

Afficheur LCD 2x16 I2C DFR0063

I2C/TWI LCD1602 Module (SKU: TOY0046)

Introduction

I2C/TWI LCD1602 Module (SKU: TOY0046)

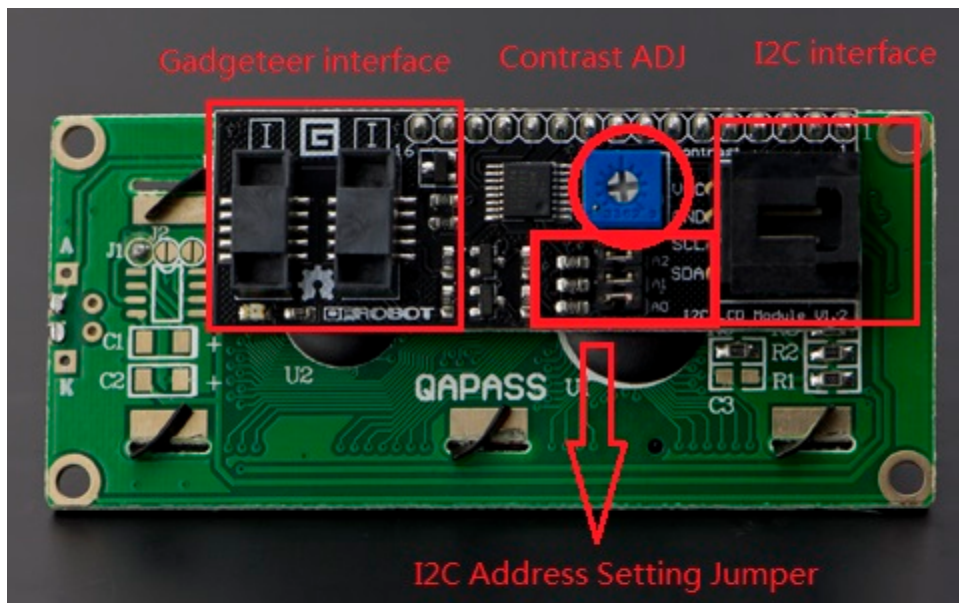
This is another great LCD display compatible with Gadgeteer modules from DFRobot. With limited pin resources, your project will quickly run out of resources using normal LCDs. With this I2C interface LCD module, you only need 2 lines (I2C) to display the information. If you already have I2C devices in 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

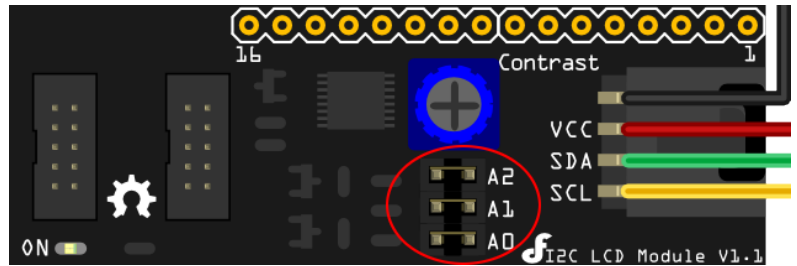
Contrast Adjust

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



Address Setting

A2	A1	A0	I2C (hex)
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	1	0x26
1	1	1	0x27

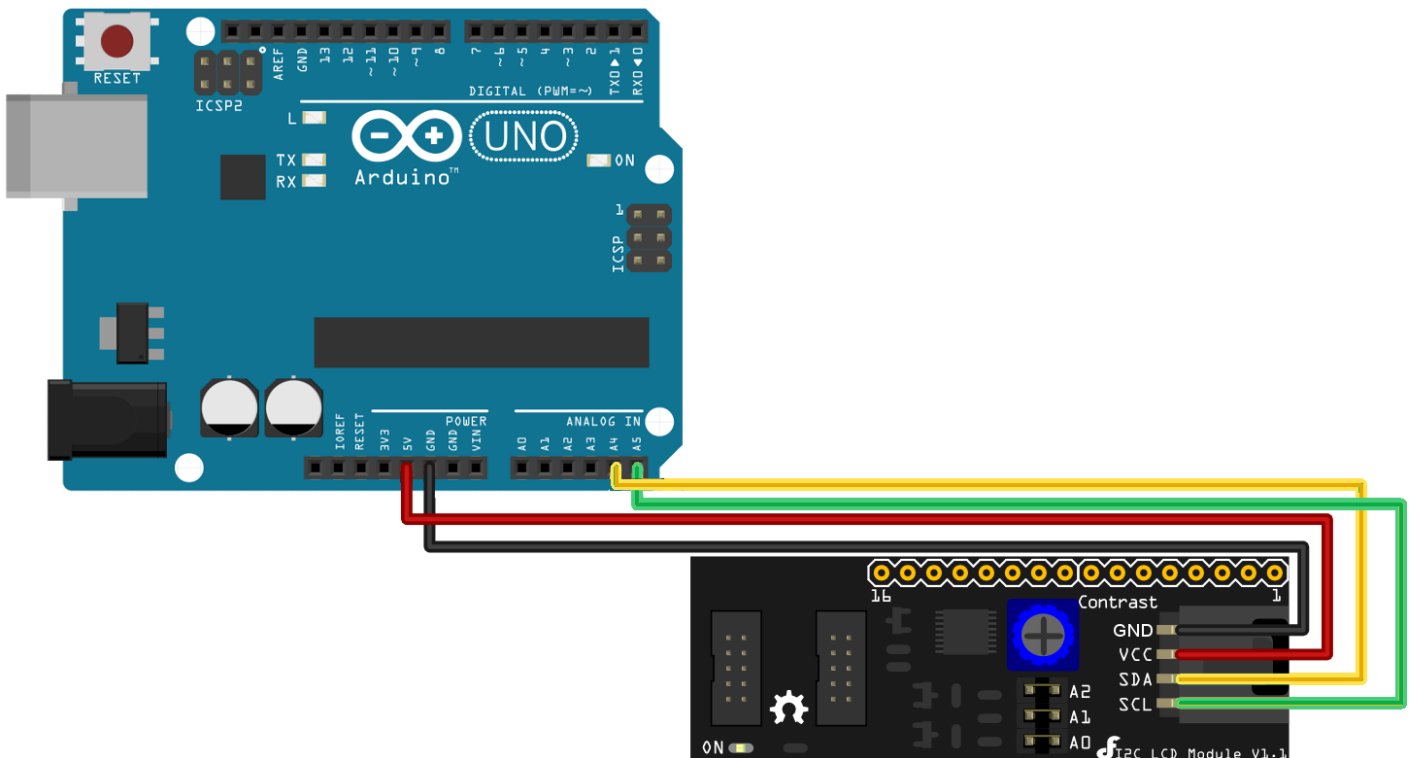


0: The Jumper Cap is connected

1: The Jumper Cap is disconnected

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

Connection Diagram



- Arduino UNO: connect SDA to Analog pin 5 and SCL to Analog pin 4 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
- **print()** // Prints text or a data to the LCD
- **write()** ; // Write a character to the LCD
- **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

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

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

void setup()
{
  lcd.begin(); // initialize the LCD

  // Turn on the backlight and print a message.
  lcd.backlight();
  lcd.print("Hello, world!");
  lcd.setCursor(0,1); // 2nd line
  lcd.print("STI 2D");
}

void loop()
{
  // Do nothing here...
}
```

Sample sketch: display data

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

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

float temperature; // the data to display

void setup()
{
  lcd.begin(); // initialize the LCD

  // Turn on the backlight and print a message.
  lcd.backlight();
  lcd.print("Hello, world!");
  lcd.setCursor(0,1); // 2nd line first character
  lcd.print("STI 2D");
}

void loop()
{
  delay(2000);
  temperature=22.7;
  lcd.clear();
  lcd.setCursor(0,0); // first line
  lcd.print("temerature:");
  lcd.setCursor(0,1); // 2nd line
  lcd.print(temperature); // display the data
}
```

Sample sketch: custom caracters

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

uint8_t bell[8] = {0x4, 0xe, 0xe, 0xe, 0x1f, 0x0, 0x4};
uint8_t note[8] = {0x2, 0x3, 0x2, 0xe, 0x1e, 0xc, 0x0};
uint8_t clock[8] = {0x0, 0xe, 0x15, 0x17, 0x11, 0xe, 0x0};
uint8_t heart[8] = {0x0, 0xa, 0x1f, 0x1f, 0xe, 0x4, 0x0};
uint8_t duck[8] = {0x0, 0xc, 0x1d, 0xf, 0xf, 0x6, 0x0};
uint8_t check[8] = {0x0, 0x1, 0x3, 0x16, 0x1c, 0x8, 0x0};
uint8_t cross[8] = {0x0, 0x1b, 0xe, 0x4, 0xe, 0x1b, 0x0};
uint8_t retarrow[8] = { 0x1, 0x1, 0x5, 0x9, 0x1f, 0x8, 0x4};

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

void setup()
{
  lcd.begin();
  lcd.backlight();
}
```

```

    lcd.createChar(0, bell);
    lcd.createChar(1, note);
    lcd.createChar(2, clock);
    lcd.createChar(3, heart);
    lcd.createChar(4, duck);
    lcd.createChar(5, check);
    lcd.createChar(6, cross);
    lcd.createChar(7, retarrow);
    lcd.home();

    lcd.print("Hello world...");
    lcd.setCursor(0, 1);
    lcd.print(" i ");
    lcd.write(3);
    lcd.print(" arduinos!");
    delay(5000);
    displayKeyCodes();
}

// display all keycodes
void displayKeyCodes(void) {
    uint8_t i = 0;

    while (1) {
        lcd.clear();
        lcd.print("Codes 0x");
        lcd.print(i, HEX);
        lcd.print("-0x");
        lcd.print(i + 16, HEX);
        lcd.setCursor(0, 1);

        for (int j = 0; j < 16; j++) {
            lcd.write(i + j);
        }
        i += 16;

        delay(4000);
    }
}

void loop()
{
    // Do nothing here...
}

```

Sample sketch: blinking cursor

```

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

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

```

```
void setup()
{
    lcd.begin(); // initialize the LCD
}

void loop()
{
    bool blinking = true;
    lcd.cursor();

    while (1) {
        if (blinking) {
            lcd.clear();
            lcd.print("No cursor blink");
            lcd.noBlink();
            blinking = false;
        } else {
            lcd.clear();
            lcd.print("Cursor blink");
            lcd.blink();
            blinking = true;
        }
        delay(4000);
    }
}
```