Real-time Debug

Texas Instruments

Embedded Development Tools

What is Real-time Debug?

- Stop Mode Debug (traditional debugging)
  - require the processor to be completely halted to access memory and registers
  - stops all threads and prevents interrupts from being handled
- Stop Mode can be used as long as system/application does not have real-time constraints, but is very undesirable for real-time applications
- Real-time Mode Debug enables programmers to:
  - examine and modify contents of memory/register locations while CPU is running and executing code
  - halt/debug application while allowing user specified time critical interrupts to be serviced without interference
- Real-time debug capabilities vary by device and are supported via different methods

Real-time Debug capabilities

- Access to memory while the processor is running
  - Supported on Tiva, Stellaris, Hercules, C28x, C66x, C64x, C55x
  - On ARM it is enabled through the DAP (Debug Access Port)
    - DAP is part of the ARM emulation logic and enables the debugger to access memory of the device without requiring the processor to enter the debug state
  - On DSP/28x it is enabled via ICEMaker hardware
  - Also possible on Cortex A devices but is not recommended when the MMU is in use (will always show physical memory and not virtual)
- Access to registers while the processor is running
  - Supported on C28x, C66x, C64x, C55x
  - Enabled through hardware (ICEMaker)
- Service interrupts while the processor is halted
  - Supported on C28x, C66x, C64x, C55x
  - Enabled through hardware (ICEMaker)

Real-time Mode: Polite and Rude

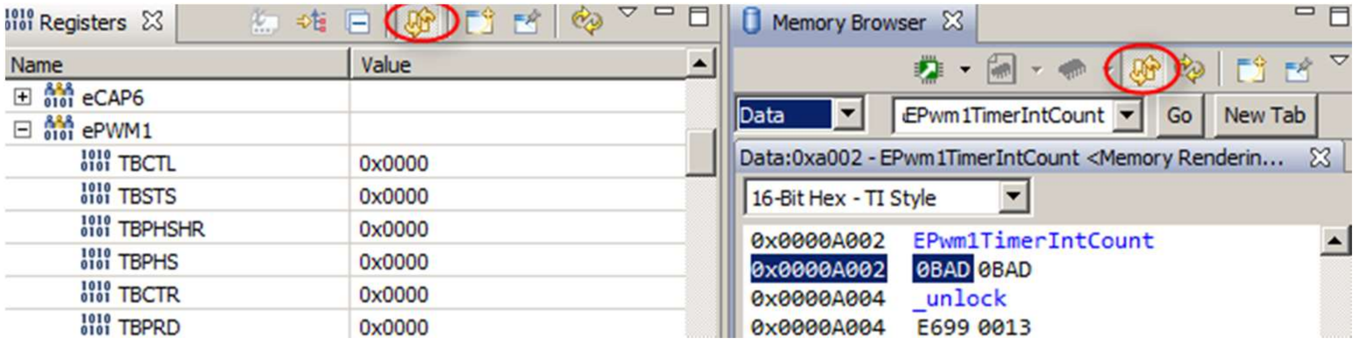
- Memory accesses and halts issued from the debugger can be blocked
  - For example, when in a critical section of code setting the DBGM (Debug Mask) bit in status/control register will block accesses
- Hardware or application can set DBGM bit
  - In some devices hardware sets it automatically when any interrupt is taken
- In Polite real-time mode (default when real-time mode is enabled)
  - DBGM settings are respected by debugger
  - Debugger will not stall the processor to make memory accesses, but rather will wait for processor to get into non-critical section of code before making the access
- In Rude real-time mode
  - DBGM settings are ignored by debugger and accesses are made anyway
  - Allows for error recovery if application sets these bits and then hangs

Real-time Interrupts

- When halted in Stop mode, all interrupts are blocked
- When halted in Real-time mode, time critical interrupts can continue to be serviced
  - The Debug Interrupt Enable Register (DBGIER on C28x) is used to designate time-critical interrupts
  - Interrupts which are enabled by both IER and DBGIER will be serviced when halted in real-time mode, regardless of global interrupt mask bit (INTM on C28x)

Viewing Real-time accesses with CCS

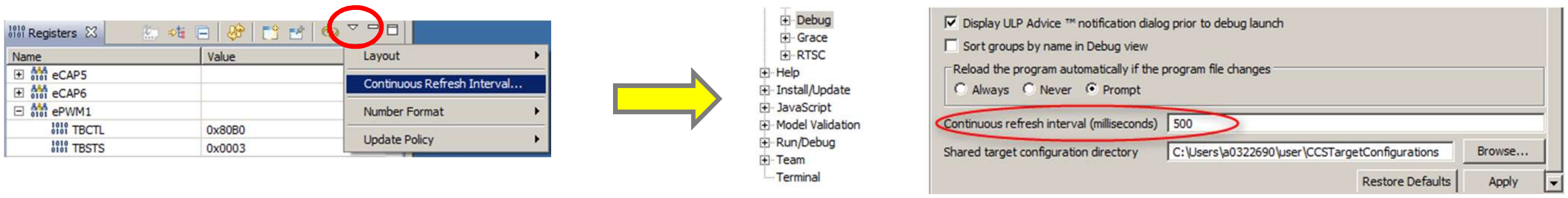
- To enable/view real-time access to memory and registers, click on Continuous Refresh in Memory Browser and Registers view in CCS



Name	Value
eCAP6	
ePWM1	
TBCTL	0x0000
TBSTS	0x0000
TBPHSHR	0x0000
TBPHS	0x0000
TBCTR	0x0000
TBPRD	0x0000

```
0x0000A002   EPwm1TimerIntCount
0x0000A002   0BAD 0BAD
0x0000A004   _unlock
0x0000A004   E699 0013
```

- Continuous Refresh will periodically refresh the debug views
  - Default refresh interval is 500 ms
  - Default refresh interval is configurable



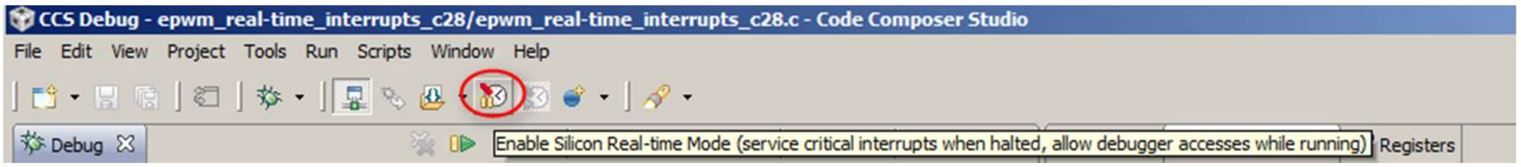
Name	Value
eCAP5	
eCAP6	
ePWM1	
TBCTL	0x80B0
TBSTS	0x0003

[✓] Display ULP Advice ™ notification dialog prior to debug launch
[ ] Sort groups by name in Debug view
Reload the program automatically if the program file changes: Always Never [✓ Prompt]
Continuous refresh interval (milliseconds): 500
Shared target configuration directory: C:\Users\a0322690\user\CCSTargetConfigurations

TEXAS INSTRUMENTS

Using Real-time mode with CCS

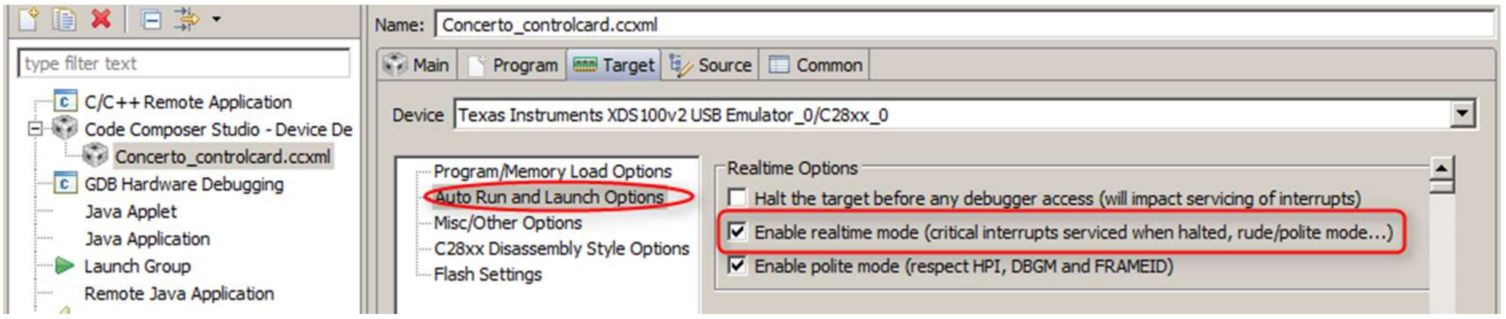
- To enable real-time mode, click on Enable Silicon Real-time Mode icon



CCS Debug - epwm_real-time_interrupts_c28/epwm_real-time_interrupts_c28.c - Code Composer Studio

Enable Silicon Real-time Mode (service critical interrupts when halted, allow debugger accesses while running)

- Or from CCS menu Run->Debug Configurations->Target tab



Name: Concerto_controlcard.ccxml

Device: Texas Instruments XDS100v2 USB Emulator_0/C28xx_0

Realtime Options

- [ ] Halt the target before any debugger access (will impact servicing of interrupts)
- [x] Enable realtime mode (critical interrupts serviced when halted, rude/polite mode...)
- [x] Enable polite mode (respect HPI, DBGM and FRAMEID)

- When enabled:
  - The debugger will respect the DBGM bit and will not stall the processor to make the memory accesses
  - Time-critical interrupts can continue to be serviced while target is halted
  - If enabled prior to launching the debugger, will allow connecting to running device

Texas Instruments

Using Real-time mode with CCS

- When real-time mode is enabled, it defaults to Polite real-time mode
- In Polite mode, debugger will prevent the target from being halted while application is servicing a time-critical interrupt
- If a debugger access requires target to be halted when servicing a time-critical interrupt, user will be asked whether Rude real-time mode should be enabled
- In Rude mode, halt requests are serviced immediately, regardless of whether the processor is executing a time-critical ISR