AVR-GCC 4 bit and 8 bit LCD library

Wed, 08/08/2007 - 10:25 — admin

Topics:

Standard alphanumeric LCD display controlled by *74HC164* LCD controller can accept 8 bit data bytes or 4 bit nibbles. Earlier my 4 bit and 8 bit LCD libraries were split in separate files as they were used in different projects. Now they are merged in to one library where simple logic is implemented to select 4 bit or 8 bit library just by modifying only three lines of code.

In the library header file there is line added:

```
//Uncomment this if LCD 4 bit interface isused
//*****************************************
#define LCD_4bit
//*****************************************
```

what allows to select different LCD modes by commenting and uncommenting this line. Also don't forget to select proper ports and pins where LCD is connected:

```
#define LCD_RS 0 //define MCU pin connected to LCD RS
#define LCD_RW 1 //define MCU pin connected to LCD R/W
#define LCD_E 2 //define MCU pin connected to LCD E
#define LCD_D0 0 //define MCU pin connected to LCD D0
#define LCD_D1 1 //define MCU pin connected to LCD D1
#define LCD_D2 2 //define MCU pin connected to LCD D1
#define LCD_D3 3 //define MCU pin connected to LCD D2
#define LCD_D4 4 //define MCU pin connected to LCD D3
#define LCD_D5 5 //define MCU pin connected to LCD D4
#define LCD_D6 6 //define MCU pin connected to LCD D5
#define LCD_D7 7 //define MCU pin connected to LCD D6
#define LDP PORTD //define MCU port connected to LCD data pins
#define LCP PORTD //define MCU port connected to LCD control pins
#define LDDR DDRD //define MCU direction register for port connected to LCD data pins
#define LCDR DDRD //define MCU direction register for port connected to LCD control pins
```

In newer library there is also couple new functionalities added:

- Predefining 8 custom LCD characters during LCD initialisation;
- LCDprogressBar() function have been adapted from AVRLib.

Complete function set:

```
void LCDsendChar(uint8_t); //forms data ready to send to 74HC164
void LCDsendCommand(uint8_t); //forms data ready to send to 74HC164
void LCDinit(void); //Initializes LCD
void LCDclr(void); //Clears LCD
void LCDhome(void); //LCD cursor home
```

void LCDstring(uint8_t, uint8_t); //Outputs string to LCD*

void LCDGotoXY(uint8_t, uint8_t); //Cursor to X Y position

void CopyStringtoLCD(const uint8_t, uint8_t, uint8_t);//copies flash string to LCD at x,y*

*void LCDdefinechar(const uint8_t *,uint8_t);//write char to LCD CGRAM*

void LCDshiftRight(uint8_t); //shift by n characters Right

void LCDshiftLeft(uint8_t); //shift by n characters Left

void LCDcursorOn(void); //Underline cursor ON

void LCDcursorOnBlink(void); //Underline blinking cursor ON

void LCDcursorOFF(void); //Cursor OFF

void LCDblank(void); //LCD blank but not cleared

void LCDvisible(void); //LCD visible

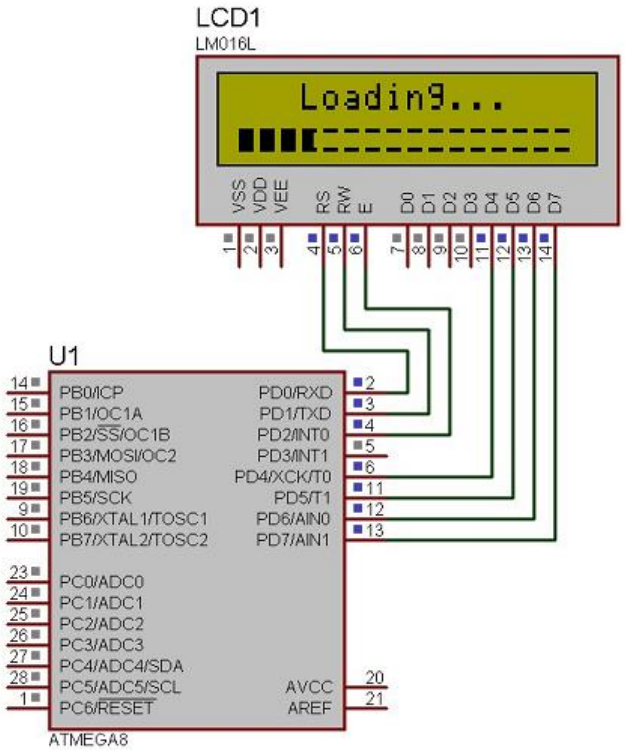
void LCDcursorLeft(uint8_t); //Shift cursor left by n

void LCDcursorRight(uint8_t); //shif cursor right by n

// displays a horizontal progress bar at the current cursor location

void LCDprogressBar(uint8_t progress, uint8_t maxprogress, uint8_t length);

I have added simple demo program that should help to start working with LCD:



File:

4 and 8 bit LCD library for AVRGCC

Driving LCD demo project

Comments

Changing PORT

DiegoHxC - Fri, 03/04/2011 - 18:19.

How can I change the port to conect the lcd screen?I mean, I m using an ATmega16 MCU and I want to use The PORTC instead of PORTD.I tried changing the header file lcd_lib.hwriting:#define LDP PORTC #define LCP PORTC #define LDDR DDRC #define LCDR DDRC But guess what nothing happens still working on PORTDor your library only works in PORTD??

I'm passenger.

'Yellow Monkey' Mute (not verified) - Fri, 06/08/2012 - 17:47.

if you want to use PORTC, please set the pin number to use with 'LCP"LDP' in lcd_lib.h. I set PC0 as RS, PC1 as E,output data to PC2 ~ PC5 in PORT C of the ATmega168.

----------<pick of lcd_lib.h>
#define LCD_RS 0
#define LCD_E 1

#define LCD_D4 2
#define LCD_D5 3
#define LCD_D6 4
#define LCD_D7 5
#define LDP PORTC
#define LCP PORTC
#define LDDR DDRC
#define LCDR DDRC
----------<E>

and lcd_lib.c is fixed data bits.
I have been changed as follows:
----------<pick of lcd_lib.c>
void LCDsendChar(uint8_t ch) //Sends Char to LCD
{

#ifdef LCD_4bit
LDP=(ch&0b11110000) >> (4 - LCD_D4);
LCP|=1<<LCD_RS;
LCP|=1<<LCD_E;
_delay_ms(1);
LCP&=~(1<<LCD_E);
LCP&=~(1<<LCD_RS);
_delay_ms(1);
LDP=((ch&0b00001111)<<4)>>(4 - LCD_D4);
LCP|=1<<LCD_RS;
LCP|=1<<LCD_E;
_delay_ms(1);
LCP&=~(1<<LCD_E);
LCP&=~(1<<LCD_RS);
_delay_ms(1);
----------<E>

Blinking LEDs
Uthman Sa'eed Ahmad (not verified) - Thu, 12/08/2011 - 14:56.

I wish that more topics will be included on Blinking LEDs

Add new comment

Your name

Subject

Comment *

- No HTML tags allowed.
- Web page addresses and e-mail addresses turn into links automatically.
- Lines and paragraphs break automatically.

More information about text formats

CAPTCHA

This question is for testing whether you are a human visitor and to prevent automated spam submissions.

Save Preview

Searching for **avr projects**?

1. **Projects** www.Pronto.com
   Find **Projects** at Great Prices.

2. Searching for **Avr Projects**? www.Answered-Questions.com
   Top answers for **Avr Projects**

3. **Avr Projects** near you www.Answered-Questions.com
   Get local answers for **Avr Projects**

4. Looking for **Avr Projects**? www.Answered-Questions.com
   Discover 100+ answers for **Avr Projects**