

Teil 4

Lektion

1

Projektkurs: Würfel

Überblick:

In diesem Tutorial verwenden wir das Max7219-Modul und den Kipp-Kugel-Schalter, um einen elektronischen Würfel zu erstellen. Nebenbei wird erläutert, wie man die Arduino-Interrupt-Funktion verwendet.

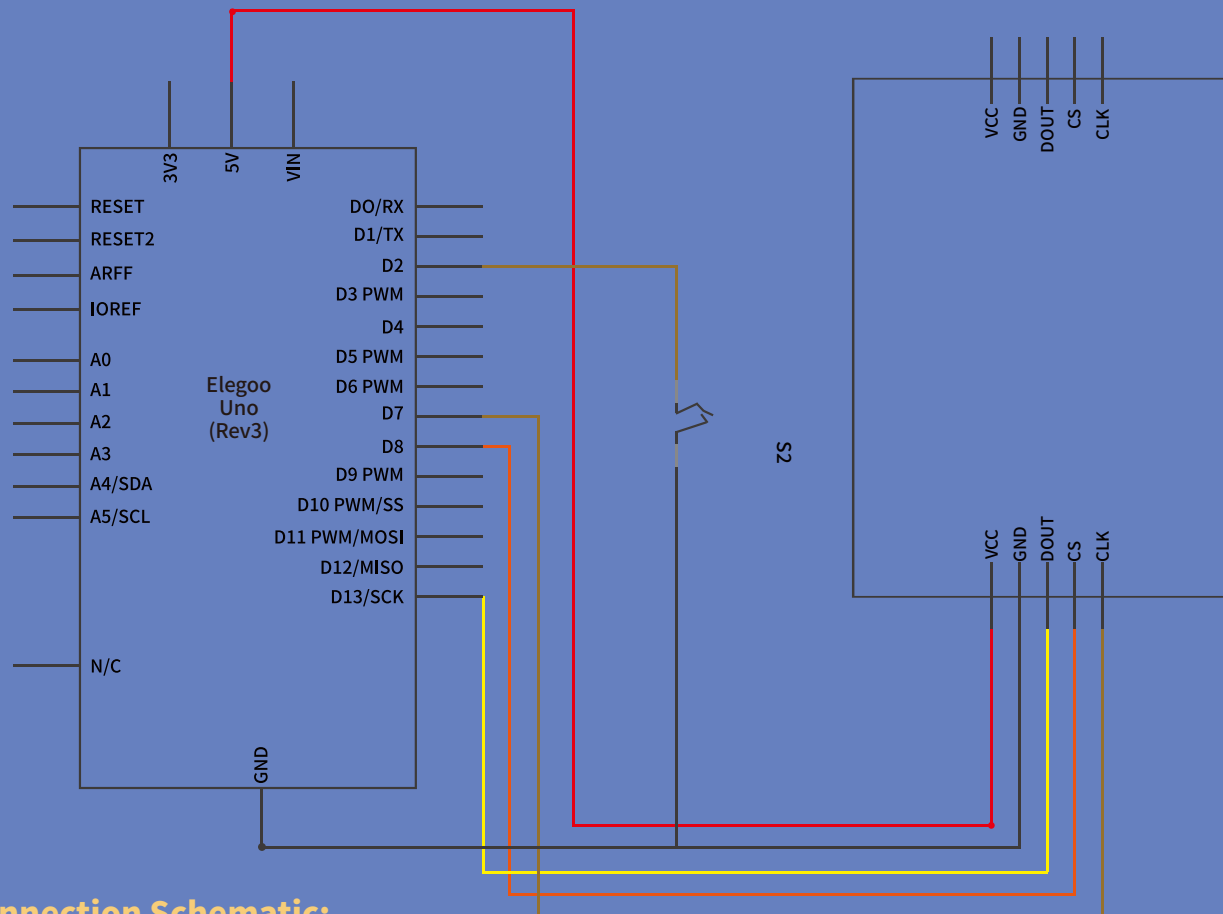
Benötigte Komponenten:

- (1) X Elegoo UNO R3
- (1) X ALL IN ONE Sensor Shield
- (1) X MAX7219 module
- (1) X Kipp-Kugel-Schalter



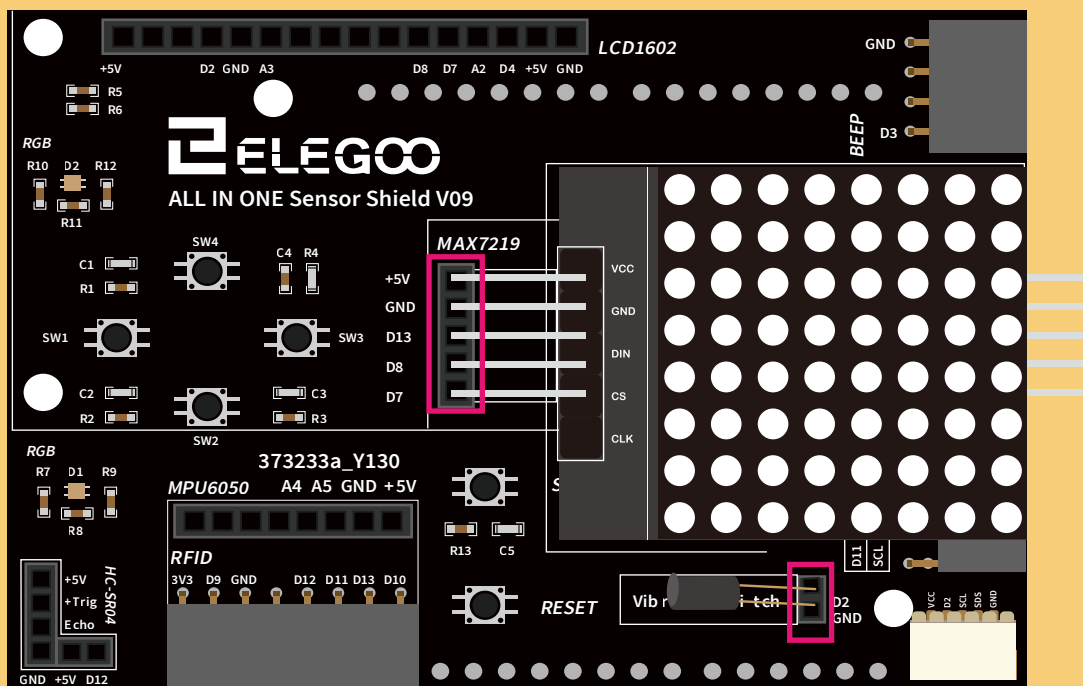
Allgemeine Beschreibung:

Beim Schütteln des Boards wechselt der Kipp-Kugel-Schalter in den Ein- und Ausschalt-Zustand. Diese Änderung löst eine Unterbrechung im Programm aus, wodurch das Max7219-Modul die Punktnummern ändert und zufällig stoppt.



Connection Schematic:

Tips: Bitte fügen Sie das erweiterte Board in die UNO ein.



Verbindungs-Diagramm:

Code:

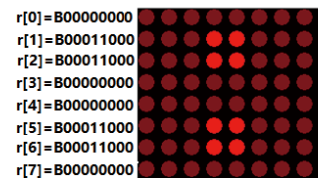
Öffnen Sie nach der Verkabelung der Komponenten das Programm im Code-Ordner “Dice” und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Hochladen von Programmen finden Sie in Lektion 5, sollten Fehler auftreten.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die <LedControl> -Bibliothek installiert haben oder installieren Sie diese gegebenenfalls neu. Andernfalls funktioniert Ihr Code nicht. Einzelheiten zum Laden einer Bibliotheksdatei finden Sie ebenfalls in Lektion 5.

Part1.

Max7219 Display

1. Stellen Sie die anzuzeigende Punktnummer ein.
Erstellen Sie zunächst ein Array und schreiben Sie die Grafikdaten der Punkte in das Array. Hier ist ein Beispiel für Anzeigepunkt 2 auf dem



```
byte r[8]={B00000000,B00011000,B00011000,B00000000,B00000000,B00011000,B00011000,  
B00000000};
```

Verwenden Sie dann `lc.setRow()`, um die Punkte auf dem Max7219-Modul anzuzeigen.

```
lc.setRow(0,0,r[0]);  
lc.setRow(0,1,r[1]);  
lc.setRow(0,2,r[2]);  
lc.setRow(0,3,r[3]);  
lc.setRow(0,4,r[4]);  
lc.setRow(0,5,r[5]);  
lc.setRow(0,6,r[6]);  
lc.setRow(0,7,r[7]);
```

Aus dem gleichen Grund haben wir verschiedene Arrays eingerichtet, um verschiedene Punkte anzuzeigen.

Part2

Random Variation of Points

1.1. Wir verwenden die Funktion `random()`, um die Punktzahl zufällig zu ändern.

`random(min, max)`

min: lower bound of the random value, inclusive (optional).

max: upper bound of the random value, exclusive.

```
randNumber = random(1, 7);
```

`random(1, 7)` can return us a number from 1-6.

2. Durch die im ersten Schritt erhaltenen Zahlen, können wir die Schalterfunktion verwenden, um die entsprechenden Punkte anzuzeigen.

```
Switch (randNumber)
{
    case 1:
        lc.setRow(0,0,a[0]);
        lc.setRow(0,1,a[1]);
        lc.setRow(0,2,a[2]);
        lc.setRow(0,3,a[3]);
        lc.setRow(0,4,a[4]);
        lc.setRow(0,5,a[5]);
        lc.setRow(0,6,a[6]);
        lc.setRow(0,7,a[7]);
        break;
    case 2:
        lc.setRow(0,0,r[0]);
        lc.setRow(0,1,r[1]);
        lc.setRow(0,2,r[2]);
        lc.setRow(0,3,r[3]);
        lc.setRow(0,4,r[4]);
        lc.setRow(0,5,r[5]);
        lc.setRow(0,6,r[6]);
        lc.setRow(0,7,r[7]);
        break;
    case 3: .....
}
```

Part 3

Punkte ändern sich nicht mehr und beginnen sich zu ändern

1. Punkte ändern sich nicht mehr

```
void loop() {  
  // put your main code here, to run repeatedly:  
  writeRandomNumberOnMatrix();  
  delay(delaytime1);  
  delaytime1=delaytime1+20;//The time for digital changes becomes larger  
  /*delaytime1>300 Digital does not change*/  
  while(delaytime1>300){  
    writeRandomNumberOnMatrix();  
    delay(delaytime1);  
    Serial.println(delaytime1);  
  }  
  randomNumber = random(1, 7);  
}
```

In der Schleife ändert sich der Punkt jedes Mal, wenn die Schleife ausgeführt wird einmal und die Verzögerungszeit wird um 20 milli Sekunden erhöht. Wenn die Verzögerungszeit größer als 300 milli Sekunden ist, so tritt das Programm in die while-Schleife ein und der Punkt ändert sich nicht mehr.

2. Punkte beginnen sich zu ändern

- Verwenden von Interrupt-Funktionen

```
attachInterrupt(digitalPinToInterrupt(pin), ISR, mode) (recommended)  
SR: the ISR to call when the interrupt occurs; this function must take no parameters and  
return nothing. This function is sometimes referred to as an interrupt service routine.  
mode: defines when the interrupt should be triggered. Four constants are predefined as  
valid values:  
· LOW to trigger the interrupt whenever the pin is low,  
· CHANGE to trigger the interrupt whenever the pin changes value  
· RISING to trigger when the pin goes from low to high,  
· FALLING for when the pin goes from high to low.
```

Beim Schütteln des Boards löst der Kipp-Kugel-Schalter einen Interrupt aus, welcher eine Verzögerungszeit = 100 verursacht. Das Programm springt also aus der while-Schleife und ändert die Nummer erneut.

```
attachInterrupt(digitalPinToInterrupt(Ball_Switch),blink,RISING);  
void blink(){  
  delaytime1=100;  
  Serial.println(delaytime1);  
}
```