

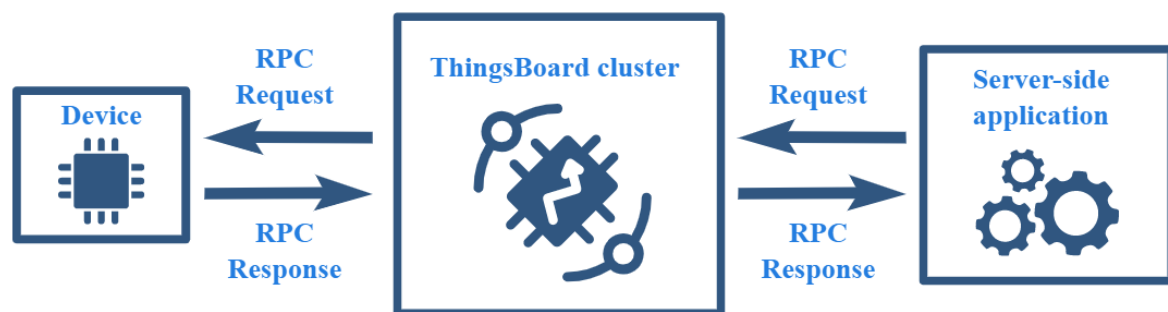
Control device with Thigsboard

Now we know how to get sensor data from the device to things board and how to display the device data in the thingsboard dashboard. In this lesson you will learn how to control a remote device through thingsboard.

In the first hands on session, you control the onboard LED using a code running on the microcontroller. In this lecture our goal is to control the same LED through a command initiated in the things board web interface.

So far, we have used http protocol to send data to the thingsboard. One limitation of HTTP/HTTPS is that HTTP servers only respond to requests from clients. It is unidirectional in communication. For this exercise we will need bi-directional communication channel. MQTT provides this capability. They both run over TCP connections, and are both client-server in architecture, but MQTT allows messages to pass in both directions between clients and servers.

As stated before, our goal is to control the device using a command that triggered through a dashboard. Following figure illustrates the communication process happening during the execution of the command



Next step is to get message from the server. In this workshop we use pubsubclient Arduino library to communicate with the server over MQTT

Please explore the library <https://pubsubclient.knolleary.net/api>

We can break down the connection process in to few main components

First, we need to connect to local WiFi network so that it can communicate through the network and reach the thingsboard server

Second, we need some sort of client , that understand how to communicate with the thingsboard server through the protocol thingsboard understand

There are many WiFi libraries and many client libraries available. However, challenge is not all libraries will be compatible with each other. In this session we will use default wifi client available through ESP8266WiFi library.

As we did with the last session, first we need to connect to the local network. Once we have established the WiFi network we can start setting up client to connect to the thingsboard.

We will use PubSubClient () constructor, PubSubClient () will create an uninitialised client instance.

Before we can use this client, we must configure it with the network client and the server information.

For example,

PubSubClient client(wifiClient) constructor will partially initialize the client with the wificlient (network client) configured in the program.

Second step is to add the server details to PubSubClient using PubSubClient* setServer (server, port) function.

In our case we will be communicating with the thingsboard instance

```
client.setServer( thingsboardServer, 1883 );
```

Then we need to connect to the virtual device in the things board. You can use different authentication mechanisms of this. In this example we will use tokens.

```
client.connect("ESP8266 Device", TOKEN, NULL)
```

if connection is successful, it will return true. We can check the connection status with the following command

```
client.connected()
```

Exercise

Write a script to connect to the thingsboard instance that you have for this workshop. Print out a message upon successful connection. If connection unsuccessful print out that information as well.