UM10493

POS Reference Design - Firmware description

Rev. 1.1 — 21 March 2012 **User manual**
211411 **COMPANY PUBLIC**

Document information

Info	Content
Keywords	PN512, TDA8026, LPC1768, Point of sales terminal, Reference Design
Abstract	This user manual presents the NXP POS_RD evaluation board. This reference design is used to demonstrate and evaluate the implementation of NXP's devices in a Point Of Sales terminal. This documentation describes the Firmware embedded in the LPC1768: How to modify, compile, and load it.

Revision history

Rev	Date	Description
1.1	20120321	Release for FW version 1.1b Section 4 Firmware description and Section 5 Demonstration application: updated
1.0	20110726	First release

Contact information

For additional information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

The UNXPOS board is a reference design made by NXP to demonstrate and evaluate a simple implementation of NXP devices in a Point Of Sales Terminal.

With this design, Contact and Contactless applications can be evaluated, as the design embeds all the needed Hardware and software for the following interfaces:

- Contactless user card with PN512 + power amplifier
- Contact user card with the TDA8026 main slot
- Up to 4 Secure Access Modules with TDA8026 slots 2 to 5.

The user interface is composed of an LCD screen and a Keyboard.

The following figure gives an overview of the reference design hardware:

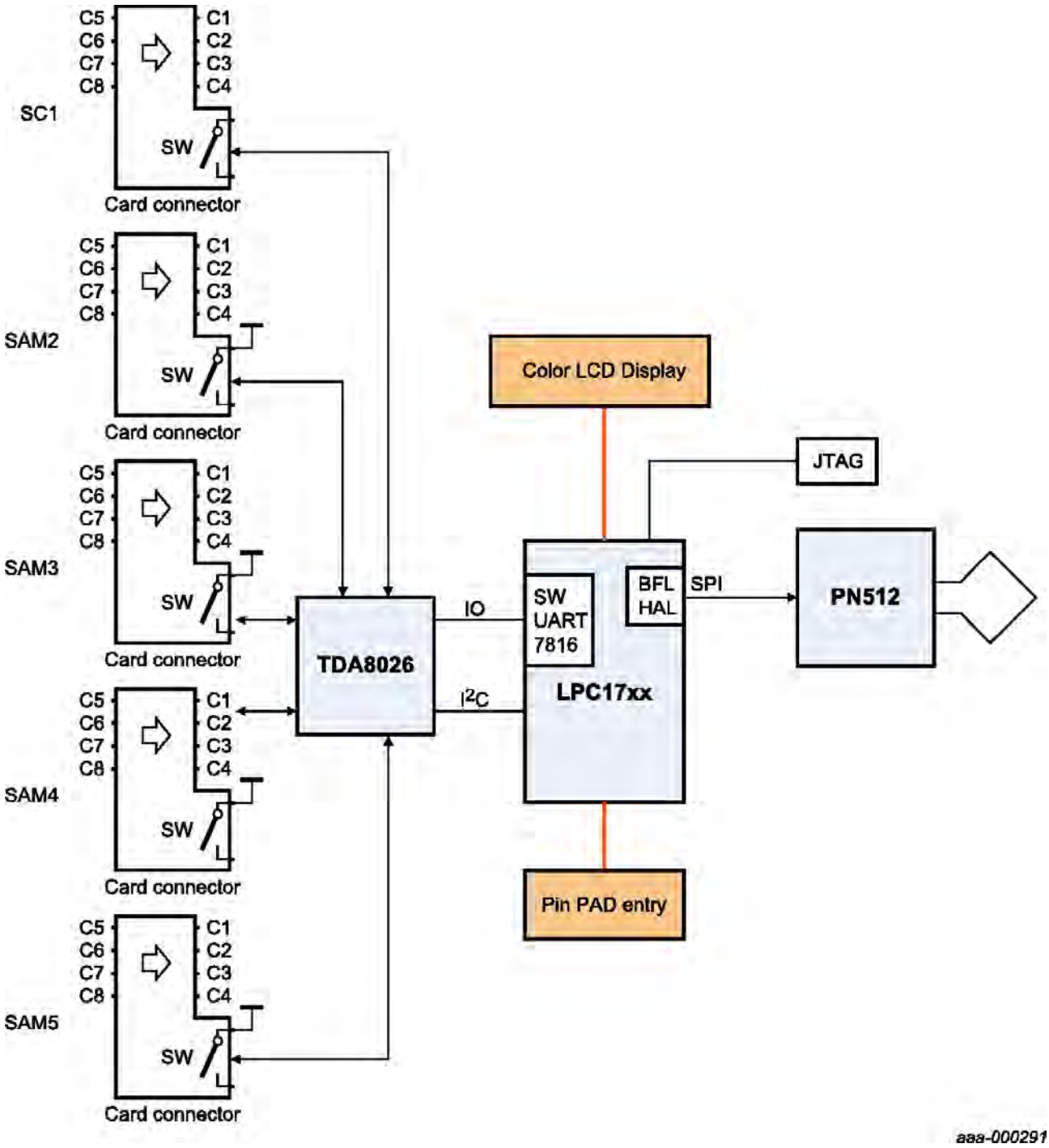

Fig 1. POS Reference Design – Hardware block diagram

2. Firmware overview

The board is delivered with a pre-loaded firmware, so that the board is ready to be used for demonstration.

The LPC1768 embeds all the low level software to drive these interfaces.

Here is an overview of the default embedded firmware:

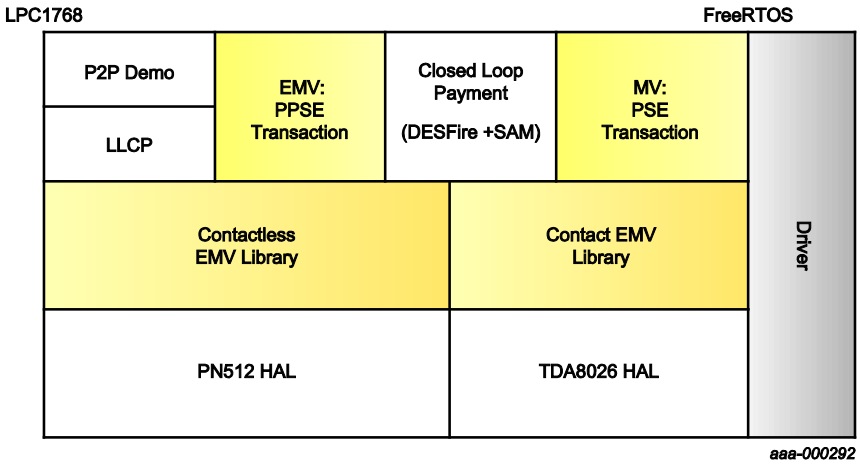

Fig 2. POS Reference Design – Software block diagram

3. Firmware tools

The firmware has been developed using the following tolls:

- LPCXpresso version 3.6.1, as editor, file manager…
- CodeSourcery version 4.5.1 as compiler
- MinGW as Linker and make tool

3.1 MinGW

MinGW is used as a linker and to provide a “make” tool in case it is not available under the used computer.

MinGW is free for download from the website:

www.nxp.com/redirect/mingw.org

The download link will guide you to the SourceForge webpage to find the direct DL link:

www.nxp.com/redirect/sourceforge.net/projects/mingw/files/

To use MinGW in this project, download the installer file from the web, and install it under C:\MinGW (should be the default path).

In case a make tool is not available under the development OS environment, the tool from MinGW can be used. To do so, the **mingw32-make.exe** file must be renamed to **make.exe**. This file is located in C\MinGW\bin

3.2 Development environment – LPCXpresso

3.2.1 Install LPCXpresso

LPCXpresso is a development environment developed by NXP together with CodeRed to develop software for NXP's LPC microcontrollers.

See LPCXpresso web page: http://www.nxp.com/lpcxpresso

The first step to use LPCXpresso is to go on the download page and download the installer.

www.nxp.com/redirect/lpcxpresso.code-red-tech.com/LPCXpresso/. The registration is necessary for downloading the tool, but the tool is then free to download.

Once the installer has been downloaded, launch the executable file and install LPCXpresso with default parameters.

After the installation is complete, launch LPCXpresso. The first thing to do after the IDE has started is to choose a workspace. For compilation reason, **it is recommended to use a path for this workspace without space in the absolute path**.

To switch the workspace after the IDE has started, choose

File>Switch Workspace>Other (see below)

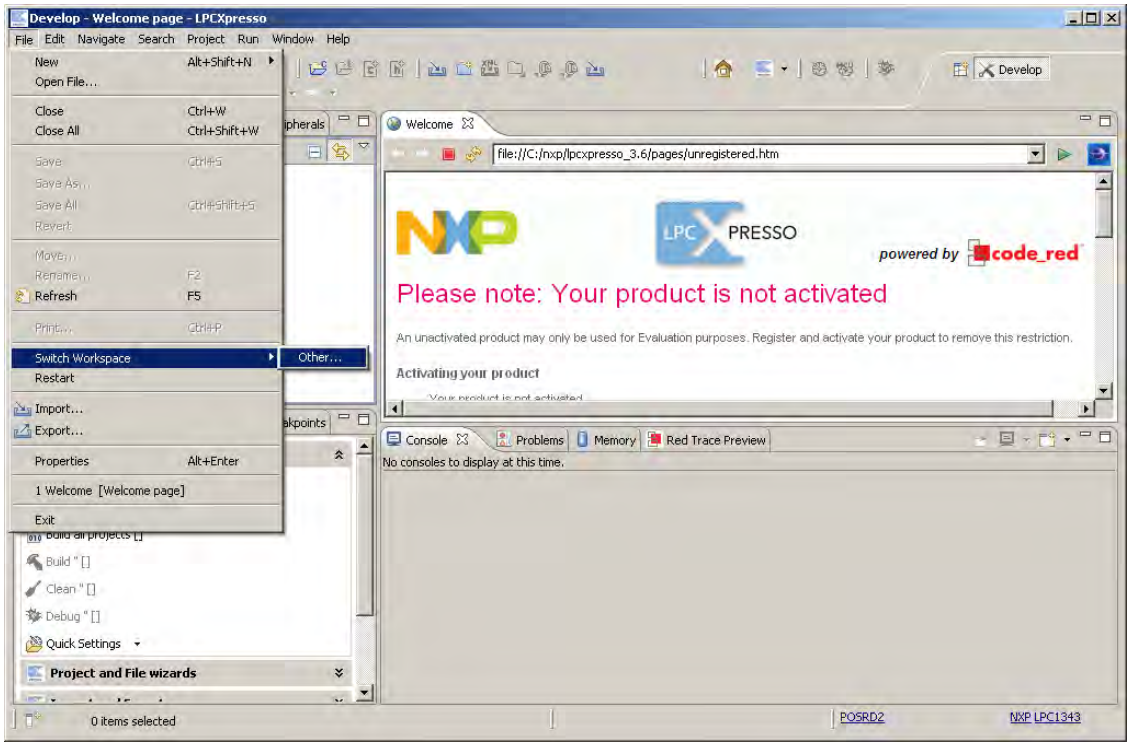

Fig 3. LPCXpresso – Switch workspace

Then choose a workspace (e.g C:\Workspace). If the folder does not exist, it will be created:

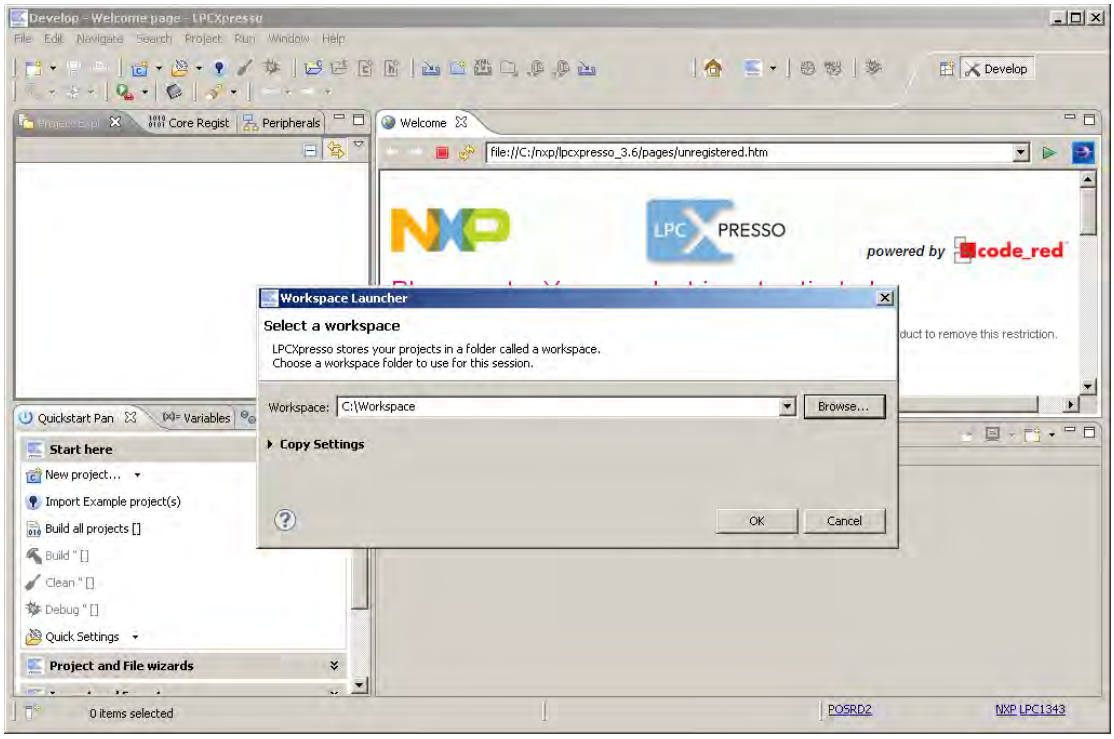

Fig 4. LPCXpresso – Choose workspace

Then click on “OK”.

After this action, LPCXpresso automatically restarts with the new workspace path.

3.2.2 Import POSRD2 Project

To start working on the POS Reference Design Firmware, it must be imported into LPCXpresso.

To do so, choose the link “Import Example Project” in the Quickstart pan (bottom left):

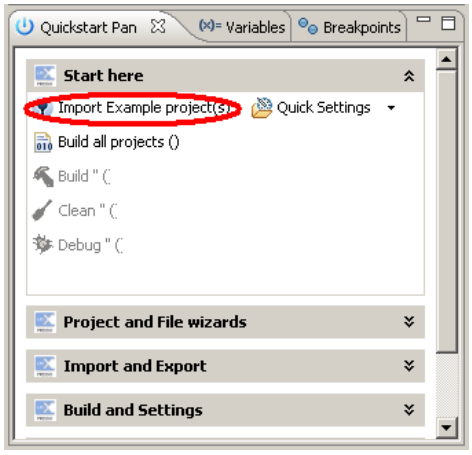

Fig 5. Import project link

Browse to find the POSRD2.zip package supplied with the Reference design data.

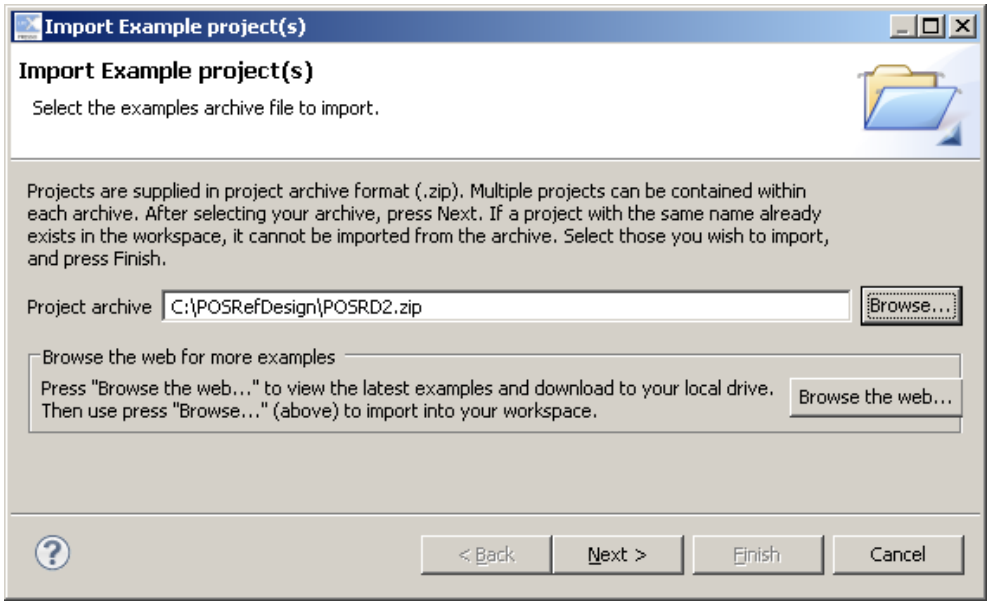

Fig 6. Browse for POSRD2.zip package

Click Next, then select the POSRD2 project checkbox and click Finish.

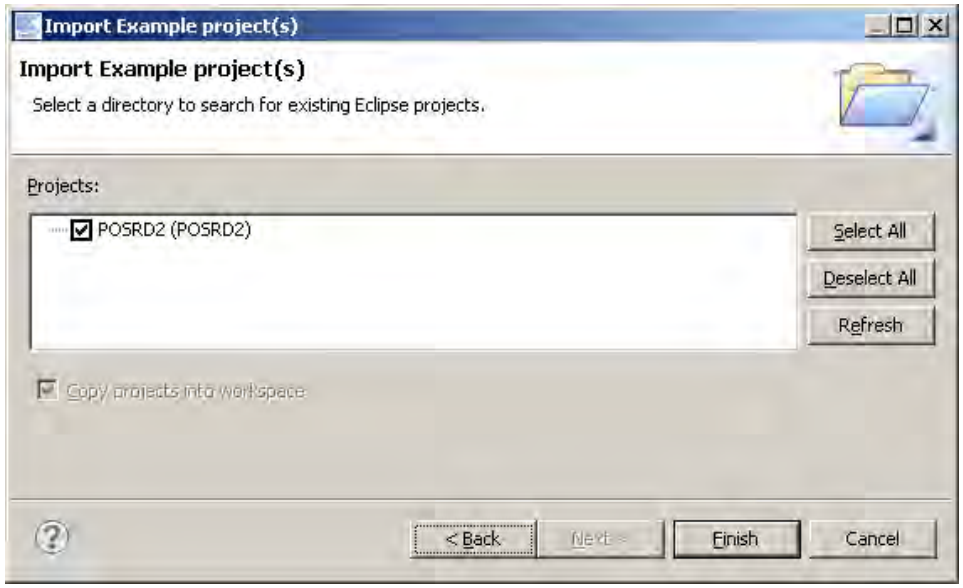

Fig 7. Select POSRD2 Project

The project is now imported into your LPCXpresso environment. The files can now be browsed from the top left pan:

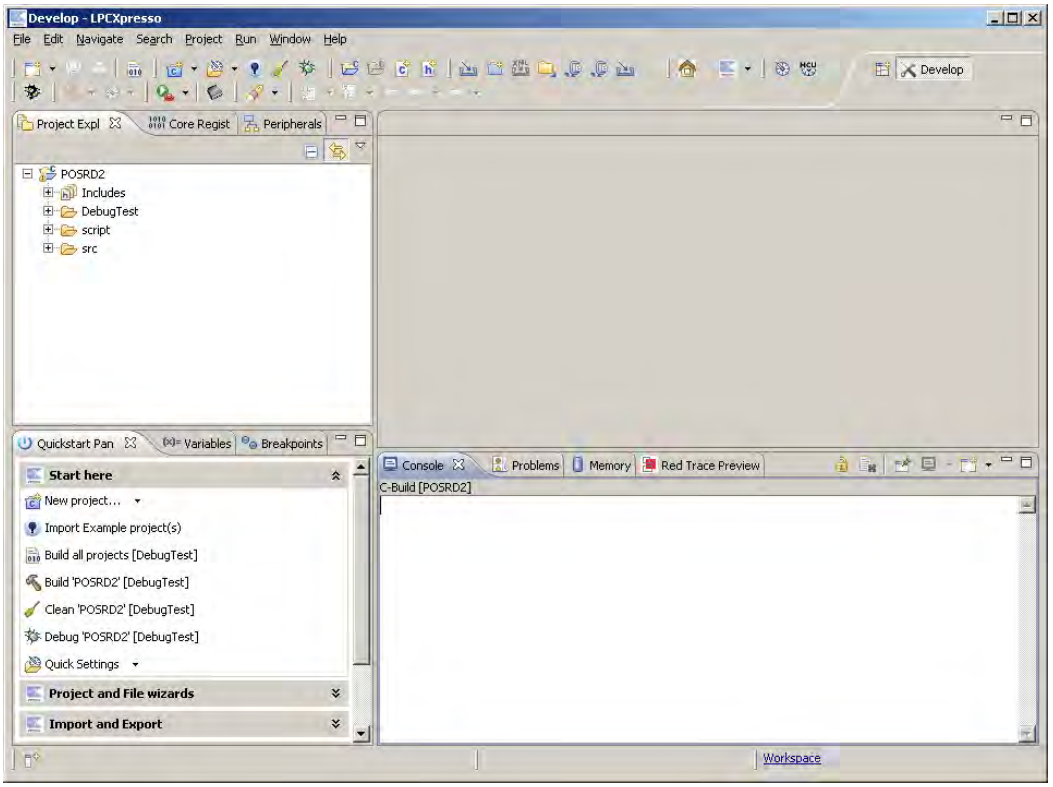

Fig 8. LPCXpresso – Project POSRD2 imported

3.3 Compiler – Code Sourcery

Before being able to compile the source code, the compiler must be installed. Code Sourcery G++ Lite has been used for this FW.

Sourcery G++ Lite edition for ARM embeds a GNU tool chain for ARM microcontrollers: www.nxp.com/redirect/codesourcery.com/sgpp/lite_edition

It can be downloaded from the CodeSourcery download webpage:

www.nxp.com/redirect/codesourcery.com/sgpp/lite/arm/portal/release1592

The IA32 Windows installer is the right file to install the compiler. The version that have been used for this POSRD2 FW release is version 2010.09-51

Once the package is downloaded, it can be installed by launching the installer.

It must be installed in the default location, with default configuration so that the LPCXpresso project can find the binaries with no change.

3.3.1 Build Source code

After Code Sourcery has installed successfully, LPCXpresso is ready to build:

Press CTRL+B or choose Menu → Project>Build All to start the compilation.

The full files must be compiled and the console should finally display the same information as below.

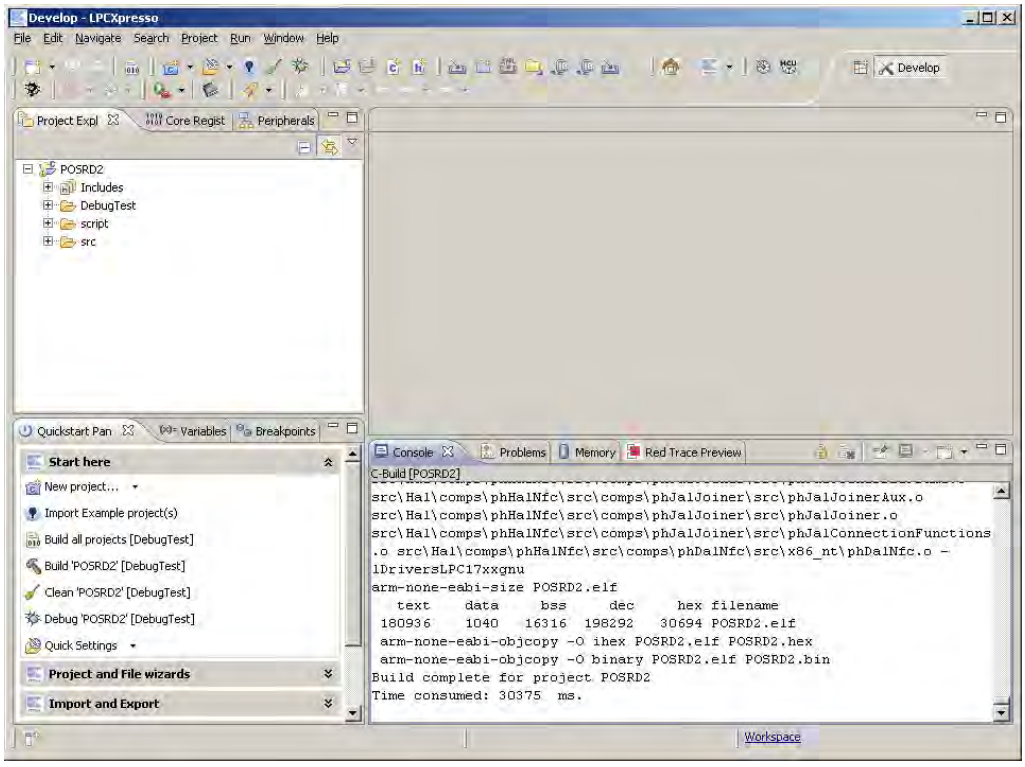

Fig 9. First compilation console output

3.4 FlashMagic

After the compilation ran correctly, the project created a .HEX file that can be loaded to the POS Ref Design board using the serial port.

To use FlashMagic, it must first be downloaded from the main FlashMagic webpage, and installed.

The download page is www.nxp.com/redirect/flashmagictool.com

3.4.1 FlashMagic direct use

After installation, run Flash Magic and configure it as follows:

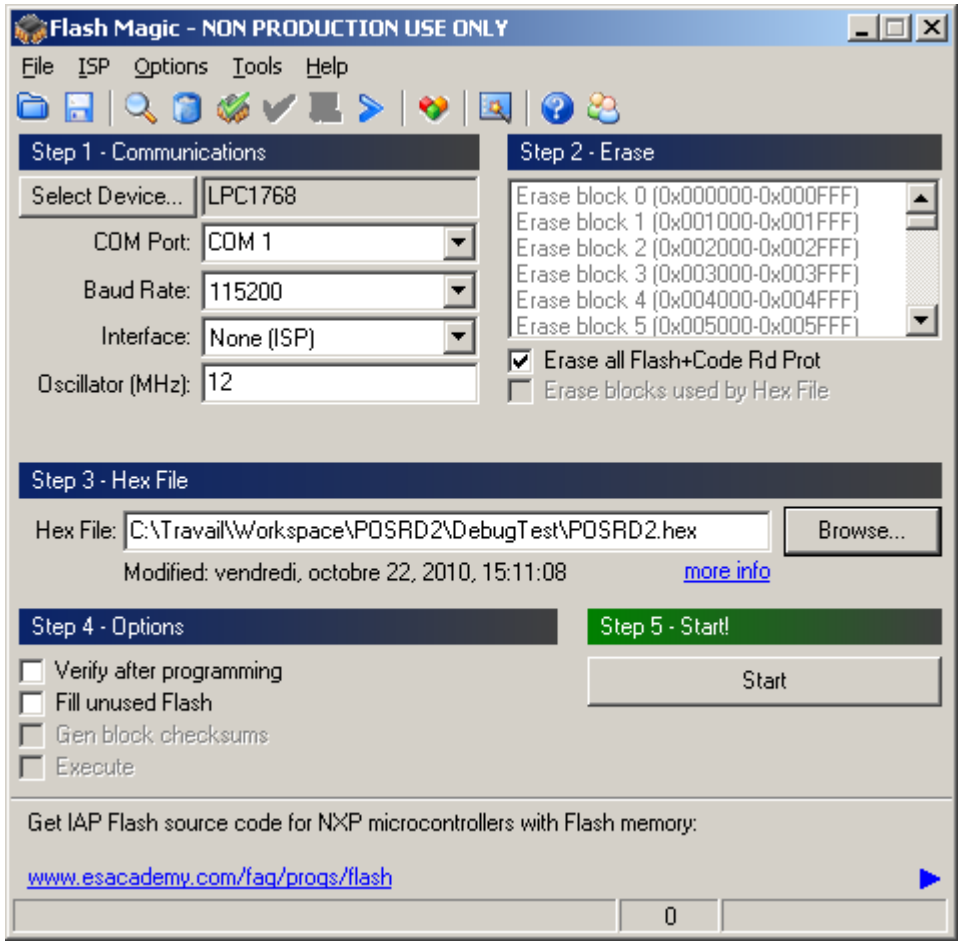

Fig 10. FlashMagic configuration

If the board is correctly supplied and connected to the right serial port (to be configured differently in FlashMagic if different than COM1), click on the “Start” button and the load starts.

FlashMagic handles the RESET and ISP pins of the microcontroller to force a restart in ISP mode (In Serial Programming). Then there is no required action from the user to load the firmware.

To have RESET and ISP work, the 2 jumpers close to the main board serial port connector must be in place (called J6 and J7 in the schematics).

After the end of the FW load with FlashMagic, the microcontroller restarts automatically and the new firmware runs.

Depending on the serial port management from the computer, the host PC may keep the LPC in reset mode when the serial port is connected. In this case, it is required either to remove the RESET jumper (J7), to unplug the serial cable, or to simply open the com port from the computer.

3.4.2 Run FlashMagic from LPCXpresso

FlashMagic does not need to be started manually to load the firmware, LPCXpresso can do it for you at the end of the build.

To configure it, under LPCXpresso, go to

Menu → Run>External tools> External tools Configuration

Here click on the 'New' button and configure the tool as follows:

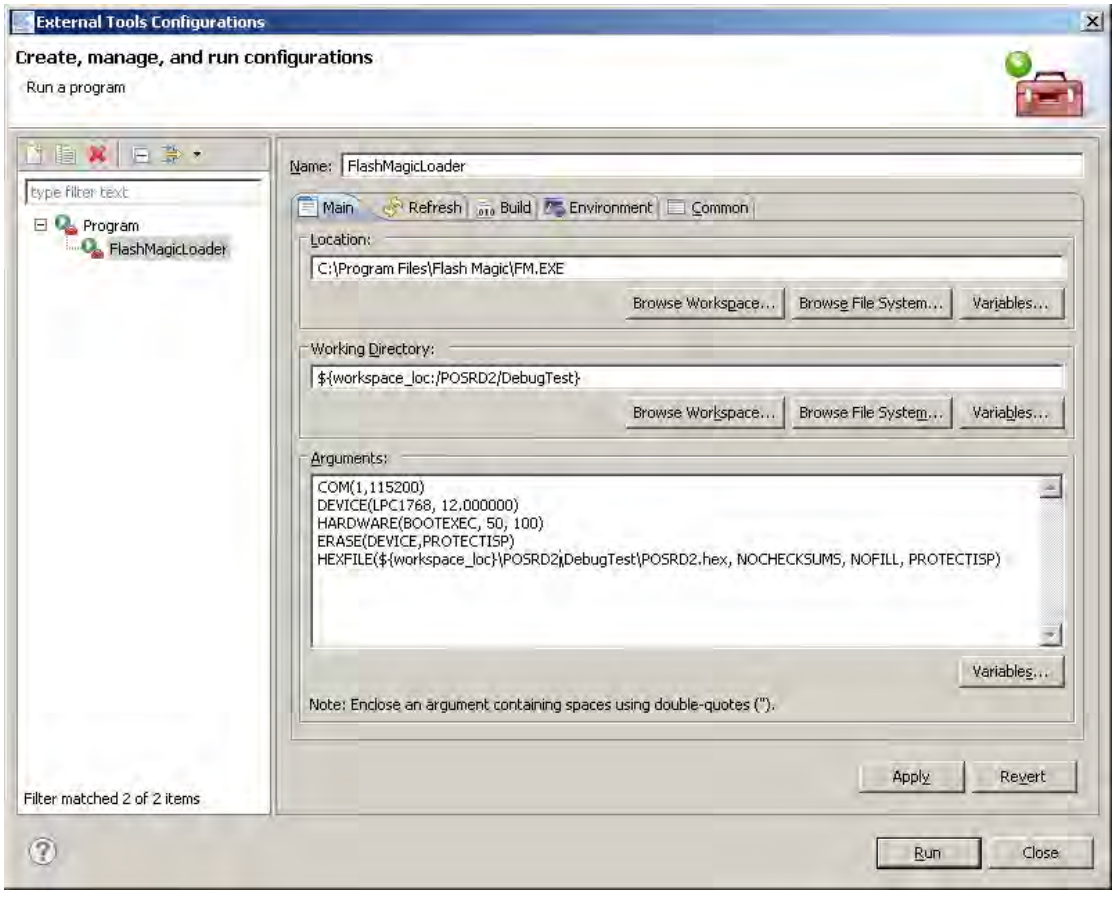

Fig 11. LPCXpresso – Configure FlashMagic

The configuration data are the following (for copy/paste):

```
C:\Program Files\Flash Magic\FM.EXE

${workspace_loc:/POSRD2/DebugTest}

COM(1,115200)
DEVICE(LPC1768, 12.000000)
HARDWARE(BOOTEXEC, 50, 100)
ERASE(DEVICE,PROTECTISP)
HEXFILE(${workspace_loc}\POSRD2\DebugTest\POSRD2.hex,
NOCHECKSUMS, NOFILL, PROTECTISP)
```

Of course, the configuration must be adapted to the environment (different names, different COM port…) The COM Port number is the first parameter of `COM(1,115200)` (here it is COM1)

After this external tool has been configured, choose run: The FW is automatically built and loaded.

3.4.3 Common errors during the .hex load

During the load using FlashMagic, either direct or embedded in LPCXpresso, the following errors can occur:

3.4.3.1 Unable to communicate

Fig 12. Unable to communicate

This means that the tool cannot open the COM port. It is either unavailable (maybe the wrong COM port has been defined in the tool configuration) or already open by another application.

Check again your COM port number, and close all applications that may use it.

3.4.3.2 Failed to autobaud

Fig 13. Failed to autobaud

This error means that the COM port has been opened, but the access to the CPU in ISP mode cannot be done.

It can happen for several reasons:

- Bad connection between the host computer and the POS board.

The connection must be done using a straight cable, with at least: RX, TX, DTR, RTS and GND.

- Jumpers missing on the board to connect DTR and RTS to ISP and RST. Connect two jumpers on J6 and J7, close to the RS232 connectors
- RST not connected on the board. On some boards (first release), the resistor R11 between the RS232 connector and the RST jumper is missing. Then the RST cannot be controlled by FlashMagic. To resolve it, connect a 0 ohm resistor or a solder bridge on the R11 pads.

4. Firmware description

4.1 Version description

4.1.1 1.0

First released FW version.

EMVCo digital tests passed: debug session report from FIME provided.

4.1.2 1.1b

Updated application: P2P at 106 kbps added for easier connection to some mobile phones

Update application state machine: go back in main menu instead of hidden menu after mobile phone data exchange

Remove unused folders and files

Not tested for EMVCo certifications

4.2 Stack overview

The following figure shows the different components of the POSRD firmware:

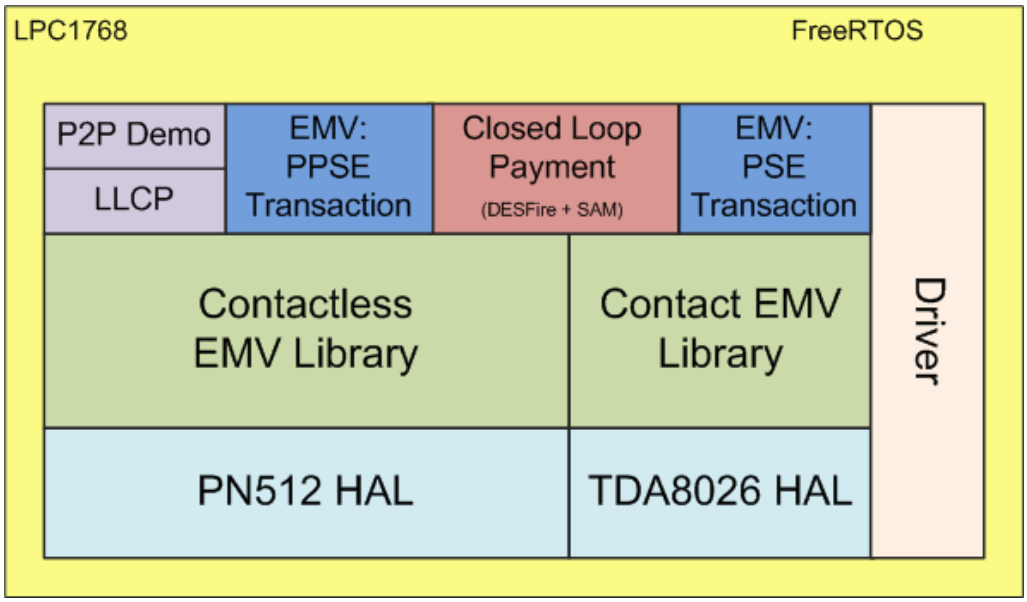

Fig 14. Firmware stack overview

4.3 Component list

Below figures show the directory structure view of the firmware components

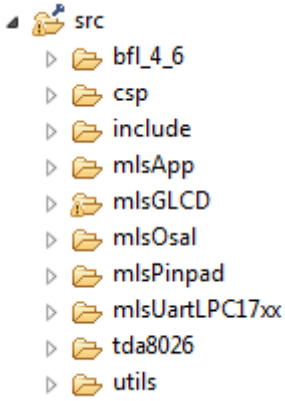

Fig 15. Files directory structure

4.3.1 bfl_4_6

The BFL (Basic function library) provides low level control of the PN512 as well as contactless (ISO14443, ISO18092, MIFARE…) protocols support

4.3.2 csp

This component provides low level control of the LPC1768 Core peripherals (timer, I2C, SPI…)

4.3.3 mlsApp

This component is the demonstration application

4.3.4 mlsGLCD

This component provides low level control of the LCD display

Two different LCD types can be used with this Reference design kit. They differ in the way to access them.

The file GLCD.c provides the two drivers, with a compilation option:

```
#define LCD_RM68050   (0)    // For LCD PCB before 2011 05 23
#define LCD_HX8347D   (1)    // For LCD PCB After 2011 05 23
#define LCD_CONTROL   LCD_HX8347D
```

Fig 16. LCD type – Compiler option

The LCD Type can be known with the build date, which is written below the LCD PCB:

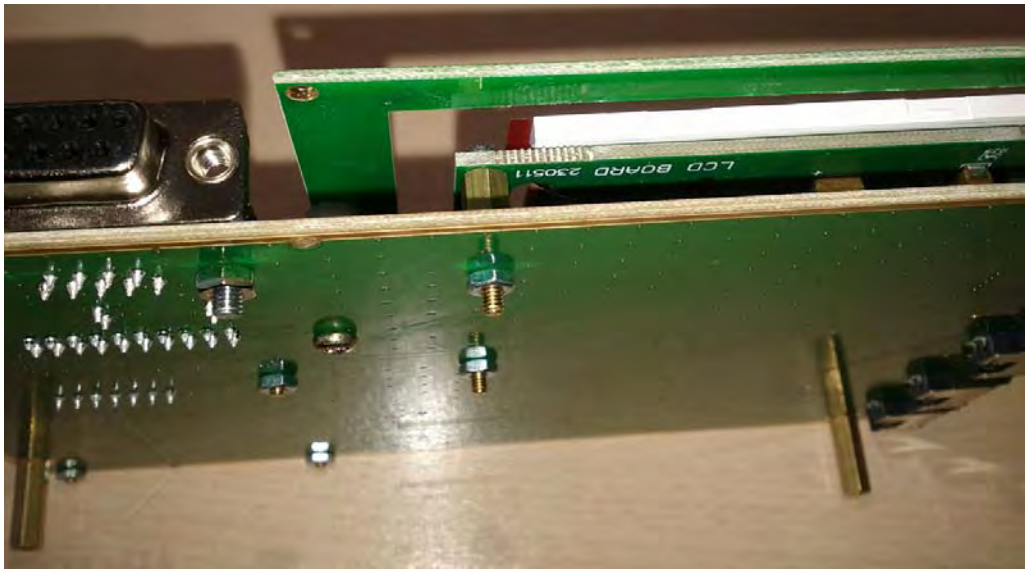

Fig 17. LCD PCB date

If the date is 2011 05 23 (“LCD BOARD 230511”) or earlier, the value LCD_CONTROL must be defined equal to LCD_HX8347D.

If the date is before 2011 05 23, LCD_CONTROL must be equal to LCD_RM68050.

4.3.5 mlsOsal

This component is made of an OS (FreeRTOS) for the LPC1768, an OS abstraction layer and a memory management module

4.3.6 mlsPinpad

This component provides low level control of the keypad

4.3.7 mlsUartLPC17xx

This component provides low level control of the LPC1768 UART

4.3.8 tda8026

This component provides low level control of ISO7816 UART (emulated using LPC1768 GPIO pins and timer) as well as ISO7816 protocol support

4.3.9 utils

This folder includes the APIs for the application to use the HW:

- Automatic Device Discovery (Contact and Contactless polling)
- Data exchange
- LLCP (protocol for NFC P2P protocol)

5. Demonstration application

5.1 Tasks description

The Demonstration software part is made on several tasks running in parallel in the microcontroller:

- The main task is only dedicated to the boot execution and then the creation of the Foreground task
- The Foreground task (which create other tasks) is responsible for managing the LCD but above all it handles the application state machine
- The Background task is responsible for executing actions (contact or contactless) according the current state
- The Card Detection task is responsible for contact and/or contactless card discovery (as well as for running the OVR-HAL process when P2P communication over LLCP protocol is initiated)
- The Keypad task is responsible for detecting when a key is pressed

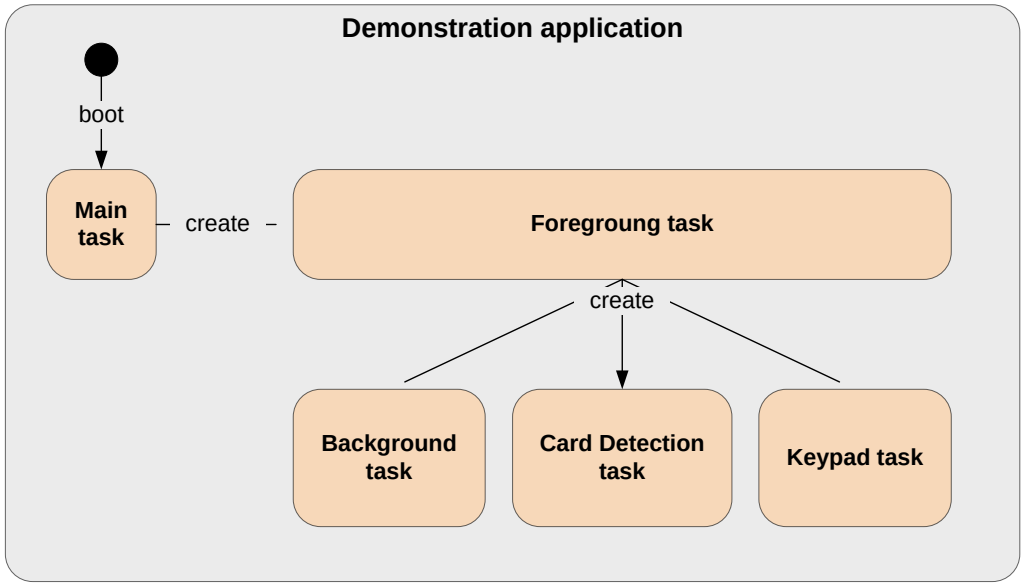

Fig 18. Demonstration application tasks

Tasks communicate together using message mechanism:

- The Foreground task indicates to the Background task which action needs to be performed according the current application state.
- The Background task indicates to the Foreground task when the action completes.
- The Card Detection task indicates to the Foreground task when a card has been discovered (and if data has to be displayed to the LCD in P2P specific case).

- The Keypad task forwards to the Foreground task information entered by the user on the keypad

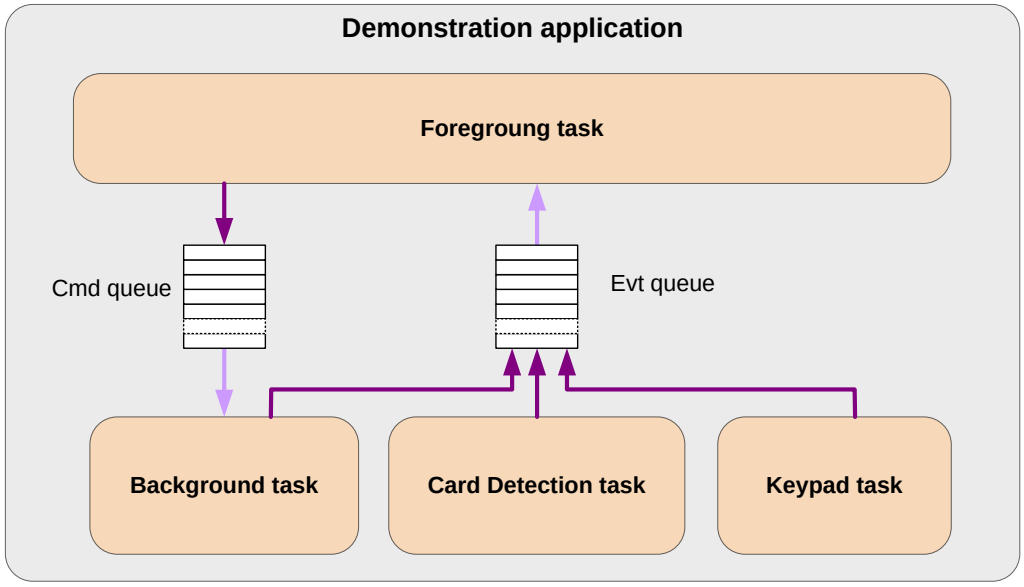

Fig 19. Tasks communication scheme

5.2 Application state machine

The state machine is handled by *mlsTUIForgroundProcess* function (*mlsTUIMan.c* file).

Each state is linked to:

- A “Draw” function (*mlsTUI**State**_Draw)* which updates the LCD screen, according the current state, and launch the execution of the related action
- An “Execute” function (*mlsTUI**State**_Execute*) which set the next state according to the current event received

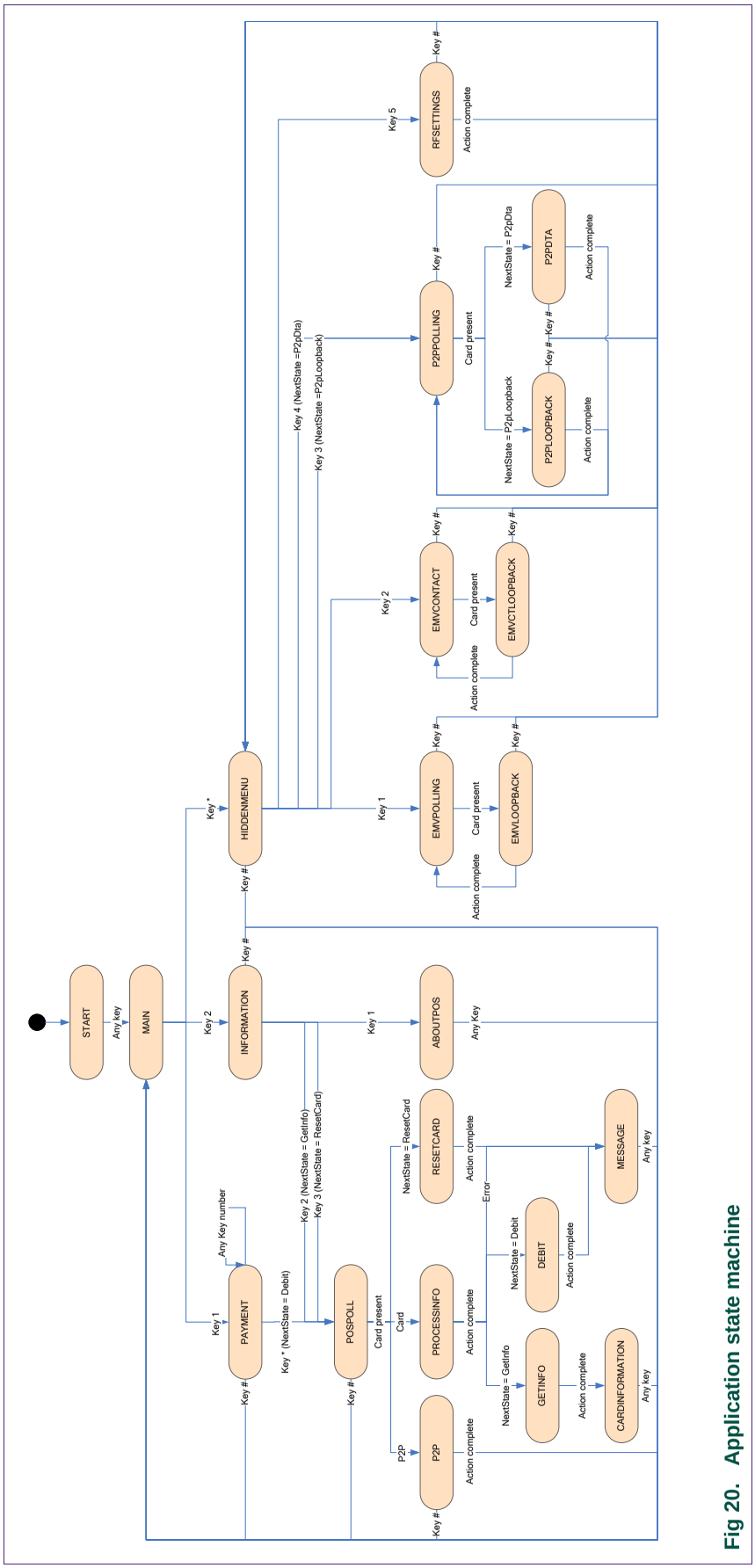

Fig 20. Application state machine

6. Legal information

6.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

6.3 Licenses

Purchase of NXP ICs with ISO/IEC 14443 type B functionality

This NXP Semiconductors IC is ISO/IEC 14443 Type B software enabled and is licensed under Innovatron's Contactless Card patents license for ISO/IEC 14443 B.

The license includes the right to use the IC in systems and/or end-user equipment.

RATP/Innovatron Technology

Purchase of NXP ICs with NFC technology

Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. A license for the patents portfolio of NXP B.V. for the NFC standards needs to be obtained at Via Licensing, the pool agent of the NFC Patent Pool, e-mail: info@vialicensing.com.

6.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

MIFARE — is a trademark of NXP B.V.

DESFire — is a trademark of NXP B.V.

7. List of figures

8. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 21 March 2012
211411
Document identifier: UM10493