Porting Nemo Mobile and Mer Project to new Hardware

Marko Saukko
sage@merproject.org
Twitter: @markosaukko

Twitter hashtags: #nemomobile and #merproject
Outline

- Generic notes
- Mer Platform SDK
- Adaptation points
  - Kernel, Graphics, Xorg, Sensors, Audio, usb, etc.
- How to build image for your adaptation
  - Packaging, patterns, kickstarts, image building..
After this presentation listener should know what are the adaptation points needed and where to start the new adaptation for Mer and Nemo.

As Nemo is build on top of Mer, Mer dependencies/requirements also apply to Nemo.

Mer and Nemo contain as few hw adaptation bits in the main repos as possible.

- Exception, e.g., mesa llvmpipe
- Report problems to bugzilla
  - [https://bugs.merproject.org/](https://bugs.merproject.org/)
  - [https://bugs.nemomobile.org](https://bugs.nemomobile.org)
When building hardware adaptation one should not need to (re)compile any of the components that are located in Mer or Nemo repositories.

Usually normal Nemo image contains around 500 packages from which 5-20 are from the hardware adaptation.

This presentation does not go through all of the details of certain components, but aims to give the general overview of points where adaptation is needed.
Mer Platform SDK

- NOTE: Platform SDK, not Application SDK
- All the tools that you need
  - Scraotchbox 2, mic, osc, mer-kickstarter, ...
- Works on any typical Linux distribution, with kernel >= 2.6.37
- Very easy to install see tl;dr; guide in wiki
- https://wiki.merproject.org/wiki/Platform_SDK
Mer Hardware Requirements

- There are no real minimum requirements
- Can work on pretty much anything, e.g.,
  - N900, N950, N9
  - Nexus 7
  - Pandaboard
  - Snowball
  - Raspberry Pi
  - ExoPC, Lenovo Ideapad S10-3t, ...
<table>
<thead>
<tr>
<th>Mer port name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>i486</td>
<td>Generic i486+ X86 port</td>
</tr>
<tr>
<td>i586</td>
<td>SSSE3 enabled X86 port</td>
</tr>
<tr>
<td>armv6l</td>
<td>ARMv6 + VFP port</td>
</tr>
<tr>
<td>armv7l</td>
<td>ARMv7 VFPv3-D16 port, softfp ABI</td>
</tr>
<tr>
<td>armv7hl</td>
<td>ARMv7 VFPv3-D16 port, hardfp ABI</td>
</tr>
<tr>
<td>armv7tnhl</td>
<td>ARMv7 VFPv3-D16 port, hardfp ABI, NEON, Thumb2</td>
</tr>
<tr>
<td>mipsel</td>
<td>ARMv7 VFPv3-D16 port, hardfp ABI, NEON, Thumb2</td>
</tr>
</tbody>
</table>

https://wiki.merproject.org/wiki/OBS_architecture_naming
Almost every hw has its own bootloader and the way it is used

- U-boot most common one
- Usually needs kernel cmdline modifications, e.g.,
  - New paths for the location of the Mer rootfs
  - Removing initrd loading if not used with Mer
Mer: Kernel

- Minimum kernel version 2.6.32
  - 2.6.32 needs some patches for cgroups
  - 2.6.37 should have everything needed

- Check the required kernel config options with mer-kernel-checks
  - https://github.com/lbt/mer-kernel-check/
  - mer_verify_kernel_config < .config
    - Checking for the modules that are needed by the different systems that are available in Mer
    - Makes difference between optional and mandatory values

- Example packaging available at https://github.com/lbt/mer-kernel-adaptation
Mer: Graphics Adaptation

- OpenGLES 2.0 recommended (for Nemo Mobile required)
- Needs to provide certain files, such as, libEGL.so.1, libGLES_CM.so.1 and libGLESv2.so.2
- If not provides mesa-llvmpipe packages will be quite easily dragged in instead
- Template packaging available at https://github.com/saukko/mer-graphics-adaptation
Mer: Xorg (1/2)

- Add required configuration to /etc/X11/xorg.conf.d/*.conf or /usr/share/X11/xorg.conf.d/*.conf
  - For example graphics driver and touch usually requires a configuration to be added
- Start with standard fbdev driver, i.e., xorg-x11-drivr-fbdev (in Mer Core)
- Multitouch with xorg-x11-drivr-mtev (in Mer Core)
• Example of mtev config

Section "InputClass"
  Identifier                   "eGalax Touchscreen"
  MatchVendor             "eGalax"
  MatchDevicePath      "/dev/input/event*"
  Driver                        "mtev"
  Option                       "Ignore"                "off"
EndSection
Zephyr is an attempt at creating a stack for use by other projects to be exploring lightweight, high-performance, next-generation UIs based on Mer, Qt5, QML compositor and Wayland.

https://wiki.merproject.org/wiki/Zephyr
Mer: Sensors (1/2)

- Sensorfw – handles rotation with accelerometer, magnetometer etc.
  - Adaptation needs config file and also in some cases new plugin to be written
  - Config files located at /etc/sensorfw/*.conf

- Currently plugins/configs are in Mer core, but those are going away. See: MER#557 and MER#558

- Example code for sysfs adaptor:
### Mer: Sensors (2/2)

**N9: /etc/sensorfw/sensord-rm_696.conf**

```ini
[accelerometer]
dataranges = "-2048=>2048"
intervals =
"0,10,20,25,40,50,100,200,250,500,1000"
transformation_matrix = "-1,0,0,0,-1,0,0,0,-1"

[als]
driver_type = 2
path = /dev/apds990x0
dataranges = "0=>65535"
intervals = 0

[keyboardslider]
input_match = gpio-keys
dataranges = "0=>2"
intervals = 0

[proximity]
driver_type = 2
path = /dev/apds990x0
dataranges = "0=>1"
intervals = 0

[magnetometer]
path = /dev/ak89750
dataranges = "-4096=>4096"
intervals = "25,50,100,200,250,500,1000"
default_interval = 1000
interval_compensation = 16
scale_coefficient = 300
calibration_rate = 100
calibration_timeout = 60000

[tap]
input_match = accelerometer
dataranges = "0=>2"
intervals = 0

[orientation]
threshold_landscape = 25
threshold_portrait = 20
overflow_min = 800
overflow_max = 1250
discard_time = 750000

[context]
orientation_poll_interval = 1000
orientation_offset = 3
stability_timeout = 60
```
Mer: ofono

• Cellular adaptation
  – At times udev rule needed
  – For new modems a plugin for ofono
  – Currently supported modems listed at http://git.kernel.org/?p=network/ofono/ofono.git;a=blob;f=doc/hardware-support.txt
• https://ofono.org/
Mer: Audio

- Good ALSA driver from pulseaudio point of view
- Pulseaudio configuration
  - New port configuration for PulseAudio, which define different routing possibilities; lineout, ihf, etc.
- Phone functionality
  - If call audio is routed through PulseAudio, new PulseAudio module for data transfer with modem
  - If call audio is separate from PulseAudio, volume handling scheme needs to be adapted for new audio route
- Package: `pulseaudio-settings-<NAME>`
- Example: [https://github.com/nemomobile/pulseaudio-settings-n950-n9](https://github.com/nemomobile/pulseaudio-settings-n950-n9)
Nemo: Resource Policy

- Makes routing decisions and after decision pulseaudio changes the audio route
- At minimum new configurations for all enforcement points; PulseAudio, X
  - Map PulseAudio enforcement configuration against ports
  - Create a `resource-policy-<NAME>` package that contains the configs, example
- Write Prolog rules for new/changed use cases
- Binary called ohmd
  - What to do when multiple audio sources are played?
  - What to do when phone rings?
  - What to do when accessories are attached?
- [https://github.com/nemomobile/policy-settings-basic](https://github.com/nemomobile/policy-settings-basic)
Mer: gstreamer

- Mer/Nemo contains only some very basic things required by other components
- Different hw support different ways of decoding/encoding the media and also most of these codecs/codes are patented/proprietary. Thus codecs are part of the adaptation.
- In Mer
  - gstreamer
  - gst-plugins-base
  - gst-plugins-bad-free
- In Nemo
  - gst-plugins-good
Nemo: MCE

- Responsible for
  - Screen blanking/unblanking with powerkey
  - Turning display backlight off/on to save battery
  - Turn display/touch off when talking on phone (proximity sensor)
  - Display dimming in dark room (ambient light sensor)
  - Suspending the device when possible
- By default should handle most of the things automatically
- Config files can be added when specific conditions needed
  - /etc/mce/*.ini
  - NEMO#616: Some legacy config leftovers in packaging
Nemo: DSME

• Responsible for
  – Watchdog kicking
  – Alarms – wakeup the device when alarm goes off
  – Temperature – shutdown if temperature too high
  – Battery level – shutdown if battery getting too low
  – Shutdown/reboot policy based on various inputs
    • if alarms in near future go to actdead mode instead of shutdown

• NOTE: DSME does not have proper configuration handling yet.
Nemo: usb-moded (1/2)

- Not needed, if no usb peripheral port
- Can use following kernel usb gadget modules: g_nokia, g_file_storage, g_mass_storage, g_ether, g_ncm and g_ffs
- Provides:
  - usb networking, also called SDK/developer mode
  - mass-storage mode – has requirements for filesystem partitioning
  - MTP (Media Transfer Protocol)
- Optional settings file at /etc/usb-moded/usb-moded.ini
  - Change default mode (usb networking)
  - Change usb networking ip
[mountpoints]
mount = /dev/mmcblk0p1

[sync]
nofua = 1

[altmount]
mount = /home/user/MyDocs

[usbmode]
mode = developer_mode

[network]
Ip = 192.168.3.15
Gateway = 192.168.3.14
Nemo: nemo-user-session

- Responsible of starting up the services for user session, e.g.,
  - Xorg, maliit, pulseaudio, lipstick, ngfd, mcompositor, ...

- Environment values in:
  - `/etc/sysconfig/nemo-mobile-ui`
  - `/etc/sysconfig/nemo-mobile-hw`
  - `/var/lib/environment/nemo/* .conf`
Packaging, patterns, kickstarts, image building, ...
Where to put all of these things?

• Create a `<adaptation-x>-configs` package and put config files needed for your adaptation there
  – Pulseaudio and policy configurations usually in separate packages
• Example: nokia-n950-configs, nemo-configs-x86-generic, ...
Patterns

● One thing to mention before building an image is patterns
● Collection of packages or other patterns to be installed easily
● For Nemo the most important to remember is "Nemo Complete" pattern
● Makes it easy to update existing images
  zypper install -f -t pattern nemo-complete
Creating a kickstart (.ks) file

- mer-kickstarter
  - Tool to handle multiple different adaptation configurations without need to copy-paste configuration lines
- nemo-kickstarter-configs
  - Nemo configs is a good base to start with
  - https://github.com/nemomobile/nemo-kickstarter-configs/
Example .ks file

```
lang en_US.UTF-8
keyboard us
timezone --utc UTC
part / --size 1500 --ondisk sda --fstype=ext3
rootpw nemo

user --name nemo --groups audio,video --password nemo

repo --name=mer-core --baseurl=http://releases.merproject.org/releases/latest/builds/armv7hl/packages --save --debuginfo
repo --name=ce-mw-shared --baseurl=http://repo.pub.meego.com/CE:/MW:/Shared/Mer_Core_armv7hl/ --save
repo --name=ce-apps --baseurl=http://repo.pub.meego.com/CE:/Apps/CE_MW_Shared_armv7hl/ --save
repo --name=ce-ux-mtf --baseurl=http://repo.pub.meego.com/CE:/UX:/MTF/CE_MW_MTF_armv7hl/ --save
repo --name=your-adaptation-repo --baseurl=http://url.to.your.adaptation.com/ --save

%packages
@Nemo Complete

# Add the adaptation packages
kernel-adaptation-x
graphics-adaptation-x
nemo-configs-x
%end
%post
...
%end

%post --nochroot
...
%end
```
Nemo: oneshot

- When you think about adding script to `%post` section of `.ks` file, better thing would be oneshot script
  - `.ks` file `%post` section only applies to new image creation not to image updates
- Oneshot scripts that are executed only once on next boot process
- Example of usage
  https://github.com/nemomobile/nemo-firstsession
Creating the image

- **mic**
  - Tool to create an image based on the kickstart files (.ks)
    
    ```
    mic cr raw example.ks --arch=armv7hl --pkgmgr=zyp
    ```

- **Image installation is device specific**
Thank you.