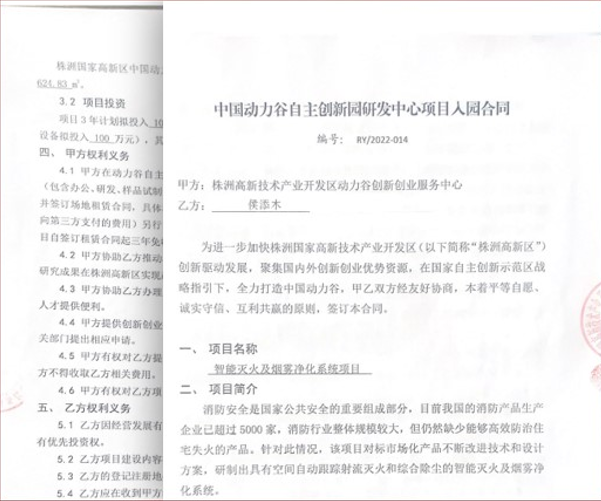
项目：雾净射流灭火器

团队概括以及成就：

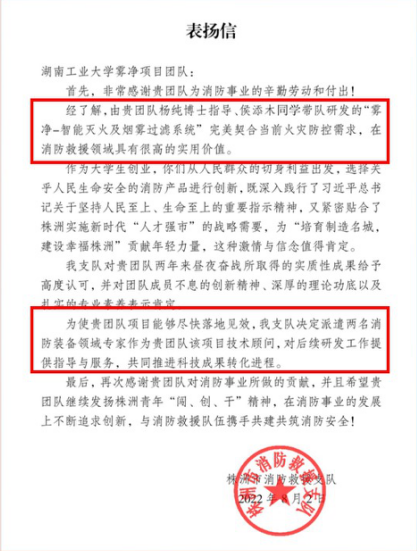
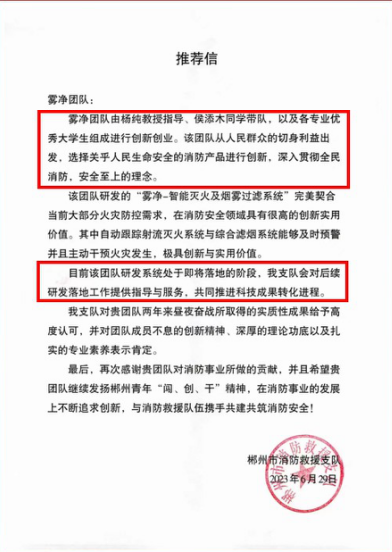
1. 项目落地株洲市高新区高科动力谷，建立600平方研发基地



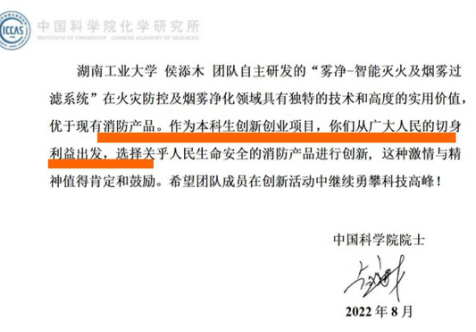
1. 与国防科技大学国科智能研究所合作将跟踪灭火系统搭载到ros车上，联合研发消防灭火机器人



1. 受到株洲消防表扬以及郴州消防推荐



1. 受到中科院赵进才院士的推荐



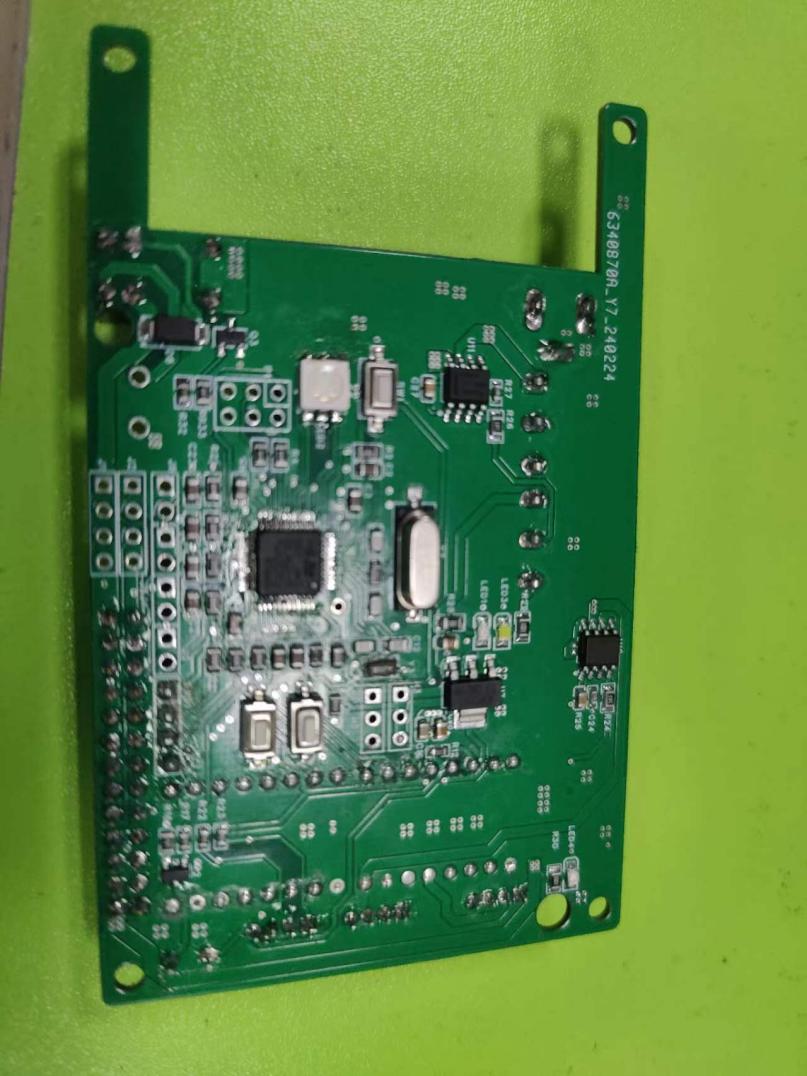
1. 一代成就







二代产品介绍：



****

主要分为：视觉算法软件设计，嵌入式开发板软件设计，嵌入式开发板设计（硬件），外观机械结构设计

嵌入式开发板软件设计：

外设：水泵（继电器控制） 减速电机 限位开关（控制电机，让电机在极限位置转向） 蜂鸣器 串口通讯（用于接收上位机坐标信号，用于调节电机对准，反馈信息给上位机）

主要程序设计：

主控制函数采取五段状态机，分别是初始状态，粗校准状态，细校准状态，灭火状态，停止状态

初始状态：检测上位机是否传来红外摄像头信号，发送反馈信号进入粗校准状态。若没有红外信号则持续巡逻（水平电机旋转直至撞到限位开关）

粗校准状态：解析上位机传来的坐标信号进行粗校准（通过上位机传来的坐标位置计算电机运动步数），直至大致校准成功（只有水平旋转）

细校准状态：解析上位机再一次传来的目标位置进行，上下左右校准

灭火状态：打开水泵，覆盖式灭火，每隔3.6秒检测一次直至灭火成功（来自上位机的信号）

结束

1. 采用的芯片主控是什么？为什么采用该主控？

采用stm32f103c8t6，该芯片生态成熟，资料多，方便开发

1. 开发平台是什么？

开发环境采用的Keill

1. 详细介绍下项目

项目是由我们雾净团队研发的，针对火灾险情的，灭火系统。采用视觉算法，检测火源，使用阻燃液射流灭火，集齐嵌入式开发板软件设计，视觉算法软件设计，嵌入式开发板设计（硬件），外观机械结构设计综合型产品级项目

1. 这个项目主要是解决什么问题/完成什么事？

市场上大多灭火装置都存在需要人为操作，不能及时处理的问题，我们的产品针对此类问题设计，全天候火灾检测，在火灾萌芽阶段全自动检测对准火焰灭火，有效解决无人看守问题

1. 你在这个项目中的职责是什么/做啦哪些事？

为主要负责电机的测试以及下位机程序设计，下位机主要包括水泵、减速电机、限位开关、蜂鸣器和串口通信，主要通过五段状态机实现任务：初始状态检测红外信号进入粗校准状态，再根据坐标信号进行细校准，启动水泵进行灭火直至成功，最终结束任务

1. 这个项目中用到了那些技术/框架？

外设控制：使用继电器控制水泵，控制减速电机和限位开关，通过下位机引脚控制（继电器类似于LED灯这种，限位开关类似于按键，减速电机就是产生PWM波控制）

（GPIO控制主要分为管脚使能，以及管脚高低电平设置）

(详细内容)电机模块：采用DRV8825驱动，控制步进电机（这是硬件配置）

1. 初始化：初始化整个电机系统，包括水平方向和垂直方向的电机初始化，包括配置电机方向控制引脚、初始化 PWM 输出和停止电机运动
2. 电机控制：控制函数接受三个参数：模式（mode）、方向（dir）和 PWM 值（pwm），模式是决定是哪一个电机，PWM（脉冲宽度调制）被用来控制电机的转速从而控制电机旋转步数
3. PWM（脉冲宽度调制）：冲宽度增大时，高电平持续时间增加，平均电压也相应增大；反之，当脉冲宽度减小时，平均电压减小。通过这种方式，可以控制被 PWM 信号控制的设备的平均功率。在电机控制中，PWM 被广泛应用来调节电机的转速。通过改变 PWM 的占空比，可以改变电机的平均电压，从而调节电机的转速。较高的占空比会导致电机以更大的平均电压运行，因此转速更快；而较低的占空比则会导致电机以更小的平均电压运行，因此转速更慢（太官方了，可以简略一点）
4. PWM控制实际操作：使用两路定时器分别控制水平垂直两个方向的电机，在定时器中断（上升沿进入中断，每一次高电平都会进入）里检测 PWM 是否达到设定的目标值，从而控制运动步数，在中断控制内加入限位开关检测，从而避免了到达边界连续而判断不及时问题
5. 定时器 1计时功能： 定时器可以用于计算时间的流逝，通常以微秒、毫秒或秒为单位。通过配置定时器的预分频器和计数器，可以实现不同的计时精度和范围 2. 中断生成： 定时器可以设置为在特定的时间间隔内生成中断信号 3. PWM 生成： 定时器常用于产生脉冲宽度调制（PWM）信号。PWM 是通过定时器的比较功能和输出控制来实现的，可以生成具有可调节占空比的脉冲信号。4. 输入捕获： 定时器可以用于捕获外部事件的时间戳，比如测量脉冲宽度、计算输入信号的周期等 5. 输出比较： 定时器可以与比较器结合使用，比较计数器的值和预设的比较值，并在匹配时执行某些操作，如产生中断、控制输出电平等 6.脉冲计数： 定时器可以用于计算输入脉冲的数量
6. Stm32 定时器：基本构成： STM32 的定时器通常由计数器、预分频器、比较器、自动重载寄存器等组件构成。1.计数器： STM32 定时器的计数器可以是 16 位或 32 位的。计数器以固定的时钟频率进行计数，可以通过配置时钟分频来控制计数速度。2. 预分频器： 预分频器用于降低计数器的时钟频率，以扩展计数范围或降低计时精度。预分频器可以配置为不同的分频比例。3. 比较器和自动重载寄存器： 比较器用于比较计数器的值和预设的比较值，当匹配时触发中断或其他操作。自动重载寄存器用于存储比较值，通常与比较器配合使用，形成定时器的定时功能。4.输入捕获和输出比较： STM32 定时器还支持输入捕获和输出比较功能，用于捕获外部事件的时间戳或比较计数器值并执行相应的操作。5.中断控制： 定时器可以配置为在特定的计数值达到时生成中断请求。中断控制器会检测计数器的值，并在匹配时产生中断信号，通知处理器执行相应的中断服务程序。6.PWM 生成： STM32 定时器可以用于产生 PWM 信号，通过配置比较器和输出控制逻辑，可以生成具有可调节占空比的 PWM 输出信号。7.时钟源： 定时器的时钟源可以是内部时钟或外部时钟，可以通过设置寄存器来选择时钟源和时钟分频（理论知识大概了解就行，我觉得）

状态机设计： 采用五段状态机设计来管理系统的不同工作状态

分别是初始状态，粗校准状态，细校准状态，灭火状态，停止状态

初始状态：检测上位机是否传来红外摄像头信号，发送反馈信号进入粗校准状态。若没有红外信号则持续巡逻（水平电机旋转直至撞到限位开关）

粗校准状态：解析上位机传来的坐标信号进行粗校准（通过上位机传来的坐标位置计算电机运动步数），直至大致校准成功（只有水平旋转）

细校准状态：解析上位机再一次传来的目标位置进行，上下左右校准

灭火状态：打开水泵，覆盖式灭火，每隔3.6秒检测一次直至灭火成功（来自上位机的信号）

结束

串口通信： 使用串口通信与上位机进行数据交换，接收坐标信号和发送反馈信息

接收来自上位机传来的十六进制数据包，格式是00 01 01 0d 0a，最后两位为校验位，以十六进制0d 0a结尾，00是方向位，中间是数据位（了解就行）（上位机处采用YOLOV8，rk3588部署识别到的坐标框，经过处理转为16进制数据包发送给下位机）

解析函数设计：当接收到 USART 数据时，首先将接收标志位置零，表示接收到数据后将不再处理。然后根据接收到的数据长度，提取出 Y 和 X 坐标值，存储到全局变量中，并设置解析完成标志位为 1。如果接收到的数据格式错误，则将解析完成标志位清零，并重新开始接收

发送反馈信息给上位机，以控制系统的行为

1. 这个项目中大家是怎么协作的？

在软件程序设计方面，需要密切的和上位机算法设计的同学交流，在上位机计算速度上进行优化讨论，可以通过一定程度上下位机的电机调节来解决此问题。在硬件设备上，由于采用自我设计的开发板，难免会遇到部分硬件器件问题，和硬件同学及时反馈，修理器件问题。在结构设计方面，和机械设计同学沟通需要预留的位置，孔位以及散热发热等问题

1. 这个项目主要的挑战是什么？

对于嵌入式软件设计来讲，主要的挑战就是电机调节问题，需要满足对准火源，还需要考虑到上位机信息反馈不及时的问题（该问题已经解决，上位机算法从传统视觉算法升级到深度学习算法，识别速度大幅度提升）

1. 你是怎么解决的？

通过长时间的调节，改变运动步数，一直修改直到找到合适的状态

1. 在这个项目中你学到了什么？
2. 学习到了单片机的编程能力。对一系列的外设的使用（直接说上面那些）
3. 学习到了团队沟通能力。开发过程中出现的问题非常非常的多，小到函数变量名字的设定，大到一整个思路的否定，在不断沟通和讨论中进行思路梳理
4. 了解到许多其他领域的问题，例如上位机的部署环境问题，机械的流体动力学问题，如何让火浇灭，项目的成本控制以及实用性问题等

（自己编一下不知道怎么总结这个）

1. 为什么要关注/学习这些技术？

本科专业是信息工程，在课程中接触到该技术，感觉学习 STM32 开发能够为涉及到多个领域的提供丰富的就业机会和发展前景，STM32 生态系统相对完善，包括官方提供的开发工具、文档、示例代码以及丰富的第三方库和社区支持，比较好入门，自身对其也比较感兴趣

1. 你喜欢什么样的开发环境？

由于使用Keil比较多，还是更加倾向于Keil

1. 你喜欢用什么操作系统、IDE、浏览器、调试工具、版本管理工具.......？

目前来讲还是较为喜欢Windows，因为使用较多图形化操作简易，但是对Ubuntu、Centos等Linux环境也有所了解（Linux环境肯定是比较好的对于这个，提到但是不细说）

IDE：集成开发环境（IDE）是一种帮助程序员高效开发软件代码的软件应用程序

喜欢Keil，但是对VScode也经常使用（VScode是Windows最NB的IDE所以要提一嘴，但是说不好就提一下不细说）

浏览器平时使用的主要是谷歌浏览器（这个应该是考核资料查询啊这些东西，Google肯定是最好的）

ST提供的ST-Link调试器或J-Link调试器来调试STM32程序。

版本管理工具喜欢用git（公司里面肯定有版本管理这个东西，说不好就直接说VScode里面有gitlens图形化插件）

# git 命令操作

## 文件基础操作

1. git init 创建.git文件

2. git remote -v 查看是否连接到仓库

## 分支操作

1. git fetch 更新分支图

2. git checkout -b zlzyang 创建新分支

3. git push origin zlzyang:zlzyang 把新建的本地分支push到远程服务器，远程分支与本地分支同名

4. git branch 查看当前本地分支

5. git branch -r 查看远程的所有分支

6. git branch -m master 将本地分支切换到master分支

7. git branch -d localBranchName 删除本地分支

8. git push origin --delete remoteBranchName 删除远程分支

## 推拉操作

1. git pull origin develop 拉develop分支

2. git push origin master 会直接推到master分支

3. git pull origin zlzyang:master 合并到master分支

4. git push origin master:develop 将本地的master分支合并到develop仓库分支

-------

1. 你觉得你和其他开发相比，有哪些优势/劣势？

具有项目实践能力（这个编一些，现场发挥）

专业问题

<https://blog.csdn.net/youzhangjing_/article/details/131536091>

(我的评价）

## **一、STM32F1和F4的区别？**

（了解吧，感觉不是特别重要）

## **二、介绍一下STM32启动过程**

（我也不知道是什么，了解一下）

## **三、介绍一下[GPIO](https://so.csdn.net/so/search?q=GPIO&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/youzhangjing_/article/details/_blank)**

（非常重要）

## **四、UART**

（非常重要）

## **五、I2C**

（重要，但是可以不完全掌握）

## **六、SPI**

（不那么重要，可以不完全掌握）

## **七、CAN**

(了解)

## **八、DMA**

（非常重要）

## **九、中断**

（非常重要）

## **RTOS的任务是怎么写的？如何切出这个任务？**

## **UCOSII中任务间的通信方式有哪几种？**

## **十三、项目使用了自定义协议，是什么结构？**

（了解，这个是技术性问题比较难，我觉得不会也没啥）

## **十四、uCOSII和Linux的差异？**

（了解Linux）

## **十五、Git提交代码**

（重要，但是可以投机取巧）

## **ucosii和ucosiii和freeRTOS比较**

## **十七、低功耗模式**

## **十八、物联网的架构**

## **十九、内存管理**

## **二十、Ucos中任务状态哪几种？任务状态之间的关系图？**

（了解，这个是技术性问题比较难，我觉得不会也没啥）

## **二十一、ADC**

（非常重要）

## **二十二、系统时钟**

（非常重要）

## **二十三、HardFault\_Handler处理**

## **二十四、TTS语音合成方法**

（了解）

## **二十五、定时器**

（非常重要）

## **二十六、状态机**

（非常重要）

## **二十七、器件选型**

（了解）