# Post IIC-SPI LCD Library to Arduino 1.0

Support LCD Contrast Adjustment with fine adjustment of a trimpot 10K Ω to any desired contrast setting.

**联系客服 forest——net 取得库文件**

**库已经更新全面支持Arduino系列已更新支持Arduino IDE 1.0。**

同时学习iic和spi两种协议!

**同时支持多种协议：IIC/I2C/TWI/SPI，通过地址设定，支持多个I2C设备。**

**可插拔设计，多用途转接板！支持所有标准16PINLCD，用户可以自行更换，默认带标准1602LCD。**

**有蓝屏和黄屏可供选择，留言说明即可。**

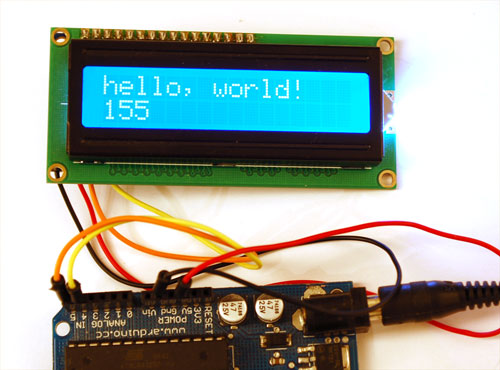
**强悍的小板,不容错过！！**

此为套件包含：

LCD1602字符液晶显示器 一块

IIC/I2C/TWI/SPI LCD扩展小板 一块

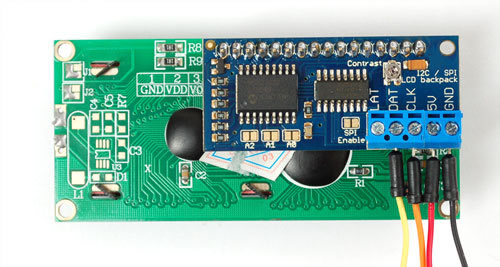
**Arduino IIC/I2C/TWI/SPI接口字符液晶显示器 LCD1602套件**



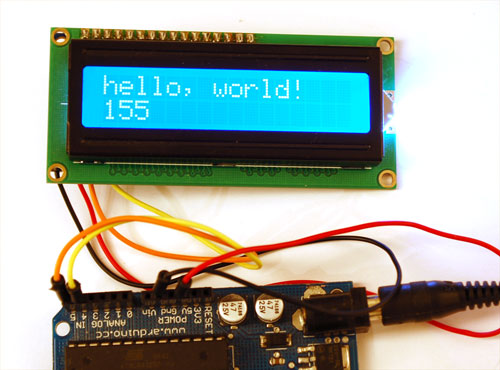
**Arduino 控制器端口实在是有限，加几个传感器、SD卡啥的，就没有端口了，还想接个1602液晶显示，怎么办？我们开发了一款I2C/SPI接口的1602LCD 就解决了上述问题，I2C只需两根线就可以实现数据显示，通过设置相关的位可以同时挂接多个IIC LCD设备。同时还支持SPI协议，爱好者可根据自己的需要进行选择连接方式。**

**购买后请联系店家索取Arduino库文件或者到售后支持网站查看产品文档。**

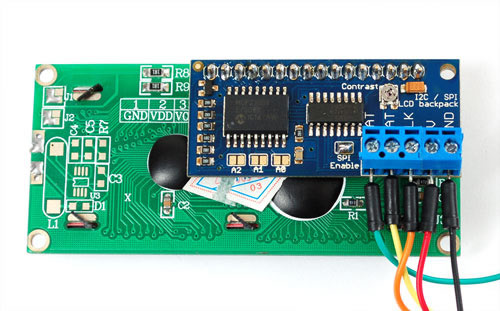
**产品参数：**  
**1.供电电压：+5V  
2.支持IIC/I2C和SPI两种协议**  
**3.具有背光灯，和对比度调节电位器  
4.显示器带IIC和SPI接线端子，可以使用普通导线即可连接，不需要购买专用的转接线，方便爱好者使用。**



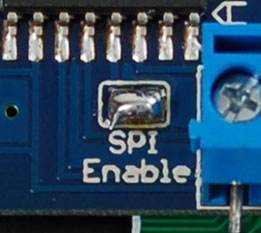
IIC连接方式



IIC连接方式和Arduino连接



SPI连接方式



启用SPI连接方式



SPI连接方式和Arduino连接

特别说明：如果你是使用Arduino Mega系列主控板IIC/I2C协议的话请注意接线和Arduino Duemilanove 2009是不一样的，在ArduinoMega系列上SDA是数字端口20，SCL是数字端口21.详见官方说明：http://www.arduino.cc/en/Reference/Wire

**Forest\_net**

**I2C Port expander and LCDs**

I just bought an I2C module for my LCD. The schematics (from a Hungarian guy who makes it) are here: <http://avr.tavir.hu/images/contents/29a.jpg>

The most important thing is the concept:

* There is a port expander (in this case, an MCP23008 8-bit port) connected to the Arduino with I2C
* The port expander has 8 outputs, using the 4Bit LCD driver, it's possible to use 4 bits for the LCD data, 3 bits for R/W, E, RS pins, and 1 bit for the LCD backlight (using a FET). Brilliant!

The greatest advantage of this is not only does it use just 2 pins, it uses I2C pins (analog in 4 and 5), and more devices can be on that I2C bus.

The only downside of it is that you have to use the Wire library, which takes a bit of space. However, in many cases you might already be using the Wire library, if you have other I2C devices attached to the Arduino.

This code can possibly be adapted to other I2C port expanders, eg. the PCF8574. I think all LCD (4bit) libraries should be merged (normal, shift register), because the different part is only outputting the data bytes, all else is common.

I didn't have any library ready, so I took the 4Bit LCD library (from Arduino forums and this wiki) and modified it.

Here is the (currently not 100% polished, beta quality) code for the library: LCDI2C4Bit.h

|  |
| --- |
| #ifndef LCDI2C4Bit\_h  #define LCDI2C4Bit\_h  #include <inttypes.h>  // IMPORTANT! Wire. must have a begin() before calling init()  class LCDI2C4Bit {  public:  LCDI2C4Bit(int devI2CAddress, int num\_lines, int lcdwidth);  void commandWrite(int command);  void init();  void print(int value);  void printIn(char value[]);  void clear();  void backLight( bool turnOn );  //non-core---------------  void cursorTo(int line\_num, int x);  //void leftScroll(int chars, int delay\_time);  //end of non-core--------  //4bit only, therefore ideally private but may be needed by user  //void commandWriteNibble(int nibble);  private:  //void pulseEnablePin();  //void pushNibble(int nibble);  //void pushByte(int value);  int myNumLines;  int myWidth;  int myAddress;  };  #endif |

LCDI2C4Bit.cpp

|  |
| --- |
| #include "LCDI2C4Bit.h"  #include <Wire.h>  extern "C" {  #include <stdio.h> //not needed yet  #include <string.h> //needed for strlen()  #include <inttypes.h>  #include "WConstants.h" //all things wiring / arduino  }  //command bytes for LCD  #define CMD\_CLR 0x01  #define CMD\_RIGHT 0x1C  #define CMD\_LEFT 0x18  #define CMD\_HOME 0x02  //stuff the library user might call---------------------------------  //constructor. num\_lines must be 1 or 2, currently.  byte dataPlusMask = 0; // TODO!!!  LCDI2C4Bit::LCDI2C4Bit( int devI2CAddress, int num\_lines, int lcdwidth) {  myNumLines = num\_lines;  myWidth = lcdwidth;  myAddress = devI2CAddress;  }  void SetMCPReg( byte deviceAddr, byte reg, byte val ) {  Wire.beginTransmission(deviceAddr);  Wire.send(reg);  Wire.send(val);  Wire.endTransmission();  }  void SendToLCD( byte deviceAddr, byte data ) {  data |= dataPlusMask;  SetMCPReg(deviceAddr,0x0A,data);  data ^= 0x80; // E  delayMicroseconds(1);  SetMCPReg(deviceAddr,0x0A,data);  data ^= 0x80; // E  delayMicroseconds(1);  SetMCPReg(deviceAddr,0x0A,data);  delay(1);  }  void WriteLCDByte( byte deviceAddr, byte bdata ) {  SendToLCD(deviceAddr,bdata >> 4);  SendToLCD(deviceAddr,bdata & 0x0F);  }  void LCDI2C4Bit::init( void ) {  dataPlusMask = 0; // initial: 0  SetMCPReg(myAddress,0x05,0x0C); // set CONFREG (0x05) to 0  SetMCPReg(myAddress,0x00,0x00); // set IOREG (0x00) to 0  //  delay(50);  SendToLCD(myAddress,0x03);  delay(5);  SendToLCD(myAddress,0x03);  delayMicroseconds(100);  SendToLCD(myAddress,0x03);  delay(5);  SendToLCD(myAddress,0x02);  WriteLCDByte(myAddress,0x28);  WriteLCDByte(myAddress,0x08);  WriteLCDByte(myAddress,0x0C); // turn on, cursor off, no blinking  delayMicroseconds(60);  WriteLCDByte(myAddress,0x01); // clear display  delay(3);  }  void LCDI2C4Bit::backLight( bool turnOn ) {  dataPlusMask |= 0x40; // Lights mask  if (!turnOn) dataPlusMask ^= 0x40;  SetMCPReg(myAddress,0x0A,dataPlusMask);  }  void LCDI2C4Bit::print( int value ) {  dataPlusMask |= 0x10; // RS  WriteLCDByte(myAddress,(byte)value);  dataPlusMask ^= 0x10; // RS  }  void LCDI2C4Bit::printIn( char value[] ) {  for ( char \*p = value; \*p != 0; p++ )  print(\*p);  }  void LCDI2C4Bit::clear() {  commandWrite(CMD\_CLR);  }  void LCDI2C4Bit::cursorTo(int line\_num, int x) {  commandWrite(CMD\_HOME);  int targetPos = x + line\_num \* myWidth;  for ( int i = 0; i < targetPos; i++)  commandWrite(0x14);  }  void LCDI2C4Bit::commandWrite( int command ) {  // RS - leave low  WriteLCDByte(myAddress,command);  delay(1);  } |

Using it, an example:

|  |
| --- |
| #include <Wire.h>  #include <LCDI2C4Bit.h>  int ADDR = 0xA7;  /\*  ;Connect the following pins from MCP23008 to LCD  ;  ;P0 - D4  ;P1 - D5  ;P2 - D6  ;P3 - D7  ;P4 - RS  ;P5 - RW (not used, set to 0 to ground for write)  ;P6 - Bl (backlight switch)  ;P7 - E  \*/  byte x = 0;  byte data = 1;  byte c;  LCDI2C4Bit lcd = LCDI2C4Bit(ADDR,4,20);  void setup()  {  Serial.begin(9600);  Wire.begin(); // join i2c bus (address optional for master)  lcd.init();  lcd.printIn("test");  WriteLCD(ADDR,'a');  lcd.clear();  WriteLCD(ADDR,'c');  lcd.cursorTo(0,0);  lcd.printIn("0");  lcd.cursorTo(1,0);  lcd.printIn("1");  lcd.cursorTo(2,0);  lcd.printIn("2");  lcd.cursorTo(3,0);  lcd.printIn("3");  }  void loop()  {  lcd.backLight(true);  delay(1000);  lcd.backLight(false);  delay(1000);  } |