

智绎心声——基于 STM32H7 的失语症患者辅助设备

队伍名：屏风吗喽队

队员：韩意 任效民 崔屿杰

指导老师：张志明 徐和根

摘要

本项目是为失语症患者设计的辅助设备，旨在帮助他们与家人更好地沟通。通过前期调研，我们发现失语症患者因中风等原因难以正常表达，给家人的理解和照顾带来障碍。因此，我们开发了基于 STM32H7 系列单片机的嵌入式设备，具备高计算性能和丰富的接口。该设备集成了多传感器和人工智能技术，实现了三大功能：1) 通过倾斜设备读取角度辅助患者选择屏幕显示的需求项；2) 利用红外传感器辅助患者确认选项和呼叫家属；3) 通过摄像头图像识别物体，帮助患者表达指认物体。此外，我们还进行了 PCB 设计和 Rhino 建模外壳，优化了硬件设计，使设备更易于患者使用。

设备的操作简便，通过倾斜和拂过红外传感器进行选项确认，提供震动反馈，以及播放音乐呼叫家属等交互方式，易于患者和家属理解，促进双方交流。技术特点包括设备倾斜控制选项、红外传感器确认选择、震动反馈、简易交互和物体识别辅助找东西。性能指标涵盖内存、处理器频率、电源、工作温度、屏幕分辨率和刷新率、蜂鸣器音量等。

在系统组成和功能说明部分，详细介绍了系统框图、硬件和软件的设计细节。硬件方面，包括单片机、摄像头、显示器、陀螺仪、蜂鸣器、震动马达、红外传感器等元件及其连接方式。软件方面，包括主循环的两大状态、各模块的流程图和关键变量。特别是 X-CUBE-AI 的使用，实现了神经网络模型在 STM32 设备上的高效推理。

最终完成的设备在机械设计、电路设计和软件界面方面均取得了成果，

并通过实物照片进行了展示。在总结部分，报告提出了设备的可扩展性和研发过程中的心得体会。

第一部分 作品概述

1.1 功能与特性

屏幕上三个选项，通过左右倾斜设备，陀螺仪会读取角度，切换不同的带选项。患者通过拂过（遮挡后松开）红外传感器，进行确认选项。相比手机等设备的点触，不容易选错选项，对于难以精准控制自己肢体动作的患者更加友好，操作的鲁棒性更强。并且本身握持的设备，进行倾斜设备和手指拂动的操作，更加简便。

进行确认选项后，设备震动给予患者“选择成功”的反馈，利于患者适应操作流程。

选择“吃饭/喝水、上厕所”选项时，会播放特定的音乐呼叫家属，屏幕界面有对应显示。家属或患者可以再次轻拂过红外传感器，关闭音乐。

而选择“找东西”选项时，设备调用设备底部的摄像头，用户抬起设备像“望远镜”一样找东西，屏幕显示画面与内置的 CUBE-AI 物体识别结果，辅助患者寻找辨别东西。找到目标时，同样可以呼叫家属。

交互简易，易于患者与家属的理解，可大大促进双方的交流，免于说话交流的不便。

1.2 应用领域

本设备的设计旨在应用于失语症患者的日常生活中，通过先进的嵌入式技术和人工智能功能，显著改善他们与家人、护理人员之间的交流和互动体验。失语症是一种由中风等因素引起的语言障碍，严重影响患者正常表达和理解能力，从而增加了他们生活中的沟通难度和社交挑战。本设备通过创新的设计和集成，为失语症患者提供了以下关键应用领域。

设备通过倾斜控制和红外传感器确认选择，帮助失语症患者简化日常交流和需求表达。利用摄像头进行物体识别，帮助患者更轻松地进行表达和指认周围的物品。配备震动反馈和音乐呼叫功能，增强交互体验的直观性和亲和力。

1.3 主要技术特点

1. **倾斜控制和红外传感器确认选择：** 患者通过倾斜设备或拂过红外传感器确认屏幕上的选项，简化操作流程，提升选择准确性和用户体验。
2. **摄像头物体识别技术：** 设备能够通过摄像头实时识别周围物体，帮助患者表达和指认需求，提高交流效率和准确性。
3. **震动反馈和音乐呼叫功能：** 设备通过震动反馈确认操作成功，并通过播放音乐呼叫家人或护理人员，增强交互体验和用户安全感。
4. **高性能嵌入式处理器和丰富接口：** 基于 STM32H7 系列单片机，具备高计算性能和丰富的接口，支持复杂的算法运算和多种外部设备连接。
5. **简易交互设计和用户友好界面：** 设备操作简便，界面设计直观，适合失语症患者和家属使用，促进交流和理解。
6. **硬件优化设计：** 包括 PCB 设计和外壳建模，优化设备的物理结构和外观设计，提升耐用性和易用性。

1.4 主要性能指标

指标	性能
内存	Flash Memory: 2MB; SRAM: 1.4MB
处理器频率	280MHz
电源	锂电池；容量：900mAh; 工作电流：0.9±0.2A
工作温度	10~55℃
屏幕分辨率	320*240
屏幕刷新率	3 帧/秒
蜂鸣器音量	75dB

1.5 主要创新点

1. **设备倾斜控制选项:** 避免传统触屏操作的误选问题,适合难以精准控制肢体动作的患者。
2. **红外传感器确认选择:** 操作简便且不易出错,适合患者使用。
3. **震动反馈:** 设备震动反馈选择成功,帮助患者适应操作流程。
4. **特定选项的简易交互:** 选择“吃饭/喝水、上厕所”选项时播放音乐呼叫家属,屏幕显示对应信息。
5. **物体识别辅助找东西:** 选择“找东西”选项时,设备启动摄像头,结合 CUBE-AI 进行物体识别,屏幕显示实时画面,辅助寻找目标物体。

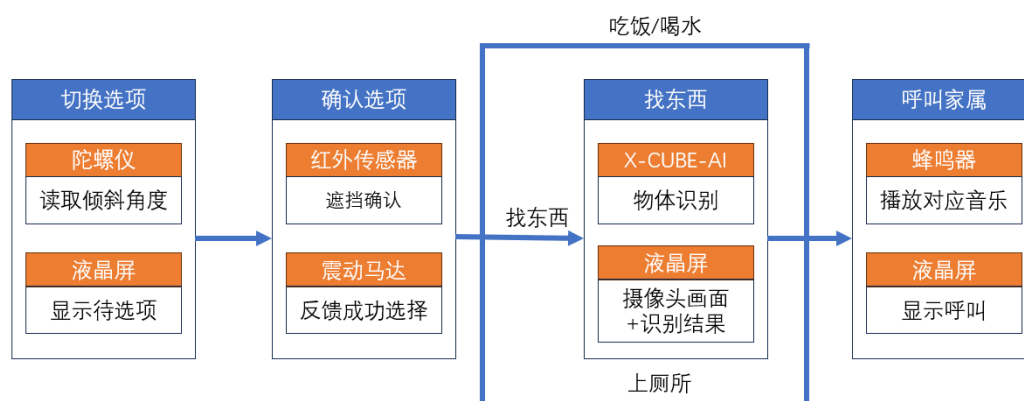
1.6 设计流程

1. **需求分析和调研:** 确定失语症患者的需求和挑战,包括交流障碍和行动能力限制的调查。
2. **系统架构设计:** 制定整体系统架构,包括硬件和软件功能划分和接口设计。
3. **硬件设计和选型:** 选择适合的单片机、传感器、摄像头和显示器,并设计 PCB 电路板。
4. **软件开发:** 开发嵌入式软件,包括状态机设计、数据处理算法、物体识别模型集成和用户界面设计。
5. **集成与测试:** 将硬件和软件模块集成,进行功能、性能和稳定性测试。
6. **优化和调整:** 根据测试结果优化设计,调整参数和界面,提升设备的使用体验和性能。

第二部分 系统组成及功能说明

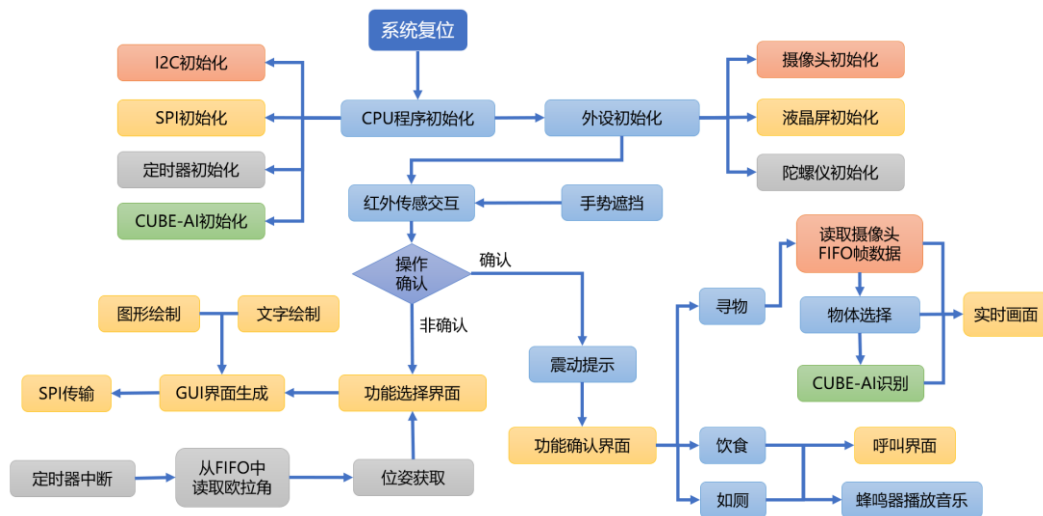
2.1 整体介绍

给出系统整体框图，各子模块标注清楚，并进行整体的文字说明，需要表达出各模块之间的关系。



系统功能框图

系统的功能框架主要分为图示的四大模块，切换选项、确认选项、找东西、呼叫家属。切换模块部分，通过读取陀螺仪角度切换液晶屏上的带选项。接下来进入确认选项模块，通过遮挡红外传感器进行确认，震动马达会震动反馈选择成功。如果是吃饭/喝水，上厕所选项，就直接进入呼叫家属选项，否则选择找东西，进入找东西模块。摄像头画面实时在液晶屏上显示，部署的 X-CUBE-AI 对于画面信息进行物体识别，并显示结果。最后都进入呼叫家属模块，蜂鸣器播放对应音乐，液晶屏显示呼叫信息。

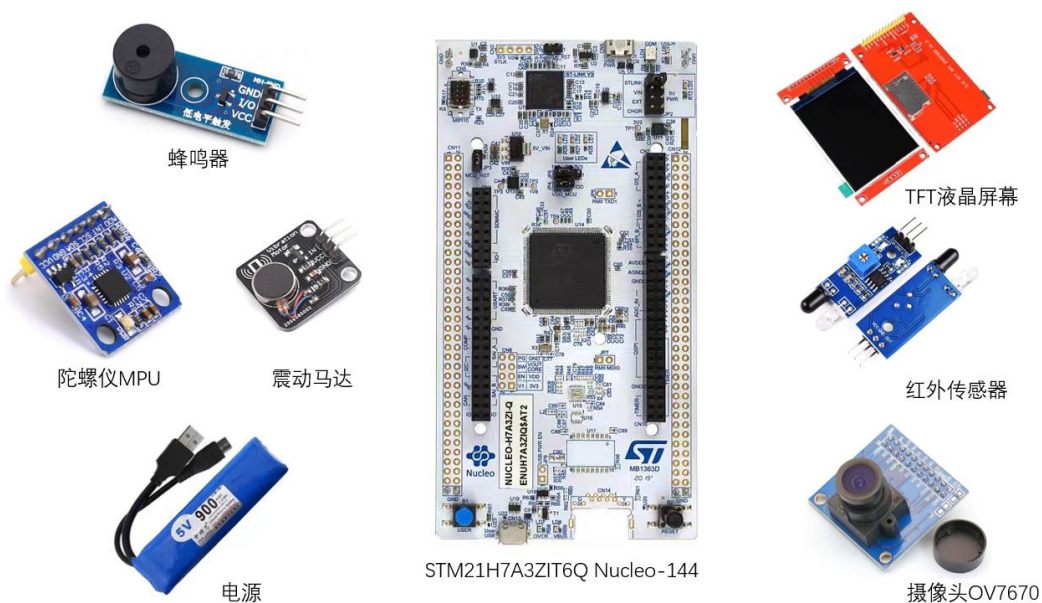


系统技术流程图

首先，系统复位后，进行 CPU 程序初始化，然后进入外设初始化阶段，包括摄像头、液晶屏和陀螺仪的初始化。同时，还需要初始化 I2C、SPI、定时器和 CUBE-AI 模块。接下来，定时器中断和从 FIFO 中读取欧拉角用于位姿获取，进入功能选择界面；与此同时，系统进入红外传感交互阶段，通过手势遮挡进行操作确认，如果确认，则进入功能选择界面，并进行震动提示；如果不确认继续停留在功能选择界面，然后，系统生成 GUI 界面，通过 SPI 传输，并进行图形绘制和文字绘制，显示于屏幕。接着，用户如果选择寻物功能，读取摄像头 FIFO 帧数据，进行物体选择和 CUBE-AI 识别，生成实时画面。用户如果通过功能选择界面进行饮食、如厕操作，将出现呼叫界面，并进行蜂鸣器播放音乐。

2.2 硬件系统介绍

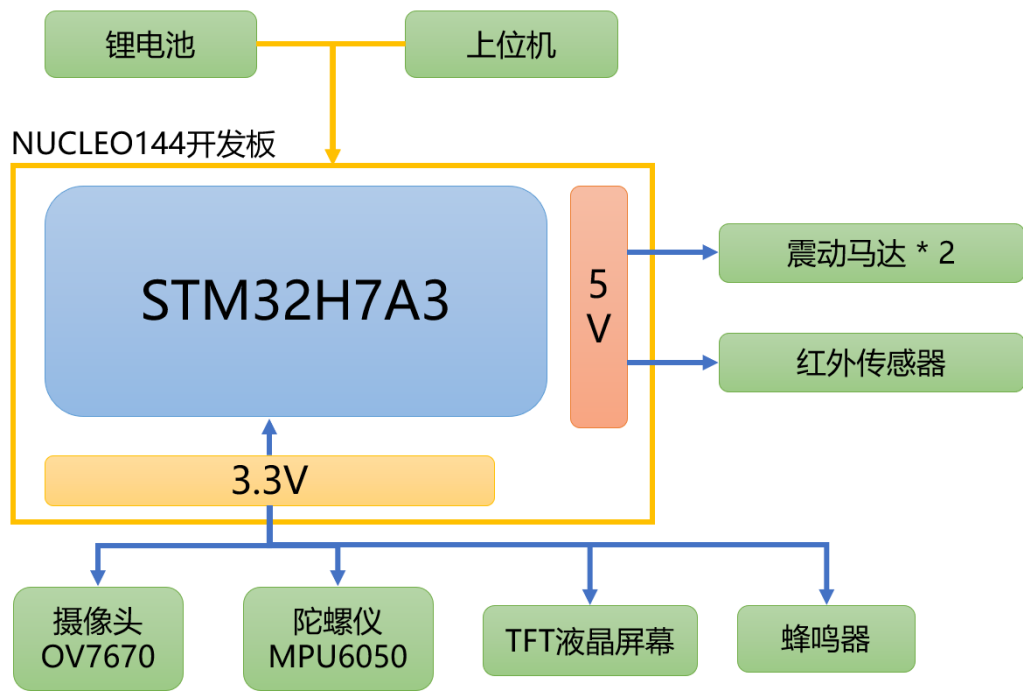
2.2.1 硬件整体介绍：



元件实物图

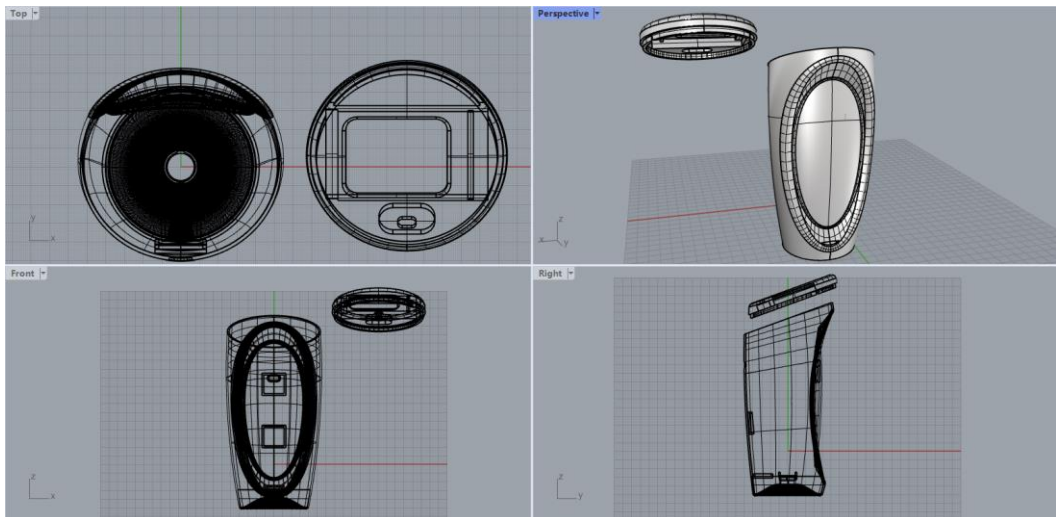
元件	型号
单片机	STM32H7A3ZI-Q
摄像头	OV7670 (240x320, 30fps, RGB565)
显示器	ST7735 (2.8" TFT LCD, 240x320, RGB565)
陀螺仪	ATK-MPU6050
蜂鸣器	无型号, 无源蜂鸣器
震动马达	无型号
红外传感器	无型号
电源	无型号 (锂电池, 900mAh, USB 公口输入, MicroUSB 公口输出)

元件与型号清单

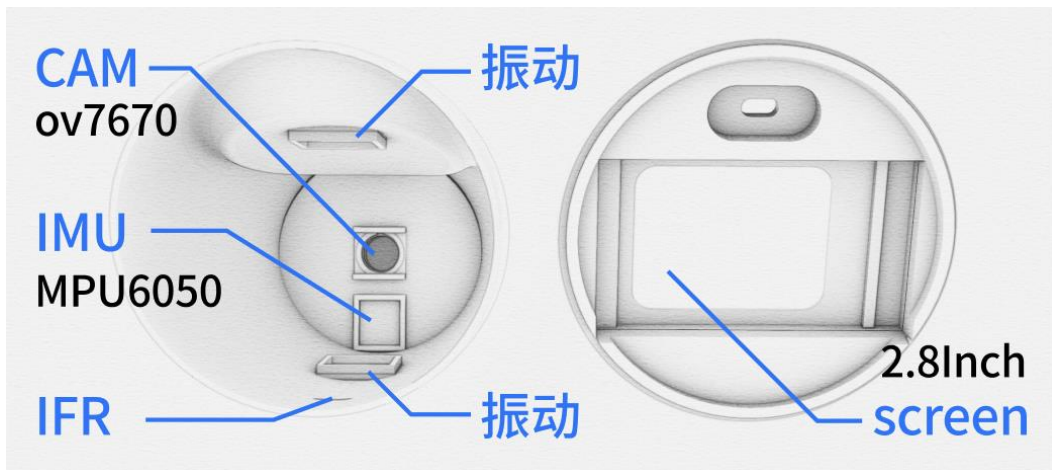


硬件系统总览图

2.2.2 机械设计介绍（如果有的话，从总体到局部，逐级给出各组件的具体设计图，可以是 CAD 文件截图或者手绘图片）；



外壳建模

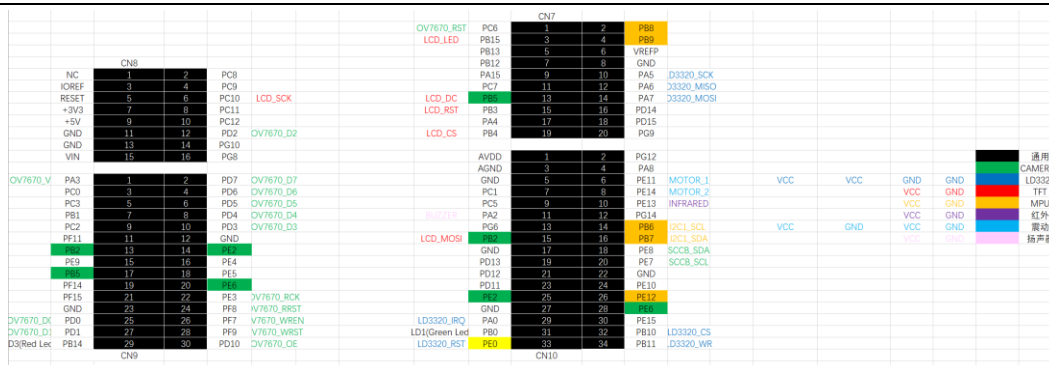


设计细节

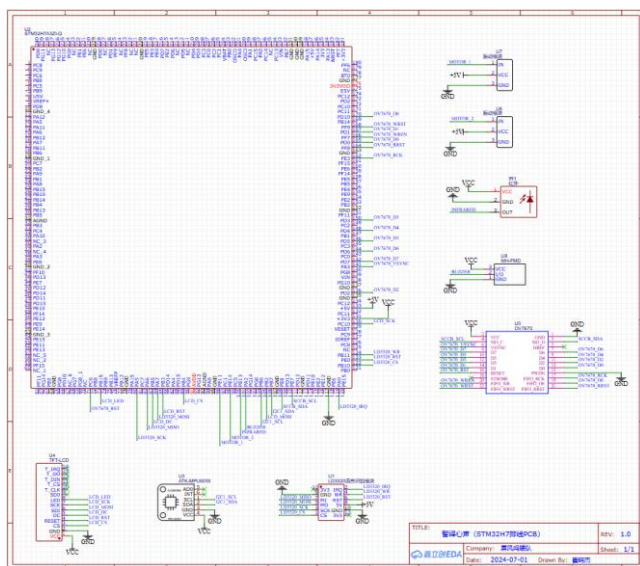
图示是本作品的外壳设计建模图，使用了 Rhino 软件完成建模，使用 3D 打印生产。本次的外壳设计采用温润的造型，融入日常生活，注重隐私保护。在使用方式上，符合直觉与人体工学。

设计细节上，根据元件尺寸设计卡口，恰到好处的防止每个元件，从外部看浑然一体，考虑到了陀螺仪初始位姿，红外线遮挡交互位置，摄像头与屏幕反向细节，非常便于使用。

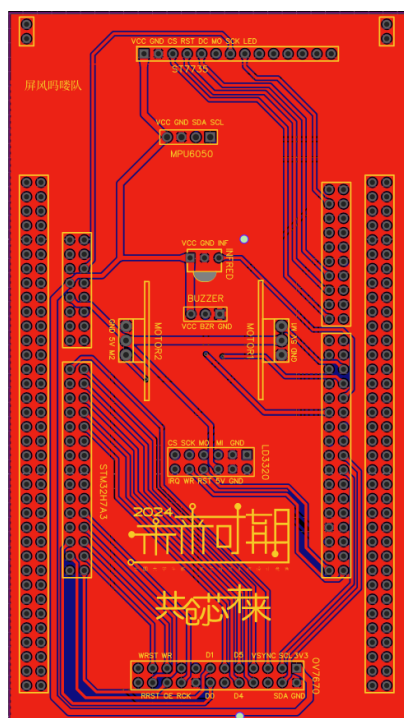
2.2.3 电路各模块介绍（从总体到局部，逐级给出各模块的具体设计图，并标记出关键的输入、输出信号线，可以是电路图、SCH 原理图、PCB 版图等截图）；



模块引脚连接图



原理图



PCB 图

由于本作品的模块众多，接线繁多，如模块引脚连接图。在 STM32 上直接插线会非常混乱，且数据传输不稳定，因此我们使用了嘉立创绘制了 PCB 用于排列布线，将此 PCB 先与 STM32 进行连接，之后就可以按区域整齐排线，便于后续放入 3D 外壳实物中。

1. 硬件系统

如硬件系统总览图所示，这就是本作品的硬件系统。核心是 STM32H7A3ZI-Q 单片机，开发调试时用上位机与单片机的 STlink 口连接，实

现设备供电与程序烧录。产品正常使用时，STlink 口与锂电池连接，仅作供电用。摄像头、陀螺仪、TFT 液晶屏幕与蜂鸣器用单片机的 3.3V 口供电，两个震动马达与红外传感器用单片机的 5V 口供电，具体的通信接口与相关实现下文会具体介绍。所有的元件与单片机的引脚连接见模块引脚连接图。

2. 单片机 STM32H7A3ZI-Q

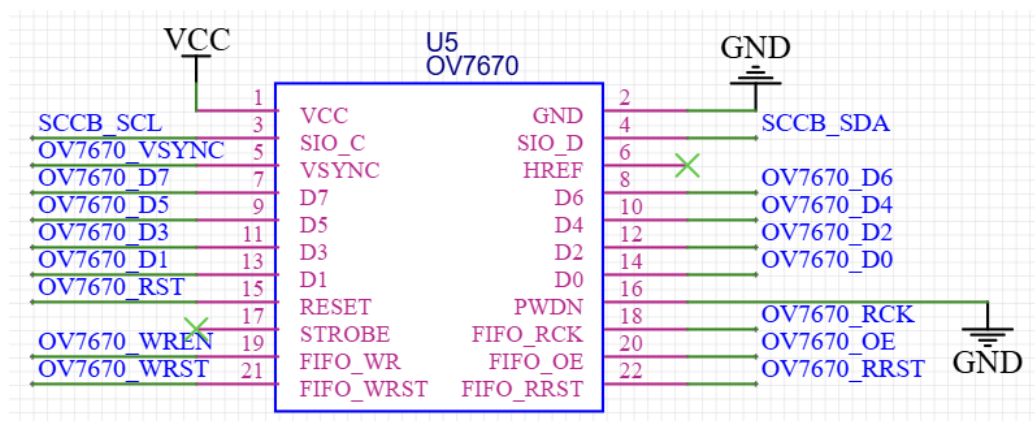
Table 12. USART3 pins

Pin name	Function	Virtual COM port (Default configuration)	ARDUINO® D0 and D1	ST morpho connection
PD8	USART3 TX	SB103 OFF, SB16 ON and SB15 OFF	SB103 OFF, SB16 OFF and SB15 ON	SB103 ON , SB16 OFF, SB15 OFF
PD9	USART3 RX	SB104 OFF, SB17 ON and SB94 OFF	SB104 OFF, SB17 OFF and SB94 ON	SB104 ON , SB17 OFF and SB94 OFF

单片机开发设置说明

如图所示的是该单片机使用说明书上的设置情况，图中的 SBxxx 是开发板背面的电阻。根据需求为了对单片机进行程序烧录开发，我们用热风枪取下开发板上的 SB103 与 SB104 电阻。之后就可以顺利通过 STlink 连接，使用 Keil5 烧录程序进行嵌入式开发。

3. 摄像头 OV7670 带 FIFO



摄像头 OV7670 引脚图

OV7670 是一种用于图像传感器和摄像模块的 CMOS 图像传感器。它广泛用于需要低成本和高性能的嵌入式系统中，如机器人、监控设备、物联网设备等。OV7670 模块分辨率调节成 320x240，帧率高达 30 帧/秒，支持各种输出格式（如 RGB、YUV 等），并带有自动图像控制功能，如自动曝光、自动白平衡和自动增益控制。

OV7670 模块的引脚配置如下：

- VCC - 电源输入（通常为 3.3V）。
- GND - 接地。
- SIO_C - SCCB 接口时钟线（用于配置寄存器）。
- SIO_D - SCCB 接口数据线（用于配置寄存器）。
- VSYNC - 垂直同步信号，用于指示一帧图像的开始。
- HREF - 行参考信号，用于指示有效图像数据行。
- D7 - 数据位 7。
- D6 - 数据位 6。
- D5 - 数据位 5。
- D4 - 数据位 4。
- D3 - 数据位 3。
- D2 - 数据位 2。
- D1 - 数据位 1。
- D0 - 数据位 0。
- RESET - 模块复位引脚，低电平有效。
- PWDN - 电源节能模式控制引脚，高电平有效。
- STROBE - 闪光灯控制引脚（未常用）。
- FIFO_RCK - FIFO 读时钟。
- FIFO_WR - FIFO 写时钟。
- FIFO_OE - FIFO 输出使能。
- FIFO_WRST - FIFO 写复位。
- FIFO_RRST - FIFO 读复位。

这些引脚使得 OV7670 能够与微控制器或 FPGA 进行连接，通过 FIFO 缓存和时钟信号来捕获和传输图像数据。

4. 显示器 ST7735

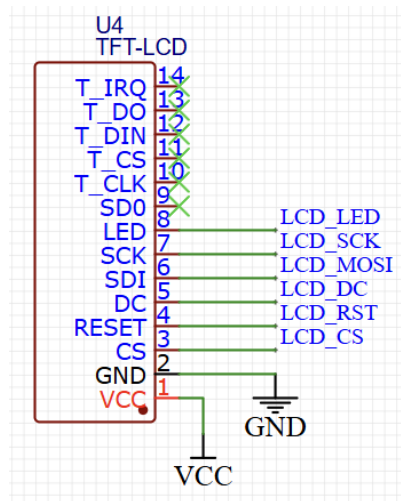
ST7735 是一种常见的 TFT LCD 控制器，广泛用于小尺寸的彩色显示屏。我们使用的显示器的尺寸是 2.8 英寸，分辨率是 320x240。ST7735 控制器通过 SPI 接口与主控芯片通信，具有较高的刷新速率和较低的功耗，适用于便携式设备、仪表、嵌入式系统等。

ST7735 模块的引脚如下：

- **VCC** - 电源输入（通常为 3.3V 或 5V）。
- **GND** - 接地。
- **LCD_CS** - 芯片选择信号，低电平使能。
- **LCD_RST** - 模块复位引脚，低电平有效。
- **LCD_DC** - 数据/命令选择引脚，高电平表示数据，低电平表示命令。
- **LCD_MOSI** - SPI 数据输入（主输出从输入）。
- **LCD_SCK** - SPI 时钟信号。
- **LCD_LED** - 背光控制引脚，通常连接到 PWM 控制或直接接电源。

在原理图中，其他引脚如 T_IRQ, T_DO, T_DIN, T_CS, T_CLK 等通常与触摸屏控制相关，而与 ST7735 控制的显示功能无直接关系。

在编写代码时，需要初始化 ST7735 模块，配置其寄存器以设置显示模式和颜色格式，然后通过 SPI 接口发送图像数据以更新显示内容。

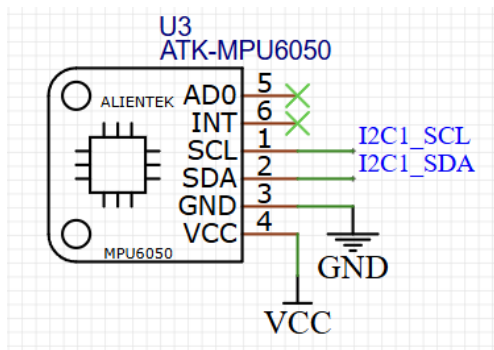


5. 陀螺仪 ATK-MPU6050

MPU6050 是一种高精度的 6 轴（3 轴加速度计和 3 轴陀螺仪）传感器，广泛应用于姿态和运动检测，如无人机、机器人和智能设备中。ATK-MPU6050 模块基于 MPU6050 传感器，通过 I2C 接口与主控芯片通信，支持高达 16 位的模拟到数字转换。

ATK-MPU6050 模块的引脚如下：

- **VCC** - 电源输入（通常为 3.3



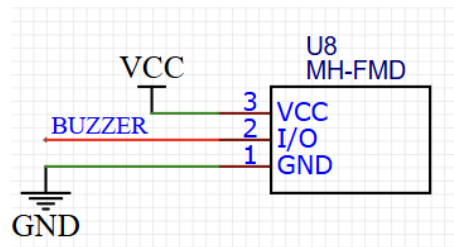
V 或 5V)。

- **GND** - 接地。
- **SCL** - I2C 时钟线，用于 I2C 通信。
- **SDA** - I2C 数据线，用于 I2C 通信。
- **INT** - 中断输出引脚，当发生特定事件时（如数据就绪），该引脚输出一个中断信号。
- **AD0** - I2C 地址选择引脚。当该引脚接地时，I2C 地址为 0x68；当该引脚接高电平时，I2C 地址为 0x69。

ATK-MPU6050 的引脚连接非常简单，通常只需要连接电源、接地和 I2C 接口的两条线即可。本作品中未使用其中断功能，因此无需连接 INT 引脚。

6. 无源蜂鸣器

无源蜂鸣器是一种常用于电子设备中的声音输出器件，通过外部驱动信号（通常是方波）来发出声音。与有源蜂鸣器不同，无源蜂鸣器需要外部电路提供周期脉冲音频信号才能工作，可以发出指定频率



的声音，适合用于播放音乐。无源蜂鸣器结构简单、成本低廉，广泛应用于报警器、计时器、电子玩具等设备中。

无源蜂鸣器模块的引脚如下：

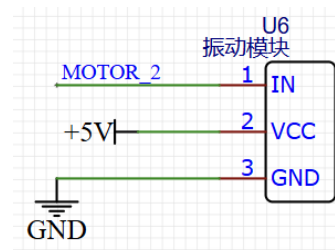
- **VCC** - 电源输入（通常为 3.3V 或 5V）。
- **GND** - 接地。
- **I/O** - 输入特定频率的方波信号

7. 震动马达

震动马达模块是一种常用于为设备提供触觉反馈的电子元件。它通过产生机械振动来传递信息，广泛应用于手机、游戏手柄、穿戴设备等场景中。震动马达模块通常由一个直流震动马达和控制电路组成，通过简单的信号控制实现振动效果。

震动马达引脚如下：

- **VCC** - 电源输入（通常为 3.3V 或 5V）。
- **GND** - 接地引脚，用于连接模块的地线。



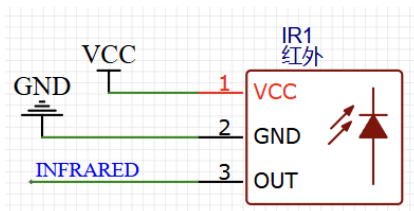
- **IN** - 控制信号输入引脚，用于控制震动马达的开关。当该引脚输入高电平信号时，震动马达启动；输入低电平信号时，震动马达停止。

8. 红外传感器

红外传感器模块通常用于检测物体或障碍物的距离或存在。

红外传感器引脚如下：

- **VCC** - 电源输入（通常为 3.3V 或 5V）。
- **GND** - 接地引脚，用于连接模块的地线。
- **OUT** : 输出引脚，通过这个引脚传输检测到的信号。输出信号的电平状态取决于检测到的物体或障碍物的存在与否。



9. 电源

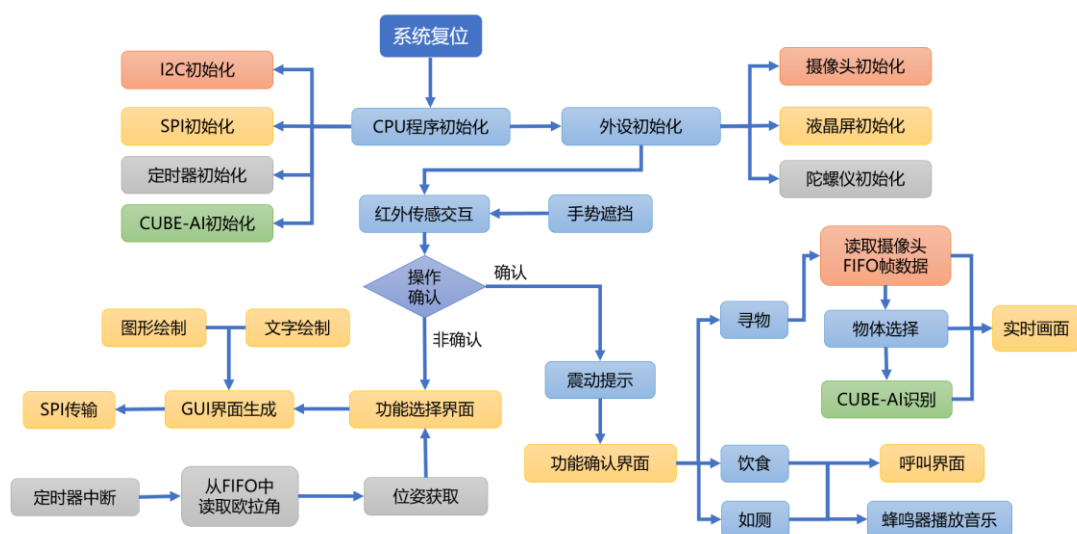
锂电池，容量 900mAh。

- USB 公口输入，便于充电。
- MicroUSB 公口输出，直接插入单片机 STlink 口供电。

2.3 软件系统介绍

2.3.1 软件整体介绍（含 PC 端或云端，结合关键图片）；

在 CPU 程序和各个外设的初始化完成后，系统进入主循环。主循环有两大状态：功能选择状态和功能运行状态，二者通过手势遮挡红外传感器来切换。系统处于功能选择状态时，在液晶屏幕上通过函数绘制 GUI 显示饮食、寻物、如厕三种状态，可通过倾斜整个产品来便捷地进行选择，其中硬件系统的位姿由定时器中断来读取，当检测到位姿发生改变时，对应到 GUI 中体现出来选择的变化；当用户手势确认选择后系统会通过震动马达提示，随后进入功能运行状态。如果用户的选择是饮食或如厕，会分别进入相应的呼叫界面，然后蜂鸣器播放音乐来通知家属；如果用户选择寻物的功能，则会打开摄像头，获取一帧数据后采用配置的神经网络进行物体识别，而摄像头拍摄的画面与识别的结果都会在屏幕上显示。在以上三种功能运行时，用户都可以随时通过手势遮挡红外模块终止该功能的运行，系统将再次进入功能选择状态。



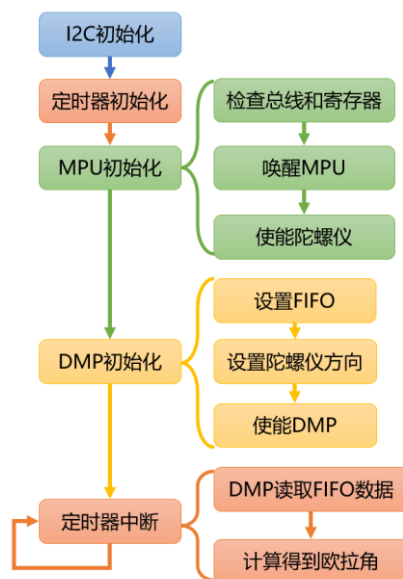
软件系统整体设计框图

2.3.2 软件各模块介绍（根据总体框图，给出各模块的具体设计说明。从顶层到底层逐次给出各函数的流程图及其关键输入、输出变量）；

1. 陀螺仪模块

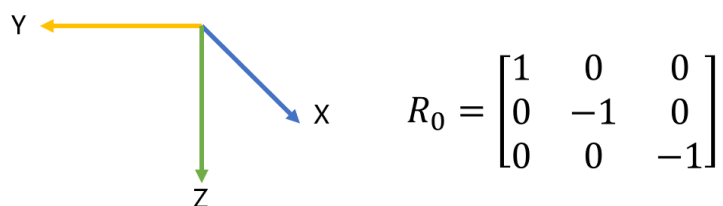
我们使用的陀螺仪为 MPU6050 模块，里面内置了 DMP 可以执行复杂的计算任务，如姿态估计和运动检测，从而减轻主处理器的负担，能够实时处理传感器数据，提供更准确和稳定的姿态信息。我们通过配置 DMP，让其处理传感器数据，执行姿态估计和手势识别等任务。DMP 处理后的数据可以通过 FIFO 缓冲区读取，这样就无需自己编写复杂的算法来处理传感器数据。MPU6050 工作流程如右：

单片机通过 IIC 与陀螺仪通信，使用定时器中断来规律地读取信息。在 CPU 程序完成初始化后，开始对 MPU6050 进行初始化，具体流程为：检查设备和总线是否准备好，发送写寄存器命令，唤醒 MPU，设置量程，关闭 FIFO，设置低通滤波器，使能陀螺仪和加速度计。然后对 DMP 进



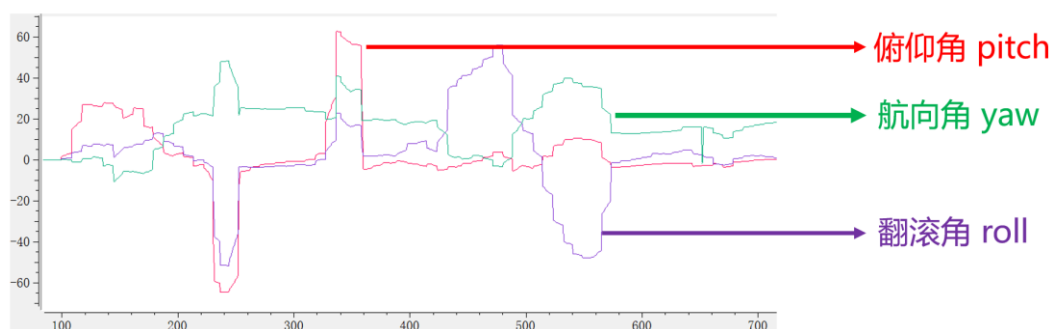
行初始化：设置 FIFO 和采样率，加载 DMP 固件，设置陀螺仪方向，设置输出速率，使能 DMP。初始化完成后就可以在定时器中断的回调函数中读取陀螺仪的信息了。

在使用 MPU6050 的陀螺仪时，我们还注意到在初始化和使用过程中比较重要且容易出错的两个方面。第一个是在 DMP 初始化时需要根据硬件的具体位姿进行初始化，定义初始姿态矩阵 R_0 来表示。如下图所示，我们在调试的过程中发现陀螺仪的姿态与标准姿态不一致，遂修改初始姿态矩阵来调整；第二个是在读取 FIFO 数据时不能太慢，否则会导致 FIFO 数据溢出，使陀螺仪不能正常工作，但陀螺仪只是我们系统的一小部分，于是我们使用定时器来解决：用定时器中断规律地读取 FIFO 数据（设置为 50ms 一次），需要时再使用，这样就保证了主程序和陀螺仪都可以正常运行。



陀螺仪初始姿态

下图显示了陀螺仪经过 DMP 处理后读取到的欧拉角数据：



陀螺仪欧拉角数据图

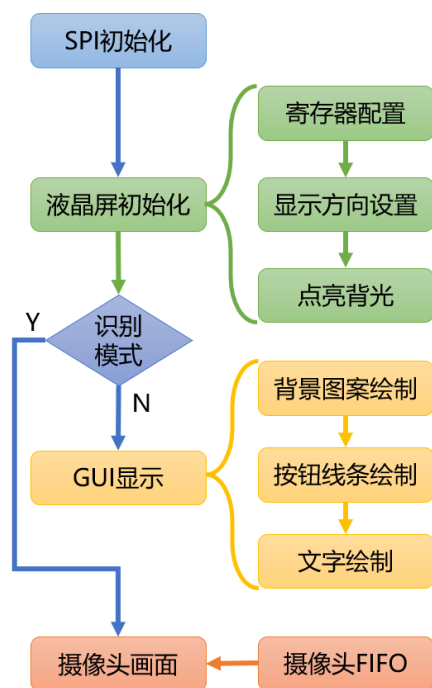
2. 液晶屏幕

我们使用的显示器是 ST7735 模块，2.8 寸 TFT LCD 屏幕。最高 30fps 刷新率，RGB565 格式显示图像。我们使用 LCD 屏幕实现 UI 界面显示，以及

摄像头画面实时显示。

单片机通过 SPI 与屏幕进行通信，SPI（串行外设接口）是一种同步串行通信协议，采用主从架构，通过 MOSI、MISO、SCK 和 SS 四条信号线实现数据传输，常用于微控制器与外设的高速通信。

SPI 通信通过主设备生成时钟信号，选择从设备后同步数据传输，主从设备在每个时钟周期内同时发送和接收数据，结束后释放从设备。SPI 具有高速传输、全双工通信、简单实现、灵活的从设备选择和低延迟等优点，适用于需要快速和实时数据传输的应用，如在 STM32 单片机与 LCD 显示屏通信中。



我们在使用了 SPI 通信的时候，尝试使用了软件 SPI 通信和硬件 SPI 通信。

软件 SPI（Software SPI）是通过微控制器的普通 I/O 引脚手动控制时钟和数据线来模拟 SPI 通信。软件 SPI 的优点是灵活性高，可以在任何 I/O 引脚上实现 SPI 通信，无需特定的硬件支持。但是我们使用后发现，此方法受 GPIO 速率限制，屏幕刷新率过慢。而后，我们转用硬件 SPI，硬件 SPI（Hardware SPI）利用微控制器内置的专用 SPI 硬件模块进行通信。硬件 SPI 的优点是传输速率高，效率高，占用 CPU 资源少，数据传输稳定，最终屏幕刷新率得到了不少的提升。

液晶屏幕的工作流程如右图：首先进行 SPI 初始化，接着进行液晶屏初始化，包括寄存器配置，显示方向设置，以及点亮背光。之后系统识别当前的显示模式，如果是 GUI 显示，那么一次进行背景图案绘制、按钮线条绘制以及文字绘制，显示在屏幕上；如果当前是摄像头画面显示，那么就读

取摄像头 FIFO 数据，实时显示在屏幕上。

在开发液晶屏幕时，遇到的问题包括闪屏、镜像等，需要根据手册，正确设置屏幕内置寄存器数据，以指定屏幕相关配置。其中，镜像可以通过设置扫描方向解决。此外，屏幕可以配置显示方向，以匹配系统需求。

3. 摄像头模块

我们使用了 OV7670 带 FIFO 型号的摄像头模块。SPI 具有高速传输、全双工通信、简单实现、灵活的从设备选择和低延迟等优点，适用于需要快速和实时数据传输的应用，如在 STM32 单片机与 LCD 显示屏通信中。

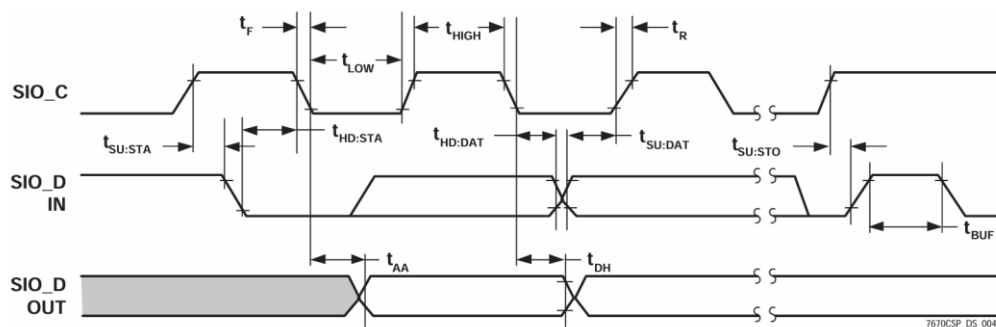
FIFO 在摄像头采集图像时将数据按顺序存入缓冲区，读取时数据按存入顺序输出，主控设备根据需求从 FIFO 中读取数据进行处理。FIFO 通过控制信号管理读写操作，确保数据完整性和顺序性。

FIFO 缓冲区可解决数据传输速率不匹配的问题，提高系统稳定性；数据顺序管理简单，降低主控设备处理负担；在图像采集应用中，FIFO 有助于缓冲大数据量，避免数据丢失，提升数据传输效率。



使用摄像头模块的原理如右图：首先初始化单片机与摄像头的通信协议 I2C。

接下来进行摄像头初始化，首先是复位 SCCB, SCCB 协议与 IIC 协议十分相似，不过 IIC 是 PHILIPS 的专利，所以 OmnVision 在 IIC 的基础上做了点小改动。SCCB 最主要是阉割了 IIC 的连续读写的功能，即每读写完一个字节就主机必须发送一个 NA 信号复位。通过查阅数据手册，SCCB 配置的时序如下：



SCCB 之后进行寄存器配置，包括 RGB 格式与光学参数配置。摄像头一共支持三种 RGB 格式：RGB565、RGB555、RGB444，为配合屏幕的 RGB565 格式我们自然选择了 RGB565 作为摄像头色彩数据的格式。而摄像头的光学参数包括以下几种，查阅数据手册后均通过写寄存器来实现：

白平衡设置：通过调节 AWB 蓝色通道增益和红色通道增益来调节白平衡；

饱和度设置：通过修改 6 个色彩矩阵实现；

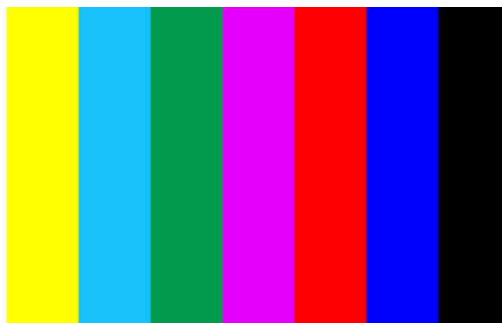
亮度和对比度设置：各对应一个唯一地址，直接修改大小即可；

滤镜模式：通过修改 TSLB、U 值、V 值来实现；

此外还有两个比较重要的位置也需要手动配置寄存器：

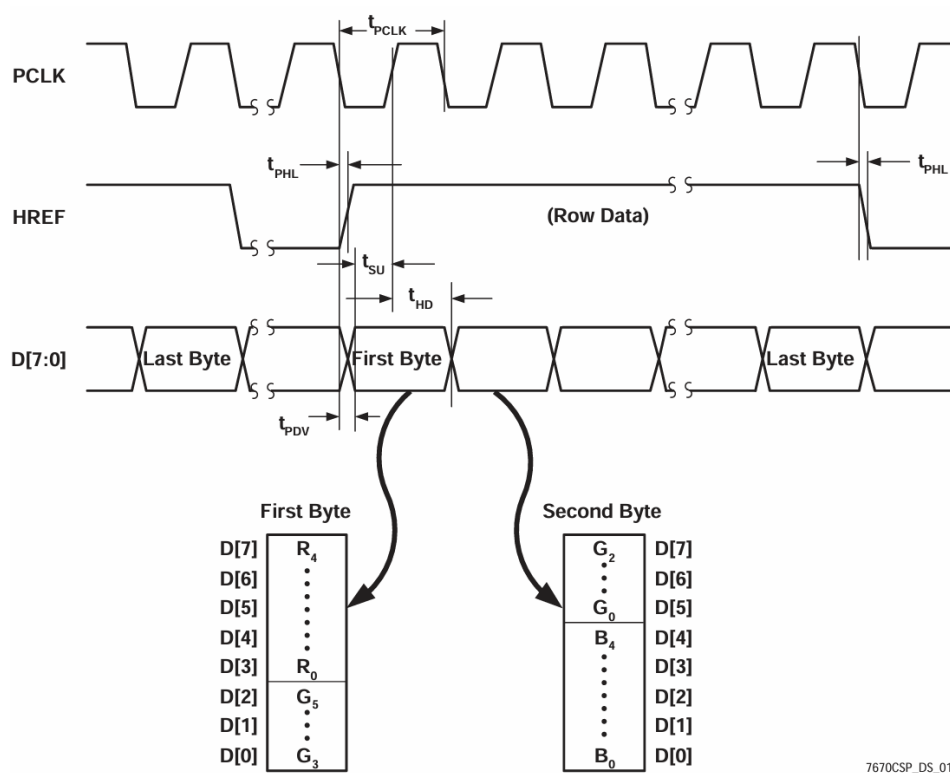
时钟设置：由于使用 CUBE-AI 需要尽可能拉高系统的时钟频率，我们的单片机频率达到了 280MHz 的上限，这个频率对于摄像头的默认设置来说过高，需要进行分频。我们通过配置摄像头的外部时钟分频和 PLL（锁相环）解决了这个问题；

彩带实验：在调试阶段如果出现花屏、黑屏、显示不全等原因时，很难确定是摄像头数据本身的问题还是 FIFO 的问题，甚至是屏幕的问题，于是可以配置一处地址将摄像头的画面变成彩带实验，通过与标准图片（如下图所示）对比可以帮助我们精准、快速地排查问题。



之后进入工作主流程——刷新帧信息，分三步走：第一步帧中断，我们为帧中断的接口配置了一个外部中断，当相机完成一帧的采集时发送中断信号，出发单片机的中断服务，在回调函数中复位写指针，并允许写入 FIFO；第二部执行时序指令，一次完成复位读指针、设置读时钟与置位读指针；第三步读取像素，设置读时钟，先读高 8 位，再读低 8 位。具体的时序控制和 RGB565 的读取方式如下图所示。这样就完成了一次帧信息的刷新，最后将信息显示在屏幕上。

RGB 565 Output Timing Diagram



7670CSP_DS_011

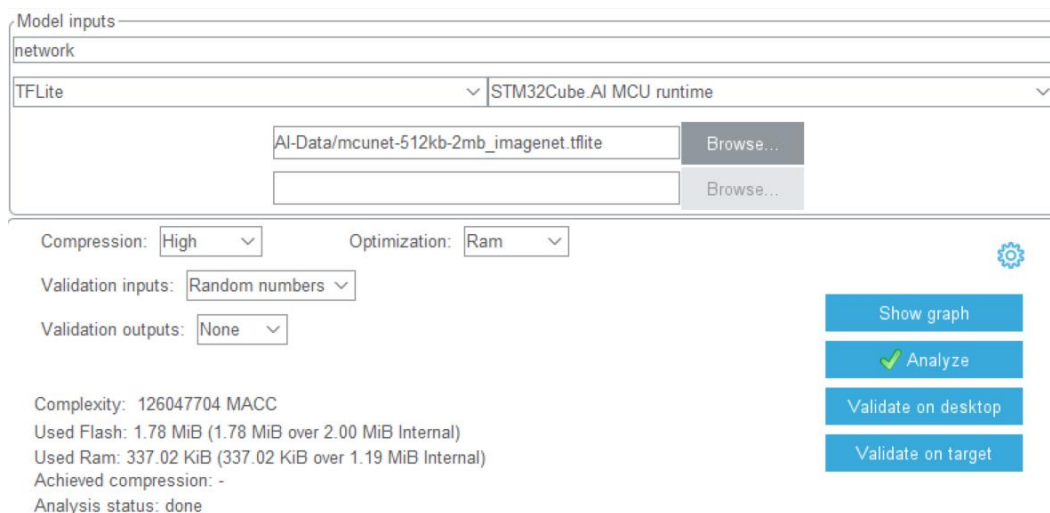
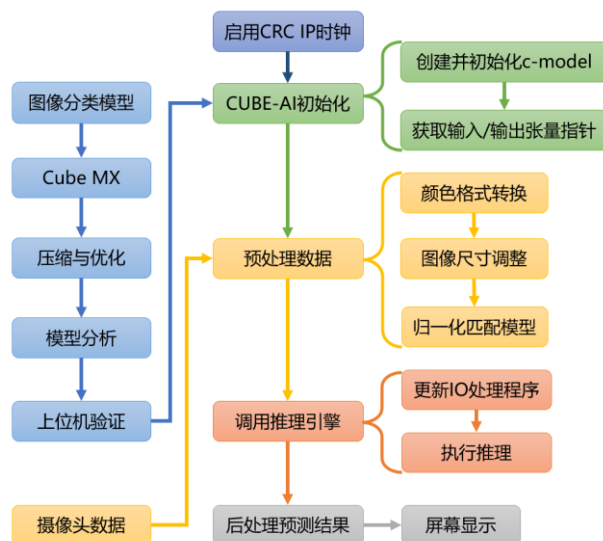
摄像头读取信号时序图

4. X-CUBE-AI

X-CUBE-AI 是 STM32 上的人工智能扩展包，它将预训练的神经网络模型转换为在 STM32 设备上运行的优化代码。该工具包通过 STM32CubeMX 集成开发环境使用，旨在实现嵌入式设备上的高效神经网络推理。

模型方面我们选用了 mit-han-lab 的 MCUNet 模型。

MCUNet 是一个为微控制器上的机器学习设计的框架，它结合了高效神经网络架构（TinyNAS）和轻量级推理引擎（TinyEngine），实现了在内存以及 Flash 受限的微控制器上进行 ImageNet 级别的实时推理。TinyNAS 通过两阶段神经架构搜索优化资源限制，TinyEngine 通过整体网络拓扑优化内存调度，显著减少内存使用并加速推理。



X-CUBE-AI 上位机运行结果

X-CUBE-AI 的流程如右图：首先将图像分类模型 MCUNet 传入 STM32CubeMX 进行压缩和优化，接着进行模型分析与上位机验证通过，

STM32CubeMX 生成相关代码。接下来启用 CRC IP 时钟，进行 CUBE-AI 初始化，包括创建并初始化 c-model，然后获取输入/输出张量指针。之后对摄像头采集的图像数据，进行预处理，以匹配模型输入数据的要求：1. 转换颜色格式，从 RGB565 转换到 RGB888；2. 调整图像尺寸，从 240*240 缩放至模型输入数据尺寸 160*160。3. 像素值归一化，调整成 $[-128, 127]$ 匹配模型。然后调用推理引擎，更新 IO 处理程序后执行推理。最后，后处理预测结果，找出预测结果向量中最大值对应的索引 id，再把 id 映射到 imagenet 数据集标签序号，找到对应的标签名称，实时在屏幕上显示分类预测结果和对应名称。

X-Cube-AI 中，输入和输出都视为一维向量，以指针形式存储。对于三通道图像数据而言，需要以行主序存储，RGB 三色存在相邻的三个字节，通道为 $[H, W, C]$ ，H、W、C 分别表示高度、宽度、通道数。对于 ImageNet 格式，输出是一个 1000 维向量，每一维的值表示预测为对应类别的可能性大小，选取最大值的索引即为对应的预测类别号。

第三部分 完成情况及性能参数

3.1 整体介绍（整个系统实物的正面、斜 45° 全局性照片）



正面照片

斜 45°全局性照片

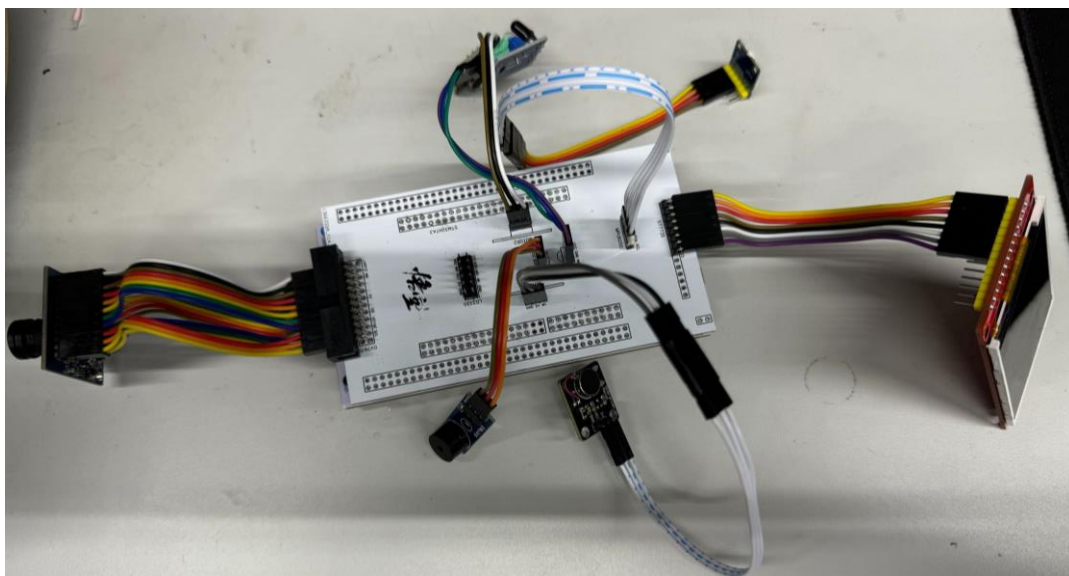
3.2 工程成果（分硬件实物、软件界面等设计结果）

3.2.1 机械成果；（实物照片）



实物照片

3.2.2 电路成果；（实物照片）



内部电路

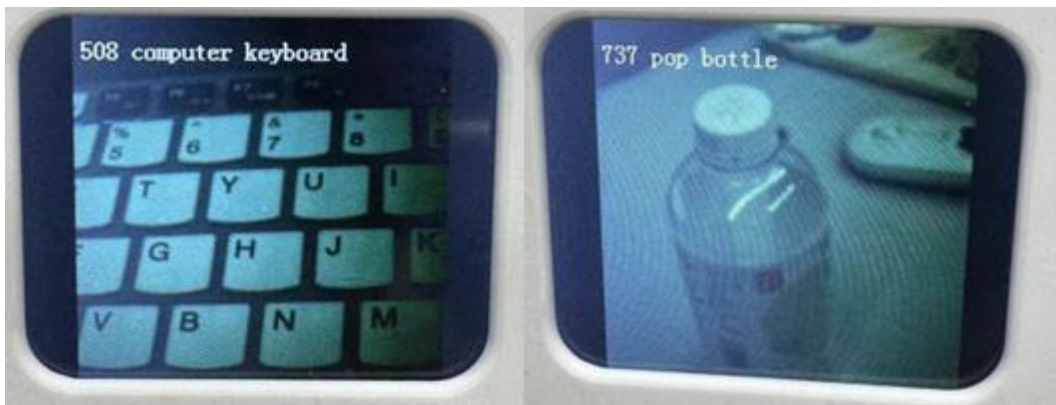


装配演示

3.2.3 软件成果：（界面照片）



UI 界面



X-CUBE-AI 效果

3.3 特性成果（逐个展示功能、性能参数等量化指标）（可加重要仪器测试或现场照片）；

1. **设备倾斜控制选项：**避免传统触屏操作的误选问题，适合难以精准控制肢体动作的患者。
2. **红外传感器确认选择：**操作简便且不易出错，适合患者使用。
3. **震动反馈：**设备震动反馈选择成功，帮助患者适应操作流程。
4. **特定选项的简易交互：**选择“吃饭/喝水、上厕所”选项时播放音乐呼叫家属，屏幕显示对应信息。
5. **物体识别辅助找东西：**选择“找东西”选项时，设备启动摄像头，结合

CUBE-AI 进行物体识别，屏幕显示实时画面，辅助寻找目标物体。

具体的效果图见上一小节。

第四部分 总结

4.1 可扩展之处

1. **功能增强：** 可通过添加新的传感器模块（如声音传感器、心率监测器等）或扩展现有传感器功能（如增加红外传感器的覆盖范围）来增强设备的功能性。
2. **通信扩展：** 可以添加无线通信模块（如 Wi-Fi、蓝牙），实现设备与外部设备（如智能手机、云服务）的连接和数据交换，进一步扩展设备的应用场景和功能。
3. **软件升级：** 设备的嵌入式软件可以通过固件升级实现功能增强和性能优化，例如增加新的交互模式或改进物体识别算法。
4. **外部接口扩展：** 设备的外部接口设计应考虑扩展性，以支持未来可能的外部设备连接或第三方应用程序开发，如 API 接口的开放设计。
5. **用户界面改进：** 根据用户反馈和市场需求，可以改进设备的用户界面设计，增加个性化设置选项或改进操作流程，提升用户友好性和可用性。

4.2 心得体会

开发使用 STM32H7 系列的单片机时，我们遇到的最大的困难，就是网上大多数的教程都是 F1 或者 F4 系列，供我们参考学习的 H7 系列单片机的开发少之又少，许多宏定义名称不同，寄存器约定也不同，无法直接套用教程提供的代码。因此我们从阅读元件手册，学习元件功能与引脚定义开始，重新编写程序，调用所有的模块。

在搭载各种模块时，我们在硬件上遇到许多困难。包括：TFT 屏幕闪屏、镜

像；摄像头画面静止不动、花屏、画面分块；通信速率过慢。摄像头的问题有时钟主频配置不对，通信协议时序错误，读取 FIFO 过慢导致被下一帧画面覆盖，中断无法触发等，在调试的过程中，彩带实验起到了重要的作用。通信时，我们先使用了软件 SPI，后因为速率过慢，又重新使用了硬件 SPI。

使用陀螺仪 MPU 时，因为初始化位姿问题，导致我们总是不能稳定地启动陀螺仪，一度让我们怀疑是自己的代码有问题，或是 CUBEMX 通信设置问题。后来我们设计了外壳，将陀螺仪位置粘贴固定住，并增加了 loading...画面，解决了该问题。

我们也曾经一度试图引入语音识别模块，LC3320，但是考虑到系统操作的便捷性和必要性，更主要的是 LD3320 模块的局限性，识别效果过差，最终我们舍弃了该模块，尽管我们花费了不少时间在此，不过还是优先考虑系统的整体性稳定性，将其舍弃。

为了保证摄像头读取数据的稳定性，我们曾考虑过使用排线，PCB 板上焊接牛角座排线插头来防反插。但是在制作 PCB 的时候，遇到了排线输入与输出的引脚布局的手性对称问题，如果更改 PCB 元件封装，则布线会变得非常困难，

嵌入式开发是一个综合的过程，需要考虑各方面功能的正确配合与合理调度。在加入 AI 功能后，要合理分配内存与 Flash 的使用，这涉及到空间占用的优化。在引入 Cube-AI 前，我们的 GUI 是通过将几张特定的图片转成数组，需要时直接调用，但这样极大地占用了单片机的内存，导致在加入模型后内存严重不足。为了给 AI 模型留出足够的空间，我们将这些图片全部删除，改为使用图形绘制函数的方式完成 GUI 的生成，节省了大量的 RAM 空间。在完成这一改动后，引入 AI 模型时便不会再出现内存不足的问题了。

此外，在进行图像预处理时，我们曾使用 malloc()函数申请堆空间，结果进入 HardFault_Handler 函数陷入死循环，无法继续运行程序，后查阅资料得知，在不借助 RTOS 的情况下，单片机上没有操作系统，因此没有内存管理能力，无法有效申请动态内存。最终，我们将变量改为静态数组变量，既避免了栈空间的溢出，也绕开了动态内存分配的问题。

第五部分 参考文献

- [1] Lin J, Chen W M, Lin Y, et al. Mcunet: Tiny deep learning on iot devices[J]. Advances in neural information processing systems, 2020, 33: 11711-11722.
- [2] 百炼成钢 123, CUBEAI 详细使用教程 (STM32 运行神经网络) ---以手写识别为例, <https://blog.csdn.net/xddwg521125/article/details/135358192>, 2024, 1, 16.
- [3] weixin_42368395, OV7670 带 FIFO 入坑指南, https://blog.csdn.net/weixin_42368395/article/details/104577056, 2020, 2, 29.
- [4] 渐晚 673, STM32f103c8t6 驱动 OV7670 带 FIFO 显示在 2.8 寸 TFT SPI 屏幕上, https://blog.csdn.net/qz_59509534/article/details/138474632, 2024, 5, 5.
- [5] 白云|苍狗, MPU6050 6 轴姿态传感器的分析与使用 (一), https://blog.csdn.net/qz_44852376/article/details/130470815, 2023, 5, 3.
- [6] eric2013, [SPI/QSPI] STM32H743 SPI4 寄存器版发送卡死问题求解的回复贴, <https://www.armbbs.cn/forum.php?mod=viewthread&tid=117991>, 2023, 2, 15.
- [7] MOV_A256, STM32 软硬件驱动 ST7735STFT 屏及遇到问题解决, https://blog.csdn.net/MOV_A256/article/details/134425358, 2023, 11, 15
- [8] happy08god, RGB565 与 RGB888 的相互转换, <https://blog.csdn.net/happy08god/article/details/10516871>, 2013, 8, 29
- [9] 折途想要敲代码, 【快速上手 STM32】SPI 通信协议&&1.8 寸 TFT-LCD, https://blog.csdn.net/m0_63235356/article/details/136977377, 2024, 04, 16
- [10] [有红包][QQ 红包]恭喜发财, STM32+TFT+OV7670 实现图片的显示, <http://t.csdnimg.cn/me3kU>, 2020, 08, 6

