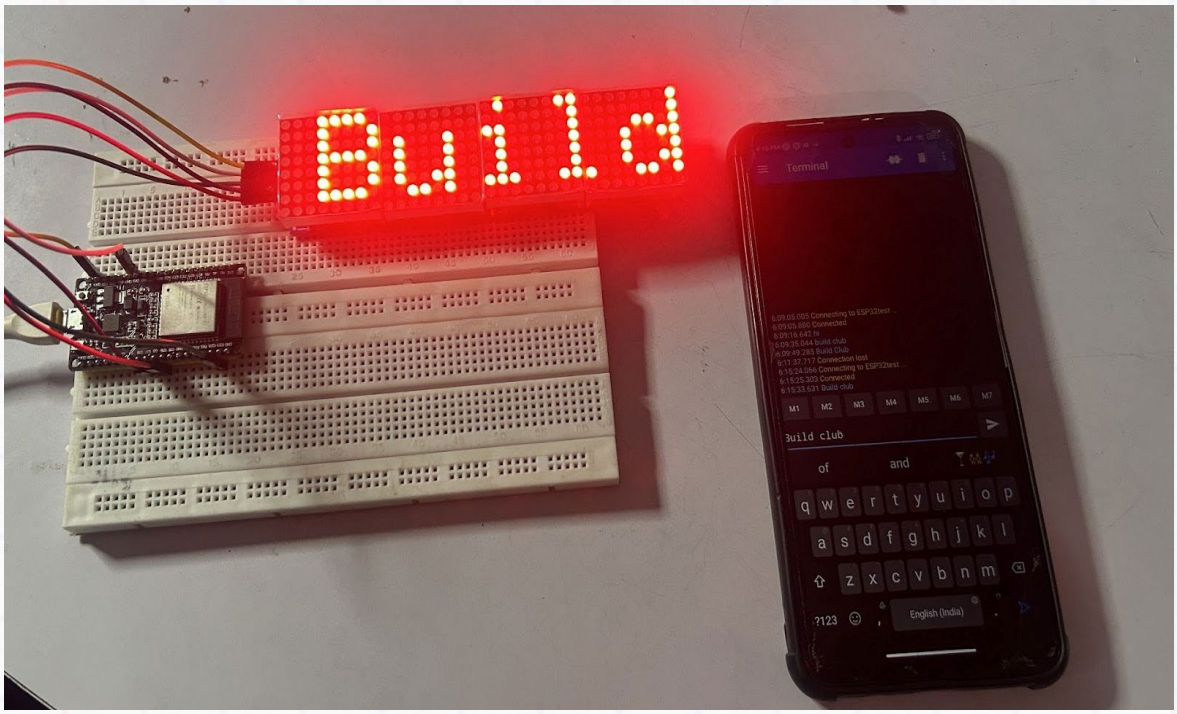
Build a Rolling LED Display



Contents

Prerequisites

Topic	Resources
Dancing LEDs Project	Dancing LEDs Project
Basic C Programming	C Full Course - Youtube
	W3Schools interactive C Tutorial
	Learn-C.org interactive Tutorial
Number Systems	Binary,Decimal & HexaDecimal

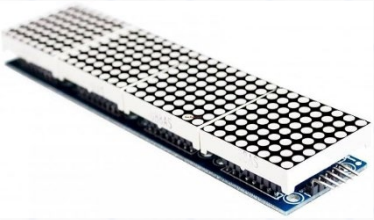
Aim

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 Module



Dot Matrix Display

Jumper Wires

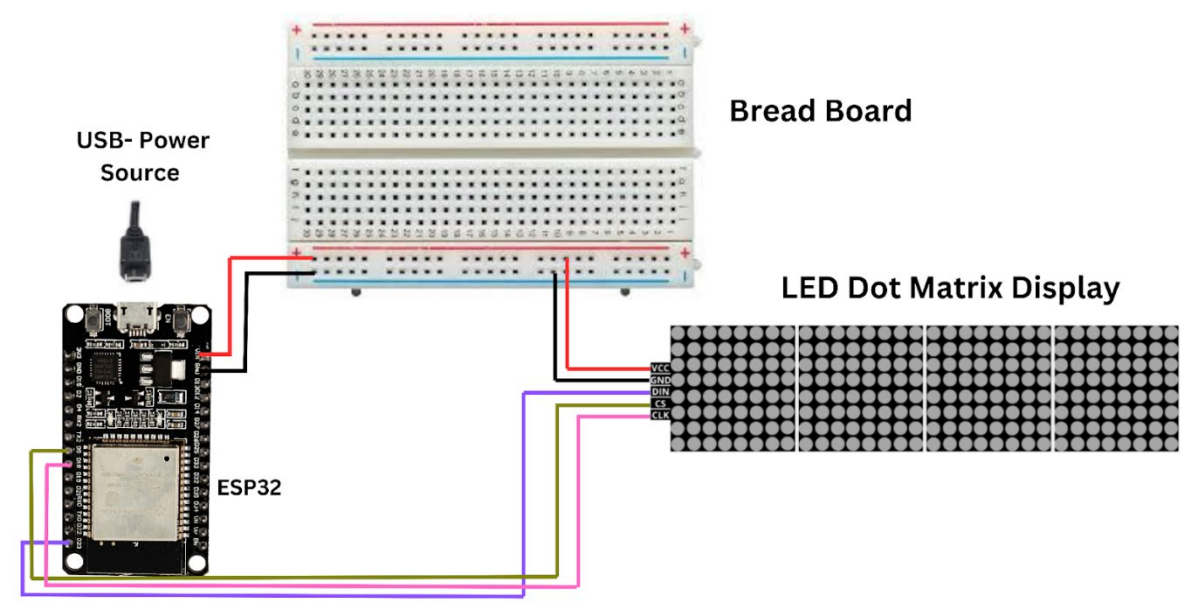
USB

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



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 D23 of ESP32
- CS to D5 of ESP32
- CLK to D18 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

- 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 = 5;
const int numberOfHorizontalDisplays = 8;
const int numberOfVerticalDisplays = 1;
Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays,
numberOfVerticalDisplays);
```

```
void setup() {

}
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
    matrix.setRotation(i, 1);     // Set rotation for all displays
  }
}
```

```
void displayStaticText(String text) {
  matrix.fillScreen(LOW);
  matrix.setCursor(0, 0);
  matrix.print(text);
  matrix.write();
};
```

To use these functions:

- Call **initializeMatrix()** in your **setup()** function.
- In your **loop()** function, use **displayStaticText("Your text here")** to show static text on the LED matrix.

Level 2: Scroll the text

Continue on Level 1, we'll add scrolling functionality:

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>

const int pinCS = 5;
const int numberOfHorizontalDisplays = 8;
const int numberOfVerticalDisplays = 1;
Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays,
numberOfVerticalDisplays);
```

```
const int wait = 40;
const int spacer = 1;
const int width = 5 + spacer; // Font width 5 pixels

int textLength = 0;

void setup() {
  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
    matrix.setRotation(i, 1);     // Set rotation for all displays
  }
  textLength = width * textin.length() + matrix.width() - 1 - spacer;
}

void loop() {

}
```

Define textin as a global variable with your desired text.

```
String textin = "Hello, World!";
```

Call **scrollText()** in your **loop()** function to continuously scroll the text.

```
void scrollText() {
  for (int i = 0; i < textLength; i++) {
    matrix.fillScreen(LOW);
    int letter = i / width;
    int x = (matrix.width() - 1) - i % width;
    int y = (matrix.height() - 8) / 2; // Center the text vertically

    while (x + width - spacer >= 0 && letter >= 0) {
      if (letter < textin.length()) {
        matrix.drawChar(x, y, textin[letter], HIGH, LOW, 1);
      }
      letter--;
      x -= width;
    }
    matrix.write();
    delay(wait);
  }
}
```

Level 3: Add Bluetooth functionality

Finally, we'll add Bluetooth to allow dynamic text input:

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

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>
#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` and enable it
#endif

BluetoothSerial SerialBT;

const int pinCS = 5;
const int numberOfHorizontalDisplays = 8;
const int numberOfVerticalDisplays = 1;
Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays,
numberOfVerticalDisplays);

const int wait = 40;
const int spacer = 1;
const int width = 5 + spacer; // Font width 5 pixels

String receivedData = "";
String textin = "Hello, World!";
int textLength = 0;

void setup() {
  Serial.begin(115200);

  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
    matrix.setRotation(i, 1);     // Set rotation for all displays
  }
  textLength = width * textin.length() + matrix.width() - 1 - spacer;
}

void loop() {
```

```
  scrollText();
}
```

Call **initializeBluetooth()** in your **setup()** function.

```
void initializeBluetooth() {
  SerialBT.begin("ESP32test"); // Bluetooth device name
  Serial.println("The device started, now you can pair it with Bluetooth!");
}
```

Call **handleBluetoothInput()** in your **loop()** function to check for and process incoming Bluetooth messages.

```
void handleBluetoothInput() {
  if (SerialBT.available()) {
    char incomingChar = SerialBT.read();
    if (incomingChar == '\n') {
      Serial.print("Received message: ");
      Serial.println(receivedData);
      textin = receivedData;
      receivedData = "";
      calculateTextLength();
    } else {
      receivedData += incomingChar;
    }
  }
}
```

- In **setup()**:
  - Call **initializeMatrix()**
  - Call **initializeBluetooth()**
  - Set initial textin value and call **calculateTextLength()**
- In **loop()**:
  - Call **handleBluetoothInput()**
  - Call **scrollText()**

Full Code

- Create a new sketch with name ***rollingDisplay.ino***
- Copy the below code and paste it on the above created file.

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>
#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` and enable it
#endif

BluetoothSerial SerialBT;
String receivedData = "";
String textin = "";
const int pinCS = 5;
const int numberOfHorizontalDisplays = 8;
const int numberOfVerticalDisplays = 1;
Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays,
numberOfVerticalDisplays);
const int wait = 40;
const int spacer = 1;
const int width = 5 + spacer; // Font width 5 pixels

bool scrollActive = true; // Flag to indicate if scrolling is active
int textLength = 0; // Length of the text message

void setup() {
  digitalWrite(pinCS, LOW);
  Serial.begin(115200);
  SerialBT.begin("ESP32test"); // Bluetooth device name
  Serial.println("The device started, now you can pair it with Bluetooth!");

  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
    matrix.setRotation(i, 1);     // Set rotation for all displays
  }
}

void loop() {
  if (SerialBT.available()) {
```

```
    char incomingChar = SerialBT.read();
    if (incomingChar == '\n') {
      // Handle the complete message stored in receivedData
      Serial.print("Received message: ");
      Serial.println(receivedData);
      String data = receivedData;
      data.remove(data.length() - 1, 1); // Remove the newline character
      textin = data;
      scrollActive = true; // Start scrolling the new message
      receivedData = "";   // Clear receivedData for the next message

      // Calculate the total length of the text message
      textLength = width * textin.length() + matrix.width() - 1 - spacer;
    } else {
      // Append the incoming character to receivedData
      receivedData += incomingChar;
    }
  }

  if (scrollActive) {
    scrollText();
  }
}

void scrollText() {
  static unsigned long lastScrollTime = 0;
  static int scrollPosition = 0;

  if (millis() - lastScrollTime >= wait) {
    lastScrollTime = millis();

    matrix.fillScreen(LOW);

    // Calculate the current position to display the text
    int letter = scrollPosition / width;
    int x = (matrix.width() - 1) - scrollPosition % width;
    int y = (matrix.height() - 8) / 2; // Center the text vertically

    // Draw each character of the text at its calculated position
    while (x + width - spacer >= 0 && letter >= 0) {
      if (letter < textin.length()) {
        matrix.drawChar(x, y, textin[letter], HIGH, LOW, 1);
      } else {
        // Clear leftover space at the end of the message
        matrix.drawChar(x, y, ' ', HIGH, LOW, 1);
```

```
      }
      letter--;
      x -= width;
    }

    matrix.write(); // Display the characters

    // Update scroll position for the next frame
    scrollPosition++;

    // Reset scroll position when entire text has been scrolled
    if (scrollPosition >= textLength) {
      scrollPosition = 0;
    }
  }
}
```

8. Test the Setup

1. After uploading the code, use a Bluetooth terminal app to connect to your ESP32 and send a text message.

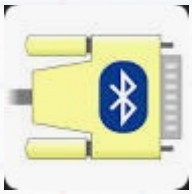
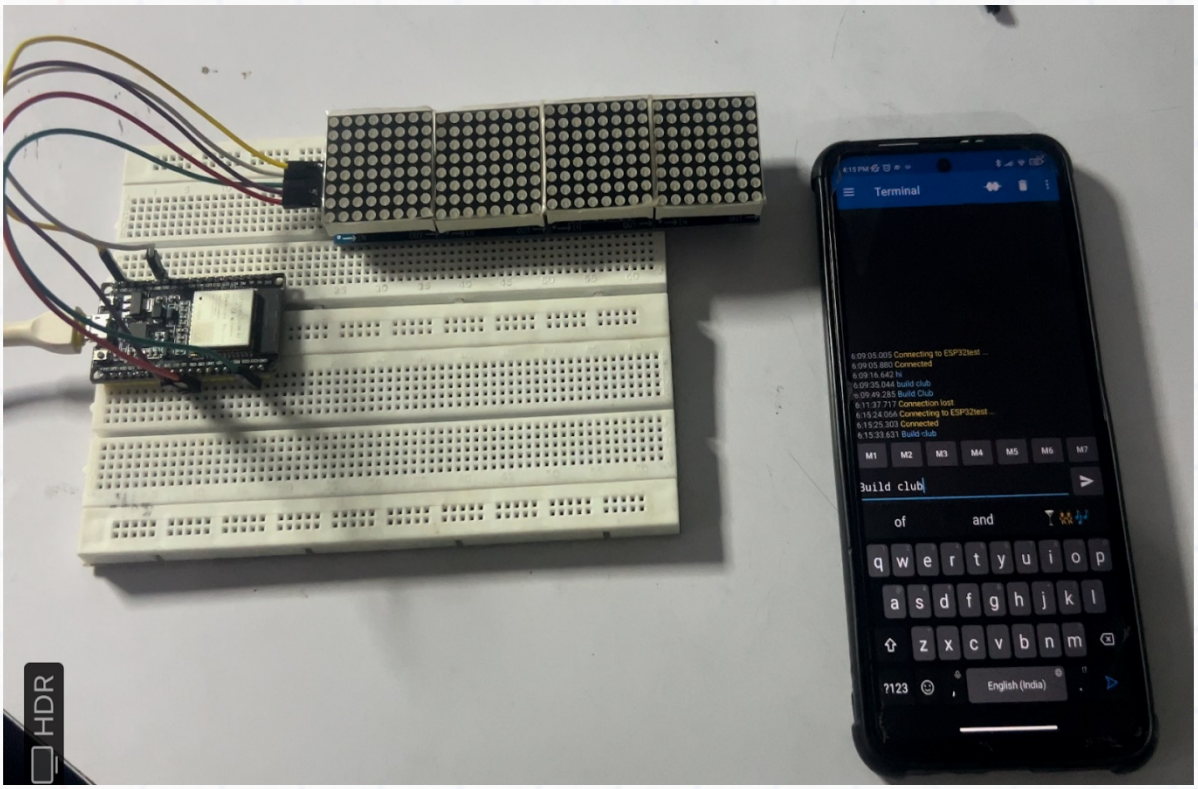


Fig: Serial Bluetooth Terminal - Mobile App

2. 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.

Whoa! You just built your own TV! Take a video clip of an animation movie and post it on the [Build Club Discord Server]