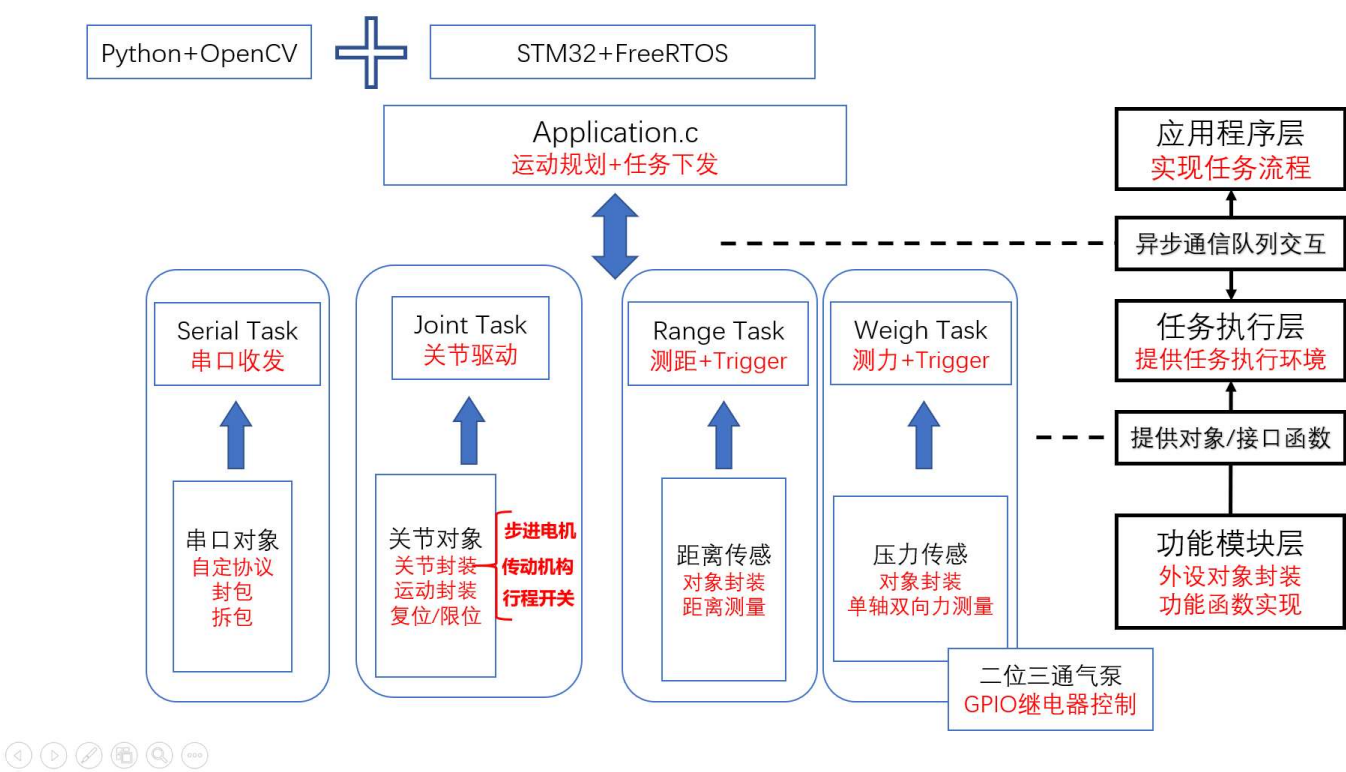


Logistic Sorting Robot

控制方案总览



具体实现

1. STM32端

1.1 功能模块层

本层完成外设对象封装以及功能函数实现

- 串口对象(serial.c & serial.h)
 - 自定简单通信协议
 - 数据域 uint8 DATA[14]
 - int16((DATA[00] << 8) | DATA[01]) 交互指令
 - int16((DATA[02] << 8) | DATA[03]) 快递X坐标
 - int16((DATA[04] << 8) | DATA[05]) 快递Y坐标
 - int16((DATA[06] << 8) | DATA[07]) 快递Z坐标
 - int16((DATA[08] << 8) | DATA[09]) 快递柜X坐标

int16((DATA[10] << 8) | DATA[11]) 快递柜Y坐标

int16((DATA[12] << 8) | DATA[13]) 快递柜Z坐标

- 封包

void SerialPackTransmit(void* pData)

- 拆包

void SerialPackExtract(SerialPackge_t *pPack, void *pData)

- **关节对象(mechanism.c & mechanism.h)**

- 关节对象封装

结构体形式->{电机, 头限位, 尾限位}

- 关节运动控制

void DriveJointMove(uint8_t jointName, float tarPos, float tarVel)

构建关节位移量与电机运动量

- 关节限位绑定

void SWiClicked(void)

void SWiReleased(void)

将限位中断信号与电机绑定, 便于后续实现复位操作

- 限位开关流程 (按下/释放)

void SwitchClickedProcess(uint8_t jointName, uint8_t SwID)

void SwitchReleasedProcess(uint8_t jointName, uint8_t SwID)

ZerosP1-> ZerosP2-> Limit

可在限位基础上实现更高精度复位

- 关节复位

void JointResetToHead(uint8_t jointName)

在以上绑定及流程设置下, 封装关节限位函数

- **压力传感器对象**

- 对象封装

WeightSensor_t* WeighingSensor_Allocate(GPIO_TypeDef* sck_port, uint32_t

sck_pin,GPIO_TypeDef* dout_port, uint32_t dout_pin)

- 测力

void WeighingSensor_Actuation(WeightSensor_t* This)

- **距离传感器对象**

- 对象封装

UltraSonic_t* Allocate_UltraSonic(GPIO_TypeDef *trigPort, uint32_t trigPin, TIM_TypeDef

*tim, TIM_HandleTypeDef *htim, uint32_t channel)

- 测距

void UltraSonic_Actuation(UltraSonic_t *This)

- **二位三通气泵 (真空吸盘力源)**

- 继电器信号 (GPIO输出)

- PUMP_ON 泵开

- PUMP_OFF 泵关

- PUMP_RELEASE_ON 泵卸压开
- PUMP_RELEASE_OFF 泵卸压关

1.2 任务执行层

利用实时操作系统任务管理机制，提供串口对象、关节对象、压力传感对象、距离传感对象任务执行环境。

1.3 应用程序层

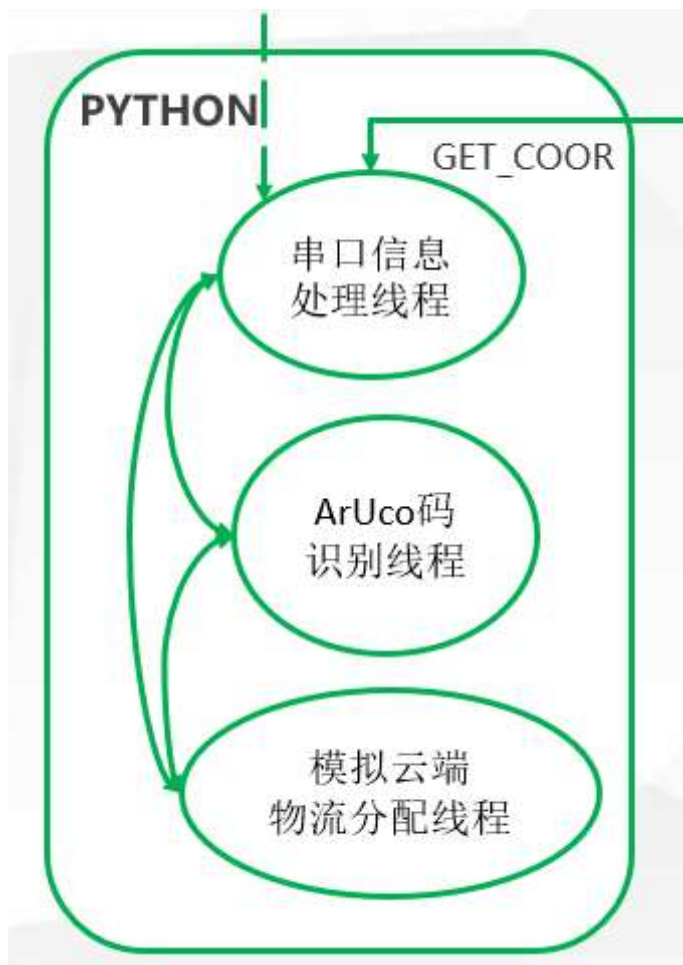
完成物流分拣机器人一批分拣任务执行全流程。

对于具体控制任务，利用实时操作系统异步队列机制下发到任务执行层交由具体对象执行。

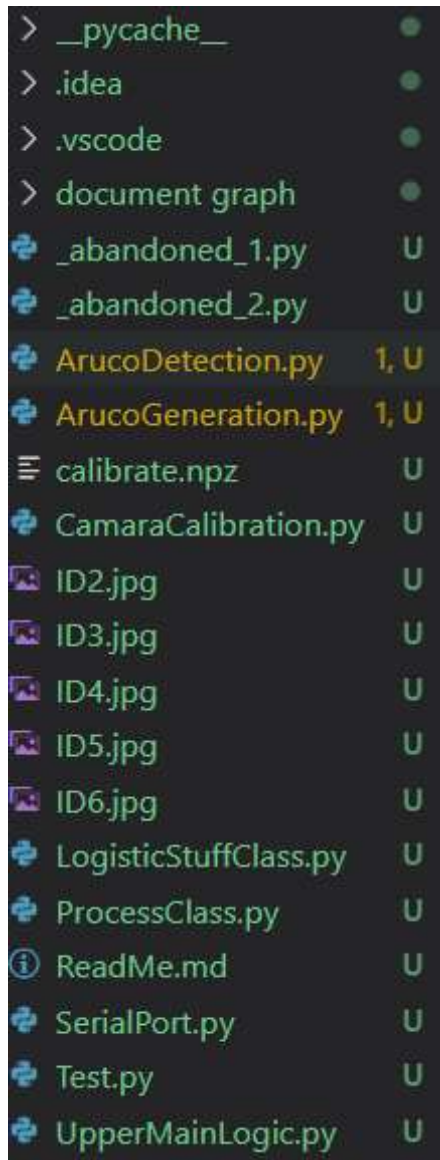
考虑了两种意外情况，do...while实现。

2. PC Python端

项目组成



项目文件



- **CameraCalibration.py**
单目相机校正系数获取
- **ArucoDetection.py**
基于OpenCV库的ArUco检测工作
- **LogisticStuffClass.py**
快递柜、快递对象（矩阵、坐标形式）
- **ProcessClass.py**
三个线程类对象（见项目组成）
- **UpperMainLogic.py**
上位机运行实例（演示视频所用的）