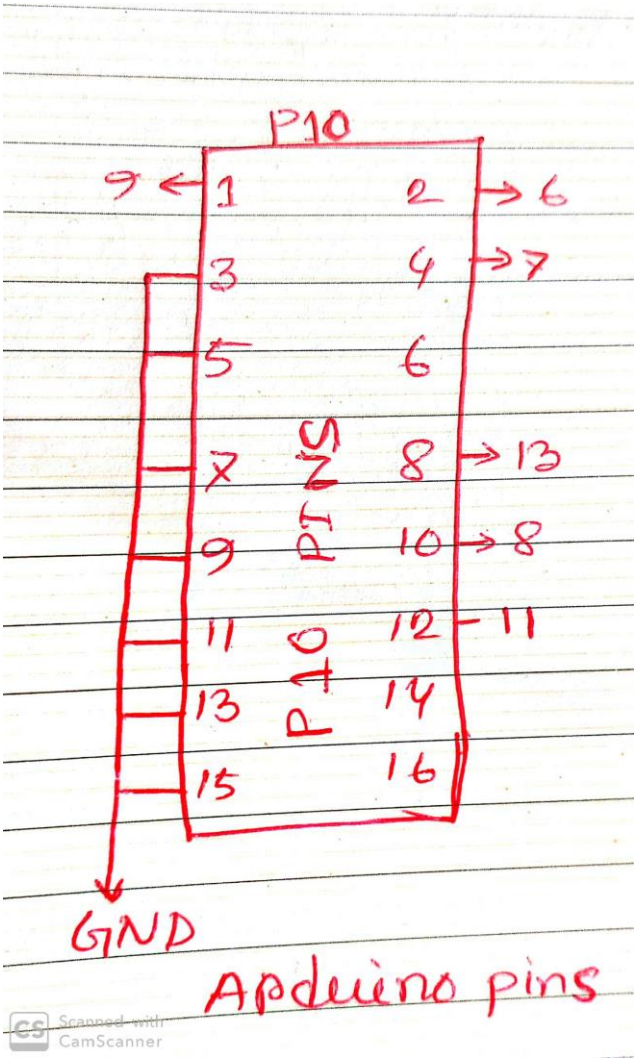
Arduino Documentation

Requirements:

1. P10 display - 1
2. Arduino Uno – 1
3. NodeMCU ESP8266 – 1
4. Wires (M2M, M2F, F2F) – as per requirement

Connection:

1. Arduino to P10:



connect pins as shown in the picture. The number inside the box represents the pins of the P10 display whereas outer pins indicates Arduino Uno boards pins.

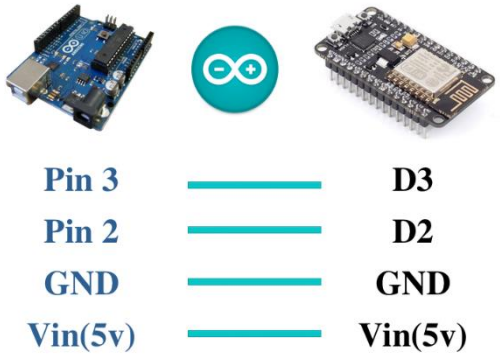
Pin 3,5,7,9,11,13 and 15 should be connected to the GND pin of Arduino board.

In case of P10 display panel, these pins are from the left side.

Make sure the Vcc of P10 panel is connected to 5V DC supply and GND to ground. This power supply is only for brightness purpose.

2. Arduino to NodeMCU ESP8266:

connect Arduino with NodeMCU as shown in the picture. Vin is the power supply to operate the ESP8266 .you need to connect ESP8266 for uploading the code and operation.



3. Code for Arduino board:

```
//Md. Raju Ahmed
//rajucse1705@gmail.com

#include <SoftwareSerial.h>
#include <SPI.h>
#include <DMD2.h>
#include <ArduinoJson.h>
#include <fonts/SystemFont5x7.h>
#include <fonts/Arial14.h>

SoftwareSerial ArduinoUno(3,2);
// Set Width to the number of displays wide you have
const int WIDTH = 1;

// You can change to a smaller font (two lines) by commenting
this line,
// and uncommenting the line after it:
//const uint8_t *FONT = Arial14;
const uint8_t *FONT = SystemFont5x7;

SPIDMD dmd(WIDTH,1);  // DMD controls the entire display
DMD_TextBox box(dmd);  // "box" provides a text box to
automatically write to/scroll the display

// the setup routine runs once when you press reset:
void setup() {
  Serial.begin(9600);
  ArduinoUno.begin(4800);
  dmd.setBrightness(30);
  dmd.selectFont(FONT);
  dmd.begin();
}

// the loop routine runs over and over again forever:
```

```
void loop() {

    while(ArduinoUno.available()>0){
      const size_t capacity = JSON_ARRAY_SIZE(1) +
JSON_OBJECT_SIZE(2) + 30;
      DynamicJsonBuffer jsonBuffer(capacity);

      const char* json = "[{\"counter\":5,\"queue\":565}]";

      JsonArray& root = jsonBuffer.parseArray(ArduinoUno);

      if(root == JsonObject::invalid()){
        return;
        }

      int counter = root[0]["counter"];
      int queue = root[0]["queue"];

      char *Co = "C-";
      char *Qu = "Q-";
      char next[10];

      sprintf(next, " %s%d %s%d", Co,counter, Qu,queue);

      char temp[10];

      if(temp != next){
         box.clear();
         dmd.clearScreen();
         for(int i = 0;i<10;i++) {
         Serial.print(next[i]);
         box.print(next[i]);
         temp[i] = next[i];
          }
        delay(1000); //delay of 1 second.
       }
   }

}
```

4. Code for ESP8266:

```
//Md. Raju Ahmed
//rajucse1705@gmail.com
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>
#include <SoftwareSerial.h>

const char* ssid = "DC";
```

```
const char* password = "CastaLia";

SoftwareSerial NodeMCU(D2,D3);

void setup()
{
  Serial.begin(9600);
  NodeMCU.begin(4800);

  pinMode(D2,INPUT);
  pinMode(D3,OUTPUT);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.println("Connecting...");
  }
}

void loop()
{
  if (WiFi.status() == WL_CONNECTED)
  {
    HTTPClient http; //Object of class HTTPClient
    //http.begin("http://jsonplaceholder.typicode.com/users/1");
    http.begin("http://my-json-
server.typicode.com/rajuAhmed1705/demo/update");
    int httpCode = http.GET();

    if (httpCode > 0)
    {

        const size_t capacity = JSON_ARRAY_SIZE(1) +
JSON_OBJECT_SIZE(2) + 30;
        DynamicJsonBuffer jsonBuffer(capacity);

        const char* json = "[{\"counter\":5,\"queue\":565}]";

        JsonArray& root =
jsonBuffer.parseArray(http.getString());

        int root_0_counter = root[0]["counter"]; // 5
        int root_0_queue = root[0]["queue"]; // 565


        root.printTo(Serial);
```

```
        Serial.println(root_0_counter);
        Serial.println(root_0_queue);


        //char *msg = strcat(strcat("C-
",root_0_counter),strcat("Q-",root_0_queue));
        //Serial.println(*msg);

        //NodeMCU.print("C-%d Q-%d",root_0_counter,root_0_queue);
        //NodeMCU.println("\n");
        //NodeMCU.print(root_0_queue);
        //NodeMCU.println("\n");

        root.printTo(NodeMCU);



    }
    http.end(); //Close connection
  }
  delay(1000);
}
```

5. Uploading codes to the board:
   Before you upload codes to the board you need to have all the library in the Arduino IDE. List of libraries are:-
   i) `ESP8266WiFi`
   ii) `ArduinoJson`
   iii) `DMD2`
   These libraries are not included in the IDE. You need to add them manually.

   If everything goes right you are good to go. Thank you.
   *codes are also included in separate file.