



福昕PDF编辑器

· 永久 · 轻巧 · 自由

点击升级会员

点击批量购买



永久使用

无限制使用次数



极速轻巧

超低资源占用，告别卡顿慢



自由编辑

享受Word一样的编辑自由



扫一扫，关注公众号



哈爾濱工業大學

HARBIN INSTITUTE OF TECHNOLOGY

《机器人设计与实践》第五讲： -- Arduino 编程入门



机电工程与自动化学院 熊小刚

邮箱: xiongxg@hit.edu.cn

地址: G栋1010

- • Dr. 熊小刚
 - Email: xiongxg@hit.edu.cn
 - G1010
 - • Research interests
 - AI+Control Algorithms+Robotics
 - • Lectures:
 - Week 11-14, Tuesday, 14:00-15:45, T3501
-

- • 这门课到底要学什么？

- 学习“海量”新知识？

- 教授知识不是主要目的！
 - 锻炼传统课程中学不到的能力

- 转变观念，突破舒适区，新的学习方式：Project-based Learning

- (1) 自学能力
 - (2) 沟通能力
 - (3) 规划能力

- (4) 信息检索能力/工具使用能力：gitee/github，百度

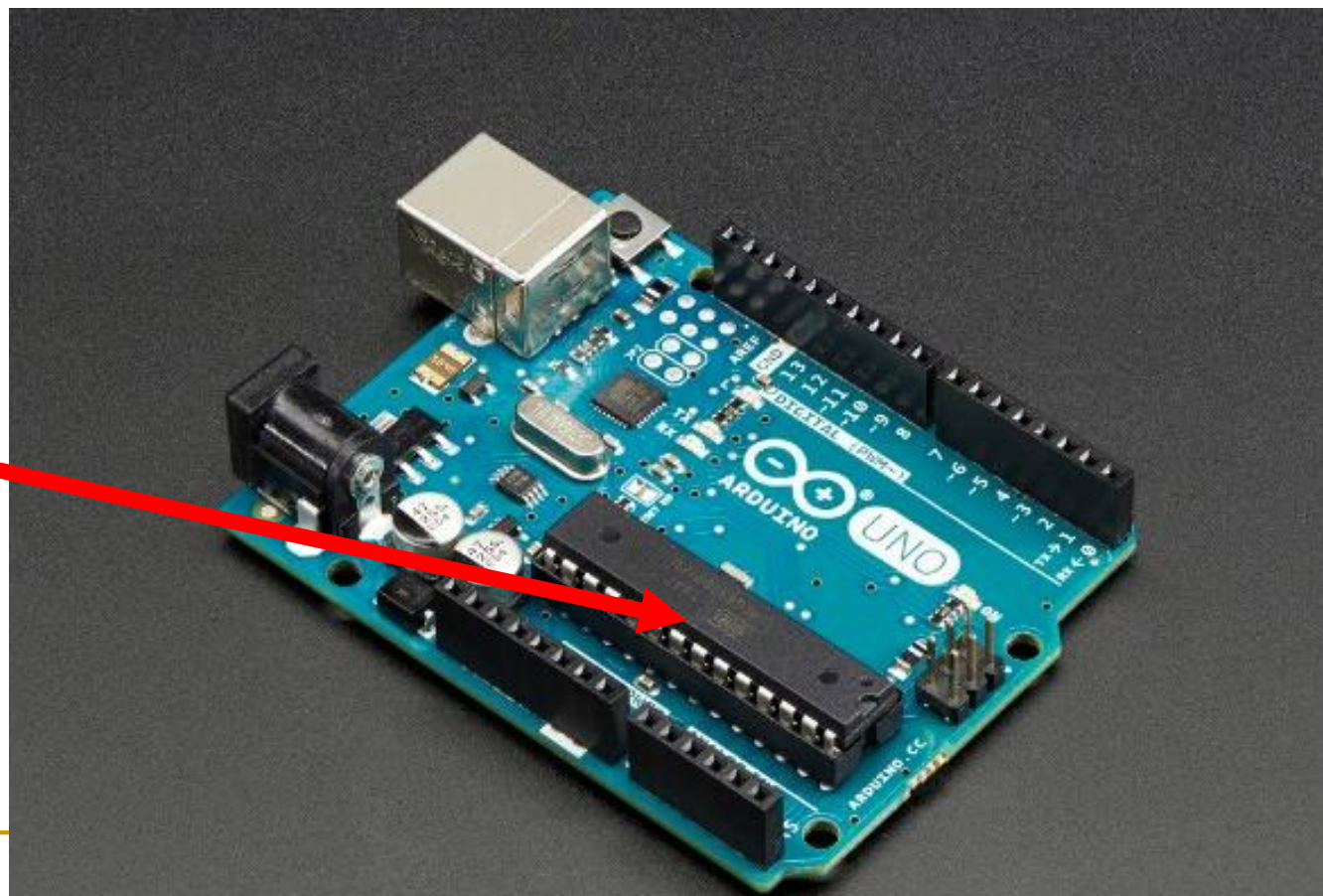


Arduino是什么？

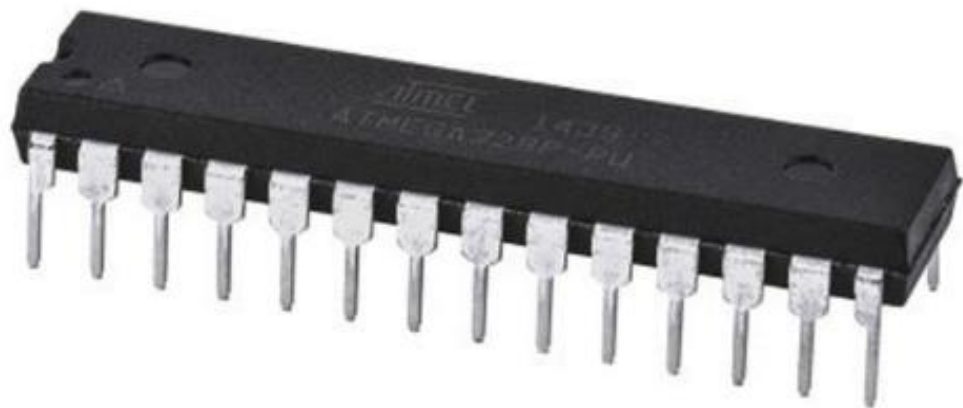


1、什么是Arduino

- 要了解Arduino就先要了解什么是单片机
- 最早的Arduino板卡核心就是一种单片机。



2、什么是单片机



- 简单
计算功能、
存储功能、
外围接口；
- 低成本；
- 体积小。

3、单片机有什么用？

■ 以下哪些应用合适使用单片机？

小型机器人玩具

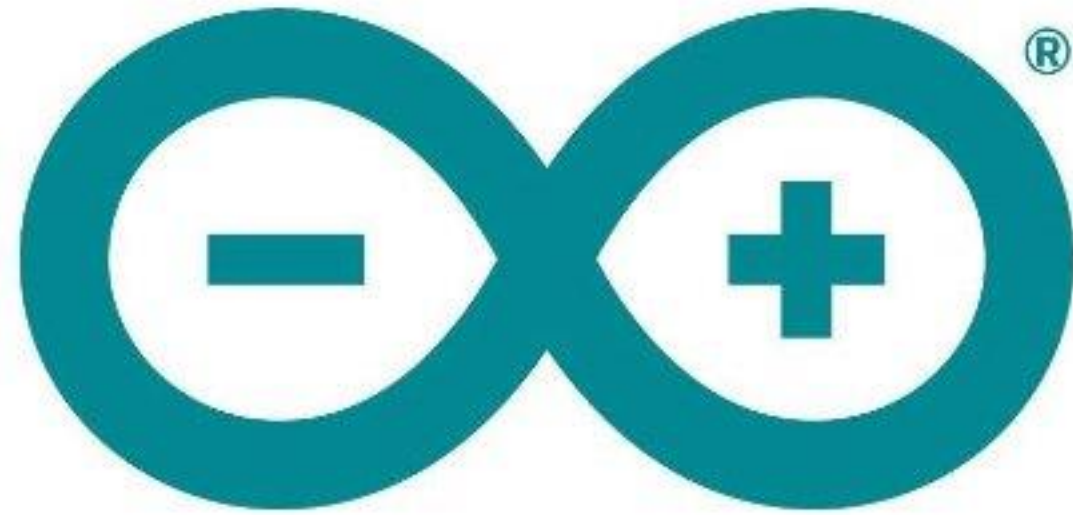
人脸~~X~~识别

智能网络~~X~~电视

电子琴

智能空调

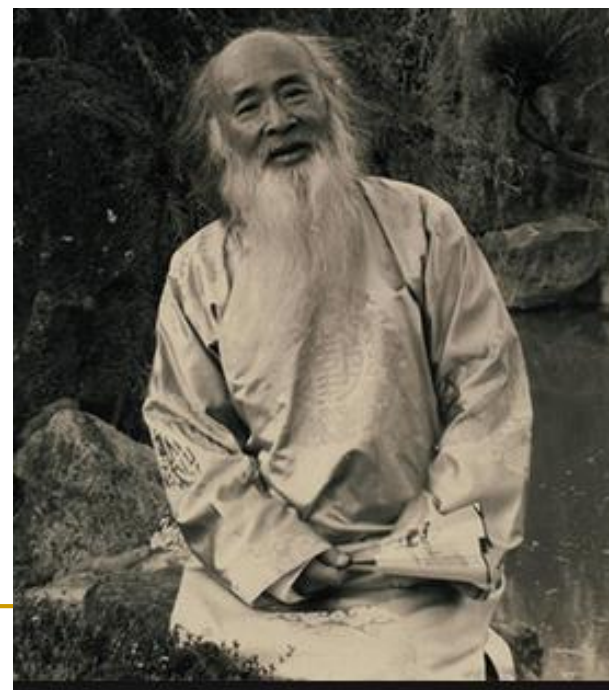
智能~~X~~手机



ARDUINO

■ Arduino 不仅仅是一个开源硬件平台，更是一个优秀的硬件开发平台，更是硬件开发的趋势。

- **Arduino**简单的开发方式使得开发者更关注创意与实现，更快的完成自己的项目开发，大大节约了学习的成本，缩短了开发的周期。





这个最经典的开源硬件项目，诞生于2005意大利的一间设计学校。Arduino的核心开发团队成员包括：

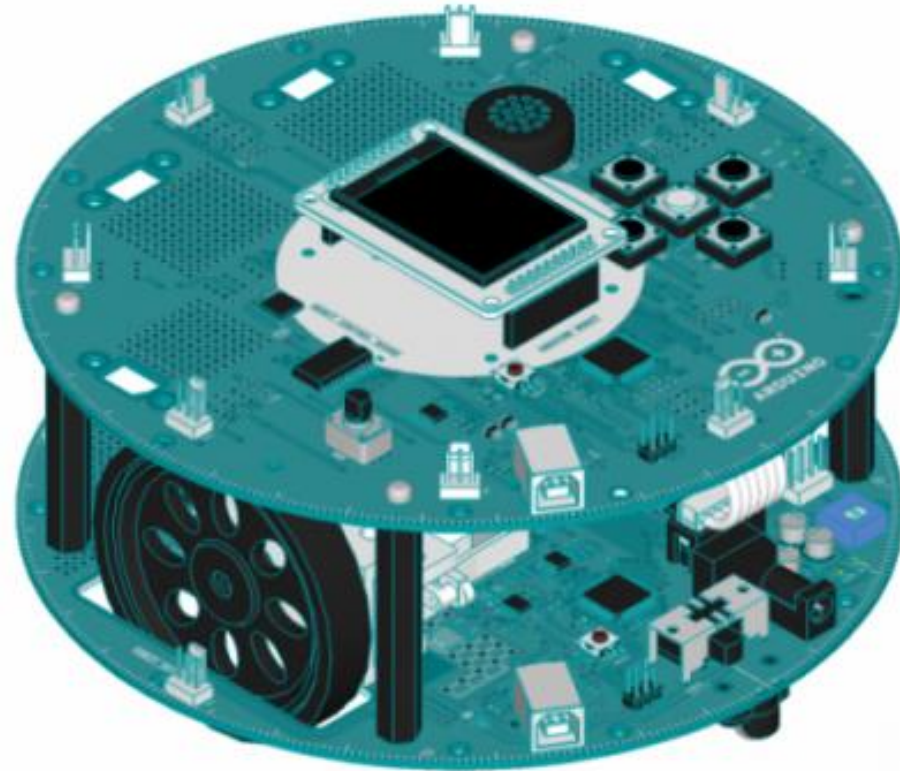
马西莫·班齐（Massimo Banzi）、大卫·奎提耶斯（David Cuartielles）、汤姆·伊果（Tom Igoe）、赞布罗塔·马提诺（Gianluca Martino）、大卫·梅利斯（David Mellis）和尼可拉斯·兰比提（Nicholas Zambetti）。

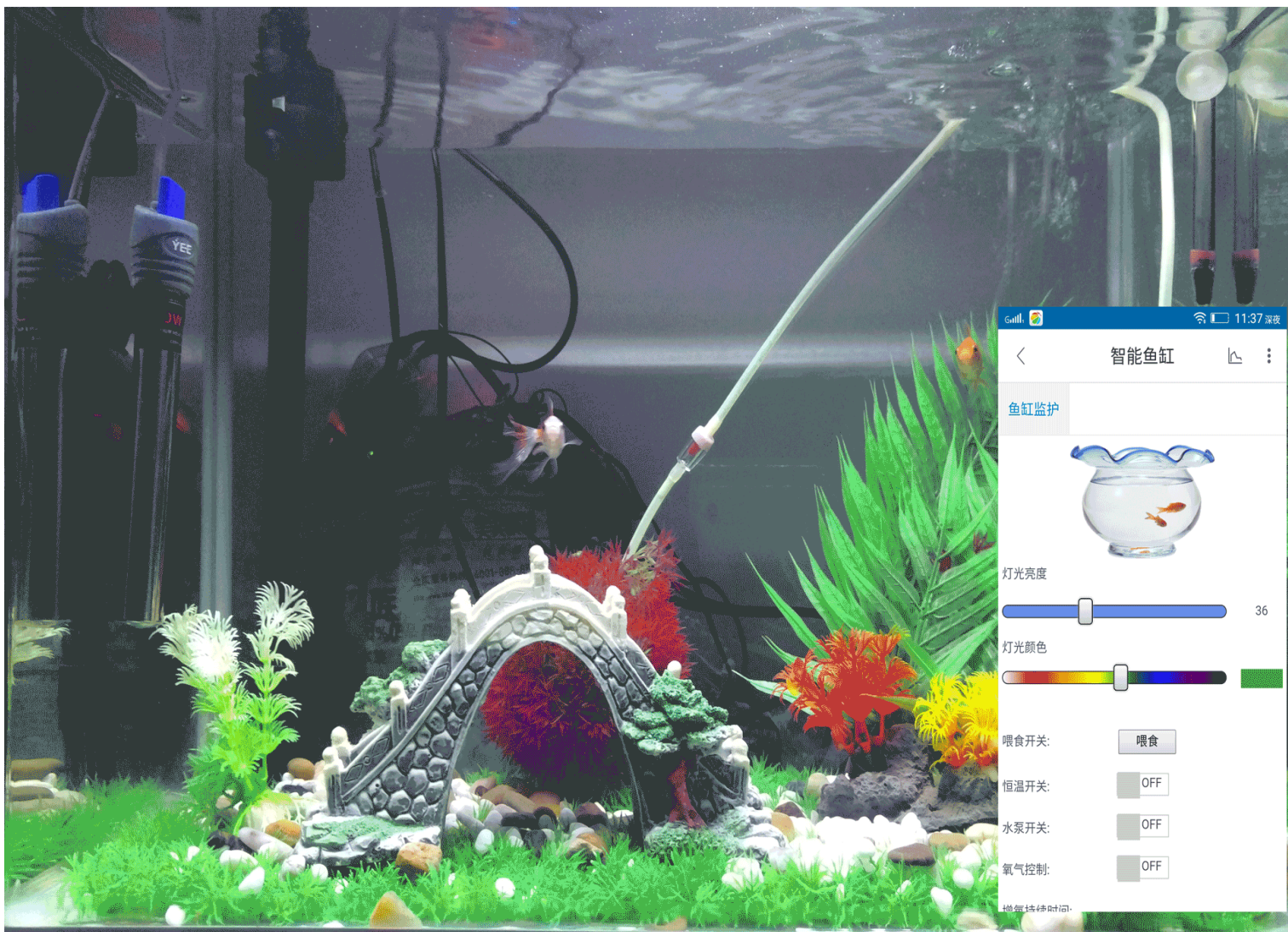


据说马西莫·班齐之前是高科技设计学校的老师，他的学生们经常抱怨找不到便宜好用的单片机，2005年冬天，Massimo Banzì跟朋友David Cuartielles讨论了这个问题，David Cuartielles是一个西班牙籍晶片工程师，当时在这所学校做访问学者。

两人决定设计自己的电路板，并引入了Banzì的学生David Mellis为电路板设计编程语言。两天以后，David Mellis就写出了程式码。又过了三天，电路板就完工了。这块电路板被命名为Arduino。

Arduino



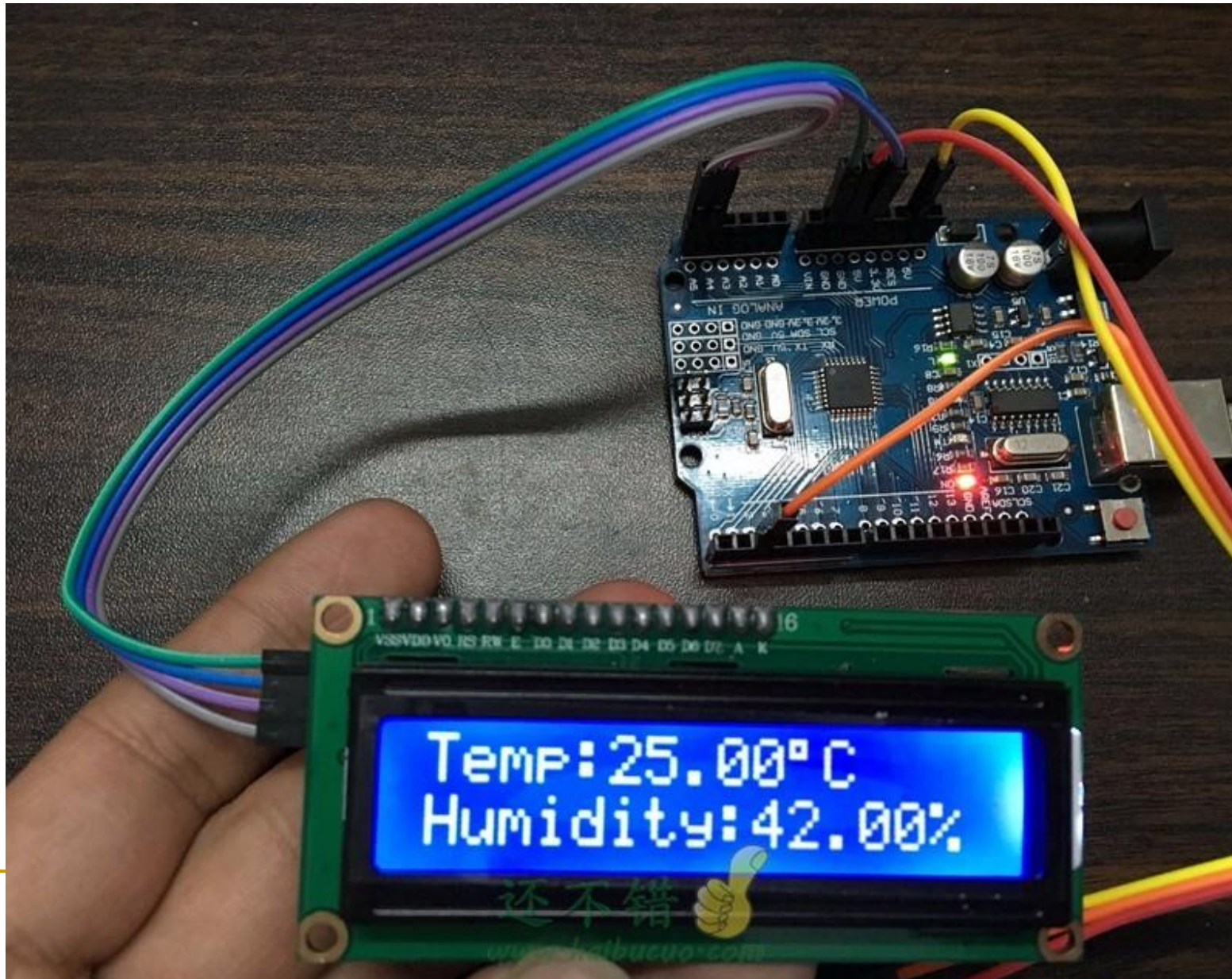


智能鱼缸

解决没有时间照料鱼，或者更好地饲养热带鱼，智能鱼缸满足一些水族等特殊生存条件。

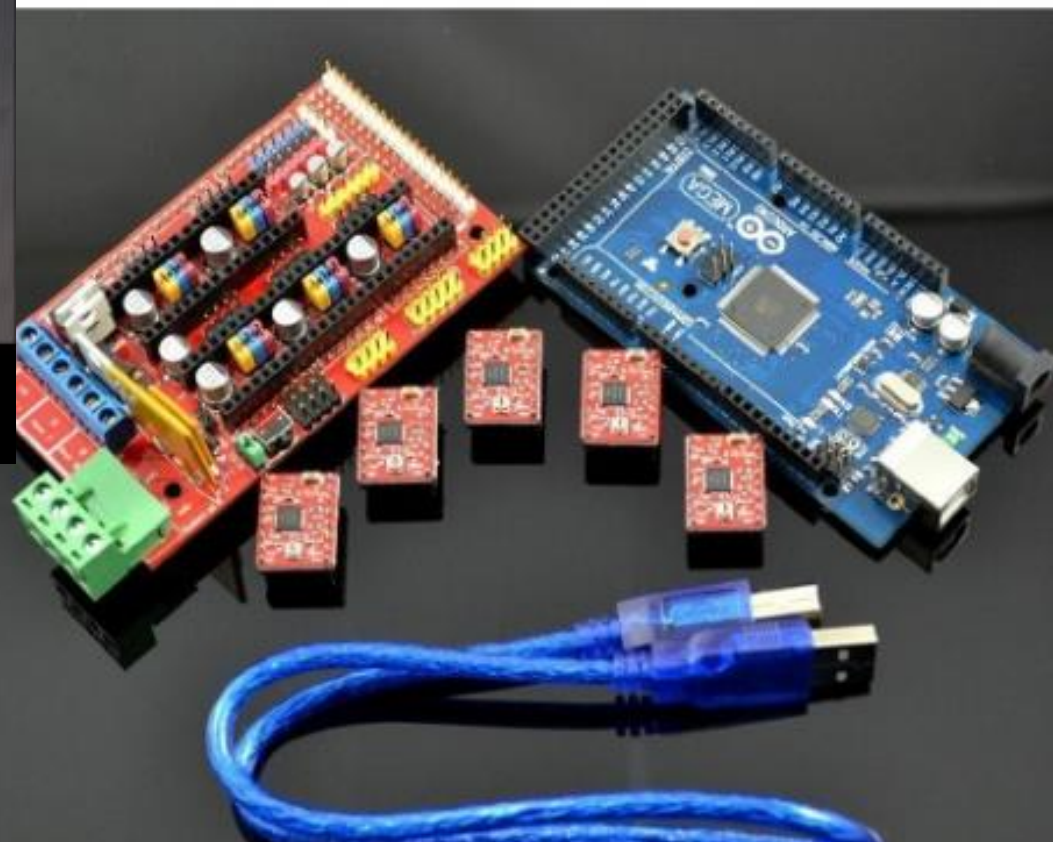
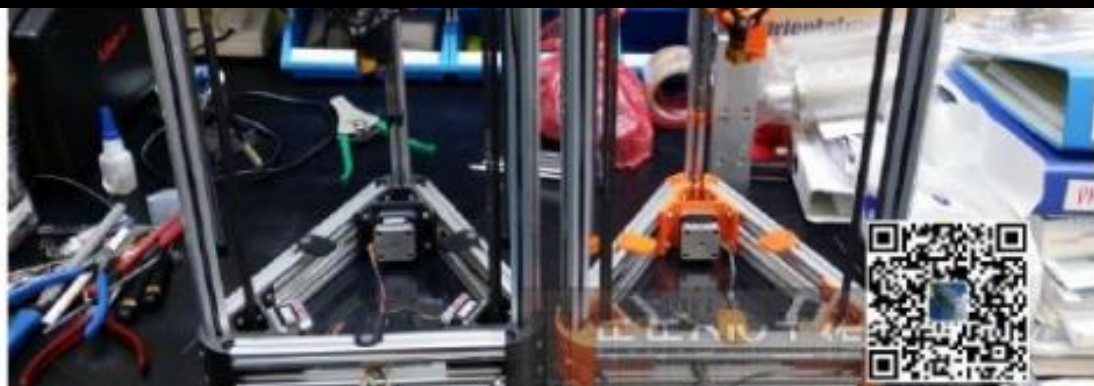
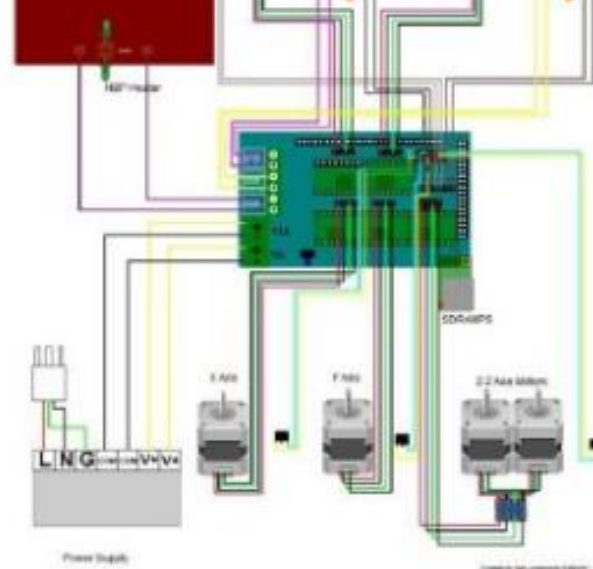
温湿度计

温湿度计可以检测所处环境的空气温度和空气湿度。

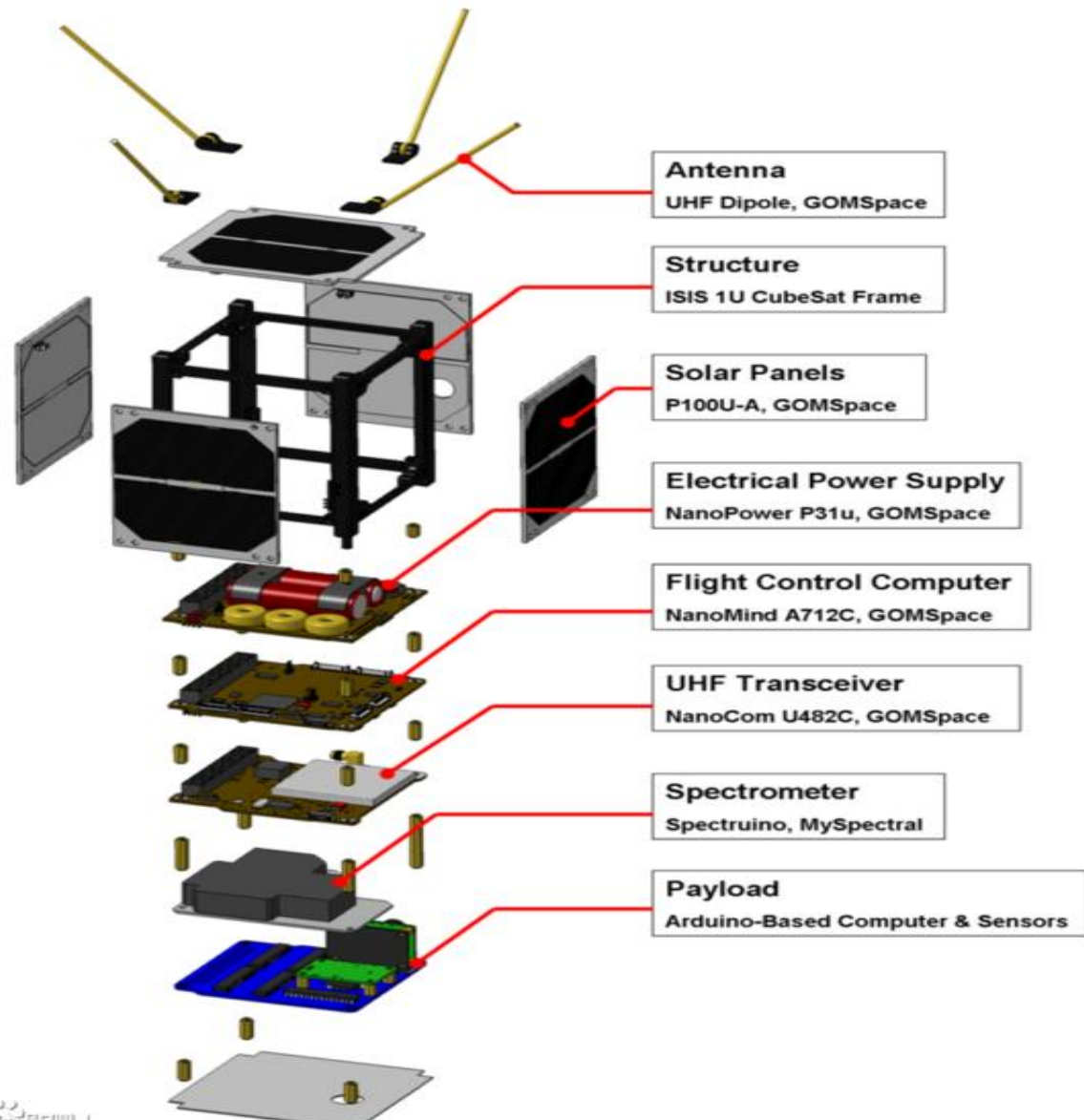
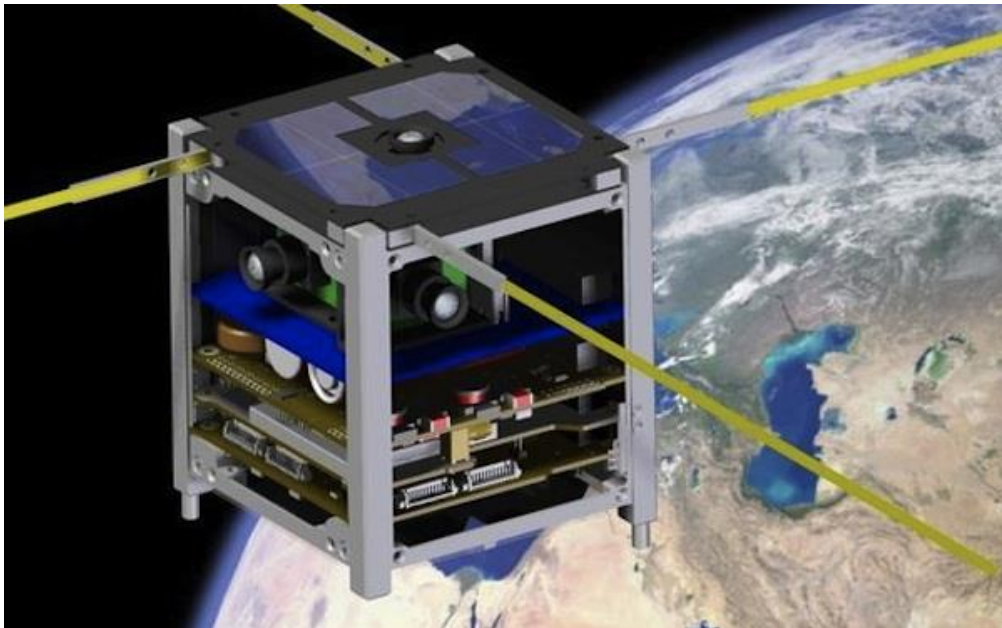




接线图



最烧钱的arduino项目：ArduSat开源卫星



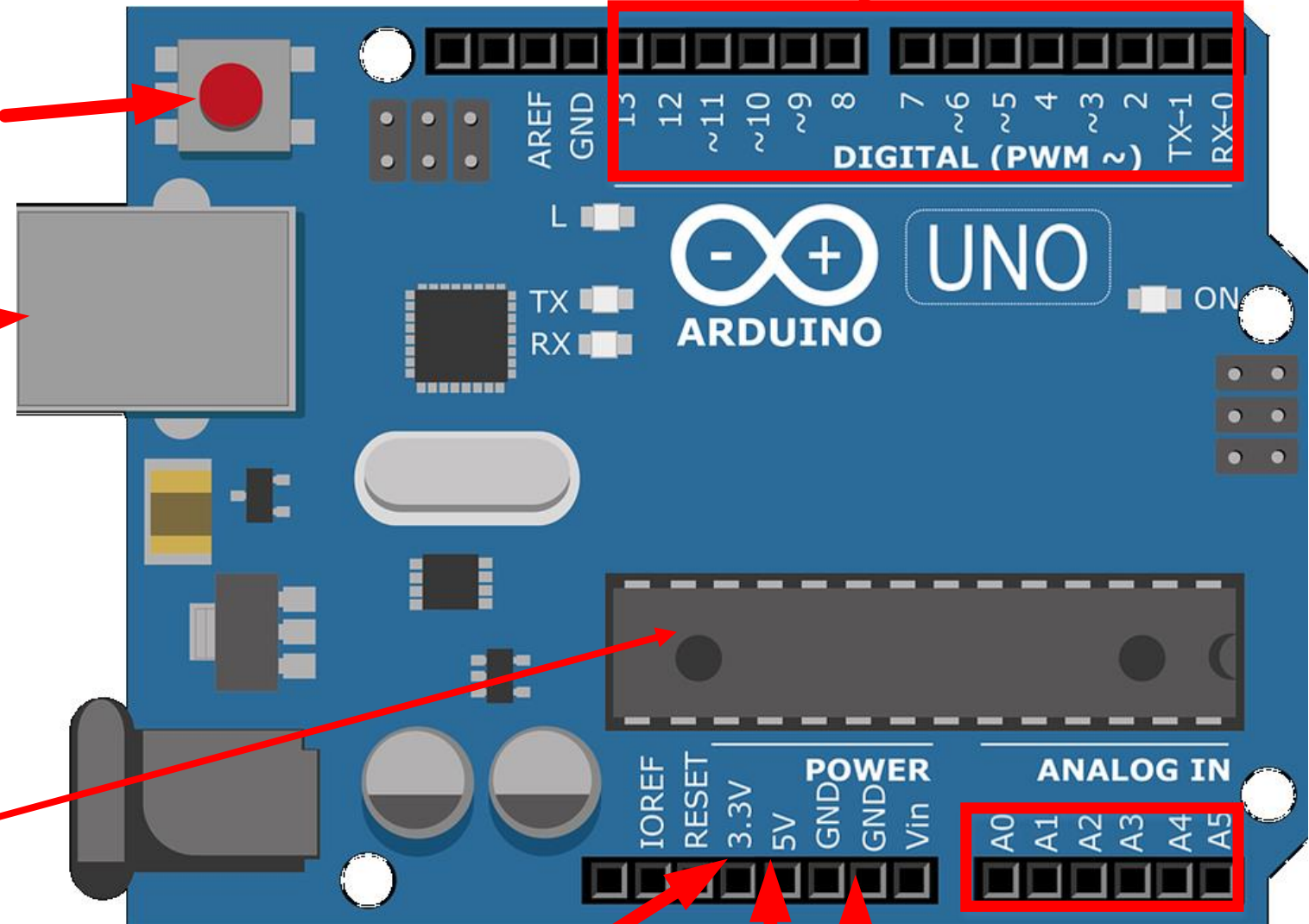
Arduino如何使用？

数字输入/输出口

复位按钮

USB接口

单片机



3.3V

5V

地

模拟输入口

基本代码框架

- void setup()

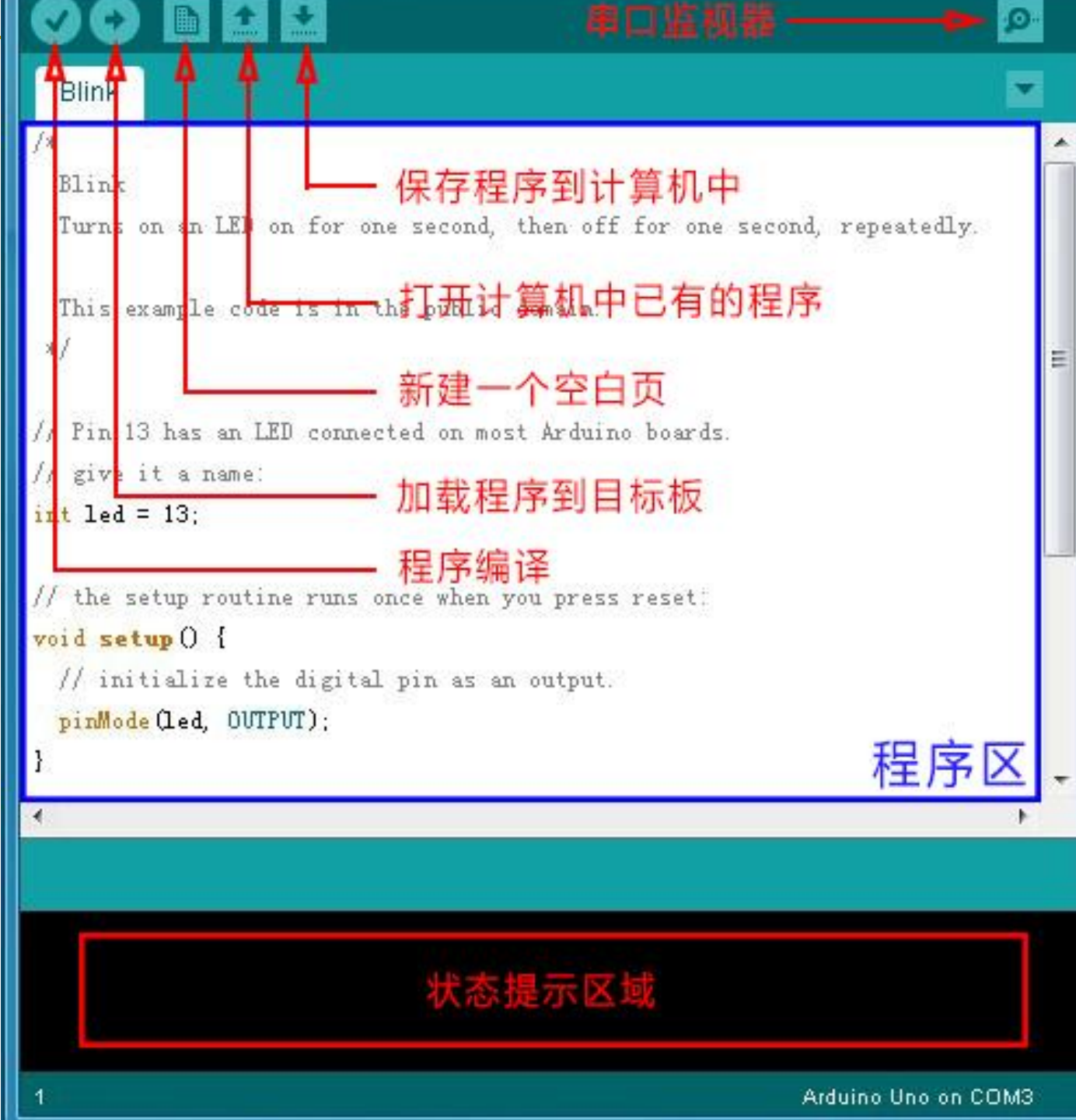
- {

- }

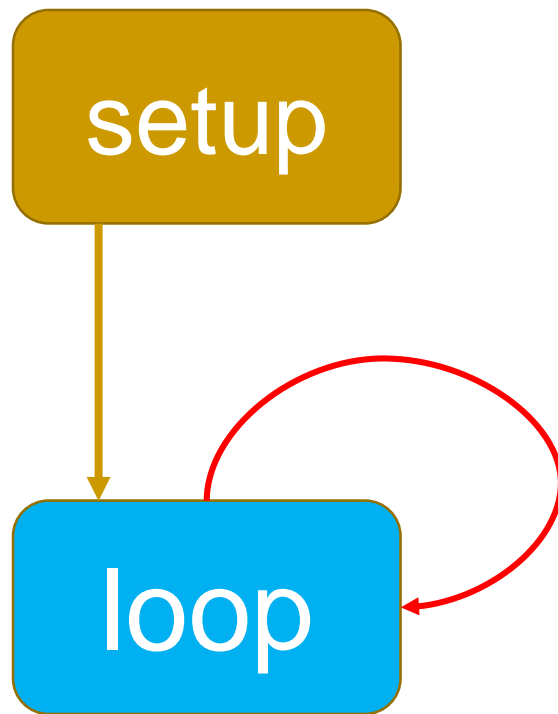
- void loop()

- {

- }



```
void setup()  
{  
}  
void loop()  
{  
}
```



执行一次

setup()之后
循环执行

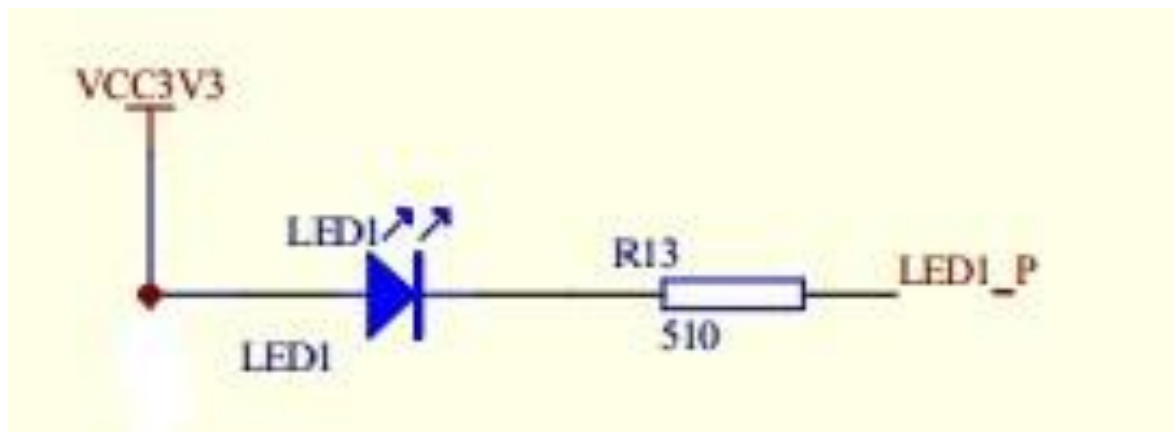
I/O □ (Input/Output)

Arduino具有I/O口引脚，具备输入输出功能：

输出功能，就是把引脚变高和变低。

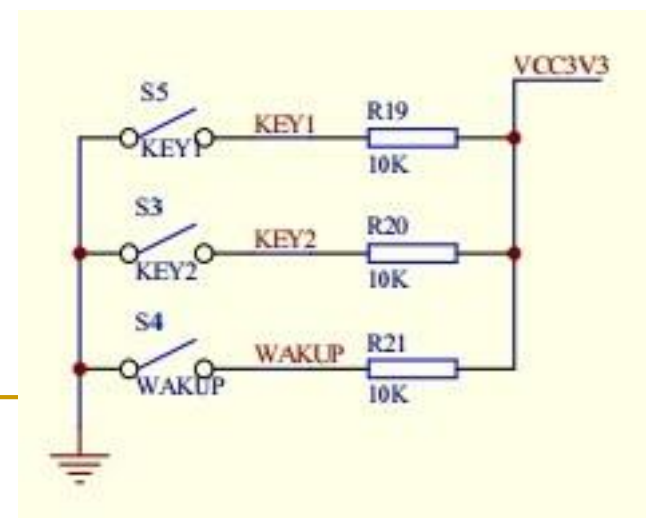
输入功能，就是检测引脚上的电平是高电平还是低电平（数字）。

输入功能，就是检测引脚上的电压（数字）。



I/O H.

输入功能什么时候用？

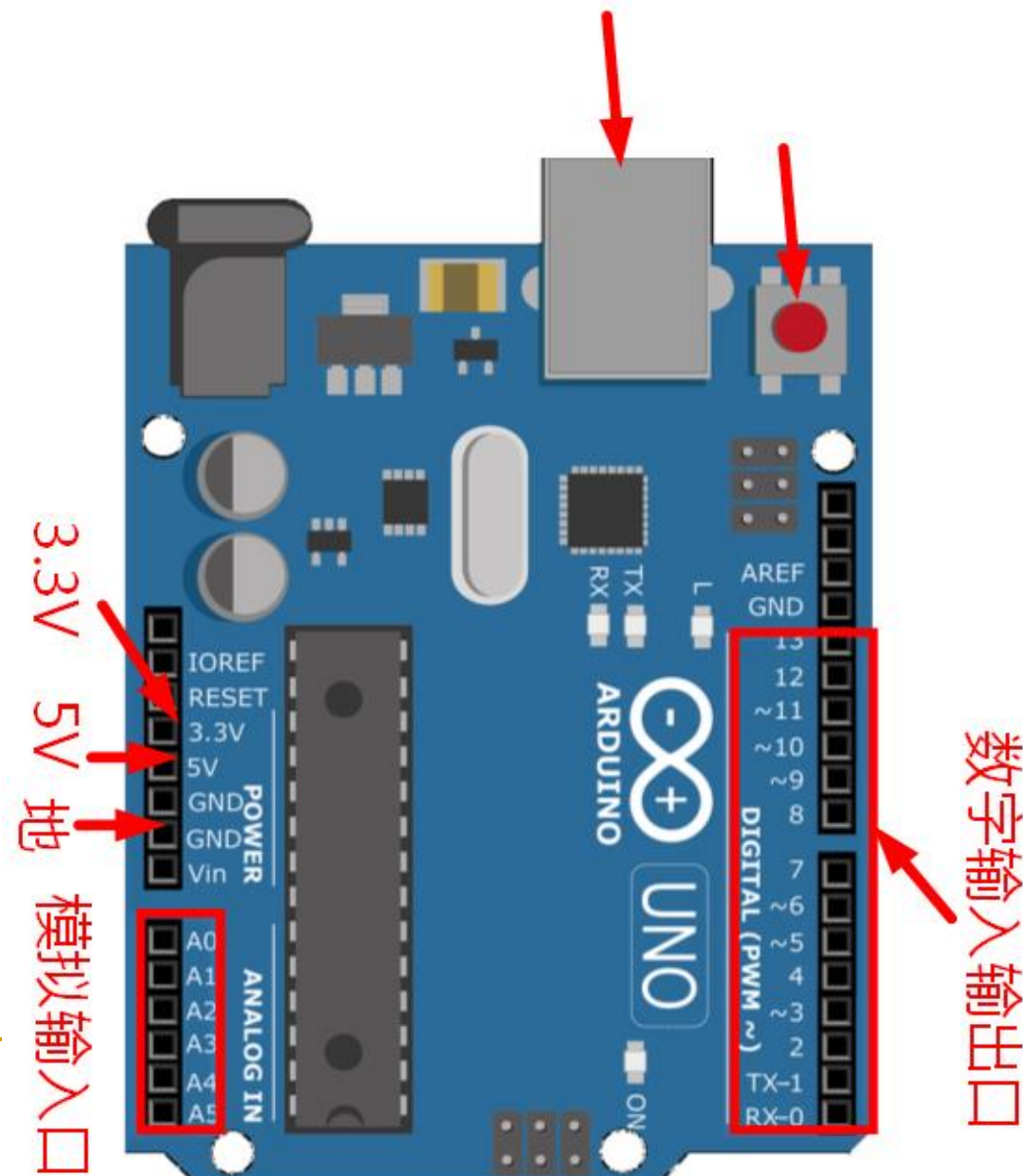


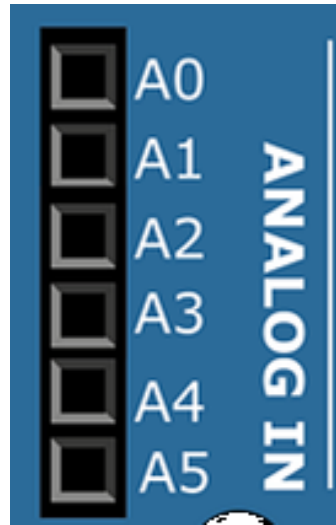
I/O口 (Input/Output)

- 数字I/O口
- 模拟I/O口

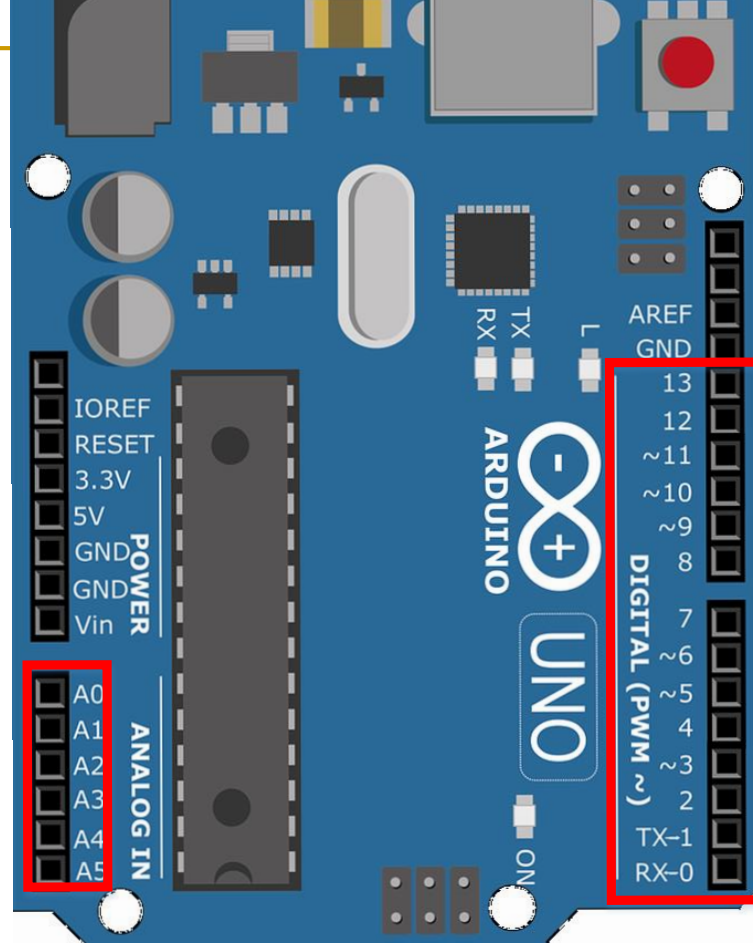
一、数字I/O口的操作函数

- 1、pinMode(pin,mode);
模式：输入或者输出
- 2、digitalWrite(pin,value);
输出
- 3、digitalRead(pin);
输入





模拟输入口



数字输入输出口

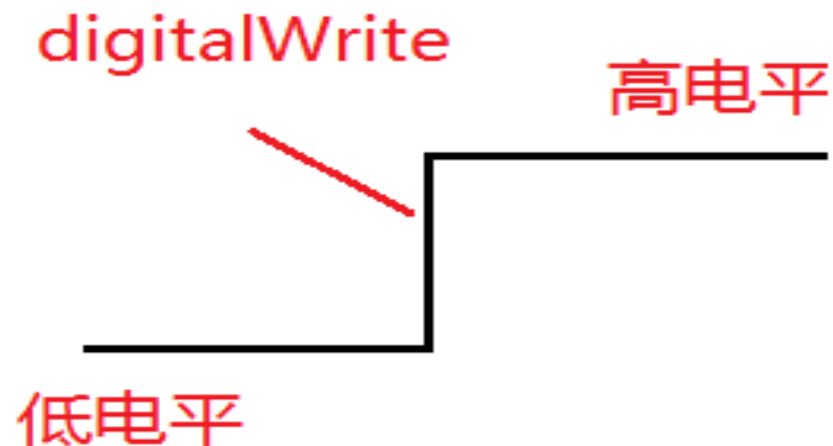
- PIN的范围是数字引脚0-13,
- 也可以把模拟引脚 (A0-A5) 作为数字引脚使用, 此时编号为14脚对应模拟引脚A0,19脚对应模拟引脚A5。

1 `pinMode(pin, mode)` 配置引脚与输出或输入模式，无返回值函数。

有两个参数：

- `pin` 参数表示要配置的引脚；
 - `mode` 参数表示设置的参数 **INPUT**（输入）和 **OUTPUT**（输出）。
-
- **INPUT** 参数用于读取信号，**OUTPUT** 用于输出控制信号。
 - 例子： `pinMode(1, OUTPUT);`

- **2、digitalWrite(pin, value)**
设置引脚的输出电压为高电平或低电平，无返回值的函数。
 - pin 参数表示所要设置的引脚，
 - value 参数表示输出的电压**HIGH**(高电平)或**LOW**(低电平)。
- 注意：使用前必须先用pinMode设置。
- 例子： `digitalWrite(1, HIGH);`



3、digitalRead(pin)

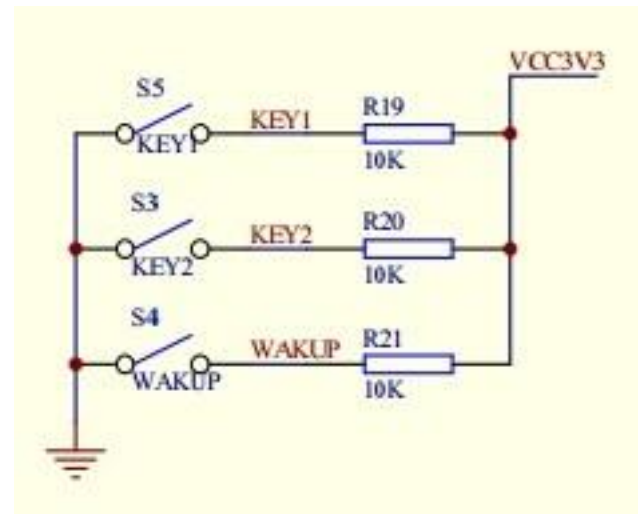
■ 获取引脚的电压情况HIGH（高电平）或者LOW(低电平)。

- pin参数表示所要读取的引脚

- 返回值：HIGH（1） 或 LOW（0）

■ 注意：使用前必须先用pinMode设置。

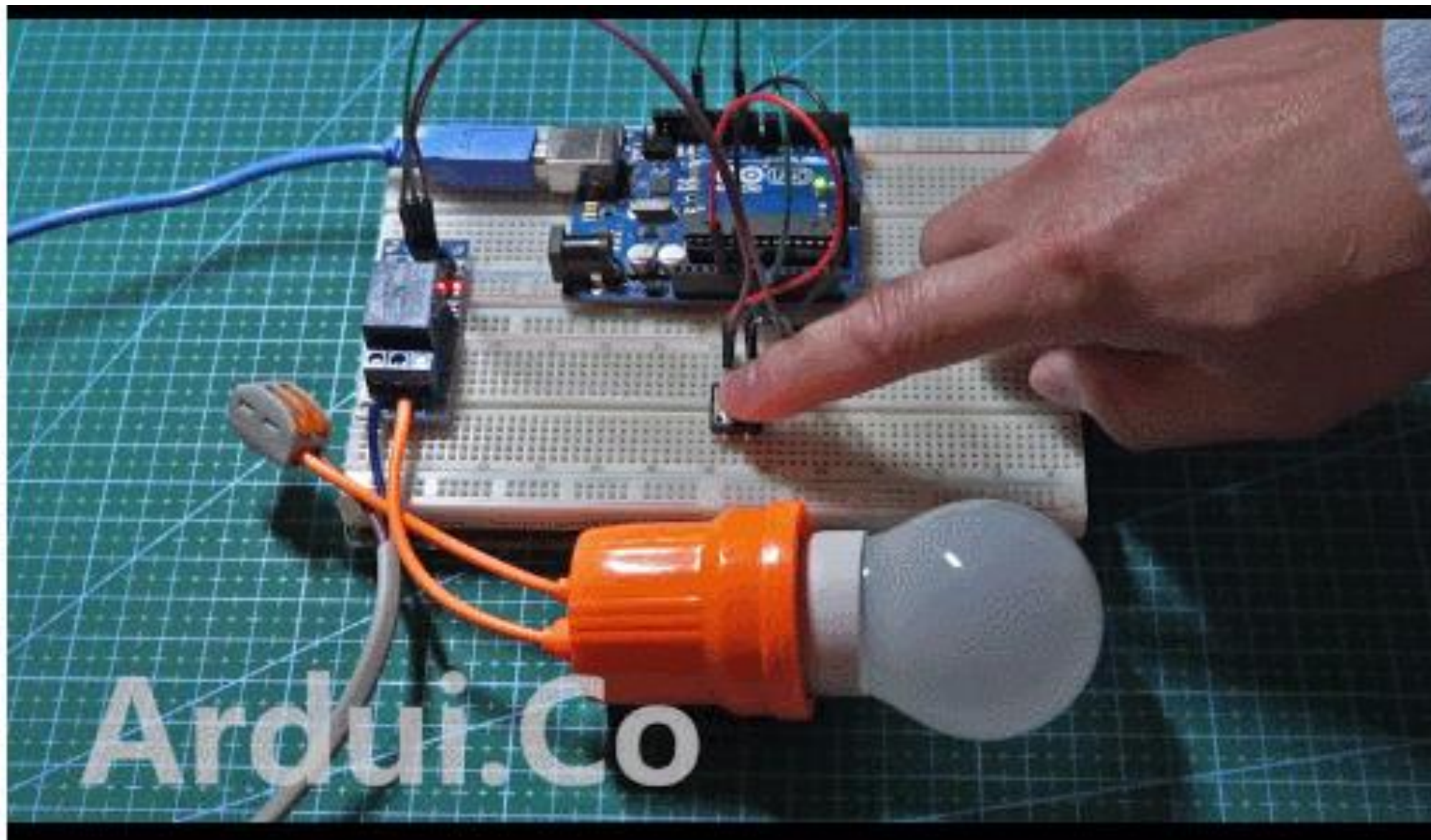
■ 例子：int a = digitalRead(1);



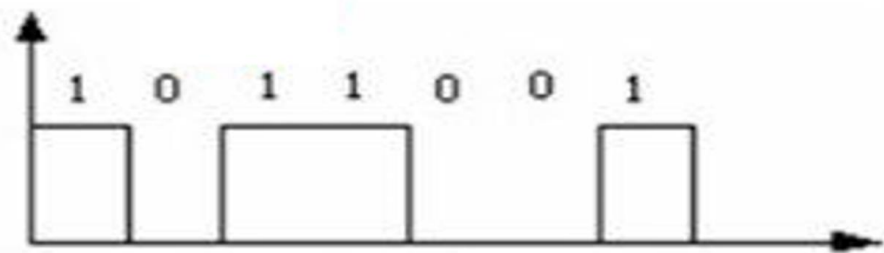
例程

```
01.  int button = 9;    //设置第9脚为按钮输入引脚
02.  int LED = 13;     //设置第13脚为输出，接到灯泡模块
03.
04.  void setup()
05.  {
06.      pinMode(button, INPUT); //设置为输入
07.      pinMode(LED, OUTPUT);  //设置为输出
08.  }
09.  void loop()
10.  {
11.      if (digitalRead(button) == LOW)           //如果读取高电平
12.          digitalWrite(LED, HIGH);             //13脚输出高电平
13.      else
14.          digitalWrite(LED, LOW);              //否则输出低电平
15.  }
```

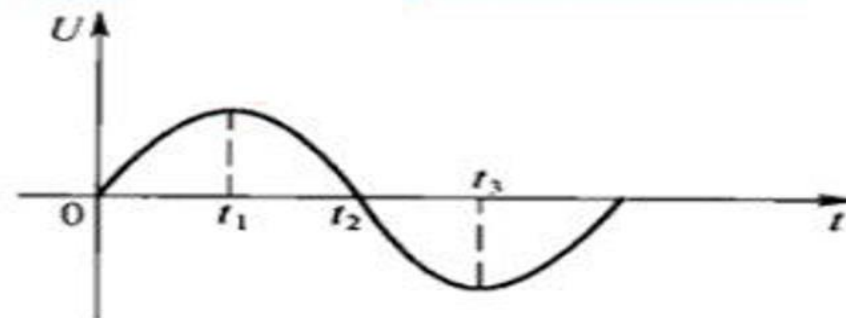

数字 I/O 口操作范围内。



突变：高低电平的转换瞬间完成
保持：一段时间维持低或高电平
幅值和时间都是离散的



在 $0-t_1$ 时间，信号电压慢慢升高
在 t_1-t_2 时间，信号电压慢慢减小
变化是连续



大气温度
声音强度
距离远近
等

传感器

大气温度
声音强度
距离远近
等

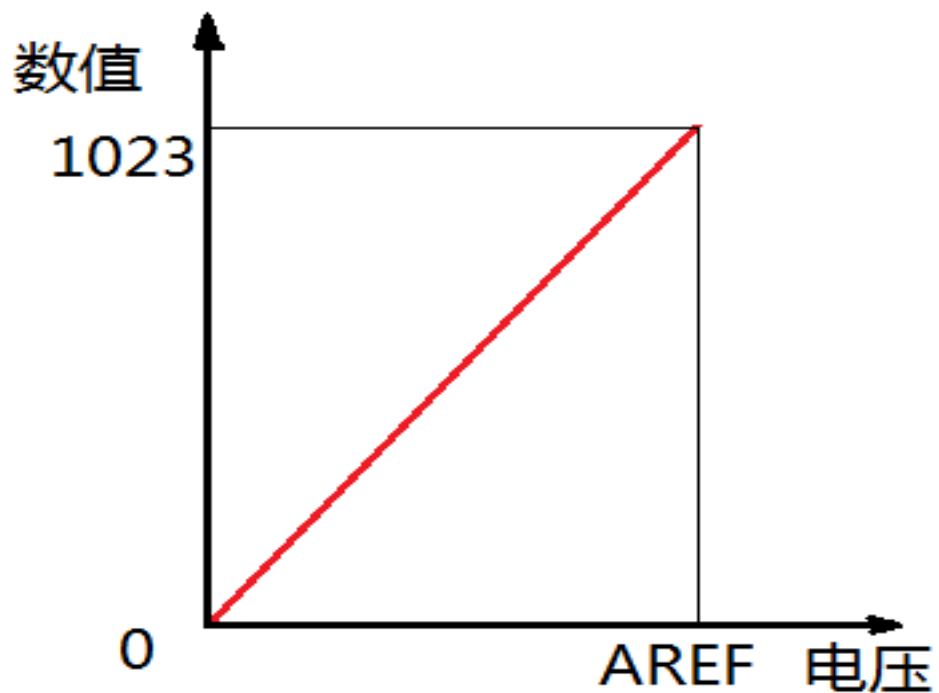
传感器



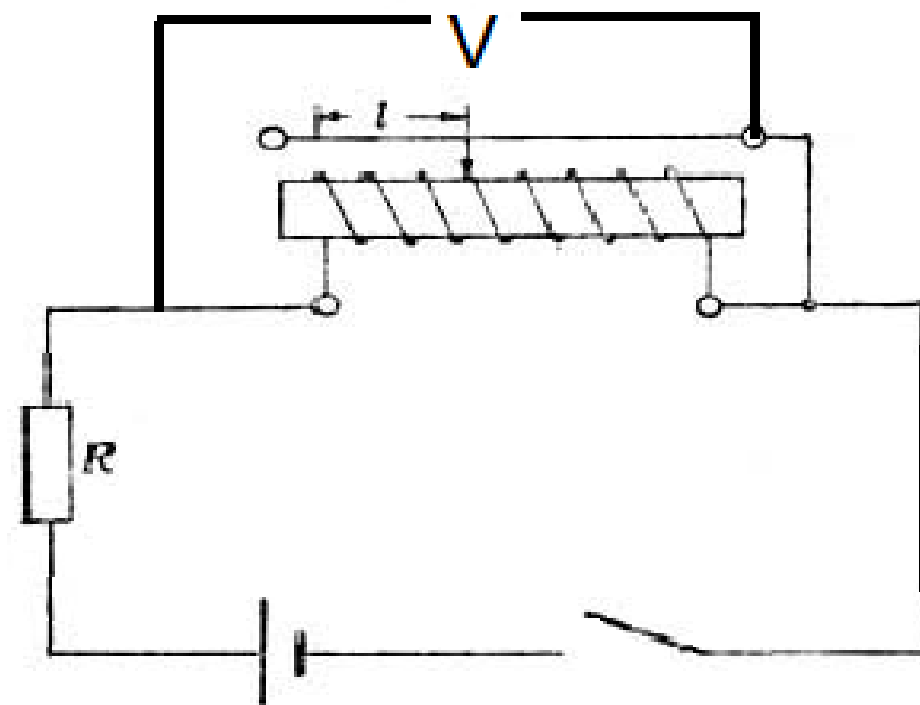
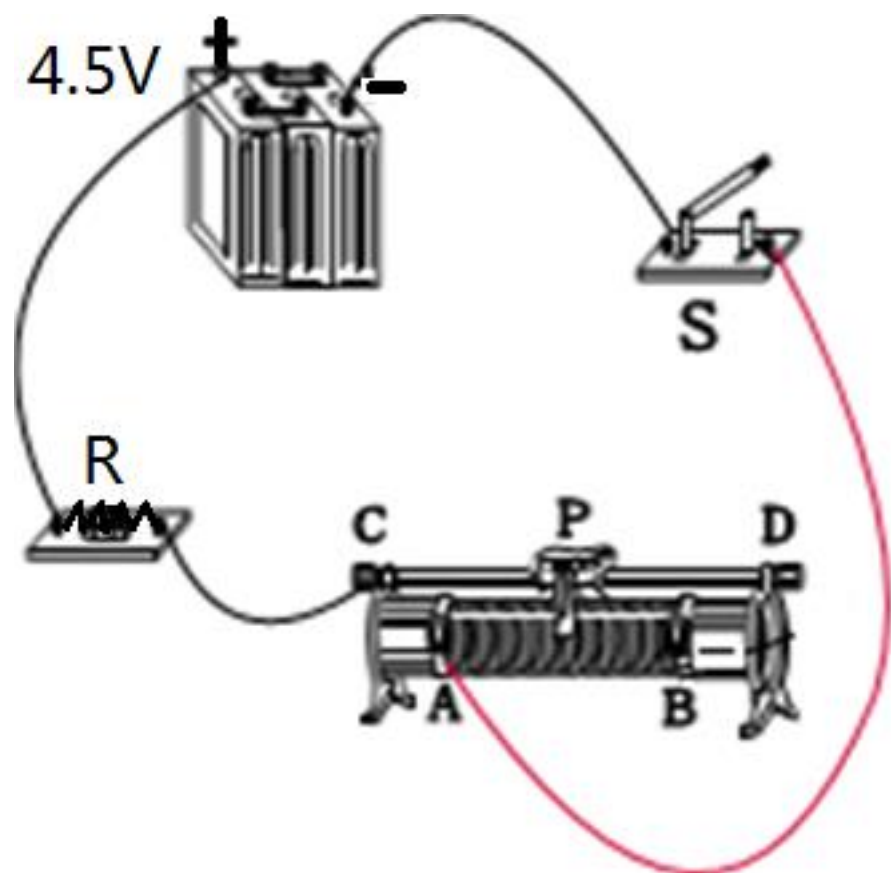
模拟电压信号
 $0V \sim AREF$



$0 \sim 1023$



线性关系



$$R_l = \frac{V}{V_b - V} R$$

例子：

1. 参考电压： $A_{REF} \leq 5V$ 如： $+5V$

被测电压范围： $0 \sim 4.5V$

经过Arduino后，最大数字： $1023 * 4.5 / 5 = 920$

电压测量分辨率： $5V / 1023 = 0.004887V$

3. 被测电压范围
： $0 \sim 12.0V$?

怎么办？

二、模拟I/O口的操作函数

- 1、analogReference(type);

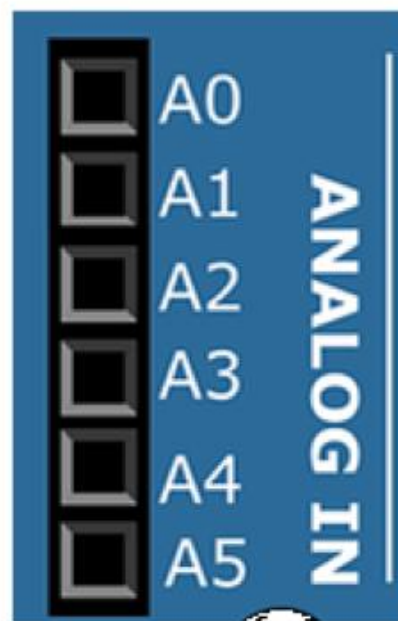
设置参考电压

- 2、analogRead(pin);

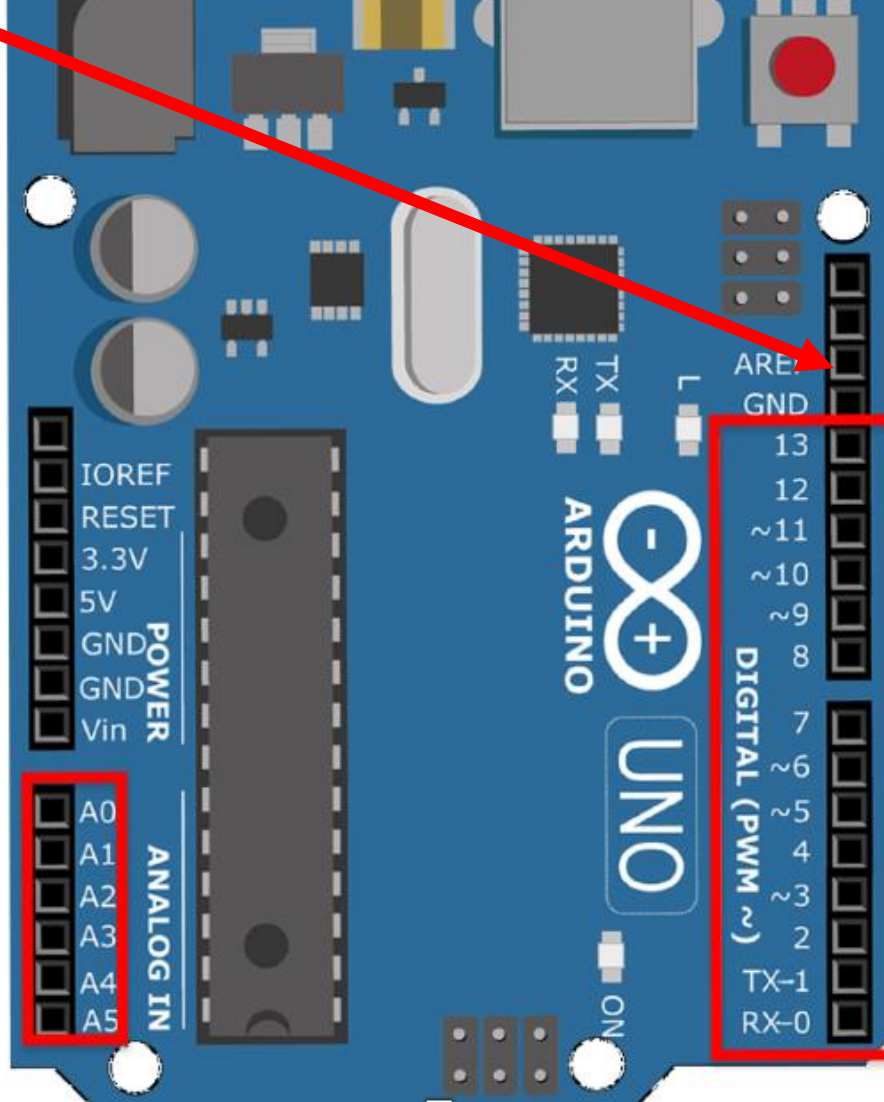
读取模拟电压值

- 3、analogWrite(pin,value);

输出模拟电压值



模拟输入口



- 共有6个模拟引脚输入（A0-A5）

- # 1、analogReference(type)
- 配置模拟引脚的参考电压。有3种类型。
 - **DEFAULT**:默认值，参考电压是5V.
 - **INTERNAL**:低电压模式，使用片内基准电压源2.56V
 - **EXTERNAL**:扩展模式,通过AREF引脚获取参考电压
 - 注意：不使用本函数的话，默认是参考电压5V。如果使用AREF接参考电压，需接个5K欧的上拉电阻。

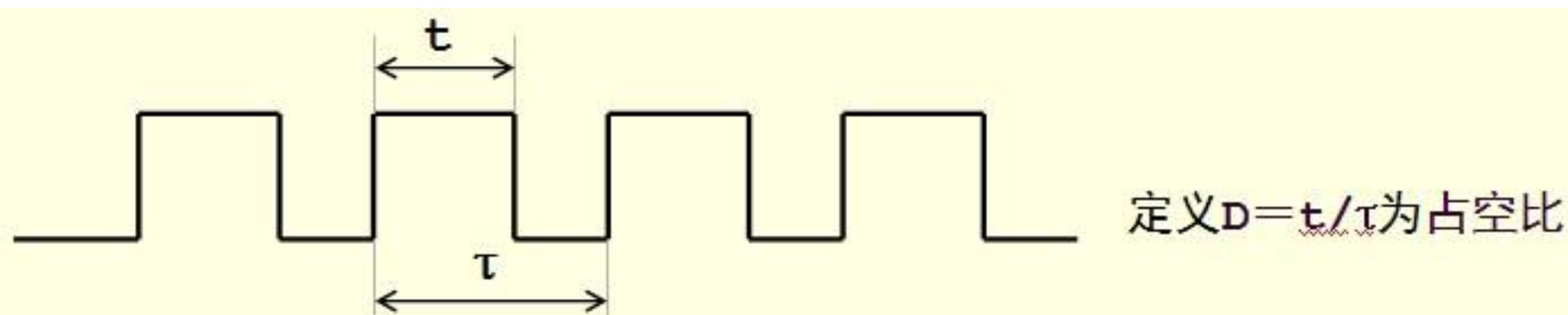
2、analogRead(pin)

- 读取引脚的模拟量电压值，每读取一次需要花100US的时间。
(即不是瞬间完成的)
 - pin参数表示所要获取模拟量电压值的引脚， 返回为int型。
精度10位，返回值从0~1023。
- 注意： 函数参数的pin范围是0~5， 对应板上的模拟口A0~A5。

3、analogWrite(pin,value)

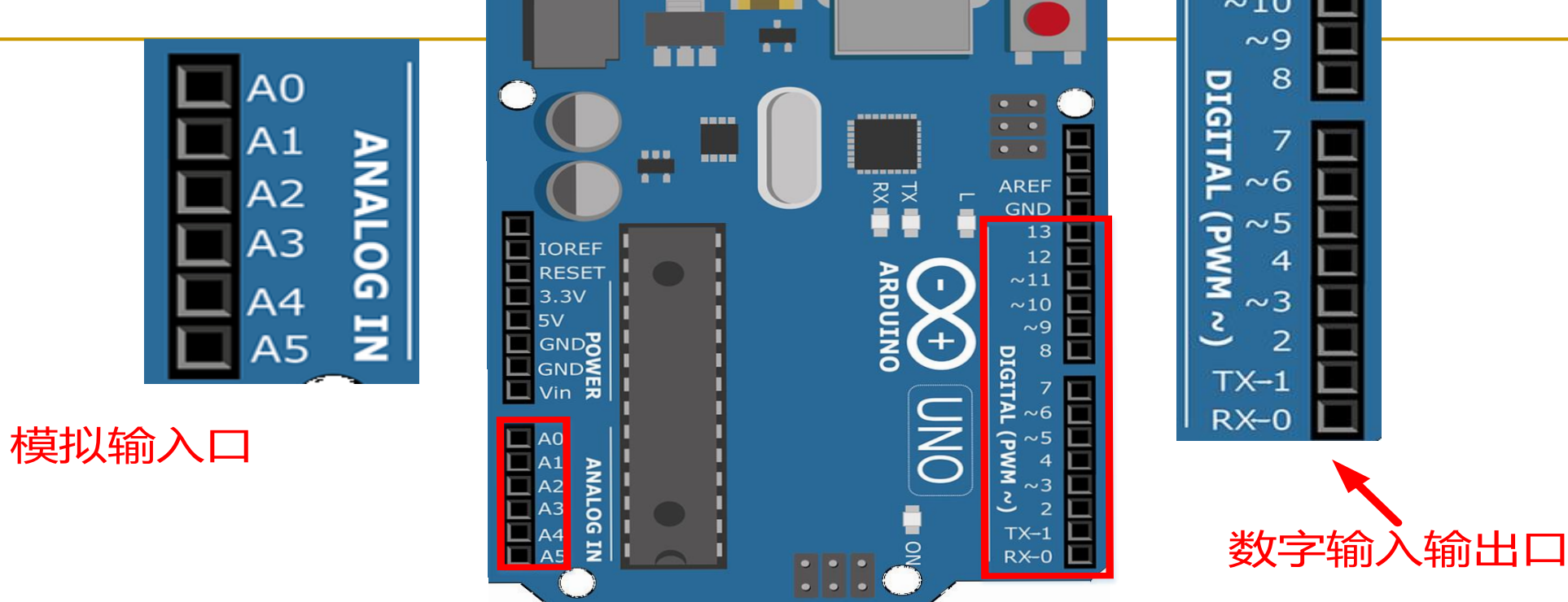
(注：不是所有引脚都适用)

- 该函数是通过PWM的方式在引脚上输出一个模拟量。



PWM波形的特点：

波形频率恒定，其占空比 D 可以改变。

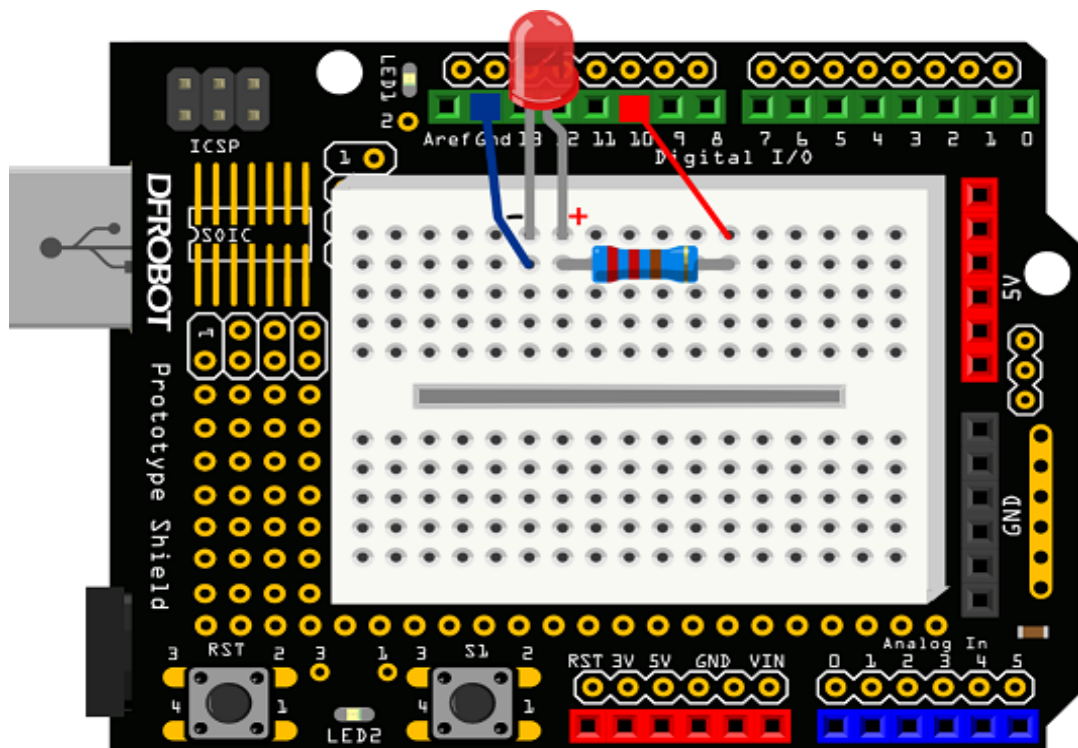


模拟输入接口

数字输入输出口

- 主要用于LED亮度控制，电机转速控制等方面。
- Arduino中的PWM的频率大约为490HZ（如何更改？）。
- UNO板上支持以下数字引脚（不是模拟输入引脚）作为PWM模拟输出：3、5、6、9、10、11。板上带PWM输出的都有~号。
- 注意：PWM输出位数为8位，从0~255。

例程：呼吸灯



Pulse Width Modulation

0% Duty Cycle - analogWrite(0)



25% Duty Cycle - analogWrite(64)



50% Duty Cycle - analogWrite(127)



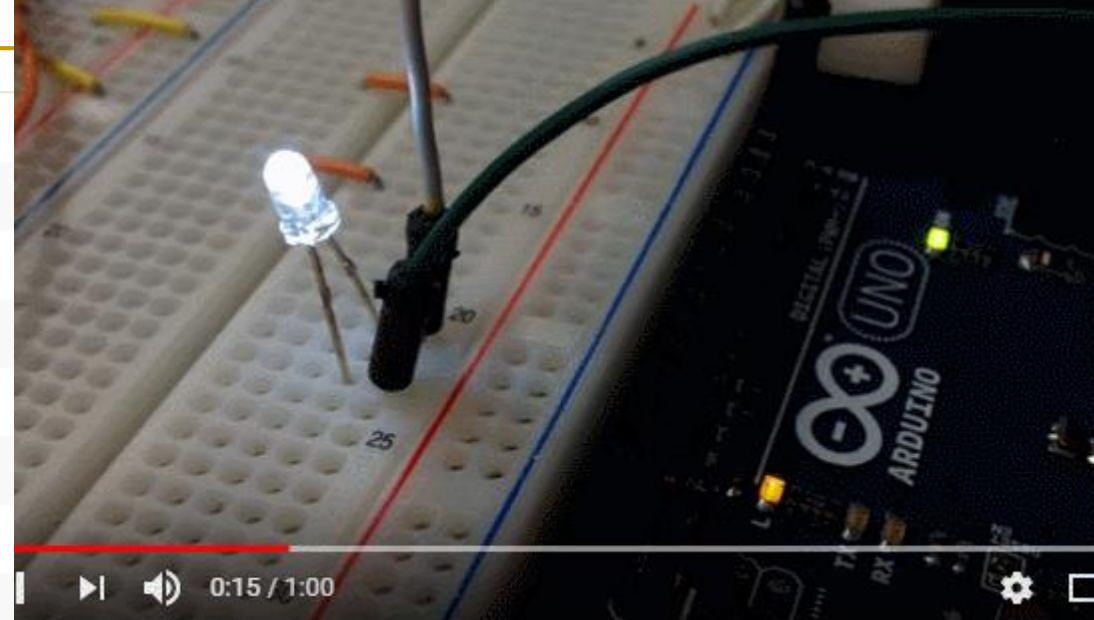
75% Duty Cycle - analogWrite(191)



100% Duty Cycle - analogWrite(255)

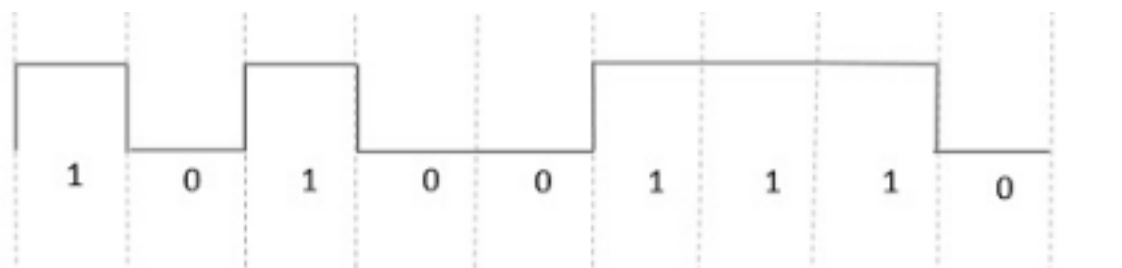


```
01. int ledPin = 10;
02. void setup() {
03.     pinMode(ledPin, OUTPUT);
04. }
05. void loop() {
06.     fadeOn(1000, 5);
07.     fadeOff(1000, 5);
08. }
09. void fadeOn(unsigned int time, int increament) {
10.     for (byte value = 0 ; value < 255; value += increament) {
11.         analogWrite(ledPin, value);
12.         delay(time / (255 / increament)); //延时函数
13.     }
14. }
15. void fadeOff(unsigned int time, int decreament) {
16.     for (byte value = 255; value > 0; value -= decreament) {
17.         analogWrite(ledPin, value);
18.         delay(time / (255 / decreament));
19.     }
20. }
```



数字通讯

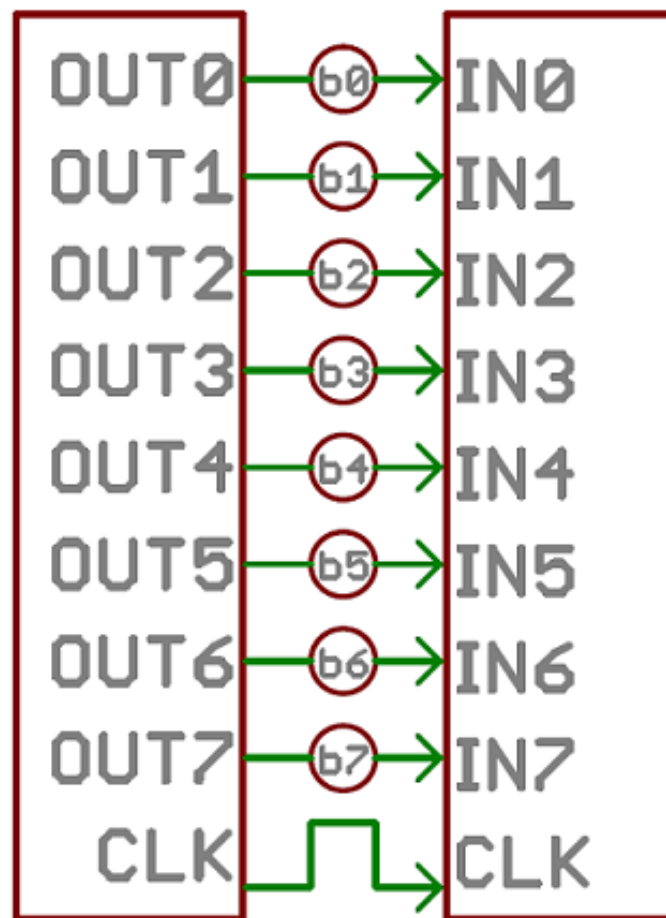
I/O: 简单的输入输出



数字通信？

串口通讯、并口通讯

1. 并行通讯



串口



并口



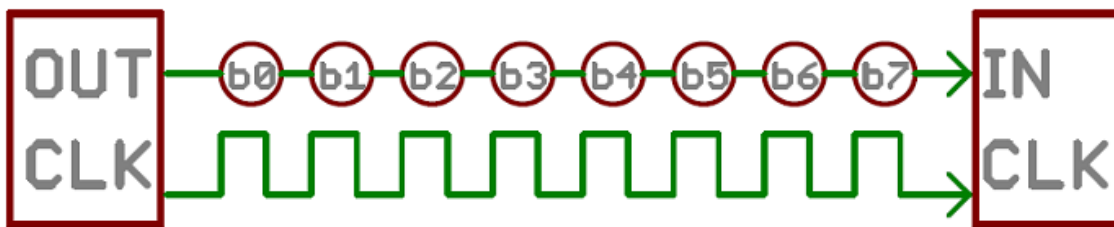
圆口



U口



2. 串行通讯



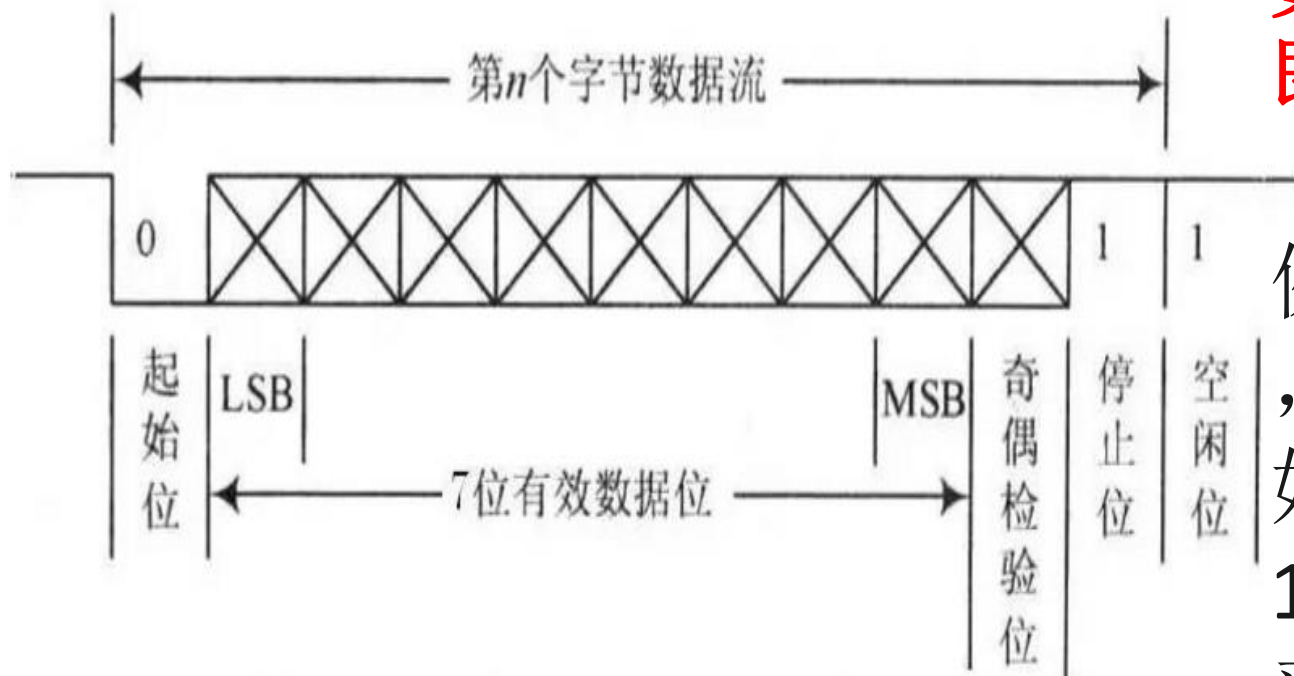
■ **数字通信** (Serial Interface) 是指数据一位一位地顺序传送

- 特点是通信线路简单，一对传输线实现双向通信。
- 串口类型很多：单片机使用的是低速串口UART, I2C, SPI等

■ 应用举例：各种传感器采集装置，GPS信号采集装置，多个单片机通讯系统，门禁刷卡系统的数据传输，机械手控制、操纵面板控制电机等等。广泛应用于低速数据传输的工程应用。

串口UART： 通用异步串行接收/发送器

工作原理是将传输数据的每个字符一位接一位地传输

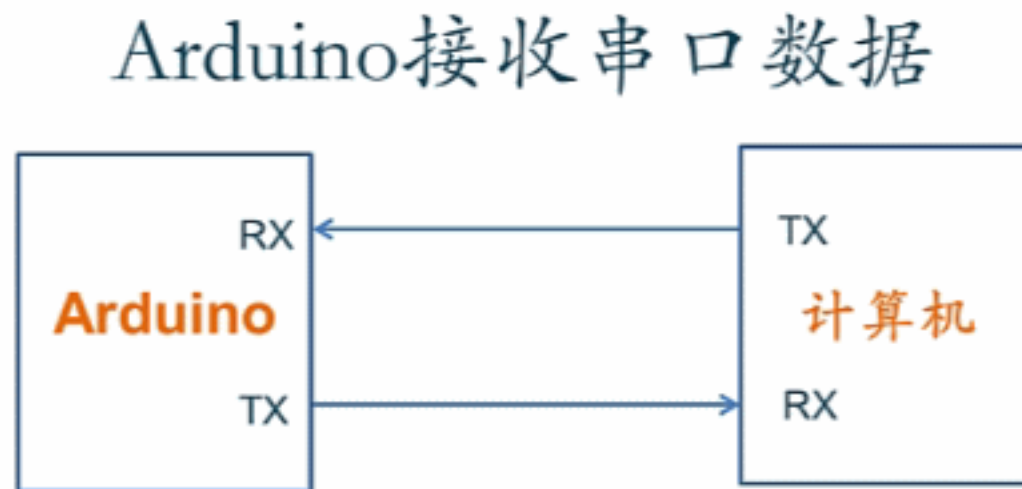
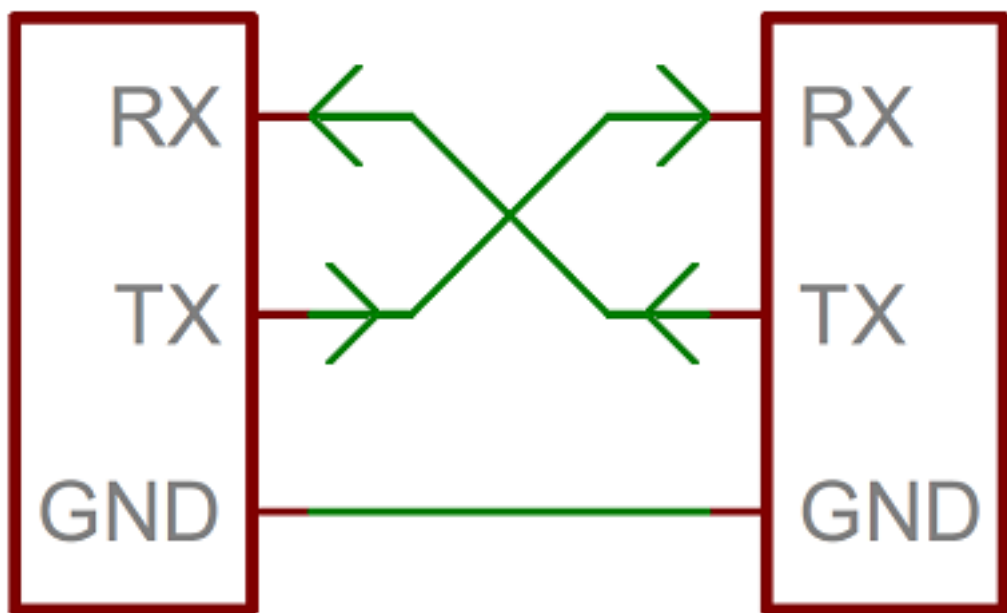


数据传送速率用波特率来表示，
即每秒钟传送的二进制位数。

例如数据传送速率为120字符/秒，而每一个字符为10位（1个起始位，7个数据位，1个校验位，1个结束位），则其传送的波特率为 $10 \times 120 = 1200$ 位/秒 = 1200波特。

串口UART： 通用异步串行接收/发送器

工作原理是将传输数据的每个字符一位接一位地传输



一、串口通讯函数

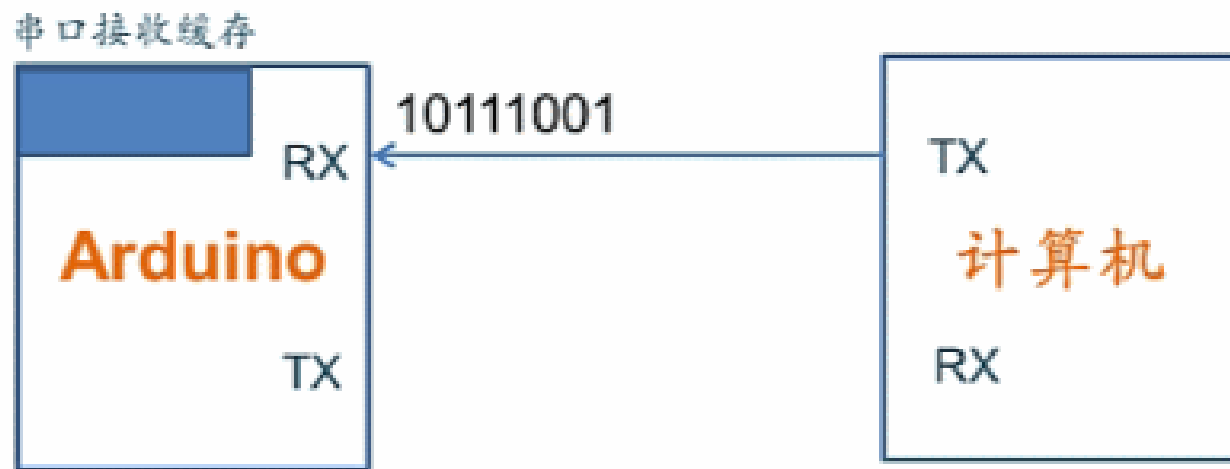
- 1、Serial.begin();
- 2、Serial.available();
- 3、Serial.read();
- 4、Serial.print();
- 5、Serial.println();

- 该函数用于设置串口的波特率。一般的波特率有9600,19200,57600,115200等。
- 波特率是指每秒传输的比特数。除以10可以得到每秒传输的字节数。
- 示范： `Serial.begin(57600);`

2、Serial.available();

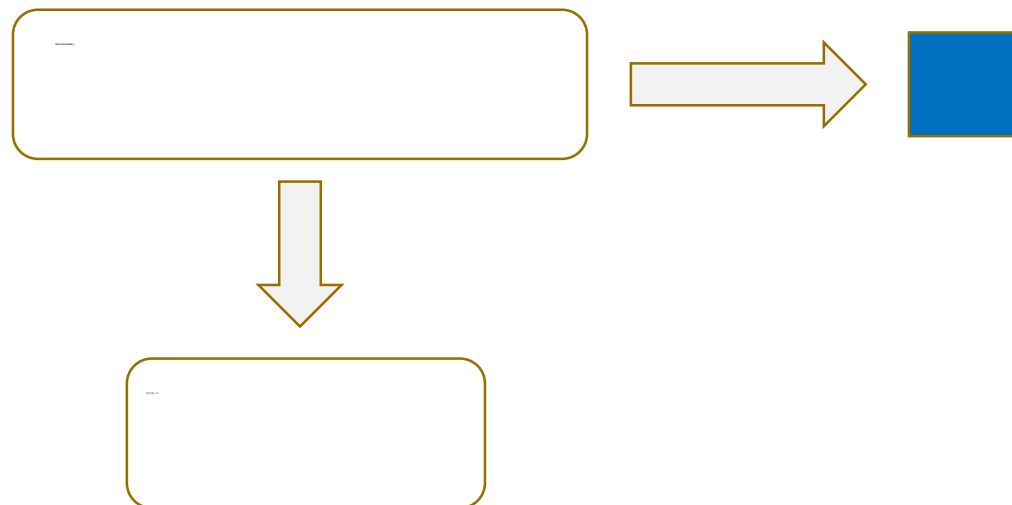
该函数用来判断串口是否收到数据，函数的返回值为int型，不带参数。

Arduino接收串口数据



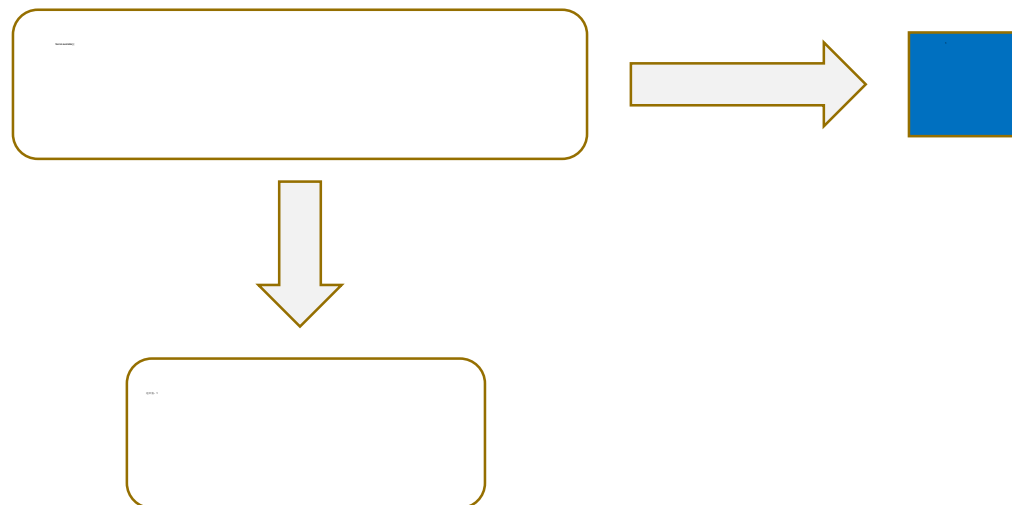
2、Serial.available();

该函数用来判断串口是否收到数据，函数的返回值为int型，不带参数。



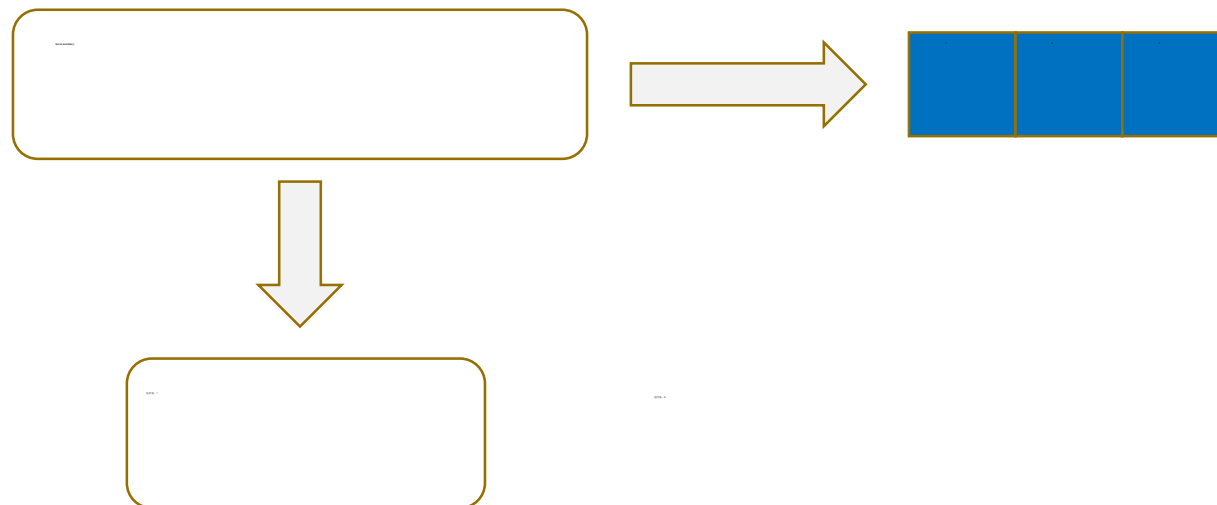
2、Serial.available();

该函数用来判断串口是否收到数据，函数的返回值为int型，不带参数。



2、Serial.available();

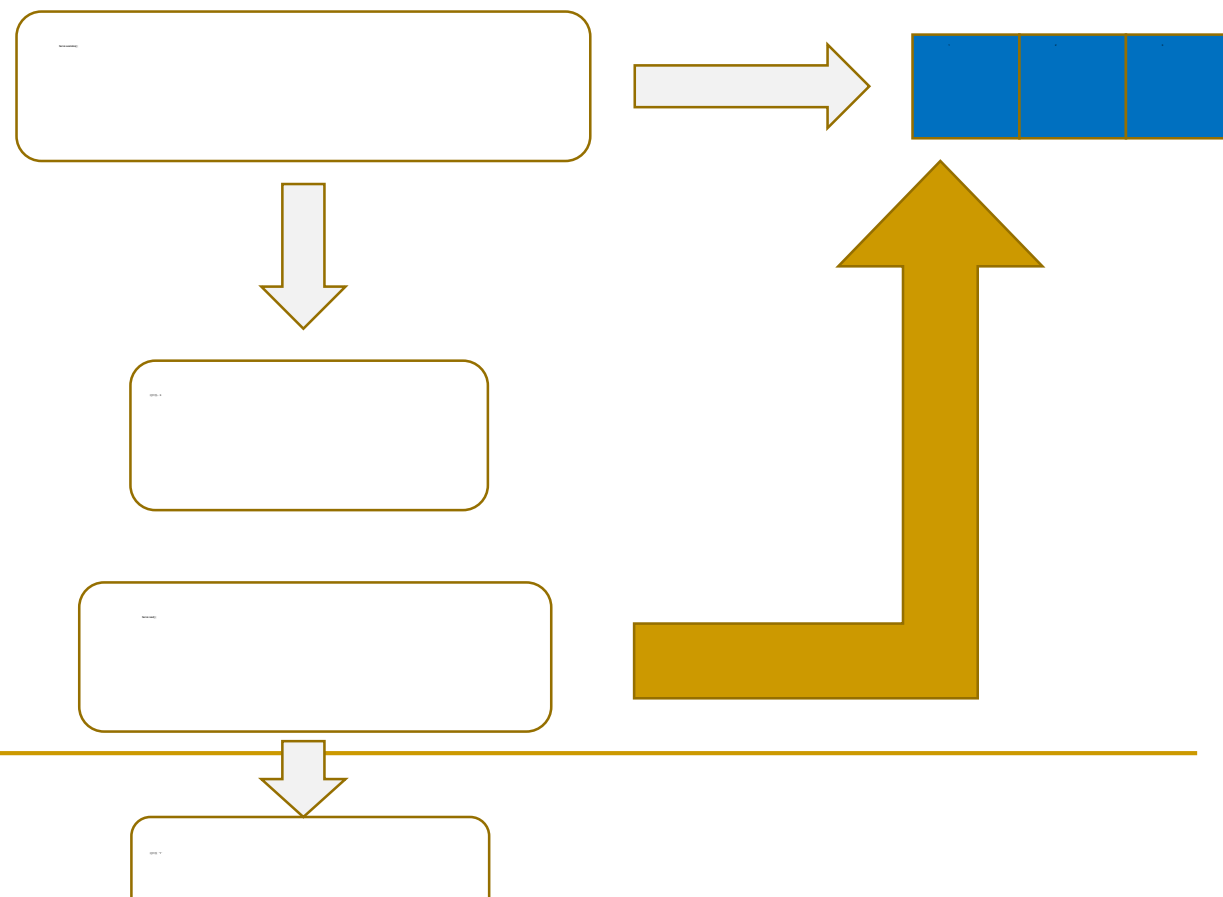
该函数用来判断串口是否收到数据，函数的返回值为int型，不带参数。



3、Serial.read();

- 将串口数据读入。该函数不带参数，返回值为串口数据，int型。

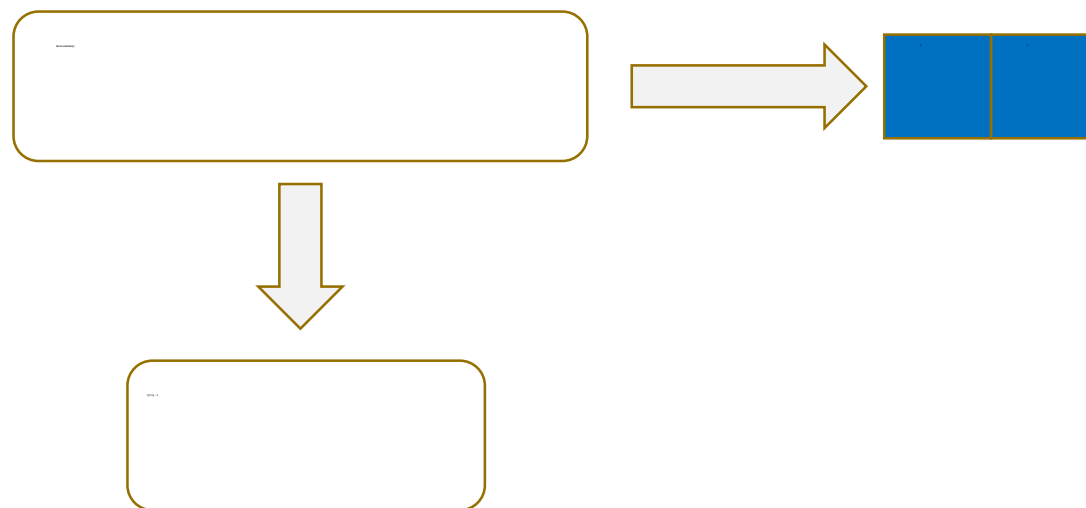
```
if(Serial.available())  
{  
    int x=Serial.read();  
}
```



3、Serial.read();

- 将串口数据读入。该函数不带参数，返回值为串口数据，int型。

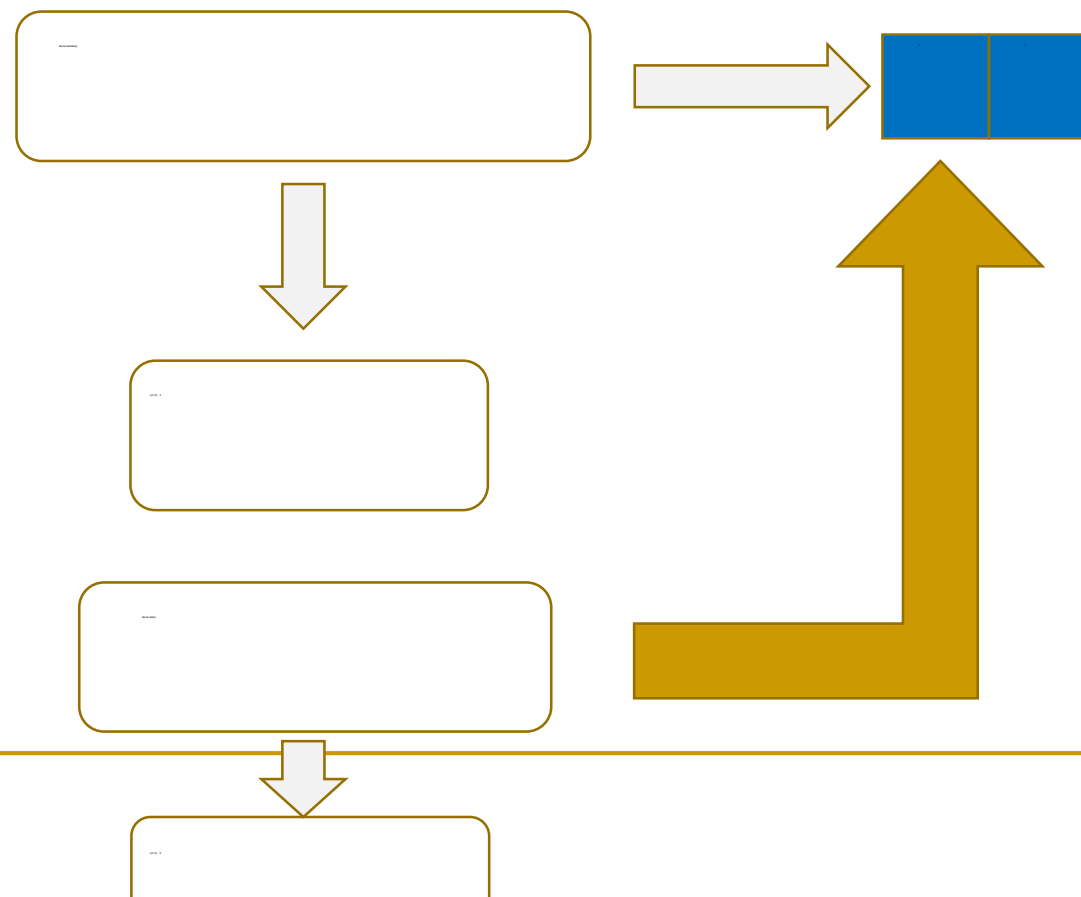
```
if(Serial.available())  
{  
    int x=Serial.read();  
}
```



3、Serial.read();

- 将串口数据读入。该函数不带参数，返回值为串口数据，int型。

```
if(Serial.available())  
{  
    int x=Serial.read();  
}
```



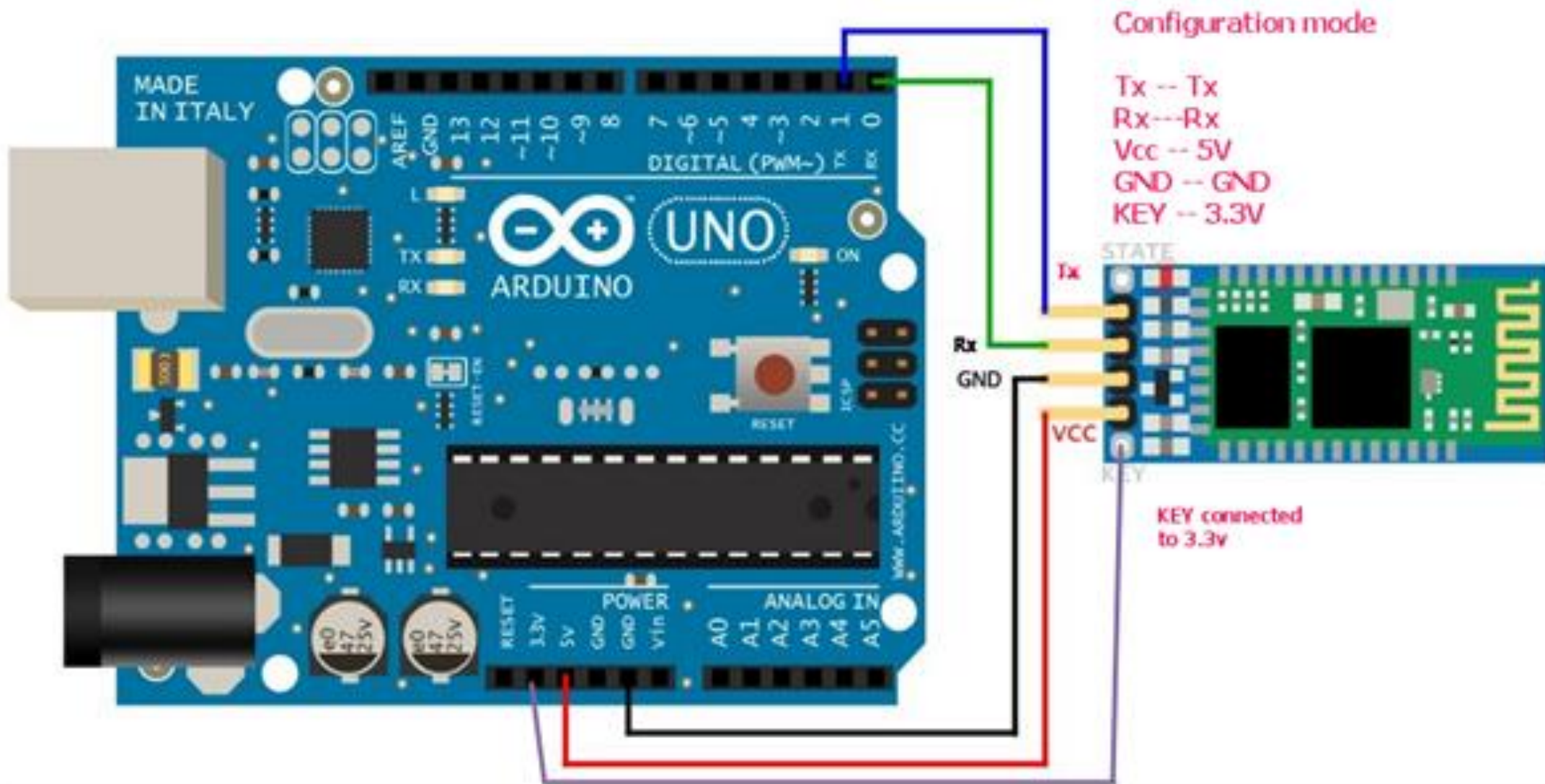
4、Serial.print();

- 该函数往串口发数据。可以发变量，也可以发字符串。
- 例句1: `Serial.print("today is good");`
- 例句2: `Serial.print(78,DEC);` 以10进制发送 “78”
- 例句3: `Serial.print(78,HEX);` 以16进制发送变量 “4E” .

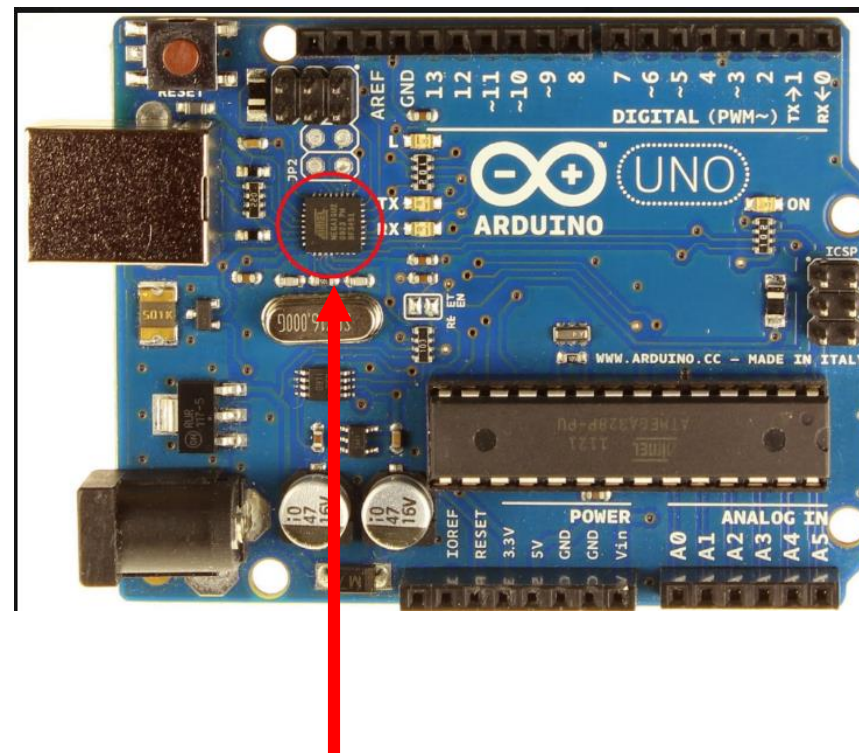
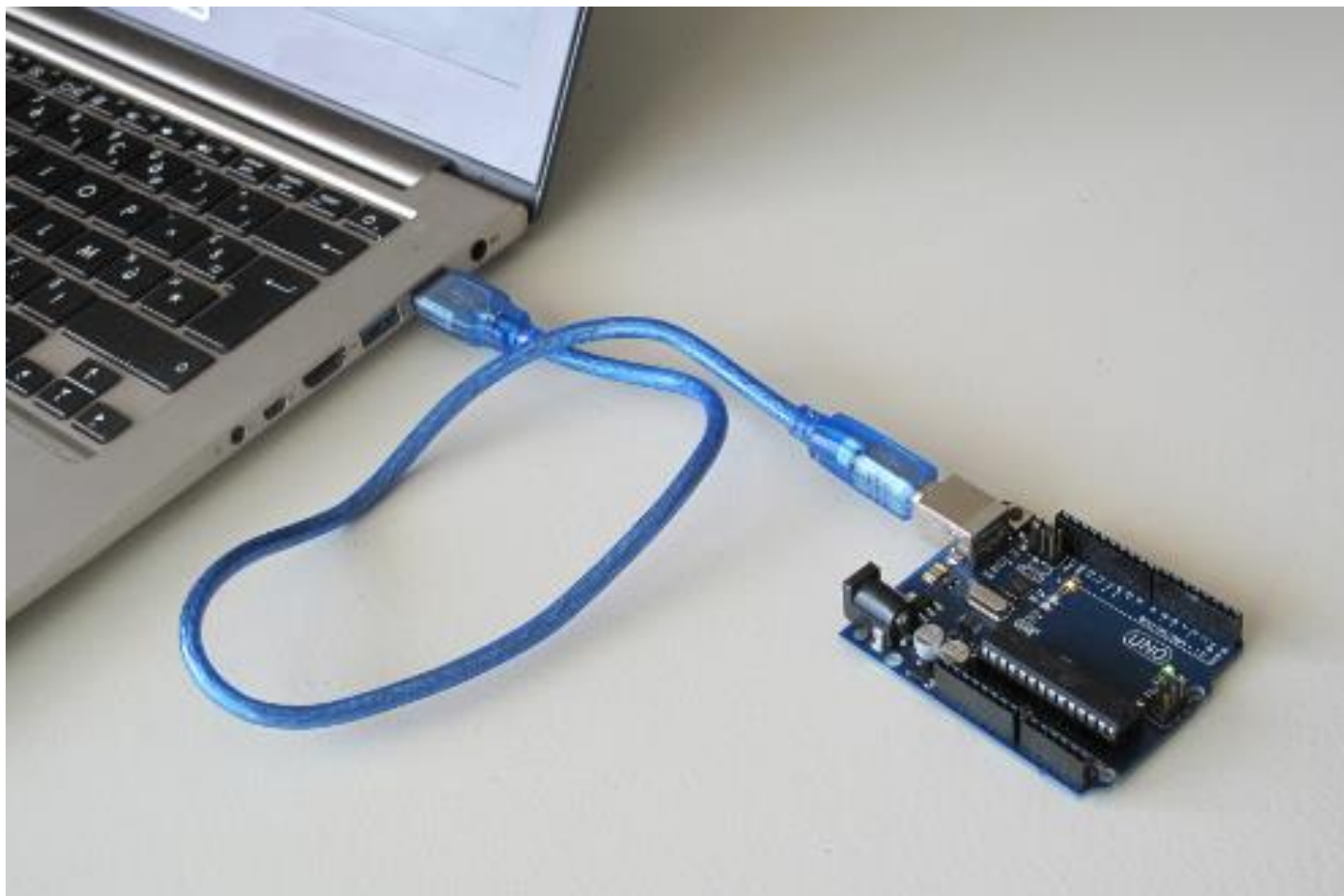
5、Serial.println();

- 该函数与Serial.print()类似，只是多了换行功能。

应用1:

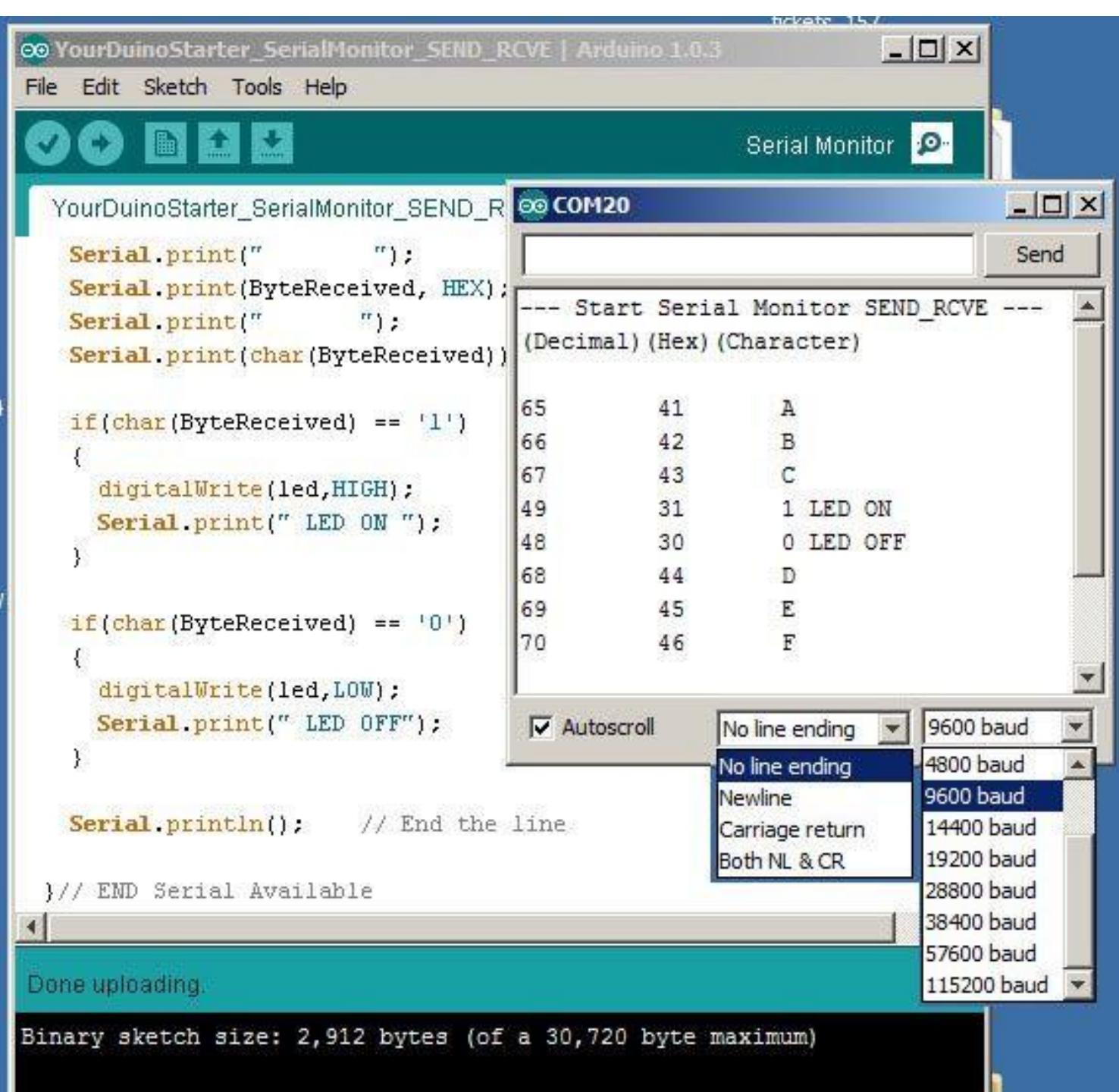


应用2：与电脑通讯



USB-TTL转
换芯片

应用2: 与时间

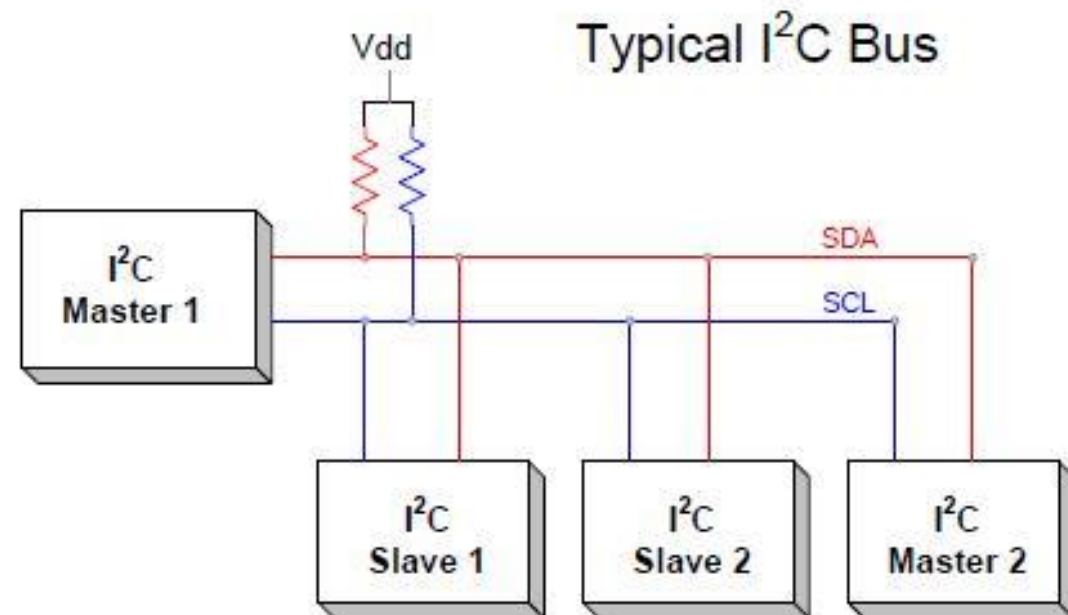
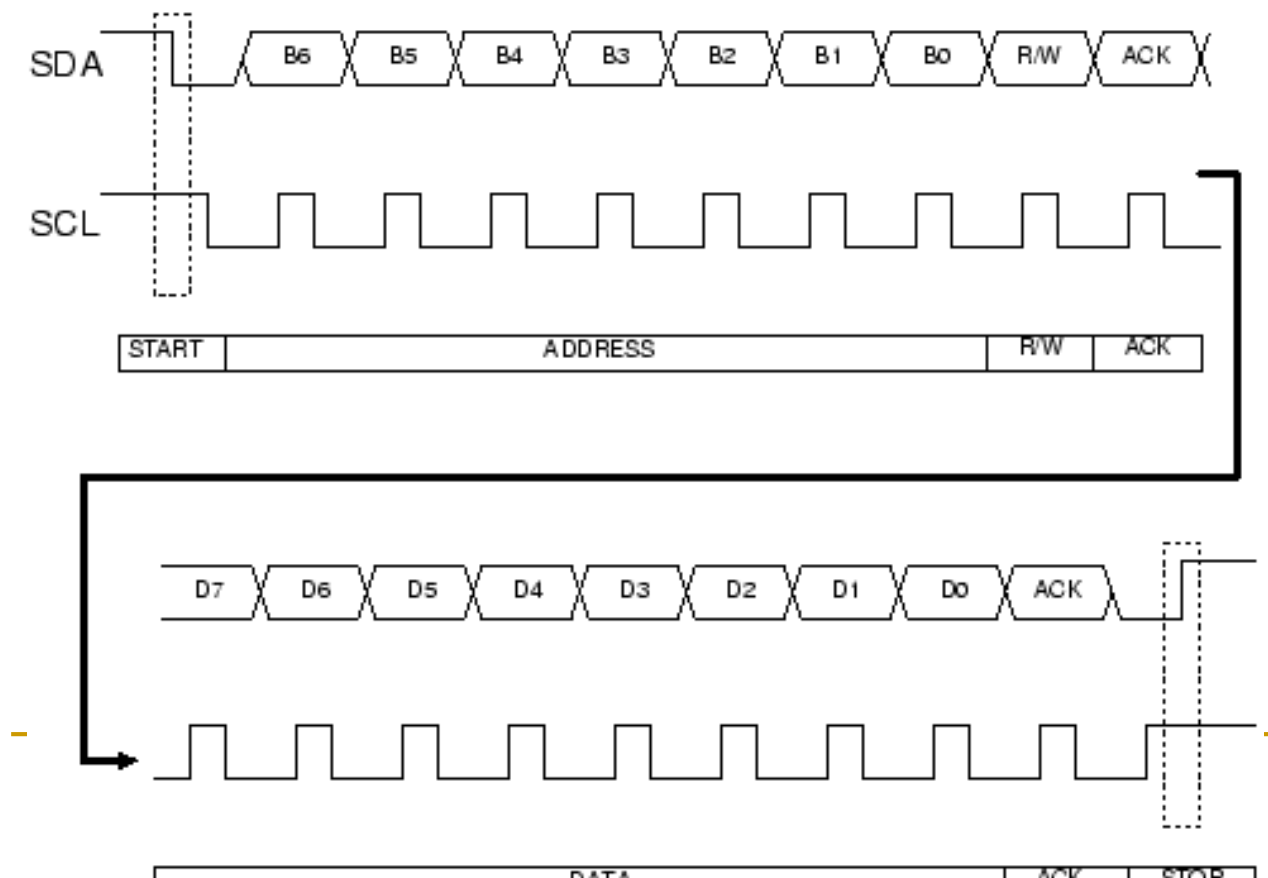


```
01.  int x=0;
02.  void setup()
03.  {
04.      Serial.begin(9600); //波特率9600
05.      Serial.print("Hello HITSZer, Please input serial data:");
06.  }
07.  void loop()
08.  {
09.      if(Serial.available())
10.      {
11.          x=Serial.read();
12.          Serial.println(x); //输出并换行
13.      }
14.  }
```

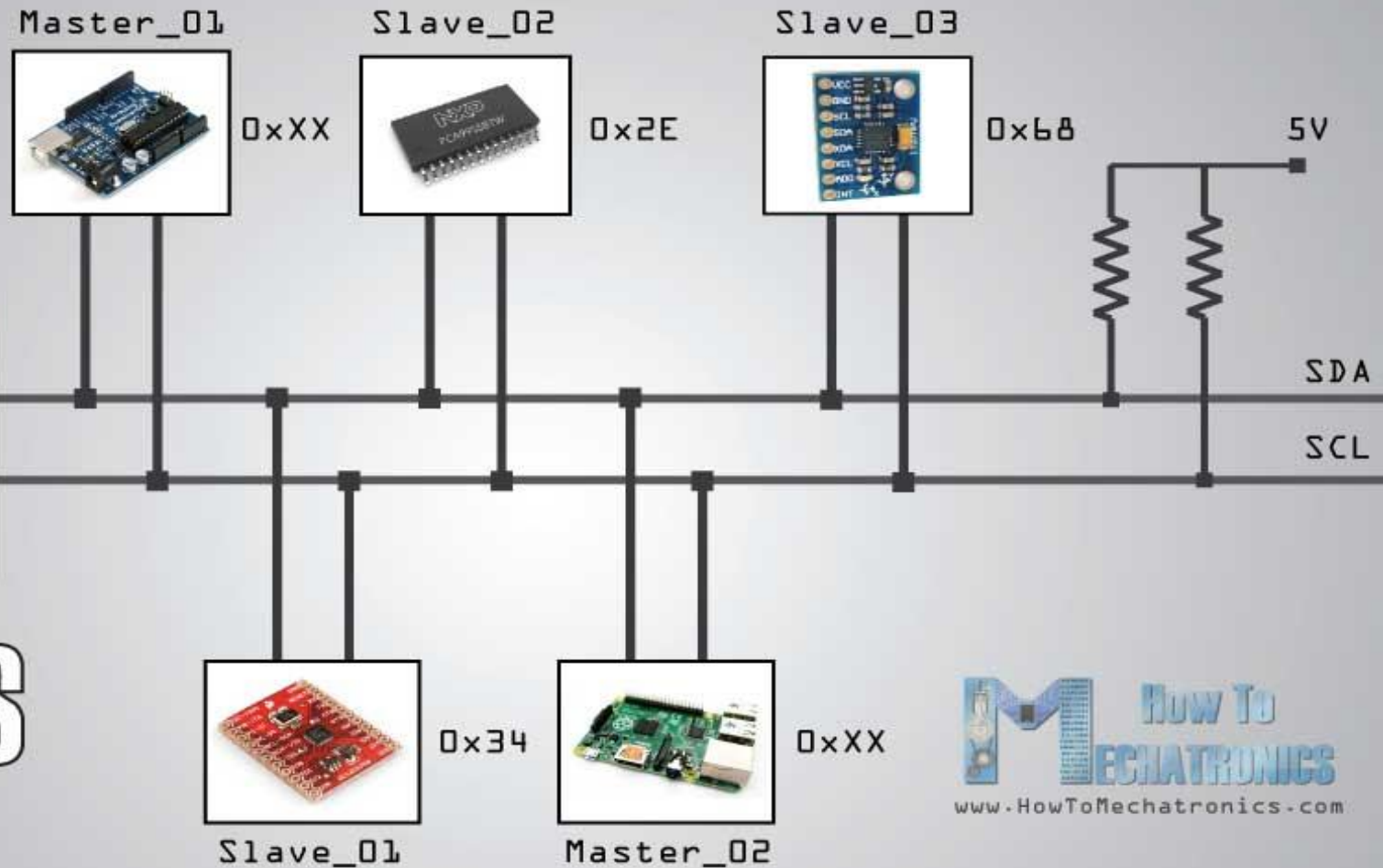


UART与I2C有什么不同点？

I2C即IIC（Inter-Integrated Circuit，集成电路总线），这种总线是一种简单、双向、二线制、同步串行总线。



How I²C BUS Works



■ 课后思考

- **Arduino**体现了什么思维方式？
- **Arduino UNO**的管脚都有哪些，什么是数字管脚？什么是模拟管脚？分别用什么函数可以操作它们？
- **Arduino**、树莓派、**NUC**、**NVIDIA Jetson Nano**有什么区别？（**Google**或者**Baidu**）
- 结合本堂课程，预习下周实验课程内容。

■ 课后学习参考网站

<https://www.arduino.cc/reference/en/>

注意事项:

- 实验课请大家**一定**要预习实验指导书;
- 学会在Gitee或者Github上托管代码。