IOT HUB CONNECTED TO A CLOUD

Project Group No 16

E/12/162

E/12/302

E/12/376

A Brief Introduction to the Problem

Smart Devices are connected to an environment to facilitate our daily requirements and improve our quality of life. The platform which connect smart devices to connect to that environment is called the IOT hub. It includes data management, security and privacy, pluggable protocols, services and application which fits our needs. IOT hub is the device which will be communicate with IOT backend and to other smart devices which cannot talk to IOT backend directly.

In this project the main objective is to make such a platform (IOT hub) which is,

- Modularized as far as possible.
- Each component can be work independently.
- Able to replace with modified model without disturbing other components.
- Able to communicate through Wi-Fi, Ethernet, USB, Serial Com, Bluetooth.
- Which support standard device discovery protocol such as upnp, zigbee.

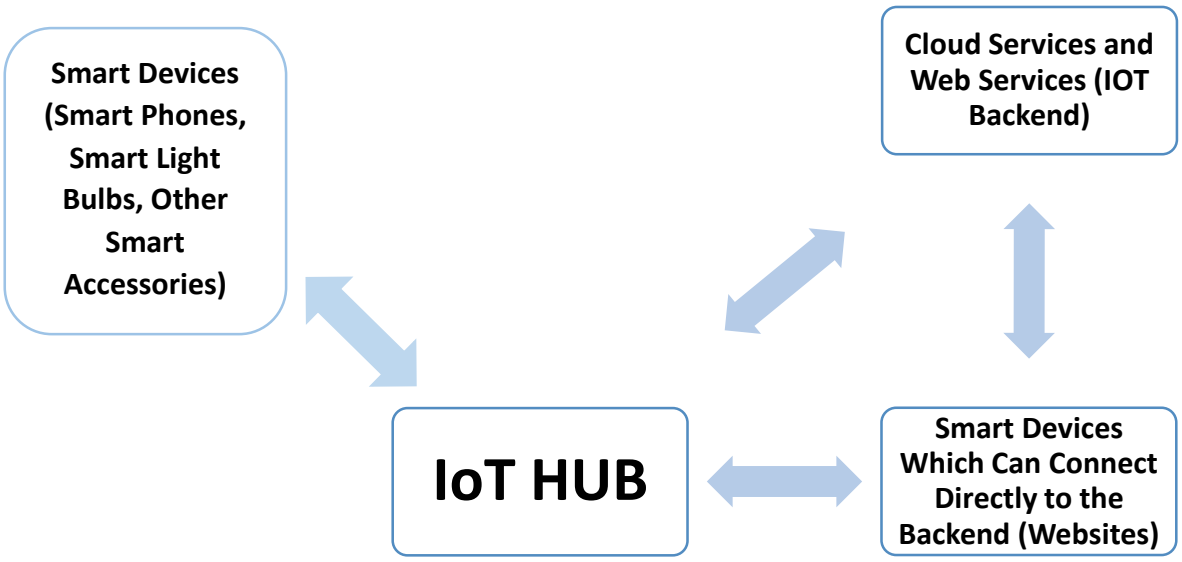

FIGURE 01: Brief Explanation of IOT Hub

An Approach to the Solution

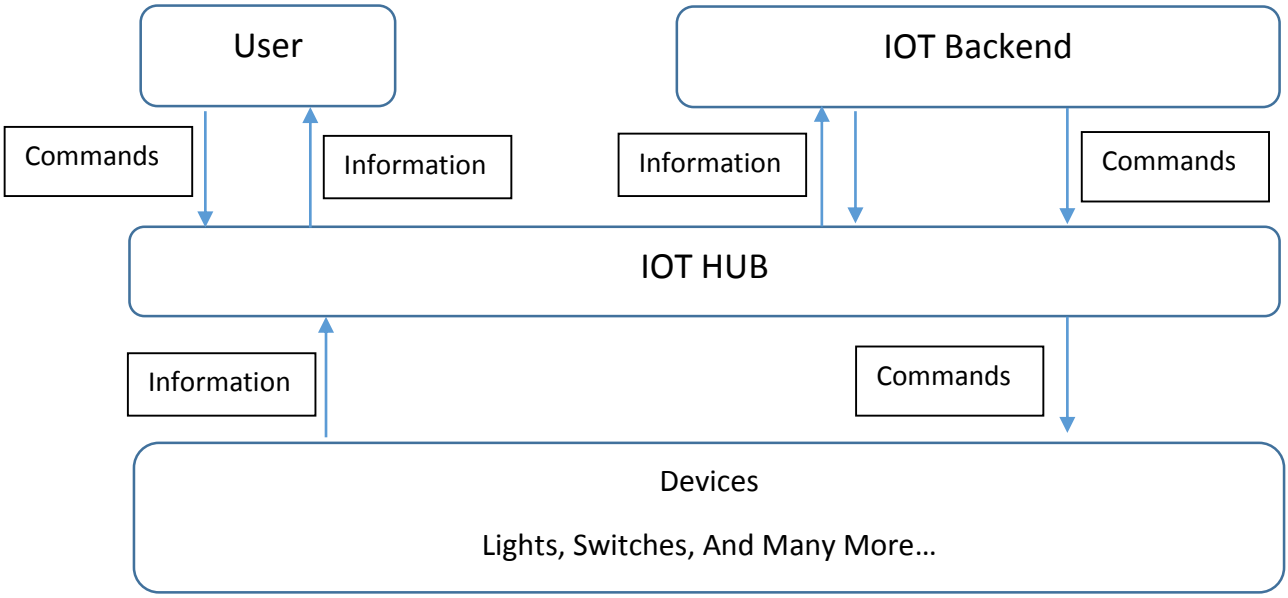

FIGURE 02: Architecture of the How IOT Hub Works and Data flow

As the solution for the above problems an IOT Hub was made and the Architecture of the IOT HUB is shown in FIGURE 02. As in above diagram IOT Hub plays the role of a bridge between IOT backend, Users and IOT Devices.

For an example let's consider the situation an user is trying to turn on an Smart light to do this,

- First we have to go to the mobile app and search for devices then subscribe for the relevant device. Devices discovery was done by IOT Hub and send info to app to user to subscribe for the device.
- Then User gets the Information about device's current status. For an example If an Smart light is on IOT backend sends info to user via IOT Hub saying the relevant lamp is on.
- If the user wants to turn off the lamp he has to do it using smart app. When the user gives the command to turn off the lamp and then IOT hub forward the command to the backend.
- Then at the backend the command was intercepted and intercepted message/command sends to the IOT Device vi IOT

Hub. When the intercepted message reach to the device and is executed the command and succeeded device sends an status update to the backend. After message reaches the backend and when the current status is updated it sends a status update to the user.

- Some applications like fire alarm devices always communicate with the backend without involving a user and status is updated continuously. Then execute some functions regarding to the device api and user's rules. For an example if temperature raises more than 40 degrees of Celsius ring the alarm. When this incident triggered when backend gets the information that temperature is high then status is updated and execute specific function and ring the alarm and user is notified.
- Devices were discovered easily by device discovery protocols.
- To the continuous communication between devices, backend and IOT HUB less weight and more efficient and fast MQTT data packets were used.
- (For Programming) Python is used cause wide usage of python in internet, networking and its support to the coding of IOT and mostly because of Availability of huge amount of libraries.

Prioritized Project Backlog

1. Gathering of Requirements and Information.
2. Obtaining needed resources (Devices – Raspberry pi, Wi-Fi Modules, Needed Software and other resources)
3. Configure the Raspberry pi to communicate through USB and Ethernet.
4. Initiating communication with Wi-Fi (connecting raspberry pi and Arduino via Wi-Fi).
5. Make “Switch” option for controlling switches connected to Arduino.
6. Making The Smart Phone App
7. Programming of Third Party Devices
8. Establishing the communication Protocols – Use MQTT protocol to Communication
9. Configuring Ethernet ports to use MQTT data packets.
10. Configuring Wi-Fi support to use MQTT protocol.
11. Configure Raspberry Pi’s message to connect to an cloud-based message broker (CloudMQTT)
12. Establishing the communication with backend using MQTT.
13. Adding Bluetooth, Wi-Fi, Serial Com Support.
14. Adding support of standard device discovery support protocols.
15. Initiating the communication link to IOT Backend enabling Push/Pull communication from the backend.
16. Improving and enhancing the Smart Phone App
17. Debugging and Finalizing The project.

Plan for First Iteration

- **Gathering of Requirements and Information. -** Required Information and related information were gathered to make the future plan.
- **Obtaining needed resources -** (Devices – Raspberry pi, Wi-Fi Modules, Arduino boards, Temperature sensors, Needed Software and other resources).
- **Configure the Raspberry pi to communicate through USB and Ethernet.**
- **Initiating communication with Wi-Fi (connecting raspberry pi and Arduino via Wi-Fi).**
- **Make "Switch" option for controlling switches connected to Arduino.**
- **Making The Smart Phone App. –** Making the first part Smart Phone App to connect, configure, communicate with IOT Hub and the IOT Back end. The first Part include connecting smart devices, configuring them.

Plan for Second Iteration

- **Establishing the communication Protocols –** Use MQTT protocol to Communication
- **Configuring Ethernet ports to use MQTT data packets.**
- **Configuring Wi-Fi support to use MQTT protocol.**
- **Configure Raspberry Pi to connect to an cloud-based message broker (CloudMQTT)**
- **Establishing the communication with backend using MQTT.**
- **Programming of Third Party Devices.**
- **Adding support of standard device discovery support protocols.**
- **Adding Bluetooth, Wi-Fi, Serial Com Support.**
- **Initiating the communication link to IOT Backend enabling Push/Pull communication from the backend.**
- **Improving and enhancing the Smart Phone App**
- **Debugging and Finalizing The project.**

Time Frame

No	Tasks	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
1	Gathering of Requirements and Information. (E/12/162, E/12/302, E/12/376)	✓						
2	Obtaining needed resources. E/12/162, E/12/302, E/12/376)	✓						
3	Configure the Raspberry pi to communicate through USB and Ethernet. E/12/162, E/12/302, E/12/376)	✓						
4	Initiating communication with Wi-Fi. E/12/162, E/12/302, E/12/376)		✓					
5	Make "Switch" option for controlling switches connected to Arduino. E/12/162, E/12/302, E/12/376)		✓	✓				
6	Making The Smart Phone App.(First Stage). E/12/162, E/12/376)			✓				
7	Establishing the communication Protocols. E/12/162, E/12/376)				✓			
8	Programming of Third Party Devices. E/12/162, E/12/302, E/12/376)				✓			
9	Adding Bluetooth, Wi-Fi, Serial Com Support. E/12/162, E/12/302, E/12/376)				✓			
10	Adding support of standard device discovery support protocols. E/12/162, E/12/302, E/12/376)					✓		
11	Configuring Ethernet ports to use MQTT data packets. E/12/162, E/12/376)					✓		
12	Configuring Wi-Fi support to use MQTT protocol. E/12/162, E/12/376)					✓		
13	Configure Raspberry Pi to connect to an cloud-based message broker (CloudMQTT). E/12/162, E/12/302, E/12/376)						✓	
14	Establishing the communication with backend using MQTT. E/12/162, E/12/302, E/12/376)						✓	
15	Initiating the communication link to IOT Backend enabling Push/Pull communication from the backend. E/12/162, E/12/302, E/12/376)						✓	
16	Improving and enhancing the Smart Phone App E/12/162, E/12/302, E/12/376)							✓

17	Debugging and Finalizing The project. E/12/162, E/12/302, E/12/376)							✓

Retrospective,

1. Development of the Solution

- According to the time frame workload was divided and distributed among group members. And then product is made by following the project plan.
- Use new technologies to make the product more reliable and efficient.

Resources we used,

- Raspberry Pi B+ is used
- Arduino
- Wi-Fi Shield
- Ethernet Shield
- Raspberry pi camera
- An Wi-Fi Access point

Technologies we used,

- Raspian OS
- For Discover devices easily some protocol were used.
- To the continuous communication between devices, backend and IOT HUB less weight and more efficient and fast MQTT data packets were used.

- (For Programming) Python is used cause wide usage of python in internet, networking and its support to the coding of IOT and mostly because of Availability of huge amount of libraries.
- For easiness and mobility of the product Android app was made.

- Arduino IDE is used to program IOT devices and to configure them.

- Postman cloud-based message services is used to debugging the project.

- HTTP protocol is used to communication besides of MQTT.

2. Technical problems faced

- How to configure network in raspberry.
- We did not know GPIO programming so we had to start form the scratch.
- We had to learn how to use MQTT data packets, MQTT protocol and how it works from the beginning
- We had to lean about different apis and how they work to build our own api.
- We had to learn how to work with restapi from the scratch.
- When connection backend to the IOT HUB at first it became more difficult.
- Configuring MQTT Libraries to the arduino is hard.
- We cannot use threads in arduino.
- Strength of the Wi-Fi hotpot isn't enough to connect more than two device at the same time. We used an smart phone as an access point to the IOT hub and only two devices can be connected to the access point.
- We had to spend more time in programming and building our own smart devices to connect to IOT Hub and for testing and debugging

cause there are no any kind of smart devices available in our department.

- We had to program/code different apis for each of these devices to work.
- Making of Smart devices were limited cause limited amount of arduino boards we had and limited amount of raspberry pis we had.
- When video streaming we had problem configuring firmware of the Raspian OS.

3. Further improvements to the project

- For discover devices easily, and more efficiently standard device discovery protocols such as upnp, zigbee can be used.
- GPS positioning could be used to add new functions and new features. For an example we could use GPS to get our position and make things more easy such as opening the automated gate to the house automatically when owner ‘s car is approaching towards the home.
- Add features to IOT hub to connect to existing devices like normal light bulbs, normal TVs, etc.

- Connect many IOT Hubs and backends and make a system of intelligent environment. With this we can share information between backends and analyze the data and make our life easier. For an Example lets thought about an system of intelligent street cameras share their information with air transport system, when an accident occurs and blocked an road to airport information about accident goes to the street camera system and then to air transport system, it can delay flights if any passenger is blocked due to accident or if street cameras

share info with local authorities it is easy to find the cause of the accident or street camera system share info with public transport system then we could alert drivers about accident and give them alternative routes to reach to their destination quickly and safely vi GPS system in their cars and digital notice boards aside the roads.

- Use new technologies like Li-Fi to improve performance and efficiency.
- We could improve our quality of services by adding support of an personal assistant like apple's siri or Windows Cortana or Sirius in Linux.
- Add Voice assistant support.