

## PiBox - Action Item #172

### PiBox component rev: xcc

24 Feb 2013 15:44 - Hammel

<b>Status:</b>	Closed	<b>Start date:</b>	24 Feb 2013
<b>Priority:</b>	Immediate	<b>Due date:</b>	
<b>Assignee:</b>	Hammel	<b>% Done:</b>	100%
<b>Category:</b>	01 - Cross Toolchain	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	0.8.0		
<b>Severity:</b>	01 - Critical		
<b>Description</b>			
PiBox current: 1.15.2 Crosstool-NG mainline: 1.18.0  It's unclear what issues this might bring, but we need the rev to pick up latest upstream Linaro support.			

#### Associated revisions

##### Revision 60d05118 - 08 Mar 2014 16:26 - Hammel

RM #172: For ctng 1.19.0 - Bump kernel to 3.10.2, Bump GCC to 4.7.3 (non-Linaro).

##### Revision c68f2b2c - 09 Mar 2014 12:31 - Hammel

RM #172: Make Crosstool-NG v1.19.0 the default.

#### History

##### #1 - 21 Mar 2013 21:14 - Hammel

- Status changed from New to In Progress
- % Done changed from 0 to 50

Added configuration and build support for Crosstool-NG 1.18.0. Tested build and package installation with latest kernel (3.6.y). Builds complete successfully. Build artifacts are yet to be tested on hardware.

##### #2 - 28 Mar 2013 19:05 - Hammel

- Priority changed from Normal to Low
- Severity changed from 03 - Medium to 04 - Low

1.18.0 generates a toolchain that can compile the kernel that boots but for some reason the kernel doesn't find the root file system. This is with the current 3.2.27 kernel and Buildroot 2013.02 that works with the 1.15.2 toolchain.

1.18.0 doesn't appear to offer any additional Linaro updates so for now I'm sticking with 1.15.2. I may try 1.16.x later.

Moving this task to lowest priority so I can get to making a public release with what is working.

##### #3 - 22 Aug 2013 10:13 - Hammel

1.15.2 builds on Fedora 16 and CentOS 6.2/6.4. It does not build on Fedora 19. The problem with the latter is that PPL fails with this error:

```
[ERROR] configure: error: Cannot find GMP version 4.1.3 or higher.
```

However, GMP 5.0.2 is built by Crosstool-NG right before PPL is built, so for some reason on F19 PPL doesn't see the just built GMP. What's worse, Fedora 19 comes with 5.1.1. If I installed the gmp-devel package the build gets past the missing GMP version but still fails in PPL due to some header incompatibilities.

So it looks like the PPL build in 1.15.2 is relying on the build hosts GMP installation and PPL 0.11.2 is not compatible with GMP 5.1.1. I'll have to try 1.15.3 or later.

#### **#4 - 22 Aug 2013 11:12 - Hammel**

Tried

- 1.15.3
- 1.16.0
- 1.17.0
- 1.18.0

on Fedora 19. All failed with messages similar to:

```
[ERROR] /home/mjhammel/src/ximba/raspberrypi/src/./bld/crosstool-ng-1.18.0.bld/work/src/ppl-0.11.2/src/mp_std_bits.defs.hh:109:7: error:
redefinition of 'class std::numeric_limits<__gmp_expr<__mpq_struct [1], __mpq_struct [1]> >'
[ERROR] /usr/include/gmpxx.h:3306:21: error: previous definition of 'class std::numeric_limits<__gmp_expr<__mpq_struct [1], __mpq_struct
[1]> >'
```

It looks like gmp 5.1.1 on the build host is not compatible with PPL 0.11.2, which is the most recent version available through Crosstools-NG.

Seems to me the bug is that Crosstool-NG builds ppl without specifying the just-built GMP as an override to the build host's GMP.

#### **#5 - 17 Oct 2013 14:32 - Hammel**

- % Done changed from 50 to 60

Crosstool-NG 1.19.0 has been tested on Fedora 19 and appears to work without error. New features provided by this release have not been reviewed for their suitability to improving PiBox. That will happen after the PiBox 0.6.0 release.

This version is being tested on CentOS 6.4 before becoming the new default version of Crosstool-NG.

#### **#6 - 18 Oct 2013 09:16 - Hammel**

- % Done changed from 60 to 70

CentOS 6.4 build completed successfully.

New images need to be tested on hardware.

#### **#7 - 02 Mar 2014 14:59 - Hammel**

- Priority changed from Low to Immediate

- Severity changed from 04 - Low to 01 - Critical

Kernel 3.10.y (3.10.31+) and Buildroot 2013.11 are working with the standard 1.15.2 toolchain. Now I need to try to build them with the 1.19.0 toolchain. I'd tried this before rev'ing the kernel and rootfs but I think I ran into problems. Unfortunately I didn't note what those problems were. So now I'm rebuilding on CentOS:

1. 1.19.0 toolchain
2. 3.10.y kernel
3. 2013.11 rootfs

If this boots and behaves generally correctly with the opkgs then I can make it the default.

#### #8 - 02 Mar 2014 14:59 - Hammel

- Target version changed from 1.0 - Atreides to 0.8.0

Retargeting for 0.8.

#### #9 - 03 Mar 2014 08:13 - Hammel

Rebuilt everything with the 1.19.0 toolchain but I get the same problem as with the 1.18.0 - "no init found".

- The kernel boots fine and the rootfs is mounted as ext3 by the kernel.
- The cmdline.txt does not contain an init= entry.
- Busybox is there and symlinked to /sbin/init.

```
$ file bin/busybox
```

```
bin/busybox: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 3.7.3, stripped
```

```
$ ls -l sbin/init
```

```
lrwxrwxrwx 1 root root 14 Mar  2 22:09 sbin/init -> ../bin/busybox
```

So I'm not sure what is broken with this toolchain. But I can be sure that it's the way the toolchain is built that causes the problem since the kernel and rootfs (and cmdline.txt) all work with 1.15.2.

One other thing: the 1.15.2 build does have support for 4.7 gcc, both GNU and Linaro versions. I can try those to get around this issue. But it would be nicer to get the latest Crosstool-NG working.

#### #10 - 03 Mar 2014 09:45 - Hammel

1.19.0 was configured to use kernel 3.10.2. I changed this back to 3.2.48, which is the version of the 3.2.x branch supported by 1.19.0.

Rebuilding xcc, after which I'll rebuild everything else and try again.

#### #11 - 06 Mar 2014 20:50 - Hammel

I built a minimal, statically linked Busybox using the 1.19.0 toolchain and it booted. This tells me there is something wrong with how the Buildroot rootfs is put together. I'm thinking it has something to do with where it finds ld.so or libc or something.

I may be able to verify this by looking at the package/customize/source directory, which is where I place the sysroot files. However, I rev'ed everything (OS, gcc, libc, etc.) in 1.19.0 and started to build everything again. So I have not buildroot tree to examine right now. I'll check it again after I do the full build.

This next build will be tested with the minimal Busybox first. Then I'll review the Buildroot build.

## #12 - 07 Mar 2014 08:58 - Hammel

1.19.0 with latest compiler/libc/OS configurations worked with a statically built Busybox. So it's back to "why doesn't the kernel see the rootfs"?

I looked at 1.19.0's sysroot and compared it to the Buildroot target build on a different machine. I noticed in the later there is this symlink under /lib:

```
$ ls -l ld*
-rwxr-xr-x 1 mjhammel mjhammel 162248 Feb  1 16:56 ld-2.13.so
lrwxrwxrwx 1 mjhammel mjhammel   10 Feb  1 16:56 ld-linux.so.3 -> ld-2.13.so
```

But under 1.19.0's sysroot it looks like this:

```
$ ls -l ld*
-rwxr-xr-x 1 mjhammel mjhammel 172016 Mar  6 21:07 ld-2.17.so
lrwxrwxrwx 1 mjhammel mjhammel   10 Mar  6 21:07 ld-linux-armhf.so.3 -> ld-2.17.so
```

So I'm wondering if the missing ld-linux.so.3 symlink is the problem. I found a discussion on using [ld-linux.so.2](#) so I ran the following on the busybox binary in the Buildroot for a 1.15.2-based rootfs:

```
$ readelf -l busybox

Elf file type is EXEC (Executable file)
Entry point 0x11ae8
There are 8 program headers, starting at offset 52

Program Headers:
Type           Offset  VirtAddr  PhysAddr  FileSiz MemSiz  Flg Align
EXIDX          0x0ad834 0x000b5834 0x000b5834 0x000c0 0x000c0 R  0x4
PHDR           0x000034 0x00008034 0x00008034 0x00100 0x00100 R E 0x4
INTERP        0x000134 0x00008134 0x00008134 0x00013 0x00013 R  0x1
              [Requesting program interpreter: /lib/ld-linux.so.3]
```

There was other stuff printed, but the important part is the reference to the program interpreter. That's the symlink in the working Buildroot. Interestingly I've hit this before because there is a comment in xcc.mk about a missing ld-linux.so but there is no code for generating that link.

What I want to do is run readelf on the busybox binary for the 1.19.0 build and see what that says. If it points to a non-existent program interpreter then I think I have my culprit.

I will build Buildroot and test it as is, expecting it to fail. As soon as Busybox is built I'll pass it through readelf to check that. Assuming the link is missing, I'll manually add that symlink to the SD card and see what happens. If that works, then the xcc build has to be adjusted for 1.19.0 without breaking 1.15.2.

### #13 - 07 Mar 2014 09:48 - Hammel

So this sucks:

```
$ readelf -l bin/busybox
```

Elf file type is EXEC (Executable file)

Entry point 0x12128

There are 8 program headers, starting at offset 52

Program Headers:

Type	Offset	VirtAddr	PhysAddr	FileSiz	MemSiz	Flg	Align
EXIDX	0x0ab9f8	0x000b39f8	0x000b39f8	0x00008	0x00008	R	0x4
PHDR	0x000034	0x00008034	0x00008034	0x00100	0x00100	R E	0x4
INTERP	0x000134	0x00008134	0x00008134	0x00019	0x00019	R	0x1

[Requesting program interpreter: /lib/ld-linux-armhf.so.3]

Which means that busybox is referencing the correct symlink. So if this build doesn't boot then I have some other problem. Ugh. Unless, after performing the sysroot install into the target rootfs the symlink disappears. Will have to check that when the build completes (it's still running). I don't think that's the issue since the customize tree is created at the start of the Buildroot build (and that copies sysroot plus other stuff into a package directory in Buildroot before running the build) and that tree has a valid symlink. Plus the copy from there to the target is done with a post-build script that uses the same copy mechanism (rsync). So I'm guessing the symlink will be there.

Will have to wait and see if it boots or not.

### #14 - 09 Mar 2014 12:38 - Hammel

- Status changed from *In Progress* to *Closed*

- % Done changed from 70 to 100

Rev'd Crosstool-NG to v1.19.0 as the default. This was tested with:

1. Linux 3.10.2
2. binutils 2.21.1a
3. gcc 4.7.3 (non-linaro)
4. glibc 2.13

This properly builds the RPi kernel 3.10.y (currently 3.10.31+ for PiBox) and Buildroot 2013.11 without webkit and friends.

The build is stable. Networking works fine with the rtl8187 driver (probably doesn't work well with the rt2800usb which seems to be broken).

Changes pushed upstream. This issue can be closed until we need the next xcc rev, which I don't expect for at least a year.