



Support ADC in the Linux i.MX RT1024 BSP

**Detailed Requirements and
Design**

rm7032-drad-1_0.doc

<i>RM:</i>	7032
<i>Revision:</i>	1.0
<i>Date:</i>	9/4/2024

TABLE OF CONTENTS

1.	OVERVIEW	3
2.	REQUIREMENTS	3
2.1.	Detailed Requirements	3
2.2.	Detailed Non-Requirements	3
3.	DESIGN	3
3.1.	Detailed Design	3
3.1.1.	<i>Design: Demo project</i>	3
3.1.2.	<i>Design: Linux ADC Device Driver</i>	3
3.2.	Effect on Related Products	3
3.3.	Changes to User Documentation	4
3.4.	Alternative Design	4
4.	TEST PLAN.....	4
4.1.	Secure Download Area.....	4
4.2.	Downloadable Files.....	4
4.3.	Test Set-Up.....	4
4.3.1.	<i>Hardware Setup</i>	4
4.3.2.	<i>Software Setup</i>	4
4.4.	Detailed Test Plan	5
4.4.1.	<i>Test Plan: Demo Project</i>	5
4.4.2.	<i>Test Plan: Linux ADC Driver</i>	5

1. Overview

The following is a high-level overview of the problem being resolved by this project:

This project develops Linux ADC device driver in the Linux BSP for the i.MX RT1024 processor.

2. Requirements

2.1. Detailed Requirements

The following are the requirements for this project:

1. Provide a Linux demo project combining all the requirements in this project.
 - *Rationale:* Needed to let Customer integrate results of this project into target embedded application.
Implementation: Section: "Design: Demo project".
Test: Section: "Test Plan: Demo Project".
2. Develop Linux ADC device driver for the i.MX RT1024 ADC controller.
 - *Rationale:* Explicit Customer requirement.
Implementation: Section: "Design: Linux ADC Device Driver".
Test: Section: "Test Plan: Linux ADC Driver".

2.2. Detailed Non-Requirements

The following are the non-requirements for this project that may otherwise not be obvious:

1. None

3. Design

3.1. Detailed Design

3.1.1. Design: Demo project

This project will enable the required ADC functionality in the Linux configuration ("embedded project") called `rootfs`, which resides in a `projects/rootfs` directory, relative to the top of the Linux i.MX RT1024 installation.

3.1.2. Design: Linux ADC Device Driver

The clock driver for IMXRT1024 SoC will be updated to support the ADC clock gates.

There is the existing `vf610_adc` driver in the Linux sources, which provides support for the ADC controller of the Vybrid and some other SoCs from the i.MX family. The i.MXRT1024 SoC has a similar ADC controller, however there are some low-level differences in the hardware interfaces (register offsets, etc). The driver will be updated to support the register set of the IMXRT1024 SoC. Support for the IMXRT1024 ADC will be enabled via the DTS kernel file, using the new `fsl,imxrt10xx-adc` compatibility string.

3.2. Effect on Related Products

This project makes the following updates in the related products:

- None

3.3. Changes to User Documentation

This project updates the following user documents:

- None

3.4. Alternative Design

The following alternative design approaches were considered by this project but then discarded for some reason:

- None

4. Test Plan

4.1. Secure Download Area

The downloadable materials developed by this project are available from a secure Web page on the Emcraft Systems web site. Specifically, proceed to the following URL to download the software materials:

for the i.MX RT1024 BSP:

- <https://www.emcraft.com/imxrtaddon/imxrt1024/adc>

The page is protected as follows:

- Login: *CONTACT EMCRAFT*
- Password: *CONTACT EMCRAFT*

4.2. Downloadable Files

The following files are available from the secure download area:

- `linux-adc.patch` - patch to the Linux kernel sources;
- `projects-adc.patch` - patch to the `rootfs` project;
- `rootfs.uImage` - prebuilt bootable Linux image;

4.3. Test Set-Up

4.3.1. Hardware Setup

The following hardware setup is required for the i.MX RT 1024 board:

- The i.MXRT1024 EVK board.

4.3.2. Software Setup

The following software setup is required:

1. Download the files listed in Section: "Downloadable Files" to the top of the Linux i.MX RT installation.

2. Install the BSP, as per the respective "Installing and activating cross development environment" document in the "Software" section on the Emcraft site.
3. From the top of the Linux installation, activate the Linux cross-compile environment by running:

```
$ . ./ACTIVATE.sh
```

4. From the top of the BSP installation, go to the Linux kernel tree and install the kernel patch, eg:

```
$ cd linux/
$ patch -p1 < ../../linux-adc.patch
```

5. From the top of the Linux installation, go to the `projects` sub-directory, and patch the `rootfs` project:

```
$ cd projects/
$ patch -p1 < ../../projects-adc.patch
```

6. Build the `rootfs` project:

```
$ cd projects/rootfs
$ make
```

4.4. Detailed Test Plan

4.4.1. Test Plan: Demo Project

Perform the following step-wise test procedure:

1. Go to the `projects/rootfs` directory, build the loadable Linux image (`rootfs.uImage`) and copy it to the TFTP directory on the host:

```
$ cd projects/rootfs
$ make
```

2. Boot the loadable Linux image (`rootfs.uImage`) to the target via TFTP and validate that it boots to the Linux shell:

```
=> run netboot
Using ethernet@402D8000 device
TFTP from server 192.168.0.103; our IP address is 192.168.0.104
Filename 'imxrt1024/rootfs.uImage'.
Load address: 0x80007fc0
Loading: #####
#####
#####
#####
#####
...
/ # uname
Linux
/ #
```

4.4.2. Test Plan: Linux ADC Driver

Perform the following step-wise test procedure:

1. Connect 3.3V to channel 9 of ADC1 (for this, one can wire J20.4 to J17.2)
2. Check that the measurement on channel 9 of ADC1 is near maximum (4096):

```
cat /sys/bus/iio/devices/iio:device0/in_voltage9_raw
4088
```

3. Connect GND to channel 9 of ADC1 (for this, one can wire J20.12 to J17.2)
4. Check that the measurement on channel 9 of ADC1 is near minimum (0):

```
/ # cat /sys/bus/iio/devices/iio:device0/in_voltage9_raw  
2
```

5. Connect a wire with known voltage applied to channel 9 of ADC1.
6. Check that the measurement on channel 9 of ADC1 corresponds to the applied voltage measured with a voltage meter:

```
/ # cat /sys/bus/iio/devices/iio:device0/in_voltage9_raw  
3267  
/ # cat /sys/bus/iio/devices/iio:device0/in_voltage_scale  
0.805664062  
/ #
```

(3278 corresponds to $3267 * 0.805664062 = 2.6V$)