## HLib - An arm-based hardware library

Generated by Doxygen 1.8.5

Fri Sep 13 2013 13:05:49

# **Contents**

1	Clas	s Index		1
	1.1	Class I	.ist	1
2	Clas	s Docu	mentation	3
	2.1	hd4478	80_c Class Reference	3
		2.1.1	Constructor & Destructor Documentation	3
			2.1.1.1 hd44780_c	3
		2.1.2	Member Function Documentation	4
			2.1.2.1 Clear	4
			2.1.2.2 EntryMode	4
			2.1.2.3 Goto	4
			2.1.2.4 Home	4
			2.1.2.5 OnOff	4
			2.1.2.6 Print	5
			2.1.2.7 Print	5
			2.1.2.8 Print	5
			2.1.2.9 Print	5
			2.1.2.10 Print	6
			2.1.2.11 Start	6
	2.2	leds_c	Class Reference	6
		2.2.1	Constructor & Destructor Documentation	6
			2.2.1.1 leds_c	6
		2.2.2	Member Function Documentation	7
			2.2.2.1 Off	7
			2.2.2.2 On	8
			2.2.2.3 Set	8
			2.2.2.4 Toggle	8
	2.3	pins_c	Class Reference	8
		2.3.1	Constructor & Destructor Documentation	9
			2.3.1.1 pins_c	9
		2.3.2	Member Function Documentation	9

iv CONTENTS

		2.3.2.1	GetInput	9
		2.3.2.2	Release	9
		2.3.2.3	SetMode	9
		2.3.2.4	SetOutOne	10
		2.3.2.5	SetOutVal	10
		2.3.2.6	SetOutZero	10
2.4	port_p	in_t Struct	Reference	11
2.5	uart_c	Class Ref	erence	11
	2.5.1	Construc	ctor & Destructor Documentation	12
		2.5.1.1	uart_c	12
	2.5.2	Member	Function Documentation	12
		2.5.2.1	Get	12
		2.5.2.2	HasData	12
		2.5.2.3	Out	12
		2.5.2.4	Out	12
		2.5.2.5	Out	13
		2.5.2.6	Out	13
		2.5.2.7	Print	13
		2.5.2.8	Print	13
		2.5.2.9	Print	14
		2.5.2.10	Print	14
		2.5.2.11	Print	14
		2.5.2.12	Shutdown	14
		2.5.2.13	Start	14
		2.5.2.14	Start	15

# **Chapter 1**

# **Class Index**

## 1.1 Class List

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

nd44/80_c	
$leds\_c \ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	
pins_c	
port_pin_t	1
uart c	1

2 Class Index

## **Chapter 2**

## **Class Documentation**

## 2.1 hd44780\_c Class Reference

## **Public Member Functions**

• hd44780\_c ()

Construction function. Do nothing.

void Start (void)

Initialize all HD44780 control pins, set default mode, and return home.

· void Clear (void)

Send Clear Display command.

void Home (void)

Send Cursor Home command.

• void EntryMode (bool shift, bool increase)

Send Entry Mode command.

· void OnOff (bool displayOn, bool cursorOn, bool cursorBlink)

Send Display On Off command.

void Goto (uint8\_t charLine, uint8\_t charCol)

Set display cursor to a specified position.

· void Print (char printChar)

Print one character to the LCD.

void Print (char \*printStr)

Print one string to the LCD.

void Print (uint32\_t printNum, uint8\_t radix)

Print one unsigned number to the LCD.

void Print (uint32\_t printNum)

Print one unsigned number to the LCD in decimal.

void Print (int32\_t printNum)

Print one signed number to the LCD in decimal.

## 2.1.1 Constructor & Destructor Documentation

2.1.1.1 hd44780\_c::hd44780\_c ( void )

Construction function. Do nothing.

Returns

None

## 2.1.2 Member Function Documentation

2.1.2.1 void hd44780\_c::Clear ( void )

Send Clear Display command.

Returns

None

2.1.2.2 void hd44780\_c::EntryMode ( bool shift, bool increase )

Send Entry Mode command.

#### **Parameters**

shift	TRUE the display will be shifted, FALSE the display will not be shifted
increase	TRUE increase cursor position, FALSE decrease cursor position

#### Returns

None

2.1.2.3 void hd44780\_c::Goto ( uint8\_t charLine, uint8\_t charCol )

Set display cursor to a specified position.

## **Parameters**

charLine	LCD's line
charCol	LCD's column

## Returns

None

2.1.2.4 void hd44780\_c::Home ( void )

Send Cursor Home command.

Returns

None

2.1.2.5 void hd44780\_c::OnOff (bool displayOn, bool cursorOn, bool cursorBlink)

Send Display On Off command.

**Parameters** 

displayOn TRUE set the display on, FALSE set the display off

cursorOn	TRUE set the cursor on, FALSE set the cursor off
cursorBlink	TRUE the cursor is blinked, FALSE the cursor is not blinked

Returns

None

2.1.2.6 void hd44780\_c::Print ( char printChar )

Print one character to the LCD.

**Parameters** 

printChar	Character to print

Returns

None

2.1.2.7 void hd44780\_c::Print ( char \* printString )

Print one string to the LCD.

**Parameters** 

printString	String to print

Returns

None

2.1.2.8 void hd44780\_c::Print ( uint32\_t printNum, uint8\_t radix )

Print one unsigned number to the LCD.

**Parameters** 

printNum	Unsigned number to print
radix	Valid values are 2, 8, 10, 16

Returns

None

2.1.2.9 void hd44780\_c::Print ( uint32\_t printNum )

Print one unsigned number to the LCD in decimal.

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

#### **Parameters**

printNum Unsigned number to print

Returns

None

2.1.2.10 void hd44780\_c::Print ( int32\_t printNum )

Print one signed number to the LCD in decimal.

**Parameters** 

printNum Unsigned number to print

Returns

None

2.1.2.11 void hd44780\_c::Start ( void )

Initialize all HD44780 control pins, set default mode, and return home.

Return values

None

## 2.2 leds\_c Class Reference

## **Public Member Functions**

leds\_c (void)

Construction. Enable clock, set output mode for LEDs' pins.

void Set (uint8\_t ledIndex, bool val)

Set state of an LED.

void On (uint8\_t ledIndex)

Turn on an LED.

• void Off (uint8\_t ledIndex)

Turn off an LED.

• void Toggle (uint8\_t ledIndex)

Toggle state of an LED.

## 2.2.1 Constructor & Destructor Documentation

2.2.1.1 leds\_c::leds\_c ( void )

Construction. Enable clock, set output mode for LEDs' pins.

Returns

None

## 2.2.2 Member Function Documentation

2.2.2.1 void leds\_c::Off ( uint8\_t ledIndex )

Turn off an LED.

**Parameters** 

ledIndex Index or the LED

Returns

None

2.2.2.2 void leds\_c::On ( uint8\_t ledIndex )

Turn on an LED.

**Parameters** 

ledIndex	Index or the LED
----------	------------------

Returns

None

2.2.2.3 void leds\_c::Set ( uint8\_t ledIndex, bool val )

Set state of an LED.

**Parameters** 

ledIndex	Index or the LED
val	TRUE turn on the LED, FALSE turn off the LED

Returns

None

2.2.2.4 void leds\_c::Toggle ( uint8\_t ledIndex )

Toggle state of an LED.

**Parameters** 

ledIndex
----------

Returns

None

## 2.3 pins\_c Class Reference

## **Public Member Functions**

• pins\_c (void)

Construction function. Enable GPIO clocks.

• void Release (uint8\_t pinIndex)

Release one pin to input floating state.

• err\_t SetMode (uint8\_t pinIndex, pin\_mode\_t mode, pin\_type\_t type)

Set operation mode for one pin.

void SetOutVal (uint8\_t pinIndex, bool val)

Set output register to a specified value.

void SetOutOne (uint8\_t pinIndex)

Set output register to one.

void SetOutZero (uint8\_t pinIndex)

Set output register to zero.

bool GetInput (uint8\_t pinIndex)

Get digital electronic value at one pin.

## 2.3.1 Constructor & Destructor Documentation

```
2.3.1.1 pins_c::pins_c ( void )
```

Construction function. Enable GPIO clocks.

Returns

None

## 2.3.2 Member Function Documentation

2.3.2.1 bool pins\_c::GetInput ( uint8\_t pinIndex )

Get digital electronic value at one pin.

**Parameters** 

pinIndex	Index of the pin
----------	------------------

#### **Return values**

true	pin is 1
false	pin is 0

## 2.3.2.2 void pins\_c::Release ( uint8\_t pinIndex )

Release one pin to input floating state.

**Parameters** 

pinIndex Index of the pin
---------------------------

Returns

None

2.3.2.3 err\_t pins\_c::SetMode ( uint8\_t pinIndex, pin\_mode\_t mode, pin\_type\_t type )

Set operation mode for one pin.

#### **Parameters**

pinIndex	Index of the pin
mode	Operation mode. Please refer the table pin map for valid configuration
type	Type of pin

#### Returns

HL\_OK, HL\_INVALID

#### Attention

Please select type corresponding with selecte mode. If you make a wrong pin configuration, your system may behave in unpredictable manner. Thereforce, we strongly recommend that you check return value of the function and make sure it is HL\_OK

2.3.2.4 void pins\_c::SetOutOne ( uint8\_t pinIndex )

Set output register to one.

#### **Parameters**

pinIndex	Index of the pin
----------	------------------

#### Returns

None

#### Attention

The actual value on the pin is also depend on pinMode, pull-up/pull-down resistor. Please ensure the pin is set as Output mode and pull-up/pull-down resistor is configured appropriately

2.3.2.5 void pins\_c::SetOutVal ( uint8\_t pinIndex, bool val )

Set output register to a specified value.

#### **Parameters**

pinIndex	Index of the pin
val	TRUE set one, FALSE set zero

#### Returns

None

## Attention

The actual value on the pin is also depend on pinMode, pull-up/pull-down resistor. Please ensure the pin is set as Output mode and pull-up/pull-down resistor is configured appropriately

2.3.2.6 void pins\_c::SetOutZero ( uint8\_t pinIndex )

Set output register to zero.

#### **Parameters**

pinIndex	Index of the pin	

#### Returns

None

#### Attention

The actual value on the pin is also depend on pinMode, pull-up/pull-down resistor. Please ensure the pin is set as Output mode and pull-up/pull-down resistor is configured appropriately

## 2.4 port\_pin\_t Struct Reference

#### **Public Attributes**

- GPIO\_TypeDef \* port
- uint16\_t pin

## 2.5 uart\_c Class Reference

#### **Public Member Functions**

uart\_c (uint8\_t uartNum)

Construction function.

err\_t Start (uint32\_t baudRate, uint16\_t stopBit)

Start serial communicating.

err\_t Start (uint32\_t baudRate)

Start serial communicating in default mode with 8 data-bit, 1 stop-bit, no-parity.

err\_t Shutdown (void)

Release I/O pins, stop UART clock.

err\_t Print (char outChar)

Send one character.

• err\_t Print (char \*outStr)

Send one string.

• err\_t Print (uint32\_t outNum, uint8\_t radix)

Convert one unsigned interger into string and then send it.

err\_t Print (uint32\_t outNum)

Convert one unsigned interger into string and then send it with default radix (decimal)

err\_t Print (int32\_t outNum)

Convert one signed interger into decimal string and then send it.

err\_t Out (uint8\_t outNum)

Send one raw 8-bit number.

err\_t Out (uint16\_t outNum)

Send one raw 16-bit number. The low-order part is sent first.

err\_t Out (uint32\_t outNum)

Send one raw 32-bit number. The low-order part is sent first.

err\_t Out (uint8\_t outBuf[], uint32\_t bufLen)

Send one buffer.

uint8\_t Get (void)

Get one received byte in receiving buffer.

bool HasData (void)

Check whether there is a new received byte in receiving buffer.

## 2.5.1 Constructor & Destructor Documentation

2.5.1.1 uart\_c::uart\_c ( uint8\_t uartNum )

Construction function.

**Parameters** 

uartNum	UART will be used.

Returns

None

#### 2.5.2 Member Function Documentation

2.5.2.1 uint8\_t uart\_c::Get ( void )

Get one received byte in receiving buffer.

Returns

Received data

## 2.5.2.2 bool uart\_c::HasData (void)

Check whether there is a new received byte in receiving buffer.

**Return values** 

TRUE	has new data
FALSE	no new data

2.5.2.3 err\_t uart\_c::Out ( uint8\_t outNum )

Send one raw 8-bit number.

Parameters

outNum	8-bit number to send

Returns

HL\_OK, HL\_NOT\_STARTED, HL\_INVALID

2.5.2.4 err\_t uart\_c::Out ( uint16\_t outNum )

Send one raw 16-bit number. The low-order part is sent first.

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

#### **Parameters**

outNum	16-bit number to send.
--------	------------------------

#### Returns

HL\_OK, HL\_NOT\_STARTED, HL\_INVALID

2.5.2.5 err\_t uart\_c::Out ( uint32\_t outNum )

Send one raw 32-bit number. The low-order part is sent first.

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

#### **Parameters**

outNum	32-bit number to send.

#### Returns

HL\_OK, HL\_NOT\_STARTED, HL\_INVALID

2.5.2.6 err\_t uart\_c::Out ( uint8\_t outBuf[], uint32\_t bufLen )

Send one buffer.

#### **Parameters**

	outBuf	Buffer to send
Ì	bufLen	Length in byte of the buffer

## Returns

HL OK, HL NOT STARTED, HL INVALID

2.5.2.7 err\_t uart\_c::Print ( char outChar )

Send one character.

#### **Parameters**

Ontonal Character to Seria		
----------------------------	--	--

#### Returns

HL\_OK, HL\_NOT\_STARTED, HL\_INVALID

2.5.2.8 err\_t uart\_c::Print ( char \* outStr )

Send one string.

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

#### **Parameters**

outStr	String to send
--------	----------------

#### Returns

HL OK, HL NOT STARTED, HL INVALID

2.5.2.9 err\_t uart\_c::Print ( uint32\_t outNum, uint8\_t radix )

Convert one unsigned interger into string and then send it.

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

#### **Parameters**

outNum	Unsigned interger number to send
radix	Valid values are 2, 8, 10, 16

#### Returns

HL\_OK, HL\_NOT\_STARTED, HL\_INVALID

2.5.2.10 err\_t uart\_c::Print ( uint32\_t outNum )

Convert one unsigned interger into string and then send it with default radix (decimal)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

#### **Parameters**

outNum	Unsigned interger number to send

#### Returns

HL OK, HL NOT STARTED, HL INVALID

2.5.2.11 err\_t uart\_c::Print ( int32\_t outNum )

Convert one signed interger into decimal string and then send it.

#### **Parameters**

outNum	Signed interger number to send @ HL_OK, HL_NOT_STARTED, HL_INVALID

2.5.2.12 err\_t uart\_c::Shutdown ( void )

Release I/O pins, stop UART clock.

Returns

HL\_OK, HL\_INVALID

2.5.2.13 err\_t uart\_c::Start ( uint32\_t baudRate, uint16\_t stopBit )

Start serial communicating.

#### **Parameters**

baudRate	UART's baud rate
stopBit	UART's stopBit

## Returns

 $\mathsf{HL\_OK}$  ,  $\mathsf{HL\_INVALID}$ 

2.5.2.14 err\_t uart\_c::Start ( uint32\_t baudRate )

Start serial communicating in default mode with 8 data-bit, 1 stop-bit, no-parity.

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

## **Parameters**

baudRate	UART's baud rate
----------	------------------

## Returns

HL\_OK, HL\_INVALID

# Index

SetMode, 9

Clear	SetOutOne, 10
hd44780_c, 4	SetOutVal, 10
	SetOutZero, 10
EntryMode	port_pin_t, 11
hd44780_c, 4	Print
0-4	hd44780_c, 5, 6
Get	uart_c, 13, 14
uart_c, 12	
GetInput	Release
pins_c, 9	pins_c, 9
Goto	0-1
hd44780_c, 4	Set
HasData	leds_c, 8
uart_c, 12	SetMode
hd44780_c, 3	pins_c, 9
Clear, 4	SetOutOne
EntryMode, 4	pins_c, 10
Goto, 4	SetOutVal
hd44780_c, 3	pins_c, 10
hd44780_c, 3	SetOutZero
Home, 4	pins_c, 10
OnOff, 4	Shutdown
Print, 5, 6	uart_c, 14
Start, 6	Start
Home	hd44780_c, 6
hd44780_c, 4	uart_c, 14, 15
1144700_0, 4	Toggle
leds_c, 6	leds_c, 8
leds_c, 6	1eus_c, 0
leds_c, 6	uart_c, 11
Off, 7	Get, 12
On, 8	HasData, 12
Set, 8	Out, 12, 13
Toggle, 8	
roggie, o	Print, 13, 14
Off	Print, 13, 14 Shutdown, 14
Off	Print, 13, 14 Shutdown, 14 Start, 14, 15
	Print, 13, 14 Shutdown, 14 Start, 14, 15 uart_c, 12
Off leds_c, 7	Print, 13, 14 Shutdown, 14 Start, 14, 15
Off leds_c, 7	Print, 13, 14 Shutdown, 14 Start, 14, 15 uart_c, 12
Off leds_c, 7 On leds_c, 8	Print, 13, 14 Shutdown, 14 Start, 14, 15 uart_c, 12
Off leds_c, 7 On leds_c, 8 OnOff	Print, 13, 14 Shutdown, 14 Start, 14, 15 uart_c, 12
Off leds_c, 7 On leds_c, 8 OnOff hd44780_c, 4	Print, 13, 14 Shutdown, 14 Start, 14, 15 uart_c, 12
Off leds_c, 7 On leds_c, 8 OnOff hd44780_c, 4 Out uart_c, 12, 13	Print, 13, 14 Shutdown, 14 Start, 14, 15 uart_c, 12
Off leds_c, 7 On leds_c, 8 OnOff hd44780_c, 4 Out uart_c, 12, 13 pins_c, 8	Print, 13, 14 Shutdown, 14 Start, 14, 15 uart_c, 12
Off leds_c, 7 On leds_c, 8 OnOff hd44780_c, 4 Out uart_c, 12, 13 pins_c, 8 GetInput, 9	Print, 13, 14 Shutdown, 14 Start, 14, 15 uart_c, 12
Off leds_c, 7 On leds_c, 8 OnOff hd44780_c, 4 Out uart_c, 12, 13 pins_c, 8	Print, 13, 14 Shutdown, 14 Start, 14, 15 uart_c, 12