

# 嵌入式C语言大作业——树莓派MK.II监控设备v1.0 不完全介绍

---

## 实现目标

- 通过树莓派控制舵机和摄像头，并且推流至服务器端实现远程监控和控制摄像头角度。
  - 增加人脸识别功能，简单的锁定人脸稳定系统。
- 

## 主要配件

- 树莓派raspberrypi 4B 4G内存
  - MG996R舵机\*3
  - SG90微型舵机\*1
  - 3D打印的开源图纸配件
  - 树莓派摄像头
  - PCA9685舵机控制板
  - LM2596可调数字显示变压模块
  - 12V5A电源适配器
  - 杜邦线若干，自攻螺丝自锁螺母若干，透明胶带若干
- 

## 大体框架

主要实现内容可分为三部分，树莓派端，阿里云服务器端，Onenet物联网平台及微信小程序端。

- 树莓派端
    - 安装了openCV库和face\_recognition库实现推流和人脸识别。
    - 利用了自带的C语言库lwiringPi向9685控制板地址写指令，通过PWM脉冲信号控制舵机。
    - 主要有三个程序，local\_main\_control.c负责中心控制舵机，接受来自python的信息。而face\_rec\_streamer.py负责将摄像头输入信息抽帧，人脸识别分析运算，然后通过rtmp协议推流至云服务器端，同时将识别信息通过嵌套字socket传给local\_main\_control.c。get\_data\_from\_server.py负责通过Onenet物联网平台和微信小程序通信，接受来自微信小程序的JSON指令并将其通过嵌套字传给local\_main\_control.c。
  - 阿里云服务器端
    - 安装了nginx负责推流，将从树莓派推上来的rtmp视频流切片转化为hls使得可以在手机原生浏览器上播放。
  - Onenet以及微信小程序
    - 编写了一个简单的微信移动端控制系统，可以通过request向Onenet数据流发送datapoints。
-

## 实现过程的吐槽

- 烧了一个树莓派，一张SD卡，玩坏两个舵机。再也不敢水冷树莓派，也不敢用手掰舵机。
- 推流延迟30s却也没啥办法很气。
- 由于推流的速度远大于每一帧人脸分析运算的速度，所以会堵塞通道导致卡顿。正解应该是写个多线程，但我不会调线程，于是直接暴力没10帧分析一次。
- 自己写的负反馈条件人脸稳定跟个憨憨一样不够灵敏。
- 微信小程序如果想拉流必须企业认证（可恶），想进办法做不到。又尝试用web去写移动控制端，结果发现onenet提供的sdk的api直接404了（悲），然后js的xmlHttpRequest喜闻乐见的跨域，然后直接放弃qwq。
- 先是推rtmp流，然后发现flash不被支持了（可恶），然后为了手机能看改成了hls流。据说b站可以有不依赖flash的前端播放工具播放http-flv流，但是我实在懒得往nginx里再add-module了（据说要重装）所以用hls也不错。
- 树莓派烧了之后差点直接remake，还好前一天正好备份了系统，然后还好备份的系统能用，最后人民币-400 ¥，煎熬了三天。
- 由于硬件接触不良疯狂排查软件问题。
- 3D打印精度有限，自己开了好几个孔还弄坏了zbq同学的扩孔器\*1。 •••
- 感谢tc同学的摄影协助和zbq同学的各种各样的帮助(-w-)和乃琳贝拉嘉然向晚珈乐的精神援助。

