PinExtender Library

Generated by Doxygen 1.8.14

Contents

1 Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

2 File Index

2.1 File List

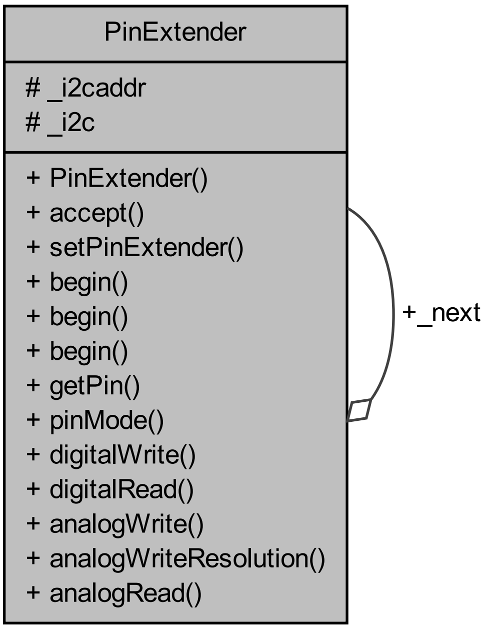
Here is a list of all files with brief descriptions:

3 Data Structure Documentation

3.1 PinExtender Class Reference

```
#include <PinExtender.h>
```

Collaboration diagram for PinExtender:



Public Member Functions

- PinExtender ()
- bool accept (uint32_t p)
- void setPinExtender (PinExtender ∗next)
- virtual void begin (TwoWire ∗MyWire, uint8_t addr)
- virtual void begin (uint8_t addr)
- virtual void begin (void)
- virtual uint32_t getPin (uint8_t channel_local)
- virtual void pinMode (uint32_t pin_arduino, uint8_t d)
- virtual void digitalWrite (uint32_t pin_arduino, uint8_t d)
- virtual uint8_t digitalRead (uint32_t pin_arduino)
- virtual void analogWrite (uint32_t pin_arduino, uint32_t ulValue)
- virtual void analogWriteResolution (int res)
- virtual uint32_t analogRead (uint32_t pin)

Data Fields

- PinExtender ∗ _next

Protected Attributes

- uint8_t _i2caddr
- TwoWire ∗ _i2c

3.1.1 Detailed Description

This file allow to add I/O by Wire(I2c) and continue to use standard Arduino api(Digital/analogRead/Write). This allow to keep you code very simple, and allow to continue to use generic(not core specific) library on extedned pin. For instance you can use a library for motor control with pin comming form a PCA9685 module that is a PWM extension module.

To do that you need to implemented a PinExtender daugther class and intantiate it and do setPinExtender(MyPin↩ExtObject); you can daisy chain it by MyPinExtObject.setPinExtender(MyPinExtObject2) and so on. to identify the pin use uint32_t variable or define like 0xaaaapppp. aaaa is the I2C device address for instance for a PCA9685, '0xaaaa' will be 0b1aaaaaa where aaaaaa fit with pin A[5:0], and pppp will be between 0 and 15 to identify pin LED[0:15] aaaa is the I2C device address for instance for a MCP23017, '0xaaaa' will be 0b0100aaa where aaa fit with pin A[2:0], and pppp will be between 0 and 7 to identify pin GPA[0:8] and between 8 and 15 to identify pin GPB[0:8]. For sure not all the api function will work correctly according to the limited capacity of these device, for example a analogRead will not work on previous device but analogWrite work on PCA9685 and have a basic behaviour on MCP23017. to get the pin number dynamically you can do : MyPinExtObject2.getPin(channel)

supported core : SAMD from zoubwolrd. To extend you core edit wiring_digital.c wiring_analog.c from your arduino core and add call to like it is done on zoubworld_arduino core:

- PinExtention_analogWriteResolution(int res);
- PinExtention_digitalWrite(uint32_t ulPin, uint32_t ulVal);
- PinExtention_digitalRead(uint32_t ulPin);
- PinExtention_pinMode(uint32_t ulPin, uint32_t ulMode);
- PinExtention_analogRead(uint32_t pin); PCA9685 and MCP23017 is already supported on dedicated library.daisy chain to manage pin

Definition at line 44 of file PinExtender.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 PinExtender()

```
PinExtender::PinExtender ( )
```

Definition at line 58 of file PinExtender.cpp.

3.1.3 Member Function Documentation

3.1.3.1 accept()

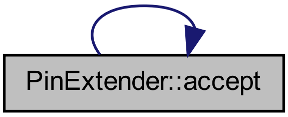
```
bool PinExtender::accept (
             uint32_t p )
```

Definition at line 46 of file PinExtender.cpp.

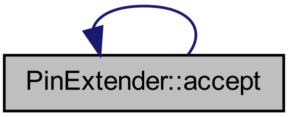
References _next, and accept().

Referenced by accept().

Here is the call graph for this function:



Here is the caller graph for this function:



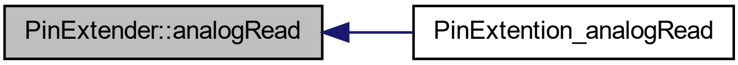
3.1.3.2 analogRead()

```
uint32_t PinExtender::analogRead (
             uint32_t pin )  [virtual]
```

Definition at line 89 of file PinExtender.cpp.

Referenced by PinExtention_analogRead().

Here is the caller graph for this function:



3.1.3.3 analogWrite()

```
void PinExtender::analogWrite (
             uint32_t pin_arduino,
             uint32_t ulValue )  [virtual]
```

Definition at line 87 of file PinExtender.cpp.

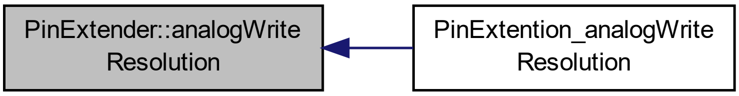
3.1.3.4 analogWriteResolution()

```
void PinExtender::analogWriteResolution (
             int res )  [virtual]
```

Definition at line 88 of file PinExtender.cpp.

Referenced by PinExtention_analogWriteResolution().

Here is the caller graph for this function:


PinExtender::analogWrite Resolution
PinExtention_analogWrite Resolution

3.1.3.5 begin() [1/3]

```
void PinExtender::begin (
             TwoWire * MyWire,
             uint8_t addr )  [virtual]
```

Definition at line 79 of file PinExtender.cpp.

3.1.3.6 begin() [2/3]

```
void PinExtender::begin (
             uint8_t addr )  [virtual]
```

Definition at line 80 of file PinExtender.cpp.

3.1.3.7 begin() [3/3]

```
void PinExtender::begin (
             void   )  [virtual]
```

Definition at line 81 of file PinExtender.cpp.

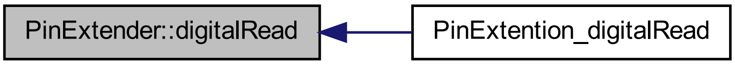
3.1.3.8 digitalRead()

```
uint8_t PinExtender::digitalRead (
             uint32_t pin_arduino )  [virtual]
```

Definition at line 86 of file PinExtender.cpp.

Referenced by PinExtention_digitalRead().

Here is the caller graph for this function:



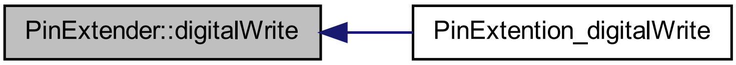
3.1.3.9 digitalWrite()

```
void PinExtender::digitalWrite (
             uint32_t pin_arduino,
             uint8_t d )  [virtual]
```

Definition at line 85 of file PinExtender.cpp.

Referenced by PinExtention_digitalWrite().

Here is the caller graph for this function:



3.1.3.10 getPin()

```
uint32_t PinExtender::getPin (
             uint8_t channel_local )  [virtual]
```

convert local pin number(uint8_t) of the board to arduino pin number(uint32_t)

Definition at line 83 of file PinExtender.cpp.

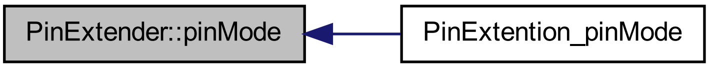
3.1.3.11 pinMode()

```
void PinExtender::pinMode (
             uint32_t pin_arduino,
             uint8_t d )  [virtual]
```

Definition at line 84 of file PinExtender.cpp.

Referenced by PinExtention_pinMode().

Here is the caller graph for this function:



3.1.3.12 setPinExtender()

```
void PinExtender::setPinExtender (
             PinExtender * next )
```

Definition at line 54 of file PinExtender.cpp.

References _next.

3.1.4 Field Documentation

3.1.4.1 _i2c

```
TwoWire* PinExtender::_i2c  [protected]
```

Definition at line 51 of file PinExtender.h.

3.1.4.2 _i2caddr

```
uint8_t PinExtender::_i2caddr  [protected]
```

Definition at line 50 of file PinExtender.h.

3.1.4.3 _next

```
PinExtender* PinExtender::_next
```

Definition at line 54 of file PinExtender.h.

Referenced by accept(), and setPinExtender().

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

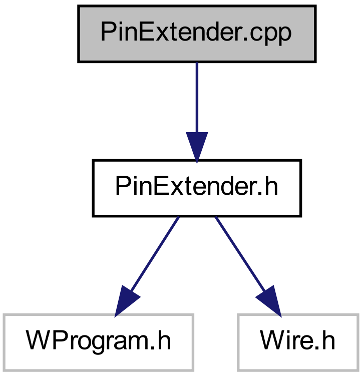
- PinExtender.h
- PinExtender.cpp

4 File Documentation

4.1 PinExtender.cpp File Reference

```
#include "PinExtender.h"
```

Include dependency graph for PinExtender.cpp:



Functions

- void PinExtention_analogWriteResolution (int res)
- void setPinExtender (PinExtender ∗next)
- void PinExtention_digitalWrite (uint32_t ulPin, uint32_t ulVal)
- int PinExtention_digitalRead (uint32_t ulPin)
- void PinExtention_pinMode (uint32_t ulPin, uint32_t ulMode)
- uint32_t PinExtention_analogRead (uint32_t pin)

Variables

- PinExtender ∗ pinExtention

4.1.1 Function Documentation

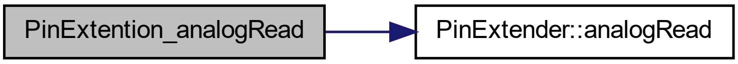
4.1.1.1 PinExtention_analogRead()

```
uint32_t PinExtention_analogRead (
             uint32_t pin )
```

Definition at line 37 of file PinExtender.cpp.

References PinExtender::analogRead(), and pinExtention.

Here is the call graph for this function:



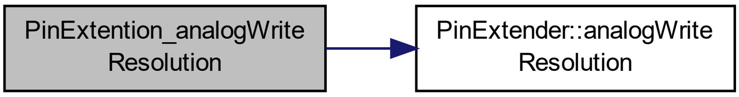
4.1.1.2 PinExtention_analogWriteResolution()

```
void PinExtention_analogWriteResolution (
             int res )
```

Definition at line 10 of file PinExtender.cpp.

References PinExtender::analogWriteResolution(), and pinExtention.

Here is the call graph for this function:


PinExtention_analogWrite Resolution
PinExtender::analogWrite Resolution

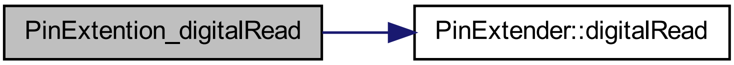
4.1.1.3 PinExtention_digitalRead()

```
int PinExtention_digitalRead (
             uint32_t ulPin )
```

Definition at line 26 of file PinExtender.cpp.

References PinExtender::digitalRead(), and pinExtention.

Here is the call graph for this function:



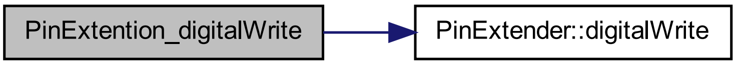
4.1.1.4 PinExtention_digitalWrite()

```
void PinExtention_digitalWrite (
             uint32_t ulPin,
             uint32_t ulVal )
```

Definition at line 21 of file PinExtender.cpp.

References PinExtender::digitalWrite(), and pinExtention.

Here is the call graph for this function:



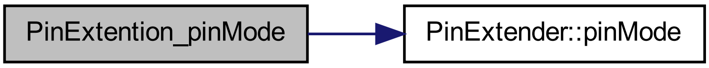
4.1.1.5 PinExtention_pinMode()

```
void PinExtention_pinMode (
            uint32_t ulPin,
            uint32_t ulMode )
```

Definition at line 32 of file PinExtender.cpp.

References pinExtention, and PinExtender::pinMode().

Here is the call graph for this function:



4.1.1.6 setPinExtender()

```
void setPinExtender (
            PinExtender * next )
```

Definition at line 17 of file PinExtender.cpp.

References pinExtention.

4.1.2 Variable Documentation

4.1.2.1 pinExtention

```
PinExtender* pinExtention
```

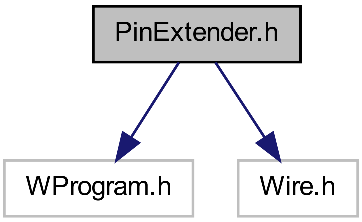
Definition at line 5 of file PinExtender.cpp.

Referenced by PinExtention_analogRead(), PinExtention_analogWriteResolution(), PinExtention_digitalRead(), PinExtention_digitalWrite(), PinExtention_pinMode(), and setPinExtender().

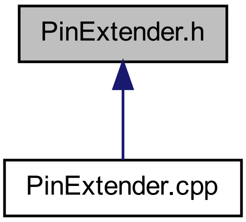
4.2 PinExtender.h File Reference

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

Include dependency graph for PinExtender.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class PinExtender

Functions

- void PinExtention_analogWriteResolution (int res)
- void PinExtention_digitalWrite (uint32_t ulPin, uint32_t ulVal)
- int PinExtention_digitalRead (uint32_t ulPin)
- void PinExtention_pinMode (uint32_t ulPin, uint32_t ulMode)
- uint32_t PinExtention_analogRead (uint32_t pin)
- void setPinExtender (PinExtender ∗next)

4.2.1 Function Documentation

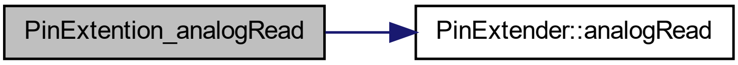
4.2.1.1 PinExtention_analogRead()

```
uint32_t PinExtention_analogRead (
             uint32_t pin )
```

Definition at line 37 of file PinExtender.cpp.

References PinExtender::analogRead(), and pinExtention.

Here is the call graph for this function:



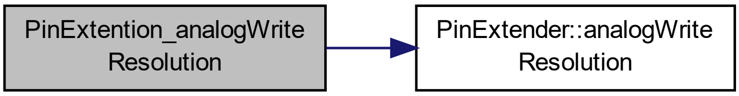
4.2.1.2 PinExtention_analogWriteResolution()

```
void PinExtention_analogWriteResolution (
             int res )
```

Definition at line 10 of file PinExtender.cpp.

References PinExtender::analogWriteResolution(), and pinExtention.

Here is the call graph for this function:



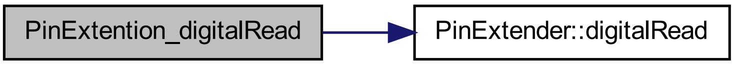
4.2.1.3 PinExtention_digitalRead()

```
int PinExtention_digitalRead (
             uint32_t ulPin )
```

Definition at line 26 of file PinExtender.cpp.

References PinExtender::digitalRead(), and pinExtention.

Here is the call graph for this function:



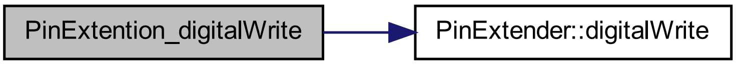
4.2.1.4 PinExtention_digitalWrite()

```
void PinExtention_digitalWrite (
             uint32_t ulPin,
             uint32_t ulVal )
```

Definition at line 21 of file PinExtender.cpp.

References PinExtender::digitalWrite(), and pinExtention.

Here is the call graph for this function:



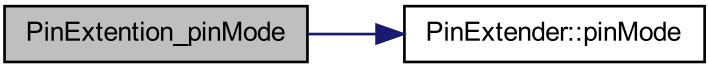
4.2.1.5 PinExtention_pinMode()

```
void PinExtention_pinMode (
             uint32_t ulPin,
             uint32_t ulMode )
```

Definition at line 32 of file PinExtender.cpp.

References pinExtention, and PinExtender::pinMode().

Here is the call graph for this function:



4.2.1.6 setPinExtender()

```
void setPinExtender (
             PinExtender * next )
```

Definition at line 17 of file PinExtender.cpp.

References pinExtention.

Index