

SBC ButtonMatrix Bedienungsanleitung MCU Extension 4x4 16-Key

Sehr geehrter Kunde,

vielen Dank, dass Sie sich für unser Produkt entschieden haben.

Im Folgenden haben wir aufgelistet, was bei der Inbetriebnahme zu beachten ist:

Verwendung mit einem Arduino Schritt 1 – Anschließen der Matrix

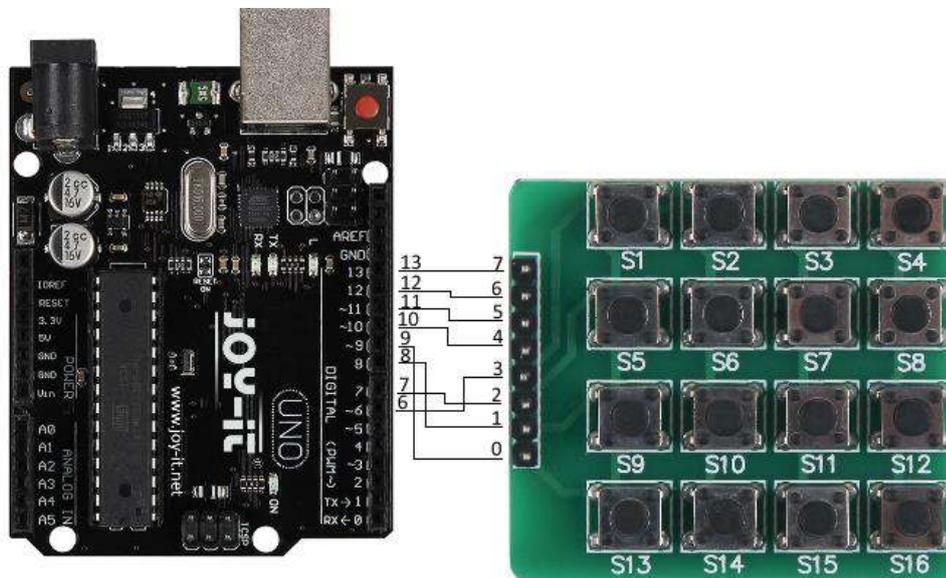


Bild 1: Schaltplan Arduino Uno & 4x4 ButtonMatrix

PIN Nr. Arduino	PIN Nr. ButtonMatrix
13	7
12	6
11	5
10	4
9	0
8	1
7	2
6	3

Tabelle 1: PIN-Verbindung zwischen Arduino und ButtonMatrix

Schließen Sie die ButtonMatrix, wie in Bild 1 bzw. in Tabelle 1 zu sehen, an die digitalen Pins (6 – 13) des Arduino an.

Schritt 2 – Installation der Matrix

Nachfolgend können Sie ein funktionsfähiges Codebeispiel entnehmen und auf Ihren Arduino übertragen.

Die Funktion der vorhandenen Knöpfe der ButtonMatrix können Sie in der **knopfDruck** Funktion nach Ihren Wünschen erweitern.

```
int reihe[]={6,7,8,9};
int spalte[]={10,11,12,13};
int col_scan;

void setup()
{
  Serial.begin(9600);
  for(int i=0;i<=3;i++)
  {
    //Initialisierung der PINs
    pinMode(reihe[i],OUTPUT);
    pinMode(spalte[i],INPUT);
    digitalWrite(spalte[i],HIGH);
  }
}

void loop()
{
  //Suche nach gedrücktem Knopf
  for(int i=0; i<=3; i++)
  {
    digitalWrite(reihe[0],HIGH);
    digitalWrite(reihe[1],HIGH);
    digitalWrite(reihe[2],HIGH);
    digitalWrite(reihe[3],HIGH);
    digitalWrite(reihe[i],LOW);

    for(int j=0; j<=3; j++)
    {
      col_scan=digitalRead(spalte[j]);
      if(col_scan==LOW)
      {
        //Wenn gedrückter Knopf erkannt, führe knopfDruck aus
        knopfDruck(i,j);
        delay(300);
      }
    }
  }
}
}
```

Code 1: Programmierung der ButtonMatrix innerhalb des Arduinos (Teil 1)

```
void knopfDruck(int i, int j)
{
  if(i==0&&j==0) //Knopf S1 gedrückt
  Serial.println("S1");
  if(i==0&&j==1) //Knopf S2 gedrückt
  Serial.println("S2");
  if(i==0&&j==2) //Knopf S3 gedrückt
  Serial.println("S3");
  if(i==0&&j==3) //Knopf S4 gedrückt
  Serial.println("S4");
  if(i==1&&j==0) //Knopf S5 gedrückt
  Serial.println("S5");
  if(i==1&&j==1) //Knopf S6 gedrückt
  Serial.println("S6");
  if(i==1&&j==2) //Knopf S7 gedrückt
  Serial.println("S7");
  if(i==1&&j==3) //Knopf S8 gedrückt
  Serial.println("S8");
  if(i==2&&j==0) //Knopf S9 gedrückt
  Serial.println("S9");
  if(i==2&&j==1) //Knopf S10 gedrückt
  Serial.println("S10");
  if(i==2&&j==2) //Knopf S11 gedrückt
  Serial.println("S11");
  if(i==2&&j==3) //Knopf S12 gedrückt
  Serial.println("S12");
  if(i==3&&j==0) //Knopf S13 gedrückt
  Serial.println("S13");
  if(i==3&&j==1) //Knopf S14 gedrückt
  Serial.println("S14");
  if(i==3&&j==2) //Knopf S15 gedrückt
  Serial.println("S15");
  if(i==3&&j==3) //Knopf S16 gedrückt
  Serial.println("S16");
}
```

Code 2: Programmierung der ButtonMatrix innerhalb des Arduinos (Teil 2)

Verwendung mit einem Raspberry Pi

Schritt 1 – Anschließen der Matrix

Schließen Sie die Matrix, wie im folgenden Bild 2, bzw. in folgender Tabelle 2, zu sehen, an die PINs des Raspberry Pis an.

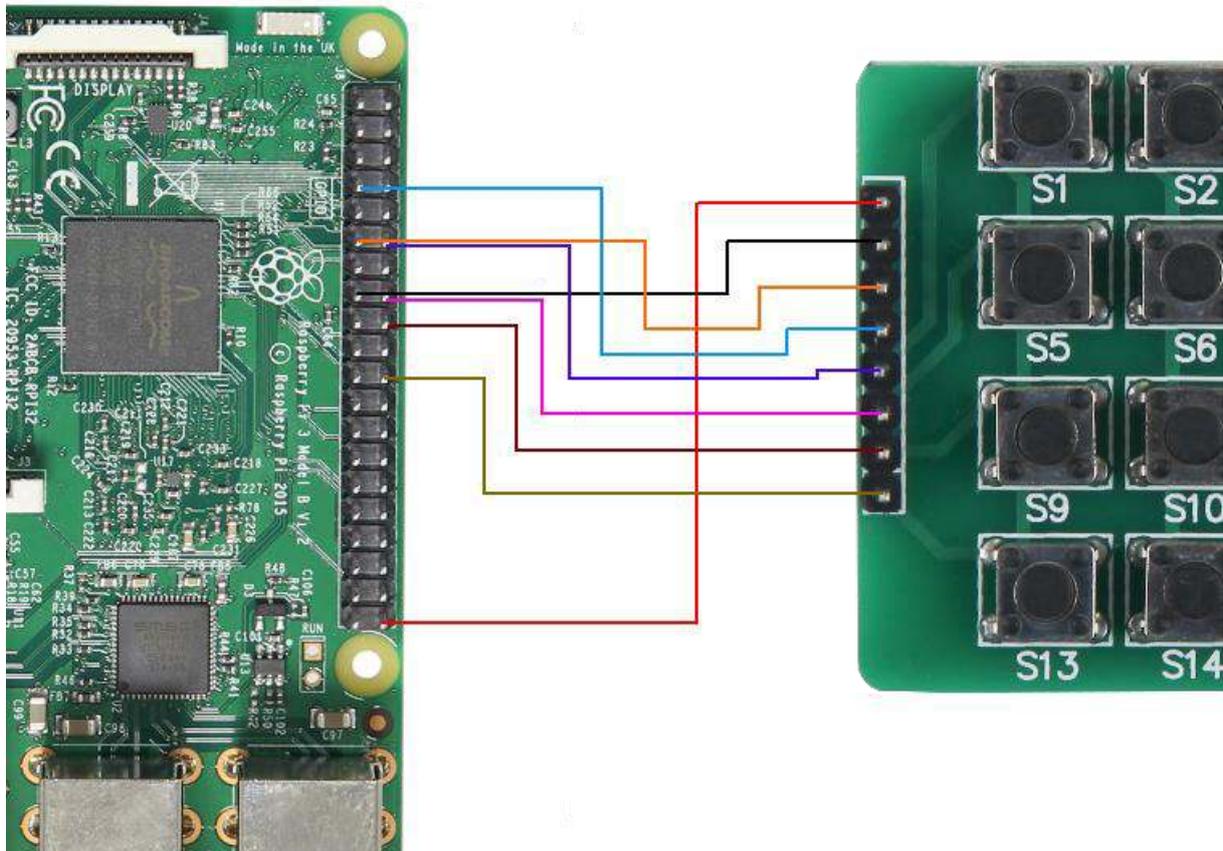


Bild 2: Verbindung zwischen Raspberry Pi und Matrix

Raspberry Pi PIN	Matrix PIN
PIN 40 (BCM 21)	1
PIN 15 (BCM 22)	2
PIN 11 (BCM 17)	3
PIN 7 (BCM 4)	4
PIN 12 (BCM 18)	5
PIN 16 (BCM 23)	6
PIN 18 (BCM 24)	7
PIN 22 (BCM 25)	8

Tabelle 2: PIN-Verbindung zwischen Raspberry Pi und Matrix

Schritt 2 – Installation der Software

Sollten Sie bereits ein aktuelles Raspbian-System auf Ihrem Raspberry verwenden, so können Sie diesen Schritt überspringen und sofort mit Schritt 3 fortfahren.

Installieren Sie auf Ihre SD-Karte mit Hilfe des „Win32 Disk Imager“-Programms das aktuelle Raspbian Image, welches Sie unter dem folgenden [Link](#) zum Download finden.

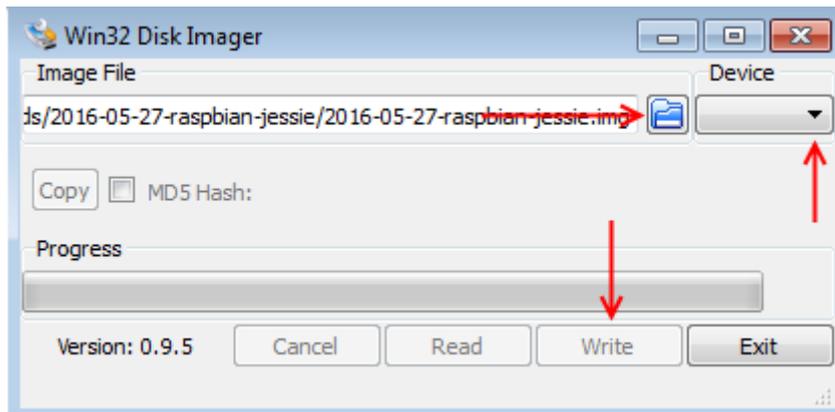


Bild 4: Screenshot des Win32 Disk Imagers

Schritt 3 – Installation der Bibliotheken

Sobald Sie die Installation abgeschlossen und das System gestartet haben, öffnen Sie die Terminal-Konsole und führen Sie folgende Kommandos aus:

```
sudo apt-get install python-pip python-dev build-essential  
sudo pip install RPi.GPIO
```

Terminal 1: Installation der GPIO Bibliothek

```
sudo apt-get install python-imaging
```

Terminal 2: Installation der Python Bibliothek

Schritt 4 – Installation der Matrix

Zunächst muss eine neue Python-Datei erstellt werden.

```
sudo nano matrix.py
```

Terminal 3: Erstellen einer Python Datei

Schreiben Sie das folgende Codebeispiel vollständig in den Editor, der sich nun geöffnet hat.

```
import time
import RPi.GPIO as GPIO

GPIO.setwarnings(False)

class keypad():
    def __init__(self, columnCount = 4):
        GPIO.setmode(GPIO.BCM)
        #Tasteneinstellungen
        if columnCount is 4:
            self.KEYPAD = [
                [1,2,3,4],
                [5,6,7,8],
                [9,10,11,12],
                [13,14,15,16]
            ]
        #PIN-Belegungen
        self.ROW = [18,23,24,25]
        self.COLUMN = [4,17,22,21]
        else:
            return

    def getKey(self):
        #Alle Spalten als Ausgang setzen
        for j in range(len(self.COLUMN)):
            GPIO.setup(self.COLUMN[j], GPIO.OUT)
            GPIO.output(self.COLUMN[j], GPIO.LOW)
        #Alle Reihen als Eingang setzen
        for i in range(len(self.ROW)):
            GPIO.setup(self.ROW[i], GPIO.IN, pull_up_down=GPIO.PUD_UP)
        #Reihen nach gedruckten Kнопfen scannen
        rowVal = -1
        for i in range(len(self.ROW)):
            tmpRead = GPIO.input(self.ROW[i])
            if tmpRead == 0:
                rowVal = i
```

Code 3: Programmierung der Matrix auf einem Raspberry (Teil 1)

```

if rowVal <0 or rowVal >3:
    self.exit()
    return

    for j in range(len(self.COLUMN)):
        GPIO.setup(self.COLUMN[j], GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

    GPIO.setup(self.ROW[rowVal], GPIO.OUT)
    GPIO.output(self.ROW[rowVal], GPIO.HIGH)

    colVal = -1
    for j in range(len(self.COLUMN)):
        tmpRead = GPIO.input(self.COLUMN[j])
        if tmpRead == 1:
            colVal=j

    if colVal <0 or colVal >3:
        self.exit()
        return

#Rueckgabe der gedruckten Taste
self.exit()
return self.KEYPAD[rowVal][colVal]

def exit(self):
#Neuinitialisierung aller Spalten und Reihen
    for i in range(len(self.ROW)):
        GPIO.setup(self.ROW[i], GPIO.IN, pull_up_down=GPIO.PUD_UP)
    for j in range(len(self.COLUMN)):
        GPIO.setup(self.COLUMN[j], GPIO.IN, pull_up_down=GPIO.PUD_UP)

if __name__ == '__main__':
    # Initialisierung des Tastenfelds
    kp = keypad()
    # Dauerschleife zur Abfrage eines Tastendrucks
    while True:
        digit = None
        while digit == None:
            digit = kp.getKey()
        # Ausgabe des Tastendrucks
        print digit
        time.sleep(0.5)

```

Code 4: Programmierung der Matrix auf einem Raspberry (Teil 2)

Die Datei kann mit **Strg+O** gespeichert und der Editor mit **Strg+X** verlassen werden.
Anschließend kann die Datei mit folgendem Befehl ausgeführt und somit getestet werden.
Die Datei kann mit **Strg+C** wieder verlassen werden.

```
sudo python matrix.py
```

Terminal 4: Ausführen der Datei