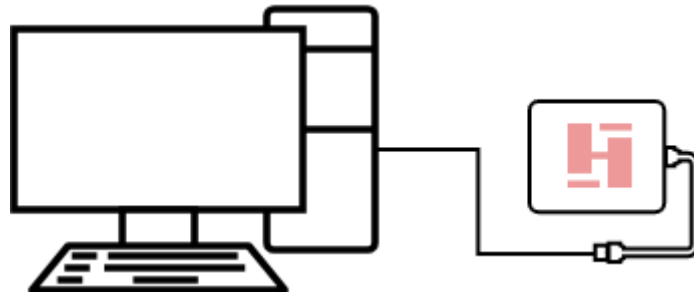


HM-CORE

立即开箱测试

1通过USB连接线将HM-CORE连接到你的计算机



HM-CORE默认内置固件可以点亮呼吸灯，并启动Linux操作系统。

2 使用串口连接到HM-CORE的系统控制台

```
U-Boot 2016.07 (Mar 14 2018 - 23:45:09 +0800)

DRAM: ECC disabled 64 MiB
MMC: sdhci@e0101000: 0
SF: Detected S25FL128S_64K with page size 256 Bytes, erase size 64 KiB, total 16 MiB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
U-BOOT for

Hit any key to stop autoboot: 0

...

EXT4-fs (mmcblk0p2): recovery complete
EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
VFS: Mounted root (ext4 filesystem) on device 179:2.
devtmpfs: mounted
Freeing unused kernel memory: 1024K (c0800000 - c0900000)

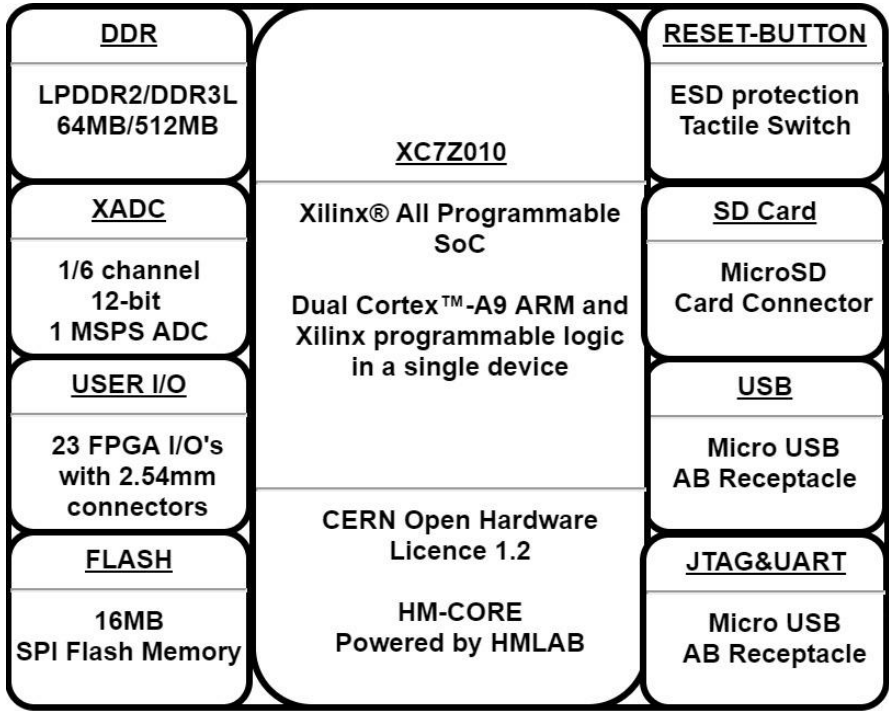
Ubuntu 12.10 hmcore ttyPS0

hmcore login:
```

打开串口软件(Windows系统首次使用需要安装[驱动程序](#))，确保串口设置（波特率115200，数据位8，停止位1，校验无，流控无）正确，若看不到启动打印信息，可以尝试发送回车或重新启动板卡。成功连接后输入用户名: root 密码: root，即可登录系统。

了解你拥有的资源

HM-CORE开源SOC硬件平台采用Arduino外形结构，板载Zynq-7010芯片、64/512MB DDR内存芯片、16MB固态FLASH配置芯片、MicroSD卡插槽、MicroUSB OTG接口、板载JTAG调试器和UART接口、微型复位键、LED指示灯和通过2.54mm接插件引出的23个用户I/O引脚。



ZYNQ SOC

Zynq AP SoC(型号：**xc7z010clg225-1**)芯片主要由两部分组成：Processing System (PS)和Programmable Logic (PL)。

PS	Processor Core: Dual-Core ARM Cortex-A9 MPCore with CoreSight L1 Cache: 32 KB Instruction, 32 KB Data per processor L2 Cache: 512 KB On-Chip Memory: 256 KB External Memory Support: DDR3, DDR3L, DDR2, LPDDR2 External Memory Support: 2x Quad-SPI, NAND, NOR DMA Channels: 8 (4 dedicated to Programmable logic) Peripherals: 2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO
PL	7 Series PL Equivalent: Artix-7 FPGA Logic Cells: 28K Look-Up Tables (LUTs): 17,600 Flip-Flops: 35,200

	Total Block RAM(# 36Kb Blocks): 2.1Mb(60) DSP Slices: 80 Analog Mixed Signal (AMS) / XADC(2) 2x 12 bit, MSPS ADCs with up to 17 Differential Inputs
--	---

PS DDR

HM-CORE-A PS端的DDR(型号：**MT42L32M16D1AB-25 IT:A**)采用Micron® 64MB Automotive Mobile LPDDR2 SDRAM，用于存储掉电可丢失的数据，通常作为内存使用。

Memory Type	LPDDR2
Effective DRAM IC Bus Width	16 Bit
ECC	Disabled
Burst Length	8
Memory Clock Frequency	400
DRAM Device Capacity	512Mbits
Speed Bin	LPDDR2 800
Bank Address Count(Bits)	2
Row Address Count(Bits)	13
Col Address Count(Bits)	10
CAS Latency(cycles)	6
CAS Write Latency(cycles)	NA
RAS to CAS Delay(cycles)	8
Precharge Time(cycles)	7
tRC(ns)	48.91
tTASmin(ns)	34.0
tFAW(ns)	25

HM-CORE-B PS端的DDR(型号：MT41J256M16 RE-125)采用Micron® 512MB low-voltage version of the 1.5V DDR3 SDRAM device，用于存储掉电可丢失的数据，通常作为内存使用。

Memory Type	DDR 3(Low Voltage)
Effective DRAM IC Bus Width	16 Bit
ECC	Disabled
Burst Length	8

Memory Clock Frequency	533.333333
DRAM Device Capacity	4096MBits
Speed Bin	DDR3_1066F
Bank Address Count(Bits)	3
Row Address Count(Bits)	15
Col Address Count(Bits)	10
CAS Latency(cycles)	7
CAS Write Latency(cycles)	6
RAS to CAS Delay(cycles)	7
Precharge Time(cycles)	7
tRC(ns)	48.91
tTASmin(ns)	35.0
tFAW(ns)	40.0

USB

USB接口PHY芯片(型号：**USB3320C-EZK**)是带有ULPI接口的高度集成的全功能高速USB 2.0收发器，用于连接USB移动设备，如U盘、无线网卡等。

Version	USB 2.0
Support	OTG, high-speed, full-speed, and low-speed
Interface	Micro-B

UART & JTAG

UART接口采用USB转UART&JTAG接口芯片(型号：**FT2232H**)将UART接口和JTAG接口集成在一个MicroUSB接口上，用于系统调试和监控(使用这个二合一接口需要安装相应驱动程序)。

Default baud rate	115200(up to 1 Mb/s)
Driver	FTDI Virtual COM Port (VCP)

Standard	IEEE Std 1149.1 Compatible Test Interface
Mode	Cascaded JTAG for PS & PL

RESET BUTTON

复位按键(型号：**KMS221GLFS**)采用一款微型SMT侧面开关，连接到了ZYNQ SOC的POR_B管脚，可用于系统复位。

Polarity	Active Low
Connect pin	POR_B

MicroSD SOCKET

MicroSD卡插槽(型号：**104031-0811**)自带插入检测功能，连接到ZYNQ PS的SD卡控制器上，用于存储设备树、内核文件、文件系统和用户数据等。

Interface	MicroSD
Style	Push-Pull
Height	1.42mm

QSPI FLASH

FLASH(型号：**S25FL127S**)为QSPI接口，容量为16MB，用于存储系统启动代码(BOOT.bin)等数据。

Interface	SPI with Multi-I/O
Density	128 Mbits (16 Mbytes)
Cycling Endurance	100,000 Program-Erase Cycles per sector, minimum
Data Retention	20 Year Data Retention, minimum

XADC

板载6选1模拟信号通道选择芯片(型号：**CD74HC4051D**)和电压转换芯片(型号：**AD8031ART**)，将采集信号接入XADC，用于简单的模数信号采集转换。

ADC	12-bit
Sample rates	1 MSPS

Bandwidths	500KHz
Voltage range	0~3.3V

I/O

共有23个用户I/O和6个电源引脚通过**2.54mm连接器**引出，可用于连接扩展板卡或主控制线路板(如图像、语音采集传感器，无人机飞控平台等)。

External reset	1
PL user I/O	12
Analog input	6
Interface	IIC-2, UART-2
Power supply	3.3V-2, 5V-1, GND-3

BOOT

HM-CORE通过**固定**的电阻连接进行BOOT启动模式控制，默认从QSPI Flash启动，启动前需要将你的BOOT.bin文件通过Xilinx SDK工具烧写进QSPI Flash。

FSBL(BOOT.bin)	QSPI Flash
PL(BOOT.bin)	QSPI Flash
U-Boot(BOOT.bin)	QSPI Flash
Kernel(image.ub)	SD Card(boot)
Device Tree(system.dtb)	SD Card(boot)
Root Filesystem	SD Card(rootfs)

准备进行开发

1 从官网下载**board files**文件，拷贝到你的Vivado安装路径下：

Example:

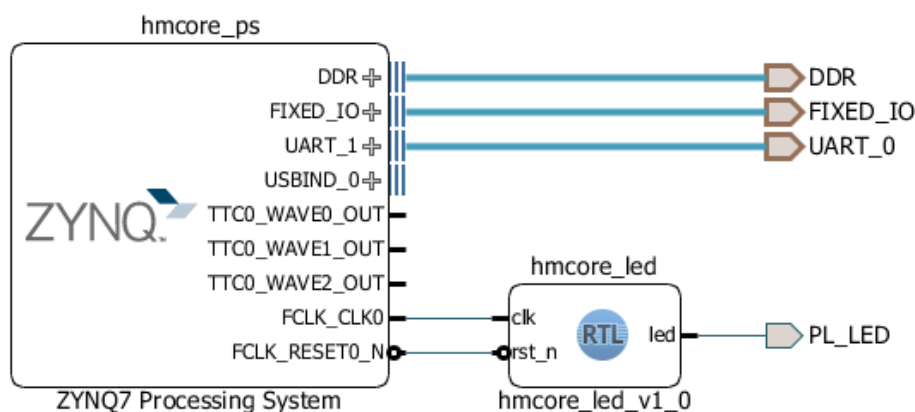
Windows: C:\Xilinx\Vivado\2016.4\data\boards\board_files

Linux: /home/Xilinx/Vivado/2016.4/data/boards/board_files

2 从官网下载配套**BIST**测试工程及系统相关文件(**U-BOOT**、**设备树**、**内核**、**文件系统**)，用于检查资源及组件是否正常。

BIST工程

将BIST(Build-In Self Test)工程文件解压后，Block design中由两个IP组成：ZYNQ和LED，ZYNQ IP用于运行裸机程序或嵌入式Linux操作系统；LED IP通过PL引脚连接呼吸灯。



了解BIST工程

ZYNQ IP配置了运行内存DDR、系统控制台交互UART、USB 2.0接口。

LED IP用于驱动呼吸灯，双机该IP可修改呼吸频率，其时钟和复位引脚来自FCLK0(100MHz)。

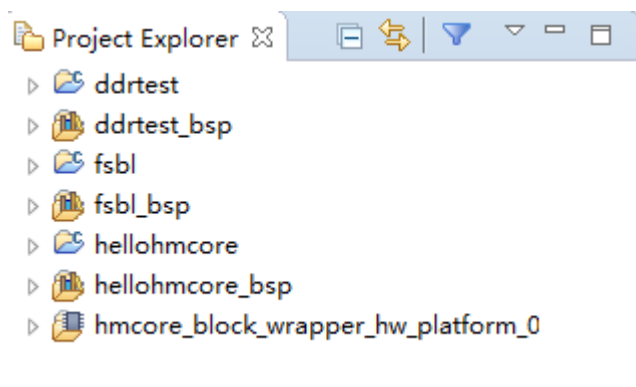
从BIST工程中提取ZYNQ配置文件用于新的设计工程

1 打开BIST工程，进入Block Design，双击打开ZYNQ IP，Presets > Save Configuration > Preset Name: hmcore-zynq & File name: hmcore-zynq.tcl点击确定，导出配置文件。

2 新建设计工程，新建Block Design，新建ZYNQ IP，从菜单栏Presets > Apply Configuration > 选择hmcore-zynq.tcl，导入配置文件。

编写SDK程序

从BIST工程菜单: Files > Launch SDK，打开Xilinx SDK开发环境，可以看到如下测试DEMO:



Hello HM-CORE

自带测试DEMO

通过USB连接线缆连接HM-CORE，右键单击hellohmcore，选择Run As > Launch on Hardware (System Debugger)，观察串口控制台打印信息。

新建测试DEMO

通过USB连接线缆连接HM-CORE，打开Xilinx SDK开发软件，通过点击File > New > Application Project，输入工程名称并选择Hello World模板，右键单击新建的工程，选择Run As > Launch on Hardware(System Debugger)，观察串口控制台打印信息。


```
Connecting to COM11...
Connected.
```

```
Hello hmc core!
Hello hmc core!
Hello hmc core!
Hello hmc core!
Hello hmc core!
```

测试DDR

自带测试DEMO

通过USB连接线连接HM-CORE，右键单击ddrtest，选择Run As > Launch on Hardware(System Debugger)，观察串口控制台打印信息。

新建测试DEMO

通过USB连接线连接HM-CORE，打开Xilinx SDK开发软件，通过点击File > New > Application Project，输入工程名称并选择Memory Tests模板，右键单击新建的工程，选择Run As > Launch on Hardware(System Debugger)，观察串口控制台打印信息。

```
Connecting to COM11...
Connected.
```

```
--Starting Memory Test Application--
```

```
NOTE: This application runs with D-Cache disabled.As a result, cacheline requests will not be generated
```

```
Testing memory region: ps7_dds_0
  Memory Controller: ps7_dds_0
    Base Address: 0x1000000
      Size: 0x3F00000 bytes
        32-bit test: PASSED!
        16-bit test: PASSED!
        8-bit test: PASSED!
```

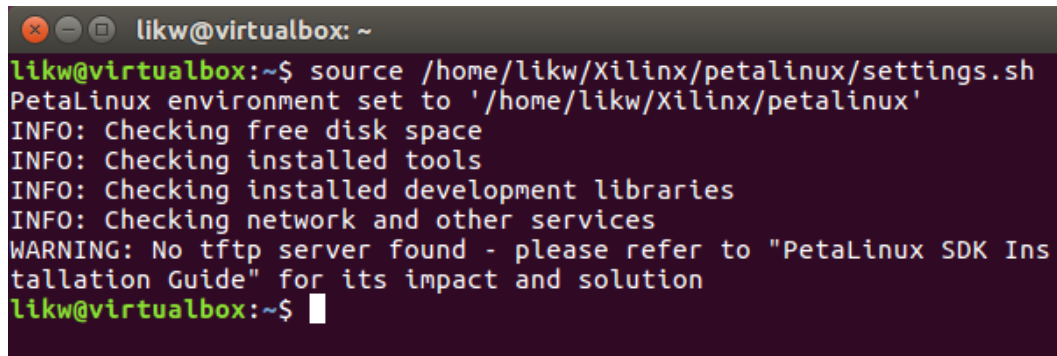
```
Testing memory region: ps7_ram_1
  Memory Controller: ps7_ram_1
    Base Address: 0xFFFF0000
      Size: 0xFE00 bytes
        32-bit test: PASSED!
        16-bit test: PASSED!
        8-bit test: PASSED!
```

```
--Memory Test Application Complete--
```

开始运行Linux系统

搭建Petalinux开发环境

下载Xilinx PetaLinux安装包，按照UG1144中的步骤进行安装。最后通过命令"*source /home/likw/Xilinx/petalinux/settings.sh*"验证安装完成。

A terminal window titled 'likw@virtualbox: ~' showing the execution of 'source /home/likw/Xilinx/petalinux/settings.sh'. The output indicates the PetaLinux environment is set to '/home/likw/Xilinx/petalinux' and lists several informational messages: 'INFO: Checking free disk space', 'INFO: Checking installed tools', 'INFO: Checking installed development libraries', and 'INFO: Checking network and other services'. A warning message follows: 'WARNING: No tftp server found - please refer to "PetaLinux SDK Installation Guide" for its impact and solution'. The prompt returns to 'likw@virtualbox:~\$'.

下载HM-CORE PetaLinux工程

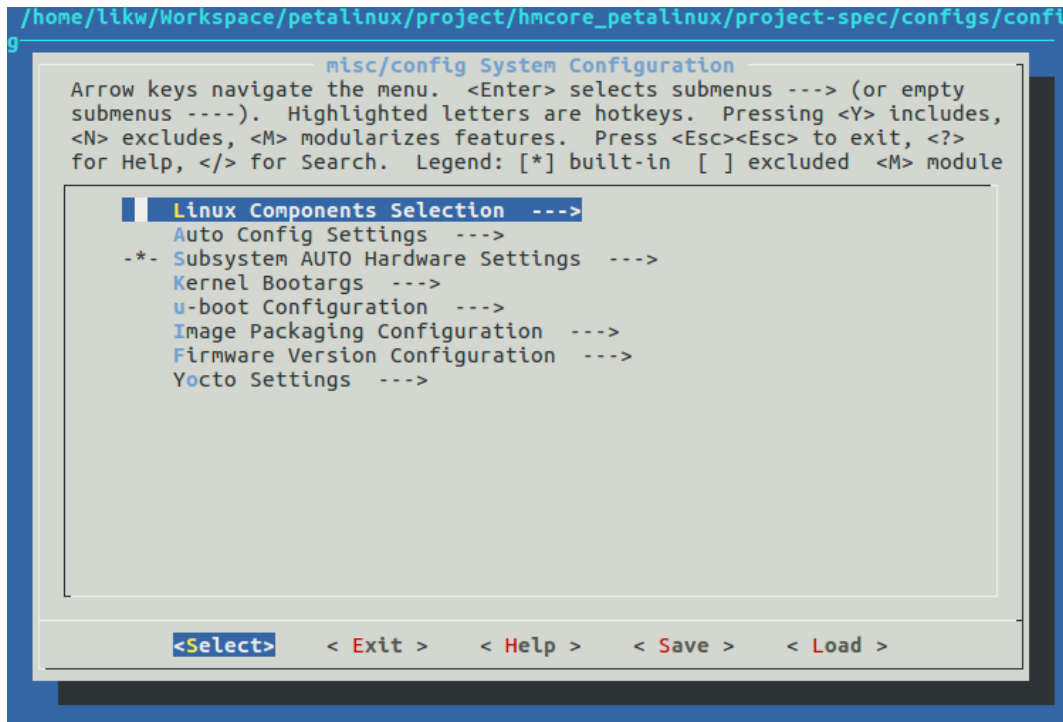
从官网下载PetaLinux工程文件，解压到你的PetaLinux工作目录：

通过命令"*petalinux-config*"进行基本配置；

通过命令"*petalinux-config -c kernel*"进行内核配置；

通过命令"*petalinux-bulid*"进行工程编译；

通过命令"*petalinux-package --boot --format BIN --fsbl ./images/linux/zynq_fsbl.elf --fpga ./images/linux/hmcore_block_wrapper.bit --u-boot --force*"生成BOOT.bin。



新建Petalinux工程

从BIST或你的Vivado设计工程hmcore_bist.sdk文件夹中复制hmcore_block_wrapper.hdf文件到PetaLinux工作目录：

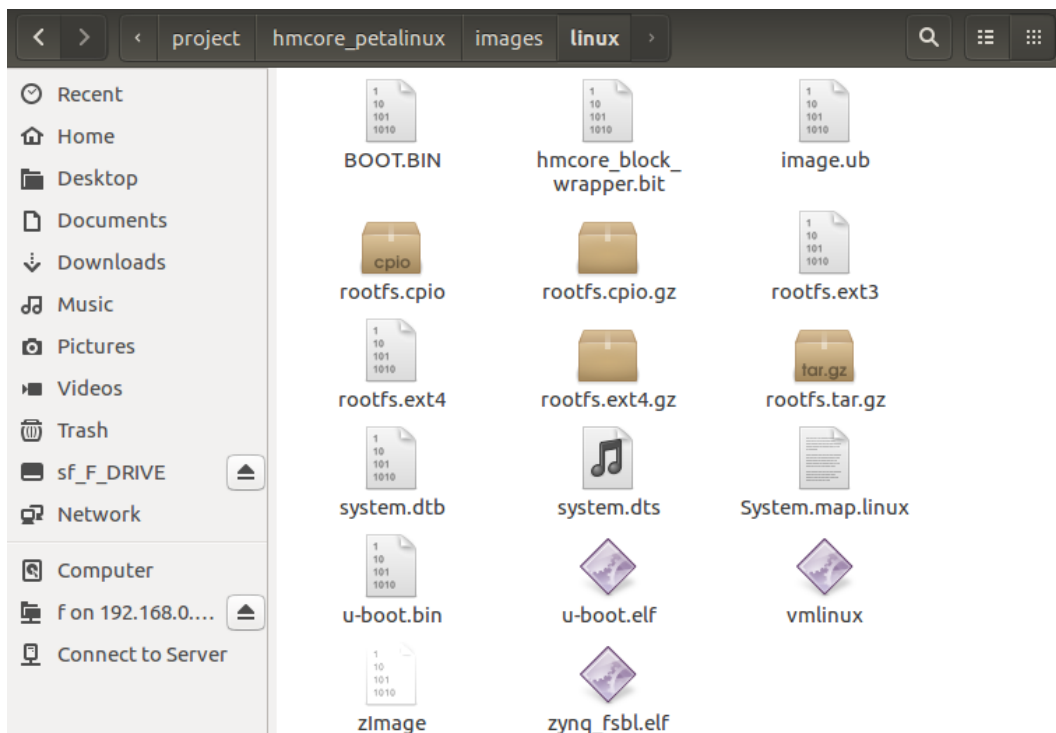
通过命令"`petalinux-config --get-hw-description=/home/likw/Workspace/petalinux/project/hmcore_petalinux`"新建工程；

通过命令"`petalinux-config`"进行基本配置；

通过命令"`petalinux-config -c kernel`"进行内核配置；

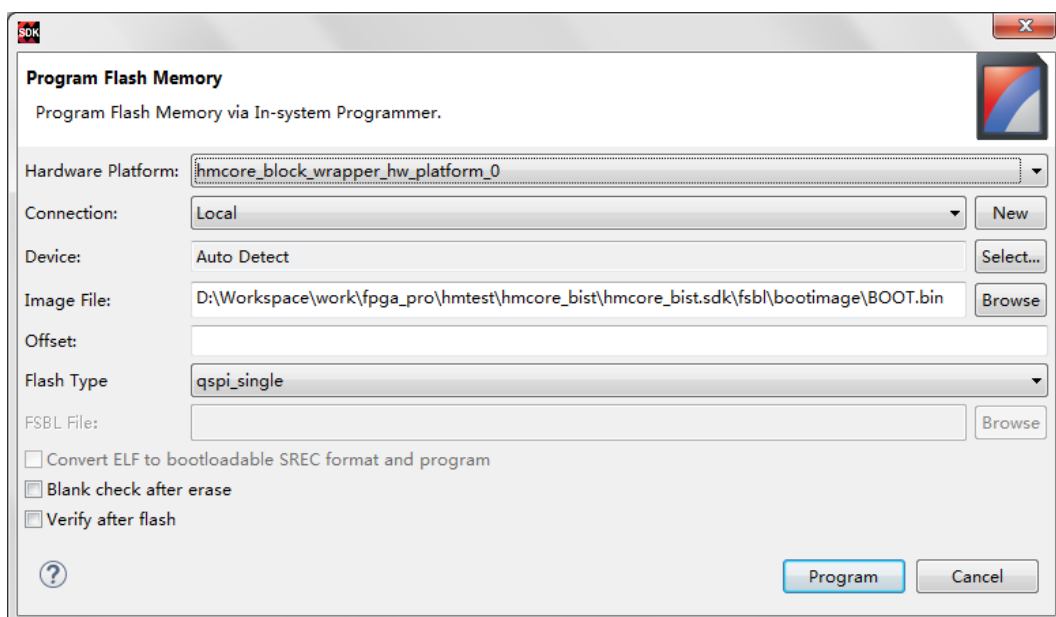
通过命令"`petalinux-bulid`"进行工程编译；

最后通过命令"`petalinux-package --boot --format BIN --fsbl ./images/linux/zynq_fsbl.elf --fp ga ./images/linux/hmcore_block_wrapper.bit --u-boot --force`"生成BOOT.bin。



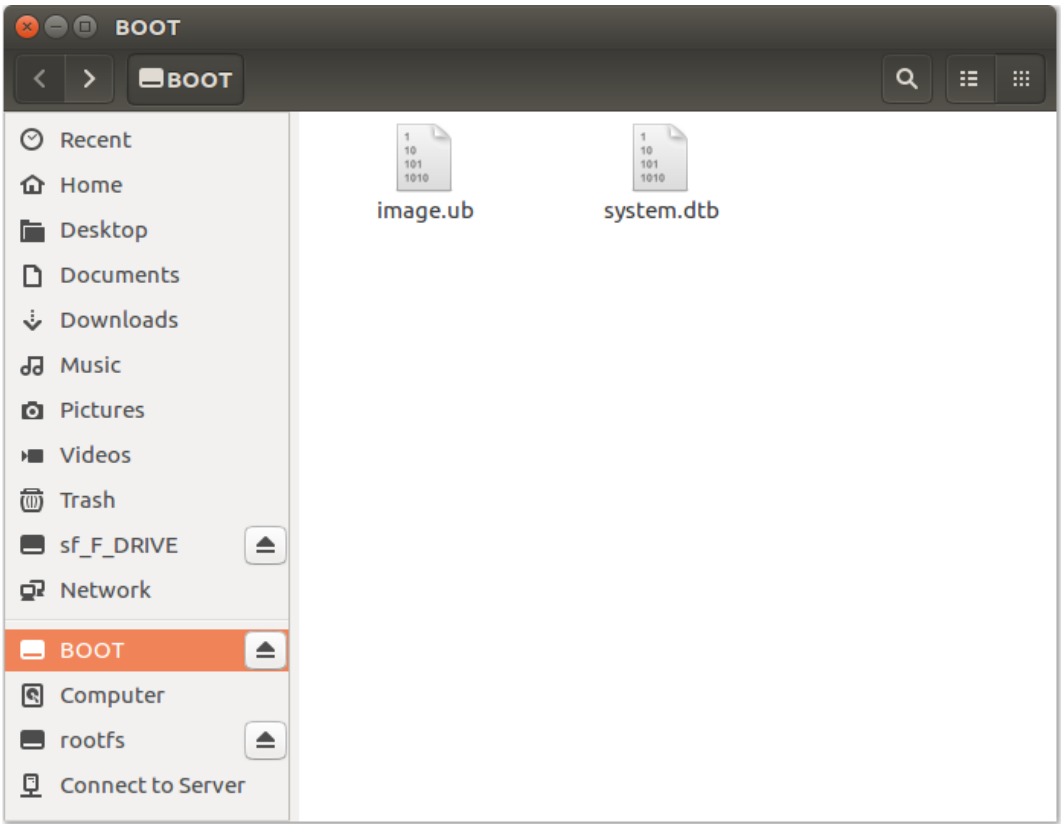
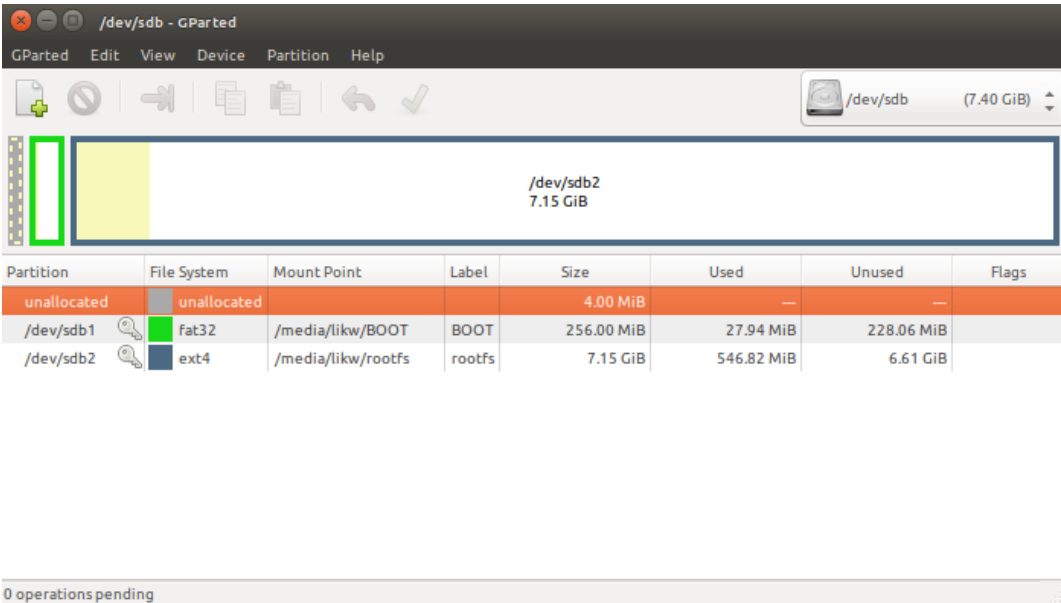
烧写BOOT.bin

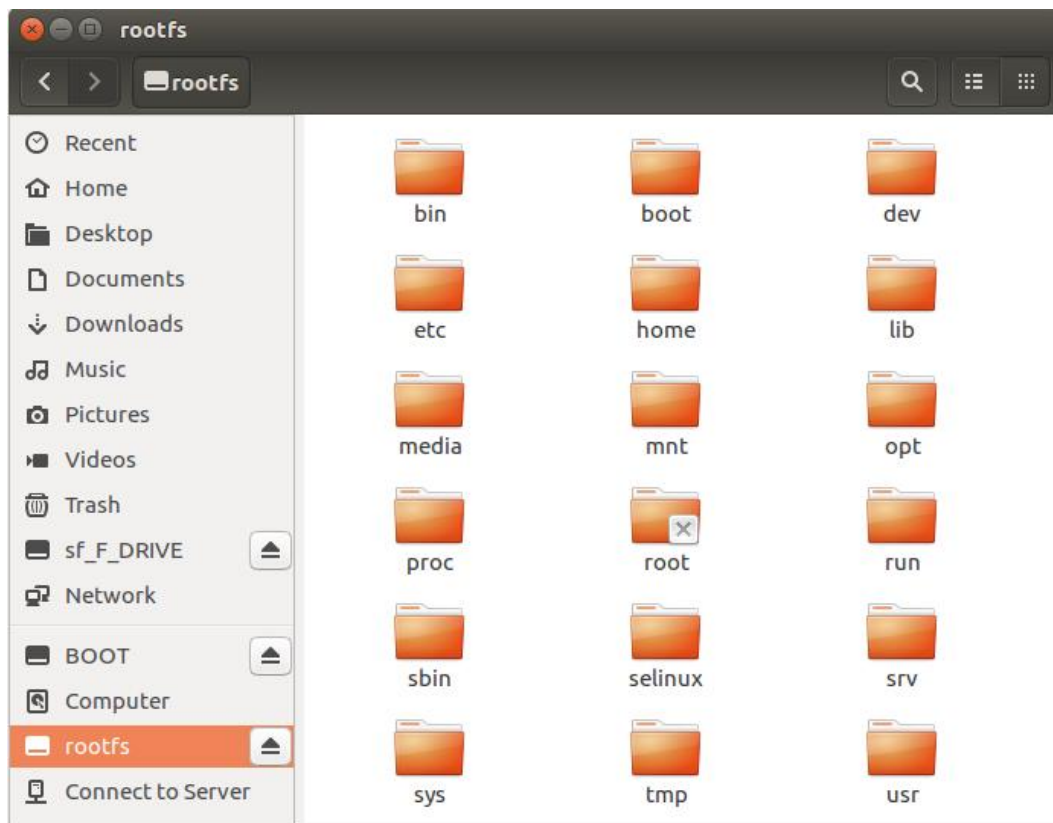
打开XilinxSDK开发环境，点击菜单Xilinx Tools > Program Flash将BOOT.bin文件烧写到QSPI Flash中。



制作SD卡

使用GParted工具对SD卡进行如下分区，将内核、设备树文件复制到BOOT分区，将文件系统复制到rootfs分区。





使用Python开发应用

内置文件系统安装了Python应用，你可以在命令行输入:python或python3启动应用:

```
Last login: Wed Dec 31 16:00:17 PST 1969 on ttyPS0
Welcome to Ubuntu 12.10 (GNU/Linux 4.6.0-xilinx armv7l)

* Documentation: https://help.ubuntu.com/
root@hmcure:~# python3
Python 3.2.3 (default, Oct 19 2012, 21:02:39)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello hmcure!")
Hello hmcure!
>>>
```

需要注意的地方

板载CLG225封装的ZYNQ芯片无法直接从SD卡直接启动，你需要将BOOT.BIN文件烧写到板载QSPI FLASH中启动FSBL、PL逻辑配置流(bit文件)和U-BOOT，然后从SD卡启动内核、加载设备树和文件系统；

HM-CORE可以使用USB形式的无线网卡连接你的WIFI网络与互联网进行连接，但首先需要安装与你的USB无线网卡适配的驱动程序。

应用到不同场景

IP和智能相机	汽车驾驶员辅助，驾驶员信息和信息娱乐
加密货币算法加解密	工业电机控制，工业网络
图像视频处理、无人驾驶、嵌入式视觉	广播相机
智能音箱热词响应	医疗诊断和成像
深度神经网络、机器学习、人工智能	LTE无线电和基带

立即开始你的设计



现在获取你的HM-CORE开发套件

登陆网址: www.hm-lab.com/hm-core获取更多信息

For more information, please visit: www.hm-lab.com/hm-core