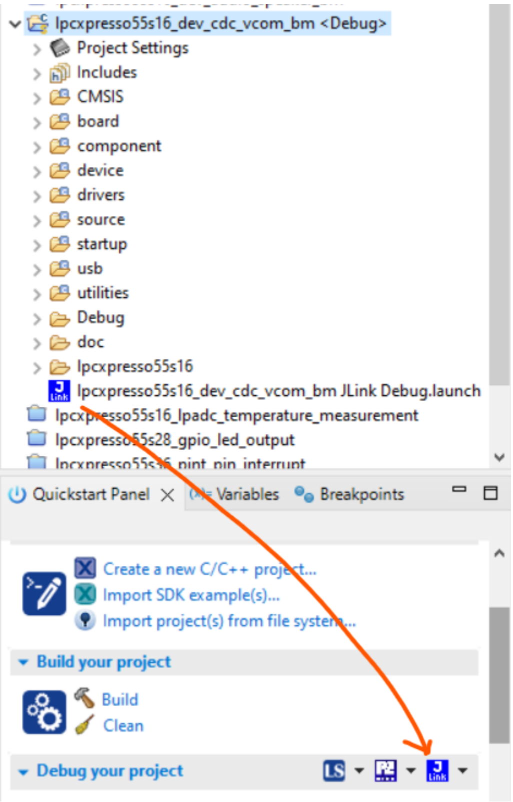
LPCScrypt & J-Link for LPC

This document was made to explain how to regain control of any LPC EVK on brick mode without using an external debugger.

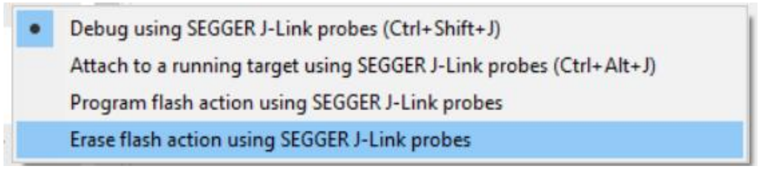
Explore the simple way

In some cases, this method forms well and is the easy way, open your IDE and select your project.


lpcxpresso55s16_dev_cdc_vcom_bm <Debug>
Project Settings
Includes
CMSIS
board
component
device
drivers
source
startup
usb
utilities
Debug
doc
lpcxpresso55s16
lpcxpresso55s16_dev_cdc_vcom_bm JLink Debug.launch
lpcxpresso55s16_lpadc_temperature_measurement
lpcxpresso55s28_gpio_led_output
lpcxpresso55s16_pint_pin_interrupt
Quickstart Panel
Variables
Breakpoints
Create a new C/C++ project...
Import SDK example(s)...
Import project(s) from file system...
Build your project
Build
Clean
Debug your project

At this point you may have a debugger configuration of your last debug session, so, you have similar icons to the image.

Open the tab below and select erase.


Debug using SEGGER J-Link probes (Ctrl+Shift+J)
Attach to a running target using SEGGER J-Link probes (Ctrl+Alt+J)
Program flash action using SEGGER J-Link probes
Erase flash action using SEGGER J-Link probes

Depending on your debugger configuration you need to select the same icons below.

Note: In some cases will be necessary to put your MCU in ISP.

And that's all.

LPCScrypt & J-Link for LPC

Introduction

First, you need to install these tools on your PC.

LPCScrypt [LPCScrypt v2.1.2 | NXP Semiconductors].
J-Link Commander. [J-Link Commander (segger.com)].

To understand this document we need to know that every EVK [Evaluation Kit] can be divided into 2 parts. The debugger on the board Link2 Figure 1 and the target LPC55sXX Figure1.

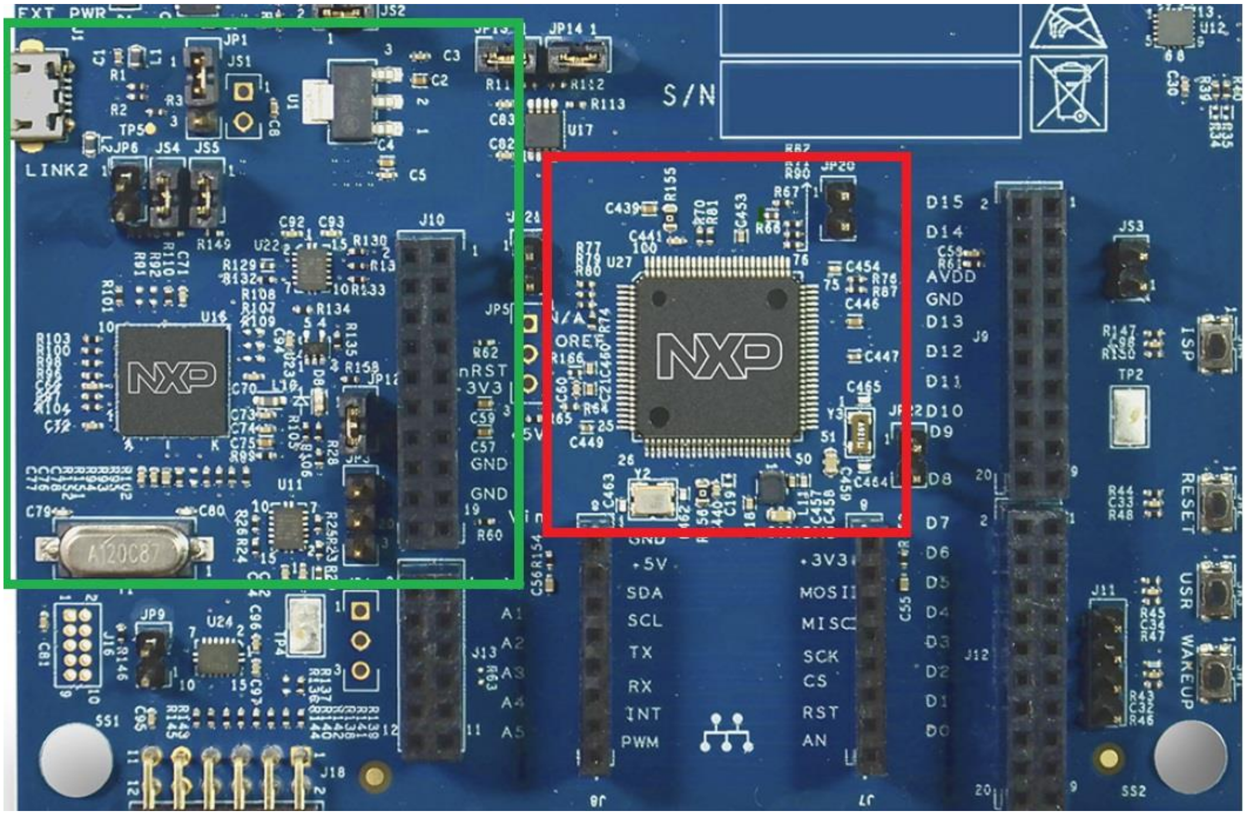


Figure 1. Link2 Red square, LPC55sXX green square

LPCScrypt & J-Link for LPC

Link2

The Link 2 (LPC4370) debugger on the board probe can be configured to support various development tools and IDEs using a variety of different downloadable firmware images.

Available firmware images include:

- J-Link by Segger.
- P&E Micro.
- LinkServer.

By default, the EVK board has the firmware LinkServer on the LPC4370 or LPC55s69 (dependent on your board), for this proposal we will see how to change to J-Link On-Board.

DFU

The EVK needs to be prepared to receive this firmware on the debugger, to do that we need to put the board on **DFU** [Device Firmware Update].

On the schematic need to find the DFU jumper to put on the board, the image below is an example of different EVKs check Figure 2.

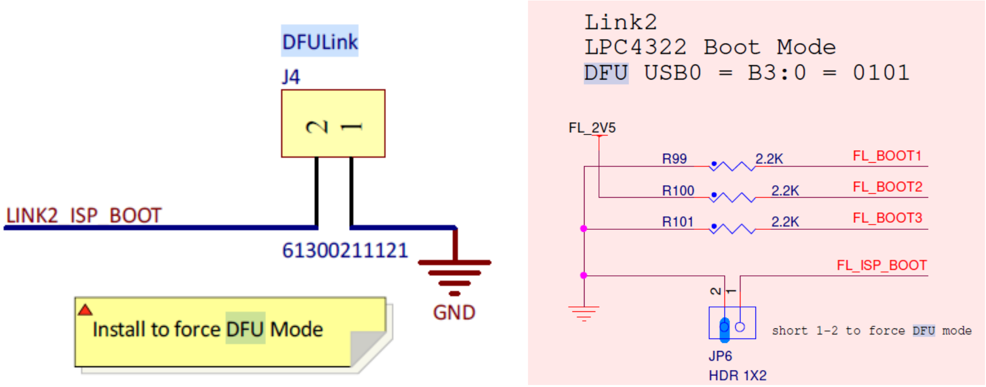


Figure 2. DFU from different boards.

Then connect it to the PC.

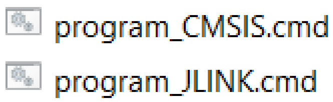
Note: The firmware update is completely reversible.

LPCScrypt & J-Link for LPC

LPCScrypt

Once installed on the PC we need to locate these files of the installation.

Root example: **C:\nxp\LPCScrypt\scripts**

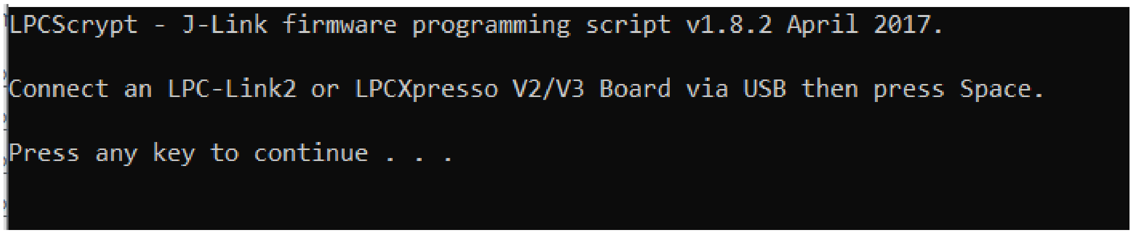


program_CMSIS.cmd

program_JLINK.cmd

When you have the board on DFU, connecting the USB Llink2 to the PC and then RUN the Scrypt (**program_JLINK**) in CMD check Figure 3.

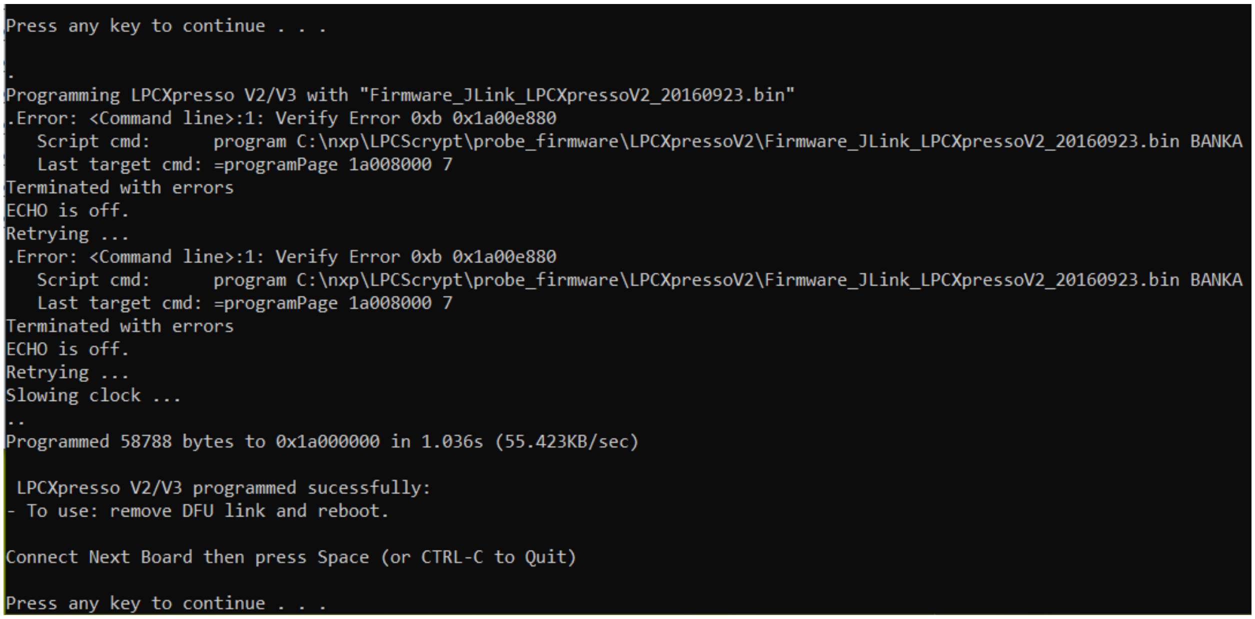
Press any key to continue…



```
LPCScrypt - J-Link firmware programming script v1.8.2 April 2017.

Connect an LPC-Link2 or LPCXpresso V2/V3 Board via USB then press Space.

Press any key to continue . . .
```

Figure 3. Flash firmware of J-Link in DFU mode.

Successfully done, at this moment the EVK has the firmware of J-Link Segger. Review the Figure 4.



```
Press any key to continue . . .

.
Programming LPCXpresso V2/V3 with "Firmware_JLink_LPCXpressoV2_20160923.bin"
.Error: <Command line>:1: Verify Error 0xb 0x1a00e880
   Script cmd:      program C:\nxp\LPCScrypt\probe_firmware\LPCXpressoV2\Firmware_JLink_LPCXpressoV2_20160923.bin BANKA
   Last target cmd: =programPage 1a008000 7
Terminated with errors
ECHO is off.
Retrying ...
.Error: <Command line>:1: Verify Error 0xb 0x1a00e880
   Script cmd:      program C:\nxp\LPCScrypt\probe_firmware\LPCXpressoV2\Firmware_JLink_LPCXpressoV2_20160923.bin BANKA
   Last target cmd: =programPage 1a008000 7
Terminated with errors
ECHO is off.
Retrying ...
Slowing clock ...
..
Programmed 58788 bytes to 0x1a000000 in 1.036s (55.423KB/sec)

 LPCXpresso V2/V3 programmed sucessfully:
- To use: remove DFU link and reboot.

Connect Next Board then press Space (or CTRL-C to Quit)

Press any key to continue . . .
```

Figure 4. The firmware was flashed successfully.

Remove the USB cable to remove the DFU jumper with the board unpowered.

LPCScrypt & J-Link for LPC

ISP

Brief of ISP (In System Programming) this method is used to recover a part programmed with a corrupted image which is not detectable by ROM.

So, to enter this mode the user needs to put a jumper in the ISP header pins, search for this in the schematic, then connect to the PC and check Figure 5.

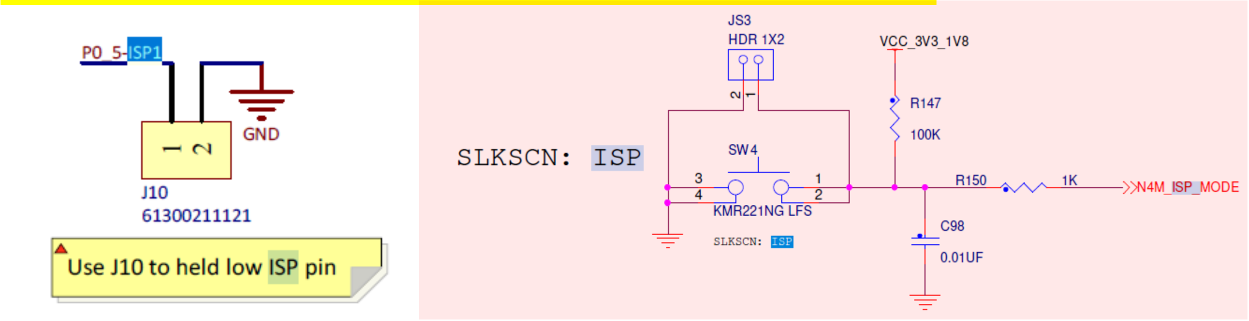
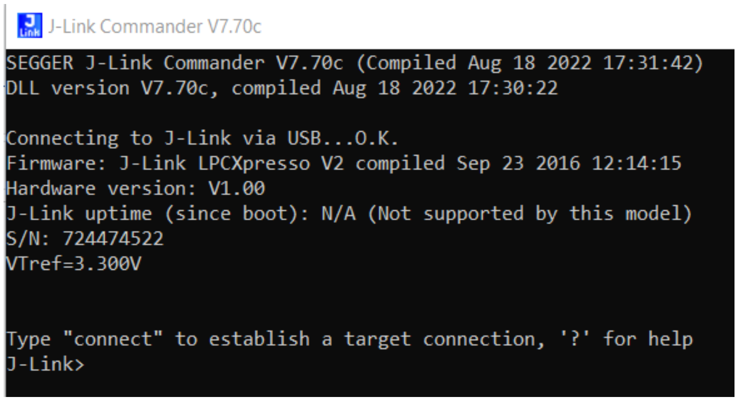


Figure 5. ISP image examples from different boards.

LPCScrypt & J-Link for LPC

J-LINK Commander

Open the software and if the communication is right the message will **appear J-Link via USB… OK** check Figure 6.



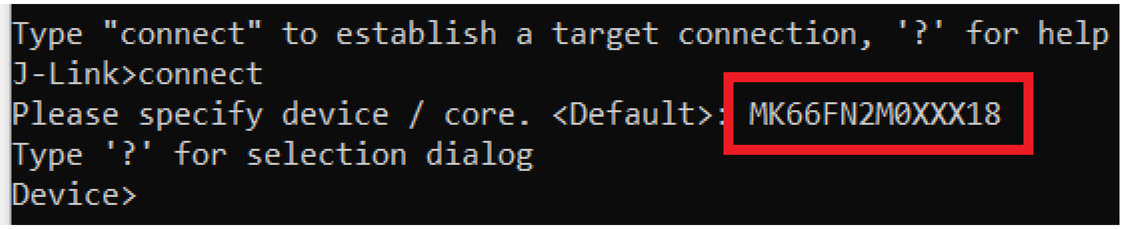
```
SEGGER J-Link Commander V7.70c (Compiled Aug 18 2022 17:31:42)
DLL version V7.70c, compiled Aug 18 2022 17:30:22

Connecting to J-Link via USB...O.K.
Firmware: J-Link LPCXpresso V2 compiled Sep 23 2016 12:14:15
Hardware version: V1.00
J-Link uptime (since boot): N/A (Not supported by this model)
S/N: 724474522
VTref=3.300V


Type "connect" to establish a target connection, '?' for help
J-Link>
```

Figure 6. The EVK is now communicating to the tool J-Link Commander.

Commands

To start the communication needs to send the command “**connect”**.

Then the tool shows your last target (if you use it), and use the next command “**?”** to change the target review Figure 7.

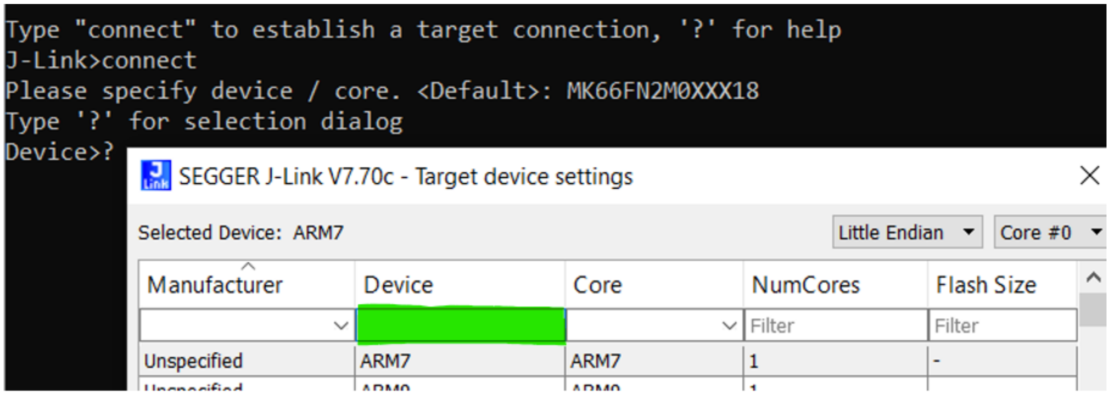


```
Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: MK66FN2M0XXX18
Type '?' for selection dialog
Device>
```

Figure 7. Review the target.

LPCScrypt & J-Link for LPC

In the new popup window in the green area, you need to put the name of the target, take care you do not put the debugger check Figure 8.



```
Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: MK66FN2M0XXX18
Type '?' for selection dialog
Device>?
```

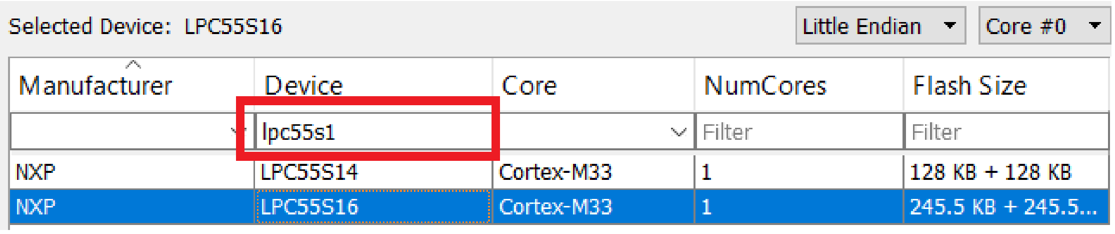
SEGGER J-Link V7.70c - Target device settings

Selected Device: ARM7

Manufacturer	Device	Core	NumCores	Flash Size
	🟩		Filter	Filter
Unspecified	ARM7	ARM7	1	-

Figure 8. In the green area, you need to put the matricula of the MCU target.

Example [LPC55s16 or LPC55s69] depending of your target check Figure 9.



Selected Device: LPC55S16

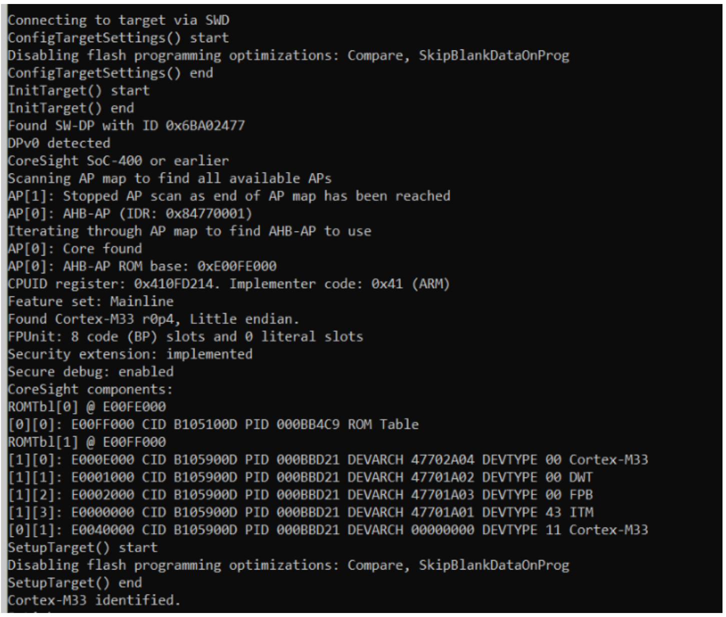
Manufacturer	Device	Core	NumCores	Flash Size
	lpc55s1		Filter	Filter
NXP	LPC55S14	Cortex-M33	1	128 KB + 128 KB
NXP	LPC55S16	Cortex-M33	1	245.5 KB + 245.5...

Figure 9. If the tool supports the MCU will show below the red square.

LPCScrypt & J-Link for LPC

In the next selections the tool is asking for the interface of communication, the communication of the EVK is SWD, and for that write “**S”** as SWD.

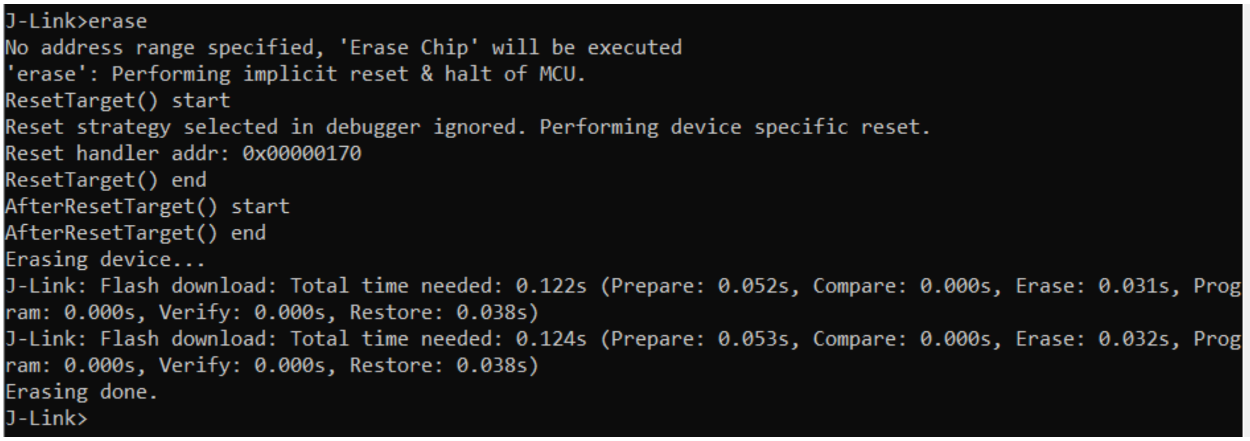
On speed, only click enter without entry. Then this will appear before the connection check Figure 10.



```
Connecting to target via SWD
ConfigTargetSettings() start
Disabling flash programming optimizations: Compare, SkipBlankDataOnProg
ConfigTargetSettings() end
InitTarget() start
InitTarget() end
Found SW-DP with ID 0x6BA02477
DPv0 detected
CoreSight SoC-400 or earlier
Scanning AP map to find all available APs
AP[1]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x84770001)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FE000
CPUID register: 0x410FD214. Implementer code: 0x41 (ARM)
Feature set: Mainline
Found Cortex-M33 r0p4, Little endian.
FPUnit: 8 code (BP) slots and 0 literal slots
Security extension: implemented
Secure debug: enabled
CoreSight components:
ROMTbl[0] @ E00FE000
[0][0]: E00FF000 CID B105100D PID 000BB4C9 ROM Table
ROMTbl[1] @ E00FF000
[1][0]: E000E000 CID B105900D PID 000BBD21 DEVARCH 47702A04 DEVTYPE 00 Cortex-M33
[1][1]: E0001000 CID B105900D PID 000BBD21 DEVARCH 47701A02 DEVTYPE 00 DWT
[1][2]: E0002000 CID B105900D PID 000BBD21 DEVARCH 47701A03 DEVTYPE 00 FPB
[1][3]: E0000000 CID B105900D PID 000BBD21 DEVARCH 47701A01 DEVTYPE 43 ITM
[0][1]: E0040000 CID B105900D PID 000BBD21 DEVARCH 00000000 DEVTYPE 11 Cortex-M33
SetupTarget() start
Disabling flash programming optimizations: Compare, SkipBlankDataOnProg
SetupTarget() end
Cortex-M33 identified.
```

Figure 10. The communication is Done.

Then use the command “**erase”** like the Figure 11.



```
J-Link>erase
No address range specified, 'Erase Chip' will be executed
'erase': Performing implicit reset & halt of MCU.
ResetTarget() start
Reset strategy selected in debugger ignored. Performing device specific reset.
Reset handler addr: 0x00000170
ResetTarget() end
AfterResetTarget() start
AfterResetTarget() end
Erasing device...
J-Link: Flash download: Total time needed: 0.122s (Prepare: 0.052s, Compare: 0.000s, Erase: 0.031s, Prog
ram: 0.000s, Verify: 0.000s, Restore: 0.038s)
J-Link: Flash download: Total time needed: 0.124s (Prepare: 0.053s, Compare: 0.000s, Erase: 0.032s, Prog
ram: 0.000s, Verify: 0.000s, Restore: 0.038s)
Erasing done.
J-Link>
```

Figure 11. The tool indicate the erase is done.

Finally disconnect the USB and remove the ISP jumper, then open the IDE and test the blinky led example, or if you wanted you could use the BLHOST.

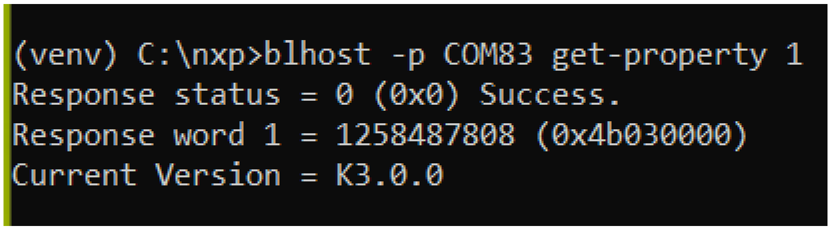
LPCScrypt & J-Link for LPC

Flashing the MCU using BLHOST.

At this step you could able to use the IDE or use BLHOST.

You could install SPSDK if you wanted [Installation Guide — SPSDK documentation].
In ISP mode…

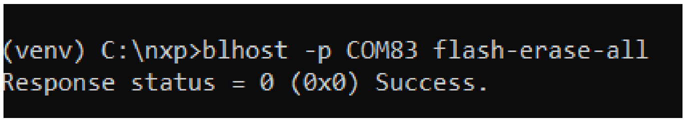
blhost -p COMxx get-property 1



```
(venv) C:\nxp>blhost -p COM83 get-property 1
Response status = 0 (0x0) Success.
Response word 1 = 1258487808 (0x4b030000)
Current Version = K3.0.0
```

Figure 12. The first command is to check the communication.

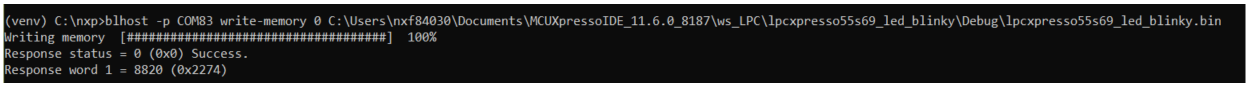
blhost -p COMxx flash-erase-all



```
(venv) C:\nxp>blhost -p COM83 flash-erase-all
Response status = 0 (0x0) Success.
```

Figure 13. Do a mass erase.

blhost -p COMxx write-memory 0 C:\root_example\Debug\led_blinky.bin



```
(venv) C:\nxp>blhost -p COM83 write-memory 0 C:\Users\nxf84030\Documents\MCUXpressoIDE_11.6.0_8187\ws_LPC\lpcxpresso55s69_led_blinky\Debug\lpcxpresso55s69_led_blinky.bin
Writing memory  [####################################]  100%
Response status = 0 (0x0) Success.
Response word 1 = 8820 (0x2274)
```

Figure 14. The flashing was done.

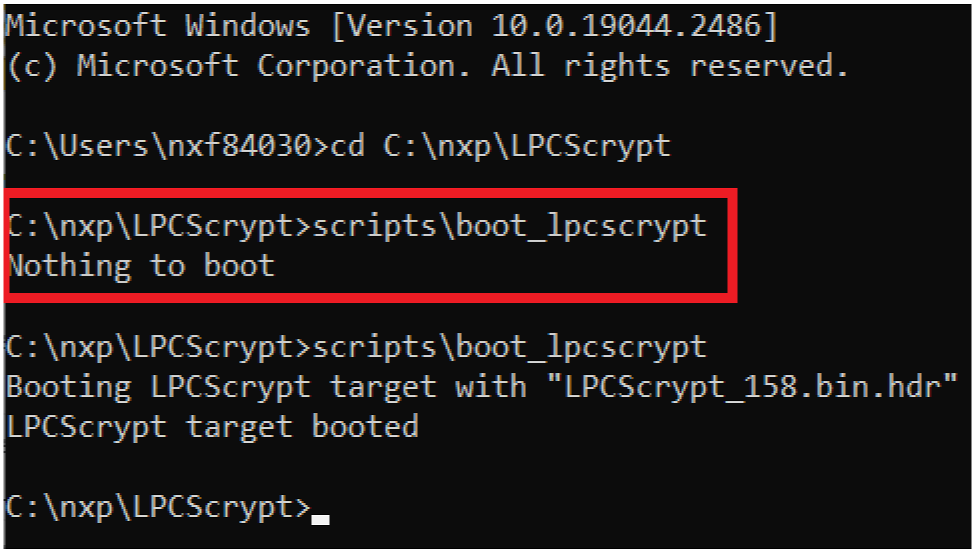
Remove the ISP jumper and reset your device.

LPCScrypt & J-Link for LPC

Common error

LPCScrypt

If the message appears "**Nothing to boot**", need to be sure the board is connected to the PC with the DFU mode in LINK 2 chek Figure 16.

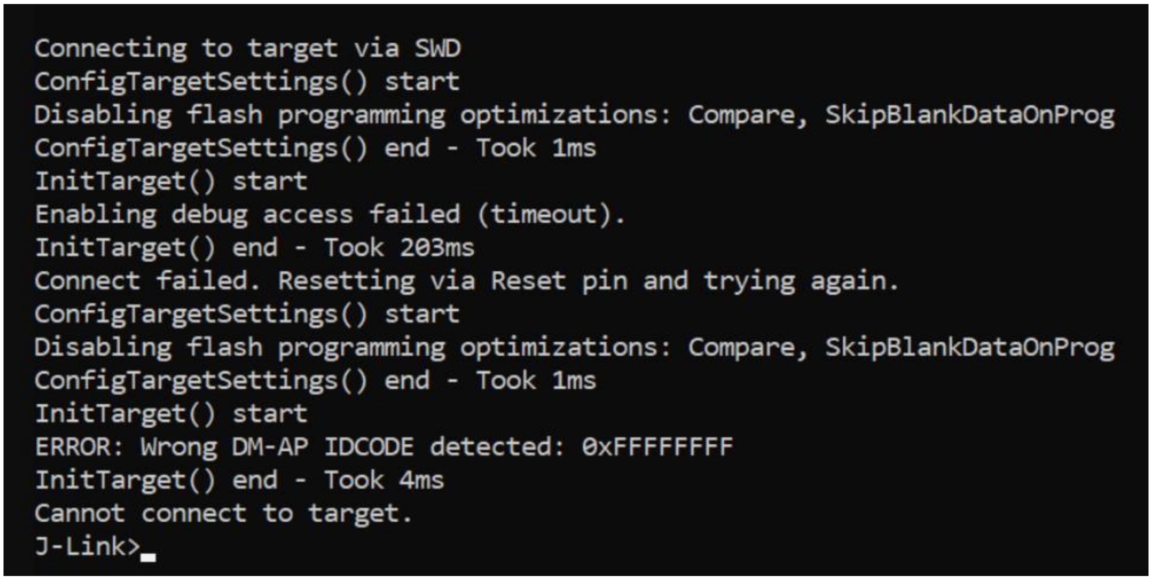


```
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nxf84030>cd C:\nxp\LPCScrypt

C:\nxp\LPCScrypt>scripts\boot_lpcscrypt
Nothing to boot

C:\nxp\LPCScrypt>scripts\boot_lpcscrypt
Booting LPCScrypt target with "LPCScrypt_158.bin.hdr"
LPCScrypt target booted

C:\nxp\LPCScrypt>_
```

Figure 16. The red square is an example of an error communication.

J-Link Commander

If a similar message appears, "**Cannot connect to target**", need to put the target in **ISP** mode, and return to the first steps with J-Link commander check Figure 18.



```
Connecting to target via SWD
ConfigTargetSettings() start
Disabling flash programming optimizations: Compare, SkipBlankDataOnProg
ConfigTargetSettings() end - Took 1ms
InitTarget() start
Enabling debug access failed (timeout).
InitTarget() end - Took 203ms
Connect failed. Resetting via Reset pin and trying again.
ConfigTargetSettings() start
Disabling flash programming optimizations: Compare, SkipBlankDataOnProg
ConfigTargetSettings() end - Took 1ms
InitTarget() start
ERROR: Wrong DM-AP IDCODE detected: 0xFFFFFFFF
InitTarget() end - Took 4ms
Cannot connect to target.
J-Link>_
```

Figure 18. Show how the communication is not done.