

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

Zu Beginn des Programms wird gefragt, ob die bestehende Datei „Lottozahlen.txt“ gelöscht werden soll.

Der Taster erfüllt mehrere Funktionen:

ein Druck auf den Taster: die Lottozahlen werden mit Datum und Uhrzeit im Seriellen Monitor angezeigt

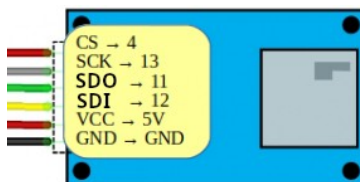
Doppelklick auf den Taster: die Lottozahlen werden mit Datum und Uhrzeit auf der SD-Karte gespeichert

langer Druck auf den Taster: die auf der SD-Karte gespeicherten Zahlen werden im Seriellen Monitor angezeigt

Die Lottozahlen sollen auf eine SD-Karte gespeichert werden, das RTC-Modul ermittelt das aktuelle Datum und die aktuelle Zeit.

Benötigte Bauteile:

- ➔ Taster
- ➔ SD-Karten-Modul
- ➔ RTC-Modul
- ➔ Leitungsdrähte



SD0 Serial Data Out

SDI Serial Data In

Pinbelegung des SD-Karten-Moduls



Achte auf die Pin-Belegung der SD-Karte!

Die SD-Karte muss mit FAT32 formatiert sein!

So sieht es aus:

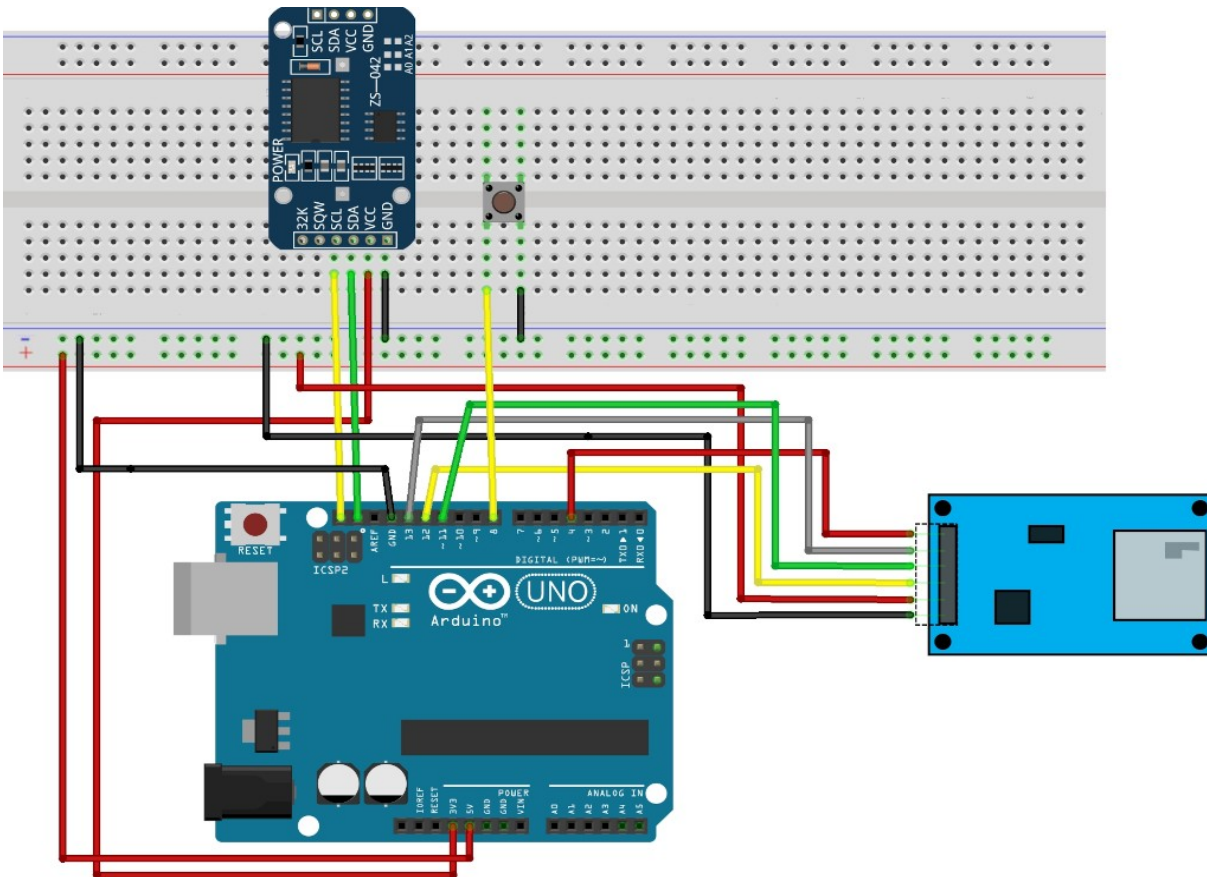
```
Ausgabe  Serieller Monitor x
Nachricht (Enter, um Nachricht an 'Arduino Uno' auf '/dev/ttyACM0') zu senden  Zeilenumbruch  9600 baud

Initialisierung abgeschlossen
kurzer Druck auf den Taster -> Lottozahlen im Seriellen Monitor anzeigen
Doppelklick auf den Taster-> Lottozahlen speichern
langer Druck auf den Taster -> Datei lesen und Lottozahlen im Seriellen Monitor anzeigen
Datei Lottozahlen.txt löschen (j/n)?
```

```
Ausgabe  Serieller Monitor x
Nachricht (Enter, um Nachricht an 'Arduino Uno' auf '/dev/ttyACM0') zu senden  Zeilenumbruch  9600 baud

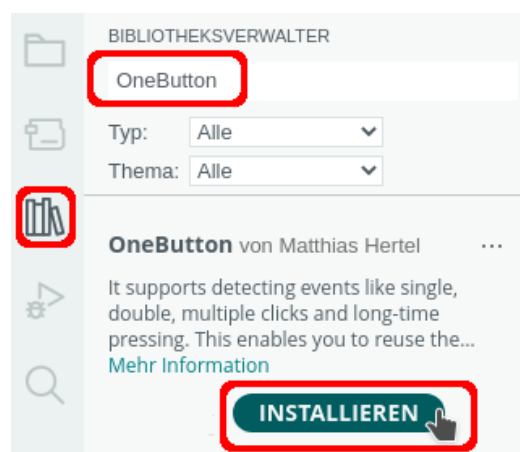
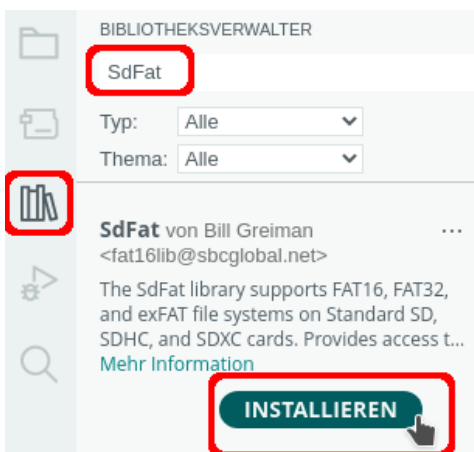
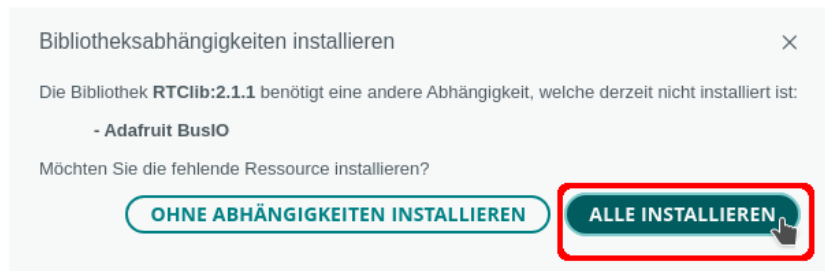
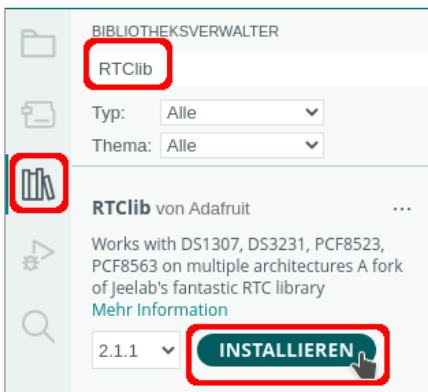
07.08.2022 Uhrzeit: 20:55:20
Gespielte Zahlen
1 3 8 11 19 35
-----
Gezogene Zahlen
10 19 21 27 35 41
-----
Treffer: 19 35
-----
07.08.2022 Uhrzeit: 20:55:35
Gespielte Zahlen
5 16 18 19 25 30
-----
Gezogene Zahlen
1 8 9 25 33 48
-----
Treffer: 25
-----
Lese Lottozahlen.txt ...
07.08.2022 Uhrzeit: 20:02:50
Gespielte Zahlen
17 23 27 28 30 36
-----
Gezogene Zahlen
9 20 30 35 40 46
-----
Treffer: 30
-----
07.08.2022 Uhrzeit: 20:02:53
Gespielte Zahlen
24 39 41 42 43 45
-----
Gezogene Zahlen
1 2 7 16 38 47
-----
keine Treffer!
-----
07.08.2022 Uhrzeit: 20:02:55
Gespielte Zahlen
3 19 22 23 38 47
-----
Gezogene Zahlen
22 26 27 34 41 43
-----
```

Baue die Schaltung auf:



fritzing

Benötigte Bibliotheken:



Binde die benötigten Bibliotheken ein und definiere die Variablen:

```
# include <SdFat.h>
# include <RTCLib.h>
# include <OneButton.h>

// Bezeichnung der SD-Karte
SdFat SD;

// Bezeichnung des RTC-Moduls
RTC_DS3231 rtc;

// Bezeichnung der Textdatei
File ZiehungLotto;

// Datenpin für das SD-Kartenmodul
int DatenPin = 4;

int TASTER = 8;

// Name des Tasters
OneButton NameTaster(TASTER, true);

// 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;
```

Im setup-Teil müssen das SD-Karten-Modul und das RTC-Modul gestartet werden und die Abfrage vorbereitet werden:

```
void setup()
{
  /*
   * wenn Datum und Zeit nicht korrekt -> Datum/Zeit setzen
   * Jahr, Monat, Tag, Stunde, Minute, Sekunde
   * rtc.adjust(DateTime(2020, 12, 13, 13, 31, 30));
   */
  // RTC-Modul starten
  rtc.begin();

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

```

// Aktionen dem Modus des Tasters zuordnen
NameTaster.attachClick(einKlick);
NameTaster.attachDoubleClick(DoppelKlick);
NameTaster.attachLongPressStop(langerDruckStopp);

/*
  Anzahl der Millisekunden festlegen
  Standardwerte gesetzt:
  PressTicks: 1000
  ClickTicks: 600
  DebounceTicks: 50
  wenn die Standardwerte gesetzt werden sollen
  können die nächsten Zeilen auskommentiert werden
*/
NameTaster.setPressTicks(1000);
NameTaster.setClickTicks(500);
NameTaster.setDebounceTicks(50);
pinMode(TASTER, INPUT_PULLUP);
Serial.begin(9600);
// 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"));

// Menü
Serial.print(F("kurzer Druck auf den Taster "));
Serial.println(F("-> Lottozahlen im Seriellen Monitor anzeigen"));
Serial.print(F("Doppelklick auf den Taster"));
Serial.println(F("-> Lottozahlen speichern"));
Serial.print(F("langer Druck auf den Taster "));
Serial.println(F("-> Datei lesen und Lottozahlen im Seriellen Monitor anzeigen"));
Serial.println(F("Datei Lottozahlen.txt l\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];
}
```

Im loop-Teil wird die Eingabe verarbeitet und der Taster abgefragt.

```
void loop()
{
  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!"));
    }
  }

  // Taster alle 10 Millisekunden abfragen
  NameTaster.tick();
  delay(10);
}
```

Die Funktionen:

einKlick() → Serielle Ausgabe der Lottozahlen

Doppelklick() → Lottozahlen auf der SD-Karte speichern

langerDruckStopp() - Datei lesen und im Seriellen Monitor anzeigen

ZahlenSortieren() → Lottozahlen sortieren

```
void einklick()
{
  String AnzahlTreffer = "";
  String GespielteZahlen = "";
  String GezogeneZahlen = "";
  int Treffer = 0;

  // aktuelle Zeit holen
  DateTime aktuell = rtc.now();

  char Datum[] = "DD.MM.YYYY";
  Serial.print(aktuell.toString(Datum));

  // Zeit schreiben
  char Zeit[] = " Uhrzeit: hh:mm:ss";
  Serial.println(aktuell.toString(Zeit));

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

  for (int i = 0; i < Anzahl; i++)
  {
    // String GespielteZahlen aus den Elementen des Arrays
    // LottoZahl zusammensetzen
    GespielteZahlen = GespielteZahlen + LottoZahl[i] + " ";
    /*
     dem Array SpielZahl (die „gespielten“ Zahlen)
     wird das identische Element des Arrays LottoZahl zugewiesen
     Dieser Schritt darf für die Ziehung der Lottozahlen
     nicht wiederholt werden.
    */
    SpielZahl[i] = LottoZahl[i];
  }

  // Ziehung der Lottozahlen
  Ziehung();

  // Zahlen sortieren
  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] + " ";
}
// Daten in die Datei schreiben
Serial.println("Gespielte Zahlen");
Serial.println(GespielteZahlen);
Serial.println("-----");
Serial.println("Gezogene Zahlen");
Serial.println(GezogeneZahlen);
Serial.println("-----");

// keine Treffer
if (Treffer == 0) Serial.println("keine Treffer!");

// getroffene Zahlen anzeigen
else Serial.println("Treffer: " + AnzahlTreffer);
Serial.println("-----");
}

```

```

void DoppelKlick()
{
    String AnzahlTreffer = "";
    String GespielteZahlen = "";
    String GezogeneZahlen = "";
    int Treffer = 0;

    // aktuelle Zeit holen
    DateTime aktuell = rtc.now();

    char Datum[] = "DD.MM.YYYY";
    Serial.print(aktuell.toString(Datum));

    // Zeit schreiben
    char Zeit[] = "Uhrzeit: hh:mm:ss";
    Serial.println(aktuell.toString(Zeit));

    // 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();

// Zahlen sortieren
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(F("Schreibe Zahlen in Lottozahlen.txt ... "));
    Serial.println(F("-----"));

    char Datum[] = "DD.MM.YYYY";
    ZiehungLotto.print(aktuell.toString(Datum));

    // Zeit schreiben
    char Zeit[] = "Uhrzeit: hh:mm:ss";
    ZiehungLotto.println(aktuell.toString(Zeit));
}
```

```
// 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(F("Abgeschlossen."));
Serial.println();
}

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

```
void langerDruckStopp()
{
  // Datei öffnen und Zahlen anzeigen
  ZiehungLotto = SD.open("Lottozahlen.txt");
  if (ZiehungLotto)
  {
    Serial.println(F("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(F("Lottozahlen konnte nicht geu00f6ffnet werden"));
  }
}
```

```
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;
      }
    }
  }
}
```