

上海交通大学黑客马拉松

Arduino 使用指南

Hackathon OMNI-Lab

目录

一、Arduino 简介	3
1. 1、什么是 Arduino	3
1. 2、Arduino 特点	3
1. 3、Arduino 性能	3
1. 4、Arduino 技术指标	3
1.4.1 概要	3
1.4.2 电源	4
1.4.3 输入输出	4
1.4.4 通信接口	5
二、搭建 Arduino 环境	5
1、安装 Arduino	5
2、安装 USB 驱动	5
三、Arduino 语言介绍	11
3.1、Arduino 程序基本构架	11
3.1.1 Arduino 程序结构	11
3.1.2 Arduino 函数功能	11
3.2、Arduino 常用编辑语言	13
3.2.1 关键词	13
3.2.2 语法符号	14
3.2.3 运算符	14
3.2.4 数据类型	14
3.2.5 数据类型转换	15
3.2.6 常量	15
四、Arduino 实例	15
4.1、准备工作	15
4.1.1、Arduino 环境	15
4.2、串口监视器使用---“Hello World!”	18
4.3、外设传感器使用---点亮普通 LED	23
4.4、外设传感器使用---点亮双色 LED	24
4.5、函数库的调用---DHT22 温湿度传感器	24

一、Arduino 简介

1. 1、什么是 Arduino

Arduino，是一个基于开放原始码的软硬件平台，构建于开放原始码 simple I/O 介面版，并且具有使用类似 Java，C 语言的 Processing/Wiring 开发环境。Arduino 可以用来开发交互产品，比如它可以读取大量的开关和传感器信号，并且可以控制各式各样的电灯、电机和其他物理设备。Arduino 项目可以是单独的，也可以在运行时和你电脑中运行的程序（例如：Flash，Processing，MaxMSP）进行通讯。Arduino 板你可以选择自己去手动组装或是购买已经组装好的；Arduino 开源的 IDE 可以免费下载得到。Arduino 的编程语言就像似在对一个类似于物理的计算平台进行相应的连线，它基于处理多媒体的编程环境。

1. 2、Arduino 特点

- 1、开放源代码的电路图设计，程序开发接口免费下载，也可依个人需求自己修改。
- 2、使用低价格的微处理控制器(AVR 系列控制器)，可以采用 USB 接口供电，不需外接电源，也可以使用外部 9VDC 输入。
- 3、Arduino 支持 ISP 在线烧，可以将新的“bootloader”固件烧入 AVR 芯片。有了 bootloader 之后，可以通过串口或者 USB to Rs232 线更新固件。
- 4、可依据官方提供的 Eagle 格式 PCB 和 SCH 电路图简化 Arduino 模组，完成独立运作的微处理控制；可简单地与传感器，各式各样的电子元件连接(例如：红外线,超音波,热敏电阻,光敏电阻,伺服马达,...等)
- 5、支持多种互动程序，如：Flash、Max/Msp、Vvvv、PD、C、Processing 等。
- 6、应用方面，利用 Arduino，突破以往只能使用鼠标、键盘、CCD 等输入的装置的互动内容，可以更简单地达成单人或多人游戏互动。

1. 3、Arduino 性能

Arduino Mega2560 也是采用 USB 接口的核心电路板，它最大的特点就是具有多达 54 路数字输入输出，特别适合需要大量 IO 接口的设计。Mega2560 的处理器核心是 ATmega2560，同时具有 54 路数字输入/输出口（其中 16 路可作为 PWM 输出），16 路模拟输入，4 路 UART 接口，一个 16MHz 晶体振荡器，一个 USB 口，一个电源插座，一个 ICSP header 和一个复位按钮。Arduino Mega2560 也能兼容为 Arduino UNO 设计的扩展板。

1. 4、Arduino 技术指标

1. 4. 1 概要

- 处理器 ATmega2560
- 工作电压 5V
- 输入电压（推荐） 7-12V
- 输入电压（范围） 6-20V
- 数字 IO 脚 54（其中 16 路作为 PWM 输出）
- 模拟输入脚 16
- IO 脚直流电流 40 mA
- 3.3V 脚直流电流 50 mA
- Flash Memory 256 KB（ATmega328，其中 8 KB 用于 bootloader）
- SRAM 8 KB
- EEPROM 4 KB
- 工作时钟 16 MHz

1.4.2 电源

Arduino Mega2560 可以通过 3 种方式供电，而且能自动选择供电方式

- 外部直流电源通过电源插座供电。
- 电池连接电源连接器的 GND 和 VIN 引脚。
- USB 接口直接供电。

电源引脚说明

- VIN --- 当外部直流电源接入电源插座时，可以通过 VIN 向外部供电；也可以通过此引脚向 Mega2560 直接供电；VIN 有电时将忽略从 USB 或者其他引脚接入的电源。
- 5V --- 通过稳压器或 USB 的 5V 电压，为 UNO 上的 5V 芯片供电。
- 3.3V --- 通过稳压器产生的 3.3V 电压，最大驱动电流 50mA。
- GND --- 地脚。

1.4.3 输入输出

1. **14 路数字输入输出：**工作电压为 5V，每一路能输出和接入最大电流为 40mA。每一路配置了 20-50K 欧姆内部上拉电阻（默认不连接）。除此之外，有些引脚有特定的功能
 - 4 路串口信号：串口 0---0(RX) and 1(TX)；串口 1---19(RX) and 18(TX)；串口 2---17(RX) and 16(TX)；串口 3---15(RX) and 14(TX)。其中串口 0 与内部 ATmega8U2 USB-to-TTL 芯片相连，提供 TTL 电压水平的串口接收信号。
 - 6 路外部中断：2(中断 0)，3(中断 1)，18(中断 5)，19(中断 4)，20(中断 3)，and 21(中断 2)。触发中断引脚，可设成上升沿、下降沿或同时触发。
 - 14 路脉冲宽度调制 PWM（0--13）：提供 14 路 8 位 PWM 输出。
 - SPI（53(SS)，51(MOSI)，50(MISO)，52(SCK)）：SPI 通信接口。

- LED（13 号）：Arduino 专门用于测试 LED 的保留接口，输出为高时点亮 LED，反之输出为低时 LED 熄灭。
2. 16 路模拟输入：每一路具有 10 位的分辨率（即输入有 1024 个不同值），默认输入信号范围为 0 到 5V，可以通过 AREF 调整输入上限。除此之外，有些引脚有特定功能
- TWI 接口（20（SDA）和 21（SCL））：支持通信接口（兼容 I2C 总线）。
3. AREF：模拟输入信号的参考电压。
 4. Reset：信号为低时复位单片机芯片。

1.4.4 通信接口

1. 串口：ATmega2560 内置的 4 路 UART 可以与外部实现串口通信；ATmega16U2 可以访问串口 0 实现 USB 上的虚拟串口。
2. 支持 TWI（兼容 I2C）接口。
3. 支持 SPI 接口。

二、搭建 Arduino 环境

1、安装 Arduino

采用 Arduino IDE 开发工具 Arduino-1.0。5-r2 免安装开发工具。打开文件夹 arduino-1.0.5-r2 下的 arduino.exe 即可使用 Arduino IDE。

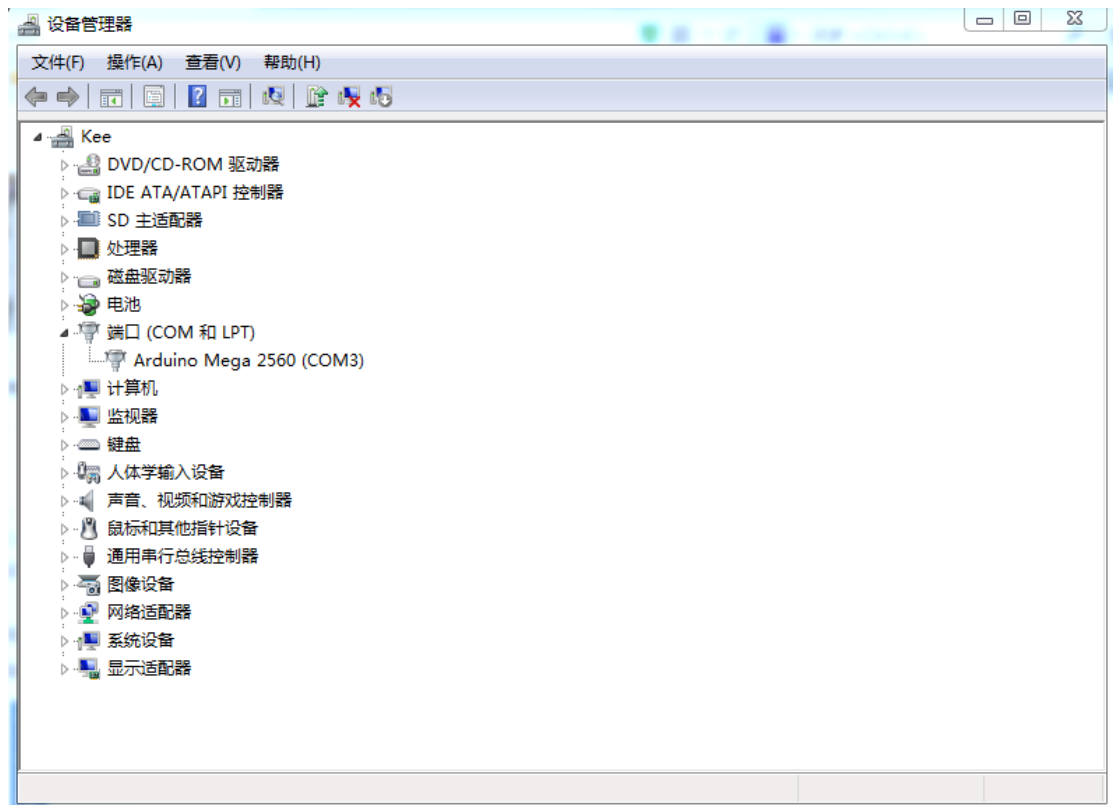
2、安装 USB 驱动

安装 USB 的驱动如下：

- （1） 连接 Arduino 开发板和电脑
- （2） 系统会自动检查设备，如果显示驱动安装成功，即可使用



此时可右击 计算机>属性> 左击“设备管理器”> 端口 查看端口情况确认驱动安装成功

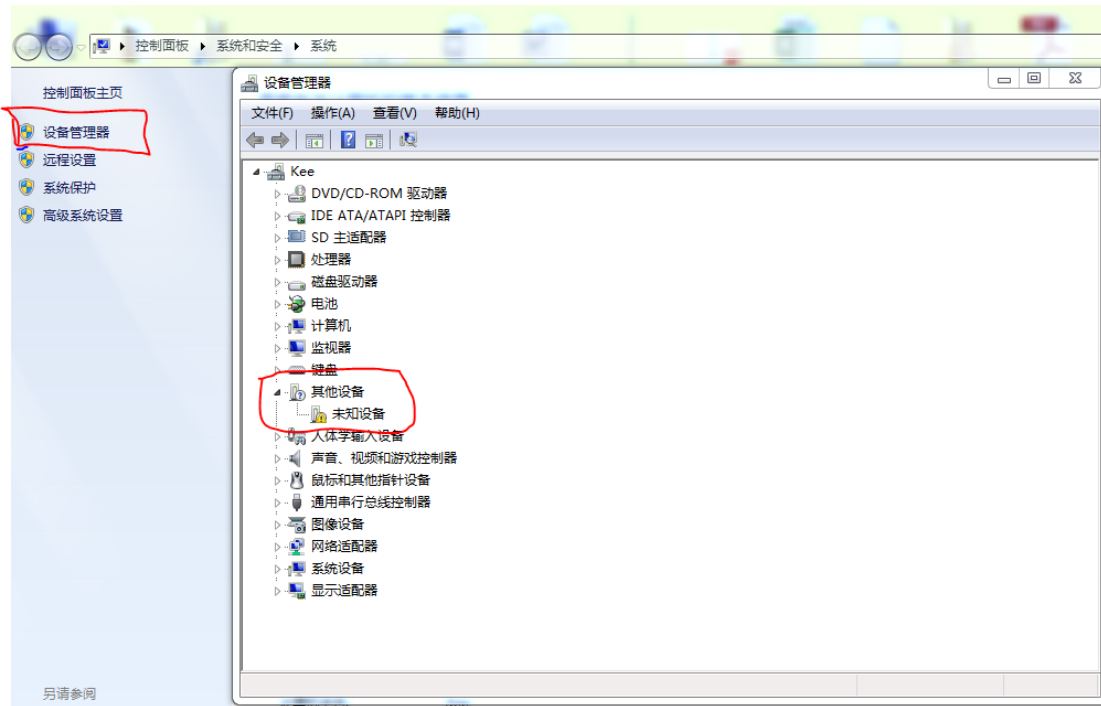


(3) 如果出现设备安装不成功情况，

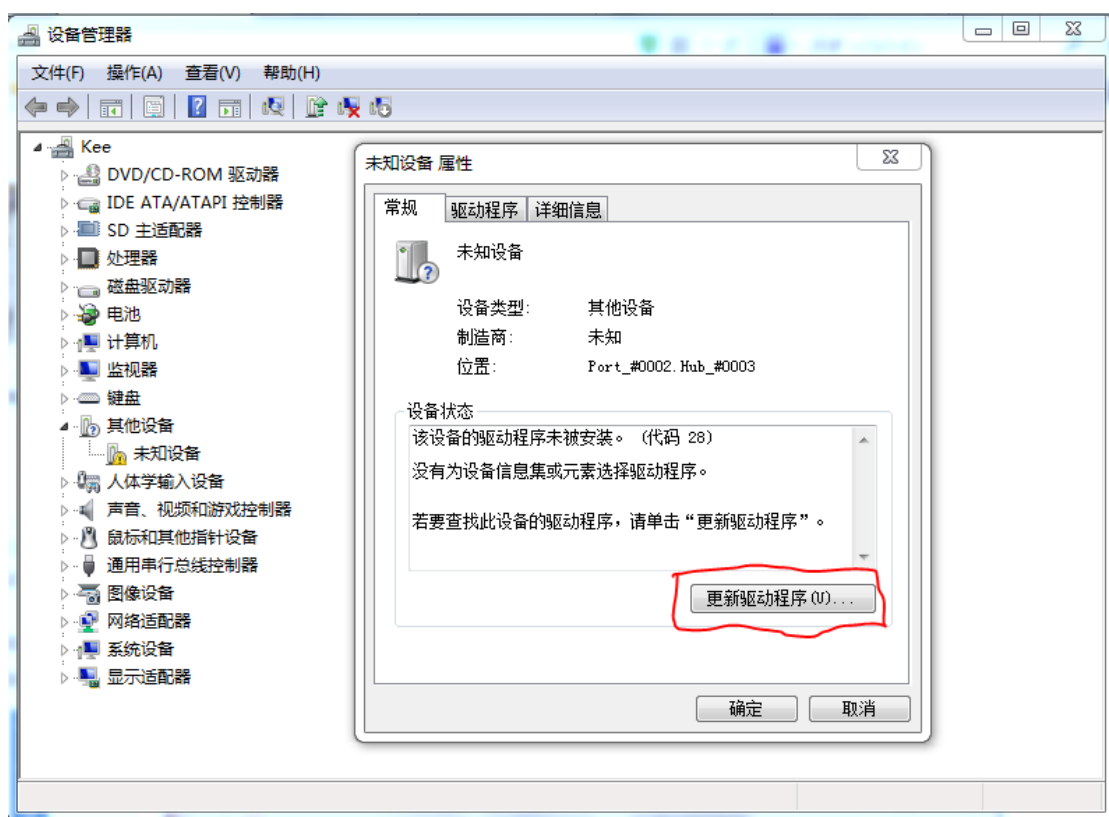


此时则需要手动安装驱动，步骤如下：

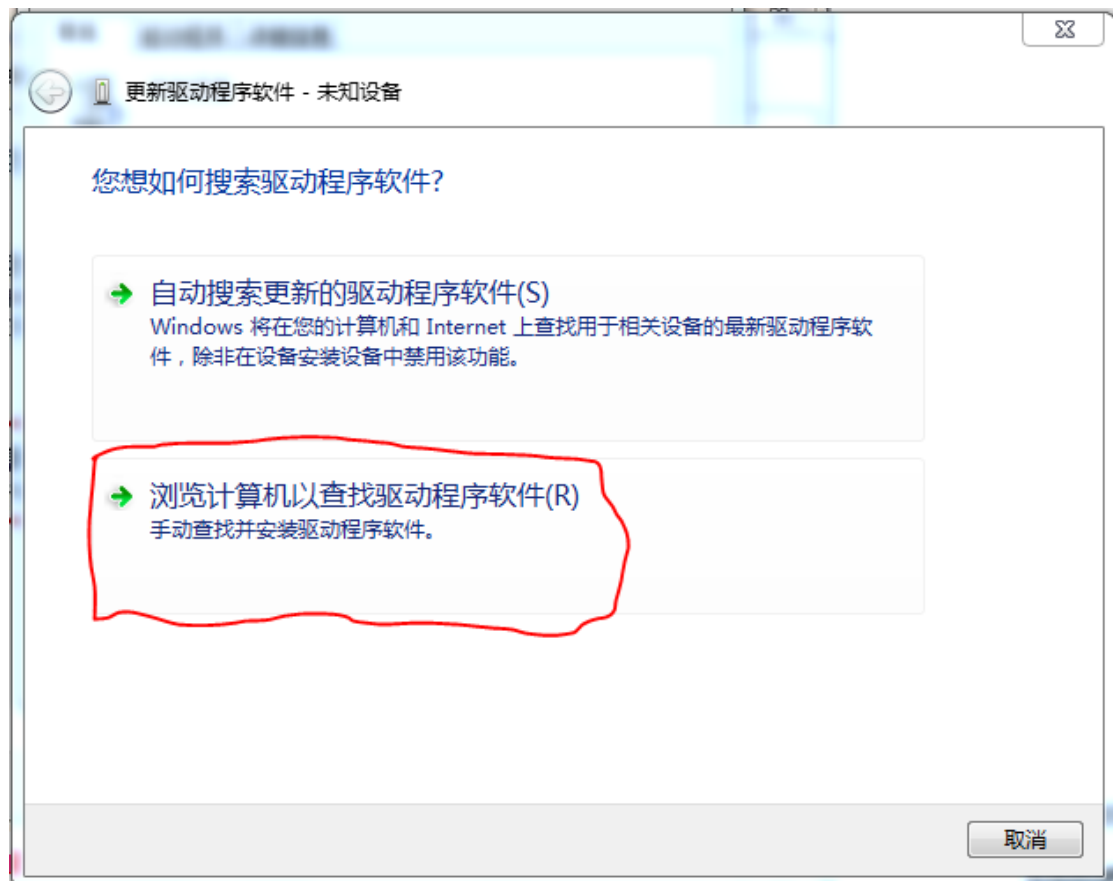
右键计算机选择“属性” >> 左击“设备管理器”>>在其他设备下 双击 “未知设备”，



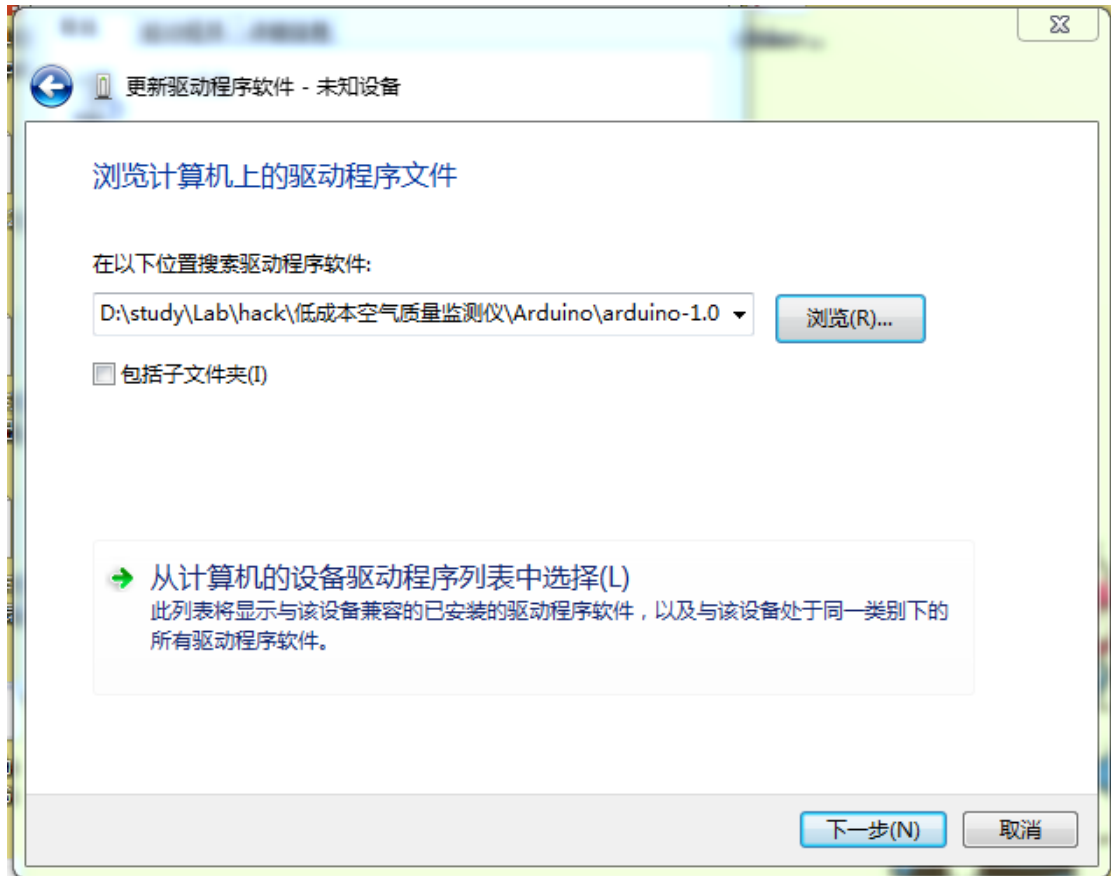
此时会出现“未知设备 属性”对话框，点击“更新驱动程序”



在出现的对话框中选择“连蓝计算机以查找驱动程序软件”

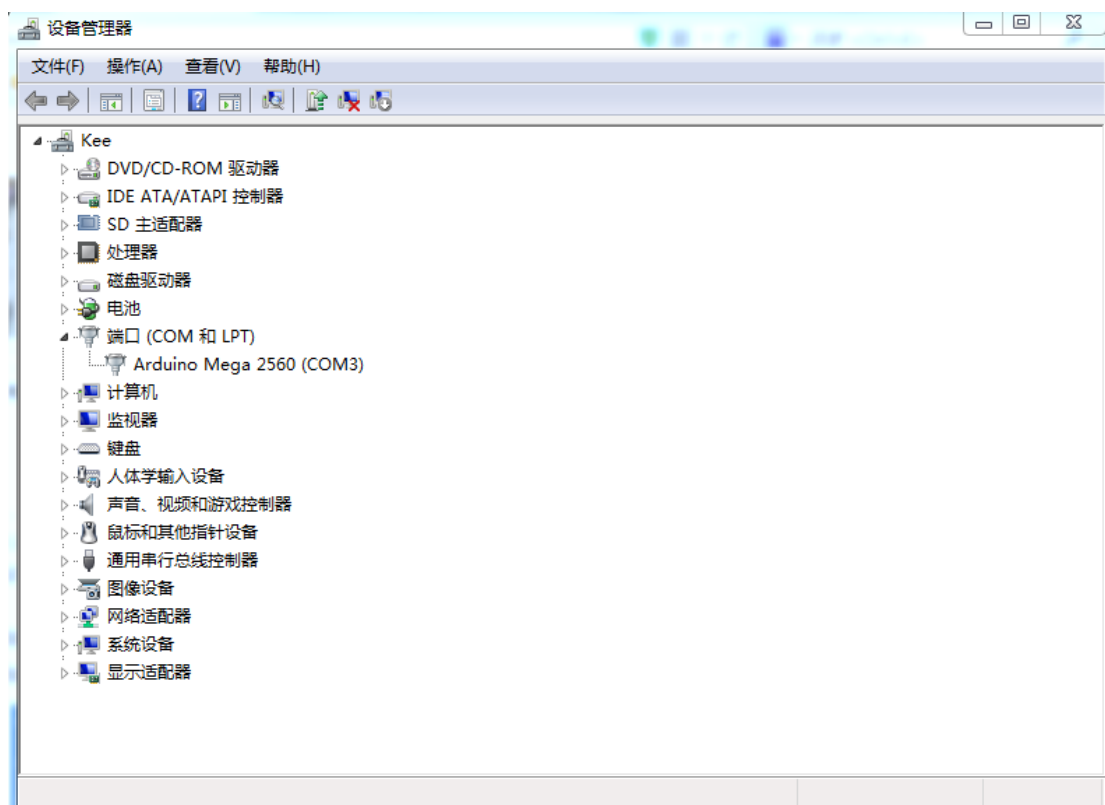
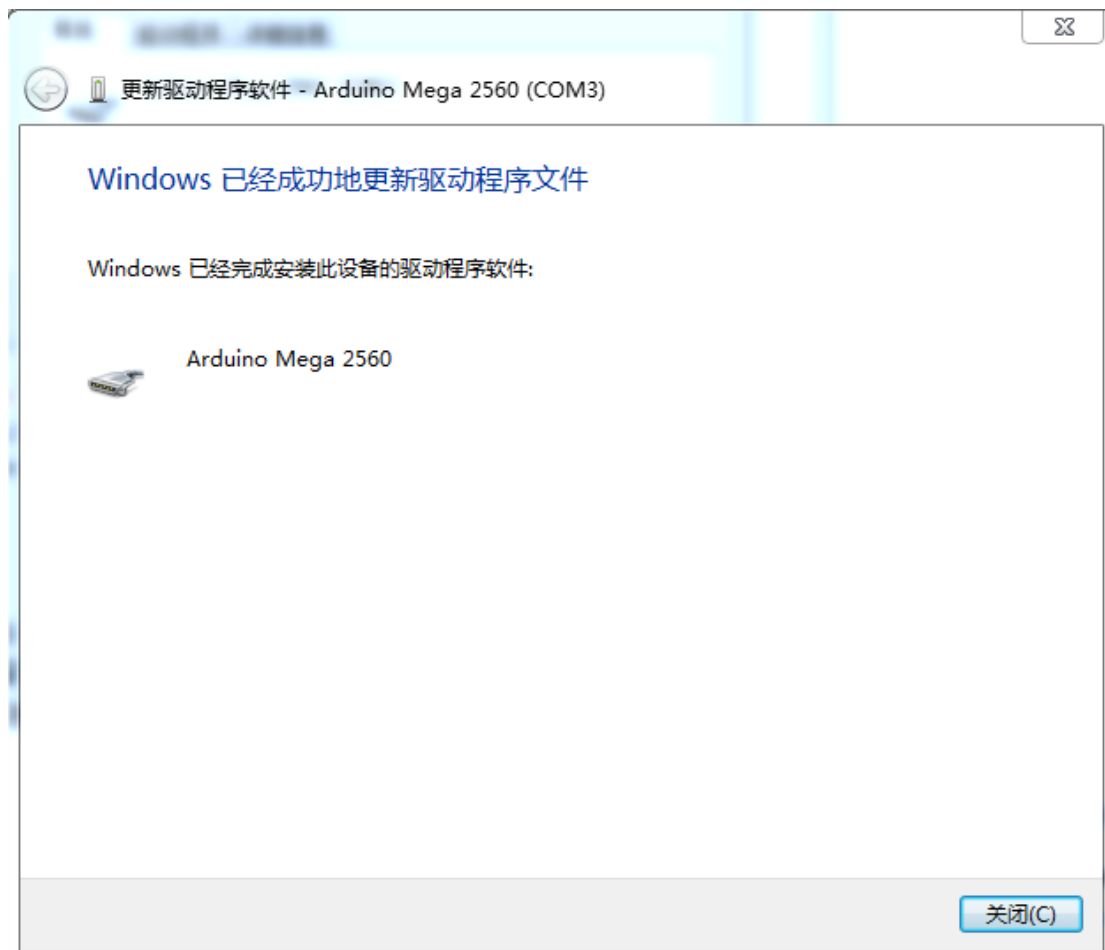


在出现的对话框中点击“浏览”找到 `arduino-1.0.5-r2` 安装文件夹下的 `drivers` 添加到路径里，然后点击“下一步”



然后点击“安装”

安装完成后出现如下提示，显示驱动安装成功



三、Arduino 语言介绍

3.1、Arduino 程序基本构架

3.1.1 Arduino 程序结构

- 1、声明变量及接口名称 (`int val;int ledPin=13;`), 调用库函数
- 2、`setup()`——函数在程序开始时使用, 可以初始化变量、接口模式、启用库等 (例如:
`pinMode(ledPin,OUTPUT);`)
- 3、`loop()` ——在`setup()`函数之后, 即初始化之后, `loop()` 让你的程序循环地被执行。使用它来运转 Arduino。

3.1.2 Arduino 函数功能

数字 I/O:

- **pinMode**(pin, mode): 数字 IO 口输入输出模式定义函数, pin 表示为0~53, mode 表示为 INPUT 或 OUTPUT。
- **Void digitalWrite**(pin, value): 数字 IO 口输出电平定义函数, pin 表示为0~53, value 表示为 HIGH 或 LOW。比如定义 HIGH 可以驱动 LED。
- **int digitalRead**(pin): 数字 IO 口读输入电平函数, pin 表示为 0~53, value 表示为 HIGH 或 LOW。比如可以读数字传感器。

模拟 I/O:

- **int analogRead**(pin): 模拟 IO 口读函数, pin 表示为0~15 比如可以读模拟传感器 (10位 AD, 0~5V 表示为0~1023)
- **Void analogWrite**(pin, value) – PWM: 数字 IO 口 PWM 输出函数, Arduino 数字 IO 口标注了 PWM 的 IO 口可使用该函数, pin 表示 2~13, value 表示为 0~255。比如可用于电机 PWM 调速或音乐播放、LED 变色控制等。

时间函数:

- **unsigned long millis()**: 返回时间函数 (单位 ms) 该函数是指, 当程序运行就开始计时并返回记录的参数, 该参数溢出大概需要50 天时间。
- **delay**(ms): 延时函数 (单位 ms)

- **delayMicroseconds**(us): 延时函数（单位 us）。

数学函数：

- **min**(x, y) 求最小值
- **max**(x, y) 求最大值
- **abs**(x) 计算绝对值
- **constrain**(x, a, b) 约束函数，下限 a，上限 b，x 必须在 ab 之间 才能返回。
- **map**(value, fromLow, fromHigh, toLow, toHigh) 约束函数，
value 必须在 fromLow 与 toLow 之间和 fromHigh 与 toHigh 之间。
- **pow**(base, exponent) 开方函数，base 的 exponent 次方。
- **sq**(x) 平方
- **sqrt**(x) 开根号

三角函数：

- **sin**(rad)
- **cos**(rad)
- **tan**(rad)

随机数函数：

- **randomSeed**(seed) 随机数端口定义函数，seed 表示读模拟 口
analogRead(pin)函数。
- long **random**(max) 随机数函数，返回数据大于等于0，小于 max。
- long **random**(min, max) 随机数函数，返回数据大于等于
min，小于 max。

外部中断函数：

- **attachInterrupt**(interrupt, , mode): 外部中断只能用到数字 IO 口2 和3，
interrupt 表示中断口初始0 或1, 表示一个功能函 数mode**LOW** 低电平中断，

CHANGE 有变化就中断**RISING** 上升沿中断，**FALLING**下降沿中断。

- **detachInterrupt**(interrupt): 中断开关，interrupt=1 开，interrupt=0 关。

中断使能函数：

- **interrupts**() 使能中断
- **noInterrupts**()终止中断

串口收发函数：

- **Serial.begin**(speed) 串口定义波特率函数，speed 表示波特率，如9600，19200 等。
- **int Serial.available**() 判断缓冲器状态。
- **int Serial.read**() 读串口并返回收到参数。
- **Serial.flush**() 清空缓冲器。
- **Serial.print**(data) 串口输出数据。
- **Serial.println**(data) 串口输出数据并带回车符

3.2、Arduino 常用编辑语言

3.2.1 关键词

- **if**
- **if...else**
- **for**
- **switch case**
- **while**
- **do... while**
- **break**
- **continue**
- **return**
- **goto**

3.2.2 语法符号

- `:`
- `Ω`
- `//`
- `/* */`

3.2.3 运算符

- `≡`
- `±`
- `:`
- `*`
- `/`
- `%`
- `≡≡`
- `!≡`
- `≤`
- `≥`
- `≤≡`
- `≥≡`
- `&&`
- `//`
- `!`
- `±±`
- `≡`
- `±≡`
- `≡≡`
- `*≡`
- `≡`

3.2.4 数据类型

- `boolean` 布尔类型
- `char`
- `Byte`
- `int`

- unsigned int
- long
- unsigned long
- float
- double
- string
- array
- void

3.2.5 数据类型转换

- char()
- byte()
- int()
- long()
- float()

3.2.6 常量

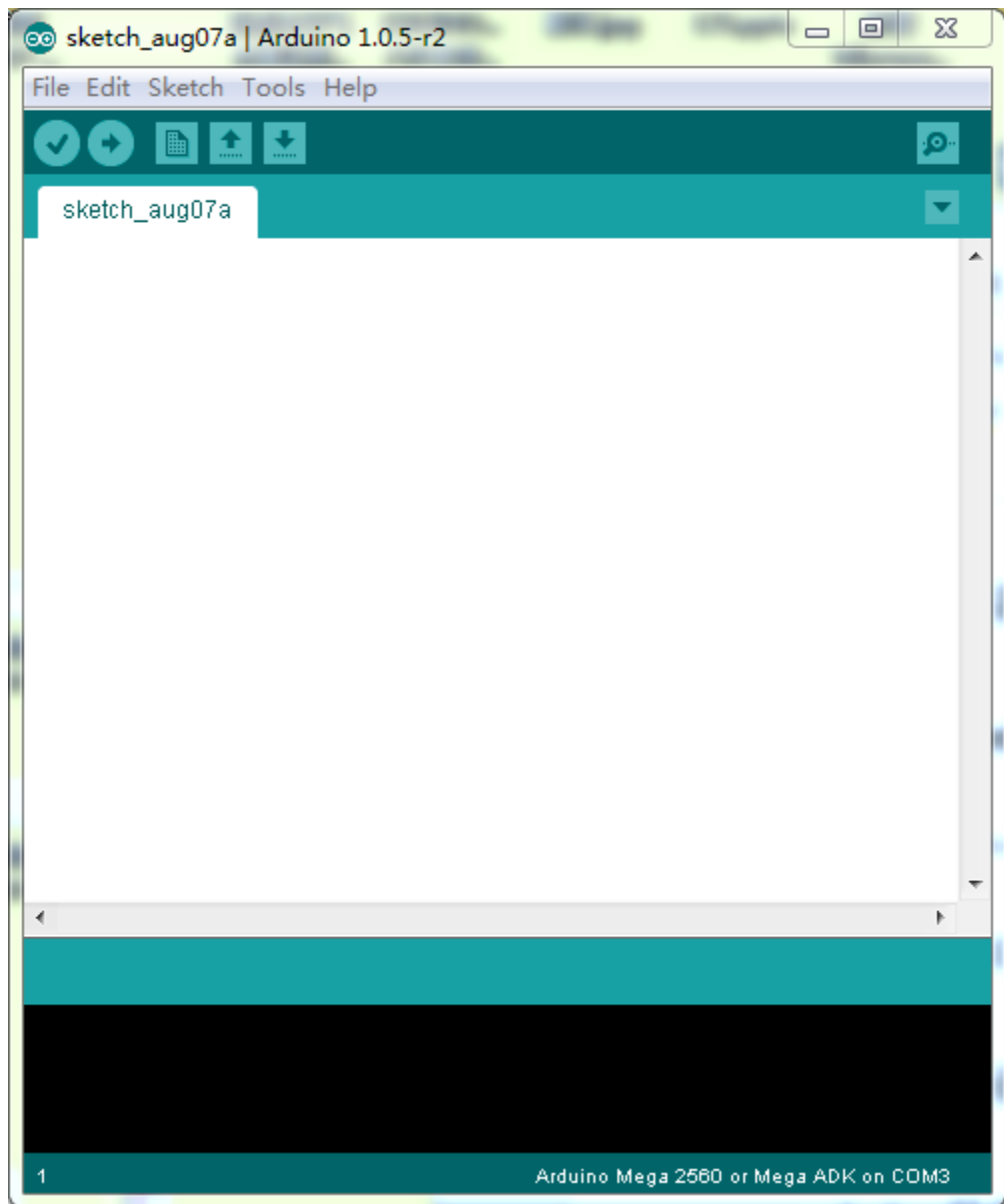
- HIGH | LOW 表示数字 IO 口的电平, HIGH 表示高电平 (1), LOW 表示低电平 (0)
- INPUT | OUTPUT 表示数字 IO 口的方向, INPUT 表示输入 (高阻态), OUTPUT 表示输出 (AVR 能提供 5V 电压 40mA 电流)
- true | false true 表示真 (1), false 表示假 (0)

四、Arduino 实例

4.1、准备工作

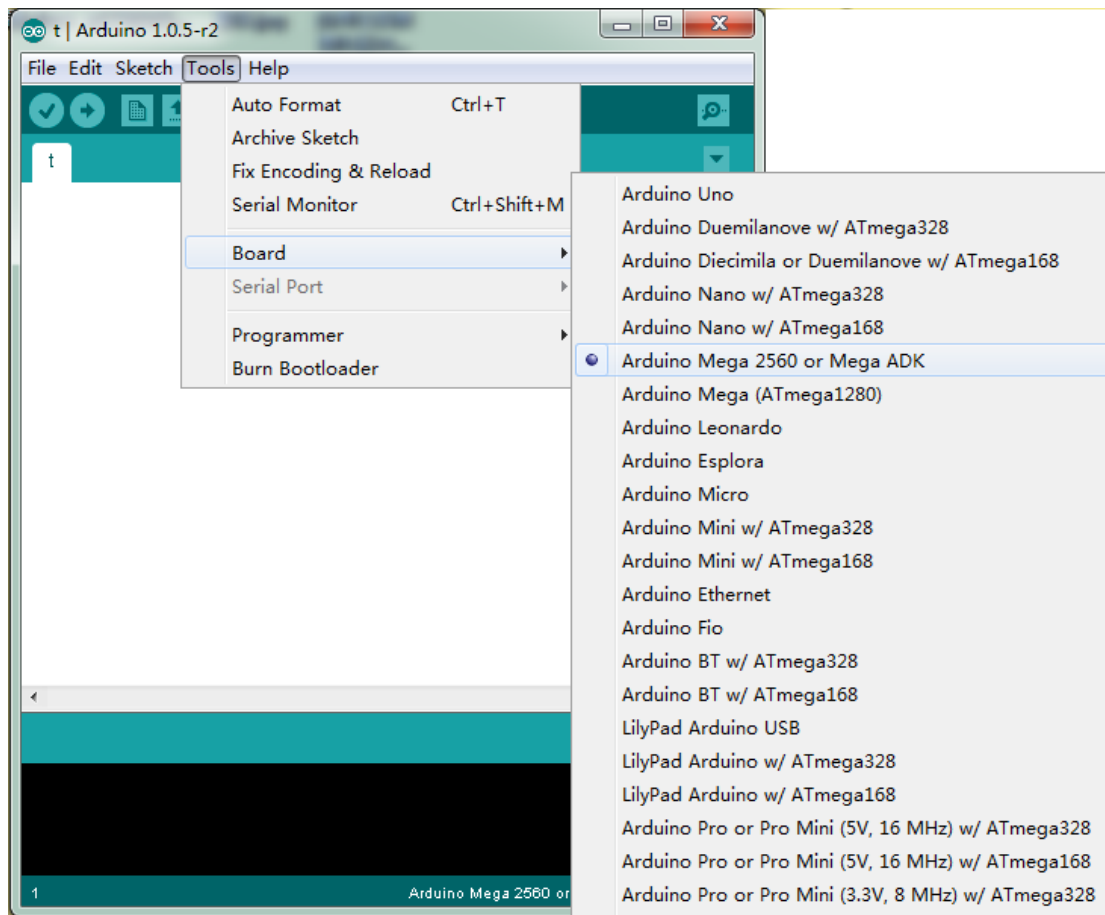
4.1.1、Arduino 环境

(1) 双击 arduino.exe 图标, 打开 Arduino IDE 界面, 如下图

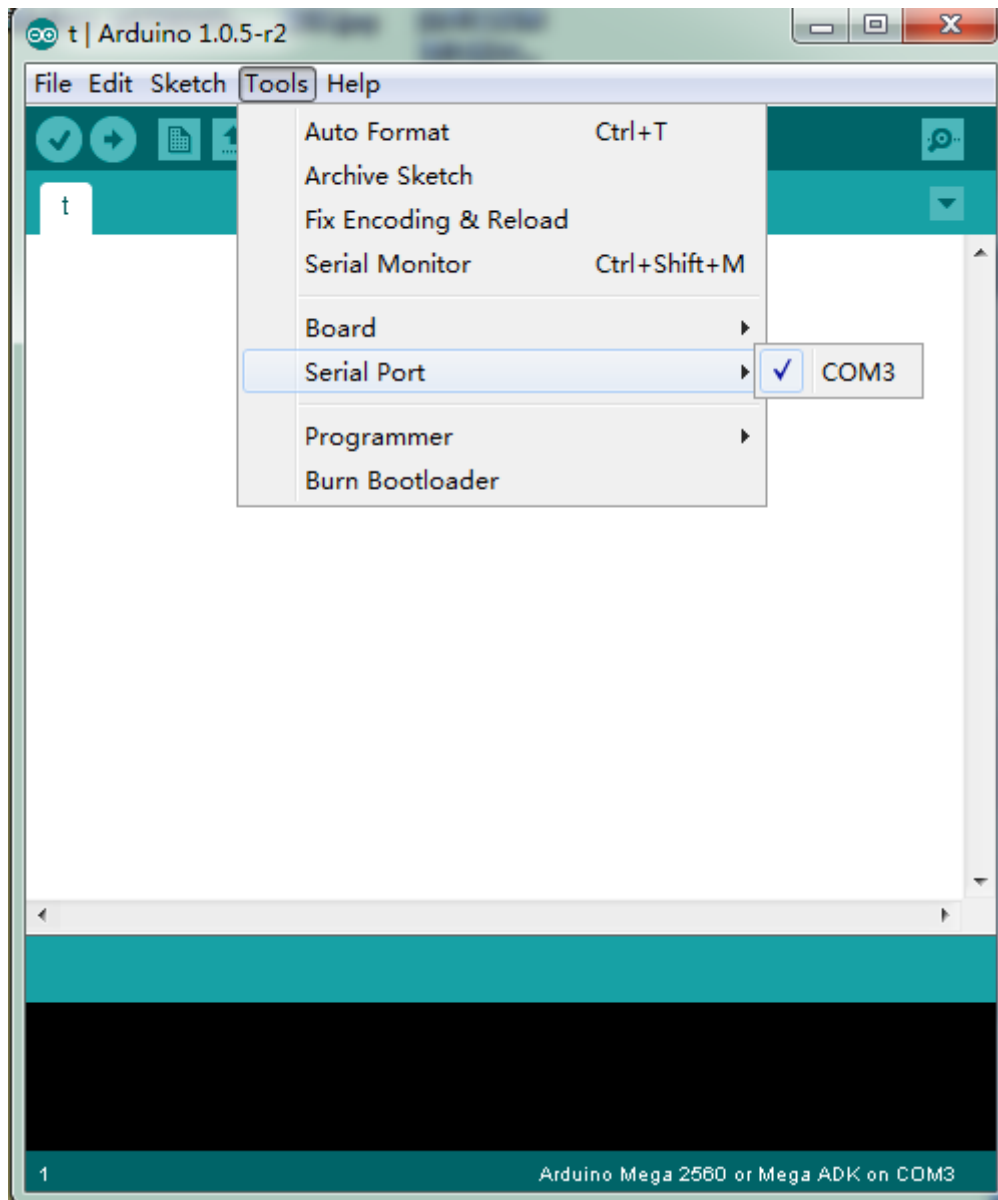


(2) 设置板子型号与串口

通过 Tools>Board>Arduino Mega 2560 or Mega ADK 设置好板子型号



通过 Tools>Serial port 选择串口



设置好以后即可进行程序烧写。

4.2、串口监视器使用——“Hello World!”

Hello world 程序：

```
void setup()
{
  Serial.begin(115200);
```

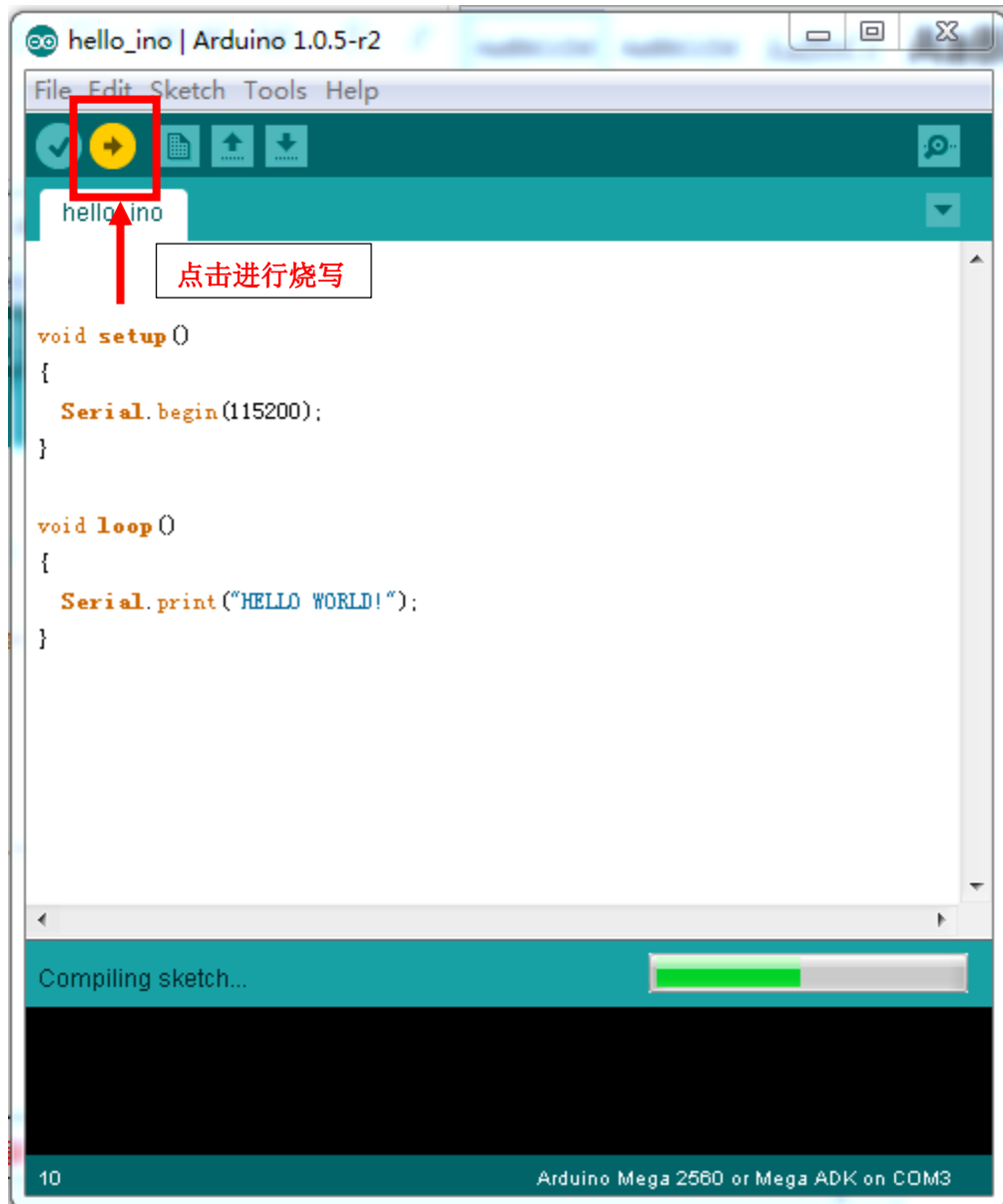
```
}  
  
void loop()  
{  
    Serial.print("HELLO WORLD!");  
}
```

将程序输入后点击左上角的v，进行编译





将程序烧写到 Arduino 板子上:



打开串口监视器，查看结果



打开串口监视器后可观察到如下现象：



4.3、外设传感器使用---点亮普通 LED

LED 程序：

```
#define LED 1                //定义 LED 的管脚

void setup()
{
    pinMode(LED,OUTPUT);    //定义 LED 管脚的模式为 OUTPUT
}

void loop()
{
    digitalWrite(LED,HIGH); //使 LED 管脚输出为高电平（HIGH）点亮 LED
}
```

4. 4、外设传感器使用——点亮双色 LED

程序:

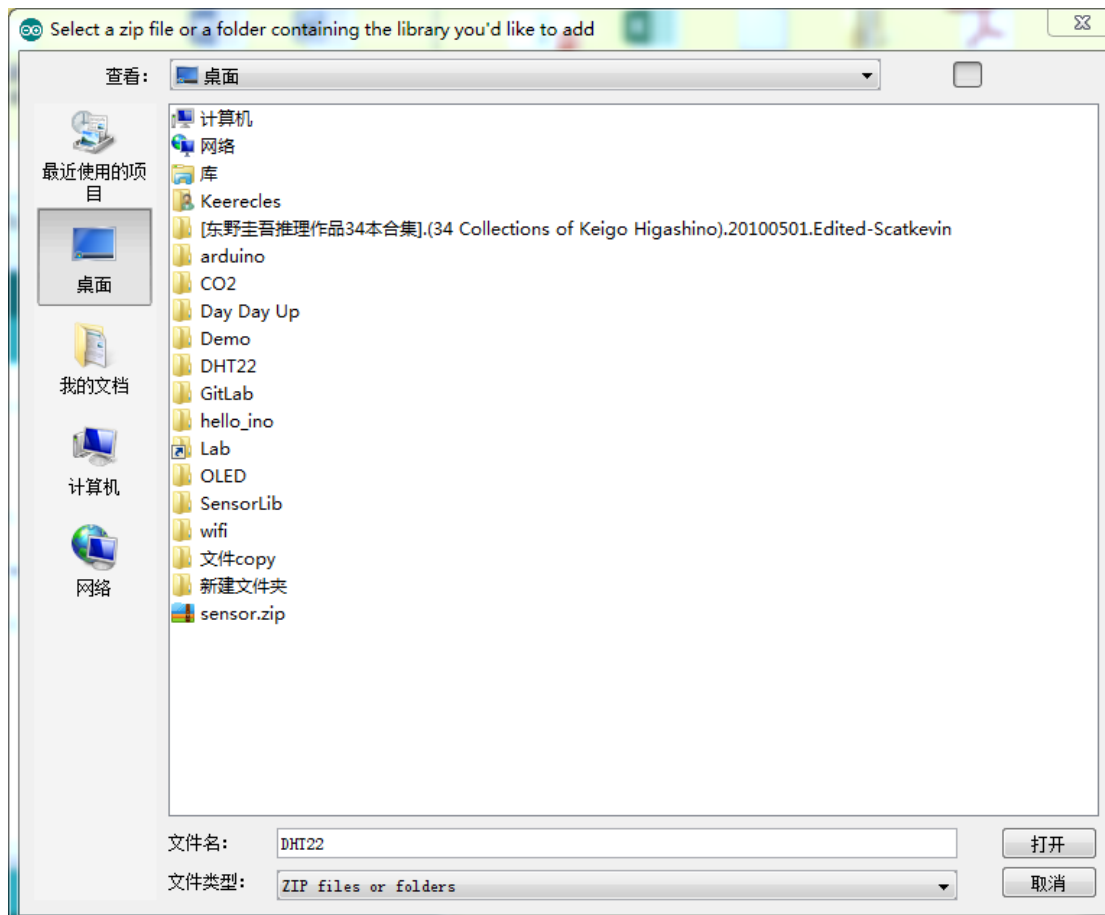
```
int redpin = 11;           //给红色 LED 设置管脚
int bluepin =10;           //给蓝色 LED 设置管脚
int greenpin =9;           //给绿色 LED 设置管脚
int val;                   //定义变量
```

```
void setup()
{
    pinMode(redpin, OUTPUT);
    pinMode(bluepin, OUTPUT);
    pinMode(greenpin, OUTPUT);
}
void loop()
{
    for(val=255; val>0; val--)
    {
        analogWrite(11, val);
        analogWrite(10, 255-val);
        analogWrite(9, 128-val);
        delay(1);
    }
    for(val=0; val<255; val++)
    {
        analogWrite(11, val);
        analogWrite(10, 255-val);
        analogWrite(9, 128-val);
        delay(1);
    }
}
```

4. 5、函数库的调用——DHT22 温湿度传感器

向 Arduino 中加入需要的库文件:

点击 Sketch>Import Library >Add Library



选中文件夹后直接点击 “打开”即可。

程序：

```
#include <dht.h>                                //调用 DHT22 温湿度传感器函数库

dht DHT;                                         //需要定义一个测量温湿度的对象 DHT

#define DHT22_PIN 7

void setup()
{
    Serial.begin(115200);                        //设置波特率
    Serial.println("DHT TEST PROGRAM ");        //打传感器信息
    Serial.print("LIBRARY VERSION: ");
    Serial.println(DHT_LIB_VERSION);
    Serial.println();
    Serial.println("Type,\tstatus,\tHumidity (%),\tTemperature (C)\tTime (us)");
}

void loop()
{
    // READ DATA
    Serial.print("DHT22, \t");
```

```

uint32_t start = micros();           //计算程序运行时间
int chk = DHT.read22(DHT22_PIN);     //读取温湿度对象的返回值进行判断
uint32_t stop = micros();

switch (chk)
{
case 0:
    Serial.print("OK,\t");
    break;
case 1:
    Serial.print("Checksum error,\t");
    break;
case 2:
    Serial.print("Time out error,\t");
    break;
default:
    Serial.print("Unknown error,\t");
    break;
}
// DISPLAY DATA
Serial.print(DHT.humidity, 1);        //读取湿度 即 DHT.humidity
Serial.print(",\t");
Serial.print(DHT.temperature, 1);     //读取温度 即 DHT.temperature
Serial.print(",\t");
Serial.print(stop - start);
Serial.println();

delay(2000);
}

```