

综合与创新性实验

本章以 DSP F2812 教学实验开发系统的典型控制系统为例,主要包括温度控制系统、交通灯控制系统、直流无刷电动机控制系统和大功率交流异步电动机控制系统,介绍了自行设计的温度信号采集和控制扩展板的工作原理及使用方法、ICETEK-MOTOR-E 控制板和 ICETEK-F2812-ACM 控制系统的工作原理及使用方法等。在实验教学开发系统 CCS 仿真环境下,实现闭环控制系统的多种控制算法,以 PID 控制算法在温度控制系统中的应用为例,包括有限冲击响应滤波器(FIR)算法、无限冲击响应激光器(IIR)算法、快速傅里叶变换(FFT)算法、卷积算法、自适应滤波器算法、抽样定理等。对 DSP 教学实验开发系统的学习和实际工程设计都有一定的参考价值。

5.1 温度控制系统

1. 实验目的

- (1) 掌握利用 ICETEK-F2812-A 评估板与 ICETEK-CTR 板上带的温度信号采集和控制扩展板的工作原理。
- (2) 熟悉 F2812 DSP 的通用 I/O 端口和定时器的编程使用。
- (3) 学习利用数字 PID 控制算法的控制温度系统。

2. 实验设备

计算机,ICETEK-F2812-EDU 实验箱,如图 5.1 所示,扩展板如图 5.2 所示,温度传感器如图 5.3 所示,加热器如图 5.4 所示等。

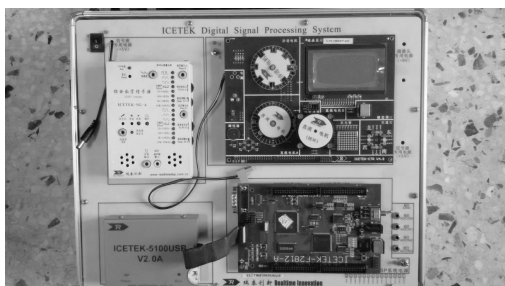


图 5.1 ICETEK-F2812-EDU 实验箱

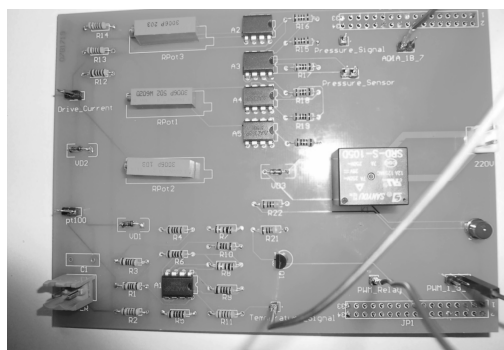


图 5.2 扩展板



图 5.3 温度传感器

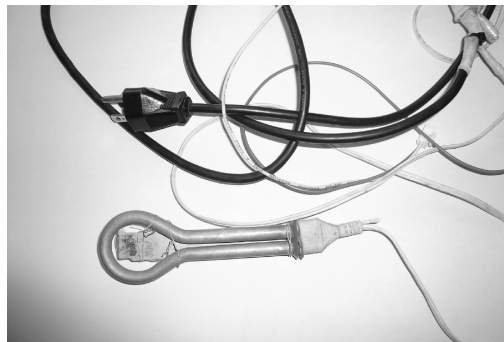


图 5.4 加热器

3. 实验内容

通过 ICETEK-F2812-A 评估板与 ICETEK-CTR 板上带的温度信号采集和控制扩展板,实现控制系统的温度控制并显示在液晶屏上。

4. 实验原理

温度检测与加热驱动板(扩展板)对水的温度进行测量,利用 A/D 转换和滤波功能计算出实时温度值,显示在液晶屏上,运用 PID 控制算法运算,利用温度检测与加热驱动板的功率输出功能,调节水的温度达到设定温度 50°C 值并保持。

温度闭环控制系统:在 DSP 系统实验箱(如图 5.1 所示)上插入扩展板(如图 5.2 所示),通过温度传感器(如图 5.3 所示)放入水箱中,对水温进行信号采集,经过 DSP 芯片中的 A/D 转换器,将温度的模拟值转换成数字量,与给定值进行比较,产生偏差输入到

PID 控制器中,通过计算形成控制系统的输出,输出到水箱里的加热器(如图 5.4 所示)上,对水箱里的水进行加热。DSP 可实时读取当前温度值,利用 DSP 中运行的控制程序,根据温度读数控制 PWM 的脉宽,调节占空比,从而实现温度的闭环控制。

液晶屏上可以显示:状态、温度值十位、温度值个位、温度值小数点 1 位;其中状态有 7 种,显示 0 表示实时温度,显示 1 表示设定温度(调节温度要达到的温度值),显示 2 表示温度上限值 60°C ,显示 3 表示温度下限值 40°C ,显示 4 表示 PID 参数的 K_p 值,显示 5 表示 PID 参数的 K_i 值,显示 6 表示 PID 参数的 K_d 值,在非实时温度状态下,通过键盘可以修改温度设定值上、下限。液晶屏还能够实时显示温度曲线的变化情况。

采用串口通信方式可以修改温度设定值和上、下限,并接收下位机传送上来的温度设定值、上下限值和 PID 参数值。

1) 硬件设计

其系统组成如图 5.5 所示。

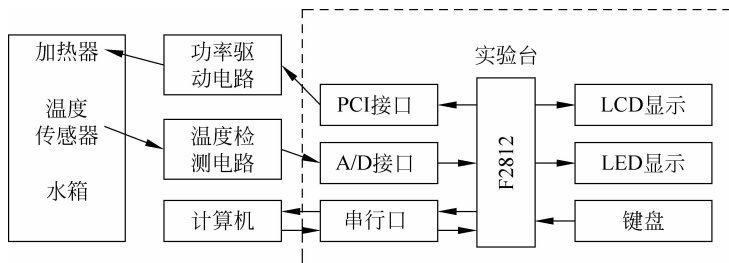


图 5.5 温度控制系统组成

温度检测电路如图 5.6 所示。温度传感器采用铂热电阻 Pt100, Pt100 在常温 $0\sim 100^{\circ}\text{C}$ 之间变化时线性度非常好。Pt100 和 R6、R7、R8 组成桥式电路,为了保证电桥输出电压信号的稳定性,电桥的输入电压通过 TL431 稳至 2.5 V。从电桥获取的差分信号通过两级运放放大后输入 DSP 实验台。由于 DSP 芯片带有 6 通道 12 位宽度的 ADC 转换器,因此温度检测电路的输出电压 AV 连接到实验台提供的 ADC 接口,即 AV 信号连接 ADC 接口 AIN3 号,温度检测电路的 GND 连接 ADC 接口的 GND,如图 5.7 所示。值得注意的是 DSP 实验台的 ADC 电路参考电压为 2.5 V,所以检测电路的输出电压 AV 不应超过 2.5 V。图 5.6 中所示的 VCC 和 GND 分别与 DSP 实验台 PCI 接口卡的 A52 和 A59 相接。

功率驱动电路如图 5.8 所示。温度控制环节采用光电耦合器驱动双向可控硅方式。光电耦合器采用 MOTOROLA 公司生产的用于触发可控硅的 MOC3083 元件,此元件具有过零检测功能,可用直流低电压、小电流来控制高电压、大电流,触发电路简单可靠,且抗干扰能力强。双向可控硅采用 BTA06。由控制信号 CtrlB 来控制加热电路,当它为低

电平时, MOC3083 输出端不导通; 当控制信号为高电平时, MOC3083 输出端导通, 双向可控硅的触发端收到触发信号并导通, 将负载电源接通, 这样加热器件开始对水箱进行加热。控制信号 CtrlB 接到 DSP 实验台 PCI 接口卡上的 I/O 口 A26, 如图 5.9 所示。

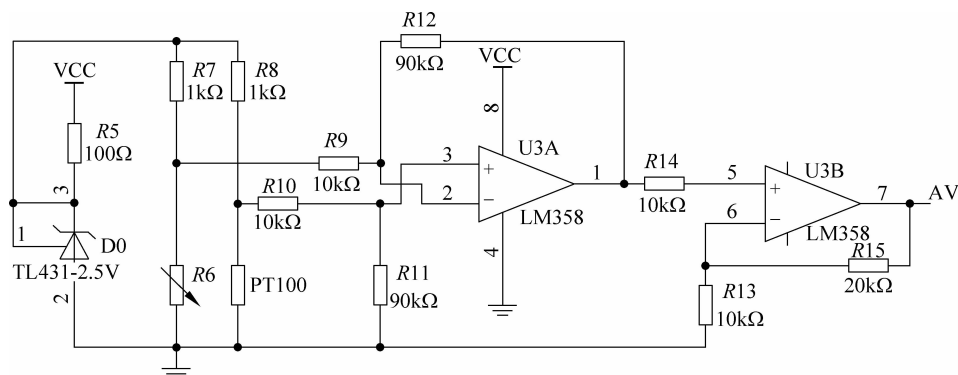


图 5.6 温度检测电路

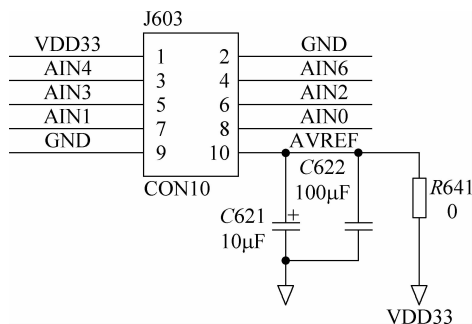


图 5.7 实验台 ADC 接口

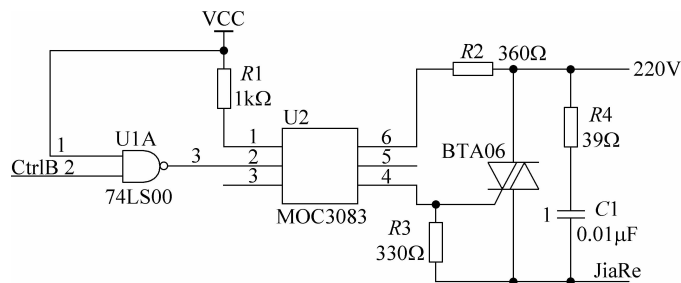


图 5.8 功率驱动电路

2) 软件设计

系统程序采用 C 语言编写非嵌入式操作系统程序。为便于程序的使用与维护,全部程序采用模块化结构,由一个主程序和若干子程序组成。主程序首先进行初始化,包括 I/O 口、中断系统、串口、IIC、ADC、LCD 等,然后对 4LED 和 LCD 进行初始化显示,接着进入检测与控制循环程序中。检测与控制程序包括 6 部分程序,有 4LED 与 LCD 显示控制、键盘检测与控制、A/D 采集与滤波处理、控制算法计算、功率控制和串口发送。

(1) 4LED 与 LCD 显示控制

4LED 指 4 个数码管,分别显示:状态、温度值十位、温度值个位、温度值小数点 1 位;其中状态有 7 种,0 表示实时温度,1 表示设定温度(调节温度要达到的温度值),2 表示温度上限值,3 表示温度下限值,4、5 和 6 分别表示 PID 参数的 K_p 、 K_i 和 K_d 初始值。在非实时温度控制状态下,通过键盘可以修改对应状态的参数值。通过液晶屏主要显示实时的

J903									
VDD33	A1	B1	VDD33	NET RXDV	A31	B31	EX_CS1		
VDD33	A2	B2	AB20	EX_CS2	A32	B32	BnWE		
NRESET	A3	B3	AB16	BnOE	A33	B33	BnWBE1		
AB1	A4	B4	AB13	NC5	A34	B34	NC6		
AB9	A5	B5	AB22	BnWEB0	A35	B35	NET RXD3		
AB11	A6	B6	AB14	NET TXD0	A36	B36	NET RXER		
AB8	A7	B7	AB21	NC7	A37	B37			
AB10	A8	B8	AB15	NET RXD1	A38	B38	NET MDIO		
AB6	A9	B9	AB18	NET RXD2	A39	B39	SPIMOSI0		
AB7	A10	B10	AB19	NET RXCLK	A40	B40	NET TXEN		
VDD33	A11	B11	VDD33	NET TXCLK	A41	B41	SPIMISO0		
AB5	A12	B12	AB17	NET COL	A42	B42	NET TXD1		
DB6	A13	B13	AB4	NET MDC	A43	B43	SPICLK0		
AB3	A14	B14	DB7	NET TXD2	A44	B44	SPIMISO1		
DB4	A15	B15	AB2	NET TXD3	A45	B45	BUT INT		
AB12	A16	B16	DB5	SPICLK1	A46	B46	NSS0		
DB2	A17	B17	AB0		A47	B47	II CSCL		
DB3	A18	B18	GND	SPIMOSI1	A48	B48	II CSDA		
DB1	A19	B19	GND	VDD33	A49	B49	VDD33		
GND	A20	B20	DB0	VDD33	A50	B50	VDD50		
DB13	A21	B21	DB15	VDD50	A51	B51	RXD1		
DB12	A22	B22	DB14	VDD50	A52	B52	RTS1		
DB9	A23	B23	DB11		A53	B53			
DB8	A24	B24	DB10		A54	B54	TXD1		
EX_IN1	A25	B25	EX_IN0		A55	B55			
EX_OUT2	A26	B26	EX_OUT1	GND	A56	B56			
EX_OUT0	A27	B27	EX_IN2	GND	A57	B57			
NC3	A28	B28	NC4	GND	A58	B58	CTS1		
NC2	A29	B29	NC1	GND	A59	B59			
NET RXD0	A30	B30	NET CRS	BUT INT1	A60	B60			
				PCI					

图 5.9 PCI 卡接口

温度曲线,曲线部分每隔 250 s 刷新一次,如图 5.10 所示。

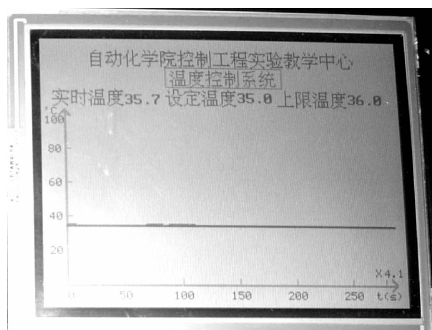


图 5.10 液晶屏显示实时的温度曲线

(2) 键盘检测与控制

利用实验台上的 4 个功能键, Fun 键为状态控制键, F 键为移位键, “+”键为参数增加键, “-”键为参数减少键。采用查询法检测键盘键值, 在某一非实时温度控制状态下, 完成各种参数的增减功能。

(3) A/D 采集与滤波处理

经温度检测电路得到的电压信号通过 A/D 转换为数字信号, 为了能去除掺杂在信号中的噪音和干扰, 采用去极值平均滤波法来处理数据, 就是将连续测量的 N 个采样值按照大小顺序排列, 然后去掉最大值和最小值, 最后将剩下的 $N-2$ 个采样值求平均值。

(4) PID 算法

PID(Proportional Internal Derivative)控制是控制工程中应用比较广泛的一种控制策略, 经过长期的工程实践, 已形成了一套完整的控制方法和典型的结构。它不仅适用于数学模型已知的控制系统中, 而且对于大多数数学模型难以确定的工业过程也可应用, 在众多工业过程控制中取得了满意的应用效果。PID 控制系统的基本原理如图 5.11 所示。

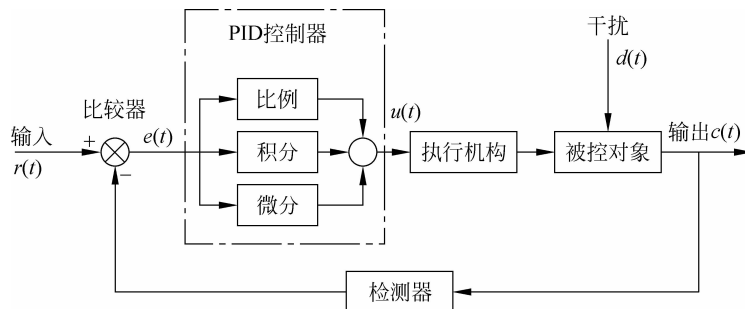


图 5.11 系统原理图

理想的模拟 PID 控制器根据给定值 $r(t)$ 与实际输出值 $c(t)$ 构成的控制偏差:

$$e(t) = r(t) - c(t)$$

将偏差的比例、积分和微分通过线性结合构成控制量,对被控对象进行控制,其控制规律为

$$u(t) = K_p[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt}] \quad (5-1)$$

式中, K_p 是比例系数, T_i 是积分时间常数, T_d 是微分时间常数。

模拟 PID 控制器的输入输出量均为模拟量,如电压、电流。而当使用计算机实现控制算法时,其输入输出必须为数字离散量。所有首先必须将上述的 PID 控制规律的连续形式变成离散形式,然后才能编程实现。PID 控制器算法的离散形式为

$$u(k) = K_p e(k) + K_i \sum_{j=0}^k e(j) + K_d [e(k) - e(k-1)] \quad (5-2)$$

其中, T 是采样周期; k 是采样序号 $k=0,1,2,\dots$; $u(k)$ 是第 k 次采样时刻计算机的输出值; $e(k)$ 是第 k 次采样时刻输入的偏差值; K_p 是比例系数; $K_i = \frac{K_p T}{T_i}$ 是积分系数; $K_d = \frac{K_p T_d}{T}$ 是微分系数。

由式(5-2)可以看出,要想计算 $u(k)$,不仅需要本次与上次的偏差信号 $e(k)$ 和 $e(k-1)$,而且还要把历次的偏差值进行相加,不仅计算烦琐,而且为保存 $e(j)$ 还要占用很多内存。因此用式(5-2)直接进行控制很不方便。为此,作如下改动。

根据递推原理,可以得到增量式 PID 控制算法:

$$\Delta u(k) = A e(k) + B e(k-1) + C e(k-2) \quad (5-3)$$

式中, $A = K_p \left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right)$, $B = K_p \left(1 + \frac{2T_d}{T}\right)$, $C = K_p \frac{T_d}{T}$ 分别为第 n 次、 $n-1$ 次和 $n-2$ 次的偏差值, K_p 、 T_i 、 T_d 分别为比例系数、积分系数和微分系数, T 为采样周期。

在整个控制过程中,采样周期 T 基本不变。只要确定比例系数 K_p 、积分时间常数 T_i 、微分时间常数 T_d , PID 中的所有参数就都确定了。有了确定不变的系数后,再加上前 3 次的采样偏差值 $e(k)$, $e(k-1)$, $e(k-2)$ 就可以算出计算机下次要输出的控制增量:

$$\begin{aligned} \Delta u(k) &= u(k) - u(k-1) \\ &= K_p [e(k) - e(k-1)] + K_i e(k) + K_d [e(k) - 2e(k-1) + e(k-2)] \end{aligned} \quad (5-4)$$

PID 参数的选择是实验成败的关键,它决定了温度控制的准确度。数字 PID 调节器参数的整定可以仿照模拟 PID 调节器参数整定的各种方法,根据工艺对控制性能的要求,决定调节器的参数。各个参数对系统性能的影响如下:

比例系数 P 对系统性能的影响:比例系数加大,使系统的动作灵敏,速度加快,稳态误差减小; P 偏大,振荡次数加多,调节时间加长; P 太大时,系统会趋于不稳定; P 太

小,又会使系统动作缓慢。 P 可以选负数,这主要是由执行机构、传感器以及控制对象的特性决定的。如果 P 的符号选择不当对象测量值就会离控制目标的设定值越来越远,如果出现这样的情况 P 的符号就一定要取反。

积分控制 I 对系统性能的影响:积分作用使系统的稳定性下降, I 小(积分作用强)会使系统不稳定,但能消除稳态误差,提高系统的控制精度。

微分控制 D 对系统性能的影响:微分作用可以改善动态特性, D 偏大时,超调量较大,调节时间较短; D 偏小时,超调量也较大,调节时间也较长;只有 D 取合适的值,才能使超调量较小,减短调节时间。

本系统的温度控制器电热元件是加热器。加热器通电加热时,内部温度都很高。当容器内温度升高至设定温度时,温度控制器会发出信号停止加热。但这时加热器的温度会高于设定温度,由于惯性加热器还将会继续进行加热,即使温度控制器发出信号停止加热,被加热器件的温度还会继续上升,然后才开始停止。当停止加热以后,被加热器件的温度会开始下降,下降到设定温度的下限时,温度控制器又开始发出加热的信号,开始加热,但加热器要把温度传递到被加热器件上需要一定的时间,这就要视加热器与被加热器件之间的介质情况而定。通常开始重新加热时,温度会继续下降。所以,传统的定点开关控制温度会有正负误差较大的现象,但这不是温度控制器本身的问题,而是整个加热系统的结构性问题,温度控制器在控制温度时产生的惯性温度误差。另外水温温度是变化比较缓慢且带有纯滞后环节的控制对象。因此针对上述问题,本实验采取多段式 PID 控制算法。

将温度偏差设定 3 个阈值,根据偏差 e 的大小对 PID 控制算法设置不同的 K_p 、 K_i 和 K_d 系数,其规则为:当 $|e(k)|$ 较大时,为避免产生过大的超调,又使系统有较快的响应;取 K_i 为零。当 $|e(k)|$ 处于中等大小时,为使系统具有较小的超调, K_p 应取小一些, K_i 、 K_d 要大小适中,以保证系统的响应速度。当 $|e(k)|$ 较小,为使系统有良好的稳态性能,应增加 K_p 、 K_d 取值。同时为避免系统在设定值附近出现振荡,并考虑系统的抗干扰性能,应适当地选取 K_d 的值。

(5) 功率控制

经多段式 PID 控制算法计算后,得到控制输出值 out ,当 $out=0$ 时,系统停止加热;当 $0<out<255$ 时,系统根据其值大小,决定加热时间的长短。由于控制信号接到 ARM 实验台 PCI 接口卡上的 I/O 口上,是属于外围器件扩展输出寄存器,因此在程序中需要定义宏,才能正确使用,方法如下:

```
#define rPCIctrl( * (volatile unsigned * )0x04000000) //定义扩展输出寄存器地址
rPCIctrl = 0x00000000; delay(50); //不加热
rPCIctrl = 0xffffffff; delay(50); //加热
```

(6) 串口向上位机发送数据

采用 RS-232 通信方式,通过实验台上的 UART0 向计算机发送处理后的实时温度值,利用串口调试助手收集数据,以便导入到 Excel 中绘制温度变化曲线并进行分析。

5. 实验操作步骤

(1) 将温度检测与加热驱动扩展板(如图 5.2 所示)插到实验台(如图 5.1 所示)上,将扩展板的电源线、输入输出信号线分别连接到传感器(如图 5.3 所示)和加热器(如图 5.4 所示)上;将水箱内加入水后,放入传感器和加热器。

(2) 串口接收设置。在 PC 上运行 Windows 自带的超级终端串口通信程序(波特率为 9600、1 位停止位、无校验位、无硬件流控制);或者使用其他串口通信程序。

(3) 调试实验例程。打开工程,工程目录为 D:\ICETEK-F2812-A-EDUlab\ DSP281x_examples\Lab0508-WD。

(4) 编译并下载程序。打开实验例程 water_control 目录下的例程,编译链接工程→Remote Connect 进行连接软件仿真器→Download 下载程序→Debug 工具条 Go 命令(F5 快捷键)→观察显示结果→Disconnect 退出调试状态。这样即完成了程序的编译、下载、调试和退出。

(5) 在调试过程中,可以通过键盘修改温度设定值和 PID 参数等值,调节水温控制系统的性能。

6. 实验结果

随着程序的运行,温度指示会不断升高,由于开始温度较低,加热控制会以最大值输出,使温度快速升高,当接近恒温(给定)值,加热控制输出值会越来越小并输出等于 0,即达到了温度要求。水温控制系统是一个大惯性滞后系统,在停止加热后,水温会继续升高出现超调,然后慢慢降低并恒定在给定温度。

7. 思考题

随着控制技术的发展,新的控制方法不断涌现,并在实际控制中取得了较好的控制效果,例如专家系统、模糊逻辑、神经网络、灰色系统理论等,它们和传统的 PID 控制策略相结合又派生出各种新型的 PID 类控制器。采用新的控制算法实现对温度的控制。

5.2 交通灯综合控制

1. 实验目的

(1) 熟悉使用 ICETEK-F2812-A 评估板控制 ICETEK-CTR 上交通灯的方法。

- (2) 掌握 TMS320F2812DSP 定时器的使用和编程。
- (3) 掌握 TMS320F2812DSP 外中断的使用和编程。
- (4) 学习复杂控制程序的设计思路。

2. 实验设备

计算机和 ICETEK-F2812-EDU 实验箱。

3. 实验内容

通过 ICETEK-F2812-A 评估板控制 ICETEK-CTR 上的交通灯,结合定时器和中断实现交通灯控制,程序流程图如图 5.12 所示。

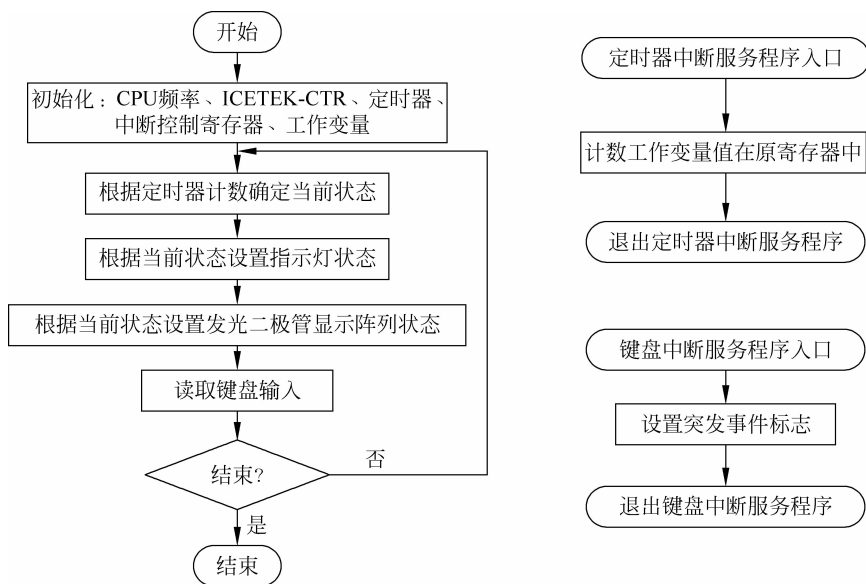


图 5.12 程序流程图

4. 实验原理

- (1) 交通灯控制要求。

利用 ICETEK-EDU 实验箱提供的设备,设计模拟实际生活中十字路口交通灯控制的程序。要求如下:

交通灯分红黄绿三色,东、南、西、北各一组,用灯光信号实现对交通的控制:绿灯信号表示通行,黄灯表示警告,红灯禁止通行,灯光闪烁表示信号即将改变。

计时显示: 8×8 点阵显示两位计数,为倒计时,每秒改变计数显示。

正常交通控制信号顺序：正常交通灯信号自动变换顺序如下所示：

- ① 南北方向绿灯,东西红灯(20 s)。
- ② 南北方向绿灯闪烁。
- ③ 南北方向黄灯。
- ④ 南北方向红灯,东西方向黄灯。
- ⑤ 东西方向绿灯(20 s)。
- ⑥ 东西方向绿灯闪烁。
- ⑦ 东西方向黄灯。
- ⑧ 返回①循环控制。

(2) 紧急情况处理。

模拟紧急情况(重要车队通过、急救车通过等)发生时,交通警察手动控制。

- ① 当任意方向通行剩余时间多于 10 s,将时间改成 10 s。
- ② 正常变换到 4 面红灯(20 s)。
- ③ 直接返回正常信号顺序的下一个通行信号(跳过闪烁绿灯、黄灯状态)。

(3) 交通灯模拟。

利用 ICETEK-CTR 上的一组发光二极管(共 12 只,分为东西南北 4 组、红黄绿 3 色)的亮灭实现交通信号的模拟。

(4) 计时显示。

利用 ICETEK-CTR 上的发光二极管显示阵列模拟显示。

发光二极管显示阵列的控制和编程方法可参考 4.2 节实验。

(5) 计时。

使用 TMS320F2812DSP 片上定时器,定时产生时钟计数,再利用此计数对应具体时间。定时器控制参考 3.4 节实验,中断控制可参考 3.3 节实验。

(6) 紧急情况。

利用 ICETEK-CTR 上键盘产生外中断,中断正常信号顺序,模拟突发情况。外中断编程控制可参考 3.3 节的程序。

(7) 程序设计。

根据设计要求,由于控制是由不同的各种状态按顺序发生的,可以采用状态机制控制方法来解决此问题,如图 5.13 所示。这种方法是:首先列举所有可能发生的状态;然后将这些状态编号,按顺序产生这些状态;状态延续的时间用程序控制。对于突发情况,可采用在正常顺序的控制中插入特殊控制序列的方式完成。时钟计数:采用 250 ms 一次中断进行累加计数。

其中,正常顺序每 112 s(计数值 448)为一个循环,状态“*”为非顺序状态。这样,只要根据计数值就可确定当前状态,根据状态再分情况处理。

状态编号	信号灯状态	状态定义	保持时间(计数值,起始时间,结束时间)	计数显示
1	南北绿灯,东西红灯	statusNSGreenEWRed	20 s(160,0,159)	20~0
2	南北绿灯闪烁,东西红灯	statusNSFlashEWRed	6 s(24,160,183)	0
3	南北黄灯,东西红灯	statusNSYellowEWRed	4 s(16,184,199)	20
4	南北红灯,东西黄灯	statusNSRedEWYellow	4 s(16,200,215)	20
5	南北红灯,东西绿灯	statusNSRedEWGreen	20 s(160,216,375)	20~1
6	南北红灯,东西绿灯闪烁	statusNSRedEWFlash	6 s(24,376,399)	0
7	南北红灯,东西黄灯	statusNSRedEWYellow	4 s(16,400,415)	20
8	南北黄灯,东西红灯	statusNSYellowEWRed	4 s(16,416,431)	20
*	南北红灯,东西红灯	StatusHold	20 s(160,0,159)	20~1

图 5.13 状态机制控制方法

对于计数显示,当处于状态 1、5、“*”中需要进行倒计时,需要计算在此状态中的计数值增量,根据增量判断是否更新计数显示。

5. 实验步骤

- (1) 实验准备。
 - ① 连接实验设备。
 - ② 连接实验箱附带的键盘的 PS2 插头到 ICETEK-CTR 的“键盘接口”P8 上。
 - ③ 将 ICETEK-CTR 板的供电电源开关拨动到“开”的位置。
- (2) 设置 Code Composer Studio 2.21,在硬件仿真(Emulator)方式下运行。
- (3) 启动 Code Composer Studio 2.21,选择 Debug→Reset CPU 菜单项。
- (4) 打开工程文件,工程目录为: D:\ ICETEK-F2812-A-EDUlab \ DSP281x _examples\lab0601-TrafficLight。
- (5) 编译并下载程序。
- (6) 运行程序观察结果,观察交通灯信号是否正常工作。
- (7) 突发事件控制。在 ICETEK-CTR 附带的小键盘上按下除 Enter 键外的按键,观察信号是否满足要求。
- (8) 结束程序运行退出。在 ICETEK-CTR 附带的小键盘上按下 Enter 键。
- (9) 退出 CCS。

6. 实验结果

程序可以完成交通灯功能,顺序循环工作。

在中断信号到来后,进入特殊过程:当前计时如果大于 10 改成 10,否则不变,等待状

态切换；切换后进入 4 面禁行状态，计数 20 s 后返回断点后的通行（有两方向是绿灯）状态。

7. 思考题

请考虑修改此实验程序完成主路与人行横道可由行人控制的交通信号控制。控制程序流程图如图 5.13 所示，具体要求如下：

- (1) 平时为主路机动车通行(绿灯)状态，人行横道为红灯。
- (2) 行人需要通过人行横道，按下交通灯控制按钮。根据以下情况处理：
 - ① 如果机动车刚刚恢复通行不超过 3 min，行人需要等待 3 min 计时满。
 - ② 否则行人等待 1 min 计时(从按下按钮时起)。
 - ③ 机动车道绿灯闪烁。
 - ④ 机动车道黄灯。
 - ⑤ 机动车道红灯，人行横道绿灯，并开始计时 1 min。
 - ⑥ 人行横道绿灯闪烁。
 - ⑦ 人行横道黄灯。
 - ⑧ 返回第①步。

5.3 无刷直流电动机控制

1. 实验目的

(1) 熟悉使用 ICETEK-F2812-A 评估板连接电动机控制板 ICETEK-MOTOR-E 的方法。

- (2) 掌握 TMS320F2812DSP 定时器的使用和编程。
- (3) 掌握 TMS320F2812DSP 外中断的使用和编程。
- (4) 掌握单路/多路模数转换(A/D)的使用和编程。
- (5) 掌握单路/多路数模转换例程(D/A)的使用和编程(多路数字波形产生例程)。
- (6) 掌握自启动(自举)的使用和编程。
- (7) 掌握异步串口通信的使用和编程。
- (8) 掌握多路信号混频的使用和编程。
- (9) 掌握 PWM 输出的使用和编程。
- (10) 掌握 CAN 总线通信的使用和编程。
- (11) 学习复杂控制程序的设计思路。

2. 实验设备

ICETEK-F2812-EDU 实验箱如图 5.14 所示, ICETEK-MOTOR-E 控制板和无刷直流电动机如图 5.15 所示。

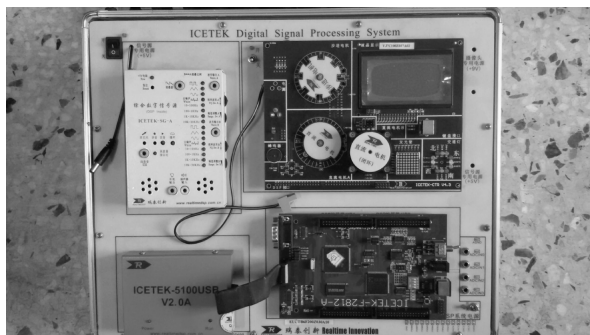


图 5.14 ICETEK-F2812-EDU 实验箱



图 5.15 ICETEK-MOTOR-E 控制板和无刷直流电动机

3. 实验内容

通过 ICETEK-MOTOR-E 控制板和无刷直流电动机,完成对三相无刷直流电动机的控制。通过霍尔传感器反馈,实现有传感器或无传感器的闭环控制。

4. 实验原理

前面介绍了直流电动机的开环和闭环控制,而 20 世纪 50 年代诞生了一种新型的控制元件——无刷直流电动机,它与直流电动机相比具有一系列的优点,近年得到了迅速发展,在许多领域的竞争中不断取代直流电动机和异步电动机。无刷直流电动机的最大特点是没有直流电动机的电刷和换向器,避免在换向时产生火花引起无线电干扰,换向器电刷装置又使直流电动机结构复杂、噪声大且维护困难。

无刷直流电动机是由电动机、转子位置传感器和电子开关电路 3 部分组成,无刷直流电动机是根据转子位置反馈信息采用电子换向运行的交流永磁同步电动机,直流电源通过开关电路向电动机定子绕组供电,电动机转子位置由位置传感器检测并提供信号去触发开关电路中的功率开关元件使之导通或截止,从而控制电动机的转动。

利用 ICETEK-EDU 实验箱和 ICETEK-MOTOR-E 控制板,设计无刷直流电动机控制系统的控制程序。运用 PID 控制算法、有限冲击响应滤波器(FIR)算法、无限冲击响应激光器(IIR)算法、快速傅里叶变换(FFT)算法、卷积算法、自适应滤波器算法、抽样定理、离散余弦变换(DCT)算法等方法,用 MATLAB 实现 FIR 滤波器的设计并实现信号滤波,最终实现对无刷直流电动机的正、反转和调速控制。

无刷直流电动机闭环调速控制系统:在 DSP 系统实验箱(如图 5.14 所示)上插入 ICETEK-MOTOR-E 控制板(如图 5.2 所示),通过霍尔传感器对转速进行信号采集,经过 DSP 芯片中的 A/D 转换器,将转速的模拟值转换成数字量,与给定值进行比较,产生偏差输入到 PID 控制器中,通过计算形成控制系统的输出,DSP 可实时读取当前转速值,显示在液晶屏上,同时可以显示的还有给定值、转速的上限值等需要显示的内容。利用 DSP 中运行的 PID 控制程序,根据计算形成控制系统的输出控制 PWM 的脉宽,调节占空比,从而实现无刷直流电动机调速的闭环控制。

电动机控制板——ICETEK-MOTOR-E,具有最大输出功率达 40 W;能完成对三相无刷直流电动机的控制;能对电动机进行正、反转控制和调速控制;具有霍尔传感器反馈,能实现有传感器或无传感器的闭环控制;具有过电流保护和死区保护;具有相电流反馈和电压反馈;+12 V 电源输入。

5. 实验步骤

(1) 实验准备。

① 连接实验设备。

② 将 ICETEK-MOTOR-E 卡插到实验台 ICETEK-F2812-A 的转接口上,将无刷直流电动机线连接到 ICETEK-MOTOR-E 控制板上。

③ 串口接收设置。在 PC 上运行 Windows 自带的超级终端串口通信程序(波特率为 9600、1 位停止位、无校验位、无硬件流控制);或者使用其他串口通信程序。

(2) 设置 Code Composer Studio 2.21 在硬件仿真(Emulator)方式下运行。

(3) 启动 Code Composer Studio 2.21,选择 Debug→Reset CPU 菜单项。

(4) 打开工程文件,工程目录为: D:\ ICETEK-F2812-A-EDUlab \ DSP281x _examples\lab0608- MOTOR。

(5) 编译并下载程序。

(6) 运行程序观察结果,观察电动机是否正常工作。

(7) 结束程序运行退出。

(8) 退出 CCS。

6. 实验结果

程序可以通过 PID 控制,完成无刷直流电动机的正、反转和调速控制工作。

7. 思考题

请考虑修改此实验程序,控制算法除了采用 PID 完成,还有有限冲击响应滤波器(FIR)算法、无限冲击响应激光器(IIR)算法、快速傅里叶变换(FFT)算法、卷积算法、自适应滤波器算法、抽样定理、离散余弦变换(DCT)算法等多种方法。

5.4 大功率交流异步电动机控制

1. 实验目的

- (1) 熟悉使用 ICETEK-F2812-ACM 的方法。
- (2) 掌握 DSP 定时器、响应外部中断的设计与应用。
- (3) 掌握 DSP 数据存取、定点数除法的设计与应用。
- (4) 掌握单路/多路模数转换(A/D)、数模转换(D/A)的设计与应用。
- (5) 掌握 LED 显示、液晶显示器控制和键盘输入的设计与应用。
- (6) 掌握自启动(自举)的设计与应用。
- (7) 掌握异步串口通信的设计与应用。
- (8) 掌握音频信号发生、多路信号混频的设计与应用。
- (9) 掌握 SPWM 发生原理、CAN 总线通信的设计与应用。
- (10) 掌握闭环直流电动机控制的设计与应用。
- (11) 学习复杂控制系统的程序设计思路。

2. 实验设备

计算机,ICETEK-F2812-ACM 实验箱和大功率交流异步电动机如图 5.16 所示。

3. 实验内容

通过 ICETEK-F2812-ACM 交流异步电动机拖动系统,实现速度环、功率环、电流环等多样化的闭环控制。



图 5.16 ICETEK-F2812-ACM 实验箱和大功率交流异步电动机

4. 实验原理

前面给出了直流电动机的开环和闭环控制、无刷直流电动机的开环和闭环控制,而在发电、输电和配电的电力系统中,交流电动机以结构上坚固耐用、造价较低等一系列的优势,成为产业用电动机的主流。交流电动机的最大特点是不需要电刷和换向器,结构简单,随着电力电子技术和控制技术的发展,交流电动机的变频调速技术和矢量控制技术开始得到应用。

交流异步电动机有转子和定子两大部分,在定、转子之间有很小的工作气隙。在使用时,交流异步电动机的励磁绕组两端施加幅值恒定的交流励磁电压,而控制绕组两端则施加经常改变的交流控制电压,当定子两相绕组都加有交流电压时,交流异步电动机将很快地转动起来,把电信号转换成机械转角或转速。

利用 ICETEK-F2812-ACM 交流异步电动机拖动系统,设计大功率交流异步电动机控制系统的控制程序。运用 PID 控制算法、有限冲击响应滤波器(FIR)算法、无限冲击响应激光器(IIR)算法、快速傅里叶变换(FFT)算法、卷积算法、自适应滤波器算法、抽样定理等方法,实现对电动机变频变压调速控制、电动机空间电压矢量控制、SPWM 产生及控制。

大功率交流异步电动机闭环控制系统: ICETEK-F2812-ACM 交流异步电动机拖动系统(如图 5.16 所示),经过 DSP 芯片中的 A/D 转换器,将转速的模拟值转换成数字量,与给定值进行比较,产生偏差输入到 PID 控制器中,通过计算形成控制系统的输出,DSP 可实时读取当前转速值,显示在液晶屏上,同时可以显示的还有给定值、转速的上限值等

需要显示的内容。利用 DSP 中运行的 PID 控制程序,根据计算形成控制系统的输出控制 PWM 的脉宽,调节占空比,从而实现大功率交流异步电动机的闭环控制。

ICETEK-F2812-ACM 交流异步电动机拖动系统,是以 TMS320F2812PGFA DSP 为主处理器件,以大功率交流异步电动机为主要控制对象,充分体现了 F2812DSP 对电动机的控制能力。该系统具有大功率电动机驱动,可以驱动 80~3000 W 电动机;能完成对交流异步电动机的控制;能对直流无刷电动机、永磁同步电动机和步进电动机进行控制和调速;能实现速度环、功率环、电流环等多样的闭环控制;具有高压区绝缘保护及过电流、过电压保护和死区保护功能;具有良好的人机接口、键盘显示等综合 DSP。

5. 实验步骤

(1) 实验准备。

① 连接实验设备。

② 将大功率交流异步电动机线连接到 ICETEK-F2812-ACM 系统上。

③ 串口接收设置。在 PC 上运行 Windows 自带的超级终端串口通信程序(波特率为 9600、1 位停止位、无校验位、无硬件流控制);或者使用其他串口通信程序。

(2) 设置 Code Composer Studio 2.21 在硬件仿真(Emulator)方式下运行。

(3) 启动 Code Composer Studio 2.21,选择 Debug→Reset CPU 菜单项。

(4) 打开工程文件,工程目录为: D:\ ICETEK-F2812-A-EDUlab \ DSP281x _ examples\Lab0609-ACM。

(5) 编译并下载程序。

(6) 运行程序观察结果,观察电动机是否正常工作。

(7) 结束程序运行退出。

(8) 退出 CCS。

6. 实验结果

程序可以通过 PID 控制,完成大功率交流异步电动机调速控制工作。

7. 思考题

请考虑修改此实验程序,控制算法除了采用 PID 完成,还有有限冲击响应滤波器(FIR)算法、无限冲击响应滤波器(IIR)算法、快速傅里叶变换(FFT)算法、卷积算法、自适应滤波器算法、抽样定理等多种方法。