

Pin-Eingabe

Auf einem Tastenfeld soll ein Pin eingegeben werden. Das Programm soll diesen Pin („1234“) abfragen. Wurde der Pin nach Druck auf die Taste # richtig eingegeben, wird die Meldung „korrekter Pin“ angezeigt.



[Pineingabe Serieller Monitor](#)

Bei dieser Aufgabe 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));

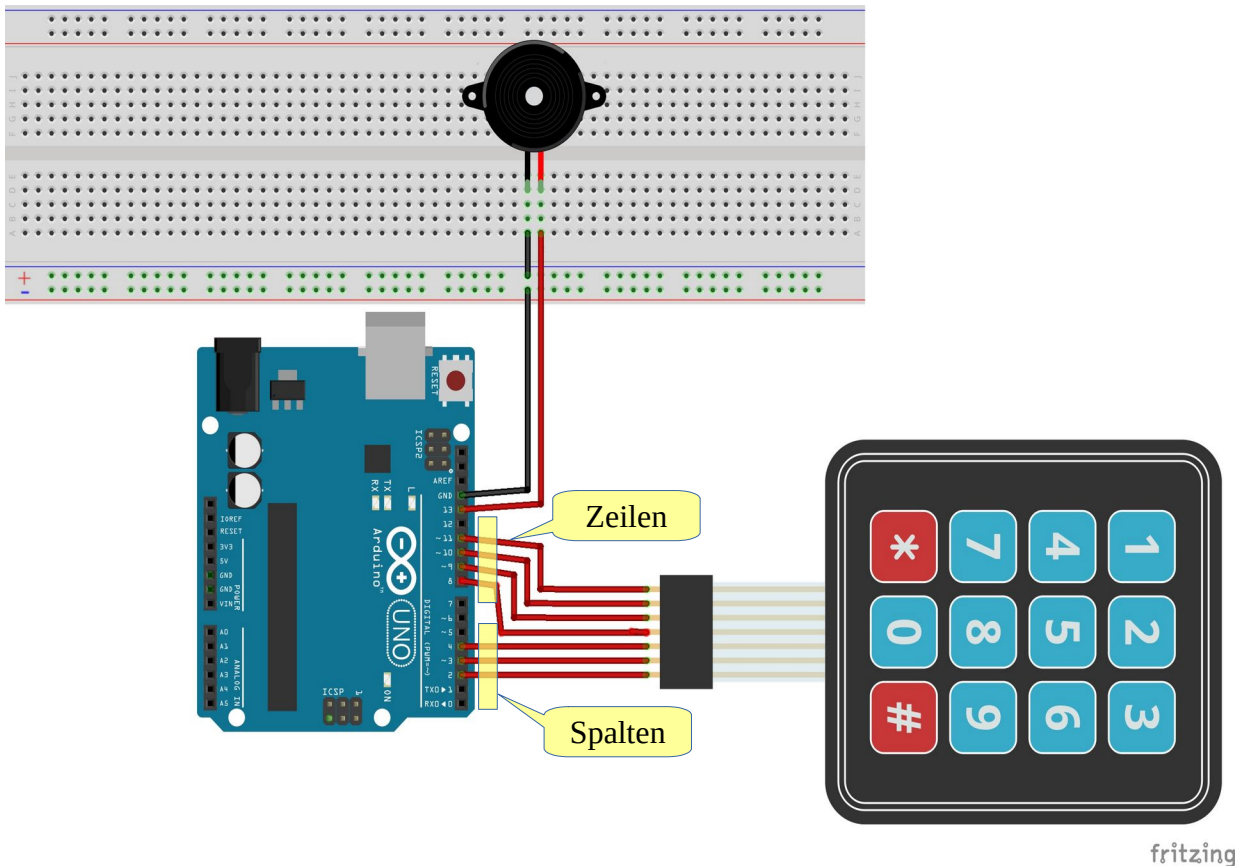
  // oder, weil Laenge schon bestimmt wurde:
  // BeispielString.substring(Suche + 1, Laenge);
  Serial.print("String vom t bis zum Ende: ");
  Serial.println(BeispielString.substring(Suche, BeispielString.length()));
}

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

Benötigte Bauteile:

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

Baue die Schaltung auf.



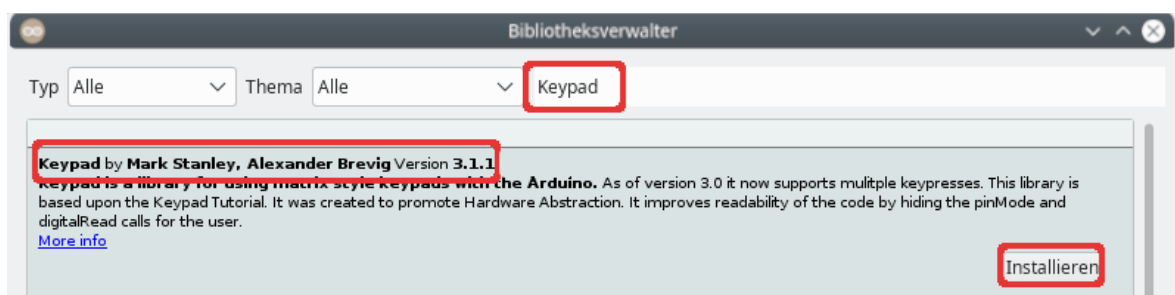
fritzing

Das Programm soll einen Pin („1234“) abfragen. Wurde der Pin nach Druck auf die Taste # richtig eingegeben, wird die Meldung „korrekter Pin“ angezeigt.

Es sind einige Schritte zur Vorbereitung nötig. Im Kopf des Programms müssen verschiedene Variable definiert werden:

- 1 Du benötigst die Bibliothek Keypad:

Sketch → Bibliothek einbinden → Bibliotheken verwalten



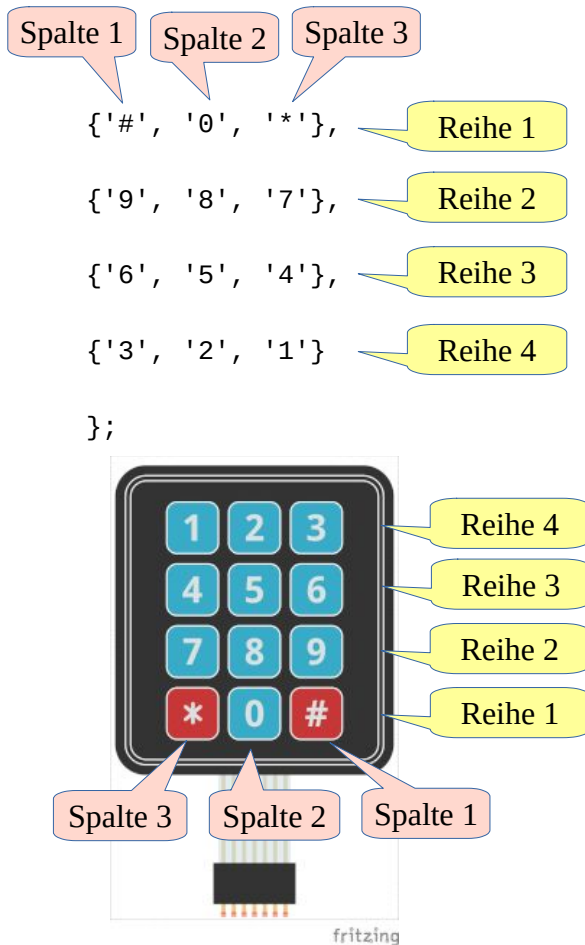
Binde die Bibliothek ein:

```
#include <Keypad.h>
```

2

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'}
};
```



3

Jedem Pin wird eindeutig eine Taste zugeordnet:

```
/*
    Größe des Tastenfeldes
    3 Spalten. 4 Zeilen
    bei der Definition der Größe eines Arrays müssen
    die Angaben mit const definiert werden
*/
const byte SPALTEN = 3;
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] = { 2, 3, 4 };  
  
// die Pins für die 4 Zeilen  
byte ReihenPins[REIHEN] = { 8, 9, 10, 11 };
```

4 Jeder Taste muss eindeutig ein Pin zugeordnet (makeKeymap) werden:

```
// TastenFeld → Name des Keypads  
// → Zuordnung der Pins zu den REIHEN und SPALTEN des Arrays  
Keypad TastenFeld = Keypad(makeKeymap(Tastens, ReihenPins, SpaltenPins,  
REIHEN, SPALTEN));
```

Die Definition der Variablen als Ganzes:

```
// Bibliothek einbinden  
# include <Keypad.h>  
  
String Eingabe;  
String Pin = "1234";  
int LAUTSPRECHER = 13;  
  
/*  
  Größe des Tastenfeldes  
  3 Spalten. 4 Zeilen  
  bei der Definition der Größe eines Arrays müssen  
  die Angaben mit const definiert werden  
*/  
const byte SPALTEN = 3;  
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] = { 2, 3, 4 };

// die Pins für die 4 Zeilen
byte ReihenPins[REIHEN] = { 8, 9, 10, 11 };

// TastenFeld → Name des Keypads
// → Zuordnung der Pins zu den REIHEN und SPALTEN des Arrays
Keypad TastenFeld = Keypad(makeKeymap(Tasten), ReihenPins, SpaltenPins, REIHEN,
SPALTEN);
```

Jetzt kann die eigentliche Programmierung beginnen.
Der setup-Teil zeigt die Meldung beim Start an:

```
void setup()
{
  pinMode(LAUTSPRECHER, OUTPUT);
  Serial.begin(9600);

  delay(500);

  // Text nur beim Start anzeigen
  Serial.println("Pin-Eingabe");
  Serial.println("mit # abschließen");
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  // gedrückte Taste abfragen
  gedruckteTaste = TastenFeld.getKey();

  // wenn eine Taste gedrückt wurde
  if (gedruckteTaste)
  {
    tone(LAUTSPRECHER, 1000);
    delay(50);
    noTone(LAUTSPRECHER);

    // gedrückte Taste wird mit dem Zeichen # verglichen
    // # soll nicht angezeigt werden
    if (gedruckteTaste != '#') Serial.print(gedruckteTaste);

    Eingabe = Eingabe + gedruckteTaste;
    // oder Kurzform:
    // Eingabe += gedruckteTaste;
```

```
// # → Vergleich durchführen
if (gedrueckteTaste == '#')
{
  // letztes Zeichen ist # → muss entfernt werden
  String Vergleich = Eingabe.substring(0, Eingabe.length() - 1);

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

Die Taste * soll eine Korrektur des Pins ermöglichen. Ersetze den loop-Teil.

```
void loop()
{
  // gedrückte Taste abfragen
  gedruckteTaste = TastenFeld.getKey();

  // wenn eine Taste gedrückt wurde
  if (gedruckteTaste)
  {
    tone(LAUTSPRECHER, 1000);
    delay(50);
    noTone(LAUTSPRECHER);

    // # soll nicht angezeigt werden
    if (gedruckteTaste != '#') Serial.print(gedruckteTaste);
    Eingabe = Eingabe + gedruckteTaste;

    // Korrektur mi *
    if (gedruckteTaste == '*' && Eingabe.length() > 1)
    {
      // letztes Zeichen ist * → muss entfernt werden
      Eingabe = Eingabe.substring(0, Eingabe.length() - 2);
      Serial.println("Korrektur: ");
      Serial.print(Eingabe);
    }

    if (gedruckteTaste == '#')
    {
      // letztes Zeichen ist # → muss entfernt werden
      String Vergleich = Eingabe.substring(0, Eingabe.length() - 1);
      if (Vergleich == Pin)
      {
        Serial.println("\tPin ist korrekt - Zugriff");
      }
      else Serial.println("\tFalscher Pin - kein Zugriff");
      Eingabe = "";
    }
  }
}
```