COMP6570 IoT Logbook

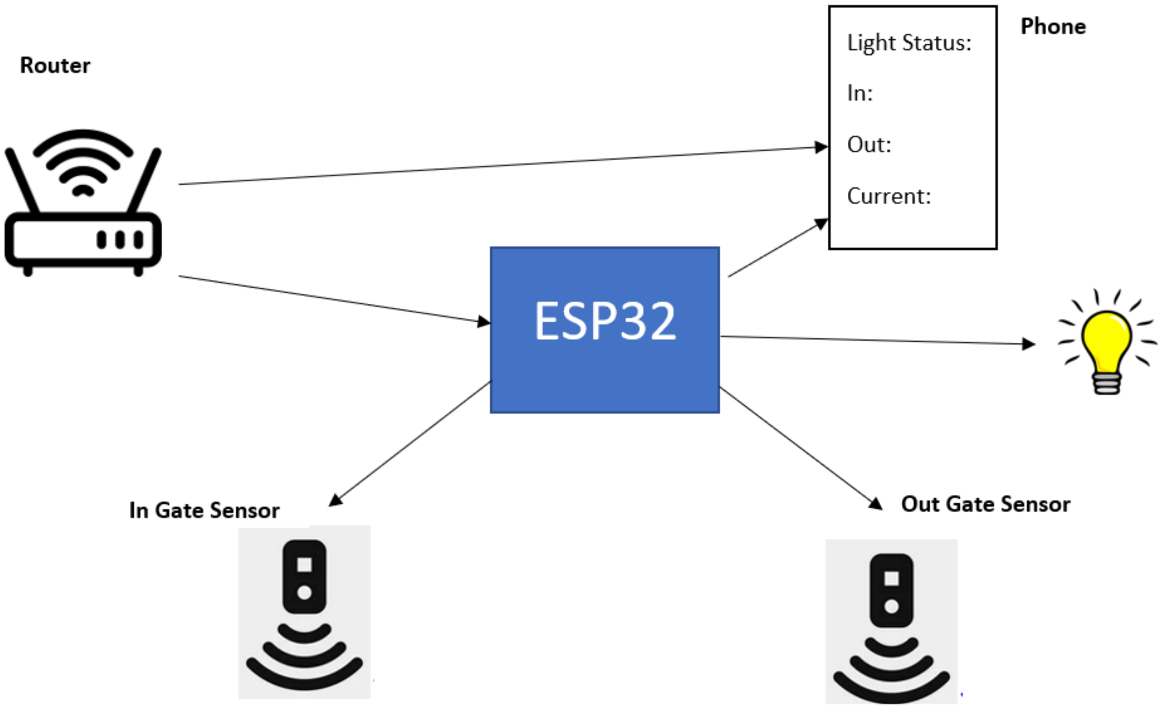
Initial Ideas and Planning

I want to create a system that can track how many people come into a room, come out of a room and currently in the room. Then depending on that information, it will turn a light off or on. This system could be useful in places where there is a separate gate for coming in and out of a room, such as a supermarket, or maybe different parts of a museum. It will help for saving power, as if there are no people in a room, there is no point in having the light switched on.

I'm thinking about using two ultrasonic sensors, as they good for detecting a change in distance (when someone walks through the sensing range). These sensors can also be used when the room is dark, which is handy if the light sources are going to be switched on. Ultrasonic sensors are also not affected my external factors such as dust, smoke and vapour, which are present in a real world environment.

I am thinking about using Wi-Fi connectivity as I want it to be able to connect to my phone to display the current data of the system. And as the range between the ESP32 and the phone could be quite large, it would be a better option over Bluetooth.

Diagram for the initial design idea from the poster



02/11/2022

I've started researching both into Bluetooth and Wi-Fi connectivity for the ESP32 and figuring out which when would be better for my system. I'm still very sure that Wi-Fi will be better for the system but I can't find a good app for my phone to get my system connected with. For now, I have set up both Bluetooth and Wi-Fi connectivity for my ESP32 so in the future when I decide I have both codes written.

09/11/2022

I have picked up my ultrasonic sensors and am now trying to get the connected. However, I wasn't aware of what all the pinouts on the ESP32 were and didn't pick up any wires for the sensors.

Instead of connecting the sensors, I researched what items I needed to get to make my system functional. I need a breadboard, around 10-12 Jumper wires and an LED.

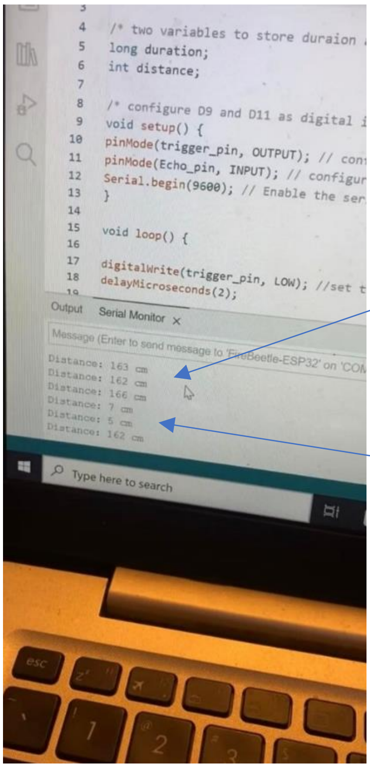
11/11/2022

I have gone to pick up my items from the shed, however, they didn't have a breadboard for me, so I picked up some additional female to male jumper wires to start with to get the sensors connected. I don't think the breadboard will be much of an issue for now, as I still think I can connect both of my ultrasonic sensors to the ESP32.

16/11/2022

I have successfully managed to connect my first ultrasonic sensor to my ESP32 and have the distance print to my serial terminal. It took quite a bit of trial and error with the pinouts and writing the correct pins in the code. I failed to connect my second ultrasonic sensor and I'm not sure why it won't display yet.

The distances printing to serial terminal for 1 sensor:



```
/* two variables to store duraion
long duration;
int distance;

/* configure D9 and D11 as digital i
void setup() {
pinMode(trigger_pin, OUTPUT); // con
pinMode(Echo_pin, INPUT); // configur
Serial.begin(9600); // Enable the ser
}

void loop() {

digitalWrite(trigger_pin, LOW); //set t
delayMicroseconds(2);
```

Output
Serial Monitor ×
Message (Enter to send message to 'FireBeetle-ESP32' on 'COM
Type here to search

```
Distance: 163 cm
Distance: 162 cm
Distance: 166 cm
Distance: 7 cm
Distance: 5 cm
Distance: 162 cm
```

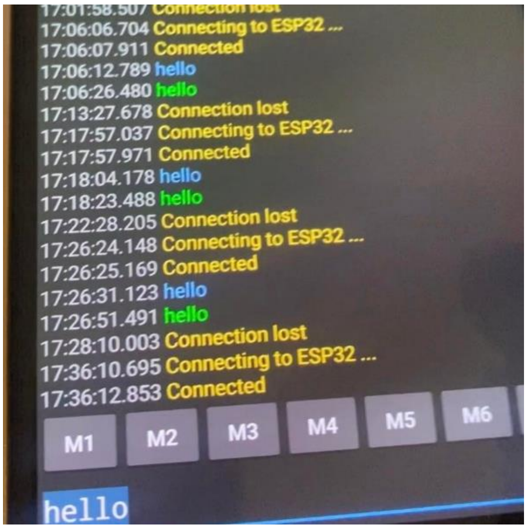
Distances

Distance when my hand was over the sensor

18/11/2022

With the video demonstration coming up, I need an app that shows I’ve gotten my connectivity working. I haven’t found a good app with Wi-Fi yet, however I found an app called Bluetooth Serial Terminal, that I have managed to connect with.

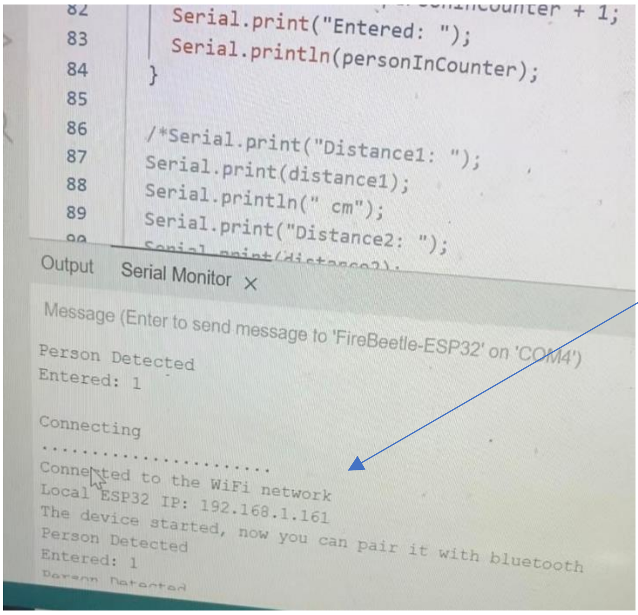
Here is a photo that shows communication between the app and the serial monitor on Arduino using Bluetooth connectivity:



```
17:01:58.507 Connection lost
17:06:06.704 Connecting to ESP32 ...
17:06:07.911 Connected
17:06:12.789 hello
17:06:26.480 hello
17:13:27.678 Connection lost
17:17:57.037 Connecting to ESP32 ...
17:17:57.971 Connected
17:18:04.178 hello
17:18:23.488 hello
17:22:28.205 Connection lost
17:26:24.148 Connecting to ESP32 ...
17:26:25.169 Connected
17:26:31.123 hello
17:26:51.491 hello
17:28:10.003 Connection lost
17:36:10.695 Connecting to ESP32 ...
17:36:12.853 Connected
```

```
hello
```

The device does also connect with Wi-Fi as I am still trying to find a better app to use:



```
                              nCounter + 1;
      Serial.print("Entered: ");
      Serial.println(personInCounter);
    }

    /*Serial.print("Distance1: ");
    Serial.print(distance1);
    Serial.println(" cm");
    Serial.print("Distance2: ");
    Serial.print(distance2);
```

Output
Serial Monitor ×
Message (Enter to send message to 'FireBeetle-ESP32' on 'COM4')

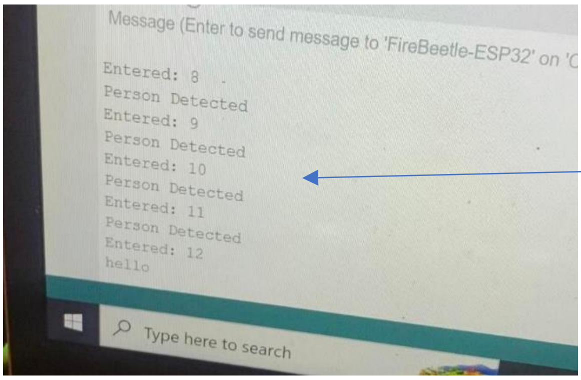
```
Person Detected
Entered: 1

Connecting
......................
Connected to the WiFi network
Local ESP32 IP: 192.168.1.161
The device started, now you can pair it with bluetooth
Person Detected
Entered: 1
Person Detected
```

Wi-Fi Connection

As I’m still not sure why the second sensor isn’t working, I’ve set up the first sensor to increment a counter representing the people coming in. I’ve set it that if the sensor detects a distance below 20cm, it will increment the counter. I set it to 20cm only for testing purposes, and in the case of the end project, it will depend on the door/gate it is covering.

Here are the outputs of the incrementations:



```
Message (Enter to send message to 'FireBeetle-ESP32' on 'C
Entered: 8
Person Detected
Entered: 9
Person Detected
Entered: 10
Person Detected
Entered: 11
Person Detected
Entered: 12
hello
```

Person in counter

19/11/2022

I have figured out why I couldn’t get my second ultrasonic sensor working, and it’s because it isn’t powered. I thought before that the 3v3 pinout in the ESP32 could be used to power the sensor, but I was wrong. It requires a 5V pinout, where the ESP32 only has 1. To fix this, I need a breadboard, or find another way to power the sensor.

20/11/2022

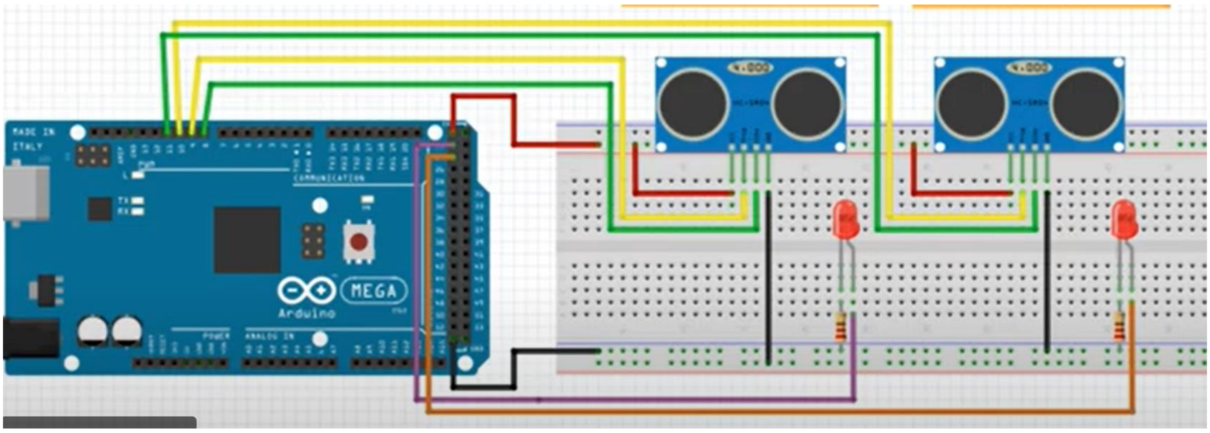
I haven’t yet gotten an external LED yet, but I have established a connection with the built in LED. I have made it so when the sensor detects an object below 20cm away from it, the LED will turn on, and then turn off it its above 20cm. This is a good start and has built my knowledge on controlling an LED for when I must add more commands for it later on.

25/11/2022

In my 1 on 1 chat for the video demonstration with a lecturer, I asked for advice on an app I could use to display my data from the system. He recommended an application called Blynk, which will use Wi-Fi connection and is extremely customisable to display the information I want. Other than Blynk, I could create my own, but I don’t have the time to do that so I will start looking into how to use Blynk. This solidified my decision to swap back to Wi-Fi over Bluetooth, as Blynk requires a Wi-Fi connection. It’s also good to swap back as my system would potential required a longer distance.

30/11/2022

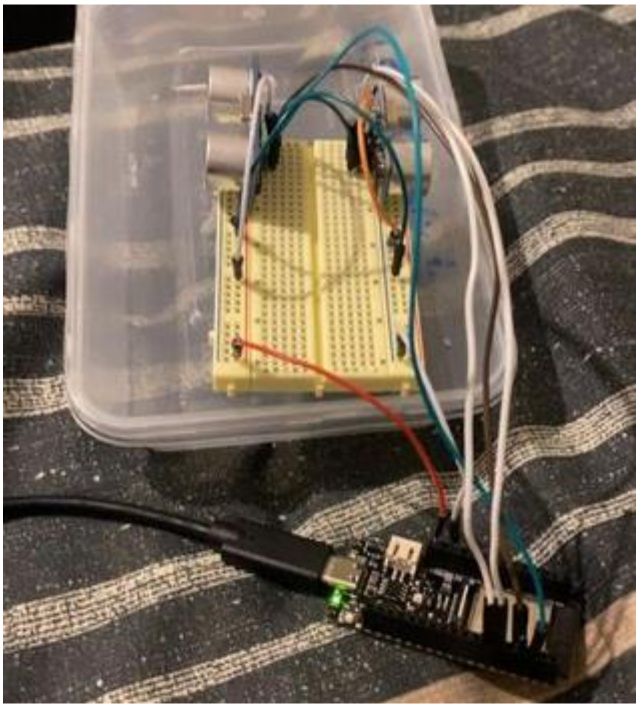
A breadboard finally became available from the shed. I then researched online of how to connect 2 ultrasonic sensors to an ESP32, and found this design that I would follow:



This design is from a video by DIY Guy Chris (https://www.youtube.com/watch?v=Lxh7hh_gl8Q), about connecting 2 ultrasonic sensors to an ESP32. In his diagram he uses 2 LED's where I only use 1 so I thought it would be a good idea to roughly follow his design.

05/12/2022

I have connected my second ultrasonic sensor to my breadboard and ESP32, and it currently looks like this:



However, I am running into a weird behaviour/error from my second sensor. The second sensor is always reading a distance of 0. I have tested both sensors individually and they both work fine, so I'm not sure what the issue is. I have triple checked all the code and I'm certain I haven't missed anything, and the ports are all correct as well. I wrote a message to the module convenor asking for some guidance as I am very stuck.

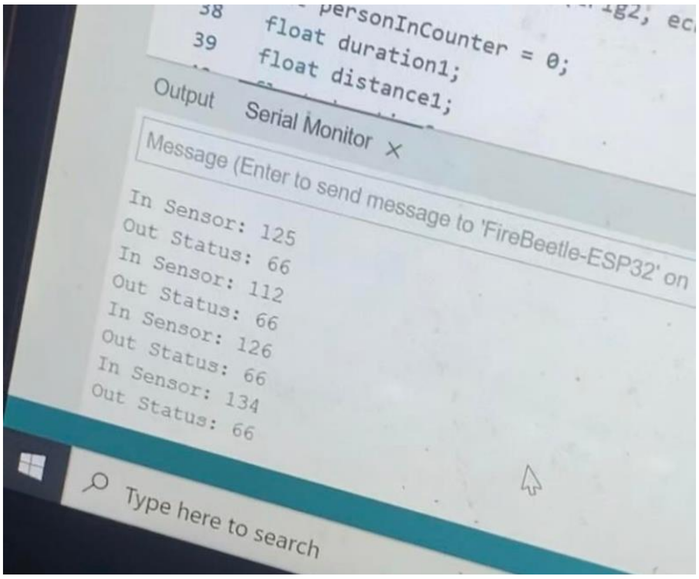
06/12/2022

The Module convenor has gotten back to me and told me that with ultrasonic sensors there is a chance of ‘cross talk’ between the two sensors, where 1 sensor picks up the signal transmitted from another sensor. He also suggested that I check both my sensors are connected to individual trigger and echo pins, which there already are. I don’t think it would be ‘cross-talk’ as even when I add a large delay between the sensors (such as 5 seconds), it was still reading 0. I will keep trying to fix this issue and my goal is to have it fixed within the next few days.

07/11/2022

I have finally fixed the issue with my second ultrasonic sensor. I changed the definition of the trigger and echo ports in my code form ‘D6’ and ‘D7’ to 14 and 13 respectfully. I’m not sure why this fixed the error, as the definition of the trigger and echo port for the first sensor remain as ‘D3’ and ‘D2’, so I find that weird. I am also now using the Ultrasonic library to get my readings.

A photo of the outputted distances:

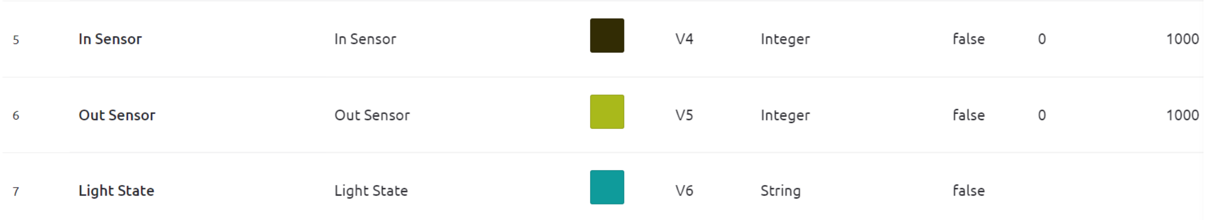


```
personInCounter = 0;
float duration1;
float distance1;
```

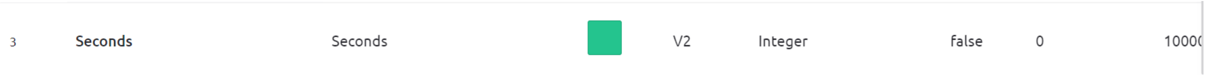
```
In Sensor: 125
Out Status: 66
In Sensor: 112
Out Status: 66
In Sensor: 126
Out Status: 66
In Sensor: 134
Out Status: 66
```

I then went on to set up a connection with Blynk. I found that this was a straight forward process and Blynk is fairly simple to use and very well documented. I connected each bit of information (runtime, people in, people out, light status) to a virtual pin in Blynk, and got it to display both on my phone and my laptop.

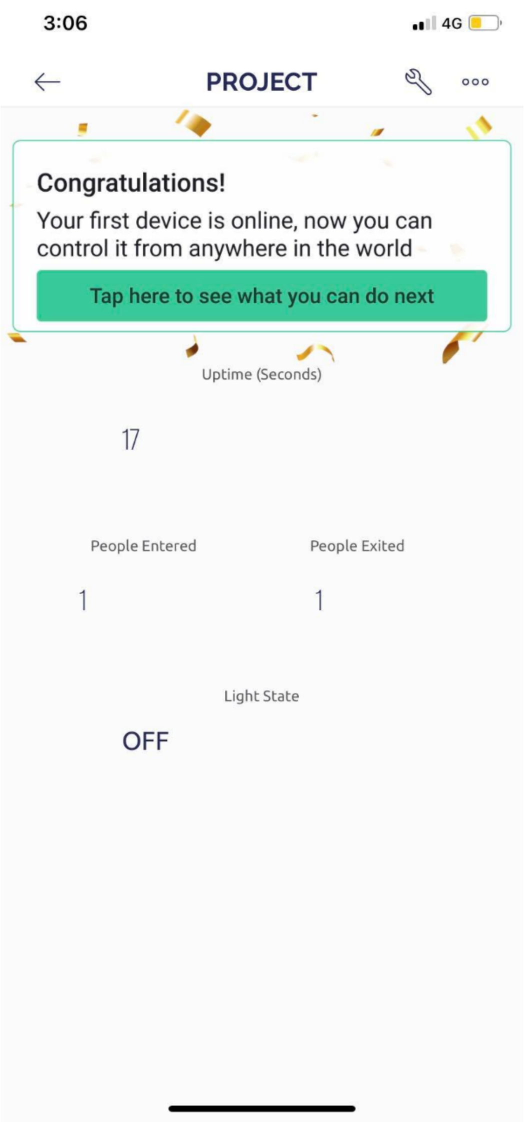
Here are some photos of the configurations and displays:



5	In Sensor	In Sensor	■	V4	Integer	false	0	1000
6	Out Sensor	Out Sensor	■	V5	Integer	false	0	1000
7	Light State	Light State	■	V6	String	false		



3	Seconds	Seconds	■	V2	Integer	false	0	10000



3:06

PROJECT

Congratulations!
Your first device is online, now you can control it from anywhere in the world

Tap here to see what you can do next

Uptime (Seconds)

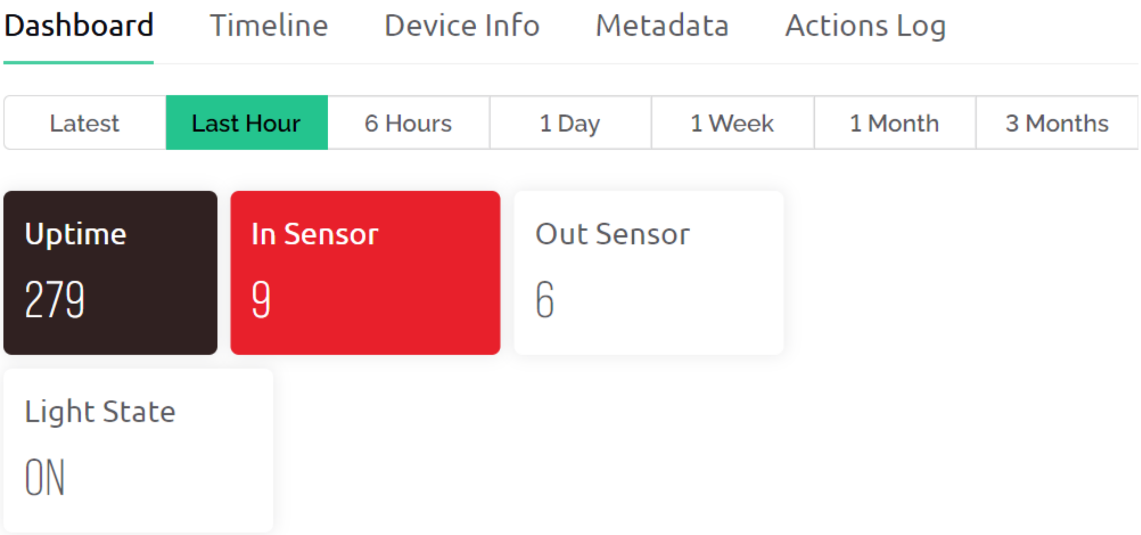
17

People Entered: 1

People Exited: 1

Light State

OFF



Dashboard | Timeline | Device Info | Metadata | Actions Log

Latest | Last Hour | 6 Hours | 1 Day | 1 Week | 1 Month | 3 Months

Uptime: 279

In Sensor: 9

Out Sensor: 6

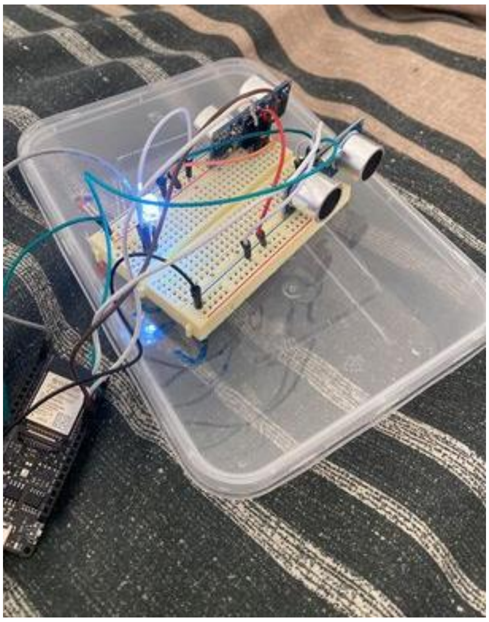
Light State: ON

In addition to this, I added some simple checks to see if an object has been detected by the sensors. I am still using 20cm as the distance for testing purposes, and each sensor will increment its own counter variable depending on if something was detected. The light status is calculated by checking to see if the person in counter is equal to the person out counter, as this would mean there are currently no people inside the room.

08/12/2022

I managed to set up my LED into my system using the breadboard. I thought it would be more complicated, however, I just had to connect one side of it to the ground port and the other to a GPIO pin to control it. So now the LED light will turn on if it calculates that there is someone in the room and turn off it there isn't.

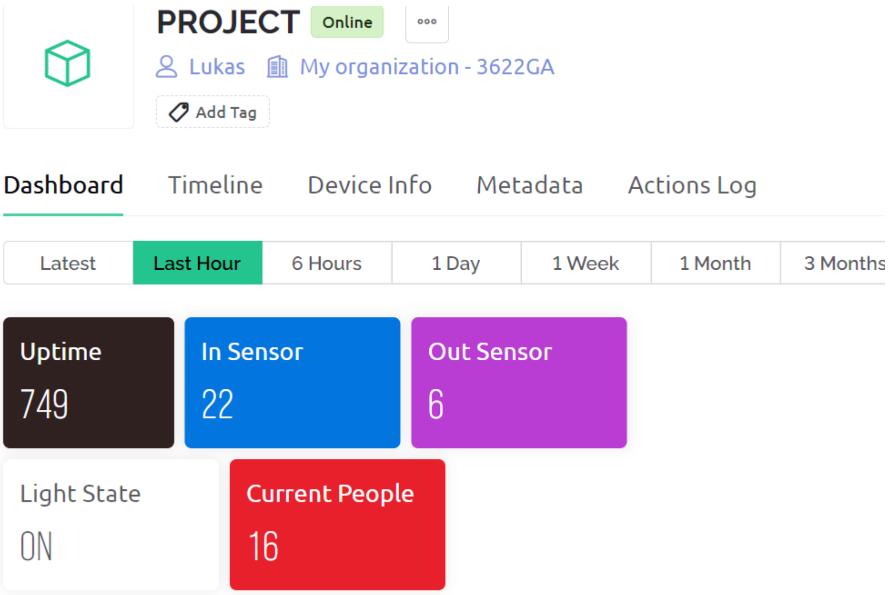
Here is a photo of the system with the LED on:



15/12/2022

I added an additional display on Blynk where it displays how many people are currently in the room. This is calculated also adding 1 to the variable if someone walks in and taking 1 away if someone walks out.

These are the photos of the outputs for web and mobile on Blynk:



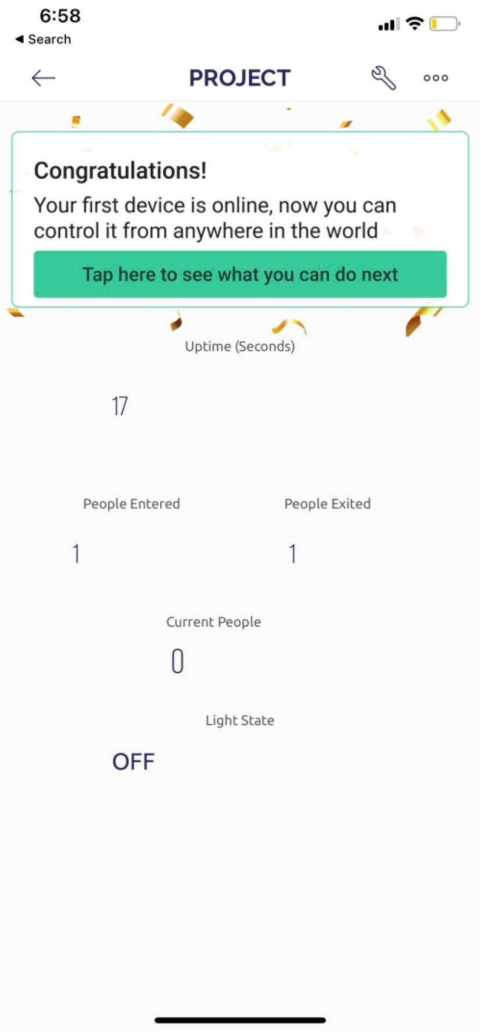
Uptime 749

In Sensor 22

Out Sensor 6

Light State ON

Current People 16



Congratulations!
Your first device is online, now you can control it from anywhere in the world

Uptime (Seconds)
17

People Entered 1

People Exited 1

Current People 0

Light State OFF

Reflection

Overall, I really enjoyed the whole process. I have never done anything related to IoT or Arduino before, so I found it very interesting to learn something new and experiment with sensors. I approached this piece of work by picking a system that covered a lot of the aspects of IoT, such as interacting with multiple sensors, using an output LED, and also displaying the information and calculations gathered online. This is why I picked this project over others, as I thought it wouldn't be impossible to finish, but it would challenge me and drive me to work harder.

What I found easy about the project was creating a Wi-Fi and Bluetooth connection on the ESP32. It was extremely well documented, and there isn't much room for error when writing the code. Along with this, I also found using Blynk surprisingly simple to implement, as the application gives you a lot of help creating a project for the first time, and the concepts are very similar to the GPIO pins on the ESP32, only difference is that its virtual.

However, this project did come with its challenges, with the main one being connecting 2 ultrasound sensors and having them working at the same time. I initially couldn't get the second one working for a few days because I was trying to use the 3v3 pinout as a power source and didn't realise I needed a breadboard to power multiple parts at the same time. And then I came into the issue of my second sensor always reading 0, which again, took a few days to figure out. I am still not 100% sure why changing the trigger and echo pins definition in the code to its respectful numbers rather than 'D6' and 'D7' make them work.

Looking back at my project, I would definitely do it differently if I were to do it again. I would put a lot more time into planning my project and researching all the parts I need. I kept having to briefly stop my work as I didn't have the parts, such as when I didn't have the jumper wires, or when I didn't have the breadboard. If I had done more research into it and drafted a sketch or diagram for it towards the start of my project, it would have gone a lot smoother and faster, meaning I could focus more on optimisation and potentially adding more features, such as potentially a graph to display the data over time. I might have also tried using infrared sensors and see how they work to compare it two ultrasonic. I'm confident I went for the right option, however, I would be nice to try out infrared as well.

In addition, I learned many things over the space of this project. One thing was to always ask for help or outsiders' opinions when I am unsure about something. There were several occasions where I would ask someone with more knowledge than me for guidance, even when I felt the issue was an easy fix, but I couldn't figure it out. Some examples of this are when before creating my poster, I asked someone in the shed how to power my second ultrasonic sensor and he said I just needed a breadboard, or when I messages the email convenor for guidance on why my second sensor was reading 0. If I didn't ask, the whole project would've gone a lot slower because all this content was very new to me. Another thing was I learned a lot of the basics for IoT and Arduino, and now feel confident to take on some larger projects.

I would like specific feedback on maybe what other additional features I could add to my project if I were to keep working on it. As someone new to IoT, this project has really piqued my interest and I find creating mini projects like this exciting and is something I would want to continue in my free time. In addition to this, what optimisation could be done on my project. I feel like my project at its current state would only work in a perfect world environment. For example, is someone stopped in the middle of the sensor, the system would count them as multiple people, if is someone was to 'exit' out the 'entrance' gate, it would count as 2 entrances rather than 1 entrance and 1 exit. If I could make changes to my system that could avoid these errors, I would be interested to hear what they are.

I do have some ideas for future developments and how it would work in a real-world environment. For example, a shop/room could use a light bulb with Wi-Fi connectivity to control it, and maybe multiple ESP32’s for the in and out gates if they were far away. Another possibility would be to use a radar sensor for if a shop only had 1 doorway, as it can detect what direction people are moving in. Lastly, I would want to add more functionality on Blynk, to display data over time. This would be very useful for businesses, as they could track how many people are coming into their shop and at what times.

Overall, this project came with its difficulties, but I found it very enjoyable. It was a great way to introduce me to IoT, and I look forward to tackling projects in the future.