

题目：课程作业一

姓名：谢业平

班级：机械工程 3 班

学号：201820501055

Arduino 产生周期和占空比可控的方波信号

谢业平

(机械工程 3 班 201810501055)

摘要：为了完成课堂上老师布置的 arduino 课程作业，即满足两个条件，一是在指定的引脚上产生一个方波信号，要求该方波信号的周期和占空比可调；二是通过串口与用户进行人机交互，可以通过发送指令调整上述参数。本文的程序主要采用多任务调度软件框架，且能够实现串口通讯。

关键字：arduino；方波信号；人机交互；框架

1、总体方案设计

本文中主要是要产生能够控制周期和占空比的方波信号，为了实现能控的目的，采用了多线程的框架 `protothread`，主要是产生了两条路线，将两个任务分开编程，达到了两个任务互相不干扰的目的。同时，要实现一定周期和占空比的方波信号，则需要提前定义好各个参数变量，并且最终采用能够输出方波信号的函数 `Serial.println()` 来实现。

1.1 产生方波信号

方波信号即为数字信号，要产生可控的周期和占空比的方波信号，也就是在可控的时间 t_1 内产生 1，可控的时间 t_2 内产生 0。需要用到 `digitalWrite()` 函数，同时考虑周期和占空比的不同。`pwm` 的频率是指每秒钟信号从高电平到低电平再回到高电平的次数，占空比是高电平持续时间和低电平持续时间之间的比例。`pwm` 的频率越高，其对输出的响应就会越快，频率越低输出响应越慢。脉冲宽度调制基本上是一个随时间变化而变化的方波。基本的 PWM 信号如下图所示。

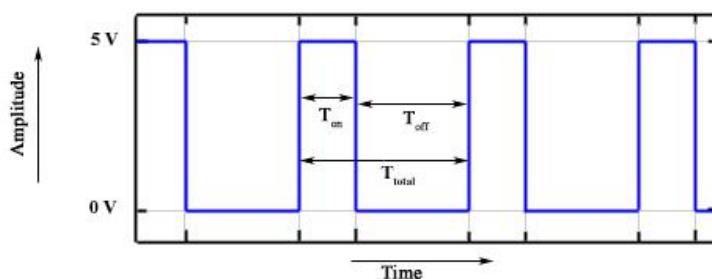


图 1 方波示意图

对于方波信号和周期，如图所示， T_{on} 表示导通时间， T_{off} 表示信号的关断时间。周期是导通和关断时间的总和，并按照以下公式计算：

$$dc = \frac{T_{on}}{T_{on} + T_{off}} = \frac{T_{on}}{T_{total}} \tag{1}$$

1.2 人机交互

人机交互是为了实现串口输入数据能控制产生方波周期和占空比的不同，考虑到周期和占空比的范围不同，应当单独判断它们的数值大小。主要用到串口函数 Serial 系列。

2、方案的调试

以下是在周期 $T=400$ ，占空比 $dc=0.2$ 时产生的方波图。

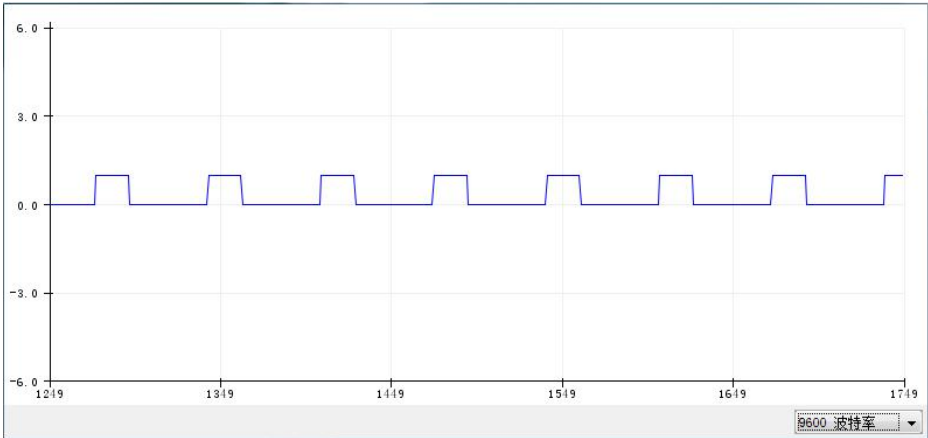


图 2 实验方波示意图

以下是串口监视器的窗口：

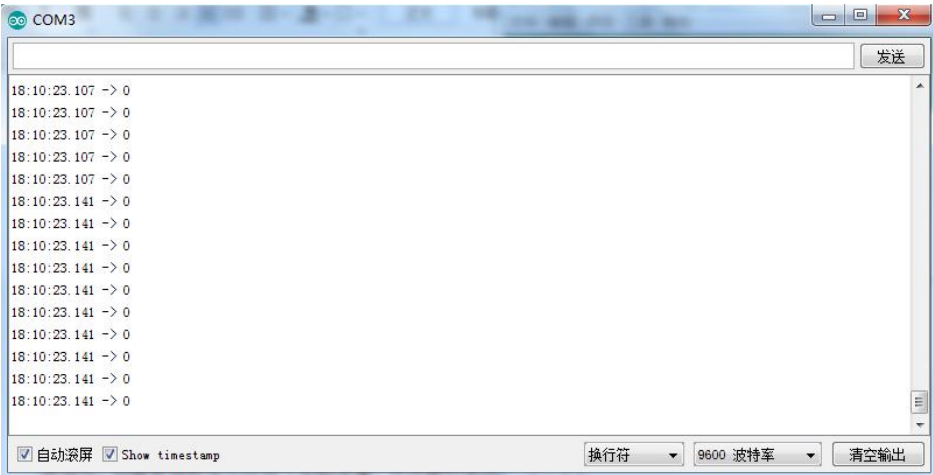


图 2 串口通信示意图

3、总结

通过对整个 `arduino` 的操作过程，解决了可调周期和占空比的方波，实现了串口的监测和通讯过程，保证整个系统能够完好的运行下去。同时，掌握了 `arduino` 的一些基本的使用，熟悉了它的操作流程。发现 `arduino` 有许多的有意思的地方，也对它产生了浓厚的兴趣。

附录

```
#define PT_USE_TIMER

#define PT_USE_SEM

#define PWM 11


#include "pt.h"//引用库


static struct pt thread1,thread2;//创建两个任务
static struct pt_sem sem_PWM;//创建一个信号量，同一时间只能一个任务占用
unsigned int T;//定义周期
float dc;//定义占空比
float temp;//定义为一个浮点数变量 temp
int M;
float N;
String A;
String B;
String str1 = "设置初始值为 T=200,dc=0.3";//输入字符串
String comdata = "";


void setup() {
    T = 200;
    dc = 0.3;
    pinMode(PWM,OUTPUT);//初始化 PWM 口和串口
    Serial.begin(9600);//设置串口波特率
    PT_SEM_INIT(&sem_PWM,1);//初始化信号量 1
    //初始化线程
    PT_INIT(&thread1);
    PT_INIT(&thread2);
}


//产生方波信号任务
static int thread1_entry(struct pt *pt)
{
    PT_BEGIN(pt);
```

```

while (1)
{
    PT_SEM_WAIT(pt,&sem_PWM);
    digitalWrite(PWM,HIGH);//发出方波信号
    temp=T*dc;
    PT_TIMER_DELAY(pt,temp);//信号持续时间
    digitalWrite(PWM,LOW);//关闭信号
    temp=T*(1-dc);
    PT_TIMER_DELAY(pt,temp);//信号关断时间
    PT_SEM_SIGNAL(pt,&sem_PWM);//继续
    PT_YIELD(pt); //检查其它的信号
}
PT_END(pt);
}

//通过串口产生人机交互任务
static int thread2_entry(struct pt *pt)
{
    PT_BEGIN(pt);
    while (1)
    {
        PT_WAIT_UNTIL(pt, Serial.available());//等到有串口消息再继续
        PT_SEM_WAIT(pt,&sem_PWM);//继续
        Serial.println(str1);//启动串口
        while(Serial.available()>0)
        {
            comdata += char(Serial.read());
            delay(2);
        }
        if(comdata.length() > 0)//判断端口的数是否大于 0
        {
            //判断周期
            if(comdata.startsWith("T="))//判断输入的内容是否以“T=”开始
            {
                A=comdata.substring(2);//将读取到的值赋值给 A
            }
        }
    }
}

```

```

    M=A.toInt();//提取 A 中数字
    if(M>=10&&M<=10000)//判断 M 的取值范围
    {
        T=M;//将 M 赋值给 T
    }
    Serial.println(T);//打印数据到串行端口
}

//判断占空比
if(comdata.startsWith("dc="))//判断输入的内容是否以“dc=”开始
{
    B=comdata.substring(3); //将读取到的值赋值给 B
    N=B.toFloat();
    if(N>=0.1&&N<=0.90)//判断 N 的取值范围
    {
        dc=N;//将 M 赋值给 T
    }
    Serial.println(dc);//打印数据到串行端口
}
comdata = "";
}
while (Serial.available())
    Serial.read();
//清空串口数据，防止又来
    PT_SEM_SIGNAL(pt,&sem_PWM);//归还使用权了
}
    PT_END(pt);
}

void loop() {
    //依次调用即可
    thread1_entry(&thread1);
    thread2_entry(&thread2);
    Serial.println(digitalRead(PWM));//串口绘图器绘制方波信号}

```