

第三届
全国大学生集成电路创新创业
大赛
CICIEC

基于 **ARM-M3** 的车牌识别系统设计报告

参赛题目： Arm 杯

队伍号： BJS82650

团队名称： 学电子不掉发

摘要

智能交通系统是将先进的信息技术、移动通信技术和计算机技术应用在交通网络，建设一种全方位的、实时准确的综合运输和管理系统，实现道路交通和机动车辆的自动化管理。对于车辆的识别检测、安全管理也提出了更高的要求。

车牌识别系统，采用的主要方法是通过图像处理技术，对采集的包含车牌的图像进行分析，提取车牌的位置，完成字符分割和识别的功能。车牌识别系统准确性主要受限于图像信息的获取，识别失败也大多数是由获取图像不理想导致。

本系统以智能交通系统为目标进行系列的应用开发，主要实现了图像数据的获取和预处理，车牌识别算法的设计，识别结果的图形化展示三个主要功能，形成了一个较为完整的车牌识别系统。在设计初期，我们利用 **Arm Cortex-M3 DesignStart** 处理器在可编程逻辑平台上构建片上系统，实现图像采集，图像处理和人机交互功能；之后是在 **FPGA** 平台上设计车牌识别的算法，使用流水线结构，实现车牌中字符的识别；最后是将识别的结果传输到 **LCD** 屏上进行显示，并通过 **ESP8266 WIFI** 模块将数据发送到 **APP** 端进行显示。

在创新点上，本项目在实现了准确性较高的车牌识别系统的同时，采用 **HSV** 格式用作图片二值化方法，获得了很好的区分效果，能够适应光线变化的不同场景，图像的提取和字符分割也取得了理想的效果，保证了车牌识别的正确率。在识别结果的展示上，不仅有 **LCD** 屏，同时还配套了云端的 **app** 方便更直观的对车牌识别的结果进行展示。

关键词： Arm -M3 内核 车牌识别 物联网

Abstract

The intelligent transportation system applies advanced information technology, mobile communication technology and computer technology to the transportation network, and build an all-round, real-time and accurate integrated transportation and management system to realize automatic management of road traffic and motor vehicles. Higher requirements have also been placed on the identification and detection and safety management of vehicles.

The main method of the license plate recognition system is to analyze the collected image containing the license plate through image processing technology, extract the position of the license plate, and complete the functions of character segmentation and recognition. The accuracy of the license plate recognition system is mainly limited by the acquisition of image information, and most of the recognition failures are caused by the unsatisfactory acquisition of images.

The system develops a series of applications with only the traffic system as the target. It mainly realizes the acquisition and preprocessing of image data, the design of license plate recognition algorithm, and the graphical display of recognition results. It forms a relatively complete license plate recognition system. In the early stage of design, we used the Arm Cortex-M3 DesignStart processor to build an on-chip system on the programmable logic platform to realize image acquisition, image processing and human-computer interaction functions. Then we designed the license plate recognition algorithm on the FPGA platform, using the pipeline structure. To realize the recognition of the characters in the license plate; finally, the result of the recognition is transmitted to the LCD screen for display, and the data is sent to the APP for display through the ESP8266 WIFI module.

KeyWords : ARM-M3 core , Internet of Things , license plate recognition system

目 录

一. 设计简介.....	- 1 -
1.1 系统总体介绍	- 1 -
1.2 系统总体流程	- 1 -
二 系统设计实现.....	- 2 -
2.1 系统硬件平台设计	- 2 -
2.2 车牌识别算法设计	- 3 -
2.2.1 车牌图像预处理.....	- 3 -
2.2.2 提取车牌区域.....	- 5 -
2.2.3 车牌字符分割.....	- 5 -
2.2.4 字符识别	- 7 -
2.3 外设驱动设计	- 9 -
2.3.1 SPI LCD 显示屏.....	- 9 -
2.4 上位机设计	- 9 -
2.4.1 上位机框架图.....	- 9 -
2.4.2 用户使用流程.....	- 9 -
三 项目功能展示.....	- 10 -
3.1 整体功能测试	- 11 -
3.2 OV5640 测试结果展示.....	- 11 -
3.3 车牌识别算法仿真.....	- 12 -
3.3.1 RGB 转 HSV.....	- 12 -
3.3.2 阈值二值化.....	- 12 -
3.3.3 投影分割	- 13 -
3.3.4 图像归一化.....	- 13 -
3.4 SPI LCD 屏测试.....	- 14 -

3.5 APP 使用测试展示.....	- 14 -
3.5.1 APP 与设备连接测试.....	- 14 -
3.5.2 APP 展示识别结果测试图	- 15 -
四 参考资料及文献.....	- 16 -
五 附录	- 17 -
5.1 系统框架 block-design 图	- 17 -
5.2 摄像头模组 block-design 图	- 18 -
5.3 图像算法模块 block-design 图.....	- 19 -

一. 设计简介

1.1 系统总体介绍

本作品是基于 ARM-M3 的车牌识别系统，包括 Xilinx Nexys4 DDR 开发板，OV5640 摄像头，SPI 驱动的 320x240 的 LCD 显示屏，ESP8266 WIFI 模块以及与识别系统配套用于显示车牌识别结果的 APP。

本系统采用在 FPGA 中搭建的 ARM-M3 软核作为中央处理器，使用 OV5640 摄像头采集图像，对图像进行预处理，然后定位车牌位置，分割车牌区域的字符，将每个字符缩放到统一大小，得到车牌中的 7 个包含字符的图像矩阵，然后使用模板匹配法进行字符识别，最后将识别出结果显示到 LCD 屏和 APP 上。

1.2 系统总体流程

如图 3-1 所示，系统上电后，先进行摄像头寄存器配置，然后将摄像头捕捉到的画面显示到 VGA 显示器上，同时定位车牌位置，分割车牌区域的字符，将每个字符缩放到统一大小，得到车牌中的 7 个包含字符的图像矩阵，然后使用模板匹配法进行字符识别，最后将结果显示至 LCD 屏幕上，并通过 WIFI 模块发送到云端 APP 上。

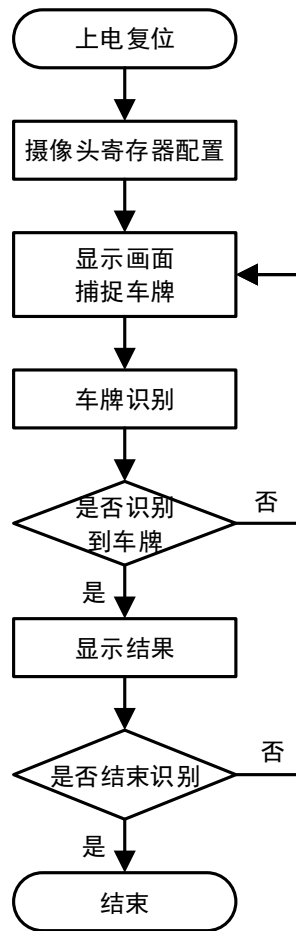


图 3-1 系统流程图

二 系统设计实现

2.1 系统硬件平台设计

本系统工作在 Nexys4DDR FPGA 板卡上，需要的外设为：OV5640 摄像头、LCD 显示屏和 ESP8266 WIFI 模块，分别实现图像采集、人机交互和数据上传到云端的功能。

系统的硬件框图如图 3-2：

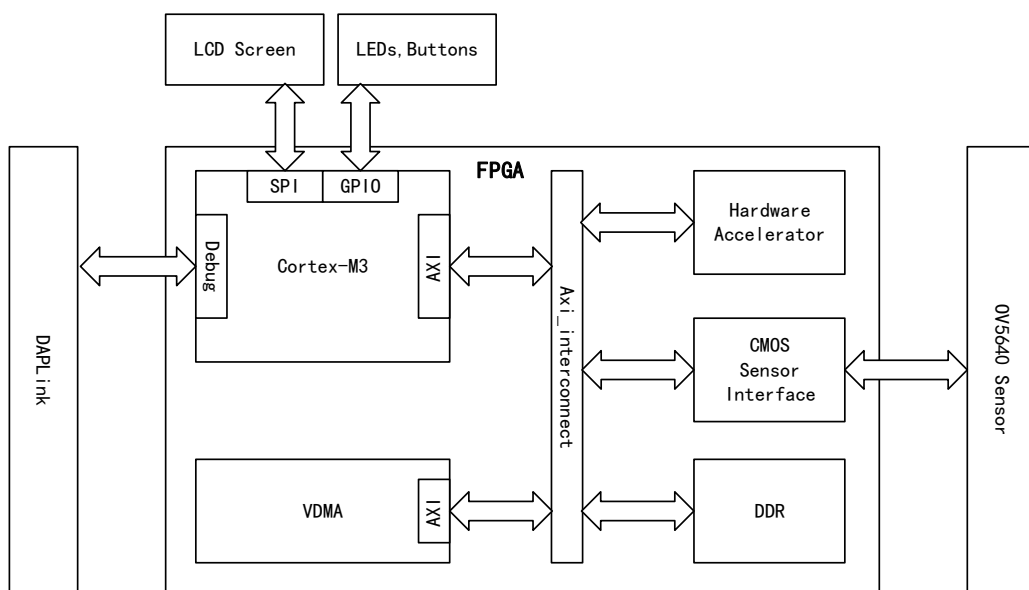


图 3-2 系统框架图

2.2 车牌识别算法设计

车牌识别的过程分为了四个部分，即车牌图像预处理、提取车牌区域、车牌字符分割和字符识别。车牌识别的算法主要通过在本 **FPGA** 中设计的硬件加速器来实现，**Cortex-M3** 在这部分主要是负责硬件加速器的配置与控制。

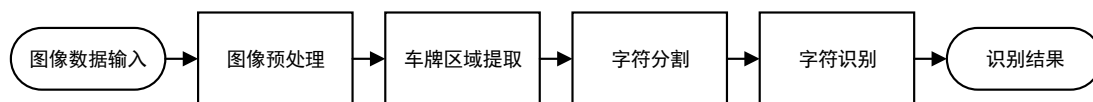


图 3-3 车牌识别算法流程

2.2.1 车牌图像预处理

结合题目的实际要求，将图像的预处理分为四步，即滤波处理、颜色空间转换、二值化、形态学滤波。

1. 滤波处理

在几种常用的滤波去噪方法中，中值滤波算法的去除椒盐噪声的能力更好，同时，它还能在去除噪声的基础上，比较完整得保留图像的边缘信息。由于摄像头获取到的原始图像中包含了大量的椒盐噪声，因此本设计中选用了中值滤波对原始图像进行去噪声处理。

中值滤波将模板内的像素按各通道颜色值的大小进行排序，然后将中间值赋给模板中心的像素，中值滤波的公式是：

$$f(x, y) = \text{median}\{g(x, y), (x, y) \in M\} \quad (3-1)$$

其中， $g(x,y)$ 和 $f(x,y)$ 分别是处理前和处理后的图像， M 为二维模板，这里，我们选用的是 3×3 的模板。中值滤波的效果如图 3-4 所示：



图 3-4 中值滤波效果图

2. 颜色空间转换

由于摄像头直接获取到的图像是 RGB565 的格式，其值受光线影响较大，直接对其进行二值化比较困难，因此我们采取将 RGB 格式转换为 HSV 格式，再进行二值化。HSV 代表色相 (Hue)、饱和度 (Sturation)、明度(Value)，本次要识别的车牌为蓝底车牌，只需要将色相的阈值限定在 200° - 280° ，饱和度和明度需要避开接近 0%和 100%的部分。

RGB 转 HSV 的公式为：

$$\begin{aligned}
 h &= \begin{cases} 0^\circ & \text{if } \max = \min \\ 60^\circ \times \frac{g-b}{\max-\min} + 0^\circ & \text{if } \max = r \text{ and } g \geq b \\ 60^\circ \times \frac{g-b}{\max-\min} + 360^\circ & \text{if } \max = r \text{ and } g < b \\ 60^\circ \times \frac{b-r}{\max-\min} + 120^\circ & \text{if } \max = g \\ 60^\circ \times \frac{r-g}{\max-\min} + 240^\circ & \text{if } \max = b \end{cases} \\
 s &= \begin{cases} 0 & \text{if } \max = 0 \\ \frac{\max-\min}{\max} = 1 - \frac{\min}{\max} & \text{otherwise} \end{cases} \\
 v &= \max
 \end{aligned} \tag{3-2}$$

由于在 FPGA 中实现时，图像的数据位宽是 8 位，因此我们将 h 除以 2 使取值范围为 $[0,180)$ ， s 和 v 乘 255 使其取值范围为 $[0,255]$ 。

3. 形态学滤波

在获取车牌的二值图像后，对其做一次开运算，以消除噪声。开运算是先对图像进行腐蚀运算，再对图像进行膨胀运算。



图 3-5 二值化与开运算

2.2.2 提取车牌区域

由于前面二值化的过程中，将车牌的蓝色背景转为白色，而车牌外的部分为黑色，因此我们用投影分割的方法来获得车牌四条边界的位置。分别将图像的像素在 x 轴和 y 轴方向求和，得到图像在两个方向上的直方图，直方图起始位置的值很小，而中间位置的值较大，可以设置一个阈值，来判断直方图的边界。我们将车牌边界外的部分做填充，这样图像中就只剩车牌中的字符为黑色，而别的部分都是白色。

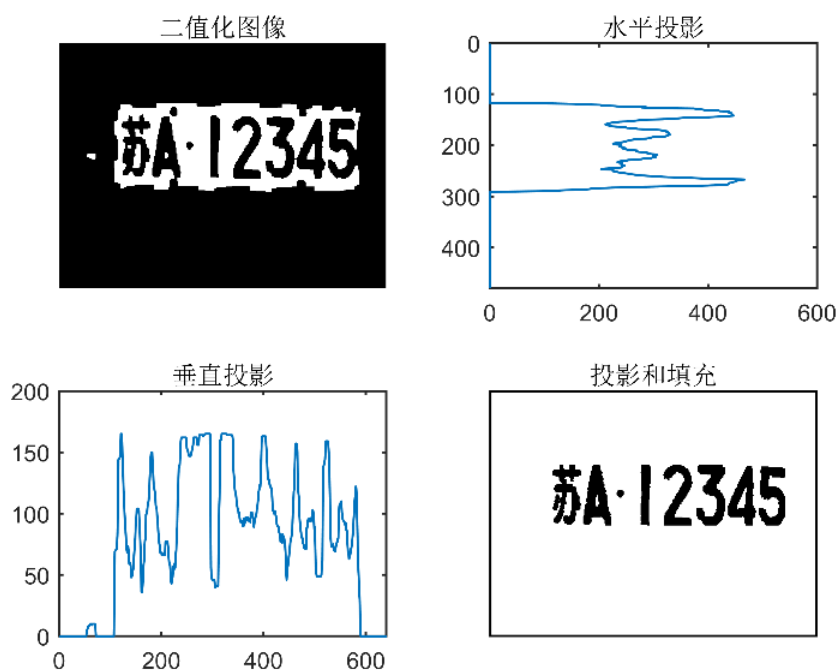


图 3-6 投影和填充

2.2.3 车牌字符分割

1. 传统投影法及其弊端

本设计中，结合车牌的字符特点，以垂直投影法为核心思想来分割车牌中的字符。传统的投影法是对图像进行垂直方向上的投影，统计每列像素中非零像素值的个数而绘制相应的直方图，然后根据图中波峰波谷的规律进行字符的分割。然而由于一些字符的中间像素点少，如 C、Q 等，传统的投影法会导致这些字符本身出现波谷，从而形成错误的分割方式。

2. 基于像素纵坐标极差的字符分割算法

根据对传统投影法的分析，本设计将统计非零像素个数的方式改为统计每列非零像素位置的最大值与最小值之差，及纵坐标的极差，从而避免了传统投影法中一些字符本身在投影曲线中会出现波谷的现象。极差字符分割的基本步骤如图 3-7:

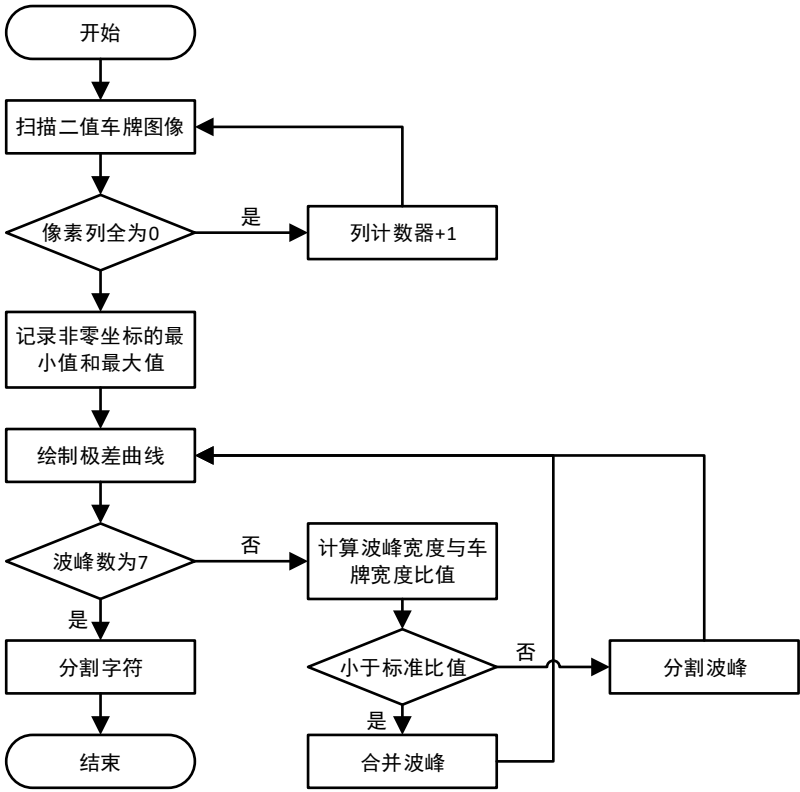


图 3-7 基于像素纵坐标极差的字符分割算法流程图

将极差曲线与传统投影法的曲线进行对比，效果如图 3-8 所示：

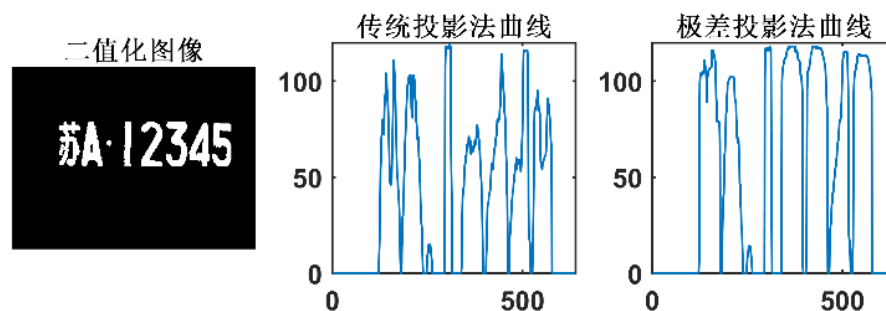


图 3-8 极差曲线与传统投影法的曲线对比图

最终字符分割的结果如图 3-9 所示：



图 3-9 字符分割结果

2.2.4 字符识别

据我国汽车车牌号的字符结构可知，车牌号字符包含汉字 32 种，即 4 个直辖市、5 个自治区和 23 个省会的简称，字母是去除 I 和 O 的 24 个英文字母以及 10 个阿拉伯数字。由此可见，字符种类只有三类，这也方便了字符的识别工作。本设计在研究模板匹配算法的基础之上，结合数理统计中利用相关系数度量两个变量之间的相关关系的思想，构造数学函数衡量二者相似度来达到识别车牌字符的目的。

1. 图像归一化

为了便于字符的识别，需要将每个字符缩放到统一大小。常用的图像缩放方法有近邻取样插值、双线性插值和三次线性插值，在这几种方式中，近邻取样插值实现起来最方便，但缩放后的效果最差，会出现阶梯状的锯齿；三次线性插值的效果最好，但其实现过程较为复杂，因此我们选择了双线性插值。

使用双线性插值，将每个字符都缩放到 28×28 的大小。由于整个图像处理过程是采用流水线的结构，因此每次只缩放一个字符，避免了暂停流水线，提高了图像处理的速度。

图像归一化的效果如图 3-10 所示：



图 3-10 归一化的字符

2. 模板匹配法及其弊端

目前模板匹配法是一种常用的字符识别算法。其基本原理是参照标准车牌字符的特征来构造一种模板，同时车牌字符要统一为标准的尺寸。以待分类的字符和模板的相似度为标准，相似度最大的那个模板对应的类就是该字符所属的类别。本设计中，使用模板与待分类字符的差值和来代表相似度，差值越小，则相似度越高。

相似度计算公式如下：

$$R(i,j)min = \sum_{i=0}^W \sum_{j=0}^H |A(i,j) - B(i,j)| \quad (3-3)$$

上式中的 $A(i,j)$ 和 $B(i,j)$ 分别代表待识别图像和模板图像， $R(i,j)$ 表示最小的差值。

然而这种模板匹配算法的缺点也比较明显，计算相似度的时候没有考虑像素点的位置信息，当模板类数增多时，准确度会下降。

3. HOG 特征提取与 SVM 匹配算法

考虑到模板匹配算法的缺点，我们计划使用 HOG 特征提取和 SVM 分类来器实现字符识别。

HOG 特征可以很好的反应图像的边缘信息，同时对图像的旋转不敏感，将原来字符的像素级别的信息提取为 HOG 特征，再用 SVM 分类器分类，可以提高字符识别的准确度。

目前以在 PC 平台上使用 OpenCV 验证了 HOG 特征 SVM 分类的识别准确度远高于传统的模板匹配算法，正逐步将其移植到 FPGA 平台上。

2.3 外设驱动设计

2.3.1 SPI LCD 显示屏

软件控制程序写于 **Cortex-M3** 中，初始化 **SPI** 与 **GPIO IP** 核，使其进入备用状态，之后通过 **SPI** 向 **LCD** 显示屏寄存器中发送初始化数据，初始化显示屏并设置显示模式，本实验采用竖屏模式显示。当识别到新的车牌号时，触发显示程序，向 **LCD** 显示屏发送需要显示的数据以及位置，屏幕分为 **10 行 2 列**，可以将 **20** 个车牌同时显示出来，屏满后自动清屏并从第一个位置开始重新显示。

2.4 上位机设计

2.4.1 上位机框架图

如图 3-11 上位机架构图所示，本系统将 **xilinx Nexys4** 开发板通过串口通信协议 **ESP8266WIFI** 模块进行通信，**ESP8266** 模块通过 **Agent** 协议与云服务器进行通信，云端 **APP** 通过调用云服务 **API** 获得下位机传输的数据，对数据进行处理后展示车牌以及包含的省市信息。

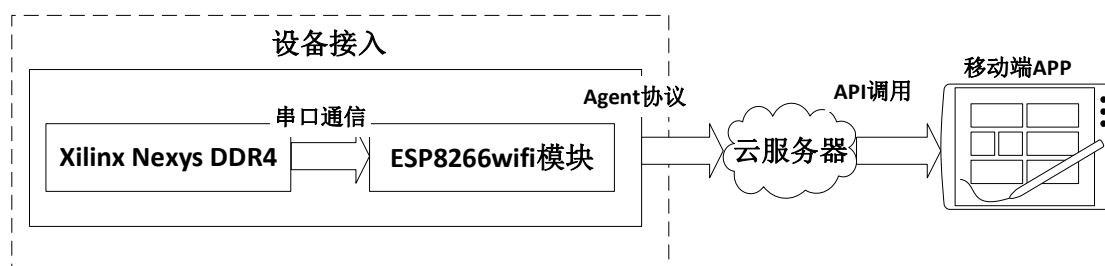


图 3-11 上位机框架图

2.4.2 用户使用流程

如图 3-12，用户通过 **APP** 端进行车牌识别结果展示时，首先通过局域网连接的方式绑定专属设备，在设备上线后可以通过设备详情页可以看到接受到的车牌信息，点击车牌信息可以获得省市具体信息。

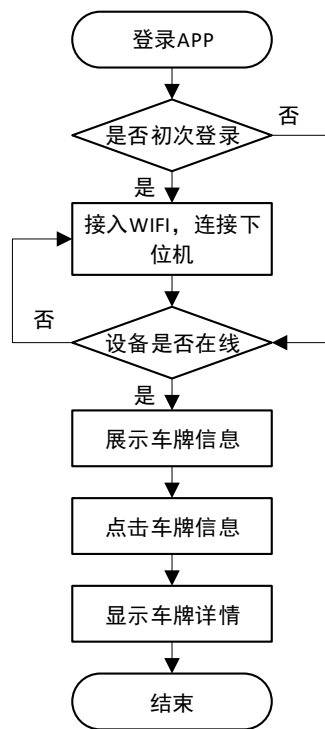


图 3-12 App 使用流程图

三 项目功能展示

3.1 资源消耗情况

本次设计使用的 FPGA 芯片是 Xilinx 公司的 XC7A100T-1CSG324C，最后的资源消耗情况如下：

Resource	Utilization	Available	Utilization %
LUT	52151	63400	82.26
LUTRAM	3919	19000	20.63
FF	56498	126800	44.56
BRAM	78	135	57.78
DSP	35	240	14.58
IO	103	210	49.05
BUFG	12	32	37.50
MMCM	4	6	66.67
PLL	1	6	16.67

3.2 整体功能测试

在光线环境较为理想的情况下，识别 20 张车牌大约需要 7 秒。测试环境如图 4-1 所示：

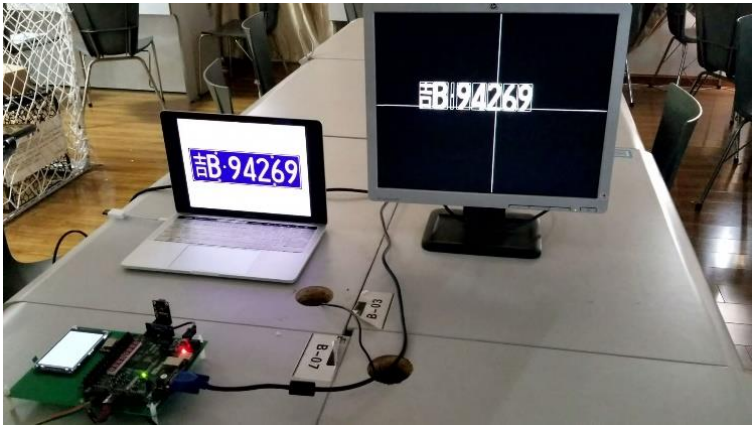


图 4-1 测试环境

3.3 OV5640 测试结果展示

OV5640 摄像头的 IIC 读写是通过 Xilinx AXI IIC 的 IP 核实现的，我们使用逻辑分析仪测试其结果，波形图如下图 4-2：

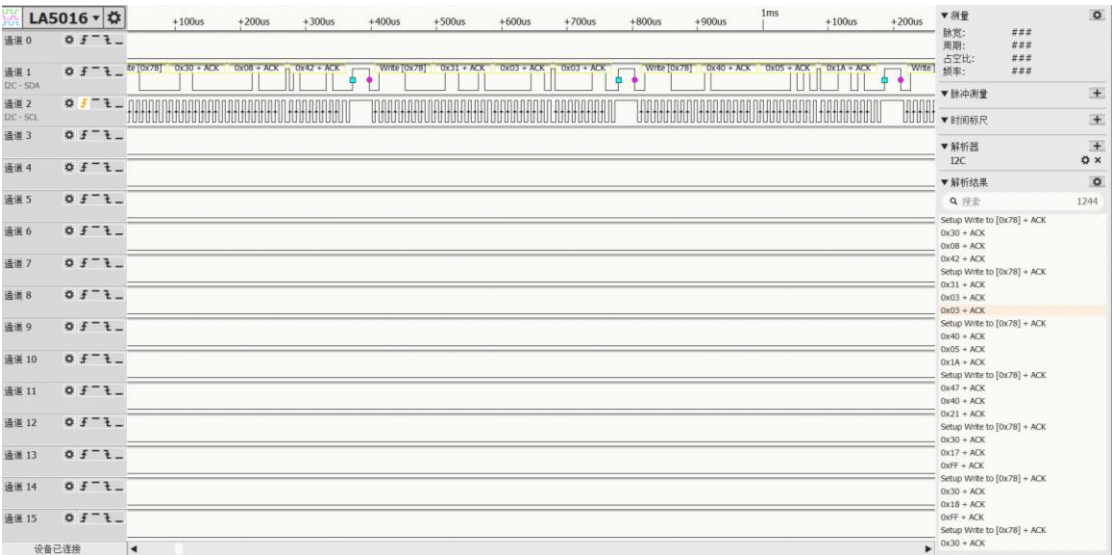


图 4-2 IIC 仿真波形图

可以看到寄存器配置数据被正确写入了 OV5640 的寄存器中,并且通过给地址 0x3019 写 0x02,再写 0x00,使 OV5640 的闪光灯闪烁一次,说明 OV5640 的寄存器被正确写入数据。

3.4 车牌识别算法仿真

车牌识别算法部分主要是通过 Xilinx 的 HLS 工具编写,可以将 C++代码转换为 RTL 代码,因此这部分的仿真主要使用 HLS 工具的 C/RTL Cosimulation 实现。

3.4.1 RGB 转 HSV



图 4-3 RGB 转 HSV RTL 仿真

在 HLS 工具中,运行 C/RTL Cosimulation,RTL 仿真的结果如上图,四张图依次为仿真输入图像,H 通道输出、S 通道输出、V 通道输出。

3.4.2 阈值二值化

在 HLS 工具中,运行 C/RTL Cosimulation,RTL 仿真的结果如下图 4-4,左边是原图输入,右边是对 HSV 图像进行阈值二值化输出的结果,可以看到,车牌

的蓝底被分离出来，另外画面中还有一些噪声，但相对与车牌占的比重很小，在下一步分割会被忽略。



图 4-4 阈值二值化图

3.4.3 投影分割

第一次投影分割将车牌的四个边沿提取出来，对车牌之外的部分进行背景色填充，从仿真图像中可以看到，这步之后，画面中只剩数字部分；

第二次投影分割只要进行垂直投影，从仿真图中可以看到每个字符都被正确分割出来了。



图 4-5 投影分割图

3.4.4 图像归一化

将每个字符缩放到 28*28 的大小。车牌中的点被缩放为一条横线，可以使用这个特征检测车牌是否存在。



图 4-6 归一化结果图

3.5 SPI LCD 屏测试

SPI LCD 逻辑分析仪波形如下：

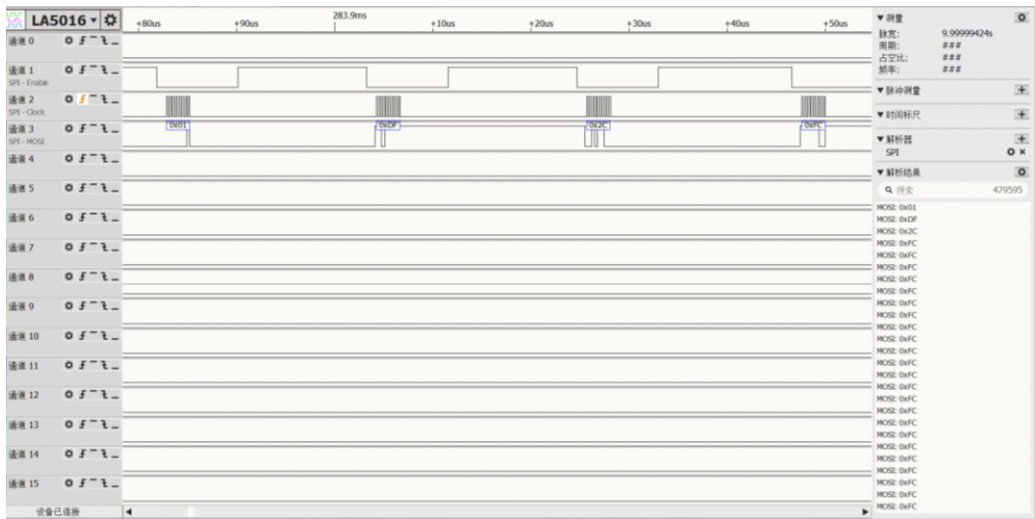


图 4-7 SPI LCD 逻辑分析仪波形

3.6 APP 使用测试展示

3.6.1 APP 与设备连接测试

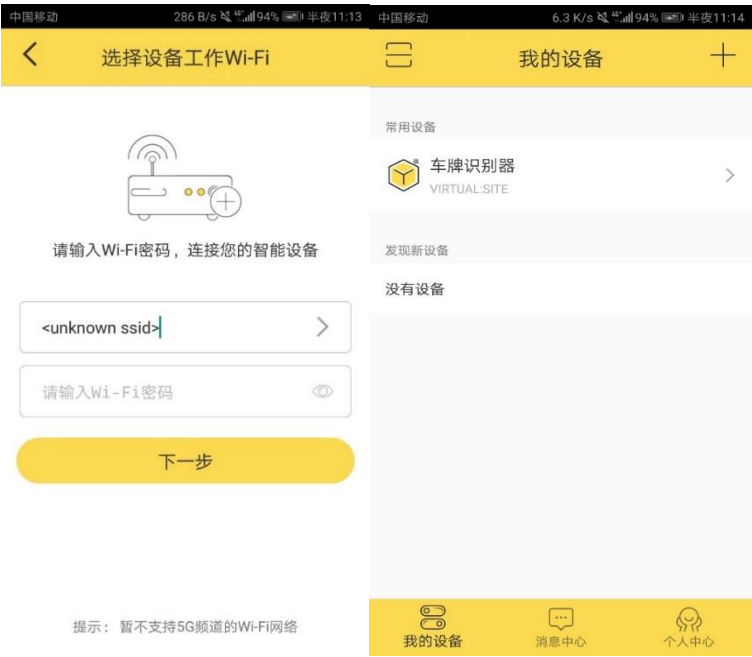


图 4-8 设备连接测试图

由上图 4-8 可以看出在同一局域网内对设备和 APP 进行配对，配对成功后 APP 首页可以展示设备名称同时可以进一步对设备数据进行管理

3.6.2 APP 展示识别结果测试图

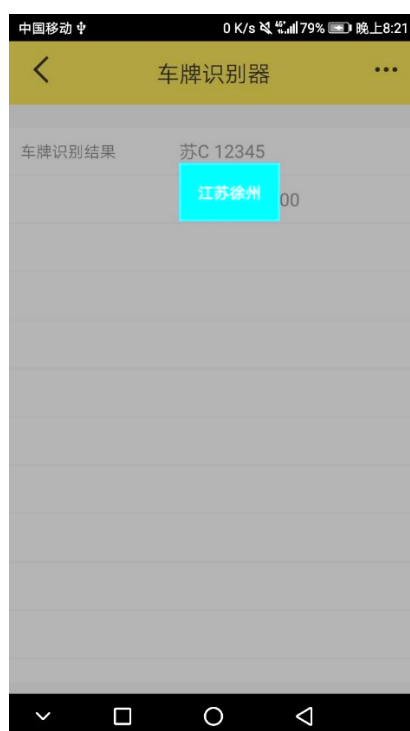


图 4-9 APP 接受数据图

如上图 4-9 所示,APP 在接受下位机数据后可以在设备展示页面进行展示，同时可以单击传输数据读取车牌代表的城市。

四 参考资料及文献

[1]王维. 基于 ZYNQ 平台硬件加速的车牌识别技术及其应用研究[D]. 北京工业大学, 2017.

[2]王凯. 基于 FPGA 的车牌识别算法的实现[D]. 福州大学, 2016.

[3]侯智. 基于 FPGA 的 SVM 车牌识别算法研究与实现[D]. 长春理工大学, 2016.

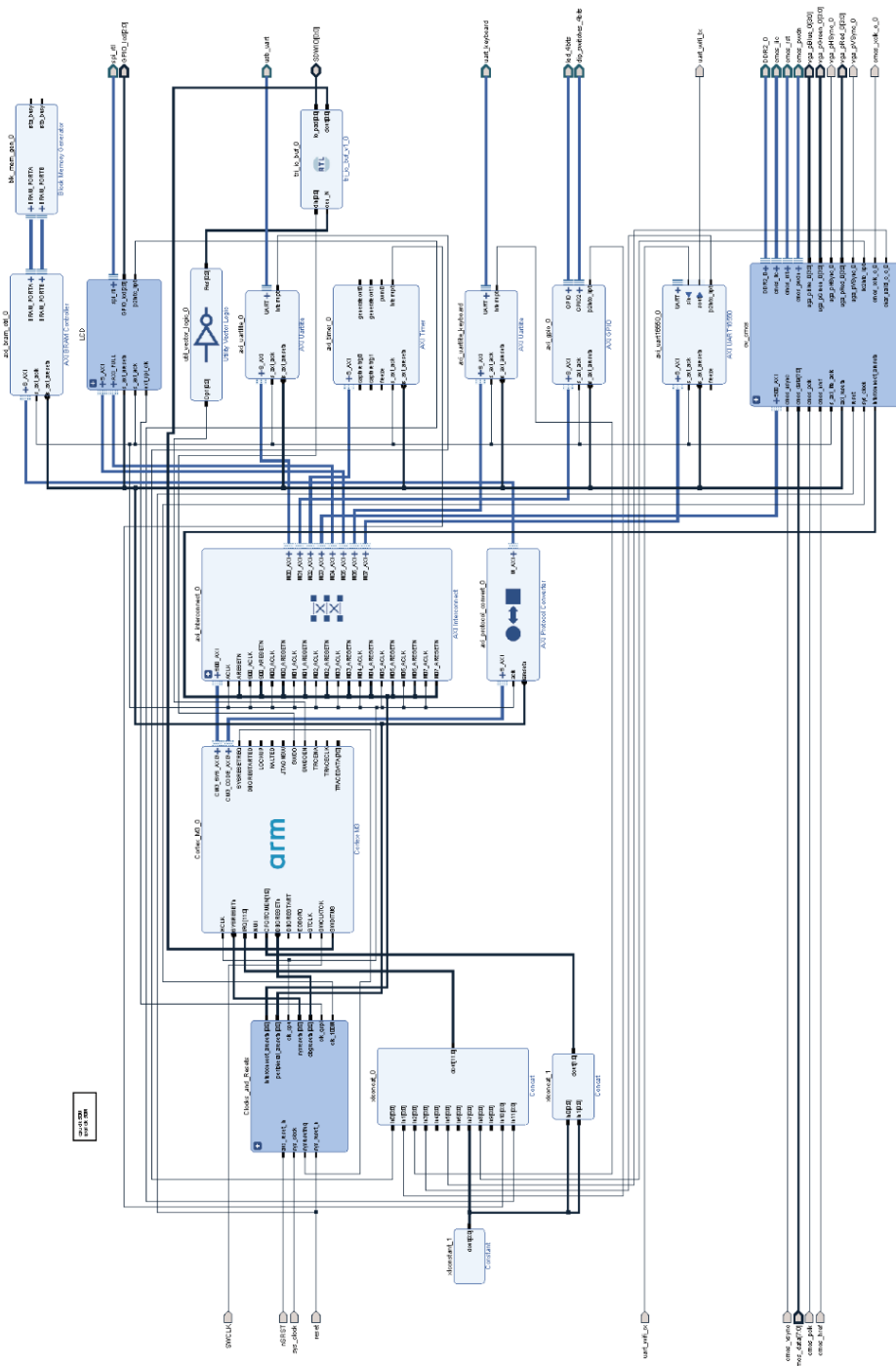
[4]高丽燕. 基于 DSP+FPGA 的图像识别系统设计与实现[D]. 南京理工大学, 2006.

[5]Karakaya, F., Altun, H., Cavuslu, M.A.. Implementation of HOG algorithm for real time object recognition applications on FPGA based embedded system[P]. Signal Processing and Communications Applications Conference, 2009. SIU 2009. IEEE 17th, 2009.

[6]高聪, 王福龙. 基于模板匹配和局部 HOG 特征的车牌识别算法[J]. 计算机系统应用, 2017, 26(01):122-128.

五 附录

5.1 系统框架 block-design 图



5.3 图像算法模块 block-design 图

