



M68HC12 *Microcontrollers*

*HD44780 Driven
LCD Display
Functions*

User Manual

*Designer Reference
Manual*

*DRMxxx/D
Rev. 00, 11/2003*

MOTOROLA.COM/SEMICONDUCTORS

HD44780 Driven LCD Display Functions

User Manual — Rev 00

by: Ladislav Makovic
Motorola Ltd.
Roznov pod Radhostem

Revision history

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

<http://www.motorola.com/semiconductors>

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision history

Date	Revision Level	Description	Page Number(s)
November 2003	00	Initial release	N/A

Section 1. Display Setup Functions11
Section 2. Position Reading Functions15
Section 3. Position Setup Functions17
Section 4. Display Clearing Functions21
Section 5. Display Reading Functions23
Section 6. Display Writting Functions27



Section 1. Display Setup Functions

1.1	Contents	11
1.2	Definitions	12
1.3	void LcdReset(void)	12
1.4	void LcdSetup(tUC setup)	12
1.5	void LcdSetCursorDirection(tUC direction)	13
1.6	void LcdSetDisplayShiftEnable(tUC onoff)	13
1.7	void LcdSetDisplayOnOff(tUC onoff)	13
1.8	void LcdSetCursorOnOff(tUC onoff)	13
1.9	void LcdSetBlinkOnOff(tUC onoff)	14
1.10	void LcdSetDisplayFont(tUC font)	14
1.11	void LcdShiftDisplay(tUC direction)	14
1.12	LcdSetAddressCGRAM(tUC address)	14

Section 2. Position Reading Functions

2.1	Contents	15
2.2	Definitions	16
2.3	tUC LcdGetColPos(void)	16
2.4	tUC LcdGetLinePos(void)	16
2.5	tUC LcdGetDDRAMPpos(void)	16

Section 3. Position Setup Functions

3.1	Contents	17
3.2	Definitions	18
3.3	void LcdReturnHome(void)	18
3.4	void LcdSetLine(tUC line)	18
3.5	LcdSetPos(tUC line, tUC col)	18

3.6	void LcdSetAddressDDRAM(tUC address)	19
3.7	void LcdMoveCursor(tUC direction)	19
3.8	void LcdRollUp(void)	19
3.9	void LcdRollDown(void)	19

Section 4. Display Clearing Functions

4.1	Contents	21
4.2	Definitions	22
4.3	void LcdClear(void)	22
4.4	LcdClearLine(tUC line)	22

Section 5. Display Reading Functions

5.1	Contents	23
5.2	Definitions	24
5.3	tUC LcdGetChar(void)	24
5.4	tUC LcdGetCharLC(tUC line, tUC col)	24
5.5	void LcdGetLineL(tUC line, tSC str)	24
5.6	void LcdGetLineLC(tUC line,tUC col, tSC str)	25
5.7	void LcdGetLineLCn(tUC line,tUC col,tUC n, tSC str)	25
5.8	void LcdGetWordLC(tUC line,tUC col, tSC str)	25

Section 6. Display Writting Functions

6.1	Contents	27
6.2	Definitions	28
6.3	void LcdWriteChar(tUC chr)	28
6.4	void LcdWriteCharLC(tUC line,tUC col,tUC chr)	28
6.5	void LcdOutText(tSC str)	28

6.6 void LcdOutTextL(tUC line, tSC str)29
6.7 void LcdOutTextLC(tUC line,tUC col, tSC str).....29
6.8 void LcdOutTextLCn(tUC line,tUC col,tUC n,tSC str)29



Section 1. Display Setup Functions

1.1 Contents

1.2	Definitions	12
1.3	void LcdReset(void)	12
1.4	void LcdSetup(tUC setup).	12
1.5	void LcdSetCursorDirection(tUC direction)	13
1.6	void LcdSetDisplayShiftEnable(tUC onoff)	13
1.7	void LcdSetDisplayOnOff(tUC onoff)	13
1.8	void LcdSetCursorOnOff(tUC onoff)	13
1.9	void LcdSetBlinkOnOff(tUC onoff)	14
1.10	void LcdSetDisplayFont(tUC font)	14
1.11	void LcdShiftDisplay(tUC direction)	14
1.12	LcdSetAddressCGRAM(tUC address)	14

Display Setup Functions

1.2 Definitions

```
#define tUL      unsigned long
#define tUI      unsigned int
#define tUC      unsigned char
#define tSC      signed char
```

1.3 void LcdReset(void)

Description : Standard display reset.
Function has to be called before first display using to be initialized for communication. The function sets display behavior to LCD_CURSORRIGHT, LCD_SHIFTOFF, LCD_DISPLAYON, LCD_CURSOROFF, LCD_BLONKOFF, LCD_SMALLFONT, and it is ready to be used.

Global Data : none
Static Global Data: none
Returns : none
Arguments : none
Special Issues : none

1.4 void LcdSetup(tUC setup)

Description : Function sets whole display to required behavior and clears display.
For display setup it is necessary to use following definitions:

```
LCD_CURSORLEFT or LCD_CURSORRIGHT => Sets cursor move direction
LCD_SHIFTON or LCD_SHIFTOFF => Enables accompanies display shift
LCD_DISPLAYOFF or LCD_DISPLAYON => Switches on/off display
LCD_CURSORON or LCD_CURSOROFF => Switches on/off cursor
LCD_BLINKON or LCD_BLINKOFF => Switches cursor blink on/off
LCD_LARGEFONT or LCD_SMALLFONT => Sets large 5x10 / small 5x8 font
```

Global Data : none
Static Global Data: none
Returns : none
Arguments : tUC setup =
((LCD_CURSORLEFT or LCD_CURSORRIGHT) |
(LCD_SHIFTON or LCD_SHIFTOFF) |
(LCD_DISPLAYOFF or LCD_DISPLAYON) |
(LCD_CURSORON or LCD_CURSOROFF) |
(LCD_BLINKON or LCD_BLINKOFF) |
(LCD_LARGEFONT or LCD_SMALLFONT))

Special Issues : none

1.5 void LcdSetCursorDirection(tUC direction)

Description : set cursor moving direction to the right or left
Global Data : none
Static Global Data: Lcd.movingSetup
Returns : none
Arguments : tUC direction = {LCD_CURSORRIGHT, LCD_CURSORLEFT}
Special Issues : none

1.6 void LcdSetDisplayShiftEnable(tUC onoff)

Description : set display shift on/off
Global Data : none
Static Global Data: Lcd.movingSetup
Returns : none
Arguments : tUC onoff = {LCD_SHIFTON, LCD_SHIFTOFF}
Special Issues : none

1.7 void LcdSetDisplayOnOff(tUC onoff)

Description : set display on/off
Global Data : none
Static Global Data: Lcd.onOffControl
Returns : none
Arguments : tUC onoff = {LCD_DISPLAYON, LCD_DISPLAYOFF}
Special Issues : none

1.8 void LcdSetCursorOnOff(tUC onoff)

Description : set cursor on/off
Global Data : none
Static Global Data: Lcd.onOffControl
Returns : none
Arguments : tUC onoff = {LCD_CURSORON, LCD_CURSOROFF}
Special Issues : none

1.9 void LcdSetBlinkOnOff(tUC onoff)

Description : set cursor blink on/off
Global Data : none
Static Global Data: Lcd.onOffControl
Returns : none
Arguments : tUC onoff = {LCD_BLINKON, LCD_BLINKOFF}
Special Issues : none

1.10 void LcdSetDisplayFont(tUC font)

Description : set display font
Global Data : none
Static Global Data: Lcd.interface
Returns : none
Arguments : tUC font = {LCD_LARGEFONT, LCD_SMALLFONT}
Special Issues : none

1.11 void LcdShiftDisplay(tUC direction)

Description : shift display
Global Data : none
Static Global Data: none
Returns : none
Arguments : tUC direction = {LCD_RIGHT, LCD_LEFT}
Special Issues : none

1.12 LcdSetAddressCGRAM(tUC address)

Description : set CG RAM address
Global Data : none
Static Global Data: none
Returns : none
Arguments : tUC address
Special Issues : none

Section 2. Position Reading Functions

2.1 Contents

2.2	Definitions	16
2.3	tUC LcdGetColPos(void)	16
2.4	tUC LcdGetLinePos(void)	16
2.5	tUC LcdGetDDRAMPos(void)	16

2.2 Definitions

```
#define      tUL          unsigned long
#define      tUI          unsigned int
#define      tUC          unsigned char
#define      tSC          signed char
```

2.3 tUC LcdGetColPos(void)

Description : read display column position
Global Data : none
Static Global Data: Lcd.cursor.address, Lcd.cursor.line, Lcd.cursor.col
Returns : display column
Arguments : none
Special Issues : If definition LCD_NO_READ_FUNCTION exists the function is not accessible.

2.4 tUC LcdGetLinePos(void)

Description : read display line position
Global Data : none
Static Global Data: Lcd.cursor.address, Lcd.cursor.line, Lcd.cursor.col
Returns : display line
Arguments : none
Special Issues : If definition LCD_NO_READ_FUNCTION exists the function is not accessible.

2.5 tUC LcdGetDDRAMPos(void)

Description : read display DDRAM address pointer
Global Data : none
Static Global Data: Lcd.cursor.address, Lcd.cursor.line, Lcd.cursor.col
Returns : DDRAM address
Arguments : none
Special Issues : If definition LCD_NO_READ_FUNCTION exists the function is not accessible.

Section 3. Position Setup Functions

3.1 Contents

3.2	Definitions	18
3.3	void LcdReturnHome(void)	18
3.4	void LcdSetLine(tUC line)	18
3.5	LcdSetPos(tUC line, tUC col)	18
3.6	void LcdSetAddressDDRAM(tUC address)	19
3.7	void LcdMoveCursor(tUC direction)	19
3.8	void LcdRollUp(void)	19
3.9	void LcdRollDown(void)	19

3.2 Definitions

```
#define tUL      unsigned long
#define tUI      unsigned int
#define tUC      unsigned char
#define tSC      signed char
```

3.3 void LcdReturnHome(void)

Description : return cursor to the home position
and return shifted display to original position.
DDRAM contents remains unchanged

Global Data : none

Static Global Data: none

Returns : none

Arguments : none

Special Issues : none

3.4 void LcdSetLine(tUC line)

Description : set DDRAM address at the first char of line x

Global Data : none

Static Global Data: none

Returns : none

Arguments : tUC line = <0,LCD_LINES)

Special Issues : none

3.5 LcdSetPos(tUC line, tUC col)

Description : set DDRAM address at position Line,Column

Global Data : none

Static Global Data: none

Returns : none

Arguments : tUC line = <0,LCD_LINES), tUC column = <0,LCD_COLUMNS)

Special Issues : none

3.6 void LcdSetAddressDDRAM(tUC address)

Description : set DDRAM address
Global Data : none
Static Global Data: none
Returns : none
Arguments : tUC address
Special Issues : none

3.7 void LcdMoveCursor(tUC direction)

Description : move cursor and set position
Global Data : none
Static Global Data: Lcd.cursor.address, Lcd.cursor.line, Lcd.cursor.col
Returns : none
Arguments : tUC direction = {LCD_RIGHT, LCD_LEFT, LCD_DOWN, LCD_UP}
Special Issues : none

3.8 void LcdRollUp(void)

Description : Roll display up and set first position on the last line.
Global Data : none
Static Global Data: none
Returns : none
Arguments : none
Special Issues : If definition LCD_NO_READ_FUNCTION exists the function is not accesible.

3.9 void LcdRollDown(void)

Description : Roll display down and set first posit. on the first line.
Global Data : none
Static Global Data: none
Returns : none
Arguments : none
Special Issues : none
Special Issues : If definition LCD_NO_READ_FUNCTION exists the function is not accesible.

Section 4. Display Clearing Functions

4.1 Contents

4.2	Definitions	22
4.3	void LcdClear(void)	22
4.4	LcdClearLine(tUC line)	22

4.2 Definitions

```
#define      tUL          unsigned long
#define      tUI          unsigned int
#define      tUC          unsigned char
#define      tSC          signed char
```

4.3 void LcdClear(void)

Description : clear all display and return the cursor to the home
Global Data : none
Static Global Data: Lcd.cursor.address, Lcd.cursor.line, Lcd.cursor.col
Returns : none
Arguments : none
Special Issues : none

4.4 LcdClearLine(tUC line)

Description : clear line on display
Global Data : none
Static Global Data: none
Returns : none
Arguments : tUC line = <0,LCD_LINES)
Special Issues : none

Section 5. Display Reading Functions

5.1 Contents

5.2	Definitions	24
5.3	tUC LcdGetChar(void)	24
5.4	tUC LcdGetCharLC(tUC line, tUC col)	24
5.5	void LcdGetLineL(tUC line, tSC str)	24
5.6	void LcdGetLineLC(tUC line,tUC col, tSC str)	25
5.7	void LcdGetLineLCn(tUC line,tUC col,tUC n, tSC str)	25
5.8	void LcdGetWordLC(tUC line,tUC col, tSC str)	25

5.2 Definitions

```
#define      tUL          unsigned long
#define      tUI          unsigned int
#define      tUC          unsigned char
#define      tSC          signed char
```

5.3 tUC LcdGetChar(void)

Description : Read character from display from actual DDRAM address
Global Data : none
Static Global Data: none
Returns : tUC read character
Arguments : none
Special Issues : If definition LCD_NO_READ_FUNCTION exists the function is not accesible.

5.4 tUC LcdGetCharLC(tUC line, tUC col)

Description : Read character from position line,col.
Current cursor (DDRAM address) remains unchanged
Global Data : none
Static Global Data: none
Returns : tUC read character
Arguments : tUC line = <0,LCD_LINES), tUC column = <0,LCD_COLUMNS)
Special Issues : If definition LCD_NO_READ_FUNCTION exists the function is not accesible.

5.5 void LcdGetLineL(tUC line, tSC str)

Description : Read line x and put it into string. Current cursor position remains unchanged.
Global Data : none
Static Global Data: none
Returns : none
Arguments : tUC line = <0,LCD_LINES), tSC string
Special Issues : String should have length 1 byte (for termination string character '\0') longer than LCD_COLUMNS.
If definition LCD_NO_READ_FUNCTION exists the function is not accesible.

5.6 void LcdGetLineLC(tUC line,tUC col, tSC str)

Description : Read text from position Line,Column till the end of the line and put it into string. Cursor position remains unchanged.

Global Data : none

Static Global Data: none

Returns : none

Arguments : tUC line = <0,LCD_LINES),
tUC column = <0,LCD_COLUMNS),
tSC string

Special Issues : String should have length 1 byte (for termination string character '\0') longer than LCD_COLUMNS.
If definition LCD_NO_READ_FUNCTION exists the function is not accesible.

5.7 void LcdGetLineLCn(tUC line,tUC col,tUC n, tSC str)

Description : Read text from position Line, Column till the end of the line or n character were read and put it into string. Cursor position remains unchanged.

Global Data : none

Static Global Data: none

Returns : none

Arguments : tUC line = <0,LCD_LINES),
tUC column = <0,LCD_COLUMNS),
tUC number of characters,
tSC string

Special Issues : String should have length 1 byte (for termination string character '\0') longer than LCD_COLUMNS.
If definition LCD_NO_READ_FUNCTION exists the function is not accesible.

5.8 void LcdGetWordLC(tUC line,tUC col, tSC str)

Description : Read text but one word only from position line,column Cursor position remains unchanged.

Global Data : none

Static Global Data: none

Returns : none

Arguments : tUC line = <0,LCD_LINES),
tUC column = <0,LCD_COLUMNS),
tSC string

Special Issues : String should have length 1 byte (for termination string character '\0') longer than LCD_COLUMNS.
If definition LCD_NO_READ_FUNCTION exists the function is not accesible.

Section 6. Display Writting Functions

6.1 Contents

6.2	Definitions	28
6.3	void LcdWriteChar(tUC chr)	28
6.4	void LcdWriteCharLC(tUC line,tUC col,tUC chr)	28
6.5	void LcdOutText(tSC str)	28
6.6	void LcdOutTextL(tUC line, tSC str)	29
6.7	void LcdOutTextLC(tUC line,tUC col, tSC str)	29
6.8	void LcdOutTextLCn(tUC line,tUC col,tUC n,tSC str).	29

6.2 Definitions

```
#define      tUL          unsigned long
#define      tUI          unsigned int
#define      tUC          unsigned char
#define      tSC          signed char
```

6.3 void LcdWriteChar(tUC chr)

Description : Write character to display on current position, cursor moves by autoincrement setup. It doesn't check end of line.

Global Data : none

Static Global Data: none

Returns : none

Arguments : tUC character

Special Issues : none

6.4 void LcdWriteCharLC(tUC line,tUC col,tUC chr)

Description : Write character to display on position Line,Column. Cursor moves by autoincrement setup but if it is on the last column then cursor stays on this position.

Global Data : none

Static Global Data: none

Returns : none

Arguments : tUC line = <0,LCD_LINES),
tUC column = <0,LCD_COLUMNS),
tUC character

Special Issues : none

6.5 void LcdOutText(tSC str)

Description : Write string to current position. String must be finished by '\0'. Maximal LCD_COLUMNS characters is written. Autoincrement setup has influence. Cursor will be placed behind the last written character. It doesn't check end of line.

Global Data : none

Static Global Data: none

Returns : none

Arguments : tSC string

Special Issues : none

6.6 void LcdOutTextL(tUC line, tSC str)

Description : Write string to line x. String must be finished by '\0'.
Maximal LCD_COLUMNS characters is written.
Autoincrement setup has no influence.
Cursor will be placed behind the last written character
or at the end of the line if string overlaps line

Global Data : none
Static Global Data: none
Returns : none
Arguments : tUC line = <0,LCD_LINES), tSC string
Special Issues : none
Special Issues : none

6.7 void LcdOutTextLC(tUC line,tUC col, tSC str)

Description : Write string to position Line,column.
String must be finished by '\0'. String is written to
display till the end of the line or '\0' is reached.
Autoincrement setup has no influence.
Cursor will be placed behind the last written character
or at the end of the line if string overlaps line

Global Data : none
Static Global Data: none
Returns : none
Arguments : tUC line = <0,LCD_LINES),
tUC column = <0,LCD_COLUMNS),
tSC string
Special Issues : none

6.8 void LcdOutTextLCn(tUC line,tUC col,tUC n,tSC str)

Description : Write string to position Line,Column.
String must be finished by '\0'.
String is written to display till given number of
characters or the end of the line or '\0' is reached.
Autoincrement setup has no influence.
Cursor will be placed behind the last written character
or at the end of the line if string overlaps line

Global Data : none
Static Global Data: none
Returns : none
Arguments : tUC line = <0,LCD_LINES),
tUC column = <0,LCD_COLUMNS),
tUC number of characters,
tSC string
Special Issues : none

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

<http://motorola.com/semiconductors>

Information in this document is provided solely to enable system and software implementers to use Motorola products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.



Motorola and the Stylized M Logo are registered in the U.S. Patent and Trademark Office. digital dna is a trademark of Motorola, Inc. All other product or service names are the property of their respective owners. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

© Motorola, Inc. 2003

DRMxxx/D