

Pin-Eingabe

Auf einem Tastenfeld soll ein Pin eingegeben werden. Das Programm soll einen zufälligen Pin abfragen. Wurde der Pin nach Druck auf die Taste # richtig eingegeben, wird im Seriellen Monitor die Meldung „korrekter Pin“ angezeigt.



[Pin-Eingabe Serieller Monitor](#)

Bei dieser Anleitung lernst du verschiedene Möglichkeiten kennen einen String auszuwerten:

BeispielString.indexOf(Zeichen)	→ bestimmt die Position eines Zeichens innerhalb des Strings
BeispielString.length()	→ ermittelt die Länge eines Strings
BeispielString.substring(Start, Ende)	→ zeigt den Teil des Strings von Start bis Ende

Beispiel:

Der String „Programmieren finde ich toll ;-)" soll auf seine Länge und die Position des „t“ untersucht werden. Anschließend wird er in zwei Teilstrings zerlegt.

```
void setup()
{
  Serial.begin(9600);
  String BeispielString = "Programmieren finde ich toll ;-)";

  // Länge bestimmen
  int Laenge = BeispielString.length();
  Serial.print("Laenge des Strings: ");
  Serial.println(Laenge);

  // das t suchen
  int Suche = BeispielString.indexOf("t");
  Serial.print("Position t: ");
  Serial.println(Suche);

  // String vom Anfang bis zum t
  BeispielString.substring(0, Suche);
  Serial.print("String bis zum t: ");
  Serial.println(BeispielString.substring(0, Suche));

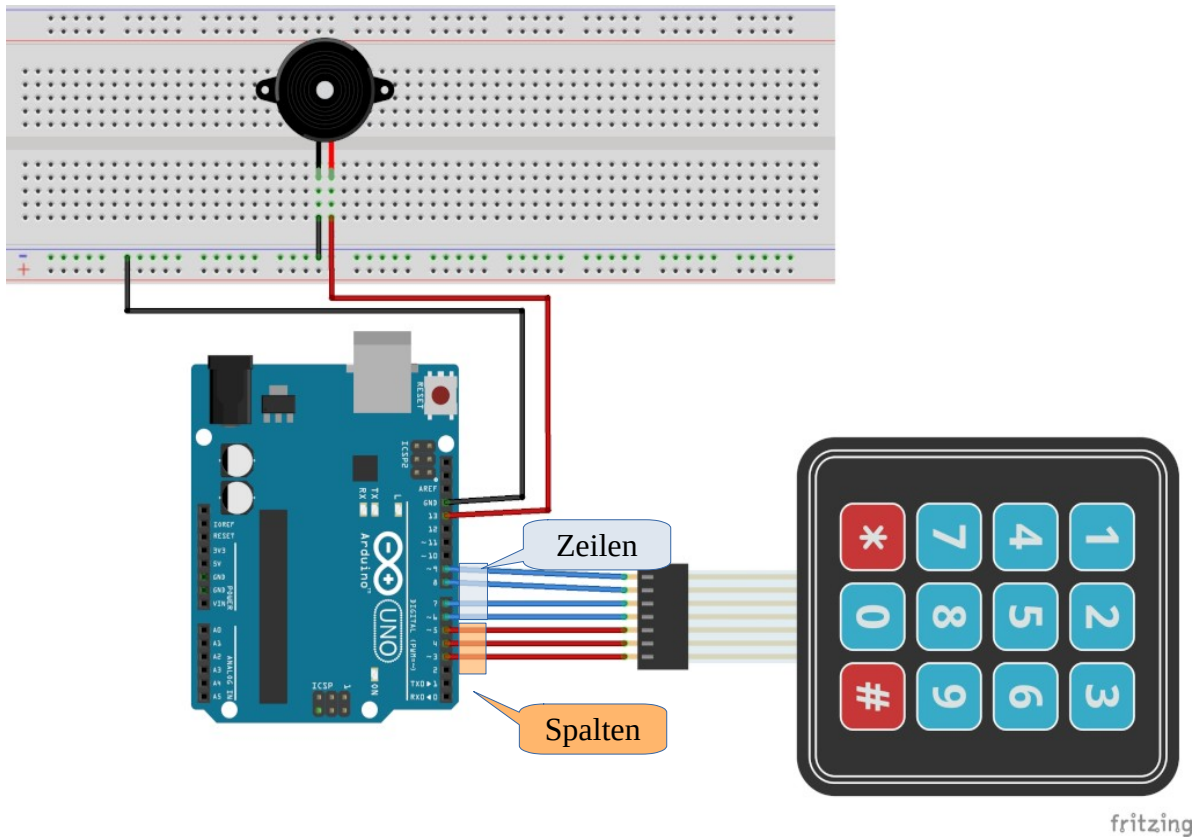
  // oder, weil Laenge schon bestimmt wurde:
  // BeispielString.substring(Suche + 1, Laenge);
  Serial.print("String vom t bis zum Ende: ");
  Serial.println(BeispielString.substring(Suche, BeispielString.length()));
}

void loop()
{
  // bleibt leer → das Programm soll nur einmal ausgeführt werden
}
```

Benötigte Bauteile:

- ➔ Lautsprecher
- ➔ Tastenfeld 3×4
- ➔ Leitungsdrähte

Baue die Schaltung auf.



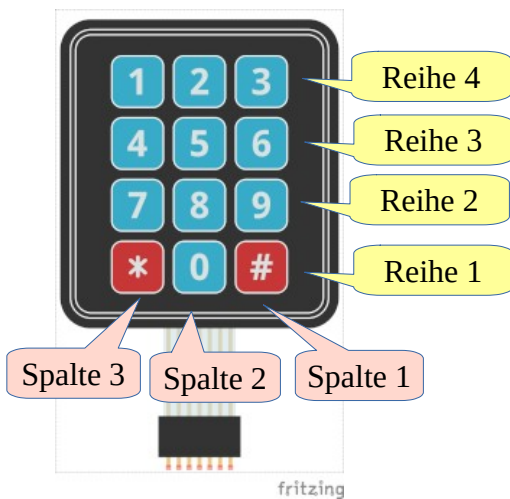
fritzing

Du benötigst die Bibliothek Adafruit Keypad:

The screenshot shows the Arduino IDE interface. In the 'BIBLIOTHEKSVERWALTER' (Library Manager) window, 'Adafruit Keypad' is selected in the search bar. Below, the library details for 'Adafruit Keypad von Adafruit' are shown, with an 'INSTALLIEREN' button highlighted. A dialog box titled 'Bibliotheksabhängigkeiten installieren' (Install library dependencies) is open, indicating that the 'Adafruit Keypad:1.3.0' library requires 'Adafruit NeoPixel', which is not installed. The dialog asks 'Möchten Sie die fehlende Ressource installieren?' (Do you want to install the missing resource?) and provides two options: 'OHNE ABHÄNGIGKEITEN INSTALLIEREN' (Install without dependencies) and 'ALLE INSTALLIEREN' (Install all), with the latter button highlighted.

Das Tastenfeld besteht aus Reihen und Spalten, die von unten nach oben in einem **Array** angeordnet werden:

```
// Array 3 x 4
char Tasten[REIHEN][SPALTEN] = {
    {'#', '0', '*'},
    {'9', '8', '7'},
    {'6', '5', '4'},
    {'3', '2', '1'}
};
```



Binde die Bibliothek ein und definiere die Variablen:

```
# include "Adafruit_Keypad.h"

int LAUTSPRECHER = 13;

// Größe des Tastenfeldes
// 3 Spalten
const byte SPALTEN = 3;

// 4 Zeilen
const byte REIHEN = 4;

// die Ziffern/Zeichen: Array 3 x 4
char Tasten[REIHEN][SPALTEN] =
{
    {'#', '0', '*'},
    {'9', '8', '7'},
    {'6', '5', '4'},
    {'3', '2', '1'}
};
```

```
// die Pins für die 3 Spalten
byte SpaltenPins[SPALTEN] = { 3, 4, 5 };

// die Pins für die 4 Zeilen
byte ReihenPins[REIHEN] = { 6, 7, 8, 9 };

// TastenFeld -> Name des Keypads
// -> Zuordnung der Pins zu den REIHEN und SPALTEN des Arrays
Adafruit_Keypad Tastenfeld = Adafruit_Keypad(makeKeymap(Tasten), ReihenPins,
SpaltenPins, REIHEN, SPALTEN);

String Vergleich;
String Pin;

// char-Array für die eingegebenen Zeichen
char Zeichen[5];

int Position = 0;
```

Der setup-Teil zeigt die Meldung beim Start an:

```
void setup()
{
  // Tastenfeld starten
  Tastenfeld.begin();

  pinMode(LAUTSPRECHER, OUTPUT);
  Serial.begin(9600);

  // auf serielle Verbindung warten
  while (!Serial) {;}
  delay(500);

  // nur beim Start anzeigen
  Serial.println("Bitte 4-stelligen Pin eingeben:");
  Serial.println("Pin-Eingabe mit # abschlie\u00dfen");
  Serial.println("Korrektur mit * ");

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

  // neuen Pin erstellen
  neuerPin();
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  // gedrückte Taste lesen
  Tastenfeld.tick();
  while (Tastenfeld.available())
  {
    keypadEvent Taste = Tastenfeld.read();

    // wenn die Taste losgelassen wurde
    if (Taste.bit.EVENT == KEY_JUST_RELEASED)
    {
      tone(LAUTSPRECHER, 1000);
      delay(50);
      noTone(LAUTSPRECHER);

      // eingegebenes Zeichen an der aktuellen Position speichern
      Zeichen[Position] = (char)Taste.bit.KEY;

      // Zeichen anzeigen
      Serial.print(Zeichen[Position]);

      // nächstes Zeichen -> Position erhöhen
      Position ++;

      // Korrektur ASCII-Wert 42 = *
      if (Taste.bit.KEY == 42)
      {
        // char-Array Zeichen leeren
        for (int i = 0; i < sizeof(Zeichen); i ++)
        {
          Zeichen[i] = NULL;
        }
        Position = 0;
        Serial.println("\tKorrektur: ");
      }

      // Zeichen ASCII-Wert 35 = #
      if (Taste.bit.KEY == 35)
      {
        // char-Array in String umwandeln
        Vergleich = String(Zeichen);

        // letzte Zeichen des Strings sind 0 und # -> müssen entfernt werden
        Vergleich = Vergleich.substring(0, Vergleich.length() - 2);
        Serial.println();
        Serial.print(Vergleich);

        // Eingabe mit Pin vergleichen
        if (Vergleich == Pin)
        {
          Serial.println("\tPin ist korrekt - Zugriff");
        }
        else Serial.println("\tFalscher Pin - kein Zugriff");
      }
    }
  }
}
```

```
// Variable zurücksetzen
Position = 0;

// neuen Pin erstellen
neuerPin();
}
}
}
```

Jetzt fehlt noch die Funktion für den zufällig erzeugten Pin.

```
String neuerPin()
{
  // zufälligen PIN bestimmen
  Pin = "";
  Serial.println("-----");

  for (int i = 0; i < sizeof(Zeichen) - 1; i ++ )
  {
    int Zahl = random(0, 10);
    Pin = Pin + String(Zahl);
  }

  // Pin anzeigen
  Serial.println("neuer Pin: " + Pin);
  return Pin;
}
```

Hartmut Waller (hartmut-waller.info/arduino-blog) Letzte Änderung: 20.12.22