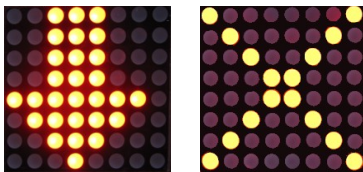
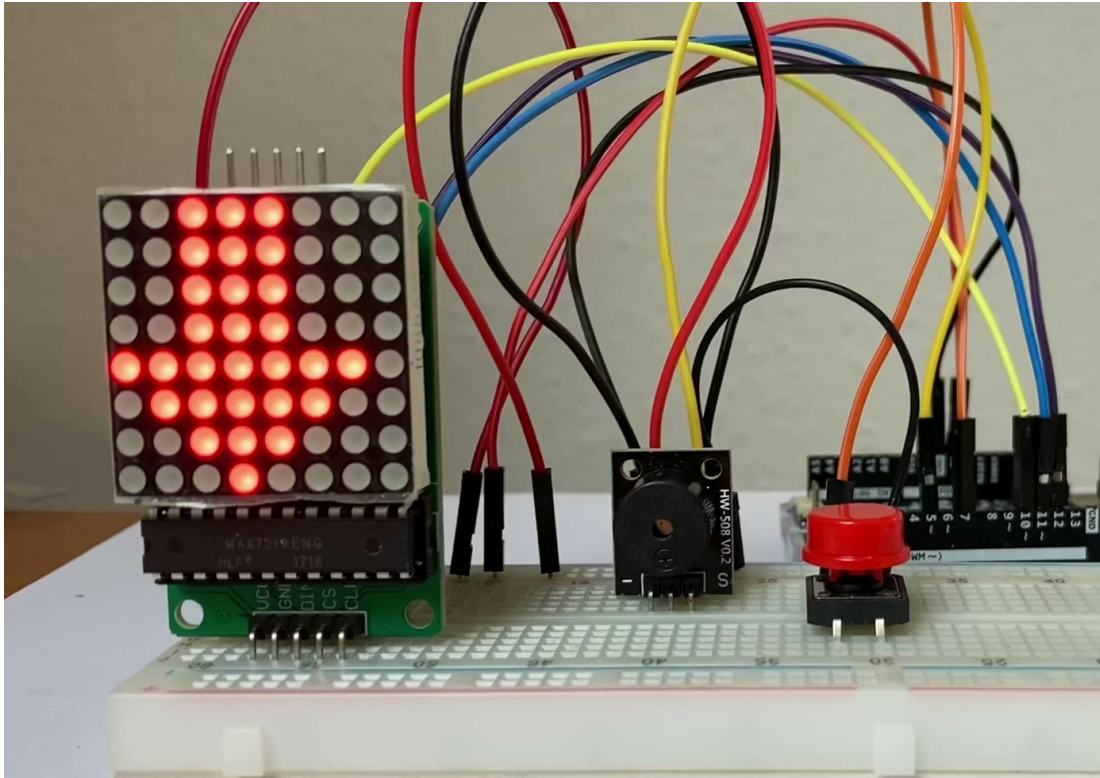


Ziel des Projekts

Die LED-Matrix soll nach einem Tasterdruck einen Countdown von 9 bis 0 Sekunden anzeigen. Zusätzlich ertönt jede Sekunde ein kurzer Ton.



Beim Start des Programms wird ein Pfeil nach unten angezeigt, am Ende ein Kreuz.

Die Hardware

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



1. Reihe

B00010000

8. Reihe

B00000000

Die Schreibweise für jede Zeile kann binär angegeben werden: 0 = aus, 1 = an.

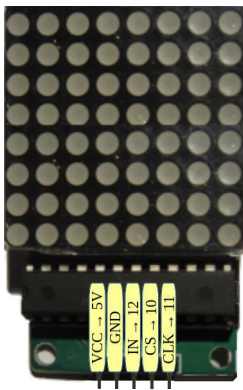
Den Werten wird ein "B" vorangestellt.

Die LED-Matrix wurde in das Steckbrett eingesetzt, daher sind die Zeilen um 90° gedreht.



```
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]);
  }
}
```

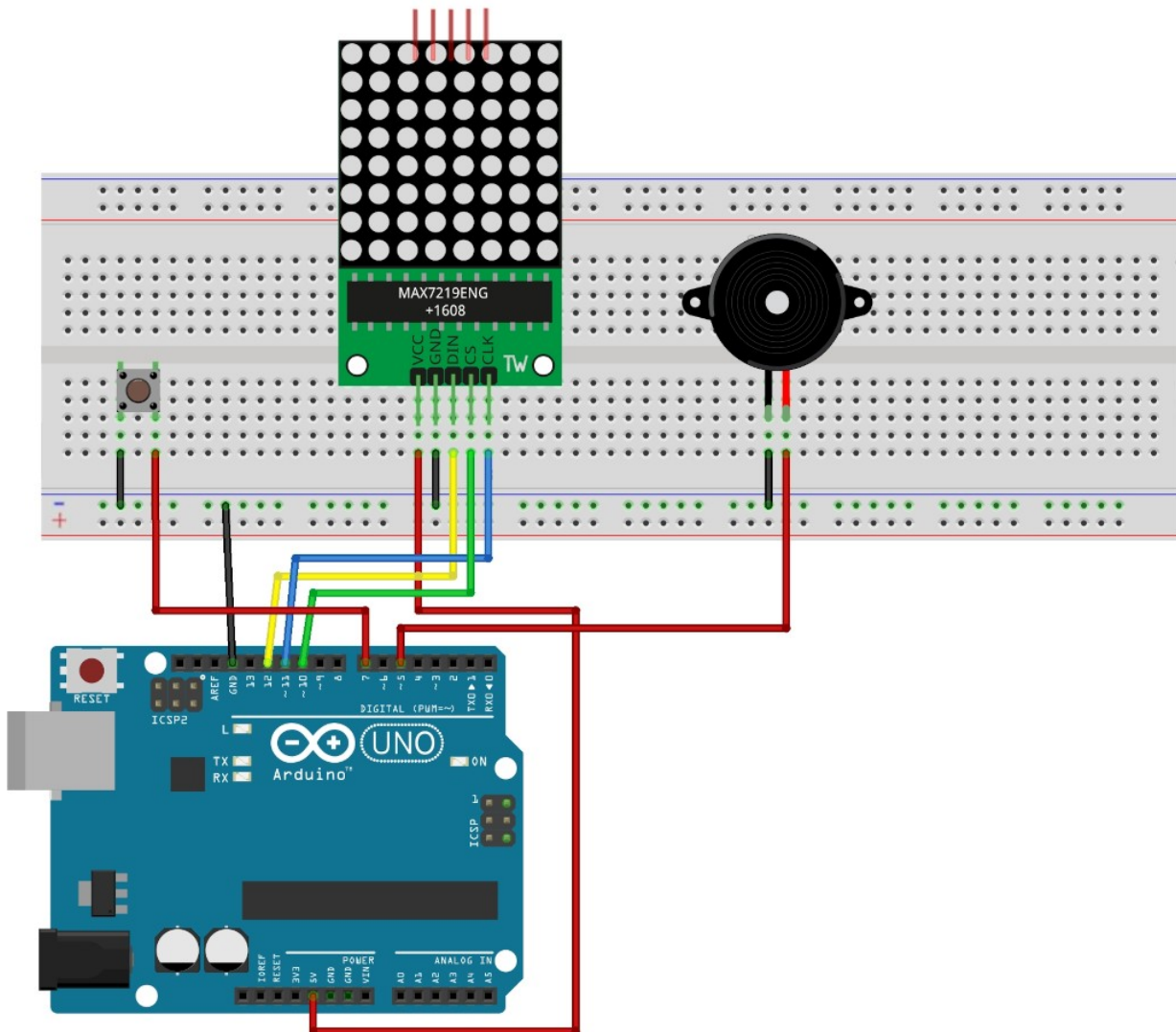


Pinbelegung der LED-Matrix.

Benötigte Bauteile

- ➔ Taster
- ➔ LED-Matrix
- ➔ Lautsprecher
- ➔ Leitungsdrähte

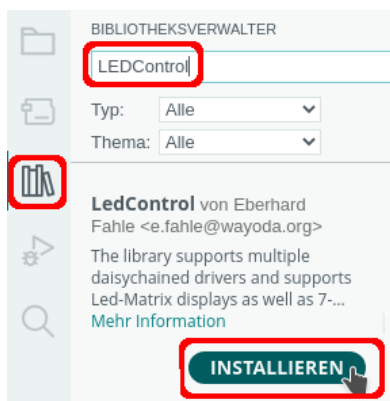
Der Schaltplan



fritzing

Benötigte Bibliothek

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

```
#include "LedControl.h"
LedControl LEDMatrix = LedControl(12, 11, 10, 1);

int TASTER = 7;
int LAUTSPRECHER = 5;

int TasterLesen;
void setup()
{
    // Matrix "aufwecken"
    LEDMatrix.shutdown(0, false);

    // mittlere Helligkeit setzen
    LEDMatrix.setIntensity(0, 8);
}
```



Im loop-Teil wird zunächst der Pfeil nach unten angezeigt und der Taster abgefragt. Anschließend wird nach jeweils einer Sekunde Pause die nächste Zahl angezeigt. Nach Ablauf der Zeit erscheint das Kreuz.

```
void loop()
{
  PfeilUnten();
  TasterLesen = digitalRead(TASTER);

  if (TasterLesen == LOW)
  {
    ZahlNeun();
    tone(LAUTSPRECHER, 1000, 100);
    delay(1000);

    ZahlAcht();
    tone(LAUTSPRECHER, 1000, 100);
    delay(1000);

    ZahlSieben();
    tone(LAUTSPRECHER, 1000, 100);
    delay(1000);

    ZahlSechs();
    tone(LAUTSPRECHER, 1000, 100);
    delay(1000);

    ZahlFuenf();
    tone(LAUTSPRECHER, 1000, 100);
    delay(1000);

    ZahlVier();
    tone(LAUTSPRECHER, 1000, 100);
    delay(1000);

    ZahlDrei();
    tone(LAUTSPRECHER, 1000, 100);
    delay(1000);

    ZahlZwei();
    tone(LAUTSPRECHER, 1000, 100);
    delay(1000);

    ZahlEins();
    tone(LAUTSPRECHER, 1000, 100);
    delay(1000);
    ZahlNull();
    delay(1000);

    Kreuz();
    tone(LAUTSPRECHER, 500, 1000);
    delay(2000);
  }
}
```



Hier sind die benötigten Funktionen für die Zahlen, den Pfeil und das Kreuz:

```
void PfeilUnten()
{
    byte Zeichen[8] =
    {
        B00010000,
        B00110000,
        B01111111,
        B11111111,
        B01111111,
        B00110000,
        B00010000,
        B00000000
    };

    for (int zaehler = 0; zaehler < 8; zaehler++)
    {
        LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);
    }
}

void ZahlEins()
{
    byte Zeichen[8] =
    {
        B00000000,
        B00000000,
        B00000100,
        B10000010,
        B11111111,
        B10000000,
        B00000000,
        B00000000
    };

    for (int zaehler = 0; zaehler < 8; zaehler++)
    {
        LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);
    }
}

void ZahlZwei()
{
    byte Zeichen[8] =
    {
        B00000000,
        B10000010,
        B11000001,
        B10100001,
        B10010001,
        B10001110,
        B00000000,
        B00000000
    };
};
```



```
for (int zaehler = 0; zaehler < 8; zaehler ++)  
{  
    LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);  
}  
}  
  
void ZahlDrei()  
{  
    byte Zeichen[8] =  
    {  
        B00000000,  
        B01000010,  
        B10000001,  
        B10001001,  
        B10001001,  
        B01110110,  
        B00000000,  
        B00000000  
    };  
  
    for (int zaehler = 0; zaehler < 8; zaehler ++)  
    {  
        LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);  
    }  
}  
  
void ZahlVier()  
{  
    byte Zeichen[8] =  
    {  
        B00000000,  
        B00110000,  
        B00101000,  
        B00100100,  
        B11110010,  
        B00100001,  
        B00000000,  
        B00000000  
    };  
  
    for (int zaehler = 0; zaehler < 8; zaehler ++)  
    {  
        LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);  
    }  
}  
  
void ZahlFuenf()  
{  
    byte Zeichen[8] =  
    {  
        B00000000,  
        B01001111,  
        B10001001,  
        B10001001,  
        B10001001,  
        B01110001,  
        B00000000,  
        B00000000  
    };  
  
    for (int zaehler = 0; zaehler < 8; zaehler ++)  
    {  
        LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);  
    }  
}
```



```
    B00000000,
    B00000000
};

for (int zaehler = 0; zaehler < 8; zaehler++)
{
    LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);
}

void ZahlSechs()
{
    byte Zeichen[8] =
    {
        B00000000,
        B01111110,
        B10001001,
        B10001001,
        B10001001,
        B01110010,
        B00000000,
        B00000000
    };

    for (int zaehler = 0; zaehler < 8; zaehler++)
    {
        LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);
    }
}

void ZahlSieben()
{
    byte Zeichen[8] =
    {
        B00000000,
        B11100001,
        B00010001,
        B00001001,
        B00000101,
        B00000011,
        B00000000,
        B00000000
    };

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




```
void ZahlAcht()
{
    byte Zeichen[8] =
    {
        B00000000,
        B01110110,
        B10001001,
        B10001001,
        B10001001,
        B01110110,
        B00000000,
        B00000000
    };

    for (int zaehler = 0; zaehler < 8; zaehler++)
    {
        LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);
    }
}

void ZahlNeun()
{
    byte Zeichen[8] =
    {
        B00000000,
        B01001110,
        B10010001,
        B10010001,
        B10010001,
        B01111110,
        B00000000,
        B00000000
    };

    for (int zaehler = 0; zaehler < 8; zaehler++)
    {
        LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);
    }
}

void ZahlNull()
{
    byte Zeichen[8] =
    {
        B00000000,
        B01111110,
        B10000001,
        B10000001,
        B10000001,
        B01111110,
        B00000000,
        B00000000
    };
};
```



```
for (int zaehler = 0; zaehler < 8; zaehler ++)  
{  
    LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);  
}  
  
void Kreuz()  
{  
    const PROGMEM byte Zeichen[8] =  
    {  
        B10000001,  
        B01000010,  
        B00100100,  
        B00011000,  
        B00011000,  
        B00100100,  
        B01000010,  
        B10000001  
    };  
  
    for (int zaehler = 0; zaehler < 8; zaehler ++)  
    {  
        LEDMatrix.setRow(0, zaehler, Zeichen[zaehler]);  
    }  
}
```

Hartmut Waller letzte Änderung: 24.11.25