

产品名称 Product Name	
域格模块	
产品版本 Produce Version	Total 19 pages 共 19 页
ALL	

域格 9X07 模块 Android 使用手册

版本_V 1.40



上海域格信息技术有限公司

YUGA Technology Co., Ltd.

All rights reserved

版权所有 侵权必究



目录

1. 修改驱动 3

2. 无线接口层介绍6

3. RIL 库文件的使用8

4. 调试9

5. 注意事项 10

6. ANDROID X86 虚拟机安装（VMWARE） 11



1. 修改驱动

9X07 模块在 Android 下，仅需加载 AT 口及 modem 口，其他无关端口需过滤。以下提供两种解决方式，可根据实际情况选择处理。

1、kernel 版本支持 RSVD 的，在 option.c（路径一般为 drivers/usb/serial/option.c）中的 option_ids 中添加 RSVD，可使驱动在加载时自动跳过 RSVD 指定的 interface。

先在文件中找到 option_ids[] 的数组，添加域格模块的 VID 和 PID。VID 0x05C6 一般已存在，需调试 GPS 功能请使用 PID 0x9125，其他普通功能 PID 为 0x9025。可根据实际情况，参照以下方法添加 VID、PID：

```
#define QUALCOMM_VENDOR_ID      0x05C6
#define YUGA_PRODUCT_9X07      0x9025
#define YUGA_PRODUCT_9X07_GPS  0x9125
```

PID 9125（需调试 GPS）设置 interface 0、1、5 不加载 option 驱动，PID 9025（不调试 GPS）设置 interface 0、1、4 不加载 option 驱动，添加内容如下：

```
{ USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07),
  .driver_info = RSVD(0) | RSVD(1) | RSVD(4) },
{ USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07_GPS),
  .driver_info = RSVD(0) | RSVD(1) | RSVD(5) },
```

2、kernel 版本支持 blacklist 的，在 option.c（路径一般为 drivers/usb/serial/option.c）中的 option_ids 中添加 blacklist，可使驱动在加载时自动跳过 blacklist 指定的 interface。

先在文件中找到 option_ids[] 的数组，添加域格模块的 VID 和 PID。VID 0x05C6 一般已存在，需调试 GPS 功能请使用 PID 0x9125，其他普通功能 PID 为 0x9025。可根据实际情况，参照以下方法添加 VID、PID：

```
#define QUALCOMM_VENDOR_ID      0x05C6
#define YUGA_PRODUCT_9X07      0x9025
#define YUGA_PRODUCT_9X07_GPS  0x9125
```

PID 9125（需调试 GPS）设置 interface 0、1、5 不加载 option 驱动，PID 9025（不调试 GPS）设置 interface 0、1、4 不加载 option 驱动，添加内容如下：



```

/***** Add for YUGA 9X07 GPS *****/
static const struct option_blacklist_info YUGA_9X07_GPS_blacklist = {
    .reserved = BIT(0) | BIT(1) | BIT(5),
};
/*****

/***** Add for YUGA 9X07 *****/
static const struct option_blacklist_info YUGA_9X07_blacklist = {
    .reserved = BIT(0) | BIT(1) | BIT(4),
};
/*****

```

添加 blacklist 到 option_ids 数组中:

```

static const struct usb_device_id option_ids[] = {
    /***** Add for YUGA 9X07 GPS *****/
    { USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07_GPS),
      .driver_info = (kernel_ulong_t)&YUGA_9X07_GPS_blacklist },
    /*****

    /***** Add for YUGA 9X07 *****/
    { USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07),
      .driver_info = (kernel_ulong_t)&YUGA_9X07_blacklist },
    /*****

```

3、对于 kernel 版本不支持在 option_ids 数组中设置 blacklist 的, 要先添加模块的 VID 和 PID。VID 0x05C6 一般已存在, 需调试 GPS 功能请使用 PID 0x9125, 其他普通功能 PID 为 0x9025。可根据实际情况, 参照以下方法添加 VID、PID:

```

#define QUALCOMM_VENDOR_ID          0x05C6
#define YUGA_PRODUCT_9X07          0x9025
#define YUGA_PRODUCT_9X07_GPS      0x9125

static const struct usb_device_id option_ids[] = {
    /***** Add for YUGA 9X07 *****/
    { USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07) },
    /*****

    /***** Add for YUGA 9X07 GPS *****/
    { USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07_GPS) },
    /*****

```

在 option_probe 函数内判断当前 interface 号进行过滤:



```
/****** Add for YUGA 9X07 *****/
if(le16_to_cpu(serial->dev->descriptor.idVendor) == QUALCOMM_VENDOR_ID &&
    le16_to_cpu(serial->dev->descriptor.idProduct) == YUGA_PRODUCT_9X07 &&
    (serial->interface->cur_altsetting->desc.bInterfaceNumber <= 1 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber >= 4)) {
    printk(KERN_INFO"Discover YUGA 9X07\n");
    return -ENODEV;
}
/******

/****** Add for YUGA 9X07 GPS *****/
if(le16_to_cpu(serial->dev->descriptor.idVendor) == QUALCOMM_VENDOR_ID &&
    le16_to_cpu(serial->dev->descriptor.idProduct) == YUGA_PRODUCT_9X07_GPS &&
    (serial->interface->cur_altsetting->desc.bInterfaceNumber <= 1 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber >= 5)) {
    printk(KERN_INFO"Discover YUGA 9X07 GPS\n");
    return -ENODEV;
}
/******
```

编译完成后，通过查询命令：ls /dev/ttyU*, PID 9025（不调试 GPS）应有 ttyUSB0、ttyUSB1 两个端口，PID 9125（需调试 GPS）应有 ttyUSB0、ttyUSB1、ttyUSB2 三个端口。

ppp 配置：

4G 模块需要使用 ppp 协议进行拨号，内核配置中必须把 ppp 相关协议打开。

以 kernel2.6.7 内核为例 menuconfig 的路径如下

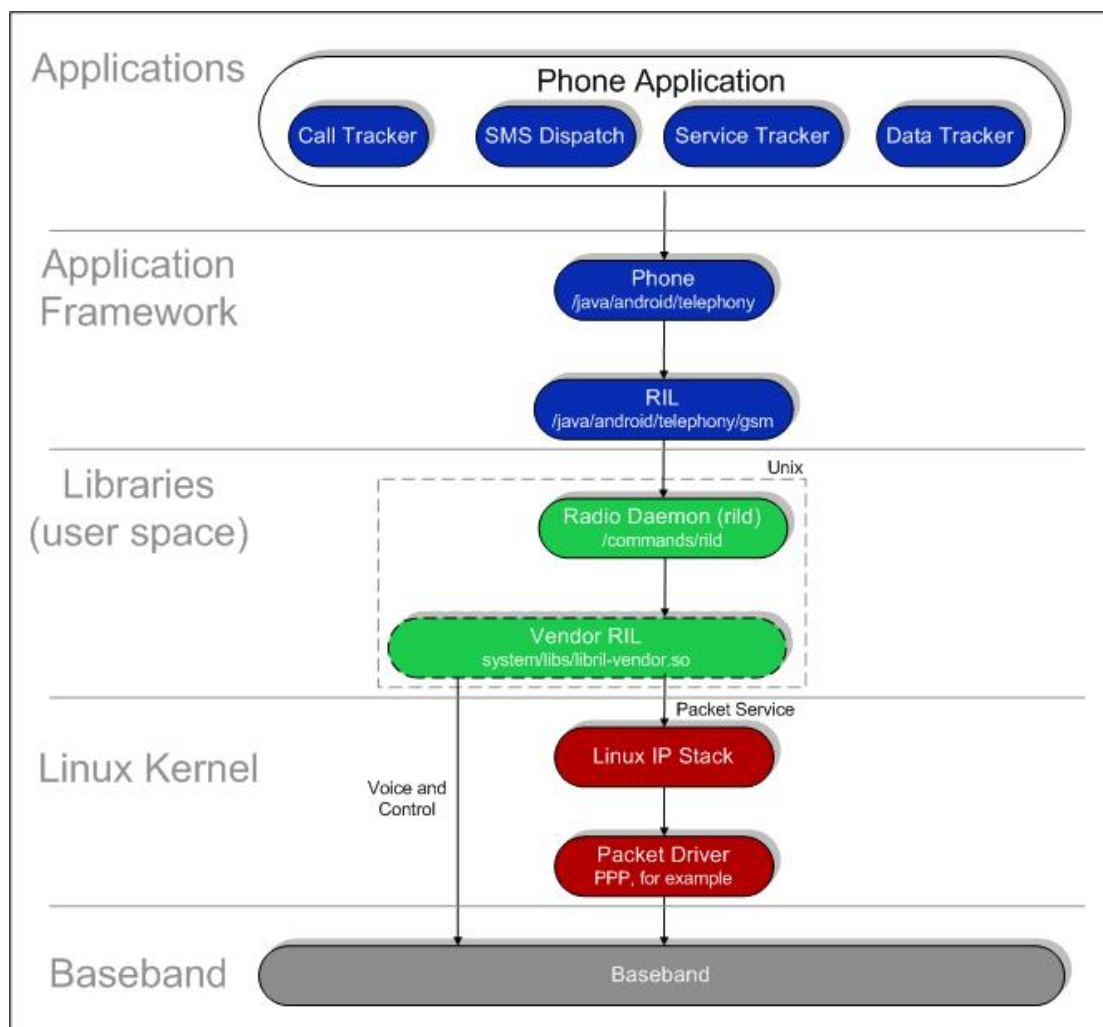
Device Drivers->Network device support->PPP 相关选项全部勾选。



2. 无线接口层介绍

当数据卡驱动被内核加载并正确驱动后，该 4G 模块才能进行无线通信功能的应用开发。

这在 Android 系统下称为无线接口层——RIL。即该文档所要说明的 RIL 驱动。



android 的 ril 位于应用程序框架与内核之间，分成了两个部分，一个部分是 rild,它负责 socket 与应用程序框架进行通信。另外一个部分是 Vendor RIL，这个部分负责向下是通过两种方式与 radio 进行通信，它们是直接与 radio 通信的 AT 指令通道和用于传输数据包的通道，数据通道用于手机的上网功能。也就是我们所提供的 RIL 驱动，是实现通信业务的核心功能模块。对于 RIL 的 java 框架部分，也被分成了两个部分，一个是 RIL 模块，这个模块主要用于与下层的 rild 进行通信，另外一个 Phone 模块，这个模块直接暴露电话功能接口给应用开发用户，供他们调用以进行电话功能的实现。这是属于 Android 应用程序的开发部分，简单了解即可。所以 RIL 驱动模块，必需是针对不同的 4G 模块指令和通



信业务功能定制的，以支不同应用需求。建议开发人员在使用我们的 RIL 驱动前，先了解下“4G 模块的特点和构造”。



3. ril 库文件的使用

将我们提供的 ril 库文件，修改成系统使用的库文件名后，push 到相应文件夹下。如将 libreference-ril.so 放到 /system/lib: adb push libreference-ril.so /system/lib



4. 调试

使用 `adb shell logcat -b radio -v time` 来获取 ril 日志。Logcat 来获取详细的日志信息。
调试如遇到问题可以把日志反馈给我们。



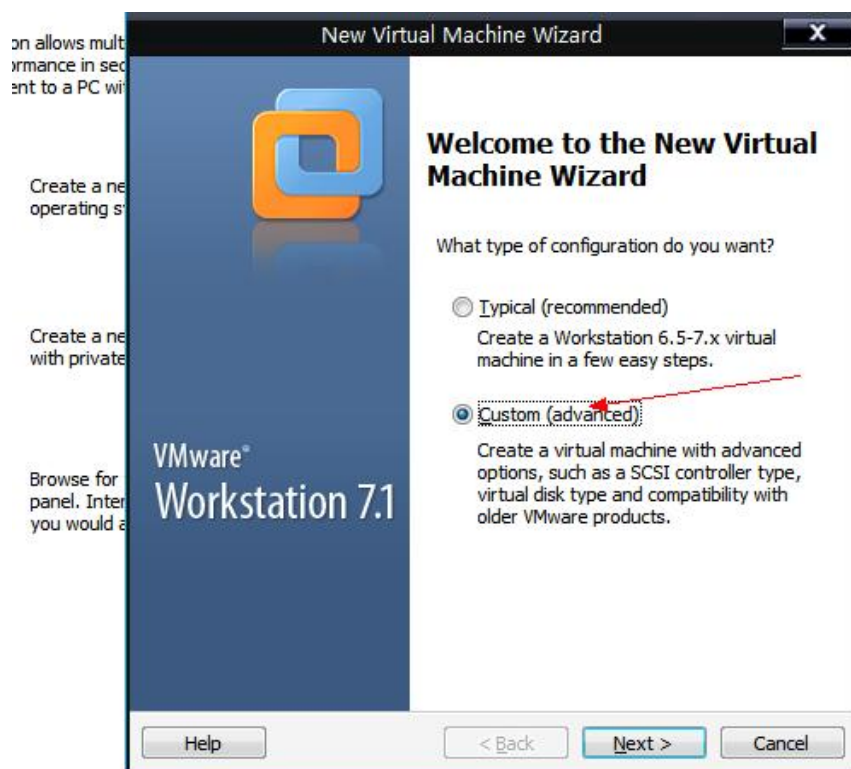
5. 注意事项

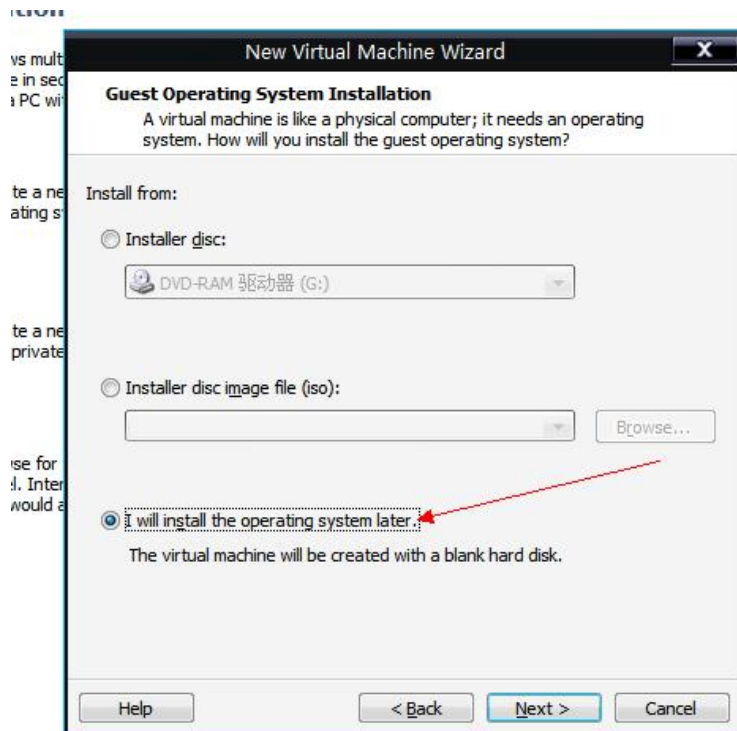
要使用模块网络功能，务必将 **wifi** 和本地网络关闭，否则系统会把无线数据断开。



6. Android X86 虚拟机安装（VMWare）

新建虚拟机，注意红色标识：





ware Workstation

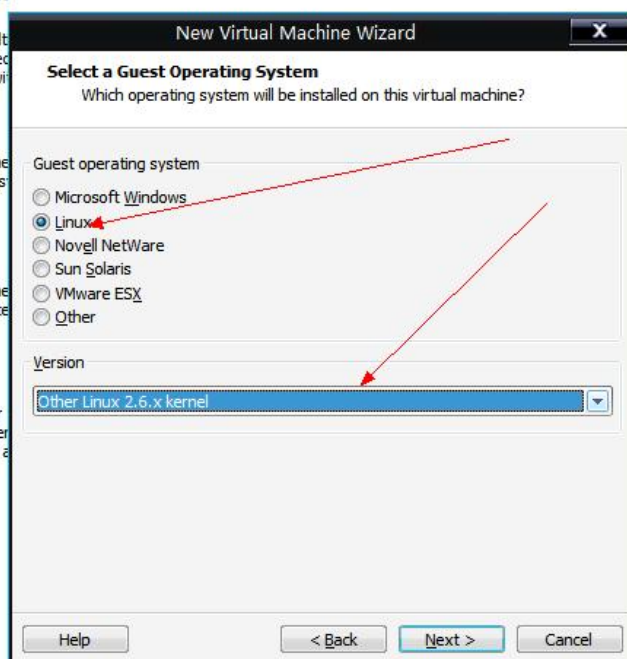
ware Workstation allows multiple virtual machines to run on a single host with high performance in seconds. A virtual machine is equivalent to a PC with its own hardware choices.

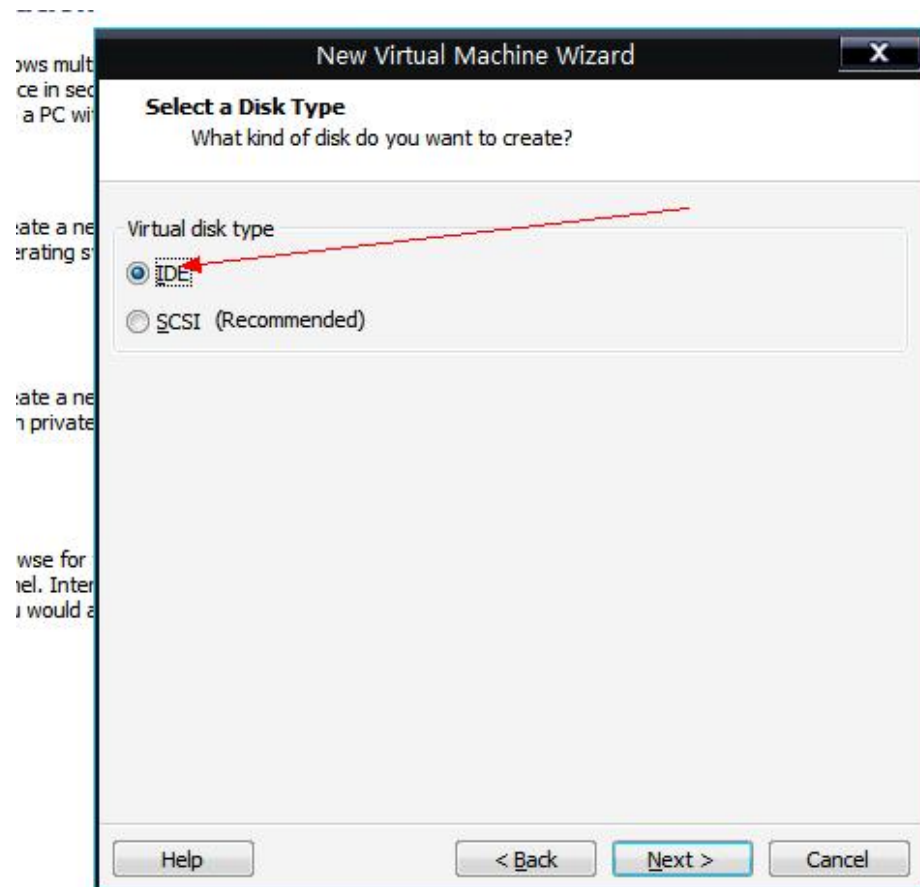


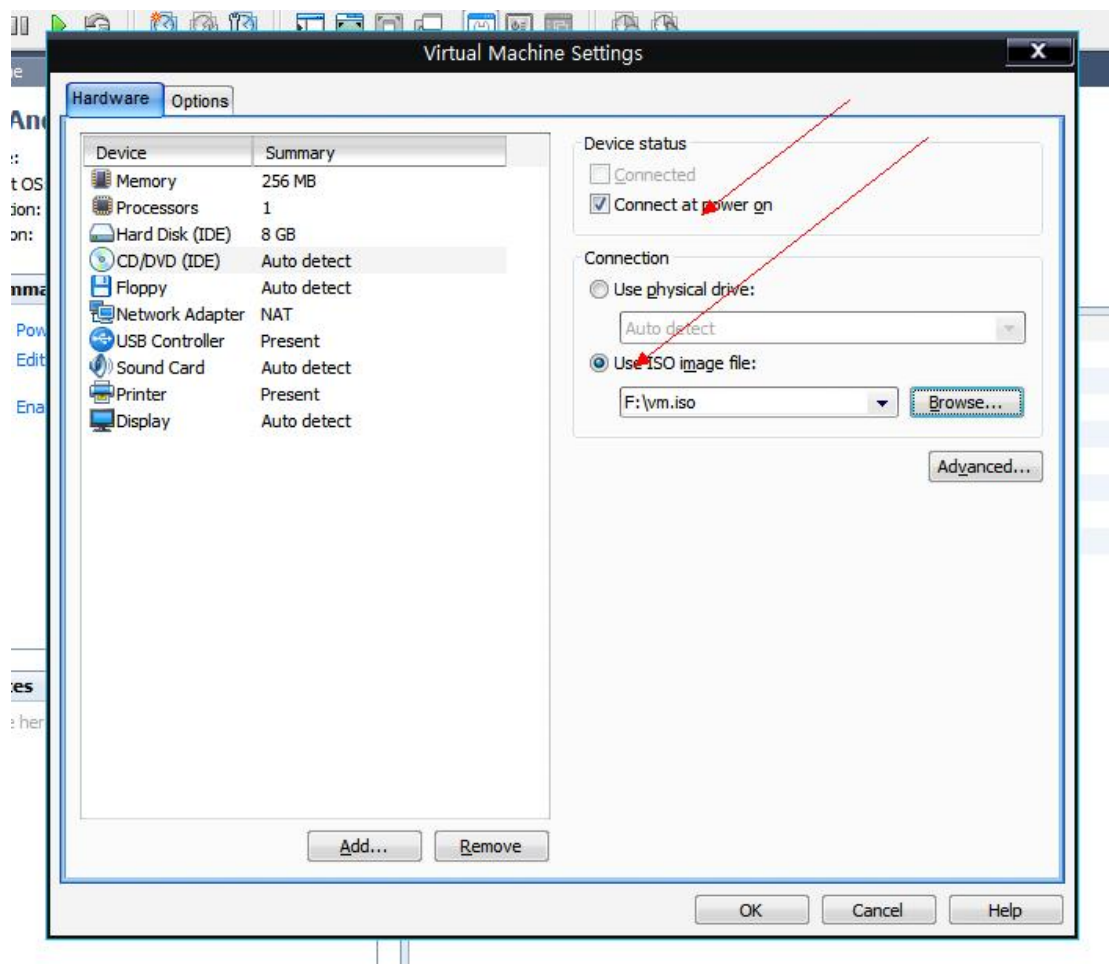
Create a new virtual machine with a new operating system.

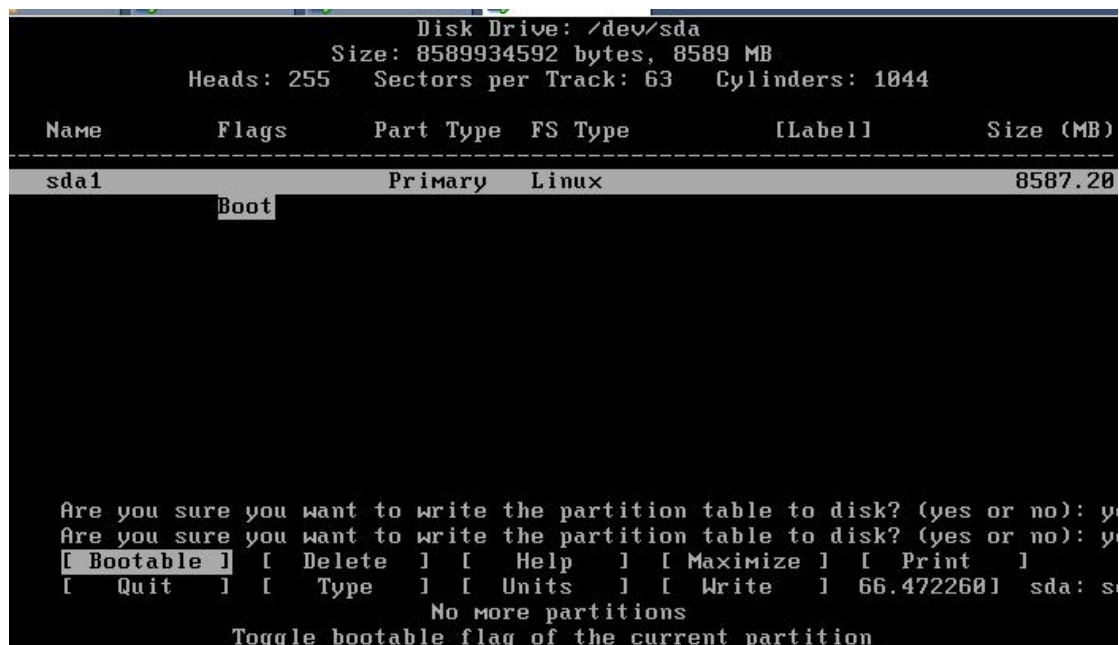
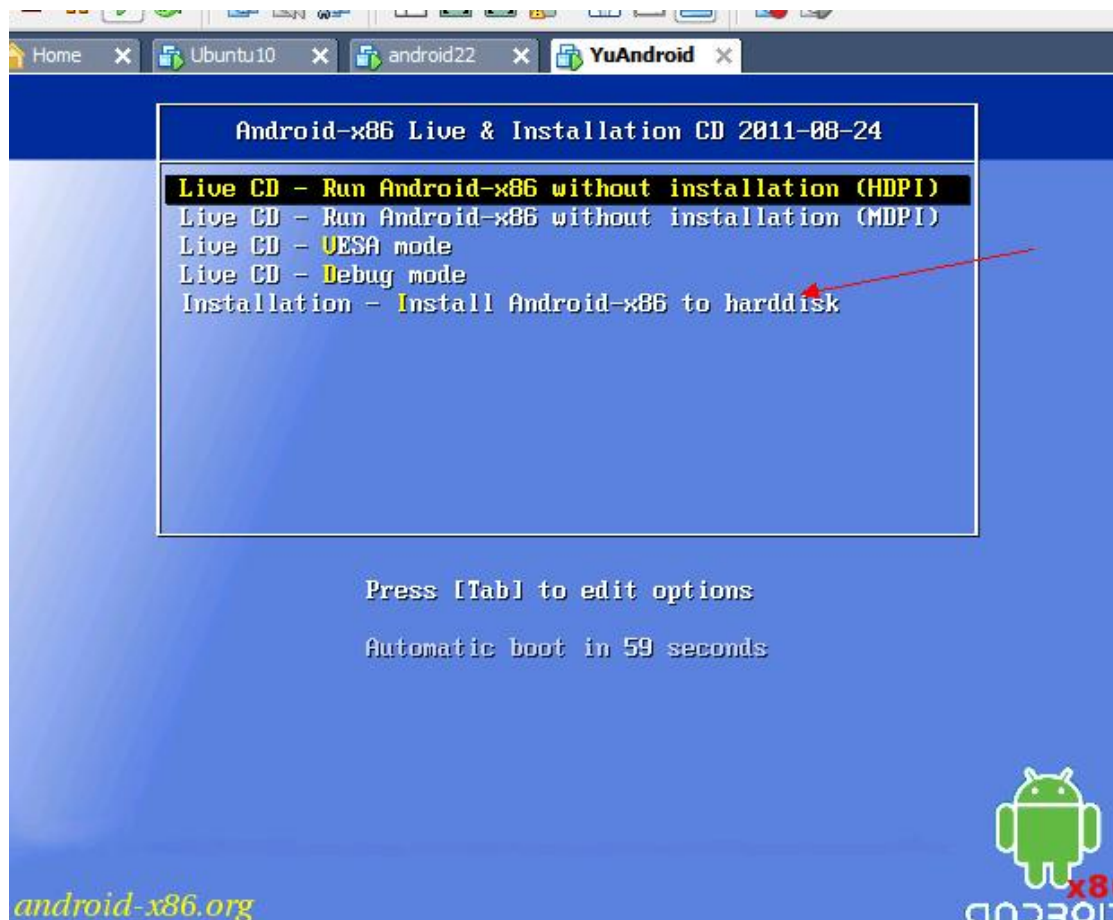
Create a new virtual machine with a private network.

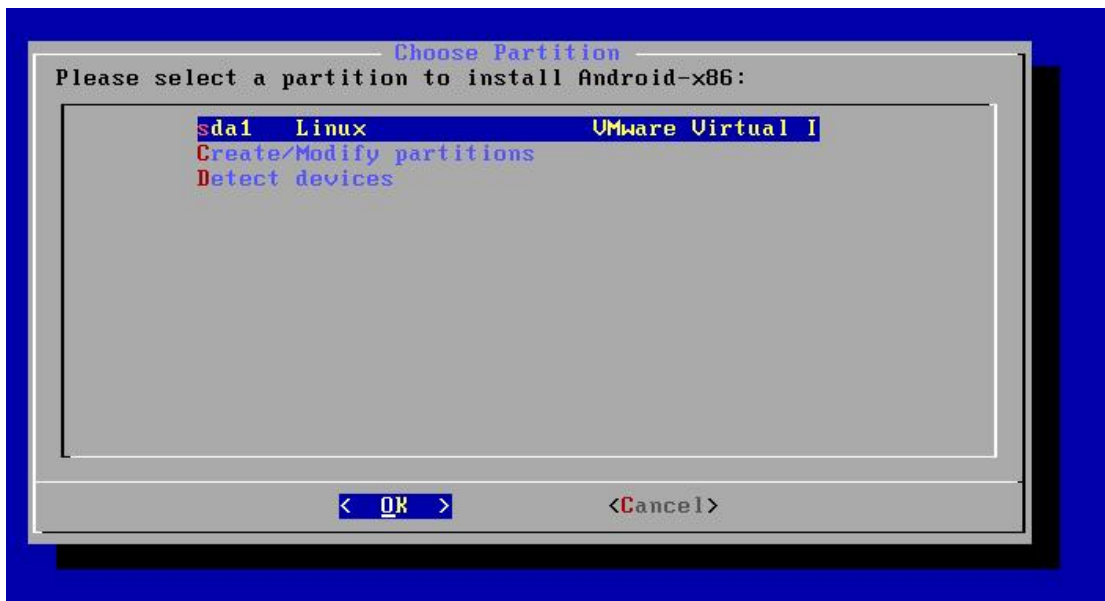
Browse for an existing virtual machine. If you would like to open an existing virtual machine, you would select this option.

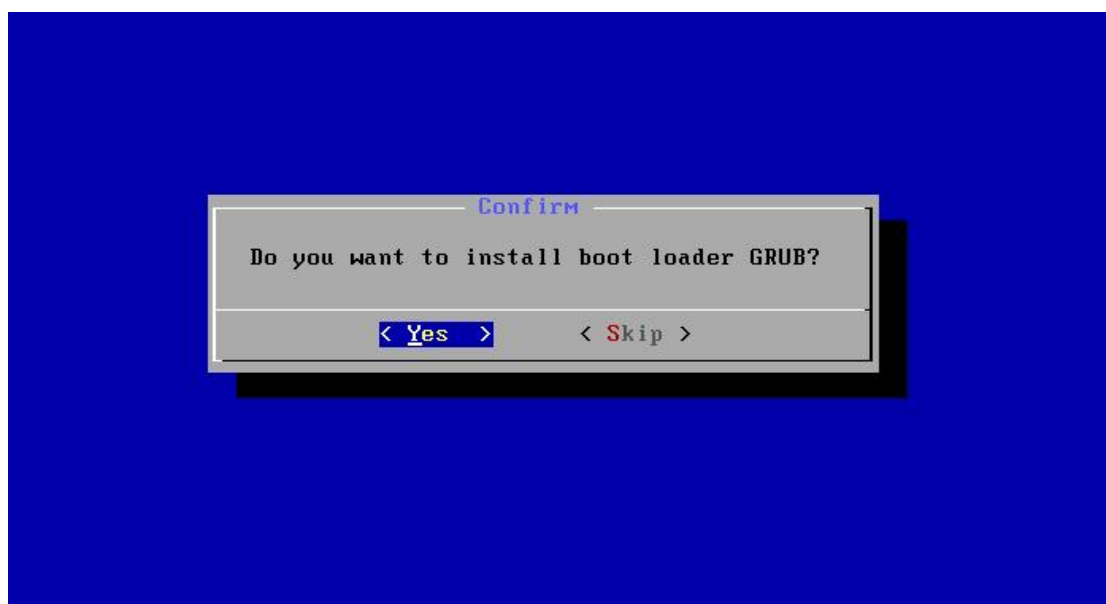
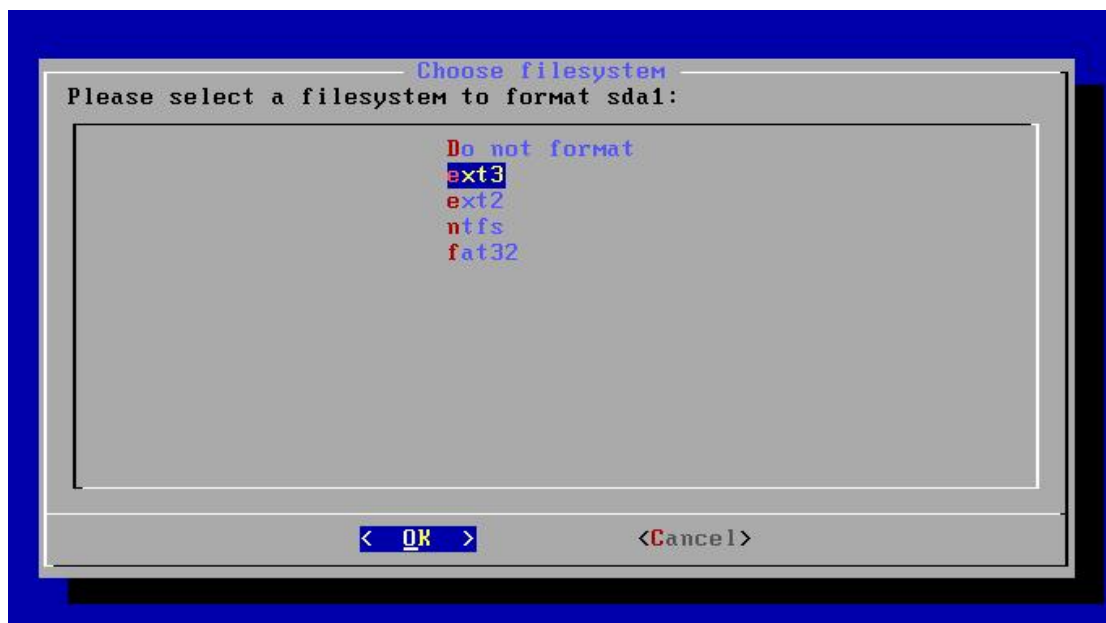


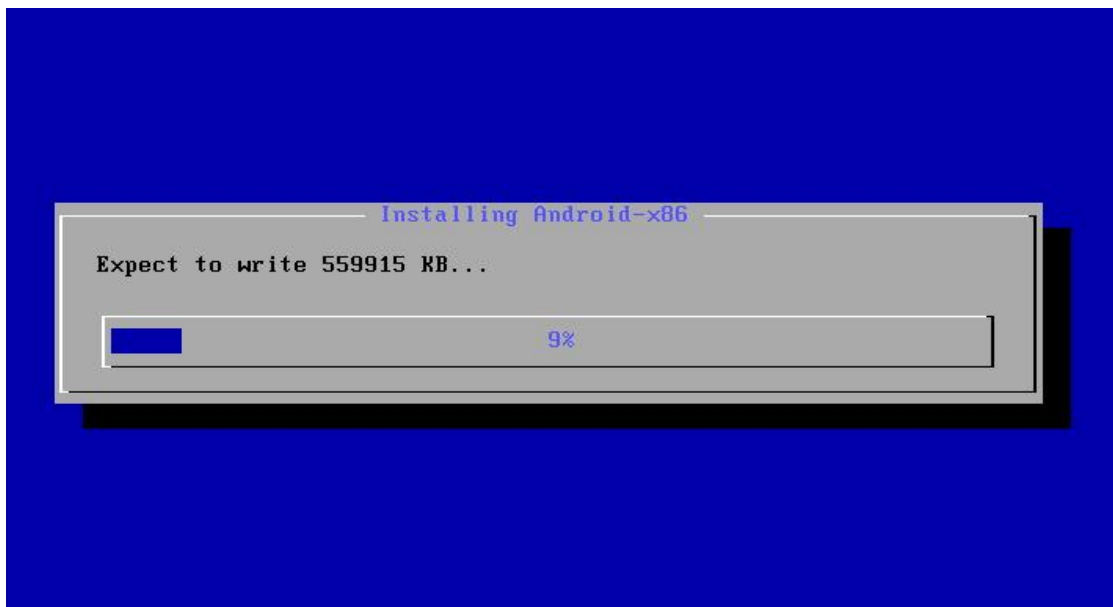
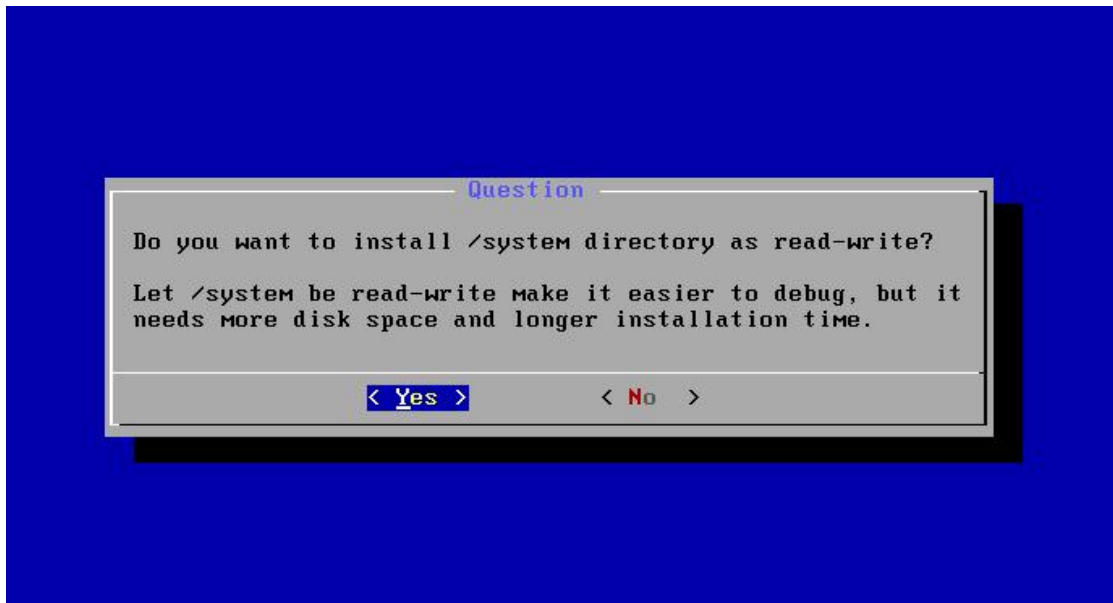


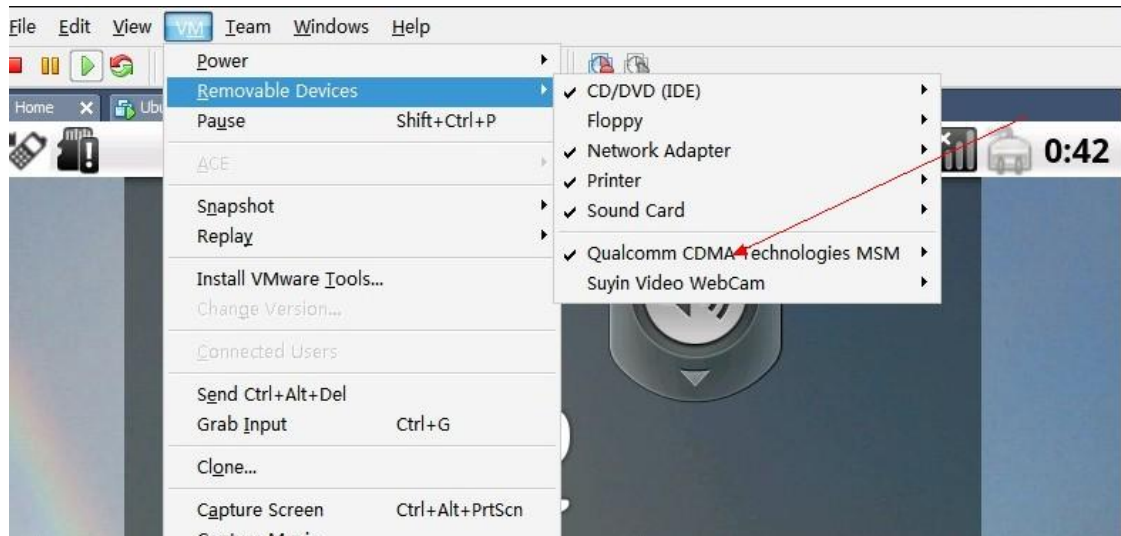












最后将设备挂入虚拟机，支持完成。虚拟机里基本可以模拟模块的功能需求。