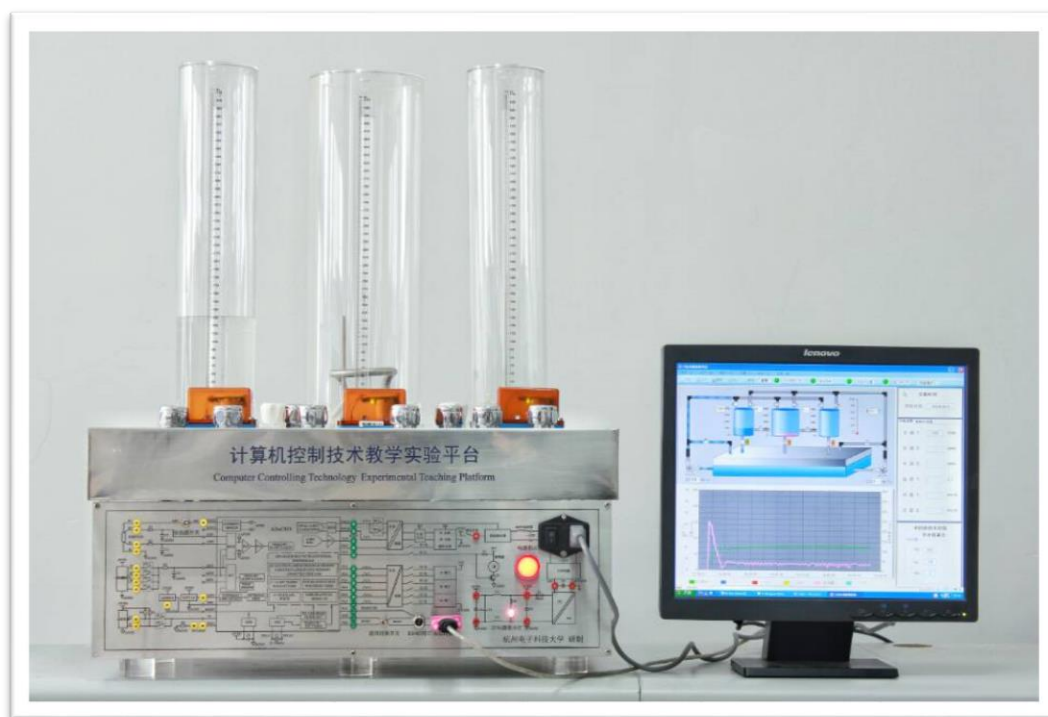




《计算机控制系统》

实验指导书



杭 州 电 子 科 技 大 学

2021 年 3 月

黄国辉、王剑、陈雪亭、夏宇栋

目 录

实验一 液位测量及误差校准实验.....	1
实验二 压力信号采集与数字滤波实验.....	8
实验三 单水箱液位 PID 控制实验	16
实验四 Matlab 混合编程和 PID 改进算法温度控制	24
实验六 基于组态王的温度 PID 控制实验（自动）	37

实验一 液位测量及误差校准实验

1. 实验目的

- 1) 学会分析本实验中液位的误差来源;
- 2) 学习并掌握基本的误差校准方法;
- 3) 学习并掌握“三容水箱参数标定实验软件”的操作。

2. 实验要求

- 1) 利用专用的“三容水箱参数标定实验软件”对液位进行校准;
- 2) 记录有关数据, 用手工进行计算与实验结果进行对比;
- 3) 记录实验数据, 分析实验结果, 并完成实验报告。

3. 系统误差的校准

3.1 误差来源

系统误差是指在相同条件下, 经过多次测量, 误差的数值(包括大小符号)保持恒定、或按某种已知的规律变化的误差。

因此, 原则上讲, 系统误差是可以通过适当的技术途径来确定并加以校正的。

在系统的测量输入通道中, 一般均存在零点偏移和漂移, 产生放大电路的增益误差及器件参数的不稳定等现象, 它们会影响测量数据的准确性, 这些误差都属于系统误差。

3.2 系统误差的校准原理

偏移校准在实际中应用最多, 并且常采用程序来实现, 称为数字调零。除了数字调零外, 还可以来用偏移和增益误差的自动校准

自动校准的基本思想是在系统开机后或每隔一定时间自动测量基准参数, 如数字电压表中的基准参数为基准电压和零电压, 然后计算误差模型, 获得并存储误差补偿因子。在正式测量时, 根据测量结果和误差补偿因子, 计算校准方程, 从而消除误差。全自动校准只适于基准参数是电信号的场合。并且它不能校正由传感器引入的误差。为了克服这种缺点, 可采用人工自动校准。

可采用其它的高精度仪器分别测出两个(两点间隔尽可能大)现时输入信号 x_1 、 x_2 的对应值 y_1 和 y_2 , 若输入信号为线性关系, 则有

$$y_1 = Mx_1 + b$$

$$y_2 = Mx_2 + b$$

则可以计算出放大系数 M 和偏移量 b 为

$$M = \frac{y_2 - y_1}{x_2 - x_1}$$

$$b = y_1 - \frac{y_2 - y_1}{x_2 - x_1} x_1$$

因此，若实际测量时输入信号 x_i ，则对应的测量输出值 y_i 可用下式计算

$$y_i = Mx_i + b$$

4. 实验电路原理

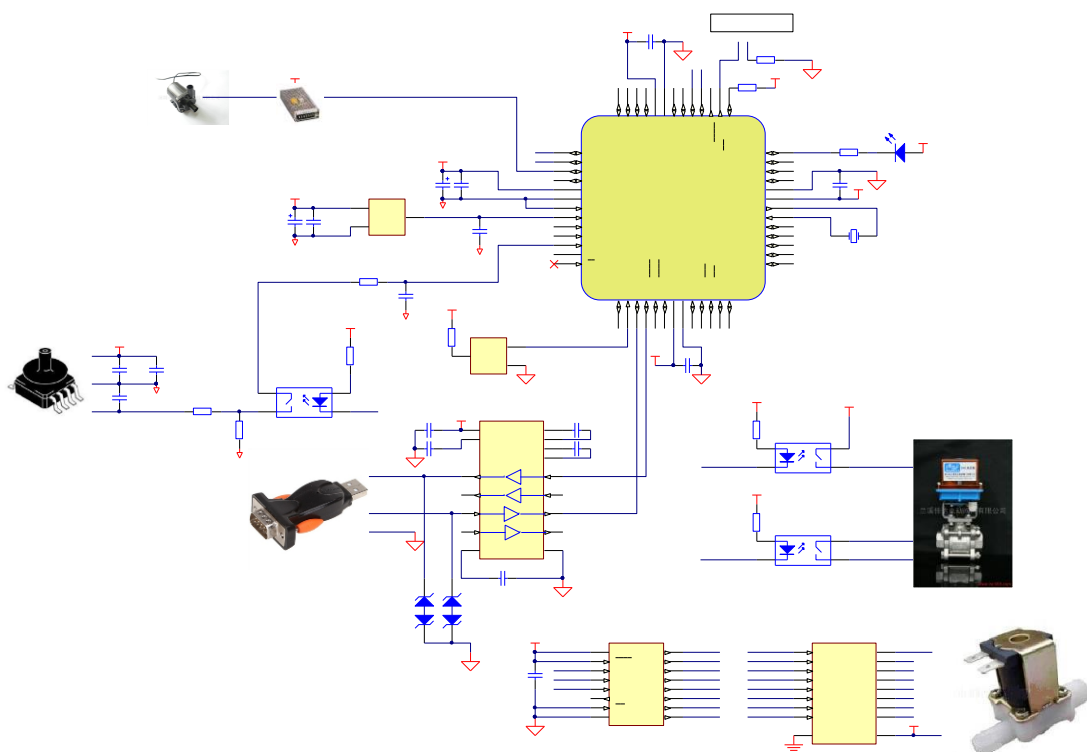


图 1 实验原理图

5. 液位采集原理以及数字滤波技术

5.1 液位传感器介绍

MPXV5004G 是一种先进的单片硅压力传感器。被测介质的压力直接作用于传感器的膜片上，使膜片产生与截止压力成正比的微位移，使传感器的电阻值发生变化，并用电路检测这一变化，转换输出一个对应于这一压力的标准测量信号。

MPXV5004G 内部除传感单元外，还包含信号调理器、温度补偿器和压力修正电路，特别适用于由单片机构成的检测系统。其芯片电路示意框图如图 2 所示。

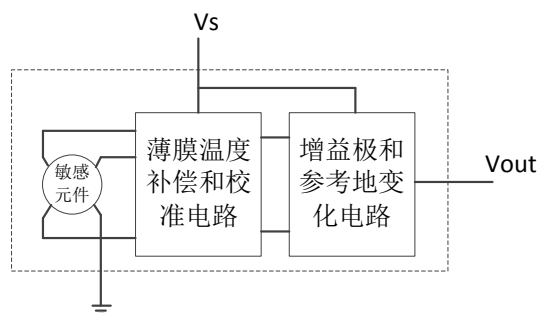


图2 压力传感器电路示意框图

MPXV5004G 压力传感器采用额定 5V 供电电压，最大测量压差为 3.92kPa（即 400mmH₂O），最大耐受压力为 16kPa，温度补偿范围为-30℃~100℃。工作温度为 10℃~60℃时，该压力传感器具有良好的线性输出关系，如式（1）所示：

$$V_{OUT} = (9.8H + 1000) \pm 75mV \quad (1)$$

其中， V_{OUT} 为输出电压， V_s 为工作电压（即电源电压）， H 为液位高度值 mm，最大误差为 75mV。

由物理常识易得，随着液位的上升，容器底部受到的压力相应增大。压力传感器采集到的模拟信号，通过 AD μ C834 中的 AIN3 单端输入，由 16 位高精度 AD 转换成数字信号。

5.2 水泵控制

在水泵控制电路中，AD μ C834单片机通过其DAC脚输出0~5V，并间接控制可调开关电源0~24V直流电压输出，实现水泵1功率和流量的控制。

5.3 电磁进水阀控制

通电时，电磁线圈产生电磁力把关闭件从阀座上提起，阀门打开；断电时，电磁力消失，弹簧把关闭件压在阀座上，阀门关闭。电磁阀控制主要利用单片机 I2C 的接口将数据以串行方式发给 74HC595 移位寄存器，再由 74HC595 并行输出给 ULN2803 方向放大控制输出。自动台的进水阀同一时间必须保证一个进水阀开启，其它电动进水阀是关闭的，如图 3 所示。

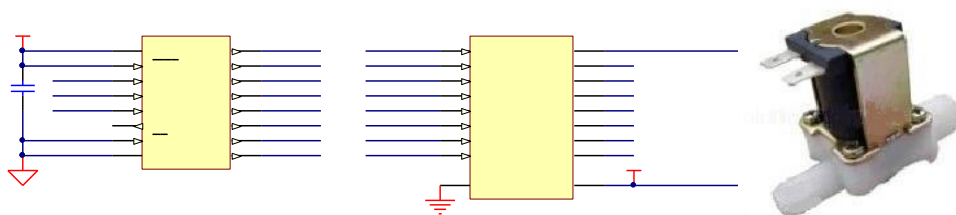


图3 基于 SN74HC595 的电磁阀控制电路

5.4 电动球阀的控制原理

在实验中放水是用电动阀来控制的。在本实验装置中的电动阀采用了24V直流控制的XHL系列电动阀门。其控制电路如图4所示。

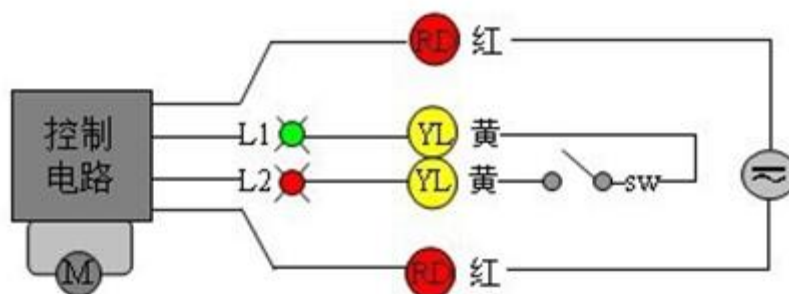


图 4 XHL 系列电动阀门控制电路

电动阀控制线共有四根线，红与黄各两根，红色用于电机的通断控制，电动阀2的开断由AD μ C834的P2.5口通过光耦继电器U12控制；两根黄线用于控制阀门开口度方向，由AD μ C834的P2.6口控制通过光耦继电器U13进行控制。当P2.6低电平时，光耦继电器闭合，则阀门内部电机正转，开口度变大，直到到最大之后自动停止，并到保持最大开口度，同时红色指示灯亮起；反之，当P2.6高电平时，光耦继电器断开，则阀门内部电机反转，开口度变小，直到到关闭自动停止，同时绿色指示灯亮起。当电动阀转动过程中，切断电源，则可以实现电动阀一定的开度设置。

经过精确计算，电动阀从全开到全关或从全关到全开用时均为 10s 左右，即 1s 的开度或关度为 10%。为此本电动阀控制就可以利用 AD μ C834 内部的 Time1 定时器实现开口度的控制。如果需要控制电动阀 20% 的开口度，那么需要定时 2s，根据该方法可以精确控制电动阀 0~100% 之间开度的控制了。

6. 三容水箱参数标定实验软件介绍

三容水箱参数标定实验软件是利用 C#语言开发设计的，实验设计参照计算机控制教学的课程要求完成。该软件主要有温度标定、液位 1 标定、液位 2 标定、液位 3 标定、流量 1 标定、流量 2 标定、共六个控制界面。打开软件，可以通过按住对应的按钮来对某个标定界面进行切换，然后做相应的标定实验。

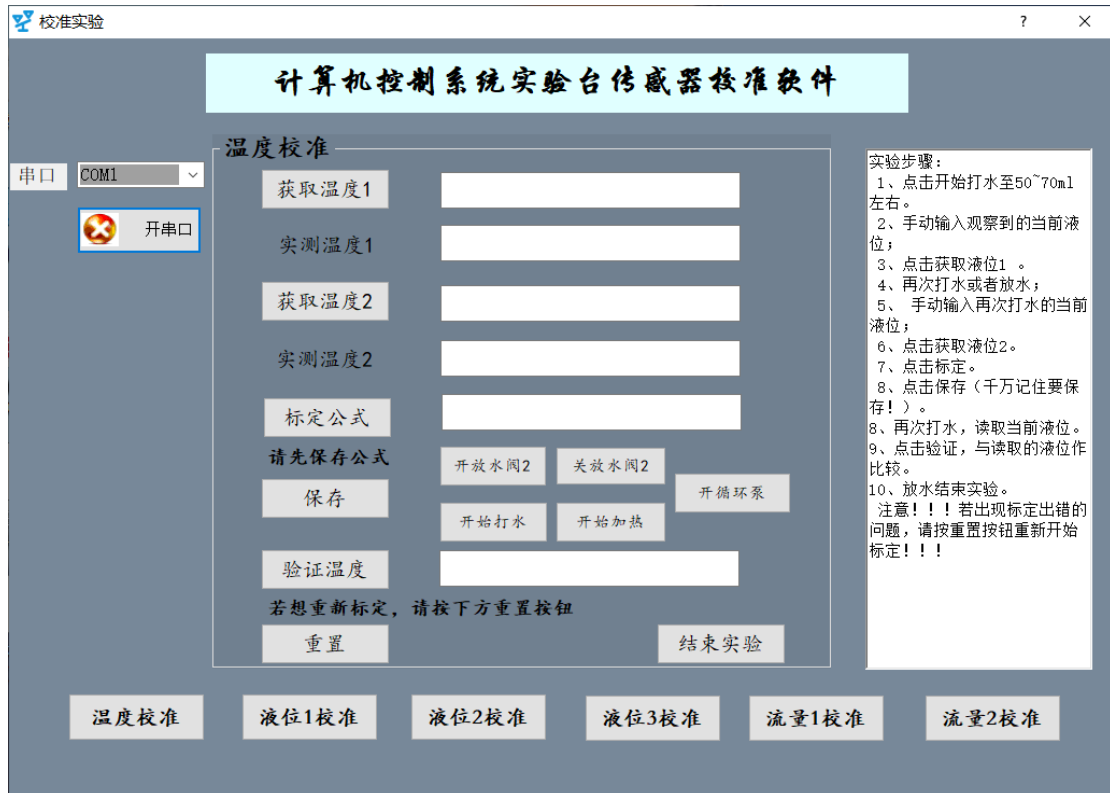


图 1 三容水箱参数标定界面(温度标定)

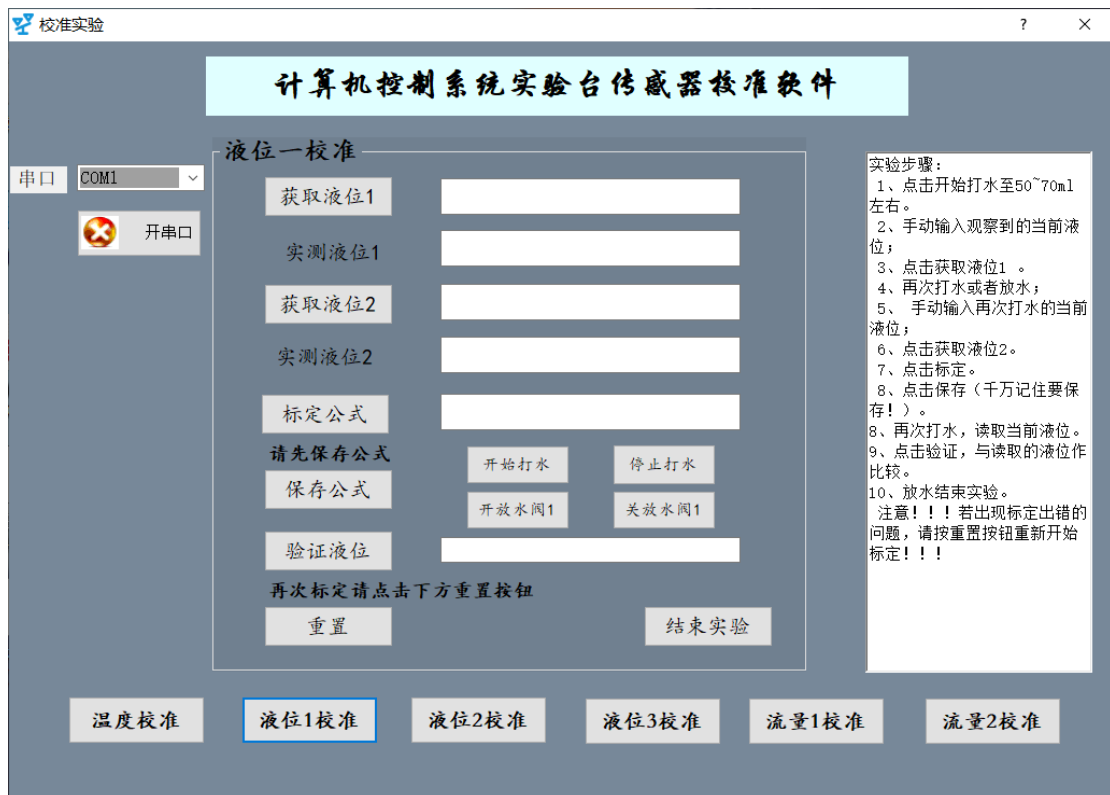


图 2 三容水箱参数标定界面(液位 1 标定)

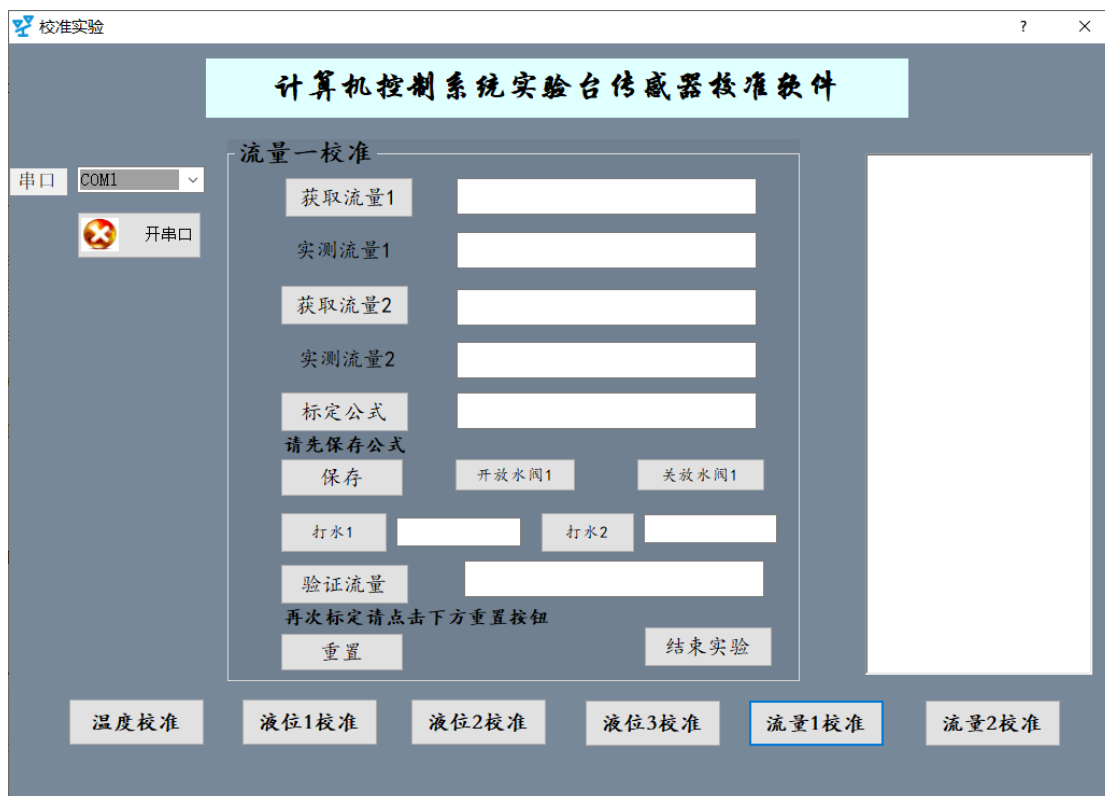


图 3 三容水箱参数标定界面（流量 1 标定）

7. 实验步骤

7.1 操作步骤

- 1) 打开桌面上标定实验软件 **传感器标定**，开串口；
- 2) 选择液位 1 标定，进入实验界面，请单击一次重置按钮还原单片机数据，再点击开始打水，当液位达到 50-70mm 左右时停止打水，点击获取液位 1 此时读取实际液位填入对应区间；然后再开始打水，当液位达到 180mm 左右时停止打水，点击获取液位 2 此时再读取实际液位填入对应区间；
- 3) 点击“标定公式”，出现标定的公式，之后该按钮不可使用（点击重置方可恢复）。
- 4) 然后点击“保存公式”，之后该按钮不可使用（点击重置以后方可恢复）。
- 5) 将水放掉一部分，再点击验证液位，观察实际液位值与采集的液位值进行比较，该过程可以进行多次，不断放水打水进行验证，保证实验数据完整性。
- 6) 若校准后的数据误差控制在 3mm 以内，则校准完成，最后点击“结束实验”
- 7) 若校准效果不好，请点击重置按钮，重新进行 2)～5) 步骤，直到满意为止。

7.2 数据记录

表 1 液位

次数	1	2	3	4	5	6	7	8	9	10
软件高度										
实际高度										

8. 问题与作业

- 1) 请说明通常情况下有几种类型的误差，并依次分析这几种类型误差的特点和产生的原因。
- 2) 分析影响液位测量精度的一些因素。
- 3) 如何有效减小测量过程中的误差。
- 4) 通常情况下误差校准的方法有哪些。
- 5) 请描述你的实验过程和步骤，并分析结果，完成实验报告。

实验二 压力信号采集与数字滤波实验

1. 实验目的

- 1) 学习压力信号采集;
- 2) 学习数字滤波技术;
- 3) 掌握压力测量数据的预处理。

2. 实验要求

- 1) 进行液位测量实验;
- 2) 将测量值控制在稳定状态后进行数据采集并记录每个稳定点, 要求采集 5 点以上;
- 3) 通过串口命令改变测量值重新进行测量;
- 4) 分别使用算术平均滤波法、中位值滤波法、递推平均滤波法、加权递推平均滤波法、一阶惯性滤波法、限幅滤波+算术平均滤波;
- 5) 利用最小二乘法对上述数字处理;
- 6) 记录实验数据, 分析实验结果, 并完成实验报告。

3 数字滤波原理

3.1 限幅滤波

限幅滤波又称程序判断法, 由于工业现场测控系统存在随机脉冲干扰, 通过变送器将尖脉冲干扰引入输入端, 从而造成测量信号的严重失真。

$$\overline{y_n} = \begin{cases} y_n, & \Delta y_n = |y_n - \overline{y_{n-1}}| \ll a \\ \overline{y_{n-1}}, & \Delta y_n = |y_n - \overline{y_{n-1}}| > a \end{cases}$$

在应用这种方法时, 关键在于 a 值的选择。因此, 通常按照参数可能的最大变化速度 V_{\max} 及采样周期 T 决定 a 值。

3.2 中位滤波

中位值滤波就是对某一被测参数连续采样 n 次 (一般 n 取奇数), 然后把 n 次采样值按大小排队, 取中间值为本次采样值。

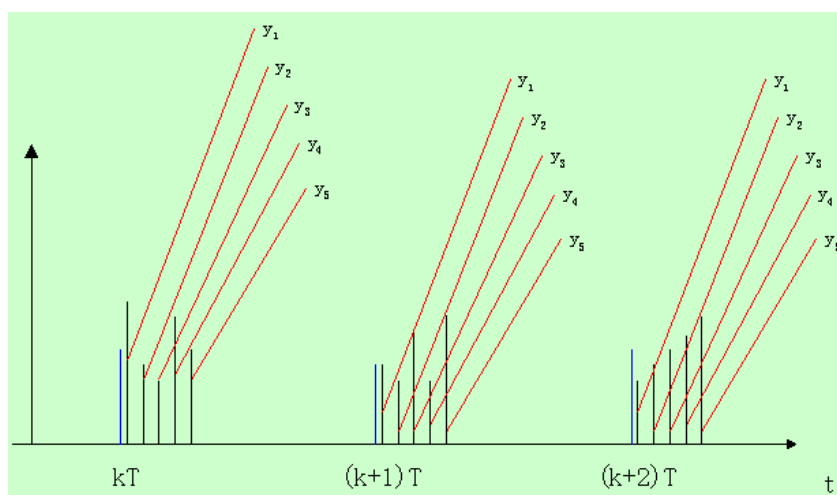


图 1 中位滤波

$$y_{n1} \leq y_{n2} \leq y_{n3} \leq y_{n4} \leq y_{n5}, \quad y_n = y_{n3}$$

中位值滤波能有效地克服因偶然因素引起的波动，采样器不稳定引起的误码等造成的脉冲干扰。对缓慢变化的过程采用中位值滤波有良好的效果。 n 越大，排序算法所占的时间越长。

3.3 算术平均滤波法

算术平均滤波法就是对采样数据 y_i 连续的 N 个测量值进行算术平均。其数学表达式为：

$$\bar{y}_n = \frac{1}{N} \sum_{i=0}^{N-1} y_{n-i}$$

算术平均滤波法适用于对一般具有随机干扰的信号进行滤波。

流量测量，通常取 $N=8 \sim 12$

压力测量，通常取 $N=4 \sim 8$

3.4 递推平均滤波法

上述的算术平均滤波法，每计算一次数据，需采样 N 次，对于采样速度较慢或要求数据计算速度较高的系统，该方法是无法使用的。

递推平均滤波法把 N 个采样数据看成一个队列，队列的长度固定为 N ，每进行一次新的采样，把采样结果放入队尾，而扔掉原来队首的一次数据。这种滤波算法称为递推平均滤波法，其数学表达式为：

$$\bar{y}_n = \frac{1}{N} \sum_{i=0}^{N-1} y_{n-i}$$

式中：

\bar{y}_n ——第 n 次采样值经滤波后的输出；

y_{n-i} ——未经滤波的第 $n-i$ 次采样值；

N——递推平均项数。

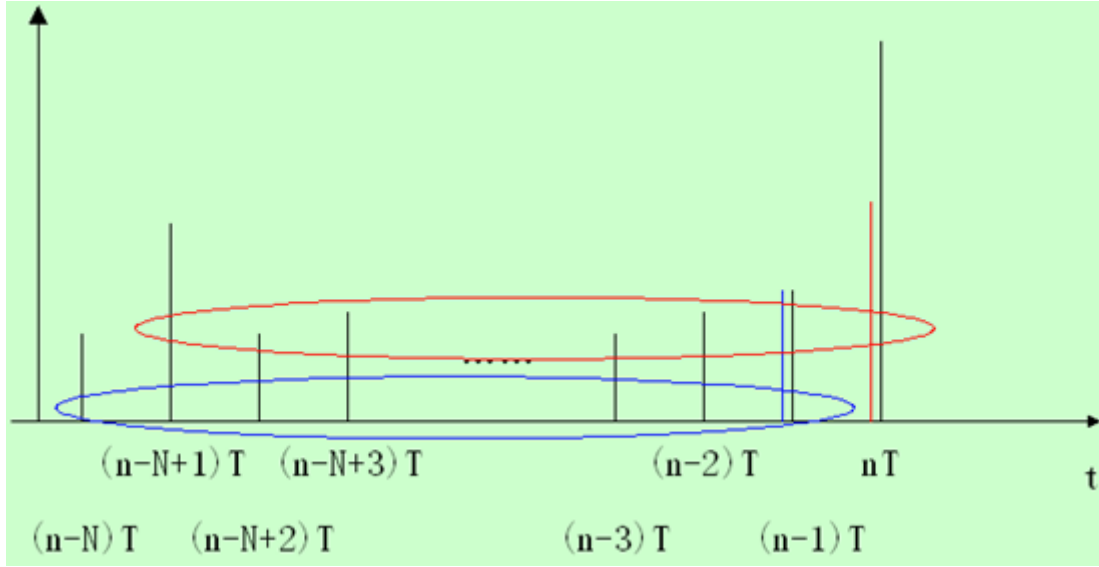


图 2 递推平均滤波法

递推平均滤波算法对周期性干扰有良好的抑制作用，平滑度高，灵敏度低；但对偶然出现的脉冲性干扰的抑制作用差，不易消除由于脉冲干扰引起的采样值偏差，因此它不适用于脉冲干扰比较严重的场合，而适用于高频振荡的系统。

3.5 加权递推平均滤波法

算术平均滤波法和递推平均滤波法中，N 次采样值在输出结果中的比重是均等的，即 $1/N$ 。用这样的滤波算法，对于测量信号会引入滞后，N 越大，滞后越严重。

为了增加最新采样数据在递推平均中的比重，以提高系统对当前采样值的灵敏度，可以采用加权递推平均滤波算法。

加权递推平均滤波算法是递推平均滤波算法的改进，即不同时刻的数据加以不同的权，通常越接近当前时刻的数据，权取得越大，N 项加权递推平均滤波算法为：

$$\bar{y}_n = \sum_{i=0}^{N-1} C_i y_{n-i}$$

式中， C_0, C_1, \dots, C_{N-1} 为常数，且满足条件：

$$C_0 + C_1 + \dots + C_{N-1} = 1 \text{ 并 } C_0 > C_1 > \dots > C_{N-1} > 0$$

常数 C_0, C_1, \dots, C_{N-1} 的选取有多种方法，其中最常用的是加权系数法。设 τ 为对象的纯滞后时间，且

$$R = 1 + e^{-\tau} + e^{-2\tau} + \dots + e^{-(N-1)\tau}$$

$$C_0 = \frac{1}{R}, C_1 = \frac{e^{-\tau}}{R}, \dots, C_{N-1} = \frac{e^{-(N-1)\tau}}{R}$$

其中 τ 为对象的纯滞后时间。

所以加权递推平均滤波算法适用于有较大纯滞后时间常数的对象和采样周期较短的系统，而对于纯滞后时间常数较小、采样周期较长、变化缓慢的信号，则不能迅速反映系统当前所受干扰的严重程度，滤波效果差。

3.6 一阶惯性滤波法

在模拟量输入通道等硬件电路中，常用一阶惯性 RC 模拟滤波器来抑制干扰，当用这种模拟方法来实现对低频干扰的滤波时，首先遇到的问题是要求滤波器有大的时间常数和高精度的 RC 网络。时间常数 T 越大，要求 RC 值越大，其漏电流也随之增大，从而使 RC 网络的误差增大，降低了滤波效果。

而一阶惯性滤波算法是一种以数字形式通过算法来实现动态的 RC 滤波方法，它能很好地克服上述模拟滤波器的缺点，在滤波常数要求大的场合，此法更为实用。

一阶惯性滤波算法为

$$\bar{y}_n = (1-a)y_n + a\bar{y}_{n-1}$$

$$a = \frac{T_f}{T+T_f} = \frac{1}{1+\frac{T}{T_f}}$$

式中

y_n ——未经滤波的第 n 次采样值；

T_f ——滤波时间常数；

T ——采样周期。

根据一阶惯性滤波的频率特性，若滤波系数 T_f 越大，则带宽越窄，滤波频率也越低。因此，需要根据实际情况，适当选取 a 值，使得被测参数既不出现明显的波纹，反应又不太迟缓。

3.7 最小二乘法

最小二乘法（又称最小平方法）是一种数学优化技术。它通过最小化误差的平方和寻找数据的最佳函数匹配。利用最小二乘法可以简便地求得未知的数据，并使得这些求得的数据与实际数据之间误差的平方和为最小。最小二乘法还可用于曲线拟合，其他一些优化问题也可通过最小化能量或最大化熵用最小二乘法来表达。最小二乘法是解决曲线拟合问题最常用的方法。

基本思路：令

$$f(x) = \alpha_1\varphi_1(x) + \alpha_2\varphi_2(x) + \cdots + \alpha_m\varphi_m(x)$$

其中， $\varphi_k(x)$ 是事先选定的一组线性无关的函数， α_k 是待定系数，

($k=1, 2, \dots, m < n$), 拟合准则是使 $y_i (i = 1, 2, \dots, n)$ 与 $f(x_i)$ 的距离 δ_i 的平方和最小, 称为最小二乘准则。

基本原理: 设 (x, y) 是一对观测量, 且 $x = [x_1, x_2, \dots, x_n]^T \in R^n, y = R$ 满足以下的理论函数

$$y = f(x, \omega)$$

其中, $\omega = [\omega_1, \omega_2, \dots, \omega_n]^T$ 为待定参数. 为了寻找函数 $f(x, \omega)$ 的参数 ω 的最优估计值, 对于给定 m (通常 $m > n$), 观测数据 $(x_i, y_i) (i = 1, 2, \dots, m)$, 求解目标函数

$$L(y, f(x, y)) = \sum_{i=1}^m [y_i - f(x_i, \omega_i)]^2$$

取最小值的参数 $\omega_i (i = 1, 2, \dots, n)$ 。求解的这类问题称为最小二乘问题, 求解该问题的方法的几何语言称为最小二乘拟合。

对于无约束最优化问题, 最小二乘法的一般形式为:

$$\min f(x) = \sum_{i=1}^m L_i^2(x) = \sum_{i=1}^m L_i^2[y_i, f(x_i, \omega_i)] = \sum_{i=1}^m [y_i - f(x_i, \omega_i)]^2$$

其中 $L_i(x) (i = 1, 2, \dots, m)$ 称为残差函数。当 $L_i(x) (i = 1, 2, \dots, m)$ 是 x 的线性函数时, 称为线性最小二乘问题, 否则称为非线性最小二乘问题。

最小二乘优化问题: 在无约束最优化问题中, 有些重要的特殊情形, 比如目标函数由若干个函数的平方和构成, 这类函数一般可以写成:

$$F(x) = \sum_{i=1}^m f_i^2(x), x \in R^n$$

其中 $x = (x_1, x_2, \dots, x_n)$, 通常要求 $m \geq n$ 我们把极小化这类函数的问题:

$$\min F(x) = \sum_{i=1}^m f_i^2(x)$$

称为最小二乘优化问题。最小二乘优化是一类比较特殊的优化问题。

4 实验步骤

4.1 操作步骤



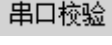
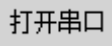

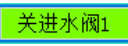
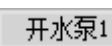
打开桌面上的计算机控制实验台专用串口调试助手图标，串口调试助手界面如图 3 所示。



图 3 串口调试助手界面图

- 1) 串口设置：选择对应的串口号，并选择波特率 9600 ，然后依次单击“串口校验”按钮 、“打开串口”按钮 。
- 2) 打开进水阀 1：在进水阀控制框中选择“开进水阀 1”按钮 ，则打开进水阀 1（即电磁阀 1）。并且单击以后会变成“关进水阀 1” ，实现进水阀实时可以关闭的状态。
- 3) 水泵 1 打水：在水泵控制框中选择“开水泵 1”按钮 ，并使水

位上升一段。

4) 关闭水泵 1: 在水泵控制框中选择“关水泵 1”按钮 关水泵1 停止打水。

5) 记录实时液位值: 在 AD 采样控制框中选择“AD 采样”按钮 AD采样 读取液位高度 (注: 所有 A/D 采集数据均为十六进制值), 采样数据的前两位为液位 1 的高度值, 需要转化成十进制数填入下表, 重复读取 5 次。

6) 电压测量: 使用万用表将表笔插入计算机控制实验台上 DAC 和接地之间就可以得到 DAC 的电压值, 重复读取 5 次。

7) 重复操作: 重复 3) ~6) 上述步骤。请进行 5 次不同液位的数据采集。

8) 放水阀 1 控制: 在放水阀控制框中选择“开放水阀 1”按钮 开放水阀1 进行放水, 此时“开放水阀 1”的会改变成“停放水阀 1”, 此时按下这个 停放水阀1 按钮就可以使得放水阀 1 就可以停下。若继续点击这个该按钮, 就会继续打开放水阀 1。而如果想要关放水阀, 点击下方“关放水阀 1”的指令, 此时 关放水阀1 按钮就会变成 停放水阀1, 以此来实现一点一点增大放水阀的开度或者关闭程度, 做好增量实验。电动放水阀打开或关闭过程中会发出声音, 但阀门到达最大或最小时, 自动停止。

4.2 数据记录

表 1 实验数据记录表格

组号	实验内容 记录次数	电压值 (1#— AGND)	液位采集高度
第一个液位	1		
	2		
	3		
	4		
	5		
第二个液位	6		
	7		
	8		
	9		
	10		
第三个液位	11		
	12		
	13		

	14		
	15		
第四个液位	16		
	17		
	18		
	19		
	20		
第五个液位	21		
	22		
	23		
	24		
	25		

1) 对液位采集的数据按照要求（算术平均滤波法、中位值滤波法、递推平均滤波法、加权递推平均滤波法、一阶惯性滤波法、限幅滤波）进行处理。

绘制关系曲线，并用最小二乘法对上述数字处理找出关系式；

2) 分析比较各种方法的数据处理结果。

5. 问题与作业

- 1) 请论述数字滤波的特点以及应用场合。
- 2) 绘制关系曲线，并用最小二乘法对上述数字处理找出关系式。
- 3) 分析误差原因和影响测量精度的因素。
- 4) 分析结果，完成实验报告。
- 5) 预习并准备实验三内容。

实验三 单水箱液位 PID 控制实验

1. 实验目的

- 1) 了解水泵控制及液位测量原理；
- 2) 了解单水箱一阶液位控制系统的工作原理；
- 3) 掌握 PID 控制算法编程及其控制参数的调整方法；
- 4) 学习三容水箱计算机控制实验台控制软件的使用方法。

2. 实验要求

- 1) 掌握 PID 控制算法各控制分量的作用；
- 2) 掌握数字 PID 算法控制原理；
- 3) 学会对模拟 PID 控制器推导其离散化控制算法；
- 4) 利用三容水箱计算机控制实验台自带控制软件，完成单水箱一阶液位 PID 控制参数的调整，达到控制最优效果；
- 5) 观察实验数据，并完成实验报告。

3. 系统结构框图

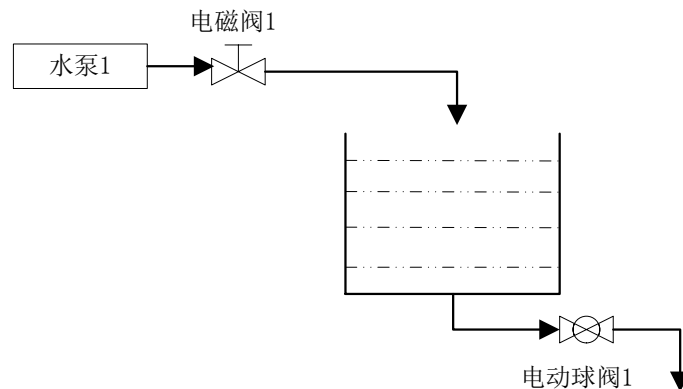


图 1 单水箱一阶液位控制结构图

4. 实验原理

水泵 1 和电动球阀 1 是控制水箱液位的主要控制器。水泵 1 控制进水口的流量，作为入水口的控制，通过电压的改变控制其出水流量；电动球阀 1 作为放水口控制出水口的流量；电磁阀 1 作为进水的总控制，在整个实验期间全开。在本实验中，设置电动球阀 1 开度不变，通过改变水泵 1 的电压实现液位控制。

4.1 电路原理

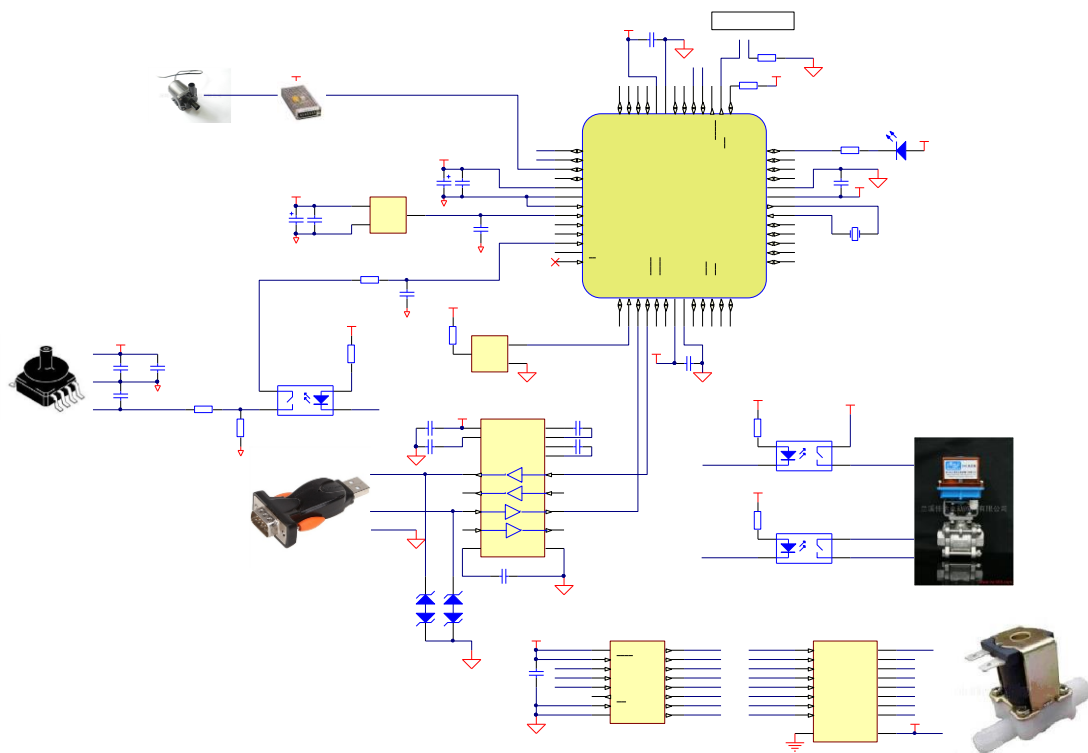


图 2 单水箱液位采集与控制电路

在水泵控制电路中，ADμC834单片机通过其DAC脚输出0~5V，并间接控制可调开关电源0~24V直流电压输出，实现水泵1功率和流量的控制。

5. 实验步骤

5.1 实验设备

三容水箱计算机控制实验台、万用表、计算机、RS232/RS485 串口线，电源线、Matalb 软件和三容水箱计算机控制软件。

5.2 设备连接与检查

- 1) 将计算机控制技术实验台的水槽中灌满水（至溢出口的下沿）。
- 2) 连接 RS232/RS485 串口线，将串口选择开关拨向使用的串口类型的方向。
- 3) 连接电源线，将电源插头接到 220V 电源上。
- 4) 打开电源，电源指示灯亮。
- 5) 使用万用表检查 5V，24V，220V 电源是否正常。

5.3 操作步骤

5.3.1 液位软件自动标定操作（若刚做过实验一液位标定实验，则该步骤可省略）



1) 打开计算机控制系统实验软件图标。即可打开实验软件，如图 3 所示。用户名请输入自己的姓名（请不要输入中文，可以输入如 zhangqiang0914），将存储在数据库中，用于历史查询。设备类型请选择“全自动水箱”。

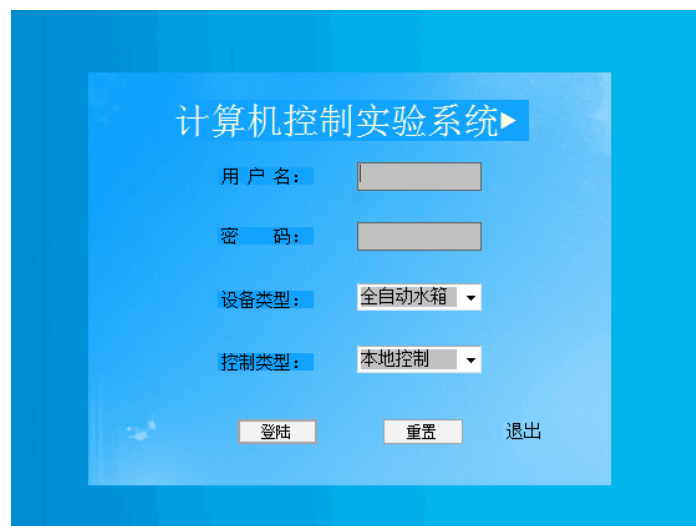


图 3 系统登陆图

注意：

1) 在实验过程中可以直接点击 FLASH 中的电动阀和水泵，就可以直接对其进行开关控制。如图 5 所示。

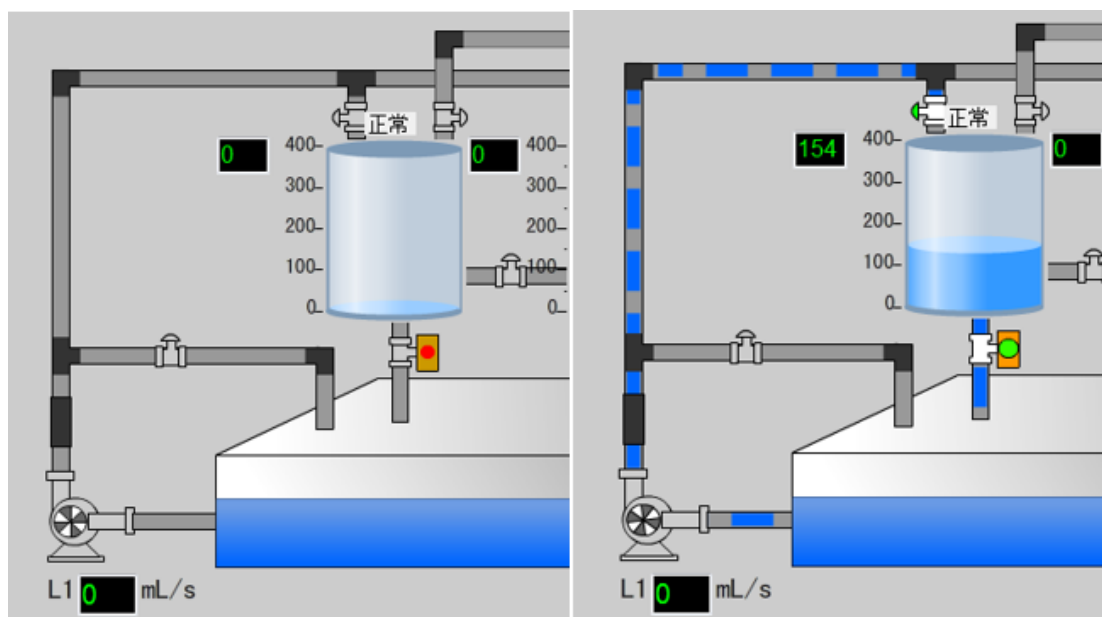


图 5 在 FLASH 中直接点击水泵或阀门进行控制

2) 若实验过程中发现液位数据不准, 请用实验一的方法先进行标定。

5.3.2 直接利用控制软件进行 PID 控制

1) 实验前将水箱中的水清空。方法 1: 直接在 FLASH 中点击电动后将其打开;
方法 2: 停止实验, 或重启实验二程序; 方法 3: 关闭并启动实验台总开关电源。

2) 进入系统主界面, 单击工具栏的“新建”或菜单栏的“系统”的“新建实验”, 如图 6 所示。

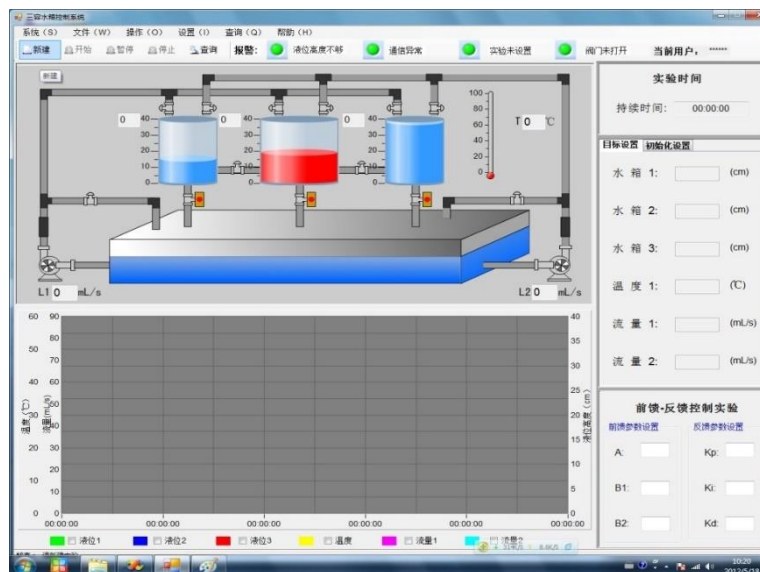


图 6 主界面

3) 进入实验选择窗口, 选择“单回路简单控制实验”, 如图 7 所示。以一阶液位控制实验为例, 进入实验设置窗口, 选择“一阶液位控制实验”。实验信息框中提示实验的内容, 实验信息会提示你是否支持混合编程, 如果没有提示支持混合编程请**不要**在“是否选择混合汇编”上打勾。窗口中还给出了实验的原理框图。本实验采用水泵作为控制执行部件, 电动阀采用固定开口度进行实验, 因此采用默认的控制选择——水泵控制。

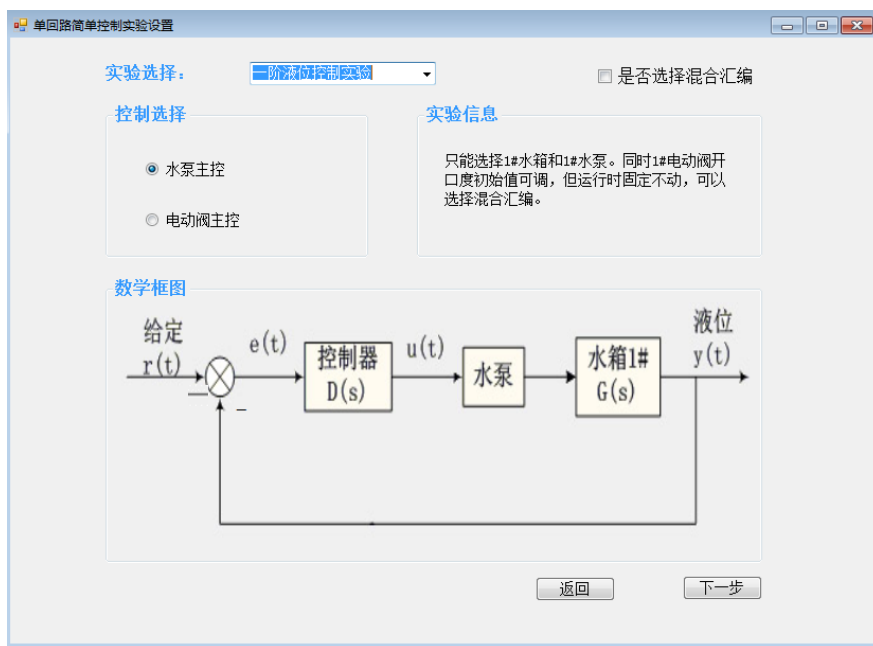


图 7 实验选择

4) 单击“下一步”，进入实验设置窗口，输入液位目标值及球阀开口度的初始值，在算法选择框中选择“普通 PID 控制算法”，如图 8 所示。电动阀开度建议 30~40%。



图 8 参数设置

5) 单击“确定”，完成实验设置，在弹出的实验提示窗口中，按照提示的步骤完成相应的操作进入下一步，勾选已完成上述选择，再点击确定按钮。注：自动台可以忽略图 9 提示，选勾确定即可。

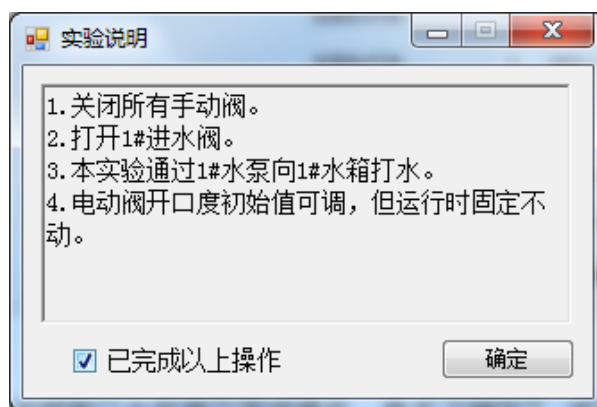


图 9 实验说明

6) 进入实验主界面后，在右下角的 PID 设置窗口中，用凑试法选择 K_p 、 K_i 、 K_d (K_p 推荐值在 1000 左右， K_i 在 10 左右，实验过程中建议不要改变 PID 的参数值) 然后点击“开始”按钮，实验开始。如图 10 所示。



图 10 PID 设置

7) 在单击“开始”按钮或者菜单栏“操作”的“开始实验”进行实验后，观察实验现象并记录数据，运行一段时间后单击“暂停”实验按钮，然后单击“查询”进入查询项记录数据并填入表格。用户也可保存曲线并导出数据到 Excel，通过 Excel 查看采集到的数据。

8) 建议每次参数设置后中途不要修改 PID 参数。每次实验完毕请将水箱里的水放完后，最后单击“停止”按钮，电动球阀关闭后，然后进入“新建”并重复上述实验步骤，以找出最佳 PID 控制参量。

6. 数据记录

通过上述实验，记录保证预设液位一定的条件下，通过修改 PID 控制参数实现对液位控制的不同效果，观察实验现象，并记录数据。

1) 保证在积分参数 K_i 和微分参数 K_d 都为 0 的前提下，逐步加大比例参数 K_p 的大小，观察系统的响应，记录不同比例参数 K_p 下的系统静差、超调量、调节时间，填入表 1。

表 1 实验数据记录

K_p						
ess						
σ_p						
t						

通过对实验数据的整理找出静差和超调量在合理范围内的 K_p 的值，最优比例系数可由此确定。

2) 在比例调节器控制实验的基础上，加上积分作用“ K_i ”，同样逐渐加大。以验证系统在 PI 调节器控制下，系统无静差产生。重新新建实验，重复实验步骤，并观察系统的响应，记录不同积分参数 K_i 下的系统静差、超调量、调节时间，填入表 2。

表 2 实验数据记录

K_i						
ess						
σ_p						
t						

通过对参实验数据的整理找出静差为 0 和超调量在合理范围内的 K_i 的值，在此过程中可以通过响应曲线的好坏反复改变比例系数和积分系数，以期得到满意的控制过程与整定参数。

3) 在 PI 调节效果不佳时可以增加 K_d 微分作用的调节，记录不同微分参数 K_d 下的系统静差、超调量、调节时间并填入表 3。

表 3 实验数据记录

K_d						
ess						
σ_p						
t						

4) 将最后实验确定的 PID 控制参数重新设置到上位机验证结果并观察实验现象记录数据到表 4 中，分析其系统静差、超调量、调节时间。

表 4 K_p K_i K_d 性能指标

ess	σ_p	t
-----	------------	---

--	--	--

7. 注意事项

- 1) 实验前将水箱中的水清空。
- 2) 若要修改 PID 参数，需要新建实验后修改。
- 3) 球阀的开度不要太小，否则会导致流量控制的过程中，水箱中的水位过高。但也不要超过 40%，否则容易出现震荡。
- 4) 在查询数据时先单击暂停按钮再去查询，查询完成后再单击停止按钮。
- 5) 实验配置完 PID 参数后点击开始实验后，实验未停止之前不应调整 PID 参数的设置，应在每次实验结束后，当再次开始实验时，修改 PID 参数。
- 6) 实验数据需要多次测量以减小误差。

8. 问题与作业

- 1) 逐步增大比例参数 K_p ，分析 K_p 对液位达到稳定时的影响和作用。
- 2) 在确定 K_p ，增大积分参数 K_i ，分析 K_i 对消除静态误差的作用。
- 3) 在已确定 K_p ， K_i 的基础上，增加微分参数 K_d ，分析对液位控制的影响及作用。
- 4) 通过上述数据表，分析 P、PI、PID 控制对稳态误差、超调量的影响。
- 5) 请描述你的实验过程和步骤，并分析结果，完成实验报告。

实验四 Matlab 混合编程和 PID 改进算法温度控制

1. 实验目的

- 1) 熟悉温度 PID 控制原理;
- 2) 熟悉单水箱温度控制工作原理;
- 3) 了解 Matlab 的 M 文件编写方法和利用 Matlab 制作相应的 dll 文件, 掌握 C#-Matlab 混合编程技术;
- 4) 学习 PID 改进算法的温度控制。

2. 实验要求

- 1) 先用 C#上位机软件完成常规 PID 温度控制;
- 2) 利用 Matlab 编写改进 PID, 实现水箱温度快速高品质控制;
- 3) 观察实验数据, 分析比较并完成实验报告。

3. 实验电路

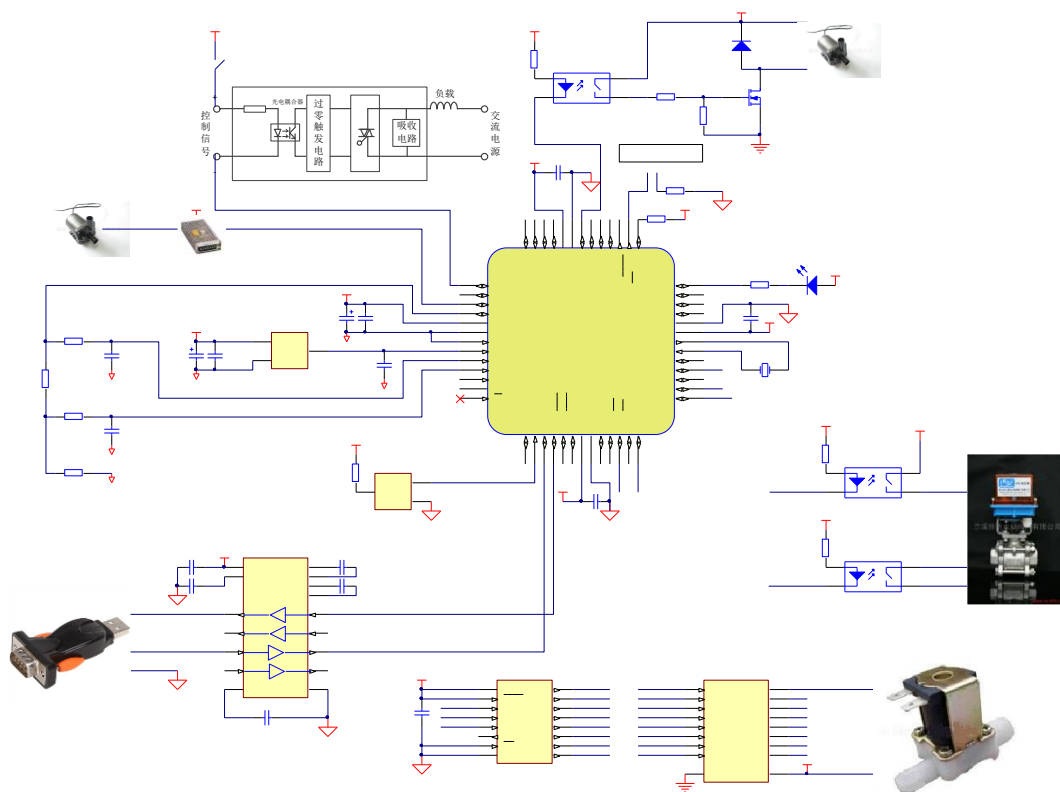


图 1 温度测控电路

其它说明:

- 1) 水泵 1 由 AD μ C834 的 DAC 输出控制开关电源输出 0~24V 变化电压控制

其流量；

2) 搅拌水泵由 P0.3 输出控制；

3.1 循环水泵的控制

该实验循环水泵直接由 P2.0 口通过光耦继电器 U19，驱动场效应管 Q7 控制水泵开关，如图 1 所示。当 P2.0 输出低电平时，光耦继电器 U19 闭合，场效应管 Q7 接通水泵电机的低电平，水泵打水；反之，水泵停止。

3.2 加热器控制

固态继电器(简称为 SSR)是一种四端有源器件，图 2 为固态继电器的结构。输入输出之间采用光耦进行隔离，过零触发电路可在交流电压变化到零状态附近时向晶闸管器件发出触发信号，从而导通负载的交流电源。

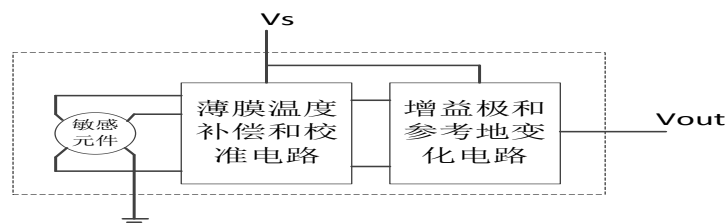


图 2 固态继电器结构

在加热控制电路中，采用交流电压600W的加热管，单片机通过PWM信号控制S208ZB固态继电器的通断，从而实现加热的需求。其控制电路如图3所示。

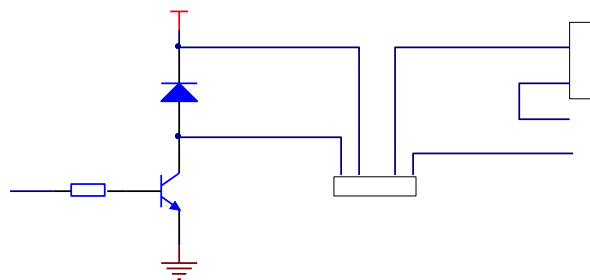


图3 加热控制电路图

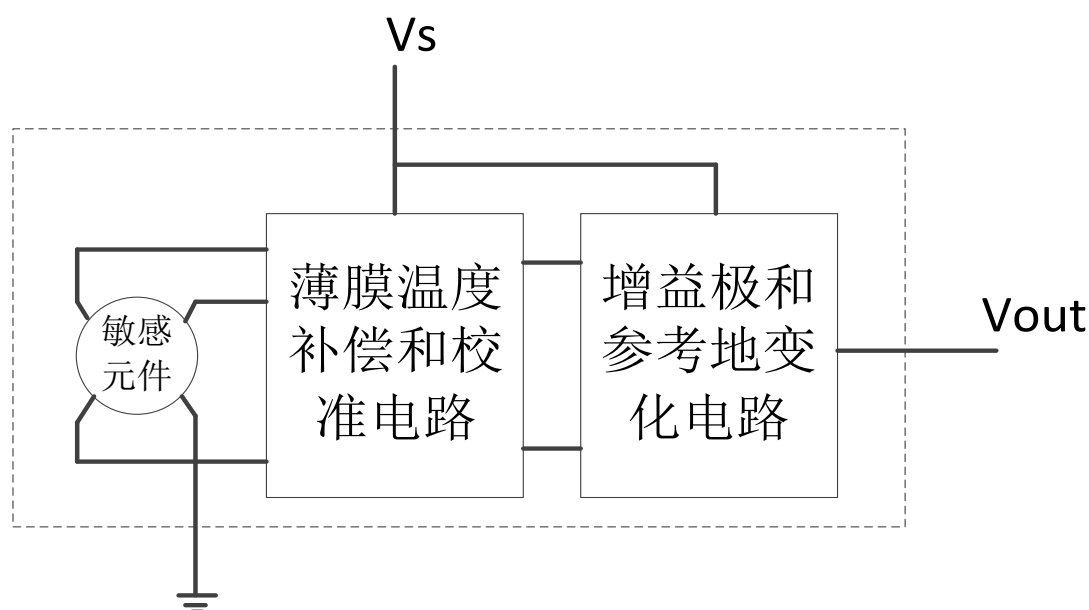


图4 PWM控制原理图

在控制电路中，采用PWM控制固态继电器实现加热控制。PWM（Pulse Width Modulation）即脉宽调制。它是利用数字电路产生的方波，通过控制其输出的占空比，对模拟电路进行控制的技术，广泛应用于测量、通讯、功率控制等许多领域。

PWM实际是微控制器PWM输出模块的工作在方式1状态下取得的，该模式下，PWM周期取决于寄存器PWM1L/H值、PWM时钟频率以及分频值。目前本控制器的PWM时钟频率就等于32768Hz，分频系数为1，PWM1L/H=8000H，也就是32768，因此PWM控制周期就1秒。而占空比取决于PWM0L/H=0~8000H。当PWM1H=80H，即128时，则占空比为100%，即全功率输出。

在功率控制中常与固态继电器联合使用，通过控制算法，改变PWM波的占空比来实现对加热器等设备的功率控制，从而达到系统的目标控制。

4. 实验步骤

4.1 实验设备

三容水箱计算机控制实验台、万用表、计算机、RS232/RS485 串口线，电源线、Matlab 软件和三容水箱计算机控制软件。

4.2 设备的检查和连接

1) 连接 RS232 串口线，将串口选择开关拨向使用的 RS232 串口类型的方

向。

- 2) 连接电源线，将电源插头接到 220V 电源上。
- 3) 打开电源，电源指示灯亮。
- 4) 使用万用表检查 5V，24V，220V 电源是否正常。

4.3 基于 C#软件常规 PID 温度实验



- 1) 打开计算机控制系统实验软件，进入登陆界面。如图 5 所示。
用户名可以输入自己的姓名，将存储在数据库中，用于历史查询。



图 5 系统登陆图

- 2) 图 6 为系统主界面，包括新建实验、查询、设置、操作等。

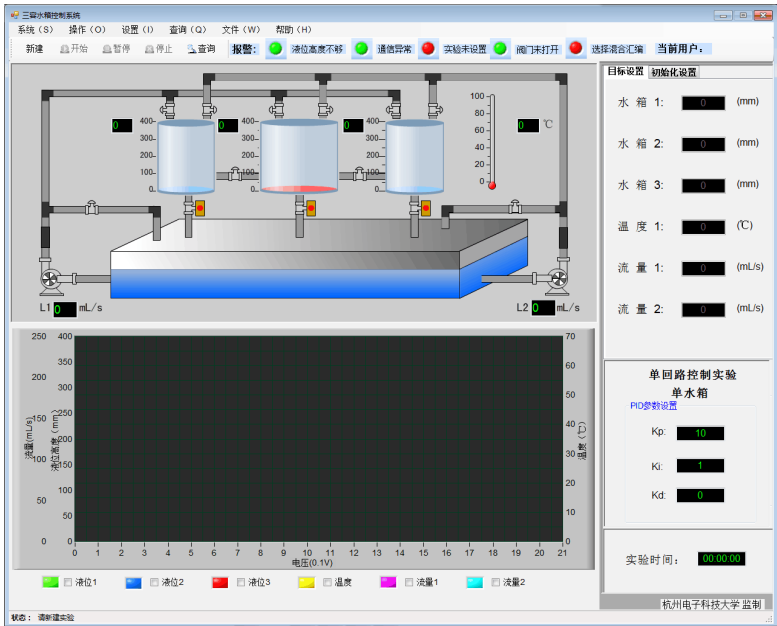


图 6 系统主界面

3) 然后单击“设置”按钮，进入“高级设置”的“串口设置”的界面如图3所示，选择合适的端口。端口的选择可以通过右击计算机进入管理选项查看；

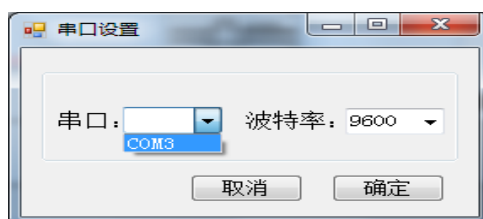


图3 串口设置界面

4) 如图4所示为新建实验后弹出的实验选择界面，本实验应选择“单回路简单控制实验”。



图4 实验选择界面

5) 图5为“单回路简单控制实验”实验界面，本实验选择中选择“温度控制试验”和混合编程，右侧“实验信息”框中会提示实验的功能，左侧可选择控制对象。

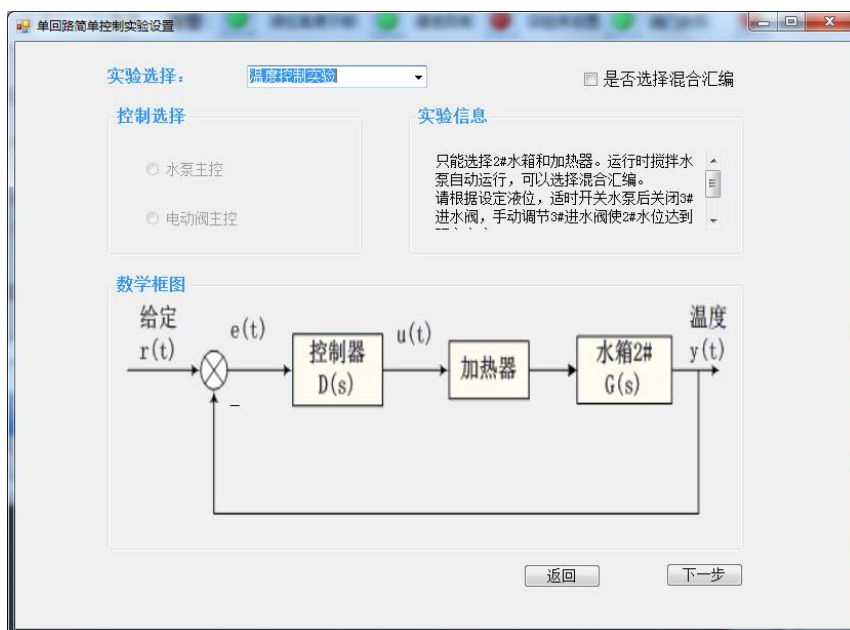


图 5 温度控制实验

6) 图 6 为“单回路简单控制实验”设置界面，可选择“普通 PID 控制算法”，输入温度和水箱 2#液位。

图 6 系统特性参数设置界面

7) 图 7 为实验操作完成提示界面，单击“确定”，可返回实验界面。若自动型实验台直接选择确定即可。

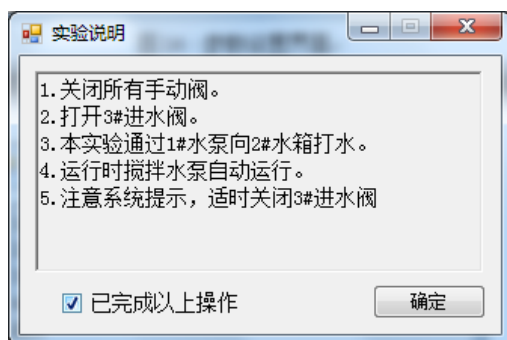
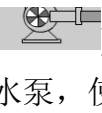
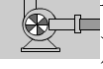


图 7 实验说明

8) 在开始试验之前先点击 FLASH 主界面右下角(), 开启水泵进行打水, 达到预定水位高度 180mm 后再点击()关闭水泵, 使 2#水箱水位在预定高度。

9) 单击“开始”按钮或者菜单栏“操作”的“开始实验”进行实验, 观察实验现象并记录数据, 运行一段时间后单击“暂停”实验按钮, 然后单击“查询”进入查询项, 记录数据并填入表格。用户也可保存曲线并导出数据到 Excel, 通过 Excel 查看采集到的数据。

10) 通过凑试法修改 PID 参数, 直到实验控制精度达到满意(以精度为主), 并记录 K_P 、 K_i 、 K_d 参数。

4.4 数据记录

通过上述实验, 记录在保证预设液位一定的条件下, 通过修改 PID 控制参数实现对温度控制的不同效果, 观察实验现象, 并记录数据。

表 1 实验数据记录

参数	第一次	第四次	第三次	第四次
K_p				
K_i				
K_d				
ess				
σ_p				

4.5 C#-Matlab混合编程原理

利用 Matlab 软件, 自行编写算法, 生成相应的 dll 文件, 上位机 (C#编写) 调用 dll 文件实现相应的控制策略 (整个过程可以理解为 Matlab 生成的 dll 相当于一个函数, 上位机调用该函数并获得该函数的返回值。)

类 C_Matlab_Mixed_Compilation 中包含的具体方法 (函数) 如下:

EasyFlowPID (简单流量 PID)、EasyOnePID (简单一阶液位 PID)、

EasyTwoPID（简单二阶液位 PID）、EasyTempPID（简单温度 PID）、
CompOnePID（复杂一阶液位 PID）、CompTwoPID（复杂二阶液位 PID）、
CascadeOnePID（一阶串级液位 PID）、CascadeTwoPID（二阶串级液位 PID）、
Feedfor_backOnePID（前馈-反馈液位 PID）、Example（示例编程 PID）。

EasyTempPID.m 文件存储的是混合编程示例程序，其它的文件只有函数入口，剩下的程序需要同学们根据不同的实验自行编写算法。

样例常规 PID 程序说明如下：需要修改

```
%-----函数入口
function result =EasyTempPID(Set_Temp,Actual_Temp,i)
% 此处 result 是该函数的输出参数；EasyTempPID 是函数名；
Set_Temp,Actual_Temp,i 是 3 个输入参数。
```

% Set_Temp 是上位机设定的被控对象温度的值；Actual_Temp 是上位机采集到的实际温度的值；i 为标记参数，第一次调用该函数时 i=0，即对你所要用的全局变量 e、u 的初始化，第二次及以后的调用时 i 均为 1，保证初始化变量只进行一次，第二次调用时 e、u 等全局变量能保持上一次的值。

参考程序：

```
function result =EasyTempPID(Set_Temp,Actual_Temp,i)
Kp=50;% 推荐在50左右
Ki=4; %推荐在10
Kd=0;% 推荐 20以下
global e
global u
global f1
if(i==0) % 初始化
    e=[0 0 0];
    u=[0,0];
end
if((Set_Temp- Actual_Temp)>=1) % 加热最大阈值范围按实际情况设定
    f1 = 128; % 128就80H，也就是PWM0H寄存器的值，实际上只取高8位
end
if(0<(Set_Temp- Actual_Temp)<1) % 执行PID
    e(3) = e(2);
    e(2) = e(1);
    e(1) = Set_Temp- Actual_Temp;
    u(2)=u(1)+Kp*(e(1)-e(2))+Ki*e(1)+Kd*(e(1)-2*e(2)+e(3));
    u(1)=u(2);
    f1 = u(2);
```

```

end
if((Set_Temp- Actual_Temp)<=0)
    f1 = 0;
end
f = f1;%控制量f1
end

```

算法返回值赋给输出参数 **result**，最终传递给上位机然后通过串口对硬件电路板的 PWM 进行控制。这里的 **result** 就是在上面实验原理介绍的 PWM0H/L 值（该值是虽 16 位的，但由于 PWM 实际有效数据段只有 100 有效，因此采用高 8 位的 PWM0H 取值即可 PWM0L 可以忽略。由于控制周期的 PWM0H/L=8000H，因此占空比就等于 PWM0H/L 与 PWM1H/L 比值。当 PWM0H=128 时，输出占空比等于 100%。uk 最后发给单片机的控制量在 0~128 之间，若计算值超过该范围由上位机程序对其进行限定）。

为了节约时间和提高 Kp、Ki、Kd 等 PID 参数的设定，在做混合编程之前应该利用三容水箱计算机控制软件先进行相关参数的调试，然后就可以直接用相关 PID 参数代入到混合编程之中。

4.6 Matlab 混合编程 PID 改进温度控制实验

4.6.1 Matlab 中 dll 文件的制作

打开 Matlab，在 Command Windows 窗口中输入 **deploytool**，弹出如图 8 所示对话框。

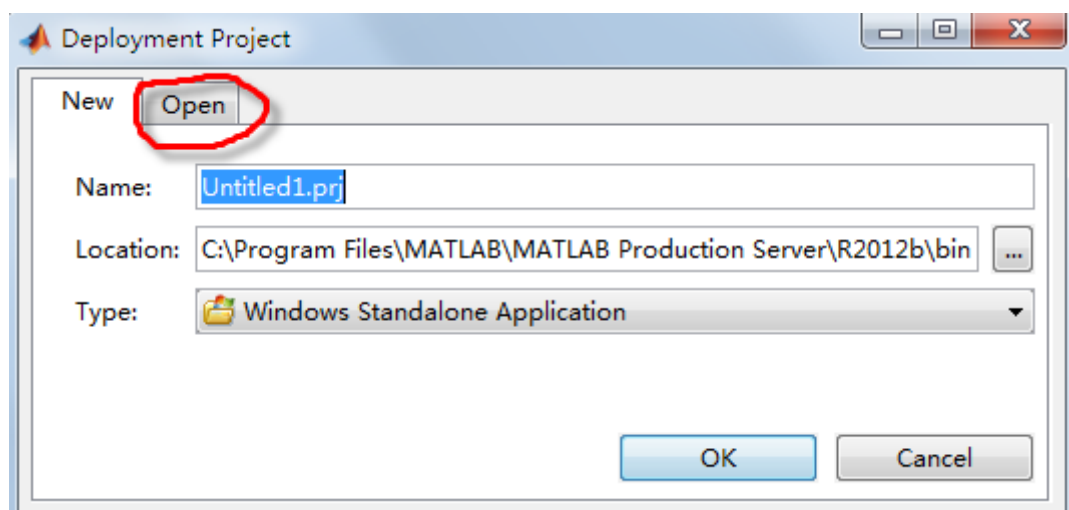


图 8 deploytool 对话框

选择 open 页面，并点击 **Browse** 按钮，如图 9 所示。

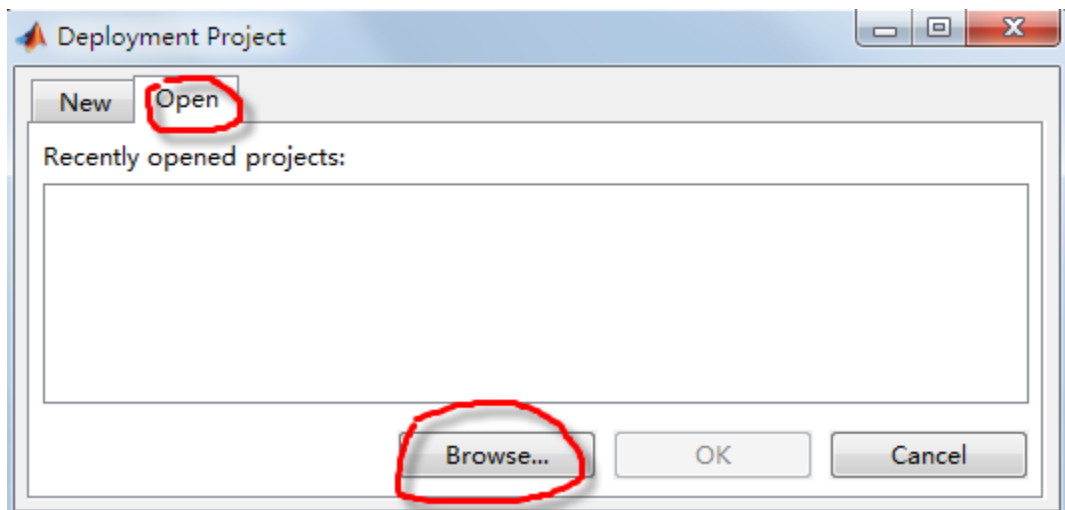


图 9 打开已有工程

在桌面上打开“混合汇编”，选择 PIDCom.prj，如图 10 所示。

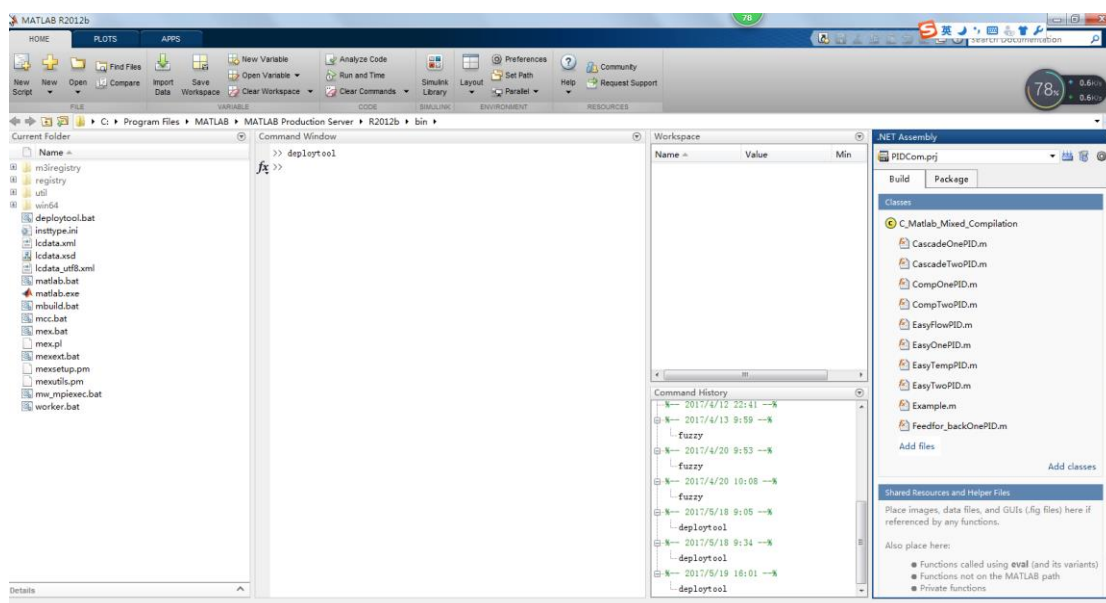


图 10 Matlab deploytool 界面

其中工程名为 PIDcom.prj，包含一个名为 C_Matlab_Mixed_Compilation 的类，类中包含 Example、EasyOnePID 等方法（函数）。其中 Example 为示例编程，其包含了一个完整的 PID 编程算法及详细的文字介绍，其 m 文件 Example.m 为只读文件，请不要随意对其操作。


在做 Matlab 混合编程和 PID 改进算法温度控制时，需要自己在 Matlab 中的 EasyTempPID.m 文件中完成相应函数的编写。在完成“EasyTempPID”算法修改后，如图 11 所示。

```

function [f] = EasyTempPID(Set_Temp, Actual_Temp, i)
Kp=50;% 推荐在50左右
Ki=4; %推荐在10
Kd=0;% 推荐 20以下
global e
global u
global f1
if(i==0) % 初始化
    e=[0 0 0];
    u=[0, 0];
end
if((Set_Temp- Actual_Temp)>=1) % 加热最大阈值范围按实际情况设定
    f1 = 128; % 为何是128? 而不是100? 还需要进一步确定
end
if(0<(Set_Temp- Actual_Temp)<1) % 执行PID
    e(3) = e(2);
    e(2) = e(1);
    e(1) = Set_Temp- Actual_Temp;
    u(2)=u(1)+Kp*(e(1)-e(2))+Ki*e(1)+Kd*(e(1)-2*e(2)+e(3));
    u(1)=u(2);
    f1 = u(2);
end
if((Set_Temp- Actual_Temp)<=0)
    f1 = 0;
end
f = f1;%控制量f1
end

```

图 11 在 EasyTempPID 中完成算法修改

在图 11 中，点击  按钮对算法进行编译，请等待 Matlab 的编译过程，如无错误编译结果见图 12 所示，如有红色字体提示错误，请修改后重新编译。将编译生成“桌面\混合编程\PIDcom\distrib\PIDcom.dll 和 PIDcom.ctf”，将它们复



制到（桌面选择计算机控制实验系统，右击 ，打开文件位置），并替换原来的 PIDcom.dll 和 PIDcom.ctf 即可(如果没有 PIDcom.ctf 文件可以只替换 PIDcom.dll 文件)。

注意：基于 C#的计算机控制系统实验教学 PC 软件需要事先安装在 D:\计算机控制系统实验教学平台\下

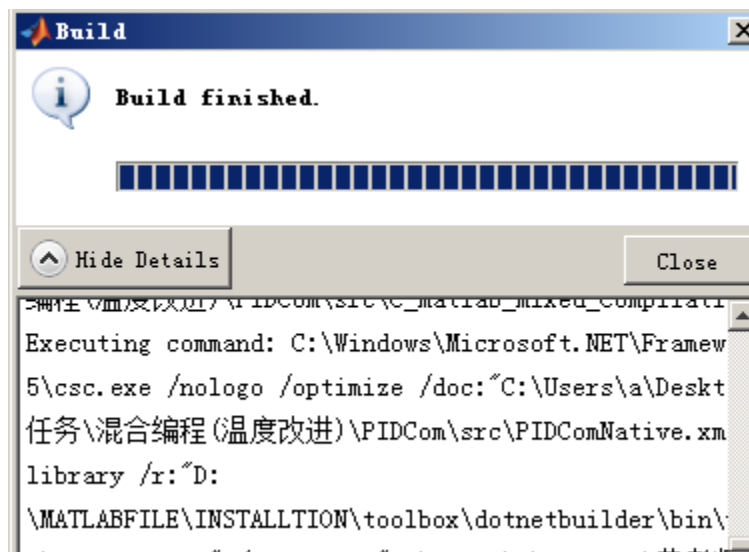


图 12 matlab 中编译结果

4.6.2 C#软件调用 dll 实验过程

1) 重复 4.3 实验步骤中的 2) ~5)。但要注意在 2) 步骤中请在右上角的“是否选择混合汇编”上打勾!!!

2) 若控制效果不佳，则修改 Matlab-PID 控制算法中的 K_p 、 K_i 、 K_d ，和控制方法再重复本实验 1) ~2) 步骤。

4.6.3 实验记录和分析

- 1) 记录每次实验的 K_p 、 K_i 、 K_d 和控制效果，并进行分析；
- 2) 通过混合编程参数整定、改进 PID 算法编写，得到较满意的液位控制曲线；
- 3) 记录自己编写的 .m 文件的程序。

通过上述实验，记录在保证预设液位一定的条件下，通过修改 PID 控制参数实现对温度控制的不同效果，观察实验现象，并记录数据。

表 2 实验数据记录

Kp				
ess				
σ_p				

通过对参实验数据的整理找出静差和超调量在合理范围内的 Kp 的值，最优比例系数可由此确定。

表 3 实验数据记录

Ki				
ess				
σ_p				

通过对参实验数据的整理找出静差为 0 和超调量在合理范围内的 Ki 的值，在此过程中可以通过响应曲线的好坏反复改变比例系数和积分系数，以期得到满意的控制过程与整定参数。

5. 注意事项

1) 实验前将水箱中的水清空。(方法 1: 直接在 FLASH 中点击电动后将其打开; 方法 2: 停止实验, 或重启实验五程序; 方法 3: 关闭并启动实验台总开关电源)

2) 使用前请自行备份“混合编程”文件夹, 不要随意更改文件夹中的内容, 自己编程需要的.m 文件可以自行保留, 完成实验后请从实验前备份的“混合编程”文件夹替换掉用过的“混合编程”文件夹。

3) 用户如果发现除了 Example.m 之外, 其它文件中除了函数入口部分外有其它内容, 请将备份的“混合编程”文件夹替换桌面上的文件夹。

6. 问题与作业

- 1) 分析常规 PID 温度控制的特点。
- 2) 论述 PID 有哪些主要改进方法, 各用于什么场合。
- 3) 你在 PID 改进算法中采用了什么措施, 分析方法及其效果。
- 4) 请根据实验台温度和液位控制系统结构原理, 分析温度控制的 PID 参数中 Kp 的值远小于液位控制时的 Kp 值。
- 5) 请描述你的实验过程和步骤, 并分析结果, 完成实验报告。

实验六 基于组态王的温度 PID 控制实验（自动）

1. 实验目的

- 1) 了解四线制 PT100 温度传感器测量原理；
- 2) 掌握组态王软件的使用方法；
- 3) 了解加热管控制温度的原理；
- 4) 了解组态王软件与基于单片机的采集控制器通讯协议；
- 5) 掌握组态王编程和 PID 控制算法编写。

2. 实验要求

- 1) 用组态王软件完成温度控制系统软件设计；
- 2) 利用组态王编写改进 PID 控制算法，实现基于固态继电器的加热管温度高品质的控制；
- 3) 完成实验数据采集与温度控制，并完成实验报告。

3. 系统结构框图

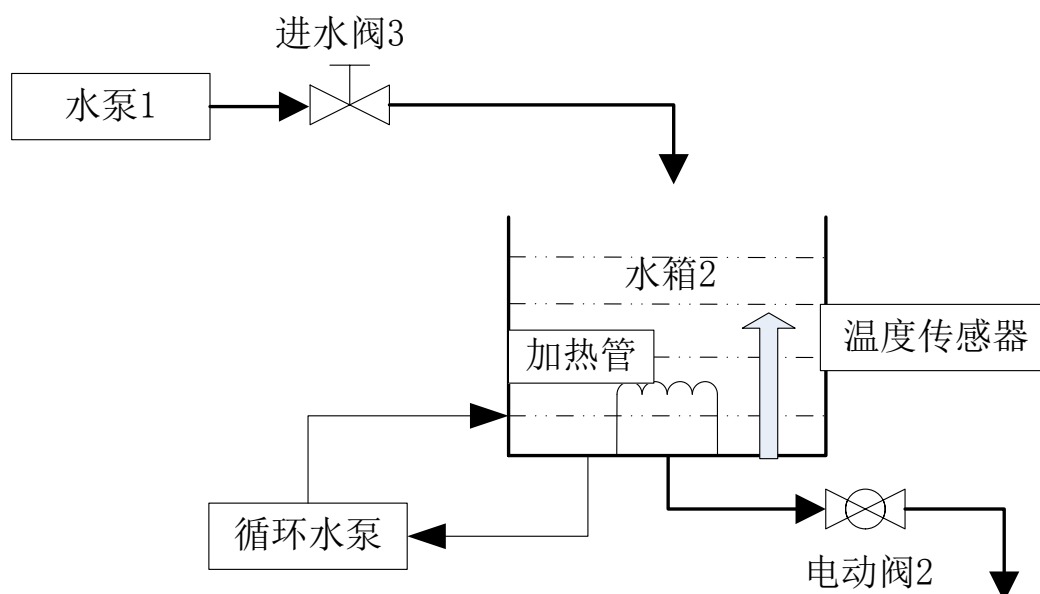


图 1 单水箱一阶温度控制结构图

4. Modbus RTU 通信通用规约

4.1 Modbus 协议简介

Modbus 协议最初由 Modicon 公司开发出来，在 1979 年末该公司成为施耐德自动化(Schneider Automation)部门的一部分，现在 Modbus 已经是工业领域全

球最流行的协议。此协议支持传统的串口链路 RS-232、RS-422、RS-485 和以太网设备。许多工业设备包括 PLC，DCS，智能仪表等都在使用 Modbus 协议作为通讯标准。

Modbus 协议包括 ASCII、RTU、TCP 等，并没有规定物理层。此协议定义了控制器能够认识 and 使用的报文结构，而不管它们是经过何种网络进行通信的。标准的 Modicon 控制器使用 RS232 实现串行的 Modbus。Modbus 的 ASCII、RTU 协议规定了报文、数据的结构、询问和应答的方式，数据通讯采用主/从方式，主站发出数据请求报文，从站接收到正确报文后就可以发送数据到主站以响应请求；主站也可以直接发报文修改从站的数据，实现双向读写。

4.2 Modbus 通用规约介绍

Modbus 协议需要对数据进行校验，串行协议中除有奇偶校验外，ASCII 模式采用 LRC 校验，RTU 模式采用 16 位 CRC 校验，但 TCP 模式没有额外规定校验，因为 TCP 协议是一个面向连接的可靠协议。另外，Modbus 采用主从问答方式收发数据，在实际使用中如果某从站站点断开后（如故障或失电），主站可以诊断出来，而当故障修复后，网络又可自动接通。因此 Modbus 协议的可靠性较好。

Modbus 协议在串行链路中 RTU 模式使用的最多，通用性很强，所以在这里仅介绍一下 Modbus RTU 协议即 Modbus 通用规约。

下表是 Modbus Rtu 支持的部分常用功能码：

功能码	名称	作用
01	读取线圈状态	取得一组逻辑线圈的当前状态（ON/OFF）
02	读取输入状态	取得一组开关输入的当前状态（ON/OFF）
03	读取保持寄存器	在一个或多个保持寄存器中取得当前的二进制值
04	读取输入寄存器	在一个或多个输入寄存器中取得当前的二进制值
05	强置单线圈	强置一个逻辑线圈的通断状态
06	预置单寄存器	把具体二进值装入一个保持寄存器

它们即可实现对下位机的数字量和模拟量的读写操作。需要指出的是，本实验平台当前通讯主要使用 03 和 06 功能码的通讯

4.3 Modbus 通用规约报文说明

1) 读取模拟量寄存器（保持寄存器）：03 功能码

上位机发送报文：

[设备地址][功能码 03][起始寄存器地址高 8 位][低 8 位][读取的寄存器数高 8 位][低 8 位][CRC 校验的低 8 位][高 8 位]

定义	设备地址	功能码	寄存器起始地址	读取数据长度	CRC 校验
数据	11H	03H	00 6B	00 03	CRC 16
字节数	1	1	2	2	2

报文示例：11 03 00 6B 00 03 76 87

报文解析：

[11]设备地址和上面的相同。

[03]功能码：读取模拟量的功能码为 03。

[00 6B]起始寄存器地址高 8 位、低 8 位：表示想读取的模拟量的起始地址，比如示例中的寄存器起始地址为 107。

[00 03]寄存器数高 8 位、低 8 位：表示从起始地址开始读多少个模拟量。示例中为读取 3 个模拟量。注意：在返回的信息中一个模拟量需要返回两个字节。

[76 87] CRC 校验同上。

下位机应答报文：

[设备地址][功能码 03][返回的字节个数][数据 1][数据 2]...[数据 n][CRC 校验的低 8 位][高 8 位]

定义	设备地址	功能码	数据字节数	返回数据	CRC 校验
数据	11H	03H	06H	DATA	CRC 16
字节数	1	1	1	6	2

报文示例：11 03 06 02 2B 00 00 00 64 C8 BA

报文解析：

[11]设备地址：和上面相同。

[03]功能码：和上面相同

[06]返回的字节个数：表示数据的字节个数，也就是数据 1，2...n 中的 n 的值。例子中返回了 3 个模拟量的数据，因为一个模拟量需要 2 个字节所以共 6 个字节。

[02 2B 00 00 00 64]数据 1...6：其中[数据 1][数据 2]分别是第 1 个模拟量的高 8 位和低 8 位，[数据 3][数据 4]是第 2 个模拟量的高 8 位和低 8 位，以此类推。示例中返回的值分别是 555，0，100。注意：在实际使用中，不同厂家规约可能会做改动，比如说对模拟量的处理是低字节在前，高字节在后，数据格式可分为有符号整型和无符号整型，有符号整型的又分为补码和绝对值两种方式等等，这是在实际分析报文需要注意的地方。

[C8 BA]CRC 校验同上。

需要注意的是，若 CRC 校核没通过，或地址不对，则返回的功能码应换为差错码 0x83

2) 写单个模拟量寄存器（保持寄存器）：06 功能码

上位机发送报文：

[设备地址][命令号 06][需下置的寄存器地址高 8 位][低 8 位][下置的数据高 8 位][低 8 位] [CRC 校验的低 8 位][高 8 位]

定义	设备地址	功能码	置位寄存器地址	下置数据	CRC 校验
数据	11H	06H	00 01	00 03	CRC 16
字节数	1	1	2	2	2

报文示例：11 06 00 01 00 03 9A 9B

报文解析：

[11]设备地址和上面的相同。

[06]功能码:写模拟量的功能码号为 06。

[00 01]需下置的寄存器地址高 8 位，低 8 位：表明了需要下置的模拟量寄存器的地址。比如示例中就是对地址为 1 的寄存器进行置数。

[00 03]下置的数据高 8 位，低 8 位：表明需要下置的模拟量数据。比如示例中就把 1 号寄存器的值设为 3。

[9A 9B]CRC 校验同上。

注意：此命令一条只能下置一个模拟量的状态。

下位机应答报文：

报文示例：11 06 00 01 00 03 9A 9B

报文解析：同上。

如果下位机执行成功，则把上位机发送的报文原样返回，否则不响应。

需要注意的是，若 **CRC** 校核没通过，或地址不对，则返回的功能码应换为差错码 **0x86**

3) 写数字量（线圈状态）：05 功能码

上位机发送报文：

[设备地址][功能码 05][需下置的寄存器地址高 8 位][低 8 位][下置的数据高 8 位][低 8 位][CRC 校验的低 8 位][高 8 位]

定义	设备地址	功能码	置位寄存器地址	下置数据	CRC 校验
数据	11H	05H	00 AC	FF 00	CRC 16
字节数	1	1	2	2	2

报文示例：11 05 00 AC FF 00 4E 8B

报文解析：

[11]设备地址和上面的相同。

[05]功能码:写数字量的功能码号为 05。

[00 AC]需下置的寄存器地址高 8 位，低 8 位：表明了需要置位的开关的地址。

[FF 00]下置的数据高 8 位，低 8 位：表明需要下置的开关量的状态。示例中为把该开关闭合。注意，此处置位命令只可以是 [FF 00] 表示闭合，[00 00] 表示断开，其他数值非法。

[4E 8B]CRC 校验同上。

注意：此命令一条只能下置一个开关量的状态。

下位机应答报文：

报文示例：11 05 00 AC FF 00 4E 8B

报文解析：同上。

如果下位机执行成功，则把上位机发送的报文原样返回，否则不响应。

5. 组态王中的设置

5.1 组态王中的温度采集参数设置



图2 组态王中的温度采集参数设置

(0-100对应0-10000相当于保留小数点后面两位数字)

5.2 组态王中的 PWM 控制参数设置

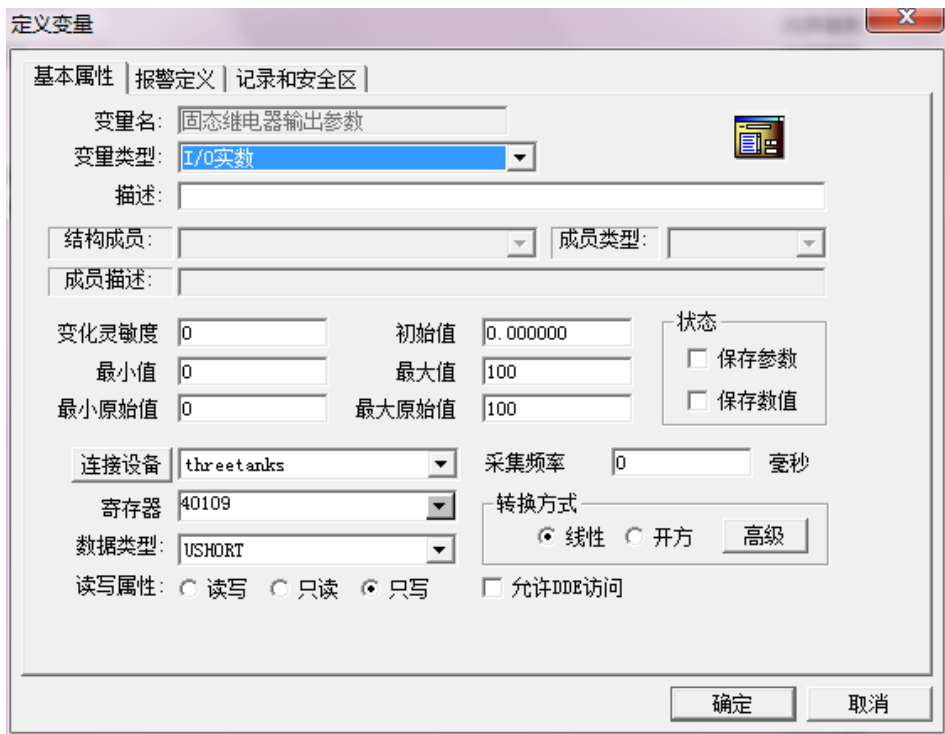


图3 加热器固态继电器 PWM 设置

（其中0-100的原始值为需要加热管的加热功率）

5.3 组态王与三容水箱数据采集控制器的通讯

组态王与单片机采用莫迪康 Modbus_RTU 通信，具体定义如图 4 所示：

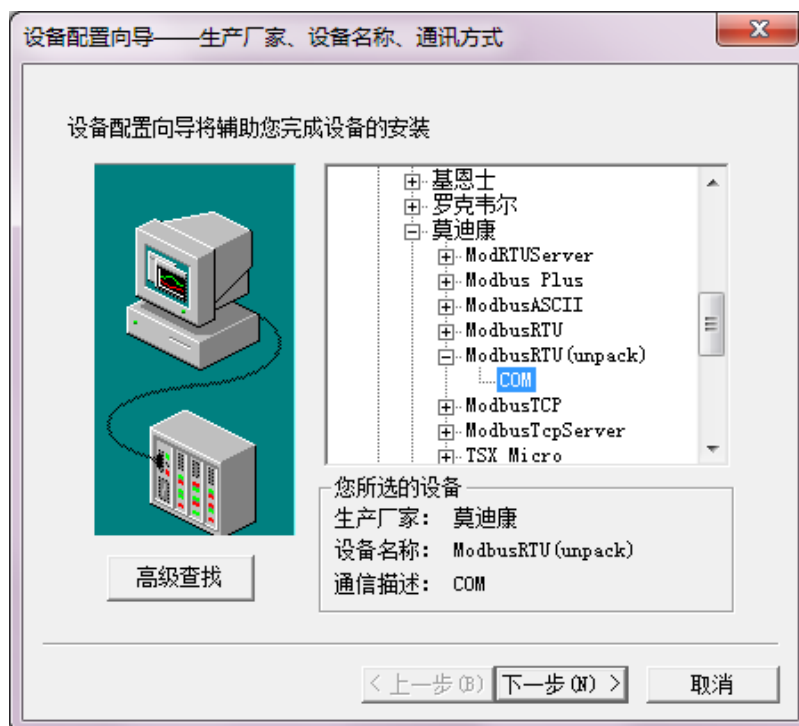


图4 通信协议选取

5.3.1 通讯口设置

通讯方式：RS-232，RS-485，RS-422 均可。

波特率：由单片机决定（2400，4800，9600，19200bps）。

字节数据格式：由单片机决定。

起始位	数据位	校验位
-----	-----	-----

注意：在组态王中设置的通讯参数如波特率，数据位，停止位，奇偶校验必须与单片机编程中的通讯参数一致

5.3.2 在组态王中定义设备地址的格式

格式：Address[:Delay: BeforeDelay]

前面的两个字符是设备地址，范围为 1—255，此地址为单片机的地址，由单片机中的程序决定；

Delay 0-60000 发送每条命令帧后的延迟时间（毫秒）

BeforeDelay 0-60000 发送每条命令帧前的延迟时间（毫秒）

注：设备地址已在串口设备建立时设置(地址为 01, 不设帧前帧后延迟时间)，如图 5 所示。

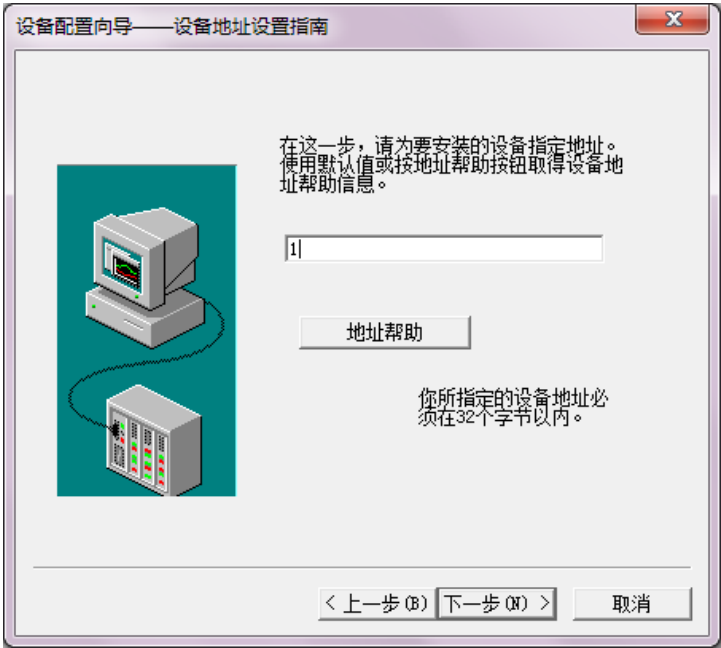


图 5 串口设备 threetanks 的地址设计界面

5.3.3 在组态王中定义的寄存器及功能码

组态王与单片机通讯采用 Modbus_Rtu 通信，详细请参考“计算机控制系统实验平台 Modbus_RTU 通信通用规约说明”

5.3.4 本实验主要参数的寄存器地址

表 1 组态王中各参数寄存器地址规定

参数名称	组态王中的寄存器地址	单片机中的寄存器地址
水泵 1 电压	40101	100
电动阀 2 开度	40105	104
循环泵开关	40103	102
加热控制	40109	108
液位 2	40002	001
温度	40005	004
进水阀 3	0013	012

注意：

计算机控制实验平台的各参数地址详见附录 1：组态王和单片机中各参数寄

寄存器地址定义。

6. 实验步骤

6.1 实验设备

三容水箱计算机控制实验台、万用表、计算机、RS232/RS485 串口线，电源线、组态王软件。

6.2 设备连接与检查

- 1) 将计算机控制系统实验台的水槽中灌满水（至溢出口的下沿）。
- 2) 连接 RS232/RS485 串口线，将串口选择开关拨向使用的串口类型的方向。
- 3) 连接电源线，将电源插头接到 220V 电源上。
- 4) 打开电源，电源指示灯亮。
- 5) 使用万用表检查 5V，24V，220V 电源是否正常。

6.3 操作步骤

1) 温度控制实验准备工作

- (1) 初始化球阀，将电动球阀 2 关闭（用串口调试工具向其发相应的命令）
- (2) 打开进水阀 3（用串口调试工具向其发相应的命令）
- (3) 打开水泵 1，对水箱 2 进行打水，水上升至 180mm 左右，停止打水（用串口调试工具向其发相应的命令）

(4) 打开循环泵（用串口调试工具向其发相应的命令）

(5) 完成实验准备工作

2) 设计组态软件

(1) 在 PC 机上打开组态王程序，新建一个工程。

(2) 新建 I/O 设备。

(3) 创建基于 PID 的温度控制画面。

(4) 创建数据库，包括

①I/O 实数变量：加热控制参数、温度采集参数

②内存实数变量：设定温度 T、TKp、TKi、TKd、Tek、Tek1、Tek2、Tuk、Tuk1.....等

③内存离散变量：开始/结束标记位等

(5) 定义动画连接和设置

(6) 编写 PID 控制程序和其它界面相关参数设置

3) 运行组态王程序，并通过调节 PID 控制参数进行温度控制。

注意：正常情况下，采集控制器程序已安装，若未下载，则需要重新下载程序：

短接下载口 Download，下载计算机控制实验 Modbus-RTU 协议单片机程序 V4（自动）。

7. 实验记录和分析

1) 记录每次实验的 TKp、TKi、TKd 和控制效果，并进行分析；

2) 通过修改 TKp、TKi、TKd，得到较满意的温度控制曲线。

8. 注意事项

1) 实验前将水箱中的水清空；

2) 保存好设计工程；

3) 水箱液位为 200mm 较为合适；

4) 温度设定范围在水温 20 以上 40 以内较为合适；

5) 改进 PID 控制参考实验五中 Matlab 程序；

6) 以下为标准 PID 控制算法参考程序。

```
//*****温度控制*****//
if(\\本站点\加热标志==1)
{
    float deflaut;
    float A;
    float B;
    float C;

    A=TKp+TKi+TKd;
    B=-(TKp+2*TKd);
    C=TKd;
    \\本站点\Tek=\\本站点\设定温度 T-\\本站点\温度采集;
    deflaut=A*Tek+B*Tek1+C*Tek2;
    \\本站点\Tuk=\\本站点\Tuk1+deflaut;

    if(\\本站点\Tuk>100)
    {
        \\本站点\Tuk=100;
    }
    else if(\\本站点\Tuk<0)
    {
        \\本站点\Tuk=0;
    }
}
```



```

    }

    \\本站点\Tuk1=\\本站点\Tuk;
    \\本站点\Tek2=\\本站点\Tek1;
    \\本站点\Tek1=\\本站点\Tek;

    \\本站点\固态继电器输出参数=\\本站点\Tuk;
}

//*****实验结束*****//
if(\\本站点\加热标志==0 )
{
    \\本站点\固态继电器输出参数=0;
}

```

附录 1：组态王和单片机中各参数寄存器地址定义

序号	参数名称	组态王中的 寄存器地址	对应单片机中的 寄存器地址
01	水泵 1 电压	40101	100
02	水泵 2 电压	40102	101
03	循环泵开关	40103	102
04	电动阀 1 开度	40104	103
05	电动阀 2 开度	40105	104
06	电动阀 3 开度	40106	105
07	加热控制参数	40107	106
08	液位 1	40001	000
09	液位 2	40002	001
10	液位 3	40003	002
11	管路压力	40004	003
12	温度	40005	004
13	流量 1	40006	005
14	流量 2	40007	006