BR213IC-HL: How to Build

Thomas Gärtner, 25.01.2013

Install new tools (please see **PF3 Integration Portal >> Documents >> Tool_Guide_PF3_IC**)

- Install tools: python **(rebuild!!!)**, python/QT, putty, cmake
- Install compiler: E003-E004-Pre20 => Multi-ARM v6.1.4_2012.5.4
  E004 => Multi-ARM v6.1.4_2013.1.4 + Patch 2013_03_28
- Update setenv_host.bat, SQE Extensions to V02.06.00

Install sw frame

- Run: **"install_all.bat",** afterwards **"goMSVCnet.bat“**
- **Hint:** **The first versions of sw frame has a path limit of 75 characters (countet drom drive root). Please short the path or name if required!**

Pre Build

- - choose solution: **Multi2000** for target or **Debug** for simulation
- - run from toolbar: Pre Build >> **Prepare Build Workspace** **…that’s all!**

Build Target

- - run from toolbar: Button **MSVC (MakeSolution)** **Attention, different to PF3 Build!!!**

Build Simulation

- - run from menu: Build >> **Build Solution (F7)**

BR213IC-HL: Basics

Integrity

- The new operating system on BR213IC-HL is INTEGRITY. The main difference to the old world is the usage of different virtual address spaces. A virtual address space has a own memory view and is protected against outstanding accesses. The important address spaces for application development are:
  - legacy: contains all non hmi modules (dssprsc (DS-SYS), syman, icom, dsfa, kbd, …)
  - hmi: contains the “new” hmi world.

Monolith / Dynamic Download

- With Integrity its possible to load address spaces in separate parts.
- What is a dynamic download? A project file (*.abs) that contains one or more virtual AddressSpaces which can be downloaded onto an INTEGRITY kernel already running on your target.
- What is a monolith? INTEGRITY kernel, BSP, additional (optional) dynamic downloads

OS – Abstraction layer (OSAL)

- ACE → Adaptive Communication Environment.
  → ACE provides components to abstract synchronisation, communication, memory management, timers, threads, signals, …
- POSIX → Portable Operating System Interface.
  → A set of standards for maintaining compatibility between operating systems. Layer between Integrity and ACE.

BR213IC-HL: Basics

MSVC Environment projects

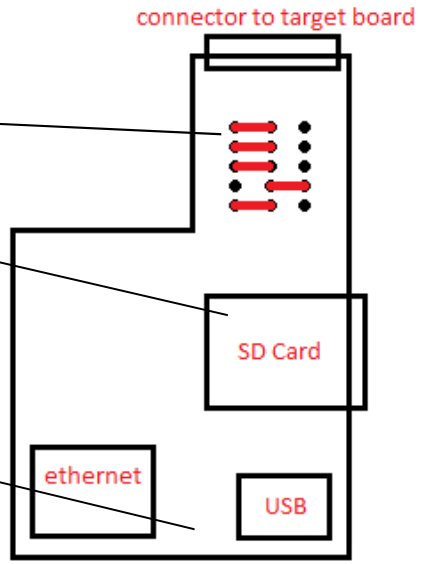
01_CORE	→ Kernelspace_Kernel_proj	INTEGRITY Kernel
	→ Userspace_BaseSystem_proj	Resource Handling and dynamic downloads
	→ Userspace_PosixShmMan_proj	POSIX abstraction
02_APP	→ BaseApp_JCP_Legacy_proj	middleware services and “historical” application
	→ HmiApp_HMI_proj	graphic subsystem,hmi framework and hmi application
03_LIB	→ _BaseServices_proj	compileable library provides common services
	→ _GHS_Integrity_proj	compileable library provides integrity/bsp services
	→ _LibHMIClient_proj	compileable library provides rpc services
	→ _OSAL_proj	compileable library provides os-abstraction layer
10_SIM	→ _Visufrwk_proj	Visualization framework (for target & simulation)
ARMx_proj		Master project file

BR213IC-HL: Debug the target (BR213 High Board)

Thomas Gärtner, 25.01.2013

Prepare Target

- Jumpers: Please set the jumpers at BR213 Debug Board as shown.
- SD Card: You need a already programmed SD card, containing uboot loader. (available from T. Gärtner)
- Connectors: Please connect ethernet (debugger) and USB (seriel) cable



Start Debugger

- Run CanOe, Power Off cluster
- run from toolbar: Button:  TFTP waits for connection
- Power On cluster TFTP load image and start INTEGRITY
  Debugger attaches via ethernet to target
  dynamic downloads “legacy” and “hmi” will loaded via rc script (takes some time)

Start Debug Session

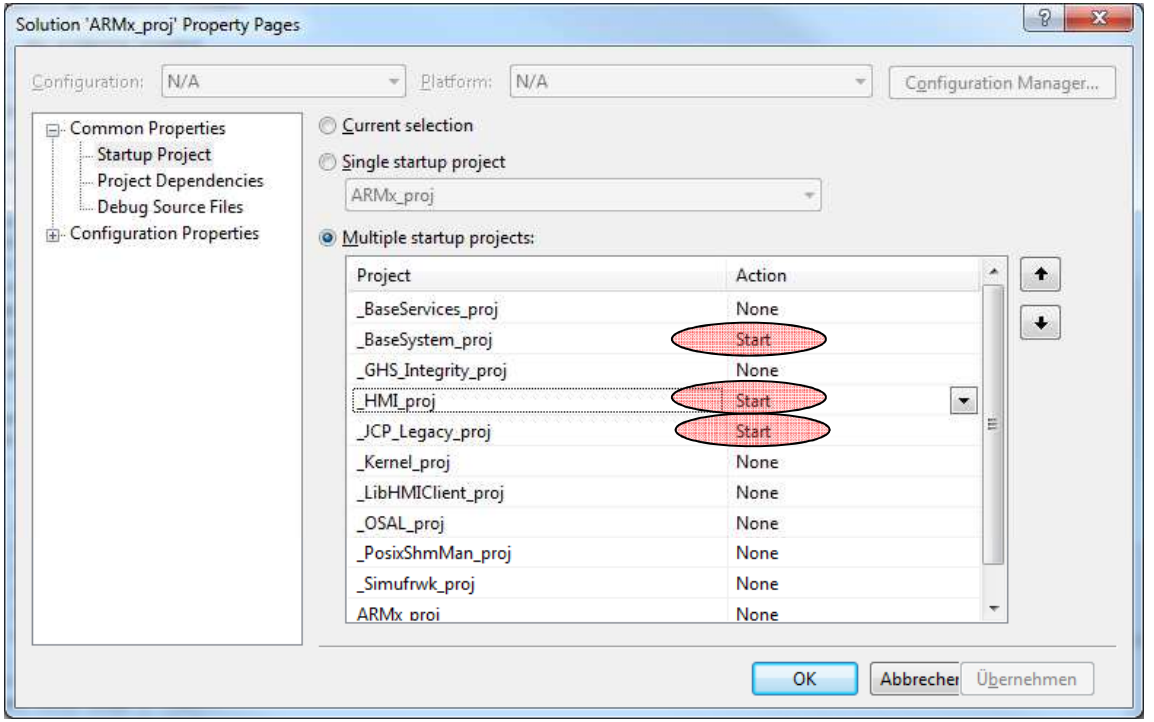
- Select dynamic download (Process) **_JCP_Legacy_proj → Task initial** → **press** Run to start process
- Select dynamic download (Process) **_HMI_proj → Task initial** → **press** Run to start process
- Attention: Breakpoints are Task/Process sensitive, that means if you want to set a breakpoint to a CAM, you have to select at first the related process → _JCP_Legacy_proj.

BR213IC-HL: run simulation

Thomas Gärtner, 25.01.2013

run simulation framework

- - build for solution “Debug” as described.
- - open properties of “**Solution ‘ARMx_proj’ (11 projects)**”:
- - Enable “**Multiple startup projects**”
- - Set BaseSystem, HMI, JCP_Legacy to “**Start**”
- Now the selected processes will start with every new debug session (F5).
- In minor cases it’s necessary to keep process HMI as “None”. Please try in case of simulation startup problems. (of course than you have to start that process manually)



Run Visuframework

- - To have additional information, you can start the visualization framework from MSVC Toolbar.
- Features:
  - Show information of: Running Processes, CRHDL, DPOOL, EVHDL
  - BR213IC-HL Simu Control Panel (very simple stimulation of HMI). This will be replaced asap with a AC Simulation!

BR213IC-HL: Debug the target (EVA Board)

Thomas Gärtner, 25.01.2013

Start Debugger

- - run from toolbar:  Button: The following steps do run automatically:
  Step 1) uBootloader loads the monolith image into RAM.
  Step 2) Integrity + basesys startup
  Step 3) Debugger attaches via ethernet to target (Freescale EVA Board)
  Step 4) dynamic downloads “legacy” and “hmi” will loaded via rc script (takes some time)

Start Debug Session

- Select dynamic download (Process) **_JCP_Legacy_proj → Task initial** → **press** Run to start process
- Select dynamic download (Process) **_HMI_proj → Task initial** → **press** Run to start process
- Attention: Breakpoints are Task/Process sensitive, that means if you want to set a breakpoint to a CAM, you have to select at first the related process → _JCP_Legacy_proj.
- Hint: With using the related AC sw frame it’s possible to control HMI with canoe button’s etc…

