

LCD library for AVR uC
00.00.01

Generated by Doxygen 1.8.2

Sun Nov 11 2012 19:12:33

Contents

| | | |
|--------------|----------------------------------|-----------|
| 1 | Main Page | 1 |
| 1.1 | Project | 1 |
| 1.2 | Specifications | 1 |
| 1.2.1 | Language | 2 |
| 1.2.2 | Target | 2 |
| 1.2.3 | Tested | 2 |
| 1.3 | Files | 2 |
| 1.4 | Dependency | 2 |
| 1.5 | More informations | 2 |
| 1.6 | Disclaimer | 2 |
| 2 | Todo List | 3 |
| 3 | Bug List | 3 |
| 4 | File Documentation | 3 |
| 4.1 | Icd.c File Reference | 3 |
| 4.1.1 | Detailed Description | 4 |
| 4.1.2 | Function Documentation | 4 |
| 4.2 | Icd.h File Reference | 7 |
| 4.2.1 | Detailed Description | 9 |
| 4.2.2 | Function Documentation | 9 |
| Index | | 13 |

1 Main Page

1.1 Project

LCD functions based on the HD4470 / KS0066u chips

Author

Patrice Nadeau
email : patricen@telwarwick.net
www : <http://nadeaup.homeip.net:8080/>

1.2 Specifications

1.2.1 Language

- C

1.2.2 Target

- ATmega48
- ATmega88
- ATmega168
- ATmega328

1.2.3 Tested

- ATmega168

1.3 Files

- [lcd.h](#)
- [lcd.c](#)

1.4 Dependency

- [utils.h](#)

1.5 More informations

Datasheet Datasheet Newhaven Display (NHD-0216BZ-RN-YBW) March 6, 2008

1.6 Disclaimer

Disclaimer Copyright (c) 2010, 2011 Patrice Nadeau

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Patrice Nadeau nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL Patrice Nadeau BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2 Todo List

File `lcd.h`

`utils.h`, `delay.h` should be replaced

Global `lcd_DisplayOnOff` (`uint8_t display`, `uint8_t cursor`, `uint8_t blink`)

Optimise the code

Global `lcd_EnterMode` (`uint8_t rightleft`, `uint8_t shift`)

Optimise the code

Global `lcd_FunctionSet` (`uint8_t line`, `uint8_t font`)

use of a case to suport 4 lines

use of a case to suport other fonts

3 Bug List

Global `lcd_BusyFlag` (`void`)

Can't work, `_Send` always return 0

need to be modified for 4 bits

Global `lcd_ReadData` (`void`)

Can't work, `_Send` always return 0

4 File Documentation

4.1 `lcd.c` File Reference

LCD on AVR.

Functions

- static void `_Clock` (`void`)
- static `uint8_t` `_Send` (`uint8_t` command)
- static void `_SendData` (`uint8_t` data)
- static void `_SendCommand` (`uint8_t` command)
- void `lcd_InitPort` (`void`)
 - Initialize the data and command register for output.*
- void `lcd_FunctionSet` (`uint8_t` line, `uint8_t` font)

- Set the number of lines and font.*

 - void [lcd_ClearDisplay](#) (void)

Clear the display and set DDRA to 20H and DDRAM to 00h.
- void [lcd_EnterMode](#) (uint8_t rightleft, uint8_t shift)

lcd_EnterMode
- void [lcd_DisplayOnOff](#) (uint8_t display, uint8_t cursor, uint8_t blink)

Turn on/off the display, cursor and blink of the LCD.
- void [lcd_DDRAM](#) (uint8_t position)

Place the cursor at position in display memory.
- void [lcd_CGRAM](#) (uint8_t address)

Set the GCRAM address.
- void [lcd_WriteData](#) (uint8_t character)

Display character on the lcd.
- uint8_t [lcdShift](#) (uint8_t cursor_display, uint8_t direction)
- void [lcd_WriteString](#) (char *string)

Write a string to the LCD.
- void [lcdReturnHome](#) (void)
- uint8_t [lcd_BusyFlag](#) (void)

Return the busy flag and the address counter.
- uint8_t [lcd_ReadData](#) (void)

Read the 8 bit data from position set earlier by lcdCGRAM or lcdDDRAM.
- void [lcd_ClearLine](#) (uint8_t line)

Clear every character on line and place cursor at beginning of line.
- void [lcd_Byte](#) (uint8_t data)

Display the binary version of a byte.
- void [lcd_DoubleByte](#) (uint16_t data)

Display the binary version of 16 bits.

4.1.1 Detailed Description

LCD on AVR.

Author

Patrice Nadeau

See Also

[lcd.h](#)

4.1.2 Function Documentation

4.1.2.1 uint8_t lcd_BusyFlag (void)

Return the busy flag and the address counter.

Returns

Busy flag, address counter

- bit **XXXXX** is busy flag () 1 busy, 0 rdy to accept command
- bit 6-0 is the address counter

Bug Can't work, `_Send` always return 0
need to be modified for 4 bits

4.1.2.2 void lcd.Byte (uint8_t data)

Display the binary version of a byte.

Parameters

| | | |
|-----------|-------------|-----------------|
| <i>in</i> | <i>data</i> | Byte to display |
|-----------|-------------|-----------------|

Note

Mostly for debugging

4.1.2.3 void lcd.CGRAM (uint8_t address)

Set the CGRAM address.

Parameters

| | | |
|-----------|----------------|---------------|
| <i>in</i> | <i>address</i> | (last 6 bits) |
|-----------|----------------|---------------|

4.1.2.4 void lcd.ClearLine (uint8_t line)

Clear every character on line and place cursor at beginning of line.

Parameters

| | | |
|-----------|-------------|--|
| <i>in</i> | <i>line</i> | The line number to clear <ul style="list-style-type: none"> • <code>lcd_LINE_x</code> |
|-----------|-------------|--|

4.1.2.5 void lcd.DDRAM (uint8_t position)

Place the cursor at *position* in display memory.

Parameters

| | | |
|-----------|-----------------|--------------------|
| <i>in</i> | <i>position</i> | Position in memory |
|-----------|-----------------|--------------------|

Note

First line 00H to 27H, second line 40H to 67H

4.1.2.6 void lcd_DisplayOnOff (uint8_t *display*, uint8_t *cursor*, uint8_t *blink*)

Turn on/off the display, cursor and blink of the LCD.

Todo Optimise the code

4.1.2.7 void lcd_DoubleByte (uint16_t *data*)

Display the binary version of 16 bits.

Parameters

| | | |
|-----------|-------------|------------------------|
| <i>in</i> | <i>data</i> | Double byte to display |
|-----------|-------------|------------------------|

Note

Mostly for debugging

4.1.2.8 void lcd_EnterMode (uint8_t *rightleft*, uint8_t *shift*)

lcd_EnterMode

Todo Optimise the code

4.1.2.9 void lcd_FunctionSet (uint8_t *line*, uint8_t *font*)

Set the number of *lines* and *font*.

Note

Only support 2 lines

Todo use of a case to suport 4 lines

Todo use of a case to suport other fonts

4.1.2.10 void lcd_InitPort (void)

Initialize the data and command register for output.

Returns**Note**

Maybe should provide software reset

4.1.2.11 uint8_t lcd_ReadData (void)

Read the 8 bit data from position set earlier by lcdCGRAM or lcdDDRAM.

Returns

uint8_t Data

Note

Address counter will also be affected (like a write)

Bug Can't work, _Send always return 0

4.1.2.12 void lcd_WriteData (uint8_t *character*)

Display *character* on the lcd.

Parameters

| | | |
|----|------------------|----------------------|
| in | <i>character</i> | Character to display |
|----|------------------|----------------------|

Note

Works for the first 127 ASCII codes

4.1.2.13 void lcd_WriteString (char * *string*)

Write a string to the LCD.

Parameters

| | | |
|----|----------------|---------------------|
| in | <i>*string</i> | Pointer to a string |
|----|----------------|---------------------|

Note

Do not check the length

4.2 lcd.h File Reference

LCD on AVR.

Macros

For the display functions

- #define `lcd_LINES_1` 1
1 line
- #define `lcd_LINES_2` 2
2 lines
- #define `lcd_LINES_4` 4
Not implemented yet.

- #define `lcd_FONT_8` 8
5 x 8 font
- #define `lcd_FONT_11` 11
8 x 11
- #define `lcd_SHIFT_DISPLAY_OFF` 0
Shift the whole display.
- #define `lcd_SHIFT_DISPLAY_RIGHTLEFT` 0
Direction to shift the display if enable.
- #define `lcd_DISPLAY_OFF` 0
1 line
- #define `lcd_CURSOR_OFF` 0
Use !lcd_CURSOR_OFF to enable.
- #define `lcd_CURSOR_BLINK_OFF` 0
Use !lcd_CURSOR_BLINK_OFF to enable.

Functions

- void `lcd_InitPort` (void)
Initialize the data and command register for output.
- void `lcd_FunctionSet` (uint8_t line, uint8_t font)
Set the number of lines and font.
- void `lcd_ClearDisplay` (void)
Clear the display and set DDRA to 20H and DDRAM to 00h.
- void `lcd_EnterMode` (uint8_t rightleft, uint8_t shift)
lcd_EnterMode
- void `lcd_DisplayOnOff` (uint8_t display, uint8_t cursor, uint8_t blink)
Turn on/off the display, cursor and blink of the LCD.
- void `lcd_DDRAM` (uint8_t position)
Place the cursor at position in display memory.
- void `lcd_WriteData` (uint8_t character)
Display character on the lcd.
- void `lcd_CGRAM` (uint8_t address)
Set the GGRAM address.
- uint8_t `lcd_Shift` (uint8_t wich, uint8_t direction)
Select which display or cursor move and to the right or left.
- void `lcd_WriteString` (char *string)
Write a string to the LCD.
- void `lcd_ReturnHome` (void)
Return cursor at 00H, return display original status if shifted.
- uint8_t `lcd_BusyFlag` (void)
Return the busy flag and the address counter.
- uint8_t `lcd_ReadData` (void)
Read the 8 bit data from position set earlier by lcdCGRAM or lcdDDRAM.
- void `lcd_ClearLine` (uint8_t line)
Clear every character on line and place cursor at beginning of line.
- void `lcd_Byte` (uint8_t data)
Display the binary version of a byte.
- void `lcd_DoubleByte` (uint16_t data)
Display the binary version of 16 bits.

4.2.1 Detailed Description

LCD on AVR. LCD functions based on the HD4470 / KS0066u chips (Datasheet Newhaven Display (NHD-0216BZ-RN-YBW) March 6, 2008)

It take for granted that the Data pins on the LCD and on the AVR are in the same order and all on the same port

Author

Patrice Nadeau

Note

For ATmega48PA/88PA/168PA/328P

Todo utils.h, delay.h should be replaced

4.2.2 Function Documentation

4.2.2.1 uint8_t lcd_BusyFlag (void)

Return the busy flag and the address counter.

Returns

Busy flag, address counter

- bit **XXXX** is busy flag () 1 busy, 0 rdy to accept command
- bit 6-0 is the address counter

Bug Can't work, _Send always return 0
need to be modified for 4 bits

4.2.2.2 void lcd_Byte (uint8_t data)

Display the binary version of a byte.

Parameters

| | | |
|-----------|-------------|-----------------|
| <i>in</i> | <i>data</i> | Byte to display |
|-----------|-------------|-----------------|

Note

Mostly for debugging

4.2.2.3 void lcd_CGRAM (uint8_t address)

Set the GCRAM address.

Parameters

| | | |
|-----------|----------------|---------------|
| <i>in</i> | <i>address</i> | (last 6 bits) |
|-----------|----------------|---------------|

4.2.2.4 void lcd.ClearLine (uint8_t line)

Clear every character on line and place cursor at beginning of line.

Parameters

| | | |
|-----------|-------------|---|
| <i>in</i> | <i>line</i> | The line number to clear <ul style="list-style-type: none"> • lcd_LINE_x |
|-----------|-------------|---|

4.2.2.5 void lcd.DDRAM (uint8_t position)

Place the cursor at *position* in display memory.

Parameters

| | | |
|-----------|-----------------|--------------------|
| <i>in</i> | <i>position</i> | Position in memory |
|-----------|-----------------|--------------------|

Note

First line 00H to 27H, second line 40H to 67H

4.2.2.6 void lcd.DisplayOnOff (uint8_t display, uint8_t cursor, uint8_t blink)

Turn on/off the display, cursor and blink of the LCD.

Parameters

| | | |
|-----------|----------------|--|
| <i>in</i> | <i>display</i> | Show the display <ul style="list-style-type: none"> • lcd_DISPLAY_OFF • !lcd_DISPLAY_OFF |
| <i>in</i> | <i>cursor</i> | Show the cursor <ul style="list-style-type: none"> • lcd_CURSOR_OFF • !lcd_CURSOR_OFF |
| <i>in</i> | <i>blink</i> | Blink the cursor <ul style="list-style-type: none"> • lcd_CURSOR_BLINK_OFF • !lcd_CURSOR_BLINK_OFF |

Todo Optimise the code

4.2.2.7 void lcd.DoubleByte (uint16_t data)

Display the binary version of 16 bits.

Parameters

| | | |
|-----------|-------------|------------------------|
| <i>in</i> | <i>data</i> | Double byte to display |
|-----------|-------------|------------------------|

Note

Mostly for debugging

4.2.2.8 void lcd_EnterMode (uint8_t *rightleft*, uint8_t *shift*)

lcd_EnterMode

Parameters

| | | |
|-----------|------------------|--|
| <i>in</i> | <i>rightleft</i> | Direction to shift the display <ul style="list-style-type: none"> • lcd_SHIFT_DISPLAY_RIGHTLEFT • !lcd_SHIFT_DISPLAY_RIGHTLEFT |
| <i>in</i> | <i>shift</i> | Show the display <ul style="list-style-type: none"> • lcd_SHIFT_DISPLAY_OFF • !lcd_SHIFT_DISPLAY_OFF |

Todo Optimise the code

4.2.2.9 void lcd_FunctionSet (uint8_t *line*, uint8_t *font*)

Set the number of *lines* and *font*.

Parameters

| | | |
|-----------|-------------|---|
| <i>in</i> | <i>line</i> | Number of lines <ul style="list-style-type: none"> • lcd_LINES_x |
| <i>in</i> | <i>font</i> | Font to use <ul style="list-style-type: none"> • lcd_FONT_x |

Note

Only support 2 lines

Todo use of a case to suport 4 lines

Todo use of a case to suport other fonts

4.2.2.10 void lcd_InitPort (void)

Initialize the data and command register for output.

Returns

Note

Maybe should provide software reset

4.2.2.11 uint8_t lcd_ReadData (void)

Read the 8 bit data from position set earlier by lcdCGRAM or lcdDDRAM.

Returns

uint8_t Data

Note

Address counter will also be affected (like a write)

Bug Can't work, _Send always return 0

4.2.2.12 void lcd_ReturnHome (void)

Return cursor at 00H, return display original status if shifted.

Note

Does not erase DDRAM
Unshift if shifted

4.2.2.13 uint8_t lcd_Shift (uint8_t *wich*, uint8_t *direction*)

Select which display or cursor move and to the right or left.

Parameters

| | | |
|----|------------------|--------------------|
| in | <i>wich</i> | Cursor or display |
| in | <i>direction</i> | Direction to shift |

Returns

Error code

Return values

| | |
|---|-----------------------------|
| 0 | successful |
| 1 | wrong cursor/display option |
| 2 | wrong direction |

4.2.2.14 void lcd_WriteData (uint8_t *character*)

Display *character* on the lcd.

Parameters

| | | |
|----|------------------|----------------------|
| in | <i>character</i> | Character to display |
|----|------------------|----------------------|

Note

Works for the first 127 ASCII codes

4.2.2.15 void lcd_WriteString (char * *string*)

Write a string to the LCD.

Parameters

| | | |
|----|----------------|---------------------|
| in | <i>*string</i> | Pointer to a string |
|----|----------------|---------------------|

Note

Do not check the length

Index

lcd.c, [3](#)

- [lcd_BusyFlag](#), [4](#)
- [lcd_Byte](#), [4](#)
- [lcd_CGRAM](#), [4](#)
- [lcd_ClearLine](#), [4](#)
- [lcd_DDRAM](#), [5](#)
- [lcd_DisplayOnOff](#), [5](#)
- [lcd_DoubleByte](#), [5](#)
- [lcd_EnterMode](#), [5](#)
- [lcd_FunctionSet](#), [5](#)
- [lcd_InitPort](#), [5](#)
- [lcd_ReadData](#), [6](#)
- [lcd_WriteData](#), [6](#)
- [lcd_WriteString](#), [6](#)

lcd.h, [6](#)

- [lcd_BusyFlag](#), [8](#)
- [lcd_Byte](#), [8](#)
- [lcd_CGRAM](#), [9](#)
- [lcd_ClearLine](#), [9](#)
- [lcd_DDRAM](#), [9](#)
- [lcd_DisplayOnOff](#), [9](#)
- [lcd_DoubleByte](#), [10](#)
- [lcd_EnterMode](#), [10](#)
- [lcd_FunctionSet](#), [10](#)
- [lcd_InitPort](#), [10](#)
- [lcd_ReadData](#), [11](#)
- [lcd_ReturnHome](#), [11](#)
- [lcd_Shift](#), [11](#)
- [lcd_WriteData](#), [12](#)
- [lcd_WriteString](#), [12](#)

[lcd_BusyFlag](#)

- [lcd.c](#), [4](#)
- [lcd.h](#), [8](#)

[lcd_Byte](#)

- [lcd.c](#), [4](#)
- [lcd.h](#), [8](#)

[lcd_CGRAM](#)

- [lcd.c](#), [4](#)
- [lcd.h](#), [9](#)

[lcd_ClearLine](#)

- [lcd.c](#), [4](#)
- [lcd.h](#), [9](#)

[lcd_DDRAM](#)

- [lcd.c](#), [5](#)
- [lcd.h](#), [9](#)

[lcd_DisplayOnOff](#)

- [lcd.c](#), [5](#)
- [lcd.h](#), [9](#)

[lcd_DoubleByte](#)

- [lcd.c](#), [5](#)
- [lcd.h](#), [10](#)

[lcd_EnterMode](#)

- [lcd.c](#), [5](#)
- [lcd.h](#), [10](#)

[lcd_FunctionSet](#)

- [lcd.c](#), [5](#)
- [lcd.h](#), [10](#)

[lcd_InitPort](#)

- [lcd.c](#), [5](#)
- [lcd.h](#), [10](#)

[lcd_ReadData](#)

- [lcd.c](#), [6](#)
- [lcd.h](#), [11](#)

[lcd_ReturnHome](#)

- [lcd.h](#), [11](#)

[lcd_Shift](#)

- [lcd.h](#), [11](#)

[lcd_WriteData](#)

- [lcd.c](#), [6](#)
- [lcd.h](#), [12](#)

[lcd_WriteString](#)

- [lcd.c](#), [6](#)
- [lcd.h](#), [12](#)