

课程作业一

潘奔 201820501031 机械工程 2 班

一、实验简介

以 Arduino 为硬件平台，利用 PWM 波输出原理与串口通讯原理，实现以下功能：

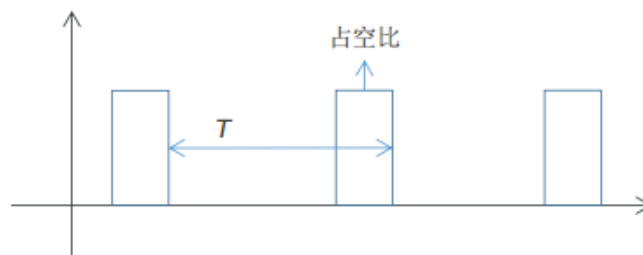
(1) 在指定的引脚上产生一个方波信号，要求该方波信号的周期和占空比可调。

(2) 通过串口与客户进行人机交互，可通过发送指令调整上述参数（信号周期与占空比）。

(3) 动态显示修改后的 PWM 波形。

说明：

(1) 假设指定的引脚编号为 7, 那么该程序应该能在 7 号引脚上生成的波形如下图所示：



T 取值范围为 $[10, 10000]$, 单位是 ms。

占空比 (Duty-cycle) 是一个比例参数，取值范围为 $[0.1-0.9]$ 。

(2) 与用户通过串口进行交互应该是独立的任务，也就是说生成的波形是可以在线动态调整的。用户输入的指令应该遵循下面的规范：

T=1000↵

dc=0.6↵

提示：

(1) 记得使用我们课堂上讲过的多任务调度软件框架。

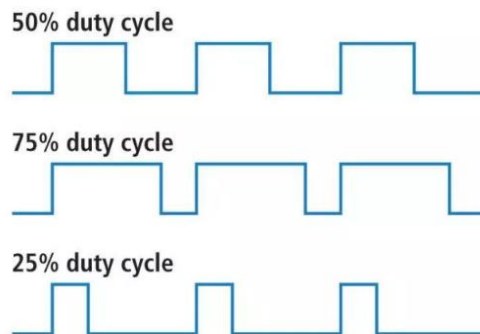
(2) 串口通讯编程实现的时候要小心，不能干扰到波形的正常生成。一个比较好的方法是开辟一个缓冲区用来临时存储串口发送来的数据，当用户输入完指令后会键入一个回车字符，可以在收到该字符 (CR/LF) 后再统一处理用户输入。

二、实验原理

2.1. PWM 原理

脉宽调制（PWM）控制方式就是对逆变电路开关器件的通断进行控制，使输出端得到一系列幅值相等的脉冲，用这些脉冲来代替正弦波或所需要的波形。也就是在输出波形的半个周期中产生多个脉冲，使各脉冲的等值电压为正弦波形，所获得的输出平滑且低次谐波少。按一定的规则对各脉冲的宽度进行调制，即可改变逆变电路输出电压的大小，也可改变输出频率。

方波高电平的宽度在一个周期里的占比被称为占空比（Duty Cycle）。改变 PWM 的占空比，可以改变输出信号的平均电压，实现模拟电压的输出。



本实验使用 Arduino UNO 实验平台，Arduino UNO 产生 PWM 波形有三种方式。

方法一：数字 IO 转换法：就是一个周期内分别自定义高电平和低电平输出各占的时间。此方法优点是周期和占空比可控，PWM 的比例更精确。

方法二：analogWrite 函数法：直接一条语句就可以产生 PWM 波形，但 PWM 频率固定，只能改变其占空比。

方法三：使用定时器：通过利用定时器 TCx 产生 PWM 波形，此方法只能选择固定的 PWM 频率。

由于本实验要求 PWM 周期和占空比可控，因此选择数字 IO 转换法产生 PWM。

2.2. 串口通讯

Arduino 串口通讯，主要是这确定了串口波特率后，运用以下的串口通讯函数，实现与 PC 端的数据交互。由于串口 Serial.Read() 得到的数据为 ASCII 型数据，因此需要将 ASCII 数据转化为实际要读取的 T、dc 数，即 double 型数据。

```

Serial.begin(); //开启串口

Serial.end(); //关闭串口

Serial.available(); //判断串口缓冲器是否有数据装入

Serial.read(); //读取串口数据

Serial.peek(); //返回下一字节(字符)输入数据，但不删除它

Serial.flush(); //清空串口缓存

Serial.print(); //写入字符串数据到串口

Serial.println(); //写入字符串数据+换行到串口

Serial.write(); //写入二进制数据到串口

Serial.SerialEvent(); //read时触发的事件函数

Serial.readBytes(buffer,length); //读取固定长度的二进制流

```

3. 3. 多线程原理

本实验需要在输出 PWM 波的同时，可以进行动态修改 PWM 周期与占空比，因此需要采用多线程编程。本实验运用 ProtoThreads 多线程框架，采用两个线程实现结果。

```

PT_INIT(pt) 初始化任务变量，只在初始化函数中执行一次就行
PT_BEGIN(pt) 启动任务处理，放在函数开始处
PT_END(pt) 结束任务，放在函数的最后
PT_WAIT_UNTIL(pt, condition) 等待某个条件（条件可以为时钟或其它变量，IO等）成立，否则直接退出本函数，下一次进入本函数就直接跳到这个地方判断
PT_WAIT_WHILE(pt, cond) 和上面一个一样，只是条件取反了
PT_WAIT_THREAD(pt, thread) 等待一个子任务执行完成
PT_SPAWN(pt, child, thread) 新建一个子任务，并等待其执行完退出
PT_RESTART(pt) 重新启动某个任务执行
PT_EXIT(pt) 任务后面的部分不执行，直接退出重新执行
PT_YIELD(pt) 锁死任务
PT_YIELD_UNTIL(pt, cond) 锁死任务并在等待条件成立，恢复执行
在pt中一共定义四种线程状态，在任务函数退出到上一级函数时返回其状态
PT_WAITING 等待
PT_EXITED 退出
PT_ENDED 结束
PT_YIELDED 锁死

```

三、程序代码

```
#define PT_USE_TIMER
#define PT_USE_SEM
#include "pt.h"
#define PWM 3
static struct pt thread1, thread2;
static struct pt_sem sem_PWM;
double T;
double dc;
double temp;
String str1 = "默认 T = 300, dc = 0.5";

/*****Arduino 单次运行程序 setup*****/
void setup() {
    T = 300;
    dc = 0.5;
    pinMode(PWM, OUTPUT);
    Serial.begin(9600);
    PT_SEM_INIT(&sem_PWM, 1);
    PT_INIT(&thread1);
    PT_INIT(&thread2);
}

/*****ASCII 码数据转化为 double 型数据*****/
//数据读取
double Read_num(void)
{
    String comdata = "";
    double numdata=0;
    int mark = 0;
    while(Serial.available()>0)
    {
        comdata += char(Serial.read());
        delay(2);
        mark = 1;
    }
    if(mark == 1)
    {
        Serial.println(comdata);
        if(comdata[0]=='T')
        {
            for(int i=2;i<comdata.length();i++)
```

```

        {
            numdata = numdata*10+(comdata[i]-'0');
        }
        comdata = String("");
        T = numdata;
        Serial.print("T = ");
        Serial.println(T);
        Serial.print("dc = ");
        Serial.println(dc);
        return numdata;
    }
    else if(comdata[0]=='d' && comdata[1]=='c')
    {
        numdata = (comdata[comdata.length()-1]-'0')*0.1;
        comdata = String("");
        dc = numdata;
        Serial.print("T = ");
        Serial.println(T);
        Serial.print("dc = ");
        Serial.println(dc);
        return numdata;
    }
    mark = 0;
}
}

/*****PWM 波形输出*****/
//PWM
static int thread1_entry(struct pt *pt)
{
    PT_BEGIN(pt);
    while (1)
    {
        PT_SEM_WAIT(pt, &sem_PWM);
        digitalWrite(PWM, HIGH);
        temp=T*dc;
        PT_TIMER_DELAY(pt, temp);
        digitalWrite(PWM, LOW);
        temp=T*(1-dc);
        PT_TIMER_DELAY(pt, temp);
        PT_SEM_SIGNAL(pt, &sem_PWM);
        PT_YIELD(pt); //Check the other events.
    }
}

```

```

    PT_END(pt);
}

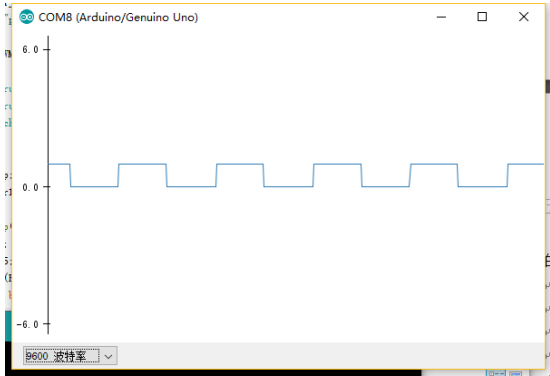
/*****串口通讯*****/
//串口
static int thread2_entry(struct pt *pt)
{
    PT_BEGIN(pt);
    while (1)
    {
        PT_WAIT_UNTIL(pt, Serial.available());
        PT_SEM_WAIT(pt, &sem_PWM);
        temp = Read_num();
        PT_SEM_SIGNAL(pt, &sem_PWM);
    }
    PT_END(pt);
}

/*****Arduino 无限循环程序*****/
void loop() {
    //Check each thread by priority
    thread1_entry(&thread1);
    thread2_entry(&thread2);
    Serial.println(digitalRead(PWM));
}

```

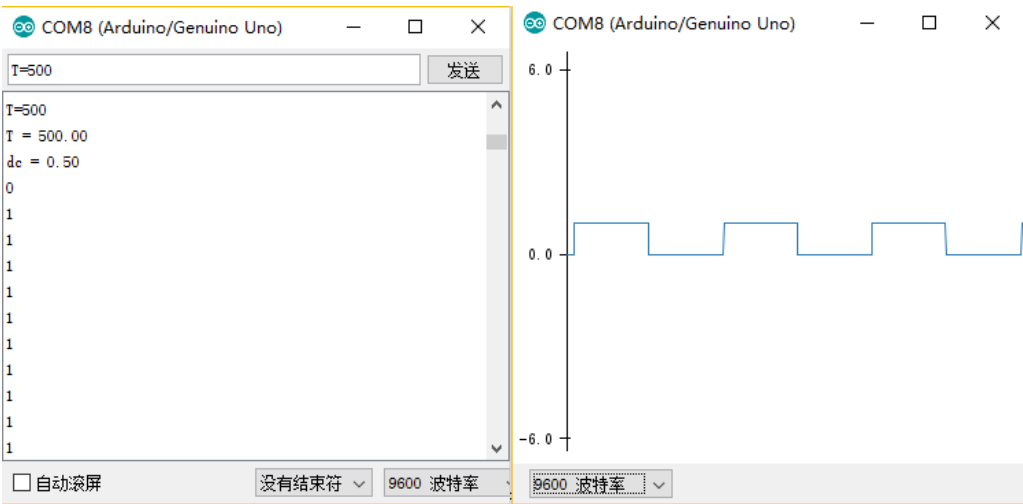
四、实验结果

当没有输入 PWM 周期与占空比时，默认固定输出周期为 300ms，占空比为 0.5 的 PWM 波形。通过 Arduino 串口绘图器可缓存图形：



通过 Arduino 串口监视器，输入要改变的周期与占空比。当按照格式输入周期或占空比的数值后，串口首先会返回输入的数据，然后再输出改变后，即当前的周期 T 与占空比 dc 值，从而输出改变后的 PWM 波形。

得到改变后的 PWM 波形（T=500ms，dc=0.5）：



得到改变后的 PWM 波形（T=300ms，dc=0.8）：

