

PCF 8574

Module PCF 8574

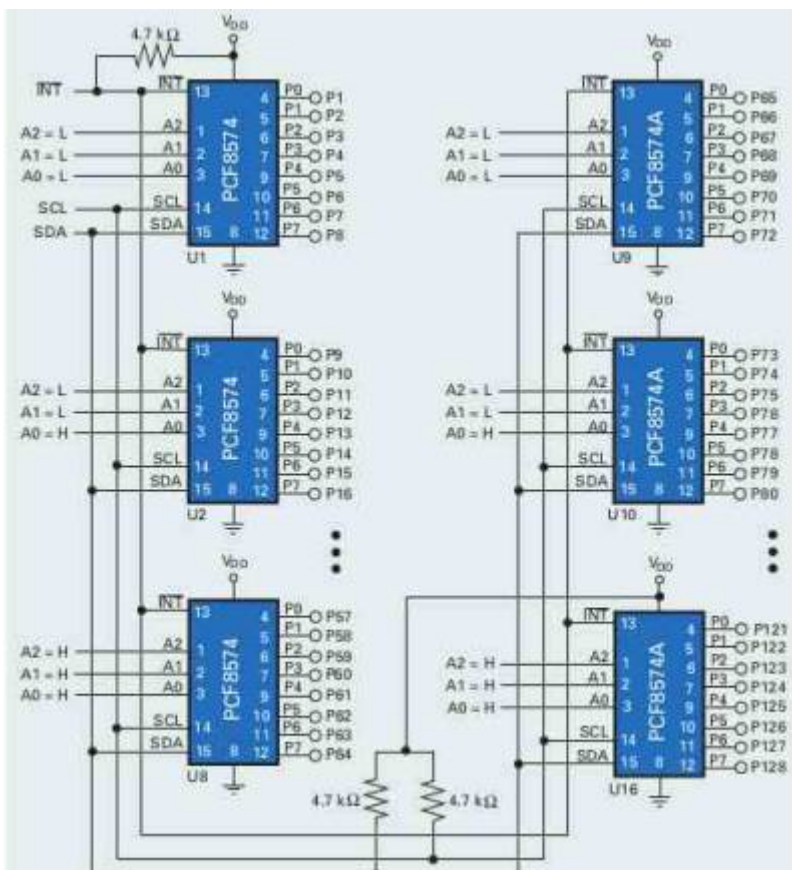


Lorsque vous jouez avec des microcontrôleurs, il est parfois nécessaire d'ajouter plus d'E/S numériques que la puce ne prend en charge de manière native.

Le module PCF8574 (PCF8574T) est l'un des moyens les plus populaires d'étendre les E/S car il utilise le bus I2C qui ne nécessite que 2 broches sur le microcontrôleur. Il fournit 8 E/S numériques supplémentaires qui peuvent facilement être connectées en série jusqu'à 64.

Notez que le module dispose d'une interface I2C facile à utiliser qui peut être configurée via des cavaliers mobiles pour utiliser l'une des huit adresses I2C.

Les trois broches d'adresse matérielle permettent à huit de chaque appareil d'être sur le même bus I2C, il peut donc y avoir jusqu'à 16 de ces extenseurs d'E/S PCF8574/PCF8574A ensemble sur le même bus I2C, prenant en charge jusqu'à 128 I/Os.



Voici les principales caractéristiques du module présenté ici :

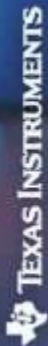
1. -8 lignes de données bidirectionnelles
2. -La fonction de boucle permet l'extension jusqu'à 8 modules / 64 lignes de données
3. -Interface I 2 C avec adresses réglables par cavalier
4. -Capacité de sortie d'interruption
5. -Compatibilité 3.3V et 5V.

Adresses I2C

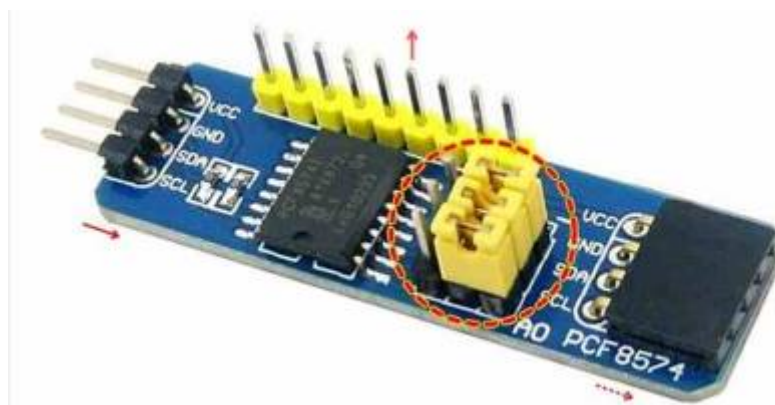
Il y a trois sauts de sélection d'adresse (A0-A2) qui déterminent quelle adresse I 2 C utiliser. Tels qu'ils sont livrés, ces cavaliers sont tous réglés sur le côté BAS (GND) comme indiqué sur l'image ci-dessus.

Cela place le module à l'adresse esclave par défaut/de base de 0x20 (hexadécimal 7 bits). Les cavaliers peuvent être déplacés de manière binaire pour changer l'adresse I 2 C (0x20 à 0x27) comme indiqué dans la carte d'adresses esclave ci-dessous.

INPUTS			PCF8574 I ² C-Bus Slave Address
A2	A1	A0	
L	L	L	20 (hexadecimal)
L	L	H	21 (hexadecimal)
L	H	L	22 (hexadecimal)
L	H	H	23 (hexadecimal)
H	L	L	24 (hexadecimal)
H	L	H	25 (hexadecimal)
H	H	L	26 (hexadecimal)
H	H	H	27 (hexadecimal)



ATTENTION : le réperage A2 et A0 peut être inversé sur la serigraphie



Bibliothèque PCF8574 pour ESP32

Utiliser la librairie PCF8574 pour esp32



Programmes exemple

BlinkOnPin0.ino

```

/*
  Blink led on PIN0
  by Mischiants Renzo <http://www.mischiants.org>

  https://www.mischiants.org/2019/01/02/pcf8574-i2c-digital-i-o-expander-
  fast-easy-usage/

  Permet de faire clignoter une LED sur la broche 0
*/

#include "Arduino.h"
#include "PCF8574.h"

// Set i2c address
PCF8574 pcf8574_1(0x20); // on peut en avoir 8 sur le bus I2C

void setup()
{
  Serial.begin(115200);
  delay(1000);

  // Set pinMode to OUTPUT
  pcf8574_1.pinMode(P0, OUTPUT);
  pcf8574_1.pinMode(P1, INPUT);

  Serial.print("Init pcf8574...");
  if (pcf8574_1.begin()){
    Serial.println("OK");
  }else{
    Serial.println("KO");
  }
}

```

```
void loop()
{
  pcf8574_1.digitalWrite(P0, HIGH);
  delay(1000);
  pcf8574_1.digitalWrite(P0, LOW);
  delay(1000);
}
```

KeyPressedPin1.ino

```
/*
  KeYPressed on PIN1
  by Mischianti Renzo <http://www.mischianti.org>

  https://www.mischianti.org/2019/01/02/pcf8574-i2c-digital-i-o-expander-fast-easy-usage/

  Test l'appui sur le Bouton poussoir broche 1
*/

#include "Arduino.h"
#include "PCF8574.h"

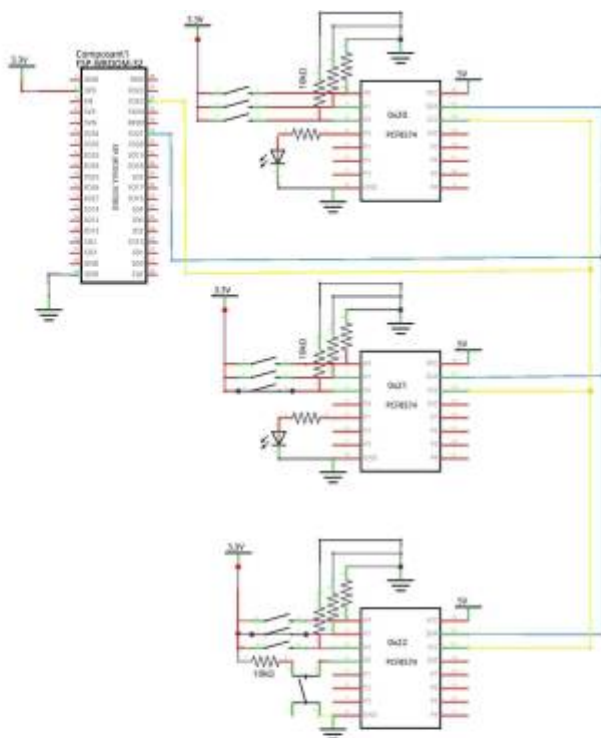
// Set i2c address
PCF8574 pcf8574_1(0x22);
unsigned long timeElapsed;
void setup()
{
  Serial.begin(115200);
  delay(1000);

  pcf8574_1.pinMode(P0, OUTPUT);
  pcf8574_1.pinMode(P1, INPUT);

  Serial.print("Init pcf8574_1...");
  if (pcf8574_1.begin()){
    Serial.println("OK");
  }else{
    Serial.println("KO");
  }
}

void loop()
{
  uint8_t val = pcf8574_1.digitalRead(P1);
  if (val==HIGH) Serial.println("Bouton Appuyer");
  delay(50);
}
```

Un schema



Address (Hex)	A2	A1	A0
0x20	LOW	LOW	LOW
0x21	LOW	LOW	HIGH
0x22	LOW	HIGH	LOW
0x23	LOW	HIGH	HIGH
0x24	HIGH	LOW	LOW
0x25	HIGH	LOW	HIGH
0x26	HIGH	HIGH	LOW
0x27	HIGH	HIGH	HIGH

fritzing



ATTENTION : A0 , A1 , A2 , peuvent etre inversé !!!

From:

<https://chanterie37.fr/fablab37110/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:

<https://chanterie37.fr/fablab37110/doku.php?id=start:arduino:esp32:pcf8574>

Last update: **2023/01/27 16:08**

