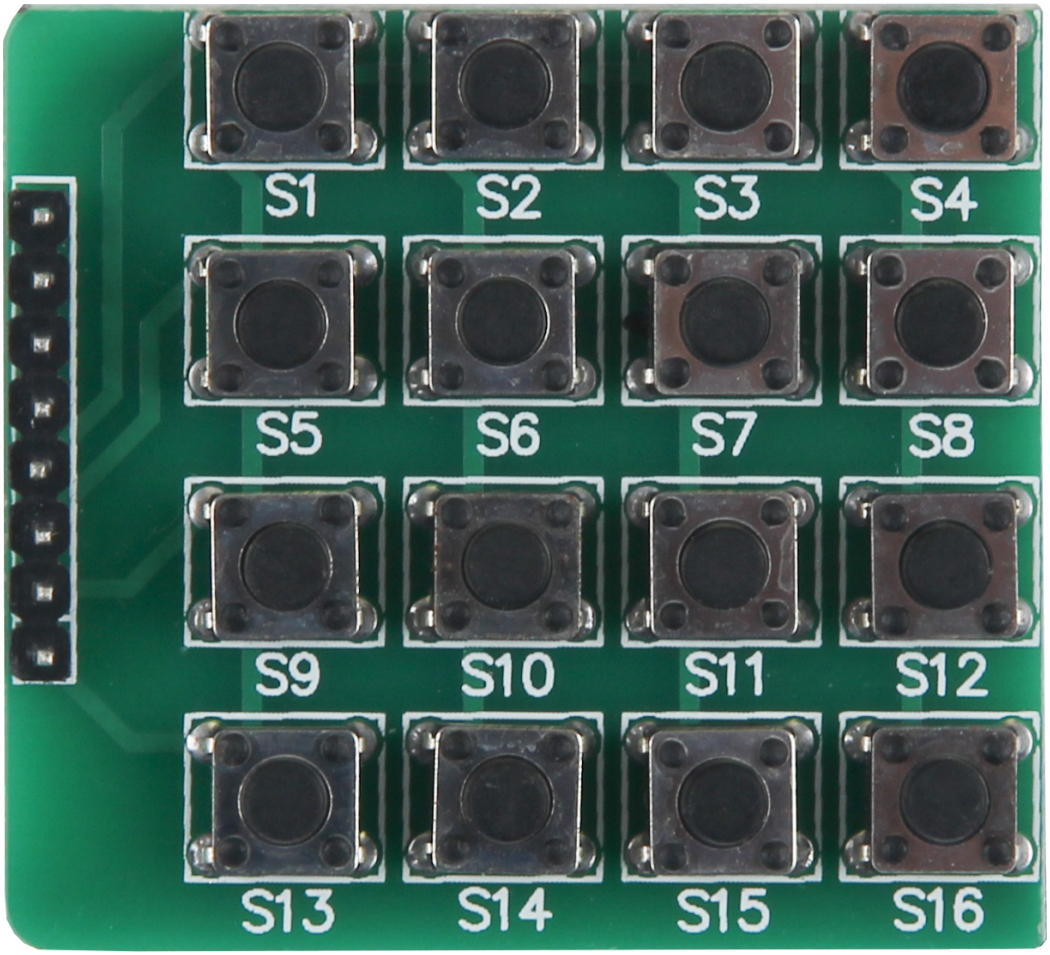
BUTTON MATRIX

4 x 4 Buttons



1. GENERAL INFORMATION

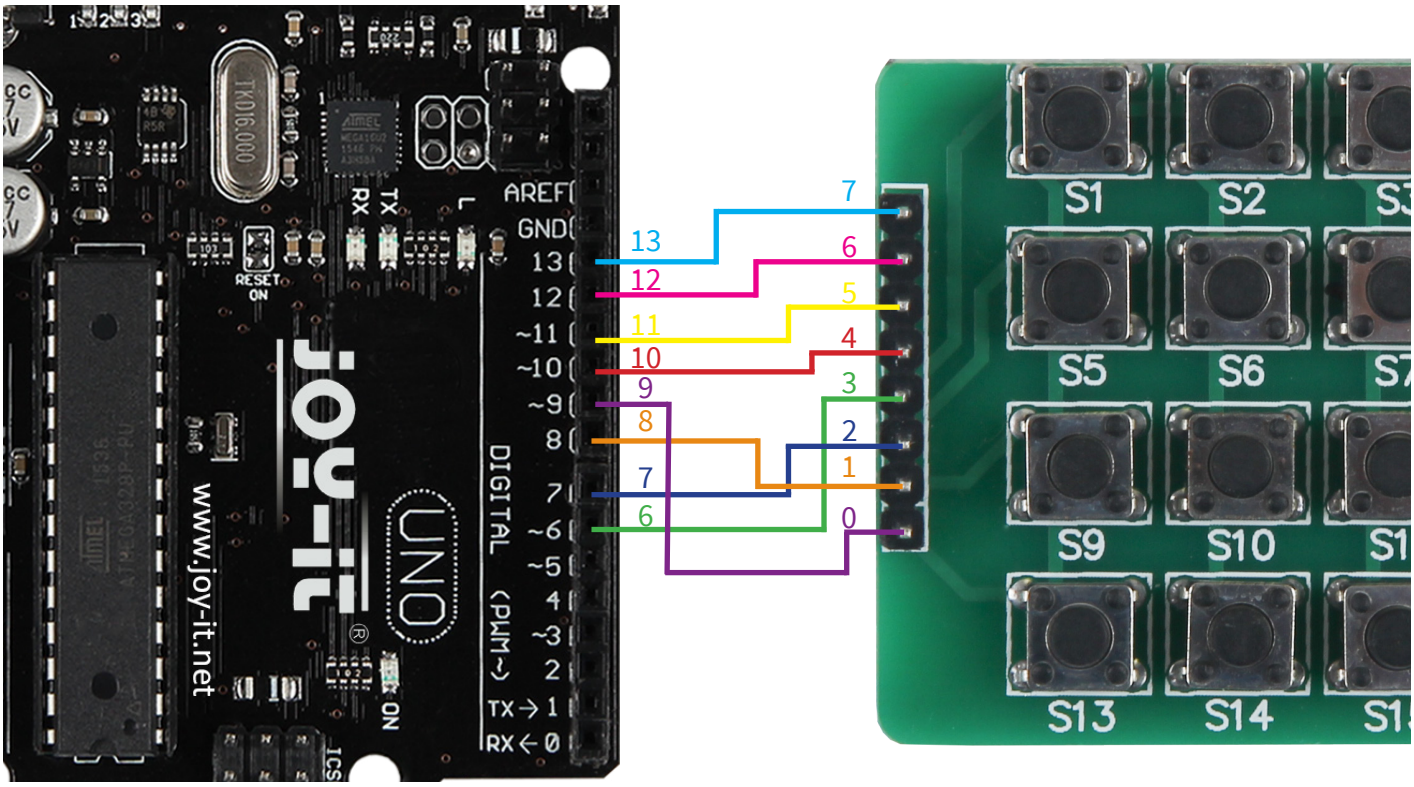
Dear customer,

thank you for choosing our product. In the following, we will show what you should note at the commissioning and during the usage.

Should you encounter any unexpected problems during use, please do not hesitate to contact us.

2. USE WITH THE ARDUINO

Connect the matrix



Pin number Arduino	Pin number ButtonMatrix
13	7
12	6
11	5
10	4
9	0
8	1
7	2
6	3

In this example the Arduino Uno was connected with the 4 x 4 ButtonMatrix.

Connect the digital pins (6-13) of the Arduino with the ButtonMatrix like shown in the picture and in the chart.

Installation of the matrix

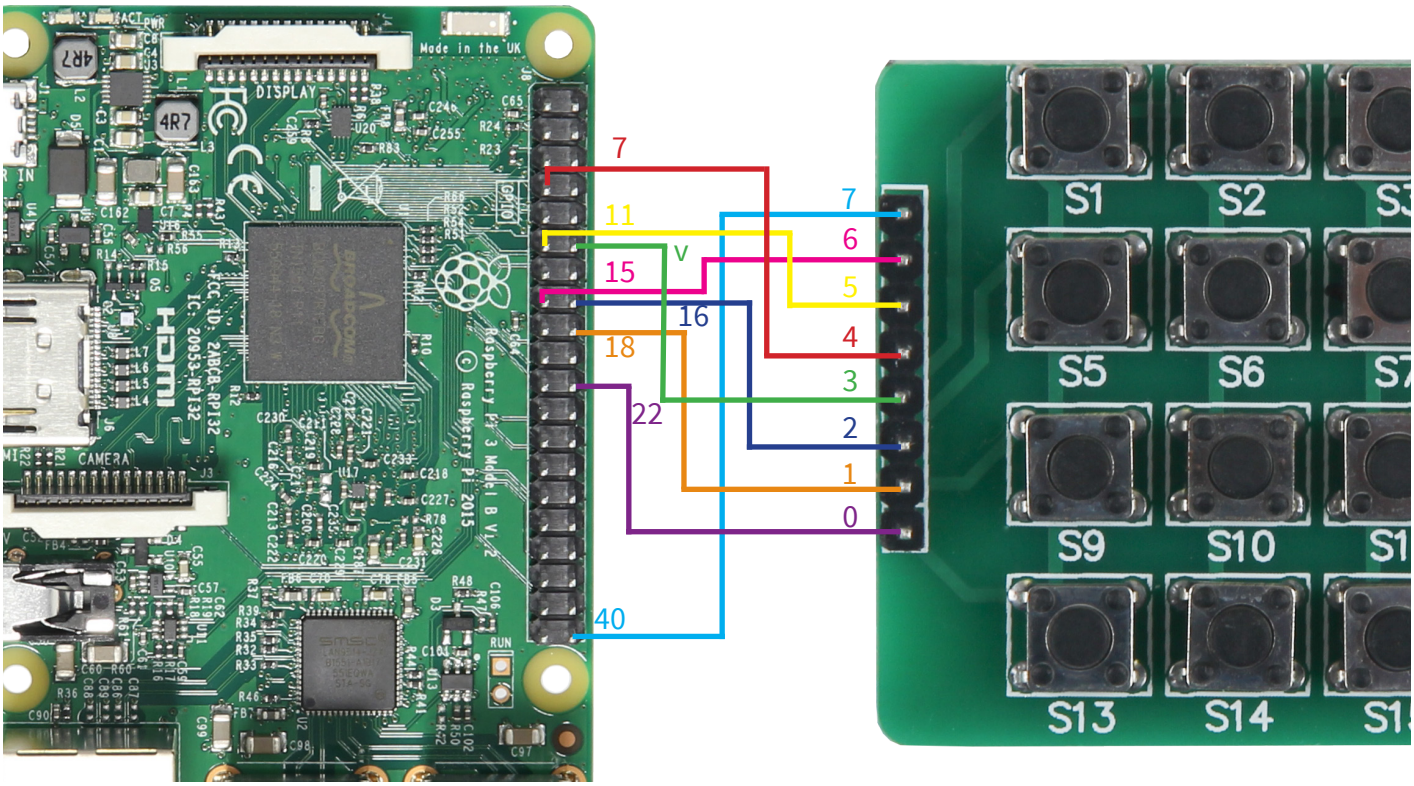
In the following, you can find a working code example which can be tranferred with the Arduino.
The function of the containing buttons can be expanded in the **knopfDruck**-function according to your wishes.

```
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++){
            //Initialization of the PINs
            pinMode(reihe[i],OUTPUT);
            pinMode(spalte[i],INPUT);
            digitalWrite(spalte[i],HIGH);
      }
}

void loop(){
      //Search for the pressed button
      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){
                        //If pressed button is detected
                        //execute knopfDruck
                        knopfDruck(i,j);
                        delay(300);
                  }
            }
      }
}

void knopfDruck(int i,int j){
      if(i==0&&j==0) //Button S1 is pressed
            Serial.println("S1");
      if(i==0&&j==1) //Button S2 is pressed
            Serial.println("S2");
      if(i==0&&j==2) //Button S3 is pressed
            Serial.println("S3");
      if(i==0&&j==3) //Button S4 is pressed
            Serial.println("S4");
      if(i==1&&j==0) //Button S5 is pressed
            Serial.println("S5");
      if(i==1&&j==1) //Button S6 is pressed
            Serial.println("S6");
```

```
      if(i==1&&j==2) //Button S7 is pressed
            Serial.println(“S7”);
      if(i==1&&j==3) //Button S8 is pressed
            Serial.println(“S8”);
      if(i==2&&j==0) //Button S9 is pressed
            Serial.println(“S9”);
      if(i==2&&j==1) //Button S10 is pressed
            Serial.println(“S10”);
      if(i==2&&j==2) //Button S11 is pressed
            Serial.println(“S11”);
      if(i==2&&j==3) //Button S12 is pressed
            Serial.println(“S12”);
      if(i==3&&j==0) //Button S13 is pressed
            Serial.println(“S13”);
      if(i==3&&j==1) //Button S14 is pressed
            Serial.println(“S14”);
      if(i==3&&j==2) //Button S15 is pressed
            Serial.println(“S15”)
      if(i==3&&j==3) //Button S16 is pressed
            Serial.println(“S16”);
}
```

3. USE WITH THE RASPBERRY PI

Connect the matrix



Pin Raspberry Pi	Pin number ButtonMatrix
PIN 40 (BCM 21)	7
PIN 15 (BCM 22)	6
PIN 11 (BCM 17)	5
PIN 7 (BCM 4)	4
PIN 12 (BCM 18)	3
PIN 16 (BCM 23)	2
PIN 18 (BCM 24)	1
PIN 22 (BCM 25)	0

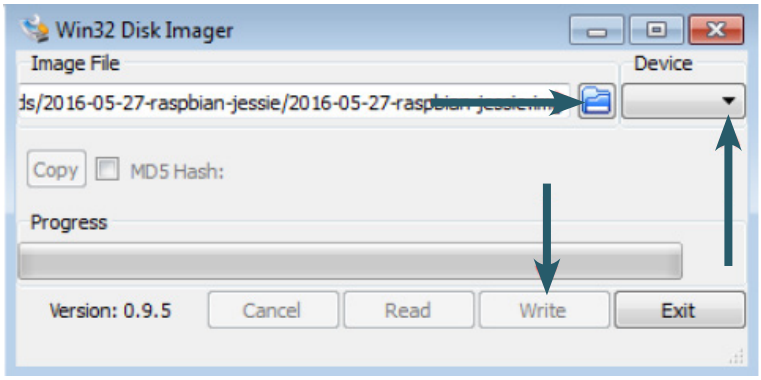
In this example the ButtonMatrix was connected with the Raspberry Pi 3 model B.

Connect the ButtonMatrix with the pins of the Raspberry Pi like shown in the picture and the chart.

Installation of software

Should you have already a current version of the Raspbian software on your Raspberry Pi, you can skip this step and continue with the installation of the libraries.

Install on your SD-card with the **„Win32 Disk Imager“**-program the current Raspbian image which can be downloaded with the following **link**.



Installation of libraries

As soon as teh installation of the software is finished and the system has started, open the terminal console and run the following commands:

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

installation of the GPIO-library

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

installaion of the Python-library

Installation of the matrix

First of all, you have to create a new Python-file:

```
sudo nano matrix.py
```

Write this code completely into the editor which has opened.

```
import RPi.GPIO as GPIO
import time

class ButtonMatrix():

      def __init__(self):
            GPIO.setmode(GPIO.BCM)
            # The IDs of the buttons are defined
            self.buttonIDs = [[4,3,2,1],[8,7,6,5],[12,11,10,9],[16,15,14,13]]
            # GPIO pines for the lines are declared
            self.rowPins = [18,23,24,25]
            # GPIO pines for the columns are declared
            self.columnPins = [21,22,17,4]
            # Define four inputs with pull up resistors
            for i in range(len(self.rowPins)):
                  GPIO.setup(self.rowPins[i],GPIO.IN,pull_up_down=GPIO.PUD_UP)
                  # Define four outputs and set them to HIGH
            for j in range(len(self.columnPins)):
                  GPIO.setup(self.columnPins[j], GPIO.OUT)
                  GPIO.output(self.columnPins[j], 1)

      def activateButton(self, rowPin, colPin):
      # Get the button number
            btnIndex = self.buttonIDs[rowPin][colPin] - 1
            print("button " + str(btnIndex + 1) + " pressed")
            # Prevents multiple button presses in a too short amount of time
            time.sleep(.3)

      def buttonHeldDown(self,pin):
            if(GPIO.input(self.rowPins[pin]) == 0):
                  return True
            return False


def main():
      # Initialization of the Button matrix
      buttons = ButtonMatrix()
      try:
            while(True):
                  for j in range(len(buttons.columnPins)):
                  # each output pin is set to LOW
                        GPIO.output(buttons.columnPins[j],0)
                        for i in range(len(buttons.rowPins)):
                              if GPIO.input(buttons.rowPins[i]) == 0:
                                    buttons.activateButton(i,j)
                                    # Do nothing while the button
                                    # is pressed.
                                    while(buttons.buttonHeldDown(i)):
                                          pass
                        GPIO.output(buttons.columnPins[j],1)
      except KeyboardInterrupt:
            GPIO.cleanup()

if __name__ == "__main__":
  main()
```

The file can be saved with **Strg+O** and the editor can be left with **Strg+X**. After that the file can be run with the following command and that way it will be tested.

```
sudo python matrix.py
```

The file can be closed again with **Strg+C**.

4. FURTHER INFORMATION

Our information and redemption obligation according to the electro-law (ElektroG)

Symbol on electrial and electronic products :
This crossed-out bin means that electrical and electronic products do **not** belong into the household waste. You must hand over your old appli-ance to a registration office. Before you can hand over the old appliance, you must remove used batteries and accumulators which are not enclosed by the device.

Return options :
As the end user, you can hand over with the purchase of a new device your old appliance (which has essentially the same functions as the new one) free of charge for disposal. Small devices which do not have outer dimensions greater than 25 cm can be submitted independently of the purchase of a new product in normal household quantities.

Possibility of restitution at our company location during our opening hours :
Simac GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

Possibility of restitution nearby :
We send you a parcel stamp with which you can send us your old appliance free of charge. For this possibility, you must contact us via e-mail at service@joy-it.net or via telephone.

Information about packaging:
Please package your old appliance safe during transport. Should you not have a suitable packaging material or you do not want to use your own material, you can contact us and we will send you an appropriate package.

CE

5. SUPPORT

If any questions remain open or problems arise after your purchase, we are available by e-mail, telephone and with a ticket support system to answer these.

E-Mail: service@joy-it.net
Ticket-System: http://support.joy-it.net
Telephone: +49 (0)2845 98469 – 66 (10 - 17 o‘clock)

For further information visit our website:s
www.joy-it.net

Published: 27.08.2019