**Arduino课堂笔记**

1.了解串行通信

串行通信是指将数据一位一位按照顺序发送或接收。例如：以串行通信方式发送数字9时，按照顺序将9的二进制位上的数字一个一个发送出去，就好比是8辆车在同一个车道上行驶，必须一辆接一辆的前行。

串行通信的特点是传输速度慢，但成本较低，用于较长距离传输数据。

2.串行通信的接口又叫串口，串口有两个引脚，一个用于数据发送，即TX引脚，另一个用于数据接收，即RX引脚，在Arduino UNO板上的0（RX）和1（TX）就是串口专用引脚。

3.编程语法

(1)串口初始化：Serial.begin(speed);

注：speed表示串口波特率，它是设定串口通信速率的参数，串口通信的双方（Arduino与计算机端的串口监视器）必须使用相同的波特率才能正常通信。

波特率是一个衡量通信速度的参数，它表示每秒传送码元的个数。串口通信常用的波特率有：300、600、1200、2400、4800、9600、14400、19200、28800、38400、57600、115200。

波特率越大，串口通信的速率越高，但通信越不稳定。

(2)输出语句

Serial.print(val);//输出时默认不换行；val表示输出的数据，可以是单个字符、字符串、整数、实数等内容。

Serial.print(val,format);//以指定形式输出数据，而且不换行。format可以分两种情况：

一种是输出数据的进制形式，包括BIN(二进制)、DEC(十进制)、OCT(八进制)、HEX（十六进制）；另一种是指定输出实数的小数点位数(默认输出2位)，例如： Serial.print(3.1415926,3)//输出小数点后3位，即3.141。

Serial.println(val);//串口输出数据，输出时默认回车换行，val表示输出的数据，与print()语句中的用法相同。

Serial.println(val,format);//以指定形式输出数据，并且回车换行。format的用法与print()语句中的用法相同。

示例程序：

void setup() {

Serial.begin(9600); //初始化串口，设定波特率为9600

}

void loop() {

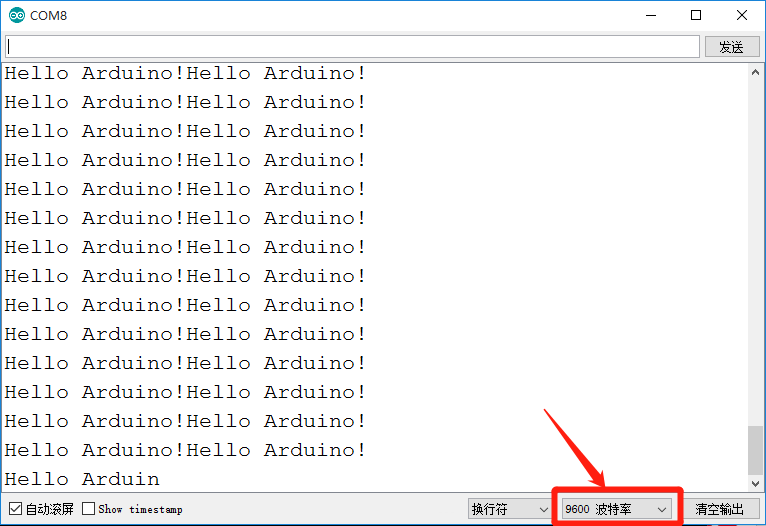
Serial.print("Hello Arduino!"); //输出内容，但不换行

Serial.println("Hello Arduino!"); //输出内容，并且换行

}

注意：串口输出的数据可以在串口监视器中查看，可以在软件中的右上角打开串口监视器，但是要先将UNO板连接到电脑上才能打开串口监视器。

打开串口监视器之后，一定要注意选择与程序中串口初始化时相同的波特率，否则可能无法正常收发数据。



(3)输入语句

Serial.read()//从串口读取数据，每读取1个字节，就会从接收缓冲区移除1字节的数据；如果缓冲区没有可读数据则返回-1。

Serial.available()//获取串口接收到的数据个数，也就是接收缓冲区中的字节数。接收缓冲区最多可保存64B的数据。此语句常用来作为判断条件，当缓冲区有数据时，就使用Serial.read()语句读取其中的数据，否则就不再读取数据。

示例1：

void setup()

{

Serial.begin(9600); //设置串口开启并使用9600波特率传输

}

void loop()

{

if (Serial.read() == 'A') //判断从串口读取的字符是否为’A’

{

Serial.println("Hello!"); //串口发送字符串”Hello!”

}

}

示例2：

void setup() {

Serial.begin(9600);

pinMode(13,OUTPUT);

}

char s; //定义字符类型的变量

void loop() {

if (Serial.available() > 0)//判断缓冲区是否有数据

{

s = Serial.read();//读取数据

}

if (s == 'a')//如果读取的数据是a则点亮LED灯

{

digitalWrite(13, HIGH);

}

if (s == 'b')//如果读取的数据是b则熄灭LED灯

{

digitalWrite(13, LOW);

}

}

4.ASCII码(美国信息交换标准代码)，是用来制订计算机中每个符号对应的代码。每个ASCII码以1个字节(Byte)储存，从0到数字127代表不同的常用符号，例如大写A的ASCII码是65，小写a则是97。在Arduino中，Serial.print(val)就是将val转换成ASCII码再进行传输示例：

void setup()

{

Serial.begin(9600);

}

char s;

void loop()

{

if (Serial.available() > 0 ) //判断串口是否接收到数据

{

s = Serial.read();

//以十进制的形式输出接收到的字符

//这个过程中Arduino会自动将字符转换成ASCII码

Serial.println(s, DEC);

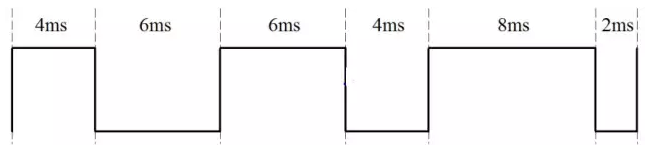
}

}



5.脉冲宽度调制（PWM）是英文“Pulse Width Modulation”的缩写，简称脉宽调制。它是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术，例如：这是一个周期是 10ms，即频率是 100Hz 的波形，但是每个周期内，高低电平脉冲宽度各不相同，这就是 PWM 的本质。

占空比是指高电平的时间占整个周期的比例。比如第一部分波形的占空比是 40%，第二部分波形占空比是 60%，第三部分波形占空比是 80%。



(1)ArduinoUNO板上只有带波浪线的引脚才能输出PWM信号，包括：3，5，6，9，10，11。

(2)Arduino UNO板上的PWM输出引脚输出的PWM值范围：0—255

(3)analogWrite(pin,value); 输出PWM，pin是要输出PWM波的引脚，参数value是PWM的脉冲宽度，范围为0—255，PWM的信号频率约为490Hz。

呼吸灯示例程序：

void setup()

{

pinMode(3, OUTPUT);

}

int n = 0;

void loop()

{

while (n <= 255)

{

analogWrite(3, n);

n += 5;

delay(50);

}

while (n <= 255)

{

analogWrite(3, n);

n -= 5;

delay(50);

}

}

6.for循环语句

for(int i=0; i<=255; i+=5)

{

//要循环执行的语句

}

for语句的执行过程可由以下4步来描述：

(1)执行变量初始化语句(int i=1)，使控制变量i获得一个初值。

(2)判断控制变量i是否满足条件表达式(i<=255)，若满足条件则执行一遍循环体，否则结束整个for语句。

(3)根据增量表达式(i+=5)，计算出控制变量所得到的新值

(4)自动转到第（2）步

呼吸灯程序示例：

void setup() {

pinMode(3, OUTPUT);

}

void loop() {

for (int i = 0; i <= 255; i++)

{

analogWrite(3, i);

delay(50);

}

for (int i = 255; i >= 0; i--)

{

analogWrite(3, i);

delay(50);

}

}