Class Documentation

hook Class Reference

`#include <hook.h>`
Inherited by **inputExtend**.

Public Member Functions

- void **attachHook** (void(*eventHook)(void))
- void **detachHook** ()

Protected Member Functions

- void **callHook** ()

Detailed Description

Utility class providing inheritable methods to implement hooks.

Author:

Keegan Morrow

Version:

1 31.01.2014

Member Function Documentation

void hook::attachHook (void(*)(void) *eventHook*)`[inline]`

Attach the function to be called.

Parameters:

eventHook	Function pointer to the function to be attached. In the form void foo().

void hook::callHook ()`[inline]`, `[protected]`

Calles the hooked function if there is one. This should be placed in the function in the inheriting class to call the hook.

void hook::detachHook ()`[inline]`

Detach the hook.

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

- utility/hook.h

inputExtend Class Reference

```
#include <inputExtend.h>
```

Inherits **hook**.

Public Member Functions

- **inputExtend** (byte, byte, byte, byte)
- boolean **extendedRead** (byte)
- byte * **byteRead** ()
- byte **byteRead** (byte)
- void **update** ()
- byte * **getPtr** ()
- byte **getSize** ()

Public Attributes

- boolean **autoUpdate**

Protected Attributes

- byte **numChips**
- byte * **boards**

Additional Inherited Members

Detailed Description

Hardware interface class for the **inputExtend** board or other boards based on the 74HC165 chip.

Author:

Keegan Morrow

Version:

3 31.01.2014

Constructor & Destructor Documentation

inputExtend::inputExtend (byte *dataPin*, byte *clockPin*, byte *latchPin*, byte *numChips*)

Sets the direction of the io pins and allocates needed memory. This should be used to declare a global object.

Parameters:

dataPin	Pin number attached to the data pin
clockPin	Pin number attached to the data pin
latchPin	Pin number attached to the latch pin
numChips	Number of boards in use

Member Function Documentation

byte * inputExtend::byteRead ()

Reads the inputs to the input buffer. THIS IS OBSOLETE, DO NOT USE FOR NEW CODE!

Returns:

Pointer to the buffer

Deprecated:

byte inputExtend::byteRead (byte *boardNumber*)

Reads one board

Parameters:

boardNumber	Board to read from.

Returns:

bytewise data from the inputs

boolean inputExtend::extendedRead (byte *pinNumber*)

Reads an individual input pin. Pin numbers are sequential from the first pin on the first board. Pin 0 is board 0 input 0, pin 8 is board 1 input 0.

Parameters:

pinNumber	Input pin to read

Returns:

State of the pin. HIGH or LOW (true or false)

byte * inputExtend::getPtr ()

Returns:

pointer to the input buffer

byte inputExtend::getSize ()

Returns:

Size in bytes of the input buffer (same as the number of chips)

void inputExtend::update ()

Update the input buffer with the current state of the pins. This function is normally called automatically when needed. In a situation where very fast reads or snapshots are needed, autoUpdate can be set to false and this can be called manually.

Member Data Documentation

boolean inputExtend::autoUpdate

Determines if **inputExtend::update()** is called automatically. Default is true.

byte* inputExtend::boards `[protected]`

Input buffer, derived classes can modify the data, but should not change the pointer address.

byte inputExtend::numChips `[protected]`

Buffer size, derived classes should not modify this.

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

- inputExtend.h
- inputExtend.cpp