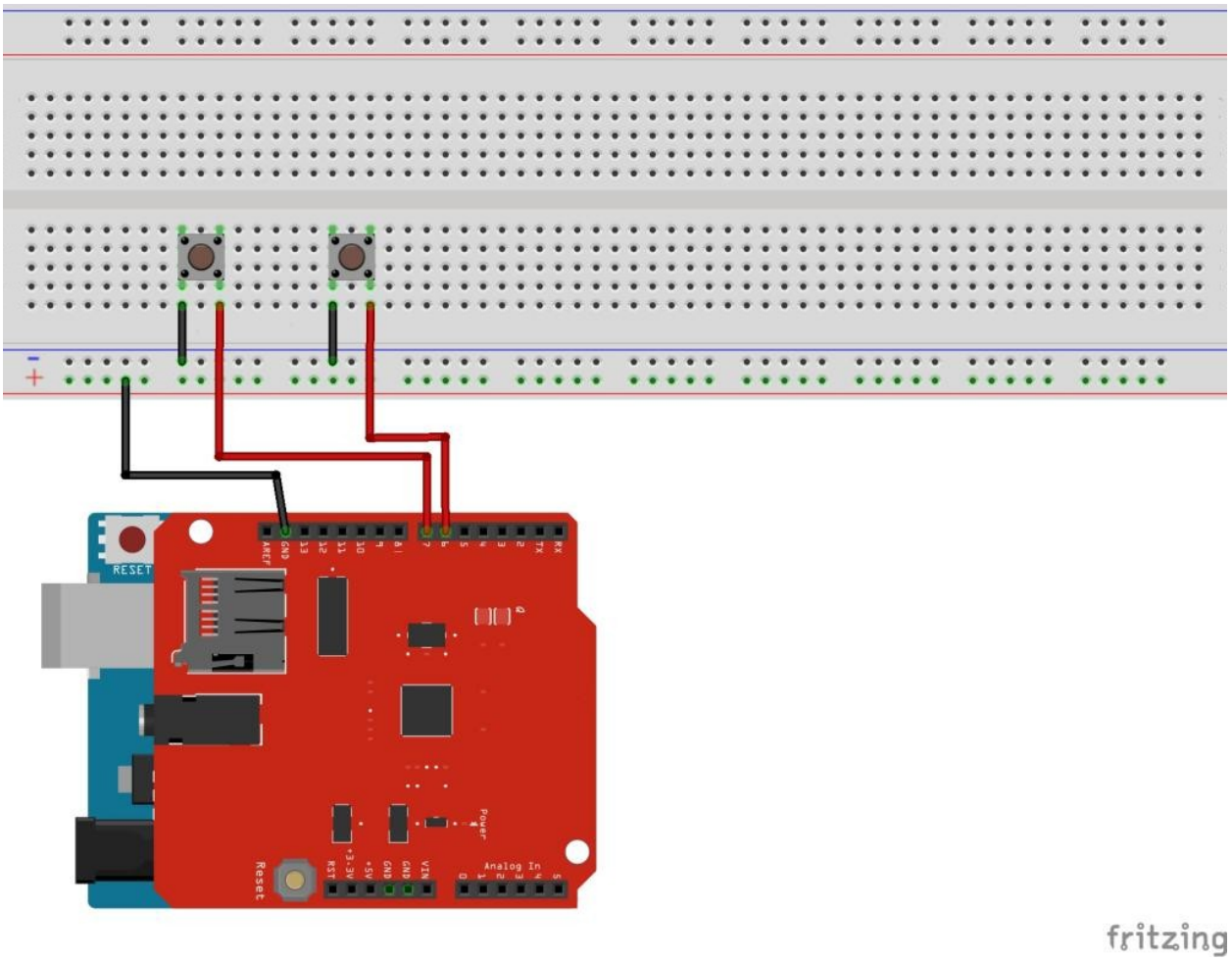
Bei einem Würfelspiel für zwei Spieler soll die gewürfelte Zahl, die Gesamtsumme und nach sechs Runden das Ergebnis des Spiels (Sieg oder Unentschieden) als Sprache ausgegeben werden.

Benötigte Bauteile:

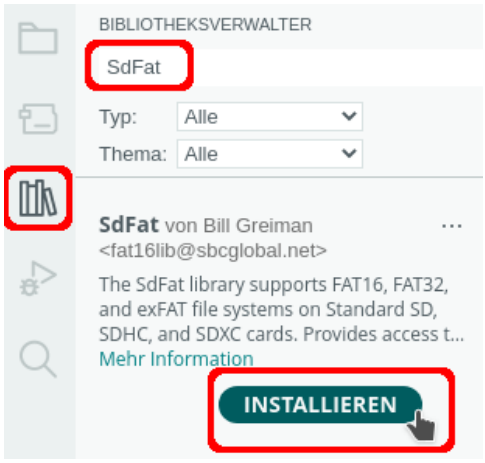
- 2 Taster
- mp3-Shield
- Leitungsdrähte

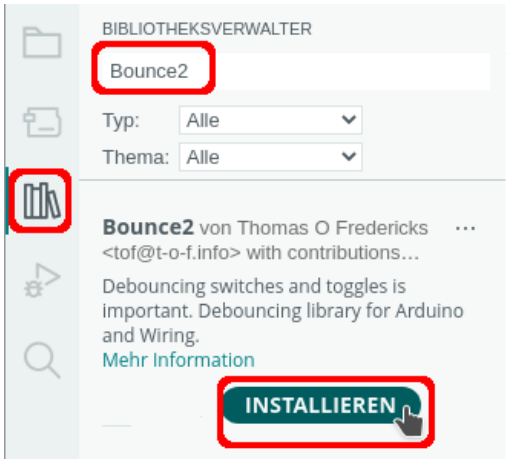
Baue die Schaltung auf.

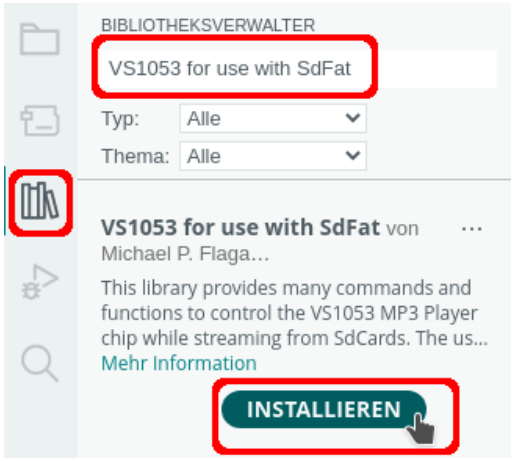


Die Sprachdateien wurden mit der Webseite https://ttsmp3.com erstellt. Du findest sie im Downloadbereich.

Benötigte Bibliotheken:







Mit Hilfe der Bibliothek Bounce2 soll das "Prellen" der Taster verhindert werden.
Binde die benötigten Bibliotheken ein und definiere die Variablen.

```
# include <vs1053_SdFat.h>
# include <SdFat.h>
# include <Bounce2.h>

SdFat sd;
vs1053 MP3player;

// die Taster
int TASTER1 = 6;
int TASTER2 = 7;

// "Prallverhinderer" für die Tasten starten
Bounce Taste_Spieler1 = Bounce();
Bounce Taste_Spieler2 = Bounce();

// Summe der Augen der beiden Spieler
int SummeSpieler1;
int SummeSpieler2;
```

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

int Zahl;
int TasterStatus;

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

Der setup-Teil. Beachte die Kommentare.

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

  // Instanzen des Objekts Bounce für jede Taste zuordnen
  // Zeitintervall einstellen
  Taste_Spieler1.attach(TASTER1);
  Taste_Spieler1.interval(20);
  Taste_Spieler2.attach(TASTER2);
  Taste_Spieler2.interval(20);

  // SD Karte lesen
  sd.begin(SD_SEL, SPI_FULL_SPEED);

  // in das Wurzelverzeichnis wechseln
  sd.chdir("/");

  // Player starten
  MP3player.begin();

  // Lautstärke setzen
  MP3player.setVolume(15, 15);

  // Zufallsgenerator starten
  randomSeed(analogRead(0));
  Serial.begin(9600);
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  // Zufallszahlen 1-6
  // solange die Sprachausgabe läuft -> Taster nicht erneut drücken
  int Minimum = 1;
  int Maximum = 7;

  // 5 Durchgänge
  while (Runde < 6)
  {
    // Spieler 1
    if (StartSpieler1)
    {
      if (Taste_Spieler1.update())
      {
        if (Taste_Spieler1.read() == LOW)
        {
          Serial.println("Runde " + String(Runde));
          Serial.println("----------------------------");

          // kurzes "Beep" spielen
          delay(1000);
          MP3player.SendSingleMIDInote();

          // würfeln
          int Zahl =  random(Minimum, Maximum);
          // Spielstand Serieller Monitor
          Serial.println("Spieler 1: gewuerfelte Zahl " + String(Zahl));
          delay(1000);

          /*
            Sprachausgabe gewürfelte Zahl
            211 -> Spieler 1
            Track(1) ... Track(6) Zahlen von 1 bis 6
            /*
            nach jeder Sprachausgabe
            ausreichend Zeit lassen
            Datei muss geladen und abgespielt werden
          */
          MP3player.playTrack(211);
          delay(2000);
          MP3player.playTrack(Zahl);

          // Summe berechnen und ausgeben
          SummeSpieler1 = SummeSpieler1 + Zahl;
          Serial.println("Spieler 1: Summe: " + String(SummeSpieler1));
          Serial.println("----------------------------");
          delay(2000);
```

```
        // Sprachausgabe: Summe ausgeben
        // 101 ... 130 Zahlen von 1 bis 30
        MP3player.playTrack(100 + SummeSpieler1);
        delay(2000);

        // Wechsel -> Spieler2 ist an der Reihe
        StartSpieler1 = !StartSpieler1;
        StartSpieler2 = !StartSpieler2;
      }
    }
  }

  // Spieler 2
  if (StartSpieler2)
  {
    if (Taste_Spieler2.update())
    {
      if (Taste_Spieler2.read() == LOW)
      {
        delay(1000);
        MP3player.SendSingleMIDInote();
        int Zahl =  random(Minimum, Maximum);
        Serial.println("Spieler 2: gewuerfelte Zahl " + String(Zahl));
        delay(1000);
        MP3player.playTrack(212);
        delay(2000);
        MP3player.playTrack(Zahl);
        SummeSpieler2 = SummeSpieler2 + Zahl;
        Serial.println("Spieler 2: Summe: " + String(SummeSpieler2));
        Serial.println("----------------------------");

        delay(2000);
        // Summe ausgeben
        MP3player.playTrack(100 + SummeSpieler2);

        /*
          ausreichend Zeit lassen
          Datei muss geladen und abgespielt werden
        */
        delay(2000);

        // nur bei Spieler2 Runde hochzählen, Spieler1 hat angefangen
        Runde ++;
        // Wechsel -> Spieler1 ist an der Reihe
        StartSpieler1 = !StartSpieler1;
        StartSpieler2 = !StartSpieler2;
      }
    }
  }
}
```

```
  // unentschieden
  if (SummeSpieler1 == SummeSpieler2)
  {
    Serial.println("unentschieden");
    Serial.println("----------------------------");
    MP3player.playTrack(200);
    delay(2000);

    // alle Werte zurücksetzen
    Runde = 1;
    SummeSpieler1 = 0;
    SummeSpieler2 = 0;
  }

  // Sieger Spieler1
  if (SummeSpieler1 > SummeSpieler2)
  {
    Serial.println("Sieger Spieler 1");
    Serial.println("----------------------------");
    MP3player.playTrack(201);
    delay(2000);

    // alle Werte zurücksetzen
    Runde = 1;
    SummeSpieler1 = 0;
    SummeSpieler2 = 0;
  }

  // Sieger Spieler2
  if (SummeSpieler1 < SummeSpieler2)
  {
    Serial.println("Sieger Spieler 2");
    Serial.println("----------------------------");
    MP3player.playTrack(202);
    delay(2000);

    // alle Werte zurücksetzen
    Runde = 1;
    SummeSpieler1 = 0;
    SummeSpieler2 = 0;
  }
}
```