liquidcrystal Documentation

Release 0.0.0

Arduino.cc

Sep 22, 2017

Contents

Contents:

CHAPTER 1

LiquidCrystal Library

This library allows an Arduino board to control LiquidCrystal displays (LCDs) based on the Hitachi HD44780 (or a compatible) chipset, which is found on most text-based LCDs. The library works with in either 4- or 8-bit mode (i.e. using 4 or 8 data lines in addition to the rs, enable, and, optionally, the rw control lines).

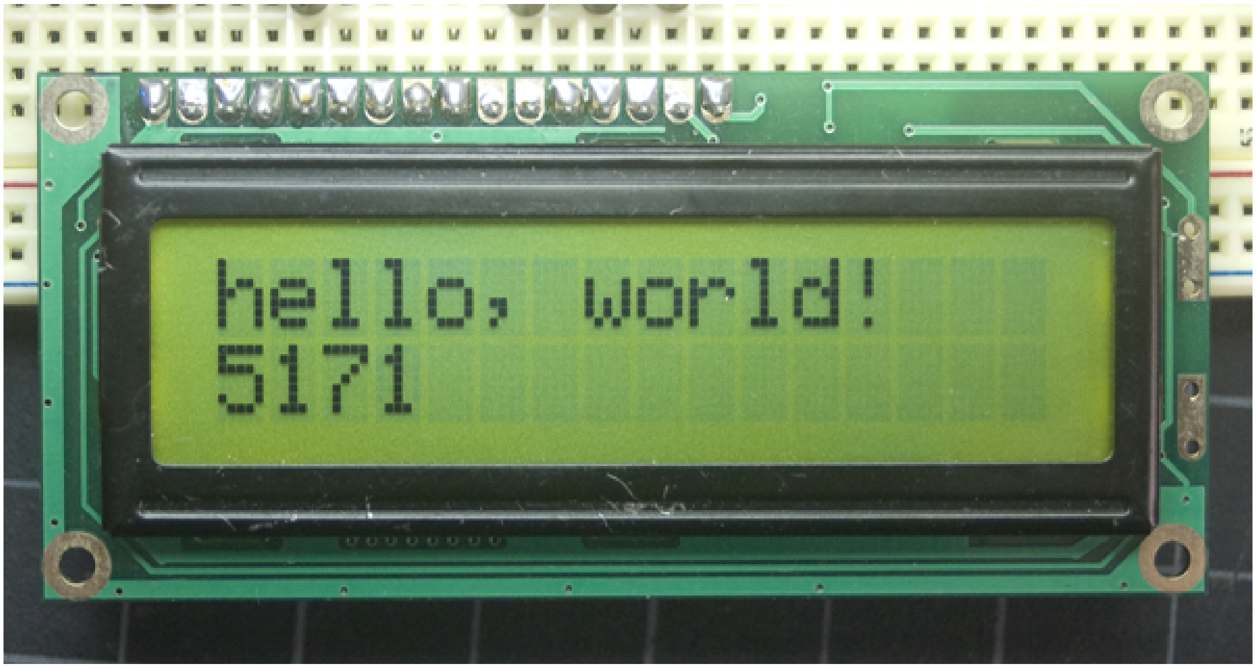

Fig. 1.1: Lcd display 1602

CHAPTER 2

Functions

LiquidCrystal()

Description

Creates a variable of type LiquidCrystal. The display can be controlled using 4 or 8 data lines. If the former, omit the pin numbers for d0 to d3 and leave those lines unconnected. The RW pin can be tied to ground instead of connected to a pin on the Arduino; if so, omit it from this function's parameters.

Syntax

```
LiquidCrystal(rs, enable, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)
```

Parameters

`rs`: the number of the Arduino pin that is connected to the RS pin on the LCD

`rw`: the number of the Arduino pin that is connected to the RW pin on the LCD (optional)

`enable`: the number of the Arduino pin that is connected to the enable pin on the LCD

`d0`, `d1`, `d2`, `d3`, `d4`, `d5`, `d6`, `d7`: the numbers of the Arduino pins that are connected to the corresponding data pins on the LCD.

`d0`, `d1`, `d2`, and `d3` are optional; if omitted, the LCD will be controlled using only the four data lines (`d4`, `d5`, `d6`, `d7`).

Example

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

void setup()
{
  lcd.begin(16,1);
  lcd.print("hello, world!");
}

void loop() {}
```

begin()

Description

Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display. begin() needs to be called before any other LCD library commands.

Syntax

```
lcd.begin(cols, rows)
```

Parameters

`lcd`: a variable of type LiquidCrystal

`cols`: the number of columns that the display has

`rows`: the number of rows that the display has

clear()

Description

Clears the LCD screen and positions the cursor in the upper-left corner.

Syntax

```
lcd.clear()
```

Parameters

`lcd`: a variable of type LiquidCrystal

home()

Description

Positions the cursor in the upper-left of the LCD. That is, use that location in outputting subsequent text to the display. To also clear the display, use the clear() function instead.

Syntax

```
lcd.home()
```

Parameters

lcd: a variable of type LiquidCrystal

setCursor()

Description

Position the LCD cursor; that is, set the location at which subsequent text written to the LCD will be displayed.

Syntax

```
lcd.setCursor(col, row)
```

Parameters

lcd: a variable of type LiquidCrystal

col: the column at which to position the cursor (with 0 being the first column)

row: the row at which to position the cursor (with 0 being the first row)

write()

Description

Write a character to the LCD.

Syntax

```
lcd.write(data)
```

Parameters

lcd: a variable of type LiquidCrystal

data: the character to write to the display

Returns

byte

`write()` will return the number of bytes written, though reading that number is optional

Example

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available()) {
    lcd.write(Serial.read());
  }
}
```

print()

Description

Prints text to the LCD.

Syntax

```
lcd.print(data)
lcd.print(data, BASE)
```

Parameters

lcd: a variable of type LiquidCrystal

data: the data to print (char, byte, int, long, or string)

BASE (optional): the base in which to print numbers:

- `BIN` for binary (base 2),

- DEC for decimal (base 10),
- OCT for octal (base 8),
- HEX for hexadecimal (base 16).

Returns

byte

print() will return the number of bytes written, though reading that number is optional

Example

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

void setup()
{
  lcd.print("hello, world!");
}

void loop() {}
```

cursor()

Description

Display the LCD cursor: an underscore (line) at the position to which the next character will be written.

Syntax

```
lcd.cursor()
```

Parameters

lcd a variable of type LiquidCrystal

Example

See: ‘’cursor() and noCursor()’‘

See also

noCursor()

noCursor()

Description

Hides the LCD cursor.

Syntax

```
lcd.noCursor()
```

Parameters

lcd: a variable of type LiquidCrystal

Example

See: ‘’cursor() and noCursor()’‘

See also

cursor()

blink()

Description

Display the blinking LCD cursor. If used in combination with the cursor() function, the result will depend on the particular display.

Syntax

```
lcd.blink()
```

Parameters

lcd: a variable of type LiquidCrystal

Example

“blink() and noBlink()”

See also

`noBlink()`

`cursor()`

noBlink()

Description

Turns off the blinking LCD cursor.

Syntax

```
lcd.noBlink()
```

Parameters

`lcd`: a variable of type LiquidCrystal

Example

“blink() and noBlink()”

See also

`blink()`

display()

Description

Turns on the LCD display, after it’s been turned off with noDisplay(). This will restore the text (and cursor) that was on the display.

Syntax

```
lcd.display()
```

Parameters

`lcd`: a variable of type LiquidCrystal

Example

“display() and noDisplay()”

See also

`noDisplay()`

noDisplay()

Description

Turns off the LCD display, without losing the text currently shown on it.

Syntax

```
lcd.noDisplay()
```

Parameters

`lcd:` a variable of type LiquidCrystal

Example

“display() and noDisplay()”

See also

`display()`

scrollDisplayLeft()

Description

Scrolls the contents of the display (text and cursor) one space to the left.

Syntax

```
lcd.scrollDisplayLeft()
```

Parameters

`lcd:` a variable of type LiquidCrystal

Example

“scrollDisplayLeft() and scrollDisplayRight()”

See also

`scrollDisplayRight()`

scrollDisplayRight()

Description

Scrolls the contents of the display (text and cursor) one space to the right.

Syntax

```
lcd.scrollDisplayRight()
```

Parameters

`lcd`: a variable of type LiquidCrystal

Example

“scrollDisplayLeft() and scrollDisplayRight()”

See also

`scrollDisplayLeft()`

autoscroll()

Description

Turns on automatic scrolling of the LCD. This causes each character output to the display to push previous characters over by one space. If the current text direction is left-to-right (the default), the display scrolls to the left; if the current direction is right-to-left, the display scrolls to the right. This has the effect of outputting each new character to the same location on the LCD.

Syntax

```
lcd.autoscroll()
```

Parameters

lcd: a variable of type LiquidCrystal

See also

`noAutoscroll()`

noAutoscroll()

Description

Turns off automatic scrolling of the LCD.

Syntax

```
lcd.noAutoscroll()
```

Parameters

lcd: a variable of type LiquidCrystal

See also

`autoscroll()`

leftToRight()

Description

Set the direction for text written to the LCD to left-to-right, the default. This means that subsequent characters written to the display will go from left to right, but does not affect previously-output text.

Syntax

```
lcd.leftToRight()
```

Parameters

lcd: a variable of type LiquidCrystal

See also

`rightToLeft()`

rightToLeft()

Description

Set the direction for text written to the LCD to right-to-left (the default is left-to-right). This means that subsequent characters written to the display will go from right to left, but does not affect previously-output text.

Syntax

```
lcd.rightToLeft()
```

Parameters

`lcd`: a variable of type LiquidCrystal

See also

`leftToRight()`

createChar()

Description

Create a custom character (gylph) for use on the LCD. Up to eight characters of 5x8 pixels are supported (numbered 0 to 7). The appearance of each custom character is specified by an array of eight bytes, one for each row. The five least significant bits of each byte determine the pixels in that row. To display a custom character on the screen, write() its number. NB : When referencing custom character "0", if it is not in a variable, you need to cast it as a byte, otherwise the compiler throws an error. See the example below.

Syntax

```
lcd.createChar(num, data)
```

Parameters

`lcd`: a variable of type LiquidCrystal

`num`: which character to create (0 to 7)

`data`: the character's pixel data

Example

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

byte smiley[8] - {
  B00000,
  B10001,
  B00000,
  B00000,
  B10001,
  B01110,
  B00000,
};

void setup() {
  lcd.createChar(0, smiley);
  lcd.begin(16, 2);
  lcd.write(byte(0));
}

void loop() {}
```

CHAPTER 3

Examples

LiquidCrystal - “Hello World!”

The LiquidCrystal library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This example sketch prints “Hello World!” to the LCD and shows the time in seconds since the Arduino was reset.

output of the sketch on a 2x16 LCD

The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The interface consists of the following pins:

- A register select (RS) pin that controls where in the LCD’s memory you’re writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD’s controller looks for instructions on what to do next.
- A Read/Write (R/W) pin that selects reading mode or writing mode
- An Enable pin that enables writing to the registers 8 data pins (D0 -D7). The states of these pins (high or low) are the bits that you’re writing to a register when you write, or the values you’re reading when you read.
- There’s also a display constrast pin (Vo),
- power supply pins (+5V and Gnd) and
- LED Backlight (Bklt+ and BKlt-) pins that you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.

The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. The LiquidCrystal Library simplifies this for you so you don’t need to know the low-level instructions.

The Hitachi-compatible LCDs can be controlled in two modes: 4-bit or 8-bit. The 4-bit mode requires seven I/O pins from the Arduino, while the 8-bit mode requires 11 pins. For displaying text on the screen, you can do most everything in 4-bit mode, so example shows how to control a 2x16 LCD in 4-bit mode.

Hardware Required

- Arduino Board
- LCD Screen (compatible with Hitachi HD44780 driver)
- pin headers to solder to the LCD display pins
- 10k Potentiometer
- breadboard
- hook-up wire

Circuit

Before wiring the LCD screen to your Arduino we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above.

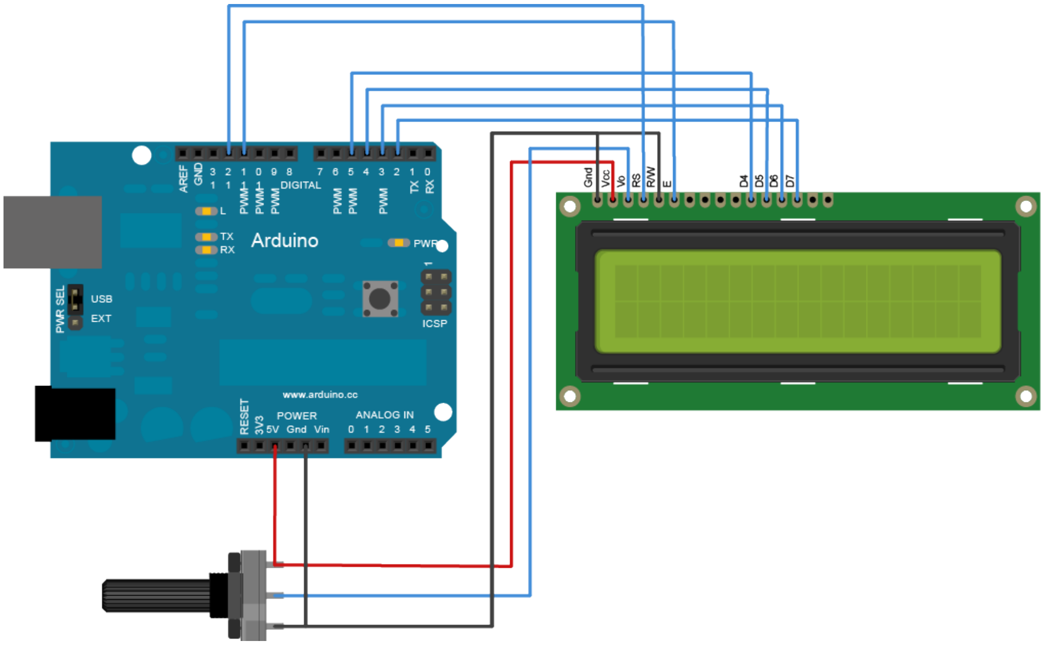

Fig. 3.1: Lcd display connections

To wire your LCD screen to your Arduino, connect the following pins:

- LCD RS pin to digital pin 12
- LCD Enable pin to digital pin 11
- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2
- Additionally, wire a 10K pot to +5V and GND, with it's wiper (output) to LCD screens VO pin (pin3).

(Image developed using Fritzing. For more circuit examples, see the Fritzing project page)

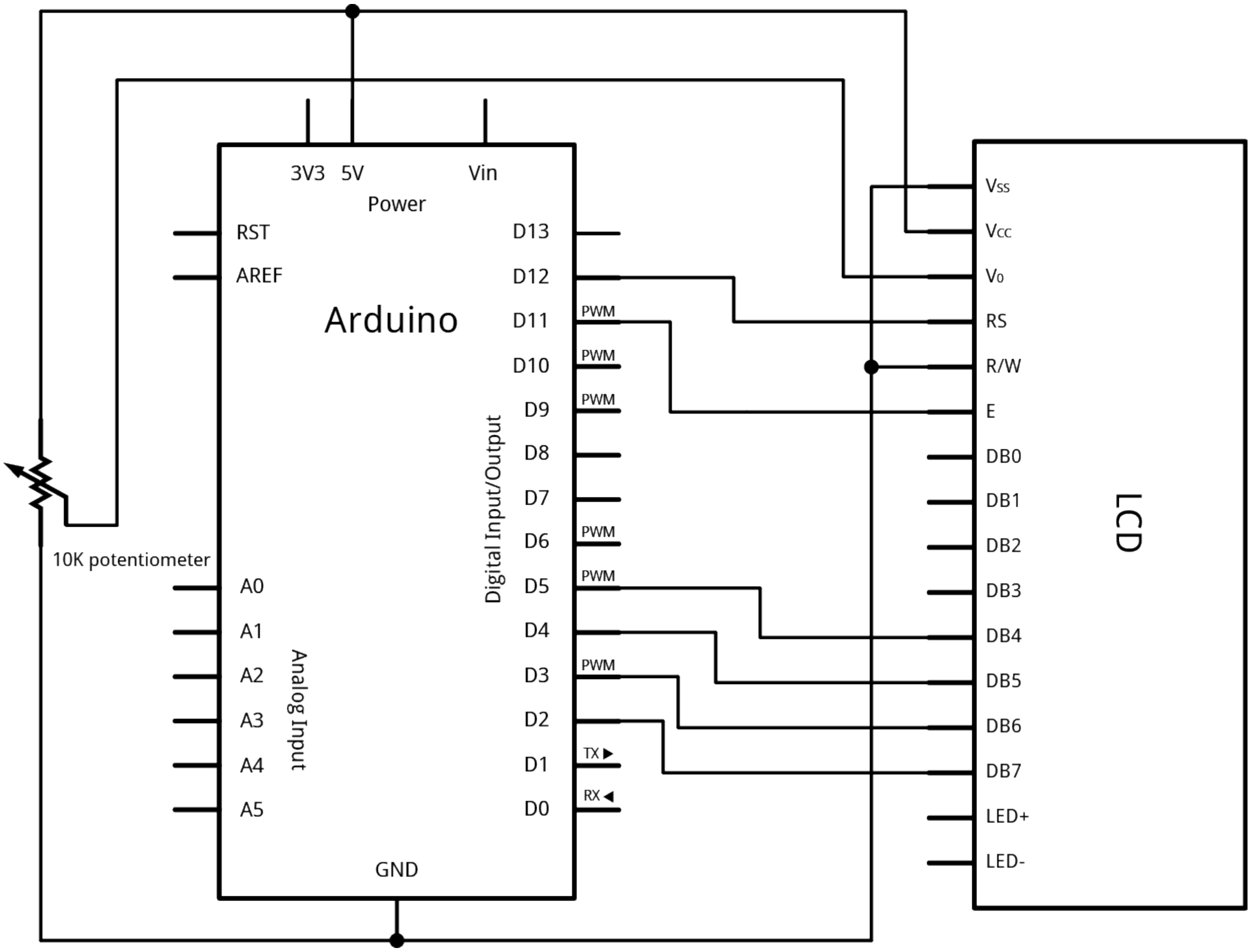

Fig. 3.2: Lcd display schem

Code

```
/*
  LiquidCrystal Library - Hello World

Demonstrates the use a 16x2 LCD display.  The LiquidCrystal
library works with all LCD displays that are compatible with the
Hitachi HD44780 driver. There are many of them out there, and you
can usually tell them by the 16-pin interface.

This sketch prints "Hello World!" to the LCD
and shows the time.

  The circuit:
* LCD RS pin to digital pin 12
* LCD Enable pin to digital pin 11
* LCD D4 pin to digital pin 5
* LCD D5 pin to digital pin 4
* LCD D6 pin to digital pin 3
* LCD D7 pin to digital pin 2
* LCD R/W pin to ground
* 10K resistor:
* ends to +5V and ground
* wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008
by David A. Mellis
library modified 5 Jul 2009
by Limor Fried (http://www.ladyada.net)
example added 9 Jul 2009
by Tom Igoe
modified 22 Nov 2010
by Tom Igoe

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/LiquidCrystal
*/

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

See Also

- lcd.begin()
- lcd.print()
- lcd.setCursor()
- Liquid Crystal Library
- Blink: control of the block-style cursor.
- Cursor: control of the underscore-style cursor.
- Display: quickly blank the display without losing what's on it.
- TextDirection: control which way text flows from the cursor.
- Scroll: scroll text left and right.
- Serial input: accepts serial input, displays it.
- SetCursor: set the cursor position.
- Autoscroll: shift text right and left.

LiquidCrystal - blink() and noBlink()

The Liquid Crystal Library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This example sketch shows how to use the blink() and noBlink() methods to blink a block-style cursor.

Hardware Required

- Arduino Board
- LCD Screen (compatible with Hitachi HD44780 driver)
- pin headers to solder to the LCD display pins
- 10k Potentiometer
- breadboard
- hook-up wire

Circuit

Before wiring the LCD screen to your Arduino we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above.

To wire your LCD screen to your Arduino, connect the following pins:

- LCD RS pin to digital pin 12
- LCD Enable pin to digital pin 11

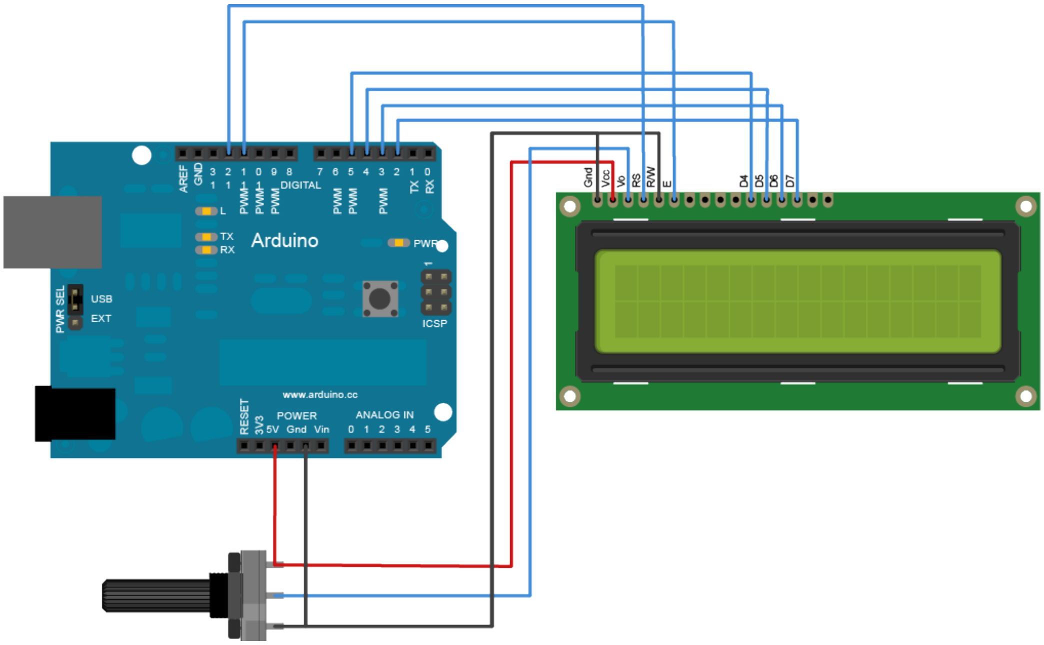

Fig. 3.3: Lcd display connections

- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2
- Additionally, wire a 10K pot to +5V and GND, with it's wiper (output) to LCD screens VO pin (pin3).

(Image developed using Fritzing. For more circuit examples, see the Fritzing project page)

Code

```
/*
  LiquidCrystal Library - Blink

Demonstrates the use a 16x2 LCD display.  The LiquidCrystal
library works with all LCD displays that are compatible with the
Hitachi HD44780 driver. There are many of them out there, and you
can usually tell them by the 16-pin interface.

This sketch prints "Hello World!" to the LCD and makes the
cursor block blink.

The circuit:
* LCD RS pin to digital pin 12
* LCD Enable pin to digital pin 11
* LCD D4 pin to digital pin 5
* LCD D5 pin to digital pin 4
* LCD D6 pin to digital pin 3
* LCD D7 pin to digital pin 2
* LCD R/W pin to ground
```

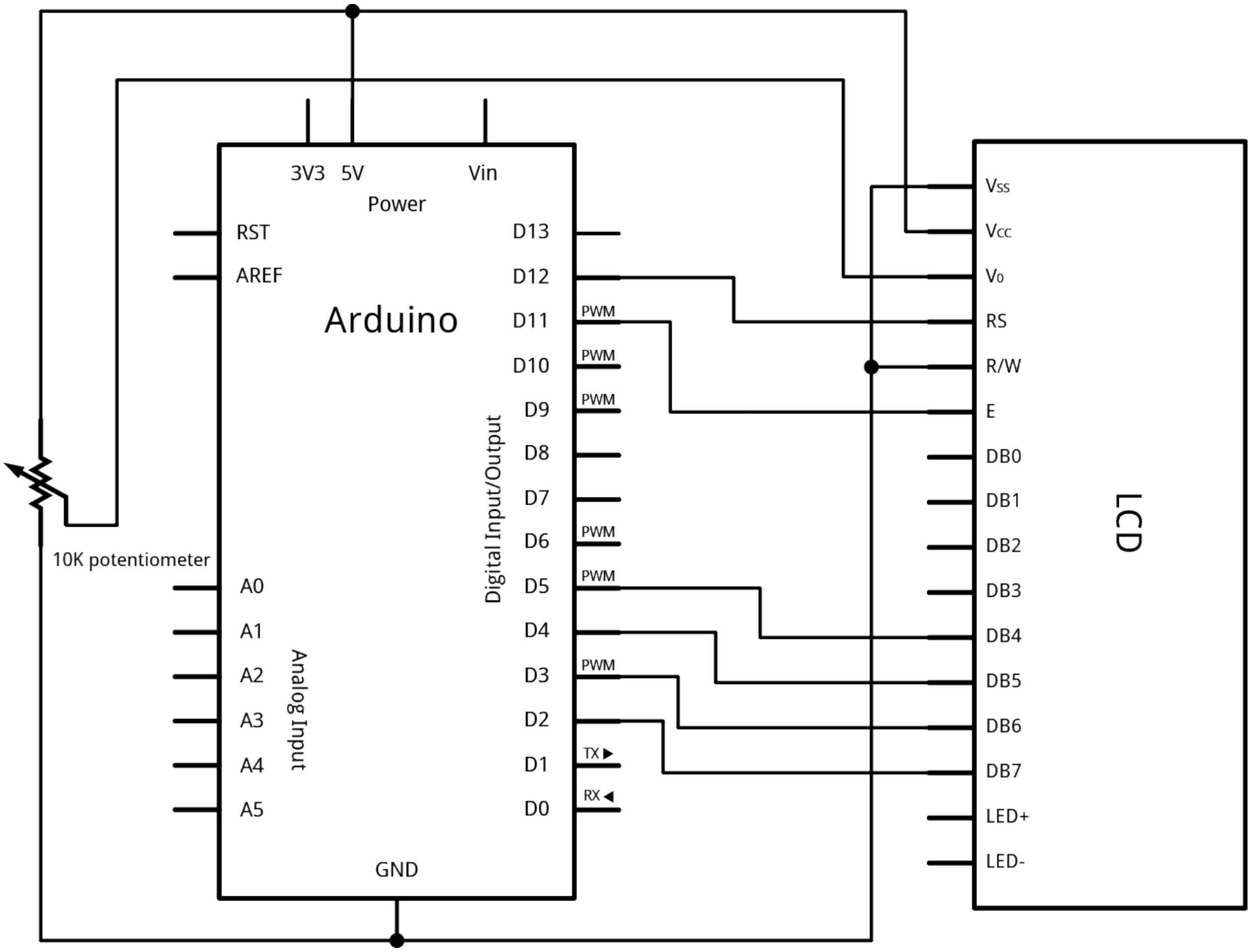

Fig. 3.4: Lcd display schem

```
* 10K resistor:
  * ends to +5V and ground
  * wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008
by David A. Mellis
library modified 5 Jul 2009
by Limor Fried (http://www.ladyada.net)
example added 9 Jul 2009
by Tom Igoe
modified 22 Nov 2010
by Tom Igoe

This example code is in the public domain.

http://arduino.cc/en/Tutorial/LiquidCrystalBlink

*/

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // Turn off the blinking cursor:
  lcd.noBlink();
  delay(3000);
  // Turn on the blinking cursor:
  lcd.blink();
  delay(3000);
}
```

See Also

- lcd.begin()
- lcd.print()
- lcd.Blink()
- lcd.noBlink()
- Liquid Crystal Library
- Blink: control of the block-style cursor.
- Cursor: control of the underscore-style cursor.
- Display: quickly blank the display without losing what’s on it.

- TextDirection: control which way text flows from the cursor.
- Scroll: scroll text left and right.
- Serial input: accepts serial input, displays it.
- SetCursor: set the cursor position.
- Autoscroll: shift text right and left.

LiquidCrystal - cursor() and noCursor()

The Liquid Crystal Library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface. This example sketch shows how to use the cursor() and noCursor() methods to control an underscore-style cursor. Hardware Required

Arduino Board LCD Screen (compatible with Hitachi HD44780 driver) pin headers to solder to the LCD display pins 10k Potentiometer breadboard hook-up wire Circuit

Before wiring the LCD screen to your Arduino we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above. To wire your LCD screen to your Arduino, connect the following pins: LCD RS pin to digital pin 12 LCD Enable pin to digital pin 11 LCD D4 pin to digital pin 5 LCD D5 pin to digital pin 4 LCD D6 pin to digital pin 3 LCD D7 pin to digital pin 2 Additionally, wire a 10K pot to +5V and GND, with it's wiper (output) to LCD screens VO pin (pin3). click the images to enlarge

image developed using Fritzing. For more circuit examples, see the Fritzing project page

Code

/*

LiquidCrystal Library - Cursor

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch prints "Hello World!" to the LCD and uses the cursor() and noCursor() methods to turn on and off the cursor.

The circuit: * LCD RS pin to digital pin 12 * LCD Enable pin to digital pin 11 * LCD D4 pin to digital pin 5 * LCD D5 pin to digital pin 4 * LCD D6 pin to digital pin 3 * LCD D7 pin to digital pin 2 * LCD R/W pin to ground * 10K resistor: * ends to +5V and ground * wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008 by David A. Mellis library modified 5 Jul 2009 by Limor Fried (http://www.ladyada.net) example added 9 Jul 2009 by Tom Igoe modified 22 Nov 2010 by Tom Igoe

This example code is in the public domain.

http://arduino.cc/en/Tutorial/LiquidCrystalCursor

*/

// include the library code: #include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() { // set up the LCD's number of columns and rows: lcd.begin(16, 2); // Print a message to the LCD. lcd.print("hello, world!");

}

void loop() { // Turn off the cursor: lcd.noCursor(); delay(500);

// Turn on the cursor:

lcd.cursor(); delay(500);

} [Get Code] See Also:

lcd.begin() lcd.print() lcd.Cursor() lcd.NoCursor() Liquid Crystal Library Blink: control of the block-style cursor. Display: quickly blank the display without losing what's on it. TextDirection: control which way text flows from the cursor. Scroll: scroll text left and right. Serial input: accepts serial input, displays it. SetCursor: set the cursor position. Autoscroll: shift text right and left.

LiquidCrystal - display() and noDisplay()

The Liquid Crystal Library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface. This example sketch shows how to use the display() and noDisplay() methods to turn on and off the display. The text to be displayed will still be preserved when you use noDisplay() so it's a quick way to blank the display without losing everything on it. Hardware Required

Arduino Board LCD Screen (compatible with Hitachi HD44780 driver) pin headers to solder to the LCD display pins 10k Potentiometer breadboard hook-up wire Circuit

Before wiring the LCD screen to your Arduino we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above. To wire your LCD screen to your Arduino, connect the following pins: LCD RS pin to digital pin 12 LCD Enable pin to digital pin 11 LCD D4 pin to digital pin 5 LCD D5 pin to digital pin 4 LCD D6 pin to digital pin 3 LCD D7 pin to digital pin 2 Additionally, wire a 10K pot to +5V and GND, with it's wiper (output) to LCD screen's VO pin (pin3). click the images to enlarge

image developed using Fritzing. For more circuit examples, see the Fritzing project page

Code

/*

LiquidCrystal Library - display() and noDisplay()

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch prints "Hello World!" to the LCD and uses the display() and noDisplay() functions to turn on and off the display.

The circuit: * LCD RS pin to digital pin 12 * LCD Enable pin to digital pin 11 * LCD D4 pin to digital pin 5 * LCD D5 pin to digital pin 4 * LCD D6 pin to digital pin 3 * LCD D7 pin to digital pin 2 * LCD R/W pin to ground * 10K resistor: * ends to +5V and ground * wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008 by David A. Mellis library modified 5 Jul 2009 by Limor Fried (http://www.ladyada.net) example added 9 Jul 2009 by Tom Igoe modified 22 Nov 2010 by Tom Igoe

This example code is in the public domain.

http://arduino.cc/en/Tutorial/LiquidCrystalDisplay

*/

// include the library code: #include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() { // set up the LCD's number of columns and rows: lcd.begin(16, 2); // Print a message to the LCD. lcd.print("hello, world!");

}

void loop() { // Turn off the display: lcd.noDisplay(); delay(500);

// Turn on the display:

lcd.display(); delay(500);

} [Get Code] See Also:

lcd.begin() lcd.print() lcd.Display() lcd.NoDisplay() Liquid Crystal Library Blink: control of the block-style cursor. Cursor: control of the underscore-style cursor. TextDirection: control which way text flows from the cursor. Scroll: scroll text left and right. Serial input: accepts serial input, displays it. SetCursor: set the cursor position. Autoscroll: shift text right and left.

LiquidCrystal - Text Direction (leftToRight() and rightToLeft()

The Liquid Crystal Library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface. This example sketch shows how to use the leftToRight() and rightToLeft() methods. These methods control which way text flows from the cursor. rightToLeft() causes text to flow to the left from the cursor, as if the display is right-justified. leftToRight() causes text to flow to the right from the cursor, as if the display is left-justified. This sketch prints a through l right to left, then m through r left to right, then s through z right to left again. Hardware Required

Arduino Board LCD Screen (compatible with Hitachi HD44780 driver) pin headers to solder to the LCD display pins 10k Potentiometer breadboard hook-up wire Circuit

Before wiring the LCD screen to your Arduino we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above. To wire your LCD screen to your Arduino, connect the following pins: LCD RS pin to digital pin 12 LCD Enable pin to digital pin 11 LCD D4 pin to digital pin 5 LCD D5 pin to digital pin 4 LCD D6 pin to digital pin 3 LCD D7 pin to digital pin 2 Additionally, wire a 10K pot to +5V and GND, with it's wiper (output) to LCD screens VO pin (pin3). click the images to enlarge

image developed using Fritzing. For more circuit examples, see the Fritzing project page

Code

/*

LiquidCrystal Library - TextDirection

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch demonstrates how to use leftToRight() and rightToLeft() to move the cursor.

The circuit: * LCD RS pin to digital pin 12 * LCD Enable pin to digital pin 11 * LCD D4 pin to digital pin 5 * LCD D5 pin to digital pin 4 * LCD D6 pin to digital pin 3 * LCD D7 pin to digital pin 2 * LCD R/W pin to ground * 10K resistor: * ends to +5V and ground * wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008 by David A. Mellis library modified 5 Jul 2009 by Limor Fried (http://www.ladyada.net) example added 9 Jul 2009 by Tom Igoe modified 22 Nov 2010 by Tom Igoe

This example code is in the public domain.

http://arduino.cc/en/Tutorial/LiquidCrystalTextDirection

*/

// include the library code: #include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int thisChar - ‘a’;

void setup() { // set up the LCD’s number of columns and rows: lcd.begin(16, 2); // turn on the cursor: lcd.cursor();

}

void loop() { // reverse directions at ‘m’: if (thisChar – ‘m’) {

// go right for the next letter lcd.rightToLeft();

} // reverse again at ‘s’: if (thisChar – ‘s’) {

// go left for the next letter lcd.leftToRight();

} // reset at ‘z’: if (thisChar > ‘z’) {

// go to (0,0): lcd.home(); // start again at 0 thisChar - ‘a’;

} // print the character lcd.write(thisChar); // wait a second: delay(1000); // increment the letter: thisChar++;

} [Get Code] See Also:

lcd.begin() lcd.print() lcd.rightToLeft() lcd.leftToRight() lcd.home() Liquid Crystal Library Blink: control of the block-style cursor. Cursor: control of the underscore-style cursor. Display: quickly blank the display without losing what’s on it. Scroll: scroll text left and right. Serial input: accepts serial input, displays it. SetCursor: set the cursor position. Autoscroll: shift text right and left.

LiquidCrystal - autoscroll()

The Liquid Crystal Library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface. This example sketch shows how to use the autoscroll() and noAutoscroll() methods to move all the text on the display left or right. autoscroll() moves all the text one space to the left each time a letter is added noAutoscroll() turns scrolling off This sketch prints the characters 0 to 9 with autoscroll off, then moves the cursor to the bottom right, turns autoscroll on, and prints them again. Hardware Required

Arduino Board LCD Screen (compatible with Hitachi HD44780 driver) pin headers to solder to the LCD display pins 10k Potentiometer breadboard hook-up wire Circuit

Before wiring the LCD screen to your Arduino we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above. To wire your LCD screen to your Arduino, connect the following pins: LCD RS pin to digital pin 12 LCD Enable pin to digital pin 11 LCD D4 pin to digital pin 5 LCD D5 pin to digital pin 4 LCD D6 pin to digital pin 3 LCD D7 pin to digital pin 2 Additionally, wire a 10K pot to +5V and GND, with it’s wiper (output) to the LCD screen’s VO pin (pin3). click the images to enlarge

image developed using Fritzing. For more circuit examples, see the Fritzing project page

Code

/*

LiquidCrystal Library - Autoscroll

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch demonstrates the use of the autoscroll() and noAutoscroll() functions to make new text scroll or not.

> The circuit: * LCD RS pin to digital pin 12 * LCD Enable pin to digital pin 11 * LCD D4 pin to digital pin 5 * LCD D5 pin to digital pin 4 * LCD D6 pin to digital pin 3 * LCD D7 pin to digital pin 2 * LCD R/W pin to ground * 10K resistor: * ends to +5V and ground * wiper to LCD VO pin (pin 3)
>
> Library originally added 18 Apr 2008 by David A. Mellis library modified 5 Jul 2009 by Limor Fried (http://www.ladyada.net) example added 9 Jul 2009 by Tom Igoe modified 22 Nov 2010 by Tom Igoe
>
> This example code is in the public domain.
>
> http://arduino.cc/en/Tutorial/LiquidCrystalAutoscroll
>
> */

// include the library code: #include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() { // set up the LCD's number of columns and rows: lcd.begin(16,2);

}

void loop() { // set the cursor to (0,0): lcd.setCursor(0, 0); // print from 0 to 9: for (int thisChar - 0; thisChar < 10; thisChar++) {

> lcd.print(thisChar); delay(500);
>
> }
>
> // set the cursor to (16,1): lcd.setCursor(16,1); // set the display to automatically scroll: lcd.autoscroll(); // print from 0 to 9: for (int thisChar - 0; thisChar < 10; thisChar++) {
>
> lcd.print(thisChar); delay(500);
>
> } // turn off automatic scrolling lcd.noAutoscroll();
>
> // clear screen for the next loop: lcd.clear();

} [Get Code] See Also:

lcd.begin() lcd.print() lcd.setCursor() lcd.autoscroll() lcd.noAutoscroll() lcd.clear() for() Blink: control of the block-style cursor. Cursor: control of the underscore-style cursor. Display: quickly blank the display without losing what's on it. TextDirection: control which way text flows from the cursor. Scroll: scroll text left and right. Serial input: accepts serial input, displays it. SetCursor: set the cursor position.

LiquidCrystal - Serial Input

The Liquid Crystal Library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface. This example sketch accepts serial input from a host computer and displays it on the LCD. To use it, upload the sketch, then open the Serial Monitor and type some characters and click Send. The text will appear on your LCD. Hardware Required

LiquidCrystal - setCursor()

The Liquid Crystal Library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface. This example sketch shows how to use the setCursor() method to reposition the cursor. To move the cursor, just call setCursor() with a row and column position. For example, for a 2x16 display: lcd.setCursor(0, 0); // top left lcd.setCursor(15, 0); // top right lcd.setCursor(0, 1); // bottom left lcd.setCursor(15, 1); // bottom right

Hardware Required

Arduino Board LCD Screen (compatible with Hitachi HD44780 driver) pin headers to solder to the LCD display pins 10k Potentiometer breadboard hook-up wire Circuit

Before wiring the LCD screen to your Arduino we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above. To wire your LCD screen to your Arduino, connect the following pins: LCD RS pin to digital pin 12 LCD Enable pin to digital pin 11 LCD D4 pin to digital pin 5 LCD D5 pin to digital pin 4 LCD D6 pin to digital pin 3 LCD D7 pin to digital pin 2 Additionally, wire a 10K pot to +5V and GND, with it's wiper (output) to LCD screens VO pin (pin3). click the images to enlarge

image developed using Fritzing. For more circuit examples, see the Fritzing project page

Code

/*

LiquidCrystal Library - setCursor

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch prints to all the positions of the LCD using the setCursor(0 method:

The circuit:

- LCD RS pin to digital pin 12
- LCD Enable pin to digital pin 11
- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2
- LCD R/W pin to ground
- 10K resistor:
- ends to +5V and ground
- wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008 by David A. Mellis library modified 5 Jul 2009 by Limor Fried (http://www.ladyada.net) example added 9 Jul 2009 by Tom Igoe modified 22 Nov 2010 by Tom Igoe

This example code is in the public domain.

http://arduino.cc/en/Tutorial/LiquidCrystalSetCursor

*/

// include the library code: #include <LiquidCrystal.h>

// these constants won't change. But you can change the size of // your LCD using them: const int numRows - 2; const int numCols - 16;

// initialize the library with the numbers of the interface pins LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() { // set up the LCD's number of columns and rows: lcd.begin(numCols,numRows);

}

void loop() { // loop from ASCII 'a' to ASCII 'z': for (int thisLetter - 'a'; thisLetter <- 'z'; thisLetter++) {

// loop over the columns: for (int thisCol - 0; thisCol < numRows; thisCol++) {

// loop over the rows: for (int thisRow - 0; thisRow < numCols; thisRow++) {

// set the cursor position: lcd.setCursor(thisRow,thisCol); // print the letter: lcd.write(thisLetter); delay(200);

}

}

}

} [Get Code] See Also:

lcd.begin() lcd.print() lcd.setCursor() for() Liquid Crystal Library Blink: control of the block-style cursor. Cursor: control of the underscore-style cursor. Display: quickly blank the display without losing what's on it. TextDirection: control which way text flows from the cursor. Scroll: scroll text left and right. Serial input: accepts serial input, displays it. Autoscroll: shift text right and left.

LiquidCrystal - scrollDisplayLeft() and scrollDisplayRight()

The Liquid Crystal Library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface. This example sketch shows how to use the scrollDisplayLeft() and scrollDisplayRight() methods to reverse the direction the text is flowing. It prints "Hello World!", scrolls it offscreen to the left, then offscreen to the right, then back to home. Hardware Required

Arduino Board LCD Screen (compatible with Hitachi HD44780 driver) pin headers to solder to the LCD display pins 10k Potentiometer breadboard hook-up wire Circuit

Before wiring the LCD screen to your Arduino we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above. To wire your LCD screen to your Arduino, connect the following pins: LCD RS pin to digital pin 12 LCD Enable pin to digital pin 11 LCD D4 pin to digital pin 5 LCD D5 pin to digital pin 4 LCD D6 pin to digital pin 3 LCD D7 pin to digital pin 2 Additionally, wire a 10K pot to +5V and GND, with it's wiper (output) to LCD screen's VO pin (pin3). click the images to enlarge

image developed using Fritzing. For more circuit examples, see the Fritzing project page

Code

/*

LiquidCrystal Library - scrollDisplayLeft() and scrollDisplayRight()

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch prints "Hello World!" to the LCD and uses the scrollDisplayLeft() and scrollDisplayRight() methods to scroll the text.

The circuit:

- LCD RS pin to digital pin 12
- LCD Enable pin to digital pin 11
- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2
- LCD R/W pin to ground

- 10K resistor:
- ends to +5V and ground
- wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008 by David A. Mellis library modified 5 Jul 2009 by Limor Fried (http://www.ladyada.net) example added 9 Jul 2009 by Tom Igoe modified 22 Nov 2010 by Tom Igoe

This example code is in the public domain.

http://arduino.cc/en/Tutorial/LiquidCrystalScroll

*/

// include the library code: #include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() { // set up the LCD's number of columns and rows: lcd.begin(16, 2); // Print a message to the LCD. lcd.print("hello, world!"); delay(1000);

}

void loop() { // scroll 13 positions (string length) to the left // to move it offscreen left: for (int positionCounter - 0; positionCounter < 13; positionCounter++) {

// scroll one position left: lcd.scrollDisplayLeft(); // wait a bit: delay(150);

}

// scroll 29 positions (string length + display length) to the right // to move it offscreen right: for (int positionCounter - 0; positionCounter < 29; positionCounter++) {

// scroll one position right: lcd.scrollDisplayRight(); // wait a bit: delay(150);

}

// scroll 16 positions (display length + string length) to the left // to move it back to center:

for (int positionCounter - 0; positionCounter < 16; positionCounter++) { // scroll one position left: lcd.scrollDisplayLeft(); // wait a bit: delay(150);

}

// delay at the end of the full loop: delay(1000);

} [Get Code] See Also:

lcd.begin() lcd.print() lcd.scrollDisplayLeft() lcd.scrollDisplayRight() Liquid Crystal Library Blink: control of the block-style cursor. Cursor: control of the underscore-style cursor. Display: quickly blank the display without losing what's on it. TextDirection: control which way text flows from the cursor. Serial input: accepts serial input, displays it. SetCursor: set the cursor position. Autoscroll: shift text right and left.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search