Application Note

Implementing Mistake-Proof Programming Design for UCD3138x Series Chips Using FUSION-PRODUCTION-GUI

Peter Ban, Harry Ma

ABSTRACT

This document was translated from a simplified Chinese source. (ZHCAG52)

During factory programming of UCD3138x series devices using FUSION-PRODUCTION-GUI, there is a significant risk of mis-programming—i.e., programming a device with firmware that does not match its chip model. To prevent such mis-programming issues during production programming, this document presents a mistake-proof programming design method for UCD3138x series chips implemented through FUSION-PRODUCTION-GUI.

Table of Contents

List of Figures

List of Tables

1 Introduction to FUSION-PRODUCTION-GUI

FUSION-PRODUCTION-GUI is a software tool designed for factory production environments, used to program, configure, and test UCD3138x series digital power controllers during high-volume manufacturing. It allows users to flash firmware, upgrade existing firmware, and integrate UCD programming routines into factory scripts to support automated production.

Figure 1-1 shows the script editing interface of FUSION-PRODUCTION-GUI. The red-highlighted area is the script area, which primarily displays task information being executed in the script; the blue-highlighted area is the configuration area, displaying configuration details for the relevant script tasks; and the green-highlighted area is the task area, showing the task information that can be added to the script area.

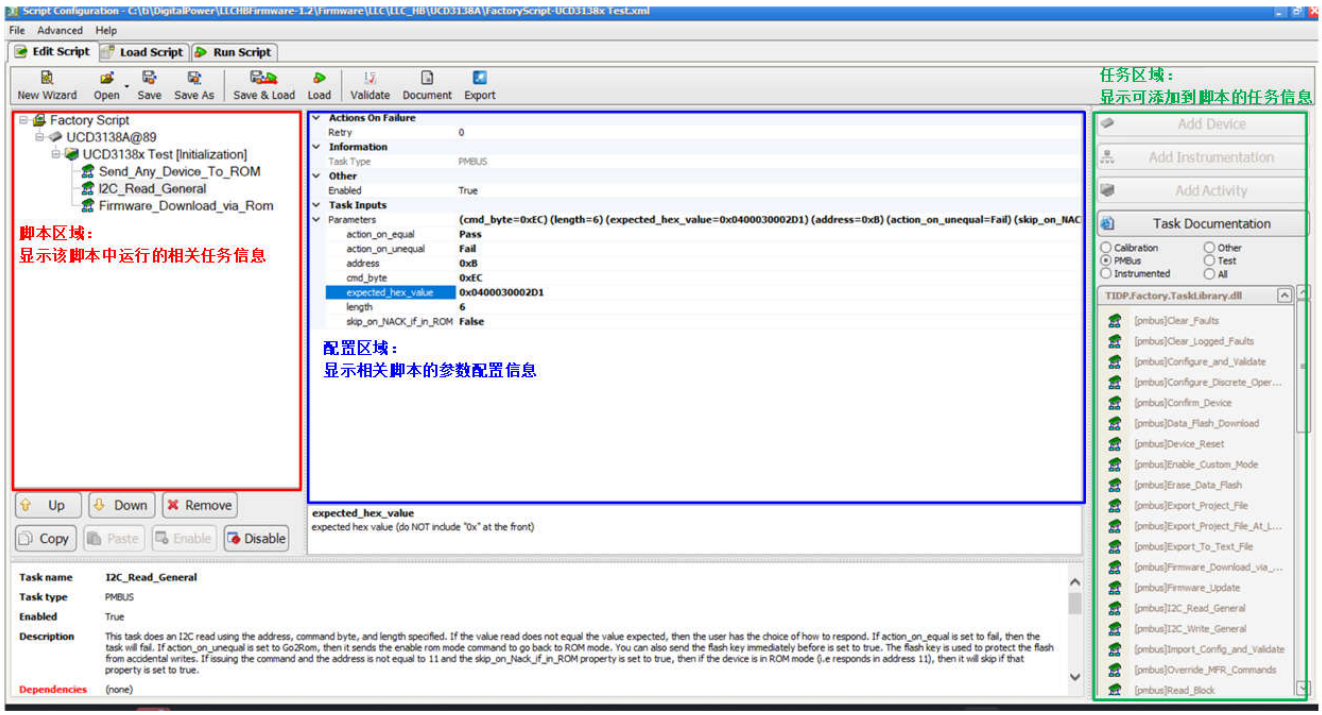


Figure 1-1. FUSION-PRODUCTION-GUI Script Editing Interface

2 Limitations of FUSION-PRODUCTION-GUI for Mistake-Proof Programming

Figure 2-1 illustrates the Boot ROM flow for UCD3138x series chips. The Boot ROM of the UCD3138 performs the following main tasks:

- Initializes the UCD3138x device
- Validates checksums. Two different checksums are used: the Boot Flash checksum for verifying the Boot Flash, and the User Program checksum for verifying the Program Flash. If either checksum matches, the ROM moves the FLASH to address 0 and jumps to execute from Program Flash.

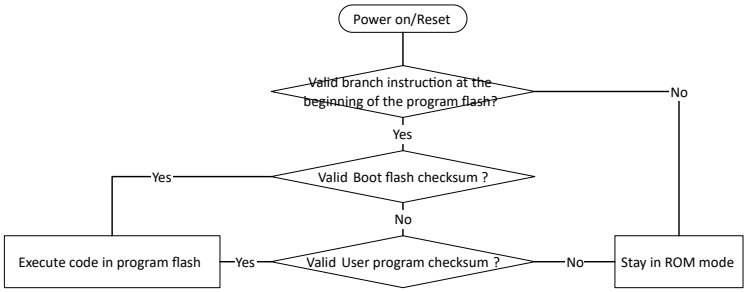


Figure 2-1. UCD3138x Boot ROM Flow

Prior to checksum validation, the UCD3138 Boot ROM first checks whether a 0xEA jump opcode exists at the start of the program flash. If the 0xEA opcode is not present, the Boot ROM does not perform checksum verification and remains in ROM mode.

Although FUSION-PRODUCTION-GUI provides a range of script tasks for programming and operating chips during factory production, the existing tasks cannot directly distinguish between different chips while in ROM mode, thus failing to implement mistake-proof programming design.

3 Implementing Mistake-Proof Programming for UCD3138x

As shown in Figure 3-1, for a UCD3138x chip that has never been programmed, the chip always remains in ROM mode. The Boot ROM takes control of the processor and configures the chip as a PMBus slave, with the device address always set to 11 (0x0B). As listed in Table 3-1, the Boot ROM provides a set of functional commands in ROM mode that can be used to control the UCD3138.

Table 3-1. UCD3138x Boot ROM Supported Commands

Boot ROM Function	Command Byte
Configure Read Address	0xFD
Read 4 Bytes	0xFA
Read 16 Bytes	0xF9
Read Next 16 Bytes	0xF8
Write 4 Bytes	0xF5
Write 16 Bytes	0xF4
Write Next 16 Bytes	0xF3
Mass Erase Flash	0xF2
Page Erase Flash	0xF1
Execute from Program Flash	0xF0
Calculate Checksum	0xEF
Read Checksum	0xEE
Read Version	**0xEC**

Among these, the Read Version (0xEC) command can be used to read the current ROM version through the Boot ROM. Figure 3-1 shows the command definition for the 0xEC instruction.

Start	Device Address & R/W (0x16)	Command Byte (0xF9)	Repeated Start
Device Address & R/W (0x17)	Block Size (0x04)	Version[31:24]	Version[32:16]
Version[15:8]	Version[7:0]	PEC	Stop

Figure 3-1. 0xEC Command Definition

Therefore, we can implement a mistake-proof design by reading the chip's ROM version information using the 0xEC command. The specific workflow is illustrated below:

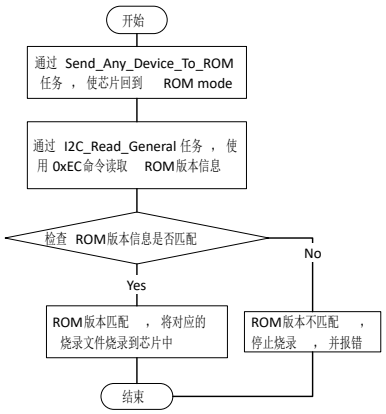


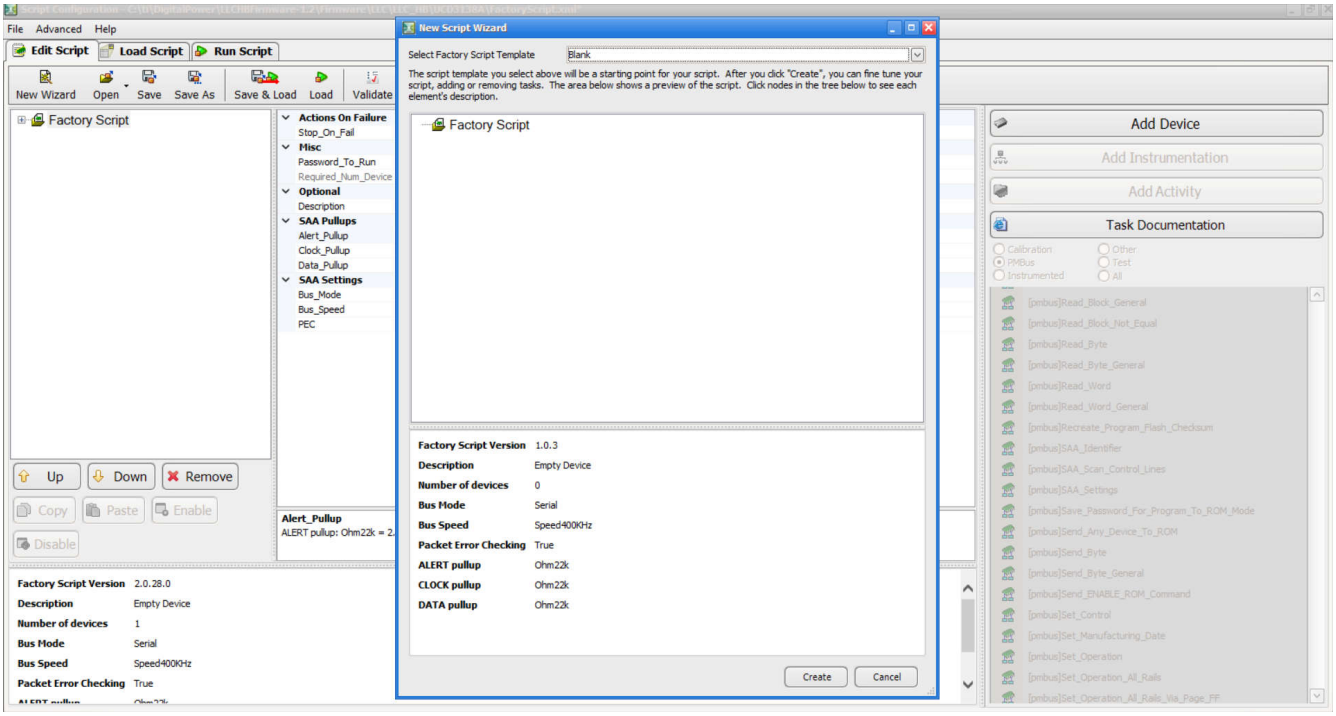
Figure 3-2. PRODUCTION-GUI Mistake-Proof Programming Flow

In actual factory programming scenarios, chips may undergo secondary programming, in which case the chip will be in Flash Mode. Therefore, the Send_Any_Device_To_ROM script task provided in FUSION-PRODUCTION-GUI can be used to force the chip to jump to ROM mode. Subsequently, the I2C_Read_General script task is used to execute the 0xEC command to read the chip's ROM version information. The ROM version is then checked for a match. If the ROM version matches, the subsequent programming process proceeds. Conversely, if the ROM version information does not match, the programming process is halted and an error is reported.

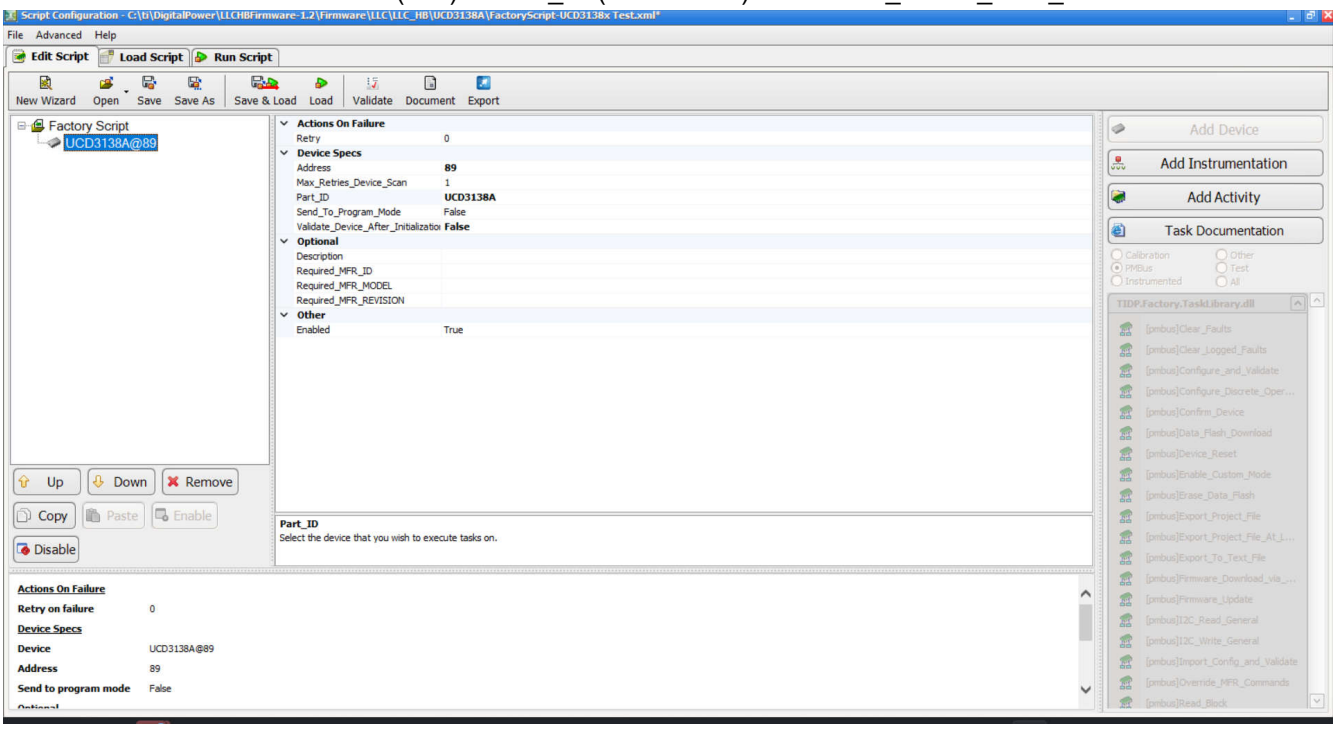
4 Experimental Verification

Taking the UCD3138A as an example, the following describes the script creation process and experimental results for implementing mistake-proof programming design based on FUSION-PRODUCTION-GUI.

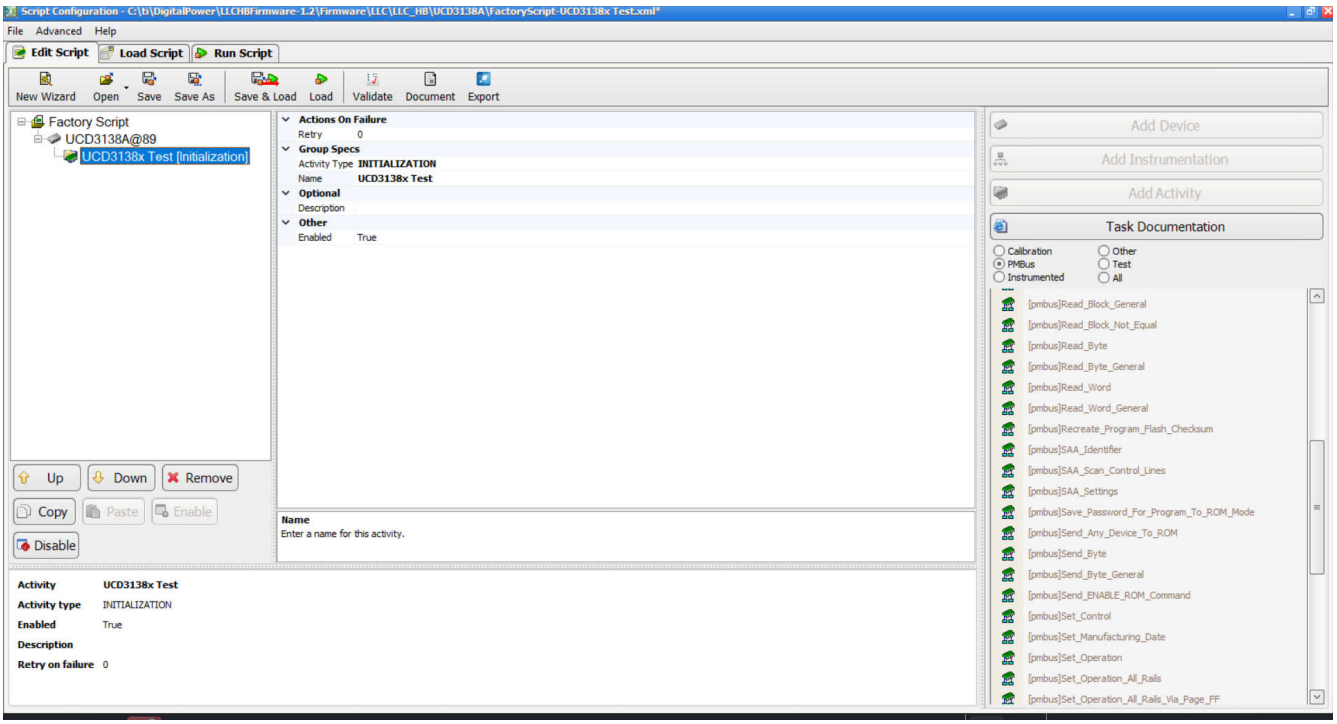
1. Click File → New Script Wizard → Blank → Create



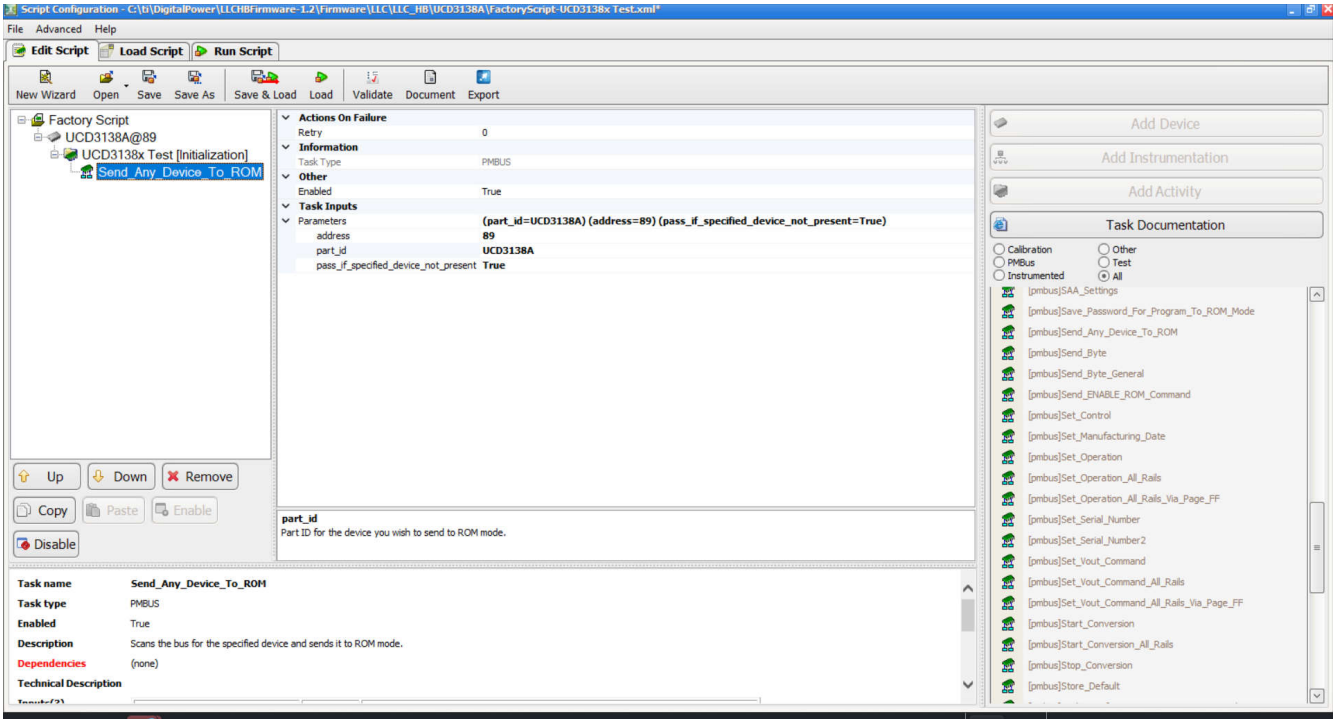
2. Click Add Device → address (89d) → Part_ID (UCD3138A) → Validate_device_After_initialization = "False"



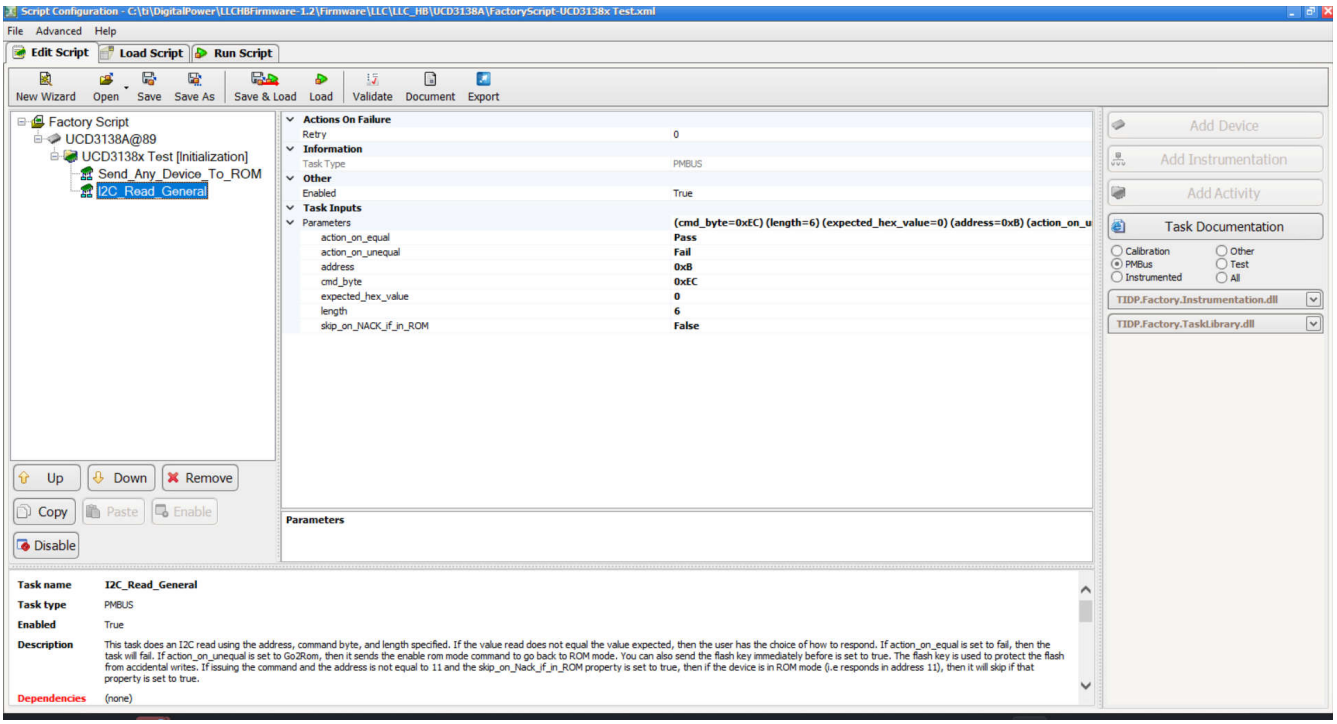
3. Click Add Activity → Activity type "INITIALIZATION" → Name "UCD3138x Test"



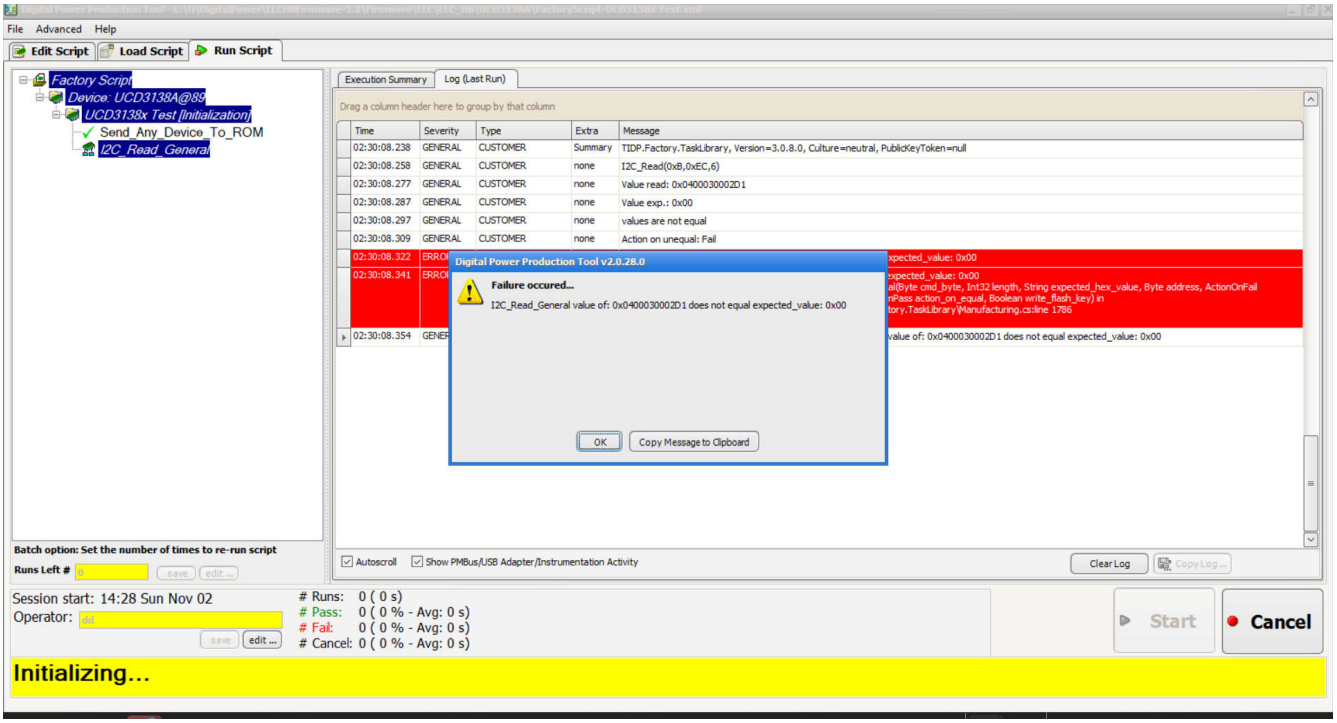
4. In Task Documentation, click "All" → select Send_Any_Device_To_ROM → address "89" → part_id "UCD3138A"



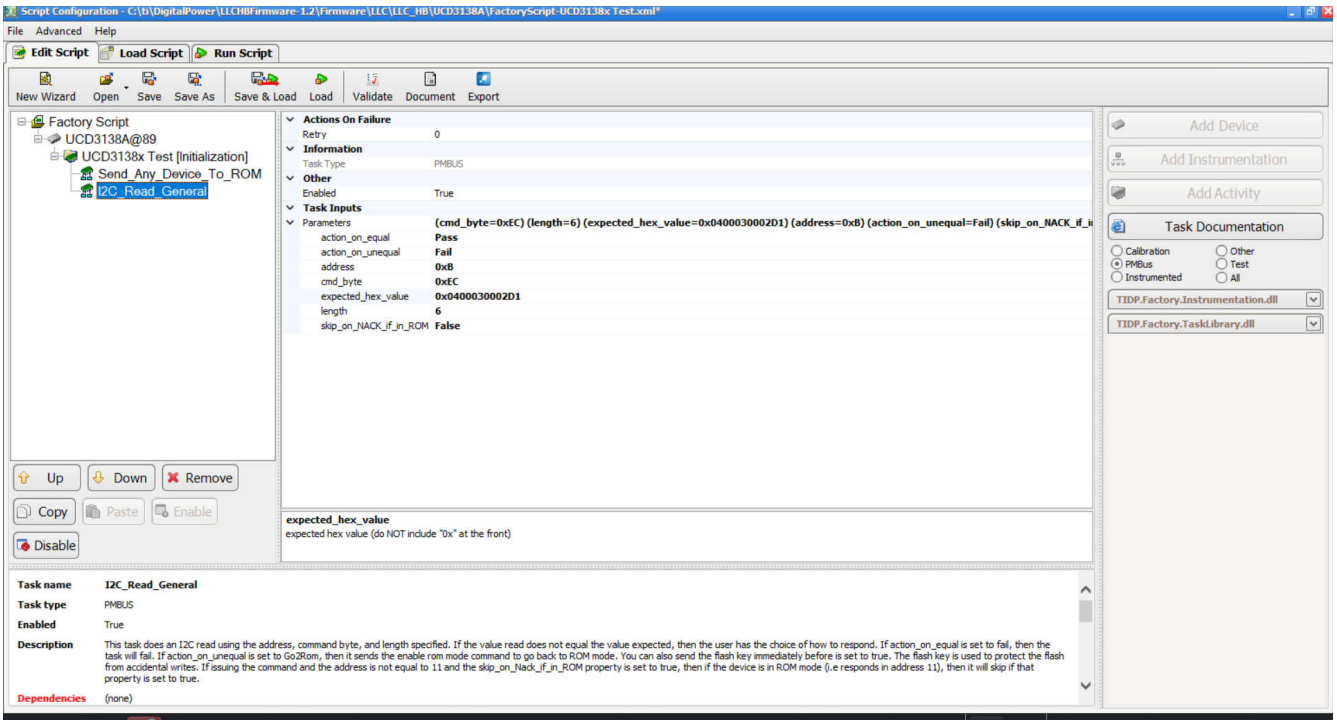
5. In Task Documentation, click I2C_Read_General → address "0xB" → cmd_byte "0xEC" → expected_hex_value "0" → length "6"



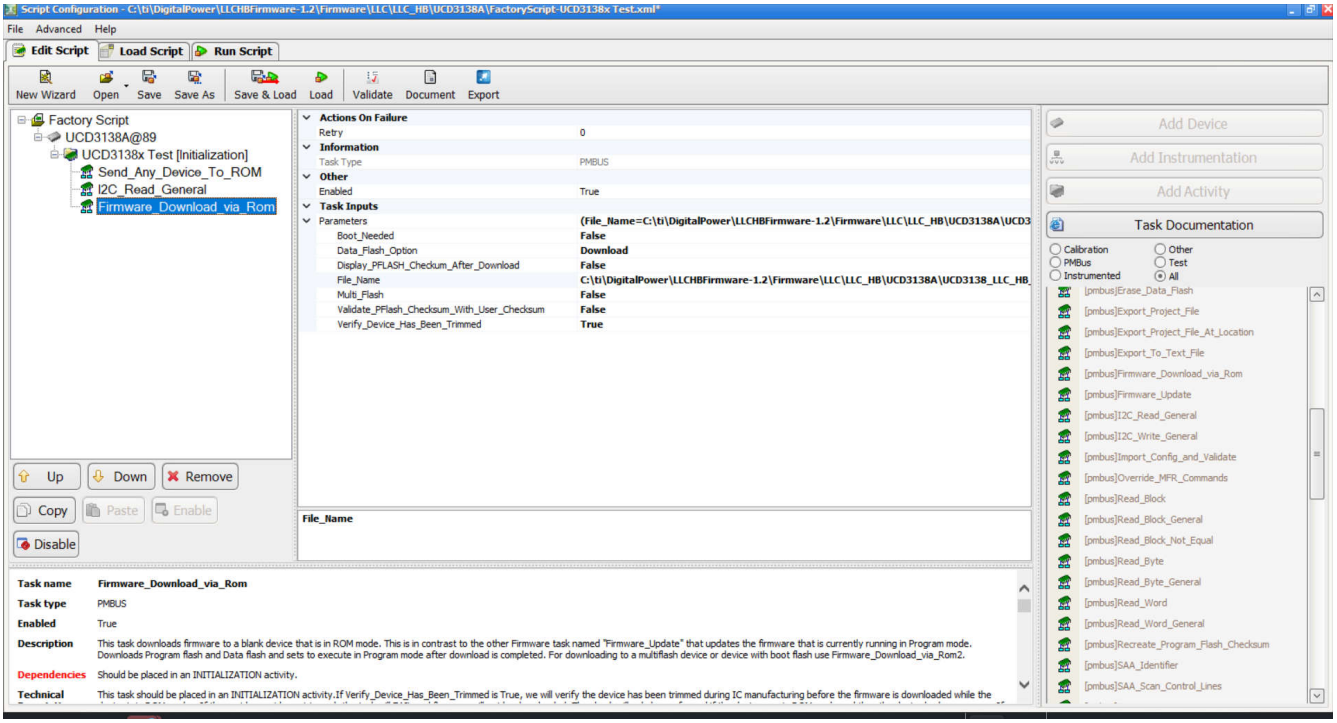
6. Click Save & Load → Start. The GUI displays a ROM version mismatch, and programming fails. The obtained ROM version is 0x0400030002D1



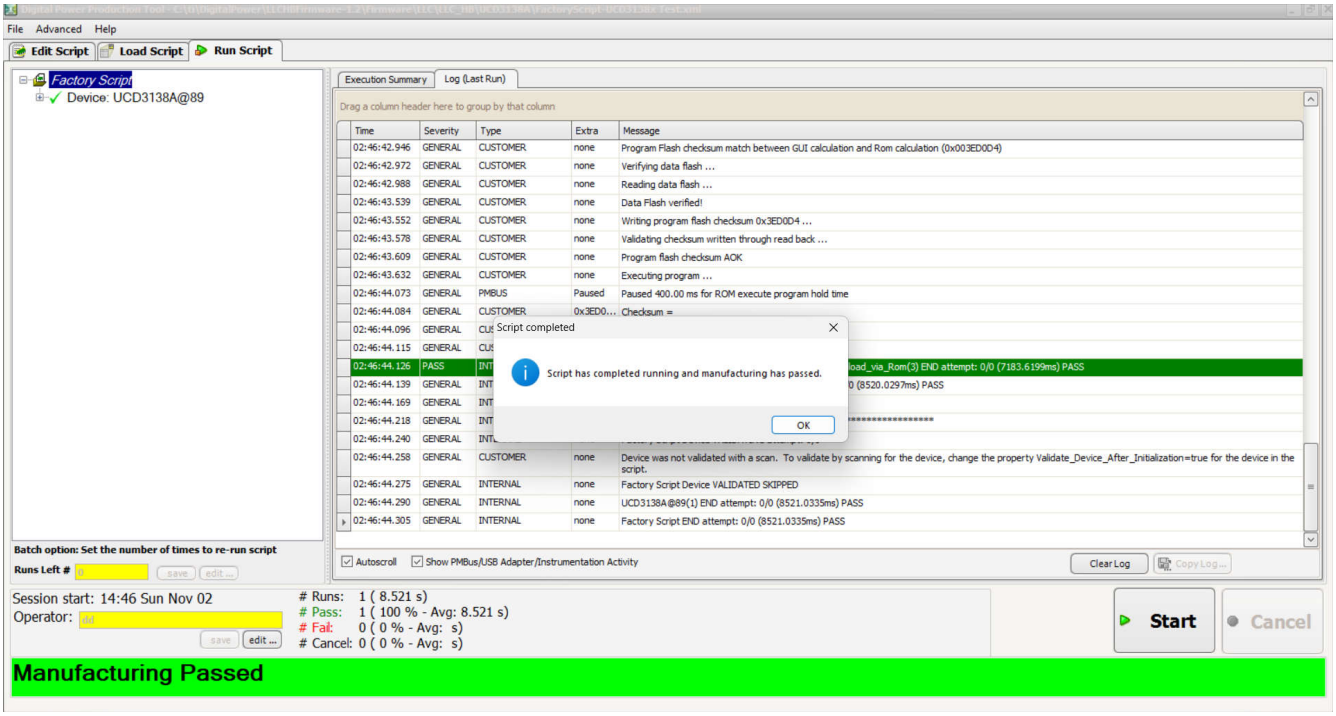
7. Update the ROM version information: expected_hex_value = "0x0400030002D1"



8. In Task Documentation, add the task Firmware_Download_via_Rom → File_name "C:\ti\DigitalPower\LLCHBFirmware-1.2\Firmware\LLC\LLC_HB\UCD3138A\UCD3138_LLC_HB_UCD3138A.x0"



9. Click Save & Load → Start. The ROM version information matches, and programming succeeds.



5 Conclusion

During factory programming of UCD3138 series chips using FUSION-PRODUCTION-GUI, there is a risk of mis-programming—i.e., programming a device with firmware that does not match its chip model—which can lead to anomalies in production programming. To prevent such mis-programming issues during manufacturing, this document presents a mistake-proof programming design method for UCD3138 series devices, implemented by verifying the ROM version through FUSION-PRODUCTION-GUI.

6 Reference documentation

1. UCD31xx Digital Power Supply Controller - Technical Reference Manual.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025