

I²C Part 2

11-25-2015 Embedded Workshop



Set up for Workshop:

While you are waiting for the Workshop to begin...

1. Make sure you are connected to the local Wifi
Guest Password:
2. **Download and Install the current version**
Arduino IDE (Version 1.6.4)
Add the Wire.h library
Sketch - Include Libraries - Manage Libraries -
"Wire" and "LiquidCrystal I2C" Author Brabander
3. **Copy Programs from Microrusty.com.**
Project6, Extra Project 7-8
[Project 1](#) [Project 5 Master](#) [Project 5 Slave](#)
[Project 6-a](#) [Program 6-b](#) [Program 7](#)
[Program 8](#)

Parts Needed: (Ask about Parts kits)

- 2 Arduino Uno, Breadboard & Wires,
- 1 - Common Cathode RGB LED , 1 – Any Color LED
- 4 - 330 ohm resistors
- 1- TC74 Temp sensor
- 1- 1602 LCD with the I2C board attached.

Come join us as we explore using an Arduino Uno to connect some two wire (I2C) devices.

I2C is a common protocol used to enable two way communication between devices.

I2C bus can be very confusing for the newcomer. Hopefully we can try to demystify the I2C bus.

If time permits will use a logic analyzer to watch the I2C Master Slave protocol in real time.

I²C Overview

The devices on the I2C bus are either masters or slaves.

Master This is the device that generates clock, starts communication, sends I2C commands and stops communication

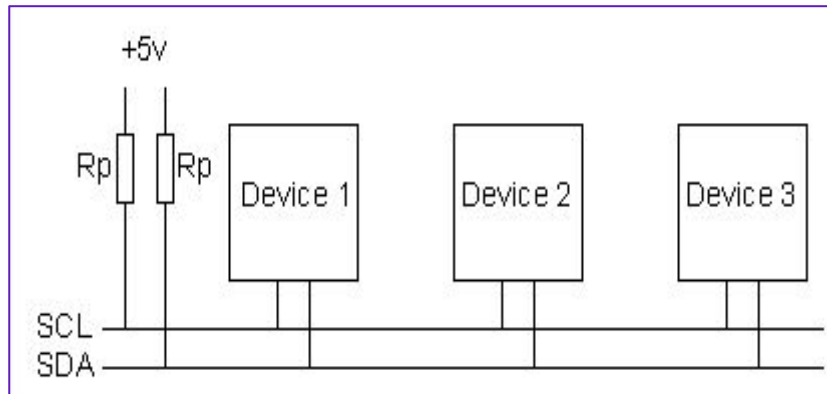
Slave This is the device that listens to the bus and is addressed by the master

Multi-master I2C can have more than one master and each can send commands

Arduino already has pull-up resistors on the SDA and SCL pins. All though this would not hurt the circuit it's not needed because when the Wire.h library is initialized it knows pins 4 and 5 are going to be used for I²C so it also activates the built-in pull-up resistors.

Bus Example: Refer to **Data sheet** for Resistor values

The I²C bus' master uses a device's 7-bit address to signal the component it wants to talk to and is shifted left to create an 8-bit value



I²C Bus & Addressing

I²C bus uses a 7-bit address that is passed along with a read/write bit. Since a byte comprises eight bits.

The extra bit is used to indicate whether the signal is being sent by the master to the slave as a 'write' or a 'read'.

Because of the I²C addresses being 7-bit numbers enables the bus to comprise up to 127 devices.

The 7-bit address is placed in bits one through seven.

The Arduino takes care of the last R/W bit for us depending on what function we're using so as long as you're using the standard Arduino Wire library.

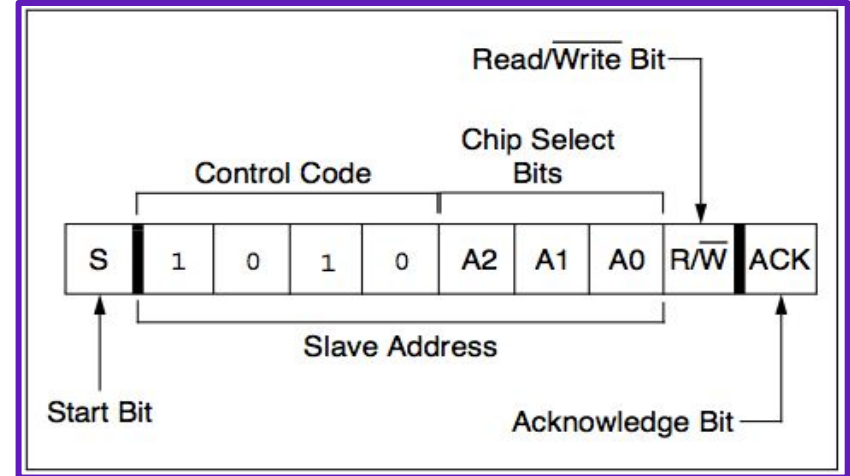
The Seven middle address bits.

The first four bits(Control Code) are hard-wired and can't be changed.

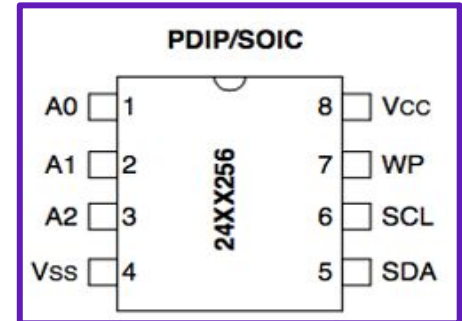
The next three bits(A2,A1,A0) are the important bits and we can change them.

What address the chip will have is dependent on what we set these pins to.

Tie pins 1,2 and 3 on the 24LC256 to GND then the chip would have address 0x50 and if tie them all Vcc then the chip would have address 0x57 .



A0	=	1001	000
A1	=	1001	001
A2	=	1001	010
A3	=	1001	011
A4	=	1001	100
A5	=	1001	101
A6	=	1001	110
A7	=	1001	111



I²C & Arduino

In order to use the I²C interface we need to include the Arduino standard Wire library `#include <Wire.h>`

Need to define the address for your device

```
#define disk1 0x50 //Address of 24LC256
```

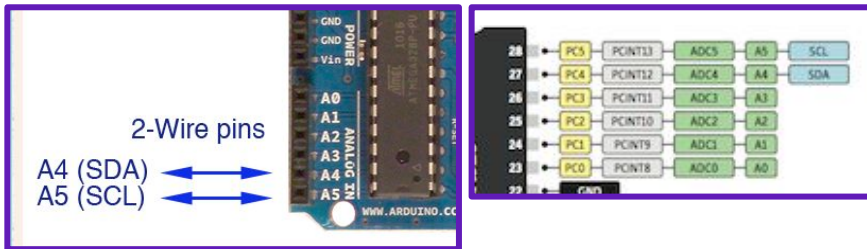
```
#define Temp 0x73 //Address of TC74A0
```

```
int address = 72; //Decimal Address of TC74A2
```

Note: Arduino versions before 1.0 use **Wire.send** and **Wire.receive**
If you are using Arduino 1.0 and above then you need to use **Wire.write** and **Wire.read**

Analog port 4 (A4) = SDA (serial data)

Analog port 5 (A5) = SCL (serial clock)



```
void setup() {  
  Serial.begin(9600);  
  // Define the LED pin as Output  
  pinMode (RLED, OUTPUT);  
  pinMode (GLED, OUTPUT);  
  pinMode (BLED, OUTPUT);  
  // Start the I2C Bus as Slave on address 9  
  Wire.begin(9);  
  // Attach a function to trigger when something is received.  
  Wire.onReceive(receiveEvent);  
}  
  
void receiveEvent(int bytes ) {  
  // Serial.println(x);  
  x = Wire.read();    // read one character from the I2C  
}  
  
void loop() {  
  //If value received is 0 blink LED for 200 ms  
  if (x == 0) {  
    Serial.print ("Red ");  
    Serial.println(x);  
    digitalWrite(RLED, HIGH);  
    delay(100);  
  }  
}
```

I²C Projects

I²C Part 1

Project 1 I²C Scanner Tool.

Project 2 Read TC74 Temp and Write to Console

Project 3 Write to & Read from eeprom

Project 4 Read TC74 Temp and Write to EEPROM

I²C Part 2

Project 5 Connect 2 Arduino Uno together using I²C

Project 6 Read TC74 Temp and Write to I2C/TWI
1602 Serial LCD Module Display 16X2

Need complete 1602 LCD with the I2C board attached.

Project 7 Using I2C PCF8574A with an LCD display connected to an Arduino

Project 8 Using I2C PCF8574A with an LCD display connected to an Arduino

Let's get started and Build the projects.

Project 5

For those who do not have 2 Arduinos then please team up with someone.

Connect the RGB LED to the slave using 330 ohm resistors.

Connect the two Arduino's using SDA & SCL pins.

Be Careful not to Connect 5 volts on both boards

Set up your breadboard with your I²C devices
Connect your breadboard to the Arduino.



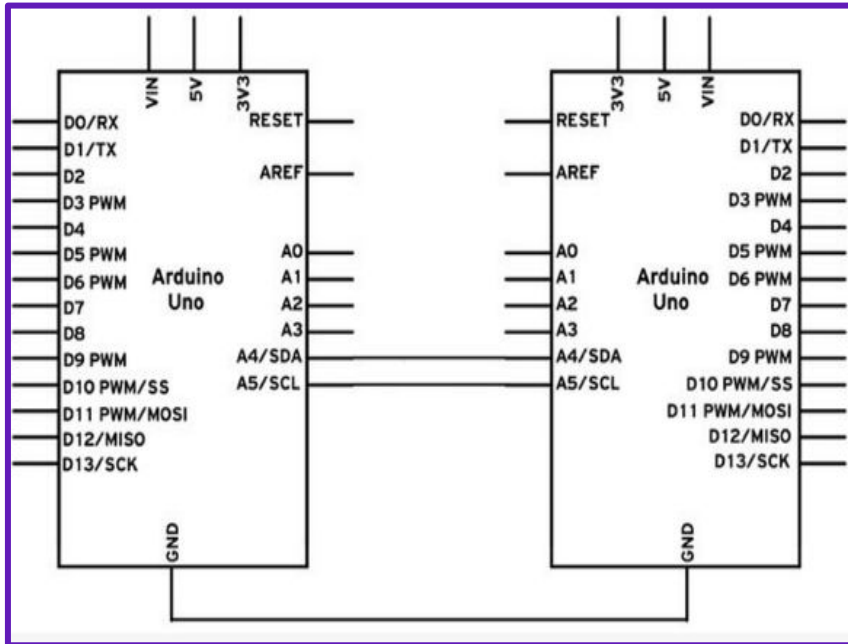
Project 5 Communication Between 2 Arduinos Using I²C

Arduino Board I²C pins

Uno, Pro Mini A4 (SDA), A5 (SCL)

Mega, Due 20 (SDA), 21 (SCL)

Leonardo, Yun 2 (SDA), 3 (SCL)



```
// Include the required Wire library for I2C
// Arduino Master code sends characters to the Arduino Slave address
#include <Wire.h>
int x = 0;
void setup() {
  // Start the I2C Bus as Master
  Wire.begin();
  Serial.begin(9600);
}
void loop() {
  Wire.beginTransmission(9); // transmit to device #9
  Wire.write(x);             // sends x
  Wire.endTransmission();   // stop transmitting
  Serial.println(x);
  x++; // Increment x
  if (x > 5) x = 0; // `reset x once it gets 6
  delay(500);
}
```

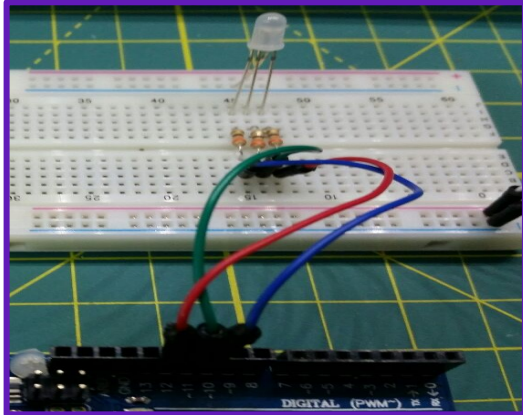

Project 5 Communication Between 2 Arduinos Using I²C

Connect SDA Master to SDA Slave
Connect SCL Master to SCL Slave
Load: arduino_Master-I2C_RGB into Master
Load: arduino_slave-I2C_RGB_9 into Slave
Connect RGB LED to Slave using pins 10, 11,12

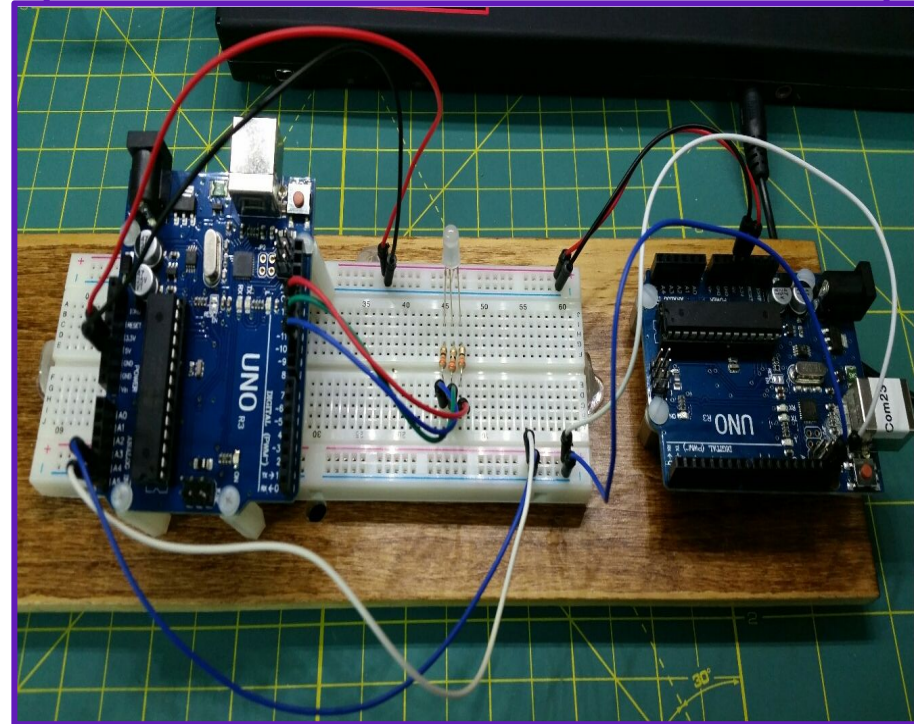
Master sends numbers 0 - 5 to the slave.
The slave changes the color of the LED

Red	0
Green	1
Blue	2
Purple	3
Turquoise	4
Yellow	5

*Both devices use
Serial Monitor
for output



If the programs are not working correctly. Check the
slave address and Run the scanner program.



Project 6-a I2C/TWI 1602 Serial LCD Module Display

PCF8574AN I2C Address:

Board 0x20~0x27

Chip 0x38 - 0x3F

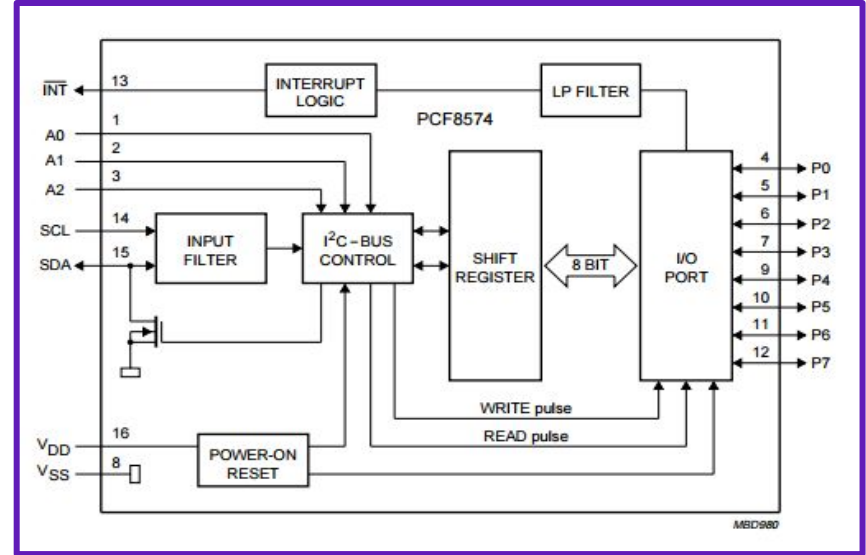
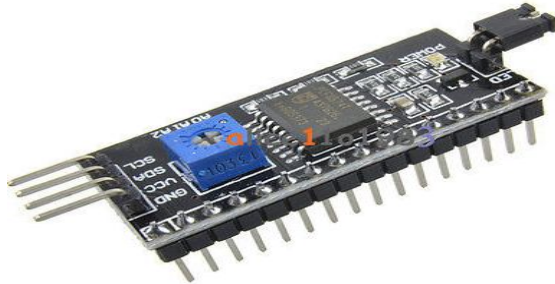
Backlight and contrast is adjusted by potentiometer

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

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

DW OR N PACKAGE
(TOP VIEW)

A0	1	16	V _{CC}
A1	2	15	SDA
A2	3	14	SCL
P0	4	13	INT
P1	5	12	P7
P2	6	11	P6
P3	7	10	P5
GND	8	9	P4



Project 6-b Read TC74 Temp and Write to I2C 1602 LCD Display

16X2 Display

Load: read_temp_Fan_LCD20_v6 -

```
LiquidCrystal_I2C myDisplay(0x27,16,2);
```

20X4 Display

Load: read_temp_Fan_LCD20_v7

```
LiquidCrystal_I2C myDisplay(0x27,20,4);
```

Make sure you have the correct display selected and comment out other display

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C myDisplay(0x27,16,2); // 16X2 LCD
//LiquidCrystal_I2C myDisplay(0x27,20,4); // 20X4 LCD
int temp_address = 72; //Check address with scanner
int f;
int c;
int reset;
void setup()
{
  myDisplay.init(); // initialize
  myDisplay.backlight(); // Turn on backlight
  Serial.begin(9600); // Create Serial Object (9600 Baud)
  Wire.begin(); // Create a Wire Object
  myDisplay.setCursor(0,0); // Set cursor |
  reset =0;
  delay (4000);
  initialization();
  SensorDisplay();
}
```

TC74 Pinout

Arduino analog pin 4 to TC74 pin 2

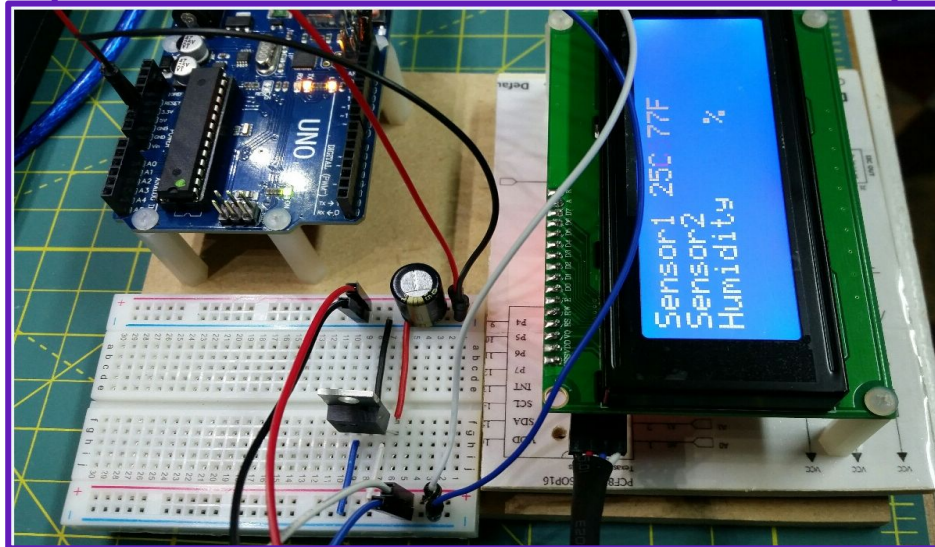
Arduino analog pin 5 to TC74 pin 4

Arduino 5V to TC74 pin 5

Arduino GND to TC74 pin 3

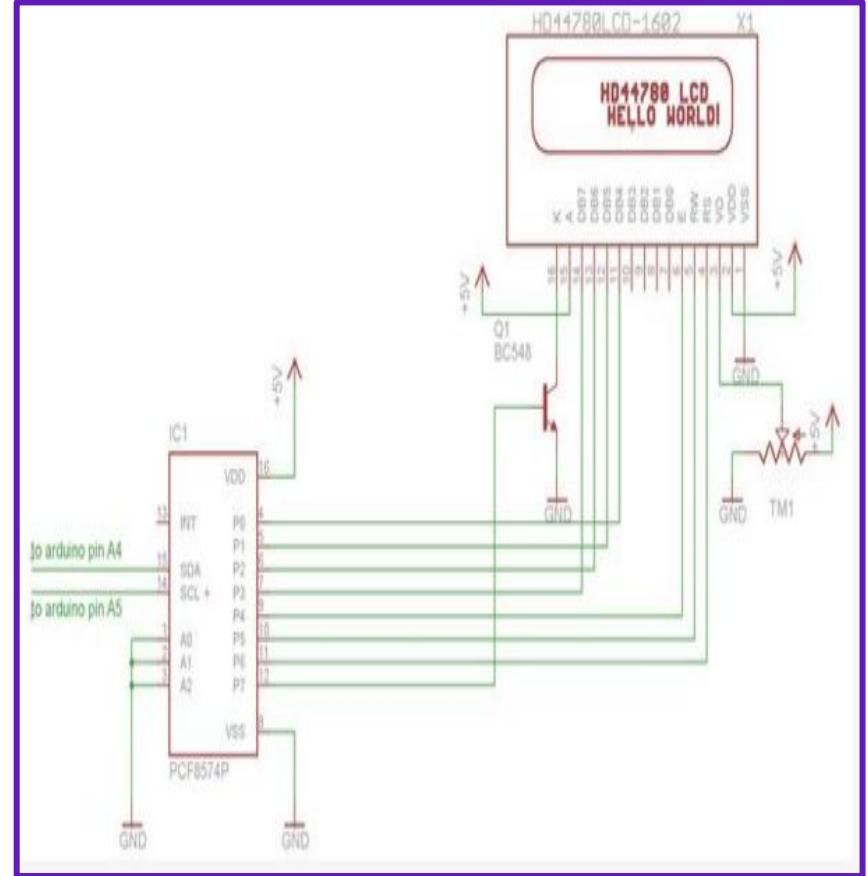
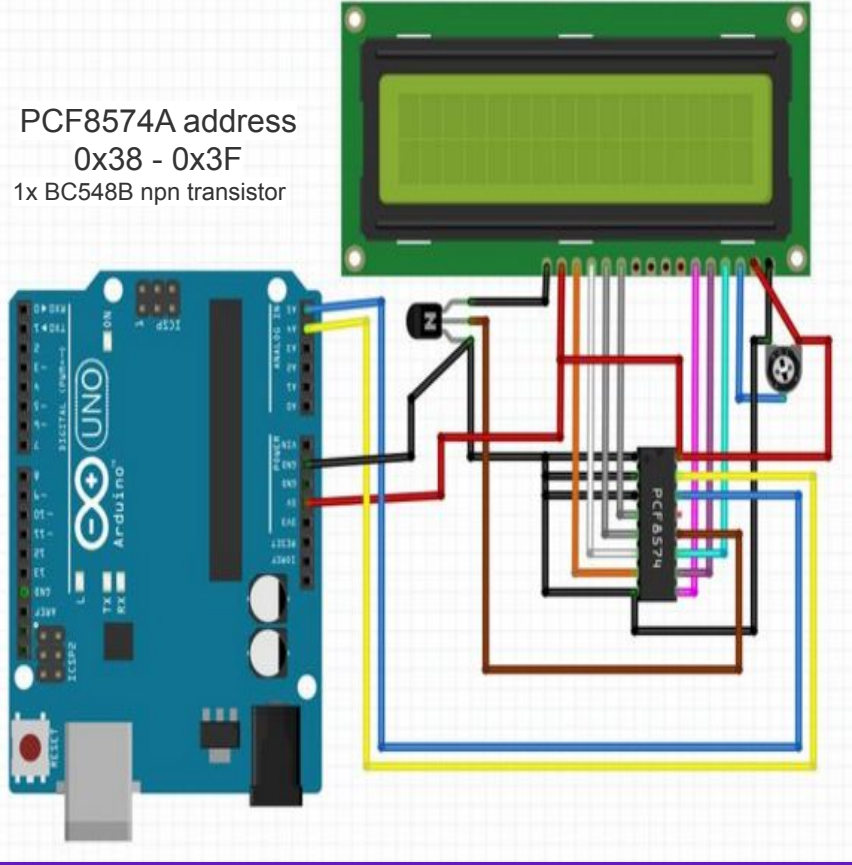
Arduino Pin 12 to Green LED

LED Resistor to GND



Project 7 LCD display I2C adapter for Arduino with PCF8574A

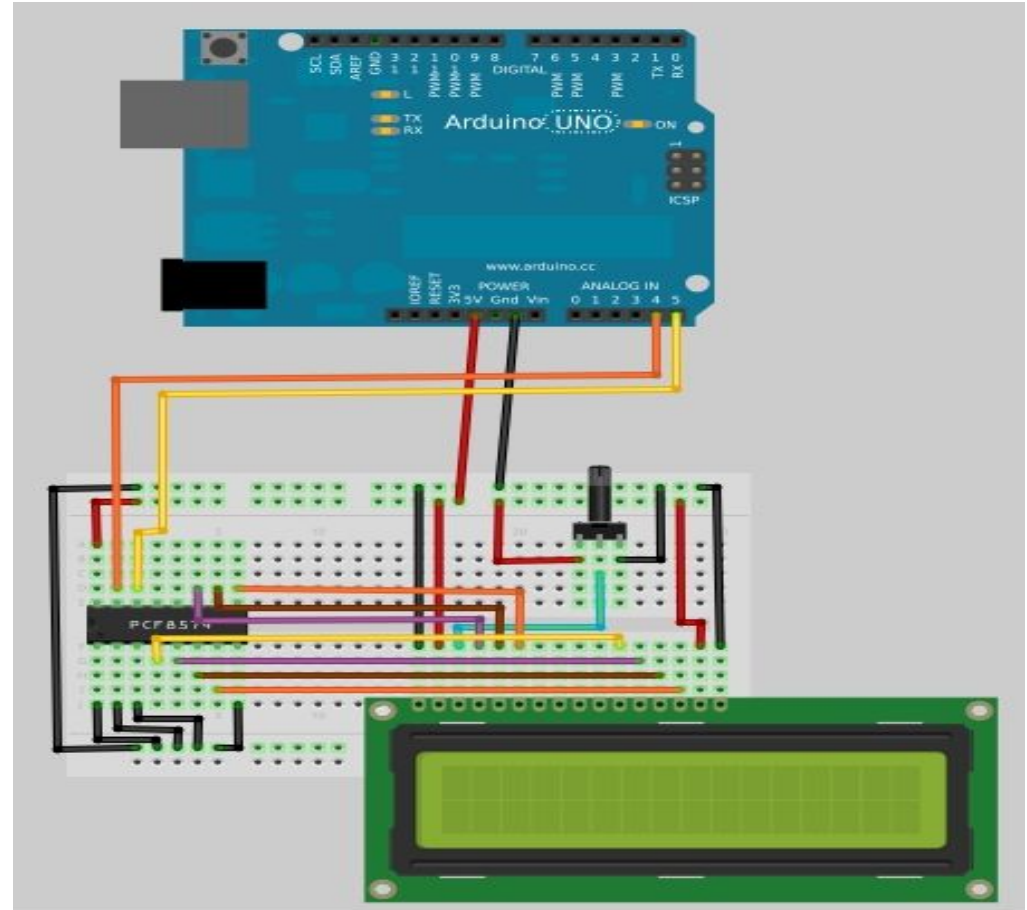
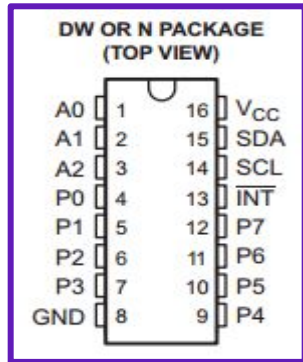
PCF8574A address
0x38 - 0x3F
1x BC548B npn transistor



Project 8 LCD display I2C adapter for Arduino with PCF8574A

1x PCF8574A (or PCF8574)
1x LCD display HD44780
1x 4.7K potentiometer

PCF8574A address 0x38 - 0x3F



Links to the Code

[Program 1 I2C Scanner](#)

[Program 5 Arduino Master](#)

[Program 5 Arduino Slave](#)

[Program 6-a I2C interface for the LCD Display](#)

[Program 6-b Read Temp & Display on LCD](#)

[Program 7 Port expand](#)

[Program 8 Port Expand LCD](#)