# Trace32 In Target Reset Awaiting Agent

- R Radoslav
- **Forum name:** #Debugging

Hello,
I'm using Trace32 PowerView for ARM
Software Version: N.2025.06.000181255
Build: 181255.
with NXP S32K344 chip.

I'm observing strange behavior when I do reset via Trace 32:
1)CPU->System Settings->RESetOut: after reset , F_EXR external reset flag as expected.
2) CPU-> In Target Reset: F_EXR flag + FCCU_FTR reset flag. Here I don't understand why FCCU_FTR occurred.

Could you explain what is difference between RESetOut and In Target Reset?
I suspect that In Target Reset does except toggling nSRST signal also something more - does it send some command to DAP or something like that?

nSRST signal is connected to the dedicated PA5 RESET_B pin on MCU side.
As per Osci, both RESetOut and In Target Reset set down reset_b for 100ms.
FCCU_FTR flag happens after In Target Reset only if FCCU NCF5 is enabled - Debug Activation Catch (FCCU is NXP Vendor IP catching some HW faulty signals from the chip) => that's why I suspect In Target Reset does something more than toggling nSRST.

Kind Regards,
Radoslav

## Comment (1)

**Ahmed Regaieg**
22 hours ago

The “In Target Reset” button is mapped to the command SYStem.RESetTarget.

The command SYStem.RESetTarget have similarities with SYStem.ResetOut, as they perform a target reset.

But SYStem.RESetTarget is closer to a SYStem.Up. Both commands reset the target, set the core into debug mode and stop the core. SYStem.RESetTarget additionally performs a Register.Init afterwards.

For context: due to some controllers requiring more actions from the debugger than just toggling the reset pin, the ResetOut evolved into such commands, which not only assert a reset but also connect to the target. This means that such commands do include communication over the DAP bus.

In order to ensure flexibility, reset-related options like TRST, EnReset, ResBreak and WaitReset can be used to modify/limitate the reset functionality.