

Troubleshooting and Run-Time Observation of GTM with PLS's Universal Debug Engine

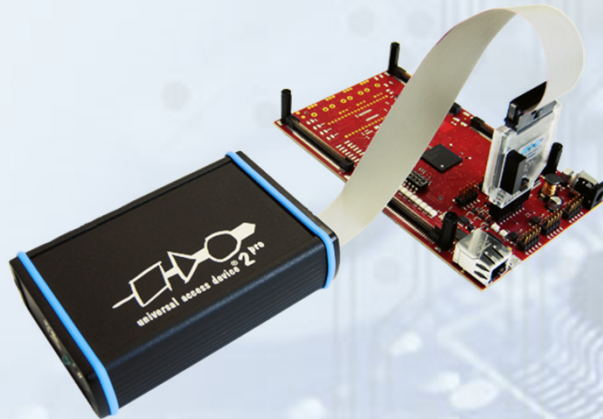
Matthias Noack
PLS Development Tools

PLS Development Tools

More than 25 years in the tool market



Leading Edge Solution
for Debugging and Test

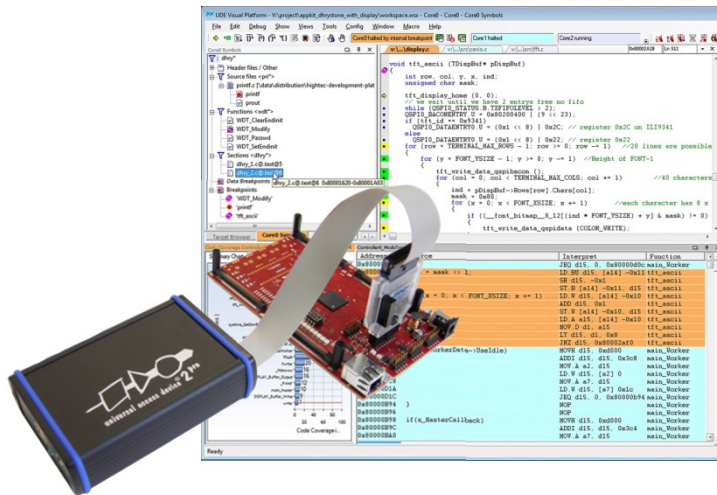
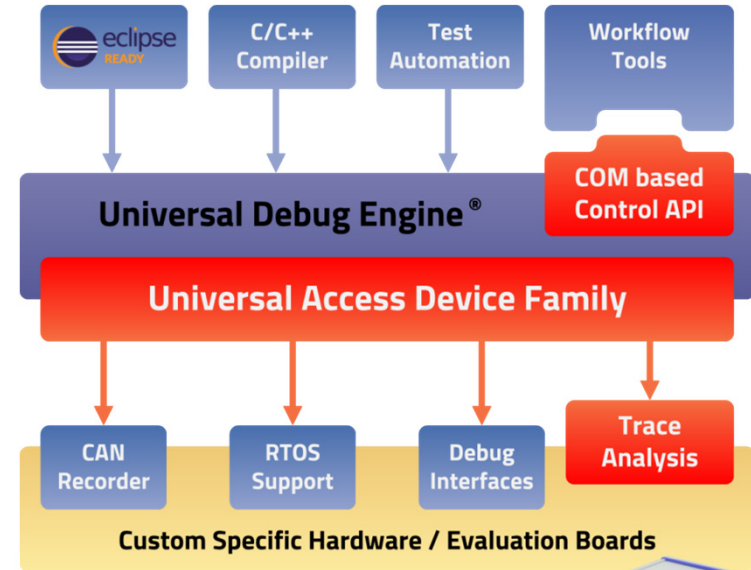


The screenshot displays the UDE Visual Platform interface. The top window shows C code for a terminal display function. The bottom window shows a trace table with columns for Address, Source, Interpret, and Function. Below the trace table is a 'Trace Code Coverage Summary' bar chart.

Address	Source	Interpret	Function
0x8000B84	JEO d15: 0; 0x80000040	main_Worker	
0x8000B88	ID.BU d15: [a14] -0x11	tft_asci1	
0x8000B8C	SH d15: -0x1	tft_asci1	
0x8000B90	ST.B [a14] -0x11, d15	tft_asci1	
0x8000B94	ID.V d15: [a14] -0x10	tft_asci1	
0x8000B98	ADD d15: 0x1	tft_asci1	
0x8000BA4	ST.V [a14] -0x10, d15	tft_asci1	
0x8000BA8	ID.A d15: [a14] -0x10	tft_asci1	
0x8000BAC	MOV.D d1: a15	tft_asci1	
0x8000B80	IT d15: d1, 0x8	tft_asci1	
0x8000B84	JEZ d15: 0x80002af0	tft_asci1	
0x800000C	if(a_WorkerData->UseIdle)	main_Worker	
0x80000D0	ADDI d15, d15, 0x3c8	main_Worker	
0x80000D4	MOV.A a2, d15	main_Worker	
0x80000D8	ID.V d15: [a2] 0	main_Worker	
0x80000DC	MOV.A a7, d15	main_Worker	
0x80000E0	ID.V d15: [a7] 0x1c	main_Worker	
0x80000E4	JEO d15: 0; 0x80000b94	main_Worker	
0x80000E8	MOV	main_Worker	
0x80000EC	MOV	main_Worker	
0x80000F0	MOVH d15, 0xd000	main_Worker	
0x80000F4	ADDI d15, d15, 0x3c4	main_Worker	
0x80000F8	MOV.A a7, d15	main_Worker	

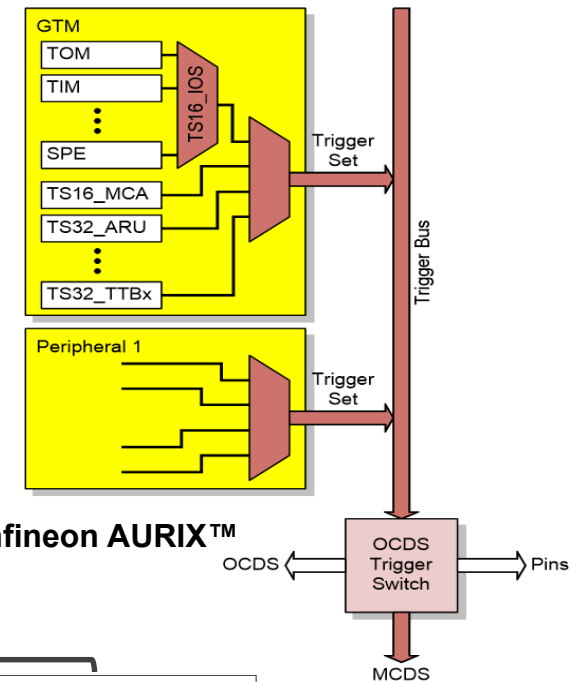
Universal Debug Engine (UDE)

- Complete toolset to debug, test & trace of deeply embedded systems
- Modular architecture
 - Software
 - Debug environment on PC
 - Hardware
 - Target access
 - Support for different Debug I/F

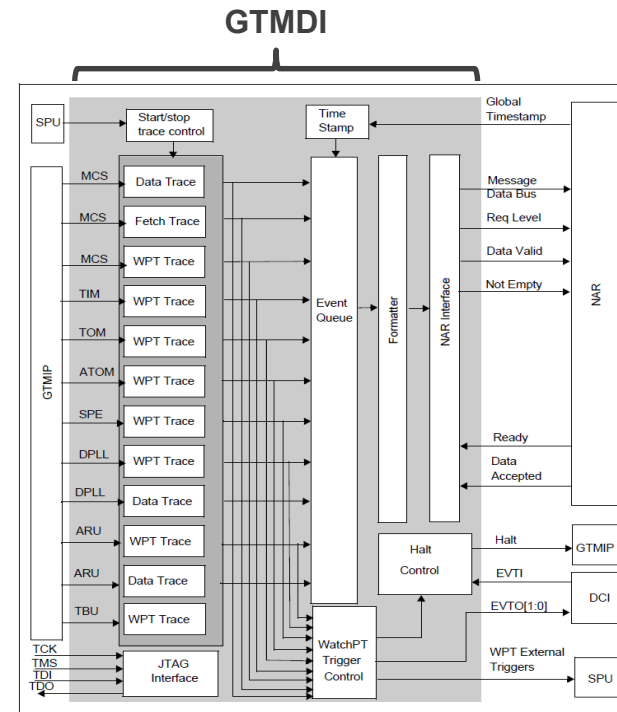


GTM Integration into Debug System

- Vendor specific
- Attached to GTM, not part of it
- GTM debug information connected to SoC's **trace** system
- Support for run-mode debugging as part of GTM IP on the roadmap



Infinion AURIX™



NXP MPC57xx
ST SPC57/SPC58

General Features of GTM Trace (1)

- MCS trace
 - Program trace (fetch trace)
 - Data trace (r/w) for RAM accesses
 - Parallel trace of single channel / multi channels
 - Address compare and triggers for code addresses
- TIM / TOM / ATOM trace
 - Two trace channels → trace of two modules in parallel
- ARU data trace
 - Two trace channels → trace of two ARU debug channels in parallel
 - ARU provides two independent debug channels
 - Configuration of ARU read address

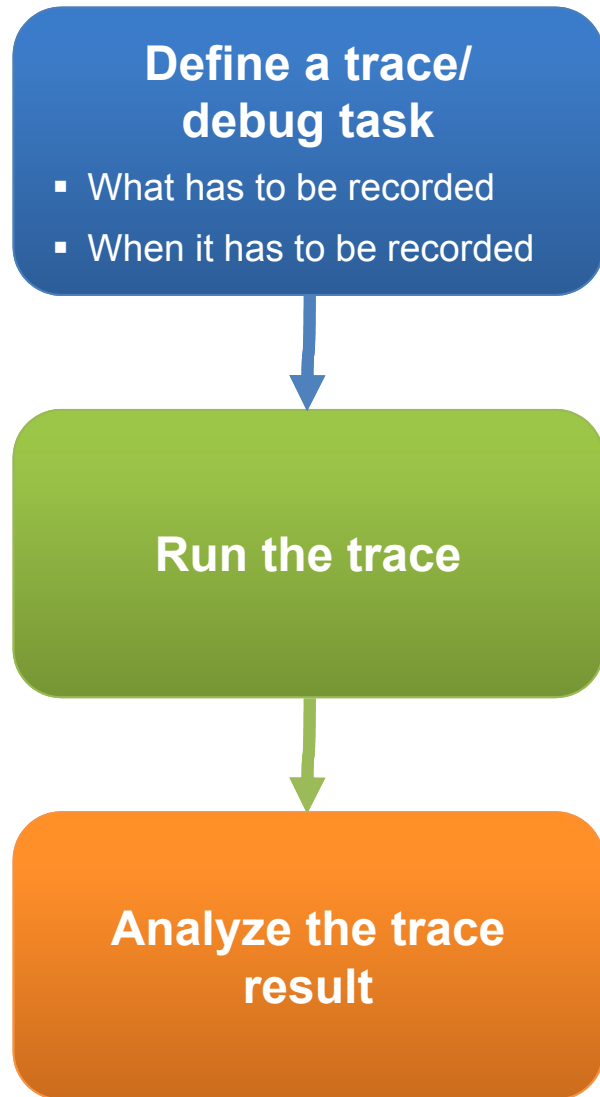
General Features of GTM Trace (2)

- DPLL data trace
 - Trace of one DPLL memory module
- TBU trace
 - Trace of timestamp of one TBU
 - Not available for MPC57xx, SPC57/SPC58 devices
- Parallel trace of GTM and main cores
 - Global timestamp
 - Cross-triggers for signaling GTM ↔ Cores (e.g. trace start / stop)

Debugging Use Cases

- Monitoring
 - Program execution of MCS channels
 - Data transfers (MCS, DPLL RAMs, ARU)
 - Module signals
- Debugging MCS execution
 - Monitoring program flow
- Debug GTM / core interaction
- Synchronous start/stop GTM together with main cores

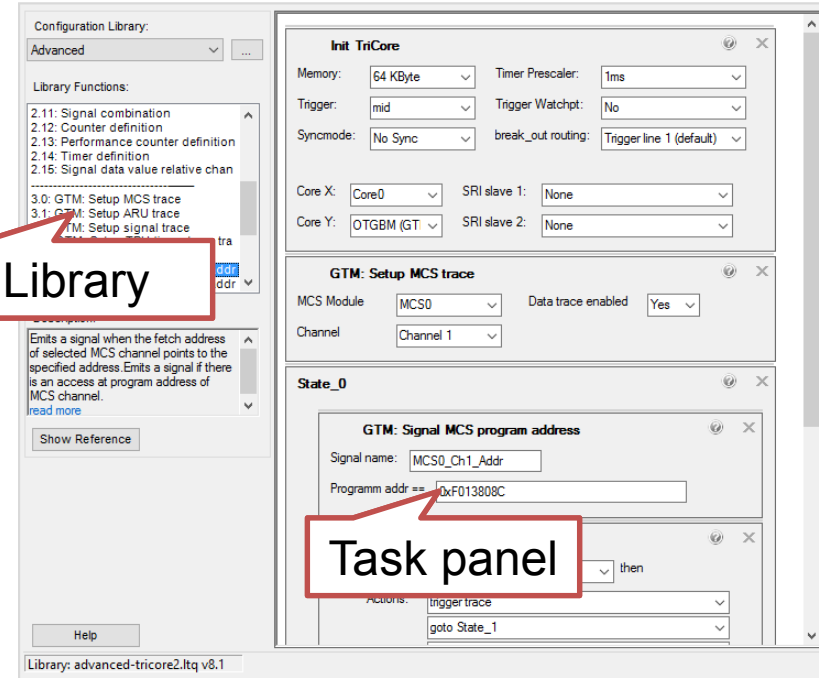
Trace based Debugging



- A trace and debug task can be composed of predefined subtasks
- Examples:
 - Observe system components (program flow, data, signals,...)
 - Initiate actions (enable/disable trace for code / data /..., count, use timers, use sequencers, etc.)

Trace based Debugging with UDE

- Universal Emulation Configurator (UEC)
 - Graphical tool to compose trace and debug tasks
 - Drag'n'drop predefined subtasks (configblocks) from library to task panel



Index	Tick	Address	Data	Interpret	Source	Signals	Submodule
1:	7	110101.34 us	0xF013806C	ARD R1, ZERO, 0x002			MCS0:Ch1
1:	8	115342.92 us	0xF0138070	MWR R1, 0x0108			MCS0:Ch1
1:	9	115343.00 us	0xF0138108	0x00596991 Data: Write			MCS0
1:	10	115343.02 us	0xF0138074	MOVL R3, 0x0000000			MCS0:Ch1
1:	11	115343.04 us	0xF0138078	AVR R1, R3, 0x01			MCS0:Ch1
1:	12	115344.08 us	0xF013807C	SUBL R0, 0x0000001			MCS0:Ch1
1:	13	115344.10 us	0xF0138080	JBS STA, Z, 0x0088			MCS0:Ch1
1:	14	115344.18 us	0xF0138088	MOVL R0, 0x0000060			MCS0:Ch1
1:	15	115344.20 us	0xF013808C	ORL STRS, 0x0000001			MCS0:Ch1
1:	16	115344.20 us		Ref-Time 92724956.029 us			
1:	17	115344.21 us	0xF0138090	MOVL R2, 0x0000002			MCS0:Ch1
1:	18	115344.22 us	0xC000036A	JEQ d15, 0x0, 0xC000046E			
1:	19	115344.23 us	0xC000036A	JEQ d15, 0x0, 0xC000046E			
1:	20	115344.23 us	0xF0138094	WURM R2, STRG, 0x0002			MCS0:Ch1
1:	21	115344.24 us	0xC000046E	CALL 0xC0001228	handle_gtm_tasks();		
1:	22	115344.47 us	0xF0138098	MOVL CTRG, 0x0000002			MCS0:Ch1
1:	23	..	0xC0001228	MOV AA a14, a10	{		
1:	24	..	0xC0001228	MOVH d15, 0xF012	GTM_AFD0_CH1_BUF_A...		
1:	25		
1:	26		
1:	27		
1:	28		
1:	29	115344.		
1:	30	115344.		MCS0:Ch1
1:	31	115344.		
1:	32		
1:	33	..	0xC00040C6	LEA a15, [a15] 0x134			
1:	34	..	0xC00040CA	ID A a14, [a15+1] 0x4			

Parallel trace of several cores and GTM

- Analyze trace
 - Sequence of executed code
 - Data transfers
 - Signals of GTM modules

Example: MCS Program Trace

- Triggered program trace
 - Trace until MCS channel reaches code address

Select GTM as trace source

Select MCS module and channel

Enable program trace and timestamps

Stop trace recording @ code address

Index	Tick	Address	Data	Interpret	Signals	Submodule
6:	110	-10485.81 us+	0xF0138084	JMP 0x006C		MCS0:Ch1
6:	111	-10485.73 us+	0xF013806C	ARD R1, ZERO, 0x002		MCS0:Ch1
6:	112	-5244.15 us+	0xF0138070	MWR R1, 0x0108		MCS0:Ch1
6:	113	-5244.07 us+	0xF0138108	Data: Write		MCS0
6:	114	-5244.05 us+	0xF0138074	MOVL R3, 0x000000		MCS0:Ch1
6:	115	-5244.03 us+	0xF0138078	AWR R1, R3, 0x01		MCS0:Ch1
6:	116	-5242.99 us+	0xF013807C	SUBL R0, 0x000001		MCS0:Ch1
6:	117	-5242.97 us+	0xF0138080	JBS STA, Z, 0x0088		MCS0:Ch1
6:	118	-5242.95 us+	0xF0138084	JMP 0x006C		MCS0:Ch1
6:	119	-5242.87 us+	0xF013806C	ARD R1, ZERO, 0x002		MCS0:Ch1
6:	120	-1.29 us+	0xF0138070	MWR R1, 0x0108		MCS0:Ch1
6:	121	-1.21 us+	0xF0138108	Data: Write		MCS0
6:	122	-1.5241.58 us between #120 and #119		MOVL R3, 0		MCS0:Ch1
6:	123	-1.17 us+	0xF0138078	AWR R1, R3		MCS0:Ch1
6:	124	-0.13 us+	0xF013807C	SUBL R0, 0		MCS0:Ch1
6:	125	-0.11 us+	0xF0138080	JBS STA, Z		MCS0:Ch1
6:	126	-0.03 us+	0xF0138088	MOVL R0, 0		MCS0:Ch1
6:	127	-0.01 us+	0xF013808C	ORL STRG, 0x000001		MCS0:Ch1
6:	128	0.00 us+		Trigger		MCS0:Ch1

Stop trace recording @ code address

Example: GTM Signal Trace

- Trace of GTM signals
 - TOM_OUT
 - TIM F_OUT
 - ATOM_CHx_OUT
 - ...
- Events based on GTM signals

Select signals and filter

Start trace recording @ signal rise

Enable signal trace and timestamps

Mark rising of events

Index	Tick	Address	Data	Interpret	Signals	Submodule
0:	0	0.00 us			TOM_OUT_low=0b????????1	TOMO
0:	1	0.00 us			F_OUT=0b????????0	TIMO
0:	2	0.01 us		TOMO_CH0 -> 1		
0:	3	0.07 us			TOM_OUT_low=0b????????1	TOMO
0:	4	0.07 us			F_OUT=0b????????1	TIMO
0:	5	0.08 us		TIMO_CH0 -> 1		
0:	6	2621.44 us			TOM_OUT_low=0b????????0	TOMO
0:	7	2621.44 us			F_OUT=0b????????1	TIMO
0:	8	2621.50 us			TOM_OUT_low=0b????????0	TOMO
0:	9	2621.50 us			F_OUT=0b????????0	TIMO
0:	10	5242.88 us			TOM_OUT_low=0b????????1	TOMO
0:	11	5242.88 us			F_OUT=0b????????0	TIMO
0:	12	5242.89 us		TOMO_CH0 -> 1		
0:	13	5242.95 us			TOM_OUT_low=0b????????1	TOMO
0:	14	5242.95 us			F_OUT=0b????????1	TIMO
0:	15	5242.96 us		TIMO_CH0 -> 1		
0:	16	7864.32 us			TOM_OUT_low=0b????????0	TOMO
0:	17	7864.32 us			F_OUT=0b????????1	TIMO
0:	18	7864.38 us			TOM_OUT_low=0b????????0	TOMO
0:	19	7864.38 us			F_OUT=0b????????0	TIMO



Example: GTM / Core Interaction

- Monitoring GTM interrupt
 - Parallel trace of MCS channel and main core
 - Exact timing of interrupt handling

Enable monitoring of GTM and Core0

Select MCS module and channel

Capture program trace

Trace recording around Core0 interrupt handler

Init TriCore

Memory: 64 KByte Timer Prescaler: 1ms

Trigger: mid Trigger Watchpt: Yes

Syncmode: Sync break_out routing: Trigger line 1 (default)

Core X: Core0 SRI slave 1: None

Core Y: OTGBM (GT) SRI slave 2: None

GTM: Setup MCS trace

MCS Module: MCS0 Data trace enabled: No

Channel: Channel 2

Emit actions

Actions: ticks on
store Core X PC
store GTM trace (observed by Core Y)

Signal program address

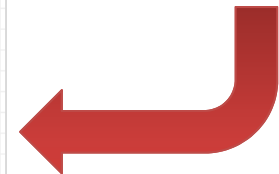
Signal name: isr_gtm_mcs0_ch2

Core X PC == isr_gtm_mcs0_ch2

Actions on condition

If isr_gtm_mcs0_ch2 then

Actions: trigger trace



Rise IRQ

Interrupt @ Core0

Begin of IRQ handler

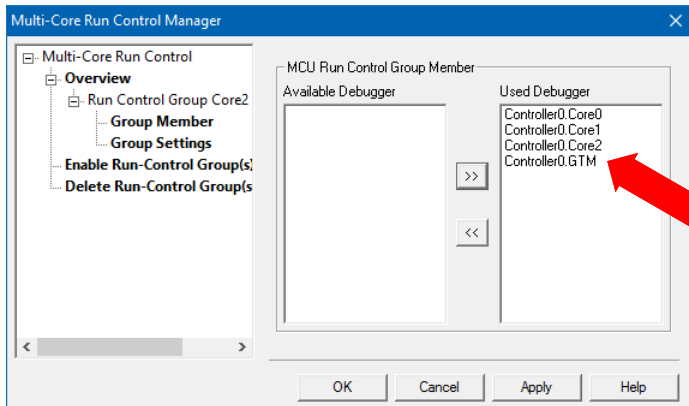
Index	Tick	Address	Interpret	Source	Function	Submodule	
1:	26034	0xC000122A	MOVH d15, 0xF012	GTM_AFD0_CH1_BUF_ACC = GTM_AF...	handle_gtm_tasks		
1:	26035	0xC000122E	ADDI d15, d15, -0x7F70		handle_gtm_tasks		
1:	26036	0xC0001232	MOVH d2, 0xF012		handle_gtm_tasks		
1:	26037	0xC0001236	ADDI d2, d2, -0x7F80		handle_gtm_tasks		
1:	26038	0xC000123A	MOV.A a15, d2		handle_gtm_tasks		
1:	26039	982.07 us	0xF01380C8	MOVL CTRG, 0x000001		MCS0:Ch2	
1:	26040	982.09 us	0xF01380CC	ADDL R0, 0x000001		MCS0:Ch2	
1:	26041	982.11 us	0xF01380D0	ORL STRG, 0x000002		MCS0:Ch2	
1:	26042	982.13 us	0xF01380D4	ORL STA, 0x000002		MCS0:Ch2	
1:	26043	981.99 us	0xC000123C	LD.W d2, [a15] 0x0	handle_gtm_tasks		
1:	26044	982.19 us	0xC000123E	Interrupt: MOV.A a15, d15	tasks		
1:	26045	982.21 us	0xF01380E8	TRG d15, 0x000001		MCS0:Ch2	
1:	26046	982.21 us	-0.05 us between #26044 and #26046				
1:	26047	0xC00040C2	MOVH.A a15, 0xD000				
1:	26048	0xC00040C6	LEA a15, [a15] 0x134				
1:	26049	982.26 us	0xC00040CA	LD.A a14, [a15+] 0x4			
1:	26050	0xC00040CC	LD.W d4, [a15]				
1:	26051	982.28 us	0xC00040CE	CALLI a14			
1:	26052	982.29 us	0xF01380C0	MOVL R1, 0x000001		MCS0:Ch2	
1:	26053	982.31 us		Trigger			
1:	26054	982.30 us	0xC0000478	MOV.AA a14, a10	{	isr_gtm_mcs0_ch2	
1:	26055	0xC000047A	SUB.A a10, 0x8			isr_gtm_mcs0_ch2	
1:	26056	0xC000047C	ST.W [a14] -0x4, d4			isr_gtm_mcs0_ch2	
1:	26057	982.31 us	0xC0000480	MOVH d15, 0xD000	if (done)	isr_gtm_mcs0_ch2	
1:	26058	0xC0000484	ADDI d15, d15, 0x0			isr_gtm_mcs0_ch2	
1:	26059	982.32 us	0xF01380C4	WURM R1, STRG, 0x0001		MCS0:Ch2	
1:	26060	982.32 us	0xC0000488	MOV.A a15, d15		isr_gtm_mcs0_ch2	
1:	26061	982.33 us	0xC000048A	LD.W d15, [a15] 0x0		isr_gtm_mcs0_ch2	
1:	26062	982.34 us	0xC000048C	TRG d15, 0x000001		isr_gtm_mcs0_ch2	

Run-Mode Debugging – Current State

Based on GTM-IP Debug feature set

- Suspend hole GTM
- Synchronous stop/start together with main cores
 - Configurable
 - Independ
 - Selected cores
 - All cores

- Access to GTM registers
 - Core registers (Registers of MCSx_CHz)



MCS: 1 Channel: 1

GPR	CTRG / STRG	Name	Value
R0	0x00000002	PC	0xF7CA0070
R1	0x00179FCE	CTRL	0x00000401
R2	0x00000002	SP_CNT	0x00000000
R3	0x00000000	SAT	<input checked="" type="checkbox"/>
R4	0x00000000	CWT	<input type="checkbox"/>
R5	0x00000000	CAT	<input type="checkbox"/>
R6	0x00000000	N	<input type="checkbox"/>
R7	0x00000000	V	<input type="checkbox"/>
CTRG	0x00000000	Z	<input type="checkbox"/>
STRG	0x00000000	CY	<input type="checkbox"/>
		MCA	<input type="checkbox"/>
		ERR	<input type="checkbox"/>
		IRQ	<input type="checkbox"/>
		EN	<input checked="" type="checkbox"/>
		ACB	0x00000001
		ACB4	<input type="checkbox"/>
		ACB3	<input type="checkbox"/>
		ACB2	<input type="checkbox"/>
		ACB1	<input type="checkbox"/>
		ACB0	<input checked="" type="checkbox"/>

- Module registers

SFR View1

Name	Value	Bit field	Value
GTM_TIM0_INP_VAL	0x00000000	TIM_IN	0x00
		F_IN	0x00
		F_OUT	0x00
GTM_TIM0_CH0_CNT	0x00684...	CNT	0x684D29
GTM_MCS0_CH0_PC	0x00000044	PC	0x0044
GTM_ARU_DATA_L	0x00000000	DATA	0x00000000
GTM_ARU_DATA_H	0x00000000	DATA	0x00000000
GTM_ARU_DBG_DATA0_H	0x01123456	DATA	0x01123456
GTM_ARU_DBG_DATA0_L	0x01C01FCE	DATA	0x01C01FCE

Outlook GTM Run-Mode Debugging

- Integration of Debug functionality into GTM-IP
 - Expected features for MCS code debugging
 - Breakpoints
 - Single stepping
- Availability of C-Compilers (e.g. HighTec)
 - C-Source code debugging

The screenshot displays the UDE 4.10 IDE interface for debugging GTM code. The main window shows the C source code for `main_mcs00.c`, which includes a `main` function and a `main_ch1` function. Comments indicate periodic counter increments and signal generation. The left pane shows the Target Manager with a tree view of the target hardware, including Controller0, Core0-5, GTM, ED, Memory, miniMcs, and McdsTrace. The right pane shows a Time / Value Chart with a periodic waveform and a Peripheral Registers table.

Name	Value	Bit field	Value
GTM_TOM0_CH0_STAT	0x00000000	OL	0x0
GTM_FIFO0_CH0_STATUS	0x00000008	UP_WM	0x1 (Fill level >= upper waterm...)
		LOW_WM	0x0 (Fill level > lower waterm...)
		FULL	0x0 (Fill level < FIFO[1]_CH[0]...)
		EMPTY	0x0 (Fill level > 0)
GTM_FIFO0_CH1_STATUS	0x00000008	UP_WM	0x1 (Fill level >= upper waterm...)
		LOW_WM	0x0 (Fill level > lower waterm...)
		FULL	0x0 (Fill level < FIFO[1]_CH[1]...)
		EMPTY	0x0 (Fill level > 0)
GTM_BRC_SRC_0_ADDR	0x000001FE	BRC_MD...	0x0 (Consistency Mode (OD...)
		ADDR	0x1FE
GTM_BRC_SRC_0_DEST	0x00000000	BRC_MD...	0x0 (Consistency Mode (OD...)
		ADDR	0x1FE
GTM_BRC_SRC_1_ADDR	0x000001FE	BRC_MD...	0x0 (Consistency Mode (OD...)
		ADDR	0x1FE
GTM_BRC_SRC_1_DEST	0x00000000	BRC_MD...	0x0 (Consistency Mode (OD...)
		ADDR	0x1FE

The Messages pane at the bottom shows the following log entries:

I...	Type	Time	Source	Message
74	Info	13:05:43...	Core0::UDEMemtool	DF1: Sector table updated
75	Success	13:05:43...	Core0::UDEDebu...	Connection to TC39xA target established: TriCore (
76	Success	13:05:43...	GTM::UDEDebugS...	Connection to TC39xA target established: GTM. ID:
77	Warning	13:05:45...	GTM::ArchSrv	unable to reread the instruction pointer while set



Leading Edge Solution for Debugging and Test

Matthias Noack

PLS Development Tools
Technologiepark
02991 Lauta
Germany

Phone +49 - 35722 - 384 - 0

Fax +49 - 35722 - 384 - 69

info@pls-mc.com
www.pls-mc.com