L-1048e.Ax ARM SystemReady IR Integration Guide Head

Manuals & Documentation

Exported on 07/19/2023

Table of Contents

L-1048e.Ax ARM SystemReady IR Integration Guide	
Document Title	L-1048e.Ax ARM SystemReady IR Integration Guide
Article Number	L-1048e.Ax
Release Date	xx.xx.xxxx
Is Branch of	L-1048e.Ax ARM SystemReady IR Integration Guide Head

1 Introduction

ARM SystemReady[1] aims to add support for a wide variety of distributions, by using UEFI[2] as a standardized way of loading operating systems and implementing a certain feature set.

The *Universal Bootloader* U-Boot[3] adds an UEFI layer to support booting standard bootloaders, like GRUB[4], and standard operating system, e.g. Ubuntu[5] or OpenSUSE[6].

With the ARM SystemReady IR[7] standard, U-Boot also supports updating itself through the EFI layer using a secure *Capsule Update*, similar to traditional BIOS updates on X86 platforms.

1 https://www.arm.com/architecture/system-architectures/systemready-certification-program
2 https://uefi.org/
3 https://source.denx.de/u-boot/u-boot
4 https://www.gnu.org/software/grub/index.html
5 https://ubuntu.com/
6 https://www.opensuse.org/
7 https://www.arm.com/architecture/system-architectures/systemready-certification-program/ir

2 Target Setup

Flash U-Boot to eMMC

For the board to be compatible with ARM SystemReady, there needs to be a suitable bootloader being flashed to the eMMC. The first release supporting ARM SystemReady on the phyCORE-i.MX 8M Plus[8] can be downloaded from our website[9].

One way of flashing the bootloader to the eMMC, is to first flash the headless image to an SD-card and write the bootloader to the eMMC via ssh on the host.

First flash the headless-image to the SD-Card:

```
host$ sudo dd if=phytec-headless-image-phyboard-pollux-imx8mp-3.wic of=/dev/sdX
status=progress
```

Then boot the target from the SD card and send the image with the command `dd` combined with `ssh` through the network to the eMMC of your device:

```
host$ sudo dd if=imx-boot status=progress | ssh root@192.168.3.11 "dd of=/dev/mmcblk2
bs=1k seek=32 conv=fsync"
```

Note

The `seek` parameter may need a different value depending on the target SoC.

SoC	`seek` (offset)
i.MX8M Mini	33 kiB
i.MX8M Nano	32 kiB
i.MX8M Plus	32 kiB

For other ways to flash the bootloader to eMMC, see the chapter "Flash eMMC" in our phyCORE-i.MX 8M Plus BSP Manual[10].

8 https://www.phytec.de/produkte/system-on-modules/phycore-imx-8m-plus/
9 https://download.phytec.de/Software/Linux/BSP-Yocto-i.MX8MP/BSP-Yocto-NXP-i.MX8MP-hardknott-nightly-2023-07-12/
10 https://www.phytec.de/cdocuments/?doc=mwDRJw#L1017e-A5phyCOREi-MX8MPlusBSPManual-FlasheMMC

2.1 Setting `bootcmd`, When Booting for the First Time

Boot the target **from eMMC** with the previously flashed bootloader. Now set the default boot command:

```
u-boot=> env set bootcmd "run efi_bootcmd;"
u-boot=> env save
```

The command `efi_bootcmd` will try to load EFI systems from any attached USB mass storage, SD card or eMMC, in that order.

3 Running a Linux Distribution

For the distribution tests, we tested the following:

1. Fedora IoT 38
2. openSUSE Leap 15.4

To perform the tests:

1. Download the distros from their website:
   a. https://fedoraproject.org/iot/download/ (Fedora IoT 38 DVD ISO image)
   b. https://get.opensuse.org/leap/15.4/#download (openSUSE Leap 15.4 AARCH64 updated offline image)
2. Flash the `.iso` image onto a USB-Stick using, e.g. Balena Etcher or simply `dd` in a console on the host workstation.
3. Insert the USB Stick into the target board. Remember to insert an SD card into the target board to install the distro onto, before booting the U-Boot. Also, create a log file for the serial console from the target board for everything you do from here. This log will be needed later for the template. This can be done e.g. by using a serial console like Putty on the host workstation.
4. Boot into the U-Boot and run the `distro_bootcmd` command (or let it boot automatically if the `bootcmd` has been modified to run the `distro_bootcmd` command as described above). It will then boot the installer image from the USB-Stick.
5. Select *Install Distro* in the GRUB menu and navigate through the installer dialog to install the distro onto the SD card.

   > ⚠ In *openSUSE Installation Dialogue,* there may be an error that firewalld could not be installed (on the last page before hitting install). This can be solved by going into the software menu and choosing "break firewalld by ignoring some of its dependencies".

6. When the distro is installed, reset or reboot the board, and remove the USB-Stick, before booting.
7. Run the `efi_bootcmd` command in U-Boot again to boot up the installed distro.
8. Now follow the steps in the "Test Linux Distribution installation" chapter from the ARM documentation here: https://developer.arm.com/documentation/DUI1101/1-1/Test-SystemReady-IR?lang=en

Ubuntu Server 22.04.2 LTS was also tried, but there is an error with the RTC (system time). The emulated RTC from U-Boot is also not working currently. The phyBOARD-Pollux[11] has an RTC, but it is optional and therefore not always present. Also, there is no driver for the RV3028 RTC in our U-Boot version 2021.04. There is a driver in the upstream U-Boot: https://source.denx.de/u-boot/u-boot at `drivers/rtc/rv3028.c` which may be used in later versions of our BSP. For now, adding support for the dedicated hardware RTC was not tested.

11 https://www.phytec.de/produkte/single-board-computer/phyboard-pollux/

4 Performing the Tests

The following instructions are provided as a reference on how we conducted the tests needed for ARM SystemReady certification. If you simply want to enable ARM SystemReady on your PHYTEC board and boot a regular Linux distribution, running these tests is not needed.

4.1 Prepare `ir_acs_life_image` on SD Card

1. Download the `ir_acs_life_image.img.xz` from https://github.com/ARM-software/arm-systemready/tree/main/IR/prebuilt_images/v21.09_1.0
2. Unpack the image with:

```
host$ xz -d ir_acs_life_image.img.xz
```

3. Flash it to the SD card (replace `/dev/sdX` with the correct device) with:

```
host$ sudo dd if=ir_acs_life_image.img of=/dev/sdX bs=1M conv=fsync
```

4. Run `fdisk` again with:

```
host $ sudo fdisk /dev/sdX
```

5. Enter `t` to change the type of a partition:

```
fdisk $ t
```

6. Select partition 1 (BOOT Partition):

```
fdisk $ 1
```

7. Select type 1 (EFI System partition):

```
fdisk $ 1
```

> You can list all possible types of partitions by entering the command `l`.

8. Write changes by entering `w`:

```
fdisk $ w
```

4.2 Capsule Update Tests

To generate an EFI capsule you need the U-Boot tool `mkeficapsule`. In a console change the current directory to a cloned u-boot-imx repository[12] from our Git server and make sure to checkout the correct tag or branch supporting your platform, e.g. `v2021.04_2.2.0-phy` as of writing this manual:

```
git clone git://git.phytec.de/u-boot-imx
cd u-boot-imx
git checkout v2021.04_2.2.0-phy
```

Build the `mkeficapsule` tool using an appropriate SDK[13]:

```
make tools-only
```

`mkeficapsule` is now located in the `tools` directory.

Make sure you use a different U-Boot version than the one currently running on the system so you can see the difference. A separate build time may be sufficient.

```
./tools/mkeficapsule --raw path/to/imx-boot --index 1 capsule1.bin
```

Copy the capsule binary to the EFI System Partition with:

```
host$ sudo mount /dev/sdX1 /mnt
host$ sudo cp capsule1.bin /mnt
host$ sync
host$ sudo umount /dev/sdX1
```

Now insert the SD card into the board and boot it up. The board should automatically boot the `ir_acs_life_image`.

Flash

Make sure to have the bootcmd set correctly as described previously, setting `bootcmd`.

Select `bbr/bsa` and press `ESC` to stop `startup.nsh` from running and entering the EFI shell. Select the right filesystem partition, e.g. by entering `fs0:`

12 https://git.phytec.de/u-boot-imx/
13 https://download.phytec.de/Software/Linux/BSP-Yocto-i.MX8MP/BSP-Yocto-NXP-i.MX8MP-PD22.1.1/sdk/ampliphy-vendor-xwayland/

```
Shell> fs0:
```

Type in the following command to update U-Boot with the previously generated capsule:

```
FS0:\> efi\boot\app\capsuleapp capsule1.bin
```

Note

If this might not work on the first try and the following messages are printed instead:

```
FS0:\> EFI\BOOT\app\CapsuleApp.efi capsule1.bin

ASSERT_EFI_ERROR (Status = Not Found)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeServicesTableLib/DxeServicesTableLib.c(58): !
EFI_ERROR (Status)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeServicesTableLib/DxeServicesTableLib.c(59):
gDS != ((void *) 0)

ASSERT_EFI_ERROR (Status = Not Found)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/Build/MdeModule/DEBUG_GCC5/AARCH64/MdeModulePkg/Application/
CapsuleApp/CapsuleApp/DEBUG/AutoGen.c(415): !EFI_ERROR (Status)

ASSERT_EFI_ERROR (Status = Not Found)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeHobLib/HobLib.c(48): !EFI_ERROR (Status)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeHobLib/HobLib.c(49): mHobList != ((void *) 0)
CapsuleApp: creating capsule descriptors at 0x9D6DC040
CapsuleApp: capsule data starts          at 0x9D509040 with size 0x17209C
CapsuleApp: capsule block/size              0x9D509040/0x17209C
CapsuleApp: failed to update capsule - Unsupported
```

Make sure you have set up the BOOT Partition of the `ir_acs_life_image` as an EFI System Partition, as described previously. Otherwise the bootloader can not save any EFI variables.
This can also happen when booting the `ir_acs_life_image` for the first time. Just reset the board and try again. Usually it should work on the second try.

There is a bug, where the exact name of the capsule can be manually written (`capsule1.bin`). When trying to update, the system would just error out with "capsule not found" even if the capsule is there. The workaround is typing `ls` to list any files. Copy the name of the `capsule1.bin` file and paste it after `efi/boot/app/capsuleapp`

```
FS0:\> EFI/BOOT/app/CapsuleApp.efi capsule1.bin

ASSERT_EFI_ERROR (Status = Not Found)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeServicesTableLib/DxeServicesTableLib.c(58): !
EFI_ERROR (Status)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeServicesTableLib/DxeServicesTableLib.c(59):
gDS != ((void *) 0)

ASSERT_EFI_ERROR (Status = Not Found)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/Build/MdeModule/DEBUG_GCC5/AARCH64/MdeModulePkg/Application/
CapsuleApp/CapsuleApp/DEBUG/AutoGen.c(415): !EFI_ERROR (Status)

ASSERT_EFI_ERROR (Status = Not Found)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeHobLib/HobLib.c(48): !EFI_ERROR (Status)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeHobLib/HobLib.c(49): mHobList != ((void *) 0)
CapsuleApp: capsule image (capsule1.bin) is not found.
FS0:\> ls
Directory of: FS0:\
00/00/0000 00:00 1,515,676 capsule1.bin
00/00/0000 00:00 <DIR> 0 EFI
00/00/0000 00:00 <DIR> 0 grub
00/00/0000 00:00 298 grub.cfg
00/00/0000 00:00 32,930,304 Image
00/00/0000 00:00 92,389,888 ramdisk-busybox.img
00/00/0000 00:00 592 ubootefi.var
5 File(s) 126,836,758 bytes
2 Dir(s)FS0:\> EFI/BOOT/app/CapsuleApp.efi capsule1.bin

ASSERT_EFI_ERROR (Status = Not Found)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeServicesTableLib/DxeServicesTableLib.c(58): !
EFI_ERROR (Status)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeServicesTableLib/DxeServicesTableLib.c(59):
gDS != ((void *) 0)

ASSERT_EFI_ERROR (Status = Not Found)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/Build/MdeModule/DEBUG_GCC5/AARCH64/MdeModulePkg/Application/
CapsuleApp/CapsuleApp/DEBUG/AutoGen.c(415): !EFI_ERROR (Status)

ASSERT_EFI_ERROR (Status = Not Found)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeHobLib/HobLib.c(48): !EFI_ERROR (Status)
ASSERT [CapsuleApp] /home/cherat01/ATEG/SystemReady/BBR/arm-systemready/IR/
scripts/edk2/MdePkg/Library/DxeHobLib/HobLib.c(49): mHobList != ((void *) 0)
```

```
CapsuleApp: creating capsule descriptors at 0x9D69F040
CapsuleApp: capsule data starts at 0x9D50A040 with size 0x17209C
CapsuleApp: capsule block/size 0x9D50A040/0x17209C
#resetting ...
```

The board will automatically reset after the update has finished and the new U-Boot will boot. During startup, there is a different version printed, e.g.:

```
U-Boot 2021.04 (Mar 25 2023 - 11:56:12 +0000)
```

Information

See this NXP forum thread for the EFI capsule update: https://community.nxp.com/t5/i-MX-Processors/i-MX8M-Mini-EVK-ARM-SystemReady-IR-EFI-Capsule-update-issues/m-p/1372756/highlight/true

4.3 General Requirements and ACS Test Suite

Checking for the general requirements of SystemReady IR compatibility is explained in the following document in ARM's documentation: https://developer.arm.com/documentation/DUI1101/1-1/Test-SystemReady-IR?lang=en

Performing the ACS tests is easy. Simply boot the entry `bbr/bsa` and wait for the test to start automatically. For the full documentation see https://developer.arm.com/documentation/DUI1101/1-1/Test-with-the-ACS?lang=en.