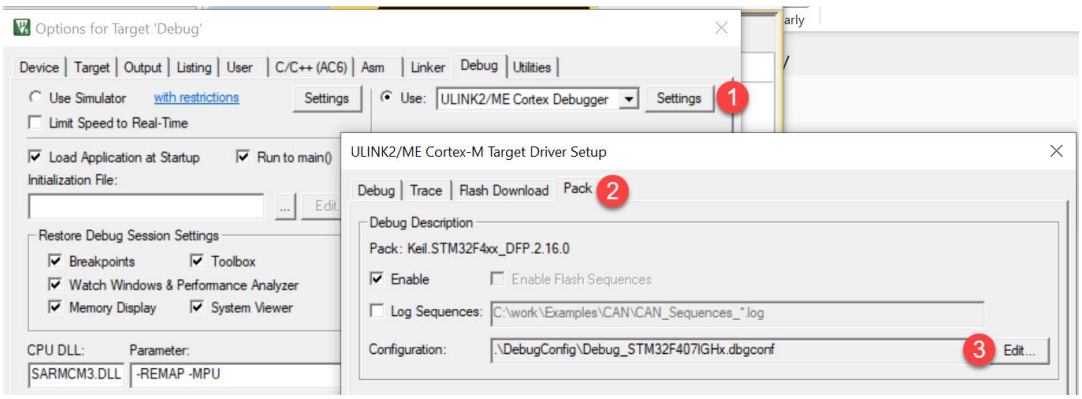
How to enable debugging in low power sleep modes on Cortex-M microcontrollers.

By Trevor Martin

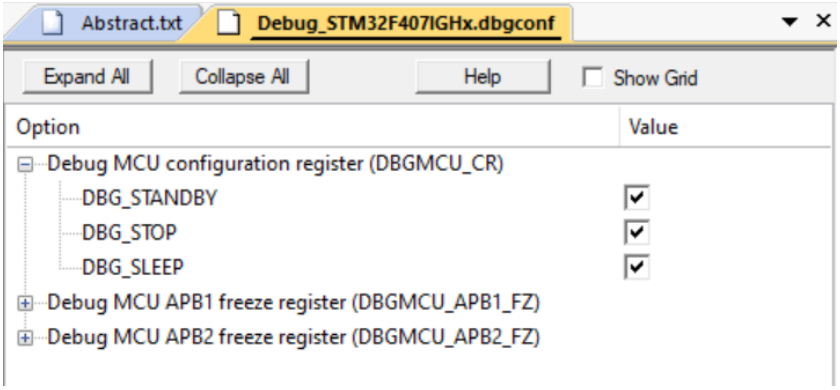
Well supported

Historically debugging low-power applications on an 8 or 16-bit microcontroller was always a challenge. As soon as the processor entered a low-power mode the debugger would lose communication with the processor and terminate the debug session.

Fortunately, low-power debug is a feature that is well-supported within the Cortex-M processor family. It is possible to enter a low-power sleep mode while keeping the debug hardware alive. However, this feature must be enabled either in your code or through a debugger script. Within the Keil debugger, low-power debug is supported by a device-specific script file that is located in the “options for target\debug\settings\pack tab” as shown below.



This file contains a configuration wizard that allows you to enable the debugger low-power support for different sleep modes. Any additional device-specific debug options will also be located in this file.



When you are working with low power modes or adjusting the microcontroller clock tree there is a risk of accidentally upsetting the on-chip debug hardware and locking yourself out of the microcontroller. As you are developing these features it is wise to place a delay loop at the beginning of your code, typically at the very start of the CMSIS void SystemInit(void) function. This will provide enough time for the debugger to connect and halt the processor before the faulty code is executed. Once you are happy with the code the delay barrier can be removed. If the worst does happen the only other option is to change the external boot pins so the processor runs the device ROM code or tries to execute external memory. This will prevent the faulty FLASH code from running and allow you to recover the device and pretend it never happened.

Further Information

For more information visit our website: www.hitex.co.uk or get in touch: info@hitex.co.uk. You can also connect with us: LinkedIn