TASKING®

HYCON

TASKING VX-toolset for MCS

Kees Tieleman

Product Owner

Detroit, October 10, 2017

tasking.com

AGENDA

- About the company
- Why TASKING GTM-IP MCS toolset
- Features Overview
- Practical Applications of TASKING toolset – Hybrid Controls
- Q&A

Altium - TASKING

- Altium does have about 500 employees worldwide
- Altium does close to 100mio revenues
- Revenue Growth of 14%
- TASKING is the Embedded Software Division within Altium

Altium Global Reach

TASKING®

HYCON

Sales and Support Centres

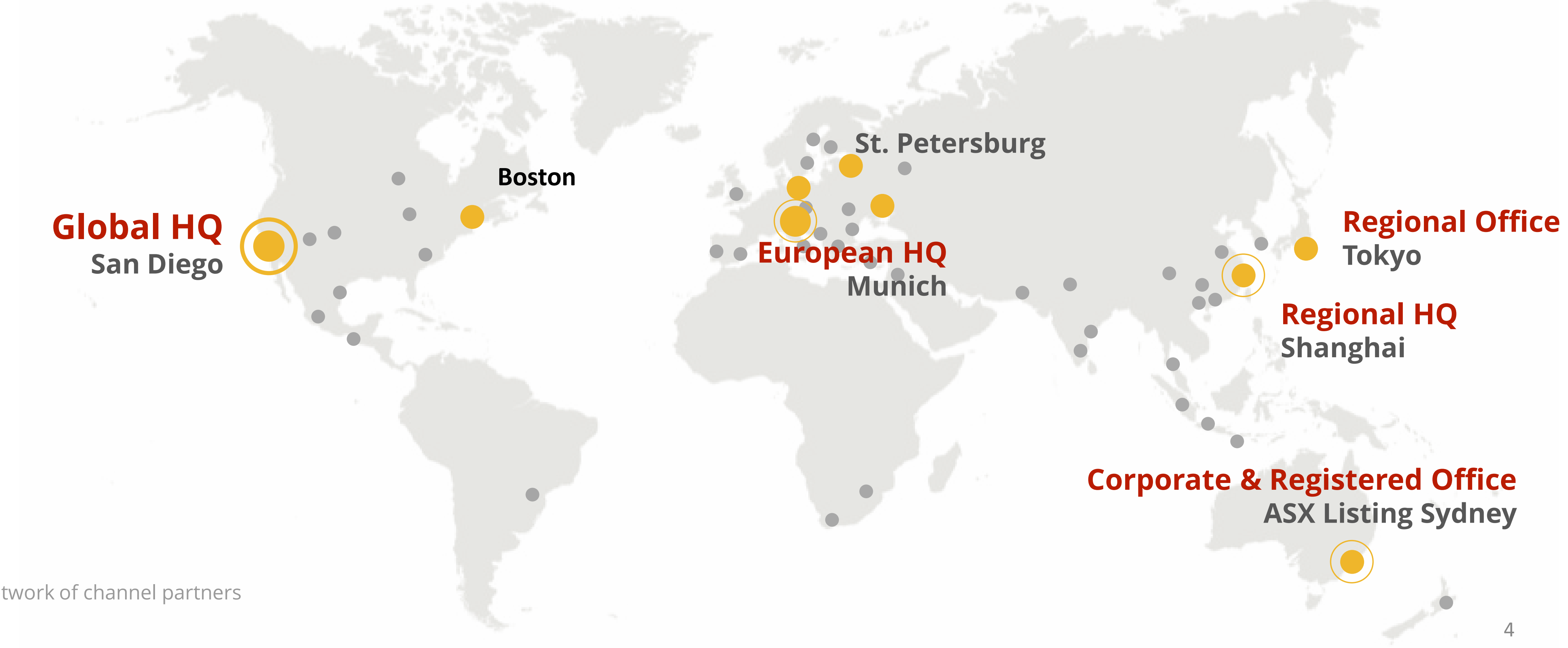
San Diego, Boston, Munich, Karlsruhe, Shanghai, Tokyo, Sydney

Research & Development Centres

San Diego, Ukraine, Netherlands, Shanghai, Munich, St. Petersburg

Main Operations Centres

San Diego, Karlsruhe, Shanghai



● Network of channel partners

TASKING Key Facts

- **25** Years of experience with dedicated team

- Committed to deliver:
  - High performance and high quality tools
    - Focus on safety and industry standards
  - Tools to help you with automotive and multicore applications
  - This with and worldwide sales and support organization

- Field-proven tools used by various OEM and Tier-1 suppliers.

- In-house developed compiler technology.

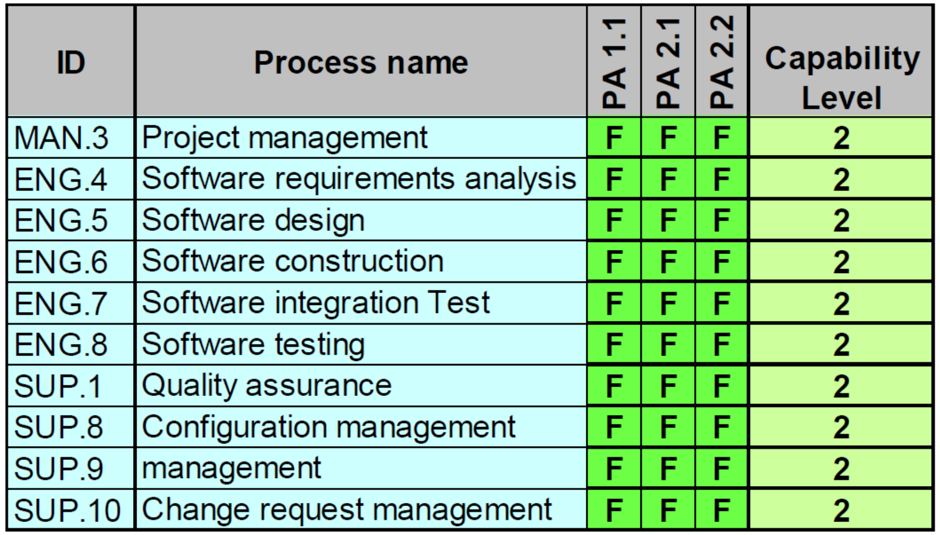
- Strong cooperation with Bosch

Automotive SPICE (ASPICE)

- **TASKING** has ASPICE capability level 2

(**A**utomotive **S**oftware **P**rocess **I**mprovement and **C**apability Determination)

- **Signifies the development process to provide quality, reliable products**



ID	Process name	PA 1.1	PA 2.1	PA 2.2	Capability Level
MAN.3	Project management	F	F	F	2
ENG.4	Software requirements analysis	F	F	F	2
ENG.5	Software design	F	F	F	2
ENG.6	Software construction	F	F	F	2
ENG.7	Software integration Test	F	F	F	2
ENG.8	Software testing	F	F	F	2
SUP.1	Quality assurance	F	F	F	2
SUP.8	Configuration management	F	F	F	2
SUP.9	management	F	F	F	2
SUP.10	Change request management	F	F	F	2

(N) Not achieved (0 - 15%)

(P) Partially achieved (>15% - 50%)

(L) Largely achieved (>50%- 85%)

(F) Fully achieved (>85% - 100%).

History of GTM-IP development

- The instruction set of GTM-IP cores of the first and second generation isn‘t designed for a C compiler
- However GTM-IP cores of the third generation have been designed with C compliance in mind
- TASKING worked closely together with BOSCH to be able to provide the world first fully compliant C compiler

GTM-IP MCS Compiler Key Benefits

- Fully ISO-C99 compliant C compiler (for GTM 3.0/3.1)
  - Program your applications in C while keeping access to GTM details
- Benefits from our long-year experience in optimization techniques
  - You must be an experienced assembly programmer if you want to beat the compiler
- Can be used as a standalone toolset for further integration with other compilers (C-array image output) or integrated in our TASKING TriCore and RH850 toolsets
  - image to be downloaded from main CPU, global symbols accessible from main CPU
- Allows for the inclusion of (and conversion of) legacy assembler code parts
  - EABI specification included with the product
- Emits information for debuggers in standard DWARF 3 format
- Enhanced safety features through ISO-26262 Safety Kit and the built-in integrated Static Analysis capabilities (MISRA-C, CERT-C)

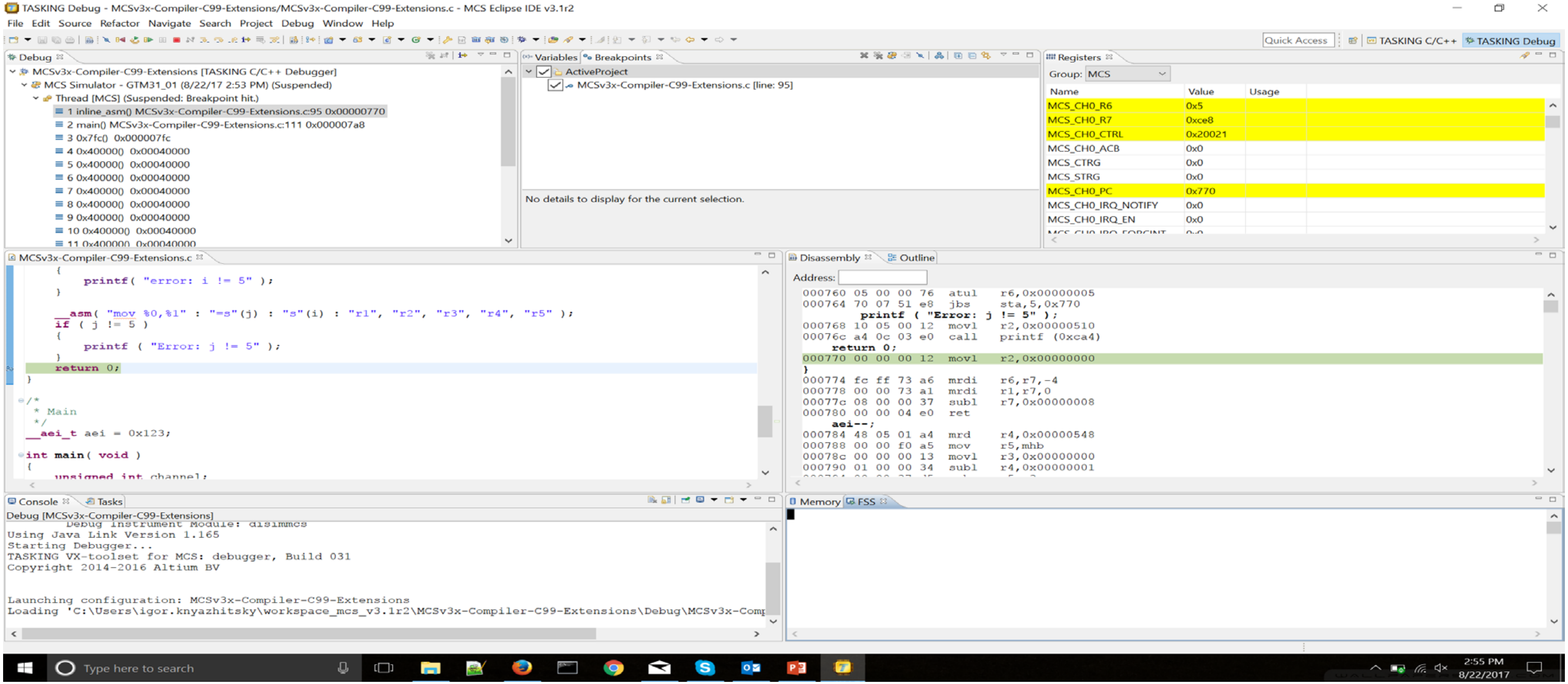
GTM-IP MCS Compiler Target Specifics

- Multiples of 24-bit data types (ints and floats)
  - special types: __int72_t, __uint72_t, _aei_t (32 bit)
- ARU Transfer Intrinsics
  - read, write, non-blocking read, __aru address space qualifier, __aru_t data type
- Bus Master Addressing Intrinsics
  - read, write, __aei address space qualifier, __aei_t data type
- Suspending Wait Instructions Intrinsics
  - using wurmx, wurcx, wuce instructions
- Channel functions
  - __channel function qualifier, channel vector table entry, channel stack initialization

GTM-IP MCS Compiler Target Specifics (ctd.)

- Extended register set usage option
  - compiler uses registers of one MCS for another
- SFR files to access registers from C code
  - automatically select proper address space (__oreg, __xoreg)
- Advanced multi-core linker (predefined configuration files)
  - e.g. swap and borrow memory, bus mappings, channels
  - e.g. channel vector tables, channel stack initialization
- Instruction set simulator for debugger
  - C and assembly level debugging

TASKING Integrated Debugger for MCS



TASKING Debug - MCSv3x-Compiler-C99-Extensions/MCSv3x-Compiler-C99-Extensions.c - MCS Eclipse IDE v3.1r2

```
MCSv3x-Compiler-C99-Extensions [TASKING C/C++ Debugger]
  MCS Simulator - GTM31_01 (8/22/17 2:53 PM) (Suspended)
    Thread [MCS] (Suspended: Breakpoint hit.)
      1 inline_asm() MCSv3x-Compiler-C99-Extensions.c:95 0x00000770
      2 main() MCSv3x-Compiler-C99-Extensions.c:111 0x000007a8
      3 0x7fc()  0x000007fc
      4 0x40000()  0x00040000
      5 0x40000()  0x00040000
      6 0x40000()  0x00040000
      7 0x40000()  0x00040000
      8 0x40000()  0x00040000
      9 0x40000()  0x00040000
      10 0x40000()  0x00040000
      11 0x40000()  0x00040000
```

☑ ActiveProject
☑ MCSv3x-Compiler-C99-Extensions.c [line: 95]

No details to display for the current selection.

Group: MCS

Name	Value	Usage
MCS_CH0_R6	0x5	
MCS_CH0_R7	0xce8	
MCS_CH0_CTRL	0x20021	
MCS_CH0_ACB	0x0	
MCS_CTRG	0x0	
MCS_STRG	0x0	
MCS_CH0_PC	0x770	
MCS_CH0_IRQ_NOTIFY	0x0	
MCS_CH0_IRQ_EN	0x0	

MCSv3x-Compiler-C99-Extensions.c

```c
    {
        printf( "error: i != 5" );
    }

    __asm( "mov %0,%1" : "=s"(j) : "s"(i) : "r1", "r2", "r3", "r4", "r5" );
    if ( j != 5 )
    {
        printf ( "Error: j != 5" );
    }
    return 0;
}

/*
 * Main
 */
__aei_t aei = 0x123;

int main( void )
{
    unsigned int channel;
```

Address:

```
000760 05 00 00 76  atul    r6,0x00000005
000764 70 07 51 e8  jbs     sta,5,0x770
        printf ( "Error: j != 5" );
000768 10 05 00 12  movl    r2,0x00000510
00076c a4 0c 03 e0  call    printf (0xca4)
    return 0;
000770 00 00 00 12  movl    r2,0x00000000
}
000774 fc ff 73 a6  mrdi    r6,r7,-4
000778 00 00 73 a1  mrdi    r1,r7,0
00077c 08 00 00 37  subl    r7,0x00000008
000780 00 00 04 e0  ret
    aei--;
000784 48 05 01 a4  mrd     r4,0x00000548
000788 00 00 f0 a5  mov     r5,mhb
00078c 00 00 00 13  movl    r3,0x00000000
000790 01 00 00 34  subl    r4,0x00000001
```

Debug [MCSv3x-Compiler-C99-Extensions]

```
     Debug Instrument Module: disimmcs
Using Java Link Version 1.165
Starting Debugger...
TASKING VX-toolset for MCS: debugger, Build 031
Copyright 2014-2016 Altium BV


Launching configuration: MCSv3x-Compiler-C99-Extensions
Loading 'C:\Users\igor.knyazhitsky\workspace_mcs_v3.1r2\MCSv3x-Compiler-C99-Extensions\Debug\MCSv3x-Comp
```

Practical applications of the TASKING toolset to program the GTM

- Introducing Hybrid Controls
  - Provide hands-on training, ECU design consulting, and software development services for Bosch GTM-IP
  - Develop NXP MC33816/PT2000 microcode, typically for direct injection applications
  - Provide consulting and design services to establish ETAS ASCET driven software development workflows
  - Develop arbitrary complex device drivers for AUTOSAR applications

- Several real-life examples from Hybrid Controls
  - Flywheel simulator using MCS
  - Crank synchronization using MCS and DPLL
  - 8 cylinder multi-pulse fuel injection using MCS and ATOM
  - Angle synchronous ADS sampling using MCS, ATOM, DMA and SDADC