

# 《计算机控制系统与接口技术》

## 课程作业二

院系： 机械与自动控制学院

姓名： 燕 超

学号： 2018G0505057

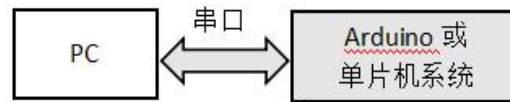
指导老师： 李晓明

完成日期： 2018 年 12 月

浙江理工大学

## 一、课程作业 2 内容

搭建以下实验系统：



在 PC 中编写软件，运行一仿真程序，模拟一系统（例如温度控制系统，电机控制系统等），要求包含该系统的模型，以及控制接口。该控制接口能够接收来自串口的控制指令。

在 Arduino 中编写控制程序，实现离散 PID 控制。要求该程序包括 PID 控制算法以及控制接口实现，该控制接口能够控制 PC 里的模型程序。

分别运行上述实验系统，在 PC 端记录控制系统的状态曲线，绘制该曲线并机进行说明。

## 二、课程作业 2 过程

### 1 PID 程序设计

下图为 PID 执行的相应代码：

```
float PID_realize(float err)
{
    float incrementSpeed;
    pid.err = err / 3000;
    pid.integral += pid.err;
    pid.voltage = pid.Kp*pid.err+pid.Ki*pid.integral+pid.Kd*(pid.err-pid.err_last);
    pid.err_last=pid.err;
    // if(pid.voltage>1)
    //     pid.voltage = 1;
    // if(pid.voltage<0)
    //     pid.voltage = 0;
    // incrementSpeed=pid.Kp*(pid.err-pid.err_next)+pid.Ki*pid.err+pid.Kd*(pid.err-2*pid.err_next+pid.err_last);
    // pid.ActualSpeed+=incrementSpeed;
    // pid.err_last=pid.err_next;
    // pid.err_next=pid.err;
    return pid.voltage;
}
```

## 2 串口通讯程序设计

下图为接收模型发过来的数据的相应代码：

```
// Receive Data Available or Character time-out
if ((tmp == UART_IIR_INTID_RDA) || (tmp == UART_IIR_INTID_CTI))
{ //接收

    for(i=0;i<MOTOR_RETURN_LEN;i++)
    {
        if(LPC_UART2->LSR & 0x01) //有有效的数据
        {
            if(gMOTOR.Index>=MOTOR_MAX) gMOTOR.Index = 0;
            gMOTOR.Buf[gMOTOR.Index] = LPC_UART2->RBR & 0xFF;
            if(gMOTOR.Buf[gMOTOR.Index] == 0x2E)
                gRUN._2E_NUM = gMOTOR.Index;
            gMOTOR.Index++;
        }
        else
        {
            break;
        }
    }

    if(gMOTOR.Buf[gMOTOR.Index-1] == 0x41)
    { //到达指定数量 进行处理
        MOTOR_Process(&gMOTOR);
    }
}
```

下图为处理输出反馈信号的代码：

```
void MOTOR_Process(struct MOTOR_BUF *m)
{
    float err, ret;
    int i;
    if(m->Buf[0] == 0x2D)
    {
        for(i=1;i<gRUN._2E_NUM;i++)
        {
            err += (m->Buf[i]-0x30)*My_Power(gRUN._2E_NUM-i-2);
        }

        err = -err;
    }
    else
    {
        for(i=0;i<gRUN._2E_NUM;i++)
            err += (m->Buf[i]-0x30)*My_Power(gRUN._2E_NUM-i-1);
    }
    ret = PID_realize(err);
    MOTOR_SEND(ret);
}
```

下图为控制信号输出到模型的代码：

```
void MOTOR_SEND(float e)
{
    uint8_t buff[8];
    int i;

    if(e<0)
    {
        buff[2] = 1;
        e=-e;
    }
    else buff[2] = 0;

    buff[0] = 0x01;
    buff[1] = 0x03;
    buff[3] = ((uint16_t)(e*1000)) & 0xff;
    buff[4] = ((uint16_t)(e*1000))>>8 & 0xff;
    buff[5] = 0x41;

    gMOTOR.Index = 0;
    for(i=0;i<MOTOR_MAX;i++)
    {
        gMOTOR.Buf[i]=0;
    }
    //发送数据
    Uart2RT(1); //切换到发送模式
    My_UART_SendBytes(2,buff,6);
    Uart2RT(0); //切换成接收模式
}
```