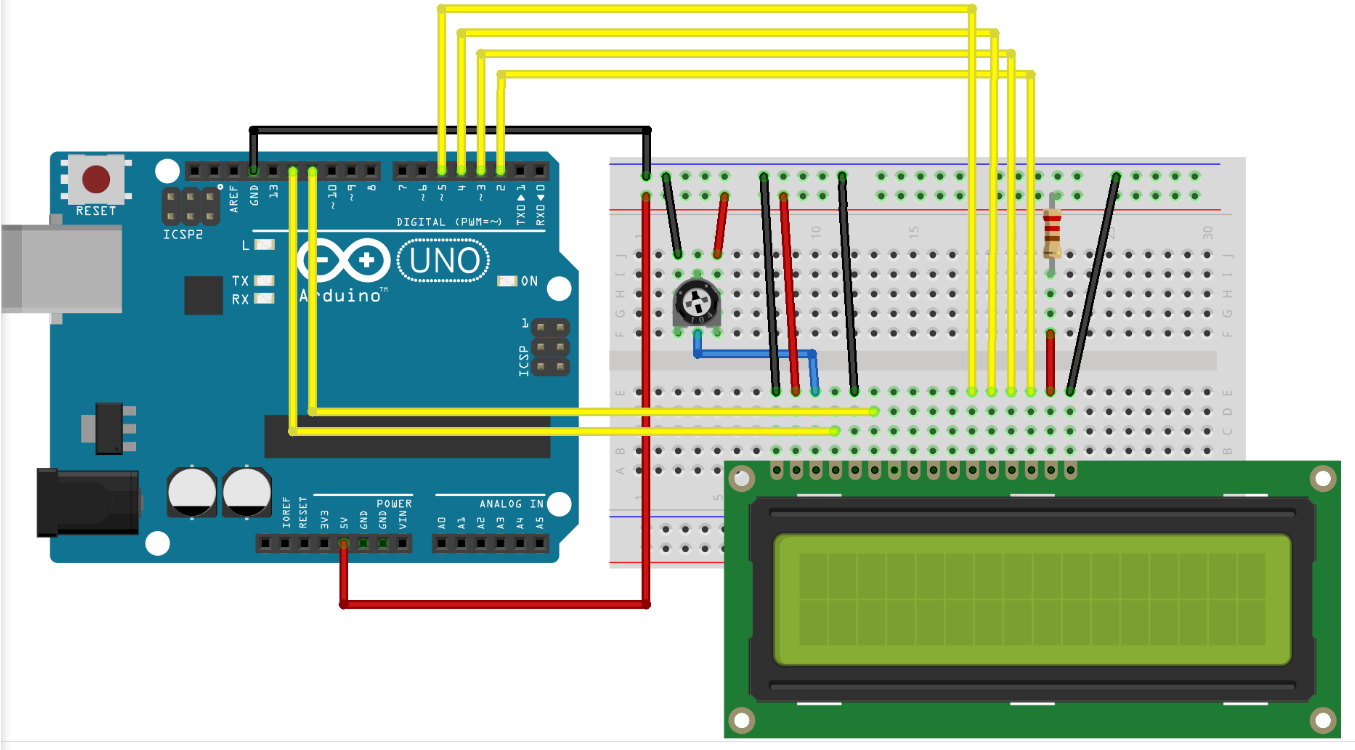
===接线方式===  
\* LCD RS pin to digital pin 12  
\* LCD Enable pin to digital pin 11  
\* LCD D4 pin to digital pin 5  
\* LCD D5 pin to digital pin 4  
\* LCD D6 pin to digital pin 3  
\* LCD D7 pin to digital pin 2  
\* LCD R/W pin to ground  
**\* 10K电位器或者1OK电阻(测试下来，5K的电阻或者电位器显示最正常）：**  
\* ends to +5V and ground  
\* wiper to LCD VO pin (pin 3)



#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

// Print a message to the LCD.

lcd.print("HELLO WORLD! ^\_^");

}

void loop() {

// set the cursor to column 0, line 1

// (note: line 1 is the second row, since counting begins with 0):

lcd.setCursor(0, 1);

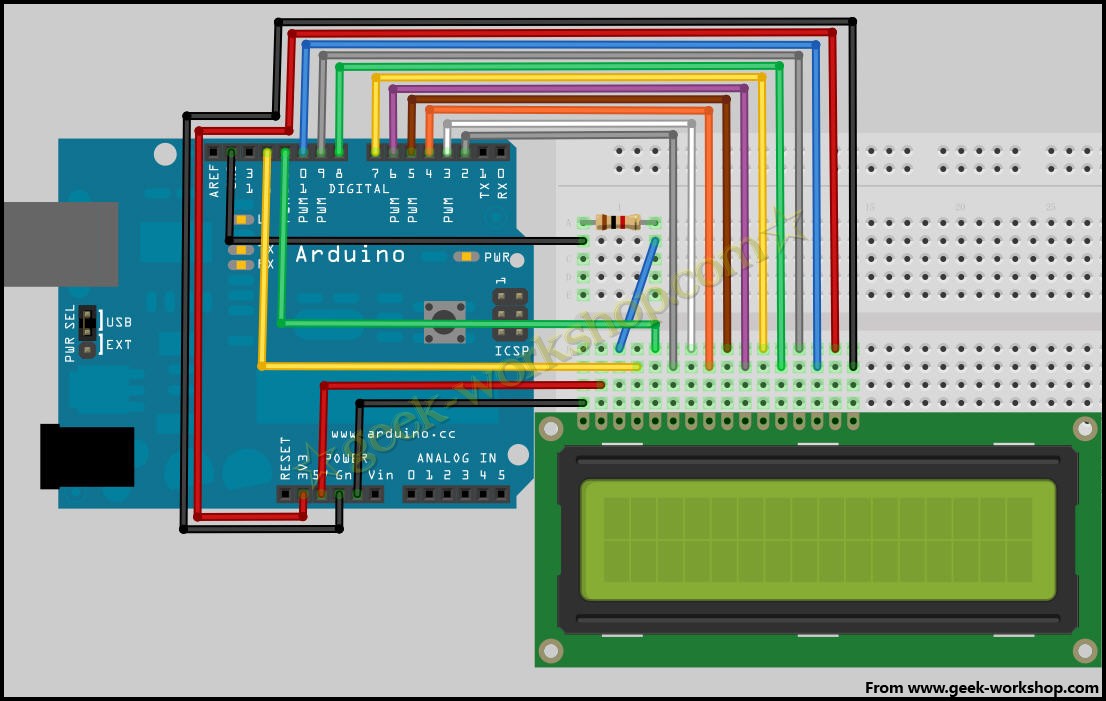
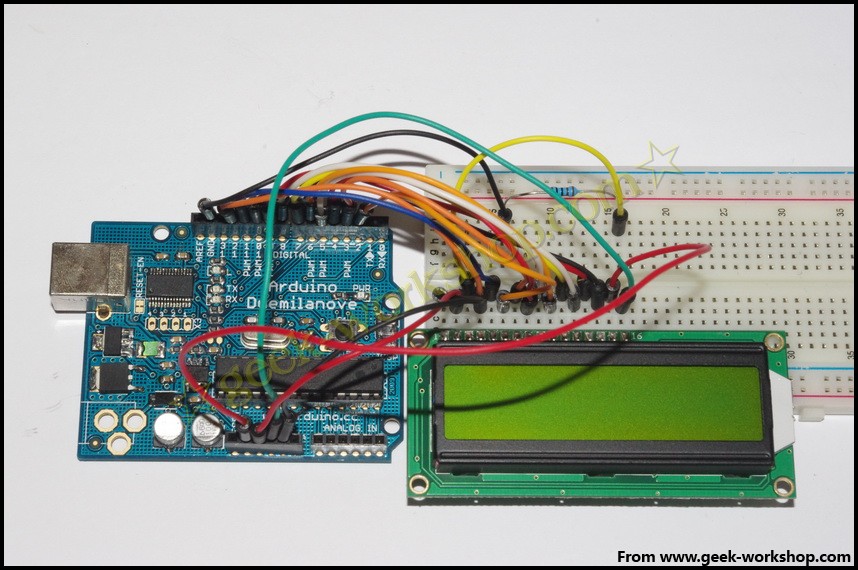
// print the number of seconds since reset:

lcd.print(millis()/1000);

}

Arduino论坛

<https://www.arduino.cc/en/Tutorial/HelloWorld?from=Tutorial.LiquidCrystal>

1602直接与arduino通信，根据产品手册描述，分8位连接法与4位连接法，咱们先使用8位连接法进行实验。硬件连接方式如下图  
  
   
  


int DI = 12;

int RW = 11;

int DB[] = {3, 4, 5, 6, 7, 8, 9, 10};//使用数组来定义总线需要的管脚

int Enable = 2;

void LcdCommandWrite(int value) {

// 定义所有引脚

int i = 0;

for (i=DB[0]; i <= DI; i++) //总线赋值

{

digitalWrite(i,value & 01);//因为1602液晶信号识别是D7-D0(不是D0-D7)，这里是用来反转信号。

value >>= 1;

}

digitalWrite(Enable,LOW);

delayMicroseconds(1);

digitalWrite(Enable,HIGH);

delayMicroseconds(1); // 延时1ms

digitalWrite(Enable,LOW);

delayMicroseconds(1); // 延时1ms

}

void LcdDataWrite(int value) {

// 定义所有引脚

int i = 0;

digitalWrite(DI, HIGH);

digitalWrite(RW, LOW);

for (i=DB[0]; i <= DB[7]; i++) {

digitalWrite(i,value & 01);

value >>= 1;

}

digitalWrite(Enable,LOW);

delayMicroseconds(1);

digitalWrite(Enable,HIGH);

delayMicroseconds(1);

digitalWrite(Enable,LOW);

delayMicroseconds(1); // 延时1ms

}

void setup (void) {

int i = 0;

for (i=Enable; i <= DI; i++) {

pinMode(i,OUTPUT);

}

delay(100);

// 短暂的停顿后初始化LCD

// 用于LCD控制需要

LcdCommandWrite(0x38); // 设置为8-bit接口，2行显示，5x7文字大小

delay(64);

LcdCommandWrite(0x38); // 设置为8-bit接口，2行显示，5x7文字大小

delay(50);

LcdCommandWrite(0x38); // 设置为8-bit接口，2行显示，5x7文字大小

delay(20);

LcdCommandWrite(0x06); // 输入方式设定

// 自动增量，没有显示移位

delay(20);

LcdCommandWrite(0x0E); // 显示设置

// 开启显示屏，光标显示，无闪烁

delay(20);

LcdCommandWrite(0x01); // 屏幕清空，光标位置归零

delay(100);

LcdCommandWrite(0x80); // 显示设置

// 开启显示屏，光标显示，无闪烁

delay(20);

}

void loop (void) {

LcdCommandWrite(0x01); // 屏幕清空，光标位置归零

delay(10);

LcdCommandWrite(0x80+3);

delay(10);

// 写入欢迎信息

LcdDataWrite('W');

LcdDataWrite('e');

LcdDataWrite('l');

LcdDataWrite('c');

LcdDataWrite('o');

LcdDataWrite('m');

LcdDataWrite('e');

LcdDataWrite(' ');

LcdDataWrite('t');

LcdDataWrite('o');

delay(10);

LcdCommandWrite(0xc0+1); // 定义光标位置为第二行第二个位置

delay(10);

LcdDataWrite('g');

LcdDataWrite('e');

LcdDataWrite('e');

LcdDataWrite('k');

LcdDataWrite('-');

LcdDataWrite('w');

LcdDataWrite('o');

LcdDataWrite('r');

LcdDataWrite('k');

LcdDataWrite('s');

LcdDataWrite('h');

LcdDataWrite('o');

LcdDataWrite('p');

delay(5000);

LcdCommandWrite(0x01); // 屏幕清空，光标位置归零

delay(10);

LcdDataWrite('I');

LcdDataWrite(' ');

LcdDataWrite('a');

LcdDataWrite('m');

LcdDataWrite(' ');

LcdDataWrite('h');

LcdDataWrite('o');

LcdDataWrite('n');

LcdDataWrite('g');

LcdDataWrite('y');

LcdDataWrite('i');

delay(3000);

LcdCommandWrite(0x02); //设置模式为新文字替换老文字，无新文字的地方显示不变。

delay(10);

LcdCommandWrite(0x80+5); //定义光标位置为第一行第六个位置

delay(10);

LcdDataWrite('t');

LcdDataWrite('h');

LcdDataWrite('e');

LcdDataWrite(' ');

LcdDataWrite('a');

LcdDataWrite('d');

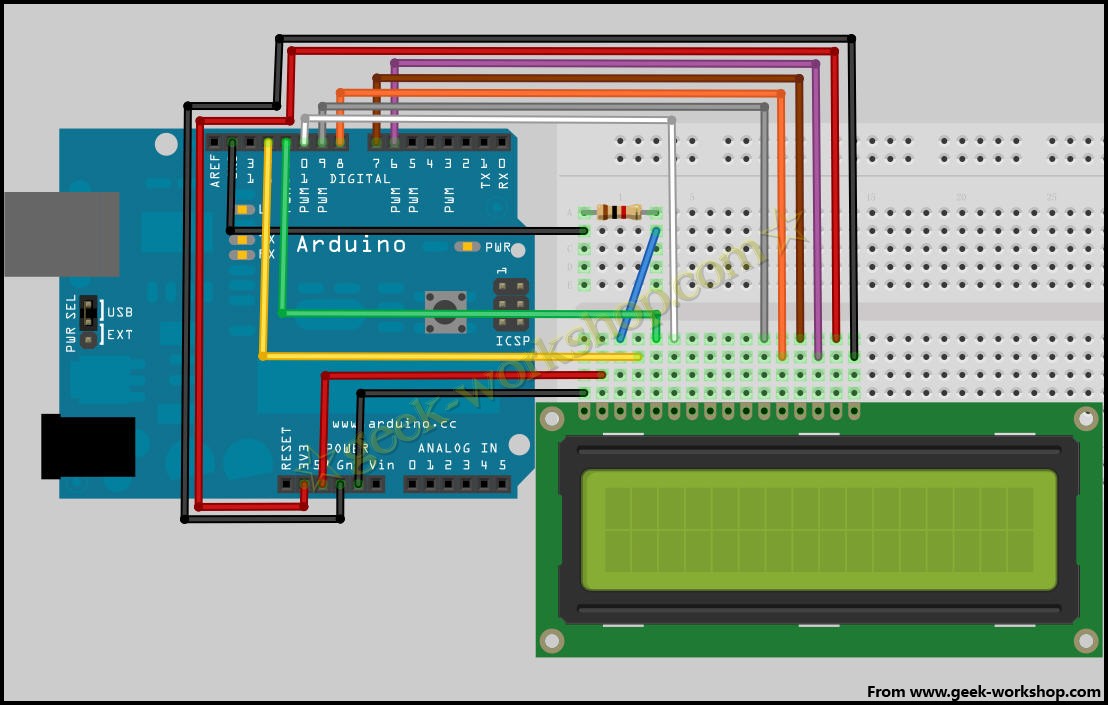
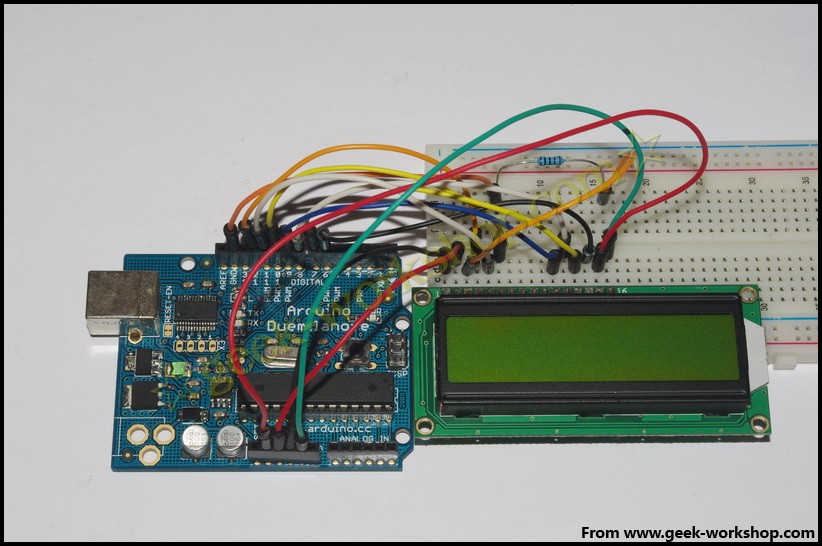
LcdDataWrite('m');

LcdDataWrite('i');

LcdDataWrite('n');

delay(5000);

}

**4位接法**  
在正常使用下，8位接法基本把arduino的数字端口占满了，如果想要多接几个传感器就没有端口了，这种情况下怎么处理呢，咱们可以使用4位接法。  
4位接法的硬件连接方法如下图  
   
  


int LCD1602\_RS=12;

int LCD1602\_RW=11;

int LCD1602\_EN=10;

int DB[] = { 6, 7, 8, 9};

char str1[]="Welcome to";

char str2[]="geek-workshop";

char str3[]="this is the";

char str4[]="4-bit interface";

void LCD\_Command\_Write(int command)

{

int i,temp;

digitalWrite( LCD1602\_RS,LOW);

digitalWrite( LCD1602\_RW,LOW);

digitalWrite( LCD1602\_EN,LOW);

temp=command & 0xf0;

for (i=DB[0]; i <= 9; i++)

{

digitalWrite(i,temp & 0x80);

temp <<= 1;

}

digitalWrite( LCD1602\_EN,HIGH);

delayMicroseconds(1);

digitalWrite( LCD1602\_EN,LOW);

temp=(command & 0x0f)<<4;

for (i=DB[0]; i <= 9; i++)

{

digitalWrite(i,temp & 0x80);

temp <<= 1;

}

digitalWrite( LCD1602\_EN,HIGH);

delayMicroseconds(1);

digitalWrite( LCD1602\_EN,LOW);

}

void LCD\_Data\_Write(int dat)

{

int i=0,temp;

digitalWrite( LCD1602\_RS,HIGH);

digitalWrite( LCD1602\_RW,LOW);

digitalWrite( LCD1602\_EN,LOW);

temp=dat & 0xf0;

for (i=DB[0]; i <= 9; i++)

{

digitalWrite(i,temp & 0x80);

temp <<= 1;

}

digitalWrite( LCD1602\_EN,HIGH);

delayMicroseconds(1);

digitalWrite( LCD1602\_EN,LOW);

temp=(dat & 0x0f)<<4;

for (i=DB[0]; i <= 9; i++)

{

digitalWrite(i,temp & 0x80);

temp <<= 1;

}

digitalWrite( LCD1602\_EN,HIGH);

delayMicroseconds(1);

digitalWrite( LCD1602\_EN,LOW);

}

void LCD\_SET\_XY( int x, int y )

{

int address;

if (y ==0) address = 0x80 + x;

else address = 0xC0 + x;

LCD\_Command\_Write(address);

}

void LCD\_Write\_Char( int x,int y,int dat)

{

LCD\_SET\_XY( x, y );

LCD\_Data\_Write(dat);

}

void LCD\_Write\_String(int X,int Y,char \*s)

{

LCD\_SET\_XY( X, Y ); //设置地址

while (\*s) //写字符串

{

LCD\_Data\_Write(\*s);

s ++;

}

}

void setup (void)

{

int i = 0;

for (i=6; i <= 12; i++)

{

pinMode(i,OUTPUT);

}

delay(100);

LCD\_Command\_Write(0x28);//4线 2行 5x7

delay(50);

LCD\_Command\_Write(0x06);

delay(50);

LCD\_Command\_Write(0x0c);

delay(50);

LCD\_Command\_Write(0x80);

delay(50);

LCD\_Command\_Write(0x01);

delay(50);

}

void loop (void)

{

LCD\_Command\_Write(0x01);

delay(50);

LCD\_Write\_String(3,0,str1);//第1行，第4个地址起

delay(50);

LCD\_Write\_String(1,1,str2);//第2行，第2个地址起

delay(5000);

LCD\_Command\_Write(0x01);

delay(50);

LCD\_Write\_String(0,0,str3);

delay(50);

LCD\_Write\_String(0,1,str4);

delay(5000);

}