



2019 年 SEU-Xilinx 国际暑期学校团队项目设计文档

(Project Paper Submission Template)

| | |
|--------|--------------------------|
| 作品名称 | 基于 SEA-S7 的二维码识别系统 |
| 组员姓名 | 戴荣时 张陈睿 程嘉轩 孙沐天 |
| 房间号及桌号 | 717/16 组 |



第一部分

小组成员分工

(各成员任务分配)

| | 姓名 | 任务分配 |
|------|-----|---------------------|
| 组长 | 戴荣时 | 二维码识别算法的设计与编写 |
| 组员 1 | 张陈睿 | 对摄像头采集的数据进行预处理，完善接口 |
| 组员 2 | 程嘉轩 | AWS 服务器端配置与通讯 |
| 组员 3 | 孙沐天 | ESP32 通讯功能调试 |

第二部分

设计概述 /Design Introduction

(请简要描述一下你的设计：1. 功能说明；2. 所用到的设备清单)

1. 功能说明

通过摄像头进行图像采集，进行像素数据的预处理，之后采集中心区域的像素数据，进行二维码的特征识别。识别结果由 fpga 通过 spi 协议发送给 ESP32，然后再传输到 AWS 云端。

2. 设备清单

MIPI 摄像头、HDMI - Mini HDMI 转接头、SEA-S7、

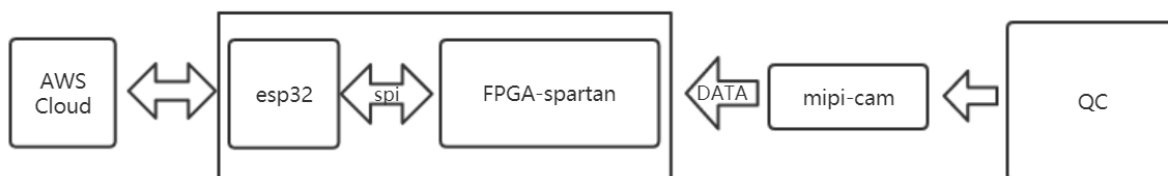
第三部分

详细设计 /Detailed Design

(请详细说明你作品要实现的所有功能以及如何组建系统以实现该功能，还包括为实现该功能需要用到的所有参数和所有操作的详细说明，必要的地方多用图表形式表述)

如因文档篇幅限制无法详述，可提供附件。

系统硬件如下

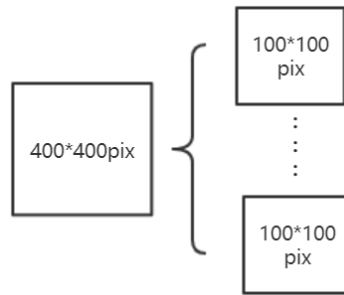


首先，从摄像头采集到的信息提取出每个像素点的 24 位 RGB 值，进行灰度转化和二值化，每个像素点的黑白信息只用 1bit 数据存储。图像处理部分，由于板子资源限制，我们截取中间部分的 400*400 的像素点进行二值化处理，并将数据结果存储到 bram，其资源以使用了 10 个 bram18k。

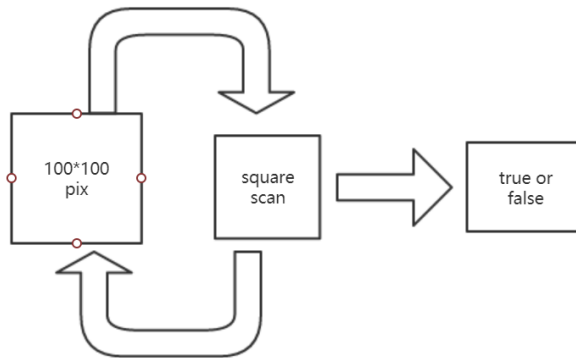
由于资源和处理速度，我们再从 400*400 中按照特定规则进行 scan 100*100 的像素模块。100*100 的像素块按照交叉原则选取，在满足信息不遗失的情况下，可以极大的缩短时间周期。条件 scan 模块，通



过采用多线宽匹配和 25 点特征匹配，采用欧氏距离进行计算误差，多次调试后并设定误差大小。



通过在 100*100 中扫描信息，若发现二维码特征后即可进行输出信息。



通过乐鑫 SDK ESP-IDF 连接 ESP32 与 AWS IoT 平台，fpga 输出是否识别到二维码的信息。

首先在 AWS IoT 控制平台创建事物、证书，记录终端节点地址，搭建 AWS 云环境。完成后对 ESP-IDF 开发环境进行搭建，步骤包括工具链安装，ESP-IDF PATH 路径设置，ESP32 开发板串口设置和程序烧录、运行、监控。

Toolchain 安装完毕后运行 mingw32.exe 打开 MSYS2 的终端创建一个 esp 目录作为开发 ESP32 应用的默认地址~/esp，接着进入到 esp 目录下，然后通过 GitHub 克隆 ESP-IDF。

工具链程序使用环境变量 IDF_PATH 来访问 ESP-IDF。将用户配置文件脚本存放在 C:/msys32/etc/profile.d/ 目录中。目的是每次打开 MSYS2 窗口时，系统都执行这些脚本。

确定 ESP-IDF 目录的路径，在脚本中文件 export_idf_path.sh 加入 export 命令并保存脚本。

紧接着使用 ESP-IDF 创建项目，建立 subscribe_publish 项目到 esp 目录并重命名为 aws_iot_demo，输入命令进入配置界面：

```
make menuconfig
```

在菜单中输入串口名称并设置 AWS IoT Endpoint Hostname。完成设置后进行编译和烧录，将证书，复制到根目录下运行，

```
make flash
```

在烧录成功后通过

```
make monitor
```

 命令来查看程序运行情况，并监测是否成功打印消息。

回到 AWS IoT 的控制台中，选择测试，订阅主题：test_topic/esp32，控制台下方会显示 ESP32 发送过来的消息。



第四部分

完成情况及性能参数 /Final Design & Performance Parameters

(作品完成情况, 附照片/视频链接)

第五部分

项目总结 /Conclusions

(项目中所用到的知识点, 项目收获及心得)

本项目的工作主要分为图像处理与云端通讯。图像处理部分我们了解并使用了一些常用的方案, 比如 rgb 色域的转化, 灰度图的展现, 并且在此基础上受到启发, 自主设计了识别二维码特征的算法。云端通讯部分, 主要了解了服务器端搭建环境与配置调试的过程。

第六部分

源代码/Source code

(作品源码 Github 链接, github 操作步骤见 *Github 项目上传简单教程*)

第七部分 (技术总结/心得/笔记等分享)

(可分享的技术文档附件或已发表的文章链接)

经过此次项目的设计与实践, 我们组最大的感受就是, 在项目设计之前应该先了解硬件资源, 当成熟的设计思想与可行的算法遇到受限的硬件资源也无计可施。在开发过程中, 我们原计划使用 1280*720 的 ram 存储图像数据, 并根据此设计了算法。当算法代码从 matlab 移植到 HLS 之后, 我们才发觉硬件平台的 ram 比较有限, 我们的算法需要三倍以上 ram 资源。不得不进行算法乃至方案的更改。导致时间紧迫, 项目完成情况落后于预期。