

## LED-Matrix - Würfeln

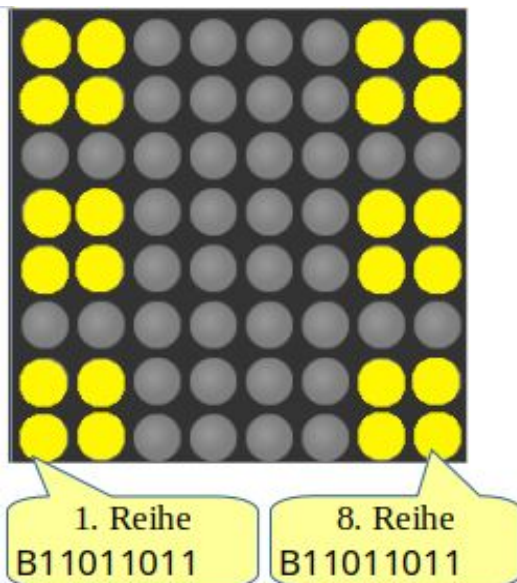
Die hier verwendete LED-Matrix mit der Bezeichnung Max7219 besteht aus 8x8 LEDs.

Wenn der Taster gedrückt wird, würfelt der Arduino eine Zahl zwischen 1 und 6. Bevor das Ergebnis angezeigt wird, zeigt die LED-Matrix in schneller Folge zufällig ausgewählte Zahlen.

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.**

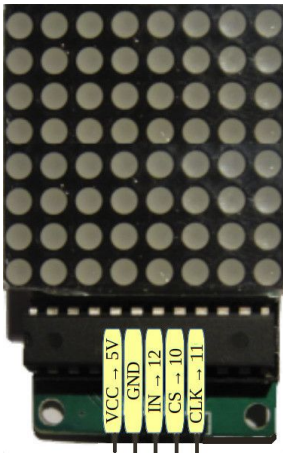
Die Schreibweise für jede Zeile kann binär angegeben werden: 0 = aus, 1 = an.



Der Programmschnipsel zeigt die 6 an:

```
void Sechs()
{
  byte Zeichen[8] =
  {
    B11000011, // 1. Reihe
    B11000011, // 2. Reihe
    B00000000, // 3. Reihe
    B11000011, // 4. Reihe
    B11000011, // 5. Reihe
    B00000000, // 6. Reihe
    B11000011, // 7. Reihe
    B11000011 // 8. Reihe
  };

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

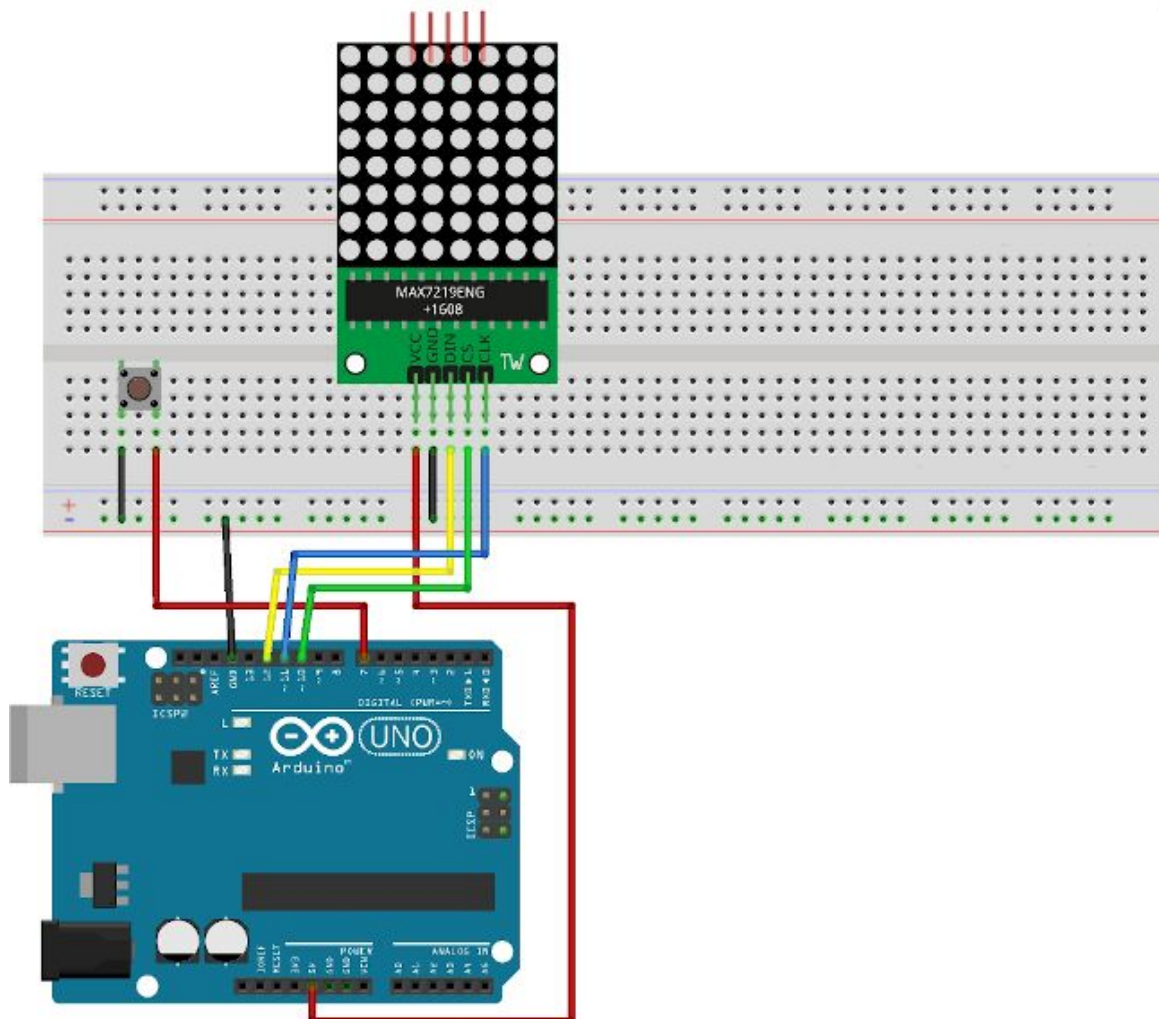


Pinbelegung der LED-Matrix

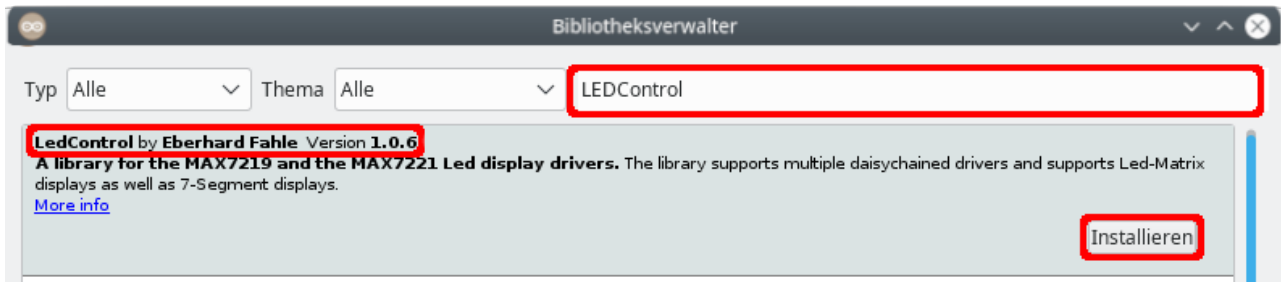
**Benötigte Bauteile:**

- ➔ Taster
- ➔ LED-Matrix
- ➔ Leitungsdrähte

Baue die Schaltung auf.



Als Erstes musst du eine Bibliothek installieren:



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

Methoden der Bibliothek LEDControl:

Methoden	Funktion/Beispiel
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

Binde die benötigte Bibliothek ein und definiere die Variablen:

```
# include <LedControl.h>

int TASTER = 7;

int TasterLesen;
int Zahl;
```

```
int Minimum = 1;
int Maximum = 7;

LedControl LEDMatrix = LedControl(12, 11, 10, 1);
```

Starte im setup-Teil die LED-Matrix und lege den pinMode für den Taster fest:

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

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

  // Zufallsgenerator starten
  randomSeed(A0);
  pinMode(TASTER, INPUT_PULLUP);
}
```

Verwende für die Ermittlung der Zufallszahl eine Funktion.

```
int ZufallsZahl()
{
  Zahl = random(Minimum, Maximum);
  return Zahl;
}
```

Die Abfrage der Zufallszahl kannst du mit switch ... case erledigen.

```
void loop()
{
  TasterLesen = digitalRead(TASTER);
  if (TasterLesen == LOW)
  {
    delay(300);

    // Funktion Zufallszahl aufrufen
    Zahl = ZufallsZahl();

    switch (Zahl)
    {
      case 1:
        Eins();
        break;

      case 2:
        Zwei();
        break;
    }
  }
}
```

```
case 3:
  Drei();
  break;

case 4:
  . . .

case 5:
  . . .

case 6:
  . . .
}
}
}
```



Verwende für die Anzeige der Würfelaugen die folgenden Methoden:

