

Manuel d'utilisation du module matrice 16 boutons poussoirs



Cher client,

Merci d'avoir acheté notre produit.

Veuillez observer les instructions ci-dessous avant la première utilisation :

Utilisation avec un Arduino (ou compatible)

Connecter le module

Branchez le module aux broches de l'Arduino comme représenté ci-dessous :

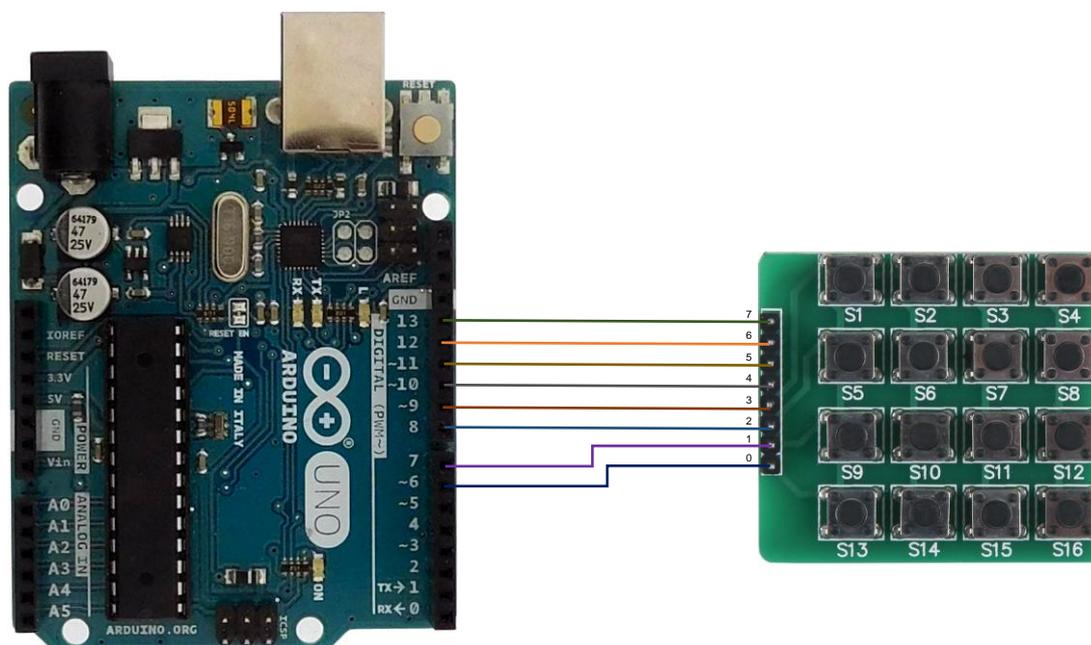


Table de correspondance :

Arduino	Module matrice 16 BP
6	0
7	1
8	2
9	3
10	4
11	5
12	6
13	7

Exemple de programme

L'exemple de code suivant (à copier dans l'IDE Arduino) permet d'afficher le nom de la touche enfoncée dans le moniteur série (Ctrl+MAJ+M pour démarrer le moniteur série) :

The image shows a screenshot of the Arduino IDE interface. The window title is "Test_MatriceBP | Arduino 1.8.0". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for saving, running, uploading, and downloading. The main editor area shows the following C++ code:

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

void setup()
{
  Serial.begin(9600);
  for (int i = 0; i <= 3; i++)
  {
    // Initialisation des broches
    pinMode(rangee[i], OUTPUT);
    pinMode(colonne[i], INPUT);
    digitalWrite(colonne[i], HIGH);
  }
  Serial.println("==== Clavier pret =====");
}

void loop()
{
  // Regarde si un bouton est enfoncé
  for (int i = 0; i <= 3; i++)
  {
    digitalWrite(rangee[0], HIGH);
    digitalWrite(rangee[1], HIGH);
    digitalWrite(rangee[2], HIGH);
    digitalWrite(rangee[3], HIGH);
    digitalWrite(rangee[i], LOW);
    for (int j = 0; j <= 3; j++)
    {
      col_scan = digitalRead(colonne[j]);
      if (col_scan == LOW)
      {
        // Lorsqu'un bouton est enfoncé, appel de la fonction toucherBouton
        // pour savoir quel bouton est enfoncé
        toucherBouton(i, j);
        delay(300);
      }
    }
  }
}
```

```
// Fonction de reconnaissance et affichage du bouton enfoncé
void toucherBouton(int i, int j)
{
  if (i == 0 && j == 0) // Bouton S1 enfoncé
    Serial.println("S1");
  if (i == 0 && j == 1) // Bouton S2 enfoncé
    Serial.println("S2");
  if (i == 0 && j == 2) // Bouton S3 enfoncé
    Serial.println("S3");
  if (i == 0 && j == 3) // Bouton S4 enfoncé
    Serial.println("S4");
  if (i == 1 && j == 0) // Bouton S5 enfoncé
    Serial.println("S5");
  if (i == 1 && j == 1) // Bouton S6 enfoncé
    Serial.println("S6");
  if (i == 1 && j == 2) // Bouton S7 enfoncé
    Serial.println("S7");
  if (i == 1 && j == 3) // Bouton S8 enfoncé
    Serial.println("S8");
  if (i == 2 && j == 0) // Bouton S9 enfoncé
    Serial.println("S9");
  if (i == 2 && j == 1) // Bouton S10 enfoncé
    Serial.println("S10");
  if (i == 2 && j == 2) // Bouton S11 enfoncé
    Serial.println("S11");
  if (i == 2 && j == 3) // Bouton S12 enfoncé
    Serial.println("S12");
  if (i == 3 && j == 0) // Bouton S13 enfoncé
    Serial.println("S13");
  if (i == 3 && j == 1) // Bouton S14 enfoncé
    Serial.println("S14");
  if (i == 3 && j == 2) // Bouton S15 enfoncé
    Serial.println("S15");
  if (i == 3 && j == 3) // Bouton S16 enfoncé
    Serial.println("S16");
}
```

Utilisation avec un Raspberry Pi

Connecter le module

Branchez le module LCD aux broches du Raspberry Pi comme représenté ci-dessous :

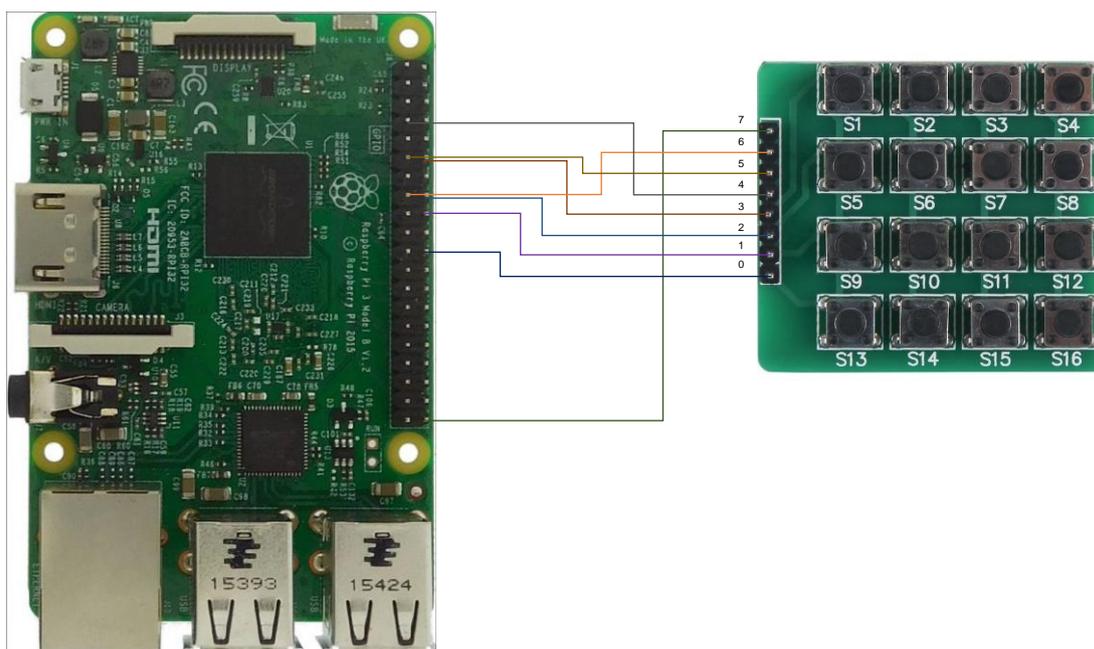


Table de correspondance :

Raspberry Pi	Module matrice 16 BP
PIN 22 (GPIO 25)	0
PIN 18 (GPIO 24)	1
PIN 16 (GPIO 23)	2
PIN 12 (GPIO 18)	3
PIN 7 (GPIO 4)	4
PIN 11 (GPIO 17)	5
PIN 15 (GPIO 22)	6
PIN 40 (GPIO 21)	7

Prérequis

Les explications et programmes qui suivent ont été créés et testés pour un Raspberry utilisant le système d'exploitation Raspbian.

Un guide d'installation est disponible ici :

<http://www.gotronic.fr/userfiles/www.gotronic.fr/files/Raspberry/GuideRPI.pdf>

Un guide de démarrage est disponible ici :

<http://www.gotronic.fr/userfiles/www.gotronic.fr/files/Raspberry/Raspberry%20Joyit.pdf>

Installation des bibliothèques

Pour installer les bibliothèques nécessaires, entrez les commandes ci-dessous dans LXTerminal :

Bibliothèque GPIO :

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

Bibliothèque Python :

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

Exemple de programme

Le programme suivant affiche à l'écran le numéro du bouton pressé.

Créez un nouveau fichier vierge *MatriceBP.py* en entrant la commande suivante :

```
sudo nano matriceBP.py
```

Et copier les lignes suivantes dans ce fichier :

```
# coding=utf-8
import time
import RPi.GPIO as GPIO
GPIO.setwarnings(False)

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

    def getKey(self):
    # Définition des colonnes en sorties
        for j in range(len(self.COLUMN)):
            GPIO.setup(self.COLUMN[j], GPIO.OUT)
            GPIO.output(self.COLUMN[j], GPIO.LOW)
    # Définition des lignes en entrées
        for i in range(len(self.ROW)):
            GPIO.setup(self.ROW[i], GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

```
# Regarde si un bouton est poussé
rowVal = -1
for i in range(len(self.ROW)):
    tmpRead = GPIO.input(self.ROW[i])
    if tmpRead == 0:
        rowVal = i
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
    # Retourne la position du bouton pressé
    self.exit()
    return self.KEYPAD[rowVal][colVal]
def exit(self):
    # Retourne le nom du bouton pressé
    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__':
# Initialisation du clavier
    kp = keypad()
    print '--- Clavier prêt ---'
# Boucle principale
    while True:
        digit = None
        while digit == None:
            digit = kp.getKey()
        # Affiche le nom de la touche enfoncée
        print digit
        time.sleep(0.5)
```

Sauvegardez le fichier (*Ctrl+O* et *Enter*) et quittez le fichier (*Ctrl+X*).

Vous pouvez exécuter le programme :

```
sudo python matriceBP.py
```

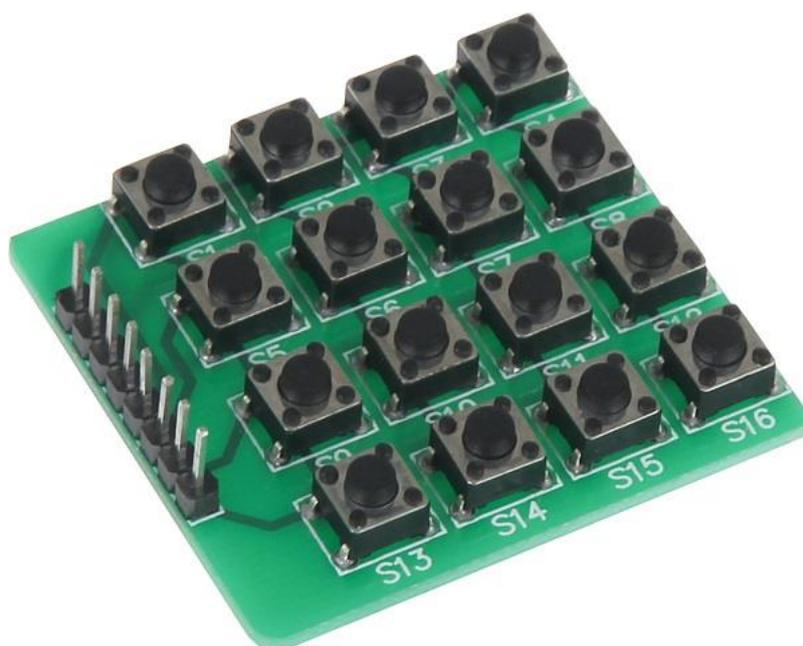
Ctrl+C pour arrêter le programme

GO TRONIC

ROBOTIQUE ET COMPOSANTS ÉLECTRONIQUES

Si vous rencontrez des problèmes, merci de nous contacter par courriel à :

sav@gotronic.fr



Coordonnées du fabricant :

JOY-IT[®]

service@joy-it.net

+49 (0)2845 9360 – 50