

嵌入式工作室23级暑期招新-IOT

物联网（IoT，Internet of Things）是指通过互联网将各种物理设备（如传感器、执行器、智能设备）连接起来，并通过这些设备之间的数据交换和通信，实现设备间的智能化、自动化和远程控制的技术和概念。物联网的核心是将传统的物理世界与互联网相连，使得设备可以通过网络进行远程监控、管理和交互。

初级项目实践



初级项目主要是帮助大家快速上手esp32，熟悉arduino的开发流程。（对于Linux环境开发熟悉的同学建议使用RTOS_SDK或者espidf框架进行开发，开发难度较大）

1. 停车摇杆控制和屏幕显示

硬件准备：

- 屏幕显示使用推荐使用SSD1306模块
- 摇杆控制推荐使用SG90舵机模块
- 主控板推荐使用ESP32C3（至于为什么选择这个模块，推荐大家到乐鑫官网上进行选型比较）



tips：这个部分主要是了解arduino平台开发的流程。关于学习路线方面可以参考b站上的太极创客，同时记得多参考乐鑫官网的资料哟ヾ(°▽°)ノ”。

任务要求：

1. 你需要在ESP32上搭建一个Web服务器，使其能够响应HTTP请求。这个服务器将用于控制摇杆和屏幕显示。
2. 在电脑端使用网络控制台,使用curl命令或者通过Python的 `requests` 库向ESP32发送HTTP请求。
3. 请求的内容信息应该以JSON格式的数据发送，包含控制舵机的指令，以实现摇杆的转动。
4. 通过在电脑端发送JSON格式的内容给ESP32，ESP32接收到后能够解析并在连接的OLED屏幕上显示电脑端发送过来的信息。

2. 摄像头监控

硬件准备：

- 主控板推荐使用ESP32-CAM模块
- 摄像头模块推荐使用OV2640模块



tips：在前面arduino平台开发的基础上，大家将通过这个项目进一步使用esp32cam摄像头模块，但是同时需要学习一下python,进行简单的网络通信交互。

任务要求：

1. 你需要在ESP32-CAM上搭建一个Web服务器，使其能够响应HTTP请求。这个服务器将用于提供实时的图像数据。
2. 在电脑端，编写一个Python程序，该程序可以通过网络与ESP32-CAM通信。使用Python中的HTTP请求库（例如 `requests` 库），向ESP32-CAM发送HTTP请求，请求获取图像数据。
3. 当Python程序收到ESP32-CAM发送的图像数据时，可以将图像展示在电脑界面上。
4. 确保ESP32-CAM上的Web服务器能够正确响应图像请求，并且Python程序能够成功获取并展示图像。

中级项目实践



Tips: 恭喜你,进入到了第二阶段.在第二个阶段中,我们将深入学习操作系统的相关知识,并尝试在操作系统上进行任务驱动.同时尝试一下ESP32的独有OTA(空中烧录功能),能够远程烧录我们的固件.



注意:在这个部分中,我希望你们能够详细记录你们的学习过程,尤其是在操作系统任务设计部分以及OTA功能部分,体现出你们的解决思路.

1. freeRTOS操作系统



tips：为什么需要操作系统，而不是单单的裸机开发？为什么在物联网领域我们选择freeRTOS操作系统？关于freeRTOS操作系统如何进行学习，B站上韦东山的教程对于入门级的选手十分友好。但是如果进一步深入了解freeRTOS，建议查看官网资料,阅读内核源码。

1. 理解进程和线程之间的区别，单片机上的任务是指线程还是进程。
2. 任务创建需要传入哪些参数，它们的作用是什么。

3. 请你阐述调用OSTimeDly的整个延时的发生流程。
4. 什么是信号量，什么是消息队列，什么是邮箱，什么情况下需要使用到它们。

任务要求:

1. 移植官方的esp_idf框架，执行官方的example的hello_world程序。
2. 从点灯开始我们的freeRTOS操作系统的学习，通过编写任务驱动，使得LED进行闪烁。
3. 尝试构建一个边端停车系统,需要满足以下要求:
 - a. 在电脑端运行边端控制系统（推荐使用python程序，较为简单），负责鉴权操作。
 - b. 摄像头模块需要定期上传图片给电脑端。（可以使用摄像头模块拍摄手机上展示的车牌号图片）
 - c. 电脑端获得摄像头模块拍摄的图片后，进行车牌号识别(这个部分可以调用云服务api,也可以选择自己本地部署一个车牌号识别模型)，获取到车牌号识别结果后，下发车牌号信息给ESP32模块在OLED屏幕上显示车牌号信息。接着进行鉴权操作(可以预先设定能够通过的车辆)，如果能够权限满足，则下发指令给ESP32模块升起摇杆，允许车辆通行。
 - d. 升起摇杆后，使用红外传感器判断车辆是否通过摇杆，如果通过摇杆，则降下摇杆。
 - e. ESP32模块同时需要定期发送摇杆状态给电脑端。
4. 希望能够在任务设计中，给出你的见解。同时希望你的系统调度能实现及时性，能够满足电脑端下发指令后，ESP32模块能够及时反应，控制相应的执行器进行操作。

2. OTA（空中烧录功能）

 Tips: 在这个部分中鼓励大家自行探索、发现和解决问题。除了OTA功能，ESP32还支持低功耗模式和ESP_MESH组网功能，有能力者可以自行探索哟。

任务要求:

1. 通过OTA能够远程烧录固件到ESP32中，同时希望在串口监视器中能够看到ESP32正确执行我们远程的固件。
2. 希望能够尝试本地主机连接ESP32直接进行远程烧录，同时尝试阿里云平台或者ONENET云平台连接ESP32实现远程固件烧录。

高级项目实践

经过上面的实践，恭喜你，成功掌握了IOT的开发的的基本技术。现在你可以自由选择你的想做的项目，非常建议和出题人一起探讨，我们将尽力帮助你实现你想做的项目，提供帮助将包含指导建议和技术支持(包括PCB设计，3D打印，系统移植等)!!!

下面这个网站是关于一些比较有趣的硬件项目的分享

[Top 6 ESP32-S3 Projects Revolutionizing IoT: From Smart Devices to Security Innovations - DFRobot Ma](#)

下面这个网站是esp32的博客网站,提供了很多很有趣的开源项目

<https://blog.espressif.com/>

下面这个项目是用ESP32C3作为主控芯片制作智能手表

<https://www.instructables.com/MutantW-V2-DIY-ESP32-S3-Smartwatch-That-You-Can-We/>

下面是智能手表的仓库的代码链接

https://gitlab.com/mutantcybernetics/mutantw-v2/-/tree/main?ref_type=heads

下面这个项目是应用于植物监测的

<https://github.com/ovidiu4/smart-plant-monitor/tree/main>

下面这个项目是构建3D打印机的

<https://github.com/MarlinFirmware/Marlin?tab=readme-ov-file>

项目提交

提交内容

 我们期待的提交内容：一份用 **markdown** 撰写的、**可读性强**的能够展现你完成招新项目的**完整过程**的笔记以及一个代码结构清晰、附带详细的README文件GitHub仓库，最后是一个展示项目效果的demo

1. **学习笔记.pdf**: 项目完成过程的学习笔记、遇到的困难与解决方案、你的所思所想等，文件命名：学习笔记
2. **github.txt**：项目 github 仓库链接（必选）
3. **demo.mp4**: 项目演示视频（必选），
4. 其它能展示你能力的补充材料（可选）

提交方式

 嵌入式工作室期待遇见一个有学习热情、自驱力强、有规划的你，希望你能合理安排招新题完成进度~

请在规定截止时间内将项目打包发送至 2377767409@qq.com，邮件标题为：**嵌入式工作室招新-学号-姓名-IOT**，打包命名：**学号-姓名-IOT.zip**

第一次提交 DDL：**2024-08-01 23:59:59**

做多少交多少：无论你完成进度如何，可能只是学习了一些理论，只有些许学习笔记，我们都希望你能够提交，这样有助于出题人更好了解大家的完成进度，适当对招新题作出一定调整~

第二次提交 DDL：**2024-09-01 23:59:59**

有疑惑别害羞    

1. 多去B站看看教程。
2. 灵活使用搜索引擎。
3. 学会去外网找资料。
4. 多参考别人的开源项目
5. 有问题私戳出题人  邱志颖  屈洪涛

 暑期招新只是将你引入门，技术的学习无终点，希望小伙伴锚定方向深耕技术，不断提高自身实力！

立即行动，加入我们，共同塑造属于我们的未来！ 