



# I2C/TWI LCD1602 Module (Gadgeteer Compatible) (SKU: DFR0063)

---

## Introduction

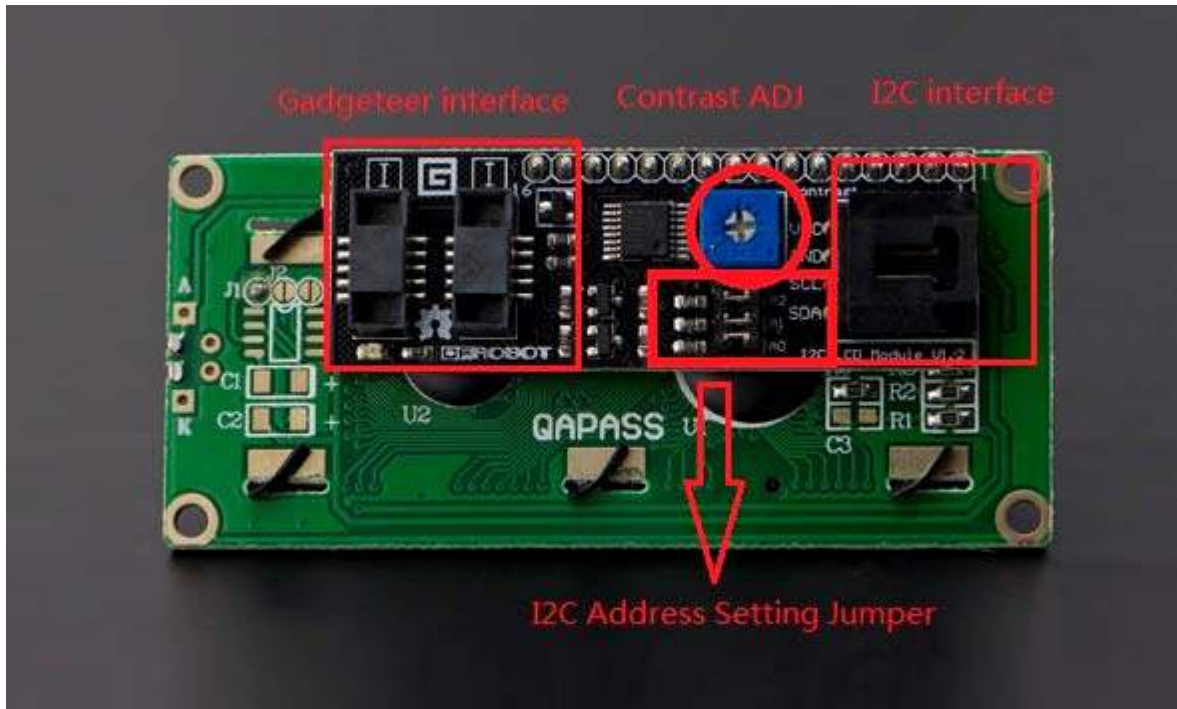
This is another great **I2C 16x2 LCD display** compatible with Gadgeteer modules from DFRobot. With limited pin resources, your project will quickly run out of resources using normal LCDs. With this I2C interface LCD module, you only need 2 lines (I2C) to display the information. If you already have I2C devices in your project, this **LCD module** actually costs no more resources at all. The address can be set from 0x20-0x27. Fantastic for **Arduino** or gadgeteer based projects.

## Specification

- I2C Address: 0x20-0x27 (0x20 default)
- Back lit (Blue with white char color)
- Supply voltage: 5V
- Interface: I2C/TWI x1, Gadgeteer interface x2
- Adjustable contrast
- Size: 82x35x18 mm

# Contrast Adjust

The contrast can be adjusted by the potential-meter displayed in the following picture.



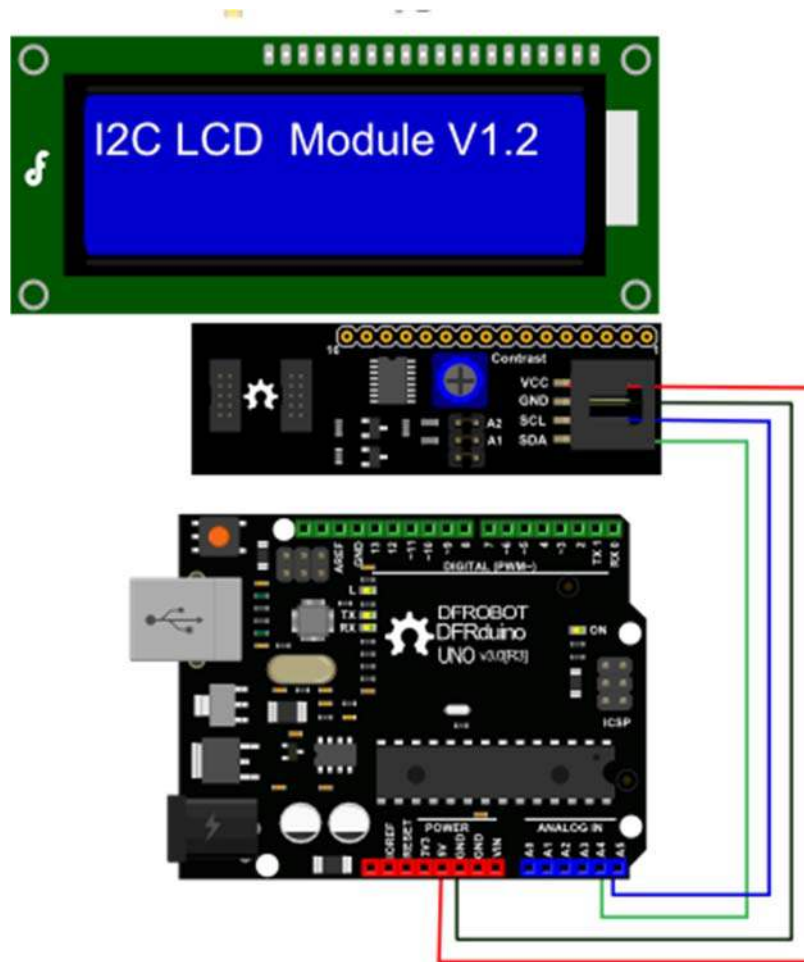
# Address Setting

A2	A1	A0	IIC Address
0	0	0	0x20
0	0	1	0x21
0	1	0	0x22
0	1	1	0x23
1	0	0	0x24
1	0	1	0x25
1	1	0	0x26
1	1	1	0x27

- 0: The Jumper Cap is connected
- 1: The Jumper Cap is disconnected

NOTE: The default address is 0x20. All the jumper caps will be connected from the factory.

## Connection



NOTE: V1.2 has a different power pinout from V1.1, please check the history version for the old connection diagram.

Arduino UNO: connect SDA to pin A4 and SCL to pin A5 on your Arduino.

Arduino Leonardo: connect SDA to digital pin 2 and SCL to digital pin 3 on your Arduino.

# Library Support Functions

- **LiquidCrystal\_I2C()** //set the LCD address for a 16 chars and 2 line display
- **init()** //Initialization for the LCD
- **clear()** //clear display, set cursor position to zero
- **home()** //set cursor position to zero
- **createChar()** //Fill the first 8 CGRAM locations with custom characters
- **setCursor()** //set the position of the cursor
- **cursor()** //Turns the underline cursor on
- **noCursor()** //Turns the underline cursor off
- **blink()** //Turn on the blinking cursor
- **noBlink()** //Turn off the blinking cursor
- **display()** //Turn the display on(quickly)
- **noDisplay()** //Turn the display Off(quickly)
- **backlight()** //Turn the backlight on
- **noBacklight()** //Turn the backlight off
- **scrollDisplayLeft()** //Make the display scroll left without changing the RAM
- **scrollDisplayRight()** //Make the display scroll right without changing the RAM
- **autoscroll()** //This will 'right justify' text from the cursor
- **noAutoscroll()** //This will 'left justify' text from the cursor
- **leftToRight()** //This is for text that flows Left to Right
- **rightToLeft()** //This is for text that flows Right to Left

## Sample Code

[Download Sample code and library](#)

```
//DFRobot.com
//Compatible with the Arduino IDE 1.0
//Library version:1.1
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x20,16,2); // set the LCD address to 0x20 for a
16 chars and 2 line display

void setup()
{
    lcd.init(); // initialize the lcd
```

```

    // Print a message to the LCD.
    lcd.backlight ();
    lcd.print ("Hello, world!");
}

void loop ()
{
}

```

Sample sketch: Control the back light of the I2C LCD1602 Module

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif

LiquidCrystal_I2C lcd(0x20,16,2); // set the LCD address to 0x20 for a
16 chars and 2 line display

void setup () {

    lcd.init (); // initialize the lcd
    lcd.backlight ();

    lcd.home ();

    lcd.print ("Hello world...");
    lcd.setCursor (0, 1);
    lcd.print ("dfrobot.com");
}

```

```

}

int backlightState = LOW;
long previousMillis = 0;
long interval = 1000;

void loop() {

    unsigned long currentMillis = millis();

    if(currentMillis - previousMillis > interval) {
        previousMillis = currentMillis;

        if (backlightState == LOW)
            backlightState = HIGH;
        else
            backlightState = LOW;

        if(backlightState == HIGH) lcd.backlight();
        else lcd.noBacklight();
    }
}

```

If you want to use the library's own sample code, pay attention to modify the initialization statement, need to change:  
LiquidCrystal\_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display  
to:  
LiquidCrystal\_I2C lcd(0x20,16,2); // set the LCD address to 0x20 for a 16 chars and 2 line display(All jumpers should be connected!)  
Because the default initialization statement is for LCD1602!