

# Regbin Tool User Guideline

# Revision history

Ver	Date	Author	Description
1.0	2020/07/18		Initial
1.1	2021/3/8		Review revision
1.2	2023/7/25		Support 1.3.6
1.3	2023/10/9		Add how to convert Jason file to bin file
1.4	2024/1/1		Add regbin_parser.exe & parsing-tool.bat
1.5	2024/12/17		Add section “import cfg file”
1.6	2025/1/20		Add init feature for tasdevice driver

# Content

- Regbin file Introduction
- Bin file making
- How to use
- Applied cases
- To be Improved

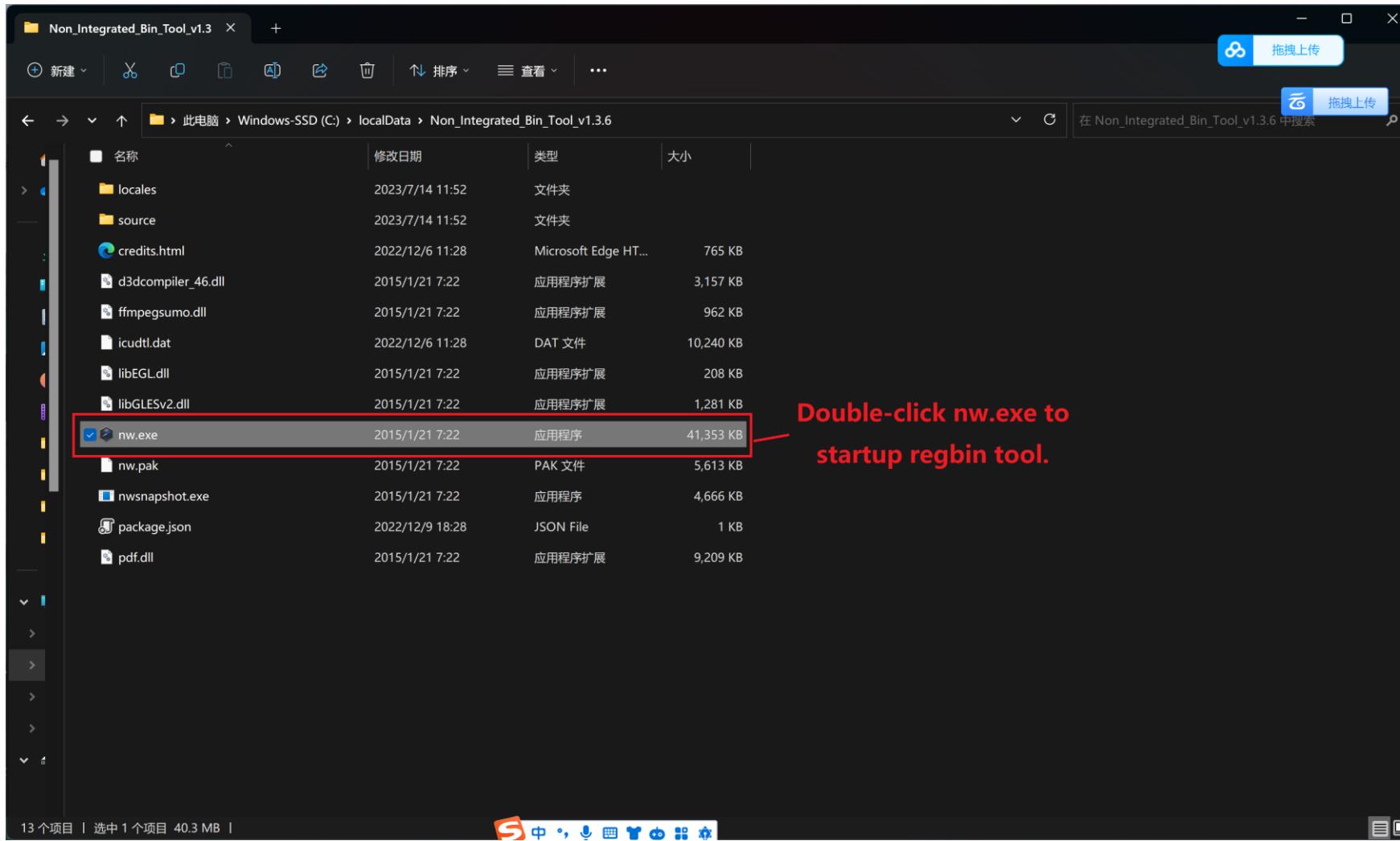
# Regbin file

- Register setting can be provided via a single file
  - No Software change for register setting change
  - Holds the entire set of application register setting for some special cases
  - Control multiple devices freely
  - applied in the chip pre-poweron, post-poweron, pre-poweroff and post-poweroff
  - For read-only use in an embedded system
    - The bin file is read only once during system bootup
  - Information in header for customer application description, such as platform vendor, chip type, chip quantity, etc.

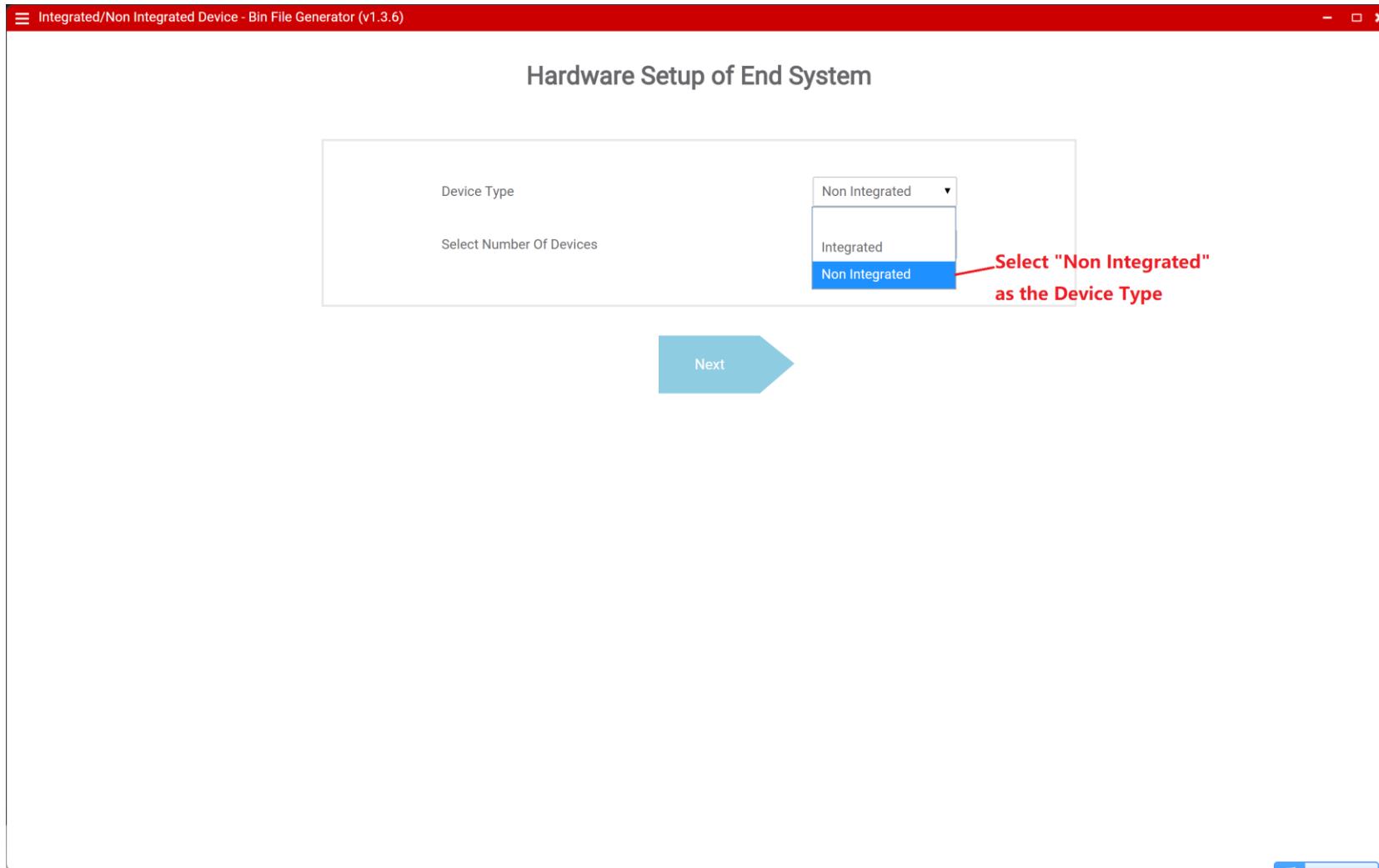
# Motivation

- There is a single file to share
  - between phone and development + debug environment
  - between peers and customers
- Helps to improve reproducibly
  - Version can be embedded in the file
  - Benefit for issue debug
- A step to reduce dependency on code build
  - Limit SW build dependency in the loop for AE debugging
- eg: In 4-slot TDM bring-up, it can save much time in building if reg-bin solution is ready at that time.

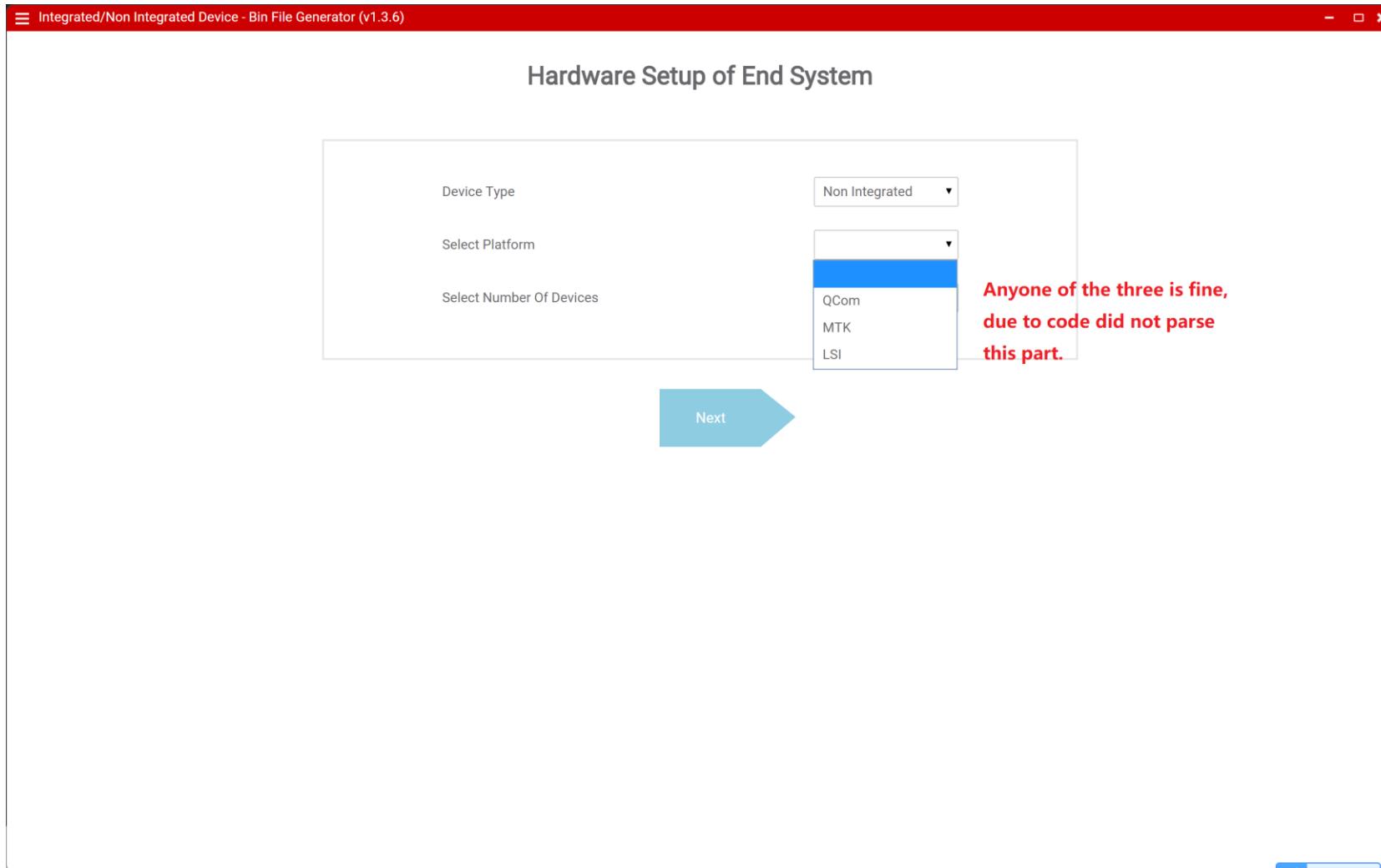
# Bin File Making I | Startup Regbin tool



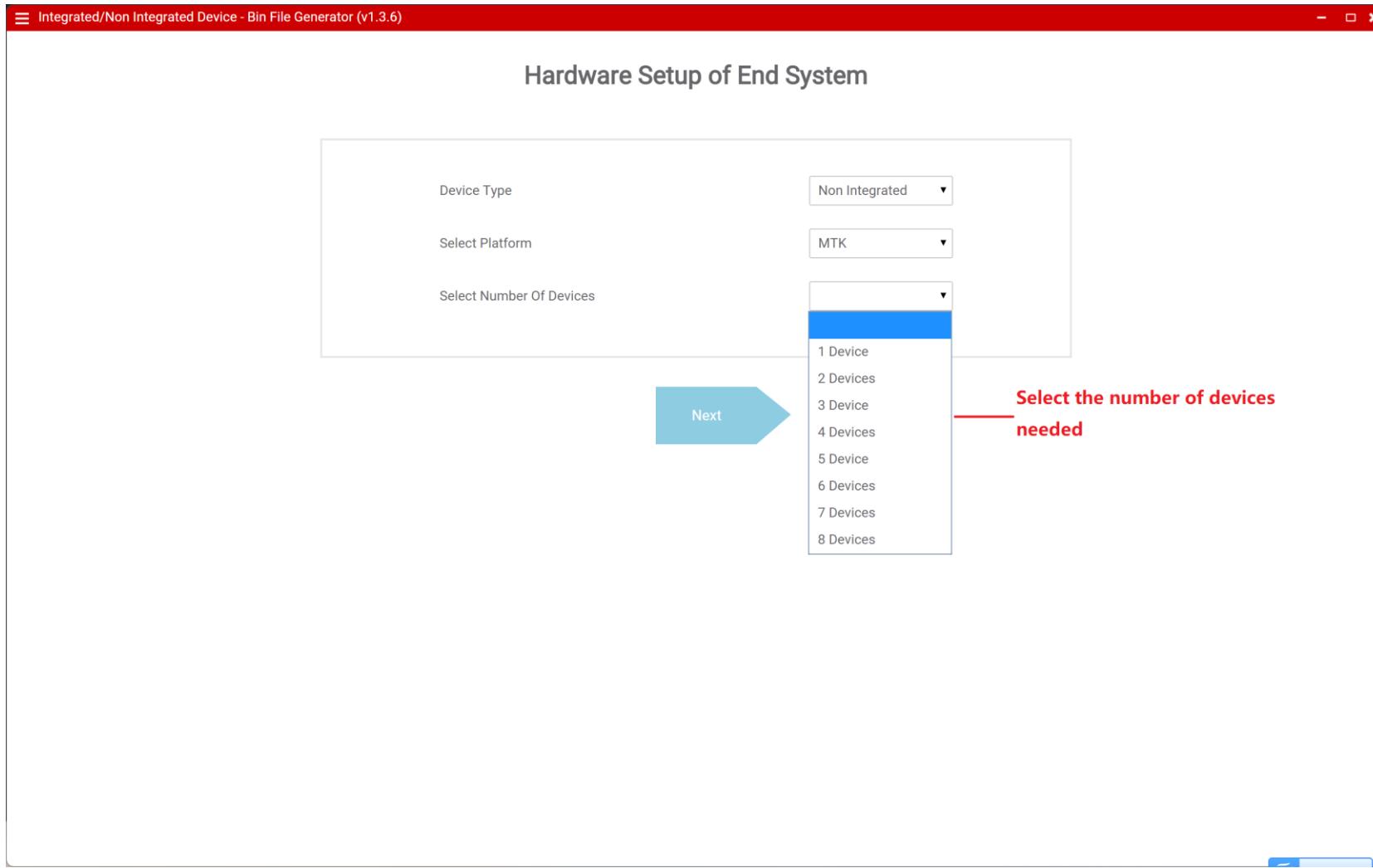
# Bin File Making II | Select Device Type



# Bin File Making III | Select Platform



# Bin File Making IV | Select Number of Devices



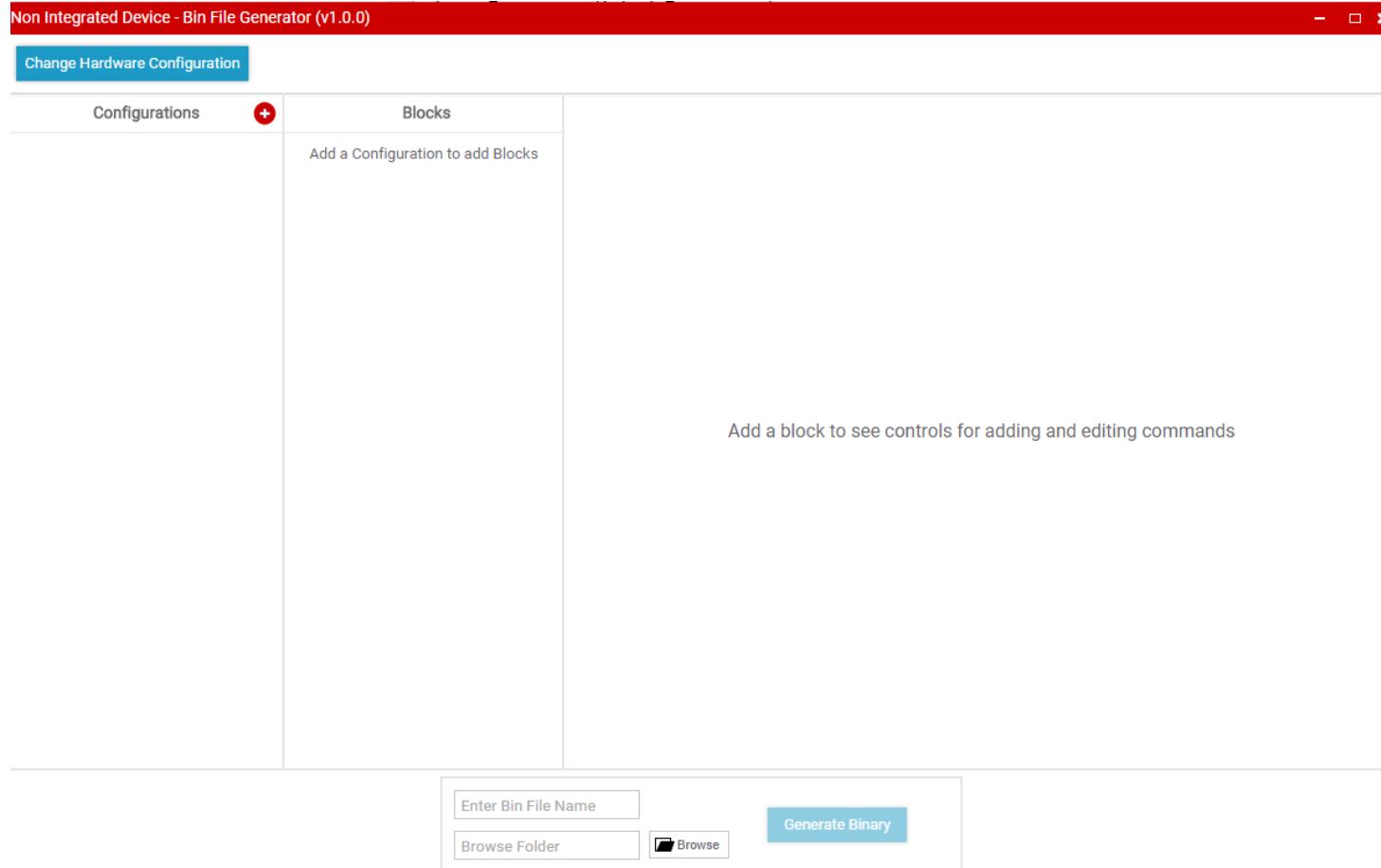
# Bin File Making V | Select Device type

The image shows two screenshots of the "Integrated/Non Integrated Device - Bin File Generator (v1.3.6)" software. Both screenshots are titled "Hardware Setup of End System".

**Left Screenshot:** This step is for "Non Integrated" devices. It includes fields for "Device Type" (Non Integrated), "Select Platform" (MTK), "Select Number Of Devices" (1 Device), and "Device 1" (a dropdown menu with options TAS2558, TAS2560, TAS2562, and TAS2564). A red callout box highlights the "Device 1" dropdown with the instruction: "Select any of them, code do not parse this part". A blue "Next" button is at the bottom right.

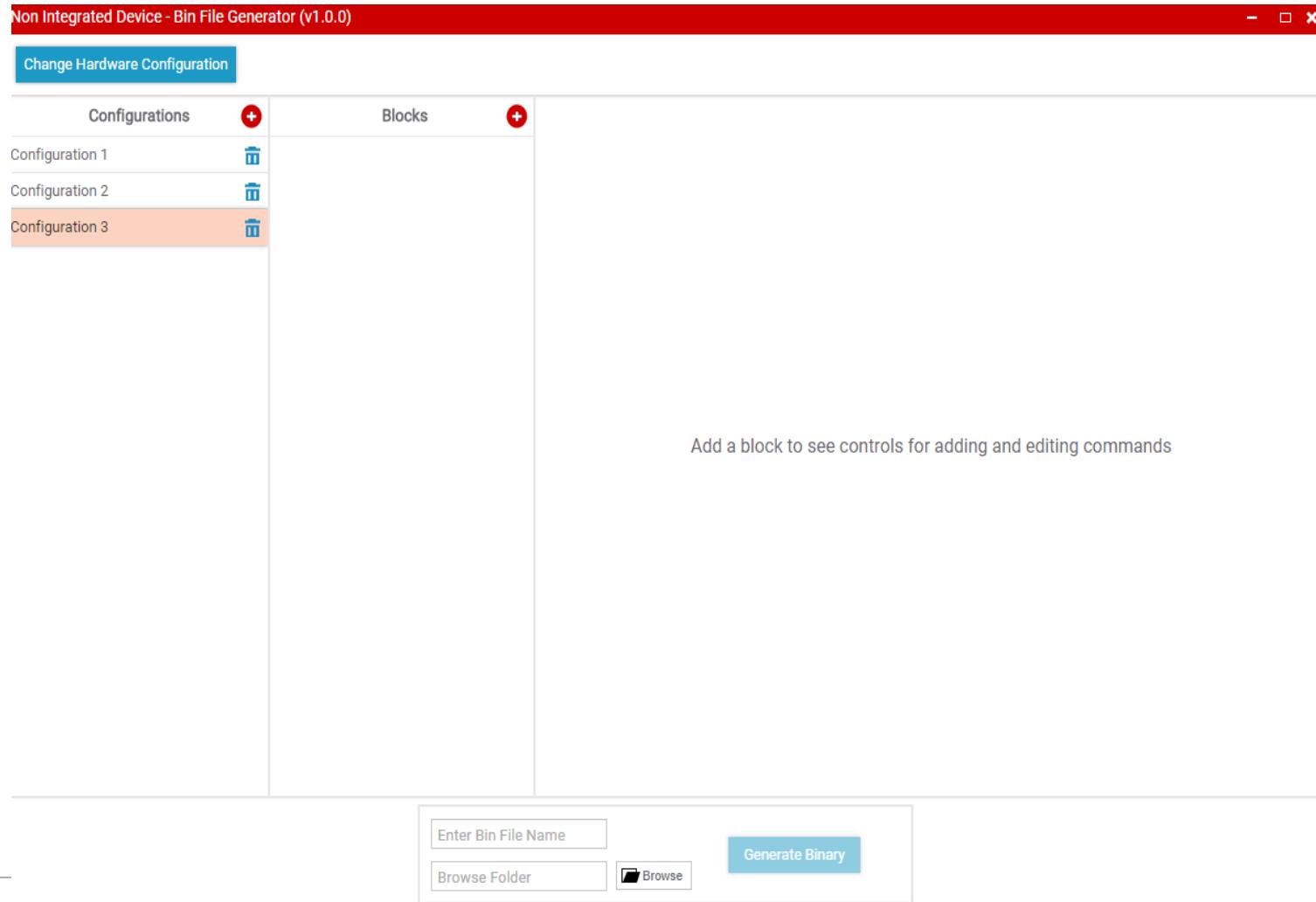
**Right Screenshot:** This step is for "Non Integrated" devices. It includes fields for "Device Type" (Non Integrated), "Select Platform" (QCom), "Select Number Of Devices" (8 Devices), and "Device 1 through Device 8" (each a dropdown menu with options TAS2562, TAS2562, TAS2558, TAS2560, TAS2564, TAS2562, TAS2562, and TAS2564). The dropdown menus for Device 1 through Device 8 are highlighted with a red border. A blue "Next" button is at the bottom right.

# Bin File Making VI | Configuration Page



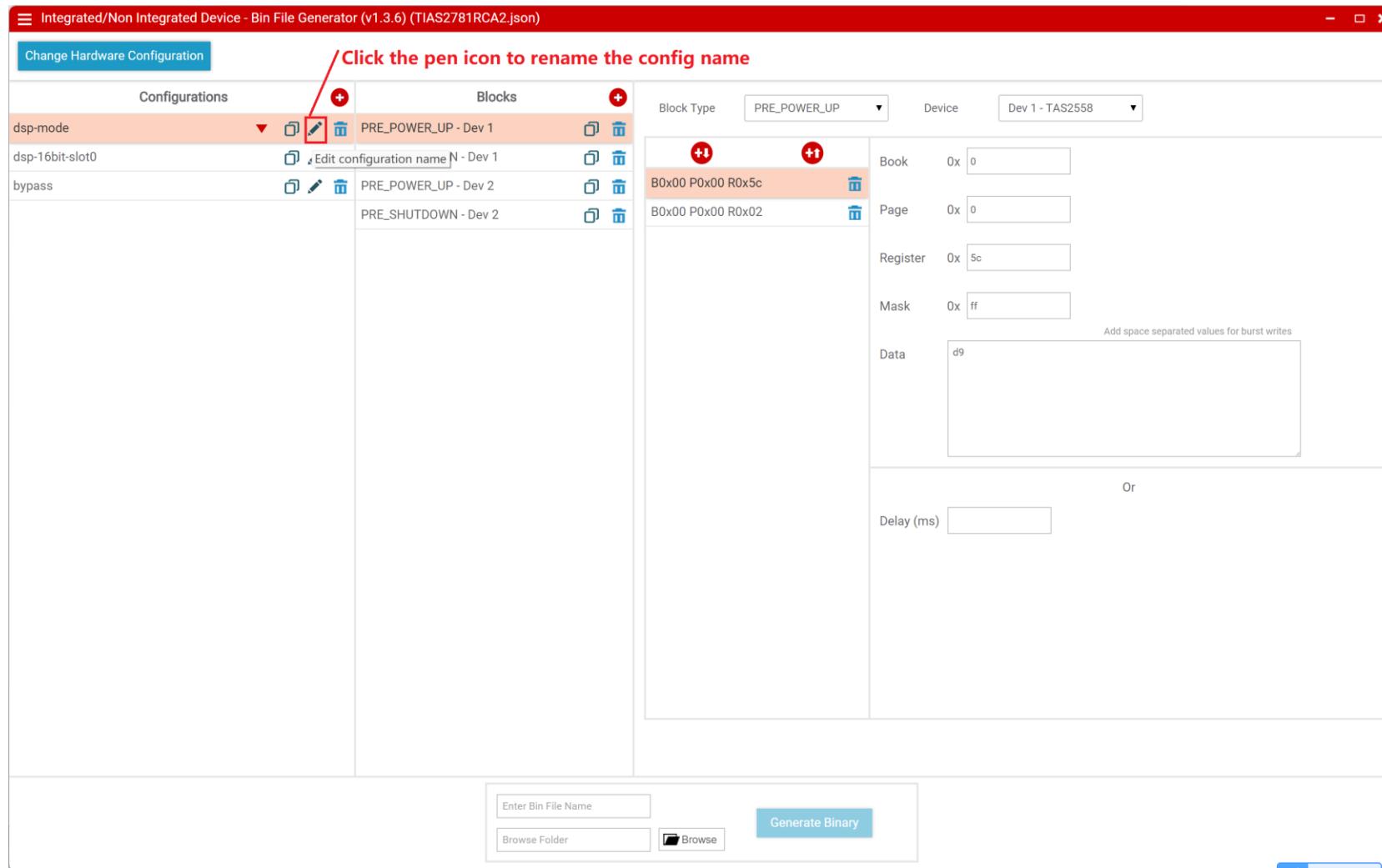
Config Page is displayed successfully on clicking Next in hardware setup page

# Bin File Making VII | Add Configuration



Configurations can be added successfully on clicking the “+” icon in the configurations box

# Bin File Making VIII | Rename Configuration name



# Bin File Making IX | Copy Configuration

Integrated/Non Integrated Device - Bin File Generator (v1.3.6) (TIAS2781RCA2.json)

Change Hardware Configuration

Configurations

	Blocks
dsp-mode	PRE_POWER_UP - Dev 1
dsp-16bit-slot0	PRE_SHUTDOWN - Dev 1
bypass	PRE_POWER_UP - Dev 2
	PRE_SHUTDOWN - Dev 2

Blocks

Block Type	Device
PRE_POWER_UP	Dev 1 - TAS2558

Click the COPY icon to copy the whole config setting.

B0x00 P0x00 R0x5c

Book 0x 0

Page 0x 0

Register 0x 5c

Mask 0x ff

Add space separated values for burst writes

Data d9

Or

Delay (ms)

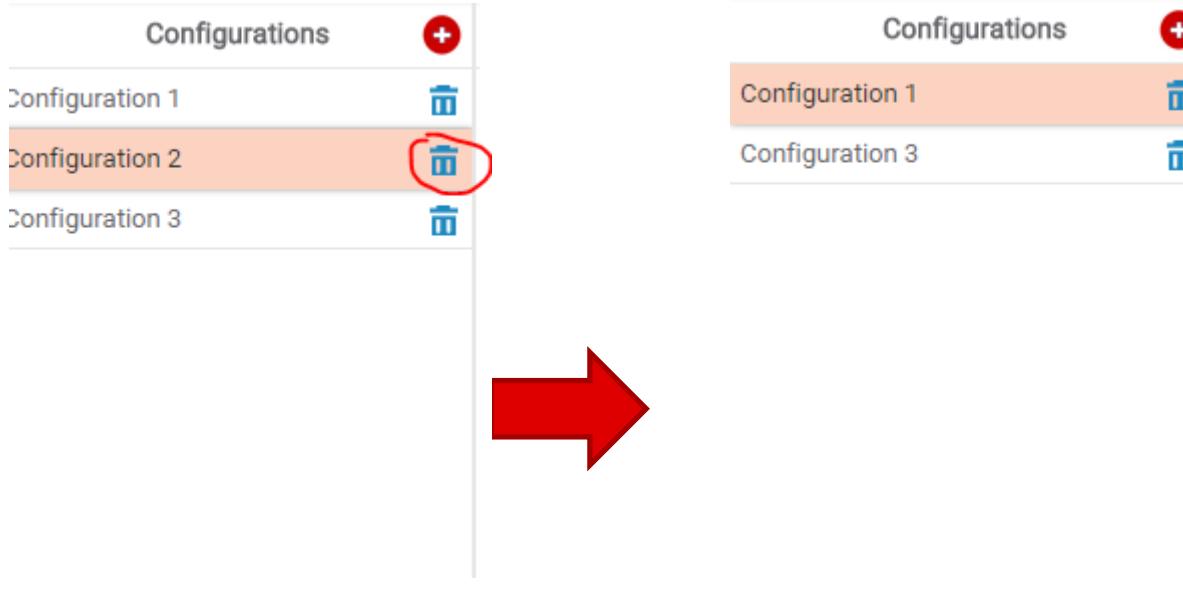
Enter Bin File Name

Browse Folder

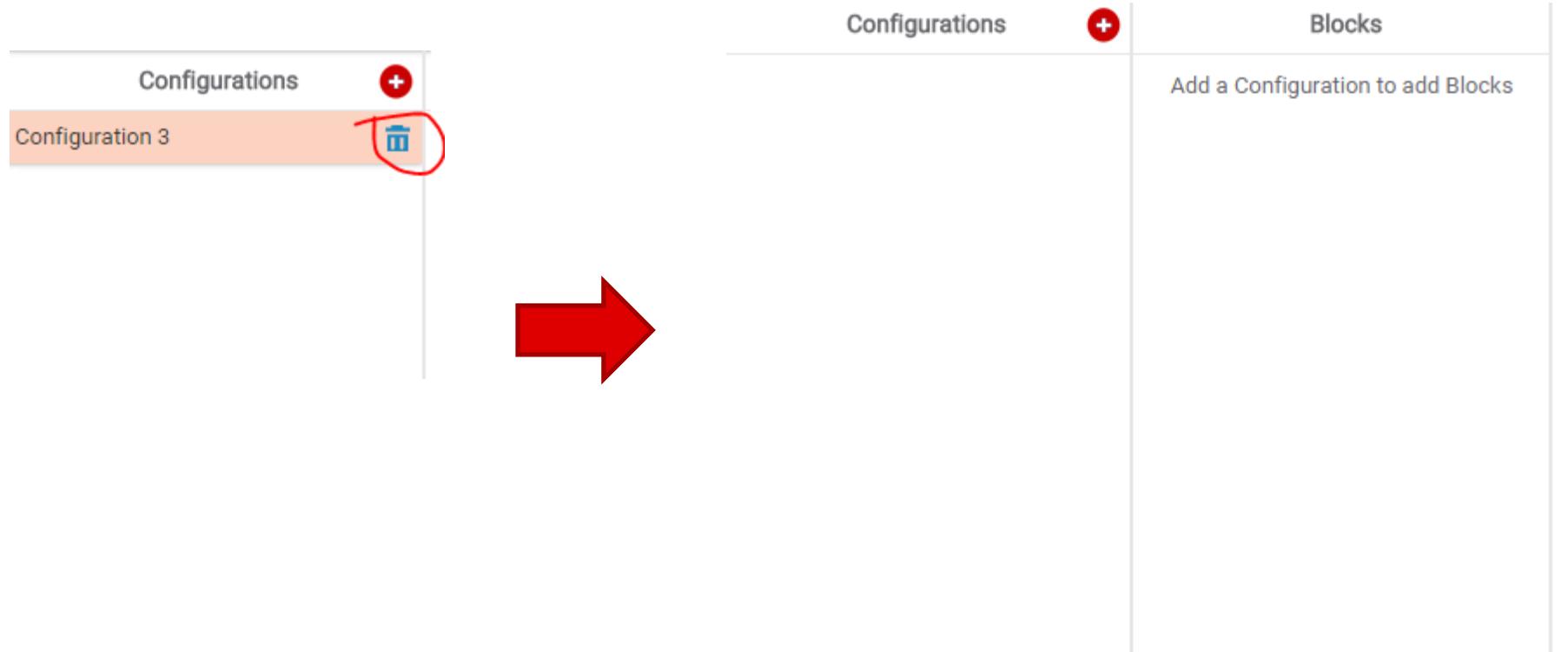
Generate Binary

14

# Bin File Making X | Delete Configuration



# Bin File Making XI | Delete Last Configuration



# Bin File Making XII | Add Blocks

Non Integrated Device - Bin File Generator (v1.0.0) - □ ×

Change Hardware Configuration

Configurations	Blocks
Configuration 1	Block 1
Configuration 2	Block 2
Configuration 3	

Block Type:  Device:

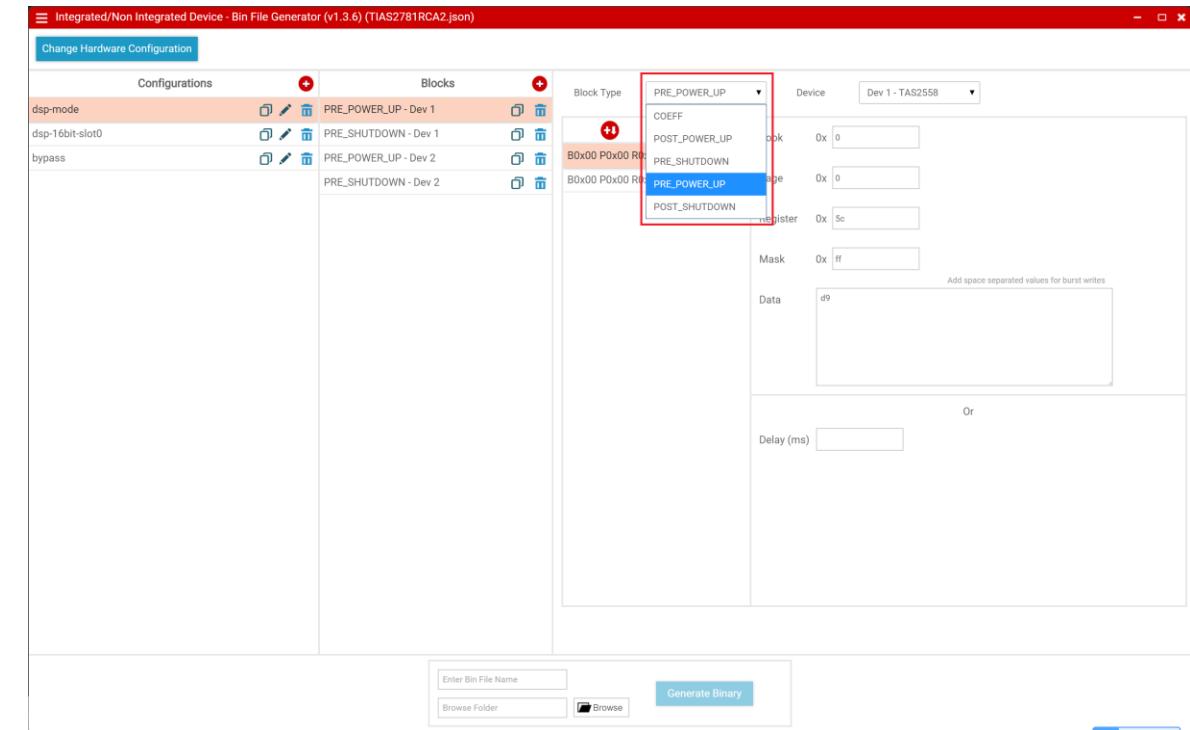
+ Add First Command

Enter Bin File Name   
Browse Folder  Generate Binary

Blocks can be added successfully on clicking the “+” icon in the Blocks box

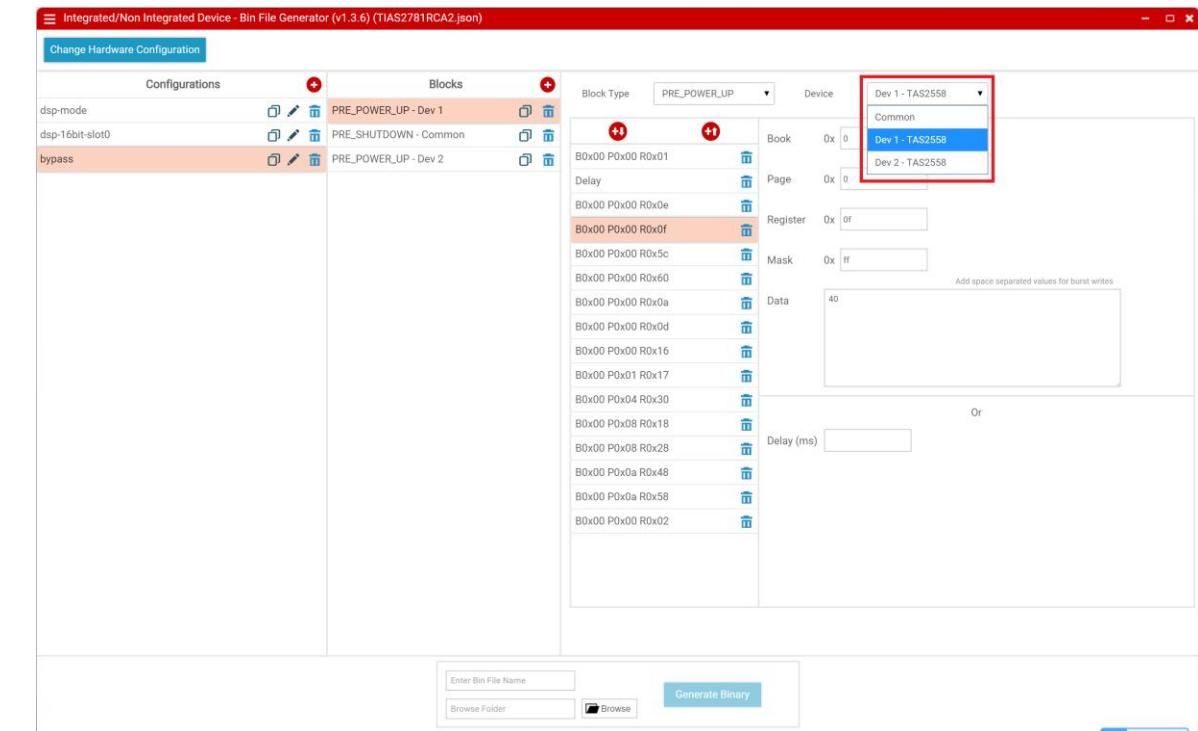
# Bin File Making XIII | Select Block Type

- The driver only support PRE\_POWER\_UP for chip power up, and PRE\_SHUTDOWN for chip shutdown.

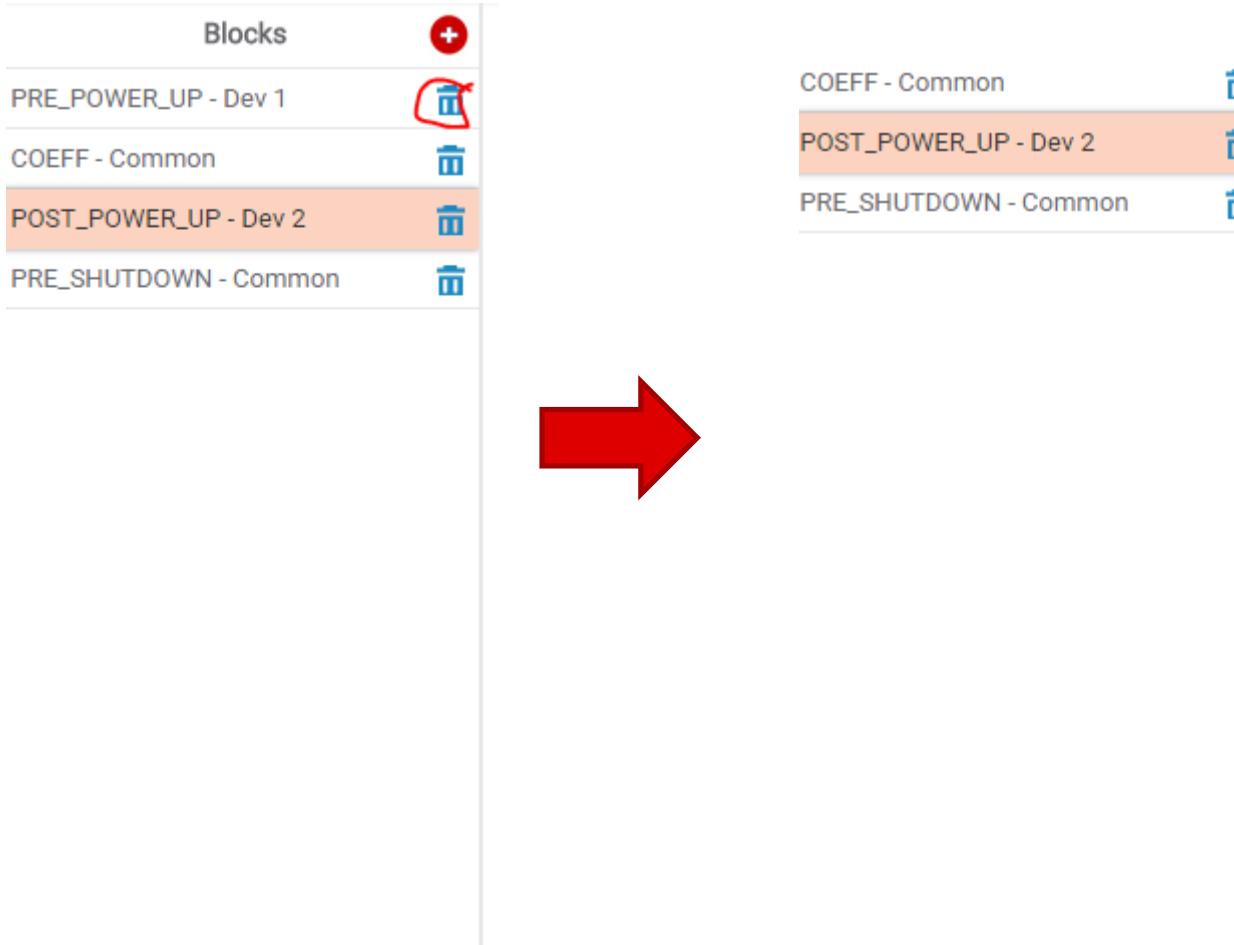


# Bin File Making XIV | Select Device for one specific Block

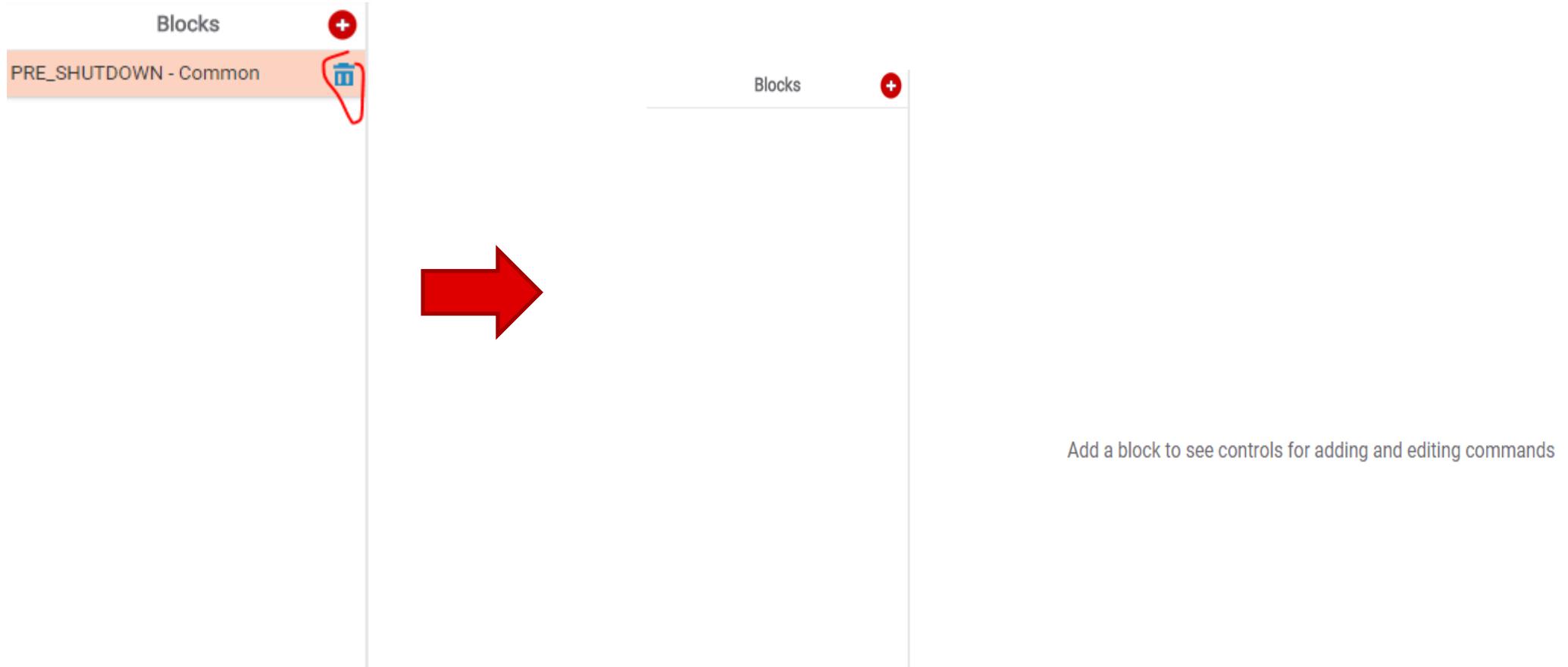
- “Common” means all the devices defined in the regbin file, for the right picture, the common means both dev 1 and dev 2.



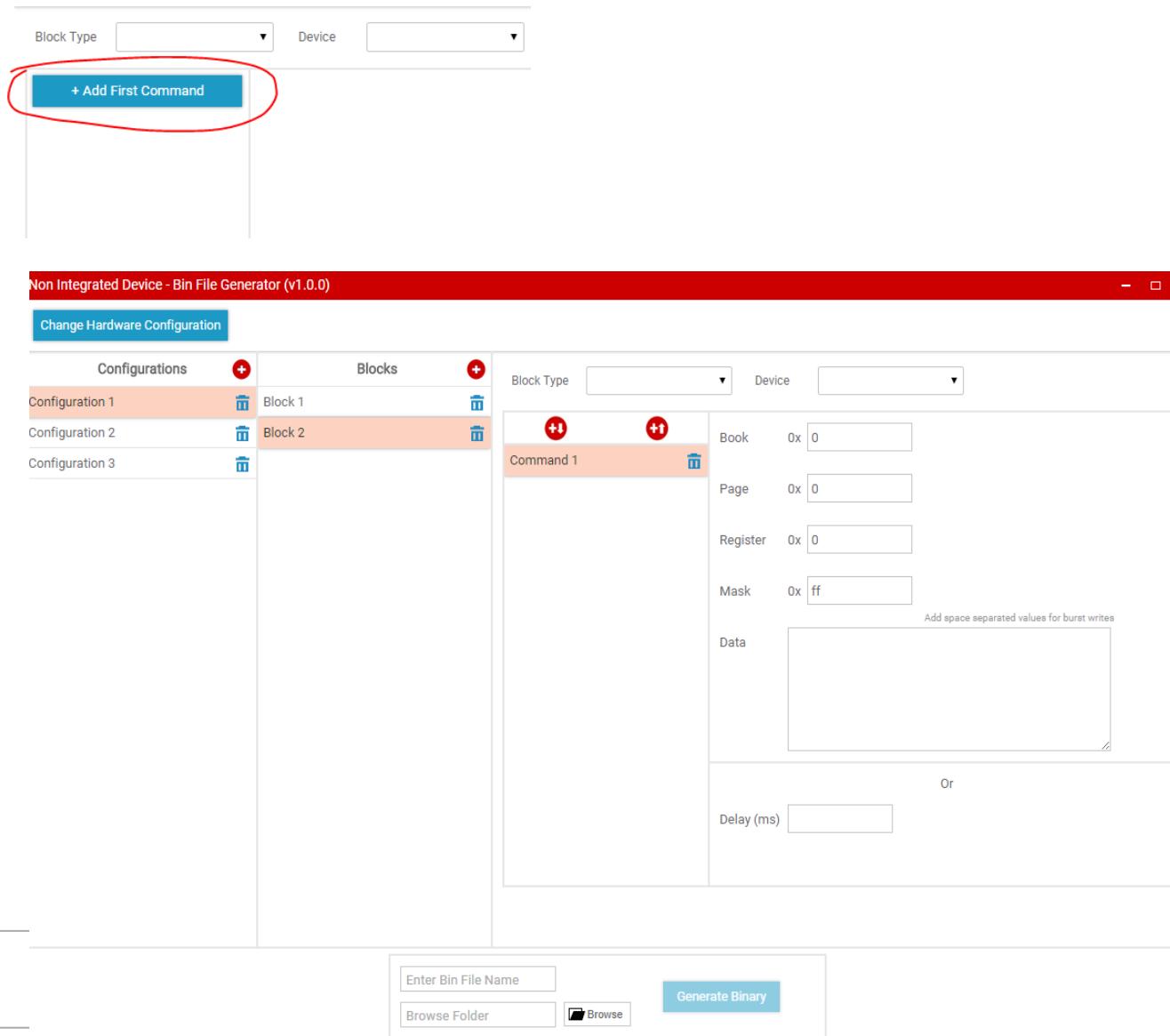
# Bin File Making XV | Delete Block



# Bin File Making XVI | Delete Last Block in Configuration



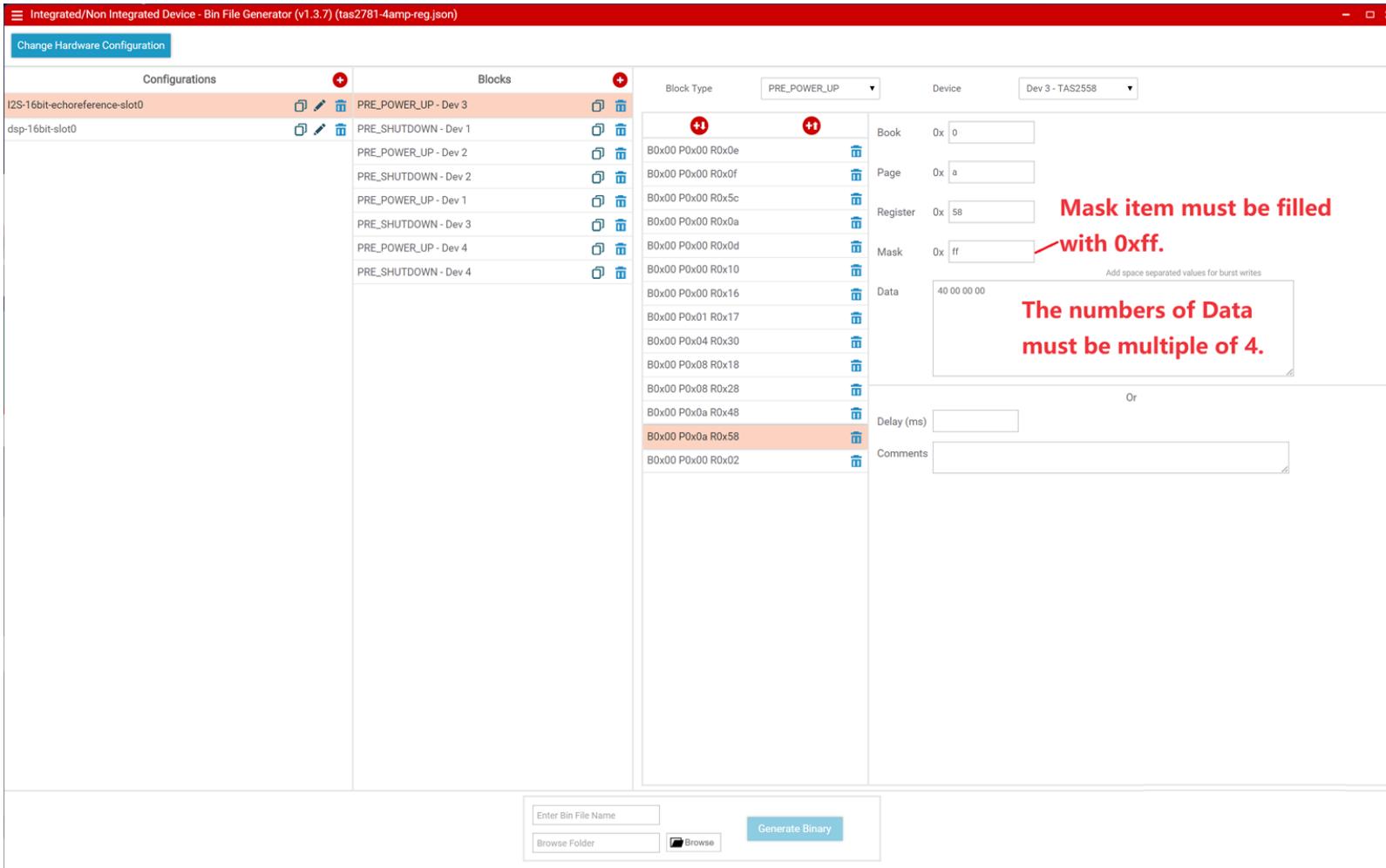
# Bin File Making XVI | Add First Command



First command can be added successfully on clicking the “Add First Command” button

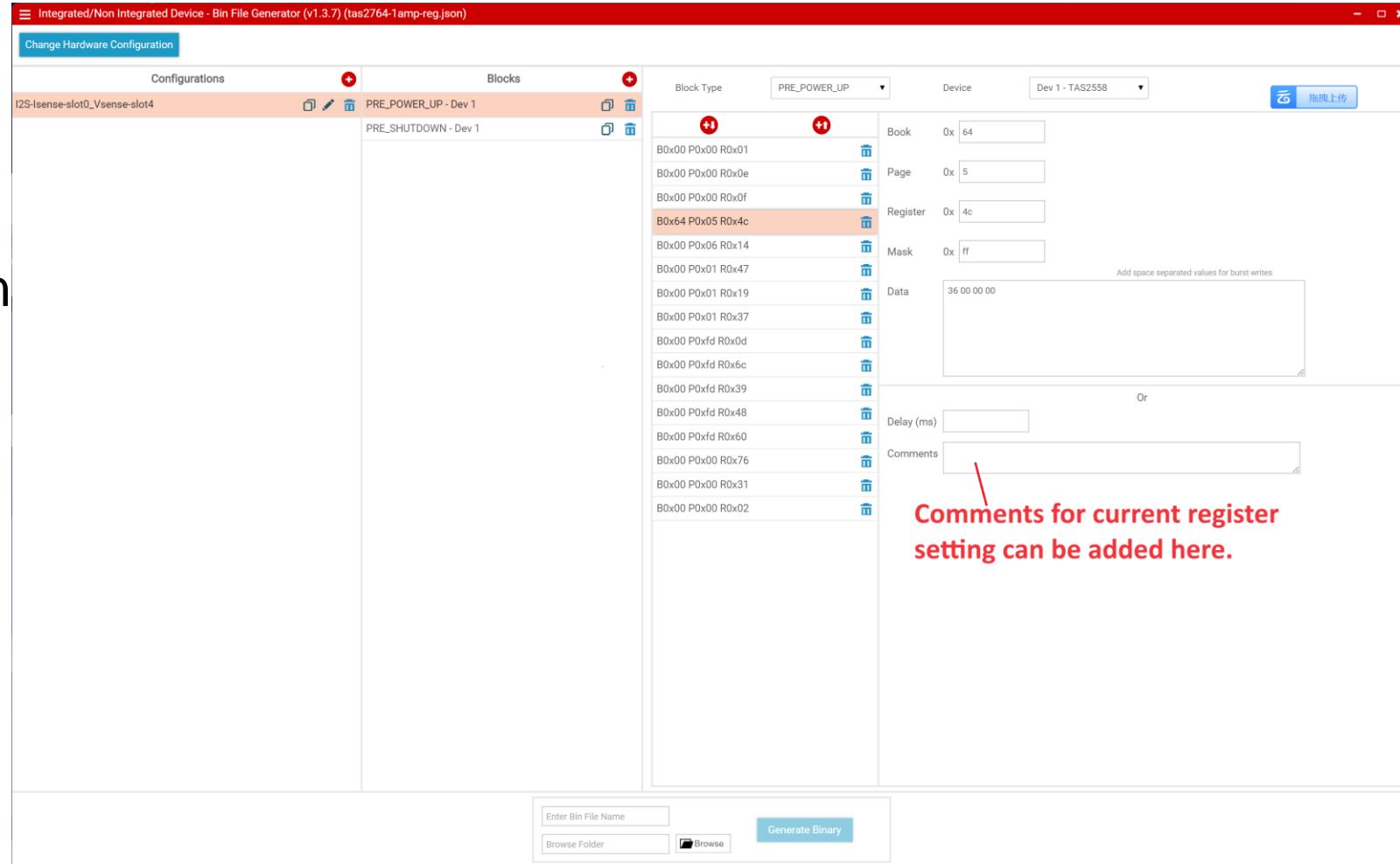
# Bin File Making XVII | Bulk write/Burst commands

- If writing BOOK0x00, PAGE0x02, REG0x64 ~ REG0x67, pls use following way.



# Bin File Making XVIII | Add comment for register setting

- Since Regbin tool r1.3.7, comment text box for each register setting item has been added, developer can add the comments freely.



# Bin File Making XIX | Add Below Selected Command

Add a piece of register setting after one certain piece of register setting

Block Type: PRE\_POWER\_UP

Device: Dev 1 - TAS2558

Book	0x 0
Page	0x 08
Delay	
Register	0x 28
Mask	0x ff
Data	40 00 00 00 Add space separated values for burst writes
Or	
Delay (ms)	

Block Type: PRE\_POWER\_UP

Device: Dev 1 - TAS2558

Book	0x 0
Page	0x 0
Register	0x 0
Mask	0x ff
Data	Add space separated values for burst writes
Or	
Delay (ms)	

# Bin File Making XX | Add above Selected Command

Add a piece of register setting before a certain register setting

Block Type	Device
PRE_POWER_UP	Dev 1 - TAS2558
B0x00 P0x00 R0x01	
Delay	
B0x00 P0x00 R0x0e	
B0x00 P0x00 R0x0f	
B0x00 P0x00 R0x5c	
B0x00 P0x00 R0x60	
B0x00 P0x00 R0x0a	
B0x00 P0x00 R0x0d	
B0x00 P0x00 R0x16	
B0x00 P0x01 R0x17	
B0x00 P0x04 R0x30	
B0x00 P0x08 R0x18	
B0x00 P0x08 R0x28	
B0x00 P0x0a R0x48	
B0x00 P0x0a R0x58	
B0x00 P0x00 R0x02	

Book 0x 0

Page 0x 0

Register 0x 1

Mask 0x ff

Data 1  Add space separated values for burst writes

Or

Delay (ms)

Block Type PRE\_POWER\_UP Device Dev 1 - TAS2558

Block Type	Device
PRE_POWER_UP	Dev 1 - TAS2558
B0x00 P0x00 R0x00	
B0x00 P0x00 R0x01	
Delay	
B0x00 P0x00 R0x0e	
B0x00 P0x00 R0x0f	
B0x00 P0x00 R0x5c	
B0x00 P0x00 R0x60	
B0x00 P0x00 R0x0a	
B0x00 P0x00 R0x0d	
B0x00 P0x00 R0x16	
B0x00 P0x01 R0x17	
B0x00 P0x04 R0x30	
B0x00 P0x08 R0x18	
B0x00 P0x08 R0x28	
B0x00 P0x0a R0x48	
B0x00 P0x0a R0x58	
B0x00 P0x00 R0x02	

Book 0x 0

Page 0x 0

Register 0x 0

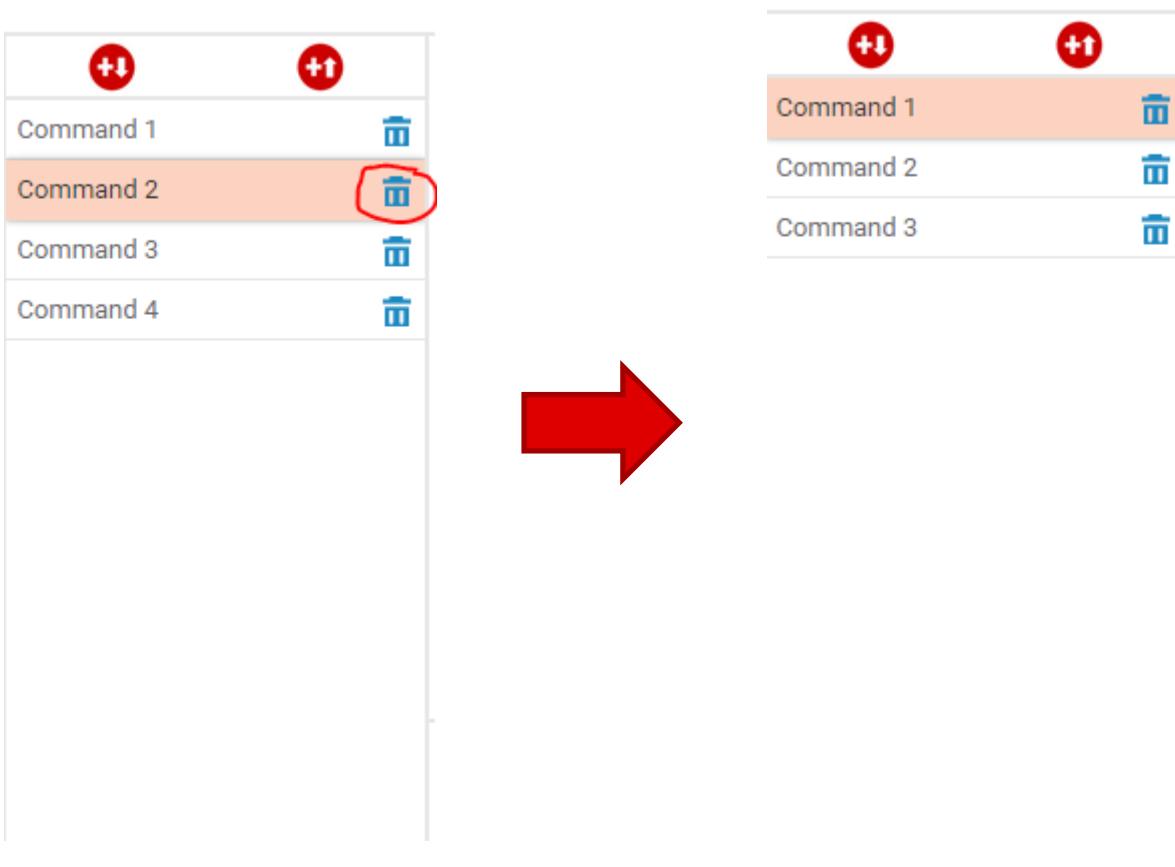
Mask 0x ff  Add space separated values for burst writes

Data

Or

Delay (ms)

# Bin File Making XXI | Delete Command



Command 2 is deleted and  
command 3 is moved up as  
command 2

# Bin File Making XXII || Save/Save as the Regbin project file | json file

Integrated/Non Integrated Device - Bin File Generator (v1.3.6) (TIAS2781RCA2.json)

New Configuration

Open

Save

Save as

Configurations

Blocks

PRE\_POWER\_UP - Dev 1

PRE\_SHUTDOWN - Dev 1

PRE\_POWER\_UP - Dev 2

PRE\_SHUTDOWN - Dev 2

Block Type: PRE\_POWER\_UP

Device: Dev 1 - TAS2558

B0x00 P0x00 R0x5c

Book: 0x 0

Page: 0x 0

Register: 0x 5c

Mask: 0x ff

Data: d9

Add space separated values for burst writes

Delay (ms):

Or

Enter Bin File Name:

Browse Folder:

Generate Binary

# Bin File Making XXIII | Export the regbin binary file I

Integrated/Non Integrated Device - Bin File Generator (v1.3.6) (TIAS2781RCA2.json)

Change Hardware Configuration

Configurations	Blocks
dsp-mode	PRE_POWER_UP - Dev 1
dsp-16bit-slot0	PRE_SHUTDOWN - Dev 1
bypass	PRE_POWER_UP - Dev 2
	PRE_SHUTDOWN - Dev 2

Block Type: PRE\_POWER\_UP | Device: Dev 1 - TAS2558

B0x00 P0x00 R0x5c

Book: 0x 0 | Page: 0x 0 | Register: 0x 5c | Mask: 0x ff

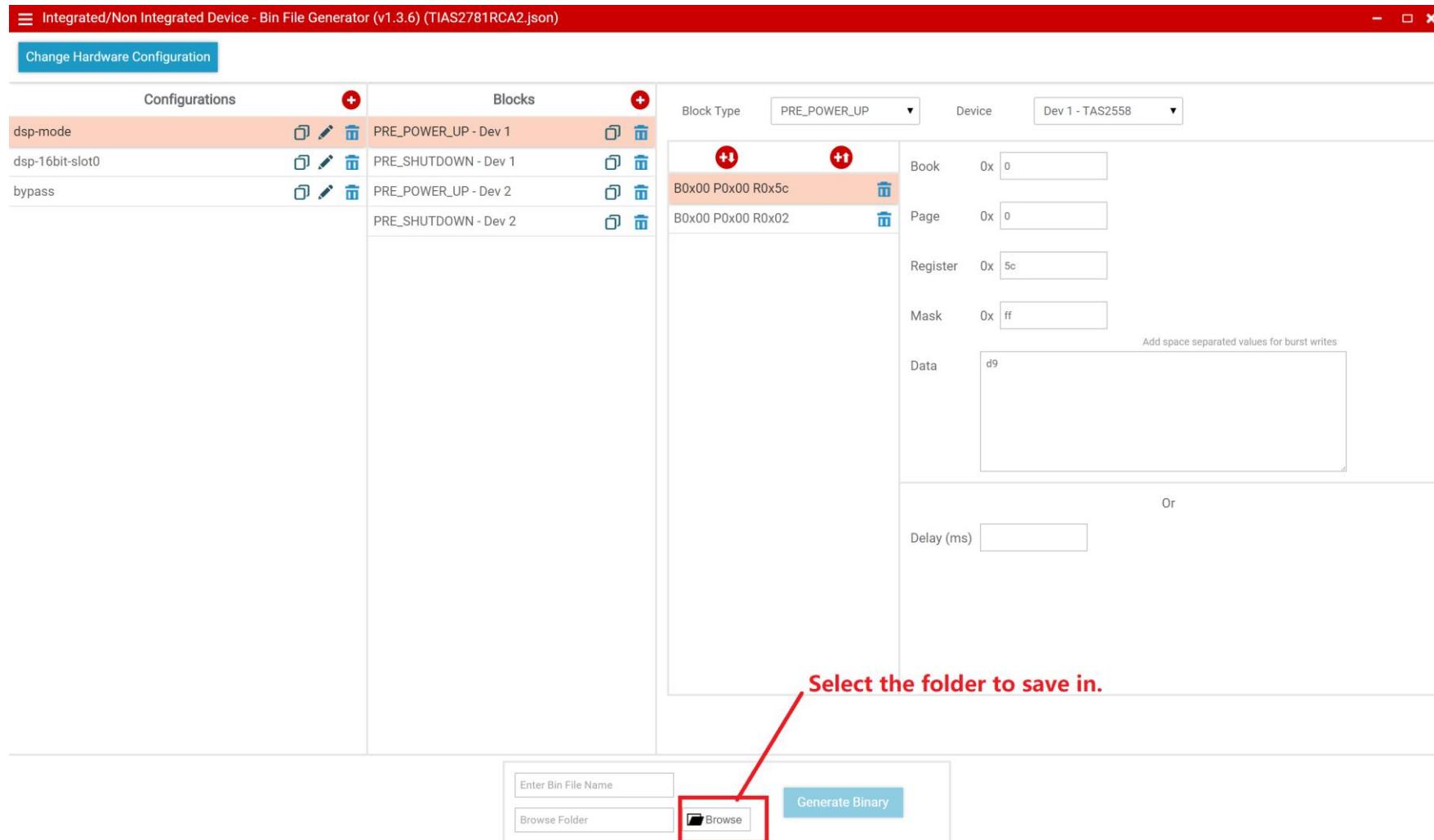
Data: d9  
Add space separated values for burst writes

Or

Delay (ms):

Select the folder to save in.

Enter Bin File Name:   
Browse Folder:  Generate Binary



# Bin File Making XXIV | Export the regbin binary file II

Integrated/Non Integrated Device - Bin File Generator (v1.3.6) (TIAS2781RCA2.json)

Change Hardware Configuration

Configurations	Blocks
dsp-mode	PRE_POWER_UP - Dev 1
dsp-16bit-slot0	PRE_SHUTDOWN - Dev 1
bypass	PRE_POWER_UP - Dev 2
	PRE_SHUTDOWN - Dev 2

Block Type: PRE\_POWER\_UP | Device: Dev 1 - TAS2558

B0x00 P0x00 R0x5c

Book: 0x 0 | Page: 0x 0 | Register: 0x 5c | Mask: 0x ff

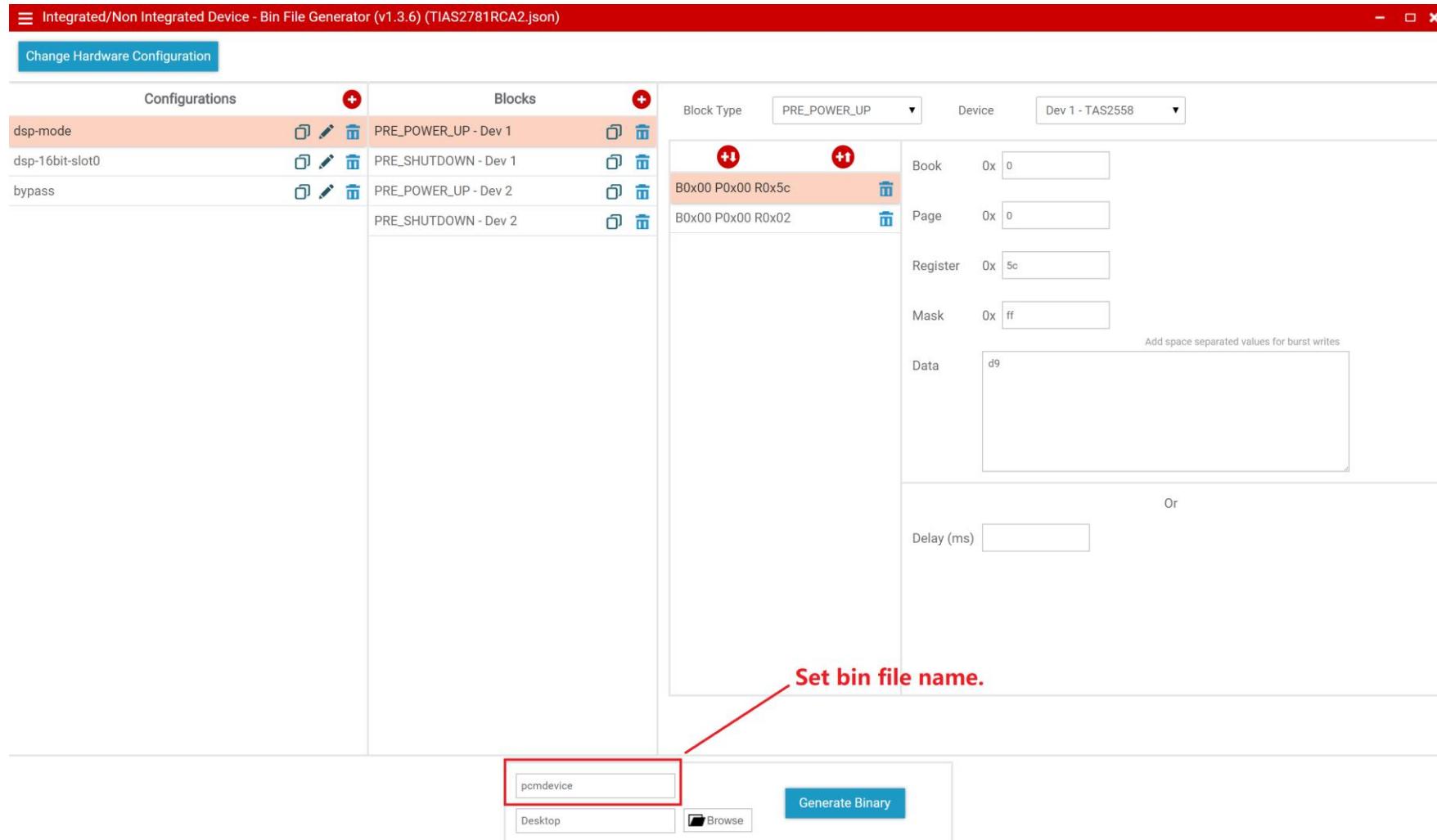
Data: d9  
Add space separated values for burst writes

Or

Delay (ms):

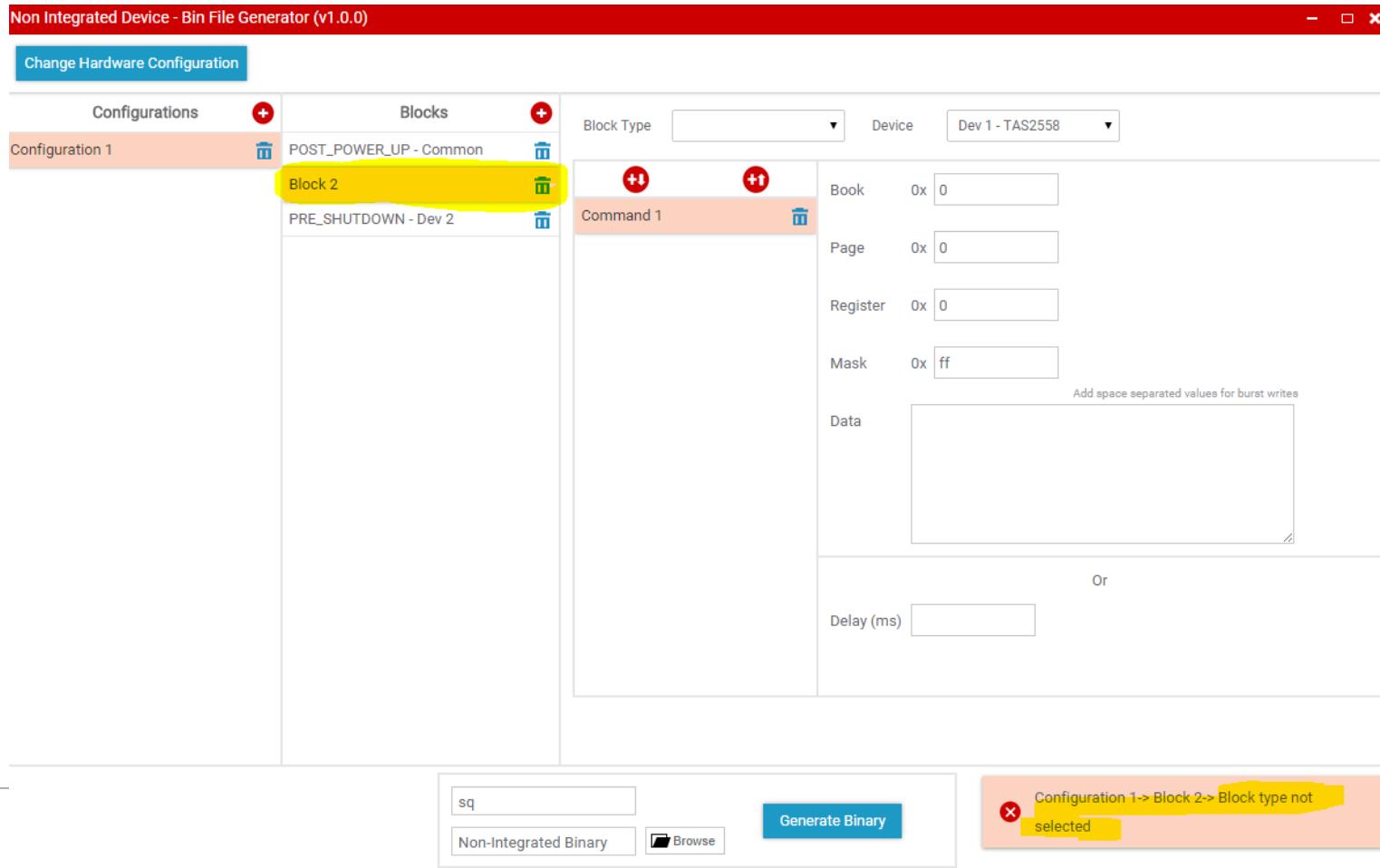
Set bin file name:  (highlighted with a red box)

Desktop  Generate Binary



# Bin File Making XXV | Data Validation Before Exporting Binary I

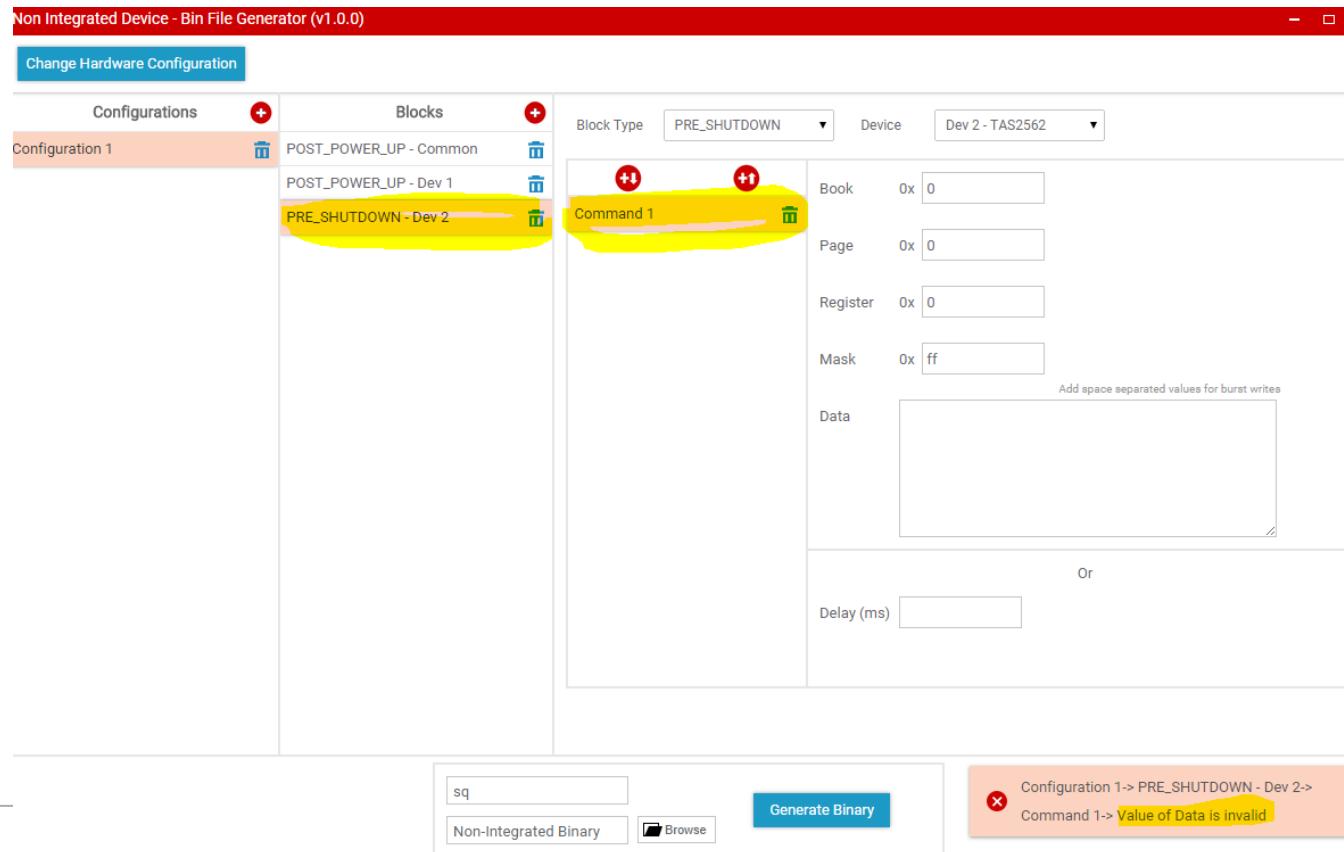
Try creating binary without providing block type for a block



1. The Block with Invalid data is selected
2. Error message shown with appropriate error

# Bin File Making XXVI -- Data Validation Before Exporting Binary II

Try creating binary without providing data or delay



1. The Block with Invalid data is selected
2. The Command with Invalid data is selected
3. Error message shown with appropriate error

# Import cfg file into the json file I

- For LPA device, json mainly stores the register setting.
- However, among MPA devices, json file stores not merely the register setting, but groups of filter coefficients, normally generated by PPC3 and saved like register setting in the cfg file. Developer usually spends much time in inputting these coefficients from cfg file into the json file manually. In order to simplify such heavy manual operation, a new feature to import the cfg file into json file has been adopted into regbin tool r1.4.1. See next page for detail.

# Import cfg file into the json file II

Integrated/Non Integrated Device - Bin File Generator (v1.3.8) (tas2563-1amp-reg-old.json)

Change Hardware Configuration

Configurations	Blocks
00-Music 16bit-16kHz-4slot	PRE_SHUTDOWN - Dev 1
01-calibration-auto-rate-16bit	PRE_POWER_UP - Dev 1
02-spk bypass -16k-4-slot- e...	
03-bypass-auto-rate-16bit-L...	
04-Music-16bit-auto-rate-i2s	
05-pdm-rec-i2s-48kHz-32bit...	
06-bypass-auto-rate-16bit-L	
07-bypass-auto-rate-16bit-R	

A new button has been added to import the filter coefficients or register settings in each Block in every Configuration.

Block Type: PRE\_POWER\_UP | Device: Dev 1 - TAS2562

Book: 0x 0 | Page: 0x 01 | Register: 0x 02 | Mask: 0x 1e  
Add space separated values for burst writes

Data: 0

Or

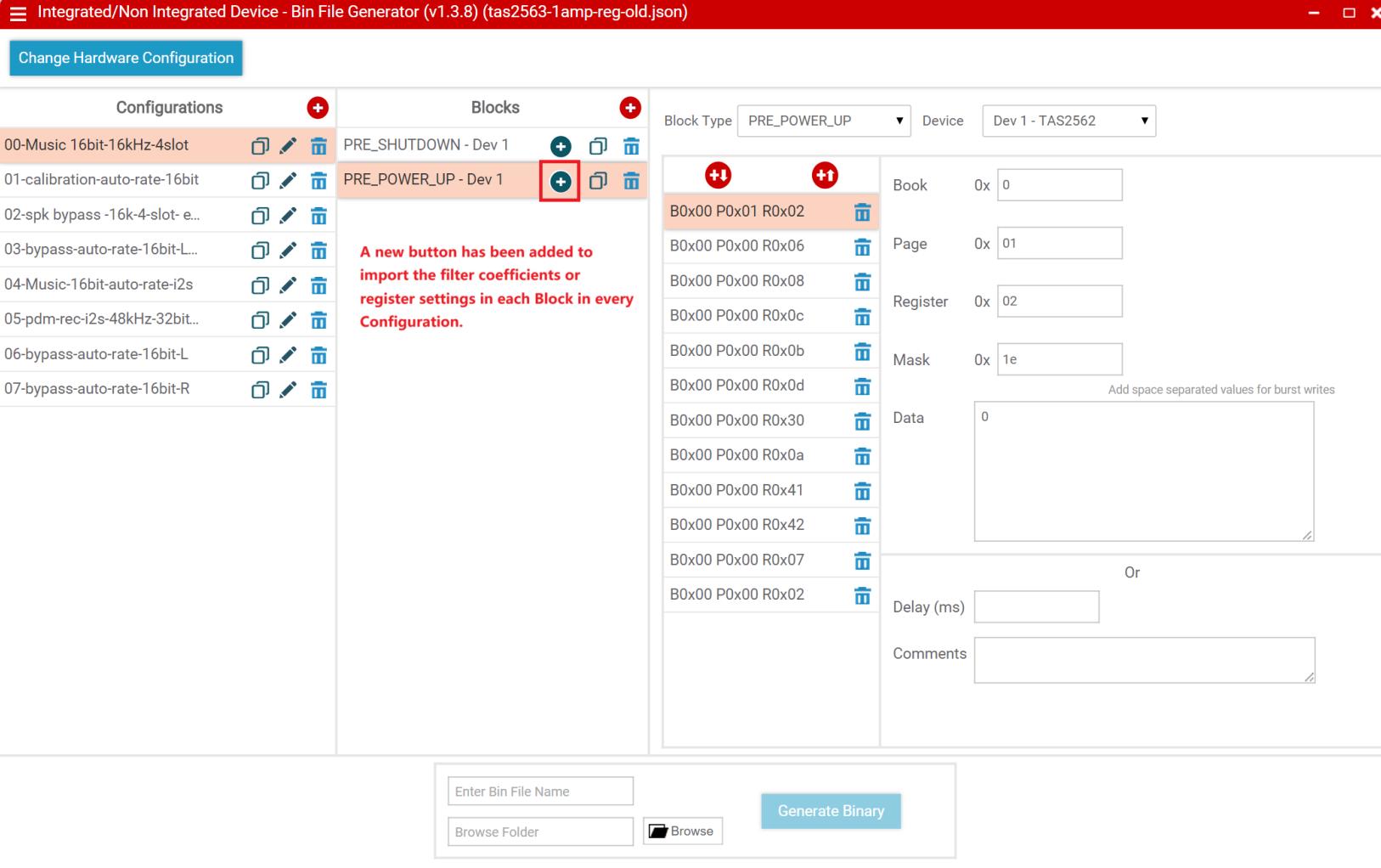
Delay (ms): [ ]

Comments: [ ]

Enter Bin File Name: [ ]

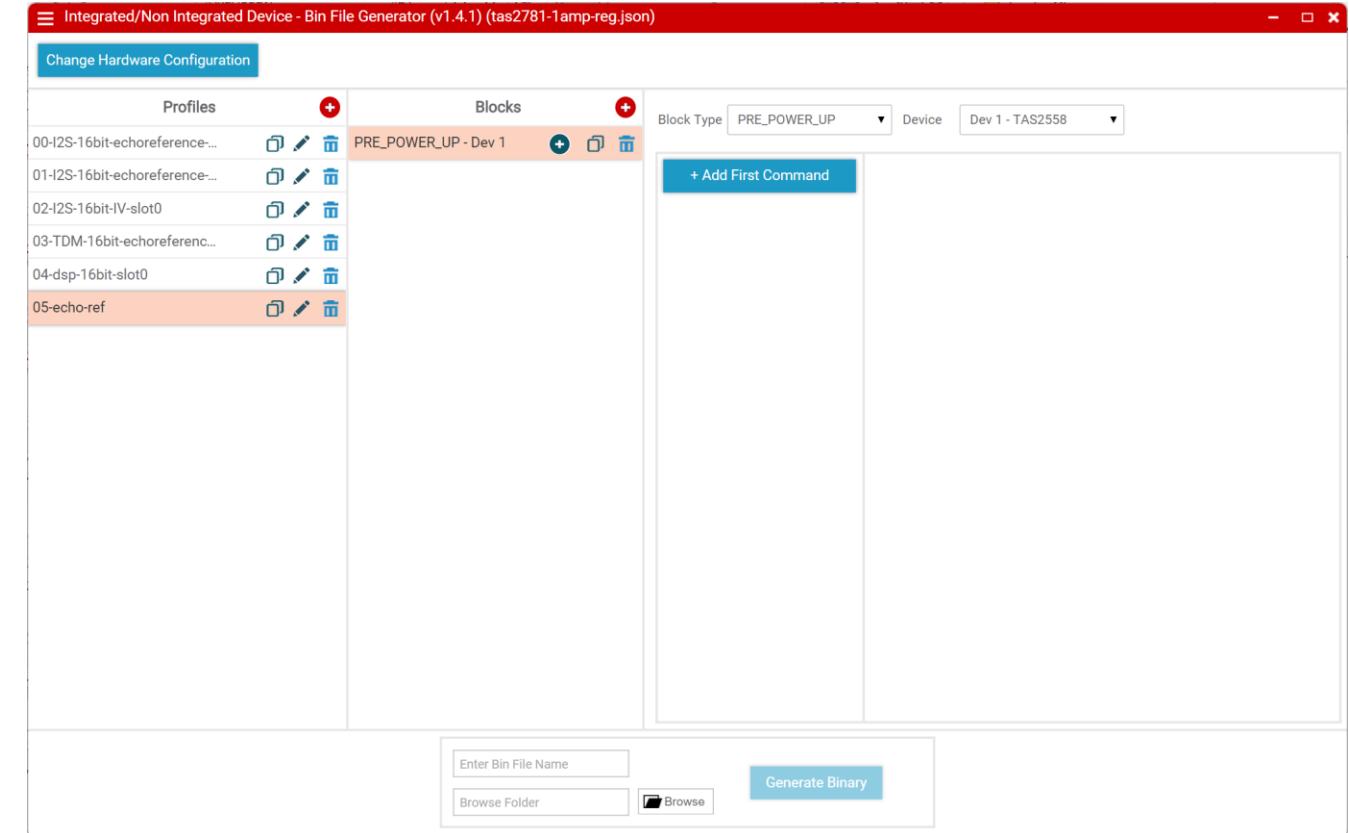
Browse Folder: [ ]

Generate Binary: [ ]

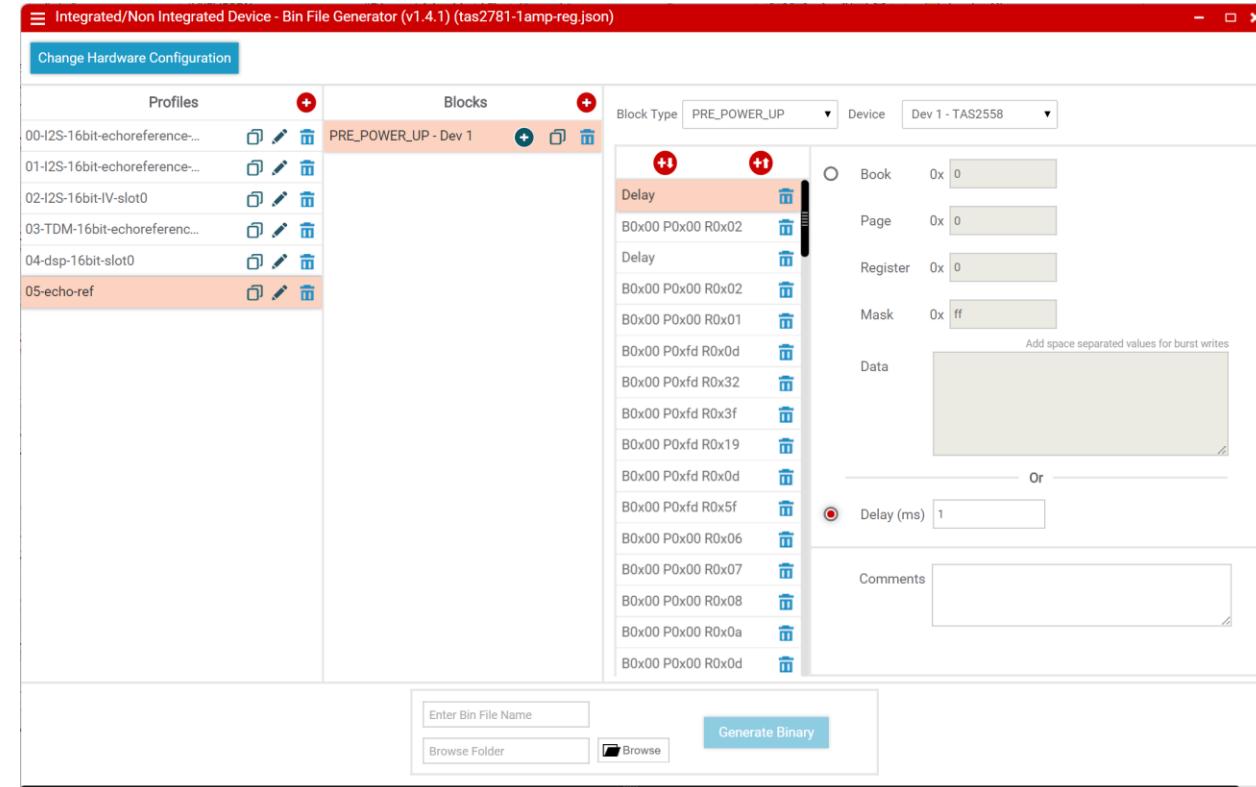
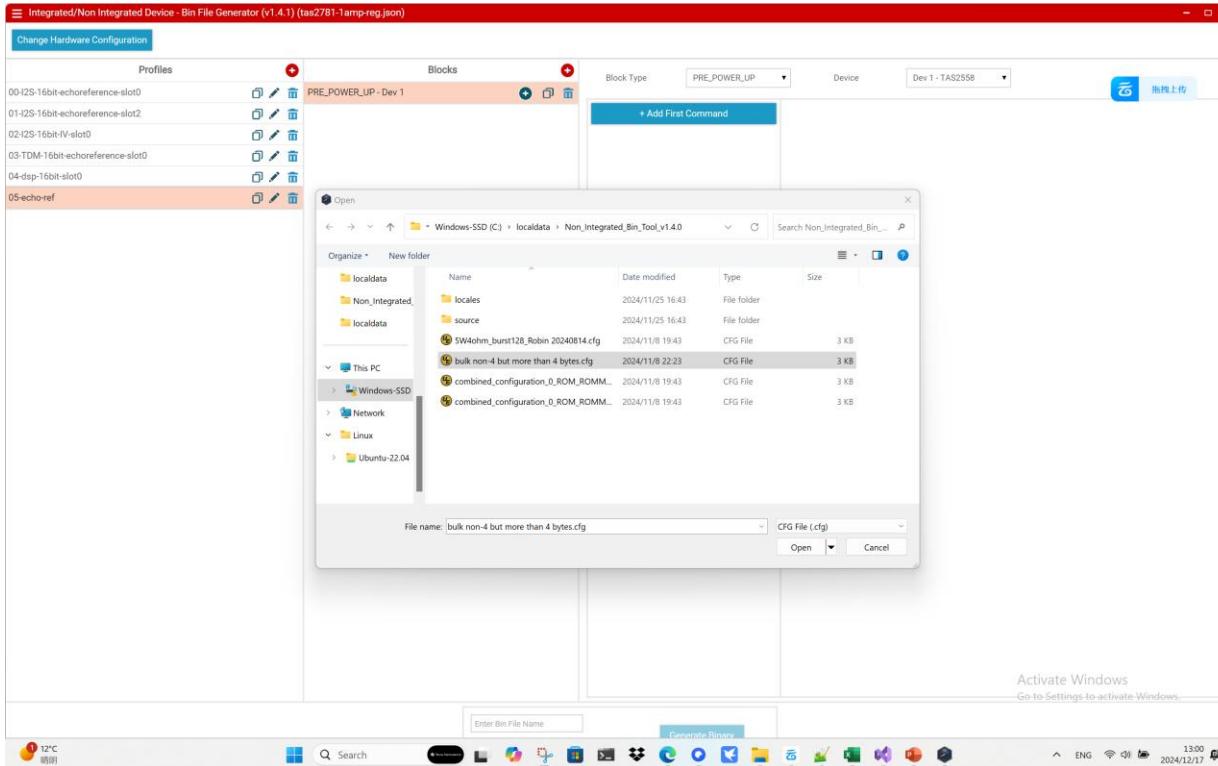


# Import cfg file into the json file III | Empty block I

- If the block is an empty one, import cfg file directly.

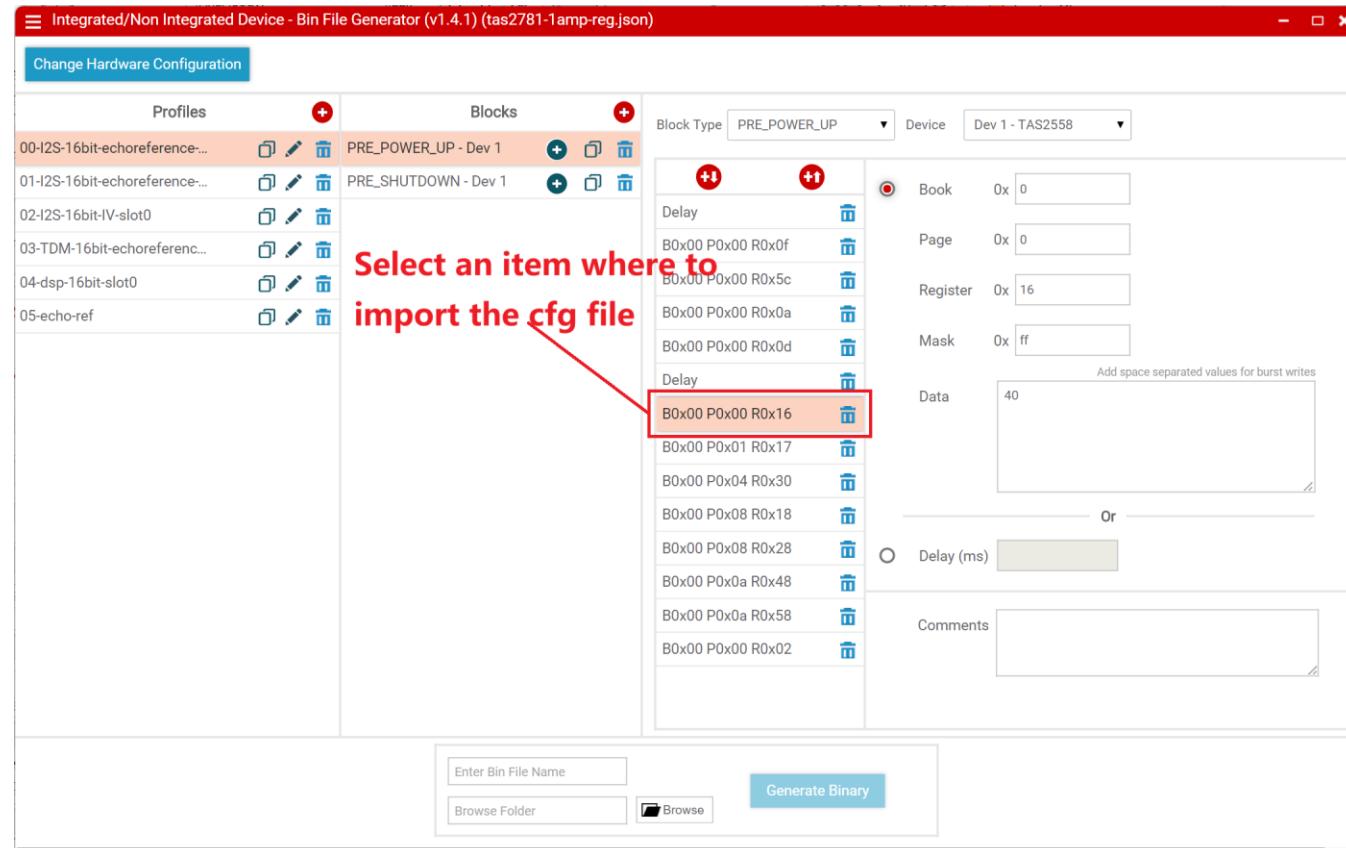


# Import cfg file into the json file IV | Empty block II

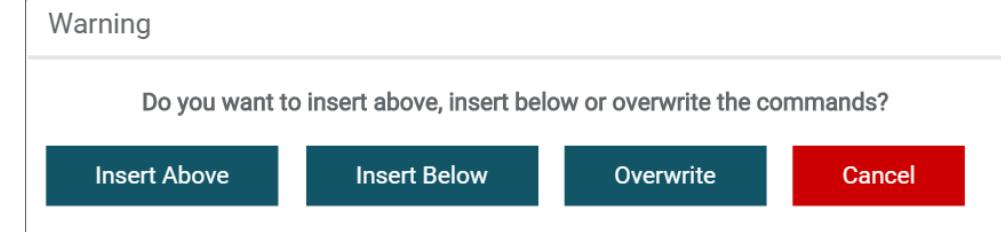
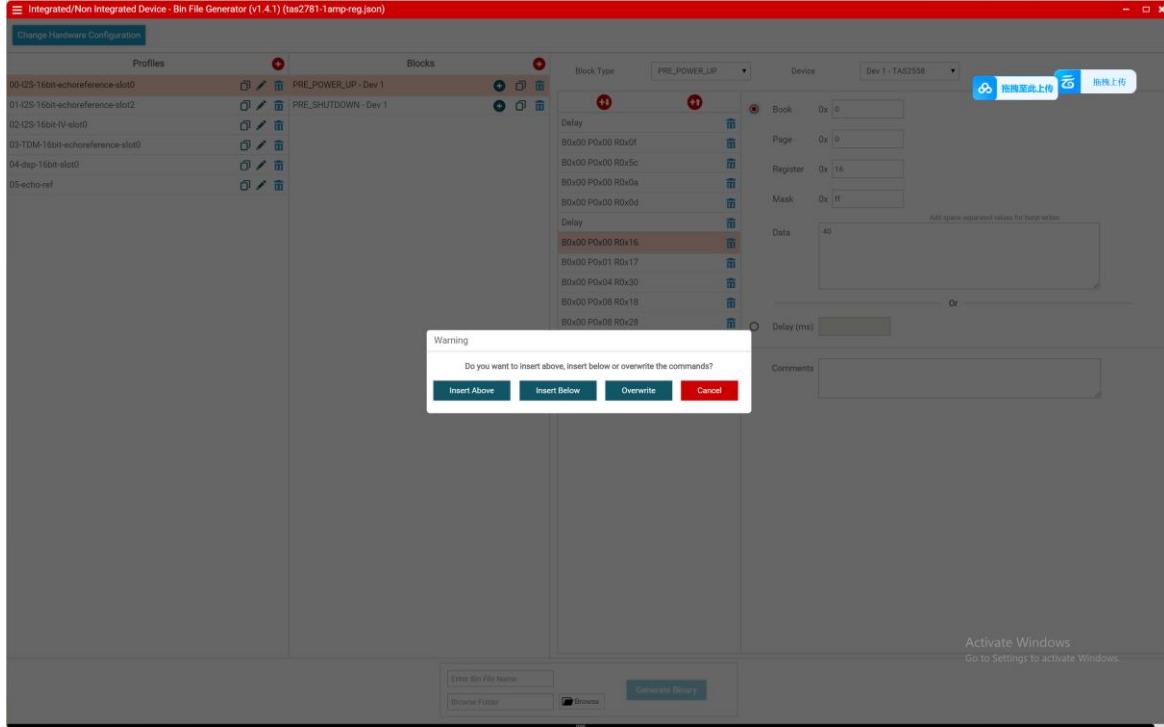


# Import cfg file into the json file V | non-empty block I

- If the block is an non-empty one, select an item in the block first.



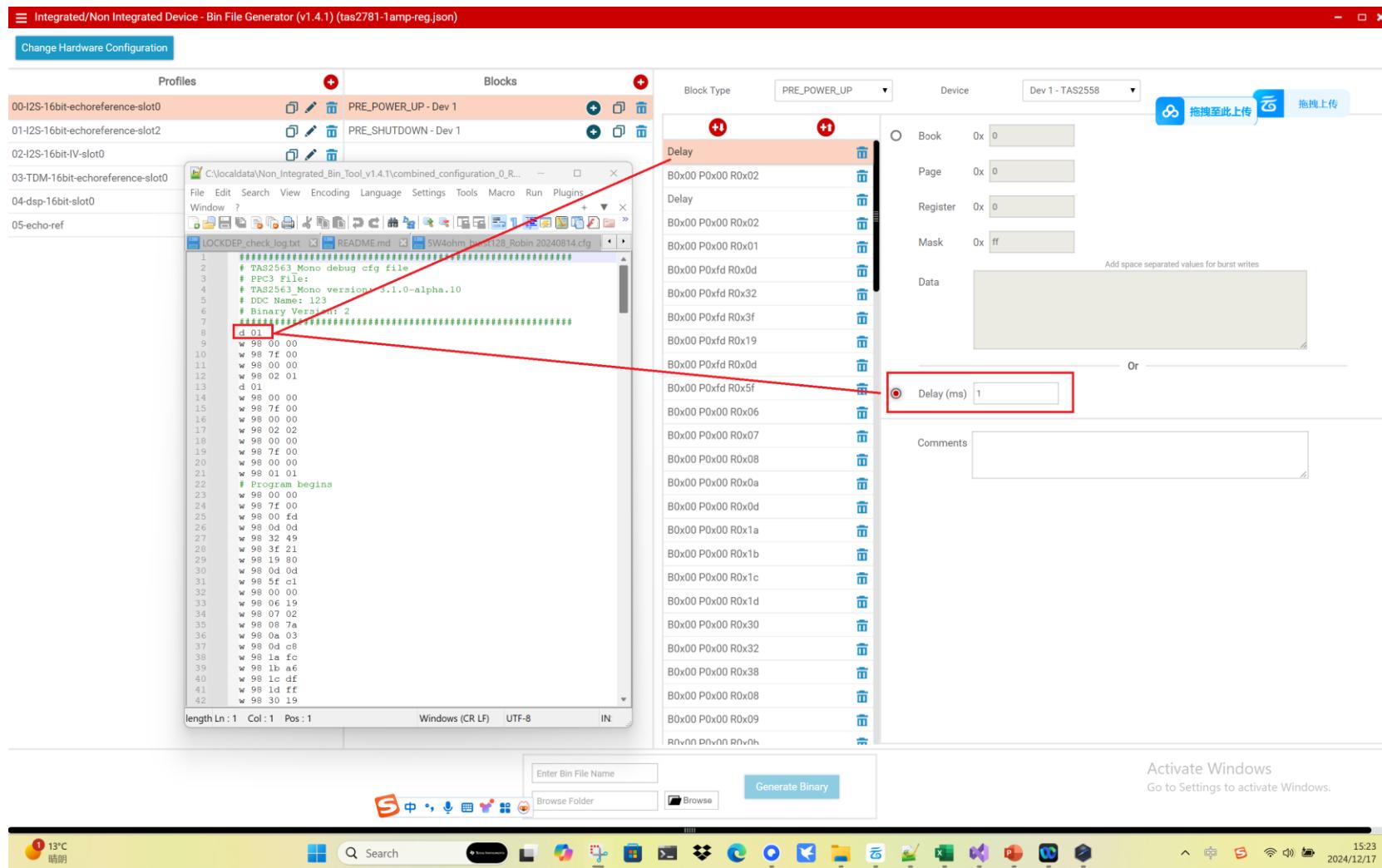
# Import cfg file into the json file VI | non-empty block II



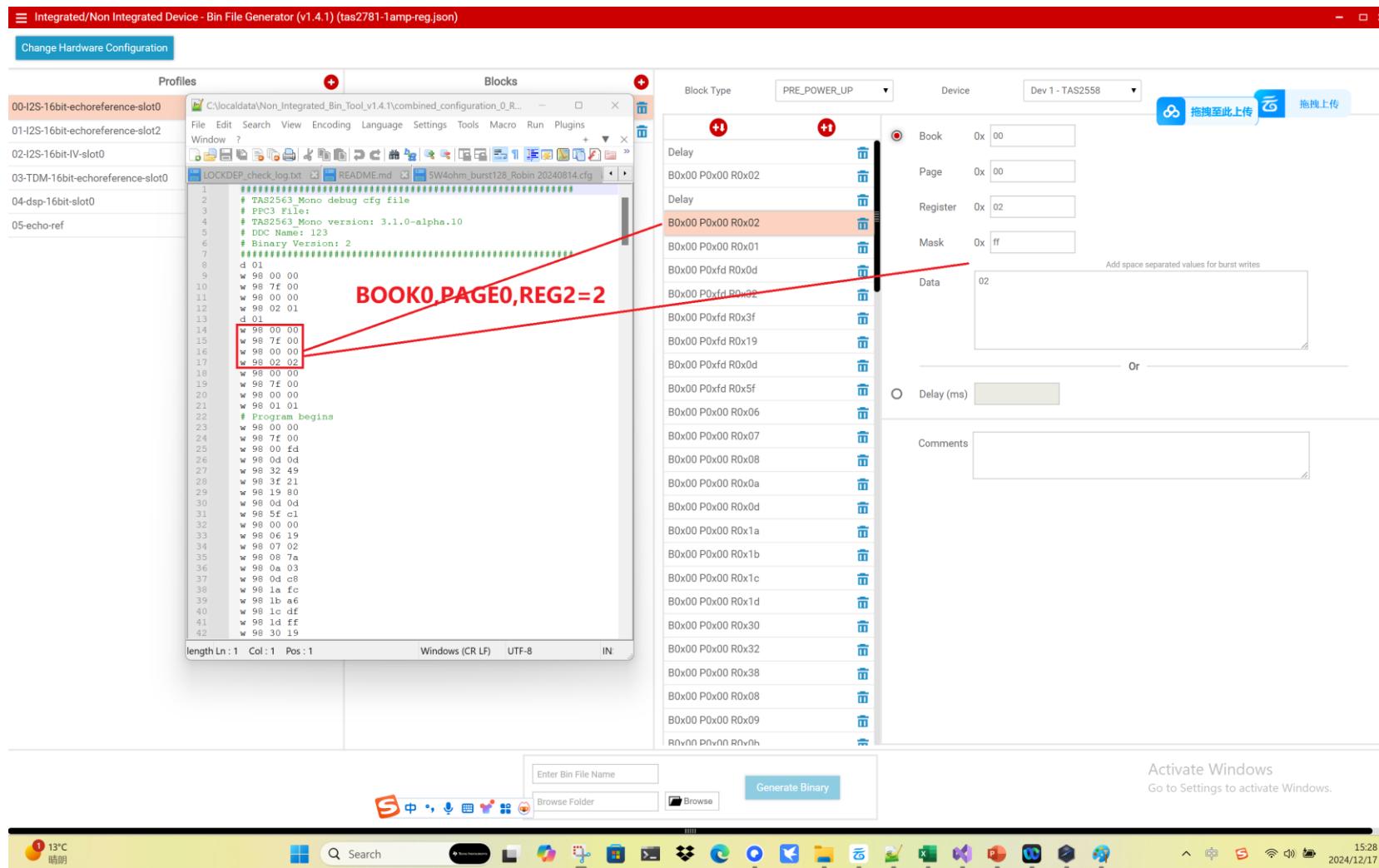
# Import cfg file into the json file VII | non-empty block III

- Insert Above: import the cfg file above the selected item.
- Insert Below: import the cfg file below the selected item.
- Overwrite: clear the block and import the cfg file, the block do not keep the old setting any more.

# Relationship between cfg commands and regbin setting I | Delay

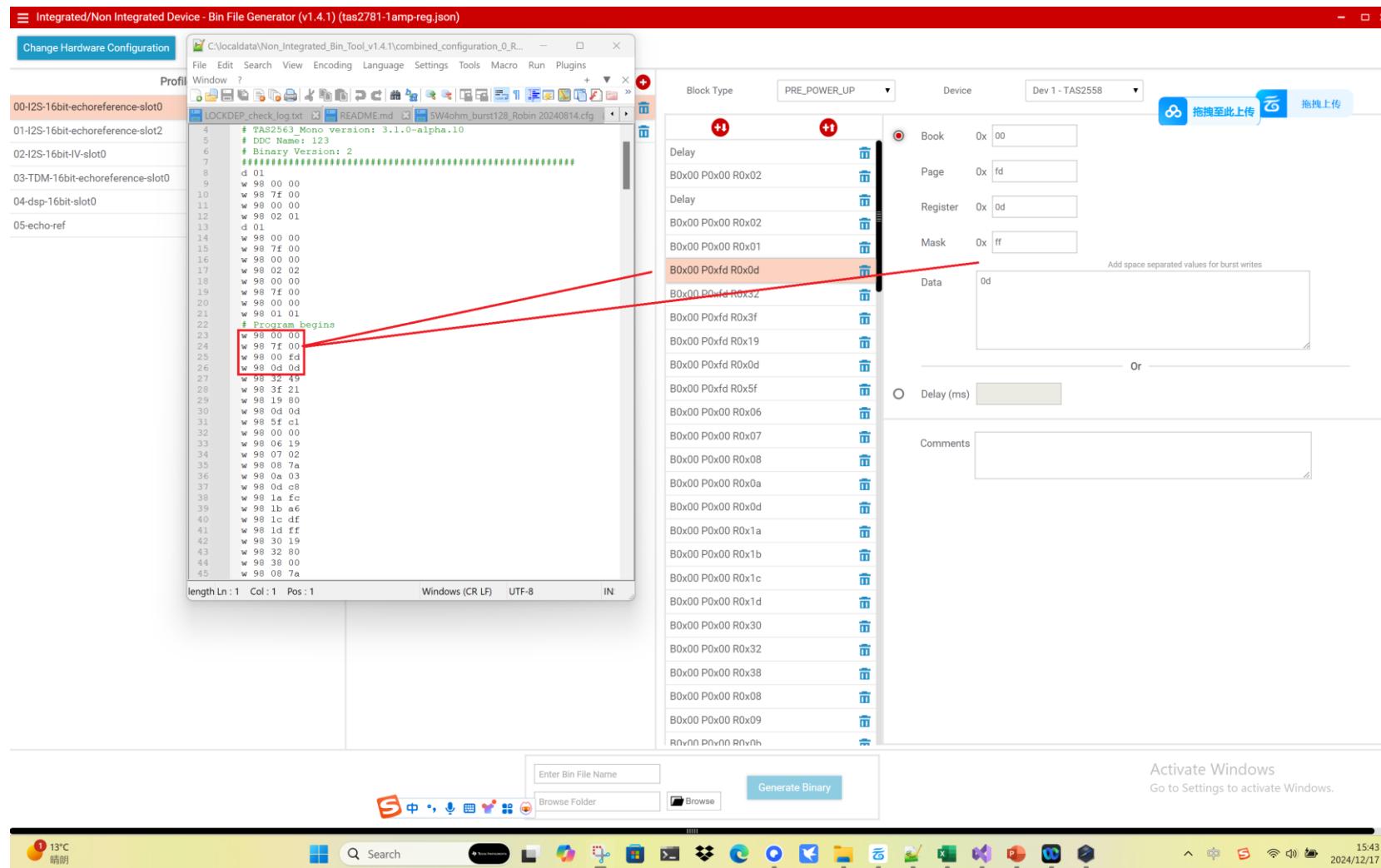


# Relationship between cfg commands and regbin setting II | Single byte write

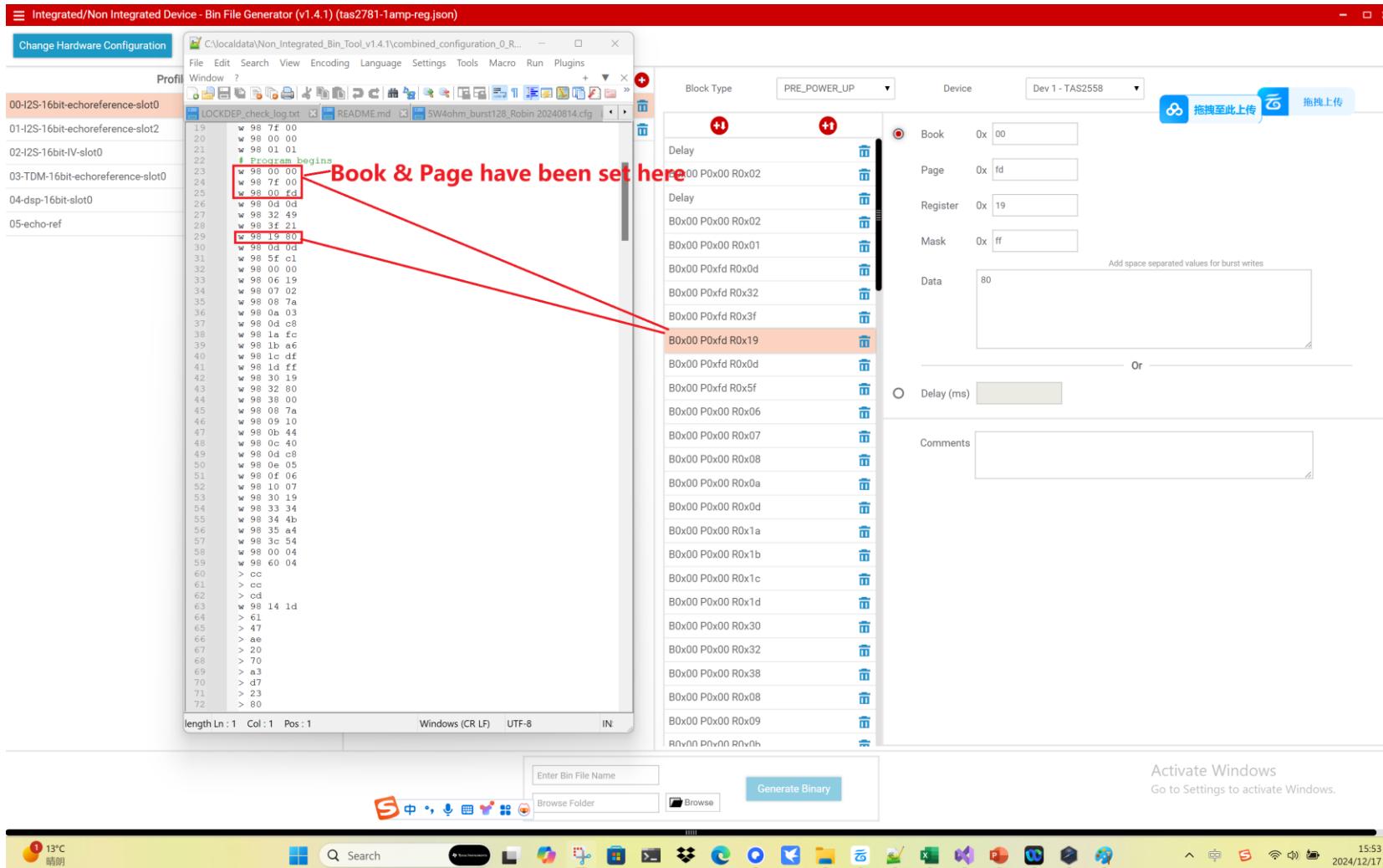


# Relationship between cfg commands and regbin setting III | Single byte write

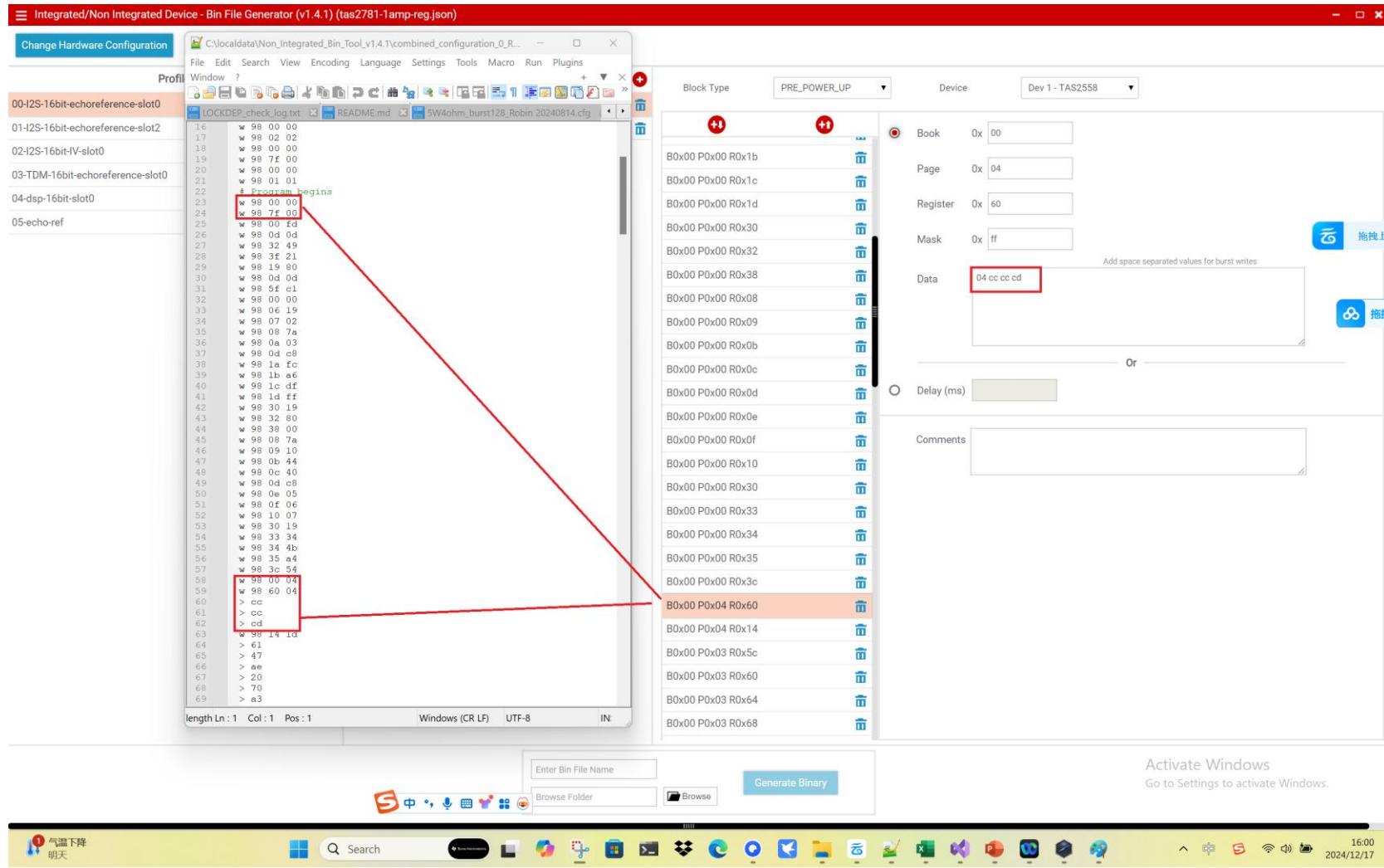
## II



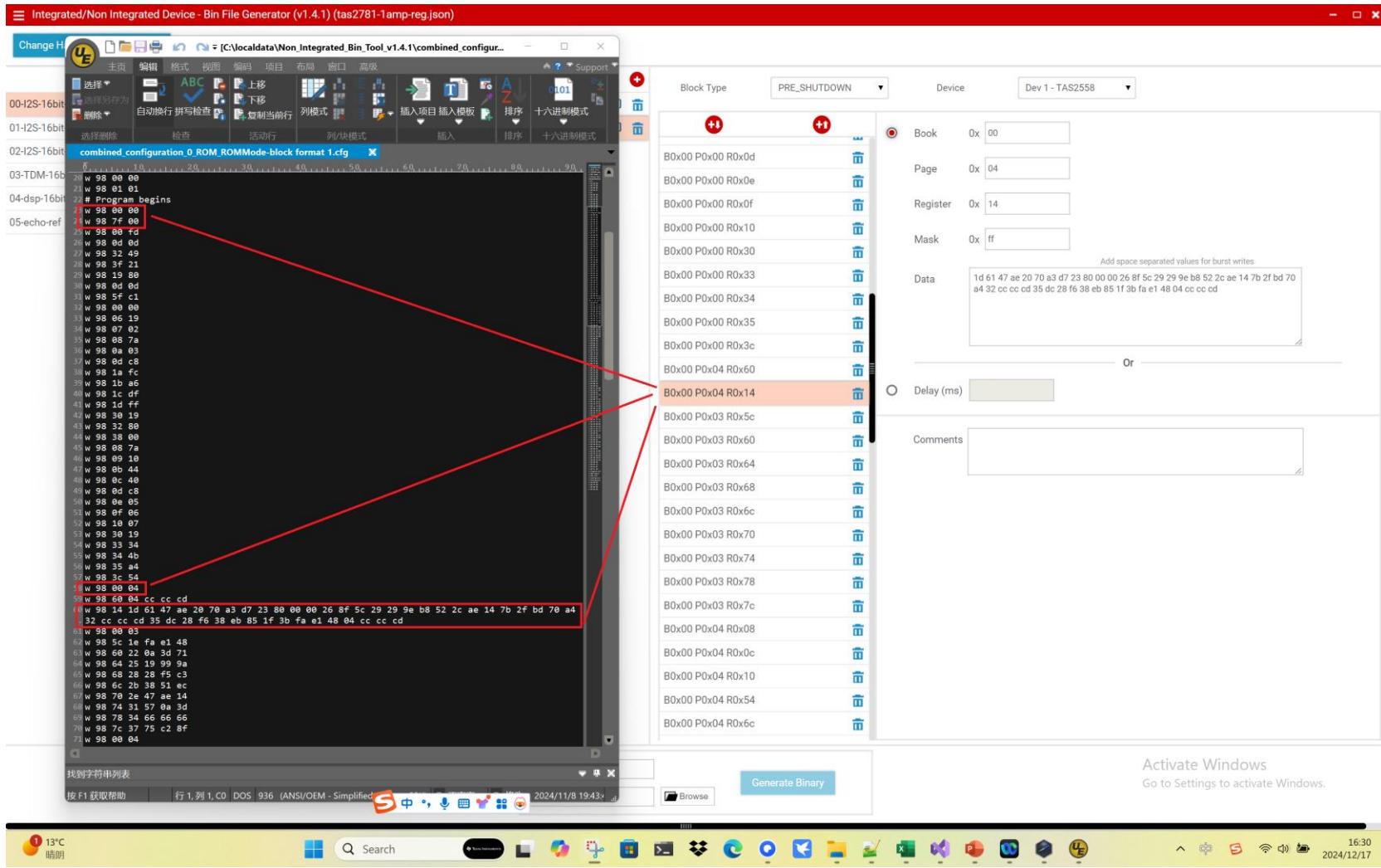
# Relationship between cfg commands and regbin setting IV | Single byte write III



# Relationship between cfg commands and regbin setting V | Bulk write I



# Relationship between cfg commands and regbin setting V | Bulk write II



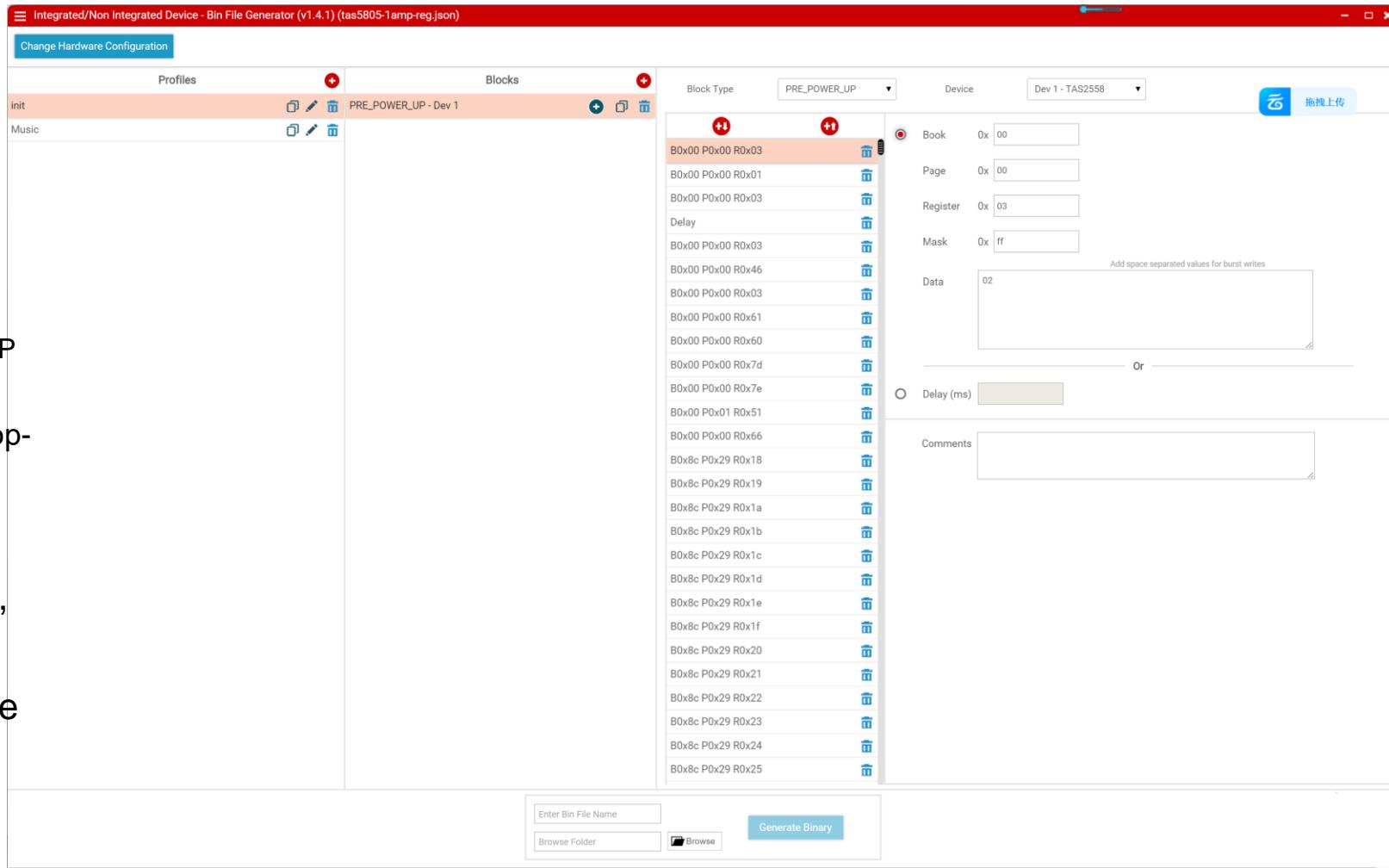
# init

tasdevice driver code supports extra initialization setting.

1. Define the key word “init” in the profile name list.
2. Set the block type as PRE\_POWER\_UP  
PS: code only supports PRE\_POWER\_UP
3. set the init setting in the third column
4. Don’t forget to set the device on the top-left of the right picture.

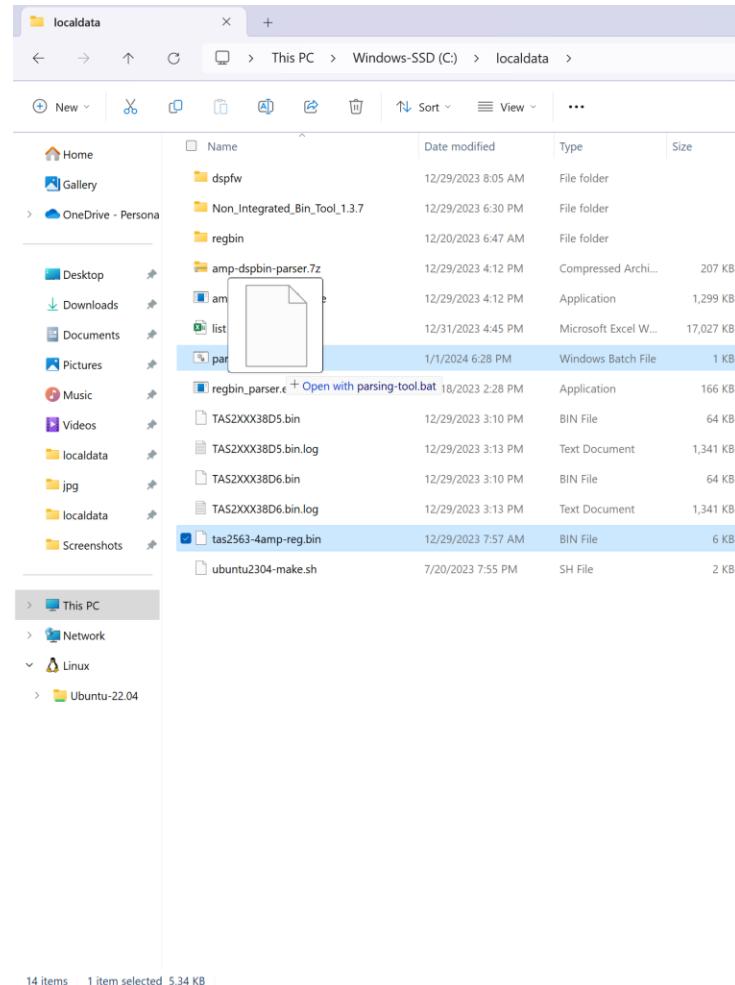
PS:

- If there’s two init profiles in the bin file, Only the smaller init profile id works.
- If no init profile in the bin file, tasdevice won’t execute initialization.

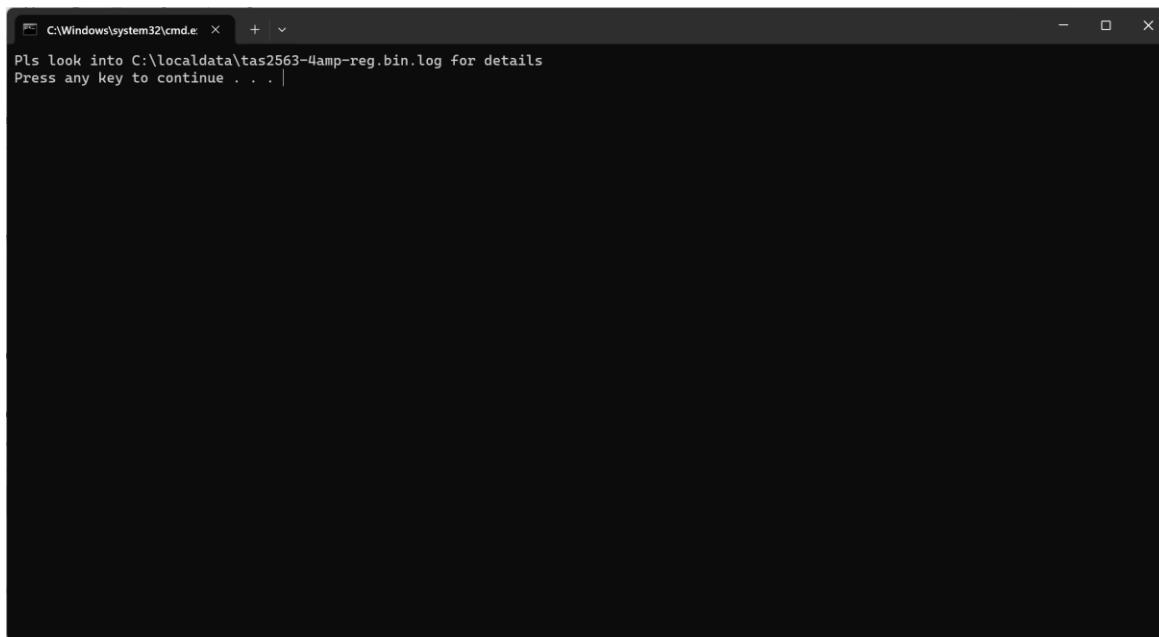


# regbin\_parser.exe & parsing-tool.bat I

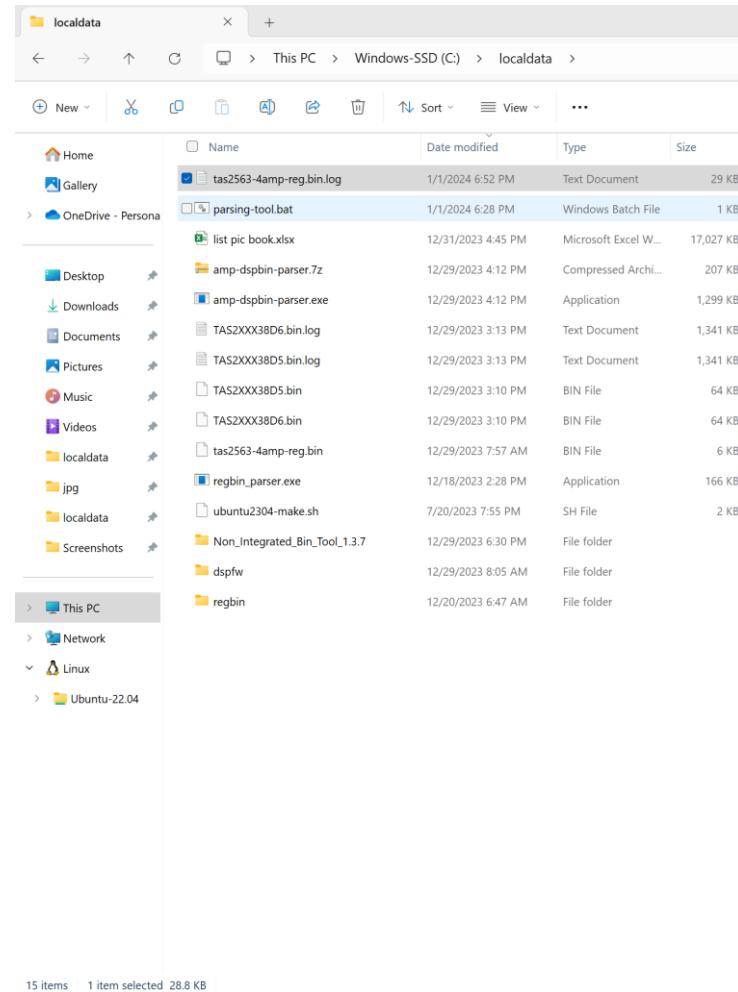
- Toolset for regbin parsing.
- Operations:
  1. Confirm parsing-tool.bat & regbin\_parser.exe are in the same folder
  2. Drag the regbin file onto the parsing-tool.bat, then release, see the right picture, tas2563-4amp-reg.bin is dragged onto the parsing-tool.bat.
  3. It will popup a command-line screen, pls check the output, the left picture next page shows a successfully parsing
  4. The log of regbin file will be generated, see the right pic next page, tas2563-4amp-reg.bin.log has already been generated.



# regbin\_parser.exe & parsing-tool.bat II



```
C:\Windows\system32\cmd.e
Pls look into C:\localdata\tas2563-4amp-reg.bin.log for details
Press any key to continue . . .
```



## Attentions

- The range of input for “TASDEVICE Profile id” depends on the sum of configs in Regbin file. It begins with 0.
- Don’t forget to restore the related registers after shutdown
- Kcontrol -- “TASDEVICE Profile id” is dynamically created if valid bin file is in the system; If no regbin file in the system, kcontrol – “TASDEVICE Profile id” won’t be created;
- No side-effect to the device and code if there still exists “TASDEVICE Profile id” in mixer\_paths.xml.

# How to use reg-bin file?

- Put the regbin file into /vendor/firmware, /vendor/etc or /lib/firmware, which location is up to the platform setting
- Three ways to config
  1. tinymix or mixer command
    - tinymix “TASDEVICE Profile id” 0 ---- bypass regbin
    - tinymix “TASDEVICE Profile id” 1 ---- select config 1 setting
    - tinymix “TASDEVICE Profile id” 2 ---- select config 2 setting
  2. mixer\_path.xml
    - e.g.
    - <ctl name=“TASDEVICE Profile id” value=“0”/> ---- bypass regbin
    - <ctl name=“TASDEVICE Profile id” value=“1”/> ---- select config 1 setting
  3. Mixer API: mixer\_get\_ctl\_by\_name and mixer\_ctl\_set\_value

# Case 1 | Swap channels On Project J20C

- Conf 1 – device 1 is left channel, device 2 is right channel
- Conf 2 – device 1 is right channel, device 2 is left channel
- In this case, the switching among conf 0, 1 and 2 by mixer path.



J20C



control\_list.PNG

## Case 2 | GREG and PRAM check on factorytest

- Conf 1 – GREG check
- Conf 2 – PRAM check
- some register setting will be restored during post-shutdown
- the switching among conf 0, 1 and 2 by mixer API.



test\_screening



tas256x\_reg.bin

## Case 3 | floor noise in receiver mode

- Conf 2 – receiver mode
- some register setting will be restored during post-shutdown
- the switching among conf 0 ~ 2 by mixer API or mixer\_path.xml.

```
Firmware init complete
main Bin loading requested: 0
conf 1
    block type:PRE_POWER_UP  device idx = 0x01
        SINGLE BYTE:
            BOOK0x00 PAGE0x00 REG0x30 VALUE = 0x1d
conf 2
    block type:PRE_POWER_UP  device idx = 0x01
        SINGLE BYTE:
            BOOK0x00 PAGE0x00 REG0x30 VALUE = 0x1d
            BOOK0x00 PAGE0x0d REG0x0d VALUE = 0x0d
            BOOK0x00 PAGE0xfd REG0x12 VALUE = 0xc0
            BOOK0x00 PAGE0xfd REG0x46 VALUE = 0x1f
            BOOK0x00 PAGE0xfd REG0x32 VALUE = 0x08
            BOOK0x00 PAGE0x00 REG0x3e VALUE = 0x00
            BOOK0x00 PAGE0x00 REG0x04 VALUE = 0xca
            BOOK0x00 PAGE0x01 REG0x21 VALUE = 0x08
    block type:POST_SHUTDOWN      device idx = 0x01
        SINGLE BYTE:
            BOOK0x00 PAGE0xfd REG0x0d VALUE = 0x0d
            BOOK0x00 PAGE0xfd REG0x12 VALUE = 0x00
            BOOK0x00 PAGE0xfd REG0x46 VALUE = 0x1f
            BOOK0x00 PAGE0xfd REG0x32 VALUE = 0x28
            BOOK0x00 PAGE0x00 REG0x3e VALUE = 0x10
            BOOK0x00 PAGE0x00 REG0x04 VALUE = 0xcf
```

## To be Improved I | bin file

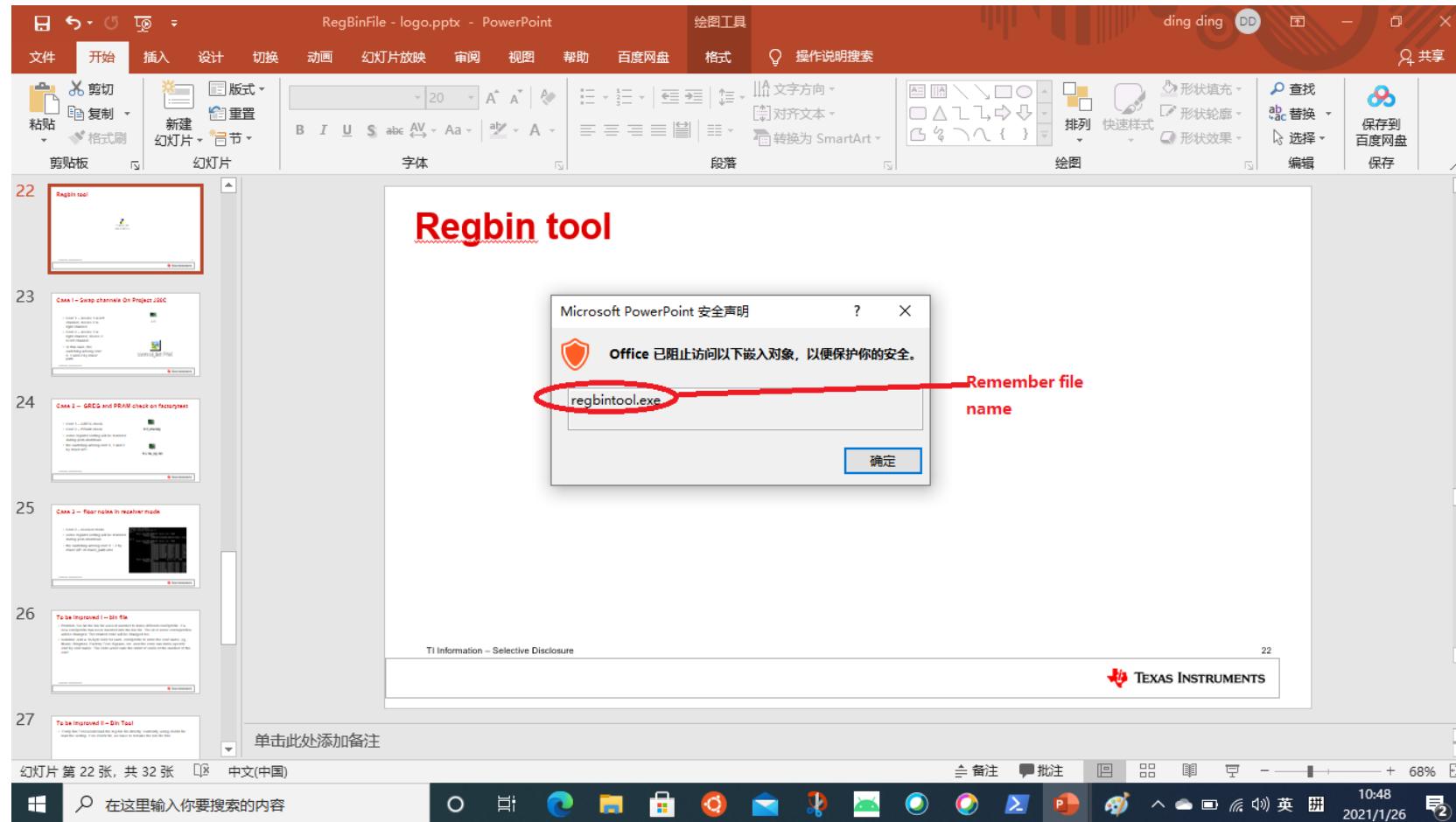
- Problem: So far the bin file uses id number to index different conf/profile. If a new conf/profile has been inserted into the bin file. The id of some confs/profiles will be changed. The related code will be changed too.
- Solution: Add a 16-byte field for each conf/profile to store the conf name, eg, Music, Ringtone, Factory Test, Bypass, etc. And the code can index specific conf by conf name. The code won't care the order of confs or the number of the conf.

## To be Improved II | Bin Tool

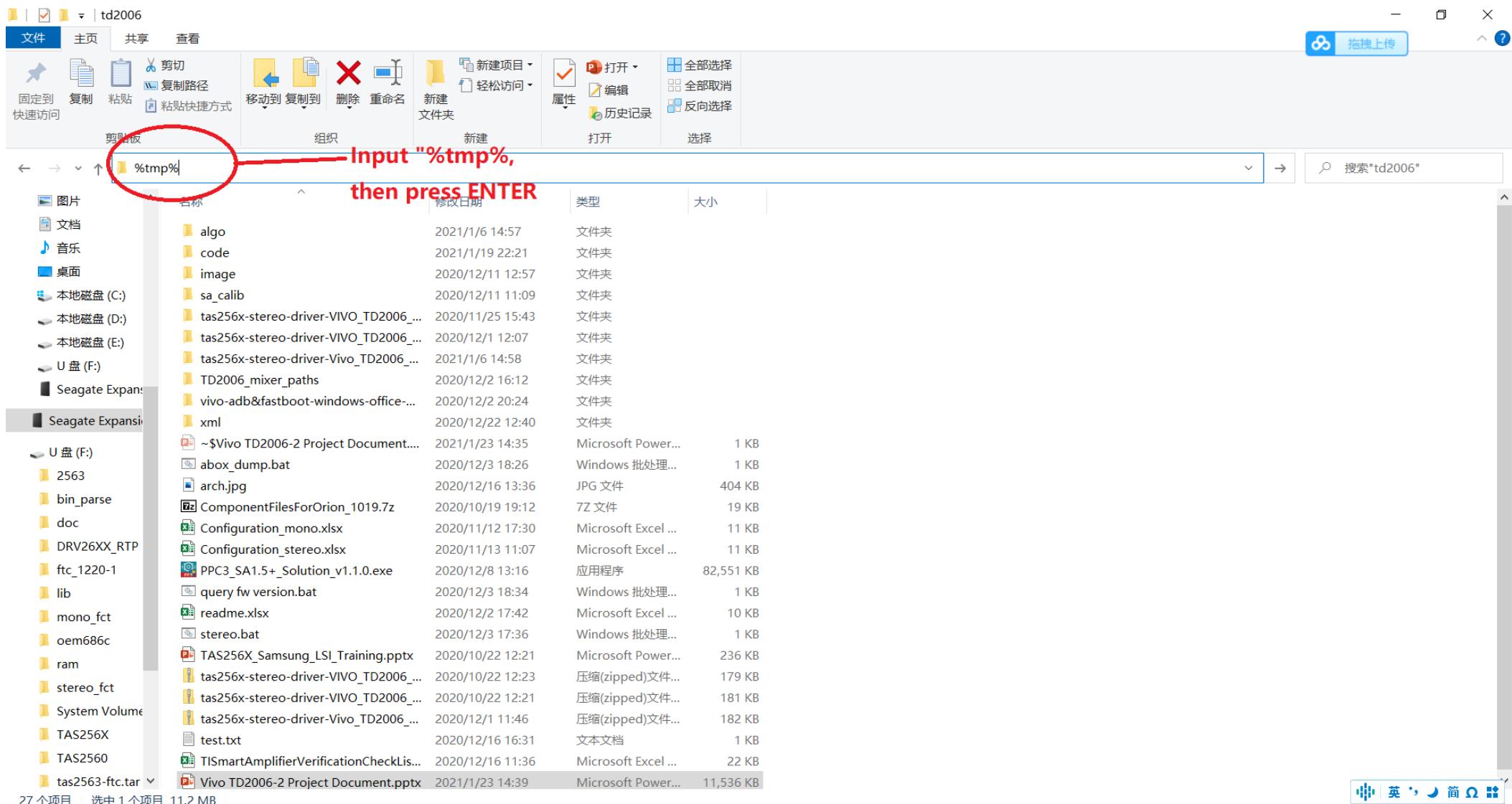
- If only Bin Tool would load the reg bin file directly. Currently, using JSON file load the setting. If no JSON file, we have to remake the bin file first.

# Appendix I

If the file object can't be opened, pls search the file into %tmp% , see the following picture.



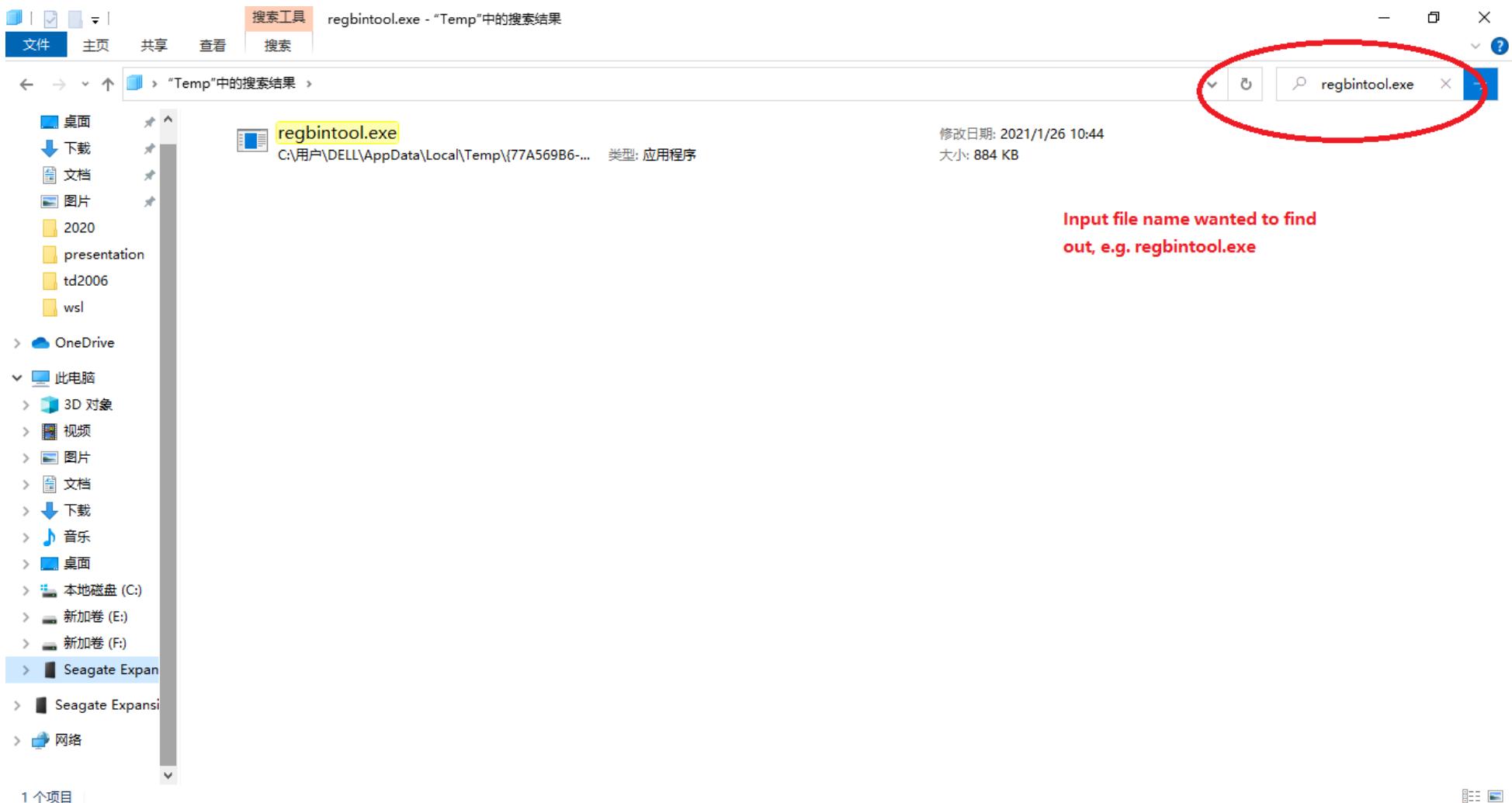
# Appendix II



27 个项目 选中 1 个项目 11.2 MB

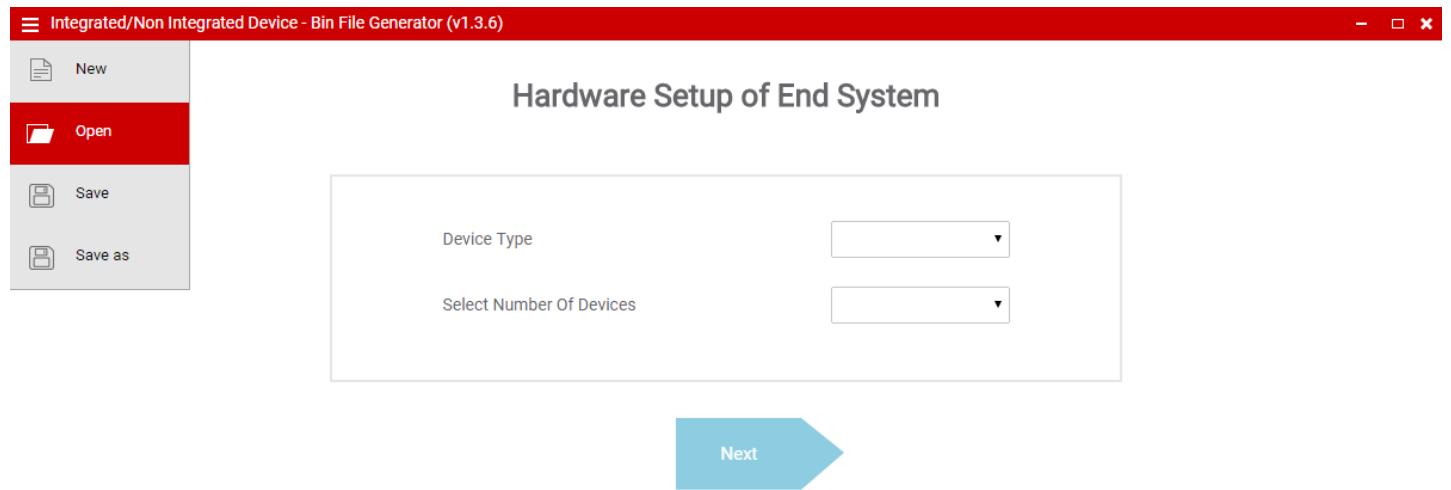
57

# Appendix III



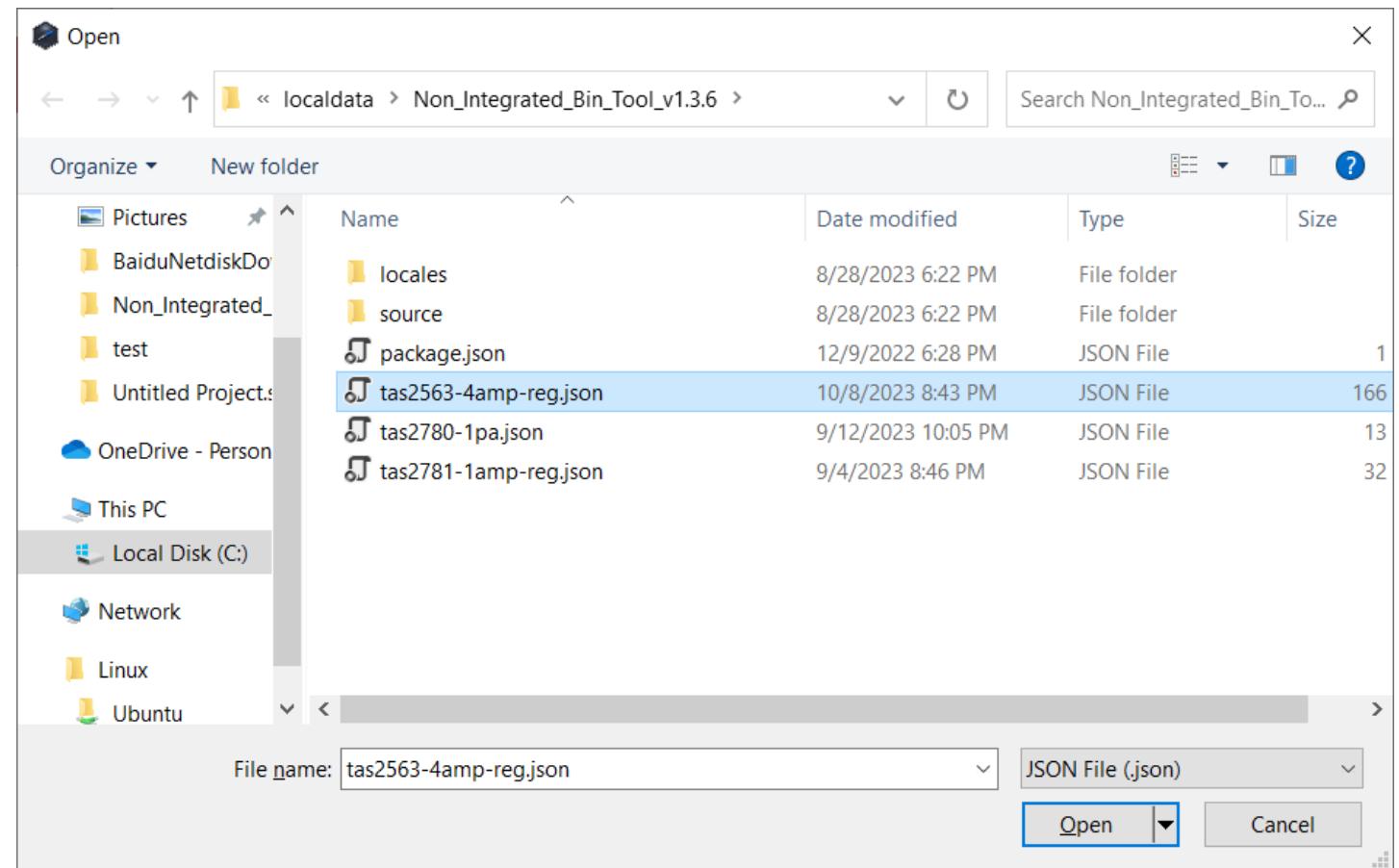
# Appendix IV | How to generate bin file from json file I

- Sometime Jason file is offered with the code, so conversion from Jason file to bin file is the mandatory job.
- STEP I: see the right pic.



## Appendix V | How to generate bin file from json file II

- STEP II: Select Jason file to be imported. See the right pic.



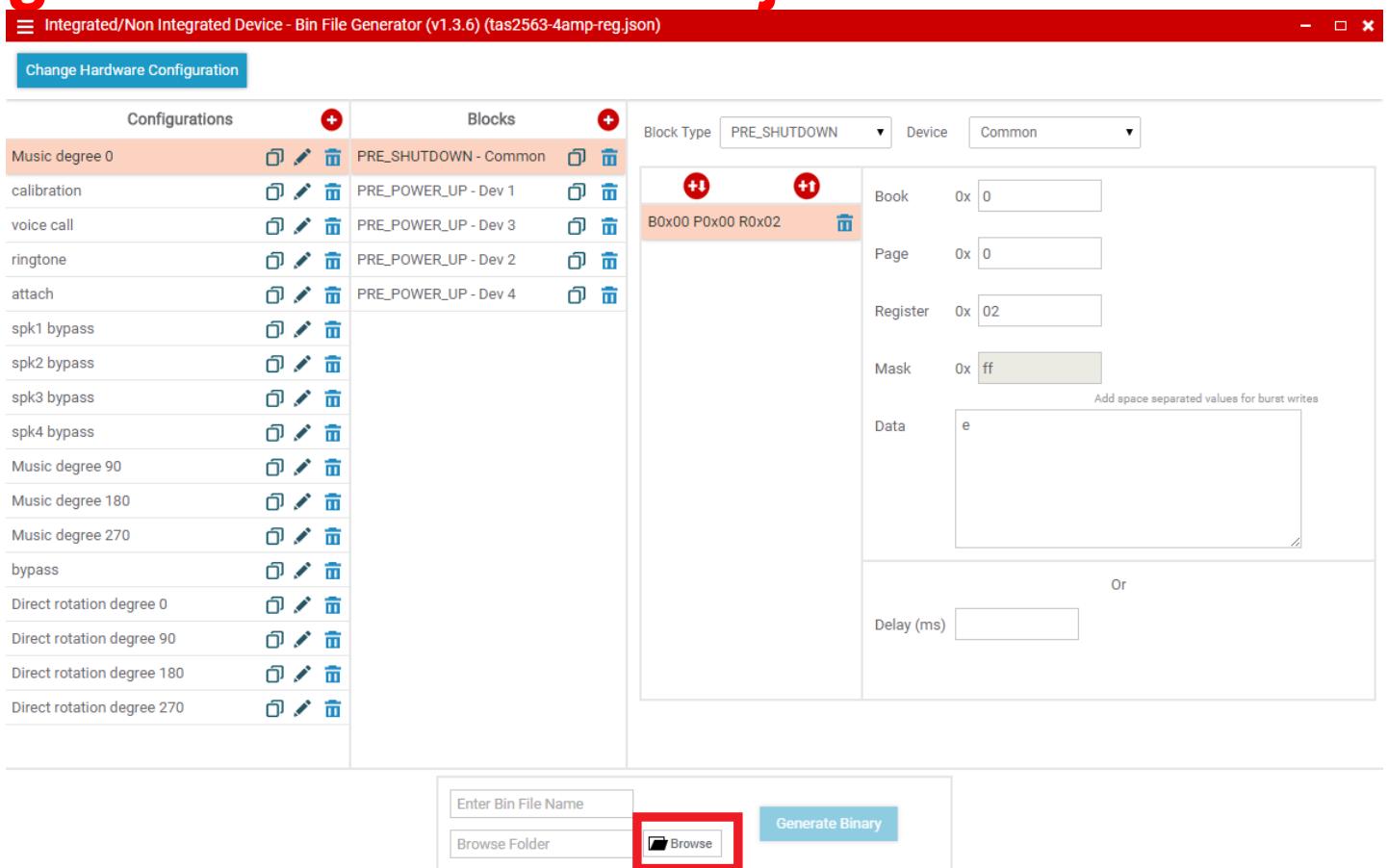
# Appendix VI | How to generate bin file from json file III



- STEP III: Click the “Next” button. See the right pic.

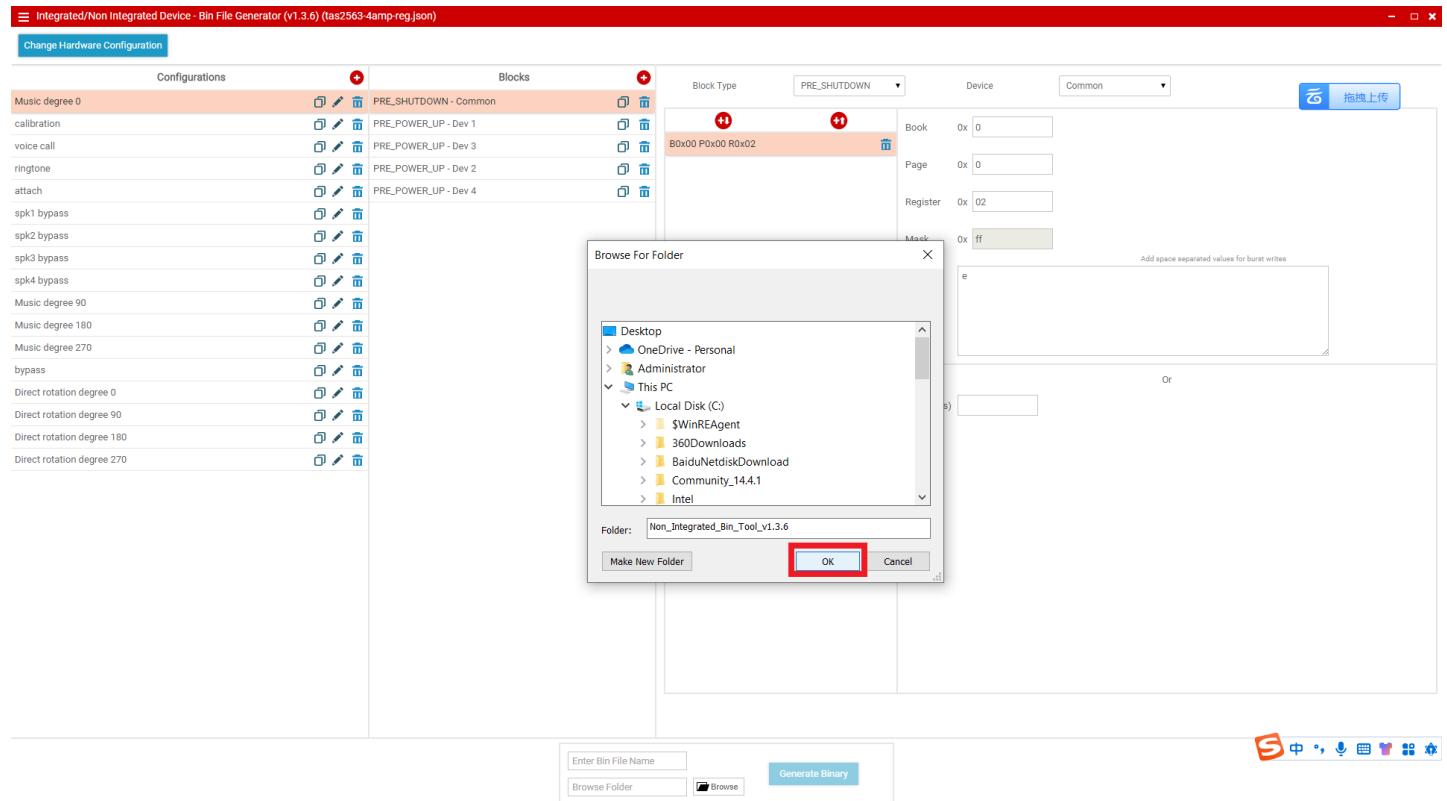
# Appendix VII | How to generate bin file from json file IV

- STEP IV: Click the “Browse” button to select a folder to save bin file. See the right pic.



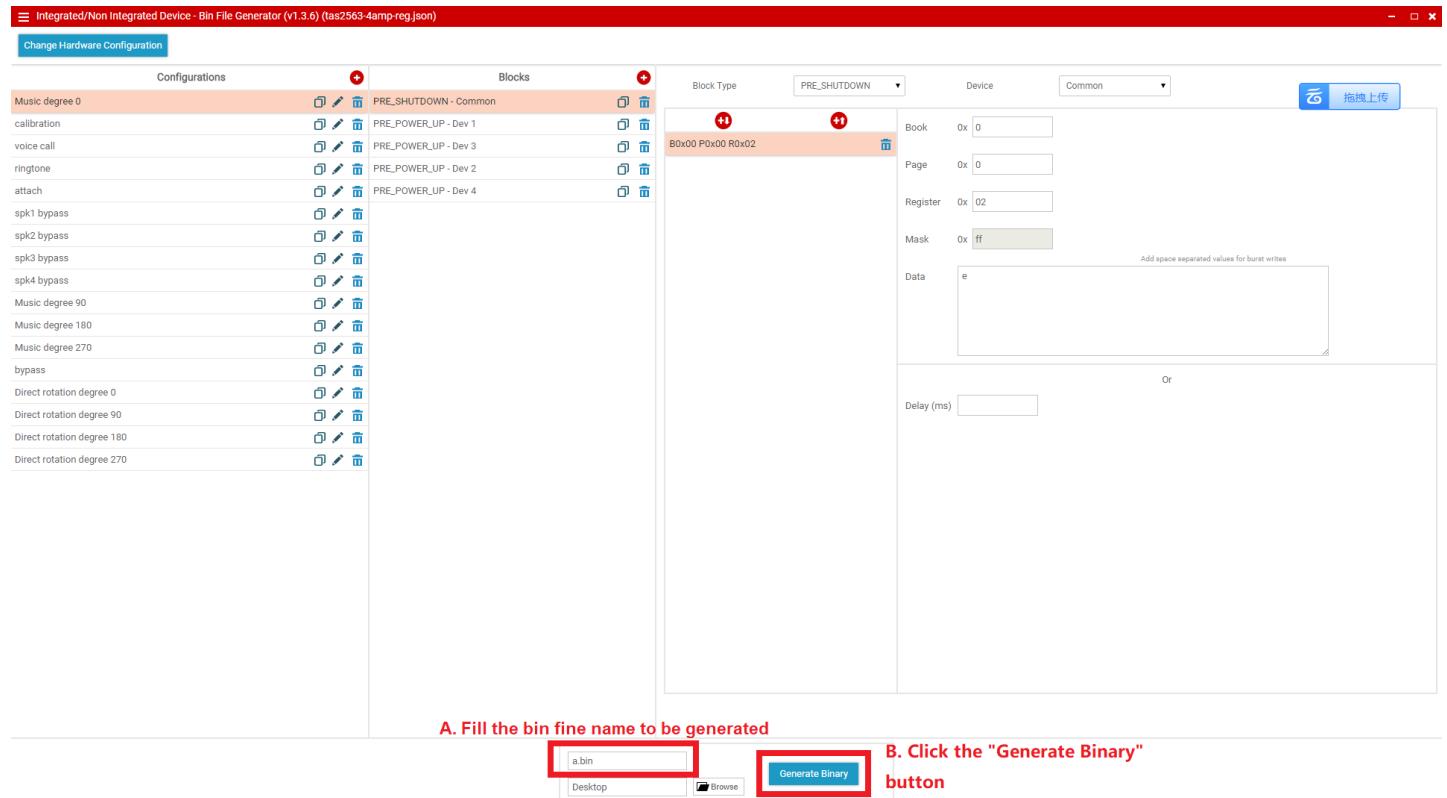
# Appendix VIII | How to generate bin file from json file V

- STEP V: Click the “OK” button, after the folder to save bin file is selected. See the right pic.



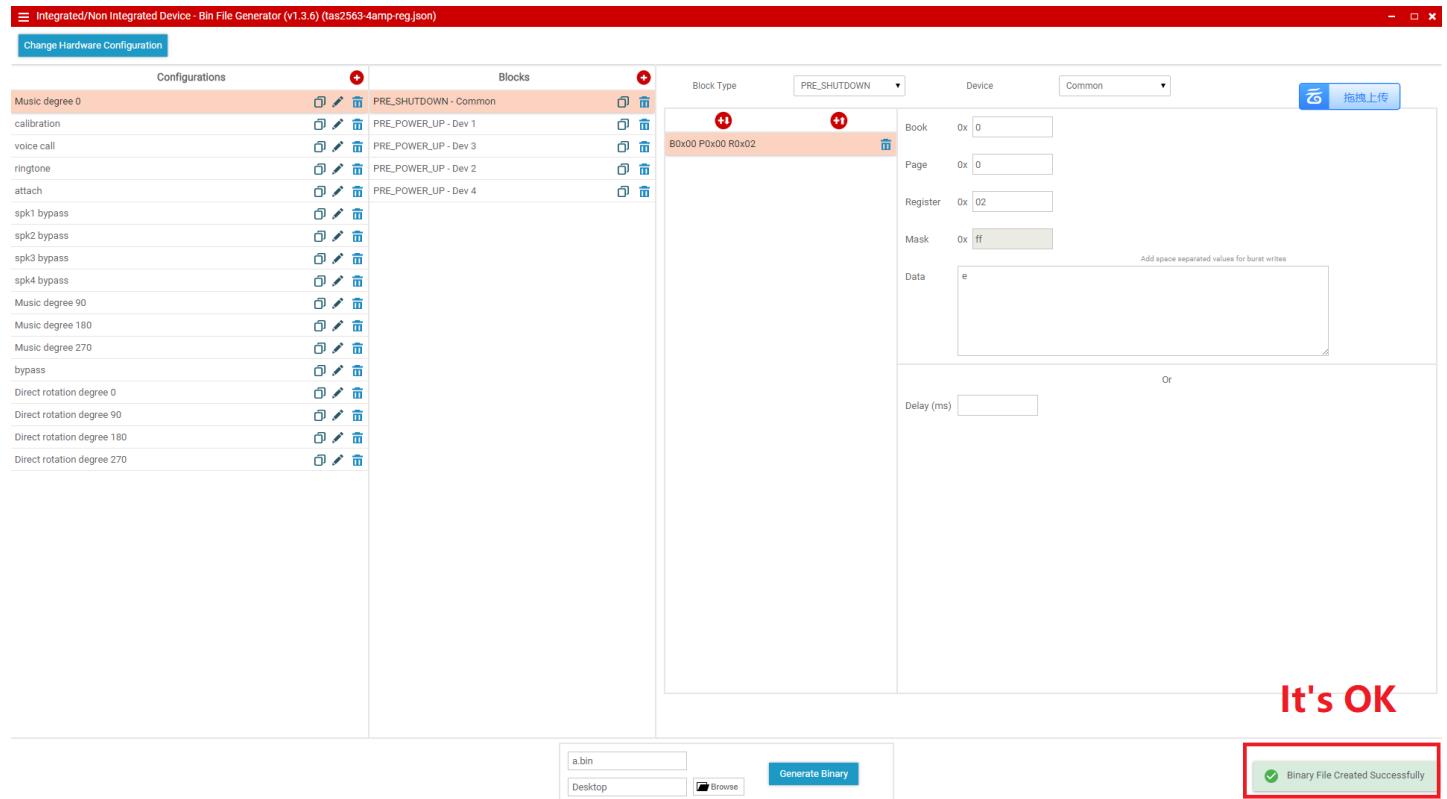
# Appendix IX | How to generate bin file from json file VI

- STEP VI:
  - A. Fill the bin file name to be generated
  - B. Click the "Generate Binary" button.



# Appendix X | How to generate bin file from json file VII

- STEP VII: if pop-up balloon occurs like the right pic, the bin file can be found in the selected folder.



It's OK

# Q&A

*Thank you!*