

基于高层次综合的 嵌入式轻量级车牌识别系统软件

前言

迎合当前 CPU+FPGA 异构计算的发展热潮，开发了一套基于 Zynq 架构与高层次综合的嵌入式轻量级车牌识别系统程序，要求在低密度低成本的 Zynq7000 系列的开发板运行，实现视频实时采集，车牌区域的定位，车牌字符的识别与显示识别结果等一系列功能，并具有高实时性与低能耗的特点。

以下就该程序作全面说明。

1 概述

该程序基于新兴的 Zynq SoC 异构平台，完成了视频采集、视频缓存，车牌图像处理和车牌字符识别的整个车牌识别任务过程。首先，摄像头采集到的视频数据作为程序的输入，程序的输出结果通过 HDMI 显示屏显示。其次，程序使用图像处理的方法，完成车牌图像的二值化处理，车牌区域定位，车牌字符切割等任务。最后，程序调用在 Zynq PL 端实现的车牌字符识别神经网络，对单个字符进行识别最终输出结果至板载 OLED 屏幕和串口终端。程序的整体架构如下图所示。

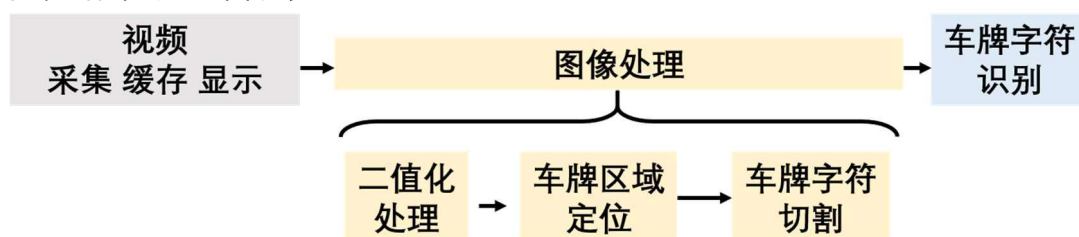


图 1 软件架构

程序程序具有较高的性能，车牌识别速率超过 10FPS，具有实时性好，功耗低等优点。

2 环境

2.1 运行硬件平台

该程序在基于 Zynq SoC 的 Zedboard 上运行。Zedboard 是第一款面向开源社区的 Zynq-7000 系列开发板，是 Xilinx 一个有重大意义的产品系列，标志着 Xilinx 在 SoC 集成度上的一个突破，实现了双核 Cortex-A9 MPCore 和最新的 28nm 7 系列可编程逻辑的紧密集成。

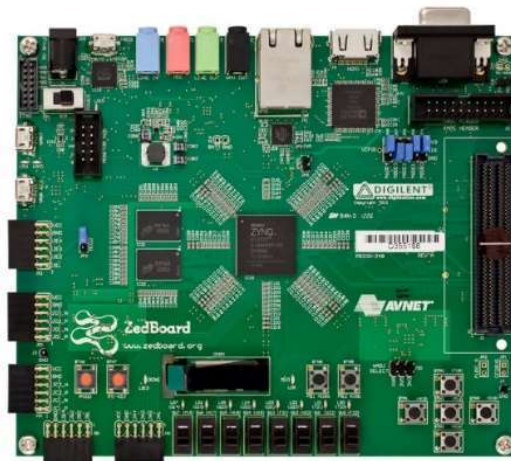


图 2 硬件平台 Zedboard

2.2 运行软件环境

该程序使用赛灵思的 Vitis 统一开发平台开发，并通过该开发平台的程序下载器将程序烧录至 Zedboard 运行。

Vitis 统一软件平台可实现在 Xilinx 异构平台（包括 FPGA、SoC 和 Versal ACAP）上开发嵌入式软件和加速应用。它可为边缘、云和混合计算应用加速提供统一编程模型。利用与高层次框架的集成，通过加速库采用 C、C++ 或 Python 进行开发，或者使用基于 RTL 的加速器和低层次运行时 API 对实现方案进行粒度更精确的控制。



图 3 Vitis 统一软件平台

3 主要功能

3.1 视频采集与实时显示

该程序驱动 OV5640 摄像头进行视频采集，采集到的视频时序信号转换成 AXI-Stream 总线协议的信号，经过 AXI VDMA 缓存至板载 DDR3 后，再通过 HDMI 驱动程序将 AXI-Stream 总线协议的信号转换成 HDMI 的差分信号，最终输出至 HDMI 显示屏，实现视频的采集和回显的功能。整个视频流数据的流向如下图所示。



图 4 视频数据总流向

3.2 车牌区域的定位

该程序对采集到的视频进行处理，通过形态学图像处理与广度搜索的算法，完成了车牌区域的粗定位，并在 HDMI 显示屏中用绿框将车牌区域框出。

3.3 车牌字符的识别

该程序对定位得到的车牌区域使用基于跳变阈值的方法，进行字符分割，并调用车牌字符识别神经网络对切割后的字符进行逐个识别，在板载 OLED 显示屏中显示车牌字符序列的识别结果，并通过串口返回上位机终端。

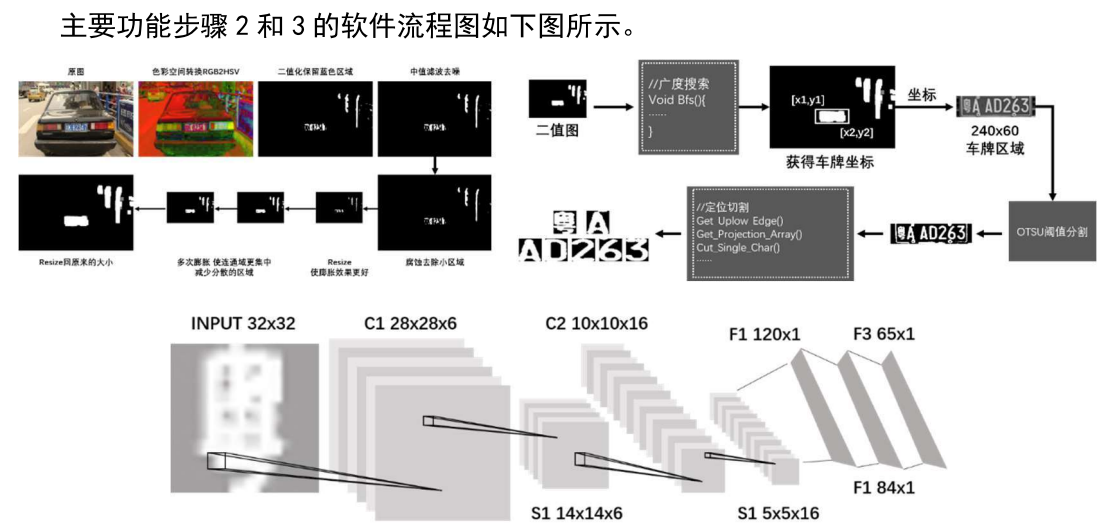


图 5 车牌区域定位，车牌字符识别的软件流程图

4 技术特点

- 1 程序功耗低，效率高
- 2 识别速度快，效果好
- 3 移植性强，可扩展性强

5 程序使用说明

5.1 硬件配置

5.1.1 硬件准备

运行程序之前，先要准备如下的基本设备：

- 1 0V5640 摄像头一个
 - 2 摄像头-PMOD 转接板一个
 - 3 Zedboard 开发板一套
 - 4 USB 连接线两条
 - 5 HDMI 显示屏一台
- 5.1.2 硬件连接

准备好所有的设备后，就可以开始硬件的连接。硬件连接的步骤如下：

- 1 将摄像头 PMOD 接头插入 Zedboard 的 PMOD A B 接口中
- 2 将显示器的 HDMI 接头与 Zedboard 连接
- 3 将 Zedboard 上的两个 USB 接口与电脑连接，其中一个作为 USB 用作串口输出，另外一个用作 USB 下载比特流对 Zedboard 进行烧录
- 4 最后将 Zedboard 与显示器分别和两个 12V 电源适配器连接
- 5 打开 Zedboard 的开关，准备运行程序
- 6 打开显示器开关，准备进行输出显示

硬件连接图如下图所示。



图 6 硬件连接

5.2 软件安装

该程序的运行环境与开发套件为赛灵思 2019.2 版本的 Vitis 统一开发平台。运行程序前，需要从赛灵思官网下载该软件平台并正确安装。

5.2.1 下载 Vitis 统一开发平台

登录赛灵思官网，注册账号并下载 Vitis 统一开发平台安装包。



图 7 Vitis 统一软件平台官网界面

5.2.2 安装 Vitis 统一开发平台

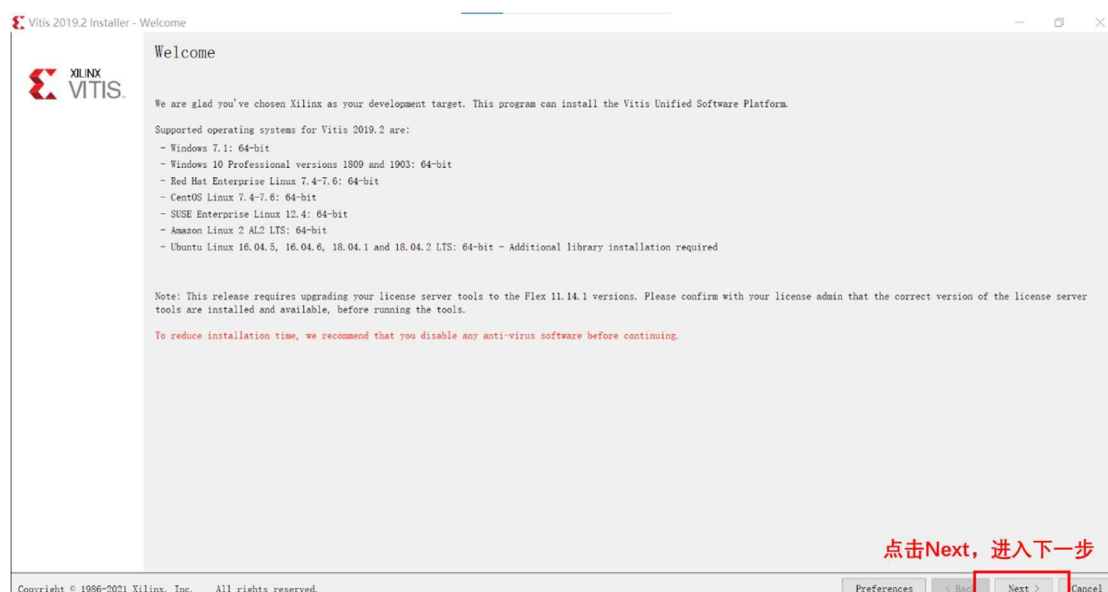
解压安装程序压缩包，进入安装包目录，双击 exe 执行文件，开始安装 Vitis 统一开发平台。

api-ms-win-crt-heap-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	20 KB
api-ms-win-crt-locale-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	19 KB
api-ms-win-crt-math-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	28 KB
api-ms-win-crt-multibyte-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	27 KB
api-ms-win-crt-private-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	70 KB
api-ms-win-crt-process-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	20 KB
api-ms-win-crt-runtime-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	23 KB
api-ms-win-crt-stdio-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	25 KB
api-ms-win-crt-string-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	25 KB
api-ms-win-crt-time-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	21 KB
api-ms-win-crt-utility-l1-1-0.dll	2019/11/7 13:10	应用程序扩展	19 KB
concrtd140.dll	2019/11/7 13:10	应用程序扩展	328 KB
msvcrt140.dll	2019/11/7 13:10	应用程序扩展	625 KB
ucrtbase.dll	2019/11/7 13:10	应用程序扩展	960 KB
vccorlib140.dll	2019/11/7 13:10	应用程序扩展	387 KB
vcruntime140.dll	2019/11/7 13:10	应用程序扩展	88 KB
xsetup	2019/11/7 13:10	文件	3 KB
xsetup.exe	2019/11/7 13:28	应用程序	439 KB

双击exe执行文件

图 8 Vitis 安装包目录

双击 exe 执行文件后，进入安装程序的开始界面，点击 Next



点击Next，进入下一步

图 9 Vitis 安装步骤 1

全部勾选 I Agree，同意要求的选项，点击 Next

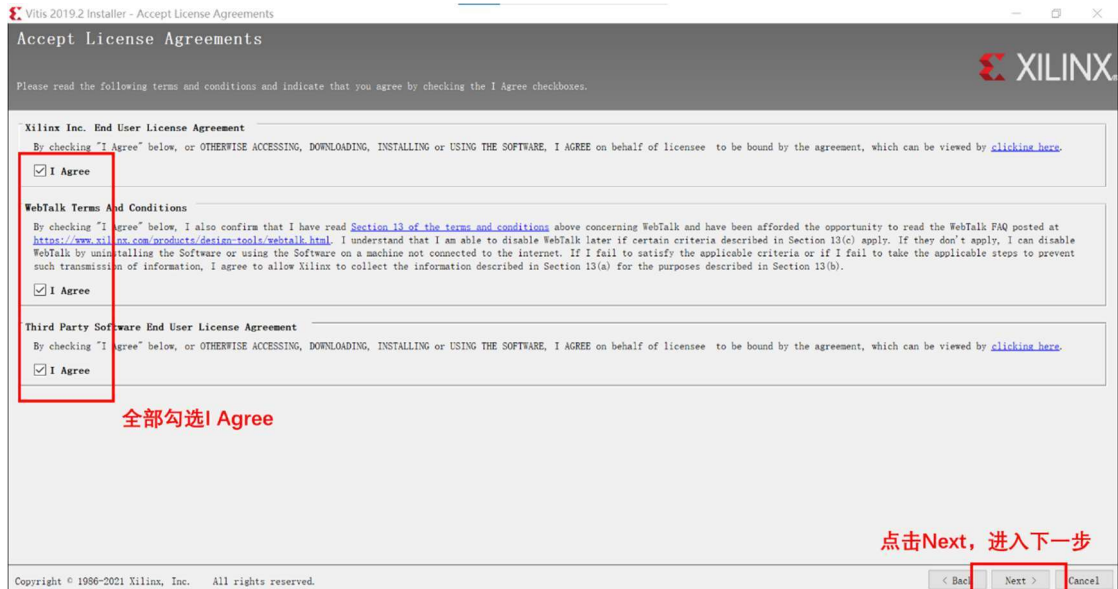


图 10 Vitis 安装步骤 2

选择安装目录，并且检查磁盘空间是否充足。Vitis 统一软件平台需要占用较大的硬盘内存空间。核实磁盘空间足够后，点击 Next 进入下一步

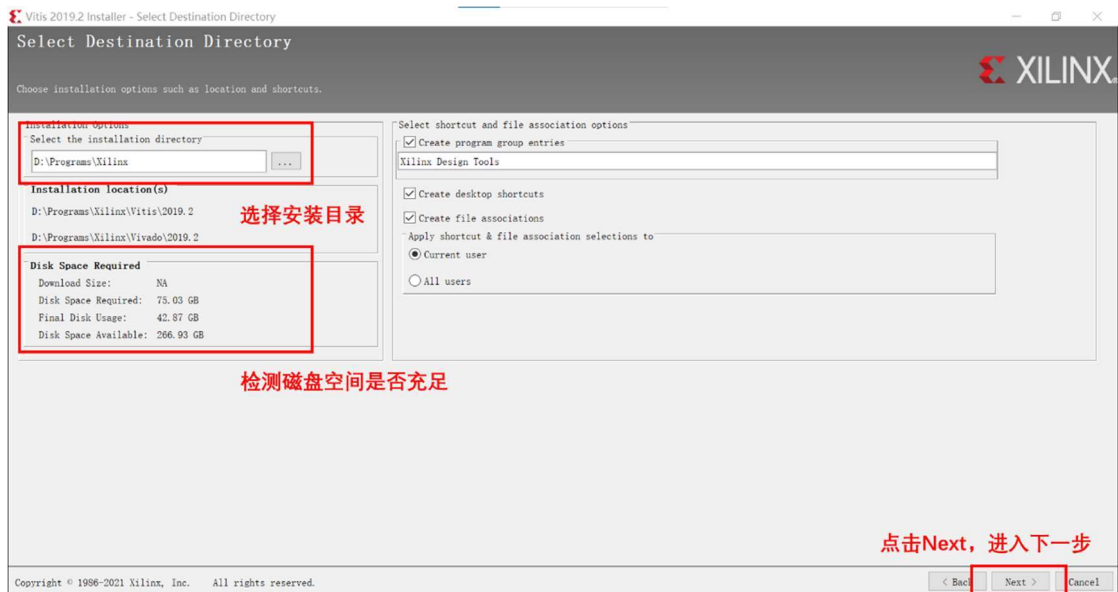


图 11 Vitis 安装步骤 1

完成安装步骤，等待 Vitis 统一软件平台安装完成。

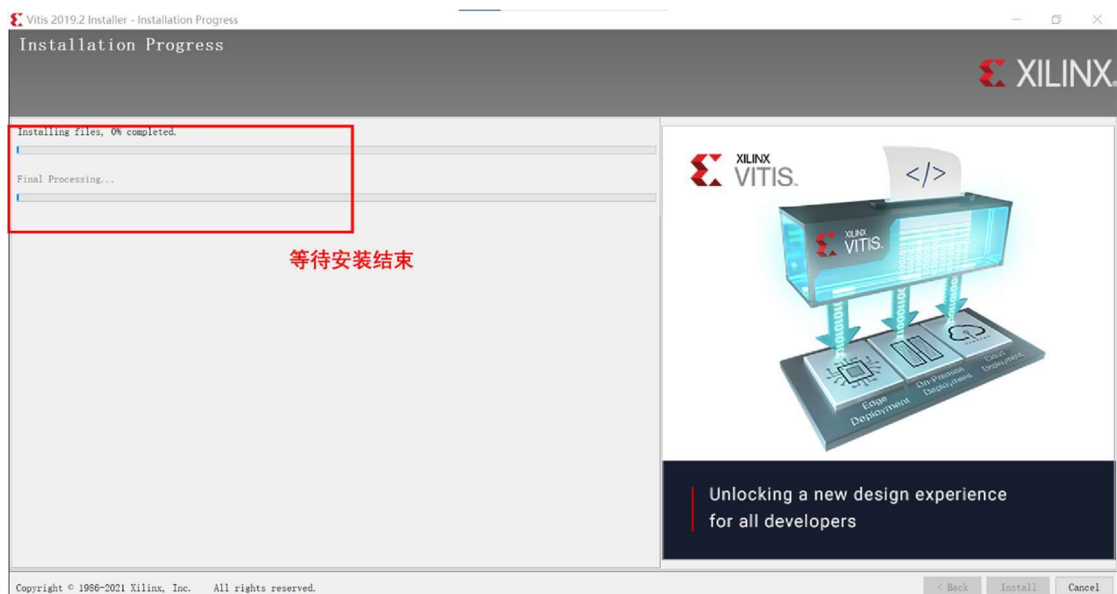


图 12 Vitis 安装步骤 3

5.3 程序运行

完成步骤 5.1 的硬件配置，步骤 5.2 的 Vitis 软件平台安装安装后，即可开始程序的运行。

程序的运行结果将会返回至串口终端。完成硬件配置后，在程序运行之前，需要首先打开串口助手。

选择开发板对应的 COM 口，设置波特率为 115200，如下图所示。

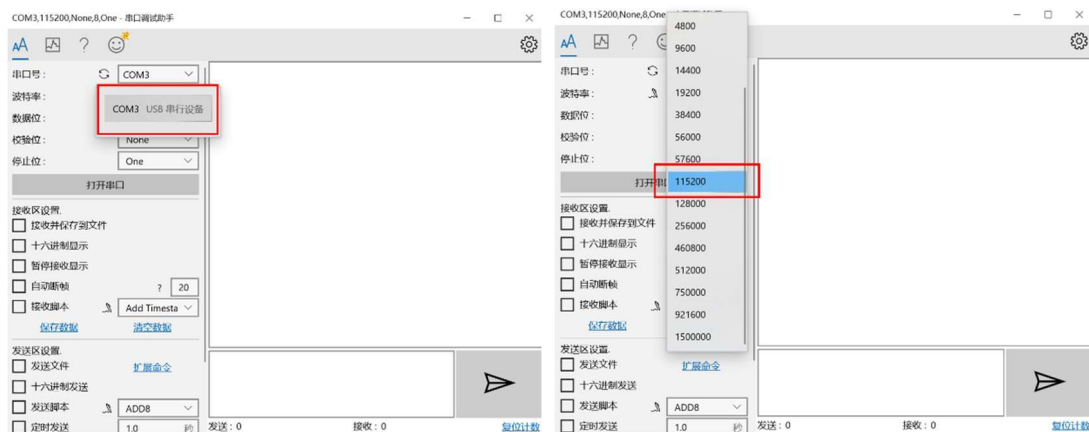


图 13 串口终端设置

COM 口和波特率设置好后，单击打开串口。该按钮由灰变绿，代表串口打开成功。

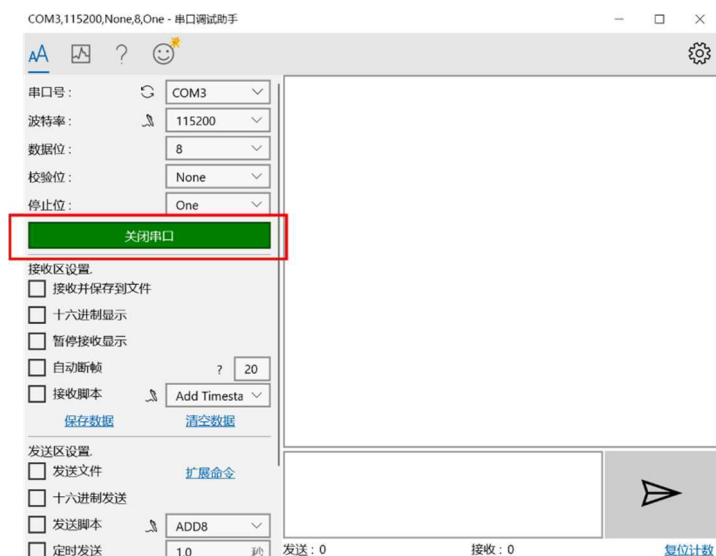


图 14 打开串口

串口终端打开后, 双击桌面 Vitis 2019.2 快捷方式, 或者在菜单栏中找到 Vitis 2019.2, 打开 Vitis 统一软件平台。此时会弹出工作目录指定窗口, 此时将目录指向程序所在的工程目录, 如下图所示。



图 15 工程目录

选择工程所在目录后, 点击 Launch, 等待程序启动。

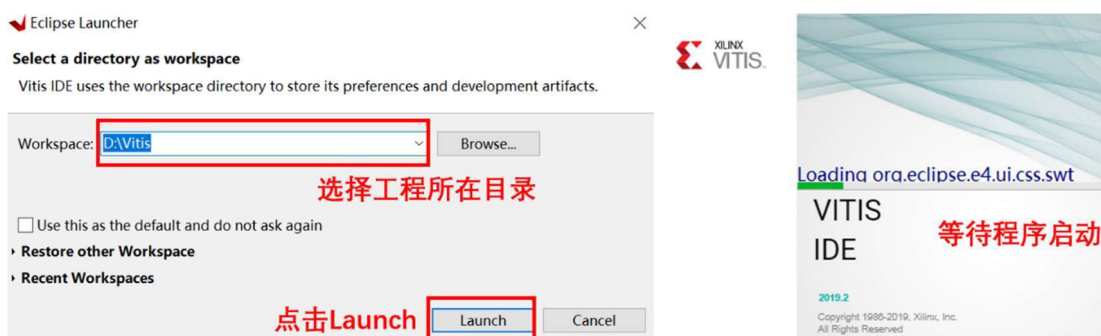


图 16 打开工程

程序启动完成后, 进入程序主界面, 如下图所示。

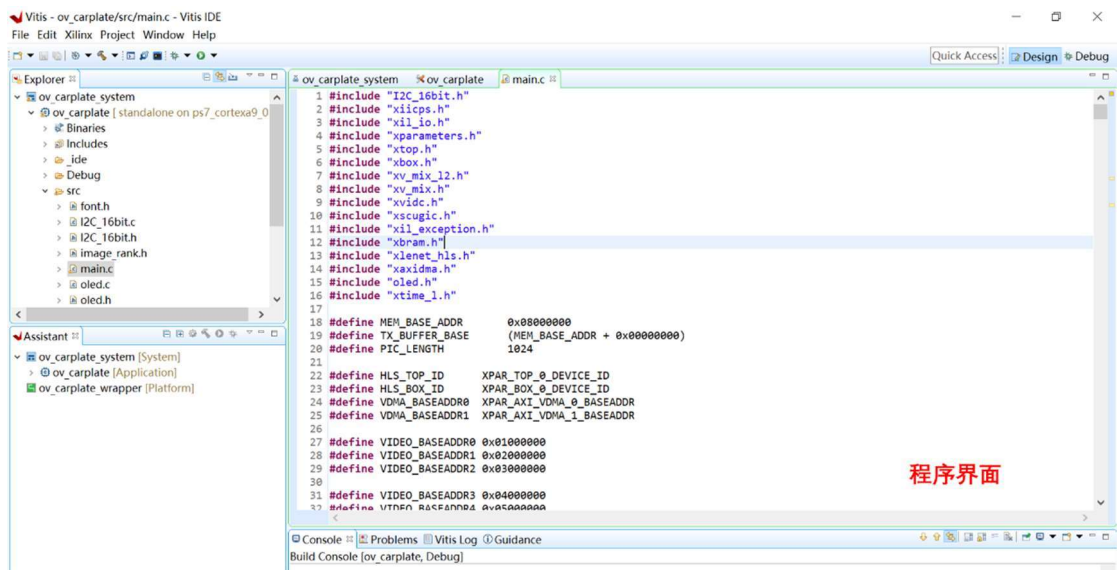


图 17 程序界面

再次确保硬件配置无误后，即可开始程序的运行。在左侧栏的目录中，选中当前的工程 `ov_carplate`，右键后选择 `Run As`，单击 `Launch On Hardware`。

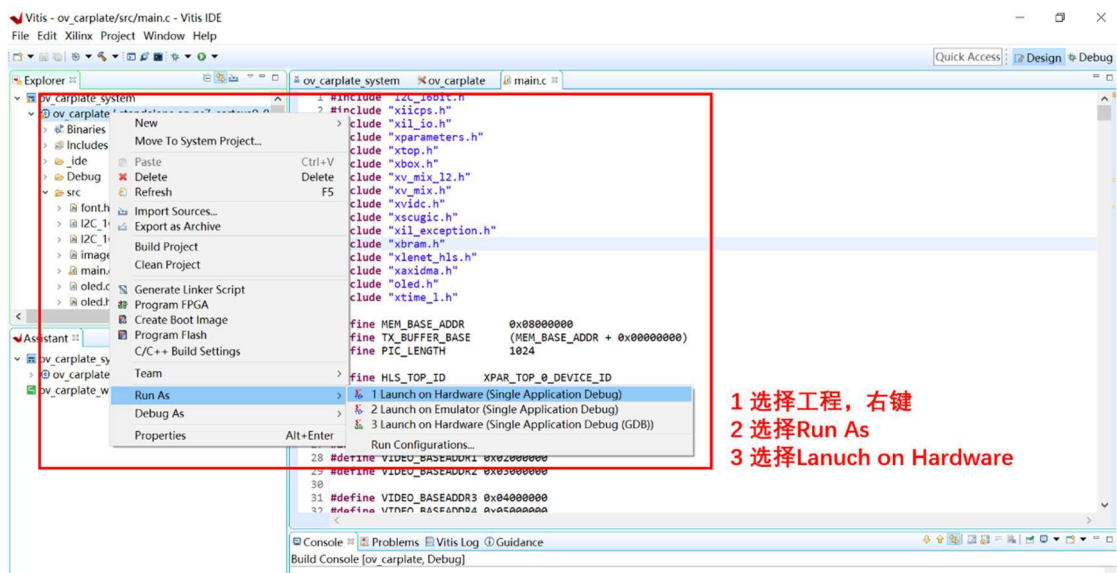


图 18 烧录程序

若硬件配置无误，则此时程序将会开始下载至开发板，如下图所示。

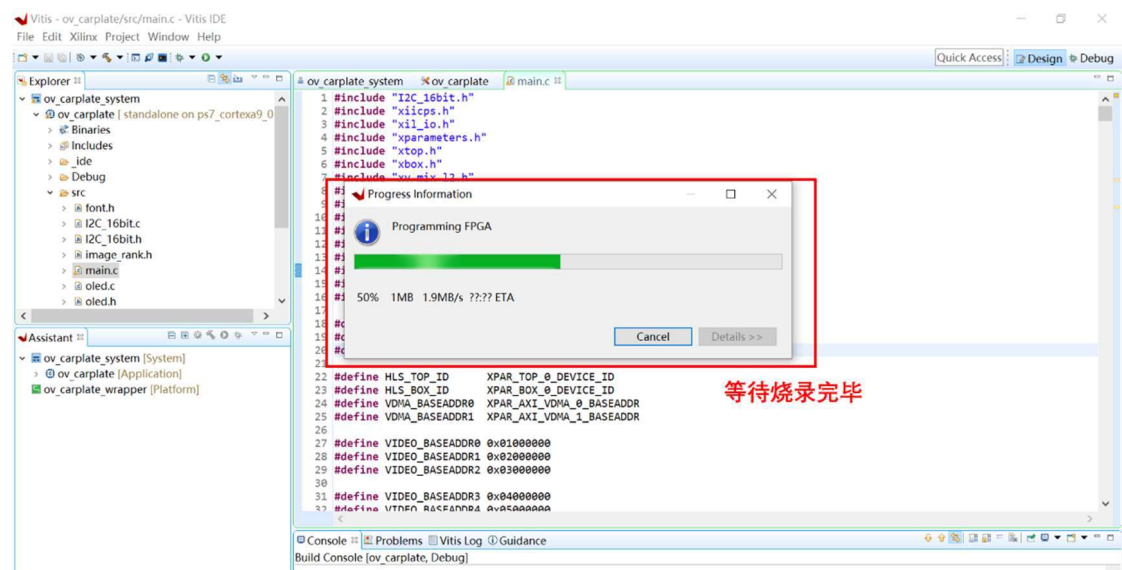


图 19 等待程序烧录

程序下载完成后，开发板的蓝灯亮起，说明程序下载成功，开始运行。

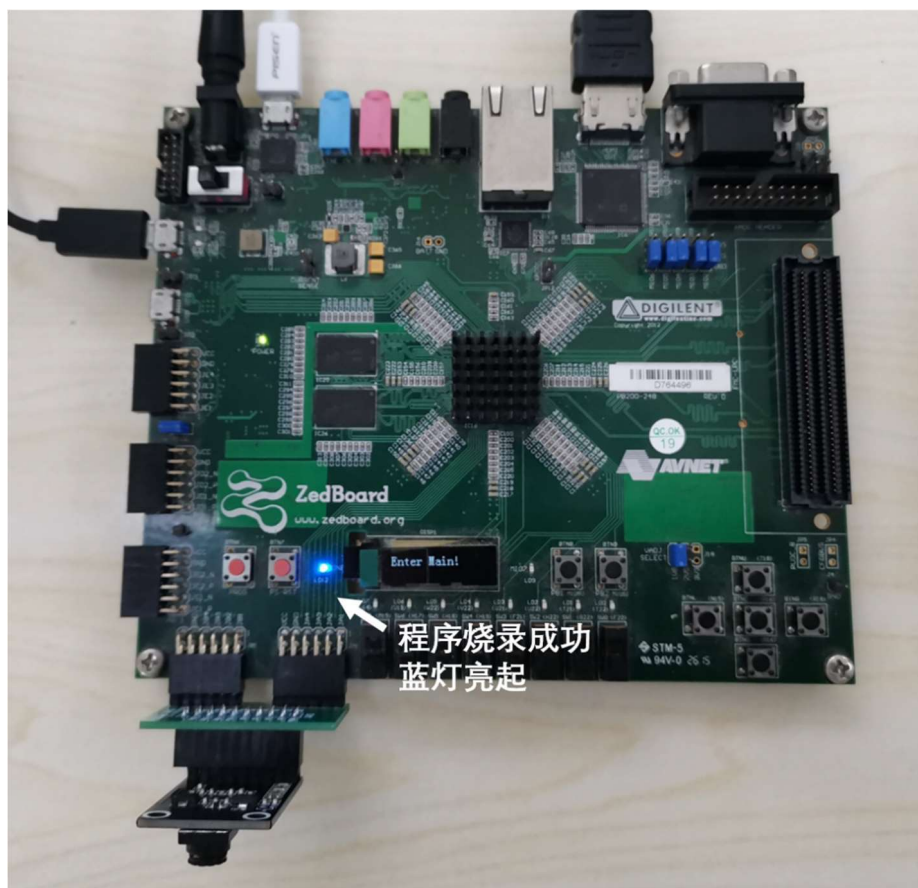


图 20 程序烧录完成蓝灯指示

此时，切换至先前打开的串口终端，发现串口终端输出一系列信息，代表程序初始化成功，如下图所示。

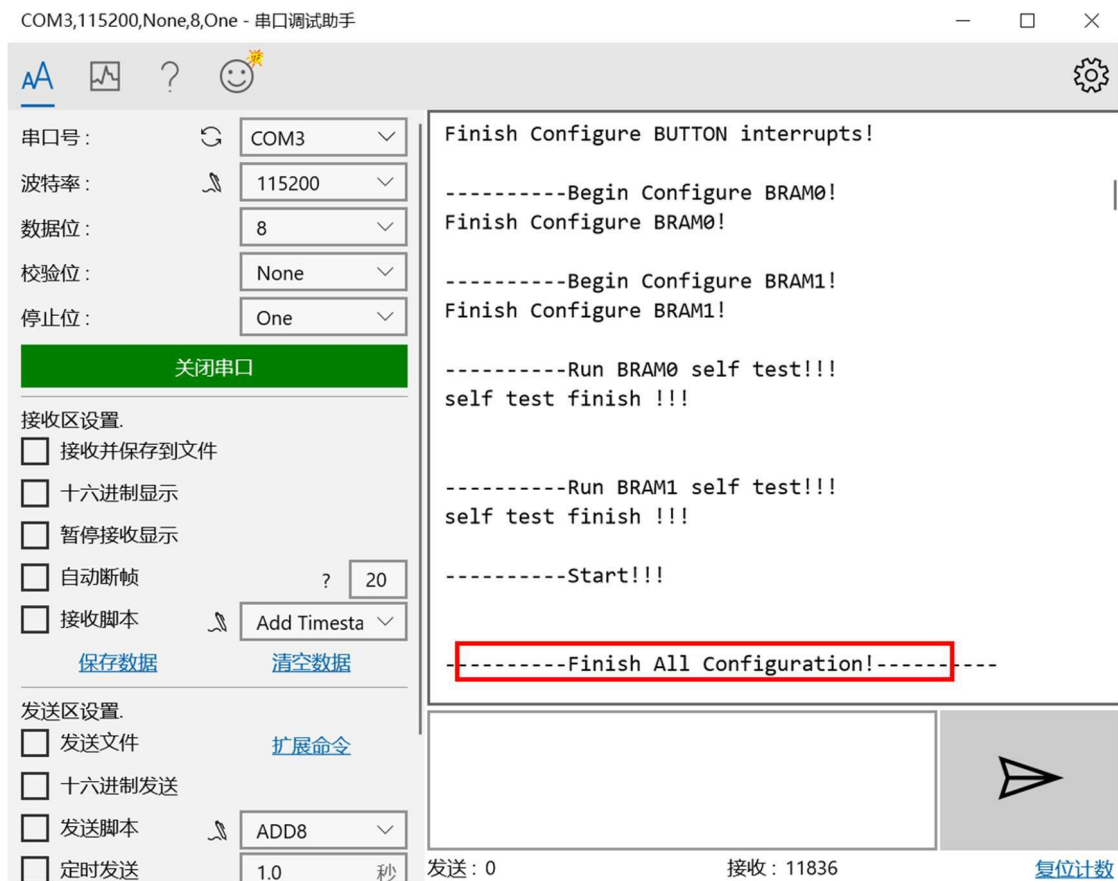


图 21 程序初始化成功

当前，程序正在运行。程序初始化后，默认运行在“视频采集-视频处理-实时显示”的模式。放置车牌图片在摄像头前，程序采集视频后，对视频进行二值化处理，并把处理后的结果实时显示在屏幕。车牌图片处理得到的二值化结果与车牌定位框如下图所示。



图 22 程序运行结果 1

而车牌定位结果的信息在串口终端也得到显示，如下图所示。

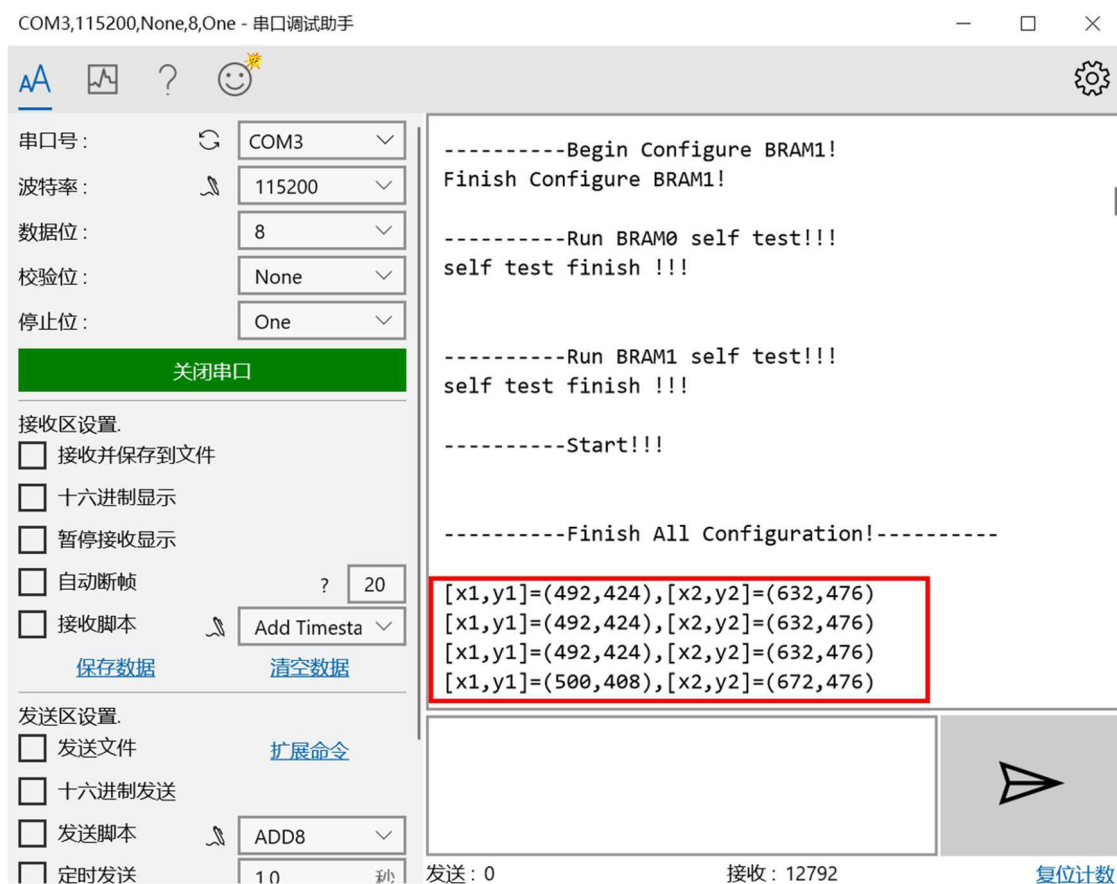


图 23 程序运行结果 2

通过波动板载按键，可以切换程序工作模式为“视频采集-实时显示”模式。此时，可以让程序显示当前未经处理的图像。板载按键位置如下图所示。

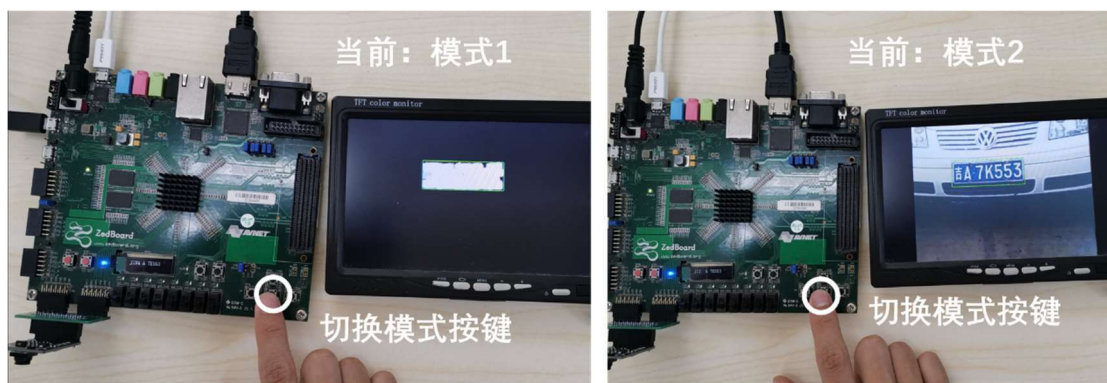


图 24 切换程序运行模式

程序接受到按键中断信号，切换工作模式，并在串口终端打印相关信息。

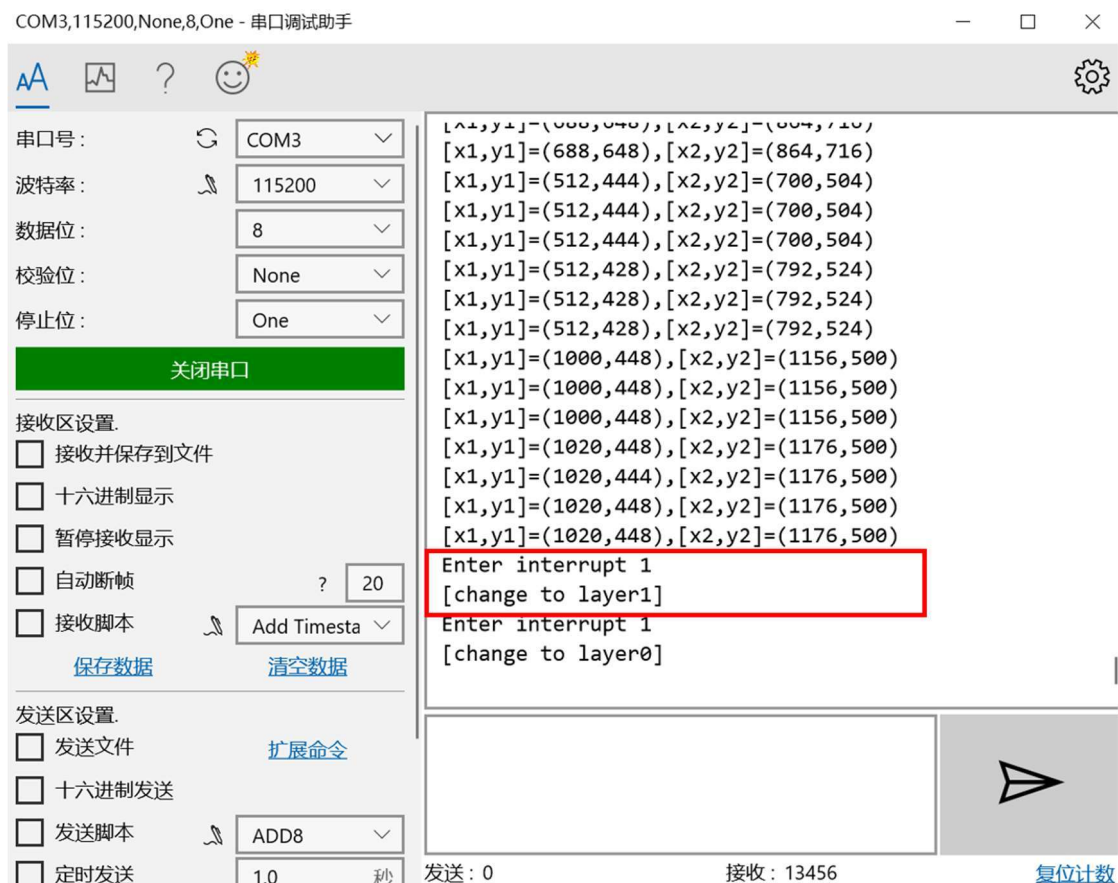


图 25 程序运行结果 3

在“视频采集-实时显示”工作模式下，程序开始进行车牌字符的识别。

观察开发板板载的 OLED 屏幕，识别得到的车牌字符结果在 OLED 上得到显示。

对照真实车牌图像区域与查找得到的车牌区域，我们可以发现二者高度重合，说明车牌区域定位查找精确。对照真实车牌字符序列与识别得到的车牌字符序列，发现识别结果正确无误，说明识别精度较高。

识别结果如下图所示。

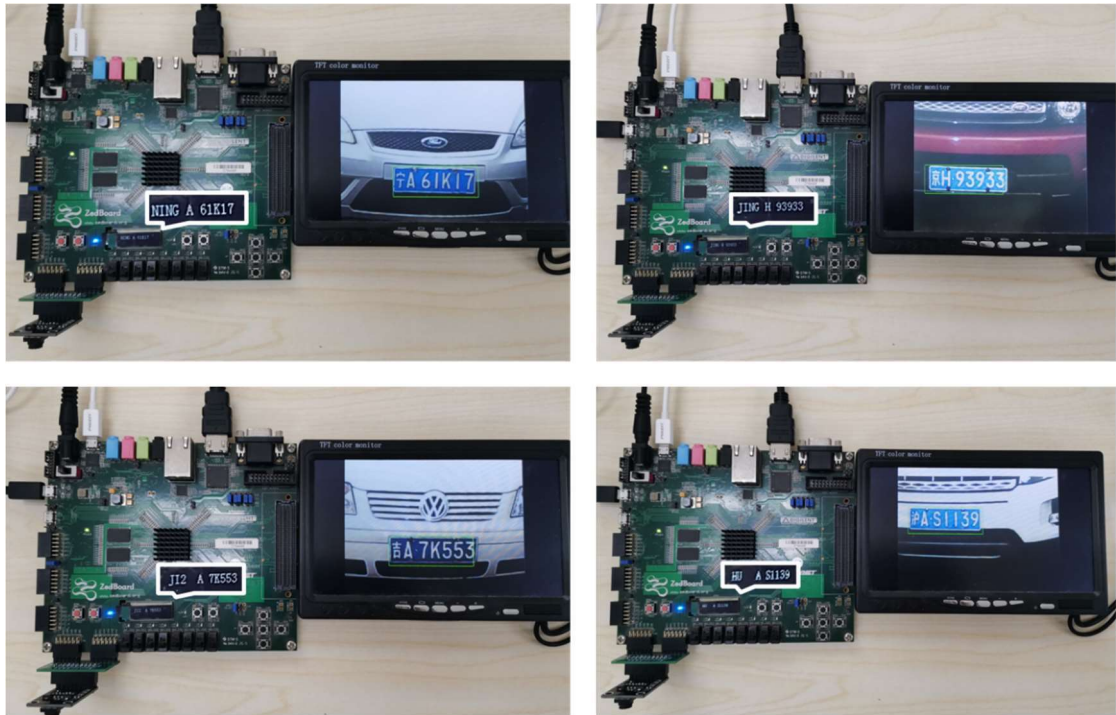


图 26 程序运行结果 4

5.4 程序退出

关闭电源开关即可退出程序。