项目申请书

个人信息

姓名：邓可笈

学校：西安电子科技大学

专业：电子与通信工程

入学年份：2019.9

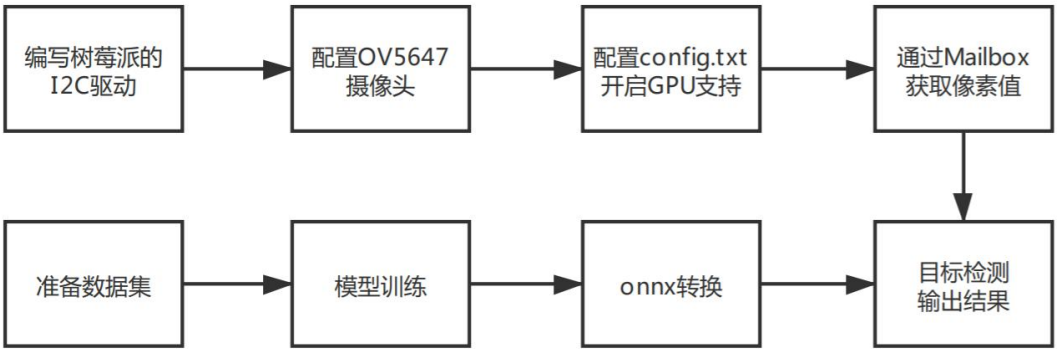
电子邮箱：13262679180@163.com

项目信息

申请项目社区：RT-Thread

申请项目名称：在树莓派 4 上用 RT-Thread 实现目标检测

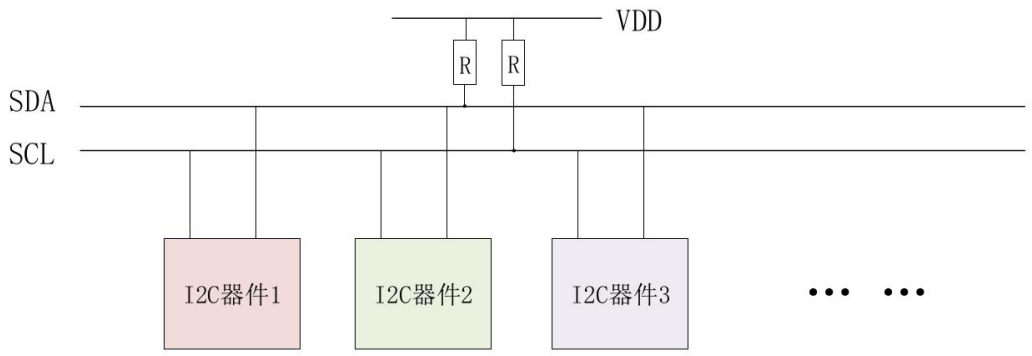
1.系统框图与设计思路



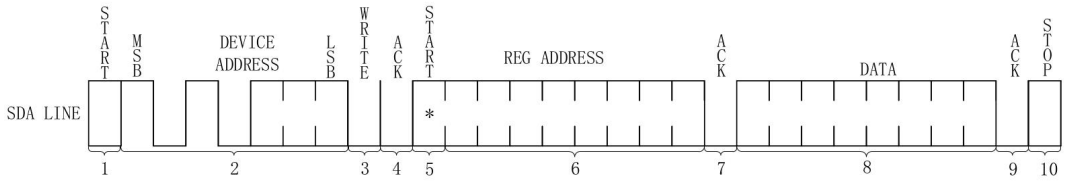
模仿 RT-Thread 树莓派 BSP 中的其它 driver 形式，基于 RTT 的 I2C 应用层驱动框架，实现 I2C 功能，可以通过 CSI 接口的 I2C 总线来配置摄像头的寄存器，从而进行图像采集任务。可以通过修改 config.txt 文件来配置启动选项，从而启动 VIdeoCore 固件，比如配置图像的大小、编码格式等。之后 ARM 端若想获取 VideoCore 接收的图像信息，需要通过 Mailbox 在指定地址处读取像素信息。将 yolo 训练后的模型，经过 onnx 转换，可以部署在树莓派上。

2.基于 RT-Thread 的树莓派 I2C 驱动

I2C 使用两条线在主控制器和从机之间进行数据通信，串行时钟线（SCL）和串行数据线（SDA）均需要上拉电阻，在它们空闲时保持高电平。I2C 支持多个从机，不同的设备有不同的器件地址，I2C 的主控制器可通过 I2C 设备的器件地址访问对应的 I2C 设备。



树莓派通过 I2C 向摄像头写入数据，配置相应的寄存器，I2C 总线单字节的写时序如下：



1）开始信号

2）I2C 设备地址（7bits）

3）I2C 读写位

4）从机发送的 ACK 应答信号

5）重新发送开始信号

6）要写入数据的寄存器地址

7）从机发送的 ACK 应答信号

8）写入寄存器的数据

9）从机发送的 ACK 应答信号

10）停止信号

Raspberry Pi4 具有主模式和快速模式（400kb/s）的 BSC 控制器（Broadcom Serial Control），BSC 总线符合 Philips I2C 总线。支持 7 位和 10 位的寻址，对于 HDMI 接口具有专用的 BSC2，在 linux 下通过 sudo raspi-config，在 Interfacing Options 中启用 I2C。

如果希望在运行 RT-Thread 时，通过 RTT 的 I2C 驱动框架来使用 I2C 设备，需要根据树莓派官方的寄存器手册来编写适用于树莓派 4 的 I2C 驱动。手册获取：

https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0.pdf

BCM2711 具有 BSC0 ~ BSC7，一共 8 个 BSC master，其中 2 和 7 用于 HDMI，不能被用户使用。通过某个 BSC master 的基地址上的偏移量就可以访问对应的 32 位寄存器。

Address Offset	Register Name	Description	Size
0x00	C	Control	32
0x04	S	Status	32
0x08	DLEN	Data Length	32
0x0C	A	Slave Address	32
0x10	FIFO	Data FIFO	32
0x14	DIV	Clock Divider	32
0x18	DEL	Data Delay	32
0x1C	CLKT	Clock Stretch Timeout	32

重要寄存器功能：

1）C 寄存器起主要的控制作用，使能 I2C 和 I2C 中断，设置 I2C 工作为主模式，选择传输方向是发送，以及使能传输应答为 ASK。

2）S 寄存器反映当前的 I2C 运行状态，比如可以获取到数据传输状态，FIFO 状态，ACK 状态等。

3）DLEN 寄存器定义在 I2C 传输中要传输或接收的数据字节数。读取寄存器给出当前传输中剩余的字节数。A 寄存器存放从设备的地址，FIFO 寄存器存放发送或接收的数据。

发送一个字节的流程为：

1）写入 1 至 DLEN 寄存器，表示发送数据的长度

2）将一个 8bit 值写入 FIFO 寄存器

3）向 A 寄存器写入从设备地址

4）将 C 寄存器的 READ 置 0，ST 置 1，开始传输

5）监测 S 寄存器 TA 位，确认传输开始

使用 RTT 的 I2C 驱动框架：

RT-Thread 的 I2C 驱动框架如官网文档：

https://www.rt-thread.org/document/site/programming-manual/device/i2c/i2c/

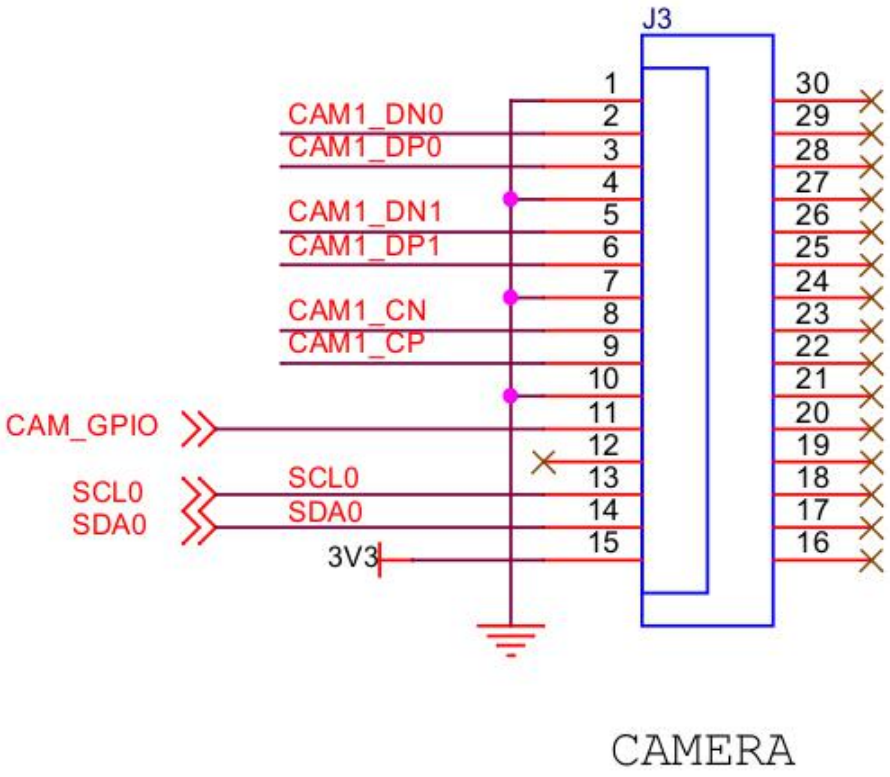
主要需要实现其中的 rt_i2c_transfer( )函数，将上述对树莓派 I2C 寄存器的配置封装成标准接口，在/bsp/raspberry-pi/raspi4/driver 中添加 drv_i2c.c 以及 drv_i2c.h

重要参考（移植树莓派 3B 中已经写好的 I2C、mbox、HDMI 等驱动）：

https://github.com/RT-Thread/rt-thread/tree/master/bsp/raspberry-pi/raspi3-64/driver

3.通过 I2C 配置 OV5647 摄像头

树莓派 4 的摄像头接口原理图：



通过 CAMERA 接口的 14 和 15 引脚，对摄像头里的寄存器通过 I2C 进行配置，返回带有像素信息的双路差分信号，通过 mailbox 通信取得像素信息。

选用树莓派官方的第一代摄像头 OV5647：

寄存器配置：https://datasheetspdf.com/datasheet/OV5647.html（待研究）

可以用过双路示波器去观察差分信号，来检验摄像头配置

4.通过 config.txt 配置 VideoCore 固件

树莓派没有传统意义上的 BIOS，系统配置参数都存在“config.txt”中，它存在引导分区上（/boot/config.txt），会在 ARM 内核初始化之前被 GPU 读取。

配置方法参考官方 Raspberry Pi4 的 BSP：

https://gitee.com/rtthread/rt-thread/tree/gitee_master/bsp/raspberry-pi/raspi4

修改烧录 SD 卡后的 config.txt 文件，将 kernel 改为 rtthread.bin 即可启动 RTT

为配置摄像头，需要参考：

https://www.raspberrypi.org/documentation/configuration/camera.md

树莓派官网提供关于摄像头的配置：

仅有 disable_camera_led = 1 这 1 项配置

https://shumeipai.nxez.com/2015/11/23/raspberry-pi-configuration-file-config-txt-nstructions.html

关于 HDMI 显示部分配置可以在 video 中找到，虽然研究 HDMI 配置并不是本次项目的主要需求，但可以通过配置 HDMI 的输出（编码、图像尺寸等），接上显示器来判断图像或视频是否获取成功。

5.通过 Mailbox 从 videocore 获取像素信息

ARM 通过 Mailbox 与 VideoCore 进行通信，每个 Mailbox 都是一个 8 位深的 32 位字 FIFO，可由 ARM 和 VideoCore 读取和写入。Mailbox0 用于 VideoCore 到 ARM 的通信，Mailbox1 用于 ARM 到 VideoCore。

MailBox 寄存器的使用和配置见：

https://github.com/raspberrypi/firmware/wiki/Accessing-mailboxes

树莓派 Mailbox 访问流程：

读取：

1. 读取状态寄存器，直到没有设置 empty 标志

2. 从读寄存器中读取数据

3. 如果低 4 位与所需的通道编号不匹配，则从 1 开始重复

4. 高 28 位是返回的数据

写入：

1. 读取状态寄存器，直到没有设置 FULL 标志

2. 将数据移至高 28 位，在低四位写入通道编号

6.目标检测的实现

选用单阶段法的经典目标检测模型 yolo-v3 或 yolo-tiny 训练基于人、交通载具等车路环境下的权重模型，然后使用 RT-Thread 已经集成了的 onnx 软件包进行转换。资料较多，就不展开叙述。

具体做法可以完全参考 RT-Thread 官网文档：

https://www.rt-thread.org/document/site/tutorial/smart-car/cnn-mnist/cnn-mnist/

将 yolo-v3 模型转换为 onnx：

https://blog.csdn.net/u013597931/article/details/89412272

7.参考

已在正文中标注

8.时间计划

2020 年 7 月 1 号 ~ 2020 年 9 月 30 日，总共 13 周

准备同步进行树莓派 3B 和 4B 的开发任务

第一阶段 I2C 驱动移植（7.1 ~ 7.14）

验收指标：可以通过 RTT 实现树莓派和外接开发板的 I2C 通信

社区提交：在树莓派 4 的 BSP 中添加 i2c.h 和 i2c.c，修改对应的若干 config 文件

第二阶段 摄像头寄存器配置（7.15 ~ 7.28）

验收指标：可以用过示波器观测到 DN 和 DP 引脚的双路差分信号

社区提交：一份 OV5647 摄像头的寄存器配置方案及说明

第三阶段 VideoCore 配置（7.29 ~ 8.11）

验收指标：可以通过 HDMI 观测到摄像头拍摄的照片或视频

社区提交：一份修改过的 config.txt

（中期验收及调整）

第四阶段 Mailbox 通信（8.19 ~ 9.8）

验收指标：可以在 ARM 端获取到摄像头拍摄的照片

社区提交：在树莓派 4 的 BSP 中添加 mailbox 的驱动及使用代码

第五阶段 目标检测（9.9 ~ 9.22）

验收指标：生活常用物体的实时目标检测

社区提交：train 代码，转换后的 onnx 模型

第六阶段 使用教程及终期报告（9.23 ~ 9.29）

验收指标：适用于 RTT 初学者的本系统使用教程

社区提交：一份最新的 BSP，将教程更新至社区官方文档，提交终期报告