

1. WMX1와 WMX2 사양 주요 비교사항

NO	항목	WMX1	WMX2	상세자료
1	통신연결 수요시간	2~4s	< 120ms ~ 2s ^(1*)	
2	Hotconnect 수요시간	2~4s	< 120ms	
3	지원 Topology	Line, Star	Line, Star, Ring	
4	Ring Hotconnect	불가능	가능	별첨 1
5	Master card (PFC-200)	필요	불필요	
6	통신 사이클	0.5ms, 1ms, 2ms, 4ms	0.125ms, 0.25ms, 0.5ms, 1ms, 2ms, 4ms	
7	1ms 통신 사양	64축, 96개 slave	64축, 256개 slave	
8	Packet Loss Timeout 구분	불가능	가능	별첨 2
9	시스템 안정성 분석	없음	있음	별첨 3
10	EtherCAT과 Simu 엔진 스위칭	엔진 재로딩으로 가능 (엔지 타입 지정)	통신 재접속으로 가능 (파일로 설정 변경)	
11	Virtual Axis 기능	없음	있음	별첨 4
12	Virtual Serial 기능	있음	있음	

⇒ WMX2는 사용자 입장에서의 Trouble Shooting 능력이 탁월 합니다.

2. WMX1와 WMX2의 개선사항 비교

NO	항목	WMX1	WMX2	상세자료
1	Hotconnect로 신규 slave 추가	불가능	가능	
2	특수 slave의 init 파일 강제 지정	필요	불필요 (Alias정의로 구분 가능)	
3	Alias 번호를 축번호로 지정	불가능	가능	
4	축 번호 및 I/O주소 지정	불가능	가능	
5	통신 안정성 분석	없음	가능	별첨 5
6	Topology 뷰어	없음	가능	
7	PDO Mapping 정보 표시	불가능	가능	

3. WMX2 변경사항

NO	항목	WMX1	WMX2	상세자료
1	개발 언어	C	C++/C#	별첨 6
2	리소스 소비	중	저	
3	유틸리티 제공	2개	5개	별첨 7
4	RTX버전	RTX 2011	RTX 2016(최신)	

[별첨2]

1. Packet Loss와 Packet Timeout 구분

I. Packet Loss ?

Packet Loss는 Master에서 송신 된 데이터가 케이블과 Slave를 통과할 경우 노이즈 때문에 실제 물리적으로 손실 된 데이터.

II. Packet Timeout ?

Packet Timeout는 Master에서 송신 된 데이터가 케이블 또는 Slave의 노이즈 때문에 Master에 늦게 도착.

III. 장점

Packet Loss는 테스트를 통해서 쉽게 노이즈가 가장 많이 타는 Slave를 찾을 수 있음.

Packet Timeout는 WMX2 유틸리티에서 제공하는 기능을 통해서 노이즈 타는 범위를 찾을 수 있음.

[별첨3]

1. 시스템 안정성 분석 데이터

Statistics
Slaves

Comm Period: 1000 us	Cycle Process Time	CE Process Time	
CycleCounter: 34035	< 20%: 100.00%	< 20%: 100.00%	<input type="button" value="Ce"/>
Over Cycle: 0	20% - 40%: 0.00%	20% - 40%: 0.00%	<input type="button" value="Comm"/>
Interrupt Interval	40% - 60%: 0.00%	40% - 60%: 0.00%	Total Axes Num: 3
< 20%: 0.00%	60% - 80%: 0.00%	60% - 80%: 0.00%	Total Tx Pdo Size (Byte): 45
20% - 50%: 0.00%	80% - 100%: 0.00%	80% - 100%: 0.00%	Total Rx Pdo Size (Byte): 57
50% - 80%: 0.00%	>= 100%: 0.00%	>= 100%: 0.00%	Total Output Size (Byte): 0
80% - 95%: 0.06%	PE Command Process Time	PE Feedback Process Time	Total Input Size (Byte): 0
95% - 105%: 99.93%	< 20%: 100.00%	< 20%: 100.00%	Ring Num: 0
105% - 120%: 0.01%	20% - 40%: 0.00%	20% - 40%: 0.00%	
120% - 150%: 0.00%	40% - 60%: 0.00%	40% - 60%: 0.00%	
150% - 180%: 0.00%	60% - 80%: 0.00%	60% - 80%: 0.00%	
>= 180%: 0.00%	80% - 100%: 0.00%	80% - 100%: 0.00%	
Avg Tx Delay (100ns): 350	>= 100%: 0.00%	>= 100%: 0.00%	
Min Tx Delay (100ns): 216	Ref Clock (ns): 173141466	HcState: Idle	
Max TxDelay (100ns): 980	Next DC Time (ns): 173935825	HcAbortCode: None	
PacketLoss: 0 - 0	Diff (ns): 694359	HcErrorCode: 0x00000000	
<input type="button" value="Reset"/>	Min Diff (ns): 415479		
Avg Packet Interval (ns): 999988	Max Diff (ns): 801279		
Min Packet Interval (ns): 896120			
Max Packet Interval (ns): 1101280			
<input type="button" value="Reset"/>			

[별첨4]

1. Virtual Axis와 Axis Index 설정

① Virtual Axis

WMX2에서는 EtherCAT 엔진과 Simulation엔진을 동시에 로딩할 수 있음.
실제 연결된 축에대한 제어만 아닌 가상 축에대한 제어도 동시 가능.

② Axis Index 설정

(1) Alias 번호로 축을 인식할 수 있음. 즉, 서보 축이며 Alias 번호가 중복되어 있지 않을 경우, 설정한 번호로 축이 지정 됨.

(2) UserDef 속성으로 강제적인 축번호 설정 가능.

[별첨5]

1. 통신 안정성 분석 데이터

Network Status ✖

Auto Mode
 Manual Mode

Id	VendorId	ProductCode	Status	Offline	Inaccessible	Case	IF Port0	RE Port0	IF Port1	RE Port1	IF Port2
0	0x0000083	0x000007C	Op	True	True	7,8,9,10	0x0	0x0	0x0	0x46	0x0
1	0x000006F	0x60380004	Op	True	True	0	0x0	0x0	0x0	0x0	0x0
2	0x000006F	0x60380004	Op	True	True	0	0x0	0x0	0x0	0x0	0x0
3	0x000006F	0x60380005	Op	True	True	0	0x0	0x0	0x0	0x0	0x0
4	0x000006F	0x515070A1	Op	True	True	0	0x0	0x0	0x0	0x0	0x0
5	0x000006F	0x515070A1	Op	True	True	0	0x0	0x0	0x0	0x0	0x0
6	0x000006F	0x515070A1	Op	True	True	0	0x0	0x0	0x0	0x0	0x0
7	0x000006F	0x525100A1	Op	True	True	0	0x0	0x0	0x0	0x0	0x0
8	0x0000083	0x00000007	Op	True	True	0	0x0	0x0	0x0	0x0	0x0
9	0x0000083	0x00000007	Op	True	True	0	0x0	0x0	0x0	0x0	0x0
10	0x00001DD	0x00005500	None	True	True		--	--	--	--	--
11	0x00001DD	0x00007062	None	True	True		--	--	--	--	--
12	0x00001DD	0x00006002	None	True	True		--	--	--	--	--
13	0x0000083	0x00000083	None	True	True		--	--	--	--	--
14	0x0000083	0x0000007C	None	True	True		--	--	--	--	--
15	0x0000083	0x00000029	None	True	True		--	--	--	--	--

[별첨6]

1. WMX2 API 변경점

- ① C언어 API를 C++언어 API로 수정.
- ② 부 함수 이름 변경.
- ③ 구조변경 또는 사양 변경 없음.
- ④ CLR 라이브러리 제공 (C#용)

차이점	WMX	WMX2
일부 함수 이름 변경	extMotion2->StartJerkPos()	jerkRatioMotion->StartPos()
C++ 언어 API로 변경	<pre> typedef struct{ long InPosWidth; //In position width long InPos2Width; //In position width 2 long InPos3Width; //In position width 3 long InPos4Width; //In position width 4 long InPos5Width; //In position width 5 short VelocityMonitorSource; //0: Calculate from feedback long PosCompWidth; //Position completed width long DPsetPosCompWidth; //Position completed width f unsigned int DPsetDelay; //Cycles within position cor -WMX_FEEDBACK_PARAM, *PWMX_FEEDBACK_PARAM; </pre>	<pre> class FeedbackParam{ public: FeedbackParam(); long inPosWidth; //In position width long inPosWidth2; //In position width 2 long inPosWidth3; //In position width 3 long inPosWidth4; //In position width 4 long inPosWidth5; //In position width 5 short velocityMonitorSource; //0: Calculate from feedback, long posCompletedWidth; //Position completed width long delayedPosSetPosCompletedWidth; //Position complet unsigned int delayedPosSetTime; //Cycles within position }; </pre>

[별첨7]

1. WMX2 유틸리티

종류	기능
WMX2 Manager	WMX2 통합 관리
General Operator	기본적인 모션 구동을 위한 보조 유틸리티
EcConfigurator	네트워크를 진단하고 관리하는 보조 유틸리티
Profile Analyzer	Axis, Digital I/O의 데이터 측정 및 분석 유틸리티

2. 각 유틸리티 별 상세기능

종류	상세기능 및 효과
WMX2 Manager	<p>① 상세기능</p> <ul style="list-style-type: none"> - 제어기의 정보 확인 - 디바이스 채널 관리 (각 Slave) - 보조 유틸리티 실행상태 확인 - 시스템 정보 저장 - 엔진/통신 상태 모니터링 <p>② 효과</p> <ul style="list-style-type: none"> - 최대 256개 Device를 통합적으로 관리가능 - Root device와 sub device를 설정
General Operator	<p>① 상세기능</p> <ul style="list-style-type: none"> - 모션 상태 제어 - Digital I/O, Analog I/O 출력 - Parameter 설정 <p>② 효과</p> <ul style="list-style-type: none"> - 기본 Motion 구동 및 상태 확인 가능 - Digital 입출력 상태 확인 가능
EcConfigurator	<p>① 상세기능</p>

	<ul style="list-style-type: none"> - 마스터의 상태 관리 - Slave의 정보 확인 - Parameter 관리 기능 - 네트워크 정보 추출 - Packet loss, Hot-connect Status 상태 표시 - Topology view <p>② 효과</p> <ul style="list-style-type: none"> - 기본 Motion 구동 및 상태 확인 가능 - Digital 입출력 상태 확인 가능 - 실제 연결된 Master, Hub, Slave의 Topology 형태를 시각적으로 확인 가능
Profile Analyzer	<p>① 상세기능</p> <ul style="list-style-type: none"> - 일반측정, Trigger 측정 - 파라미터 선택 비교 - 단축 모션 프로파일 2D 차트 제공 - Linear, circular interpolation 경로 시각화 <p>② 효과</p> <ul style="list-style-type: none"> - 다축의 모션 프로파일 측정 및 분석 가능 - 파라미터 별 프로파일 비교 가능