



March 2022 - work in progress. needs to cover all supported HWM's. Will be split into separate documents.

Document name	development_refguide_di
Document purpose	Reference Guide for Hardware Model Direct Integration
Version	1.00, April 2022
Author	Stuart Winter <mozes@slackware>

Brain Dump

[Brain Dump onto LQ](#) - needs sorting into individual pages

[Brain Dump on LQ 2.0](#) - Kernel module loader example

Help / Support

Please post questions to the [Slackware ARM forum](#).

Reference Guide for Hardware Model Direct Integration into Slackware AArch64

This reference guide is to enable Hardware Custodians to determine what components of Slackware to consider and modify, in order to add and maintain Hardware Model support.

By way of a plugin system, this author has made extensive provision for the idiosyncrasies found in the range of Hardware Models found within the ARM ecosystem. The plugin system provides hooks into key stages of building the Slackware OS and its components (bootware, kernel etc.) and at OS boot and run time. Custom scripts have the ability to inject dynamically-created code and provide signaling into later parts of the execution flow. Whilst not every eventuality has been catered for, in general there should be enough within the environment and control structure to onboard most Hardware Models.

This guide is intended to act more of an index because each component (e.g. script, configuration file) has extensive inline annotation providing its purpose and context and a description of most logic, along with rationale where this author deemed appropriate.

This document is aimed at advanced Slackware and Linux users.

Slackware ARM Development Kit

Using any of the build system referenced here requires the [Slackware ARM development kit](#) to be installed and configured natively on an installation of Slackware ARM.

The Slackware ARM Development Kit is contained within the 'd' package series and will be installed as part of a full Slackware installation.

Bootware

The boot loader employed on the RockPro64 and PineBook Pro is [U-Boot](#).

Secure Boot

Presently none of the bootware assets supplied by Slackware ARM are signed. However, the Slackware Linux project does have a suitable signing key that can be used when necessary for Directly Integrated Hardware Models (i.e. supported directly by Slackware ARM).

If the Hardware Model you're interested mandates Secure Boot, you'll need to figure this out for yourself.

Trusted Firmware-A (ARM Trusted Firmware)

The ARM Trusted Firmware is embedded within the supplied U-Boot loader, and aren't used by the end-user directly. However, some platforms require the Trusted Firmware to boot a Kernel.

The single build script is used to produce the trusted firmware assets for all supported Hardware Models.

Asset(s)	Description
ATF binaries	ATF for RK3399
Build script	Builds the Trusted Firmware for supported Hardware Models

U-Boot Boot Loader

Recovery / Initialisation SD Card image

U-Boot configured to boot from SPI Flash

U-Boot configured to boot from SD Card

Built and available but not used for either the Operating System Installation nor Operating System Proper. The image can be written to an SD Card as a header and booted, but U-Boot would fail to boot sometimes after changes to the /boot filesystem, so it was unsuitable as a solution.

Boot Media

* OS and Installer Bootware for SD cards

Slackware Installer

Note: these scripts are not yet available within the Slackware ARM source tree. They need to be extracted from the Installer image.

Script	Called from	Purpose	Hardware Models
/etc/rc.d/rc.installerconfig	init	Hardware Model Installer Runtime Configuration Adjustments (CPU fan control, installer font size etc.)	All
/usr/lib/setup/armedslack-clock-set	/etc/rc.d/rc.S	If wrong time, set the clock with Slackware Installer build date. This indicates no RTC.	All
/usr/lib/setup/armedslack-SeTpartitions	/usr/lib/setup/SeTpartitions	Automatic Handling of the OS /boot Partition	All

Slackware ARM Post Installation Setup Scripts

The Slackware ARM post installation setup scripts are contained within /usr/lib/setup/armedslack-postinstall-scripts and are all called from /usr/lib/setup/armedslack-run-postinstall-scripts

/usr/lib/setup/armedslack-run-postinstall-scripts is sourced/run from /usr/lib/setup/SeTconfig immediately prior to running the Slackware package setup scripts (named setup.*) within the newly installed OS' /var/log/setup/ directory.

These post installation setup scripts are to configure the Operating System, but the scripts are not part of any Slackware package (unlike the rest of the post installation scripts within Slackware). This is simplify the work required to port Slackware to another architecture - rather than editing numerous scripts within the Slackware Installer, we run one script which launches the others.

Script	Purpose	Hardware Models
bootloader-flash	Installing the Boot Loader onto SPI Flash	Not Raspberry Pi
setconsole	Configure Linux cmdline 'console=' setting	All
os-initrd-mgr	Configures the OS InitRD (Operating System Initial RAM Disk)	All
removeinstaller	Removes the Slackware Installer from the /boot partition	All

Script	Purpose	Hardware Models
hwm-os-configure	Helper launch script for Hardware Model-specific configuration (see below)	-

Operating System Configuration (per Hardware Model)

hwm-os-configure

This runs helper scripts found within the Slackware Installer. These scripts are named after the SoC they support, e.g. /usr/share/hwm-configure/platform/\$ARCH/helper.scr/rk3399

Examples of usage include:

- Configuring suspend/hibernate options within /etc/elogind/logind.conf and within /boot/extlinux/extlinux.conf

Operating System configuration assets

Some static assets are installed to the Operating System to provide sane/useful defaults for particular Hardware Models.

/usr/share/hwm-configure/platform/\$ARCH/assets/rk3399/snd/pinebookpro/asound.state

An ALSA config also exists for the rockpro64: /usr/share/hwm-configure/platform/\$ARCH/assets/rk3399/snd/rockpro64/asound.state

Disabling Filesystem Checks

If the Hardware Model does not have a battery-backed RTC (Real Time Clock), the Hardware Model will boot with a date that passed many years ago (in some cases the year will be 1970, which is the Epoch of UNIX). The affect of this is such that the filesystem checks will run every single time the system boots, because the last mount time will be significantly different from present date (assuming that the system time was corrected (usually via an NTP time sync) at boot).

Hardware Model	Status
RockPro64	Not required

Should the Hardware Model not have an RTC, there is a change to make within the Slackware Installer. /usr/lib/setup/armedslack-nofscheck

Firmware

RockPro64/Pinebook Pro: /lib/firmware/rockchip


Slackware Operating System

Packages

Slackware ARM needs to modify some packages and add new ones. Below follows a list of the packages that have been added or modified.

Package: a/kernel-firmware

Hardware Model	Rock Pro64 / Pinebook Pro
Asset	Src
platform/aarch64/rk3399/rk3399	Firmware packaging script helper
platform/aarch64/rk3399/get-firmware.sh	Script to retrieve firmware assets
platform/aarch64/rk3399/assets/*	Firmware assets
Hardware Model	Raspberry Pi 4
Asset	Src
platform/aarch64/bcm2711/bcm2711/	Firmware packaging script helper
platform/aarch64/bcm2711/get-firmware.sh	Script to retrieve firmware assets
platform/aarch64/bcm2711/assets/*	Firmware assets
Change Management Process	
Provide the origin of the patch set (one or more URLs - https, git, etc.) and they will be retrieved from there.	

Licences must be included

Serial Console System Login

This is a hard requirement

To enable a TTY that presents a OS login prompt, which enables system access via the Serial adapter. All Hardware Models must support this.

Package Series	Package	Asset changed	Src
a	sysvinit-scripts	/install/doinst.sh	Install script additions

The additions script has inline documentation.

Change Management Process

This is a simple script and should not be made more complex, because this code typically never changes, thus modularising it serves no utility. Please email a diff to the Slackware ARM devel mailing list and it will be applied

Linux Virtual Console (Local Console (HDMI video, USB Keyboard / Mouse))

Console Video Display

Set within the Kernel Module Loader. Variable name..

X11/Xorg Window System

Package: x/x11-skel Asset: doinst.sh

The x11-skel package's post-installation script detects the RockPro64 and Pinebook Pro at install time, and deletes /etc/X11/xorg.conf enabling Xorg to automatically configure the video display.


Other Hardware Models contain configuration within this doinst.sh script.

Slackware Operating System Post Boot Configuration

Package: a/sysvinit-scripts Assets: /etc/rc.d/rc.platform - load helper scripts from /etc/rc.d/rc.platform.d/<arch> These scripts perform such things as setting the LCD panel brightness, setting fan speeds, etc.

Used on the RockPro64, Pinebook Pro.

Kernel

 All Hardware Models are supported by a single generic Kernel.

Hardware Model	Package Series	Package	Asset	Src
All	k	kernel-source	configs/config-armv8	Slackware AArch64 generic Kernel configuration

Change Management Process


Email the new Kernel configuration to the Slackware ARM devel mailing list with a summary of what the changes were for.

Kernel Package Build Helpers

The Pinebook Pro (RK3399 SoC) requires a firmware blob to be installed within the OS InitRD (see further down) in order to light up one of the display ports. To facilitate this, a helper script is executed from the kernel.SlackBuild script when the OS InitRD is constructed. This helper extracts the relevant firmware blob from the Slackware a/kernel-firmware package.

Package Series	Package	Asset	Src
k	kernel-source	platform/aarch64/rk3399	rk3399 Kernel Package Build Helper

Kernel Patches



Generally once a set of patches are available for a Hardware Model, they will continue to apply to the particular Linux Kernel release that they were developed against. In this author's experience, unless the OS becomes unstable as a result of any of that patch set, that patch set can remain for many months without maintenance. It is not expected that Hardware Model Custodians are continually providing patches for the same Kernel major release. If it's an absolute requirement to continually update the patch set, then this Hardware Model is not a suitable candidate for Direct Integration support within Slackware ARM/AArch64 (although it certainly may become one at a later date once the support has matured).

Change Management Process


Provide the origin of the patch set (one or more URLs - https, git, etc.) and they will be retrieved from there.

Operating System Initial RAM Disk ('OS Initrd')

<http://ftp.arm.slackware.com/slackwarearm/slackwareaarch64-current/source/a/mkinitrd/README.initrd.arm> needed?

Firmware within the OS InitRD

The RockPro64 requires a small blob of firmware for one of the display drivers. This is included within /lib/firmware/rockchip



Only firmware that is 100% required at Kernel module load time is added to the OS InitRD in order to keep the size down and to reduce overlap between the Slackware a/kernel-firmware package. Firmware for a display driver is a must. Firmware for Bluetooth et al is not.

Kernel Module Loader



The Kernel Module Loader is a shared resource with the Slackware Installer. The assets are stored within the Kernel source directory.

The Kernel Module Loader is responsible for loading software drivers (in the form of Kernel Modules) for a variety of common hardware, plus hardware for specific Hardware Models. The goal of the Kernel Module Loader is to load just enough drivers to access the storage upon which the Operating System resides, at which point the boot process can transition to loading the Operating System, which provides a far greater array of hardware support than is required within the early stages of booting Linux and the Slackware OS.

Each Hardware Model typically requires some specific software drivers to light up the hardware. The RockPro64's examples act as templates for all Hardware Models.

Asset	Description
k/SlkOS-initrd-overlay/load_kernel_modules.scr/platform/aarch64/rk3399	Kernel Module Configuration script for RK3399 Hardware Models (including RockPro64, PineBook Pro)



For the settings, view the script directly (see the table above)

Change Management Process
Email the new configuration to the Slackware ARM devel mailing list, or provide a URL from which it may be obtained.

* Note about modprobe.d within installer and os-initrd-mgr conditionally installing the correct version depending on the Hardware Model. SOC_NAME in the Kernel module loader plugins needs to match the file name.

source/k/SlkOS-initrd-overlay/usr/share/hwm-configure/platform/aarch64/modprobe.d	Hardware Model/SoC modprobe configurations
---	--

These are populated by /load_kernel_modules at run time, depending on which Hardware Model is detected by the Kernel Module loader plugin scripts.



Presently the modprobe configuration for the Hardware Model is not copied or in sync with the OS' copy. If this becomes an issue in the future, it will be synchronised by init within the OS InitRD.

Hardware Management

Monitoring

Not yet implemented

[Fan control](#)

Installation Documentation

* board support documentation and YT video url * board-specific documentation - docs.slackware.com
* devel list - explanation. Not to develop Slackware, but for development changes for ARM/AArch64 specifically. If the ARM/AArch64 build scripts have bugs or could be adjusted to better support ARM (e.g. with a configuration option to improve performance), subscribe to the list if not already, and send the suggestion here. * management

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

https://docs.slackware.com/slackwarearm:development_refguide_di

Last update: **2024/01/21 12:28 (UTC)**

