Letter of Transmittal

Dear Dr. Muresan,

I am pleased to present Group 38's final report on our design solution for the problem of inefficient heating and cooling systems in residential homes. This report represents the culmination of our efforts up to this point, including the development of a working prototype that involves a web app communicating with Arduino microcontrollers.

The report is organized into four sections: problem definition, background information, design methodology, and results and conclusions. We conducted a comprehensive analysis of the problem, utilizing research and data collection to gain a deeper understanding of the issue. We then explained our design approach, which involved developing and refining the prototype and final design. The report provides a detailed description of the selected design solution, including its implementation and operation.

This final report builds upon our previous deliverables, including the proposal, proposal memo, interim report, and interim memo. We have also included a prototype, which demonstrates the functionality of our design solution. Also, a design presentation including our prototype and poster showcased our work to a broader audience of engineering professionals.

We hope that this final report demonstrates the depth and quality of our work and provides valuable insights into the problem of inefficient heating and cooling systems in residential homes.

Thank you for your support and guidance throughout this project.

Sincerely,

Group 38

UNIVERSITY of GUELPH

School Of Engineering

ENGG*41X0: Winter 2022

Final Report

Instructor: Dr. Radu Muresan

Date: April 10, 2023

Liam Fayle (1041711)

Johnathan Toet (1060897)

Seth Harlaar (1109524)

Uday Dhandha (1041130)

Signature Page

In signing this report I certify that I have been an active member of the team and provided approximately equal contribution to the work. I take shared credit and responsibility for the content of this report. I understand that taking credit for work that is not my own is a form of academic misconduct and will be treated as such.

Liam Fayle

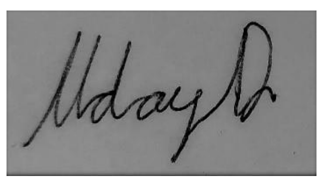
Team Member One Name — Team Member One Signature

Johnathan Toet

Team Member Two Name — Team Member Two Signature

Seth Harlaar

Team Member Three Name — Team Member Three Signature



Uday Dhandha

Team Member Four Name — Team Member Four Signature

Date: April 10, 2023

Executive Summary

Residential energy is a necessity for modern home heating and cooling. Canadians have spent~$18.6 billion on residential energy. While modern homes are well insulated, central air heating is not very efficient. This inefficiency is due to the lack of control outside of temperature and on/off. Currently, some thermostats provide more control with programmable thermostats and smart thermostats.

Programmable thermostats are one of the oldest methods used to control central heating and cooling. These thermostats allow the user to set certain temperatures at predefined schedules set by the user. The schedules may change based on different variables in the user's life as well as the season. While a set schedule may be beneficial for the user, it must be manually changed when something changes in the user's life. This means that programmable thermostats are tedious to use. Additionally, if the user does not change the schedule, the thermostat proves to be inefficient because it sets the temperature at incorrect times. Another current solution is smart thermostats.

Smart thermostats are more flexible and user-friendly than programmable thermostats. These thermostats allow the user to set and change temperatures and schedules from their phone through an app. This makes the operation of the thermostat less tedious. While this is beneficial for users with irregular schedules, it still requires the user to set the schedules and temperatures when there are lifestyle changes. Since smart thermostats are more advanced than programmable, they are more expensive as well. The design solution aims to minimize cost and maximize efficiency.

The chosen design solution is a zoned HVAC control system. This solution involves using automatic registers for actuation, rechargeable batteries for power delivery, and Wi-Fi for communication with the main controller as well as other nodes in the operating environment. The solution combines motion-based and schedule-based control to maximize flexibility and overall efficiency. The final design implements all these features, as well as more customization for temperature setting. A constant mode is built in, where users can choose to have the room set to a constant temperature at all times.

The final design is made up of three major components: the sensor module, the website, and the control module.

The sensor module includes a motion sensor, temperature sensor, an Arduino Nano RP2040, which are all stored in a small plastic box. The Arduino Nano is programmed using Arduino's native programming language. It implements a loop that periodically checks if motion has been detected, then sends the motion data, along with the temperature data, to a website via HTTP POST requests.

The website, which also functions as the user interface for interacting with the system, was built using Python Flask. The website has four modes of operation: constant, motion, schedule, and hybrid. The schedules for both scheduled and hybrid mode are stored in the websites database as JSON for quick access. It also uses a decision tree approach that considers the mode of operation when calculating the error between the target temperature and the actual temperature.

The error calculated by the website is sent to the final part of the design, the control module. The control module is made up of a plastic housing, a servo, logic level converter, and an Arduino Nano RP2040. The Arduino is programmed using the same native programming language and a similar method. It makes HTTP GET requests to the website in a loop, requesting the temperature error. The error is then applied to a discrete PID transfer function to calculate the desired change in angle of the register opening.

Contents

List of Tables

List of Figures

Introduction to the Design
Problem Statement

In Canada a significant amount of money is spent each year on energy in the residential section. In 2017, Canadians spent $30 billion on residential energy [1]. Of these 30 billion dollars, about 62% went to space heating [2], amounting to about $18.6 billion. More specifically, since 52% of all heating systems in 2019 were forced air systems [3], we can say that roughly $9 billion dollars are spent on heating Canadian houses via forced air systems.

Given this amount of spending each year on heating, there must be room for savings. In a study conducted in the United States, it was found that a conservative estimate of 39% of primary residential energy is wasted, with 30.2% of energy wasted due to heating and cooling of unoccupied rooms, heating and cooling of unoccupied houses, and thermostat oversetting [4]. This is a very significant portion of energy that is being wasted and can be claimed back by more efficient usage of Canada's heating and cooling systems.

Some of these savings can be achieved by improving the systems used by Canadians every day where they fall short. One of the largest flaws with forced air systems is the ineffectiveness of centralized control systems. The control systems act as the brain of the furnace. It relies on temperature data to know when to turn on and off. Forced air systems typically have a single thermostat. This often leads to inconsistent and inefficient heating in houses [5]. The controller treats the house like one large room, providing feedback from only one point. This assumption ignores things like elevation, room size, and the natural currents which affect the room's temperature. It doesn't consider that each house layout is unique and is occupied differently [5, 6]. Therefore, there is a significant need to improve the efficiency and flexibility of control systems for forced air heating systems [5, 6].

One method of improving the centralized control systems is to implement zoned control. This means that each room is treated separately with its own thermostat and heating control. It has been shown that there are significant savings available for systems that implement this method, at as low as 11.2% for bungalow houses lived in by a young family with doors open all the time. If the doors were always closed in the same setting, then there is potential for 23% of savings [7]. This study did occur in London and Glasgow, where the climate only reaches a minimum of -10.8 degrees Celsius, and energy demands because of heating can be expected to be lower. If the same zoning control can be applied to all houses in Ontario with the same results, then close to $9.6 billion * 11.2% ~ $1 billion could be saved each year on heating, with potential for ~ $2 billion savings. Along with savings for homeowners, the Ontario government subsidizes part of electricity bills [8]. Reducing the electricity bill for homeowners would alleviate some of the stress on the Ontario government.

The improvement of centralized heating systems will also have positive environmental impacts. In 2019 Canadian homes produced 43 megatons of GHG, not including electricity [9]. With electricity this increases to 63 megatons of GHG [9], representing roughly 10% of Canada's overall GHG emissions [10]. If the same estimate of energy waste found in USA [4] can be applied to Canada, then 63 megatons *0.39 = 24.5 megatons of GHG can be completely avoided just by eliminating energy waste from residential homes. A more efficient system would produce less emissions, lowering Canada's impact on climate change. This would be a great move forward for Canada towards a lower carbon footprint and as an effort to abide by the Rio Agreement.

Along with efficiency implications, there are also comfort implications. Living and working in a poorly regulated temperature environment is not conducive to living a healthy or productive life and can make some areas of the home uncomfortable to use.

Our design solution for this problem must follow some general guidelines to be acceptable. Firstly, since our main goal of this project is to improve the efficiency of centralized forced air systems, we must improve the efficiency. We can test this by using simulations of a model of the original centralized system, and then comparing it to a simulation of the model of our improved, zoned control system.

The user is a very important aspect of this project, since their comfort and survival is the ultimate goal of heating systems in homes. For their comfort and survivability, the system must be able to maintain the temperatures the user desires in the specified zones in their home. This includes having different temperatures for each room. The user must also be able to program the system, so that they can select the desired temperature for each room. The system must provide a better, or at the minimum, equal comfort level from a centralized heating system.

Other significant stakeholders are HVAC companies, which install and service heating systems. Therefore, they would work with any solution. Their business could be impacted depending on the effectiveness of the solution. Renters could also benefit from a design solution that is retrofittable and does not require much house modification, especially since rental units are likely to be the most energy inefficient [11].

Overall, this design solution has potential to positively impact many people, local business, and the government. If our design solution can effectively tackle the problem of inefficient centralized heating systems, then large savings can be acquired.

Objectives

We have three simple design objectives for this project. Our first objective is to design and develop a control system that will be compatible with central forced air heating systems and will utilize automated registers to regulate the flow of hot air to different rooms or zones within a building. The control system will be engineered to integrate seamlessly with the existing heating system infrastructure and function in a user-friendly and efficient manner. Secondly, we aim to design a control system that will significantly reduce the energy consumption of the heating system. We plan to do this with the use of more temperature sensors, motion sensors, and temperature scheduling from the user. The end result will be a more efficient heating system that consumes less energy, thereby reducing energy costs and minimizing environmental impact. Our final objective is to design a control system that provides the user with a higher level of control and customization options. We aim to achieve this by incorporating features such as programmable temperature settings, the ability to control heating zones independently, and a user-friendly interface that enables the user to easily interact with the system.

Background

How the heating systems work

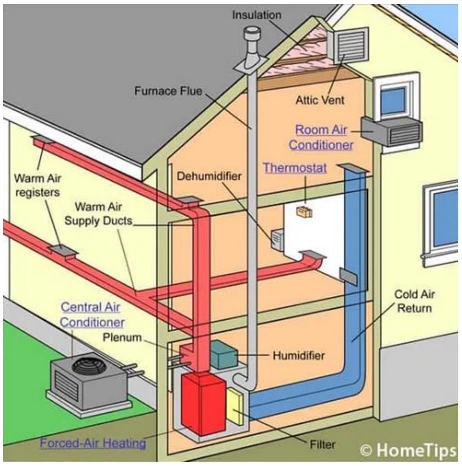


Figure 1 Forced-Air Central Heating System

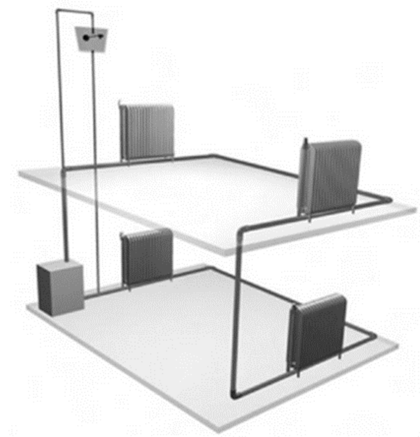


Figure 2 Boiler Central Heating System

The first kind of central heating is a boiler, which heats up water to around 82-90 degrees Celsius [12] via electricity, gas, or oil [13]. The warm water is then distributed through pipes around the house to radiators, a pump is used to circulate the water [14]. Each radiator has an inlet where hot water comes in and a heat exchanger cools the water down while heating the air. The cold water is then returned to the boiler through a return line.

A furnace works by heating air by burning gas or oil. The flames of the burning material heat a metal heat exchanger, which heats the air [15]. In terms of distribution throughout the house, there are two main ways this is done. A gravity furnace simply heats the air and uses the basic physical principle that hot air rises to distribute the air. There is also a cold air return that allows for flow of air through the house. This produces a convection effect. In a forced-air system, there is a fan that blows the air through the ducts, forcing it to move quicker than via convection [16].

A common model of a heating system control that was used in the past can be seen in Figure 3:

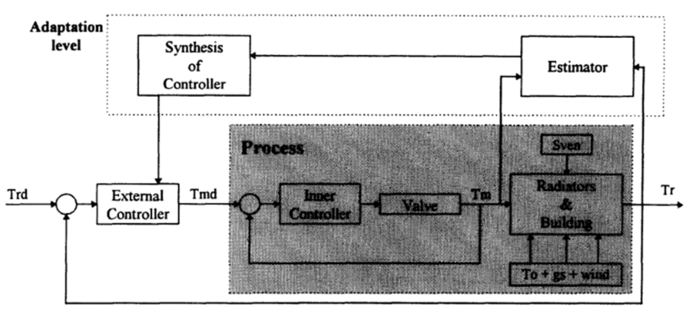


Figure 3 Example of Heating Control System

In some cases, the external controller is a simple PID controller that controls the temperature of the building, where the inner control could be a on-off predictive controller, usually for controlling the input temperature of the transfer mechanism. In the case of central heating systems, the external controller would also be for the room temperature, and the inner control would control the heat of air flowing through the vents [17].

Some more recent models use a PD controller, a PI controller, or a generalized predictive controller [18]. Nowadays there are even systems that use neural networks and fuzzy technology to control heating systems [19, 20].

There are many varieties of heating control systems and one thing they all have in common is that they are controlling the output for the whole building. Typical houses are built with no zone control for heating. If zone control for heating could be implemented, then there could be energy savings on rooms that are not used as much as other rooms but still get heated equally.

Another problem with central heating systems is that they typically only use one or two thermostats which are generally located in the common areas of the household. This is an obvious problem with heating control of houses because if the temperature of a separate room is unknown then there is no way for the system to adjust the input to the room to regulate the temperature. Another issue with this is that the temperature is only local to the thermostat. There could be differences in temperature in opposite ends of large rooms or the household, which leads to inaccurate control.

The larger a building is, the more air there is in the building. The more air there is, the slower the heating of the air will be. This makes the response time of the system that controls the temperature very long, making it hard to control. This is because of the changing conditions during the time between control order and system response time [21]. Other factors that make it hard to test controllers of heating systems, and thus make accurate ones, are things like the need to maintain a comfortable temperature, the changing and current weather conditions, and the duration of experiments required [22].

Residential Energy Consumption

Residential energy consumption is a large part of Canada's total energy consumption. In 2018 it consumed 12% of Canada's total energy usage and produced 65.6 megatons of CO_2 emissions [23]. This is more energy than the agricultural and commercial/institutional sectors combined [23]. Of the 12%, 64% of residential energy was used by Canadians was used for space heating [23]. This means about 7.6% of Canada's total energy consumption was used to heat residential homes.

Natural gas and electricity were the most common energy sources for residential heating [23]. Gas accounted for 53% of energy consumed and electricity accounted for 25% [23]. This is very different from 2^{nd} energy usage which had gas at 30.7% and electricity at 20.2% [23]. Even when compared to residential energy consumption gas accounted for 46% and electricity accounted for 38% [23]. This means that residential heating uses proportionally more natural gas then the average sector.

Canada and Ontario have several programs currently aimed at homeowners to improve residential energy consumption. In January 2023, Canada launched the Canada Greener Homes Grant Program. This program gives homeowners rebates for projects improving home energy efficiency [24]. This includes rebates for smart thermostats and heat pumps.

Enbridge Gas is currently running a Home Efficiency Rebate (HER) program for their customers. The program provides rebates for home improvements including home insulation, changing windows, and upgrading furnaces [25]. Rebates can total up to $5000 but require multiple improvements [25]. Enbridge Gas also launched the Home Efficiency Rebate Plus program in partnership with Canada Greener Homes Grant Program [26]].

Ontario's program is more. Ontario Renovates gives grants and loans to low to moderate income homeowners [27]. This money can be used for a variety of repairs, accessibility changes, and energy efficiency improvements. Specifically, it can be used to repair or upgrade home heating systems [27]. The program is run by cities, municipalities, and counties so exact rebates vary by region [27].

ENERGY STAR Canada is a partnership between the Canadian Government and industry to promote very energy efficient products [28]. ENERGY STAR develops technical specifications for certifying products, such as connected thermostats and furnaces, as efficient [29]. Manufacturers can design a product to meet the specifications. If they meet the specifications and pass inspection, they can use the ENERGY STAR trademark on their product. This trademark provides consumers with an easy way to identify energy saving products [29, 30]. ENERGY STAR Canada also gives annual awards for the companies and institutions that have made outstanding contributions to energy efficiency [31].

Constraints

1. The solution must improve the energy efficiency of the forced air furnace system.
2. The solution must meet legislation requirements. This includes, but is not limited to HVAC, Building, and Electrical code.
3. The solution must be unintrusive and subtle. It shouldn't look out of place in a typical finished home. The system must not require extra space created to store the finished design.
4. The solution must not modify the existing forced air furnace. In our cause modifying the furnace means changing or adding components inside the furnace unit such as fans or heating elements.
5. The solution must be safe. It can not be a fire hazard. It must use safe, non-toxic materials.

Criteria

1. The solution should maximize efficiency. The primary goal of this project is improving the efficiency of the control system. This is measured in energy consumed per year and the dollar cost of heating the home for one year.
2. The solution should minimize cost. Part of the goal is to reduce the heating bills of clients. This is not helpful for the client if it costs significant resources to implement the solution. This is measured in two values. The first is the installation cost of the solution. The second is the yearly cost to maintain the system.
3. The solution should be easy to install for a qualified individual. It should require minimal changes to the house to be installed. By qualified individual we mean someone with experience with maintaining and building residential buildings. This will be rated by the amount of work, special expertise, and time it takes to install and set up the system.
4. The solution should minimize waste. An ideal solution would have a very small carbon footprint. Any materials selected should reflect this criterion. This is rated by the waste produced by making the materials and electronics used to build the solution.
5. The solution should be easy to operate. It should have an intuitive user interface and good user experience. A successful design would be usable by a wide range of people.

Assumptions

We will assume the following is a typical system. This will serve as a baseline for comparing the efficiency of the control system. The building is a single family detached home. The heating system is a forced air furnace powered by gas. It is controlled by a single thermostat in a central location of the building. The thermostat is not programable.

We will assume that the average yearly cost for heating houses is $2100. This is based off Ontario's average amount spent on heating in 2019. We assume that the average home in Canada has 6 rooms, and each room has roughly 2 air supply vents. A room is a space totally enclosed by walls and at least one door to isolate the room. Each house is occupied by 4 residents.

Existing Technology

There are 4 main categories of product that try to solve this problem. These are programmable thermostats, smart thermostats, zoned systems, and occupation prediction models.

Programmable Thermostats

Programmable thermostats are the oldest solution for controlling a furnace. These thermostats generally have a few scheduling options [32]. Generally, users can set a daytime and a nighttime temperature. Some programmable thermostats offer more scheduling flexibility [33]. This includes different schedules for weekdays versus weekends and different schedules for different weekdays.

Programmable thermostats generally have two other modes, called hold mode and override mode. Hold mode turns the thermostat into manual mode [33]. The furnace will always use the temperature as the setpoint when in the mode. Override mode temporarily turn the furnace on manual mode. When the thermostat gets to a change in the schedule it reverts to the schedule [33].

Despite their name programmable thermostats have significant limitations. The user is generally locked into specific schedules [aa]. This means if one of the possible schedules does not match the home's use then it is inefficient. Additionally, the thermostat must be properly programmed to improve the energy efficiency of the system [33].

Smart Thermostats

Smart thermostats are designed to be an improvement over programmable thermostats by providing more flexible control. Smart thermostats have a better user interface such as a phone app or a display with buttons built into the thermostat [32]. Ecobee, Nest, and Honeywell are examples of companies building smart thermostats [33].

Instead of setting temperatures for a predefined schedule, smart thermostats let the user create their own schedule [aa]. This allows the system to accurately reflect the user's schedule and save more energy. This makes it particularly useful for people with irregular schedules [32]. This flexibility makes smart thermostats more expensive than programmable heating systems.

Zoned Heating Systems

Zoning is defined as the "division of a building or groups of buildings into separately controlled spaces" by the American Society of Heating, Refrigerating and Air-Conditioning Engineers. In heating systems this means the building is separated into different rooms or floors [34]. There areas are treated as individual units and controlled separately [34, 35].

Every zone has its own thermostat. The thermostat is used to set the desired temperature and monitor the zone's temperature [35]. The thermostat communicates with a damper. The damper is used to control hot, or cold, air flow into the zone [35]. This allows each zone to be controlled independently.

Zoned systems can potentially cause overpressure ducts. If too many dampers are closed, and the furnace is running air pressure could build up in the ducts [35]. This could damage the ducts and furnace. This means it is important for the zoned system to be able to communicate with the furnace.

Another potential issue is leakage between different rooms [35]. In Ontario, most houses don't have insulation in interior walls. This means rooms with higher temperatures could leak air into other rooms.

Occupation Prediction Models

Occupancy prediction models are machine learning models used to predict when rooms will be occupied [35, 36]. The idea is to predict when a zone is in use and therefore should be heated [ad]. This type of model is implemented by some smart thermostats, but it requires additional hardware besides the thermostat [31].

One problem with the prediction models is preconditioning. When a person enters the zone, it should be at the idea temperature. This means the system must start heating the zone before someone arrives. If the preconditioning starts too soon energy is wasted. If it is too late the user's comfort suffers.

The model collects data by using PIR motion detectors. The detectors are installed in different areas and results are saved. The information can be used to get occupation patterns over time.

Methodology

We developed our solution in several steps. The first step was developing the hardware schematics for the final design. We had already developed rough schematics for the interim report. We considered several potential changes. Our next step was developing the program to control the sensor and control modules. At the same time, we developed the website with our user interface. Finally, we implemented the control algorithm on the website and set up the WIFI communication between the components.

Hardware Development

The first change we considered was using the onboard temperature sensor on the Arduino Nano RP2040 for the sensor module. The Nano RP2040 includes a LSM6DSOXTR Inertial Measurement Unit which includes a temperature sensor. This would have saved money since we wouldn't have had to purchase a separate temperature sensor. It also includes a library which easily lets us read the current temperature instead of writing our own code.

We decided to use the separate temperature sensor for several reasons. The primary reason is the temperature sensor on the board didn't give good feedback. Since the sensor is on the board it would sense the heat generated by the board. This gave an approximately 5-8 degrees Celsius increase in the board's temperature. Since we are trying to accurately control the room's temperature and save money, we need good feedback. Also, the temperature sensor would be inside the modules container so that wouldn't help the sensor's accuracy. Additionally, the cost of a temperature sensor is only a few dollars, so it doesn't significantly increase the cost of the sensor module.

We considered two different types of actuations for the control module. In the interim report we specifically mentioned using a servo. We also investigated using a stepper motor. Stepper motors are good for precise movement. This would allow us to accurately adjust the angle of the vent. The issues with stepper motors are energy efficiency and extra logic required. Stepper motors draw current even when they aren't moving. A stepper motor would also require an H-bridge so we can drive the motor and turn it in both directions. Since we are powering the control module with a battery, and we want the components in the vent to be as small as possible we decided to stick with the servo.

Software Development

We started by developing the sensor module. The code for this module was developed through several iterations. First, we connected the hardware interrupt to the motion sensor. Then we set up a flag called "motionDetected" that was true when motion was detected. In the loop of the code, we added a print message that triggered when the flag was true. Another flag called "motionDetectedInPeriod" was set true.

Next, we added a periodic interrupt so we could check if there was there was motion in the last period. The periodic interrupt set a flag called "periodEnded" to true when triggered. In the loop there is a check for when the period has ended. The issue with this design is there is potential for motion to incorrectly be detected. Specifically, motion could happen at the start of the nth period, and it will be considered active till the check at the end of the nth+1 period. This creates a situation where the room stays active twice as long as it should.

Instead, we decided to save the last time the motion ISR was triggered. This way we can check that the current time minus the last time the motion ISR is less than the room's active time. This is a more accurate implementation of our algorithm.

Finally, we added the code for the temperature sensor. This was much simpler as we just sample the temperature sensor in the loop when the "periodEnded" flag is true.

We went through several different iterations for the control module. Originally, we calculated the servo angle directly from the error. For example, an error of 2 to -2 degrees Celsius resulted in servo angle of 45 degrees. The issue with this is the servo angle was static. Regardless of the air flow and temperature furnace the same angle was given. This could lead to situations where the vent would constantly flip between several different angles instead of approaching a steady state angle. Also, the system only responded to the current feedback.

We improved the design by incorporating a discrete PID controller. This controller allows us to consider the current error, the error summation, and the difference between the current error and the pervious error. The calculated value is entered in the same brackets as before to get the servo angle. This was better as it allowed us to consider error buildup over time and sudden changes in the error. It still had issues as the servo angle was still directly tied to the error.

Our final improvement involved changing from calculating the angle directly from the PID output to using the PID output to make a change to the current angle. This way we could step by step adjust the servo angle, so it produces the desired output. We use the following equation so that an error of 1 degree Celsius resulted in a 3-degree change of the servo angle.

$$Angle\ Error = \frac{PID\ Output}{30} \times 90 = PID\ Output \times 3$$

Finally, we capped the values so the servo angle can be between 0 (fully open) and 80 (mostly closed) degrees. This way we don't overcompensate when trying to control the system. Notice we cap the closing angle at 80 degrees. This way pressure can't build up in the ducts and damage the ducts or furnace.

Once the basic modules and website were developed, we worked on the WIFI communication. This involved deciding on who initiates the request and what information is sent. We decided to module initiate the requests. This way we could use the periodic interrupt to send or get data on a consistent period. The sensor module will periodically send a HTTP POST request to the website. This way the website will be periodically updated with the latest information. We decided to have the sensor module send the current temperature and if the room is active. This way we have the information required to apply the algorithm. The control module will periodically send a HTTP GET request to the website.

Part of deciding what information to send between the components was determining where the algorithm was applied. We chose to have the website responsible for the algorithm. This is primary because we already have the schedule and current room information on the website. This way we don't have to find a way to package all this information, send it to the control module, and then parse it to calculate the temperature error. Instead, we had the website calculate the current temperature error as a float and that is returned when the HTTP GET request is made. This has the added benefit of reducing the bandwidth consumption of the design.

Final Design

Our design had three major components. These are the sensor module, the website, and the control module. The sensor module is responsible for detecting when the room is occupied and providing the room's temperature. The website is the user interface for our design. It allows users to set up their own unique schedule. It is also responsible for implementing the schedule and communication with both the sensor and control modules. The control module is responsible for controlling air flow into the room.

Sensor Module

The sensor module consists of 3 main parts. These are a motion sensor, a temperature sensor, and an Arduino Nano RP2040. Figure 4 shows the schematic diagram for this module. A 6-volt power supply is used to power the circuit. The PIR sensor is connected to digital pin 10. The temperature sensor is connected to analog pin 0.

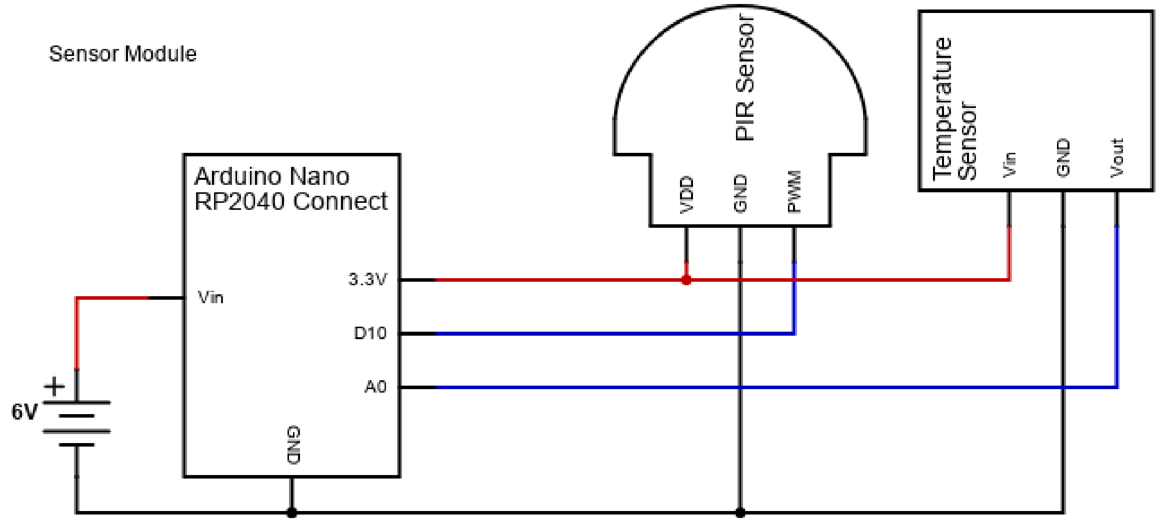


Figure 4. The Sensor Module's schematic diagram.

The components will be stored inside a small plastic box. The motion sensor and temperature sensor will stick out of the box. A small hatch will be on the back to allow the user access to the batteries that power the design.

The hardware is programmed using Arduino's programming language, which is based on C++. In the setup we connect the board to the local WIFI network. Then we set up digital pins 10 and analog pin 0 as inputs. Finally, we set up the interrupts. Figure 5 shows the pseudocode used to setup the sensor module.

```
void setup()
{
  //Connect the Arduino board to WIFI

  //Setup digital pin 10 as an input

  //Setup analog pin 0 as an input

  //Connect digital pin 10 to the MotionISR so it triggers on the rising edge

  //Setup the periodic interrupt so it triggers the PeriodicISR every 20 seconds
}
```

Figure 5. Pseudocode for the Sensor Module's initial setup.

The sensor module uses two interruptions. The first interrupt is a hardware interrupt connected to digital pin 10. This interrupt triggers on a rising edge form the PIR sensor's output signal. When triggered the interrupt service routine (ISR) sets a flag called "roomOccupied" to 1 and saves the current time of the Arduino's internal clock to "lastTimeMotionDetected".

The other interrupt is a periodic interrupt timer. This interrupt triggers every 20s. The ISR sets the "periodEnded" flag to true. This flag is used to tell the board to sample the room's current temperature.

Error! Reference source not found. shows the pseudocode used to implement both routines. We chose to add the minimum amount of code necessary to the routines and have the body handle the flags.

The one exception to this is getting the current time in the motion interrupt service routine. This in the routine because we want to get the current time as close as possible too when the motion sensor is triggered. This way we can accurately check if the motion sensor was triggered in the last 20 seconds.

```
void MotionISR()
{
  //Disable interrupts

  //Set roomOccupied to 1

  //Store lastTimeMotionDetected to current board time

  //Enable interrupts
}

void PeriodicISR()
{
  //Disable interrupts

  //Set periodEnded to true

  //Enable interrupts
}
```

Figure 6. Pseudocode for the motion ISR and the periodic ISR.

In the loop of the program, we check when the "periodEnded" flag is true. When it is we set "periodEnded" to false. After this we perform several steps. First, we check if it has been more than 20 seconds since the last time the motion sensor was triggered. If so "roomOccupied" is set to 0. Next, we sample the room's temperature. We combine this information into a json format. Finally, we send the json to the website via a HTTP POST request. **Error! Reference source not found.** shows the pseudocode for the loop.

```
void loop()
{
  /*
  If (periodEnded)
  {
    Set periodEnded to false

    If (current time - lastTimeMotionDetected > 20 seconds)
    {
      Set roomOccupied to 0
    }

    Set currentTemperature to current values from the temperature sensor

    If connected to wifi
    {
      Combined roomOccupied and currentTemperature values into a json compliant string

      Send the data to the website using an HTTP POST request.
    }
  }
  */
}
```

Figure 7. Pseudocode for the loop function of the Sensor Module.

For design day we developed a basic prototype of this module. Figure 8 shows an image of the implementation and Figure 9 shows an image of the serial output used to shows how the program worked. The prototype was very similar to our final design. The key difference is we used the on-board temperature sensor. This was primarily for simplifying the implementation. We don't recommend this for the final design as the temperature sensor is not accurate.

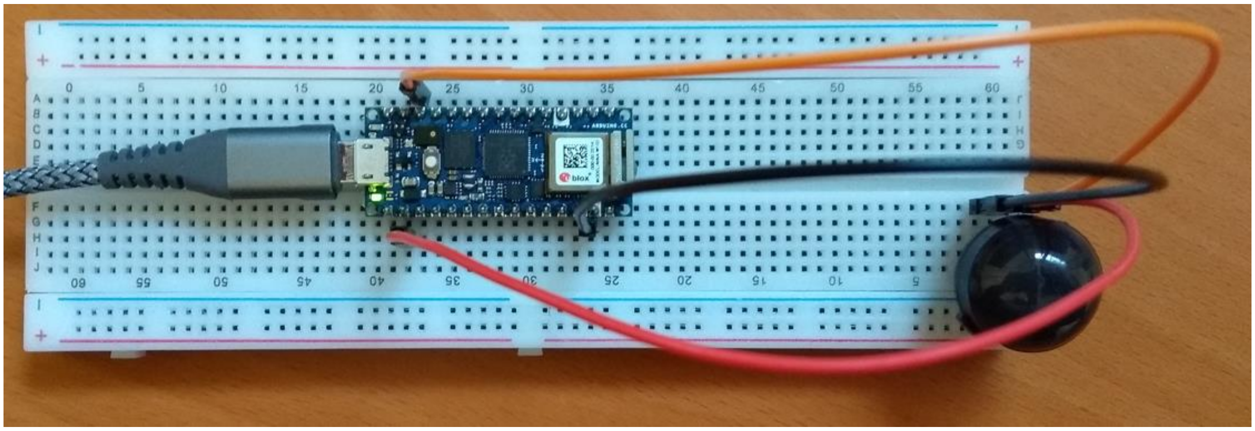


Figure 8. Sensor Module prototype.

```
13:41:06.590 -> Period Ended
13:41:06.590 -> Temperature: 26.37
13:41:06.590 -> Motion detected in last 20000 ms: 1
13:41:06.590 ->
13:41:06.869 -> Making POST request with temperature and motion sensor data
13:41:06.869 -> Data: {"temperature": "26.37", "motion": "1"}
```

Figure 9. A sample serial output from the Sensor Module prototype.

Figure 9 shows a sample output from the prototype. The last print statement shows the data sent to the website. This data is formatted as a json file so the website can easily understand the information.

Website

Our team has developed a user interface for our design using a web application, which we believe will serve the needs of users effectively. The following paragraphs provide a detailed description of the web application's features, its backend and frontend development, and the communication protocol used between the web application and Arduinos.

The backend of the web application was developed using Python Flask. The Flask micro-framework is lightweight and flexible, making it ideal for small to medium-sized web applications. Flask provided us with a good level of abstraction and flexibility when developing the backend of our web application. The web application receives HTTP requests from the Arduino through the "/arduino/sendValues" endpoint. These requests store information about motion detection and temperature levels in a database at the backend. Additionally, the Arduino sends an HTTP request to get the error value (the difference between the current setpoint value selected by the user and the current temperature in the room) through the endpoint "/arduino/getValues". This information is then used to control the HVAC system and make decisions based on the user's selected mode of operation.

For the front-end development of the web application, we used HTML, CSS, and jQuery. HTML is the standard markup language for web applications, and CSS is used to define the application's styling. jQuery is a fast and efficient JavaScript library used to simplify and streamline the front-end development process. The web application provides users with the ability to add and remove rooms and configure each room individually according to its mode of operation and setpoints. The user can select one of the four available modes of operation: constant, motion, schedule, or hybrid.

The constant mode allows the user to set a fixed temperature value for a particular room. This mode is suitable for rooms that do not require any adjustments based on occupancy or time of day.

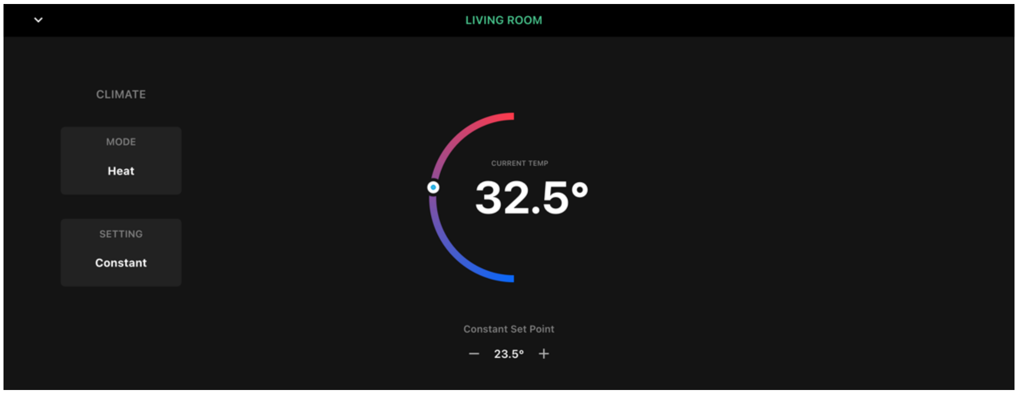


Figure 10. Constant Mode Page

The motion mode allows the user to set both an inactive setpoint and an active setpoint for a particular room. The inactive setpoint is used when the room is not occupied, and the active setpoint is used when the room is occupied. This mode is ideal for rooms that experience varying levels of occupancy throughout the day.

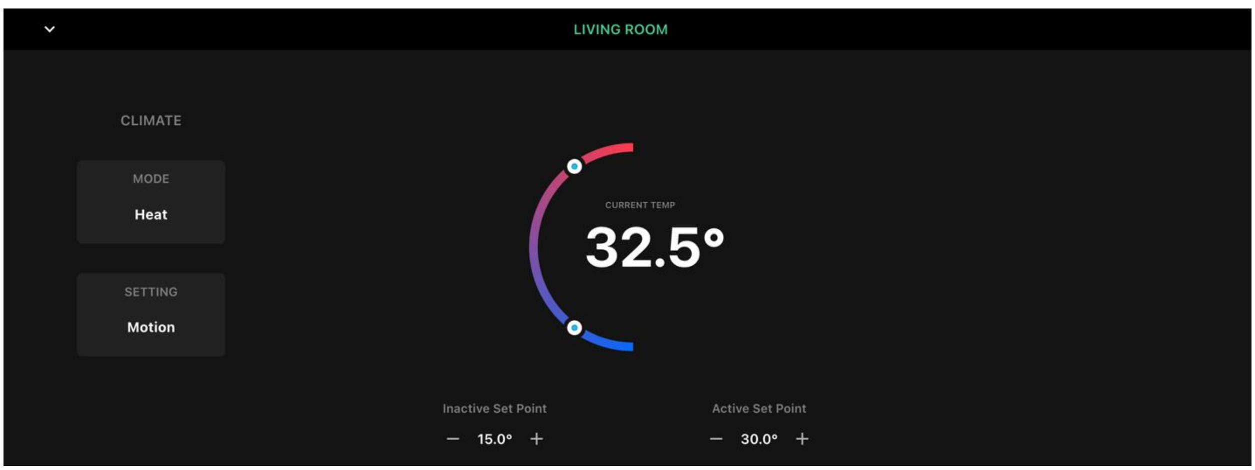


Figure 11. Constant Mode Page

The schedule mode allows the user to schedule fixed setpoint temperature values at different times and days of the week. This mode is suitable for rooms with consistent occupancy.

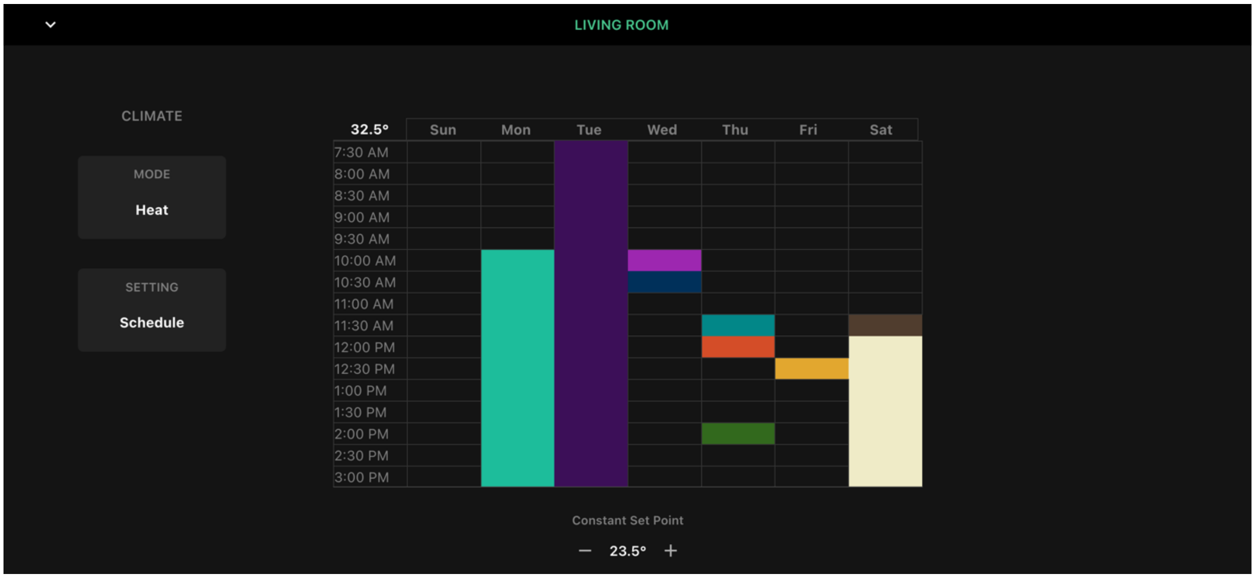


Figure 12. Schedule Mode Page

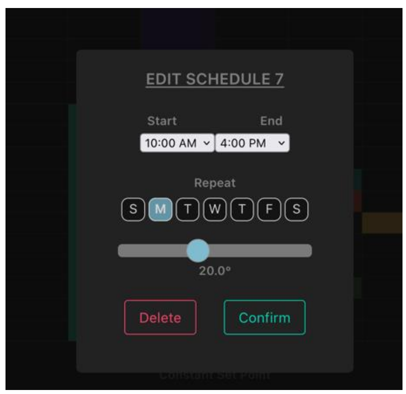


Figure 13. Schedule Mode Input

The hybrid mode is similar to the schedule mode in that it allows users to set different setpoints at various times and days. However, it also allows each scheduled time to have both an inactive and active setpoint like in the motion mode. This mode is ideal for rooms that experience varying occupancy and environmental conditions throughout the day and week.

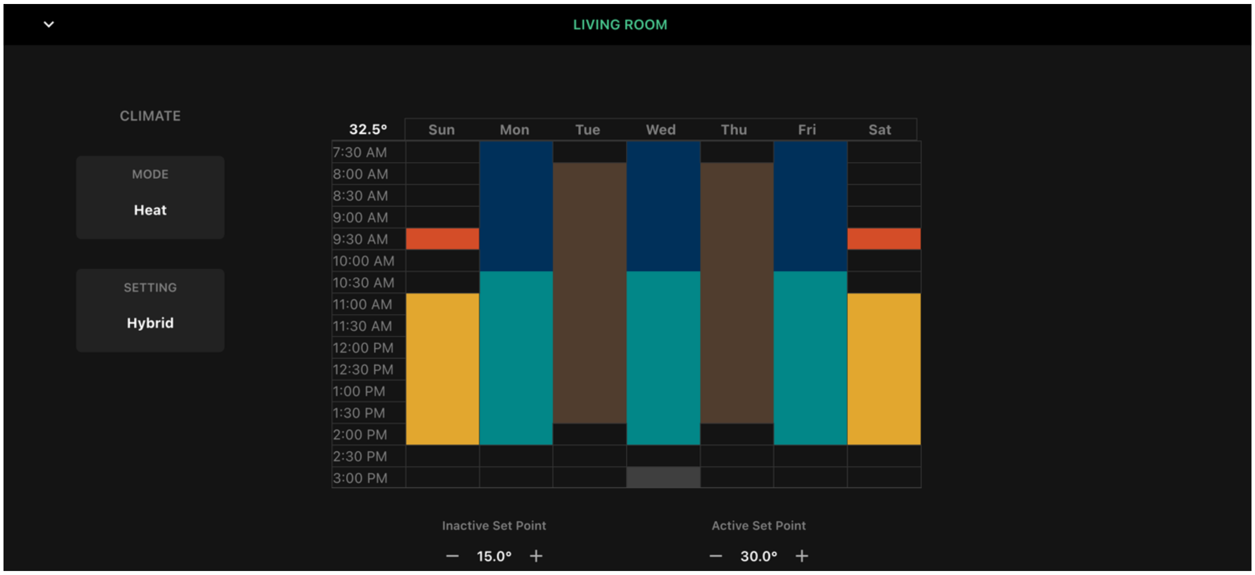


Figure 12. Hybrid Mode Page

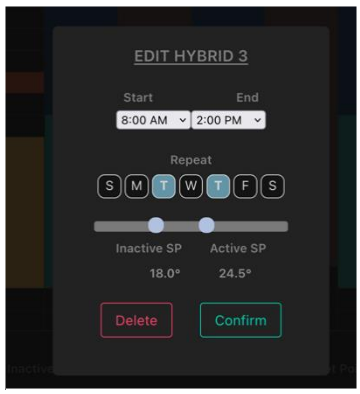


Figure 13. Hybrid Mode Input

The schedules and hybrid schedules are efficiently managed and stored in the web application's database as a JSON object, which allows for quick and convenient access to each schedule's start and end times. By utilizing this storage method, the web application can seamlessly loop over all schedules associated with a specific room to determine which custom setpoints should be applied at any given time. This approach ensures efficient and streamlined management of the room's temperature control and allows for rapid and accurate adjustments to be made based on occupancy and environmental conditions.

The algorithm used to calculate the error value for controlling the HVAC system is a sophisticated decision tree approach that considers the current mode of operation of the room. In the constant mode, the error value is simply computed as the difference between the setpoint and the current temperature. For rooms in motion mode, the algorithm checks the occupancy status of the room by querying the "currentMotion" value stored in the database. If the room is occupied, the error value is calculated as the difference between the active setpoint and the current temperature. If the room is unoccupied, the error value is computed as the difference between the inactive setpoint and the current temperature.

For rooms in schedule mode, the algorithm checks whether there is a custom schedule at the current time. If there is, the error value is calculated as the difference between the custom scheduled setpoint and the current temperature. Otherwise, the standard constant setpoint minus the current temperature is returned. In hybrid mode, the algorithm considers both the custom schedule and the occupancy status of the room. If there is a custom schedule at the current time, the algorithm checks whether the room is occupied. If it is, the error value is calculated as the difference between the custom active setpoint and the current temperature. If the room is unoccupied, the error value is computed as the difference between the custom inactive setpoint and the current temperature. If there is no custom schedule at the current time, the algorithm returns the standard active setpoint minus the current temperature if the room is occupied, and the standard inactive setpoint minus the current temperature if the room is unoccupied.

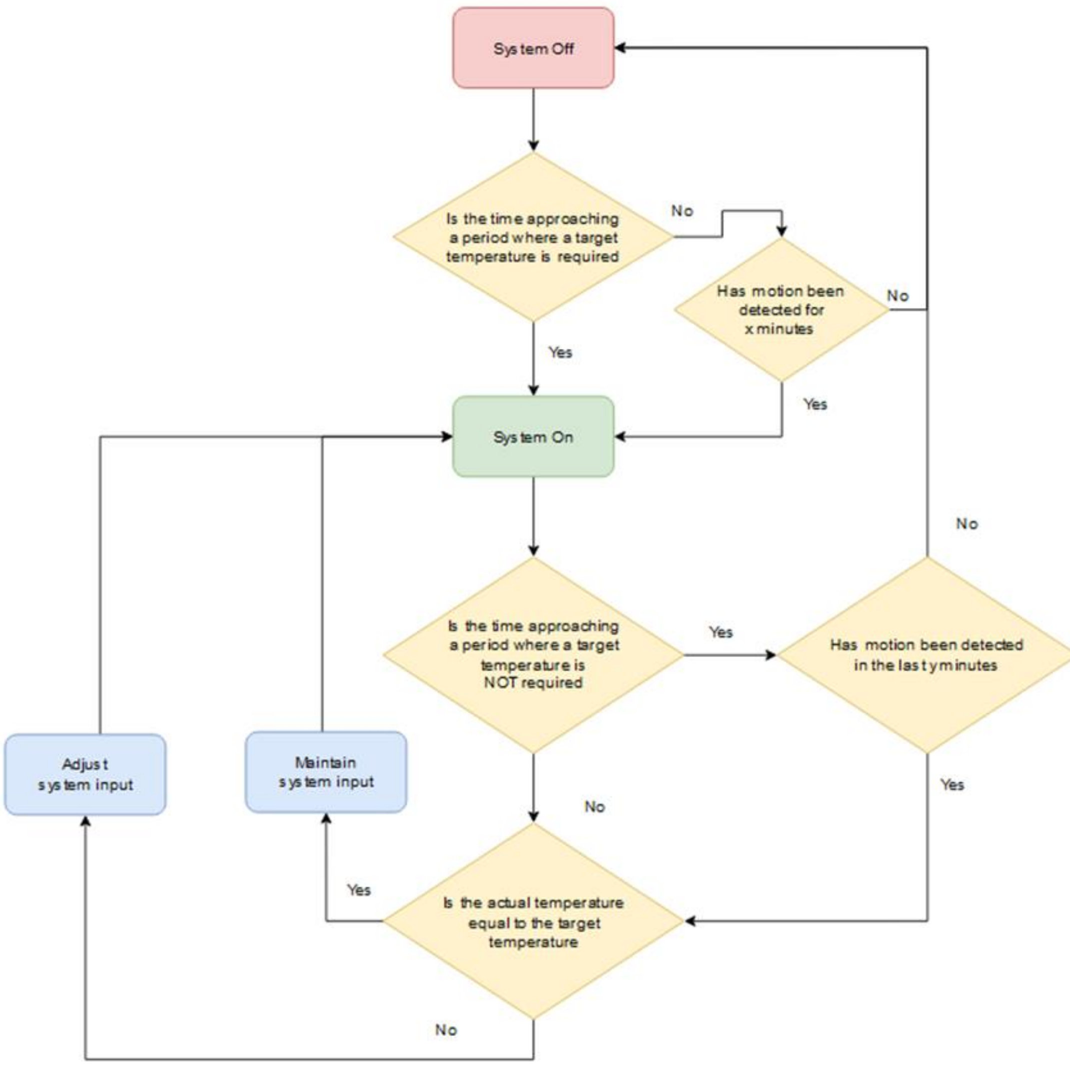


Figure 14. The flowchart used to represent our hybrid control algorithm.

Control Module

The control module has three main parts. These are a logic level converter, a servo, and an Arduino Nano RP2040. Figure 15 shows the schematic diagram for this module. A 6-volt power supply is used to power the circuit. The digital output is connected to digital pin 10. This signal goes through the logic level converter, so it changes the signal from a V_{DD} of 3.3 volts to 6 volts. This is required since we were unable to find a servo with a V_{DD} of 3.3 volts.

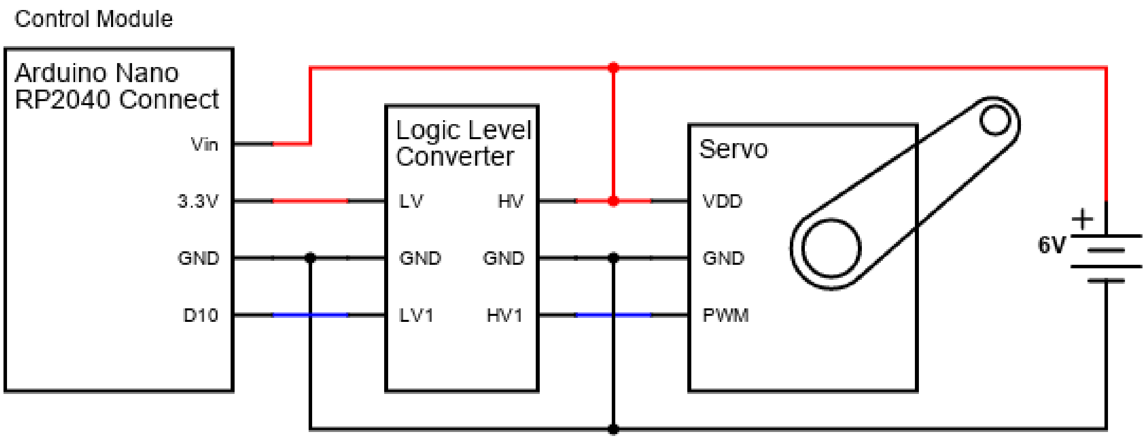


Figure 15. The Control Module's schematic diagram.

The setup and ISR pseudocode for the control module is very similar to the pseudocode for the sensor module. In the setup we connect the board to the local WIFI network. Then we set up digital pins 10 as an output. Finally, we set up the periodic interrupt. Figure 17 shows the pseudocode used to setup the sensor module. Figure 16 shows the pseudocode used to implement the periodic ISR.

```
void setup()
{
  //Connect the Arduino board to WIFI

  //Setup digital pin 10 as an output

  //Setup the periodic interrupt so it triggers the PeriodicISR every 20 seconds
}
```

Figure 17. Pseudocode for the Control Module's initial setup.

```
void PeriodicISR()
{
  //Disable interrupts

  //Set periodEnded to true

  //Enable interrupts
}
```

Figure 16. Pseudocode the control sensor's periodic ISR.

The loop pseudocode is the major difference between the control module and the sensor module. When the "periodEnded" flag is true the module makes a HTTP GET request to the website. In response it gets the error between the room's current temperature and the desired temperature. We take this error and apply it to a discrete PID transfer function shown below. In the equation K_p, K_i, and K_d are the system equations. S is the sum of all errors from period n to period 0. E[n] is the current temperature and e[n-1] is the previous error.

$$PID\ Output = K_p \times e[n] + K_i \times s + K_d \times (e[n] - e[n-1])$$

Once we have the PID output we convert it into a change in the servo's angle. We add this error to the current angle and confirm it is between 0 and 80 degrees where 0 degrees is fully open, and 90 degrees is fully closed. If it isn't we This angle opens or closes the vent, affecting air flow out of the vents. We chose to never fully close the vent, so we don't have air pressure buildup in the vents. This could potentially damage the furnace and ducts. We developed the following equation to calculate the angle. We divided it by 30. This allows us to slowly adjust the servo by 3 degrees for every degree Celsius of temperature error. Shows the pseudocode for the loop. Shows the pseudocode for the loop of the control module.

```
void loop()
{
  /*
  If (periodEnded)
  {
    Set periodEnded to false

    If connected to wifi
    {
      Get the current temperature error from the website using an HTTP GET request.

      Calculate the new temperature error sum

      Calculate the PID output using the PID equation

      Calculate the angle error using the PID output

      Calculate the servo angle by adding the angle error to the current angle

      If angle > 80
        Set angle to 80

      If angle < 0
        Set angle to 0

      Apply servo angle to the servo

      Set the previous temperature error to the current error
    }
  }
  */
}
```

Figure 18. Pseudocode for the loop function of the Control Module.

$$Angle\ Error = \frac{PID\ Output}{30} \times 90 = PID\ Output \times 3$$

We chose to use a PID transfer function for several reasons. First, we are constantly providing feedback to the system. Therefore, a closed loop control system is effective for our desired results. Second, since we are getting the temperature error from the website, it is easy to implement the function. Finally, we have experience working with PID transfer functions, particularly in temperature systems.

Since each room is unique in terms of size, layout, and number of vents the K_p, K_i, and K_d values must be calibrated for the system. The best way to do this is an open loop calibration. The steps are outlined below.

1. Completely open the vents in the room
2. Let the room's temperature settle to a constant value.
3. Half close the vents, so the angle is 45 degrees.
4. Monitor the data to determine the K, T_D, and τ.
5. Use K, T_D, and τ to calculate K_P, K_I, and K_D

Figure shows the output for our prototype. We couldn't completely implement our final design as we didn't have the logic level converter or servo to show the output. Instead, we setup the Arduino to output the current temperature error and the calculated servo angle.

```
13:53:00.219 -> Making GET request for temperature and setpoint
13:53:00.642 -> Current error: -5.20 degrees Celsius
13:53:00.642 -> Servo Angle at 62 degrees
```

Figure 19. A sample serial output from the Control Module prototype.

Results and Discussion

We were only able to build a partial prototype of our final design. This made it hard to evaluate our design. From our research we know that zoned heating can provide savings of 23% with 10% savings in the worst case. Therefore, we expect to save $210 yearly in the worst case and $483 yearly in the best case. We expect our design to be more efficient than normal zoned heating systems since the zones are only heated when occupied.

In the Bill of Materials, we calculated the cost to build each module. Assuming 2 control modules and 1 sensor module per room we expect the cost per room to be $288.04. Therefore, we expect the cost to be $1728.24 for an average house. This is a significant cost when compared to the yearly savings. In the worst case it will take roughly 9 years for just material costs to be paid back. In the best case it will take roughly 4 years.

Like a smart thermostat, the exact savings are ultimately up to the user and their lifestyle. This includes correctly using the schedule and what temperature levels the user is comfortable with. Therefore, it is important the final design includes a clear manual explaining the benefits of programming the solution. The manual should include examples showing how the solution should be programmed to get the most out of their system. It must be clear, so the end user understands the benefits of properly setting the schedule to match their lifestyle.

For example, a user who uses a lower inactive temperature will save more money than someone with a higher inactive temperature. Also, someone who has an inconsistent schedule will benefit more from the hybrid schedule.

Consider the example below. In this case we have a user who is inconsistent with moving between three rooms in a house. Some days they may spend all their time in room 1. On other days they move between the 3 rooms constantly. With a traditional smart thermostat all three rooms would be heated at the desired temperature. The hybrid algorithm allows us to only heat one room to the active point. This allows us to put the other two rooms into an inactive state and use less energy to heat them. Assuming a temperature of 25 for the scheduled and active rooms and 20 for unused rooms, we could reduce the electricity bill. In general, a change of 1 degree Celsius saves you 2% of the energy bill. The hybrid schedule changes the average temperature from 25 degrees to 21.67 degrees. This reduces the energy bill by roughly 6.6%. If we decrease the inactive temperature to 15 degrees, the energy bill could be reduced by 40%, just by reducing the average temperature of the house.

Table 1. A table showing how rooms are heated for a traditional schedule and our hybrid schedule algorithms.

		Schedule Based			Hybrid based		
Time	Occupying	Room 1	Room 2	Room 3	Room 1	Room 2	Room 3
10:00	Room 1	On	On	On	On	Off	Off
10:30	Room 1	On	On	On	On	Off	Off
11:00	Room 2	On	On	On	Off	On	Off
11:30	Room 2	On	On	On	Off	On	Off
12:00	Room 3	On	On	On	Off	Off	On
12:30	Room 1	On	On	On	On	Off	Off
1:00	Room 1	On	On	On	On	Off	Off
1:30	Room 1	On	On	On	On	Off	Off

A more realistic situation is a home with 6 rooms and 4 residents. In the worst case each resident occupies a different room such as the scenario in Table 2.

Table 3 shows we still save energy by only heating 4 of the 6 rooms to the active temperature. Assuming an active temperature of 25 degrees and an inactive temperature of 20 degrees we get an average temperature of 23.33 degrees. This would give savings of roughly 0.033%. With an inactive temperature of 15 degrees there could be savings of roughly 6.67%.

Table 2. Example occupancy pattern for a house with 6 rooms and 4 residents.

	Occupying			
Time	Resident 1	Resident 2	Resident 3	Resident 4
10:00	Room 1	Room 2	Room 3	Room 4
10:30	Room 6	Room 2	Room 3	Room 4
11:00	Room 1	Room 5	Room 3	Room 4
11:30	Room 1	Room 2	Room 3	Room 4
12:00	Room 1	Room 2	Room 3	Room 6
12:30	Room 1	Room 2	Room 5	Room 4
1:00	Room 5	Room 2	Room 3	Room 4
1:30	Room 1	Room 2	Room 3	Room 4

Table 3. Which rooms are active based on the occupancy pattern in table 2.

	Hybrid based					
Time	**Room 1**	**Room 2**	**Room 3**	**Room 4**	**Room 5**	**Room 6**
10:00	On	On	On	On	Off	Off
10:30	Off	On	On	On	Off	On
11:00	On	Off	On	On	On	Off
11:30	On	On	On	On	Off	Off
12:00	On	On	On	Off	Off	On
12:30	On	On	Off	On	On	Off
1:00	Off	On	On	On	Off	Off
1:30	On	On	On	On	Off	Off

Socially, there are benefits to our design. Unlike the normal smart thermostat this solution does a better job automatically adapting to the user's schedule. This is because the motion sensor will automatically heat the room to the active setup point when it is occupied. It is not locked into a rigid structure like a traditional smart thermostat. It can adapt and change the system without being reprogrammed.

The zoned heating lets users customize the system to their needs. A poorly regulated environment is not conducive to a healthy, productive life. By allowing each room to be controlled separately each room can be customized based on the environment and the user's needs.

Bill of Materials:

The list of material below is the required material to build the final design.

Sensor Module:

The total cost for the sensor module is $97.46 with tax. Exact amounts and supplies for each item are listed in the Appendix in Table 5.

1. 1 Arduino NANO RP2040 Connect
2. 1 PIR motion sensor
3. 1 Temperature Sensor
4. 1 4 AA battery holder
5. 4 AA batteries
6. Simple plastic case

Control Module:

The total cost for the control module is $95.29 with tax. Exact amounts and supplier for each item are listed in the Appendix in Table 6.

1. 1 Arduino NANO RP2040 Connect
2. 1 SG90 Servo
3. 1 Logic Level Converter
4. 1 4 AA battery holder
5. 4 AA batteries
6. 1 Custom vent cover

Life Cycle Impact Assessment

This life cycle impact assessment was completed using the Eco-indicator 99 method and is a cradle-to-grave assessment. We used the process defined by the Ministry of Housing [bb].

We considered the production, use, and disposal of the system. We did the calculations based on the assumption that each room would require 2 control modules and 1 sensor module. Thus, the life cycle impact assessment is per room where the system is used.

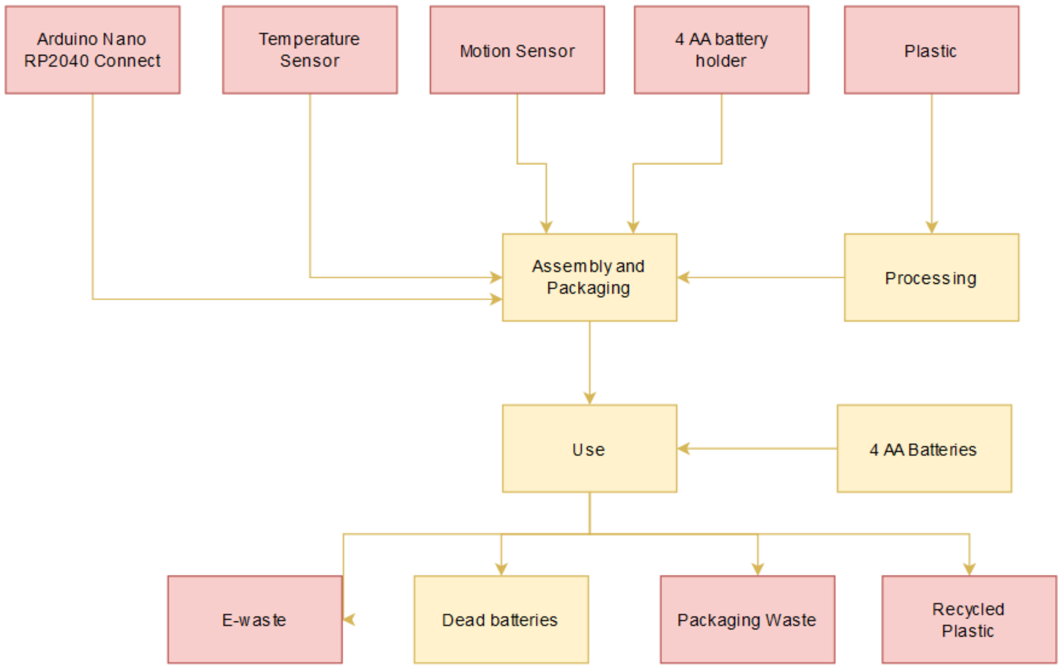


Figure 20. Life Cycle Process for Sensor Module

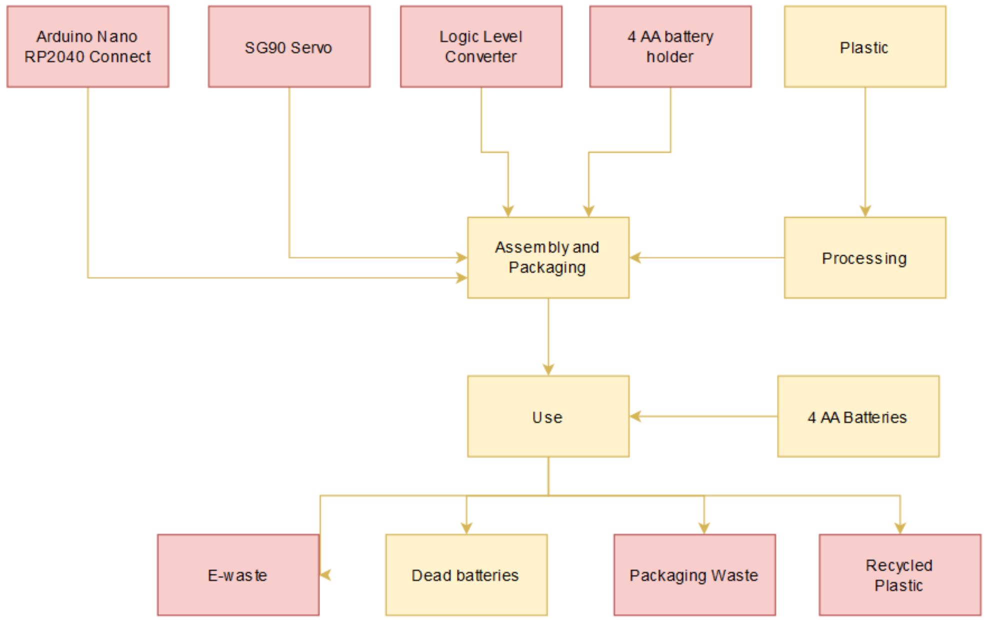


Figure 21. Life cycle Process of Control Module

This life cycle impact assessment was completed using the Eco-indicator 99 method and is a cradle-to-grave assessment. We used the process defined by the Ministry of Housing [bb].

We considered the production, use, and disposal of the system. We did the calculations based on the assumption that each room would require 2 control modules and 1 sensor module. Thus, the life cycle impact assessment is per room where the system is used.

The size of a design will significantly reduce its overall environmental impact when we consider the size of the necessary parts of the design. For example, 1 Arduino Nano RP2040 weighs only 6 grams and would likely have an indicator somewhere on the order of 300-600 for production, none for use, and somewhere between 3-5 for landfill. Doing a quick calculation, we can see that a total of 0.006*(450 + 4) = 2.724 mPt. This is 0.24% of the total mPt calculated without the Arduino. Thus, we can safely ignore it. However, there are still factors to consider during production, use, and disposal. The plastic casings and batteries are the most reasonable parts to evaluate. During production, the molding of the plastic and the impacts of packaging must be considered. During use, the impact of recharging a battery 400 times should be considered. Finally, during disposal, the impact of dead batteries going to the landfill must be addressed. Despite the small size of the device, these factors still contribute to its overall environmental impact. Therefore, it's crucial to consider them throughout the product's lifecycle to minimize its negative impact on the environment. Other factors like the recycling of the plastics used in the design and packaging will lead to an equal and opposite reduction in total mPt, meaning we can safely ignore them as well, while still considering their processing.

Table 4 Eco-indicator 99 Calculations for Life Cycle Impcat Assessment

Process/Material	Amount	Indicator	Result
Plastic molding	2*0.25kg g + 1*0.05kg	44 [bb]	24.2
Packaging Carton	2*0.25kg g + 1*0.05kg	69 [bb]	37.95
Recharge Batteries	400*0.025kWh*4	26 [bb]	1,040
Battery Disposal	4*AA NiMH battery	0.8 [ba]	3.2
Total [mPt]			1,105.35

For these values, we used estimations for plastic use and packaging. For plastic molding, we estimated the housing for control modules to be 250 grams, and the housing for the sensor module was estimated at 50 grams. For packaging, we assumed that each module would be sold separately and that would require 1 packaging carton per module. The indicator values for each row were taken from their respective sources.

From

Table 4, we can see that the total mPt produced by the system is 1,105.35 mPt. The majority of this impact comes from the charging of batteries. In reality, it is unlikely that the user would use the system for 400 charges, and more likely that the batteries will reach their 10-year guaranteed lifetime [bc]. The impact of throwing out the battery instead of recharging it each time would be much greater than recharging.

Conclusions

In conclusion, the proposed design solution proves to be an exceptional solution for the objectives in place for the project. The objectives were that the design must be user-friendly, efficient, and retrofittable with the current HVAC system, it must significantly reduce energy costs, and it must provide a higher level of customization than the current thermostat.

The design solution is user-friendly, efficient, and retrofittable. The proposed system is user-friendly since there is no need for extensive technical knowledge to set up the system. It is efficient since it is powered by battery, and it is low power, so a long lifespan is expected. Finally, it is retrofittable because it can be installed in existing registers in a home.

The design solution can significantly reduce the energy costs in a user's home. The prototype was not implemented, but research shows that closing registers in unused zones can reduce the energy bill between 10% and 23% depending on how the user chooses to customize it.

Finally, the design solution is very customizable with the website. The website, which interfaces with the control board, has three different modes: schedule, motion, and hybrid mode. These modes can be chosen by the user depending on their lifestyle. The schedule mode allows the user to set different temperatures for every hour of every day of the week for each room or zone. The motion mode only opens the register in the zone where motion is detected, saving the user the time it takes to set up a schedule. The hybrid mode combines the schedule and motion modes.

Overall, the design solution effectively achieves the objectives set out for the project.

Recommendations

There are a few different considerations for the future development of this project. Firstly, a complete prototype must be developed to assess its strengths and weaknesses in its specific application. Once the weaknesses are specified, the reduction in cost per room can be investigated. The ESP8266 board can be used to reduce the total cost. This board is significantly less expensive than the current Arduino NANO RP2040 with all the required capabilities for the system. Additionally, the ESP8266 is well documented in many open-source projects, which means there is plenty of ongoing support. With the development and testing of the complete prototype, a more detailed estimate can be made of the energy cost savings for the user. A complete prototype is a significant part of further testing of the feasibility of this design solution.

Along with a complete prototype, other potential applications can be investigated as well. Firstly, the focus of this design solution was residential homes which have preexisting HVAC systems in place. While this may be the most cost-effective method of implementing the design, the system can be integrated deeper into the HVAC system in a residential home that has not been fully built yet. This deeper integration can be done by creating a communication interface between the zoning system and the furnace, allowing more control. Furthermore, the commercial application can be investigated for its feasibility. The main obstacle in commercial applications is the large variety of commercial buildings. This means that the system must be very customizable to accommodate for a larger variety of building types.

References

1. N. R. C. Government of Canada, “Energy Use in Canada: Trends Publications,” Sep. 01, 2011. https://oee.nrcan.gc.ca/publications/statistics/trends/2017/residential.cfm (accessed Oct. 15, 2022).

2. N. R. Canada, “Heating equipment for residential use,” Dec. 05, 2013. https://www.nrcan.gc.ca/energy-efficiency/products/heating-equipment-for-residential-use/13740 (accessed Oct. 15, 2022).

3. S. C. Government of Canada, “Primary heating systems and type of energy,” Mar. 10, 2015. https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=3810028601 (accessed Oct. 15, 2022).

4. R. J. Meyers, E. D. Williams, and H. S. Matthews, “Scoping the potential of monitoring and control technologies to reduce energy use in homes”, *Energy and Buildings*, vol. 42, no. 5, pp. 563–569, May 2010, doi: 10.1016/j.enbuild.2009.10.026.

5. J. A. Jornet-Monteverde and J. J. Galiana-Merino, “Low-Cost Conversion of Single-Zone HVAC Systems to Multi-Zone Control Systems Using Low-Power Wireless Sensor Networks,” *Sensors*, vol. 20, no. 13, p. 3611, Jun. 2020, doi: 10.3390/s20133611.

[6. W. Guo and M. Zhou, "Technologies toward thermal comfort-based and energy-efficient HVAC systems: A review," *2009 IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 3883-3888, doi: 10.1109/ICSMC.2009.5346631.

7. J. Cockroft, A. Cowie, A. Samuel, P. Strachan, “Potential energy savings achievable by zoned control of individual rooms in UK housing compared to standard central heating systems,” *Energy and Buildings*, Volume 136, pages 1-11, February 2017

8. S. C. Government of Canada, “Residency ownership of residential properties by property type and period of construction, provinces of Nova Scotia, Ontario and British Columbia,” Mar. 12, 2019. https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=4610001801 (accessed Nov. 04, 2022).

[9]. N. R. C. Government of Canada, “Residential Sector - GHG Emissions,” Jan. 30, 2020. https://oee.nrcan.gc.ca/corporate/statistics/neud/dpa/showTable.cfm?type=AN&sector=aaa&juris=00&rn=2&page=0 (accessed Oct. 28, 2022).

10. E.N.R Government of Canada, “Greenhouse gas emissions – Canada.ca,” 2022. https://www.canada.ca/en/environment-climate-change/services/environmental-indicators/greenhouse-gas-emissions.html (accessed Feb 15, 2023).

11. J. Curtis, N. Devitt, and A. Whelan, ‘Using census and administrative records to identify the location and occupancy type of energy inefficient residential properties’, Sustainable Cities and Society, vol. 18, pp. 56–65, Nov. 2015, doi: 10.1016/j.scs.2015.06.001.

12. S. Kubba, “Chapter 9 – Impact of Energy and Atmosphere,” in *Handbook of Green Building Design and Construction*, 2nd, S. Kubba, Ed. Amsterdam: Butterworth-Heinemann, 2017, 443-571

13. RANSHAW. How Does a Boiler Work? “How Does a Boiler Work? A New York Tech Explains : Ranshaw.” https://ranshaw.com/help-guides/tips/how-does-a-boiler-work (accessed Oct 29, 2022).

14. P. Mitchel, “Heating Circuits,” *Central Heating, Installation, Maintenance and Repair*, Great Britian: Brailsford Press, 2008, 13-36

15. “Types of Heating Systems | Smarter House.” https://smarterhouse.org/heating-systems/types-heating-systems (accessed Oct 29, 2022).

16. D. Vandervort, “How Central Heating Works.” https://www.hometips.com/how-it-works/forced-air-heating-systems.html (accessed Oct 29, 2022).

17. M. García-Sanz, “A reduced model of central heating systems as a realistic scenario for analyzing control strategies,” *Applied Mathematical Modelling*, Volume 21, Issue 9, pages 535-545, Sept 2019

18. S. Ogonowski, “Modelling of the heating system in small building for control,” *Energy and Buildings*, Volume 42, Issue 9, pages 1510-1516, Sept 2010

19. N. Morel, M. Bauer, M. El-Khoury, J. Krauss, “NEUROBAT, A PREDICTIVE AND ADAPTIVE HEATING CONTROL SYSTEM USING ARTIFICIAL NEURAL NETWORKS,” *International Journal of Solar Energy*, Volume 21, Issue 2-3, pages 161-201, Mar 2007

20. F. Mendi, K. Boran, M. K. Kulekci, “Fuzzy controlled central heating system,” *International Journal of Energy Research*, Volume 26, Issue 15, pages 1313-1322, Dec 2002

21. M. García-Sanz and J. Flórez, “A Los-Cost Control Experiment for Systems with Large Time Constants and Unknown Time Delays,” *The International Journal of Electrical Engineering & Education*, Volume 31, Issue 3, pages 252-263, 1994

22. N. R. Canada, “Keeping the Heat In,” Natural Resources Canada, Mar. 2014. Accessed: Nov. 04, 2022. [Online]. Available: https://www.nrcan.gc.ca/energy-efficiency/homes/make-your-home-more-energy-efficient/keeping-the-heat/15768

23. N. R. C. Government of Canada, “Energy Use in Canada: Trends Publications,” Sep. 01, 2011. https://oee.nrcan.gc.ca/publications/statistics/trends/2018/totalsectors.cfm (accessed Feb. 16, 2023).

24. “Canada Greener Homes Grant.” https://natural-resources.canada.ca/energy-efficiency/homes/canada-greener-homes-initiative/canada-greener-homes-grant/canada-greener-homes-grant/23441 (accessed Feb. 16, 2023).

25. “Home Efficiency Rebate Plus | Enbridge Gas.” https://www.enbridgegas.com/residential/rebates-energy-conservation/home-efficiency-rebate-plus (accessed Feb. 16, 2023).

26. “Home Efficiency Rebate | Enbridge Gas.” https://www.enbridgegas.com/residential/rebates-energy-conservation/home-efficiency-rebate#rebates (accessed Feb. 16, 2023).

27. “Homeownership Funding - Housing - Wellington County.” https://www.wellington.ca/en/social-services/hs-homeownership-funding.aspx#Ontario-Renovates (accessed Feb. 16, 2023).

28. N. R. Canada, “about,” Jul. 09, 2019. https://natural-resources.canada.ca/energy-efficiency/energy-star-canada/about/22177 (accessed Feb. 16, 2023).

29. “What is ENERGY STAR.” https://www.energystar.gov/about (accessed Feb. 16, 2023).

30. “ENERGY STAR CANADA 2016 Annual Report,” Natural Resources Canada’s Office of Energy Efficiency. [Online]. Available: https://natural-resources.canada.ca/sites/nrcan/files/oee/pdf/energystar/english/ENERGYSTAR2016AnnualReport-EN.pdf

31. N. R. Canada, “ENERGY STAR Canada Awards,” Nov. 28, 2013. https://natural-resources.canada.ca/energy-efficiency/energy-star-canada/about/energy-star-announcements/energy-star-canada-awards/13524 (accessed Feb. 16, 2023).

32. D. Sönmez and K. Dinçer, "A review of modern residential thermostats for home automation to provide energy efficiency," *2016 4th International Istanbul Smart Grid Congress and Fair (ICSG)*, Istanbul, Turkey, 2016, pp. 1-4, doi: 10.1109/SGCF.2016.7492430.

33. O. Ayan and B. Turkay, "Smart Thermostats for Home Automation Systems and Energy Savings from Smart Thermostats," *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, Istanbul, Turkey, 2018, pp. 1-6, doi: 10.1109/CEIT.2018.8751790.

34. J. Rodriguez and N. Fumo, "Zoned heating, ventilation, and air–conditioning residential systems: A systematic review," *Journal of Building Engineering*, vol. 43, p. 102925, Nov. 2021, 10.1016/j.jobe.2021.102925.

35. F. F. Hernández, J. M. P. Suárez, J. A. B. Cantalejo, and M. C. G. Muriano, "Impact of zoning heating and air conditioning control systems in users comfort and energy efficiency in residential buildings," *Energy Conversion and Management*, vol. 267, p. 115954, Sep. 2022. 10.1016/j.enconman.2022.115954.

36. E. Barrett and S. Linder, "Autonomous HVAC Control, A Reinforcement Learning Approach," in *Machine Learning and Knowledge Discovery in Databases*, Cham, 2015, pp. 3–19. doi: 10.1007/978-3-319-23461-8_1.

37. Environmental Assessment of Renewable Energy Conversion Technologies.

38. "Eco-indicator 99 Manual for Designers," Oct. 2000.

39 "Rechargeable AA Batteries | AAA Rechargeable Batteries | Duracell," *Duracell Batteries | AA, AAA, Rechargeable, Coin Button*. https://www.duracell.com/en-us/product/rechargeable-battery/ (accessed Apr. 10, 2023).

Appendices

Item Costs

Table 5. Cost Analysis for the Sensor Module.

Item	Cost	Quantity	Supplier
Arduino NANO RP2040 Connect	$47.10	1	https://www.mouser.ca/ProductDetail/782-ABX00053
Panasonic Wall Board Mount PIR Sensor	$15.87	1	https://www.mouser.ca/ProductDetail/667-EKMC1604112
Microchip Temperature Sensor	$0.88	1	https://www.mouser.ca/ProductDetail/579-MCP9700-E-TO
AA battery holder	$4.50	1	https://www.mouser.ca/ProductDetail/Eagle-Plastic-Devices/12BH350-GR?qs=rHWvavlphBeBgXzuRuUvnw%3D%3D
AA Battery	$0.54	4	https://www.mouser.ca/ProductDetail/Panasonic-Battery/LR6XWA-B2?qs=1Zl%2FqaC79saZu5UEOwg5aw%3D%3D
Simple plastic case	$15.74	1	Estimated value roughly based on cost for custom vent cover (multiply by 0.75).
Total	$86.25		
Total with tax	$97.46		

Table 6. Cost Analysis for the Control Module.

Item	Cost	Quantity	Supplier
Arduino NANO RP2040 Connect	$47.10	1	https://www.mouser.ca/ProductDetail/782-ABX00053
SG95 Servo	$5.52	1	https://www.mouser.ca/ProductDetail/DFRobot/SER0043?qs=5aG0NVq1C4zNL6r3%252B8mR2g%3D%3D
Logic Level Converter	$4.07	1	https://www.mouser.ca/ProductDetail/SparkFun/BOB-12009?qs=WyAARYrbSnb%252BGYLWggQnjQ%3D%3D
AA battery holder	$4.50	1	https://www.mouser.ca/ProductDetail/Eagle-Plastic-Devices/12BH350-GR?qs=rHWvavlphBeBgXzuRuUvnw%3D%3D
AA Battery	$0.54	4	https://www.mouser.ca/ProductDetail/Panasonic-Battery/LR6XWA-B2?qs=1Zl%2FqaC79saZu5UEOwg5aw%3D%3D
Custom Vent Cover	$20.98	1	Estimated based on cost for normal register multiplied by 2. https://www.canadiantire.ca/en/pdp/imperial-louvered-plastic-floor-register-adjustable-dishwasher-safe-white-4-x-10-in-0643041p.html?loc=plp
Total	$84.33		
Total with tax	$95.29		

Pseudocode

```
void MotionISR()
{
  //Disable interrupts

  //Set roomOccupied to 1

  //Store lastTimeMotionDetected to current board time

  //Enable interrupts
}

void PeriodicISR()
{
  //Disable interrupts

  //Set periodEnded to true

  //Enable interrupts
}

void setup()
{
  //Connect the Arduino board to WIFI

  //Setup digital pin 10 as an input

  //Setup analog pin 0 as an input

  //Connect digital pin 10 to the MotionISR so it triggers on the rising edge

  //Setup the periodic interrupt so it triggers the PeriodicISR every 20 seconds
}

void loop()
{
  /*
  If (periodEnded)
  {
    Set periodEnded to false

    If (current time - lastTimeMotionDetected > 20 seconds)
    {
      Set roomOccupied to 0
    }

    Set currentTemperature to current values from the temperature sensor

    If connected to wifi
    {
      Combined roomOccupied and currentTemperature values into a json compliant string

      Send the data to the website using an HTTP POST request.
    }
  }
  */
}
```

Figure 22. Figure 23. The complete pseudocode for the Sensor Module.

```
void PeriodicISR()
{
  //Disable interrupts

  //Set periodEnded to true

  //Enable interrupts
}

void setup()
{
  //Connect the Arduino board to WIFI

  //Setup digital pin 10 as an output

  //Setup the periodic interrupt so it triggers the PeriodicISR every 20 seconds
}

void loop()
{
  /*
  If (periodEnded)
  {
    Set periodEnded to false

    If connected to wifi
    {
      Get the current temperature error from the website using an HTTP GET request.

      Calculate the new temperature error sum

      Calculate the PID output using the PID equation

      Calculate the angle error using the PID output

      Calculate the servo angle by adding the angle error to the current angle

      If angle > 80
        Set angle to 80

      If angle < 0
        Set angle to 0

      Apply servo angle to the servo

      Set the previous temperature error to the current error
    }
  }
  */
}
```

Figure 24. Figure 25. The complete pseudocode for the Control Module.