

# syt EK - Taschenrechner

---

@author: Sebastian Sailer

@version: 2023-06-15

## 1. Einführung

---

Für Ein und Ausgeben kann man bei Microcontrollern verschiedene Bauteile verwenden. Es gibt verschiedene arten von In- & Outputs. Als Eingab kann man z.B.: ein Matrix Keyboard Modul nehmen.

## 2. Projektbeschreibung

Es soll ein Taschenrechner mittels einem "4\*4 Matrix Keyboard Modul" umgesetzt werde. Anfangs war geplant, das Ergebnis und die Zwischenschritte auf einem LCD Display, oder einem "7-Segment-Display" angezeigt werden. Allerdings

habe ich zu wenige Pins um das zu realisieren.

## 3. Arbeitsschritte

**Code:**

```

// Imports
#include <Keypad.h>

// gibt an wieviel Zeilen/Reihen das Numpad hat
const byte ROWS = 4;
const byte COLS = 4;

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

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

// Keypad Objekt erstellen
Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS);

void setup() {
    // Setup serial monitor
    Serial.begin(9600);
}

void loop() {
    //deklaration der Werte
    char wert1 = '\0';
    char customKey = '\0';
    char wert2 = '\0';

    // Eingabe von wert1
    while (wert1 == '\0') {
        wert1 = customKeypad.getKey();
        delay(50);
    }

    // Eingabe von Operator (customKey)
    while (customKey == '\0') {
        customKey = customKeypad.getKey();
        delay(50);
    }
}

```

```

// Eingabe von wert2
while (wert2 == '\0') {
    wert2 = customKeypad.getKey();
    delay(50);
}
//aufrufen der Methode rechnung
rechnung(customKey, wert1, wert2);
}

void rechnung(char customKey, char wert1, char wert2) {
    int ergebnis = 0; //deklaraton und initialisierung von ergebnis

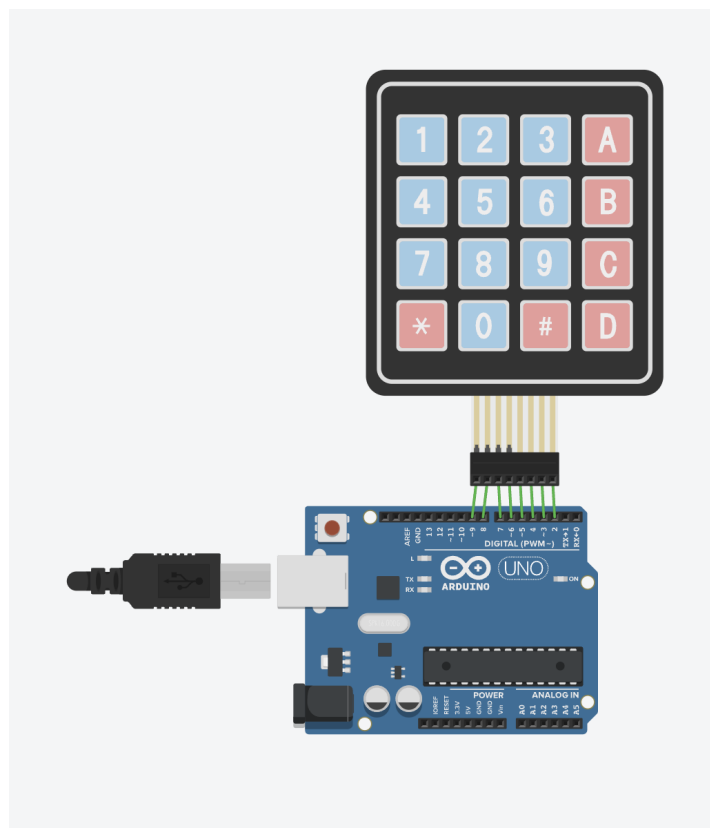
    if (customKey) {
        //Im Falle des ausgewählten Zeichens A soll addiert werden
        if (customKey == 'A') {
            ergebnis = atoi(String(wert1).c_str()) +
atoi(String(wert2).c_str());
            Serial.println("Summe: " + String(ergebnis));
        }
        //Im Falle des ausgewählten Zeichens B soll subtrahiert werden
        else if (customKey == 'B') {
            ergebnis = atoi(String(wert1).c_str()) -
atoi(String(wert2).c_str());
            Serial.println("Subtraktion: " + String(ergebnis));
        }
        //Im Falle des ausgewählten Zeichens C soll dividiert werden
        else if (customKey == 'C') {
            ergebnis = atoi(String(wert1).c_str()) /
atoi(String(wert2).c_str());
            Serial.println("Division: " + String(ergebnis));
        }
        //Im Falle des ausgewählten Zeichens D soll multipliziert werden
        else if (customKey == 'D') {
            ergebnis = atoi(String(wert1).c_str()) *
atoi(String(wert2).c_str());
            Serial.println("Multiplikation: " + String(ergebnis));
        }
        //ist es eine weitere Zahl, soll diese wieder ausgegeben werden
        else {
            Serial.println(customKey);
        }
    }
}
}

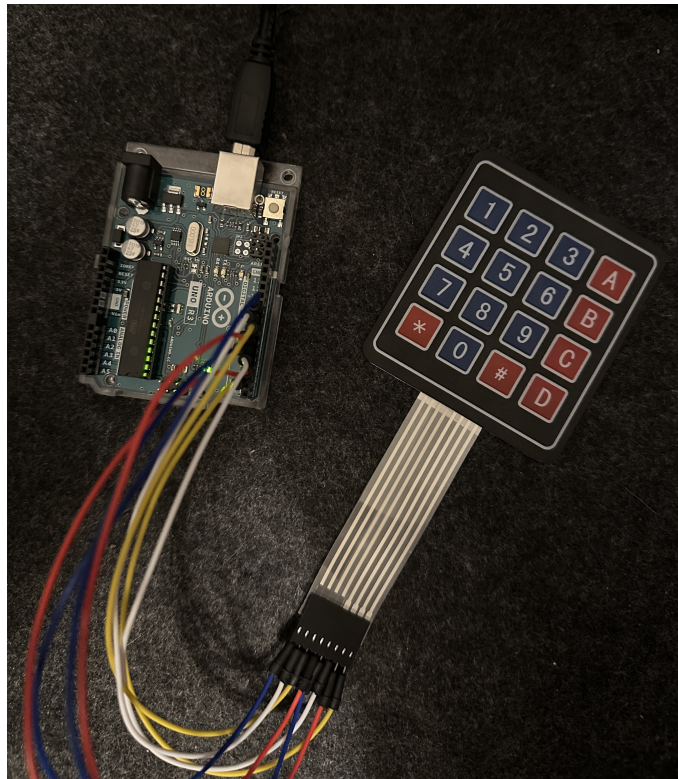
```

**Ausgabe:**

```
Output  Serial Monitor ×
Message (Enter to send message)
Summe: 7
Subtraktion: 2
Division: 0
Multiplikation: 2
Summe: 10
Multiplikation: 40
Summe: 7
Multiplikation: 0
Subtraktion: -7
Summe: 6
Subtraktion: -3
```

## Bilder:





## 4. Zusammenfassung

Insgesamt war es ein sehr lustiges und simples Projekt, welches allerdings einiges an Pins des Arduino beansprucht. Ich würde es allerdings noch gerne weiter ausführen mit z.B.: einem LCD Display etc.

Um mir wieder in Erinnerung zu rufen, wie man in C programmiert und wie man das Keyboard ansteuert hab ich mir die Codes des 2. Links angeschaut, und anschließend weiter geschrieben/erweitert.

## Quellen:

[Using Keypads with Arduino - Build an Electronic Lock](#)

[Arduino Lektion 68: 4x4 Matrix Tastatur - Technik Blog](#)

[Using Keypads with Arduino - Build an Electronic Lock - YouTube](#)