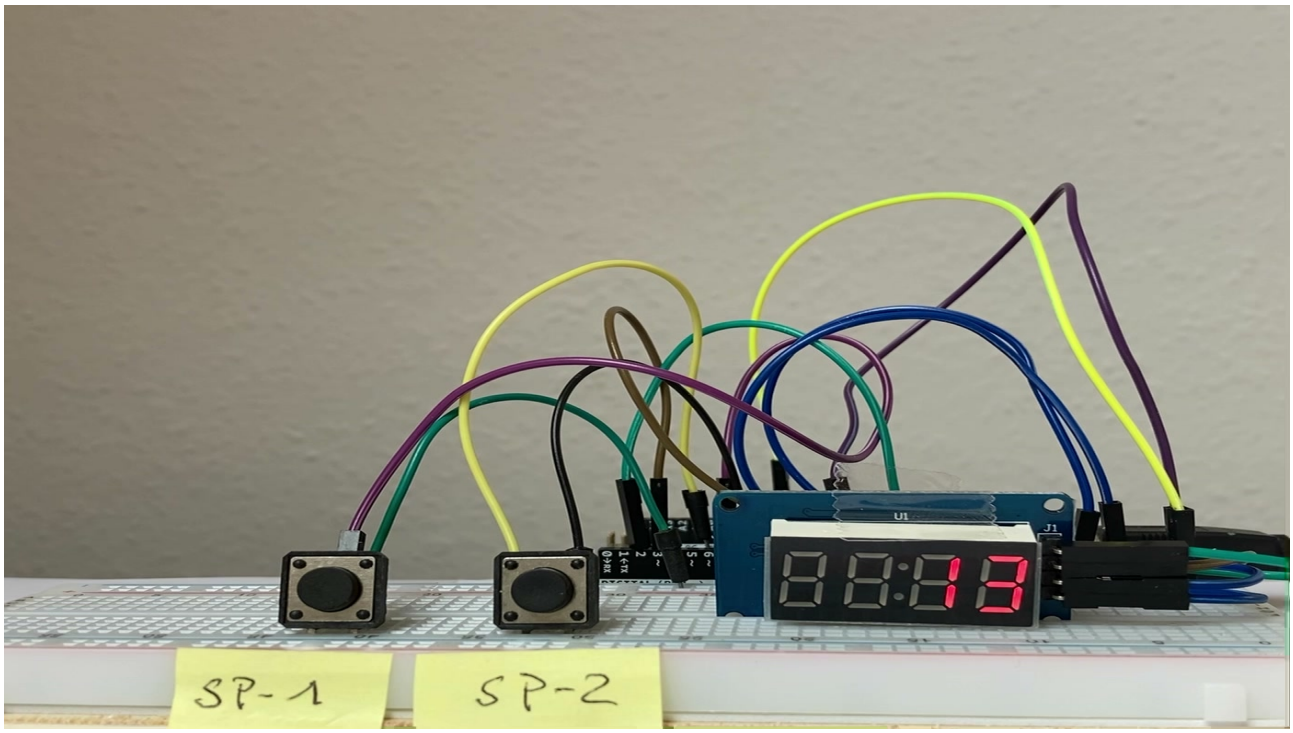
Würfelspiel für zwei Spieler mit einer 7-Segment-Anzeige

Auf dem 7-Segment-Display soll ein Würfelspiel für zwei Spieler*innen realisiert werden. Nach einem Druck auf die Taster wird eine Zahl zwischen 1 und 6 gewürfelt. Auf dem Display wird die Reihenfolge der Spieler*innen angezeigt, die gewürfelte Zahl und die Summe der bisherigen Würfe. Zum Schluss wird die siegreiche Spielerin/der siegreiche Spieler angezeigt.

So sieht die Schaltung aus:



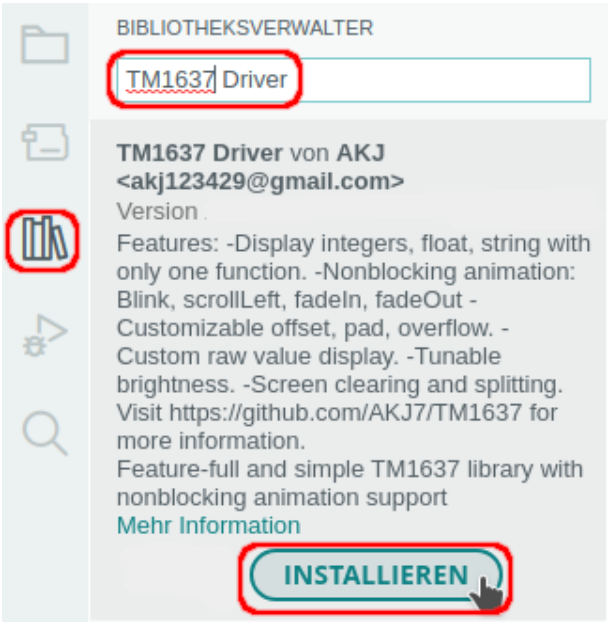
So sieht es aus:

erste Spielerin erster Spieler	Punktestand	zweite Spielerin zweiter Spieler	Punktestand	Anzeige der Siegerin/des Siegers		Summe
SP-1	06:13	SP-2	15:24	SIEG	SP-2	24

Die 7-Segment-Anzeige besteht aus 4 Zeichen. Jedes dieser Zeichen besteht aus 7 sogenannten Segmenten, die einzeln angesteuert werden können. Die Anschlüsse:

CLK → beliebiger digitaler Pin
DIO → beliebiger digitaler Pin
VCC → 5 V
GND → Ground

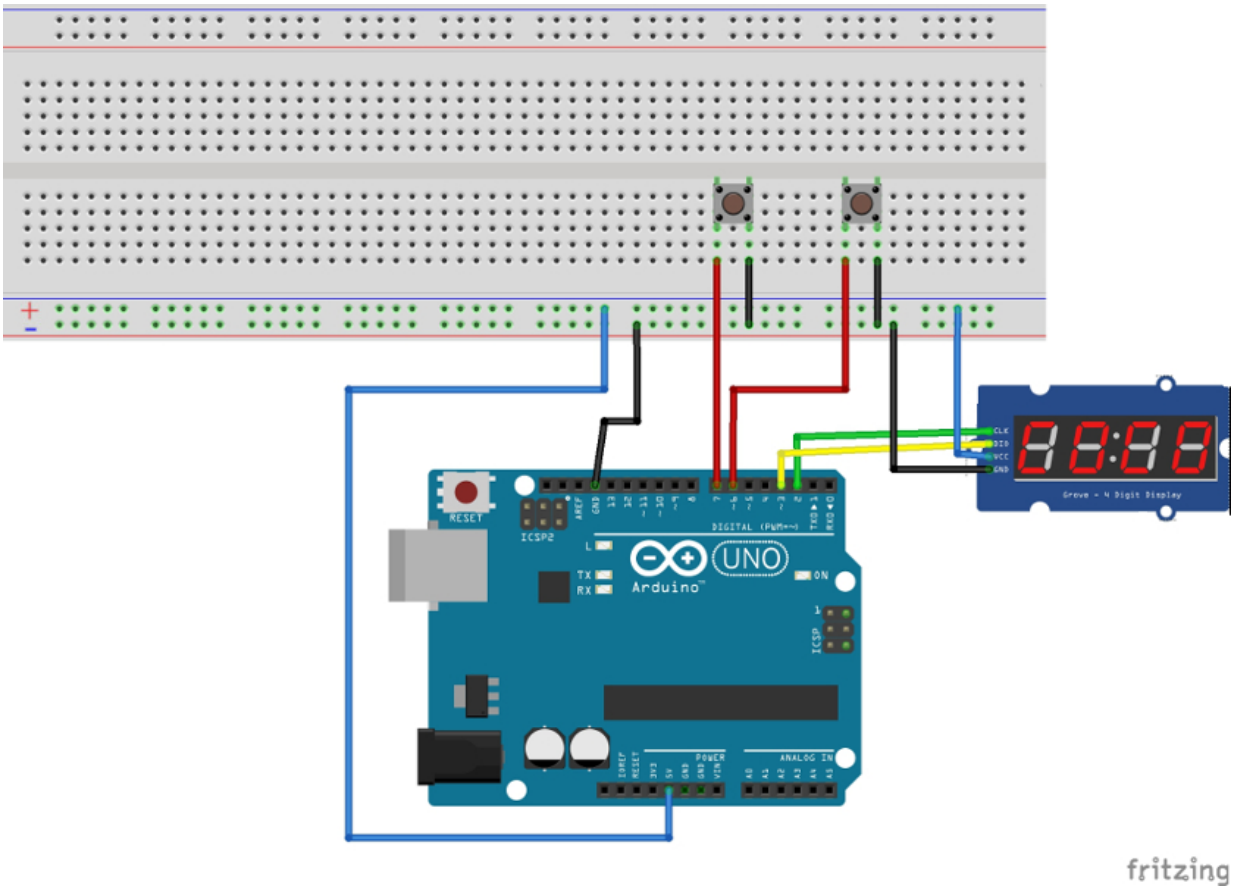
Als Erstes musst du die Bibliothek TM1637 Driver installieren



Benötigte Bauteile:

- 2 Taster
- 7-Segment-Anzeige
- Leitungsdrähte

Baue die Schaltung auf:



Beispiel: Auf Tasterdruck startet ein Countdown von 10 bis 0. Am Ende zeigt das Display „StoP“, nach einer kurzen Wartezeit erscheint „LOS“.

```
# include <TM1637.h>

TM1637 Anzeige(2, 3);

int TASTER = 6;

void setup()
{
  // Modul starten
  Anzeige.begin();

  // Helligkeit setzen
  Anzeige.setBrightness(3);

  pinMode(TASTER, INPUT_PULLUP);
  Anzeige.clearScreen();
  Anzeige.display("LOS");
  delay(1000);
}

void loop()
{
  int TasterStatus = digitalRead(TASTER);
  if (TasterStatus == LOW)
  {
    for (int i = 10; i > 0; i--)
    {
      // 10 -> Position 2, keine Nullen (rechts/links)auffüllen
      if (i == 10) Anzeige.display(i, false, false, 2);
      else Anzeige.display(i, false, false, 3);

      delay(1000);
    }

    Anzeige.display("Stop");
  }
}
```

Methoden der Bibliothek TM1637 (AKJ)

Befehl	**Parameter**
begin()	Modul starten
setBrightness(int)	0-15 → Helligkeit des Displays setzen
clearScreen()	Anzeige löschen

display(String)
display(int)
display(String/int, false, false, Position)
	display"Text");
	display(10);
	diaplay(1, false, false, 1);
	Position beginnt mit 0

colonOn()	Doppelpunkt einschalten
colonOff()	Doppelpunkt ausschalten

Binde die benötigten Bibliotheken ein und definiere die Variablen. Beachte die Kommentare.

```
// benötigte Bibliotheken
# include <TM1637.h>
# include <Bounce2.h>

// Spieler*in 1
int TASTER1 = 7;

// Spieler*in 2
int TASTER2 = 6;

// Definition Display
TM1637 Anzeige(3, 4);

// Summe der Augen der beiden Spieler*innen
int SummeSpieler1;
int SummeSpieler2;

// Anzahl der Runden, Start mit 1
int Runde = 1;

int Zahl;

// Reihenfolge der Spieler*innen festlegen, Spieler1 startet
bool StartSpieler1 = true;
bool StartSpieler2 = false;

// formatierter String für die Anzeige der Punkte
String Ergebnis;

// Bibliothek Bounce2
// "Prellverhinderer" für die Tasten starten
Bounce SP1 = Bounce();
Bounce SP2 = Bounce();
```

Der setup-Teil

```
void setup()
{
  pinMode(TASTER1, INPUT_PULLUP);
  pinMode(TASTER2, INPUT_PULLUP);

  // Instanzen des Objekts Bounce für jede Taste zuordnen
  // Zeitintervall einstellen
  SP1.attach(TASTER1);
  SP1.interval(20);
  SP2.attach(TASTER2);
  SP2.interval(20);

  // Modul starten
  Anzeige.begin();
  Anzeige.clearScreen();

  // Helligkeit setzen(0-15)
  Anzeige.setBrightness(2);
  Anzeige.display("SP-1");

  // Zufallsgenerator starten
  randomSeed(analogRead(0));
}
```

Der loop-Teil. Beachte die Kommentare.

```
vvoid loop()
{
  // Minimum und Maximum der Würfelaugen
  int Minimum = 1;
  int Maximum = 7;

  // 6 Runden spielen
  while (Runde < 4)
  {
    // Spieler*in 1 ist an der Reihe StartSpieler1 -> true
    if (StartSpieler1)
    {
      // Taste 1 gedrückt
      if (SP1.update())
      {
        if (SP1.read() == LOW)
        {
          // Wechsel → Spieler2 ist an der Reihe
          StartSpieler1 = !StartSpieler1;
          StartSpieler2 = !StartSpieler2;
```

```
        Anzeige.clearScreen();

        // Aufruf der Funktion ZufallsZahl
        SummeSpieler1 = SummeSpieler1 + ZufallsZahl(Minimum, Maximum);

        // : einschalten
        Anzeige.colonOn();

        /*
          Ergebnis formatieren:
          beide Zahlen < 10 -> führende 0 setzen
          eine der Zahlen < 10 -> führende 0 setzen
        */
        if (SummeSpieler1 < 10 && SummeSpieler2 < 10) Ergebnis = "0" +
        String(SummeSpieler1) + "0" + String(SummeSpieler2);
        else if (SummeSpieler1 >= 10 && SummeSpieler2 >= 10) Ergebnis =
        String(SummeSpieler1) + String(SummeSpieler2);
        else if (SummeSpieler1 < 10) Ergebnis = "0" + String(SummeSpieler1) +
        String(SummeSpieler2);
        else if (SummeSpieler2 < 10 ) Ergebnis = String(SummeSpieler1) + "0" +
        String(SummeSpieler2);
        Anzeige.display(Ergebnis);

        delay(2000);

        // : ausschalten
        Anzeige.colonOff();
        Anzeige.display("SP-2");
      }
    }
  }

  // Spieler*in 2 ist an der Reihe StartSpieler2 → true
  if (StartSpieler2)
  {
    // Taste 1 gedrückt
    if (SP2.update())
    {
      if (SP2.read() == LOW)
      {
        // Wechsel → Spieler1 ist an der Reihe
        StartSpieler1 = !StartSpieler1;
        StartSpieler2 = !StartSpieler2;

        Anzeige.clearScreen();

        // Aufruf der Funktion ZufallsZahl
        SummeSpieler2 = SummeSpieler2 + ZufallsZahl(Minimum, Maximum);

        Anzeige.colonOn();

        if (SummeSpieler1 < 10 && SummeSpieler2 < 10) Ergebnis = "0" +
        String(SummeSpieler1) + "0" + String(SummeSpieler2);
        else if (SummeSpieler1 >= 10 && SummeSpieler2 >= 10) Ergebnis =
        String(SummeSpieler1) + String(SummeSpieler2);
```

```
          else if (SummeSpieler1 < 10) Ergebnis = "0" + String(SummeSpieler1) +
          String(SummeSpieler2);
          else if (SummeSpieler2 < 10 ) Ergebnis = String(SummeSpieler1) + "0" +
          String(SummeSpieler2);
          Anzeige.display(Ergebnis);

          delay(2000);
          Anzeige.colonOff();
          Anzeige.clearScreen();

          Anzeige.display("SP-1");

          // nur bei Spieler2 Runde hochzählen, Spieler1 hat angefangen
          Runde ++;
        }
      }
    }
}

// unentschieden
if (SummeSpieler1 == SummeSpieler2)
{
  Anzeige.clearScreen();
  Anzeige.display(" = =");
  delay(2000);

  // alle Werte zurücksetzen
  Runde = 1;
  SummeSpieler1 = 0;
  SummeSpieler2 = 0;
  Anzeige.clearScreen();
  Anzeige.display("SP-1");
}

// Sieger Spieler1
if (SummeSpieler1 > SummeSpieler2)
{
  Anzeige.display("SIEG");
  delay(2000);
  Anzeige.clearScreen();

  Anzeige.display("SP-1");
  delay(2000);

  // Punktestand anzeigen
  Anzeige.clearScreen();

  // Anzeige rechtsbündig formatieren
  if (SummeSpieler1 < 10) Anzeige.display(SummeSpieler1, false, false, 3);
  else  Anzeige.display(SummeSpieler1, false, false, 2);
  delay(2000);

  // Neustart: alle Werte zurücksetzen, SP-1 anzeigen
  Runde = 1;
  SummeSpieler1 = 0;
```

```
    SummeSpieler2 = 0;
    Anzeige.clearScreen();
    Anzeige.display("SP-1");
  }

  // Sieger Spieler2
  if (SummeSpieler1 < SummeSpieler2)
  {
    Anzeige.display("SIEG");
    delay(2000);
    Anzeige.clearScreen();
    Anzeige.display("SP-2");
    delay(2000);
    Anzeige.clearScreen();

    if (SummeSpieler2 < 10) Anzeige.display(SummeSpieler2, false, false, 3);
    else  Anzeige.display(SummeSpieler2, false, false, 2);
    delay(2000);

    // Neustart: alle Werte zurücksetzen, SP-1 anzeigen
    Runde = 1;
    SummeSpieler1 = 0;
    SummeSpieler2 = 0;
    Anzeige.clearScreen();
    Anzeige.display("SP-1");
  }
}
```

Jetzt fehlt nur noch die Funktion ZufallsZahl:

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

Letzte Änderung: 28.08.22