ESP-CLOUD INTERFACE

# Pre-requisites:

1. Arduino
2. Arduino packages
3. Basics of PHP and MySQL
4. Knowledge of HTTP and MQTT protocol

# Cisco UT VPN

The iot server is hosted in the UT Network. As of now, the server has no IP external to the UT Network. You can use the server only if you are a part of UT Network.

# Framework

Currently, we support communication via HTTP and MQTT protocols on our IOT Server. We assume that the reader of the document is familiar with MQTT concepts such as broker, pub-sub clients and JSON format.

## HTTP Server

Our goal is to establish communication between the ESP module and the IoT server via HTTP messages. For that, we installed an Apache webserver on our IoT server. Next, we connected this Apache webserver to a MySQL database. Finally, we coded a simple PHP script to redirect the incoming HTTP queries to the SQL database.

**Proof of Concept**

As a proof of concept, we decided to publish dummy values of temperature, pressure and humidity from the ESP module to the web server.

**Dependencies:**

Arduino libraries:

1. ESP8266WiFi.h
2. ESP8266WiFiMulti.h
3. ESP8266HTTPClient.h

* We begin by first programming the ESP to publish simple HTTP messages to our server.

**Note**: Our server DOES not support external IPs. Hence, the ESP must be a part of UT Network.

A simple Arduino file establishing the HTTP functionality can be found at

*$REPO/esp\_server\_iface/http\_to\_server/esp\_code/BasicHttpClient/BasicHttpClient.ino/*

* We have already set a MySQL database on our server with a table(password for the database=root). The SQL database is called iot1 and the table is called pht. It is a good idea to check that we are able to login into our IoT server and then also check the SQL Database.

If you fail at the next step, check if you are part of the UT Network.

>> ssh -X username@10.157.130.77

Next, we check the database and the contents of the tables.

>> sudo mysql -p iot1

You should see the prompt change to mysql. Next, we list the contents of the table.

mysql > SELECT \* FROM pht;

This will list down the contents of the table.

* Next, we would like to like a php script to parse the HTTP message and embed sql in it to load/store the contents from the database. We provide an example script extracts the pressure, temperature and humidity value from the HTTP message published by the ESP and stores it in the database. The script is located at

*$REPO/tree/master/esp\_server\_iface/http\_to\_server/cloud\_code/test.php*

Copy this script at /var/www/html/. To test if the script works, open a browser and enter this link-

[*http://10.157.130.77/test.php?p=4&h=1&t=2*](http://10.157.130.77/test.php?p=4&h=1&t=2)

Check your table(using the previous step) to check if the value appears correctly in your database.

* Now that we have successfully implemented and tested the HTTP-SQL interface functionality, we can power the ESP and check if it successfully publishes message.

MySQL supports real time visualization. I found [this](http://www.sqldashboards.com/download) to be particularly easy to use.

## <TODO>

## MQTT Server

MQTT has been used more and more in the IoT ecosystem. We decided to support MQTT capability as well.