Code Composer Studio 5.0 Multicore Debugging Demo

Dobrin Alexiev – Texas Instruments Inc.

TI Information - Selective Disclosure





CCS 5.0 - background

- CCS is using DSF to integrate to the backend debugger
- Focused on Device Debugging
- The target system is defined in a target configuration file (CCXML)
 - It is a hierarchy of objects (boards, connections, cores, etc.)
 - Different users want to see different layouts of the target system
 - Each Debug session is using one CCXML file
- Backend support
 - supports synchronous stepping (and other commands) on a group of cores
 - has memory and registers on different node types (not only cores)



Debug View integrations

- Displays the layout of the system
 - as hierarchy of nodes
 - some nodes have stacks
- The user can customize the layout
 - hide types of nodes
 - hide instances of nodes
 - group nodes in custom groups
 - customization can be persisted
- Debug commands can be issued on any node in the hierarchy

Currently implemented layout

- a 💱 Beagle_win32_(TIXDS560).ccxml [Code Composer Studio Device D
 - a 🔗 Non Debuggable Devices
 - TI XDS560 Emulator_0/IcePick_C_0 (Disconnected)
 - TI XDS560 Emulator_0/CS_DAP_PC_0 (Disconnected)
 - TI XDS560 Emulator_0/C64XP_0 (Disconnected : Unknown)
 - TI XDS560 Emulator_0/Cortex_A8_0 (Disconnected : Unknown)

Desired layout

- ▲ C TI XDS560 Emulator_0
 - MAP4430_0
 - ▲ IcePick_D
 - 🔺 🔌 Tesla
 - - ARM9_ICONT1
 - a 🔌 iCONT2
 - ARM9_ICONT2
 - a 💸 Ducati1
 - CS_DAP_RTOS
 Cortex_M3_RTOS
 - Cortex M3 0
 - a 🔌 Ducati2
 - CS_DAP_ISS
 - Cortex_M3_ISS Cortex_M3_1

TI Information – Selective Disclosure



Next steps

- Work with the CDT community to define use cases for:
 - presenting different layouts in the debug view
 - workflows for changing the layouts (grouping, hiding)
- Identify and address DSF changes to support the above use cases, once defined.

TI Information - Selective Disclosure

