Design:

There is a rise in use of the toilet in the hostels ever since the lockdown started. More students are now spending their day within the hostel premises therefore rooms with common toilet are sometimes all occupied. Since toilets are on one side of the corridor, students of the rooms of the other side of the corridor must walk all the way to the other side just to see that all toilets are occupied now. A solution is made to see if the toilet is busy or not using the IOT solution. The solution used Esp32 as a controller with IR sensor (TCRT5000 attached on the LM39 potentiometer). The basic workflow of the solution is as shown below in the figure.

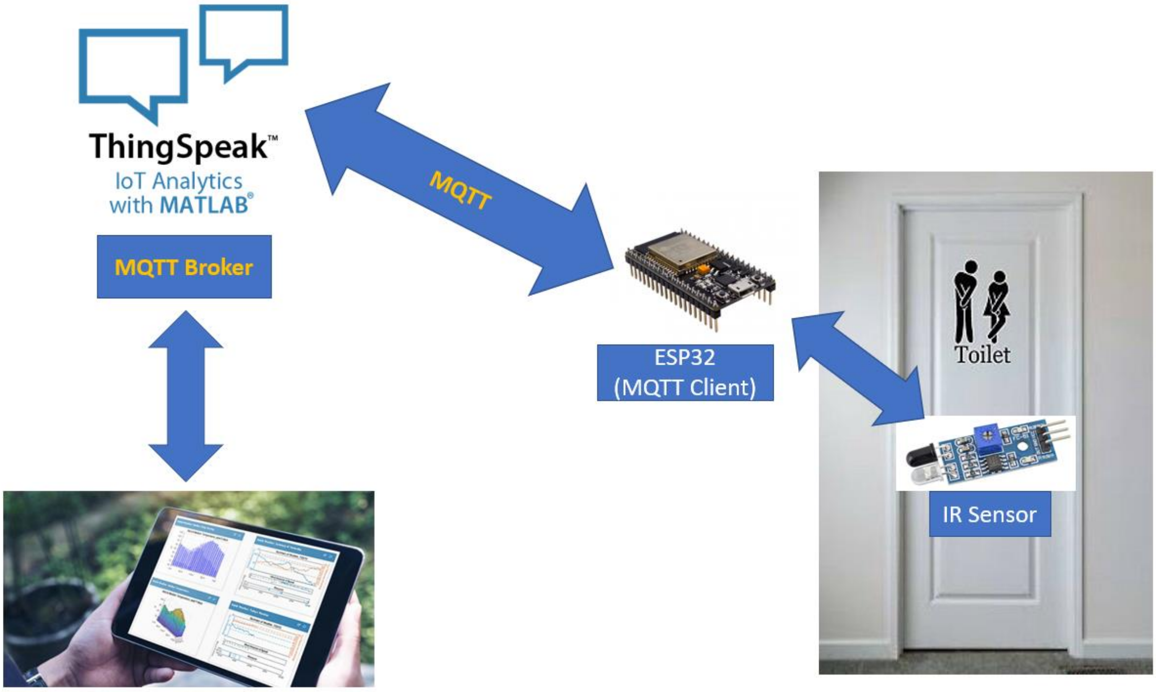

Figure 1: Basic Flow of the system

The Esp32 is a microcontroller attached with Wi-Fi ability in one single circuit which is configured with micropython framework. The Esp32 is used to send the IR sensor signal to the Thingspeak cloud which is then visualized on the Thingspeak UI on its website. The Thingspeak is an IOT solution provided by MATLAB [4]. It uses the MQTT protocol. The MQTT stands for Message Queuing Telemetry Transport. It is becoming one of the most popular standards for IOT solutions as it allows the machine-to-machine communication with small footprint code. The communication consists of Broker, Client and Subscriber in contrast to traditional method where there is a direct communication from one point to another. The advantage of this is that lower bandwidth of internet is used and lower latency [7].

The publisher is someone (or a device such as esp32) who sends the data to the server cloud. In this case this server is the Thingspeak server. Now, if anyone over the internet want to see this data, they can do so by subscribing to the Thingspeak server. To access the specific data from the server, channel ID is used. Once the subscriber has taken the data, it can be used for the visualization in any way possible. If the subscriber wants to send back the action to the publisher, in this case, the scenario is opposite. The subscriber will act as a publisher and publisher will act as a subscriber. When the code is written in the esp32, it is given the channel ID so that when data is sent to the server, it is attached with the channel ID. Besides channel ID, there are two API keys namely Read API Key and Write API Key. The Write API key is used by the esp32 so that when it sends the data to the Thingspeak server, it can tell the server that it wants to write data (using write API key) on that specific channel ID. The Read API is used when the subscriber wants to access the data.

Development:

The code written in micropyhton firstly imports the network library which is used to connect to the home Wi-Fi. Then some further libraries are imported which are then used to initialize the sensor pins and variables. The cnct object is used to store the connection information to the client. The connection to the client is made using the MQTTClient() function. The inputs are the server's name which is the mqtt.thingspeak.com and the el-4012_toilet_status is used as a topic. The ssl=True option tells the function that it is used to improve the security of the system. This option encrypts the information between the server and the client. Although SSL means Secure Socket Layer, but it is commonly known as Transport Layer Security (TLS) [5]. The client messages the server to initiate the communication then the server sends the encrypted version of the certificate. The client then checks the certificate and then it creates and sends back the encrypted key to the server [3]. The TLS connection takes place after the TCP (Transmission Control Protocol) connection which is a three-way connection protocol.
The data exchanged in TLS layer is the application data whereas TCP connection only ensure the secure communication path for TLS layer. The TCP starts with client sending the SYN (Synchronize) message to the Server. The server replies with the ACK-SYN (Acknowledge Synchronize) message where ACK tells the client that it has received the message and SYN tells that it is acknowledging that specific sequence of number code. In the third step, the client sends the ACK (Acknowledge) message which means now a secure path is made between the server and client.
The object cnct is then connected to the server of the specifications mentioned above. The Thingspeak server will store the data with the topic name which is supposed to be unique. The Thingspeak uses the format "channels/" + channel_id + "/publish/" + write_api_key as a topic name to publish the data on the server [8]. This way the subscriber, using the Thingspeak UI, can access the data only when it has the same exact channel ID and API Key.

Now, the lcd screen is initialized and then the IR sensor. Then comes the main part where the code will run for rest of the time until it is on. The while loop is an infinite loop to keep the program within it therefore the publishing process can be automated. Within the while loop the first thing is that the signal value from the IR sensor is taken and displayed on the lcd screen. Then this value is checked for the specific value using the IF condition. If the value of the sensor is less than 2000, this means the toilet is occupied otherwise it is open. Then this data is published on the Thingspeak server using the topic with specific format mentioned above.

The data on the server is seen on the Thingspeak website when it is login with the specific username and password. Only those channels are shown in this account which are created by this user. Once the data is published from the esp32, it can be seen on the Thingspeak private view. It means that only the user can see the data. Thingspeak comes with the great ability to use this data to make useful graphs or indications. Therefore, the data subscribed to, is used to make line plot to see the variation in the sensor value with time as shown in figure 2 below. To show weather, the toilet is occupied or not, the indicator is used which goes red whenever the value is below 2000 (whenever the toilet is closed/occupied), in figure 4 below. The meter is also used to see the variation of the value when the toilet is closed or open. The figure 3 shows the meter gives a better visualization than the line plot since the line plot contains every single value and some might be incorrect.

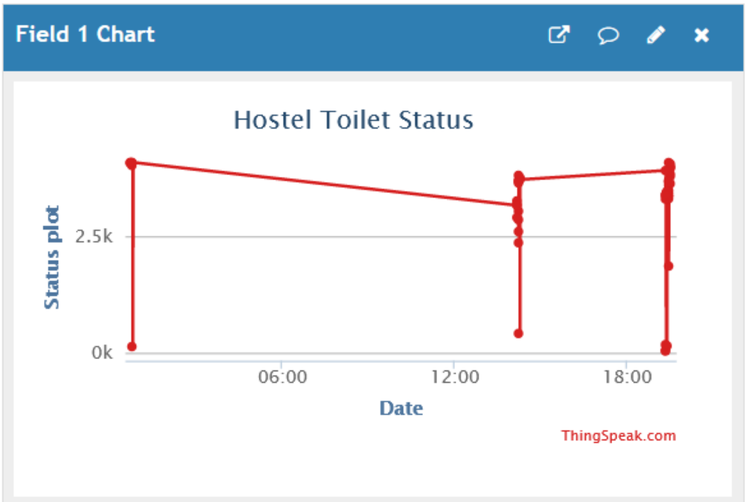

Figure 2: Line plot of Thingspeak

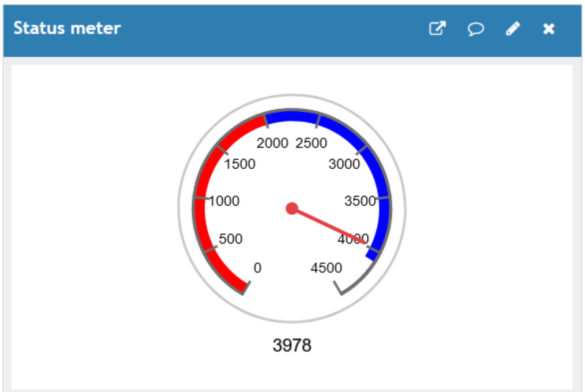

Figure 3: Status meter

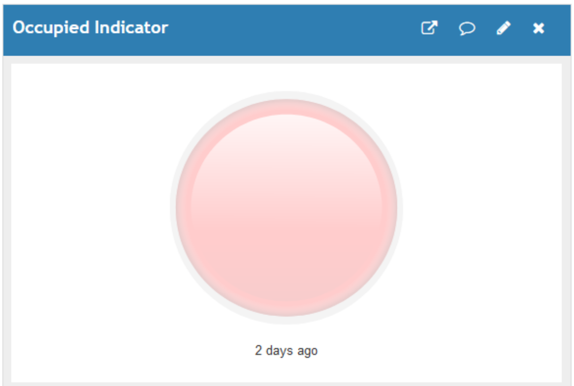

Figure 4: Occupied Status

Testing:

The code is tested using the home toilet and it shows the delay of 15-20 seconds to see the data on the Thingspeak. Since, not everyone will open the browser to check the toilet status, there is a mobile app of the Thingspeak, called Thingview, is used. This way anyone can check the toilet status easily. The Wi-Fi connectivity shows error while connecting sometimes so that can be fixed later fixed by checking the connectivity status. If the status is false, then connection try must be repeated until successful connection. To ensure the working of the publishing part, I quickly refreshed Thingspeak page but for first few seconds it did not show anything. It took it about a minute and half to start visualizing the data in Thingspeak UI. I then used a timer to check time to see the data on Thingspeak from the time I changed the value on the IR sensor. The duration was 20 seconds on average. It was then confirmed on the internet that it is the average delay when using free version of Thingspeak.

When the sensor is practically attached to the bathroom door, it sometimes did not give the correct value once closed. It was then concluded that sometimes the light of the bathroom interfered with the sensor light. The solution for this is that black plastic cover around the boundary of the sensor lights could fix the problem.

Evaluation:

For the demonstration, only one toilet is used, in reality there are several toilets. In that situation, one IR sensor is connected to each toilet door and they are then connected to the esp32. This way maximum of 16 IR sensors can be connected to the single esp32. On average hostels or public places have 2-8 toilet rooms together so using multiple IR sensors with one esp32 will suffice hence reducing the cost of using multiple processors. This will also conserve the power that would otherwise be used to power each esp32. One unique innovation in this system is the use of TCRT5000 sensor which does not require any sensor libraries therefore reducing overall code size.

In terms of size, the sensor is only 3x2 cm which is small enough to install above the door handle without any restricted movement. Since the sensors will be stationed in one place and the esp32 as well, therefore, the system can be directly powered from the wall. The entire system will take 5V 240mA when the Wi-Fi is being used. The further advancement to the system is that it can be set to enter the deep sleep mode when not in use. But this idea can only be implemented in places where toilets are not in use at specific times such as office. The esp32 can be coded to go to deep sleep mode for hours the office is not in use at all. This way the system will only use up-to 30mA of current hence conserving power.

The cost of the system:

- 4 IR sensors = 4 £ (assuming there are four toilets together) [6]
- ESP32 = 8 £ [2]
- 18 AWG connecting wire roll = 6 £ [1]

Total = 18 £

This is the basic initial cost of the entire system. If there is a need to increase the data transfer rate, then Thingspeak subscription of 475 £/year can be purchased.

References:

[1] 18 Gauge Silicone Wire 16.4 Feet (8.2 Feet Black and 8.2 Feet Red) Soft and Flexible Low Impedance,18 AWG Silicone Wire 150 Strands of 0.08 mm Copper Wire for RC Car, Drone, Airplane Battery etc . Available: https://www.amazon.co.uk/Silicone-Flexible-Impedance-Strands-Airplane/dp/B07CZ9TM79/ref=sr_1_3?dchild=1&keywords=18+awg+wire&qid=1618160161&s=diy&sr=1-3.

[2] AZDelivery ESP32 Dev Kit C V4 NodeMCU ESP32 WROOM-32 WLAN/WiFi Development Board including E-Book! (Successor module of the ESP32 Dev Kit) . Available: https://www.amazon.co.uk/AZDelivery-ESP-32-Dev-including-eBook/dp/B07Z83MF5W/ref=sr_1_6?dchild=1&keywords=esp32&qid=1618160282&sr=8-6.

[3] How Does SSL Work?. Available: https://www.entrust.com/resources/certificate-solutions/learn/how-does-ssl-work.

[4] MQTT API. Available: https://uk.mathworks.com/help/thingspeak/mqtt-api.html.

[5] What Happens in a TLS Handshake? | SSL Handshake. Available: https://www.cloudflare.com/en-gb/learning/ssl/what-happens-in-a-tls-handshake/.

[6] Youmile 5-Pack IR Infrared Obstacle Avoidance Sensor Module for Arduino Smart Car Robot 3 wire . Available: https://www.amazon.co.uk/Youmile-Infrared-Obstacle-Avoidance-Arduino/dp/B07PY3CVSV/ref=sr_1_9?dchild=1&keywords=Infrared+Sensor&qid=1618159931&sr=8-9.

[7] TRACY, P., November 29, 2016-last update, MQTT protocol minimizes network bandwidth for the internet of things. Available: https://www.rcrwireless.com/20161129/fundamentals/mqtt-internet-of-things-tag31-tag99.

[8] WISINTAINER, M., December 07, 2018-last update, MicroPython – Getting Started with MQTT on ESP32/ESP8266. Available: https://randomnerdtutorials.com/micropython-mqtt-esp32-esp8266/.