Handson Technology

User Guide

I2C Interface 20x4 LCD Module – Blue Screen

This is I2C interface 20x4 LCD display module, a new high-quality 4 line 20 character LCD module with on-board contrast control adjustment, backlight and I2C communication interface. For Arduino beginners, no more cumbersome and complex LCD driver circuit connection. The real significance advantages of this I2C Serial LCD module will simplify the circuit connection, save some I/O pins on Arduino board, simplified firmware development with widely available Arduino library. This is Negative Transmissive white character with blue background.

SKU: DSP1207

Brief Data:

- Operating Voltage: 5Vdc.
- Display Mode: Blue STN.
- Polarizer Type: White Negative Transmissive.
- 5X8 dots with cursor.
- I2C Address (Default):
  - 0x3F > PCF8574AT.
  - 0X27 > PCF8574T.
- Interface: I2C to 4bits LCD data and control lines.
- Contrast Adjustment: On board Potentiometer.
- Backlight Control: Firmware or jumper wire.
- Board Size: 98x60 mm.

Setting Up:

Hitachi's HD44780 based character LCD are very cheap and widely available, and is an essential part for any project that displays information. Using the LCD piggy-back board, desired data can be displayed on the LCD through the I2C bus. In principle, such backpacks are built around PCF8574 (from NXP) which is a general purpose bidirectional 8 bit I/O port expander that uses the I2C protocol. The PCF8574 is a silicon CMOS circuit provides general purpose remote I/O expansion (an 8-bit quasi-bidirectional) for most microcontroller families via the two-line bidirectional bus (I2C-bus). Note that most piggy-back modules are centered around PCF8574T (SO16 package of PCF8574 in DIP16 package) with a default slave address of 0x27. If your piggy-back board holds a PCF8574AT chip, then the default slave address will change to 0x3F. In short, if the piggy-back board is based on PCF8574T and the address connections (A0-A1-A2) are not bridged with solder it will have the slave address 0x27.

Address selection pads in the I2C-to-LCD piggy-back board.

Table 5. PCF8574A address map

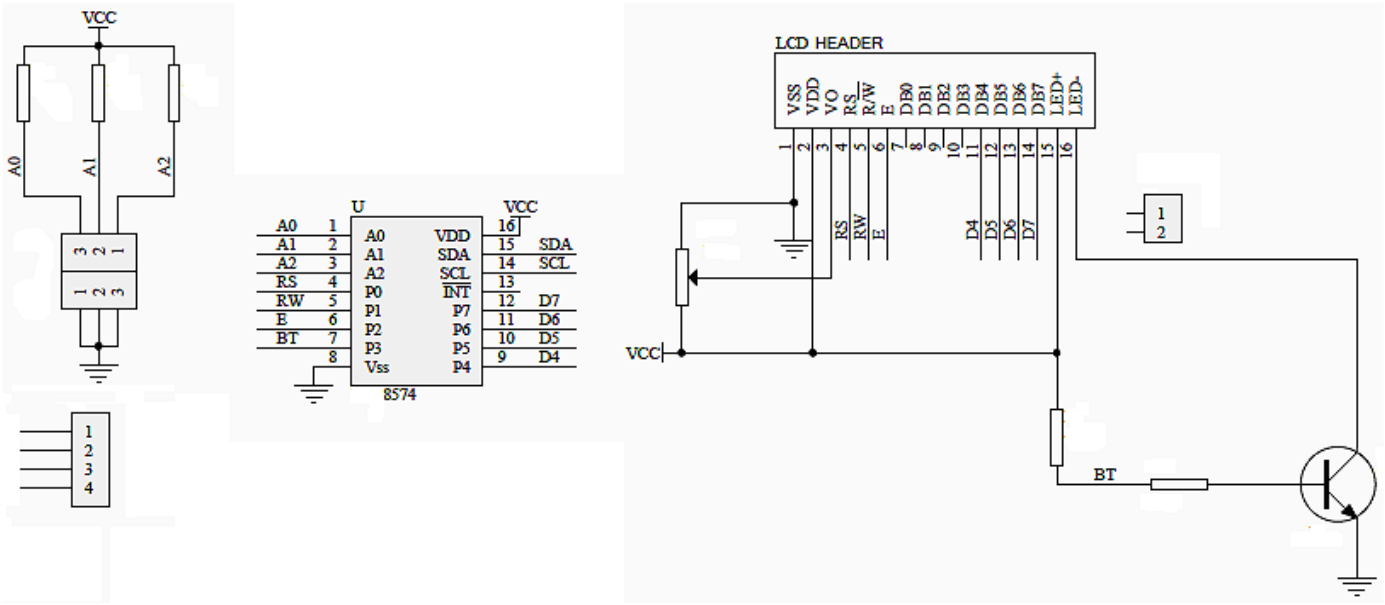
Pin connectivity			Address of PCF8574A								Address byte value		7-bit hexadecimal address without R/W
A2	A1	A0	A6	A5	A4	A3	A2	A1	A0	R/W	Write	Read	
V_{SS}	V_{SS}	V_{SS}	0	1	1	1	0	0	0	-	70h	71h	38h
V_{SS}	V_{SS}	V_{DD}	0	1	1	1	0	0	1	-	72h	73h	39h
V_{SS}	V_{DD}	V_{SS}	0	1	1	1	0	1	0	-	74h	75h	3Ah
V_{SS}	V_{DD}	V_{DD}	0	1	1	1	0	1	1	-	76h	77h	3Bh
V_{DD}	V_{SS}	V_{SS}	0	1	1	1	1	0	0	-	78h	79h	3Ch
V_{DD}	V_{SS}	V_{DD}	0	1	1	1	1	0	1	-	7Ah	7Bh	3Dh
V_{DD}	V_{DD}	V_{SS}	0	1	1	1	1	1	0	-	7Ch	7Dh	3Eh
V_{DD}	V_{DD}	V_{DD}	0	1	1	1	1	1	1	-	7Eh	7Fh	3Fh

Address Setting of PCD8574A (extract from PCF8574A data specs).

Note: When the pad A0~A2 is open, the pin is pull up to VDD. When the pin is solder shorted, it is pull down to VSS.

The default setting of this module is A0~A2 all open, so is pull up to VDD. The address is 3Fh in this case.

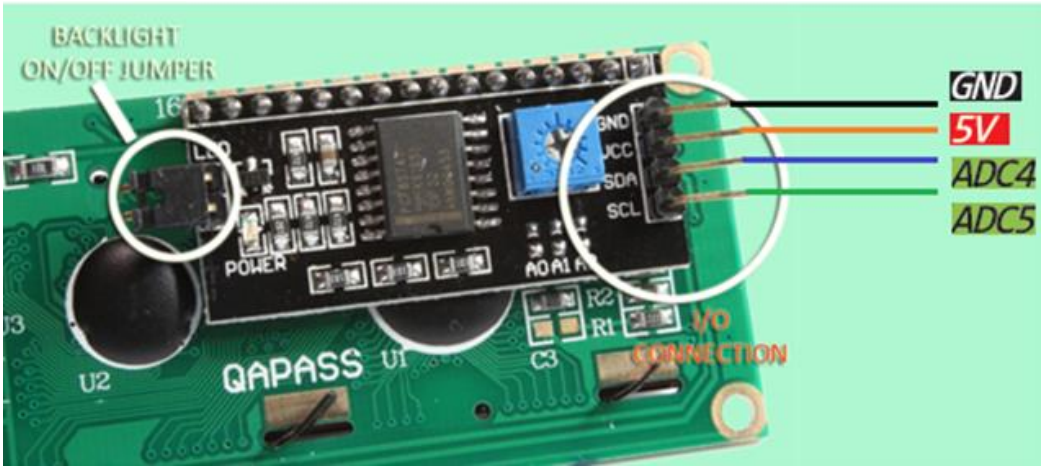
Reference circuit diagram of an Arduino-compatible LCD backpack is shown below. What follows next is information on how to use one of these inexpensive backpacks to interface with a microcontroller in ways it was exactly intended.



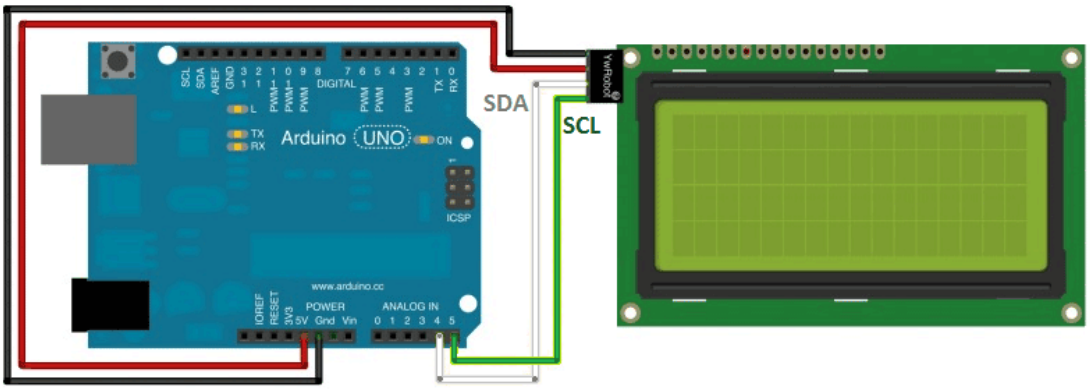
Reference circuit diagram of the I2C-to-LCD piggy-back board.

I2C LCD Display.

At first you need to solder the I2C-to-LCD piggy-back board to the 16-pins LCD module. Ensure that the I2C-to-LCD piggy-back board pins are straight and fit in the LCD module, then solder in the first pin while keeping the I2C-to-LCD piggy-back board in the same plane with the LCD module. Once you have finished the soldering work, get four jumper wires and connect the LCD module to your Arduino as per the instruction given below.



LCD display to Arduino wiring.



Arduino Setup

For this experiment it is necessary to download and install the “Arduino I2C LCD” library. First of all, rename the existing “LiquidCrystal” library folder in your Arduino libraries folder as a backup, and proceed to the rest of the process.

https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads

Next, copy-paste this example sketch Listing-1 for the experiment into the blank code window, verify, and then upload.

Arduino Sketch Listing-1:

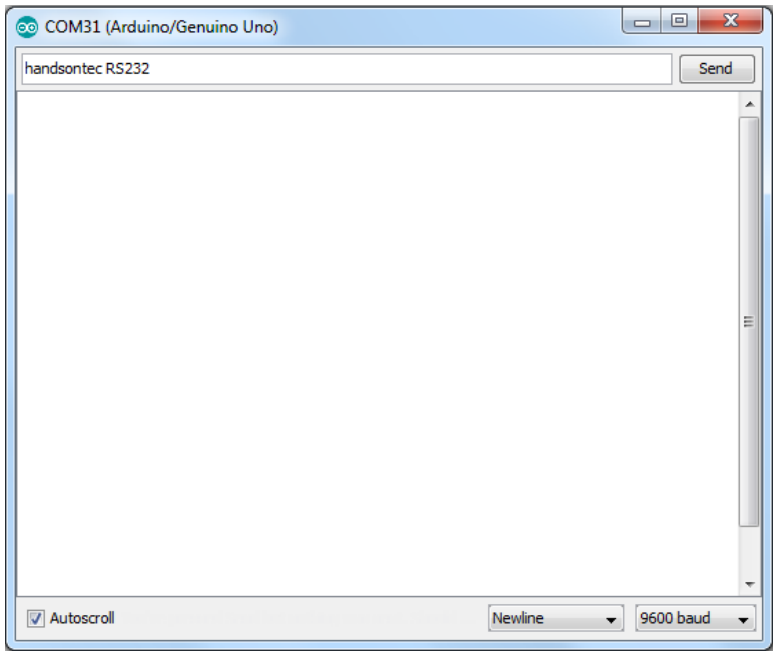
```
/*=========================================================================
//  Author      : Handson Technology
//  Project     : I2C to LCD with Arduino Uno
//  Description : LCD with I2C Interface.
//  LiquidCrystal Library - I2C Serial to LCD
//  Source-Code : I2C LCD.ino
//=========================================================================
*/

/*-----( Import needed libraries )-----*/
#include <Wire.h> // Comes with Arduino IDE
// Get the LCD I2C Library here:
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
// Move any other LCD libraries to another folder or delete them
// See Library "Docs" folder for possible commands etc.

#include <LiquidCrystal_I2C.h>
/*-----( Declare Constants )-----*/
// set the LCD address to 0x3F for PCF8574AT with A0,A1,A0 address line open, default
setting.
// Set the pins on the I2C chip used for LCD connections:
//                    (addr, en,rw,rs,d4,d5,d6,d7,bl,blpol)
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  // Set the LCD I2C
address

/*-----( Declare Variables )-----*/


void setup()   /*----( SETUP: RUNS ONCE )----*/
{
  Serial.begin(9600);  // Used to type in characters

  lcd.begin(20,4);         // initialize the lcd for 20 chars 4 lines, turn on
backlight

// ------- Quick 3 blinks of backlight  -------------
  for(int i = 0; i< 3; i++)
  {
    lcd.backlight();
    delay(250);
    lcd.noBacklight();
    delay(250);
  }
  lcd.backlight(); // finish with backlight on

//-------- Write characters on the display ------------------
  // NOTE: Cursor Position: Lines and Characters start at 0
  lcd.setCursor(3,0); //Start at character 4 on line 0
  lcd.print("Hello, world!");
  delay(1000);
  lcd.setCursor(2,1);
  lcd.print("From Handsontec  ");
```

```
  delay(1000);
  lcd.setCursor(0,2);
  lcd.print("20 by 4 Line Display");
  lcd.setCursor(0,3);
  delay(2000);
  lcd.print(" www.handsontec.com ");
  delay(8000);
  // Wait and then tell user they can start the Serial Monitor and type in characters
to
  // Display. (Set Serial Monitor option to "No Line Ending")
  lcd.setCursor(0,0); //Start at character 0 on line 0
  lcd.print("Start Serial Monitor");
  lcd.setCursor(0,1);
  lcd.print("Type char to display");


}/*--(end setup )---*/


void loop()   /*----( LOOP: RUNS CONSTANTLY )----*/
{
  {
    // when characters arrive over the serial port...
    if (Serial.available()) {
      // wait a bit for the entire message to arrive
      delay(100);
      // clear the screen
      lcd.clear();
      // read all the available characters
      while (Serial.available() > 0) {
        // display each character to the LCD
        lcd.write(Serial.read());
      }
    }
  }

}/* --(end main loop )-- */


/* ( THE END ) */
```

If you are 100% sure that everything is okay, but you don't see any characters on the display, try to adjust the contrast control pot of the backpack and set it a position where the characters are bright and the background does not have dirty boxes behind the characters. Following is a partial view of author's experiment with the above described code with 20x4 display module. Since the display used by the author is a very clear bright "black on yellow" type, it is very difficult to get a good catch due to polarization effects.

This sketch will also display character send from serial Monitor:

In Arduino IDE, go to “Tools” > “Serial Monitor”. Set the correct baud rate at 9600. Type the character on the top empty space and hit “SEND”.



The string of character will be displayed on the LCD module.

Resources:

- Tutorial: Interfacing with Liquid Crystal Display
- LCD Application Manual
- Complete Guide to Arduino LCD Interfacing (PDF)

Handsontec.com

We have the parts for your ideas

HandsOn Technology provides a multimedia and interactive platform for everyone interested in electronics. From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment. Analog and digital, practical and theoretical; software and hardware.

open source hardware

Hands*On* Technology support Open Source Hardware (OSHW) Development Platform.

Learn : Design : Share

handsontec.com

The Face behind our product quality…

In a world of constant change and continuous technological development, a new or replacement product is never far away – and they all need to be tested.
Many vendors simply import and sell wihtout checks and this cannot be the ultimate interests of anyone, particularly the customer. Every part sell on Handsotec is fully tested. So when buying from Handsontec products range, you can be confident you're getting outstanding quality and value.

We keep adding the new parts so that you can get rolling on your next project.

Breakout Boards & Modules

Connectors

Electro-Mechanical Parts

Engineering Material

Mechanical Hardware

Electronics Components

Power Supply

Arduino Board & Shield

Tools & Accessory