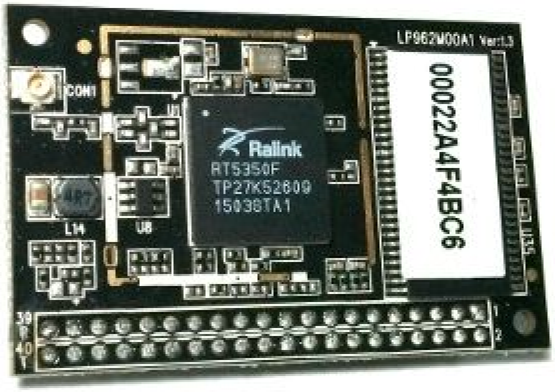
NixCore X1
Upgrading Firmware

NIXD01002
Revision 1.0 - October 2015

NixCore Website:
http://nixcores.com

NixCore X1 Product Page:
http://nixcores.com/nixcore_x1.php



Overview

OpenWRT is a GNU Linux based operating system designed for embedded devices such as the NixCore X1 Computer On Module (COM). The NixCore Operating system is based on OpenWRT 15.05 “Chaos Calmer”. To update the NixCore X1 operating system with a new build you will need the following:

The bare minimum:

- Serial cable (USB to RS232, or USB to TTL serial)
- A NixCore X1 system image
- A terminal application
  - Putty (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)
- Network patch or “crossover” cable
- A computer with a TFTP server
  - https://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol

NOTE: These instructions are how to wipe the Linux system from your NixCore and install a new one. No data will be saved between images, backup all data before flashing OS images.

Summary:

The NixCore X1 contains a copy of U-Boot as the bootloader for the system. U-Boot is the first thing to run on the NixCore X1 and setups the system for the real Linux kernel. Next the Linux kernel is loaded into memory and started. This is similar to how the BIOS works on a PC. U-Boot also contains the ability to read and write to the entire flash storage and thus can update the Linux image on the NixCore X1.

The NixCore X1 is updatable using Ethernet and the TFTP download protocol.

Network Install

The Trivial File Transfer Protocol (TFTP) is a simple protocol which allows for data to be sent via a network. U-Boot supports this protocol and can retrieve files from any TFTP server on a network. Because we are downloading via the network the NixCore X1 must have an Ethernet network connection. The NixCore X1DEV board has a RJ-45 network jack that will allow U-Boot to talk on a network

Network configuration:

It is highly recommended to attach the NixCore X1 to a secondary network that is under the full control of the developer. The U-Boot application does not have support for DHCP or DNS and will be configured with a static IP for TFTP transfer. A additional ethernet connection on the development PC can allow for a single wire connection between the NixCore X1 and the development PC.

Setting up a TFTP server:

GNU Linux

Linux machines can install a tftp server from software repositories. Following are instructions for popular distributions:

- Ubuntu: http://askubuntu.com/questions/201505/how-do-i-install-and-run-a-tftp-server
- Fedora: https://docs.fedoraproject.org/en-US/Fedora/18/html/Installation_Guide/ch13s05s03.html
- Debian: http://www.cyberciti.biz/faq/install-configure-tftp-server-ubuntu-debian-howto/
- Arch: https://wiki.archlinux.org/index.php/TFTP

Windows:

There are a number of TFTP servers for Windows. Tftputil has been tested with NixCore and confirmed to work: http://sourceforge.net/projects/tftputil/

The system image is a binary file named in the format XXXXXX-sysupgrade.bin and contains all the files needed as well as the Linux kernel. Download a firmware from the NixCore website or build one and save the sysupgrade.bin file to the hosting directory of the TFTP server.

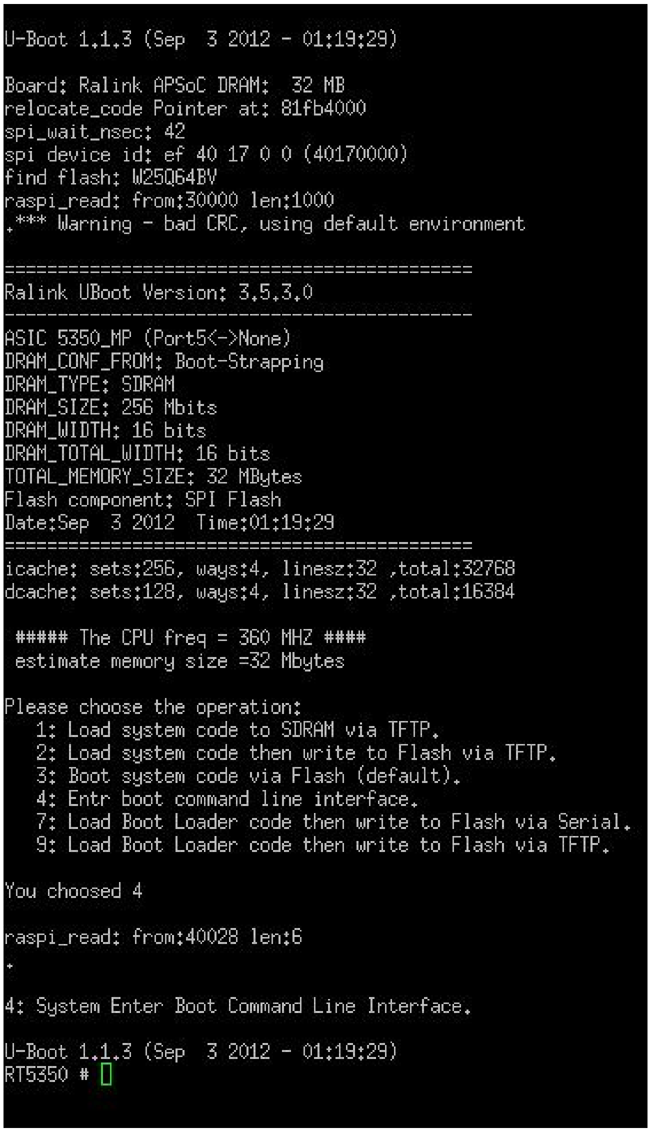
Connect to the NixCore X1:

A terminal application such as Putty should be used to connect to the UART2 pins of the NixCore X1. If you have the NixCore X1DEV the serial port is connected to UART2.

Apply power to the the NixCore X1 and immediately send the '4' character from the terminal application. This will tell the bootloader to stop booting the system and allow the user to update the flash.

If you see kernel boot messages on the terminal remove power and try again.

This is the screen you should see:



```
U-Boot 1.1.3 (Sep  3 2012 - 01:19:29)

Board: Ralink APSoC DRAM:  32 MB
relocate_code Pointer at: 81fb4000
spi_wait_nsec: 42
spi device id: ef 40 17 0 0 (40170000)
find flash: W25Q64BV
raspi_read: from:30000 len:1000
.*** Warning - bad CRC, using default environment

============================================
Ralink UBoot Version: 3.5.3.0
--------------------------------------------
ASIC 5350_MP (Port5<->None)
DRAM_CONF_FROM: Boot-Strapping
DRAM_TYPE: SDRAM
DRAM_SIZE: 256 Mbits
DRAM_WIDTH: 16 bits
DRAM_TOTAL_WIDTH: 16 bits
TOTAL_MEMORY_SIZE: 32 MBytes
Flash component: SPI Flash
Date:Sep  3 2012  Time:01:19:29
============================================
icache: sets:256, ways:4, linesz:32 ,total:32768
dcache: sets:128, ways:4, linesz:32 ,total:16384

 ##### The CPU freq = 360 MHZ ####
 estimate memory size =32 Mbytes

Please choose the operation:
   1: Load system code to SDRAM via TFTP.
   2: Load system code then write to Flash via TFTP.
   3: Boot system code via Flash (default).
   4: Entr boot command line interface.
   7: Load Boot Loader code then write to Flash via Serial.
   9: Load Boot Loader code then write to Flash via TFTP.

You choosed 4

raspi_read: from:40028 len:6
.

4: System Enter Boot Command Line Interface.

U-Boot 1.1.3 (Sep  3 2012 - 01:19:29)
RT5350 # 
```

Setting the UBoot Environment:

U-Boot has a number of variables that it stores on flash, including a static IP address for the board and the server IP address for where to look for a TFTP server. Since U-Boot does not have a DHCP server these variables need to be updated for your network setup. Before changing variables determine the IP address of the TFTP server and an unused IP address on the network the board is connected to.

The way to update a U-Boot variable is with the ‘setenv’ command. Here are the commands to update the local IP address for the board and the server IP address:

```
setenv ipaddr 192.168.1.123
setenv serverip 192.168.1.5
```

This will temporarily configure the board to have IP address 192.168.1.123. Once a TFTP transfer is started the board will try and connect to a TFTP server at IP 192.168.1.5.

If you wish to save these settings all variables can be written to flash with the ‘saveenv’ command.

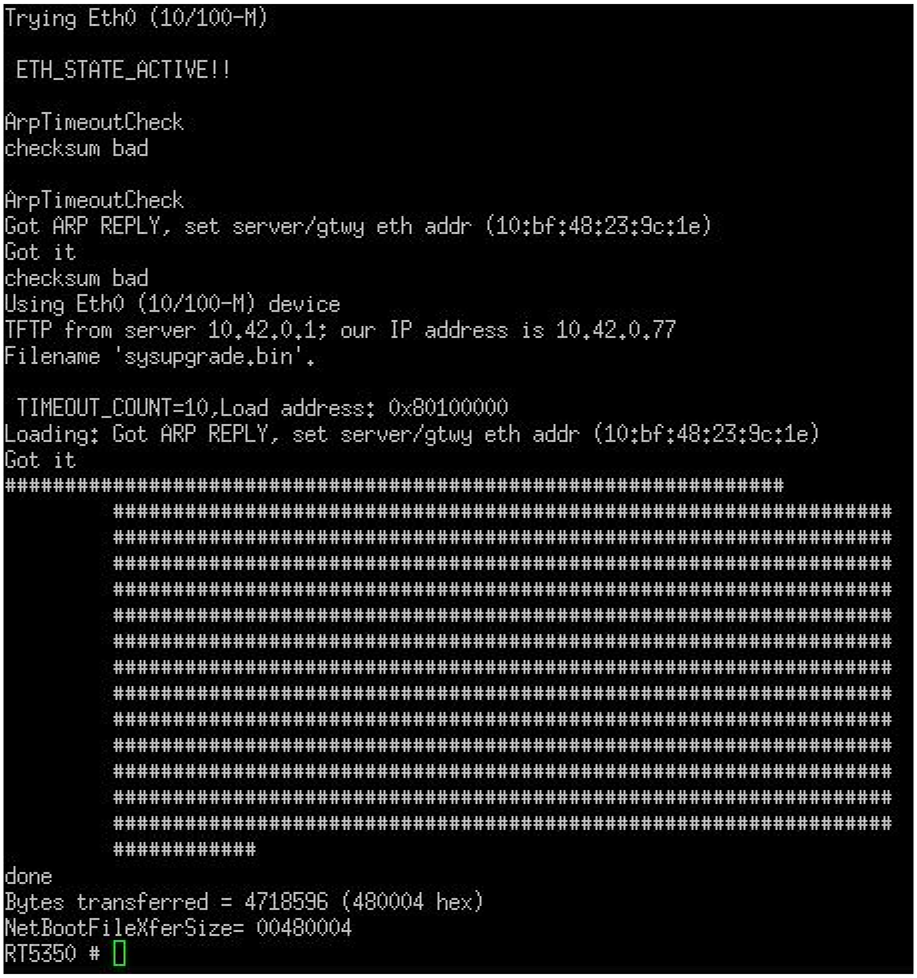
Downloading via tftp:

The tftpboot command will start a TFTP transfer and load the given file to the given RAM address. As the file is downloaded U-Boot shows a series of ‘#’ chracters, if you do not see these characters check your network settings, file permission settings on the server, and physical connection.

Replace sysupgrade.bin with the filename of the image you wish to flash to the board

```
tftpboot 0x80100000 sysupgrade.bin
```

Example of a transfer:



```
Trying Eth0 (10/100-M)

 ETH_STATE_ACTIVE!!

ArpTimeoutCheck
checksum bad

ArpTimeoutCheck
Got ARP REPLY, set server/gtwy eth addr (10:bf:48:23:9c:1e)
Got it
checksum bad
Using Eth0 (10/100-M) device
TFTP from server 10.42.0.1; our IP address is 10.42.0.77
Filename 'sysupgrade.bin'.

 TIMEOUT_COUNT=10,Load address: 0x80100000
Loading: Got ARP REPLY, set server/gtwy eth addr (10:bf:48:23:9c:1e)
Got it
#################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         ############
done
Bytes transferred = 4718596 (480004 hex)
NetBootFileXferSize= 00480004
RT5350 # 
```

After the transfer is finished take note of the number of bytes transfered. Write down the hex value from the transfer (In this example it is 480004)

Writing the Image:

After the file is downloaded via TFTP it is stored in RAM, to move it from RAM to flash you need to erase the Linux system section of the flash and write this new data in its place. The following are the U-Boot commands to write the image

Erase the Linux system section: (*NOTE: This will wipe the device*)

erase linux

Write the new image to the flash with the following command:

cp.linux XXXXXXXXX

Where XXXXXXXX is the HEX sysupgrade.bin file size you recorded in the previous step. In the example image the correct value to enter is 480004

Finally, reboot the board with:

```
reset
```

Your board should reboot and you should see the Linux kernel boot messages of your new system. If you get an error while loading the kernel power cycle the board and press ‘4’ to return to the U-Boot terminal to download a new firmware.

Revisions

1.0	10/5/2015	Initial revision.