山东职业学院
SHANDONG POLYTECHNIC

毕 业 设 计 论 文

题　　目：基于单片机的转辙机功能模拟设计

系　　部：铁道系

专　　业：铁道信号自动控制

班　　级：1933

姓　　名：曾繁华

学　　号：20103023305

指导教师：孙会祥

2021 年 12 月

山东职业学院

毕业设计（论文）任务书

班 级	铁道信号1933	学生姓名	曾繁华	指导教师	孙会祥
设计（论文）题目		基于单片机的转辙机功能模拟设计			
主要研究内容	转辙机是用以可靠地转换道岔位置，改变道岔开通方向，锁闭道岔尖轨，反映道岔位置的重要的铁道信号基础设备，它可以很好地保证行车安全，提高运输效率，改善行车人员的劳动强度。 本课题在分析转辙机组成结构及工作原理的基础上，设计基于单片机并采用直流电动机或步进电机作为执行器的转辙机功能模拟装置，能够正确模拟转辙机的工作过程。根据设计要求完成系统软硬件设计、仿真并调试运行。				
主要技术指标或研究目标	1.对转辙机模拟装置设计任务进行分析； 2.综述转辙机的实现方案，比较各方案的优缺点； 3.分析转辙机工作原理，提出基于单片机的转辙机模拟装置设计方案； 4.完成装置硬件原理图设计，确定元器件清单；搭建硬件装置及电路并仿真调试； 5.完成软件设计及软硬件联调仿真； 6.完成毕业设计论文。				
基本要求	1.独立完成毕业设计工作； 2.设计合理、满足功能要求； 3.结构简单可行； 4.工作量充足				
主要参考资料及文献	1.铁路信号基础设备； 2.电子技术基础； 3.传感器与检测技术； 4.电路设计与仿真； 5.单片机应用技术； 6. CAD 制图；				

摘　　要

基于现实中转辙机结构及原理，使用 eda 辅助系统进行硬件结构设计。采用 c 语言、verilog 硬件描述语言为范类型单片机、逻辑阵列分别提供提供软、硬逻辑。使用 G 语言作为 visa 通讯界面，提供操作及测试环境。另外针对现有产品组件进行有许可的利用或改进。

关键词：comsol；c；verilog hdl；avr；cpld；labview

目　　录

引　言

由于技术发展及开源运动引发大规模的技术更新换代，同样推动知识产权的进步发展。只有对于新技术的深入研发和积极认可，才能够始终保证技术的先进性。

本设计论文研究实际背景为对于电气结构的模拟，但此设计不局限于仿真，设计的模型具有实际可应用性及相对先进性.

本文参与讨论或直接使用的参考资料均为协议发行资料，故不存在版权纠纷。

本文中引用文献由于其公开特性等原因无任何授权，仅作相关引用。

任何参考形式及文件不代表本人立场。

主要参考资源、引用文献来源方式：Swiss Federal Institute of Intellectual Property、Universität Zürich、git-pip、wiki、sync-ipfs。

对应托管平台需予以履行许可说明的权利及义务。对于损害相关利益和对应知识产权保护法律条款，代表本人意愿的机构或个人有权发起诉讼、维权。详情参见附录 A。

任何未经本人授权的二次开发及发布均视为侵权行为。非盈利性目的可自由使用内容，但应标注来源为本文。

第一章 需求转化及逻辑分析

1.1 单片机控制系统

1.1.1 执行条件框架

通用表述：

(1)进路处于锁闭状态．进路上的道岔应不能转换。

(2)道岔区段有车占用或道岔区段轨道电路发生故障。道岔不能转换。

(3)道岔一经起动．就应转换到底。

(4)道岔起动电路接通后．由于电路故障。道岔没有转动，此时应能自动切断起动电路。

(5)由于某种原因，道岔不能转换到底，应能使道岔操回原位。

(6)道岔转换到底，其起动电路应自动切断。

（7）位置表示

（8）失表报警

其对应需求为：

（1）外部保留-标志 切断电机 优先级：0

（2）外部占用-标志（复用保留-标志） 切断电机 优先级：0

（3）内部转换-标志 切断下级中断程序 优先级：1

（4）外部故障-标志（复用保留-标志） 切断电机 优先级：0

（5）外部操作-标志 调用运动子程序 优先级：3

（6）内部运行-标志 运行运动子程序 优先级：2

（7）内部显示-标志 反馈运行状态 优先级：3

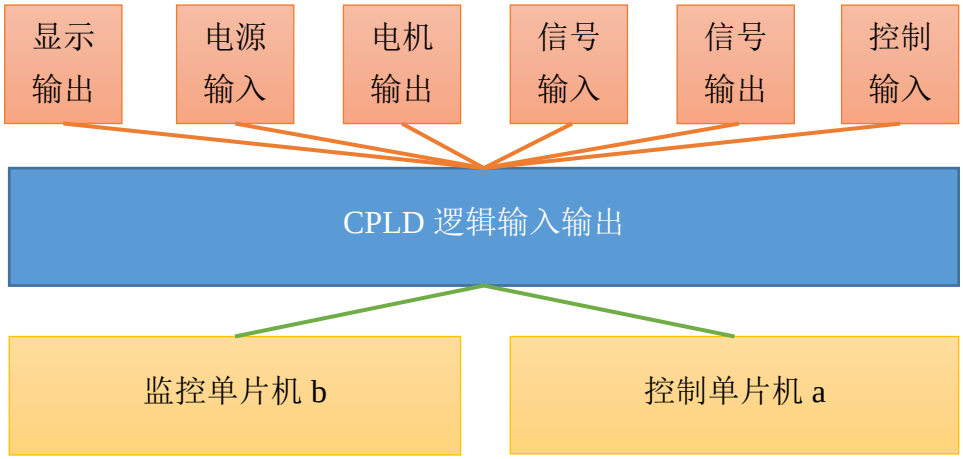
（8）外部显示-标志 反馈运行状态 优先级：不确定

优先级为 0 的标志位脱离单片机执行为独立裁决执行机构，剩余优先级均为内部执行使用，无状态直接输出

因此为两套裁决系统：0 级为 cpld 逻辑系统，剩余为单片机 a

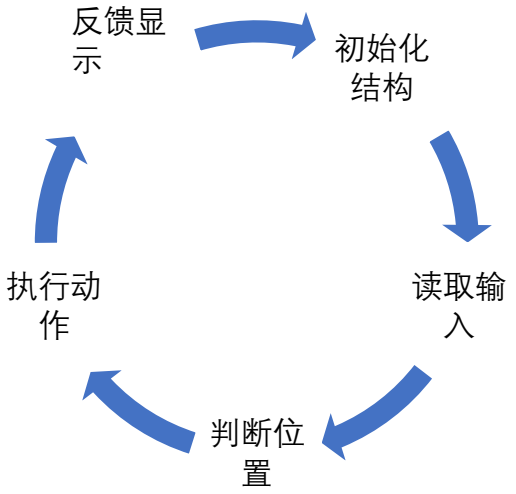
1. 1. 2 关系层次

1.1.1.1 整体硬件关系



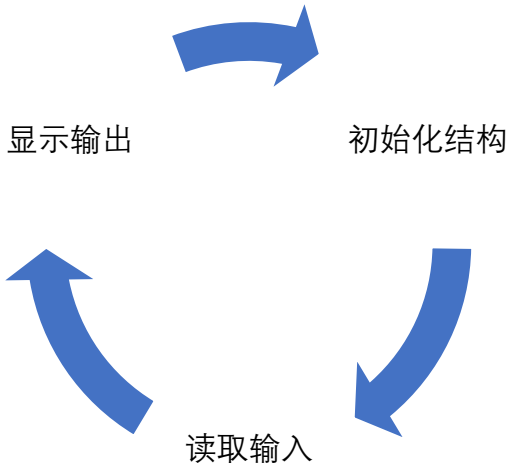
1.1.2.1 单片机 a 程序流程

首先上电后执行固件加载（仅运行一次），然后执行初始化结构，进行循环。外部故障仲裁装置使用 cpld（带步进电机控制器）。



1.1.2.1 单片机 b 程序流程

首先上电后执行固件加载（仅运行一次），然后执行初始化结构，进行循环。
外部故障仲裁装置使用 cpld（带步进电机控制器）。



1. 1. 3 接口器件及通讯可靠性

考虑到此控制系统应用环境恶劣，舍弃常用的快速插接件，（6 脚牛角接口仅为固件调试接口，在实际应用中无与此接口连接的电缆）使用均为螺纹紧固的连接器，其中供电接头使用 sma 螺纹接头，通讯接口采用常见的 d-sub 9 接口（两侧螺丝连接），接线端子使用螺丝挤压式接线端子。该系统接口设计方案大大提高了安全性、免维护性、连接强度。同时电源连接线与远距离串口线使用屏蔽线缆，再次保证其抗干扰性，使其能够在极端条件下正常工作，需要指出的是，由于 rs232 使用绝对电平进行通信，远距离传输时即使是采用屏蔽线也会造成不可避免的数据故障，因此在远距离传输（30m）以上时，通过附加 rs-232 转 rs-485 接头将其转换为相对差分电平延长传送距离，在更远情况（5km）下可考虑使用光端机进行流式传输。上述均为无校验情况下的传输情况，在串口通讯其本身协议支持的范围内，降低通讯波特率，增加校验位也是一种使得无错误传输距离变长可行的方法。以下单片机程序设计中均省略该步，需使用时只需在单片机的 iolib 和上位机的程序参数稍作更改即可。

1.1.4 线路板设计应用可靠性

对于该线路板首先考虑的问题是其机械强度，使用玻纤复合电路板（8 层玻璃纤维纺织层以上）可以大大提高线路板的抗老化性、以及柔韧度（相比于环氧树脂材料以及电木复合材料而言）避免热疲劳产生的焊盘脱落等严重故障，若应用于普通非酸性氛围中，其寿命是十分可观的。相比陶瓷线路基板，其制造周期短，成本低，韧性高，但缺点是抗腐蚀性明显不足。综合考虑现场环境的条件，故选用玻纤板作为线路基板。

其次应该考虑的是其故障的直观显现，也就是线路板在出现异常损坏时能够及时发现更换，故选用白色作为表面绝缘涂层颜色，在发生机械损伤（划痕，裂纹）、热损伤（敷铜线路熔断）时其直观的表现能够与正常情况加以区分以替换。

再其次是通过对电路板的铺铜可以提高其抗干扰性，同时对铺铜层进行网格交错（首层 90°，次层 45°）能够在保证其抗干扰性的同时避免热膨胀引起的表面铺铜脱落

1.1.5 单片机选择

单片机是整个控制器的核心，他的可靠性在整体中起着主导作用，选择一个能够在极端环境下正常使用的单片机能够极大提高整体的可靠程度，相比复杂指令集的传统型单片机而言，精简指令集单片机能够以更低的热功耗，更高的效能运行。在相同的价格下其优势更加明显，且大部分精简指令集内部设备丰富，能够显著降低开发周期和难度，同时发生错误的原因及情况更容易预料。值得一提的是，对于像转辙机这样的设备，对于芯片的熔丝只读保护和防止读取程序能够进一步提供可靠性和安全性。综合市场行情及出厂数据分析，atmega 系列相比 at89 系列更适合用于该种场合。在本论文中选用是市场价格最低的单片机 atmega8 进行设计，同样对于该系列中编号不同的单片机仅需采取不同的 runtime 即可程序完全移植。Atmega8 中分为多种芯片封装形式，本论文中采取-pu 封装，采用双列直插通孔连接，综合耐候性强，焊接方便。对于特殊应用，同时提出 avr 芯片硬件架构设计方案（verilog），其可与大部分 atmega avr 架构原生芯片相兼容。

1.1.6 其他原件参数选取

除上述连接器、单片机外，周围原件皆使用过孔安装式，增加其机械强度，可靠性高。电容为整个电路中最脆弱的环节，因此并未使用易老化或损坏、耐候性差的电解电容。本设计中使用为独石电容（多层瓷片），相比电解电容几乎没有寿命和温度的限制，且不存在固定的极性，降低故障的发生率。对于有功率容量型器件，为提高可靠性，可使用同等参数，功率容量更大的器件，避免器件表面过热引起的故障。

1.2cpld 硬逻辑替换方案

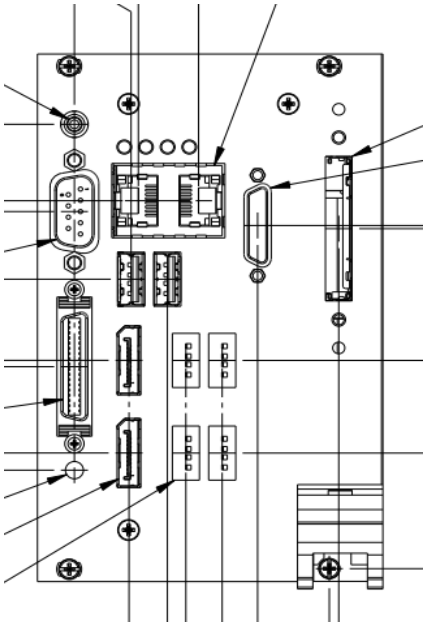
由于考虑到单片机内代码需在内置 rom 中解释运行，仿真可能会有所局限，且 cpld 可编程逻辑阵列基于硬件逻辑门的架构，意外错误率极低，且可自行扩充指令集

1.3 工业 cpci/pxi 控制系统

对于上位机，选择具有多种通信方式的 pxi 器件是最佳选择，在一个 3u 高度的 pxi 机箱里，允许使用多热插拔通讯板、电源模块和可冗余、实时同步的控制器。Pxi 的最大特点是在 cpci 的基础上增加了高精度时钟通道，使得所有器件均可实时同步，且配备控制器机箱级计量及通信总线，可直接使用 pc 接管该机箱上的全部器件。该特种计算机的使用，可显著提高稳定性和可靠性。

机器框架内可添加额外的扩展板卡以扩充通讯端口的数量，同时每次对于硬件板卡的损坏、移动、替换，系统内有对应日志系统进行报警和记录，用以故障排除和定期检视。

下图为对于现有 pxie 控制器其具有的接口：lan（用于实现令牌环局域网网）、e-sata（外部可替换式大容量磁盘接口）、gpib IEEE-488（通用仪器复用总线）、lpt IEEE-1284（基本并行接口）、serial rs-232/rs-485（可调模式的通用串行总线）、ttl 4pin、usb3.0、cfexpress（新一代可替换式高速电子盘）。可以满足绝大多数场景的应用。操作系统选用的是 ni-labview-rt 系统，软实时和硬实时的结合使得该种控制系统结构能够最大限度的保证安全性。

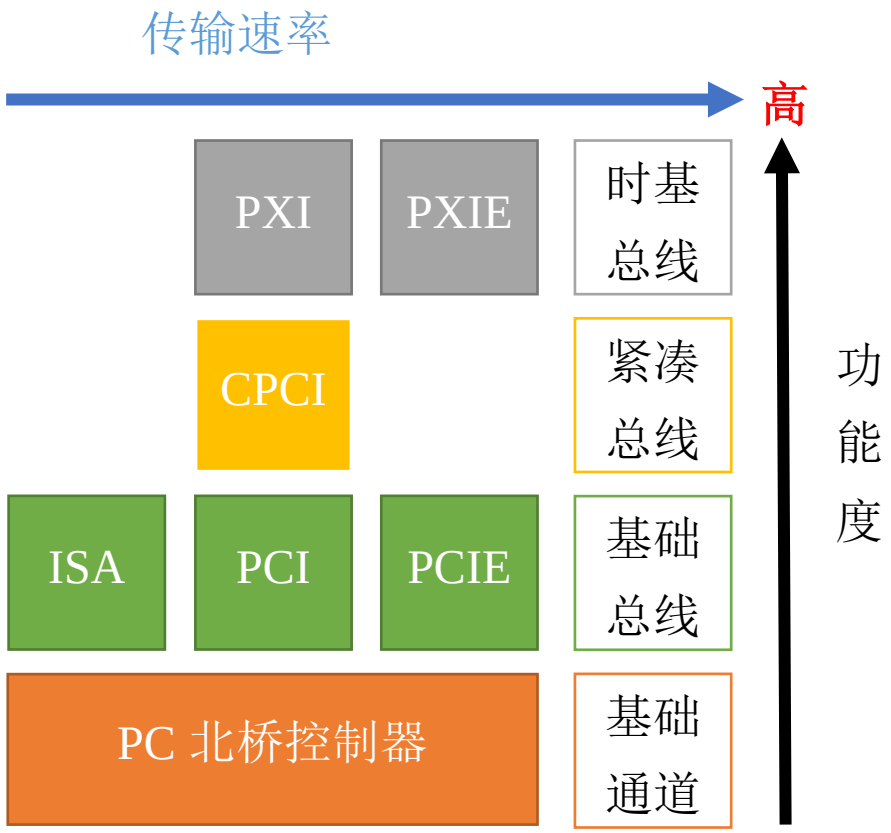


本设计中仅涉及上位机软件及电源硬件（psu）的设计，其他硬件部分可能涉及 ni 的专利，因知识产权故不作研究，专利详情见附录 B

1.3.1labview 上位机软件方案

1.3.2psu 电源设计方案

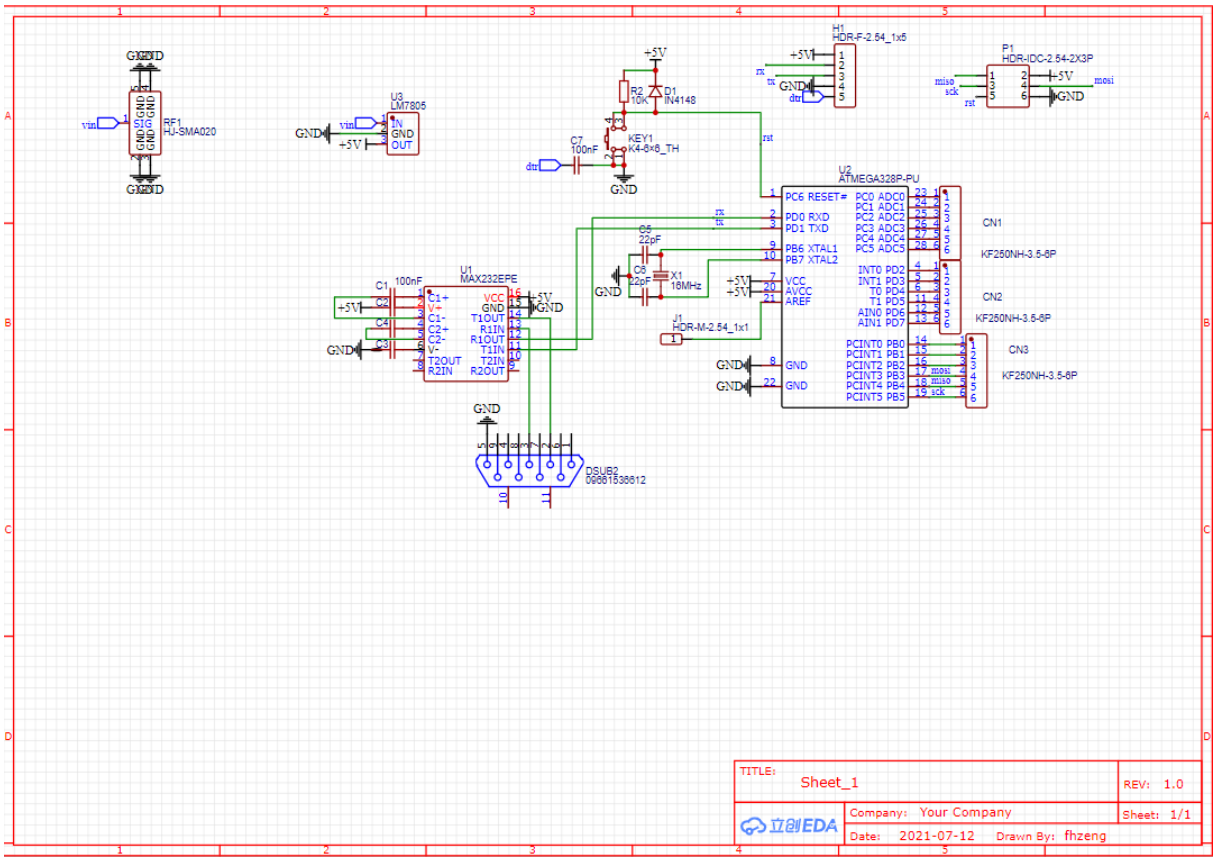
Cpci 与 pxi 等目前常用的设备总线的关系如下



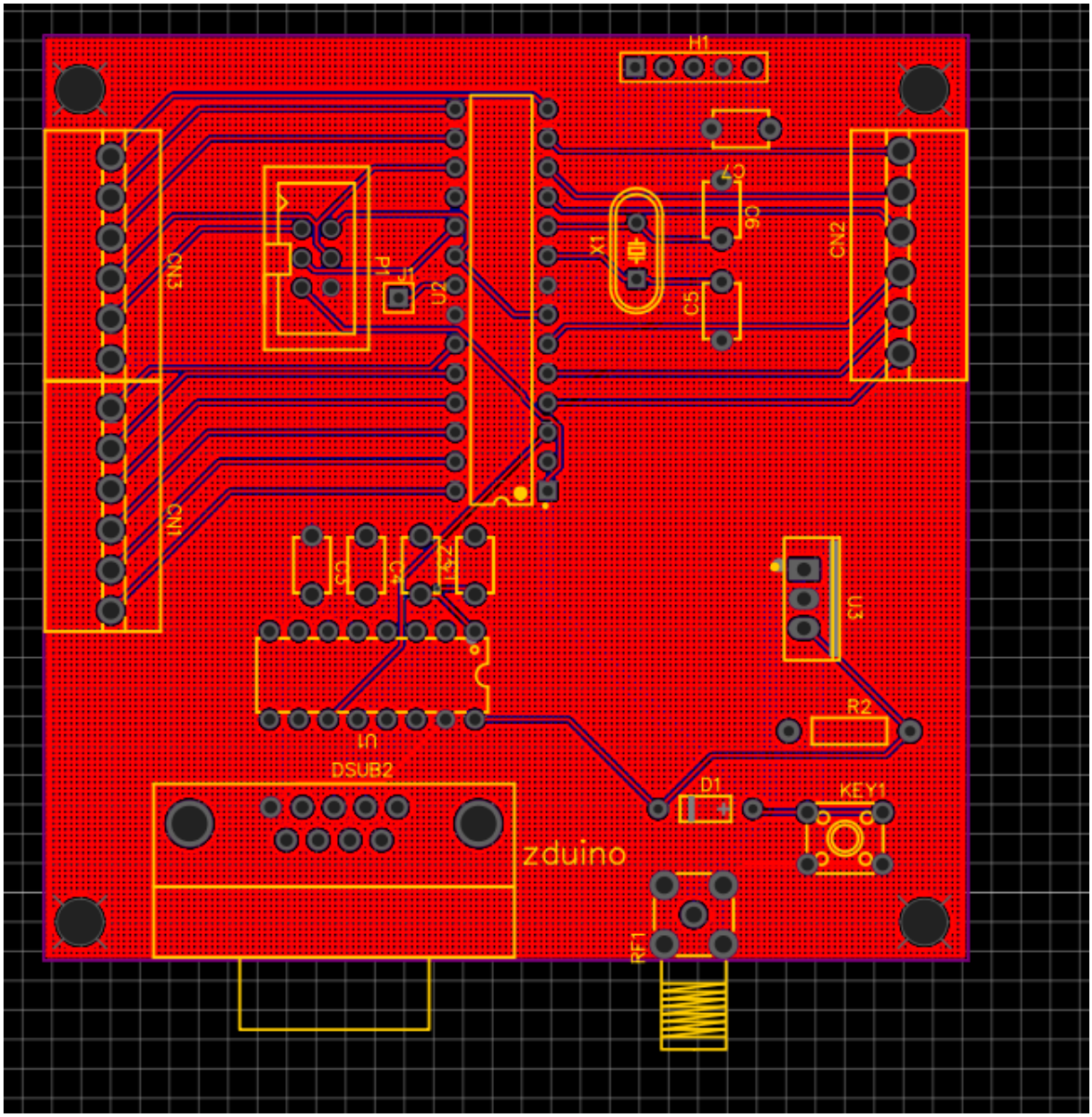
第二章 硬件部分设计

2.1 通用工业 avr 控制板设计

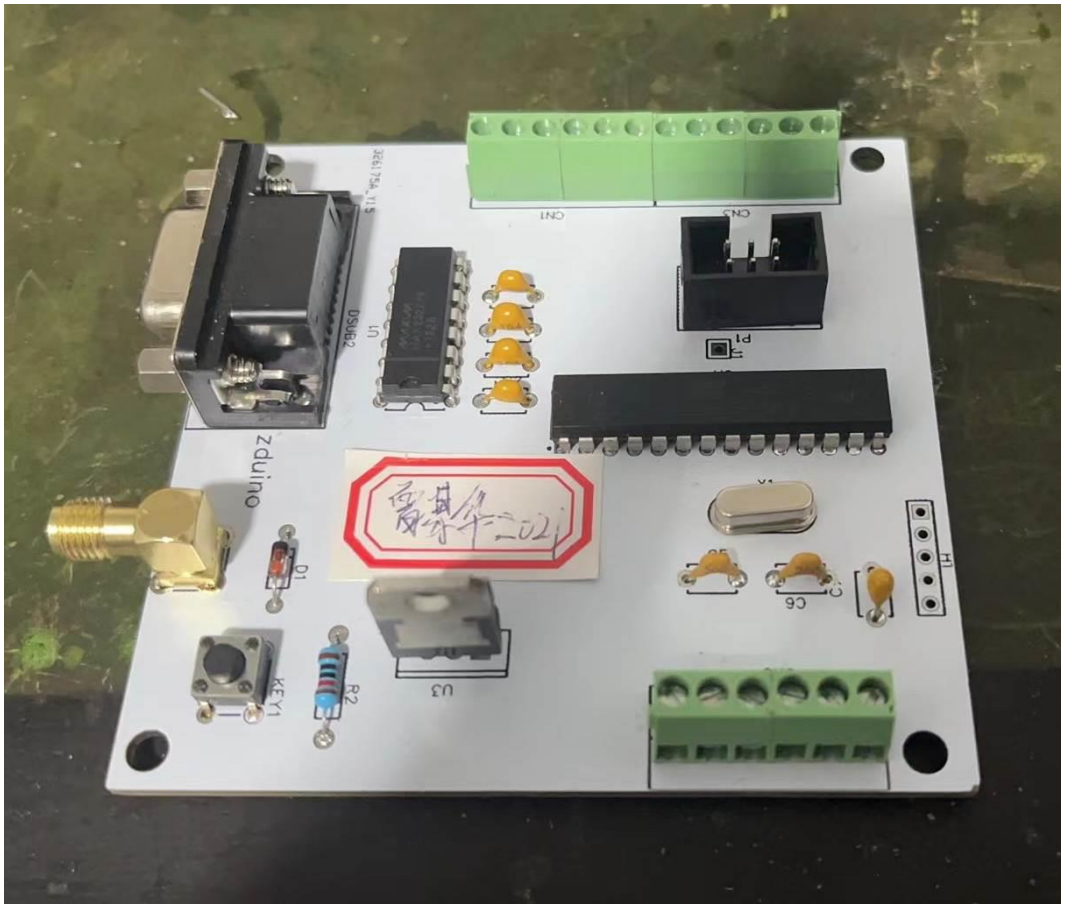
2.1.1 电路原理图



2.1.2 布线示例



2. 1. 3 实物外形



2.2 替代性芯片内核架构设计方案（verilog）

```
//算数单元
`define is_alu          5'h00
// 挂钩
`define is_inc          5'h01
`define is_dec          5'h02
`define is_com          5'h03
`define is_adiw_or_sbiw  5'h04
`define is_movw         5'h05
`define is_clx_or_sex 5'h06
`define is_mulu         5'h07

// 内存
`define is_out          5'h08
```

```
`define is_in       5'h09
`define is_lds      5'h0a
`define is_ld_xyz5'h0b
`define is_ld_yz_plus_q   5'h0c
`define is_lpm      5'h0d
`define is_pop      5'h0e
`define is_push     5'h0f

// 流控
`define is_ret      5'h10
`define is_cpse     5'h11
`define is_sbrc_or_sbrs    5'h12
`define is_brbc_or_brbs   5'h13
`define is_jmp      5'h14
`define is_call     5'h15
`define is_ijmp     5'h16
`define is_rjmp     5'h17
`define is_rcall    5'h18
`define is_sbis_or_sbic    5'h19
`define is_bld_or_bst 5'h1a
//指令
`define OP_MOVE   4'h0
`define OP_MOVR   4'h1
`define OP_ADD    4'h2
`define OP_SUB 4'h3
`define OP_ADW    4'h4
`define OP_SBW    4'h5 // OP_ADW | 1
`define OP_NEG4'h6
`define OP_SWAP   4'h7
`define OP_ASR 4'h8
`define OP_ROR4'h9 // OP_ASR | 1
`define OP_LSR  4'hA
`define OP_AND    4'hB
`define OP_EOR 4'hC
`define OP_OR   4'hD
`define OP_SREG   4'hE
```

```
`define OP_MUL    4'hF
//方法
//`define CONFIG_SREG_SI
//`define CONFIG_SREG_ST
//`define CONFIG_SREG_SH
`define CONFIG_SREG_SS
`define CONFIG_SREG_SV
`define CONFIG_SREG_SN
`define CONFIG_SREG_SZ
`define CONFIG_SREG_SC



`define CONFIG_OP_ADD
`define CONFIG_OP_SUB
`define CONFIG_OP_ADW_OR_SBW
`define CONFIG_OP_SBW
`define CONFIG_OP_NEG
`define CONFIG_OP_SWAP
`define CONFIG_OP_ASR_OR_ROR
`define CONFIG_OP_LSR
`define CONFIG_OP_AND
`define CONFIG_OP_EOR
`define CONFIG_OP_OR
`define CONFIG_OP_SREG
//`define CONFIG_OP_MUL

//////////////////////////算术单元
module avalu(
    input clk,
    input reset,

    input [15:0] Rd_in,
    input [7:0] Rr_in,
    input [7:0] sreg_in,
    input [3:0] op,
```

```
    input use_carry,
    input [7:0] keep_sreg,

    output [15:0] R_out,
    output [7:0] sreg_out
);
    reg [7:0] R;
    reg [7:0] Rh;
    assign R_out = { Rh, R };

    wire [7:0] Rr = Rr_in;
    wire [7:0] Rd = Rd_in[7:0];
    reg SI, ST, SH, SS, SV, SN, SZ, SC;
    assign sreg_out = {

//发送模块
module uart_tx(
    input clk,
    input reset,
    input baud_x1,
    output serial,
    output reg ready,
    input [7:0] data,
    input data_strobe
);


    reg [7+1+1:0]     shiftreg;


    reg               serial_r;
    assign           serial = !serial_r;

    //assign           ready = (shiftreg == 0);

    /*
```

```
 * 状态机
 */

always @(posedge clk)
  if (reset) begin
      shiftreg <= 0;
      serial_r <= 0;
  end
  else if (data_strobe) begin
$display("UART TX %02x", data);
      shiftreg <= {
 1'b1, //停止位
 data,
 1'b0    // start bit (inverted)
};
ready <= 0;
  end
  else if (baud_x1) begin
      if (shiftreg == 0)
begin
        /* 线程优先 */
        serial_r <= 0;
   ready <= 1;
end else
        serial_r <= !shiftreg[0];
//  位移
      shiftreg <= {1'b0, shiftreg[7+1+1:1]};
 end else
     ready <= (shiftreg == 0);

endmodule


//////////////////////接收模块
module uart_rx(
```

```
    input clk,
    input reset,
    input baud_x4,
    input serial,
    output [7:0] data,
    output data_strobe
);

    wire            serial_sync;
    d_flipflop_pair input_dff(clk, reset, serial, serial_sync);

    /*
     *状态机
     */
    reg [8:0]       shiftreg;
    reg [5:0]       state;
    reg             data_strobe;
    wire [3:0]      bit_count = state[5:2];
    wire [1:0]      bit_phase = state[1:0];

    wire            sampling_phase = (bit_phase == 1);
    wire            start_bit = (bit_count == 0 && sampling_phase);
    wire            stop_bit = (bit_count == 9 && sampling_phase);

    wire            waiting_for_start = (state == 0 && serial_sync == 1);

    wire            error = ( (start_bit && serial_sync == 1) ||
                              (stop_bit && serial_sync == 0) );

    assign          data = shiftreg[7:0];

    always @(posedge clk or posedge reset)
      if (reset) begin
          state <= 0;
          data_strobe <= 0;
      end
```

```
        else if (baud_x4) begin

            if (waiting_for_start || error || stop_bit)
               state <= 0;
            else
               state <= state + 1;

            if (bit_phase == 1)
               shiftreg <= { serial_sync, shiftreg[8:1] };

            data_strobe <= stop_bit && !error;

        end
        else begin
            data_strobe <= 0;
        end

endmodule

//////////////////////串口
module uart(
    input clk,
    input reset,
    // 逻辑
    input [7:0] baud_div,
    input tx_strobe,
    input [7:0] tx_data,
    output tx_ready,
    output rx_ready,
    // 物理
    input rx_in,
    output tx_out
);
    reg [7:0] counter;
    reg baud_x1;
```

```
    always @(posedge clk)
    begin
     if (counter == 0) begin
          baud_x1 <= 1;
          counter <= baud_div;
     end else begin
          baud_x1 <= 0;
          counter <= counter - 1;
     end
    end

    uart_tx tx(
     .clk(clk),
     .reset(reset),
     .baud_x1(baud_x1),
     .ready(tx_ready),
     .data(tx_data),
     .data_strobe(tx_strobe),
     .serial(tx_out)
    );
endmodule

//////////////////////片上系统
module avsoc(
    input clk,
    input reset,

    output [7:0] port_b,
    input [7:0] pin_b,
    output [7:0] ddr_b,
    output serial_tx,
    input serial_rx
);
    localparam CODEBITS = 12;

    //程序存储器
```

```
reg [15:0] code[0:(1 << CODEBITS) - 1];
reg [15:0] cdata;
wire [15:0] pc;

always @(posedge clk)
 cdata <= code[pc[CODEBITS-1:0]];


// 数据存储器
wire [15:0] addr;
wire [7:0] ram_data;
wire [7:0] wdata;
wire wen;
wire ren;

avram ram(
 .clk(clk),
 .wen(wen),
 .addr(addr),
 .wdata(wdata),
 .rdata(ram_data)
);

// 输入输出直接映射
wire [6:0] io_addr = addr[6:0];
wire io_sel = addr[15:7] == 0;
wire io_ren = io_sel & ren;
wire io_wen = io_sel & wen;

// 串口
wire [7:0] uart_data;
wire uart_valid;
avuart uart(
 .clk(clk),
 .reset(reset),
 //逻辑
```

```
 .addr(io_addr),
 .ren(io_ren),
 .wen(io_wen),
 .wdata(wdata),
 .rdata(uart_data),
 .valid(uart_valid),
 // 物理
 .tx_out(serial_tx)
);

// 定时器
wire [7:0] tcnt1_data;
wire tcnt1_valid;
avtimer #(.BASE(7'h3F)) tcnt1(
 .clk(clk),
 .reset(reset),
 // 逻辑
 .addr(io_addr),
 .ren(io_ren),
 .wen(io_wen),
 .wdata(wdata),
 .rdata(tcnt1_data),
 .valid(tcnt1_valid)
);

// 原 avr b 口
wire [7:0] portb_data;
wire portb_valid;
avgpio #(.BASE(7'h36)) portb_dev(
 .clk(clk),
 .reset(reset),
 // 逻辑
 .addr(io_addr),
 .ren(io_ren),
 .wen(io_wen),
```

```
 .wdata(wdata),
 .rdata(portb_data),
 .valid(portb_valid),
 // 物理
 .port(port_b),
 .pin(pin_b),
 .ddr(ddr_b)
);

// 内存响应
reg [7:0] rdata;
always @(*) begin
 rdata = ram_data;
 if (uart_valid) rdata = uart_data;
 if (tcnt1_valid) rdata = tcnt1_data;
 if (portb_valid) rdata = portb_data;
end

always @(posedge clk)
 if (wen) $display("WR %04x <= %02x", addr, wdata);

avcore core(
 .clk(clk),
 .reset(reset),

 //程序存储器
 .pc(pc),
 .cdata(cdata),

 // 数据存储器、读写
 .data_addr(addr),
 .data_wen(wen),
 .data_ren(ren),
 .data_read(rdata),
 .data_write(wdata)
);
```

```
endmodule
////////////////寄存器
module avregs(
    input clk,
    input reset,

    // 读端口
    input [5:0] a,
    input [5:0] b,
    output [15:0] Ra,
    output [7:0] Rb,

    // 写端口
    input write,
    input write_word,
    input [5:0] d,
    input [15:0] Rd
);
    // 内存虚拟化
    reg [15:0] ram_a[0:15];
    reg [15:0] ram_b[0:15];

    initial $readmemh("zero.hex", ram_a);//用现有文件映射
    initial $readmemh("zero.hex", ram_b);

    wire [4:0] a_word = a[5:1];
    wire [4:0] b_word = b[5:1];
    wire [4:0] d_word = d[5:1];

    reg [1:0] al_src;
    reg [1:0] ah_src;
    reg [1:0] bl_src;

    reg [15:0] Ra_ram;
```

```
reg [15:0] Rb_ram;

reg [15:0] Ra;
reg [ 7:0] Rb;
reg [15:0] cache_Rd;

always @(posedge clk) if (!reset) begin
 Ra_ram <= ram_a[a_word];
 Rb_ram <= ram_b[b_word];

 al_src <= { 1'b0, a[0] };
 ah_src <= { 1'b0, 1'b1 };
 bl_src <= { 1'b0, b[0] };

 if (write) begin
      cache_Rd <= Rd;

      if (write_word) begin

           ram_a[d_word][15:0] <= Rd;
           ram_b[d_word][15:0] <= Rd;


           if (a_word == d_word) begin
                if (a[0] == 0) begin

                     al_src <= 2'b10;
                     ah_src <= 2'b11;
                end else
                if (a[0] == 1) begin

                     al_src <= 2'b11;
                end
           end

           if (b_word == d_word) begin
```

```
            if (b[0] == 0) begin

                bl_src <= 2'b10;
            end else begin

                bl_src <= 2'b11;
            end
        end
    end else
    if (d[0] == 0) begin

        //$display("R[%dL] <= %02x", d_word, Rd[7:0]);
        ram_a[d_word][7:0] <= Rd[7:0];
        ram_b[d_word][7:0] <= Rd[7:0];

        if (a_word == d_word && a[0] == 0) begin

            al_src <= 2'b10;
        end

        if (b == d) begin

            bl_src <= 2'b10;
        end
    end else begin

        //$display("R[%dH] <= %02x", d_word, Rd[7:0]);
        ram_a[d_word][15:8] <= Rd[7:0];
        ram_b[d_word][15:8] <= Rd[7:0];

        if (a_word == d_word) begin
            if (a[0] == 0) begin

                ah_src <= 2'b10;
            end else begin
```

```
                    al_src <= 2'b10;
                end
            end

            if (b == d) begin

                bl_src <= 2'b10;
            end
        end
    end
    end

    // 缓冲区
    always @(*) begin
    case(al_src)
    2'b00: Ra[7:0] = Ra_ram[7:0];
    2'b01: Ra[7:0] = Ra_ram[15:8];
    2'b10: Ra[7:0] = cache_Rd[7:0];
    2'b11: Ra[7:0] = cache_Rd[15:8];
    endcase

    case(ah_src)
    2'b00: Ra[15:8] = Ra_ram[7:0];
    2'b01: Ra[15:8] = Ra_ram[15:8];
    2'b10: Ra[15:8] = cache_Rd[7:0];
    2'b11: Ra[15:8] = cache_Rd[15:8];
    endcase

    case(bl_src)
    2'b00: Rb[7:0] = Rb_ram[7:0];
    2'b01: Rb[7:0] = Rb_ram[15:8];
    2'b10: Rb[7:0] = cache_Rd[7:0];
    2'b11: Rb[7:0] = cache_Rd[15:8];
    endcase
    end
endmodule
```

```
/////内存
module avram(
    input clk,
    input wen,
    input [15:0] addr,
    input [7:0] wdata,
    output [7:0] rdata
);
    reg [7:0] ram[0:65535];
    reg [7:0] rdata;

    always @(posedge clk)
    begin
     rdata <= ram[addr];
     if (wen)
          ram[addr] <= wdata;
    end

endmodule
///////////////硬件模块
module avuart (
    input clk,
    input reset,
    // IO  总线
    input ren,
    input wen,
    input [6:0] addr,
    input [7:0] wdata,
    output reg [7:0] rdata,
    output reg valid,
    //  物理结构
    input rx_in,
    output tx_out
);
    parameter BASE = 7'h2D;
    reg [7:0] uart_baud_div = `UART_DIV;
```

```
    reg [7:0] uart_tx_data;
    reg uart_tx_strobe;
    wire uart_tx_ready;

    wire [7:0] uart_status_reg = { 7'b0000000, uart_tx_ready };

    uart uart(
     .clk(clk),
     .reset(reset),
     .baud_div(uart_baud_div),
     .tx_strobe(uart_tx_strobe),
     .tx_data(uart_tx_data),
     .tx_ready(uart_tx_ready),
     .tx_out(tx_out)
     //.rx_strobe(uart_rx_
     //.rx_data(usidr
    );

    always @(posedge clk) begin
     uart_tx_strobe <= 0;
     valid <= 0;

     if (wen) case(addr)
     BASE + 0: uart_baud_div <= wdata;
     BASE + 2: { uart_tx_data, uart_tx_strobe } <= { wdata, 1'b1 };
     endcase

     if (ren) case(addr)
     BASE + 0: { rdata, valid } <= { uart_baud_div, 1'b1 };
     BASE + 1: { rdata, valid } <= { uart_status_reg, 1'b1 };
     endcase
    end
endmodule
///////定时器
Module avtimer(
    input clk,
```

```
    input reset,
    // IO bus
    input ren,
    input wen,
    input [6:0] addr,
    input [7:0] wdata,
    output reg [7:0] rdata,
    output reg valid
);
    parameter BASE = 7'h4F;

    // 时钟速度
    reg [7:0] tcnt1 = 8'h55;

    always @(posedge clk) begin
     valid <= 0;
     tcnt1 <= tcnt1 + 1;

     if (wen) case(addr)
     BASE + 0: tcnt1 <= wdata;
     endcase

     if (ren) case(addr)
     BASE + 0: { rdata, valid } <= { tcnt1, 1'b1 };
     endcase
    end
endmodule
//////////控制输入、输出
module avgpio(
    input clk,
    input reset,
    // IO 总线
    input ren,
    input wen,
    input [6:0] addr,
    input [7:0] wdata,
```

```
    output reg [7:0] rdata,
    output reg valid,
    // 物理结构
    output [7:0] port,
    input [7:0] pin,
    output [7:0] ddr
);
    parameter BASE = 7'h36;
    reg [7:0] port;
    reg [7:0] ddr;

    always @(posedge clk) begin
     if (reset) begin
          port <= 0;
          ddr <= 0;
     end

     valid <= 0;

     if (wen) case(addr)
     //BASE + 0: pin <= wdata;
     BASE + 1: ddr <= wdata;
     BASE + 2: begin
          $display("PORT %02x", wdata);
          port <= wdata;
     end
     endcase

     if (ren) case(addr)
     BASE + 0: { rdata, valid } <= { pin, 1'b1 };
     BASE + 1: { rdata, valid } <= { ddr, 1'b1 };
     BASE + 2: { rdata, valid } <= { port, 1'b1 };
     endcase
    end
endmodule
```

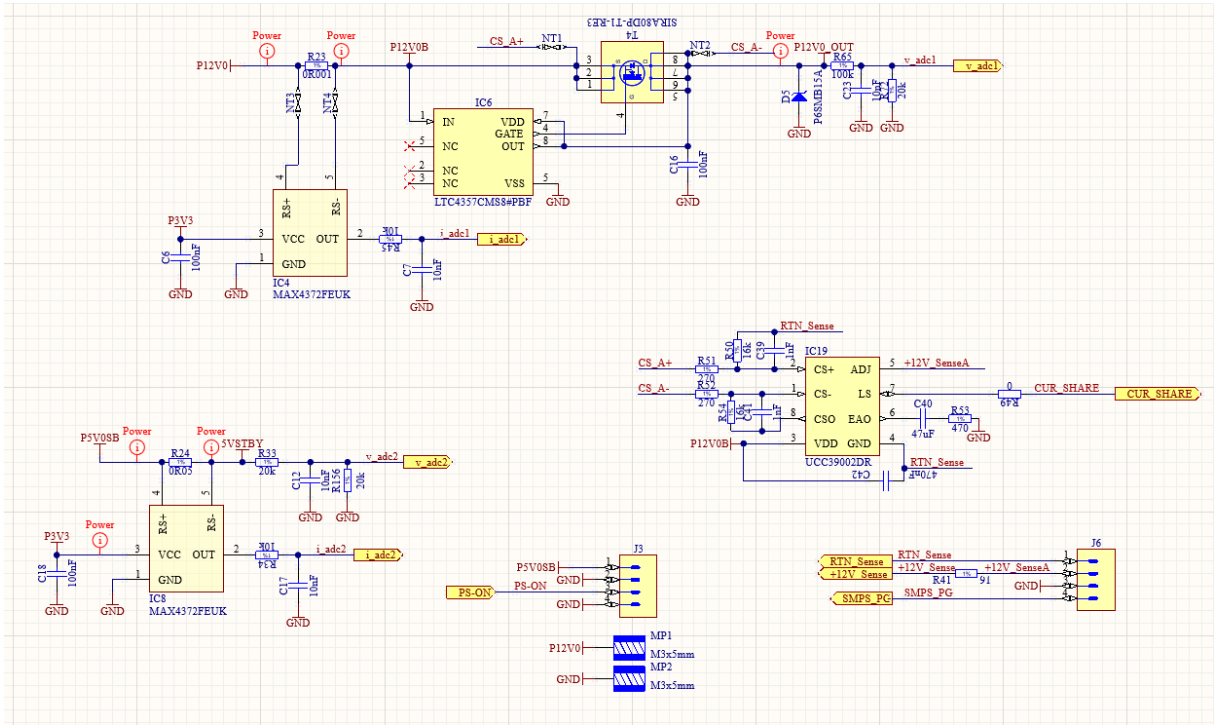
```
//////////////////////////汇编解释器
module avinstr(
    input [15:0] opcode,
    output [4:0] instr,
    output [3:0] alu_op,
    output alu_rdi,
    output alu_store,
    output alu_carry
);
    // 寄存器
    wire [5:0] op_Rd = opcode[8:4]; // 0-31

    /* 结构解码 */
    reg [4:0] instr;
    reg [3:0] alu_op;
    reg alu_store;
    reg alu_carry;
    reg alu_rdi;

`define ALU_OP(op, store, carry) \
    begin \
     alu_op = op; \
     alu_store = store; \
     alu_carry = carry; \
    end

`define ALU_OP_RDI(op, store, carry) \
    begin \
     alu_op = op; \
     alu_store = store; \
     alu_carry = carry; \
     alu_rdi = 1; \
    end


    always @(*) begin
```

```
instr = `is_alu;
alu_op = 0;
alu_store = 0;
alu_carry = 0;
alu_rdi = 0;
casez({opcode[15:9],opcode[3:0]})
11'b0000_000_????: if (opcode[8] == 1'b1) instr = `is_movw; // else NOP
11'b0000_01?_????: `ALU_OP(`OP_SUB, 0, 1) // CPC Rd,Rr
11'b0000_10?_????: `ALU_OP(`OP_SUB, 1, 1) // SBC Rd, Rr
11'b0000_11?_????: `ALU_OP(`OP_ADD, 1, 0) // ADD Rd, Rd
11'b0001_00?_????: instr = `is_cpse;
11'b0001_01?_????: `ALU_OP(`OP_SUB, 0, 0) // CP Rd,Rr
11'b0001_10?_????: `ALU_OP(`OP_SUB, 1, 0) // SUB Rd, Rr
11'b0001_11?_????: `ALU_OP(`OP_ADD, 1, 1) // ADC Rd, Rr
11'b0010_00?_????: `ALU_OP(`OP_AND, 1, 0) // AND Rd, Rr
11'b0010_01?_????: `ALU_OP(`OP_EOR, 1, 0) // EOR Rd, Rr
11'b0010_10?_????: `ALU_OP(`OP_OR,    1, 0) // OR Rd, Rr
11'b0010_11?_????: `ALU_OP(`OP_MOVR, 1, 0) // MOV Rd, Rr
11'b0011_???_????: `ALU_OP_RDI(`OP_SUB, 0, 0) // CPI Rdi, K
11'b0100_???_????: `ALU_OP_RDI(`OP_SUB, 1, 1) // SBCI Rdi, K
11'b0101_???_????: `ALU_OP_RDI(`OP_SUB, 1, 0) // SUBI Rdi, K
11'b0110_???_????: `ALU_OP_RDI(`OP_OR, 1, 0) // ORI Rdi, K
11'b0111_???_????: `ALU_OP_RDI(`OP_AND, 1, 0) // ANDI Rdi, K
11'b1001_00?_0000: instr = `is_lds;
11'b1001_000_010?: instr = `is_lpm; // Z
11'b1000_00?_0000: instr = `is_ld_xyz; // z
11'b1000_00?_1000: instr = `is_ld_xyz; // Y
11'b1001_00?_1100: instr = `is_ld_xyz; // X
11'b1001_00?_0001: instr = `is_ld_xyz; // Z+
11'b1001_00?_0010: instr = `is_ld_xyz; // -Z
11'b1001_00?_1001: instr = `is_ld_xyz; // Y+
11'b1001_00?_1010: instr = `is_ld_xyz; // -Y
11'b1001_00?_1101: instr = `is_ld_xyz; // X+
11'b1001_00?_1110: instr = `is_ld_xyz; // -X
11'b10?0_???_????: instr = `is_ld_yz_plus_q;
11'b1001_000_1111: instr = `is_pop;
```

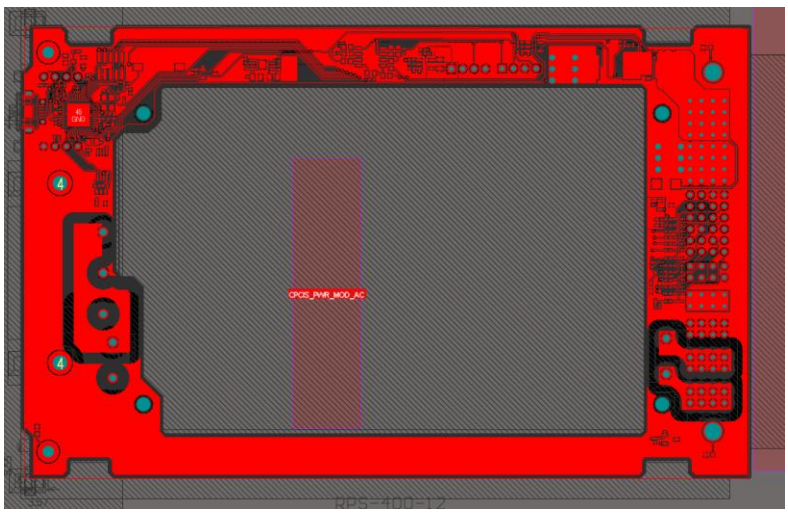
```
    11'b1001_001_1111: instr = `is_push;
    11'b1001_010_0000: instr = `is_com;
    11'b1001_010_0001: `ALU_OP(`OP_NEG, 1, 0) // NEG
    11'b1001_010_0010: `ALU_OP(`OP_SWAP, 1, 0) // SWAP
    11'b1001_010_0011: instr = `is_inc; // INC
    //11'b1001_010?_0100: instr = `is_nop; //
    11'b1001_010_0101: `ALU_OP(`OP_ASR, 1, 0) // ASR
    11'b1001_010_0110: `ALU_OP(`OP_LSR, 1, 0) // LSR
    11'b1001_010_0111: `ALU_OP(`OP_ROR, 1, 0) // ROR
    11'b1001_010_1000: begin
        casez(opcode[8:4])
        5'b0????: instr = `is_clx_or_sex;
        5'b10000: instr = `is_ret;
        5'b11100: instr = `is_lpm;
        endcase
    end
    11'b1001_010_1001: instr = `is_ijmp;
    11'b1001_010_1010: instr = `is_dec; // DEC Rd
    11'b1001_010_110?: instr = `is_jmp;
    11'b1001_010_1111: instr = `is_call;
    11'b1001_011_????: instr = `is_adiw_or_sbiw;
    11'b1001_11?_????: instr = `is_mulu;
    11'b1001_10?_????: instr = `is_sbis_or_sbic;
    11'b1011_0??_????: instr = `is_in;
    11'b1011_1??_????: instr = `is_out;
    11'b1100_???_????: instr = `is_rjmp;
    11'b1101_???_????: instr = `is_rcall;
    11'b1110_???_????: `ALU_OP_RDI(`OP_MOVR, 1, 0)
    11'b1111_0??_????: instr = `is_brbc_or_brbs;
    11'b1111_10?_0???: instr = `is_bld_or_bst;
    11'b1111_11?_0???: instr = `is_sbrc_or_sbrs;
    endcase
   end
endmodule
```

2. 3cpci 兼容 psu 设计

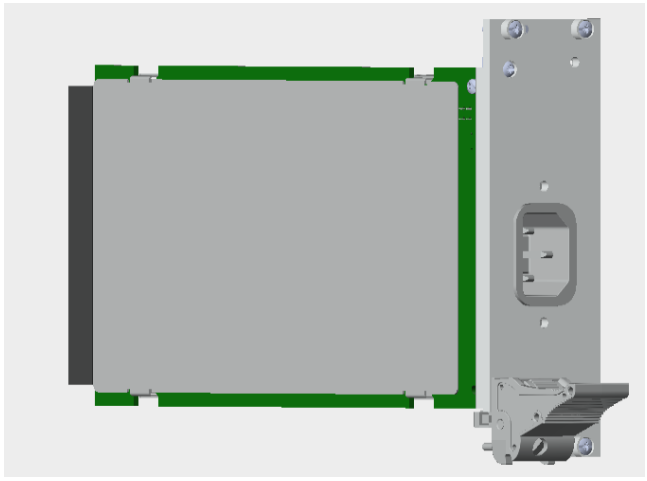
2. 3. 1 电路原理图

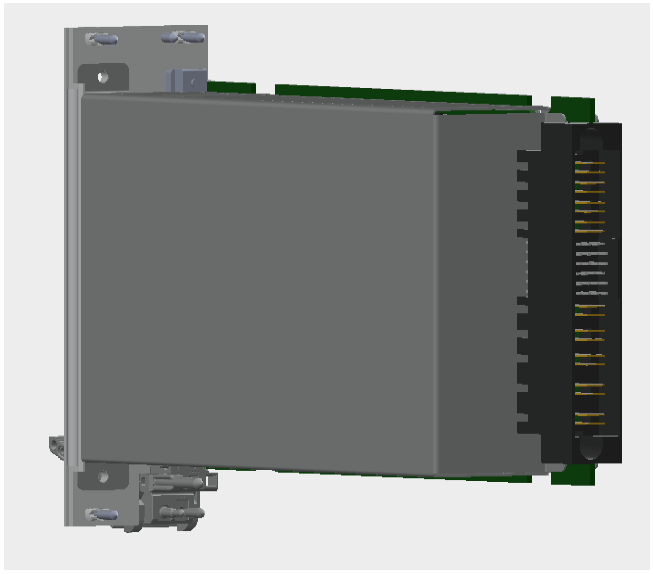


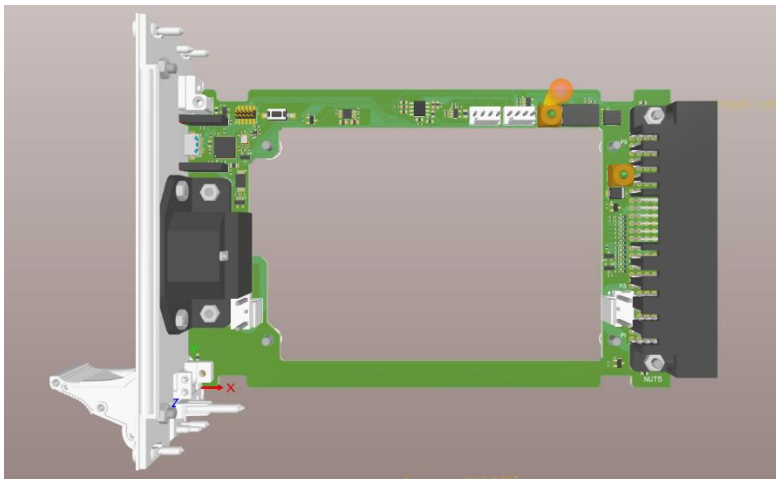
2. 3. 2 布线示例



2.3.4 仿真外形图







第三章 软件部分设计

3.1 单片机状态机程序

3.1.1 单片机 a 代码（c）

```
Run_once(){//单次运行
Visa.begin(9600);//定义 visa 串口资源波特率 9600
pm(zl,INPUT_PULLUP);//定义定位状态逻辑输入内部上拉
pm(zr,INPUT_PULLUP); //定义反位状态逻辑输入内部上拉
pm(zt,INPUT_PULLUP);//定义挤岔急停逻辑输入内部上拉
pm(zp,OUTPUT); //定义脉冲逻辑输出
pm(zd,OUTPUT);//定义方向逻辑输出
}

Run_always(){
////////////////////////////////////////////////////////////////////////////////
comdata= Visa.parseInt();//即时获取一个串口输入数字

if(zs==3&&dr(zl)==0){ //判断状态步标志 zs 以及定位标志
comdata=2;//紧急停止命令
zs=0;//状态清零
}
if(zs==1&&dr(zr)==0){
comdata=2;
zs=0;
}
if(zt==0)//挤岔急停
comdata=2;
////////////////////////////////////////////////////////////////////////////////
switch (comdata) {//命令执行选择
```

```
case 3:

aw(zp,128);
dw(zd,1);
zs=3;
break;


case 1:

aw(zp,128);
dw(zd,0);
zs=1;
break;


case 2:
dw(zp,0);
zs=2;
break;
}
Visa.print(zs);//返回道岔状态
}
```

3.1.2 单片机 b 代码（c）

```
Run_once(){
Visa.begin(9600);
}
Run_always(){
Visa.print(ar(0)+1000);//读取并转换电流值，+1000 避免字长变化
Delay(20);
}
```

3.2 avr studio 单片机 runtime

3.2.1 io lib（c）

```
#ifndef bio_h
#define bio_h

#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <math.h>

#include <avr/pgmspace.h>
#include <avr/io.h>
#include <avr/interrupt.h>

#include "binary.h"

#ifdef __cplusplus
extern "C"{
#endif

void yield(void);

#define HIGH 0x1
#define LOW  0x0

#define INPUT 0x0
#define OUTPUT 0x1
#define INPUT_PULLUP 0x2

#define PI 3.14159265358979323846264338327950
#define HALF_PI 1.57079632679489661923132169163975
#define TWO_PI 6.28318530717958647692528676655900
#define DEG_TO_RAD 0.017453292519943295769236907684886
#define RAD_TO_DEG 57.295779513082320876798154814105
```

```
#define EULER 2.718281828459045235360287471352

#define SERIAL  0x0
#define DISPLAY 0x1

#define LSBFIRST 0
#define MSBFIRST 1

#define CHANGE 1
#define FALLING 2
#define RISING 3

#define INTERNAL 3
#define DEFAULT 1
#define EXTERNAL 0

#ifdef abs
#undef abs
#endif

#define min(a,b) ((a)<(b)?(a):(b))
#define max(a,b) ((a)>(b)?(a):(b))
#define abs(x) ((x)>0?(x):-(x))
#define constrain(amt,low,high) ((amt)<(low)?(low):((amt)>(high)?(high):(amt)))
#define round(x)     ((x)>=0?(long)((x)+0.5):(long)((x)-0.5))
#define radians(deg) ((deg)*DEG_TO_RAD)
#define degrees(rad) ((rad)*RAD_TO_DEG)
#define sq(x) ((x)*(x))

#define interrupts() sei()
#define noInterrupts() cli()

#define clockCyclesPerMicrosecond() ( F_CPU / 1000000L )
#define clockCyclesToMicroseconds(a) ( (a) / clockCyclesPerMicrosecond() )
#define microsecondsToClockCycles(a) ( (a) * clockCyclesPerMicrosecond() )
```

```
#define lowByte(w) ((uint8_t) ((w) & 0xff))
#define highByte(w) ((uint8_t) ((w) >> 8))

#define bitRead(value, bit) (((value) >> (bit)) & 0x01)
#define bitSet(value, bit) ((value) |= (1UL << (bit)))
#define bitClear(value, bit) ((value) &= ~(1UL << (bit)))
#define bitToggle(value, bit) ((value) ^= (1UL << (bit)))
#define bitWrite(value, bit, bitvalue) ((bitvalue) ? bitSet(value, bit) : bitClear(value, bit))

#ifndef _NOP
#define _NOP() do { __asm__ volatile ("nop"); } while (0)
#endif

typedef unsigned int word;

#define bit(b) (1UL << (b))

typedef bool boolean;
typedef uint8_t byte;

void init(void);
void initVariant(void);

int atexit(void (*func)()) __attribute__((weak));

void pm(uint8_t pin, uint8_t mode);
void dw(uint8_t pin, uint8_t val);
int dr (uint8_t pin);
int ar(uint8_t pin);
void ar(uint8_t mode);
void aw (uint8_t pin, int val);

void delay(unsigned long ms);

void run_once(void);
void run_always(void);
```

```
#define analogInPinToBit(P) (P)

extern const uint16_t PROGMEM port_to_mode_PGM[];
extern const uint16_t PROGMEM port_to_input_PGM[];
extern const uint16_t PROGMEM port_to_output_PGM[];

extern const uint8_t PROGMEM digital_pin_to_port_PGM[];
extern const uint8_t PROGMEM digital_pin_to_bit_mask_PGM[];
extern const uint8_t PROGMEM digital_pin_to_timer_PGM[];

#define digitalPinToPort(P) ( pgm_read_byte( digital_pin_to_port_PGM + (P) ) )
#define digitalPinToBitMask(P) ( pgm_read_byte( digital_pin_to_bit_mask_PGM + (P) ) )
#define digitalPinToTimer(P) ( pgm_read_byte( digital_pin_to_timer_PGM + (P) ) )
#define analogInPinToBit(P) (P)
#define portOutputRegister(P) ( (volatile uint8_t *)( pgm_read_word( port_to_output_PGM +
(P))) )
#define portInputRegister(P) ( (volatile uint8_t *)( pgm_read_word( port_to_input_PGM +
(P))) )
#define portModeRegister(P) ( (volatile uint8_t *)( pgm_read_word( port_to_mode_PGM +
(P))) )

#define NOT_A_PIN 0
#define NOT_A_PORT 0

#define NOT_AN_INTERRUPT -1

#ifdef ARDUINO_MAIN
#define PA 1
#define PB 2
#define PC 3
#define PD 4
#define PE 5
#define PF 6
#define PG 7
#define PH 8
```

```
#define PJ 10
#define PK 11
#define PL 12
#endif

#define NOT_ON_TIMER 0
#define TIMER0A 1
#define TIMER0B 2
#define TIMER1A 3
#define TIMER1B 4
#define TIMER1C 5
#define TIMER2  6
#define TIMER2A 7
#define TIMER2B 8

#define TIMER3A 9
#define TIMER3B 10
#define TIMER3C 11
#define TIMER4A 12
#define TIMER4B 13
#define TIMER4C 14
#define TIMER4D 15
#define TIMER5A 16
#define TIMER5B 17
#define TIMER5C 18

#ifdef __cplusplus
} // extern "C"
#endif

uint16_t makeWord(uint16_t w);
uint16_t makeWord(byte h, byte l);
//////////////////////////////////////////////////芯片型号引脚定义
#define digitalPinHasPWM(p)             ((p) == 9 || (p) == 10 || (p) == 11)
#else
```

```
#define digitalPinHasPWM(p)                ((p) == 3 || (p) == 5 || (p) == 6 || (p) == 9 || (p) == 10
|| (p) == 11)
#endif

#define PIN_SPI_SS       (10)
#define PIN_SPI_MOSI     (11)
#define PIN_SPI_MISO     (12)
#define PIN_SPI_SCK      (13)

static const uint8_t SS      = PIN_SPI_SS;
static const uint8_t MOSI = PIN_SPI_MOSI;
static const uint8_t MISO = PIN_SPI_MISO;
static const uint8_t SCK     = PIN_SPI_SCK;

#define PIN_WIRE_SDA               (18)
#define PIN_WIRE_SCL               (19)

static const uint8_t SDA = PIN_WIRE_SDA;
static const uint8_t SCL = PIN_WIRE_SCL;

#define LED_BUILTIN 13

#define PIN_A0      (14)
#define PIN_A1      (15)
#define PIN_A2      (16)
#define PIN_A3      (17)
#define PIN_A4      (18)
#define PIN_A5      (19)
#define PIN_A6      (20)
#define PIN_A7      (21)

static const uint8_t A0 = PIN_A0;
static const uint8_t A1 = PIN_A1;
static const uint8_t A2 = PIN_A2;
static const uint8_t A3 = PIN_A3;
static const uint8_t A4 = PIN_A4;
```

```
static const uint8_t A5 = PIN_A5;
static const uint8_t A6 = PIN_A6;
static const uint8_t A7 = PIN_A7;

#define digitalPinToPCICR(p)        (((p) >= 0 && (p) <= 21) ? (&PCICR) : ((uint8_t *)0))
#define digitalPinToPCICRbit(p) (((p) <= 7) ? 2 : (((p) <= 13) ? 0 : 1))
#define digitalPinToPCMSK(p)        (((p) <= 7) ? (&PCMSK2) : (((p) <= 13) ? (&PCMSK0) :
(((p) <= 21) ? (&PCMSK1) : ((uint8_t *)0))))
#define digitalPinToPCMSKbit(p) (((p) <= 7) ? (p) : (((p) <= 13) ? ((p) - 8) : ((p) - 14)))

#define digitalPinToInterrupt(p)    ((p) == 2 ? 0 : ((p) == 3 ? 1 : NOT_AN_INTERRUPT))
#endif
```

3.2.2 bootloader（c\asm）

```
 #include <inttypes.h>
#include <avr/io.h>
#include <avr/pgmspace.h>
#include <avr/eeprom.h>
#include <avr/interrupt.h>
#include <util/delay.h>



#define MAX_TIME_COUNT (F_CPU>>1)
#define HW_VER       0x02
#define SW_MAJOR 0x01
#define SW_MINOR 0x12

#ifndef outb
    #define outb(sfr,val)    (_SFR_BYTE(sfr) = (val))
#endif
#ifndef inb
    #define inb(sfr) _SFR_BYTE(sfr)
#endif

#ifndef cbi
```

```
    #define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
#ifndef sbi
    #define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif

#define LED_DDR    DDRB
#define LED_PORT PORTB
#define LED_PIN    PINB
#define LED          PINB5


#define SIG1      0x1E
#define SIG2      0x93
#define SIG3      0x07
#define PAGE_SIZE 0x20U


void putch(char);
char getch(void);
void getNch(uint8_t);
void byte_response(uint8_t);
void nothing_response(void);

union address_union {
   uint16_t word;
   uint8_t    byte[2];
} address;

union length_union {
   uint16_t word;
   uint8_t    byte[2];
} length;

struct flags_struct {
   unsigned eeprom : 1;
```

```
   unsigned rampz    : 1;
} flags;

uint8_t buff[256];

uint8_t pagesz=0x80;

uint8_t i;

void (*app_start)(void) = 0x0000;

int main(void)
{
   uint8_t ch,ch2;
   uint16_t w;



   asm volatile("nop\n\t");

   UBRRH = (((F_CPU/BAUD_RATE)/16)-1)>>8;        // set baud rate
   UBRRL = (((F_CPU/BAUD_RATE)/16)-1);
   UCSRB = (1<<RXEN)|(1<<TXEN);    // enable Rx & Tx
   UCSRC = (1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);    // config USART; 8N1

   /*  提示灯*/
   sbi(LED_DDR,LED);
     for (i = 0; i < 16; i++) {
      outb(LED_PORT, inb(LED_PORT) ^ _BV(LED));
      _delay_loop_2(0);
     }



   for (;;) {
```

```
 ch = getch();

 if(ch=='0') {
    nothing_response();
 }
else if(ch=='1') {
      if (getch() == ' ') {//命令识别
            putch(0x14);
            putch('A');
            putch('V');
            putch('R');
            putch(' ');
            putch('I');
            putch('S');
            putch('P');
            putch(0x10);
    }
 }


else if(ch=='@') {
    ch2 = getch();
    if (ch2>0x85) getch();
    nothing_response();
 }


else if(ch=='A') {
    ch2 = getch();
    if(ch2==0x80) byte_response(HW_VER);          //  硬件版本
    else if(ch2==0x81) byte_response(SW_MAJOR);//  软件版本
    else if(ch2==0x82) byte_response(SW_MINOR);//  监视软件版本
    //else if(ch2==0x98) byte_response(0x03);          // avrstudio 专用
    else byte_response(0x00);
 }
```

```
else if(ch=='B') {
   getNch(20);
   nothing_response();
}

else if(ch=='E') {
   getNch(5);
   nothing_response();
}


else if(ch=='P')
  nothing_response();


else if(ch=='Q')
   nothing_response();

else if(ch=='R') {
   nothing_response();
}

else if(ch=='U') {
   address.byte[0] = getch();
   address.byte[1] = getch();
   nothing_response();
}

else if(ch=='V') {
   getNch(4);
   byte_response(0x00);
}

else if(ch=='d') {
```

```
        length.byte[1] = getch();
        length.byte[0] = getch();
        flags.eeprom = 0;
        if (getch() == 'E') flags.eeprom = 1;
        for (w=0;w<length.word;w++) {
           buff[w] = getch();                                    /
        }
        if (getch() == ' ') {
              if (flags.eeprom) {
                   for(w=0;w<length.word;w++) {
                        eeprom_wb(address.word,buff[w]);
                        address.word++;
                   }
              } else {

                   cli();
                   while(bit_is_set(EECR,EEWE));
                   asm volatile(//汇编调用
                              "clr     r17       \n\t"
                              "lds     r30,address  \n\t"
                              "lds     r31,address+1    \n\t"
                              "lsl r30                  \n\t"
                              "rol r31                       \n\t"
                              "ldi     r28,lo8(buff)\n\t"
                              "ldi     r29,hi8(buff)\n\t"
                              "lds     r24,length    \n\t"
                              "lds     r25,length+1 \n\t"
                              "sbrs r24,0       \n\t"
                              "rjmp length_loop         \n\t"
                              "adiw r24,1       \n\t"
                              "length_loop:          \n\t"

                              "cpi     r17,0x00      \n\t"
                              "brne   no_page_erase    \n\t"
                              "rcall    wait_spm          \n\t"
//                            "wait_spm1:            \n\t"
```

```
//                  "lds    r16,%0      \n\t"
//                  "andi   r16,1       \n\t"
//                  "cpi    r16,1       \n\t"
//                  "breq   wait_spm1   \n\t"
                    "ldi    r16,0x03    \n\t"
                    "sts    %0,r16      \n\t"
                    "spm            \n\t"
                    "rcall  wait_spm    \n\t"
//                  "wait_spm2:         \n\t"
//                  "lds    r16,%0      \n\t"
//                  "andi   r16,1       \n\t"
//                  "cpi    r16,1       \n\t"
//                  "breq   wait_spm2   \n\t"

                    "ldi    r16,0x11    \n\t"
                    "sts    %0,r16      \n\t"
                    "spm            \n\t"
                    "no_page_erase:     \n\t"
                    "ldr0,Y+        \n\t"
                    "ldr1,Y+        \n\t"

                    "rcall  wait_spm    \n\t"
//                  "wait_spm3:         \n\t"
//                  "lds    r16,%0      \n\t"
//                  "andi   r16,1       \n\t"
//                  "cpi    r16,1       \n\t"
//                  "breq   wait_spm3   \n\t"
                    "ldi    r16,0x01    \n\t"
                    "sts    %0,r16      \n\t"
                    "spm            \n\t"

                    "inc    r17     \n\t"
                    "cpi r17,%1         \n\t"
                    "brlo   same_page   \n\t"
                    "write_page:        \n\t"
                    "clr    r17     \n\t"t
```

```
                    "rcall   wait_spm          \n\t"
//                  "wait_spm4:          \n\t"
//                  "lds    r16,%0       \n\t"
//                  "andi   r16,1                \n\t"
//                  "cpi    r16,1                \n\t"
//                  "breq   wait_spm4            \n\t"
                    "ldi    r16,0x05     \n\t"
                    "sts    %0,r16       \n\t"
                    "spm              \n\t"
                    "rcall   wait_spm          \n\t"
//                  "wait_spm5:          \n\t"
//                  "lds    r16,%0       \n\t"
//                  "andi   r16,1                \n\t"
//                  "cpi    r16,1                \n\t"
//                  "breq   wait_spm5            \n\t"

                    "ldi    r16,0x11     \n\t"
                    "sts    %0,r16       \n\t"
                    "spm              \n\t"
                    "same_page:          \n\t"
                    "adiw   r30,2        \n\t"
                    "sbiw   r24,2        \n\t"
                    "breq   final_write  \n\t"
                    "rjmp   length_loop  \n\t"

                    "wait_spm:   \n\t"
                    "lds    r16,%0       \n\t"
                    "andi   r16,1                \n\t"
                    "cpi    r16,1                \n\t"
                    "breq   wait_spm           \n\t"
                    "ret              \n\t"

                    "final_write:        \n\t"
                    "cpi    r17,0        \n\t"
                    "breq   block_done   \n\t"
                    "adiw   r24,2        \n\t"
```

```
                        "rjmp   write_page   \n\t"
                        "block_done:         \n\t"
                        "clr    __zero_reg__    \n\t"
                        :    "=m"    (SPMCR)    :    "M"    (PAGE_SIZE)    :
"r0","r16","r17","r24","r25","r28","r29","r30","r31");

            }
            putch(0x14);
            putch(0x10);
        }
    }
    else if(ch=='t') {
      length.byte[1] = getch();
      length.byte[0] = getch();
      if (getch() == 'E') flags.eeprom = 1;
      else {
            flags.eeprom = 0;
            address.word = address.word << 1;
      }
      if (getch() == ' ') {
            putch(0x14);
            for (w=0;w < length.word;w++) {                    /
                if (flags.eeprom) {
                    putch(eeprom_rb(address.word));
                    address.word++;
                } else {
                    if (!flags.rampz) putch(pgm_read_byte_near(address.word));
                    address.word++;
                }
            }
            putch(0x10);
      }
    }

    else if(ch=='u') {
      if (getch() == ' ') {
```

```
                putch(0x14);
                putch(SIG1);
                putch(SIG2);
                putch(SIG3);
                putch(0x10);
            }
        }


        else if(ch=='v') {
            byte_response(0x00);
        }
//        } else {
//            time_count++;
//            if (time_count>=MAX_TIME_COUNT) {
//                app_start();
//            }
//    }
       }
}

void putch(char ch)
{

    while (!(inb(UCSRA) & _BV(UDRE)));
    outb(UDR,ch);
}

char getch(void)
{
      uint32_t count = 0;
    while(!(inb(UCSRA) & _BV(RXC))) {

        count++;
        if (count > MAX_TIME_COUNT)
            app_start();
```

```
    }
    return (inb(UDR));
}

void getNch(uint8_t count)
{
    uint8_t i;
    for(i=0;i<count;i++) {

      getch();
    }
}

void byte_response(uint8_t val)
{
    if (getch() == ' ') {
        putch(0x14);
        putch(val);
        putch(0x10);
    }
}

void nothing_response(void)
{
    if (getch() == ' ') {
        putch(0x14);
        putch(0x10);
    }
}
```

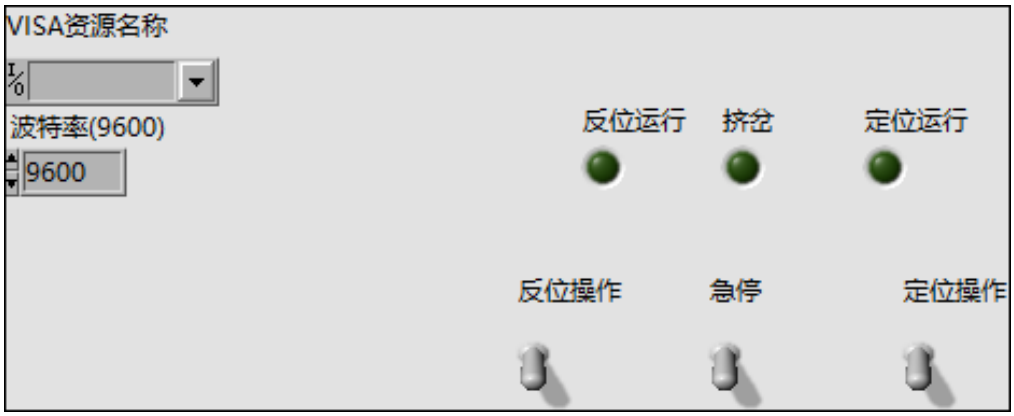
3.3 cpld 逻辑输入输出编写（verilog）

略

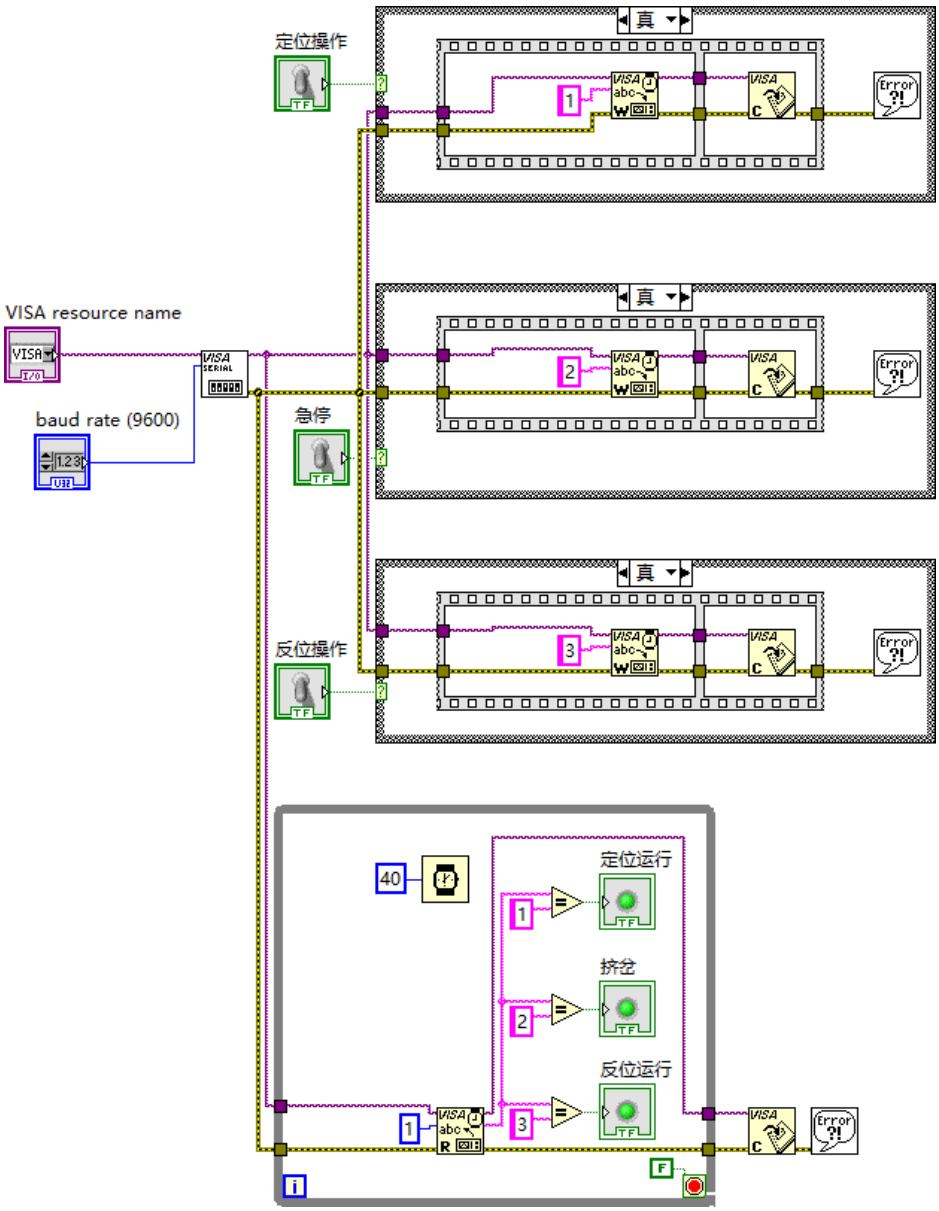
3.4 labview visa 通讯程序编写

3.4.1 操作机

3.4.1.1 前面板

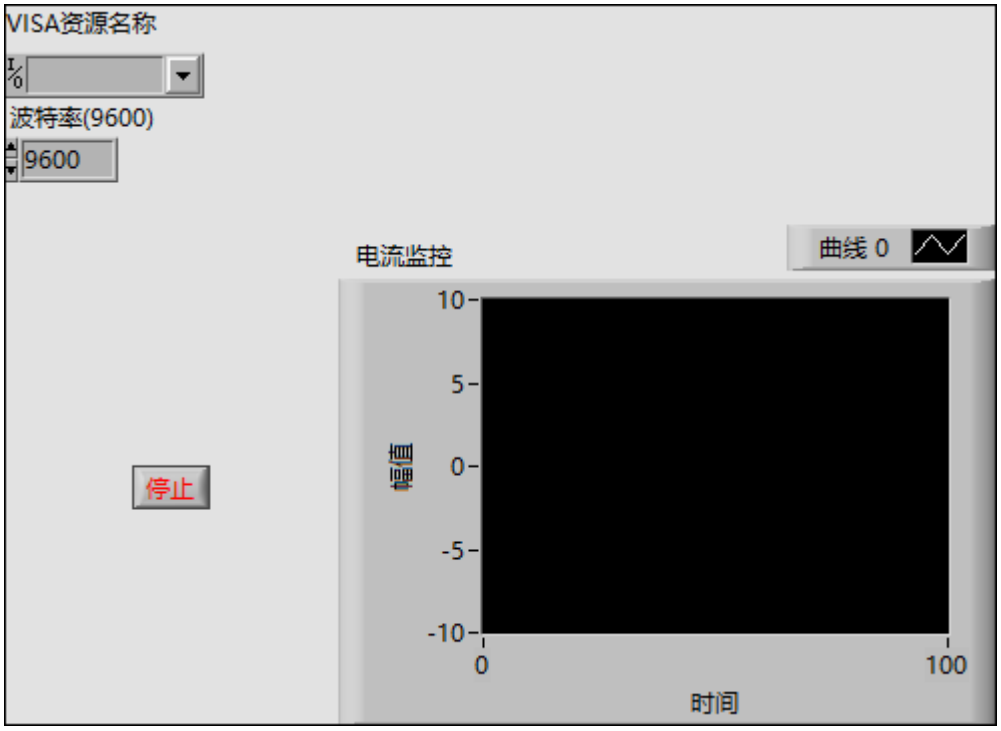


3.4.1.2 程序框图

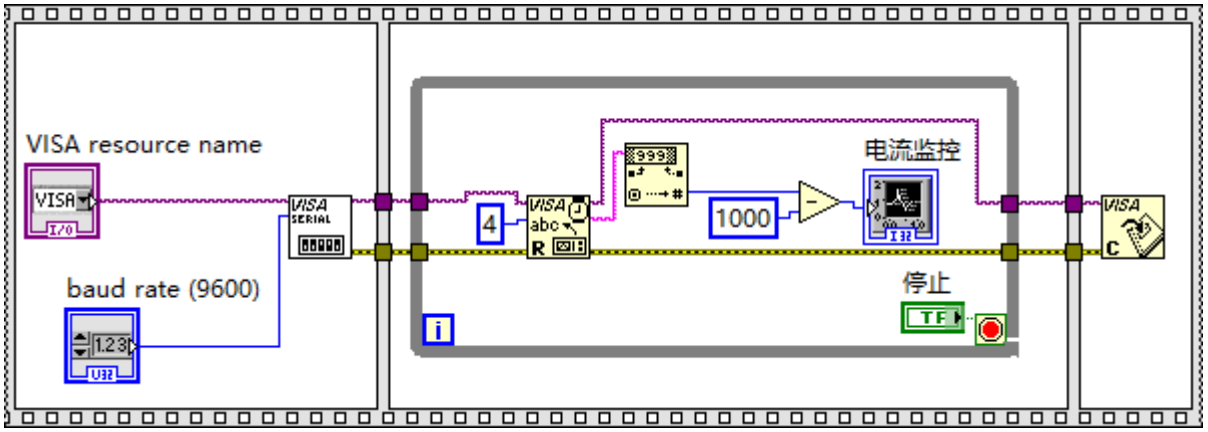

定位操作
真
VISA resource name
baud rate (9600)
急停
反位操作
40
定位运行
挤岔
反位运行
Error

3.4.2 电流监控机

3.4.2.1 前面板



3.4.2.2 程序框图



结　　论

经过电路仿真及小型化模型的构建，本文中设计部分均通过有效性验证，实地测试环节由于限制无法进行。但在进行过的验证环节中构建的设备均达到文中提到的实际目标，在完成了本文开头的任务计划书所规定的内容外，做出许多改进与创新，同时进一步提高设备可用性、安全性、兼容性、稳定性、可靠性。

参 考 文 献

附录 A 公开发行许可证

Mozilla Public License Version 2.0
==================================

1. Definitions

1.1. "Contributor"
    means each individual or legal entity that creates, contributes to
    the creation of, or owns Covered Software.

1.2. "Contributor Version"
    means the combination of the Contributions of others (if any) used
    by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
    means Covered Software of a particular Contributor.

1.4. "Covered Software"
    means Source Code Form to which the initial Contributor has attached
    the notice in Exhibit A, the Executable Form of such Source Code
    Form, and Modifications of such Source Code Form, in each case
    including portions thereof.

1.5. "Incompatible With Secondary Licenses"
    means

    (a) that the initial Contributor has attached the notice described
        in Exhibit B to the Covered Software; or

    (b) that the Covered Software was made available under the terms of
        version 1.1 or earlier of the License, but not also under the
        terms of a Secondary License.

1.6. "Executable Form"
    means any form of the work other than Source Code Form.

1.7. "Larger Work"
    means a work that combines Covered Software with other material, in
    a separate file or files, that is not Covered Software.

1.8. "License"
    means this document.

1.9. "Licensable"
    means having the right to grant, to the maximum extent possible,
    whether at the time of the initial grant or subsequently, any and

all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its

Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after

```
Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent
infringement claim (excluding declaratory judgment actions,
counter-claims, and cross-claims) alleging that a Contributor Version
directly or indirectly infringes any patent, then the rights granted to
You by any and all Contributors for the Covered Software under Section
2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all
end user license agreements (excluding distributors and resellers) which
have been validly granted by You or Your distributors under this License
prior to termination shall survive termination.

************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the       *
*  Covered Software is free of defects, merchantable, fit for a        *
*  particular purpose or non-infringing. The entire risk as to the     *
*  quality and performance of the Covered Software is with You.        *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,   *
*  repair, or correction. This disclaimer of warranty constitutes an   *
*  essential part of this License. No use of any Covered Software is   *
*  authorized under this License except under this disclaimer.         *
*                                                                      *
************************************************************************

************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
*  --------------------------                                          *
*                                                                      *
*  Under no circumstances and under no legal theory, whether tort      *
*  (including negligence), contract, or otherwise, shall any           *
*  Contributor, or anyone who distributes Covered Software as          *
*  permitted above, be liable to You for any direct, indirect,         *
*  special, incidental, or consequential damages of any character      *
*  including, without limitation, damages for lost profits, loss of    *
*  goodwill, work stoppage, computer failure or malfunction, or any    *
*  and all other commercial damages or losses, even if such party      *
*  shall have been informed of the possibility of such damages. This   *
*  limitation of liability shall not apply to liability for death or   *
*  personal injury resulting from such party's negligence to the       *
*  extent applicable law prohibits such limitation. Some               *
```

```
*  jurisdictions do not allow the exclusion or limitation of          *
*  incidental or consequential damages, so this exclusion and         *
*  limitation may not apply to You.                                  *
*                                                                    *
**********************************************************************

8. Litigation
-------------

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place of business and such litigation shall be governed by laws of that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous
----------------

This License represents the complete agreement concerning the subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
necessary to make it enforceable. Any law or regulation which provides
that the language of a contract shall be construed against the drafter
shall not be used to construe this License against a Contributor.

10. Versions of the License
---------------------------

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section
10.3, no one other than the license steward has the right to modify or
publish new versions of this License. Each version will be given a
distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version
of the License under which You originally received the Covered Software,
or under the terms of any subsequent version published by the license
steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to
create a new license for such software, you may create and use a
modified version of this License if you rename the license and remove
any references to the name of the license steward (except to note that
such modified license differs from this License).
```

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

  This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

  This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

附录 B NI 专利信息

National Instruments Fieldbus hardware products (including Foundation Fieldbus, CAN and DeviceNet) are covered by one or more of the following Patents:

U.S. Patent No(s).: 5,938,754

MXI-3 and MXI-4 Module Products

National Instruments MXI-3 and MXI-4 Module products are covered by one or more of the following Patents:

U.S. Patent No(s).: 6,425,033; 6,418,504; 6,950,440; 6,968,464

Radio Frequency (RF) Products

National Instruments RF products are covered by one or more of the following Patents:

U.S. Patent No(s).: 7,268,642

CompactRIO Products

National Instruments CompactRIO products are covered by one or more of the following Patents:

U.S. Patent No(s).: 6,823,283; 7,254,512

PXI Express

National Instruments PXI Express products are covered by one or more of the following Patents:

U.S. Patent No(s).: 6,981,086; 6,941,390

PC Cards

PC Cards manufactured and/or sold by National Instruments are licensed under one or more of the following Patents:

U.S. Patent No(s).: 4,603,320; 4,972,470

NI-ELVIS

The NI-ELVIS product is covered by one or more of the following Patents:

U.S. Patent No(s).: 6,895,563

Smart Camera

National Instruments Smart Camera products are covered by one or more of the following Patents:

U.S. Patent No(s).: 7,327,396

Other U.S. and International Patents Pending

For the most updated listing of patents which may cover the National Instruments product(s) being used, please see www.ni.com/patents

致　　谢

感谢孙会祥导师对我的悉心栽培和严格要求，以及提出的项目设计任务大纲和论文期间做出的指导。

感谢城轨创客空间给予的支持和帮助：前团队指导巩文东讲师、前团队项目助理徐立欣同学以及众多同学。

感谢对本项目伸出援手的同学和朋友们：山职钱进、潘瑞洋、焦念龙、刘泳峰、高昕，西安交大崔玉清，山大 Jason Sun 、Weqian Lee，山建胡俊伟等。

感谢提供资源的社区贡献者：Victor、Gkasprow、Facchinm、DarkSector

感谢本设计中提供支持、评估的组织（排名不分先后）：Microsoft（Azure、Github、365）、Lattice Semi、Swiss Academic、National Instruments、Microchip、Mouser Electronics、LCEDA、Dassault Systèmes 、Altium。