Power Optimization of Gateworks Ventana 5200 board

A Project Report

submitted by

M SUDARSAN

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.

April 2016

THESIS CERTIFICATE

This is to certify that the thesis entitled **Power Optimization of Gateworks Ventana 5200 board**, submitted by **M Sudarsan**, to the Indian Institute of Technology Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. V Kamakoti
Research Guide
Professor
Dept. of Computer Science and Engineering
IIT Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

I would like to thank Dr. V. Kamakoti without whose constant support and motivation this project work would not have been possible. I would also like to thank Dr. Vasan and Dr. Shankar for providing valuable technical support and guidance throughout the course of my project. Special thanks to all the network lab personnel for assisting me during my project work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The GW5200 is a member of the Gateworks 5th generation Ventana family of network processor boards targeted for a wide variety of indoor and outdoor applications. The GW5200 features the Freescale™ i.MX6 Dual Core ARM® Cortex™-A9 SoC processor operating at 800MHz, 512Mbytes of DDR3-800 SDRAM, and 256Mbytes of System Flash.The Gateworks System Controller provides embedded features such as real time clock, voltage and temperature monitor, serial EEPROM, programmable pushbutton switch, and advanced power management with programmable board shut-down and wake-up for remote sensor applications. The Freescale™ ARM® Cortex™ A9 i.MX6 CPU includes many peripherals for supporting seamless voice, video, and data connectivity in small embedded applications. The overall system power consumption depends on both software optimization and how the system hardware is implemented. Since the hardware is fixed, the power consumption depends solely on software optimization.

CHAPTER 2

Gateworks Ventana Board

2.1 JTAG

JTAG stands for Joint Test Action Group. JTAG Adapter is used for flashing firmware to Gateworks Boards as well as serial console access. JTAG USB is used to access Gateworks devices for debugging and developing purposes. The following commands are used to configure jtag_usb utility:

```
sudarsan@south-asia:/var/www$ ls /dev/ttyUSB*
/dev/ttyUSB0  /dev/ttyUSB1  /dev/ttyUSB2
sudarsan@south-asia:/var/www$ sudo rmmod ftdi_sio
sudarsan@south-asia:~$ sudo mkdir jtag
sudarsan@south-asia:~$ cd jtag/
sudarsan@south-asia:~/jtag$ sudo wget http://svn.gateworks.com/jtag/linux/
x86/jtag_usbv4
sudarsan@south-asia:~/jtag$ sudo chmod 777 jtag_usbv4
sudarsan@south-asia:~/jtag$ sudo ./jtag_usbv4
```

2.2 Bootloader

The entire bootloader can be updated via JTAG using the Gateworks JTAG dongle and the Linux jtag_usbv4 software:

```
sudo rmmod ftdi_sio
sudo jtag_usbv4 -p u-boot_spl.bin
```

The u-boot.img can be updated from u-boot using the command:

```
tftp ${loadaddr} u-boot.img && mmc erase 0x8a 0x500 && mmc write
${loadaddr} 0x8a 0x500
```

2.3 Building images

Getting the source code

```
git clone https://github.com/Gateworks/openwrt/ gateworks-openwrt -b 16.02
cd gateworks-openwrt
```

To build image:

```
make -C gateworks/ imx6
```

OpenWrt can be customized by running the command:

```
make menuconfig
```

In this new window, you can select/deselect packages you want installed. If kernel customization is required, use the command:

```
make kernel_menuconfig
```

Once customization complete, a simple

```
make -j8 V=s
```

will re-run the make command to re-create the rootfs.
Often the error will be reported at the end of the build. However, the error will have actually occurred earlier on. Scroll up in the logs and look for an error. Typically it is from a package or dependency that was not able to be downloaded. To achieve better debugging information after a failure, re-run the build with the following command:

```
make -j1 V=s
```

or

```
make -j1 V=99
```

To delete the contents of the directories /bin and /build_dir the following command used:

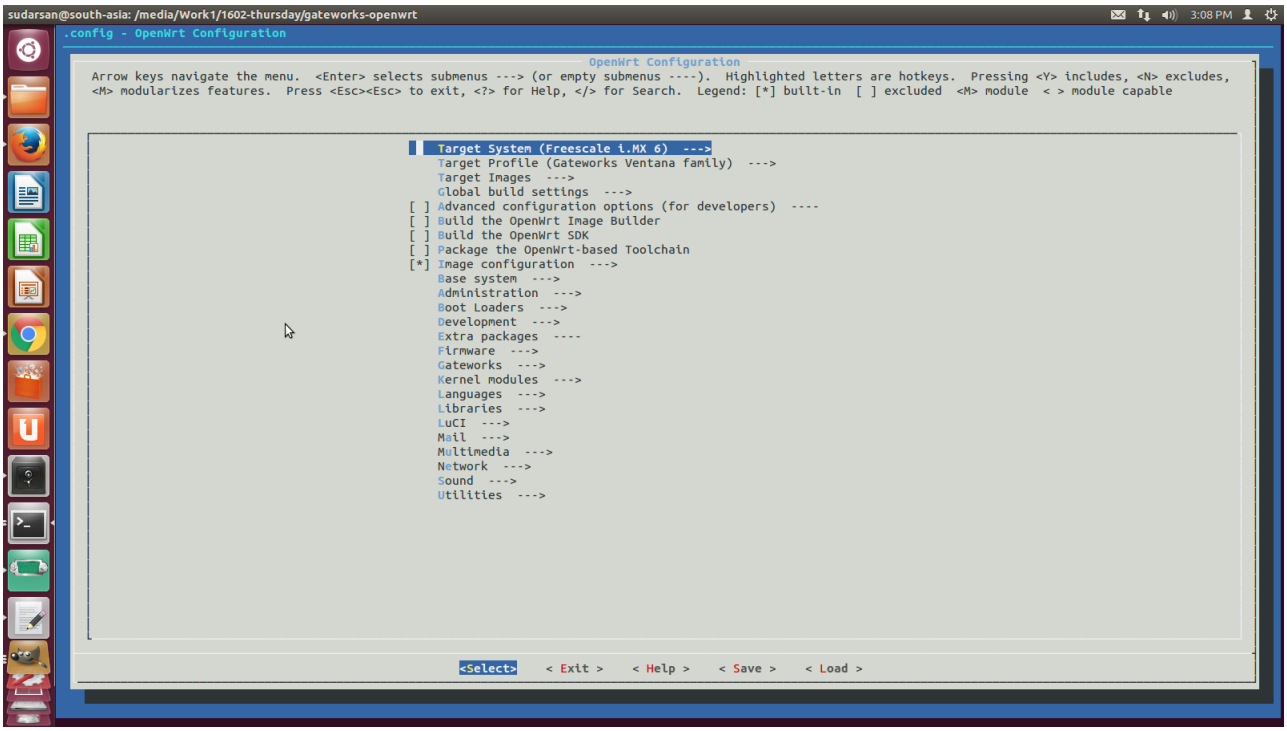
```
make clean
```

The command

```
make dirclean
```

deletes contents of the directories /bin and /build_dir and additionally /staging_dir and /toolchain (=the cross-compile tools) and /logs. Dirclean is the basic "Full clean" operation.

Figure 2.1: menuconfig window



2.4 .config files

In theory there are 2 separate .config files. One is the Gateworks .config in the folder:

```
gateworks/configs/imx6/.config
```

and the other is the default OpenWrt .config in the folder:

```
gateworks-openwrt/.config
```

When the first build is run using the Makefile

```
make -C gateworks/ imx6
```

This basically copies the gateworks .config and overwrites the default .config.

Then, when the *`make menuconfig`* is done, it is now only modifying the default

location .config in the main gateworks-openwrt directory.
The *diff* command can be used to compare two .config files and find the packages missing. It can be used as an useful tool for debugging and detecting the package which causes error.

2.5 TFTP server

The TFTP server used is atftpd. For installing atftpd the command used is:

```
sudo apt-get install atftpd
```

To Modify the config file:

```
USE_INETD=false
OPTIONS="--tftpd-timeout 300 --retry-timeout 5 --port=69
--maxthread 100 --verbose=7 /tftpboot"
```

tftpboot directory's permission and ownership is modified using the command:

```
sudo mkdir /tftpboot
sudo chmod -R 777 /tftpboot
sudo chown -R nobody /tftpboot
```

To start and restart the service:

```
sudo /etc/init.d/atftpd restart
```

To confirm the service is running, naThe overall system power consumption depends on both software optimization and how the system hardware is implemented.vigate to the tftpboot folder and use the command:

```
ps -aef | grep atftpd
```

2.6 Firmware Installation via TFTP

Ventana board is connected to the network using ethernet cable. ipaddress and serverip are set in uboot (boot the board and break into the bootloader command line on the serial console):

```
setenv ipaddr 10.116.65.191
setenv serverip 10.116.65.124
```

ipaddress is set as 10.116.65.191 and serverip as 10.116.65.124.
Set the image to be burnt into the board using the command:

```
setenv image_rootfs gateworks-imx6-ventana-squashfs-nand_normal.ubi
```

Save environment variables:

```
saveenv
```

Run the nand_update script:

```
run nand_update
```

boot:

```
boot
```

To reset the bootloader variables, the following commands are used:

```
env default -f -a
saveenv
reset
```

2.6.1 Power cycle

Power cycling the board sometimes helps to solve the errors arising during booting of customized image.
To power cycle the board, first remove the small coin cell battery on the board (be sure the main board power is unplugged before removing the battery).
Plug the power cable back in and unplug it. Keep repeating it 5 times while watching the serial console.

2.6.2 Network configuration

When the boot process is complete, the following commands can be used to connect the board to the network:

```
ifconfig eth0 192.168.1.1 netmask 255.255.255.0
ifconfig br-lan 10.116.65.191 netmask 255.255.255.0
route add default gw 10.116.65.1 netmask 0.0.0.0
ping 10.116.65.124
```

CHAPTER 3

Dynamic Voltage and Frequency Scaling

Modern processors allow their core clocks to be scaled in order to trade-off performance vs power or simple to save power when performance is not needed. This is referred to as 'Dynamic Frequency Scaling'. In some cases processors also allow you to scale the core voltages down when using lower clock-speeds which is referred to as 'Dynamic Voltage Scaling'.

Governors are kernel models that can drive CPU core frequency/voltage operating points based on an algorithm. Currently the following governors can be used:

1. performance - sets the frequency statically to the highest available CPU frequency
2. powersave - sets the frequency statically to the lowest available CPU frequency
3. userspace - set the frequency from a userspace program
4. ondemand - adjust based on utilization
5. conservative - adjust based on utilization but be a bit more conservative by adjusting gradually

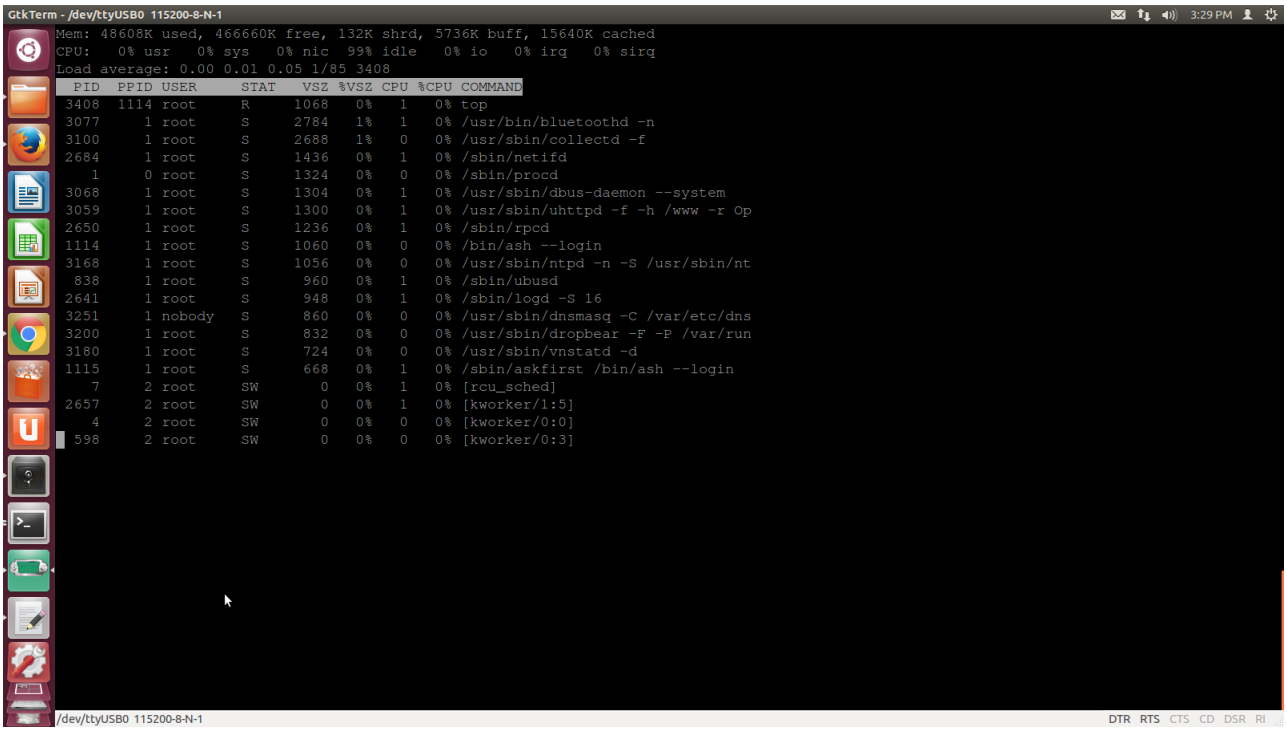
To show the governor for cpu0:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

To show available frequencies (in MHz) for cpu0:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
```

Figure 3.1: Processor usage



Conservative governor can be set for a good mix of power-saving and performance

```
echo conservative > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Set to lowest power mode using the powersave governor:

```
echo powersave > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Set to highest performance mode using the performance governor:

```
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

CHAPTER 4

Interrupt Steering

The 'affinity' of an interrupt handler can be get or set via

```
/proc/irq/<interrupt>/smp_affinity
```

which is a bitmask of what CPU cores the interrupt handler can run on. By default the affinity for each handler is set to allow all available cores (ie for a dual-core system a value of 3 means bit0 (CPU0) and bit1 (CPU1) are both set). If you want a particular interrupt handler to always occur on a specific CPU you can change that bitmask. To see what interrupt handlers are configured and what interrupt they are on look at /proc/interrupts.

```
root@OpenWrt:/# cat /proc/interrupts
           CPU0       CPU1
 16:      18108      13485       GIC  29 Edge      twd
 17:          0          0       GPC  55 Level     i.MX Timer Tick
 18:      28166          0       GPC  13 Level     mxs-dma
 19:       7118          0       GPC  15 Level     bch
 20:          0          0       GPC 115 Level     120000.hdmi
 24:          0          0       GPC  26 Level     2020000.serial
 25:          0          0       GPC  46 Level     2028000.ssi
 37:          0          0  gpio-mxc   4 Edge      gsc
 59:          0          0  gpio-mxc  26 Edge      pps.-1
```

```
237:          0          0  gpio-mxc   0 Edge      2198000.usdhc cd
272:          0          0       GPC  49 Level     imx_thermal
283:          0          0       GPC   2 Level     sdma
284:          0          0       GPC  43 Level     2184000.usb
285:          0          0       GPC  40 Level     2184200.usb
286:      14292          0       GIC 150 Level     2188000.ethernet
287:          0          0       GIC 151 Level     2188000.ethernet
288:          0          0       GPC  24 Level     mmc0
289:         63          0       GPC  36 Level     21a0000.i2c
290:          0          0       GPC  37 Level     21a4000.i2c
291:        299          0       GPC  38 Level     21a8000.i2c
293:       1435          0       GPC  27 Level     21e8000.serial
294:          0          0       GPC  30 Level     21f4000.serial
299:          0          0       gsc   0 Edge      button
300:          0          0       gsc   1 Edge      key-erased
301:          0          0       gsc   2 Edge      eeprom-wp
303:          0          0       gsc   4 Edge      0-0023
304:          0          0       gsc   5 Edge      tamper
306:          0          0       gsc   7 Edge      button-held
307:          0          0   pca953x   0 Edge      gpio_keys
323:          0          0       IPU 457 Edge      (null)
324:          4          0       IPU 451 Edge      (null)
325:      36904          0       IPU  23 Edge      imx_drm
326:          0          0       IPU  28 Edge      imx_drm
327:          2          0       GIC 137 Level     2101000.jr0
328:          0          0       GIC 138 Level     2102000.jr1
```

```
IPI0:          0          0  CPU wakeup interrupts
IPI1:          0          0  Timer broadcast interrupts
IPI2:       2960      17320  Rescheduling interrupts
IPI3:          0          0  Function call interrupts
IPI4:         94       1276  Single function call interrupts
IPI5:          0          0  CPU stop interrupts
IPI6:          0          0  IRQ work interrupts
IPI7:          0          0  completion interrupts
Err:           0
root@OpenWrt:/#
```

APPENDIX A

Loading Image

```
root@OpenWrt:/# reboot

root@OpenWrt:/#

U-Boot SPL 2015.04-gcc7a886 (Dec 17 2015 - 10:27:21)

Booting from NAND

PMIC:  LTC3676

NAND : 256 MiB

U-Boot 2015.04-gcc7a886 (Dec 17 2015 - 10:27:21)

CPU:   Freescale i.MX6DL rev1.2 at 792MHz

CPU:   Industrial temperature grade (-40C to 105C) at 56C

Reset cause: WDOG

 WDOG1 Reset cause: SFTW

 WDOG2 Reset cause: POR

I2C:   ready

DRAM:  512 MiB

PMIC:  LTC3676

NAND:  256 MiB

MMC:   FSL_SDHC: 0

No panel detected: default to HDMI

Display: HDMI (1024x768)

In:    serial

Out:   serial

Err:   serial
```

```
Gateworks Corporation Copyright 2014
Model: GW5200-C1
MFGDate: 03-31-2015
Serial:663022
GSC:   v46 0xd50b WDT:disabled board temp at 46C
RTC:   1457646044
DIO0:  GPIO1_IO16 (gpio-16)
DIO1:  GPIO1_IO19 (gpio-19)
DIO2:  GPIO1_IO17 (gpio-17)
DIO3:  GPIO1_IO20 (gpio-20)
RS232: enabled
Net:   FEC [PRIME], usb_ether
Error: usb_ether address not set.
Hit any key to stop autoboot:  0
Ventana >
Ventana > setenv ipaddr 10.116.65.191
Ventana > setenv serverip 10.116.65.124
Ventana > setenv image_rootfs gateworks-imx6-ventana-squashfs-
nand_normal.ubi
Ventana > saveenv
Saving Environment to NAND...
Erasing redundant NAND...
Erasing at 0x1080000 -- 100% complete.
Writing to redundant NAND... OK
Ventana > run nand_update
Updating NAND from 10.116.65.124:gateworks-imx6-ventana-squashfs-
```

```
nand_normal.ubi...
Using FEC device
TFTP from server 10.116.65.124; our IP address is 10.116.65.191
Filename 'gateworks-imx6-ventana-squashfs-nand_normal.ubi'.
Load address: 0x12000000
Loading: ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
```

```
 ##################################################################
 ##################################################################
 ##################################################################
 ############
 13.9 MiB/s
done
Bytes transferred = 23068672 (1600000 hex)
NAND erase.part: device 0 offset 0x1100000, size 0xef00000
Erasing at 0xffe0000 -- 100% complete.
OK
NAND write: device 0 offset 0x1100000, size 0x1600000
 23068672 bytes written: OK
Ventana > boot
Attempting usb boot...
starting USB...
USB0:   Port not available.
USB1:   USB EHCI 1.00
scanning bus 1 for devices... 1 USB Device(s) found
       scanning usb for storage devices... 0 Storage Device(s) found
       scanning usb for ethernet devices... 0 Ethernet Device(s) found
USB device 0: unknown device
** Bad device usb 0 **
** Bad device usb 0 **
Attempting mmc boot...
MMC: no card present
MMC: no card present
```

```
** Bad device mmc 0 **
MMC: no card present
** Bad device mmc 0 **
Attempting sata boot...
** Bad device size - sata 0 **
** Bad device size - sata 0 **
Attempting flash boot...
UBI: attaching mtd1 to ubi0
UBI: scanning is finished
UBI: volume 2 ("rootfs_data") re-sized from 9 to 1597 LEBs
UBI: attached mtd1 (name "mtd=2", size 239 MiB) to ubi0
UBI: PEB size: 131072 bytes (128 KiB), LEB size: 126976 bytes
UBI: min./max. I/O unit sizes: 2048/2048, sub-page size 2048
UBI: VID header offset: 2048 (aligned 2048), data offset: 4096
UBI: good PEBs: 1912, bad PEBs: 0, corrupted PEBs: 0
UBI: user volume: 3, internal volumes: 1, max. volumes count: 128
UBI: max/mean erase counter: 1/0, WL threshold: 4096,
image sequence number: 845547390
UBI: available PEBs: 0, total reserved PEBs: 1912,
PEBs reserved for bad PEB handling: 40
Loading file '/6x_bootscript-ventana' to addr 0x12000000
 with size 2423 (0x00000977)...
Done
## Executing script at 12000000
Gateworks Ventana OpenWrt Boot script v1.00
Using dtype from env: nand
```

```
Booting from NAND...
mtdparts:mtdparts=nand:16m(uboot),1m(env),239m(ubi)
Loading file '/uImage' to addr 0x10800000 with size 1536408 (0x00177198)...
Done
** File not found /imx6dl-gw5200-c1.dtb **
** File not found /imx6dl-gw5200.dtb **
Loading file '/imx6dl-gw52xx.dtb' to addr 0x18000000 with size
34293 (0x000085f5)...
Done
Loaded DTB from /imx6dl-gw52xx.dtb
## Booting kernel from Legacy Image at 10800000 ...
   Image Name:   ARM OpenWrt Linux-4.4
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    1536344 Bytes = 1.5 MiB
   Load Address: 10008000
   Entry Point:  10008000
   Verifying Checksum ... OK
## Flattened Device Tree blob at 18000000
   Booting using the fdt blob at 0x18000000
EHCI failed to shut down host controller.
   Loading Kernel Image ... OK
   Using Device Tree in place at 18000000, end 1800b5f4
   Updating MTD partitions...
   Adjusting FDT per EEPROM for gw5200-c1...
   Config LDO-enabled mode
Starting kernel ...
```

```
Press the [f] key and hit [enter] to enter failsafe mode
Press the [1], [2], [3] or [4] key and hit [enter] to select the debug level
Failed to connect to ubus
Please press Enter to activate this console.
BusyBox v1.24.1 () built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 DESIGNATED DRIVER (Gateworks Ventana 16.02@4fba0b9, r48868)
 -----------------------------------------------------
  * 2 oz. Orange Juice         Combine all juices in a
  * 2 oz. Pineapple Juice      tall glass filled with
  * 2 oz. Grapefruit Juice     ice, stir well.
  * 2 oz. Cranberry Juice
 -----------------------------------------------------
root@(none):/#
root@(none):/#
```

References

http://trac.gateworks.com/wiki/OpenWrt/building#StepbyStepInstructions

http://trac.gateworks.com/wiki/ventana/openwrt

http://trac.gateworks.com/wiki/tftpserver

http://trac.gateworks.com/wiki/jtag_instructions#CreatingjtagableimagesforVentana

http://trac.gateworks.com/wiki/OpenWrt/Configuration

http://trac.gateworks.com/wiki/ventana/power

http://trac.gateworks.com/wiki/DVFS

http://trac.gateworks.com/wiki/multicoreprocessing