



AVR-GCC LCD library – mixed pin support



Fri, 12/16/2011 - 18:06 — admin

Topics:

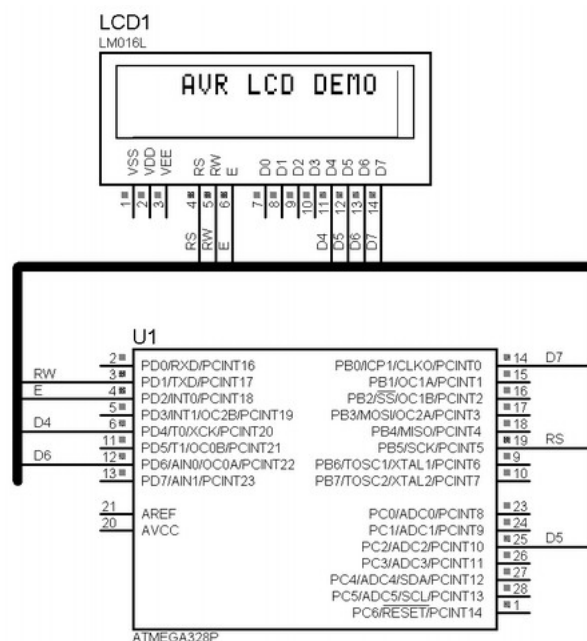
Some time ago we have posted alphanumeric AVR-GCC LCD library. It works fine in 8-bit and 4-bit modes. But it has some limitations that some people may find annoying. One of them is requirement that LCD pins has to be byte aligned for instance in 8 bit mode LCD_D0 ... LCD_D7 pins has to be connected to AVR single AVR port. Similar situation is with 4-bit mode where LCD data pins has to be connected to single port 4, 5, 6 and 7 pins. For both modes control pins RS, RW and E has to be connected to single port.



And this is how most LCD libraries work when you try to look for one in the internet. In reality things may be a bit different each microcontroller pin has at least several alternative functions available like ADC, INT, I2C, USART and if project requires using one or another and you still need LCD standard libraries won't work as most likely you won't be able to get all particular port pins connected to LCD. You gotta use what's left. This is why I decided to find a little time and modify LCD library to support these cases. Didn't want to write everything from scratch or change its functionality – just wanted it to work with existing projects but have more freedom with new ones. So basically I left standard 8-bit and 4-bit same. The main change is adding two more modes: 8-bit mix and 4-bit mix. These modes allow connected LCD to any free pins of microcontroller.

Configuring LCD for 4-bit mixed pin mode

Lets look what you need to start using these features. We can do this by selecting a simple example. This time 4-bit mixed mode:



As you can see LCD pins are connected to Atmega328P in following way:

RS -> PB5

RW -> PD1

E -> PD2
 D4 -> PD4
 D5 -> PC2
 D6 -> PD6
 D7 -> PB0

So we get unaligned situation. First of all we need to edit pin assignments in lcd_lib.h. First of all uncomment one of following defines that indicate the mode chosen

```
//LCD 4 bit interface is used (single port pins)

//#define LCD_4BIT

//LCD 8 bit interface is used (single port pins)

//#define LCD_8BIT

//LCD 4 bit interface is used (mixed port pins)

#define LCD_4BIT_M

//LCD 8 bit interface is used (mixed port pins)

//#define LCD_8BIT_M
```

This time we use LCD_4BIT_M

Then we need to associate LCD pins with port pin numbers. If LCD_RS is connected to PB5 pin then we write 5:

```
#define LCD_RS 5 //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 D2

#define LCD_D3 3 //define MCU pin connected to LCD D3

#define LCD_D4 4 //define MCU pin connected to LCD D4

#define LCD_D5 2 //define MCU pin connected to LCD D5

#define LCD_D6 6 //define MCU pin connected to LCD D6

#define LCD_D7 0 //define MCU pin connected to LCD D7
```

And now we have to define port and data direction register for each pin. As pins may be connected to different ports – we need to work with individual pins. We edit this part:

```
#ifndef LCD_4BIT_M || LCD_8BIT_M //8- bit or 4 - bit mode

#define LDPRS PORTB //RS pin assignment

#define LDDRS DDRB

#define LDPRW PORTD //RW pin assignment

#define LDDRW DDRD

#define LDPE PORTD //E pin assignment

#define LDDE DDRD

#define LDPD0 PORTD //D0 pin assignment

#define LDD0 DDRD

#define LDPD1 PORTD //D1 pin assignment

#define LDD1 DDRD

#define LDPD2 PORTD //D2 pin assignment

#define LDD2 DDRD

#define LDPD3 PORTD //D3 pin assignment

#define LDD3 DDRD

#define LDPD4 PORTD //D4 pin assignment

#define LDD4 DDRD

#define LDPD5 PORTC //D5 pin assignment

#define LDD5 DDRC

#define LDPD6 PORTD //D6 pin assignment

#define LDD6 DDRD

#define LDPD7 PORTB //D7 pin assignment

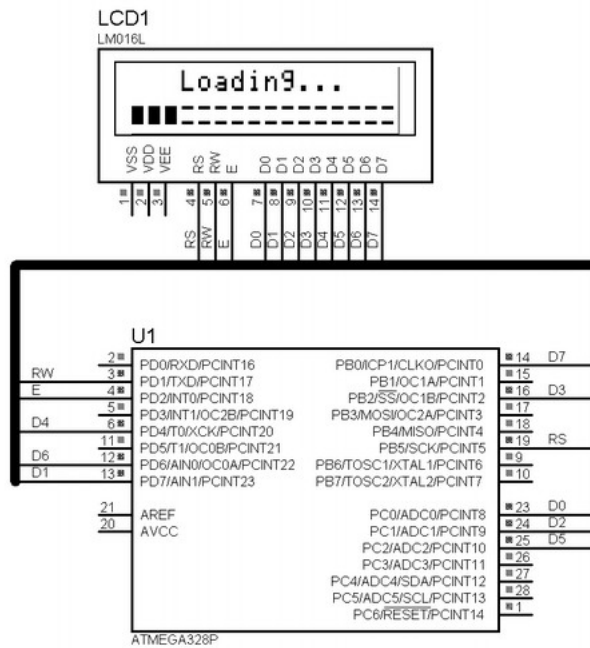
#define LDD7 DDRB
```

```
#endif
```

This is practically it. We can start using LCD as we did in old library version.

Configuring LCD for 8-bit mixed pin mode

To make sure things are working correctly lets set up project for 8-bit mixed mode. To do so we connect LCD as follows:



Again we uncomment following mode:

```
#define LCD_8BIT_M
```

Then assign pin numbers:

```
#define LCD_RS 5 //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 7 //define MCU pin connected to LCD D1
#define LCD_D2 1 //define MCU pin connected to LCD D2
#define LCD_D3 2 //define MCU pin connected to LCD D3
#define LCD_D4 4 //define MCU pin connected to LCD D4
#define LCD_D5 2 //define MCU pin connected to LCD D5
#define LCD_D6 6 //define MCU pin connected to LCD D6
#define LCD_D7 0 //define MCU pin connected to LCD D7
```

and finally we chose ports for each pin assigned pin:

```
#if defined (LCD_4BIT_M) || defined (LCD_8BIT_M)
#define LDPRS PORTB //RS pin assignment
#define LDDRS DDRB

#define LDPRW PORTD //RW pin assignment
#define LDDRW DDRD

#define LDPE PORTD //E pin assignment
#define LDDE DDRD

#define LDPD0 PORTC //D0 pin assignment
#define LDD0 DDRD

#define LDPD1 PORTD //D1 pin assignment
#define LDD1 DDRD

#define LDPD2 PORTC //D2 pin assignment
#define LDD2 DDRD

#define LDPD3 PORTB //D3 pin assignment
#define LDD3 DDRB
```

```

#define LDPD4 PORTD //D4 pin assignment

#define LDD4 DDRD

#define LDPD5 PORTC //D5 pin assignment

#define LDD5 DDRC

#define LDPD6 PORTD //D6 pin assignment

#define LDD6 DDRD

#define LDPD7 PORTB //D7 pin assignment

#define LDD7 DDRB

#endif

```

Configuring LCD for normal 4-bit

In case you need to use LCD in byte aligned way like we used to do in old version of LCD library. To do so we need to uncomment mode:

```
#define LCD_4BIT
```

We still need to define pin numbers for for control and data pins.

```

#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 D2

#define LCD_D3 3 //define MCU pin connected to LCD D3

#define LCD_D4 4 //define MCU pin connected to LCD D4

#define LCD_D5 5 //define MCU pin connected to LCD D5

#define LCD_D6 6 //define MCU pin connected to LCD D6

#define LCD_D7 7 //define MCU pin connected to LCD D7

```

Then we only need to define data and control ports as follows:

```

#if defined (LCD_4BIT) || defined (LCD_8BIT) //if aligned mode

#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

#endif

```

same situation with normal 8 bit mode.

It is obviously that in pin aligned mode LCD update is faster as either 4-bit or 8-bit mode accepts data faster – byte or nibble operation. In mixed mode each individual pin needs to be set separately. This increases number of instructions used to transfer byte. For instance in order to send a byte in LCD_8BIT_M mode I used a helper function:

```

static void LCDMix_8Bit(uint8_t data)

{

if((data)&(0b10000000)) LDPD7 |=1<<LCD_D7;

else LDPD7 &=~(1<<LCD_D7);

if((data)&(0b01000000)) LDPD6 |=1<<LCD_D6;

else LDPD6 &=~(1<<LCD_D6);

if((data)&(0b00100000)) LDPD5 |=1<<LCD_D5;

else LDPD5&=~(1<<LCD_D5);

if((data)&(0b00010000)) LDPD4 |=1<<LCD_D4;

else LDPD4 &=~(1<<LCD_D4);

if((data)&(0b00001000)) LDPD3 |=1<<LCD_D3;

else LDPD3 &=~(1<<LCD_D3);

if((data)&(0b00000100)) LDPD2 |=1<<LCD_D2;

else LDPD2 &=~(1<<LCD_D2);

if((data)&(0b00000010)) LDPD1 |=1<<LCD_D1;

else LDPD1&=~(1<<LCD_D1);





if((data)&(0b00000001)) LDPD0 |=1<<LCD_D0;

```

```
else LDPD0 &=~(1<<LCD_D0);
}
}
```

Where every bit in data byte is tested and then corresponding port pin is set or reset. Generally speaking if your application simply indicates information that doesn't have to be updated frequently any mode is fine. But if you use LCD for dynamic indication like animations or other fancy way that needs fast LCD update then probably chose normal 8-bit mode or at least 4-bit. If you find errors or difficulties to use this lib, fell free to post a comment.

File:

-  AVR-GCC LCD library with mixed pins modes
-  AVR LCD 4-bit mixed pin mode example AVRStudio5
-  AVR LCD 8-bit mixed pin mode example AVRStudio5
-  AVR LCD 4-bit mixed pin mode example (Programmes Notepad and WinAVR)

Comments

Problem with compiling

Anonymous (not verified) - Fri, 04/20/2012 - 13:36.

Dear friend,
For some reason, my WinAVR doesn't want to compile this.
It's your project, unmodified and I keep getting the same error.

Any thoughts?

Thank you.

Compiling C: main.c

```
avr-gcc -c -mmcu=attiny13 -l. -gdwarf-2 -DF_CPU=1000000UL -Os -funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums -Wall -Wstrict-prototypes
-Wa,-adhlns= ./main.lst -std=gnu99 -MMD -MP -MF .dep/main.o.d main.c -o main.o
In file included from main.c:18:
```

```
lcd_lib.h:44:19: warning: extra tokens at end of #ifdef directive
lcd_lib.h:69:17: warning: extra tokens at end of #ifdef directive
```

Linking: main.elf

```
avr-gcc -mmcu=attiny13 -l. -gdwarf-2 -DF_CPU=1000000UL -Os -funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums -Wall -Wstrict-prototypes
-Wa,-adhlns=main.o -std=gnu99 -MMD -MP -MF .dep/main.elf.d main.o --output main.elf -Wl,-Map=main.map,--cref -lm
```

main.o: In function `demoanimation':

```
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:65: undefined reference to `LCDclr'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:66: undefined reference to `LCDdefinechar'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:67: undefined reference to `CopyStringtoLCD'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:70: undefined reference to `LCDGotoXY'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:71: undefined reference to `LCDsendChar'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:73: undefined reference to `LCDGotoXY'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:74: undefined reference to `LCDsendChar'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:76: undefined reference to `LCDGotoXY'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:77: undefined reference to `LCDsendChar'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:79: undefined reference to `LCDGotoXY'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:80: undefined reference to `LCDsendChar'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:82: undefined reference to `LCDGotoXY'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:83: undefined reference to `LCDsendChar'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:85: undefined reference to `LCDGotoXY'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:86: undefined reference to `LCDsendChar'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:88: undefined reference to `LCDGotoXY'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:89: undefined reference to `LCDsendChar'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:91: undefined reference to `LCDGotoXY'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:92: undefined reference to `LCDsendChar'
main.o: In function `progress':
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:50: undefined reference to `LCDclr'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:51: undefined reference to `CopyStringtoLCD'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:53: undefined reference to `LCDclr'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:54: undefined reference to `CopyStringtoLCD'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:58: undefined reference to `LCDGotoXY'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:59: undefined reference to `LCDprogressBar'
main.o: In function `main':
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:99: undefined reference to `LCDinit'
C:\Users\Administrator\Desktop\avrlcd\avrlcd/main.c:100: undefined reference to `LCDclr'
```

Could you provide what IDE

admin - Fri, 04/20/2012 - 14:03.

Could you provide what IDE (AVRStudio) are you using. Are you using WinAVR or avrtoolchain. At a glance it seems that lcd_lib.c isn't included in to project source list.

IDE

Sasha - Fri, 04/20/2012 - 14:07.

Programmers Notepad 2 :SDownloaded WinAVR and did some projects, worked fine.When I've tried to compile yours script, the above error appeared.Currently Installing Atmel AVRStudio 5 and then I'll try to compile it with it.P.S. I've noticed that you didn't Include lcd_lib.c, just the header file in your examples?:confused:

Examples were written with

admin - Fri, 04/20/2012 - 14:39.

Examples were written with AVRStudio 5. Files were included in project tree and makefile were generated automatically. With Programmes Notepad you need to edit makefile manually. In a minute I am gonna add Prorgammers notepad example you will be able to compile and try.

Please download (AVR LCD 4

admin - Fri, 04/20/2012 - 14:42.

Please download (AVR LCD 4-bit mixed pin mode example (Programmes Notepad and WinAVR) it and confirm it compiles OK within Programmers notepad.

Sasha

Sasha - Fri, 04/20/2012 - 14:52.

Getting the error while downloading:The requested page "/comment/sites/default/files/storyfiles/avrlcd4bitmix_PN.zip" could not be found. :S

It looks like it works now.

Sasha - Fri, 04/20/2012 - 14:57.

Thank you very much. Compiled Ok.Now I'll try to modify lcd_lib.h so that PINS correspond to real config.^^

Download Links seems to be OK

admin - Fri, 04/20/2012 - 15:01.

Download Links seems to be OK. Double checked. Could be if you were trying to download while I was updating post :D

;)Maybe so...Well... compiled

Sasha - Fri, 04/20/2012 - 16:43.

;)Maybe so...Well... compiled it, burned it to my ATTINY2313 and it works... well... almost ;)First 8 charachters are displayed, second 8 not.But when I try to write to line 2 (3), it displays those character.Tried with 2 different LCD displays and they both behave like that.Will have to tweak output a bit, but for my application it will be sufficient.Thank you a lot! ;)

may be you're pointing LCD

admin - Fri, 04/20/2012 - 16:57.

may be you're pointing LCD coordinates wrongly... FYI:

```
LCDGotoXY(0, 0); //first row first column
LCDGotoXY(0, 1); //second row first column
LCDGotoXY(15,0); //first row last column
LCDGotoXY(15,1); //second row las column
```

RE: may be you're pointing LCD

Sasha - Fri, 04/20/2012 - 17:00.

Looks like my 1x16 displays are actually 2x8 side by side. :S

Another question... doesn't have anything to do with this

Sasha - Fri, 04/20/2012 - 17:02.

Is there any way to do Programming on USBASP from AVR Studio 5 ?

AVRStudio doesnt support such

admin - Fri, 04/20/2012 - 17:28.

AVRStudio doesnt support such programmers - only original ones. Maybe you can try setting up some automation using external makefile. But again out of box AVRtoolchain doesnt include avrdude utility as many other. Quite some manual preparation...

display number or analog values

Anonymous (not verified) - Fri, 05/04/2012 - 11:29.

how can i display integers or analog values from this programm on lcd.....
please help me.....
thanks in advance

re: display numbers or analog values

Sasha - Fri, 10/19/2012 - 10:28.

did you tried this?`itoa(integer value, storage_string, base);` **example:**`string value;itoa(240, value, 10);CopyStringtoLCD(value,0,0);`

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.

mission

newperaf

Type the two words:



Save

Preview

Searching for **avr gcc lcd library mixed pin support**?

1. [Searching for Avr Gcc lcd library mixed pin support?](#)
Top answers for **Avr Gcc lcd library mixed pin support**

www.Answered-Questions.com

2. [Avr Gcc lcd library mixed pin support near you](#)
Get local answers for **Avr Gcc lcd library mixed pin support**

www.Answered-Questions.com

3. [Looking for Avr Gcc lcd library mixed pin support?](#)
Discover 100+ answers for **Avr Gcc lcd library mixed pin support**

www.Answered-Questions.com

4. [Avr Gcc lcd library mixed pin support](#)
Top answers for **Avr Gcc lcd library mixed pin support**

www.Answered-Questions.com

AdChoices

Chitika | Opt out?