Lime Microsystems Limited

Surrey Technology Centre
Occam Road
The Surrey Research Park
Guildford, Surrey GU2 7YG
United Kingdom

Tel:	+44 (0) 1483 685 063
Fax:	+44 (0) 1428 656 662
e-mail:	enquiries@limemicro.com

STREAM Development kit
User Guide

- User Guide for Linux OS Installation -

Version: 1.1

REVISION HISTORY

The following table shows the revision history of this document:

Date	Version	Description of Revisions
27/02/2015	1.0	Initial version
03/04/2015	1.1	Added document licence info (Creative Commons Attribution 3.0 Unported)

Table of Contents

Table of Figures

1 Requirements

Required Software:

- CyControl (Cypress USB control center)
- OpenOCD (>=0.8.0)
- Tera Term
- Zadig

Required Hardware:

- Altera USB Blaster
- 3.3V USB-to-Serial cable
- USB A to microB cable

Current manual describes how to load Linux image to FPGA via FX3 controller.
Information regarding software architecture and links to source codes are available in `\Stream\stream_openrisc_soc_v2\doc` directory.

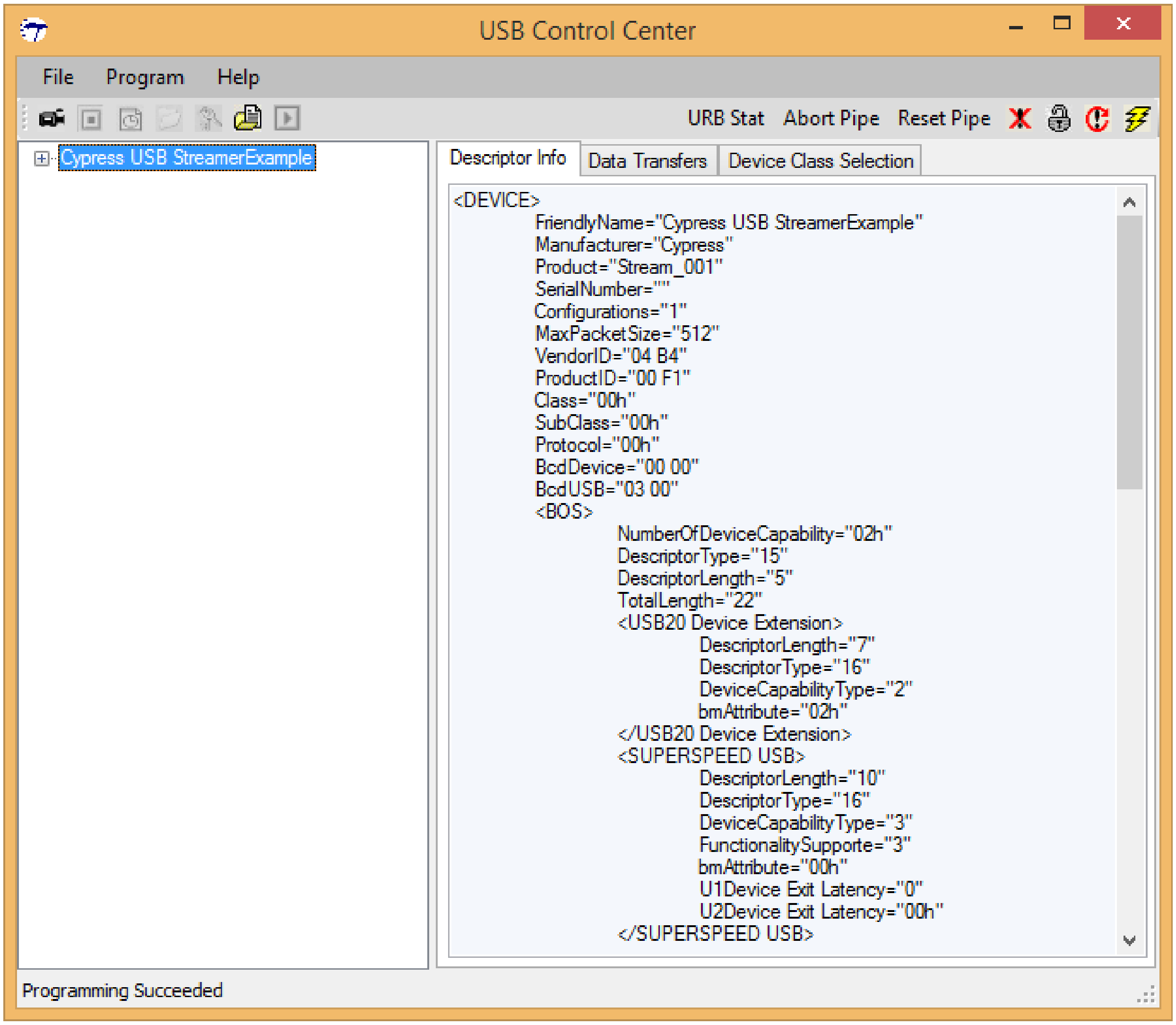
2 Step by step instructions

Step 1. Installing USB drivers

Use J2 USB3.0 to connect to PC.
To install Cypress FX3 USB3.0 (IC6) drivers navigate driver installer to `\Stream\Stream_LMS7EVB_distro_05v\fx3\drivers\bin\`

Step 2. Load FX3 firmware

Run CyControl from `\Stream\Stream_LMS7EVB_distro_05v\fx3\software`
Press Program → FX3 → RAM → navigate to …`\fx3\firmware_img\usb_trx_V2` → Open.
“Cypress USB BootLoader” should change to “Cypress USB StreamerExample”.



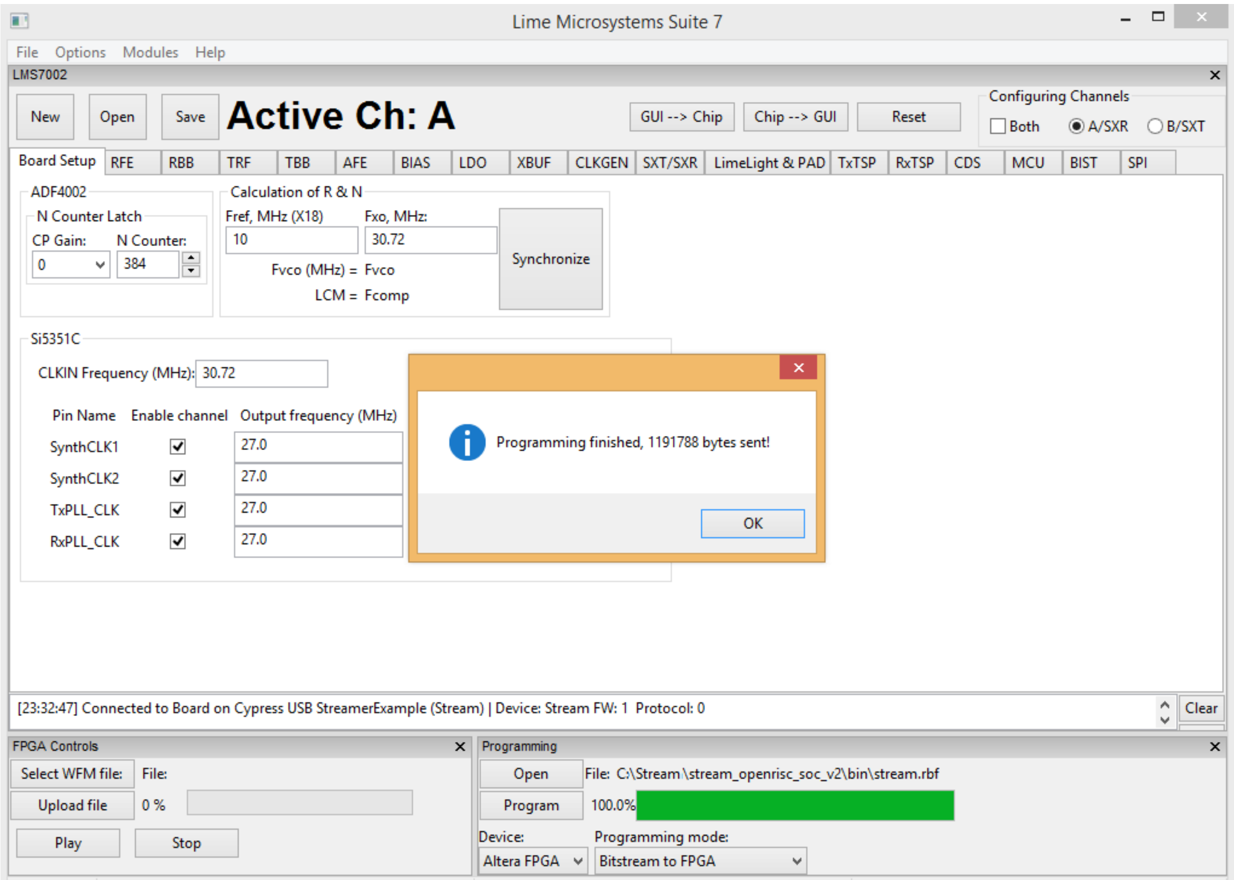
```
<DEVICE>
        FriendlyName="Cypress USB StreamerExample"
        Manufacturer="Cypress"
        Product="Stream_001"
        SerialNumber=""
        Configurations="1"
        MaxPacketSize="512"
        VendorID="04 B4"
        ProductID="00 F1"
        Class="00h"
        SubClass="00h"
        Protocol="00h"
        BcdDevice="00 00"
        BcdUSB="03 00"
        <BOS>
                NumberOfDeviceCapability="02h"
                DescriptorType="15"
                DescriptorLength="5"
                TotalLength="22"
                <USB20 Device Extension>
                        DescriptorLength="7"
                        DescriptorType="16"
                        DeviceCapabilityType="2"
                        bmAttribute="02h"
                </USB20 Device Extension>
                <SUPERSPEED USB>
                        DescriptorLength="10"
                        DescriptorType="16"
                        DeviceCapabilityType="3"
                        FunctionalitySupporte="3"
                        bmAttribute="00h"
                        U1Device Exit Latency="0"
                        U2Device Exit Latency="00h"
                </SUPERSPEED USB>
```

Programming Succeeded

Figure . CyControl Cypress USB StreamerExample should appear

Step 3. LMS7Suite for FPGA programming

Run LMS7Suite from `\Stream\Stream_LMS7EVB_distro_05v\gui directory`.
Go to Options → Connection Settings → Cypress USB StreamerExample (Stream) → OK
Now connection between FX3 controller and PC is set.
Now it is time to load RISC architecture in to FPGA: go to Programming section. Press Open and navigate to `\Stream\stream_openrisc_soc_v2\bin\stream.rbf`. Press Program.


Lime Microsystems Suite 7
File Options Modules Help
LMS7002
New Open Save
GUI --> Chip Chip --> GUI Reset
Board Setup RFE RBB TRF TBB AFE BIAS LDO XBUF CLKGEN SXT/SXR LimeLight & PAD TxTSP RxTSP CDS MCU BIST SPI
Synchronize
Select WFM file: Upload file Play Stop
Open Program
Clear

Active Ch: A

Configuring Channels: □ Both ● A/SXR ○ B/SXT

ADF4002 – N Counter Latch: CP Gain: 0; N Counter: 384

Calculation of R & N: Fref, MHz (X18): 10; Fxo, MHz: 30.72; Fvco (MHz) = Fvco; LCM = Fcomp

Si5351C – CLKIN Frequency (MHz): 30.72

Pin Name	Enable channel	Output frequency (MHz)
SynthCLK1	☑	27.0
SynthCLK2	☑	27.0
TxPLL_CLK	☑	27.0
RxPLL_CLK	☑	27.0

Programming finished, 1191788 bytes sent!

OK

[23:32:47] Connected to Board on Cypress USB StreamerExample (Stream) | Device: Stream FW: 1 Protocol: 0

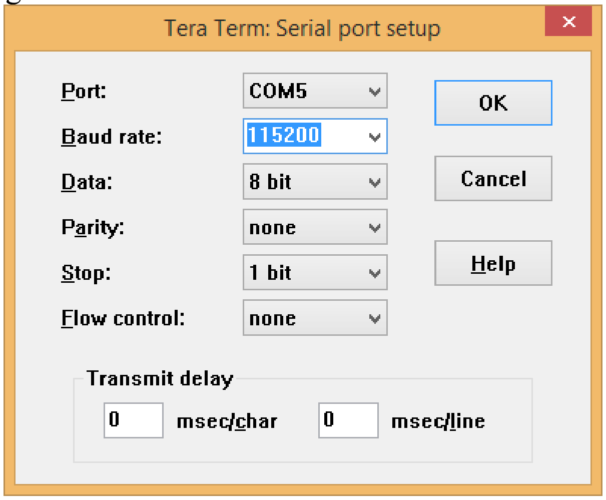
FPGA Controls – File: ; 0 %

Programming – File: C:\Stream\stream_openrisc_soc_v2\bin\stream.rbf; 100.0%; Device: Altera FPGA; Programming mode: Bitstream to FPGA

Figure . FPGA programming via LMS7Suite

Step 4. Serial connection setup

Connect 3.3V USB-to-Serial cable to J1 (FPGA GPIOs header).
Pin 1: FPGA RX pin (**orange**)
Pin 3: FPGA TX pin (**yellow**)
Pin 10: GND (**black**)
Establish connection using Tera Term via Serial Port.



Tera Term: Serial port setup

Port: COM5
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

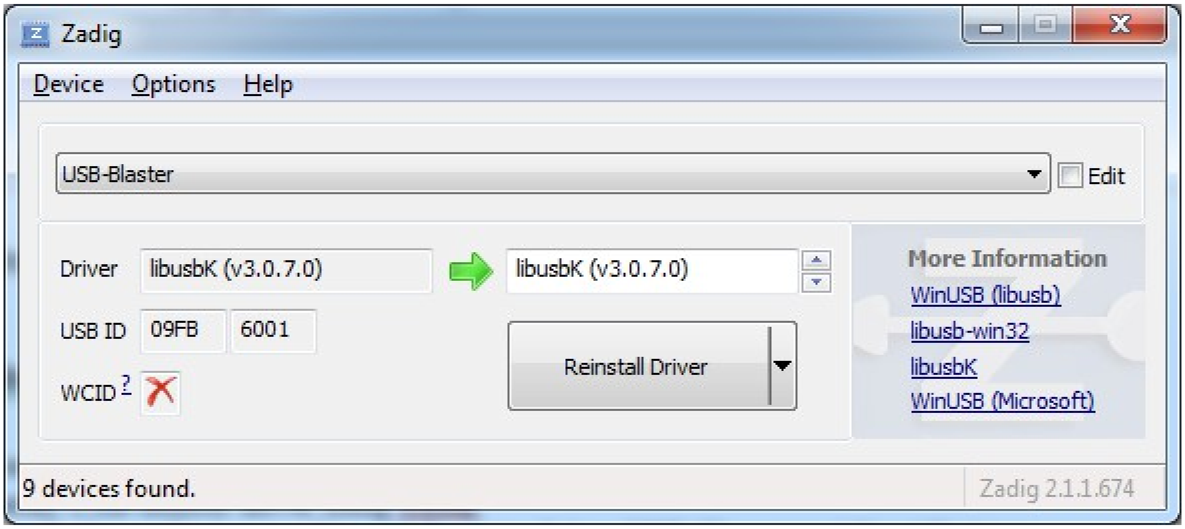
Transmit delay: 0 msec/char 0 msec/line

Figure . Tera Term Serial Port setup

FPGA is not ready yet, so Terminal window is empty.

Step 5. Altera USB Blaster

Connect Altera USB Blaster to J23 FPGA JTAG header.
Install USB-Blaster driver using Zadig.
Run Zadig → Options → List All Devices. Select USB-Blaster and libusbK driver.



USB-Blaster

Driver libusbK (v3.0.7.0) → libusbK (v3.0.7.0)

USB ID 09FB 6001

WCID ?

9 devices found.

Zadig 2.1.1.674

Figure . Zadig driver installer

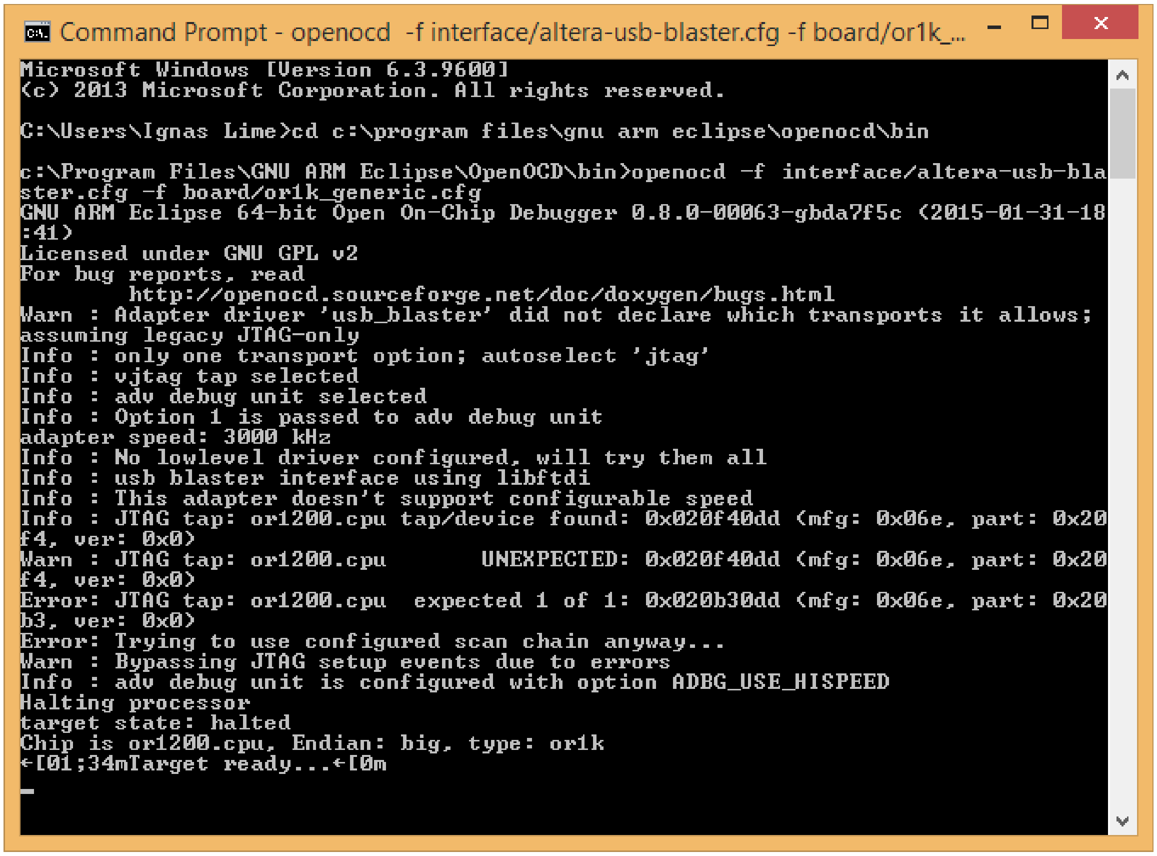
Step 6. OpenOCD installation

Install OpenOCD.
Copy **vmlinux** (Linux image) from `\stream_openrisc_soc_v2\bin` to OpenOCD installation directory `C:\Program Files\GNU ARM Eclipse\OpenOCD\bin`
Run Windows Command Prompt.
Change current directory to OpenOCD installation directory:

```
cd C:\Program Files\GNU ARM Eclipse\OpenOCD\bin
```

Launch OpenOCD with:

```
Openocd -f interface/altera-usb-blaster.cfg -f board/or1k_generic.cfg
```


Command Prompt - openocd -f interface/altera-usb-blaster.cfg -f board/or1k_...

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Ignas Lime>cd c:\program files\gnu arm eclipse\openocd\bin

c:\Program Files\GNU ARM Eclipse\OpenOCD\bin>openocd -f interface/altera-usb-bla
ster.cfg -f board/or1k_generic.cfg
GNU ARM Eclipse 64-bit Open On-Chip Debugger 0.8.0-00063-gbda7f5c (2015-01-31-18
:41)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.sourceforge.net/doc/doxygen/bugs.html
Warn : Adapter driver 'usb_blaster' did not declare which transports it allows;
assuming legacy JTAG-only
Info : only one transport option; autoselect 'jtag'
Info : vjtag tap selected
Info : adv debug unit selected
Info : Option 1 is passed to adv debug unit
adapter speed: 3000 kHz
Info : No lowlevel driver configured, will try them all
Info : usb blaster interface using libftdi
Info : This adapter doesn't support configurable speed
Info : JTAG tap: or1200.cpu tap/device found: 0x020f40dd (mfg: 0x06e, part: 0x20
f4, ver: 0x0)
Warn : JTAG tap: or1200.cpu       UNEXPECTED: 0x020f40dd (mfg: 0x06e, part: 0x20
f4, ver: 0x0)
Error: JTAG tap: or1200.cpu  expected 1 of 1: 0x020b30dd (mfg: 0x06e, part: 0x20
b3, ver: 0x0)
Error: Trying to use configured scan chain anyway...
Warn : Bypassing JTAG setup events due to errors
Info : adv debug unit is configured with option ADBG_USE_HISPEED
Halting processor
target state: halted
Chip is or1200.cpu, Endian: big, type: or1k
←[01;34mTarget ready...←[0m
_
```

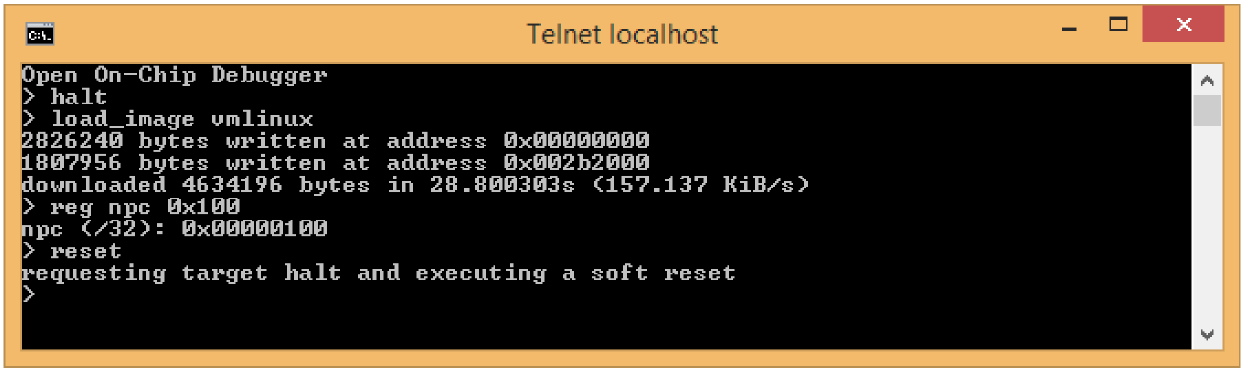
Figure . Launching OpenOCD

Step 7. Installing Linux

Turn ON Windows Telnet (Control Panel → Programs → Turn Windows features on or off)

Open a new Command Prompt window and connect to the OpenOCD proxy with `telnet localhost 4444`.

Run the following command to load the Linux image `halt; load_image vmlinux; reg npc 0x100; reset`.



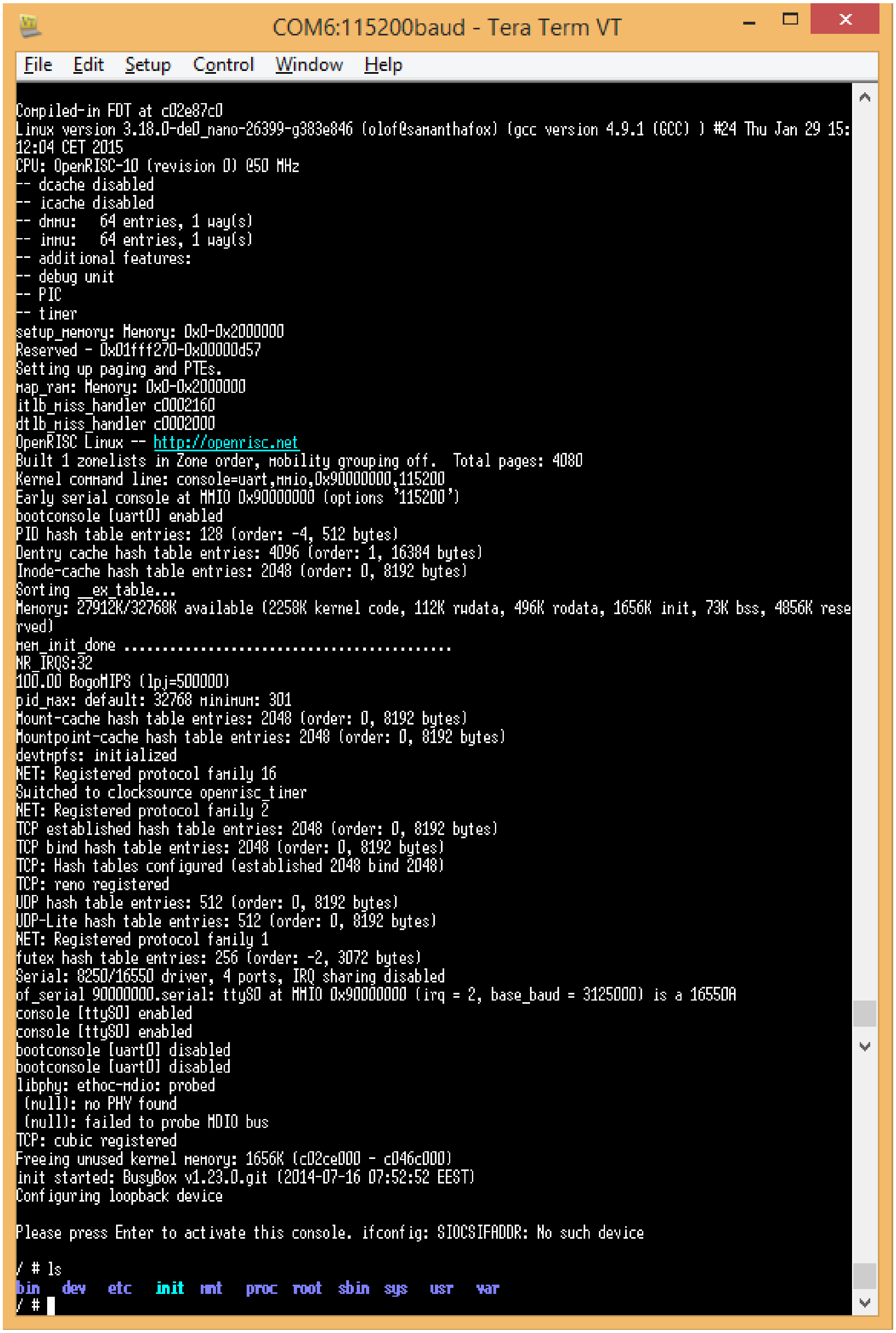
```
Open On-Chip Debugger
> halt
> load_image vmlinux
2826240 bytes written at address 0x00000000
1807956 bytes written at address 0x002b2000
downloaded 4634196 bytes in 28.800303s (157.137 KiB/s)
> reg npc 0x100
npc (/32): 0x00000100
> reset
requesting target halt and executing a soft reset
>
```

Figure . Loading Linux image

The Linux boot process should now start in the terminal connected to the serial console.

After the boot is complete you should be greeted by a message saying `Please press Enter to activate this console.`

Press Enter and you can start using Linux commands.


COM6:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Compiled-in FDT at c02e87c0
Linux version 3.18.0-de0_nano-26399-g383e846 (olof@samanthafox) (gcc version 4.9.1 (GCC) ) #24 Thu Jan 29 15:
12:04 CET 2015
CPU: OpenRISC-10 (revision 0) @50 MHz
-- dcache disabled
-- icache disabled
-- dmmu:   64 entries, 1 way(s)
-- immu:   64 entries, 1 way(s)
-- additional features:
-- debug unit
-- PIC
-- timer
setup_memory: Memory: 0x0-0x2000000
Reserved - 0x01fff270-0x00000d57
Setting up paging and PTEs.
map_ram: Memory: 0x0-0x2000000
itlb_miss_handler c0002160
dtlb_miss_handler c0002000
OpenRISC Linux -- http://openrisc.net
Built 1 zonelists in Zone order, mobility grouping off.  Total pages: 4080
Kernel command line: console=uart,mmio,0x90000000,115200
Early serial console at MMIO 0x90000000 (options '115200')
bootconsole [uart0] enabled
PID hash table entries: 128 (order: -4, 512 bytes)
Dentry cache hash table entries: 4096 (order: 1, 16384 bytes)
Inode-cache hash table entries: 2048 (order: 0, 8192 bytes)
Sorting __ex_table...
Memory: 27912K/32768K available (2258K kernel code, 112K rwdata, 496K rodata, 1656K init, 73K bss, 4856K rese
rved)
mem_init_done ..........................................
NR_IRQS:32
100.00 BogoMIPS (lpj=500000)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 2048 (order: 0, 8192 bytes)
Mountpoint-cache hash table entries: 2048 (order: 0, 8192 bytes)
devtmpfs: initialized
NET: Registered protocol family 16
Switched to clocksource openrisc_timer
NET: Registered protocol family 2
TCP established hash table entries: 2048 (order: 0, 8192 bytes)
TCP bind hash table entries: 2048 (order: 0, 8192 bytes)
TCP: Hash tables configured (established 2048 bind 2048)
TCP: reno registered
UDP hash table entries: 512 (order: 0, 8192 bytes)
UDP-Lite hash table entries: 512 (order: 0, 8192 bytes)
NET: Registered protocol family 1
futex hash table entries: 256 (order: -2, 3072 bytes)
Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
of_serial 90000000.serial: ttyS0 at MMIO 0x90000000 (irq = 2, base_baud = 3125000) is a 16550A
console [ttyS0] enabled
console [ttyS0] enabled
bootconsole [uart0] disabled
bootconsole [uart0] disabled
libphy: ethoc-mdio: probed
 (null): no PHY found
 (null): failed to probe MDIO bus
TCP: cubic registered
Freeing unused kernel memory: 1656K (c02ce000 - c046c000)
init started: BusyBox v1.23.0.git (2014-07-16 07:52:52 EEST)
Configuring loopback device

Please press Enter to activate this console. ifconfig: SIOCSIFADDR: No such device

/ # ls
bin   dev   etc   init  mnt   proc  root  sbin  sys   usr   var
/ #
```

Figure . Linux loading completed