

---

# **Louvre** *Documentation*

***Release I***

**Silica**

**Mar 16, 2017**



|          |                                |          |
|----------|--------------------------------|----------|
| <b>1</b> | <b>Introduction</b>            | <b>3</b> |
| 1.1      | Quick Start Guide . . . . .    | 4        |
| 1.1.1    | Unboxing . . . . .             | 4        |
| 1.1.2    | Power Mode . . . . .           | 5        |
| 1.1.3    | Powerless Mode . . . . .       | 9        |
| 1.2      | Development . . . . .          | 22       |
| 1.2.1    | Download LPCExpresso . . . . . | 23       |
| 1.2.2    | Firmware Projects . . . . .    | 25       |
| 1.3      | Board . . . . .                | 38       |
| 1.3.1    | Block Diagram . . . . .        | 38       |
| 1.4      | Appendix . . . . .             | 39       |



Version 1.0.01

Copyright (C)2016 Avnet Silica company

Date 11 May 2015





**Silica Louvre board** is an useful product to evaluate the capabilities of **NFC** with **NXP** technology. The Louvre board doesn't have any battery; when needed the power is supplied exclusively from the NFC antenna. The firmware provided will show the operation:

◦ **Powerless Mode** with two possible applications:

1. Display will show a picture updatable via smartphone, see in *Louvre Image Sender*
2. LPC11U37 firmware can be updated via smartphone, see in *Louvre Firmware Update*

◦ **Power Mode:**

User interface will be displayed on e-paper when connected to USB (only as power source), see in *Power Mode*

- Showing sensor data
- (S1, S2, S3, S4) will show the last touched capacitive buttons

In this documentation we show step by step the features of the firmware and how debug it.

---

**Note:** If you want quickly see how the demos work, go here *Quick Start Guide!*

---

For board schematics and source code go to [ArchiTech website](#).

### General Description

- Small form factor
- NXP NTAG I2C

- NXP Cortex M0 LPC11U37 Microcontroller
- 2.7 e-paper display
- SE98ATP temperature sensor
- Mini-USB connector
- Possibility to operate without battery
- Full size antenna to maximize energy harvesting
- Four capacitive buttons to enable user interaction

Contents:

## Quick Start Guide

In this page it is showed the unboxing of the Louvre and how to use the board with its software.

**Warning:** The applications have been tested with a nexus 5.

## Unboxing

The product is supplied with the box like this:



And this is the content of box:



---

**Note:** The board itself has been programmed with the *Bootloader Firmware* and the *Application Firmware*.

---

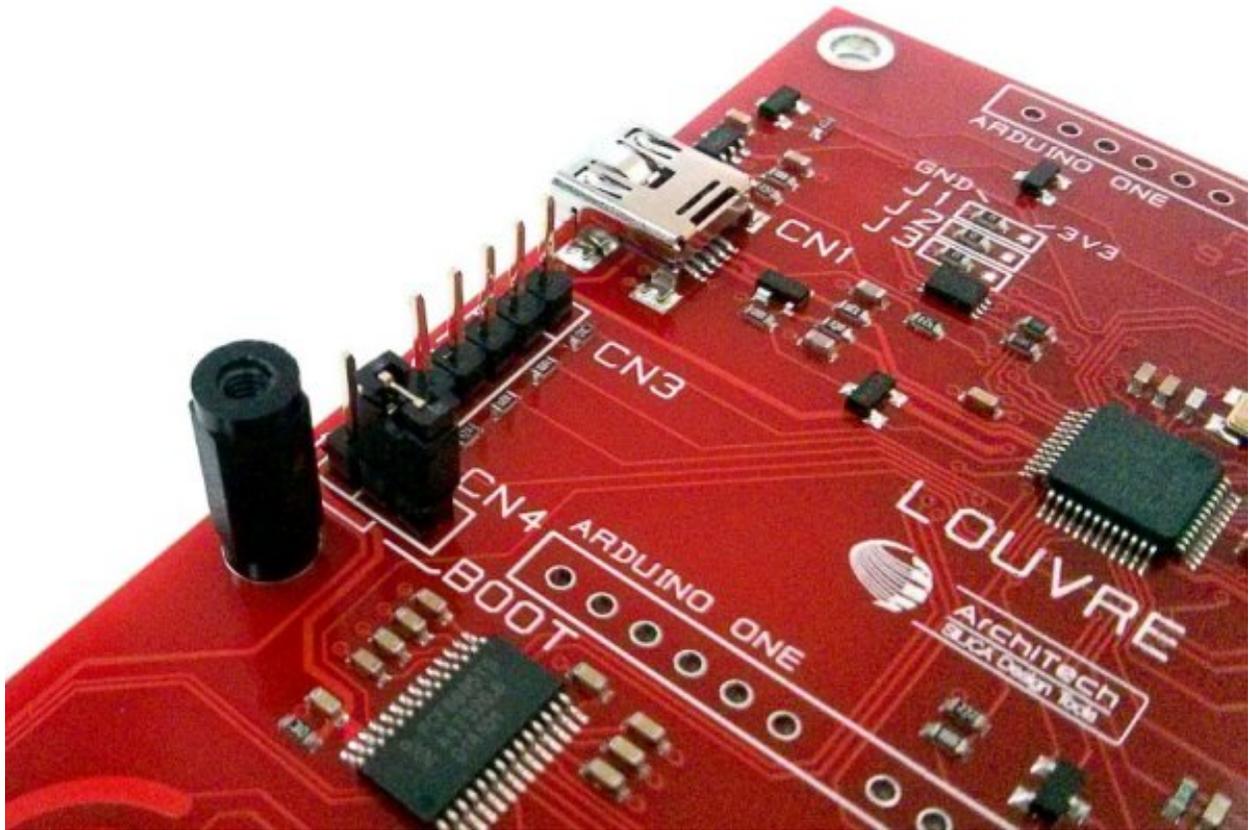
The board can run two possible modalities:

- *Power Mode*: it runs supplied by mini-USB cable
- *powerless\_mode*: it runs supplied by NFC antenna

## Power Mode

In order to run the **Power mode** interface it is needed supply the board via a mini usb cable:

1. Check the jumper **CN4** is setted correctly:



2. Insert mini-USB cable from PC to the CN1 connector of the board.



3. The Louvre display on the e-paper screen the temperature read from **SE98ATP** sensor and the last capacitive button pressed.



4. If you press a capacitive button S1, S2, S3 or S4 this will displayed on the screen.



## Powerless Mode

The powerless mode works with two different Android applications:

- *Louvre Image Sender*: it sends images to the Louvre and the board will show them.
- *Louvre Firmware Update*: it sends a newer *Application Firmware* to the Louvre board.

## Louvre Image Sender

With this application :

- use **NFC** to store an image on the **NTAG IIC** nonvolatile memory mounted on the Louvre board
- using power harnessed from the **NFC** field, the Louvre microcontroller will read the image from **NTAG** using the **IIC** interface and will update the e-paper display
- which shows the image even when the Louvre is not powered up.

---

### Note:

- It is required have to enabled the NFC feature on the Android device
  - If you have a cover on your phone, please remove it.
  - The source project is downloadable on [github](#)
  - The application note [AN11367](#) helps how write Android application with NFC technology
-

1. Download and install the Android Application via **Google Play Store**, you can use the Qr Code below or search in the [Google Play Store](#) with the name [Louvre Image Sender](#).



2. Start the application **LouvreImgSender** on the android device:



3. The application displays the images that you can send to the Louvre. Select an image do you like:



4. Lay down the tablet on the Louvre. The Android device must cover all the antenna area of the Louvre.



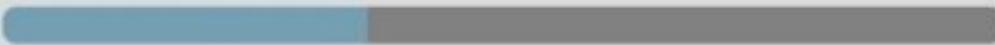
5. The data will be transferred



Select an image then tap the tag to write on it.



Sending selected image ...



Block 632 of 1735



6. Once the data transfer is done a message is reported on the device. Do not remove the device from the Louvre until the e-paper display doesn't show the selected image.



7. Finally the display will show the image selected



## Louvre Firmware Update

With this software a new *Application Firmware* is downloaded and programmed into the Louvre microcontroller, always getting energy only from the NFC. From this application it is possible also retrieve some tag settings.

---

### Note:

- It is required the bootloader installed on the Louvre microcontroller to enable this download even the part is in-field.
- It is required have to enabled the NFC feature on the Android device
- If you have a cover on your phone, please remove it.
- The java source project is downloadable on [github](#)
- The application note [AN11367](#) helps how write Android application with NFC technology

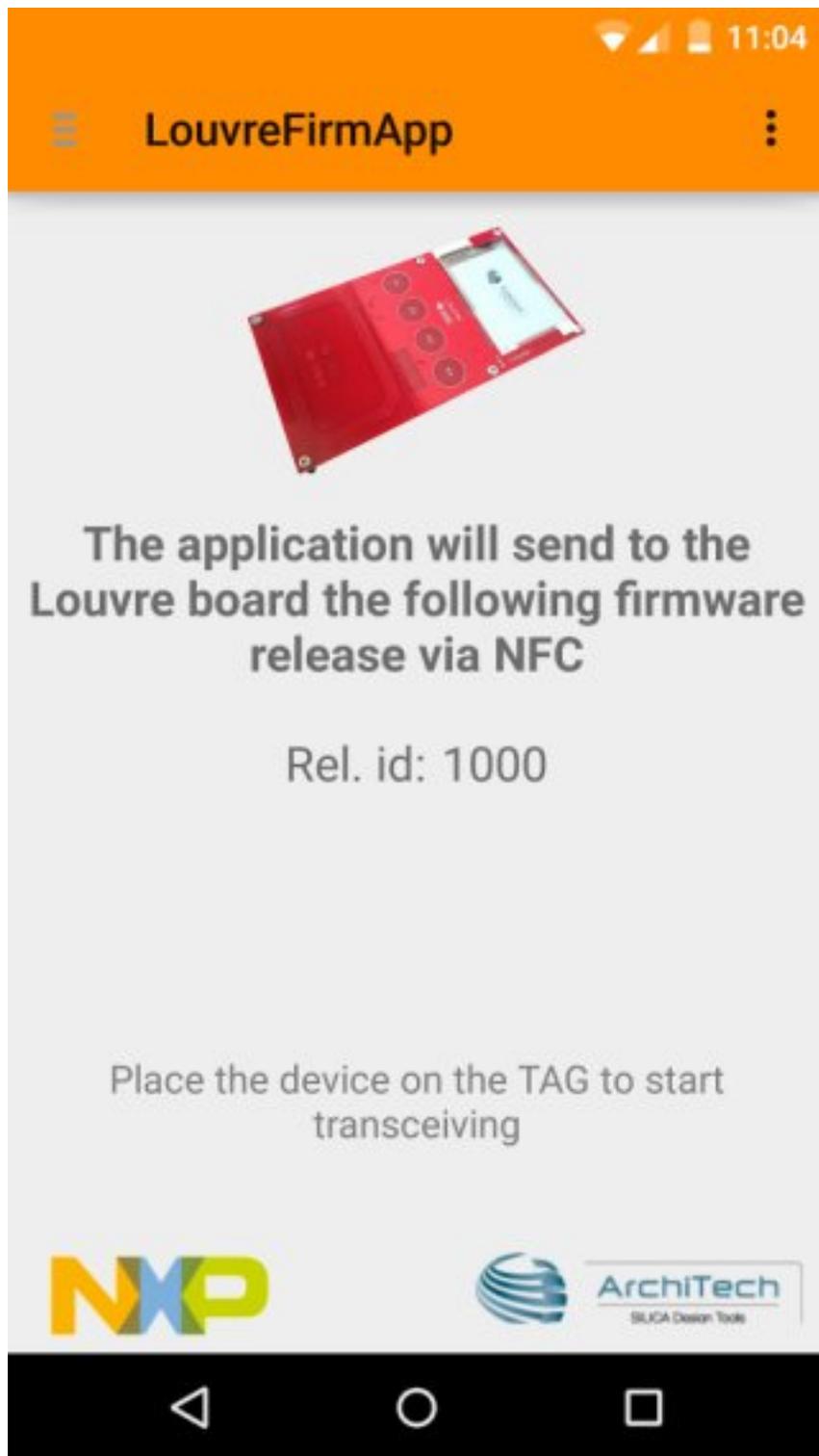
- 
1. Download and install the Android Application via **Google Play Store**, you can use the **QR Code** below or search in the **Google Play Store** with the name **Louvre Firmware Update**.



2. Start the application **LouvreFirmUpdate** on the android device:



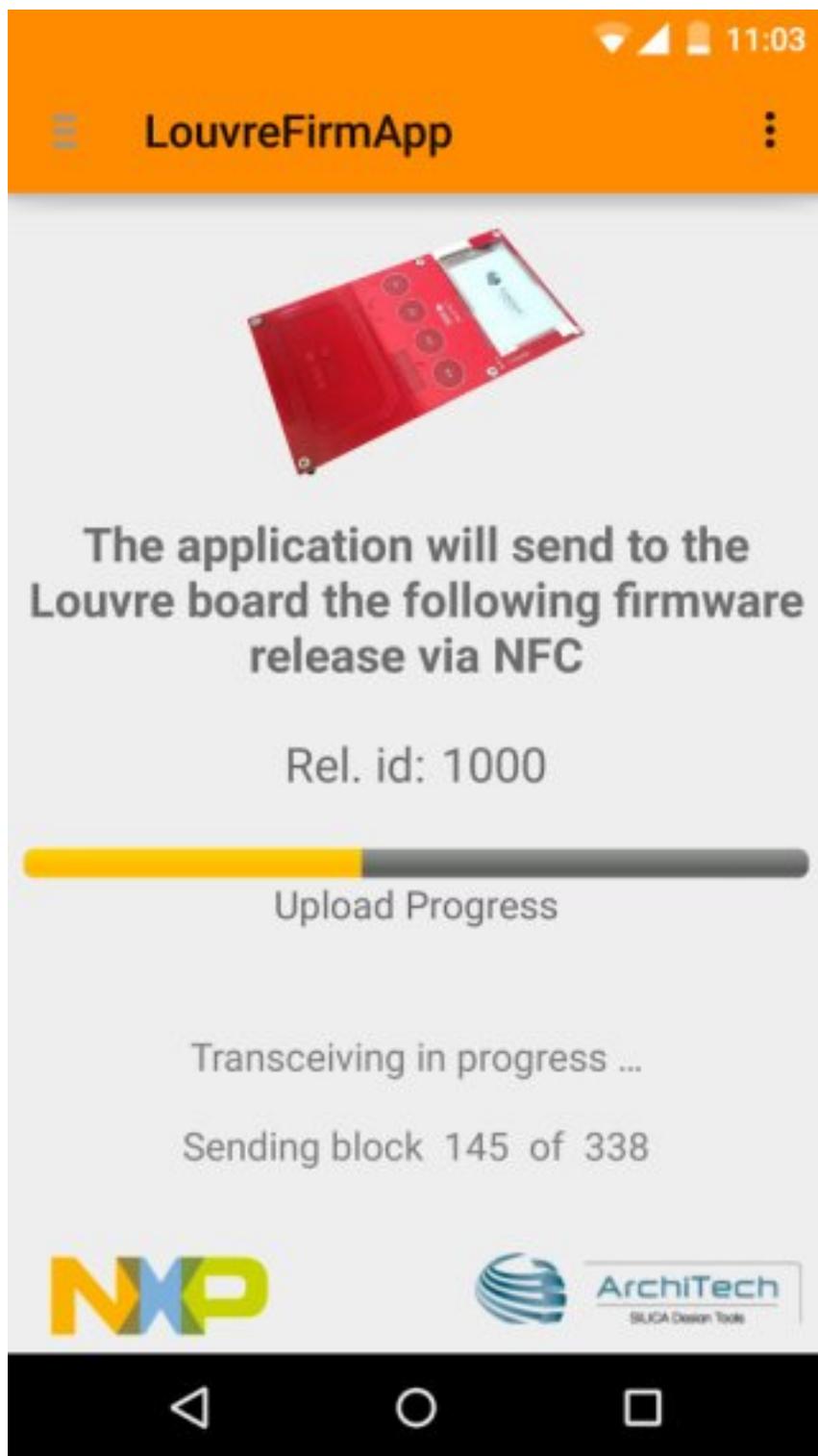
3. The application shows the main screen:



4. Lay down the tablet on the Louvre. The Android device must cover all the antenna area of the Louvre.



5. The new firmware will be transferred:



6. Once the firmware transfer is done a message is reported on the device.



Now the board is programmed with a newer version of the *Application Firmware*. In order to have always the last application firmware, keep updated the **Firmware Download Application** via [Google Play Store](#).

## Development

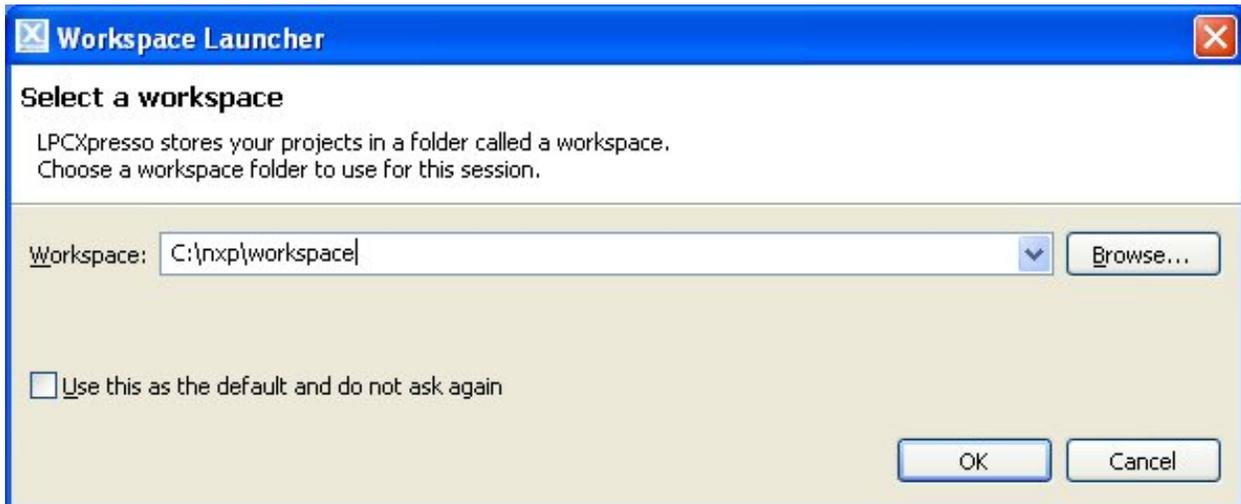
To develop on Louvre board are required:

- Software:

- LPCXpresso
- Application firmware project, compatible with bootloadaer
- Bootloader firmware project
- Hardware:
  - Louvre Board
  - NXP LPC-Link (version 1 or 2), please refer to [embeddedartists](#) website. In this guide we have tested both the LPC-Link 1 of this [board](#) and the LPC-Link 2.

## Download LPCExpresso

The developement software used is LPCxpresso, we suggest to install at the least version 7.3. This application is freely downloadable from lpcware site. Go to [lpcxpresso](#) webpage Install the LPCxpresso following default procedure. In this guide the SDK is installed in the default path `C:\nxp` and we refer to this in our example. After the installation you need to register in order to activate the **Free Edition** of the LPCXpresso. Launch the SDK selecting as workspace folder `C:\nxp\workspace`.



Select from menu *Help -> Activate -> Create serial number and register (Free Edition)*. Check the options *Open in external browser* and *Copy Serial Number to clipboard* then press *Ok* button.



From the website you need to register and insert the serial number. It will display your activation code.

## Submission #35505

[Home](#) » [LPCxpresso Key Activation](#) » [Submissions](#)

Thank you for registering your LPCxpresso product. YOUR ACTIVATION CODE IS DISPLAYED BELOW - activation code in a few minutes.

Your product registration history can be located here or by selecting the 'My LPCxpresso activations' me

You must be logged in with the account you used to register your product to view activation key history

Registration Information

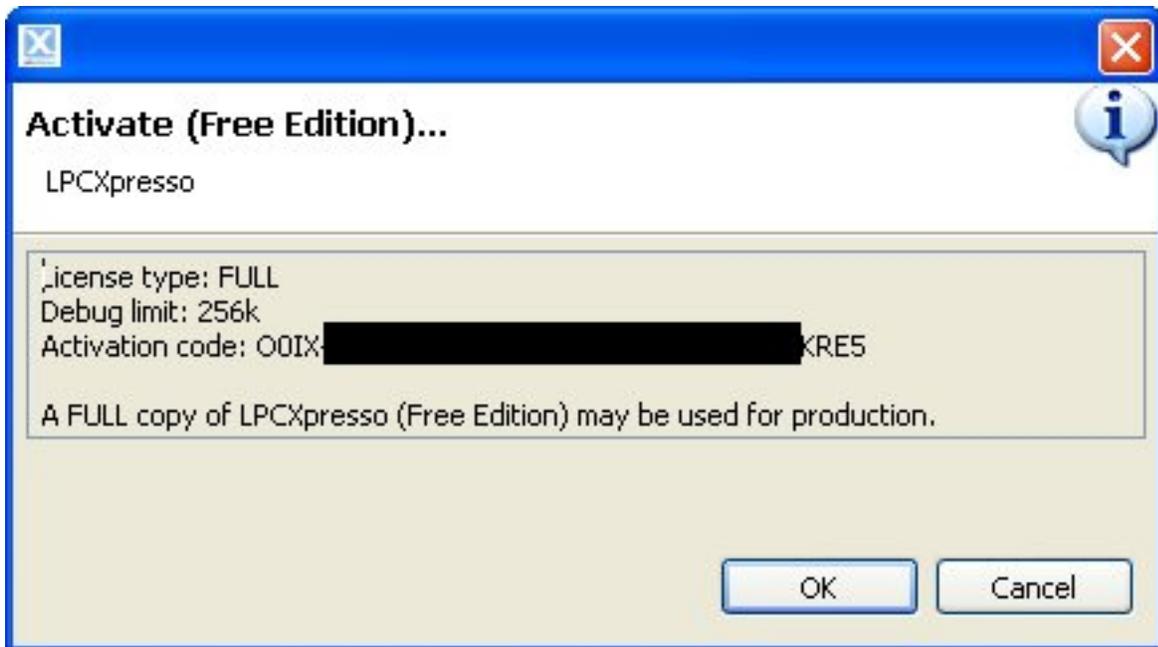
**Serial Number:**

DRK3 [REDACTED]-F4A1

**LPCxpresso Activation Key:**

OOIX [REDACTED]-KRE5

The activation code must be inserted into the window-form selectionable from the menu *Help -> Activate -> Activate (Free Edition)*



If all is done correctly now you are able to deploy with the Free Edition LPCXpresso, permitting you to debug up to 256KB.

## Firmware Projects

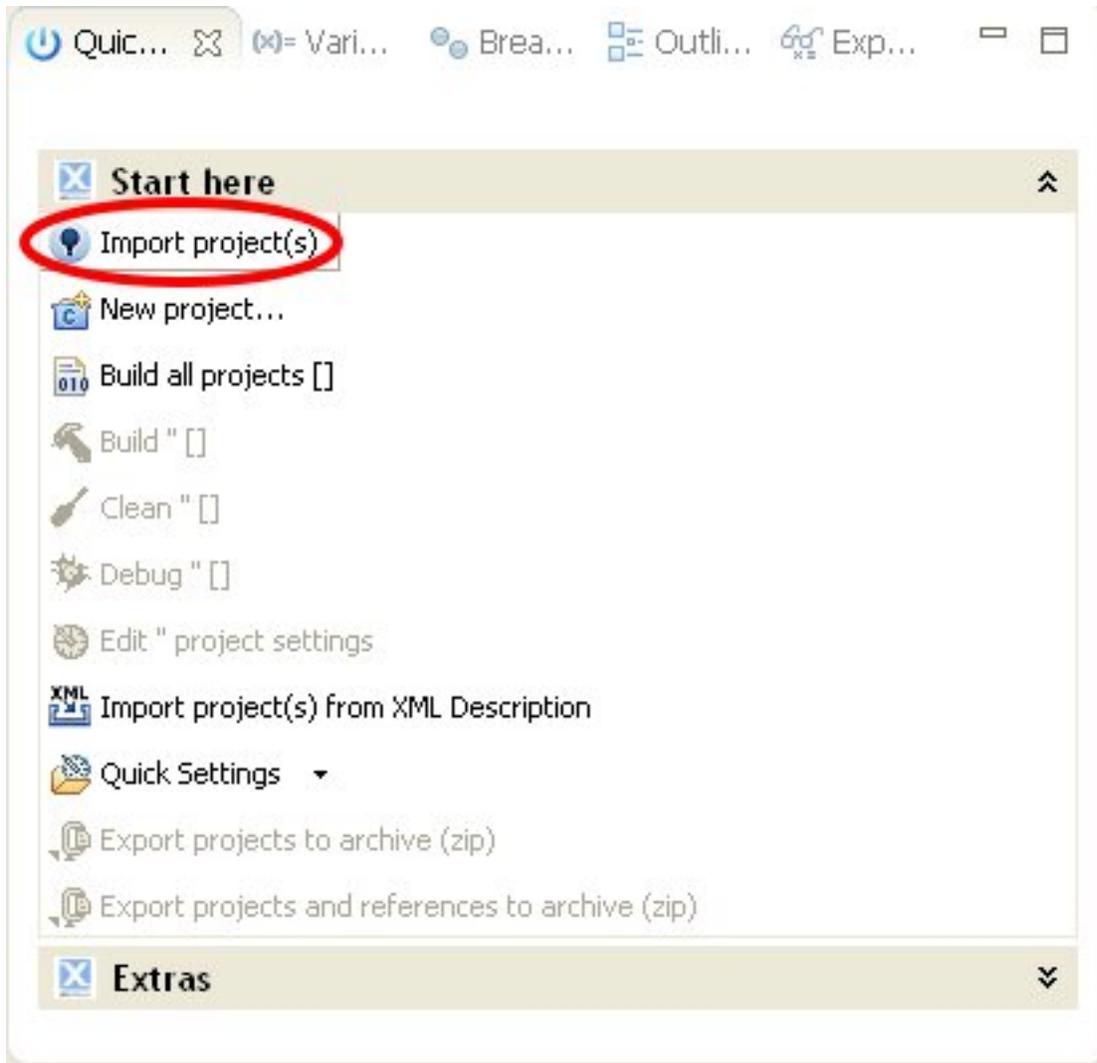
To get the last source of the firmware and bootloader, go to [architechboards website](#). This site provides software and documentation needed to develop with the Louvre board. Save it in your bookmarks to be updated with the new releases. In the homepage select the **Louvre board** in the **Products** menu.

## Application Firmware

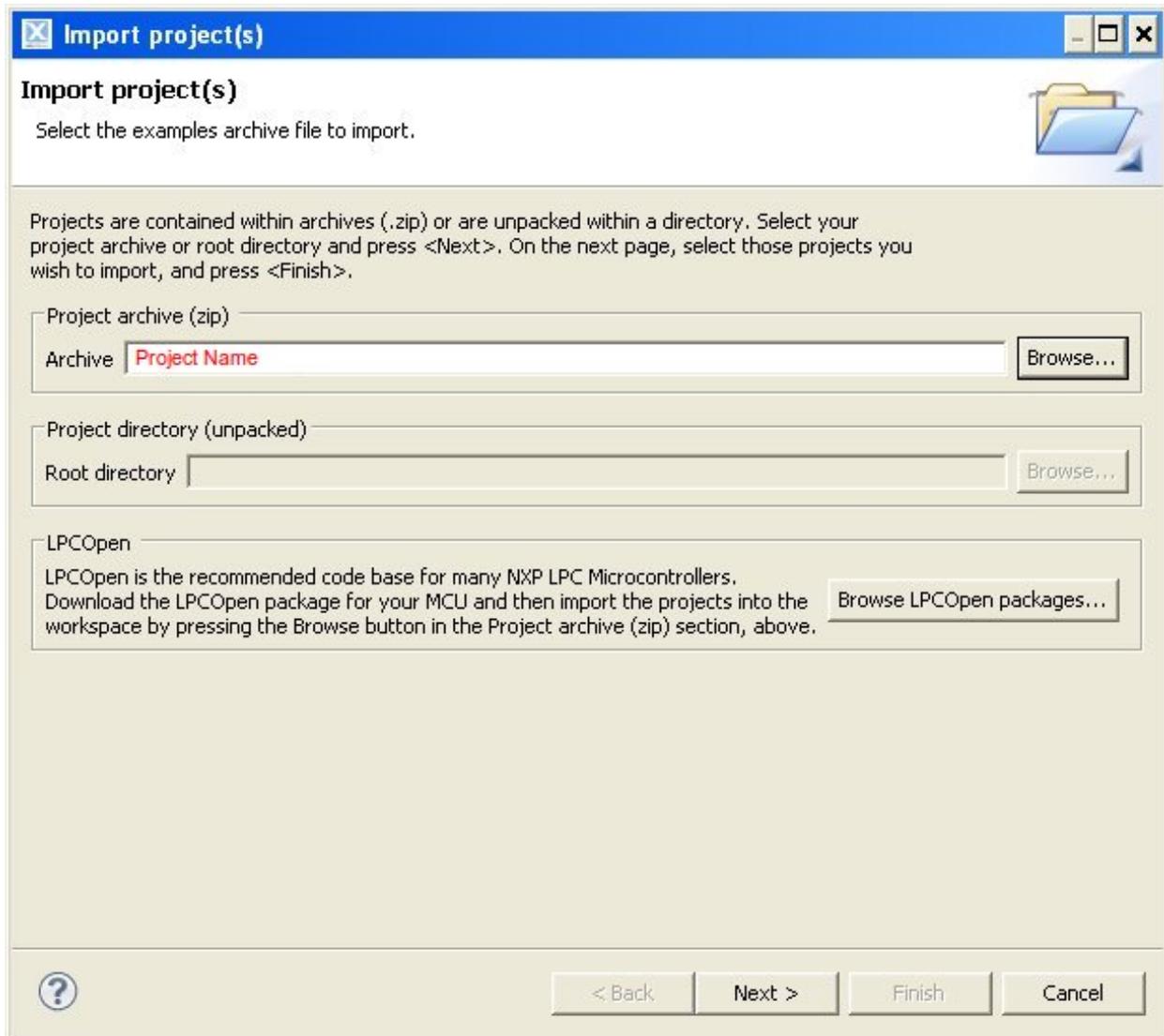
This is the software which permits to the Louvre to retrieve the images on the Louvre board via the Android application called *Louvre Image Sender* and run the *Power Mode* interface.

### Import Application Project

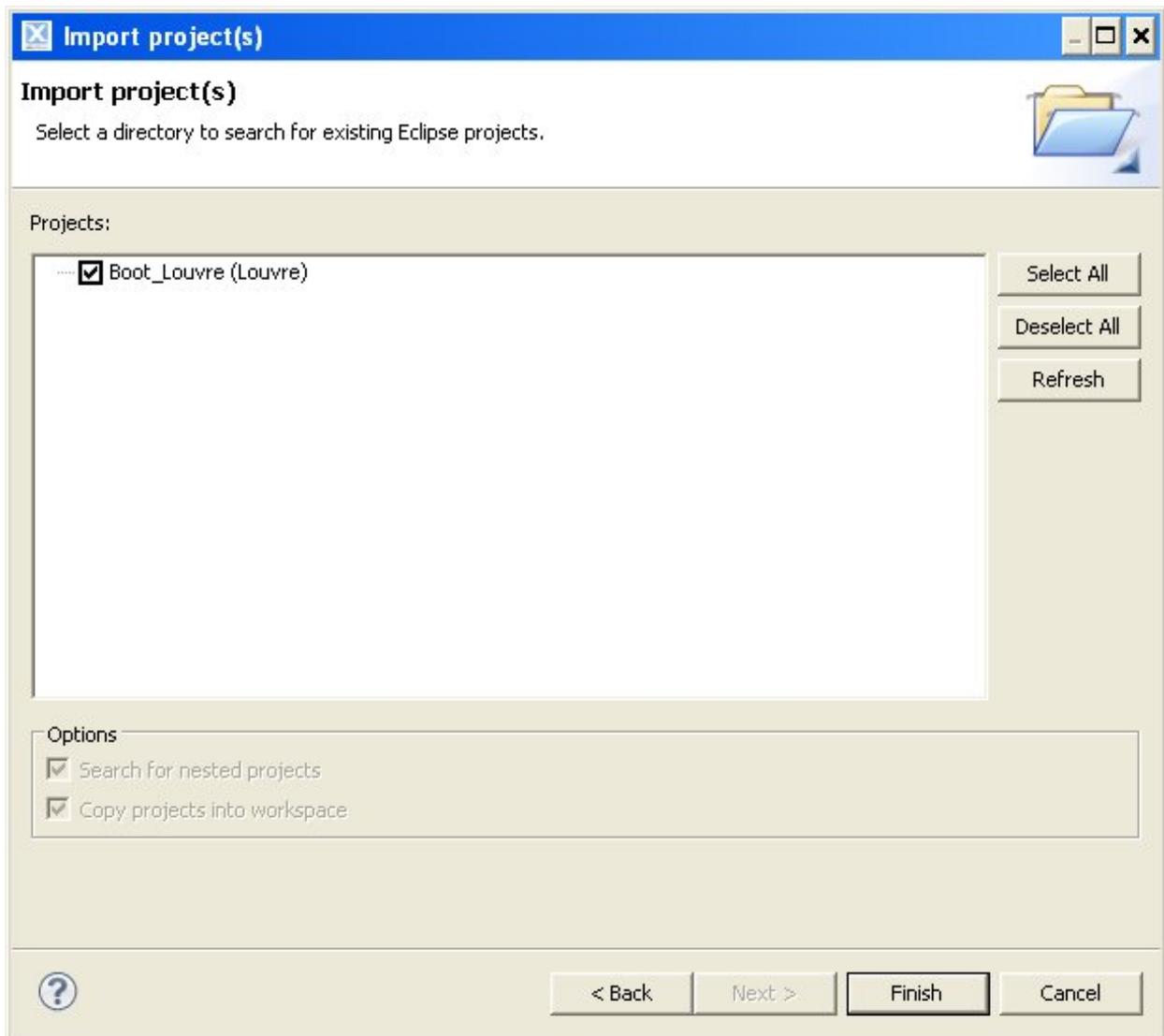
You can import the project directly from the packed file downloaded before. In order to do this, press on *Import Project*:



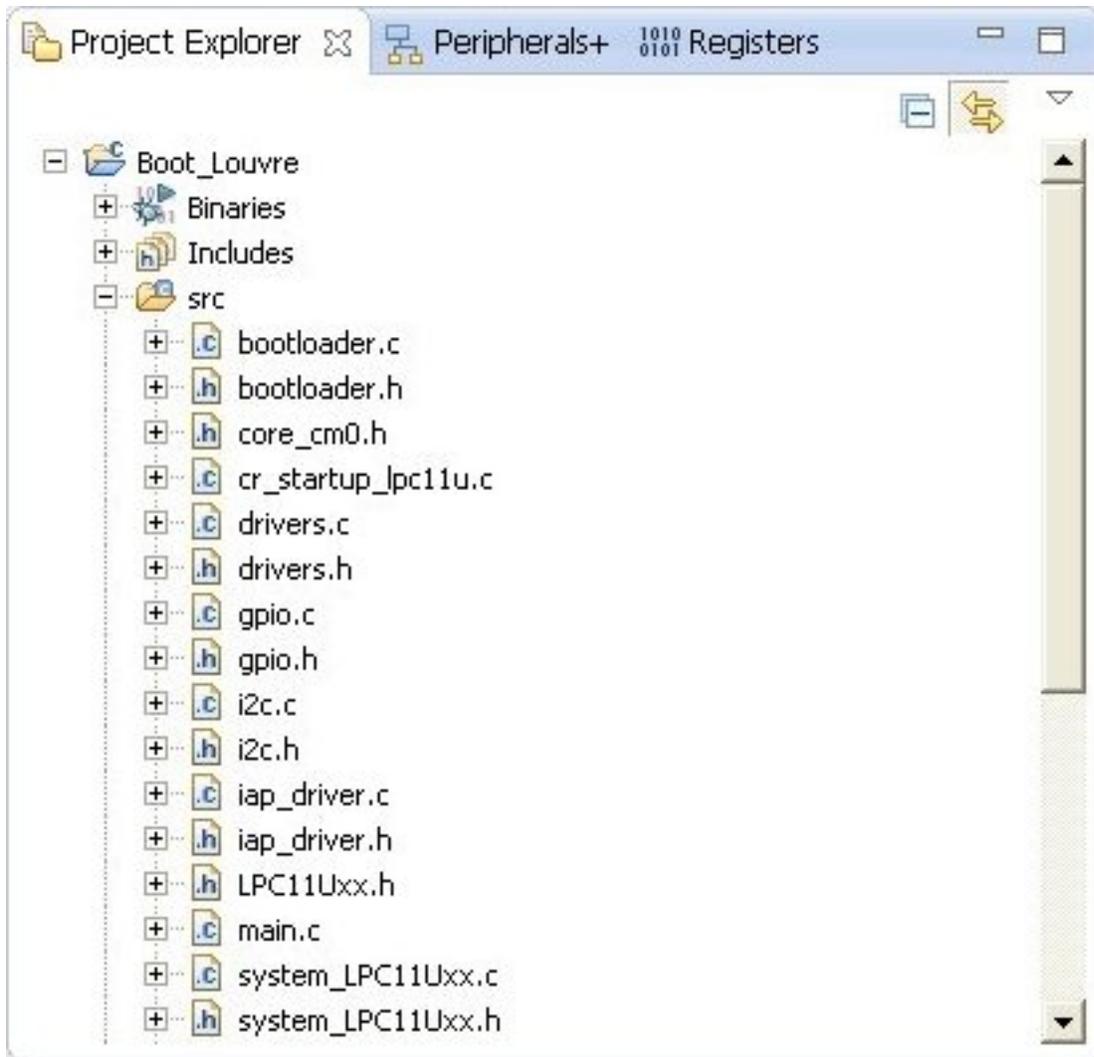
Then browse and select the **ArchiTech\_Louvre\_Firmware.zip** file, click on *next* button:



Select all and click another time on *next* button:



Now you have the project loaded:



Project files will be moved under the workspace directory.

## Download into Louvre

In order to download the firmware into Louvre board you have to:

1. Recompile the **Louvre Firmware Update** Android application with the firmware.hex file built previously. You can find the source project on [github](#). To change the downloaded firmware, the file *firmware.hex* must be changed under the [this folder](#).
2. Use the *Louvre Firmware Update* to upload the Louvre board.

---

**Note:** It is not possible to debug the application due to the relocation of memory in order to be compatible with the bootloader firmware.

---

## Firmware Details

All libraries code used is imported directly into the project: *i2c.c*, *ssp.c*, *gpio.c*, *timer.c* and *adc.c*.  
 The file *adc.c* is not used.  
 The code for the e-paper is in the modules *eink.c* and *fonts.c*.

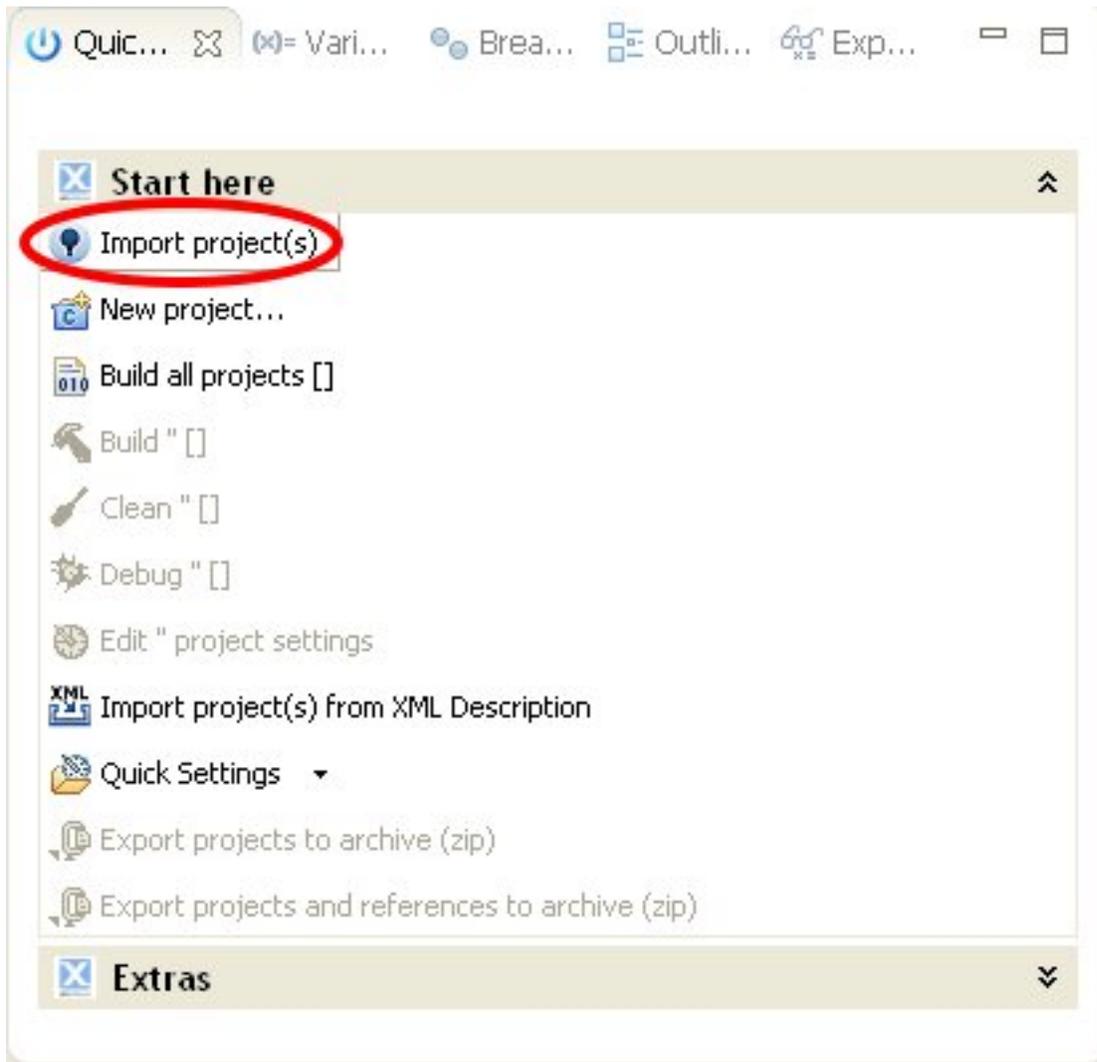
The module `driver.c` is for manage temperature sensor, capacitive buttons and nfc protocol.  
To have more details, the source code is provided with comments.

## Bootloader Firmware

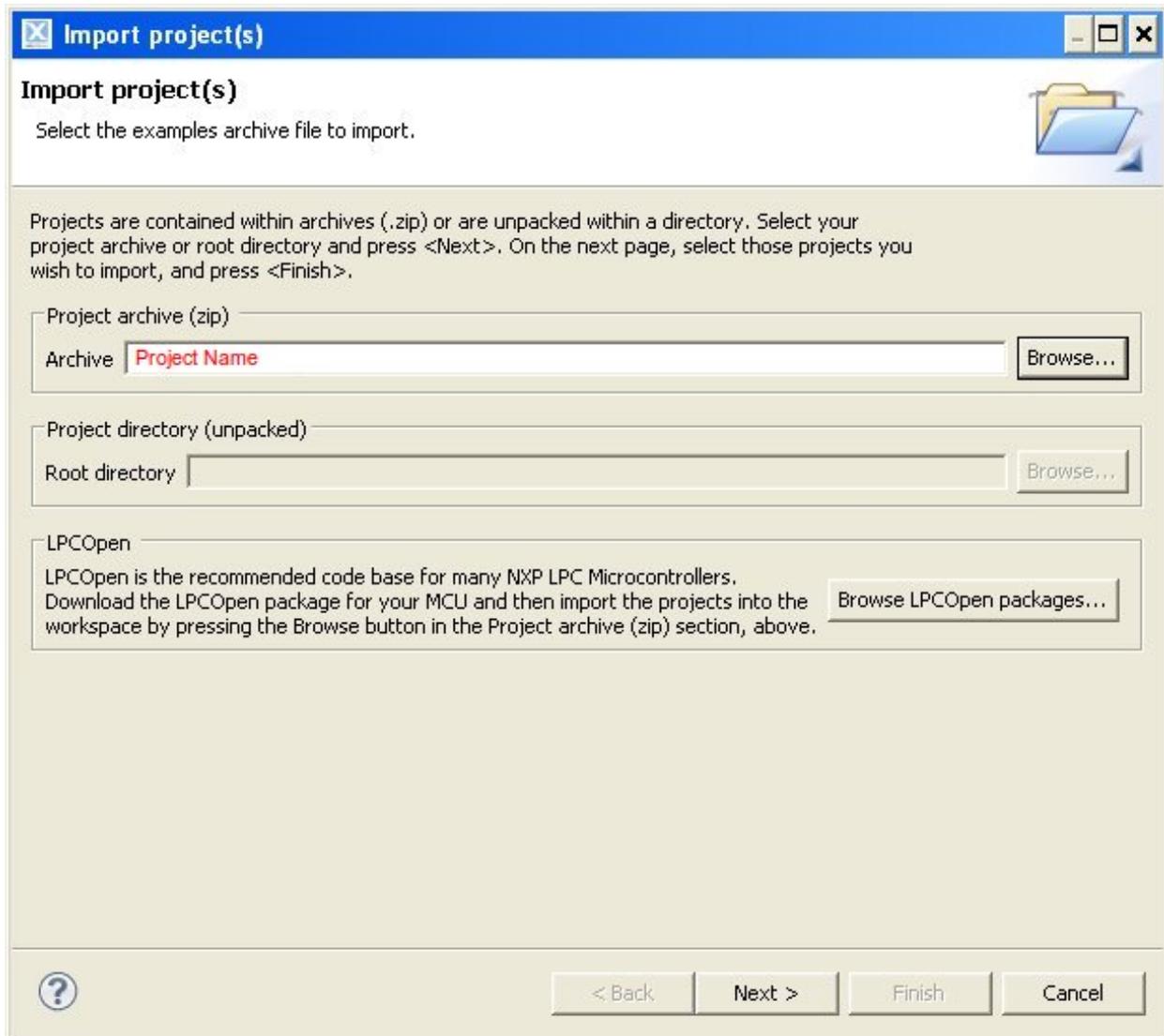
This is the software which permits to the Louvre to retrieve a newer *Application Firmware* via the Android application called *Louvre Firmware Update*.

### Import Bootloader Project

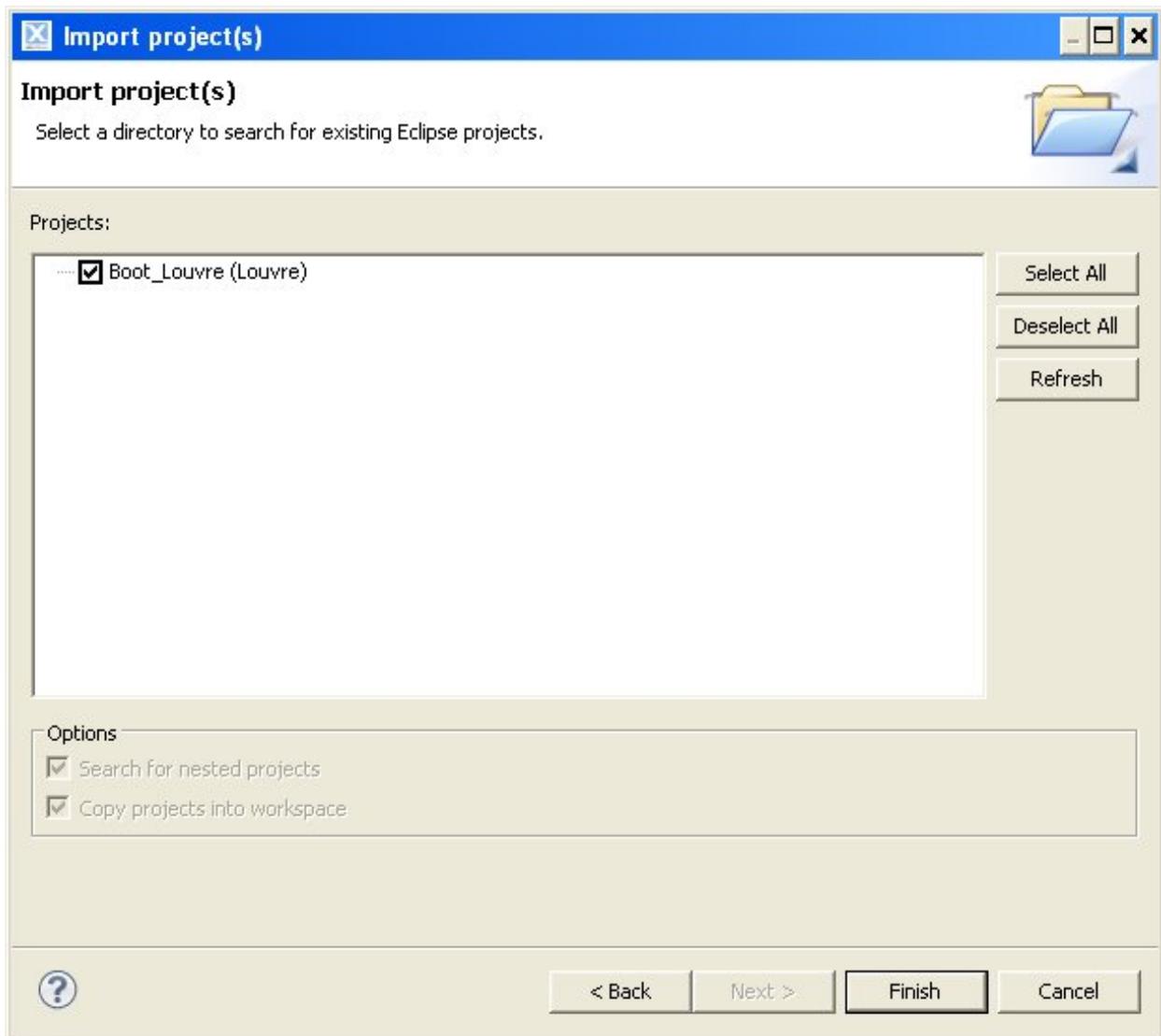
You can import the project directly from the packed file downloaded before. In order to do this, press on *Import Project*:



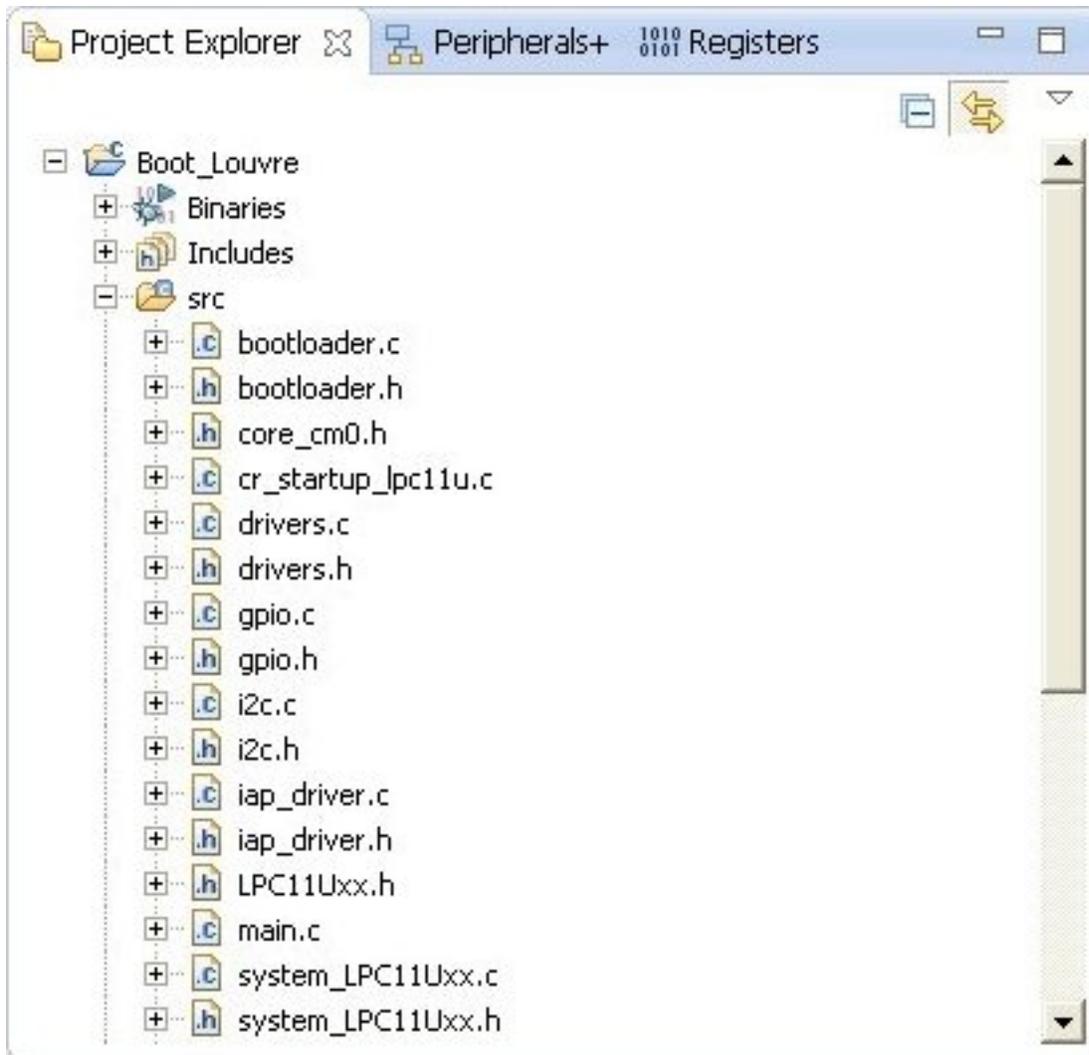
Then browse and select the **ArchiTech\_Louvre\_Boot\_Firmware.zip** file, click on *next* button:



Select all and click another time on *next* button:



Now you have the project loaded:



Project files will be moved under the workspace directory.

---

**Important:**

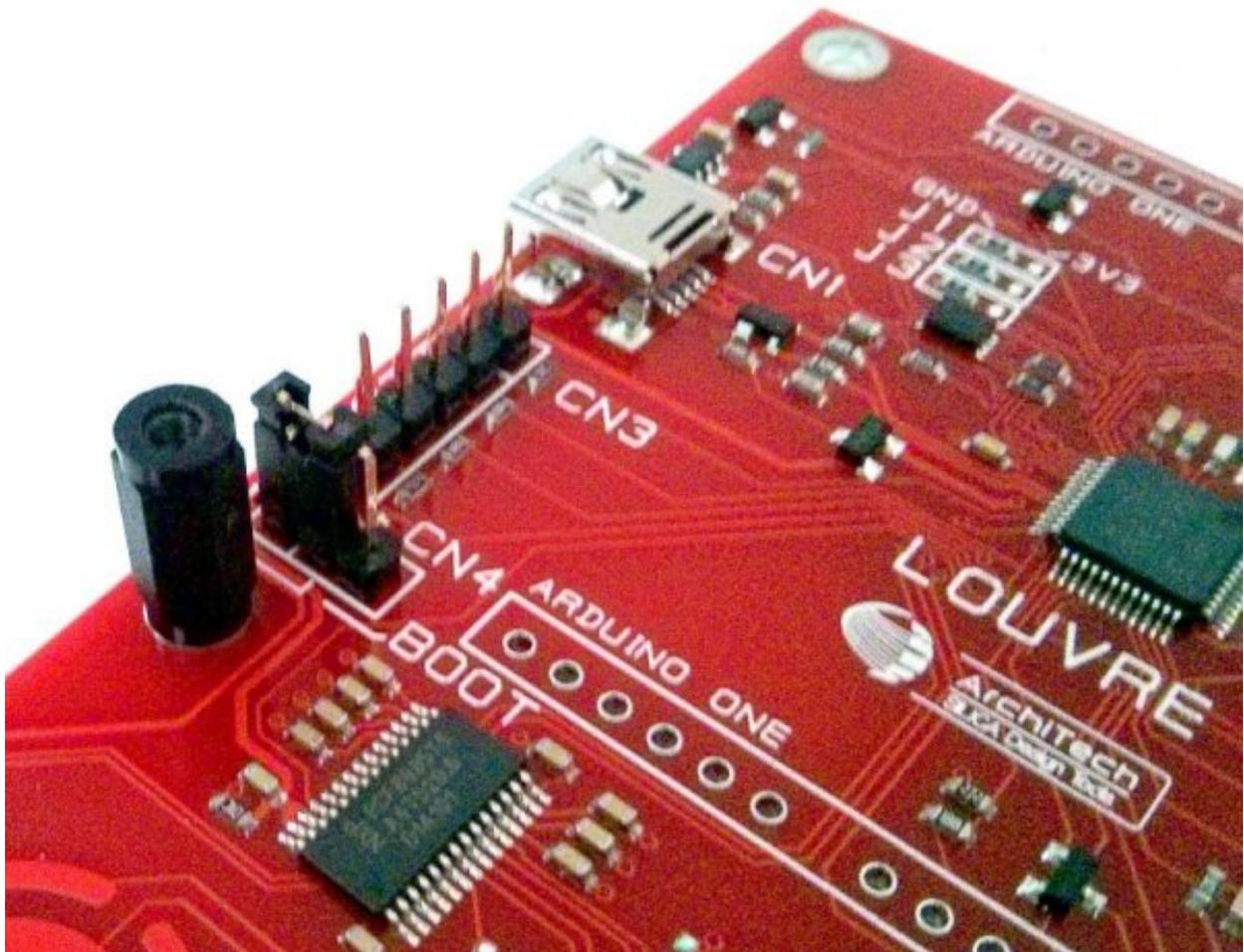
Before proceed, copy (from the *bootloader source project*, in the folder *padto*) the **padto.exe** file under the path:  
`C:\nxp\LPCXpresso_7.6.2_326\lpcxpresso\bin`

---

**Download into Louvre**

Once you have compiled the project you have the **firmware.bin** file output into *debug* folder. To upload it in the Louvre board you have two choice. **The first, the easiest** is with mini-USB cable:

1. Check the jumper **CN4** is setted correctly:



2. it's insert mini-usb cable from PC to the **CN1** connector of the board.



3. The PC will see the Louvre board as a storage memory, delete the old **firmware.bin** file and copy the newer **firmware.bin** file into the mass storage.

The second choice is using the **LPC-Link 1** or **LPC-Link 2**:

- **LPC-Link 1**

1. make sure that you have connected the LPC-Link to PC. Connect with a cable the **J4** LPC-Link connector to the **CN3** connector of the Louvre board. Here the connection scheme:

| pin J4 | pin CN3 | Signal Names    |
|--------|---------|-----------------|
| 3      | 2       | JTAG_TMS_SWDIO  |
| 5      | 4       | JTAG_TCLK_SWCLK |
| 11     | 1       | JTAG_RESET      |
| 15     | 3       | GND             |

2. To Flash the project you need to build it, from the menu go to *Project -> Build All*. The console window will display the compilation.

```

CDT Build Console [SW]
make --no-print-directory post-build
Performing post-build steps
arm-none-eabi-size SW.axf; # arm-none-eabi-objcopy -O ihex SW.axf SW.hex ;
  text   data    bss     dec     hex filename
 17628    4     6596   24228   5ea4 SW.axf

11:36:49 Build Finished (took 5s.829ms)
    
```

3. Then click on the *processor* icon.



- **LPC-Link 2**

1. Connect **LPC-Link 2** with the pc

---

**Note:** **LPC-Link 2** can draw a significant amount of current from the usb. The best solution is to connect LPC-Link 2 to a powered hub.

---

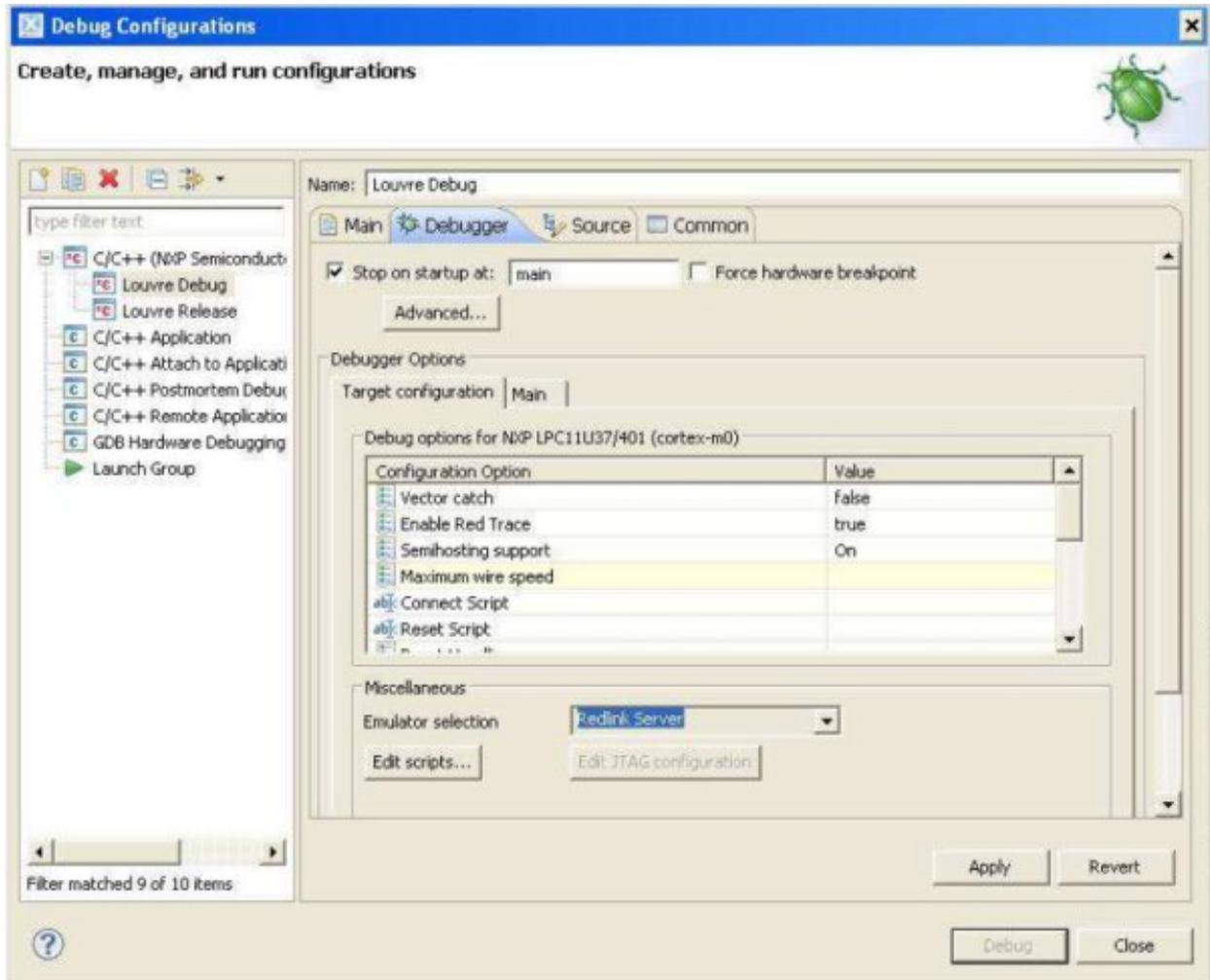
2. If drivers are required choose the automatic sequence of installation

3. Connect LPC-Link 2 with the target with a cable the **J4** LPC-Link connector to the **CN3** connector of the Louvre board. Here the connection scheme:

| pin J4      | pin CN3 | Signal Names    |
|-------------|---------|-----------------|
| 2           | 2       | JTAG_TMS_SWDIO  |
| 4           | 4       | JTAG_TCLK_SWCLK |
| 10          | 1       | JTAG_RESET      |
| 3 or 5 or 9 | 3       | GND             |

4. Open LPCXpresso and load the project then *run->debug configurations*, select *debugger* tab

5. On *emulator selection* voice choose *Redlink server*



6. Click *Apply* and *Close* buttons, then click on the *processor* icon:



### Bootloader Details

The bootloader is the first program executed, located in the first two blocks of the flash memory (**0x0000 - 0x1FFF**) for a total size of **8K** bytes. It controls the presence of a NFC signal with a new valid application; if so, received data are written in flash. After, it calculates the validity of the CRC flash memory, if it is valid then copies the applicative interrupt vectors ram memory (**0x1000000 - 0x100000bf**) and launches it. If the application is not valid then it checks continuously NFC signal waiting a new program to download.

#### Flash memory map:

**0000 00BF** bootloader code vector table

**00C0 1FFB** bootloader code

**1FFC 1FFF** software version

#### Ram memory map:

**0x10000000 0x100000BF** applicative interrupt vectors table  
**0x100000C0** bootloader/applicative ram memory

**Applicative:**

**Flash memory map:**

**2000 2003** applicative checksum (used by bootloader to verify the code)  
**2004 2007** applicative length  
**2008 200B** applicative version  
**200C 20BF** applicative code vector table (will be copied in the ram memory)  
**20C0** applicative code

**Ram memory map:**

**0x10000000 0x100000BF** applicative interrupt vectors table  
**0x100000C0** ram memory

---

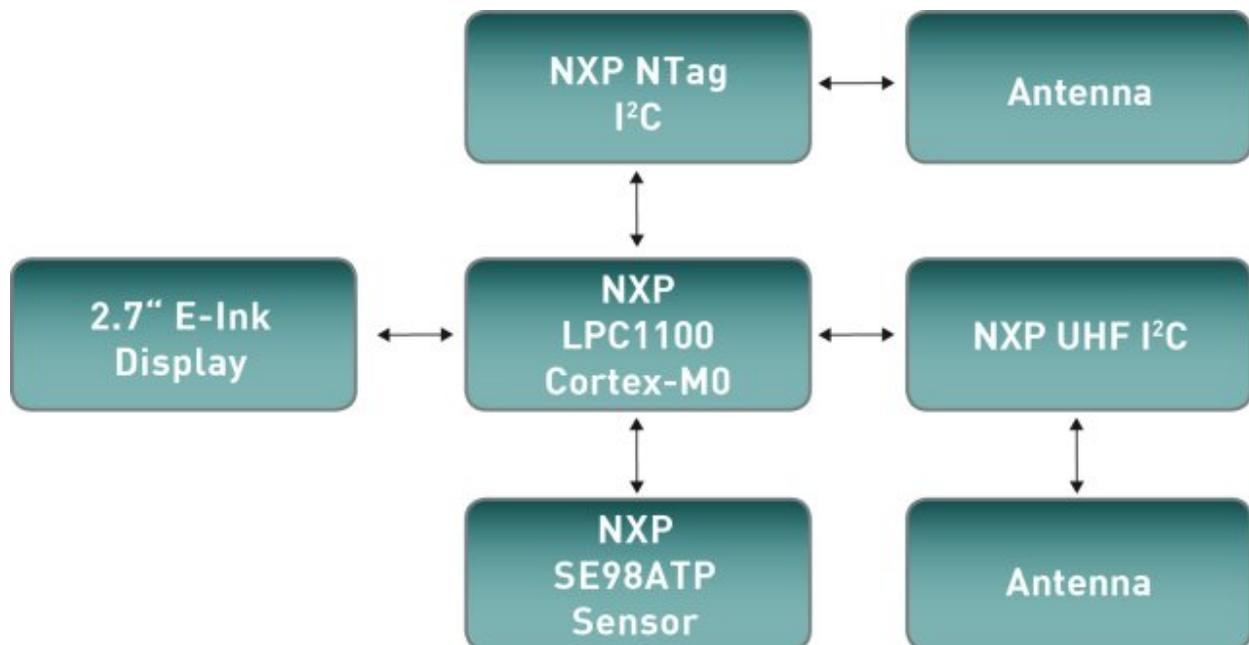
**Important:**

- Due to the optimization, it is not possible debugging the bootloader
  - You can get the info about the protocol used by bootloader from [ArchiTech website](#), the document is named **LouvreBootloaderProtocol.pdf**
- 

## Board

### Block Diagram

The Louvre board demo is fitted with these devices:



- NXP Cortex M0 LPC11U37 Microcontroller
- SE98ATP temperature sensor
- NXP NTAG: The NTAG family has been developed by NXP Semiconductors as standard NFC tag ICs to be used in mass market applications such as retail, gaming and consumer electronics, in combination with NFC devices or NFC compliant Proximity Coupling Devices.
- 2.7" e-paper EM027BS013 from Pervasive Displays: The 2.7" panel has resolution of 264 x 176 (117 dpi).
- NXP UHF. not used by firmware
- miniusb: used to power supply the board (see demo 2)
- jumper CN4: used for boot mode of the LPC11U34 (see upgrade section)

To download the schematics please visit [architechboards Louvre](#) webpage.

## Appendix

- **Louvre Image Sender** is the Android application which sends images to the Louvre board.
- **Louvre Firmware Update** is the Android application which sends a newer **Application Firmware** to the Louvre board.
- **Application Firmware** is the software which permits to the Louvre to retrieve the images on the Louvre board via the Android application called **Louvre Image Sender** and run the power mode interface.
- **Bootloader Firmware** is the software which permits to the Louvre to retrieve a newer **Application Firmware** via the Android application called **Louvre Firmware Update**.
- **Play Store** it is the google app store (or app marketplace), it is a digital distribution platform for mobile apps.
- **APK** stand for Android application PacKage and it is the package format used to distribute and install application software onto Android operating system.
- **e-paper** is a specific proprietary type of electronic paper manufactured by [Pervasive Display](#)
- **NTAG** The NTAG family has been developed by NXP Semiconductors as standard NFC tag ICs to be used in mass market applications such as retail, gaming and consumer electronics, in combination with NFC devices or NFC compliant Proximity Coupling Devices. See [NXP NFC with NTAG page](#).
- **LPC** NXP's portfolio of 32-bit LPC microcontrollers builds on 10 years of leadership and includes more than 330 options that cover every application class, from entry-level designs to high-end systems that run Linux OS. See [LPC](#).



## A

Android Application - Louvre Firmware Update, 16

Android Application - Louvre Image Sender, 9

Appendix, 39

## B

Board Details, 38

## C

Contents, 4

## D

Development, 22

Download LPCExpresso, 23

## I

Introduction, 1

## L

LPC-Link 1 connections, 35

LPC-Link 2 connections, 36

## P

Power Mode, 5

Powerless Mode, 9

## Q

Quick Start Guide, 4

## S

Source - Application Firmware, 25

Source - Bootloader Firmware, 30

## U

Unboxing, 4