

Hardware Chip User manual

Motor Control Chip

CAMC-QI



Product Information

Full information about other AJINEXTEK products are available by visiting our Web Site at:

Home Page : www.ajinextek.com

E-mail : support@ajinextek.com

Useful Contact Information

Customer Support Seoul

Tel : 82-31-436-2180~2 Fax: 82-31-436-2183

Customer Support Taegu

Tel : 82-53-593-3700~2 Fax: 82-53-593-3703



AJINEXTEK's sales team is always available to assist you in making your decision the final choice of boards or systems is solely and wholly the responsibility of the buyer. AJINEXTEK's entire liability in respect of the board or systems is as set out in AJINEXTEK's standard terms and conditions of sale

Contents

1. Outline	6
2. Specification	8
2.1. Specification table	8
2.2. Block Diagram	9
2.3. In/Out pin assignment	10
2.4. Terminal pin description	11
3. Address Map	14
3.1. Axis address range	14
3.2. Each axis internal address map	14
3.2.1. WRITE OPERATION	14
3.2.2. READ OPERATION	14
4. Port description	15
4.1. DATA PORT(DATAP)	15
4.2. COMMAND WRITE PORT(CMDP)	15
4.3. Universal In/Out Write/Read PORT(UIOP)	15
4.4. Main status Read PORT(STATP)	15
5. Command description	17
5.1. Command 실행 방법	17
5.2. Command list	21
5.3. 명령어 상세 설명 및 기능 설명	24
5.3.1. COMMANDS FOR SETTING REGISTER ABOUT DRIVE	24
5.3.2. UNIVERSAL IN/OUT	29
5.3.3. DRIVE START COMMANDS	30
5.3.4. CONTROL COMMANDS	30
5.3.5. COMMANDS FOR SETTING PERIPHERAL FUNCTION REGISTERS.	31
5.3.6. COMMANDS FOR SETTING SCRIPT FUNCTION REGISTERS.	34
5.3.7. COMMANDS FOR SETTING COUNTER AND COMPARATOR CONFIGURATION.	44
5.3.8. COMMANDS FOR SETTING ENVIRONMENTS AND CHECKING STATUS.	46
6. Functional Description	56
6.1. Position override.	56
6.1.1. 정속 구동 중 [POS]값이 기존의 값보다 큰 값으로 변경되었을 때	56
6.1.2. 감속 구동 중 [POS]값이 기존의 값보다 큰 값으로 변경되었을 때	56
6.1.3. 구동 중 [POS]값이 기존의 값보다 작은 값으로 변경되었을 때	57
6.2. 원점 구동	57
6.2.1. 원점 복귀([DCFG][6~0] = 0X20~0X21)	57
6.2.2. 원점 이탈([DCFG][6~0] = 0X22~0X23)	63
6.2.3. 원점 검색([DCFG][6~0] = 0X24~0X25)	63
6.3. MPG(Manual Pulse Generation) input function	65
7. Electrical Characteristics	66
7.1. Absolute Maximum Rating(VSS = 0V)	66
7.2. Recommended Operation condition(VSS = 0V)	66
7.3. DC Characteristics	66
7.4. AC Characteristics	66

7.4.1. CPU INTERFACE (VDD = 5V ± 0.25V, IVDD = 1.8V, TA = 0 TO +70℃)	66
7.4.2. CPU INTERFACE (VDD = 3.3V ± 0.15V, IVDD = 1.8V, TA = 0 TO +70℃)	67
7.4.3. IO INTERFACE	69
8. Package diagram	70
9. Appendix	71
9.1. Precaution	71
9.1.1. 구동 중 [SSTOP], [STOP], [ESTOP] 명령어 실행 시 주의 사항	71
9.1.2. 트리거 기능 사용 중 주의 사항	71
9.1.3. 서보 위치 결정 완료 기능(인포지션 기능) 사용 중 주의 사항	71
9.1.4. 삼각 구동 방지 기능 사용 중 주의 사항	71
9.2. 레지스트 및 명령어 일람	73
9.3. 찾아 보기(하단 페이지 번호)	76

Revision History

Development Issue	Marking version	Comments
Rev. 0.5 issue 2.0	Preliminary version	Oct. 19, 2006
Rev. 1.0 issue 1.0	Preliminary version	Dec. 13, 2006
Rev. 1.0 issue 1.3	Rev 1.0	Apr. 04, 2007
Rev. 1.0 issue 1.4	Rev 1.0	May. 28, 2007. Fix pin assignment and description.
Rev. 1.0 issue 1.5	Rev 1.0	Jul. 04, 2007. Fix command description for [PDCFG]
Rev. 1.3 issue 0.4	Rev 1.0	Apr. 28, 2008. Fix command description for [TMPRP1/2]
Rev. 1.3 issue 0.5	Rev 1.0	Jul. 15, 2008. Delete [CNTCF3] register functions.
Rev. 1.3 issue 0.6	Rev 1.0	Aug. 08, 2008. Modify pin description table.
Rev. 1.3 issue 0.7	Rev 1.0	Apr. 11, 2010. Fix command description for [UCFG1]

1. Outline

CAMC-QI는 4축 모션 제어용 IC로 스텝 모터 및 펄스 입력형 서버 드라이버를 제어하기 위한 위치/속도/보간 제어용 고속 펄스를 출력할 수 있다.

이 칩은 CPU 버스 인터페이스를 통하여 전달된 Command를 참조하여 CPU의 부하를 감소시키면서 연속적인 모션제어를 구현할 수 있다.

CAMC-QI는 모터 구동 중 발생하는 특정 이벤트 시간에 실행할 Command를 예약할 수 있는 Script 기능이 있어 모터 구동 중의 특정 동작을 구현하기가 용이하다. 모든 파라미터들은 소프트웨어적으로 설정할 수 있으며, 펄스 출력 중에도 변경이 가능하다. CAMC-QI의 자세한 특징은 다음과 같다.

▶ 독립 4축 제어.

각축의 구동에 필요한 기능들이 독립 적으로 구현되어 있기 때문에 4축을 동시에 모든 기능들을 사용하며 최고 주파수 출력 펄스로 제어 할 수 있다.

▶ 최대 13.107 Mpps 펄스 출력

최대 펄스출력속도는 입력 클럭 39.3216MHz(사용권장 주파수)를 기준으로 약 13.107 Mpps(pulse per second)이다.

▶ 다양한 가감속 제어

S-Curve/직선 가감속 속도 프로파일을 자유자재로 만들 수 있으며, 가속/감속에 서로 다른 종류의 가감속을 적용할 수 있다. 또한 가속이 없거나 감속이 없는 속도 프로파일을 제공한다. 다양한 종류의 속도 프로파일을 연속적으로 구동함으로써 이동 구간마다 적당한 속도로 간단하게 적용할 수 있어 고정밀 제어를 가능하게 한다.

▶ 보간 제어 기능

최대 4축 직선 보간 및 임의 2축 원호 보간 기능이 가능하며, 멀티 칩 보간을 지원한다. 보간 중 선속일정 기능을 지원하며, 보간 정밀도는 출력 되는 펄스 하나에 해당하는 거리의 0.5배이다. 특히 벡터 속도 보상 기능이 있어 향상된 보간 제어가 가능하다.

▶ 8/16 bit CPU interface

8-bit 및 16-bit Data bus를 제공한다. 또한 연속 적인 제어 명령을 수행하기 위하여 7개의 명령어 버퍼가 구성되어 고속 CPU에 대한 command hold 시간은 최소화 할 수 있다.

▶ 드라이브 예약 기능.

모든 드라이브는 최대 7개까지 Previous register에 예약 가능하다. 특히 연속 보간 중 짧은 보간 패턴 때문에 구동 속도에 제약을 최소화 할 수 있다.

▶ 다양한 드라이브 모드 제공

연속 드라이브, 신호 및 원점 검출 드라이브, 지정 펄스 수 드라이브, 센서 위치 결정 드라이브, 외부 펄스 구동 드라이브, 직선 보간 드라이브, 원호보간 드라이브, 슬레이브 드라이버를 지원한다.

▶ 인터럽트 발생기능

각 축당 64개의 다양한 경우에 대한 인터럽트를 생성 할 수 있어 광범위한 경우에 대한 제어가 가능하다.

▶ 다양한 펄스 출력 기능

접속한 모터 드라이브의 사양에 맞추어 펄스 출력방식을 선택할 수 있도록 8 종류의 출력 방식이 갖추어져 있다. 펄스 출력 형태는 Software적으로 설정 가능하다.

▶ Limit 정지 기능

기계 계의 over-run limit 신호로서 정/역 방향 각각에 급정지용/감속정지용 limit 신호 입력 단자가 있다. 특히 각 방향에 대한 사용 여부를 따로 설정 할 수 있으며 유효한 입력 레벨은 시스템의 설정에 따라 가변 할 수 있다.

▶ 탈조 검출 기능

내부 카운터 값과 외부 카운터 값의 차이 값의 범위를 범용 비교기 레지스터에 설정할 수 있다. 검출된 결과를 스크립트 이벤트로 사용하여 급정지 하거나 감속정지 하도록 설정 할 수 있다.

▶ 스크립트(Script) 처리 기능

다양한 종류의 이벤트가 발생 하였을 때 실행될 읽기/쓰기 명령 코드를 미리 예약 저장해 둘 수 있는데, 축당 6개의 명령어 스크립트가 하나의 이벤트에 대하여 동시에 수행될 수 있으며, 특히 1번/2번 스크립트는 순차적인 이벤트에 대한 명령 수행 등을 위하여 크기가 15인 QUEUE 형태를 가지고 있다.

▶ 제어 신호의 설정 기능

외부에서 입력되는 제어 신호의 유효 입력 레벨 및 사용 유무, 사용 유형에 대하여 다양한 설정이 가능하다.

▶ 지정 펄스 수 드라이버 기능 강화

S자 속도 프로파일로 구동 시 삼각 구동 방지 기능이 기본적으로 제공되며, 연속적인 지정 펄스 수 드라이버를 사용하여 이동 거리마다 다양한 속도 패턴을 적용할 수 있다.

▶ 엔코드 입력 내장 (A, B, Z 상 입력 처리)

A, B 상 입력으로 외부 카운터가 동작하여 기계 계의 현재 위치를 알 수 있고 원점 검색 시 자동으로 초기화 할 수 있는 기능이 있어 고속으로 정밀한 원점 검색이 가능하다.

▶ 입력 신호 디지털 필터 내장

외부에서 입력되는 신호의 노이즈를 제거하는 필터 기능을 가지고 있다. 39.3216MHz 기준 최대 26mSec 펄스폭 신호까지 필터 된다.

▶ 다양한 입력에 대한 카운터 및 비교기 내장.

축당 5개의 28bit의 범용 카운터/비교기가 가 있다. 특히 카운터와 연결된 비교기의 결과로 생성된 이벤트를 스크립트 기능과 연동하여 모든 명령을 예약 실행 할 수 있다.

▶ 순환 위치 카운트 기능

내부/외부 위치 순환 카운트 기능이 있어, 상한 값과 하한 값 범위 내에서 링 카운트로 동작하게 된다.

▶ 소프트웨어 리미트 설정 기능

▶ 타이머 기능 내장

축당 2개의 타이머가 내장되어 있어 주기적인 동작을 스크립터와 연동하여 구동 할 수 있다.

▶ 범용 입력/출력 단자

측당 12개의 범용 입출력 단자가 있다.

▶ 트리거 펄스 출력 기능

각축의 카운터 #1/#2 값을 기준으로 일정 주기 또는 특정 값에서 트리거 펄스를 발생할 수 있다. 특히 진행 방향에 따른 설정 및 주기 트리거 시작/종료 값을 설정하여 고속으로 이동 중 정확한 위치에 대하여 트리거 펄스를 발생 시킬 수 있다. 특히 절대 위치 트리거 위치는 15개의 queue에 미리 저장 해 둘 수 있어 고속 구동 중 간격이 일정하지 않고 세밀한 위치에서도 트리거 발생을 가능하게 한다.

▶ 서버 드라이버 인터페이스 제공

잔여 펄스 제거, 인포지션 기능, 알람 모니터링 기능을 서버 드라이버와 연결하여 부가 기능을 수행한다.

▶ 외부 입력 펄스에 의한 구동.

모든 드라이버의 기준 구동 펄스를 외부로부터 입력 받아 실행할 수 있다.

▶ 다양한 홈 검색 기능

다양한 속도 프로파일을 사용하여 원점 복귀를 수행하며, 구동 순서를 조합하여 정밀한 홈 검색 기능을 제공한다.

▶ 다축 동기 시작 기능

여러 축의 (동일 IC 내 또는 외부 IC간) 동기 구동을 위해 Synchronization start 입/출력 핀이 있다. 이 입력 핀이 활성화 되기 전까지는 드라이브를 시작하더라도 펄스를 출력하지 않으며, 다축 동기 기능을 사용하는 축들 중 임의의 축에서 Synchronization start 핀을 활성화 하면서 드라이버를 시작 할 수 있다. 따라서, 이 핀을 이용하여 여러 개의 CAMC-QI의 펄스 출력을 동시에 시작할 수 있다.

▶ 다축 동기 종료 기능

여러 축의 (동일 IC 내 또는 외부 IC간) 동기 정지를 위해 Synchronization stop 입/출력 핀이 있다. 이 입력 핀이 활성화 되면 해당 축은 드라이브를 중지 하며, 다축 동기 정지 기능을 사용하는 축들 중 임의의 축에서 Synchronization stop 핀을 활성화 하면서 드라이버를 중지 할 수 있다. 따라서, 이 핀을 이용하여 여러 개의 CAMC-QI의 펄스 출력을 동시에 종료할 수 있다.

2. Specification

2.1. Specification table

Power supply	: IO : $+3.3V \pm 5\%$ or $+5.0 \pm 5\%$, Core : $+1.8V \pm 5\%$
Reference clock	: Max 39.3216 MHz(standard)
Control axes	: 4 axes.
Velocity setting	: 16-bit (65,536)
Acceleration/Deceleration rate	: 16-bit (65,536)
Magnitude of output pulse rate	: 0.01 ~ 200(Reference clock 39.3216MHz) 0.01 times - 0.01 ~ 655.35 pps 1 times - 1.0 ~ 65,535 pps 50 times - 50 ~ 3,276,750 pps 200 times - 200 ~ 13,107,000 pps
Deceleration pointer setting	: 28bit (0 ~ 268,435,456)
Drive functions	: Preset pulse drive, continuous drive Sensor and ORG search drive Manual pulse generation drive, Interpolation drive.
Interpolation drive	: Any 3 axes linear interpolation. Any 2 axes circular interpolation. Interpolation error : 0.5 of 1 pulse distance.
Override function	: Velocity/Position override during driving.
Velocity profiles	: Symetric/Asymetric S-curve profile. Symetric/Asymetric partial S-curve profile. Symetric/Asymetric Linear profile.
Servo drive interface	: Digital pulse method.
Output pulse method	: One pulse/Two pulse/Two phase(Total 10 methods)
Universal in/out	: 12 channel for each axis(CMOS Schmitt Trigger Level)
Comparators	: 5 comparator for each axis(28-bit) (Sources : Internal position, External position, Deviation, MPG pulse, Event, Velocity)
Counter function	: 5 universal counters for each axis(28-bit)
Script/Data capture function	: Simultaneous event check(4EA each axis), Command reservation : 32 EA for each axis.
Position triggering function	: Pulse generation at Periodic distance/Absolute position.

2.2. Block Diagram

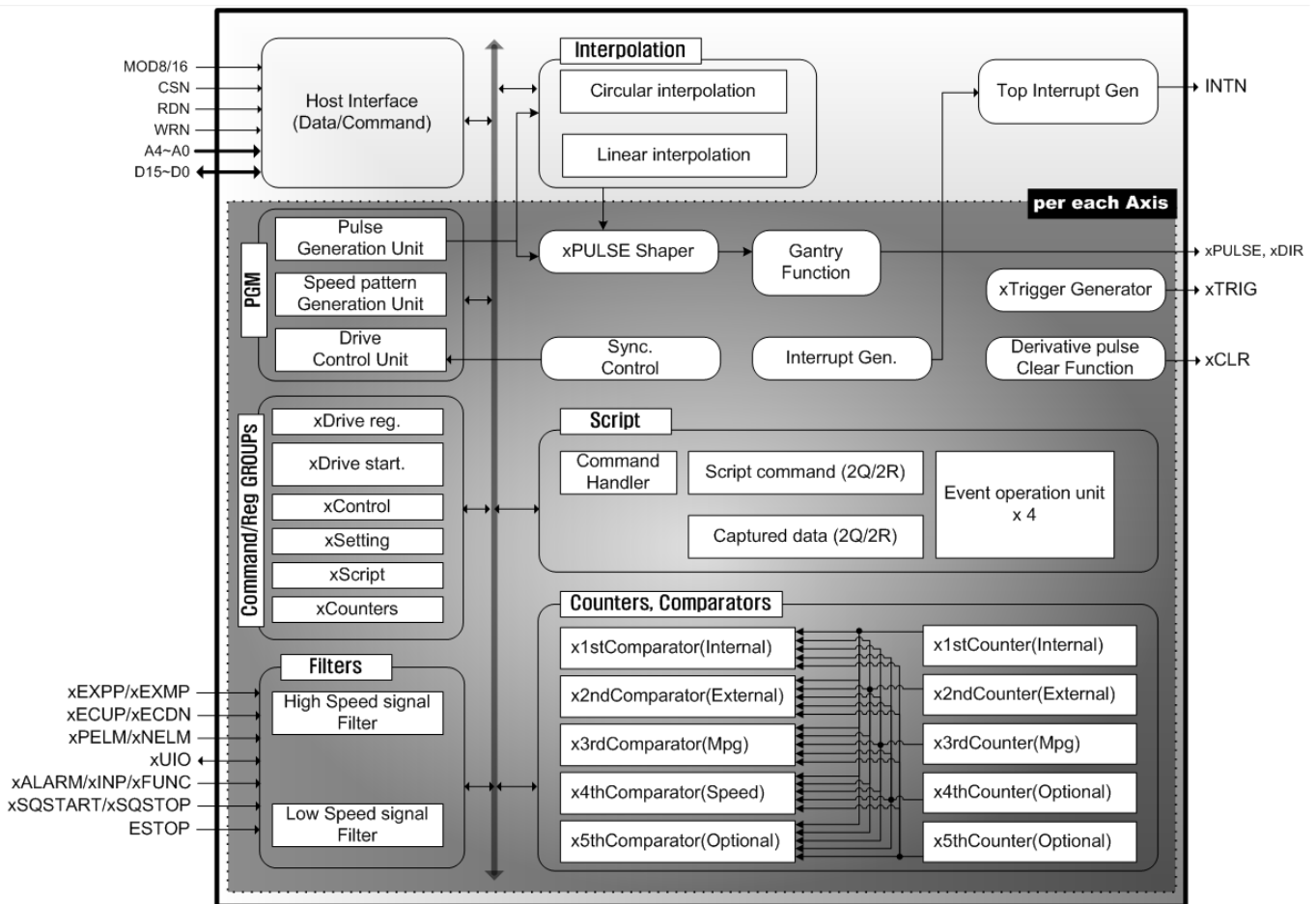


그림 2-1. CAMC - QI Functional block diagram

2.3. In/Out pin assignment

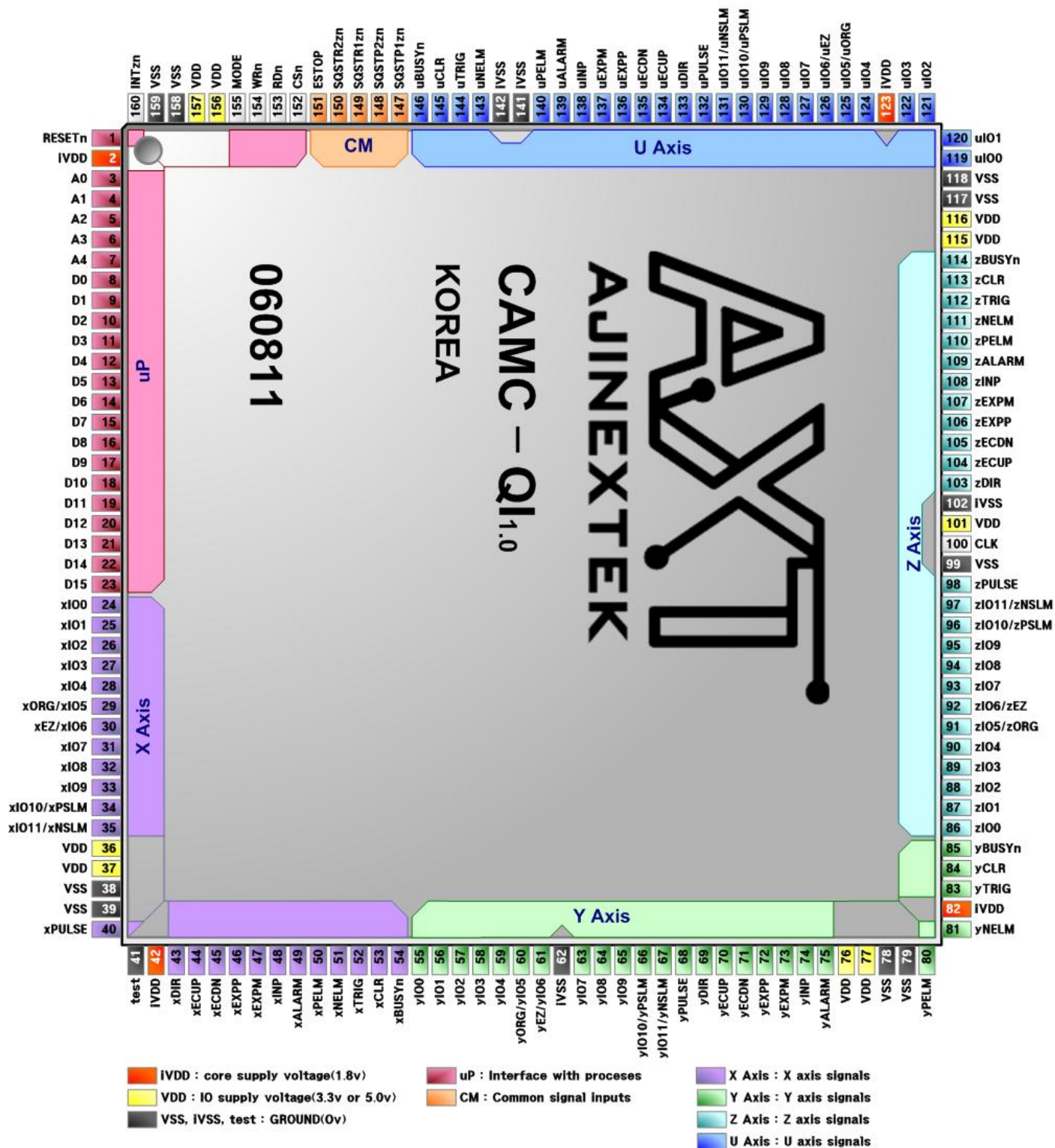


그림 2-2. Pin assignment(160pinLQFP, 26x26 mm, Pitch: 0.5 mm)

2.4.Terminal pin description

신호명	단자 번호	속성	신호의 설명
RESETn	1	입력 Pull-UP	CAMC-QI 를 리셋(초기화)하는 신호이며, CLK가 10사이클 이상인 동안 RESETn을 Low로 하면 리셋 됩니다. 전원투입 시에는 반드시 본 IC를 RESETn신호로 리셋 해야 한다. 【주의】 CLK가 입력되지 않으면 RESET*을 Low로 해도 리셋 되지 않는다.
A(0~4)	3~7	입력	포트를 선택하기 위한 어드레스 신호이다.
D(0~15)	8~23	양방향 8mA	양방향의 16비트 데이터 버스. 시스템의 데이터 버스에 접속하여, CS*=Low와 RD*=Low일 때 출력상태가 된다. 이 외에는 High impedance 입력상태가 되며 데이터 버스를 8비트로 사용하는 경우는 상위 D15~D8은 사용하지 않으므로, D15~D8 open 또는 10K이상으로 사용하여 풀업하면 된다.
xIO(0~4) yIO(0~4) zIO(0~4) uIO(0~4)	24~28 55~59 86~90 119~122	양방향 8mA	Universal IN/OUT:범용 입/출력 신호이다. 특히 IO3,IO4 는 one-shot 펄스 형태의 출력이 가능하다. 초기 출력으로 설정되어 있다.
xIO5/ORG yIO5/ORG zIO5/ORG uIO5/ORG	29 60 91 125	양방향 8mA	범용 입력/원점 검색 신호 입력 : 원점 센서 신호의 입력 및 범용 입출력 신호. 초기 입력으로 설정되어 있다.
xIO6/EZ yIO6/EZ zIO6/EZ uIO6/EZ	30 61 92 126	양방향 8mA	범용 입력/Z phase 입력 : 엔코더 Z상 신호의 입력 및 범용 입출력 신호. 사용 모드는 초기 입력으로 설정되어 있다.
xIO(7~9) yIO(7~9) zIO(7~9) uIO(7~9)	31~33 63~65 93~95 127~129	양방향 8mA	Universal IN/OUT:범용 입/출력 신호이다. 특히 IO3,IO4 는 one-shot 펄스 형태의 출력이 가능하다. 초기 입력으로 설정되어 있다.
xIO10/PSLM yIO10/PSLM zIO10/PSLM uIO10/PSLM	34 66 96 130	양방향 8mA	Over Run Slowdown Limit+ : +방향의 감속 정지 오버런 리미트 신호/범용 입출력. 리미트 신호로 사용시 유효/무효, 액티브 레벨을 모드 선택할 수 있다. 유효로 설정된 경우 +방향의 드라이브 펄스 출력 중에, 이 신호가 액티브 되면 드라이브는 감속 정지한다. 범용 입/출력으로 사용시 입력 및 출력 모드를 선택 설정 가능하다. 초기 범용 입력으로 동작한다.
xIO11/NSLM yIO11/NSLM zIO11/NSLM uIO11/NSLM	35 67 97 131	양방향 8mA	Over Run Slowdown Limit- : -방향의 감속 정지 오버런 리미트 신호/범용 입출력. 리미트 신호로 사용시 유효/무효, 액티브 레벨을 모드 선택할 수 있다. 유효로 설정된 경우 -방향의 드라이브 펄스 출력 중에, 이 신호가 액티브 되면 드라이브는 감속 정지한다. 범용 입/출력으로 사용시 입력 및 출력 모드를 선택 설정 가능하다. 초기 범용 입력으로 동작한다.
xPULSE yPULSE zPULSE uPULSE	40 68 98 132	출력 8mA	펄스/방향 : 리셋시의 상태는 Low레벨이 되어 있고, 드라이브 동작에 들어가면 duty50%(정속시)의 펄스 또는 드라이브의 방향이 출력된다. 펄스의 출력 레벨 및 펄스/방향은 모드선택으로 변경한다
xDIR yDIR zDIR uDIR	43 69 103 133	출력 8mA	방향/펄스 : 리셋시의 상태는 Low레벨이 되어 있고, 드라이브 동작에 들어가면 duty50%(정속시)의 펄스 또는 드라이브의 방향이 출력된다. 펄스의 출력 레벨 및 펄스/방향은 모드선택으로 변경한다
xECUP yECUP zECUP uECUP	44 70 104 134	입력* Pull-UP	Encoder A phase : 엔코더 A상 신호의 입력입니다. B상 신호와 함께, 내부에서 업/다운 펄스로 변환되고, 실 위치를 위한 카운터 입력이 된다.
xECDN yECDN zECDN uECDN	45 71 105 135	입력* Pull-UP	Encoder B phase : 엔코더 B상 신호의 입력입니다. A상 신호와 함께, 내부에서 업/다운 펄스로 변환되고, 실 위치를 위한 카운터 입력이 된다.

신호명	단자 번호	속성	신호의 설명
xEXPP yEXPP zEXPP uEXPP	46 72 106 136	입력* Pull-UP	External positive reference : 외부에서 입력되는 기준 펄스 입력의 Positive 펄스 입력 단자이다. 입력 신호형태는 ECUP과 동일하게 적용된다.
xEXMP yEXMP zEXMP uEXMP	47 73 107 137	입력* Pull-UP	External negative reference : 외부에서 입력되는 기준 펄스 입력의 Negative 펄스 입력 단자이다. 입력 신호형태는 ECDN과 동일하게 적용된다.
xINP yINP zINP uINP	48 74 108 138	입력* Pull-UP	Inposition : 서보모터 드라이브의 인포지션 (위치 결정 완료)출력에 대응하는 입력 신호입니다. 유효/무효, 액티브 레벨은 사용자가 설정할 수 있는데, 유효로 설정하면 드라이브 종료 후, 이 신호가 액티브가 되는 것을 기다리고 있으며 드라이브 BUSY 상태를 유지한다.
xALARM yALARM zALARM uALARM	49 75 109 139	입력* Pull-UP	Servo Alarm : 서보모터 드라이브의 알람 출력에 대응하는 입력 신호. 유효/무효, 액티브 레벨은 모드 선택할 수 있다. 유효로 설정한 경우 드라이브 중 ALARM신호에 액티브 되면 현재 구동중인 드라이브가 정지 한다.
xPELM yPELM zPELM uPELM	50 80 110 140	입력* Pull-UP	Over Run Emergency Limit+ : +방향의 오버런 리미트 신호. 필터 기능 무효인 경우, 2 CLK 이상의 액티브 펄스 폭이 필요하다. 유효/무효, 액티브 레벨을 모드 선택할 수 있다. 유효로 설정된 경우 +방향의 드라이브 펄스 출력 중에, 이 신호가 액티브 되면 드라이브는 급정지한다.
xNELM yNELM zNELM uNELM	51 81 111 143	입력* Pull-UP	Over Run Emergency Limit- : -방향의 오버런 리미트 신호. 필터 기능 무효인 경우, 2 CLK 이상의 액티브 펄스 폭이 필요하다. 유효/무효, 액티브 레벨을 모드 선택할 수 있다. 유효로 설정된 경우 -방향의 드라이브 펄스 출력 중에, 이 신호가 액티브 되면 드라이브는 급정지한다.
xTRIG yTRIG zTRIG uTRIG	52 83 112 144	출력 8mA	Trigger 출력 신호 : Trigger 기능 사용시 출력을 위한 신호이다. 출력 신호의 액티브 레벨은 선택 가능하다. 초기 출력 신호 레벨은 Low이다.
xCLR yCLR zCLR uCLR	53 84 113 145	출력 8mA	서보 드라이브의 잔여 펄스 제거 출력 신호이다. 출력 형태(펄스, 레벨) 및 액티브 레벨을 설정하여 사용한다.
xBUSY yBUSY zBUSY uBUSY	54 85 114 146	출력 8mA	드라이브 구동 중 High 레벨 신호를 출력한다. Inposition 대기 동안에도 High 레벨을 출력한다.
CLK	100	입력	내부동기회로를 동작시키는 클럭 신호이다. 주파수 39.3216MHz를 입력한다. CLK 입력을 기준으로 신호 필터 주파수, 출력 펄스 신호 생성, 가감속 시간 결정 등이 이루어지므로 주의가 필요하다.
SQSTR1zn SQSTR2zn	147 148	양방향 Open- Drain	동기 맞춤 입/출력 : 동기 구동 시작 기능이 유효로 설정되었을 때, 구동 명령실행 후 이 신호가 Low일 때 드라이브 구동이 시작된다. 10kΩ 저항으로 풀업하여 외부 신호 및 다른 칩들과 연결한다.
SQSTP1zn SQSTP2zn	149 150	양방향 Open- Drain	동기 맞춤 입/출력 : 동기 구동 종료 기능이 유효로 설정되었을 때, 구동 명령실행 후 이 신호가 Low일 때 드라이브 구동이 종료된다. 10kΩ 저항으로 풀업하여 외부 신호 및 다른 칩들과 연결한다.
ESTOP	151	입력* Pull-UP	Emergency stop 입력 : 각 축의 비상정지 신호 입력에 대한 기능이 유효로 설정되었을 때, 이 신호가 액티브 레벨로 입력되면 드라이브 구동이 즉시 정지 된다.
CSn	152	입력 Pull-UP	Chip Select : 본 IC를 I/O 디바이스로써 선택하기 위한 입력신호이다. 본 IC를 읽기/쓰기 액세스 할 때, Low 레벨로 한다.
RDn	153	입력 Pull-UP	Read Strobe : 본 IC의 레지스터로부터 데이터를 읽을 때에 Low로 합니다. CS*을 Low로 하고 RD*을 Low로 하면, RD*이 Low 기간만, A4~A0의 어드레스 신호에 따라 선택된 레지스터의 데이터가 데이터 버스로 출력된다.

신호명	단자 번호	속성	신호의 설명
WRn	154	입력 Pull-UP	Write Strobe : 본 IC의 레지스터로 데이터를 입력할 때 Low로 한다. WR*이 Low 기간은 CSn 및 A4~A0을 확정해야 하며, WR*이 ↑일 때, 데이터 버스의 내용이 레지스터에 latch되므로, WRN의 ↑ 전후에는 D15~D0의 값이 확정해 있지 않으면 안 된다.
MODE	155	입력 Pull-DN	High=16비트, Low=8비트 : 16비트 데이터 버스/8비트 데이터 버스를 선택한다. Hi레벨로 하면 16비트 데이터 버스가 되고 IC 내의 읽기/쓰기 포트를 16비트로 액세스하며, Low레벨로 하면 데이터 버스는 D7~D0의 8비트만 유효하게 되고, 읽기/쓰기 포트를 8비트로 액세스 한다.
INTzn	160	출력 Open- Drain	Interrupt : 상위 CPU에 대한 인터럽트요구 신호입니다. 각축의 인터럽트 Flag가 하나 이상이 Set되면 INT*은 액티브 레벨이 된다. 인터럽트 액티브 레벨 선택 가능하며 초기 액티브 레벨은 High이다. 10kΩ 저항으로 풀업하여 외부 신호 및 다른 칩들과 연결한다.
test	41		IC 테스트 신호 입력. 정상 동작을 위해서는 반드시 그라운드(0V)로 연결해야 한다.
VSS, IVSS	38,39,62,78,79, 99,102,117,118, 141,142,158,159		그라운드(0V). 반드시 13개 모두의 단자를 접속.
IVDD	2,42,82,123		+1.8V전원. 반드시 4개 모두의 단자를 접속.
VDD	36,37,76,77,101, 115,116,156,157		+3.3V전원 또는 +5.0 중 선택. 반드시 9 모두의 단자를 동일 한 전원으로 연결해야 한다.

입력*) Schmitt trigger 입력 입니다.

3. Address Map

3.1. Axis address range

A4	A3	Description
0	0	X 축 할당 영역
0	1	Y 축 할당 영역
1	0	Z 축 할당 영역
1	1	U 축 할당 영역

3.2. Each axis internal address map

3.2.1. Write operation

1) 8-bit interface(mode8_16 pin = '0')

A2	A1	A0	PORT assign	Description
0	0	0	DATAPL0	Write data port lsb0(7~0 bit)
0	0	1	DATAPL1	Write data port lsb1(15~8 bit)
0	1	0	DATAPM0	Write data port msb0(23~16 bit)
0	1	1	DATAPM1	Write data port msb1(31~24 bit)
1	0	0	CMDP0	Write command.
1	0	1	CMDP1	Assign axis for command execution.
1	1	0	UIOP0	Write UIO0~UIO7 terminal *1
1	1	1	UIOP1	Write UIO8~UIO11 terminal *1

Note *1) UIO핀의 모드가 출력으로 설정된 것에 유효하다.

2) 16-bit interface(mode8_16 = '1')

A2	A1	PORT assign	Description
0	0	DATAPL	Write data port lsb(15~0 bit)
0	1	DATAPM	Write data port msb(31~16 bit)
1	0	CMDP	Write command and axis assignment.
1	1	UIOP	Write UIO0~UIO11 terminal *1

Note *1) UIO핀의 모드가 출력으로 설정된 것에 유효하다.

3.2.2. Read operation

3) 8-bit interface(mode8_16 pin = '0')

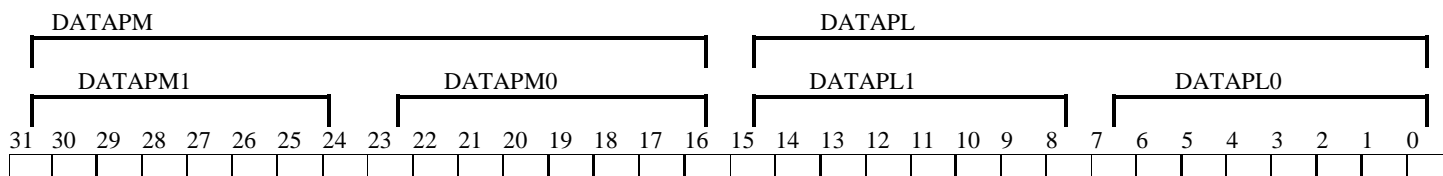
A2	A1	A0	PORT assign	Description
0	0	0	DATAPL0	Read data port lsb0(7~0 bit)
0	0	1	DATAPL1	Read data port lsb1(15~8 bit)
0	1	0	DATAPM0	Read data port msb0(23~16 bit)
0	1	1	DATAPM1	Read data port msb1(31~24 bit)
1	0	0	STATP0	Read axis main status(7~0 bit)
1	0	1	STATP1	Read axis main status(15~8 bit)
1	1	0	UIOP0	Read UIO0~UIO7 terminal
1	1	1	UIOP1	Read UIO8~UIO11 terminal and axis sub status.

4) 16-bit interface(mode8_16 = '1')

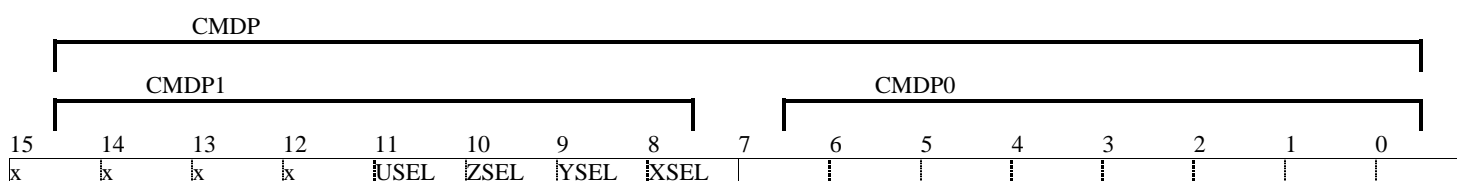
A2	A1	PORT assign	Description
0	0	DATAPL	Read data port lsb(15~0 bit)
0	1	DATAPM	Read data port msb(31~16 bit)
1	0	STATP	Read axis status(15~0 bit).
1	1	UIOP	Read UIO0~UIO11 terminal and sub status.

4. Port description

4.1. DATA PORT(DATAP)

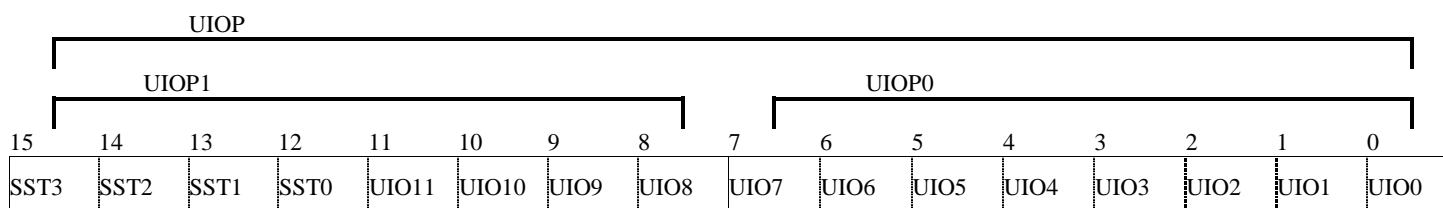


4.2. COMMAND WRITE PORT(CMDP)



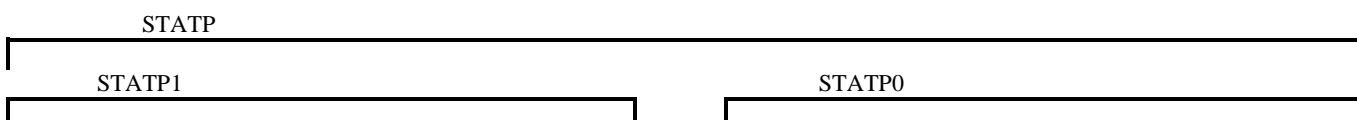
NAME	BIT	Description
XSEL	D11	U axis assignment
YSEL	D10	Z axis assignment
ZSEL	D9	Y axis assignment
USEL	D8	X axis assignment

4.3. Universal In/Out Write/Read PORT(UIOP)



AXIS	assign	bit	Description
XAXIS	XSST3	D15	Axis sub status 3 : Interrupt generated in U axis.
	XSST2	D14	Axis sub status 2 : Interrupt generated in Z axis.
	XSST1	D13	Axis sub status 1 : Interrupt generated in Y axis.
	XSST0	D12	Axis sub status 0 : Interrupt generated in X axis.
YAXIS	YSST3	D15	Axis sub status 3 : Trigger position queue of U axis is full.
	YSST2	D14	Axis sub status 2 : Trigger position queue of Y axis is full.
	YSST1	D13	Axis sub status 1 : Trigger position queue of Y axis is full.
	YSST0	D12	Axis sub status 0 : Trigger position queue of X axis is full.
ZAXIS	ZSST3	D15	Axis sub status 3 : Error stop condition is met in U axis.
	ZSST2	D14	Axis sub status 2 : Error stop condition is met in Z axis.
	ZSST1	D13	Axis sub status 1 : Error stop condition is met in Y axis.
	ZSST0	D12	Axis sub status 0 : Error stop condition is met in X axis.
UAXIS	USST3	D15	Axis sub status 3 : Current speed is same with OBJ and constant speed driving in U axis.
	USST2	D14	Axis sub status 2 : Current speed is same with OBJ and constant speed driving in Z axis.
	USST1	D13	Axis sub status 1 : Current speed is same with OBJ and constant speed driving in Y axis.
	USST0	D12	Axis sub status 0 : Current speed is same with OBJ and constant speed driving in X axis.

4.4. Main status Read PORT(STATP)



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MST15	MST14	MST13	MST12	MST11	MST10	MST9	MST8	MST7	MST6	MST5	MST4	MST3	MST2	MST1	MST0

NAME	BIT	Description
MST15	D15	Previous register queue is full.
MST14	D14	Previous register queue is empty.
MST13	D13	Script 4 is occupied.
MST12	D12	Script 3 is occupied.
MST11	D11	Script queue2 is full.
MST10	D10	Script queue1 is full.
MST9	D9	Captured data 4 is latched.
MST8	D8	Captured data 3 is latched.
MST7	D7	Captured data queue 2 is full.
MST6	D6	Captured data queue 1 is full.
MST5	D5	ALARM input level is activated.
MST4	D4	NLM input level is activated.
MST3	D3	PLM input level is activated.
MST2	D2	Command busy[access busy].
MST1	D1	Drive data is ready or drive pause..
MST0	D0	Drive busy.

5. Command description

5.1.Command 실행 방법

CAMC-QI의 명령어는 크게 읽기와 쓰기 명령어로 이루어 지는데 특정 레지스터 값을 읽어 들이기(code 0x00~0x7F 영역) 위해서는 먼저 읽기 명령어를 실행(Command port)한 후 데이터 포트(DATAPL0, DATAPL1, DATAPM0, DATAPM1port)를 읽어 필요한(유효한) 비트만큼 잘라 확인하면 되고, 레지스터에 값을 설정(code 0x80~0xFF 영역)하기 위해서는 필요한 데이터를 데이터 포트에 기입 후 쓰기 명령어를 실행하면 수행된다. 다음은 명령어 수행 방법 순서를 보여준다.

▶ 레지스터 쓰기 명령어 실행 순서 및 타이밍 차트

기본 적인 레지스터 쓰기 동작은 아래의 순서도에서 볼 수 있듯이 레지스터 값을 8-bit 또는 16-bit 단위로 분할하여 비트 위치에 맞는 데이터 포트에 기입한 후 레지스터 쓰기 명령어를 명령어 포트에 기입하여 이루어 진다. 레지스터의 크기에 따라 상위 데이터 포트에는 별도의 데이터를 쓰지 않아도 되며, 데이터 포트 기입 순서는 의미가 없으나 각 비트 위치에 주의 하여야 한다.

32-bit 레지스터 기준으로 16bit 모드로 사용할 경우 설정 시간이 반으로 줄어든다. 8bit 모드로 사용할 경우 다축 동시 수행으로 명령을 수행하기 위해서는 CMDP0에 명령어를 쓰기에 앞서 CMDP1에 명령어를 수행할 축에 대한 정보를 기입해 두어야 한다. 다축 동시 쓰기 명령어 수행 시 사용되는 데이터 값은 각 축의 데이터 포트 내용이기 때문에 동시 수행 명령어에 앞서 각 축의 데이터 포트에 원하는 값을 기입해 두어야 한다. 한번 설정된 CMDP1은 계속 유지되기 때문에 이후 실행할 명령어가 단축 실행이라면 CMDP1를 0x00로 초기화 해야 한다.

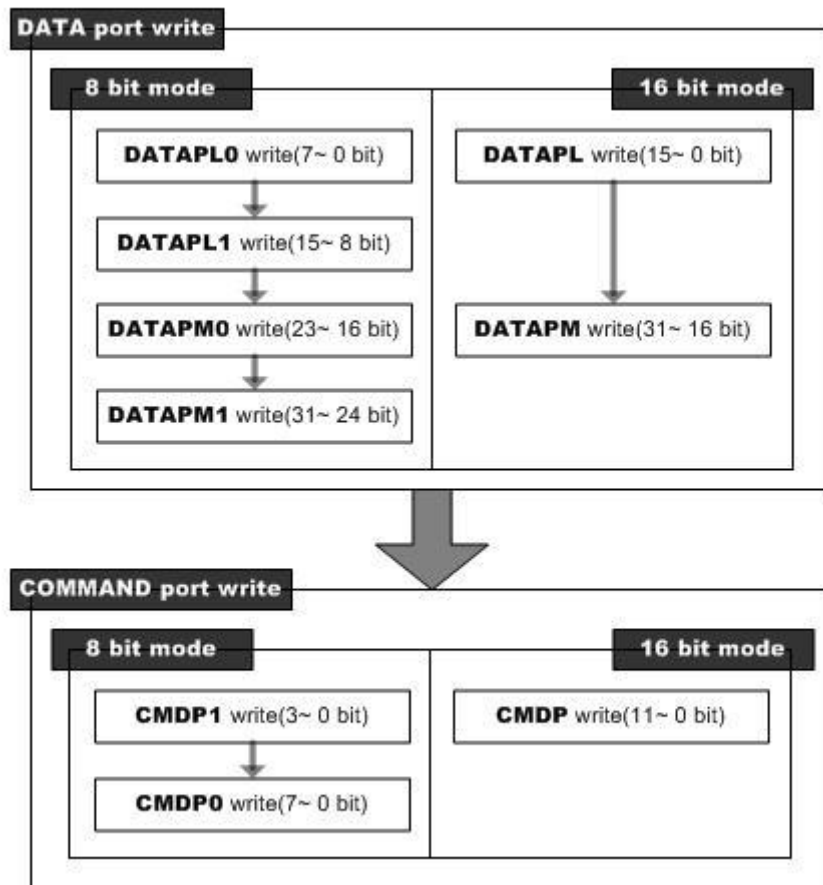


그림 5-1. 레지스터 쓰기 순서도

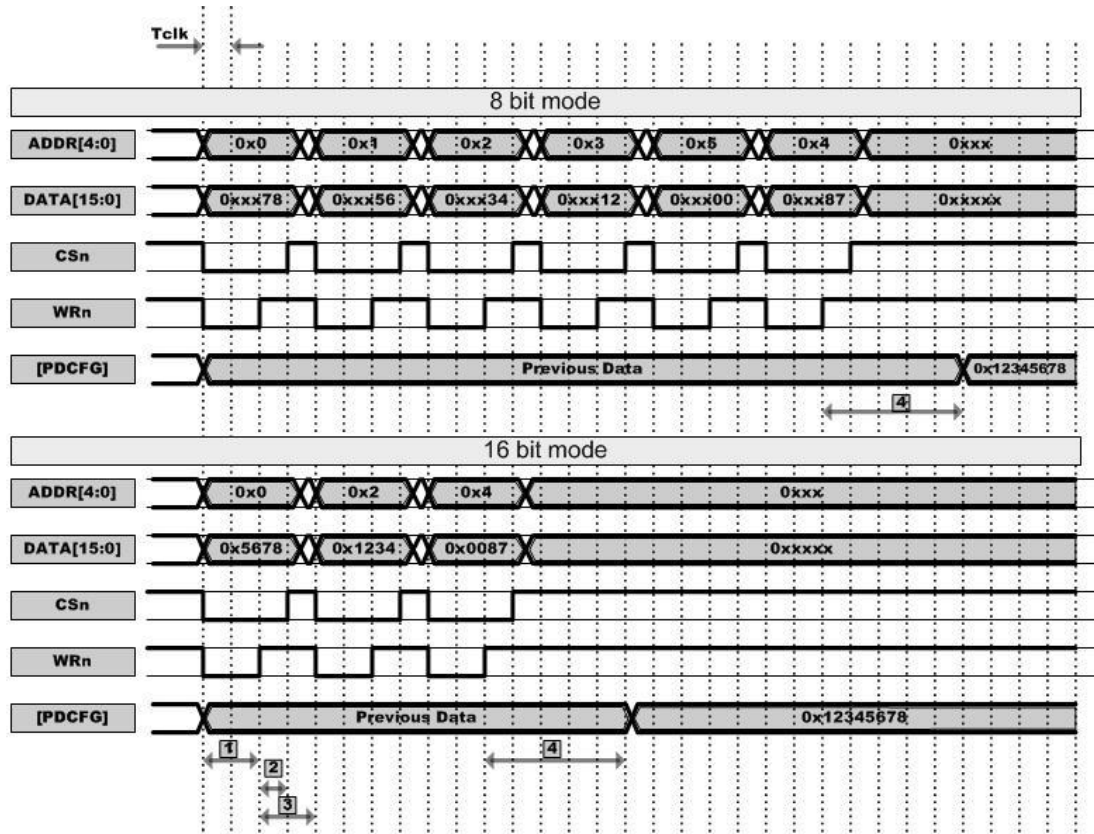


그림 5-2. [PDCFG] 쓰기 타이밍 차트

그림 5-2에서 X 축의 [PDCFG] 레지스터에 0x12345678을 기입하기 위한 각 입력 터미널의 Tclk(CLK 입력 신호 주기)에 대한 타이밍 차트이다. 그림 5-2에서 표시된 최소 시간은 다음과 같다.

Index	Description	Width
1	Write strobe 최소 폭	15 nSec
2	WRn to CSn 최소 delay, WRn to ADDR/DATA bus hold 최소 시간	0 nSec
3	WRn Hold 최소 폭	2 * Tclk
4	명령어 실행 소요 시간	5 * Tclk

CMDP0 기입 후 DATAP에 기입된 0x12345678 값이 [PDCFG] 레지스터에 반영되는 데 5*Tclk가 소요되며 이 시간 동안은 CMDP0에 기입되는 내용은 무시된다. 즉 연속적인 명령어 수행 시 최소 5*Tclk이상의 Delay가 필요하며, 고속의 CPU가 CAMC-QI를 액세스 할 경우 STATP[2]의 값을 읽어 보고 '0'이면 다음 명령어를 실행하면 된다.

▶ 레지스터 읽기 명령어 실행 순서 및 타이밍 차트

기본적인 레지스터 읽기 동작은 아래의 순서도에서 볼 수 있듯이 읽기 명령어를 수행한 후 그 결과를 데이터 포트로부터 읽어 각 비트 값을 합치는 순서로 완료된다. 레지스터의 크기에 따라 상위 데이터 포트 데이터는 읽지 않아도 되며, 데이터 포트 읽는 순서는 의미가 없으나 각 비트 위치에 주의 하여야 한다.

32-bit 레지스터 기준으로 16bit 모드로 사용할 경우 설정 시간이 더 작게 소요된다. 8bit 모드로 사용할 경우 다축 동시 수행으로 명령을 수행하기 위해서는 **CMDP0**에 명령어를 쓰기에 앞서 **CMDP1**에 명령어를 수행할 축에 대한 정보를 기입해 두어야 하며 읽기 명령어 결과는 각 축의 데이터 포트에 반영된다. 즉 각 축의 특정 레지스터 값을 동일한 시점에서 읽어보기 위해 동시 읽기 명령어를 수행하여 그 결과를 각 축의 데이터 포트를 읽어서 확인한다. 한번 설정된 **CMDP1**은 계속 유지되기 때문에 이후 실행할 명령어가 단축 실행이라면 **CMDP1**을 0x00로 초기화 해야 한다.

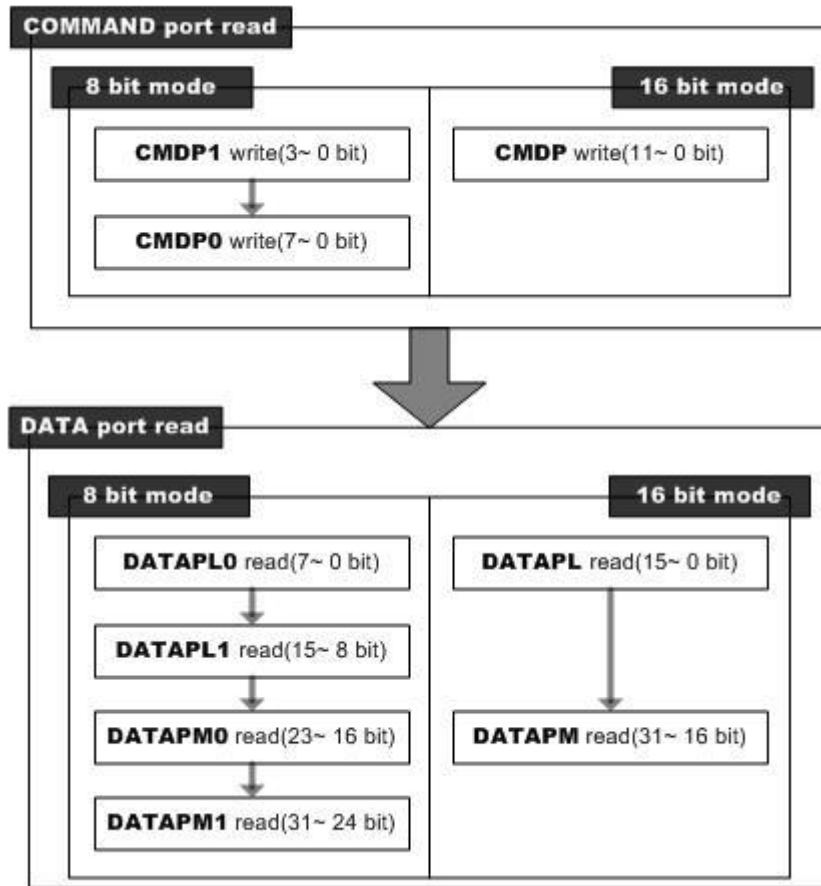


그림 5-3. 레지스터 읽기 순서도

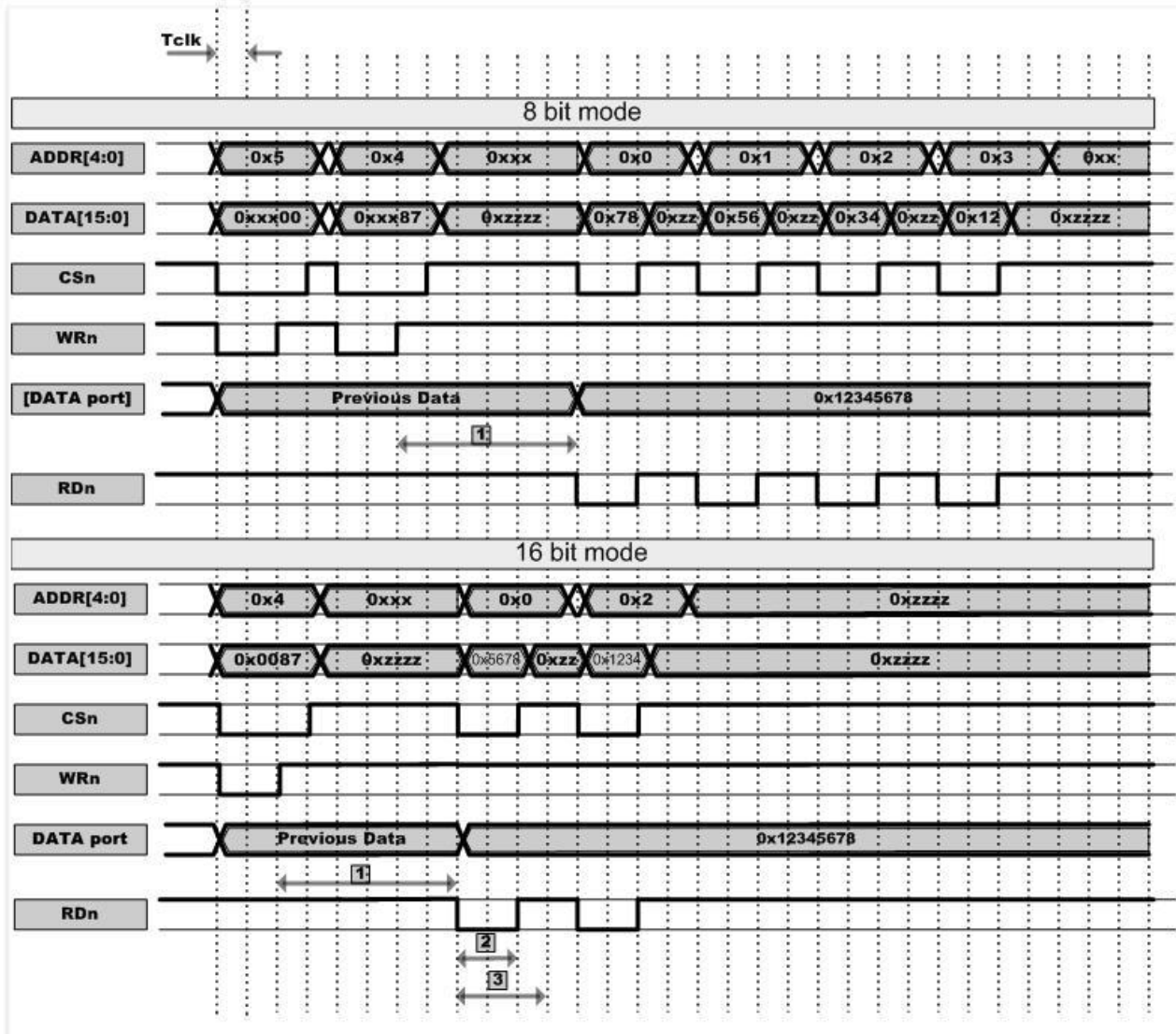


그림 5-4. [PDCFG] 읽기 명령어

그림 5-4에서 [PDCFG] 레지스터 읽기를 위한 각 입력 터미널의 Tclk(CLK 입력 신호 주기)에 대한 타이밍 차트이다. 그림 5-4에서 표시된 최소 시간은 다음과 같다.

Index	Description	Width
1	Read 명령어 소요 시간	6 * Tclk
2	RDn 최소 폭	15 nSec
3	ADDR 버스 유지 최소 폭	0 nSec

CMDP0 기입 후 DATAP에 [PCFG]의 값 0x12345678가 래치되는데 6*Tclk가 소요되며 이 시간 이후에 DATAP를 읽어야 된다. 또한 읽기 명령어에 소용되는 시간 동안 다른 명령어가 입력되어도 무시되며, 고속의 CPU가 CAMC-QI를 액세스 할 경우 STATP[2]의 값을 읽어 보고 '0'이면 DATAP 값을 읽거나 다른 명령어를 실행하면 된다.

5.2.Command list

각 축에 레지스터 관련, 드라이버 구동 관련, 드라이버 제거 관련, 기능 설정 관련, 스크립트 기능 설정 관련, 카운터 설정 관련 명령들이 제공된다.

다음의 표에서 전체 COMMAND의 일람표를 표시한다.

CODE (Read/Write)	CONTENTS(Previous registers/etc)	W	DEFAULT
0x00/0x80	Previous Speed magnitude data [PRANGE]	16	0xEA5F
0x01/0x81	Previous Start/Stop speed data [PSTD]	16	0x1
0x02/0x82	Previous Object speed data [POBJ]	16	0x1
0x03/0x83	Previous Acceleration rate data [PRATE1]	16	0xFFFF
0x04/0x84	Previous Deceleration rate data [PRATE2]	16	0xFFFF
0x05/0x85	Previous S drive region during acceleration [PSW1]	15	0x7FFF
0x06/0x86	Previous S drive region during deceleration [PSW2]	15	0x7FFF
0x07/0x87	Previous Drive configure [PDCFG]	32	0x0
0x08/0x88	Previous Slow down/rear pulse amount data [PREAR]	28	0x0
0x09/0x89	Previous Drive pulse amount data, Interpolation end position [PPOS]	28	0x0
0x0A/0x8A	Previous Circular Interpolation center, Master axis target position for multiple chip linear interpolation [PCENT]	28	0x0
0x0B/0x8B	Previous Interpolation step number [PISNUM]	28	0x0
/0x8C	Clear previous registers [CLRPRE]		
/0x8D	Shift previous register data [POPPRE]		
0x0E/0x8E	Restore data ports/Backup data ports [PORTMA]	32	0x0
0x0F	Current speed [CURSPD]	16	0x0

CODE (Read/Write)	CONTENTS(Working registers)	W	DEFAULT
0x10/0x90	Working Speed magnitude data [RANGE]	16	0x0
0x11/0x91	Working Start/Stop speed data [STD]	16	0x0
0x12/0x92	Working Object speed data [OBJ]	16	0x0
0x13/0x93	Working Acceleration rate data [RATE1]	16	0x0
0x14/0x94	Working Deceleration rate data [RATE2]	16	0x0
0x15/0x95	Working S drive region during acceleration [SW1]	15	0x0
0x16/0x96	Working S drive region during deceleration [SW2]	15	0x0
0x17/0x97	Working Drive configure [DCFG]	32	0x0
0x18/0x98	Working Slow down/rear pulse amount data [REAR]	28	0x0
0x19/0x99	Working Drive pulse amount data, Interpolation end position [POS]	28	0x0
0x1A/0x9A	Working Circular Interpolation center, Master axis target position for multiple chip linear interpolation [CENT]	28	0x0
0x1B/0x9B	Working Interpolation step number [ISNUM]	28	0x0
0x1C/	Remain pulse data after stopping preset drive function abnormally [REMAIN]	28	0x0
0x1F/0x9F	Original search object speed [OBJORG]	16	0x1

CODE(HEX)	CONTENTS (Universal in/out setting)	W	DEFAULT
0x1D/0x9D	Universal in/out terminal mode[Uiom] ('0' : input, '1' : output)	12	0x01F
0x1E/0x9E	Universal in/out value[Uio]	15	0x0000

CODE(HEX)	CONTENTS (Drive start command)
/0xA0	Normal profile mode drive start(STD→OBJ→STD).[STRN]
/0xA1	Start at OBJ profile mode drive start.(OBJ→STD).[STRO]
/0xA2	Constant speed profile #1 drive start.(OBJ).[STRCO]
/0xA3	Constant speed profile #2 drive start.(STD).[STRCS]
/0xDC	Normal profile mode drive start(STD→OBJ→STD).[ASTRN] (with DCFG 7~0 bit data in data port 0)
/0xDD	Start at OBJ profile mode drive start.(OBJ→STD).[ASTRO] (with DCFG 7~0 bit data in data port 0)
/0xDE	Constant speed profile #1 drive start.(OBJ).[ASTRCO] (with DCFG 7~0 bit data in data port 0)
/0xDF	Constant speed profile #2 drive start.(STD).[ASTRCS] (with DCFG 7~0 bit data in data port 0)

CODE(HEX)	CONTENTS (Drive control command)
/0xA4	Slow Down stop.[SSTOP]

/0xA5	Immediately stop.[STOP]
/0xA6	Output one shot of the start pulse from SQSTR1 terminal.[SQRO1]
/0xA7	Output one shot of the start pulse from SQSTR2 terminal.[SQRO2]
/0xA8	Execution sync start function same as SQSTR1 input.[SQRI1]
/0xA9	Execution sync start function same as SQSTR2 input.[SQRI2]
/0xAA	Output one shot of the stop pulse form SQSTP1 terminal.[SQPO1]
/0xAB	Output one shot of the stop pulse form SQSTP2 terminal.[SQPO2]
0x2C/	Interpolation step counter.[ISCNT]
0x2D/	Interpolation step counter for advanced deceleration mode. [ISACNT]
/0xAE	Emergency stop all axis.[ESTOP]
/0xAF	Software reset.[SWRESET]
/0xB0	Driven pulse amount during last driving(Interpolation step counter for path move).[DRPCNT]
/0xB1	Interrupt generation command.[INTGEN]

CODE(HEX) Read/Write	CONTENTS (Peripheral function setting)	W	DEFAULT
/0xB2	Trigger queue POP[TRGQPOP]		
0x33/0xB3	Trigger/Timer configure[TRTMCF]	10	0x0F0
0x34/0xB4	Software negative limit position.[SNLMT]	28	0x8000000
0x35/0xB5	Software positive limit position. [SPLMT]	28	0x7FFFFFFF
0x36/0xB6	Trigger pulse width. [TRGPW]	32	0xFFFFFFFF00
0x37/0xB7	Trigger function start position.[TRGSP]	28	0x8000000
0x38/0xB8	Trigger function end position.[TRGEP]	28	0x7FFFFFFF
0x39/0xB9	Push trigger position or period data to queue.[PTRGPOS]	28	0x0
/0xBA	Clear trigger position or period queue.[CLRTRG]		
/0xBB	Generate one shot trigger pulse.[TRGGEN]		
0x3C/0xBC	Timer 1 period[TMRP1]	28	0x0
0x3D/0xBD	Timer 2 period[TMRP2]	28	0x0
0x3E/0xBE	Timer1 stop/start!.[TMR1GEN]		
0x3F/0xBF	Timer2 stop/start!.[TMR2GEN]		
0x60/0xE0	ERC reset/set[ERCRS]		

CODE(HEX) (Read/Write)	CONTENTS (Script1/2/3 setting registers)	W	DEFAULT
0x40/0xC0	Script1 control queue register[SCRCON1]	25	0x0
0x41/0xC1	Script1 command queue register[SCRCMD1](0xCF for NOOP)	32	0x0
0x42/0xC2	Script1 execution data queue register[SCRDAT1]	32	0x0
0x43/	Script1 captured data queue register(top of depth 15 queue)[CQ1]	32	0x0
0x44/0xC4	Script1 flag control register.[SCRCFG1]	16	0xF0F0
0x45/0xC5	Script2 control queue register[SCRCON2]	32	0x0
0x46/0xC6	Script2 command queue register[SCRCMD2]	32	0x0
0x47/0xC7	Script2 execution data queue register[SCRDAT2]	32	0x0
0x48/	Script2 captured data queue register(top of depth 16 queue)[CQ2]	32	0x0
0x49/0xC9	Script2 flag control register.[SCRCFG2]	16	0xF0F0
0x4A/0xCA	Script3 control register[SCRCON3]	32	0x0
0x4B/0xCB	Script3 command register[SCRCMD3]	32	0x0
0x4C/0xCC	Script3 execution data register[SCRDAT3]	32	0x0
0x4D/	Script3 captured data register(top of depth 16 queue)[CQ3]	32	0x0
0x4E/0xCE	Don't care.		
/0xCF	[No operation code for Script reservation command].		

CODE(HEX) (Read/Write)	CONTENTS (Script4 setting/Script status registers)	W	DEFAULT
0x50/0xD0	Script4 control register[SCRCON4]	32	0x0
0x51/0xD1	Script4 command register[SCRCMD4]	32	0x0
0x52/0xD2	Script4 execution data register[SCRDAT4]	32	0x0
0x53/	Script4 captured data register [CQ4]	32	0x0
0x54/0xD4	Target source data setting.[SCRTG]	16	0x0
0x55/	Script status #1[SCRSTAT1]	31	0x00005555
0x56/	Script status #2[SCRSTAT2]	15	0x0000
/0xD7	Initialize script queues with target selection.[INITSQ]	4	

CODE(HEX) (Read/Write)	CONTENTS (Script4 setting/Script status registers)	W	DEFAULT
/0xD8	Initialize captured data queue with target selection.[INITCQ]	4	
0x59/0xD9	Set enable mode with target selection.[SCRM]	4[9]	0x0F
/0xDA	Pop script #1 queue.[SQ1POP]		
/0xDB	Pop script #2 queue.[SQ2POP]		

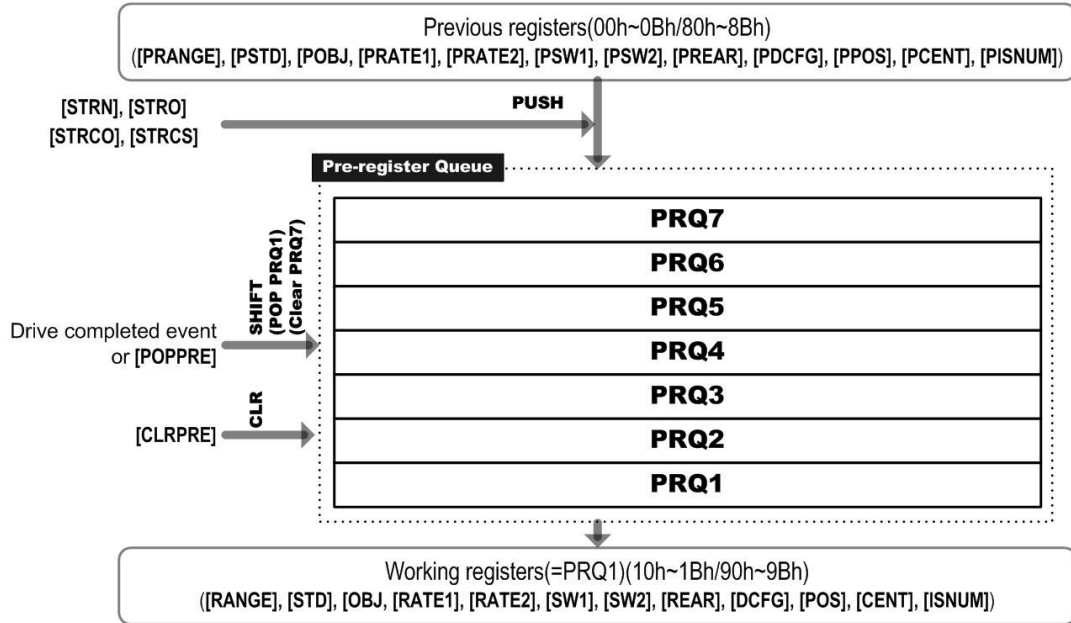
CODE(HEX) (Read/Write)	CONTENTS (Counter function registers)	W	DEFAULT
0x61/0xE1	Counter lower bound data[CNTLB]	28	0x8000000
0x62/0xE2	Counter upper bound data[CNTUB]	28	0x7FFFFFFF
0x63/0xE3	Counter configure #1 [CNTCF1]	32	0x0
0x64/0xE4	Counter configure #2 [CNTCF2]	32	0x0
0x66/0xE6	Counter #1 data[CNT1]	28	0x0
0x67/0xE7	Counter #2 data[CNT2]	28	0x0
0x68/0xE8	Counter #3 data[CNT3]	28	0x0
0x69/0xE9	Counter #4 data[CNT4]	28	0x0
0x6A/0xEA	Counter #5 data[CNT5]	28	0x0
0x6B/0xEB	Counter #1 comparator data[CNTC1]	28	0x0
0x6C/0xEC	Counter #2 comparator data[CNTC2]	28	0x0
0x6D/0xED	Counter #3 comparator data[CNTC3]	28	0x0
0x6E/0xEE	Counter #4 comparator data[CNTC4]	28	0x0
0x6F/0xEF	Counter #5 comparator data[CNTC5]	28	0x0

CODE(HEX) (Read/Write)	CONTENTS (Configure/Status registers)	W	DEFAULT
0x70/0xF0	Configure register #1[UCFG1]	29	0x0
0x71/0xF1	Configure register #2[UCFG2]	29	0x0
0x72/0xF2	Configure register #3[UCFG3]	32	0x0
0x73/0xF3	Configure register #4[UCFG4]	32	0x0
0x77/0xF7	Interrupt bank #1 mask[IMASK1]	32	0x0
0x78/0xF8	Interrupt bank #2 mask[IMASK2]	32	0x0
0x79/0xF9	Status register #1[STAT1]/End status clear[ESCLR]	32	0x0
0x7A	Status register #2[STAT2]	32	0x0
0x7B	Status register #3[STAT3]	32	0x0
0x7C	Status register #4[STAT4]	32	0x0
0x7D	Status register #5[STAT5]	32	0x0
0x7E/0xFE	Interrupt bank#1 flag/interrupt flag#1 clear[IFLAG1]	32	0x0
0x7F/0xFF	Interrupt bank#2 flag/interrupt flag#2 clear[IFLAG2]	32	0x0

5.3. 명령어 상세 설명 및 기능 설명

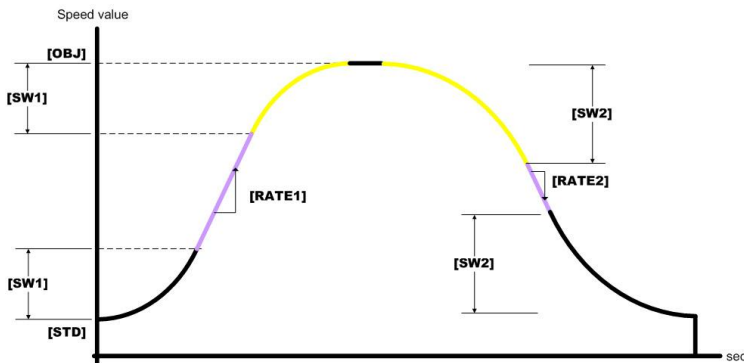
5.3.1.Commands for setting register about drive

CAMC-QI는 속도 프로파일 생성과 관련된 레지스터, 보간 드라이버 관련 설정 레지스터, 지정 위치 관련 설정 레지스터는 최대 7개까지 예약 할 수 있다. 이는 Previous 레지스터에 예약할 값들을 기입하고 드라이브 시작 명령어를 실행 함으로써 예약 Queue에 적체 되며, 적체된 Queue의 PRQ1이 현재 실행 중이거나 실행 대기하고 있는 Working 레지스터 값이 된다. 다음의 그림은 레지스터 예약 및 실행에 관한 기능도를 보여준다.



위의 그림과 같이 설정되는 값을 사용하여 드라이브 시작 조건이 만족하면 CAMC-QI는 드라이브 시작 명령의 종류에 따라 속도 프로파일을 생성한다. 이때 사용되는 레지스터들의 종류와 적용 형태는 다음과 같다.

Registers	Description	Width	Unit
[(P)STD]	Start/Stop speed	16	nS
[(P)OBJ]	Object speed	16	nS
[(P)RATE1]	Acceleration rate	16	nS
[(P)RATE2]	Deceleration rate	16	nS
[(P)SW1]	S-curve Acc. range	15	nS
[(P)SW2]	S-curve Dec. range	15	nS



▶ **[(P)RANGE][0x00(0x10)/0x80(0x90)]: 내부 속도 정보에 대한 출력 주파수 배율을 설정하는 값(16 bit).**

설정 범위는 3~65,535(0x0003~0xFFFF)이며, Reset 후 초기 값은 5999(0xEA5F)이다. 다음의 식에 의해 출력 펄스의 주파수 배율이 결정된다.

$$Frequency\ resolution = \frac{Reference\ clock\ frequency\ [Hz]}{[RANGE] \times 65536}$$

[RANGE] 값으로 정해진 주파수 배율로 생성할 수 있는 출력 펄스 주파수는 다음의 예와 같다.(reference clock = 39.3216MHz)

[RANGE]	배율	출력 주파수 범위[pps]	[RANGE]	배율	출력 주파수 범위[pps]
3	200	200~13,107,200	299	2	2~131,070
5	100	100~6,553,500	599	1	1~65,535
11	50	50~3,276,750	1199	0.5	0.5~32,767.5
23	25	25~1,638,375	2999	0.2	0.2~13,107
59	10	10~655,350	5999	0.1	0.1~6,553.5
119	5	5~327,675	59999	0.01	0.01~655.35

▶ **[(P)STD][0x01(0x11)/0x81(0x91)]: 속도 프로파일 생성시 시작/종료 속도 설정 값(16 bit).**

설정범위는 1~65,535(0x0001~0xFFFF)이며 RESET 후 기본값은 0(0x0000)이다. 시작/종료 속도 값으로 정해지는 출력 펄스 속도는 다음의 식과 같이 구해진다.

$$Start / Stop\ speed\ [pps] = [STD] \times \frac{Reference\ clock\ frequency\ [Hz]}{[RANGE] \times 65536}$$

▶ **[(P)OBJ][0x02(0x12)/0x82(0x92)]: 속도 프로파일 생성시 목표 속도 설정 값 (16 bit).**

설정범위는 1~65,535(0x0001~0xFFFF)이며 RESET 후 기본값은 0(0x0000)이다. 목표 속도 값으로 정해지는 출력 펄스 속도는 다음의 식과 같이 구해진다.

$$Object\ speed\ [pps] = [OBJ] \times \frac{Reference\ clock\ frequency\ [Hz]}{[RANGE] \times 65536}$$

▶ **[(P)RATE1][0x03(0x13)/0x83(0x93)]: 속도 프로파일 생성시 가속 율 설정 값 (16 bit).**

설정범위는 1~65,535(0x0001~0xFFFF)이며 RESET 후 기본값은 0(0x0000)이다. 가속 율로 정해지는 가속 시간은 다음의 식과 같이 구해진다. [(P)RATE2]값이 0이면 [(P)RATE1]값은 감속 율로도 사용된다.

--- 직선 가감속 사용시 가속 시간

$$Acceleration\ time(sec) = \frac{(OBJ - STD) \times [RATE1] \times 4}{Reference\ clock[Hz]}$$

--- S자 가감속(Full-S) 사용시 가속 시간

$$Acceleration\ time(sec) = \frac{(OBJ - STD) \times [RATE1] \times 8}{Reference\ clock[Hz]}$$

--- S자 가감속(Partial-S) 사용시 가속 시간

$$Acceleration\ time(sec) = \frac{(OBJ - STD + 2SW1) \times [RATE1] \times 4}{Reference\ clock[Hz]}$$

▶ **[(P)RATE2][0x04(0x14)/0x84(0x94)]: 속도 프로파일 생성시 감속 율 설정 값 (16 bit).**

설정범위는 0~65,535(0x0000~0xFFFF)이며 RESET 후 기본값은 0(0x0000)이다. 감속 율로 정해지는 가속 시간은 다음의 식과 같이 구해진다. [(P)RATE2]값이 0으로 설정되면 감속 율을 [(P)RATE1]을 사용하여 대칭 가감속 프로파일을 생성한다.

--- 직선 가감속 사용시 가속 시간

$$Deceleration\ time(sec) = \frac{(OBJ - STD) \times [RATE2] \times 4}{Reference\ clock[Hz]}$$

--- S자 가감속(Full-S) 사용시 가속 시간

$$Deceleration\ time(sec) = \frac{(OBJ - STD) \times [RATE2] \times 8}{Reference\ clock[Hz]}$$

--- S자 가감속(Partial-S) 사용시 가속 시간

$$Deceleration\ time(sec) = \frac{(OBJ - STD + 2SW1) \times [RATE2] \times 4}{Reference\ clock[Hz]}$$

▶ [(P)SW1][0x05(0x15)/0x85(0x95)]: 가속시 S-curve 범위 설정 값(15bit)

설정범위는 0~32,767(0x0000~0x7FFF)이며 RESET 후 기본값은 0(0x0000)이다. [(P)STD] 값에서 [(P)OBJ]로 가속하는 범위 내에서 S-curve 범위를 지정하는 값으로 속도 프로파일 모드가 S-curve로 [(P)DCFG] 값이 설정되었을 때 유효하다.

--- 가속시 S-curve 범위 속도

$$S_{atop}[pps] = [SW1] \times \frac{Reference\ clock\ frequency\ [Hz]}{[RANGE] \times 65536}$$

▶ [(P)SW2][0x06(0x16)/0x86(0x96)]: 감속시 S-curve 범위 설정 값(15bit)

설정범위는 0~32,767(0x0000~0x7FFF)이며 RESET 후 기본값은 0(0x0000)이다. [(P)SW2] 값이 0 일 때 감속시 S-curve 범위는 [(P)SW1]과 같게 설정된다. [(P)OBJ] 값에서 [(P)STD]로 감속하는 범위 내에서 S-curve 범위를 지정하는 값으로 속도 프로파일 모드가 S-curve로 [(P)DCFG] 값이 설정되었을 때 유효하다.

--- 감속시 S-curve 범위 속도

$$S_{dtop}[pps] = [SW2] \times \frac{Reference\ clock\ frequency\ [Hz]}{[RANGE] \times 65536}$$

▶ [(P)DCFG][0x07(0x17)/0x87(0x97)]: 드라이브 구동 모드 설정값(32bit)

드라이브 구동 모드를 설정하는 값으로 단위 구동에 대한 드라이브 종류/속도 프로파일/구동 시작 및 종료 옵션/펄스 출력 여부/구동 완료 조건 등을 설정한다.

Bits	Descriptions
6~0bit	"0010000": 비정상 종료시 남아있는 지정 펄스수 만큼 구동한다. "0010001": [POS] 설정값 만큼 구동한다. "0010010": [CNT1]이 [POS] 값이 되도록 구동한다. "0010011": [CNT2]이 [POS] 값이 되도록 구동한다. "0010100": [CNT1]이 0이 되도록 구동한다. "0010101": [CNT2]이 0이 되도록 구동한다. "0010110": CW 방향으로 1 펄스 구동 "0010111": CCW 방향으로 1 펄스 구동 "0011000": Reserved. "0011001": Reserved. "0011010": X 축과 동일한 펄스 출력(Y,Z,U only) "0011011": Y 축과 동일한 펄스 출력 (X,Z,U only) "0011100": Z 축과 동일한 펄스 출력 (X,Y,U only) "0011101": U 축과 동일한 펄스 출력 (X,Y,Z only) "0011110": Reserved. "0011111": Reserved. "0100000": CW 방향으로 원점 복귀 구동. "0100001": CCW 방향으로 원점 복귀 구동. "0100010": CW 방향으로 원점 이탈 구동.

Bits	Descriptions
	<p>“0100011”: CCW 방향으로 원점 이탈 구동.</p> <p>“0100100”: CW 방향으로 원점 검색 구동. ([UCFG1](28~25) < “0101” 일 때 한정)</p> <p>“0100101”: CCW 방향으로 원점 검색 구동. ([UCFG1](28~25) < “0101” 일 때 한정)</p> <p>“0100110”: CW 방향으로 선택 신호 검색 후 감속 정지 구동. ([UCFG2](7~5)에서 신호 선택)</p> <p>“0100111”: CCW 방향으로 선택 신호 검색 후 감속 정지 구동. ([UCFG2](7~5)에서 신호 선택)</p> <p>“0101000”: CW 방향으로 선택 신호 검색 후 급 정지 구동. ([UCFG2](7~5)에서 신호 선택)</p> <p>“0101001”: CCW 방향으로 선택 신호 검색 후 급 정지 구동. ([UCFG2](7~5)에서 신호 선택)</p> <p>“0101010”: Reserved.</p> <p>“0101011”: Reserved.</p> <p>“0101100”: Reserved.</p> <p>“0101101”: Reserved.</p> <p>“0101110”: Reserved.</p> <p>“0101111”: Reserved.</p> <p>“1001000”: CW 방향으로 연속 구동.</p> <p>“1001001”: CCW 방향으로 연속 구동.</p> <p>“1001010”: Reserved.</p> <p>“1001011”: Reserved.</p> <p>“1001100”: Reserved.</p> <p>“1001101”: Reserved.</p> <p>“1001110”: Reserved.</p> <p>“1001111”: Reserved.</p> <p>“1010000”: 외부 펄스에 의한 연속 구동.</p> <p>“1010001”: 외부 펄스에 의한 [POS] 거리 구동.</p> <p>“1010010”: 외부 펄스에 의한 [CNT1]이 [POS]가 되도록 구동.</p> <p>“1010011”: 외부 펄스에 의한 [CNT2]이 [POS]가 되도록 구동.</p> <p>“1010100”: 외부 펄스에 의한 [CNT1]이 0 가 되도록 구동.</p> <p>“1010101”: 외부 펄스에 의한 [CNT2]이 0 가 되도록 구동.</p> <p>“1010110”: 외부 펄스에 의한 연속 직선 보간(한 칩 안에서)</p> <p>“1010111”: 외부 펄스에 의한 직선 보간(한 칩 안에서)</p> <p>“1011000”: 외부 펄스에 의한 연속 직선 보간(다른 칩간)</p> <p>“1011001”: 외부 펄스에 의한 직선 보간(다른 칩간)</p> <p>“1011010”: 외부 펄스에 의한 CW 방향 원호 보간.</p> <p>“1011011”: 외부 펄스에 의한 CCW 방향 원호 보간.</p> <p>“1011100”: 외부 펄스에 의한 CW 방향 연속 원호 보간.</p> <p>“1011101”: 외부 펄스에 의한 CCW 방향 연속 원호 보간.</p> <p>“1011110”: Reserved.</p> <p>“1011111”: Reserved.</p> <p>“1100000”: 연속 직선 보간(한 칩 안에서)</p> <p>“1100001”: 직선 보간(한 칩 안에서)</p> <p>“1100010”: 연속 직선 보간(다른 칩간)</p> <p>“1100011”: 직선 보간(다른 칩간)</p> <p>“1100100”: CW 원호보간.</p> <p>“1100101”: CCW 원호보간</p> <p>“1100110”: U 축 출력 펄스를 기준으로 한 CW 원호 보간.</p> <p>“1100111”: U 축 출력 펄스를 기준으로 한 CCW 원호 보간.</p> <p>“1101000”: CW 연속 원호보간.</p> <p>“1101001”: CCW 연속 원호보간.</p> <p>“1101010”: Reserved.</p> <p>“1101011”: Reserved.</p> <p>“1101100”: Reserved.</p> <p>“1101101”: Reserved.</p> <p>“1101110”: Reserved.</p> <p>“1101111”: Reserved.</p> <p>“1111111”: 인포지션 기능을 활성화 한다.(단위 구동이 인포지션 기능을 사용할 경우 사용)</p>
7 bit :	구동 종료 후 인터럽트를 발생 시킨다.
8 bit	가감속 속도 프로파일 모드를 설정한다.(0 : 직선 프로파일, 1 : S-curve 속도 프로파일)
9 bit	지정 거리 드라이브시 감속 방법을 설정한다.(0 : 자동 감속, 1 : 남은 펄스가 [PREAR] 이하일 때 감속시작)
11~10 bit	드라이브 구동 시작 방법

Bits	Descriptions
	“00”: 바로 시작. “01”: SQSTR1 입력 신호 입력 이후 시작. “10”: SQSTR2 입력 신호 입력 이후 시작. “11”: 구동 조건 설정 만족 시 시작.
15~12bit	드라이브 구동 시작 조건 설정(드라이브 구동 시작 방법을 조건 시작으로 설정 시 유효) “0000”: [SQRII][0xA8] 명령어 입력 “0001”: X 축 동기 시작 이벤트 발생, 이벤트 카운터 선택 설정#1,#2 공유([CNTCF1][31~16]으로 설정) “0010”: Y 축 동기 시작 이벤트 발생, 이벤트 카운터 선택 설정#1,#2 공유([CNTCF1][31~16]으로 설정) “0011”: Z 축 동기 시작 이벤트 발생, 이벤트 카운터 선택 설정#1,#2 공유([CNTCF1][31~16]으로 설정) “0100”: U 축 동기 시작 이벤트 발생, 이벤트 카운터 선택 설정#1,#2 공유([CNTCF1][31~16]으로 설정) “0101”: X 축 정지. “0110”: Y 축 정지. “0111”: Z 축 정지. “1000”: U 축 정지. “1001”: X축 Y축 정지. “1010”: X축 Z축 정지. “1011”: X축 U축 정지. “1100”: Y축 Z축 정지. “1101”: Y축 U축 정지. “1110”: Z축 U축 정지.
17~16 bit	동기 정지 기능 사용 설정. “00”: 동기 정지 기능 사용하지 않음. “01”: SQSTP1 터미널에 신호 입력 시 정지. “10”: SQSTP2 터미널에 신호 입력 시 정지. “11”: SQSTP1 터미널 또는 SQSTP2 터미널에 신호 입력 시 정지.
19~18 bit	동기 정지 기능 핀 사용 설정. “00”: 사용하지 않음. “01”: 정지 시 SQSTP1 터미널에 동기 정지 신호 출력. “10”: 정지 시 SQSTP2 터미널에 동기 정지 신호 출력. “11”: 정지 시 SQSTP1/SQSTP2 터미널에 동기 정지 신호 출력.
20 bit	구동 중 [CNT1] 사용 유무(0: 사용함, 1: 사용하지 않음)
21 bit	구동 중 pulse 출력 여부(0: 출력 함, 1: 출력하지 않음(virtual mode))
22 bit	지정 펄스수 구동 시 거리 오버라이드가 현재 구동 완료 거리보다 작을 때 정지 후 반대 방향 구동 사용 여부(‘0’: 사용하지 않음, ‘1’: 사용함)
23 bit	지정 펄스수 구동 시 목표 속도 보상 기능 사용 유무(‘0’: 사용하지 않음, 1: 사용함)
24 bit	인포지션 기능 사용 유무(‘0’: 사용하지 않음, 1: 사용함)
25 bit	감속 정지 조건에서 목표 속도 보상 기능 사용 유무(‘0’: 사용하지 않음, 1: 사용함)
26 bit	구동 시작 시 해당 축의 축 선택 포트의 내용 중 X 축 선택 값이 설정되었음.
27 bit	구동 시작 시 해당 축의 축 선택 포트의 내용 중 Y 축 선택 값이 설정되었음.
28 bit	구동 시작 시 해당 축의 축 선택 포트의 내용 중 Z 축 선택 값이 설정되었음.
29 bit	구동 시작 시 해당 축의 축 선택 포트의 내용 중 U 축 선택 값이 설정되었음.
30 bit	[STRCO] 구동 시작 명령어 속도 프로파일 모드 설정.(‘0’: 감속 정지 사용, 1: 급정지)
31 bit	보간 중 선속 일정 사용 유무(‘0’: 사용 하지 않음, 사용함)

▶ [(P)REAR][0x08(0x18)/0x88(0x98)]: 지정 펄스 드라이브 수동 감속 위치/자동 감속시 남은 펄스 설정(28bit)

설정범위는 -134,217,728~134,217,727(0x8000000~0x7FFFFFFF)이며 RESET 후 기본값은 0(0x0000)이다. 지정 펄스수 드라이브시 드라이브 구동 시 수동 가감속 위치를 설정하거나 자동 가감속시 감속 위치의 읍셋을 조정한다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4bit는 27bit로 sign-extension 된다.

▶ [(P)POS][0x09(0x19)/0x89(0x99)]: 지정 펄스 드라이브 구동 펄스수/보간 종료 위치 설정(28bit)

설정범위는 -134,217,728~134,217,727(0x8000000~0x7FFFFFFF)이며 RESET 후 기본값은 0(0x0000)이다. 지정 펄스수 드라이브시 드라이브 펄스 수/절대 위치 값 및 보간 드라이브시 종료 좌표로 사용된다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4bit는 27bit로 sign-

extension 된다.

▶ **[(P)CENT][0x0A(0x1A)/0x8A(0x9A)] : 원호보간 중심 좌표 설정/칩간 직선보간시 장축 거리 설정(28bit)**

설정범위는 -134,217,728~134,217,727(0x8000000~0x7FFFFFFF)이며 RESET 후 기본값은 0(0x0000)이다. 원호보간시 해당 축의 중심 좌표를 설정하거나 1개 이상의 CAMC-QI를 사용하여 직선 보간을 할 때 장축(master) 거리를 설정한다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4bit는 27bit로 sign-extension 된다.

▶ **[(P)ISNUM][0x0B(0x1B)/0x8B(0x9B)] : 원호보간 중심 좌표 설정/칩간 직선보간시 장축 거리 설정(28bit)**

설정범위는 0~268,435,455(0x0~0xFFFFFFFF)이며 RESET 후 기본값은 0(0x0000)이다. 보간 드라이브시 가감속에 필요한 보간 스텝 개수를 설정한다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4bit는 '0'이다.

▶ **[CLRPRE][0x8C]: Previous register queue 내용을 초기화**

현재 설정되어 있는 Previous register queue의 내용을 초기화 한다.

▶ **[POPPRE][0x8D] : Previous register queue 내용을 갱신.**

현재 설정되어 있는 Previous register queue의 내용을 POP/SHIFT 한다.

▶ **[POPTMA][0x0E/8E] : 현재 설정되어 있는 DATAPM/DATAPL의 내용을 저장/복구**

현재 설정되어 있는 DATA PORT 내용을 저장하고 복구한다. 인터럽트 서비스 루틴 시작 시 저장하고 서비스 루틴을 빠져나갈 때 저장해둔 값으로 데이터 포트를 복구한다.

▶ **[CURSPD][0x0F] : 현재 구동중인 속도값(16bit)**

현재 구동 중인 속도값을 표시한다. 구동 방향과 관계 없이 값의 범위는 0~65535(0x0 ~ 0xFFFF)이다.

▶ **[REMAIN][0x1C] : 지정펄스 구동 시 비정상 종료로 남은 펄스 량(28 bit).**

남은 펄스 드라이브시 구동하는 펄스 값이 되며 범위는 -134,217,728~134,217,727(0x8000000~0x7FFFFFFF) 이다.

▶ **[OBJORG][0x1F/0x9F] : 원점 검색 시 사용되는 전용 목표 속도(16bit).**

원점 검색 시 사용되는 속도 값이며 범위는 0~65,535(0x0001~0xFFFF) 이다.

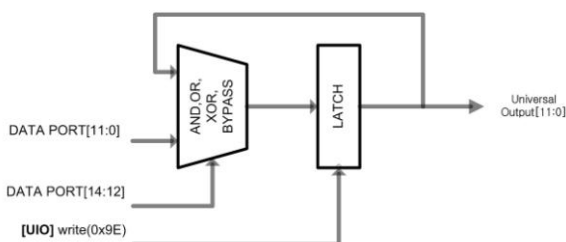
5.3.2.Universal in/out

▶ **[UIOM][0x1D/0x9D] : 범용 입출력 터미널 모드 설정(12bit).**

각 축 12 개의 범용입출력 터미널의 입력/출력 모드를 설정한다. Reset 후 0x01F로 범용 입출력 0~4는 출력으로 나머지는 입력으로 동작한다. 각 비트는 해당 입출력 터미널의 입력/출력 모드로 0 일 때 입력으로, 1 일 때 출력으로 동작한다.

▶ **[UIO][0x1E/0x9E] : 범용 입출력 값 설정 및 현재 상태 보기(12bit).**

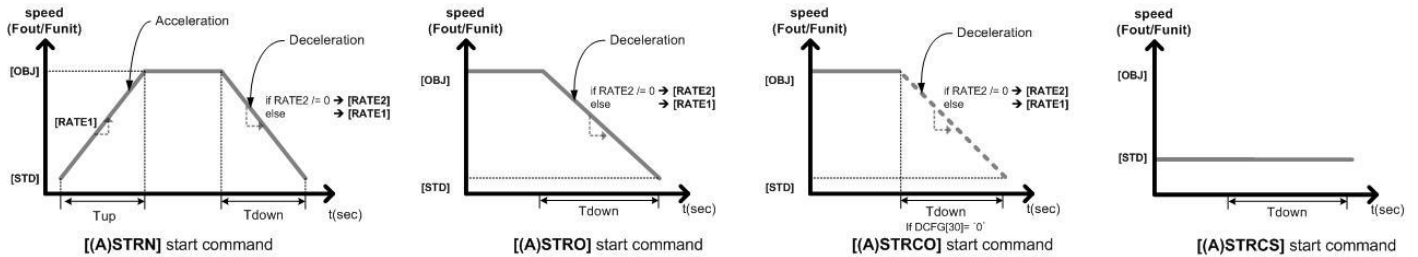
범용 입출력 터미널 값을 설정하고 현재 터미널 입/출력 값을 확인한다. [UIO] 쓰기 명령어는 범용 입출력 터미널 중 [UIOM] 에서 출력으로 설정된 것만 유효하다. 특히 데이터 포트 14~12 bit 내용에 따라 현재 범용 입출력 터미널 값과 입력된 데이터 포트간 연산을 통해서 범용 출력 값을 변경할 수 있다.



DATAP [14~12]	[UIO] write command
000	DATA port[11~0] → UIO[11:0]
001	DATA port[11~0] and UIO[11~0] → UIO[11:0]
010	DATA port[11~0] or UIO[11~0] → UIO[11:0]
011	DATA port[11~0] xor UIO[11~0] → UIO[11:0]
1xx	DATA port[3]이 '1' 일 때 UIO[3]에 one shot pulse DATA port[4]이 '1' 일 때 UIO[4]에 one shot pulse * Pulse width는 [UCFG4] 에서 설정.

5.3.3. Drive start commands

Previous 레지스터에 드라이브 구동에 관계된 값들을 설정하고 구동 시작 명령어를 실행하면 [PDCFG][6~0]에서 선택된 구동이 시작 또는 예약 된다. 현재 Previous register queue가 비어 있고 구동 시작 조건이 가능한 상태이면 즉시 구동하고 그렇지 않으면 예약된다. 예약된 구동 명령은 구동 시작 조건([DCFG][15~10])이 만족하였을 때 실행된다. 구동 시작 명령어는 [DCFG][6~0]의 설정 방법에 따라 두 가지로 구분되는데 [PDCFG][6~0]를 사용하는 것과 DATAP[6~0]를 사용하는 것이 있다. 또한 구동 시작 명령어는 속도 프로파일을 구별하여 세부적으로 나뉘어진다. 특히 [ASTxx]를 사용하면 한번의 명령어 수행으로 구동 드라이버 종류와 프로파일을 선택할 수 있다.



▶ **[(A)STRN][0xA0(0xDC)]**: 기본 속도 프로파일을 이용한 구동 예약 또는 시작.

드라이버 관련 Previous 레지스터들을 Previous 레지스터 큐에 예약하는 명령어 중 속도 프로파일이 시작 속도에서 목표속도까지 가속하여 다시 시작 속도로 감속하는 형태를 가진다.

▶ **[(A)STRO][0xA1(0xDD)]**: 목표속도[OBJ]에서 구동 시작하는 프로파일을 이용한 구동 예약 또는 시작.

목표 속도에서 시작하여 감속 조건을 만나면 시작 속도로 감속 후 구동 정지한다.

▶ **[(A)STRCO][0xA2(0xDE)]**: 목표속도[OBJ]에서 구동 시작하는 프로파일을 이용한 구동 예약 또는 시작.

목표 속도에서 시작하여 정속으로 구동 후 감속 없이 종료조건에서 구동 정지 한다. 단 [DCFG][30]을 '1'로 설정 하였다면 [(A)STRO] 동일하게 동작한다.

▶ **[(A)STRCS][0xA3(0xDF)]**: 시작속도[STD]에서 구동 시작하는 프로파일을 이용한 구동 예약 또는 시작.

시작 속도에서 시작하여 정속으로 구동 후 종료조건에서 구동 정지 한다.

5.3.4. Control commands

구동 정지 및 동기 시작/정지 신호 처리, 보간 스텝 확인, 초기화, 인터럽트 강제 발생에 관계된 명령어들로 구성된다.

▶ 구동 정지 명령어 :

Command(code)	Description
[SSTOP][0xA4]	해당 축의 현재 구동 드라이버를 감속 정지한다.
[STOP][0xA5]	해당 축의 현재 구동 드라이버를 급정지한다.
[ESTOP][0xAE]	전 축의 현재 구동 드라이버를 급정지한다.

▶ **[SQRO1][0xA6]**: 여러 칩 축간 동기 기능을 위한 SQSTR1 터미널 출력.

SQSTR1 터미널 포트에 one-shot pulse를 출력한다. 펄스 폭은 8 * Reference clock period이다. SQSTR1 터미널은 평소 출력이 High-Z(Open)상태이며 [SQRO1] 명령 실행시 8 * Reference clock period 시간 동안 '0'가 되었다가 다시 High-Z(Open)상태로 복귀한다. 이 명령어를 수행하여 현재 [DCFG][17~16]이 "01"로 구동 대기중인 축이 구동을 시작하게 할 수 있어 여러 칩의 축간 동기를 위한 기능을 구현할 수 있다.

▶ **[SQRO2][0xA7]**: 여러 칩 축간 동기 기능을 위한 SQSTR2 터미널 출력.

SQSTR2 터미널 포트에 one-shot pulse를 출력한다. 펄스 폭은 8 * Reference clock period이다. SQSTR2 터미널은 평소 출력이 High-Z(Open)상태이며 [SQRO2] 명령 실행시 8 * Reference clock period 시간 동안 '0'가 되었다가 다시 High-Z(Open)상태로 복귀한다. 이 명령어를 수행하여 현재 [DCFG][17~16]이 "10"로 구동 대기중인 축이 구동을 시작하게 할 수 있어 여러 칩의 축간 동기를 위한 기능을 구현할 수 있다.

▶ **[SQRI1][0xA8] : 칩 내부 측간 동기(SQSTR1 기다리는 중간).**

이 명령어를 수행하면 SQSTR1 터미널에 입력이 들어온 것과 같은 동작을 수행하는데, 현재 [DCFG][17~16]이 “01”로 구동 대기중인 측이 구동을 시작하게 된다. 즉 하나의 칩 내부에서 SQSTR1 신호를 기다리는 측들간의 동기를 SQSTR1 터미널의 신호 출력 없이 구현할 수 있다.

▶ **[SQRI2][0xA9] 칩 내부 측간 동기(SQSTR2 기다리는 중간).**

이 명령어를 수행하면 SQSTR2 터미널에 입력이 들어온 것과 같은 동작을 수행하는데, 현재 [DCFG][17~16]이 “10”로 구동 대기중인 측이 구동을 시작하게 된다. 즉 하나의 칩 내부에서 SQSTR1 신호를 기다리는 측들간의 동기를 SQSTR2 터미널의 신호 출력 없이 구현할 수 있다.

▶ **[SQPO1][0xAA] : SQSTP1 터미널 출력.**

SQSTP1 터미널 포트에 one-shot pulse를 출력한다. 펄스 폭은 $8 * \text{Reference clock period}$ 이다. SQSTP1 터미널은 평소 출력이 High-Z(Open)상태이며 [SQRO1] 명령 실행시 $8 * \text{Reference clock period}$ 시간 동안 ‘0’가 되었다가 다시 High-Z(Open)상태로 복귀한다. 이 명령어를 수행하여 현재 [DCFG][19~18]이 “01”로 설정되어 구동 중인 측이 감속 정지 또는 급정지 하여, 여러 칩의 측간 동기 정지 기능을 구현할 수 있다. 감속정지/급정지 선택은 [UCFG4][26] 에서 정한다.

▶ **[SQPO2][0xAB] : SQSTP2 터미널 출력.**

SQSTP2 터미널 포트에 one-shot pulse를 출력한다. 펄스 폭은 $8 * \text{Reference clock period}$ 이다. SQSTP2 터미널은 평소 출력이 High-Z(Open)상태이며 [SQRO1] 명령 실행시 $8 * \text{Reference clock period}$ 시간 동안 ‘0’가 되었다가 다시 High-Z(Open)상태로 복귀한다. 이 명령어를 수행하여 현재 [DCFG][19~18]이 “10”로 설정되어 구동 중인 측이 감속 정지 또는 급정지 하여 여러 칩의 측간 동기 정지 기능을 구현할 수 있다. 감속정지/급정지 선택은 [UCFG4][27] 에서 정한다.

▶ **[ISCNT][0x2C] : 보간 스텝 확인(28 bit).**

보간 제어 스텝 개수를 한다. 이 값은 보간 드라이브 시작 시 초기화 되고 보간 진행 중 보간 제어 펄스의 개수를 카운터 한다. 보간 중 가감속을 위해 필요한 보간 스텝을 정할 때 계산에 의한 방법과 가상 모드(펄스 출력 하지 않음) 구동 하여 [ISCNT] 값을 [(P)ISNUM] 설정하는데 사용할 수 있다.

▶ **[ISACNT][0x2D] : 확장 보간 스텝 확인(28 bit).**

보간 구동 중 펄스 속도 보상을 하지 않은 보간 제어 스텝 개수를 표시한다. 이 값은 보간 드라이브 시작 시 초기화 되고 보간 진행 중 보간 제어 펄스의 개수를 카운터 한다. 보간 중 가감속을 위해 필요한 보간 스텝을 정할 때 계산에 의한 방법과 가상 모드(펄스 출력 하지 않음) 구동 하여 [ISACNT] 값을 [(P)ISNUM] 설정하는데 사용할 수 있다.

▶ **[SWRESET][0xAF] : 초기화.**

모든 내부 레지스터 값을 초기화한다. 명령 수행 후 최소 Reference clock period * 10 이후 초기화가 완료 된다.

▶ **[DRPCNT][0xB0] : 마지막 구동 드라이브 출력 펄스수(28bit).**

최후 구동된 드라이브에 의해 출력된 펄스수를 표시한다. 이 값은 구동 시작 시 초기화 되고 구동 중 출력된 펄스를 카운터 한다.

▶ **[INTGEN][0xB1] : 인터럽트 강제 발생 명령.**

인터럽트 강제 발생 명령어이다. [IMASK1][31]이 ‘1’로 설정되었을 때 유효하다.

5.3.5.Commands for setting Peripheral function registers.

부가기능 설정에 관한 명령어들이며, 트리거 기능 설정 및 타이머기능, 소프트웨어 리미트, 서보 잔여펄스 제거 신호 설정들로 구성된다.

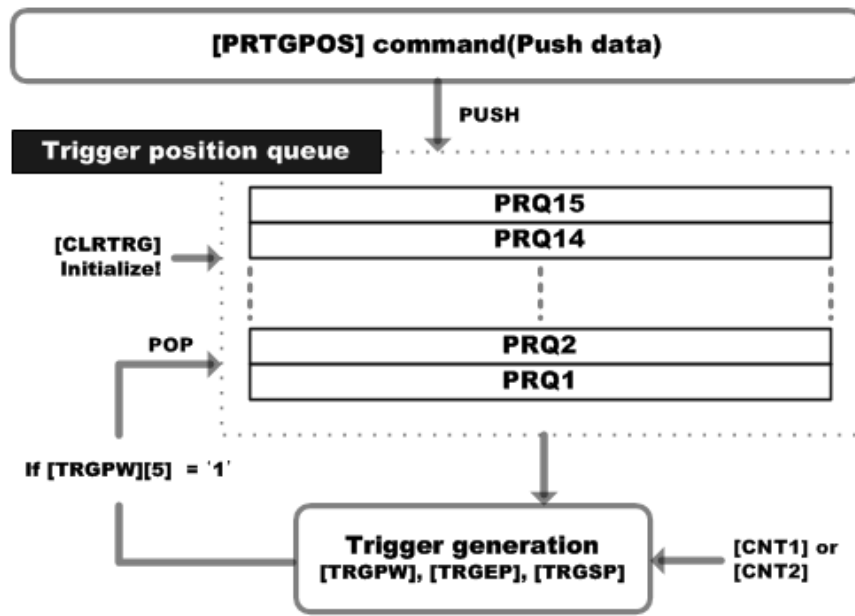


그림 5-5. 트리거 기능과 관련된 레지스터 기능블록도

▶ **[TRGQPOP][0xB2] : 트리거 위치 설정 레지스터 QUEUE POP.**

트리거 기능 사용시 사용될 트리거 위치 QUEUE 데이터를 POP한다. 현재 1 개 이상의 트리거 위치 설정 데이터가 입력되어 있다면 POP후 다음 데이터가 트리거 위치 데이터로 사용된다.

▶ **[TRTMCF][0x33/0xB3] : 트리거 위치 설정 QUEUE 관련 설정/타이머 기능 설정(10 bit)**

트리거 위치 설정 QUEUE는 Depth 15로 구성되는데 Empty/Full 인덱스 상태 표시 Flag를 감시하여 구동 중 새로운 트리거 위치 데이터를 Queue가 empty되기 전에 기입하여 고속 구동 중 다수의 트리거를 생성할 수 있다. 이때 사용되는 인덱스 상태표시 flag의 기준 데이터 개수를 조정 할 수 있는데 [TRTMCF][7~0]에서 설정한다.

[TRTMCF]	Description
[3~0]	트리거 위치 설정 Queue의 Empty flag 기준 데이터 수(Default : "0000")
[7~4]	트리거 위치 설정 Queue의 Full flag 기준 데이터 수(Default : "1111")

내장된 2개의 타이머는 설정된 시간 주기 마다 또는 시간 경과 후 인터럽트 또는 스크립터에서 사용하는 이벤트를 발생 시킨다. 이때 타이머의 시간 경과 조건 발생을 설정 시간 주기마다 할 것인지 또는 시간 경과 후 한번만 할 것 인가를 [TRTMCF][9~8]에서 설정한다.

[TRTMCF]	Description
[8]	1번 타이머 동작 설정('0': 시간 경과 후 한번, '1': 매 시간 경과 마다.)
[9]	2번 타이머 동작 설정('0': 시간 경과 후 한번, '1': 매 시간 경과 마다.)

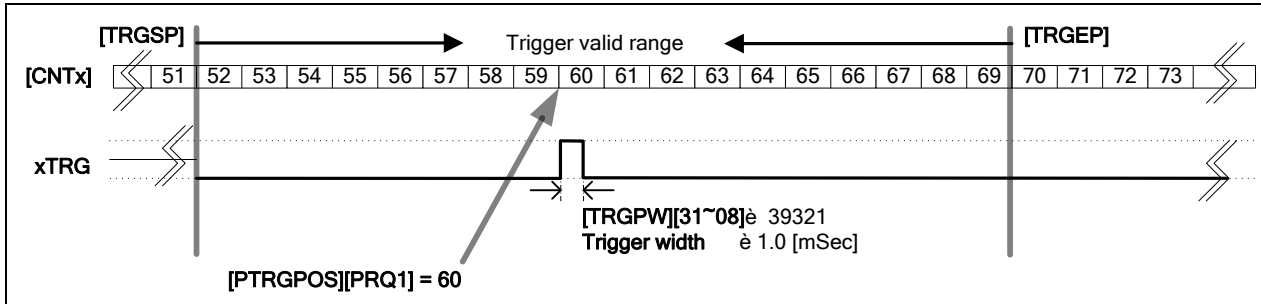
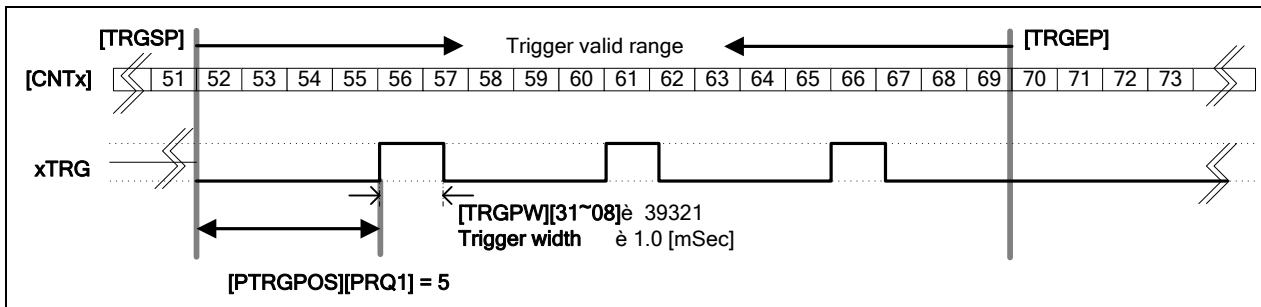
▶ **[TRGPW][0x36/0xB6] : 트리거 기능 관련 설정 레지스터(32 bit).**

트리거 기능 사용시 트리거 동작 모드/트리거 기준 카운터/트리거 사용 구동 방향/트리거 출력 신호 레벨/트리거 펄스 폭을 설정한다.

표 5-1. [TRGPW] 세부내용

[TRGPW]	Description
[0]	트리거 동작 모드 설정('0': 거리 주기마다 발생, '1': 트리거 설정 위치에서 발생)
[1]	트리거 기준 카운터 설정('0': 카운터 #1, '1': 카운터 #2)

[2]	트리거 기준 카운터 방향 사용 유무('0': 양방향 모두, '1': 설정 방향)
[3]	트리거 기준 카운터 방향 설정('0': 증가, '1': 감소)
[4]	트리거 출력 신호 레벨('0': High(정상시 Low), '1': Low(정상시 High))
[5]	트리거 발생시 트리거 위치 queue pop 여부('0': Pop 하지 않음, '1': 트리거 발생시 pop)
[6]	Don't care
[7]	트리거 출력 사용 유무('0': 트리거 출력 신호 사용하지 않음, '1': 사용함)
[31~8]	트리거 발생시 설정 신호 레벨 출력 시간 설정(트리거 펄스폭) 트리거 펄스 폭 = $T_{clk} * [TRGPW][31:8]$, ($T_{clk} \sim 16777215 * T_{clk}$)

그림 5-6 절대 위치 트리거 출력(위치:60, 폭:39321* T_{CLK} , Active level:1)그림 5-7 주기 위치 트리거 출력(주기:10, 폭 : 39321* T_{CLK} , Active level:1)

▶ [TRGSP][0x37/0xB7]: 트리거 발생 유효 범위 시작 위치 설정(28bit)

설정범위는 -134,217,728~134,217,727(0x80000000~0x7FFFFFFF)이고 [TRGEP]보다 작은 값으로 설정 되어야 하며, RESET 후 기본값은 -134,217,728 (0x80000000)이다. 트리거 기능 사용시 기준 카운터 값이 설정 범위 ([TRGSP]~[TRGEP]) 안에 있을 때 트리거 위치/주기에서 유효하다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4개 bit는 27bit 값으로 sign-extension 된다.

▶ [TRGEP][0x38/0xB8]: 트리거 발생 유효 범위 종료 위치 설정(28bit)

설정범위는 -134,217,728~134,217,727(0x80000000~0x7FFFFFFF)이며 [TRGSP]보다 큰 값으로 설정 되어야 하며, [RESET 후 기본값은 134,217,727(0x7FFFFFFF)이다. 트리거 기능 사용시 기준 카운터 값이 설정 범위 ([TRGSP]~[TRGEP]) 안에 있을 때 트리거 위치/주기에서 유효하다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4개 bit는 27bit 값으로 sign-extension 된다.

▶ [PTRGPOS][0x39/0xB9]: 트리거 발생 거리주기/절대위치 설정 큐 데이터 설정(push)(28bit)

트리거 발생 거리주기/절대위치 정보는 Depth 15의 Queue에 입력되는데, [PTRGPOS] 명령어로 데이터 포트의 28bit 값을 Queue에 push 한다. 설정범위는 -134,217,728~134,217,727(0x80000000~0x7FFFFFFF)이며 트리거 발생시 Pop하도록 설정 되었다면 설정된 queue 데이터가 하나씩 감소하게 되고, Queue의 데이터가 모두 pop하게 되면 트리거는 발생하지 않는다.

▶ [CLRTRG][0xBA]: 트리거 발생 거리주기/절대위치 설정 큐 데이터 초기화

트리거 발생 거리주기/절대위치 정보는 Depth 15의 Queue 데이터를 초기화 하여 Queue를 비운다.

▶ **[TRGGEN][0xBB] : 트리거 강제 발생**

트리거 주기가 0보다 큰 값이고 트리거 출력을 사용하도록 설정 하였다면 카운터 값과 트리거 거리주기/절대위치 값의 비교 결과와 상관없이 트리거 신호를 명령어로 발생 시킬 수 있다.

▶ **[TMRP1][0x3C/0xBC] : 타이머 #1 시간 주기 설정(28bit)**

설정범위는 0(0x00000000)~268,435,455(0xFFFFFFFF)이며 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4개 bit는 '0'으로 sign-extended 된다. 타이머 #1의 주기는 $[TMRP1] * Tclk$ 가 되며 $Tclk = 39.3216MHz$ 일 때 최장 시간 주기는 약 6.8초 이다.

▶ **[TMRP2][0x3D/0xBD] : 타이머 #2 시간 주기 설정(28bit)**

설정범위는 0(0x00000000)~268,435,455(0xFFFFFFFF)이며 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4개 bit는 '0'으로 sign-extended 된다. 타이머 #2의 주기는 $[TMRP2] * Tclk$ 가 되며 $Tclk = 39.3216MHz$ 일 때 최장 시간 주기는 약 6.8초 이다.

▶ **[TMR1GEN][0x3E/0xBE] : 타이머 #1 기능 시작 명령어.**

타이머 기능을 시작[0xBE]/종료[0x3E]하는 명령어이다. 타이머 기능을 시작하면 설정된 타이머 주기 이후 타이머 이벤트가 발생 하게 된다. 만일 타이머 기능을 한번만 사용하게 설정되어 있다면 자동으로 타이머 이벤트 발생후 종료하게 된다.

▶ **[TMR2GEN][0x3F/0xBF] : 타이머 #2 기능 시작 명령어.**

타이머 기능을 시작[0xBF]/종료[0x3F]하는 명령어이다. 타이머 기능을 시작하면 설정된 타이머 주기 이후 타이머 이벤트가 발생 하게 된다. 만일 타이머 기능을 한번만 사용하게 설정되어 있다면 자동으로 타이머 이벤트 발생 후 종료하게 된다.

▶ **[ERCRS][0x60/0xE0] : 잔여 펄스 제거 신호 출력 시작[0xE0]/정지[0x60]**

잔여 펄스 제거 신호를 출력을 제어하는 명령어이다. 잔여 펄스 제거 신호 출력이 레벨로 되어 있을경우 현재 출력을 정지하거나 강제로 발생 시킬 수 있으며, 펄스 출력으로 되어 있을경우 펄스 출력을 한번 생성 하도록 할 수 있다.

5.3.6.Commands for setting Script function registers.

CAMC-QI는 모든 명령어를 Script 기능 블록에 예약 할 수 있는데, 측당 Depth 15인 명령어 예약 Queue 2개와 Depth 1인 명령어 레지스트 2개로 동시에 4개의 명령어에 대하여 이벤트를 감시 할 수 있다. 예약된 명령어가 읽기 명령어일 경우 Caption queue 또는 Caption 레지스터로 저장된다.

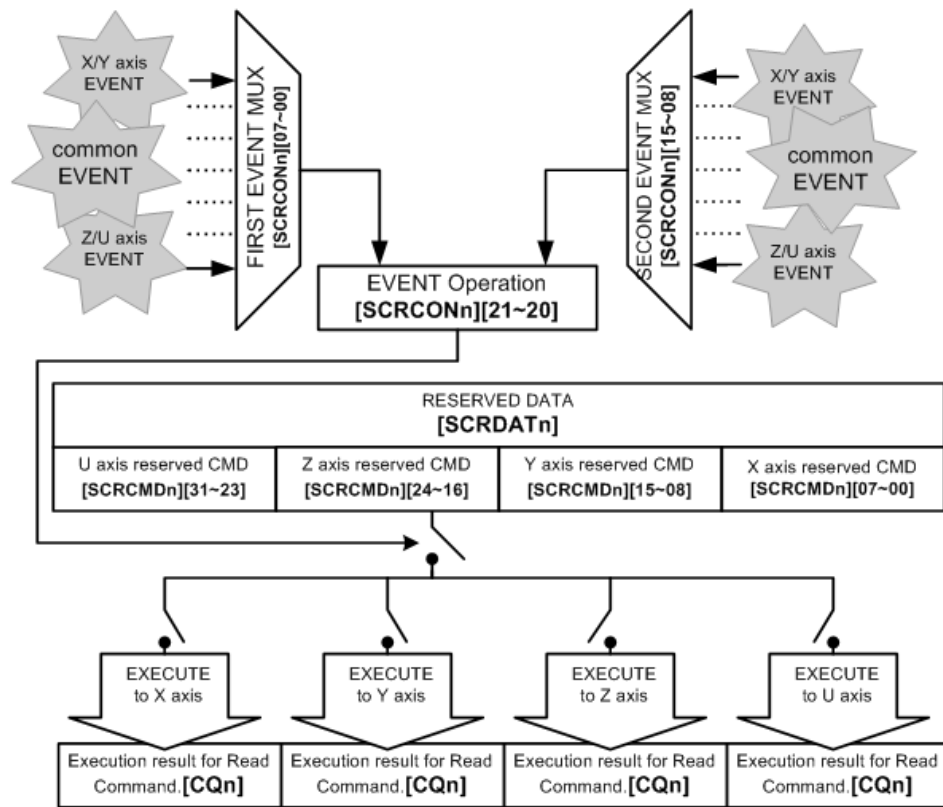


그림 5-8. 스크립트 기능 블록도

표 5-2. 스크립트 이벤트 리스트

CODE	NAME	Description
0x00	NOOP	NO operation.
0x01	DRVEND	Drive end event(inposition function excluded).
0x02	DECEL	Deceleration state.
0x03	CONST	Constant speed state.
0x04	ACCEL	Acceleration state.
0x05	CNT1L	Counter1 < Comparater1 state.
0x06	CNT1E	Counter1 = Comparater1 state.
0x07	CNT1G	Counter1 > Comparater1 state.
0x08	CNT1LE	Counter1 ≤ Comparater1 state.
0x09	CNT1GE	Counter1 ≥ Comparater1 state.
0x0A	CNT1EUP	Counter1 = Comparater1 event during counting up.
0x0B	CNT1EDN	Counter1 = Comparater1 event during counting down.
0x0C	CNT1BND	Counter1 is same with boundary value.
0x0D	CNT2L	Counter2 < Comparater2 state.
0x0E	CNT2E	Counter2 = Comparater2 state.
0x0F	CNT2G	Counter2 > Comparater2 state.
0x10	CNT2LE	Counter2 ≤ Comparater2 state.
0x11	CNT2GE	Counter2 ≥ Comparater2 state.
0x12	CNT2EUP	Counter2 = Comparater2 event during counting up.
0x13	CNT2EDN	Counter2 = Comparater2 event during counting down.
0x14	CNT2BND	Counter2 is same with boundary value.

0x15	CNT3L	Counter3 < Comparater3 state.
0x16	CNT3E	Counter3 = Comparater3 state.
0x17	CNT3G	Counter3 > Comparater3 state.
0x18	CNT3LE	Counter3 ≤ Comparater3 state.
0x19	CNT3GE	Counter3 ≥ Comparater3 state.
0x1A	CNT3EUP	Counter3 = Comparater3 event during counting up.
0x1B	CNT3EDN	Counter3 = Comparater3 event during counting down.
0x1C	CNT3BND	Counter3 is same with boundary value.
0x1D	CNT4L	Counter4 < Comparater4 state.
0x1E	CNT4E	Counter4 = Comparater4 state.
0x1F	CNT4G	Counter4 > Comparater4 state.
0x20	CNT4LE	Counter4 ≤ Comparater4 state.
0x21	CNT4GE	Counter4 ≥ Comparater4 state.
0x22	CNT4EUP	Counter4 = Comparater4 event during counting up.
0x23	CNT4EDN	Counter4 = Comparater4 event during counting down.
0x24	CNT4BND	Counter4 is same with boundary value.
0x25	CNT5L	Counter5 < Comparater5 state.
0x26	CNT5E	Counter5 = Comparater5 state.
0x27	CNT5G	Counter5 > Comparater5 state.
0x28	CNT5LE	Counter5 ≤ Comparater5 state.
0x29	CNT5GE	Counter5 ≥ Comparater5 state.
0x2A	CNT5EUP	Counter5 = Comparater5 event during counting up.
0x2B	CNT5EDN	Counter5 = Comparater5 event during counting down.
0x2C	CNT5BND	Counter5 is same with boundary value.
0x2D	DEVL	Counter4 value < Comparater4 state.
0x2E	DEVE	Counter4 value = Comparater4 state.
0x2F	DEVG	Counter4 value > Comparater4 state.
0x30	DEVLE	Counter4 value ≤ Comparater4 state.
0x31	DEVGE	Counter4 value ≥ Comparater4 state.
0x32	PELM	PELM input signal is activated state.
0x33	NELM	NELM input signal is activated state.
0x34	PSLM	PSLM input signal is activated state.
0x35	NSLM	NSLM input signal is activated state.
0x36	ALARM	ALAMR input signal is activated state.
0x37	INPOS	INPOSITION input signal is activated state.
0x38	ESTOP	ESTOP input signal is activated state.
0x39	ORG	ORG input signal is activated state.
0x3A	Z_PHASE	Z_PHASE input signal is activated state.
0x3B	ECUP	ECUP input signal is high level state.
0x3C	ECDN	ECDN input signal is high level state.
0x3D	EXPP	EXPP input signal is high level state.
0x3E	EXMP	EXMP input signal is high level state.
0x3F	SQSTR1	SYNC Start1 input signal is activated state(activated)
0x40	SQSTR2	SYNC Start2 input signal is activated state(activated)
0x41	SQSTP1	SYNC STOP1 input signal is activated state(activated)
0x42	SQSTP2	SYNC STOP2 input signal is activated state(activated)
0x43	ALARMS	At least one alarm signal of each axis is activated state.
0x44	UIO0	UIO0 data is high state.
0x45	UIO1	UIO1 data is high state.
0x46	UIO2	UIO2 data is high state.
0x47	UIO3	UIO3 data is high state.
0x48	UIO4	UIO4 data is high state.
0x49	UIO5	UIO5 data is high state.
0x4A	UIO6	UIO6 data is high state.

0x4B	UIO7	UIO7 data is high state.
0x4C	UIO8	UIO8 data is high state.
0x4D	UIO9	UIO9 data is high state.
0x4E	UIO10	UIO10 data is high state.
0x4F	UIO11	UIO11 data is high state.
0x50	ERC	ERC output is activated.
0x51	TRG	TRIGGER signal is activated.
0x52	PREQI0	Previous queue data index 0 bit is high state.
0x53	PREQI1	Previous queue data index 0 bit is high state.
0x54	PREQI2	Previous queue data index 0 bit is high.
0x55	PREQZ	Previous queue is empty state.
0x56	PREQF	Previous queue is full state.
0x57	MPGE1	MPG first stage is overflowed state.
0x58	MPGE2	MPG second stage is overflowed state.
0x59	MPGE3	MPG third stage is overflowed state.
0x5A	MPGERR	MPG all state is overflowed state.
0x5B	TRGCNT0	TRIGGER queue index bit 0 is high state.
0x5C	TRGCNT1	TRIGGER queue index bit 1 is high state.
0x5D	TRGCNT2	TRIGGER queue index bit 2 is high state.
0x5E	TRGCNT3	TRIGGER queue index bit 3 is high state.
0x5F	TRGQEPT	TRIGGER queue is empty state.
0x60	TRGQFULL	TRIGGER queue is full state.
0x61	DPAUSE	Drive paused state.
0x62	ESTOPEXE	Emergency stop occurred
0x63	SSTOPEXE	Slowdown stop occurred
0x64	PLMTSTOP	Limit stop event occurred during positive driving.
0x65	NLMTSTOP	Limit stop event occurred during negative driving.
0x66	OPLMTSTOP	Optional limit stop event occurred during positive driving.
0x67	ONLMTSTOP	Optional limit stop event occurred during negative driving.
0x68	PSWESTOP	Software emergency limit stop event occurred.(CW)
0x69	NSWESTOP	Software emergency limit stop event occurred.(CCW)
0x6A	PSWSSTOP	Software slowdown limit stop event occurred.(CW)
0x6B	NSWSSTOP	Software slowdown limit stop event occurred.(CCW)
0x6C	ALMSTOP	Emergency stop event occurred by alarm signal function.
0x6D	ESTOPSTOP	Emergency stop event occurred by estop signal function.
0x6E	ESTOPCMD	Emergency stop event occurred by command.
0x6F	SSTOPCMD	Slowdown stop event occurred by command.
0x70	ALLSTCMD	Emergency stop event occurred by all stop command.
0x71	SYSTOP1	SYNC stop1 event occurred.
0x72	SYSTOP2	SYNC stop2 event occurred.
0x73	ENCODERR	Encoder input error event occurred.
0x74	MPGERR	MPG input error event occurred.
0x75	ORGOK	Original drive is executed successfully.
0x76	SSCHOK	Signal search drive is executed successfully.
0x77	UIO0	UIO0 data is low state.
0x78	UIO1	UIO1 data is low state.
0x79	UIO2	UIO2 data is low state.
0x7A	UIO3	UIO3 data is low state.
0x7B	UIO4	UIO4 data is low state.
0x7C	UIO5	UIO5 data is low state.
0x7D	UIO6	UIO6 data is low state.
0x7E	UIO7	UIO7 data is low state.
0x7F	UIO8	UIO8 data is low state.
0x80	UIO9	UIO9 data is low state.

0x81	UIO10	UIO10 data is low state.
0x82	UIO11	UIO11 data is low state.
0x83	UIO0	UIO0 rising edge event occurred.
0x84	UIO1	UIO1 rising edge event occurred.
0x85	UIO2	UIO2 rising edge event occurred.
0x86	UIO3	UIO3 rising edge event occurred.
0x87	UIO4	UIO4 rising edge event occurred.
0x88	UIO5	UIO5 rising edge event occurred.
0x89	UIO6	UIO6 rising edge event occurred.
0x8A	UIO7	UIO7 rising edge event occurred.
0x8B	UIO8	UIO8 rising edge event occurred.
0x8C	UIO9	UIO9 rising edge event occurred.
0x8D	UIO10	UIO10 rising edge event occurred.
0x8E	UIO11	UIO11 rising edge event occurred.
0x8F	UIO0	UIO0 falling edge event occurred.
0x90	UIO1	UIO1 falling edge event occurred.
0x91	UIO2	UIO2 falling edge event occurred.
0x92	UIO3	UIO3 falling edge event occurred.
0x93	UIO4	UIO4 falling edge event occurred.
0x94	UIO5	UIO5 falling edge event occurred.
0x95	UIO6	UIO6 falling edge event occurred.
0x96	UIO7	UIO7 falling edge event occurred.
0x97	UIO8	UIO8 falling edge event occurred.
0x98	UIO9	UIO9 falling edge event occurred.
0x99	UIO10	UIO10 falling edge event occurred.
0x9A	UIO11	UIO11 falling edge event occurred.
0x9B	DRVSTR	Drive started.
0x9C	DNSTR	Speed down event occurred.
0x9D	COSTR	Constant speed event occurred.
0x9E	UPSTR	Speed up event occurred.
0x9F	CONTISTR	Continuous drive started.
0xA0	PRESETSTR	Preset drive started.
0xA1	MPGSTR	MPG drive started.
0xA2	ORGSTR	Original drive started.
0xA3	SSCHSTR	Signal search drive started.
0xA4	PATHSTR	Interpolation drive started.
0xA5	SLAVESTR	Slave drive started.
0xA6	CCWSTR	CCW direction drive started.
0xA7	INPWAIT	Inposition wait event occurred.
0xA8	LINSTR	Linear drive stated.
0xA9	CIRSTR	Circular drive started.
0xAA	DRVSTOPII	Drive stopped.(inposition state included)
0xAB	DNEND	Speed down end event occurred.
0xAC	COEND	Constant speed end event occurred.
0xAD	UPEND	Speed up end event occurred.
0xAE	CONTIEND	Continuous drive ended.
0xAF	PRESETEND	Preset drive ended.
0xB0	MPGEND	MPG drive ended.
0xB1	ORGEND	Original drive ended.
0xB2	SSCHEND	Signal search drive ended.
0xB3	PATHEND	Interpolation drive ended.
0xB4	SLAVEEND	Slave drive ended.
0xB5	CCWEND	CCW direction drive ended.
0xB6	INPEND	Escape from Inposition waiting.

0xB7	LINEND	Linear drive ended.
0xB8	CIREND	Circular drive ended.
0xB9	BUSY	During driving state.
0xBA	NBUSY	During not driving state.
0xBB	TMR1EX	Timer1 expired event.
0xBC	TMR2EX	Timer2 expired event.
0xBD	DRVENDIII	Drive(that interrupt enable bit is set to high) end event
0xBE	ERROR	Error stop occurred.
0xBF	NOP	NOP.

▶ [SCRCON1][0x40/0xC0]: 스크립트 #1 기능 설정 명령어(25bit)

스크립트 #1의 기능 설정 명령어로 감시 이벤트 설정/이벤트 연산/이벤트 입력 측/실행시 인터럽트 사용/명령 실행 데이터 설정/실행모드(한번,항상)를 설정한다. 특히 [SCRCMD1], [SCRDAT1] 으로 설정된 임시 저장 값을 [SCRCON1] 명령어 수행과 동기 하여 예약되기 때문에 [SCRCMD1], [SCRDAT1] 설정 이후에 [SCRCON1] 명령을 수행하여야 한다. 스크립트 #1의 기능 설정은 15 Depth의 queue 형태를 가지며 먼저 예약된 것이 먼저 수행된다. 수행된 이후 지워지거나 Queue의 마지막 위치에 다시 예약 될 수 있다. 세부내용은 다음과 같다.

[SCRCON1]	Description
[7~0]	첫번째 감시 이벤트 설정(0xFF일 경우 무조건 수행)
[15~8]	두번째 감시 이벤트 설정
[17~16]	첫번째 감시 이벤트 입력 측 설정.
[19~18]	두번째 감시 이벤트 입력 측 설정.
[21~20]	첫번째와 두번째 이벤트 연산 설정 “00”: 연산하지 않음(첫번째 감시 이벤트로만 실행) “01”: 첫번째 이벤트 조건 AND 두번째 이벤트 조건 “10”: 첫번째 이벤트 조건 OR 두번째 이벤트 조건 “11”: 첫번째 이벤트 조건 XOR 두번째 이벤트 조건
[22]	설정 명령 수행 시 인터럽트 수행 여부. 해당 인터럽트 마스크 설정 필요.(‘0’: 사용 안 함, ‘1’: 사용)
[23]	명령 수행 시 필요한 입력데이터 설정(‘0’: [SCRDAT1] 사용, ‘1’: [SCRTG]에서 지정한 내용)
[24]	이벤트 감시 조건 만족 시 예약 명령어 사용 방법 ‘0’: 한번만 수행, 수행 후 Queue에서 지움 ‘1’: 계속 수행, 수행 후 Queue 마지막 데이터로 재 예약.

▶ [SCRCMD1][0x41/0xC1]: 스크립트 #1 수행 시 실행할 명령어 예약 명령어(32bit)

스크립트 #1의 설정된 이벤트 조건이 만족할 때 각 측에 실행할 명령어를 예약하기 위한 명령어이다. 만일 특정 측에 대하여 이벤트에 대한 명령어 수행이 필요하지 않다면 [0xCF]로 기입하면 된다.

[SCRCMD1]	Description
[7~0]	X 측 에서 실행할 명령어
[15~8]	Y 측 에서 실행할 명령어
[23~16]	Z 측 에서 실행할 명령어
[31~24]	U 측 에서 실행할 명령어

▶ [SCRDAT1][0x42/0xC2]: 스크립트 #1 수행 시 실행할 명령어에 필요한 데이터 예약 명령어(32bit)

스크립트 #1의 설정된 이벤트 조건이 만족할 때 각 측에 실행할 명령어가 필요한 데이터를 예약하기 위한 명령어이다. 실행 데이터는 [SCRDAT1]과 [SCRTG]에서 지정한 레지스터를 선택할 수 있으며 실행 명령어가 데이터를 필요로 하지 않으면 설정하지 않아도 된다.

▶ [CQ1][0x43]: 스크립트 #1로 읽기 명령어 수행 시 결과 값(32bit)

스크립트 #1의 예약된 명령어 중 읽기 명령어의 실행 결과는 Caption queue(Depth 15)에 저장되는데 [CQ1] 명령을 수행하여 저장된 결과값을 읽어볼 수 있다. [CQ1] 수행 후 읽혀진 데이터는 Caption queue에서 삭제된다.

▶ [SCRCFG1][0x44/0xC4]: 스크립트 #1의 Queue에 관계된 크기에 관계된 flag 설정(16bit)

스크립트 #1의 실행 Queue(명령어, 데이터, 이벤트 설정)와 읽기 결과 저장 Queue는 Depth 15로 구성되어 있는데,

현재 설정되어 있는 데이터의 개수에 따라 Full/Empty flag를 감시하여 새로운 값을 입력한다거나 새로이 갱신된 결과를 읽어 보는 등의 동작을 수행할 수 있다. 따라서 Full/Empty flag가 지시하는 현재 데이터의량을 조정함으로써 사용자는 Queue 형태의 예약 및 결과 값을 매 이벤트에 빠짐없이 적용할 수 있다.

[SCRCFG1]	Description
[3~0]	실행 Queue의 Empty size(0~15), 리셋 후 초기 값(0)
[7~4]	실행 Queue의 Full size(0~15), 리셋 후 초기 값(15)
[11~8]	갈무리 Queue의 Empty size(0~15), 리셋 후 초기 값(0)
[15~12]	갈무리 Queue의 Full size(0~15), 리셋 후 초기 값(15)

▶ [SCRCON2][0x45/0xC5]: 스크립트 #2 기능 설정 명령어(25bit)

스크립트 #2의 기능 설정 명령어로 감시 이벤트 설정/이벤트 연산/이벤트 입력 측/실행시 인터럽트 사용/명령 실행 데이터 설정/실행모드(한번,항상)를 설정한다. 특히 [SCRCON2], [SCRDAT2] 으로 설정된 임시 저장 값을 [SCRCON2] 명령어 수행과 동기 하여 예약되기 때문에 [SCRCON2], [SCRDAT2] 설정 이후에 [SCRCON2] 명령을 수행하여야 한다. 스크립트 #1의 기능 설정은 15 Depth의 queue 형태를 가지며 먼저 예약된 것이 먼저 수행된다. 수행된 이후 지워지거나 Queue의 마지막 위치에 다시 예약 될 수 있다. 세부내용은 다음과 같다.

[SCRCON2]	Description
[7~0]	첫번째 감시 이벤트 설정(0xFF일 경우 무조건 수행)
[15~8]	두번째 감시 이벤트 설정
[17~16]	첫번째 감시 이벤트 입력 측 설정.
[19~18]	두번째 감시 이벤트 입력 측 설정.
[21~20]	첫번째와 두번째 이벤트 연산 설정
[22]	설정 명령 수행 시 인터럽트 수행 여부. 해당 인터럽트 마스크 설정 필요.(‘0’ : 사용 안 함, ‘1’ : 사용)
[23]	명령 수행 시 필요한 입력데이터 설정(‘0’ : [SCRDAT2] 사용, ‘1’ : [SCRTG]에서 지정한 내용)
[24]	이벤트 감시 조건 만족 시 예약 명령어 사용 방법 ‘0’ : 한번만 수행, 수행 후 Queue에서 지움 ‘1’ : 계속 수행, 수행 후 Queue 마지막 데이터로 재 예약.

▶ [SCRCON2][0x46/0xC6]: 스크립트 #2 수행 시 실행할 명령어 예약 명령어(32bit)

스크립트 #2의 설정된 이벤트 조건이 만족할 때 각 측에 실행할 명령어를 예약하기 위한 명령어이다. 만일 특정 측에 대하여 이벤트에 대한 명령어 수행이 필요하지 않다면 [0xCF]로 기입하면 된다.

[SCRCON2]	Description
[7~0]	X 측 에서 실행할 명령어
[15~8]	Y 측 에서 실행할 명령어
[23~16]	Z 측 에서 실행할 명령어
[31~24]	U 측 에서 실행할 명령어

▶ [SCRDAT2][0x47/0xC7]: 스크립트 #2 수행 시 실행할 명령어에 필요한 데이터 예약 명령어(32bit)

스크립트 #2의 설정된 이벤트 조건이 만족할 때 각 측에 실행할 명령어가 필요한 데이터를 예약하기 위한 명령어이다. 실행 데이터는 [SCRDAT2]과 [SCRTG]에서 지정한 레지스터로 선택할 수 있으며 실행 명령어가 데이터를 필요로 하지 않으면 설정하지 않아도 된다.

▶ [CQ2][0x48]: 스크립트 #1로 읽기 명령어 수행 시 결과 값(32bit)

스크립트 #2의 예약된 명령어 중 읽기 명령어의 실행 결과는 Caption queue(Depth 15)에 저장되는데 [CQ2] 명령을 수행하여 저장된 결과값을 읽어볼 수 있다. [CQ2] 읽기 수행 후 읽혀진 데이터는 Caption queue에서 자동으로 삭제된다.

▶ [SCRCFG2][0x49/0xC9]: 스크립트 #2의 Queue에 관계된 크기에 관계된 flag 설정(16bit)

스크립트 #2의 실행 Queue(명령어, 데이터, 이벤트 설정)와 읽기 결과 저장 Queue는 Depth 15로 구성되어 있는데, 현재 설정되어 있는 데이터의 개수에 따라 Full/Empty flag를 감시하여 새로운 값을 입력한다거나 새로이 갱신된 결과를 읽어 보는 등의 동작을 수행할 수 있다. 따라서 Full/Empty flag가 지시하는 현재 데이터의량을 조정함으로써 사용자는 Queue 형태의 예약 및 결과 값을 매 이벤트에 빠짐없이 적용할 수 있다.

[SCRCFG2]	Description
[3~0]	스크립트 #2 실행 Queue의 Empty size(0~15), 리셋 후 초기 값(0)
[7~4]	스크립트 #2 실행 Queue의 Full size(0~15), 리셋 후 초기 값(15)
[11~8]	스크립트 #2 갈무리 Queue의 Empty size(0~15), 리셋 후 초기 값(0)
[15~12]	스크립트 #2 갈무리 Queue의 Full size(0~15), 리셋 후 초기 값(15)

▶ [SCRCON3][0x4A/0xCA] : 스크립트 #3 기능 설정 명령어(25bit)

스크립트 #3의 기능 설정 명령어로 감시 이벤트 설정/이벤트 연산/이벤트 입력 측/실행시 인터럽트 사용/명령 실행 데이터 설정/실행모드(한번,항상)를 설정한다. 특히 [SCRCON3], [SCRDAT3] 으로 설정된 임시 저장 값을 [SCRCON3] 명령어 수행과 동기 하여 예약되기 때문에 [SCRCON3], [SCRDAT3] 설정 이후에 [SCRCON3] 명령을 수행하여야 한다. 스크립트 #3의 기능 설정은 1 Depth의 레지스터 형태를 가지며 수행 후 삭제 또는 지속적인 이벤트 감시가 가능하다. 세부내용은 다음과 같다.

[SCRCON3]	Description
[7~0]	첫번째 감시 이벤트 설정(0xFF일 경우 무조건 수행)
[15~8]	두번째 감시 이벤트 설정
[17~16]	첫번째 감시 이벤트 입력 측 설정.
[19~18]	두번째 감시 이벤트 입력 측 설정.
[21~20]	첫번째와 두번째 이벤트 연산 설정
[22]	설정 명령 수행 시 인터럽트 수행 여부. 해당 인터럽트 마스크 설정 필요.(‘0’ : 사용 안 함, ‘1’ : 사용)
[23]	명령 수행 시 필요한 입력데이터 설정(‘0’ : [SCRDAT3] 사용, ‘1’ : [SCRTG]에서 지정한 내용)
[24]	이벤트 감시 조건 만족 시 예약 명령어 사용 방법 ‘0’ : 한번만 수행, 수행 후 지움. ‘1’ : 이벤트 감시 조건 만족 시 마다 계속 수행.

▶ [SRCMD3][0x4B/0xCB] : 스크립트 #3 수행 시 실행할 명령어 예약 명령어(32bit)

스크립트 #3의 설정된 이벤트 조건이 만족할 때 각 측에 실행할 명령어를 예약하기 위한 명령어이다. 만일 특정 측에 대하여 이벤트에 대한 명령어 수행이 필요하지 않다면 [0xCF]로 기입하면 된다.

[SRCMD3]	Description
[7~0]	X 측 에서 실행할 명령어
[15~8]	Y 측 에서 실행할 명령어
[23~16]	Z 측 에서 실행할 명령어
[31~24]	U 측 에서 실행할 명령어

▶ [SCRDAT3][0x4C/0xCC] : 스크립트 #3 수행 시 실행할 명령어에 필요한 데이터 예약 명령어(32bit)

스크립트 #3의 설정된 이벤트 조건이 만족할 때 각 측에 실행할 명령어가 필요한 데이터를 예약하기 위한 명령어이다. 실행 데이터는 [SCRDAT3]과 [SCRTG]에서 지정한 레지스터로 선택할 수 있으며 실행 명령어가 데이터를 필요로 하지 않으면 설정하지 않아도 된다.

▶ [CQ3][0x4D] : 스크립트 #3로 읽기 명령어 수행 시 결과 값(32bit)

스크립트 #3의 예약된 명령어 중 읽기 명령어의 실행 결과는 Caption 레지스터에 저장되는데 [CQ3] 명령을 수행하여 저장된 결과값을 읽어볼 수 있다. [CQ3] 수행 후 읽혀진 데이터는 삭제된다.

▶ [SCRCON4][0x50/0xD0] : 스크립트 #4 기능 설정 명령어(25bit)

스크립트 #4의 기능 설정 명령어로 감시 이벤트 설정/이벤트 연산/이벤트 입력 측/실행시 인터럽트 사용/명령 실행 데이터 설정/실행모드(한번,항상)를 설정한다. 특히 [SCRCON4], [SCRDAT4] 으로 설정된 임시 저장 값을 [SCRCON4] 명령어 수행과 동기 하여 예약되기 때문에 [SCRCON4], [SCRDAT4] 설정 이후에 [SCRCON4] 명령을 수행하여야 한다. 스크립트 #4의 기능 설정은 1 Depth의 레지스터 형태를 가지며 수행 후 삭제 또는 지속적인 이벤트 감시가 가능하다. 세부내용은 다음과 같다.

[SCRCON2]	Description
[7~0]	첫번째 감시 이벤트 설정(0xFF일 경우 무조건 수행)
[15~8]	두번째 감시 이벤트 설정

[17~16]	첫번째 감시 이벤트 입력 축 설정.
[19~18]	두번째 감시 이벤트 입력 축 설정.
[21~20]	첫번째와 두번째 이벤트 연산 설정.
[22]	설정 명령 수행 시 인터럽트 수행 여부. 해당 인터럽트 마스크 설정 필요.('0' : 사용 안 함, '1' : 사용)
[23]	명령 수행 시 필요한 입력데이터 설정('0' : [SCRDAT4] 사용, '1' : [SCRTG]에서 지정한 내용)
[24]	이벤트 감시 조건 만족 시 예약 명령어 사용 방법 '0' : 한번만 수행, 수행 후 지움. '1' : 이벤트 감시 조건 만족 시 마다 계속 수행.

▶ [SRCMD4][0x51/0xD1] : 스크립트 #4 수행 시 실행할 명령어 예약 명령어(32bit)

스크립트 #4의 설정된 이벤트 조건이 만족할 때 각 축에 실행할 명령어를 예약하기 위한 명령어이다. 만일 특정 축에 대하여 이벤트에 대한 명령어 수행이 필요하지 않다면 [0xCF]로 기입하면 된다.

[SRCMD4]	Description
[7~0]	X 축 에서 실행할 명령어
[15~8]	Y 축 에서 실행할 명령어
[23~16]	Z 축 에서 실행할 명령어
[31~24]	U 축 에서 실행할 명령어

▶ [SCRDAT4][0x52/0xD2] : 스크립트 #4 수행 시 실행할 명령어에 필요한 데이터 예약 명령어(32bit)

스크립트 #4의 설정된 이벤트 조건이 만족할 때 각 축에 실행할 명령어가 필요한 데이터를 예약하기 위한 명령어이다. 실행 데이터는 [SCRDAT4]과 [SCRTG]에서 지정한 레지스터로 선택할 수 있으며 실행 명령어가 데이터를 필요로 하지 않으면 설정하지 않아도 된다.

▶ [CQ4][0x53] : 스크립트 #4로 읽기 명령어 수행 시 결과 값(32bit)

스크립트 #4의 예약된 명령어 중 읽기 명령어의 실행 결과는 Caption 레지스터에 저장되는데 [CQ4] 명령을 수행하여 저장된 결과값을 읽어볼 수 있다. [CQ4] 수행 후 읽혀진 데이터는 삭제된다.

▶ [SCRTG][0x54/0xD4] : 스크립트 명령 수행 시 사용할 데이터 입력 설정(16bit)

스크립트 수행 시 필요한 데이터는 [SCRDATn]값과 [SCRTG]에서 지정한 레지스터로 사용할 수 있다. 각 스크립트에서 사용할 지정 레지스터는 다음 표와 같다.

[SCRTG]	Description
[3~0]	스크립트 #1의 명령어 실행시 사용할 실행 데이터 지정 "0000" : 카운터 #1 "0001" : 카운터 #2 "0010" : 카운터 #3 "0011" : 카운터 #4 "0100" : 카운터 #5 "0101" : CQ1 데이터. "0110" : CQ2 데이터. "0111" : CQ3 데이터. "1000" : CQ4 데이터. "1001" : 0. "1010" : 타이머 #1 현재 카운터 값. "1011" : 타이머 #2 현재 카운터 값. "1100" : 인터럽트 flag #1 "1101" : 인터럽트 flag #2 "1110" : 현재 구동 드라이브로 출력한 펄스수 "1111" : 현재 속도
[7~4]	스크립트 #2의 명령어 실행시 사용할 실행 데이터 지정
[11~8]	스크립트 #3의 명령어 실행시 사용할 실행 데이터 지정
[31~12]	스크립트 #4의 명령어 실행시 사용할 실행 데이터 지정

► [SCRSTAT1][0x55] : 스크립트 기능 블록의 현재 상태 표시(32bit)

스크립트 기능 블록의 현재 상태를 확인한다.

[SCRSTAT1]	Description
[0]	스크립트 #1 실행 Queue empty
[1]	스크립트 #1 실행 Queue full
[2]	스크립트 #1 읽기 결과 Queue empty
[3]	스크립트 #1 읽기 실행 Queue full
[4]	스크립트 #2 실행 Queue empty
[5]	스크립트 #2 실행 Queue full
[6]	스크립트 #2 읽기 결과 Queue empty
[7]	스크립트 #2 읽기 실행 Queue full
[8]	스크립트 #3 실행 레지스터 비어 있음.
[9]	스크립트 #3 실행 레지스터 예약 됨.
[10]	스크립트 #3 읽기 결과 레지스터 비어 있음.
[11]	스크립트 #3 읽기 결과 레지스터 새로운 데이터로 갱신됨.
[12]	스크립트 #4 실행 레지스터 비어 있음.
[13]	스크립트 #4 실행 레지스터 예약 됨.
[14]	스크립트 #4 읽기 결과 레지스터 비어 있음.
[15]	스크립트 #4 읽기 결과 레지스터 새로운 데이터로 갱신됨.
[19~16]	스크립트 #1 실행 Queue 데이터 개수(0~15)
[23~20]	스크립트 #1 읽기 실행 결과 Queue 데이터 개수(0~15)
[27~24]	스크립트 #2 실행 Queue 데이터 개수(0~15)
[31~28]	스크립트 #2 읽기 실행 결과 Queue 데이터 개수(0~15)

► [SCRSTAT2][0x56] : 스크립트 기능 블록의 현재 상태 표시(16bit)

스크립트 기능 블록 실행 명령어 예약 Queue와 읽기 결과 값 저장 Queue의 현재 데이터 개수를 확인한다.

[SCRSTAT2]	Description
[3~0]	스크립트 #1 실행 Queue 데이터 개수(0~15)
[7~4]	스크립트 #1 읽기 실행 결과 Queue 데이터 개수(0~15)
[11~8]	스크립트 #2 실행 Queue 데이터 개수(0~15)
[15~12]	스크립트 #2 읽기 실행 결과 Queue 데이터 개수(0~15)

► [INITSQ][0xD7] : 스크립트 기능 블록의 실행 명령어 예약 초기화(4bit)

스크립트 기능 블록의 실행 명령어 예약 내용을 초기화 한다. 스크립트 #1~#4을 지정하여 초기화 하는데 각 비트의 내용은 다음과 같다.

[INITSQ]	Description
[0]	스크립트 #1 실행 명령어 예약 Queue 초기화.
[1]	스크립트 #2 실행 명령어 예약 Queue 초기화.
[2]	스크립트 #3 실행 명령어 예약 레지스터 초기화.
[3]	스크립트 #4 실행 명령어 예약 레지스터 초기화.

► [INITCQ][0xD8] : 스크립트 기능 블록의 읽기 실행 결과 초기화(4bit)

스크립트 기능 블록의 읽기 예약 명령어 실행 결과를 저장하는 Queue 및 레지스터 내용을 초기화 한다. 스크립트 #1~#4을 지정하여 초기화 하는데 각 비트의 내용은 다음과 같다.

[INITCQ]	Description
[0]	스크립트 #1 읽기 실행 결과 Queue 초기화.
[1]	스크립트 #2 읽기 실행 결과 Queue 초기화.
[2]	스크립트 #3 읽기 실행 결과 레지스터 초기화.

[3]	스크립트 #4 읽기 실행 결과 레지스터 초기화.
-----	----------------------------

▶ **[SCRM][0x59/0xD9] : 스크립트 예약 명령어 이벤트 감시 시작 설정(4bit/8bit)**

예약된 스크립트 명령어를 이벤트 감시를 시작할 것인가 멈출 것인가를 정하는 명령어이다.

[SCRM](Read)	Description
[0]	스크립트 #1 이벤트 감시 시작 유무('0' : Off, '1' : On)
[1]	스크립트 #2 이벤트 감시 시작 유무('0' : Off, '1' : On)
[2]	스크립트 #3 이벤트 감시 시작 유무('0' : Off, '1' : On)
[3]	스크립트 #4 이벤트 감시 시작 유무('0' : Off, '1' : On)

[SCRM] 은 DATA port 내용으로 변경 할 수도 있고, 현재의 값을 읽어 보지 않고 현재 설정되어 있는 값과 DATA port 내용을 연산하여 원하는 비트만 변경할 수 있다.

[SCRM] (Write)	Description
[0]	스크립트 #1 이벤트 감시 시작 유무 설정 데이터(operand)
[1]	스크립트 #2 이벤트 감시 시작 유무 설정 데이터(operand)
[2]	스크립트 #3 이벤트 감시 시작 유무 설정 데이터(operand)
[3]	스크립트 #4 이벤트 감시 시작 유무 설정 데이터(operand)
[6~4]	Don't care
[8~7]	현재 설정값과 operand값 사이 연산 종류 설정. "00": 연산하지 않음, 각 operand값을 [SCRM]으로 설정. "01": 각 비트 OR 연산 "10": 각 비트 AND 연산 "11": 각 비트 XOR 연산

▶ **[SQ1POP][0xDA] : 스크립트 #1 실행 명령어 예약 Queue POP**

스크립트 #1의 [SCRCON1], [SCR CMD1], [SCR DAT1] 예약 Queue의 실행 대기 데이터를 1개 지운다.

▶ **[SQ2POP][0xDB] : 스크립트 #2 실행 명령어 예약 Queue POP**

스크립트 #2의 [SCRCON2], [SCR CMD2], [SCR DAT2] 예약 Queue의 실행 대기 데이터를 1개 지운다.

5.3.7.Commands for setting counter and comparator configuration.

측당 5개의 범용 카운터와 비교기에 관한 설정 명령들로 구성된다.

▶ **[CNTLB][0x61/0xE1] : 범용 카운터 하한 값(28bit)**

범용 카운터의 범위를 설정 할 수 있는데 기본적으로는 -134,217,728~134,217,727(0x80000000~0x7FFFFFFF)이지만, [CNTCF2]의 설정에 따라 그 범위가 [CNTCL]~[CNTUB]가 된다. [CNTLB]의 설정범위는 -134,217,728(0x80000000) ~ [CNTUB] 이다. 리셋 후 초기 값은 -134,217,728(0x80000000)이다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4bit는 27bit로 sign-extention 된다.

▶ **[CNTUB][0x62/0xE2] : 범용 카운터 상한 값(28bit)**

[CNTUB]의 설정범위는 [CNTLB] ~ 134,217,727(0x80000000~0x7FFFFFFF) 이다. 리셋 후 초기 값은 134,217,727(0x7FFFFFFF)이다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4bit는 27bit로 sign-extention 된다.

▶ **[CNTCF1][0x63/0xE3] : 범용 카운터 입력 선택/카운터 이벤트 설정(32bit)**

범용 카운터의 카운터 소스를 설정하고 카운터가 이벤트 카운터로 동작할 때 사용할 이벤트를 선택한다.

[CNTCF1]	Description
[2~0]	카운터 #1 입력 선택 “000”: 출력 펄스 “001”: 외부 인코더(ECUP/ECDN) 입력 “010”: 외부 MPG(EXPP/EXMP) 입력 “011”: [CNTCF1][23~16]에서 설정된 이벤트 “100”: [CNTCF1][31~24]에서 설정된 이벤트 “110”: 카운터 #1의 값에 [CURSPD]값을 반영 others: 카운터 하지 않음.
[5~3]	카운터 #2 입력 선택 “000”: 출력 펄스 “001”: 외부 인코더(ECUP/ECDN) 입력 “010”: 외부 MPG(EXPP/EXMP) 입력 “011”: [CNTCF1][23~16]에서 설정된 이벤트 “100”: [CNTCF1][31~24]에서 설정된 이벤트 “110”: 카운터 #2의 값에 [CURSPD]값을 반영 others: 카운터 하지 않음.
[8~6]	카운터 #3 입력 선택 “000”: 출력 펄스 “001”: 외부 인코더(ECUP/ECDN) 입력 “010”: 외부 MPG(EXPP/EXMP) 입력 “011”: [CNTCF1][23~16]에서 설정된 이벤트 “100”: [CNTCF1][31~24]에서 설정된 이벤트 “110”: 카운터 #3의 값에 [CURSPD]값을 반영 others: 카운터 하지 않음.
[11~9]	카운터 #4 입력 선택 “000”: 출력 펄스 “001”: 외부 인코더(ECUP/ECDN) 입력 “010”: 외부 MPG(EXPP/EXMP) 입력 “011”: [CNTCF1][23~16]에서 설정된 이벤트 “100”: [CNTCF1][31~24]에서 설정된 이벤트 “111”: 카운터 #4의 값에 ([CNT1]-[CNT2])값을 반영 others: 카운터 하지 않음.
[14~12]	카운터 #5 입력 선택 “000”: 출력 펄스 “001”: 외부 인코더(ECUP/ECDN) 입력 “010”: 외부 MPG(EXPP/EXMP) 입력 “011”: [CNTCF1][23~16]에서 설정된 이벤트 “100”: [CNTCF1][31~24]에서 설정된 이벤트 “110”: 카운터 #5의 값에 [CURSPD]값을 반영 others: 카운터 하지 않음.
[15]	Don't care
[23~16]	카운터가 이벤트 카운터로 동작 시 사용할 이벤트 선택 #1(스크립트 이벤트 내역 참조)
[31~24]	카운터가 이벤트 카운터로 동작 시 사용할 이벤트 선택 #2(스크립트 이벤트 내역 참조)

► [CNTCF2][0x64/0xE4]: 범용 카운터 초기화 조건 설정/범위 설정(32bit)

범용 카운터는 설정된 조건일 때 초기화(초기 값[0]으로 변경)되는데 이 기능을 위한 초기화 조건을 설정하고 카운터 범위를 [CNTCL]~[CNTUB]로 사용할 것인가를 설정한다.

[CNTCF2]	Description
[4~0]	카운터 #1 초기화 조건 설정. “00000”: 초기화 기능 사용하지 않음. “00001”: 인터럽트 출력이 설정된 구동 종료 이벤트 발생 “00010”: 구동 종료 이벤트 발생. “00011”: 구동 시작 이벤트 발생. “00100”: 카운터 #1 < 비교기 #1 이벤트 발생.

	“00101”: 카운터 #1 = 비교기 #1 이벤트 발생. “00110”: 카운터 #1 > 비교기 #1 이벤트 발생. “00111”: 카운터 #2 < 비교기 #2 이벤트 발생. “01000”: 카운터 #2 = 비교기 #2 이벤트 발생. “01001”: 카운터 #2 > 비교기 #2 이벤트 발생. “01010”: 카운터 #3 < 비교기 #3 이벤트 발생. “01011”: 카운터 #3 = 비교기 #3 이벤트 발생. “01100”: 카운터 #3 > 비교기 #3 이벤트 발생. “01101”: 카운터 #4 < 비교기 #4 이벤트 발생. “01110”: 카운터 #4 = 비교기 #4 이벤트 발생. “01111”: 카운터 #4 > 비교기 #4 이벤트 발생. “10000”: 카운터 #5 < 비교기 #5 이벤트 발생. “10001”: 카운터 #5 = 비교기 #5 이벤트 발생. “10010”: 카운터 #5 > 비교기 #5 이벤트 발생. “10011”: 타이머 #1 이벤트 발생. “10100”: 타이머 #2 이벤트 발생. “10101”: 구동 예약 queue가 모두 실행되어 비어짐. “10110”: 구동 예약 queue가 가득 찼음. “10111”: 트리거 거리 주기/절대 위치 queue 비어짐. “11000”: 트리거 거리 주기/절대 위치 queue 가득 찼음. “11001”: 트리거 발생 이벤트. “11010”: 스크립트 #1 명령어 예약 queue 비어짐. “11011”: 스크립트 #2 명령어 예약 queue 비어짐. “11100”: 스크립트 #3 명령어 예약 레지스터 비어짐. “11101”: 스크립트 #4 명령어 예약 레지스터 비어짐. “11110”: 알람 신호 입력 이벤트. “11111”: ABS([CNT1] - [CNT 2]) ≥ 비교기 #4 이벤트 발생.
[9~5]	카운터 #2 초기화 조건 설정(세부 내용 카운터 #1과 동일)
[14~10]	카운터 #3 초기화 조건 설정(세부 내용 카운터 #1과 동일)
[19~15]	카운터 #4 초기화 조건 설정(세부 내용 카운터 #1과 동일)
[24~20]	카운터 #5 초기화 조건 설정(세부 내용 카운터 #1과 동일)
[25]	카운터 #1의 카운터 범위 [CNTLB] ~ [CNTUB]로 설정 유무
[26]	카운터 #2의 카운터 범위 [CNTLB] ~ [CNTUB]로 설정 유무
[27]	카운터 #3의 카운터 범위 [CNTLB] ~ [CNTUB]로 설정 유무
[28]	카운터 #4의 카운터 범위 [CNTLB] ~ [CNTUB]로 설정 유무
[29]	카운터 #5의 카운터 범위 [CNTLB] ~ [CNTUB]로 설정 유무
[30]	‘0’
[31]	‘0’

▶ [CNT1]~[CNT5] [0x66~0x6A/0xE6~0xEA] : 범용 카운터 값(28bit)

다양한 종류의 입력을 카운터 할 수 있는 범용의 스텝 가변 28bit 카운터 값이다. 카운터의 현재 값을 직접 바꾸거나 현재 동작중인 카운터 값을 읽어 볼 수 있다. [CNT1]~[CNT5]은 리셋 후 출력 펄스를 카운터 하도록 설정되며, 카운터 스텝은 1이 되고 값은 0x00000000이 된다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4bit는 27bit로 sign-extension 된다

▶ [CNTC1]~[CNTC5][0x6B~0x6F/0xEB~0xEF] : 범용 카운터 비교기 데이터 값(28bit)

각 범용 카운터 마다 비교기 기능이 있는데 이때 사용할 비교 데이터 값을 설정하고 확인 할 수 있다. [CNTC1]~[CNTC5]은 리셋 후 값은 0x00000000이 된다. 설정 시 데이터 포트 (DATAPM1)의 상위 4bit(7~4 bit)는 Don't care로 처리되며 읽기 명령어 수행 시 상위 4bit는 27bit로 sign-extension 된다

5.3.8.Commands for setting environments and checking status.

CAMC-QI 는 외부 기계 계 리미트 센서 및 원점 신호, 서버 드라이브의 인코더 신호, 알람, 구동 완료 등 여러 가지 입출력 신호를 사용하는데 이때 신호선 마다 사용하는 레벨 및 유효 에지 등을 설정해야 하며, 기타 부가 기능을 사용하기 위하여 초기 사용 모드 등을 설정하여야 한다. 또한 현재 입력 신호 상태/비교기 결과/구동 상태 등을 표

시하는 status 를 확인할 수 있다.

► [UCFG1][0x70/0xF0] : 입력 신호 사용 레벨/입력 신호 기능 사용 유무/원점 검색 모드 등 설정(29bit)

입력 신호의 사용 레벨을 설정하고 각 입력 신호 별 기능 사용 유무를 설정한다. 또한 원점 검색 및 신호 검색에 사용할 Z 상 입력 신호에 대한 설정과 원점 검색 모드 등을 설정한다.

[UCFG1]	Description
[0]	정방향 리미트(PELM) 신호 입력 레벨 설정.
[1]	역방향 리미트(NELM) 신호 입력 레벨 설정.
[2]	부가 정방향 감속 정지 리미트 신호 입력 레벨 설정.
[3]	부가 역방향 감속 정지 리미트 신호 입력 레벨 설정.
[4]	서보 알람 신호 입력 레벨 설정.
[5]	서보 위치 결정 완료(Inposition)신호 입력 레벨 설정.
[6]	비상 정지 입력 신호 레벨 설정.
[7]	원점 입력 신호 레벨 설정.
[8]	Z 상 입력 신호 레벨 설정.
[9]	정방향 구동 중 정방향 리미트(PELM) 신호 인가시 정지 유무.
[10]	역방향 구동 중 역방향 리미트(NELM) 신호 인가시 정지 유무.
[11]	정방향 구동 중 부가 정방향 감속 정지 리미트 신호 인가시 감속 정지 유무.
[12]	역방향 구동 중 부가 역방향 감속 정지 리미트 신호 인가시 감속 정지 유무.
[13]	구동 중 알람 입력 인가시 정지 유무.
[14]	구동 중 비상 정지 신호 인가시 급정지 유무.
[15]	신호 검색 및 원점 검색 시 사용할 Z 상 신호 예지 선택
[19~16]	원점 검색 시 Z 상 예지 검색 개수(0~15)
[20]	원점 검색 완료시 카운터 #1 초기화 유무
[21]	원점 검색 완료시 카운터 #2 초기화 유무
[22]	원점 검색 완료시 카운터 #3 초기화 유무
[23]	원점 검색 완료시 카운터 #4 초기화 유무
[24]	원점 검색 완료시 카운터 #5 초기화 유무
[28~25]	원점 검색 모드 설정.
[29]	고급 보간 구동 감속 모드

► [UCFG2][0x71/0xF1] : 신호 검색 설정/입력 신호 필터 설정 (29bit)

신호 검색 신호 선택 및 입력 신호들의 디지털 노이즈 필터를 설정한다.

[UCFG2]	Description
[0]	신호 검색 구동 종료시 카운터 #1 초기화 유무
[1]	신호 검색 구동 종료시 카운터 #2 초기화 유무
[2]	신호 검색 구동 종료시 카운터 #3 초기화 유무
[3]	신호 검색 구동 종료시 카운터 #4 초기화 유무
[4]	신호 검색 구동 종료시 카운터 #5 초기화 유무
[7~5]	신호 검색 시 사용할 신호 선택 “000”: UIO5(ORG) 입력. “001”: UIO6(Z 상) 입력. “010”: UIO7 입력. “011”: UIO8 입력. “100”: UIO9 입력. “101”: UIO10 입력. “110”: UIO11 입력. “111”: INPOS 입력.
[8]	신호 검색 신호 예지 선택(0: 선택 신호 하향 예지 검색, 1: 선택 신호 상향 예지 검색)
[12~9]	정/역 방향 리미트 신호의 디지털 필터 상수 설정.
[16~13]	서보 위치 결정 완료신호/서보 알람신호 디지털 필터 상수 설정.
[20~17]	UIO0~4/UIO7~9 가 입력으로 설정 되었을 때 디지털 필터 상수 설정.

[24~21]	UIO5~6가 입력으로 설정되었을 때 디지털 필터 상수 설정.
[28~25]	UIO10~11가 입력으로 설정되었을 때 디지털 필터 상수 설정.

디지털 필터 설정	필터 신호 폭(설정 폭 보다 작은 신호 무시됨.)	
"0000"	8	* Tclk
"0001"	128	* Tclk
"0010"	512	* Tclk
"0011"	2048	* Tclk
"0100"	8192	* Tclk
"0101"	32768	* Tclk
"0110"	131072	* Tclk
"0111"	524288	* Tclk
"1000"	64	* Tclk
"1001"	256	* Tclk
"1010"	1024	* Tclk
"1011"	4092	* Tclk
"1100"	16384	* Tclk
"1101"	65536	* Tclk
"1110"	262144	* Tclk
"1111"	1048576	* Tclk

▶ [UCFG3][0x72/0xF2] : 출력 펄스 방식/소프트웨어 리미트 기능 설정/MPG 기능 설정/공용 입력 신호 필터 설정(X축 한정) (32bit)

출력 펄스 형태를 설정하고 소프트웨어 리미트 기능에 관한 설정 및 MPG 기능 설정을 수행한다. 특히 X축 에서만 유효하게 공용 신호들(비상 정지 신호(estop), 동기 정지(sqstp1/2), 동기 시작(sqstr1/2))의 디지털 필터를 설정한다.

[UCFG3]	Description
[2~0]	출력 펄스 방식 설정.*
[3]	출력 펄스 신호 레벨 설정.*
[4]	정방향 소프트웨어 리미트 사용 유무.(‘0’: 사용하지 않음, ‘1’: 사용함)
[5]	역방향 소프트웨어 리미트 사용 유무.(‘0’: 사용하지 않음, ‘1’: 사용함)
[6]	정방향 소프트웨어 리미트 동작 방법 설정(‘0’: 급정지, ‘1’: 감속정지)
[7]	역방향 소프트웨어 리미트 동작 방법 설정(‘0’: 급정지, ‘1’: 감속정지)
[9~8]	소프트웨어 리미트 기준 카운터 설정 “00”: 카운터 #1 기준 “01”: 카운터 #2 기준 “10”: 카운터 #3 기준 “11”: 카운터 #4 기준
[15~10]	MPG(수동 펄스 발생 장치 입력)구동 시 곱하기 값(입력 값에 +1로 계산됨)
[27~16]	MPG(수동 펄스 발생 장치 입력)구동 시 나누기 값(입력 값에 +1로 계산됨)
[31~28]	공용신호(ESTOP, SQSTP1/2, SQSTR1/2) 의 디지털 필터 설정(X 축에만 유효)

표 5-3. 출력 펄스 형태

[UCFG3][2~0]	[UCFG3][3] = ‘0’	[UCFG3][3] = ‘1’
“000” (One pulse mode)	<p>Active High pulse</p> <p>Level 'Low' Level 'High'</p>	<p>Active High pulse</p> <p>Level 'High' Level 'Low'</p>

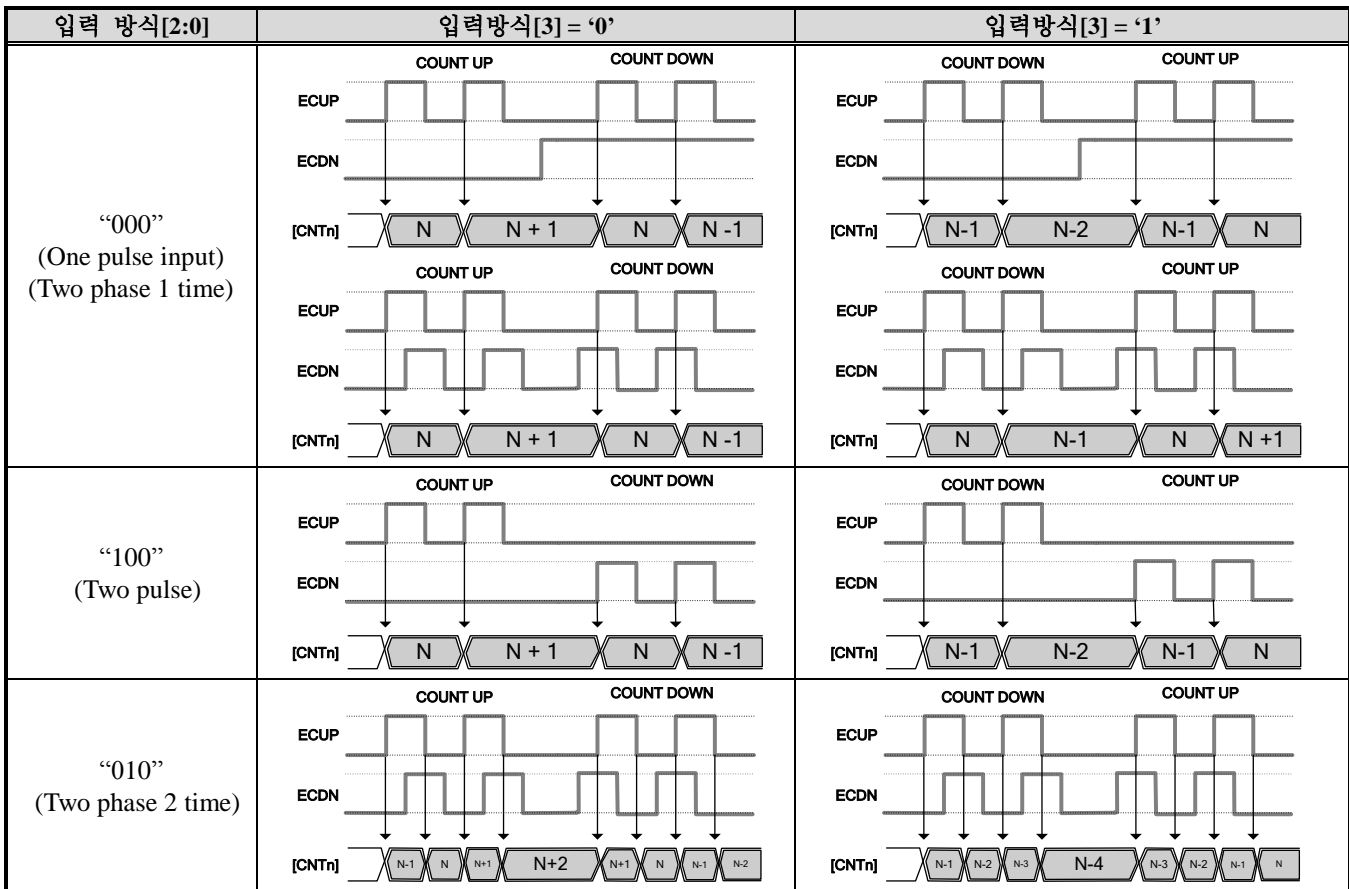
<p>“001” (One pulse mode)</p>	<p>CW CCW</p>	<p>CW CCW</p>
<p>“010” (Two pulse mode)</p>	<p>CW CCW</p>	<p>CW CCW</p>
<p>“011” (Two pulse mode)</p>	<p>CW CCW</p>	<p>CW CCW</p>
<p>“100” (Two phase mode)</p>	<p>CW CCW</p>	<p>CW CCW</p>

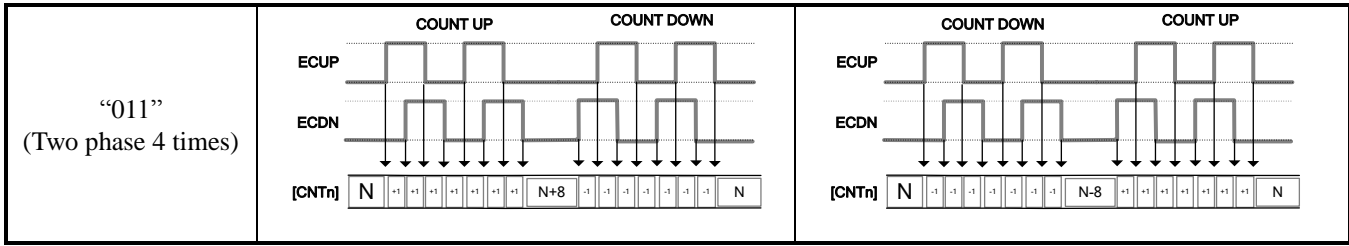
▶ [UCFG4][0x73/0xF3] : MPG(EXPP, EXMP) 입력 설정/Encoder(ECUP,ECDN) 입력 설정/UIO one-shot 펄스 시간 설정/서버 잔여 펄스 초기화 기능 설정/에러 발생시 구동 예약 Queue 초기화 설정/동기 정지 모드 설정/리미트 동작 설정/알람 입력 설정 (30bit)

[UCFG4]	Description
[3~0]	MPG 신호의 입력 방식 설정.*
[5~4]	MPG 입력 신호 필터 설정 “00” : 4 clock depth. (9.83040 MHz when $t_{CLK} = 39.3216\text{MHz}$) “01” : 6 clock depth. (6.55360 MHz when $t_{CLK} = 39.3216\text{MHz}$) “10” : 10 clock depth. (3.93216 MHz when $t_{CLK} = 39.3216\text{MHz}$) “11” : 18 clock depth. (2.18453 MHz when $t_{CLK} = 39.3216\text{MHz}$)
[9~6]	Encoder 신호의 입력 방식 설정.*
[11~10]	Encoder 입력 신호 필터 설정 “00” : 4 clock depth. (9.83040 MHz when $t_{CLK} = 39.3216\text{MHz}$) “01” : 6 clock depth. (6.55360 MHz when $t_{CLK} = 39.3216\text{MHz}$) “10” : 10 clock depth. (3.93216 MHz when $t_{CLK} = 39.3216\text{MHz}$) “11” : 18 clock depth. (2.18453 MHz when $t_{CLK} = 39.3216\text{MHz}$)
[13~12]	UIO3 one-shot 펄스 폭 설정 “00” : $T_{clk} * 3932160$ (ex. When $F_{clk} = 39.3216\text{MHz} \rightarrow 1\text{mS}$) “01” : $T_{clk} * 7864320$ (ex. When $F_{clk} = 39.3216\text{MHz} \rightarrow 2\text{mS}$) “10” : $T_c * 15728640$ (ex. When $F_{clk} = 39.3216\text{MHz} \rightarrow 4\text{mS}$) “11” : $T_c * 31457280$ (ex. When $F_{clk} = 39.3216\text{MHz} \rightarrow 8\text{mS}$)
[15~14]	UIO4 one-shot 펄스 폭 설정(UIO3와 동일 내용)

[17~16]	서버 잔여펄스 제거 출력(CLR) 신호 OFF 시간 설정. “00”: No wait “01”: 10 uSec “10”: 1 mSec “11”: 100 mSec
[20~18]	서버 잔여펄스 제거 출력(CLR) 신호 펄스 폭 설정. “000”: 10 uSec “001”: 100 uSec “010”: 500 uSec “011”: 1 mSec “100”: 10 mSec “101”: 50 mSec “110”: 100 mSec “111”: Logic level(reset by CMD[0x60])
[21]	서버 잔여펄스 제거 출력(CLR) 신호 사용 레벨 설정.
[22]	리미트/알람/비상 정지/동기정지 신호에 의한 정지 시 서버 잔여 펄스 제거 신호 출력 유무
[23]	원점 신호 검색 완료시 서버 잔여펄스 제거 출력 유무
[24]	리미트/알람/비상정지/동기정지시 구동 설정 예약 Queue 초기화 유무
[25]	급정지 명령어/모든 축 급정지 명령어 수행 시 구동 설정 예약 Queue 초기화 유무
[26]	SQSTP1에 의한 동기 정지 기능 사용 설정(‘0’: 급정지, ‘1’: 감속 정지)
[27]	SQSTP2에 의한 동기 정지 기능 사용 설정(‘0’: 급정지, ‘1’: 감속 정지)
[28]	리미트 신호 기능 설정(‘0’: 신호 입력 시 급정지, ‘1’: 감속 정지)
[29]	알람 정지 기능 설정(‘0’: 해당 축 알람 신호 인가시, ‘1’: 전 축 알람 중 하나 이상 인가시)

표 5-4. 인코더/MPG 펄스 입력형태





► [IMASK1][0x77/0xF7] : 인터럽트 #1 그룹의 각 인터럽트 조건에 대한 사용 유무 설정(32bit)

[IMASK1]	Description
[0]	인터럽트 발생 사용 설정된 구동 종료시.
[1]	구동 종료시.
[2]	구동 시작 시.
[3]	카운터 #1 < 비교기 #1 이벤트 발생.
[4]	카운터 #1 = 비교기 #1 이벤트 발생.
[5]	카운터 #1 > 비교기 #1 이벤트 발생.
[6]	카운터 #2 < 비교기 #2 이벤트 발생.
[7]	카운터 #2 = 비교기 #2 이벤트 발생.
[8]	카운터 #2 > 비교기 #2 이벤트 발생.
[9]	카운터 #3 < 비교기 #3 이벤트 발생.
[10]	카운터 #3 = 비교기 #3 이벤트 발생.
[11]	카운터 #3 > 비교기 #3 이벤트 발생.
[12]	카운터 #4 < 비교기 #4 이벤트 발생.
[13]	카운터 #4 = 비교기 #4 이벤트 발생.
[14]	카운터 #4 > 비교기 #4 이벤트 발생.
[15]	카운터 #5 < 비교기 #5 이벤트 발생.
[16]	카운터 #5 = 비교기 #5 이벤트 발생.
[17]	카운터 #5 > 비교기 #5 이벤트 발생.
[18]	타이머 #1 이벤트 발생.
[19]	타이머 #2 이벤트 발생.
[20]	구동 예약 설정 Queue 비워짐.
[21]	구동 예약 설정 Queue 가득 찼.
[22]	트리거 발생 거리 주기/절대위치 Queue 비워짐.
[23]	트리거 발생 거리 주기/절대위치 Queue 가득 찼.
[24]	트리거 신호 발생 이벤트.
[25]	스크립트 #1 명령어 예약 설정 Queue 비워짐.
[26]	스크립트 #2 명령어 예약 설정 Queue 비워짐.
[27]	스크립트 #3 명령어 예약 설정 레지스터 실행되어 초기화 됨.
[28]	스크립트 #4 명령어 예약 설정 레지스터 실행되어 초기화 됨.
[29]	서보 알람신호 인가됨.
[30]	[CNT1] - [CNT2] ≥ [CNTC4] 이벤트 발생.
[31]	인터럽트 발생 명령어[INTGEN] 실행.

► [IMASK2][0x78/0xF8] : 인터럽트 #2 그룹의 각 인터럽트 조건에 대한 사용 유무 설정(32bit)

[IMASK2]	Description
[0]	스크립트 #1 읽기 명령 결과 Queue 가 가득 찼.
[1]	스크립트 #2 읽기 명령 결과 Queue 가 가득 찼.
[2]	스크립트 #3 읽기 명령 결과 레지스터가 새로운 데이터로 갱신됨.
[3]	스크립트 #4 읽기 명령 결과 레지스터가 새로운 데이터로 갱신됨.
[4]	스크립트 #1의 예약 명령어 중 실행 시 인터럽트 발생으로 설정된 명령어 실행됨.
[5]	스크립트 #2의 예약 명령어 중 실행 시 인터럽트 발생으로 설정된 명령어 실행됨.
[6]	스크립트 #3의 예약 명령어 실행 시 인터럽트 발생으로 설정된 명령어 실행됨.
[7]	스크립트 #4의 예약 명령어 실행 시 인터럽트 발생으로 설정된 명령어 실행됨.
[8]	구동 시작.

[9]	서보 위치 결정 완료(Inposition) 기능을 사용한 구동 종료 조건 발생.
[10]	이벤트 카운터로 동작 시 사용할 이벤트 선택 #1 조건 발생
[11]	이벤트 카운터로 동작 시 사용할 이벤트 선택 #2 조건 발생
[12]	SQSTR1 신호 인가 됨.
[13]	SQSTR2 신호 인가 됨.
[14]	UIO0 터미널 신호가 '1'로 변함.
[15]	UIO1 터미널 신호가 '1'로 변함.
[16]	UIO2 터미널 신호가 '1'로 변함.
[17]	UIO3 터미널 신호가 '1'로 변함.
[18]	UIO4 터미널 신호가 '1'로 변함.
[19]	UIO5 터미널 신호가 '1'로 변함.
[20]	UIO6 터미널 신호가 '1'로 변함.
[21]	UIO7 터미널 신호가 '1'로 변함.
[22]	UIO8 터미널 신호가 '1'로 변함.
[23]	UIO9 터미널 신호가 '1'로 변함.
[24]	UIO10 터미널 신호가 '1'로 변함.
[25]	UIO11 터미널 신호가 '1'로 변함.
[26]	오류 정지 조건(LMT, ESTOP, STOP, ESTOP CMD,ALARM) 발생
[27]	보간 중 데이터 설정 오류 발생.
[28]	Don't care
[29]	리미트 신호(PELM, NELM)신호가 입력 됨.
[30]	부가 리미트 신호(PSLM, NSLM)신호가 입력 됨.
[31]	비상 정지 신호(ESTOP)신호가 입력됨.

▶ **[STAT1]/[ESCLR][0x79/0xF9] : 구동 종료 결과 표시, 오류 발생 표시/종료 원인 상태 초기화(32bit)**

[STAT1][21~0]는 구동 중 비정상 원인에 의하여 종료되었다는 정보/신호검색,원점검색 성공 정보를 표시한다. 해당 종료 조건이 한번이라도 발생하면 해당 비트는 [ESCLR]가 실행 될 때까지 '1' 을 유지한다. [STAT1][31~28]

[STAT1]	Description
[0]	정방향 리미트 신호(PELM)에 의한 구동 종료
[1]	역방향 리미트 신호(NELM)에 의한 구동 종료
[2]	정방향 부가 리미트 신호(PSLM)에 의한 구동 종료
[3]	역방향 부가 리미트 신호(NSLM)에 의한 구동 종료
[4]	정방향 소프트 리미트 급정지 기능에 의한 구동 종료
[5]	역방향 소프트 리미트 급정지 기능에 의한 구동 종료
[6]	정방향 소프트 리미트 감속정지 기능에 의한 구동 종료
[7]	역방향 소프트 리미트 감속정지 기능에 의한 구동 종료
[8]	서보 알람 기능에 의한 구동 종료.
[9]	비상 정지 신호 입력에 의한 구동 종료.
[10]	급 정지 명령에 의한 구동 종료
[11]	감속 정지 명령에 의한 구동 종료
[12]	전축 급정지 명령에 의한 구동 종료
[13]	동기 정지 기능 #1(SQSTP1)에 의한 구동 종료.
[14]	동기 정지 기능 #2(SQSTP2)에 의한 구동 종료.
[15]	인코더 입력(ECUP, ECDN) 오류 발생
[16]	MPG 입력(EXPP, EXMP) 오류 발생.
[17]	원점 검색 성공 종료.
[18]	신호 검색 성공 종료.
[19]	보간 데이터 이상으로 구동 종료.
[20]	비정상 구동 정지 발생(alm,estop,limit,data error,stop cmd,estop cmd,all estop cmd)으로 구동 종료.
[21]	MPG 기능 블록 펄스 버퍼 오버플로우 발생.
[27~22]	Don't care
[28]	현재/마지막 구동 드라이브 방향
[29]	잔여 펄스 제거 신호 출력 중.
[30]	비정상 구동 정지 원인 상태. (alm,estop,limit,data error)

[31]	보간 드라이브 데이터 오류 상태.(축 선택 오류, 칩간 직선 보간시 장축 거리 0, 원점 [0,0])
------	--

▶ [STAT2][0x7A]: 현재 구동 상태를 표시.(27bit)

[STAT2]	Description
[0]	구동 중(펄스 출력 중, 펄스 출력 후 서보 위치 완료 신호 대기 중)
[1]	감속 구동 중
[2]	등속 구동 중
[3]	가속 구동 중
[4]	연속 드라이브 구동 중
[5]	지정 거리 드라이브 구동 중
[6]	MPG 드라이브 구동 중
[7]	원점검색 드라이브 구동 중
[8]	신호 검색 드라이브 구동 중
[9]	보간 드라이브 구동 중
[10]	Slave 드라이브 구동 중
[11]	현재 구동 드라이브 방향(보간 드라이브에서는 표시 정보 다름)
[12]	펄스 출력 후 서보위치 완료 신호 대기중.
[13]	직선 보간 드라이브 구동 중.
[14]	원호 보간 드라이브 구동 중.
[15]	펄스 출력 중.
[18~16]	구동 예약 Queue 데이터 개수(0~7)
[19]	구동 예약 Queue 비어 있음.
[20]	구동 예약 Queue 가득 찼.
[22~21]	현재 구동 드라이브의 속도 모드
[23]	MPG 버퍼 #1 Full
[24]	MPG 버퍼 #2 Full
[25]	MPG 버퍼 #3 Full
[26]	MPG 버퍼 데이터 overflow(MPG 버퍼 #1,#2,#3 Full)
[27]	Don't care
[28]	Don't care
[29]	Don't care
[30]	Don't care
[31]	Don't care

▶ [STAT3][0x7B]: 입력 신호들의 현재 상태를 표시(19bit)

사용자가 사용 레벨을 설정 할 수 있는 신호의 경우 터미널의 신호 레벨과 설정된 사용 레벨의 비교 결과를 표시하고 사용 레벨 설정이 없는 신호의 경우 터미널의 현재 상태를 표시한다.

[STAT3]	Description
[0]	정 방향 리미트 신호(PELM) 현재 상태.
[1]	역 방향 리미트 신호(NELM) 현재 상태.
[2]	부가 정 방향 감속 정지 리미트 신호(PSLM) 현재 상태.
[3]	부가 역 방향 감속 정지 리미트 신호(NSLM) 현재 상태.
[4]	서보 알람(ALARM) 신호 현재 상태.
[5]	서보 위치 결정 완료 신호(INPOS) 현재 상태.
[6]	비상 정지 신호(ESTOP) 현재 상태.
[7]	원점 신호(ORG)현재 상태.
[8]	Z 상 입력 신호 현재 상태.
[9]	ECUP 터미널 신호 상태.
[10]	ECDN 터미널 신호 상태.
[11]	EXPP 터미널 신호 상태.
[12]	EXMP 터미널 신호 상태.
[13]	SQSTR1 터미널 신호 상태.
[14]	SQSTR2 터미널 신호 상태.
[15]	SQSTP1 터미널 신호 상태.

[16]	SQSTP2 터미널 신호 상태.
[18]	MODE 터미널 신호 상태.

▶ **[STAT4][0x7C] : 비교기의 현재 결과를 표시(30bit)**

내장된 5개의 범용 카운터에 각 비교 값이 있는데 [STAT4]에서 비교기의 결과를 확인 할 수 있다.

[STAT4]	Description
[0]	비교기 #1의 결과 표시 ([CNT1] < [CNTC1])
[1]	비교기 #1의 결과 표시 ([CNT1] = [CNTC1])
[2]	비교기 #1의 결과 표시 ([CNT1] > [CNTC1])
[3]	비교기 #1의 결과 표시 ([CNT1] ≤ [CNTC1])
[4]	비교기 #1의 결과 표시 ([CNT1] ≥ [CNTC1])
[5]	비교기 #2의 결과 표시 ([CNT2] < [CNTC2])
[6]	비교기 #2의 결과 표시 ([CNT2] = [CNTC2])
[7]	비교기 #2의 결과 표시 ([CNT2] > [CNTC2])
[8]	비교기 #2의 결과 표시 ([CNT2] ≤ [CNTC2])
[9]	비교기 #2의 결과 표시 ([CNT2] ≥ [CNTC2])
[10]	비교기 #3의 결과 표시 ([CNT3] < [CNTC3])
[11]	비교기 #3의 결과 표시 ([CNT3] = [CNTC3])
[12]	비교기 #3의 결과 표시 ([CNT3] > [CNTC3])
[13]	비교기 #3의 결과 표시 ([CNT3] ≤ [CNTC3])
[14]	비교기 #3의 결과 표시 ([CNT3] ≥ [CNTC3])
[15]	비교기 #4의 결과 표시 ([CNT4] < [CNTC4])
[16]	비교기 #4의 결과 표시 ([CNT4] = [CNTC4])
[17]	비교기 #4의 결과 표시 ([CNT4] > [CNTC4])
[18]	비교기 #4의 결과 표시 ([CNT4] ≤ [CNTC4])
[19]	비교기 #4의 결과 표시 ([CNT4] ≥ [CNTC4])
[20]	비교기 #5의 결과 표시 ([CNT5] < [CNTC5])
[21]	비교기 #5의 결과 표시 ([CNT5] = [CNTC5])
[22]	비교기 #5의 결과 표시 ([CNT5] > [CNTC5])
[23]	비교기 #5의 결과 표시 ([CNT5] ≤ [CNTC5])
[24]	비교기 #5의 결과 표시 ([CNT5] ≥ [CNTC5])
[25]	ABS([CNT1] - [CNT2]) < [CNTC4]
[26]	ABS([CNT1] - [CNT2]) = [CNTC4]
[27]	ABS([CNT1] - [CNT2]) > [CNTC4]
[28]	ABS([CNT1] - [CNT2]) ≤ [CNTC4]
[29]	ABS([CNT1] - [CNT2]) ≥ [CNTC4]

▶ **[STAT5][0x7D] : 트리거 및 타이머 상태 표시(8bit)**

트리거 거리 주기/절대 위치 Queue 상태 및 타이머 기능 현재 상태를 표시한다.

[STAT5]	Description
[3~0]	현재 트리거 거리주기/절대위치 Queue 데이터 개수(0~15)
[4]	트리거 거리주기/절대위치 Queue 비어 있음.
[5]	트리거 거리주기/절대위치 Queue 가득 찼.
[6]	타이머 #1 작동 중
[7]	타이머 #2 작동 중

▶ **[IFLAG1][0x7E/0xFE] : 인터럽트 #1 그룹의 발생 상태를 표시(32bit)**

[IMASK1]에서 '1'로 설정된 해당 비트의 조건이 발생하면 [IFLAG1]의 해당 비트가 '1'로 셋 된다. [IFLAG1]의 하나의 비트라도 '1'이 되면 INT 터미널 레벨이 High-Z에서 '0'로 출력 된다. '1'인 [IFLAG1]의 특정 비트를 초기화 하기 위해서는 DATA port에 해당 비트 위치에 '1'을 기입하고 [IFLAG1] 쓰기 명령을 실행 하여야 한다.

▶ **[IFLAG2][0x7F/0xFF] : 트리거 및 타이머 상태 표시(8bit)**

[IMASK2]에서 '1'로 설정된 해당 비트의 조건이 발생하면 [IFLAG2]의 해당 비트가 '1'로 셋 된다. [IFLAG2]의 하

나의 비트라도 ‘1’이 되면 INT 터미널 레벨이 High-Z에서 ‘0’로 출력 된다. ‘1’인 **[IFALG2]**의 특정 비트를 초기화하기 위해서는 DATA port에 해당 비트 위치에 ‘1’을 기입하고 **[IFLAG2]** 쓰기 명령을 실행 하여야 한다.

6. Functional Description

6.1. Position override.

지정 거리 구동 중([DCFG][6~0] = 0x11) [POS] 값이 변경되었을 때 다음 그림과 같은 세 가지의 경우가 발생한다.

6.1.1.정속 구동 중 [POS]값이 기존의 값보다 큰 값으로 변경되었을 때

지정 거리 구동 중 정속도 구간에서 [POS] 값이 기존의 값보다 큰 값으로 변경되면 아래의 그림과 같이 처음 감속하려고 계산되었던 위치보다 이후 새로이 계산된 위치에서 감속을 시작한다.

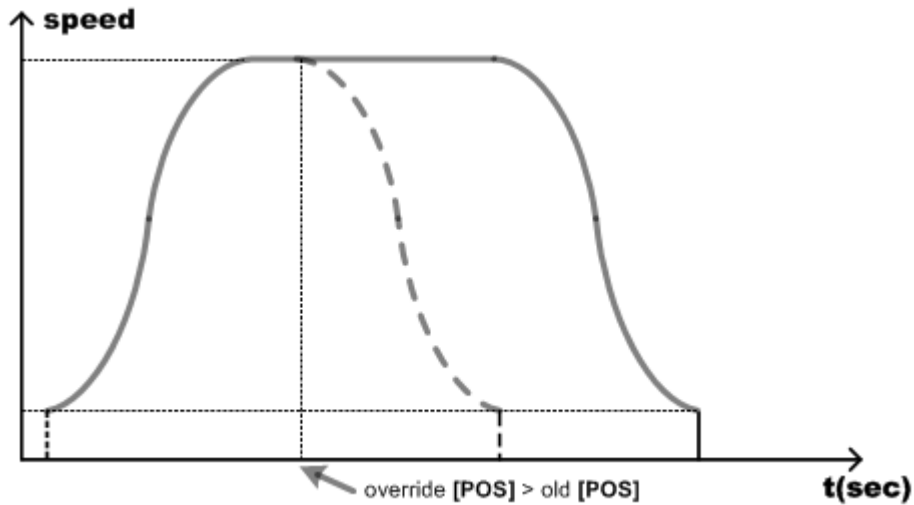


그림 6-1. 거리 override(old [POS] < new [POS])

6.1.2.감속 구동 중 [POS]값이 기존의 값보다 큰 값으로 변경되었을 때

지정 거리 구동 중 감속 구간에서 [POS] 값이 기존의 값보다 큰 값으로 변경되면 아래의 그림과 같이 감속 도중 다시 목표속도로 가속하여 이후 새로이 계산된 위치에서 감속을 시작한다.

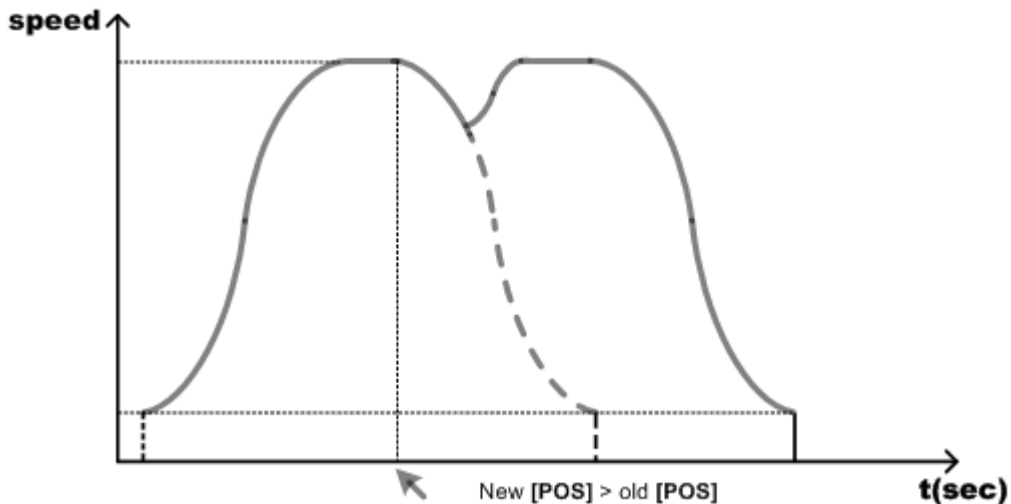


그림 6-2. 거리 override(old [POS] < new [POS])

6.1.3. 구동 중 [POS]값이 기존의 값보다 작은 값으로 변경되었을 때

지정 거리 구동 중 [POS] 값을 기존의 설정된 [POS] 값 보다 작게 설정되고 현재 구동한 거리가 새롭게 설정된 [POS] 값을 넘었을 즉시 감속 반대로 구동하여 새롭게 설정된 [POS] 값에 맞춘다. 단 이 기능을 사용하기 위해서는 구동 중 [DCFG][22]가 '1'로 설정되어야 하며, 그렇지 않을 경우 즉시 정지한다.

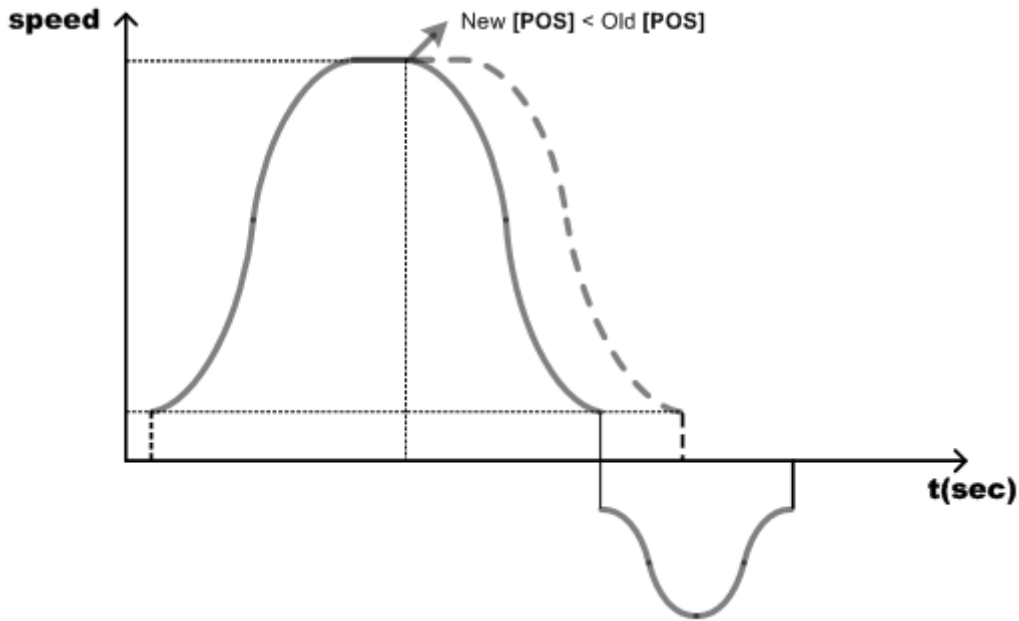


그림 6-3. 거리 override(old [POS] > new [POS])

6.2.원점 구동

원점 센서 검색 구동 중([DCFG][6~0] = 0x21~0x25) ⁴¹ 원점 검색 모드에 따른 구동 프로파일의 형태는 다음과 같다. 특히 원점 검색 완료 후 자동으로 [UCFG1][20~21]에서 설정된 값으로 내부 카운터가 초기화 되기 때문에 실제 기구부의 원점 위치를 정확하게 결정 할 수 있다. 원점 검색 설명 그림에서 다음의 표시된 부분이 [CNTx] 값이 초기화 되는 순간이다.

⚡ [CNTx] clear time/[CLR] signal generation time.

또한 [CNTx] 순간 서버 드라이브의 잔여 펄스를 제거 할 수 있도록 [UCFG4][23] 설정 값에 따라 [CLR] 단자에 출력을 자동으로 생성 할 수 있다.

6.2.1.원점 복귀([DCFG][6~0] = 0x20~0x21)

원점 센서를 향하는 방향으로 하여 구동하는 것으로 원점 모드([UCFG1][28~25])에 따라 다음과 같은 구동 형태를 가진다. 예제로 보여지는 내용의 공통적인 설정은 다음과 같다.

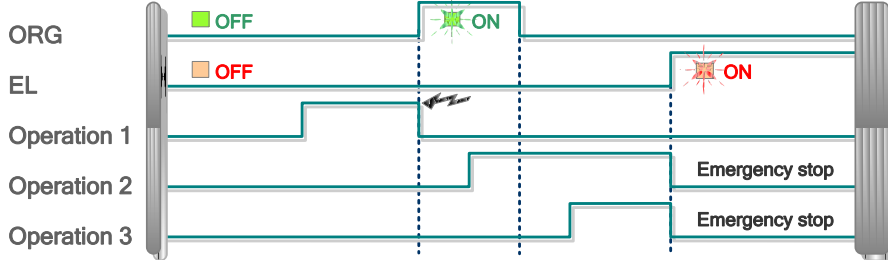
- End limit : [UCFG1][1:0] = "11"
- Org level : [UCFG1][7] = 1
- Z phase input level : [UCFG1][8] = 1
- Z phase input count value : [UCFG1][19:16] = "0001"

► [UCFG1][28~25] : “0000”

구동 시작 후 원점 신호(xIO5)의 설정 레벨로의 상향 에지에서 구동을 정지한다. 속도 프로파일은 구동 시작 종류에 따라 달라진다.

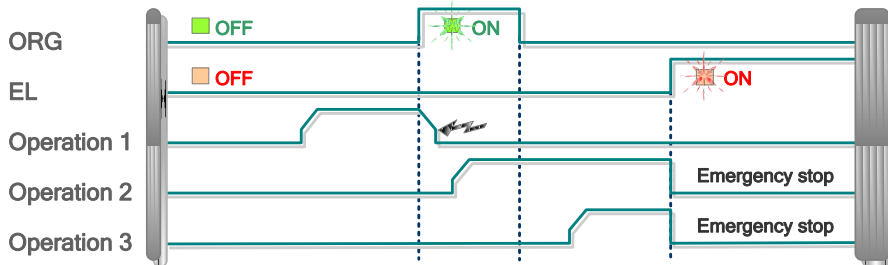
Low Speed[STRCS] operation

< Sensor : ELM ([UCFG1][0~1] = 1), ORG([UCFG1][7] = 1) >



High Speed[STRN] operation

< Sensor : ELM([UCFG1][0~1] = 1), ORG([UCFG1][7] = 1) >

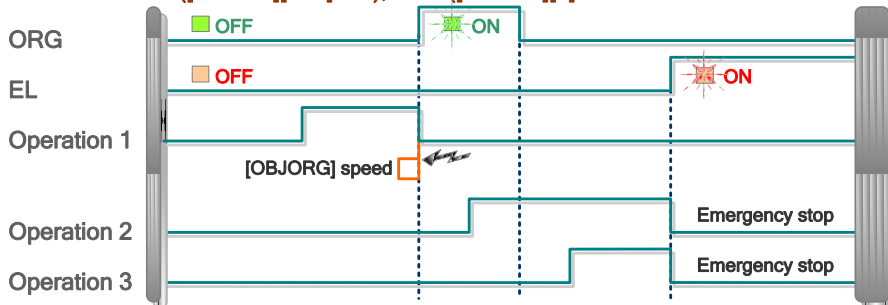


► [UCFG1][28~25] : “0001”

구동 시작 후 원점 신호(xIO5)의 설정 레벨로의 상향 에지에서 구동을 정지한다. 이후 [ORGOBJ]속도로 반대로 구동 후 다시 원점 신호의 하향 에지를 만나면 처음 설정된 구동 방향으로 구동 후 원점 상향 에지에서 구동을 완료한다.

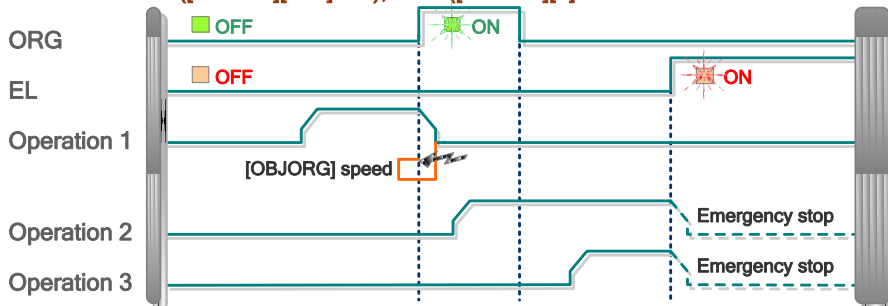
Low[STRCS] Speed operation

< Sensor : ELM([UCFG1][0~1] = 1), ORG([UCFG1][7] = 1) >



High Speed[STRN] operation

< Sensor : ELM([UCFG1][0~1] = 1), ORG([UCFG1][7] = 1) >



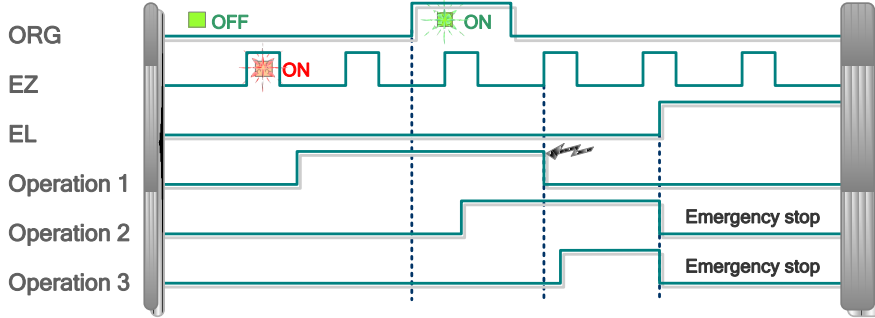
▶ [UCFG1][28~25] : “0010”

구동 시작 후 원점 신호(IO5/ORG)의 설정 레벨로의 상향 에지에서 부터 [UCFG1][19~15] 에서 설정된 만큼(+ 1) EZ 신호의 설정 레벨로의 상향 에지를 만나면 구동을 정지한다.

Low Speed[STRCO] operation

< Sensor : ELM([UCFG1][0~1] = 1), ORG([UCFG1][7] = 1, EZ([UCFG1][8] = 1>

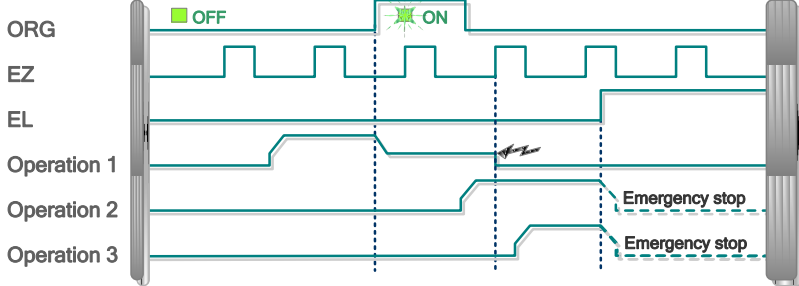
< Z cnt : EZCNT([UCFG1][19:16] = “0001”>



High Speed[STRCN] operation

< Sensor : ELM([UCFG1][0~1] = 1), ORG([UCFG1][7] = 1, EZ([UCFG1][8] = 1>

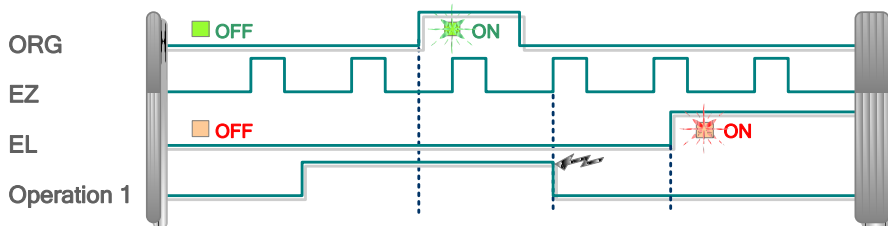
< Z cnt : EZCNT([UCFG1][19:16] = “0001”>

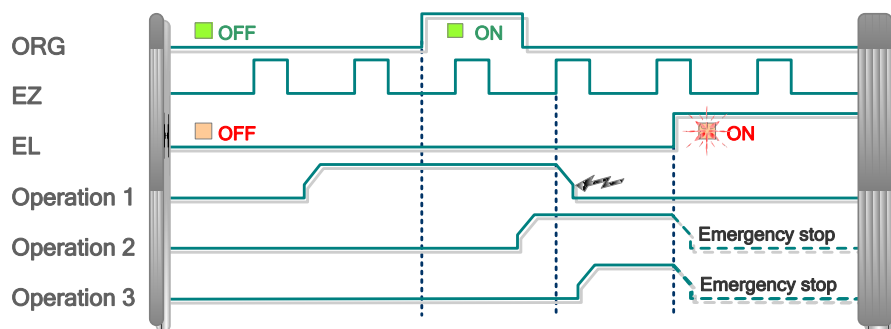


▶ [UCFG1][28~25] : “0011”

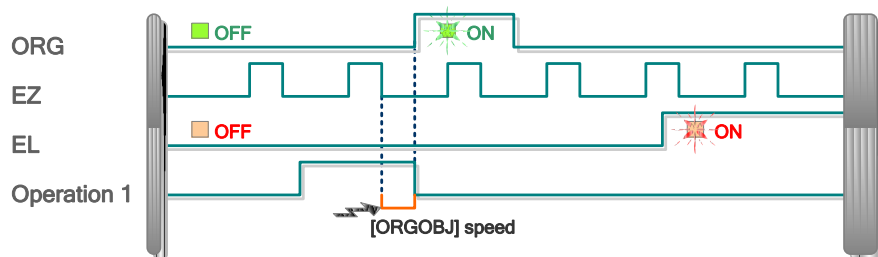
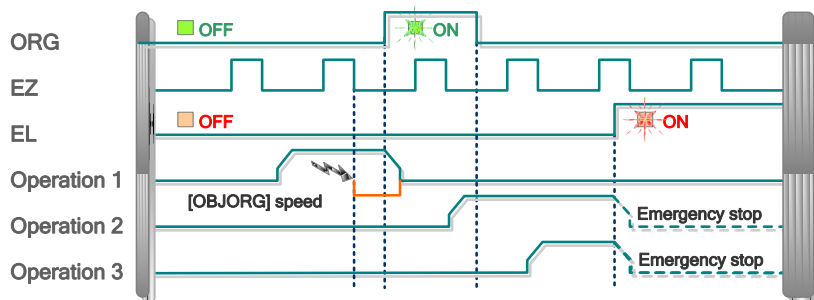
구동 시작 후 원점 신호(IO5/ORG)의 설정 레벨로의 상향 에지에서 부터 [UCFG1][19~15] 에서 설정된 만큼(+ 1) EZ 신호의 설정 레벨로의 상향 에지를 만나면 구동을 정지한다.

Low Speed[STRCS] operation



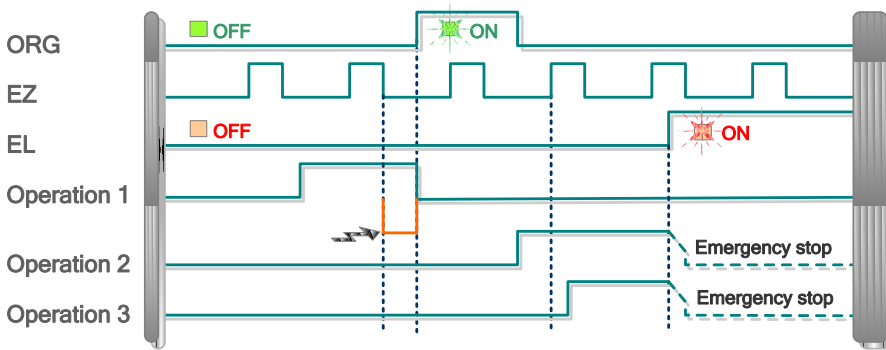
High Speed[STRN] operation**► [UCFG1][28~25] : “0100”**

구동 시작 후 원점 신호(IO5/ORG)의 설정 레벨로의 상향 에지에서 정지후 반대방향으로 [ORGOBJ] 속도로 [UCFG1][19~15] 에서 설정된 만큼(+ 1) EZ 신호의 설정 레벨로의 상향 에지를 만날 때 까지 구동한다.

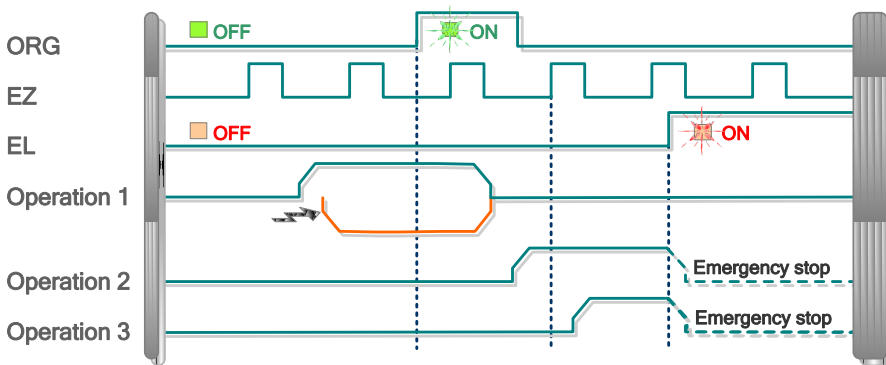
Low Speed[STRCS] operation**High Speed[STRN] operation****► [UCFG1][28~25] : “0101”**

구동 시작 후 원점 신호(IO5/ORG)의 설정 레벨로의 상향 에지에서 정지후 반대방향으로 [UCFG1][19~15] 에서 설정된 만큼(+ 1) EZ 신호의 설정 레벨로의 상향 에지를 만날 때 까지 구동한다.

Low Speed[STRCS] operation



High Speed[STRN] operation



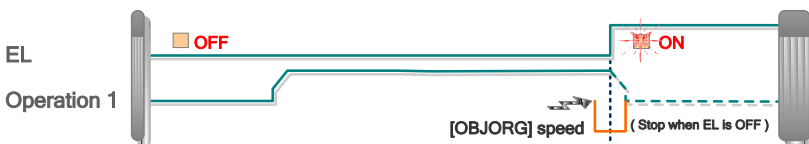
► [UCFG1][28~25] : “0110”

구동 시작 후 원점 ELM 설정 레벨로의 상향 에지에서 정지후 반대방향으로 [ORGOBJ] 속도로 ELM 신호의 하향 에지가 발생될 때 까지 구동한다. 구동 중 ELM신호에 의한 구동 정지 기능은 무시된다.

Low Speed[STRCS] operation



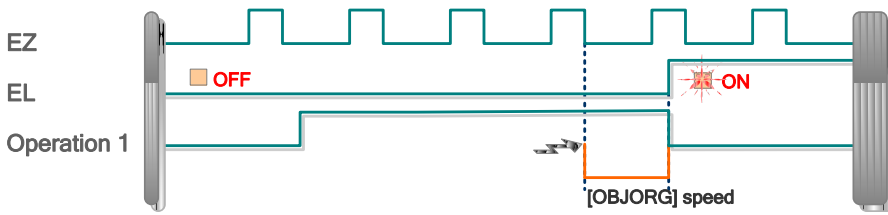
High Speed[STRCN] operation



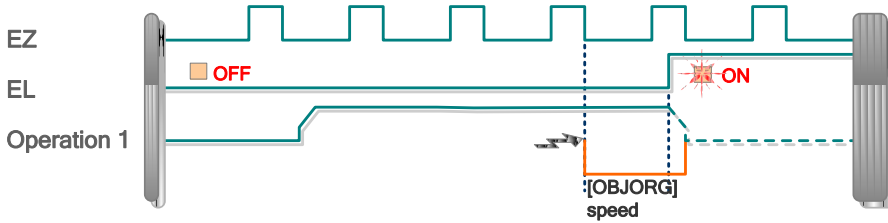
► [UCFG1][28~25] : “0111”

구동 시작 후 원점 ELM 설정 레벨로의 상향 에지에서 정지후 반대방향으로 [ORGOBJ] 속도로 [UCFG1][19~15]에서 설정된 만큼(+ 1) EZ 신호의 설정 레벨로의 상향 에지를 만날 때 까지 구동한다. 구동 중 ELM신호에 의한 구동 정지 기능은 무시된다.

Low Speed[STRCS] operation



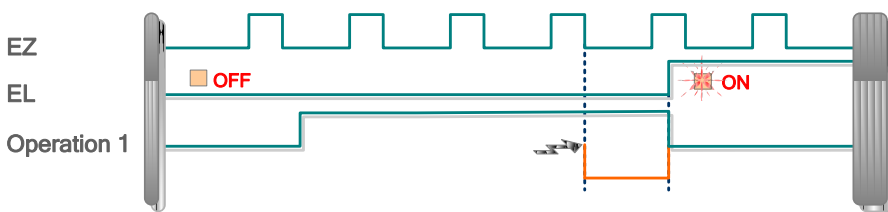
High Speed[STRN] operation



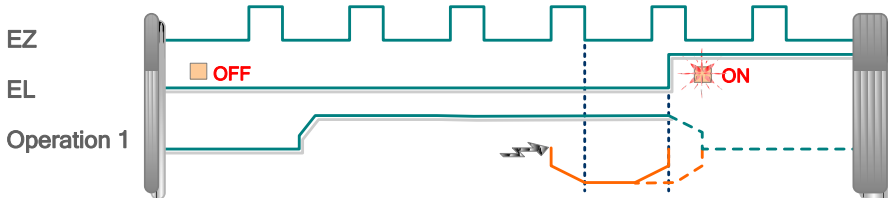
▶ [UCFG1][28~25] : “1000”

구동 시작 후 원점 ELM 설정 레벨로의 상향 에지에서 정지후 반대방향으로 [UCFG1][19~15] 에서 설정된 만큼 (+1) EZ 신호의 설정 레벨로의 상향 에지를 만날 때 까지 구동한다. 구동 중 ELM신호에 의한 구동 정지 기능은 무시된다.

Low Speed[STRCS] operation



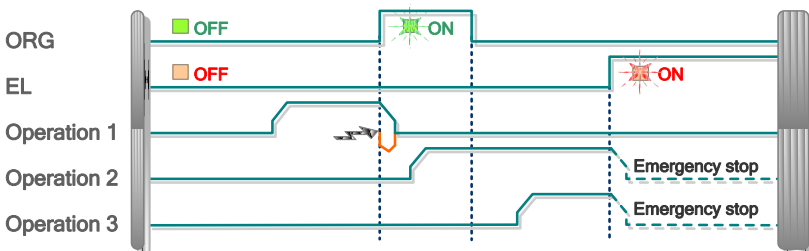
High Speed[STRN] operation



▶ [UCFG1][28~25] : “1001”

[UCFG1][28~25] : “0000”와 같이 원점 리턴 구동 후 반대 방향으로 [CNT2] 값이 0이 되도록([DCFG][6~0] = 0x15) 구동한다.

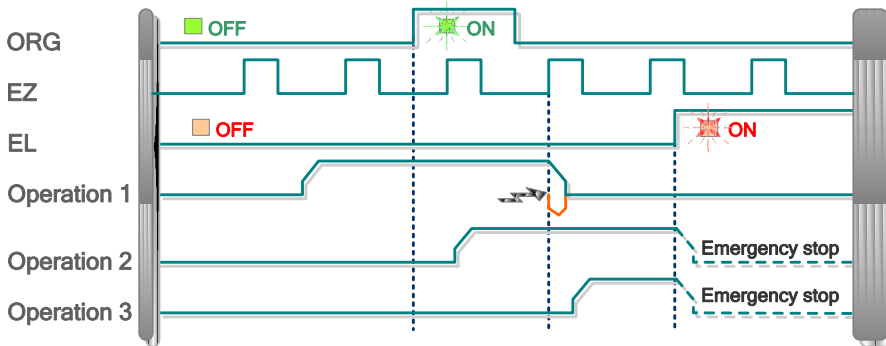
High Speed[STRN] operation



▶ [UCFG1][28~25] : “1010”

[UCFG1][28~25] : “0011”와 같이 원점 리턴 구동 후 반대 방향으로 [CNT2] 값이 0이 되도록([DCFG][6~0] = 0x15) 구동한다.

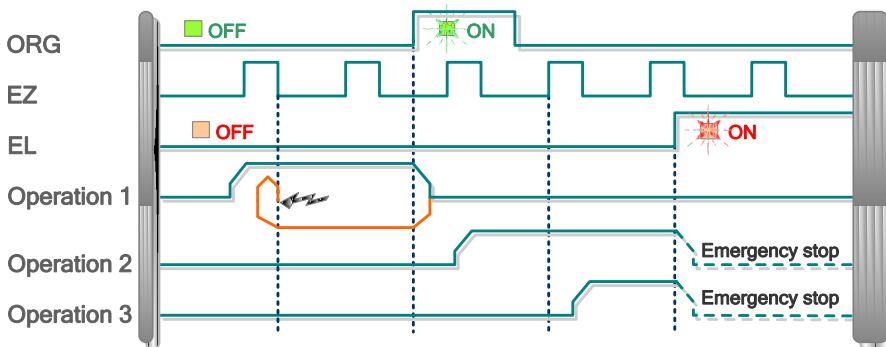
High Speed[STRN] operation



▶ [UCFG1][28~25] : “1011”

[UCFG1][28~25] : “0101”와 같이 원점 리턴 구동 후 반대 방향으로 [CNT2] 값이 0이 되도록([DCFG][6~0] = 0x15) 구동한다.

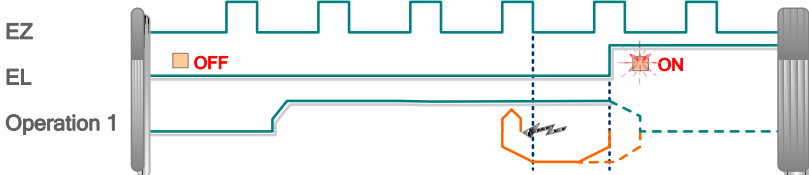
High Speed[STRN] operation



▶ [UCFG1][28~25] : “1100”

[UCFG1][28~25] : “1000”와 같이 원점 리턴 구동 후 반대 방향으로 [CNT2] 값이 0이 되도록([DCFG][6~0] = 0x15) 구동한다.

High Speed[STRN] operation



6.2.2.원점 이탈([DCFG][6~0] = 0x22~0x23)

현재 원점 상태가 설정 레벨일 경우 구동하여 원점 센서 입력 신호가 설정 레벨이 되지 않을 때 까지 구동한다. 만일 현재 원점 센서 입력 상태가 설정 레벨이 아닐 경우 구동 하지 않는다.

6.2.3.원점 검색([DCFG][6~0] = 0x24~0x25)

원점 복귀 구동을 연속으로 구동하여 현재 원점 센서의 위치에 상관없이 한번의 구동으로 원점을 검색할 수 있도록

한다. 원점 검색에서 사용하는 원점 복귀 모드([UCFG1][28~25])는 “0000”~”0101”까지 유효하다. 다음의 3가지 경우에 대하여 각각 다른 방법으로 원점 검색을 완료 한다.

▶ 현재 원점 센서 입력이 설정된 레벨로 입력되고 있을 때

반대 방향으로 [POS] 거리 만큼 원점 이탈 구동을 한 후 [UCFG1][28~25]에 설정된 원점 복귀 모드를 사용하여 원점 복귀 구동을 한다. 원점 센서에서 벗어 날 때까지 [POS] 거리 구동을 반복한다.

▶ 현재 원점 센서 입력이 설정된 레벨이 아니고 구동 중 ELM를 만나기 전 원점을 만났을 경우

[UCFG1][28~25]에 설정된 원점 복귀 모드를 사용하여 원점 복귀 구동을 한다.

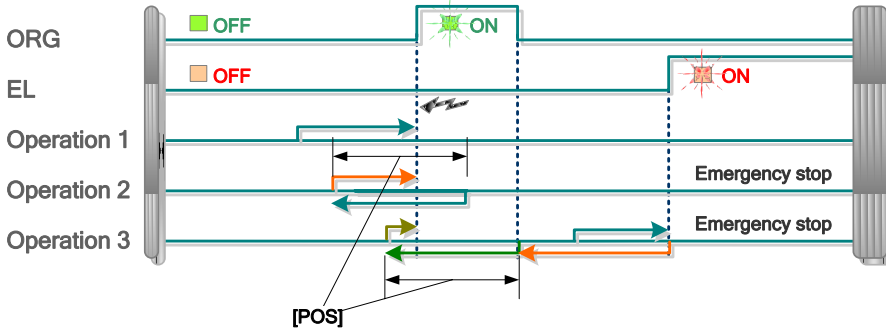
▶ 현재 원점 센서 입력이 설정된 레벨이 아니고 구동 중 원점 입력 보다 ELM 신호를 만났을 때

ELM 신호에 의한 구동 종료 후 반대 방향으로 원점 센서의 상향 에지를 검색한다. 검색 완료 후 [POS] 설정 값 만큼 원점 센서에서 벗어날 때 까지 추가 구동후 처음 구동 방향으로 [UCFG1][28~25]에 설정된 원점 복귀 모드를 사용하여 원점 검색구동 완료 한다.

▶ [UCFG1][28~25] = “0000”일 때 원점 검색 구동시 각 위치에서의 구동은 다음과 같다.

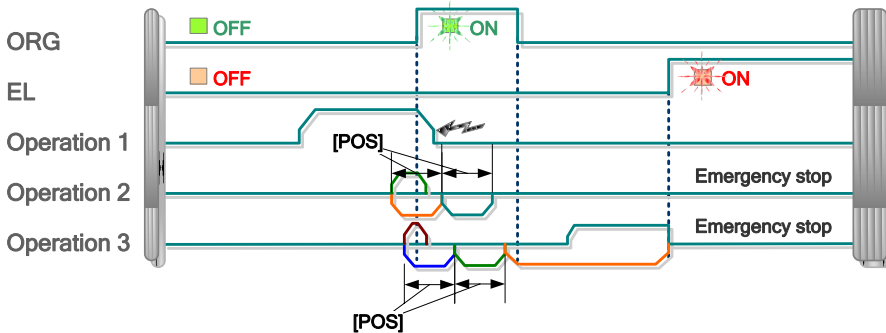
Low Speed[STRCS] operation

< Sensor : ELM ([UCFG1][0~1] = 1), ORG([UCFG1][7] = 1, [UCFG1][28~25] = “0000”>



High Speed[STRN] operation

< Sensor : ELM([UCFG1][0~1] = 1), ORG([UCFG1][7] = 1)>

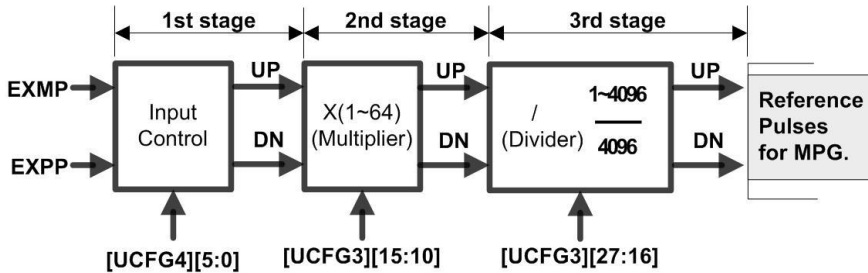


6.3. MPG(Manual Pulse Generation) input function

외부 펄스 입력을 기준으로 펄스를 출력하는 기능을 MPG(사용자 입력 펄스에 의한 구동)이라고 하는데, 로터리 인코더 또는 디지털 접점을 이용하여 [EXPP]/[EXMP] 단자에 [ECUP]/[ECDN] 같은 형태의 펄스를 입력하면 설정된 배율에 따라 펄스를 출력하거나, 보간 구동의 한 스텝 구동 펄스로 사용된다.

입력되는 펄스 형태는 외부 인코더 입력과 같은 형태로 사용되는데, 표5-4에 상세히 설명되어 있다. 입력형태 설정 및 필터는 [UCFG4][5~0]값으로 설정한다. 입력형태설정에 따라 해석된 결과에 배율을 곱하거나 분주하여 사용하는데 그 설정은 다음의 레지스터에서 설정한다.

- 입력 펄스 배율 설정 : [UCFG3][15~10], 설정 범위는 0~63 이며 설정값에 + 1하여 적용된다.
- 입력 펄스 분주비 설정 : [UCFG3][27~16], 설정 범위는 0~4095이며 설정값에 + 1하여 사용된다.



외부 펄스 입력을 기준으로 2nd Stage와 3rd Stage에 각각 임시 펄스 저장 카운터가 증가 되고 출력되는 펄스에 따라 감소된다. 만약 출력 펄스보다 입력 펄스에 의한 카운터의 증가가 빨라 내부 임시 카운터가 Overflow될 경우 외부 펄스 입력 카운터 값과 출력 펄스 카운터 값의 차이가 발생 할 수 있다. 따라서 출력 펄스 주파수가 외부 펄스 입력에 의한 카운터의 증가 주파수보다 충분히 큰 값이 되도록 속도 프로파일 데이터([POBJ], [PRANGE])를 설정하고 구동 하여야 한다.

7. Electrical Characteristics

7.1. Absolute Maximum Rating(VSS = 0V)

항 목	Symbol	Rating	Unit
Supply voltage	IVDD	-0.5 ~ 2.5	V
	VDD	-0.5 ~ 6	V
Input voltage	Vin	VDD = 3.3, -0.3 ~ VDD + 0.3	V
		VDD = 5, -0.5 ~ VDD + 0.5	V
Input current	Iin	10	mA
Output current	Iout	10	mA
Storage temperature	Tstg	-65 ~ +150	℃

7.2. Recommended Operation condition(VSS = 0V)

항 목	Symbol	Rating	Unit
Supply voltage	IVDD	1.62 ~ 1.98	V
	VDD	2.97 ~ 3.63 or 4.5 ~ 5.5	V
Ambient temperature	Ta	-45 ~ +85	℃

7.3. DC Characteristics

항 목		Condition	Min	Normal	Max	Unit
Input High voltage	V _{IH}	VDD = 5V	4.0		VDD + 0.3	V
		VDD = 3.3V	2.4		VDD + 0.3	V
Input Low voltage	V _{IL}	VDD = 5V	-0.3		0.8	V
		VDD = 3.3V	-0.3		0.6	V
Threshold point	V _T	VDD = 5V	2.07	2.18	2.41	V
		VDD = 3.3V	1.33	1.35	1.46	V
Schmitt trig Low to High	V _{T+}	VDD = 5V	2.6	2.77	3.04	V
		VDD = 3.3V	1.67	1.75	1.88	V
Schmitt trig High to Low	V _{T-}	VDD = 5V	1.53	1.62	1.79	V
		VDD = 3.3V	0.92	0.96	1.07	V
Junction Temperature	T _j		0	25	125	℃
Input Leakage Current	I _L				± 10	uA
Tri-State Input Leakage	I _{OZ}				± 10	uA
Pull-up Resistor	R _{PU}	VDD = 5V	32K	39K	51K	Ω
		VDD = 3.3V	48K	66K	90K	Ω
Pull-down Resistor	R _{DN}	VDD = 5V	63K	108K	244K	Ω
		VDD = 3.3V	42K	72K	172K	Ω
Output low voltage	V _{OL}				0.8	V
Output high voltage	V _{OH}		VDD - 0.8			V
Current consumption(IVDD)	iIdd	f _{CLK} = 40MHz			100	mA

7.4. AC Characteristics

7.4.1. CPU Interface (VDD = 5V ± 0.25V, IVDD = 1.8V, Ta = 0 to +70℃)

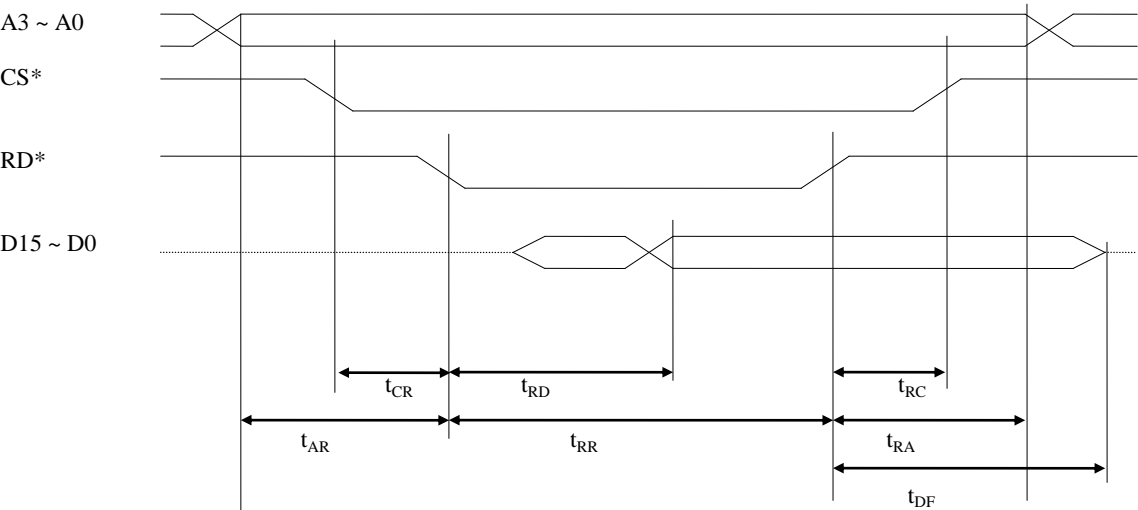
항 목		Condition	Min	Max	Unit
Clock frequency	f _{CLK}			40	MHz
Clock period	t _{CLK}		25		nS
Clock Low duration	t _{pWL}		10		nS
Clock High duration	t _{pWH}		10		nS
Chip selection stable time	t _{CR}		0		nS
Chip selection retention time	t _{RC}		0		nS

Read address stable time	t_{AR}		15		nS
Read address retention time	t_{RA}		0		nS
Read pulse width	t_{RR}		15		nS
Data delay time	t_{RD}	CL=50pF		10	nS
Data float delay time	t_{DF}	CL=50pF		9	nS
Write address stable time	t_{AW}		0		nS
Write address retention time	t_{WA}		0		nS
Write pulse width	t_{WW}		15		nS
Data retention time	t_{WD}		0		nS
Reset pulse width	t_{RST}		10 t_{CLK}		nS
Reset operation time	t_{RSTM}			5 t_{CLK}	nS

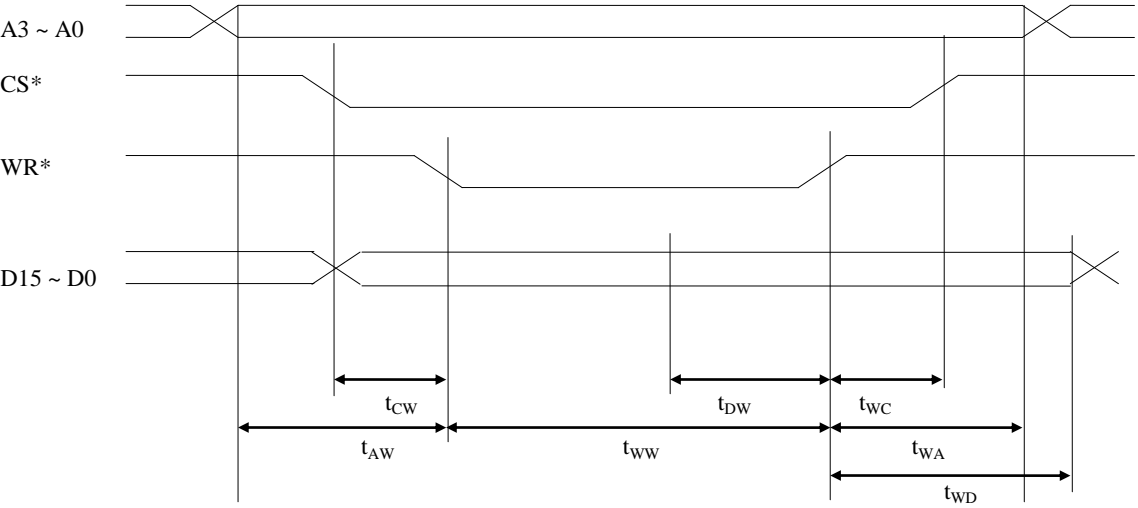
7.4.2.CPU Interface (VDD = 3.3V \pm 0.15V, IVDD = 1.8V, Ta = 0 to +70℃)

항 목	Condition	Min	Max	Unit
Clock frequency	f_{CLK}		40	MHz
Clock period	t_{CLK}	25		nS
Clock Low duration	t_{pWL}	10		nS
Clock High duration	t_{pWH}	12		nS
Chip selection stable time	t_{CR}	0		nS
Chip selection retention time	t_{RC}	0		nS
Read address stable time	t_{AR}	0		nS
Read address retention time	t_{RA}	0		nS
Read pulse width	t_{RR}	16		nS
Data delay time	t_{RD}	CL=50pF	12	nS
Data float delay time	t_{DF}	CL=50pF	11	nS
Chip selection stable time	t_{CW}	0		nS
Chip selection retention time	T_{WC}	0		nS
Write address stable time	t_{AW}	0		nS
Write address retention time	t_{WA}	0		nS
Write pulse width	t_{WW}	16		nS
Data retention time	t_{WD}	0		nS
Reset pulse width	t_{RST}	10 t_{CLK}		nS
Reset operation time	t_{RSTM}		5 t_{CLK}	nS

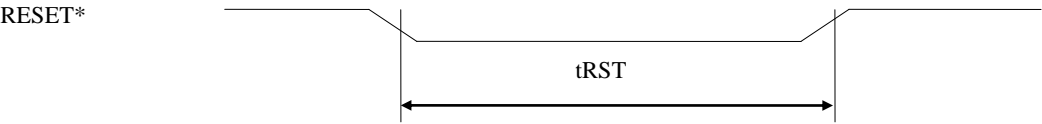
► Read timing



► Write timing



► RESET* timing



7.4.3. IO Interface

▶ 입력신호 timing

설정된 펄스폭 보다 작은 신호는 무시된다.

항 목		설 정	필터 신호 폭	Unit
PELM, NELM	t_{STPW}	[UCFG2][12~9]*	$2 * t_{CLK} \sim 1048576 * t_{CLK}$	nS
INP, ALARM	t_{SERV}	[UCFG2][16~13]*	$2 * t_{CLK} \sim 1048576 * t_{CLK}$	nS
UIO0~4, UIO7~9	t_{UIOG1}	[UCFG2][20~17]*	$2 * t_{CLK} \sim 1048576 * t_{CLK}$	nS
UIO5~6	t_{UIOG2}	[UCFG2][24~21]*	$2 * t_{CLK} \sim 1048576 * t_{CLK}$	nS
UIO10~11	t_{UIOG3}	[UCFG2][28~25]*	$2 * t_{CLK} \sim 1048576 * t_{CLK}$	nS
ESTOP, SQSTP1/2, SQSTR 1/2	t_{SIO}	[UCFG3][31~28]*	$2 * t_{CLK} \sim 1048576 * t_{CLK}$	nS
EXPP, EXMP	t_{PULSAR}	[UCFG4][4~4]**	$1 * t_{CLK} \sim 8 * t_{CLK}$	nS
ECUP, ECDN	t_{ENC}	[UCFG4][11~10]**	$1 * t_{CLK} \sim 8 * t_{CLK}$	nS

디지털 필터 설정*	필터 신호 폭
“0000”	8 * t_{CLK}
“0001”	128 * t_{CLK}
“0010”	512 * t_{CLK}
“0011”	2048 * t_{CLK}
“0100”	8192 * t_{CLK}
“0101”	32768 * t_{CLK}
“0110”	131072 * t_{CLK}
“0111”	524288 * t_{CLK}
“1000”	64 * t_{CLK}
“1001”	256 * t_{CLK}
“1010”	1024 * t_{CLK}
“1011”	4092 * t_{CLK}
“1100”	16384 * t_{CLK}
“1101”	65536 * t_{CLK}
“1110”	262144 * t_{CLK}
“1111”	1048576 * t_{CLK}

디지털 필터 설정**	필터 신호 폭
“00”	1 * t_{CLK}
“01”	2 * t_{CLK}
“10”	4 * t_{CLK}
“11”	8 * t_{CLK}

▶ 출력신호 timing

항 목			Unit
구동 명령 → BUSY(HIGH)	t_{STARTC}	$6 * t_{CLK}$	nS
SQSTR1/2 → BUSY(HIGH)	t_{STARTS}	$12 * t_{CLK}$, 디지털 필터(“0000”)	nS
트리거 발생 명령 → TRIG	t_{TRIG1}	$7 * t_{CLK}$	nS
ECDN/ECUP → TRIG	t_{TRIG2}	$15 * t_{CLK}$, 디지털 필터(“11”)	nS
구동 완료 → INT	t_{INT1}	$3 * t_{CLK}$	nS
INP → BUSY(LOW)	t_{INP1}	$12 * t_{CLK}$, 디지털 필터(“0000”)	nS
EXPP/EXMP → PULSE/DIR	t_{MPG}	$18 * t_{CLK}$, 디지털 필터(“11”)	nS

8. Package diagram

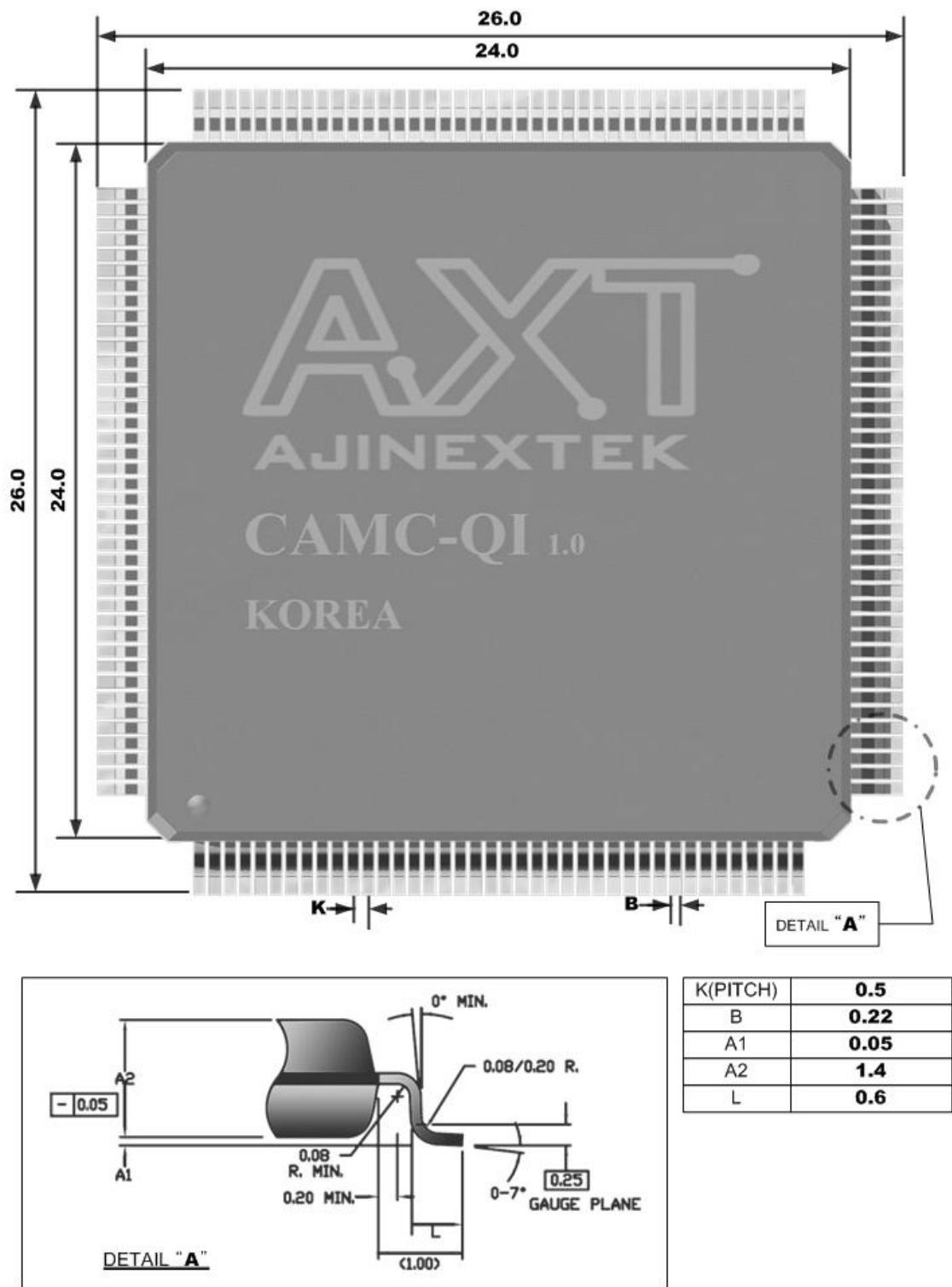


그림 8-1. CAMC-QI package drawing(Unit : mm)

9. Appendix

9.1. Precaution

9.1.1. 구동 중 [SSTOP], [STOP], [ESTOP] 명령어 실행 시 주의 사항

지정거리 구동 및 보간 구동 중(펄스 출력 중) 사용자에게 의한 [SSTOP], [STOP], [ESTOP]이 발생할 경우 출력중인 펄스가 비정상적으로 잘리는 경우가 발생하여 내부 위치 정보와 외부 위치 정보가 차이가 날 수 있다. 이러한 현상을 방지 하기 위하여 다음과 같은 방법을 사용하여야 한다.

1) 지정거리 구동 중 [SSTOP] 명령어 수행.

지정 거리 구동 중 감속 위치가 결정되어 감속 중 일 때 [SSTOP] 명령어를 수행하게 되면 감속후 마지막 펄스가 잘리기 때문에 내부 위치 정보와 외부 위치 정보가 차이 날수 있다. 이에 따른 문제를 해결하기 위해서는 지정 거리 구동 중 [SSTOP] 명령 수행은 현재 상태를 확인하여 감속 중이 아닐 때만 수행 하도록 한다.

2) 보간 구동 중 [STOP]/[ESTOP] 명령어 수행.

보간 구동 중 일 때 [STOP]/[ESTOP] 명령어를 수행하게 되면 출력 펄스가 명령수행 순간 잘리기 때문에 내부 위치 정보와 외부 위치 정보가 차이가 날 수 있다. 이에 따른 문제를 피하기 위해서는 보간 구동 중 [STOP]/[ESTOP] 명령어 수행 후 시스템의 원점 검색을 다시 하거나 외부 위치를 기준으로 내부 위치 값을 갱신 하여 주어야 한다.*

*) 외부 위치를 알 수 없는 시스템에서 사용할 경우 내부의 스크립트 기능 및 타이머 기능, 트리거 기능을 사용하여 피할 수 있습니다. 상세내용은 문의하시기 바랍니다.

9.1.2. 트리거 기능 사용 중 주의 사항

1) 트리거 범위 설정 값의 적용에 대한 주의 사항

트리거 출력 기능을 사용할 때 설정하는 트리거 사용 범위 값이 적용되는 방법은 다음과 같다.

▶ 트리거 기준 카운터가 감소하여 트리거 사용 범위 하한 값에서 벗어 날 때

실제 트리거 사용 범위는 $[TRGSP] - 2$ 로 확대 적용 된다.

▶ 트리거 기준 카운터가 증가하여 트리거 사용 범위 상한 값에서 벗어 날 때

실제 트리거 사용 범위는 $[TRGEP] + 2$ 로 확대 적용 된다.

특히 위치 주기에 대한 트리거 발생 기능 사용시 위치 주기를 2 이하로 설정한다면 구동 방향에 따라서 실제 트리거 발생 범위에 대하여 다음과 같이 $[TRGSP]$, $[TRGEP]$ 를 설정하여야 한다.

▶ 트리거 기준 카운터가 감소하여 트리거 사용 범위 하한 값에서 벗어 날 때

$[TRGSP] \leftarrow$ 실제 트리거 범위 하한 값 + 2

▶ 트리거 기준 카운터가 증가하여 트리거 사용 범위 상한 값에서 벗어 날 때

$[TRGEP] \leftarrow$ 실제 트리거 범위 상한 값 - 2

2) 트리거 위치/주기 Queue의 사용에 대한 주의 사항

트리거 기능 사용 중 설정하는 트리거 위치/주기 설정은 Queue에 적체되기 때문에 기존에 설정된 정보를 사용하지 않을 경우 $[CLRTRG]$ 로 초기화 한 다음 설정하여야 한다.

9.1.3. 서보 위치 결정 완료 기능(인포지션 기능) 사용 중 주의 사항

단위 구동 완료 후 $[INP]$ 단자로 입력되는 서보위치 완료 신호의 기능을 사용하기 위해서는 $[DCFG][24]$ 를 '1'로 설정하여 구동 시작 명령을 실행하고, 이어서 $[DCFG][6\sim0] = "111111"$ 로 설정하여 구동 시작을 해주어야 단위 구동 완료 후 인포지션 기능이 활성화 된다.

9.1.4. 삼각 구동 방지 기능 사용 중 주의 사항

S자 속도 프로파일을 사용하여 지정거리 구동 중 짧은 거리 구동 시 목표 속도까지 도달하지 않고 구동이 종료되는 경우와 가속 중 감속정지를 해야 하는 경우 순간적인 속도 변화를 방지하기 위하여 삼각구동 방지 기능이 제공된다. 이때 다음과 같이 주의하여 사용하여야 한다.

▶ 삼각 구동 방지 기능은 가속 중 감속정지 및 짧은 거리 구동에 대하여 한가지만 선택하여 사용하여야 한다.

지정 거리 구동 중 가속 중 감속정지 수행에 대한 삼각 구동 방지($[(P)DCFG][25]$)와 짧은 거리에 대한 삼각구

동 방지[(P)DCFG][23]을 모두다 활성화 시켜둘 경우 감속정지 명령어에 의한 삼각구동 발생시 급정지할 수 있으므로 두 기능 중 한가지만 활성화 하여 사용하여야 한다. 지정 거리 구동 중 두 가지 경우에 대하여 모두가 삼각구동 방지를 하기 위해서는 짧은 거리에 대한 삼각구동 방지 기능을 비활성화 하고 삼각구동이 발생하지 않도록 구동 속도를 아래의 식에 따라 계산된 값과 구동에 사용하려고 하였던 값 중 작은 값을 [POBJ]로 입력한다.

$$[POBJ] = \sqrt{\frac{[PRANGE] \times 32768 \times |[PPOS]|}{([PRATE1] + [PRATE2]) \times 2}} + [PSTD]^2$$

9.2. 레지스트 및 명령어 일람

CODE (Read/Write)	CONTENTS(Previous registers/etc)	W	DEFAULT
0x00/0x80	Previous Speed magnitude data [PRANGE]	16	0xEA5F
0x01/0x81	Previous Start/Stop speed data [PSTD]	16	0x1
0x02/0x82	Previous Object speed data [POBJ]	16	0x1
0x03/0x83	Previous Acceleration rate data [PRATE1]	16	0xFFFF
0x04/0x84	Previous Deceleration rate data [PRATE2]	16	0xFFFF
0x05/0x85	Previous S drive region during acceleration [PSW1]	15	0x7FFF
0x06/0x86	Previous S drive region during deceleration [PSW2]	15	0x7FFF
0x07/0x87	Previous Drive configure [PDCFG]	32	0x0
0x08/0x88	Previous Slow down/rear pulse amount data [PREAR]	28	0x0
0x09/0x89	Previous Drive pulse amount data, Interpolation end position [PPOS]	28	0x0
0x0A/0x8A	Previous Circular Interpolation center, Master axis target position for multiple chip linear interpolation [PCENT]	28	0x0
0x0B/0x8B	Previous Interpolation step number [PISNUM]	28	0x0
/0x8C	Clear previous registers [CLRPRE]		
/0x8D	Shift previous register data [POPPRE]		
0x0E/0x8E	Restore data ports/Backup data ports [PORTMA]	32	0x0
0x0F	Current speed [CURSPD]	16	0x0

CODE (Read/Write)	CONTENTS(Working registers)	W	DEFAULT
0x10/0x90	Working Speed magnitude data [RANGE]	16	0x0
0x11/0x91	Working Start/Stop speed data [STD]	16	0x0
0x12/0x92	Working Object speed data [OBJ]	16	0x0
0x13/0x93	Working Acceleration rate data [RATE1]	16	0x0
0x14/0x94	Working Deceleration rate data [RATE2]	16	0x0
0x15/0x95	Working S drive region during acceleration [SW1]	15	0x0
0x16/0x96	Working S drive region during deceleration [SW2]	15	0x0
0x17/0x97	Working Drive configure [DCFG]	32	0x0
0x18/0x98	Working Slow down/rear pulse amount data [REAR]	28	0x0
0x19/0x99	Working Drive pulse amount data, Interpolation end position [POS]	28	0x0
0x1A/0x9A	Working Circular Interpolation center, Master axis target position for multiple chip linear interpolation [CENT]	28	0x0
0x1B/0x9B	Working Interpolation step number [ISNUM]	28	0x0
0x1C/	Remain pulse data after stopping preset drive function abnormally [REMAIN]	28	0x0
0x1F/0x9F	Original search object speed [OBJORG]	16	0x1

CODE(HEX)	CONTENTS (Universal in/out setting)	W	DEFAULT
0x1D/0x9D	Universal in/out terminal mode[UIOM] ('0' : input, '1' : output)	12	0x01F
0x1E/0x9E	Universal in/out value[UIO]	15	0x0000

CODE(HEX)	CONTENTS (Drive start command) -
/0xA0	Normal profile mode drive start(STD→OBJ→STD).[STRN]
/0xA1	Start at OBJ profile mode drive start.(OBJ→STD).[STRO]
/0xA2	Constant speed profile #1 drive start.(OBJ).[STRCO]
/0xA3	Constant speed profile #2 drive start.(STD).[STRCS]
/0xDC	Normal profile mode drive start(STD→OBJ→STD).[ASTRN] (with DCFG 7~0 bit data in data port 0)
/0xDD	Start at OBJ profile mode drive start.(OBJ→STD).[ASTRO] (with DCFG 7~0 bit data in data port 0)
/0xDE	Constant speed profile #1 drive start.(OBJ).[ASTRCO] (with DCFG 7~0 bit data in data port 0)
/0xDF	Constant speed profile #2 drive start.(STD).[ASTRCS] (with DCFG 7~0 bit data in data port 0)

CODE(HEX)	CONTENTS (Drive control command)
/0xA4	Slow Down stop.[SSTOP]
/0xA5	Immediately stop.[STOP]
/0xA6	Output one shot of the start pulse from SQSTR1 terminal.[SQRO1]
/0xA7	Output one shot of the start pulse from SQSTR2 terminal.[SQRO2]

/0xA8	Execution sync start function same as SQSTR1 input.[SQRI1]
/0xA9	Execution sync start function same as SQSTR2 input.[SQRI2]
/0xAA	Output one shot of the stop pulse form SQSTP1 terminal.[SQPO1]
/0xAB	Output one shot of the stop pulse form SQSTP2 terminal.[SQPO2]
0x2C/	Interpolation step counter.[ISCNT]
0x2D/	Interpolation step counter for advanced deceleration mode. [ISACNT]
/0xAE	Emergency stop all axis.[ESTOP]
/0xAF	Software reset.[SWRESET]
/0xB0	Driven pulse amount during last driving(Interpolation step counter for path move).[DRPCNT]
/0xB1	Interrupt generation command.[INTGEN]

CODE(HEX) Read/Write	CONTENTS (Peripheral function setting)	W	DEFULT
/0xB2	Trigger queue POP[TRGQPOP]		
0x33/0xB3	Trigger/Timer configure[TRTMCF]	10	0x0F0
0x34/0xB4	Software negative limit position.[SNLMT]	28	0x8000000
0x35/0xB5	Software positive limit position. [SPLMT]	28	0x7FFFFFFF
0x36/0xB6	Trigger pulse width. [TRGPW]	32	0xFFFFFFFF00
0x37/0xB7	Trigger function start position.[TRGSP]	28	0x8000000
0x38/0xB8	Trigger function end position.[TRGEP]	28	0x7FFFFFFF
0x39/0xB9	Push trigger position or period data to queue.[PTRGPOS]	28	0x0
/0xBA	Clear trigger position or period queue.[CLRTRG]		
/0xBB	Generate one shot trigger pulse.[TRGGEN]		
0x3C/0xBC	Timer 1 period[TMRP1]	28	0x0
0x3D/0xBD	Timer 2 period[TMRP2]	28	0x0
0x3E/0xBE	Timer1 stop/start!.[TMR1GEN]		
0x3F/0xBF	Timer2 stop/start!.[TMR2GEN]		
0x60/0xE0	ERC reset/set[ERCRS]		

CODE(HEX) (Read/Write)	CONTENTS (Script1/2/3 setting registers)	W	DEFAULT
0x40/0xC0	Script1 control queue register[SCRCON1]	25	0x0
0x41/0xC1	Script1 command queue register[SCRCMD1](0xCF for NOOP)	32	0x0
0x42/0xC2	Script1 execution data queue register[SCRDAT1]	32	0x0
0x43/	Script1 captured data queue register(top of depth 15 queue)[CQ1]	32	0x0
0x44/0xC4	Script1 flag control register.[SCRCFG1]	16	0xF0F0
0x45/0xC5	Script2 control queue register[SCRCON2]	32	0x0
0x46/0xC6	Script2 command queue register[SCRCMD2]	32	0x0
0x47/0xC7	Script2 execution data queue register[SCRDAT2]	32	0x0
0x48/	Script2 captured data queue register(top of depth 16 queue)[CQ2]	32	0x0
0x49/0xC9	Script2 flag control register.[SCRCFG2]	16	0xF0F0
0x4A/0xCA	Script3 control register[SCRCON3]	32	0x0
0x4B/0xCB	Script3 command register[SCRCMD3]	32	0x0
0x4C/0xCC	Script3 execution data register[SCRDAT3]	32	0x0
0x4D/	Script3 captured data register(top of depth 16 queue)[CQ3]	32	0x0
0x4E/0xCE	Don't care.		
/0xCF	[No operation code for Script reservation command].		

CODE(HEX) (Read/Write)	CONTENTS (Script4 setting/Script status registers)	W	DEFULT
0x50/0xD0	Script4 control register[SCRCON4]	32	0x0
0x51/0xD1	Script4 command register[SCRCMD4]	32	0x0
0x52/0xD2	Script4 execution data register[SCRDAT4]	32	0x0
0x53/	Script4 captured data register [CQ4]	32	0x0
0x54/0xD4	Target source data setting.[SCRTG]	16	0x0
0x55/	Script status #1[SCRSTAT1]	31	0x00005555
0x56/	Script status #2[SCRSTAT2]	15	0x0000
/0xD7	Initialize script queues with target selection.[INITSQ]	4	
/0xD8	Initialize captured data queue with target selection.[INITCQ]	4	
0x59/0xD9	Set enable mode with target selection.[SCRM]	4[9]	0x0F
/0xDA	Pop script #1 queue.[SQ1POP]		

CODE(HEX) (Read/Write)	CONTENTS (Script4 setting/Script status registers)	W	DEFAULT
/0xDB	Pop script #2 queue.[SQ2POP]		

CODE(HEX) (Read/Write)	CONTENTS (Counter function registers)	W	DEFAULT
0x61/0xE1	Counter lower bound data[CNTLB]	28	0x8000000
0x62/0xE2	Counter upper bound data[CNTUB]	28	0x7FFFFFFF
0x63/0xE3	Counter configure #1 [CNTCF1]	32	0x0
0x64/0xE4	Counter configure #2 [CNTCF2]	32	0x0
0x66/0xE6	Counter #1 data[CNT1]	28	0x0
0x67/0xE7	Counter #2 data[CNT2]	28	0x0
0x68/0xE8	Counter #3 data[CNT3]	28	0x0
0x69/0xE9	Counter #4 data[CNT4]	28	0x0
0x6A/0xEA	Counter #5 data[CNT5]	28	0x0
0x6B/0xEB	Counter #1 comparator data[CNTC1]	28	0x0
0x6C/0xEC	Counter #2 comparator data[CNTC2]	28	0x0
0x6D/0xED	Counter #3 comparator data[CNTC3]	28	0x0
0x6E/0xEE	Counter #4 comparator data[CNTC4]	28	0x0
0x6F/0xEF	Counter #5 comparator data[CNTC5]	28	0x0

CODE(HEX) (Read/Write)	CONTENTS (Configure/Status registers)	W	DEFAULT
0x70/0xF0	Configure register #1[UCFG1]	29	0x0
0x71/0xF1	Configure register #2[UCFG2]	29	0x0
0x72/0xF2	Configure register #3[UCFG3]	32	0x0
0x73/0xF3	Configure register #4[UCFG4]	32	0x0
0x77/0xF7	Interrupt bank #1 mask[IMASK1]	32	0x0
0x78/0xF8	Interrupt bank #2 mask[IMASK2]	32	0x0
0x79/0xF9	Status register #1[STAT1]/End status clear[ESCLR]	32	0x0
0x7A	Status register #2[STAT2]	32	0x0
0x7B	Status register #3[STAT3]	32	0x0
0x7C	Status register #4[STAT4]	32	0x0
0x7D	Status register #5[STAT5]	32	0x0
0x7E/0xFE	Interrupt bank#1 flag/interrupt flag#1 clear[IFLAG1]	32	0x0
0x7F/0xFF	Interrupt bank#2 flag/interrupt flag#2 clear[IFLAG2]	32	0x0

9.3. 찾아 보기(하단 페이지 번호)

A

ASTRCO	30
ASTRCS	30
ASTRN	30
ASTRO	30

C

CENT	29
CLRPRE	29
CLRTRG	33
CNT1	46
CNT2	46
CNT3	46
CNT4	46
CNT5	46
CNTC1	47
CNTC2	47
CNTC3	47
CNTC4	47
CNTC5	47
CNTCF1	44
CNTCF2	45
CNTLB	44
CNTUB	44
CQ1	39
CQ2	40
CQ3	41
CQ4	42
CURSPD	

D

DCFG	26
DRPCNT	31

E

ERCR	34
ESCLR	52
ESTOP	30, 70

I

IFLAG1	55
IFLAG2	55
IMASK1	51
IMASK2	51
INITCQ	43
INITSQ	43
INTGEN	31
ISACNT	31
ISCNT	31
ISNUM	29

O

OBJ	25
OBJORG	29

P

PCENT	29
PDCFG	26
PISNUM	29
POBJ	25
POPPRE	29
POS	28
PPORTMA	29
PPOS	28
PRANGE	25
PRATE1	25
PRATE2	25
PREAR	28
PSTD	25
PSW1	26
PSW2	26
PTRGPOS	33

R

RANGE	25
RATE1	25
RATE2	25
REAR	28
REMAIN	29

S

SCRCFG1	39
SCRCFG2	40
SCRCMD1	39
SCRCMD2	40
SCRCMD3	41
SCRCMD4	42
SCRCON1	39
SCRCON2	40
SCRCON3	41
SCRCON4	42
SCRDAT1	39
SCRDAT2	40
SCRDAT3	41
SCRDAT4	42
SCRM	43
SCRSTAT1	42
SCRSTAT2	42
SCRTG	42
SNLMT	22
SPLMT	22
SQ1POP	44
SQ2POP	44

SQRI1	31
SQRI2	31
SQRO1	31
SQRO2	31
SQSTP1	31
SQSTP2	31
SSTOP	30,70
STAT1	52
STAT2	53
STAT3	53
STAT4	54
STAT5	54
STD	25
STOP	30,70
STRCO	30
STRCS	30
STRN	30
STRO	30
SW1	26
SW2	26
SWRESET	31

T

TMR1GEN	34
TMR2GEN	34
TMRP1	34
TMRP2	34
TRGEP	33
TRGGEN	34
TRGPW	32
TRGQPOP	32
TRGSP	33
TRTMCF	32

U

UCFG1	47
UCFG2	47
UCFG3	48
UCFG4	49
UIO	29
UIOM	29

이 설명서의 내용은 예고 없이 변경될 수 있습니다. 용례에 사용된 회사, 기관, 제품, 인물 및 사건 등은 실제 데이터가 아닙니다. 어떠한 실제 회사, 기관, 제품, 인물 또는 사건과도 연관시킬 의도가 없으며 그렇게 유추해서도 안됩니다. 해당 저작권법을 준수하는 것은 사용자의 책임입니다. 저작권에서의 권리와는 별도로, 이 설명서의 어떠한 부분도 (주)아진엑스텍의 명시적인 서면 승인 없이는 어떠한 형식이나 수단(전기적, 기계적, 복사기에 의한 복사, 디스크 복사 또는 다른 방법) 또는 다른 목적으로도 복제되거나, 검색 시스템에 저장 또는 도입되거나, 전송될 수 없습니다.

(주)아진엑스텍은 이 설명서 본안에 관련된 특허권, 상표권, 저작권 또는 기타 지적 소유권 등을 보유할 수 있습니다. 서면 사용권 계약에 따라 (주)아진엑스텍으로부터 귀하에게 명시적으로 제공된 권리 이외에, 이 설명서의 제공은 귀하에게 이러한 특허권, 저작권 또는 기타 지적 소유권 등에 대한 어떠한 사용권도 허용하지 않습니다.