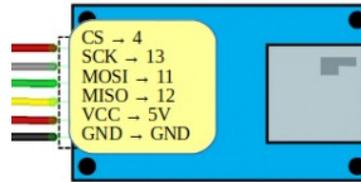


Der Arduino „spielt“ für dich 6 Zahlen. Anschließend werden die Lottozahlen gezogen, aufsteigend sortiert und auf die Anzahl der richtigen Zahlen untersucht.

Die Lottozahlen sollen mit einem SD-Kartenmodul auf eine SD-Karte gespeichert werden.

Benötigte Bauteile:

- ➔ Berührungssensor
- ➔ SD-Kartenleser
- ➔ Leitungsdrähte



Pinbelegung des SD-Karten-Moduls



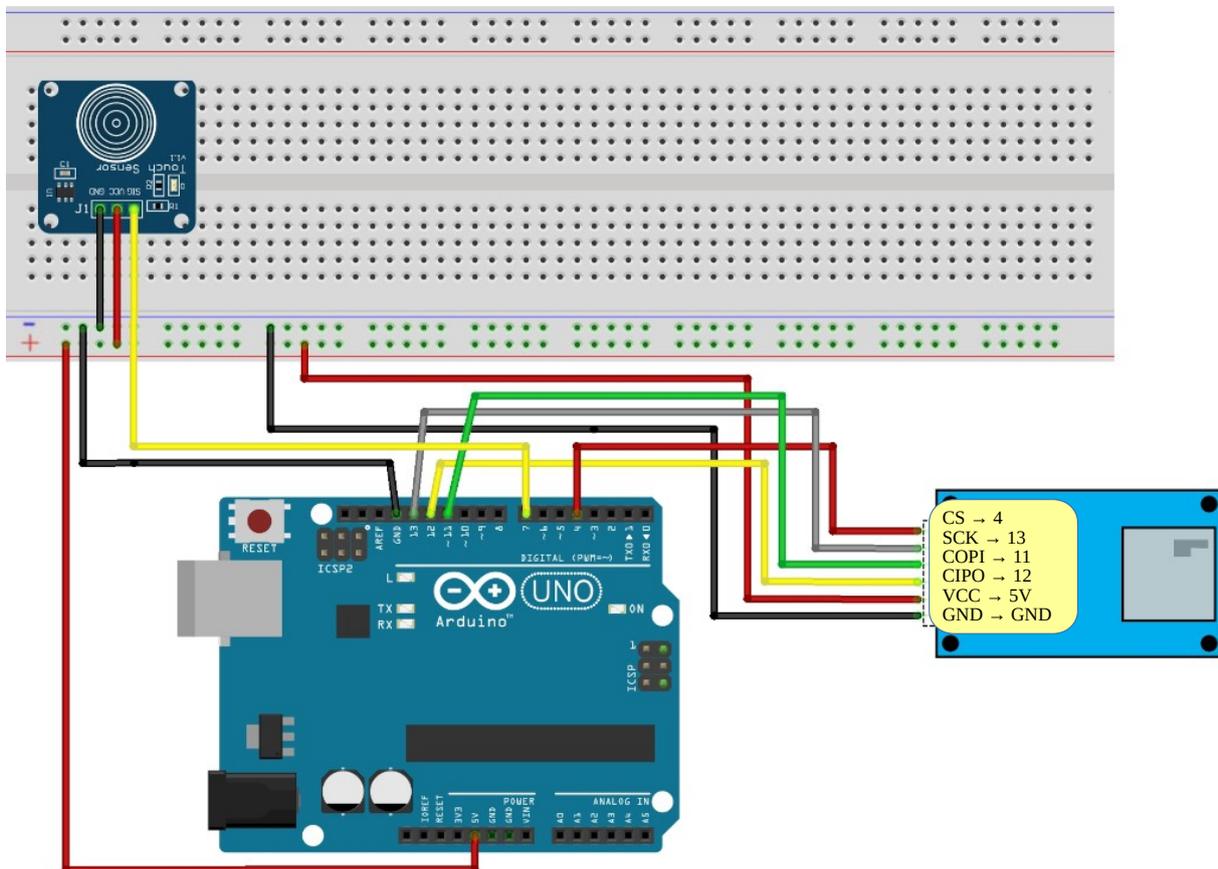
Achte auf die Pin-Belegung der SD-Karte!

Die SD-Karte muss mit FAT32 formatiert sein!

Benötigte Bauteile:

- ➔ Berührungssensor
- ➔ SD-Karten-Modul
- ➔ Leitungsdrähte

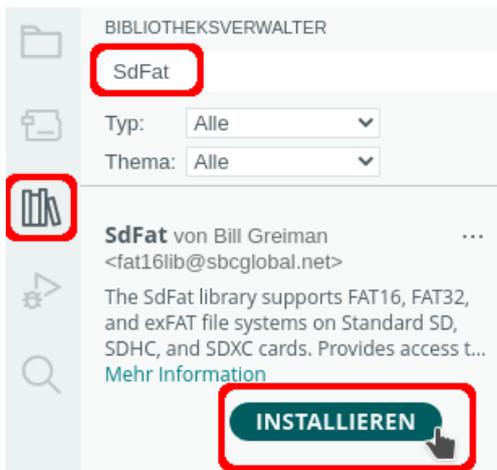
Baue die Schaltung auf:



fritzing

Als Bibliotheken stehen zwei zur Auswahl:

Die bereits eingebaute Bibliothek SD oder die Bibliothek SdFat.

Sketch → **Bibliothek einbinden** → **Bibliotheken verwalten**


Die Bibliothek SdFat hat gegenüber der Bibliothek SD den Vorteil, dass sie auch mit längeren Dateinamen (über die DOS-Konvention 8.3 hinaus) umgehen kann. In der eigentlichen Programmierung unterscheiden sie sich nicht.

Beim Programmstart soll abgefragt werden, ob die bisherige Datei gelöscht werden soll.

Binde die benötigten Bibliotheken ein und definiere die Variablen::

```
# include <SdFat.h>
/*
  Alternative:
  die bereits eingebaute Bibliothek verwenden:
  # include <SD.h>
  Der Dateiname muss dann auf 8.3 gekürzt werden
*/

// Bezeichnung der SD-Karte
// entfällt bei der Bibliothek SD
SdFat SD;

// Bezeichnung der Textdatei
File ZiehungLotto;

// Datenpin für das SD-Kartenmodul
int DatenPin = 10;

// Daten-Logger-Shield
// int DatenPin = 10;

// Pin des Sensors
int SENSOR = 7;
```

```
// Array für die gezogenen Zahlen
int LottoZahl[6];
// Array für die angekreuzten Zahlen
```

```
int SpielZahl[6];

// Minimum/Maximum der Zufallszahlen
int Minimum = 1;
int Maximum = 49;

// Anzahl der zu ziehenden Zahlen
int Anzahl = 6;
int SensorLesen;
```

Im setup-Teil muss die SD-Karte gestartet werden und die Abfrage vorbereitet werden:

```
void setup()
{
  // Zufallsgenerator starten
  randomSeed(analogRead(0));

  // bei Verwendung eines Tasters:
  pinMode(SENSOR, INPUT_PULLUP);
  //pinMode(SENSOR, INPUT);
  Serial.begin(115200);

  // auf serielle Verbindung warten
  while (!Serial) { ; }
  /*
   SD-Karte mit Angabe des Datenpins starten
   wenn die Intialisierung fehlschlägt
   - keine SD-Karte vorhanden
   - falsche Pinbelegung
   → es wird eine Fehlermeldung angezeigt
  */
  if (!SD.begin(DatenPin))
  {
    Serial.println(F("Initialisierung fehlgeschlagen!"));
  }
  else Serial.println(F("Initialisierung abgeschlossen"));
  Serial.println("Datei Lottozahlen.txt \u00f6schen (j/n)?");
}
```



Verwende für Bestimmung der Spielzahlen und die Ziehung der Lottozahlen die Funktion Ziehung:

```

int Ziehung()
{
    for (int i = 0; i < Anzahl; i++)
    {
        /*
            die Lottozahl gilt solange als bereits gezogen
            bis in der for-Schleife nachgewiesen wird, dass sie neu ist
            und BereitsGezogen den Wert false hat
        */
        bool BereitsGezogen = true;
        while (BereitsGezogen)
        {
            // Zahl ziehen
            LottoZahl[i] = random(Minimum, Maximum);
            BereitsGezogen = false;
            /*
                Zufallszahl mit den bereits gezogenen Zahlen vergleichen
                i wird in der ersten for-Schleife hochgezählt
                alle bisherigen Zahlen (ii) werden mit den bisher gezogenen
                (außer der gerade gezogenen) Zahlen verglichen (ii < i)
                ist sie identisch, erhält BereitsGezogen den Wert true
                es wird erneut eine Zufallszahl bestimmt
                wenn die Zahl neu ist, (BereitsGezogen hat noch den Wert false)
                wird die while-Schleife verlassen und die nächste Zahl
                gezogen
            */
            for (int ii = 0; ii < i; ii++)
            {
                if (LottoZahl[i] == LottoZahl[ii]) BereitsGezogen = true;
            }
        }
    }
    /*
        das Array mit den Zahlen wird an das Hauptprogramm zurückgegeben
        beim ersten Aufruf sind es die gespielten Zahlen
        beim zweiten Aufruf die gezogenen Lottozahlen
    */
    return LottoZahl[Anzahl];
}
    
```



Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  while (Serial.available() > 0) {
    while (Serial.available() > 0)
    {
      char Eingabe = Serial.read();

      /*
       prüfen, ob die Datei existiert
       wenn ja → Datei Lottozahlen.txt löschen
      */
      if (Eingabe == 'j')
      {
        if (SD.exists("Lottozahlen.txt"))
        {
          // Datei entfernen
          SD.remove("Lottozahlen.txt");
          Serial.println(F("Datei Lottozahlen.txt entfernt!"));
        }
        else
        {
          Serial.println(F("Datei Lottozahlen.txt existiert nicht!"));
        }
      }

      // bei jeder anderen Taste
      else
      {
        // \u00fc = ü
        Serial.println(F("Lottozahlen werden der Datei Lottozahlen.txt hinzugef\u00fcgt!"));
      }
    }
  }
  String AnzahlTreffer = "";
  String GespielteZahlen = "";
  String GezogeneZahlen = "";
  int Treffer = 0;
  SensorLesen = digitalRead(SENSOR);

  // bei Verwendung eines Tasters:
  // if (SensorLesen == LOW)
  if (SensorLesen == LOW)
  {
    delay(200);

    // Ziehung der "angekreuzten" Zahlen
    Ziehung();
    ZahlenSortieren(LottoZahl, 6);

    // String GespielteZahlen aus den Elementen des Arrays LottoZahl zusammensetzen
    for (int i = 0; i < Anzahl; i++)
    {
      GespielteZahlen = GespielteZahlen + LottoZahl[i] + " ";
      SpielZahl[i] = LottoZahl[i];
    }
  }
}
```

```

// Ziehung der Lottozahlen
Ziehung();
ZahlenSortieren(LottoZahl, 6);

// Vergleichen der Elemente der Arrays SpielZahl und LottoZahl
for (int i = 0; i < Anzahl; i++) {
  for (int ii = 0; ii < Anzahl; ii++)
  {
    // Übereinstimmung gefunden -> Treffer um 1 erhöhen
    // AnzahlTreffer: String für übereinstimmende Zahlen erzeugen
    if (SpielZahl[i] == LottoZahl[ii]) {
      AnzahlTreffer = AnzahlTreffer + SpielZahl[i] + " ";
      Treffer++;
    }
  }
}

// String GezogeneZahlen aus den Elementen des Arrays LottoZahl
// zusammensetzen
for (int i = 0; i < Anzahl; i++)
{
  GezogeneZahlen = GezogeneZahlen + LottoZahl[i] + " ";
}

/*
  Datei zum Schreiben (FILE_WRITE) öffnen
  wenn sie noch nicht existiert, wird sie erstellt
*/
ZiehungLotto = SD.open("Lottozahlen.txt", FILE_WRITE);
if (ZiehungLotto)
{
  Serial.println("Schreibe Zahlen in Lottozahlen.txt ... ");
  Serial.println("-----");

  // Daten in die Datei schreiben
  ZiehungLotto.println("Gespielte Zahlen");
  ZiehungLotto.println(GespielteZahlen);
  ZiehungLotto.println("-----");
  ZiehungLotto.println("Gezogene Zahlen");
  ZiehungLotto.println(GezogeneZahlen);
  ZiehungLotto.println("-----");

  // keine Treffer
  if (Treffer == 0) ZiehungLotto.println("keine Treffer!");
  // getroffene Zahlen anzeigen
  else ZiehungLotto.println("Treffer: " + AnzahlTreffer);
  ZiehungLotto.println("-----");

  // Datei schließen
  ZiehungLotto.close();
  Serial.println("Abgeschlossen.");
  Serial.println();
}

```

```
else
{
  Serial.println("Datei Lottozahlen.txt konnte nicht gelesen werden");
}

// Datei öffnen und Zahlen anzeigen
ZiehungLotto = SD.open("Lottozahlen.txt");
if (ZiehungLotto)
{
  Serial.println("Lese Lottozahlen.txt ...");

  // solange sich Zeilen in der Datei befinden ...
  while (ZiehungLotto.available())
  {
    // ... werden sie gelesen und im Seriellen Monitor ausgegeben
    Serial.write(ZiehungLotto.read());
  }
  // Datei schließen
  ZiehungLotto.close();
}
else
{
  Serial.println("Lottozahlen konnte nicht geu00f6ffnet werden");
}
}
```

Die Funktion ZahlenSortieren sortiert die Zahlen aufsteigend.

```
void ZahlenSortieren(int Zahlen[], int Groesse)
{
  // unsortierten Bereich des Arrays durchlaufen
  for (int i = 0; i < (Groesse - 1); i++)
  {
    // bei jedem Durchlauf wird das jeweils letzte Element weggelassen
    for (int ii = 0; ii < (Groesse - (i + 1)); ii++)
    {
      /*
       wenn die aktuelle Zahl größer als die nachfolgende Zahl ist
       -> aktuelle Zahl temporär speichern -> Zahlen vertauschen
       -> temporäre Zahl der nachfolgenden Zahl zuweisen
      */
      if (Zahlen[ii] > Zahlen[ii + 1])
      {
        int zwischengespeicherteZahl = Zahlen[ii];
        Zahlen[ii] = Zahlen[ii + 1];
        Zahlen[ii + 1] = zwischengespeicherteZahl;
      }
    }
  }
}
```