2022

Unit 19: Internet of Things

ASSIGNMENT TWO

CONNOR THOMPSON

Contents

Gantt chart – start of assignment

https://docs.google.com/spreadsheets/d/1XcXVrVD389lQYUD2eKKH6MMhmQNzpsXZlwSnSC1tpdE/edit?usp=sharing

Design Stage

Diagram of the design

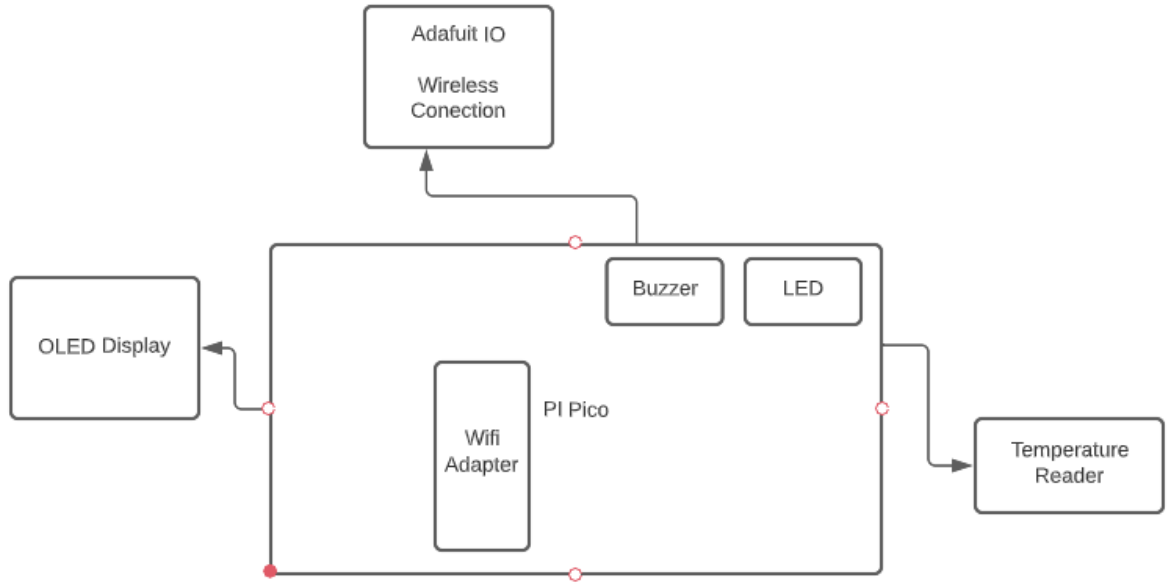


Figure1: Design of the prototype

This is my design for my prototype of the system. In the design the components involved include a temperature reader, WIFI adapter, buzzer, OLED display and an LED that can be found on the board. The components used are from the equipment we have been provided for this assignment due to being in an educational situation. In the design, I also show the use of Adafruit.IO for sending and receiving the data through a wireless connection.

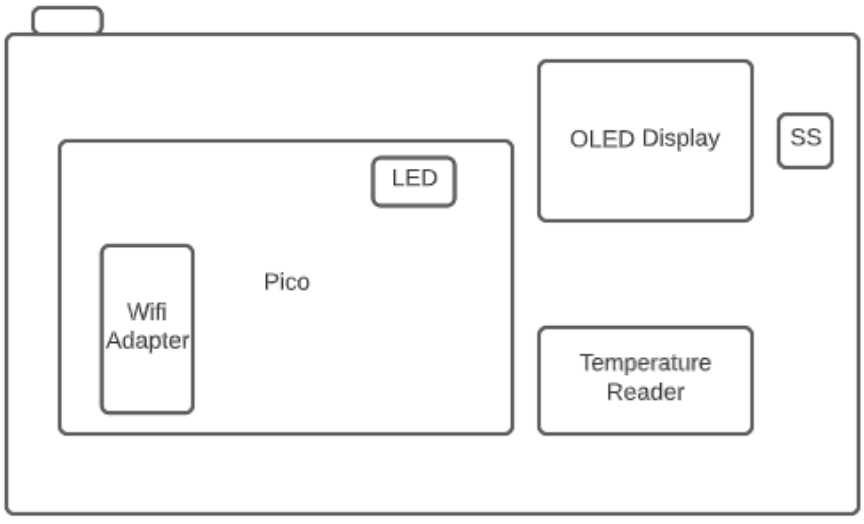


Figure2: Case holding prototype design

This diagram is an idea of a holding case if the prototype was to be made in production. The case would be a box with curved edges and would have transparent acrylic to show certain components and hardware such as the OLED and the Pico to ensure that the correct lights are lit up on the board to indicate the connection to the components and also to show the LED on the board. There would also be some small holes near the buzzer on the Pico to allow the sound to exit the case and get heard by the end-users.

Architecture

M2M system

Machine-to-Machine is the process of two devices communicating to each other using any type of communication wired or wireless that doesn't require any human interaction to start or finish the communication between the two devices. I have multiple options of communication that I can use with the prototype which have been assigned to design.

Smart Services

Sensor Networking

A sensor network is a network that uses sensors to monitor and track the physical condition of an environment and forward the gathered data to one location of which it can be monitored and used as needed by the person that is collecting the data. This would be really good for the project because we have to collect data of an environmental area of which we will be using temperature sensors to gather the data and send that data over a wireless connection to a feed using Adafruit to collect and organise the data into different sets according to the data being collected and then do as what is required with the following data. In this case, if the temperature hits a specific number, then an alarm goes off informing the company of a drastic or big change within the environment that they work in and that it requires attention. The company could also then use Adafruit and the data sets to view and monitor the temperature themselves at any time using a data feed of which all the data would be organised or put into a table to monitor the changes over the give time periods.

Sensors (Input)

DHT11

One of the sensors that would be used is the DHT11 temperature and humidity sensor. This sensor can measure and monitor the temperature of a room or environmental area within a building or outside of the building. The DHT11 is a good choice of sensor in the fact that it is being used within a building, meaning that it will be monitoring and gathering the temperature of a room most likely. The DHT11 is able to monitor temperatures ranging from 0-40 degrees with a +- of 2 degrees. This would be a good option for the company because there is a low chance that they would need to monitor a room that is over 40 degrees unless it was to be some sort of server room. However, the company could use a DHT22, not so much for the bigger temperature reading scale but the accuracy and durability of the component. The DHT22 ranges from -40 to +125 degrees with a +- of 0.5 degree making the reading more accurate than the DHT11 because it has a +- of 2 degrees. This would help the company monitor the environment better and get more accurate reading sent into the data sets which are viewed via Adafruit. With its durability to withstand more crucial temperatures, if

something was to happen to the building which would inflict those kinds of temperatures. The DHT22 would stay functional and warn the company of the temperature within the environment that they have its setup in. However, given the fact that I am in an educational situation, I am going to have to use the given hardware which would be the DHT11 but in a real-world scenario, the DHT22 could be more beneficial and sustainable for the company and the project prototype.

Actuators (Output)

OLED

One of my outputs that may be used in my prototype is an OLED display. It would be helpful to have an OLED display because it can equally show the temperature of the environment and any of the company staff could then look at the system in the room to see the temperature as well, instead of always having the use the data feed online through Adafruit. It would be up to the company, whether or not they are after a decently sized OLED display or if they would be okay with a small one as long as they can see the temperature of the room/environment. This would be beneficial as a warning display because it could tell anyone in the room if the environment is too hot and requires action to control and stabilise the area.

Buzzer

The buzzer would be used in the design and prototype of the environment alarm system because it would give off a sound of which the company staff would be able to here. The buzzer could give off a constant tone or a patterned tone to get the companies attention and would be activated if the area that the system is covering and monitoring hits a certain temperature of which the company sets and if that temperature is hit, then the buzzer goes off giving off an alarm and informs the company to take whatever action is need to control or maintain the temperature within that area or environment. Due to being in an educational matter, I will have to use the buzzer on the raspberry Pi Pico; However, a speaker kit for raspberry pi's or a Bluetooth speaker could be used to provide the sound of the alarm.

Dashboard Outline

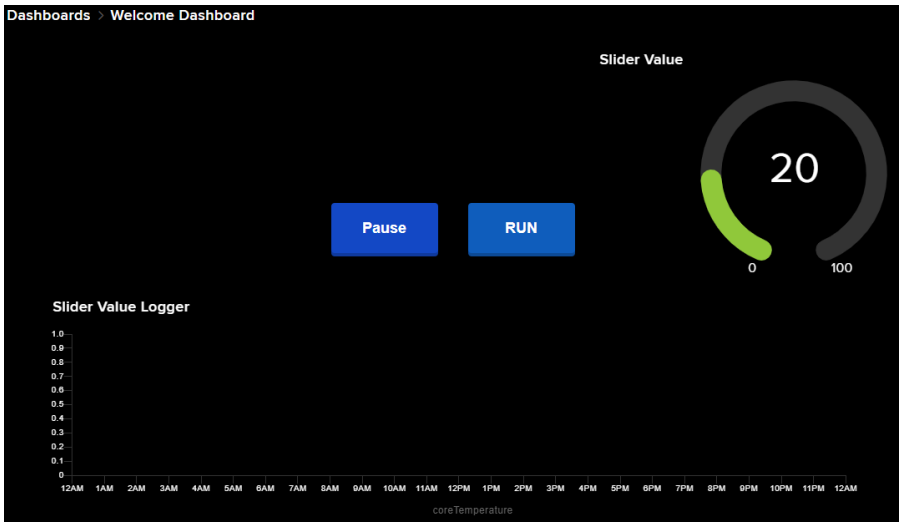


Figure 3: Dashboard layout for Adafruit

What Feeds am I going to use?

I am going to create and use a data feed that has features such as a pause and run feature or a timer feature of which they can control it and tell it not to operate during closed office times. This could be vital in sustaining the system or the possible extra cost of running it during closed times at the building.

What triggers will be set on Adafruit?

The trigger I will be using, is a temperature trigger. This trigger would work in conjunction with the buzzer because if the temperature set in the trigger is met and activated. Then the raspberry pi Pico would be told to turn on the buzzer to notify the company staff that the certain temperature has been hit and that the company need to take action to control and stabilise the environment they are monitoring.

Three temperature triggers would have to be set so that the feeds are different and aren't activating two functions at the time. Therefore, there would be a trigger for (below set temperature), (equal to set temperature) and (over set temperature). With each trigger having a different temperature differential target. Depending on which trigger is activated, depends on what action shall be taken; Will it activate the buzzer or just keep sending temperature data to the feed.

What buttons on Adafruit will be used?

I am planning on using two buttons on Adafruit to help the functioning and monitoring of the alarm system. The first button would be a run button if the system was to for whatever reason stop and would allow the company to get the system running again. The second button would be a pause button, this button would be used to put the system to a halt if the system needed some maintenance or if there were any hardware issues. It would be important to have a button the stops the alarm system and turns off the components because it could be a big risk and not very safe if the components were to remain on during the maintenance.

Quality of Service

With QoS, I have got three options of QoS 0, QoS 1 and QoS 2. QoS 0 focuses on delivering the best effort basis but without providing any form of confirmation of receipt. This means that there would be no confirmation of whether or not the message did arrive to the desired system. It also uses telemetry to gather information in the idea of that a message isn't received, a value can be still sent based off of the analysis of the other information that was collected. QoS 1 ensures that there is a receipt of confirmation because it makes sure that the message arrives to the system, but this can lead to a possibility of duplicated data because if the first message doesn't arrive in the given time, then it will send another and therefore can create duplicated data. Whereas with QoS 2, it guarantees the delivery of the message to the system in exactly one attempt without sending it again. This means that there wouldn't be any duplicated data that could affect the processing and actions of the system because it sends the message and delivers it in one shot to the system.

Standards

Frameworks

The framework that will be used in the prototype is the IEEE, also known as the P2413. This framework is a type of method for sharing data, consistency, and security of messages over a

network of which components such as sensors and actuators and any other forms of devices can work together, regardless of the technology given for communication.

Communication Needed

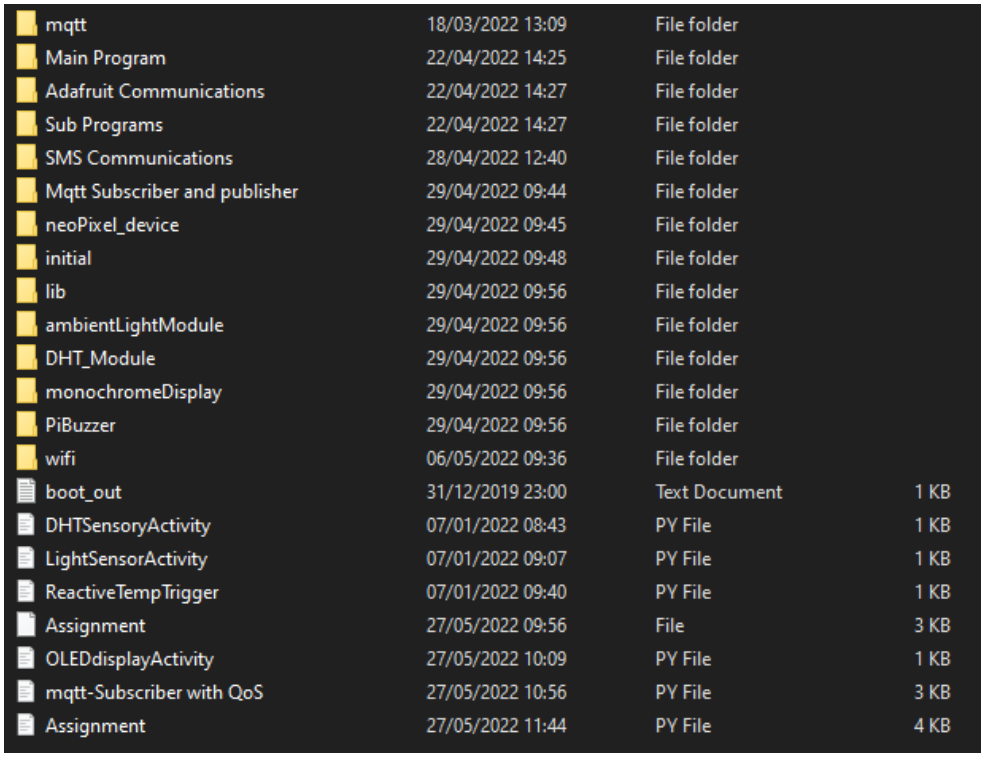
System inter-device types

I2C – Inter-Integrated Circuit Communication

I2C is the protocol that relies on sending data in a serial format using only data wires SDA and SCL. The data line cannot change when the clock line is high, but it can only change when the clock line is low, according to I2C protocols. Because the two lines are open drain, a pull-up resistor is necessary to make the lines high, as the I2C bus devices are active low. The information is sent in packets of 9 bits each. The two lines are used to send and receive data; a serial clock pin (SCL) that the Arduino Master board pulses at a regular interval, and a serial data pin (SDA) over which data is sent between the two devices.

File and folder layout

The file and folder layout are pretty important in the sense that it needs to be easy to follow so that the company can update any files that receive new versions and that they can track and replace the files with ease. The folder and file layout I am proceeding with can be found below:



mqtt	18/03/2022 13:09	File folder	
Main Program	22/04/2022 14:25	File folder	
Adafruit Communications	22/04/2022 14:27	File folder	
Sub Programs	22/04/2022 14:27	File folder	
SMS Communications	28/04/2022 12:40	File folder	
Mqtt Subscriber and publisher	29/04/2022 09:44	File folder	
neoPixel_device	29/04/2022 09:45	File folder	
initial	29/04/2022 09:48	File folder	
lib	29/04/2022 09:56	File folder	
ambientLightModule	29/04/2022 09:56	File folder	
DHT_Module	29/04/2022 09:56	File folder	
monochromeDisplay	29/04/2022 09:56	File folder	
PiBuzzer	29/04/2022 09:56	File folder	
wifi	06/05/2022 09:36	File folder	
boot_out	31/12/2019 23:00	Text Document	1 KB
DHTSensoryActivity	07/01/2022 08:43	PY File	1 KB
LightSensorActivity	07/01/2022 09:07	PY File	1 KB
ReactiveTempTrigger	07/01/2022 09:40	PY File	1 KB
Assignment	27/05/2022 09:56	File	3 KB
OLEDdisplayActivity	27/05/2022 10:09	PY File	1 KB
mqtt-Subscriber with QoS	27/05/2022 10:56	PY File	3 KB
Assignment	27/05/2022 11:44	PY File	4 KB

Figure4: File and folder Layout

Security

Potential Threats

Being a system with wireless connections to networks and having hardware that can connect to other services. There are a lot of potential threats that could affect the system, some of these include:

- **Hackers:** Hackers would be able to hack into the device and steal or encrypt all the files on the raspberry pi and that would then break the system and it wouldn't be able to fulfil its purpose that it was designed and made for. The hackers could also potentially get into Adafruit and mess with the remote-control features and create false data and cause problems for the company when they are analysing it and could possibly lock out the company and change the passwords, so that the company can't get in and collect the data readings from the system.
- **Viruses:** Viruses are another problem; these could delete and corrupt files and also steal any information that they find. The only way a virus could get onto the system is via manual installation or through a file they download and put onto the pi if the company is installing a system update and get a fake file without knowing and give the virus access to the system and all its contents.
- **Natural disasters:** A natural disaster like flooding or an earthquake could also be a problem for the system and corrupting the information on it unless they have a backup of the systems contents or have access to Adafruit to obtain the data that the system had been recording and collecting.

Physical Security

Physical security is a good thing to have when it comes to securing systems or important equipment. Some physical security that can be used to protect this system include:

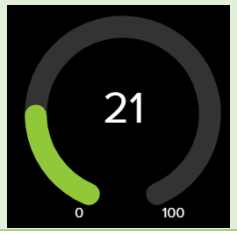
- **CCTV:** CCTV would be a good option for security because if the system is in a hallway and there is CCTV, if the system got exposed to some deadly software through manual installation. Then the CCTV camera would be able to see who the culprit was and can then take further actions.
- **Locks:** If the system was to be in a specific room, having locks on the doors that gain access to that room would be another really good idea because it further prevents people from having 100% access to the system and doing something to it.
- **Access Control:** Access controls such as regularly changing the password to the Adafruit dashboard every week and making sure that only the correct personnel have access to the remote access option and also provides further protection to the data and information been collected and generated by the system and stored on Adafruit.

Remote into the system

When it comes to remote access, the company will be able to access the system and control it remotely through Adafruit. Features that Adafruit provide in supporting remote access include:

- Dashboards to create buttons for the raspberry pi of which they can code and use to remotely carry out functions of the raspberry pi Pico.

Test Plan

Test No.	Testing	Expected Outcome	Actual Outcome	Works – Y/N	Solution if not working
1	That the pi connects to the internet	That it returns my request	It did as expected	Y	N/A
2	That the pi can connect to adafruit	That it displays a connection message	It displayed a message stating that it has connected to adafruit	Y	N/A
3	That the pi can grab and send the temperature to adafruit	That it prints "sending the temperature: temp" and shows a sent message	It did as expected and found the reading on adafruit	Y - 	N/A
4	That the pi activates the LED and buzzer if the temperature hits a certain mark	That the LED switches between blue and red and the buzzer gives off a simple sound to act as the alarm	It did as expected displaying red and blue lights and provides a sound as the alarm	Y	N/A
5	That the pi displays the temperature on the OLED	That it displays and updates the temperature on the OLED display	It did as expected	Y	N/A
6	Pause button on adafruit	The button when clicked should display a message on the terminal saying it is paused and should stop gathering temperature and close the display	It did as fully expected	Y	N/A
7	Run button on adafruit	Should go back to reading and sending the	It did as expected	Y	N/A

		temperature and turn the OLED back on to display the temperatures			
8	Relay button on adafruit	The switch should activate a red LED on the board and turn on the relay	It didn't turn on the really but could operate the LED just fine, I am not too sure what the problem is	N	A possible solution would be to make it manual but wouldn't fully meet the requirement of it being remotely controlled on adafruit.

Reviews

Reviewer	Feedback Given
Reviewer 1	The design is simple and easy, and has nearly everything as required for the company, but I would add the relay to act like controlling other systems
Reviewer 2	The design is good and shows where everything is and what is being used and also shows that the pi will connect to adafruit wirelessly

How does it meet the client requirements?

The requirements are as followed:

- Detect temperature changes: MET
- Provide them to the user in real-time and in data sets: MET
- Offer remote monitoring and management: MET
- Control environment control systems to ensure the environment is stable: Partially met

The fourth client requirement isn't fully met because it will require manual activation due to a problem in the program that won't allow it to be remotely controlled. Therefore, it would only meet half of that requirement.

Optimisation of the design and program based on the feedback

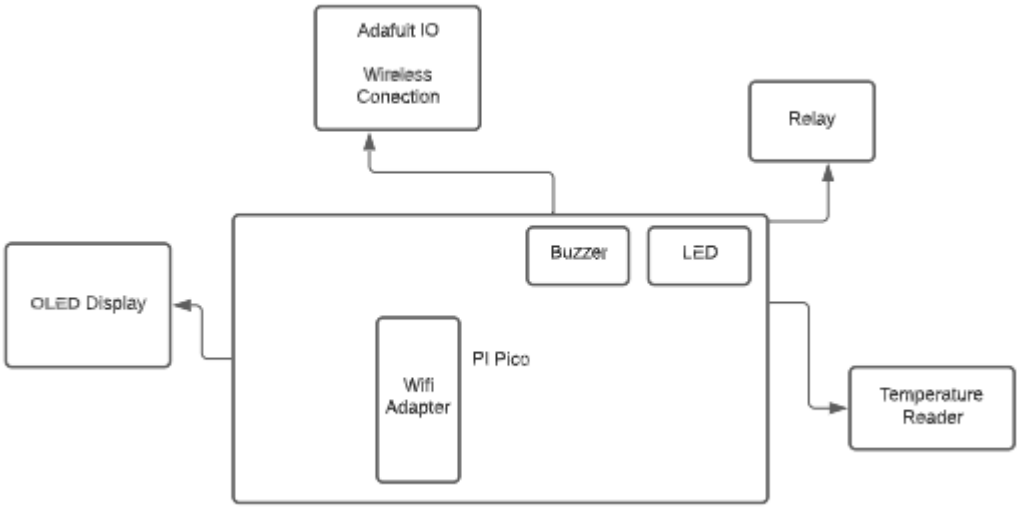
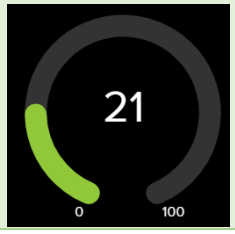


Figure4: Optimised design

Based off of the feedback given, I have added the relay to the design to show that it is part of the system and to control other devices that are bigger. The relay was added to the program and a switch to control it on adafruit but does not work and was not able to find out why or how to fix it.

Regression test plan

Test No.	Testing	Expected Outcome	Actual Outcome	Works – Y/N	Solution if not working
1	That the pi connects to the internet	That it returns my request	It did as expected	Y	N/A
2	That the pi can connect to adafruit	That it displays a connection message	It displayed a message stating that it has connected to adafruit	Y	N/A
3	That the pi can grab and send the temperature to adafruit	That it prints “sending the temperature: temp” and shows a sent message	It did as expected and found the reading on adafruit	Y - 	N/A
4	That the pi activates the LED and buzzer if the temperature hits a certain mark	That the LED switches between blue and red and the buzzer gives off a simple sound	It did as expected displaying red and blue lights and provides a sound as the alarm	Y	N/A

		to act as the alarm			
5	That the pi displays the temperature on the OLED	That it displays and updates the temperature on the OLED display	It did as expected	Y	N/A
6	Pause button on adafruit	The button when clicked should display a message on the terminal saying it is paused and should stop gathering temperature and close the display	It did as fully expected	Y	N/A
7	Run button on adafruit	Should go back to reading and sending the temperature and turn the OLED back on to display the temperatures	It did as expected	Y	N/A
8	Relay button on adafruit	The switch should activate a red LED on the board and turn on the relay	The relay turned on, as well as the LED to act as if controlling an environment controlling system	Y	The solution made was to remove the toggle variabls and just state its on and off state using “relay.value = 1” and removing the “relay = True” statement because it made a value determination error.

Gantt chart at the end product

https://docs.google.com/spreadsheets/d/1_81nbbLo7uG2_M6Db2ELXeiPpBZMoXao_kiNqLW3RMc/edit?usp=sharing