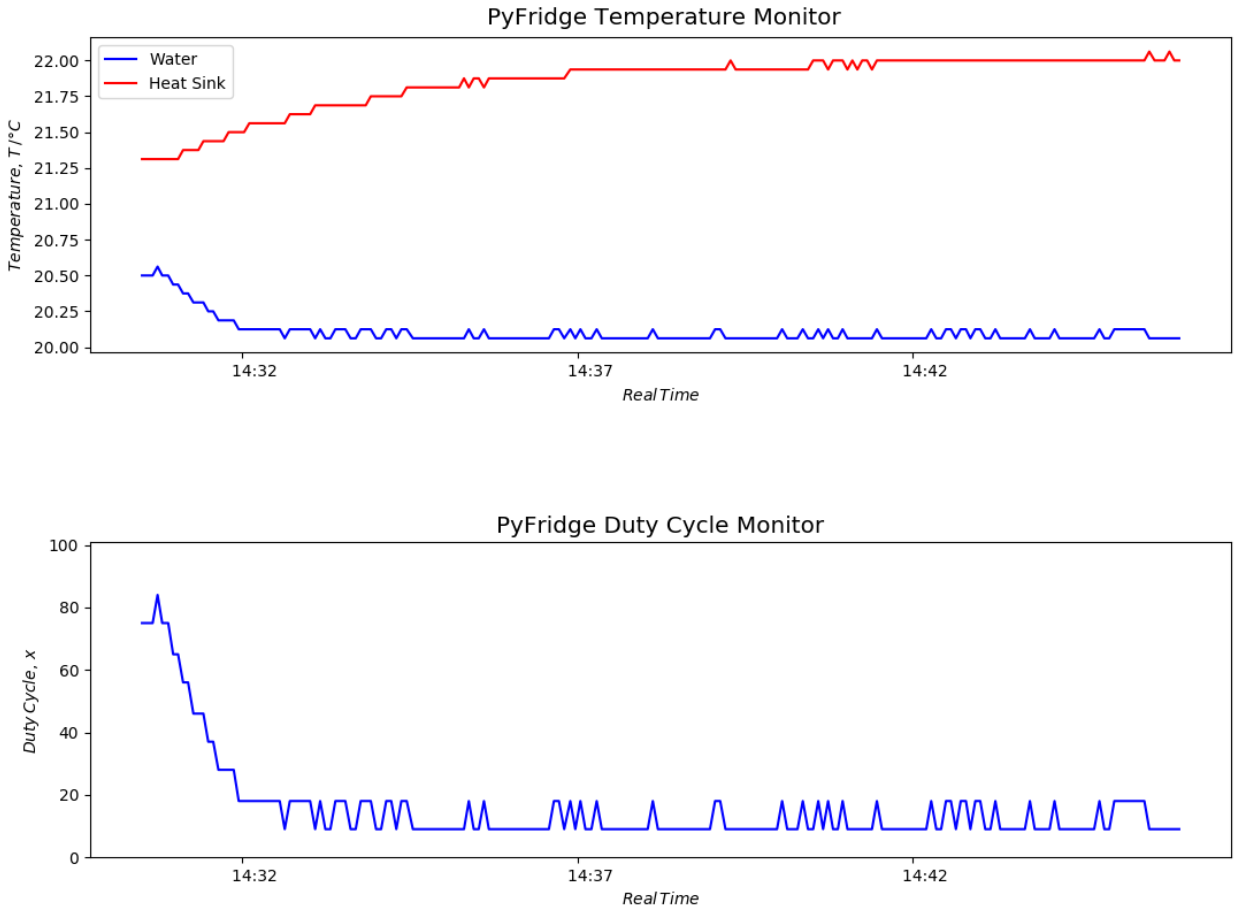



**Figure 10:** Temperature of the refrigerator : 20.0°C

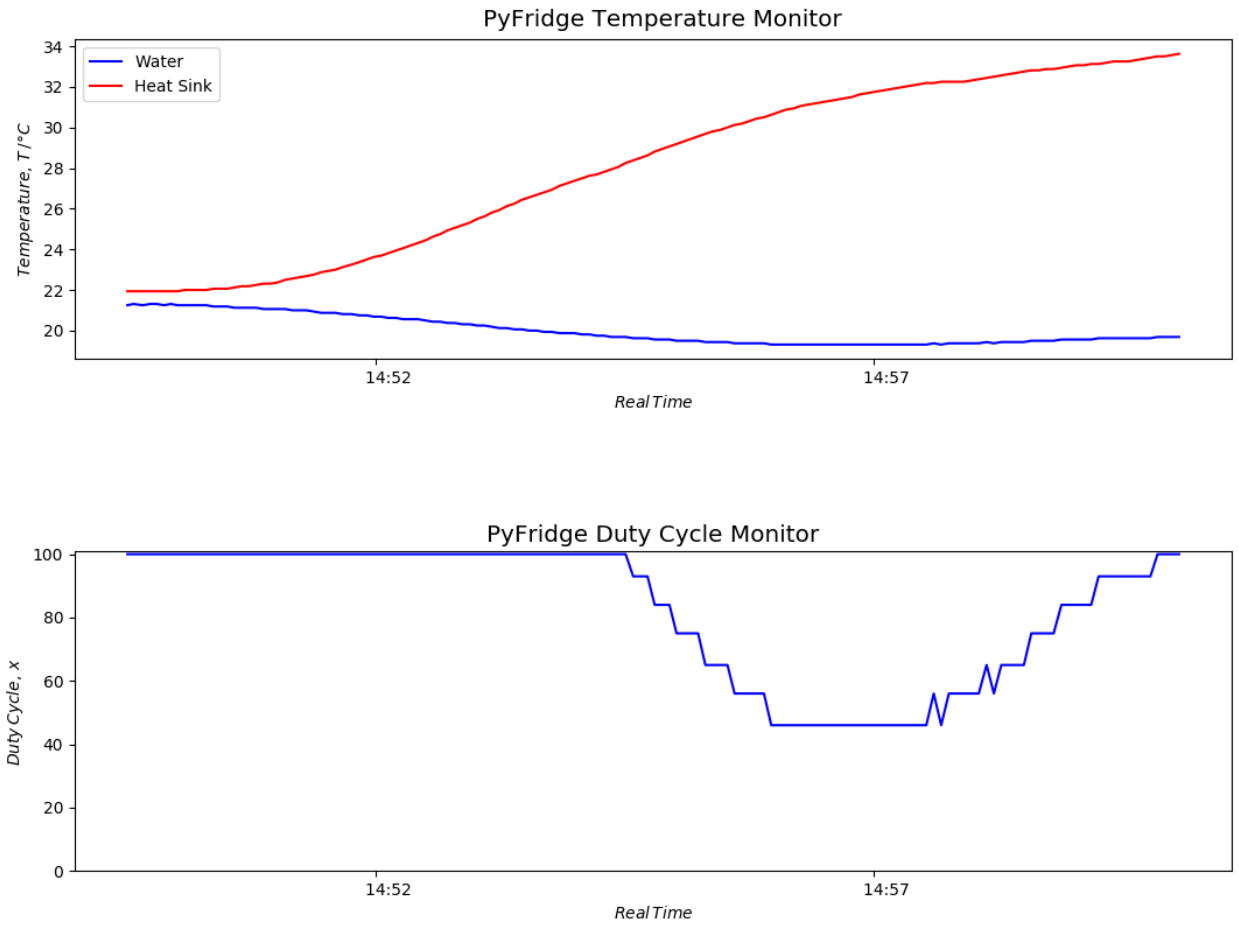



**Figure 11:** Temperature of the refrigerator : 19.0°C

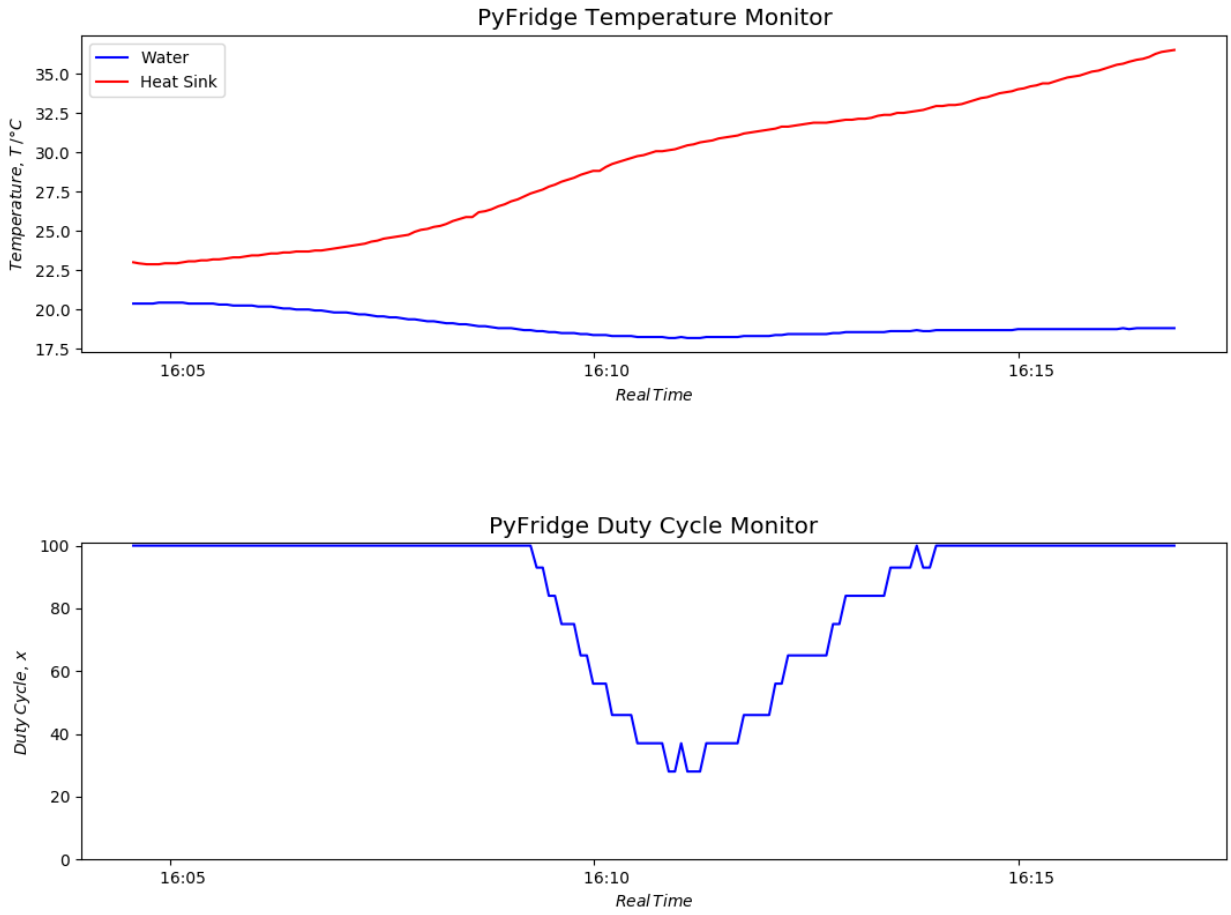



**Figure 12:** Temperature of the refrigerator : 18.0°C