Luke Grice

ECE497 IOT

Final Report

**Trinary Automated Home Heating and Cooling**

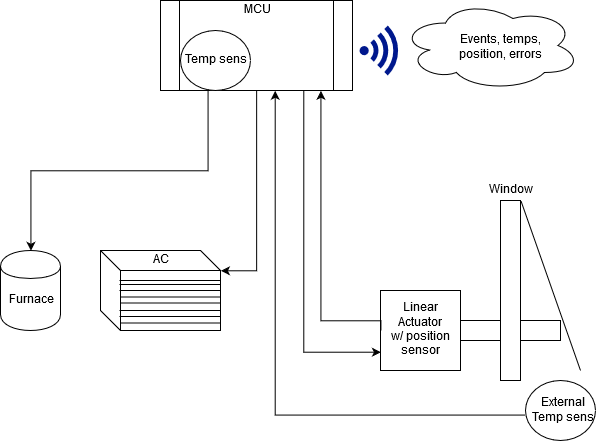
**Project Motivation**

A typical thermostat is set either to cool or to heat. The device works by simply turning on or off the furnace or the A/C to increase or decrease the inside temperature. In order to save electricity, one has to manually open and close windows at certain times to cool or heat the house without the use of the furnace or the AC unit.

**Objectives**

This project uses an outdoor thermistor and an indoor thermistor to determine whether opening the windows will bring the internal temperature to the desired level. If it determines that opening the windows is beneficial, then it opens the window via a linear actuator. Else, the last resort is to turn on either the furnace or the AC.

**Design**

****

*Figure 1: Temperature Control Block Diagram*

From Figure 1 see that the MCU will control the heating and cooling elements along with the linear actuator which will open and close the window. The actuator will return its position and this is how the MCU will know the state of the window. If the window is in the incorrect state the MCU will report an error state. Under normal operation the MCU will compare the internal temperature and the external temperature and determine which method of heating or cooling to use. Table 1 shows the possible states. The connecting lines for inputs and outputs to and from the MCU do not necessarily represent wired connections. It may be possible to wirelessly send the external temperature, and similarly, wirelessly control the various appliances. This will depend on the cost, and ease of access to the controls that would be needed.

*Table 1: MCU states*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Desired Temp** | **Internal Temp** | **External Temp** | **Window** | **Furnace** | **A/C** |
| 20oC | 20oC | NA | Closed | Off | Off |
| 20oC | >20oC | > internal | Closed | Off | On |
| 20oC | >20oC | < internal | Open | Off | Off |
| 20oC | <20oC | > internal | Open | Off | Off |
| 20oC | <20oC | < internal | Closed | On | Off |

To show the logic the MCU must perform, Figure 2 contains a flow chart of how the logic results in each of the possible states. Temperature is checked every 15 seconds. After this, various if statements will be evaluated in sequence to traverse the decision tree down to the correct state. After the desired state is selected an additional check will be performed to ensure that everything responded as expected. If any appliance does not return the commanded state, or no return is seen then the device will report an error. These errors will be represented by error numbers and this will be sent to one of the fields on Thingspeak. The error numbers are as follows:

1. No Error
2. Window Open when should be Closed
3. Window Closed when should be Open
4. Window state unknown
5. Furnace On when should be Off
6. Furnace Off when should be On
7. Furnace state unknown
8. AC On when should be Off
9. AC Off when should be On
10. AC state unknown

After each time the decision tree is traversed the board will upload to the Thingspeak fields:

1 Internal Temp

2 External Temp

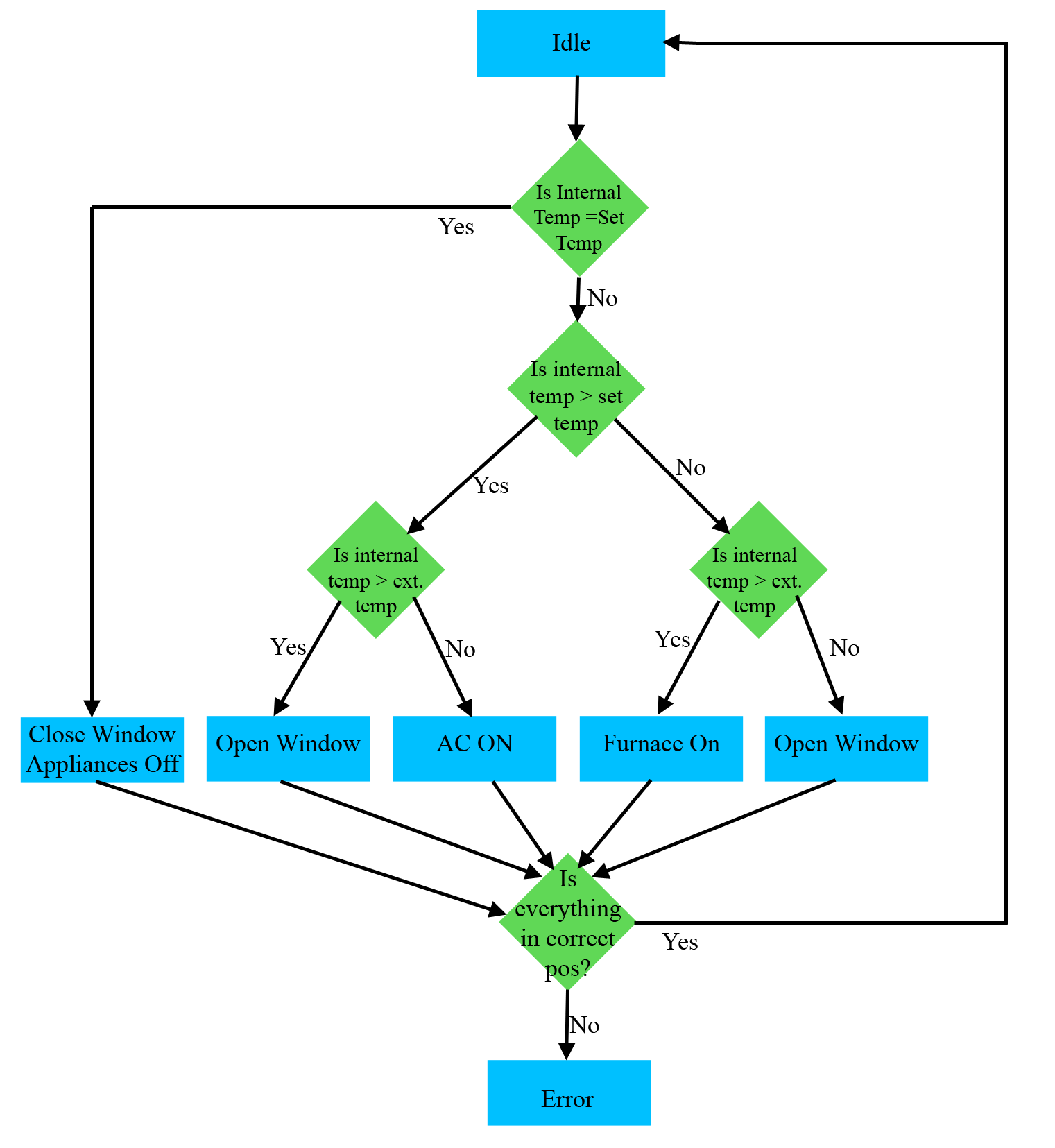
3 State:

0: All Off, 1: Window Open-cooling, 2: Window Open-heating, 3: AC On, 4: Furnace On

4 Error Number

A current sensor might be a good way to check if one of the appliances is running. However, this was not implemented

**Operation**

****

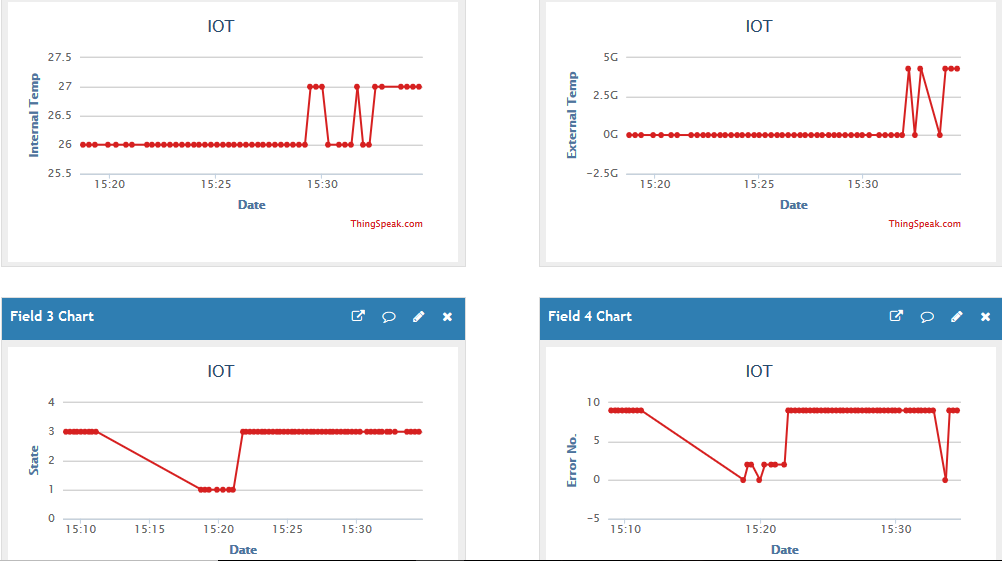
This chain of if statements essentially asks if opening the window will bring the indoor temperature down to or up to the desired level. If the answer is yes, then the window is opened instead of using electricity to alter the temperature. As an additional benefit, when the window is open it will introduce fresh air into the home.

**Results**

Proof of concept was shown using a small wooden “window” made from a hobby-lobby knick-knack shelf. The window was operated via a linear actuator, and for cost purposes the state of the window was queried via a push button connecting two GPIO pins. Originally, a K-type thermocouple was to be used for the external temperature sensor, but it returned impossible values so a second IOT board was used. In the same manner, external temp was requested and sent via the SPI protocol. Power to the AC and Furnace was controlled via a 2-way relay control which was spliced into a 120V power strip.

Ultimately, the main method set up the peripherals, then ran through the logic described in Figure 2. The temperatures were compared, and the appropriate action was taken. When it was hotter than desired inside, and the external temp was cooler than inside the window would open. Else, one half of the power strip’s circuit would close so that power became available to the appropriate half. The top half was labeled AC and the bottom half, Furnace.

Error reporting was done in two ways. During an error state LED2 would light up and an error number would be reported to ThingSpeak (Figure 3). At any time, the home owner could get online and see the state of the heating/cooling. Every 15 seconds the state of the setup and any errors can be seen online. To correct errors, the User push button was setup as an external interrupt and set a local variable. Part of the IF Statement chain is to check this variable, and if true enter a error correction state. In this state the window’s position would be queried, and if incorrect it would be commanded to the correct position. After waiting a period greater than 12 seconds the window position can be queried again, and if now correct the error is cleared. Else, the error condition stays.



*Figure 3: ThingSpeak Field Screenshot*