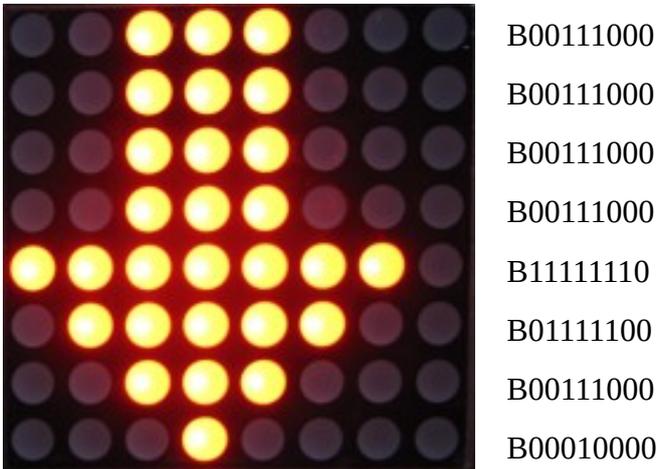


Die hier verwendete LED-Matrix mit der Bezeichnung Max7219 besteht aus 8x8 LEDs.
 Die einzelnen LEDs werden in Zeilen und Spalten angesprochen.
 Die LED-Matrix wurde in das Steckbrett eingesetzt, daher verlaufen die Reihen vertikal und die Spalten horizontal.

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

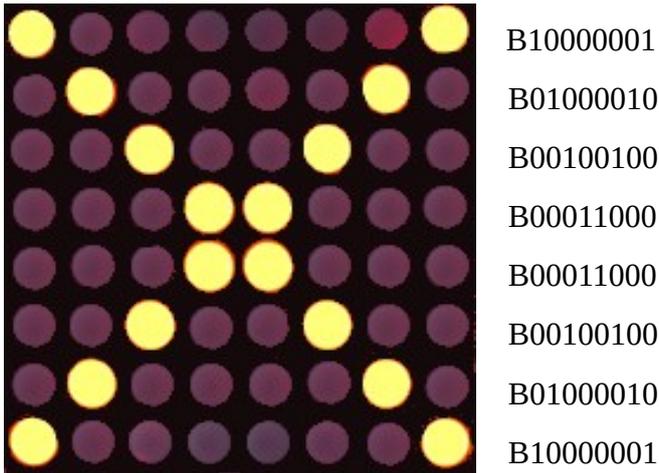
Beispiele:



Der Programmschnipsel zeigt den Pfeil an:

```
void PfeilUnten()
{
  byte Zeichen[8] =
  {
    B00111000,
    B00111000,
    B00111000,
    B00111000,
    B11111110,
    B01111100,
    B00111000,
    B00010000
  };

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

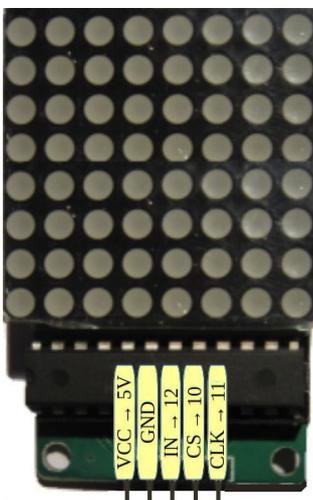


Der Programmschnipsel zeigt das Kreuz an:

```
byte Zeichen[8] =
{
  B10000001, // 1. Reihe
  B01000010, // 2. Reihe
  B00100100, // 3. Reihe
  B00011000, // 4. Reihe
  B00011000, // 5. Reihe
  B00100100, // 6. Reihe
  B01000010, // 7. Reihe
  B10000001 // 8. Reihe
};

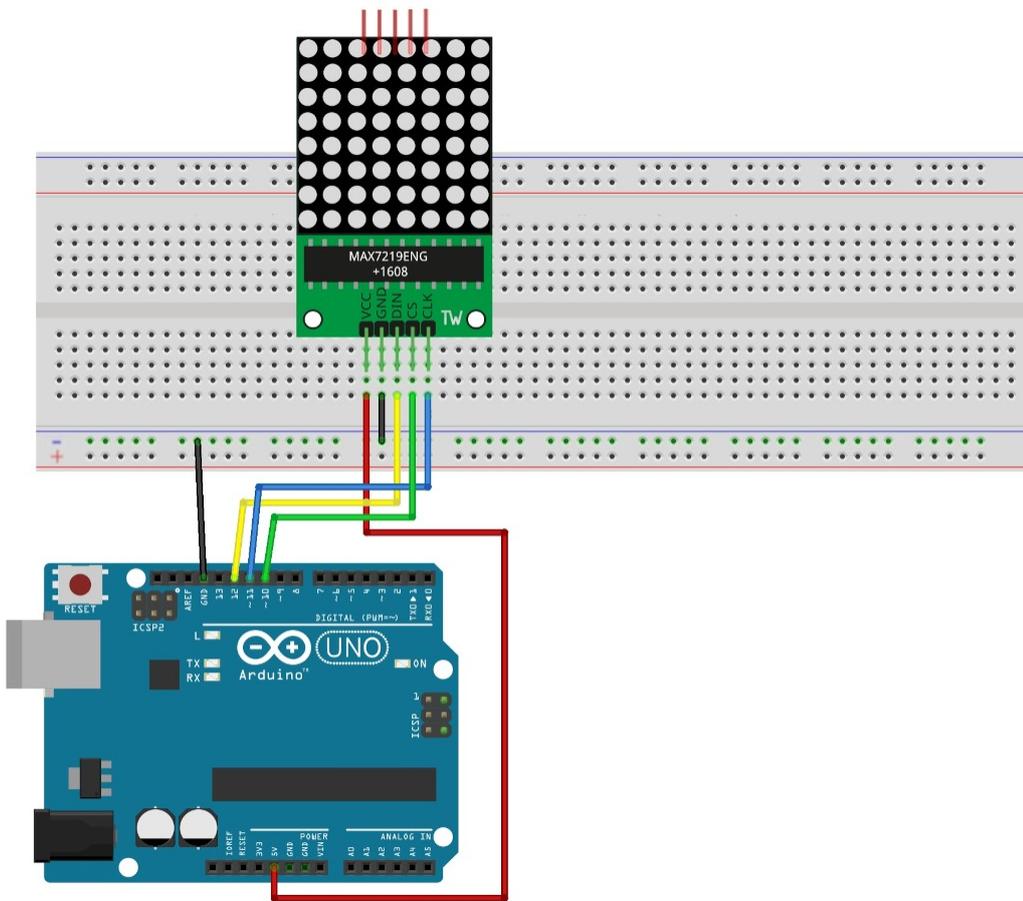
// Matrix anzeigen
for (int i = 0; i < 8; i ++)
{
  LEDMatrix.setRow(0, i, Zeichen[i]);
}
```

SchlieÙe die LED-Matrix an.



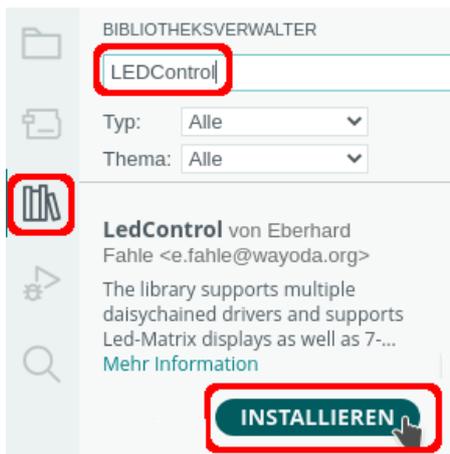
Pinbelegung der LED-Matrix.

Baue die Schaltung auf:



fritzing

Als Erstes musst du eine Bibliothek installieren:



Im Kopf des Programms musst du die Bibliothek einbinden und die Pinbelegung der LED-Matrix festlegen:

```
# include <LedControl.h>
/*
  VCC → 5V
  GND
  Pin 12 → DATA IN Pin
  Pin 10 → CLK Pin
  Pin 11 → CS Pin
  1 → eine Matrix angeschlossen
*/
LedControl LEDMatrix = LedControl(12, 11, 10, 1);
```



Jedem Befehl muss der Name der LED-Matrix vorangestellt werden.

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

Im Beispiel wird ein Lauflicht erzeugt, das die Spalten (Columns) von unten nach oben und dann die Zeilen (Rows) von links nach rechts durchläuft.

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



Verändere das Beispiel so, dass in jeder Zeile die LEDs nacheinander leuchten.



```
// Bibliothek LedControl einbinden
# include <LedControl.h>

/*
  Pinbelegung
  VCC → 5V
  GND
  Pin 12 → DATA IN Pin
  Pin 11 → CLK Pin
  Pin 10 → CS Pin
*/
LedControl LEDMatrix = LedControl(12, 11, 10, 1);

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

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

void loop()
{
  /*
    setRow/setColumn erwartet drei Parameter:
    - die Adresse der LED-Matrix → 0
    - die Angabe der Zeile
    - den Wert der LED 1 = an, 0 = aus
  */

  // spaltenweise
  for (int i = 0; i < 7; i++)
  {
    // alle an
    LEDMatrix.setColumn(0, i, B11111111);
    delay(100);
    // alle aus
    LEDMatrix.setColumn(0, i, B00000000);
    delay(100);
  }

  // und zurück
  for (int i = 7; i >= 0; i--)
  {
    // alle an
    LEDMatrix.setColumn(0, i, B11111111);
    delay(100);
    // alle aus
    LEDMatrix.setColumn(0, i, B00000000);
    delay(100);
  }
}
```

```
// zeilenweise
for (int i = 0; i < 7; i ++)
{
  // alle an
  LEDMatrix.setRow(0, i, B11111111);
  delay(100);

  // alle aus
  LEDMatrix.setRow(0, i, B00000000);
  delay(100);
}

// und zurück
for (int i = 7; i >= 0; i--)
{
  // alle an
  LEDMatrix.setRow(0, i, B11111111);
  delay(100);
  // alle aus
  LEDMatrix.setRow(0, i, B00000000);
  delay(100);
}
}
```

Hartmut Waller (hartmut-waller.info/arduino-blog) Letzte Änderung: 10.05.24