

A close up of a sign

Description automatically generated

Inplay Stellar Gateway

SW User Guild

***Draft***

Revision History

|  |  |  |
| --- | --- | --- |
| **Ver** | **Notes** | **Update Date** |
| Pre. Draft | Initial draft | 03/28/2024 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Contents**

[Stellar Gateway Board overview and PIN Configuration 3](#_Toc5211)

[AT32F413 software download guild 3](#_Toc4452)

[IN612L Firmware Download Guild 6](#_Toc16938)

[AT32F413 Sample Project 9](#_Toc22436)

[Attachement 11](#_Toc385)

# Stellar Gateway Board overview and PIN Configuration

As shown in Figure 1, Stellar Gateway consists of two parts: gateway module and carry board. Carry board contains PCIE socket, USB type-C connector, as well as JTAG and SPI/I2C pins from AT32.

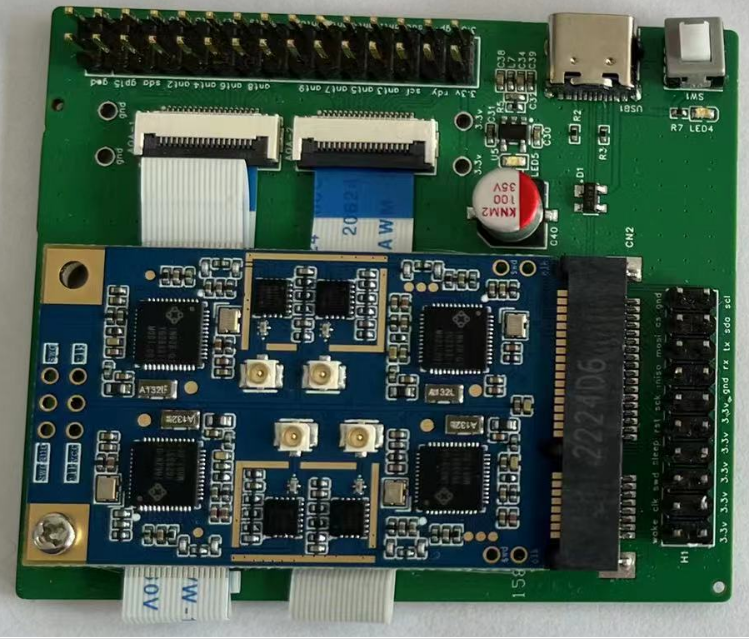


Figure 1: Stella Gateway connector pin definition

# AT32F413 software download guild

The first thing to do is to program the AT32 firmware, all subsequent operations rely on the AT32 firmware running. We are using Keil with J-LINK to develop AT32 software and download firmware for AT32F413 device.

* Step 1:

Download [Keil package](https://www.arterytek.com/download/PACK/Keil5_AT32MCU_AddOn_V2.2.5.zip) and install it, to support AT32 MCU to run in KEIL MDK.

* Step 2:

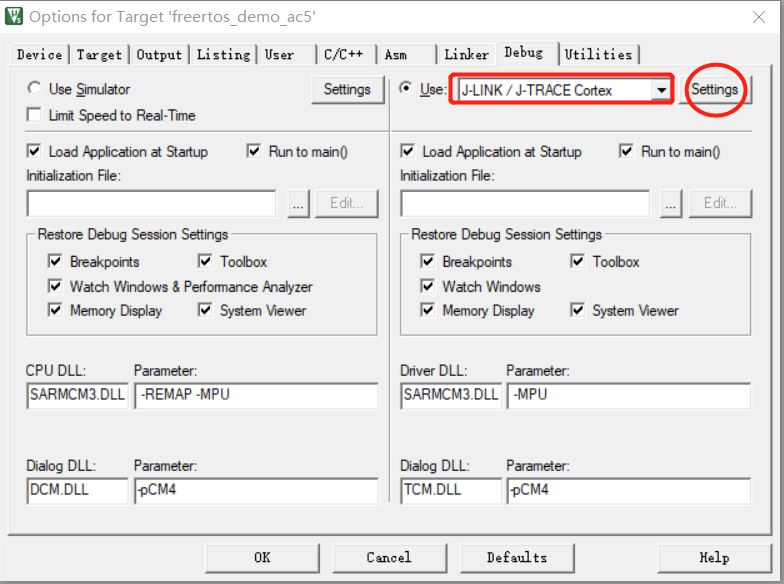
Clone AT32F413 software development environment and controller project from [git-hub](https://github.com/InPlay-Inc/Stellar-Gateway.git) repository. Open “virtual\_comport.uvoptx” under “[Controller](https://github.com/InPlay-Inc/Stellar-Gateway/tree/main/Controller)/[AT32F413\_Firmware\_Library\_V2.0.9](https://github.com/InPlay-Inc/Stellar-Gateway/tree/main/Controller/AT32F413_Firmware_Library_V2.0.9)/[project](https://github.com/InPlay-Inc/Stellar-Gateway/tree/main/Controller/AT32F413_Firmware_Library_V2.0.9/project)/[at\_start\_f413](https://github.com/InPlay-Inc/Stellar-Gateway/tree/main/Controller/AT32F413_Firmware_Library_V2.0.9/project/at_start_f413)/[Stellar](https://github.com/InPlay-Inc/Stellar-Gateway/tree/main/Controller/AT32F413_Firmware_Library_V2.0.9/project/at_start_f413/Stellar)/mdk\_v5/” directory with KEIL MDK. Rebuild it.

* Step 3:

Make a JLINK connection with AT32 through SWD pins on carry board as show in Figure 2. In KEIL MDK, press “ALT+F7”, option for target dialog is shown as Figure 3. In “Debug” page, choose “J-LINK / J-TRACE Cortex” and then click “Setting”, choose “SW” port and there will be available SW devices shown.



Figure 2: J-LINK connection with AT32



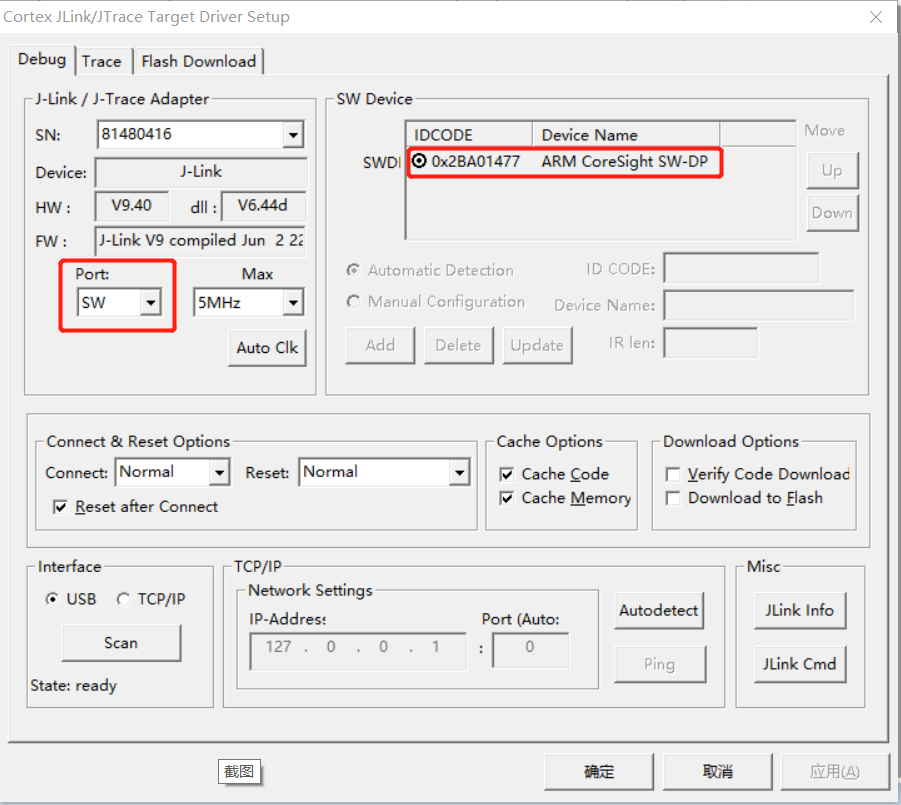


Figure 3: Keil – Setup J-LINK connection

Change to “Flash Download” page, choose correct programming algorithm for AT32F413 and RAM address.

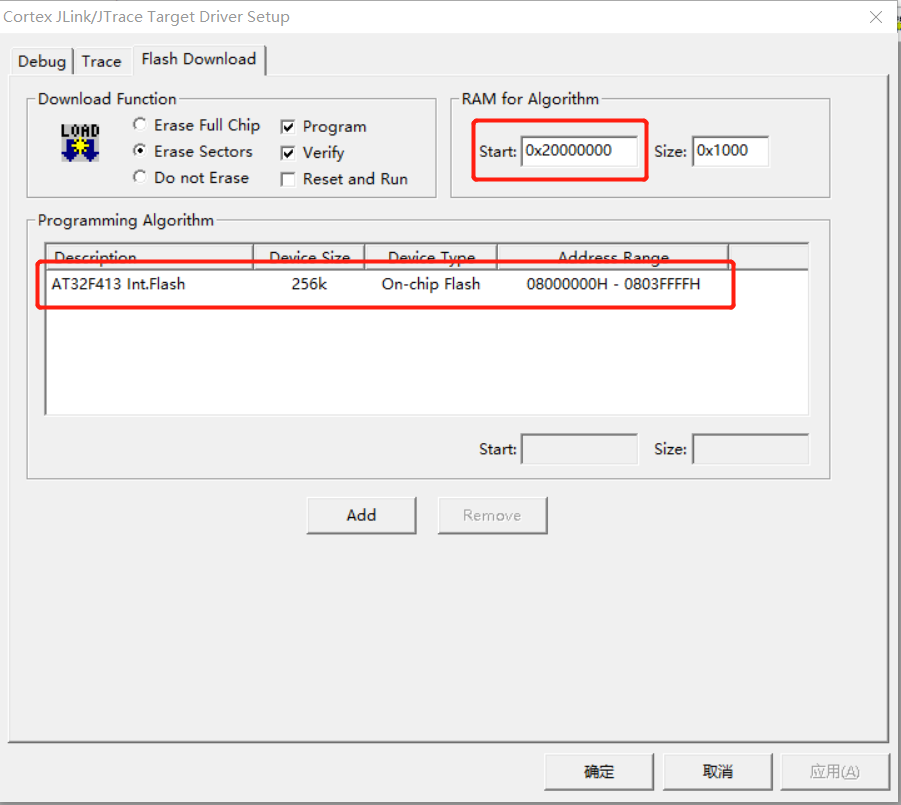


Figure 4: Keil – Programming Algorithm

Now go back to main screen of project, choose “Flash->Download” menu item to start programming.

* Step 4:

Press SW1 key to reboot Stellar Gateway. On default, you can see following log info from USART5 of AT32. USART5 pins are as below.

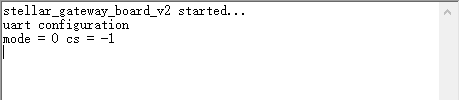


Figure 5: USART log of AT32

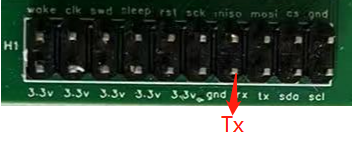


Figure 6: USART5 pin location

# IN612L Firmware Download Guild

We have our IN612Ls programmed when the board is shipped to you. But since it is also under development, it may be necessary to update firmware of IN612L. It relies on AT32 firmware as well as proper IO configuration. Here is the detail steps to program IN612L:

* Step 1:

Plug in USB type-C cable to carry board and connect it to PC. Power on the board by pressing down SW1 button. You will now see a new COM port appears on PC window operation system.

* Step 2:

Select which IN612L to download by connecting “sda” and “scl” pin to different level, as show in Figure 7.

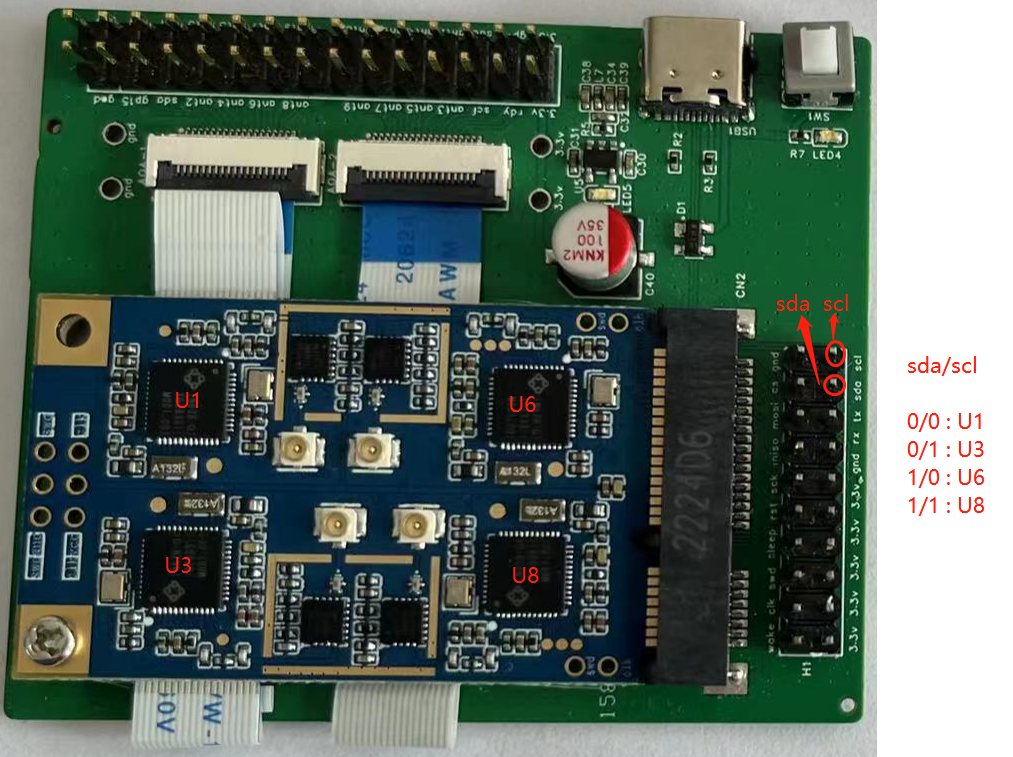


Figure 7: In612L Chip selection IO configuration

* Step 3:

Run *[Inplay Programmer](#_Inplay Programmer)* and select that new COM port and click “Connect” button.

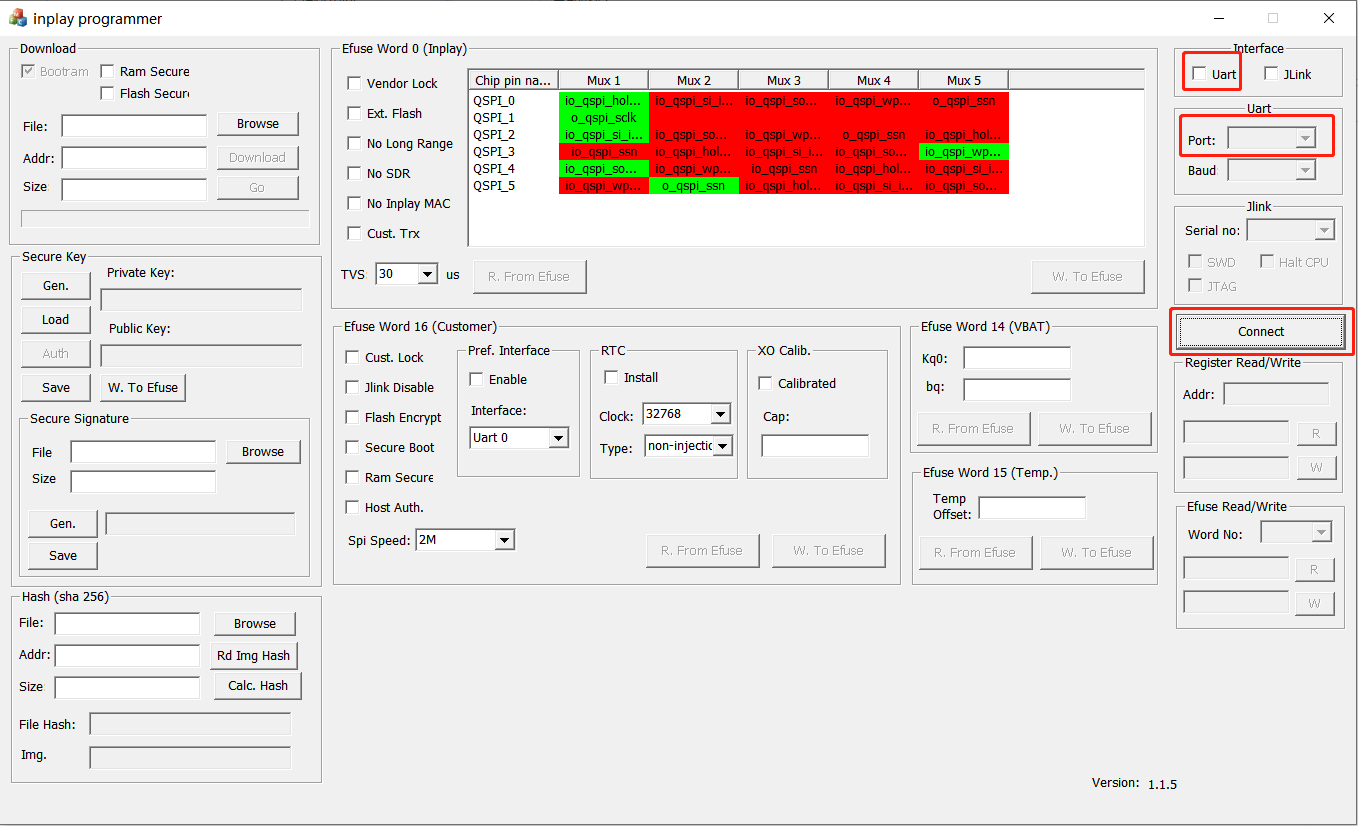


Figure 8: Inplay Programmer – Connect UI

After a while, you will see popup dialog in Figure 9 which means connection is established.

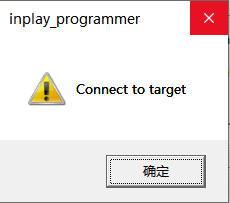


Figure 9: Inplay Programmer – Popup Dialog after connect

* Step 3:

Choose the firmware binary file and click “Download” to start programming. After finished, press “Disconnect” botton to stop connection.

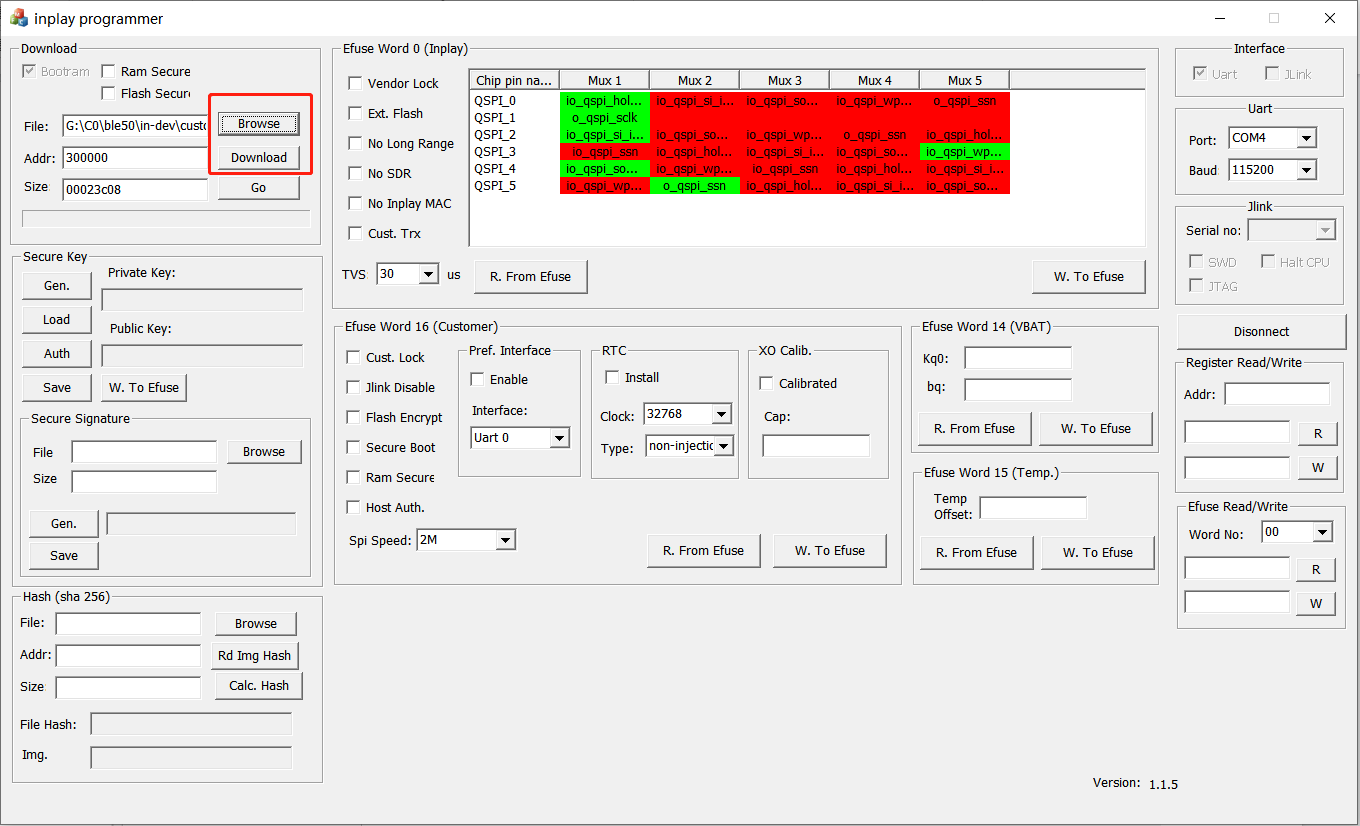


Figure 5: Inplay Programmer – Download UI

* Step 4:

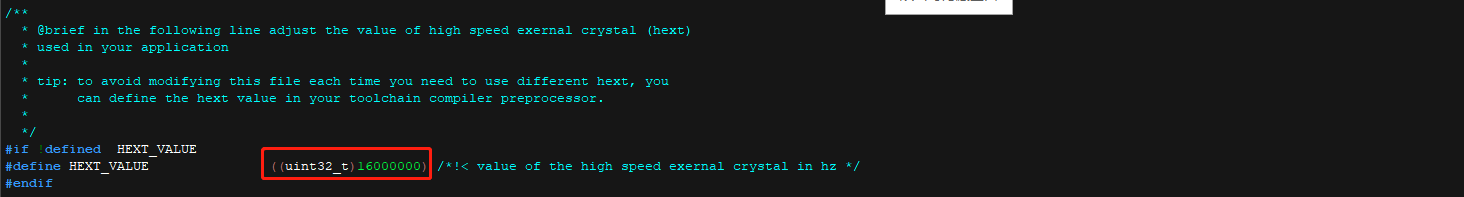
Repeat from step2 to select another IN612L to program until all IN612Ls are programmed. From then on, IN612Ls are functional available and accept command.

# AT32F413 Sample Project

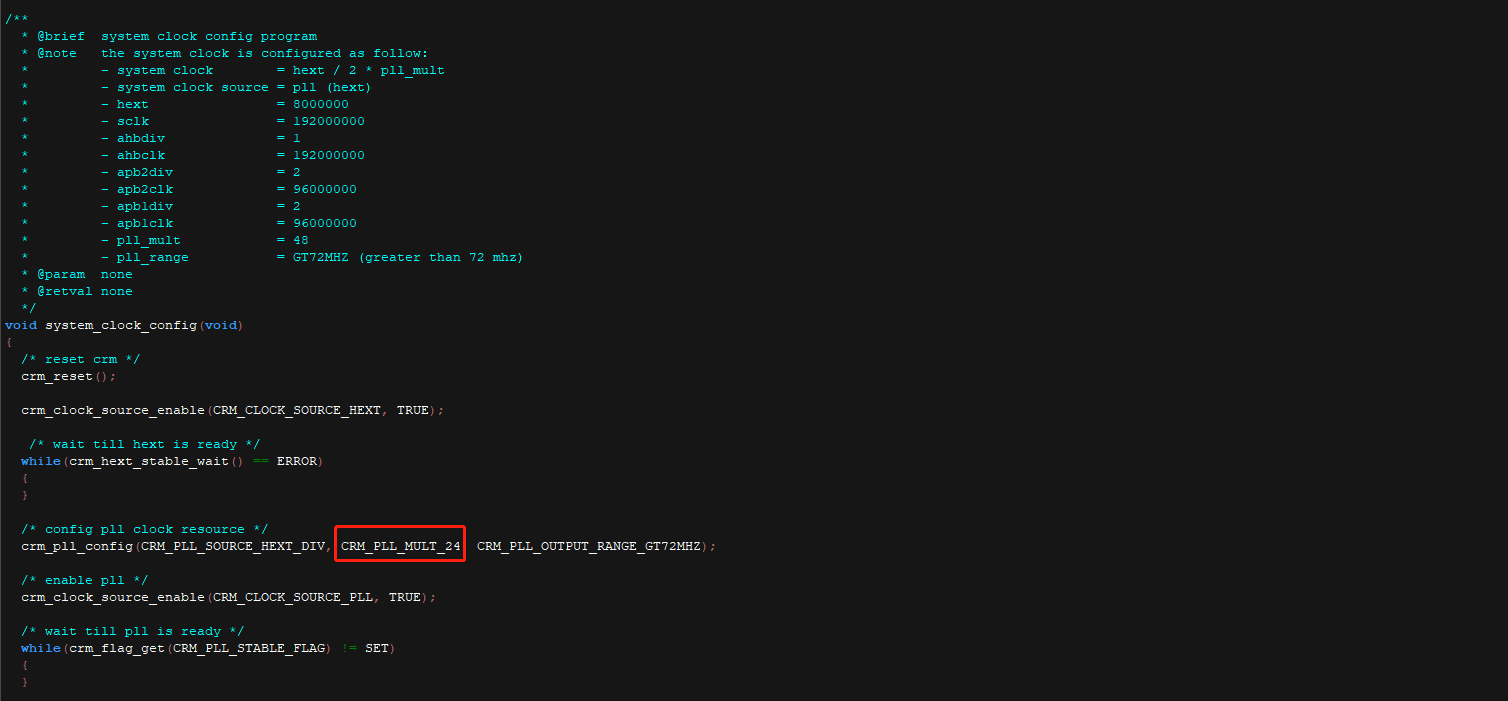
Details about AT32F413 software development, firmware download, please refer to related official SDK. There are two key points that are MUST for Stellar Gateway development:

1. On Stellar Gateway board, the external crystal of AT32F413 is 16MHz instead of 8MHz of AT32 reference board. So the definition of high speed external crystal value and related PLL configuration should be changed.

*[at32f413\_conf.h](http://192.168.0.115/doku.php?do=export_code&id=zz&codeblock=0" \o "下载片段)*



*at32f413\_clock.c*



1. As Controller, AT32F413 MCU use RST\_x signal (PB1/PC3/PA8/PB6) to control the chip\_en of all IN612Ls and another BOOST\_x signal (PB0/PC2/PC9/PB5) to control whether or not the IN612L enters boot mode. So the proper thing that any AT32F413 software should do first is to output high on each BOOST\_x pin and low level pulse on RST\_x pin to enable IN612Ls power on reset.

The above two changes are included among attached sample project.

1. Please refer to <https://inplay-inc.github.io/docs/solutions/stellar/stellar-gateway-wireless-communication-protocol-implementation-guide.html> for detail data protocols.

# Attachement

## Inplay Programmer

