PCF8574-I2C

Generated on Mon Jan 19 2026 09:05:40 for PCF8574-I2C by Doxygen 1.9.8

Mon Jan 19 2026 09:05:40

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:

Chapter 3

File Index

3.1 File List

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

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 shiftLeft (uint8_t numberOfShifts=1U)

  shift bits of port to the left
- int8_t rotateLeft ()

  rotate Port to the left
- int8_t shiftRight (uint8_t numberOfShifts=1U)

  shift bits of port to the right
- 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

address	of PCF8574 device
wire	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

0	if device is connected
-2	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

pin	give pin number 0...7; or ommit for all read all port pins

Returns

int8_t return error code or value

Return values

>0	read was ok => return pin/Port value
<0	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

pin	number of port pin P0...P7
value	value of pin 0 or 1

Returns

int8_t status of the write command

Return values

0	successfull
-1	pin error; wrong pin number
-2	error on i2c connection
-3	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

value	for port to set

Returns

int8_t status of the write command

Return values

0	successfull
-2	error on i2c connection

4.1.2.8 shiftLeft()

```
int8_t PCF8574::shiftLeft (
            uint8_t numberOfShifts = 1U )
```

shift bits of port to the left

Parameters

numberOfShifts	amount of shifts to the left (optional)

Returns

int8_t status of the write command

Return values

0	successfull
-2	error on i2c connection
-3	error in number of shifts parameter

4.1.2.9 shiftRight()

```
int8_t PCF8574::shiftRight (
            uint8_t numberOfShifts = 1U )
```

shift bits of port to the right

Parameters

numberOfShifts	amount of shifts to the right (optional)

Returns

int8_t status of the write command

Return values

0	successfull
-2	error on i2c connection
-3	error in number of shifts parameter

4.1.2.10 toggle()

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

toggle pins by give a mask which pin to toggle

Parameters

mask	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)

Version

1.1.0

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** {"1.1.0"}
- 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)

Version

1.1.0

Date

2026-01-12

Copyright

Copyright (c) 2026

5.3 PCF8574-I2C.h

Go to the documentation of this file.

```
#pragma once

#define __PCF8574_I2C_H__

#include "Wire.h"

namespace PCF8574_I2C {

    constexpr const char *PCF8574_LIB_VERSION   {"1.1.0"};

    constexpr int8_t PCF8574_STATE_OK           {0x00};

    constexpr int8_t PCF8574_ERROR_PIN          {-1};

```

```
  constexpr int8_t PCF8574_ERROR_I2C          {-2};

  constexpr int8_t PCF8574_ERROR_VALUE        {-3};


  class PCF8574 {
     public:
        PCF8574(const uint8_t address = 0x20, TwoWire* wire = &Wire);

        int8_t begin() const;

        int8_t resetPort();

        int16_t readPin(int8_t pin = -1);

        int8_t setPin(uint8_t pin, uint8_t value);

        int8_t setPort(uint8_t value);

        int8_t toggle(uint8_t mask);

        int8_t shiftLeft(uint8_t numberOfShifts = 1U);

        int8_t rotateLeft();

        int8_t shiftRight(uint8_t numberOfShifts = 1U);

        int8_t rotateRight();


     private:
        // internal address of PCF8574 device
        uint8_t _address;

        // internal pointer to wire object
        TwoWire* _wire;

        // holds the last error
        int8_t _error;

        // holds the las readings from device
        uint8_t _input{0};

        // holds the last sending to device
        uint8_t _output{0xFF};


        bool isDevicePresent() const;

        int8_t readPort();

        int8_t writePort(uint8_t value);

        bool isPinValid(int8_t pin);


  };

}
```

Index