

Ziel des Projekts

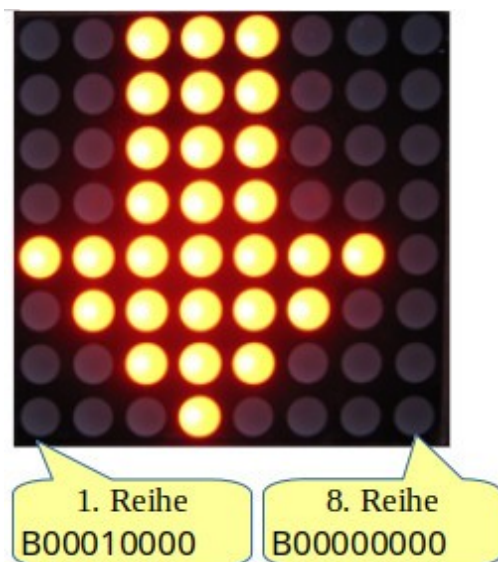
Auf einer LED-Matrix wird eine wählbare Anzahl von LEDs zufällig auf der LED-Matrix verteilt. Ein Joystick bewegt eine LED über das "Spielfeld", und "löscht" alle LEDs, die sich auf der jeweiligen Position befinden.

Die Hardware

LED-Matrix

Die hier verwendete LED-Matrix mit der Bezeichnung Max7219 besteht aus 8x8 LEDs. Die einzelnen LEDs werden in Zeilen und Spalten angesprochen.

Die Schreibweise für jede Zeile kann binär angegeben werden: 0 = aus, 1 = an. Den Werten wird ein "B" vorangestellt.

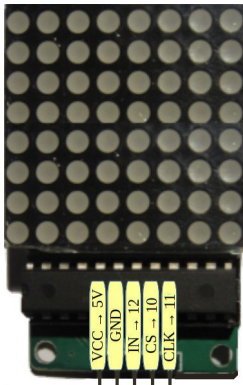


Die LED-Matrix wurde in das Steckbrett eingesetzt, daher verlaufen die Reihen vertikal und die Spalten horizontal.

```
void PfeilUnten()
{
    byte Zeichen[8] =
    {
        B00010000, // 1. Reihe
        B00110000, // 2. Reihe
        B01111111, // 3. Reihe
        B11111111, // 4. Reihe
        B01111111, // 5. Reihe
        B00110000, // 6. Reihe
        B00010000, // 7. Reihe
        B00000000  // 8. Reihe
    };
};
```

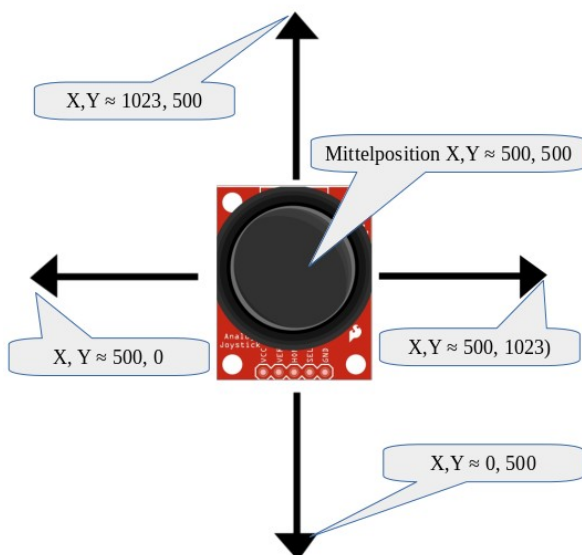


```
// Matrix anzeigen
for (int i = 0; i < 8; i ++){
  LEDMatrix.setRow(0, i, Zeichen[i]);
}
}
```



Pinbelegung der LED-Matrix.

Joystick

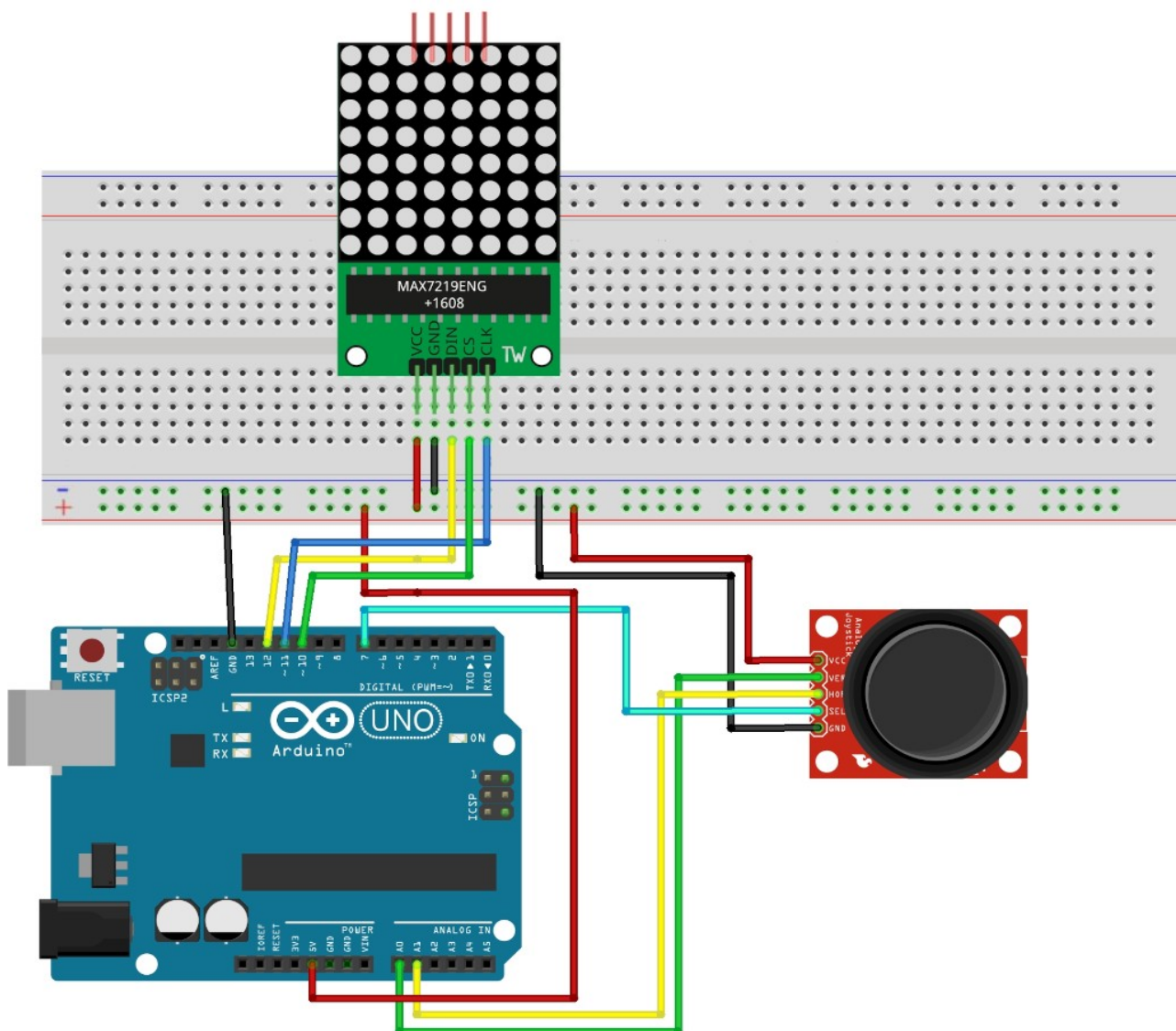


Der Joystick besteht aus zwei Potentiometern, jeweils einer für die X-Achse und einer für die Y-Achse. Beide lesen bei den Bewegungen die Spannung und liefern dem Arduino jeweils einen analogen Wert, der zwischen 0 und 1023 liegt. Der Joystick besteht aus zwei Potentiometern, jeweils einer für die X-Achse und einer für die Y-Achse. Beide lesen bei den Bewegungen die Spannung und liefern dem Arduino jeweils einen analogen Wert, der zwischen 0 und 1023 liegt.

Benötigte Bauteile

- Joystick
- LED-Matrix
- Leitungsdrähte

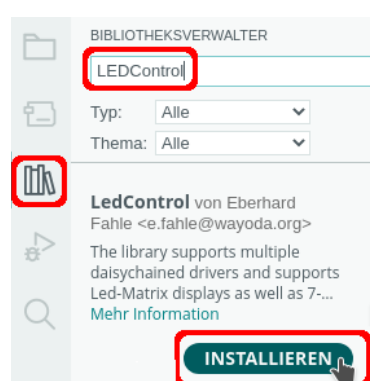
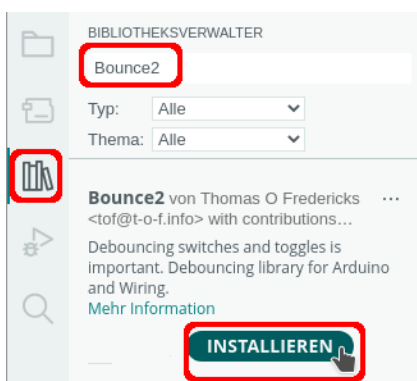
Der Schaltplan



fritzing

Benötigte Bibliotheken

Sketch → Bibliothek einbinden → Bibliothek verwalten





Funktionen der Bibliothek LEDControl

Schlüsselwort	Aktion
LedControl Name_der_Matrix(Data-In, CLK, CS, AnzahlMatrix)	LED-Matrix initialisieren: LedControl LEDMatrix = LedControl(12, 11, 10, 1);
shutDown(NummerMatrix, true/false)	Matrix aufwecken: shutDown(0, false);
setIntensity(NummerMatrix, Helligkeit)	Helligkeit setzen (0–20) setIntensity(0, 8);
clearDisplay(NummerMatrix)	clearDisplay(0);
setLed(NummerMatrix, Zeile, Spalte, true/false);	einzelne LED schalten setLed(0, 0, 0, true);
setRow(NummerMatrix, Zeile, Byte)	zeilenweise ein Byte schreiben: setRow(0, Zeile, B10000000); setRow kann Werte zwischen 0 und 7 haben
setColumn(NummerMatrix, Spalte, Byte)	spaltenweise ein Byte schreiben: setColumn(0, Spalte, B10000000); setColumn kann Werte zwischen 0 und 7 haben

Das Programm

Im Kopf des Programms musst du die benötigte Bibliotheken einbinden und die Pinbelegung der LED-Matrix festlegen.

```
#include "LedControl.h"
#include "Bounce2.h"

// Joystick-Button zuordnen
Bounce ZeitStoppen = Bounce();

// Joystick analoge Pins
int XAchse = A0;
int YAchse = A1;

// Joystick Knopf
int JoystickButton = 7;

// Variablen für die Auswertung der Bewegung des Joysticks
int PositionX;
int PositionY;

// Startposition der LED
int Reihe = 4;
int Spalte = 4;
```



```
// Anzahl zufälliger LEDs
int AnzahlLED = 5;

// Variablen der Zeit
float StartZeit;
float VerstricheneZeit;
float Sekunden;

// Variable für den Neustart
bool Start = false;

/*
  VCC -> 5V
  GND
  Pin 12 -> DATA IN-pin
    Pin 11 -> CLK-pin
    Pin 10 -> CS-pin
*/
LedControl LEDMatrix = LedControl(12, 11, 10, 1);
```

Im setup-Teil wird die LED-Matrix gestartet und der pinMode des Taster und des Lautsprechers festgelegt.

```
void setup()
{
  Serial.begin(9600);
  ZeitStoppen.attach(JoystickButton);

  // Zufallsgenerator starten
  randomSeed(analogRead(5));

  // Matrix "aufwecken"
  LEDMatrix.shutdown(0, false);

  // mittlere Helligkeit setzen
  LEDMatrix.setIntensity(0, 2);
  PfeilUnten();
  pinMode(JoystickButton, INPUT_PULLUP);

  // Start-LED setzen
  LEDMatrix.setLed(0, Reihe, Spalte, true);

  // Zeit starten
  StartZeit = millis();
}
```



Erstelle Funktionen für den „Bau“ des Parcours und für die Darstellung des Pfeils:

```
void ParcoursBauen()
{
    LEDMatrix.clearDisplay(0);
    int Minimum = 0;
    int Maximum = 7;

    for (int i = 0; i < AnzahlLED; i++)
    {
        int Spalte = random(Minimum, Maximum);
        int Zeile = random(Minimum, Maximum);
        LEDMatrix.setLed(0, Spalte, Zeile, true);
    }
}

void PfeilUnten()
{
    byte Zeichen[8] =
    {
        B00010000, // 1. Reihe
        B00110000, // 2. Reihe
        B01111111, // 3. Reihe
        B11111111, // 4. Reihe
        B01111111, // 5. Reihe
        B00110000, // 6. Reihe
        B00010000, // 7. Reihe
        B00000000 // 8. Reihe
    };

    // Matrix anzeigen
    for (int i = 0; i < 8; i++)
    {
        LEDMatrix.setRow(0, i, Zeichen[i]);
    }
}
```

Der loop-Teil:

```
void loop()
{
    // Start
    if (Start)
    {
        ParcoursBauen();
        Start = false;
        StartZeit = millis();
        int Reihe = 4;
        int Spalte = 4;
        LEDMatrix.setLed(0, Reihe, Spalte, true);
    }
}
```



```
// Joystick-Button lesen
if (ZeitStoppen.update())
{
    if (ZeitStoppen.read() == LOW)
    {
        Start = true;

        // Zeit berechnen
        float Sekunden;
        VerstricheneZeit = millis() - StartZeit;
        Sekunden = VerstricheneZeit / 1000;
        String GesamtSekunden = String(Sekunden);

        // . durch , ersetzen
        GesamtSekunden.replace(".", ",");

        // Ausgabe im Seriellen Monitor
        Serial.println("Sekunden insgesamt: " + GesamtSekunden + " Sekunden");

        // Minuten berechnen
        int Minute = int(Sekunden) / 60;

        // nur Ausgabe der Minuten wenn Minute > 0
        if (Minute > 0)
        {
            // Ausgabe verschönern, wenn Minute > 1 -> Ausgabe "Minuten"
            // "Minute"
            if (Minute > 1)
            {
                Serial.print(String(Minute) + " Minuten ");
            }
            else
            {
                Serial.print(String(Minute) + " Minute ");
            }
        }
    }

    // von Sekunden Anzahl der Minuten abziehen
    Sekunden = Sekunden - Minute * 60;

    // Sekunden in String umwandeln damit . durch , ersetzt werden kann
    String AnzahlSekunden = String(Sekunden);
    // . durch , ersetzen
    AnzahlSekunden.replace(".", ",");
    Serial.println(AnzahlSekunden + " Sekunden");
}

// Bewegung der X-Achse lesen
PositionX = analogRead(XAchse);
```



```
// Bewegung X-Achse nach oben
if (PositionX > 600)
{
    // zu schnelles "Springen" verhindern
    delay(200);
    LEDMatrix.setLed(0, Reihe, Spalte, false);
    if (Spalte > 0) Spalte--;
    LEDMatrix.setLed(0, Reihe, Spalte, true);
}

// Bewegung X-Achse nach unten
if (PositionX < 300)
{
    delay(200);
    LEDMatrix.setLed(0, Reihe, Spalte, false);
    if (Spalte < 7) Spalte++;
    LEDMatrix.setLed(0, Reihe, Spalte, true);
}

// Bewegung der Y-Achse lesen
PositionY = analogRead(YAchse);

// Bewegung Y-Achse nach rechts
if (PositionY > 600)
{
    delay(200);
    LEDMatrix.setLed(0, Reihe, Spalte, false);
    if (Reihe > 0) Reihe--;
    LEDMatrix.setLed(0, Reihe, Spalte, true);
}

// Bewegung Y-Achse nach links
if (PositionY < 300)
{
    delay(200);
    LEDMatrix.setLed(0, Reihe, Spalte, false);
    if (Reihe < 7) Reihe++;
    LEDMatrix.setLed(0, Reihe, Spalte, true);
}
}
```