**《计算机控制及接口技术》**

**实验报告**

**院 系 机械与自动控制学院**

**学生姓名 栗硕**

**学 号 2018G0505021**

**指导教师 李晓明**

**完成日期 2018年2月**

**浙江理工大学**

作业 1

以 Arduino 为硬件平台， 编写程序实现以下功能：

(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) 通过串口与用户进行人机交互， 可通过发送指令调整上述参数

程序：

1. 指定引脚编号为9，输出周期与占空比的不同通过在arduino平台上接发光二极管显示。硬件连接如下图所示，将发光二极管的一脚接在9号引脚上，另一脚接在接地上。矩形波参数设置和发光二极管亮度参数设置如下：

//设置led为引脚9

int led = 9;

//设置矩形波参数

float T=3000;

float dc=0.5;

//设置亮度参数

float t = T\*dc;

1. 加入框架

包括框架头文件

#include "TaskScheduler.h"

#include <Wire.h>

为方便删除任务，使用函数Sch.addTask(任务名称,开始时间,执行周期,1)时，会返回这个任务的ID，将这个ID赋给一个变量。需要删除任务时，用删除任务函数Sch.deleteTask(任务ID)，就能把任务删除。

1. 设置串口缓存区，以用来临时存储串口发送来的数据，避免干扰到波形的正常生成。当当串口收到回车字符（'\r'）时，代表用户输入完毕，arduino再统一处理用户输入的数据，具体代码如下：

//设置串口缓冲、数据长度参数、标志位

char dtaUart1[15];

char dtaUart2[15];

char dtaLen = 0;

char DealFlag1=0;

char DealFlag2=0;

//读串口与写缓存区

void serialEvent()

{

char temp;

unsigned int a,c;

float m;

char b;

while(Serial.available())

{

temp=Serial.read(); //读串口数据

dtaUart[dtaLen++] =temp ; //将串口数据写入缓存区

if(temp=='\r') //当串口输入为回车键时

{

DealFlag1=1; //标志位1置1

//判断起始位

if(dtaUart[0]=='T')

{

dtaLen=dtaLen-2;

for(c=0,b=2;dtaLen>1;dtaLen--)

{

switch(dtaUart[b++])

{

case '0':a=0;break;

case '1':a=1;break;

case '2':a=2;break;

case '3':a=3;break;

case '4':a=4;break;

case '5':a=5;break;

case '6':a=6;break;

case '7':a=7;break;

case '8':a=8;break;

case '9':a=9;break;

}

c=c\*10+a; //计算输入值

dtaLen=0；

}

T=c; //给周期T赋值

}

else if(dtaUart[0]=='d') //检测起始位

{ char DealFlag2=1; //标志位2置1

dtaLen=dtaLen-5;

for(m=0,b=5;dtaLen>1;dtaLen--)

{

switch(dtaUart[b++])

{

case '0':a=0;break;

case '1':a=1;break;

case '2':a=2;break;

case '3':a=3;break;

case '4':a=4;break;

case '5':a=5;break;

case '6':a=6;break;

case '7':a=7;break;

case '8':a=8;break;

case '9':a=9;break;

}

m=m\*10+a;

}

m=m/100; //由于一开始计算时将树脂放大了100倍，故此处除以100

dtaLen=0;

dc=m; //将占空比值赋予dc

}

t=m\*T;

//输出周期及占空比

Serial.println("T=");

Serial.println(T);

Serial.println("dc=");

Serial.println(dc);

pwm= Sch.addTask(led9pwm,1); //重新将pwm发生函数加入任务

}

//当T与dc都接受到串口的数据时，将setUpdate函数加入任务框架

if（char DealFlag1=1）

if（char DealFlag2=1）

Sch.addTask(setUpdate,1);

break;

}

}

1. Arduino读取缓存区里的数据

void setUpdate()

{

Sch.deleteTask(pwm); //将旧的pwm函数从任务中删除

int a;

char b;

Sch.deleteTask(pwm);

if(dtaUart[0]=='T')

{

dtaLen=dtaLen-3;

for(a=(int)dtaUart[2], b=2;dtaLen;dtaLen--)

{

a=a\*10+dtaUart[++b];

}

T=a;

}

else if(dtaUart[0]=='d')

{

dtaLen=dtaLen-3;

for(a=(float)dtaUart[3],b=3;dtaLen;dtaLen--)

{

a=a\*10+dtaUart[++b];

}

dc=a;

}

DealFlag1=0;

DealFlag2=0;

}

1. pwm主程序

int pwm;//设置多任务ID

pwm= Sch.addTask(led9pwm,1); //将PWM程序加入任务框架

void led9pwm()

{

digitalWrite(led,HIGH); //将引脚9拉高

delay(t); //设置延迟

digitalWrite(led,LOW);// //将引脚9拉低

delay(T-t);//设置延迟

}

1. 初始化函数

void setup()

{

pinMode(led, OUTPUT);

Serial.begin(9600);//串口初始化

Sch.init();

pwm= Sch.addTask(led9pwm,1);

Sch.start();

}

7、主循环

void loop()

{

Sch.dispatchTasks();

}