

기술 문서

External Time-Base

External Time-Base (Electronic Cams)

외부 시간 기반 동작 (전자 캠)

August 6, 2007

외부 시간 기반 제어 (External Time-Base Control)

External Time-Base Control (Electronic Cams)

외부 축에 대한 좀 더 정밀한 동기 방법에는 **time-base** 제어가 있으며, 이는 입력 신호의 주파수가 이송과 프로그램의 실행 속도를 결정하는 것입니다. **Time-base** 제어는 전체 좌표계에 대해서 동시에 동작하게 됩니다. 어떤 엔코더 레지스터를 통해서 입력 주파수를 받을 것인지 결정하고, 입력 주파수와 프로그램간의 실행 비율을 규정하게 됩니다. **Time-base** 는 입력 주파수에 비례해서 이송 속도를 변경할 뿐 아니라, 전체 위치 동기를 유지하게 합니다 (입력주파수가 0 일 때에도 마찬가지임.). 이 방법은 여러 줄 나사산 내기 등에 적용될 수 있습니다.

What is Time-Base Control?

Turbo PMAC 의 모션 언어에 대한 설명을 보면, 위치 궤적은 시간의 함수로 표현하고 있습니다. 이송 방법이 속도 또는 이동시간, 어느 것으로 설정되든 간에, 최종적으로 궤적은 위치 대(vs) 시간의 함수로 정의됩니다.

이는 대부분의 어플리케이션에 잘 적용될 수 있습니다. 그러나, 많은 어플리케이션에서, Turbo PMAC 의 축들을 Turbo PMAC 에서 제어하지 않는 외부 축들에 동기(종속)시켜서 사용하려 합니다 (또는 간혹, 다른 좌표계에 있는 별도의 독립적인 축에 동기 할 때도 있습니다.). 이러한 많은 어플리케이션에서, 우리는 Turbo PMAC 의 궤적을 시간이 아닌 주축(마스터 축)의 위치에 대한 함수로 정의하기를 원하고 있습니다.

Real-Time Input Frequency (실시간 입력 주파수)

Turbo PMAC 에서 이를 구현하는 방법에 대해서 위치를 시간에 대한 함수로 표현하고 있지만, 실제로는 시간을 주축(마스터 축)에 의한 이동거리에 비례하도록 만드는 것입니다. 이는 counts/msec 단위로 정의되어 있는 주축의 위치 센서의 입력을 실시간 입력 주파수(RTIF)로 정의하는 것입니다. 예를 들면, RTIF 를 32 cts/msec 로 정의 합니다. 그런 다음, **time-base** 모드에서, 프로그램상의 1 밀리세크(msec)는 물리적인 거리가 어떻게 되건 간에 실제로는 주축 엔코더의 32cts 를 의미하는 것입니다. 만약에 슬레이브(Slave) 모션 프로그램에서 2 초 동안 움직이도록 프로그램이 작성되었다면, 이는 주축 엔코더 기준으로 64000counts 가 입력되어야 완료가 되는 것입니다.

Constraints on Selection of RTIF (RTIF 를 선택할 때의 제약사항)

만약 Turbo PMAC 에서 time-base 계산을 수행할 때에, 분해능과 이동 범위에 제약이 없다면, 실시간 입력 주파수는 매우 임의적으로 아무 것이나 선택될 수 있을 것입니다 (어떤 주파수라도 RTIF 로 선택될 수 있으며, 모션 프로그램은 그 선택된 RTIF 에 대해서 작성 되어질 수 있습니다.). 그러나, Turbo PMAC 은 time-base 계산을 정수연산으로 수행하기 때문에, 그 분해능과 설정 범위 또한 24bit 로 제한됩니다.

RTIF 를 선택하는 데는 아래의 2 가지 제약 사항이 있습니다.

1. RTIF 로부터 계산되는 TBSF(Time-Base Scale Factor)는 반드시 정수가 되어야 합니다. Turbo PMAC 에서 계산을 위해 필요한 값은 cts/msec 단위의 주파수가 아니라, 이의 역수인 msec/ct 입니다. 이 값이 정수 범위 내에 있기 위해서는, $2^{17}(131072)$ 을 주파수로 나눈 값이 정수가 되도록 하는 것입니다.

만약 RTIF 로 100cts/msec 가 선택된다면, TBSF 는 $131072/100 = 131.027$ 로 계산되어 정수가 아님을 알 수 있습니다. Turbo PMAC 은 여기서 정수 부분인 131 만을 사용할 수 있으므로, 소수점 이하는 잘려버릴 것입니다.

RTIF 는 cts/msec 단위로 계산했을 때에 2 의 승수(예, 32, 64, 128 등)로 선택될 때에만 항상 정수 값의 TBSF 를 만들 수 있습니다. 또한, RTIF 를 정수로 나누었을 때에 값이 2 의 승수와 같으면 정상적으로 동작하게 됩니다. 예를 들면, $204.8 \text{ cts/msec} (=2048/10, 2^{11}/10)$ 일 때에는 TBSF 는 $2^{17} / (2^{11} * 10) = 640$ 이 됩니다.

2. Time-base 계산은 IF/RTIF 가 kHz 단위의 Servo Update Frequency 와 같을 때에 IF(Input Frequency, 입력주파수)로 포화상태에 이릅니다. 기본 서보 업데이트 주기인 2.25kHz 에서 RTIF 가 32cts/msec 이면, 최대 입력 주파수는 $32 * 2.25 = 72 \text{ cts/msec}$ 에서 포화상태에 이릅니다. 만약에 시스템이 100 cts/msec 까지 동작할 수 있는 경우에는, RTIF = 32 cts/msec 이면 수용될 수 없으며, RTIF = 64 cts/msec 로 선택되어야 합니다 ($100/64 = 1.5625 < 2.25$).

최대 입력 주파수보다 큰 값으로 RTIF 를 설정하는 것은 전혀 문제가 없습니다.

How Time-Base Control Works (Time-Base 제어는 어떻게 동작하는가)

Time-Base 제어는 마지막 서보 사이클로부터 경과한 시간에 대해서 매 서보 사이클마다 지령 위치를 계산하는 루틴에 의해서 동작하게 됩니다. (변수 I10 이 실제 경과 시간에 대한 정보를 포함하고 있습니다.) 서보 사이클간의 실제 경과 시간은 서보 루프에 따라서 변하거나 유동적이지만

않습니다. 이것은 외부 주파수에 따라 변하는 지령 궤적의 변화율일 뿐이며, 좌표계의 모든 궤적이 동시에 변하게 되므로, 공간상에서의 경로는 변하지 않습니다.

Instruction for Using an External Time-Base Signal

외부 time-base 를 사용하기 위해서는 여러 단계의 설정이 필요하게 됩니다. 그러나, 한 번 설정이 완료되면, time-base 제어는 사용자와 프로그램에 대해서 명백하며, 자동으로 동작하게 됩니다. 일반적인 설정에 대한 단계가 아래에 자세히 나와 있습니다.

Step1 : Signal Decoding (신호 해석)

이 신호는 Turbo PMAC 에 들어오는 증분 엔코더 입력(Channel A and B)의 하나입니다. 이 신호는 반드시 4 체배 신호(엔코더에서의 출력)이거나, 펄스와 방향(펄스는 A, 방향은 B) 신호여야 합니다. 사용되는 엔코더 신호에 대하여, I7mn0 가 디코딩 방법을 결정해서, "count"가 어떤 것인지를 정의하게 됩니다. 예를 들어, 4 체배 신호에 대해서는 I7mn0 = 3 또는 7 로 설정해서 엔코더 사이클 당 4 개의 카운트를 하도록 정의합니다. 반면에 I7mn0 = 2 또는 6 일 때에는 엔코더 사이클 당 2 개의 카운트만을 하도록 정의합니다. 3 과 7 일 때와 2 와 6 일 때의 차이점은, 디코더가 샘플 하는 방식의 차이로 나타납니다.

주의

마스터 시그널이 항상 증가하는 방향으로만 가도록 해야 합니다.
카운트가 감소하게 되면, 음수의 time-base 가 적용되어 Turbo PMAC 에서는 이를 처리할 수 없게 됩니다.

Step2 : Interpolation (보간)

신호로부터 들어온 값은 일단 해석되고 카운트가 되고 나면, 위치 피드백 신호가 그렇듯이, 매 서보 사이클마다 엔코더 컨버전 테이블에 전달되게 됩니다. 여기서는 1/T 변환 방법을 쓰는 것을 강력히 추천합니다. 왜냐하면, 이 방법이 매우 정교한 서브-카운트에 대한 보간 방법을 제시하기 때문에 (카운터와 관련된 타이머를 사용해서) time-base 정보를 매우 부드럽게 만들 수 있는 것입니다. 입력 신호로부터 카운트하는 과정이 반드시 이런 방법으로 될 수 있도록 변환 테이블을 설정하도록 해야 합니다. 엔코더 변환 테이블에 대한 공장 출하시의 기본설정은 현재 엔코더 카운트에 대한 1/T 변환으로 되어 있습니다. 보다 상세한 사항은 엔코더 변환 테이블에 대한 설명을 참조하시기 바랍니다.

Step 3 : Time Base Calculation (시간기반 계산)

엔코더 변환 테이블의 개별 엔트리들은 위의 2 단계에서 보간 위치 정보를 얻어 내어서, 이 값을 이전 서보 사이클의 위치 값에서 빼고, 다시 이 값에 스케일 값을 곱해서 서보 사이클에 대한 **time-base** 를 구하게 됩니다. (이 타임 베이스 값은 위치 업데이트 계산에서의 곱셈 값이 되며, 업데이트 량은 마지막 서보 사이클에서의 타임 베이스 신호로부터의 카운트 값에 비례하게 됩니다.)

이번 단계에서의 두 가지 설정 아이템은 소스에 대한 정보(보간된 위치 레지스터)와 스케일 인자입니다. 이 둘은 모두 엔코더 변환 테이블의(I 변수로 된) 항목들(entries)입니다. 이 값을 입력하는데 대한 보다 상세한 사항은 엔코더 변환 테이블에 대한 설명을 참조하시기 바랍니다.

Time-base 변환에 대한 식은 다음과 같습니다 :

$$\% \text{ value} = (100.0 * \text{SCALE_FACTOR} * \text{INPUT_FREQ}) / 2^N$$

여기서 %값(federate override 값으로 이미 알고 있음)은 위치 업데이트에 대한 속도를 제어합니다. 이 값이 100.0 이 되면, 프로그램과 이동명령은 "실시간"으로 동작하게 됩니다. (다시 말해, 시간과 속도가 프로그램에서 설정한 대로 움직인다는 말입니다.)

Time-base 추종(제어)을 정상적으로 설정하기 위해서는 'SCALE_FACTOR' 값이 반드시 정수로 정해져야 합니다. 'INPUT_FREQ'는 I7mn0 과 입력신호에 의해서 결정되는 counts/msec 단위의 카운트 속도입니다. 'N'은 일반적인(untriggered) time-base 에서는 '17'이며 ($2^{17} = 131072$), triggered time-base 에서는 '14'입니다 ($2^{14} = 16384$) . Triggered time-base 에 관한 자세한 사항은 뒤에 다시 언급하도록 하겠습니다.

프로그램과 이송이 지정된 속도에서 수행되도록 하기 위해서는, 물론 실시간 입력 주파수에 의해서 결정되긴 하지만, 스케일 팩터(scale factor)를 설정해야 합니다. 이러한 경우는 비율이 100.0 이 되며, 스케일 팩터에 대해서 식을 풀면 아래와 같습니다.

$$\text{SCALE_FACTOR} = 2^N / (\text{REAL_TIME_INPUT_FREQ})$$

스케일 팩터 값은 반드시 정수가 되어야 하고, 2 의 승수로 나눌 것이기 때문에, 실시간 입력 주파수를 counts/msec 단위의 2 의 승수로 결정해 주어야 합니다. 예를 들어, 최대 속도가 60000 counts/sec 의 입력 주파수를 가지는 시스템이 있다고 가정합니다. 이때, 실시간 입력 주파수는 64 counts/msec 가 되도록 설정합니다. 이 값은 untriggered time-base 에 대해서 스케일 팩터 값을 ' $131072/64 = 2048$ ' 로 만들어 줍니다.

지금까지 설명한 내용은 마스터 주파수에 대한 비례한 레지스터의 설정 값입니다. 이제는 이것을 모션 프로그램에 적용하는 방법에 대해서 설명하겠습니다.

Step 4 : Using the Time-Base Calculation

Time-base 값은 좌표계에서 동작합니다. 각 좌표계마다 이 좌표계가 time-base 정보를 어디서 찾을 것인가를 설정하는 I 변수가 있습니다. 이 값은 좌표계 x 에 대해서 Isx93 에 설정됩니다. (sx = 51~ 66 for C.S.1 ~ C.S.16). Isx93 에 대한 기본 값은 소프트웨어 제어 하에 있는 레지스터의 어드레스를 가리키고 있으며, 외부 주파수에 의한 제어를 가리키지는 않습니다. 외부 time-base 를 사용하려는 좌표계에 대해서는, 반드시 엔코더 변환 테이블에서 결정된 스케일 된 time-base 값의 주소를 넣어주어야 합니다.

예를 들면, 엔코더 변환 테이블이 여덟 개의 1/T 엔트리들(I8000~I8007 을 사용하는)을 사용할 때, 이어서 'I8008', 'I8009'를 사용해서 time-base 엔트리를 입력하면, time-base 의 변환 결과는 I8009 (address X:\$350A)에 해당하는 X-register 에 쓰여지게 됩니다. 이 값을 좌표계 1 번의 time-base 값으로 쓰기 위해서는 'I5193=\$350A'(어드레스를 바로 지정) 또는 'I5193=@I8009'(I 변수의 위치로 어드레스를 간접 지정) 명령을 사용하면 됩니다. 두 명령 모두 같은 결과를 나타냅니다.

이 I 변수가 한 번 설정되고 나면, 이 좌표계에 할당되어 있는 모든 모터들은 프로그램 또는 일반 이송(Jog 또는 Home)에서도 모두 외부 주파수에 의해 제어를 받게 됩니다.

I 변수 'Isx94'는 좌표계 x 에 대한 time-base 값의 최대 변화율을 제어하게 됩니다. Time-base 값을 호스트에서 명령할 때에는 (%n 명령을 사용해서), 새로운 time-base 값에 대해서 서서히 변화도록 충분히 낮게 설정해야 합니다. 그러나, time-base 소스로서 외부 신호에 동기를 유지하기 위해서는, 신호에 대해서 time-base 가 항상 변화할 수 있도록 이 값을 가능한 크게 설정해야 합니다.

(최대값은 8388608 입니다.) 이 값을 낮게 설정하는 것은 부드럽게 따라가게 할 수 있지만, 추종 시에 '틀어짐'이 발생할 수 있습니다. 외부 time-base 에서 Isx94 의 제한 값이 사용된다면, 주축에 대한 위치 동기를 잃어버릴 수 있습니다.

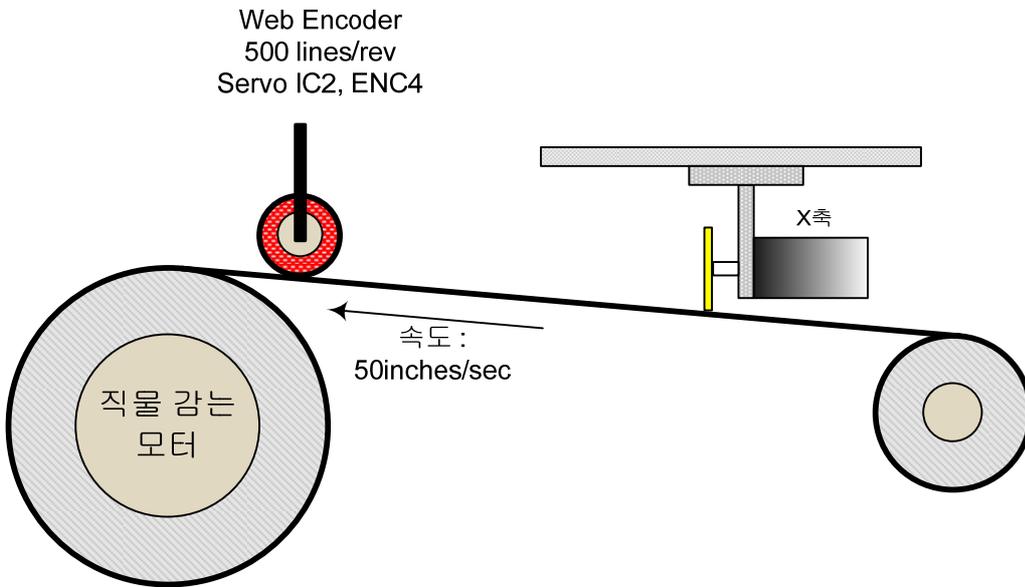
Step 5 : Writing the Program (프로그램 작성하기)

외부 time-base 제어상에서 프로그램을 작성할 때에는, 입력 신호가 항상 실시간 주파수에 있다고 생각하고 단순하게 작성합니다. 이제 동작할 때에는, 프로그램은 입력 주파수에 비례해서 실행하게 됩니다. 이송 시간과 속도에 있어서는 실수 값으로 설정하는 것이 가능합니다.

'DWELL' 명령은 입력 주파수에 상관없이 항상 실시간으로 동작한다는 것을 명심해야 합니다. 프로그램 내부에서 입력 주파수에 비례해서 잠시 멈춤을 하고 싶을 때에는, 'DWELL'이 아닌 'DELAY'를 사용해야 합니다.

Time-Base Example

만약에 초당 50 인치의 공칭 속도로 움직이는 직물이 있다고 하면, 거기에 인치 당 500 라인의 4 체배 엔코더를 설치합니다. 또한 Turbo PMAC 에서 제어하는 가로질러 자르는 축이 있습니다. 직물이 공칭 속도로 움직일 때에, 0.75 초 동안에 자르는 축이 움직이고 다음 2.50 초 동안에 다른 동작을 시작 합니다. 직물의 엔코더는 Servo IC 2 번의 엔코더 4 번에 입력되어 있습니다.



[Standard Time-base, 주축과 종축]

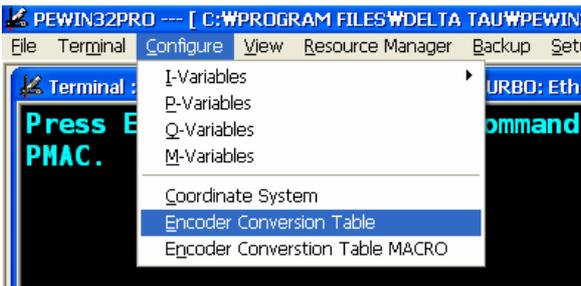
Step 1 : Signal Decoding

직물의 엔코더가 Servo IC 2 번의 엔코더 4 번에 연결되어 있으므로, 'I7240'으로 디코딩 방법을 설정합니다. 최대 해상도로 설정하기 위해서는 'I7240'을 '3' 또는 '7'로 입력해서 4 체배로 설정합니다. 먼저 '3'을 설정해 봅니다. 매뉴얼에 있는 'Suggested M-Variable'(권장 M 변수) 을 보면, 이 엔코더에 대해서 UMAC 은 'M401', 일반 Turbo PMAC 은 'M1201'이 위치 값을 나타내는 M 변수로 정의되어 있습니다. 'M401' 또는 'M1201'에 대한 M 변수를 정의하고, 직물의 엔코더를 어플리케이션에서 사용할 방향으로 돌리면서, 이 값을 반복해서 읽어보도록 합니다. (PEWIN 의 Watch Window 창을 이용하시면 됩니다.). 엔코더가 돌아갈 때에 이 값이 증가하면 'I7240'은

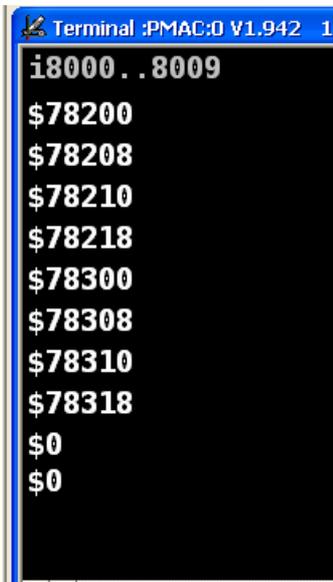
정상적으로 설정된 것입니다. 만약에 이 값이 감소하면 'I7240'을 '7'로 변경하십시오. (만약 값이 변하지 않으면, 엔코더 결선 상태를 확인하십시오.)

Step 2 : Interpolation

다음으로, 엔코더 변환 테이블의 현재 설정을 살펴 봅니다. 엔코더는 반드시 1/T 변환을 사용해서 처리되도록 해야 합니다. 이를 설정하는 가장 쉬운 방법은 PEWIN의 Configuration 메뉴를 사용하는 것입니다.



만약 보는 것이 여의치 않으면, 바로 I8000~의 변수를 살펴 봅니다. 다음과 같이 'I8000..8009'와 같은 온라인 명령을 쓰고, 아래와 같은 응답을 받아 옵니다.



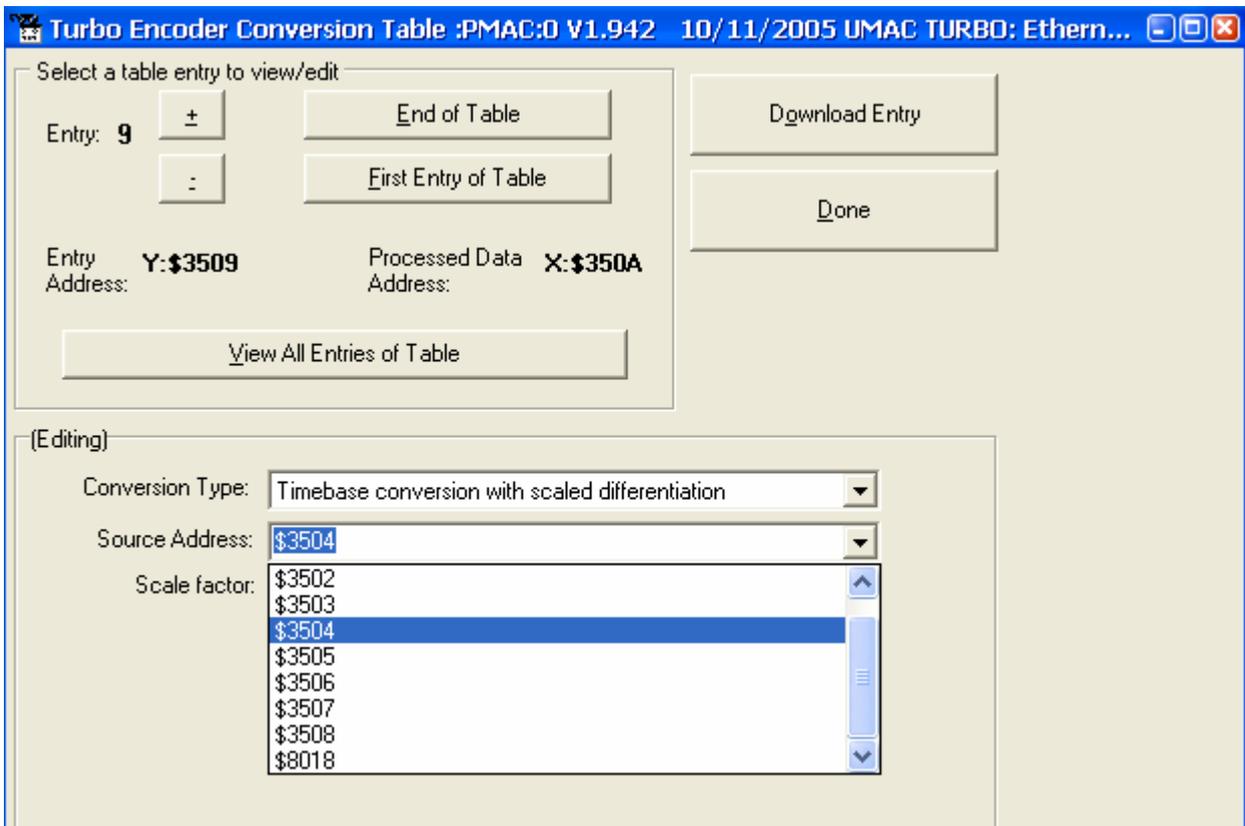
'I8000' 변수에 대한 상세한 설명은 "Software Reference Manual"을 참고하시기 바랍니다. 위의 테이블에서는 Servo IC 2 와 3 번의 4 채널 모두에 대해서 1/T 변환을 사용하고 있으므로, 우리가 사용할 주축 엔코더도 처리가 되고 있습니다.

Step 3 : Time-Base Calculation

이제는 보관된 위치를 time-base 형식으로 변환하기 위해서 테이블의 항목을 설정해야 합니다. 위에 리포팅 된 값들을 저렇게 날로 보거나, 또는 PEWIN의 Configuration 메뉴에서 좀더 명확하게

보거나, 마찬가지로 time-base 설정 항목이 들어있지 않다는 것을 알 수 있으므로, 하나를 반드시 생성해야 합니다.

소스(source) 엔코더는 테이블의 4 번째 라인에서 'I8003'으로 처리되고 있습니다. 이 항목의 어드레스를 찾아보면 '\$3504'인 것을 확인합니다. 만약에 PEWIN 의 Configuration 메뉴를 사용한다면, 테이블의 맨 마지막에 항목을 추가하면 됩니다. 이때, Time-Base 방법을 선택하고 어드레스를 입력합니다. (리스트에서 선택해서 넣거나, 직접 \$3504 어드레스를 입력합니다.). Turbo PMAC 에 바로 설정할 때에는, 온라인 명령으로 'I8008=\$403504'라고 입력하면 됩니다. (맨 앞의 '4'는 time-base 항목을 의미합니다.)



이제는 스케일 팩터(scale factor)를 계산할 차례입니다. 공칭 속도가 50 inches/sec 이고, 엔코더의 해상도가 500 cycles/inch 이며, 4 체배를 사용합니다. 계산하면 아래와 같습니다.

$$\begin{aligned}
 &50 \text{ inches/sec} * 500 \text{ cycles/inch} * 4 \text{ counts/cycle} \\
 &= 100,000 \text{ counts/sec} \\
 &= 100 \text{ counts/msec}
 \end{aligned}$$

수학 계산에서 결과가 2 의 승수이면 매우 처리하기 좋으므로, 실시간 카운트 속도를 128 count/msec 로 선언합니다. 그리고 나면 스케일 팩터는 $131072/128 = 1024$ 로 계산됩니다. 만약 PEWIN 을 사용한다면 이 값을 적당한 필드(scale factor 항목)에 넣으시기 바랍니다. Turbo PMAC 에 직접 설정하실 때에는, 온라인 명령으로 'I8009=1024'라고 입력하시면 됩니다.

Step 4 : Using the Time-Base Calculation

위 설정은 좌표계 1 번에서 동작하기 때문에, 이 time-base 값을 가리킬 수 있도록, 이 항목의 두 번째 라인의 어드레스를 I5193 에 할당합니다. ('I5193=@I8009 또는 'I5193=\$350A'). 'I5194'는 최대 값인 '8388608'을 설정해서 급격한 변화에 동기를 잃어버리지 않도록 합니다.

Step 5 : Writing the Program

프로그램을 작성할 때에는, 실시간 입력 주파수에서 동작한다고 하면, 이 경우에는 공칭 속도에서 동작하는 것과는 달리 대략 28%정도 빨리 동작하게 됩니다. 그러므로, 모든 프로그램의 속도는 28% 더 빨라지게 되고, 프로그램의 시간은 28% 짧아지게 됩니다. 공칭 컷 시간을 750msec (0.75sec)로 선택하고, 이 값에 다시 100/128 을 곱하면 정확하게 585.9375msec 를 얻게 됩니다. 2500msec 도 이와 비슷하게 스케일 되어 1953.125msec 를 얻게 됩니다. (만약에 이 숫자가 정확하게 나누어 떨어지지 않으면, 프로그램에 수식을 바로 입력하셔도 됩니다. Turbo PMAC 은 48-bit 정밀도의 실수 연산을 수행하게 됩니다.). 메인 프로그램의 루프는 아래와 같습니다.

```
WHILE (M11=1) ; Cut as long as input is true
TM 585.9375 ; Cut move time
X10000 ; Actual cut move
DELAY 500 ; Hold; part of 1953.125 msec return
TM 953.125 ; Return time; part of 1953.125 msec
X0 ; Actual return move
DELAY 500 ; Hold; part of 1953.125 msec return
ENDWHILE
```

Triggered Time Base

이제껏 설명했던 **time-base** 기술은 슬레이브 좌표계가 전적으로 마스터(주축)에 동기해서 움직이는 것에 대해 설명하였습니다. 그러나, 이러한 동기는 마스터의 특정 위치에 대해서 동기화하는 방법을 제공하지는 않습니다. 슬레이브의 동작 사이클은 마스터 사이클에서 일시적으로 벗어날 수 있고, 'registration mark'에 의한 위치 캡처 등을 포함한 몇몇 특별한 기술에 대해서는 서로 위상을 맞추어서 사이클을 동작시킬 필요가 있습니다.

대부분의 **time-base** 어플리케이션들은 마스터와 슬레이브 사이클이 서로 위상을 맞추기 위해서 늘어남, 미끄러짐, 불규칙한 유격 때문에 지속적으로 재등록 해야 하는 경우를 제외하고는 별다른 동작을 필요로 하지 않습니다.(예를 들어 프린트되지 않은 그냥 종이를 길이에 따라 자를 경우) 이러한 어플리케이션들은 표준 **time-base** 기능을 사용할 수 있습니다.

그러나, 마스터에 대해서 위상을 맞추어야 하는 어플리케이션들이 있으며, 이들은 위상을 맞추기 위한 등록 절차가 어렵거나 불가능한 경우입니다. 이때는 엔코더 변환 테이블의 **triggered time-base** 기능을 사용할 수 있습니다. 이러한 기술은 **trigger** 에 의해서 캡처된 마스터의 위치에 대해서 완벽한 동기화를 구현할 수 있게 합니다. 이는 **trigger** 가 들어오기 전까지는 **time-base** 를 정지시켜 두었다가, **trigger** 에 의해 캡처된 위치에 대해서 **time-base** 를 시작시킴으로써 가능하게 합니다.

엔코더 변환 테이블에 있는 **Triggered time-base** 의 설정항목은 표준 **time-base** 설정과 비슷합니다. 이것은 두 줄의 항목이며, 첫째 줄은 처리방법을 규정하고 마스터 엔코더에 대한 소스 어드레스를 설정하는 것이고, 둘째 줄은 **time-base** 스케일 팩터를 설정하는 것입니다. **Triggered time-base** 와 표준 **time-base** 설정항목에는 크게 세 가지의 차이점이 있습니다. 첫째로, 처리방법을 규정하는 값이 다르고, 이 값은 트리거(trigger) 처리 과정에 따라서 값이 변하게 됩니다. (\$90,\$A0,\$B0, 반대로 표준 **time-base** 에서는 \$40 으로 고정.). 둘째로, 소스 어드레스는 변환 테이블의 처리된 엔코더 데이터가 아니라, 실제 마스터 엔코더의 카운터 레지스터입니다. 이때 주의할 점은, **PMAC2** 타입에 대해서는 반드시 **Bit19** 를 '1'로 설정하는 것입니다. 셋째로, 스케일 팩터는 표준 **time-base** 와는 달리 $2^{14}/RTIF$ ($16384/RTIF$) 로 계산된 값을 사용해야 합니다.(일반 **PMAC** 은 그대로 $2^{17}/RTIF$ 를 사용함.). 이 항목들에 대한 보다 상세한 내용은 엔코더 변환 테이블에 대한 설명을 참고하시기 바랍니다.

Instructions for the Triggered Time-Base

Triggered time-base 기능을 사용하기 위해서는 적당한 **I** 변수 설정, **M** 변수 정의, 변환 테이블 항목, 모션 프로그램 작성, **PLC** 프로그램 작성 등이 필요하게 됩니다. 이 과정을 아래에 하나씩 설명하도록 하겠습니다.

Step 1 : Signal Decode Setup

마스터 신호의 해석 부분은 표준 **time-base** 설정 부분과 동일합니다. 4 체배 또는 펄스와 방향 신호는 반드시 카운트가 증가하는 방향으로 해석되어야 합니다. 이는 'I7mn0'로 설정합니다.

Step 2 : Interpolation and Time-Base Setup

Triggered **time-base** 변환은 $1/T$ 카운트 보간과 보간된 값으로부터 **time-base** 계산을 수행하는 것을 동시에 처리하게 됩니다. 이 방법에서는, $1/T$ 보간은 3 개의 추가적인 나머지 카운트 비트를 생성하고(총 8 bit), 그 결과 수치를 "일반" $1/T$ 값보다 8 배 더 크게 만듭니다. 초기 설정 시에, triggered **time-base** 설정항목은 일반적으로 실행상태(잠금상태나 트리거를 대기상태가 아닌)로 변환 테이블에 생성되게 됩니다. **Time-base** 스케일 팩터 또한 여기서 설정되는데, 이 값은 실시간 입력 주파수에 대해서 $1/8$ 인 것을 제외하고는, 표준 **time-base** 설정부분과 동일하게 계산됩니다.

Step 3 : Writing the Motion Program

Triggered **time-base** 에 사용될 모션 프로그램을 작성할 때에는, 모든 축들이 트리거를 기다리는 위치로 이동해서 정지해 있도록 해야 합니다. 이 부분이 모션 시작 부분에 있지 않으면, 그 부분은 바로 'DWELL' 명령으로 진행되도록 설정해야 합니다.

트리거(trigger)에 의해서 시작될 이송에 대한 계산의 처음 부분에서, 시작하자마자 움직이지 않도록 **time-base** 는 반드시 잠금(frozen)상태로 되어 있어야 합니다. 이 부분은 변환 테이블의 triggered **time-base** 항목의 처리 비트에 할당된 M 변수를 사용하면 간단히 구현이 가능합니다. 만약에 이전의 이송이 다른 **time-base** 소스에 의해서 움직였다면, 그 좌표계에 대한 **time-base** 어드레스인 Isx93 값은 반드시 triggered **time-base** 항목으로 바뀌어야 합니다.

모션 프로그램 내에서의 이러한 명령들은 트리거 된 이후에 처음으로 이송하는데 대한 명령과 계산 바로 뒤에 따라 와야 합니다. **Time-base** 잠금 상태에서는, Turbo PMAC 은 모든 계산을 수행하지만, 이 이송에 대한 실제 구동은 하지 않습니다. Turbo PMAC 은 **time-base** 가 시작되자마자 움직일 준비가 되도록, 변수 I11(계산 지연시간)값을 반드시 '0'으로 설정해야 합니다.

Step 4 : Arming the Trigger

이송을 계산하는 모션 프로그램은 이 계산이 끝나기 전에 트리거가 발생하지 않고서는, 자체적으로 트리거를 장전(arm)시킬 수 없습니다. 만약 이런 일이 발생한다면, 프로그램은 우리가 의도한 동기화를 벗어나게 되는 것입니다. 그러므로 안정된 동작을 위해서는, 트리거는 반드시 PLC 프로그램과 같이 이송 계산이 끝나기 전에는 수행되지 않는 **task** 에서 장전되어야 합니다.

트리거를 발생시키는 데에는 PLC 에서 간단한 조건 분기를 작성하는 것으로 충분합니다. 이는 단지 time-base 가 잠금 상태인지를 보고, 그 경우에 PLC 프로그램에서 트리거를 장전하는 것입니다. PLC 프로그램은 모션 프로그램 실행 중에 끼워들 수 없기 때문에, 모션 프로그램이 이송에 대한 계산을 끝내고 난 후에 트리거를 장전할 수 있도록 보장합니다.

Step 5 : Starting on the Trigger

트리거가 일단 장전되고 나면, Turbo PMAC 은 마스터 엔코더에서 위치 캡처 트리거가 발생하기를 기다립니다. 변수 I7mn2 와 I7mn3 이 어떤 신호의 어떤 에지(Edge)에서 트리거를 발생할 것인지를 결정하게 됩니다. Turbo PMAC 에서 트리거가 발생하면, 캡처된 위치를 time-base 의 시작점으로 사용해서 곧바로 time-base 를 시작하게 됩니다.

Triggered Time-Base Example

좌표계 1 번에 모터 1 번이 있습니다. 이는 회전당 2500 라인의 분해능을 가진 엔코더가 장착된 회전 모터이며, 이 모터에서 부하단 까지는 3 : 1 의 비율로 감속되어 있습니다. 이 모터는 Turbo PMAC 의 Servo IC 2 번의 4 번째 엔코더에 연결된 마스터 엔코더에 종속되어 동작하게 됩니다. 마스터 엔코더는 회전당 4096 라인의 분해능을 가졌고, 보통 600 rpm 정도로 동작합니다. 동작 명령이 내려진 후에는, X 축이 마스터의 인덱스 펄스(index pulse)입력을 기다렸다가, 45 도를 지나가게 됩니다. 다음 마스터의 36 도 회전에 대해서, 이 축이 가속을 마쳐야 하고, 그런 다음에 144 도 회전동안 동작하고 마지막 36 도 동안에 감속을 마치게 됩니다. 이 동작을 수행하면서 A 축은 반드시 완전한 1 바퀴를 이송해야 합니다.

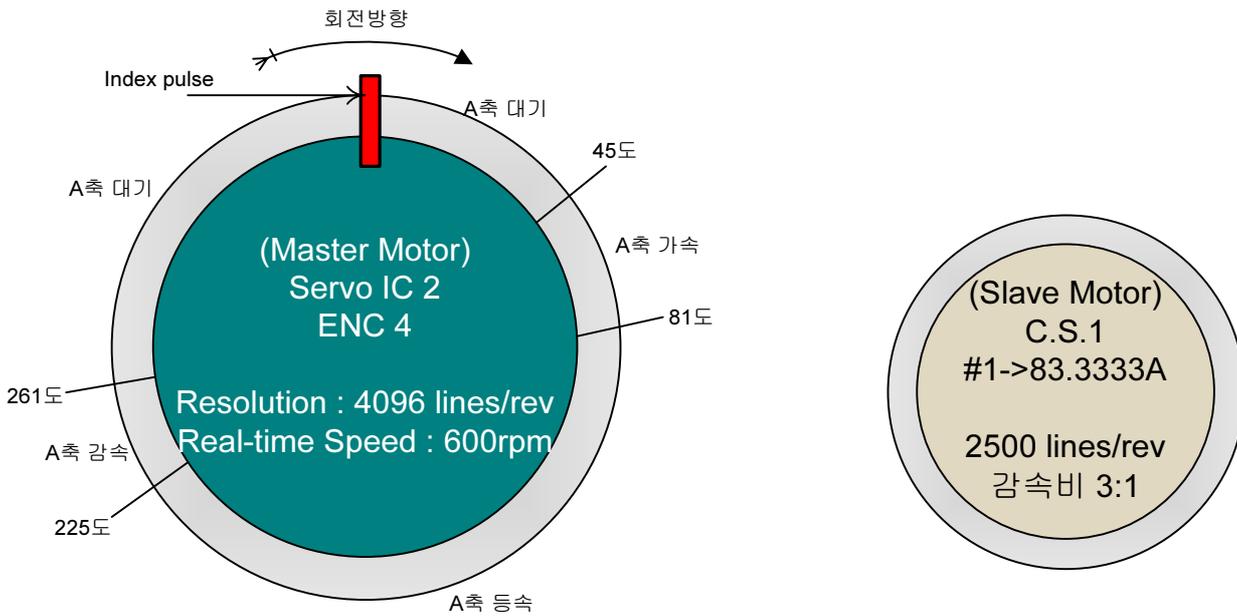
우리는 triggered time-base 를 사용할 것이며, 마스터 엔코더의 인덱스 신호를 사용해서 트리거를 할 것입니다. 600 rpm 을 마스터의 "실시간"속도로 선택하고, 실시간 입력 주파수(RTIF)를 counts/msec 단위로 계산할 것입니다.

$$600 \text{ rev/min} * (\text{min}/60 \text{ sec}) * (4096 \text{ lines/rev}) * (4 \text{ counts/line}) * (\text{sec}/1000 \text{ msec}) \\ = 163.84 \text{ counts/msec}$$

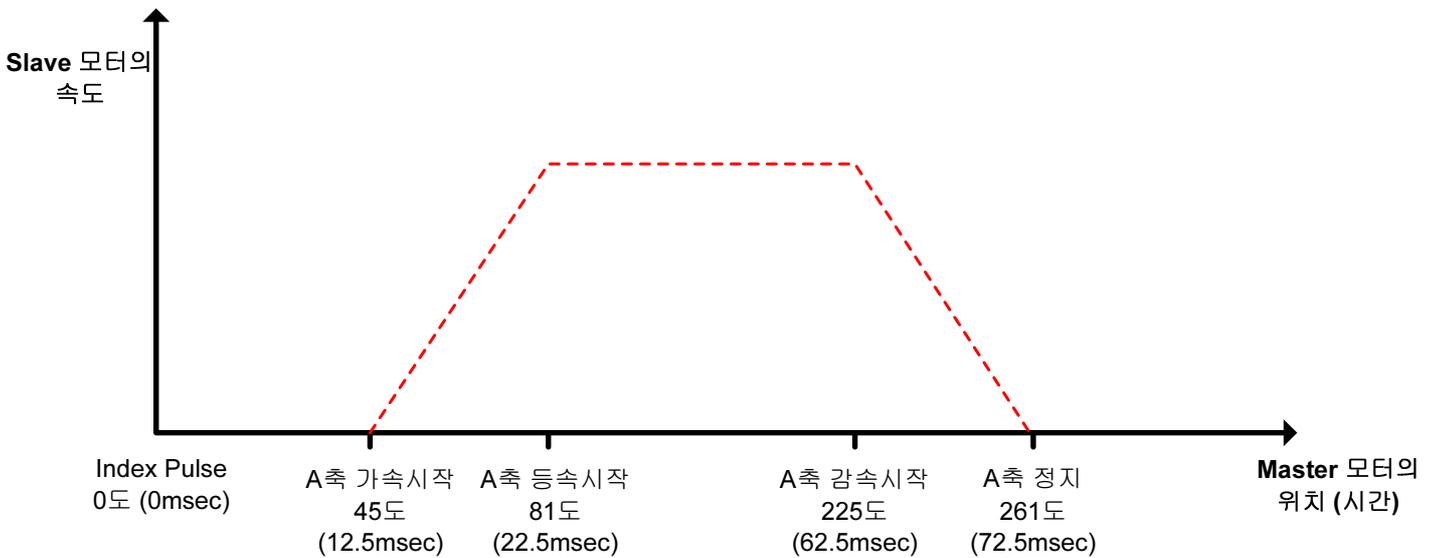
Time-base 스케일 팩터(SF)는 다음과 같습니다.

$$SF = 16384 / RTIF = 16384 / 163.84 = 100 \text{ (decimal)}$$

실시간 속도인 600 rpm(10rps)로 동작할 때에, 마스터의 1 회전은 100 msec 의 시간에 해당됩니다. 따라서, 마스터의 45 도는 12.5msec 에 해당합니다.



[주축과 종축간의 관계, Index Pulse에 Trigger]



[주축과 종축의 구동 타이밍 그래프]

Set-up and Definitions

I7240 = 3 ; Servo IC 2 번의 4 번째 채널에 대해서 4 체배로 디코딩하고,
; 모터 이동에 따라서 카운트가 증가하도록 설정해야 합니다.

I7242 = 1 ; Enc 4 번의 index pulse 의 rising edge 에서 트리거를 캡처하도록 설정합니다.

I8008 = \$AF8218 ; 기본 변환 테이블의 마지막에 triggered time-base 항목을 추가합니다.(UMAC 의 경우 PMAC2)

; \$A 처리는 triggered time-base 설정에서, 실행상태(트리거 된 후)입니다.

; \$78218 은 IC 2 의 ENC4 레지스터를 가리킵니다.

I8009 = 100 ; 스케일 팩터는 100 입니다. 결과는 \$350A 어드레스에 저장됩니다.

M199->Y:\$3509,20,4 ; 변환 테이블 항목의 방법 설정 = \$9 잠금상태, \$B 장전상태, \$A 실행상태

&1 ; 좌표계 1 번을 설정합니다.

#1->83.33333333A ; 모터 1 번은 C.S.1 번에 A 축으로 정의, $3*2500*4 \text{ cts/rev}/(360\text{deg/rev})$

Motion program

모션 프로그램은 time-base 를 잠그고, 처음 이송 부분을 계산합니다. 그러나, 실제 이 이송의 실행은 트리거가 발생하고 나서야 이뤄집니다.

CLOSE

OPEN PROG 12 CLEAR

I5193=\$350A ; time-base 소스 어드레스를 변환 테이블의 triggered time-base 로 변경합니다.

; (2 번째 라인의 어드레스)

DWELL 0 ; 프로그램에서의 선행연산을 멈춥니다.

M199 = \$9 ; time-base 를 잠금 상태로 바꿉니다.

LINEAR ; Linear move mode

INC ; 상대위치 이동으로 설정

TA 10 ; 마스터의 36 도에 해당하는 10msec 를 설정.

TS 0 ; No S-Curve

DELAY 12.5 ; 마스터의 45 도에 해당하는 12.5msec 를 설정.

TM 50 ; 마스터의 36+144 도에 해당하는 50msec 를 설정.

A 360 ; 슬레이브 축의 완전한 한 바퀴 이송

CLOSE

PLC program

PLC 프로그램은 단순히 time-base 가 잠겼는지를 감시하게 됩니다. 일단 잠기고 나면, time-base 를 장전하게 됩니다. 장전된 상태에서는, triggered time-base 변환 테이블에서는 매 서보 사이클마다 트리거가 발생하기를 기다립니다.

CLOSE

OPEN PLC 10 CLEAR

IF (M199=\$9) ; time-base 가 잠긴 상태인가?

M199 = \$B ; 트리거를 장전합니다.

ENDIF

CLOSE