“基于 NI myDAQ 的自动控制原理实验套件” 设计说明

自动控制原理实验室

2019 年 12 月 11 日

Abstract

本项目以 MATLAB/Simulink 为实验环境，将 National Instruments 公司出品的 myDAQ 可编程测量设备作为模拟/数字接口，使计算机与外部硬件交互，进而设计了用于自动控制原理教学实验的三套实验设备。每个实验所需的硬件以一块板卡的形式与 myDAQ 连接，并可与 MATLAB 中的软件控制器或虚拟受控对象交互。通过更换板卡，学生可选择进行不同的实验。

Contents

1 NI myDAQ 接线端子适配器

1.1 设计目的

NI myDAQ 可编程测量设备的右侧带有 20 个螺丝接线端子，原本被设计用于接入单根导线，可为原型实验板上的项目供电，并通过模拟数字 I/O 通道读取数据。在此项目中 myDAQ 将与固定的数个实验板卡连接，且实验板卡需要经常更换，因此需要设计一种适配器板卡，将螺丝接线端子转换为易于插拔的连接器。

1.2 连接器选型

考虑到设备由经验不一定丰富的学生进行操作，用于与实验板卡对接的连接器需要满足以下要求：

1. 连接器或线缆至少可以承载 20 路信号，其中包含 5V/15V 供电线路。
2. 具有良好的防呆（防止接反、接错位）特性；
3. 需要足够牢固，具有良好的机械性能；
4. 使用寿命足够长。

最初考虑的是使用 2.54mm 间距的排针与排座进行对接。但是这种方法十分容易被连接错位，并且机械强度不够高。之后考虑了在转接器和实验板卡之间使用线缆连接。USB Type-C 连接器在区分正反插的条件下可以承载 20 路信号，但是这种连接器和线缆成本高且不易买到。最终使用的是 D-Sub 25 针连接器。连接器可被焊在转接器右侧和实验板卡左侧，可以直接对接，也可以使用线缆连接。插座周围有金属外壳，可以防止接反和接错位。

1.3 硬件与 PCB 设计

转接器上除了包含必要的电源滤波、退耦电路外，设计了 300mA 自恢复保险丝来提供额外电气保护。myDAQ 设备上只有一个蓝色指示灯，当 USB 设备枚举成功时会点亮。转接器上为 ±15V 模拟组件电源、5V 数字组件电源预留了指示灯。如出现电源电流过大的情况，myDAQ 自动关闭电源或自恢复保险丝断路，指示灯熄灭，可帮助及时发现故障。

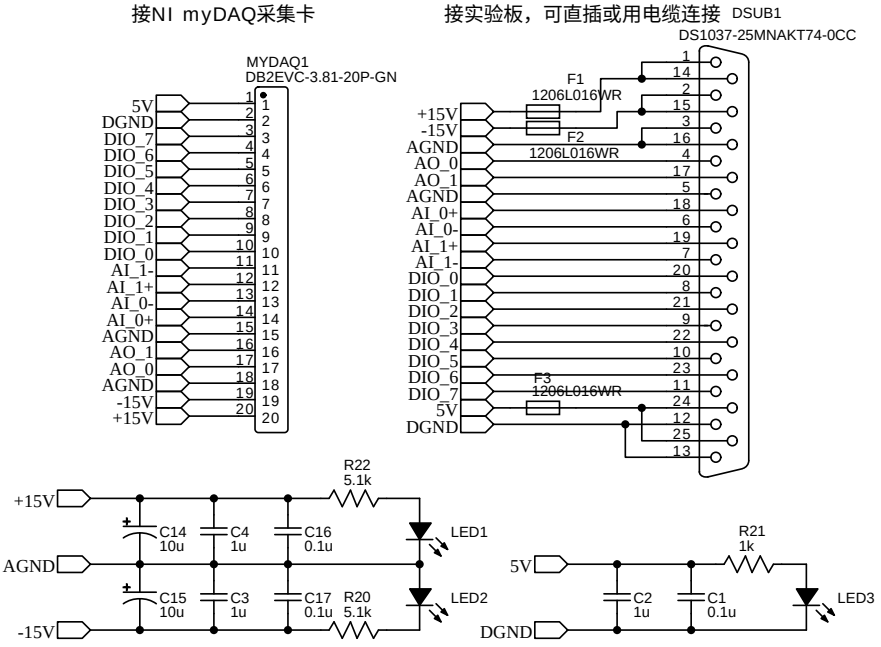

Figure 1: myDAQ 接线端子适配器的原理图

2 模拟电路比例-积分-微分（PID）控制器

2.1 设计目的

板卡上设计了基于运算放大器的 PID 调节器电路。myDAQ 以模拟电压的形式给出一个误差量，电路将通过其比例-积分-微分环节输出一个调节量，并反馈到 myDAQ 的模拟输入通道中。在

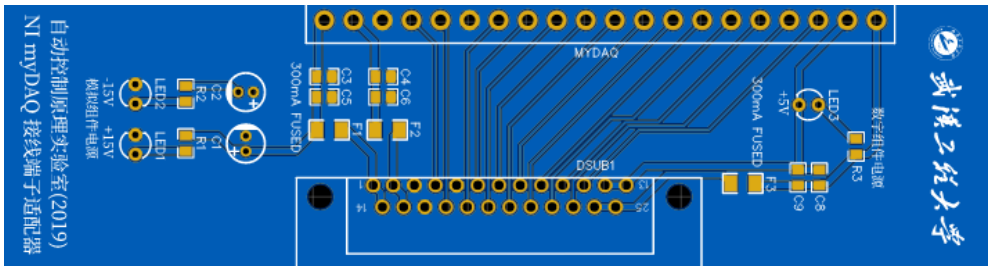

Figure 2: myDAQ 接线端子适配器的 PCB 设计图

Simulink 环境中以离散传递函数的形式建立一个虚拟对象，例如一个虚拟直流电机，并使此对象接受硬件 PID 调节器的控制。学生可使用拨码开关设置 PID 调节器的参数，使用虚拟示波器等 Simulink 中的工具观察 PID 调节器与受控对象的状态。

2.2 PID 调节器与硬件设计

图 3为 PID 调节器的原理图。SW1 – SW5 为 DIP 拨码开关，在进行实验时，学生可选择不同参数的电阻或电容接入电路。

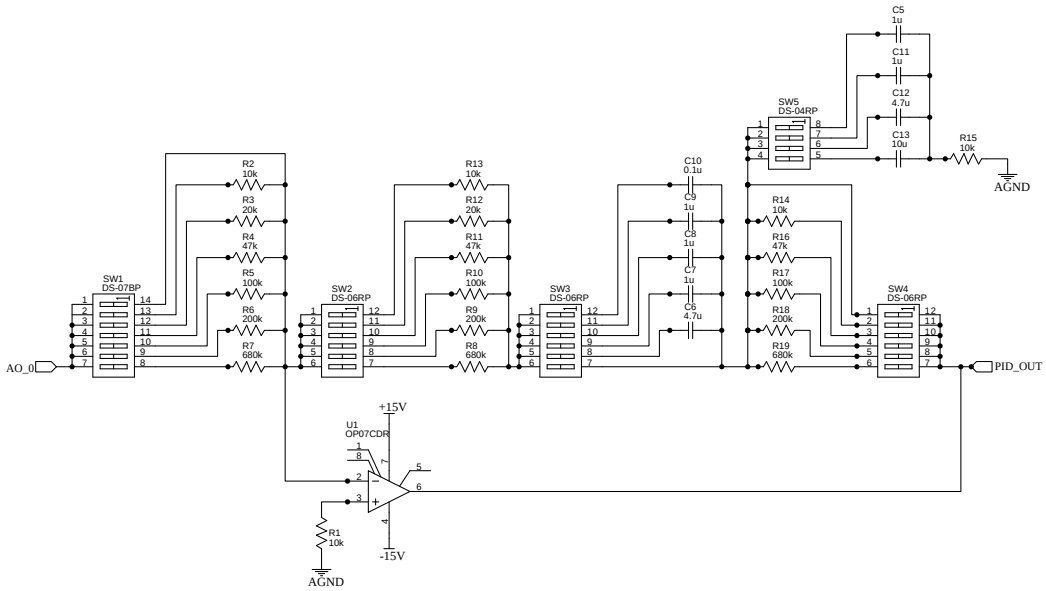

Figure 3: PID 调节器的原理图

图 4为 PID 实验板卡的设计图。板卡上设计了原理图丝印，并将拨码开关放置于与丝印原理图匹配的位置，方便学生调节和理解。板上设计了用于供电和信号引出的焊盘，可安装 2mm 香蕉插座。接入外接电源后，板卡脱离 myDAQ 实验环境也可使用。

板上设计了调节器输出指示灯，可用于指示 PID 调节器正在增加或减少控制量的状态。板上有一个单圈电位器，用于手动输出-10V – +10V 模拟电压接入被控对象。PID 和手动模式可使用钮子开关 U7 进行切换。

2.3 虚拟对象的 Simulink 模型

在 Simulink 中设计了图 5所示的模型。其中的虚拟对象（电机）以一个二阶离散传递函数表示：

$$\hat{y}(k) = \frac{0.3816 \cdot z^{-1}}{1 - 0.4219 \cdot z^{-1} - 0.4749 \cdot z^{-2}} \cdot u(k) \tag{1}$$

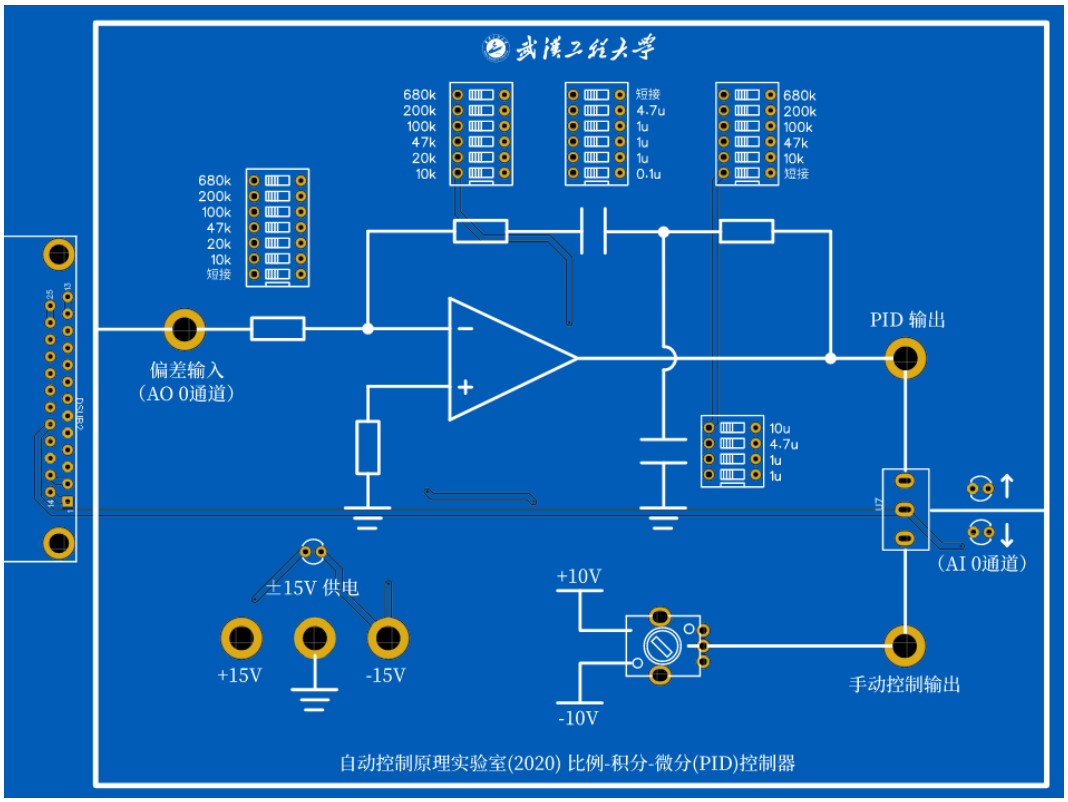

Figure 4: PID 控制器实验 PCB 设计图

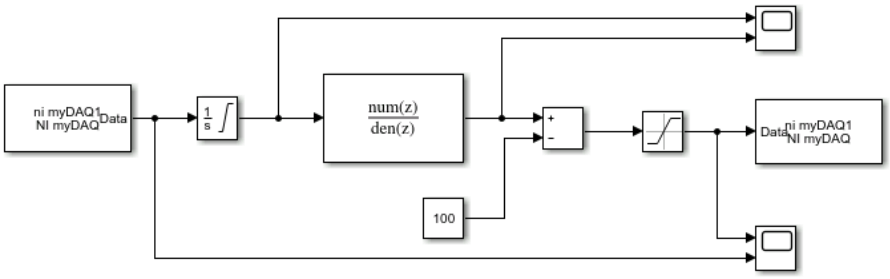

Figure 5: 用于实验的 Simulink 模型

此传递函数是由一个实际直流电机利用系统辨识方法生成的。图 6为在同一激励源下，模型的响应与实际电机的响应。由图可知二阶模型具有良好的拟合效果，并且误差在可接受范围内，可以用于实验。

3 直流电机调速实验

3.1 设计目的

板卡上设计了直流电机驱动、转速测量与电流测量电路。myDAQ 输出一个 0–10V 的模拟电压信号来控制板上电机的功率，电机的状态经过光电对管和电流采样电路测量，以 0–10V 模拟电压的形式送至 myDAQ 的模拟输入端口上，对应转速测量范围 0 – 10000 RPM，电流测量范围 0 – 1000 mA。学生可在 Simulink 环境中编写控制器（P，PI，PID 等）对电机转速进行测控，或者实现直流电机的电流-转速双闭环调节器。

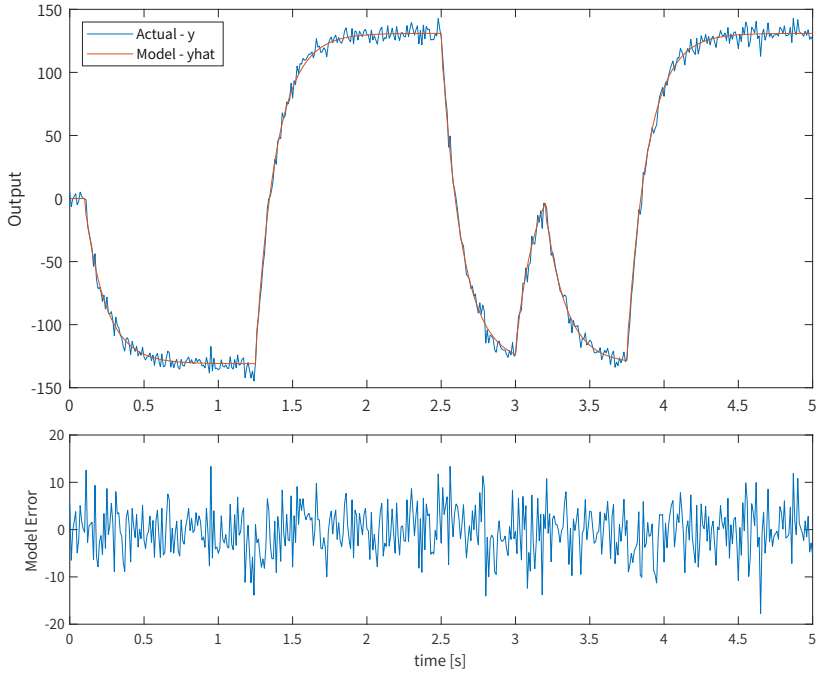

Figure 6: 同一激励源下电机响应与二阶传递函数模型响应的比较

3.2 光电转速测量电路

图 7为电机转速测量电路原理图。U2 为 ITR9606 型红外光电传感器，U3 为 STC 公司出品的 STC15W204S 型 8051 内核嵌入式微控制器，U4 为 LM358 型运算放大器，使用 15V 单端供电方式。

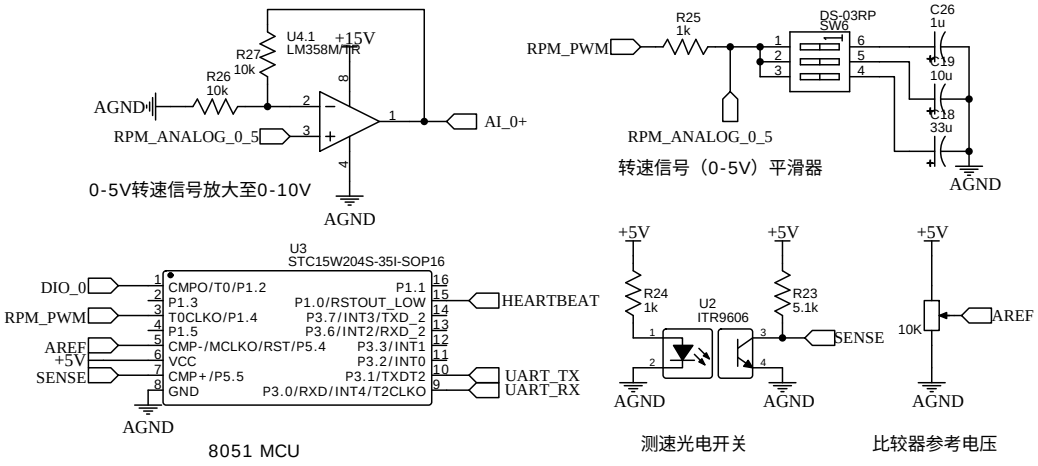

Figure 7: 光电转速测量电路原理图

直流电机通过专用支架安装在实验板上，其输出轴上带有 12 孔的码盘，码盘的打孔区位于 ITR9606 型红外光电传感器的槽中。电机每旋转一周，光电传感器将产生 12 个周期的高低电位信号。嵌入式微控制器 U3 内置一路模拟电压比较器，并可产生比较器中断。通过编写应用程序即可测出单位时间内光电传感器输出信号的周期数，进而测得电机转速。用于实现此功能的 MCU 程序已列于附录中。

MCU 根据所测出的转速产生一路推挽输出模式的 PWM 信号，经 RC 电路平滑为直流电压后，被 U4 放大至 2 倍，形成范围为 0–10V 的转速电压信号。学生可根据需要选择不同大小的电容，决定电压信号的平滑程度。选择电容较小时转速信号响应迅速，但可在 myDAQ 端测出

较大噪声。选择电容较大时可获得高信噪比的信号，同时也会引入延迟。

3.3 电机驱动与电流采样电路

图 8为电机驱动与电流采样电路的原理图。Q2 为 Fairchild 公司出品的 8N60C 型 N 沟道增强型 MOSFET（可用其他型号 N 沟道增强型 MOSFET 替换，但需重新调节栅极电压增益）。

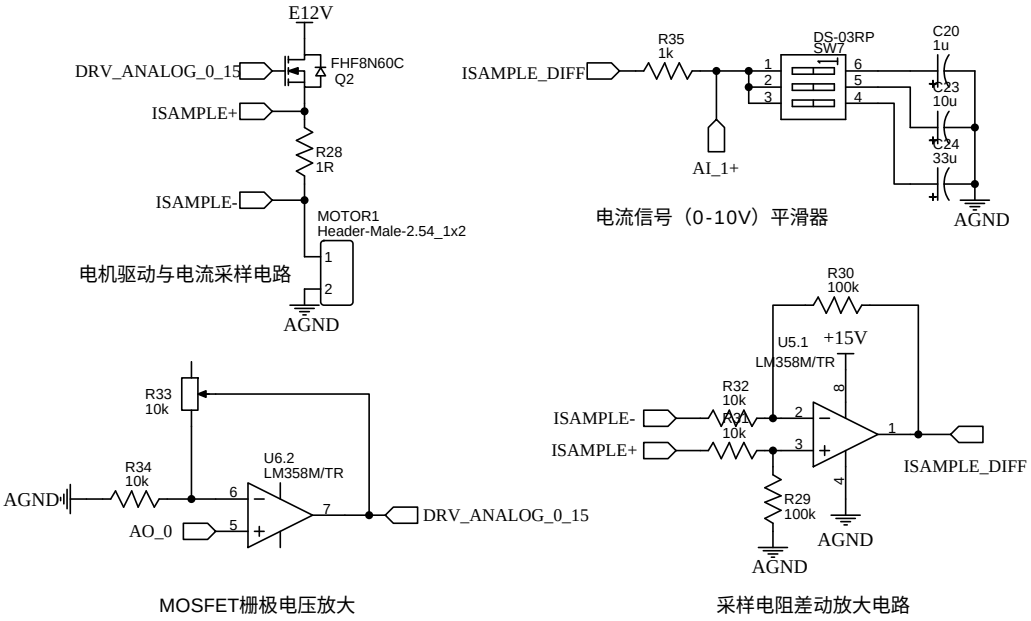

Figure 8: 电机驱动与电流采样电路原理图

myDAQ 提供的 0 – 10V 功率控制信号经 U6.2 放大输入 Q2 的栅极。调试电路时，运行提供的 MATLAB 程序手动给出 10V 控制信号，并调节 R33 使 Q2 临界达到饱和状态（电机转速临界增加到最大）。在该配置下，当 myDAQ 提供 0 – 10V 信号时，Q2 工作在可变电阻区，等效于将电阻 R_{DS} 串联在电机回路中，实现对电机的调速。

电机工作时，电流流过阻值为 1Ω 的功率电阻 R28，其两端产生与电流线性相关的电势差，经差动放大电路放大 10 倍后送至 myDAQ 的模拟输入端。此时 1A 电流对应 10V 电压，即 myDAQ 的最大测量范围。经过测试，电机两端电压 12V 的条件下发生堵转时，电流不超过 0.8A，故此测量范围适合进行实验。

3.4 实验板卡 PCB 设计

图 9为直流电机调速实验的 PCB 设计图。学生除可观察电机运转状态外，还可控制 MCU 内置电压比较器的比较阈值，观察比较结果，设置电压信号 RC 滤波器的平滑程度，以及观察频率测量程序的运行情况。

myDAQ 的电源输出总功率不可超过 500mW。板卡上预留了一个 12V 外部电源输入接口（内正外负），用以为直流电机提供高功率驱动电源。

3.5 Simulink 实验模型示例

4 温度控制实验

4.1 设计目的

板卡上设计了温度测量，加热器与风扇驱动电路，以及一个大型金属功率电阻作为被控对象，用于进行温度闭环控制实验。myDAQ 输出一个 0–10V 的模拟电压信号来控制加热器驱动电路中

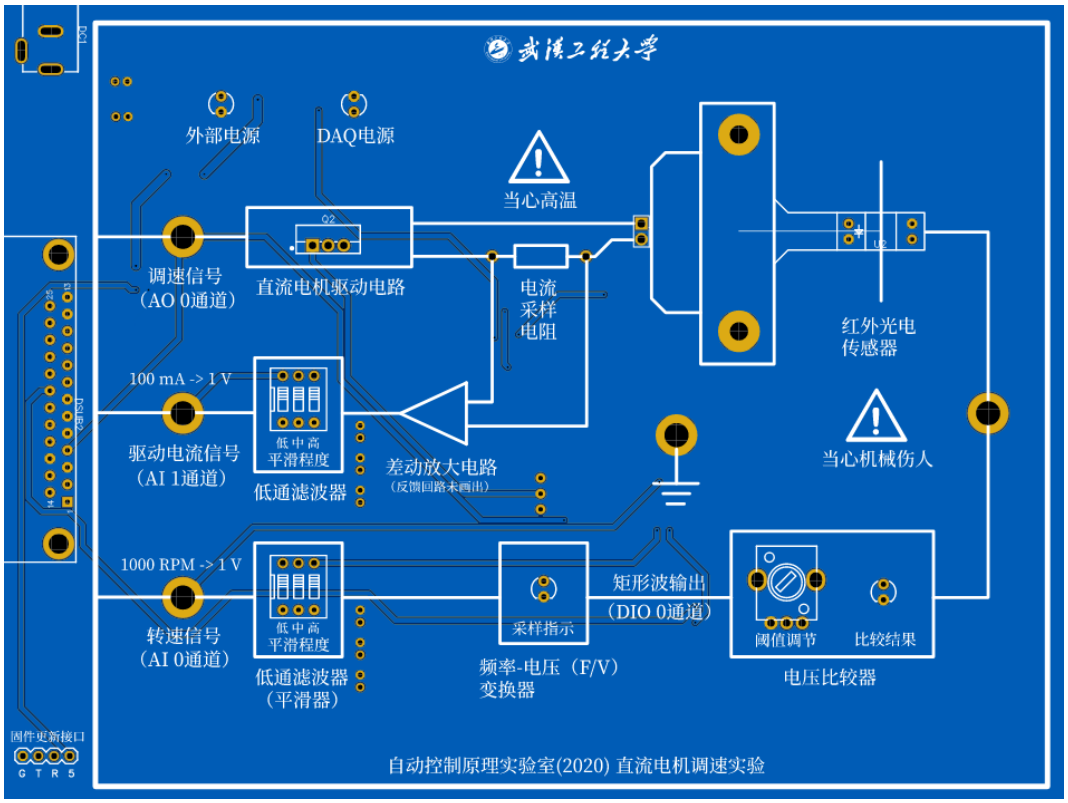

Figure 9: 直流电机调速实验 PCB 设计图

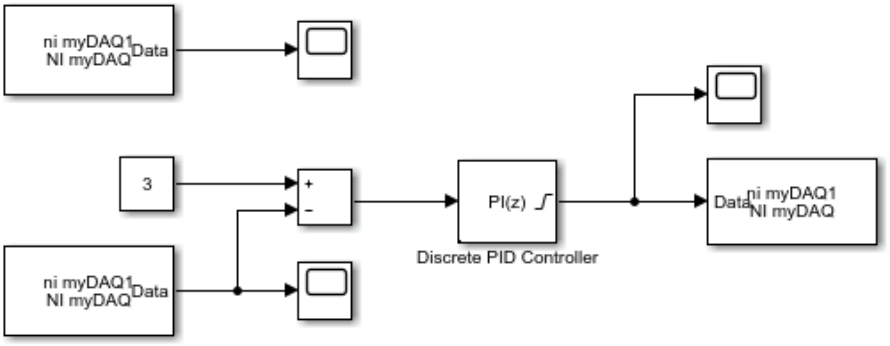

Figure 10: 直流电机调速实验 Simulink 模型示例

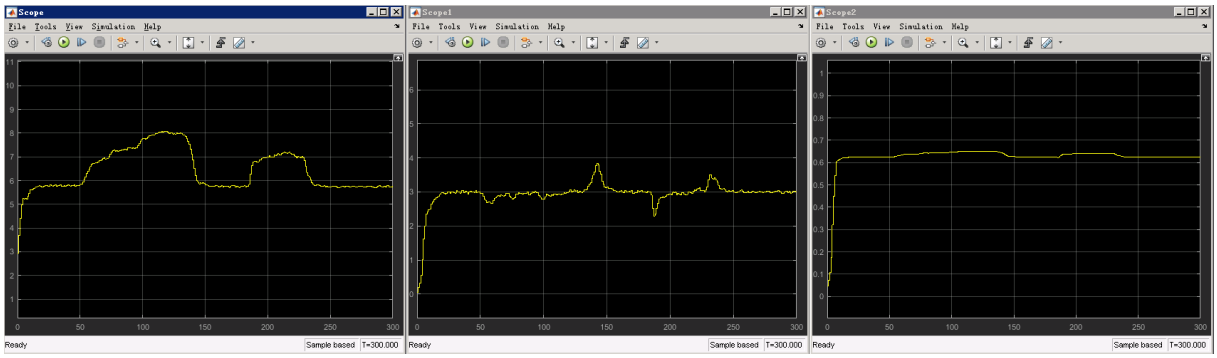

Figure 11: 直流电机调速实验 Simulink 运行结果

PWM 的占空比，实现对加热器功率的调节。板上设计有 AD590 温度测量集成电路，其信号经过运算放大器变换成 0–10V 模拟电压送至 myDAQ 的模拟输入通道，对应的测量范围为 0–100 ℃。学生可在 Simulink 环境中编写控制器来使被控对象（功率电阻）的温度稳定在某一设定值，也可开启风扇来调节被控对象的散热功率，以测试控制器的控制效果。

4.2 PWM 生成与加热器驱动电路

在上一节介绍的电机调速实验中，MOSFET 工作在线性区实现电机调速。经过实测，电机的平均电流在 0.2 – 0.5A 之间，只要不出现长时间堵转的情况，不会造成 MOSFET 严重发热。在本实验中，加热器电流可达 1A，需要采用 PWM 驱动的方式，使 MOSFET 仅工作在饱和和截止区，避免 MOSFET 发热。图 12为 PWM 生成电路与加热器驱动电路的原理图。

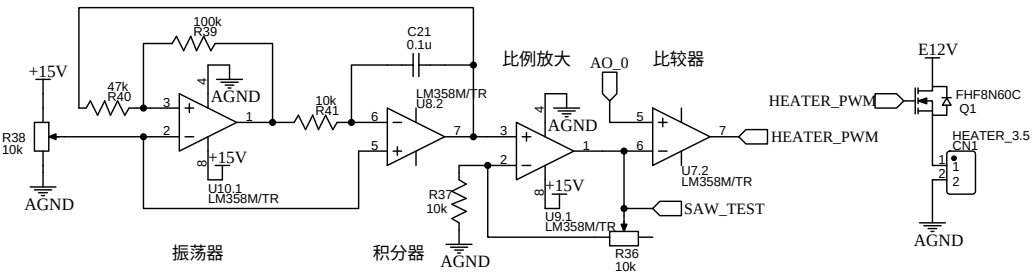

Figure 12: PWM 生成与加热器驱动电路原理图

运放 U10.1 构成自激振荡电路生成一定频率的方波，被运放 U8.2 积分后形成三角波。U9.1 构成比例放大电路，将三角波的振幅调整至 0 – 10 V。U7.2 以开环方式工作，作为电压比较器，将来自 myDAQ 的模拟电压与三角波进行比较，输出 PWM 波。当来自 myDAQ 的电压变化时，PWM 的占空比相应变化，myDAQ 的最大输出电压 10V 使 PWM 波的占空比为 100%，即恒为高电平，加热器以最大功率加热。

4.3 温度测量电路

图 13为温度测量电路的原理图。U12 为 ADI 公司出品的 AD590 型双端温度变送器，其输出电流信号与绝对温度成比例，每开尔文（K）对应电流 1μA。R46 为电流采样电阻。

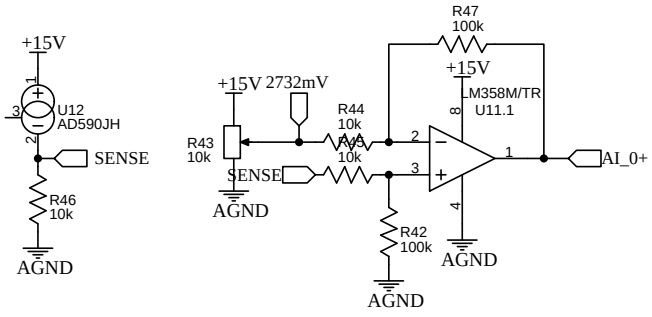

Figure 13: 温度测量电路原理图

R46 上的电压被 U11.1 构成的差动放大器减去 2732mV（对应 0℃ 与绝对零度-273℃ 的温差），并放大至 10 倍，得到代表温度的电压。0℃ 为 0V，100℃ 为 10V（myDAQ 的最大测量范围）。

4.4 散热风扇控制电路

图 14为散热风扇控制电路。U14 为 817C 型光电耦合器，K1 为松乐公司出品的 SRS-05VDC 型电磁继电器。

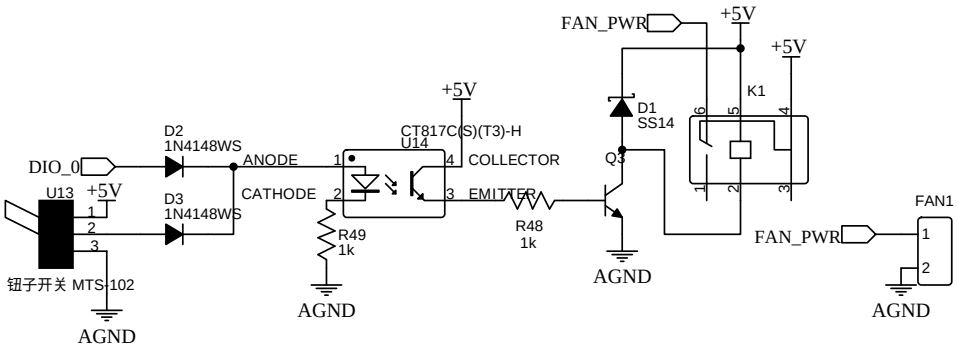

Figure 14: 散热风扇控制电路原理图

FAN1 为间距 2.54mm 的插座，用于连接 5V 0.12A 无刷风扇。钮子开关 U13 为散热风扇强制启动开关，便于学生随时调整系统散热功率或使受控对象快速降温。二极管 D2，D3 构成或门，myDAQ 的数字输出或钮子开关中的任意一个为高电平，即启动风扇。

4.5 实验板卡 PCB 设计

图 15为温度控制实验的 PCB 设计图。学生可观察被控对象的状态，以及加热器、风扇的运行状态。

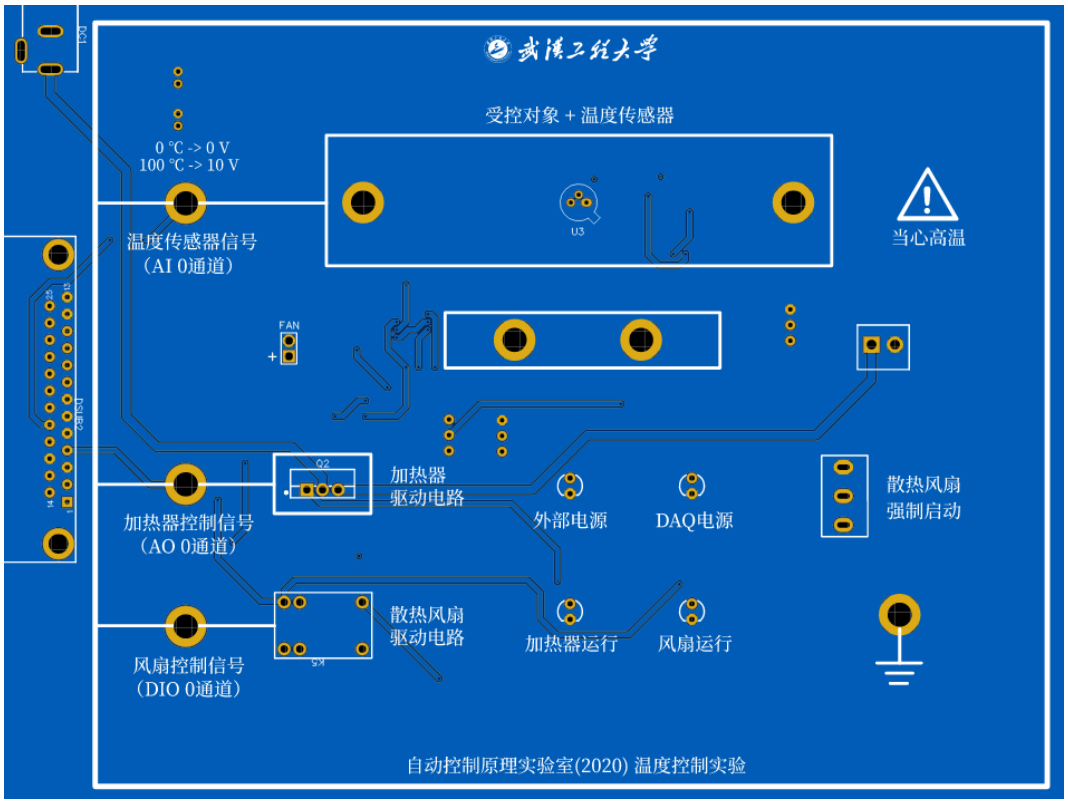

Figure 15: 温度控制实验 PCB 设计图

A 8051 MCU 电机测速固件

程序需使用 Realview Keil A51 汇编器编译，使用 STC-ISP 下载固件时需选择内部 PLL 时钟频率为 35MHz。

```
;; SYSCLK 24 MHz

;; Extra GPIO data and mode SFRs
        p5 data 0c8h
        p1m1 data 091h
        p1m0 data 092h
        p3m1 data 0b1h
        p3m0 data 0b2h
        p5m1 data 0c9h
        p5m0 data 0cah

;; Comparator SFRs
        cmpcr1 data 0e6h
        cmpen equ 080h
        cmpif equ 040h
        pie equ 020h
        nie equ 010h
        pis equ 008h
        nis equ 004h
        cmpoe equ 002h
        cmpres equ 001h

        cmpcr2 data 0e7h
        invcmpo equ 080h
        disfit equ 040h
        lcdty equ 03fh

;; Timer SFRs and constants
        auxr data 08eh
        intclko data 08fh
        ie2 data 0afh
        t2h data 0d6h
        t2l data 0d7h
        pwmcnt equ 10000h       ; Duty range: 65535 downto 0
        rotcnt equ 1c23h        ; Encoder count interrupt interval 50Hz

;; Variables
        pwmflag bit 20h.0
        pwmdutyh_h equ 031h
        pwmdutyh_l equ 030h
        pwmdutyl_h equ 033h
        pwmdutyl_l equ 032h

;; Onboard peripherals
        dac bit p1.4
        led bit p1.0

org 0000h                       ; Reset
        ljmp reset

org 00abh                       ; Comparator interrupt
        ljmp cmp_isr

org 000bh                       ; Timer 0 interrupt
        ljmp t0_isr

org 0063h
        ljmp t2_isr

org 0100h                       ; Principal
reset:
        ;; Stack initialise
        mov sp, #60h

        ;; GPIO initialise
```

```
        mov p1m0, #00010101b    ; DAC, LED, CMP PP mode
        mov p1m1, #00000000b
        mov p3m0, #00000000b
        mov p3m1, #00000000b
        mov p5m0, #00000000b
        mov p5m1, #00000000b

        ;; Comparator initialise
        mov cmpcr1, #11110110b  ; cmpen, cmpif, pie, nie, pis, nis, cmpoe, cmpres(ro)
        mov cmpcr2, #00001000b  ; invcmpo, disfit, lcdty(5 downto 0)

        ;; Timer (PWM) initialise
        orl auxr, #10000000b    ; Timer 0 1T mode
        mov intclko, #00000001b ; Timer 0 clock output enable
        anl tmod, #11110000b    ; Timer 0 mode 0
        mov tl0, #00001000b
        mov th0, #00001000b
        setb dac
        clr pwmflag
        setb tr0                ; Timer 0 start
        setb et0                ; Timer 0 interrupt enable

        ;; Timer (RPM counter) initialise
        anl auxr, #11111011b    ; Timer 2 12T mode;
        mov t2l, #low rotcnt
        mov t2h, #high rotcnt
        orl auxr, #00010000b    ; Timer 2 start
        orl ie2, #00000100b     ; Timer 2 interrupt enable

        ;; Interrupt initialise
        setb ea

        ;; Select working registers in PSW
        clr rs0
        clr rs1

loop:
        cjne r1, #5d, loop
        mov r1, #0d             ; Reset interval counter
count:
        cpl led
        inc r0
        mov pwmdutyh_h, r0
        mov pwmdutyh_l, #0d

        mov a, #0ffh            ; inv16 algorithm: subtract 0ffh by pwmdutyh
        clr c
        subb a, #0d
        mov pwmdutyl_l, a
        mov a, #0ffh
        subb a, r0
        mov pwmdutyl_h, a
        mov r0, #0d             ; Reset pulse counter

restart:
        sjmp loop

cmp_isr:
        anl cmpcr1, #not cmpif
        inc r0
        reti

t0_isr:
        cpl pwmflag
        jnb pwmflag, rdylow
rdyhigh:
        mov tl0, pwmdutyl_l
        mov th0, pwmdutyl_h
        jmp t0_isrret
rdylow:
```

```
        mov tl0, pwmdutyh_l
        mov th0, pwmdutyh_h
t0_isrret:
        reti

t2_isr:
        inc r1
        anl ie2, #11111011b     ; Re-enable interrupt
        orl ie2, #00000100b
        reti

end
```