

实验三：基于HiLens的口罩识别系统设计

实验目的

1

了解I2C的通信机制，掌握I2C设备驱动程序的开发

2

进一步熟悉Socket编程

3

了解使用HiLens在端设备进行智能应用开发的基本流程

实验内容

- 1、在Linux上编写OLED显示屏的I2C设备驱动程序：
 - 实现I2C设备驱动中的i2c_driver接口，包含dev_init()、dev_exit()、dev_i2c_remove()、dev_i2c_probe()、dev_i2c_register()函数以及i2c_device_id设备ID表
 - 实现I2C设备所挂接的OLED设备的读写操作：使用Page Addressing Mode来写OLED显示屏，因此需要实现基于buffer块的OLED读/写操作函数
- 2、使用HiLens Studio新建口罩识别技能，并开发嵌入式服务器和客户端
 - HiLens进行口罩识别，识别结果传递给GW3399，由GW3399将结果显示在8×8 LED点阵和OLED屏幕上

PART-A: I2C设备驱动开发

实验原理 —— I2C协议

- I2C总线(Inter-Integrated Circuit Bus) 最早是由Philips半导体（现被NXP收购）开发的一种简单、双向二线制同步串行总线。它只需要两根线（串行数据线SDA和时钟线SCL）即可在连接于总线上的器件之间传送信息

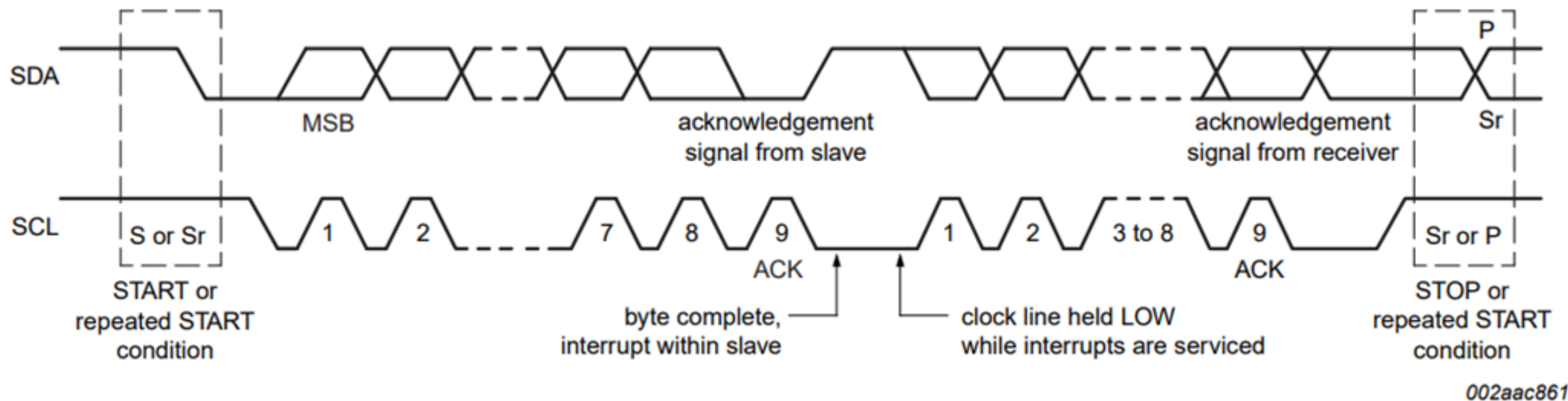
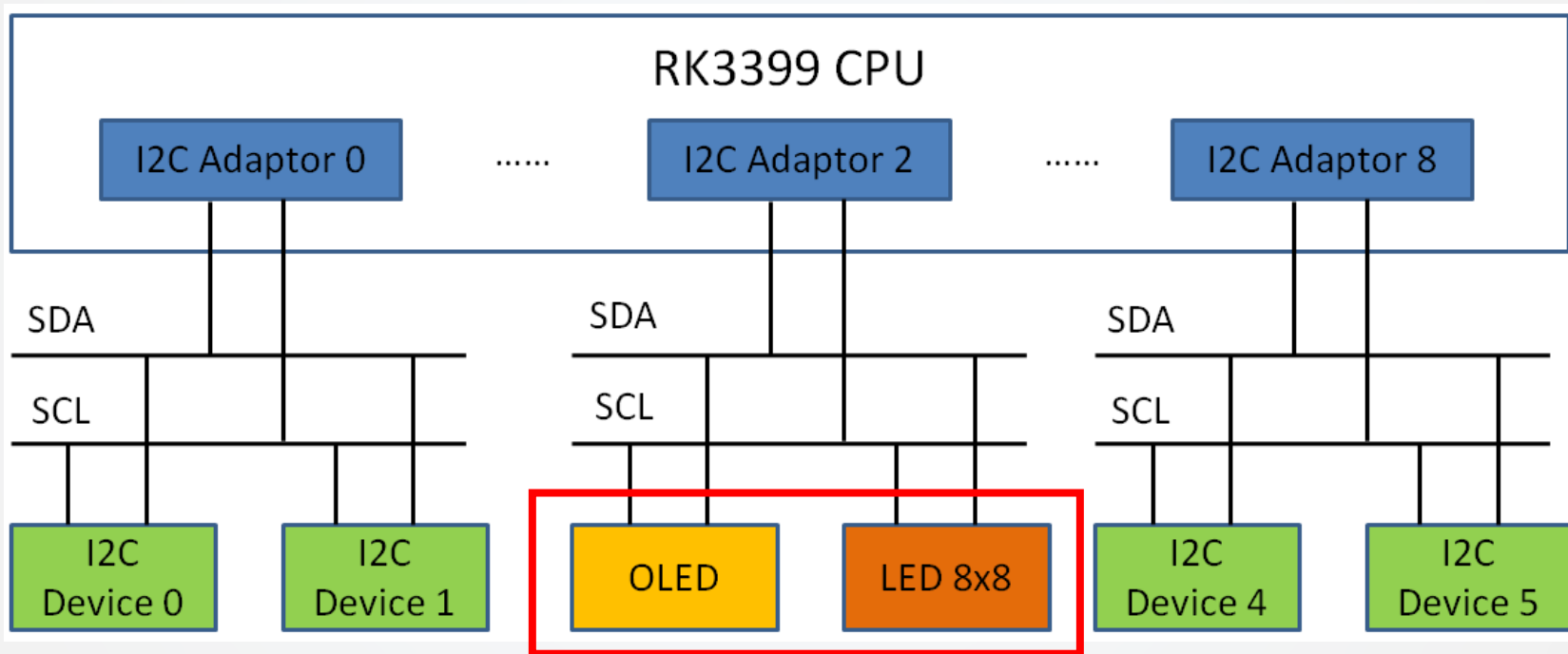


Fig 6. Data transfer on the I²C-bus

实验原理 —— RK3399 I2C接口

- RK3399提供了9组I2C总线接口 (I2C0 ~ I2C8) , 每组接入的从设备地址不能有相同
- OLED和8×8 LED点阵都挂到I2C2 (I2C Adaptor 2) 总线接口上



OLED从设备地址: 0x3C

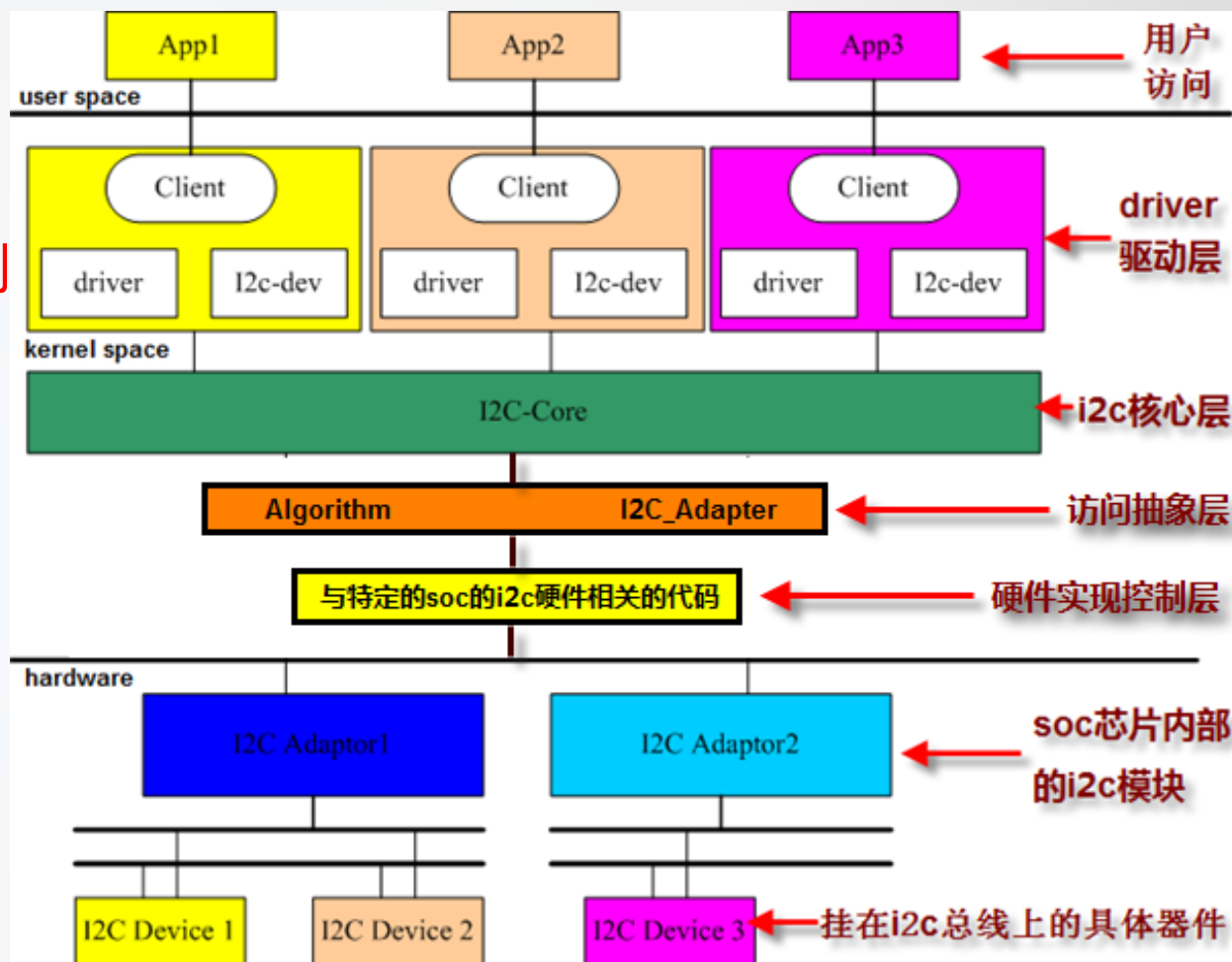
b7	b6	b5	b4	b3	b2	b1	b0
0	1	1	1	1	0	SA0	R/W#

8x8 LED点阵从设备地址: 0x70

b7	b6	b5	b4	b3	b2	b1	b0
1	1	1	0	0	0	0	R/W#

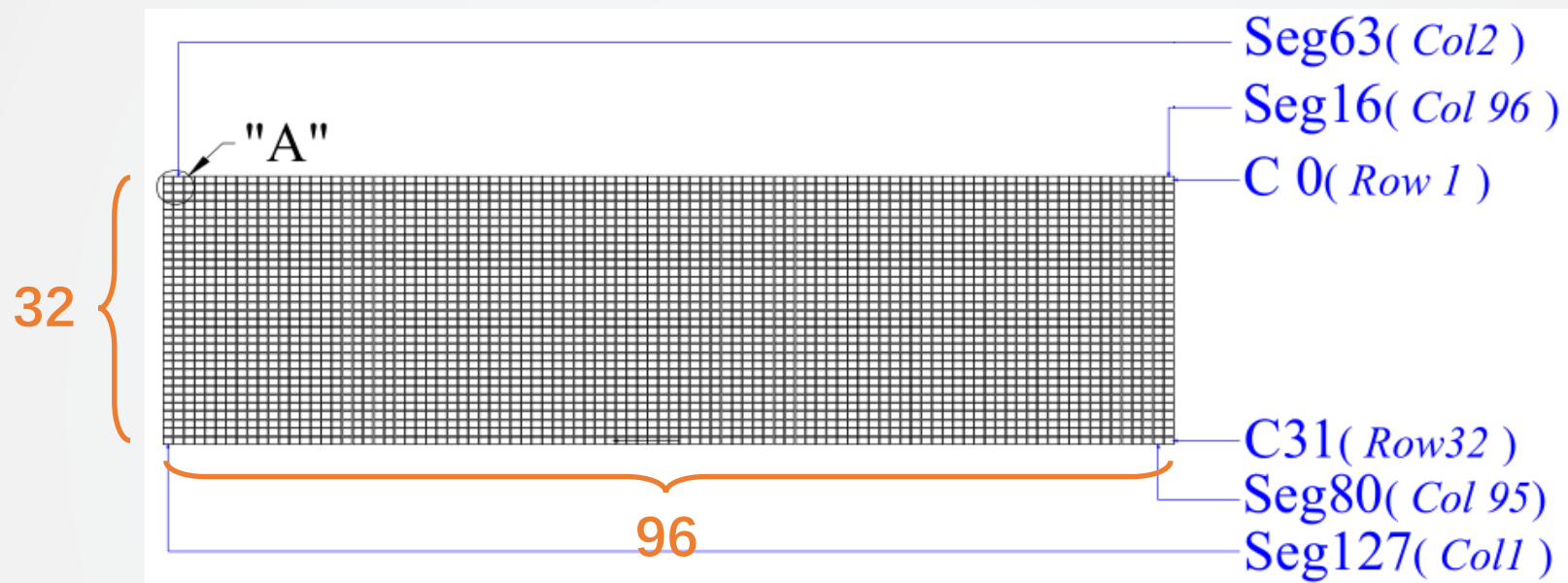
实验原理 —— Linux I2C驱动

- 为什么要分I2C总线驱动与I2C设备驱动?
- I2C总线驱动: 控制着I2C的时序, 相对固定
- I2C设备驱动: 每个具体设备的设备地址、控制命令、数据都不同
- 总线驱动决定总线和设备之间如何发数据, 设备驱动决定总线与设备之间发什么数据
- 体现了Linux内核中, 机制与策略分离的思想
- 总线驱动: drivers/i2c/busses/i2c-rk3x.c
- 核心层: drivers/i2c/i2c-core.c
- 设备驱动: 需要实现



实验原理 —— OLED设备

■ OLED (SSD1316) : 96×32像素



Page Addressing
Mode

	COL0	COL1	COL95
PAGE0	→			
PAGE1	→			
PAGE2	→			
PAGE3	→			

实验原理 —— OLED驱动

■ 按字符设备驱动

- `register()`: 注册设备, 包括获取I2C Adapter、根据从设备地址创建I2C Device
- `probe()`: 设备与驱动匹配时被调用, 可用于初始化设备
- `read()`: 从设备驱动中的缓存读取数据
- `write()`: 将数据写入设备驱动中的缓存

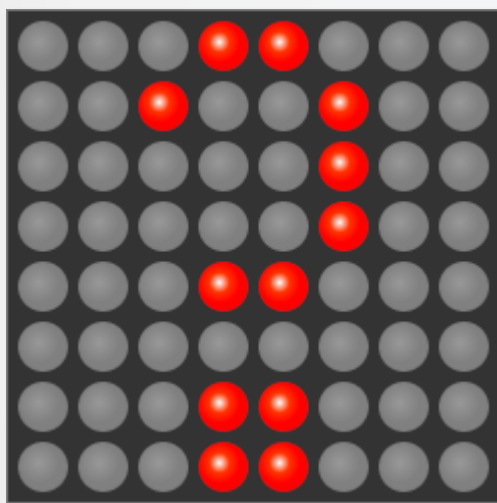
■ SSD1316芯片厂商提供了裸机驱动代码

- 阅读并分析厂商提供的驱动代码
- 根据裸机驱动代码, 编写OLED的Linux设备驱动代码
- 同时可参考 8×8 LED点阵的设备驱动代码

实验原理 —— display程序

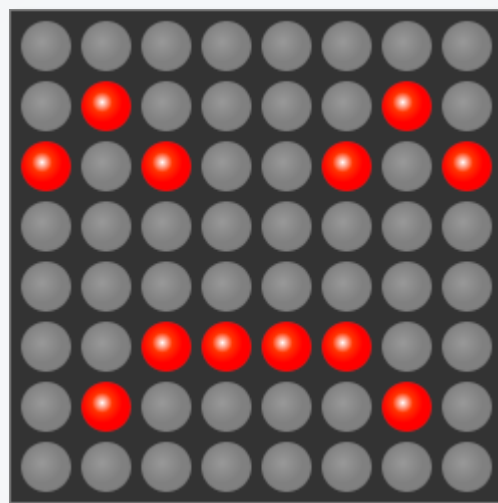
■ 根据命令行参数，在 8×8 LED点阵、OLED屏上显示图案或文字

□ `$./display <0/1/2>`



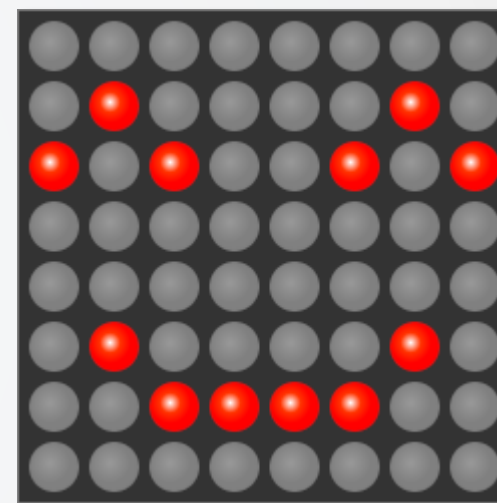
unknown

① 检测不到人脸



no mask

② 检测到没戴口罩



has mask

③ 检测到戴了口罩

PART-B: 口罩识别应用开发

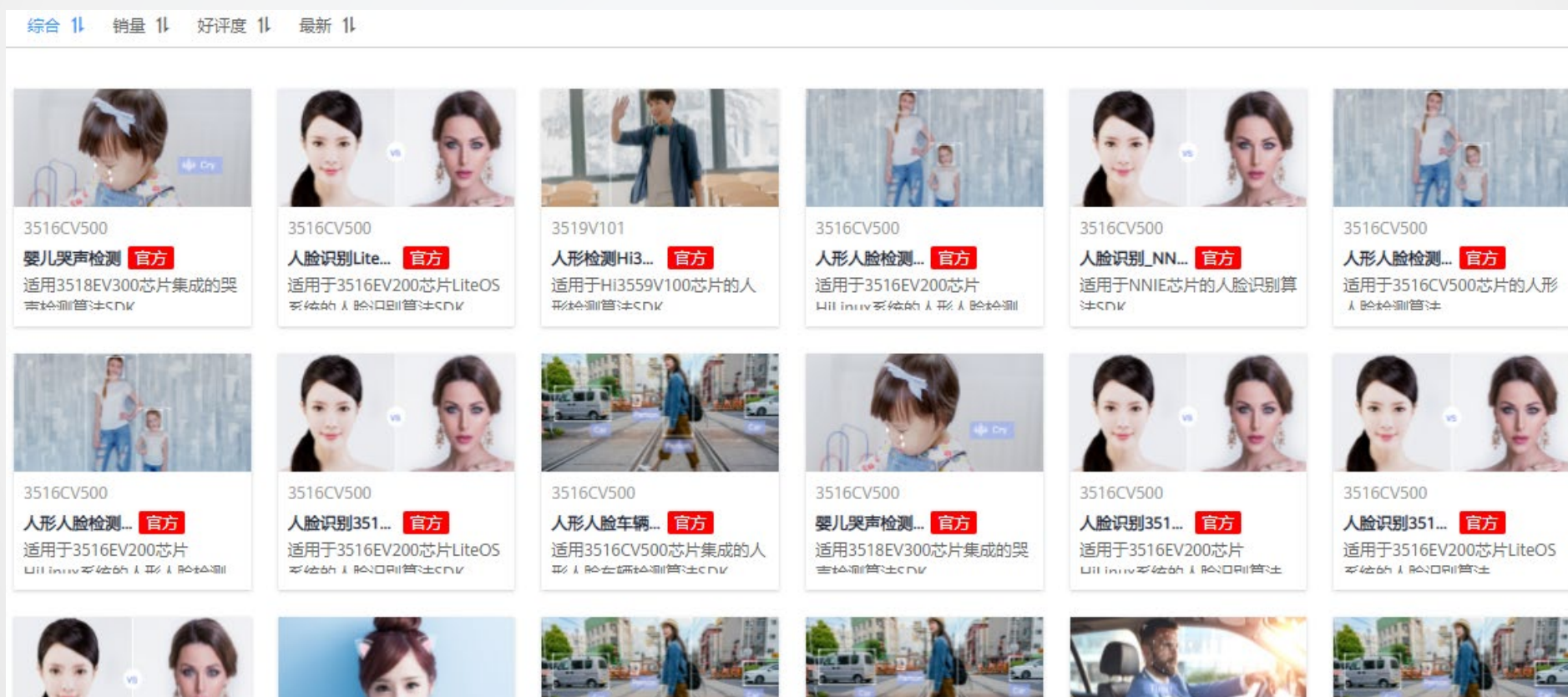
实验原理 —— HiLens

- HiLens是华为开发的端云协同的多模态AI开发应用平台，支持视觉及听觉AI应用开发、AI应用在线部署、海量设备管理等功能



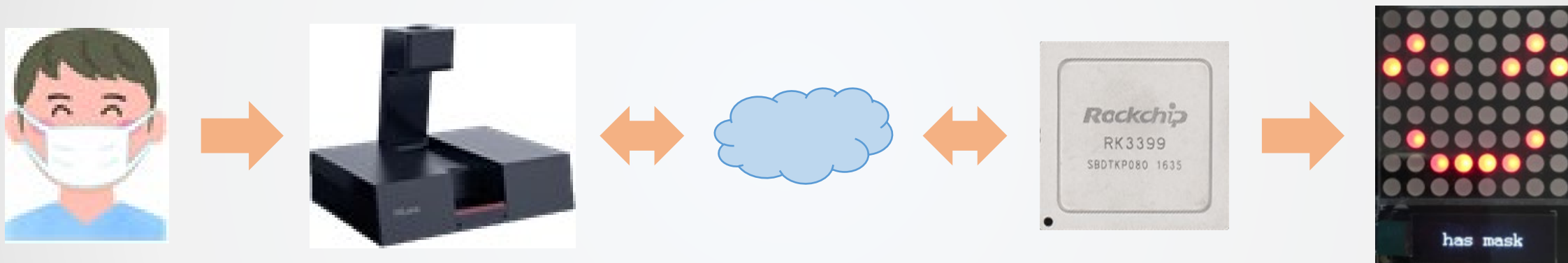
实验原理 —— 基于HiLens的口罩识别

- 华为云技能市场中，提供了许多技能模板/demo
- 开发者可在HiLens Studio中对技能进行二次开发



实验原理 —— 基于HiLens的口罩识别

- HiLens侧：口罩识别、Web/Http服务器
- GW3399侧：客户端、display应用程序



- 附加题 (+1~5)：对上述系统进行任意扩展
 - 比如：将摄像头的画面实时地显示到GW3399的屏幕、添加人脸识别和记录的功能，etc

实验步骤

- 1、获取驱动程序包（gw3399-driver），编写OLED的设备驱动程序
 - 烧写带桌面的Ubuntu系统
 - 在GW3399上加载 8×8 LED点阵驱动和OLED屏幕驱动，运行测试程序
- 2、编写display应用程序及客户端
 - 根据命令行参数，在 8×8 LED点阵、OLED屏幕上显示相应的图案或文字
 - 在GW3399上搭建TCP/Web客户端
- 3、在HiLens上开发口罩识别应用
 - 在HiLens Studio上新建口罩识别技能
 - 在HiLens Studio上搭建TCP/Web服务器
- 4、系统集成和联调

检查与提交

- 验收时，需对系统进行现场演示

- 提交内容：

```
- 学号_姓名_EmbSys实验3
  |- XXX.doc/pdf /* 0. 实验报告 */
  |- ssd1316      /* 1. OLED设备驱动 */
    |- ssd1316.c  /* OLED驱动的源文件 */
    |- Makefile   /* 修改过的Makefile */
  |- mask_app     /* 2. 修改过or自行编写的口罩识别应用相关的源文件及ELF文件 */
    |- display.c  /* 根据命令行参数，在LED点阵和OLED上显示识别结果 */
    |- display    /* 编译后的display程序的ELF文件 */
    |- Makefile   /* 修改过的Makefile */
    |- ssd1316.ko /* OLED屏的显示驱动 */
    |- mask_detection.py /* 客户端源文件(不要求必须使用Python编写) */
  |- hilens_skill /* 3. 修改过or自行编写的相关的HiLens技能源文件 */
    |- main.py
    |- utils.py
```

- 将文件夹打包成.zip，以 “学号_姓名_EmbSys实验3.zip” 命名并提交

开始实验