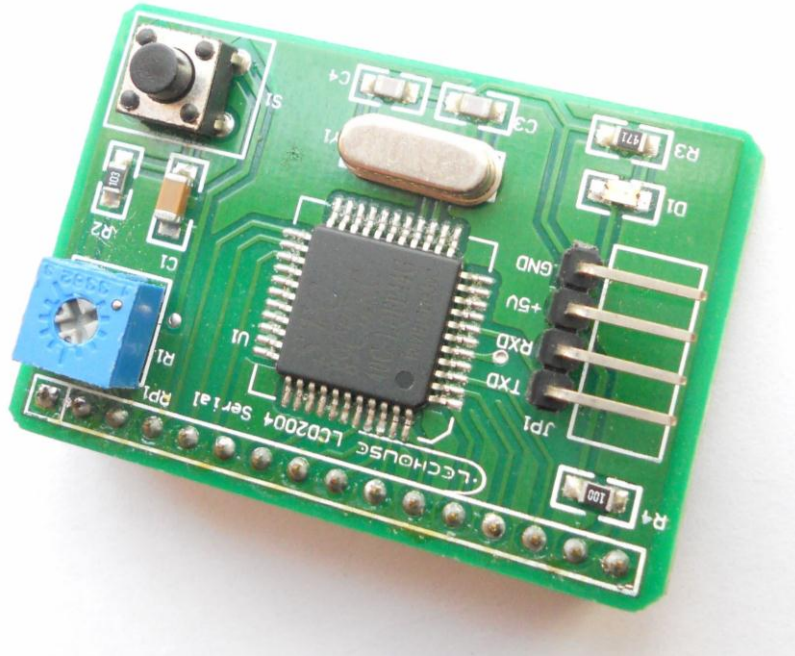
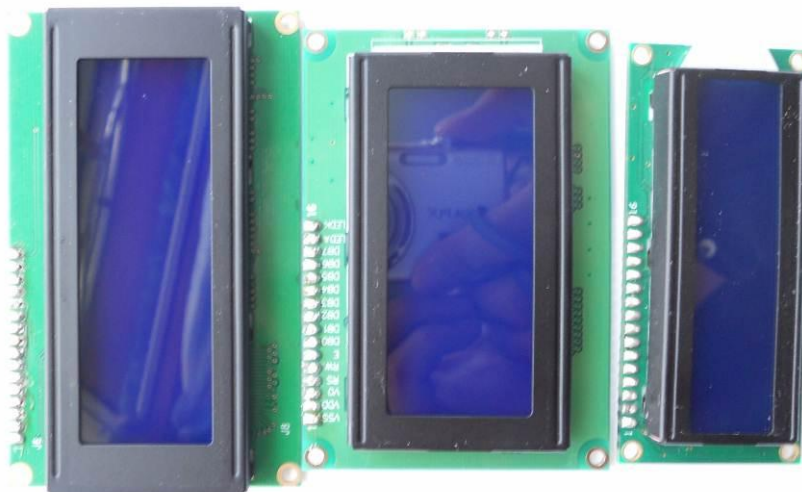


UART LCD Controller Module



Description



LCD2004

LCD1604

LCD1602

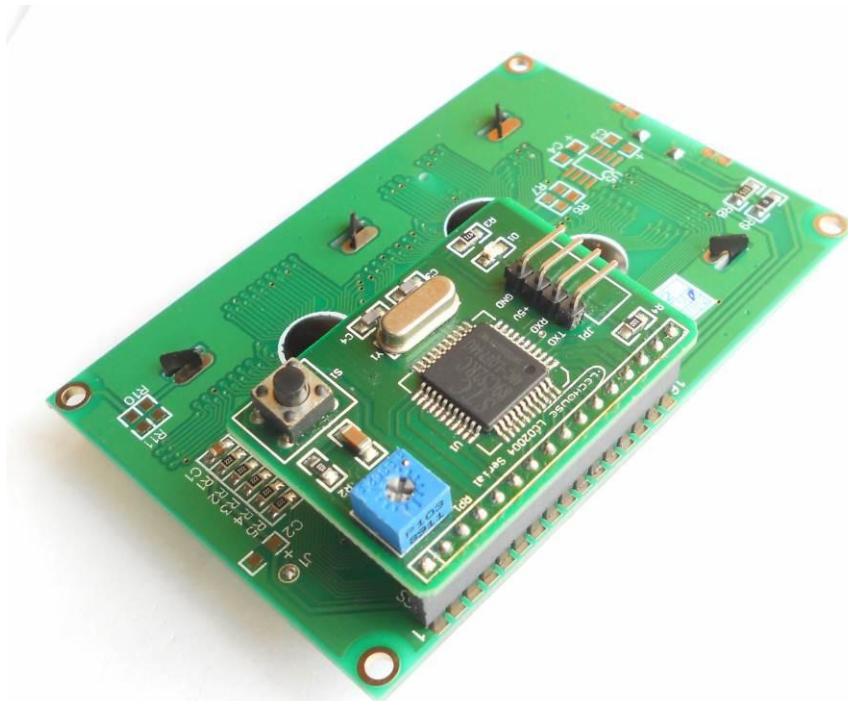
It is not very easy to control those LCD1602, LCD1604 or LCD2004 modules above, especially for Arduino users. It will take as much as 16 pins of Arduino. If you want to control other modules with Arduino meanwhile, of course, you will find many wires and probably no enough available pins on Arduino board. So we designed this UART controller. It is compatible with LCD1602, LCD1604 and LCD2004. It

only takes 4 pins on Arduino (in fact, only 2 data pins- Tx and Rx). You could control it via hardware serial interface on Arduino, or any 2 data pins with SoftSerial library.

Hardware connection

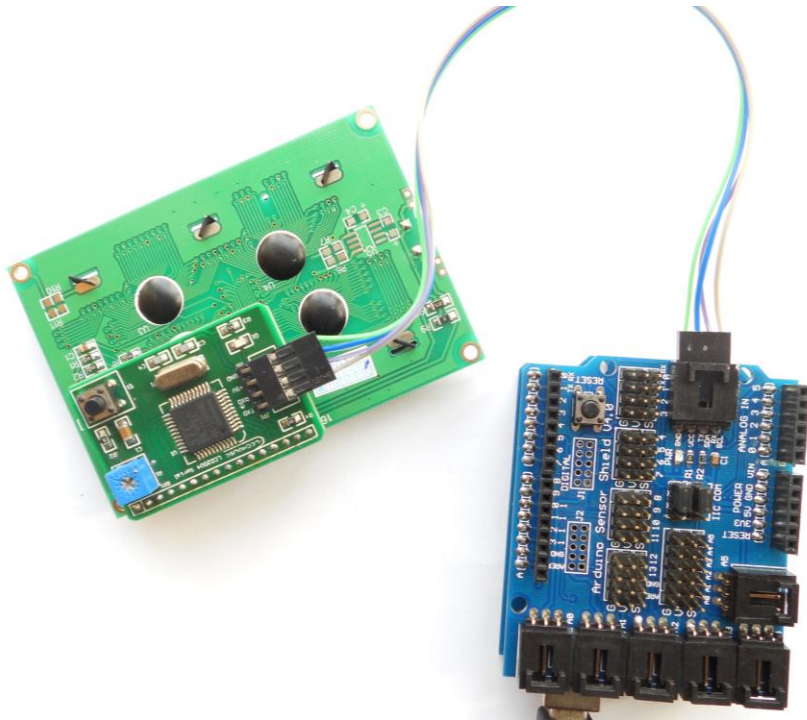
- **VCC:** +5V
- **GND:** Ground
- **TXD:** UART transmission pin, connect with host RX pin.
- **RXD:** UART reception pin, connect with host TX pin.

Just plug in the LCD on the controller as below:



Connecting with Arduino is pretty easy if you have Arduino Sensor Shield. If you don't have sensor shield, connect it in the following way:

Controller 5V	---	Arduino 5V
Controller GND	---	Arduino GND
Controller TXD	---	Arduino RX (Pin 0)
Controller RXD	---	Arduino TX (Pin 1)



Command

Operation is very easy. Just send command to LCD controller via UART interface.

The basic command format is “\$cmd [par1] [par2]\n”. It begins with the “\$” character and ends with “\n”. For some commands, parameter is optional, and some commands have no parameter.

Note

1. There is no space between \$ and cmd, so as to between [par2] and \n in the command line. But there is one space between cmd and [par1] , [par1] and [par2].
2. Command is compatible with both upper and lower case, or mix mode. For example, ‘cLeAr’ also works well as “CLEAR” or “clear”.
3. This controller will change the line automatically. For controlling LCD1602, it could display 32 characters, with 16 in each of the 2 lines. If you print “ABCDEFGHJKLMNOPQRSTUVWXYZ0123456”, on LCD the first line will be “ABCDEFGHJKLMNOP”, and the second line will be “QRSTUVWXYZ012345”. Note that the last “6” will be discarded because the LCD could only display 32 characters.
4. Screwing the changeable resistor on the module could change the brightness of the LCD. If your lcd displays nothing on it, you might need to do this changing.

The following table will show you details about how to use those commands.

Command	Format	Description	Example for Arduino
LCD1602	<code>\$lcd1602\n</code>	Choose LCD type to control: LCD1602	<code>Serial.print("\$LCD1602 \n");</code> <code>Serial.print("\$LCD1604 \n");</code>
LCD1604	<code>\$lcd1604\n</code>	LCD1604	<code>Serial.print("\$LCD2004 \n");</code>
LCD2004	<code>\$lcd2004\n</code>	LCD2004 This setting will be stored this module. It means, if you use it to control LCD2004 and won't change the LCD type, you only need to use this command once.	
HOME	<code>\$home\n</code>	The HOME command is to move the cursor to the start position (0,0).	<code>Serial.print("\$HOME\n");</code>
GO	<code>\$GO x y\n</code>	The "Go x y" command is to move the cursor to the coordinate (x,y) on the LCD. x : horizontal ordinate. y : vertical coordinate. x and y start from 0. If the LCD is LCD2004, it has 4 lines and 20 characters per line. So the range of x is 0-4, y is 0-19. The start position is (0,0).	<code>Serial.print("\$GO 3 10\n");</code>
CLEAR	<code>\$clear\n</code>	The CLEAR command is to clear all the characters on the LCD. The cursor will return to start point "0, 0".	<code>Serial.print("\$CLEAR\n");</code>
PRINT	<code>\$print [para]\n</code>	This command is to print your characters from the current cursor position. [Para] is the characters which will be displayed on LCD.	<code>Serial.print("\$PRINT Hello World!\n");</code>
CURSOR	<code>\$cursor [status]\n</code>	This command to set the cursor on, off or blink. [status]: ON, OFF, or BLINKING	<code>//Show cursor</code> <code>Serial.print("\$CURSOR ON\n");</code> <code>//Hide cursor</code> <code>Serial.print("\$CURSOR OFF\n");</code> <code>//Set cursor blinking</code> <code>Serial.print("\$CURSOR BLINKING\n");</code>
CLOSE	<code>\$close\n</code>	LCD display off	<code>Serial.print("\$CLOSE \n");</code>
OPEN	<code>\$open\n</code>	LCD display on	<code>Serial.print("\$OPEN \n");</code>

Example for Arduino

This demo simply shows how to display some characters with this LCD controller. We use to control LCD2004 here.

```
void setup(void)
{
  /* baud 9600 */
  Serial.begin(9600);

  /** set LCD type
  LCD1602, LCD1604 and LCD2004 are supported**/
  Serial.print("$LCD1604\n");
  delay(2000);

  /** clear screen */
  Serial.print("$clear\n");

  /** go home (top left corner)*/
  Serial.print("$Home\n");
  delay(2000);

  /** cursor on and blinking */
  Serial.print("$CURSOR BLINKING\n");
  delay(2000);

  Serial.print("$go 0 0\n");
  delay(500);

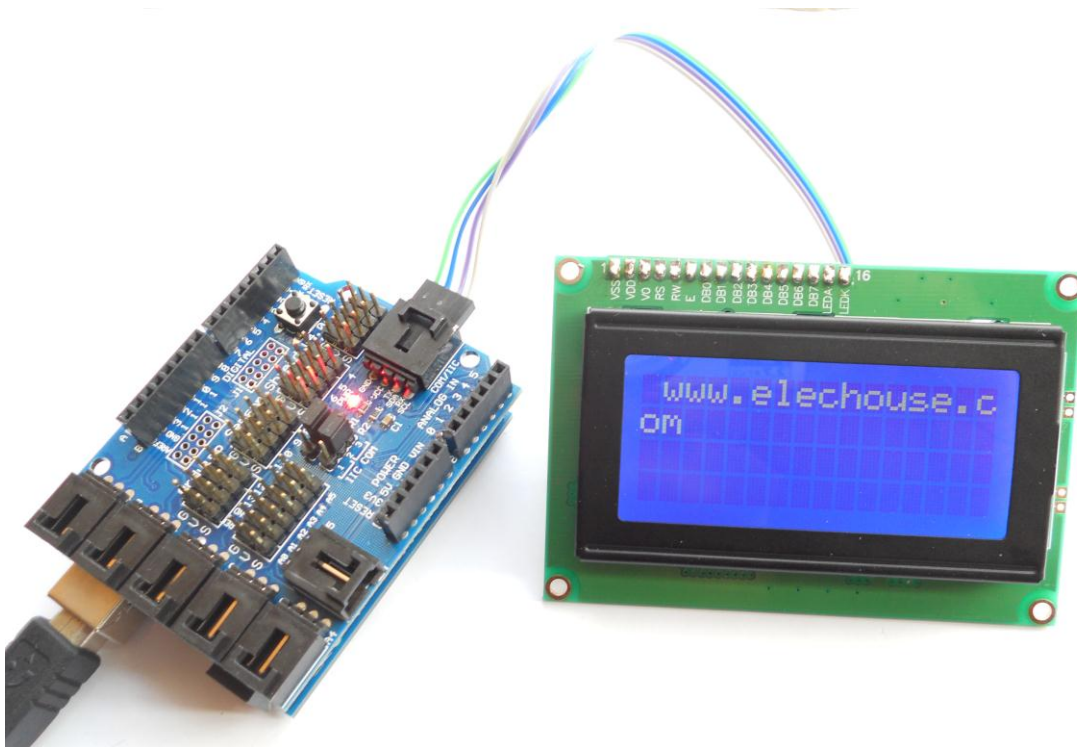
  Serial.print("$go 1 0\n");
  delay(500);

  Serial.print("$go 0 1\n");
  delay(500);

  Serial.print("$go 1 1\n");
  delay(500);

  Serial.print("$go 1 1\n");
  delay(500);
```

```
/** cursor off */  
Serial.print("$cURSor Off\n");  
delay(100);  
  
Serial.print("$go 0 0\n");  
}  
  
void loop(void)  
{  
  /** write character to LCD */  
  Serial.print("$prinT  www.elechouse.com \n");//here $print is OK.  
  Serial.print("$prinT  LCD controller\n");  
  delay(1000);  
  Serial.print("$clear\n");  
  delay(1000);  
}
```



Disclaimer and Revisions

The information in this document may change without notice. Please visit www.elechouse.com or mail to service@elechouse.com for updating.

Revision History