

Code-Schloss mit Tastatur und Servo-Motor

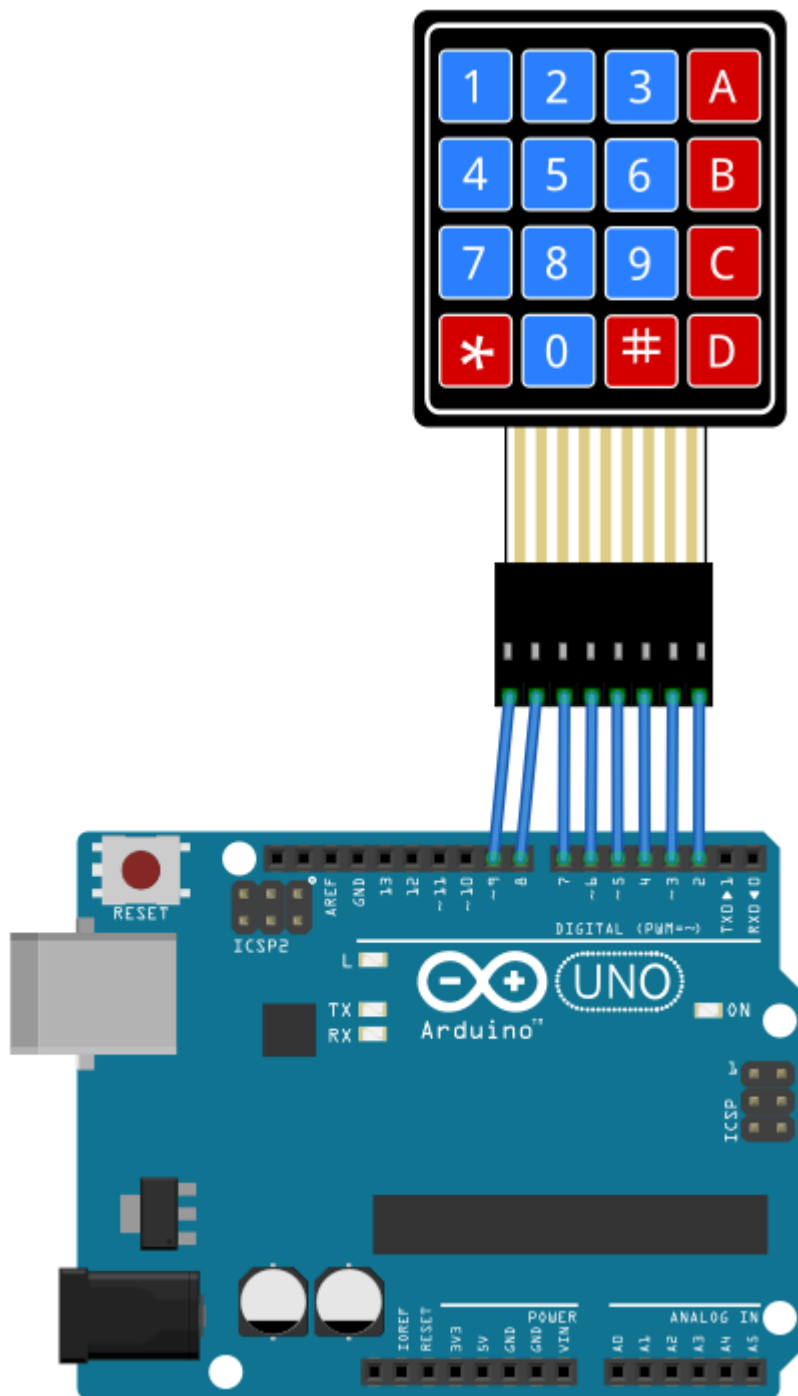


In diesem Projekt baust du dir ein Code-Schloss mit einer Folientastatur und einem Servo-Motor. Nur per Eingabe der richtigen Kombination wird sich der Servo-Motor in Bewegung setzen und damit eine mögliche Verriegelung lösen.

DIE FOLIENASTATUR ANSCHLIESSEN

Für ein Code-Schloss braucht es natürlich ein Gerät, über das man den Code eingeben kann. Hierfür eignet sich die 4x4 Folientastatur perfekt.

Zunächst installierst du die Verbindung zum Arduino. Die Tastatur besitzt 8 Buchsen; stecke hier ebenso viele Kabel hinein und verbinde diese der Reihe nach mit den Digitalpins 9 bis 2 an deinem Arduino:



DIE PASSENDE BIBLIOTHEK

Um dir die Steuerung der Tastatur so einfach wie möglich zu machen, gibt es eine Bibliothek. Diesmal nutzt du hierfür jedoch nicht den Bibliotheksverwalter, sondern bindest die Bibliothek manuell ein. [Lade sie dir zunächst hier herunter.](#)

Wähle nun im Menü der Arduino IDE den Punkt **Sketch -> Bibliothek einbinden -> .ZIP Bibliothek hinzufügen**

Für einen ersten Test der Tastatur verwendest du nun den Beispiel-Sketch, der in der Bibliothek mitgeliefert wird. Öffne diesen unter **Datei -> Beispiele -> Keypad -> CustomKeypad**

Bevor wir einen Blick auf den Code werfen, lade den Sketch zunächst auf deinen Arduino. Öffne nun den Seriellen Monitor und drücke eine der Tasten auf der Tastatur. Erscheint das entsprechende Zeichen in der Ausgabe? Wenn ja, perfekt.

DER BEISPIEL-SKETCH

Lass uns einen ganz kurzen Blick auf diesen Sketch werfen. Der meiste Code sind Funktionen der Bibliothek Keypad, aber zwei Dinge sind besonders interessant. Zunächst siehst du im Code die Anzahl der Tasten je Reihe und Spalte sowie eine Matrix, die die Werte bzw. Zeichen der Tastatur bestimmt:

```
const byte ROWS = 4;
const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
```

Das sind die Zeichen, die du auch auf der Tastatur selbst findest. Solltest du einmal andere Zeichen verarbeiten wollen, könntest du diese hier eintragen und damit einer Taste auf der Folientastatur zuweisen.

Der Anschluss der Tastatur am Arduino verbirgt sich in diesen beiden Variablen:

```
byte rowPins[ROWS] = {9, 8, 7, 6};  
byte colPins[COLS] = {5, 4, 3, 2};
```

Das sind die Pins, die du auch im Anschlussplan oben findest. Wenn du einmal einen dieser Pins unbedingt für ein anderes Bauteil verwenden musst, kannst du hier der Tastatur einen abweichenden Digitalpin zuweisen.

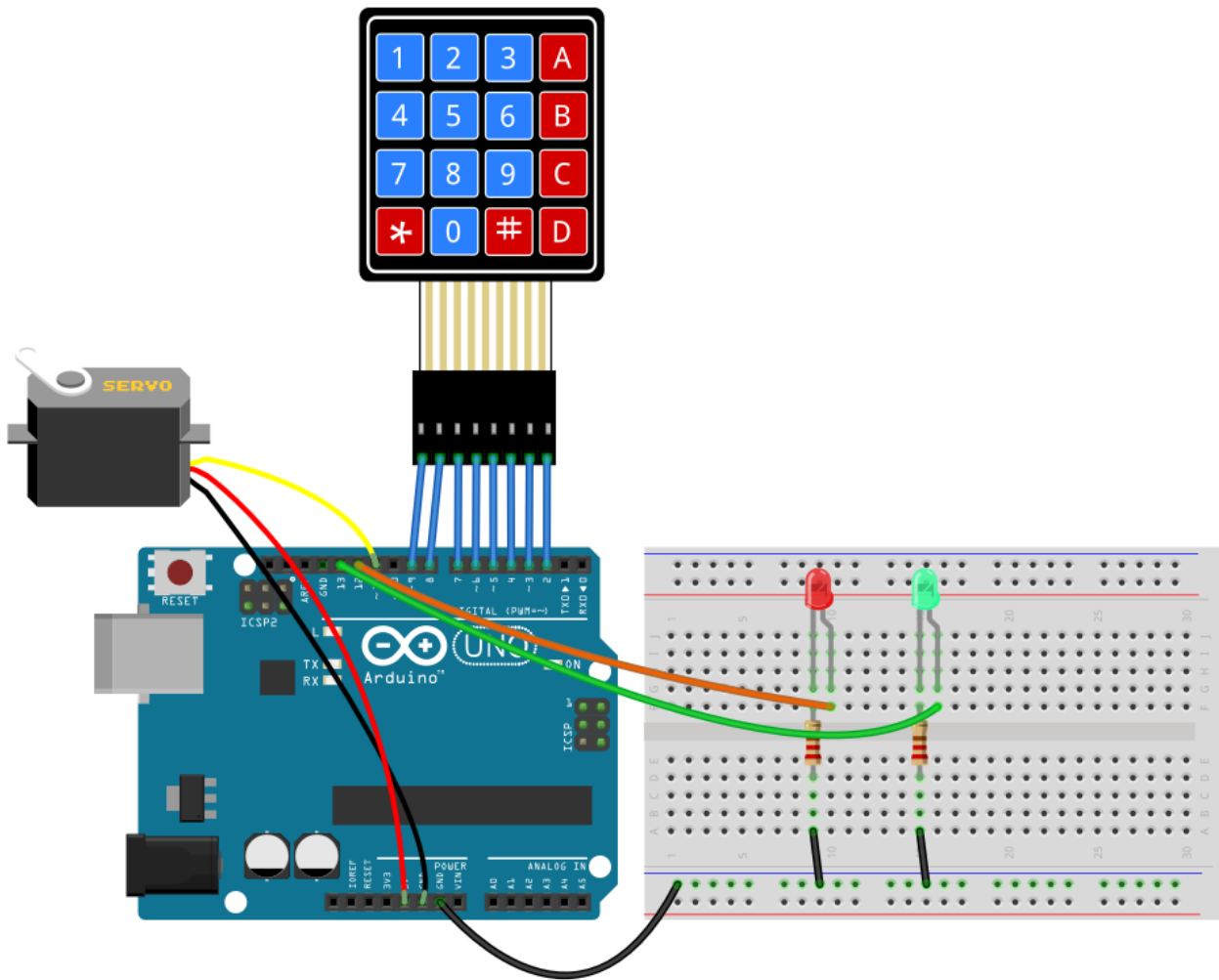
Außerdem wird mit diesen Parametern das Objekt customKeypad erzeugt:

```
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins,  
colPins, ROWS, COLS);
```

Und weiter geht's – als nächstes installierst du zwei LEDs sowie den Servo-Motor und sorgst dafür, dass dieser sich nur nach der Eingabe des richtigen Codes bewegt.

DAS SCHLOSS AUFBAUEN UND PROGRAMMIEREN

Verbinde als erstes deinen Servo-Motor mit dem Arduino. Du kannst ihn entweder auf deinem Breadboard installieren und von dort aus verbinden – oder du steckst die Kabel des Servos direkt in die Pins GND, 5V und 11, so wie auf dieser Skizze.



Die Reihenfolge der Anschlüsse GND, VCC und Signal können sich je nach Fabrikat des Motors unterscheiden. Bitte achte hierauf beim Anschluss.

Hinzu kommen noch zwei LEDs samt zwei Vorwiderständen mit je 220Ω . Schließe diese an die Digitalpins 12 und 13 an. Mit diesen zeigst du später an, ob der Code richtig oder falsch eingegeben wurde.

DER SKETCH FÜR DEIN CODE-SCHLOSS

Damit kommen wir schon zum Kern des Projekts. Der Servo soll sich nur bewegen, wenn jemand den richtigen Code per Tastatur eingibt. Dann dreht sich der Zeiger zur Seite und gibt z.B. den Deckel einer Box frei. Wie du das Gehäuse der Box

gestaltest, ist natürlich dir überlassen.

Zu Beginn des Sketchs bindest du wie gewohnt zunächst die benötigten Bibliotheken ein:

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

Anschließend folgen die Informationen für das Keypad, die du schon kennst:

```
const byte ROWS = 4;
const byte COLS = 4;
```

```
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
```

```
byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins,
colPins, ROWS, COLS);
```

Nun benötigst du drei Variablen bzw. Konstanten, die die gedrückte Taste **key**, den gesamten eingegebenen Code **inputCode** sowie das von dir hinterlegte Passwort **code** beinhalten. Gegen letzteres wirst du später die Eingabe prüfen. Wenn du später deinen eigenen Code hinterlegen möchtest, kannst du das hier tun.

```
char key;
String inputCode;
const String code = "1103A"; //Der von dir festgelegte Code
```

Zuletzt fehlen noch ein Objekt für den Servo-Motor und die

beiden Variablen für die Anschlüsse der LEDs.

```
Servo servo;  
int redLED = 12;  
int greenLED = 13;
```

DIE SETUP-FUNKTION

Hier gibt es sicherlich nichts Neues für dich. Du startest den Seriellen Monitor, legst die **pinMode** für die LEDs fest und den Anschluss-Pin des Servo-Motors.

```
void setup() {  
  
    Serial.begin(9600);  
    pinMode(redLED, OUTPUT);  
    pinMode(greenLED, OUTPUT);  
    servo.attach(11);  
}
```

DER LOOP

Als erstes benötigst du eine Variable, in der du den Wert der Taste speicherst, die gerade gedrückt wurde. Das ist hier die Variable **key** mit dem Typ **char** für Character. Den Wert speicherst du mithilfe der Funktion **customKeypad.getKey()**.

```
char key = customKeypad.getKey();
```

Nun folgt eine Reihe von bedingten Anweisungen. Das Code-Schloss verwendet zwei Tasten, um das Schloss zu verriegeln (mit der Taste *****) oder die Eingabe zu prüfen (mit **#**). Diese Tasten darfst du also nicht in deinem hinterlegten Code verwenden. Die erste If-Abfrage "horcht" auf die Taste ***** und verriegelt das Schloss, indem sie den Servo auf einen Winkel

von 90° stellt. Natürlich kannst du diesen Winkel nach deinen Bedürfnissen ändern.

Zusätzlich schaltet sie die rote LED an (und die grüne aus) und leert die Variable **inputCode**, damit hier für die nächste Eingabe keine "alten" Zeichen mehr zu finden sind.

```
if (key == '*') {  
    inputCode = "";  
    Serial.println("Schloss verriegelt.");  
    delay(1000);  
    servo.write(90);  
    digitalWrite(greenLED, LOW);  
    digitalWrite(redLED, HIGH);  
}
```

Die zweite Abfrage folgt gleich darauf mit einem **else if** und wird angesteuert, wenn die Taste # gedrückt wurde. In diesem Fall folgen wiederum zwei weitere Abfragen – eine, die ausgeführt wird, wenn der Code korrekt eingegeben wurde und die andere, falls dem nicht so ist.

```
} else if (key == '#') {  
    if (inputCode == code) {  
        Serial.println("Der Code ist korrekt. Öffne das  
Schloss...");  
        digitalWrite(greenLED, HIGH);  
        digitalWrite(redLED, LOW);  
        servo.write(0);  
    } else {  
        Serial.println("Der Code ist falsch!");  
        digitalWrite(greenLED, LOW);  
        digitalWrite(redLED, HIGH);  
    }  
}
```

Falls der Code stimmt, also **inputCode == code** zutrifft, leuchtet die grüne LED und der Servo steuert auf 0°. Bei einer inkorrekten Eingabe, leuchtet entsprechend die rote LED – und nichts bewegt sich. In beiden Fällen wird die Variable für den

einggegebenen Code wieder geleert:

```
inputCode = "";
```

Jetzt hast du alles, um das Schloss auf Tastendruck zu verriegeln. Außerdem um zu prüfen, ob der Code richtig oder falsch ist und entsprechend zu reagieren. Aber wie wird die Variable **inputCode** eigentlich befüllt?

Das geschieht in einem letzten **else** – die Anweisung, die bei allen Tasten außer * und # ausgeführt wird. Hier fügst du der Variablen **inputCode** einfach das zuletzt gedrückte Zeichen hinzu:

```
} else {  
    inputCode += key;  
    Serial.println(inputCode);  
}
```

Hier nun der gesamte Sketch – viel Spaß mit deinem neuen Code-Schloss! :=

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

```
const byte ROWS = 4;  
const byte COLS = 4;
```

```
char hexaKeys[ROWS][COLS] = {  
    {'1', '2', '3', 'A'},  
    {'4', '5', '6', 'B'},  
    {'7', '8', '9', 'C'},  
    {'*', '0', '#', 'D'}  
};
```

```
byte rowPins[ROWS] = {9, 8, 7, 6};  
byte colPins[COLS] = {5, 4, 3, 2};  
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins,  
colPins, ROWS, COLS);
```

```

char key;
String inputCode;
const String code = "1103A"; //Der von dir festgelegte Code

Servo servo;
int redLED = 12;
int greenLED = 13;

void setup() {

    Serial.begin(9600);
    pinMode(redLED, OUTPUT);
    pinMode(greenLED, OUTPUT);
    servo.attach(11);
}

void loop() {

    char key = customKeypad.getKey();

    if (key) {

        if (key == '*') {
            inputCode = "";
            Serial.println("Schloss verriegelt.");
            delay(1000);
            servo.write(90);
            digitalWrite(greenLED, LOW);
            digitalWrite(redLED, HIGH);

        } else if (key == '#') {
            if (inputCode == code) {
                Serial.println("Der Code ist korrekt. Öffne das
Schloss...");
                digitalWrite(greenLED, HIGH);
                digitalWrite(redLED, LOW);
                servo.write(0);

            } else {
                Serial.println("Der Code ist falsch!");
            }
        }
    }
}

```

```
        digitalWrite(greenLED, LOW);
        digitalWrite(redLED, HIGH);
    }

    inputCode = "";

    } else {
        inputCode += key;
        Serial.println(inputCode);
    }
}
}
```

Statt eines Codes kannst du natürlich auch andere Methoden zur Zugangsbegrenzung verwenden. Die Grundlagen zu RFID. lernst du in unserem [Arduino RFID Tutorial](#).