MCP23008-I2C

Generated by Doxygen 1.9.1

Chapter 1

MCP23008-I2C Library

[![Compile Examples]]()

Arduino Library for MCP23008, a 8-port GPIO exander

1.1 Contents

1.2 Library Documentation

The library documentaition ins mainly placed in the following pdf document refman.pdf.
Additionally in combination with the technical datasheet of microchip MCP23008-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 MCP23008-I2C library in the Code

Include the library

#include <MCP23008-I2C.h>

1.4 License

This library is licensed under MIT Licence.

MCP23008-I2C License

1.5 Helpful Links

- ESP8266-01-Adapter

Chapter 2

Namespace Index

2.1 Namespace List

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

Chapter 5

Namespace Documentation

5.1 MCP23008_Constants Namespace Reference

Namespace of MCP23008 Constants. Contains mainly the register description.

Variables

- constexpr uint8_t MCP23008_IODIR_REG {0x00}
  I/O Direction Register Address (IODIR)
- constexpr uint8_t MCP23008_IPOL_REG {0x01}
  Input Polarity Register (IPOL)
- constexpr uint8_t MCP23008_GPINTEN_REG {0x02}
  Interrupt-On-Change Control Register (GPINTEN)
- constexpr uint8_t MCP23008_DEFVAL_REG {0x03}
  Default Compare Register for Interrupt-On-Change.
- constexpr uint8_t MCP23008_INTCON_REG {0x04}
  Interrupt Control Register (INTCON)
- constexpr uint8_t MCP23008_IOCON_REG {0x05}
  Configuration Register (IOCON)
- constexpr uint8_t MCP23008_GPPU_REG {0x06}
  Pull-Up Resistor Configuration Register (GPPU)
- constexpr uint8_t MCP23008_INTF_REG {0x07}
  Interrupt Flag Register (INTF)
- constexpr uint8_t MCP23008_INTCAP_REG {0x08}
  Interrupt Capture Register (INTCAP)
- constexpr uint8_t MCP23008_GPIO_REG {0x09}
  Port Register (GPIO)
- constexpr uint8_t MCP23008_OLAT_REG {0x0A}
  Output Latch Register (OLAT)
- constexpr uint8_t MCP23008_IOCON_SEQOP {0x20}
  The Sequential Operation (SEQOP) bit.
- constexpr uint8_t MCP23008_IOCON_DISSLW {0x10}
  Slew Rate control bit for SDA output.
- constexpr uint8_t MCP23008_IOCON_ODR {0x04}
  The Open-Drain control bit (ODR)
- constexpr uint8_t MCP23008_IOCON_INTPOL {0x02}
  The Input Polarity Control bit (INTPOL)

5.1.1 Detailed Description

Namespace of MCP23008 Constants. Contains mainly the register description.

5.1.2 Variable Documentation

5.1.2.1 MCP23008_IOCON_DISSLW

```
constexpr uint8_t MCP23008_Constants::MCP23008_IOCON_DISSLW {0x10}  [constexpr]
```

Slew Rate control bit for SDA output.

The Slew Rate (DISSLW) bit controls the slew rate function on the SDA pin. If enabled, the SDA slew rate will be controlled when driving from a high to a low

5.1.2.2 MCP23008_IOCON_INTPOL

```
constexpr uint8_t MCP23008_Constants::MCP23008_IOCON_INTPOL {0x02}  [constexpr]
```

The Input Polarity Control bit (INTPOL)

The Interrupt Polarity (INTPOL) control bit sets the polarity of the INT pin. This bit is functional only when the ODR bit is cleared, configuring the INT pin as active push-pull.

5.1.2.3 MCP23008_IOCON_ODR

```
constexpr uint8_t MCP23008_Constants::MCP23008_IOCON_ODR {0x04}  [constexpr]
```

The Open-Drain control bit (ODR)

The Open-Drain (ODR) control bit enables/disables the INT pin for open-drain configuration

5.1.2.4 MCP23008_IOCON_SEQOP

```
constexpr uint8_t MCP23008_Constants::MCP23008_IOCON_SEQOP {0x20}  [constexpr]
```

The Sequential Operation (SEQOP) bit.

The Sequential Operation (SEQOP) controls the incrementing function of the Address Pointer. If the Address Pointer is disabled, the Address Pointer does not automatically increment after each byte is clocked during a serial transfer. This feature is useful when it is desired to continuously poll (read) or modify (write) a register.

5.2 MCP23008_I2C Namespace Reference

namespace of MCP23008. includes class declaration, errors, and states

Classes

- class MCP23008

  Class MCP23008.

Variables

- constexpr const char * **MCP23008_LIB_VERSION** {"1.0.0"}
- constexpr int8_t **MCP23008_STATE_OK** {0x00}
- constexpr int8_t **MCP23008_ERROR_PIN** {-1}
- constexpr int8_t **MCP23008_ERROR_I2C** {-2}
- constexpr int8_t **MCP23008_ERROR_VALUE** {-3}

5.2.1 Detailed Description

namespace of MCP23008. includes class declaration, errors, and states

Chapter 6

Class Documentation

6.1 MCP23008_I2C::MCP23008 Class Reference

Class MCP23008.

`#include <MCP23008-I2C.h>`

Public Member Functions

- MCP23008 (uint8_t address=0x20, TwoWire *wire=&Wire)
  Construct a new MCP23008 object.
- int8_t begin (bool inputPullUp=true) const
  init MCP23008 instance
- int8_t isConnected () const
  check connection status
- uint8_t getAddress () const
  Get the address of device.
- int setPinMode1 (uint8_t pin, uint8_t mode) const
  set pinMode of a single pin (IODIR)
- int write1 (uint8_t pin, uint8_t value) const
  write value for a single pin to OLAT register (OLAT)
- int read1 (uint8_t pin) const
  read value for a single pin from GPIO register (GPIO)
- int setPolarity (uint8_t pin, bool reversed) const
  Set the polarity of a single pin in the Input polarity register (IPOL)
- int getPolarity (uint8_t pin) const
  Get the polarity of a single pin of Input polarity register (IPOL)
- int setPullup (uint8_t pin, bool pullup) const
  Set the Pull-up register for on pin (GPPU)
- int getPullup (uint8_t pin) const
  Get the Pull-up register for one pin (GPPU)
- int8_t setPinMode8 (uint8_t mask) const
  set mask for pinMode in I/O register for all pins at once (INTCON)
- int getPinMode8 () const
  Read I/O Direction register (IODIR)

- int8_t write8 (uint8_t value) const

  write 8-bit value at once to Output Latch register (OLAT)
- int read8 () const

  read 8 bit at once from GPIO register (GPIO)
- int8_t setPolarity8 (uint8_t mask) const

  Set the polarity in 8-bit at once in Input polarity register (IPOL) If a bit is set, the corresponding GPIO register bit will reflect the inverted value on the pin.
- int getPolarity8 () const

  Get the polarity in 8-bit at once in Input polarity register (IPOL) If a bit is set, the corresponding GPIO register bit will reflect the inverted value on the pin.
- int8_t setPullup8 (uint8_t mask) const

  Set Pull-up for all 8 pins at once (GPPU)
- int getPullup8 () const

  Get Pull-up for all 8 pins at once (GPPU)
- int setInterrupt (uint8_t pin, uint8_t mode) const

  Set the Interrupt Control Register for specified pin (INTCON)
- int disableInterrupt (uint8_t pin) const

  Disable interrupt on specified pin (INTCON)
- int readInterruptFlagRegister () const

  Read the Interrupt Flag Register (INTF)
- int readInterruptCaptureRegister () const

  Read the interrupt capture register (INTCAP)
- int setInterruptPolarity (uint8_t polarity) const

  Set the Interrupt Polarity in IOCON Register.
- int getInterruptPolarity () const

  Read the Interrupt Polarity.

Protected Member Functions

- int8_t writeReg (uint8_t regAddress, uint8_t value) const

  I2C write value to MCP23008 register.
- int readReg (uint8_t regAddress) const

  I2c Read value of MCP23008 register.

Protected Attributes

- uint8_t _address

  address of MCP23008 device
- TwoWire * _wire

  pointer of wire instance

6.1.1 Detailed Description

Class MCP23008.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 MCP23008()

```
MCP23008::MCP23008 (
            uint8_t address = 0x20,
            TwoWire * wire = &Wire )
```

Construct a new MCP23008 object.

Parameters

address	optional address of I2C device; default = 0x20;
wire	optional address of Wire instance; default = &Wire;

6.1.3 Member Function Documentation

6.1.3.1 begin()

```
int8_t MCP23008::begin (
            bool inputPullUp = true ) const
```

init MCP23008 instance

Check connection status and set Pull-up resistors if needed (by default).

Parameters

inputPullUp	optional force all inputs with Pull-up; default = true;

Returns

status of begin

Return values

0	state OK
<0	error code

6.1.3.2 disableInterrupt()

```
int MCP23008::disableInterrupt (
            uint8_t pin ) const
```

Disable interrupt on specified pin (INTCON)

Parameters

pin	number of pin to clear the interrupt (0...7)

Returns

int status

Return values

0	state OK
<0	error code

6.1.3.3 getAddress()

```
uint8_t MCP23008_I2C::MCP23008::getAddress ( ) const  [inline]
```

Get the address of device.

Returns

uint8_t address

6.1.3.4 getInterruptPolarity()

```
int MCP23008::getInterruptPolarity ( ) const
```

Read the Interrupt Polarity.

Returns

int status

Return values

2	Opden-drain (ODR)
1	active-high
0	active-low
<0	error code

6.1.3.5 getPinMode8()

```
int MCP23008::getPinMode8 ( ) const
```

Read I/O Direction register (IODIR)

Returns

value of register

6.1.3.6 getPolarity()

```
int MCP23008::getPolarity (
            uint8_t pin ) const
```

Get the polarity of a single pin of Input polarity register (IPOL)

If a bit is set, the corresponding GPIO register bit will reflect the inverted value on the pin.

Parameters

pin	pin number of pin (0...7)

Returns

int status of polatity for pin

Return values

0	noninverted => GPIO pin will reflect the same logic state on input pin
1	inverted => GPIO pin will reflect the opposite logic state on input pin
<0	error code

6.1.3.7 getPolarity8()

```
int MCP23008::getPolarity8 ( ) const
```

Get the polarity in 8-bit at once in Input polarity register (IPOL) If a bit is set, the corresponding GPIO register bit will reflect the inverted value on the pin.

Returns

int status

Return values

>=0	register value
<0	error code

6.1.3.8 getPullup()

```
int MCP23008::getPullup (
            uint8_t pin ) const
```

Get the Pull-up register for one pin (GPPU)

If a bit is set and the corresponding pin is configured as an input, the corresponding PORT pin is internally pulled up with a 100 kOhm resistor.

Parameters

pin	pin number of pin (0...7)

Returns

int status of Pull-up for pin

Return values

0	Pull-up disabled
1	Pull-up enabled
<0	error code

6.1.3.9 getPullup8()

```
int MCP23008::getPullup8 ( ) const
```

Get Pull-up for all 8 pins at once (GPPU)

If a bit is set and the corresponding pin is configured as an input, the corresponding PORT pin is internally pulled up with a 100 kOhm resistor.

Returns

int status

Return values

>=0	register value
<0	error code

6.1.3.10 isConnected()

```
int8_t MCP23008::isConnected ( ) const
```

check connection status

Returns

int8_t status of connection

Return values

1	connection OK
<0	error code

6.1.3.11 read1()

```
int MCP23008::read1 (
            uint8_t pin ) const
```

read value for a single pin from GPIO register (GPIO)

The GPIO register reflects the value on the port. Reading from this register reads the port.

Parameters

pin	pin number of pin 0...7

Returns

int status of gpio register for pin

Return values

0	pin is in LOW state
1	pin is in HIGH
<0	error code

6.1.3.12 read8()

```
int MCP23008::read8 ( ) const
```

read 8 bit at once from GPIO register (GPIO)

The GPIO register reflects the value on the port. Reading from this register reads the port.

Returns

int status value of GPIO register

Return values

>=0	register value
<0	error code

6.1.3.13 readInterruptCaptureRegister()

```
int MCP23008::readInterruptCaptureRegister ( ) const
```

Read the interrupt capture register (INTCAP)

Returns

int read state of interrupt capture register

Return values

>=0	register value
<0	error code

6.1.3.14 readInterruptFlagRegister()

```
int MCP23008::readInterruptFlagRegister ( ) const
```

Read the Interrupt Flag Register (INTF)

The INTF register reflects the interrupt condition on the PORT pins of any pin that is enabled for interrupts via the GPINTEN register. A 'set' bit indicates that the associated pin caused the interrupt.

Returns

int read state of interrupt flag register

Return values

>=0	register value
<0	error code

6.1.3.15 readReg()

```
int MCP23008::readReg (
            uint8_t regAddress ) const  [protected]
```

I2c Read value of MCP23008 register.

Parameters

regAddress	address of specific register

Returns

int read status

Return values

>=0	register value
<0	error code

6.1.3.16 setInterrupt()

```
int MCP23008::setInterrupt (
            uint8_t pin,
            uint8_t mode ) const
```

Set the Interrupt Control Register for specified pin (INTCON)

If a bit is set, the corresponding I/O pin is compared against the associated bit in the DEFVAL register. If a bit value is clear, the corresponding I/O pin is compared against the previous value.

Parameters

pin	number of pin to set the interrupt (0...7)
mode	mode of interrupt (RISING, FALLING, CHANGE)

Returns

int status

Return values

0	state OK
<0	error code

6.1.3.17 setInterruptPolarity()

```
int MCP23008::setInterruptPolarity (
            uint8_t polarity ) const
```

Set the Interrupt Polarity in IOCON Register.

The Interrupt Polarity (INTPOL) control bit sets the polarity of the INT pin. This bit is functional only when the ODR bit is cleared, configuring the INT pin as active push-pull.

2 = Open-drain Output (ODR) 1 = active-high 0 = active-low

Parameters

polarity	value (2...0)

Returns

int status

Return values

0	state OK
<0	error code

6.1.3.18 setPinMode1()

```
int MCP23008::setPinMode1 (
            uint8_t pin,
            uint8_t mode ) const
```

set pinMode of a single pin (IODIR)

Parameters

pin	pin number of pin 0...7
mode	mode of pin (INPUT, INPUT_PULLUP, OUTPUT)

Returns

int status of write in IODIR register

Return values

0	state OK
<0	error code

6.1.3.19 setPinMode8()

```
int8_t MCP23008::setPinMode8 (
            uint8_t mask ) const
```

set mask for pinMode in I/O register for all pins at once (INTCON)

outputMode = 0

inputMode = 1

Bit pattern to set in hex: 0x10

in binary: 0b00010000

in decimal: 16

Parameters

mask	bit mask to set

Returns

int status of write to I/O Register

Return values

0	state OK
<0	error code

6.1.3.20 setPolarity()

```
int MCP23008::setPolarity (
            uint8_t pin,
            bool reversed ) const
```

Set the polarity of a single pin in the Input polarity register (IPOL)

If a bit is set, the corresponding GPIO register bit will reflect the inverted value on the pin.

Parameters

pin	pin number of pin 0...7
reversed	true or false

Returns

int status

Return values

0	state OK
<0	error code

6.1.3.21 setPolarity8()

```
int8_t MCP23008::setPolarity8 (
            uint8_t mask ) const
```

Set the polarity in 8-bit at once in Input polarity register (IPOL) If a bit is set, the corresponding GPIO register bit will reflect the inverted value on the pin.

Parameters

mask	to write

Bit pattern to set in hex: 0x10

in binary: 0b00010000

in decimal: 16

Returns

int status

Return values

0	state OK
<0	error code

6.1.3.22 setPullup()

```
int MCP23008::setPullup (
```

```
            uint8_t pin,
            bool pullup ) const
```

Set the Pull-up register for on pin (GPPU)

If a bit is set and the corresponding pin is configured as an input, the corresponding PORT pin is internally pulled up with a 100 kOhm resistor.

Parameters

pin	pin number of pin 0...7
pullup	set Pull-up true/false

Returns

int status

Return values

0	state OK
<0	error code

6.1.3.23 setPullup8()

```
int8_t MCP23008::setPullup8 (
            uint8_t mask ) const
```

Set Pull-up for all 8 pins at once (GPPU)

If a bit is set and the corresponding pin is configured as an input, the corresponding PORT pin is internally pulled up with a 100 kOhm resistor.

Parameters

mask	mask for Pull-up to set

Returns

int status

Return values

0	state OK
<0	error code

6.1.3.24 write1()

```
int MCP23008::write1 (
            uint8_t pin,
            uint8_t value ) const
```

write value for a single pin to OLAT register (OLAT)

The OLAT register provides access to the output latches. A write to this register modifies the output latches that modify the pins configured as outputs.

Parameters

pin	pin number of pin 0...7
value	to write 0/1

Returns

int statusof write in olat register

Return values

0	state OK
<0	error code

6.1.3.25 write8()

```
int8_t MCP23008::write8 (
            uint8_t value ) const
```

write 8-bit value at once to Output Latch register (OLAT)

The OLAT register provides access to the output latches. A read from this register results in a read of the OLAT and not the port itself. A write to this register modifies the output latches that modify the pins configured as outputs.

Parameters

value	value to write in hex, bin or decimal

Returns

int status of write to Output Latch register

Return values

0	state OK
<0	error code

6.1.3.26 writeReg()

```
int8_t MCP23008::writeReg (
            uint8_t regAddress,
            uint8_t value ) const  [protected]
```

I2C write value to MCP23008 register.

Parameters

regAddress	address of specific register
value	value to write

Returns

int8_t write status

Return values

=0	write OK
<0	error code

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

- src/MCP23008-I2C.h
- src/MCP23008-I2C.cpp

Chapter 7

File Documentation

7.1 src/MCP23008-Constants.h File Reference

MCP23008 Constants and Register short Descriptions.

Namespaces

- MCP23008_Constants

  Namespace of MCP23008 Constants. Contains mainly the register description.

Variables

- constexpr uint8_t MCP23008_Constants::MCP23008_IODIR_REG {0x00}

  I/O Direction Register Address (IODIR)
- constexpr uint8_t MCP23008_Constants::MCP23008_IPOL_REG {0x01}

  Input Polarity Register (IPOL)
- constexpr uint8_t MCP23008_Constants::MCP23008_GPINTEN_REG {0x02}

  Interrupt-On-Change Control Register (GPINTEN)
- constexpr uint8_t MCP23008_Constants::MCP23008_DEFVAL_REG {0x03}

  Default Compare Register for Interrupt-On-Change.
- constexpr uint8_t MCP23008_Constants::MCP23008_INTCON_REG {0x04}

  Interrupt Control Register (INTCON)
- constexpr uint8_t MCP23008_Constants::MCP23008_IOCON_REG {0x05}

  Configuration Register (IOCON)
- constexpr uint8_t MCP23008_Constants::MCP23008_GPPU_REG {0x06}

  Pull-Up Resistor Configuration Register (GPPU)
- constexpr uint8_t MCP23008_Constants::MCP23008_INTF_REG {0x07}

  Interrupt Flag Register (INTF)
- constexpr uint8_t MCP23008_Constants::MCP23008_INTCAP_REG {0x08}

  Interrupt Capture Register (INTCAP)
- constexpr uint8_t MCP23008_Constants::MCP23008_GPIO_REG {0x09}

  Port Register (GPIO)
- constexpr uint8_t MCP23008_Constants::MCP23008_OLAT_REG {0x0A}

  Output Latch Register (OLAT)

- constexpr uint8_t MCP23008_Constants::MCP23008_IOCON_SEQOP {0x20}
  The Sequential Operation (SEQOP) bit.
- constexpr uint8_t MCP23008_Constants::MCP23008_IOCON_DISSLW {0x10}
  Slew Rate control bit for SDA output.
- constexpr uint8_t MCP23008_Constants::MCP23008_IOCON_ODR {0x04}
  The Open-Drain control bit (ODR)
- constexpr uint8_t MCP23008_Constants::MCP23008_IOCON_INTPOL {0x02}
  The Input Polarity Control bit (INTPOL)

7.1.1 Detailed Description

MCP23008 Constants and Register short Descriptions.

Author

Frank Häfele

Date

27.12.2024

Version

1.0.0

See also

https://github.com/hasenradball/MCP23008-I2C

7.2 src/MCP23008-I2C.cpp File Reference

MCP23008 Function and Class Definitions.

```
#include "MCP23008-I2C.h"
#include "MCP23008-Constants.h"
```

7.2.1 Detailed Description

MCP23008 Function and Class Definitions.

Author

Frank Häfele

Date

27.12.2024

Version

1.0.0

See also

https://github.com/hasenradball/MCP23008-I2C

7.3 src/MCP23008-I2C.h File Reference

MCP23008 Declarations.

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

Classes

- class MCP23008_I2C::MCP23008

  Class MCP23008.

Namespaces

- MCP23008_I2C

  namespace of MCP23008. includes class declaration, errors, and states

Variables

- constexpr const char * **MCP23008_I2C::MCP23008_LIB_VERSION** {"1.0.0"}
- constexpr int8_t **MCP23008_I2C::MCP23008_STATE_OK** {0x00}
- constexpr int8_t **MCP23008_I2C::MCP23008_ERROR_PIN** {-1}
- constexpr int8_t **MCP23008_I2C::MCP23008_ERROR_I2C** {-2}
- constexpr int8_t **MCP23008_I2C::MCP23008_ERROR_VALUE** {-3}

7.3.1 Detailed Description

MCP23008 Declarations.

Author

Frank Häfele

Date

27.12.2024

Version

1.0.0

See also

https://github.com/hasenradball/MCP23008-I2C

Index