

# AM335x/AMIC110 GEL files

May 2017

# Rahul Prabhu

## Software Application Engineer, Catalog Processors

- **Career**

- Earned Master in Electrical Engineering degree at the University of Houston in Houston, Texas
- Current position: Software applications engineer in Software Applications team at Texas Instruments Dallas

- **Expertise**

- Responsible for developing software solutions for DSP and ARM Processors.
- Extensive experience in Bootloader, Security and RTOS application development.

# Agenda

- GEL Basics and Functions
- GEL GUI Integration
- How to Debug a GEL file
- AM335x ICE GEL file
- Creating a Custom GEL file (Example: AMIC110 ICE)
- Useful tools and utilities.

# GEL Basics:

- **GEL : General Extension Language**

- Similar to C code
- Rich set of built-in functions.

- **Why are GEL files used ?**

- Startup GEL files are used to automate device initialization when Code Composer Studio connects to the target board.
- Best way to bring up new custom hardware and software.
- Primary functions:
  - Initialize device clocks (MPU PLL, CORE PLL, DDR PLL, PER PLL & Display PLL)
  - Power on modules/peripherals
  - Initialize and test external memory like DDR
  - Bring up ARM in known clean state to load and debug

# GEL Basics: GEL functions and Comparison with C

GEL Built-in functions	C Equivalent	Comments
OnTargetConnect()	Main()	First function GEL executes when run/connect to target
GEL_TextOut("string")	Printf("string")	Print string to IO console
GEL_LoadGEL("FileName")	include	Includes other GEL files
for ; if else ; while	Same	
<b>Datatypes:</b> int, float, double, etc	Same	
<b>Operators:</b> +, -, %, ==, &&, >, <	Same	

## Other similarities:

- Macros defines
- Pointer and addressing
- Use Semicolon at the end of each line of code
- Commenting scheme using /\* Comment \*/ and //

# Advanced GEL Functions:

GEL Built-in Functions	Purpose of these functions
<b>Memory Mapping:</b> GEL_MapOff, GEL_MapAdd, GEL_MapOn, Gel_MapDelete, GEL_MapReset	Creates memory map and provides guidance regarding memory access (r, w, ASx)
<b>Callback functions:</b> OnReset, OnRestart, OnHalt, OnPreFileLoaded, OnFileLoaded	Callback function based on emulation events.
<b>Timer functions:</b> GEL_SetTimer, GEL_CancelTimer	Create a periodic timer that call another GEL function when the timer goes off
<b>GEL Trace:</b> GEL_Trace (level)	Enable GEL tracing for debugging

## References :

- **App Notes:** [Creating Device Initialization GEL Files](#)
- [http://processors.wiki.ti.com/index.php/GEL\\_Example\\_-\\_Periodic\\_Recoding\\_of\\_a\\_Variable](http://processors.wiki.ti.com/index.php/GEL_Example_-_Periodic_Recoding_of_a_Variable)

# GEL Integration with CCS GUI :

- **Hotmenus**

It is possible to associate a GEL function with a menu item. These menu items will show up on the User Scripts menu in the Debug perspective in CCS.

- **Sliders**

Sliders allow you to attach function that adjusts a variable to a GUI slider control

- **Dialogs**

GEL dialogs prompt you to enter a value that is used as a parameter to a GEL function.

- **Loaded GEL Files**

You can view loaded GEL files by going to the Control Panel in CCS, from the tools menu select **GEL Files**.

- **GEL Console Output**

GEL\_TextOut() output is directed to the CCS Console output window.

Reference: [http://processors.wiki.ti.com/index.php/GEL#GUI\\_Integration](http://processors.wiki.ti.com/index.php/GEL#GUI_Integration)

# GEL Register Access:

- Accessing registers from GEL can be done using a direct definition using `#define` (details at the online CCS Help) or using any register name shown in the *Registers* view of CCS.

```
//*****  
//Watchdog Timer registers  
//*****  
#define      WDT1_BASE_ADDR                (0x44E35000)  
#define      WDT1_WSPR                    (WDT1_BASE_ADDR + 0x48)  
  
//*****  
//Read write prototype  
//*****  
#define WR_MEM_32(addr, data) *(unsigned int*)(addr) = (unsigned int)(data)  
#define RD_MEM_32(addr)      *(unsigned int*)(addr)  
#define UWORD32                unsigned int
```

## Writing to the register

```
//Watchdog is disabled to eliminate warm resets during debug  
Disable_Watchdog()  
{  
    WR_MEM_32(WDT1_WSPR, 0x0000AAAA);  
    WR_MEM_32(WDT1_WSPR, 0x00005555);  
}
```



# Debugging a GEL file using GEL\_Trace

- This feature allows you trace the execution of GEL scripts using GEL built in function **GEL\_Trace( *level* )**  
where *level* is an integer 0, 1 or 2.
  - 0 - no trace (default value)
  - 1 - trace entry/exit of GEL functions
  - 2 - trace execution of each statement in the functions (including entry/exit)

By default, the trace will be sent to the CCS console. You can redirect the output using **GEL\_EnableFileOutput** and **GEL\_DisableFileOutput**.

# AM335x ICE GEL file

- Live demonstration and look at the TMDXICE3359.gel and review of GEL concepts.

# Creating Custom GEL File:

- **Getting Started:** Always use TI evaluation platform GEL as a starting point.

**GEL Location:** ccsv7\ccs\_base\emulation\boards\ice\_am3359\gel

- **Modifying and confirming the Clocks**

- Critical to hookup SYSBOOT[15:14] pins correctly
- TI provides clock setting for all CLKIN values based on values read from SYSBOOT if you want to program it to any OPP supported frequency.
- CLKOUT1 and CLKOUT2 observation pins can be used to confirm clocks.
  - NOTE: You can use a scope to confirmed MPU PLL

**Clock Tree tool:** [AM335x Clock Tree Tool](#) AM335x DPLL Calculator

- **DDR configuration:**

- DDR PHY configuration and software leveling
- EMIF timing configuration

**DDR Configuration tools:**

- [AM335x DDR PHY register configuration for DDR3 using Software Leveling](#)
- [AM335x EMIF Configuration tips](#)
- [Tuning the DDR3 Timings on BeagleBoneBlack Workshop](#)

**Good Reference Example for DDR changes:**

AM335x ICE vs AMIC110 ICE GEL comparison.

# When do I switch from using GEL to Bootloaders.

- Users can rely on a GEL file to perform tasks such as clock and EMIF initialization only in debug environment.
- Gel is only used to develop confidence in the SOC and board configuration
- When your application nears production, the settings made in GEL—for example, to configure clock and the DDR timing—need to be moved to your bootloader code.
- Moving code from GEL to bootloader is simple as the code is already using C syntax.
- Easiest way to modify the software is to diff the GEL for the platforms and update the values in an existing TI board software.

# Back Up Slide:

Figure 8-15. CLKOUT Signals

