



Technical Note

NXP S32G: Cold Start with Cortex-M7 and A53 core

May 2023



This Technical Note describes how to start debugging the Cortex-M7 or Cortex-A53 cores on S32R45x and S32G2/3 MCU via winIDEA initialization files if no valid boot image is found.

Initialization files are attached to this document.

On NXP microcontroller only Hardware Security Engine (HSE) starts to run after reset. It reads boot image either from QSPI flash or SD card and starts either the Cortex-M7 or the Cortex-A53 core. If no valid image is found, no core is started. All cores remain in reset and the device enters Serial Boot mode allowing the code to be loaded to RAM via CAN or UART interface.

BlueBox debugger can override Serial Boot mode and through a Cold Start procedure gains access to RAM too. Using winIDEA initialization files, a small program with endless loop is written to RAM and then the M7 / A-53 core is released from reset. Initialization files use AXI access type since standard Virtual Memory area access is not functioning due to the M7 / A-53 core being in reset.

Tool requirements:

- ✓ winIDEA 9.17.158 or newer
- ✓ BlueBox iC5700 or iC5000
- ✓ ARM HSSTP II Active Probe or 10-pin CoreSight Debug Adapter or 20-pin CoreSight Debug Adapter

winIDEA Configuration

Run and debug core with HSE

If the boot image is available HSE initializes only the part of RAM where the code image is loaded. If you want to load code to another part of RAM, the RAM Initialization script must be specified, so that the debugger can load the code anywhere in RAM.

1. Create a new workspace and select your device.
2. Open *Hardware / CPU Options / Reset* and select **Stop and preset** as RESET method.
3. Define *Post RESET delay*. HSE must have enough time to read the boot image and start the application core.
4. Make sure *Initialization before Programming / Connect* is **Default**.
5. Specify the RAM initialization script under *Initialization before Programming / Initialize*.
6. Perform:
 - Reset - HSE boots the application from SD Card / QSPI Flash, and no other part of RAM will be initialized.
 - Download - Resets the target, initialize the RAM (because of the specified script) and HSE boots the application from SD Card / QSPI flash, the entire RAM will be initialized (so if we specified any program files they can and will be downloaded at this point)

After reset, HSE core is the only core which is running. It reads the boot image, copies the code image to RAM and starts the core responsible for running the application.

Perform Cold start (run a core with winIDEA)

If the boot image is not available, or is invalid, HSE will not start any of the application cores. All the application cores remain in reset. To be able to debug such a target, winIDEA can start one of the application cores. That is done by the *Cold Start* script, which writes an endless loop to RAM and starts the specified core. The RAM location where this loop is written can be specified as a script parameter. The RAM initialization script in this case is not necessary, because RAM is first initialized by the *Cold Start* script.

1. Create a new workspace and select your device.
2. Open *Hardware / CPU Options / Reset* and make sure:
 - Boot core is the core you want to cold start from.
 - Latch target RESET in the RESET pin section in unchecked
 - Stop and Preset in the RESET method drop-down menu is selected
3. Make sure the *Cold Start* script is added to *Initialization before programming / Connect*.
4. Specify the core that you want to start, and the start address where the code is loaded as the script parameters via the arrow button.

CPU Options S32R455_M7
✕

Reset

Debugging

Hardware

JTAG

Analyzer

Aurora

SoC

CTM/CTI

⊕ Cores

Require correct device ID when attaching to SoC

Boot core M7_0

RESET pin

RESET pin connected

Latch target RESET Stop after target RESET

RESET method Stop and preset

RESET duration 200 ms

Post RESET delay 2000 ms

Target RESET timeout 100 ms

Drive RESET while Detached no

RESET drive mode Open Drain

RAM Initialization

On-chip Never

Some devices require RAM ECC initialization before the application can use it

- Never: no RAM init is performed. RAM will behave the same as when the device is run without a debugger.
- Always: all on-chip RAM is initialized. Use this if your application doesn't contain RAM init code yet.
- Auto: all on-chip RAM is initialized before download. Before FLASH operation, only the minimum amount of RAM needed for FLASH operation is initialized.

At Initialization

Disable watchdog Suspend peripherals while stopped ...

Detect watchdog reset for 20 ms

Initialization before Programming

Connect \$(SFR_FILE_DIR)\S32R45x_ColdStart.cpp

Initialize \$(SFR_FILE_DIR)/S32R45x_RAM_Init.INI

Reset CPU after Download (Invalidates initialization)
Initialization before Debug session will be performed.