NFC Read Library for i.MX7Dual Sabre

1. Introduction

The NFC Reader Library is a feature complete software support library for NXP's NFC frontend ICs. It is designed to give developers a faster and simpler way to deliver NFC-enabled products. This multi-layer library, written in C, makes it easy to create NFC based applications.

This document provides instructions on how to install the NFC Reader Library on imx7dsabresd and communicate with PN5180, a NFC frontend. The document describes the steps required to connect the board to an OM25180TWR, the wire connections, the changes in the device tree, and the library configuration.

2. Building the Linux image and the Bal kernel module

This section describes how to build the Linux image using Yocto and how to compile the Bal kernel module.

Requirements: a Linux host PC (ex. Ubuntu 14.04/16.04) and root permissions.
Step 1:
To download the required host packages, use:

```
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat libsdl1.2-dev
```

Contents

Specific for Ubuntu:
```
$ sudo apt-get install libsdl1.2-dev xterm sed cvs subversion coreutils texi2html docbook-
utils python-pysqlite2 help2man make gcc g++ desktop-file-utils \
libgl1-mesa-dev libglu1-mesa-dev mercurial autoconf automake groff curl lzop asciidoc
```

Step 2:
To setup the repo utility (a tool written on top of git), run the commands:
```
$ mkdir ~/bin (this step may not be needed if the bin folder already exists)
$ curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
```

Step 3:
Then add the following line to .bashrc to ensure the ~/bin is in the PATH variable.
```
$ export PATH=~/bin:$PATH
```

Step 4:
To download the Freescale Yocto Project Community BSP:
```
$ mkdir fsl-release-bsp
$ cd fsl-release-bsp
$ repo init -u git://git.freescale.com/imx/fsl-arm-yocto-bsp.git -b imx-4.1-krogoth
$ repo sync
```

Step 5:
To build the image, in the fsl-release-bsp run the commands:
```
$ mkdir buildDevSpi
$ DISTRO=fsl-imx-xwayland MACHINE=imx7dsabresd source fsl-setup-release.sh -b buildDevSpi
$ bitbake fsl-image-machine-test
```

Step 6:
To build the toolchain, run:
```
$ bitbake meta-toolchain
$ cd buildDevSpi/tmp/deploy/sdk/
$ ./fsl-imx-xwayland-glibc-x86_64-meta-toolchain-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
```
Accept the default parameters.

Step 7:
In order to deploy the image on an SD card use:
```
$ sudo dd if=fsl-image-machine-test-imx7dsabresd.sdcard of=<sd card> bs=1M && sync
```
The image is found in buildDevSpi/tmp/deploy/images/imx7dsabresd.

Step 8:
For the kernel module compilation, setup the console environment:
```
$ . /opt/fsl-imx-xwayland/4.1.14-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
```

Then, in the kernel module directory, replace the path with the linux build directory in the Makefile and run:
```
$ make
```
The bal.ko is the compiled module.

Step 9:
To use the kernel menuconfig, run:
```
$ bitbake -c menuconfig linux-imx
```

Another useful command, for rebuilding the linux kernel image, is:
```
$ bitbake -f -c compile linux-imx; bitbake -f -c deploy linux-imx; bitbake -f -c compile fsl-
image-machine-test; bitbake -f fsl-image-machine-test
```

3. Host interface

The interface of the PN5180 to a host is based on a SPI interface, extended by the signal line BUSY. Only half-duplex data transfer is supported and no chaining is allowed, meaning that the whole instruction has to be sent, or the whole receiver buffer has to be read out.

The module is connected to the i.MX7D board using the mikro bus expansion port in the following way: MK_BUS_CS, MK_BUS_SCK, MK_BUS_MOSI, MK_BUS_MISO are used for the SPI bus lines. MK_BUS_INT, MK_BUS_RX, MK_BUS_TX are used for the BUSY, RESET and IRQ lines.

The pin configuration will be the following: GPIO6-IO22 will be the CS, GPIO6-IO14 will be BUSY, GPIO-IO12 will be RESET and GPIO-IO13 will be IRQ.
The DWL pin, which can be used for firmware update, will be connected to GND. A common ground is also required.

Table 1.   **Connections Table**

Jumper	Jumper Pins	Description	i.MX7D I/O	Tower Edge
J4	1 – 2	SPI Clk Selection	ECSPI3_SCLK (SAI2_RX_DATA)	B7
J20	1 – 2	PN5180 Reset	GPIO6_IO12 (SAI1_RX_DATA)	B8
J1	1 – 2	SPI SS0	GPIO6_IO22 (SAI2_TX_DATA)	B9
J3	1 – 2	SPI MOSI	ECSPI3_MOSI (SAI2_TX_BCLK)	B10
J2	1 – 2	SPI MISO	ECSPI3_MISO (SAI2_TX_SYNC)	B11
J19	2 – 3	PN5180 BUSY	GPIO6_IO14 (SAI1_TX_SYNC)	B58
J5	1 – 2	PN5180 IRQ	GPIO6_IO13 (SAI1_TX_BCLK)	B62
X	X	PN5180 DWL	GND	B52
X	X	GND	GND	B2

NFC Read Library for i.MX7Dual Sabre, Application Note, Rev. 0, 02/2018

4. Kernel configuration

In order to allow the library to manage the RESET, IRQ and BUSY pins, the options for Debug GPIO and Userspace I/O drivers must be enabled (in menuconfig, Device Drivers -> GPIO Support -> Debug GPIO and Device Drivers -> Userspace I/O -> Userspace I/O platform driver with generic IRQ).

For controlling the SPI, there are two options: spidev or NXP bal.

For spidev, it is necessary to select Device Drivers -> SPI support -> User mode SPI and apply the imx7d-sdb_spidev.patch (it also does the required pinmuxing).

When using NXP bal, it is necessary to compile the module, initialize it with insmod and apply the imx7d-sdb_bal.patch (it also does the required pinmuxing).

5. Library configuration

For the library configuration, <lib-folder>/Platform/DAL/Board_Imx6ulevkPn5180.h must be replaced with Board_Imx7dsabresdPn5180_bal.h or Board_Imx7dsabresdPn5180_spidev.h (based on the selected SPI interface).

For compilation, the command is:
```
$ ./build.sh yocto /opt/fsl-imx-xwayland/4.1.15-2.1.0/sysroots/
```

The last parameter is the location of the toolchain generated by yocto.
A build folder is generated outside of the source code folder. The applications from the ComplianceApp can be deployed on the board in order to test the functionality provided.

6. References

- i.MX Yocto Project User's Guide describes in more details the image build process.

- NFC Reader Library for Linux Installation describes the steps for connecting a board to an OM25180FDK (the variant that does not use the tower model).

- The document PN5180A0XX-C1-C2 provides more details regarding the PN5180 component functionality.

7. Revision History

Table 2. **Revision history**

Revision number	Date	Substantive changes
0	02/2018	Initial release

arm