摘要

本次设计将采用STM32F1作为控制芯片，使用RM装甲板，运用数模电、电路设计、单片机原理、C语言等相关知识设计出自动化旋转靶标。

本论文设计基于2019年全国大学生机器人大赛Robomaster赛事中的旋转靶标系统。设计一个五角型叶面靶标，在五角靶标系统初始化后自动旋转，每个靶标叶面顶部设有压力传感器，可以对外力射击进行测量统计，并在每个靶标叶面的条形LED屏上显示一定的信息，全部5个靶标被射中时，系统灯带闪烁若干时间，以示击打成功。

本设计产品基于STM32F1单片机，

+整体运行介绍

关键词：

**Abstract**

目录

1. 绪论
2. 旋转靶标控制系统硬件设计
3. 旋转靶标控制系统软件设计

总结和展望

参考文献

致谢

附录

硬件

STM32F103ZET6\_LQFP144:

Frequency ：72Mhz

Core：ARM,Cortex-M3

Flash：512Kbytes

RAM：64Kbytes

V:2-3.6V

.1内核：

**ARM 32**位的**Cortex™-M3 CPU**

最高72MHz工作频率，在存储器的0等待周

期访问时可达1.25DMips/MHz(Dhrystone

单周期乘法和硬件除法

存储器

- 从256K至512K字节的闪存程序存储器

- 高达64K字节的SRAM

- 带4个片选的静态存储器控制器。支持CF卡、SRAM、 PSRAM、 NOR和NAND存储器

时钟、复位和电源管理

- 2.0～3.6伏供电和I/O引脚

- 上电/断电复位(POR/PDR)、可编程电压监测

器(PVD)

- 4～16MHz晶体振荡器

- 内嵌经出厂调校的8MHz的RC振荡器

- 内嵌带校准的40kHz的RC振荡器

- 带校准功能的32kHz RTC振荡器

■ 低功耗

- 睡眠、停机和待机模式

- VBAT为RTC和后备寄存器供电

■ **3**个**12**位模数转换器， **1μs**转换时间**(**多达**21**个

输入通道**)**

- 转换范围： 0至3.6V

- 三倍采样和保持功能

- 温度传感器

■ **2** 通道 **12** 位 **D/A** 转换器

**DMA**： **12** 通道 **DMA** 控制器

- 支持的外设：定时器、 ADC、 DAC、 SDIO、

I2S、 SPI、 I2C和USART

■ 调试模式

- 串行单线调试(SWD)和JTAG接口

- Cortex-M3内嵌跟踪模块(ETM)

■ 多达**112**个快速**I/O**端口

- 51/80/112个多功能双向的I/O口，所有I/O口

可以映像到16个外部中断；几乎所有端口均

可容忍5V信号

多达**11**个定时器

- 多达4个16位定时器，每个定时器有多达4个

用于输入捕获/输出比较/PWM或脉冲计数的

通道和增量编码器输入

- 2个16位带死区控制和紧急刹车，用于电机

控制的PWM高级控制定时器

- 2个看门狗定时器(独立的和窗口型的)

- 系统时间定时器： 24位自减型计数器

- 2个16位基本定时器用于驱动DAC

■ 多达**13**个通信接口

- 多达2个I2C接口(支持SMBus/PMBus)

- 多达5个USART接口(支持ISO7816， LIN，

IrDA接口和调制解调控制)

- 多达3个SPI接口(18M位/秒)， 2个可复用为

I2S接口

- CAN接口(2.0B 主动)

- USB 2.0全速接口

- SDIO接口

■ **CRC**计算单元， **96**位的芯片唯一代码

配件

M3508主要面向行业应用市场，具备强劲的动力和伺服性能，可服役于物流搬运、机器人等领域。适用于10-50Kg重量级的机器人移动平台和执行机构

搭配C620电调使用时，具有位置反馈、温度检测和信息存储的功能。

M3508直流无刷减速电机参数

|  |  |
| --- | --- |
| 额定电压： | 24V |
| 空载转速： | 482rpm |
| 持续最大扭矩： | 3N·m |
| 3N·m下最大转速： | 469rpm |
| 使用环境温度： | 0-50° C |
| 重量： | 365g |
| 外径： | 42mm |
| 总长度： | 98 |
| 输出轴： | D型带螺纹孔 |
| 输出轴直径： | 10mm |

 支持通过CAN总线获取电机温度、转子位置和转子转速等信息，切换电机时可无需进行霍尔校准

C620无刷电机调速器参数

|  |  |
| --- | --- |
| 额定电压： | 24V |
| 重量： | 35g |
| 尺寸（长宽高，不含线）： | 49.4\*25.8\*11.5mm |
| 带线总长度： | 344±15mm |
| 信号类型： | CAN指令、PWM |
| 最大持续电流： | 20A |

此处添加.SCH文件

主控电路板涉及到的IO口：

GPIOA11:can接收

GPIOA12:can发送

电子芯片购买，查看数据手册网站

[*http://www.ic37.com/*](http://www.ic37.com/)*，*<https://www.wlxmall.com/>，<https://www.digikey.cn/>

AD学习

AD新建工程

新建集成库工程

|  |  |  |
| --- | --- | --- |
| 1.新建- | -项目--集成库 | 下面会自动生成history的文件夹 |
| 2.保存 | workspace.dsnwrk 到自己的文件夹 |  |
|  | 集成库.LibPkg 到自己的文件夹 |  |
| 3.新建 | 库->原理图库.schLib |  |
|  | 绘制原理图器件: | 设计原理图中的电阻电容 |
| 4.新建 | 库->PCB元器件库.pcblib |  |
|  | 绘制PCB器件: | 设计实际元器件大小 |
|  | 3D模型是模拟的实际元器件eg：电容。按3可以查看元器件的3d状态，确定实物是否可以正确焊接在PCB上面  按shift转动3d模型 |  |
| 5.导入 | PCB3D库.pcb3DLib | AD无法绘制PCB3D库，可通过Pro等软件绘制后导入AD |
|  | 将原理图库的器件和PCB库的器件对应起来 |  |
| 6. | 工程->编译.schlib,.LibPkg  编译后生成集成库.IntLib | 集成库.IntLib ：是最终目的，在之后画32原理图以及生成PCB时可以直接使用（.IntLib既有原理图.schlib里的电阻电容又有PCB时.pcblib的配套电阻电容） |

新建PCB工程

|  |  |  |
| --- | --- | --- |
| 新建 | 项目->PCB工程 |  |
| 2.保存 | workspace.dsnwrk 到自己的文件夹 |  |
|  | PCB工程.PrjPCB 到自己的文件夹 |  |
| 3.新建 | 原理图.schdoc  添加.intlib集成库，绘制32总的原理图使用 |  |
| 4.新建 | PCB.pcbdoc |  |
| 5.自动生成PCB | 设计（D）->Update PCB Document->  执行更改 |  |

AD绘制原理图库器件.schlib

|  |  |
| --- | --- |
| 放置管脚 | 按tab，出现designator(标识)，用来绘制PCB连接时的标志符 |
|  | 管脚上有一端有×，朝外。按空格键可旋转器件方向 |
|  | 如何设置芯片的管脚顺序查看芯片管脚手册 |
|  | 管脚低电平有效的书写方法：P\S\E\N\ |
| 某元器件原理图画好设置属性名称 | 双击AD左边界面的元器件名称，右边界面出现Proper |
|  | designator（标识）eg：填写U?, 使用时自动排序为U1还是U2,U3 |
|  | Links里面的name：填写该元器件名称，是电阻电容还是其他 |
|  | Param里面设置参数大小，eg：电阻大小为10k |

AD绘制PCB库器件.pchlib

|  |  |  |
| --- | --- | --- |
| 放置焊盘 | 按tab或者双击焊盘可设置焊盘属性 |  |
| 绘制电阻时 | 实际电阻两端焊盘间距离为300mil |  |
|  | 点击：编辑->设置参考->位置 |  |
|  | 把第一个焊盘的中心设置为参考点 |  |
|  | 放置第二个焊盘可直接看左下角界面X=300mil即可 |  |
| 检测距离 | 点击：报告（R）->测量 |  |
| 电阻走线 | 在Top Overlay层上操作，因为只是布置外观 |  |

|  |  |
| --- | --- |
| 难点： | 焊盘间距离必须通过实际元器件大小确定 |
|  | 粉红色mech层为PCB实物占地面积表面层，它可以查看实物3d是否和PCB吻合 |

AD绘制PCB

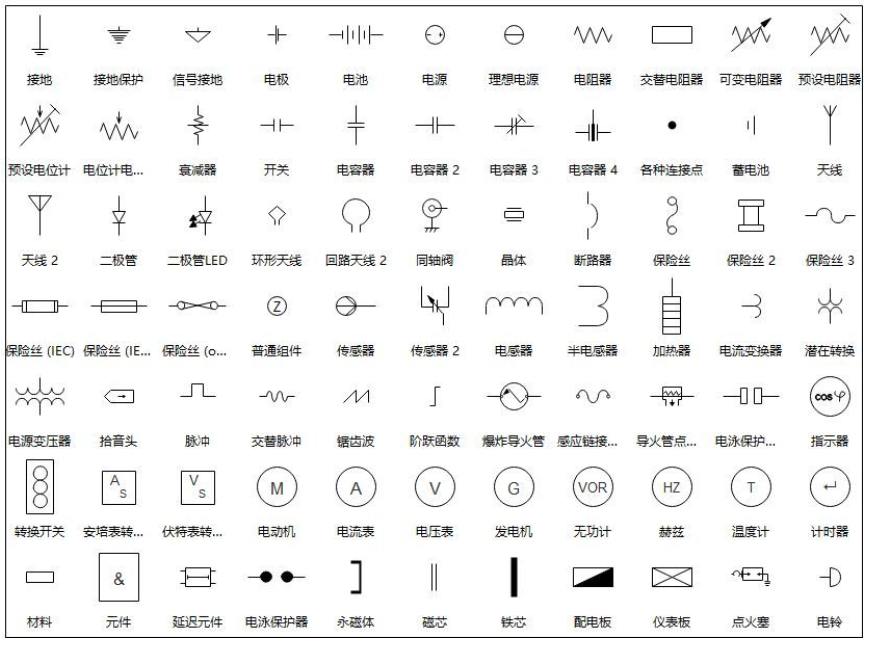
1mil=0.0245mm

|  |  |  |
| --- | --- | --- |
| 基本操作 | 放置--走线 |  |
|  | 编辑--原点--复位 | 设置原点，定位尺寸方便 |
|  | 按2查看2D模型，3查看3D模型 |  |
|  | 按空格旋转 |  |

|  |  |
| --- | --- |
| 设计板子形状，大小 | 在粉红色Mech层走线布置一个长方形，按shift选中 |
|  | 设计(D)--板子形状(S)--按照选择对象定义(D) |
| 布线规则 | 设计--规则  布线--自动布线--全部 |

|  |  |
| --- | --- |
| 知识点 | 把滤波电容放在电源模块附近，效果会好点 |
|  | 布线的时候，上层布横线，下层就布竖线，若上下层都布横线会出现干扰  线与线之间会出现电容啥的 |
|  | 布线的时候不要布置成环路，环路面积小 |
|  | VCC线比其他信号线布置的粗一点 |
|  | GND线最好铺铜 |
|  | 线越短越粗好 |
|  | 过孔也称金属化孔。在双面板和多层板中，为连通各层之间的印制导线，在各层需要连通的[导线](https://baike.baidu.com/item/%E5%AF%BC%E7%BA%BF/1413914)的交汇处钻上一个公共孔，即过孔。过孔的参数主要有孔的外径和钻孔尺寸。  孔本身存在着对地的寄生电容，同时也存在着寄生电感，往往也会给电路的设计带来很大的负面效应。 |

电路图形符号



.1传感器电路板

.1电路板介绍

组成部分：有4片LM393芯片，1片74HCT332芯片，4个输入输出模块。

传感器电路板上 有4个输入接口，分别输入4路压力传感器信号，每路信号经过LM393电压比较器芯片处理后输出一个信号（是否检测到压力），4个芯片共输出4路信号后输入74HCT32芯片，该芯片再输出一个信号到一个排针上。

.1.1 LM393芯片

1. LM393介绍

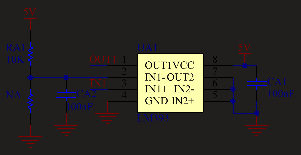
LM393是电压比较器芯片。**正向输入的电压高于负向输入电压，比较器将输出高电平；反之则低。**它有8个引脚：

|  |  |
| --- | --- |
| 引脚组成： | 4个GND |
|  | 5V |
|  | 输入 |
|  | 输出 |
|  | 网络引脚 |

(2)LM393**应用：**

* 输入两路信号
* 以反向输入（-）为参考电压
* 以正向输入（+）为需要判断的信号
* 输出接入一个LED，负极接入输出端，正极接Vcc

(3)LM393原理图



.1.2 74HCT32芯片

(1)74HCT32介绍

74HCT324芯片是4路2输入或门，输入包括钳位二极管，使得能够使用限流电阻器来将输入接口连接到超过的电压。它有14个引脚。

|  |  |
| --- | --- |
| 用到的引脚 | 没有用到的引脚 |
| OUT | 2个OUTA |
| OUT1~OUT4 | 2个OUTB |
| 1个5V的VCC  4个引脚接GND |  |

(2)74HCT32特点和好处

* 宽电源电压范围为2V~6V
* 74HCT32输入电平:TTL电平
* 对称的输出阻抗+
* 高噪声抗扰度
* 低功耗
* 平衡传输延迟
* 符合EDEC标准ESD7A
* 温度范围 -40℃~85℃

(3)74HCT32原理图



.2灯条驱动原理图

.1灯条驱动电路介绍

（1）整体布局说明

灯条驱动电路板在P11,P12引入10个排针IO口引脚输入到2个LED显示器的驱动器芯片TD62081FG上，驱动器芯片再输出10个信号到Kn，最后输出到LED灯条上。Xl4005芯片输出电压12V接一个LED，MP2482DN芯片输出电压5V接一个LED。在PCB布局上，Kn、Sn、Pn布置在一块，布置在五角型外围，内部有2组输入排针引脚及LED显示驱动芯片，以及电压转换电路。

（2）灯条驱动电路板共有4个模块，U1~U4。如下是模块介绍：

|  |  |
| --- | --- |
| 1个24V转12V/4A模块 | 芯片xl4005 |
| 1个24V转5V模块 | 芯片MP2482DN |
| 2个LED显示器的驱动器 | 芯片TD62081FG |
| 10个按键模块（12V） | K1~K10,S1~S10 |

（3）如下是排针引脚介绍：

|  |  |  |
| --- | --- | --- |
| 排针 | P1~P13 |  |
| P1~P10引脚: | RA1、BA1,24V~RE2、BE2,24V | 10个信号引脚 |
| P11~P12引脚： | 2个5V,2个GND,IN1a、IN1b~IN5a、IN5b | 10个信号引脚 |
| P13 | 24V,GND | 0 |

.2芯片xl4005

芯片xl4005是一个6引脚芯片，它是一个5A 、300KHz 、32V的降压DC - DC变换器。

1. 芯片xl4005特性

* 输入电压范围：宽5V至32V
* 输出电压范围可调：从0.8V到30V
* 最高占空比: 100%
* 最小电压降: 0.6V
* 固定开关频率: 300KHz
* 恒定输出电流能力：5A
* 内部优化功率MOSFET
* 成员拥有高优良的线路和负载调节
* TTL关闭功能
* 具有迟滞功能的EN引脚
* 内置热停堆功能，
* 内置限流功能
* 内置输出短保护功能，
* 可用于TO-263软件包

1. 芯片xl4005应用程序

* 液晶显示器
* 液晶电视
* 数码相框设置框ADSL
* 调制解调器电信/网络设备

（3）芯片xl4005一般的描述

XL4005是300KHz固定频率PWM降压DC/DC变换器，可驱动5A级负载，效率高，纹波低，线路和负载调节性能好。要求最小数量的外部组件，requlator是简单的使用和包括内部频率补偿和频振荡器

PWM控制电路能够将占空比从O线性调整到100%。一个使能功能，一个过电流保护功能内置。当保护功能不足时，运行频率由300KHz降至60KHz。内置内部补偿块以最小化外部组件计数。

（4）芯片xl4005原理图



.3芯片MP2482DN

1. 芯片MP2482DN介绍

SR2026是一款单片降压型开关,模式转换器具有一个内置的内部电源MOSFET。它有8个引脚。它实现了5A的连续输出,目前在宽输入电压范围内,有优异的负载和线路调整。电流模式能提供快速的瞬态，响应和简化环路稳定性。故障状态下保护包括逐周期，电流限制和热关断。在8引脚SOIC封装，可裸漏焊盘。

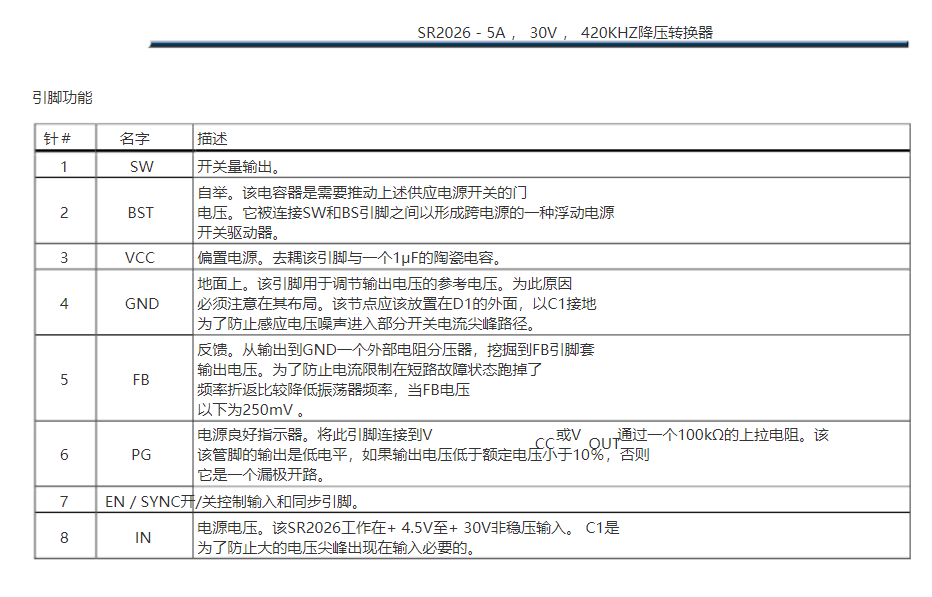
（2）芯片MP2482DN特点

* 广泛的4.5V至30V输入工作范围
* 5A输出电流
* 50mΩ的内部功率MOSFET开关
* 电源良好指示器
* 固定的420kHz频率
* 可同步高达1.5MHz
* 逐周期过流保护用
* 热关断
* 输出从0.81V可调
* 稳定的低ESR陶瓷输出
* 电容器
* 采用热增强型8引脚
* SOIC封装

（3）芯片MP2482DN应用

* 数字机顶盒
* 个人视频录像机
* 宽带通信
* 平板电视和显示器

（4）芯片MP2482DN引脚说明



（5）芯片MP2482DN原理图



.4芯片TD62081FG

（1）芯片TD62081FG介绍

TD62081FG18是一个18引脚的芯片。TD62081AP/CP/F/AF系列是由8对NPN达林顿对组成的高压、大电流达林顿驱动器。所有单元的特点是集成箝位二极管用于开关感应负载应用包括继电器，锤子，灯和显示(LED)驱动器。

（2）芯片TD62081FG的特点

* 输出电流（单输出）：500毫安（最大） （ TD62081AP / F / AF系列）
* 持续的高电压输出：35 V （最小）。（ TD62081F系列）
* 输出钳位二极管
* 投入使用的各类逻辑兼容。
* 封装形式-F ， AF ： SOP- 18针

（3）芯片TD62081FG原理图



点阵驱动电路图

.1电路板说明

(1)整个电路板的输入输出，中间变换说明:?

|  |  |
| --- | --- |
| 4片电路中74HC245 | 起总线缓冲作用， |
| 1片74HC138 | 构成2 - 8 译码器，实现8 路的行选通，选通信号经由4953MOS 管激励为行驱动，行驱动为H1 ~ H8,每一行驱动负责32 × 32 点阵两行的选通( 如H1 控制第1 行和第9 行，H2 控制第2 行和第10 行) . |
| 12片74HC595 | 串行数据由74HC245 缓冲后，送入级联的74HC595 串并转换和驱动，电路中共设计有12片74HC595,其中4 片用于第1 行~ 第16 行的列数据转换，另外4 片用于第17 行~ 第32 行的列数据转换。 |

(2)电路板输入输出引脚介绍

两组排针，每组14个

P1输入：

5V,GND

A\_I~C\_I

SCK\_I,RCK\_I

SI\_I,

OE(R)\_I,OE(B)\_I,OE(G)\_I

P2输出：

同上

输入排针的SCK\_I,A\_I,B\_I,C\_I输入到74HC245芯片的输入引脚上

.2芯片74HC595

(1)芯片74HC595介绍

本电路的U1-U12都是该芯片。该芯片是8位串行输入，串行/并行输出移位寄存器。它是高速硅栅CMOS器件和引脚兼容低功耗肖特基TTL （ LSTTL ） 。它们是与JEDEC规定的遵守标准号 7A 。

74HCT595是8级串行移位寄存器与存储寄存器和三态输出。该寄存器具有独立的时钟。数据被移位的移位寄存器的时钟输入端（ SHCP ）的正向跳变。

在每个寄存器中的数据传送到存储寄存器上的正向。存储寄存器时钟输入（ STCP ）的过渡。如果两个时钟是连接在一起的，该移位寄存器将总是一个时钟脉冲超前的存储寄存器的。移位寄存器有一个串行输入（ DS）和一个串行标准输出（ Q7S ），用于级联。它也设置有异步复位（低电平有效）所有8个移位寄存器级。该存储寄存器有8个平行的三态总线驱动器输出。在存贮寄存器中的数据出现在每当输出使能输入端（OE）为低电平的输出。

(2) 芯片74HC595特点

* 8位串行输入
* 8位串行或并行输出
* 存储寄存器具有三态输出
* 移位寄存器直接明确
* 100兆赫（典型值）移频出
* ESD保护：
* HBM JESD22- A114F超过2000伏
* MM JESD22 - A115 - A超过200 V
* 多种封装选择
* 温度范围：-40℃~85℃

(3) 芯片74HC595应用

* 串行到并行的数据转换
* 遥控器保持寄存器

(4) 芯片74HC595引脚说明

本电路板用到12片该芯片，并且分成三组：

* 公共的SCK引脚，RCK引脚，OE(R)引脚，电源驱动为5V
* SI控制信号接第一片芯片的SER引脚，第一片芯片的QH1引脚再接到第2片芯片的SER引脚，以此类推
* 芯片的8个输出引脚QA~QH输出到点阵矩阵中

(5) 芯片74HC595原理图



.3芯片74HC138

(1) 芯片74HC138介绍

该74HC138芯片是一款高速CMOS器件。该器件接受的输入引脚的3位二进制加权地址A0，A1和A2和启用时将产生一个低电平输出与remaing 7为高。有两个低电平有效使能输入端E1和E2 ，以及一种活性高电平使能输入E3 。残疾人设备状态导致所有输出为高。发生在启用状态， E1和E2置低，

E3置为高电平。多能线允许解码器的并联扩容以创建4至16线的版本，没有额外的部件和5至32版本与另外一台逆变器的。

(2) 芯片74HC138特点

* 宽电源电压范围从2.0V至6.0V
* CMOS低功耗
* 施密特触发器动作在所有输入
* 输入接受高达6.0V
* 每JESD 22 ESD保护测试
* 超过200 -V机型号（ A115 -A ）
* 超过2000 -V人体模型（ A114 -A ）
* 超过1000 -V带电器件模型（ C101C ）
* 闭锁超过每JESD 78D ， II类100毫安

(3) 芯片74HC138应用

* 内存芯片选择解码
* 解复用
* 单线外设控制
* 从单片机控制，因为允许简单的串行比特流
* 根据需要许多外围线

(4) 芯片74HC138引脚说明

* 电源驱动为5V
* 三个输入引脚ABC组成3位二进制地址
* 芯片8个输出引脚输出到点阵驱动的r17~r20，r29~r32

(5) 芯片74HC138原理图



.3芯片74HC245

(1) 芯片74HC245介绍

该芯片是一个8路3态总线收发器，本电路板的U14-U17都是该芯片。

八路三态同相总线收发器。 该74HC245与引出线的LS245是相同的。该器件的输入与标准CMOS输出兼容;与上拉电阻，它们与LSTTL输出兼容。在HC245 是用于三态倒相收。发数据总线之间的2路异步通信。该装置有一个低电平有效输出使能引脚，用于放置I / O端口进入高阻抗状态。方向控制决定从A是否数据流向乙或从B到A的。

(2) 芯片74HC245特点

* 输出驱动能力： 15输入通道负载
* 输出直接连接到CMOS ， NMOS和TTL
* 工作电压范围： 2.0〜 6.0 V
* 低输入电流： 1.0mA
* CMOS器件的高抗噪声特性
* 符合由JEDEC标准规定的要求第7A号
* 芯片的复杂性： 308场效应管或77个等效门
* 这是一个Pb - Free设备

(3) 芯片74HC245引脚说明

本电路板用到4个该芯片

|  |  |  |
| --- | --- | --- |
| 第1片 | 输入 | SCK\_I,A\_I,B\_I,C\_I |
|  | 输出 | SCK，138\_A/B/C |
| 第3片 | 输入 | 为第一片的输出 |
|  | 输出 | SCK\_II,A/B/C\_II |
| 第2片 | 输入 | RCK\_I,SI\_I,OE（R）\_I,OE(B)\_I,OE(G)\_I |
|  | 输出 | RCK,SI,OE(R),OE(B),OE(G) |
| 第4片 | 输入 | 为第2片的输出引脚 |
|  | 输出 | RCK\_II,SI\_II,OE（R）\_II,OE(B)\_II,OE(G)\_II |
|  | 再输出 | 到输出排针上 |

(4) 芯片74HC245原理图



主控STM32电路板

.1电路板介绍

（1）电路板芯片组成

|  |  |  |
| --- | --- | --- |
| U2 | 芯片MIC3975YMM | 8引脚 |
| U4 | 芯片SN65HV32 | **8引脚,CAN收发器** |
| U5 | PL2303SA | **8引脚,** |

（2）电路板排针引脚介绍

|  |  |
| --- | --- |
| P1 | 下载口 |
| P2 | USART1 |
| P3 | CAN1 |
| P4 | CAN1 |
| PA,PB | 7引脚，灯条驱动接口，共10给IO口；PE0~PE6,PB8~PB9,PC13 |
| JP1~JP5 | 14引脚，点阵驱动IO口；共70 |
| J1 | 5引脚，5V |
| J2 | 5引脚，3.3V |
| J3 | 5引脚，传感器接口PB5~PB7,PC10~PC11 |

.2芯片MIC3975YM8 M

（1）芯片MIC3975YMM介绍

该MIC3975是750毫安低压差线性电压调整器提供低电压，高电流输出从

非常小的封装。是一个8引脚芯片.采用麦瑞半导体专有的超级βetaPNP ™传输元件，该MIC3975提供极低差（一般为300mV ，在750毫安）和低接地电流（通常6.5毫安在750毫安） 。该MIC3975非常适合需要CON组PC附加卡VERT从标准的5V到3.3V或3.0V ， 3.3V至2.5V或2.5V至1.8V或1.65V 。保证最大压差为500mV的在所有工作条件允许的MIC3975从电源可低至3.0V和1.8V或2.5V提供从1.65V的电源可低至2.25V 。该MIC3975得到充分的保护与过电流限制，热关断，扭转电池保护。固定5.0V ， 3.3V ， 3.0 2.5V ， 1.8V ，1.65V和对电压可用能。可调输出电压选项可用于电压下降到1.24V 。

（2）芯片MIC3975YMM特点

* 固定和可调电压输出为1.24V
* 300mV的典型压差在750毫安
* 非常适用于3.0V至2.5V的转换
* 非常适用于2.5V至1.8V或1.65V转换
* 稳定的陶瓷电容
* 750毫安最低保证输出电流
* 1％的初始精度
* 低接地电流
* 电流限制和热关断
* 反向电池保护
* 反向漏电保护
* 快速瞬态响应

（3）芯片MIC3975YMM应用

* 光纤模块
* LDO适用于PC线性稳压器外接卡
* 的PowerPC ™电源
* 高英法fi效率线性电源
* SMPS后置稳压器
* 多媒体和PC处理器供应
* 电池充电器
* 低电压微控制器和数字逻辑

（4）芯片MIC3975YMM电路图



.3芯片PL2303SA

(1) 芯片PL2303SA介绍

该芯片是一个USB转串口RS232桥控制器。作一个USB端口和一个标准RS232串口之间的桥梁。两大容量片上缓存容纳从两个不同的总线数据流。该USB大容量型数据采用最大数据传输。自动握手是在支持串行端口。与它们相比，要高得多波特率可以相比于传统的UART控制器来实现。该器件还符合USB电源管理和远程唤醒方式。只有最低功率从期间暂停主机消耗。通过采用SOIC -28封装集成所有的功能这一点，芯片是适合于电缆嵌入。用户只需简单的勾线插入PC或集线器上的USB端口，然后它们可以连接到任何RS-232设备。

(2) 芯片PL2303SA特点

* 完全符合USB规范V1.1和USB CDC V1.1
* 支持RS232串行接口
* 支持自动握手模式
* 支持远程唤醒和电源管理
* 256字节的缓冲液各为上行和下行数据流
* 支持默认ROM或外部EEPROM进行设备配置
* 片上USB收发器
* 片上晶体振荡器的12M赫兹运行
* 支持Windows 98 / SE ， ME ， 2000 ， XP中，Windows CE3.0 CE .NET ， Linux和Mac OS

(3) 芯片PL2303SA原理图



.3芯片SN65HV32

（1）芯片SN65HV32介绍

该芯片是一个8引脚的CAN收发器，是一个高输出RS -485收发器。该SN65HVD05 结合三态差分线路驱动器和差动线路接收器。它们被设计为平衡数据传输和互操作ANSI TIA/EIA-485-A和ISO 8482E符合标准的设备。该驱动器设计以提供比差分输出电压大于这需要通过这些标准增加噪声利润率。该驱动器和接收器具有高电平有效和低电平有效使分别，它可以是外部连接在一起，以用作方向控制权。驱动器差分输出和接收器差分输入内部连接以形成一差分输入/输出（ I / O）总线端口即旨在提供司机被禁用或不供电。这些器件具有宽的正面和负面的共模电压范围。

（2）芯片SN65HV32特点

* 2.5 V最小差分输出电压
* 进入54 Ω负载
* 开路，短路和空闲总线接收器无故障
* 1/8th单位负载选项可用（最多256个在总线节点）
* 总线引脚ESD保护超过16 kV的HBM
* 驱动器输出压摆率控制选项电的兼容ANSI
* TIA / EIA- 485 -A标准
* 低电流待机模式， 1 μA典型
* 无毛刺上电和掉电保护热插拔应用
* 引脚兼容行业标准SN7517

（3）芯片SN65HV32应用

* 数据传输长或有损线
* 或电气噪声环境
* PROFIBUS总线接口
* 工业过程控制网络
* 点销售终端（ POS ）网络
* 电力计量
* 楼宇自动化
* 数字电机控制

（4）芯片SN65HV32原理图



[三.软件介绍](#三软件介绍)

笔记软件

1使用OneNote 管理自己做毕设时的笔记，将一些知识点，易忘内容，提醒事项记录在内，可以手机，ipad，PC同时在线记录。

2有道云笔记支持Markdown，Markdown是一种轻量级的「标记语言」，通常为程序员群体所用，目前它已是全球最大的技术分享网站 GitHub 和技术问答网站 StackOverFlow 的御用书写格式。

利用Markdown可以做什么？

1.代码高亮

2.制作待事项To-do List

3.高效绘制 流程图、序列图、甘特图、表格。方便我们梳理设计毕设流程图。

4.书写数学公式

Markdown常用代码：制作流程图时百度使用

3XMind 是一款非常实用的商业[思维导图](https://baike.baidu.com/item/%E6%80%9D%E7%BB%B4%E5%AF%BC%E5%9B%BE)软件

|  |  |  |  |
| --- | --- | --- | --- |
| XMind不仅可以绘制思维导图，还能绘制 | | | [鱼骨图](https://baike.baidu.com/item/%E9%B1%BC%E9%AA%A8%E5%9B%BE)、二维图、  [树形图](https://baike.baidu.com/item/%E6%A0%91%E5%BD%A2%E5%9B%BE)、  [逻辑图](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E5%9B%BE)、  [组织结构图](https://baike.baidu.com/item/%E7%BB%84%E7%BB%87%E7%BB%93%E6%9E%84%E5%9B%BE)（Org、Tree、Logic Chart、Fishbone）。并且，可以方便地从这些展示形式之间进行转换。 |
| 可以导入 | MindManager、FreeMind数据文件。灵活的定制节点外观、插入图标。丰富的样式和主题。输出格式有：HTML、图片 | |

|  |  |
| --- | --- |
| XMind 的文件可以导出成 | Microsoft Word、Microsoft ，  [PowerPoint](https://baike.baidu.com/item/PowerPoint)、  [PDF](https://baike.baidu.com/item/PDF)、图片（包括PNG、JPG、GIF、BMP 等）、  [RTF](https://baike.baidu.com/item/RTF)、  [TXT](https://baike.baidu.com/item/TXT)等格式， |
| Table | 子标题 |
| Enter | 同级标题 |
| 空格 | 输入文字 |

git

在我们编写程序的时候，需要不断修改程序，并且可能出现我们要借鉴使用昨天的程序，所以得经常保存不同版本的程序，直接在文件夹里面复制粘贴费事，而且占用大量存储空间，因此我们这里用到了git。

Git版本控制系统可以告诉你每次的改动，比如在第5行加了一个单词“Linux”，在第8行删了一个单词“Windows”。而图片、视频这些二进制文件，虽然也能由版本控制系统管理，但没法跟踪文件的变化，只能把二进制文件每次改动串起来，也就是只知道图片从100KB改成了120KB，但到底改了啥，版本控制系统不知道，也没法知道。

除此之外，我们可以在在GitHub上免费托管Git仓库，因github是国外网站，进入可能需要时间等待，因此我们也可以使用国内的Git托管服务——[Gitee](https://gitee.com/?utm_source=blog_lxf)。

在使用git控制不同版本程序的时候我们经常使用一下命令码。

.1使用git时常用的基本操作

|  |  |
| --- | --- |
| 目录 |  |
| Pwd | 显示当前路径 |
| ls | 列出当前目录中的所有文件 |
| reset | 清屏 |
| 改变路径 |  |
| cd e:\GitRepository | 改变当前目录到\* |
| cd .. | 退回到上一个目录 |
| 文件/夹操作 |  |
| mkdir: | 新建一个文件夹 |
| touch : | 新建一个文件 |
| rm: | 删除一个文件 |
| mv | 移动文件,；mv 目标文件夹 目标文件 |
| 版本库操作 |  |
| Git init | 把当前目录文件夹设置为版本库 |
| git add readme.txt | 把.txt文件添加到版本库里面，可多次使用，最后使用一个commit提交 |
| $ git commit -m "this change explain" | 把本行代码上述添加到版本库里的文件一次性提交 |
| $ git status | 查看仓库文件是否修改、是否添加提交 |
| $ git diff | 查看仓库文件修改的地方 |
| $ git log | 查看某文件过去修改的不同版本日志。log：日志 |
| $ git reset --hard ID号前几位 | 把文件恢复到 某个版本，新版本/旧版本 |
| $ cat readme.txt | 查看文本内容 |
| $ git reflog | 查看用户曾经的命令，可显示ID号，方便更改版本。ref参考 |
| $rm test.txt | 删除工作区的某个文件 |
| $ git rm test.txt | 删除版本库中的某个文件 |
| $ git remote rm origin | 删除远程库origin |

.2Git bash分支命令码

|  |  |
| --- | --- |
| 分支 |  |
| $ git checkout -- readme.txt | 让工作区文件撤销恢复到最近一次git commit或git add时的状态，是哪种状态由文件是否复制到了暂存区stage |
| $ git checkout -b dev origin/dev | 本地git创建的dev分支  与github里origin仓库里的dev分支对应 |
| =$git branch dev  $git checkout dev | $创建dev分支  $切换到dev分支 |
| $git checkout master | 切换到master分支 |
| $git branch | 查看分支 |
| $ git branch -d dev | 删除分支dev |
| $ git branch -D dev | 删除未提交的分支（分支修改了半天不用它了） |
| $ git merge dev | 把分支dev合并到当前分支  merge：合并 |
| $ git merge --no-ff -m "merge with no-ff" dev | 合并分支要创建一个新的commit，所以加上-m参数，把commit描述写进去。不使用快进合并 |
| git log --graph | 查看分支合并图。 |
| Git stash | 当前文件修改之后没有add到暂存区，没有commit提交时用本行代码把工作现场储藏起来，储藏到stash |
| git stash list | 查看存放到stash的工作现场 |
| Git stash apply | 把stash里的工作现场恢复到曾经的路径 |
| Git stash drop | 把stash里的内容删除 |
| git cherry-pick ID号 | 把某次提交的修改复制到当前分支上。写32的时候经常用到，在主控板做了修改，驱动板也要做同样的，直接用本行代码 |

.3Git bash标签命令码

|  |  |
| --- | --- |
| 打标签 |  |
| Git tag v1.0 | 默认在最新提交的commit上打上标签v1.0 |
| Git tag v0.9 f52c633 | F52c633 需要打标签的ID号 |
| $ git tag -a v0.1 -m "version 0.1 released" 1094adb | -a指定标签名  -m指定说明文字 |
| Git tag -d v1.0 | 删除标签 |
| 查看标签 |  |
| Git tag | 查看标签标签按字母排序，不是发布时间 |
| git show v0.9 | 查看标签信息 |

MDK介绍

本次毕设MDK版本为MDK5.29，该版本使用 uVision5 IDE 集成开发环境，是目前针对 ARM 处理器的最新版本，是 Cortex M 内核处理器的良好开发工具。

MDK5 由两个部分组成：

|  |  |
| --- | --- |
| MDK Core |  |
| Software Packs | 可以独立于工具链进行新芯片支持和中间库的升级。 |

MDK Core 又分成四个部分：

|  |  |
| --- | --- |
| uVision IDE with Editor | （编辑器） |
| ARMC/C++ Compiler | （编译器）， |
| Pack Installer | （包安装器） |
| uVision Debugger with Trace | （调试跟踪器） |

Software Packs（包安装器）又分为三个小部分，：

|  |  |
| --- | --- |
| Device | （芯片支持）， |
| CMSIS | （ARM Cortex 微控制器软件接口标准） |
| Mdidleware | （中间库）通过包安装器， |

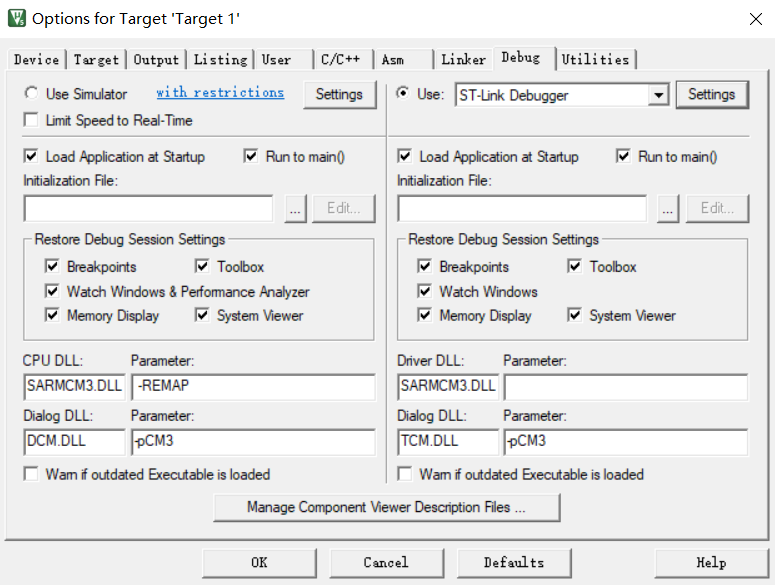
我们可以安装最新的组件，从而支持新的器件、提供新的设备驱动库以及最新例程等，加速产品开发进度

MDK下载安装

1. 在 [www.keil.com](http://www.keil.com) 官网 下载MDK5.29软件，并下载芯片的开发包Keil.STM32F1xx\_DFP
2. STM32F1开发有寄存器版本,HAL库版本，标准库版本。HAL 库和标准库  
   本质上是一样的，都是提供底层硬件操作 API，而且在使用上也是大同小异。这里选用标准库版本。

程序下载方式

STM32F1的程序下载方式有USB、串口、 JTAG、 SWD 。但串口只能下载代码，并不能实时跟踪调试，而JTAG占用IO口线较多，所以这里选择SWD下载，选择调试工具ST-LINK。除此之外还有 JLINK 、ULINK 等调试工具可供选择。

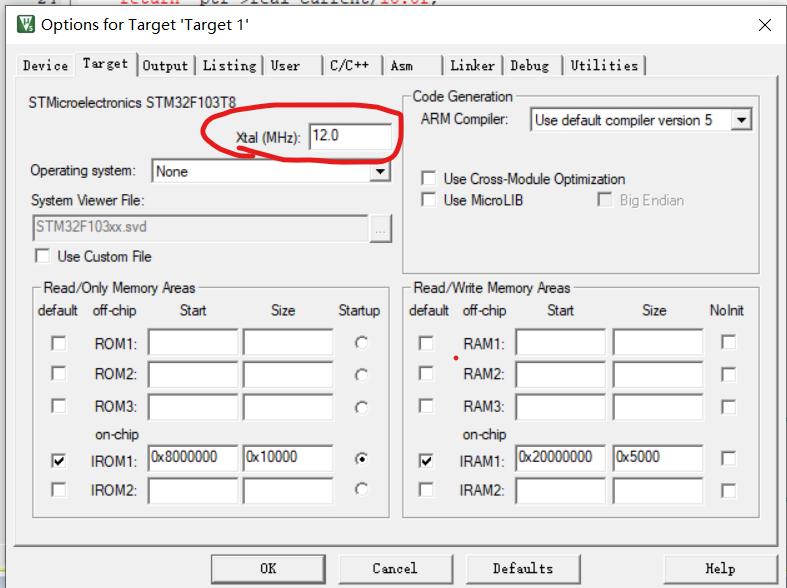


程序仿真调试

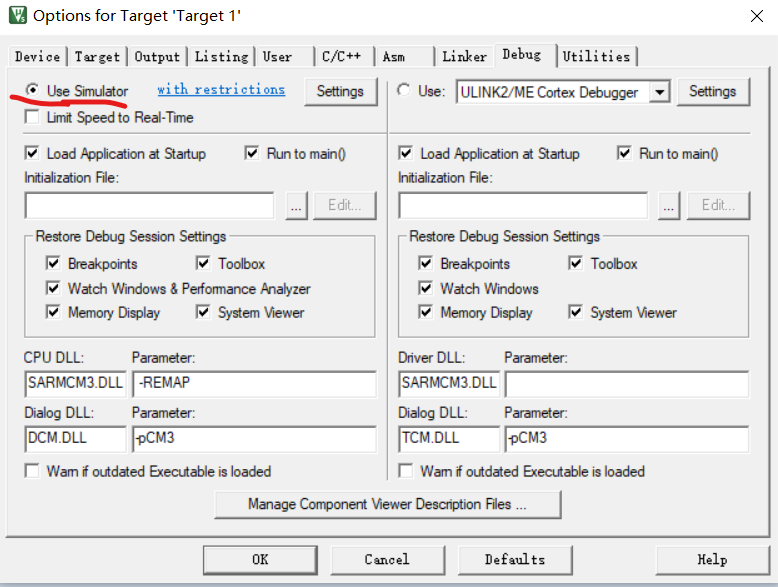
|  |  |
| --- | --- |
| 软件仿真调试 | 无实物，纯软件和代码调试,  只能调试简单的I/O口操作，高电平低电平简单的看了出来 |
| 硬件仿真调试 | 通过下载器与电路板连接起来，可调试复杂外设BUG |
|  | 看到代码在单片机内部里面如何运行、变量是如何传递、变量的变化情况等等。 |

MDK仿真调试步骤

1.进入Target,输入单片机芯片所用的晶振：



2.进入Debug,选择 Use Simulator



3.仿真无法运行，修改在debug下面配置成如下参数

|  |  |  |  |
| --- | --- | --- | --- |
| SARMCM3.DLL | -REMAP | SARMCM3.DLL | -REMAP |
| DARMSTM.DLL | -pSTM32F103T8 | TARMSTM.DLL | -pSTM32F103T8 |

|  |  |
| --- | --- |
| SARMCM3.DLL | 是一个动态链接库，SARMCM3表示是ARM Cortex-M3架构 |
| DARMSTM.DLL、TARMSTM.DLL | 指明了是STMicroelectronics（意法半导体）的ARM架构的产品。 |
| -pCM3 | 表示Cortex-M3系列的参数，由于Cortex-M3只是个大框架，具体的厂商实现细节不一样，所以得修改的具体点。 |

4.

点击调试按钮：

C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\17B55C2C.tmp

5.调试方法

|  |  |
| --- | --- |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F1930F1A.tmp | 黄色的箭头代表的就是程序当前运行的位置 |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F74BB638.tmp | RST（复位），单击之后，程序就会跑到最开始的位置运行，即可回到main函数的开头处 |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\5AD0D106.tmp | 全速运行，单击之后程序就会全速跑起来 |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\42618B04.tmp | 停止按钮，当程序全速运行起来时候，单击停止按钮程序就会立即停止，可以观察程序运行到哪里去了。 |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A5539BB2.tmp | Step，单步执行，当碰到子函数时，则进入子函数。 |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\3D2F2690.tmp | Step Over:单步执行，碰见子函数时不进入，将子函数当作一个整体来运行。 |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\B5959B1E.tmp | Step Out；单步执行，程序若在子函数内部执行，则跳出子函数。 |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\442B94DC.tmp | Run to Cursor Line:运行到光标处。 |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\FE7DBB4A.tmp | Serial Windows :串口输出窗口。 |
| C:\Users\lenovo\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\CE98A1E8.tmp | Analysis Windows；逻辑分析窗口，该窗口下有Logic Analyzer，可以查看波形、变量等数值的观察窗口、 程序运行的时间。 |
| 断点 | 单击复位之后，可以看到C语言程序的窗口左侧有灰色或者保持着原来的颜色，其中有灰色的地方是我们可以设置断点的地方，不能设置断点的地方是因为keil具有程序优化的功能，如果我们想在任何地方都可以设置断点的话，需要更改工程选项里的优化等级，把等级设置为0即可，即高速keil不要对程序进行优化。 |
| 计算单片机运行了多长时间 | 设置断点和查看sec，sec显示了单片机运行的时间。sec位于Registers窗口里面 |
| 查看某一个变量或者寄存器的值, | 点击view-> watch windows,在里面的name选项输入想要查看的变量的名字或者寄存器的名字，随着程序的运行，我们就可以实时看到他们的值的变化 |

四.软件介绍

STM32F103ZET6

主控程序

主控程序用到的模块：

1

CAN1初始化：CAN一次性传送一个ID数据时，可以传送8帧数据

can1接收中断程序：

接收5个扇形装甲板的击打信息：

ID:0x301-305

Data[0]：0未击中 1击打正确 2击打错误

接收风车电机的信息:

ID:0x204

Data[0]Data[1]：编码器机械角度

Data[2]Data[3]：速度\_每分钟转数

Data[4]Data[5]：实际电流值

Data[7]：电机温度

2.

USART1初始化

USART1接收中断：

仅接收颜色数据：红色/蓝色

3.

TIM2初始化：移位定时器

TIM3初始化：计算时间击打能量机关

定时器定时时间计算：

定时器时钟频率f=时钟源频率/(预分频值+1)=72MHz/ (psc+1)

定时器定时时间=(自动重装载周期值+1) / 定时器时钟频率=(arr+1) /f=(arr+1) \*t

Tim2\_Init(999,7199); // 72Mhz/7200=10000 0.0001\*1000=0.1s=100ms

Tim3\_Init(6999,7199); // 0.0001\*7000=0.7s=700ms

TIM2向上计数溢出中断：随机数种子seed++

TIM3向上计数溢出中断：计数一定时间运行模式进入acting模式，并开始time\_count++;

4.LED初始化

5.

中断优先级：抢占优先级 子优先级CAN1接收：2 0

USART1： 1 0

TIM2: 7 1

TIM3: 6 0

6.

延时函数

7.

初始化PID相关参数

循环程序：

Fail\_Task：

5个扇形上的装甲板信息清0，.ready=0，hitted=0;并通过CAN1发送，发送5个扇形 的：颜色|准备状态|运行模式 TIM3重新使能

Success\_Task：

通过CAN1发送5个扇形的：颜色|准备状态|运行模式，激活成功

TIM3重新使能

Acting任务：

刚上电，随机让一个扇叶准备

检查5个扇形装甲板的击打状态：击打错误，模式=失败

检查某个被选中的扇形装甲板的击打状态：击打成功，选择下一次的扇形装甲板并通过CAN发送5个扇形的：颜色|准备状态|运行模式

处理can接收数据：

ID:0X201

获得：PID控制实际值

PID控制计算并CAN1发送数据

ID:0x200,3508电机

Data[]:pid.out电机输出电流值

主控程序总结：

主控32 接收5个扇形装甲板的击中信息；旋转标靶电机的机械角度，速度值，电流值，温度值；灯的颜色。通过一系列处理数据，确定灯的闪亮灯，进行PID控制。并通过CAN发送5个扇形的：颜色|准备状态|运行模式；电机的PID控制电流值

主控程序IO口：

CAN1: A11,A12

USART1: A2发送A3接收

LED: A5 6 7 8 B0 1 2 5 6 7

点阵驱动板程序

程序用到的模块：

点阵驱动IO口GPIOA0-8初始化配置输出

用来读取扇形装甲板ID1-5的拨码开关初始化配置

TIM2初始化：箭头移动

TIM2中断：CAN1发送此时某个扇形装甲板的击打信息.hitted

点阵驱动LED数据移动，用于图形移动

TIM3初始化：图形累加

返回不同片上时钟的频率： 检验晶振是否正确

CAN1初始化：

can1接收中断程序：

接收ID:0X300

接收扇形击打板被击打信息

Data[0]-Data[5]:5个扇形装甲板信息

Data[]的每一位：

7 6 Colour Ready 3 ActingFailed ActingSucess Acting

按键传感器IO口以及外部中断初始化配置

外部中断程序：

通过ready和acting的状态来确定此时某个扇形装甲板的击打信息.hitted

并通过CAN1发送击打信息

获取此时拨码开关确定的扇形装甲板ID

循环程序：

看门狗喂狗

对点阵显示的控制:

刚开始：点阵驱动LED全亮

总的击打状态：击打成功点阵驱动LED全？ ，击打错误点阵驱动LED全亮

击打错误：点阵驱动LED全亮 ，运行模式清0

击打成功：TIM3使能，点阵驱动LED Up ，运行模式清0

击打进行中，没打中：TIM3使能， DataFlag=BoardFlag=0;

点阵驱动部分的程序说明

void DataShift(unsigned char Data)：

若数据为1个1、7个0这种 JP3\_SI=1；

否则 JP3\_SI=0；

void DataSelect(char SlcNumber)：

选择数字=001，JP3\_A=1

选择数字=010，JP3\_B=1

选择数字=100，JP3\_C=1

否则JP3\_A=0，JP3\_B=0，JP3\_C=0

void ring\_shift\_left(unsigned char \*data)：

调整8个数据:将每个数据的最低位调整到最高位

void DotArrayShowFull(Bit16 colour,u8 State)：

确认点阵驱动IO口的状态让点阵全亮/灭

总结：点阵驱动程序 通过CAN1中断接收点阵驱动的颜色，5个扇形装甲板的状态信息.ready、激活成功失败。通过外部中断确定5个扇形装甲板的击打信息..hitted并通过CAN1发送给灯条板和主控板，并。

程序IO口：

点阵驱动IO口：GPIOA0-8:推挽输出，置高电平

拨码开关输入IO口：GPIOB-3-6，读取扇形装甲板ID号1-5，颜色ID(颜色ID暂时不用)

传感器接口:外部中断:A9

Can1: A11can接收 A12can发送

点阵驱动IO口：

#define JP3\_A PAout(0)

#define JP3\_B PAout(1)

#define JP3\_C PAout(2)

#define JP3\_SCK PAout(3)

#define JP3\_RCK PAout(4)

#define JP3\_SI PAout(5)

#define JP3\_OE\_Blue PAout(6)

#define JP3\_OE\_Red PAout(7)

#define JP3\_OE\_Green PAout(8)

#define Blue 0

#define Red 1

灯条程序

程序模块：

CAN1初始化：CAN一次性传送一个ID数据时，可以传送8帧数据

can1接收中断程序：

接收5个扇形装甲板的击打信息：

ID:0x301-305

Data[0]：0未击中 1击打正确 2击打错误

接收信息

ID:0x300，CAN\_MASTER\_ID

Data[0]：运行模式acting = 1, //正在激活

success = 2, //激活成功

fail = 4, //激活失败

Data[1]- Data[5]：5个扇形装甲板的准备状态ready

LED初始化配置：A5 6 7 9

B0 1 2 5 6 7

循环程序：

运行模式一直检测：

Success：

5个扇形装甲板依次随机全被击打，激活成功

灯条LED 10个IO口全亮

Fail：

未能在 2.5 秒内击中随机点亮的装甲模块 /击中非随机点亮的装甲模块，激活 失败

灯条LED 10个IO口全灭；

5个扇形装甲板的.ready .hitted 全为0

Acting：

查询5个扇形装甲板的.ready .hitted信息

.ready=1 .hitted=0：里面的灯灭，外面装甲板周围的灯亮

.ready= .hitted=1：全亮

.ready= .hitted=2：全灭

灯条程序总结：

灯条主控通过CAN1接收5个扇形装甲板的准备状态ready、击打信息（是否打中，击打错误）以及运行模式正在激活acting/激活成功 success/ 激活失败fail。并实时自动运行灯条显示：(正在激活)仅装甲板外面的灯亮、（激活成功）全亮、（激活失败）全灭

程序IO口：

CAN1: A11,A12

LED: A5 6 7 9 B0 1 2 5 6 7

程序定义说明

灯条程序：

flabellum[i].ready=1;允许击打某扇形装甲板

flabellum[i].ready=0;不允许击打某扇形装甲板

flabellum[i].hitted=0;某扇形装甲板未被击打

flabellum[i].hitted=1; 允许击打的某扇形装甲板被击打

flabellum[i].hitted=2; 不允许击打的某扇形装甲板被击打