**XGK 프로그램의 XGI 변환**

**Analysis Document**

**LS CONFIDENTIAL & PROPRIETARY**

This work contains valuable, confidential, and proprietary information. Disclosure, use, or reproduction outside of LSIS Inc. is prohibited except as authorized in writing. The laws protect this unpublished work. If you have received this material / document in error, please notify us immediately, and return the original material / document without making any copies.

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 버전 | 날짜 | 수정내용 | 작성자 | 검토자 | 비고 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

빈 페이지입니다.

# 목차

[목차 4](#_Toc317695717)

[개요 6](#_Toc317695718)

[1. 개발 범위 6](#_Toc317695719)

[2. 참조 문서 6](#_Toc317695720)

[3. 주요 용어 6](#_Toc317695721)

[제약사항 7](#_Toc317695722)

[1. 디바이스 종류 별 제약사항 7](#_Toc317695723)

[2. 타이머 해상도 7](#_Toc317695724)

[3. 순차 후입 우선명령 8](#_Toc317695725)

[디바이스의 변환 9](#_Toc317695726)

[1. 디바이스의 매핑 9](#_Toc317695727)

[2. 비트 디바이스의 형태 변환 9](#_Toc317695728)

[3. 삭제된 플래그 경우 9](#_Toc317695729)

[4. 변수 명 변경된 경우 9](#_Toc317695730)

[5. 타입이 변경된 경우 9](#_Toc317695731)

[프로그램의 변경 10](#_Toc317695732)

[1. 접점의 변경 10](#_Toc317695733)

[2. 코일의 변경 10](#_Toc317695734)

[3. 실행 제어 응용 명령어 10](#_Toc317695735)

[4. 비교 응용 명령어(입력 단) 11](#_Toc317695736)

[5. 응용 명령어 11](#_Toc317695737)

[6. 펄스 응용 명령어 12](#_Toc317695738)

[7. 타이머(카운터) 리셋 13](#_Toc317695739)

[8. 암묵적 데이터 처리 명령어 13](#_Toc317695740)

[9. 그룹 명령어 14](#_Toc317695741)

[10. 변환 할 수 없는 명령어 15](#_Toc317695742)

[사용자 명령어 통계 16](#_Toc317695743)

[1. LGD DCS UNIT 16](#_Toc317695744)

[1) 디바이스 형태 별 통계 16](#_Toc317695745)

[2) 디바이스 종류 별 통계 16](#_Toc317695746)

[3) 명령어 사용 통계 16](#_Toc317695747)

[4) 통계 요약 16](#_Toc317695748)

[사용자 요구사항 17](#_Toc317695749)

[1. 디바이스 매핑 17](#_Toc317695750)

[2. 디바이스 형태에 대한 수작업 17](#_Toc317695751)

[3. 타이머 변수의 추적 17](#_Toc317695752)

[4. 암묵적 데이터 처리 명령어 17](#_Toc317695753)

[5. 그룹 명령어의 처리 17](#_Toc317695754)

[6. 지원 하지 않는 명령어 17](#_Toc317695755)

[추가 의견 18](#_Toc317695756)

[1. 지원하지 않는 명령어 추가 개발 18](#_Toc317695757)

[2. 간접 디바이스 18](#_Toc317695758)

[로직 변환 일정 19](#_Toc317695759)

# 개요

기존 작성된 XGK 프로그램을 XGI 형태로 변경하는데 있어서, 제약 사항, 변경 방법, 및 사용자가 취해야 할 후 작업에 대해서 기술한다.

## 개발 범위

## 참조 문서

1. XG5000 소프트웨어
2. XGK / XGB 명령어집
3. XGI / XGR/ XEC 명령어집

## 주요 용어

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

# 제약사항

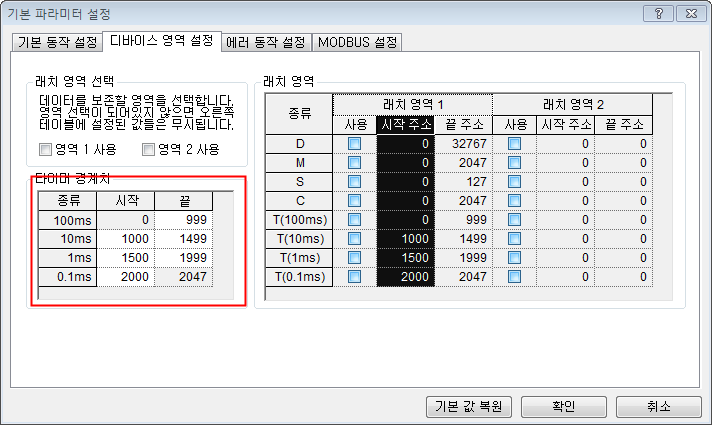
## 디바이스 종류 별 제약사항

XGK 타입은 P, M, L, K의 디바이스 형태뿐 아니라, 디바이스를 이용한 간접 주소 지정 방식을 사용하고 있으나, XGI에대서는 해당 타입을 지원하지 않는다.

|  |  |  |
| --- | --- | --- |
| **디바이스 종류** | **변경방식** | **예** |
| 일반 디바이스 | 사용자 선택에 의한 디바이스 변경 | P, M, L, K, … |
| 인덱스 디바이스 | 배열로 변경, 그러나 제한적임 | P0000[Z00] |
| 간접 지정 디바이스 | 변경 불가 | #P0000 |
| 간접 지정 인덱스 디바이스 | 변경 불가 | #P0000[Z000] |

## 타이머 해상도

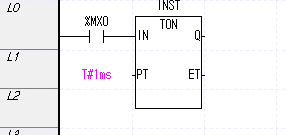
XGK의 경우에는 기본 파라미터를 통해서 타이머의 해상도를 지정할 수 있다. 현재 XGK에서는 다음과 같은 방식으로 타이머에 대한 해상도를 설정한다.



즉 T 디바이스는 사용자가 선택한 파라미터에 따라 0.1ms ~ 100ms의 해상도를 갖는 타이머로 각각 사용할 수 있다. 아래의 그림은 T000이 100ms로 설정하였을 경우 1000ms 이후에 ON이 되는 타이머를 나타낸다.



반면, XGI에서 모든 타이머는 1ms의 해상도를 갖게 되므로 아래와 같이 변경되어야 한다.



그러나, 타이머의 설정치가 상수가 아닌 변수(디바이스)로 설정된 경우에는 타이머에 대한 값을 적절히 변경할 수 없다.

## 순차 후입 우선명령

XGK에서는 S디바이스와 SET/OUT 명령어를 이용하여 순차제어 혹은 후입 우선의 명령어를 제어할 수 있다. 그러나 해당하는 XGI명령어는 없으므로 해당 디바이스는 1:1로 변환될 수 없다.

# 디바이스의 변환

## 디바이스의 매핑

XGK의 디바이스는 XGI와 일치하지 않으므로, XGK 타입의 프로젝트를 XGI로 변환하기 위해서는 사용하고 있는 디바이스에 대한 변환이 필요하다. 디바이스의 변환은 다음과 같은 규칙을 따른다.

|  |  |
| --- | --- |
| **디바이스** | **변경 방식** |
| P | I, Q 디바이스로의 매핑  자동 매핑(IO 파라미터 기준), 수동 매핑 |
| T/C | 사용한 타이머 또는 카운터에 대해서 변수로 선언함. (예, T0 => TON\_INST 변수), 접점/명령어 사용 여부에 따라서 인스턴스 메모리 사용 |
| Z | XGI로 변환할 디바이스 없음, 사용된 디바이스만큼 변수로 선언해서 사용할 수 있음. (배열의 인덱스로 사용) |
| S | XGI로 변환할 디바이스 없음. 예) S00.0 🡪 \_S0.0 |
| D | A 디바이스 영역으로 매핑해서 사용 |

## 비트 디바이스의 형태 변환

코드 및 설명 XGK의 비트 디바이스는 워드+16진수 비트 오프셋으로 표시된다. (예) M1234A  M1234 워드의 10번째 비트) 이를 XGI로 표시하는 경우 %MX19754 형태의 비트로 표시하는 방법과 %MW1234.10의 워드.비트 오프셋 형태로 변환 되도록 선택할 수 있다.

## 삭제된 플래그 경우

결과 창에 삭제된 플래그 명 표시한다. (예, \_AC\_FAIL\_CNT), 플래그명은 프로그램에 표시한다.

P2P, 고속링크, PID 플래그는 기존과 동일하다

## 변수 명 변경된 경우

여러 개의 플래그 명을 하나의 플래그 명으로 표시한다.

(예, \_CYCLE\_TASK\_SCAN0\_CUR => \_CYCLE\_TASK\_SCAN\_TIME[0,0])

## 타입이 변경된 경우

기존에 배열 타입이 없어서 플래그 명을 나열했던 것은 하나로 표시 한다.

(예, \_BASE\_INFO0 => \_BASE\_INFO[0])

# 프로그램의 변경

접점, 코일과 같은 비트 로직을 제외한 모든 명령어의 표현 방식 및 위치에 대한 제약사항의 차이로 인하여, 1:1로 변경될 수는 없으며, 다음과 같은 방식으로 자동 변환 된다.

## 접점의 변경

두 타입의 프로젝트 모두 동일한 형태의 접점을 이용할 수 있으므로, 해당 로직은 그대로 유지된다.

|  |  |  |
| --- | --- | --- |
| **접점 형태** | **변경 전(XGK)** | **변경 후(XGI)** |
| 평상시 열린 접점 | --| |-- | --| |-- |
| 평상시 닫힌 접점 | --| / |-- | --| / |-- |
| 양변환 검출 접점 | --| P |-- | --| P |-- |
| 음변환 검출 접점 | --| N |-- | --| N |-- |
| 반전 접점 | --\*-- | --\*-- |

## 코일의 변경

두 타입의 프로젝트 모두 동일한 형태의 접점을 이용할 수 있으므로, 해당 로직은 그대로 유지 된다.

|  |  |  |
| --- | --- | --- |
| **코일 형태** | **변경 전(XGK)** | **변경 후(XGI)** |
| 코일 | --( )-- | --( )-- |
| 역 코일 | --( / )-- | --( / )-- |
| 셋 코일 | --( S )-- | --( S )-- |
| 리셋 코일 | --( R )-- | --( R )-- |
| 양변환 검출 코일 | --( P )-- | --( P )-- |
| 음변환 검출 코일 | --( N )-- | --( N )-- |

## 실행 제어 응용 명령어

XGK의 명령어 중 루프, 서브루틴, 점프 등의 명령어는 XGI에도 그대로 존재하므로, 1:1로 변환된다, 다만, 화면상에 그려지는 방식은 박스 타입에서 코일 형태로 다소 변경된다.

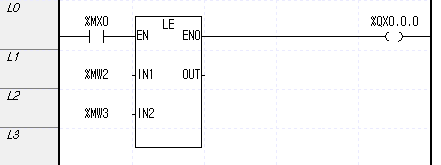
|  |  |  |
| --- | --- | --- |
| **명령어 형태** | **변경 전(XGK)** | **변경 후(XGI)** |
| BREAK | ---------[ **BREAK** ]--| | ---------( **BREAK** )--| |
| CALL | -----[ **CALL** NAME ]--| | -----( **CALL** NAME )--| |
| END | -----------[ **END** ]--| | -----------( **END** )--| |
| FOR | ------[ **FOR** 2000 ]--| | ------( **FOR** 2000 )--| |
| INIT\_DONE | ----[ **ININT\_DONE** ]--| | ----( **ININT\_DONE** )--| |
| JMP | ------[ **JMP NAME** ]--| | ------( **JMP** NAME )--| |
| NEXT | ----------[ **NEXT** ]--| | ----------( **NEXT** )--| |
| RET | -----------[ **RET** ]--| | -----------( **RET** )--| |
| SBRT | -----[ **SBRT** NAME ]--| | -----( **SBRT** NAME )--| |

## 비교 응용 명령어(입력 단)

XGK에서 입력 단에 사용되는 비교 명령어는 XGK의 다른 명령어와 다르게 접점의 위치에 사용된다. 따라서 해당 명령어는 XGI의 비교 명령어와 1:1로 변환되어 사용 가능하다.



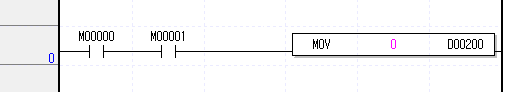
**그림 3 XGK 비교명령**



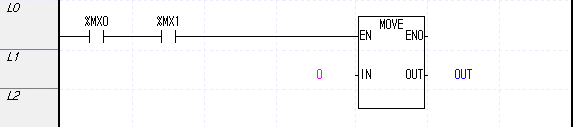
**그림 4 XGI 변환**

## 응용 명령어

XGK에서 출력 에 사용되는 비교 명령어의 경우 XGI의 오른쪽 파워라인 바로 이전으로 위치하게 된다.



**그림 5 XGK 응용명령**



**그림 6 XGI 변환**

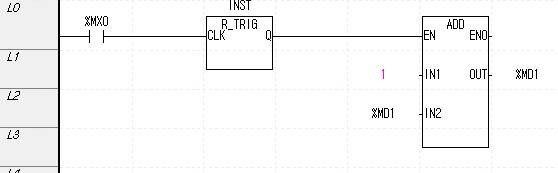
## 펄스 응용 명령어

XGI에서는 XGK에 존재하는 ADDP, MOVP와 같이 상승 에지에서 동작하는 응용명령어가 존재하지 않는다. 따라서 해당 명령어는 다음과 같은 형태로 변경되어야 한다.



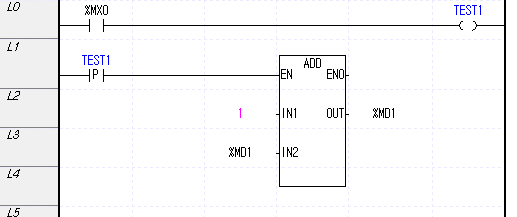
**그림 7 XGK 펄스 응용 명령어**

첫째, R\_TRIG 펑션 블록에 대한 인스턴스를 추가하여 변환한다.



**그림 8 XGI의 R\_TRIG 변환**

둘째, 출력에 임시 코일을 사용하고, 해당 임시 변수를 펄스 입력으로 사용하는 함수로 변경한다.

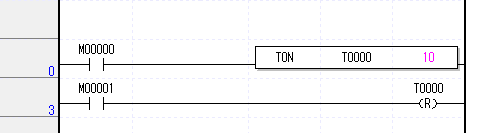


**그림 9 XGI의 더미 출력을 이용한 변환**

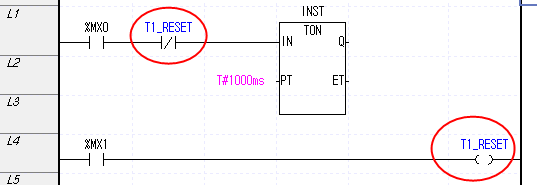
첫 번째 방법의 경우에는, R\_TRIG에 대한 인스턴스 메모리가 계속적으로 늘어나는 단점이 있으며, 두 번째 방법은 인스턴스 증가는 없으나, 기존에 사용자가 작성하였던 프로그램의 모양이 변경되는 단점이 있다. 이 부분은 사용자 옵션으로 선택하는 것도 가능하다.

## 타이머(카운터) 리셋

XGK의 타이머는 T 디바이스를 이용하여 현재 값을 표시하며, 리셋 코일에 사용하여 해당 타이머를 초기화 하게 된다. 그러나, XGI의 타이머는 해당 펑션 블록에 대한 인스턴스의 파라미터 값을 이용하여 초기화 해야 하므로 다음과 같은 방식으로 변환 된다.



**그림 10 XGK 타이머 리셋**

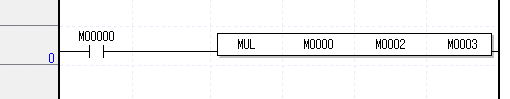


**그림 11 XGI 타이머 변환**

즉, XGK에서는 없었던 접점이 XGI에서는 타이머 리셋을 위한 접점이 추가된다. 카운터도 마찬가지로 리셋 접점이 추가되어야 한다.

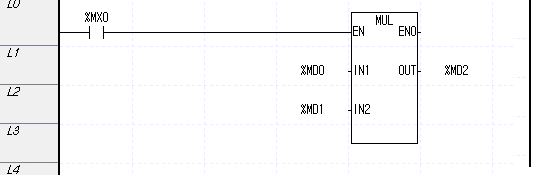
## 암묵적 데이터 처리 명령어

XGK의 경우에는 제한적인 데이터 타입만 제공하여 MUL과 같이 WORD x WORD = DWORD가 되는 형태의 암묵적인 데이터 확장이 존재한다. 예들 들어, 아래 그림에서 XGK에서 M0, M2는 워드 데이터 타입이며, 계산된 결과값인 M3는 M4를 포함한 결과값이 저장되게 된다.



**그림 12 XGK 암묵적 변환 명령**

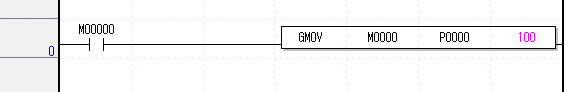
XGI에서는 모든 입력 파라미터와 출력 파라미터가 동일한 데이터 타입만을 취하게 되므로, 아래와 같이 DWROD 타입으로 연산이 되거나, 아니면 INT\_TO\_DINT 함수를 통하여 개별적으로 확장되어야 한다.



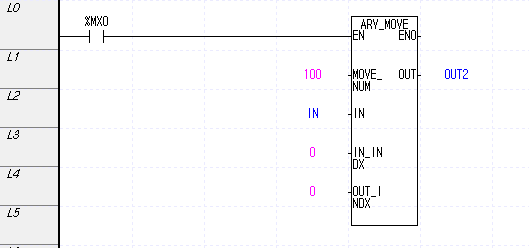
**그림 13 XGI 변환**

## 그룹 명령어

XGK에서는 배열이 제공되지 않는 관계로 데이터를 그룹으로 이동하거나, 초기화 해야 하는 경우를 대비하여 그룹 명령어들이 존재한다. 그러나 해당 명령어가 상수가 아닌 변수에 의해서 데이터의 크기가 정해지는 경우에는 배열을 이용할 수가 없으므로, 제한적으로 변환된다.



**그림 14 XGK 그룹 명령**

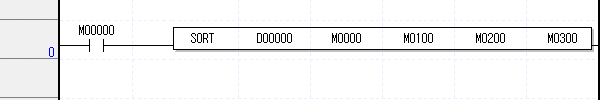


**그림 15 XGI 변환 예**

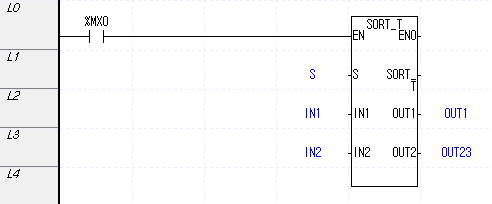
위와 같은 명령어는 약 217개로 명령어 중 15%를 차지한다.

## 변환 할 수 없는 명령어

명령어가 1:1 치환되지 않는 경우에 해당 명령어는 더미 명령어로 변환된다.



**그림 16 XGK 만 존재하는 명령**



**그림 17 XGI 변환**

# 사용자 명령어 통계

## LGD DCS UNIT

### 디바이스 형태 별 통계

|  |  |  |
| --- | --- | --- |
| **디바이스 형태** | **빈도수(회)** | **비율(%)** |
| 일반 디바이스 | 1968 | 100% |
| 인덱스 디바이스 | 0 | 0% |
| 간접 지정 디바이스 | 0 | 0% |
| 간접 지정 인덱스 디바이스 | 0 | 0% |

### 디바이스 종류 별 통계

|  |  |  |
| --- | --- | --- |
| **디바이스 종류** | **빈도수(회)** | **비율(%)** |
| M | 1106 | 56% |
| T | 280 | 14% |
| K | 226 | 11% |
| F | 219 | 11% |
| R | 89 | 4.5% |
| L | 34 | 1.7% |
| U | 14 | 0.7% |

### 명령어 사용 통계

|  |  |  |
| --- | --- | --- |
| **명령어 종류** | **빈도수(회)** | **비율(%)** |
| 접점, 코일 | 1968 | 66% |
| 응용 명령어 | 915 | 30.6% |
| 그룹 명령어 | 100 | 3.35% |

### 통계 요약

본 사용자 프로그램은 일반 디바이스만으로 프로그램 되어 있으며, 66% 이상의 명령어가 접점 코일 명령어로 66%는 1:1 변환이 가능하다. 또한 응용 명령어의 경우에는 변환을 통해서 가능하지만 3.35%에 해당하는 그룹 명령어의 경우에는 사용자가 직접 수동으로 변경하여야 한다.

# 사용자 요구사항

XGK 프로그램에서 XGI로의 변환의 경우 프로그램의 구조적인 차이로 인하여 1:1로 변환되지 않는 경우가 발생하므로 사용자는 다음과 같은 작업을 취해야 한다.

## 디바이스 매핑

XGK에서 지원하는 디바이스의 일부만 XGI에서 지원하므로, XGK에서 사용한 디바이스에 대해서 자동변수를 포함하여 어떠한 디바이스로 매핑할 것인지 설정하여야 한다.

## 디바이스 형태에 대한 수작업

인덱스, 간접 지정, 간접 지정 인덱스에 대한 형태를 지원하지 않으므로, 해당 형태의 디바이스를 사용한 프로그램의 경우에는 배열로 변경하여 프로그램 하거나, 경우에 따라서는 프로그램을 재 구성하여야 한다.

## 타이머 변수의 추적

타이머의 경우 설정 치에 상수가 아닌 변수를 사용한 경우에는 해당 변수를 추적하여 값을 변경하여야 한다. 변환 시 해당 변수를 출력으로 사용한 명령어를 리포트 하므로, 리포트 된 명령어를 따라가면서 확인 또는 수작업으로 변경하여야 한다.

## 암묵적 데이터 처리 명령어

MUL, DIV 등과 같은 암묵적인 데이터 처리 명령어에 대해서는 데이터 타입에 대해서 명시적으로 변환하는 코드를 삽입하거나, 데이터 타입을 변경하여야 한다.

## 그룹 명령어의 처리

XGI에서는 그룹 명령어를 지원하지 않으므로, 사용자는 리포트 된 내용을 이용하여 해당 프로그램을 확인 후 수정하여야 한다.

## 지원 하지 않는 명령어

XGI에서 지원하지 않아 더미 명령어로 변환된 명령어 대해서는 수작업으로 변경하여야 한다.

# 추가 의견

## 지원하지 않는 명령어 추가 개발

XGK에서는 지원하지만 XGI에서는 지원하지 않는 명령어에 대해서 새로운 변환 방법을 고려하는 것 보다 XGI용으로 명령어를 추가하는 것이 장기적인 관점에서 더 유리할 수 있다. 예를 들어, MOV4, MOV8과 같은 비트 처리 명령어와 펄스로 동작하는 응용 명령어가 이에 해당한다. 그러나, 현실적으로 펄스로 동작하는 응용 명령어를 구현하려면, 거의 모든 명령어에 대해서 펄스 형 명령어를 추가 구현해야 한다.

## 간접 디바이스

XGK에서는 인덱스 및 간접 주소 지정 방식을 이용한 데이터 처리 로직을 많이 포함하고 있다. 따라서, 로직의 재 구성없이 1:1 변환으로써는 데이터 처리가 불가능 하므로, XGI에서도 인덱스 디바이스 형태도 고려해 보아야 한다. 예) %MW0[IDX]

# 참고

## XGI/XGK 플래그 비교

|  |  |  |  |
| --- | --- | --- | --- |
| XGI Flag Name | | XGK Flag Name | |
| \_1OFF | BOOL | \_1OFF | BIT |
| \_1ON | BOOL | \_1ON | BIT |
| \_AB\_SD\_ER | BOOL | \_ABSD\_ER | BIT |
| \_AC\_F\_CNT | UINT | \_AC\_F\_CNT | WORD |
|  |  | \_AC\_FAIL\_CNT | DWORD |
| \_ALL\_OFF | BOOL | \_ALL\_OFF | BIT |
| \_ANC\_ERR | WORD | \_ANC\_ERR | WORD |
| \_ANC\_WAR | WORD | \_ANC\_WAR | WORD |
| \_ANNUM\_ER | BOOL | \_ANNUM\_ER | BIT |
| \_ANNUM\_WAR | BOOL | \_ANNUM\_WAR | BIT |
| \_ARY\_IDX\_ERR | BOOL |  |  |
| \_ARY\_IDX\_LER | BOOL |  |  |
| \_BASE\_EMASK\_INFO | DWORD | \_BASE\_EMASK\_INFO | DWORD |
| \_BASE\_INFO\_ER | BOOL | \_BASE\_INFO\_ER | BIT |
| \_BASE\_INFO | ARRAY[0..7] OF WORD | \_BASE\_INFO0 | WORD |
|  |  | \_BASE\_INFO1 | WORD |
|  |  | \_BASE\_INFO2 | WORD |
|  |  | \_BASE\_INFO3 | WORD |
|  |  | \_BASE\_INFO4 | WORD |
|  |  | \_BASE\_INFO5 | WORD |
|  |  | \_BASE\_INFO6 | WORD |
|  |  | \_BASE\_INFO7 | WORD |
| \_BASE\_POWER\_ER | BOOL | \_BASE\_POWER\_ER | BIT |
| \_BASE\_SKIP\_INFO | DWORD | \_BASE\_SKIP\_INFO | DWORD |
| \_BAT\_ER | BOOL | \_BAT\_ER | BIT |
| \_BPRM\_ER | BOOL | \_BPRM\_ER | BIT |
|  |  | \_BUF\_FULL\_CNT | DWORD |
|  |  | \_CA\_CNT | DWORD |
|  |  | \_CA\_ERR\_CNT | DWORD |
|  |  | \_CA\_LIM\_CNT | DWORD |
|  |  | \_CARRY | BIT |
|  |  | \_CB1 | BIT |
|  |  | \_CB2 | BIT |
|  |  | \_CDOVER\_ER | BIT |
| \_CHK\_ANC\_ERR | BOOL | \_CHK\_ANC\_ERR | BIT |
| \_CHK\_ANC\_WAR | BOOL | \_CHK\_ANC\_WAR | BIT |
| \_CMOD\_KEY | BOOL | \_CMOD\_KEY | BIT |
| \_CMOD\_LPADT | BOOL | \_CMOD\_LPADT | BIT |
| \_CMOD\_RLINK | BOOL | \_CMOD\_RLINK | BIT |
| \_CMOD\_RPADT | BOOL | \_CMOD\_RPADT | BIT |
|  |  | \_CMP\_RESULT | WORD |
| \_CNF\_ER | DWORD | \_CNF\_ER | DWORD |
| \_CNF\_WAR | DWORD | \_CNF\_WAR | DWORD |
| \_CODE\_ER | BOOL | \_CODE\_ER | BIT |
| \_CONSTANT\_ER | BOOL | \_COMPILE\_ER | BIT |
|  |  | \_COMPILE\_MODE | BIT |
|  |  | \_CONSTANT\_ER | BIT |
| \_CP\_ER\_PMT | BOOL | \_CP\_ER\_PMT | BIT |
| \_CPPRM\_ER | BOOL | \_CPPRM\_ER | BIT |
|  |  | \_CPU\_ER | BIT |
| \_CPU\_TYPE | WORD | \_CPU\_TYPE | WORD |
| \_CPU\_VER | WORD | \_CPU\_VER | WORD |
| \_CYCLE\_TASK\_SCAN\_WR | BOOL | \_CYCLE\_TASK\_SCAN\_WR | BIT |
| \_CYCLE\_TASK\_SCAN\_TIME | ARRAY[0..31, 0..2] OF WORD | \_CYCLE\_TASK\_SCAN0\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN0\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN0\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN1\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN1\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN1\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN10\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN10\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN10\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN11\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN11\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN11\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN12\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN12\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN12\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN13\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN13\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN13\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN14\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN14\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN14\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN15\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN15\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN15\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN16\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN16\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN16\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN17\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN17\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN17\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN18\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN18\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN18\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN19\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN19\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN19\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN2\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN2\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN2\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN20\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN20\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN20\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN21\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN21\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN21\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN22\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN22\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN22\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN23\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN23\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN23\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN24\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN24\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN24\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN25\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN25\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN25\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN26\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN26\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN26\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN27\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN27\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN27\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN28\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN28\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN28\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN29\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN29\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN29\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN3\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN3\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN3\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN30\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN30\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN30\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN31\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN31\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN31\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN4\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN4\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN4\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN5\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN5\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN5\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN6\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN6\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN6\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN7\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN7\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN7\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN8\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN8\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN8\_MIN | WORD |
|  |  | \_CYCLE\_TASK\_SCAN9\_CUR | WORD |
|  |  | \_CYCLE\_TASK\_SCAN9\_MAX | WORD |
|  |  | \_CYCLE\_TASK\_SCAN9\_MIN | WORD |
|  |  | \_DBCK\_ER | BIT |
| \_DEBUG | BOOL | \_DEBUG | BIT |
| \_EMASK\_ON | BOOL | \_EMASK\_ON | BIT |
|  |  | \_EQU | BIT |
| \_ERR | BOOL |  |  |
|  |  | \_ERR\_HIS\_CNT | DWORD |
|  |  | \_ERR\_STEP | DWORD |
| \_ERROR | BOOL | \_ERROR | BIT |
| \_ESTOP\_ON | BOOL | \_ESTOP\_ON | BIT |
| \_FALS\_NUM | UINT | \_FALS\_NUM | WORD |
| \_FORCE\_IN | BOOL | \_FORCE\_IN | BIT |
| \_FORCE\_OUT | BOOL | \_FORCE\_OUT | BIT |
|  |  | \_FPU\_FLAG\_E | BIT |
|  |  | \_FPU\_FLAG\_I | BIT |
|  |  | \_FPU\_FLAG\_O | BIT |
|  |  | \_FPU\_FLAG\_U | BIT |
|  |  | \_FPU\_FLAG\_V | BIT |
|  |  | \_FPU\_FLAG\_Z | BIT |
|  |  | \_FPU\_INFO | WORD |
|  |  | \_FPU\_LFLAG\_I | BIT |
|  |  | \_FPU\_LFLAG\_O | BIT |
|  |  | \_FPU\_LFLAG\_U | BIT |
|  |  | \_FPU\_LFLAG\_V | BIT |
|  |  | \_FPU\_LFLAG\_Z | BIT |
| \_FUSE\_ER | BOOL | \_FUSE\_ER | BIT |
| \_FUSE\_ER\_N | WORD | \_FUSE\_ER\_N | WORD |
| \_FUSE\_ER\_PMT | BOOL | \_FUSE\_ER\_PMT | BIT |
| \_FUSE\_ERR | ARRAY[0..7] OF WORD | \_FUSE\_ER0 | WORD |
|  |  | \_FUSE\_ER1 | WORD |
|  |  | \_FUSE\_ER2 | WORD |
|  |  | \_FUSE\_ER3 | WORD |
|  |  | \_FUSE\_ER4 | WORD |
|  |  | \_FUSE\_ER5 | WORD |
|  |  | \_FUSE\_ER6 | WORD |
|  |  | \_FUSE\_ER7 | WORD |
|  |  | \_GET\_CNT | DWORD |
|  |  | \_GT | BIT |
|  |  | \_GTE | BIT |
|  |  | \_HBCK\_ER | BIT |
| \_HS\_WAR | BOOL |  |  |
| \_HS\_WAR\_W | WORD |  |  |
| \_HS\_WARN | ARRAY[0..11] OF BOOL | \_HS\_WAR1 | BIT |
|  |  | \_HS\_WAR10 | BIT |
|  |  | \_HS\_WAR11 | BIT |
|  |  | \_HS\_WAR12 | BIT |
|  |  | \_HS\_WAR2 | BIT |
|  |  | \_HS\_WAR3 | BIT |
|  |  | \_HS\_WAR4 | BIT |
|  |  | \_HS\_WAR5 | BIT |
|  |  | \_HS\_WAR6 | BIT |
|  |  | \_HS\_WAR7 | BIT |
|  |  | \_HS\_WAR8 | BIT |
|  |  | \_HS\_WAR9 | BIT |
| \_HS\_ENABLE\_STATE | ARRAY[0..11] OF BOOL | \_HS1\_ENABLE\_STATE | BIT |
| \_HS\_REQ | ARRAY[0..11] OF BOOL | \_HS1\_REQ | BIT |
| \_HS\_REQ\_NUM | ARRAY[0..11] OF BOOL | \_HS1\_REQ\_NUM | BIT |
|  |  | \_HS10\_ENABLE\_STATE | BIT |
|  |  | \_HS10\_REQ | BIT |
|  |  | \_HS10\_REQ\_NUM | BIT |
|  |  | \_HS11\_ENABLE\_STATE | BIT |
|  |  | \_HS11\_REQ | BIT |
|  |  | \_HS11\_REQ\_NUM | BIT |
|  |  | \_HS12\_ENABLE\_STATE | BIT |
|  |  | \_HS12\_REQ | BIT |
|  |  | \_HS12\_REQ\_NUM | BIT |
|  |  | \_HS2\_ENABLE\_STATE | BIT |
|  |  | \_HS2\_REQ | BIT |
|  |  | \_HS2\_REQ\_NUM | BIT |
|  |  | \_HS3\_ENABLE\_STATE | BIT |
|  |  | \_HS3\_REQ | BIT |
|  |  | \_HS3\_REQ\_NUM | BIT |
|  |  | \_HS4\_ENABLE\_STATE | BIT |
|  |  | \_HS4\_REQ | BIT |
|  |  | \_HS4\_REQ\_NUM | BIT |
|  |  | \_HS5\_ENABLE\_STATE | BIT |
|  |  | \_HS5\_REQ | BIT |
|  |  | \_HS5\_REQ\_NUM | BIT |
|  |  | \_HS6\_ENABLE\_STATE | BIT |
|  |  | \_HS6\_REQ | BIT |
|  |  | \_HS6\_REQ\_NUM | BIT |
|  |  | \_HS7\_ENABLE\_STATE | BIT |
|  |  | \_HS7\_REQ | BIT |
|  |  | \_HS7\_REQ\_NUM | BIT |
|  |  | \_HS8\_ENABLE\_STATE | BIT |
|  |  | \_HS8\_REQ | BIT |
|  |  | \_HS8\_REQ\_NUM | BIT |
|  |  | \_HS9\_ENABLE\_STATE | BIT |
|  |  | \_HS9\_REQ | BIT |
|  |  | \_HS9\_REQ\_NUM | BIT |
|  |  | \_HUND\_WK | WORD |
|  |  | \_HUND\_WK\_DT | WORD |
| \_INIT\_DONE | BOOL | \_INIT\_DONE | BIT |
| \_INIT\_RUN | BOOL | \_INIT\_RUN | BIT |
|  |  | \_INST\_ER | BIT |
| \_IO\_DEER | BOOL | \_IO\_DEER | BIT |
| \_IO\_DEER\_N | WORD | \_IO\_DEER\_N | WORD |
| \_IO\_DEERR | ARRAY[0..7] OF WORD | \_IO\_DEER0 | WORD |
|  |  | \_IO\_DEER1 | WORD |
|  |  | \_IO\_DEER2 | WORD |
|  |  | \_IO\_DEER3 | WORD |
|  |  | \_IO\_DEER4 | WORD |
|  |  | \_IO\_DEER5 | WORD |
|  |  | \_IO\_DEER6 | WORD |
|  |  | \_IO\_DEER7 | WORD |
| \_IO\_ER\_PMT | BOOL | \_IO\_ER\_PMT | BIT |
|  |  | \_IO\_RWER | BIT |
|  |  | \_IO\_RWER\_N | WORD |
|  |  | \_IO\_RWER0 | WORD |
|  |  | \_IO\_RWER1 | WORD |
|  |  | \_IO\_RWER2 | WORD |
|  |  | \_IO\_RWER3 | WORD |
|  |  | \_IO\_RWER4 | WORD |
|  |  | \_IO\_RWER5 | WORD |
|  |  | \_IO\_RWER6 | WORD |
|  |  | \_IO\_RWER7 | WORD |
| \_IO\_TYER | BOOL | \_IO\_TYER | BIT |
| \_IO\_TYER\_N | WORD | \_IO\_TYER\_N | WORD |
| \_IO\_TYERR | ARRAY[0..7] OF WORD | \_IO\_TYER0 | WORD |
|  |  | \_IO\_TYER1 | WORD |
|  |  | \_IO\_TYER2 | WORD |
|  |  | \_IO\_TYER3 | WORD |
|  |  | \_IO\_TYER4 | WORD |
|  |  | \_IO\_TYER5 | WORD |
|  |  | \_IO\_TYER6 | WORD |
|  |  | \_IO\_TYER7 | WORD |
| \_IOPRM\_ER | BOOL | \_IOPRM\_ER | BIT |
|  |  | \_IP\_IFER | BIT |
|  |  | \_IP\_IFER\_N | WORD |
|  |  | \_IP\_IFER0 | WORD |
|  |  | \_IP\_IFER1 | WORD |
|  |  | \_IP\_IFER2 | WORD |
|  |  | \_IP\_IFER3 | WORD |
|  |  | \_IP\_IFER4 | WORD |
|  |  | \_IP\_IFER5 | WORD |
|  |  | \_IP\_IFER6 | WORD |
|  |  | \_IP\_IFER7 | WORD |
| \_KEY | DWORD | \_KEY | DWORD |
|  |  | \_KEY\_PREV | DWORD |
| \_LER | BOOL | \_LER | BIT |
|  |  | \_LER\_LATCH | BIT |
| \_LOCAL\_CON | BOOL | \_LOCAL\_CON | BIT |
|  |  | \_LOG\_FULL | BIT |
|  |  | \_LOG\_ROTATE | DWORD |
|  |  | \_LOGIC\_RESULT | WORD |
|  |  | \_LT | BIT |
|  |  | \_LTE | BIT |
|  |  | \_MOD\_HIS\_CNT | DWORD |
|  |  | \_MOD\_RD\_ERR\_CNT | DWORD |
|  |  | \_MOD\_WR\_ERR\_CNT | DWORD |
|  |  | \_MODBUS\_CON | BIT |
| \_MON\_ON | BOOL | \_MON\_ON | BIT |
|  |  | \_MON\_YEAR | WORD |
|  |  | \_MON\_YEAR\_DT | WORD |
|  |  | \_NEQ | BIT |
| \_OFF | BOOL | \_OFF | BIT |
| \_ON | BOOL | \_ON | BIT |
| \_OS\_DATE | DWORD | \_OS\_DATE | DWORD |
| \_OS\_VER | DWORD | \_OS\_VER | DWORD |
| \_OS\_VER\_PATCH | DWORD | \_OS\_VER\_PATCH | DWORD |
| \_P2P\_WAR | BOOL |  |  |
| \_P2P\_WAR\_W | WORD |  |  |
| \_P2P\_WARN | ARRAY[0..7] OF BOOL | \_P2P\_WAR1 | BIT |
|  |  | \_P2P\_WAR2 | BIT |
|  |  | \_P2P\_WAR3 | BIT |
|  |  | \_P2P\_WAR4 | BIT |
|  |  | \_P2P\_WAR5 | BIT |
|  |  | \_P2P\_WAR6 | BIT |
|  |  | \_P2P\_WAR7 | BIT |
|  |  | \_P2P\_WAR8 | BIT |
| \_P2P\_ENABLE\_STATE | ARRAY[0..7] OF BOOL | \_P2P1\_ENABLE\_STATE | BIT |
| \_P2P\_REQ | ARRAY[0..7] OF BOOL | \_P2P1\_REQ | BIT |
| \_P2P\_REQ\_NUM | ARRAY[0..7] OF BOOL | \_P2P1\_REQ\_NUM | BIT |
|  |  | \_P2P2\_ENABLE\_STATE | BIT |
|  |  | \_P2P2\_REQ | BIT |
|  |  | \_P2P2\_REQ\_NUM | BIT |
|  |  | \_P2P3\_ENABLE\_STATE | BIT |
|  |  | \_P2P3\_REQ | BIT |
|  |  | \_P2P3\_REQ\_NUM | BIT |
|  |  | \_P2P4\_ENABLE\_STATE | BIT |
|  |  | \_P2P4\_REQ | BIT |
|  |  | \_P2P4\_REQ\_NUM | BIT |
|  |  | \_P2P5\_ENABLE\_STATE | BIT |
|  |  | \_P2P5\_REQ | BIT |
|  |  | \_P2P5\_REQ\_NUM | BIT |
|  |  | \_P2P6\_ENABLE\_STATE | BIT |
|  |  | \_P2P6\_REQ | BIT |
|  |  | \_P2P6\_REQ\_NUM | BIT |
|  |  | \_P2P7\_ENABLE\_STATE | BIT |
|  |  | \_P2P7\_REQ | BIT |
|  |  | \_P2P7\_REQ\_NUM | BIT |
|  |  | \_P2P8\_ENABLE\_STATE | BIT |
|  |  | \_P2P8\_REQ | BIT |
|  |  | \_P2P8\_REQ\_NUM | BIT |
| \_PB1 | BOOL | \_PB1 | BIT |
| \_PB2 | BOOL | \_PB2 | BIT |
| \_PGM\_ER | BOOL | \_PGM\_ER | BIT |
|  |  | \_PUT\_CNT | DWORD |
|  |  | \_PUTGET\_ERR0 | WORD |
|  |  | \_PUTGET\_ERR1 | WORD |
|  |  | \_PUTGET\_ERR2 | WORD |
|  |  | \_PUTGET\_ERR3 | WORD |
|  |  | \_PUTGET\_ERR4 | WORD |
|  |  | \_PUTGET\_ERR5 | WORD |
|  |  | \_PUTGET\_ERR6 | WORD |
|  |  | \_PUTGET\_ERR7 | WORD |
|  |  | \_PUTGET\_NDR0 | WORD |
|  |  | \_PUTGET\_NDR1 | WORD |
|  |  | \_PUTGET\_NDR2 | WORD |
|  |  | \_PUTGET\_NDR3 | WORD |
|  |  | \_PUTGET\_NDR4 | WORD |
|  |  | \_PUTGET\_NDR5 | WORD |
|  |  | \_PUTGET\_NDR6 | WORD |
|  |  | \_PUTGET\_NDR7 | WORD |
| \_RBANK\_NUM | UINT | \_RBANK\_NUM | WORD |
|  |  | \_RBLOCK\_ER\_FLAG | DWORD |
|  |  | \_RBLOCK\_RD\_FLAG | DWORD |
|  |  | \_RBLOCK\_STATE | WORD |
|  |  | \_RBLOCK\_WR\_FLAG | DWORD |
|  |  | \_REF\_COUNT | DWORD |
|  |  | \_REF\_ERR\_CNT | DWORD |
|  |  | \_REF\_LIM\_CNT | DWORD |
|  |  | \_REF\_NG\_CNT | DWORD |
|  |  | \_REF\_OK\_CNT | DWORD |
| \_REMOTE\_CON | BOOL | \_REMOTE\_CON | BIT |
| \_RTC\_DATE | DATE | \_RTC\_DATE | WORD |
| \_RTC\_ER | BOOL | \_RTC\_ER | BIT |
| \_RTC\_TIME | ARRAY[0..7] OF BYTE |  |  |
| \_RTC\_TIME\_USER | ARRAY[0..7] OF BYTE |  |  |
| \_RTC\_TOD | TIME\_OF\_DAY | \_RTC\_TOD | DWORD |
| \_RTC\_WEEK | UINT | \_RTC\_WEEK | WORD |
| \_RTC\_WR | BOOL | \_RTC\_WR | BIT |
| \_RUN | BOOL | \_RUN | BIT |
| \_RUN\_EDIT\_CHK | BOOL | \_RUN\_EDIT\_CHK | BIT |
| \_RUN\_EDIT\_DONE | BOOL | \_RUN\_EDIT\_DONE | BIT |
| \_RUN\_EDIT\_NG | BOOL | \_RUN\_EDIT\_NG | BIT |
| \_RUN\_EDIT\_ST | BOOL | \_RUN\_EDIT\_ST | BIT |
| \_SCAN\_CUR | UINT | \_SCAN\_CUR | WORD |
| \_SCAN\_MAX | UINT | \_SCAN\_MAX | WORD |
| \_SCAN\_MIN | UINT | \_SCAN\_MIN | WORD |
| \_SCAN\_WR | BOOL | \_SCAN\_WR | BIT |
|  |  | \_SEC\_MIN | WORD |
|  |  | \_SEC\_MIN\_DT | WORD |
| \_SKIP\_ON | BOOL | \_SKIP\_ON | BIT |
| \_SLOT\_EMASK\_INFO | ARRAY[0..7] OF WORD | \_SLOT\_EMASK\_INFO\_0 | WORD |
|  |  | \_SLOT\_EMASK\_INFO\_1 | WORD |
|  |  | \_SLOT\_EMASK\_INFO\_2 | WORD |
|  |  | \_SLOT\_EMASK\_INFO\_3 | WORD |
|  |  | \_SLOT\_EMASK\_INFO\_4 | WORD |
|  |  | \_SLOT\_EMASK\_INFO\_5 | WORD |
|  |  | \_SLOT\_EMASK\_INFO\_6 | WORD |
|  |  | \_SLOT\_EMASK\_INFO\_7 | WORD |
| \_SLOT\_SKIP\_INFO | ARRAY[0..7] OF WORD | \_SLOT\_SKIP\_INFO\_0 | WORD |
|  |  | \_SLOT\_SKIP\_INFO\_1 | WORD |
|  |  | \_SLOT\_SKIP\_INFO\_2 | WORD |
|  |  | \_SLOT\_SKIP\_INFO\_3 | WORD |
|  |  | \_SLOT\_SKIP\_INFO\_4 | WORD |
|  |  | \_SLOT\_SKIP\_INFO\_5 | WORD |
|  |  | \_SLOT\_SKIP\_INFO\_6 | WORD |
|  |  | \_SLOT\_SKIP\_INFO\_7 | WORD |
| \_SOE\_LOG\_CNT | WORD | \_SOE\_LOG\_CNT | WORD |
| \_SOE\_LOG\_ROTATE | WORD | \_SOE\_LOG\_ROTATE | WORD |
| \_SOE\_READ\_LOG\_CNT | WORD | \_SOE\_READ\_LOG\_CNT | WORD |
| \_SOE\_READ\_LOG\_ROTATE | WORD | \_SOE\_READ\_LOG\_ROTATE | WORD |
| \_SP\_ER\_PMT | BOOL | \_SP\_ER\_PMT | BIT |
| \_SPPRM\_ER | BOOL | \_SPPRM\_ER | BIT |
| \_STOG | BOOL | \_STOG | BIT |
| \_STOP | BOOL | \_STOP | BIT |
| \_SWDT\_ER | BOOL | \_SWDT\_ER | BIT |
|  |  | \_SYS\_HIS\_CNT | DWORD |
| \_SYS\_STATE | DWORD | \_SYS\_STATE | DWORD |
| \_T100MS | BOOL | \_T100MS | BIT |
| \_T10S | BOOL | \_T10S | BIT |
| \_T1S | BOOL | \_T1S | BIT |
| \_T200MS | BOOL | \_T200MS | BIT |
| \_T20MS | BOOL | \_T20MS | BIT |
| \_T20S | BOOL | \_T20S | BIT |
| \_T2S | BOOL | \_T2S | BIT |
| \_T60S | BOOL | \_T60S | BIT |
| \_TASK\_ER | BOOL | \_TASK\_ER | BIT |
|  |  | \_TIME\_DAY | WORD |
|  |  | \_TIME\_DAY\_DT | WORD |
|  |  | \_TMRIDX\_ER | BIT |
|  |  | \_USER\_CLK | WORD |
|  |  | \_USER\_F | WORD |
|  |  | \_USER\_STAUS\_F | WORD |
|  |  | \_USER\_WRITE\_F | WORD |
|  |  | \_USR\_CLK0 | BIT |
|  |  | \_USR\_CLK1 | BIT |
|  |  | \_USR\_CLK2 | BIT |
|  |  | \_USR\_CLK3 | BIT |
|  |  | \_USR\_CLK4 | BIT |
|  |  | \_USR\_CLK5 | BIT |
|  |  | \_USR\_CLK6 | BIT |
|  |  | \_USR\_CLK7 | BIT |
| \_USTOP\_ON | BOOL | \_USTOP\_ON | BIT |
| \_WDT\_ER | BOOL | \_WDT\_ER | BIT |
|  |  | \_ZERO | BIT |
| \_\_H\_AC\_FAIL\_CNT | UDINT |  |  |
| \_\_H\_BUF\_FULL\_CNT | UDINT |  |  |
| \_\_H\_CA\_CNT | UDINT |  |  |
| \_\_H\_CA\_ERR\_CNT | UDINT |  |  |
| \_\_H\_CA\_LIM\_CNT | UDINT |  |  |
| \_\_H\_CON\_PATH\_STATE | ARRAY[0..7] OF BYTE |  |  |
| \_\_H\_DA\_RD\_ERR\_CNT | UDINT |  |  |
| \_\_H\_DA\_WR\_ERR\_CNT | UDINT |  |  |
| \_\_H\_ERR\_HIS\_CNT | UDINT |  |  |
| \_\_H\_LOG\_ROTATE | DWORD |  |  |
| \_\_H\_MOD\_HIS\_CNT | UDINT |  |  |
| \_\_H\_REF\_CNT | UDINT |  |  |
| \_\_H\_REF\_ERR\_CNT | UDINT |  |  |
| \_\_H\_REF\_LIM\_CNT | UDINT |  |  |
| \_\_H\_REF\_NG\_CNT | UDINT |  |  |
| \_\_H\_REF\_OK\_CNT | UDINT |  |  |
| \_\_H\_SYS\_HIS\_CNT | UDINT |  |  |

## XGK 그룹 명령어

|  |  |  |  |
| --- | --- | --- | --- |
| 명령어 | 명령어 | 명령어 | 명령어 |
| ABAND | BOR | DETECT | EBCMP |
| ABOR | BRL | DETECTP | ENCO |
| ABXNR | BRLP | DIS | ENCOP |
| ABXOR | BRR | DISP | FMOV |
| ANDDG< | BRRP | DIV | FMOVP |
| ANDDG<= | BRST | DIVB | GADD |
| ANDDG<> | BRSTP | DIVBP | GADDP |
| ANDDG= | BTOW | DIVP | GBCD |
| ANDDG> | BTOWP | DIVU | GBCDP |
| ANDDG>= | BXNR | DIVUP | GBIN |
| ANDG< | BXOR | DMAX | GBINP |
| ANDG<= | DATERD | DMAXP | GBMOV |
| ANDG<> | DATERDP | DMIN | GBMOVP |
| ANDG= | DATEWR | DMINP | GDEQ |
| ANDG> | DATEWRP | DMUX | GDEQP |
| ANDG>= | DAVE | DMUXP | GDGE |
| ASC | DAVEP | DSCAL | GDGEP |
| ASCP | DDIV | DSCALP | GDGT |
| AVE | DDIVB | DSCH | GDGTP |
| AVEP | DDIVBP | DSCHP | GDLE |
| BAND | DDIVP | DSORT | GDLEP |
| BCDL | DDIVU | DSUM | GDLT |
| BCDLP | DDIVUP | DSUMP | GDLTP |
| BCDR | DECO | DTCMP | GDNE |
| BCDRP | DECOP | DTCMPP | GDNEP |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 명령어 | 명령어 | 명령어 | 명령어 | 명령어 |
| GEQ | GWANDP | MAX | RAMP | UNIP |
| GEQP | GWOR | MAXP | RBCD | VRD |
| GET | GWORP | MID | RBCDP | VWR |
| GETM | GWXNR | MIDP | REPLACE | WSFL |
| GETMP | GWXNRP | MIN | REPLACEP | WSFLP |
| GETP | GWXOR | MINP | RSCAL | WSFR |
| GGE | GWXORP | MUX | RSCALP | WSFRP |
| GGEP | GXCHG | MUXP | RTRAMP | WSFT |
| GGT | GXCHGP | ORDG< | SCAL | WSFTP |
| GGTP | HEX | ORDG<= | SCALP | WTOB |
| GLE | HEXP | ORDG<> | SCH | WTOBP |
| GLEP | LBCD | ORDG= | SCHP | XPWR |
| GLT | LBCDP | ORDG> | SEG | XSRD |
| GLTP | LOADDG< | ORDG>= | SEGP | XSWR |
| GMOV | LOADDG<= | ORG< | SORT | XTWR |
| GMOVP | LOADDG<> | ORG<= | SR | XVRD |
| GNE | LOADDG= | ORG<> | SRD | XVWR |
| GNEP | LOADDG> | ORG= | SUM |  |
| GSUB | LOADDG>= | ORG> | SUMP |  |
| GSUBP | LOADG< | ORG>= | TCMP |  |
| GSWAP | LOADG<= | PUT | TCMPP |  |
| GSWAP2 | LOADG<> | PUTM | TFLK |  |
| GSWAP2P | LOADG= | PUTMP | TRAMP |  |
| GSWAPP | LOADG> | PUTP | TWR |  |
| GWAND | LOADG>= | PWR | UNI |  |