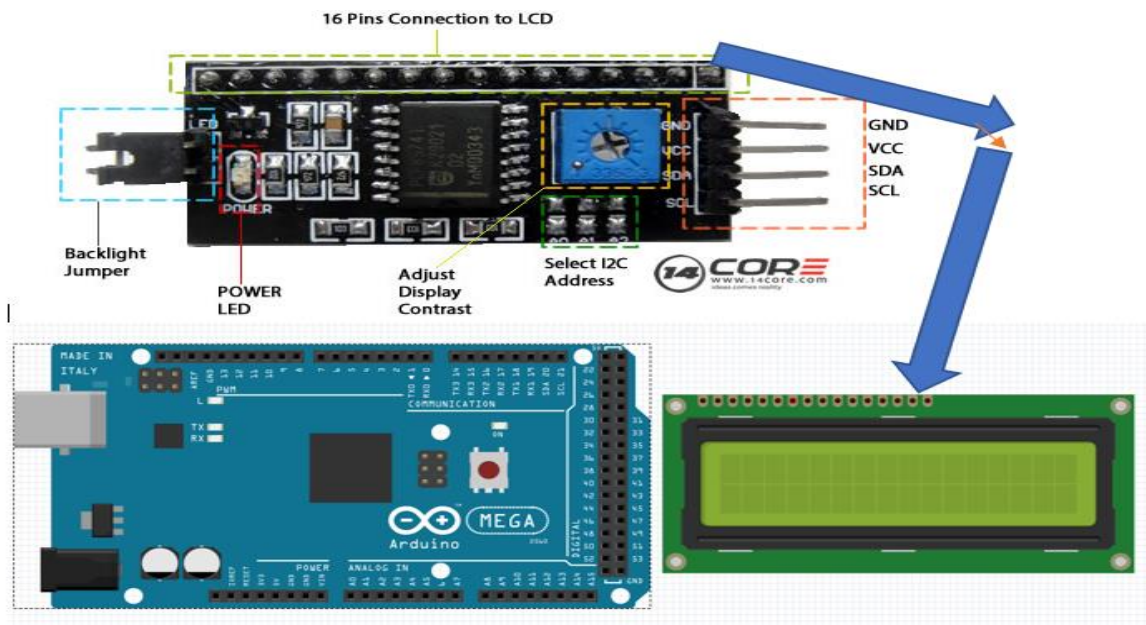


```

/* This code is for LCD(16x2) I2C with serial Interface
 * This code is Modify by Asis Nath aknath@uci.edu
 * PCF8574A Serial Interface
 * The source code for the library can be found at
 * https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-
library/blob/master/LiquidCrystal_I2C.h
 * https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library
 * This code used in the PCF8574A Serial Interface.
 * .....
 * Wiring
 * PCF8574A pins looked from the top side
 *
 * ---
 * |GND|
 * |VCC|
 * |SDA|
 * |SCL|
 * ---
 * GND - GND
 * VCC - 5V
 * SDA - A4 // this is only for Arduino UNO
 * SCL - A5 // this is only for Arduino UNO
 * Arduino Mega SDA and SCL connected with PCF8574A's SDA and SCL
 * Blue tap is the potentiometer to adjust brightness.
 * The other pins goes exactly the same as the LCD pin on the breadboard.
 * .....
 * Datasheet
 *
http://pdf.datasheetcatalog.com/datasheet/NXP\_Semiconductors/PCF8574\_PCF8574A
.pdf
 *
 */

```

Wiring diagram



```

// -----
// i2c_LCD_scanner to determine the address which is needed in the LCD
code
// Asis Nath
//
https://playground.arduino.cc/Main/I2cScanner?action=sourceblock&num=1
// This sketch tests the standard 7-bit addresses
// Devices with higher bit address might not be seen properly.
//
#include <Wire.h>
void setup()
{
  Wire.begin();
  Serial.begin(9600);
  while (!Serial);          // wait for serial monitor
  Serial.println("\nI2C Scanner");
}
void loop()
{
  byte error, address;
  int nDevices;
  Serial.println("Scanning...");
  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {
    // The i2c_scanner uses the return value of
    // the Wire.endTransmission to see if
    // a device did acknowledge to the address.
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println(" !");

      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknown error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");
  delay(5000);          // wait 5 seconds for next scan
}
//end

```

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// Set the LCD address to 0x3F for a 16 chars and 2 line display
// you can get the address from scanner code
LiquidCrystal_I2C lcd(0x3F, 16, 2);
int interval = 1000;
unsigned long time_now = 0;
void setup()
{
    // initialize the LCD
    lcd.init();
    lcd.backlight();
}
void loop()
{
    lcd.setCursor(0,0);
    // it will start from the 1st row and 1st column
    /* here is the 16X2 pixcle setcursor possition works
    * 0,0 1,0 2,0 3,0 4,0 5,0 6,0 7,0 8,0 9,0 10,0 11,0 12,0 13,0 14,0 15,0
    * 0,1 1,1 2,1 3,1 4,1 5,1 6,1 7,1 8,1 9,1 10,1 11,1 12,1 13,1 14,1 15,1
    * note that even though it can be control every pixcle possition, but
    * make sure it does not overlap      * the data
    * It is only set the starting point. Make sure you have enough pixcle
    * possition for 1st one before
    * setting the 2nd pixcle set.
    * One way to slove this issue to use autoscroll() funtion, but you may
    * have to use delay instate of
    * millis().
    *
    */

    /* for (int i = 0; i <10; i++)
    * // just a loop so it shift and display the data
    // it is good when you have more than 16 data in an array
    //{
    lcd.print("Hello, world!");
    {
    */
    lcd.autoscroll();
    // get the time
    if (millis() > time_now + interval)
    {
        for (int i = 0; i <10; i++){
            lcd.print("Hello, world!");
        }
        // Use the snapshot to set track time until next event
        time_now= millis();
    }
    //lcd.setCursor(0,1);
    // it will start from the 2st row and 1st column
    // lcd.autoscroll();
    //lcd.print("Hello, world!");
    //delay (1000);
    //lcd.noAutoscroll();
    }
    //end

```

