CtrlFlow - Distributed IoT Thermo System

Yoseph Hassan, Joseph Mansy, Muhamad Elassar, Vinh Tran

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This paper is representative of all of the methods and techniques used in developing and designing a small scale representation of an Internet of Things smart thermo system. This will delve into not only the hardware associated with the design but also the software component as well as the testing and administrative procedures put in place throughout the duration of the project. This project allows for users of the system to control different temperatures in different rooms based on their individual needs. This is performed via a mobile application component which allows for remote control of the specific temperatures in each room. These rooms are equipped with specially made vents connected to microcontrollers which affect the opening and closing of these vents. These temperatures cannot be controlled concurrently but rather are staggered after completing the request of temperature change of one room and moving on to the other. Temperatures can be mostly maintained by closing vents and preserving the current temperature.**

I. Introduction

In Florida and other regions where temperatures can significantly affect comfort greatly, modern HVAC systems are simply not providing an adequate solution. These systems typically cool on full blast until the desired temperature is reached and then turn off. This process obviously has its downfalls as it is quite primitive in its approach to heating and cooling. Individuals often require different temperatures to meet their comfort needs and while these systems help mitigate the issue by allowing for some version of temperature control they are far from a comprehensive solution which allows for users to have more personalized management of their respective rooms. Our product was conceived to further lessen the issues mentioned earlier and also provide a bit more convenience to actually controlling the temperature. This is accomplished via connecting several microcontrollers to a main hub then connecting the auxiliary microcontrollers to specially made vents which open or close depending on the current temperature of the room and the target temp to be reached by the heating/cooling apparatus. All of the hardware in the system is expected to communicate wirelessly via LoRa. Apart from the boards communicating between themselves wirelessly we also allowed for wireless communication between the boards and the mobile device controlling the temperatures of the rooms. This decision was based on improved convenience as it would be inconvenient for users to have to access an immobile device to alter temperature.

The wireless nature of the project is to promote easier setup for the user and prevent unnecessary wire clutter throughout the home. This falls under further convenience to create an easy to use environment for inexperienced users.

The expectation of this project is to provide a small scale representation of a home to provide a proof of concept of what can be relatively easily accomplished in a full scale implementation of the design. Given the maximum range of the LoRa standard the design would not require much modification in order to be applied in a full scale setting as in an actual modern home the distances between the boards would be much more significant than in the model.

II. System Components

A. Internet of Things (IoT)

The Internet of Things, or IoT, describes the network of billions of physical devices that are connected to the internet, all collecting and sharing data. These devices are known as “things” and are embedded with sensors, software, and other technologies. They also are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. Thanks to extremely cheap computer chips and the ubiquity of wireless networks, it’s possible to turn anything, from something as small as a pen to something as big as a cruise ship, into a part of the IoT.

Fig. 0. Internet of things diagram representation.

The smart devices that are intermittently connected through IoT share the data that they collected to an IoT gateway or other edge device where the data is then sent to the cloud to be analyzed and managed. In our case, we only require one smart device to be connected through IoT, the central hub thermostat. However, in some cases these devices communicate and share information with other devices that are related and perform actions based on the data they receive from one another. Being within an IoT network allows the devices to do the heavy lifting without any human action required. However, people can interact with the devices. For this project, the user will set up the central hub thermostat and control the temperatures in each room using it.

The simplest way to dive into the IoT network and get devices set up to communicate with each other is to use an IoT platform. An IoT platform gives developers a head start in incorporating their smart devices into an IoT system by providing built-in tools, features, and capabilities to make the life cycle of IoT a lot easier and less costly for the consumer. There are many big tech companies today that have developed their own IoT platforms for developers to delve into, with lots of provided documentation and big community support. The IoT platform used for this project is Arduino IoT Cloud.

B. Microcontroller

One of the most crucial components in our design is the microcontroller we chose for our circuit boards as these were what would be used throughout all of the boards throughout the distributed system. These microcontrollers are in charge of processing all of the logic we implemented into our project therefore making them an integral part of the overall design and implementation.

The microcontroller we decided to move forward with was the ATSAMD21G18A-AUT. Our reasoning behind using this chip is that the SAMD21 chips inherit most of the benefits of the aforementioned ATMega4809 while being faster and more power efficient. More importantly this MCU has many more configurations than really any of the other MCUs we had encountered when researching. There are several configurations that offer more or less pins and different program memory capacities. This is useful as we can use the same development environment and choose the MCU that best fits our needs for each separate device and optimize cost.

C. Thermoelectric Modules

Another integral part of the design is the thermoelectric module or peltier cooler. This component is the main device being used to emulate the heating and cooling ability of a modern HVAC system. This device is used in conjunction with a heatsink and fan to provide the heating or cooling required per each room in the model.

This component will be the driving force behind heating and cooling the system. Another reasoning we decided to use this device is due to the ease at which it can change modes from heating to cooling. By simply switching the polarity either side of the device can become hotter or cooler while running a current through the device.

These devices use a phenomenon known as the Peltier effect. The Peltier effect is used to create a heat flux at the link of two separate materials. These devices are then able to transfer heat from one side to the other via the expenditure of electrical energy. It is for this reason that this device will be ideal for the application associated with our design in which it is modeled after the heating and cooling in a home environment.

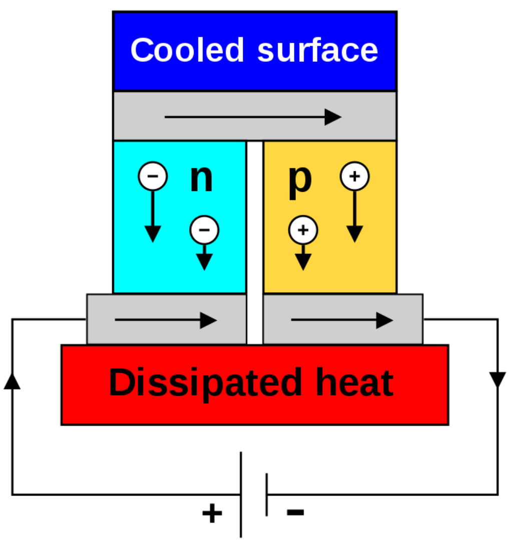


Fig. 1. Visual diagram representation of the peltier module basic design.

This Peltier effect is what occurs when an electrical current flows through a thermocouple circuit. This results in the heat being released on one side of the junction and being absorbed at the other end of the junction. The Peltier effect was named after French physicist, Jean Charles Athanase Peltier, who made its discovery in 1834.

The Equation associated with Peltier heat is shown below:

$$Q = (\Pi_A - \Pi_B) I \quad (1)$$

Such that *A* and *B* are conductors, *A* and *B* are the Peltier coefficients of their respective conductors, and *I* is the electrical current from conductors *A* to *B*. These Peltier coefficients are measurements in terms of the amount of heat which is carried per unit charge.

D. Dynamic Vent mechanism

These vents were specially made in order to accomplish the design we had conceived. This is another major component integral to the overall design of our project as it is the main factor used in controlling heating and cooling apart from the peltier cooler. This device is what controlled the insulation of heat in the system. When the heating or cooling mode is altered this device will be directly affected as if heating is activated the vent will open.

If cooling is activated the vent mechanism will cause the vent to open. Conversely if heating is activated the vent will also open in order to heat the current room. When maintaining the room temperature which has met the target the vent will then close. When the current temperature falls above or below the target the vent will once again open.

The vents being used in the design were custom 3D printed vents which utilize a valve mechanism connected servo in order to control the opening and closing. More specifically, the opening and closing mechanism is modeled after a ball located within a box connected to the end of the tubes carrying the hot or cool air.

Servo motors are simple to operate actuators. These motors have 3 wires, power, control signal and ground. Servos are most commonly powered by 4.8-5.5V through its power wire. Meanwhile the signal wire is used to receive control signals from MCU, which will make the servo turn its axis and hold at the requested angle. The position of a servo is determined by square waves with different duty cycles. Most servos share the same control signal of 50Hz squarewave, where 5% duty cycle for default position, 10% duty cycle for max position (which is 180 degree relative to the default position). These wave signal can be created with ease by micro controllers.

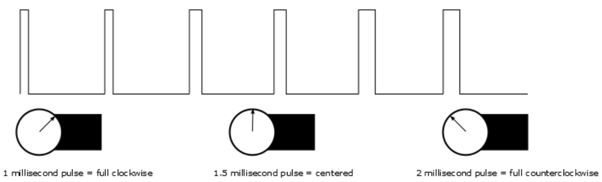


Fig. 2. Servo rotations associated with specific pulse widths ranging from 1ms to 2ms.

Servo motor can differentiate the signals because it has a built-in microcontroller. The microcontroller performs 2 tasks, reading the current axis position and reading the requested angle. If there's a difference between the current axis angle and the requested angle, the microcontroller will spin a motor in the required direction until the two angles match. Since the microcontroller constantly reads the input signal (every 20 ms), it'll react immediately if there's any subtle changes, making the servo extremely reactive and easy to use.

Due to the flexibility of 3D printing a moving part is able to be printed in place with minimal assembly. As such with careful design and appropriate tolerances we are able to print a sufficiently air tight ball valve.

E. Linear Voltage Regulator

This voltage regulator is being used in conjunction with the main hub as it is needed to convert 5V to 3.3V in order to be compatible with the hub's printed circuit board design.

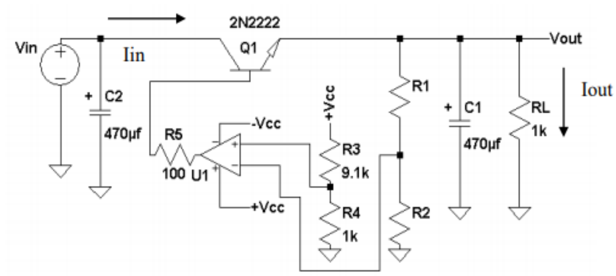


Fig. 3. Schematic representation of linear voltage regulator being used with the hub.

Linear voltage regulators utilized an operational amplifier with a NPN transistor. In a nutshell, the operational amplifier will control the NPN transistor to keep the output voltage at a desired voltage level. The desired voltage level can be programmed with R1 and R2: Vout = Vref*((R1+R2)/R2); in the following figure, Vref is Vcc*0.099. The output voltage will always be below the input voltage. Additionally the desired output voltage is only achieved when the input is greater than the desired voltage, this difference is known as dropout voltage. There are regulators that have low dropout voltage, also known as LDO.

F. Switching Voltage Regulator

In addition to the main hub requiring a voltage regulator to convert to 3.3V, the auxiliary boards also require some version of voltage conversion. In the case of these boards battery power is being utilized and this voltage can drop below 3.3V or increase to a higher value. For this, the best option was a switching voltage regulator.

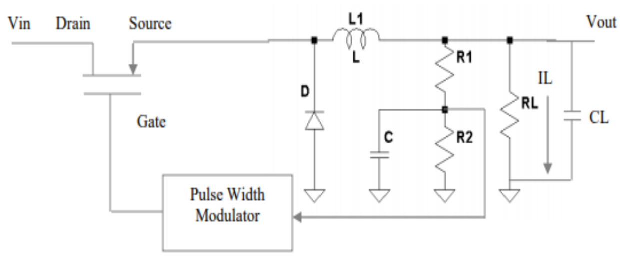


Fig. 4. Schematic representation of switching voltage regulator.

Switching voltage regulators are much more complex as more logic is required to control a gate (which is usually a MOSFET). Switching voltage regulators are usually available as highly integrated ICs. The circuit works by switching the MOSFET in a pre-programmed frequency, effectively charging and discharging the inductor. The desired voltage is programmed by a voltage divider, which will affect the charging period of the inductor. In practical applications, potentiometers are used, as resistors have tolerance and can't be accurate.

III. Network Topology and Standards

A. Star topology

Star topology is one of the methods to establish connections between devices within an Iot network. This wireless topology will connect every sensor node on an IoT network to a central hub, known as the gateway. The gateway can then connect the information to a cloud service, usually by wifi.

The connection between the gateway and sensor nodes are wireless point to point connection. This gives a security advantage since each connection between a sensor node and gateway is already established. Additionally, since each device has its own connection path to the gateway, troubleshooting the network is straightforward. If any device encountered a problem, from low battery, connection issue to being attacked by hackers, the gateway can detect and send reports to the user.

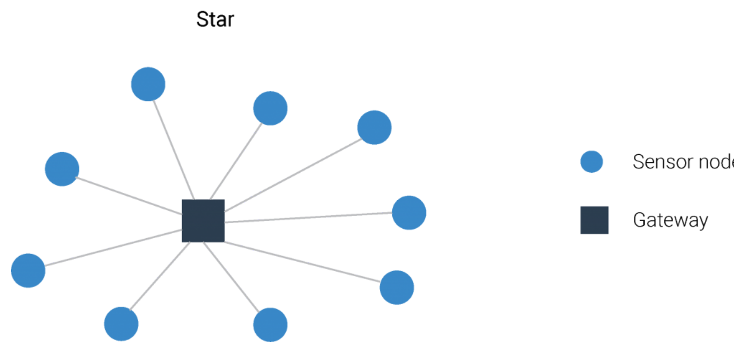


Fig. 5. Diagram representing star topology with sensor nodes.

This, however, also makes the gateway the most vulnerable link of the IoT network. If errors occur on the gateway and crash it, the whole network stops. This means valuable information will not be able to be recorded, connection between end devices will be stopped completely. In more serious cases where IoT is responsible for security, such as protecting a house, the user's privacy will be at risk.

B. Mesh topology

Mesh topology is an approach that overcomes the limitations of star topology. In mesh topology, each device can relay each other's signal. A gateway device is in the mesh network to connect the entire IoT to the cloud service, so not all devices are required to perform this task.

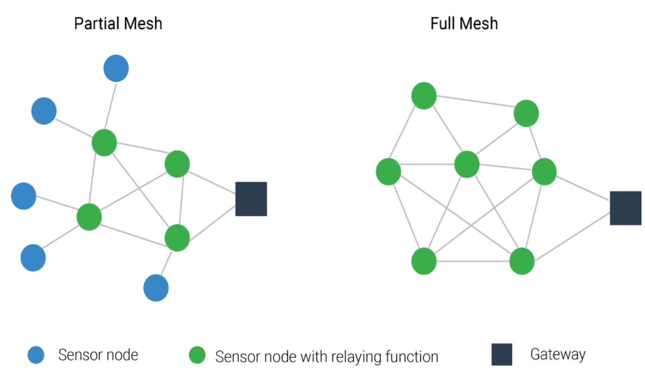


Fig. 6. Diagram representing mesh topology with sensor nodes.

Implementing a mesh topology can be very challenging, however. Firstly, since each device will make connections with more devices, there must be more security measures to make sure the network stays secured.

Secondly, the network might require a lot of relay nodes to cover the entire application area, so implementation can be expensive. Finally, mesh topology requires each device to handle data autonomously, so the code can be very complex.

C. LoRa

LoRa was the chosen communication standard for our devices. LoRa, which stands for long range, is a low power wireless communication protocol developed by Semtech. LoRa operates at sub-gigahertz frequencies of 433 MHz to 923 MHz depending on the location or continent it is being operated in. LoRa is able to achieve a data transfer rate up to 27 kbps, depending on the spreading factor of the signal.

LoRa operates by chirp spread spectrum, which uses a chirp pulse, wide in bandwidth and increases linearly in frequency, to encode the signal. The resulting signal power is spread over a wide spectrum. Unlike other spread spectrum modulations like direct-sequence spread spectrum (DSSS) or frequency-hopping spread spectrum (FHSS), chirp spread spectrum doesn't require a pseudo random noise to encode and decode the signal.

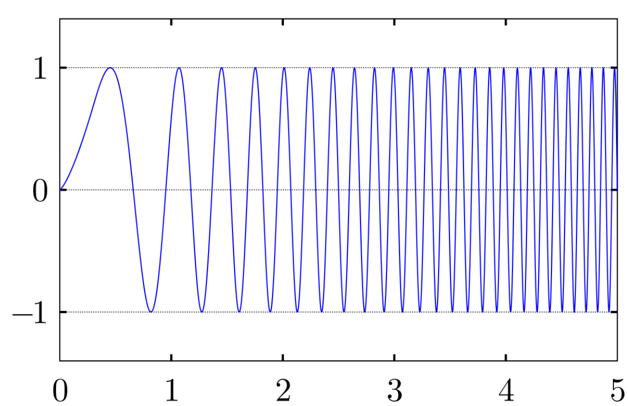


Fig. 7. Diagram of a chirp pulse, a sinusoidal wave that increases in frequency linearly over time.

Before the output signal is modulated, the chirp pulse is repeated indefinitely. LoRa will then modulate each segment of the pulses to represent the original data. Each symbol of the data is represented in a certain pattern of frequency. This pattern is what carries the information.

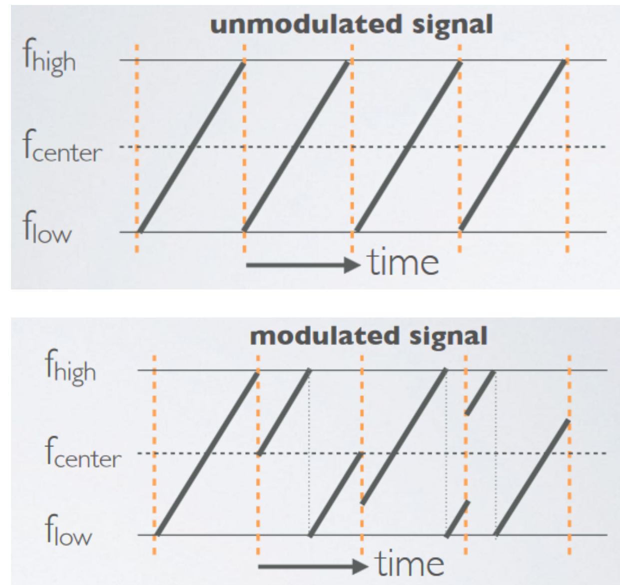


Fig. 8. Output signal's frequency over time, before modulation and after modulation.

As seen in figure 7, there are 4 distinct patterns after the signal is modulated. The receiver just needs to catch how the pattern changes in order to decode a symbol of the signal. Since the pattern is quite distinct, the receiver doesn't require a highly accurate nor synchronized clock to capture the packet. The slope in which the frequency increases linearly can be adjusted with spread factor. The higher the spread factor, the longer the chirp pulse and linear slope, meaning slower the data transfer rate.

These unique chirps patterns eliminate common errors caused by the Doppler effect, where a signal receives higher frequencies than it should, even at low frequencies thanks to its linearity. Additionally, chirp spectrum spreading has high noise immunity and multi-path fading, similar to other spread spectrum modulations.

Due to the modulation allowing high signal to noise ratio, LoRa doesn't require much power to send signals. Low power consumption of LoRa makes it possible for battery operated devices to last for 10 years. Chirp spectrum spreading allows LoRa to transmit at distances up to 5 km to 15 km, depending on the environment.

All of these characteristics make LoRa a suitable communication standard for our devices. As for the method of setting up the connection role of each IoT device, we decided that star topology would be the most suitable thanks to its simplicity. The thermostat will act as a hub, receiving and sending data to the cloud. The thermostat will also read the temperature sensor and control the vent, following the star topology.

IV. Cloud Services & Mobile Application

In order to send and receive data between the user and the central thermostat hub, an IoT cloud service was required to act as a connector to handle requests or commands. In addition, the cloud service also acts as storage for the user's data, so that previous commands and temperatures are saved in the application. There are many notable IoT platforms that have a core solution with basic functionality and a set of additional modules you can add when necessary. After extensive research, we found that Arduino IoT Cloud suited our needs the best.

Although other platforms also have a core solution and would be sufficient for this project, Arduino IoT Cloud provided us with a lot of built-in functionality due to our utilization of an Arduino chip. Using Arduino IoT Cloud with our Nano 33 IoT chip allows us to run software, read sensors, control actuators, and communicate over the cloud via WiFi. Once this communication is configured, a user interface is required so that data can be viewed and sent. The most convenient and user-friendly way to accomplish this was a mobile application.

With a mobile application, the user will be able to easily view and change the temperature in each room that contains a smart vent with a click of a button on their mobile device. Since requests will constantly be sent between the mobile app and the IoT service, a mobile application platform that is compatible with Arduino IoT Cloud was required. This led us to take advantage of the built-in mobile application capabilities Arduino IoT Cloud provided. This includes a customizable user interface to display each room's temperature and a stepper to change the temperature. The mobile application can be installed for both Android and iOS.

V. System Concept

At a high level the flow of control follows as such. The user is able to utilize the mobile application which will communicate with the IoT cloud services. The central hub will be in charge of interconnecting between the auxiliary boards which include temperature sensors and also control the servos for opening and closing valves.

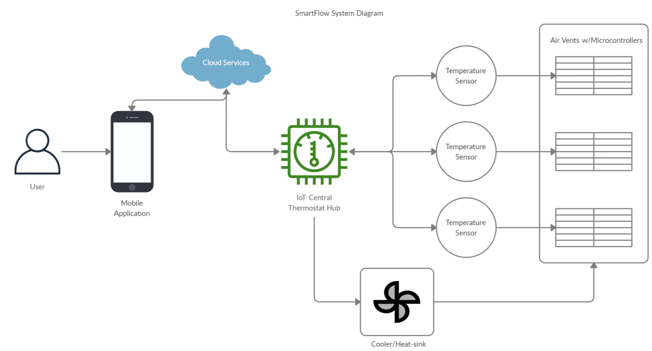


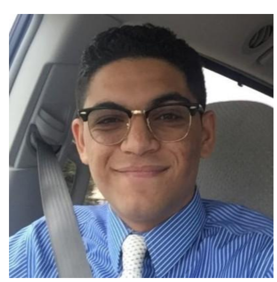
Fig. 9. Diagram representing mesh topology with sensor nodes.

In our system there are three types of devices - the central hub and the two types of peripheral devices. Given the temperature targets received from the web server over WiFi the central hub will keep track of temperature recordings from each room and send control signals to the AC system and the Vents so that the air can be directed towards where it's needed most.

We wanted to design our application with functionality in mind first. The primary purpose of the application is to control the AC system in a home via the user's smartphone. As such the simplest design possible for our application is best in order to provide the best user experience.

VI. Team Members

Our team is made up of a group of electrical and computer engineers with various skill sets that each offer their own expertise on certain aspects of the product's design.



Yoseph Hassan is currently a senior at the University of Central Florida and will receive his Bachelor's of Science in Computer Engineering in May of 2021. He has attended the University of Central Florida for a few years now and plans to continue his Masters and possibly his PhD studies. He is currently working as a research and development intern at Leidos. His primary interests lie in power artificial intelligence, machine learning, data science and mathematics.

Joseph Mansy is currently a senior at the University of Central Florida and will receive his Bachelor's of Science in Computer Engineering in May of 2021. He joined UCF in the fall 2018 semester. He has worked as an intern at Lockheed Martin through the UCF CWEP program. After graduating he plans on working in the software industry before returning to complete his masters.

Muhamad Elassar is currently a senior at the University of Central Florida and will receive his Bachelor's of Science in Computer Engineering in May of 2021. He is currently a Software Engineer Intern at Leidos and plans to continue working in the tech industry as a software engineer. He also plans to earn a master's degree in Computer Science in the near future.

Vinh Tran is currently a senior at the University of Central Florida and will receive his Bachelor's of Science in Electrical Engineering in May of 2021. He has attended the University of Central Florida since 2016. His interest has been programming low power embedded systems and microprocessors. He has no interest in continuing with his master degree and will begin his work after getting his bachelor.

References

[1] "Internet of Things (IOT)." SAS: Analytics Software & Solutions. Retrieved December 2, 2020, from https://www.sas.com/en_us/insights/big-data/internet-of-things.html

[2] File:Thermoelectric Cooler Diagram.svg. Wikimedia Commons. Retrieved 21, 2020. from https://commons.wikimedia.org/wiki/File:Thermoelectric_Cooler_Diagram.svg

[3] "Hobby Servo Tutorial" Sparkfun. Retrieved December 6, 2020, from https://learn.sparkfun.com/tutorials/hobby-servo-tutorial/all

[4] "Internet of Things: What It Is, How It Works, Examples and More" Just Creative. Retrieved December 2, 2020 from https://justcreative.com/internet-of-things-explained/

[5] "Mesh vs. Star Topology – How to Find the Right Architecture for Your IoT Networks". BehrTech. Retrieved December 2, 2020 from https://behrtech.com/blog/mesh-vs-star-topology/

[6] "File:Linear-chirp.svg" Wikimedia Commons. Retrieved November 21, 2020 from https://commons.wikimedia.org/wiki/File:Linear-chirp.svg