

## PCF8574-I2C

Generated on Sat Jan 17 2026 22:52:36 for PCF8574-I2C by Doxygen 1.9.8

Sat Jan 17 2026 22:52:36



---

<b>1 PCF8574-I2C Library</b>	<b>1</b>
1.1 Contents . . . . .	1
1.2 Library Documentation . . . . .	1
1.3 Library Usage . . . . .	1
1.3.1 Controllers . . . . .	1
1.3.2 Usage the PCF8574-I2C Library in the Code . . . . .	2
1.3.3 Status Codes of PCF8574 . . . . .	2
1.4 License . . . . .	2
1.5 Helpful Links . . . . .	2
<b>2 Class Index</b>	<b>3</b>
2.1 Class List . . . . .	3
<b>3 File Index</b>	<b>5</b>
3.1 File List . . . . .	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 PCF8574_I2C::PCF8574 Class Reference . . . . .	7
4.1.1 Constructor & Destructor Documentation . . . . .	7
4.1.1.1 PCF8574() . . . . .	7
4.1.2 Member Function Documentation . . . . .	8
4.1.2.1 begin() . . . . .	8
4.1.2.2 readPin() . . . . .	8
4.1.2.3 resetPort() . . . . .	8
4.1.2.4 rotateLeft() . . . . .	9
4.1.2.5 rotateRight() . . . . .	9
4.1.2.6 setPin() . . . . .	9
4.1.2.7 setPort() . . . . .	10
4.1.2.8 toggle() . . . . .	10
<b>5 File Documentation</b>	<b>13</b>
5.1 src/PCF8574-I2C.cpp File Reference . . . . .	13
5.1.1 Detailed Description . . . . .	13
5.2 src/PCF8574-I2C.h File Reference . . . . .	13
5.2.1 Detailed Description . . . . .	14
5.3 PCF8574-I2C.h . . . . .	14
<b>Index</b>	<b>17</b>



# Chapter 1

## PCF8574-I2C Library

Arduino Library for PCF8574, a 8-port GPIO exander via i2c

### 1.1 Contents

- Library Documentation
- Library Usage
- License
- Helpful Links

### 1.2 Library Documentation

The library documentation is mainly placed in the following pdf document [refman.pdf](#) or located under the following github pages [github.io](#).

Additionally in combination with the technical datasheet of microchip [PCF8574-Datasheet](#).

### 1.3 Library Usage

#### 1.3.1 Controllers

The library is intended to be used on each microcontroller for Example:

- Arnuino Nano
- Arduino Nano 33 IOT
- ESP8266
- ESP32
- etc ...

### 1.3.2 Usage the PCF8574-I2C Library in the Code

Include the library in you project via:

```
#include <PCF8574-I2C.h>
```

Instance an new PCF8574 object by:

```
PCF8574_I2C::PCF8574 pcf{0x20, &Wire};  
or simply use implicit defined Wire object like:  
PCF8574_I2C::PCF8574 pcf{0x20};
```

Now you can use the object and his members as normal like:

```
PCF8574_I2C::PCF8574 pcf{0x20, &Wire};  
void setup() {  
    Serial.begin(115200);  
    Serial.print("\n\nPCF8574 Test file with ESP8266-01\n");  
  
    Wire.begin(2, 0);  
  
    if (pcf.begin() == PCF8574_I2C::PCF8574_STATE_OK) {  
        Serial.print("\tPCF8574 Connection OK!\n");  
    }  
    else {  
        Serial.print("\tNO PCF8574 device found!\n");  
    }  
    pcf.resetPort();  
}
```

Please refer to the examples and the above mentioned documentation files.

### 1.3.3 Status Codes of PCF8574

The following status codes exists:

- PCF8574\_STATE\_OK {0};
- PCF8574\_ERROR\_PIN {-1};
- PCF8574\_ERROR\_I2C {-2};
- PCF8574\_ERROR\_VALUE {-3};

## 1.4 License

This library is licensed under MIT Licence.

[PCF8574-I2C License](#)

## 1.5 Helpful Links

- [ESP8266-01-Adapter](#)

# **Chapter 2**

## **Class Index**

### **2.1 Class List**

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">PCF8574_I2C::PCF8574</a> . . . . .	7
--	---



# Chapter 3

## File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">src/PCF8574-I2C.cpp</a>	Library for a PCF8574 GPIO expander . . . . .	13
<a href="#">src/PCF8574-I2C.h</a>	Library for a PCF8574 GPIO expander . . . . .	13



# Chapter 4

## Class Documentation

### 4.1 PCF8574\_I2C::PCF8574 Class Reference

#### Public Member Functions

- `PCF8574` (const uint8\_t address=0x20, TwoWire \*wire=&Wire)  
*Construct a new `PCF8574` object.*
- `int8_t begin () const`  
*begin method which initializes and verifies connection. Calls `isDevicePresent()`*
- `int8_t resetPort ()`  
*reset the `PCF8574` device, set all port pins to input*
- `int16_t readPin (int8_t pin=-1)`  
*read pin(s) from port*
- `int8_t setPin (uint8_t pin, uint8_t value)`  
*Set the a specific pin on the PCF8474 to 0 or 1.*
- `int8_t setPort (uint8_t value)`  
*Set the Port at once to an value.*
- `int8_t toggle (uint8_t mask)`  
*toggle pins by give a mask which pin to toggle*
- `int8_t rotateLeft ()`  
*rotate Port to the left*
- `int8_t rotateRight ()`  
*rotate port to the right*

#### 4.1.1 Constructor & Destructor Documentation

##### 4.1.1.1 `PCF8574()`

```
PCF8574::PCF8574 (
    const uint8_t address = 0x20,
    TwoWire * wire = &Wire )
```

Construct a new `PCF8574` object.

**Parameters**

<i>address</i>	of <a href="#">PCF8574</a> device
<i>wire</i>	pointer of TwoWire object

**4.1.2 Member Function Documentation****4.1.2.1 begin()**

```
int8_t PCF8574::begin ( ) const
```

begin method which initializes and verifies connection. Calls isDevicePresent()

**Return values**

<i>0</i>	if device is connected
<i>-2</i>	if connection failed due to missing connection or line error.

**4.1.2.2 readPin()**

```
int16_t PCF8574::readPin (
    int8_t pin = -1 )
```

read pin(s) from port

**Parameters**

<i>pin</i>	give pin number 0...7; or ommit for all read all port pins
------------	--

**Returns**

*int8\_t* return error code or value

**Return values**

<i>&gt;0</i>	read was ok => return pin/Port value
<i>&lt;0</i>	error during readPin => return error code

**4.1.2.3 resetPort()**

```
int8_t PCF8574::resetPort ( )
```

reset the [PCF8574](#) device, set all port pins to input

**Returns**

*int8\_t* status of the write command

Return values

0	successfull
-2	error on i2c connection

#### 4.1.2.4 rotateLeft()

```
int8_t PCF8574::rotateLeft ( )
```

rotate Port to the left

Returns

```
int8_t status of the write command
```

Return values

0	successfull
-2	error on i2c connection

#### 4.1.2.5 rotateRight()

```
int8_t PCF8574::rotateRight ( )
```

rotate port to the right

Returns

```
int8_t status of the write command
```

Return values

0	successfull
-2	error on i2c connection

#### 4.1.2.6 setPin()

```
int8_t PCF8574::setPin (
    uint8_t pin,
    uint8_t value )
```

Set the a specific pin on the PCF8474 to 0 or 1.

Parameters

<i>pin</i>	number of port pin P0...P7
<i>value</i>	value of pin 0 or 1

**Returns**

```
int8_t status of the write command
```

**Return values**

<i>0</i>	successfull
<i>-1</i>	pin error; wrong pin number
<i>-2</i>	error on i2c connection
<i>-3</i>	wrong value for pin

**4.1.2.7 setPort()**

```
int8_t PCF8574::setPort (
    uint8_t value )
```

Set the Port at once to an value.

**Parameters**

<i>value</i>	for port to set
--------------	-----------------

**Returns**

```
int8_t status of the write command
```

**Return values**

<i>0</i>	successfull
<i>-2</i>	error on i2c connection

**4.1.2.8 toggle()**

```
int8_t PCF8574::toggle (
    uint8_t mask )
```

toggle pins by give a mask which pin to toggle

**Parameters**

<i>mask</i>	define which pin to toggle like: 0b10010101
-------------	---

**Returns**

```
int8_t status of the write command
```

**Return values**

0	successfull
-2	error on i2c connection

The documentation for this class was generated from the following files:

- src/[PCF8574-I2C.h](#)
- src/[PCF8574-I2C.cpp](#)



# Chapter 5

## File Documentation

### 5.1 src/PCF8574-I2C.cpp File Reference

Library for a PCF8574 GPIO expander.

```
#include "PCF8574-I2C.h"
```

#### 5.1.1 Detailed Description

Library for a PCF8574 GPIO expander.

Author

Frank Häfele ( [mail@frankhaefele.de](mailto:mail@frankhaefele.de))

Version

1.0.0

Date

2026-01-12

Copyright

Copyright (c) 2026

### 5.2 src/PCF8574-I2C.h File Reference

Library for a PCF8574 GPIO expander.

```
#include "Wire.h"
```

## Classes

- class [PCF8574\\_I2C::PCF8574](#)

## Variables

- constexpr const char \* **PCF8574\_I2C::PCF8574\_LIB\_VERSION** {"1.0.0"}
- constexpr int8\_t **PCF8574\_I2C::PCF8574\_STATE\_OK** {0x00}  
*constant which states all ok, no error*
- constexpr int8\_t **PCF8574\_I2C::PCF8574\_ERROR\_PIN** {-1}  
*constant which states a wrong pin number was used*
- constexpr int8\_t **PCF8574\_I2C::PCF8574\_ERROR\_I2C** {-2}  
*constant which states an error during I2C communication*
- constexpr int8\_t **PCF8574\_I2C::PCF8574\_ERROR\_VALUE** {-3}  
*constant which states that there was an error regarding a parameter value*

### 5.2.1 Detailed Description

Library for a PCF8574 GPIO expander.

#### Author

Frank Häfele ( [mail@frankhaefele.de](mailto:mail@frankhaefele.de))

#### Version

1.0.0

#### Date

2026-01-12

#### Copyright

Copyright (c) 2026

## 5.3 PCF8574-I2C.h

[Go to the documentation of this file.](#)

```
00001
00012 #pragma once
00013
00014 #define __PCF8574_I2C_H__
00015
00016 #include "Wire.h"
00017
00018 namespace PCF8574_I2C {
00019
00020     constexpr const char *PCF8574_LIB_VERSION {"1.0.0"};
00021
00026     constexpr int8_t PCF8574_STATE_OK           {0x00};
00027
00032     constexpr int8_t PCF8574_ERROR_PIN          {-1};
00033
```

```
00038     constexpr int8_t PCF8574_ERROR_I2C           {-2};  
00039  
00044     constexpr int8_t PCF8574_ERROR_VALUE         {-3};  
00045  
00046  
00047     class PCF8574 {  
00048         public:  
00049             PCF8574(const uint8_t address = 0x20, TwoWire* wire = &Wire);  
00050  
00064             int8_t begin() const;  
00065  
00074             int8_t resetPort();  
00075  
00085             int16_t readPin(int8_t pin = -1);  
00086  
00099             int8_t setPin(uint8_t pin, uint8_t value);  
00100  
00110             int8_t setPort(uint8_t value);  
00111  
00121             int8_t toggle(uint8_t mask);  
00122  
00131             int8_t rotateLeft();  
00132  
00141             int8_t rotateRight();  
00142  
00143  
00144         private:  
00145             // internal address of PCF8574 device  
00146             uint8_t _address;  
00147  
00148             // internal pointer to wire object  
00149             TwoWire* _wire;  
00150  
00151             // holds the last error  
00152             int8_t _error;  
00153  
00154             // holds the las readings from device  
00155             uint8_t _input{0};  
00156  
00157             // holds the last sending to device  
00158             uint8_t _output{0xFF};  
00159  
00160  
00169             bool isDevicePresent() const;  
00170  
00179             int8_t readPort();  
00180  
00190             int8_t writePort(uint8_t value);  
00191  
00199             bool isPinValid(int8_t pin);  
00200  
00201  
00202     };  
00203  
00204 }
```



# Index

begin  
    PCF8574\_I2C::PCF8574, 8

PCF8574  
    PCF8574\_I2C::PCF8574, 7

PCF8574-I2C Library, 1

PCF8574\_I2C::PCF8574, 7  
    begin, 8  
        PCF8574, 7  
    readPin, 8  
    resetPort, 8  
    rotateLeft, 9  
    rotateRight, 9  
    setPin, 9  
    setPort, 10  
    toggle, 10

readPin  
    PCF8574\_I2C::PCF8574, 8

resetPort  
    PCF8574\_I2C::PCF8574, 8

rotateLeft  
    PCF8574\_I2C::PCF8574, 9

rotateRight  
    PCF8574\_I2C::PCF8574, 9

setPin  
    PCF8574\_I2C::PCF8574, 9

setPort  
    PCF8574\_I2C::PCF8574, 10

src/PCF8574-I2C.cpp, 13

src/PCF8574-I2C.h, 13, 14

toggle  
    PCF8574\_I2C::PCF8574, 10