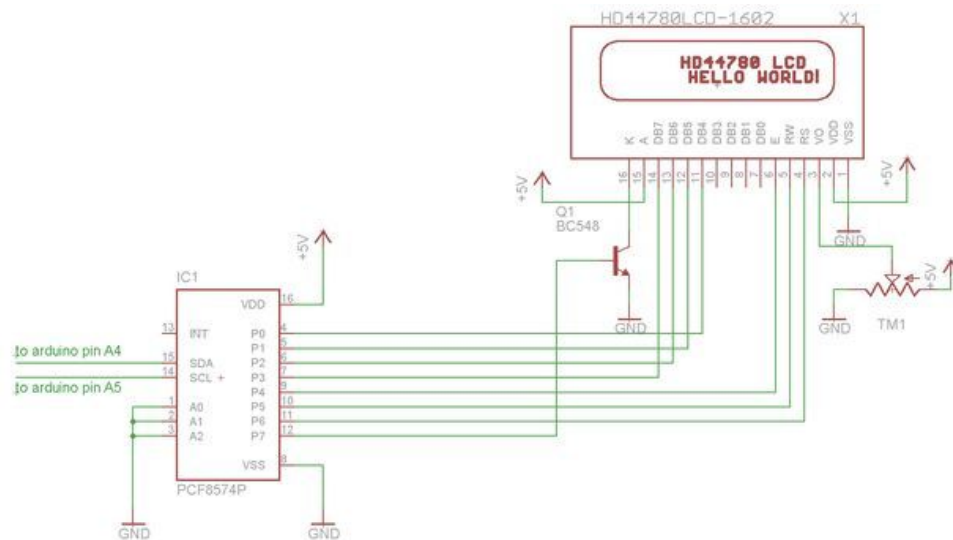


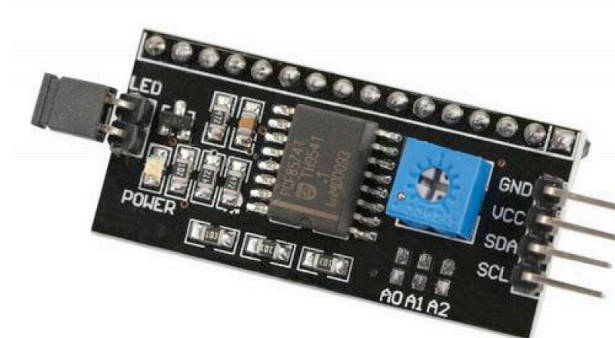
Using the I2C LCD adapter

The serial I2C LCD display adapter converts parallel based 16 x 2 character LCD display into a serial i2C LCD that can be controlled through just 2 wires.

The adapter uses the PCF8574 chip that serves as I/O expander that communicates with Arduino or any other microcontroller by using I2C protocol.

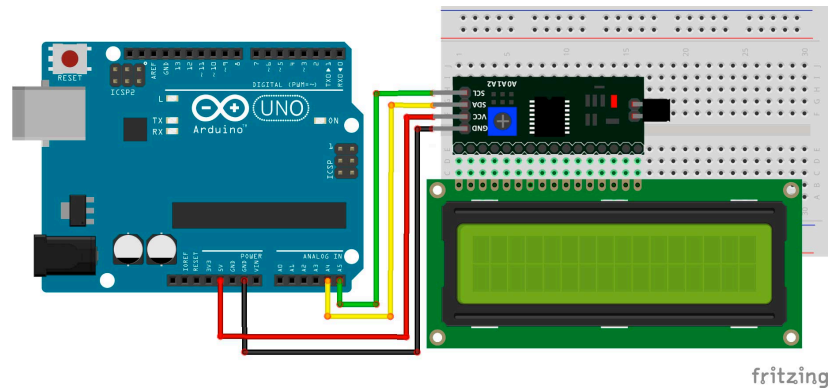


A total of 8 LCD displays can be connected to the same two wire I2C bus with each board having a different address. The default I2C address is **0X27** and may be changed to any of the following 0X20~0X27 via soldering A0 A1 A2 pins.



Connecting the I2C adapter

To connect the I2C adapter to the LCD display it is just only necessary to follow the next schema that interconnects each LCD pin with the corresponding in the adapter.



The pins labelled as VCC and GND must be also wired to the power supply present in the Arduino board.

The SDA and SCL pins, which are the I2C data and clock lines, will be wired to the Arduino pins A4 and A5. These pins are not only for analog inputs but also they are dedicated to the I2C bus.

Accessing to the I2C-LCD display

There are many libraries in Internet to use the I2C-LCD display. These libraries send the correct codes to the LCD display trough the I2C bus.

There is an example using the [LiquidCrystal_I2C.h](#) library that you can download from the Virtual Campus.

```
#include <Wire.h>           // I2C access
#include <LiquidCrystal_I2C.h> // LCD library

// LCD 16x2 on I2C 0x27 address
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

void setup()
{
    Serial.begin(9600);

    lcd.begin(16, 2); // Columns & Rows
    lcd.home ();     // Set (X,Y)=(0,0)

    lcd.print("Hello World");
    lcd.setCursor ( 0, 1 );
    lcd.print(" ARDUINO");

    delay(500);
}

void loop()
{
    int i;

    // Scrool right
    for ( int i = 0; i < 5; i++ ) {
        lcd.scrollDisplayRight();
        delay (500);
    }

    // Scroll left
    for ( int i = 0; i < 5; i++ ) {
        lcd.scrollDisplayLeft();
        delay (500);
    }
}
```