

- 智能废料打包系统：物联网平台
 - 下位机：单片机平台
 - 操作模块：
 - 图传 + 管理模块：
 - 上位机：Win平台
 - 后台管理系统
 - 前端展示：Vue
- 项目目录结构
- 技术实现
 - 下位机：单片机平台
 - 操作模块：
 - 图传 + 管理模块：
 - 上位机：Win平台
 - 后台管理系统
 - 前端：

智能废料打包系统：物联网平台

下位机：单片机平台

操作模块：

- 使用STM32作为主控模块，淘晶驰电容屏作为显示模块；
- 两者之间通过串口通信；
- STM32通过储存箱底部的压力传感器，得到废料的重量，显示在屏幕上；
- STM32通过放置在存储箱顶部的红外传感器，检测是否已满，并将状态显示在屏幕上；
- STM32通过人体传感器检测是否有人在周围，通知图传模块（ESP32）进行拍照检测，并将检测状态（检测中、通过、未通过）显示在屏幕中；
- 通过后，用户通过点击屏幕上的按钮，进行放废料、取废料的操作；
- STM32单片机控制存储箱内的电机进行相应的操作；
- 操作完成后，通过串口发送给管理模块（ESP32）该用户操作信息；
- 刷新重量、存储箱是否已满状态。

图传 + 管理模块：

- 使用ESP32作为主控模块，通过MQTT协议连接上位机系统，通过摄像头捕捉人脸信息，通过串口给STM32传递信息；
- 连接上位机系统，通知上位机有一个废料存储箱已上线，并获得上位机分发给的唯一ID编号；
- 实时检测STM32发送的人体检测信息，若有则开启摄像头，并向上位机上传视频，上位机检测是否是有权限用户，并将信息通过MQTT协议发回ESP32状态信息；
- 检测结束，有权限，检测接受STM32发送的操作信息，并发送给上位机系统；
- ESP32进入待机状态。

上位机：Win平台

后台管理系统

- MQTT服务器：
 - 接受ESP32发送的信息，并转发给Java服务器端；
- Java服务器：
 - 通过监听MQTT服务的消息，并进行相应的处理；
 - 人脸检测消息，调用人脸检测API进行检测，将检测信息通过MQTT服务器发回给ESP32；
 - 用户操作信息，处理用户操作信息，使其格式化，并存储在数据库中；
 - 提供前端数据；
 - 有几个废料存储箱，废料存储箱的重量、是否已满信息；
 - 对应的废料存储箱，底下的用户操作信息；
 - 人脸录入，权限管理：
 - 通过管理员，将人脸录入，并赋予其相应的权限（可以操作那个废料存储箱的权限，可以进行何种操作的权限，（存储、取走））。

前端展示：Vue

- 通过浏览器进行界面展示；
 - 管理员登录；

- 人脸录入；
- 权限管理；
- 信息查看：
 - 废料存储箱信息；
 - 人员操作信息；
 - 人员权限信息。

以上为系统的基本架构和模块设计，将实现智能废料打包系统的各个层次整合在一起。

项目目录结构

- **HostSystem/**
 - **Frontend/**
 - README.md <!-- 前端项目说明文档 -->
 - **Backend/**
 - **MQTT_Server/**
 - **Java_Server/**
 - README.md <!-- 后端项目说明文档 -->
 - **Python_Server/**
 - README.md <!-- 人脸识别项目说明文档 -->
- **SlaveSystem/**
 - **STM32/**
 - README.md <!-- STM32项目说明文档 -->
 - **ESP32/**
 - README.md <!-- ESP32项目说明文档 -->
- **Documents/**
 - <!-- 存放项目文档，如设计文档、API文档等 -->
- **Tests/**
 - <!-- 存放测试相关文件 -->
- .gitignore <!-- Git忽略文件配置 -->

- LICENSE <!-- 项目许可证 -->
- README.md <!-- 项目说明文档 -->

技术实现

下位机：单片机平台

操作模块：

- stm32

图传 + 管理模块：

- ESP32

上位机：Win平台

后台管理系统

- MQTT服务器：
 - 接受ESP32发送的信息，并转发给Java服务器端；
- Java服务器：
 - 架构：springboot
 - 数据库：mangodb
- python服务：深度学习pytorch

前端：

- vue3+TypeScript+vite