**Build a Rolling LED Display**

A calculator and a digital sign

Description automatically generated

# Contents

## Prerequisites

## Aim

## Components

## Connections

### - Circuit Diagram

### - Detailed Steps

## Software

### - Launching the IDE

### - Code

## Tasks

Prerequisites

|  |  |
| --- | --- |
| Topic | Resources |
| Dancing LEDs Project | Dancing LEDs Project |
| Basic C Programming | [C Full Course - Youtube](https://www.youtube.com/watch?v=87SH2Cn0s9A) |
|  | [W3Schools interactive C Tutorial](https://www.w3schools.com/c/c_getstarted.php) |
|  | [Learn-C.org interactive Tutorial](https://www.learn-c.org/) |
| Number Systems | [Binary,Decimal & HexaDecimal](https://www.youtube.com/watch?v=J3ekb5JnX8Y) |

Aim

Build an LED matrix display and send messages to it wirelessly from your phone using Bluetooth. This includes controlling the Direction and Speed of scrolling as well as Brightness of the display remotely from your phone.

Components

1. Esp 32 Module

2. LED Dot Matrix display - MAX 7219 board with four 8x8 displays

mounted.

3. Breadboard

4. Jumper wires (9 male to female, 2 male to male)

5. 1 USB cables - 1 for powering the Esp32 from laptop to Esp32.

|  |  |
| --- | --- |
| ESP32 C-6 Module | Dot Matrix Display |

|  |  |  |
| --- | --- | --- |
| A group of colorful wires  Description automatically generated  Jumper Wires | A close-up of a usb cable  Description automatically generated  USB | A close-up of a white circuit board  Description automatically generated  Breadboard |

Connections

Safety tip: Always ensure that the connections to the components are correct and completed before connecting the power supply to the Esp32.

Circuit Diagram

A rolling display with a wire

AI-generated content may be incorrect.

Follow the detailed steps in the following pages to complete the circuit.

NOTE: Before starting the connections, verify that all the jumper wires are working using a multimeter. Also ensure that the connections are strong, or else the setup may not work.

Detailed Connection Steps

Take 5 female-to-male jumper wires and connect them to the pins (VCC, GND,

DIN, CS, CLK) of the MAX7219 display board as shown below. Connect the other ends as per the below connections:

- VCC to Red line of breadboard

- GND to Blue line of breadboard

- DIN to GPIO15 of ESP32

- CS to GPIO4 of ESP32

- CLK to GPIO19 of ESP32

You are now ready to work on the software for the project.

Software

Launching the IDE for our project

1. Install the Arduino IDE

 If you haven't already installed the Arduino IDE, download and install it from the official Arduino website.

2. Install the ESP32 Board in Arduino IDE

* Open the Arduino IDE.
* Go to **File > Preferences.**
* In the "Additional Board Manager URLs" field, add the following URL:

|  |
| --- |
| [https://raw.githubusercontent.com/espressif/arduino-esp32/gh pages/package\_esp32\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh%20pages/package_esp32_index.json) |

* Go to **Tools > Board > Boards Manager**.
* Search for ESP32 and install the esp32 package by Espressif Systems.

3. Install Required Libraries

* Go to **Sketch > Include Library > Manage Libraries**.

Search for and install the following libraries:

* Adafruit GFX Library
* Max72xxPanel

-Inside the extracted project file, you will find folder named

“arduino-Max72xxPanel-master”. Copy them and paste them inside the ‘Libraries’ folder located in Documents > Arduino > Libraries.

* BluetoothSerial (if not included by default with the ESP32 package)

4. Connect Your ESP32 Board

* Connect your ESP32 board to your computer using a USB cable.
* Go to Tools > Board > ESP32 Dev Module.
* Go to Tools > Port and select the COM port to which your ESP32 is connected.

5. Prepare the Hardware

Connect your LED matrix to the ESP32. For the example code, ensure the CS pin of the matrix is connected to GPIO 5 of the ESP32.

6. Upload the Code

* Copy the provided code below into a new sketch in the Arduino IDE.
* Click the upload button (right arrow) to compile and upload the code to your ESP32.

7. Pairing the ESP32 via Bluetooth

* Once the code is uploaded, open the Serial Monitor from **Tools > Serial Monitor** and set the baud rate to 115200.
* Pair your ESP32 with your computer or smartphone. The device name should be ESP32test.
* Once paired, you can send text messages to the ESP32 via a Bluetooth terminal app on your smartphone or a serial terminal on your computer.

Once the code is uploaded to ESP32, the LED matrix will start displaying the text in various alignments.

Code to Upload

Ensure you have the following code in your Arduino IDE before uploading:

Level 1: Just display the text

For this level, we'll focus on setting up the LED matrix and displaying static text.

* Create a new sketch on arduino IDE and name it displayText.ino.
* Copy the code given below and make some changes according to the input.

Here's the simplified code:

|  |
| --- |
| #include <SPI.h>  #include <Adafruit\_GFX.h>  #include <Max72xxPanel.h>  const int pinCS = 4;  // CLK - GPIO 19 (For C-6 Board)  // DIN - GPIO 15 (For C-6 Board)  const int numberOfHorizontalDisplays = 4;  const int numberOfVerticalDisplays = 1;  Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays, numberOfVerticalDisplays);  void setup()  {  initializeMatrix();  displayStaticText("Hello");  }  void loop()  {  }  void initializeMatrix()  {  matrix.setIntensity(15); // Adjust the brightness between 0 and 15  for (int i = 0; i < numberOfHorizontalDisplays; i++)  {  matrix.setPosition(i, i, 0); // Set positions for all displays. First i is the index, second i is the horizontal panel and third is the vertical panel  matrix.setRotation(i, 1); // Set rotation for all displays  }  }  void displayStaticText(String text)  {  matrix.fillScreen(LOW);  matrix.setCursor(1, 0);  matrix.print(text);  matrix.write();  } |

Level 2: Scroll the text

|  |
| --- |
| #include <SPI.h>  #include <Adafruit\_GFX.h>  #include <Max72xxPanel.h>  const int pinCS = 4;  const int numberOfHorizontalDisplays = 4;  const int numberOfVerticalDisplays = 1;  //Pin Number 15 - DIN  //Pin Number 19 - CLK  Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays, numberOfVerticalDisplays);  void setup() {    initializeMatrix();  }  void loop() {    displayScrollingText("Hello");  }  void initializeMatrix()  {    matrix.setIntensity(1); // Adjust brightness    for (int i = 0; i < numberOfHorizontalDisplays; i++)    {      matrix.setPosition(i, i, 0); // Set panel positions      matrix.setRotation(i, 1);    // Set panel rotation    }  }  void displayScrollingText(String text)  {    int textWidth = text.length() \* 6; // Each character is 6 pixels wide    for (int i = numberOfHorizontalDisplays \* 8; i>-textWidth; i--)  // To scroll text from right to left    {      matrix.fillScreen(LOW);       // Clear the display      matrix.setCursor(i, 1);       // Move text horizontally      matrix.print(text);           // Draw the text      matrix.write();               // Update the display      delay(100);                    // Adjust speed of scrolling    }  } |

Level 3: Add Wireless Control functionality via IP (Internet Protocol)

|  |
| --- |
| #include <WiFi.h>  #include <MD\_Parola.h>  #include <MD\_MAX72XX.h>  #include <SPI.h>  // ---------------- DISPLAY CONFIGURATION ----------------  #define HARDWARE\_TYPE MD\_MAX72XX::FC16\_HW  #define MAX\_DEVICES 4  #define DATA\_PIN 15  #define CS\_PIN 4  #define CLK\_PIN 19  MD\_Parola display = MD\_Parola(HARDWARE\_TYPE, DATA\_PIN, CLK\_PIN, CS\_PIN, MAX\_DEVICES);  // ---------------- WIFI CONFIGURATION ----------------  const char\* ssid = "YOUR\_WIFI\_SSID";  const char\* password = "YOUR\_WIFI\_PASSWORD";  WiFiServer server(80);  // ---------------- GLOBAL STATE ----------------  String inputText = "HELLO!";  uint16\_t scrollSpeed = 100; // in milliseconds (corrected)  textEffect\_t scrollDir = PA\_SCROLL\_LEFT;  uint8\_t brightness = 3;  void setup() {  Serial.begin(115200);  display.begin();  display.setIntensity(brightness);  display.displayClear();  display.displayScroll(inputText.c\_str(), PA\_LEFT, scrollDir, scrollSpeed);  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println("\nWiFi connected: ");  Serial.println(WiFi.localIP());  server.begin();  }  void loop() {  if (display.displayAnimate()) display.displayReset();  WiFiClient client = server.available();  if (client) {  String request = client.readStringUntil('\r');  client.flush();  // --- Parse Parameters Safely ---  if (request.indexOf("GET /?") >= 0) {  if (request.indexOf("text=") >= 0) {  int start = request.indexOf("text=") + 5;  int end = request.indexOf("&", start);  if (end == -1) end = request.indexOf(" ", start);  inputText = request.substring(start, end);  inputText.replace("+", " ");  inputText.replace("%20", " ");  }  if (request.indexOf("speed=") >= 0) {  int start = request.indexOf("speed=") + 6;  int end = request.indexOf("&", start);  if (end == -1) end = request.indexOf(" ", start);  scrollSpeed = request.substring(start, end).toInt();  }  if (request.indexOf("bright=") >= 0) {  int start = request.indexOf("bright=") + 7;  int end = request.indexOf("&", start);  if (end == -1) end = request.indexOf(" ", start);  brightness = constrain(request.substring(start, end).toInt(), 0, 15);  }  if (request.indexOf("dir=") >= 0) {  int start = request.indexOf("dir=") + 4;  int end = request.indexOf("&", start);  if (end == -1) end = request.indexOf(" ", start);  String dirVal = request.substring(start, end);  if (dirVal == "right") scrollDir = PA\_SCROLL\_RIGHT;  else scrollDir = PA\_SCROLL\_LEFT;  }  // Update display  display.setIntensity(brightness);  display.displayClear();  display.displayScroll(inputText.c\_str(), PA\_LEFT, scrollDir, scrollSpeed);  }  // --- HTML Response ---  client.println("HTTP/1.1 200 OK");  client.println("Content-Type: text/html\n");  client.println("<!DOCTYPE html><html><head><title>LED Matrix</title>");  client.println("<meta name='viewport' content='width=device-width, initial-scale=1.0'>");  client.println("<style>");  client.println("body{background:#111;color:#fff;font-family:sans-serif;padding:20px;text-align:center;}");  client.println("input,select{padding:10px;margin:5px;border:none;border-radius:5px;}");  client.println("input[type=range]{width:80%;}");  client.println("button{padding:10px 20px;border:none;background:#09f;color:#fff;border-radius:5px;}");  client.println("</style></head><body>");  client.println("<h2>ESP32 LED Matrix Control</h2>");  client.println("<form method='GET'>");  // Text  client.println("Text:<br><input type='text' name='text' value='" + inputText + "'><br>");  // Scroll speed (invert user input logic)  int userSpeed = scrollSpeed;  client.println("Speed (ms):<br><input type='range' min='10' max='300' name='speed' value='" + String(userSpeed) + "'><br>");  // Brightness  client.println("Brightness:<br><input type='range' min='0' max='15' name='bright' value='" + String(brightness) + "'><br>");  // Direction  client.println("Direction:<br><select name='dir'>");  client.println("<option value='left'" + String(scrollDir == PA\_SCROLL\_LEFT ? " selected" : "") + ">Left</option>");  client.println("<option value='right'" + String(scrollDir == PA\_SCROLL\_RIGHT ? " selected" : "") + ">Right</option>");  client.println("</select><br><br>");  client.println("<button type='submit'>Update</button>");  client.println("</form></body></html>");  delay(10);  client.stop();  }  } |

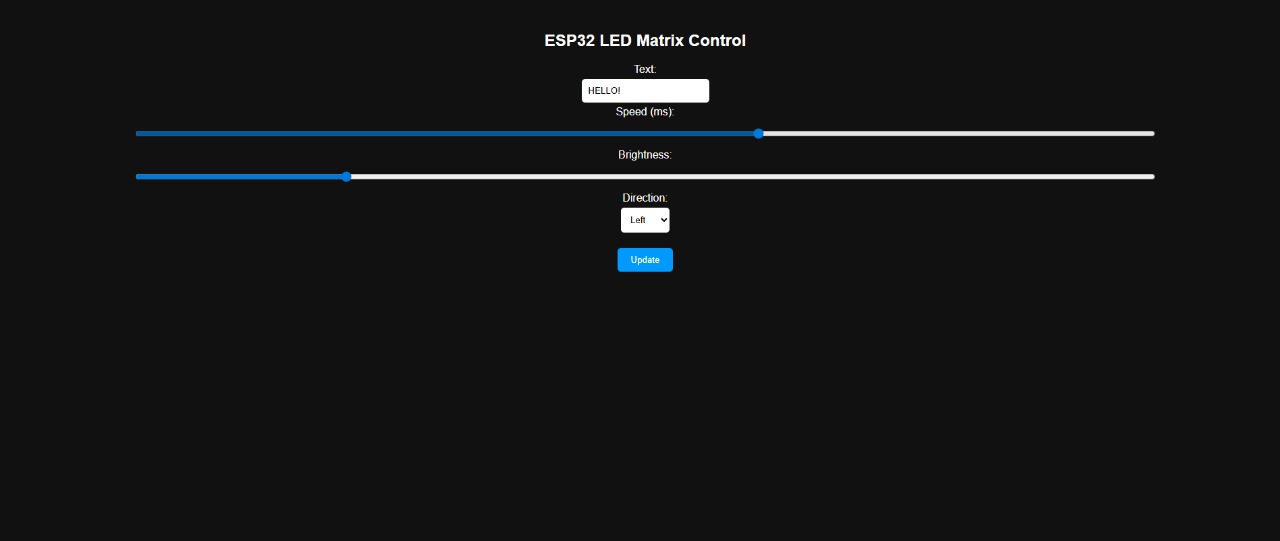
8. Test the Setup

1. After uploading the code, open serial monitor and copy the IP server id like (192.168.43.50) and paste it on your local network web browser to enter the text.

A screenshot of a computer

AI-generated content may be incorrect.

1. After paste the IP server id in the local web browser, it looks like below figure



1. The LED matrix should start displaying the text in various alignments as specified in the code.

Tasks:

1) Write code to print digits 0-9 on the display. Try similarly for punctuation marks ‘!’, ‘?’, ‘;’.

2) Design 3 emojis or symbols of your choice and print them on the display.

3) Write a program that animates a ‘Stick man’ walking from one end of the display to the other.

4) Plan an animation clip of minimum 10 seconds with a storyline and run it on the display.