

MDSTECH · DT+
SW TESTING SOLUTION



DT+Trace는 CPU, RTOS에 상관없이 다양한 개발 환경에서 커버리지 측정, 코드 디버깅, 성능 측정이 가능한 통합 분석 솔루션입니다.

제품 사양

- JTAG memory dump, Data writing, Ethernet, GPIO 등 다양한 인터페이스 지원
- C/C++, C#, JAVA, Python 지원

동작 및 분석 원리

- 소스코드 내 함수의 입·출구, 분기 등 블록 단위로 필요한 곳에 자동으로 Test Point 삽입
- 기존 개발 환경에서 컴파일 진행
- 타깃 실행 후 결과 값 수집

Test Point (Macro) 입력



소스코드 자동 삽입

기존 개발환경 사용



드라이버 추가 후 빌드

타깃의 움직임을 Trace



프로그램 수행 결과 수집



Data writing / Ethernet / JTAG / UART / CAN



ASYNC / GPIO(4/2/1 pin)



USB 3.0



DT+Trace



DT+FS

Coverage 분석 (코드 실행률 측정)

- 소프트웨어의 품질 향상을 위해 Statement, Branch, MC/DC 및 Call coverage 분석을 지원합니다.

Statement / Branch Coverage

Function	Source	if setting matrix	Normal	if/while ID	Execution	Statements	Statements/Function	Number of Ex...	Passed Ex...	Branch/Function
func01	main.c	1	1	1	1	100.00%	100.00%	1	1	100.00%
func02	main.c	2	2	2	2	100.00%	100.00%	2	2	100.00%
func03	main.c	3	3	3	3	100.00%	100.00%	3	3	100.00%
func04	main.c	4	4	4	4	100.00%	100.00%	4	4	100.00%
func05	main.c	5	5	5	5	100.00%	100.00%	5	5	100.00%
func06	main.c	6	6	6	6	100.00%	100.00%	6	6	100.00%
func07	main.c	7	7	7	7	100.00%	100.00%	7	7	100.00%
func08	main.c	8	8	8	8	100.00%	100.00%	8	8	100.00%
func09	main.c	9	9	9	9	100.00%	100.00%	9	9	100.00%
func10	main.c	10	10	10	10	100.00%	100.00%	10	10	100.00%
func11	main.c	11	11	11	11	100.00%	100.00%	11	11	100.00%
func12	main.c	12	12	12	12	100.00%	100.00%	12	12	100.00%
func13	main.c	13	13	13	13	100.00%	100.00%	13	13	100.00%
func14	main.c	14	14	14	14	100.00%	100.00%	14	14	100.00%
func15	main.c	15	15	15	15	100.00%	100.00%	15	15	100.00%
func16	main.c	16	16	16	16	100.00%	100.00%	16	16	100.00%
func17	main.c	17	17	17	17	100.00%	100.00%	17	17	100.00%
func18	main.c	18	18	18	18	100.00%	100.00%	18	18	100.00%
func19	main.c	19	19	19	19	100.00%	100.00%	19	19	100.00%
func20	main.c	20	20	20	20	100.00%	100.00%	20	20	100.00%
func21	main.c	21	21	21	21	100.00%	100.00%	21	21	100.00%
func22	main.c	22	22	22	22	100.00%	100.00%	22	22	100.00%
func23	main.c	23	23	23	23	100.00%	100.00%	23	23	100.00%
func24	main.c	24	24	24	24	100.00%	100.00%	24	24	100.00%
func25	main.c	25	25	25	25	100.00%	100.00%	25	25	100.00%
func26	main.c	26	26	26	26	100.00%	100.00%	26	26	100.00%
func27	main.c	27	27	27	27	100.00%	100.00%	27	27	100.00%
func28	main.c	28	28	28	28	100.00%	100.00%	28	28	100.00%
func29	main.c	29	29	29	29	100.00%	100.00%	29	29	100.00%
func30	main.c	30	30	30	30	100.00%	100.00%	30	30	100.00%
func31	main.c	31	31	31	31	100.00%	100.00%	31	31	100.00%
func32	main.c	32	32	32	32	100.00%	100.00%	32	32	100.00%
func33	main.c	33	33	33	33	100.00%	100.00%	33	33	100.00%
func34	main.c	34	34	34	34	100.00%	100.00%	34	34	100.00%
func35	main.c	35	35	35	35	100.00%	100.00%	35	35	100.00%
func36	main.c	36	36	36	36	100.00%	100.00%	36	36	100.00%
func37	main.c	37	37	37	37	100.00%	100.00%	37	37	100.00%
func38	main.c	38	38	38	38	100.00%	100.00%	38	38	100.00%
func39	main.c	39	39	39	39	100.00%	100.00%	39	39	100.00%
func40	main.c	40	40	40	40	100.00%	100.00%	40	40	100.00%
func41	main.c	41	41	41	41	100.00%	100.00%	41	41	100.00%
func42	main.c	42	42	42	42	100.00%	100.00%	42	42	100.00%
func43	main.c	43	43	43	43	100.00%	100.00%	43	43	100.00%
func44	main.c	44	44	44	44	100.00%	100.00%	44	44	100.00%
func45	main.c	45	45	45	45	100.00%	100.00%	45	45	100.00%
func46	main.c	46	46	46	46	100.00%	100.00%	46	46	100.00%
func47	main.c	47	47	47	47	100.00%	100.00%	47	47	100.00%
func48	main.c	48	48	48	48	100.00%	100.00%	48	48	100.00%
func49	main.c	49	49	49	49	100.00%	100.00%	49	49	100.00%
func50	main.c	50	50	50	50	100.00%	100.00%	50	50	100.00%
func51	main.c	51	51	51	51	100.00%	100.00%	51	51	100.00%
func52	main.c	52	52	52	52	100.00%	100.00%	52	52	100.00%
func53	main.c	53	53	53	53	100.00%	100.00%	53	53	100.00%
func54	main.c	54	54	54	54	100.00%	100.00%	54	54	100.00%
func55	main.c	55	55	55	55	100.00%	100.00%	55	55	100.00%
func56	main.c	56	56	56	56	100.00%	100.00%	56	56	100.00%
func57	main.c	57	57	57	57	100.00%	100.00%	57	57	100.00%
func58	main.c	58	58	58	58	100.00%	100.00%	58	58	100.00%
func59	main.c	59	59	59	59	100.00%	100.00%	59	59	100.00%
func60	main.c	60	60	60	60	100.00%	100.00%	60	60	100.00%
func61	main.c	61	61	61	61	100.00%	100.00%	61	61	100.00%
func62	main.c	62	62	62	62	100.00%	100.00%	62	62	100.00%
func63	main.c	63	63	63	63	100.00%	100.00%	63	63	100.00%
func64	main.c	64	64	64	64	100.00%	100.00%	64	64	100.00%
func65	main.c	65	65	65	65	100.00%	100.00%	65	65	100.00%
func66	main.c	66	66	66	66	100.00%	100.00%	66	66	100.00%
func67	main.c	67	67	67	67	100.00%	100.00%	67	67	100.00%
func68	main.c	68	68	68	68	100.00%	100.00%	68	68	100.00%
func69	main.c	69	69	69	69	100.00%	100.00%	69	69	100.00%
func70	main.c	70	70	70	70	100.00%	100.00%	70	70	100.00%
func71	main.c	71	71	71	71	100.00%	100.00%	71	71	100.00%
func72	main.c	72	72	72	72	100.00%	100.00%	72	72	100.00%
func73	main.c	73	73	73	73	100.00%	100.00%	73	73	100.00%
func74	main.c	74	74	74	74	100.00%	100.00%	74	74	100.00%
func75	main.c	75	75	75	75	100.00%	100.00%	75	75	100.00%
func76	main.c	76	76	76	76	100.00%	100.00%	76	76	100.00%
func77	main.c	77	77	77	77	100.00%	100.00%	77	77	100.00%
func78	main.c	78	78	78	78	100.00%	100.00%	78	78	100.00%
func79	main.c	79	79	79	79	100.00%	100.00%	79	79	100.00%
func80	main.c	80	80	80	80	100.00%	100.00%	80	80	100.00%
func81	main.c	81	81	81	81	100.00%	100.00%	81	81	100.00%
func82	main.c	82	82	82	82	100.00%	100.00%	82	82	100.00%
func83	main.c	83	83	83	83	100.00%	100.00%	83	83	100.00%
func84	main.c	84	84	84	84	100.00%	100.00%	84	84	100.00%
func85	main.c	85	85	85	85	100.00%	100.00%	85	85	100.00%
func86	main.c	86	86	86	86	100.00%	100.00%	86	86	100.00%
func87	main.c	87	87	87	87	100.00%	100.00%	87	87	100.00%
func88	main.c	88	88	88	88	100.00%	100.00%	88	88	100.00%
func89	main.c	89	89	89	89	100.00%	100.00%	89	89	100.00%
func90	main.c	90	90	90	90	100.00%	100.00%	90	90	100.00%
func91	main.c	91	91	91	91	100.00%	100.00%	91	91	100.00%
func92	main.c	92	92	92	92	100.00%	100.00%	92	92	100.00%
func93	main.c	93	93	93	93	100.00%	100.00%	93	93	100.00%
func94	main.c	94	94	94	94	100.00%	100.00%	94	94	100.00%
func95	main.c	95	95	95	95	100.00%	100.00%	95	95	100.00%
func96	main.c	96	96	96	96	100.00%	100.00%	96	96	100.00%
func97	main.c	97	97	97	97	100.00%	100.00%	97	97	100.00%
func98	main.c	98	98	98	98	100.00%	100.00%	98	98	100.00%
func99	main.c	99	99	99	99	100.00%	100.00%	99	99	100.00%
func100	main.c	100	100	100	100	100.00%	100.00%	100	100	100.00%

DT+Trace

Function Coverage

Function Coverage Report : Repo_201225_141003

Source	Function	Execute...	Coverage
DisplayRenderer.c	7	5	71.43%
Task_Buzzer.c	1	1	100.00%
Task_Display.c	1	1	100.00%
Task_Hardware.c	11	8	72.73%
Task_LED.c	1	1	100.00%
Task_RGB-LED.c	1	1	100.00%
tasks.c	2	0	0.00%

DT+Trace

MC/DC Coverage

No.	A	B	C	D	Result
1	T	F	T	F	True
2	T	F	T	F	False
3	T	F	F	T	False
4	T	F	F	T	False
5	F	T	T	F	False

DT+Trace

Function Call Coverage

Task_Buzzer	Task_Buzzer.c	5	4	80.00%
prochardware_LED_AD	Task_Hardware.c	26	25	96.15%
OfRender Init	DisplayRenderer.c	4	4	100.00%
OfRender_SetDisplayString	DisplayRenderer.c	3	3	100.00%
OfRender_BeginScene	DisplayRenderer.c	2	2	100.00%
OfRender_EndScene	DisplayRenderer.c	1	1	100.00%
OfRender_DrawDeviceFor...	DisplayRenderer.c	1	1	100.00%
OfRender_Present	DisplayRenderer.c	1	1	100.00%
Task_Display	Task_Display.c	10	10	100.00%
initHardware	Task_Hardware.c	5	5	100.00%
Task_Hardware	Task_Hardware.c	4	4	100.00%
initHardware_LED_AD	Task_Hardware.c	8	8	100.00%
initHardware_PWM_DSP	Task_Hardware.c	29	29	100.00%
Task_LED	Task_LED.c	5	5	100.00%

DT+Trace

setBuzzerPact	Task_Hardware.c	2	1	50.00%
OfRender Init	DisplayRenderer.c	1	1	100.00%
OfRender_SetDisplayString	DisplayRenderer.c	1	1	100.00%
OfRender_BeginScene	DisplayRenderer.c	2	2	100.00%
OfRender_EndScene	DisplayRenderer.c	2	2	100.00%
OfRender_DrawDeviceFor...	DisplayRenderer.c	6	6	100.00%
OfRender_Present	DisplayRenderer.c	1	1	100.00%
getDistance	Task_Hardware.c	3	3	100.00%
getDMSVoltage	Task_Hardware.c	1	1	100.00%
setLEDState	Task_Hardware.c	2	2	100.00%
getColorVolume	Task_Hardware.c	1	1	100.00%
setRGBLEDColor	Task_Hardware.c	2	2	100.00%

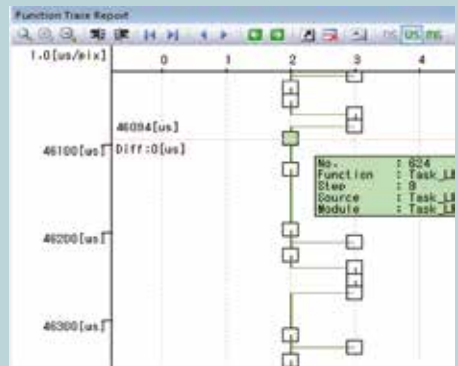
Debugging (코드 디버깅)

- 코드 흐름, 변수 값, 함수 호출 관계 등을 분석할 수 있는 기능을 제공합니다.
- 각 CPU, Task, Function 단위로 필터링하여 코드 흐름을 파악할 수 있습니다.

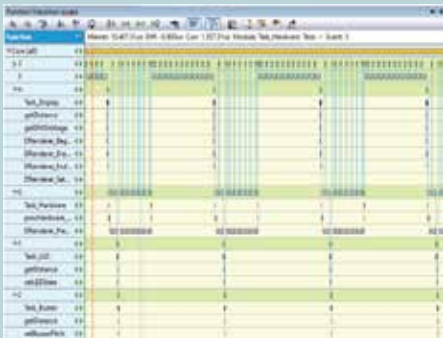
코드 구조 트리 분석



Function Trace Report



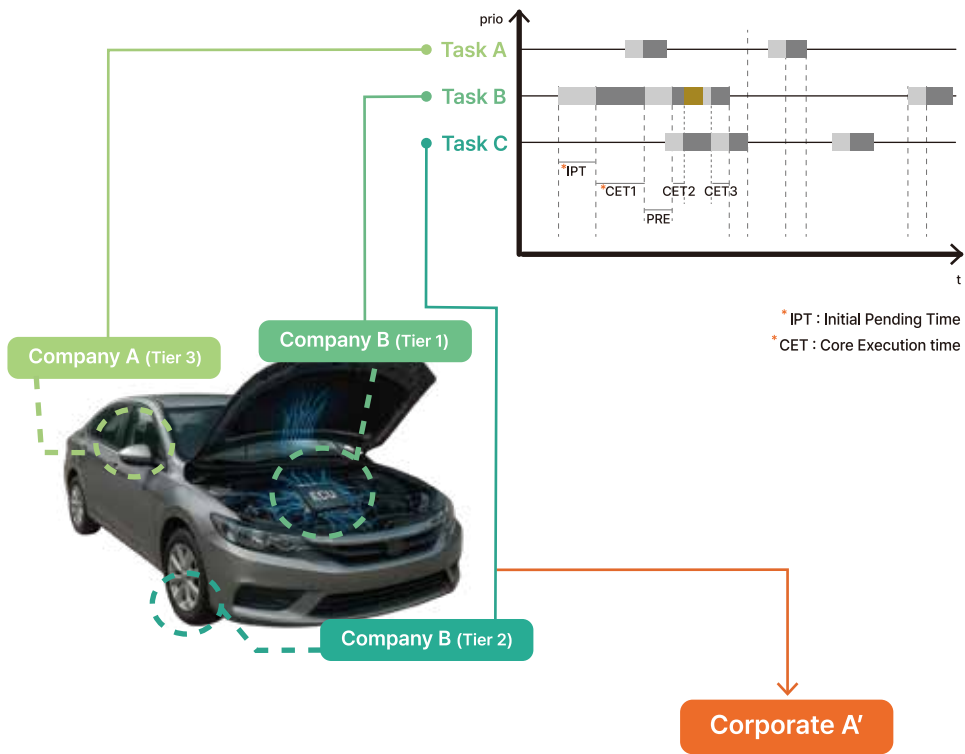
Function Transition Scope



변수 값 모니터링



Automotive



Name by OEM(automotive corporate)



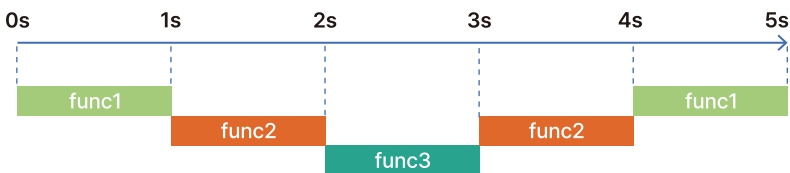
Automotive

완성차 제조 업체(OEM) 의 요구사항 만족을 위한 ECU 시간 분석

- Automotive 분야에서는 안전 검증을 위해 여러 가지 테스트 항목을 충족하는 것이 중요
- OEM에서 요구하는 TASK 수행 시간 측정 및 검증을 위해 사용
- 자원사용량 검증을 위한 CPU 사용량 측정

Benefit

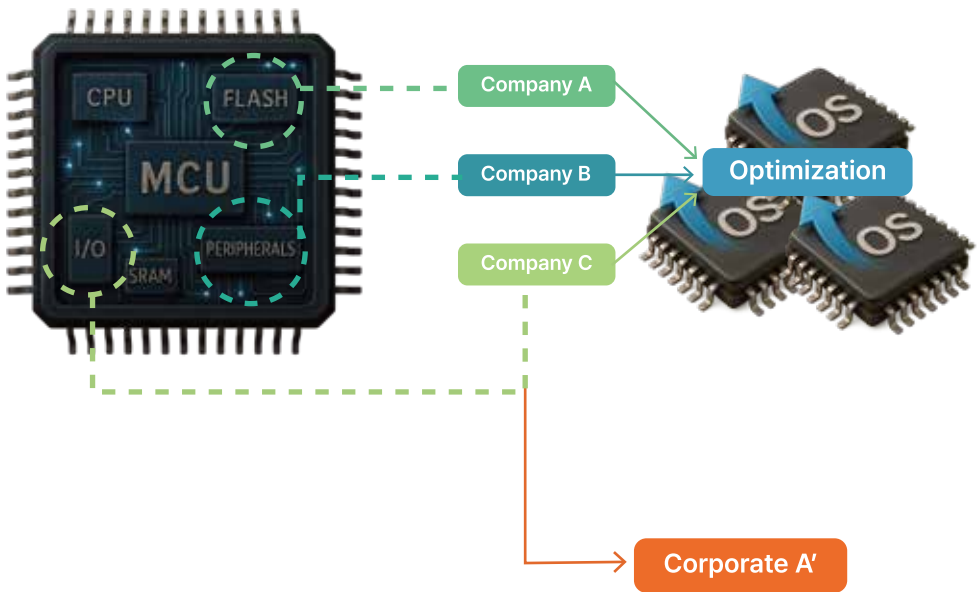
- CPU, 컴파일러 등에 의존성이 없어 다양한 프로젝트에 동시 사용 가능
- 타 동적 분석 툴 대비 빠르게 처리되는 탐침 코드로 발생하는 오버헤드 최소화
- 순수 수행 시간, 총 수행 시간 등의 항목들을 사용자의 선택에 맞게 데이터화
- 동작 흐름 및 시간 수치 UI 제공으로 사용자 편의성 증시
- 다양한 보드 설계에 맞추어 GPIO, CAN, UART 등의 다양한 방식으로 연결 가능
- ISO 26262 대응 - 기능안전 표준 준수, ASIL 레벨별 테스트 커버리지 확보 가능



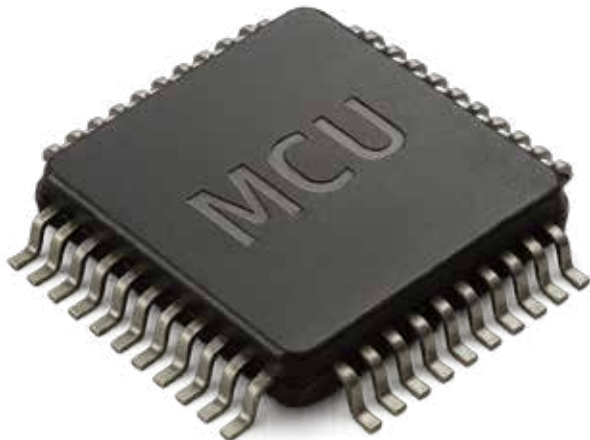
	Total time (FuncIn-FuncOut)	Total time (Flat mode)
Func1	5sec	2sec
Func2	3sec	2sec
Func3	1sec	1sec

Execution Time Report (FuncIn-FuncOut ↔ Flat mode)

SoC / Industries



Name by ODM(Original Design Manufaturing)



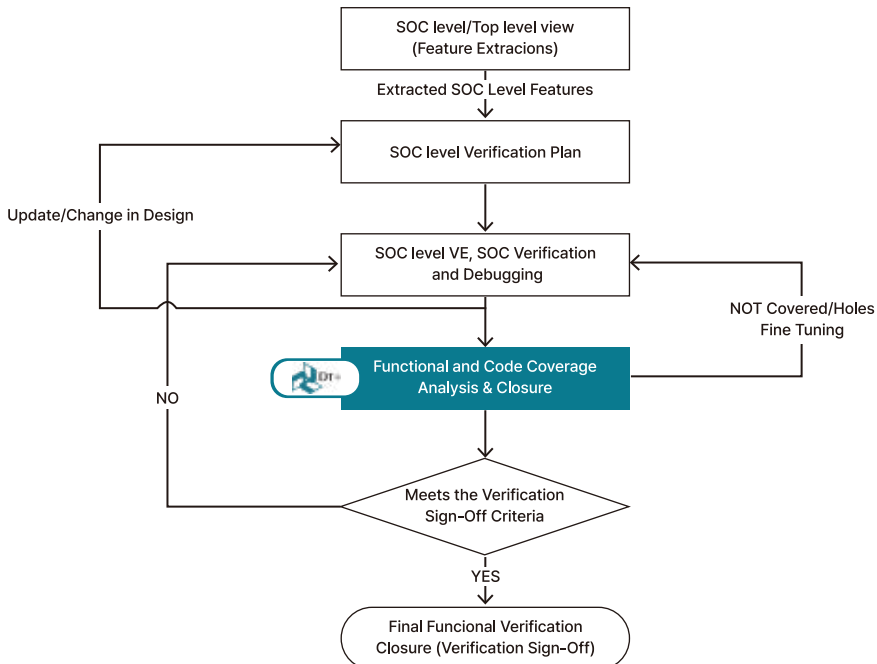
SoC / Industries

컨트롤러 및 자체 개발 프로그램의 최적화

- 기존 개발 코드 중 레거시 코드를 확인하여 함수 flow 및 performance data 분석
- 컨트롤러의 SW 품질향상 목적 statement와 branch 커버리지를 측정 및 검증
- Sudden Power-off test 대응이 가능하여 power가 끊기는 순간까지 로그를 확인

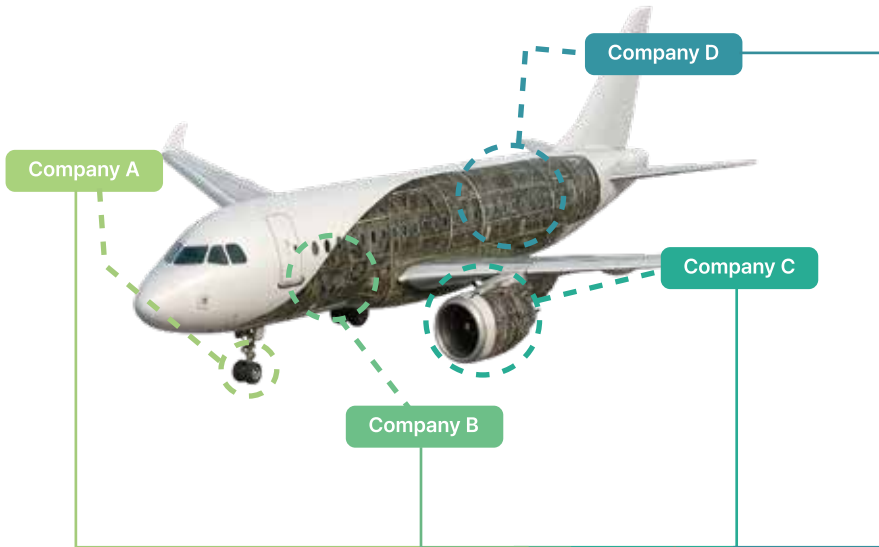
Benefit

- Benchmark로는 불가능한 커버리지 측정, 함수, Task, 경로 별 분석 가능
- 설계 상 trace 데이터를 받아오지 못하는 경우에도 오버헤드 최소화한 trace 가능
- 리소스를 과도하게 사용하고 있는 특정 TASK나 함수를 Ratio Scope로 손쉽게 확인
- CPU 외 Direct Memory Access, 외부 메모리 접근에 의한 IP 간섭 시각화
- 처리 시간/성능/오버런 여부에 대한 수치 기반 리포트 생성하여 제공



<SoC-level functional verification flowchart>

Aerosapce & Avionics



Corporate A'

Name by mission-critical/aerospace industry corporate



Aerosapce & Avionics

대규모 체계 개발 프로젝트

- 1개의 체계개발 사업 내 다수의 모듈을 협력업체에서 각각 개발하는 구조

Benefit

- CPU 및 컴파일러의 의존성이 없는 DT+ 의 강점으로 동적 시험 시 제약사항이 없음
- 직관적인 인터페이스 및 블록 단위 코드 커버리지는 동작 속도가 느린 개발 환경에서도 도입
- 인증 기관 및 사업 주관사에서 다수의 제어기에 대해 개발 코드에 대한 커버리지 100% 달성 유무 확인
- 무기체계 소프트웨어 개발 및 관리 매뉴얼의 동적시험 대상 언어(C/C++, C#, JAVA) 지원

Coverage Report

- DT+ SW를 통해 리포트 출력

* MCDC, Call coverage는 DT+FS 버전에서 제공



Coverage data source unit

Published on: 2025-06-02 16:07:32

Project: AnalogBoxDemo.dmxj

Profile: Prof_250512_170733

Report: Resp_250512_170801

Execution date: 2025-05-12 17:08:01

Module	Source	Effective TP	Execution (Calculation target)	Number of Branch Router/TP	Passed Branch Router/TP	Statement(%)	Branch(%)
All Module	-	440	254	298	47	58	16
1_Main	main.c	8	0	2	0	0.0	0.0
2_Source	buzzer.c	36	10	23	0	27.8	0.0
	test_display.c	39	27	25	11	69.2	44.0
	test_driver.c	22	15	4	3	68.2	75.0
	rgbled.c	46	29	32	16	63.0	50.0
	sensor.c	25	8	X	X	32.0	0.0
	switch1.c	21	12	18	0	57.1	0.0
	switch2.c	13	7	9	0	53.8	0.0



ISO



(주)MDS테크

경기도 성남시 분당구 대왕판교로 644번길 49 DTC타워 9층

T. 031-627-3088

E. DT@mdstech.co.kr

H. www.dtplus.ai

 MDSTECH