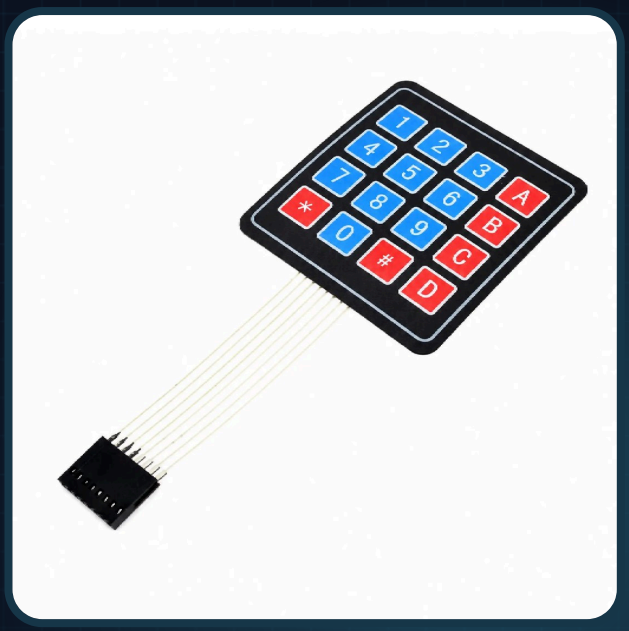
EINGABE · 16 TASTEN

Keypad 4×4

16 Tasten auf acht Anschlüssen – für Codeschlösser, Menüs und Bedienpanels. Erstes Codeschloss in rund 15 Minuten mit dem UNO R3.



TASTEN	16 (0–9, A–D, *, #)
ANSCHLÜSSE	8
BAUHÖHE	ca. 1 mm
ELEKTRONIK	passiv

01 **Verkabeln**
8 Leitungen zum UNO

02 **Einrichten**
IDE, Treiber, Bibliothek

03 **Hochladen**
Beispiel-Sketch

04 **Eingeben**
Codeschloss

01 Auf einen Blick

Die Folientastatur ist als Matrix aus vier Zeilen und vier Spalten aufgebaut. Der Mikrocontroller schaltet die Zeilen nacheinander und liest die Spalten – aus der Kreuzung ergibt sich die Taste. Das spart die Hälfte der Pins gegenüber 16 einzelnen Tastern.

Technische Daten

Tasten	**16: 0–9, A–D, * und #**
Prinzip	**4 × 4 Matrix, passiv**
Anschluss	**8-polige Buchse, 2,54 mm**
Kontaktlast	**max. 35 V DC / 100 mA**
Logik	**3,3 V und 5 V**
Rückseite	**selbstklebend**
Bauhöhe	**knapp 1 mm**

Lieferumfang

1 × **4×4-Folientastatur (16 Tasten) mit 8-poligem Anschlusskabel**

Keine Elektronik
Keine Widerstände, keine Dioden – die internen Pull-ups des Mikrocontrollers genügen.

Eine Taste zur Zeit
Ohne Dioden lassen sich mehrere gleichzeitig gedrückte Tasten nicht sicher unterscheiden.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
R1	Zeile 1	3,3 / 5 V	Tasten 1 2 3 A – an D9.
R2	Zeile 2	3,3 / 5 V	Tasten 4 5 6 B – an D8.
R3	Zeile 3	3,3 / 5 V	Tasten 7 8 9 C – an D7.
R4	Zeile 4	3,3 / 5 V	Tasten * 0 # D – an D6.
C1	Spalte 1	3,3 / 5 V	Tasten 1 4 7 * – an D5.
C2	Spalte 2	3,3 / 5 V	Tasten 2 5 8 0 – an D4.
C3	Spalte 3	3,3 / 5 V	Tasten 3 6 9 # – an D3.
C4	Spalte 4	3,3 / 5 V	Tasten A B C D – an D2.

Pin 1 ist links, wenn die Tasten zu dir zeigen: R1 · R2 · R3 · R4 · C1 · C2 · C3 · C4.

Was du zusätzlich brauchst

UNO R3 mit USB-C
ATmega328P, CH340G-USB-Wandler
`MKR-UNO-R3-013`

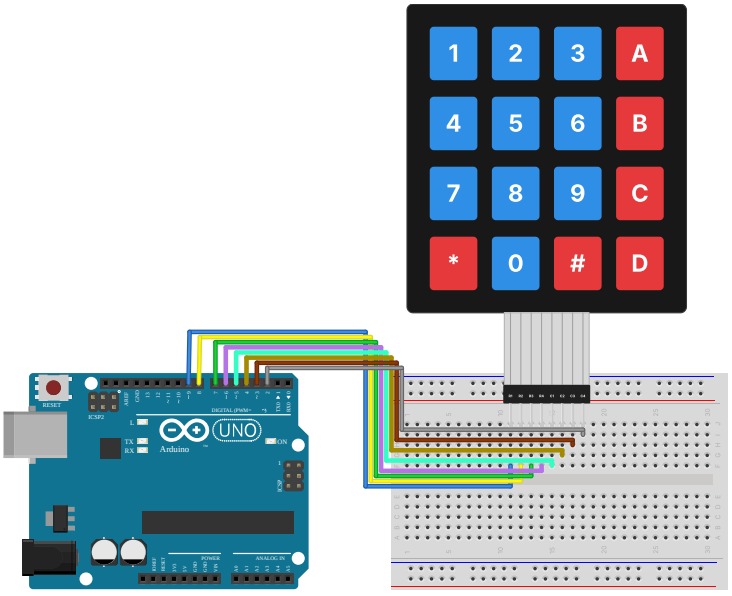
Jumper-Kabel male–male
40 Stück, 10 cm
`MKR-JUMPER-MM-10CM`

USB-C-Datenkabel
Kein reines Ladekabel – es muss Daten übertragen

02 Verkabeln

Acht Leitungen, keine Versorgung: Pin 1 bis 8 der Tastatur gehen der Reihe nach an D9 bis D2.

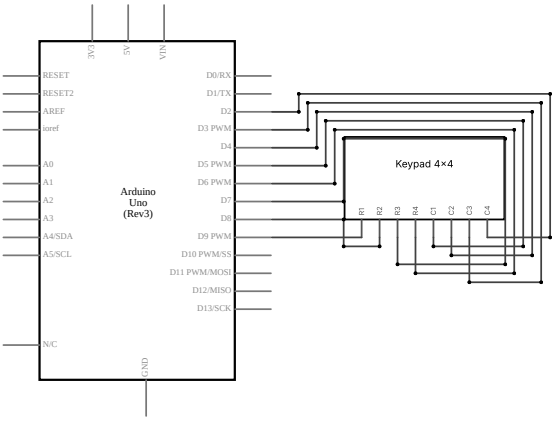
STECKPLATINEN-ANSICHT



Verbindungen

MODUL		UNO R3	LEITUNG
R1	→	D9	Zeile 1
R2	→	D8	Zeile 2
R3	→	D7	Zeile 3
R4	→	D6	Zeile 4
C1	→	D5	Spalte 1
C2	→	D4	Spalte 2
C3	→	D3	Spalte 3
C4	→	D2	Spalte 4

SCHALTPLAN



Verdrahtung erstellt mit Fritzing

Buchse am Kabel
Die Tastatur endet in einer Buchsenleiste: male–male-Jumper direkt hineinstecken oder eine Stiftleiste aufs Steckbrett setzen.

03 Einrichten & Hochladen

Schritte 1 bis 3 machst du nur ein einziges Mal – danach ist dein UNO R3 für alle weiteren Projekte startklar.

1 Arduino IDE installieren

Lade die aktuelle Arduino IDE 2 von `arduino.cc/en/software` herunter und installiere sie.

2 Treiber für den CH340G prüfen

Der UNO R3 nutzt den CH340G als USB-Seriell-Wandler. Aktuelle Systeme erkennen ihn meist automatisch. Taucht in Schritt 3 kein Port auf, installiere den Treiber „CH341SER“ des Chipherstellers WCH (`wch-ic.com`).

3 Board und Port wählen

UNO per USB-C anschließen. Dann `Werkzeuge → Board → Arduino AVR Boards → Arduino Uno` wählen und unter `Werkzeuge → Port` den neu aufgetauchten COM-Port (Windows) bzw. `/dev/ttyUSB0` (Linux) / `/dev/cu.wchusbserial…` (macOS).

4 Bibliothek installieren

Bibliotheksverwalter öffnen (`Strg` + `Umschalt` + `I`), nach **Keypad** von Mark Stanley, Alexander Brevig suchen und installieren.

5 Sketch hochladen

Neuen Sketch anlegen, den Code von Seite 5 einfügen und mit dem Pfeil-Symbol hochladen (`Strg` + `U`). Unten erscheint „Hochladen abgeschlossen“.

6 Seriellen Monitor öffnen

`Strg` + `Umschalt` + `M`, unten rechts **9600 Baud** einstellen. Jede gedrückte Taste erscheint mit der bisherigen Eingabe.

BEISPIELAUSGABE · SERIELLER MONITOR · 9600 BAUD

```
Code eingeben, mit # bestaetigen, * loescht.
Taste 1 Eingabe: 1
Taste 2 Eingabe: 12
Taste 3 Eingabe: 123
Taste 4 Eingabe: 1234
-> Code richtig!
```

Schnelltest

Tippe 1 2 3 4 und dann #: „Code richtig!“. Jede andere Folge ergibt „Falscher Code“, * löscht die Eingabe.

04 Der erste Sketch

Ein vollständiges Codeschloss: Tasten sammeln, mit # prüfen, mit * löschen.

MAKEROO_KEYPAD_Start.ino

```
// MAKEROO · Keypad 4x4 · Codeschloss
#include <Keypad.h>

const byte ZEILEN = 4, SPALTEN = 4;
char tasten[ZEILEN][SPALTEN] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}};
byte zeilenPins[ZEILEN] = {9, 8, 7, 6};         // R1 bis R4
byte spaltenPins[SPALTEN] = {5, 4, 3, 2};       // C1 bis C4
Keypad keypad(makeKeymap(tasten), zeilenPins, spaltenPins, ZEILEN, SPALTEN);

const String CODE = "1234";                     // dein Geheimcode
String eingabe = "";

void setup() {
  Serial.begin(9600);
  Serial.println(F("Code eingeben, mit # bestaetigen, * loescht."));
}

void loop() {
  char t = keypad.getKey();
  if (!t) return;                               // keine Taste gedrueckt
  if (t == '*') {
    eingabe = "";
    Serial.println(F("  geloescht"));
  } else if (t == '#') {
    Serial.println(eingabe == CODE ? F("  -> Code richtig!") : F("  -> Falscher Code"));
    eingabe = "";
  } else {
    eingabe += t;
    Serial.print(F("Taste "));
    Serial.print(t);
    Serial.print(F("   Eingabe: "));
    Serial.println(eingabe);
  }
}
```

Zeile 5-9
Tastenbelegung
Spiegelt den Aufdruck – so liefert die Bibliothek direkt das richtige Zeichen.

Zeile 10-11
Pins
Zeilen an D9–D6, Spalten an D5–D2. Andere Pins? Nur hier ändern.

Zeile 14
Geheimcode
Beliebig lang, auch mit Buchstaben A–D.

Zeile 23-24
Taste lesen
Liefert das Zeichen oder 0, wenn nichts gedrückt ist – blockiert nie.

Zeile 28-30
Prüfen
Vergleicht die Eingabe mit dem Code und setzt sie zurück.

Geprüft: kompiliert für Arduino Uno mit Keypad 3.1.1 (5.1 kB Flash = 16 %, 359 Byte RAM).

05 Wenn es nicht klappt

SYMPTOM	LÖSUNG
Falsche Zeichen	Kabel einen Pin versetzt oder Reihenfolge gedreht. Pin 1 (R1) gehört an D9.
Ganze Reihe reagiert nicht	Eine Leitung sitzt locker – Jumper fester in die Buchse stecken.
Tasten doppelt	Prellen: Die Bibliothek entprellt mit 10 ms; mit `keypad.setDebounceTime(50);` in `setup()` erhöhen.
A, B, C, D vertauscht mit *, 0, #	Zeilen- und Spaltenpins vertauscht – die beiden Arrays im Sketch tauschen.
Nur wirre Zeichen im Monitor	Baudrate im seriellen Monitor auf 9600 stellen.

06 Wie geht's weiter?

Mit ESP32 & ESPHome

Am ESP32 z. B. Zeilen an **GPIO13, 12, 14, 27**, Spalten an **GPIO26, 25, 33, 32**. In ESPHome:

```
matrix_keypad:
  id: tastatur
  rows:
    - pin: GPIO13  # bis GPIO27
  columns:
    - pin: GPIO26  # bis GPIO32
  keys: "123A456B789C*0#D"
key_collector:
  - source_id: tastatur
    end_keys: "#"
```

Projektideen

Codeschloss
Mit Relais einen Türöffner schalten

Taschenrechner
Ergebnis auf dem LCD

Menü
A–D als Funktionstasten

Tresor
Servo als Riegel

Passt dazu

LCD1602 mit I2C
Zwei Zeilen Klartext
`MKR-LCD1602-I2C`

Relais KY-019
Verbraucher schalten
`MKR-KY019-034`

2x SG90 Servo 360°
Bewegung per PWM
`MKR-SG90-360G-003`

0,96" OLED-Display
Werte direkt anzeigen
`MKR-0.96-OLED-015`

Sicherheitshinweise. Elektronikmodul für Ausbildungs-, Entwicklungs- und Experimentierzwecke. Nur mit Kleinspannung betreiben, niemals an Netzspannung. Nicht geeignet für Kinder unter 14 Jahren; Nutzung nur unter Aufsicht. Elektroaltgeräte gehören nicht in den Hausmüll – Rückgabe kostenlos bei kommunalen Sammelstellen.

Verantwortlicher Wirtschaftsakteur (GPSR): Alexander Prez · makeroo · Hinterm Bahnhof 4 · 90513 Zirndorf · support@makeroo.de
Fragen? Schreib uns mit einem Foto deiner Verkabelung.
Arduino ist eine Marke der Arduino SA und dient nur der Kompatibilitätsangabe.