

# [I2C] PCF8574: Módulo de expansión y código para Arduino

Existen en el mercado diferentes placas que integran este circuito y que nos permiten realizar proyectos de forma cómoda y rápida. Suelen disponer de la opción de configurar con A0, A1 y A2 la dirección final I2C del módulo mediante puentes o soldaduras.



## Ejemplos de código y librerías para Arduino

- [Creating Arduino Library for PCF8574 I/O Expander / GitHub repository to access PCF8574 library](#) ⇒ pcf8574-8-bit-i-o-expander-main.zip

Fix Me!

- pcf8574-master.zip

Fix Me!

⇒

```
#include "PCF8574.h"
```

- [pcf8574 Arduino Library List](#) ⇒ pcf8574-1.0.3.zip

Fix Me!

- Artículos
  - [PCF8574. Expansor I2C de 8 bits de puertos GPIO digitales de entrada y salida.](#)
  - [Comunicación I2C con Arduino lo mejor de dos mundos](#)
  - [MÁS PINES DIGITALES CON ARDUINO Y EXPANSOR E/S I2C PCF8574](#)
- Youtube
  - [Expansor de puertos: 74HC595 + bit banging y PCF8574](#)
  - [Ampliación de pines digitales para Arduino con el expansor PCF8574](#)

## Ejemplos sin librería

1. El siguiente código muestra el uso del PCF8574 como salida. En el ejemplo, encendemos y apagamos en bucle cada uno de los pines. Observar que la señal del canal se envía negada, porque el PCF8574 actúa como sumidero de corriente y el pin está activo cuando la salida es LOW.

[pcf8574\\_salida.ino](#)

```
#include <Wire.h>

const int pcfAddress = 0x20;

void setup()
{
  Wire.begin();
  Serial.begin(9600);
}
```

```
void loop()
{
  for (short channel = 0; channel < 8; channel++)
  {
    // Escribir dato en cada uno de los 8 canales
    Wire.beginTransmission(pcfAddress);
    Wire.write(~(1 << channel));
    Wire.endTransmission();

    // Leer dato del canal
    delay(500);
  }
}
```

2. El siguiente código muestra el uso del PCF8574 como entrada. En el ejemplo realizamos la lectura progresiva de cada uno de los pines, y mostramos su valor por puerto serie.

#### [pcf8574\\_entrada.ino](#)

```
#include <Wire.h>

const int pcfAddress = 0x20;

void setup()
{
  Wire.begin();
  Serial.begin(9600);
}

void loop()
{
  short channel = 1;
  byte value = 0;

  // Leer dato del canal
  Wire.requestFrom(pcfAddress, 1 << channel);
  if (Wire.available()) {
    value = Wire.read();
  }
  Wire.endTransmission();

  // Mostrar el valor por el monitor serie
  Serial.println(value);
}
```

- **Interrupciones.** INTERRUPTOR OUT (INT); Es un pin de drenaje abierto por lo que se necesita conectar una resistencia de PULL-UP, esto se realiza para lograr un nivel alto (5V) constante. Este puerto funciona de la siguiente manera: Cuando detecta un flanco ya sea ascendente o descendente en cualquiera de los pines P1 a P7, este pin (INT) inmediatamente activa un mosfet interno de canal N, por ende tendremos un 0 V cada vez que se obtenga un cambio de estado en esos pines (P1 - P7).
  - NOTA: si se usa un microcontrolador (ATmega) que sea programado a través del IDE de arduino ya no será necesario colocar una resistencia de PULL-UP, ya que internamente cuenta una resistencia de PULL-UP, dicha resistencia interna se podrá activar desde la programación.

#### [interrupcion.ino](#)

```
int pin_INT =2;// INTERRUPTOR DEL CI PCF8574
```

```
pinMode( pin_INT, INPUT_PULLUP);
```



- [PCF8574 circuito expensor de pines](#)
- si quieres definir varios pines al mismo tiempo es mas facil usar numero binarios.

```
pcf8574.pinMode(0b11111111, OUTPUT); //Define P0 a P7 como salida.
```

## Ejemplos con librería



pcf8574\_arduino\_library-master.zip

1. Alternativamente podemos emplear una librería como la desarrollada por SkyWood disponible en [https://github.com/skywodd/pcf8574\\_arduino\\_library](https://github.com/skywodd/pcf8574_arduino_library). La librería proporciona códigos de ejemplo para ilustrar su uso. No obstante, aquí tenéis un ejemplo resumido que muestra su uso

[pcf8574\\_lib.ino](#)

```
#include <Wire.h>
#include "PCF8574.h"

PCF8574 expander;

void setup()
{
  Serial.begin(9600);

  expander.begin(0x20);

  /* Setup some PCF8574 pins for demo */
  expander.pinMode(0, OUTPUT);
  expander.pinMode(1, OUTPUT);
  expander.pinMode(2, OUTPUT);
  expander.pinMode(3, INPUT_PULLUP);

  /* Blink hardware LED for debug */
  digitalWrite(13, HIGH);

  /* Toggle PCF8574 output 0 for demo */
  expander.toggle();

  /* Blink hardware LED for debug */
  digitalWrite(13, LOW);
}

void loop()
{
}
```

- [pulsador.ino](#)

```
// Juan A. Villalpando
// kio4.com

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

PCF8574 pcf8574(0x20); // Dirección del módulo
int LED13 = 13;

void setup(){
  pcf8574.begin();
  pinMode(LED13,OUTPUT);
  pcf8574.pinMode(P1, INPUT); // Pulsador en P1 del módulo
}
void loop(){
  uint8_t P1_Read = pcf8574.digitalRead(P1);
  if (P1_Read == HIGH){
    digitalWrite(LED13, HIGH);
  } else {digitalWrite(LED13, LOW);}

  delay(50);
}
```

From:  
<https://www.euloxio.myds.me/dokuwiki/> - Euloxio wiki

Permanent link:  
[https://www.euloxio.myds.me/dokuwiki/doku.php/doc:tec:lab:bus\\_i2c:pcf8574\\_modulo:inicio?rev=1774719265](https://www.euloxio.myds.me/dokuwiki/doku.php/doc:tec:lab:bus_i2c:pcf8574_modulo:inicio?rev=1774719265)

Last update: 2026/03/28 18:34

