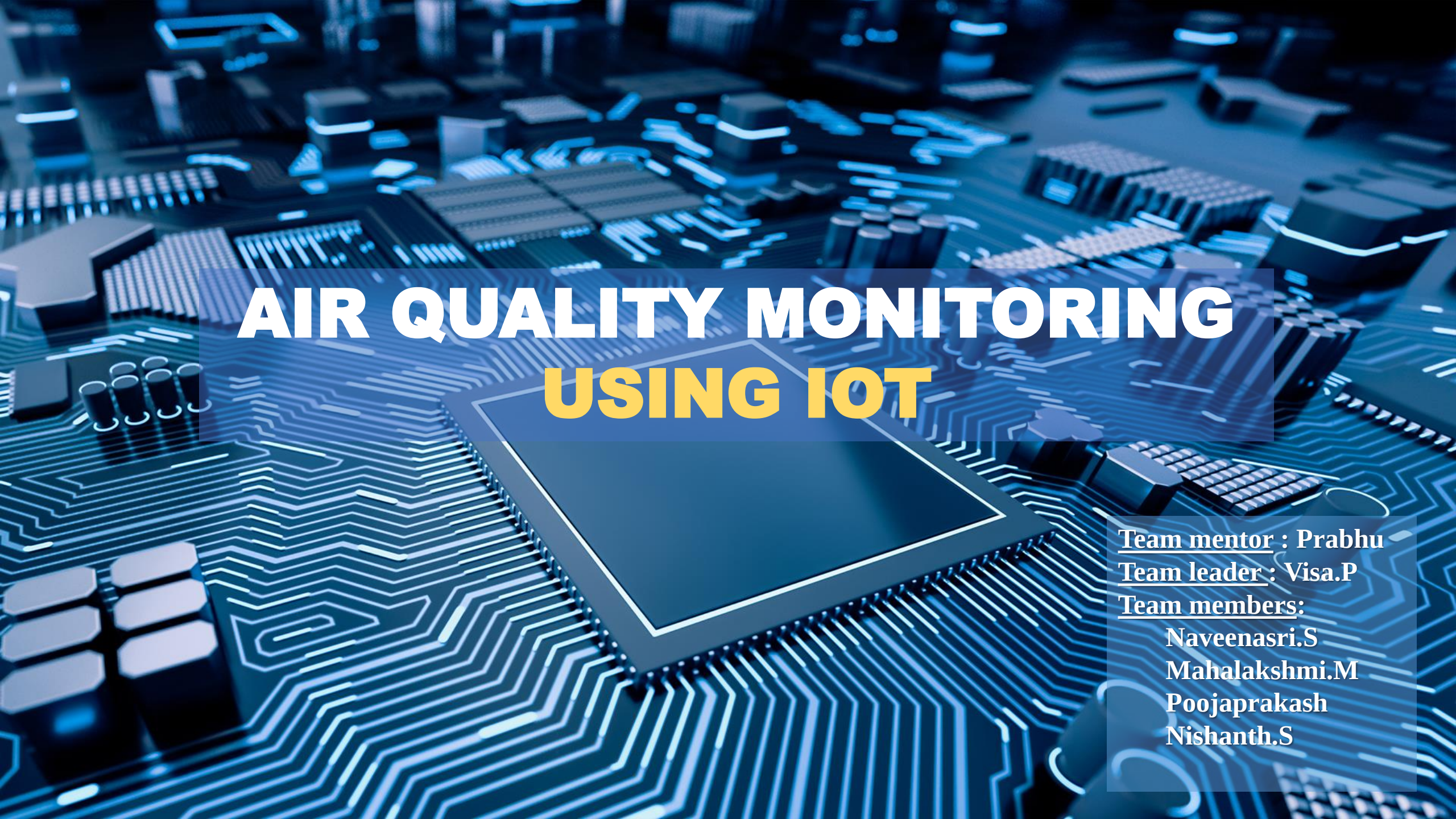


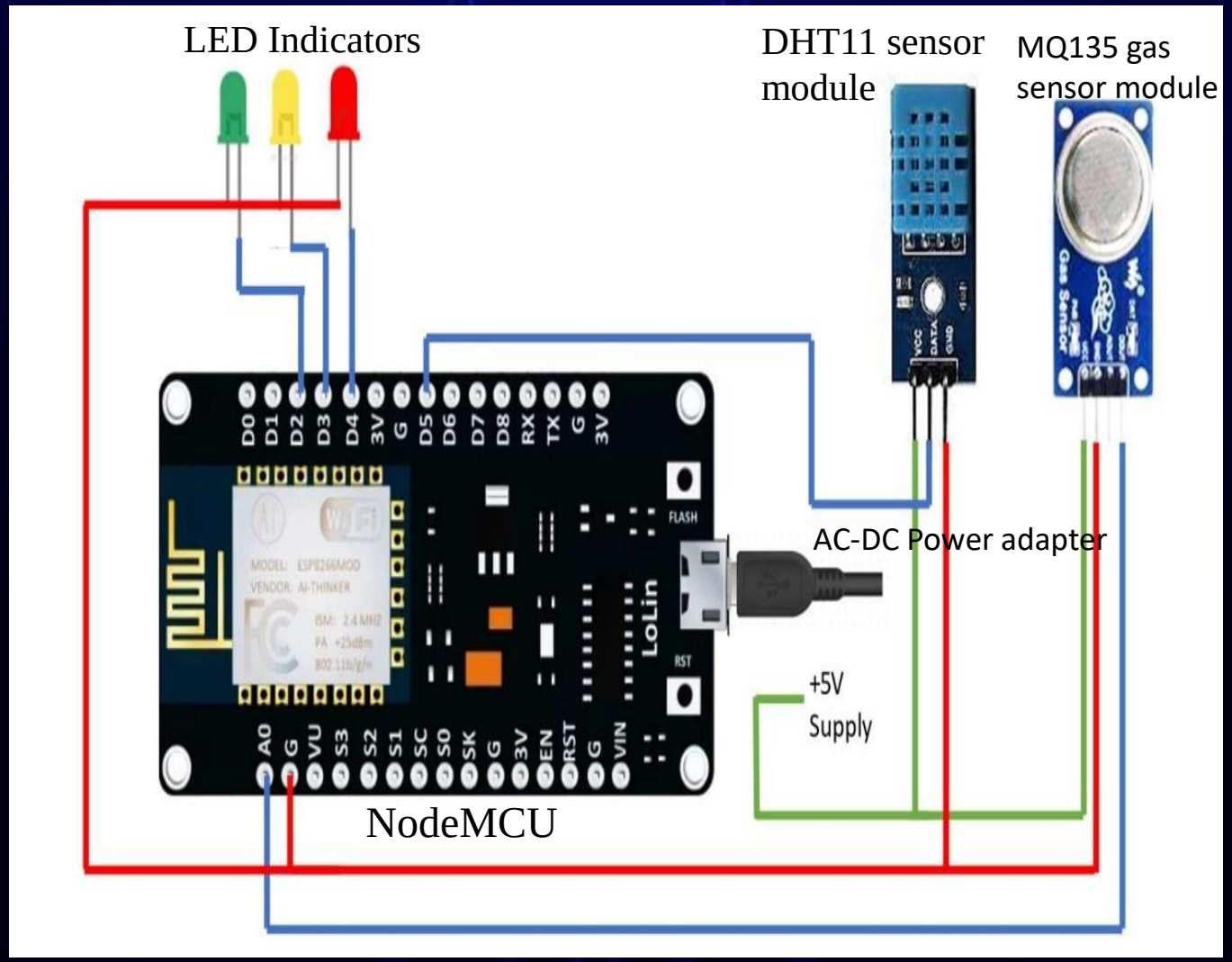
AIR QUALITY MONITORING
USING IOT

Team mentor : Prabhu
Team leader : Visa.P
Team members:
Naveenasri.S
Mahalakshmi.M
Poojaprakash
Nishanth.S

ABSTRACT

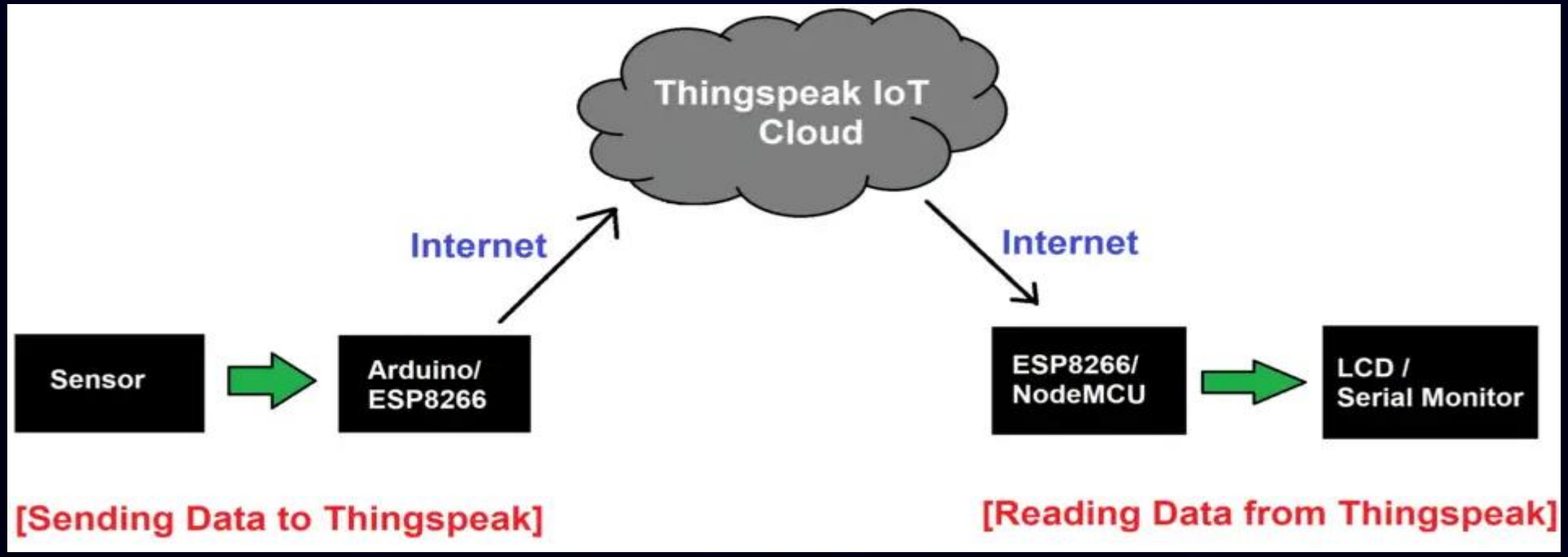
- Air pollution is one of the biggest threats to the present-day environment.
- Everyone is being affected by air pollution day by day including humans, animals, crops, cities, forests and aquatic ecosystems.
- Besides that, it should be controlled at a certain level to prevent the increasing rate of global warming.
- This project aims to design an IOT-based air pollution monitoring system using the internet from anywhere using a computer or mobile to monitor the air quality of the surroundings and environment.
- In this system, NodeMCU plays the main controlling role. It has been programmed in a manner, such that, it senses the sensory signals from the sensors and shows the quality level via led indicators.
- Sensor responses are fed to the NodeMCU which displays the monitored data in the ThingSpeak cloud which can be utilized for analyzing the air quality of that area.

SET-UP OF THIS EXPERIMENT



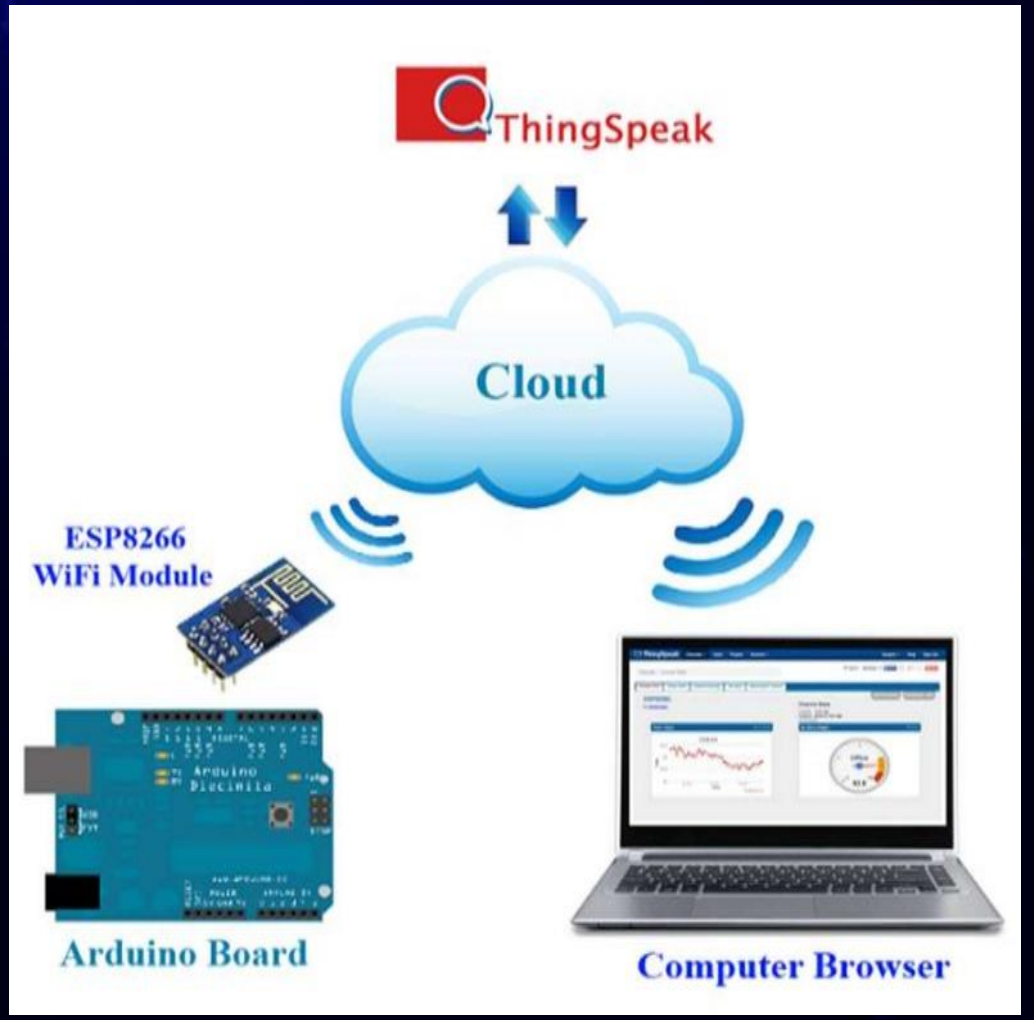
SOFTWARE COMPONENTS

1. ThinkSpeak Cloud
2. Arduino IDE



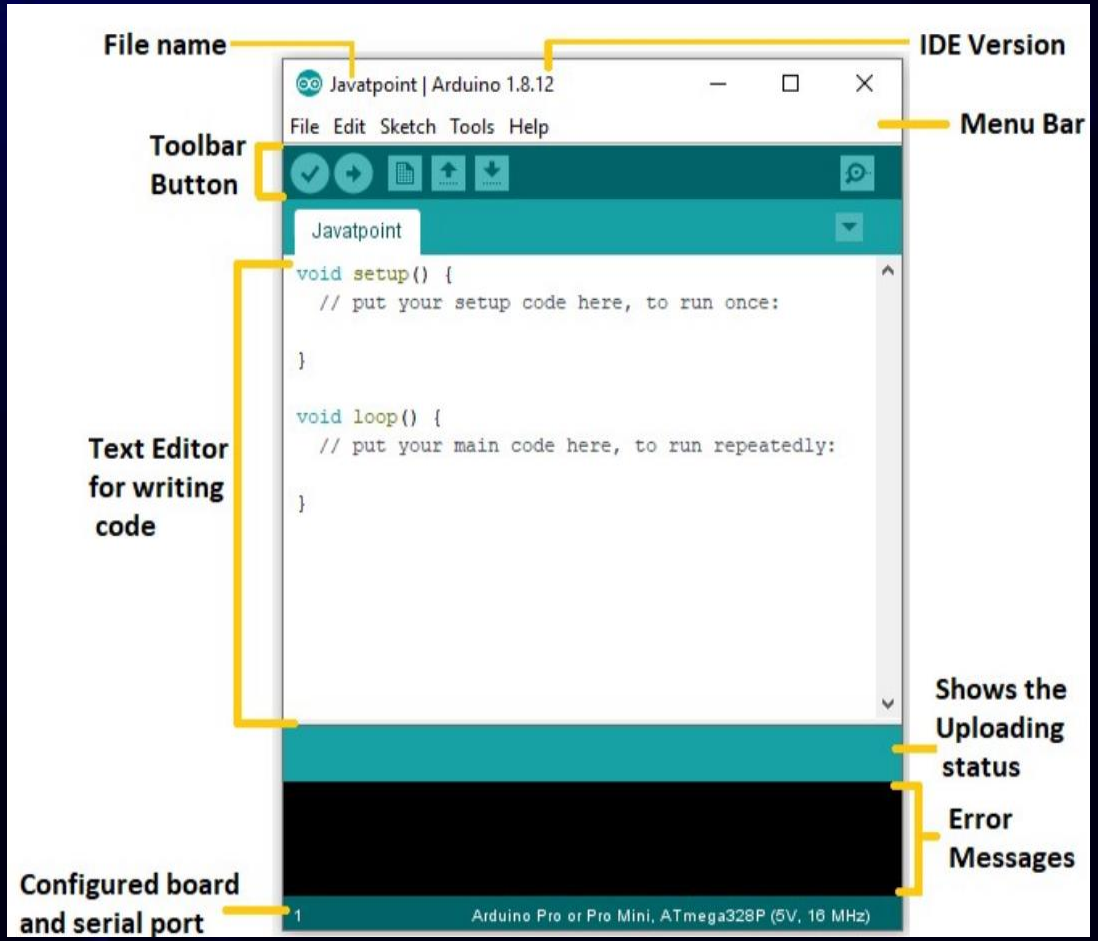
THINKSPEAK CLOUD

- ThingSpeak is open-source software written in Ruby which allows users to communicate with internet-enabled devices.
- It facilitates data access, retrieval, and logging of data by providing an API to both the devices and social network websites.
- ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications.
- ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks, allowing ThingSpeak users to analyze and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from MathWorks.



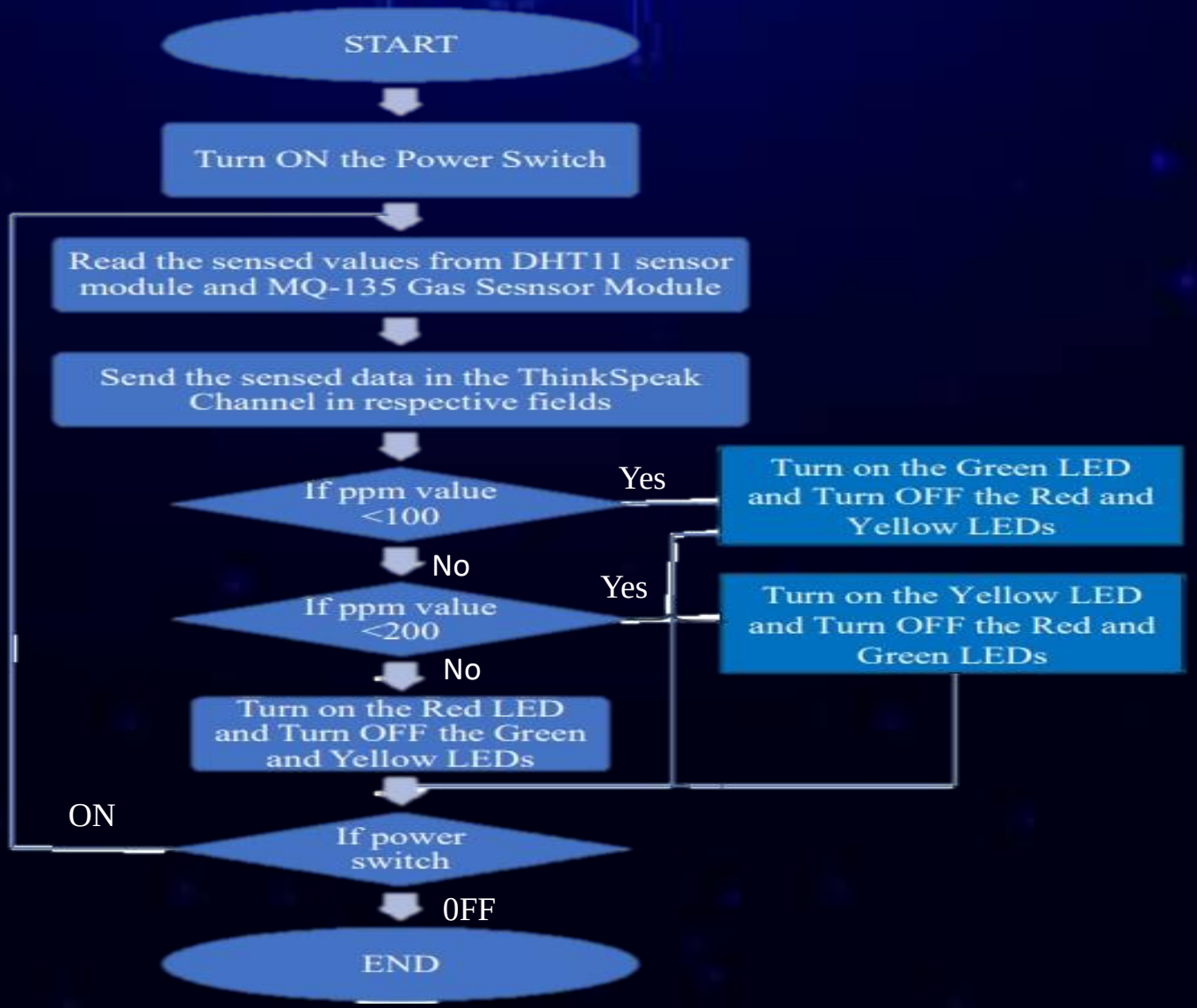
Arduino IDE

- The Arduino IDE is open-source software, which is used to write and upload code to the Arduino boards.
- The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux.
- It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment.
- The program or code written in the Arduino IDE is often called sketching.
- We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'



SOFTWARE IMPLEMENTATION

WORKING ALGORITHM:



SOFTWARE CODE for Calibration of MQ135 Sensor

```
#using python script
 import time

 def read_analog(pin):
     # Simulate analogRead function for the purpose of this example
     # Replace this with your actual method to read analog values
     return 512  # Example value, replace with your analog reading logic

 def setup():
     print("Setting up...")
     # You can set up any necessary components or configurations here
     print("Setup complete")

 def loop():
     while True:
         sensorValue = 0.0
         print("Sensor Reading = " + str(read_analog(0)))
```

```
for x in range(500):
    sensorValue += read_analog(0)

sensorValue /= 500.0
sensor_volt = sensorValue * (5.0 / 1023.0)
RS_air = (5.0 / sensor_volt) - 1.0
R0 = RS_air / 3.7

print("R0 = " + str(R0))
time.sleep(1)

if __name__ == "__main__":
        setup()
        try:
loop()
except KeyboardInterrupt:
print("\nScript terminated by user")
```

EXECUTION OF THE MAIN PROGRAM

```
#using python script
  import time
  import requests
  import Adafruit_DHT
  import math
  import RPi.GPIO as GPIO

  # Define your sensor and pin settings
  DHT_SENSOR = Adafruit_DHT.DHT11
  DHT_PIN = 5
  MQ_135_PIN = 0  # Analog pin for MQ-135

  # Linear regression parameters
  m = -0.3376
  b = 0.7165
  R0 = 3.12  # Sensor resistance in fresh air
```

```
# ThingSpeak settings
myChannelNumber = 123456  # Channel ID
myWriteAPIKey = "API_Key"

# LED pin assignments
LED_GREEN = 2
LED_YELLOW = 3
LED_RED = 4

# Set up LED pins
LED_PINS = [LED_GREEN, LED_YELLOW, LED_RED]

for pin in LED_PINS:
    GPIO.setup(pin, GPIO.OUT)

# Connect to Wi-Fi
WiFi_Name = "Your_SSID"
WiFi_Password = "Your_Password"

# Create a session for ThingSpeak
session = requests.Session()
session.verify = False  # Disable SSL certificate verification
```

```
while True:
    # Read DHT sensor values
    h, t = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)

    sensorValue = analogRead(MQ_135_PIN)
    sensor_volt = sensorValue * (5.0 / 1023.0)
    RS_gas = (5.0 / sensor_volt) - 1.0
    ratio = RS_gas / R0

    ppm_log = (math.log10(ratio) - b) / m
    ppm = math.pow(10, ppm_log)

    print("Temperature: " + str(t))
    print("Humidity: " + str(h))
    print("Our desired PPM = " + str(ppm)

    # Upload data to ThingSpeak

session.post(f'https://api.thingspeak.com/update?api_key={myWriteAPIKey}&field1={t}&field2={h}&field3={ppm}')
    time.sleep(20)
```

```
# Control LEDs based on ppm value
    if ppm <= 100:
        GPIO.output(LED_GREEN, GPIO.HIGH)
        GPIO.output(LED_YELLOW, GPIO.LOW)
        GPIO.output(LED_RED, GPIO.LOW)
    elif ppm <= 200:
        GPIO.output(LED_GREEN, GPIO.LOW)
        GPIO.output(LED_YELLOW, GPIO.HIGH)
        GPIO.output(LED_RED, GPIO.LOW)
    else:
        GPIO.output(LED_GREEN, GPIO.LOW)
        GPIO.output(LED_YELLOW, GPIO.LOW)
        GPIO.output(LED_RED, GPIO.HIGH)

    time.sleep(2)
```

CONCLUSION

In this project, IoT based on the measurement and display of Air Quality Index (AQI), Humidity, and Temperature of the atmosphere have been performed. From the information obtained from the project, it is possible to calculate Air Quality in PPM. The disadvantage of the MQ135 sensor is that specifically, it can’t tell the Carbon Monoxide or Carbon Dioxide level in the atmosphere, but the advantage of MQ135 is that it is able to detect smoke, CO, CO2, NH4, etc harmful gases.

THANK YOU!!!