

# Code Composer Studio v4

Why should I upgrade?

# Agenda

- Summary
- Schedule
- Benefits

# CCSv4 Summary

- What is it?
  - Major upgrade to CCS
  - Based on Eclipse open source software framework
  - New registration/licensing/updating mechanism and model
- Why Eclipse?
  - Quickly becoming a standard for IDEs
  - Excellent software architecture
  - Ability to leverage the work of others
  - Wide selection of 3<sup>rd</sup> party plug-ins available
- When?
  - Try it out today!
- How?
  - Restructuring of our debug stack
  - Porting of existing features to Eclipse
  - Taking the time to make sure migration will be as smooth as possible

# Schedule

- Beta 1: October 2008
- Beta 2: December 2008
- Beta 3: March 2009
- Release to Market: June 2009

Reasons to upgrade

# **BENEFITS**

# Windowing Environment

- Problems:
  - Today's embedded IDEs offer a large selection of features however fitting all of your windows into the IDE is a challenge
  - You use different windows at different times
- Solutions:
  - A comprehensive windowing solution that allows you to maximize the available screen space but still have all functionality at your finger tips
  - Ability to create different perspectives that have the windows that you use most for a given development activity readily available

# CCSv3.3 Environment

**Limited space for windows  
Often have to make windows small  
Can only have a few windows open**

```
*****  
Description: This application uses Probe  
(a sine wave). It then takes this signal,  
factor to it.  
Filename: Sine.c  
*****  
#incl  
#inc  
// g  
int  
// declare and initialize a IO buffer  
BufferContents currentBuffer;  
// Define some functions  
static void processing();  
static void dataIO();
```

Address	Disassembly	Comment
0x0000E20	main	
0x0000E20	0x01BC54F6	0x00002000
0x0000E28	0x00029810	0x0187202A
0x0000E30	0x0208C028	0x02400068
0x0000E38	0x0180006A	0x00000000
0x0000E40	C9RL0, C9DW\$L\$_main\$29B,	
0x0000E40	0x00001F10	0x0187282A
0x0000E48	0x0180006A	0x00004000
0x0000E50	C9RL1	

Address	Disassembly	Comment
00002280		c_int00:
00002280	07857A2A	
00002284	07C0006A	
00002288	07BF09F2	
0000228C	07082E2A	
00002290	0740006A	
00002294	0208C02A	
00002298	0200006A	

Build Complete,  
0 Errors, 0 Warnings, 0 Remarks.

Register	Value
A0	00000000
A1	00000000
A2	00000000
A3	00000000
A4	00000000
A5	00000000
A6	00000000

File: C:\Development\Tools\CCSv3.3\tutorial\sim64xx\sinewave\sine.c Ln 1, Col 1

# CCSv4 Environment

**Customize toolbars & menus**

**Perspectives contain separate window arrangements depending on what you are doing.**

**Tabbed editor windows**

**Tab data displays together to save space**

**Fast view windows don't display until you click on them**

```
1#include <stdio.h>
2#include "main.h"
3
4void main(void)
5    john(1);
6    john(0);
7}
8
9void john(int flag) {
10    if (flag == 1) {
11        printf("hello world\n");
12    }
13    else {
14        rocks();
15    }
16}
```

Name	Value

Name	Value
Core Reg	
RegisterPairs	

Address	Disassembly	Comment
0x118056e4	011B CALLP.S2	john (PC+16
0x118056e6	2626 MVK.L1	1,A4
	C\$RLO:	
0x118056e8	011B CALLP.S2	john (PC+16
0x118056ea	0626 MVK.L1	0,A4
	C\$RL1:	
0x118056ec	71F7 LDW.D2T2	+++B15[2],B
0x118056ee	A1EF BNOP.S2	B3,5
	john:	
0x118056f0	01BC94F6 STW.D2T2	B3,*SP--[4]

```
Initializing ..... (Completed)
js:> |
```

# Source Code Editor

- Problem:
  - Most IDEs contain an editor with limited functionality requiring the purchase of an additional external editor
- Solution:
  - CCSv4 includes an excellent editor with equivalent functionality to the majority of commercial editors
    - Code completion (auto-parameter info...)
    - Jump to definition/declaration
    - Outline view of current source file
    - Local history of source file changes
    - Compare files
    - Back/forward/back to last edit location
    - ...

# Multi-processor Environment

- Problem:
  - Many devices today include more than one processing core and often reside in a system with many other devices. Displaying debug information from many different cores typically requires many IDE windows.
- Solution:
  - CCSv4 allows you to have a single IDE window and to change the debug context of the IDE to any of the cores in the system.
  - You can also “pin” the context of a debug display to a specific core.
  - If desired you can open a top level IDE for any core

# CCSv3.3 Multi-core Environment

The image displays the Code Composer Studio (CCS) interface for a multi-core environment. On the left, a stack of windows represents individual cores, each titled "/TC16486 Simulator, little endian/C64+\_1 - 64xx (Simulator) - Code Composer Studio". The main window shows the IDE with a file explorer on the left containing "GEL files" and "Projects". The status bar at the bottom indicates "LE" and "For Help, press F1".

On the right, the "CCStudio: Parallel Debug Manager" window is open, showing a table of core statuses:

Name	CPU Status	Processor	Mode	Program	Endianness	OS
TC16486 S...	Halted	64xx	Stop-mode[...]	Unknown	Little Endian	None
TC16486 S...	Halted	64xx	Stop-mode[...]	Unknown	Little Endian	None
TC16486 S...	Halted	64xx	Stop-mode[...]	Unknown	Little Endian	None
TC16486 S...	Halted	64xx	Stop-mode[...]	Unknown	Little Endian	None
TC16486 S...	Halted	64xx	Stop-mode[...]	Unknown	Little Endian	None
TC16486 S...	Halted	64xx	Stop-mode[...]	Unknown	Little Endian	None

Two callout boxes provide additional information:

- Parallel debug manager to see status of all cores** (points to the Parallel Debug Manager window)
- Separate top level IDE windows for each core Can actually run out of windows resources** (points to the stack of simulator windows)

# CCSv4 Multi-core Environment

Debug - main.c - Code Composer Studio

File Edit Refactor Navigate Search Project Target Scope Tools Profile Run Window Help

Debug

TCI6486 Simulator, little endian/C64+\_1 [Non-Project Debug Session]

TCI6486 Simulator, little endian/C64+\_6 [Non-Project Debug Session]

TCI6486 Simulator, little endian/C64+\_4 [Non-Project Debug Session]

main.c

```
1#include <stdio.h>
2
3void main(void) {
4    printf("hello world\n");
5}
6
7
```

Disassembly

main:

Address	Hex	Asm	Comment
0x00806540	01BC94F6	STW.D2T2	B3,*SP--[4]
0x00806544	00002000	NOP	2
0x00806548	01909228	MVK.S1	0x2124,A3
0x0080654c	01807868	MVKH.S1	0xf00000,A3
0x00806550	1FFF6013	CALLP.S2	printf(PC-1280 =
0x00806554	01BC22E4	STW.D2T1	A3,*+SP[1]
		LDW.D2T2	++SP[4],B3
		.fthead	p, l, W, BU, nobr,
		BNOP.S2	B3,5
0x00806564	00000000	NOP	
0x00806568	00000000	NOP	

Console

TCI6486 Simulator, little endian/C64+\_1 [Non-Project Debug Session] TCI6486 Simulator, little

Scripting Console

Initializing .... (Completed)

js:>

LE Writable Smart Insert 3:1

Use the Debug view to select the context

Displays show content for the current debug context

# Project Management

- Problem:
  - Typically you have more than one project on going at a time, with each project being at a different stage in development and often using different versions of compile tools or operating systems.
- Solution:
  - CCSv4 allows you to set the version of the compiler and DSP/BIOS that each individual project will use. Allowing projects in maintenance mode to continue to use the tools they were deployed with and enabling new projects to use the latest high performance tools

# Tool Integration & Customization

- Problem:
  - More than just an embedded debugger is required during product development
- Solution:
  - CCSv4 is based on Eclipse which has a huge selection of 3<sup>rd</sup> party plug-ins available (code analysis, source code control, modelling, Perl development...)
    - <http://www.eclipseplugincentral.com>
  - The Eclipse plug-in development environment allows for the creation of your own custom tooling
    - Wizards for creating plug-ins quickly

# Scripting

- Problem:
  - Some tasks such as testing need to run for hours or days without user interaction
  - Need to be able to automate common tasks
- Solution:
  - CCSv4 has a complete scripting environment allowing for the automation of repetitive tasks such as testing and performance benchmarking.
  - The CCSv4 Scripting Console allows you to type commands or to execute scripts within the IDE.

# Image Analysis

- Problem:
  - Analyzing the output of an imaging or video algorithm requires looking at the data in its native format (i.e. the image or a frame of video).
- Solution:
  - The Image Display in CCSv4 supports viewing images in many different formats.
    - Ex. Interleaved YUV 4.2.2

# IDE Familiarity

- Problem:
  - Developers work with a number of different development environments. Thus needing to become familiar with the work flow of different tools.
- Solution:
  - CCSv4 is based on the Eclipse open source software framework which is used by many different embedded development environments:
    - ARM Ltd, MontaVista, Enea, WindRiver, QNX...

# Licensing

- Problems:
  - Mid to large size customers want floating (server) license options
  - Free Evaluation Tools and DSK tools are out of date the day they are created
- Solution:
  - Integration of FlexNET licensing allows for a variety of licensing options (node locked, floating, time based...).
  - Full tools, DSK tools, Free tools are all the same image and are kept up to date via the update manager

# Update Delivery

- Problems:
  - People are unsure of what updates are needed
  - Downloading updates is painful
- Solution:
  - CCS will automatically check for updates on startup and indicate if content is available
  - Spectrum Digital & Blackhawk drivers are included in the CCS install
  - Service releases will only install content relevant to your installation (i.e. C2000 users only see C2000 content)
  - Much faster file server!!!!!!!!!!

# Getting Involved with CCSv4

- Website:
  - [http://tiexpressdsp.com/index.php?title=Category:Code\\_Composer\\_Studio\\_v4](http://tiexpressdsp.com/index.php?title=Category:Code_Composer_Studio_v4)
  - Feel free to contribute
  - Contains links to download CCS, helpful tips & tutorials