

BuildAThermostat ReflectionLog

(CSE2920 – Challenge) – Stefan Such

How has your program changed from planning to coding to now? Explain.

Planning was seldom needed for this assignment, the exact operation of the thermostat was written out on the phidgets website:

- When using a thermostat, there is a **set temperature**. This is the temperature that the user sets. In your program, this should start at 21°C or 70°F.
- Every time the user presses the green button, increase the set temperature by 1°.
- Every time the user presses the red button, decrease the set temperature by 1°.
- Print the current temperature and set temperature to the screen every 10 seconds.
- If the current temperature is within 2° of the set temperature, turn on the green LED.
- If the current temperature is not within 2° of the set temperature, turn on the red LED.

To start, I copied the declaration, addressing, and opening functions from my HotOrCold assignment and added appropriate methods to declare/address/open the buttons as well.

```
//declaration zone
TemperatureSensor temperatureSensor = new TemperatureSensor();
DigitalOutput redLED = new DigitalOutput();
DigitalOutput greenLED = new DigitalOutput();
DigitalInput redButton = new DigitalInput();
DigitalInput greenButton = new DigitalInput();
double currentTemperature;

//address the "dumb" phidgets
redLED.setHubPort(1);
redLED.setIsHubPortDevice(true);
greenLED.setHubPort(4);
greenLED.setIsHubPortDevice(true);
redButton.setHubPort(0);
redButton.setIsHubPortDevice(true);
greenButton.setHubPort(3);
greenButton.setIsHubPortDevice(true);

//open devices
temperatureSensor.open(1000);
redLED.open(1000);
greenLED.open(1000);
redButton.open(1000);
greenButton.open(1000);
```

The most difficult part of the program is using different polling speeds for the buttons and the sensors. I want the user's button presses to register immediately, So I have to poll the buttons every split second, but the information has to be printed every 10 seconds.

There's one obvious way (at least to me) to do this; Have the loop occur 100 times every second, and on every 1,000th iteration of the loop I could print out the temperature information.

It would probably be smarter to decouple the button polling from the temperature gathering, though.

My button detection code was copied from my UseButtonsAndLEDs assignment, for convenience.

Overall, this assignment wasn't too difficult. It took a bit of my physics knowledge to work with frequencies for the polling speed, though.