What Is CoAP?

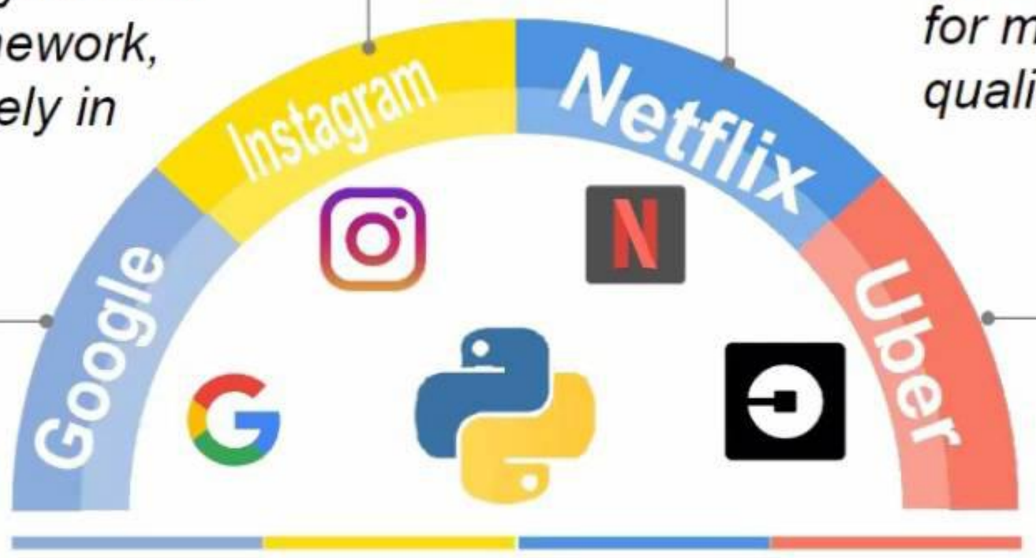
Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. CoAP is designed to enable simple, constrained devices to join the IoT even through constrained networks with low bandwidth and low availability. It is generally used for machine-to-machine (M2M) applications such as smart energy and building automation. The protocol was designed by the Internet Engineering Task Force (IETF), CoAP is specified in IETF RFC 7252.

Scanned by TapScanner

Why is python famous?

Instagram currently features the world's largest deployment of the Django web framework, which is written entirely in Python..

Python is used in tools for monitoring data quality, managing data



Google got started using Python in their tech Stack

Uber is using ML to enable an efficient ride-sharing marketplace

Harish's screen

Scanned by TapScanner

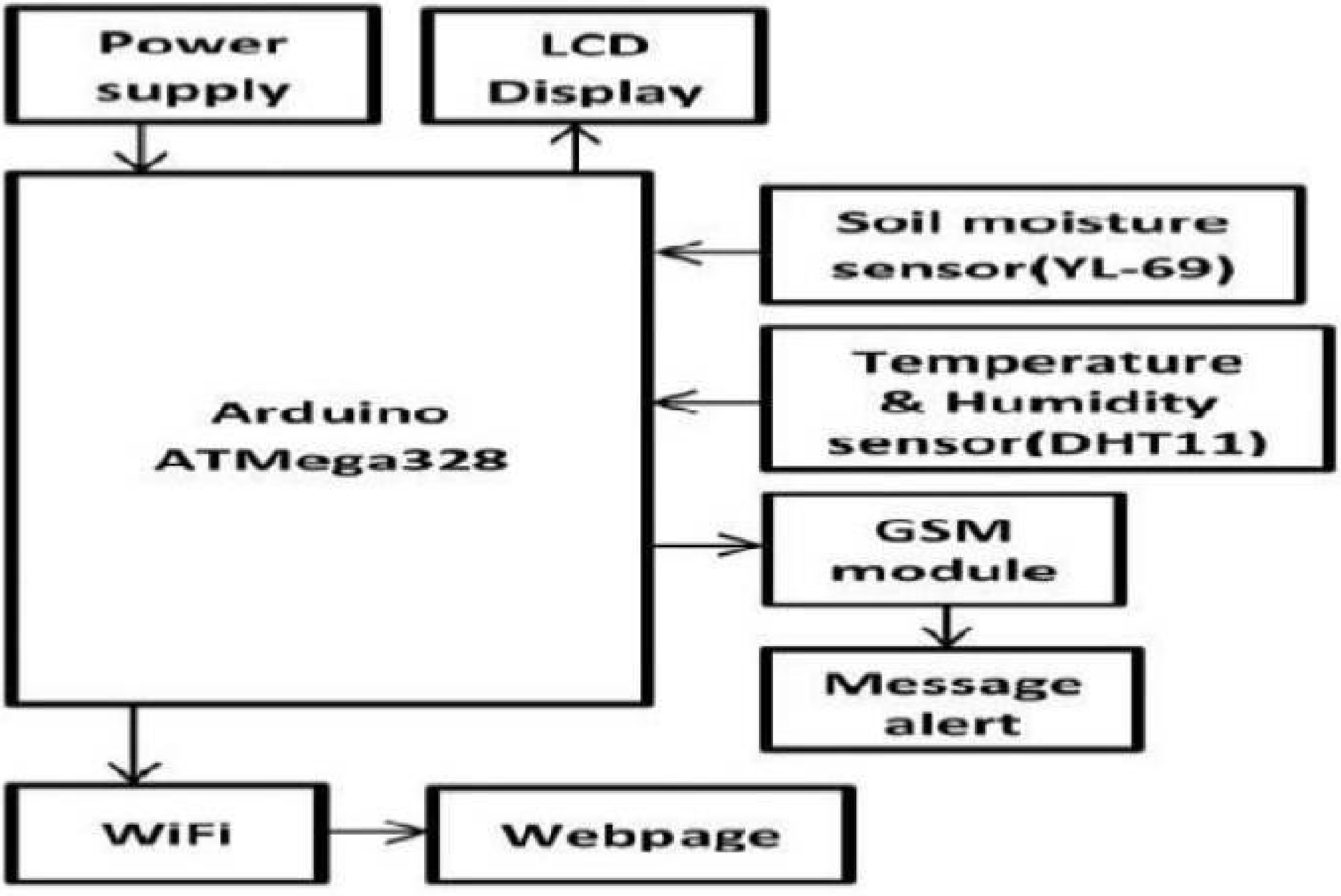


Fig 4.1 Block diagram for crop monitoring

Scanned by TapScanner

Usecase-3:

IOT based Smart Crop Protection System For Agriculture

Description:

- Crops in the farms are many times devastated by the wild as well as domestic animals and low productivity of crops is one of the reasons for this. It is not possible to stay 24 hours in the farm to guard the crops.
- An intelligent crop protection system helps the farmers in protecting the crop from the animals and birds which destroy the crop. This system shall also include remote monitoring and control of pump to avoid the farmer to visit the farm in nighttime.
- Solution Requirements:
  - Safety of people & animal
  - Low-cost solutions, lower dependency on power
  - Simple solution to suite the farmer community

Scanned by TapScanner

Usecase-3:

IOT based Smart Crop Protection System For Agriculture

Social Impact

Improve the productivity, Save lives of farmers

Business Model/Impact

Community based solution by FAO's
Solution through contract farming

Recommended Technology Stack

Computer Vision API's, GSM Modules, ESP8266, IBM Watson IoT Platform, Android Application

Existing Solutions

https://www.agrivi.com/blog/top-five-strategies-to-protect-crops-from-wild-animals/

References

https://article.murata.com/en-eu/article/measures-against-wildlife-damage-through-iot
https://www.fao.org/3/cb2447en/cb2447en.pdf

Scanned by TapScanner

After installing the libraries to Arduino IDE, start the code by including the required libraries files.

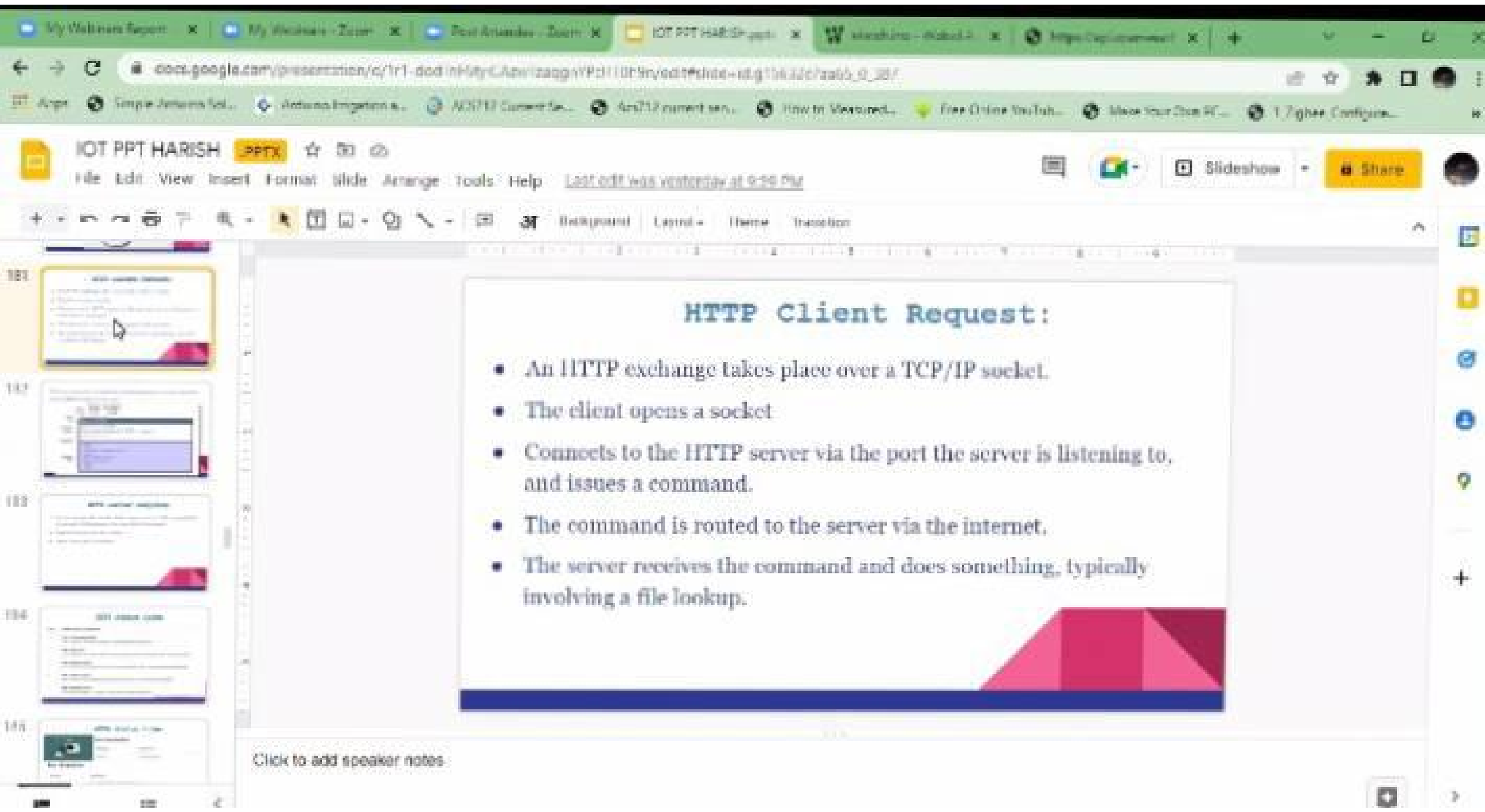
```
#include <ESP8266WiFi.h>
#include <DallasTemperature.h>
#include <OneWire.h>
#include "DHT.h"
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <ArduinoJson.h>
```

Then enter the Wi-Fi and Adafruit IO credentials that you copied from the Adafruit IO server. These will include the MQTT server, Port No, User Name, and AIO Key.

```
const char *ssid =  "Wi-Fi Name";
const char *pass =  "Wi-Fi password"
#define MQTT_SERV "io.adafruit.com"
#define MQTT_PORT 1883
#define MQTT_NAME "Adafruit IO Usern
#define MQTT_PASS "AIO Key"
```

Then set up the Adafruit IO feeds for storing the sensor data and controlling LED and water pump. In my case, I have defined four feeds to store different

Scanned by TapScanner



Scanned by TapScanner