**Internet of Things**

**Lab Report 8**

**Hafiz Ahmad**

**19l-1316**

**Section-7A2**

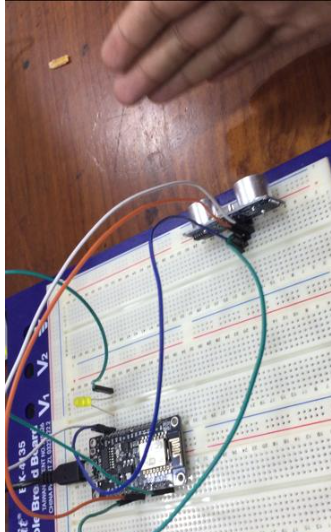
**Introduction to IoT Platforms**

**INTRODUCTION:**

An IoT platform is a collection of parts that let developers distribute applications, collect data remotely, secure connectivity, and manage sensors.The connectivity of the devices is managed by an IoT platform, which also lets developers create new mobile software applications.An on-premises software suite known as an IoT platform or a cloud service known as an IoT platform as a service (PaaS) monitors, manages, and controls a variety of endpoints, frequently through applications that business units install on the platform.Any Internet of Things product must include an IoT platform.It can help you get to product-market fit faster, reduce development costs, reduce risk, and speed up time to market. An Internet of Things platform is a collection of technologies that serve as the foundation for your product's development.The "infrastructure" you need to create the particular features of your solution is provided by IoT platforms.An IoT platform's objective is to provide your application with all generic functionality so you can concentrate on developing features that set your product apart and add value to customers.IoT platforms help you speed up the time to market for your product by taking over the non-differentiated functionality. When people talk about IoT platforms, they frequently begin with technical jargon like transport protocols, rules engines, data lakes, and other terms.Even though these considerations are significant and require careful planning, they do not provide a clear illustration of how an IoT platform can assist you.

**OBJECTIVES:**

To learn about IoT (Internet of Things) based automation and connect with Cloud through wifi.



**Lab code:**

/\*

    This sketch demonstrates how to set up a simple HTTP-like server.

    The server will set a GPIO pin depending on the request

<http://server_ip/gpio/0> will set the GPIO2 low,

<http://server_ip/gpio/1> will set the GPIO2 high

    server\_ip is the IP address of the ESP8266 module, will be

    printed to Serial when the module is connected.

\*/

#include <ESP8266WiFi.h>

#ifndef STASSID

#define STASSID "EE"

#define STAPSK  "ee123456"

#endif

const char\* ssid = STASSID;

const char\* password = STAPSK;

// Create an instance of the server

// specify the port to listen on as an argument

WiFiServer server(80);

void setup() {

  Serial.begin(115200);

  // prepare LED

  pinMode(LED\_BUILTIN, OUTPUT);

  digitalWrite(LED\_BUILTIN, 0);

  // Connect to WiFi network

  Serial.println();

  Serial.println();

  Serial.print(F("Connecting to "));

  Serial.println(ssid);

  WiFi.mode(WIFI\_STA);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(F("."));

  }

  Serial.println();

  Serial.println(F("WiFi connected"));

  // Start the server

  server.begin();

  Serial.println(F("Server started"));

  // Print the IP address

  Serial.println(WiFi.localIP());

}

void loop() {

  // Check if a client has connected

  WiFiClient client = server.available();

  if (!client) {

    return;

  }

  Serial.println(F("new client"));

  client.setTimeout(5000); // default is 1000

  // Read the first line of the request

  String req = client.readStringUntil('\r');

  Serial.println(F("request: "));

  Serial.println(req);

  // Match the request

  int val;

  if (req.indexOf(F("/gpio/0")) != -1) {

    val = 0;

  } else if (req.indexOf(F("/gpio/1")) != -1) {

    val = 1;

  } else {

    Serial.println(F("invalid request"));

    val = digitalRead(LED\_BUILTIN);

  }

int toggle, b=0;

toggle = digitalRead(LED\_BUILTIN);

  // Set LED according to the request

 if (val==0)

 {

  toggle = 1;

  digitalWrite(LED\_BUILTIN, toggle);

  delay(2000);

 }

 else if(val==1)

 {

  toggle=0;

  digitalWrite(LED\_BUILTIN, toggle);

   delay(2000);

  }

  else

  {

    }

  // read/ignore the rest of the request

  // do not client.flush(): it is for output only, see below

  while (client.available()) {

    // byte by byte is not very efficient

    client.read();

  }

  // Send the response to the client

  // it is OK for multiple small client.print/write,

  // because nagle algorithm will group them into one single packet

  client.print(F("HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<html>\r\nGPIO is now "));

  client.print((toggle) ? F("high") : F("low"));

  client.print(F("<br><br>Click <a href='http://"));

  client.print(WiFi.localIP());

  client.print(F("/gpio/1'>here</a> to switch LED GPIO on, or <a href='http://"));

  client.print(WiFi.localIP());

  client.print(F("/gpio/0'>here</a> to switch LED GPIO off.</html>"));

  // The client will actually be \*flushed\* then disconnected

  // when the function returns and 'client' object is destroyed (out-of-scope)

  // flush = ensure written data are received by the other side

  Serial.println(F("Disconnecting from client"));

}

**Application:**

The ThingSpeakTM is a cloud-based Internet of Things analytics platform service that lets you aggregate, visualize, and analyze live data streams.Data uploaded by your devices to ThingSpeak can be instantly visualized with ThingSpeak.Real-time data collection, data processing, visualizations, apps, and plugins are all part of ThingSpeak.A ThingSpeak Channel is the core of ThingSpeak.You send your data to be stored through a channel.There are three location fields, one status field, and eight fields for any kind of data in each channel.Channels in ThingSpeak store data.Send data from devices to a ThingSpeak channel or upload data from the web.Transform and visualize data using these apps, or initiate an action.See the Tutorial:MATLAB and ThingSpeak to create a channel.Within ThingSpeak, you can learn more about MATLAB®.The IoT analytics platform service ThingSpeak enables live data streams to be aggregated, visualized, and analyzed in the cloud.Using web services like Twitter® and Twilio®, you can send alerts, instantly visualize live data, and send data to ThingSpeakTM from your devices.

**Issues:**

we never find any issue regarding this lab.

**Conclusion:**

In this lab we perform IoT (Internet of Things) based door and fan automation on cloud how to upload data on cloud and cloud based decisions. An IoT product should:

* Use sensors to gather data from the real world
* Local data analysis (edge computing)
* Join the cloud to send and receive commands and data.
* Keep information on the cloud
* Perform cloud data analysis to produce insights
* Using knowledge, direct the "objects" to carry out certain duties.