GREEN HOUSE MONITORING AND CONTROLLING BY IBM WATSON

Team name: Innovators

Teams members: 1: D. Madhuri

2: G. Kavya

3: P. Venkat sai

ABSTRACT:

In the 21st century, one of the most significant technologies is the Internet of Things (IoT) which has rapidly developed covering hundreds of applications in the civil, health, military and agriculture areas. In modern greenhooperationuses, several measurement points are required to trace down the local climate parameters in different parts of a large scale greenhouse in order to ensure proper of the greenhouse automation system. This can be done using prototype consisting of DHT11, mq-135 and mq-2 which are used to measure greenhouses’ temperature, air quality, detection of harmful gases and humidity. Measurement data have been shared with the help of IoT. With this system farmer can control their greenhouse from their mobile phones or computers which have internet connection.

HARDWARE COMPONENTS:

1: DHT-11

DHT-11 sensor is a sensor which is used to calculate and measure the temperature and humidity values. This module can be applied to environmental temperature and environmental humidity.

This sensor includes a resistive-type humidity measurement

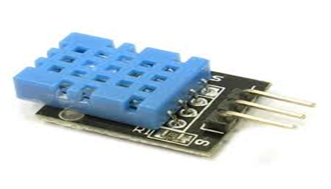
component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

Pin No. Symbol Descriptions

1. DOUT Communication port

2. GND Power ground

3. VCC Positive power supply (3.3V-5.5V)



2: NODEMCU ESP8266:

NodeMCU is an open source Lua based firmware for the ESP8266 Wi-Fi SOC from Espressif and uses an on-module flash-based SPIFFS file system. NodeMCU is implemented in C and is layered on the Espressif NON-OS SDK.

The firmware was initially developed as is a companion project to the popular ESP8266-based NodeMCU development modules, but the project is now community-supported, and the firmware can now be run on any ESP module

I/O index ESP8266 pin

0 [\*] GPIO16

1 GPIO5

2 GPIO4

3 GPIO0

4 GPIO2

5 GPIO14

6 GPIO12

7 GPIO13

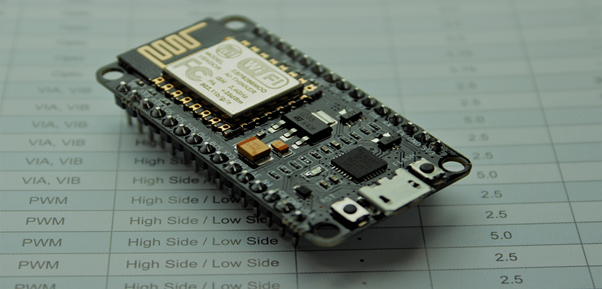
8 GPIO15

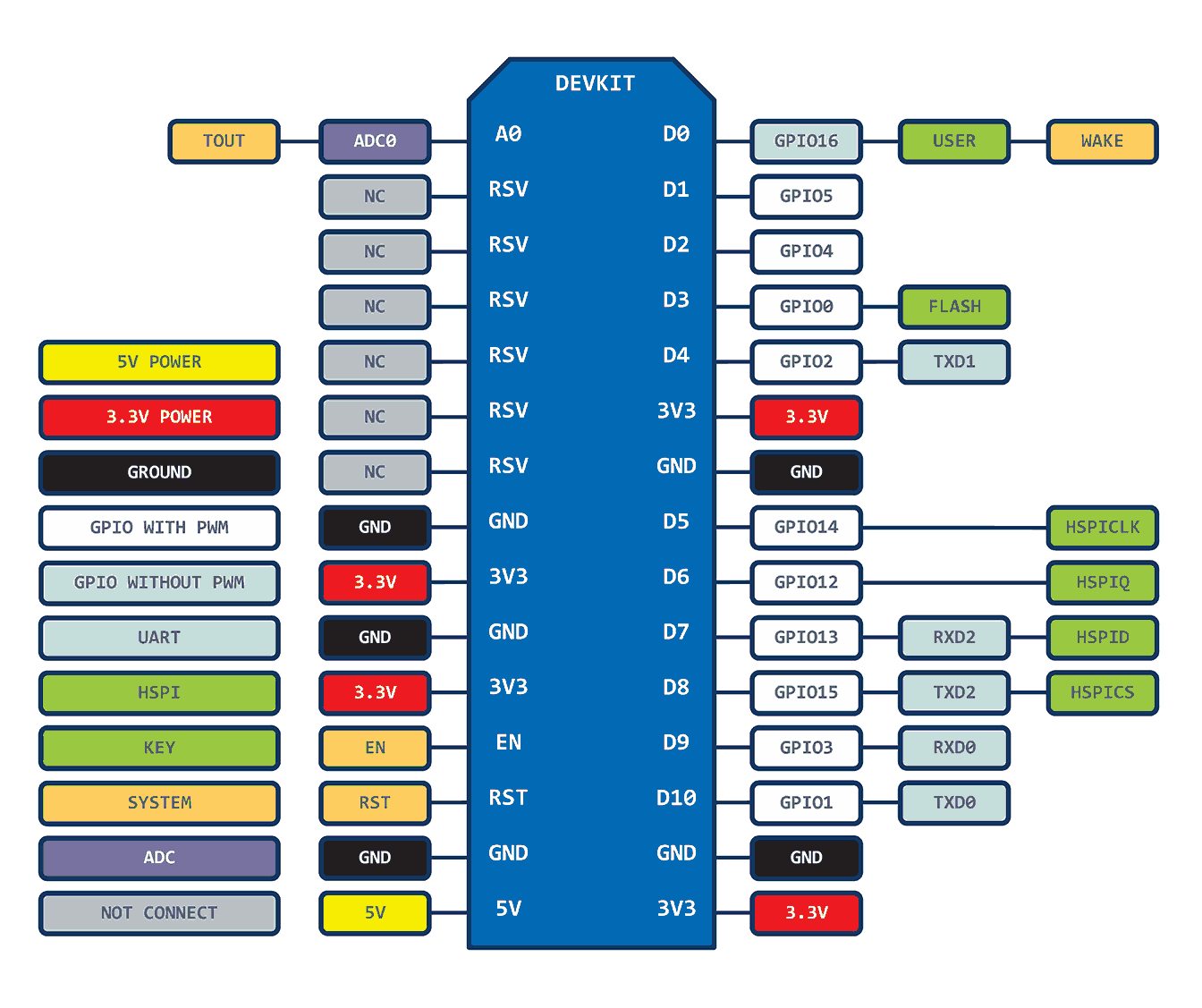
9 GPIO3

10 GPIO1

11 GPIO9

12 GPIO10





NODEMCU

3:MQ-135:

Air quality sensor for detecting a wide range of gases, including NH3, NOx, alcohol, benzene, smoke and CO2. Ideal for use in office or factory. MQ135 gas sensor has high sensitivity to Ammonia, Sulfide and Benze steam, also sensitive to smoke and other harmful gases. It is with low cost and particularly suitable for Air quality monitoring application.

Features:

* High Sensitivity.
* High sensitivity to Ammonia, Sulfide and Benzene.
* Stable and Long Life.
* Detection Range: 10 - 300 ppm NH3, 10 - 1000 ppm Benzene, 10 - 300 Alcohol.
* Heater Voltage: 5.0V.
* Dimensions: 18mm Diameter, 17mm High excluding pins, Pins - 6mm High
* Long life and low cost



Applications:

* Domestic air pollution detector
* Industrial air pollution detector
* Portable air pollution detector

4:MQ-2:

The Grove - Gas Sensor(MQ2) module is useful for gas leakage detection (home and industry). It is suitable for detecting H2, LPG, CH4, CO, Alcohol, Smoke or Propane. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible. The sensitivity of the sensor can be adjusted by potentiometer.

Features:

* Wide detecting scope
* Stable and long lifetime
* Fast response and High sensitivity

Application Ideas:

* Gas leakage detection.
* Toys.



Pin No Pin name

1. VCC (3.3V – 5 V)
2. GND
3. Digital out
4. Analog out

Why Make Use of Greenhouse Automation Systems?

That is a question that I am sure goes through the mind of every greenhouse grower at some time or another. Is it really worth the cost? Will my employees embrace it and employ everything that an automated control system has to offer?

Automation takes on a lot of different forms. Here we will be talking about controlling the internal environment and automating the irrigation system with fertilizer (fertigation). It is taking the tasks that we ask our employees to do every day and making those tasks simpler and efficient by automated them. Not to mention the cost savings on labour, energy, water and fertilizer.

Overall, greenhouses can be irrigated automatically so growers can water crops with minimal staff while maintaining time to inspect and manage the crop growth. Greenhouse automation is a proven standard in Europe, especially flower growers in Holland. As greenhouses continue to become more popular in North America (especially with the newer trend towards medical marjuana), automation will grow toward becoming the new standard (amongst all growers) as legislation regarding water use and runoff becomes stricter.

There are Several Potential Benefits to Installing a greenhouse monitoring System:

1.Improved labour efficiency  
2. Reduced energy consumption  
3. Improved reliability  
4. Improved crop quality and yield  
5. Opportunity to expand and add more equipment and sensors  
6. Opportunity to modify with dimmable lights or variable speed fans, which can save energy while optimizing crop requirements



Some more advantages of greenhouse monitoring are:

* Increase in fertility
* Better productivity
* Dependence on other people is reduced
* Percentage of germination of seeds is high in green house

Software components:

1.ARDUINO:

**Arduino** is open source hardware and [software](https://en.wikipedia.org/wiki/Open-source_software) company, project and user community that designs and manufactures [single-board microcontrollers](https://en.wikipedia.org/wiki/Single-board_microcontroller) and [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) kits for building digital devices and interactive objects that can sense and control both physically and digitally. Its products are licensed under the [GNU Lesser General Public License](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License) (LGPL) or the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License)(GPL),[[1]](https://en.wikipedia.org/wiki/Arduino#cite_note-1) permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as [do-it-yourself](https://en.wikipedia.org/wiki/Do-it-yourself) (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various expansion boards or [breadboards](https://en.wikipedia.org/wiki/Breadboards) (*shields*) and other circuits. The boards feature serial communications interfaces, including [Universal Serial Bus](https://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). In addition to using traditional [compiler](https://en.wikipedia.org/wiki/Compiler) [toolchains](https://en.wikipedia.org/wiki/Toolchains), the Arduino project provides an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) based on the [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) language project.

2.IBM Watson cloud platform:

**Watson** on the **IBM Cloud** allows you to integrate the world's most powerful AI into your application and store, train and manage your data in the most secure **cloud**

**Arduino code:**

**#include <ESP8266WiFi.h>**

**#include <PubSubClient.h>**

**int mq135value;**

**int mq2value;**

**//-------- Customise these values -----------**

**const char\* ssid = "MLRITM@DGL";**

**const char\* password = "digi@123";**

**//////////////////////////**

**#include "DHT.h"**

**#define DHTPIN D2 // what pin we're connected to**

**#define DHTTYPE DHT11 // define type of sensor DHT 11**

**DHT dht (DHTPIN, DHTTYPE);**

**///////////////////////**

**#define ORG "xm610m"**

**#define DEVICE\_TYPE "nodemcu0404"**

**#define DEVICE\_ID "0071"**

**#define TOKEN "9381228129"**

**//-------- Customise the above values --------**

**char server[] = ORG ".messaging.internetofthings.ibmcloud.com";**

**char topic[] = "iot-2/evt/madhuri/fmt/Json";**

**char authMethod[] = "use-token-auth";**

**char token[] = TOKEN;**

**char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;**

**WiFiClient wifiClient;**

**PubSubClient client(server, 1883,wifiClient);**

**void setup() {**

**Serial.begin(115200);**

**Serial.println();**

**dht.begin();**

**pinMode(D0, OUTPUT);**

**Serial.print("Connecting to ");**

**Serial.print(ssid);**

**WiFi.begin(ssid, password);**

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

**delay(500);**

**Serial.print(".");**

**}**

**Serial.println("");**

**Serial.print("WiFi connected, IP address: ");**

**Serial.println(WiFi.localIP());**

**}**

**void loop() {**

**mq2value = analogRead(A0); // o-1**

**mq135value = digitalRead(D0); // 0 or 1024**

**Serial.println("mq2value" + String(mq2value));**

**Serial.println("mq135value" + String(mq135value));**

**float h = dht.readHumidity();**

**float t = dht.readTemperature();**

**if (isnan(h) || isnan(t))**

**{**

**Serial.println("Failed to read from DHT sensor!");**

**delay(1000);**

**return;**

**}**

**PublishData(t,h,mq2value,mq135value);**

**delay(1000);**

**}**

**void PublishData(float temp, float humid,int mq2,int mq135){**

**if (!!!client.connected()) {**

**Serial.print("Reconnecting client to ");**

**Serial.println(server);**

**while (!!!client.connect(clientId, authMethod, token)) {**

**Serial.print(".");**

**delay(500);**

**}**

**Serial.println();**

**}**

**String payload = "{\"d\":{\"temperature\":";**

**payload += temp;**

**payload+="," "\"humidity\":";**

**payload += humid;**

**payload+="," "\"mq2\":";**

**payload += mq2;**

**payload+="," "\"mq135\":";**

**payload += mq135;**

**payload += "}}";**

**Serial.print("Sending payload: ");**

**Serial.println(payload);**

**if (client.publish(topic, (char\*) payload.c\_str())) {**

**Serial.println("Publish ok");**

**} else {**

**Serial.println("Publish failed");**

**}**

**}**

**CONCLUSION:**

**Our system enables people to monitor and manage growing conditions of their greenhouse. The use sensor nodes, internet connection, and the cloud will deliver real-time updates about plants and help people grow plants more efficiently, with all observation and conventional tests results conclude that our project will provide a solution for automating greenhouse activities and irrigation activities. Implementation of such a system in the field can definitely help to improve the yield of the crops and overall production, and with its quality to cost ratio, it will be affordable to the majority of the agricultural community and also to agro-based industries.**