

PICO COMPUTING INC

Embedded Linux USB Host Controller

Version 1.0.0.0

12/17/2008

Introduction:

This document will illustrate the use of Pico Computing's embedded Linux to control USB 2.0 full speed host using the E-14 or E-15 FPGA card and the E-14/15 USB NAND Standalone Board. There are 7 steps to this demo ending with the host controller mounting a USB Thumb Drive. Note: This will only work on the E-14 and E-15.

Needed Items:

- E-14 or E-15 FPGA Card
- E-14 /15 USB NAND Standalone Board
- Paddle board for E-14 or E-15 (provides the Ethernet and UART connectors)
- Embedded Linux
- USB Thumb Drive

Step 1: Download and Install Embedded Linux

Download linux installer from:

ftp://guest:guest@ftp.picocomputing.com/dropbox/jeremy/linux/linux_installer_12_17_2008.zip

EDK project for E-14 FX20:

ftp://guest:guest@ftp.picocomputing.com/dropbox/jeremy/linux/E14FX20_LL_Temac_RAM125_USB.zip

EDK project for E-14 FX60:

ftp://guest:guest@ftp.picocomputing.com/dropbox/jeremy/linux/E14FX60_LL_Temac_RAM125_USB.zip

EDK project for E-15 FX60:

ftp://guest:guest@ftp.picocomputing.com/dropbox/jeremy/linux/E15FX60_LL_TEMAC_RAM175_USB.zip

Create a folder and unzip the files, and inside that directory you will find "build.bat".

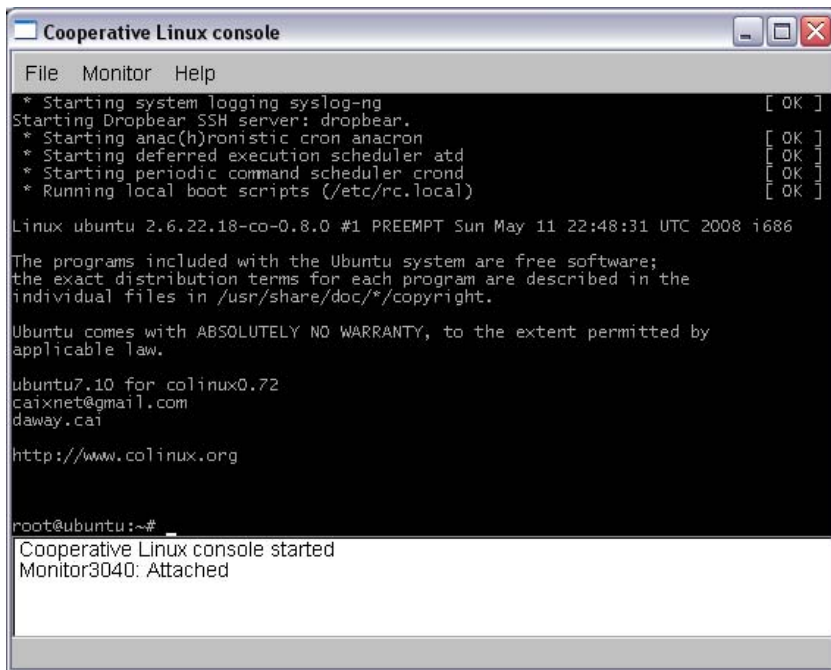
Run "build.bat" - You can run this by starting a command prompt or by double clicking it in windows.

This will start off the installer which will copy the source files to C:\Pico\colinux, and also create an image space for installing the Linux. Then the CoLinux will start and unzip the source files into the image space. This will take a few minutes.

Step 2: Creating the Linux Executable

Once this is finished, go ahead and start CoLinux by running "startColinux.cmd" (You can run this by starting a command prompt or by double clicking it in windows), which is in c:\pico\colinux.

When co-linux boots you should see this:



```
Cooperative Linux console
File Monitor Help
* Starting system logging syslog-ng [ OK ]
Starting Dropbear SSH server: dropbear.
* Starting anac(h)ronistic cron anacron [ OK ]
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Running local boot scripts (/etc/rc.local) [ OK ]

Linux ubuntu 2.6.22.18-co-0.8.0 #1 PREEMPT Sun May 11 22:48:31 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu7.10 for colinux0.72
caixnet@gmail.com
daway.cai

http://www.colinux.org

root@ubuntu:~#
Cooperative Linux console started
Monitor3040: Attached
```

From here go ahead and type: “cd /usr/src/linux-2.6.pico”. This will take you to the directory to build the Linux executable.

The command “pmk” is used for building our Linux executable. Typing: “pmk help | less” will display the different options for building the Linux kernel (typing “!” and enter returns to the consul).

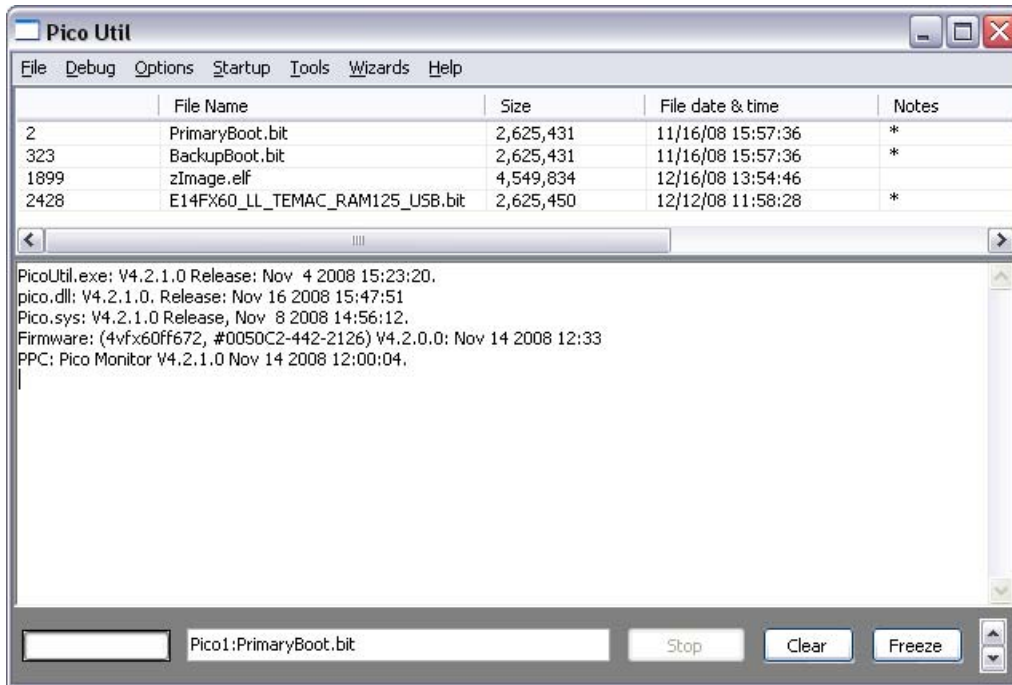
For this particular demo we want to build the Linux with USB support, type: “pmk maxconfig”. This will create a .config file with all the settings for the Linux kernel.

Now type “pmk”, this will start the process to build the Linux executable (this will take several minutes). When the Linux executable is created, there should be a “zImage.elf” in the C:\pico\colinux directory.

To boot Linux on the card we need to load the Linux executable onto the flash.

Step 3: Writing files to FLASH

Place the E-14 or E-15 card in the slot, and open Pico Utility:



There are two files that need to be on the flash for this demo:

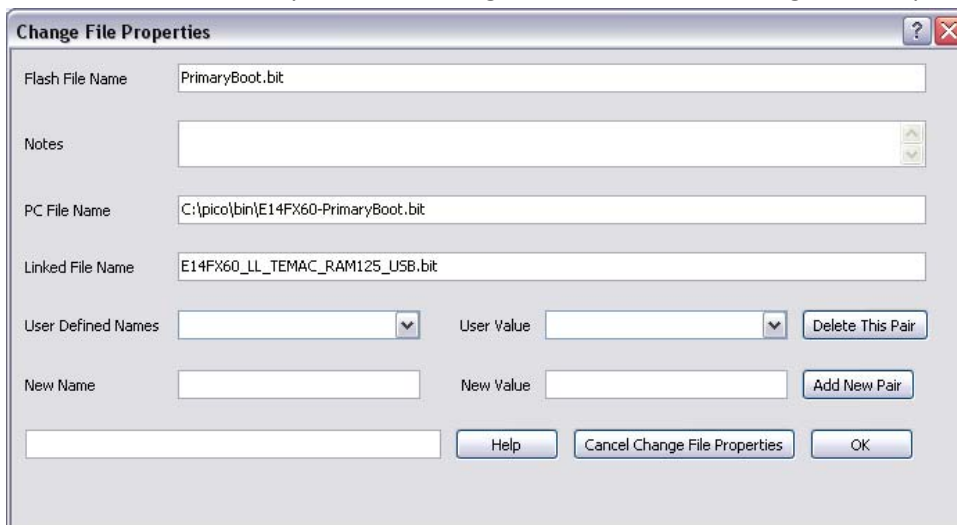
1. Bit image from the E-14 or E-15 EDK project which is included in the zip file under the implementation directory.
2. zImage.elf.

To load these files onto the flash go to file and select “Write File to Flash ROM from PC File” and point to the corresponding directory.

Step 4: Linking bit and elf files

The next step is to link images together; this way when the card boots it will automatically load the bit image and boot up Linux.

To do this, select PrimaryBoot.bit then right click and select “Change File Properties”:



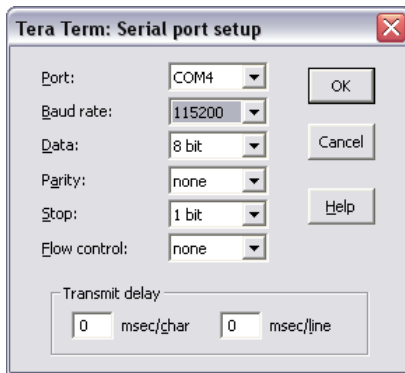
In the Linked File Name box type the name of the bit image that was taken from the EDK project for the E-14 or E-15. Then select the linked bit image, and again right click and select “Change File Properties”. This time in the Linked File Name put zImage.elf.

Step 5: Setting up and Running Linux

Place the card with the linked images into the E-14/15 USB NAND Standalone Board. Next connect the paddle board, and attach a RS-232 cable and an Ethernet cable (optional, but required for Step 6). Place a USB thumb drive into the port labeled Host 1 on the standalone board (at this time we do not have support for Host #2). Linux will auto negotiate (10/100/1000 as well as support DHCP).

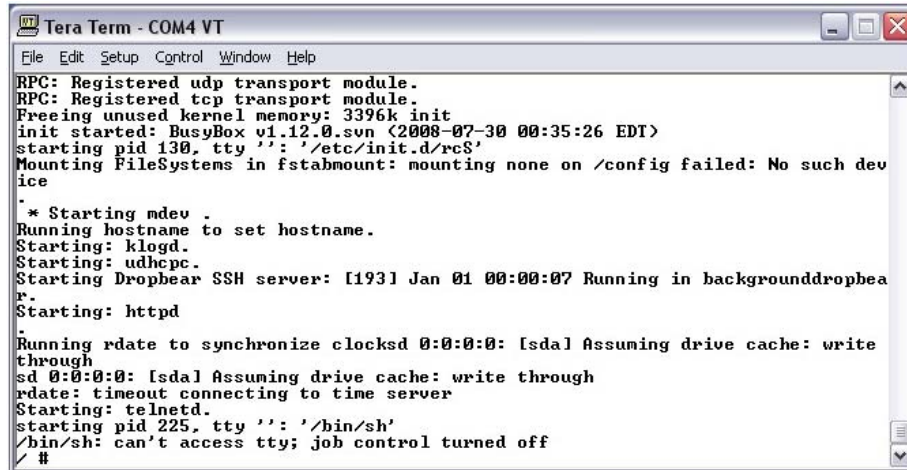
Go ahead and start any Serial Terminal Program, such as HyperTerminal or Tera Term.

Here are the settings for RS-232 Terminal:



When power is applied to the standalone board, what will happen is:

1. “primaryboot.bit” will be loaded.
2. The card will detect if it is in standalone mode and then boot the linked bit image(s).
3. Once the linked bit image is loaded, then Linux will be copied from the flash into the RAM.
4. Then the PowerPC will unzip and execute Linux from the RAM.
5. Text should start appearing on the RS-232 terminal. Once Linux has booted you should see this:

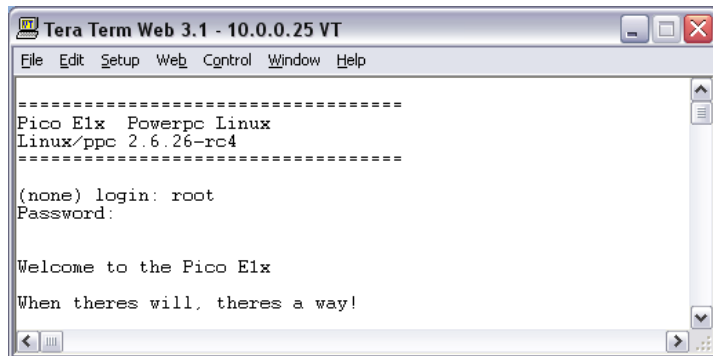


Step 6: Interface to the card over Ethernet using Telnet.

1. At the command prompt type “ifconfig eth0”. If your network has a dhcp server, then the card should get an IP address, and when you type this command you should see “inet addr XX.XX.XX.XX”.

2. If you don't have a dhcp server type "ifconfig eth0 10.0.0.80", and replace that ip address with the one you want. This should assign an ip address to the card.
3. To connect to the card over Telnet, start Tera Term or HyperTerminal and select TCP/IP connection. The card does support Telnet or SSH and the port is 23, and then enter the ip address. The login is root and the password is root.

Once you have logged in you should see this:

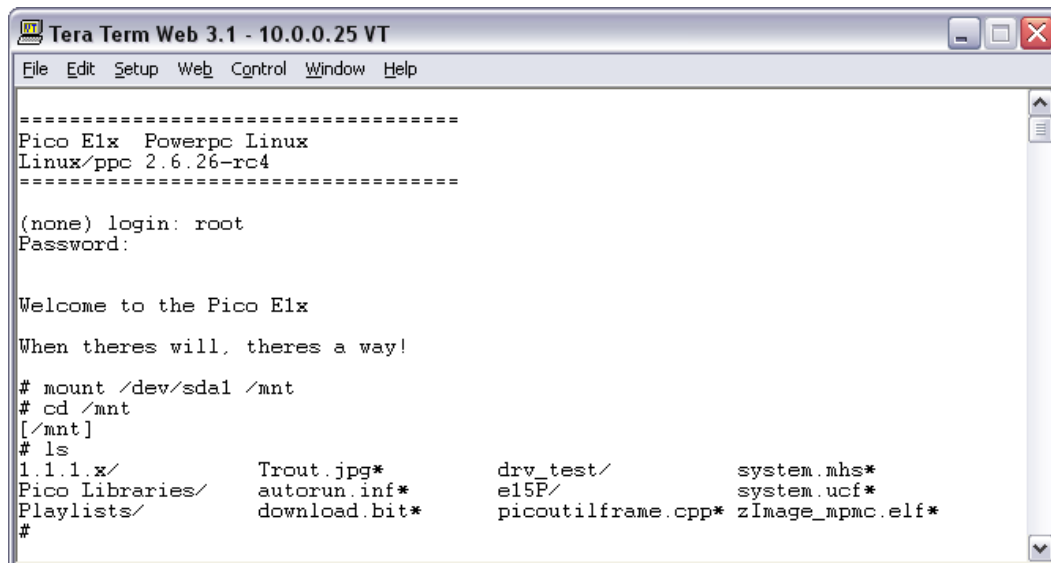


```
Tera Term Web 3.1 - 10.0.0.25 VT
File Edit Setup Web Control Window Help
=====
Pico Elx Powerpc Linux
Linux/ppc 2.6.26-rc4
=====
(none) login: root
Password:

Welcome to the Pico Elx
When theres will, theres a way!
```

Step 7: Mount USB Thumb Drive

To mount the USB drive type "mount /dev/sda1 /mnt" where mnt can be replaced by whatever name you want. This should mount the USB drive. Once this is done, type "cd /mnt" and then type "ls". You should now be reading the contents of the USB flash drive. Here is an example of this:



```
Tera Term Web 3.1 - 10.0.0.25 VT
File Edit Setup Web Control Window Help
=====
Pico Elx Powerpc Linux
Linux/ppc 2.6.26-rc4
=====
(none) login: root
Password:

Welcome to the Pico Elx
When theres will, theres a way!

# mount /dev/sda1 /mnt
# cd /mnt
[/mnt]
# ls
1 1.1.x/          Trout.jpg*      drv_test/      system.mhs*
Pico Libraries/ autorun.inf*   e15P/          system.ucf*
Playlists/       download.bit*  picoutilframe.cpp* zImage_mpmc.elf*
#
```

You can also type "dmesg", and see this:

```
Tera Term Web 3.1 - 10.0.0.25 VT
File Edit Setup Web Control Window Help
scsi 0:0:0:0: Direct-Access Memorex Flashdrive 303B PMAP PQ: 0 ANSI: 0 CCS
sd 0:0:0:0: [sda] 251776 512-byte hardware sectors (129 MB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 23 00 00 00
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] 251776 512-byte hardware sectors (129 MB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 23 00 00 00
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
usb-storage: device scan complete
c67x00 c67x00.0: ClearPortFeature (0): C_RESET
usb 1-1: reset full speed USB device using c67x00 and address 2
c67x00 c67x00.0: ClearPortFeature (0): C_RESET
sd 0:0:0:0: [sda] Result: hostbyte=0x07 driverbyte=0x00
end_request: I/O error, dev sda, sector 113672
c67x00 c67x00.0: ClearPortFeature (0): C_RESET
usb 1-1: reset full speed USB device using c67x00 and address 2
c67x00 c67x00.0: ClearPortFeature (0): C_RESET
sd 0:0:0:0: [sda] Result: hostbyte=0x07 driverbyte=0x00
end_request: I/O error, dev sda, sector 216520
#
```

Here you can see information about the USB device. Here I'm using a Memorex Flash drive 129 MB.