**CHAPTER 3**

**DESIGN AND IMPLEMENT OF THE SYSTEM**

This chapter represents about the design and implementation of the proposed

system. It also suggests implementing the comparison between MQTT and HTTP.

**3.1. Overall Design of the System**

Overall design of data logging with MQTT and HTTP is as shown in Figure 3.1 and Figure 3.2.Both protocol connected to raspberry Pi board.

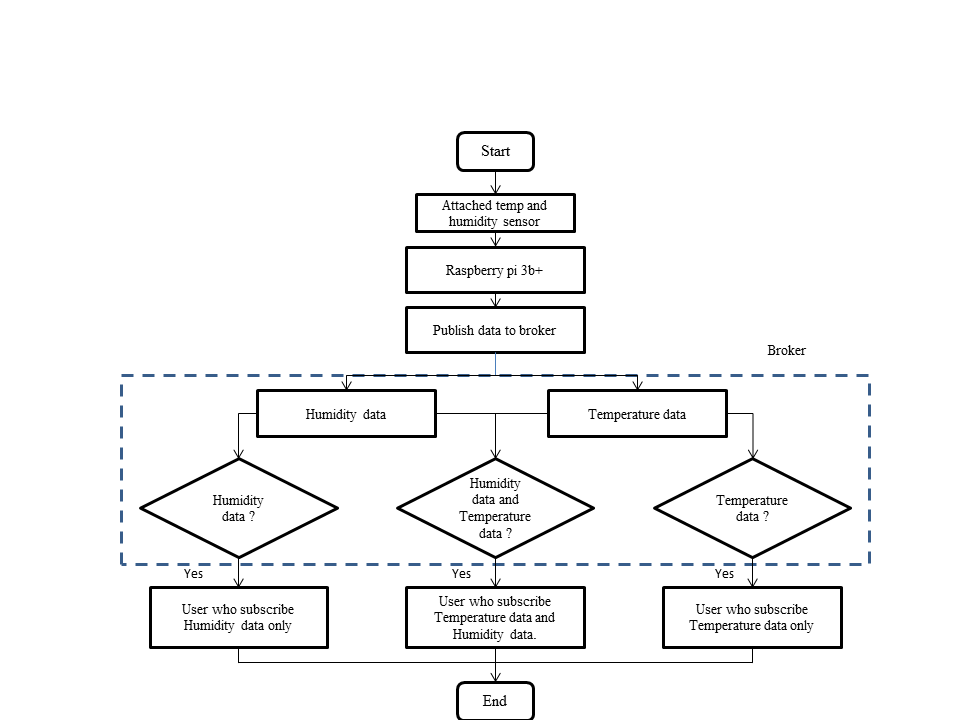


Figure 3.1 Overall design of data logging with MQTT

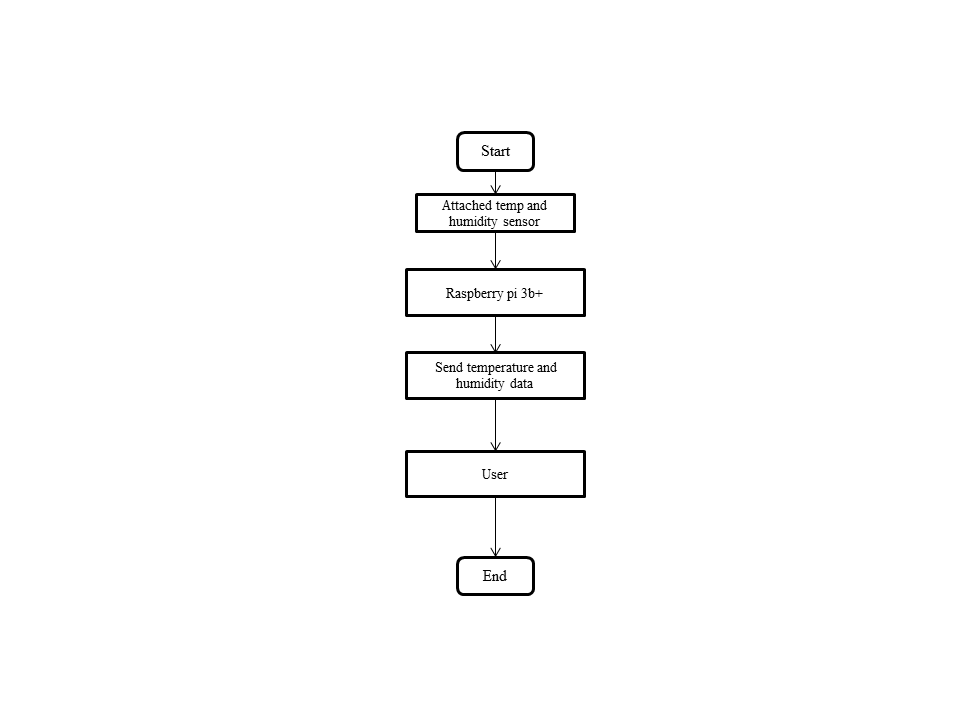


Figure 3.2 Overall design of data logging with HTTP

**3.2. Implementation of MQTT**

MQTT is a lightweight publish and subscribe system where you can publish and receive messages as a client. MQTT is a simple messaging protocol, designed for constrained devices with low-bandwidth. So, it’s the perfect solution for Internet of Things applications. MQTT allows you to send commands to control outputs, read and publish data from sensor nodes and much more. As shown in figure 3.3 and Figure 3.4.

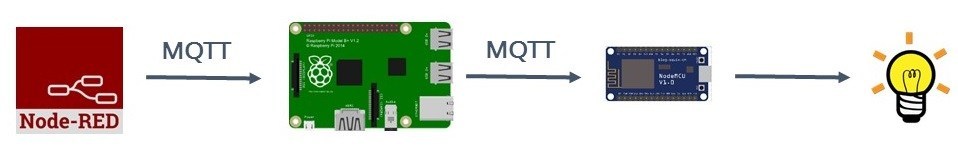


Figure 3.3 MQTT data output

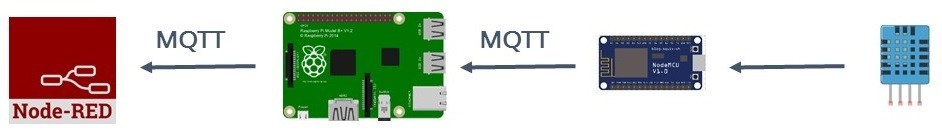


Figure 3.4 MQTT data input

Therefore, it makes it really easy to establish a communication between multiple devices. Figure 3.5 is architecture of MQTT protocol.

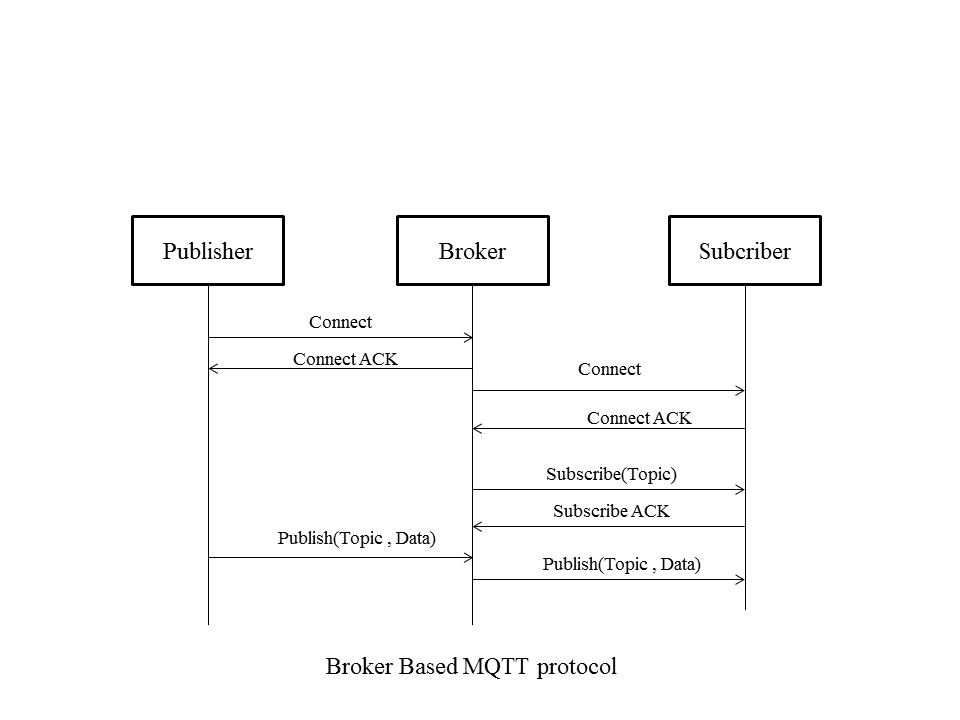


Figure 3.5 MQTT Architecture

**3.2.1. MQTT Basic Concept**

In MQTT there are a few basic concepts are as follows:

* Publish/Subscribe
* Messages
* Topics
* Broker

**3.2.2. MQTT – Publish and Subscribe**

The first concept is the publish and subscribe system. In a publish and subscribe system, a device can publish a message on a topic, or it can be subscribed to a particular topic to receive messages.



Figure 3.6 MQTT Publish/Subscribe

* In Figure 3.6, Device 1 publishes on a topic.
* Device 2 is subscribed to the same topic as device 1 is publishing in.
* So, device 2 receives the message.

**3.2.3. MQTT – Messages**

Messages are the information that you want to exchange between your devices. Whether it’s a command or data.

**3.2.4. MQTT – Topics**

Another important concept are the *topics*. Topics are the way you register interest for incoming messages or how you specify where you want to publish the message.

Topics are represented with strings separated by a forward slash. Each forward slash indicates a topic level. Here’s an example on how you would create a topic for a lamp in your home office in Figure 3.7.

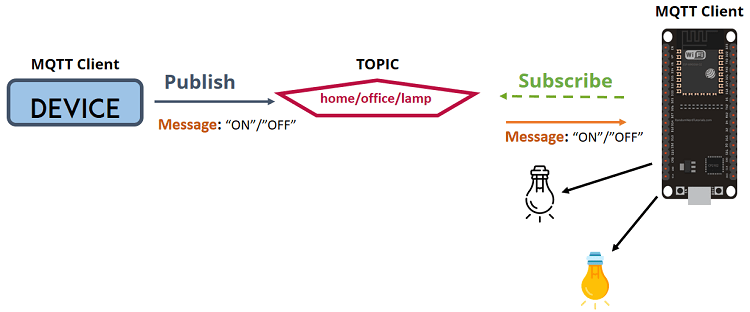


Figure 3.7 MQTT Topic

**3.2.5. MQTT – Broker**

The broker is primarily responsible for **receiving** all messages, **filtering** the messages, **decide** who is interested in them and then **publishing** the message to all subscribed clients. As shown in Figure 3.8:

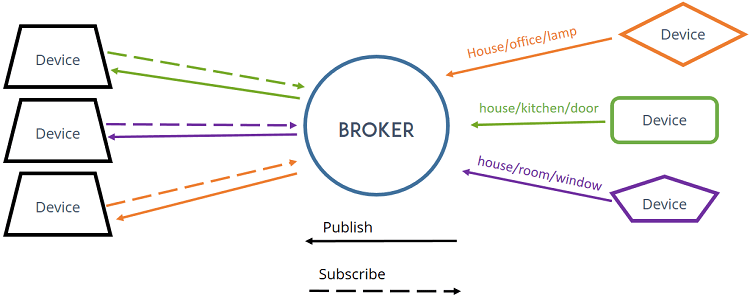


Figure 3.8 MQTT Broker

There are several brokers you can use. In our home automation projects we use the Mosquitto broker which can be installed in the Raspberry Pi. Alternatively, you can use a cloud MQTT broker.

**3.3. Implementation of HTTP**

Internet network is built to communicate via HTTP (Hyper Text Transfer Protocol). Various data, from images to texts, are sent over internet every day. HTTP is as a primary protocol interface to move a wide range of data quickly, easily, and stable from server to user devices such as browser. HTTP is built on TCP. HTTP ensures that data transmitted from one device to another will not corrupt so that the integrity of data transmitted is assured. HTTP is an open communication protocol that can be read by any devices that have been developed for HTTP protocol as browser or smartphone through browser application. An HTTP transaction consists of two parts: request command (request) sent from client to server, and response command (response) sent from server to client. The process of response and request is submitted using a data block with specific format known as HTTP Message. The messages are sent by HTTP which moves in one direction. Figure 3.9 is HTTP architecture.

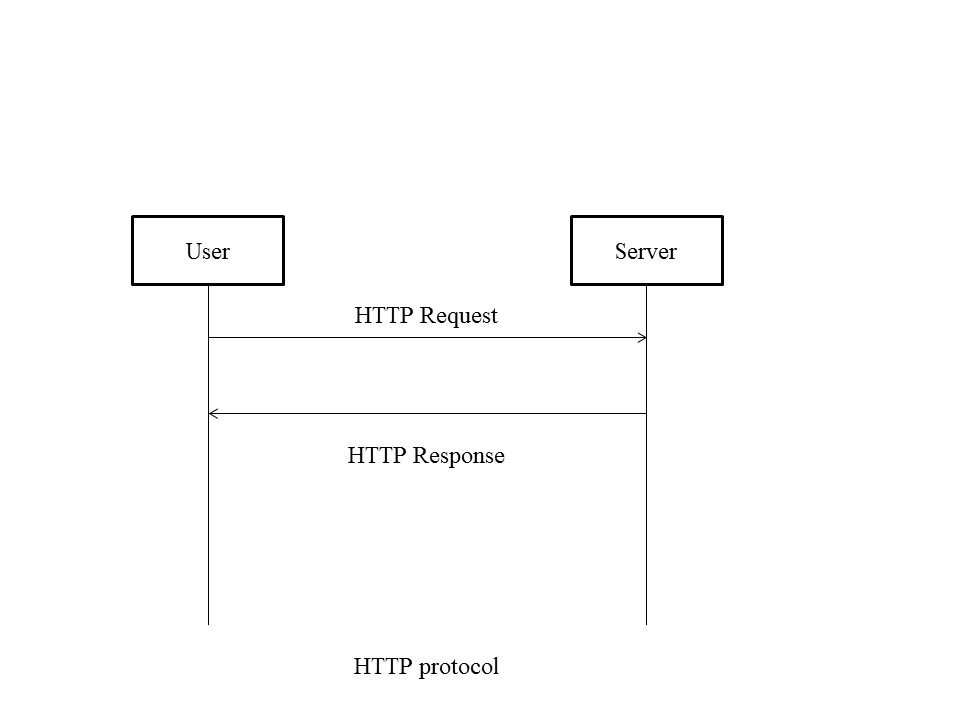


Figure 3.9 HTTP Architecture

**3.3. Comparison between MQTT and HTTP**

HTTP is the most popular and widely used protocol. But over the last years MQTT rapidly gain tractions. Developers have to choose between them when we are talking about IoT development.

**3.3.1. Design and Messaging**

MQTT is data centric whereas HTTP is document-centric. HTTP is request-response protocol for client-server computing and not always optimized for mobile devices. Main solid benefits of MQTT in these terms are lightweightness (MQTT transfers data as a byte array) and publish/subscribe model, which makes it perfect for resource-constrained devices and help to save battery.Besides, publish/subscribe model provides clients with independent existence from one another and enhance the reliability of the whole system. When one client is out of order the whole system can keep on working properly.

**3.3.2. Speed and Delivery**

According to measurements in 3G networks, throughput of MQTT is 93 times faster than HTTP’s.Besides, in comparison to HTTP, MQTT Protocol ensures high delivery guarantees. There are 3 levels of Quality of Services:

- at most once: guarantees a best effort delivery.

- at least once: guaranteed that a message will be delivered at least once. But the message can also be delivered more than once.

- exactly once: guarantees that each message is received only once by the counterpart

MQTT also provides users with options of Last will & Testament and Retained messages. The first means that in case of unexpected disconnection of a client all subscribed clients will get a message from a broker. Retained message means that a newly subscribed client will get an immediate status update.HTTP Protocol has none of these abilities.

**3.3.3. Complexity and Message Size**

MQTT has pretty short specification. There are only CONNECT, PUBLISH, SUBSCRIBE, UNSUBSCRIBE and DISCONNECT types that are significant for developers. Whereas HTTP specifications are much longer.MQTT has a very short message header and the smallest packet message size of 2 bytes. Using text message format by HTTP protocol allows it to compose lengthy headers and messages. It helps to eliminate troubles because it can be read by humans, but at the same time it’s needless for resource-constrained devices.

**3.4. Conclusion**

MQTT Protocol is easy of use. It is essential when response time, throughput, lower battery and bandwidth usage are on the first place for future solutions. It’s also perfect in case of intermittent connectivity.HTTP is worthy and extendable. But MQTT is more suitable when it is referred to IoT development.