Wifi Enabled Weather Data Display

Github : <https://github.com/HarithKK/WIFI_LED_METRIX>

Video : https://youtu.be/Ci6tZcui5nY

This Project is basically target to display weather data collected by weather stations to public people. Most probably wifi is being facilitated to urban areas of where people are usually being. So that wifi is the best communication media to update information. Here 16x32 P10 LED matrix is used as the display equipment and backend microcontroller is used to control the patterns and panel. This display is programmed to visualize Temperature, Humidity, Wind speed, wind Direction, etc.

Basically this was designed to have 3 horizontal 16x32 P 10 LED displays. But it has included the extended facility. Figure 1 shows the internal block diagram of the system.

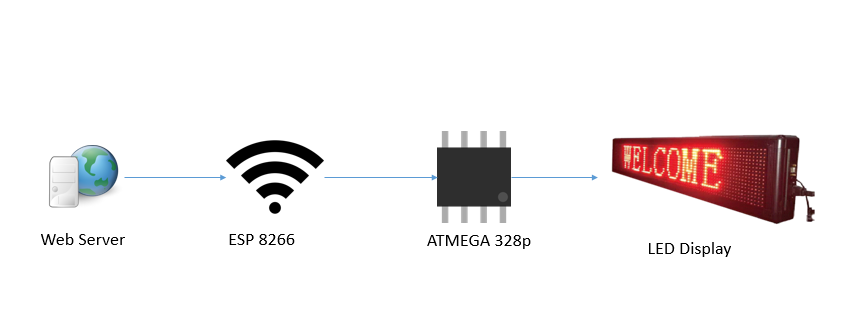


Figure 1. Block Diagram

# Web Server

Web server is the one which provides data which collect from the stations. It provides an API for get data as JSON format.

{

"Record\_time":"2018-04-24 12:19:03",

"Humidity":"73",

"Ext\_temp":"29",

"Int\_temp":"29",

"Intensity":"33",

"Win\_dir":"28.71",

"Win\_speed":"0",

"Rain\_gauge":"0",

"Pressure":"100861",

"Soil\_Moisture":"0"

}

This is sample data format of the API given. We can request data from the station via calling station ID.

# ESP 8266

ESP8266 is a one of common WIFI module. It has inbuilt wifi controlling functions as well as customized programmable microcontroller. Node MCU is one of the development board which based on ESP8266 module. In this project communication with web server will be done by this module.

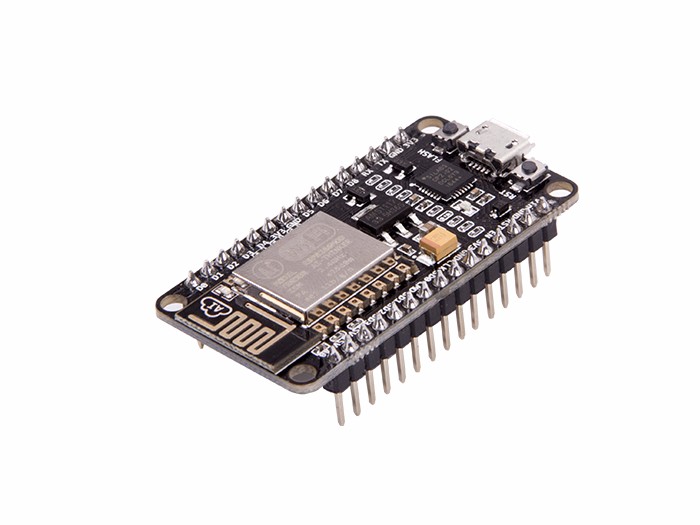


Figure 2. Node MCU module

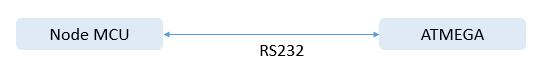
We have programmed this to do the communication with server and by RS232 interface Main controller is communication with this module.

Figure 3 Communication

## Program

We have used ES

#include <ESP8266WiFi.h>

#include <ESP8266WiFiMulti.h>

#include <ESP8266HTTPClient.h>

char\* ssid = "SEC";

char\* password = "sec12Sg920342248";

char\* url ="http://www.slpiot.org/api/getLatestRecordForLCD/5bf82c59-7ec0-4f";

//[{"Record\_time":"2018-04-24 12:19:03","Humidity":"73","Ext\_temp":"29","Int\_temp":"29","Intensity":"33","Win\_dir":"28.71","Win\_speed":"0","Rain\_gauge":"0","Pressure":"100861","Soil\_Moisture":"0"}]

ESP8266WiFiMulti WiFiMulti;

#define USE\_SERIAL Serial

void setup() {

Serial.begin(2400);

WiFi.mode(WIFI\_STA);

WiFiMulti.addAP(ssid, password);

}

void loop() {

if(Serial.available()){

char c = Serial.read();

if(c=='R'){

ReadData();

}

Serial.flush();

}

}

void ReadData(){

// wait for WiFi connection

if((WiFiMulti.run() == WL\_CONNECTED)) {

HTTPClient http;

http.begin(url); //HTTP

int httpCode = http.GET();

if(httpCode > 0) {

if(httpCode == HTTP\_CODE\_OK) {

String payload = getArray( http.getString());

USE\_SERIAL.println(payload);

}

} else {

USE\_SERIAL.println("ERR");

}

http.end();

}

}

String getArray(String h){

String y="";

uint8\_t ch =0;

for(int i=0;i<h.length();i++){

char f = h.charAt(i);

if(f=='{')

ch=1;

if(ch==1){

y += f;

}

if(f=='}')

ch=0;

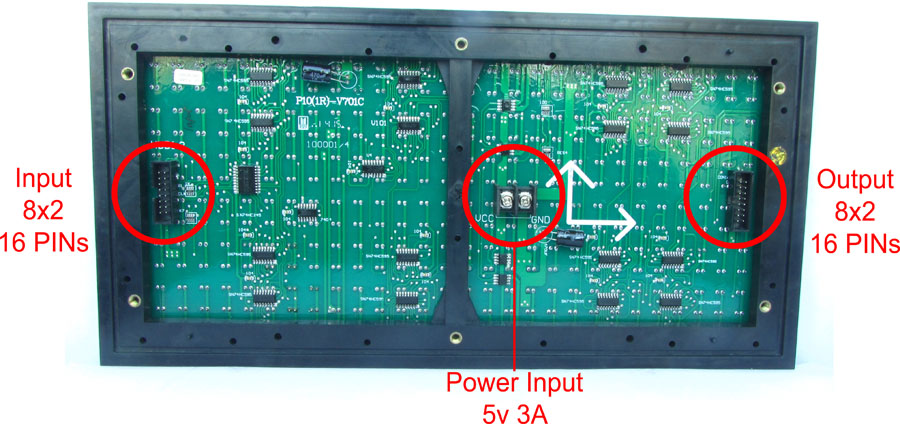
}

return y;

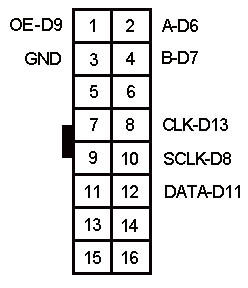
}

# P10 Interface

P 10 is shift controlled Led Matrix. Each channel is run by 74HC25, 74LS138 and 74LS595 shift drivers. In this project we have used Red color LED matrix.



Input connector has this ports.



EN Communicate as Output enable. A and B is considering data with Rows and Columns. CLK, SCLK, DATA is connected to SPI port of the microcontroller.

|  |  |
| --- | --- |
| Panel Pin | MCU Pin (Arduino uno Indexed) |
| 1 | D9 |
| 2 | D6 |
| 4 | D7 |
| 8 | D3 |
| 10 | D8 |
| 12 | D11 |

This is the pattern Code for ATMEGA328P. for drive the Metrix we used DMD2 Library.

<https://github.com/pkourany/DMD2_Library> is the github link for library.

Here is the code.

#include <SoftwareSerial.h>

#include <ArduinoJson.h>

#include <SPI.h>

#include <DMD2.h>

#include <SystemFont5x7.h>

#include <Arial\_Black\_16.h>

#define W 3

#define H 1

SoftDMD dmd1(W,H); // DMD controls the entire display

SoftDMD dmd2(W,H);

#define BRIGHT 150

int width\_changer=8;

SoftwareSerial wifi(2,3);

String message;

unsigned long last=0;

void setup() {

// put your setup code here, to run once:

wifi.begin(2400);

delay(3000);

Serial.begin(9600);

message = ReadWifi();

dmd1.setBrightness(BRIGHT);

dmd1.selectFont(Arial\_Black\_16);

dmd1.begin();

dmd2.setBrightness(BRIGHT);

dmd2.selectFont(SystemFont5x7);

dmd2.begin();

dmd2.drawString(0,0,F("KW Setup"));

dmd2.fillScreen(true);

delay(10);

dmd2.fillScreen(false);

delay(10);

}

int count =0;

void loop() {

if(count > 30){

asm volatile (" jmp 0");

}

DynamicJsonBuffer jsonBuffer;

JsonObject& root = jsonBuffer.parseObject(message);

drawTemperature(root[F("Ext\_temp")].as<String>());

drawHumidity(root[F("Humidity")].as<String>());

delay(2000);

for(int i=0;i<16;i++)

dmd1.scrollY(-1);

drawWDS(getVane(root[F("Win\_dir")].as<String>()),root[F("Win\_speed")].as<String>(),0);

delay(2000);

for(int i=0;i<16;i++)

dmd2.scrollY(-1);

drawRGP(root[F("Rain\_gauge")].as<String>(),root[F("Pressure")].as<String>(),0);

delay(2000);

for(int i=0;i<16;i++)

dmd2.scrollY(-1);

drawSML(root[F("Soil\_Moisture")].as<String>(),root[F("Intensity")].as<String>(),0);

delay(2000);

dmd2.fillScreen(false);

count++;

}

String getVane(String s){

return s;

}

String ReadWifi(){

Serial.println("RE");

wifi.println("R");

String rec=String("");

unsigned long st = millis();

uint8\_t et =0;

while(((millis() - st) < 10000UL) && et == 0) {

if(wifi.available()){

char c= (char)wifi.read();

rec += c;

Serial.write(c);

if(c == '}'){

et = 1;

}

}

}

wifi.flush();

return rec;

}

void drawWDS(String WD,String WS,int h){

String s = F("W. Vane : ");

s.concat(WD);

dmd2.drawString(0,0+h,s);

s = F("W. Speed : ");

s.concat(WS);

dmd2.drawString(0,9+h,s);

}

void drawRGP(String RG,String P,int h){

String s = F("Rain Fall : ");

s.concat(RG);

dmd2.drawString(0,0+h,s);

s = F("Pressure : ");

s.concat(P);

dmd2.drawString(0,9+h,s);

}

void drawSML(String SM,String L,int h){

String s = F("Soil M. : ");

s.concat(SM);

dmd2.drawString(0,0+h,s);

s = F("Solar In : ");

s.concat(L);

dmd2.drawString(0,9+h,s);

}

void drawTemperature(String temp){

dmd1.drawString(1+ width\_changer,0,temp);

dmd1.drawCircle(21 + width\_changer,1,1);

dmd1.drawString(24 + width\_changer,0,"C");

}

void drawHumidity(String h){

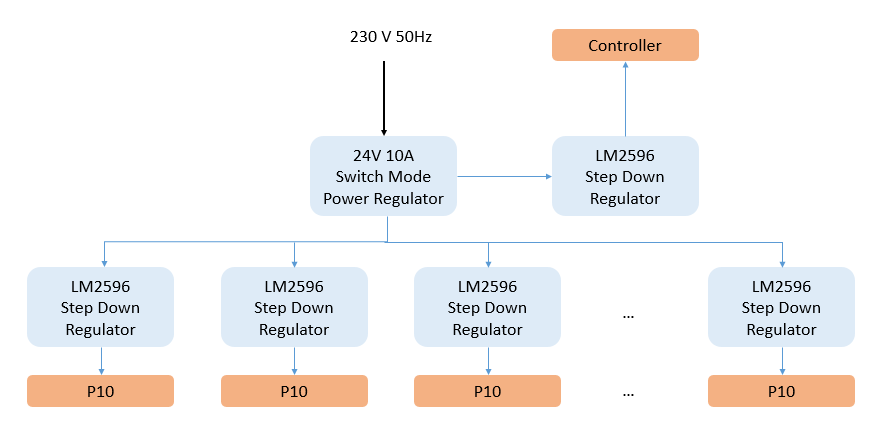
h.concat('%');

dmd1.drawString(50+width\_changer,0,h);

}

# Power Plan

Each module needs 5v Power Supply most probably full screen show display gets around 3A current. For that we have used LM2596 Step down Modules and 24V 10A Switch Mode Power Supply.



# Implemented System

