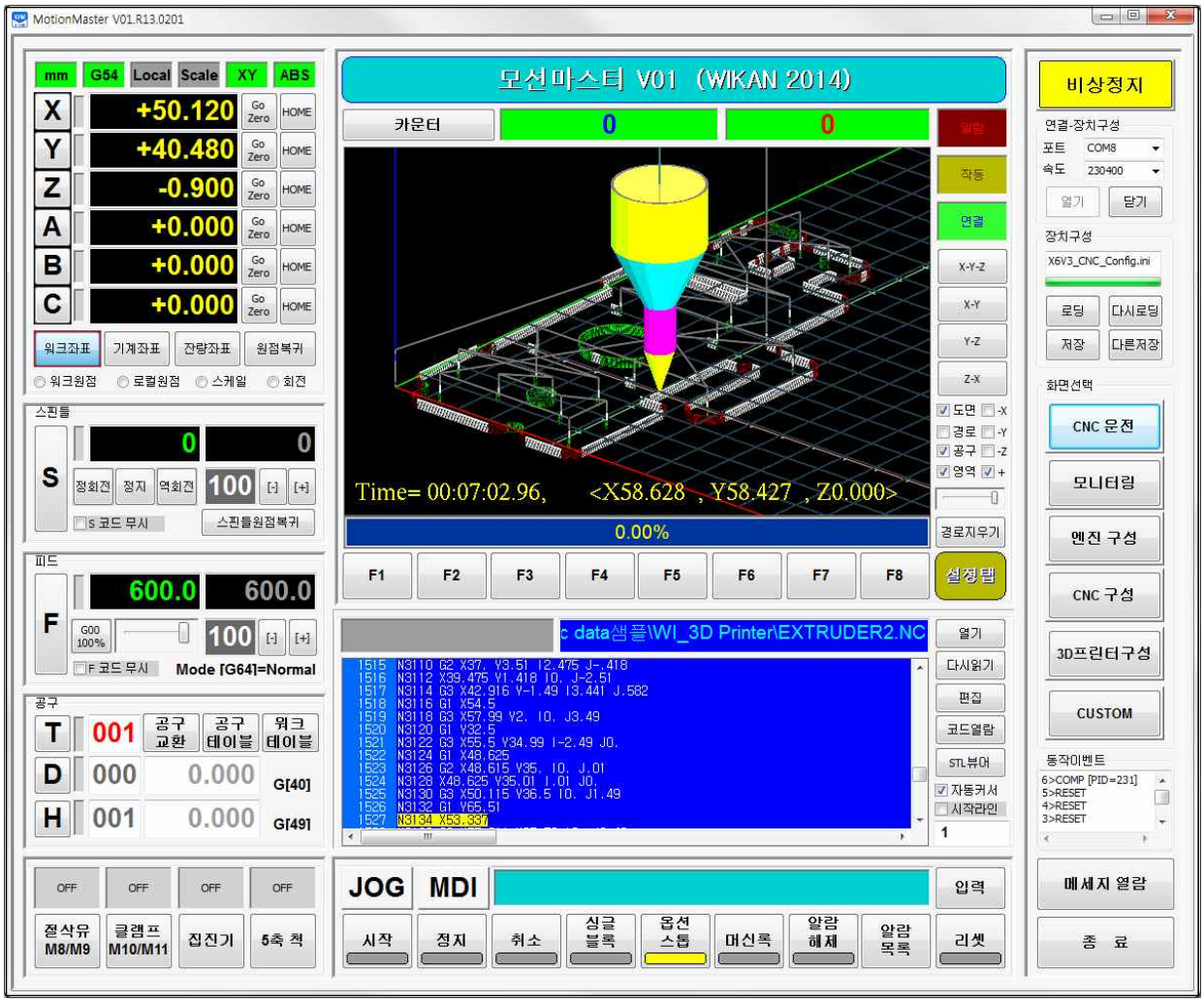
모션 마스터
(Motion Master V01)



WIKAN.CO.KR
All rights reserved
2017-10-10

경고 및 주의 사항

주의사항은 '경고'와 '주의' 로 구분되고, 심볼마크의 정의는 아래와 같습니다.

경고

지시사항을 위반하였을 때, 심각한 상해나 사망 사고가 발생할 가능성이 있는 위험한 상황을 나타냅니다.

주의

지시사항을 위반하였을 때, 경중상을 입을 가능성이 있는 위험한 상태나 물적 손해사고를 입을 가능성이 있는 상황을 나타냅니다.

경고

1.생명이나 재산상의 영향이 큰기기에 사용할 경우 반드시 2중으로 안전장치를 부착해서 사용하십시오. 화재,인사사고,재산상의 막대한 손실이 발생할 수 있습니다.

2. 동작 온도,습도,기압 등 사용환경 범위를 벗어나는 곳에서는 사용하지 마십시오.

3. 국내,국제법상 허가되지 않거나 불법적인 응용분야등에 사용하지 마십시오

4. 부식성 가스나 폭발성 가스가 있는 장소에서는 사용하지 마십시오.

5. 기기가 위험한 상황을 초래할 가능성이 있는 장소에는 방호 대책을 세워 사용하십시오.

6. 감전 방지를 위해 취부 탈착 시 또는 배선 작업을 할 경우에는 전원을 차단하여 주십시오.

7. 임의로 제품을 개조하지 마십시오. 화재 위험이 있습니다.

8. 제품 동작 중에는 커넥터를 분리하거나 전원을 차단하지 마십시오. 인사사고, 재산상의 손실, 오동작 위험등이 있습니다.

주의

1. 전원을 인가하기 전에 반드시 접속이 정확한지 확인하십시오.

2. 부식/인화/폭발 위험, 물기 및 진동이 많은 곳에서 사용하지 마십시오.

3. 손상방지 및 고장방지를 위해 정격에 맞는 전원 전압을 사용하고, 성능을 초과해서 사용하지 마십시오.

4. 제품의 제어 입출력 신호를 확인 후 전원을 투입해야 예상치 못한 신호에 따른 장치파손이나 부상을 예방할수 있습니다.

5. 입출력 단자는 감전의 위험이 있으니 신체 및 통전물 접촉에 주의하십시오.

6. 동작 중에는 표면이 고온인 상태 일수 있으므로 제품을 만지지 마십시오.

7. 동작 중에는 항상 비상정지가 가능하도록 하십시오.

8. 이상이 발생했을 경우 화재나 부상의 우려가 있으므로 전원을 차단하십시오.

9. 절연 저항 측정이나 절연 내압 시험 시 감전의 우려가 있으므로 단자를 만지지 마십시오.

10. 청소 시 감전 및 화재의 우려가 있으므로 물기가 없는 마른 수건으로 청소하십시오.

11. 제품 폐기 시 산업폐기물로 처리하십시오

Document History

D/-*ate	Descriptions	Version
13/10/17	[배포] MotionMaster_v01_r13-user_manual-r01.pdf	R01.R13
14/01/08	[수정] MotionMaster_v01_r13-user_manual-r01.pdf - 매크로 설명부 추가	R01.R13.0163
14/01/10	[수정] MotionMaster_v01_r13-user_manual-r01.pdf	R01.R13.0164
14/12/23	[개정] R13.208 이후버전에 대한 전체적인 개정	R01.R13.0208
16/02/19	[개정] R13.313 이후버전에 대한 전체적인 개정	R01.R13.0313
17/10/10	[개정] R13.335 이후버전에 대한 전체적인 개정	R01.R13.0335

[차례]

1. 개요
1.1. 모션마스터 특징 ····11
1.2. 모션컨트롤러 ····12
1.2.1. 위치결정기와 CNC 컨트롤러 ····12
1.2.2. 위치제어형과 속도제어형 모션컨트롤러 ····12
1.2.3. PC 기반형, PLC 기반형 모션컨트롤러와 독립형 모션컨트롤러 ····12
1.3. 위칸 모션컨트롤러와 모션시스템의 연결 구성 예 ····14
2. 프로그램 설치/제거 및 시작하기 ····15
2.1. USB 드라이버 설치 ····15
2.1.1. USB로 장치 연결시 주의사항 ····15
2.1.2. 설치 환경 ····15
2.1.3. 드라이버 설치 시 주의사항 ····15
2.1.4. USB 드라이버 파일 ····16
2.1.5. 설치 ····17
2.1.6. 드라이버 설치 확인 ····19
2.1.7. 드라이버 제거 ····20
2.2. 프로그램 설치/제거 ····21
2.2.1. 프로그램 설치 환경 ····21
2.2.2. 프로그램 설치 파일 ····21
2.2.3. 프로그램 설치 ····21
2.2.4. 프로그램 실행 ····24
2.2.5. 프로그램 제거 ····25
2.2.6. 설치파일 구성 ····26
2.3. 프로그램 시작하기 ····27
2.3.1. 전체 프로그램 훑터보기 ····28
2.3.2. 모션컨트롤러의 연결과 구성파일 관리 ····29
2.3.3. 프로그램 실행 시작하기 ····30
2.3.4. 모니터링 화면에서 간단하게 움직여 보기 ····31
2.3.5. 직선보간, 원호보간 동작하기 ····33
3. 전체 동작화면 요약 ····37
3.1. 동작탭 화면 ····37
3.2. CNC (or 3D PRINT) 운전 화면 ····40
3.3. 모니터링 화면 ····41
3.4. 엔진구성 화면 ····42
3.5. CNC 구성 화면 ····43
3.6. MotionMaster_Main.ini 구성파일 ····44
4. CNC 장치 연결 기초 가이드 ····47
4.1. 모터드라이브 신호 연결 및 신호방식 ····48
4.1.1. 전기신호 ····48
4.1.2. 신호방식 ····50
4.1.3. 엔코더 피드백 신호 ····51
4.1.4. 그 외 축 부가 입출력 신호연결 ····52

4.1.5. 모션마스터 프로그램의 축신호 설정 ····53
4.2. 각 축별 위치 파라메터 설정 ····54
4.2.1. 각 축의 이동방향 설정 ····54
4.2.2. 모션보드의 축맵핑 ····55
4.2.3. 1펄스 단위이동량과 전자기어비 설정 ····56
4.2.4. 소프트리미트, 하드리미트에 의한 동작한계치 설정 ····57
4.2.5. 백래쉬 ····58
4.3. 수동 JOG동작을 위한 기초 설정 ····59
4.4. JOG 조작에 의한 축별 IO신호 점검 ····61
4.5. 원점복귀 파라메터 설정 ····66
4.5.1. 엔진구성 화면에서 동작방향과 동작모드 설정 ····66
4.5.2. CNC구성 화면에서 원점복귀동작 주요 파라메터 설정 ····67
4.5.3. 개별 축 원점복귀 동작 실행 ····68
4.5.4. 시퀀스 동작을 갖는 전체 원점복귀 동작 실행 ····69
4.5.5. 원점복귀 동작 반복위치 정밀도 검사 ····70
4.6. CNC 속도 동작제어 파라메터 설정 ····71
4.6.1. 절삭속도 특성파라메터 설정 ····72
4.6.2. 급속이송속도 특성파라메터 설정 ····73
4.6.3. 기계장치간의 절삭 속도 제어방식 선택 ····74
4.7. 주축장치 설정 ····75
4.8. 주변장치 및 기타 입출력장치 설정 ····76
5. CNC 자동 운전 ····77
5.1. NC 코드 프로그램 (G코드 프로그램) ····77
5.2. G 코드의 생성 방식 ····79
5.3. CNC 운전화면에서의 G코드 실행방식 ····80
5.4. XYZ 좌표계 이해하기 ····81
5.5. CNC운전 자동실행 ····83
5.5.1. 모션마스터 프로그램 실행 및 초기화 ····83
5.5.2. 원점복귀 실행 ····84
5.5.3. G코드파일 로딩 ····85
5.5.4. 가공원점(워크좌표) 설정 ····86
5.5.5. 실행 및 완료 ····88
6. 모니터링 화면 ····89
6.1. 모션엔진의 주요 상태확인 및 시스템명령 ····90
6.1.1. LINK ····90
6.1.2. ALARM ····90
6.1.3. ESTOP ····90
6.1.4. PAUSE ····90
6.1.5. STEP ····90
6.1.6. ALARM LIST ····91
6.1.7. 리셋버튼 ····92
6.2. 위치좌표 확인/설정 ····94
6.3. 속도기능 확인/설정 ····95
6.3.1. 속도모드 ····95
6.3.2. 속도배율 ····95

6.3.3. 선속도 일정축 ····95
6.3.4. 속도 오버라이딩 ····96
6.3.5. 속도성분들 ····96
6.3.6. 속도항목 빠르게 다른축에 복사하기 ····96
6.4. 축별 동작상태 확인 ····97
6.4.1. 동작상태 ····97
6.4.2. 축 신호상태 ····97
6.4.3. 서보신호 상태 ····98
6.5. 시간 그래프(Time Graph) 모니터링 ····99
6.6. IO(입출력) 포트의 모니터링 ····100
6.6.1. GIO 입력포트 ····100
6.6.2. ADC 입력 ····100
6.6.3. GIO 출력포트 ····101
6.6.4. PWM 출력 ····101
6.7. 3D 그래프 위치 모니터링 ····102
6.7.1. 3D 투영방식 ····103
6.7.2. 3D그래프 축선택 ····103
6.7.3. 3D그래프 배율선택 ····103
6.7.4. 3D드로잉 옵션 ····103
6.7.5. 3D 위치추적선 버퍼제어 ····103
6.7.6. 3D 위치추적선 버퍼사이즈 조정 ····103
6.7.7. 3D 그래프 화면 ····103
6.8. 모션엔진의 동작상태 확인 및 동작옵션 설정 ····104
6.8.1. 모션엔진 동작상태(STATUS) 항목설명 ····104
6.8.2. 모션엔진 동작옵션(OPTION) 설명 ····105
6.9. 모션 명령 실행 ····106
6.9.1. 동작 보조명령 ····106
6.9.2. 조그동작 ····106
6.9.3. 기본명령동작 ····107
6.9.4. 추가명령동작 ····112
6.10. 통신상태 및 엔진버전 표시 ····119
7. 엔진구성 화면 ····120
7.1. 모션엔진 주요 구성설정 ····121
7.1.1. 통신 ····121
7.1.2. 명령 ····122
7.1.3. BusyQueue ····122
7.1.4. 상태타이머 ····122
7.1.5. 입출력포트 모드 ····123
7.1.6. 위치모드 ····123
7.1.7. 위치표시 ····124
7.1.8. 알람옵션 ····124
7.1.9. 상태패킷 옵션 ····124
7.1.10. 인터럽트 발생옵션 ····124
7.1.11. 속도제어 옵션 ····125
7.1.12. 보간동작 옵션 ····126

7.1.13. 전자기어비 옵션 ····126
7.2. 모션엔진 축별 구성설정 ····128
7.2.1. 펄스 혹은 엔코더 ····128
7.2.2. 속도 제어 ····128
7.2.3. 리미트, 소프트리미트 ····129
7.2.4. 원점복귀 ····129
7.2.5. 백래쉬 / 드웰타임 ····131
7.2.6. 서보인터페이스 ····131
7.3. 모션엔진 GIO/EIO 입출력 구성설정 ····133
7.4. 모션엔진 구성설정 설명문 ····134
7.5. 업로드, 다운로드 기능 ····134
8. CNC 운전화면 ····135
8.1. 위치좌표 제어화면 ····137
8.2. 스핀들(주축) 제어화면 ····140
8.3. 피드(이송속도) 제어화면 ····141
8.4. 공구 및 보정 제어화면 ····142
8.5. 주변장치 제어화면 ····144
8.6. 메인 모니터화면 ····145
8.6.1. 안내표시 ····146
8.6.2. 자동카운터 관리 ····147
8.6.3. 3D 그래프 ····148
8.6.4. 시그날(신호) 램프 ····149
8.6.5. 그래픽 옵션 ····149
8.6.6. 진행률 표시 ····149
8.6.7. 사용자 기능키 ····150
8.6.8. 동작탭 화면전환 ····151
8.7. NC코드파일 관리화면 ····152
8.8. 메인 조작화면 ····153
8.8.1. JOG 운전방식 ····153
8.8.2. MDI 운전 ····157
8.8.3. 자동 운전 ····157
9. CNC 구성화면 ····159
9.1. 축별 위치정보 설정 ····160
9.2. 축별 속도 파라메터 설정 ····162
9.3. 축별 원점복귀 파라메터 설정 ····163
9.4. CNC 메인구성 설정 ····165
9.4.1. 메인 설정 ····165
9.4.2. CNC코드옵션 ····165
9.4.3. 사이클 동작옵션 ····166
9.4.4. 추가설정 ····166
9.5. CNC 속도구성 설정 ····167
9.5.1. 절삭피드 속도설정(G01,G02,G03) ····167
9.5.2. 급속이송 속도설정(G00) ····168
9.5.3. 스핀들 회전속도 설정 ····168
9.5.4. 추가 속도제어 설정 ····172

9.6. CNC 보조장치 설정 ····175
9.6.1. M코드 보조입출력 설정 ····175
9.6.2. 전용입출력 설정 ····175
9.6.3. 제어스위치 설정 ····176
9.6.4. ELX36 확장보드 연결 ····176
9.6.5. MPG 연결옵션 ····177
9.6.6. JOG 동작옵션 ····178
9.6.7. 사용자 변경 버튼 / 램프 설정 ····179
9.7. CNC 동작 테이블 ····180
9.7.1. 데이터 관리 테이블 ····180
9.7.2. 메모리 관리 테이블 ····181
9.7.3. 원점 관리 테이블 ····181
9.7.4. 워크좌표 테이블 ····182
9.7.5. 공구 테이블 ····182
9.7.6. 매크로 테이블 ····183
9.7.7. 자동 프로그램 테이블 ····183
9.8. 기타 설정 ····184
9.9. 설정 적용과 결과 리포트 ····184
10. 매크로 확장과 V스크립터 ····185
10.1. 기본문법 ····185
10.1.1. V 스크립트 코드블럭 { } ····186
10.1.2. 제어문 ····188
10.1.3. 매크로기능으로 헬리컬보간 만들기 ····189
10.2. V스크립트언어 연산자 및 함수들 ····190
10.2.1. 연산자 ····190
10.2.2. 수학함수 ····191
10.2.3. PLC 기능지원 ····192
10.2.4. PLC 기능지원 : M 메모리 관련함수 ····192
10.2.5. PLC 기능지원 : 타이머 관련함수 ····192
10.2.6. PLC 기능지원 : 입출력 관련함수 ····193
10.2.7. 시간 관련함수 ····194
10.2.8. 메모리 관련함수 ····195
10.2.9. 문자열 관련함수 ····196
10.2.10. 파일 관련함수 ····197
10.2.11. 사운드 관련함수 ····198
10.2.12. 그래프 관련함수 ····198
10.2.13. 공통다이얼로그함수 ····198
10.2.14. 메시지박스 함수 ····199
10.2.15. 내부 인터프리터 관련함수 ····199
10.2.16. 시스템 명령 및 기타함수 ····199
10.2.17. DLL 관련함수 ····200
10.2.18. 시스템 메모리 데이터 액세스 ····200
10.2.19. 모션 지원함수 ····201
10.2.20. 모션 지원함수 2 ····203
10.2.21. 모션 지원함수 3 ····208

11. 외부 매크로 호출 ····209
11.1. 외부 매크로 호출 테이블 ····209
11.2. 외부 매크로 호출 사용법 ····210
11.2.1. 호출 하는 쪽(부모코드) ····210
11.2.2. 호출 되는 쪽(자식코드) : M10000.nc파일 ····210
11.2.3. 매크로 내의 매크로 호출 ····210
11.2.4. 외부 매크로 인자전달 예제 ····211
12. 시스템 메모리 ····213
12.1. 시스템 메모리 그룹 ····213
12.2. 좌표계 ····214
12.3. 자동동작 ····214
12.4. 스캔모드 ····214
12.5. 종료동작 ····214
12.6. MDI실행상태 ····214
12.7. 메인쓰레드 실행속도제어 ····215
12.8. FEED / 스핀들 RPM ····215
12.9. 잔량좌표/목표좌표 ····215
12.10. 마스터 인에이블 ····215
12.11. 자동시작라인 ····215
12.12. 커스텀 기본동작 지원 ····216
12.13. CNC 동작 상태 ····217
12.14. 기계 원점 ····218
12.15. 원점복귀 완료상태 ····218
12.16. 현재시간 ····218
12.17. 동작 시간 및 이동거리 관리 ····219
12.18. 스캔타임 ····219
12.19. 동작버전 ····220
12.20. 원점복귀 동작정보 ····220
12.21. 동작카운터 ····220
12.22. 축 엔코더 ····220
12.23. 타이머 플래그 ····221
12.24. 알람 코드 ····222
12.25. 자동프로그램파일정보 ····222
12.26. 조그 동작정보 ····223
12.27. MPG 동작정보 ····224
12.28. JOG 추가 정보 ····225
12.29. 인자 Parameter(X,Y,Z,I,J,K...) ····226
12.30. 패턴출력 인자 ····229
12.31. 조이스틱 정보 ····230
12.32. 전용 IO M코드 실행 ····231
12.33. 전용IO 상태 ····231
12.34. 워크좌표계 원점 ····232
12.35. CNC연산옵션 (공구번호,보정,절대-상대,극,평면,사이클,...) ····233
12.36. 공구테이블 ····234
12.37. 모션보드 메인 상태 ····235

12.38. 모션보드 축별 동작상태 ····236
12.39. PIX48 입출력 ····237
12.40. 모션보드 입출력 ····238
12.41. PC 화면정보 ····239
12.42. 드래그 앤 드롭 파일명정보 ····240
12.43. PC장치 마우스 정보 ····240
12.44. PC장치 키보드 정보 ····241
12.45. 커스텀 UI 인자전달용 ····242
12.46. CNC 설정구성 ····243
12.47. 자동동작 연산참조 ····245
12.48. 공구교환 ····246
12.49. 공구측정 ····247
12.50. MDI 스트링 영역 ····248
12.51. 로그 파일명 ····248
12.52. 시간 스트링 영역 ····248
12.53. 자동작업테이블 영역 ····248
12.54. 3D GRAPH 사용자그리기 속성영역 ····249
12.55. 메인 3D 그래프 기본속성 영역 ····250
12.56. 3D GRAPH 사용자그리기 영역 ····251
13. 사용자 자동화 기능 ····254
13.1. 모션마스터 CNC운영 전체흐름과 관련된 외부스크립터파일 ····255
13.2. 제어입력 혹은 제어명령 변화와 관련된 스크립트파일 요약 ····256
13.3. 화면제어와 관련된 스크립트파일 ····257
13.4. 배경제어 스크립트 파일 (Background.code) ····258

부록I. 지원 NC코드 요약 ····261
A. G코드 ····261
B. M코드 ····265
1. 기본 M코드 ····265
2. 응용 M코드 ····266
부록II. 사이클 동작 ····267
A. 고정사이클 동작 개요 ····267
B. 드릴 사이클(G81) ····268
C. 카운터 보어 사이클 (G82) ····269
D. 심공 드릴 사이클 (G83) ····270
E. 고속 펙킹 드릴 사이클 (G73) ····271
F. 탭핑 사이클 (G84) ····272
G. 역탭핑 사이클 (G74) ····273
H. 보링 사이클 (G85) ····274
I. 드웰이 추가된 보링 사이클 (G89) ····274
J. 보링 사이클 2 (G86) ····275
K. 정밀 보링 사이클(G76) ····276
L. 백보링 사이클(G87) ····277
부록III. MPG 연결 ····278
A. 모션마스터 MPG 동작특징 ····278

B. MPG의 구성요소 ····279
C. MPG Setup 윈도우 ····280
D. MPG 구성요소들의 신호와 연결방식 및 설정 ····282
E. 엔코더 휠 (ENCODER WHEEL) ····282
1. 연결 ····282
2. MPG SETUP 윈도우 설정 ····284
F. 축 선택 및 배율선택 스위치 ····287
1. SELECTOR 스위치 접점방식과 연결 ····287
2. SELECTOR 스위치와 ADC입력의 연결 ····288
3. BCD 스위치 접점방식과 연결 ····289
4. ONE SWITCH 방식의 연결 ····290
5. 축선택(이동배율선택) 설정 ····291
G. 기타 연결 설정 ····293
1. 비상정지(ESTOP) 스위치 연결 및 설정 ····293
2. 동작가능(ENABLE) 스위치 ····293
3. 휠 동작의 축이동 방향반전 옵션 ····294

1. 개요

1.1. 모션마스터 특징

모션마스터(Motion Master)는 위칸모션보드를 이용하여 다양한 모션시스템에 응용할 수 있는 모션운영프로그램입니다. 기본적으로 모션보드를 활용한 CNC 운전기능을 중심으로 구성되어 있습니다.

모션마스터는 기본적인 CNC공작기계의 G코드 운영방식 뿐만 아니라 내장된 실시간 인터프리터를 이용한 언어방식 및 사용자 화면구성 등 다양한 자동화 기능들을 포함합니다.

아래는 기본적인 모션마스터의 기능들에 대한 맵(Map)을 보여줍니다.

순번	기능	설명
1	CNC	기본적으로 밀링형 CNC 동작제어를 지원합니다. 내부적으로 V스크립터 언어기능과 매크로기능에 의한 사용자 G,M코드 확장으로 다양한 머신의 자동화응용에 확장될 수 있습니다.
2	모니터링	모션보드의 동작상태를 표현하고 기초적인 제어를 실행합니다.
3	기계별 구성 설정	하나의 프로그램으로 다양한 구성별 기계를 운영할 수 있습니다.
4	V 스크립터	G코드와 함께 자체 실시간해석언어인 V 스크립터기능을 내장하고 있어 수학함수, 모션제어함수, 파일 및 입출력함수 등 다수의 함수를 지원하여 다양한 응용분에 활용될 수 있습니다.
5	사용자 화면	사용자가 직접 화면을 구성하고 각 화면구성요소들에 대한 동작들을 구현할 수 있는 기능을 제공합니다.
6	사용자 위저드	팝업메뉴나 특정한 목적의 *간단한 자동G코드 생성기와 같은 동작들을 사용자에 의해 추가될 수 있습니다.
7	자동화 프로그램	V스크립터 언어를 통해서 실시간 실행되는 백그라운드 실행코드를 지원함으로써 PLC와 같은 자동화프로그램 역할을 구현할 수 있습니다.
8	매크로 확장	사용자가 작성한 NC코드파일을 매크로테이블에 Gxx, Mxx의 이름으로 등록하여 호출해서 사용할 수 있습니다.
9	PC 입출력 확장	조이스틱, 시리얼포트, 이더넷, 키보드, 마우스 동작을 읽어들여 다양한 모션제어 확장에 활용될 수 있습니다. 그 외 백그라운드 실행코드에 의해 윈도우즈 정규DLL을 실행할 수 있습니다.
10	3D 프린팅	FDM방식의 3D 프린터 G코드를 실행합니다.

1.2. 모션컨트롤러

모션컨트롤러란 좌표계 방식을 이용하여 작업장치의 위치,속도를 수치적으로 제어하는 컨트롤러를 말합니다. 대부분 좌표계는 XYZ의 직교좌표계를 사용합니다.

모션마스터는 위칸 모션컨트롤러와 연결되며 기본적으로 XYZ 직교좌표계(Cartesian Coordinate System)를 기준으로 작업종단장치의 움직임을 제어합니다.

모션컨트롤러는 다양한 종류로 나눌 수 있으며 여기서는 중요한 2가지 관점에서만 분류해 보고 위칸모션컨트롤러가 어느 부류에 속하는지 살펴보겠습니다.

1.2.1. 위치결정기와 CNC 컨트롤러

먼저 모션컨트롤러는 사용목적에 따라 위치결정컨트롤러와 CNC 컨트롤러 나누어 볼 수 있습니다.
* CNC : Computerized Numerical Control (마이크로프로세서를 활용하여 수치적으로 모션을 제어하는 것)

위치결정기는 대부분 미리 프로그램 된 순서대로 지정된 속도로써 목적경로를 움직이는 분야에 사용됩니다.
반면에 CNC컨트롤러는 CNC전용의 코드그룹(G코드,M코드)에 의해서 작성된 일련의 작업지침들을 읽어서 기계시스템이 허용하는 범위 내에서 작업지침에 맞게 움직임을 제어하도록 만들어진 컨트롤러입니다.
서로 비슷한 목적으로 사용될 수 있지만 CNC컨트롤러는 기본적으로 G코드 인터프리터에 의해 다양한 모션동작들이 대부분 표준화 되어 있으며, 모션동작에 따라 자동으로 기계시스템에 적절한 가감속 동작이 이루어지는 점이 다릅니다. CNC전용의 코드그룹들은 많은 부분들이 표준화되어 있어 상호호환성이 좋으며, 추가적으로 기계특성에 맞는 동작을 커스터마이징하는 기능이 갖추고 있어 활용하기 편리합니다.

위치결정기는 PC와 PLC기반으로 프로그램되어 반송기, 디스펜서, PICK & PLACE, 및 솔더링, 비젼 및 검사장비 등 다양한 산업자동화 모션컨트롤 영역에서 많이 사용됩니다.
CNC컨트롤러는 CNC밀링,선반,머시닝센터,자동레이저절단,워터젯,플라즈마키팅,와이어커팅, 자동조각장치 등의 다양한 CNC 가공장치에 많이 사용됩니다. 최근 CNC컨트롤러의 비용이 낮아지고, 환경에 유연하게 적응하기 위해 위치결정기시스템분야에서도 CNC컨트롤러가 사용되는 경우가 늘고 있습니다.

1.2.2. 위치제어형과 속도제어형 모션컨트롤러

대부분의 위치결정기는 펄스구동형 위치제어형 모션컨트롤러를 사용합니다. 펄스구동형이란 각 XYZ축의 모터(액추에이터)들에 대한 움직임을 펄스단위로 제어하는 것을 말합니다. 그러므로 하나의 펄스에 해당하는 최소단위 움직임이 정의되고, 모터를 회전시키는 구동드라이브는 펄스열 입력을 받아 지령된 위치로 이동시켜 줍니다.
동작부하관성이 작거나 소규모 CNC모션컨트롤러에서 펄스구동형의 위치제어형 모션컨트롤러를 사용할 수 있습니다.

속도제어형 모션컨트롤러는 대부분 CNC가공머신에서 절삭부하의 변동에 적응하여 최적의 움직임을 제어하기 위해 모터 움직임을 직접 피드백 받아서 모터의 속도를 조절하여 원하는 경로와 속도를 제어하는 방식의 모션컨트롤러를 말합니다. 속도제어형 모션컨트롤러는 모터의 회전상태를 피드백 받아서 속도지령을 만들기 때문에 서보모터와 서보드라이브를 사용해야 하며, 때로는 모션컨트롤러 자체적으로 모터드라이브를 내장하는 경우가 있습니다.

1.2.3. PC 기반형, PLC 기반형 모션컨트롤러와 독립형 모션컨트롤러

산업자동화에서 모션컨트롤러는 크게 PC기반형 모션컨트롤러와 PLC기반형 모션컨트롤러, 그리고 전용의 독립모션제어기 형태로 사용되어 집니다. 이들은 모션컨트롤러에게 지령을 내리는 상위제어기의 형태에 따라 분류된 것입니

다.

* PLC : Programmable Logic Controller (프로그램가능한 로직제어기로써 산업자동화분야에서 PC를 대신해 보다 안정적이고 간편하며 단순한 프로그램 기능을 갖춘 컨트롤러)

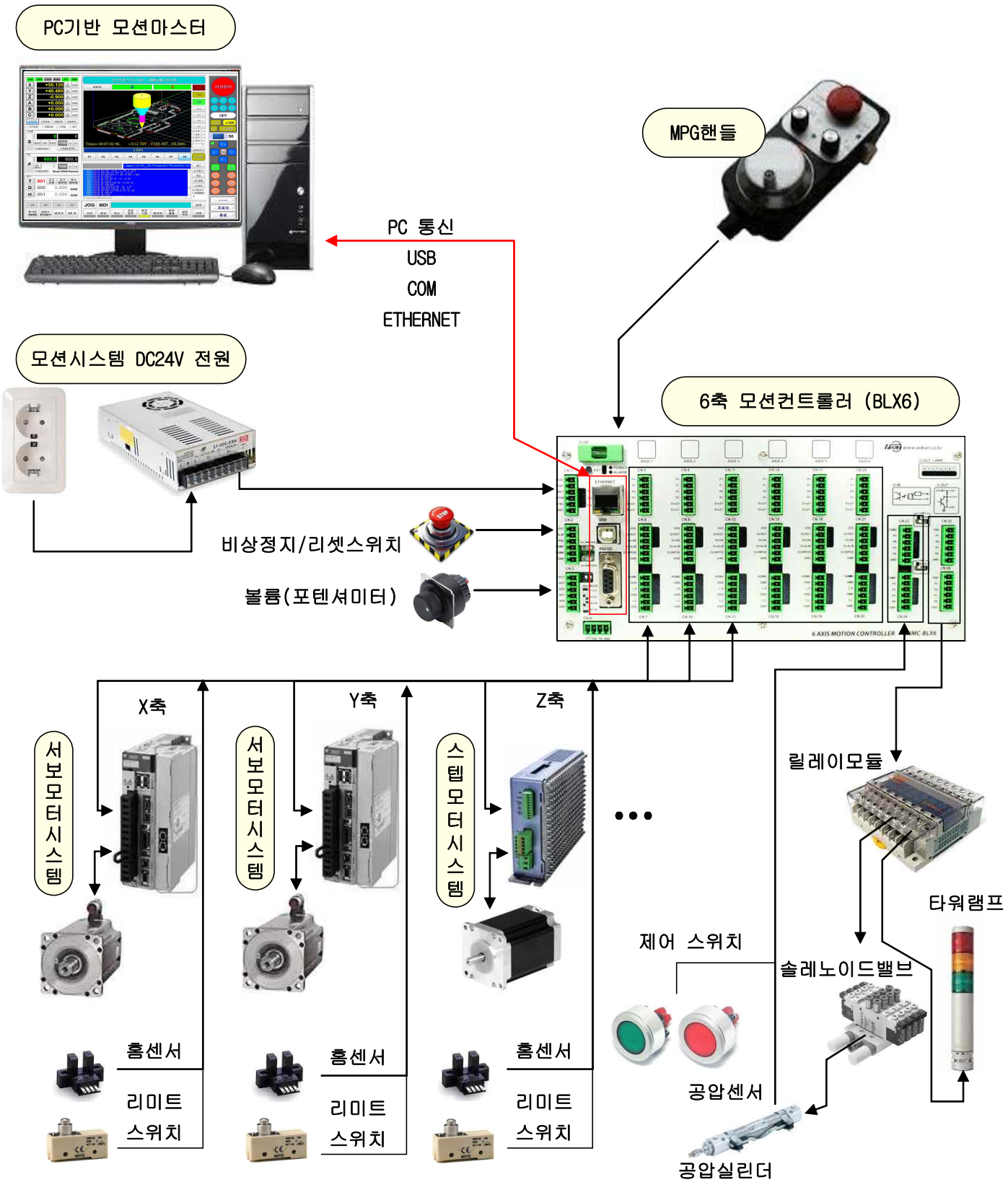
PLC 기반형은 대부분 위치결정기 시스템에서 사용되며, CNC모션컨트롤 시스템에서는 사용되지 않습니다.
특별히 한정된 동작의 위치결정시스템에서만 PLC기반이 사용되며, 보다 복잡하고 빠르며 지능적인 동작을 필요로 하는 자동화장비(비젼 및 PICK & PLACE(SMT장비) 등)에서는 PC 기반형이 많이 사용되어 집니다.

반면에 CNC 컨트롤러의 경우 PLC 기반형은 사용되지 않으며 대체로 PC기반형과 독립형(스탠드얼론) 형태로 나누어 집니다. (CNC 시스템에서는 CNC 컨트롤러 규모가 PLC시스템규모보다 커서 CNC 컨트롤러내부에 PLC시스템이 내장된 경우가 많습니다.)
독립형은 산업용PC 혹은 전용의 마이크로프로세서모듈로써 CNC기능을 구현하여 안정성이 높습니다.
PC 기반형은 범용의 PC시스템을 그대로 이용하여 CNC기능을 구축하는 모션컨트롤러형태를 말합니다.

X6V3 모션엔진을 장착한 위칸 모션컨트롤러는 펄스구동방식의 위치제어형 모션컨트롤러 이며, 모션마스터를 이용하여 응용될 경우 PC 기반형의 위치결정 및 CNC컨트롤러 전체에 응용될 수 있습니다.

1.3. 위칸 모션컨트롤러와 모션시스템의 연결 구성 예

아래에는 BLX6의 위칸모션보드를 이용한 모션시스템의 다양한 연결 예를 보여줍니다.
(아래그림은 직접적인 연결이 아닌 개념적인 기기연결을 보여줍니다.)



2. 프로그램 설치/제거 및 시작하기

2.1절 과 2.2절는 각가 USB드라이브와 모션마스터 프로그램설치/제거편으로 초기설치시만 참조하세요.

2.1. USB 드라이버 설치

위칸모션컨트롤러의 USB연결방식을 사용할 경우 전용의 USB드라이버를 설치해야 합니다.
USB드라이브는 모션 마스터프로그램의 설치전, 후에 관계없이 모션마스터와 모션컨트롤보드가 연결되기 전에 먼저 설치해 주시면 됩니다. 이더넷연결이나 기본적인 COM연결만을 사용할 경우 USB드라이브 연결을 필요로 하지 않습니다.

2.1.1. USB로 장치 연결시 주의사항

USB로 연결되어 사용할 경우 아래의 유의사항을 꼭 참고하세요.

USB로 연결되는 기계시스템은 노이즈발생이 작은 소규모의 스텝모터등의 액추에이터를 사용하는 모션시스템에서 안전합니다. USB통신신호는 통상 5V이하의 신호를 높은 주파수의 통신대역을 사용하기 때문 노이즈에 취약합니다.
위칸모션컨트롤러는 내부적으로 USB, SERIAL, EHTERNET 모두 RS232C 프로토콜로 사용됩니다. 그러므로 설치되는 USB드라이브 역시 USB를 COM으로 전환시켜주는 가상 COM(시리얼)통신 드라이브 입니다.

기계시스템과 PC시스템의 전원을 공유할 경우 모터의 동작이나 서보의 ON/OFF상태에 따라 발생된 노이즈로 인해 USB포트가 끊어질 수 있습니다. 이때 가상시리얼포트로 맵핑된 USB가 사라짐으로 인해서 모션마스터 프로그램이 정상적으로 작동되지 않을 수 있습니다.

부득이 USB시스템으로 기계시스템과 연결할 경우 노이즈발생을 조금이라도 억제시키고자 한다면 PC전원을 기계시스템전원과 분리시키며, 모터구동회로와 같이 큰 노이즈를 발생시키는 원인이 될 수 있는 전원단에는 반드시 적절한 용량의 노이즈필터를 장착하는 것이 도움이 됩니다.

노이즈가 지속적으로 발생된다면 시리얼통신이나 이더넷통신을 통해서 모션보드와 연결하는 것이 안정된 통신방법일 수 있습니다.

2.1.2. 설치 환경

- 모션보드의 USB 드라이버는 Windows XP/Server 2003/Vista/7/8 버전에서 사용할 수 있습니다.
- 사용자의 Windows 버전 및 환경 설정에 따라서 설치 화면이 본 설명서와 상이 할 수 있습니다.

2.1.3. 드라이버 설치 시 주의사항

- 드라이버 설치가 정상적으로 종료되기 전까지는 모션보드를 컴퓨터에 연결하지 마십시오.
- 모션보드 연결 시 USB 허브(HUB)를 사용할 경우 USB 전원이 부족할 수 있습니다. 다른 USB 장치와 같은 허브에 연결하지 마십시오.
- USB 허브(HUB)를 사용할 경우 허브(HUB)에 따라 설치 및 연결상태가 나빠질 수 있습니다. 설치 및 연결상태가 바르지 않을 경우 USB 허브를 사용하지 않고 PC와 직접 연결하여 사용해 주십시오.
- 모션보드를 컴퓨터에 연결할 때에는 USB 케이블의 제품 종류에 따라 연결품질이 나빠질 수 있습니다. 가능하면 제품에 포함된 USB케이블을 사용해주십시오.

2.1.4. USB 드라이버 파일

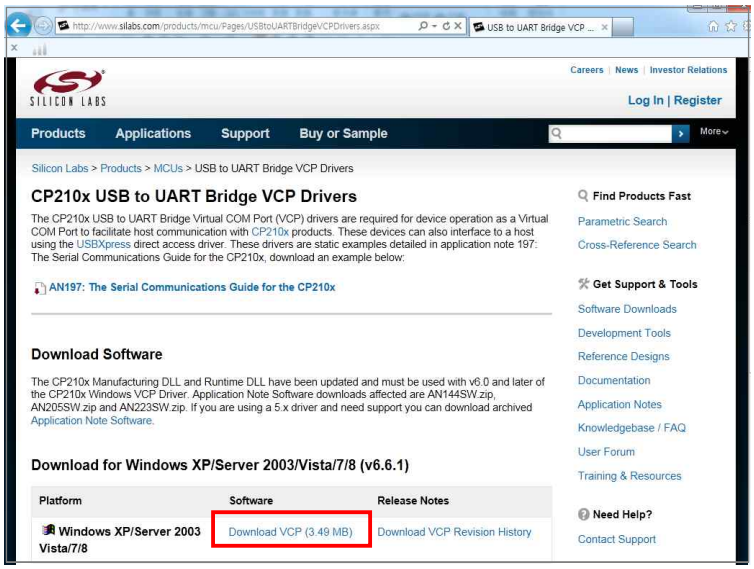
FileName	지원 OS	드라이버 버전
CP210x_VCP_Windows.zip	Windows XP/Server 2003/Vista/7/8	v6.6.1

- USB 드라이버 설치 파일은 홈페이지 자료실(http://www.wikan.co.kr/제품지원/자료실)에서 받을 수 있습니다.
- 최신 USB 드라이버 파일은 아래 그림처럼

http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx

에서 다운로드 받을 수 있습니다.

"VCP Driver Kit" 링크를 클릭하여 드라이버를 다운로드 받습니다.
(다운로드 받은 파일이 압축파일(.zip)로 제공된 경우 압축을 해제 후 사용하십시오.)



2.1.5. 설치

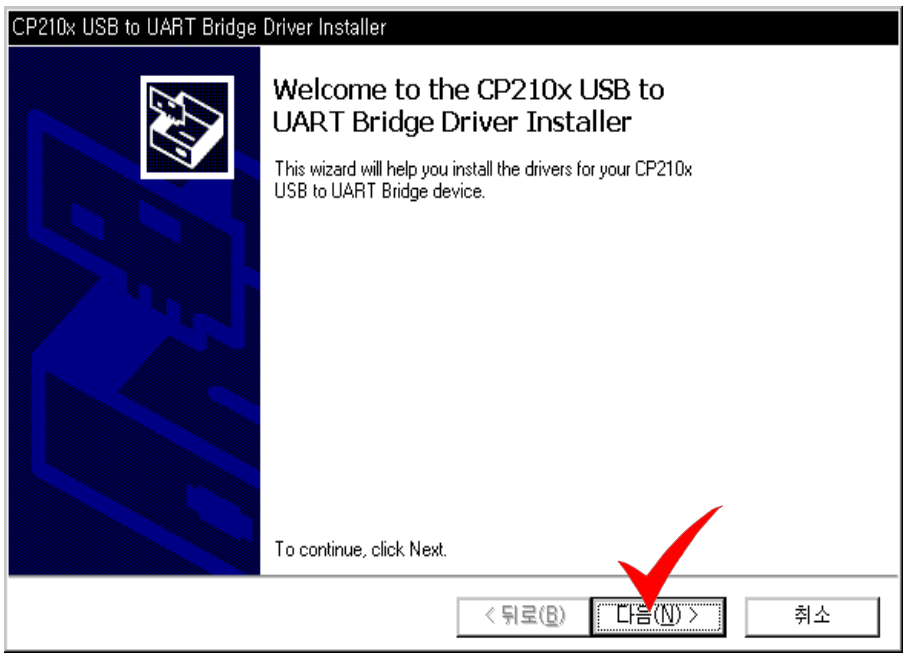
① 설치 파일 “CP210x_VCP_Windows.zip”을 PC의 임의의 폴더에 저장합니다.
② 설치 파일의 압축을 해제합니다. 압축을 해제하면 다음 그림처럼 드라이버 설치 파일이 생성됩니다.

<드라이버 설치 파일>

③ 설치 프로그램 실행
- 32 비트 시스템일 경우 드라이버 설치 파일 중 “CP210xVCPInstaller_x86.exe” 파일을 실행합니다.
- 62 비트 시스템일 경우 드라이버 설치 파일 중 “CP210xVCPInstaller_x64.exe” 파일을 실행합니다.

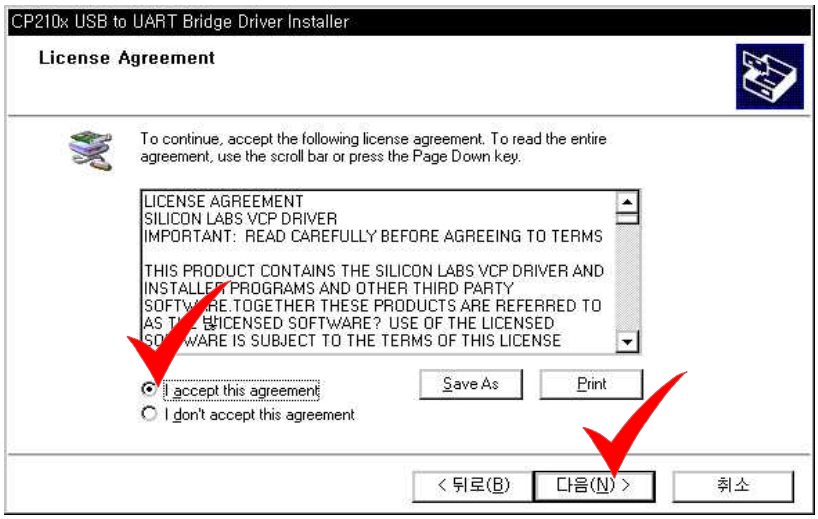
④ 설치과정
- 설치 과정은 32비트 시스템과 64비트 시스템 모두 동일합니다.

<설치화면 -STEP 1>



설치 진행을 위해 “다음” 버튼을 누릅니다.

<설치화면 -STEP 2>

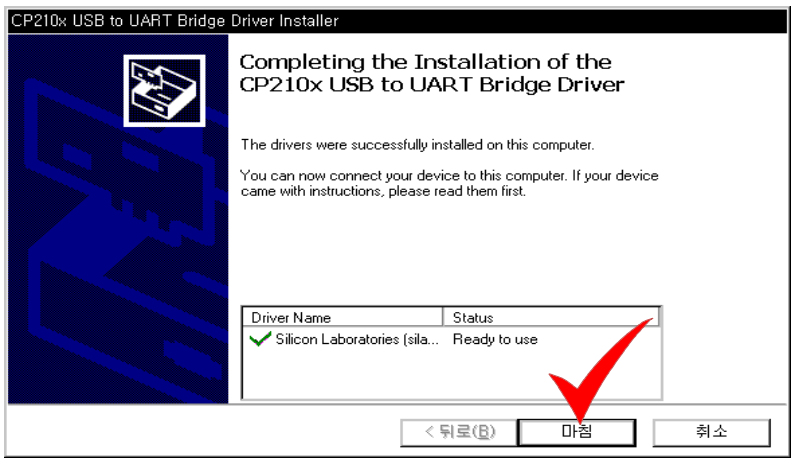


USB 드라이버를 사용하려면 최종 사용자 USB 드라이버 사용권 계약의 내용을 읽고 동의해야 합니다.
USB 드라이버 사용권 계약 내용에 동의하지 않는 경우 USB 드라이버 설치를 진행 할 수 없습니다.
사용권 계약 내용을 자세히 읽어본 후 내용에 동의하면 “I accept this agreement” 버튼을 클릭합니다.
설치 진행을 위해 “다음” 버튼을 클릭합니다.

<설치화면 -STEP 3>

USB 드라이버 설치가 진행 됩니다. 설치되는 동안 기다려 주십시오.

<설치화면 -STEP 4 : 설치종료>



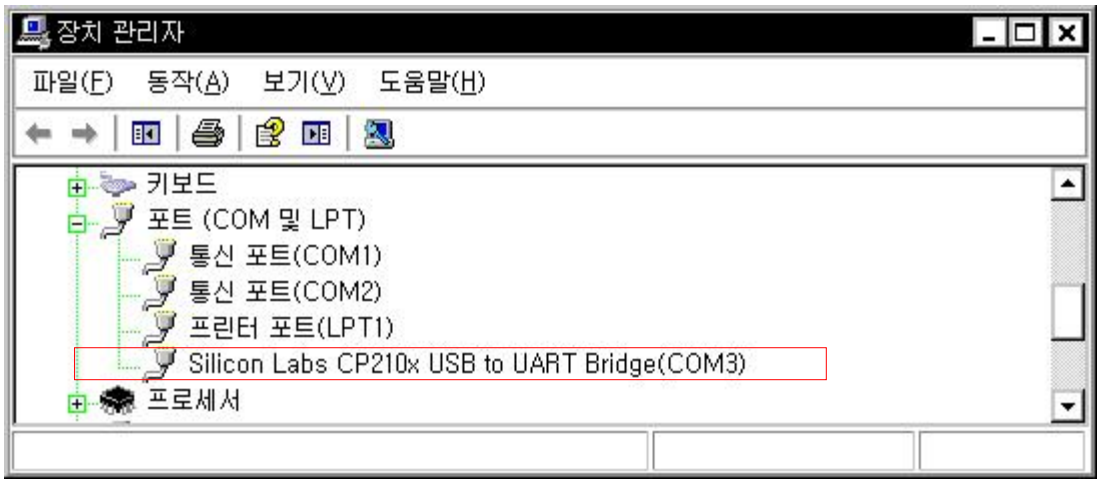
드라이버 설치가 정상적으로 완료되었습니다. “마침” 버튼을 클릭하여 설치를 종료합니다.

2.1.6. 드라이버 설치 확인

- 드라이버가 성공적으로 설치되었다면 아래의 순서를 통해서 정상적인 연결을 확인할 수 있습니다.

① 모션보드를 컴퓨터와 연결합니다.
② 컴퓨터의 장치관리자>포트" 항목에서 "Silicon Labs CP210x ..."을 통해 정상적인 드라이버설치와 모션보드연결을 확인할 수 있습니다.

< USB 드라이버 인식 성공>



(COM 포트 번호는 PC 환경마다 다르게 지정될 수 있습니다.)

③ 드라이버가 설치가 잘못 되었을 경우 아래 그림처럼 "장치관리자>포트"에서 확인 할 수 있습니다.
< USB 드라이버 인식 실패 >

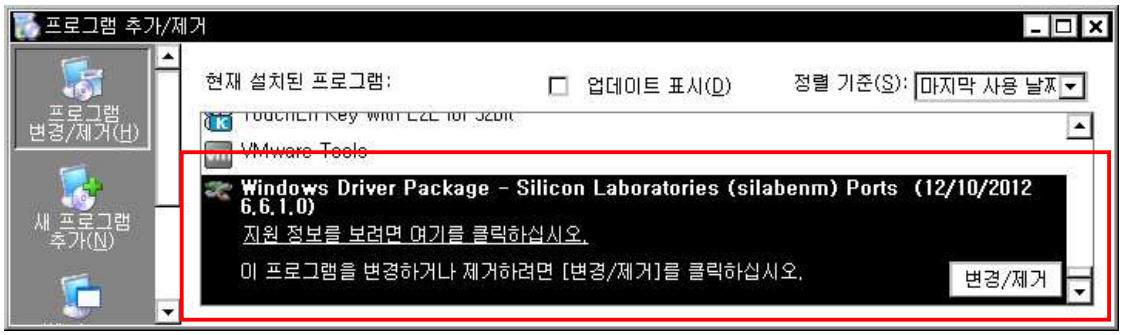
USB 드라이버 인식에 실패 했을 경우 아래의 순서로 다시 실행합니다.

- 모션 보드 연결을 분리하십시오.
- USB 드라이버를 제거 하십시오.
- USB 드라이버를 다시 설치하십시오.

2.1.7. 드라이버 제거

① 제어판->프로그램 추가/삭제를 실행합니다.
② 아래 그림과 같이 “프로그램 추가/삭제” 리스트에서 다음 항목을 선택하여 드라이버를 삭제해 주십시오.
- “Windows Driver Package - Silicon Laboratories Ports”

< USB 드라이버 프로그램 삭제>



위 그림과 같이 “Windows Driver Package - Silicon Laboratories Ports”항목을 선택하고 “변경/제거”버튼을 클릭합니다.

“예” 버튼을 클릭하여 드라이버를 제거합니다.

2.2. 프로그램 설치/제거

2.2.1. 프로그램 설치 환경

- Motion Master 프로그램의 버전에 따라 설치 방법이 본 설명서와 상이 할 수 있습니다.
- Motion Master 프로그램은 Windows XP/Server 2003/Vista/7/8 버전에서 사용할 수 있습니다.
- 사용자의 Windows 버전 및 환경 설정에 따라서 설치 화면이 본 설명서와 상이 할 수 있습니다.

2.2.2. 프로그램 설치 파일

- Motion Master 설치 파일은 홈페이지 자료실(http://www.wikan.co.kr/제품지원/자료실)에서 받을 수 있습니다.

FileName	지원 OS	프로그램 버전
MotionMaster_Installer_V01_R13.exe	Windows XP/Server 2003/Vista/7/8	v01.00

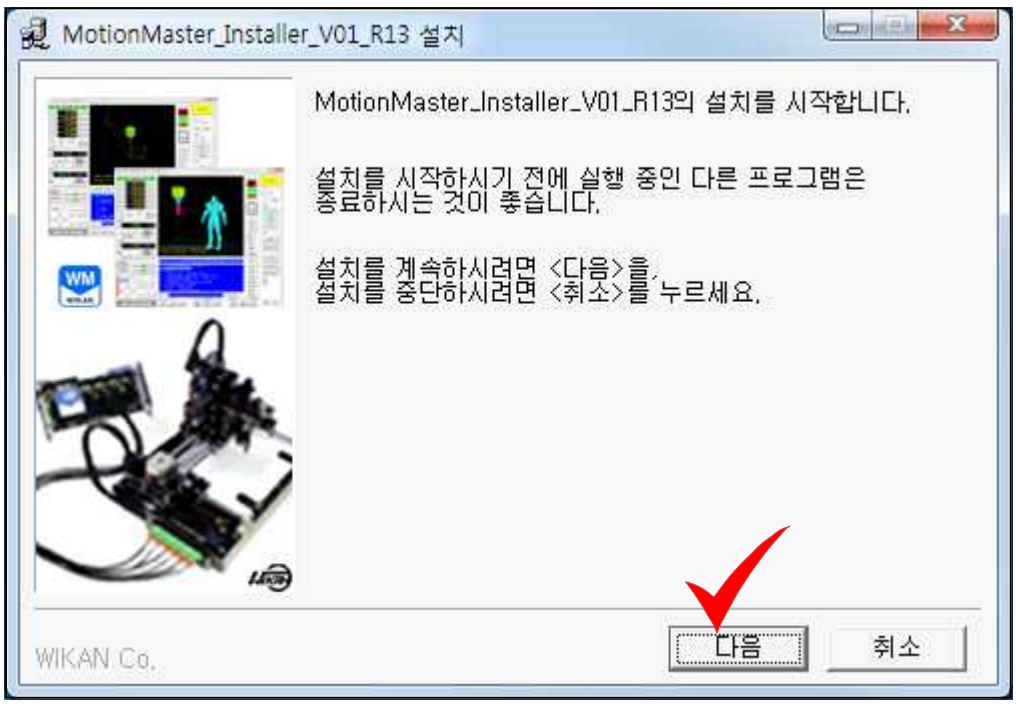
2.2.3. 프로그램 설치

① 설치 프로그램 파일 “MotionMaster_Installer_V01_R13.exe”을 PC의 임의의 폴더에 저장합니다.

② 설치 과정

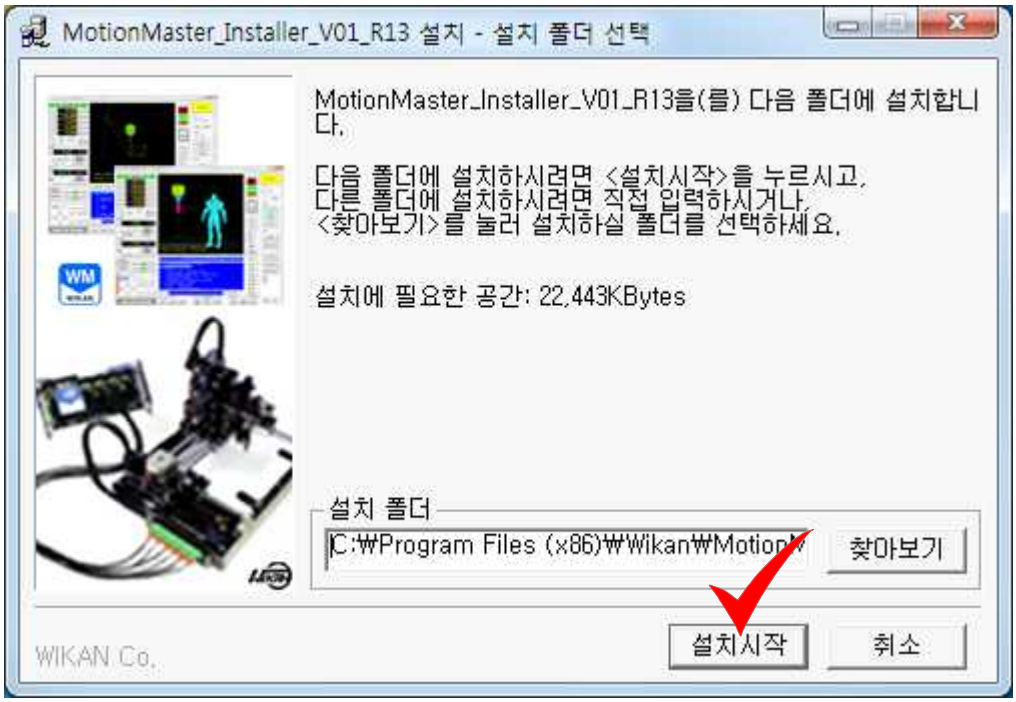
설치 프로그램을 실행합니다.

<설치화면 -STEP 1>



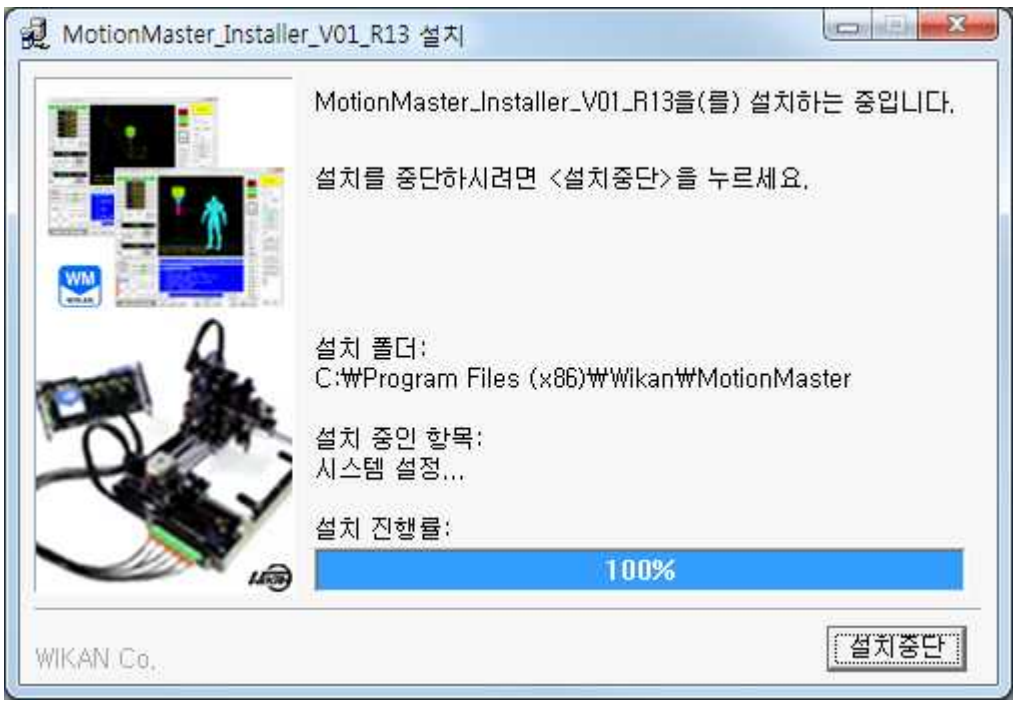
설치 창이 나타나면 설치 진행을 위해 “다음” 버튼을 누릅니다.

<설치화면 -STEP 2>



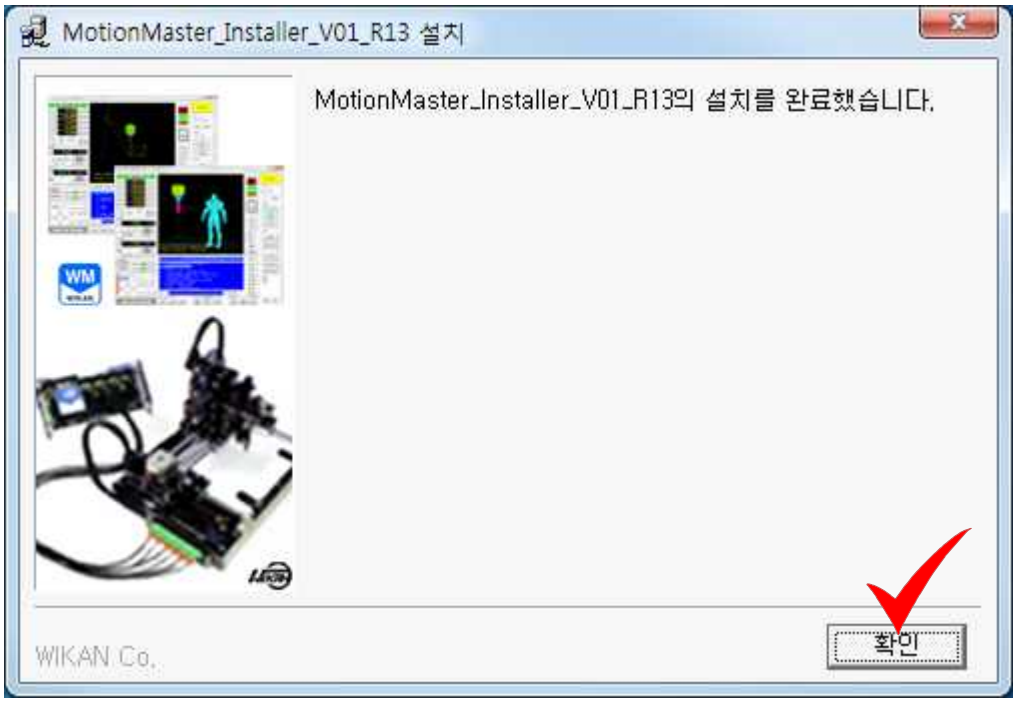
설치 경로를 확인합니다.
기본 설치 폴더는 “c:₩Program Files₩Wikan₩MotionMaster” 입니다.
설치에 필요한 공간은 프로그램 버전 또는 설치 환경에 따라 달라질 수 있습니다.
설치 진행을 위해 “설치 시작” 버튼을 누릅니다.

<설치화면 -STEP 3>



설치가 진행 됩니다. 설치 진행률이 100%가 될 때까지 잠시 기다립니다.
설치를 중단하고 싶을 경우 “설치 중단” 버튼을 누릅니다.

<설치화면 -STEP 4>



설치가 완료 되었습니다.
“확인” 버튼을 누릅니다.

<설치화면 -STEP 5 : 설치 종료>
바탕화면에 Motion Master 바로가기 아이콘이 생성됩니다.
프로그램 그룹에 WikanWMotion Master가 생성됩니다.
설치가 완료된 후 다음과 같이 프로그램 그룹 폴더가 열립니다.

- 프로그램 메뉴 설명

프로그램 메뉴	설명
Uninstall Motion Master V01_R13	Motion Master 프로그램을 삭제 합니다.(“프로그램 삭제” 참고)
Motion Master V01_R13	Motion Master 프로그램을 실행 합니다.
Motion Master V01 사용자 매뉴얼-R01.pdf	Motion Master 매뉴얼 파일을 엽니다. PDF 뷰어가 있어야 열 수 있습니다.

2.2.4. 프로그램 실행

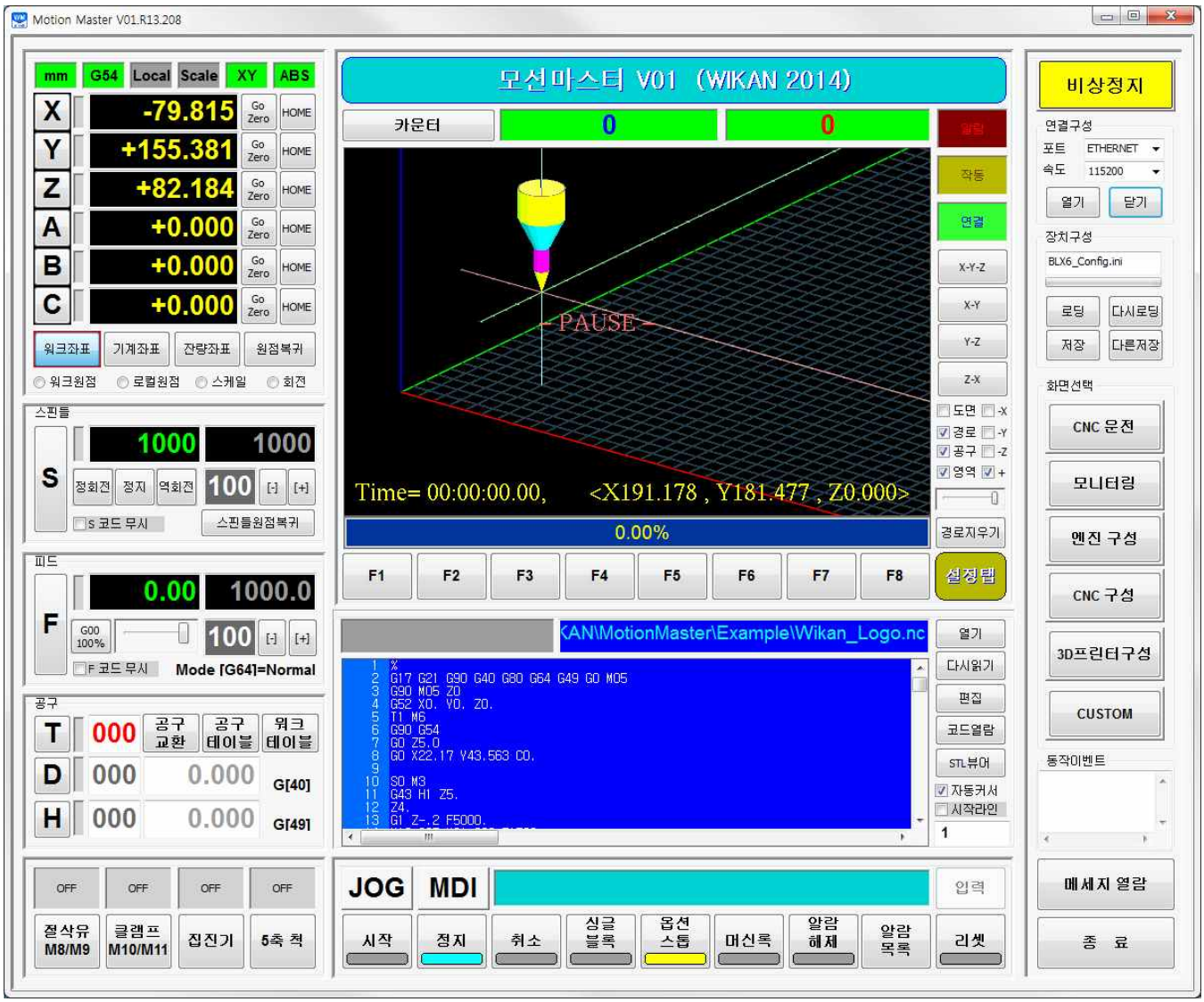
① 프로그램 실행

< 프로그램 그룹 선택 실행 >

< 바탕화면 아이콘 선택 실행 >

< 프로그램 실행 화면 >

정상적으로 설치되면 프로그램 실행 시 다음 그림과 같은 화면이 나타납니다.



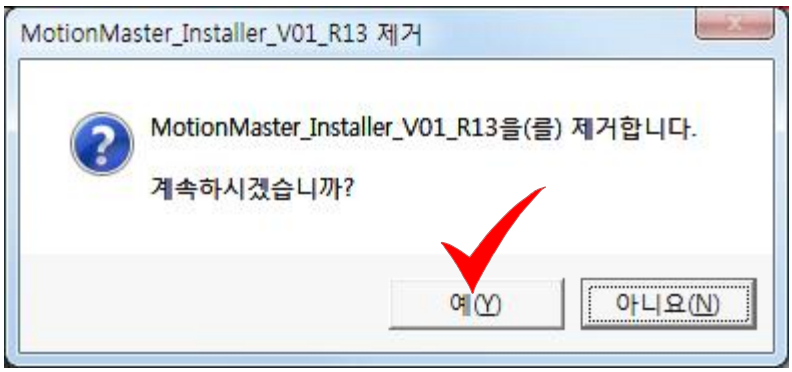
2.2.5. 프로그램 제거

① 제어판->프로그램 추가/삭제를 실행합니다.

② 아래 그림과 같이 “프로그램 추가/삭제” 리스트에서 다음 항목을 선택하여 프로그램을 삭제해 주십시오.

- “Motion Master V01_R13”

③ 위 그림과 같이 “Motion Master V01_R13” 항목을 선택하고 “변경/제거” 버튼을 클릭합니다.



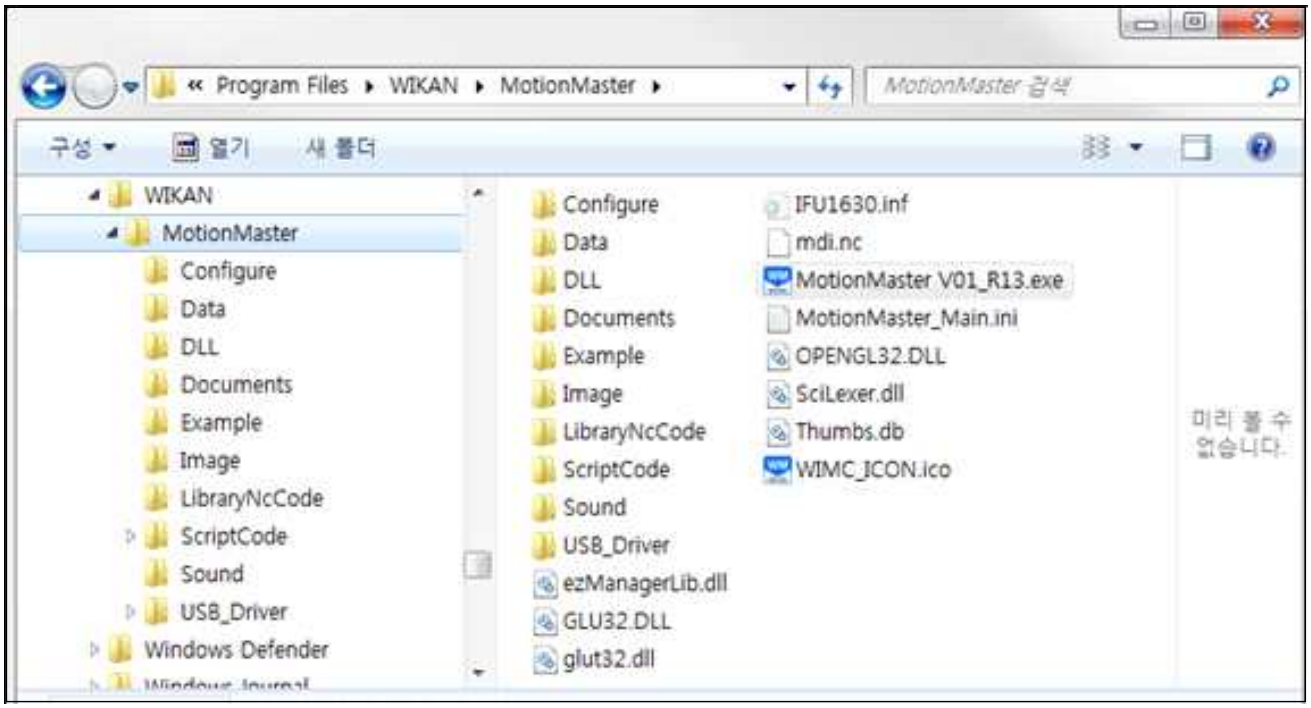
“예” 버튼을 클릭하여 프로그램을 제거합니다.

④ 제거가 완료되었습니다.

2.2.6. 설치파일 구성

기본 설치 폴더는 “c:₩Program Files₩WikanWMotionMaster” 입니다. 윈도우즈 64비트 시스템에서는 “c:₩Program Files (x86)₩WikanWMotionMaster” 입니다. 아래 표는 설치 파일들의 구성을 보여줍니다.

<그림>



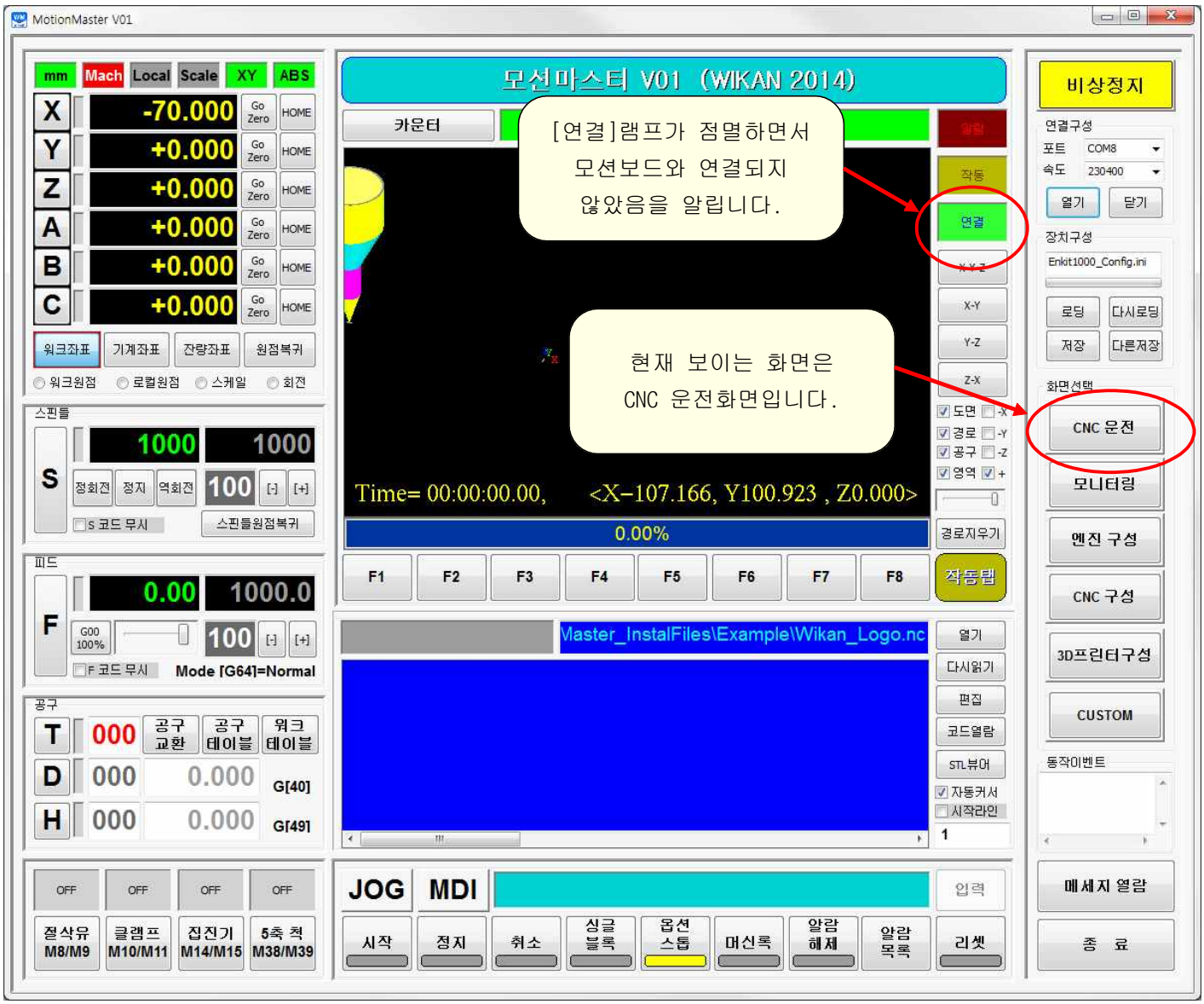
<표>

하위폴더	주요 파일들	파일 내용
설치폴더	MotionMaster V01_R13.exe	Motion Master 실행파일.
	MotionMaster_Main.ini	Motion Master프로그램이 초기에 시작될 때 읽어들여 보드의 구성파일과 연결방식 및 시작화면설정 등을 기록하는 파일.
	WINC_NCCODE.txt	NC Code 요약설명 TEXT파일.
Configure	X6V3_Config.ini	모션엔진 설정이 담긴 구성파일.
	Enkit1000_Config.ini	Enkit-1000 제품의 설정이 담긴 구성파일.
	BLX36_Config.ini	BLX36보드의 초기화 구성파일.
Documents	MOTION_MASTER V01 사용자 매뉴얼-R01.pdf	Motion Master 사용자 매뉴얼파일.
	WINC_NCCODE.txt	NC Code 요약설명 TEXT파일.
Example	*.nc 파일	샘플 NC 코드 파일.
Data	WINC_Adcx.dat	데이터변환관리를 위한 테이블 파일.
DLL	*.DLL	dll 파일.
Image	*.bmp, *.gif, *.jpg, *.png	이미지파일.
Sound	*.wav	웨이브(음성)파일
USB_Driver	CP210x_VCP_Windows	모션컨트롤러 USB드라이브 파일.
LibraryNcCode	*.nc	매크로확장 G,M코드 파일
ScriptCode	*.code	V스크립터 지원파일

2.3. 프로그램 시작하기

모션컨틀보드와 연결되기 전에 먼저 간단하게 전체화면 구성을 살펴봅니다.
모션마스터 프로그램을 실행하면 아래와 같은 로딩화면이 나오면서,

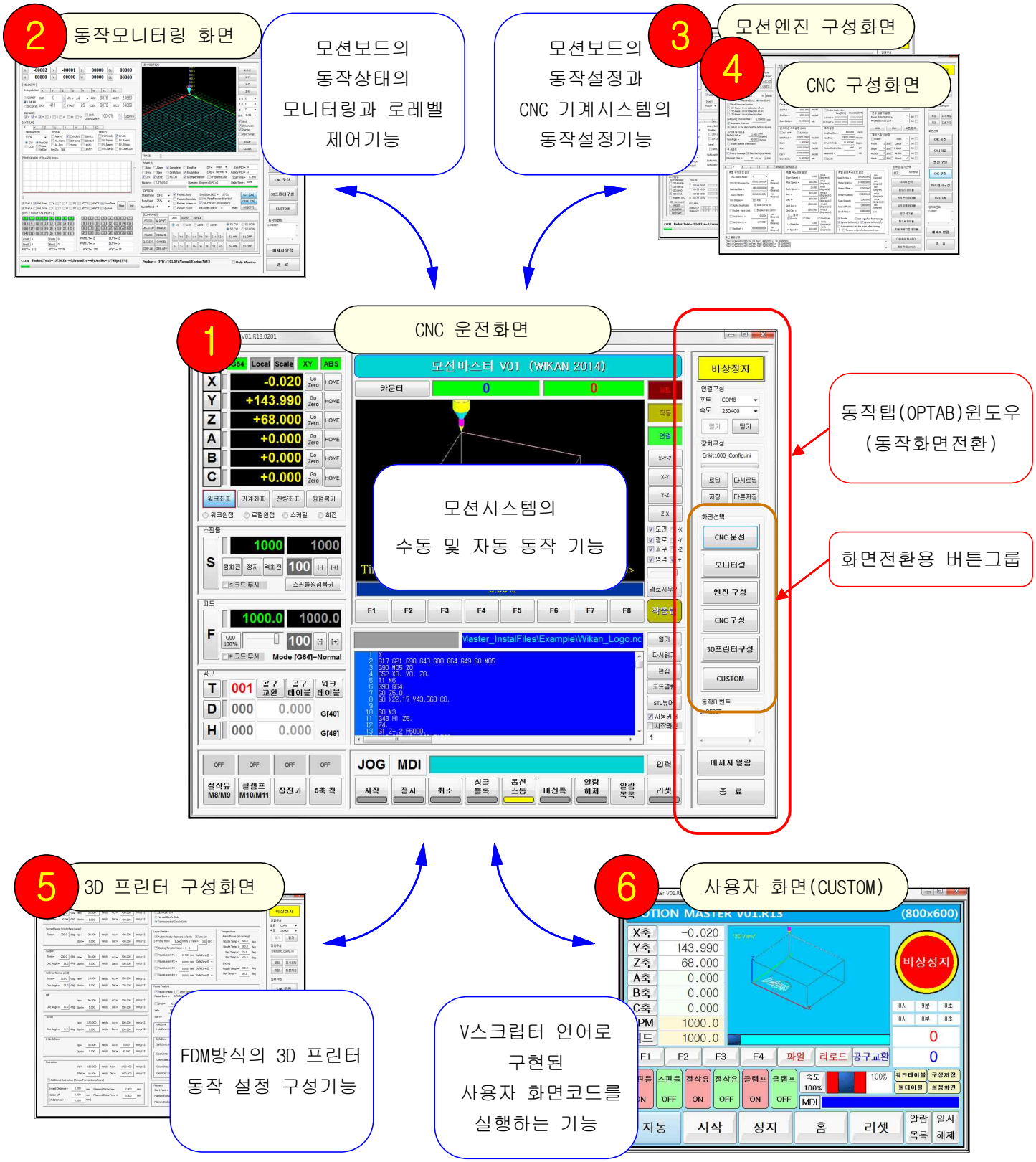
몇 초후 연결이 없으면 다음의 CNC운전화면이 나타납니다.
이때 모션보드와 연결이 되지 않았으므로 아래의 [연결]램프가 점멸하는 것을 확인할 수 있습니다.



2.3.1. 전체 프로그램 훑터보기

모션마스터는 기본적으로 CNC운전화면을 중심으로 각각 기능별 화면으로 전환될 수 있습니다.
오른쪽 길쭉한 윈도우가 기본적인 연결 및 장치구성을 열고, 동작화면을 전환시키는 동작탭 윈도우 입니다.
대부분은 CNC운전화면에서 사용될 것이며, 특별히 기계시스템의 초기 셋업과정에서 설정화면과 모니터링 화면이 사용될 것입니다. 아래에 조작화면들의 구성과 기능들을 간략한 그림으로 설명합니다.

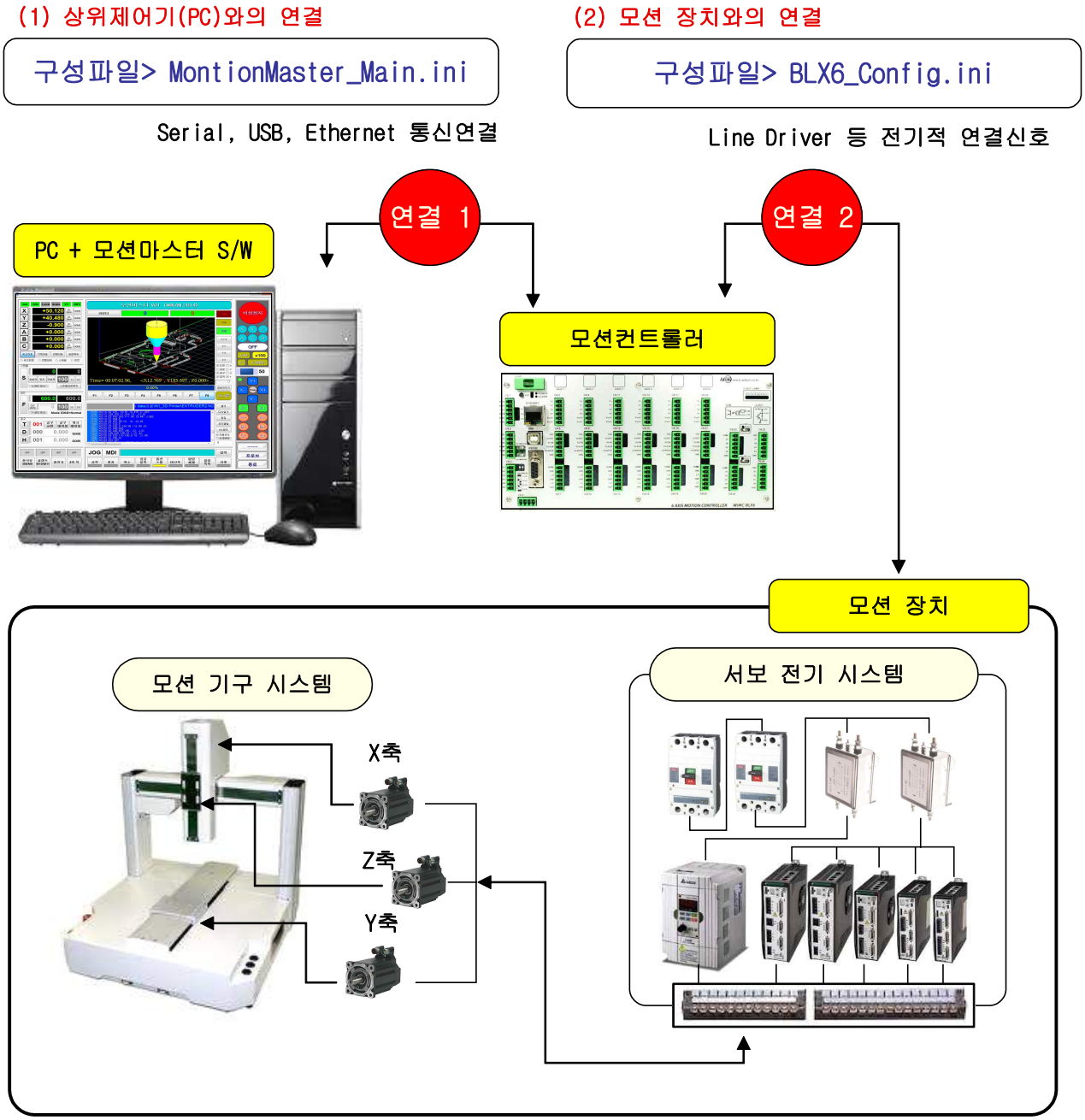
<그림>



2.3.2. 모션컨트롤러의 연결과 구성파일 관리

아래의 그림은 모션컨트롤러 관점에서 바라본 2가지의 방향의 연결을 보여줍니다. 하나는 PC와 같은 상위제어기(모션마스터)와의 연결이며 다른 하나는 모션장치와의 연결입니다.

모션 마스터프로그램은 그림과 같이 모션컨트롤러 입장에서 연결 설정 될 수 있는 2가지 구성설정파일로 나누어서 관리하게 됩니다.

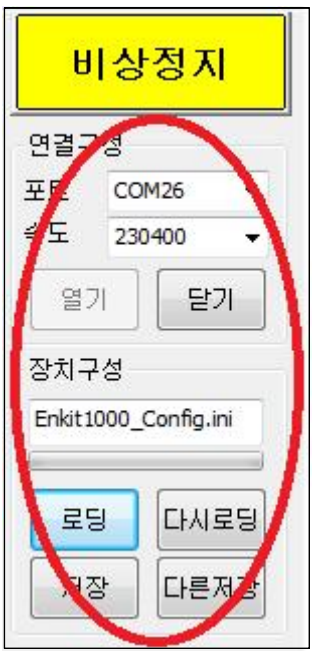


MotionMaster_Main.ini 구성파일은 모션마스터의 실행환경(PC성능,화면설정등)과 모션컨트롤러의 연결방식 및 초기로딩 할 기계구성파일의 종류 등을 기술한 파일이며, BLX6_Config.ini파일은 기계의 특성마다 사용자가 구성을 변경하여 임의적으로 설정하는 파일입니다. 일반적으로 BLX6_Config.ini 장치구성파일은 설치폴더의 하위폴더인 Config폴더에서 지정되지만 사용자가 임의로 지정할 수 있습니다.

2.3.3. 프로그램 실행 시작하기

처음 프로그램을 실행시키면 아래의 연결램프가 점멸(깜박)하면서 프로그램이 모션보드와 연결되지 않았음을 표시합니다. 아래의 순서대로 실행하여 프로그램과 모션보드가 서로 통신하면서 연결되면 깜박임이 멈추고 점등하게 됩니다.

먼저 모션컨트롤러와의 통신을 위해서 아래의 동작탭 윈도우의 [연결구성]과 [장치구성]항목을 연결순서에 맞게 설정해 줍니다.



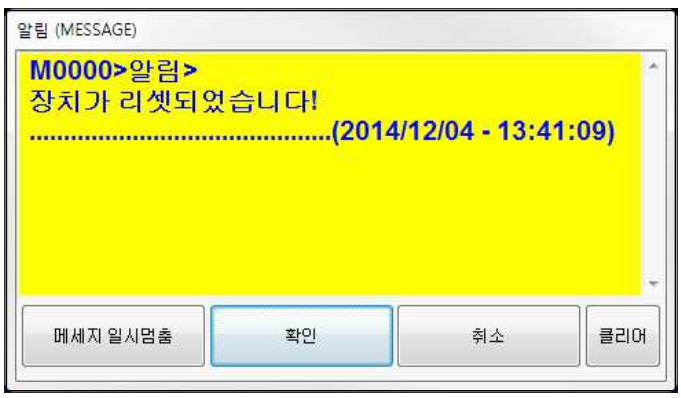
<연결순서>

① 가장먼저 연결포트를 선택합니다. 모션보드는 시리얼통신(RS232C통신), USB 혹은 ETHERNET으로 연결될 수 있습니다. 연결에 맞는 PC의 통신포트를 선택해 주세요. USB로 연결하더라도 가상시리얼통신포트로 변환되므로, 해당 USB포트와 링크된 가상시리얼포트 번호를 선택하셔야 합니다.

② 연결포트를 선택하셨다면 이제 연결속도를 선택합니다. 보드마다 연결속도를 선택하는 방법이 다를 수 있으니 해당 제품의 통신속도 설정편 참조해주세요.
대부분의 모션보드는 딥스위치로 설정될 수 있으며, 일부는 고정된 통신속도로 지정될 수 있으므로 해당 모션보드의 매뉴얼을 참조해 주세요..

③ 다음으로 장치구성파일을 선택합니다. 현재 장치구성파일의 선택이 바르지 않다면 [로딩]버튼을 눌러줍니다. 그러면 구성파일을 선택할 수 있는 파일선택창이 나타납니다. 대부분 구성파일은 설치폴더의 Config하위폴더에 저장되어 있습니다. 해당 모션보드와 기계장치에 적절한 구성파일을 선택합니다.(여기에서는 Enkit1000_Config.ini구성으로 예시되었습니다)

④연결포트와 연결속도 및 창치구성파일이 바르게 설정되었다면 [열기]버튼을 눌러주어 서로 통신을 시작하도록 합니다. 통신이 시작되면 [연결]램프가 점멸을 중지하고 연속적으로 점등하게 됩니다.
연결이 된 후 “Enkit1000_Config.ini”라는 구성파일을 찾아서 초기화 과정이 진행됩니다. 이 과정은 CNC구성설정에 맞도록 모션보드를 초기화 하며 CNC동작을 위한 프로그램 초기화 과정이 이루어지는 단계입니다. 이때 수초에서 수십초의 시간이 소요될 수 있습니다. (BLX6모션보드를 구매하셨다면 BLX6_Config.ini파일로 장치구성을 지정합니다.)
초기화과정이 완료되면 아래와 같은 메시지화면이 출력하게 됩니다.

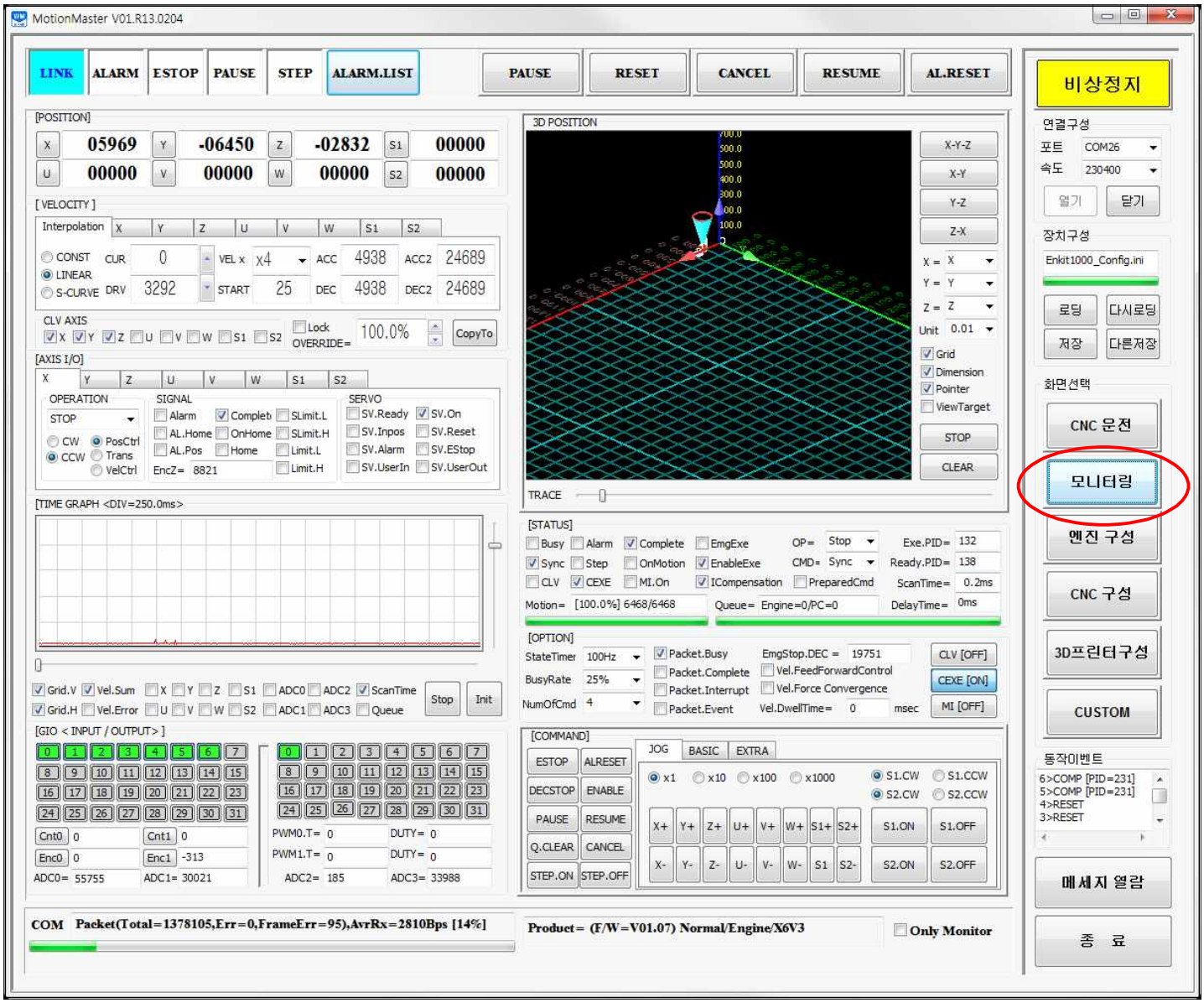


※ 최종 선택된 연결포트, 연결속도 및 구성파일이 기억됨으로 다음번 프로그램 실행시 자동으로 해당포트와 연결속도 및 구성파일이 자동으로 선택되어 집니다.

2.3.4. 모니터링 화면에서 간단하게 움직여 보기

모션보드가 초기화되었다면 현재설정된 속도, 가속도 등의 상태로써 모니터링 조작화면을 통해 간단하게 움직임을 조작할 수 있습니다.

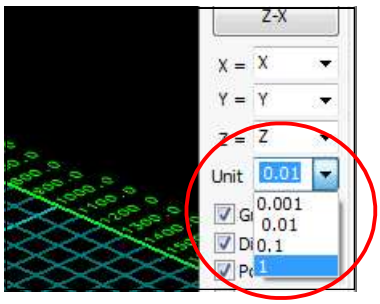
먼저 동작탭윈도우의 [모니터링]버튼을 선택합니다. 그러면 아래와 같은 모니터링 조작화면이 나타납니다.



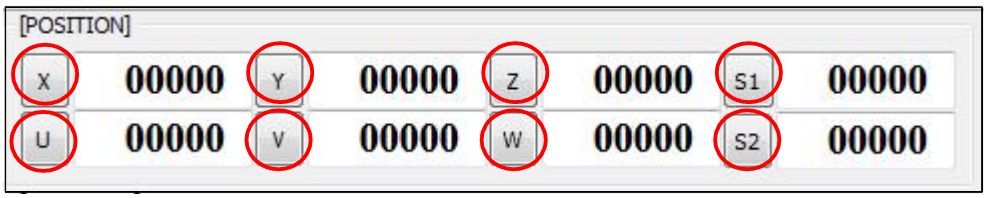
모니터링조작화면은 모션컨트롤러의 아주 기본적인 동작상태를 설정 및 확인할 수 있는 화면입니다. 여기서 동작되는 움직임의 단위는 펄스단위입니다. 그러므로 여기서 설정되는 속도, 가속도는 PPS(Pulse/sec)단위를 기준으로 하며, 모션보드의 다양한 입출력상태를 확인할 수 있습니다.

다음의 순번대로 조작하여 JOG 버튼을 클릭하여 간단하게 작동시켜 봅니다.
(여기서는 CNC의 조그기능이 아닌 모니터링조작화면에서의 펄스단위 조그동작기능을 이용합니다.)

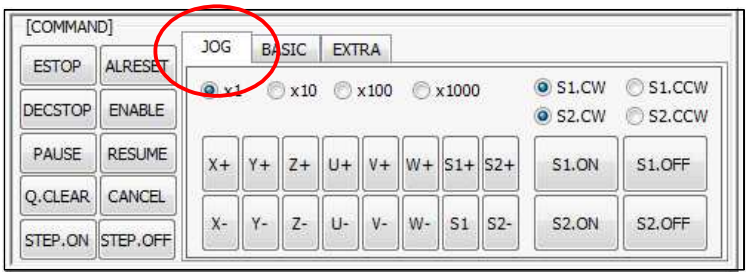
① 먼저 3D 그래프의 움직임좌표단위(펄스단위)를 0.01에서 1 로 바꾸어 줍니다.



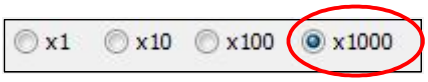
② 현재 좌표 X,Y,Z,U,V,W,S1,S2 값을 모두 0의 값으로 바꾸어 줍니다.
위치제어부의 [X], [Y]...버튼을 눌러주면 0으로 초기화 됩니다.



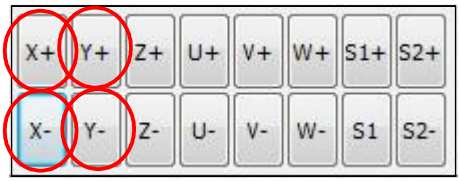
③ [명령화면]의 [JOG]탭을 선택합니다.



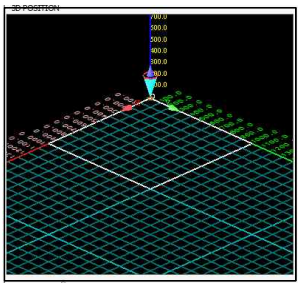
④ 조그동작의 다양한 버튼이 나타납니다. 조그동작은 축을 개별적으로 움직여 볼 수 있는 기능입니다.
축별 버튼을 눌러주면 해당방향으로 이동합니다. 이때 한번 눌러 줄 때마다 몇 펄스씩 움직일지를 아래의 선택버튼으로 설정해 줍니다. x1000 을 선택합니다.



⑤ [X+], [Y+], [X-], [Y-]버튼을 순서대로 눌러줍니다.



⑥ 좌표값의 변화와 그래프상의 이동을 확인할 수 있습니다.그래프상의 이동 경로는 흰색선으로 표시됩니다.

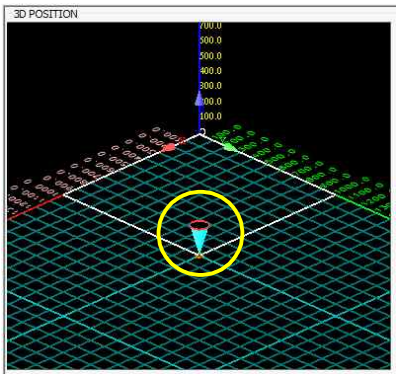


2.3.5. 직선보간, 원호보간 동작하기

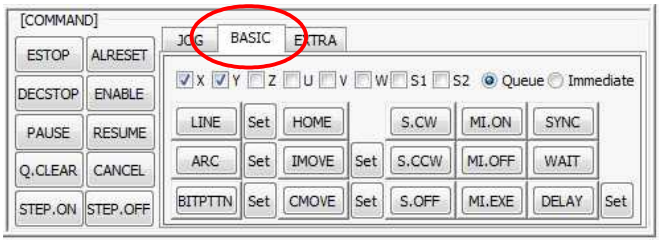
여기서는 조그버튼을 통해 간단한 움직임이 아니라, 직선과 원호보간 동작을 테스트해 봅니다.

우선 이 동작을 수행하기 위해서는 위칸의 ENKIT 1000 키트제품과 같이 임의적 기계동작시에 위험요소가 없을 때만 동작시켜 봅니다. 아래의 순번대로 조작하여 직선보간과 원호보간 동작을 테스트해 봅니다.

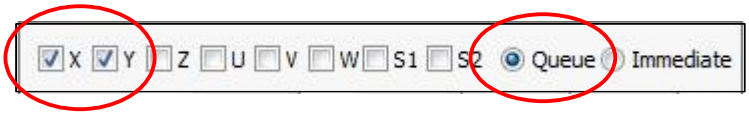
① 조그동작을 통해 위치 X,Y를 적당히 1000,1000 으로 이동해 봅니다.
3D 그래프상의 포인터(Pointer)위치가 아래와 같이 움직였을 것입니다.



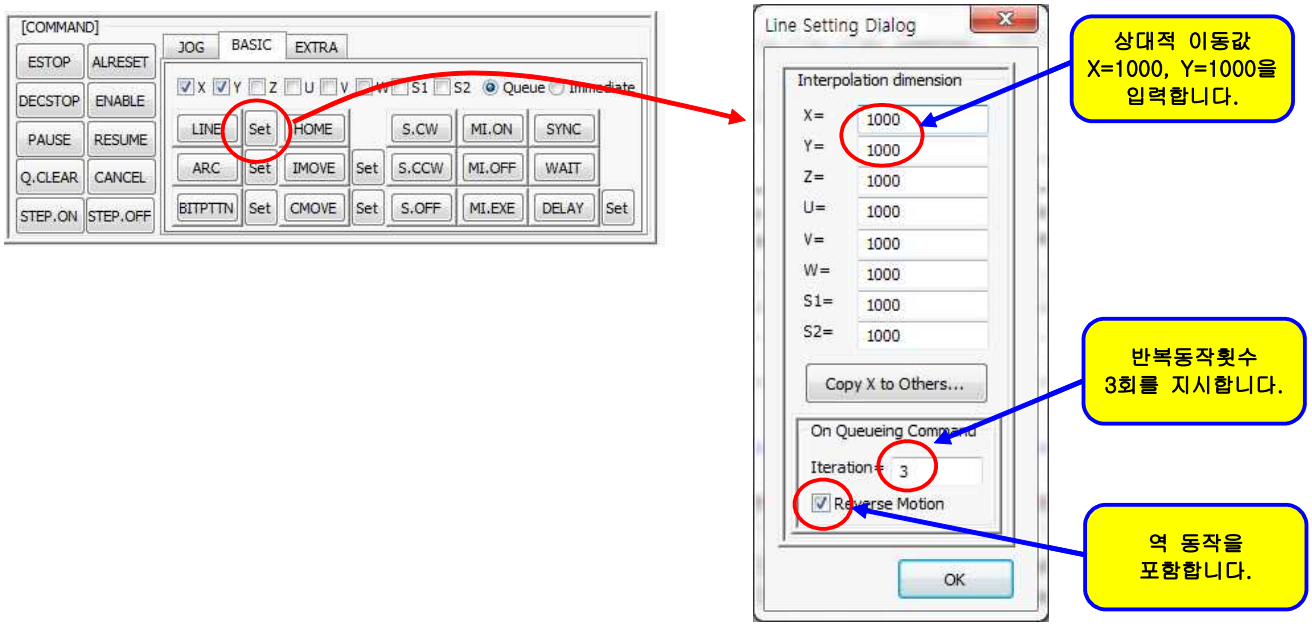
② 직선보간을 위해 [명령화면]의 기본 명령탭인 [BASIC]을 선택하면 아래와 같은 다양한 명령들이 나타납니다.
이들 명령들은 모션엔진에서 지원하는 기본적인 모션동작명령들 중 주요한 명령들로 구성되었습니다.



③ 보간동작을 위해서 먼저 명령 실행축과 명령실행방식을 선택합니다.
명령실행축은 X,Y축만 선택하고 명령실행방식은 명령큐를 이용한 방식을 선택합니다.



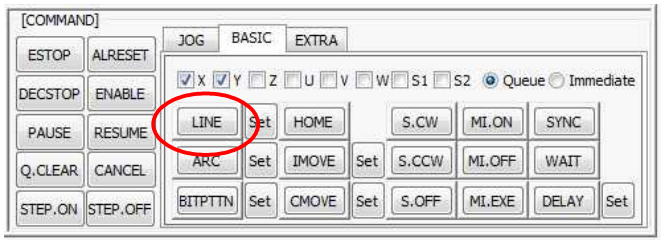
④ [LINE]의 동작설정 버튼[SET]눌러 LINE명령 파라메터를 설정합니다.
- 상대적 이동값 X=1000, Y=1000 을 설정합니다.
- 반복동작횟수(Iteration) 3회를 지시합니다.
- 역동작(Reserve Motion)을 설정합니다.
- 설정창을 닫습니다.

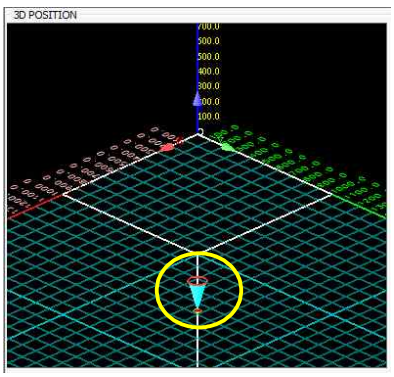


⑤ 연속실행 옵션을 해제합니다.

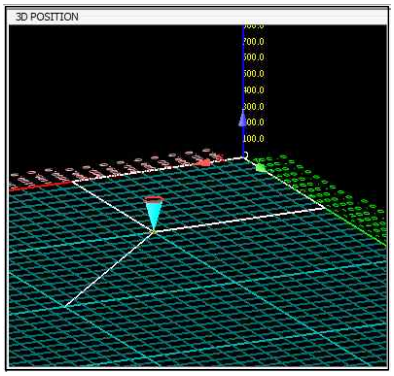
역동작을 설정하였으므로 방향 전환시 이동속도를 가감속하기 위해서 연속실행(CEXE) 옵션을 해제해야만, 기계시스템에 충격을 주지 않습니다.

⑥ [LINE]버튼을 눌러 LINE명령일 실행해 봅니다. 3D 그래프화면의 포인터(Pointer)이동을 확인해 볼 수 있습니다.





- 3d 그래프화면에 마우스를 올리고 마우스의 중간버튼(주로 휠버튼)을 누른상태에서 마우스를 왼쪽방향으로 살짝 움직이면 보는 위치 (Viewer position)을 바꿀 수 있습니다.

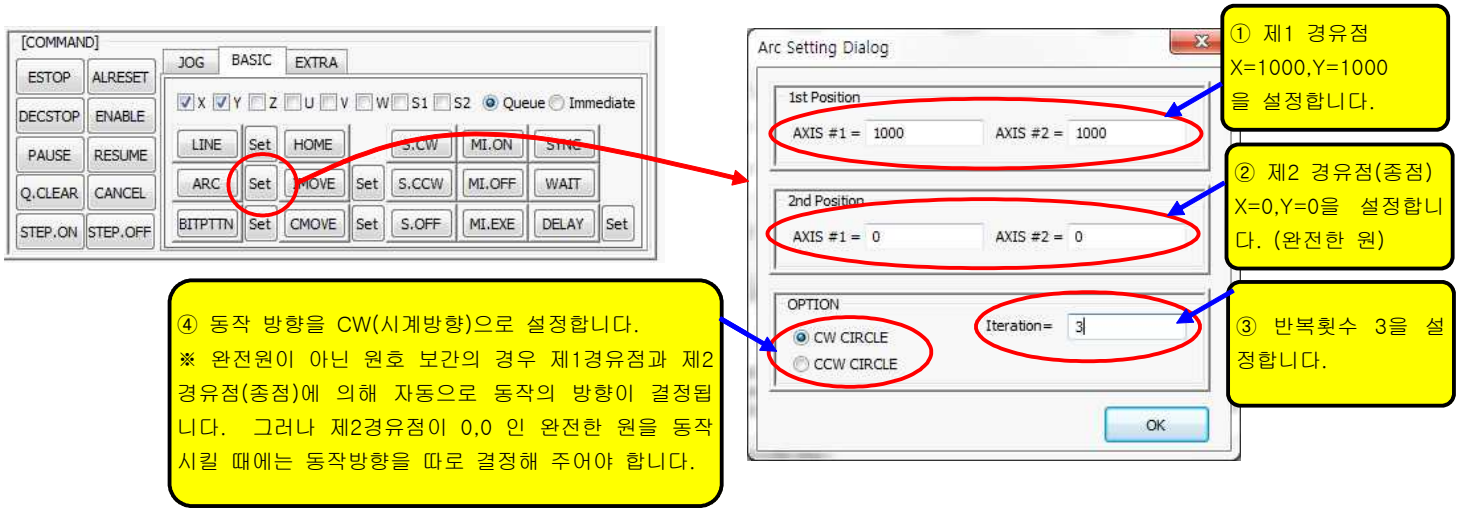


원래대로 복귀하려면 X-Y-Z 버튼을 눌러 주면 됩니다.

※ 전체 그래프화면의 이동은 왼쪽버튼을 클릭한 상태에서 마우스를 이동해 주면 됩니다. 3D 그래프화면의 마우스 조절기능을 간단하게 요약하면 아래와 같습니다.

- 마우스 스크롤 : 줌(Zoom) In/Out 기능
- 마우스 휠버튼 + 이동 : 회전(Rotate) 기능
- 마우스 왼쪽버튼 + 이동 : 이동(Move) 기능

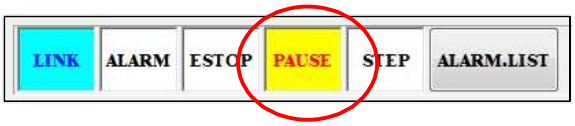
⑦ 원보호간 동작을 위해 설정버튼 [SET]을 눌러 아래와 같이 설정해줍니다.



※원호보간은 항상 2축만을 사용함으로 X,Y 축 혹은 Y,Z축 등으로 2개의 축만을 선택해야만 알람이 발생하지 않습니다. 만약 X,Y,Z축을 선택하고 원호보간을 실행하면 아래와 같이 [EMG.STOP]스위치가 점멸하면서 실행오류를 알려줍니다.

완전히 리셋을 필요로 하지 않는 오류이면 알람리셋 [RST.AL]

버튼을 눌러줍니다. 그러면 알람상태는 해제되고 일시정지램프[PAUSE]가 노란색으로 점멸하면서 현재 모션보드가 일시정지상태임을 알려줍니다.



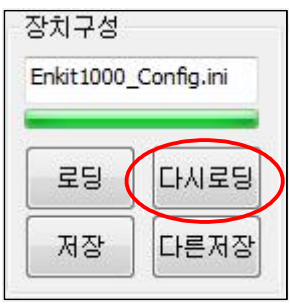
그리고 취소명령[CANCEL]

을 누르면 현재알람상태가 취소되고 남아 있는 명령큐의 명령들도 모두 없어지게 되고 정상실행상태로 돌아옵니다. 만약 현재명령만을 취소하고 명령큐에 남아 있는 다른 명령들을 이어서 실행하려면 [CANCEL]버튼 대신 [RESUME]

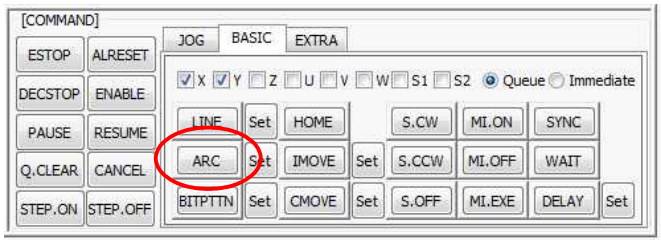
버튼을 눌러 줍니다.

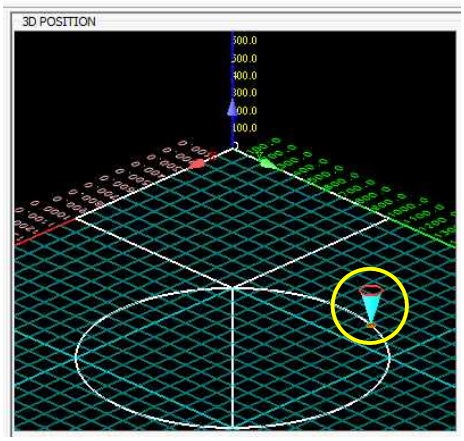
알람리셋 명령에 의해서도 알람이 해제되지 않는다면 리셋버튼

을 눌러 모션엔진을 소프트 리셋 합니다. 소프트리셋(SOFT RESET)은 통신이 가능한 상태에서 통신명령에 의해 리셋동작을 수행하는 것이며, 하드리셋(HARD RESET)은 통신일 불가능해서 모션보드의 리셋스위치를 누르거나 전원을 껐다 켜는 동작을 말합니다.)
리셋 되면 다시 구성파일에 의한 보드초기화가 수행됩니다. 만약 초기화가 수행되지 않았을 때에는 동작탭화면의 장치구성에서 [다시로딩]버튼을 눌러주면 이미 지정된 구성파일로서 재초기화가 이루어집니다.



⑧ [ARC]버튼을 눌러 원호보간동작을 실행합니다. 3D 그래프의 포인터움직임을 확인합니다.





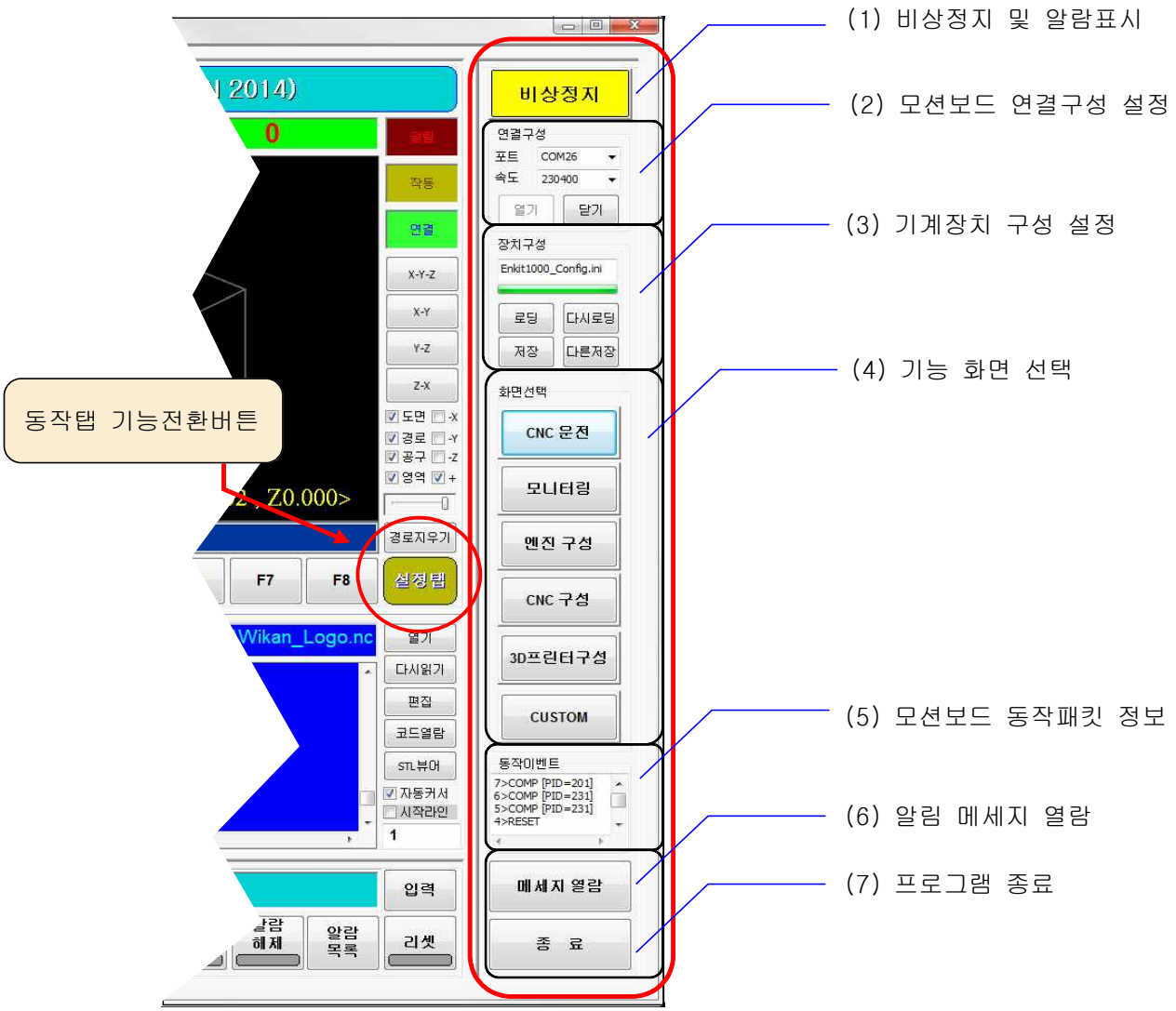
☞ 간단하지만 모션동작을 시험운전을 마쳤습니다.

3. 전체 동작화면 요약

모션마스터는 아래의 동작탭화면을 통해서 다양한 기능의 다른 동작화면으로 전환될 수 있습니다.
여기서는 각각의 동작화면에 대한 기능들을 간략히 요약해 봅니다.
먼저 아래의 동작탭화면의 구성을 살펴봅니다.

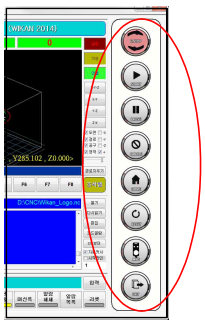
※ 모션마스터의 프로그램 화면은 프로그램 버전 및 별도의 알림없이 변화될 수 있습니다.

3.1. 동작탭 화면



동작탭화면은 동작탭 기능전환버튼에 의해 몇가지 아래와 같이 또 다른 기능의 화면으로 전환될 수 있지만 여기서는 가장 기본적인 [설정탭]기능의 화면을 보여주고 있습니다.

조작탭 화면



조그탭 화면

아래에는 동작탭의 각 기능별 내용들을 기술합니다.

(1) 비상정지 및 알람 램프

모션보드(혹은 기계동작)의 동작을 중지시키는 버튼입니다. 또한 보드내부의 알람 발생시에 적색으로 점멸하여 알람발생을 알려주기도 합니다. 해제하려면 리셋버튼[RESET]나 알람리셋버튼[RST.AL]를 눌러줍니다.

(2) 연결구성 관리

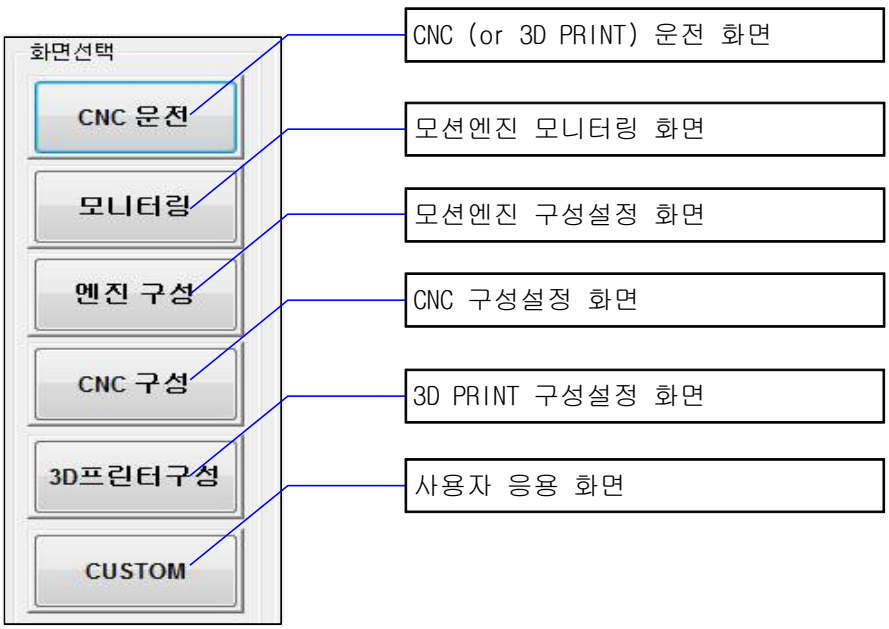
X6V3 모션엔진에 기반한 위칸모션컨트롤러는 기본적으로 RS232C, USB 및 제품에 따라라서 ETHERNET통신을 지원합니다. USB와 통신할 경우 해당 모션보드에 적절한 USB드라이브를 설치 한 후 시도할 수 있습니다. 통신속도의 설정은 제품마다 조금씩 다를 수 있으나, 기본적으로 115200BPS, 230400BPS, 460800BPS, 921600BPS 중의 하나로 선택되어 질 수 있습니다. (일반적으로 115200 혹은 230400을 선택합니다.)

(3) 구성설정파일 관리

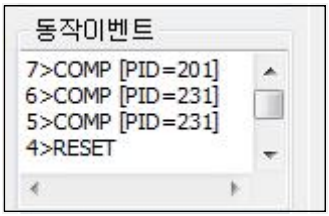
WIMC_CONFIG.INI 파일은 모션마스터 내에서 설정된 모든 파라메터들을 기록[SAVE]하고, 모션보드를 원하는 구성에 맞게 한 번에 초기화 할 수 있는 기능[LOAD]을 갖습니다. 구성설정파일은 임의로 지정하여 저장 혹은 로드 할 수 있으나, 모션마스터가 시작 될 때 항상 WIMC_CONFIG.INI 이름의 파일을 읽어들여 모션보드를 초기화 합니다.
[RELOAD] 명령은 현재 구성파일을 다시 읽어들여 보드를 재 셋팅하는 역할을 합니다.

(4) 기능선택화면

각 버튼을 누르면 해당 동작화면으로 전환됩니다.
각각의 화면에 대한 내용들은 다음절부터 간략하게 설명됩니다.



(5) 상태 패킷 정보

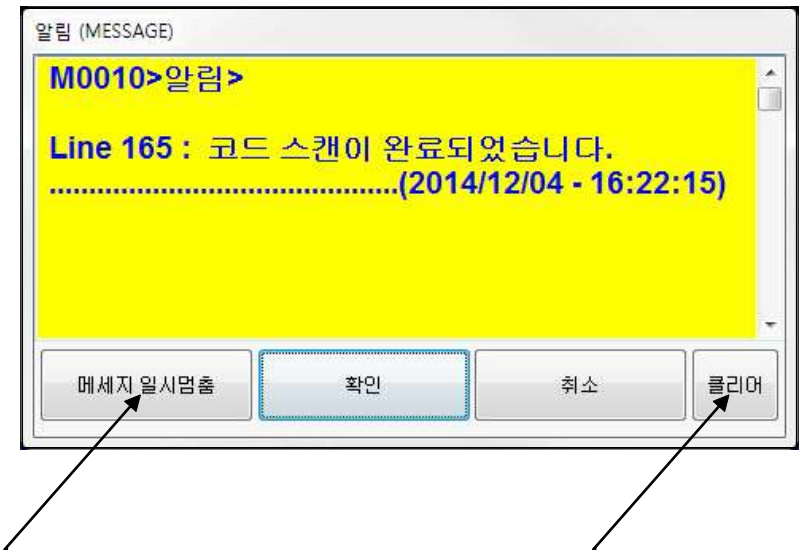


모션보드로부터 수신되는 정보들 중 특별한 상태나 이벤트 등이 발생했을 때 해당정보(메시지)를 표시하는 영역입니다. 예를 들면 모션보드가 리셋 될 경우, 모션엔진은 자동으로 리셋상태 패킷(PACKET)을 발생하게 되고, 리셋상태 메시지를 표시하게 됩니다.

(6) 알림 메세지 열람

모션마스터의 주요상태변화에 따라 아래와 같은 알림 메세지 창이 일정시간동안 표시됩니다. 주로, 리셋상태, 알람상태, 코드스캔상태, 원점복귀 완료상태,스크립트 오류상태 및 사용자 임의 메시지 발생 등 다양한 상태에서 알림 메세지 창이 발생되고, [메세지 열람] 버튼을 눌러주면 최근의 메시지 정보를 확인할 수 있습니다.

<알림 메시지 창>



메시자가 일정 시간후 사라지는 것을 방지합니다.

이전에 발생된 전체 메시지를 삭제합니다.

(7) 프로그램 종료

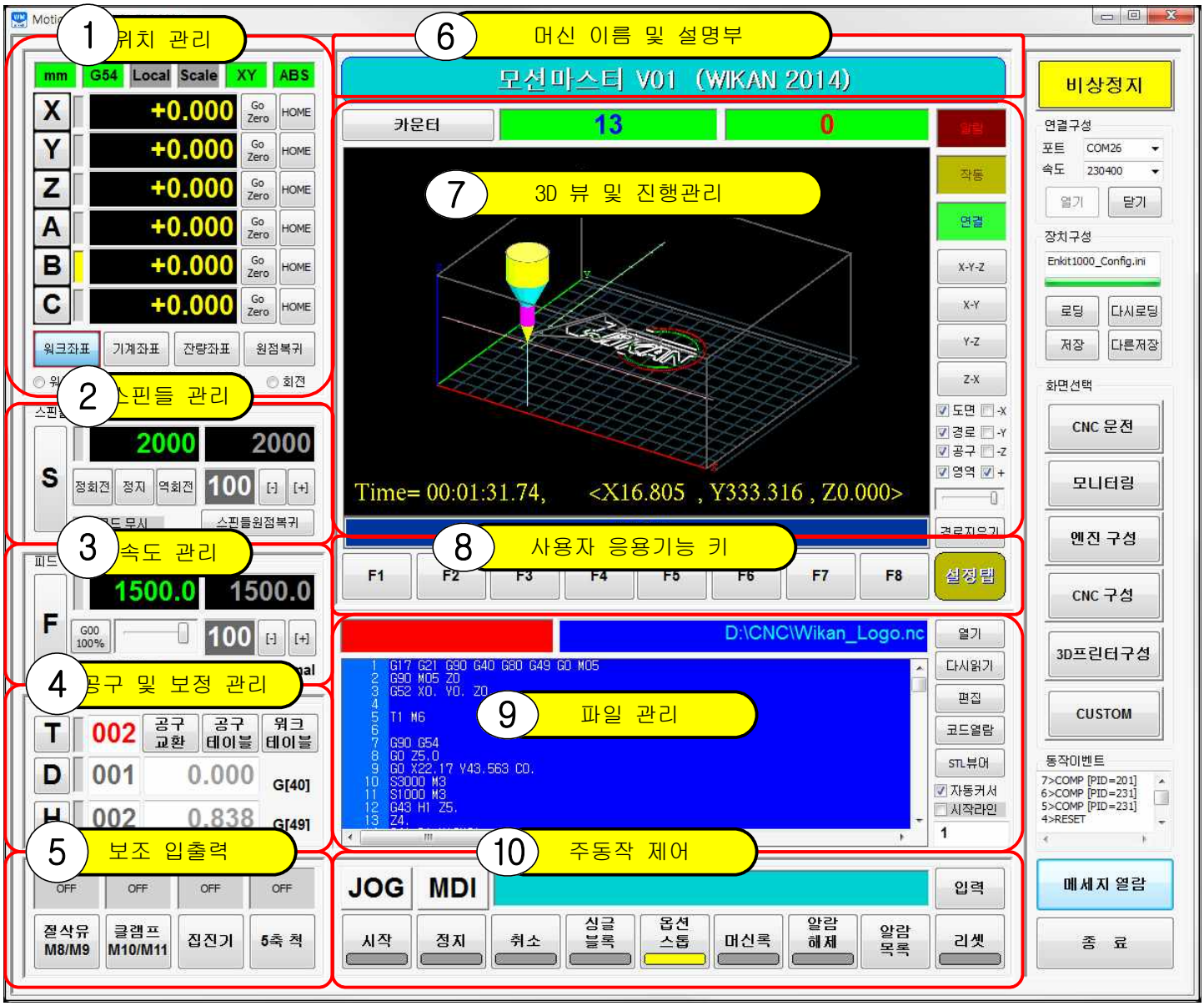
프로그램을 종료시킵니다.

3.2. CNC (or 3D PRINT) 운전 화면

위칸 모션컨트롤러와 연결된 기계장치를 G코드기반의 밀링형 CNC 형태로 운영할 수 있는 응용프로그램 화면입니다. G코드 기반으로 운영되는 FDM방식의 3D프린터의 운영도 가능합니다.

CNC 운전화면은 아래의 몇 가지 동작 및 관리영역으로 나누어 집니다. 아래의 화면그림은 CNC운전화면의 각 관리영역들을 표시합니다.

(1) 위치관리
(2) 스핀들관리
(3) 속도관리
(4) 공구 및 보정관리
(5) 보조입출력
(6) 머신이름 및 설명부
(7) 3D 뷰 및 진행관리
(8) 사용자 응용기능 키
(9) 파일관리
(10) 주동작 제어

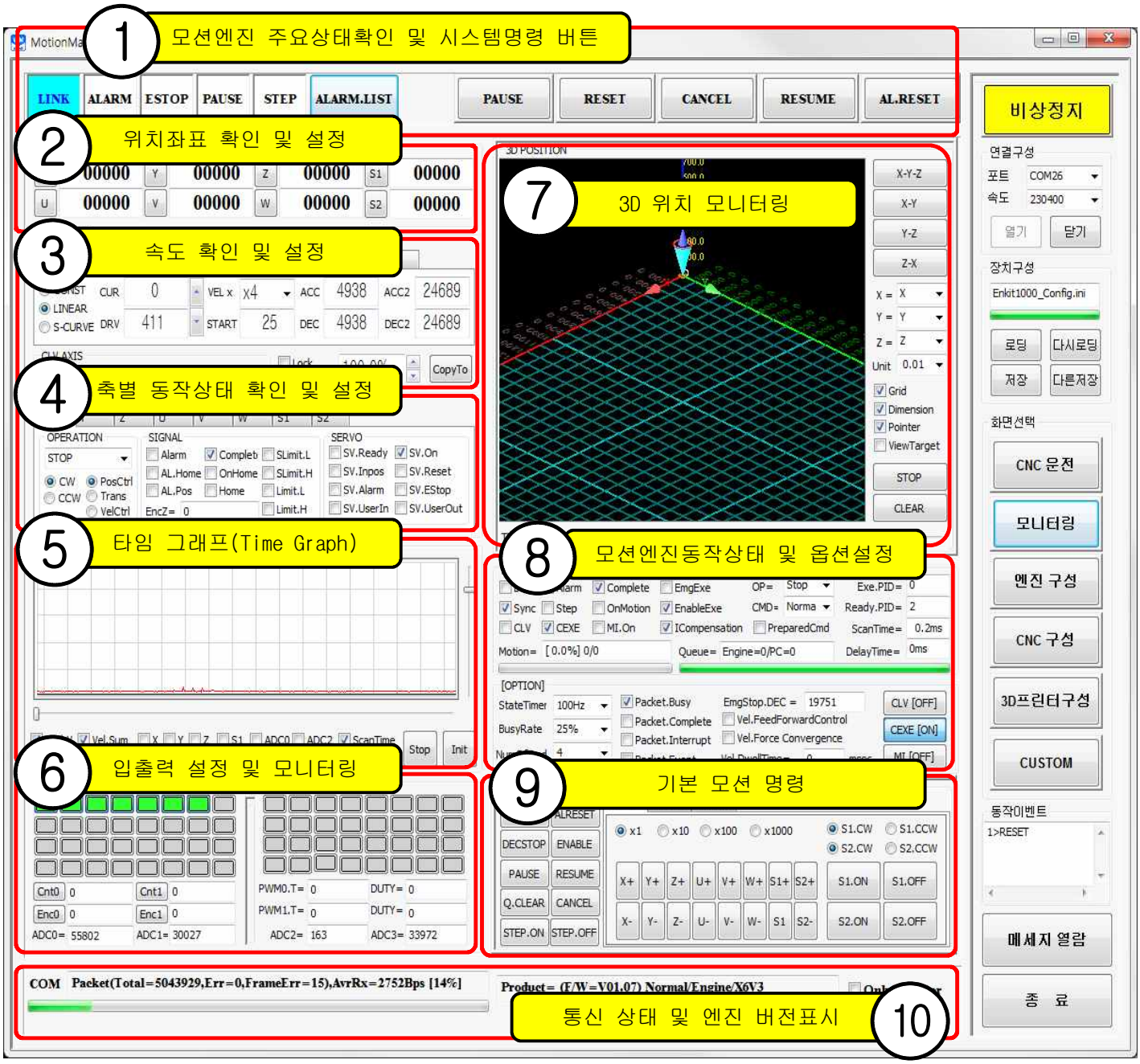


3.3. 모니터링 화면

위칸모션컨트롤러는 내부에 기본적으로 자체 개발된 X6V3 모션엔진이 장착되어 있습니다. 실제 모션마스터는 모션보드와 통신하면서 동작명령을 주고, 상태를 받으면서 운영됩니다. 모니터링화면은 모션보드와 직접적으로 통신하면서 모션보드에 펄스단위의 로레벨명령을 전송하고, 현재 동작상태를 직접적으로 모니터링할 수 있는 화면입니다. 주로 기계와 모션보드(모션컨트롤러)간의 전기적신호방식이나 동작방향 혹은 입출력을 확인등과 같은 아주 기본적인 전기적 상태를 확인하는데 사용됩니다.

모니터링화면은 "엔진구성" 화면과 직접적으로 관련되어 엔진구성에 따른 동작상태를 즉각적으로 반영하게 됩니다. 또한 엔진구성의 일부내용이나 자주 변경될 수 있어 엔진구성에 포함되지 않는 내용들을 설정하는 기능도 포함됩니다.

모니터링 화면과 주요한 기능들은 아래의 그림을 참조하실 수 있습니다.

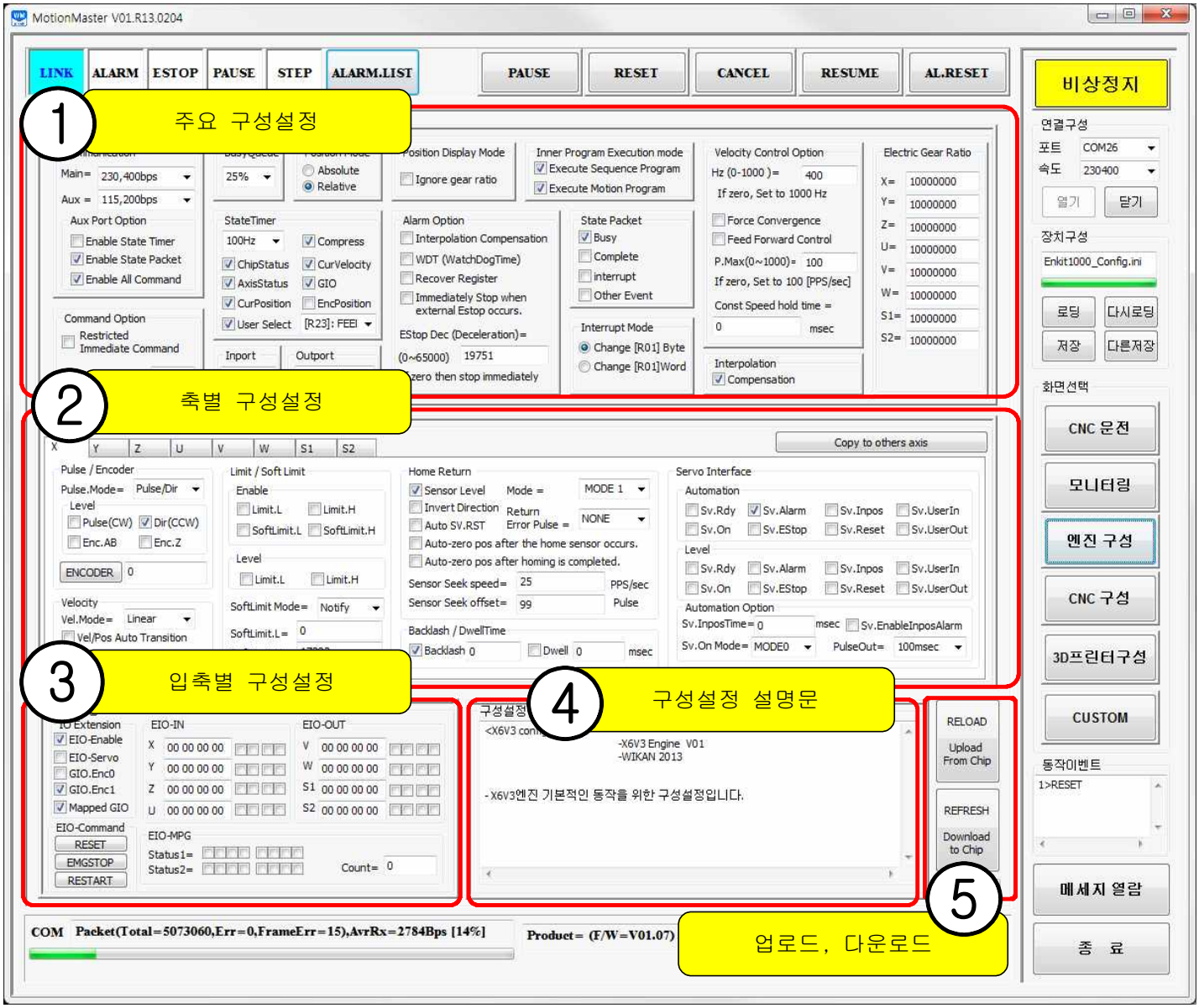


3.4. 엔진구성 화면

엔진구성 화면은 모션컨트롤러의 기본적인 동작특성과 모터드라이브간의 신호체계 및 기계장치의 동작방향 및 동작특성 그리고 전기적 접속 방식 등 모션컨트롤러에 내장된 모션엔진칩을 직접적으로 구성 설정하는 기능을 갖습니다.

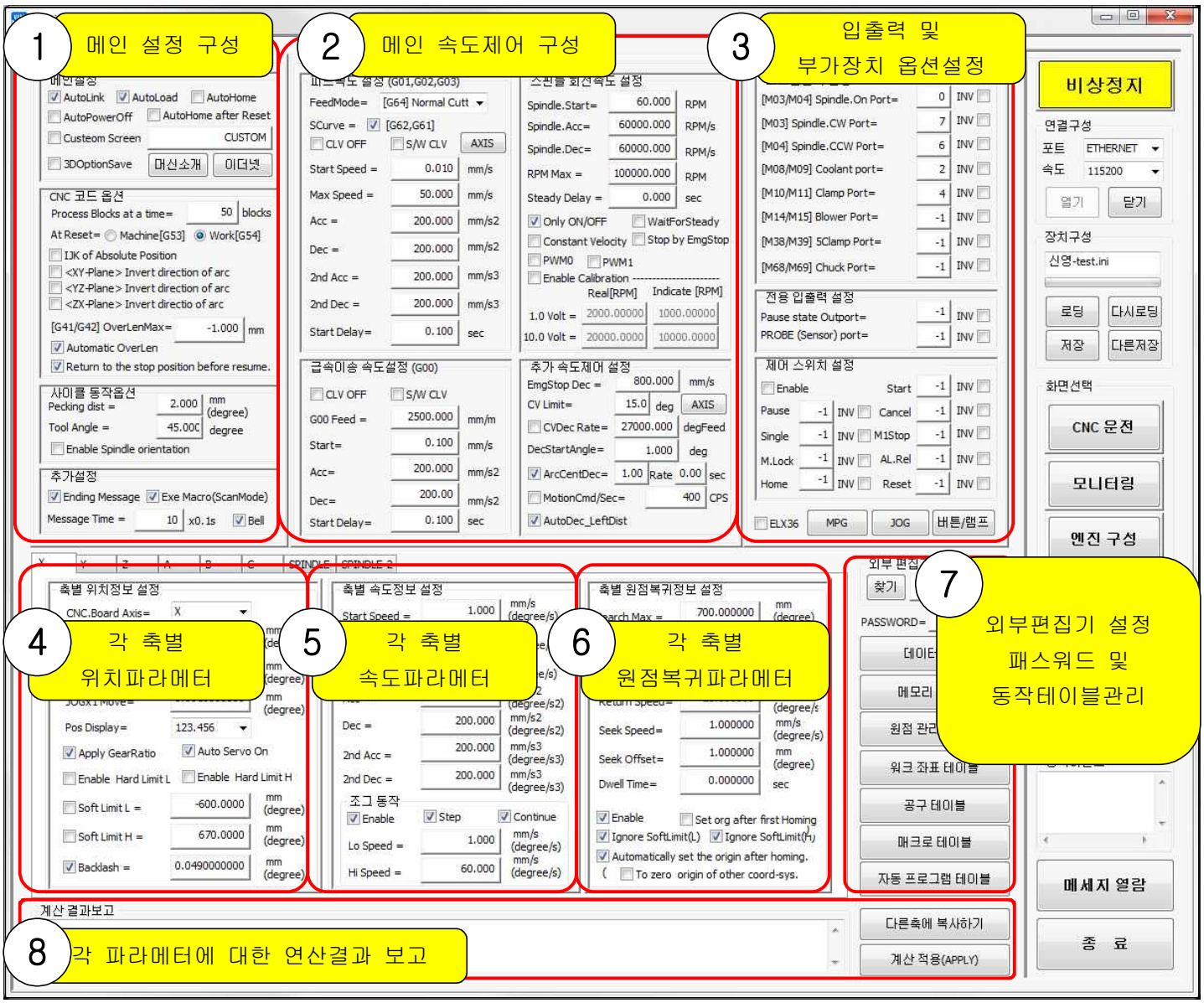
아래에는 주요한 설정그룹과 각각의 설정그룹에 대한 엔진구성 화면의 영역을 표시합니다.

(1) 모션엔진 주요 구성설정
(2) 모션엔진 축별 구성설정
(3) 모션엔진 GIO/EIO 입출력 구성설정
(4) 모션엔진 구성설정 설명문
(5) 업로드, 다운로드 기능



3.5. CNC 구성 화면

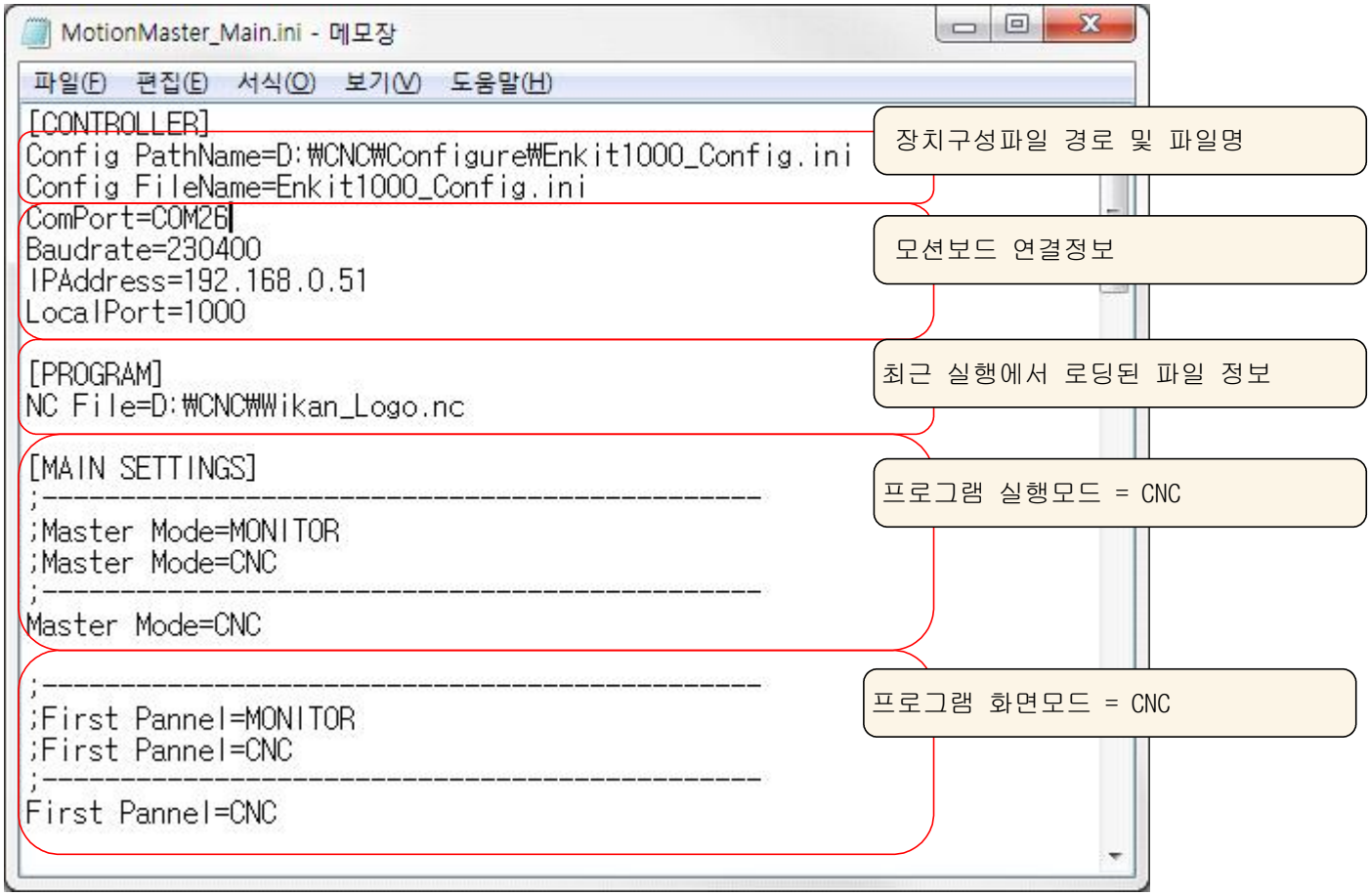
CNC 구성 화면은 CNC운용을 위한 기본적인 설정을 포함하고 있습니다. 엔진구성이 모션컨트롤러의 기본적인 동작과 디지털 펄스단위의 전기적신호체계 등을 설정하는 곳인 반면에 CNC 구성설정은 펄스위치제어방식의 모션컨트롤러가 XYZ 실좌표계 시스템에 어떻게 맵핑되어 위치와 속도 등의 각 동작특성이 조절되는지에 대한 설정들을 포함하게 됩니다. 다소 많은 옵션들로 구성되며 전체적으로는 아래의 그림과 같은 분류의 기능들로 요약될 수 있습니다.



3.6. MotionMaster_Main.ini 구성파일

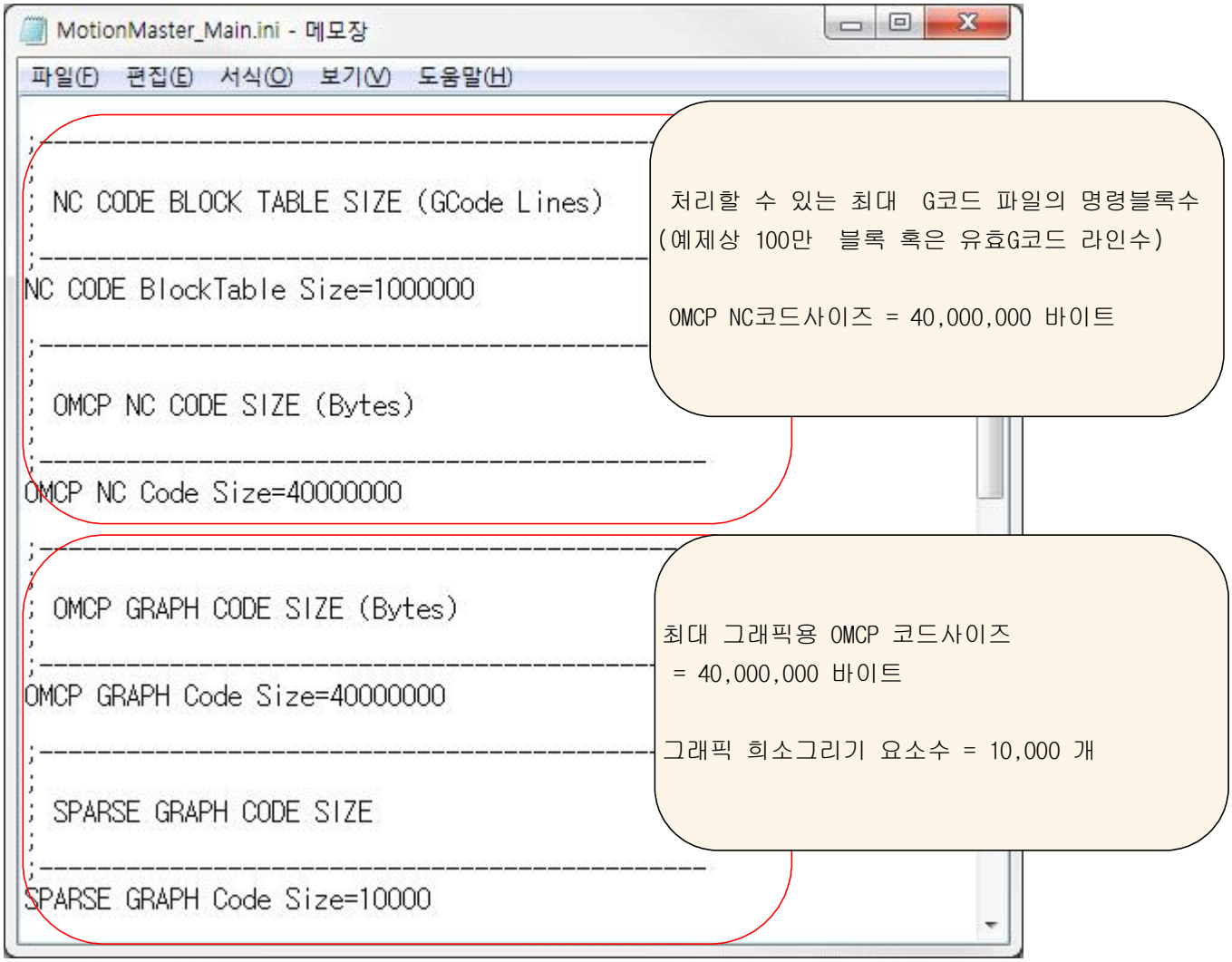
MotionMaster_Main.ini 구성파일은 모션마스터와 모션컨트롤러가 연결되어 실행되는 기본적인 환경설정을 포함하는 파일입니다.
이 파일을 구성하는 정보들은 프로그램 실행성능에 영향을 주며, 모션보드연결 및 화면구성에 영향을 줍니다.
이 파일은 아스키 텍스트파일로 구성되었으며 NotePad와 같은 간단한 편집기로 수정될 수 있습니다.
이들 중 주요한 정보들을 살펴봅니다.

(1) 장치구성파일과 연결정보 및 프로그램 초기모드



상기 구성중 장치구성파일의 경로파일명과 모션보드 연결 정보 및 최근 로딩된 파일정보등은 프로그램이 정상적으로 실행되고 종료되면 자동으로 저장됩니다.
이 구성정보에 의해서 CNC구성화면 메인설정의 AutoLink(자동연결), AutoLoad(자동로딩) 옵션을 체크할 경우 자동으로 모션보드와 연결되고 자동으로 이전 실행에서 로딩된 최근파일을 자동으로 로딩하는 동작의 참조로 사용되게 됩니다.

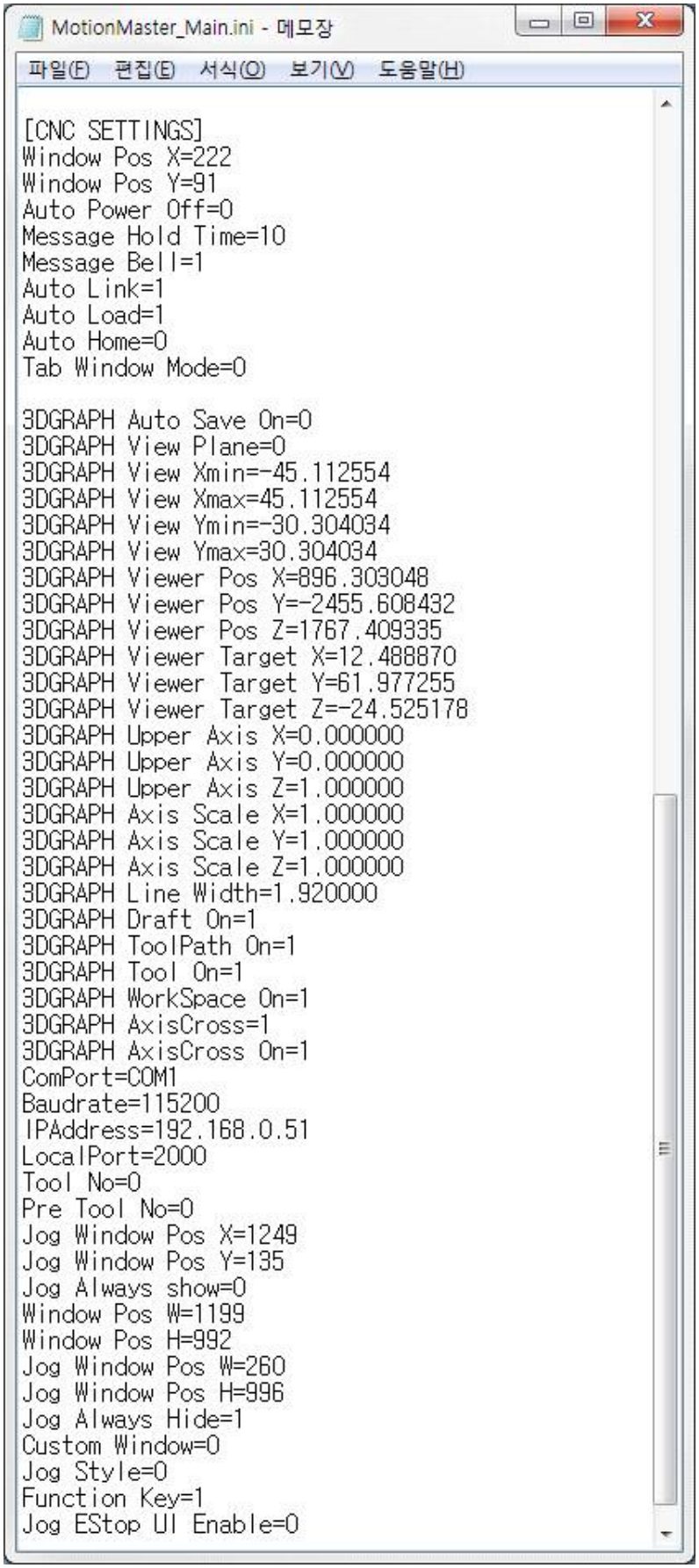
(2) G코드 실행규모 및 3D그래프 표현규모



G코드명령은 내부적으로 내부에서 규정된 OMCP코드로 전환됩니다.
G코드명령처리 최대수는 시스템에 따라 다르지만 최대 약 300만 라인까지만 가능합니다.
한 번에 처리되어야 하는 G코드량에 많을 수록 많은량의 하드디스크메모리를 요구하게 됩니다.
사용 용도에 맞게 적절한 크기를 조절할 수 있습니다.

OMCP GRAPH Code Size는 내부의 3D 그래픽처리를 위한 메모리 할당크기입니다.
SPARSE GRAPH Code Size 는 3D 그래픽을 최소화 그리기동작에 적용될 그리기요소(Entity)수 입니다.

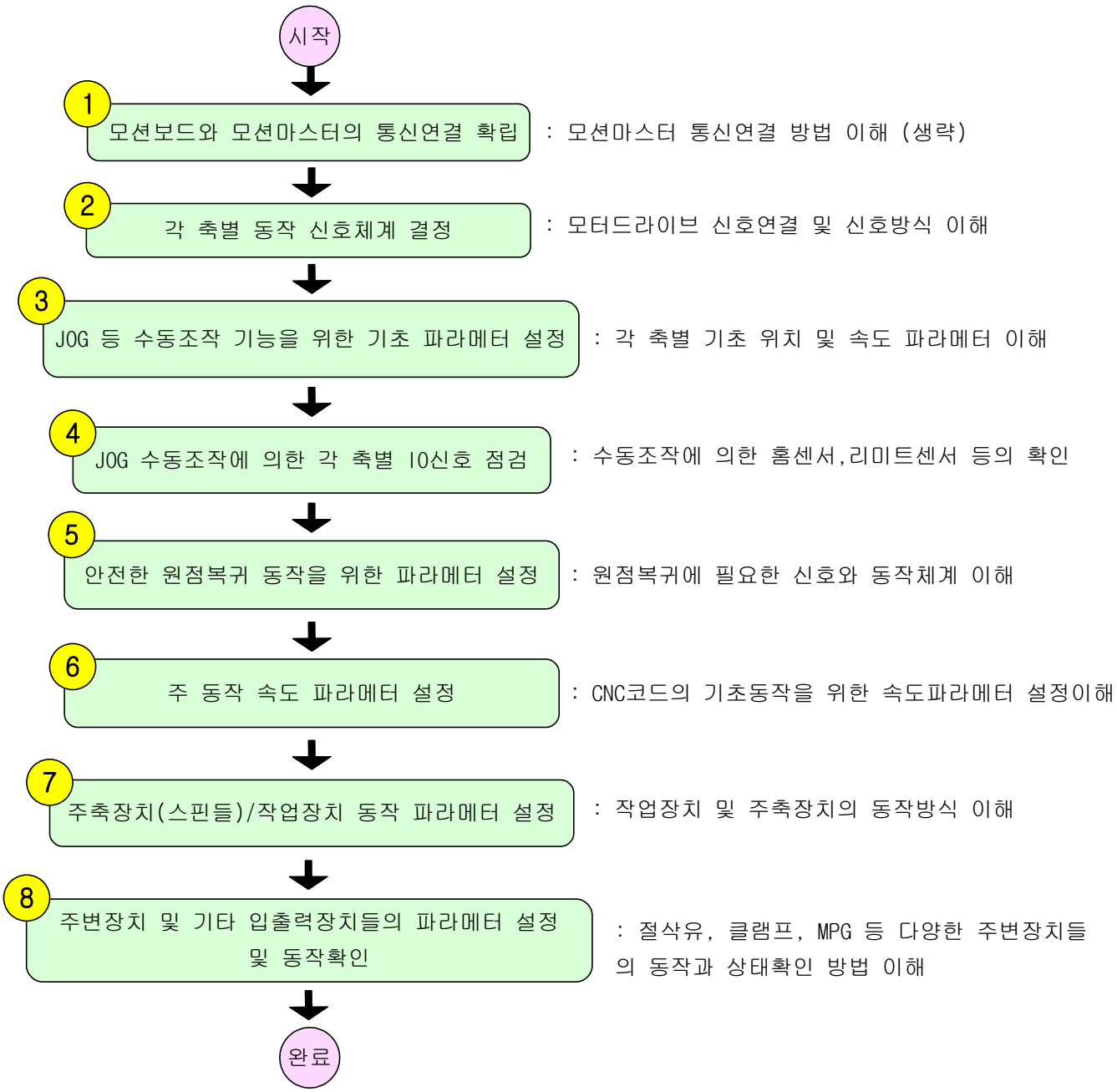
(3) 아래의 구성설정들은 프로그램의 화면생성위치와 프로그램내의 화면구성 과 CNC동작과 관련된 몇 가지 기본설정에 관한 내용을 포함합니다.



4. CNC 장치 연결 기초 가이드

이 장은 모션마스터와 모션컨트롤보드가 연결되었을 때 CNC장치와 모션컨트롤러 및 모션마스터가 바르게 동작하기 위한 기본적인 파라메터 설정 절차를 설명합니다. 사용자의 머신종류와 특징에 따라 다소 다른 옵션들이 요구될 수 있지만 이 장에서는 보다 공통적으로 적용되는 구성요소들에 대한 파라메터들을 중심으로 엔진구성과 CNC구성화면에서 관련된 내용들을 살펴봅니다.

먼저 아래의 그림을 통해 일반적인 CNC 기계장치를 처음 작동 시킬때 필요로 하는 절차의 흐름을 파악하고 다음 절에서 각 설정절차에 필요한 세부적인 내용들을 살펴봅니다.(우선 기계시스템과 전기제어시스템의 기초적인 조립 완성 상태로 가정하겠습니다.)



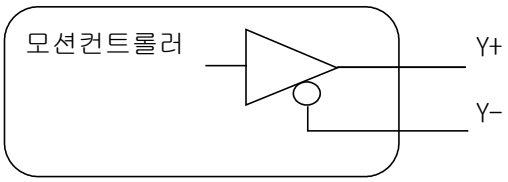
4.1. 모터드라이브 신호 연결 및 신호방식

먼저 일반적인 전기제어시스템에서의 전기신호종류를 간략히 살펴보고, 서보모터드라이브의 신호종류 및 연결을 간단히 살펴봅니다. 이들 신호에 대한 이해가 별도롤 필요하지 않을 경우 “모션마스터 프로그램의 축신호 설정”절 편으로 바로 이동하시면 됩니다.

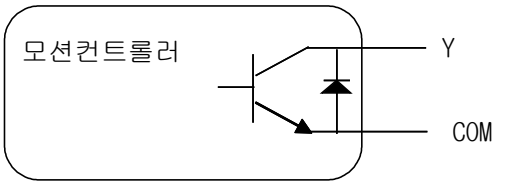
4.1.1. 전기신호

모터드라이브는 일반적으로 위치제어에 알맞은 스텝모터드라이브 혹은 서보모터드라이브와 연결되는 것을 기준으로 설명합니다. 모션컨트롤러와 모터드라이브의 연결은 크게 축의 펄스동작 신호연결과 모터드라이브 IO제어 연결이 있습니다. 먼저 모션컨트롤러가 모터드라이브의 축 펄스 전기신호를 연결하기 위해 아래의 3가지 종류의 연결신호 중 하나를 선택하게 됩니다.

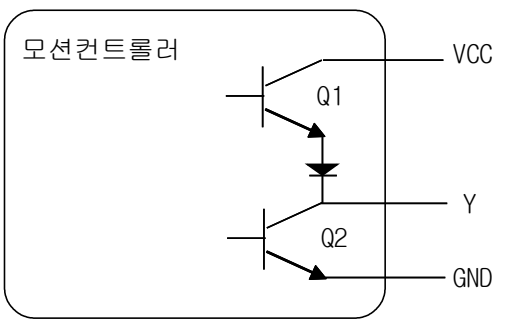
첫째는 차동라인신호(Differential Line driver signal) 혹은 RS422 신호방식으로 불리는 5V 전압규격의 차동전압 방식 입니다. 전기적 기호는 보통 아래와 같이 표현됩니다.
이 방식은 Y+가 Y-보다 높은 전압상태(5V정도)일 때(활성상태)와 반대로 Y-가 Y+보다 높은 전압상태일 때 (비활성상태)의 두가지 상태로 의미를 전달하는 전기신호방식입니다.



다음으로 오픈콜렉터신호방식입니다. 이 방식은 COM과 Y가 전기적으로 연결될 때(활성상태)와 연결되지 않을 때(비활성상태)의 두 가지 상태로 의미를 전달하는 전기신호방식입니다.
이때 Y신호의 전압레벨은 항상 COM신호의 전압레벨보다 높거나 같아야 합니다.



마지막 방식은 토템폴방식으로써 Q1과 Q2가 서로 반대로 통전되어 VCC전압이 Y로 연결되거나(활성상태), 반대로 GND전압이 Y로 연결 되어(비활성상태) 전기신호를 전달하는 방식입니다.



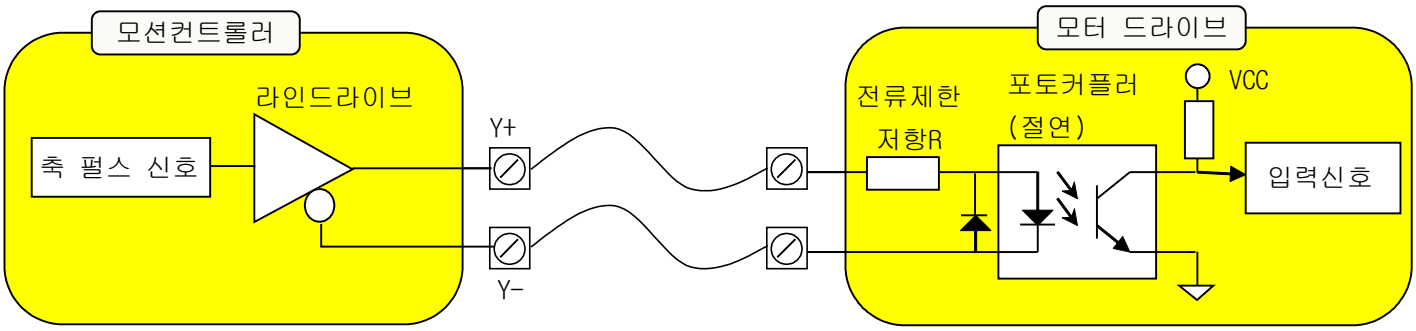
이러한 전기적신호방식은 단지 모션컨트롤러에게만 적용되는 방식이 아니라 대부분의 전기컨트롤러의 일반적인 전기신호 연결방식으로 사용됩니다.

이들 신호 중 대부분 모션컨트롤러는 노이즈내성이 비교적 강한 첫번째 라인드라이브방식의 신호연결을 선호합니다. 위칸모션컨트롤러도 대부분 첫번째 방식의 신호연결을 제공할 것입니다. 마지막 토템폴방식의 신호연결은 모션컨트롤러에서 잘 사용되지 않는 방식입니다.

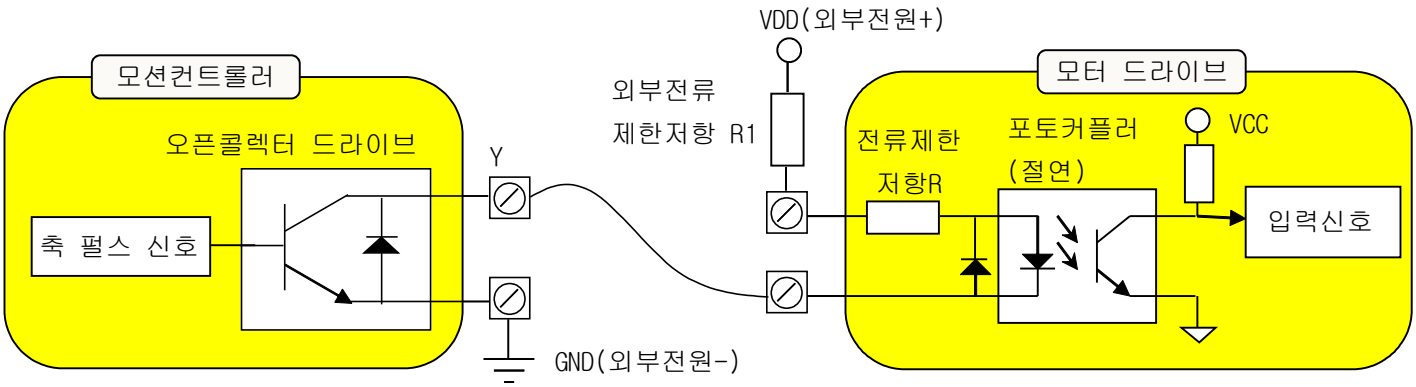
대부분의 제어회로장치들의 입력은 포토커플러에 의해 절연된 경우가 많습니다. 특히 모터드라이브의 펄스구동부는 고속의 펄스열을 받을 수 있는 높은 주파수특성을 갖는 포토커플러를 주로 사용하게 됩니다.

여기서는 라인드라이브와 오픈콜렉터 신호방식으로 출력되는 모션컨트롤러와 모터드라이브간의 연결회로를 살펴봅니다.

<차동 라인 드라이브방식>



<오픈콜렉터 드라이브방식>



모터드라이브의 펄스입력단의 전류제한저항은 5V입력영역으로 180~330옴 범위에서 내정된 경우가 많습니다. (특별히 입력전압의 범위가 24V인 경우다면 2K옴 부근의 저항치값을 갖고 있습니다.)
모션컨트롤러가 차동라인 드라이브방식으로 신호를 출력할 경우 대부분 Y+신호와 Y-신호를 구분하여 모터드라이브 입력단에 그대로 연결해 줄 수 있습니다.
그러나 오프콜렉터 신호를 출력할 경우 연결방식은 외부전원의 전압에 따라 전류제한저항 R1을 추가적으로 설치하여 그림과 같이 연결해 주어야 합니다. 외부전원이 24V이면 R1은 1.8K~2.2K옴 사이이면 적당합니다. 만약 외부전원을 5V로 한다면 R1을 없애고 VDD를 바로 접속할 수도 있습니다.

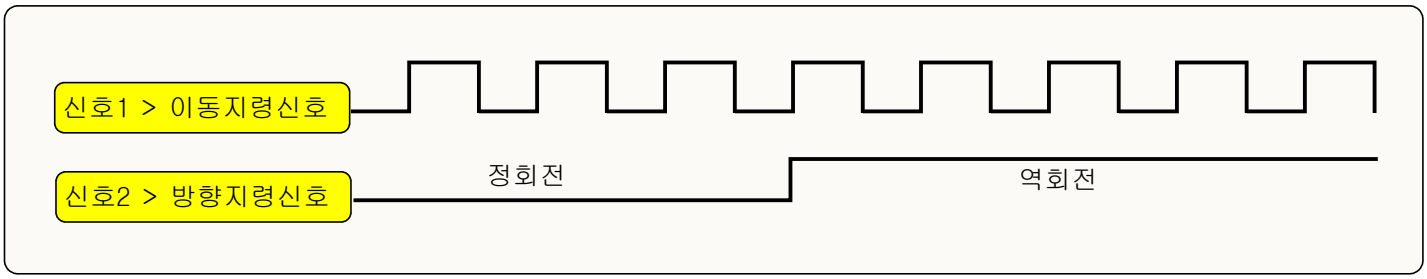
일반적으로 오픈콜렉터 드라이브는 축펄스신호보다는 축제어용 부가IO신호와 같은 입반출력신호로 많이 사용됩니다.

4.1.2. 신호방식

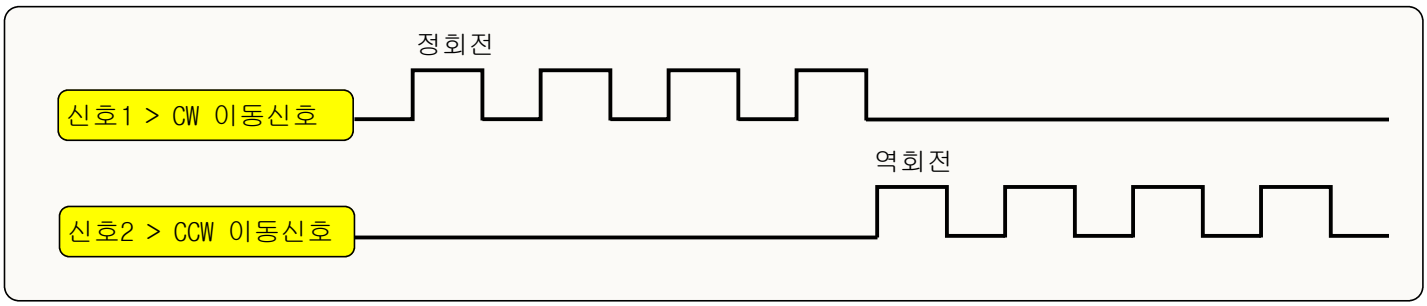
앞절에서 전기적 신호형태를 살펴보았다면 이번 절에서는 이러한 전기적신호의 변화를 어떤 방식으로 위치지령신호로 인식하는지에 대해서 설명합니다.
일반적인 펄스구동방식의 위치제어형 모터드라이브는 아래와 같이 크게 3가지 방식 중 하나로 신호를 처리합니다.
(1) 1펄스 방식 (Pulse / Direction)
(2) 2펄스 방식 (CW Pulse / CCW Pulse)
(3) 위상차방식 (Phase A/B)

모션컨트롤러는 하나의 축을 제어하기 위해 펄스신호 2개가 준비되어 있습니다. 이 펄스신호를 어떤 방식으로 사용하는냐에 따라 상기의 신호방식으로 나누어집니다.

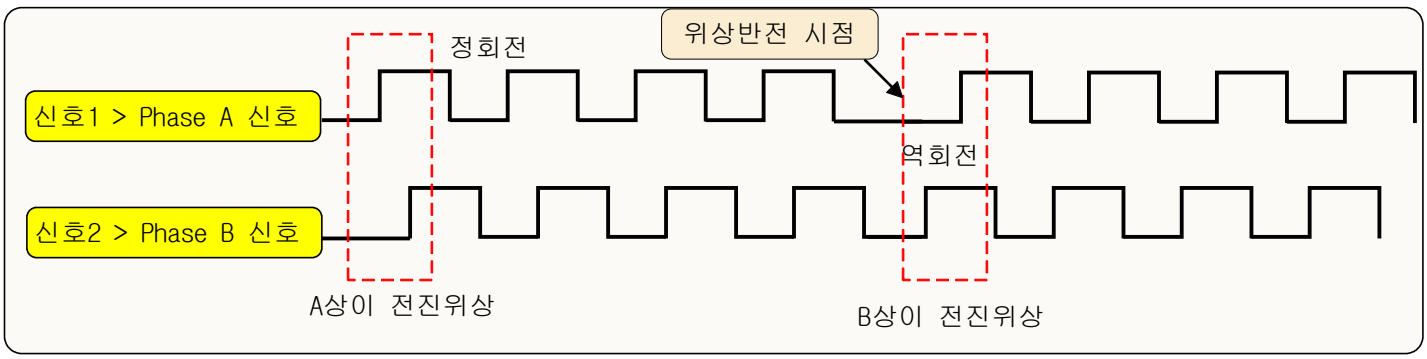
1펄스 방식은 아래그림과 같이 이동지령에 해당하는 하나의 신호와 이동방향을 지령하는 신호로써 위치를 지령하는 방식입니다.



2펄스 방식은 CW방향(시계회전방향) 이동지령과 CCW방향(반시계방향) 이동에 해당하는 각각의 펄스신호로써 위치를 지령하는 방식입니다.



위상차방식은 증분형 엔코드신호(Incremental encoder signal)와 동일한 방식으로 두개의 신호위상(Phase A, Phase B)차에 의한 전진위상과 후퇴위상을 구분하여 이동을 지령하는 방식입니다.



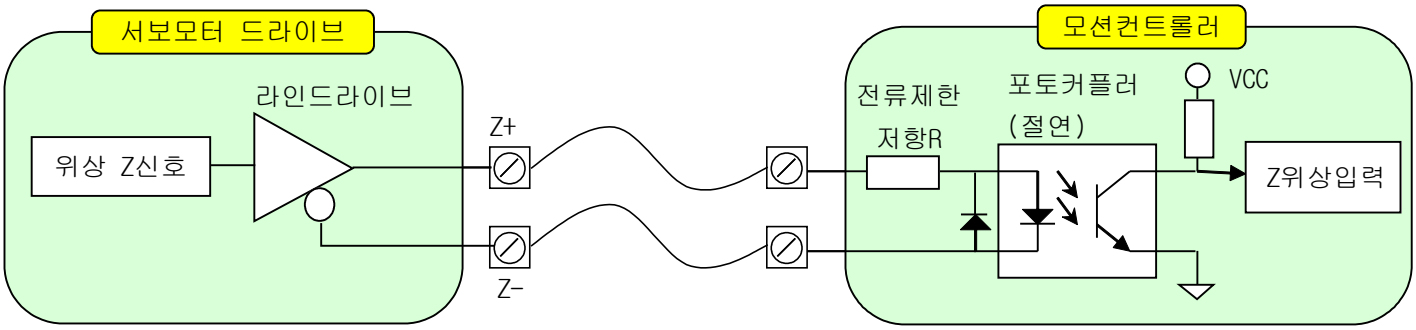
일반적으로 1펄스방식 혹은 2펄스방식을 많이 사용합니다. 또한 활성화상태(동작지령)신호를 H레벨 혹인 1인경우를 **정논리방식**라 부르며 비활성화상태(동작대기상태)신호가 H레벨 혹은 1인 경우를 **부논리방식**이라고 합니다.

4.1.3. 엔코더 피드백 신호

서보모터드라이브의 경우 모터회전에 따라 증분형 위상차신호인 위상A, 위상B, 위상Z신호가 출력됩니다. 모션컨트롤러가 엔코더의 신호를 피드백 받는 경우 이들 신호를 입력받아 내부적으로 카운트하는 기능을 가지고 있으면 이 신호를 연결하여 사용합니다.
엔코드 신호는 모터드라이브에서 모션컨트롤러방향으로 전달되는 신호로써 방향만 반대일뿐 상기절의 신호방식 중 위상차방식과 동일한 방식입니다. 또한 대부분 전기적신호형태로 라인드라이브방식일 취하고 있습니다.

현재 위칸모션컨트롤러 BLX6 등은 엔코더의 피드백신호 위상A와 위상B신호를 받아들이지 않습니다. 그러나 모터가 한 바퀴 회전시에 하나의 펄스가 출력되는 위상Z신호는 입력받게 됩니다. 위상Z신호는 특히 원점복귀동작에서 보다 정확한 위치의 복귀완료 위치가 필요로 할 때 유용하게 사용될 수 있습니다.

단지 신호의 방향만 다를 뿐 아래와 같이 라인드라이브의 축펄스신호연결과 동일한 방식입니다.



4.1.4. 그 외 축 부가 입출력 신호연결

대부분 컨트롤러에서 축의 위치지령펄스(출력) 및 엔코더신호(입력)연결은 각각 차동라인드라이브 방식과 고속포토커플러방식을 사용합니다. 이 들 신호이외에도 하나의 축을 동작시키기 위해서 아래와 같은 몇가지 부가 축 신호들이 사용됩니다. 이 신호 대부분은 출력은 오픈콜렉터방식으로 입력은 포토커플러방식으로 사용됩니다.
여기서는 간단히 이들 신호들의 종류와 기능에 대해서만 요약하며 보다 자세한 내용은 각 신호들을 이용하는 장들에서 설명될 것입니다. (아래의 신호 입력과 출력용어는 모션컨트롤러 입장에서 기술합니다.)

(1) 홈센서 신호입력
모션의 원점복귀동작은 홈센서신호를 찾아서 이동하는 모션동작입니다. 홈센서신호를 찾게 되면 지정된 방식대로 동작정지하여 원점복귀동작을 완료하게 됩니다. 대부분 기계는 실제기계좌표계와 CNC프로그램상의 위치를 일치시키기 위해서 원점복귀를 실행하게됩니다.

(2) 리미트센서 신호입력
리미트센서 신호란 축의 이동거리 제한을 위해 정방향 이동거리 제한신호와 역방향 이동거리 제한신호를 말합니다. 일반적으로 기계의 충돌 및 이탈을 방지하기 위한 목적으로 이동한계를 지정할 때 사용합니다. 물리적인 신호를 사용하여 이동을 제한하는 것을 하드리미트방식 이라하며, 내부적으로 이동위치값의 상한과 하한을 제한하여 이동한계를 설정하는 방식을 소프트 리미트방식이라고 합니다. 소프트리미트방식으로 기계움직임을 제한하기 위해서는 반드시 원점복귀동작에 의해서 CNC프로그램에서의 위치와 실제 기계위치를 일치시켜주어야 합니다.

(3) 서보온(SERVO ON)신호 출력
서보온 신호는 서보드라이브가 서보모터에 전기를 인가하여 서보동작을 수행하라는 지시입니다. 일반적으로는 CNC기계 시스템에 전기가 인가되면 바로 서보온 동작에 의해 각 축별 모터에 전기가 바로 인입되어 서보동작을 즉각적으로 이루어집니다. 그러나 때때로 서보 온을 해제하여 모터를 프리상태로 두는 경우도 필요할 수 있습니다. 특히 서보드라이브의 특정파라메터는 서보온 상태에서 조작되지 않을 수도 있습니다.

모션마스터프로그램은 서보온신호를 실행시 자동으로 ON할 수도 있으며 수동으로 ON/OFF해제할 수 있는 M코드를 제공합니다.

(4) 서보 브레이크(SERVO BRAKE)신호 출력
모션마스터프로그램과 위칸모션컨트롤러는 서보 브레이크신호는 별도로 사용하지 않습니다. 서보 브레이크신호는 Z축과 같이 모터에 전기가 인가되지 않으면 중력에 의해 슬라이딩 혹은 처짐이 발생할 때 필요로 할 수 있습니다. 중력에 의해 미끄럼이 발생되는 경우 브레이크가 있는 서보모터와 서보드라이브를 사용하며 서보브레이크출력은 서보온 신호를 반전하여 사용할 수 있습니다.

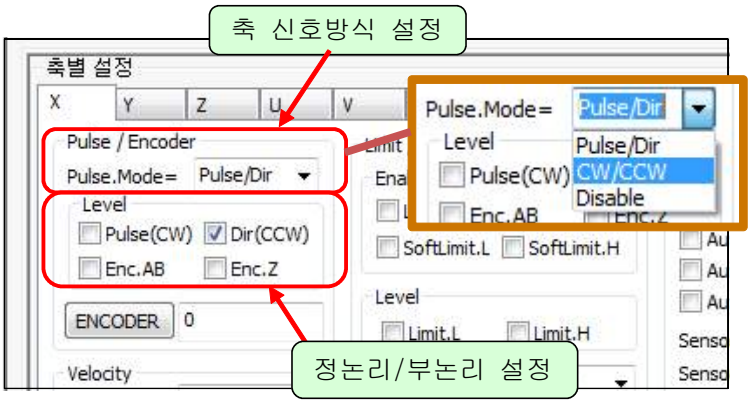
(6) 서보 알람(SERVO ALARM)신호 입력
서보드라이브에 이상이 발견되어 모션컨트롤러에게 동작중지요청을 하는 신호입니다.

(5) 서보 리셋(SERVO RESET)신호 출력
처음 서보를 동작시키거나 서보의 위치오차를 클리어하거나 서보에서 발생된 알람신호를 제거하기 위한 목적으로 사용되는 신호입니다. 모션마스터와 위칸모션컨트롤러는 리셋동작에서 서보리셋신호를 발생합니다.

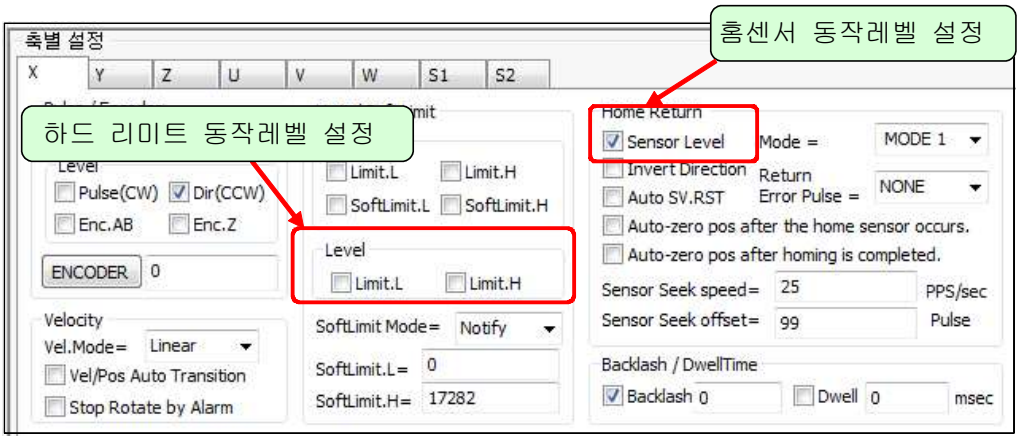
4.1.5. 모션마스터 프로그램의 축신호 설정

(1) 축 신호방식 설정

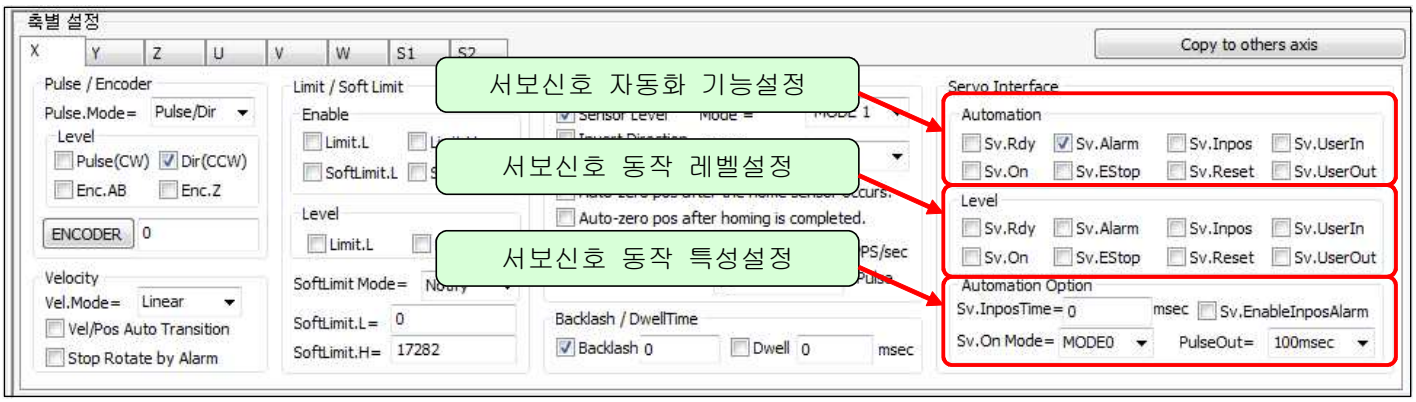
축신호 방식은 아래의 엔진구성화면에서 설정하게 됩니다.



(2) 홈센서,리미트 센서신호 레벨 설정



(3) 서보 부가 입출력 신호동작 설정



※ BLX6 등의 위칸모션컨트롤러는 서보온, 서보알람, 서보리셋의 부가적인 서보입출력신호는 모든 동작축에서 공유하여 사용합니다. 그러므로 모션마스터프로그램에서 각 서보부가 신호에 대한 동작활성화 및 동작레벨설정은 모든 축에 대해서 동일하게 설정되어야 유효합니다.

4.2. 각 축별 위치 파라메터 설정

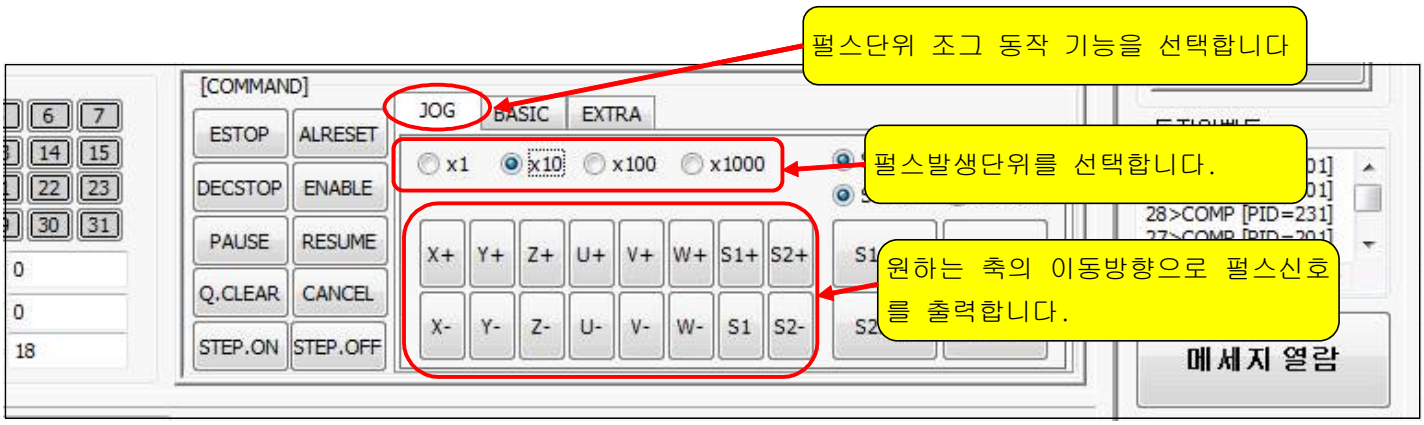
모터드라이브와의 축 연결이 완료되었다면 이제 XYZ좌표계시스템에서 정확한 이동 동작을 실행하기 위해 각 축별 위치파라메터 설정에 대해서 설명하며 아래의 순서대로 설정조작을 실행합니다.

(1) 각 축의 이동방향 설정
(2) 모션보드의 축 맵핑
(3) 1펄스 단위 이동량 설정
(4) 전자기어비 설정
(5) 소프트리미트, 하드리미트 설정에 의한 동작한계치 정의
(6) 백래쉬 설정

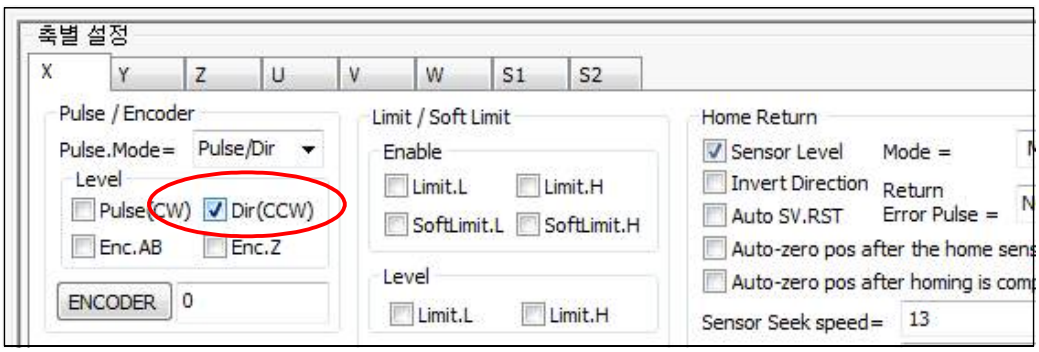
4.2.1. 각 축의 이동방향 설정

각 축이 단위 이동량 설정이 완료되었다면 각 축에 작은 펄스를 발생시키고 이동방향을 동작에 맞게 교정해줍니다. 이때 축펄스신호가 1펄스방식 즉 이동펄스(Pulse)와 방향신호(Dir)로 구성되었을 경우는 엔진구성화면의 방향신호만을 반전시켜 간단하게 교정될 수 있습니다. 그러나 2펄스 방식 즉 CW펄스신호와 CCW펄스신호 방식을 사용할 경우 배선을 바꾸어주거나 서보드라이브의 경우 동작방향 반전 파라메터를 설정해서 바꾸어 주어야 합니다.

우선 각 축의 전기적신호연결이 완료된 상태임을 가정합니다.
아래는 “모니터링” 화면을 통해서 간단하게 각 축에 소량의 이동펄스를 발생시키는 방법 소개합니다.



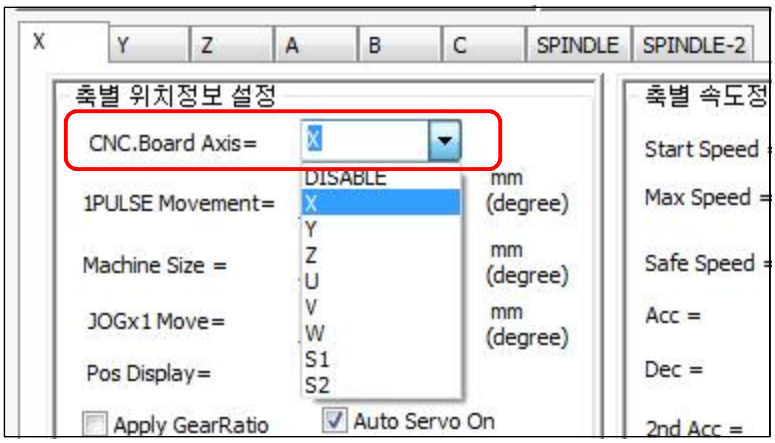
좌표증가방향의 버튼과 감소방향의 버튼에 눌러 기계움직임을 확인합니다. 만약 이동방향이 반대이면 신호체계에 맞는 방향전환 방식을 선택해야 합니다. 아래의 그림은 1펄스 방식에서 이동방향 전환시 해당하는 “엔진구성” 화면의 설정부분을 표시합니다.



4.2.2. 모션보드의 축맵핑

모션마스터 프로그램은 기본적으로 XYZ의 직교좌표계(Cartesian Coordinate System)를 속에서 모션의 움직임을 제어하게 됩니다. 실제 최대 6개의 좌표시스템으로써 일반적으로 X,Y,Z 위치3좌표와 A,B,C 방향3좌표로 구성되어 집니다. 이들 실세계좌표계시스템은 모션보드의 특정 제어축과 연결되어야 함으로 가장 먼저 해야하는 작업이 바로 모션보드와 CNC좌표계시스템의 축을 맵핑하는 작업입니다.

일반적인 모션보드는 제1축, 제2축, ...의 형태로 많이 규정되어지며, 위칸모션보드는 X,Y,Z,U,V,W,S1,S2 의 제어축명칭을 갖고 있습니다.
특수한 상황을 제외하고는 대부분 모션보드의 X축(혹은 제1축)을 CNC좌표계시스템의 X축으로 맵핑하고
모션보드의 Y축(혹은 제1축)을 CNC좌표계시스템의 Y축으로 순서적인 맵핑을 하게 될 것입니다.
맵핑방식은 아래의 CNC구성화면 그림을 참조하실 수 있습니다.
(단지 이동축이 아닌 회전자체를 목적으로 하는 스핀들축에 대한 맵핑은 특별하게 간주되어 처리됩니다. 스핀들 축에 대한 맵핑과 동작설정은 따로이 뒷절에서 살펴보겠습니다.)



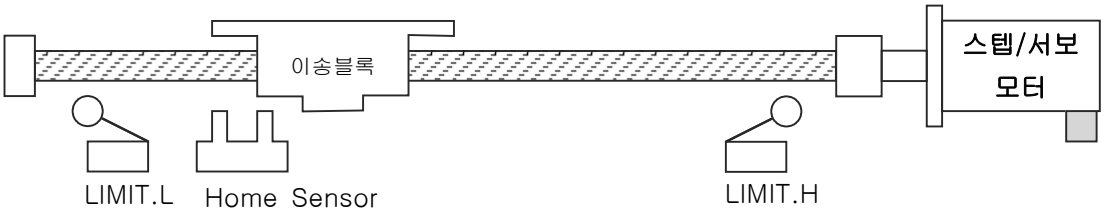
사용되지 않는 축은 DISABLE로 선택합니다.

이러한 축맵핑 혹은 축지정기능은 M코드를 통해서 실행될 수 있어 프로그램실행도중에 축지정의 교환함으로써 다양한 복합기능을 수행할 수 있습니다. 일례로 A축을 제2의 Y축으로 지정하여 마치 2개의 Y축이 존재하듯이 작업을 전환시킬 수도 있으며 서보드라이브와 같이 펄스입력을 받는 스핀들축일 경우 C축에 맵핑하여 다른 축과 함께 보간동작을 수행할 수도 있게 됩니다.

4.2.3. 1펄스 단위이동량과 전자기어비 설정

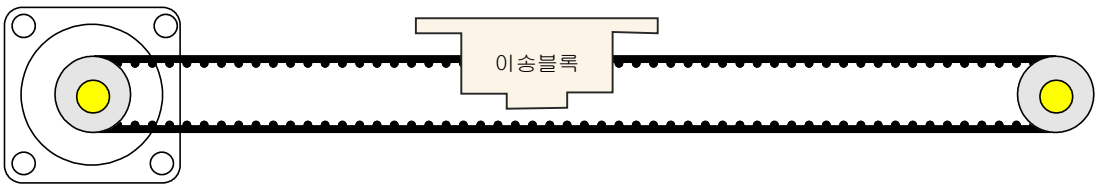
축이 맵핑된 후 가장 먼저 1펄스의 이동지령에 대해 기계가 실제로 움직이는 이동량을 mm 혹은 degree(각도) 단위로 설정합니다.

1펄스 단위이동량에 대한 수치값은 아래와 같이 간단하게 계산될 수 있습니다.
<리드 스크류 이동메카니즘의 예>



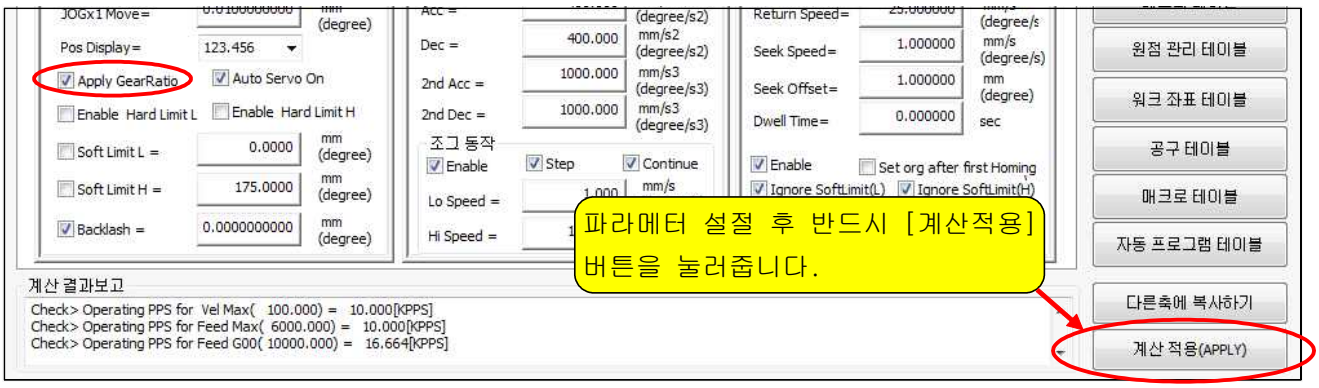
위와 같은 리드스크류타입의 이동시스템에서 서보모터의 1회전 펄스 분해능이 3200 (PPR :Pulse Per Rotation) 회전이며 리드스크류의 1회전 이동량이 5mm (Lead)일때 1펄스 단위이동량 U는
U = 5 mm / 3200 = 0.0015625 mm 입니다.

<타이밍벨트 이동메카니즘의 예>



타이밍풀리와 타이밍벨트 이동시스템의 경우 이동거리 산출이 정확하지 않을 수 있지만 이론적으로 타이밍풀리의 피치원직경을 기준으로 계산합니다. 모터의 1회전 분해능이 3200 (PPR)이고 피치원직경 Dp = 10.125mm 일때 단위펄스이동량 U = πx Dp / PPR = 3.141592654 *11.125 / 3200 = 0.010921943 mm 입니다.

각 축별로 정확하게 계산되어 수치를 입력합니다. 일반적으로 각 축마다 펄스 단위이동량의 비율이 1: 1 로 동일하도록 설정하는 것이 유리합니다. 만약 x축과 y축의 펄스단위이동량이 다르다면 전자기어비 설정사용을 고려해야 합니다. 위칸모션컨트롤러는 내부적으로 각축의 펄스출력비율을 조절하는 전자기어비기능을 갖추고 있습니다. 전자기어비 기능은 각 축의 펄스단위이동량이 서로 다르면서 해당축이 원호보간을 사용할 경우에 적용합니다. 원호보간은 기본적으로 각 축이 1:1 비율로 움직인다는 가정하에서 동작합니다. 그러므로 X, Y축의 펄스단위이동량이 서로 다르고 원호보간동작을 필요하다면 아래와 같이 각각 X축 Y축에 대해 전자기어비 기능을 활성화 시켜줍니다.
<CNC 구성화면>

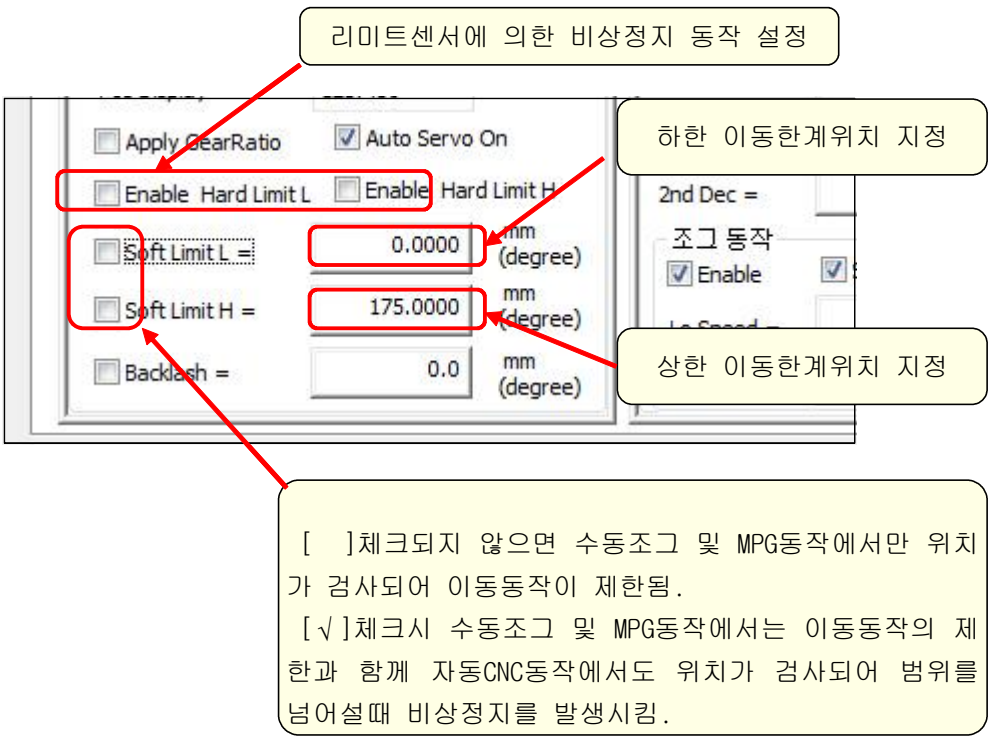


위칸모션컨트롤러의 전자기어비 기능을 활성화 할 경우 전체적인 펄스출력속도의 제한치가 약 20%정도 낮아지게 됩니다. 서보모터 드라이브의 경우 자체적으로 전자기어비의 기능이 있어 각 축별 단위이동거리가 1:1의 비율되도록 맞추어 주는 것이 바람직합니다.
위칸모션컨트롤러의 전자기어비 기능은 스텝모터드라이브와 같이 전자기어비의 기능을 갖추지 않으면서, 서로 다른 축이동비율로써 원호보간을 수행하는 기구들에 대해서 유용합니다. 또한 전자기어비 기능을 활용할 때 상호 축 이동비율이 1:2 혹은 1:3이하의 비율에서 설정하는 것이 전체적인 동작성능에 유리합니다.

모션컨트롤러의 전자기어비설정시 가장 분해능이 높은 축을 기준으로 다른 축의 펄스발생연산이 수행되기 때문입니다. 예를 들면 X축과 Y축의 펄스단위 이동량의 비율이 1:10 일때 Y축의 이동분해능이 1축보다 10배가 크다면 Y축은 최대펄스발생주파수의 1/10 속도가 한계지점이 됩니다. 물론 이러한 전자기어비 설정은 원호보간을 사용하지 않는 축에 대해서는 설정할 필요가 없습니다.

4.2.4. 소프트리미트, 하드리미트에 의한 동작한계치 설정

“CNC 구성” 화면의 소프트리미트(SoftLimit) 설정은 원점복귀 동작 후 실제 기계위치와 와 프로그램의 좌표가 매칭 되었을때 CNC 명령에 의한 좌표이동한계를 수치적으로 제한하는 기능을 갖고 있습니다. 이것은 이동한계를 전기기계적 스위치에 의한 하드리미트 방식과 비교됩니다. 소프트리미트는 반드시 원점복귀동작을 수행하거나 임의적으로 현재 기계위치를 지정한 후 바르게 동작되지만 하드리미트의 경우 현재 기계위치에 관계없이 리미트센서가 작동하면 알람이 발생하고 정지하게 됩니다.

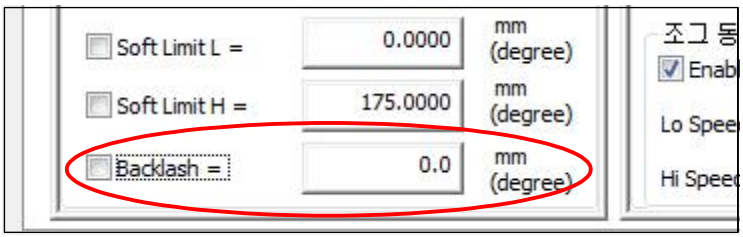


4.2.5. 백래쉬

각 축의 이동 메카지즘 특성에 따라 회전방향을 바꿀 때 일정한 영역까지 이동이 되지 않는 불감대영역이 존재할 수 있습니다. 이러한 특성을 가질 경우 방향전환시 일정거리를 더 움직여주도록 설정해 주는 부분이 백래쉬 입니다. 백래쉬의 크기 단위는 각 축의 펄스이동단위량과 동일합니다. 항상 0에서 양의 실수를 사용합니다.

백래쉬에 대한 펄스의 출력속도는 현재 모션동작의 펄스출력속도와 동일합니다. 스텝모터의 경우 백래쉬량을 크게 설정하면 모터회전의 JERK(충격)가 발생될 수 도 있습니다. 서보모터의 경우 백래쉬펄스는 공회전과 같은 저관성영역에서 흡수되므로 현재 모션동작의 속도 저하없이 백래쉬 동작을 실행할 수 있습니다.

<CNC 구성 화면>

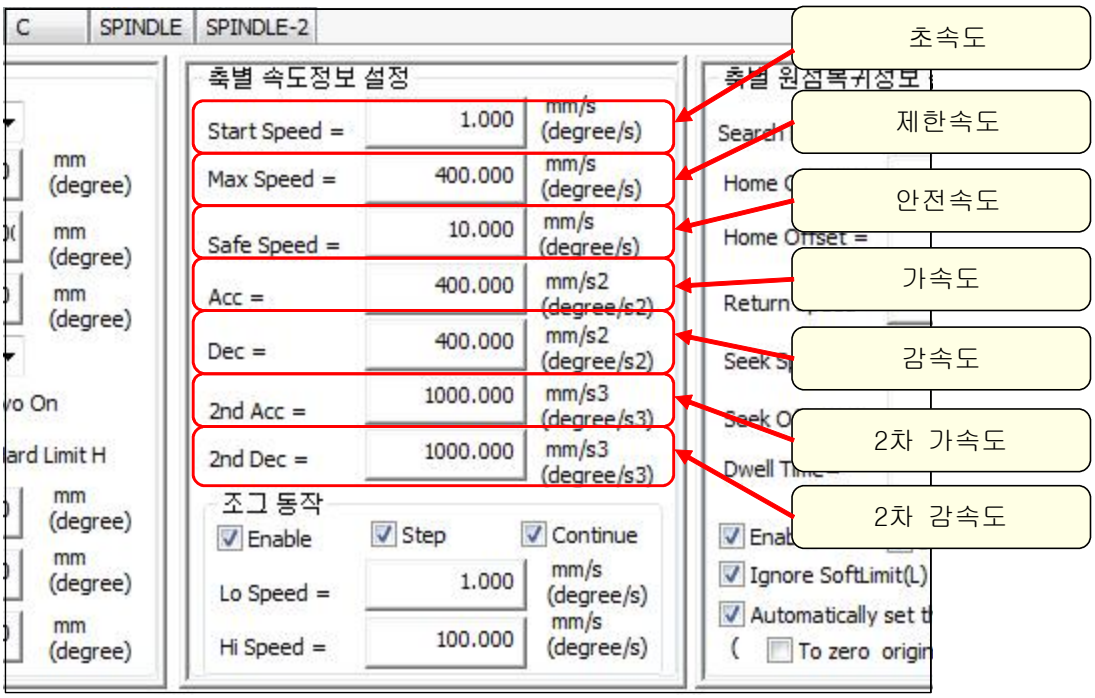


4.3. 수동 JOG동작을 위한 기초 설정

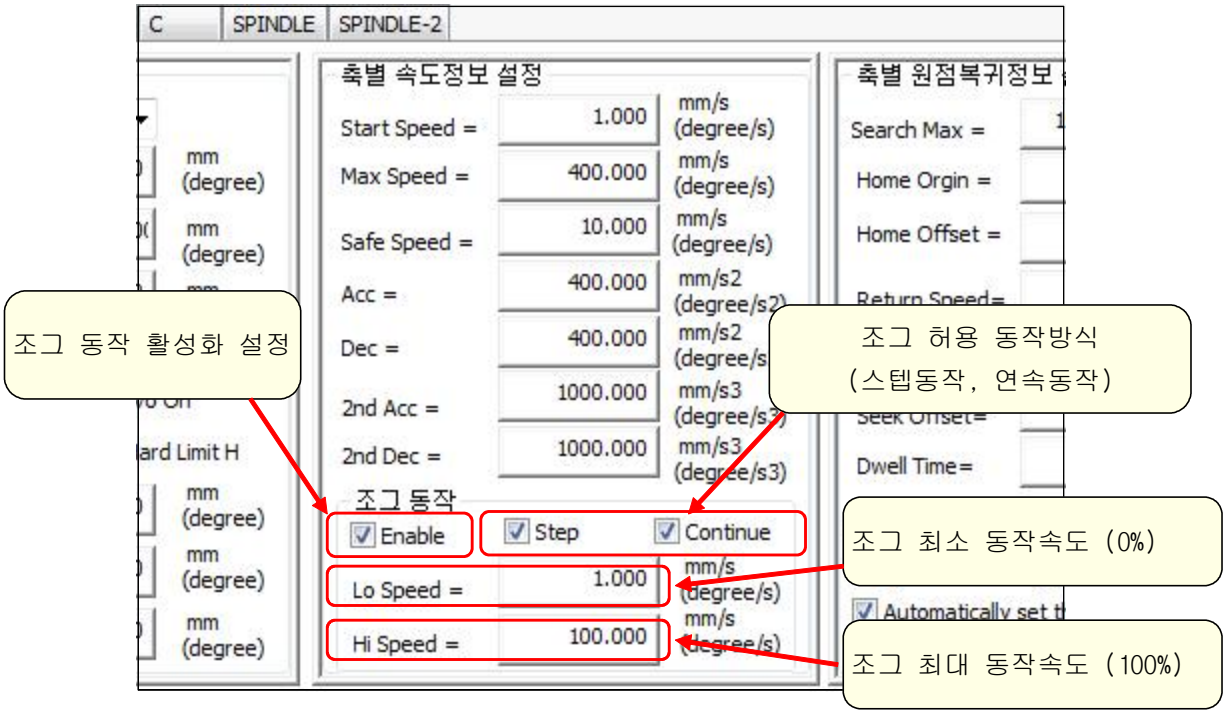
상기 위치제어 파라메터와 함께 각 축별 기본 속도제어 파라메터 및 조그동작 특성파라메터의 설정을 통해 수동조그동작을 실행할 수 있게 됩니다. CNC수동조작이 가능해 지면 수동움직임을 통해 홈센서, 리미트센서들의 신호감지 상태등 CNC기계장치의 움직임과 관련된 센서장치의 동작을 정확히 확인할 수 있게 됩니다.

먼저 각 축별 다양한 속도제어 파라메터들을 설정해봅니다.
기계시스템은 각 축마다의 동특성이 달라서 특정축은 빠른 움직임을 허용하지만 또 다른 축은 제한된 속도와 가속도 범위내에서 움직여야 할 필요가 있습니다.
속도제어파라메터는 크게 아래와 같이 총7가지입니다. 각 파라메터의 기능에 대해 간략히 기술해봅니다.
(1) 초속도 : 각 축의 이동 동작이 시작되는 속도
(2) 제한속도 : 각 축의 최대 허용 동작속도
(3) 안전속도 : 임시정지상태 등 특정상황에서 원래위치로 복귀하는 안전동작 속도
(4) 가속도 : 각 축의 허용 가속도계수
(5) 감속도 : 각 축의 허용 감속도계수
(6) 2차 가속도 : S자커브와 같은 2차 가감속 동작방식의 2차 가속계수
(7) 2차 감속도 :S자커브와 2차 가감속 동작방식의 2차 감속계수

이러한 각 축별 동작속도와 가속도 설정의 단위는 각각 mm/sec 혹 mm/sec² 입니다.
특히 가속도는 그 축이 낼 수 있는 추력(힘)에 대한 이동블록의 관성특성이므로 움직임에 무리가 없는 값을 선택해 주어야 합니다. 대체로 움직임이 이 둔할 경우 가속도값은 100~200 이하의 값을 가지며, 보다 빠른 움직임 경우 200~400 이상의 값을 취할 수 있습니다. 이러한 범위를 참조하여 자신의 기계시스템의 힘에 맞게 설정합니다.

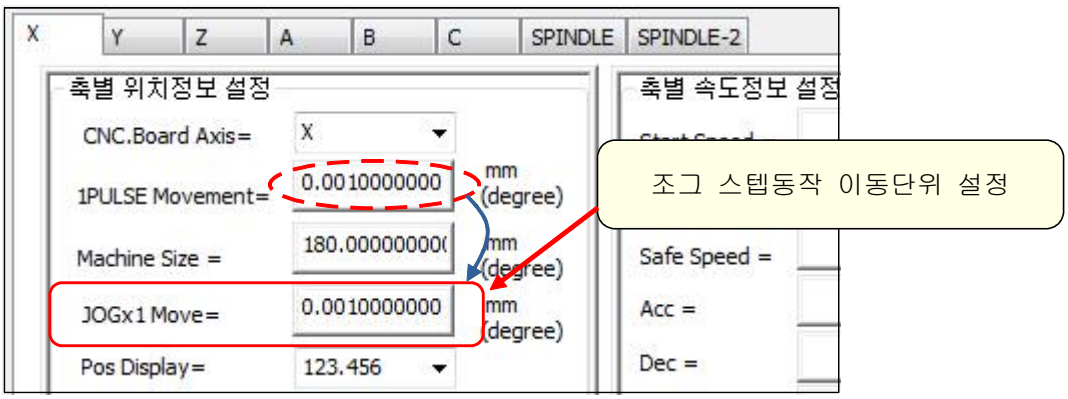


기본적인 각 축의 속도파라메터가 지정되었다면 다음으로 조그동작에서 사용될 속도영역과 동작방식을 설정합니다.
조그 동작은 키보드나 마우스의 클릭에 의해서도 동작될 수 있으며, 추후 모션보드의 GIO입출력(일반입출력)신호에 의해서도 동작될 수 있는 유용한 CNC동작방식입니다.
여기서는 먼저 조그동작을 활성화시키고, 조그동작의 종류와 조그동작 속도영역 등 조그 동작에 필요한 기본적인 내용 설정만을 간략히 설명합니다.



조그동작은 스텝동작과 연속동작으로 구분되는 2가지 작동방식이 있습니다. 키보드에 의한 동작 예로써 설명하면
스텝동작은 키입력이 발생할때마다 지정된 일정거리단위를 한스텝씩 이동하는 동작입니다.
연속동작은 키가 눌러질때 지정된 속도로 움직이고 키가 떨어질때 정지하는 동작입니다.
스텝동작은 특정위치 근방에서 이동스텝단위로 보다 정확하게 이동할 때 사용되며 연속동작은 특정근방의 위치까지 빠르게 이동할 때 사용됩니다.

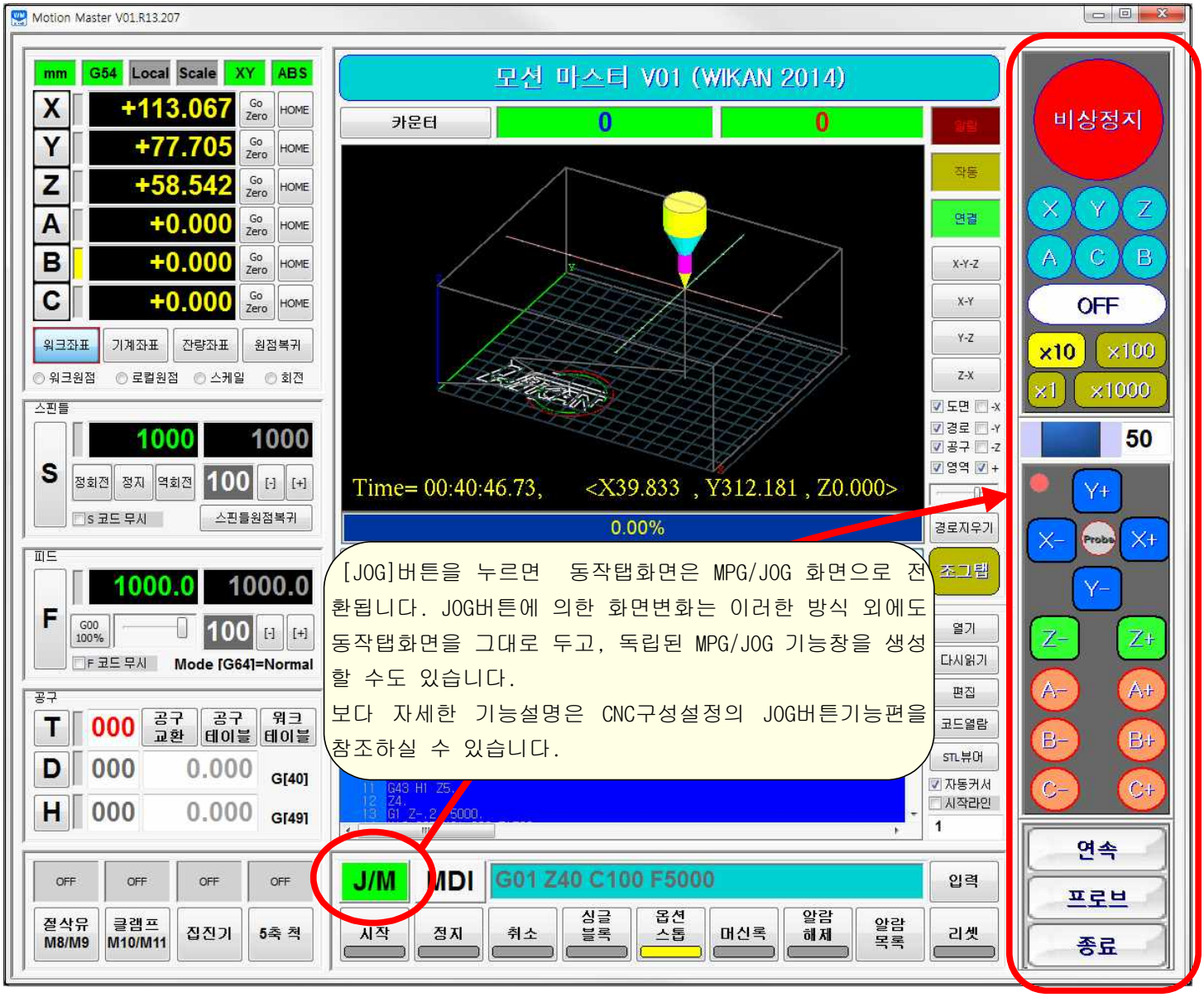
스텝동작에서의 이동단위는 "CNC 구성 “화면의 각 축마다 JOGx1 Move= 항목으로 설정되며 대부분 1펄스 단위이동량과 동일하게 설정합니다.



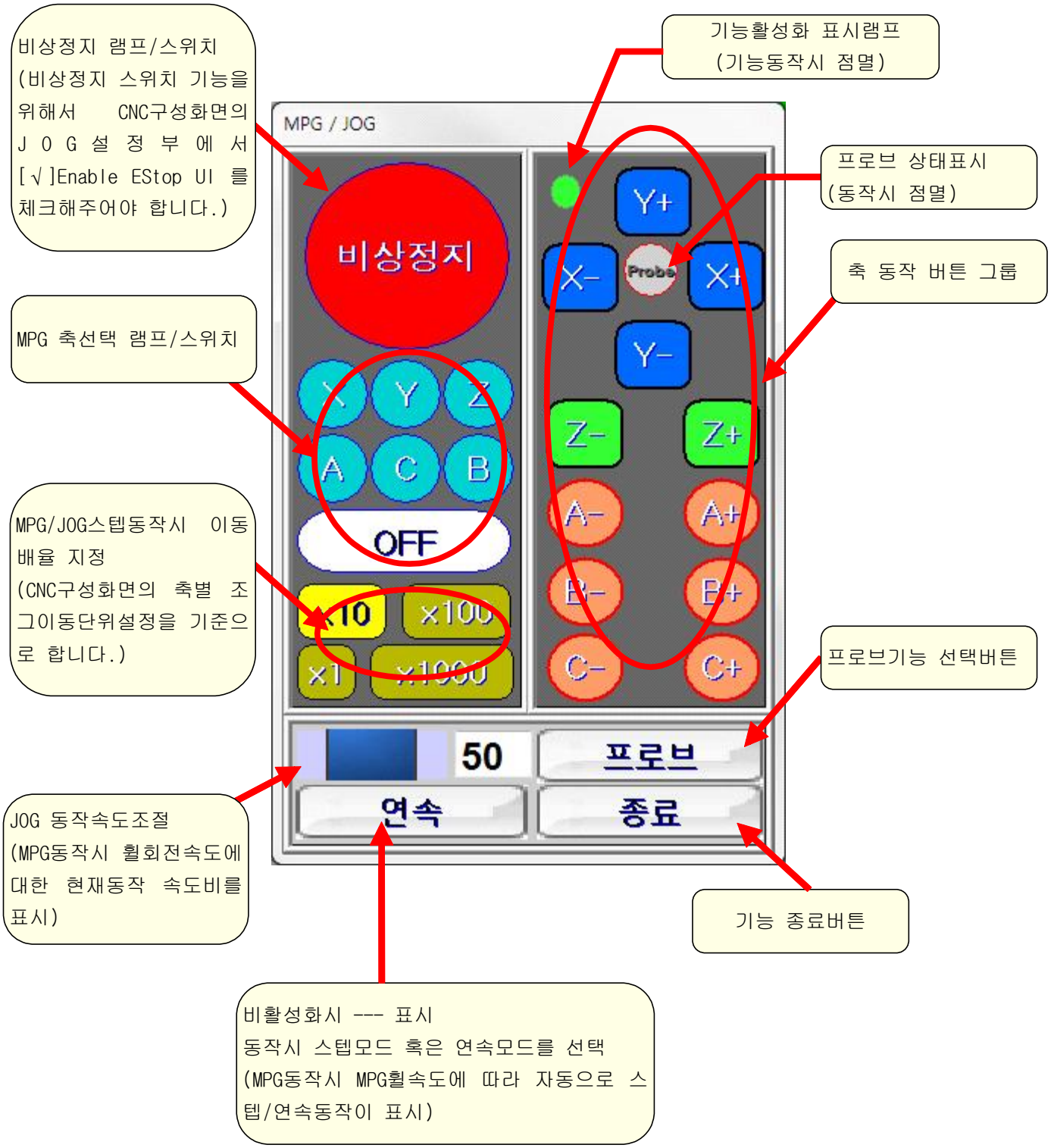
4.4. JOG 조작에 의한 축별 IO신호 점검

CNC 장치의 JOG 수동조작을 위한 기초적인 설정이 완료되었다면 CNC 운전화면에서 JOG동작을 아래의 절차에 따라 실행해 봅니다.

(1) CNC 운전화면에서 JOG버튼 실행



(2) 조그버튼에 의한 화면변화는 몇 가지 방식에 의해 동작될 수 있지만 전체적인 기능은 동일합니다.
아래 MPG/JOG 화면은 Jog스타일 2번에 해당하는 화면입니다.
(MPG/JOG화면은 MPG기능과 JOG기능이 혼용되어 있는 화면입니다.)



(3) JOG/MPG 화면을 이용하여 움직일 경우 마우스의 클릭동작에 의존하게 됩니다. 모니터가 터치방식의 입력을 지원한다면 터치에 의해 동작될 것입니다.
그러나 현재 JOG/MPG 기능이 활성화 된 상태라면 아래와 같이 기본적인 키보드버튼들과 몇가지 기능들이 맵핑되어 있어 간단하게 키보드 동작으로 조그 동작을 실행할 수 있게 됩니다.

- 백스페이스키 = 조그 모드 (스텝/연속) 전환
- / 키 = 조그 속도비 감소
- * 키 = 조그 속도비 증가
- - 키 = 조그 이동배율 감소 (단, MPG의 스위치 등에 의해 배율조절이 강제 지정되어 있지 않을 때만)
- + 키 = 조그 이동배율 감소 (단, MPG의 스위치 등에 의해 배율조절이 강제 지정되어 있지 않을 때만)
- 위쪽 화살표 키 = Y축 좌표증가 이동
- 아래쪽 화살표 키 = Y축 좌표감소 이동
- 왼쪽 화살표 키 = X축 좌표감소 이동
- 오른쪽 화살표 키 = X축 좌표증가 이동
- Page Down 키 = Z축 좌표감소 이동
- Page Up 키 = Z축 좌표증가 이동

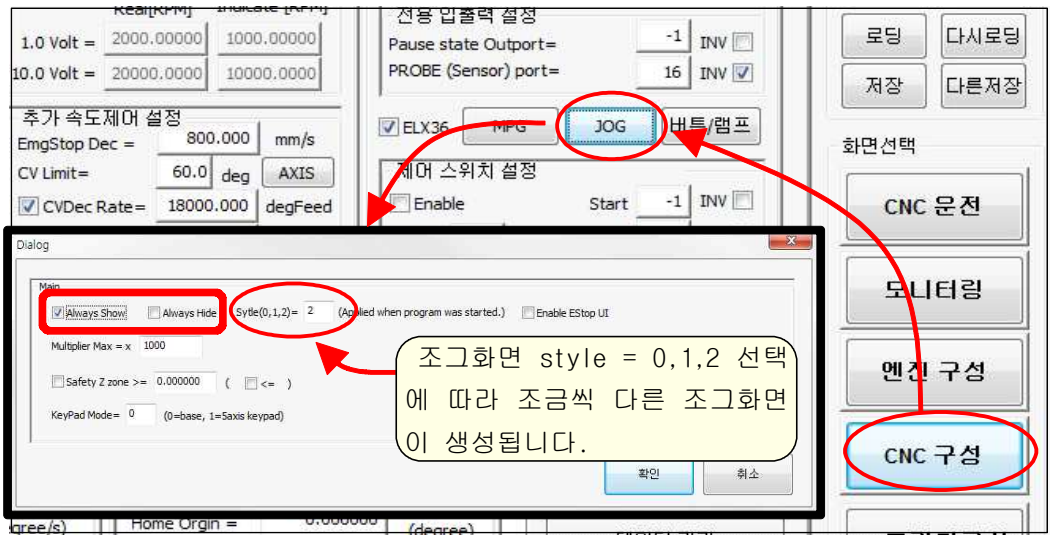
(상기 키의 기능은 동일하지만 CNC구성화면읜 [JOG]설정화면의 KeyPad Mode 지정값에 따라 A축, B축을 직접적으로 동작시킬 수 있는 키 맵핑이 추가됩니다.)

(4) 아래의 절차대로 이동 동작을 테스트합니다.
먼저 스텝이동모드로 동작해 봅니다.
- 조그 동작속도를 10%이하의 비교적 작은 속도범위로 조절합니다.
- 조그 모드를 스텝모드로 전환합니다.
- 조그 이동배율을 x10 배율로 선택합니다.
- 각 축의 조그 동작버튼을 눌러봅니다.

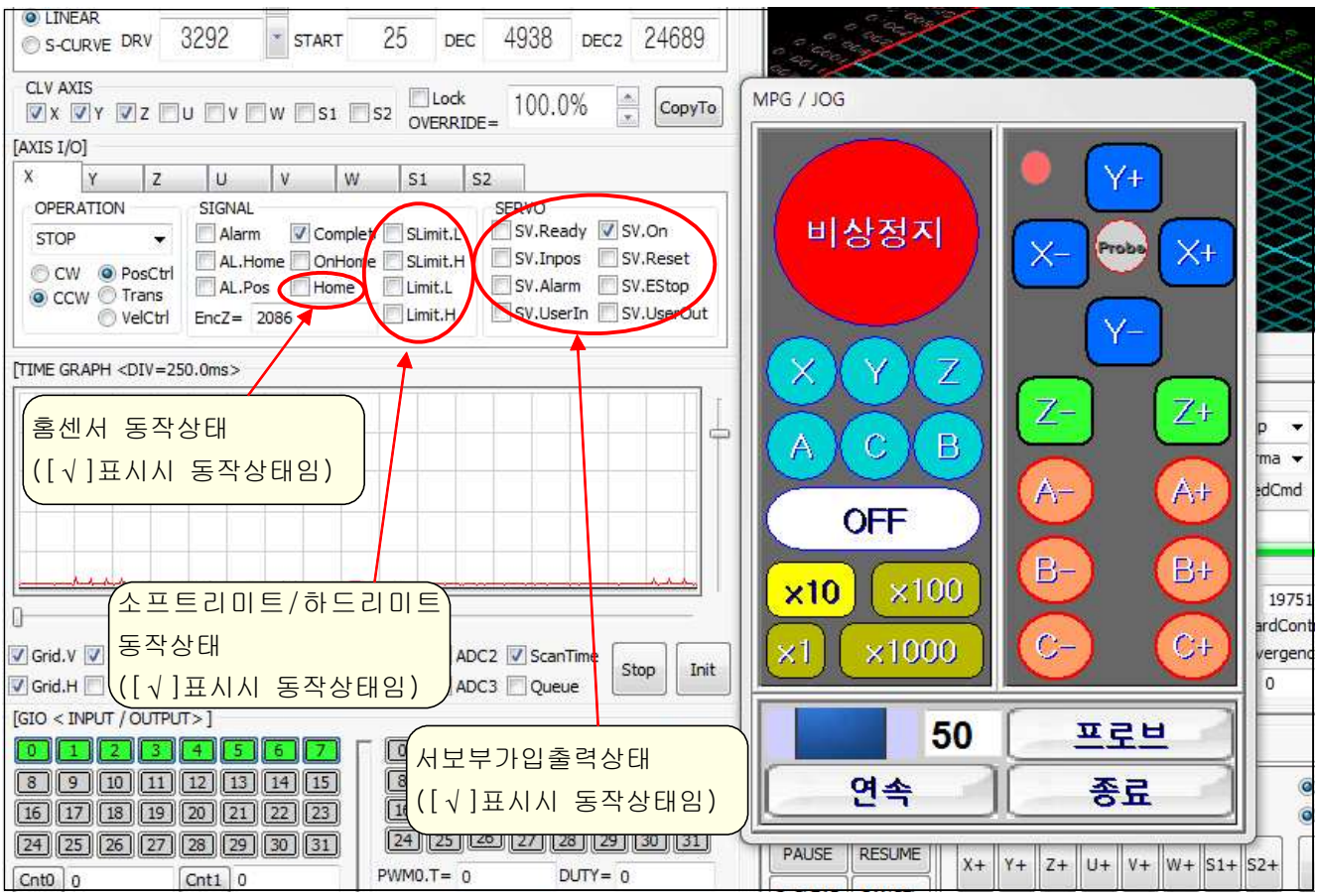
각 축의 조그 동작버튼 눌러주어 한 스텝식 이동시켜보며 이동방향과 좌표증감방향이 동일한지 한 번 더 확인해 봅니다.

다음으로 연속이동 동작을 테스트합니다. 연속이동 동작은 조그 모드를 연속모드로 전환하여 조그 동작버튼을 눌러봅니다. 조그 동작버튼을 누르면 이동하고 떨어지면 정지하게 됩니다.

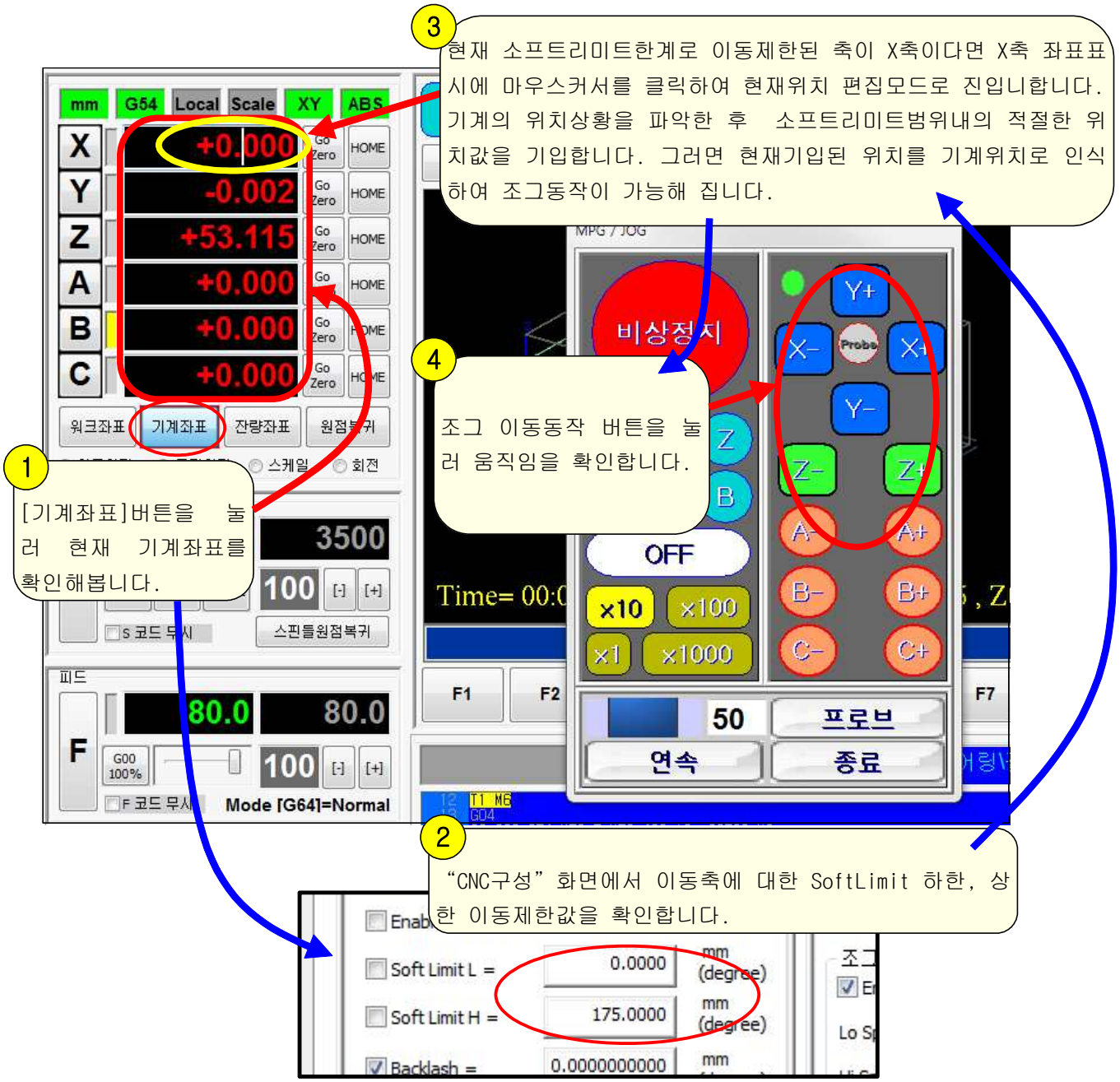
(5) 조그동작을 이용하여 축이동과 관련된 홈센서와 리미트 센서의 동작을 확인해 봅니다.
홈센서와 리미트센서는 축의 이동에 의해 동작감지됨으로 수동조그동작을 통해서 감지동작이 정확한지를 확인합니다.
각 센서들의 입출력상태를 확인하는 기본적인 화면은 “모니터링” 화면입니다. 그러나 조그동작을 활성화 시킬 경우 옵션에 따라서 동작탭의 화면인 조그탭전용화면으로 바뀌어서 모니터링화면으로 전환할 수 없게 됩니다. 이러한 경우 독립된 [MPG/JOG]화면을 띄워서 조그기능을 실행하면 다른 동작화면 전환시에도 조그기능을 유지한채 사용될 수 있습니다. 독립된 [MPG/JOG]화면을 띄우기 위해서 아래의 “CNC구성” 설정화면의 [JOG]버튼을 눌러 [√]Always Show 를 체크하시고 [ ]Always Hide는 체크를 해제합니다.



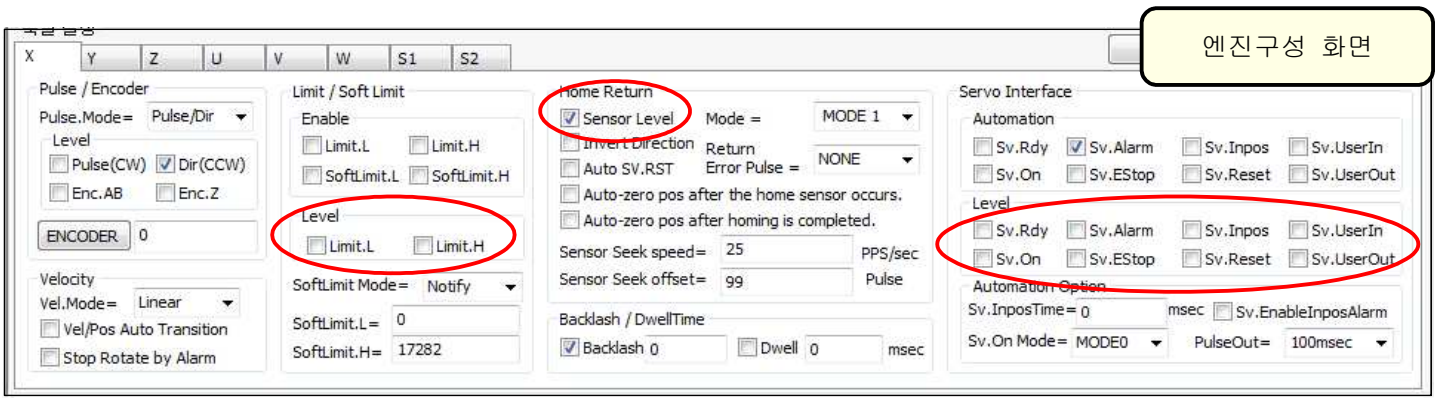
조그를 천천히 이동시키면사 아래의 모니터링 화면에서 각 축별 홈센서와 리미트센서 및 서보부가입출력신호들의의 동작상황(체크로표시)을 확인합니다.



만약 조그이동버튼을 눌러도 기계의 움직임이 없다면 가장 먼저 현재 기계위차가 소프트리미트에 의한 이동한계값에 의해 제한된 것이지 아래의 그림절차를 통해 확인합니다.



각 축의 홈센서 등의 신호레벨이 반대로 되어 있을 경우 "엔진구성" 화면의 해당 축 IO별 Level 항목을 체크해 주어 동작 활성화 신호상태 반대로 설정할 수 있습니다.



4.5. 원점복귀 파라메터 설정

축의 동작과 원점복귀 센서동작이 확인되었다면 원점복귀동작이 안전하게 수행되기 위한 파라메터를 설정해야 합니다. 대부분 CNC기계는 전원이 인가된 후 가장 먼저 원점복귀 동작을 수행하는 것이 일반적입니다. 원점복귀 동작은 기계의 실제위치와 프로그램의 좌표위치를 일치시키기 위한 동작입니다. 원점복귀가 정상적으로 완료되어야만 수동조그동작등에서 움직임을 안전하게 제한시키는 소프트리미트 동작이 올바르게 작동하게 됩니다.

이제 원점복귀 동작을 올바르게 실행시키기 위한 주요 파라메터설정들을 간략히 살펴보고, 원점복귀 동작을 실행시켜봅니다. **※보다 자세한 원점복귀 동작의 내용과 파라메터 설명은 "CNC 구성 화면" 장의 축별원점복귀 파라메터설정편을 참조해 주십시오.**

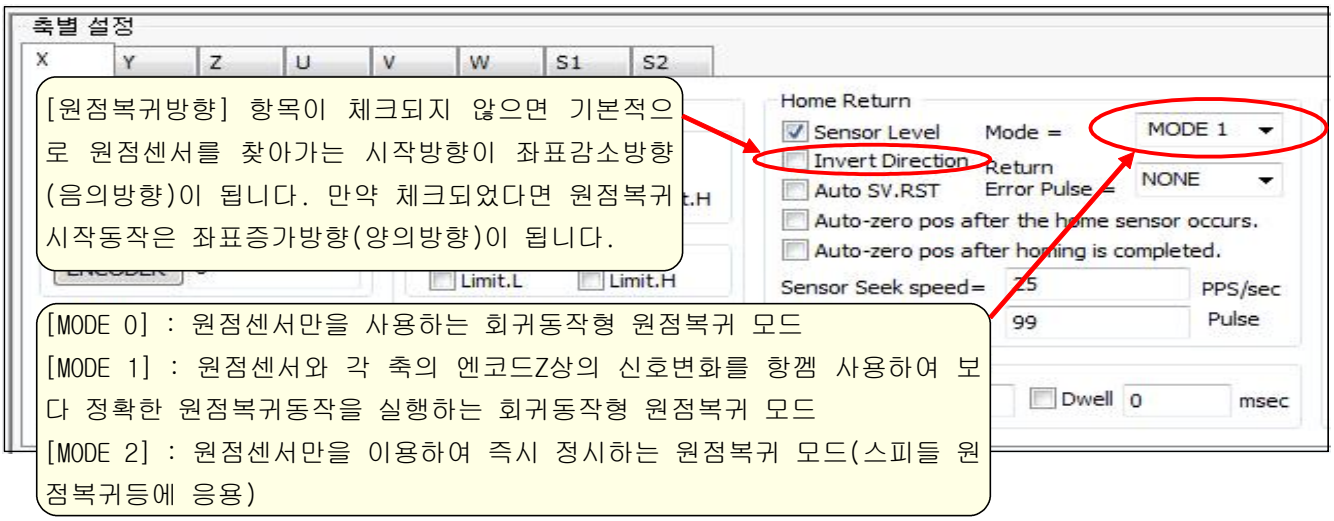
안전한 원점복귀 동작을 위해 확인하고 설정해야 할 사항들을 나열하면 아래와 같습니다.

(1) 축이동 동작 확인
(2) 원점센서 동작확인
(3) 원점복귀 동작방향 설정
(4) 원점복귀 동작모드 설정
(5) 원점복귀 동작 활성화시키기
(6) 원점복귀 동작의 최대이동거리 설정
(7) 원점복귀 동작 완료 후 기계정지 위치설정
(8) 원점복귀 동작 후 옵셋(시프트)이동량 설정
(9) 원점복귀 동작 속도 설정
(10) 원점복귀 센서 탐색속도 설정
(11) 원점복귀 동작시 소프트리미트 동작방식 설정
(12) 원점복귀 동작완료 후 자동 좌표계시스템의 셋팅방식 설정

상기 동작확인 및 설정절차 중 (1)항과 (2)항은 상기절을 실행을 통해 이미 완료된 것으로 가정합니다. 먼저 상기 절차대로 파라메터를 설정한 후 개별적인 축의 원점복귀 동작을 실행하고 전체 원점복귀 동작을 실행하는 순서로 테스트합니다. 먼저 아래의 그림절차대로 원점복귀 동작을 위한 파라메터를 설정해 봅니다.

4.5.1. 엔진구성 화면에서 동작방향과 동작모드 설정

원점복귀방향은 "엔진구성" 화면 축별설정의 HomeReturn 그룹의 [ ]Invert Direction 항목을 체크해줌으로써 동작이 반전됩니다.



4.5.2. CNC구성 화면에서 원점복귀동작 주요 파라메터 설정

기초적인 원점복귀동작 방향과 동작모드가 설정되었다면 그외 주요한 파라메터설정들은 CNC 구성화면의 축별 원점복귀정보 설정그룹에서 다양한 기능들을 설정할 수 있습니다. 아래 그림의 절차대로 파라메터를 설정해 봅니다.

2 [원점복귀동작 최대이동거리] : 원점복귀 동작실행의 최대이동거리를 설정합니다. 원점복귀 동작에서 소요된 거리가 이 설정값을 초과하면 원점복귀동작 알람이 발생합니다.

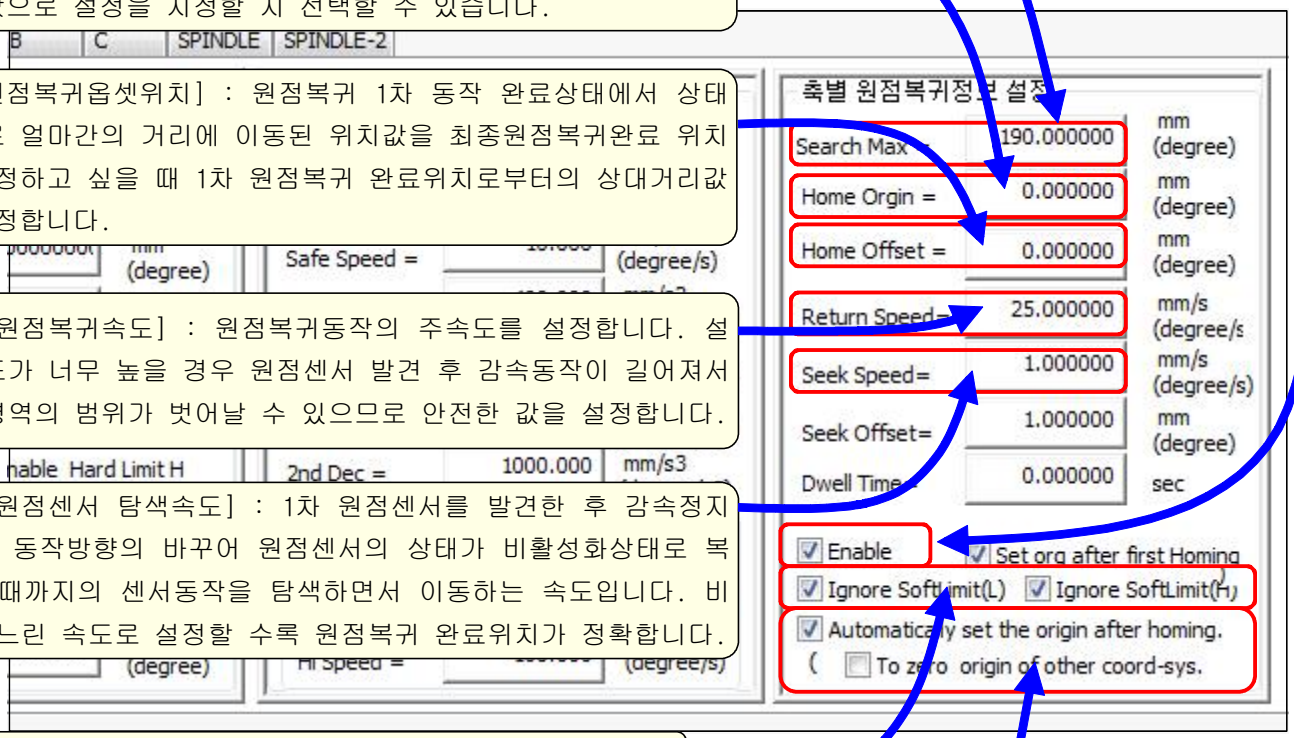
1 [원점복귀 활성화] : 원점복귀 동작을 가능하게 합니다.

3 [원점복귀 기계위치] : 원점복귀동작 완료 후 원점기계위치값을 설정합니다. 일반적으로 원점복귀가 완료되면 여기에 지정된 값으로 기계좌표가 설정되게 됩니다. 모션마스터는 자동으로 이 좌표값으로 설정을 지정할 지 선택할 수 있습니다.

4 [원점복귀옵셋위치] : 원점복귀 1차 동작 완료상태에서 상태적으로 얼마간의 거리에 이동된 위치값을 최종원점복귀완료 위치로 지정하고 싶을 때 1차 원점복귀 완료위치로부터의 상대거리값을 설정합니다.

5 [원점복귀속도] : 원점복귀동작의 주속도를 설정합니다. 설정속도가 너무 높을 경우 원점센서 발견 후 감속동작이 길어져서 이동영역의 범위가 벗어날 수 있으므로 안전한 값을 설정합니다.

6 [원점센서 탐색속도] : 1차 원점센서를 발견한 후 감속정지한 후 동작방향의 바꾸어 원점센서의 상태가 비활성화상태로 복구될 때까지의 센서동작을 탐색하면서 이동하는 속도입니다. 비교적 느린 속도로 설정할 수록 원점복귀 완료위치가 정확합니다.



7 [소프트리미트 비활성화] : 원점복귀 동작시 소프트리미트동작을 비활성화 시킵니다.

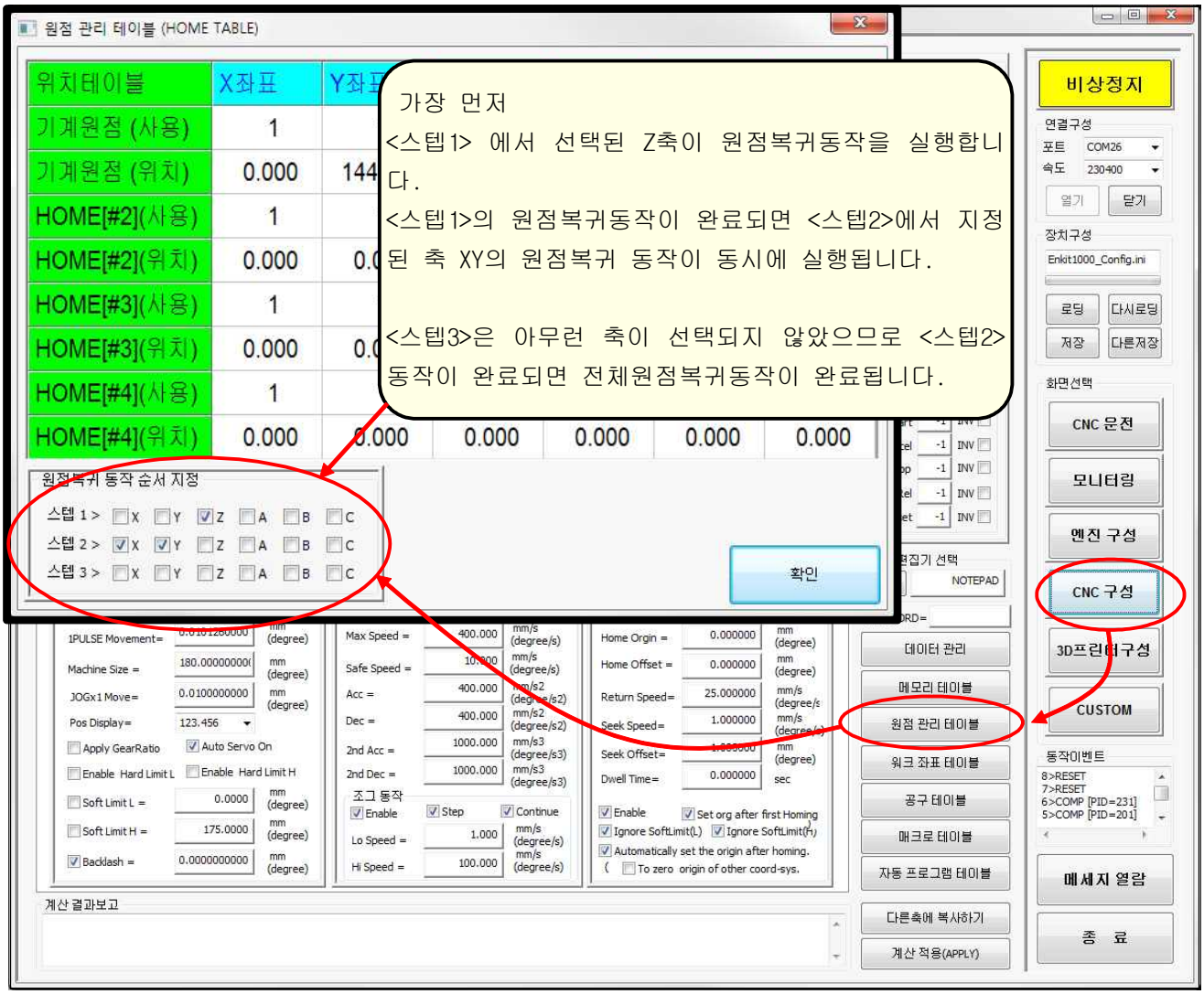
8 [좌표계 자동셋팅 설정] : 원점복귀 동작 완료 후 자동으로 좌표계를 설정하는 방식들을 설정합니다.
여기서는 [√] Automaticlaly set the orgin 만 선택합니다.

4.5.3. 개별 축 원점복귀 동작 실행

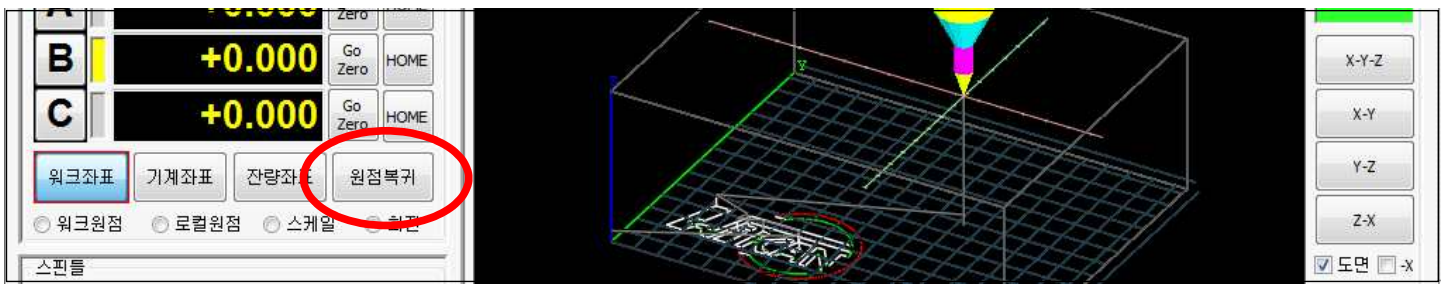
원점복귀의 주요한 파라메터를 설정한 후 CNC운전화면에서 축별 원점복귀 명령을 아래의 절차대로 실행해 봅니다. 처음 실행할 때에는 원점복귀 속도를 낮게 설정하여 이상 동작 발생시 [비상정지]버튼을 눌러 안전하게 정지시킬 수 있도록 합니다. 원점복귀시 비상정지 시키면 반드시 [리셋버]튼을 울러 재 초기화 시켜줍니다.

4.5.4. 시퀀스 동작을 갖는 전체 원점복귀 동작 실행

각 축별 원점복귀 동작은 주로 기계 초기 설정시에 유용합니다. 매번 기계사용시 각 축별로 원점복귀 동작을 순서별로 실행시킬 필요없이, 전체 원점복귀 동작을 통해서 지정된 순서로 원점복귀를 실행할 수 있습니다.
전체원점복귀동작을 위해서 먼저 아래의 "CNC구성 "화면의 원점관리테이블에서 3가지 스텝별 원점복귀동작 축을 순서대로 설정합니다.

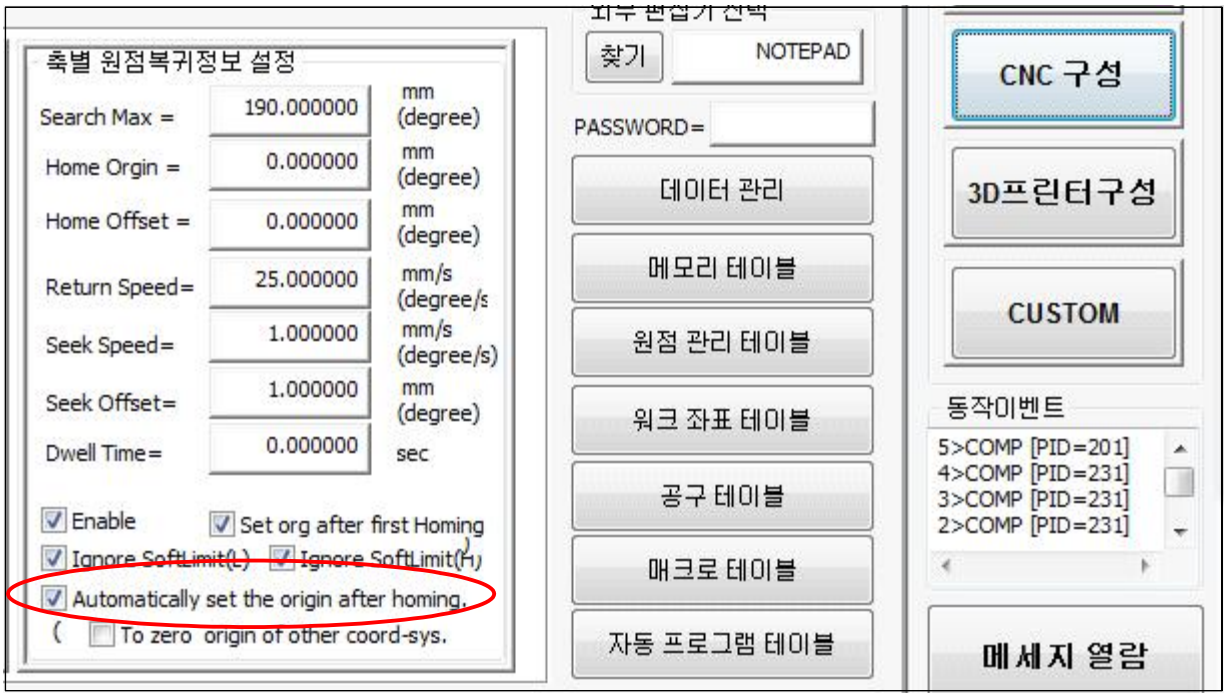


설정이 완료되면 "CNC운전" 화면의 [원점복귀]버튼을 눌러줍니다. 동작 중 이상이 발생하면 [비상정지]버튼에 의해 원점복귀동작을 정지시킨 후 [리셋]버튼을 눌러주어 초기화 시키고 확인 후 재시도 합니다.

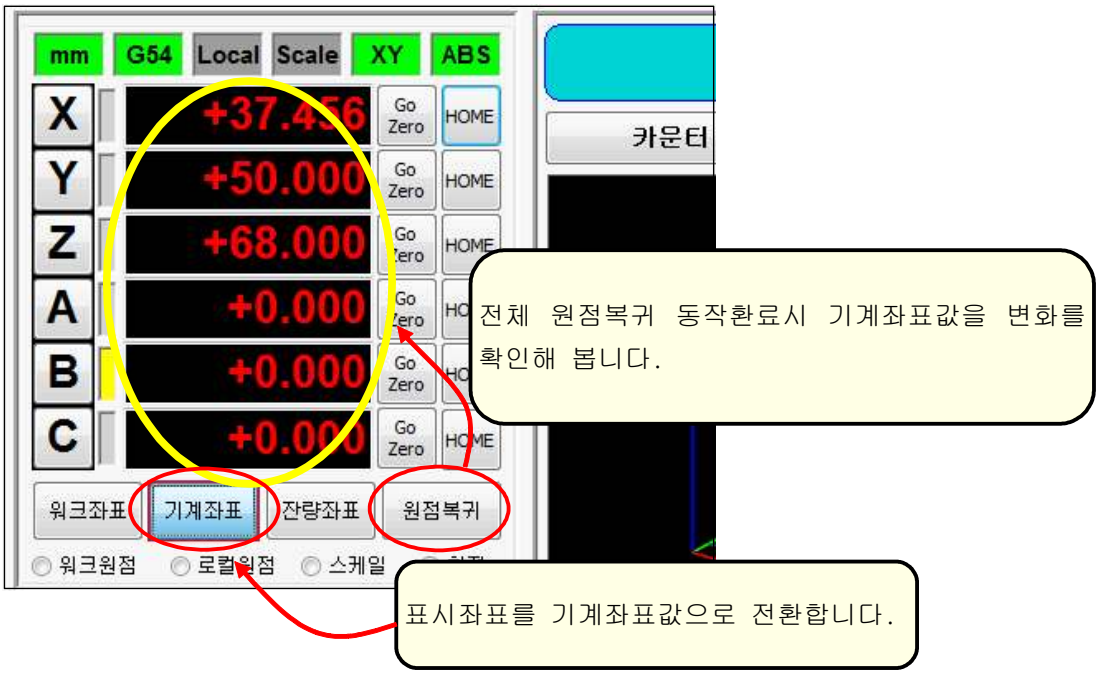


4.5.5. 원점복귀 동작 반복위치 정밀도 검사

현재 원점복귀 동작 구성설정은 원점복귀 동작이 완료되면 강제로 특정 기계좌표값을 설정하도록 구성되어 있습니다. 이러한 동작은 "CNC구성“화면의 ”축별원점복귀정보 설정“그룹의 [ ]Automatically set the orign after homming 이라는 항목을 체크했기 때문입니다.



원점복귀 동작이 정상적으로 완료되었다면 상기 항목의 체크를 없앤 후 "CNC운전“화면에서 전체원점복귀 동작을 반복 실행하여 원점복귀동작완료 후 반복된 좌표값의 변화를 확인해 볼 수 있습니다. 이로써 홈센서의 반응속도에 맞게 보다 정밀한 원점복귀 동작을 수행할 수 있도록 원점복귀 속도와 홈센서 탐색속도등을 조정할 수 있습니다. 또한 서보드라이브와 같이 엔코더 Z상을 지원할 경우 ”엔진구성“화면의 Home Return그룹에서 원점복귀모드를 1로 바꾸어 Z상에 의한 보다 정확한 원점복귀 동작을 시도해 볼 수 있습니다.



“

4.6. CNC 속도 동작제어 파라메터 설정

모션마스터프로그램에서 운영되는 CNC 제어장치는 기본적으로 G코드체계에서 동작됩니다. 모션제어는 정확한 위치 및 경로와 더불어 원하는 위치 및 경로로 바르게 이동하기 위해 속도제어를 포함하게 됩니다. G코드 체계에서는 F 혹은 S어드레스코드로써 단순하게 속도 동작특성을 설정할 수 있습니다. 이러한 간단한 속도지령은 사용자에게 많은 편의를 주지만, 기계입장에서는 이들 코드만으로는 움직임을 생성하기에 정보가 부족하게 됩니다.

기계움직임을 사용자 입장에서 고려하면 절삭, 커팅, 용접 등 가공모재에 특정 작업(효과)을 가하는 움직임이 있고, 다른 하나는 목적하는 작업위치로 급속이송하는 움직임이 있습니다. 급속이송은 목적하는 작업을 위한 준비이동에 해당합니다. 마지막으로 스핀들과 같은 주축으로 불리는 작업축의 회전이 있습니다.

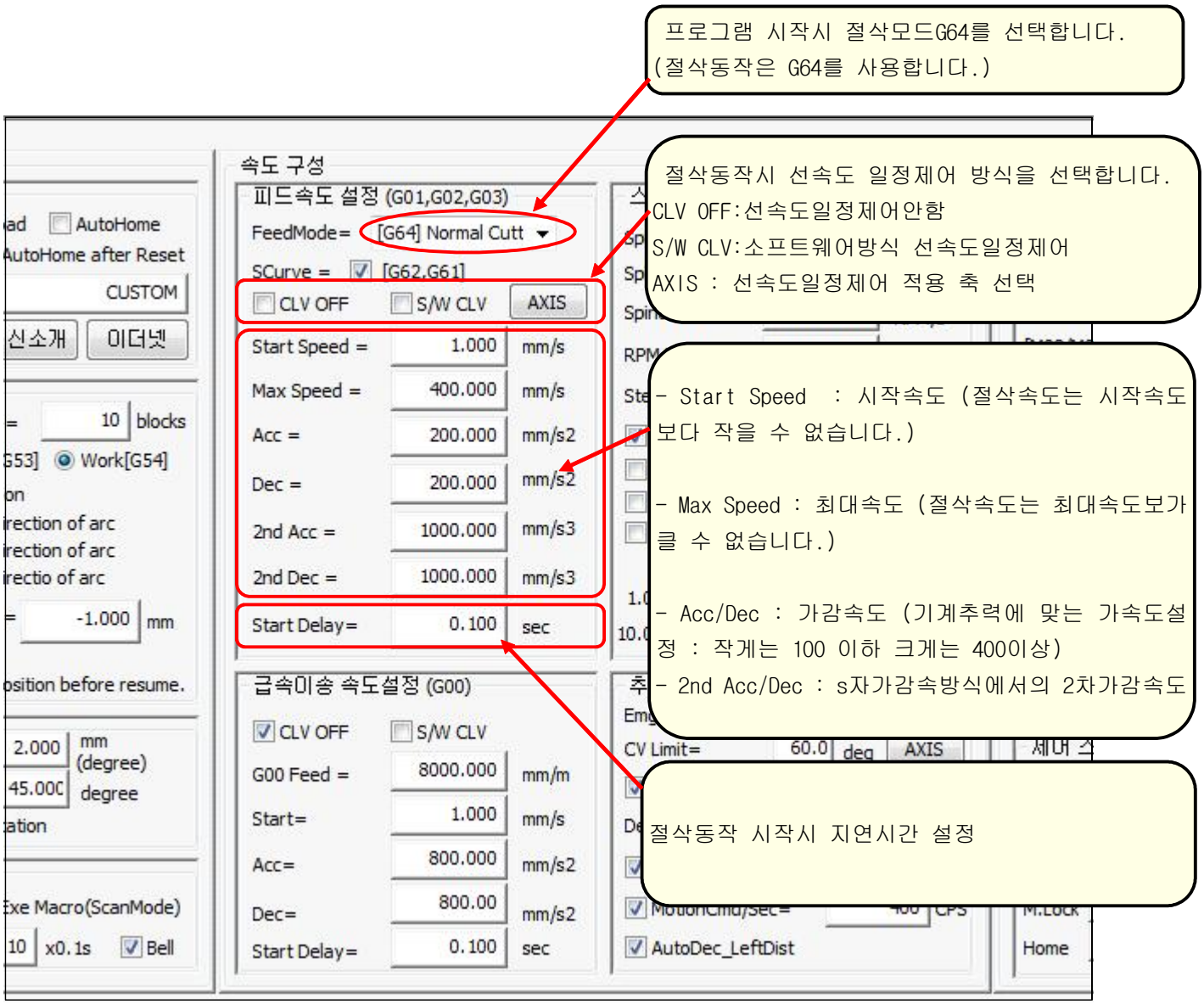
"CNC 구성" 화면의 "속도구성" 그룹에서는 이들의 속도특성에 대한 다양한 동작특성 파라메터들을 포함하고 있습니다. 이들 구성요소들 확인하기 전에 먼저 간단히 상기 동작속도의 종류를 요약해 봅니다.
(1) **절삭속도** (FEED 혹은 작업속도 : 절삭,커팅,용접등의 목적잡업속도) => G코드 블록내에서 F어드레스로 표현.
(2) **급속이송속도** (준비이동속도) => G코드내에서 속도조절기능없음. G00코드에 의해 동작자체만 표현.
(3) **주축속도** (작업장치속도) => G코드 블록내에서 S어드레스로써 표현.

여기서는 먼저 절삭속도(FEED) 와 급속이송속도의 특성에 대해서 간단하게 설정해 봅니다.

4.6.1. 절삭속도 특성파라메터 설정

G코드 상에서 대표적으로 G01(직선절삭), G02(CW방향 원호절삭) G03(CCW방향 원호절삭) 코드의 동작속도는 F 어드레스코드 (FEED코드)로써 규정합니다.

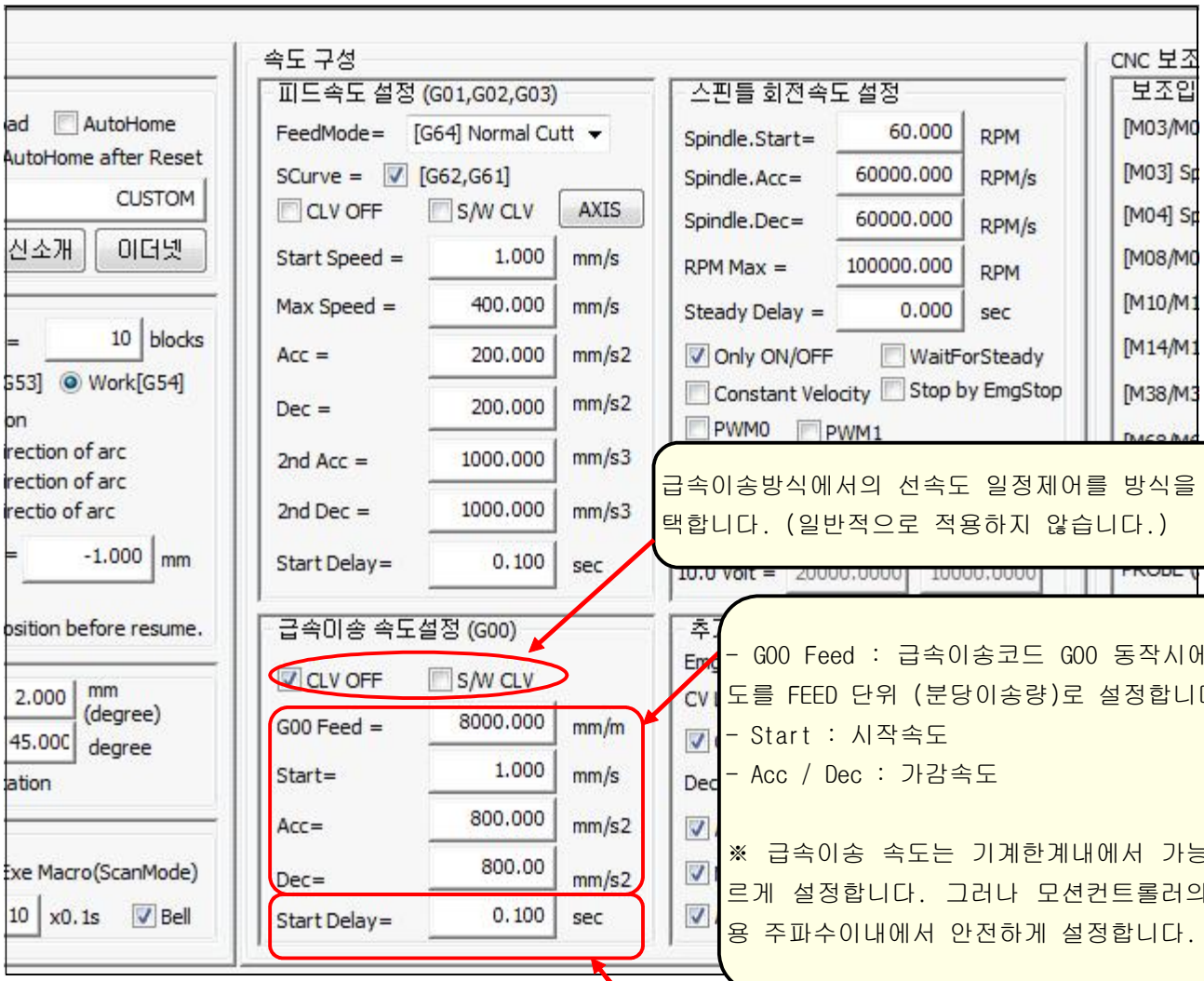
개별 축의 속도정보 설정 항목과 같이 시작속도, 최대제한속도, 가속도등의 정보를 포함해서 추가적으로 절삭모드, 속도프로파일, 선속도일정제어방식, 선속도일정제어축, 동작지연값 등이 제공됩니다.
(이들 설정항목들은 모션엔진의 버전과 프로그램버전에 따라 다소 차이가 있을 수 있습니다.)



4.6.2. 급속이송속도 특성파라메터 설정

G코드 상에서 대표적으로 G00 코드의 동작속도에 적용되는 속도제어 동작특성을 규정합니다.

개별 축의 속도정보 설정항목과 같이 시작속도, 최대제한속도, 가속도등의 정보를 포함해서 추가적으로 선속도일정 제어방식, 동작지연값 등이 제공됩니다.
(이들 설정항목들은 모션엔진의 버전과 프로그램버전에 따라 다소 차이가 있을 수 있습니다)



급속이송방식에서의 선속도 일정제어를 방식을 선택합니다. (일반적으로 적용하지 않습니다.)

- G00 Feed : 급속이송코드 G00 동작시에 이동속도를 FEED 단위 (분당이송량)로 설정합니다.
- Start : 시작속도
- Acc / Dec : 가감속도

※ 급속이송 속도는 기계한계내에서 가능하면 빠르게 설정합니다. 그러나 모션컨트롤러의 최대사용 주파수이내에서 안전하게 설정합니다.

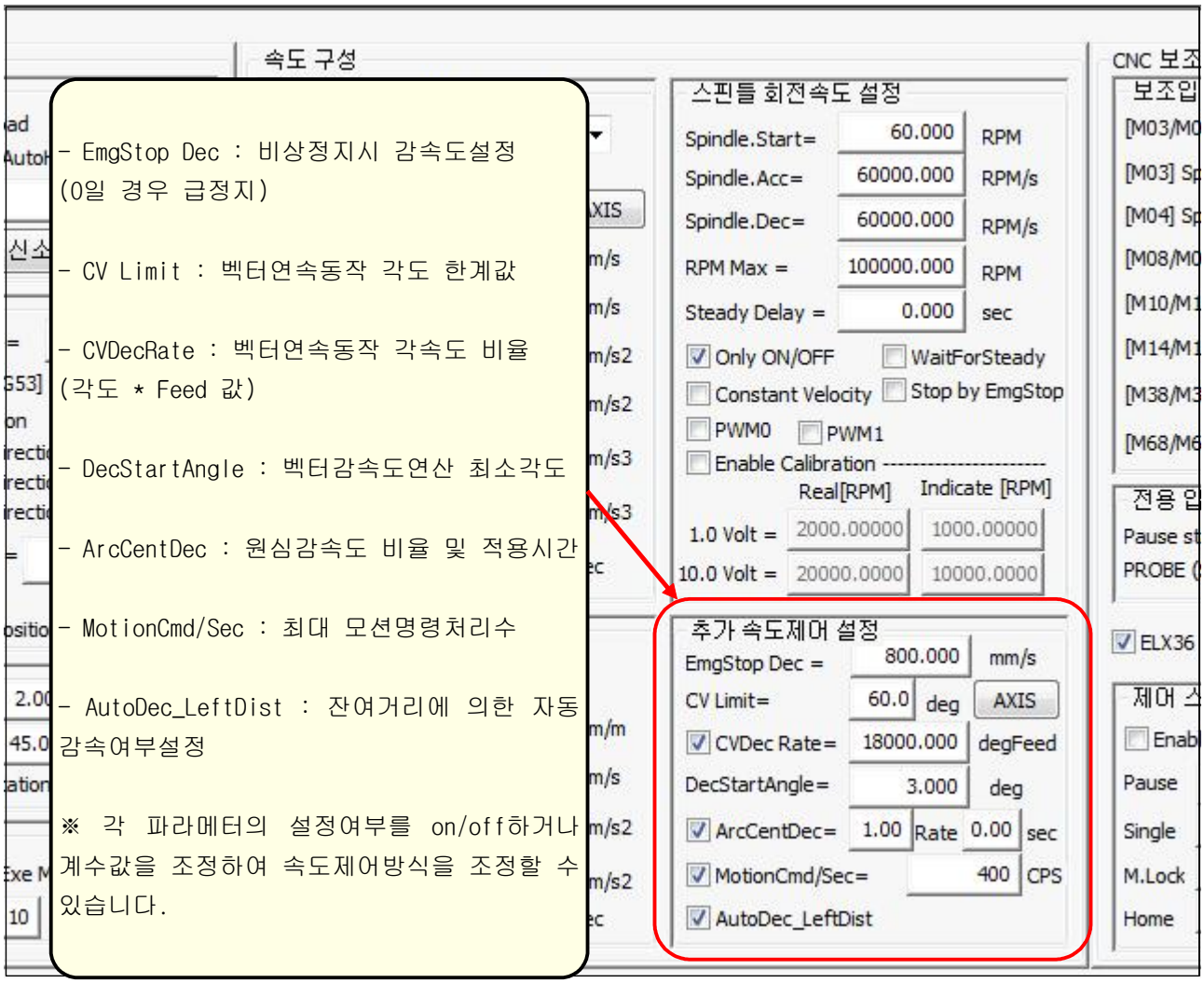
급속이송이 처음 시작될 때 지연시간(휴식시간)을 설정합니다.

4.6.3. 기계장치간의 절삭 속도 제어방식 선택

절삭동작 혹은 작업동작은 기계의 종류마다 각각 다른 작업이 될 수 있습니다.
일반적으로 스핀들을 이용한 밀링형태의 절삭장치와 레이저를 이용한 커팅용 CNC장치는 작업 동작시에 속도제어 혹은 가감속의 방식이 달라집니다.

예를 들면 직선이나 원호들이 상호 연속해서 경로를 형성할 경우 절삭가공기는 기계 및 공구의 부하(힘)를 고려하여 자동으로 감속하는 경향을 갖고 있지만, 레이저 절단의 경우 이러한 감속경향은 열의 집중화를 발생하여 작업품질을 저해할 수 있습니다. 그러므로 레이저 절단기의 경우 최종 움직이는 헤더장치의 관성부하가 크지 않을 경우 최대한 선속도를 일정하게 유지하거나 G코드상에서 열확산 등이 작게 분포되도록 절단 경로순서를 최적화 하게 됩니다.

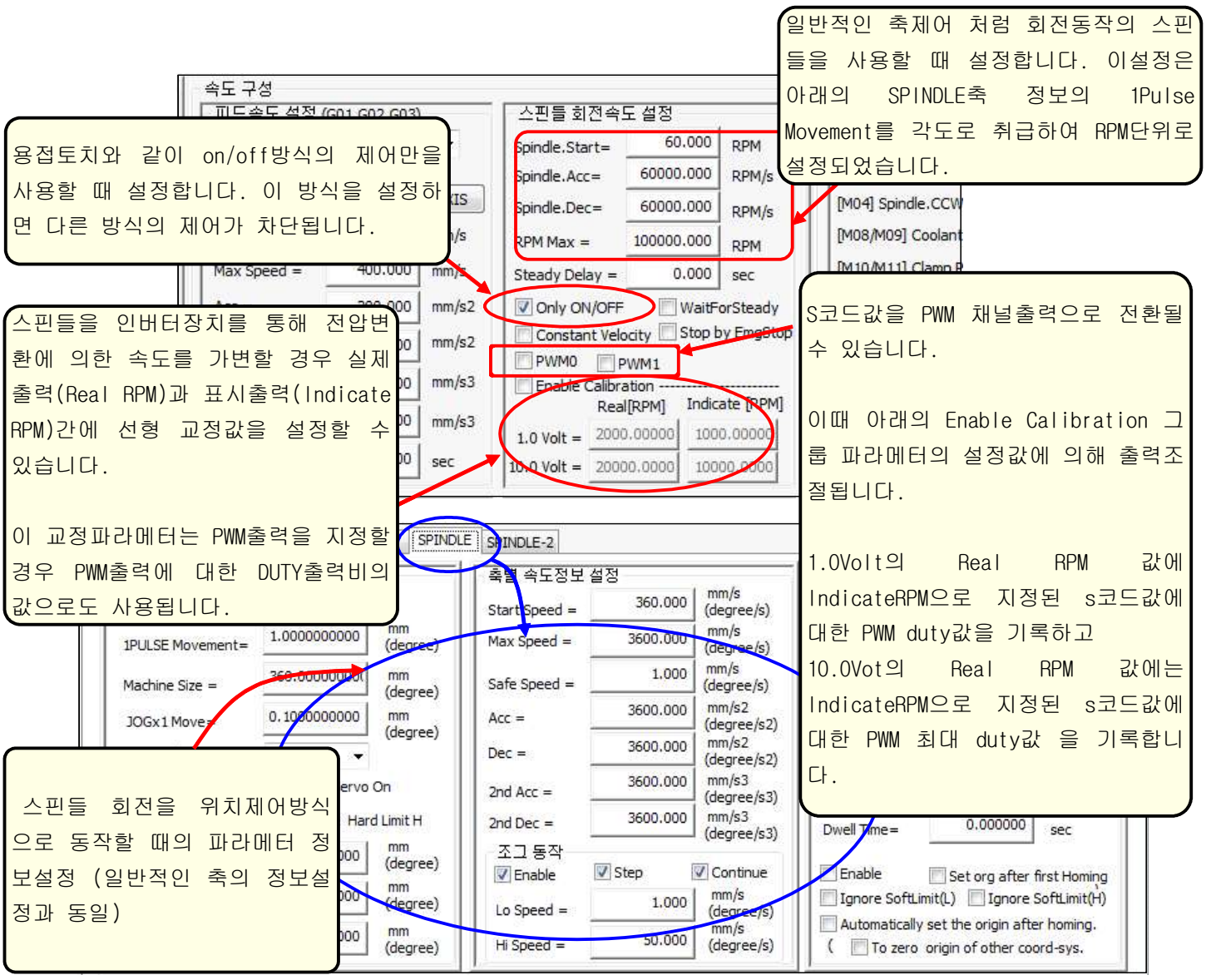
모션 마스터프로그램에서는 CNC장치의 특성마다 작업동작의 속도제어방식을 조절할 수 있는 아래의 몇가지 파라메터와 코드를 제공합니다. (보다 자세한 내용은 “CNC구성 화면” 편을 참조해 주세요)



4.7. 주축장치 설정

주축 장치는 흔히 모재가공에 직접적인 회전력이나 절삭력과 같은 최종 성형력이나 에너지를 공급하는 장치로써 밀링형태의 기계에서는 스핀들에 해당합니다. 이 주축장치는 CNC기계장치의 종류마다 다양해서 CNC레이저절단기의 경우 레이저빔의 발생장치가 될 수 있고, CNC 용접로봇의 경우 용접기나 용접토치가 될 수도 있습니다. G코드 체계에서는 일반적으로 S어드레스코드와 M3/M4/M5 M보조코드를 사용하여 주축동작을 지원하고 있으며 실제 모션마스터는 사용자가 임의적인 코드를 만들거나 스크립터코드를 통해서 주축장치와 같은 다양한 말단 작업장치들의 동작을 지원해 줄 수 있습니다. 여기서는 간단하게 스핀들과 같은 회전 작업장치에 대해서 동작특성을 조절할 수 있는 파라메터들을 살펴봅니다.

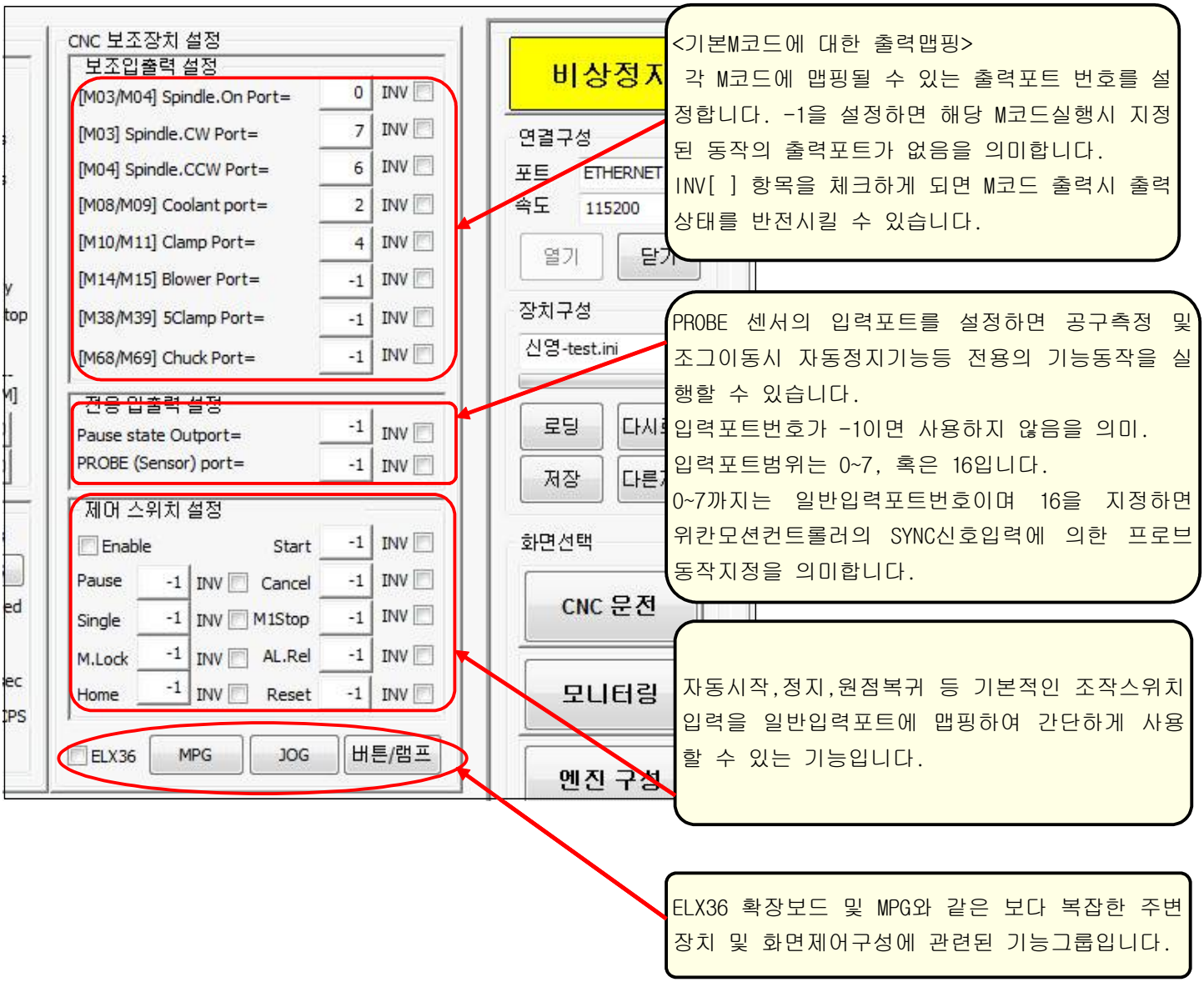
모션마스터는 아래의 제시된 동작방식들을 갖는 주축장치들에 대해 적절한 주축동작을 수행할 수 있는 파라메터를 제공합니다. 아래의 장치형태중 하나의 방식을 선택하여 사용합니다.
(자세한 설명은 “CNC 구성화면” 편을 참조해 주세요)
(1) 단순한 ON/OFF 형태의 주축장치
(2) 속도제어 동작을 포함하는 주축장치
(3) 위치 및 속도제어 동작을 포함하는 주축장치



4.8. 주변장치 및 기타 입출력장치 설정

CNC기계장치는 XYZ이동장치와 스핀들과 같은 주축장치 이외에도 도어, 램프, 절삭유와 블로워 등의 주변장치와 시작 스위치, 정지스위치, 비상정지, 조그스위치, MPG 혹은 ATC(자동공구교환장치) 등 다양한 입출력장치와 연결될 수 있습니다.
모션마스터프로그램은 다양한 장치의 동작을 자동프로그램에서 조절할 수 있도록 기본적인 M코드의 입출력 맵핑을 설정하거나 별도의 매크로확장에 의한 스크립트언어를 통해서 지원하고 있습니다.

먼저 가장 기본적인 ON/OFF스위치형 입출력장치들은 아래와 같이 몇가지 M코드 및 스위치연결지정으로 간단하게 맵핑하여 사용될 수 있습니다. 또한 MPG와 같이 보다 복잡한 장치연결을 위해 별도의 설정도 포함하고 있습니다.



<기본M코드에 대한 출력맵핑>
각 M코드에 맵핑될 수 있는 출력포트 번호를 설정합니다. -1을 설정하면 해당 M코드실행시 지정된 동작의 출력포트가 없음을 의미합니다.
INV[ ] 항목을 체크하게 되면 M코드 출력시 출력상태를 반전시킬 수 있습니다.

PROBE 센서의 입력포트를 설정하면 공구측정 및 조그이동시 자동정지기능등 전용의 기능동작을 실행할 수 있습니다.
입력포트번호가 -1이면 사용하지 않음을 의미.
입력포트범위는 0~7, 혹은 16입니다.
0~7까지는 일반입력포트번호이며 16을 지정하면 위칸모션컨트롤러의 SYNC신호입력에 의한 프로브 동작지정을 의미합니다.

자동시작,정지,원점복귀 등 기본적인 조작스위치 입력을 일반입력포트에 맵핑하여 간단하게 사용할 수 있는 기능입니다.

ELX36 확장보드 및 MPG와 같은 보다 복잡한 주변장치 및 화면제어구성에 관련된 기능그룹입니다.

5. CNC 자동 운전

이 장은 CNC운전화면을 통해서 기초적인 자동모드의 동작을 실행해 봅니다.

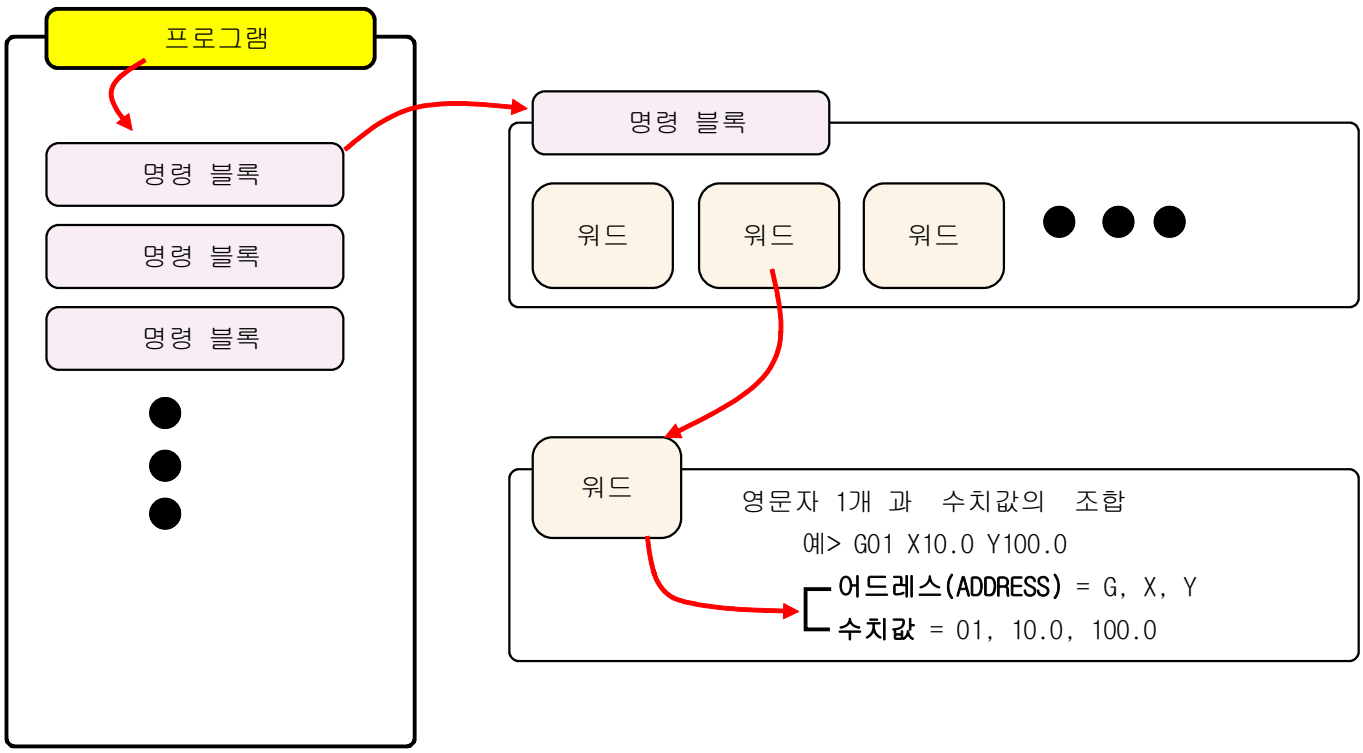
CNC 자동운전은 G코드 체계를 기반으로 만들어진 파일을 읽어들어 XYZ 직교좌표계 머신을 목적에 맞게 자동으로 움직여 주는 동작입니다.
CNC기계장치는 밀링형태의 절삭장치나, 레이저 및 플라즈마장치 등을 이용한 절단 혹은 재단장치, 용접이나 솔더링과 같은 융접장치, 팔렛타이징과 같은 적재 및 이송장치 등 다양한 목적동작을 갖는 장치들이 컴퓨터의 수치제어능력과 조합되어 보다 정확한 반복 작업을 수행할 수 있도록 만들진 기계장치입니다.

특히 모션마스터프로그램은 G코드 체계뿐만 아니라 G코드의 확장된 매크로언어방식의 제어문구조과 스크립터언어방식으로 변수를 선언하고 수식을 실시간으로 해석하며 다양한 메뉴들을 생성하여 선택적으로 움직여 줄 수 있는 기능들을 포함하고 있어, 여러 자동화분야에 다양하게 응용될 수 있습니다.

여기서는 먼저 가장 기본적인 G코드기반에서만 먼저 운영해 봅니다.

5.1. NC 코드 프로그램 (G코드 프로그램)

CNC운영의 기본이 되는 NC코드 프로그램은 하나 이상의 워드(WORD)로 구성된 명령 블록(BLOCK)의 나열입니다.
워드는 하나의 영문코드와 수치값으로 조합된 구성을 의미하며 하나의 영문코드를 어드레스(ADDRESS)라고 합니다.
아래의 그림으로 간단히 요약해 봅니다.



워드에 사용되는 영문자 어드레스(ADDRESS)는 각 문자마다 특별한 의미를 지닙니다. 아래에 몇 가지 요약해 봅니

다.
N : 명령번호(시퀀스번호) Sequence Number
G : 동작준비기능(동작 혹은 설정기능)
X,Y,Z,U,V,W : 좌표값
F : 이송(속도)기능
S : 주축기능
T : 공구기능
M : 보조기능

이러한 명령블록의 형성 규칙을 예를 통해서 보다 구체적으로 살펴봅니다.
N0001 G90 G21 G01 X100.0 Y100.0 F1000 S2000 M3;
- 시퀀스(명령)번호 N0001 은 항상 맨 처음에 등장합니다.
- G90 코드는 G90 이후의 좌표값 워드들을 절대좌표로 취급하라느 설정(준비)입니다. (G91 : 상대좌표)
- G21 코드는 G21 이후 좌표값 워드들의 단위를 mm단위로 취급하라는 설정(준비)입니다. (G21 : 인치단위)
- G01 코드는 이후의 좌표값 워드를 참조하여 직선이동을 준비하라 명령입니다.
- X100.0 Y100.0 은 각 좌표값입니다.
- F1000 은 현재부터 급속이송을 제외한 절삭이송명령의 동작속도를 FEED 1000으로 설정하라는 명령입니다.
(밀링형태의 NC코드에서 F는 일반적으로 분당 이송량 즉 1분에 몇 mm를 이송할 것인가에 대한 단위입니다.)
- S2000 은 스핀들 회전속도를 2000RPM으로 설정하라는 명령입니다.
- M3은 스핀들을 CW방향으로 회전시키는 보조명령입니다.
마지막으로 세미콜론은 현재 명령블록의 종료를 의미하며 이후로 나오는 문자들은 해석되지 않습니다.
하나의 명령블록은 한 줄에서 종료되며 다음 명령블록은 다음 줄에서 부터 시작됩니다.

<어드레스 기능요약표>

주요기능	어드레스	설명
프로그램 번호	O	현재 프로그램 번호로써 서브루틴호출의 기준번호로 참조됨
시퀀스 번호	N	해당 명령블록의 명령번호 (GOTO문에서 참조됨)
동작 준비기능	G	각 모션동작 및 동작옵션 설정 기능
좌표	X,Y,Z,A,B,C	동작 종점 좌표 지정
	U,V,W	각 X,Y,Z축 동작 종점 증분 좌표 지정
	R,I,J,K	원호보간 반경 및 중심점 좌표지정
이송기능	F	분당이송 혹은 회전당이송 등의 속도지령
주축기능	S	주축의 RPM회전수 지정
공구기능	T	공구번호 지정(혹은 공구교환번호 지정)
보조기능	M	기계 주변장치 조작 IO 제어 및 보조동작 제어용
보정기능	H, D	공구길이, 공구경 보정번호
파라메터	P, K, L, Q	사이클동작 및 보조프로그램(서브루틴)호출 등의 다양한 파라메터 전달에 사용

5.2. G 코드의 생성 방식

<직접 코딩>

NC코드프로그램 혹은 간단히 G코드라고 표기하도록 하겠습니다. G코드는 문자코드로 구성되어 있으므로 복잡하지 않은 움직임은 사용자가 직접 여러 명령어드레스와 좌표값들을 입력하여 구성할 수 있습니다.

모션마스터 프로그램의 설치폴더의 ExampleW 폴더에 기본 예제들 중 일부는 직접 작성되어 코딩되었습니다.

<CAM 프로그램 활용한 G코드 생성>

그러나 CAD와 같은 도면이나 보다 복잡하고 반복된 다량의 모션동작(혹은 가공동작)을 직접 생성하기는 쉽지 않습니다. 더구다나 한번 동작 후 프로그램을 조금씩 반복 수정하여 교정시켜 최종 목적동작을 만들 경우에도 결코 만만하지 않은 작업이 됩니다. XYZ밀링형태의 절삭가공이나 레이저절단과 같은 가공을 도면정보(AUTOCAD의 DXF파일 등)를 통해서 자동으로 G코드를 생성하는 프로그램을 **CAM프로그램**이라고 합니다. CAM프로그램은 다양한 종류들이 나와 있지만 기본적으로 조금은 고가의 유료제품군들이 많이 있습니다. CAM프로그램을 이용하면 보다 간편하게 작업코드를 생성할 수도 있습니다.

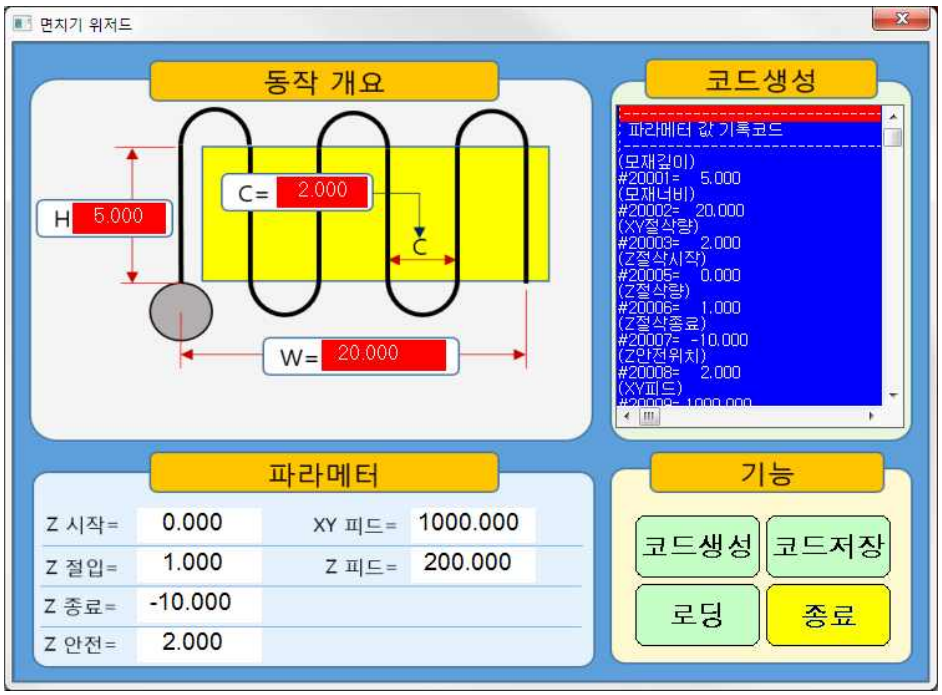
현재 모션마스터에는 기본적으로 CAM기능을 갖고 있지 않습니다. (추후 모션마스터 업그레이드 버전에서 레이저 및 라우터와 같은 기본 절단용 CAM 기능을 준비될 수 있습니다.)

<티칭방식에 의한 G코드 생성>

CAM프로그램은 주로 가공목적의 G코드를 생성하는데 많이 사용됩니다. 그러나 모션동작은 가공분야 뿐만 아니라 솔더링, 조립, 운반, 적재등 다양한 분야에서 사용될 수 있으며 많은 응용에서 티칭방식(교사방식)에 의한 자동모션코드생성 방식도 있습니다. 현재 모션마스터는 티칭용 자동코드생성화면을 갖추고 있지 않지만 스크립터언어를 통해서 이러한 기능을 구현할 수 있도록 하고 있습니다.

<간단한 사용자 위저드 기능에 의한 G코드 생성>

단순한 동작의 반복작업과 같은 G코드들은 스크립터언어를 통해 사용자가 직접 자동코드를 생성할 수 있는 방법이 준비되어 있습니다. 아래와 같은 면치기 위저드화면이 모션마스터의 스크립터언어를 이용하여 각재의 면치기를 실행하는 G코드를 자동으로 생성하는 프로그램을 구현해 본 것입니다. (현재 F8 버튼과 연결되어 있습니다.)



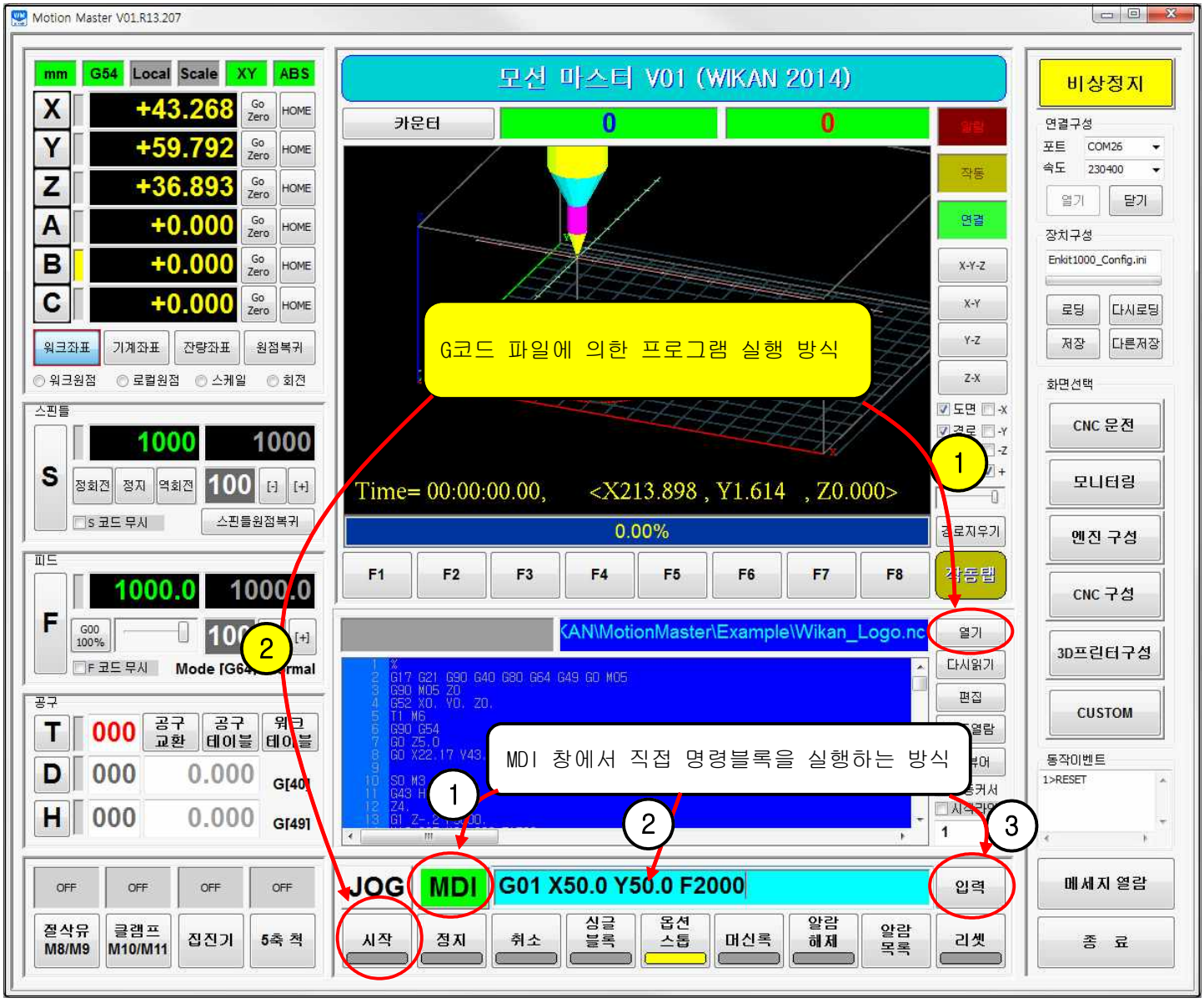
5.3. CNC 운전화면에서의 G코드 실행방식

동작탭화면의 [CNC운전]버튼을 선택하면 CNC운전화면이 나타납니다. (자세한 내용은 “CNC 운전화면” 편을 참조하세요)

CNC 운전화면에서 G코드를 실행하는 방식은 크게 두가지 방식이 있습니다.

하나는 MDI(Manual Direct Input:수동 명령직접입력)창에서 하나의 명령블록을 입력하여 실행하는 방법이 있으며 다른 하나는 G코드로 작성된 파일을 읽어들여 [시작]버튼을 통해서 실행하는 방법입니다.

각각의 방법은 아래의 CNC운전화면상에 간단히 설명합니다.



MDI창에 의한 입력방식은 조그동작에서 처럼 수동으로 목적 동작을 하나씩 움직이기 위해 사용됩니다.

G코드 파일에 의한 프로그램을 실행할 때에는 먼저 G코드파일 읽은 후 읽혀진 G코드파일을 3D그래프상에서 주요 궤적을 확인하고 시작버튼을 눌러 실행하게 됩니다. G코드 파일일 읽어들여 실행하는 부분을 간단하게 요약하였지만, 실제 작업 대상물의 위치가 임의적인 위치에 고정되었다면 조그동작을 통해서 작업대상물의 초기가공위치까지 이동한 후 워크좌표를 초기화 하는 등의 작업을 추가적으로 필요 할 수 있습니다.

5.4. XYZ 좌표계 이해하기

이번 절에서는 CNC 기계시스템과 G코드 체계의 좌표계 시스템에 대해서 설명합니다.
XYZ기계시스템에서 어떤 지점의 좌표값을 규정하기 위해서는 기계시스템의 최종 움직임을 발생하는 부분과 구속된 하나의 위치점이 필요하게 됩니다. 그 위치점의 좌표값을 기계좌표값으로 정의할 수 있습니다.

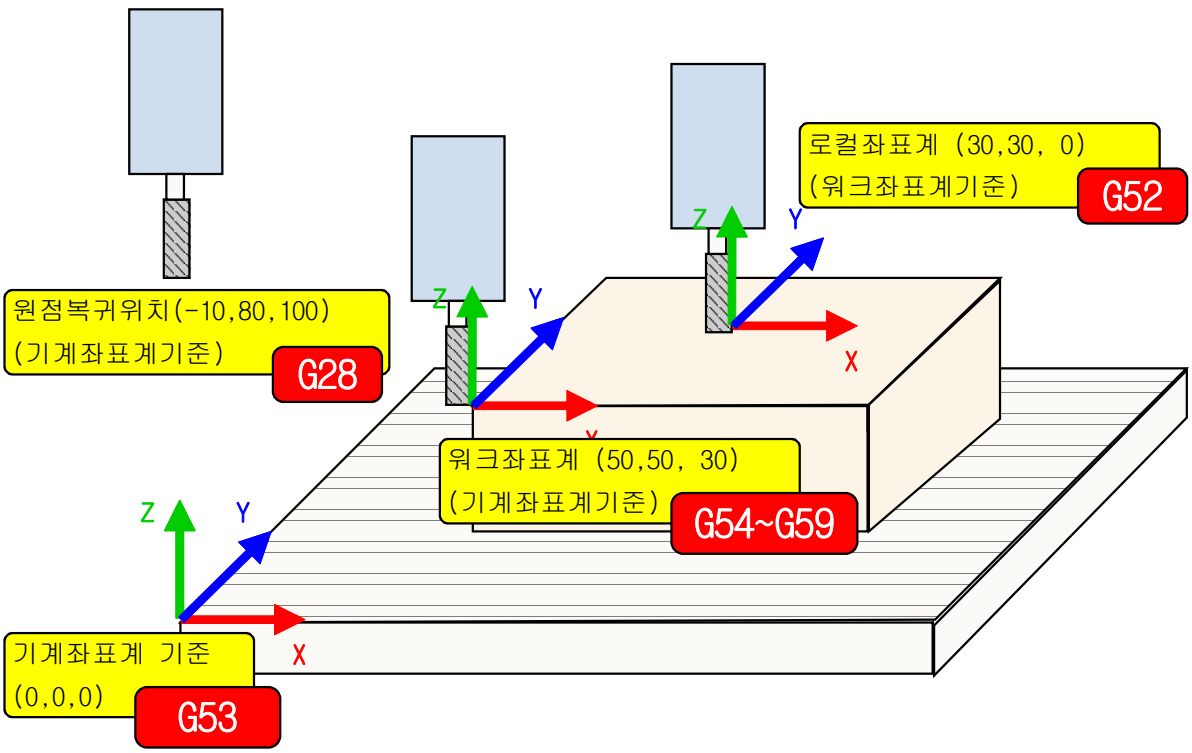
밀링형태의 CNC기계를 가정한다면 대체로 스핀들에 장착된 공구의 끝단부를 측정 기준위치로 삼을 수 있습니다. 이러한 기준 위치점을 만들기 위해 기준공구의 개념을 두어 이 스핀들에 장착한 후 공구의 끝단위치를 기계 위치로 규정하게 됩니다. (이러한 기준공구 개념은 공구교환시 다른 공구의 길이교정 등의 보정값에서도 활용됩니다.)

이렇게 순수 기계 절대 위치에 의존하는 좌표계를 기계좌표계라고 합니다. 기계좌표계는 원점복귀동작 완료를 통해서 초기화 되어 CNC프로그램과 상호연결 됩니다.

특정 CNC기계들은 이러한 구속된 기계좌표값에 의존하여 매번 동일한 자동동작을 수행하는 경우도 있지만, 대부분 범용 CNC기계들은 기계좌표계 상에서 임의위치를 기준위치(0 영위치)로 하는 새로운 좌표계를 사용하는 경우가 많습니다. 특히 범용의 가공기계들은 가공대상모재를 임의의 위치에 클램핑 한 후 스핀들에 장착된 공구를 가공시작점으로 이동시킨 후 이 지점을 기준으로 G코드를 실행하는 경우가 많습니다.
이 처럼 기계좌표계를 기준으로 가공모재의 임의지점을 새로운 원점으로 만든 좌표계를 워크좌표계 혹은 공작물좌표계(작업좌표계)라고 부릅니다.

가공모재 위에 같은 형상의 가공이 배열형태로 여러 개가 존재한다면 워크좌표계 위에 또다시 새로운 변위지점을 원점으로 하는 로컬좌표계를 만들 수도 있습니다. 로컬좌표계의 기준변위를 변화시켜가면서 G코드를 실행하면 하나의 가공코드로 동일한 형태의 여러 개의 가공동작을 실행할 수 있게 됩니다.

이들 좌표계개념을 그림으로 요약하면 아래와 같습니다.



프로그램할 때 각각 기준좌표계를 어떤 좌표계로 사용해서 위치를 표현하는가를 구분해 주는 G코드는 아래와 같습니다.

G53 : 이하 좌표값 워드들은 기계좌표계를 기준으로 표현된 위치임을 알림.
G54 : 이하 좌표값 워드들은 워크좌표계 G54 기준점으로 표현된 위치임을 알림
G55 : 이하 좌표값 워드들은 워크좌표계 G55 기준점으로 표현된 위치임을 알림
G56 : 이하 좌표값 워드들은 워크좌표계 G56 기준점으로 표현된 위치임을 알림
G57 : 이하 좌표값 워드들은 워크좌표계 G57 기준점으로 표현된 위치임을 알림
G58 : 이하 좌표값 워드들은 워크좌표계 G58 기준점으로 표현된 위치임을 알림
G59 : 이하 좌표값 워드들은 워크좌표계 G59 기준점으로 표현된 위치임을 알림
G52 : 이하 좌표값 워드들은 로컬좌표계 G52 기준점으로 표현된 위치임을 알림

모션마스터의 로컬좌표계 G52는 현재 좌표계설정이 기계좌표계일 경우 기계좌표계를 기준으로 변위된 로컬 좌표계를 형성하고, 워크좌표계일 경우 워크좌표계를 기준으로 변위된 로컬 좌표계를 형성하게 됩니다. 로컬좌표계설정은 아래와 같이 간단합니다.

G54
G52 X30.0Y30.0Z0 ; G54 좌표계를 기준으로 X방향으로 30.0, Y방향으로 30.0 변위된 지점을 0,0,0 으로 취하는 로컬좌표계 설정
G52 X0Y0Z0 ; 로컬좌표계 취소

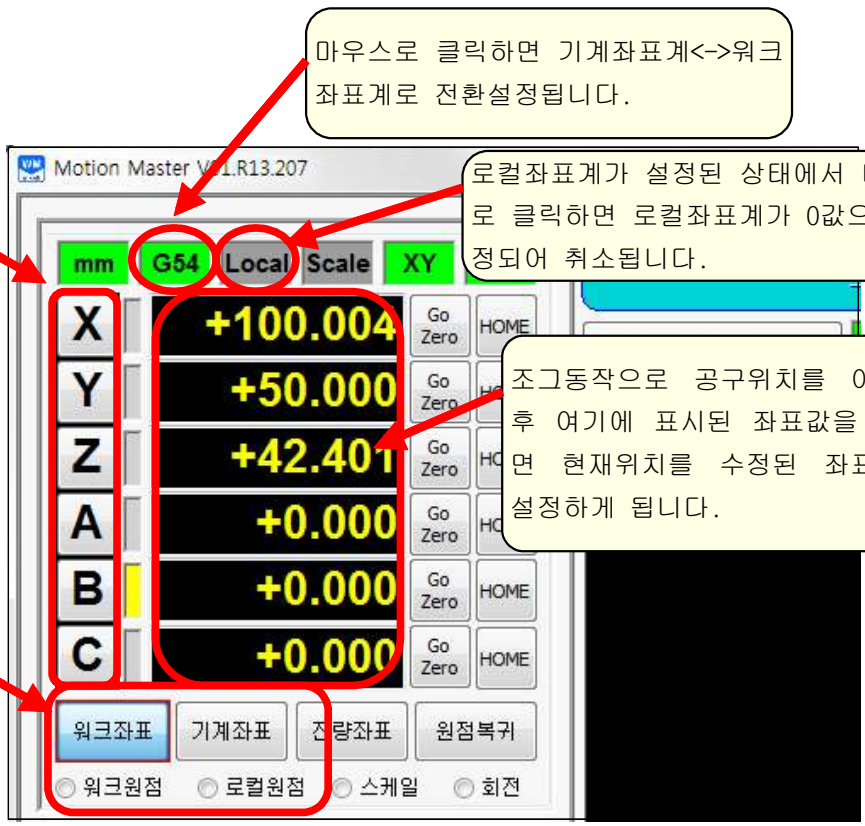
모션마스터 프로그램에서는 이러한 좌표계변환을 보다 쉽게 전환하기 위해서 아래와 같은 기능들이 준비되어 있습니다.

각 축버튼을 눌러주면 현재 위치를 현재좌표계에 대한 기준위치값으로 강제설정합니다.

현재좌표계가 기계좌표계이면 원점복귀위치값으로 설정되고
워크좌표계이면 0.000으로 설정됩니다.

현재 좌표계 종류에 관계없이 기계좌표계상의 위치와 워크좌표계상의 위치를 전환시켜 표시할 수 있으며

현재 워크좌표계의 기준변위나 로컬좌표계의 기준변위값들을 확인할 수 있습니다.



5.5. CNC운전 자동실행

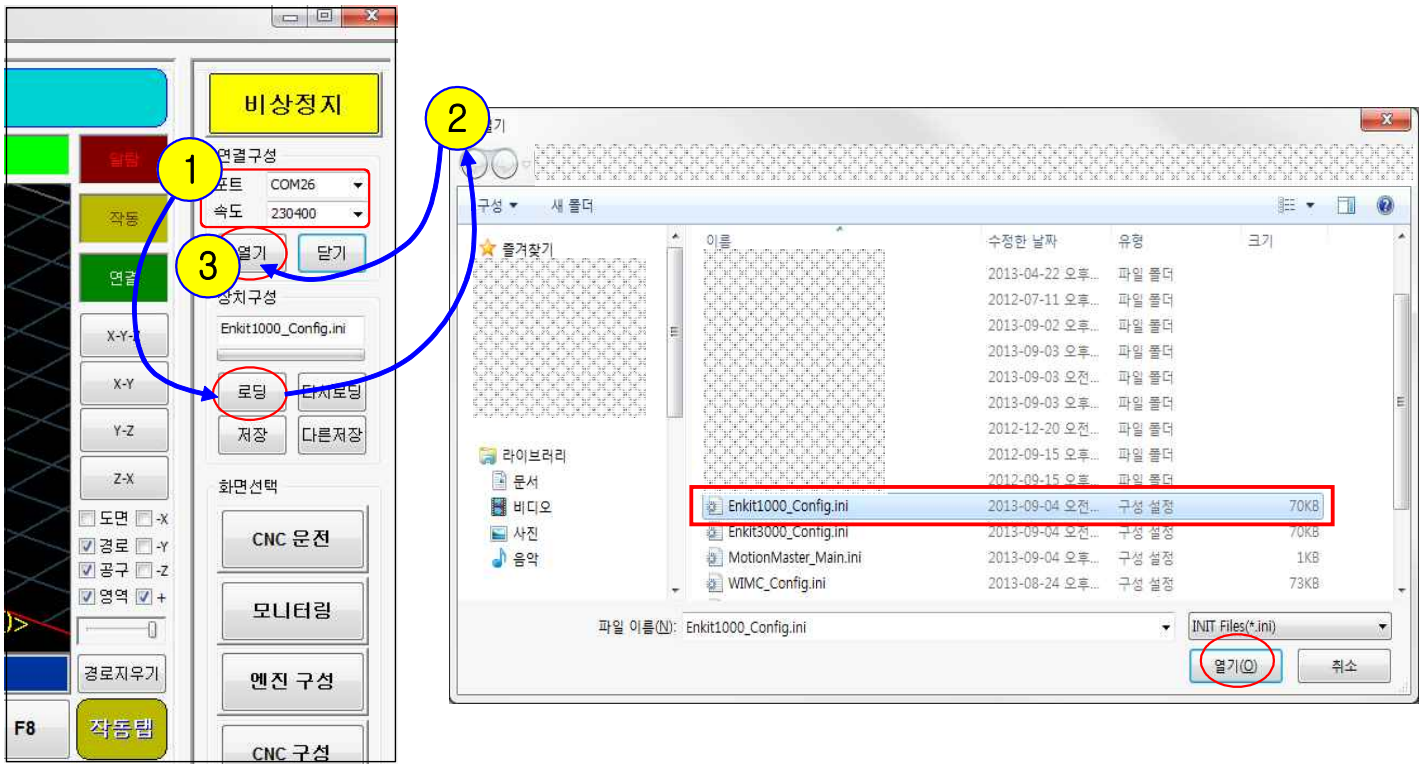
이번 절에서는 모션마스터프로그램의 실행 초기단계에서부터 G코드파일을 읽어들여 자동으로 실행하는 과정 전체 흐름을 설명합니다. 전체 과정의 흐름순서를 요약하면 아래와 같습니다.

(1) 프로그램 실행 및 초기화
(2) 원점복귀 실행
(3) G코드파일 로딩
(4) 가공 원점(워크좌표) 설정 및 G코드 리로딩
(5) 실행 및 완료

5.5.1. 모션마스터 프로그램 실행 및 초기화

모션마스터 프로그램 실행아이콘

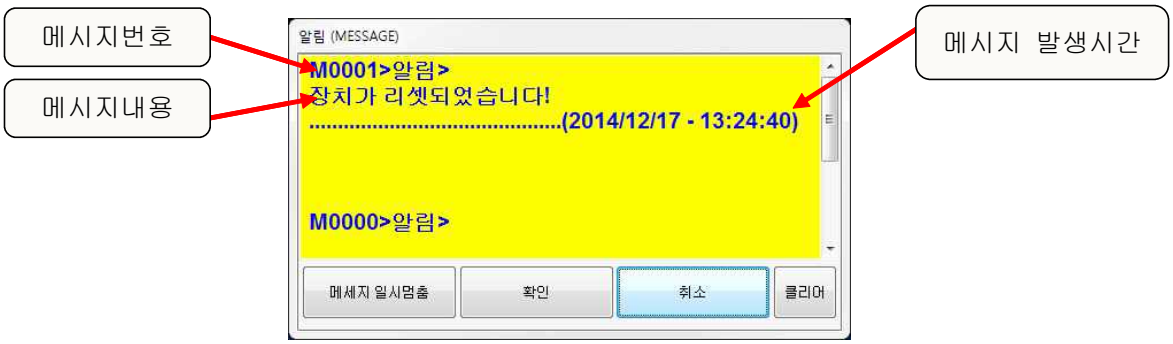
을 클릭하여 프로그램을 실행합니다.
먼저 모션컨트롤러와 통신연결을 위해 동작탭화면의 연결구성을 설정합니다.
연결구성은 기본적으로 연결포트와 연결속도를 선택하는 것으로 완료됩니다. [열기]버튼을 누르기 전에 먼저 장치구성항목그룹에서 [로딩]버튼을 눌러 기계장치 혹은 모션컨트롤러장치에 적절한 구성초기화 파일을 선택합니다.
기본적인 구성초기화파일은 프로그램설치폴더의 Config하위 폴더에 포함되어 있습니다.



정상적으로 통신이 이루어졌다면

램프가 점등되고 리셋초기화 과정이 진행됩니다.

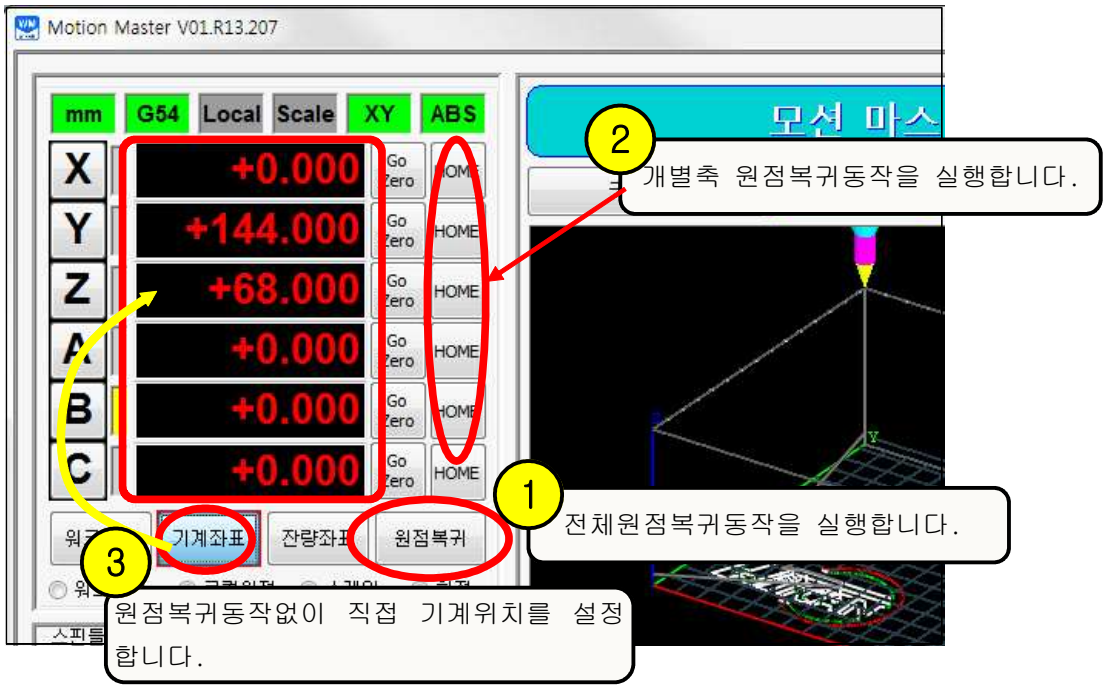
리셋초기화 과정이 완료되면 아래의 메시지창(알림창)이 표시된 후 몇 초후에 자동으로 사라집니다.



5.5.2. 원점복귀 실행

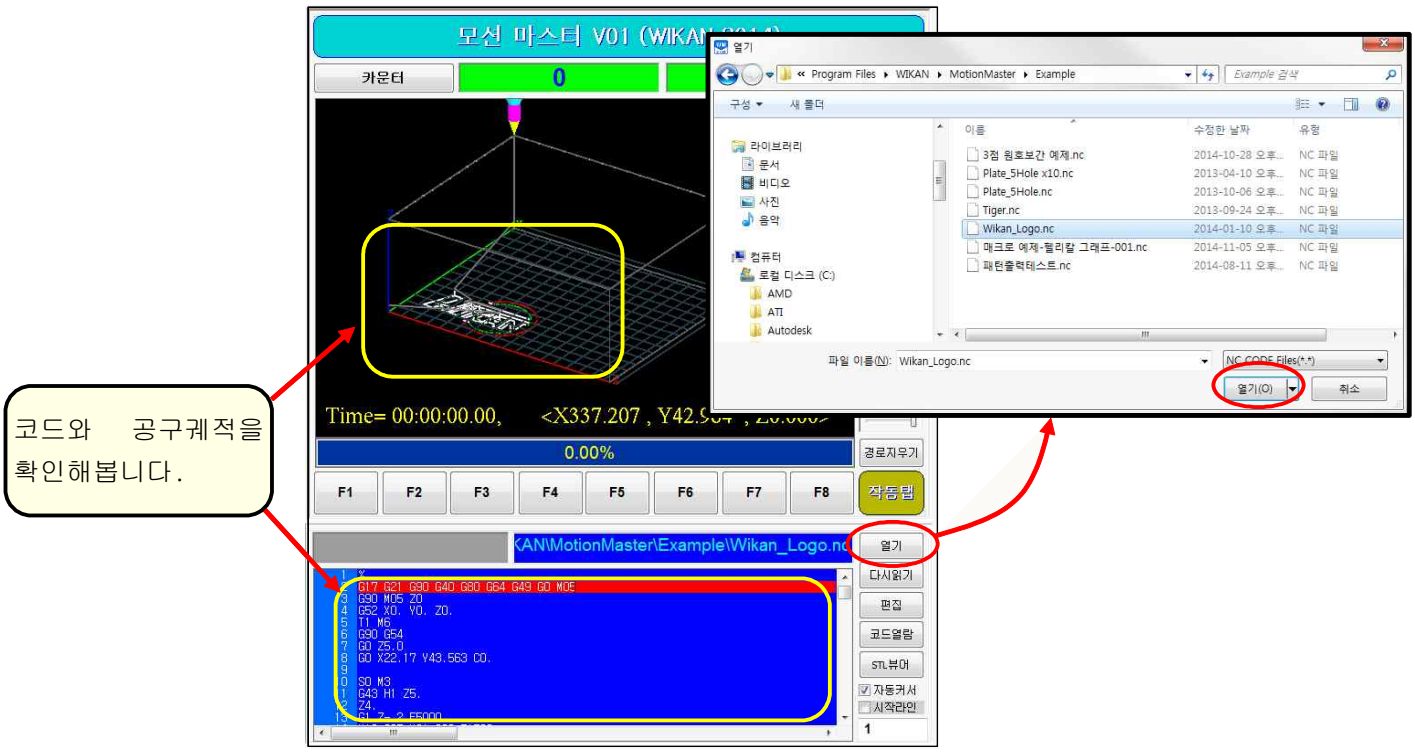
초기화과정이 완료되면 가장 먼저 원점복귀 동작을 실행합니다. 원점복귀 동작은 실제 기계의 위치와 모션마스터프로그램상의 좌표를 연결하는 기능을 합니다. 이러한 기계좌표의 연결은 CNC 동작의 기준이 됩니다.
특히 수동조그 혹은 MPG동작을 제한하는 소프트리미트 상/하한 위치가 바르게 동작하기 위해서는 반드시 현재 기계위치를 인식할 필요가 있습니다.

정확한 원점복귀동작이 필요로 하지 않는 CNC장치라면 원점복귀명령 대신 아래와 같이 기계좌표값을 직접 입력하여 대강의 위치를 설정하여 사용할 수 도 있습니다.어떠한 방식이로든 CNC동작을 위해서는 기계위치와 프로그램좌표간의 맵핑이 필요하게 됩니다. 아래에 기계좌표와 프로그램을 연결하는 3가지 방식을 보여줍니다.



5.5.3. G코드파일 로딩

프로그램 설치폴더의 하위폴더 ExampleW에 몇가지 샘플 G코드 예제가 포함되어 있습니다. 이들 중에서 Wikan_Logo.nc 파일을 열어서 코드와 공구궤적을 확인해 봅니다.



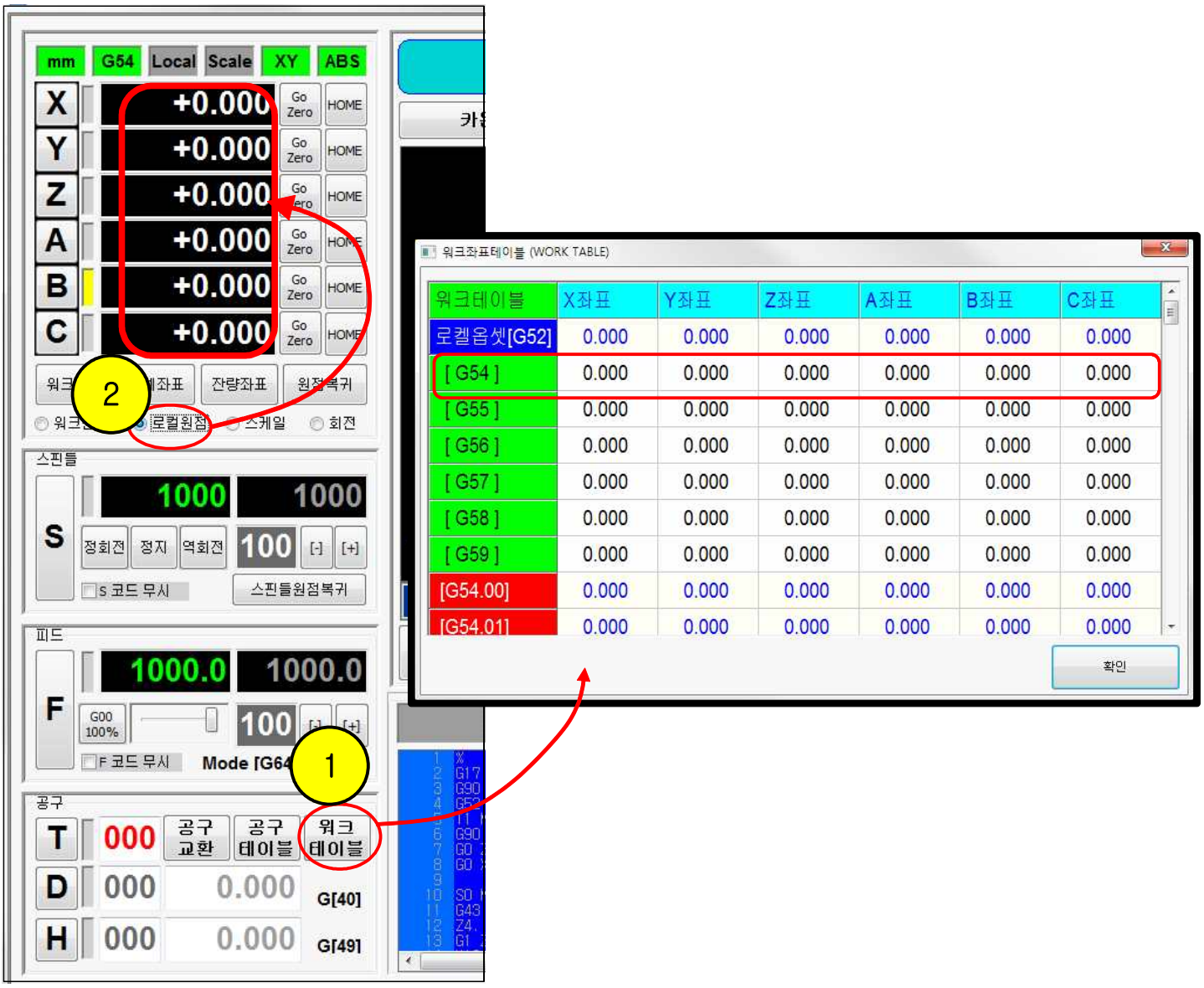
<3D 화면 마우스 사용요약>
- Zoom In/Out : Mouse Scroll
- Rotate : Mouse Middle(Wheel)Button + MouseMove
- Move : Mouse Left Button + MouseMove

<3D 공구궤적 색깔별 동작코드 구분>
- [■]회색은 G00 코드인 급속이송을 표현합니다.
- [ ]흰색은 G01 코드인 직선보간을 표현합니다.
- [■]녹색은 G02 코드인 CW원호보간을 표현합니다.
- [■]적색은 G03 코드인 CCW원호보간을 표현합니다.
- [■]황색은 Tool의 이동경로를 표현합니다.

5.5.4. 가공원점(워크좌표) 설정

프로그램을 처음 실행하면 일반적으로 G54좌표계로 설정되어 있을 것입니다. 현재 기준좌표계를 바꾸려면 아래의 버튼을 클릭하거나 MDI 창에서 원하는 좌표계설정용 G코드를 입력한 후 실행하면 됩니다.

워크좌표는 G54~G59까지 총6개의 기준좌표계 중 하나를 선택할 수 있습니다. 모션마스터는 G54.00 ~ G54.99까지 총 100개의 워크좌표를 추가로 사용할 수 있습니다. 이들 워크좌표계의 기준위치들 (기계좌표에 대한 상대위치)을 확인하려면 다음 그림과 같이 (1) [워크테이블] 버튼을 눌러서 전체 워크좌표계속에서 확인하거나 (2)좌표 표시창을 통해서 현재워크좌표계에 대한 기준위치를 확인 할 수 있습니다.



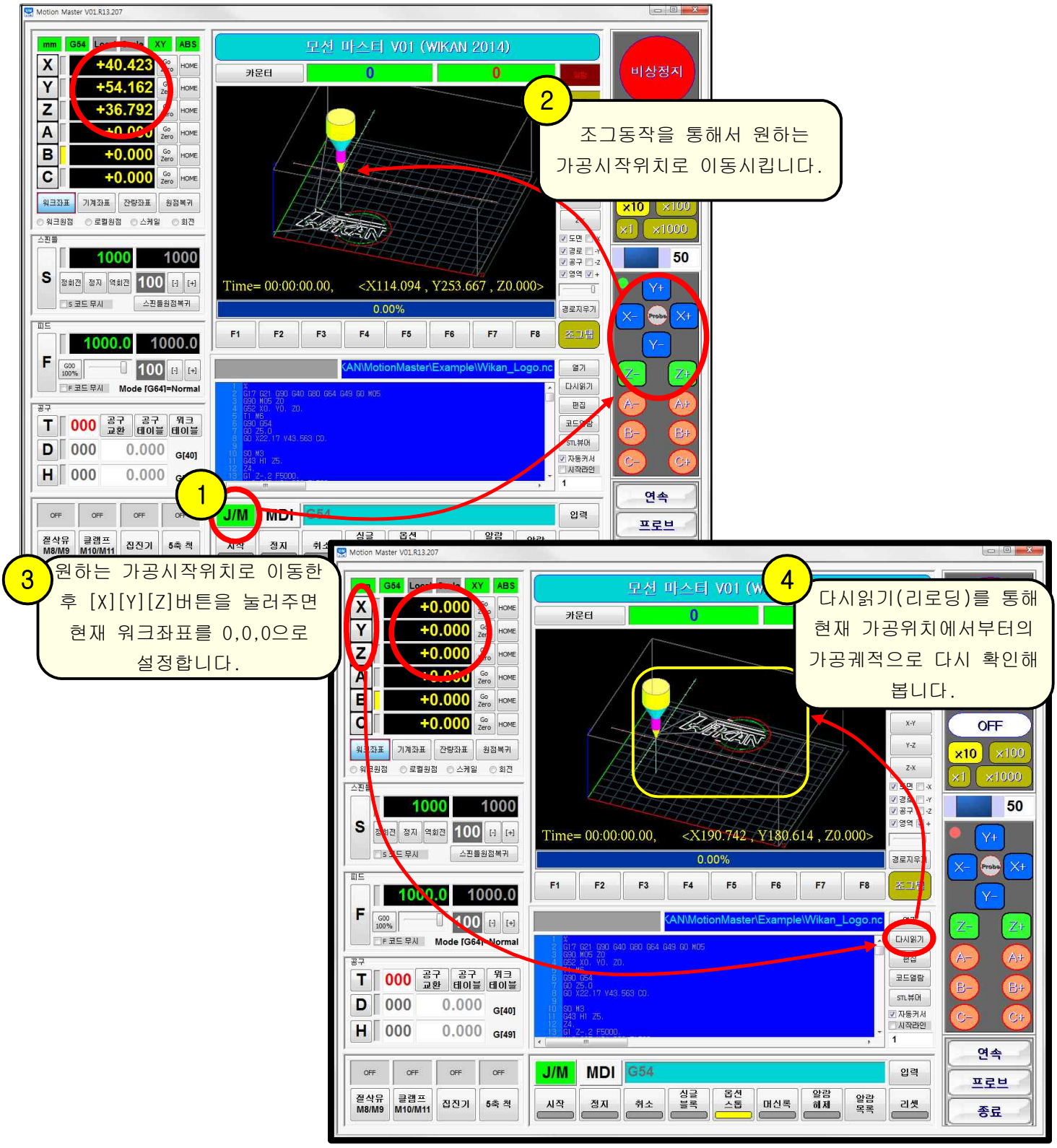
이러한 워크좌표계의 기준위치는 아래의 절차대로 실행하면 조그 혹은 MPG동작을 통해서 사용자가 임의적으로 지정될 수 있습니다.

(1) [JOG]버튼을 눌러 수동 이동이 가능하도록 합니다.

(2) MPG/JOG 화면을 이용하여 원하는 위치로 공구를 이동시킵니다.

(3) 현재 위치를 워크좌표계(G54) 기준원점(0,0,0)으로 설정합니다.

(4) [다시읽기]버튼을 눌러 새로운위치 에서의 가공궤적을 확인해봅니다.



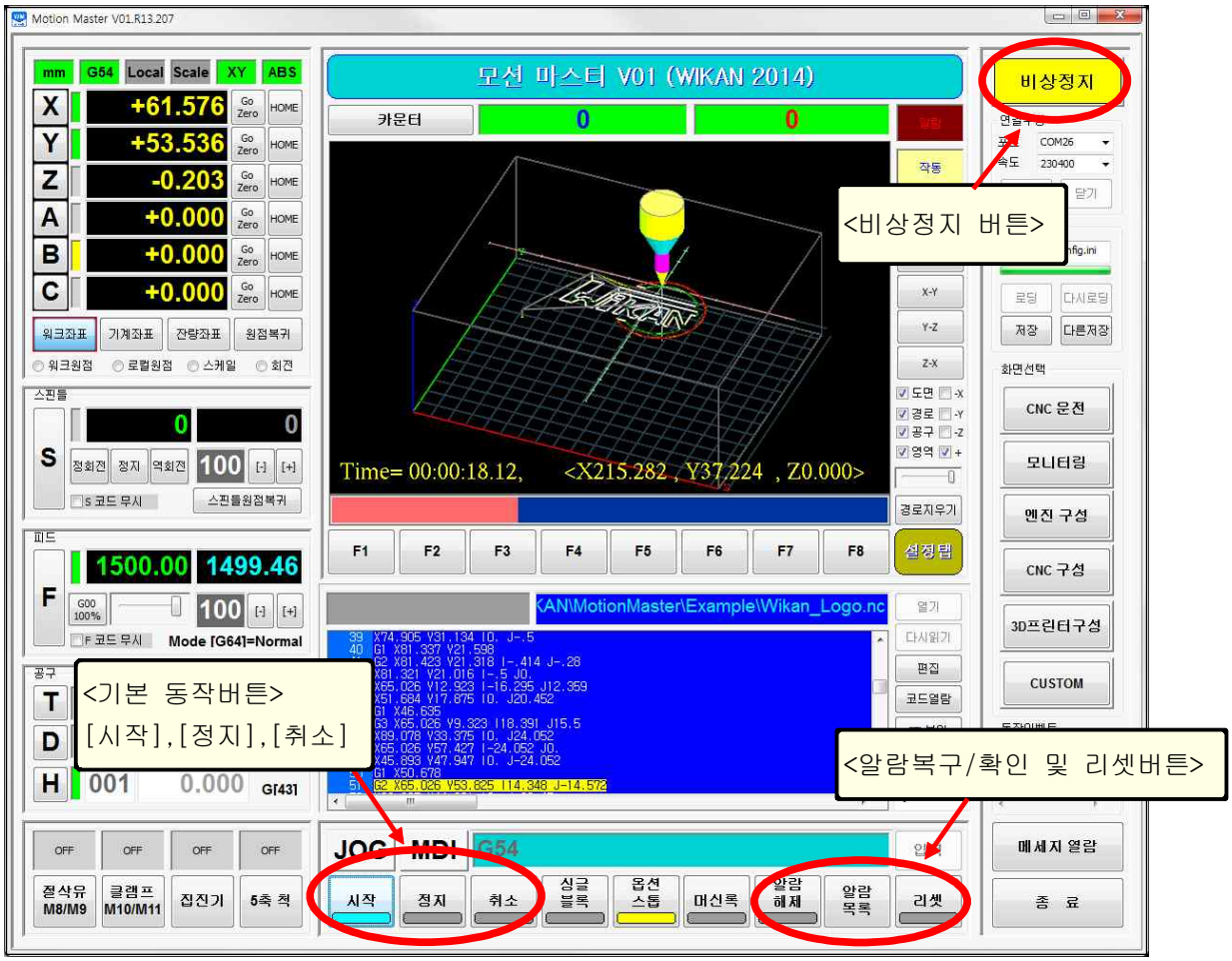
5.5.5. 실행 및 완료

가공 초기위치의 이동과 워크좌표의 설정이 완료되었다면 G코드파일을 자동으로 실행해 봅니다.
(※만약 [머신록]버튼을 눌러 현재 머신록 상태일 경우 [시작]버튼을 눌러 실행하면, 실제기계는 움직이지 않으면서 모션마스터프로그램상의 좌표와 3D화면은 동작그대로의 움직임을 보여줍니다. 동작을 확인하면 [머신록]버튼을 다시 눌러주어 해제시켜 주어야 합니다.)

[시작]버튼을 눌러주면 3D뷰화면에서 공구의 움직임을 확인해 볼 수 있습니다.
[일시정지]버튼을 누르면 "-PAUSE-" 메시지가 점멸하면서 동작이 일시정지하게 됩니다.
다시 [시작]버튼을 누르면 동작을 재개합니다.
만약 일시정지상태에서 [취소]버튼을 누르면 현재 G코드프로그램 동작전체를 취소하게 됩니다.

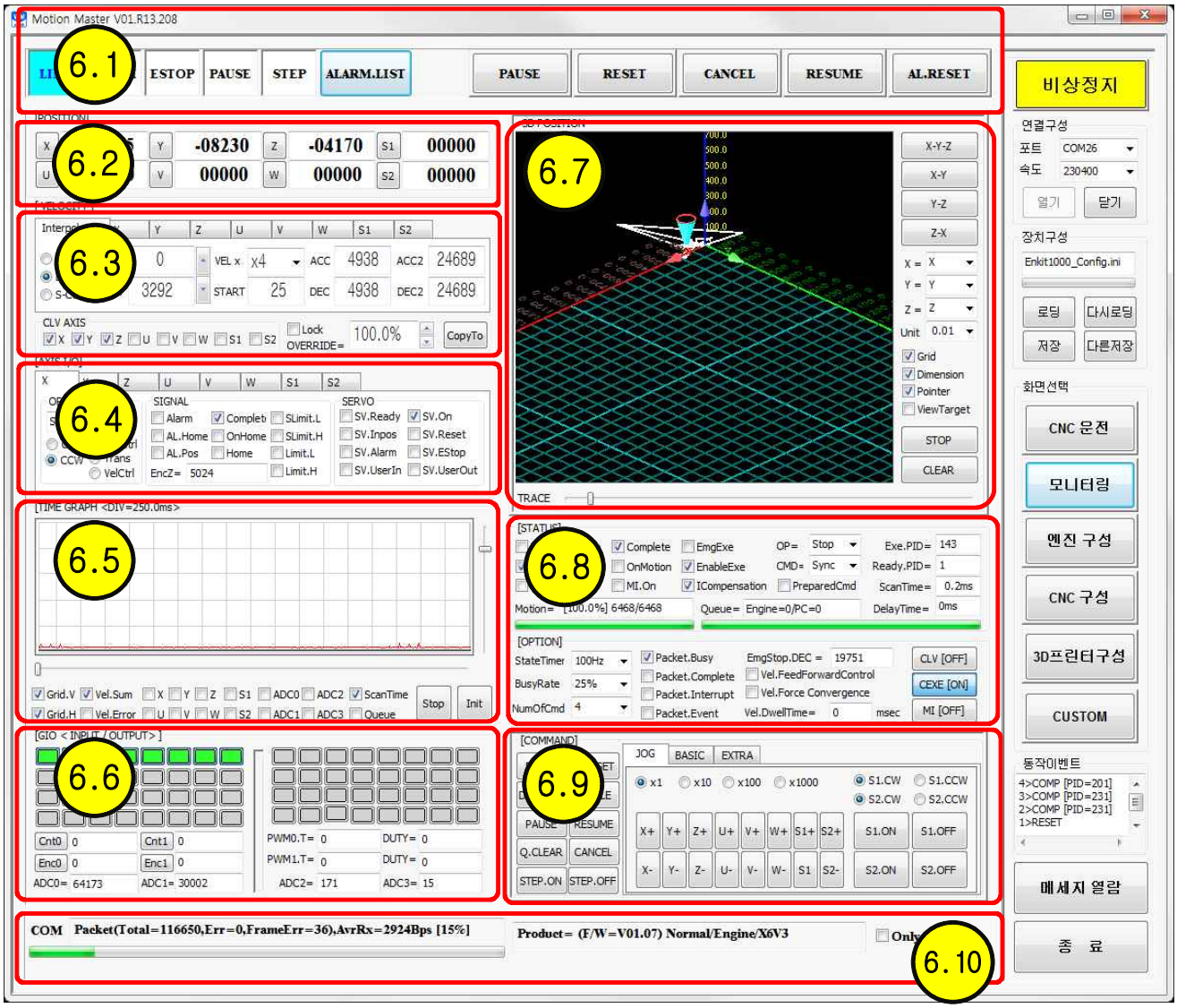
동작중 언제라도 [비상정지]버튼을 눌러 급정지할 수 있습니다. 일시정지동작은 알람상태가 아니며 정지시 현재 동작 감속도비율로 감속하며 정지합니다. 반면에 [비상정지]버튼을 눌러 정지하게되면 "CNC구성"에서 설정된 EmgStop감속비율로 급감속하며 정지하고, 알람상태가 됩니다.
이 상태에서 동작을 재개하려면 [알람해제]버튼을 눌러주어 알람이 해제가능한지 확인해 봅니다. 알람이 해제되었다면 다시 [알람해제]버튼을 눌러 알람해제 이전상태로 되돌아가 가야합니다. 그리고 [시작]버튼을 눌러 주면 자동동작이 재개됩니다.

※만약 [알람해제]버튼을 눌러도 알람상태가 해제되지 않는다면 알람을 발생시킨 원인을 확인하고 이유를 제거해야만 알람해제가 가능합니다. 알람해제상태 가능하지 않다면 [리셋]버튼을 눌러 현재프로그램실행전체를 취소하고 다시 리셋상태로 돌아갈 수 있습니다. 리셋상태로 되돌아가더라도 특정알람상태가 해결되지 않으면 계속 알람상태를 유지할 수도 있습니다.



6. 모니터링 화면

모션엔진 모니터링 화면은 모션컨트롤러의 기본적인 동작상태들을 확인하고 파라메터를 실시간으로 변경할 수 있는 화면입니다. 모니터링 화면은 위칸모션컨트롤러에 내장된 X6V3계열 엔진의 동작을 보여주는 화면으로써 동작들에 대한 다양한 정보들(위치,속도, 통신, 명령 및 IO상태)을 자세하게 보여줍니다.



6.1. 모션엔진의 주요 상태확인 및 시스템명령

6.1.1. LINK

모션보드와의 연결상태를 표시합니다. 연결되지 않을 경우 점멸합니다. 연결되면

상태로 점등합니다.

6.1.2. ALARM

모션보드에 알람이 발생할 때

상태로 점멸합니다. 알람 해제시 [알람리셋]

버튼을 클릭 합니다. 그러면 알람이 해제되고 PAUSE 상태가 됩니다. PAUSE 상태에서 다음동작을 계속하려면 [동작재개]

버튼을 클릭하며, 이어지는 동작들을 취소하려면 [동작취소]

버튼을 클릭합니다.

※ 알람의 원인이 해제되지 않았다면 [알람리셋] 버튼을 클릭하여도 해제되지 않을 수 있습니다.

※

[알람리스트] 버튼을 클릭하면 알람의 원인이 확인됩니다. 이들 중에서 PROCESS ERROR가 발생할 경우 [리셋] 혹은 [알람리셋]으로는 정확하게 복구될 수 없으므로 모션보드의 하드웨어 리셋을 실행해야 합니다. 하드웨어 리셋스위치를 눌러주거나, 리셋스위치가 없다면 전원을 껐다 잠시 후 다시 켜주어야 합니다.

6.1.3. ESTOP

모션보드가 외부의 신호에 의해 비상정지 될 때

상태로 점멸합니다.
ESTOP 상태의 해제도 알람상태의 해제방식과 동일합니다. 단, [알람리셋] 버튼의 클릭없이 바로 [동작재개] 혹은 [동작취소]를 실행할 수 도 있습니다.

6.1.4. PAUSE

모션보드가 일시정지상태가 될 때

상태로 점멸하게 됩니다.
PAUSE 상태는 ESTOP과 동일하게 모션동작을 중지시킵니다. 그러나 알람을 발생하지 않으므로 [알람리셋]이 필요하지 않고 바로 [동작재개]

혹은 [동작취소]

명령을 선택할 수 있습니다.

6.1.5. STEP

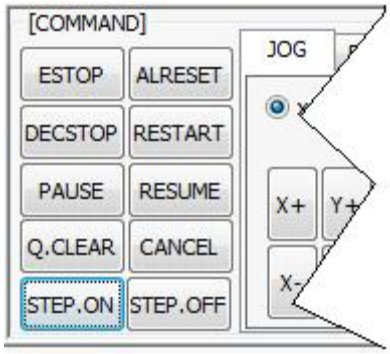
모션보드가 STEP동작 상태일 때

상태로 점멸합니다.
STEP 동작상태는 STEP동작을 활성화시키는 명령에 의해서 실행됩니다. 이 명령은 [COMMAND] 창내에 [STEP.ON] 버튼으로 실행됩니다. STEP 명령상태가 되면 모션엔진내의 명령큐실행은 하나씩 실행되는 상태가 됩니다.

이때 명령큐의 다음 스텝(STEP)을 실행하는 명령은 [동작재개]

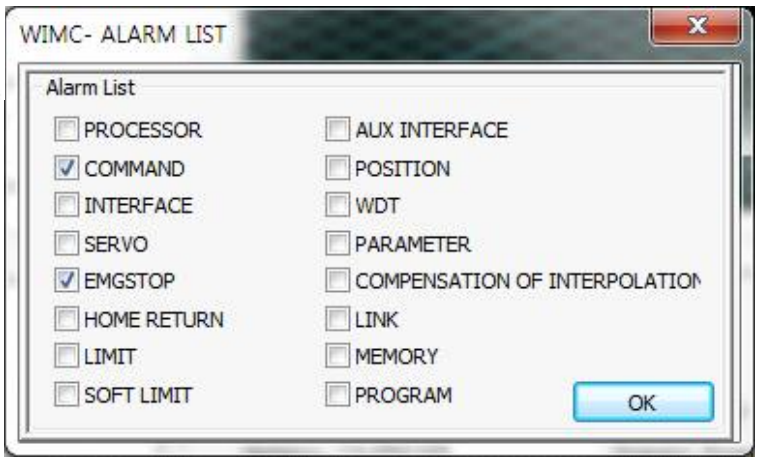
명령이며 다음 스텝을 취소하는 명령은 [동작취소]

명령입니다.
[STEP.OFF] 명령으로 STEP 명령상태를 해제할 수 있습니다.



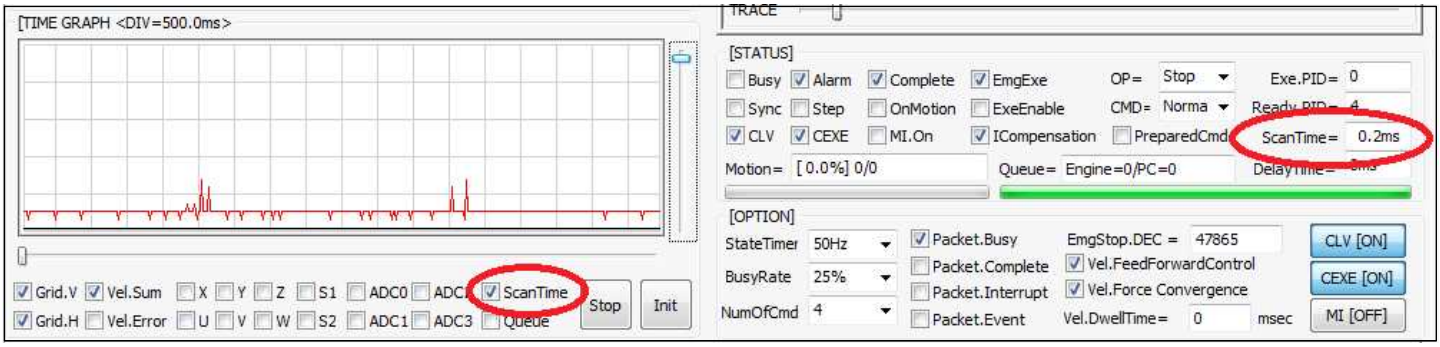
6.1.6. ALARM LIST

[알람리스트] ALARM.LIST 버튼을 클릭하면 현재 알람상태일 때 알람의 발생 원인을 표시하는 창이 나타납니다.



● PROCESSOR : 내부 명령큐 오버플로우 등의 시스템 문제가 발생될 때.
⇒복구> 하드웨어 리셋 필요.
● COMMAND : 지원되지 않는 명령이나 명령의 형식(축지정 등의)이 적절하지 않을 때.
⇒복구> [알람리셋] 후 [동작재개] 혹은 [동작취소] 명령
● INTERFACE : PC와 같은 상위제어기와 모션엔진의 메인통신포트와의 통신장애
⇒복구> [알람리셋] 후 [동작재개] 혹은 [동작취소] 명령
※ 주의> 통신장애는 통신커넥터의 접속불량, 주변노이즈환경, 리셋 동작시 일시적 통신불량상태 등 다양한 환경에서 발생될 수 있습니다. 기계의 초기상태라면 단순히 [알람리셋]이나 다시 초기화 할 수 있지만 기계명령의 지속적인 실행상태(명령큐의 실행상태) 도중에 발생하였다면 알람을 복구하더라도 알람상태를 발생시킨 일부 명령이 실행되지 못했으므로 현재 명령큐의 연속된 동작명령의 결과는 목적하는 결과와 차이를 발생할 수 있습니다.
● SERVO : SERVO ALARM신호가 발생되었거나 SERVO INPOS신호가 특정시간내에 입력되지 않을 때 발생.
● EMGSTOP : 비상정지 상태일 때.
단독으로 EMGSTOP이 상태가 체크[√]되었다면 외부 EMGSTOP신호에 의해서 발생된 것이며,
COMMAND 오류와 함께 EMGSTOP 오류가 같이 체크[√]되었다면 EMGSTOP 통신명령에 의해서 발생된 것입니다.
⇒복구> [알람리셋] 후 [동작재개] 혹은 [동작취소] 명령
● HOME RETURN : 원점복귀시 원점복귀동작의 최대 이동거리(탐색거리)를 벗어났거나, 원점복귀완료지점의 위치오차가 지정된 량보다 클 때 발생.
⇒복구> [알람리셋] 후 [동작재개] 혹은 [동작취소] 명령
● LIMIT : 외부 리미트 신호에 의해서 발생됩니다. 리미트신호에 의해서 알람이 발생하기 위해서는 LIMIT신호의 활성화(ENABLE)가 필요합니다.
⇒복구> 먼저, 리미트신호의 발생원인을 제거하거나, LIMIT신호를 비활성화(DISABLE)상태로 만들든 후에 [알람리셋] 과 [동작재개] 혹은 [동작취소] 명령을 실행합니다.
● SOFT LIMIT : 내부의 소프트리미트 위치지점을 벗어날 경우 알람이 발생합니다. SOFT LIMIT 알람을 발생하기 위해서는 SOFT LIMIT기능을 활성화(ENABLE)하여야 합니다.
⇒복구> 먼저, SOFT LIMIT 신호를 비활성화(DISABLE)상태로 만들든 후에 [알람리셋] 과 [동작재개] 혹은 [동작취소] 명령을 실행합니다.
● AUX INTERFACE : 보조연결포트 인터페이스 오류 발생, 보조연결포트를 지원하는 모션엔진의 경우에 발생하며 내용은 INTERFACE 오류와 동일합니다.

● POSITION : 위치제어 동작중에 발생된 오류입니다. 현재 SERVO INPOS 동작이 활성화(ENABLE)된 상태에서 지정된 시간내에 발생되지 않을 때 POSITION ALARM이 발생합니다.
⇒복구> [알람리셋] 후 [동작재개] 혹은 [동작취소] 명령
● WDT : 모션엔진의 내부 실행속도가 현저하게 떨어질 때 발생합니다. 모션엔진의 내부 실행속도는 ScanTime이라는 내부 파라메터를 통해서 확인할 수 있습니다.



ScanTime이 모션엔진 내부에 판단한 특정시간대역보다 너무 길어졌다면 모션엔진의 실행능력이 떨어진 것으로 간주하여 Watch Dog Timer [WDT] 알람을 발생합니다. 모션엔진의 실행능력이 떨어지면 통신 응답성들의 상태들도 나빠지게 됩니다. 대부분 모션실행방식(동작축수, 전자기어비사용유무, 백래시사용유무)에서 동작주파수(PPS)가 너무 높을 때 발생됩니다. 일반적인 알람복구를 수행하기 전에 모션엔진이 적절한 속도(ScanTime < 1~2msec)를 유지할 수 있도록 옵션들을 조절합니다.
⇒복구> [알람리셋] 후 [동작재개] 혹은 [동작취소] 명령

● PARAMETER 오류 : 명령 파라메터 지정이 잘못될 때 발생합니다. 이 오류는 모션엔진에 따라 발생하지 않을 수 있습니다.
⇒복구> [알람리셋] 후 [동작재개] 혹은 [동작취소] 명령

● COMPENSATION OF INTERPOLATION : 보간동작 중 보상제어동작의 실패로 발생.
예> 원호보간의 경우 중심좌표와 종점좌표가 적절하지 않아서 모션실행결과의 종점좌표가, 명령으로 지정한 종점좌표와 일치 하지 않을 수 있습니다. 이때 종점좌표의 위치차이가 2PULSE 이상의 값이 발생하면 [보간보상알람]이 발생합니다.
이 알람은 기계를 비상정지 시키지 않으므로 필요시 사용자 개발프로그램에서 동작정지를 처리해 주어야 합니다.
⇒복구> [알람리셋] 후 [동작재개] 혹은 [동작취소] 명령

● LINK : 외부부가장치와의 링크에서 문제가 발생되었을 때 발생합니다. (현재 사용되지 않습니다)
● MEMORY : 외부 메모리장치이 동작에 이상이 있을 때 발생합니다. (현재 사용되지 않습니다)
● PROGRAM : 내부 프로그램기능 동작에 이상이 있을 때 발생합니다. (현재 사용되지 않습니다)
- 추가적으로 조건 Delay명령에서 조건이 만족되지 않은채 지정된 시간 이상 초과될 때 Command Error와 함께 발생됩니다.
⇒복구> [알람리셋] 후 [동작재개] 혹은 [동작취소] 명령

6.1.7. 리셋버튼

알람리셋[AL.RST]버튼으로 알람이 해제되지 않는다면, [리셋]을 눌러 엔진을 초기화 할 수 있습니다. 이때 아래와 같은 메세지가 몇 차례 나타날 수 있습니다. 이들 메시지는 CNC 동작프로그램에서 예상치 못한 리셋신호가 발생했음을 알려주는 메시지입니다.

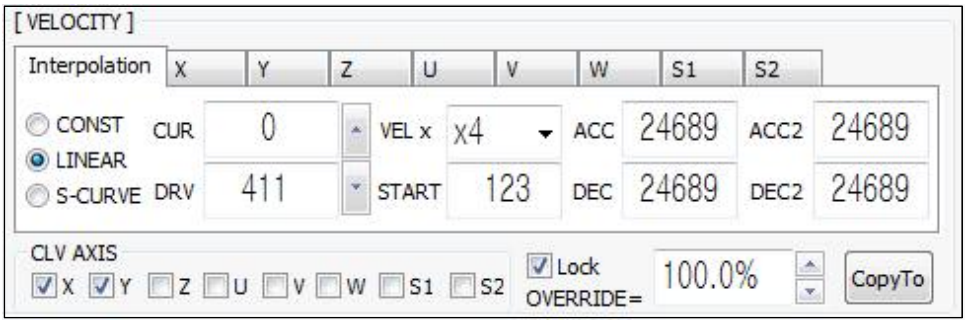
[확인버튼]을 눌러 준 후 선택화면의 RELOAD 버튼을 눌러 구성설정의 다시 로딩합니다.

Result Message
Message 1> Machine has been reset!
Message 2> Emergency! External Reset occured
확인

6.2. 위치좌표 확인/설정

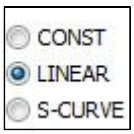
모션엔진의 각 축별 현재위치를 표시합니다. 각 축별 NAME 버튼을 클릭하면 현재위치는 0으로 리셋됩니다.
위치 변경시 마우스를 표시창에 두고 임의값으로 변경할 수 있습니다.
변경시 안전한 연산 위치 영역은 -8,388,607 ~ +8,388,607 (24비트영역)입니다.

6.3. 속도기능 확인/설정



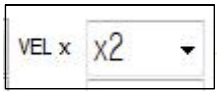
속도기능 창은 총 9개의 탭(Interpolation, X,Y,Z,U,V,W,S1,S2)으로 이루어집니다.
Interpolation 탭은 보간명령에 의해 동작될 때 지정되는 속도성분들이며, 그 외 개별축 속도성분은 개별적인 축동작명령들(MOV계열, SPINDLE회전명령 등)에 의해 동작될 때 지정되는 속도성분들입니다.

6.3.1. 속도모드



- CONST : 고정 속도모드 = 가감속없이 바로 구동속도로 동작, 급정지
- LINEAR : 1차 선형 가감속도모드 (사다리꼴가감속 : Trapzoidal)
- S-CURVE : 2차 S자 가감속도모드 (S커브 가감속 : S-Curve)

6.3.2. 속도배율

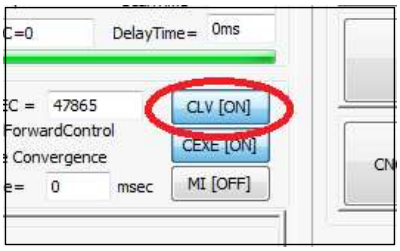


속도배율은 2의 n승 배율이 적용됩니다. 속도배율은 각 축마다 적용되지 않고, 하나의 적용 값이 전체에 적용됩니다. 단 스핀들의 회전명령은 고정배율 x16 이 적용됩니다.

6.3.3. 선속도 일정축

: 직교좌표계를 구성하는 축을 선택합니다.

왼쪽 두 그림처럼 선속도 일정축을 선택하고 , [선속도 일정]=[CLV.ON] 모드이면 보간동작의 구동속도는 보간주축의 속도가 아닌 보간합성축의 속도가 됩니다. 선속도일정모드는 [OPTION]창의 [CLV ON]버튼을 클릭하면 활성화(ENABLE)혹은 비활성화(DISABLE)할 수 있습니다.



보간주축은 특정보간동작에서 상대적으로 가장 많은 펄스를 발생하는 축을 말합니다. 즉 X축으로 100펄스, Y축으로 300펄스를 움직이는 직선보간 명령이 있다면 Y축이 현재명령의 보간주축이 되며 선속도 일정모드가 아니면 Y축의 속도가 구동속도가 됩니다. 원호보간의 경우 원호의 각도에 따라 보간주축은 계속 바뀝니다. 그러므로 선속도 일정모드가 아니면 원호의 8사분면 구간별로 구동속도가 계속 달라지게 됩니다. 그러나 선속도 일정모드이면 보간주축에 관계없이 경로에 대한 곡선속도(선속도)가 구동속도에 일치하도록 움직입니다.

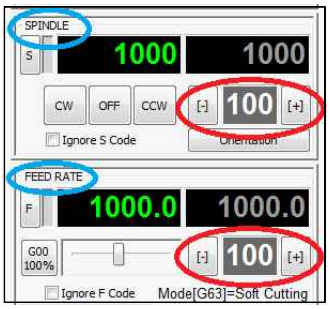
이러한 동작은 모션엔진 내부에서 보간동작을 실행할 때, 선속도 일정축으로 지정된 축들 중 1축만 움직이는 영역에서는 구동속도의 x 1배, 2축이 동시에 움직이는 영역에서는 1/√2 배, 3축이 동시에 움직이는 영역에서는 1/√3 배의 속도로 움직임으로써 구현됩니다. (물론 이들 선속도 일정축은 직교좌표계를 기반으로 합니다.)

일반적으로 움직이는 경로내의 속도가 중요하지 않고 최종점의 위치만이 고려되는 보다 빠른 움직임이 필요할 때는 선속도 일정모드를 사용하지 않으며, 기계가공 과 같이 움직이는 속도가 일정하게 유지되어야 할 경우에 선속도 일정모드를 사용합니다.

6.3.4. 속도 오버라이딩

: 보간동작의 지정된 속도를 배율값으로 조정합니다.

구동속도의 배율을 지정하여 구동속도의 변경없이 전체동작속도를 변화시킬 수 있습니다. 실제 모션엔진은 보간속도의 오버라이딩과 스핀들 회전속도의 오버라이딩이 따로 구분되어 있으나 모니터링 프로그램은 이들을 하나의 명령으로 동시에 배율 조정합니다. Lock기능은 보간속도 오버라이딩기능에 영향을 받지 않고 항상 100%로 동작하도록 하는 기능입니다. Lock기능은 스핀들 회전속도에는 영향을 주지 않습니다.

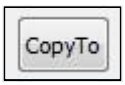


왼쪽 그림처럼 CNC 조작화면에서는 구분을 두어 스핀들과 이동동작의 개별적인 오버라이딩 기능으로 나누어져 있는 것을 확인할 수 있습니다.

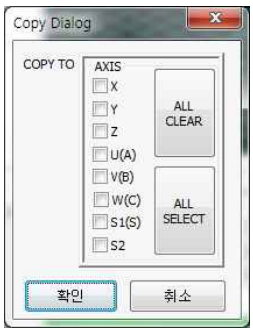
6.3.5. 속도성분들

● 초속도 (START): 모션동작이 시작되고, 종료되는 속도로써 Pulse / sec [PPS] 혹은 [Hz]단위입니다.
● 1차 가속도 (ACC):모션동작이 가감속모드일 때 1차 가속도 값으로써 PPS/sec입니다.
● 1차 감속도 (DEC):모션동작이 가감속모드일 때 1차 감속도 값으로써 PPS/sec입니다.
● 2차 가속도 (ACC2):모션동작이 2차 가감속모드(S커브 가감속모드)일 때 2차 가속도 값으로써 PPS/sec² c입니다.
● 2차 감속도 (DEC2):모션동작이 2차 가감속모드(S커브 가감속모드)일 때 2차 감속도 값으로써 PPS/sec² c입니다.
● 구동속도 (DRV): 모션동작의 최종 목표도달 속도입니다.
● 현재속도 (CUR): 현재 속도값으로써 읽기전용입니다. 현재속도값은 배율이 적용된 값을 표시합니다. 스핀들회전명령의 경우 x16 배율이 적용되어 표시됩니다.

6.3.6. 속도항목 빠르게 다른축에 복사하기

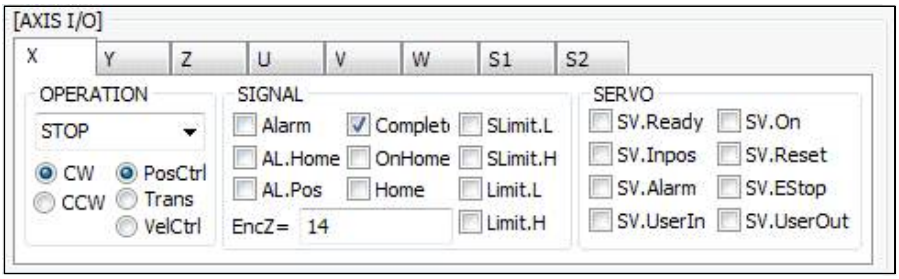


버튼을 누르면 아래와 같은 복사할 대상축 선택창이 나옵니다.



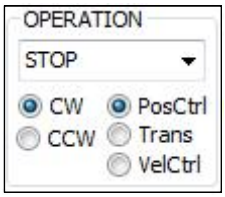
복사대상축을 선택하고 [확인]버튼을 누르면 현재 탭의 속도성분값들이 복사대상축에 복사 됩니다.

6.4. 축별 동작상태 확인



각 축별로 지원되는 다양한 상태들을 확인할 수 있습니다. 모션엔진의 종류에 따라서 지원되는 축의 종류와 해당 축에서 지원하는 상태값 등의 차이가 있을 수 있습니다.

6.4.1. 동작상태



● 현재축의 동작상태가 정지시=STOP, 가속동작시=ACC, 감속동작시=DEC, 안정속도도달시=STEADY를 표시합니다.

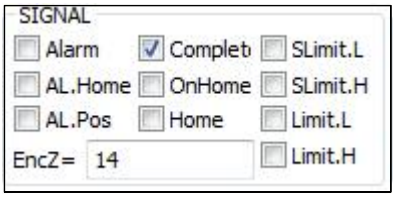
● CW는 (위치 +증가)방향으로 회전함을 지시하고,CCW는 (위치-감소)방향으로 회전함을 지시합니다.

● PosCtrl 은 현재 축의 제어동작이 위치제어를 위한 동작임을 가리키고, VelCtrl은 현재 축의 제어동작이 단순한 회전목적인 속도제어를 위한 동작임을 가리킵니다. Trans는 PosCtrl에서 VelCtrl로 동작이 바뀌거나, VelCtrl에서 PosCtrl로 동작이 바꾸는 과정에서 있음을 알립니다.

※ 스핀들의 경우 일반적인 절삭시에는 고속의 회전제어을 요구하지만 탭핑과 같은 작업에서는 Z축과의 선형 보간동작을 요구합니다. 이때 고속회전중인 스핀들을 바로 보간동작에 적용하면 급격한 속도변화가 일어납니다. 모션엔진은 고속회전중인 스핀들을 자동으로 1차 감속 정지시키고, 완전히 정지된 후에 직선보간 동작인 위치제어동작으로 자동으로 전향되도록 할 수 있습니다. 이때 PosCtrl/Trans/VelCtrl의 각 상태값을 얻으면 현재 어떤 동작상태인지를 확인할 수 있습니다.

6.4.2. 축 신호상태

: 각각의 축 신호상태를 표시합니다.



● Alarm : 각축의 동작 상태에서 발생된 알람신호를 표시합니다. 예를 들면, 특정축의 원점복귀 동작중 알람이 발생되었다면, 우선 전체알람신호로써 현재 알람상태임을 확인할 수 있고, 다시 각 축별 알람신호를 확인함으로써 어느 축의 원점복귀 동작에서 알람이 발생되었는지 확인할 수 있습니다.

● AL.Home : 원점복귀중 알람발생 표시

● AL.Pos : 서보의 Inpos 자동화 동작에서 알람발생 표시

● Complete : 현재축의 모션동작이 완료되었음을 표시

● OnHome : 현재축이 원점복귀동작 중임을 표시

● Home : 현재축의 원점센서의 신호상태를 표시

● SLimit.L : 현재축의 Low 소프트리미트 감지상태 표시

● SLimit.H : 현재축의 High 소프트리미트 감지상태 표시

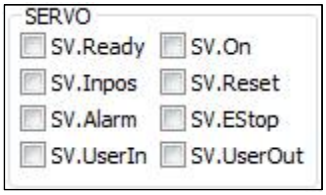
● Limit.L : 현재축의 Low 리미트센서의 신호상태를 표시

● Limit.H : 현재축의 High 리미트센서의 신호상태를 표시

※ X6V3엔진은 Limit.L신호와 Limit.H신호가 구분되어 있지않고 각 축마다 Limit신호 하나로 통합되어 있습니다. 이때 현재 동작을 정지시킨 원인이 Limit.L인지 Limit.H인지의 구분은 실제 정지시의 축이동방향(CW, CCW)을 확인하여 구별하거나, 현재 위치정보로써 추정할 수 있습니다.

6.4.3. 서보신호 상태

: 각 축의 서보드라이브 신호상태를 표시합니다.



- SV.Ready : (입력) 서보드라이브의 동작준비완료 신호 입력
- SV.Inpos : (입력) 서보드라이브의 추종동작(서보동작)완료 신호 입력
- SV.Alarm : (입력) 서보드라이브의 알람신호 입력
- SV.UserIn: (입력) 사용자가 임의적으로 사용할 수 있는 서보드라이브의 신호 입력
- SV.On : (출력) 서보드라이브에게 모터전원을 인가하라는 신호 출력
- SV.Reset : (출력) 서보드라이브의 초기시작 및 알람 등의 이유로 리셋을 요구하는 신호 출력
- SV.EStop : (출력) 서보드라이브에게 현재동작을 비상정지하라는 신호 출력
- SV.UserOut: (출력) 사용자가 임의적으로 사용할 수 있는 서보드라이브에 대한 신호 출력

※ X6V3엔진의 경우 서보와 관련된 신호는 SV.Reset, SV.Alarm, SV.On, SV.Inpos의 신호만 존재하며, 이 신호들은 각각의 축마다 지원되지 않고 모든 축에 OR 된 결합으로 지원되고 있습니다.

6.5. 시간 그래프(Time Graph) 모니터링

모션엔진의 내부상태를 시간대역별로 확인할 수 있습니다. [Stop] 버튼을 누르면 현재그래프표시가 정지됩니다. 시간흐름을 재개하려면 다시 버튼을 눌러주면 됩니다. [Init] 버튼은 화면이 정지하였거나 확대 스케일값을 초기화할 때 사용합니다.

시간그래프의 표시항목은 아래와 같습니다.

● Grid.V : 수직 그리드 표시
● Grid.H : 수평 그리드 표시
● Vel.Sum : 보간동작 속도 표시
● Vel.Error : 속도 P제어 오차값 표시
● X,Y,Z,U,V,W,S1,S2 : 각축의 속도 표시
- ADC0, ADC1, ADC2, ADC3 : 모션엔진에 내장되 ADC(아날로그->디지탈변환 채널) 값표시
- ScanTime : 모션엔진의 메인제어루프 실행시간
- Queue : 모션엔진의 현재 사용된 명령큐의 크기

6.6. IO(입출력) 포트의 모니터링

6.6.1. GIO 입력포트

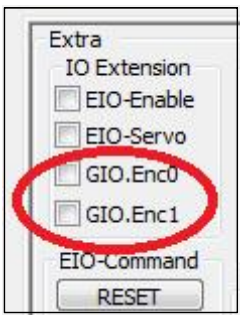
X6V3엔진은 기본적으로 8개의 디지털 입력포트를 갖습니다.

8개의 입력포트를 각각 0, 1, ..., 7번으로 할 때 특정 입력포트는 일반적인 입력포트이외에 카운터입력과 엔코더 입력등의 부수적인 기능들과 중첩하고 있습니다.

X6V3 엔진의 경우 입력포트 기능은 아래와 같습니다.

기능 / 포트	7	6	5	4	3	2	1	0
일반 디지털입력포트	○	○	○	○	○	○	○	○
UP COUNTER 채널 #1 (16비트 카운터)								UP신호
UP COUNTER 채널 #2 (16비트 카운터)							UP신호	
4체배 엔코더카운터 #1 (32비트 카운터)					B상	A상		
4체배 엔코더카운터 #2 (32비트 카운터)			B상	A상				

※ 4체배 엔코더카운터의 기능을 사용하기 위해서는 “모션엔진 구성설정 화면”의 Extra 설정란의 GIO.Enc0 와 GIO.Enc1 항목을 활성화 되어 있어야만 동작됩니다.



6.6.2. ADC 입력

X6V3엔진에는 기본적으로 하드웨어 10비트분해능을 갖는 ADC컨버터가 4채널이 내장되어 16비트로 확장되어 연결되어 있습니다.

6.6.3. GIO 출력포트

X6V3엔진은 기본적으로 8개의 디지털 출력포트를 갖습니다.

각 출력포트의 값은

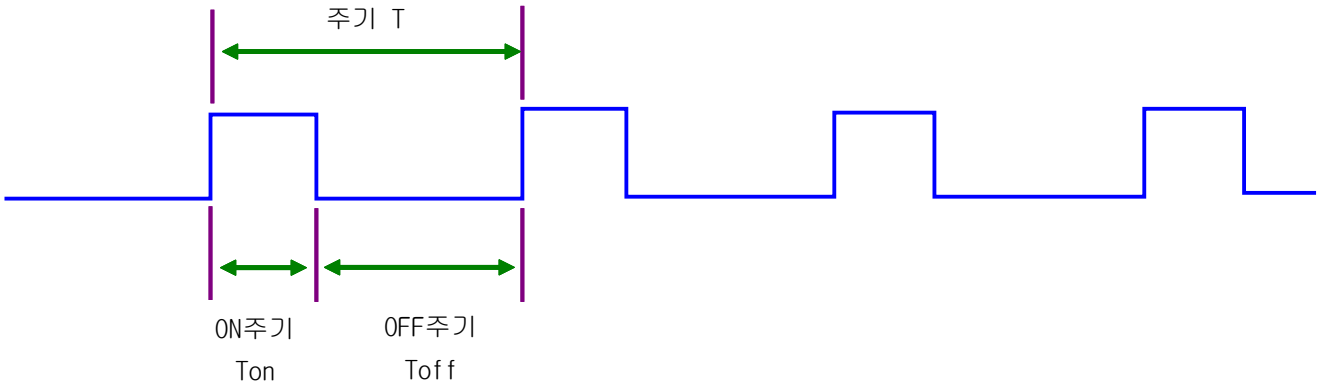
중 하나를 클릭함으로써 변경할 수 있습니다.

출력포트 2,3,4,5 번은 X6V3엔진에 내장된 4개의 PID 제어기 출력용 PWM신호로 사용될 수도 있습니다.

자세한 내용은 “(4.9.)모션 명령 실행” 편의 “(4.9.4.)추가명령동작” 을 참조하세요.

6.6.4. PWM 출력

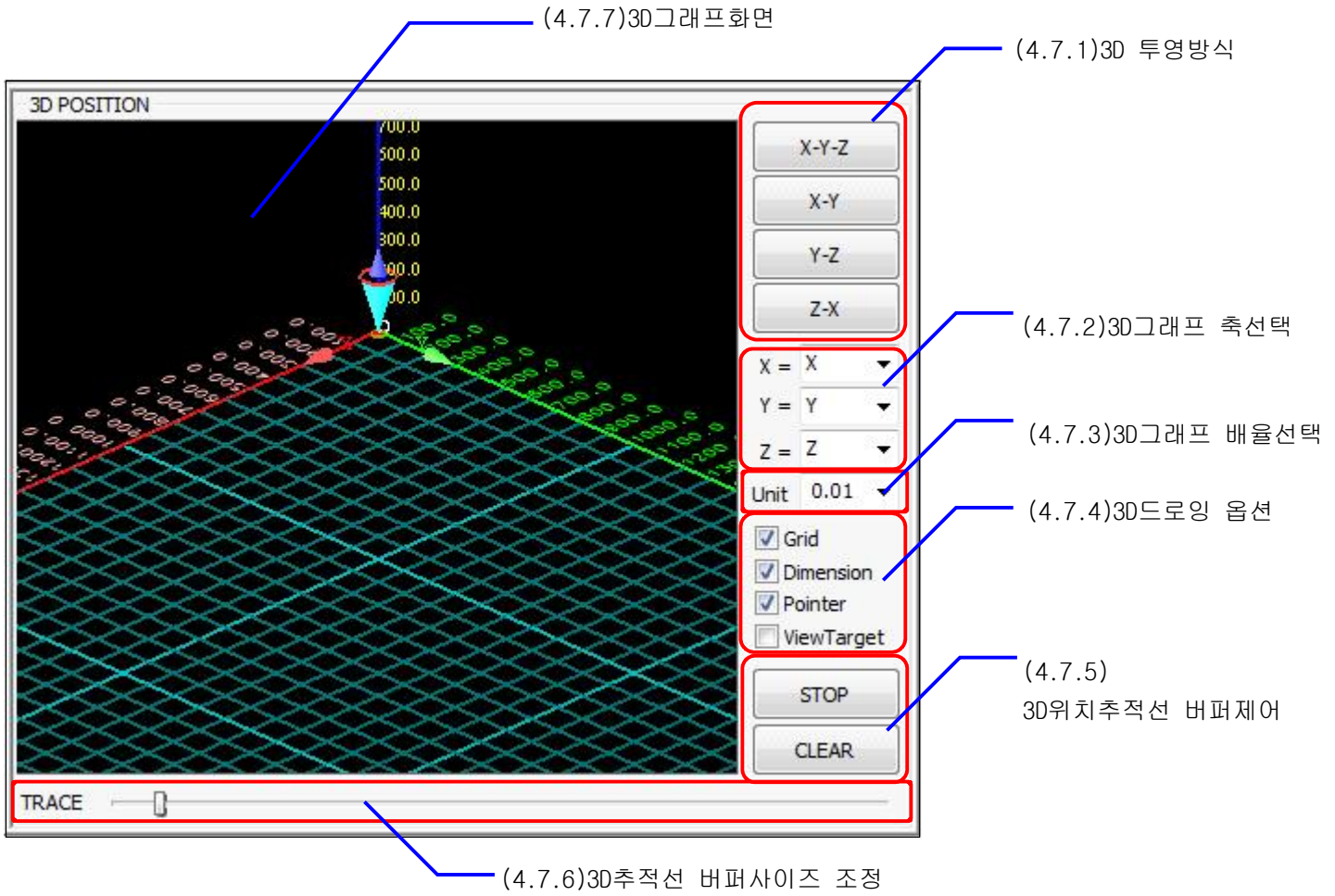
X6V3엔진은 PWM출력 2채널을 갖고 있습니다. PWM출력은 완전한 하드웨어방식이 아닌 내부타이머에 의한 동작방식으로 +-0.1%의 출력오차를 갖고 있습니다. PWM 동작 기준주파수는 20kHz입니다.



- PWM 주기 T = PWM0.T= 400 / 20kHz [sec]
- PWM ON 주기 Ton = DUTY= 370 / 20kHz [sec]
- PWM Duty Ratio = Ton / T x 100 [%]

6.7. 3D 그래프 위치 모니터링

모션의 현재위치좌표값을 3D 그래프로 위치추적을 할 수 있습니다. 추적데이터는 상태타이머의 Hz 에 의존해서 위치값이 변화 될 때 마다 기록하여 3D 좌표상의 직선(Line)으로 표시합니다.



6.7.1. 3D 투영방식

: 3D투영방식은 기본적으로 원근표현이 없는 ISOMETRIC 투영방식입니다.

●X-Y-Z : XYZ공간투영
●X-Y : XY평면투영
●Y-Z : XY평면투영
●Z-X : XY평면투영

6.7.2. 3D그래프 축선택

: 모션보드의 임의축을 그래프상의 X,Y,Z 축과 맵핑합니다.

6.7.3. 3D그래프 배율선택

: 1펄스의 단위를 0.001부터 1.0까지 10배율단위로 선택할 수 있습니다.

6.7.4. 3D드로잉 옵션

● Grid : XY평면의 그리드 표시
● Dimension : X,Y,Z 축상의 치수 표시
● Pointer : 현재위치 지시침 표시
● View Target : 3D 투영의 관찰대상(Viewing Target) 위치에 마크 표시

6.7.5. 3D 위치추적선 버퍼제어

● [STOP] : 3D 추적선 버퍼의 위치내용이 상태타이머의 정보갱신에 의해 변경되지 않도록 정지시킵니다.
● [CLEAR] :3D 추적선 버퍼의 내용전체를 지웁니다.

6.7.6. 3D 위치추적선 버퍼사이즈 조정

● 3D 화면에 표시할 위치 추적선의 길이를 조절합니다.

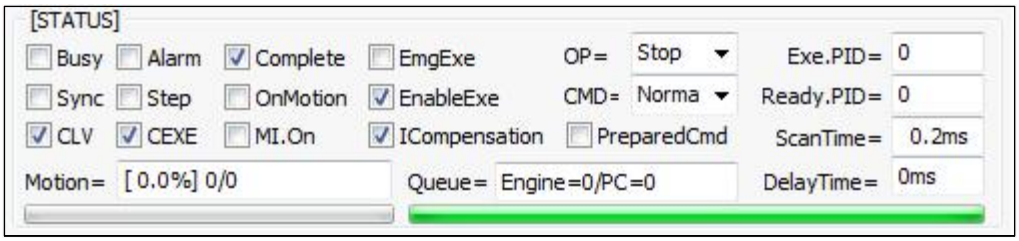
6.7.7. 3D 그래프 화면

● 줌 인/아웃(Zoom in/out) : 마우스 스크롤
● 이동 : 마우스 왼쪽버튼 클릭 + 마우스 이동
● 회전 : 마우스 중간(휠)버튼 클릭 + 마우스 이동
(경도변화 : 마우스 중간(휠)버튼 클릭 + 마우스 수평이동)
(위도변화 : 마우스 중간(휠)버튼 클릭 + 마우스 수직이동)

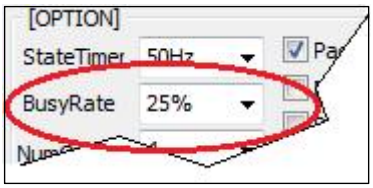
※ 회전의 중심은 View Target으로 확인하세요,
※ View Target을 관찰하고자 하는 대상위치의 중심에 두면 편리합니다.

6.8. 모션엔진의 동작상태 확인 및 동작옵션 설정

6.8.1. 모션엔진 동작상태(STATUS) 항목설명



● Busy : 수신된 명령들의 데이터가 엔진 내부 수신메모리의 지정된 %를 초과할 때 Busy신호가 발생합니다.

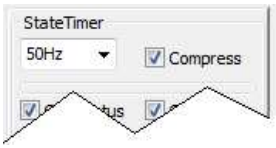
Busy신호 발생의 기준이 되는 수신메모리의 크기는 [OPTION]창의 

BusyRate 항목에서 25%/50%/75%/100% 중 하나를 선택할 수 있습니다(기본적으로 25%로 설정합니다).

● Alarm : 알람발생을 표시합니다.
● Complete : 보간동작이 완료되었음을 표시합니다.
● EmgExe : 현재 일시적 중지상태로 정상적인 실행상태가 아닌 비상실행상태임을 표시합니다.
● Sync : 모션엔진의 sync신호상태를 표시합니다.
● Step : 현재 명령실행상태가 step 명령실행상태임을 나타냅니다.
● OnMotion : 현재 모션동작이 실행되고 있는 상태임을 나타냅니다.
● EnableExe : 모든 명령실행이 가능함을 지시합니다.
내부 알람이나 비상정지로 인해 비상실행상태로 진입되었을 때 EnalbeExe가 비활성되어 즉시실행명령만 가능하며 큐명령은 실행할 수 없게 됩니다.
● CLV : 선속도 일정동작모드임를 표시합니다.
● CEXE : 연속실행동작모드임를 표시합니다.
● MI.On : 다중보간동작모드임를 표시합니다.
● ICompensation : 보간보상모드임을 표시합니다.
● PreparedCmd :현재모션이 실행상태일 때 다음명령이 실행준비가 되었음을 표시합니다.
● Op = Stop/Acc/Dec/Steady : 현재 보간동작의 속도상태를 나타냅니다.
● CMD = Normal / Sync / Wait / Delay : 현재실행중인 명령의 종류를 나타냅니다.
Sync/Wait/Delay이외의 모든명령은 Normal상태로 표시됩니다.
● Exe.PID = 0 : 모션동작의 실행명령 패킷ID를 값을 표시합니다.
● Ready.PID = 0 : 현재 모션동작이 실행중일 때 다음명령의 패킷ID값을 가리키거나, 혹은 현재 모션동작이 실행중이 아니고, 현재명령이 보간명령이 아닌 경우에는 현재명령의 패킷ID 값을 표시합니다.
● ScanTime = 0.2ms : 모션엔진의 메인제어루프 실행시간을 표시합니다.
● DelayTime = 0ms : DELAY 명령상태일 때 현재 지연 시간값을 표시합니다. 딜레이시간은 실제 32비트이지만 표시값은 16비트값에 대한 십진수만 표시됩니다.
● Queue = Engine=xx / PC=yy : 현재 모션엔진내부에 대기된 큐명령의 수(Engine=xx)와 PC 소프트웨어(라이브러리)상에서 대기된 명령큐(PC=yy)의 수를 표시합니다.
● Motion [xx.x%] yy/zz : 현재 보간명령의 진행률을 xx.x%로 표시합니다.
yy는 현재 보간명령의 실제 진행된 펄스수이며 zz는 현재명령에 대해 소요되는 펄스수에 대한 추정값입니다. 직선보간과 비트패턴보간의 경우 정확하게 추정하지만, 원호보간의 경우 +/- 8펄스 정도의 오차를 가질 수 있으며, 원점복귀 등의 센서동작에 의존할 경우 정확하지 않은 값을 표시합니다.

6.8.2. 모션엔진 동작옵션(OPTION) 설명

● StateTimer = 25Hz/50Hz/100Hz/200Hz/250Hz/500Hz/1kHz : 상태타이머의 발생주파수를 설정합니다. 상태타이머는 모션엔진이 PC와 같은 상위제어기에게 자동으로 자신의 다양한 정보들을 전달하는 기능을 갖습니다.

전달되는 정보의 종류는 “(5.1.)모션엔진 주요 구성설정 화면”의 “(5.1.4.)상태타이머” 

을 참조하세요.

● BusyRate = 25%/50%/75%/100% : Busy신호를 발생하는 내부수신메모리버퍼의 크기 기준을 지정합니다.

● NumOfCmd = 1,2,3....14,15,16 : 메인제어루프가 1회 실행될 때 최대 실행할 명령의 수를 지정합니다.

● Packet.Busy : Busy상태가 되었을 때 Busy상태패킷을 전송할 지에 대한 유무를 지정합니다.

● Packet.Complete : 모션동작 완료상태가 되었을 때 Complete 상태패킷을 전송할 지에 대한 유무를 지정합니다.

● Packet.Interrupt : 모션엔진의 [STATUS]변화가 발생되었을 때 Interrupt상태패킷을 전송할 지에 대한 유무를 지정합니다.

● Packet.Event : 모션엔진에서 지원하는 이벤트상태가 발생되었을 때 Event패킷을 전송할 지에 대한 유무를 지정합니다.

● EmgStop.Dec = xxxx : 비상정지시의 1차감속도 PPS/sec를 지정합니다. 이 값이 0 이면 즉시정지하게 되고, 0 이 아니면 해당하는 감속도로 1차감속하여 정지합니다.

● Vel.FeedForwardControl : 모션엔진 내부의 속도제어기가 피드포워드제어를 수행하도록 지정합니다.
피드포워드제어를 수행하면 종료시 수렴속도가 초속도 최대한 접근하도록 도와줍니다.

● Vel.Force convergence : 모션엔진 내부의 속도제어기가 현재 설정된 구동속도에 최대한 접근하도록 지정합니다.

● Vel.DwellTime = xxx ms; 보간동작중 속도가 가속과 감속이 변화하는 영역에서 지정된 시간(msec)의 휴지기간을 갖도록 유도합니다. 이 시간값은 정속의 도달시 뿐만아니라 명령의 시작시에도 영향을 줄 수 있어 명령실행을 지연시킬 수도 있습니다. 그러므로 천천히 실행되며 안정하게 실행할 때 DwellTime을 사용할 수 있습니다. 사용하지 않으면 0의 값을 입력합니다.

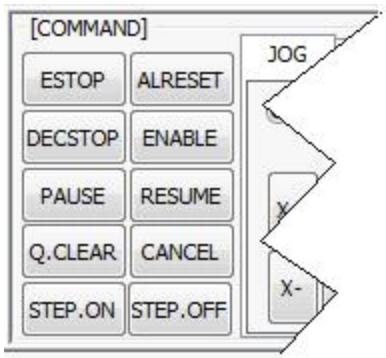
● [CLV ON/OFF] : 선속도일정동작을 ON/OFF합니다.

● [CEXE ON/OFF] : 연속실행동작을 ON/OFF합니다.

● [MI ON/OFF] : 다중보간동작을 ON/OFF합니다.

6.9. 모션 명령 실행

6.9.1. 동작 보조명령



● ESTOP

● ALRESET : 모션엔진 내부에 발생된 알람상태를 클리어합니다.
그러나 알람발생원을 제거하지 않으며 또다시 알람을 발생하게 됩니다.

● DECSTOP : 현재 모션명령을 현재 속도제어방식으로 감속하여 정지시킵니다.
비교적 차분하게 정지시키는 명령입니다.

● ENABLE : 비상정지를 발생시키는 알람이 발생하였거나 비상정지명령을 실행시키면 즉시실행명령 이외의 큐명령은 실행되지 않습니다. 큐명령을 실행하기 위해서 [ENABLE.EXE] 명령으로 큐명령 실행가능하게 하는 역할을 합니다.

● PAUSE : 알람을 발생하지 않고 비상정지상태를 만듭니다. 이 상태를 일시정지 상태로 부릅니다.

● RESUME : 비상정지상태(혹은 일시정지상태)에서 중지된 현재명령을 다시 실행시켜 정상실행상태로 만드는 명령입니다.

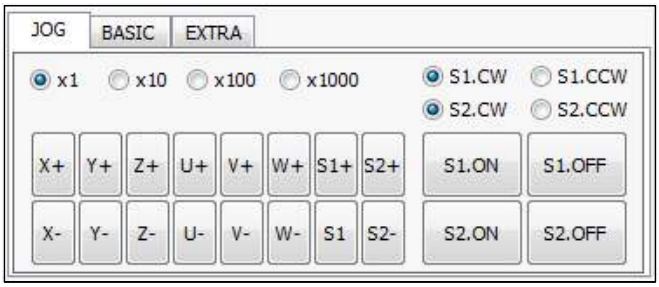
● Q.CLEAR : 모션엔진 내부와 소프트웨어(라이브러리) 상의 모든 명령큐를 클리어합니다.

● CANCEL : 비상정지상태(혹은 일시정지상태)에서 모션엔진내부와 소프트웨어 명령큐를 클리어하고 중지된 현재명령을 취소하면서 정상실행상태로 전환시키는 명령입니다.

● STEP.ON : 큐명령을 하나씩 하나씩 실행하는 스텝실행모드로 만드는 명령입니다. 스텝실행모드에서는 명령큐의 명령이 [RESUME]명령에 의해서 하나씩 실행됩니다. [CANCEL]명령으로 다음 실행될 큐명령을 실행하지 않고 버릴 수 있습니다.

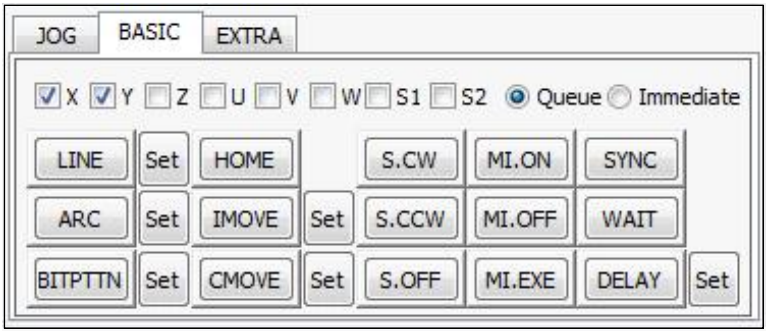
● STEP.OFF : 스텝실행모드를 정상실행모드로 만드는 명령입니다.

6.9.2. 조그동작



조그동작은 개별적인 축들을 해당축의 이동버튼이 눌러질때마다 지정된 펄스단위로 이동시킵니다.
펄스단위는 x1, x10, x100, x1000 배율이 가능합니다. 단, 모션엔진이 지원하는 축의 이동만 가능합니다.

6.9.3. 기본명령동작



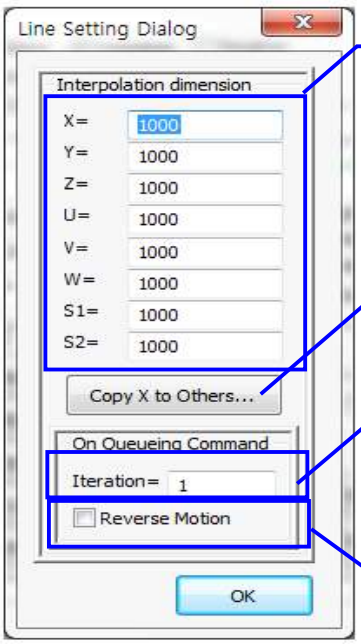
- 기본명령을 실행하기 전에 먼저 각 명령에 적용될 속도성분들이 적절히 설정되어 있는지 확인합니다.
- 기본명령을 실행하기 위해서 실행축을 선택하고

큐명령으로 전송할지 즉시명령으로 전송할지를 지정합니다.

● LINE / Set

: Set버튼을 선택하면 아래와 같은 직선보간의 동작설정화면이 나타납니다.

이들 옵션을 설정한 후 [LINE보간명령]을 클릭하면 동작을 실행합니다.



이동할 축의 상대적거리(distance)를 지정합니다.

X축의 이동량을 다른 축에 똑같이 복사할 때 클릭합니다.

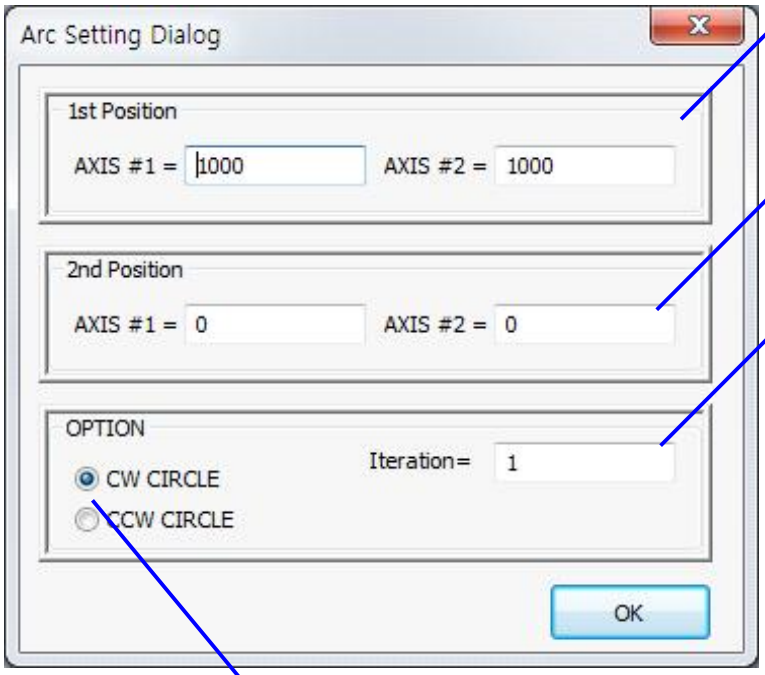
큐명령 전송한다면 명령반복회수(Iteration)을 지정합니다.

큐명령 전송한다면 역 동작 보간명령을 같이 전송할지를 지정합니다.

● ARC / Set

: Set버튼을 선택하면 아래와 같은 원호보간의 동작설정화면이 나타납니다.

이들 옵션을 설정한 후 [ARC보간명령]을 클릭하면 동작을 실행합니다. 원호보간은 반드시 두 개의 축상에서만 가능합니다. 그러므로 제1축(AXIS#1)과 제2축(AXIS#2)로 표현됩니다. 제1축과 제2축은 축의 지정순서에 의해 자동으로 결정됩니다. 예를 들어 X,Z축상의 원호보간일 경우 제1축(AXIS#1) = X축이며, 제2축(AXIS#2) = Y축입니다.



원호보간 제1 경유점을 지정합니다.
현재위치에서의 상대적 위치입니다.

원호보간 제2 경유점(종점)을 지정합니다.
0,0을 지정할 경우, 시점과 종점이 일치하므로 원호가 아닌 완전한 원보간을 지시합니다.

동작반복회수를 지정합니다.

원호가 아닌 완전원일 경우 원 보간의 회전방향을 지정합니다.

●BITPTTN / Set

: Set버튼을 선택하면 아래와 같은 비트패턴보간의 동작설정화면이 나타납니다.

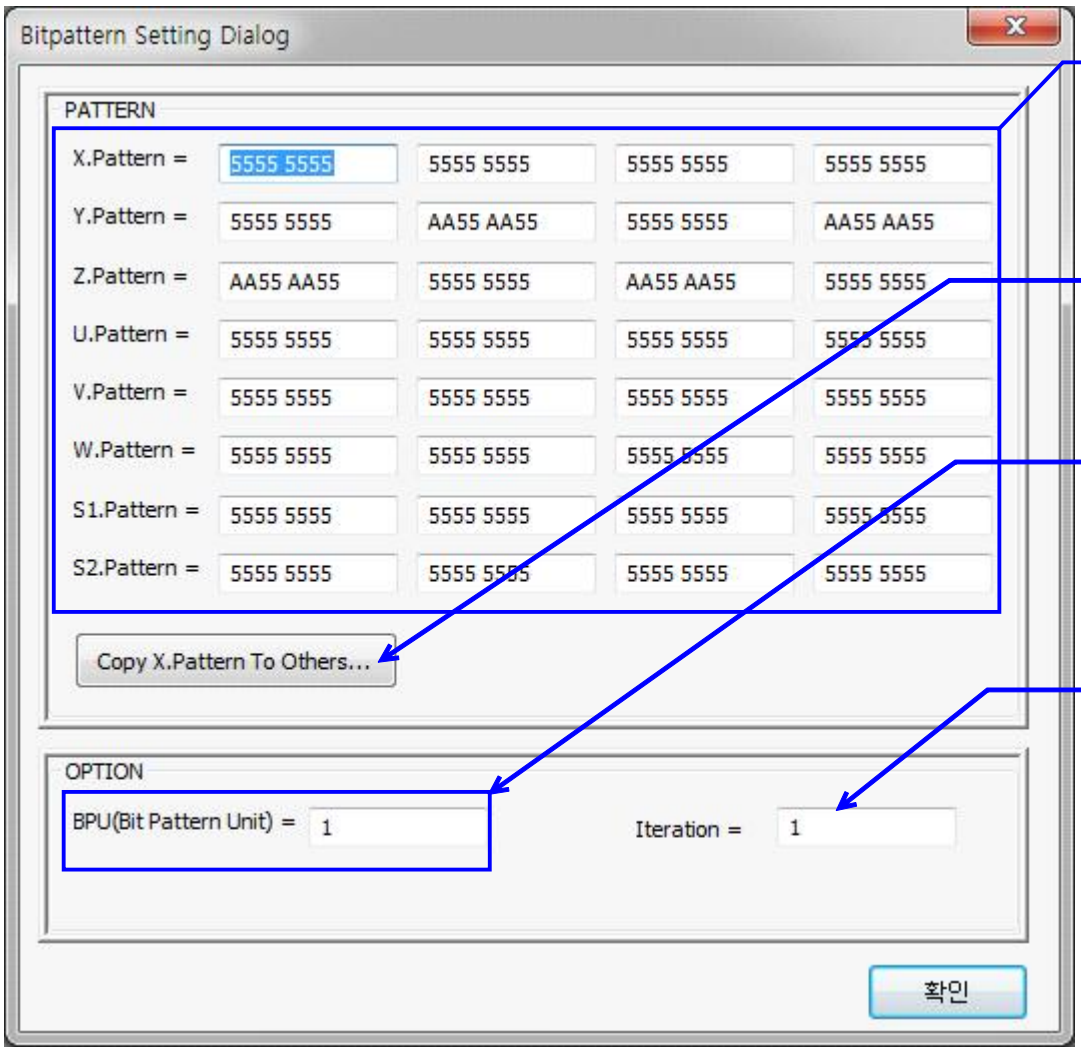
비트패턴보간은 다수의 짧은 직선의 데이터들에 대한 처리를 향상시키기 위한 보간입니다.

각 축마다 32x4=128비트씩 주어지며, 각 축의 한 스텝에 대한 정보는 LSB(최하위 비트)부터 2비트씩 나누어 총 64개의 스텝으로 구성됩니다.

각 스텝의 비트가 이진수 00 이면 펄스발생없음, 01 이면 CW(+방향), 10 이면 CCW(-방향)이 됩니다. 11이면 펄스패턴의 종료를 가리킵니다. 펄스패턴의 종료는 출력하고자 하는 축들의 펄스가 모두 00 (펄스없음)일 때에도 발생합니다.

비트패턴 보간에서 하나의 비트스텝에 대한 펄스의 발생단위를 지정할 수 있습니다. 이 값을 BPU(비트 패턴 유닛)로써 16비트값을 갖습니다. 이 값은 비트패턴에 대한 스케일 조절 역할을 합니다.

비트패턴보간 동작을 실행하기 위해 다음과 같은 설정화면의 항목들을 정확히 기입한 후 [BITPTTN]명령을 클릭합니다.



각 축의 출력비트패턴을 16진수값으로 32비트씩 4개의 필드에 기록합니다.

X축 데이터를 다른 대상 축에 복사합니다.

비트패턴의 단위펄스수를 지정합니다.

동작반복회수를 지정합닌.

예로써 XY축에 대한 +방향의 4펄스단위 계단형태 이동에 대한 비트패턴은

X축 비트패턴값 = 5555 0000,5555 0000,5555 0000,5555 0000

Y축 비트패턴값 = 0000 5555,0000 5555,0000 5555,0000 5555 로 표현될 수 있습니다.

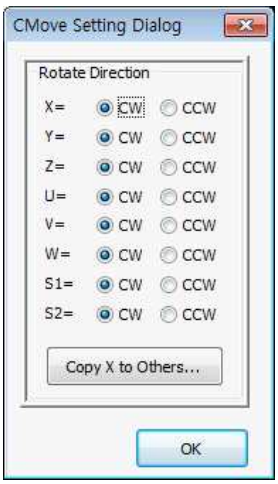
● HOME : X6V3엔진은 기본적으로 1개의 보간동작용 메인모션엔진과 4개의 독립이동용 보조모션엔진, 그리고 1개의 스핀들용 모션엔진으로 구성됩니다. 원점복귀동작을 수행에 관여되는 모션엔진은 메인모션엔진과 보조모션엔진으로써 동시에 최대 5축까지 원점복귀가 가능합니다.
그러나 현재 모니터링화면에서의 HOME명령은 보조모션엔진에 대한 HOME명령만을 사용함으로써 최대 4축까지 동시 원점복귀를 수행할 수 있습니다. 만약 6개의 축을 모두 지정하였다면 선두의 XYZU축이 먼저 원점복귀를 실행하고, 원점복귀가 먼저 종료된 축의 보조모션엔진이 반환되어 자동으로 원점복귀를 수행하지 않은 다음 축(V)에 대한 원점복귀를 수행하게 됩니다.

● IMOVE : 독립이송명령으로써 보조모션엔진을 사용합니다. 다른 축과의 보간동작을 수행하지 않으며, 단일축에 대한 PTP동작만을 수행합니다. X6V3엔진은 최대 4개의 축에 대해서 동시 실행이 가능합니다.
[SET] 버튼을 클리하면 LINE 명령의 설정화면과 동일한 설정화면이 나타납니다. LINE명령과 동일한 방식으로 설정한 후 [IMOVE]실행버튼을 클릭하면 독립이송명령[IMOVE]가 실행됩니다.

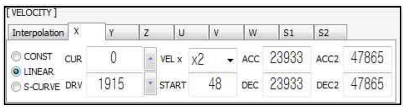
IMOVE 명령의 속도는 개별축 속도항목

들의 설정으로 결정되며, 보간동작의 속도 설정항목과는 무관합니다.

● CMOVE : 독립연속이송명령으로써 보조모션엔진을 사용합니다. 독립연속이송명령은 방향과 속도만 지정하면 해당 축이 계속적으로 이동하는 명령입니다. 스핀들회전명령도 동일한 방식이지만, CMOVE명령과의 차이점은 CMOVE명령의 실행은 다른 보간이나 이동명령처럼 좌표값이 바뀌지만, 스핀들회전명령은 좌표값의 변화가 없습니다.



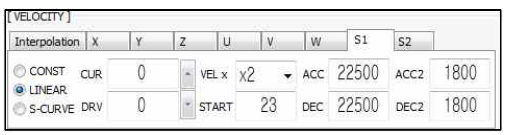
CMOVE 명령을 실행하기 위한 설정화면은 아래와 같으며 단순히 방향만을 설정합니다.
CMOVE 명령의 동작속도는 속도설정화면에서 명령 대상 축에 대한 속도항목들



의 설정으로 결정됩니다.

CMOVE에 의해 이동하는 축을 중지 시키려면 해당 축의 구동속도를 0으로 설정합니다.

● S.CW : 스핀들 CW회전명령
스핀들 CW회전명령은 스핀들회전용 모션엔진을 사용하여 실행됨으로써, 주모션엔진과 보조모션엔진을 사용하지 않습니다. 그러므로 스핀들 축에 대한 회전은 독립적이며, 내부적으로 독립된 하드웨어로 수행됨으로써 모션엔진의 실행속도에 부하를 주지도 않습니다.
X6V3엔진의 스핀들 회전펄스의 출력주파수는 최대 10MHz까지 동작할 수 있지만 일반적으로 1MHz이하에서 사용하는 것이 바람직합니다.

스핀들 회전명령의 속도는 해당 스핀들축에 대한 속도 

설정값으로 결정됩니다.

● S.CCW : 스핀들 CCW회전 명령
스핀들 CCW회전명령 내용과 동일하며, 회전방향만 반대방향입니다.

● S.OFF : 스핀들 회전중지 명령
스핀들 회전중지 방식은 회전중지명령을 통해서도 가능하며, 스핀들 구동속도를 0으로 설정하여 중지시킬 있습니다.

● MI.ON : 다중보간 시작 명령 (Multiple Interpolation)
다중보간은 서로 다른 두 개의 보간명령을 동시에 시작해서 동시에 종료하기 위한 옵션명령입니다.
다중보간을 통해서 원호명령과 직선명령을 동시에 수행하면 헬리컬보간을 수행할 수 있습니다.
다중보간을 실행하는 순서는 아래와 같습니다.
(1) 다중보간 시작명령 MI.ON
(2) 제1 보간명령
(3) 제2 보간명령
(4) 다중보간 실행명령 MI.EXE
(5) STEP(2)~STEP(4) 반복
(6) 다중보간 종료명령 MI.OFF

● MI.OFF : 다중보간 종료 명령 (다중보간 시작명령(MI.ON)을 참조하세요.)
● MI.EXE : 다중보간 실행 명령 (다중보간 시작명령(MI.ON)을 참조하세요.)

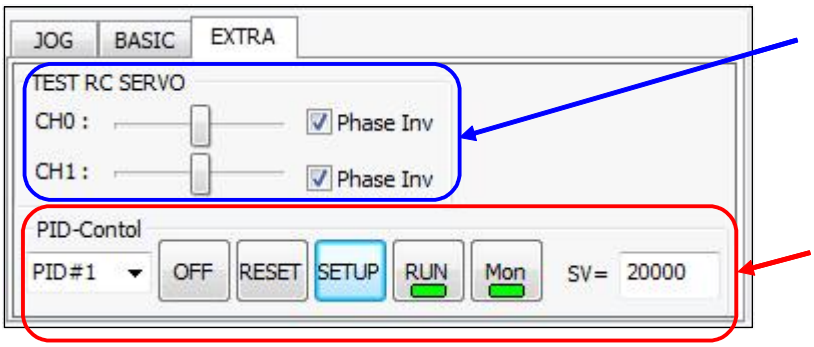
● SYNC : 동기 명령
SYNC명령은 모션엔진의 SYNC신호가 입력될 때까지 대기하는 명령입니다. SYNC신호가 입력되면 큐의 다음명령들을 실행하게 됩니다.

● WAIT : 대기 명령
WAIT명령은 SYNC명령과 유사하게 명령실행을 대기합니다. [RESUME]명령에 의해 WAIT상태에서 벗어날 수 있습니다.

● DELAY : 지연 명령
DELAY 명령은 지정된 시간만큼 휴식하고 다음 명령들을 실행합니다.
DELAY 명령은 큐명령에서만 유효합니다.

6.9.4. 추가명령동작

추가명령동작은 모션엔진에 내장된 서브컨트롤러의 기능들을 시운전할 수 있는 곳입니다. 위칸모션엔진은 PWM발생기 2채널과 PID제어기 4채널을 내장하고 있습니다. 특히 PID제어기의 동작은 여러 가지 입력과 출력으로 연결되어 온도제어 등의 유용한 부가장치로 확장 가능합니다.



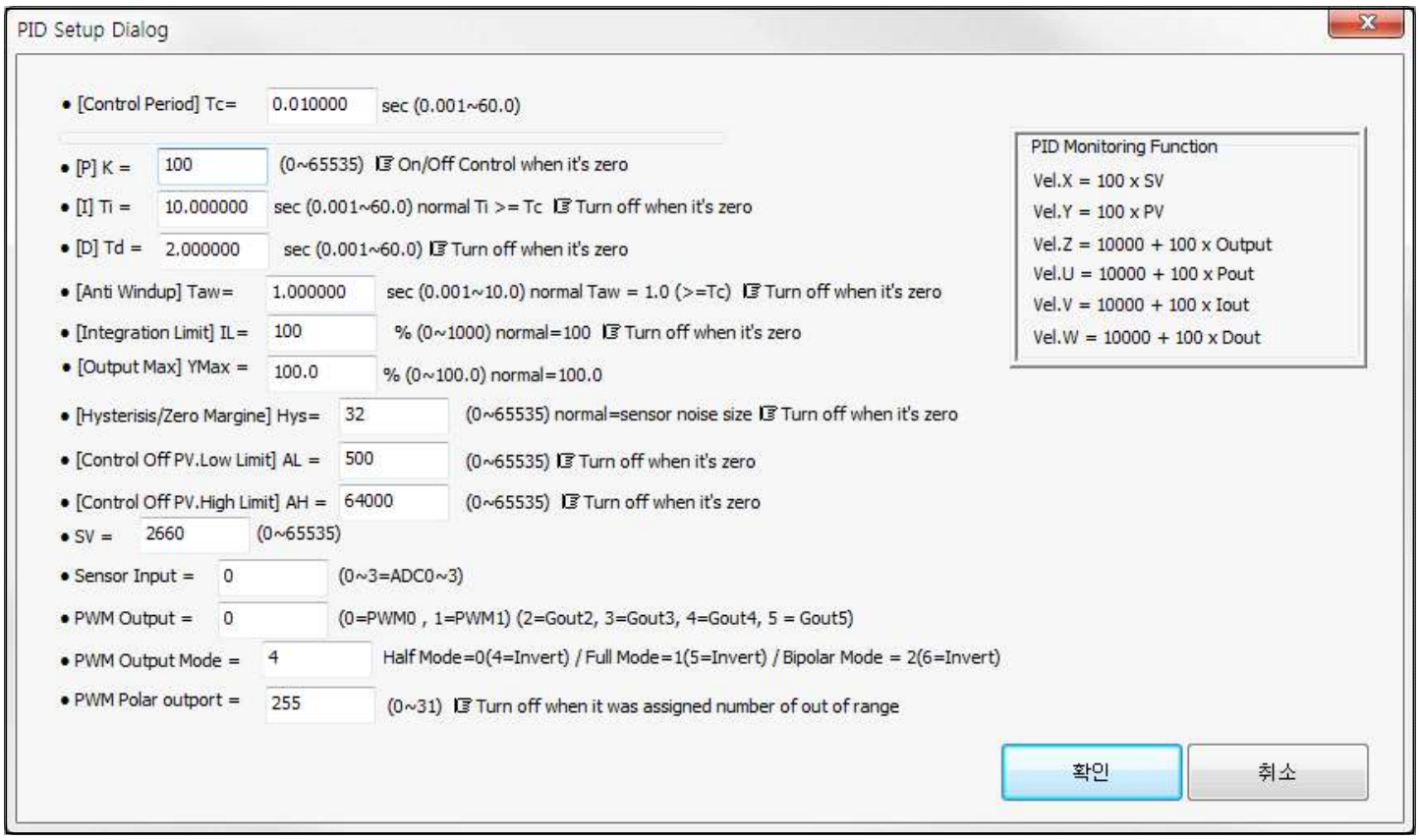
PWM 기능을 이용한 RC서보모터 테스트기능

내장 PID제어기 테스트 기능

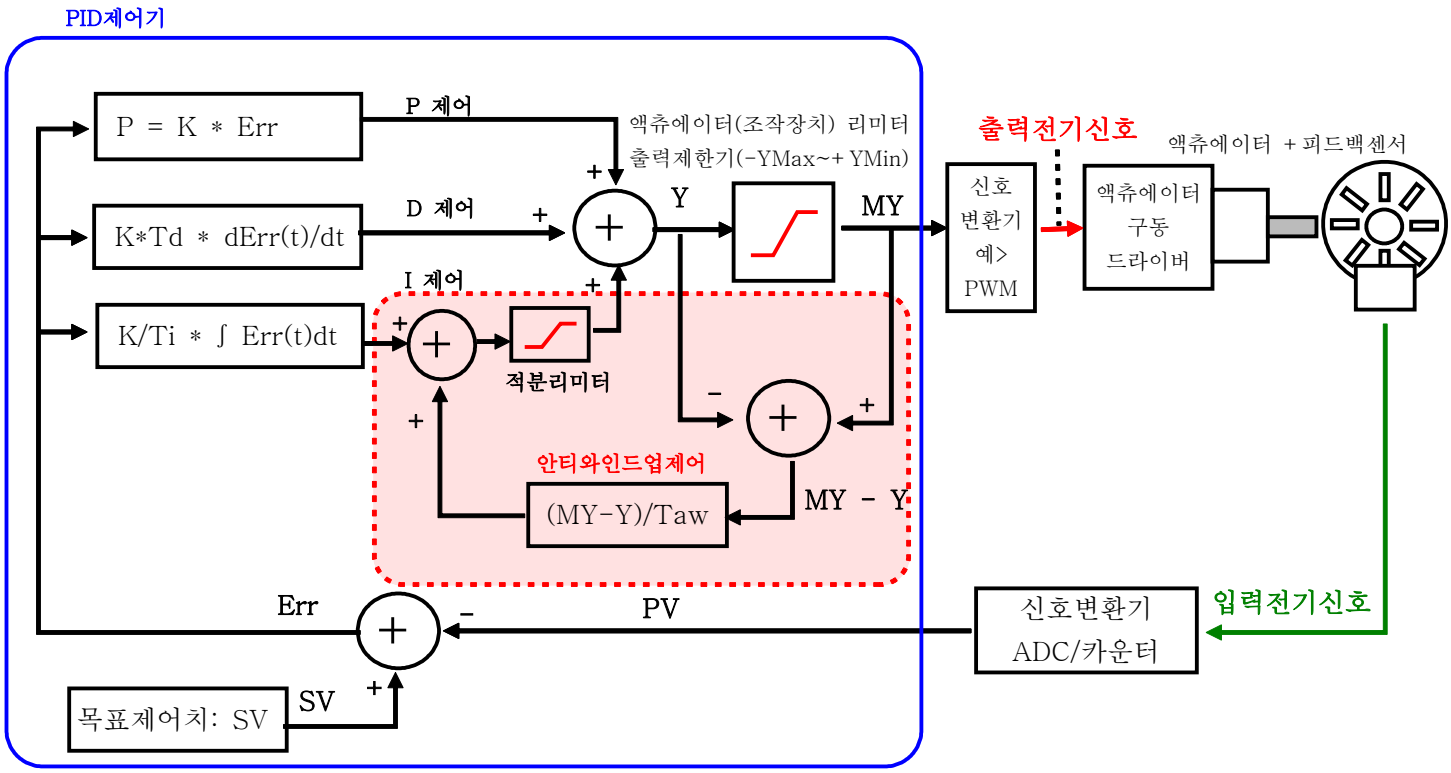
● X6V3엔진은 2개의 PWM채널을 갖습니다. PWM출력이 이용하면 간단히 RC SERVO모터의 회전을 테스트해 볼 수 있습니다. X6V3엔진의 PWM은 내부적인 타이머에 의한 펌웨어적 파형이이므로, 하드웨어적으로 구현된 PWM파형만큼의 정밀성과 안정성을 갖출 수는 없지만, 해상도가 높지 않은 회전에서 이용될 수 있습니다.

● X6V3 엔진은 4개의 PID제어기를 내장하고 있습니다.
PID제어기 옵션에 따라, ON/OFF제어, P제어, PI제어, PD제어 등도 가능합니다.

PID제어기를 정확하게 동작하기 위해서는 SETUP 버튼을 눌러 아래의 파라메터들을 설정해야 합니다.
이들 파라메터들에 대한 내용은 PID제어에 대한 이해를 우선적으로 필요로 합니다.



<X6V3 모션엔진 PID제어기 기본 동작 다이어그램>



먼저 각 파라메터들의 기능들을 요약해 봅니다.

- [Control Period] Tc : PID제어주기로써 지정된 시간단위로 연산이 이루어지며 출력도 이 주기로 발생됩니다.
출력이 PWM출력으로 지정되었다면 PWM의 동작주기도 Tc 와 동일합니다.
이 값은 제어시스템에서 가장 먼저 선택되어야하는 파라메터로써 제어시스템이 동특성이 얼마나 빨리 응답하는가에 대한 기본적인 내용을 포함합니다. 디지털제어에서는 기본적으로 샘플링시간과 연관됩니다.
만약 제어속도가 빠르지 않는 온도제어시스템이다면 Tc는 대부분 0.1초 이상일 것입니다.

- [P] K : 비례대 (0~65535)
비례대값은 P제어 출력이 100%가 출력되는 센서값 PV와 목표제어값 SV간의 오차값(Err)이 됩니다.
센서와 목표치제어 값은 0~65535 로써 16비트 부호없는 정수값을 기준으로 합니다.
그러므로 K = 1000 이면 오차값Err이 1000될 때 비례제어출력 값이 100%에 도달합니다.
K값이 작을수록 비례이득이 커집니다. (다이어그램의 K값과 다르며 비율 변환되어 사용됩니다.)
※ K값이 0 이면 PID제어는 ON/OFF제어로 전환됩니다.

-[I] Ti : 적분시간 (Reset Time) (0.001~60.0초)
비례제어가 수렴했을 때 발생하는 Static Err(정적오차)값이 리셋되도록 적분제어를 실행하는 이득값과 관련됩니다. 제어주기 Tc보다는 커야 합니다(작으면 제어는 진동합니다). Ti값이 작을수록 적분제어의 이득값이 커져 보다 빠르게 정적오차를 제거하지만, 시스템이 진동할 수 있습니다.
※ Ti값이 0 이면 적분제어는 동작하지 않습니다.

-[D] Td : 미분시간 (0.001~60.0초)
외부의 외란적상황(급작스러운 외부변화:노이즈, 충격, 챔버문열림/닫힘등)에 적응하는 능력을 가진 미분제어를 실행하는 이득값과 관련됩니다. Td값은 클수록 미분제어영향이 커집니다.
※ Td값이 0 이면 미분제어는 동작하지 않습니다.

-[Anti Windup] Taw : 안티와인드업 (0.001~60초)
제어출력값 Y가 액추에이터의 조작값MY 범위(-100%~+100%)를 초과할 때 적분동작을 제한하는 기능을 합니다.
적분동작을 제한하지 않으면 적분량이 지나치게 커져서 목표값에 도달하더라도 적분오차가 사라지기 전까지 출력상태는 거의 변화하지 않게됩니다. 안티와인드업동작은 이러한 과적분에 의한 오버슈트(Overshoot)나 언더슈트(Undershoot)현상을 줄이는 기능을 합니다.
※ Taw값이 0 이면 안티와인드업은 동작하지 않습니다.

-[Integration Limit] IL : 적분리미터 (0%~1000%)
제어시스템에 적합하지 않은 제어파라메터의 설정 혹은 외부외란에 의한 급작스런 상황변경등에서 오차Err가 커질 경우 적분량이 급수적으로 증가할 수 있습니다. 이때 적분리미터에 의해 적분제어량을 제한함으로써 예측지 못한 출력결과를 막을 수 있습니다.
적분리미터값은 적분제어에 의한 조작출력Y의 최대값100%를 기준으로 얼마나 허용할지를 결정합니다. 시스템에 따라 적분제어의 영향력은 달라질 수 있으나 낮을수록 안전합니다.
※ IL값이 0 이면 적분리미터는 동작하지 않습니다.

-[Output Max] YMax : 최대 출력값 (0.0~100.0%)
제어시스템은 내부적으로 -100.0% ~ +100.0%를 출력합니다. 이 출력값은 YMax값에 의해 -YMax ~ +YMax 로 조절됩니다. 히터제어의 경우 최대 출력값의 제한은 최대사용전력을 제한하는 효과를 가집니다.
※ YMax 값이 0이면 출력이 차단(0)됩니다.
※ 단, 출력방식(PWM Output Mode) 이 1일 때 YMax가 100.0이 아니면 0%의 출력에 도달할 수 없으므로 유의하셔야 합니다.

-[Hysterisis / Zero Margine] Hys : 히스테리시스(불감대역)
센서의 노이즈나 출력동작의 미세한 진동등에 대해서 PID제어기가 동작변화를 크게 일으키지 않는 불감대영역을 결정하는 역할 합니다.
주로 센서의 기준입력값인 0~65535값 내에서 센서의 불안정한 비트들의 최대값으로 설정할 수 있습니다.
예를 들면 X6V3은 내부적으로 10비트 해상도를 갖는 ADC가 내장되었지만 출력은 16비트로 확장되어 있습니다.
그러므로 6비트값은 불안정한 값을 가질 수 있으므로 Hys = 2^6 = 32값으로 지정할 수 있습니다.
※ Hys값이 0 이면 히스테리시스는 동작하지 않습니다.

- [Control Of PV.Low Limit] AL : 로레벨 알람 (0~65535)
센서값(PV) 이 AL값 이하일 때 출력을 0%로 차단하는 역할을 합니다.
※ AL값이 0 이면 로레벨알람은 동작하지 않습니다.

- [Control Of PV.High Limit] AH : 하이레벨 알람 (0~65535)
센서값(PV) 이 AH값 이상일 때 출력을 0%로 차단하는 역할을 합니다.
※ AH값이 0 이면 하이레벨알람은 동작하지 않습니다.

- SV :제어목표값 (0~65535)
제어목표값으로써 입력센서(PV)가 PID제어로써 도달하고자 하는 값을 설정합니다.

- ADC Channel : PV값과 연결되는 센서입력장치의 종류를 선택합니다.
모션엔진마다 각 값에 대한 연결장치의 종류가 다를 수 있습니다.
X6V3엔진은 아래와 같습니다.
① 0,1,2,3 : (기본적인 범위) 각각 ADC 채널0,1,2,3과 연결됩니다.
② 24, 25 : 각각 GIO 엔코더카운터 채널 0, 1과 연결됩니다.

- PWM Channel : 액츄에이터출력 MY값과 연결되는 변환장치의 종류를 선택합니다.
모션엔진마다 각 값에 대한 연결장치의 종류가 다를 수 있습니다.
X6V3엔진은 아래와 같습니다.
① 0, 1 : 각각 PWM채널 0, 1과 연결됩니다.
② 2 : GIO 출력포트 bit2를 통해서 PWM신호로 연결됩니다.
3 : GIO 출력포트 bit3를 통해서 PWM신호로 연결됩니다.
4 : GIO 출력포트 bit4를 통해서 PWM신호로 연결됩니다.
5 : GIO 출력포트 bit5를 통해서 PWM신호로 연결됩니다.

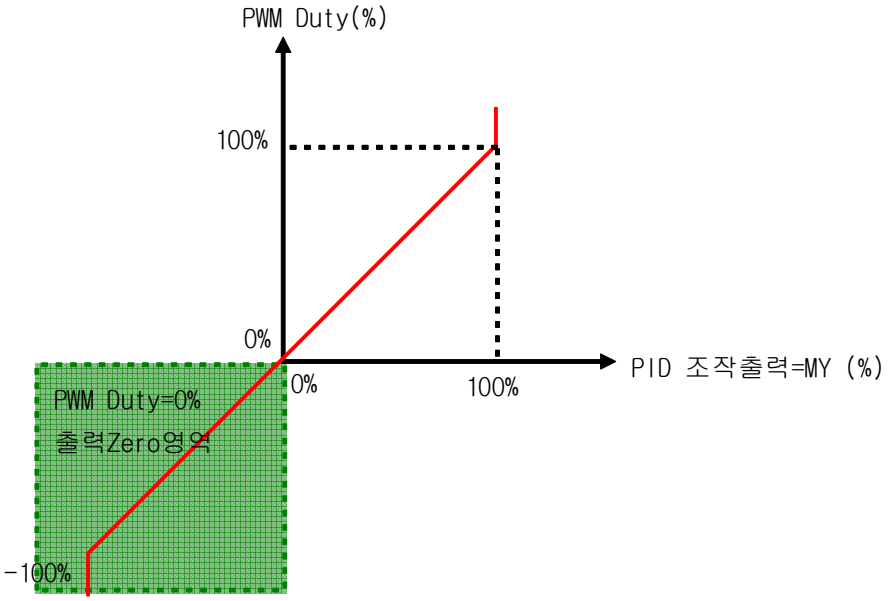
- PWM Output Mode : PWM 출력모드

PID제어기는 내부적으로 출력MY값이 -100.00%~+100.00%값을 출력합니다.

이 값을 PWM 연결신호로 선택할 때 아래의 3가지 방식 중 하나를 선택할 수 있습니다.

<0 방식>

PID제어기 출력중 0~100%만을 PWM의 출력(Duty)에 사용합니다.

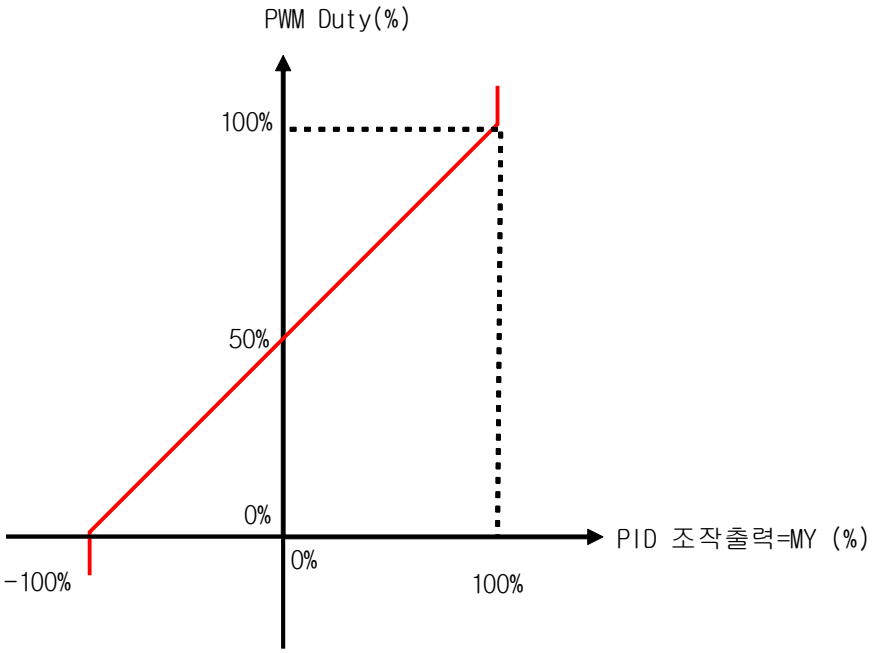


<4 방식>

0방식의 출력반전 방식입니다.

<1 방식>

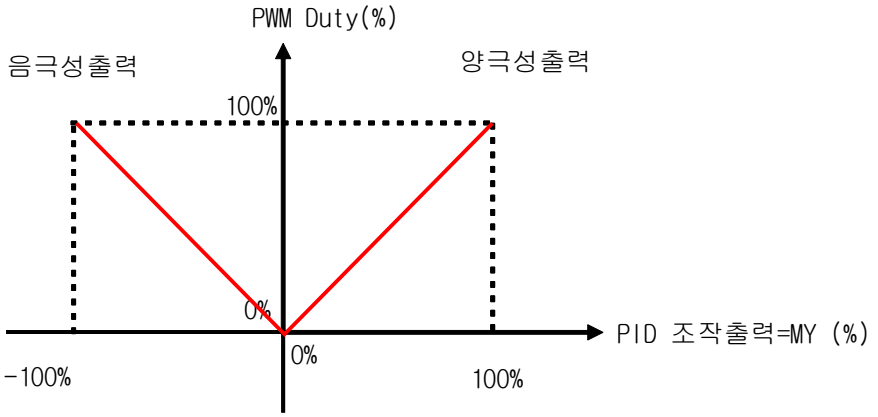
PID제어기 출력 -100~100%를 PWM의 출력(Duty) 0~100%로 스케일링하여 사용합니다.



<5 방식>

1방식의 출력반전 방식입니다.

<2 방식>
별도의 극성출력포트를 설정하고 PID출력(-100 ~ +100%)의 절대값을 취해서 PWM 출력(Duty)값으로 사용합니다.
별도의 극성출력포트설정은 다음에 나오는 PID 파라메터인 "PWM Polar outport "설정을 참조하세요.



<6 방식>
2방식의 출력반전 방식입니다.

- PWM Polar outport : PWM 극성 출력포트 지정
PWM출력모드가 2일 때 극성출력을 필요로 합니다. 이때 극성출력을 GIO출력포트 중 임의로 지정할 수 있습니다.
음극성 신호(PID제어기 출력값이 -값)일 때 출력이 1이 됩니다. 양극성 신호(PID제어기 출력값이 +값)일 때 출력은 0이 됩니다.
X6V3엔진은 GIO출력포트가 8개 존재하므로 0~7 중 하나를 선택합니다.
※ 출력포트지정값이 0~31의 범위가 아닐 때 극성출력은 동작하지 않습니다.

● PID제어기 테스트버튼

- PID 제어기 선택

총 4개중 하나를 선택하여 동작시킵니다. 선택시마다 RUN과 MON(모니터링) 램프에 의해 현재선택된 PID제어기가 작동중인지, 모니터링동작 중인지 확인할 수 있습니다.

- PID 제어기 OFF :

버튼을 누르면 PID제어기의 RUN램프가 꺼지면서 제어기동작이 중지됩니다. 출력은 0이 되며 극성출력도 0 이됩니다.

- PID 제어기 RESET :

버튼을 누르면 PID제어기의 동작이 리셋됩니다. 출력이 0이 되며, 내부파라메터는 변화되지 않습니다.

- PID 제어기 SETUP :

상기 파라메터 설정편을 참조하세요.

- PID 제어기 RUN :

버튼을 누르면 설정된 파라메터값으로 PID제어동작을 시작합니다. 시작과 동시에 램프가 녹색으로 점등합니다.

- PID 제어기 MONITOR: PID제어기가 리셋상태이거나 동작상태에 있다면

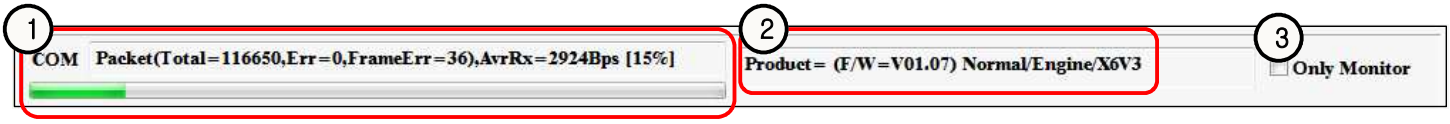
버튼을 눌러 내부동작값을 모니터링 할 수 있습니다. 모니터링 방법은 PID제어기의 내부값이 모션엔진의 축별 현재속도값에 나타납니다.
그러므로 Time Graph의 X,Y,Z,U,V,W 의 항을 체크[√]하여 필요로 하는 정보를 시간그래프로 확인할 수 있습니다.
각 축별 현재속도값과 PID제어 내부연산결과는 아래와 같이 맵핑됩니다.

개별축 속도	PID제어기 내부연산결과값	원래값 범위	정수화 연산방법
X	내부 SV 의 정수화된 값	0.0~100.0	100 x SV
Y	내부 PV 의 정수화되 값	0.0~100.0	100 x PV
Z	내부 MY(조작출력)의 정수화된 값	-100.0~100.0	10000 + 100 x MY
U	내부 P제어출력의 정수화된 값	-100.0~100.0	10000 + 100 x P
V	내부 I제어출력의 정수화된 값	-100.0~100.0	10000 + 100 x I
W	내부 D제어출력의 정수화된 값	-100.0~100.0	10000 + 100 x D

모니터링 기능은 모든 PID제어기 중 하나만 선택되어야 데이터값이 교차되지 않습니다. 또한 현재 모션엔진의 축개별 모션동작이 수행되지 않는 상태이어야 안전합니다.

※ PID제어 모니터링 기능은 현재 PID제어기의 동작 경향을 대체적인 그래프 표현을 위한 목적입니다. 그러므로 모션엔진의 다른 동작과 함께 사용하지 않는 것이 바람직하며, PID제어의 동작상태를 확인했다면 모니터링을 반드시 OFF하여 모션엔진의 내부 속도관련 메모리를 정상적인 사용방식으로 복원시킵니다.

6.10. 통신상태 및 엔진버전 표시



(1) 현재 모션마스터 프로그램과 모션보드간의 통신상태를 표시합니다.
Packet(Total=xxxxx, Err=yyyy, FramErr = zzzz), AvrRx = kkkkBps[jj %] 형식으로 표시합니다.
Total = xxxx는 전체 수신패킷에 대한 카운터값입니다.
Err = yyyy 는 전체 수신패킷에 대한 패킷오류카운터 값입니다. 이 값이 1 이상이면 시스템의 노이즈를 확인해야 합니다.
FrameErr = 패킷의 동기화 프레임오류를 가리킵니다. 주로 PC와 모션이 통신을 시작할 때 상호 통신동기를 찾는 과정에서 FrameErr오류가 발생될 수 있습니다. 또한 모션보드와 통신속도가 매칭되지 않을 경우에도 프레임오류가 발생합니다.
AvrRx = kkk BPS[jj %] 는 현재통신속도에서의 데이터수신율을 표시합니다. kkkBps(Byte per sec)는 초당 평균수신 바이트 수이며, jj%는 현재통신대역에서 수신데이터의 점유율을 표시합니다. 이 점유율은 진행바 ( ) 로도 표시됩니다.

(2) 모션컨트롤러에 장착된 모션엔진의 종류와 엔진버전을 표시합니다.
Product=(F/W=V01.07) Normal/Engine/X6V3
모션엔진 펌웨어버전은 V01.07 이며 X6V3엔진이 내장되었음을 알려줍니다.

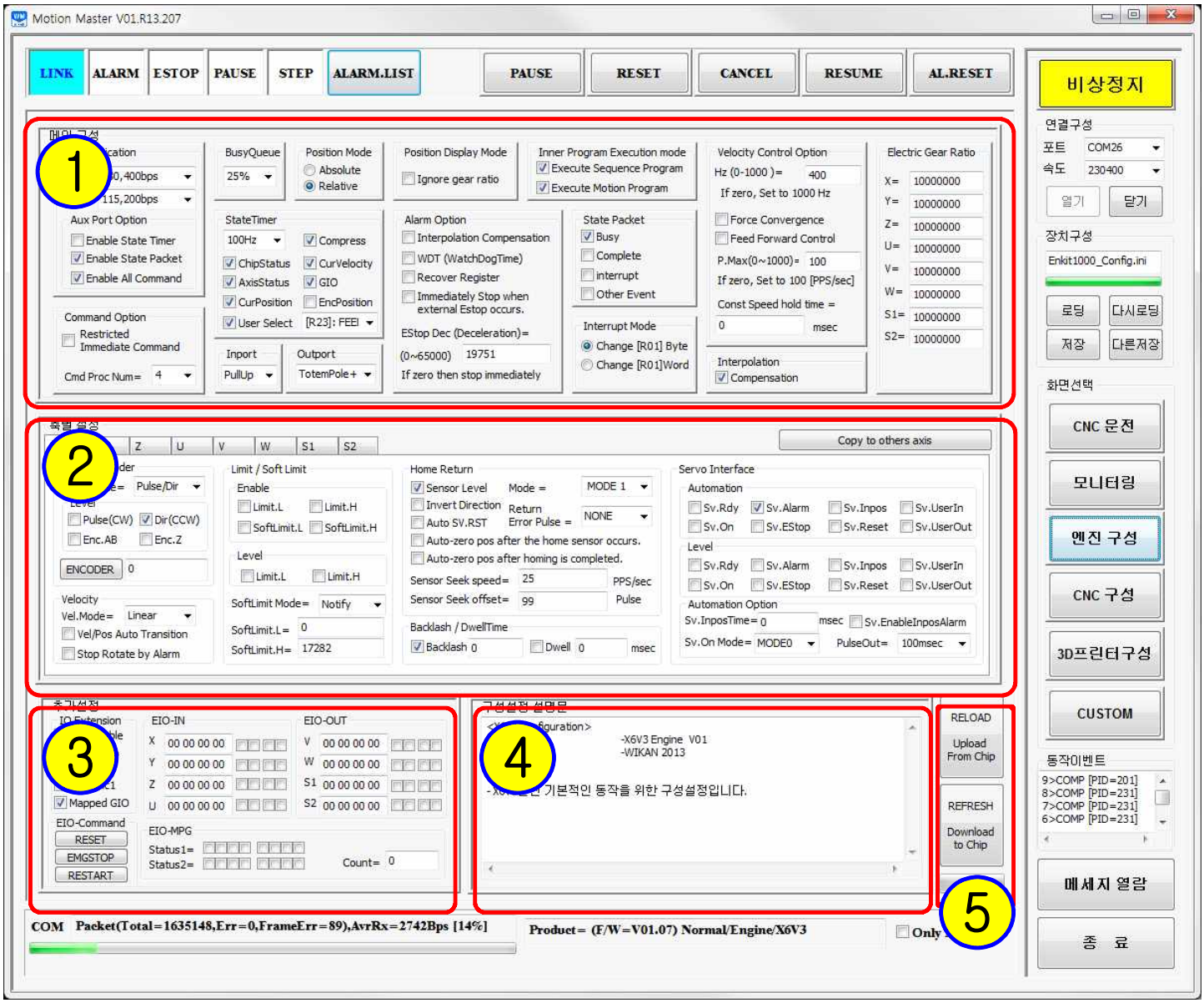
(3) 모션마스터프로그램의 동작에서 엔진구성과 모니터링부분만을 사용할 때 설정하는 옵션입니다.
구성파일의 저장 역시 CNC 영역이 제외됩니다.

7. 엔진구성 화면

위칸모션컨트롤러는 X6V3계열의 모션엔진칩을 포함하고 있습니다. 실제 모든 모션동작은 모션엔진칩에 의해서 실행됩니다. 엔진구성화면은 이러한 모션엔진의 기능을 설정하는 화면입니다. 엔진구성화면은 대부분 전기적 접속 및 모션동작을 위한 보다 하위레벨의 구성설정과 관련됩니다. 주요 설정그룹을 살펴보면 아래와 같습니다.

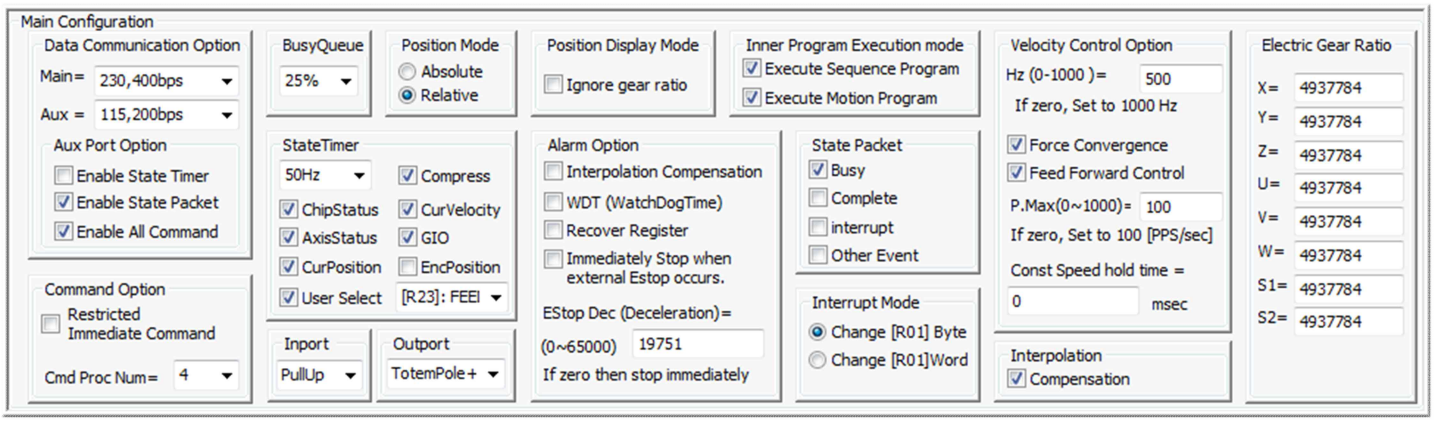
(1) 모션엔진 주요 구성설정
(2) 모션엔진 축별 구성설정
(3) 모션엔진 GIO/EIO 입출력 구성설정
(4) 모션엔진 구성설정 설명문
(5) 업로드, 다운로드 기능

각각의 기능들에 대한 보다 자세한 내용은 모션엔진 설명서를 참조해 주세요.



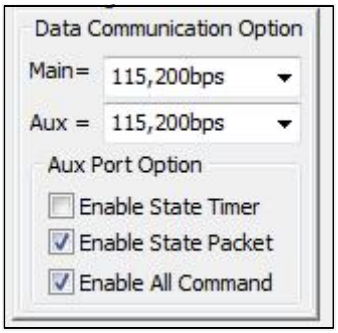
7.1. 모션엔진 주요 구성설정

모션엔진의 전체동작에 관여되는 주요설정들을 관리하는 영역입니다.



각 영역별로 아래와 같은 구성설정 그룹으로 나눌 수 있습니다.

7.1.1. 통신



모션엔진은 기본적으로 메인연결포트(Main)를 내장하고 있으며, 제품에 따라 보조연결포트(Aux)가 내장될 수도 있습니다. 통신속도는 외부선택스위치에 의해 강제설정 될 수 있으므로 각 제품들의 통신설정방법을 참조하세요.

X6V3 엔진은 Main포트 1개를 내장하며, Aux포트는 지원하지 않습니다. Main포트의 통신속도는 스위치에 강제 설정되므로 X6V3엔진이 내장된 보드와의 연결시 속도설정을 맞게 지정하여야 합니다.

Main 포트는 항상 열려있어 통신이 가능하지만, Aux포트는 기본적인 설정을 통해서만 통신기능들이 가능 합니다. Aux포트가 지원된다면 아래의 기능을 사용할 수 있습니다.

●Enable State Timer : 상태타이머가 발생되었을 때 지정된 정보들을 Aux포트로도 전송하는 것을 허용하는 옵션

●Enable State Packet : Busy상태, Complete상태, Interrupt상태가 발생되었을 때 각 상태패킷을 전송하는 것을 허용하는 옵션

●Enable All Command : 모든 명령을 수신하여 실행할 수 있도록 허용하는 옵션입니다. 이 옵션이 허용되지 않으면 즉시명령방식의 시스템명령(RESET, RESUME, CANCEL, ALRESET 명령 등)만이 허용됩니다.

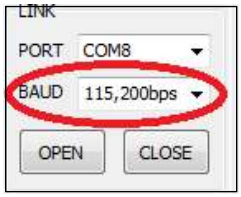
※ 통신속도 바꾸는 과정

일반적으로 모션엔진의 통신속도는 초기상태의 통신속도를 사용하는 것이 가장 안정됩니다. 그러나 빠른 입출력상태처리 등의 특별한 성능이 요구된다면 통신속도를 전환하여 대응할 수 있습니다.

제품마다 통신속도를 바꾸는 방식은 크게 하드웨어 스위치에 의한 통신설정변경방식과 모션엔진의 통신설정 파라메터를 통한 설정변경방식이 있습니다. 제품에 따라 둘 중 하나의 방식을 택하며 현재 X6V3엔진은 하드웨어 스위치에 의한 통신속도변경이므로 간단히 모션보드에서 스위치로 설정된 통신속도에 맞추어 포트를 열면 됩니다.

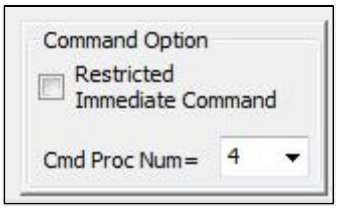
여기서는 모션엔진 내부통신속도 파라메터 변경을 통해서 통신속도를 바꾸는 모션보드를 대상으로 설명합니다.

먼저, 모션보드의 Main포트와 PC가 연결되었고, 115200bps 연결속도를 230,400bps로 변경하는 것을 가정합니다.
STEP.1 모션엔진 구성설정의 Main=115200bps를 Main=230400bps로 변경합니다.

STEP.2 선택화면의 통신속도설정 

을 230,400 bsp로 동일하게 설정합니다.

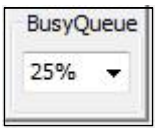
7.1.2. 명령



●Restricted Immediate Command 옵션은 즉시명령실행으로는 축을 움직이는 모션동작명령이 실행되지 않도록 제한을 가하는 옵션입니다. 오직 큐명령으로만 모션동작이 가능토록하는 안전옵션입니다.

● Cmd Proc Num = 4 : 메인제어루프가 1회실행될 모션명령이 실행될 수 있는 최대수를 지정합니다. 1~8내외가 적절합니다.

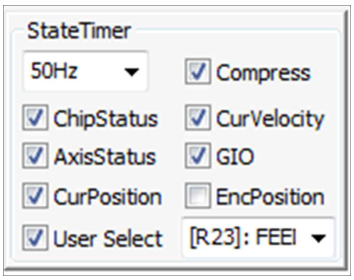
7.1.3. BusyQueue



● Busy상태의 기준이 되는 내부수신메모리의 경계를 설정합니다.

7.1.4. 상태타이머

상태타이머는 모션엔진이 상위제어기(PC)에게 자신의 다양한 정보를 주기적으로 전달하기 위해 사용됩니다.



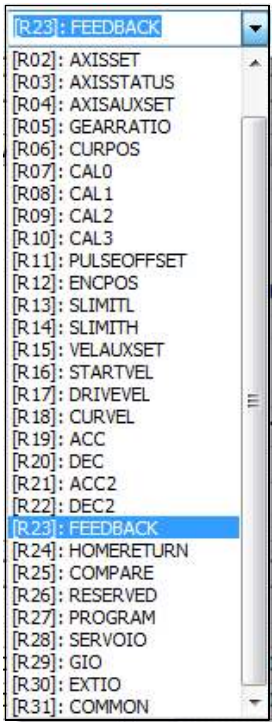
● Hz : 25Hz/50Hz/100Hz/200Hz/250Hz/500Hz/1kHz 중 하나를 선택할 수 있습니다. Hz를 높이기 위해서는 적절한 통신속도와 상태타이머 전송정보들을 선택해야 합니다.

● Compress : 상태타이머가 정보를 보낼 때 압축된 방식으로 보낼지에 대한 선택옵션입니다. 압축방식은 여러 정보들을 하나의 헤더를 갖는 패킷으로 전송하며, 이전 전송정보와 비교하여 변화가 생긴 정보만을 전송하는 방식을 사용합니다. 일반적으로 Compress를 선택하는 것이 통신부하에 유리합니다.

그 외 상태타이머의 전송정보들의 종류를 선택합니다.

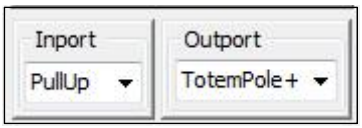
● ChipStatus : 모션엔진의 주요상태에 관련된 정보를 상태타이머를 통해 전송합니다.
● AxisStatus : 모션엔진의 각 축별동작상태에 관련된 정보를 상태타이머를 통해 전송합니다.
● CurPosition: 모션엔진의 현재 위치정보를 상태타이머를 통해 전송합니다.
● EncPosition: 모션엔진의 엔코도 위치정보를 상태타이머를 통해 전송합니다.
● CurVelocity : 모션엔진의 현재 속도관련 정보를 상태타이머를 통해 전송합니다.
● GIO : 모션엔진의 현재 GIO 입출력신호정보를 상태타이머를 통해 전송합니다.

● User Select:
모션엔진의 정보를 담고 있는 임의의 레지스터를 선택합니다.
“[R 레지스터번호] : 레지스터명” 으로 표기되었습니다.



레지스터		정보설명	레지스터		정보설명
번호	이름		번호	이름	
R00	CHIPSET	모션엔진 주요 구성설정	R16	STARTVEL	시작속도(초속도) 정보설정
R01	CHIPSTATUS	모션엔진 주요 동작,신호상태	R17	DRIVEVEL	목표속도(구동속도) 정보설정
R02	AXISSET	모션엔진 축 구성설정	R18	CURVEL	현재속도 정보
R03	AXISSTATUS	모션엔진 축 동작,신호상태	R19	ACC	가속도 설정
R04	AXISAUXSET	모션엔진 축 부가설정	R20	DEC	감속도 설정
R05	GEARRATIO	전자기어비 정보	R21	ACC2	2차 가속도 설정
R06	CURPOS	현재위치 카운터정보	R22	DEC2	2차 감속도 설정
R07	CAL0	명령파라메터 0	R23	FEEDBACK	엔코더z카운터 및 서보피드백정보
R08	CAL1	명령파라메터 1	R24	HOMERETURN	원점복귀 동작 구성설정
R09	CAL2	명령파라메터 2	R25	COMARE	비교기 동작 구성설정
R10	CAL3	명령파라메터 3	R26	AUTOCAL	자동연산기능 구성설정
R11	PULSEOFFSET	펄스동작 설정정보	R27	PROGRAM	현재 사용하지 않음.
R12	ENCPOS	현재엔코더 카운터정보	R28	SERVO	서보입출력 구성설정
R13	SLIMITL	소프트리미트 하한설정값	R29	GIO	범용입출력 정보
R14	SLIMITH	소프트리미트 상한설정값	R30	EIO	확장입출력 정보
R15	VELAUXSET	속도 부가구성설정 정보	R31	CUSTOM	PID 와 같은 부가 컨틀로장치 정보

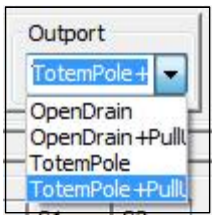
7.1.5. 입출력포트 모드



모션엔진의 입력포트와 출력포트 회로특성 모드를 설정합니다.
입력포트의 경우 내부 Pull-Up저항 연결상태를 설정할 수 있으며 출력포트의 경우 open-drain출력과 totempole출력 및 내부 Pull-Up저항 연결유무를 선택할 수 있습니다. 아래는 입출력 각각의 선택항목들을 보여줍니다.

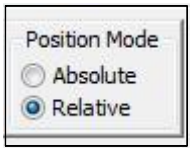
<입력회로 선택항목>

<출력회로 선택항목>



7.1.6. 위치모드

보간명령의 위치변위값을 절대좌표값과 상대좌표값으로 선택합니다.
모션엔진의 직선, 원호보간의 동작방식이 기본적으로 상대좌표값에 기초함으로 위치모드는 일반적으로 상대좌표값을 사용합니다. 만약 절대좌표값을 이용하여 보간동작을 수행할 때에는 이전 모션명령이 완전이 완료된 상태에서 다음명령을 실행하는 방식일 때만 올바른 움직임을 얻을 수 있습니다.



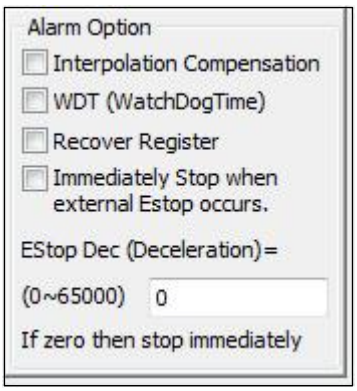
- Abosulte : 절대위치방식
- Relative : 상대위치방식

7.1.7. 위치표시

현재위치의 표현방법이 전자기어비율을 무시한 적용방식인지를 선택합니다. 일반적으로 전자기어비를 적용한 위치표시를 선택합니다. 이 옵션은 전자기어비동작을 시뮬레이션으로 확인할 때 사용합니다.

7.1.8. 알람옵션

알람을 발생시킬 원인과 알람발생시의 동작방식 등을 선택합니다.



● Interpolation Compensation : 보간 보상동작이 특정 펄스수를 초과할 때 알람을 발생을 지시합니다.(원호보간이 종점오차 2펄스 초과시 알람발생)

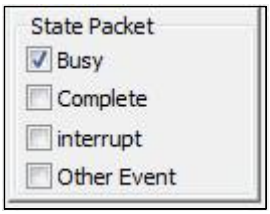
● WDT : 메인제어루프의 실행능력이 저하될 때 알람 발생을 지시합니다.

● Recover Register : 알람발생시 모션엔진은 비상정지상태가 됩니다. 모션엔진은 비상정지상태가 되기 직전의 위치정보를 보관해 둡니다. 이 옵션이 활성화 되면, 비상정지상태에서 축을 움직여 좌표를 변경한 후에 비상정지상태를 해제하여 되돌아 가면 처음 저장된 위치정보로 되돌아가서 중지된 명령을 이어서 실행하게 됩니다. 그러므로 비상정지상태에서의 축 움직임에 대한 효과를 없앨 수 있습니다.

● Immediately stop when external estop occurs : 모션엔진의 estop신호가 발생될 때 즉시정지동작을 수행하도록 지시합니다. 설정하지 않는면 감속정지동작을 수행합니다. 감속정지동작은 현재 설정된 가감속 방식으로 정지합니다.

● EStop Dec(Deceleration) = xxx : 이 값이 0 이 아닌 값이 지정되면 비상정지에 의한 즉지정지방식에서 펄스를 즉각 중단하지 않고, 이 값을 감속계수로 하는 1차감속방식으로 정지합니다.

7.1.9. 상태패킷 옵션



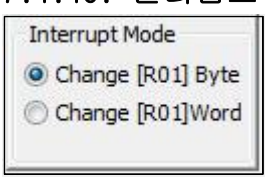
● Busy : 모션엔진이 Busy상태일 때 Busy상태패킷을 전송합니다.

● Complete : 현재 모션동작명령에 대한 동작이 완료될 때 Complete상태패킷을 전송합니다.

● Interrupt : 모션엔진의 R01레지스터(칩상태레지스터)의 내용이 변화할 때 Interrupt상태패킷을 전송합니다. R01 레지스터의 내용은 “모션엔진 모니터링 화면”의 [STATUS] 항목란의 표시내용을 담고 있습니다. 이 내용의 값이 변화될 때마다 Interrup상태패킷을 전송하도록 허용합니다.

● other Event : 모션엔진 내의 특정이벤트상태가 발생했을 때 event상태패킷을 전송하도록 허용합니다.

7.1.10. 인터럽트 발생옵션

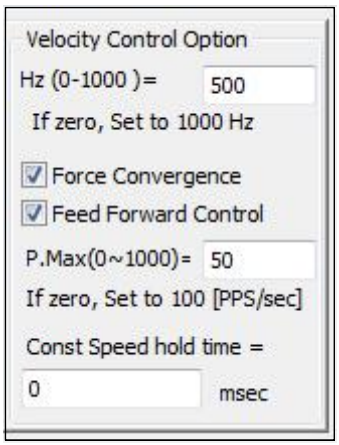


● Change [R01].Byte : 인터럽트 상태패킷을 발생을 R01레지스터(칩상태레지스터)의 하위 1바이트 변화에 대해서만 적용합니다.

하위 1바이트의 정보내용은 아래와 같습니다.
- Busy신호 유무
- Complete신호 유무
- Sync신호 유무
- Step실행상태 유무
- Emg실행상태 유무
- 큐명령 실행가능상태 유무
- MotionOp 모션동작 실행중 유무
- Alarm 발생유무

● Change [R01]. Word : 인터럽트 상태패킷을 발생을 R01레지스터(칩상태레지스터)의 하위 2바이트 변화에 대해서만 적용합니다.
하위 1바이트 정보내용에 이어 다음 정보내용이 추가됩니다.
- 현재 보간동작의 가감속모드
- 현재 보간동작의 가감속상태
- 현재 명령상태 : Normal명령상태/Sync명령상태/Wait명령상태/Delay명령상태
- 다중보간 실행상태 유무
- 실행준비를 마치고 대기하는 명령존재 유무

7.1.11. 속도제어 옵션



모션엔진은 기본적으로 보간(위치)제어기와 속도제어기의 조합으로 볼 수 있습니다. 보간(위치)제어기는 각 모션동작에 필요한 펄스를 만들고, 속도제어기는 속도설정 프로파일에 맞게 펄스의 발생주기를 생성하는 역할을 합니다.
속도제어 옵션은 이러한 속도제어기의 동작방식을 설정하는 옵션들입니다.

● Hz(0-1000) : 속도제어기의 연산주기를 설정합니다. RESET시 500Hz이며 0으로 설정할 때에도 디폴트 500Hz로 지정됩니다.

● Force Convergenc : Constant 혹은 Linear 속도제어방식은 동작펄스수만 여유가 있다면 항상 구동속도에 정확하게 도달할 수 있습니다.
그러나 S자 가감속의 경우 동작펄스 수가 충분하더라도 최종 안정속도(Steady Velocity)가 구동속도와 작은 차이를 보일 수 있습니다. 이것은 속도제어기와 위치제어기간의 제어시간과 디지탈연산의 오차에서 발생합니다. 이 옵션을 설정하면 최대한 구동속도로 강제수렴토록 속도제어기를 동작시킵니다.

● Feed Forward Control : 속도제어기는 항상 현재 동작가능한 펄스수가 얼마나 남았는지를 조사하여 감속시점을 결정합니다. 그러나 속도제어기의 동작속도는 최대 1kHz이며 펄스는 최대 100kHz까지도 동작하게 됩니다. 그러므로 속도제어기는 100Pulse이상의 연산오차를 가질 수도 있습니다. 이러한 연산결과는 감속의 종료시 초속도에 일치하지 못하는 등의 현상을 발생합니다. 이때 피드포워드제어를 수행하면 감속의 최종결과가 초속도에 도달할 수 있도록 제어를 수행합니다.

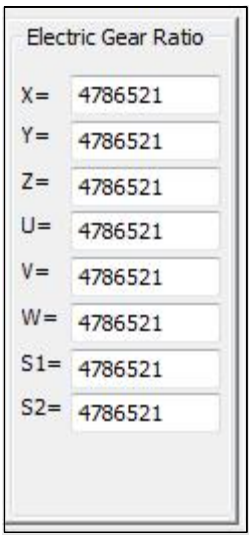
● P.Max(0~1000) : 피드 포워드 제어는 기본적으로 P(비례)제어입니다. 그러므로 제어강도를 결정하는 비례계수값을 지정합니다. 비례계수값은 0~1000이내의 값을 사용하며 리셋시 100의 값을 갖습니다. P값이 커지면 속도의 급작스러운 변화가 발생되어 마지막 감속에서 속도발진현상이 발생될 수 있습니다.

● Const Speed hold time : 정속도 일정 유지시간 으로써 속도제어기가 가감속변화사이의 안정상태(Steady)상태가 최소한 유지될 수 있도록 0 이 아닌값을 지정할 수 있습니다.
이 값은 “모션엔진 모니터링 화면”의 Vel.DwellTime 파라메터와 동일합니다. 정속동 일정 유지시간은 때때로 모션명령동작의 실행지연을 발생할 수도 있습니다. 모션명령이 완료된 후에 다음 모션명령이 시작될 때에서 가감속의 변화가 감지될 수 있으며 이때에도 정속도 일정 유지시간이 적용되면 모션명령의 실행이 지연되는 것 같은 현상을 발생합니다. 이 현상은 비교적 빠르지 않은 연속동작에서는 큰 영향을 미치지 않지만 초당 수백번의 직선보간을 수행해야 하는 등의 동작에서는 지연 영향을 미칠 수 있습니다.

7.1.12. 보간동작 옵션

보간모션 동작의 결과위치가 명령의 최종위치와 다를 때 명령의 최종위치에 도달하도록 강제로 펄스를 발생하는 역할을 합니다.

7.1.13. 전자기어비 옵션



전자기어비는 스텝모터드라이브와 같이 액추에이터를 동작시키는 드라이브가 자체 전자기어비 기능을 갖고 있지 않으며, 기계시스템이 각 축의 펄스당 이동거리가 서로 다를 때 사용될 수 있습니다.

전자기어비 기능은 모션엔진에 동작부하를 주어 전체 펄스발생최대 속도를 낮추게 됩니다. X6V3모션엔진은 전자기어비가 적용되지 않을 경우 4축 동시보간이 최대 80kHz까지 동작될 수 있지만, 전자기어비가 적용되면 최대 70kHz (실제 출력펄스의 주파수)이하로 사용하는 것이 안정합니다.

● 전자기어비의 동작방식
- 먼저 모든 축의 전자기어비 값이 동일하다면 전자기어비가 적용되지 않은 것과 동일합니다.
- 전자기어비는 각 축별로 발생되는 펄스의 비율을 조절하게 됩니다.

예를 들면, 전자기어비 X값은 1000, Y값은 800, Z값은 500 으로 설정되었고 나머지 축들 모두 1000 의 값이 설정되었다고 가정합니다.
이렇게 설정되면 모션엔진은 내부적으로 펄스발생값 1000 이라는 이동값(펄스값)으로 모든 축을 움직이려고 할 때 X축은 1000 펄스를 그대로 출력하지만 Y축은 800펄스, Z축은 500펄스로 조정되어 출력됩니다. 나머지 축들은 X축 전자기어비값으로 설정되었으므로 모두 1000펄스를 출력하게 됩니다.

이때 전자기어비의 가장 큰값을 갖는 X축을 전자기어비 주축이라고 부릅니다. 전자기어비 주축은 내부적으로 보간연산의 기준이 되며 다른 축들은 전자기어비 주축의 연산에 대한 비율로 계산되어 펄스를 발생합니다.
만약 Y축 단독의 펄스를 출력하는 보간 동작이더라 하더라도 이 전자기어 주축을 기준으로 하는 연산은 그대로 적용되어서 Y축의 실제 펄스가 800Hz로 출력되고 있더라도 내부적으로 연산하는 속도는 1000Hz와 동일하게 됩니다.

그러므로 전자기어비를 설정할 때에는 각 축의 펄스발생비율값 중 가장 작은 값과 가장 큰 값의 비율차가 1에 가까운 것이 바람직합니다.
만약 X,Y축의 전자기어비가 최대 100:1의 비율값을 갖게 된다면 단순히 X가 100펄스를 출력할 때 Y는 1펄스만을 출력하는다는 의미 뿐만 아니라, 모션엔진이 최대 100kHz까지 동작할 수 있더라도, Y축은 최대 1kHz까지만 출력할 수 있게 제한된다는 의미이기도 합니다.

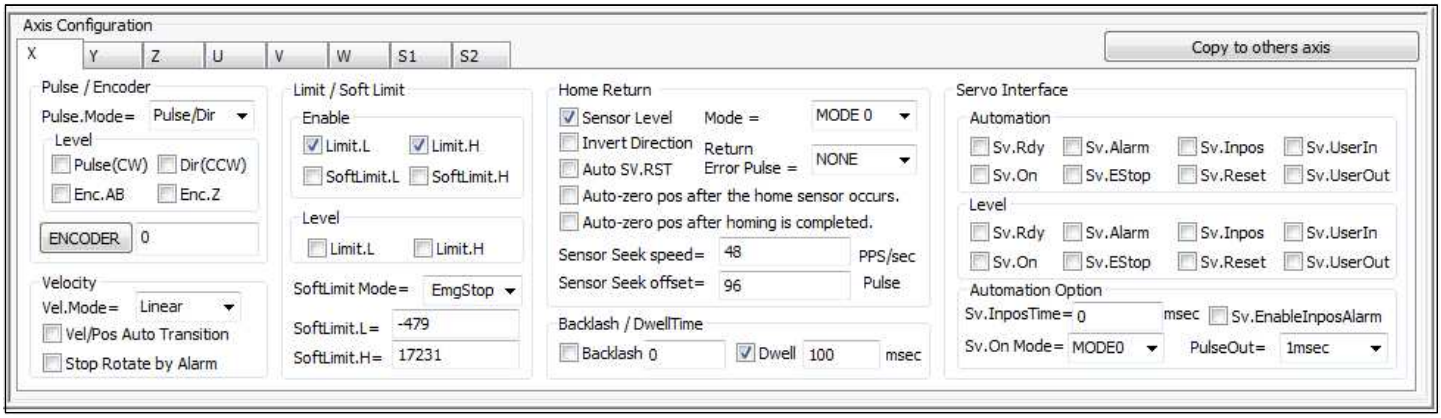
실제 전자기어비는 원호보간을 사용할 때 비율이 맞는 원의 생성을 위한 것입니다. 만약 모든 명령이 직선만을 갖

고 있다면 전자기어비를 사용할 필요는 없습니다.

※ 전자기어비 사용시 유의사항
(1) 전자기어비는 모션엔진의 실행에 부담을 주므로 서보드라이브와 같은 인텔리전트형 드라이브들은 내부에 전자기어비 기능을 갖추고 있으니 가능한 한 모터드라이브단에서 설정하는 것이 바람직합니다.
(2) 전자기어비를 적용하고 싶지 않은 축들은 적용된 축들의 전자기어비값들 중에서 가장 큰값으로 설정하면 됩니다.
(3) 각 축의 전자기어비 값의 차이가 크지 않는 것이 성능에 유리합니다.
(4) 전자기어비는 실제 원호보간을 사용하는 축에서만 지정합니다. 직선보간이나 일반적인 이동명령만을 사용하는 축들은 전자기어비를 설정하지 않습니다.

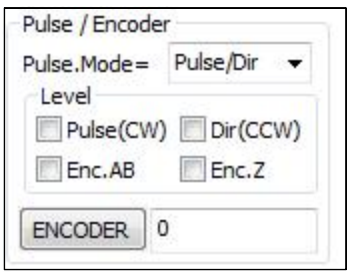
7.2. 모션엔진 축별 구성설정

모션엔진의 개별적인 축 제어에 관여되는 설정들을 관리하는 영역입니다.



각 영역별로 아래와 같은 구성설정 그룹으로 나눌 수 있습니다.

7.2.1. 펄스 혹은 엔코더



● Pulse.Mode : 모션엔진 각 축의 펄스출력신호를 Pulse-Dir/CW-CCW/None 3가지 펄스 출력방식으로 지원합니다.

Pulse-Dir방식은 펄스신호와 방향신호 방식으로써 흔히 1 Pulse방식이라고도 합니다.

CW-CCW방식은 시계방향회전펄스와 반시계방향회전펄스를 출력하는 방식으로 흔히 2 Pulse방식이라고도 합니다.

None은 실제 모터에 신호를 출력하지 않고 내부적인 연산만을 수행하는 방식으로 CNC의 머신록(Machine Lock)기능과 같은 시뮬레이션용으로 사용하기 위한 방식입니다.

● Level은 출력신호의 활성화 레벨을 설정합니다.

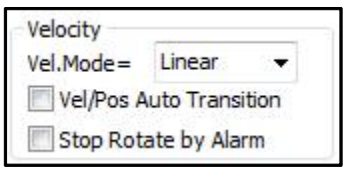
체크[√]되었다면 활성상태(Active상태)일 때 논리레벨 H(1)을 출력하는 정논리 방식이며, 체크[√] 되지 않았다면 반대의 부논리방식으로 출력합니다.

- Enc.AB / Enc.Z는 각각 엔코더의 A상 , B상, Z상을 의미합니다.

X6V3D엔진에서는 Enc.AB는 지원하지 않습니다.

●ENCODER 항목은 실제 설정값이기 보다는 표시값입니다. 모션엔진의 엔코더위치정보를 표시합니다. [ENCODER]버튼을 클릭하며 엔코더위치값이 0 이 됩니다.

7.2.2. 속도 제어



● Vel.Mode = 각 축별 모션동작의 (가감)속도 제어방식을 설정합니다.

“모션엔진 모니터링 화면”의 축별 구성설정란의 속도제어모드항과 동일합니다.

총 Const/Linear/S-Curve 3가지 동작방식중 하나를 설정합니다.

개별 축의 속도제어방식에 대한 설정은 보간명령에 적용되는 파라메터가 아니라 개별적인 축동작과 관련된 명령에서 적용되는 속도제어방식입니다. 보간명령은 1개 혹은 그이상의 축들의 조합에 의해 특정 경로를 정확히 형성하여 움직이는 명령이며, 이 명령에 적용되는 속도제어모드는 “모션엔진 모니터링 화면”의 축별 구성설정란의 Interpolation(보간)탭의 속도제어모드항에서 설정됩니다.

개별 축의 속도제어방식은 독립이송명령(IMOVE), 연속이송명령(CMOVE), 스핀들회전명령, 개별원점복귀명령(IHOME)에 적용됩니다.

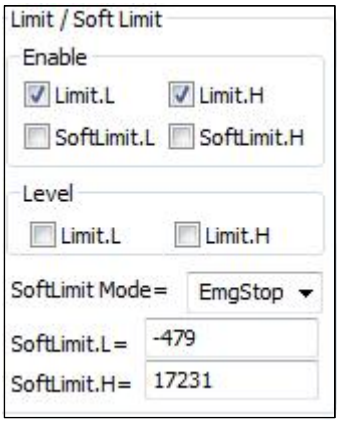
● Vel/Pos Auto Transition :(스핀들축 전용) 위칸 모션엔진의 스핀들 축은 위치를 고려하지 않고 고속의 회전제어 방식(1MHz까지 출력가능)과 보간동작(100kHz이하)과 같이 위치와 속도를 동시에 제어하는 방식을 모두 지원합니다. 이때 해당 스핀들 축이 위치를 고려하지 않고 회전만을 목적으로 제어되고 있을 경우 Velocity Control Mode상

태에 있다고 하며, 보간동작과 같이 위치와 속도 모두 제어대상이 되고 있을 경우 Position Control Mode상태에 있다고 합니다.

만약 해당 스핀들 축이 고속의 회전제어상태인 Velocity Control Mode 상태에 있을 때 보간동작명령을 전송하면 모션엔진은 알람을 발생합니다. 정확한 동작을 위해서는 고속회전제어를 정지시킨 후에 필요로 하면 스핀들 축을 원점복귀 하여 정위치 시키고 나서 보간동작명령을 수행하는 것이 바람직합니다.
이때 스핀들을 정지하되 정위치 시키지는 안는다면 Vel/Pos Auto Transition 옵션을 설정하게 되면 고속의 회전제어상태에서 보간동작명령을 전송하면 자동으로 감속정지한 후에 안전하게 보간동작을 수행하게 됩니다.

● Stop Rotate by Alarm : (스핀들축 전용) 알람이 발생되었을 때 위치제어축들은 정해진 감속정지방식으로 정지하게 됩니다. 그러나 회전제어상태(Velocity Control Mode)상태에 있는 스핀들의 경우 정지하지 않습니다.
만약 알람상태에서 스핀들 축을 정지시키려면 이 옵션을 활성화 합니다.

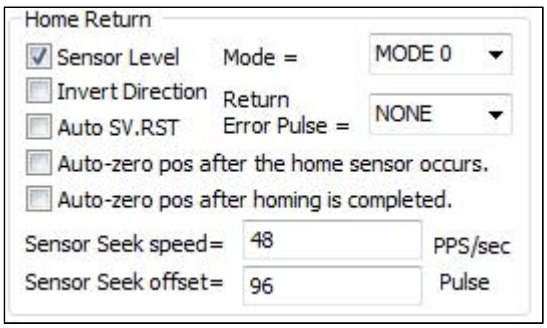
7.2.3. 리미트, 소프트리미트



리미트와 소프트리미트 기능은 각각의 동작축이 특정동작범위 영역을 벗어났을 때 알람을 발생시키는 역할을 합니다.
● Enable : 리미트와 소프트리미트 기능의 활성화 여부를 설정합니다.
리미트 기능은 모션엔진의 LIMIT신호에 의해 발생되고,
소프트리미트 기능은 모션엔진의 내부 소프트리미트 파라메터에 의해 발생됩니다.
● Level : 리미트 신호입력의 동작레벨을 결정합니다. 체크[√] 상태일 때 논리레벨 H(레벨 1)의 상태에서 동작하는 정논리레벨이며, 체크[√] 상태가 아닐 때 반대방식의 부논리레벨로 동작합니다.
● SoftLimitMode 는 소프트리미트가 발생되었을 때 어떤 방식으로 동작할 것인지를 결정합니다. 총 3가지 방식으로써 Notify / DecStop / EmgStop 방식을 지원합니다.
- Notify : 알람신호만을 발생시키고 기기를 정지하지 않습니다.
- DecStop : 알람신호와 함께 감속정지방식으로 정지합니다.
- EmgStop : 알람신호와 함께 즉시정지방식으로 정지합니다.
● SoftLimit.L / SoftLimit.H : 최대 32비트 부호있는 정수값의 한계까지 가능합니다.

7.2.4. 원점복귀

원점복귀는 기본적으로 외부의 홈센서를 기준으로 동작합니다.X6V3 엔진은 3가지 방식의 원점복귀 동작방식을 갖고 있으며, 아래와 같은 여러 가지 옵션설정을 포함합니다.



●Senser Level: 체크[√]상태일 때 홈센서 신호가 H레벨 상태에서 홈센서 감지상태(활성화상태)가 되는 정논리방식이며, 체크[√]상태가 아닐 때 반대의 부논리방식입니다.

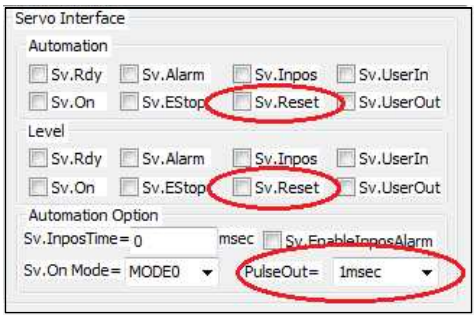
● Mode : 원점복귀 방식은 Mode0/Mode1/Mode2 3가지 방식이있으며 각각 아래와 같은 동작을 합니다.
-Mode 0 동작방식 :
①홈센서 검색이동(Search Home Sensor)
②홈센서 검출 감속정지 (Dec Stop after Sensing Home)
③드웰지연 (동작안정지연)

④역방향 홈센서 정밀검출 이동 (Seek Home Sensor)
⑤홈센서 검출
⑥홈센서 OFF상태에서 정지

-Mode 1 동작방식 : Mode 0의 동작방식에서 마지막 6번 동작에서 정지하지 않고 각축의 엔코더 z상의 변화검출에서 정지합니다.
①~⑤ : Mode 0 방식과 동일
⑥ 홈센서 OFF상태에서 엔코더 Z상 변화감지 시작
⑦ 엔코더 Z상 감지시 정지

-Mode 2 동작방식 : 일반적인 원점복귀는 홈센서를 검출후 역회전하여 다시금 홈센서를 검출하고 홈센서를 이탈시 정지하는 동작을 수행합니다. 그러나 스핀들과 같은 장치들의 경우 홈센서가 구동장치의 1회전이내에 검출되며 역회전동작이 필요로 하지 않을 경우 Mode2 동작방식을 사용할 수 있습니다. Mode2 동작방식은 아래와 같이 2개 동작으로 간단하게 구성됩니다.
①홈센서 검색이동(Search Home Sensor)
②홈센서 검출시 급정지(Immediately Stop after Sensing Home)

● Invert Direction : 일반적으로 홈센서의 검출하는 초기동작은 - 진행방향(CCW방향)으로 진행됩니다. 이 동작방향을 반대로 설정할 때 사용하는 옵션입니다.
● Return Error Pulse : 원점복귀 완료 후 원점복귀 오류를 검출하게 될 오차 펄스수를 지정합니다.
None 으로 지정할 시 원점복귀 완료 후 위치오차를 검출하여 알람을 발생시키는 동작을 수행하지 않습니다.
그 외 펄스수를 지정하면 원점복귀 후 0의 위치에서 지정될 펄스 수 만큼의 오차를 벗어나면 원점복귀 알람을 발생합니다.
● Auto SV.RST : 원점복귀 완료 후 서보리셋 펄스 신호를 발생하는 옵션입니다.



이 기능이 제대로 동작하기 위해서는 [Servo Interface]의 서보리셋신호(SV.Reset)의 자동동작(Automation)을 설정하고 서보리셋 활성화 신호의 레벨(Level)을 설정한 후, 펄스의 폭에 대한 지정(PulseOut)을 해주어야 합니다.

● Auto-zero pos after the home sensor occurs :홈센서가 검출될 때마다 해당 축의 위치를 강제로 0으로 설정하는 옵션입니다.
● Auto-zero pos after homing is completed : 원점복귀 동작이 완료된 직후 해당 축의 위치를 강제로 0으로 설정하는 옵션입니다.
● Sensor Seek Speed : 원점복귀 동작(Mode0/Mode1에서만) 중 홈센서를 검출하기 위한 역회전 동작에서 센서를 정밀하게 검출하는 속도를 지정합니다. 단위는 PPS[Pulse Per Second]입니다.
● Sesnor Seek Offset : 이 설정은 IHome 명령에서는 효과가 없으며 Home명령에서만 유효합니다. Home 명령은 원점복귀 동작(Mode0/Mode1에서만) 중 홈센서 검출을 위한 역동작에서 가감속이 이루어 지며 이때 홈센서의 위치를 기준으로 Seek Offset 펄스수 이전에 감속정지하여 Seek Speed로 정밀하게 홈센서를 검출하게 됩니다.
※ 원점복귀시 동작속도(①홈센서 검색속도)
원점복귀 명령은 IHome 명령과 Home명령 두가지가 있습니다. IHome명령은 보조모션엔진을 사용하여 수행하므로 모션엔진에 내장된 보조모션엔진의 수만큼 동시에 원점복귀를 수행합니다. Home명령은 주모션엔진을 사용하여 수행하므로 한번에 하나의 축만이 원점복귀 수행이 가능합니다. 이들 두 종류의 원점복귀 명령의 동작속도는 각 축별 속

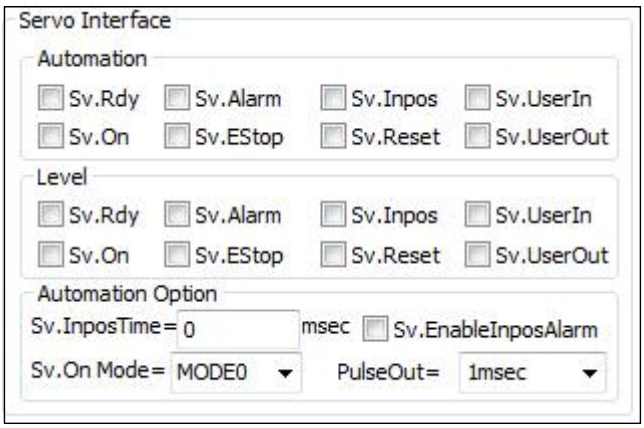
도와 가속도 성분으로 동일하게 적용받습니다.
그러나 원점복귀시 센서를 검출해서 감속정지하는 방식은 Home명령의 경우 보간동작 (가감)속도제어방식에 의존하고 IHome명령의 경우 각 축별 (가감)속도제어방식에 적용받습니다.
X6V3 엔진의 경우 1개의 주모션엔진과 4개의 보조모션엔진이 존재하므로 동시 원점복귀동작은 최대 5개의 축만 가능하며 나머지 하나의 축은 이들 중 먼저 종료한 모션엔진에 할당해서 동작시킬 수 있습니다.

7.2.5. 백래쉬 / 드웰타임

● Backlash 설정값은 모터가 회전방향을 바꿀 때마다 추가로 발생하여 보상하는 펄스값입니다. 이때 펄스의 출력속도는 Backlash가 발생된 시점의 속도를 그대로 유지합니다. 그러므로 백래시 동작을 사용할 경우 동작속도에 주의해야 합니다.
● Dwell : 드웰타임을 msec단위로 지정합니다. 각 축마다 지정하는 드웰타임은 각 축별로 동작되는 명령에서 유효합니다. 현재 원점복귀동작에서 센서검색 후 감속정지한 후에 동작안정화 지연시간에 사용됩니다.

7.2.6. 서보인터페이스

서보인터페이스는 제어되는 대상 축이 서보드라이브와 같은 장치들을 사용될 때 각 서보드라이브와 연결되어 입출력하는 신호들의 동작방식들을 규정합니다.



일반적인 서보드라이브들의 주요한 공통신호들에 대한 구성성절만이 있으므로 추가적인 서보드라이브 신호들을 사용할 때는 일반입출력(GIO) 혹은 확장입출력(EIO)신호선을 이용해야 합니다.
서보인터페이스 신호는 모션엔진마다 각각의 신호에 대한 지원여부와 동작방식에 있어서 차이가 날 수 있습니다.

※ 모션엔진에서 지원하는 서보입출력 신호들
-입력신호 : 준비신호(Ready), 알람신호(Alarm), 위치결정완료신호(Inposition), 사용자 지정신호(UserIn)
-출력신호 : 서보모터전원투입신호(On), 비상정지신호(Estop), 리셋신호(Reset), 사용자 지정신호(UserOut)

● Automation : 서보드라이브의 입출력신호들을 사용자가 임의적으로 프로그램하여 사용하지 않고 모션엔진 내부에서 자동으로 신호를 발생시키는 기능을 설정하는 항목입니다. 각 신호별 자동화동작상태에서의 작동방식은 아래와 같습니다.
-Sv.Rdy(준비신호자동화) : 준비신호가 활성화 되지 않은 상태에서 모션동작명령이 실행되면 알람을 발생합니다.
-Sv.Alarm(알람신호자동화) : 서보드라이브의 알람신호는 모션엔진의 엔진의 알람동작으로 연결됩니다.
-Sv.Inpos(위치결정완료신호자동화) : 보간명령이 시작될 때 항상 이전명령에 대한 Sv.Inpos신호를 확인하며 Sv.InpsTime 으로 지정된 시간이내에 Sv.Inpos 신호가 검출되지 않으면 알람이 발생시키거나 그대로 동작시킬 수 있습니다. Automatio Option 기능과 함께 사용됩니다.
-Sv.UserIn/Sv.UserOut : 사용자지정신호에 대한 자동화 동작은 단순히 Sv.UserIn신호가 Sv.UserOut신호로 연결되는 동작입니다.

-Sv.On : 서보ON신호에 대한 자동화동작은 총 4가지 방식으로 [Automatio Option]의 Sv.OnMode에 따른 동작을 수행합니다. [Automatio Option]의 Sv.OnMode 설명을 참조해 주세요
-Sv.Estop : 모션엔진의 내부 알람에 의한 비상정지(비상실행)상태일때 서보 EStop신호가 활성화 됩니다.
- Sv.Reset : 원점복귀 완료시에 자동으로 지정된 펄스폭의 펄스신호를 발생합니다.
[Home Return]항목을 참조하세요.
- Sv.UserOut : Sv.UserIn신호를 그대로 출력합니다.

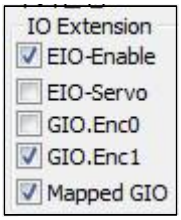
● Level : 각 서보입력 혹은 출력신호의 활성화 상태를 지정합니다. 이 설정의 체크여부로써 각 신호들의 반전입출력이 가능합니다.

● Automation Option : 서보입출력 자동동작에 필요로 하는 옵션정보들을 설정합니다.
- Sv.InposTime : 서보Inpos 신호가 자동화되었을 때, 보간명령의 시작시 Inpos신호를 대기하는 최대시간을 msec단위로 지정합니다.
- Sv.EnableInposAlarm : 서보 Inpos 신호가 자동화되었을 때, 보간명령의 시작시 Sv.InposTime으로 지정된 시간동안 서보 Inpos 신호를 감지하지 못하면 알람을 발생하도록 지시합니다.
- Sv.OnMode : 총 4가지 방식중 하나로 자동화 동작을 실행할 수 있습니다.
① Mode 0 : Servo Alarm 신호가 발생될 때를 제외하고 항상 ON상태 출력
② Mode 1 : Servo Alarm 신호 혹은 동작중지형 알람상태일 때는 OFF상태 출력, 그 외 항상 ON상태 출력
③ Mode 2 : Mode 0 의 ON 조건을 만족하고 모션동작이 실행될 때만 ON상태 출력
④ Mode 3 : Mode 1 의 ON 조건을 만족하고 모션동작이 실행될 때만 ON상태 출력
- PulseOut = (1msec/10msec/100msec/500msec() 중 하나를 선택합니다. PulseOut 파라메터는 Sv.Reset출력과 같이 지정된 일정시간동안 ON상태인 펄스형태를 출력할 때, 펄스주기를 설정하는 항목입니다.

7.3. 모션엔진 GIO/EIO 입출력 구성설정

일반입출력의 특별한 기능을 설정하거나, 추가장착된 확장IO(EIO 혹은 ExtIO)보드의 동작, 구성설정하는 영역입니다.

● IO 확장옵션



- EIO_Enable : EIO확장 보드가 설치될 때 활성화 시킵니다.
- EIO_Servo : EIO확장 보드가 서보입출력 신호를 지원할 때 활성화 시킵니다.
- GIO.Enc0 : GIO의 특정입력포트에 엔코더 채널 0번 동작을 수행시킬 때 활성화 시킵니다.
- GIO.Enc1 : GIO의 특정입력포트에 엔코더 채널 1번 동작을 수행시킬 때 활성화 시킵니다.
- Mapped GIO : EIO확장 보드의 입출력포트를 GIO입출력 포트로 맵핑합니다.

● 확장보드 시스템명령 : 설치된 확장보드에 리셋/비상정지/재시작 명령을 실행합니다.

확장보드의 시스템명령에 대한 동작은 해당 확장보드의 매뉴얼을 참조해 주세요.

● 확장보드 연결 및 입출력확인

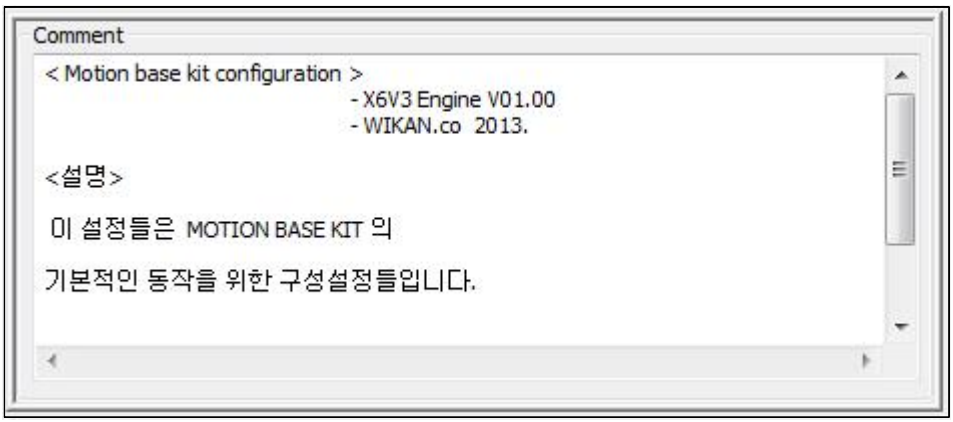
● MPG기능을 갖는 확장보드의 MPG 입출력 포트 확인

- status1/2 : MPG 입력포트 활성화 상태값을 체크[√]로 표시합니다.
- Count : MPG의 현재 엔코더 카운터값을 표시합니다.

※ 확장보드에 대한 상세한 내용은 확장보드 매뉴얼을 참조해 주세요.

7.4. 모션엔진 구성설정 설명문

모션마스터프로그램의 다소 복잡한 구성설정내용들은 해당 기계시스템에 적합하게 설정된 후 파일로 저장되어 언제든지 로드되어 모션엔진이 장착된 보드를 기계시스템에 맞게 간단하게 셋팅할 수 있습니다. 이때 해당 구성파일의 셋팅내용이 어떠한 기계시스템을 위한 정보인지 등에 대한 내용을 사용자가 임의의 설명을 부여하여, 구성파일 로드시 바른 구성파일인지를 빠르게 확인하는 기능을 갖습니다.



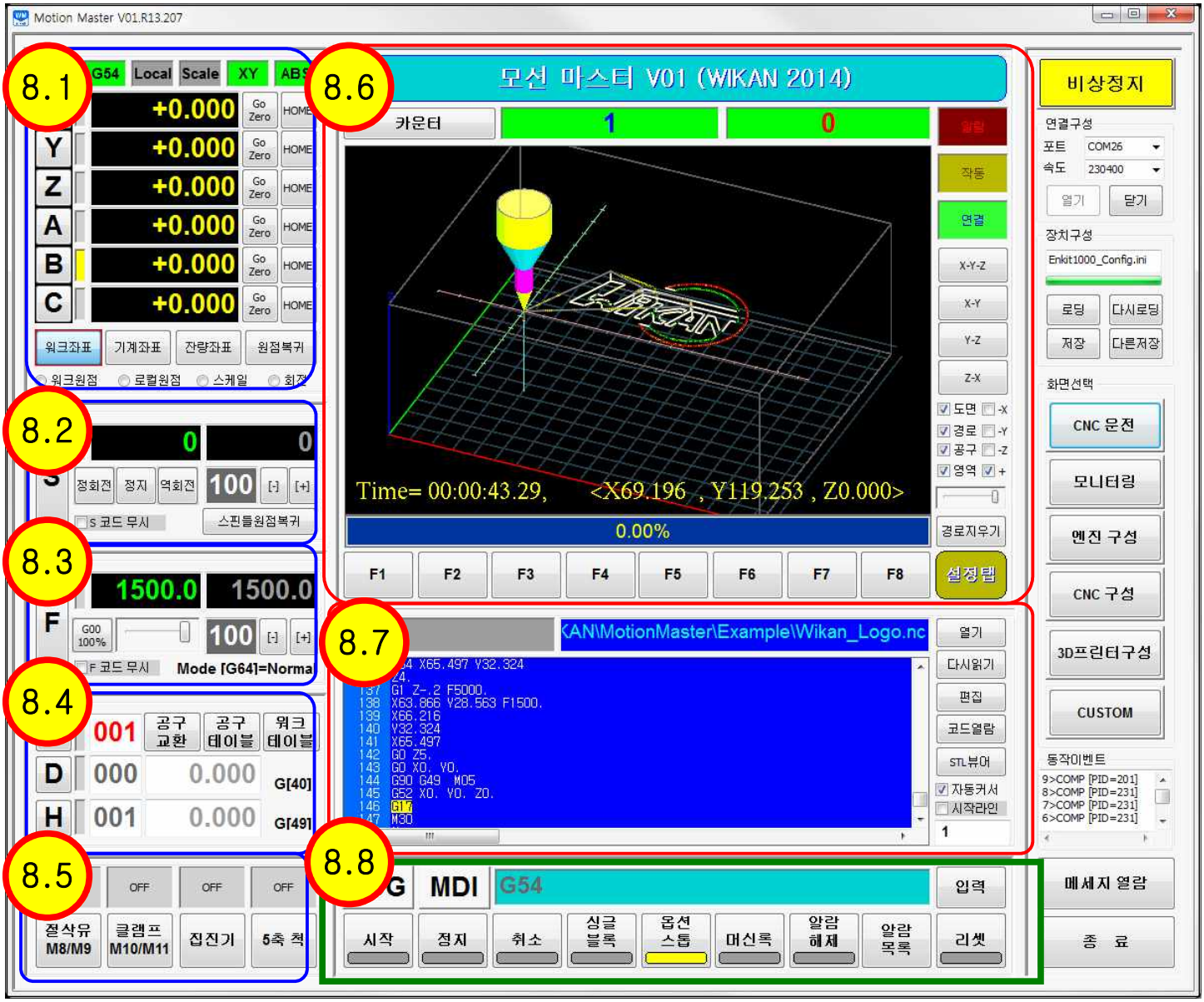
7.5. 업로드, 다운로드 기능

- RELOAD : 현재 모션엔진의 구성내용을 “모션엔진 구성설정 화면” 에 표시합니다.
- REFRESH : 현재 “모션엔진 구성설정 화면”의 구성내용을 모션엔진에 전달합니다.

실제 “모션엔진 구성설정 화면”의 내용을 변경하는 즉시 모션엔진에 변경된 구성내용이 전달됩니다. 그러므로 이 기능들은 다시 한번 확인겸 실행하거나 즉시 변경된 내용이 모션엔진에 적용되지 않았을 때 다시금 시도하는 버튼입니다.

8. CNC 운전화면

CNC 운전화면은 위칸모션보드를 이용하여 밀링형태의 CNC장치로써 사용할 때 사용될 수 있는 화면입니다.
(모션마스터 프로그램 버전에 따라 기능과 화면이 달라질 수 있습니다.)



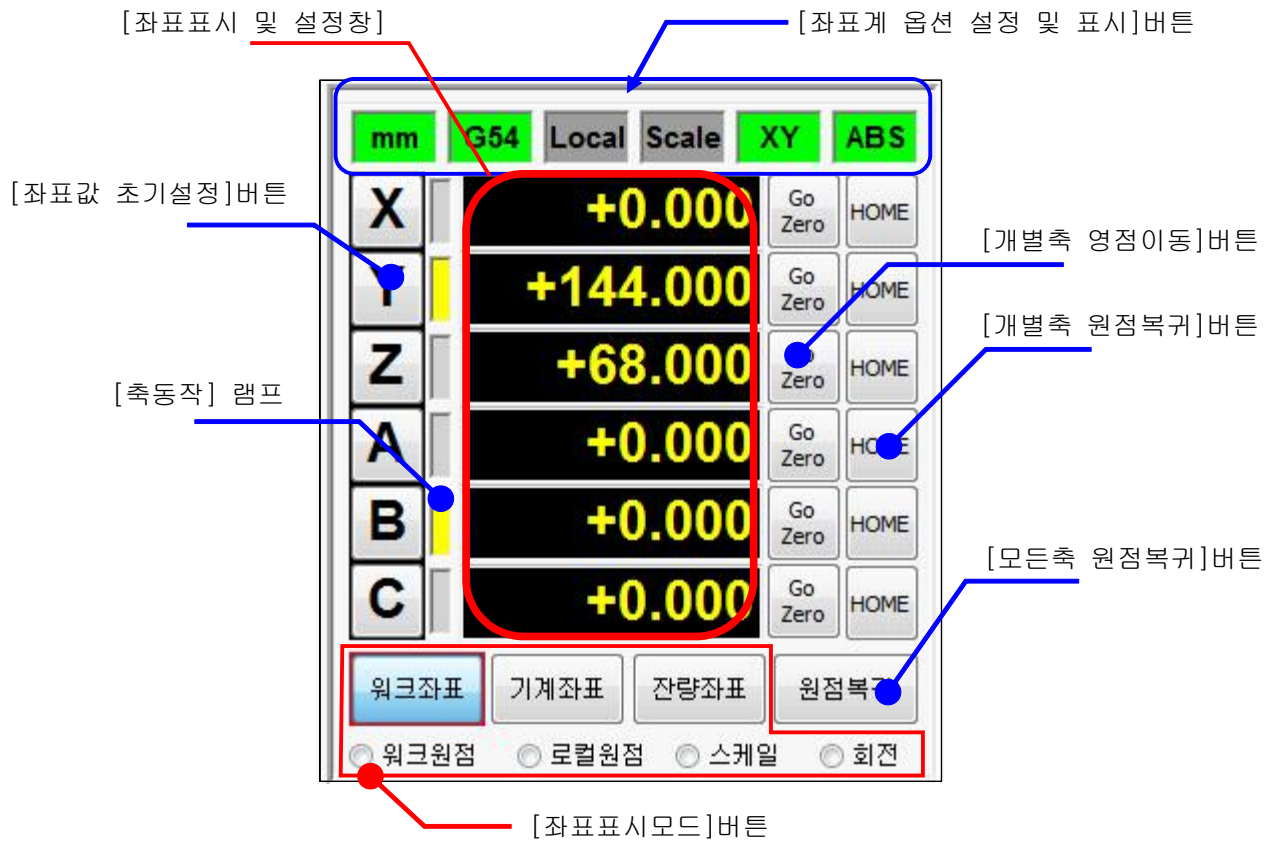
각 기능별 화면을 분류하면 아래와 같습니다. (그림번호참조하세요)

(8.1) 위치좌표 제어화면
(8.2) 스핀들(주축) 제어화면
(8.3) 피드(이송속도) 제어화면
(8.4) 공구 및 보정 제어화면
(8.5) 주변장치 제어화면
(8.6) 메인모니터 화면
(8.7) NC코드파일 관리화면
(8.8) 메인 조작화면

먼저, CNC운전화면의 기능들을 간단히 소개합니다.
- G코드 기반입니다.
- 초당 최대 300~1000블록을 실행합니다.
- 머신록(MachLock)기능에 의한 시뮬레이션이 가능합니다.
- 싱글블록실행과 옵션널 스톱기능이 있습니다.
- 기계좌표계(G53), 공작물좌표계(G54,G55...G59), 로컬좌표게(G52), 기계좌표변경(G92)을 지원합니다.
- 스케일(G51/G50), 회전(G68/G69) 지원합니다.
- 다양한 고정사이클(드릴,카운터보어,고속심공,고속펙킹, 탭핑, 보링,정밀보링등...)지원 합니다.
- 서브프로그램호출(M98/M99) 기능이 있습니다.
- 스핀들 및 피드오버라이딩 기능이 있습니다.
- 다양한 입출력장치 ON/OFF 기능이 있습니다.
- 연속이동과 스텝이동이 가능한 조그기능이 있습니다.
- MDI(Manual Direct Input) 기능이 있습니다.
- 3D 공구이동추적선기능이 있습니다.
- 임의 라인에서 G코드블록을 시작할 수 있습니다.
- 3D 프린팅을 위한 코드를 지원합니다 (3d 프린팅코드를 생성하는 CURA 소프트웨어를 지원합니다.)
- 사용자 정의 기능의 키를 사용할 수 있습니다.
- 매크로확장 및 V스크립터 지원합니다.

8.1. 위치좌표 제어화면

기계의 위치 및 좌표시스템과 관련된 제어화면입니다.



● 기계좌표는 기계에 전원이 투입되고 초기 원점복귀를 수행한 후 [기계홈 좌표 설정]버튼을 각 축마다 클릭하면 초기화 됩니다. 이때 설정되는 기계좌표값은 “CNC 구성설정화면”의 각 축별 구성요소항 중 Home Orgin=위치값 으로 결정됩니다.

공작기계를 기계좌표계상의 위치로 동작시킬 때 G53코드를 사용합니다. 이후 G코드의 X,Y,Z좌표값은 기계좌표값으로 인식됩니다.

※ [기계홈좌표설정]버튼을 누르게 되면 현재좌표값이 기계 Orgin좌표값으로 설정됨과 동시에 다른 좌표계들(공작물좌표계 등)은 0으로 설정됩니다.

● 공작물좌표는 기계좌표계에서 설정된 기준좌표값 0,0,0을 사용하지 않고 가공재료(모재)의 임의의 위치를 0,0,0으로 사용할 때 적용하는 좌표계입니다. 기본적으로 프로그램 실행초기화단계에서 G54 공작물좌표계를 사용합니다.(물론 바꿀 수 있습니다) 또한 다수의 공작물좌표계(G54,G55,G56,G57,G58,G59)를 사용할 수 있습니다.
G54 명령으로 공작물좌표계를 지정하였다면 이후 G코드의 X,Y,Z좌표값은 공작물 좌표계가 기준이 됩니다.

※ 임의의 지점을 공작물좌표계 원점 0,0,0으로 설정하는 방법은 아래와 같습니다.

① 가장 먼저 전원을 투입하면 기계원점복귀를 수행합니다.
② [기계홈좌표설정]버튼을 눌러 현재기계의 위치가 인식되도록 합니다.
③ 가공 모재를 올려놓습니다.
④ MDI 명령줄에 G54를 입력하여 공작물좌표계로 전환합니다.(기본적으로 G54좌표계이므로 생략가능합니다.)
⑤ 조그기능을 이용하여 원하는 위치로 작업공구(스핀들 등)를 옮깁니다.
⑥ 아래와 같은 방법으로 현재 위치를 공작물좌표계 0,0,0 원점을 설정합니다.
- 먼저 [좌표표시모드]를 Work로 설정합니다.
- 원하는 축의 [좌표값 0설정] 버튼을 눌러 해당 축에 대한 공작물좌표값이 0이 되로록 합니다.
※ [좌표표시창]은 지정된 [좌표표시모드]에 대한 좌표값을 표시함과 동시에 설정도 가능합니다. 그러므로 [좌표표시창]에서 원하는 축의 표시값을 0,0,0으로 기입하여 바꿀 수도 있습니다. 이 방식은 현재위치를 0,0,0 아닌 임의의 위치설정을 가능하게 합니다.

● 로컬좌표계는 기계좌표계 혹은 공작물좌표계의 원점을 기준으로 또 다시 새로운 원점을 사용할 수 있게 합니다. G52 X100Y100을 입력하면, 현재 기계좌표계일 때는 기계영점0,0,0을 기준으로 X100,Y100위치를 현재좌표 X0,Y0로 인식하게 됩니다. 공작물좌표계일때는 공작물좌표0,0,0을 기준으로 X100,Y100위치를 현재좌표 X0,Y0으로 인식하게 됩니다.
로컬좌표계를 취소하려면 G52 X0Y0Z0 하면 G52 로컬좌표계적용은 무효화됩니다.

상기 3가지 좌표계를 적용하여 기계의 최종위치를 계산하면 아래와 같이 됩니다.

기계최종 위치 = 공작물좌표계영점위치(혹은 기계 0,0,0 영점위치) + 로컬좌표계영점위치 + G코드 X,Y,Z좌표값

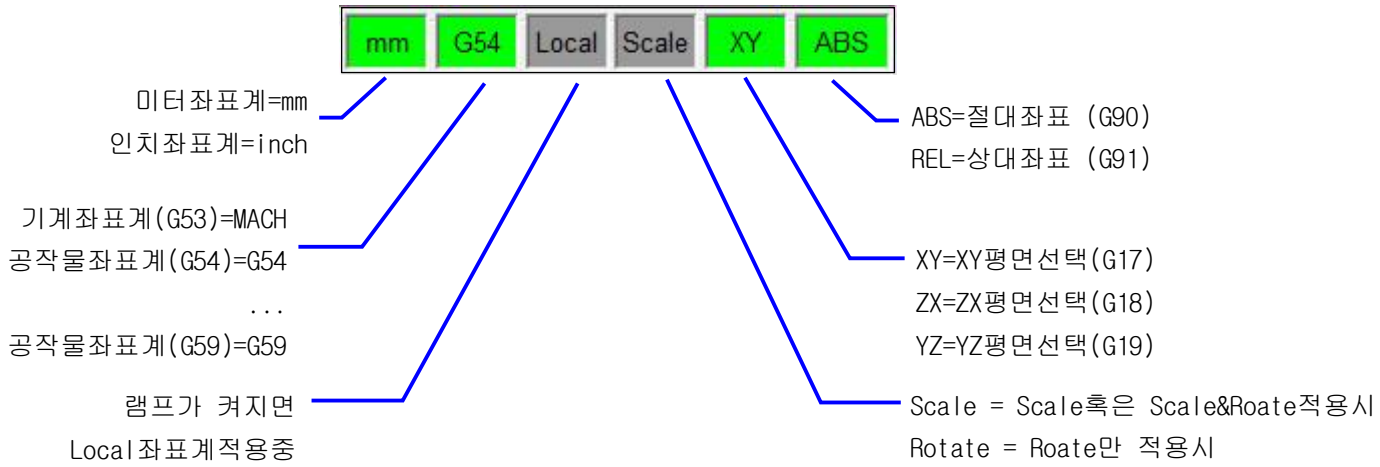
● 좌표표시모드

좌표표시창에 표시할 좌표의 종류를 선택합니다. 좌표표시장에 마우스포커스를 두면 편집이 가능하며 이때 좌표표시모드에 따라 제각기 다른 기능을 수행합니다.

아래의 좌표표시모드에 따른 좌표표시창의 표시기능과 편집기능에 대한 표를 참조하세요. <표>

좌표표시모드	좌표 표시 창		추가설명
	좌표표시기능	좌표편집기능	
워크좌표	공작물좌표계+로컬좌표계 표시	현재기계위치를 공작물좌표계의 임의위치로 지정	현재좌표계가 기계좌표계로 설정되어 있으면 , 기계좌표값과 동일합니다.
기계좌표	기계좌표계+로컬좌표계 표시	현재 기계위치를 변경	
워크원점	공작물좌표계 원점좌표 표시	공작물좌표계 원점좌표 편집	
로컬원점	로컬좌표계 원점좌표 표시	로컬좌표계 원점좌표 편집	
스케일	각 축의 스케일값 표시	각 축의 스케일값 편집	1이 아닌값을 가지는 축이 있을 때 스케일 적용됩니다.
회전	X,Y,Z축은 회전 기준위치값 표시 B축은 회전각도 표시	X,Y,Z축은 회전 기준위치값 편집 B축은 회전각도 편집	회전각도가 0이 아닐 때 회전적용됩니다.

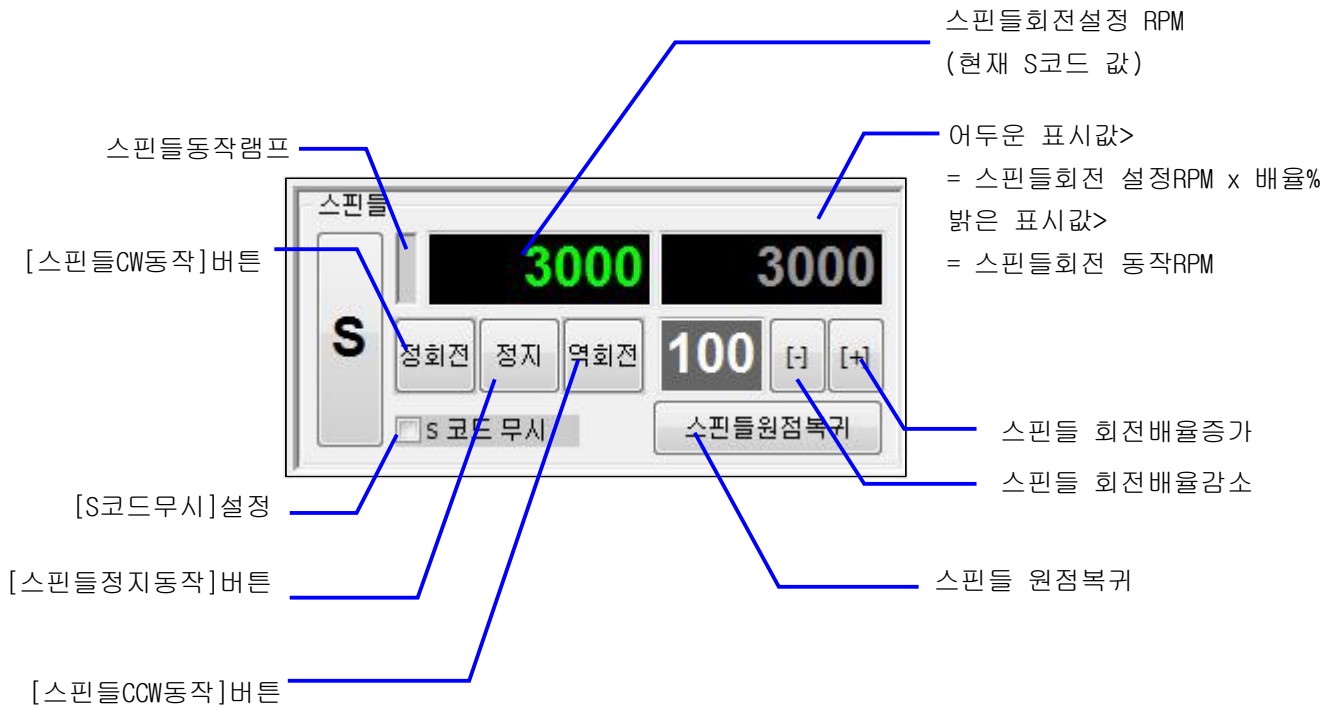
● 좌표계 옵션설정 및 표시 버튼



각 버튼은 현재 좌표계옵션상태를 표시하는 램프인 동시에 마우스로 클릭하게 되면 옵션설정이 바뀌게 되는 버튼 역할을 합니다.

8.2. 스핀들(주축) 제어화면

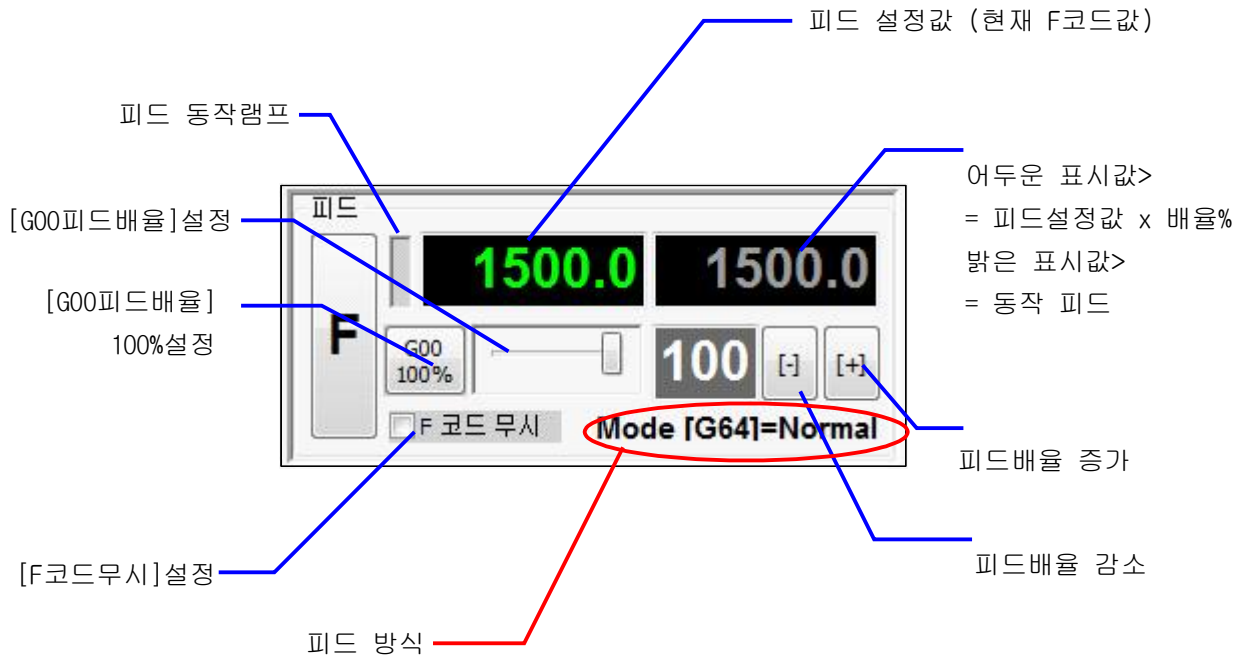
주축인 스핀들의 제어하는 화면입니다.



※[S코드 무시]설정을 체크[√]하면 자동실행시 내부G코드의 S코드에 대한 처리를 수행하지 않습니다.
이것은 G코드내부의 프로그램된 스핀들 속도가 아닌 사용자 임의의 스핀들 동작으로 사용될 때 유용합니다.
이 옵션이 실행될 경우 노란색으로 점멸하여 사용자에게 알려줍니다.

8.3. 피드(이송속도) 제어화면

기계의 이동속도를 제어하는 화면입니다.



※ 피드 방식에 의한 속도모드변환

피드방식은 G64(노멀커팅), G63(하드커팅), G62(소프트이동), G61(정확한위치이동) 의 4가지 방식입니다.

일반적으로 G64방식의 피드동작방식을 사용합니다.

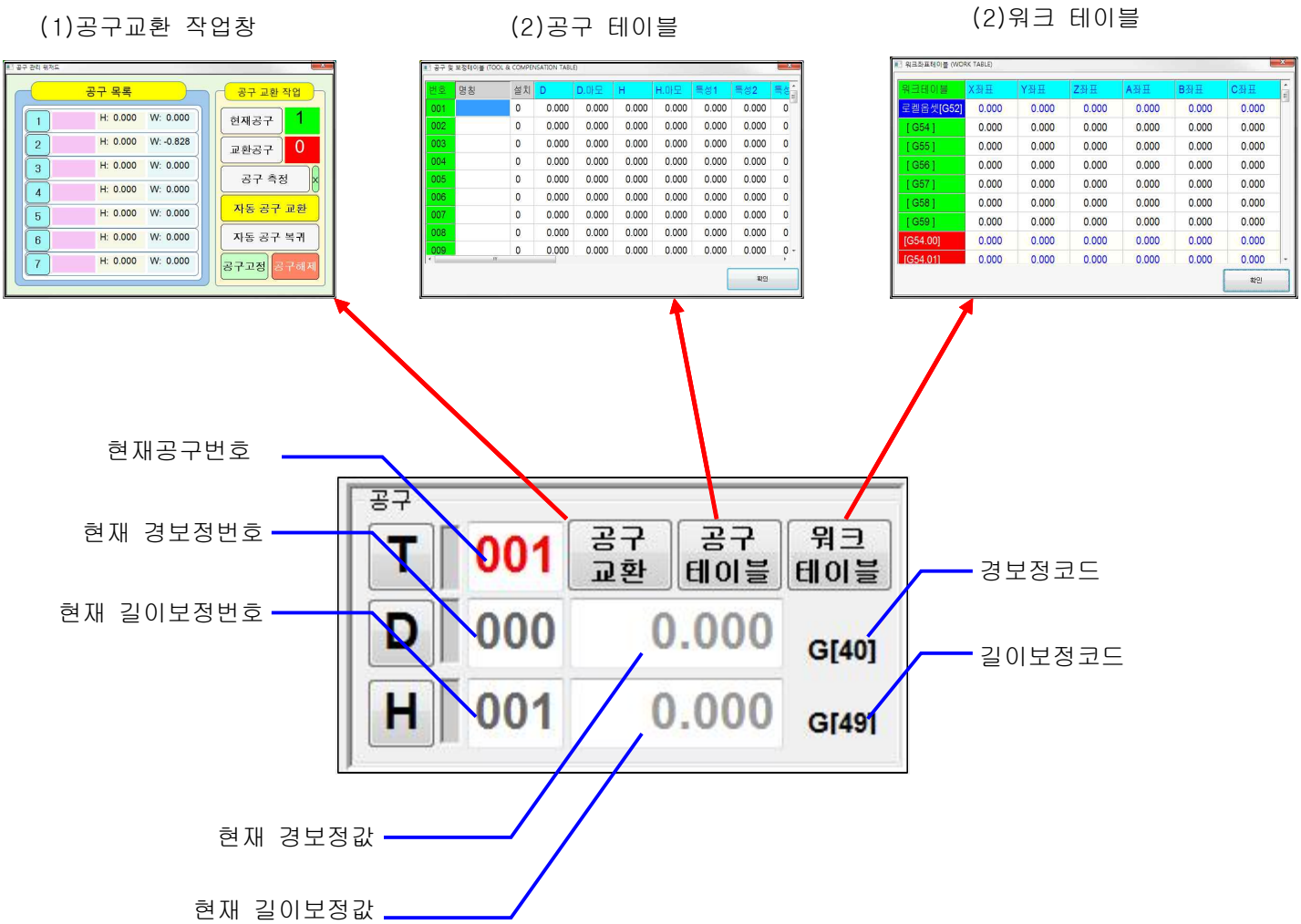
G63(하드커팅)방식은 저관성, 저속에서 G01과 같은 절삭이동시 가감속없이 정속으로 이동하는 방식입니다.

G62(소프트커팅)방식은 절삭이동시 S자 가감속에 의해 부드럽게 이동하는 방식입니다.

G61의 정확한 이동방식은 모든 이동명령단위마다 가감속하여, 명령을 단위별로 이동하는 방식입니다.

8.4. 공구 및 보정 제어화면

공구와 공구보정에 관련된 제어화면입니다.



공구교환 작업창은 스크립터코드로 작성되어 사용자가 임의적으로 디자인할 수 있는 화면입니다.
(현재 공국교환 작업창 화면은 프로그램 버전에 따라 달라질 수 있습니다.)

[공구교환] 버튼을 누르면 실행되는 스크립터코드파일은 프로그램설치폴더의 하위폴더 ScriptCode폴더에 있는 "UserKey_ToolChangeMain_UI.Code"파일입니다.
이 파일의 내용은 모션마스터에 내장된 V스크립터 인터프리터에 의해 실행됩니다. V스크립터에 대한 문법내용은 "V스크립터언어 “편을 참조하실 수 있습니다.
여기서는 간단하게 내부코드를 살펴보겠습니다.

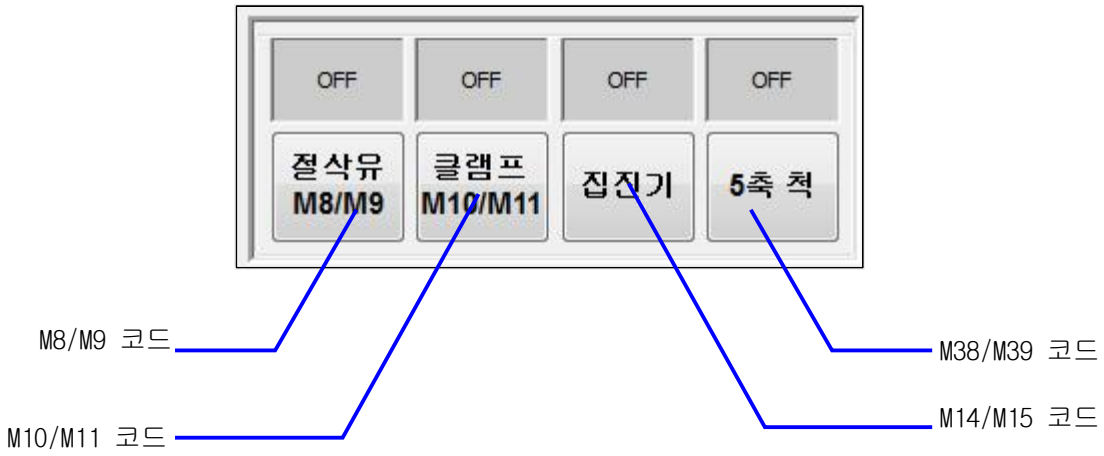
```
{
        /*-------------------------------------------

                화면 전체 클리어

        -------------------------------------------*/
        VAR  dlgMssg[10](0);
        dlgMssg[0] = "type=all, clear=1, show=0";
        WINC_SET_USER_WINDOW(1, dlgMssg, 1);
}

// 공구교환 화면을 호출한다.
call("WWToolChangeCodeWWToolChangeMain_UI.code");

return(0);
```

ToolChange폴더의 ToolChangeMain_UI.code를 불러와서 최종 실행합니다.

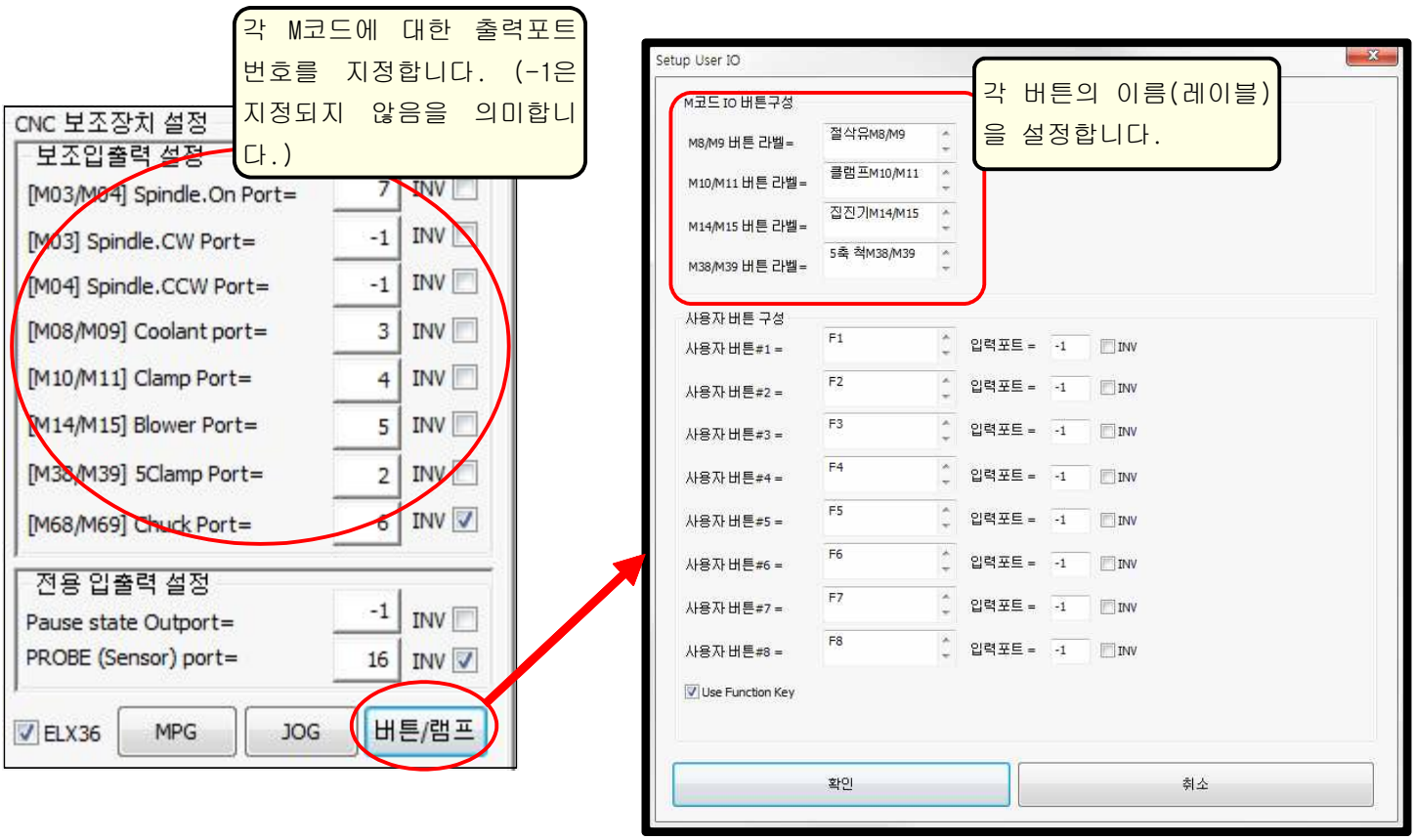
공구테이블과 워크테이블에 대한 자세한 설명은 “여러가지 CNC동작 테이블” 편을 참조하세요.

8.5. 주변장치 제어화면

몇가지 지정된 M코드와 관련된 몇가지 주변장치를 제어하는 화면입니다.
이 화면의 명칭과 연결포트번호는 “CNC구성” 화면을 통해 사용자가 임의로 변경할 수 있습니다.

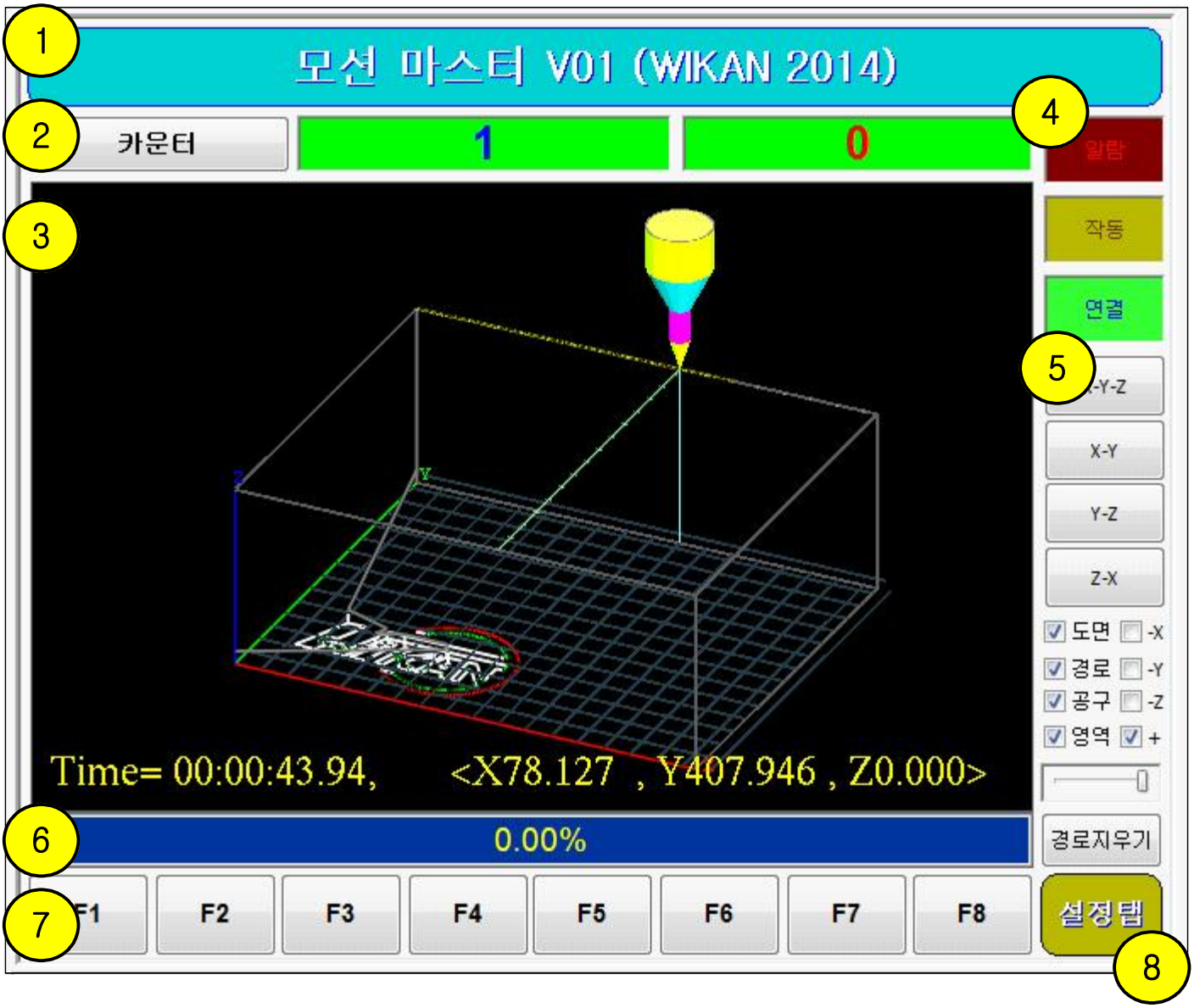


각 M코드에 대한 설정과 명칭(레이블)설정은 아래와 같이 “CNC 구성” 화면의 CNC보조장치설정 그룹에 포함되어 있습니다.



8.6. 메인 모니터화면

메인 모니터화면은 전체적인 동작모니터링를 3D그래프를 통해 표시하며 Fn키와 같은 사용자기능키 등을 포함하고 있는 화면입니다.

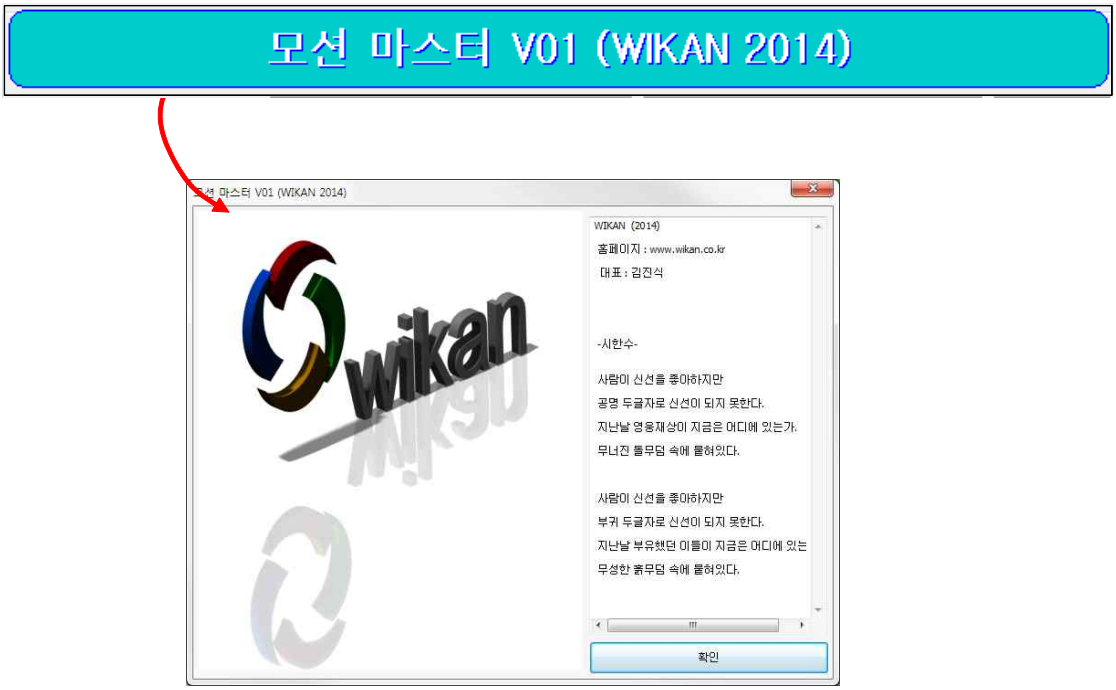


이 화면구성을 8가지 기능별로 구분하여 정리하면 아래와 같습니다.

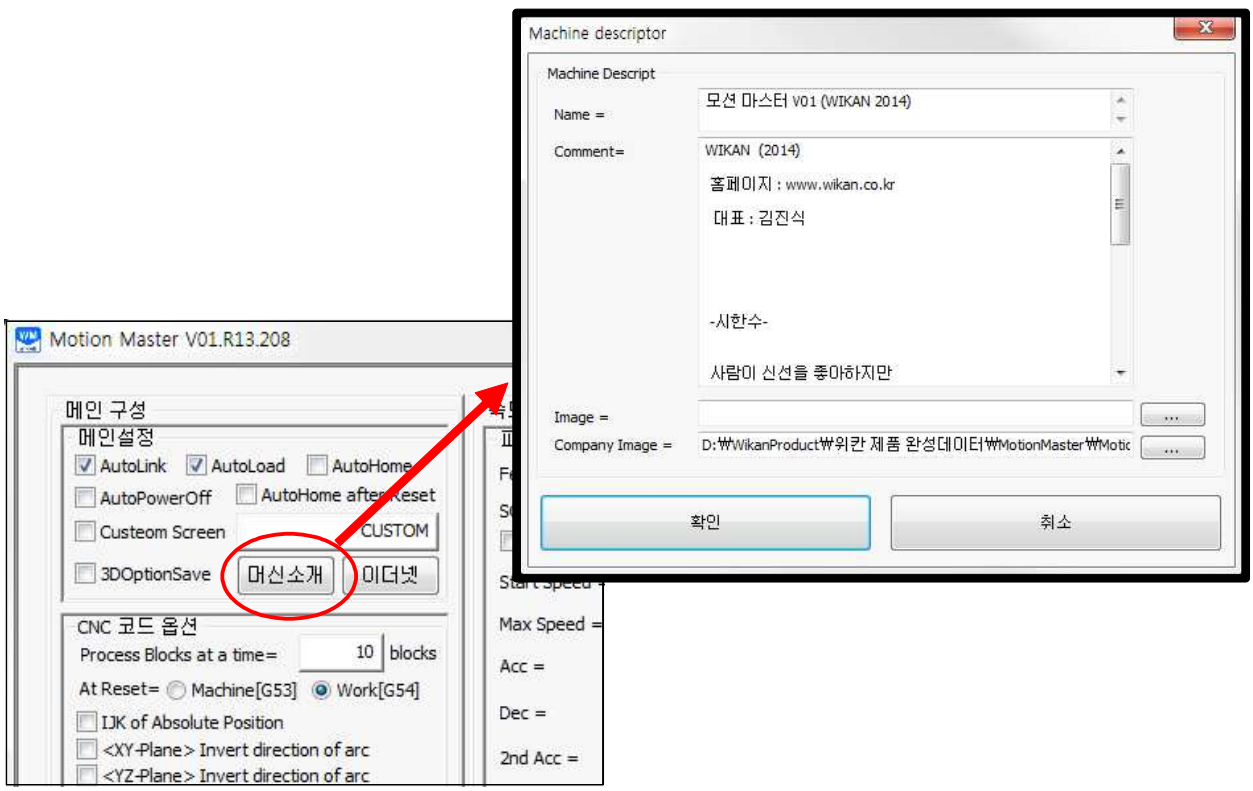
(1) 안내표시
(2) 동작카운터 관리
(3) 3D 그래프
(4) 시그날 램프
(5) 그래픽 옵션
(6) 진행률 표시
(7) 사용자 기능키
(8) 동작탭 화면전환

8.6.1. 안내표시

안내표시화면을 클릭하면 아래와 같이 사용자에 의해서 변경가능한 안내창이 나타납니다.

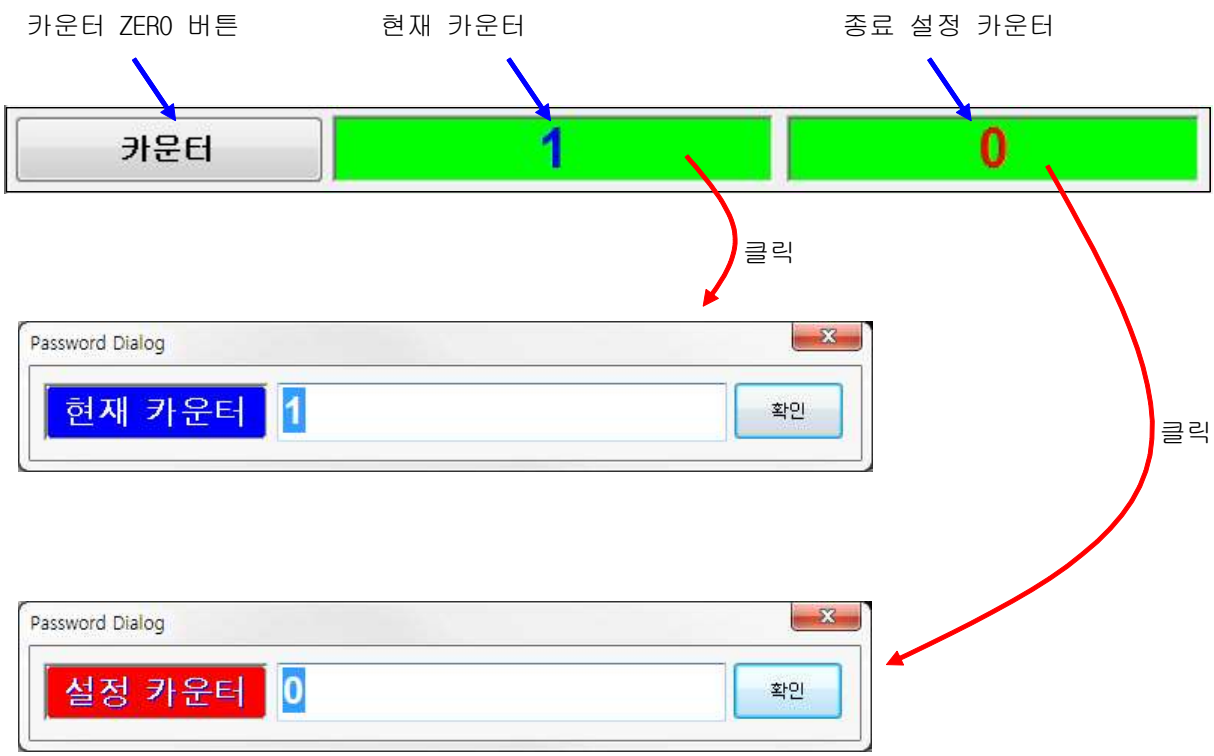


이러한 안내창은 “CNC구성화면”의 메인설정그룹의 [머신소개]버튼을 눌러 그림파일과 안내내용문구 등을 수정할 수 있습니다.



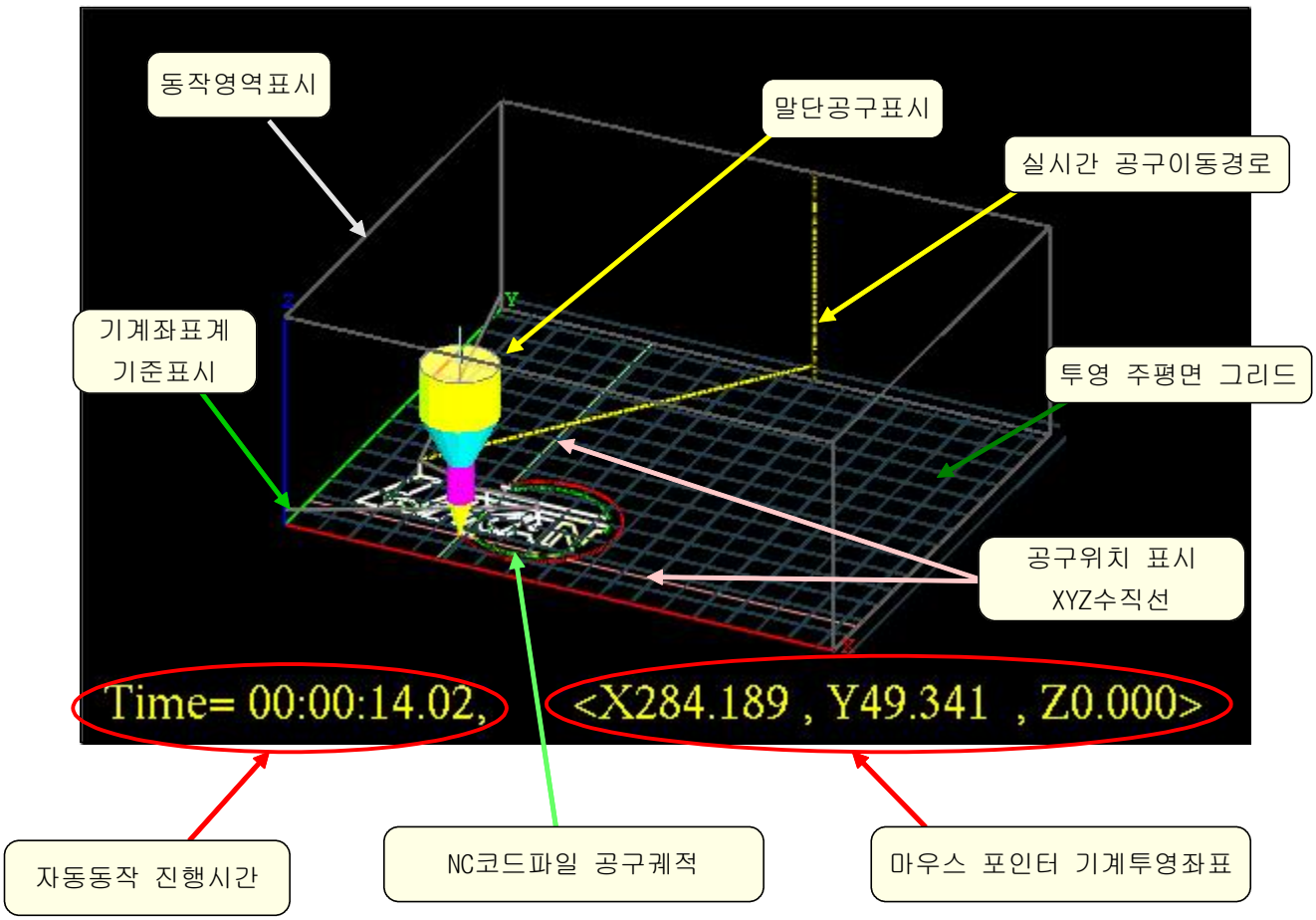
8.6.2. 자동카운터 관리

자동카운터값는 자동NC코드 실행시 1증가 됩니다. 설정값에 도달하면 [시작]버튼을 눌러도 실행되지 않습니다.
설정값이 0이면 자동카운터값에 관계없이 항상 실행가능합니다.
카운터 버튼을 클릭하면 카운터값이 0으로 클리어되며 각 카운터값 표시를 클릭하면 카운터값의 설정창이 나타납니다.



8.6.3. 3D 그래프

3D 그래프화면은 NC코드파일의 공구궤적을 도면처럼 표시해주는 역할 이외에도 기계실제 좌표와 프로그램좌표계간의 일치성을 간단하게 확인하고, 동작영역 및 진행시간 등 추가적인 정보들도 표현해주어 보다 안전하게 기계를 운영 할 수 있도록 도와줍니다.



8.6.4. 시그날(신호) 램프

시그날램프는 모션마스터프로그램의 아래의 3가지 동작상태를 표시합니다.

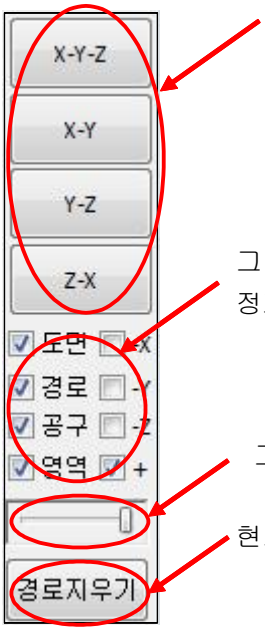
(1) 알람상태
- 알람 발생시 점멸합니다.

(2) 모션동작 상태
- 모션동작 발생시 점등합니다.

(3) 통신연결 상태
- 모션보드와 정상적으로 통신이 연결되면 점등하며 연결되지 않으면 점멸합니다.

8.6.5. 그래픽 옵션

3D 그래프를 표현하는 옵션을 포함합니다.



3D 그래프 투영방식 선택

그래픽표현 요소들의 그리기ON/OFF설정과 X,Y,Z,축 반전표현 설정

그래프선 굵기 조절

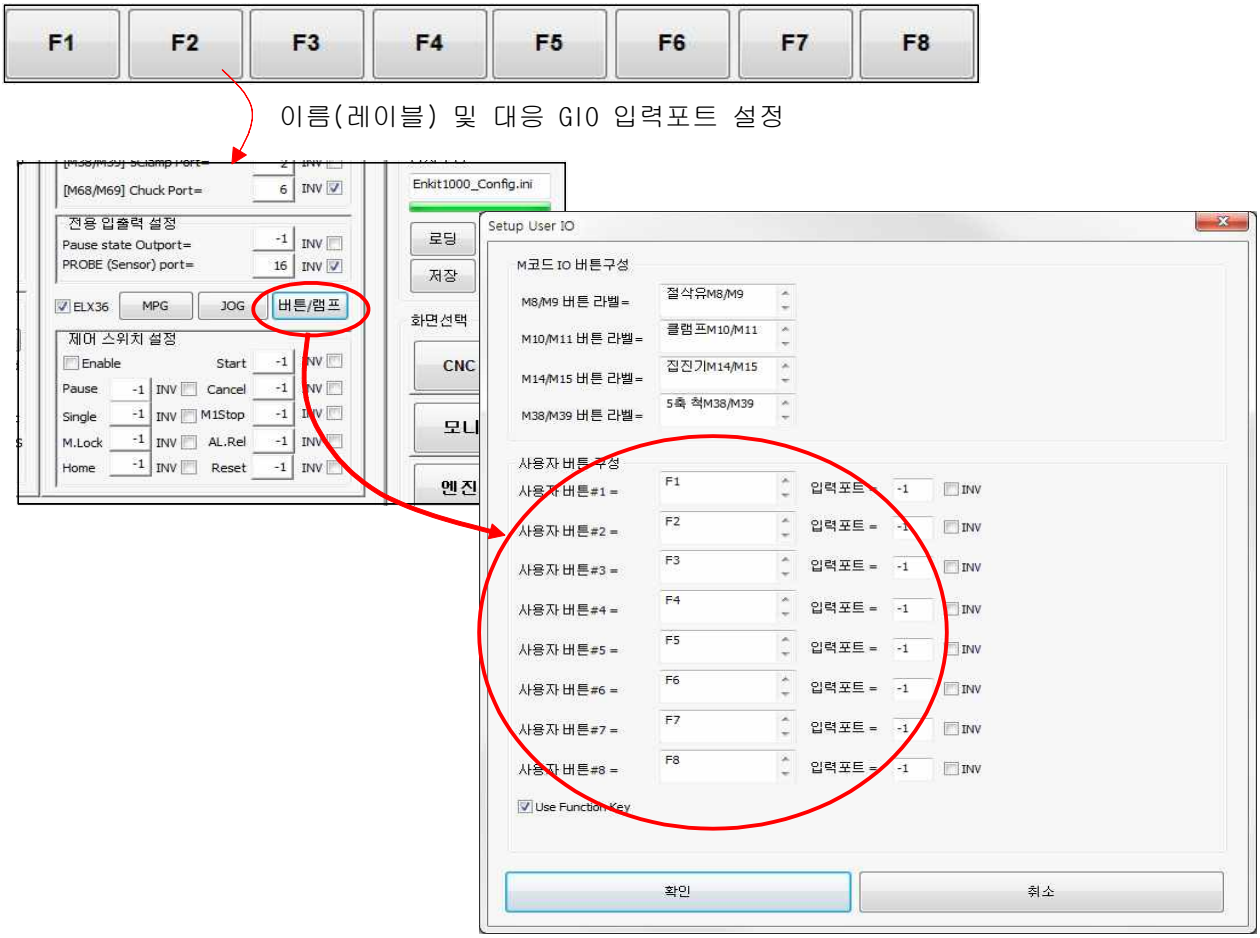
현재 경로표시 지우기(노랑색)

8.6.6. 진행률 표시

전체 NC명령 블록에 대한 현재까지의 명령블록실행을 %로 표시합니다.
WHILE이과 GOTO문 등의 매크로확장문법이나 서브루틴호출에 의한 명령블록 점프시에 %표시는 바르지 않을 수 있습니다.

8.6.7. 사용자 기능키

사용자 기능키는 V스크립터 언어를 통해서 사용자가 각 펑션키의 버튼기능을 정의할 수 있습니다.
또한 "CNC구성화면"의 [버튼/램프] 설정버튼을 클릭하면 사용자 IO설정화면이 나타납니다. 여기에서 펑션키의 이름과 입력포트번호 등을 기입하여 외부스위치에 의한 펑션키 동작을 구현할 수도 있습니다.



해당하는 펑션키 입력이 발생하면 실행폴더의 하위폴더 ScriptCodeW폴더 내에 있는
UserKey_Fx.code 와 UserKey_Fx_UI.code 의 스크립트파일을 자동으로 불러와서 실행하게 됩니다.
UI가 붙은 스크립터파일은 화면의 변화와 같이 모션동작에 직접적인 영향을 주지 않는 동작의 코드를 구성할 때 주로 사용될 수 있습니다.
아래 간단히 UserKey_F1_UI.code (F1키가 눌러질때 실행되는 스크립터코드) 와 UserKey_F8_UI.code (F8키가 눌러질때 실행되는 스크립트코드) 파일의 내용을 보여줍니다.

UserKey_F1_UI.code

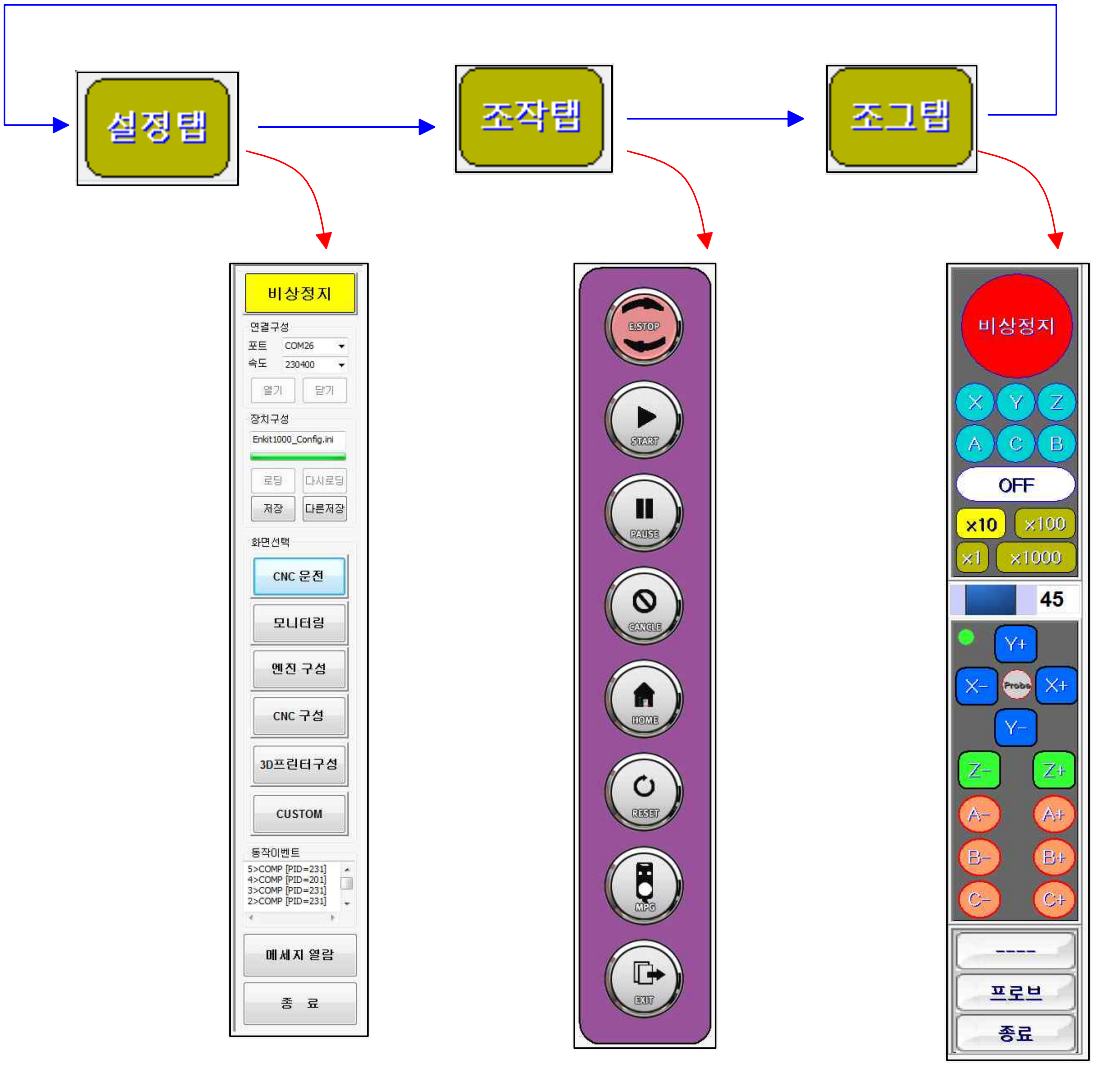
```
//0번 자동프로그램 테이블 로딩
WINC_LOAD_FILE_FROM_TABLE(0);
return (0);
```

UserKey_F8_UI.code

```
// 면치기 위저드실행
call("WWWizardCodeWWWizard_FaceCutWWFaceCut_UI.code");
```

8.6.8. 동작탭 화면전환

동작탭 화면전환버튼을 눌러주면 각각 [설정탭]화면 [조작탭]화면 [조그탭]화면으로 순서적으로 전환됩니다.



만약 이들 중 특정 동작탭화면만을 사용한다면 ScriptCodeW 폴더내의 UserKey_OpTab_UI.code 라는 스크립트파일을 메모장으로 열어 내부 코드를 수정해 주어야 합니다. 아래는 UserKey_OpTab_UI.code 파일 내용 일부를 표시합니다.

```
VAR TABMODE:#37001;
VAR TapOpMode;

//탭화면 종류선택
TapOpMode = 1;

// 기본탭, 실행탭, 조그탭 모두 사용
if( TapOpMode == 1 )
{
        if( TABMODE == 0 )              { return (1); }
        else if( TABMODE == 1 )         { return (2); }
        else                            { return (0); }
}
```

1, 2, 3 값 중 다른 하나를 선택하면 서로 다른 동작탭화면 전환을 확인할 수 있습니다.

항상 return (0); 을 하면 설정탭화면으로만 전환됩니다.

```
VAR TABMODE:#37001;
VAR TapOpMode;

//설정탭화면으로만 전환할 때
return (0);
```

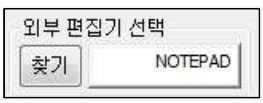
8.7. NC코드파일 관리화면

<NC코드 파일관리 버튼>

● [열기] : 클릭시 파일선택창이 나타나며 NC코드파일을 선택할 수 있습니다.

● [다시읽기] : 공구를 등을 옮기고 원점위치를 바꾼 후 최동 다시 RELOAD를 하여 도면의 실제위치를 확인합니다.

※ OPEN 후에 공작물좌표계등을 변경하면 도면상의 위치와 실제 툴의 이동경로과 3D 그래프화면상에 서로 다르게 표현될 수 있습니다.

● [편집] : 외부 에디터프로그램을 실행하여 현재파일을 수정할 수 있습니다. 수정을 했다면 [다시읽기]버튼을 눌러 현재파일을 다시 읽도록 합니다. 외부 에디터 프로그램은 CNC구성 화면의 

에서 선택할 수 있습니다.

● [코드열람]: NC코드들에 간단한 도움말 파일이 오픈됩니다.

● [STL뷰어] : 3D 프린팅모드에서 STL파일(ASCII/BINARY 모두)을 간단히 표현해주는 기능을 합니다.

STL파일을 열면 NC코드파일의 그래프가 사라지게 되므로 NC코드파일의 그래프를 다시 표현하려면 RELOAD하여 줍니다.(그래프는 STL파일과 NC코드파일 중 하나만을 표현할 수 있습니다)

8.8. 메인 조작화면

메인조작화면은 크게 아래 3가지 운전방식으로 CNC모션동작을 제어하게 됩니다.

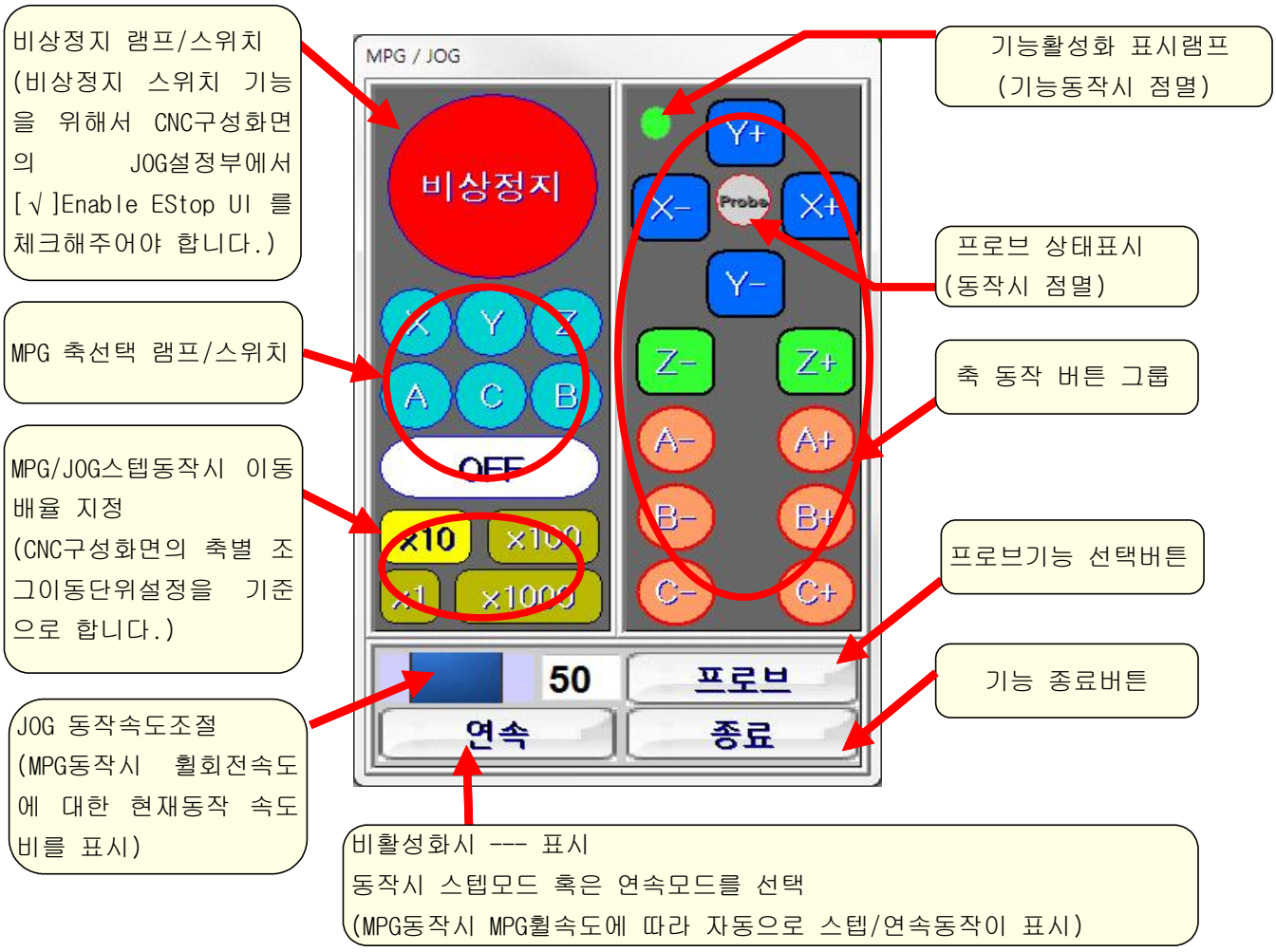
(1) JOG 운전방식

(2) MDI 운전방식

(3) 자동운전방식

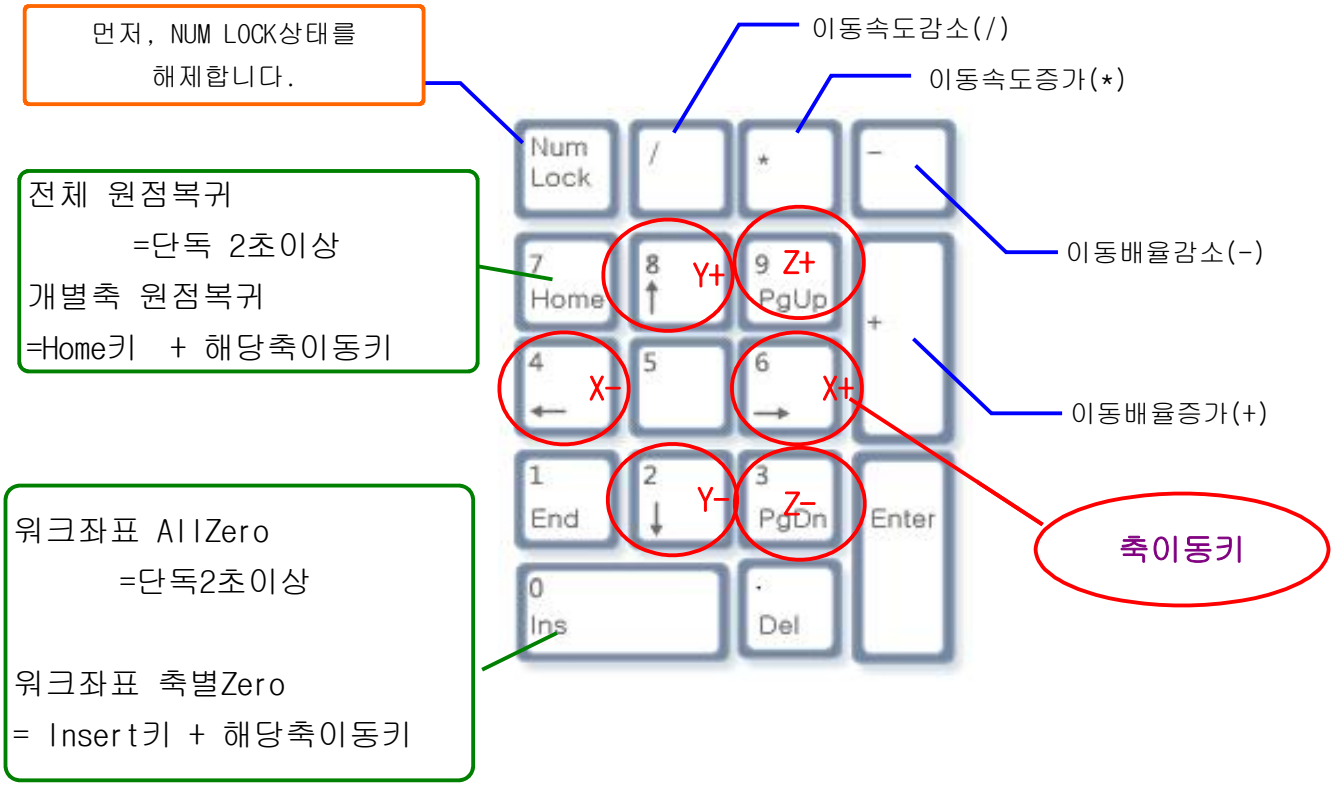
8.8.1. JOG 운전방식

JOG 버튼을 누르면 조그키나 MPG(Manual Pulse Generator:수동펄스발생기) 혹은 조이스틱과 같은 수동방식으로 모션 제어할 수 있는 운전모드가 실행되며 아래와 같은 조그제어화면이 생성됩니다. CNC구성설정의 옵션에 따라 독립된 조그제어화면이 생성되지 않고 동작탭화면이 조그탭화면으로 전환되는 경우도 있습니다. 어떤 방식의 조그제어화면이더라도 대부분 아래와 비슷한 기능을 내포하고 있습니다.



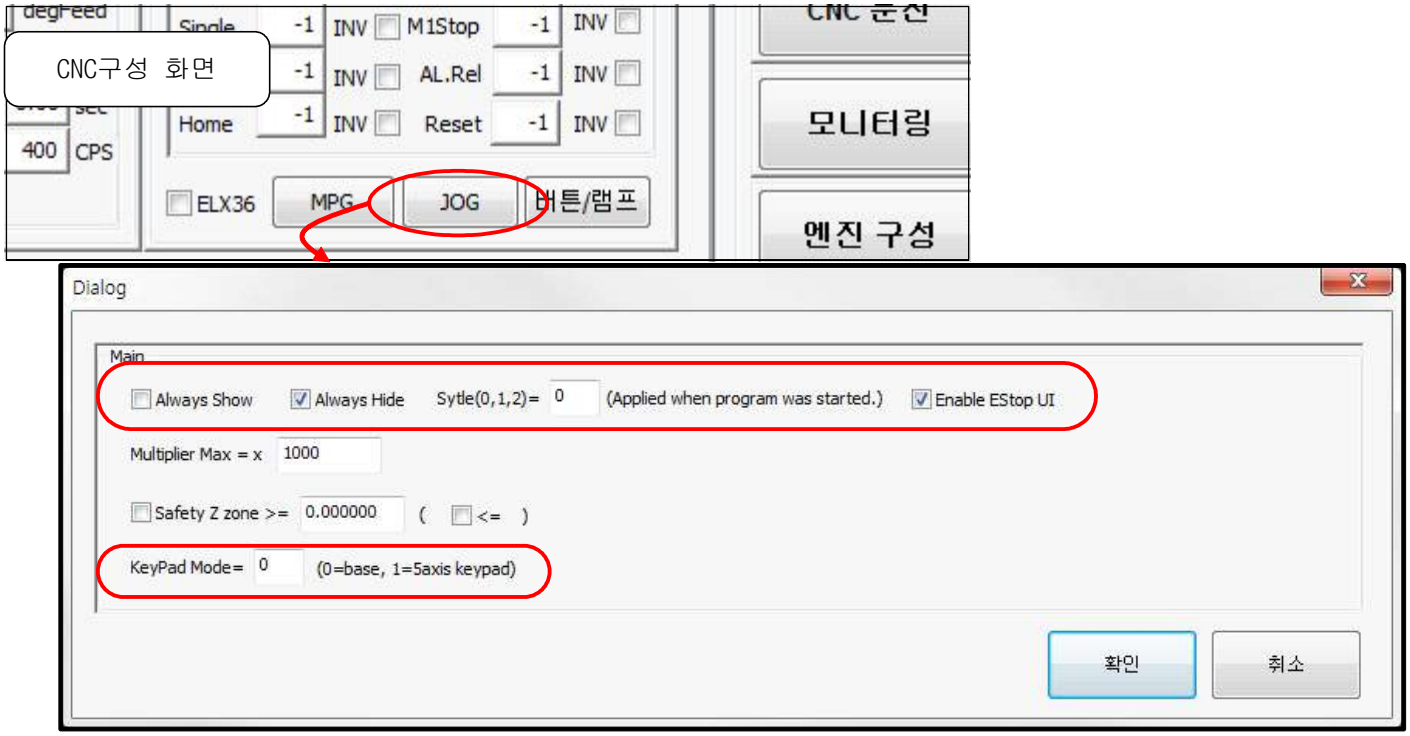
조그동작은 [연속모드]와 [스텝모드]로 작동될 수 있습니다. 연속모드는 마우스 혹은 키보드의 눌러짐에 의해 움직이고 떨어짐에 의해 정지되는 방식이고, 스텝모드는 키보드가 눌러질 때마다 선택된 이동배율만큼 단위로 이동하는 방식입니다.
마우스로 조작될 경우 해당 위치의 버튼을 클릭하는 것으로 실행되면 키보드를 사용할 경우 아래와 같이 특별한 키보드와 각 축별 혹은 버튼의 동작이 연결되어 있습니다.
키보드맵핑은 아래와 같은 일반적인 소형키패드장치로 조그동작을 구현할 수 있도록 구성되었습니다.

키패드의 조그동작기능은 아래와 같습니다.



이러한 키패드의 조그기능은 XYZ 3축 조그동작을 기준으로 합니다. 만약 4축 혹은 5축동작을 키패드를 통해 움직이기기 위해서는 “CNC구성화면”의 [JOG]버튼을 눌러 JOG설정창이 나타나면 KeyPad Mode 값을 1로 설정해 주면 5축까지 키패드를 이용해서 각 축의 이동동작을 실행할 수 있게 됩니다.
이때 Home키와 End키는 4축 동작으로 맵핑되고, Ins키와 Del키는 5축동작으로 맵핑됩니다.

조그제어화면의 생성방식과 키패드 기능맵핑등의 조그옵션은 "CNC구성화면“의 [JOG]설정버튼을 통해 JOG설정창 내에서 변경될 수 있습니다.

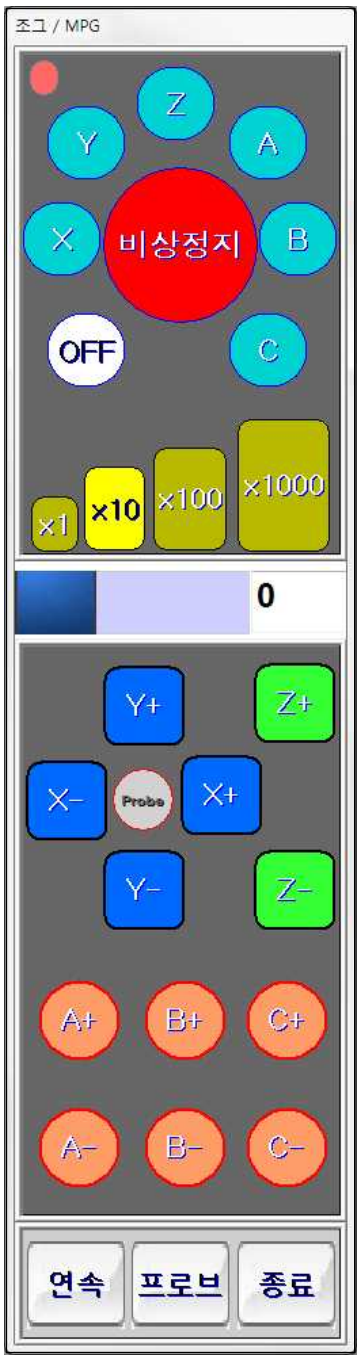


- [ ]Always Show 속성은 모션마스터프로그램의 실행시 항상 조그제어화면이 독립된 창으로 보여지는 속성입니다.
- [ ]Always Hide 속성은 조그제어화면이 동작탭화면의 전환으로만 나타나고 독립된 창으로 보여지지 않도록 설정하는 속성입니다.

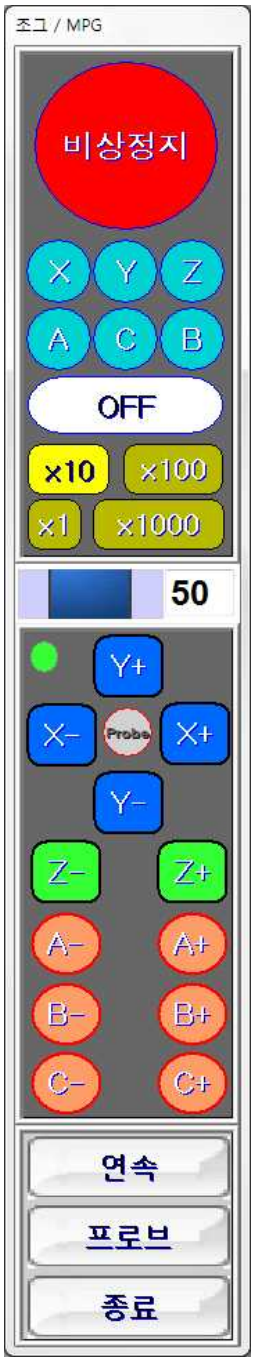
두가지 속성 모두를 설정하지 않으면 기본적으로 CNC운전화면의 [JOG]버튼을 눌를 때만 독립된 조그제어화면이 나타납니다.

- Style 속성에 따라 독립 조그제어화면의 모양이 달라집니다. 다음 그림은 각각 Style=0, 1, 2에 대한 조그제어화면을 보여줍니다. 화면의 해상도에 맞게 선택하실 수 있습니다. Style 값이 변화되면 프로그램이 종료된 후 다시 시작해야만 변경된 설정이 적용됩니다.

Style = 0
조그제어화면



Style = 1
조그제어화면



Style = 2
조그제어화면

- Multiplier Max 옵션은 스텝모드 혹은 MPG 제어시에 사용되는 최대 이동배율의 한계를 지정합니다.

- [ ]Safety Z Zone 옵션은 Z축이 특정 위치보다 작거나, 혹은 크게 될 때 다른 축의 이동을 금지시키고 메시지를 발생시키는 역할을 합니다.

- KeyPad Mode 설정은 KeyPad의 기능이 XYZ 3축기계용인지 5축 기계용인지를 지정하게 됩니다. 0이면 3축 기능으로 맵핑되고 1이면 5축 기능으로 맵핑됩니다.

8.8.2. MDI 운전

MDI (Manual Direct Input)는 수동으로 명령을 직접입력하여 운전하는 방식입니다.

버튼을 누르면 입력창이 활성화 되며 키보드를 통해서 G코드 명령을 실행합니다.
MDI로 실행할 수 있는 명령은 한줄로 제한되어 있어 하나의 명령블록만 실행됩니다.

8.8.3. 자동 운전

NC코드파일이 로딩 되면 아래의 자동동작버튼에 의해 프로그램을 실행시킬 수 있습니다.

(1) 프로그램 시작/정지 기능

●

: NC코드 파일을 로드되었다면 [시작]버튼으로 프로그램을 실행합니다.

●

: 실행도중에 [정지]버튼을 누르면 실행이 감속정지 되고, 다시 [시작]버튼을 누르면 실행이 재개됩니다. [취소L]버튼을 누르면 모든 실행이 중지되고 프로그램실행을 완료시킵니다.
※ NC코드파일의 실행도중에 ESC키를 누르게 되면 [정지]기능이 작동됩니다.

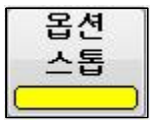
●

: 실행도중에 바로 [CANCEL]할 수 없으며 먼저 [PAUSE]버튼을 누른 후 [CANCEL]버튼을 사용 할 수 있습니다.

(2) 프로그램 실행 옵션

●

: NC코드를 실행하기 전에 [싱글블록]옵션을 클릭하여 램프가 켜지면 한 블록씩 실행되도록 합니다. 단, 실행도중에 [Single Block] 입력은 받지 않습니다.

● 

: 옵션널 스톱기능이 활성화 되면 NC코드 실행도중 M1 코드를 만나면 일시정지하게 됩니다. [시작] 버튼을 눌러주면 다음 블록부터 실행됩니다.

●

: NC코드를 실행하기 전에 [머신록]기능을 활성화 하면 기계는 움직이지 않으면서 내부적인 처리를 수행하는 시뮬레이션 기능을 갖습니다. 동작 완료 혹은 취소 후 [머신록]버튼을 다시 눌러주면 원래상태로 회복됩니다.

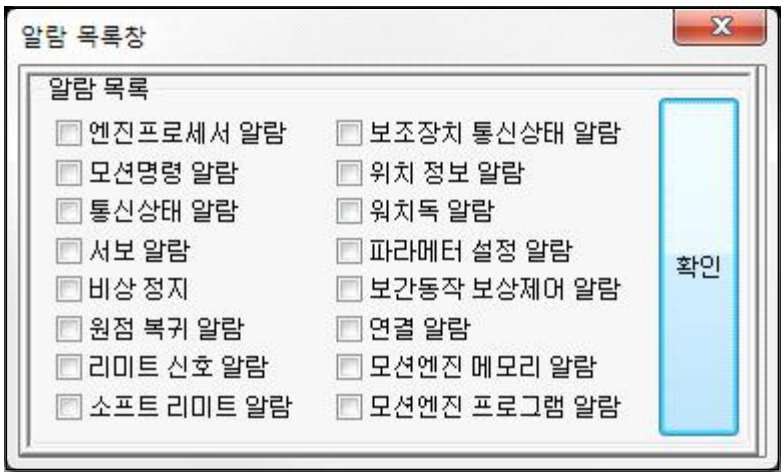
(3) 알람 및 리셋 관리 스위치

●

: 알람이 발생되었을 알람을 해제할 수 있는 기능입니다. 강제 알람 OFF 상태일 때 램프가 점멸합니다. 알람을 강제로 해제한 후 알람이 발생된 원인을 제거해 줍니다. 그리고 반드시 [알람해제]버튼을 눌러 원래상태로 복귀시켜야 합니다. 알람원인이 제거되지 않으면 원래상태 복귀시 알람이 다시 발생할 수 있습니다.

●

: 아래와 같은 현재 알람상태를 보여줍니다.

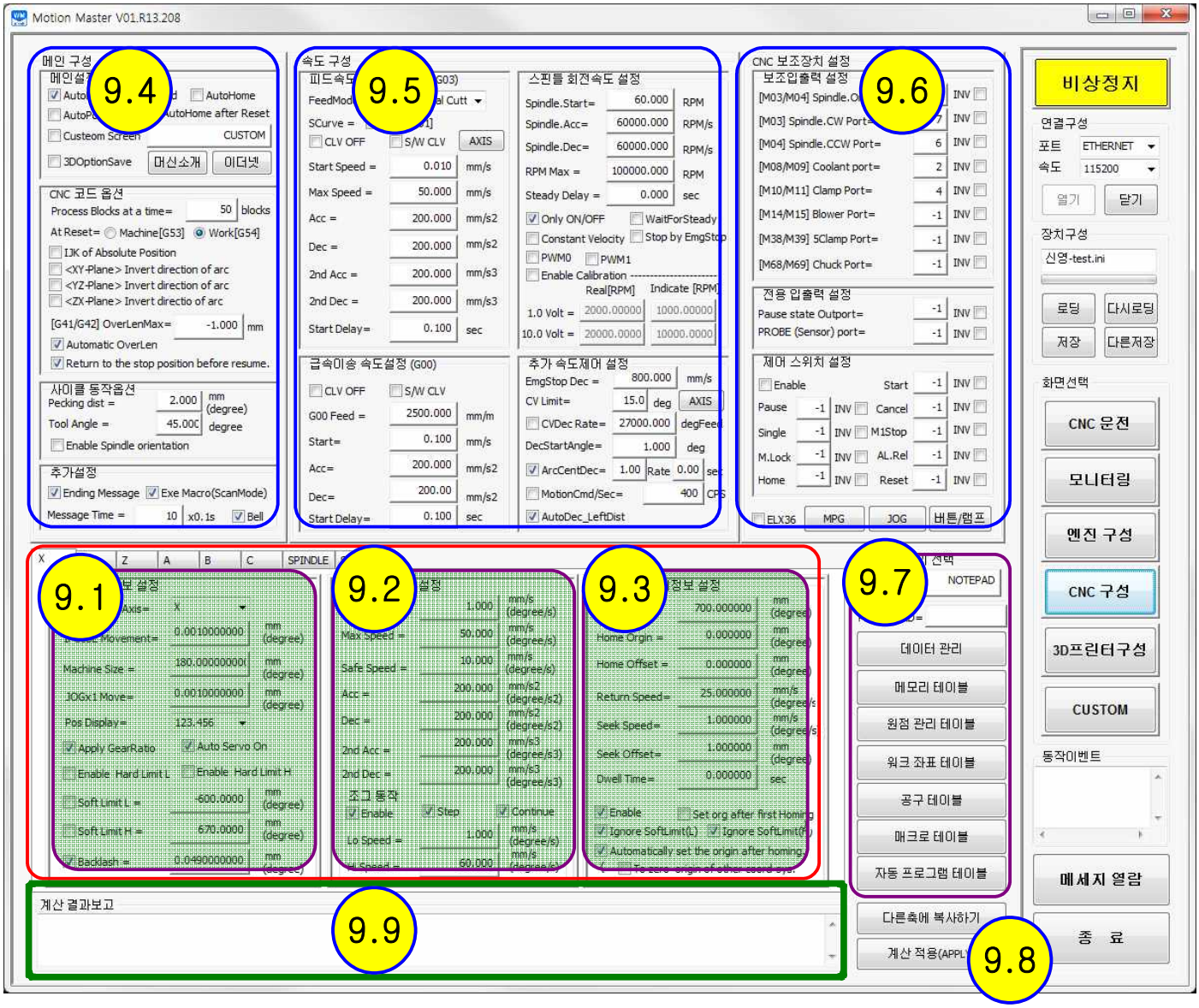


●

: 알람이 발생되었을 때 대부분은 [리셋]버튼을 눌러 회복합니다. AUTO 동작중일 때에는 [리셋]버튼을 누르면 현재 동작이 급정지 되면서 초기화됩니다.

9. CNC 구성화면

CNC 구성설정 화면은 CNC운용을 위한 기본적인 구성설정과 함께, 모션엔진의 실수체계지원을 위한 구성설정이 포함되어 있습니다.



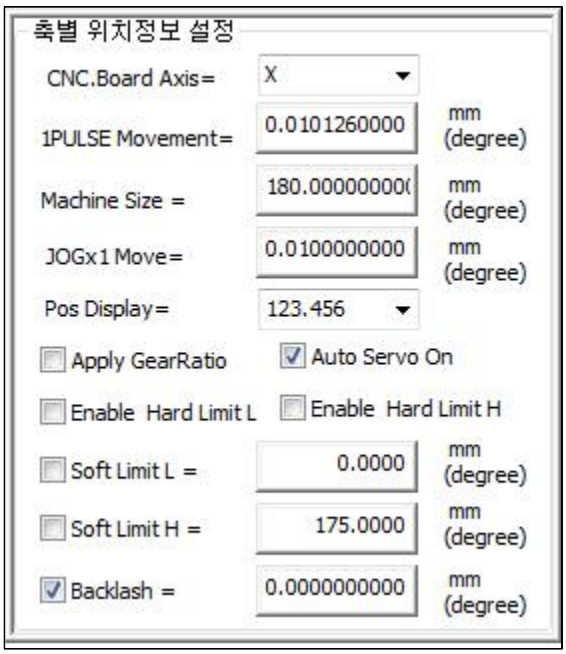
각 번호는 기본적인 설정순서입니다. 물론 임의적인 순서로 설정하여도 무관하지만 각 기능별 설명의 편의를 위해서 번호 순번대로 설명해 갑니다.

(9.1) 축별 위치정보 설정
(9.2) 축별 속도정보 설정
(9.3) 축별 원점복귀정보 설정
(9.4) CNC 메인구성 설정
(9.5) CNC 속도구성 설정
(9.6) CNC 보조장치 설정
(9.7) CNC 동작 테이블
(9.8) 기타설정
(9.9) 설정적용과 결과리포트

※모든 설정이 완료되면 동작탭 화면 장치구성의 [저장] 버튼을 눌러 설정내용이 구성파일에 기록되도록 합니다.

9.1. 축별 위치정보 설정

“(9.1.) 축별 위치정보 설정”과 “(9.2.) 축별 속도정보 설정”, 및 “(9.3.)의 축별 원점복귀정보 설정”은 상기 그림의 축선택 탭을 지정한 후 파라메터를 변경하면 해당하는 축에만 적용됩니다. 상기 축선택 탭에 지정된 축 이름은 모션보드의 축이름이 아니라 CNC동작 축이름이 됩니다.



● CNC.Board Axis : CNC축을 실제 모션보드의 동작축과 연결합니다.
스핀들 축을 제외한 임의축에 지정할 수 있습니다. Disable을 지정하면 해당축을 사용하지 않음을 의미합니다.

● 1Pulse Movement : 1펄스마다 움직이는 거리을 설정합니다. 모든 위치와 속도값의 기준이 되며, 단위계는 mm 혹은 degree(각도)입니다.

● CNC.Machine Size : 기계의 X축 최대 행정거리를 설정합니다.(사용하지 않음)

● CNC.JOGx1 Move : 조그 1단위 이동거리를 설정합니다.

● CNC.Pos Display : "CNC조작화면“의 위치좌표표시의 소수점자리수를 결정합니다.

● **Apply GearRatio : 모션엔진에서 전자기어비를 적용할 축인지를 결정합니다.**

※ CNC는 기본적으로 동작 축 모두에 전자기어비를 적용하는 것을 기준으로 합니다. 전자기어비를 적용하게 되면 1펄스 이동거리가 가장 작은 축을 기준으로 연산되며 나머지 축들은 계산된 특정비율로 연산됩니다. 기준이 되는 이 축을 연산주축이라고 부릅니다. 그러므로 축간의 1펄스당 이동거리비율이 크게 다를 경우 최대펄스 출력속도가 현저히 낮아 질 수 있습니다. 예를 들면 X축의 1펄스당 이동거리가 0.01mm 이지만 Y축의 1펄스당 이동거리가 0.001mm 이면 Y축이 연산의 주축이 되어 Y축이 최대 100KPPS로 동작할 수 있더라도 X축은 1/10인 최대 10KPSS까지만 동작할 수 있게 됩니다. 서보드라이브와 같은 지능적드라이브를 사용할 경우 드라이브내의 전자기어비 기능을 이용하는 것이 바람직하며 스텝모터와 같이 내부적으로 전자기어비 기능을 갖추고 있지 않을 때는 모션엔진의 전자기어비기능을 적용할 수 있습니다. 이때 각 축별 이동비율이 지나치게 달라지면 동작속도효율이 매우 낮아지게 됩니다. 그러므로 각 축간의 동작비율이 최대 1:2~1:3의 비율을 초과하지 않는 것이 바람직합니다.

만약 X,Y축에 비해 Z축이 지나치게 분해능이 높아서 전체적인 동작속도를 현저하게 낮추고 있다면 X,Y축만 전자기어비를 선택하고 나머지 Z축은 선택하지 않습니다. 이때 Z축은 다른 X,Y축과 함께 원호보간동작을 실행할 수 없게 됩니다. 원호보간동작을 강제로 실행시키면 Z축과 나머지 X 혹은 Y축간의 원호보간은 전자기어비기능이 적용되지

않아 펄스비율에 따른 타원보간형태를 갖게 됩니다.

● Auto Servo On : 모션보드가 축별 서보ON신호를 갖는다면 자동으로 서보ON신호를 발생합니다.

● Enable HardLimit L : 하드웨어 하한 리미트스위치 동작을 설정합니다. 리미트신호가 검출되면 비상정지 후 알람이 발생됩니다.

● Enable HardLimit H : 하드웨어 상한 리미트스위치 동작을 설정합니다. 리미트신호가 검출되면 비상정지 후 알람이 발생됩니다.

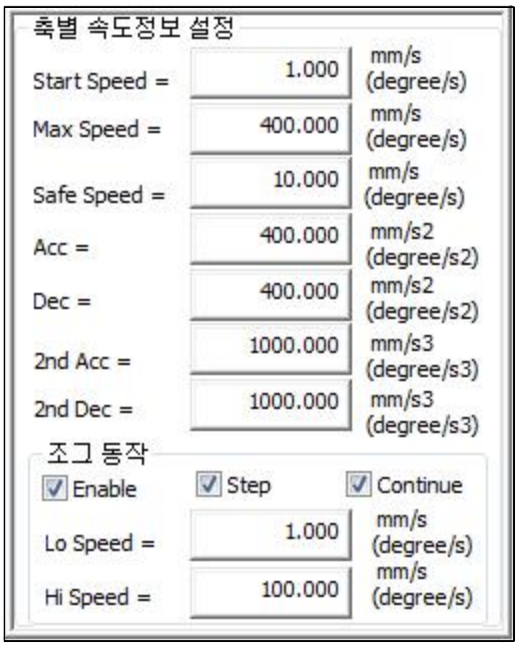
● SoftLimit.L : 소프트리미트 하한위치를 설정합니다. 기계위치로써 조그동작의 하한동작에도 영향을 미칩니다. 체크[√]를 설정하게 되면 소프트리미트 하한위치를 벗어나면 알람을 발생시킵니다.

● SoftLimit.H : 소프트리미트 상한위치를 설정합니다. 기계위치로써 조그동작의 상한동작에도 영향을 미칩니다. 체크[√]를 설정하게 되면 소프트리미트 상한위치를 벗어나면 알람을 발생시킵니다.

● Backlash : 백래쉬 동작시 이동거리를 설정합니다. 체크[√]설정을 하여야 동작합니다.

9.2. 축별 속도 파라메터 설정

축별 속도파라메터는 축동작속도를 제한하며 조그동작과 원점복귀 동작에 직접적으로 적용되는 파라메터입니다.



● Start Speed : 모션동작의 시작속도입니다. 0 에 가까울 수로 모션동작의 연결속도가 느려질 수 있습니다. 스텝모터의 경우 탈조하지 않는 범위내에서 설정하는 것이 바람직합니다. 단위는 mm/sec 혹은 degree/sec입니다.

※ 축별시작속도가 너무 낮으면 해당축의 MPG동작반응이 낮아질 수 있습니다.

● Max Speed : 모션동작의 최대속도를 설정합니다. (내부적으로 속도배율값을 결정하기 위한 기준값입니다.)

● Acc : 모션동작의 1차 가속도값을 설정합니다.

● Dec : 모션동작의 1차 감속도값을 설정합니다.

● 2nd Acc : 모션동작의 2차 가속도값을 설정합니다.

● 2nd Dec : 모션동작의 2차 감속도값을 설정합니다.

-조그 동작-

● Enable : 조그동작을 가능하도록 설정합니다.

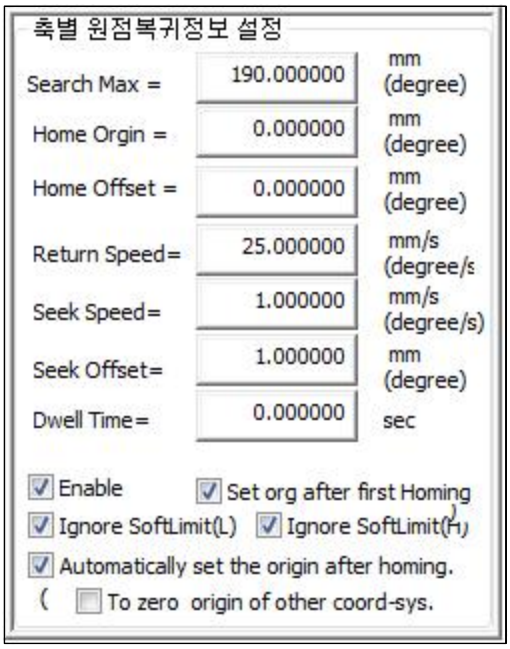
● Step : 조그동작 스텝모드를 가능하도록 설정합니다.

● Continue : 조그동작 연속모드를 가능하도록 설정합니다.

● CNC.JOG> Lo.Speed : 조그동작의 최저속도를 설정합니다.

● CNC.JOG> Hi.Speed : 조그동작의 최대속도를 설정합니다.

9.3. 축별 원점복귀 파라메터 설정



●Search Max : 원점검색을 위한 최대 이동거리를 설정합니다. 원점복귀 명령동작거리가 이 거리를 벗어나면 원점검색이 실패한 것이 되며 알람이 발생됩니다.

● Home Orgin : 원점복귀가 완료 되었을 때 기계위치값을 설정합니다.

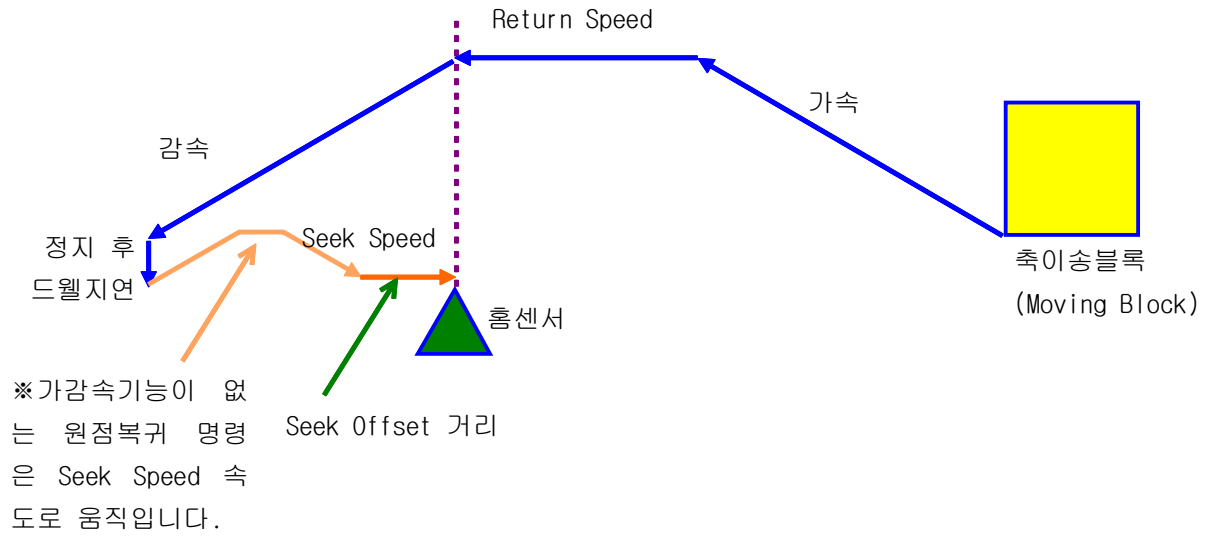
● Home Offset : 원점복귀 동작이 완료된 후 지정된 위치량만큼 이동합니다. (이 동작은 전체원점복귀 동작에서만 적용됩니다.)

● Return Speed : 원점복귀동작의 속도를 설정합니다. 이 속도를 너무 빠르면 홈센서를 검출하여 정지하는데 필요로 하는 거리가 길어집니다.

● Seek Speed : 홈센서를 검출정지한 후 역회전하여 홈센서를 재탐색하는 속도입니다. 이 속도가 작을수록 홈센서의 위치를 정확하게 찾는데 도움이 됩니다.

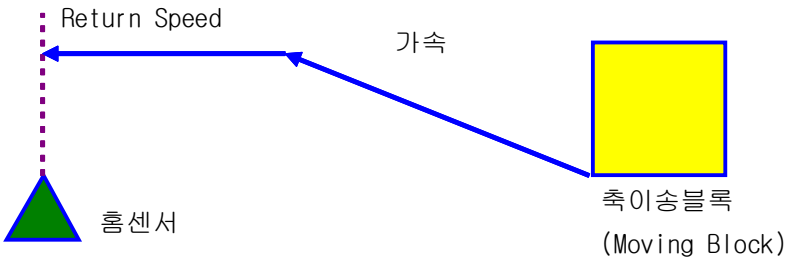
● Seek Offset : 모션엔진의 원점복귀 명령들 중에는 역회전 홈센서 재탐색시에 가감속을 수행하여 보다 빠르게 원점복귀를 수행하는 기능을 가질 수 있습니다. 이때 Seek Offset을 참조하여 이전에 검출된 홈센서의 위치를 기준으로 Seek Offset거리 이전에 가감속을 끝내고 Seek Speed로 홈센서를 재탐색하게 됩니다.

● Dwell Time : 홈센서(원점센서) 검출정지 후 일정시간(msec단위까지) 안정화 휴지기간을 갖도록 설정합니다.



※ 엔코더z상을 이용한 원점복귀 방식으로 지정하려면 “모션엔진 구성설정화면”의 [Home return Mode]설정값을 [Mode1]로 설정하면 됩니다.

※ “모션엔진 구성설정화면”의 [Home Return Mode]를 2로 설정하면 아래와 같이 단순하 원점복귀를 실행합니다.



● Enable : 원점복귀동작을 가능하도록 설정합니다.

● Set org after first homing : 첫번째 원점복귀 동작완료시 현재기계위치를 Home orgin 좌표값으로 설정합니다.

● Automatically set the origin after homing : 원점복귀 동작완효시 현재위치를 Home origin의 좌표값으로 설정

합니다.

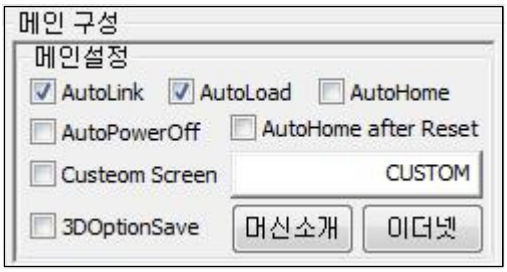
● (To zero origin of other coord-sys) : Automatically set the origin after homing 항목이 설정되어 있는 상태에서 원점복귀버튼에 의해 원점복귀가 실행된 후 공작물좌표계등의 다른 좌표계 기준위치값을 모두 zero로 설정하게 됩니다.

[다른축에 복사하기]버튼을 클릭하면 아래와 같은 복사 대상(Target)축 선택 다이얼로그가 나타납니다. 복사대상 축들을 선택하고 [확인]버튼을 누르면 현재선택축의 설정값들을 복사대상 축들의 설정값으로 복사합니다.

9.4. CNC 메인구성 설정

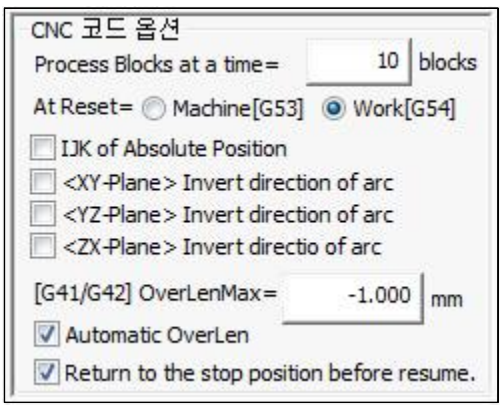
9.4.1. 메인 설정

프로그램 실행단계의 주요 옵션들을 설정합니다.



● AutoLink : 프로그램 실행시 가장 최근에 연결된 장치구성으로 자동연결을 시도합니다.
● AutoLoad : 프로그램 실행시 가장 최근에 오픈된 파일을 자동으로 로딩합니다.
● AutoHome : 프로그램 실행시 전체원점복귀 동작을 실행합니다.
● AutoPowerOff : 프로그램 종료시 PC전체시스템을 OFF합니다.
● AutoHome after Reset : 리셋 동작후 항상 원점복귀 동작을 실행합니다.
● Custom Screen [CUSTOM] : 프로그램 시작시 커스텀화면을 첫화면으로 지정합니다. 동작탭 화면상의 버튼이름을 임의로 지정할 수 있습니다.
● 3DOptionSave : 프로그램 실행시 가장 최근에 저장된 3D그래프 정보를 불러옵니다.

9.4.2. CNC코드옵션



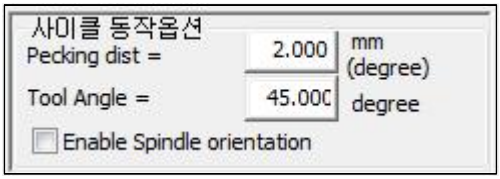
● Process blocks at a time : NC코드를 실행할 때 한번에 몇 블록씩 실행을 처리할 지를 결정합니다. 위칸 모션보드은 기본적으로 내부의 명령큐를 이용하여 실행합니다. 그러므로 NC코드들은 미리 읽혀져서 모션보드의 명령큐에 지령을 내리면 연속된 동작을 수행합니다. 이때 한번에 몇 블록씩 명령을 처리하여 명령큐에 지령을 내릴 것인지를 지정합니다.
● Reset=Machine[G53] / Reset=Work[G54] : 리셋동작시 기본적인 좌표계를 선택합니다. G53은 기계좌표계를 선택하는 것이며, G54는 공작물좌표계를 선택하는 것입니다. 일반적으로 G54가 선택되며 G54의 기준위치(Orientation) 값이 0,0,0일 경우 기계좌표값과 동일합니다.
● IJK of Absolute position : 체크[√]되면 I, J, K 코드값을 절대위치값으로 해석합니다. 체크되지 않으면 상대위치값으로 해석합니다.
● <XY-Plane>Invert direction of arc: xy평면에서의 원호보간 방향반전(CW<->CCW)
● <YZ-Plane>Invert direction of arc: yz평면에서의 원호보간 방향반전(CW<->CCW)
● <ZX-Plane>Invert direction of arc: zx평면에서의 원호보간 방향반전(CW<->CCW)

● [G41/G42] OverLenMax : 경보정에 의한 벱선제어 경로가 종점의 법센제어점보다 지정된 거리이상 벗어나면 자동으로 원호보간이 삽입됩니다. -1이면 이 동작을 사용하지 않습니다.

● Automatic OverLen : 내부적으로 자동으로 원호보간을 삽입하여 경보정에 의한 법선제어처리를 수행합니다.

● Return to the stop position before resume : 일시정지에 조그동작등에 의해 위치가 이동한 후 동작재개 할 때 처음 정지된 위치로 되돌아가서 동작재개할 수 있도록 설정합니다.

9.4.3. 사이클 동작옵션

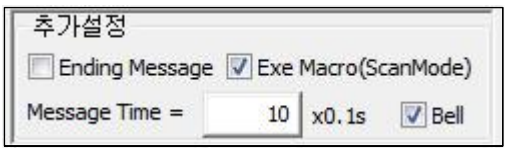


● Pecking dist : 펙킹사이클 동작에서 pecking 동작 이동거리를 설정합니다.

● Tool Angle : 정밀보링사이클에서 툴의 정위치각도와 반대방향으로 후퇴하는 동작을 위한 설정값입니다. 이 값은 일반적으로 S축의 Home Origin값과 동일하게 설정됩니다.

● Enable Spindle orientation : 정밀보링사이클에서 스핀들의 정위치정지동작(스핀들 원점복귀 동작)을 허용할지를 결정합니다.

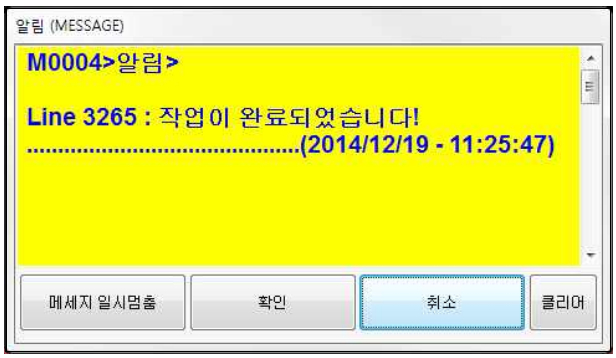
9.4.4. 추가설정



● Ending Message : 자동프로그램실행 종료후 종료메시지를 출력합니다.

● Exe Macro(ScanMode) : NC코드파일을 읽는과정(스캔과정)에서 V스크립터실행등을 금지합니다.

● Message Time : 특정한 상황에서 발생되는 아래의 메시지창이 보여지는 시간을 설정합니다. 0으로 설정되면 메시지창이 나타나지 않습니다.

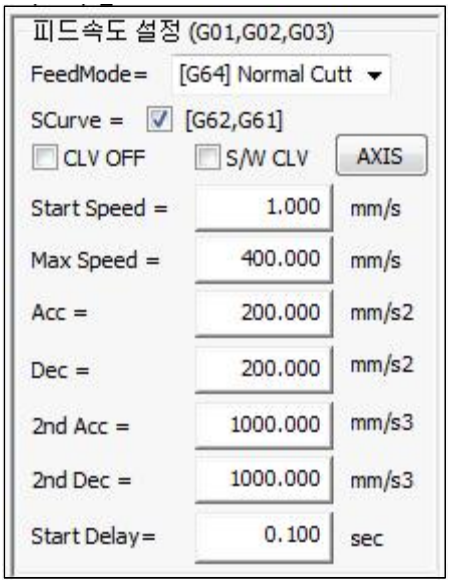


● Bell : 메시지창이 발생될 때 알림소리를 출력합니다.

9.5. CNC 속도구성 설정

9.5.1. 절삭피드 속도설정(G01,G02,G03)

보간동작(G01,G02,G03)의 모션동작에서 적용되는 속도 파라메터 설정그룹입니다.



● FeedMode : 일반적인 절삭은 [G64]Normal Cutting을 선택합니다.

[G63] Hard Cutting모드는 절삭시 가감속없는 정속이동방식입니다.

[G62] Soft Cutting모드는 S자 가감속에 의한 이동방식입니다.

[G61] Exact Stop모드는 각 모션마다 가감속하여 각 모션마다의 경유 종점위치가 보다 정확하게 이동시키는 모드입니다.

● SCurve : 특정모드에서 SCurve로 동작하는 것을 설정합니다.

※ 현재 모션마스터 R13.208 기준은 S가감속도제어를 부분적 지원하고 있습니다.

● CLV OFF : 선속도일정제어 방식을 사용하지 않음을 설정합니다.

● [AXIS] : 선속도제어가 적용될 축을 선택합니다.

● S/W CLV : 선속도일정제어 방식을 모션엔진에 의존하지 않고 모션마스터프로그램에 의한 소프트웨어명령으로 처리하도로 설정합니다.

● Start Speed : 초기 시작 및 종료속도입니다. 단위는 mm/sec 혹은 degree/sec입니다.

● Max Speed : 동작 최대속도입니다. 이 속도값의 변경은 F코드로 실행되는 최대속도값을 제한합니다.

● Acc : 1차 가속도값 입니다.

● Dec : 1차 감속도값입니다.

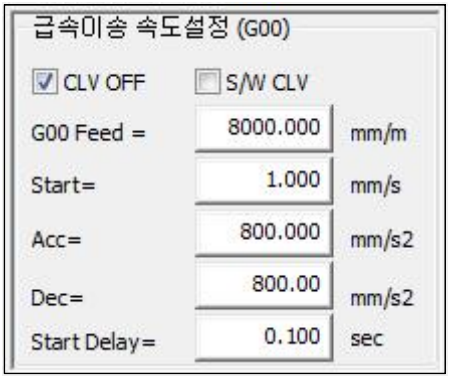
● 2nd Acc : 2차 가속도값 입니다.

● 2nd Dec : 2차 감속도값 입니다.

● Start Delay : 연속되지 않는 첫 절삭명령(G01,G02,G03)시작시에 지정된 시간값만큼 지연동작 합니다.

9.5.2. 급속이송 속도설정(G00)

G00코드의 급속이송동작에 대한 파라메터 설정그룹입니다.



● CLV OFF : 선속도일정제어 방식을 사용하지 않음을 설정합니다.

● S/W CLV : 선속도일정제어 방식을 모션엔진에 의존하지 않고 모션마스터프로그램에 의한 소프트웨어명령으로 처리하도로 설정합니다. S/W CLV : 선속도일정제어 방식을 모션엔진에 의존하지 않고 모션마스터프로그램에 의한 소프트웨어명령으로 처리하도로 설정합니다.

※ 일반적으로 급속이송은 선속도 일정방식을 사용하지 않거나, S/W CLV방식으로 선택합니다.

● G00 Feed : 급속이송동작 속도를 분당이송거리로 설정합니다. (주의! 초당이동거리와 구별해주세요.)

:각 모션보드의 최대동작속도 이내에서 안전한 범위내로 설정합니다.

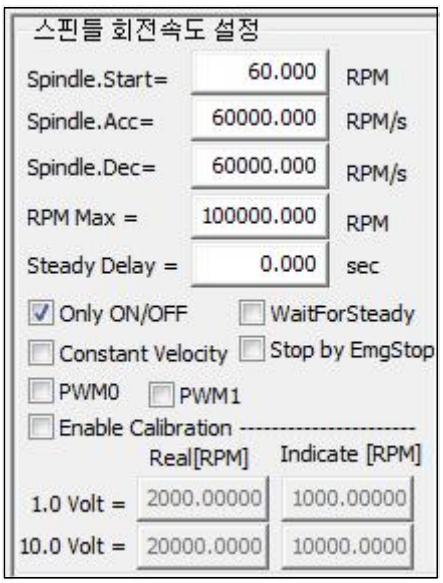
● Acc : 1차 가속도값 입니다.

● Dec : 1차 감속도값입니다.

● Start Delay : 연속되지 않는 첫 급속이송명령 시작시에 지정된 시간값만큼 지연동작합니다.

9.5.3. 스핀들 회전속도 설정

S코드와 M3/M4/M5의 스핀들 제어명령의 동작특성에 관련된 파라메터 그룹입니다.



● Spindle.Start : 스핀들 고속회전동작의 시작 RPM을 설정합니다.

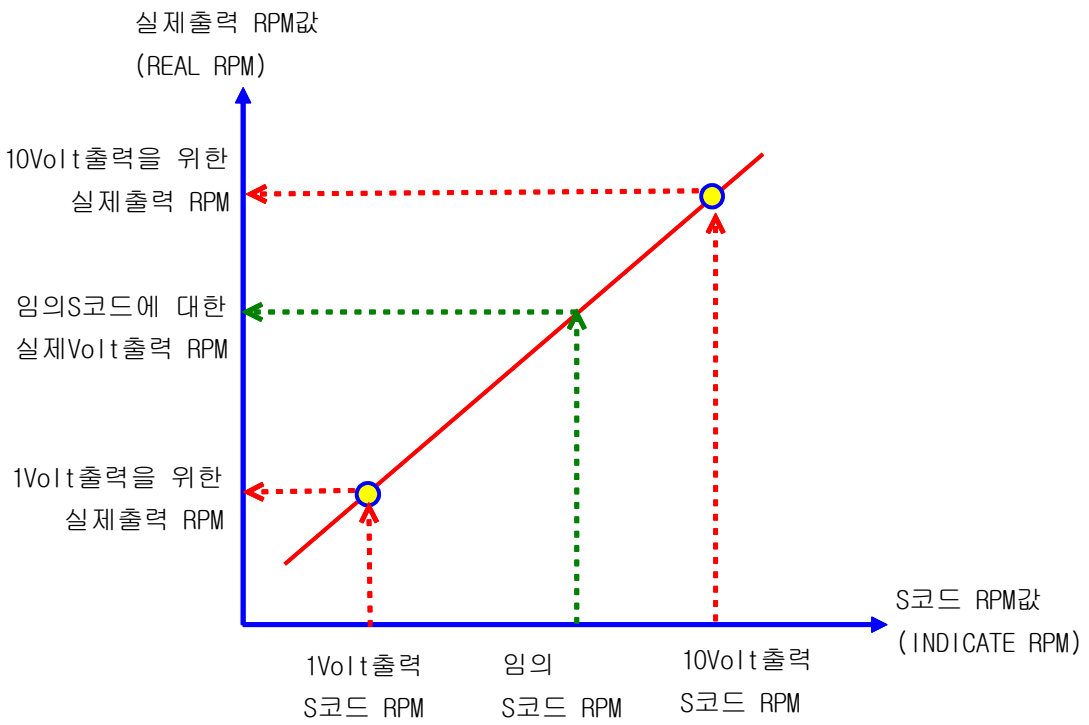
● Spindle.Acc : 스핀들 고속회전동작의 가속도 RPM을 설정합니다.

● Spindle.Dec : 스핀들 고속회전동작의 감속도 RPM을 설정합니다.

● RPM Max : 스핀들 회전 최대RPM을 설정합니다. S코드값의 최대 동작값을 지시합니다.

● Constant Velocity : 스핀들의 회전동작에 가감속이 없는 일정속도제어방식일 때 사용합니다. 스핀들축이 회전동작기기가 아닌, 레이저장비와 같은 즉각적인 출력장치이거나 탭동작과 같은 저속 동작일 때 유용합니다.
● WaitForSteady : 스핀들 속도변화 코드가 실행된 이후 스핀들이 정상속도에 도달할 때까지 대기동작을 실행합니다.
● Only ON/OFF : 스핀들이 단순 on/off 동작을 하는 장치일 때 사용됩니다. 이때 S코드 값에 의해 스핀들의 회전제어는 수행하지 않고, 단지 스핀들회전과 관련된 입출력(M03/M04/M05)동작만 수행하게 됩니다.
이 옵션이 설정되어있으면 다른 속도제어파라메터는 적용되지 않습니다.
● Stop by EmgStop : 비상정지시 스핀들도 함께 정지 할 것인지를 설정합니다.
● PWM0 : PWM 채널0 을 통해 속도제어를 출력합니다.
● PWM1 : PWM 채널1 을 통해 속도제어를 출력합니다.
※ PWMx를 선택하면 아래의 [√]Enable Calbration을 체크하여
Real[RPM]의 1.0V항목과 10.0V항목에 각 전압에 대한 PWM출력값을 기록합니다.
X6V3엔진에 기반한 모션컨트롤러는 기본적으로 20KHz 기반주파수에서 PWM동작 수행합니다.
모션보드마다 차이를 가질 수 있지만 10.0V 에 대한 PWM최대출력값은 100~200 사이에서 선택하는 것이 유리합니다.
10.0V 값이 100 일 때 1.0V 값은 10 으로 설정합니다.
10.0V 값이 200 일 때 1.0V 값은 20 으로 설정합니다.

● 스핀들 전압출력 교정> Enable : 모션보드에 따라 스핀들 축은 펄스와 함께 전압을 출력하는 기능을 갖고 있을 수 있습니다. 이때 전압출력은 펄스주파수를 전압크기로 변환시키는 FtoV(Frequence to Voltage)회로에 의존하므로 스핀들의 rpm과 매칭하고 싶은 실제 전압출력과의 선형관계가 발생합니다. “스핀들 전압출력 교정” 파라메터는 이러한 매핑관계를 설정합니다.Enable을 체크[√]하면 S코드에 의한 스핀들 RPM출력 주파수는 다음과 같이 교정되어 출력됩니다.



● 스핀들 전압출력 교정방법>
스핀들 전압출력은 주로 인버터장치와 같이 전압입력에 의해 속도를 조절하는 장치를 위해서 사용됩니다.
이때 출력전압에 대한 RPM값의 변환은 인버터 내부 파라메터에 의존합니다. 모션베이트키트의 모션트레이닝보드를 기준예로서 설명하겠습니다.

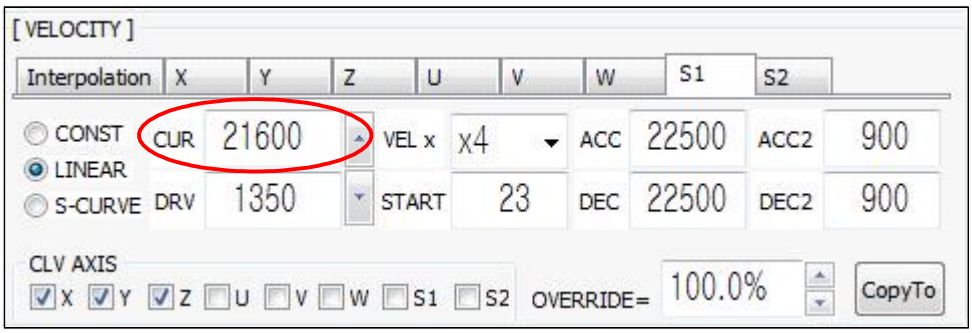
보드와 연결된 인버터의 동작이 1V 입력에서 360 RPM이 회전되고 10V에서 3600 RPM이 회전되도록 구성되었다고 가정합니다. 그러면 NC코드 프로그램에서는 S360 이라고 지정하면 스핀들 회전시 1V가 출력되고, S3600이라고 지정하면 스핀들 회전시 10V가 출력되면 됩니다.

X6V3엔진의 회전제어 펄스 주파수는 0~2Mhz(MPPS)까지 출력하지만 내부의 FtoV(주파수-전압)변환 회로는 특정주파수 범위내에서만 작동합니다. 모션트레이닝보드의 경우 0~20kHz(kPPS) 범위내에서 사용되어야 합니다.

모션엔진은 스핀들이 펄스에 의해 동작한다고 간주하므로 결국 펄스를 발생시킬 것이며 이 값이 20kHz를 크게 넘지 않도록 스핀들축의 파라메터를 설정합니다. 단위는 (Degree)각도로 가정합니다.

-개별축 위치 파라메터 설정> 1Pulse Movement = 1 도
-전체속도 파라메터 설정> Spindle.Start = 60 RPM (=360도/초 = 360Hz)
-전체속도 파라메터 설정> Spindle.Acc = 60000 RPM
-전체속도 파라메터 설정> Spindle.Dec = 60000 RPM
-CNC 주요 파라메터 설정> Spindle.RPM Max = 3600 RPM (=21,600도/초 = 21.6kHz)

위와 같이 설정한 후 “CNC조작 화면”의 MDI창에 S3600을 입력하고 [ENTER]버튼을 누릅니다. 그리고 스핀들제어창에서 [CW]버튼을 클릭하면 스핀들 축이 회전펄스를 출력하게 됩니다. 이때 출력주파수는 “모션엔진 모니터링”창의 “속도모니터링”창에서 확인할 수 있습니다.



여기까지가 모션보드의 전압출력을 사용하기 위한 기본적인 설정입니다.
이제 이 값을 기준으로 교정값을 설정해봅니다.

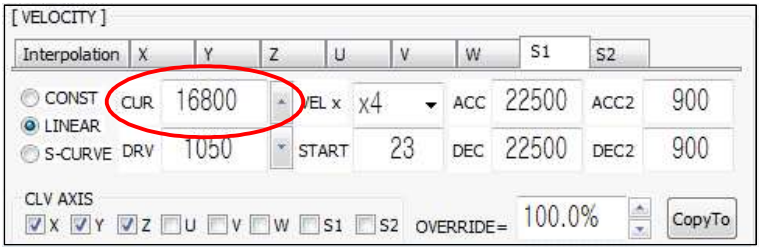
교정하기 위해서 스핀들축의 전압출력을 확인할 수 있는 연결구성을 만듭니다.
인버터와 연결되어 있다면 인버터의 현재 RPM 출력값을 그대로 읽을 수 있으며, 인버터와 연결되지 않았다면 모션보드의 스핀들 전압출력 포트를 전압측정계로 연결하여야 합니다.

(1) 먼저 인버터와 연결되었다고 가정하고 인버터를 통해서 RPM을 읽어서 설정하는 방법입니다.

- MDI창에 S360를 입력실행하고 [CW]회전시킨 후 인버터의 회전속도를 RPM을 읽어봅니다.
읽혀진 RPM값을 220 RPM 이라고 가정합니다.
- MDI창에 S3600를 입력실행하고 [CW]회전시킨 후 인버터의 회전속도를 RPM을 읽어봅니다.
읽혀진 RPM값을 2800 RPM 이라고 가정합니다.
※여기서는 교정값 1.0Volt와 10.0Volt의 기준값 의미를 두지 않아도 됩니다.
그러면 아래와 같이 스핀들전압출력 교정파라메터를 입력합니다.

[√]ENABLE	REAL [RPM]	INDICATE[RPM]
1.0 Volt	220	360
10.0 Volt	2800	3600

이와 같이 설정한 후 스핀들을 S3600 으로 회전하면 실제 주파수는 16.8kHz 로 출력됩니다.
(실제로 2800rpm 으로 출력해야 하므로 2800/60초*360도/sec = 16.8kHz입니다.)



(2) 인버터의 연결이 없다면 전압측정기로 기준전압 1.0V와 10.0V로 기준점을 만들어 교정할 수 있습니다.
이때 스핀들 전압출력 포트에 전압측정기를 먼저 연결합니다.
그리고 S코드로 사용될 기준값을 S100 이면 1Volt출력, S1000 이면 10Volt출력 하는 것으로 설정합니다.
- MDI창의 S100을 입력하고 [CW]스위치로 스핀들을 회전시킨 후 전압출력을 측정합니다.
- 전압이 1Volt를 지시하지 않는다면 S값을 1Volt출력에 맞게 조절해 봅니다.
- 1Volt를 지시한 S코드값을 기록합니다. 이 값을 220 이라고 가정합니다.
- 전압이 10Volt를 지시하도록 S값을 증가시킵니다.
- 10Volt를 지시한 S코드값을 기록합니다. 이 값을 2800 이라고 가정합니다.

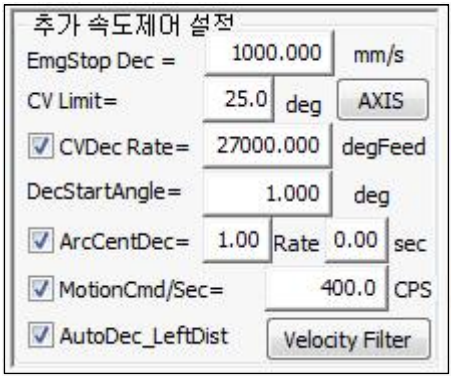
그러면 아래와 같은 스핀들전압출력 교정파라메터값을 설정하면 됩니다.

[√]ENABLE	REAL [RPM]	INDICATE[RPM]
1.0 Volt	220	100
10.0 Volt	2800	1000

이와 같이 설정한 후 스핀들을 S1000 으로 회전하면 실제 주파수는 (1)의 결과와 같은 16.8kHz로 출력됩니다.

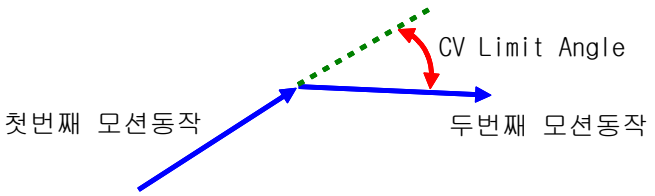
9.5.4. 추가 속도제어 설정

절삭속도의 자동가감속제어와 비상정지감속도 등의 추가적인 속도제어 파라메터를 설정하는 그룹입니다.

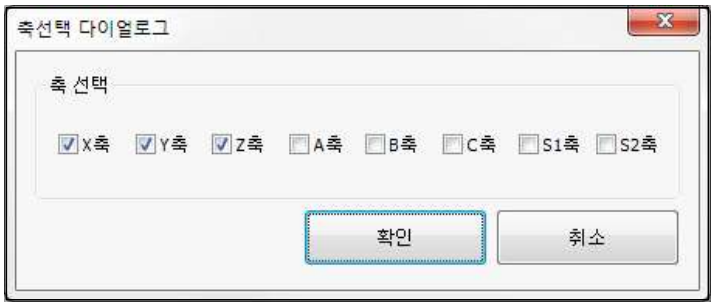


● EmgStop Dec : 비상정지시 1차 감속계수를 지정합니다. 이 값이 0 이면 비상정지시 급정지(펄스를 즉각 차단)하게 됩니다. 소형장비에서 일반적인 동작이지만, 동작속도가 큰 장비의 경우 급정지는 기계에 큰 충격을 줄 수 있습니다. 이때 기계에 맞는 1차 감속계수를 지정하면 지정된 감속도 정지할 수 있습니다.

● CV Limit Angle : 속도의 연속성을 유지하기 위한 이동각도의 최대값을 설정합니다.

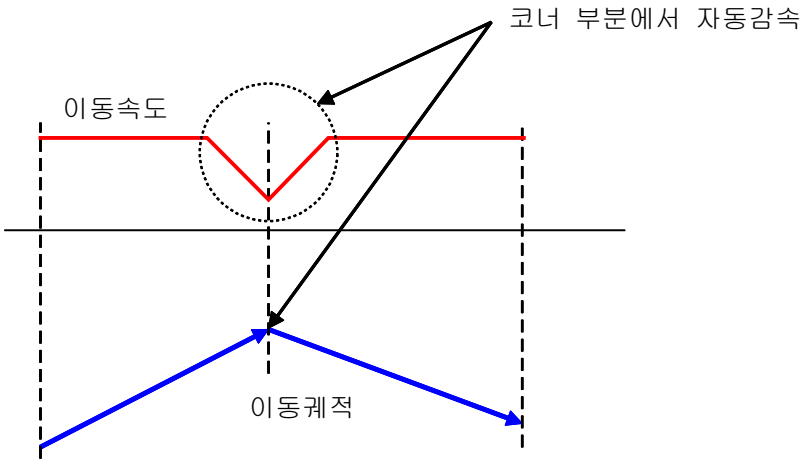


● [AXIS] : CV 각도계산이 적용되는 축을 선택합니다.



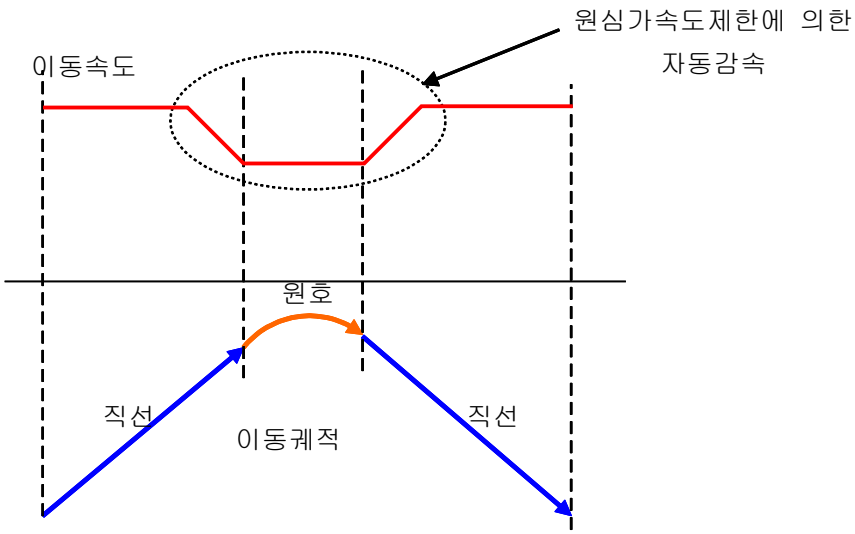
●CVDecRate : CV각의 변화에 따라 유지되는 속도비율을 설정합니다. 속도비율값은 연속이동이 허용되는 CV각도 * 피드속도(분당이송속도)값으로 나타냅니다.
이 비율값의 의미는 이동벡터의 교차각도에 따라 자동으로 절삭속도를 감속시키는 비율을 의미합니다.

예> 연속이동이 허용되는 CV각이 30도 이고, 30도 교차각에서 허용되는 최대 연속속도(분당이송속도 FEED)를 1000 이라고 하면 CVDecRate = 30 * 1000 = 30000 이 됩니다.
만약 이동궤적의 벡터 교차각도CV가 15도이면 FEED2000까지 허용되고 교차각도CV가 60도이면 FEED500까지 허용됩니다.
CVDecRate는 아래와 같이 이동궤적의 갑작스러운 교차각도에 따른 충격을 감소시키기 위한 감속비율입니다.



● DecStartAngle : 자동가감속도에 의해 적용되는 최소각도를 설정합니다.

● ArcCentDec : 원호보간이나 직선그룹에 의한 유사원궤적을 형성할 때 지정된 기계 원심가속도한계치값을 기준으로 자동감속제어를 수행합니다. 원심가속도의 한계값은 절삭피드속도설정의 Acc값을 기준으로하는 배율값입니다. 1.0 이면 절삭피드가속도 Acc값의 1배값을 원심가속도한계값으로 제한합니다. 다음의 sec (원심가속시간:초) 파라메터에 값을 기입하면 원심가속도가 미치는 영향의 시간을 조사해서 자동감속여부에 반영합니다. 일반적으로 0.0으로 설정하기 기능을 off하여 사용합니다.

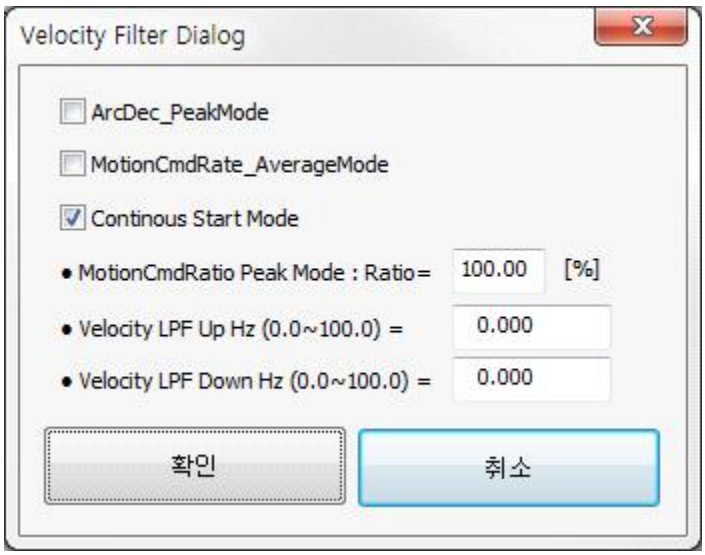


※ CVLimit 이외의 다른 자동가감속동작은 X6V3엔진버전 V01.07 이후버전에서 동작됩니다. 그 이하의 버전을 갖는 모션제품은 엔진 업그레이드가 필요합니다. (본사에 문의해 주세요.)

● MotionCmd/Sec : 최대 평균연속명령처리 시간으로써 초당 모션명령G코드블록을 처리하는 수와 비슷합니다. 짧은거리의 모션명령이 빠르게 실행될 경우 이 명령처리속도이상 초과되지 않도록 자동으로 감속하게 됩니다. 이 항목을 체크하게 되면 절삭속도제어 모든 영역에서 검사됩니다. 체크를 해제하면 내부적으로 필요영역에서만 검사합니다.

● AutoDec_LeftDist : 감속정지에 필요로 하는 이동거리보다도 짧은 이동거리를 갖는 다수의 G코드들로 구성될 때 미리읽기를 수행하여 이동가능한 잔여거리에 의해 자동가감속하는 옵션입니다.

● [Velocity Filter] 속도필터를 클릭하면 아래의 추가적인 속도제어옵션창이 보여집니다.
보다 세밀한 속도제어옵션은 대체로 짧은 직선상의 연속적인 명령처리를 필요로 하는 3D가공시 보다 상세한 속도제어 옵션들입니다.



● ArcDec_PeakMode(피크모드) : 직선조합에 의한 원심감속을 결정할 때 최대감속지점 중심으로 속도를 감속시킵니다. (설정하지 않으면 평균모드로 작동합니다. 직선조합에 의한 원심감속 평균값 적용)

● MotionCmdRate_AverageMode(명령평균처리수모드) : 초당 명령처리수제한을 앞서지정된 명령처리수값을 기준으로 평균적으로 제한합니다.
(해제하면 내부적으로 최대가능한 명령처리수를 내부적으로 계산하여 처리합니다. = 피크모드)

● Continous Start Mode = 명령처리속도를 벗어나는 명령에서도 최종 속도를 이전속도로 유지해서 출력하는 옵션 (모션엔진 V01.08 이후 버전에서만 적용됩니다.) 3D 가공시 명령처리속도를 벗어나면 속도가 초속도에서 다시 시작하는 현상을 막아줍니다,

● MotionCmdRatio Peak Mode:Ratio= 100.00[%] : 앞서 최대명령처리수제한이 피크모드로 설정되었을때 내부적으로 계산한 제한명령처리수의 배율수를 %단위로 결정합니다.

● Velocity LPF Up Hz : 속도상승에 관한 저주파 필터를 적용합니다.

● Velocity LPF Down Hz : 속도하강에 관한 저주파 필터를 적용합니다.

※ 속도세부 설정에 관한 TIP

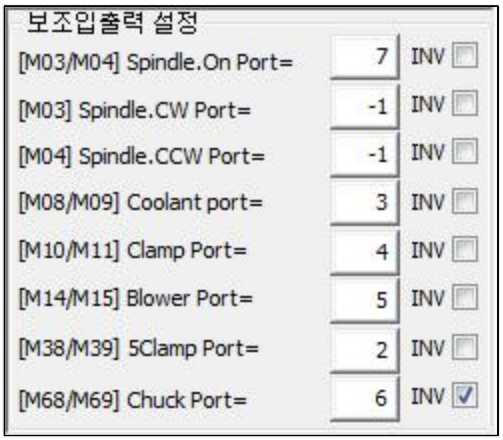
- 2.5D 가공시는 일반적으로 CVLimit=60.0, CVDecRate=18000.0, ArcCentDec=1.0 의 비율로 옵션을 적절히 설정하면 됩니다.

- 3D 가공과 같은 연속된 짧은 직선들을 보다 빠르게 부드럽게 조절하기 위해서 [ ]MotionCmd/Sec 를 해제해 줍니다., 그리고 Velocity Filter Dialog의 [√]Continous Start Mode를 설정합니다. 만약 속도상승에 대한 저주파필터를 10.0(hz)적용하면 갑작스러운 속도상승을 조절할 수 있습니다 그러나 속도하강에 대한 저주파 필터값을 적용할 경우 코너등의 자동감속이 필요한 영역에서 감속이 늦게 이루어져 가공시 힘을 받을 수도 있으므로 0.0값으로 설정하여 속도하강시 저주파필터를 적용하지 않는 것이 좋습니다.

9.6. CNC 보조장치 설정

9.6.1. M코드 보조입출력 설정

기본적으로 몇가지 M코드에 대한 입출력포트를 지정하는 그룹입니다.
포트번호는 GIO포트의 비트번호로써 0~31까지입니다. X6V3엔진을 장착한 모션보드는 기본적으로 GOut포트가 8개 있으므로 최대 0~7까지의 포트번호를 사용할 수 있니다.-1로 지정되면 사용하지 않음을 뜻합니다.
(※ELX36과 같은 확장보드를 추가하면 입출력포트가 증가됩니다.)



- [M03/M04] Spindle.OnPort : 스핀들이 정회전 혹은 역회전할 때 출력되는 출력포트번호를 지정합니다.
- [M03] Spindle.CWPort : 스핀들이 정회전할 때 출력되는 출력포트번호를 지정합니다.
- [M04] Spindle.CCWPort : 스핀들이 역회전할 때 출력되는 출력포트번호를 지정합니다.
- [M08/M09] Flood.Outport : 절삭유동작 출력포트번호를 지정합니다.
- [M10/M11] Clamp.Outport : 클램프동작 출력포트번호를 지정합니다.
- [M14/M15] Blow.Outport : 블로워동작 출력포트번호를 지정합니다.
- [M38/M39] 5Clamp.Outport : 5축 클램프동작 출력포트번호를 지정합니다.
- [M68/M69] Chuck.Outport : 메인 척동작 출력포트번호를 지정합니다.
- INV[ ] : 이 항목을 체크하면 출력신호를 반전시킵니다.

9.6.2. 전용입출력 설정

- Pause state Outport : 정지상태일 때 활성화되는 출력포트번호를 지정합니다. INV[ ] 항목을 체크하면 반전하여 출력합니다.
- PROB(Sensor) port : 조그동작이나 공구측정동작에서 사용되는 프로브센서의 입력포트를 지정합니다. 0~7 번 혹은 16번을 선택합니다. 16번을 선택하면 GIO입력포트가 아니라 모션보드의 Sync 전용신호입력을 의미합니다.
- SV.On (Z) Redirction : 일반적으로 z축은 서보브레이크를 사용할 수 있습니다. 서보브레이크 동작을 SV.ON동작과 연동해서 동작시킬 경우 SV.ON 출력을 GIO출력포트로 복사하는 기능을 합니다.

9.6.3. 제어스위치 설정

자동조작과 관련된 제어스위치를 외부 입력포트로 맵핑할 수 있는 설정그룹입니다.

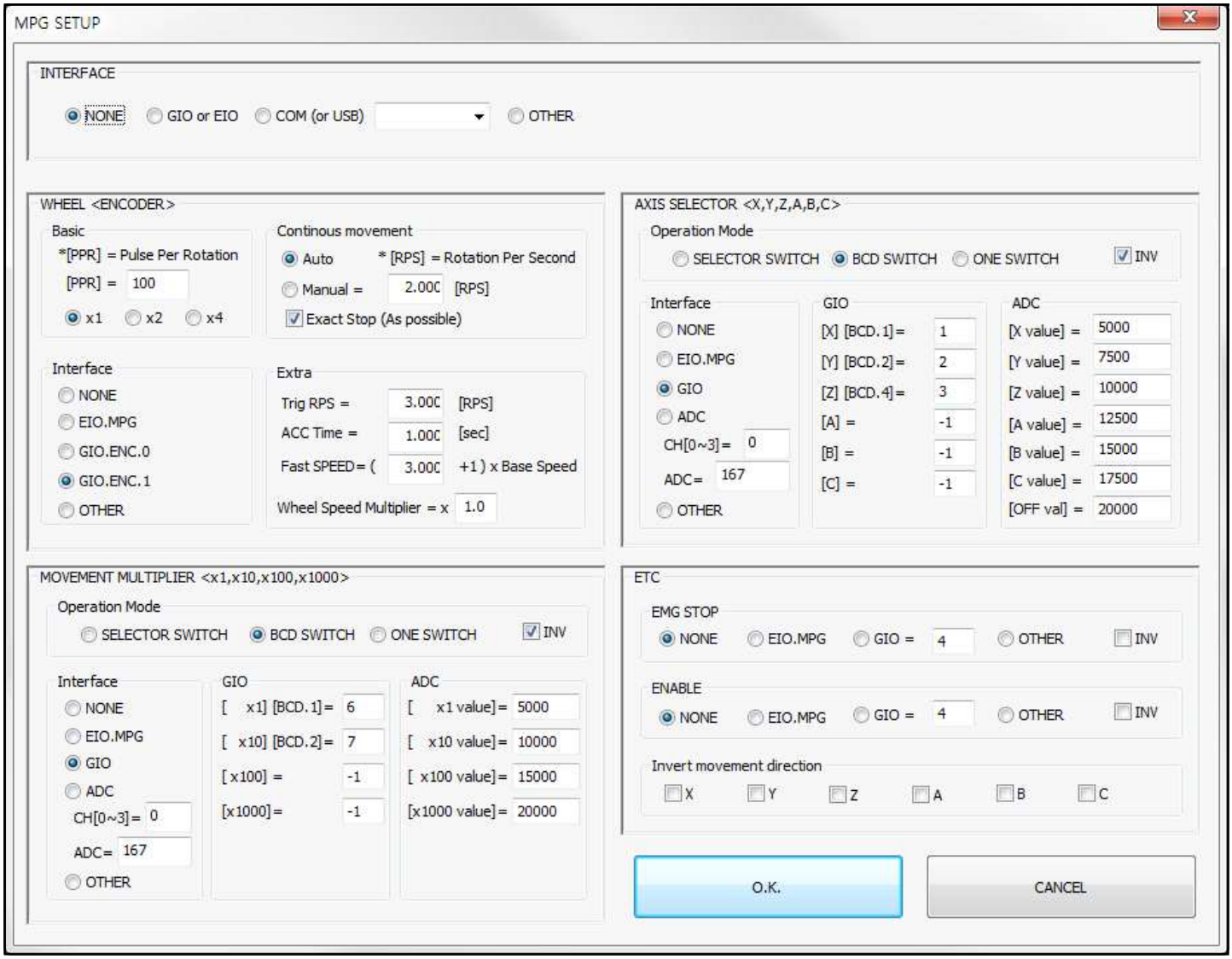
- Enable : 전체 제어스위치 맵핑여부를 설정합니다.
- Start : 자동시작버튼과 연결되는 입력포트를 지정합니다.
- Pause : 일시정지버튼과 연결되는 입력포트를 지정합니다.
- Cancel : 동작취소버튼과 연결되는 입력포트를 지정합니다.
- Single : 싱글블록과 연결되는 입력포트를 지정합니다.
- M1Stop : 옵션널스톱스위치와 연결되는 입력포트를 지정합니다.
- M.Lock : 머신록과 연결되는 입력포트를 지정합니다.
- AL.Rel : 알람일시해제버튼과 연결되는 입력포트를 지정합니다.
- Home : (전체)원점복귀버튼과 연결되는 입력포트를 지정합니다.
- Reset : 리셋버튼과 연결되는 입력포트를 지정합니다.
- INV[ ] : 이 항목을 체크하면 입력 동작신호를 반전시킵니다.

9.6.4. ELX36 확장보드 연결

아래의 사진과 같은 ELX36보트와 같은 확장보드를 연결할 때 사용되는 옵션입니다.

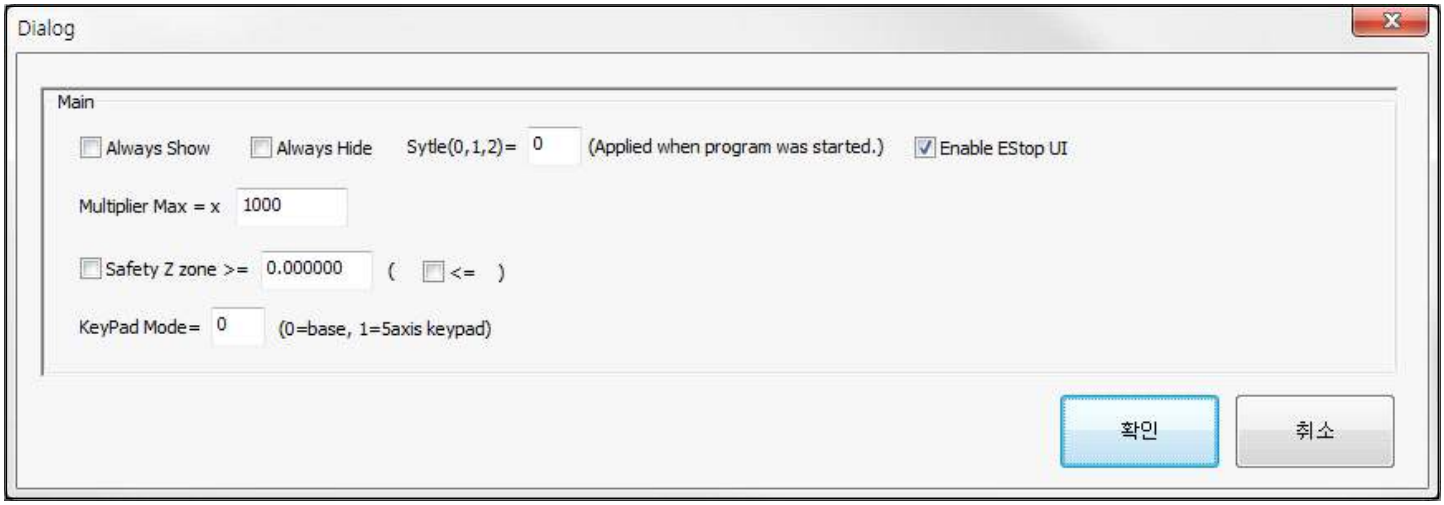
9.6.5. MPG 연결옵션

버튼을 클릭하면 아래와 같은 다소 복잡한 MPG 연결옵션창이 나타납니다. 이 설정들에 대한 자세한 내용은 “부록”의 MPG연결편 이나 “MPG연결 매뉴얼”이라는 별도의 문서를 통해 참조하실 수 있습니다.



9.6.6. JOG 동작옵션

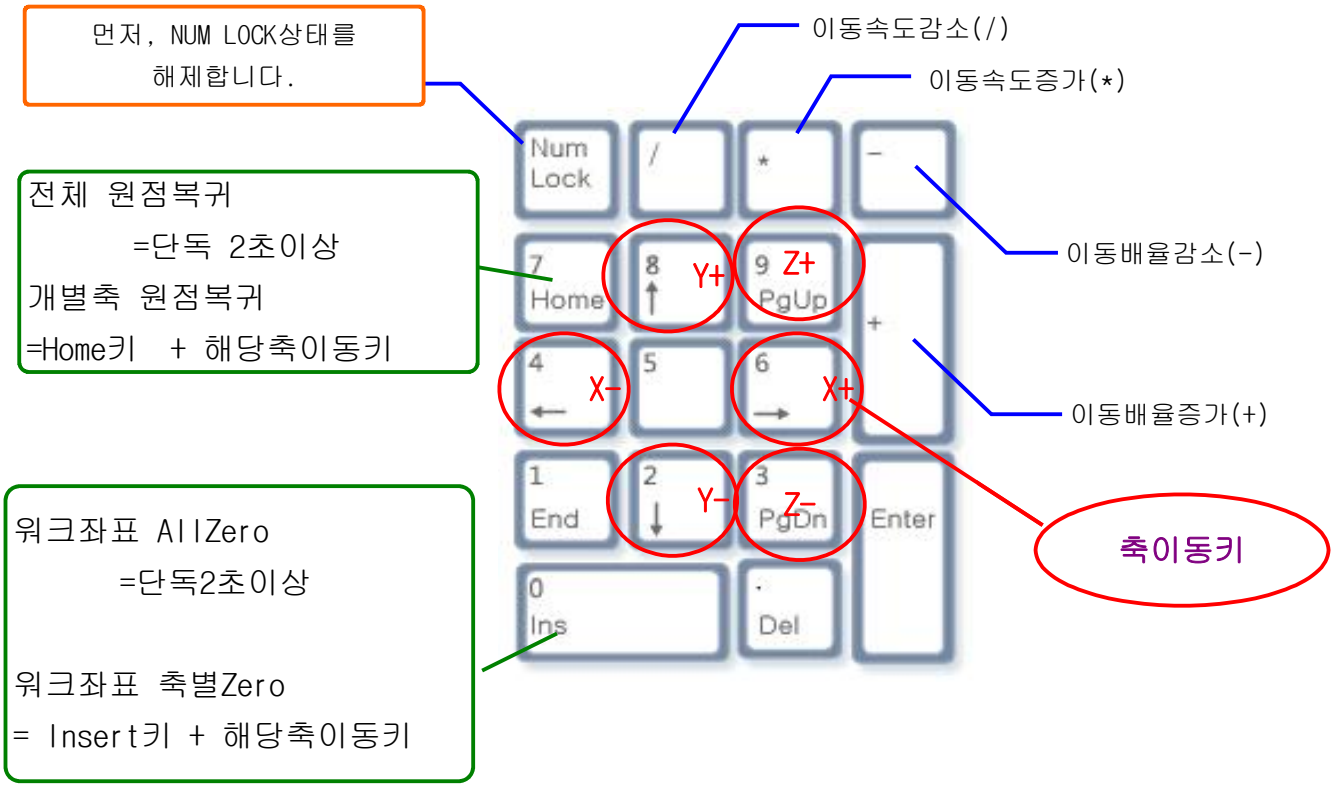
JOG 버튼을 클릭하면 아래와 같은 조그동작관련 옵션설정창이 나타납니다.



● [ ]Always Show 속성은 모션마스터프로그램의 실행시 항상 조그제어화면이 독립된 창으로 보여지는 속성입니다.
● [ ]Always Hide 속성은 조그제어화면이 동작탭화면의 전환으로만 나타나고 독립된 창으로 보여지지 않도록 설정하는 속성입니다.
두가지 속성 모두를 설정하지 않으면 기본적으로 CNC운전화면의 [JOG]버튼을 눌를 때만 독립된 조그제어화면이 나타납니다.

● Style 속성에 따라 독립 조그제어화면의 모양이 달라집니다. 다음 그림은 각각 Style=0, 1, 2에 대한 조그제어화면을 보여줍니다. 화면의 해상도에 맞게 선택하실 수 있습니다. Style 값이 변화되면 프로그램이 종료된 후 다시 시작해야만 변경된 설정이 적용됩니다.
● Enable EStop UI 는 아래의 조그제어화면상의 [비상정지]버튼/램프를 클릭할 경우 비상정지동작을 실행할 것인지를 지정하는 옵션입니다. 체크된 상태에서만 비상정지동작이 가능하며, 체크되어 있지 않다면 단지 알람상태를 점멸표시하는 램프로만 사용됩니다.

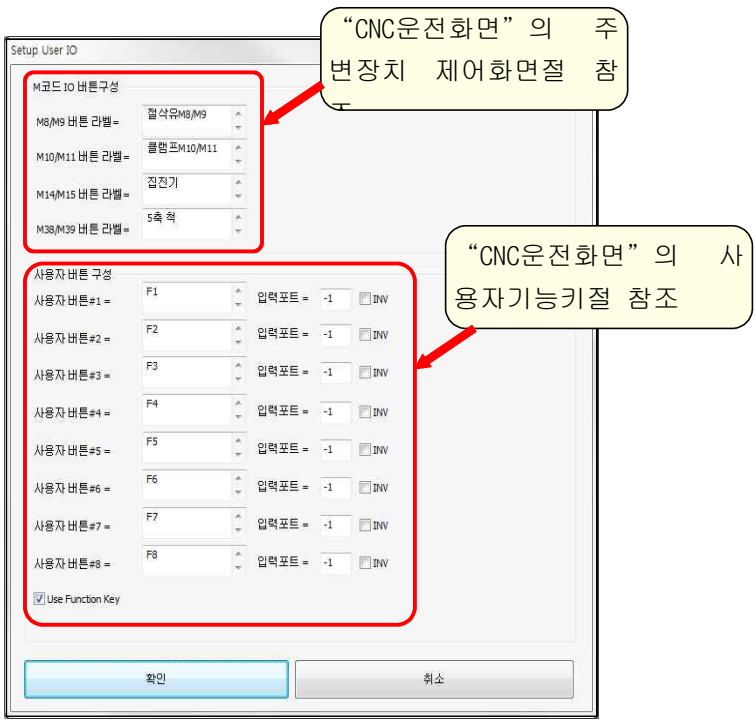
● KeyPad Mode : 0 이면 키패드입력방식이 XYZ 3축기계용으로 설정되고 1로 설정하면 5축기계용으로 설정됩니다. 3축의 기본키패드 기능맵은 아래와 같습니다. 5축용 키패드기능은 Home키와 End키는 4축 동작으로 맵핑되고, Ins키와 Del키는 5축동작으로 맵핑 됩니다.



9.6.7. 사용자 변경 버튼 / 램프 설정

버튼/램프 버튼을 클릭하면 아래와 같이 사용자에 의해 변경가능한 버튼 및 램프의 설정창이 나타납니다.

이 설정창은 CNC운전화면과 관련되면 각 파라메터와 관련된 내용은 “CNC운전화면”편의 주변장치 제어화면과 사용자기능키 절을 참조하실 수 있습니다.



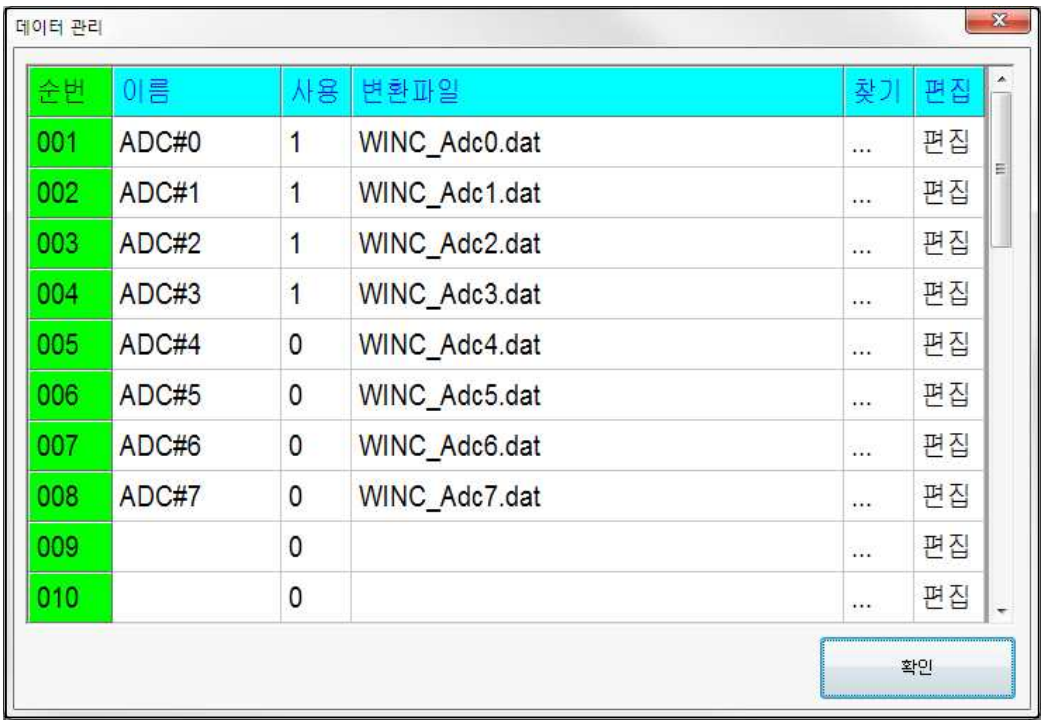
9.7. CNC 동작 테이블

CNC운전에서 필요 로하는 다양한 동작테이블을 관리하는 그룹입니다. 아래와 같이 총 7가지의 CNC동작 테이블이 있습니다. 각 테이블의 기능들은 G코드체계 및 전체 프로그램운영과 밀접하게 관련됩니다. 여기서 각 테이블별 내용과 특징을 살펴봅니다.

9.7.1. 데이터 관리 테이블

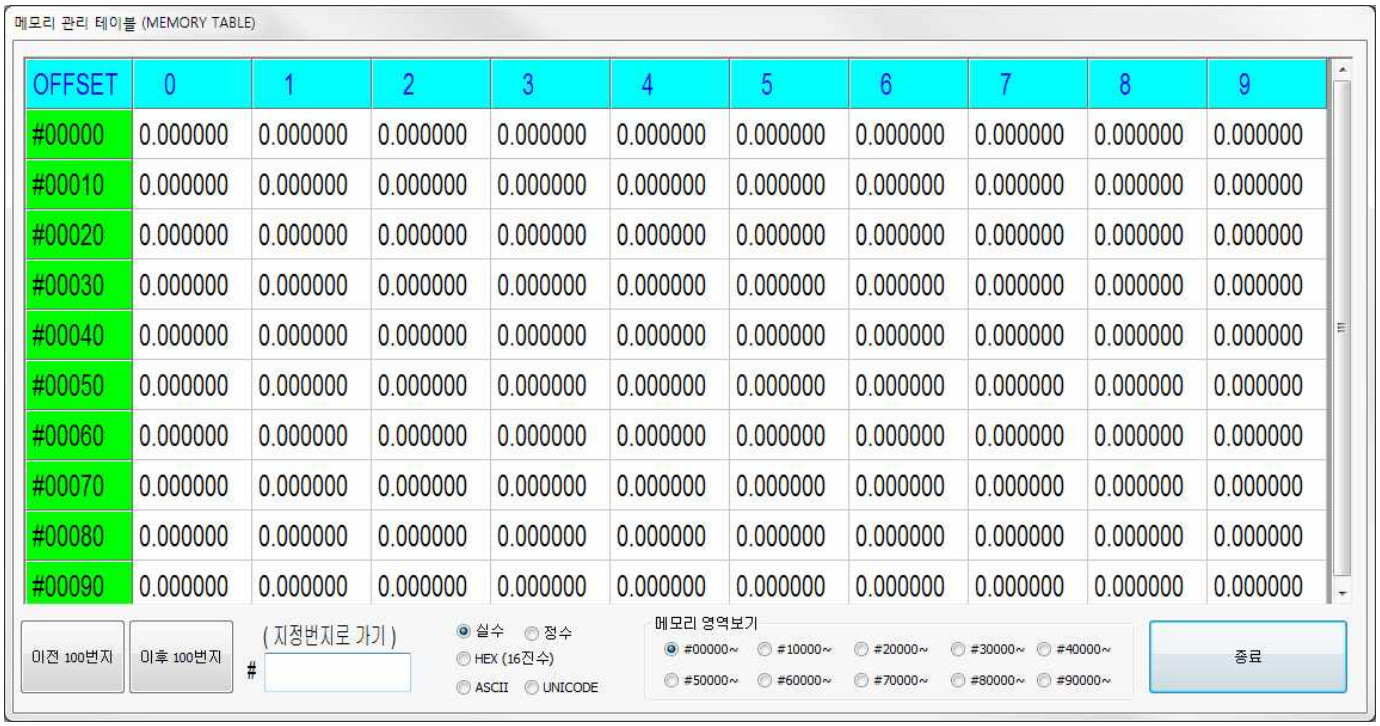
[데이터관리]버튼을 클릭하면 아래와 같은 데이터관리테이블이 나타납니다.

현재 데이터관리테이블에는 모션보드의 ADC입력데이터를 변환시켜주는 변환파일과 ADC입력채널과의 맵핑을 관리합니다.



9.7.2. 메모리 관리 테이블

메모리관리테이블 모션마스터의 CNC운전 전체에 걸친 주요 동작데이터가 기록되어 있는 시스템메모리에 대해 모니터링하고 편집하는 기능을 갖고 있습니다. 시스템메모리에 대한 내용은 “시스템 메모리”편을 참조할 수 있으며 메모리 관리테이블을 통해서 각 시스템메모리의 동작들을 확인해 볼 수 있습니다.



9.7.3. 원점 관리 테이블

원점복귀시 기계좌표값을 관리하며 전체원점복귀 동작 시퀀스를 설정하는 기능을 갖고 있습니다.

9.7.4. 워크좌표 테이블

CNC동작의 다양한 워크좌표들에 대한 기준변위들을 관리하는 테이블입니다.

워크테이블	X좌표	Y좌표	Z좌표	A좌표	B좌표	C좌표
로켈옵셋[G52]	0.000	0.000	0.000	0.000	0.000	0.000
[G54]	0.000	0.000	0.000	0.000	0.000	0.000
[G55]	0.000	0.000	0.000	0.000	0.000	0.000
[G56]	0.000	0.000	0.000	0.000	0.000	0.000
[G57]	0.000	0.000	0.000	0.000	0.000	0.000
[G58]	0.000	0.000	0.000	0.000	0.000	0.000
[G59]	0.000	0.000	0.000	0.000	0.000	0.000
[G54.00]	0.000	0.000	0.000	0.000	0.000	0.000
[G54.01]	0.000	0.000	0.000	0.000	0.000	0.000

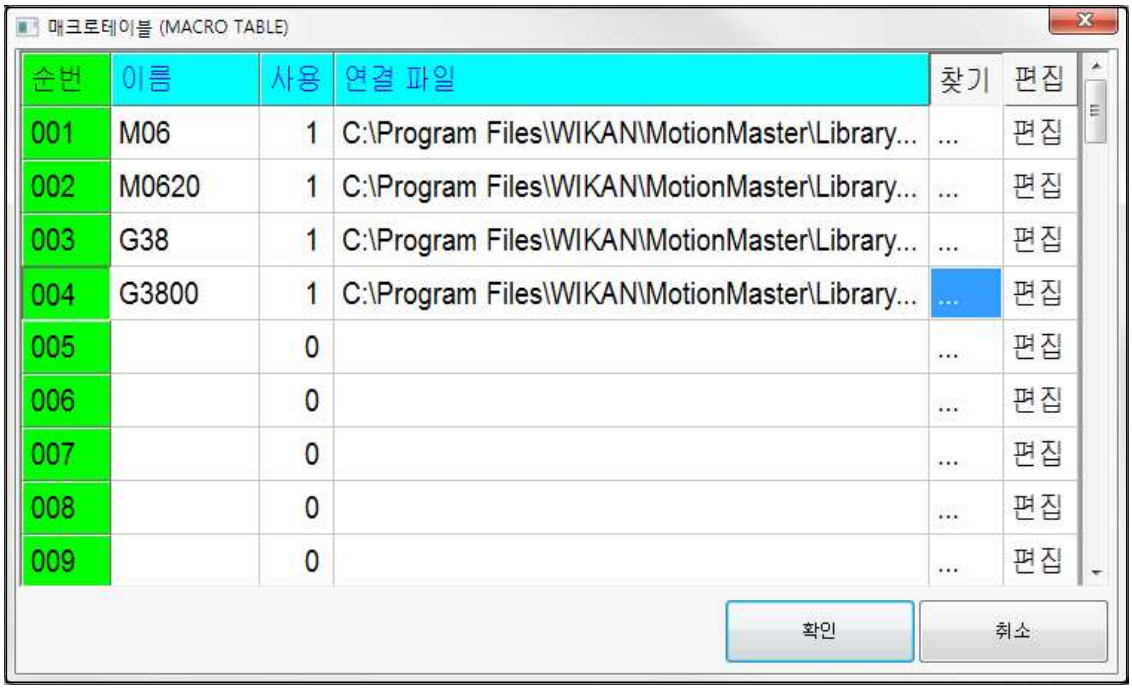
9.7.5. 공구 테이블

공구자동교환을 위한 데이터와 공구보정데이터 그리고 공구사용시간, 수명 등 다양한 공구의 특성들을 관리하는 테이블 입니다.

번호	명칭	설치	D	D.마모	H	H.마모	특성1	특성2	특성
001		0	0.000	0.000	0.000	0.000	0.000	0.000	0
002		0	0.000	0.000	0.000	0.000	0.000	0.000	0
003		0	0.000	0.000	0.000	0.000	0.000	0.000	0
004		0	0.000	0.000	0.000	0.000	0.000	0.000	0
005		0	0.000	0.000	0.000	0.000	0.000	0.000	0
006		0	0.000	0.000	0.000	0.000	0.000	0.000	0
007		0	0.000	0.000	0.000	0.000	0.000	0.000	0
008		0	0.000	0.000	0.000	0.000	0.000	0.000	0
009		0	0.000	0.000	0.000	0.000	0.000	0.000	0

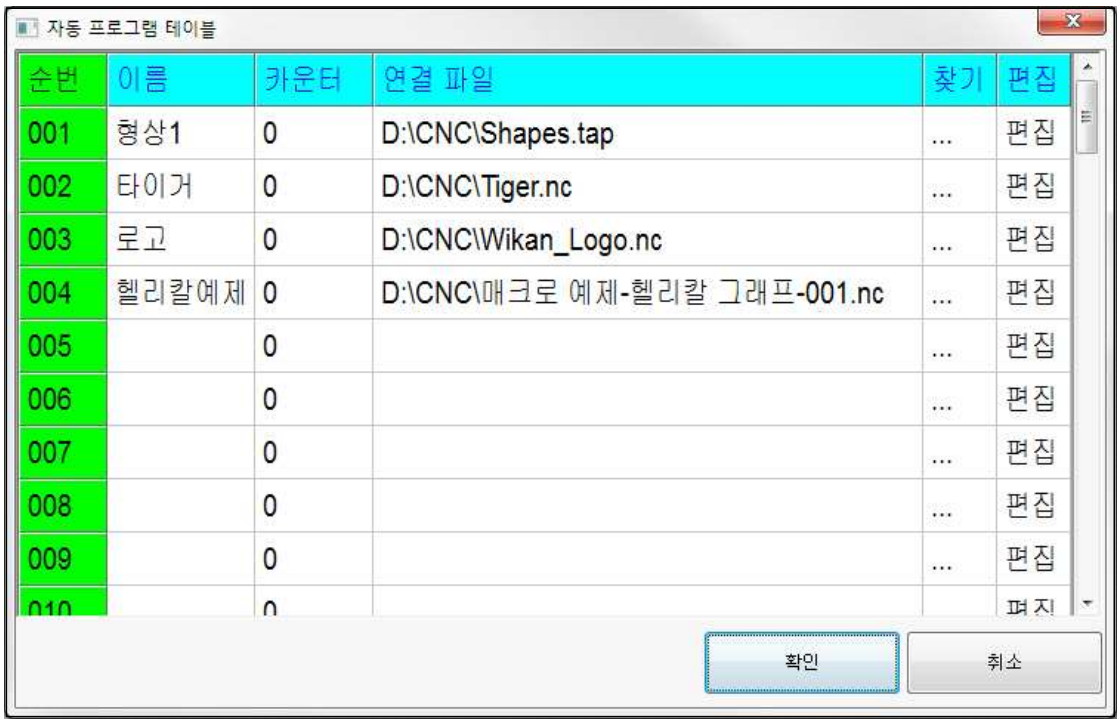
9.7.6. 매크로 테이블

매크로 테이블은 사용자에 의해 정의 가능한 G코드와 M코드를 정의하고 이 코드드 의해 실행되는 실행코드파일을 연결하는 역할을 합니다.



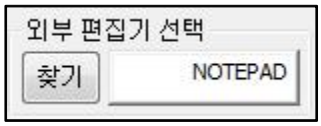
9.7.7. 자동 프로그램 테이블

자동 프로그램 테이블은 사용자의 스크립트 기능에 의해 자동으로 NC코드파일을 로딩하는 기능 갖습니다. 각 프로그램 마다의 짧은 설명 혹은 이름을 포함할 수도 있습니다.

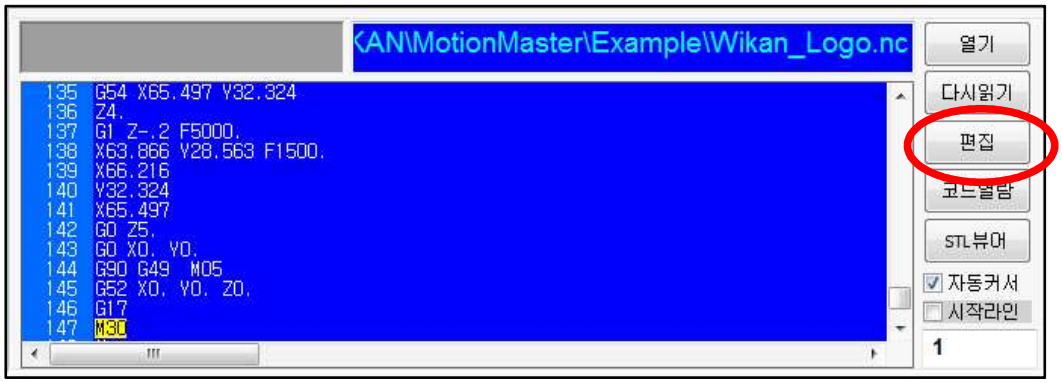


9.8. 기타 설정

<외부편집기선택>



● External NC Code Editor : “CNC조작화면"에서 “NC코드 관리"창의 [EDIT]버튼을 클릭하여 NC코드를 편집할 때 사용할 외부프로그램을 등록합니다.



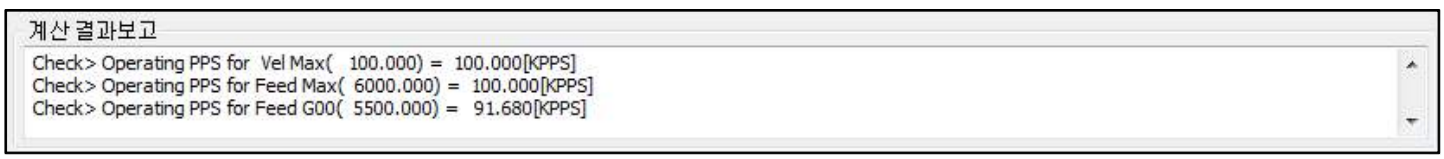
<보안암호설정>

● Password : 모션마스터프로그램은 보안암호를 설정하여 자동프로그램파일을 암호화 하여 읽고 실행하는 기능을 갖고 있습니다. 보안암호는 모션보드의 기능과 관련되어 특정 모션보드와 연결되어야만 프로그램의 실행이 가능하도록 제한됩니다. 이러한 기능을 위해서 G코드는 별도의 프로그램을 통해 암호화 코드로 전환되고 모션마스터는 이 파일을 복호화하여 읽여들여 실행합니다. (암호화에 대한 자세한 내용은 별도로 문의해 주세요)

9.9. 설정 적용과 결과 리포트

버튼을 클릭하면 현재 설정된 내용에 대한 내부연산을 수행하며, 설정내용을 적용합니다. 그리고 결과리포트를 수행합니다. 실제 대부분의 파라메터들은 변경과 동시에 적용됩니다. 그러나 위치파라메터와 속도파라메터들이 변경될 경우 내부적으로 전체 축들의 설정값들을 한번에 계산하여야 합니다. 이때 [계산적용(Apply)]를 버튼을 누르면 바르게 연산된 결과내용을 모션보드와 CNC운전 프로그램에 적용됩니다.

※ 참고> 내부적으로 주요한 연산은 전자기어비에 대한 계산과 전자기어비에 의한 오차계산, 그리고 속도배율수를 계산하는 것입니다. 이러한 내부연산에 따른 결과내용은 “모션엔진 구성설정화면”의 전자기어비설정내용 과 “모션엔진 모니터링화면”의 속도배율값 내용으로 확인해 볼 수 있습니다. 속도배율값은 최대속도성분과 1차 가감속도, 2차 가감속도의 비율을 기준으로 계산됩니다.



[계산적용]버튼을 클릭하면 현재설정된 내용에 대한 내부적연산결과중 필요로 하는 부분을 리포팅합니다. 좌측의 리포팅 내용을 살펴보면 ① VelMax값에 대한 펄스주파수(PPS)값, ② FeedMax값에 대한 펄스주파수(PPS)값, ③ G00 Feed값에 대한 펄스주파수(PPS)값 과 ④각 축별 전자기어비 적용으로 인한 1펄스당 오차발생량을 표시합니다.
PPS값은 최대값은 모션엔진이 허용하는 값을 넘지 않게 설정합니다.

10. 매크로 확장과 V스크립터

모션마스터는 기본적인 G코드와 함께 매크로 확장과 V스크립트에 의한 실행을 지원합니다.
매크로 확장에 의해 다양한 G코드와 M코드를 사용자가 임의적으로 생성하거나, IF문, WHILE문, GOTO문 등의 실행 제어명령을 사용할 수 있으며, V스크립터언어로써 다양한 함수들(수학함수,입출력합수 등...)을 사용하여 보다 광범위한 모션동작제어를 구현할 수 있습니다.

10.1. 기본문법

모션마스터는 기본적으로 명령블록단위의 문자들을 읽어 들여 한 블록씩 처리해 갑니다. 가장 일반적인 명령블록은는 G코드블록으로써 한 줄의 명령이 하나의 명령블록이 됩니다.
그 외에도 IF, WHILE, GOTO문과 같은 제어문 명령블록이 있으며, { }와 같이 중괄호로 둘려 싸여져 내장된 인터프리터로 해석되는 스크립터블록이 있습니다.
마지막으로 명령 자체가 아닌 주석문(설명문)에 해당하는 주석문블록이 있습니다.
아래에 간단한 예제와 함께 정리하였습니다.

(1) G코드 명령블록
- 범용 G코드 명령블록은 기본적 G코드와 M코드 등의 일반적인 조합으로 하나의 줄이 하나의 명령블록을 형성합니다. 이들 코드들에 대한 자세한 설명은 부록의 G코드설명부를 참조하실 수 있습니다.

```
예> G01 X100 Y100 Z100
```

(2) 실행제어 명령블록
- 조건문: IF()
- 반복문: WHILE() ~ END
- 분기문: GOTO 문, RETURN문

```
예> WHILE( angle < 360 )
            {
                moveX = radial * COS(angle);
                moveY = radial * SIN(angle);
                angle = angle + 10;
            }
            G01 X(moveX) Y(moveY)
    END
```

(3) { } 스크립터 코드블록
- 모션마스터에 내장된 인터프리터에 의한 실행

```
{
        VAR radial(10), angle(0), moveX(0), moveY(0);
        moveX = radial * COS(angle);
        moveY = radial * SIN(angle);
}
G01 X(moveX) Y(moveY)
```

(4) 주석문 블록
- 한줄 주석문 //
- 다줄 주석문 /* .... */

10.1.1. V 스크립트 코드블럭 { }

모션마스터 내부 인터프리터(V스크립터 인터프리터)를 호출하여 실행하기 위한 코드블록으로써 중괄호“{” ... “}”에 의해 둘러싸여집니다. { }코드는 ASCII코드(멀티바이트코드)로 작성되며 대소문자를 구별하여 인식합니다. 이 문법은 간단하면서 추후 지속적으로 다양한 함수들과 편리한 문법들이 추가될 수 있습니다.중괄호로 싸여진 하나의 스크립터 코드블록은 하나의 G코드블록처럼 실행됩니다. 인터프리터는 스크립터코드를 한 줄씩 읽어 실시간으로 처리됩니다.

스크립터 언어는 아래와 같이 비교적 단순한 5가지 종류의 문장으로 구성됩니다.
(1) 변수선언문
: 모든 변수는 8바이트 배정도 실수형 변수입니다. 변수명은 최대 30자까지이며 변수메모리의 전체크기는 1000개까지입니다.(버전에 따라 달라질 수 있습니다.)
: 변수명은 숫자 및 연산자기호등으로 시작할 수 없으며 내정된 키워드(변수선언 및 함수명 등...)로 지정 할 수 없습니다.

```
예> { VAR X, Y, Z  // 단일변수 X,Y,Z의 생성 }
예> { VAR A[3], B[4][4]  // 배열 변수 A, B의 생성, 최대 2차원 배열까지만 지원됩니다. }
```

<변수의 상수초기화 선언방식>
VAR X(1234); //변수 X를 선언하고 1234 값으로 초기화합니다.

(2) PRINT 문

```
예> { PRINT X, Y, Z // 변수 X,Y,Z값을 확인하는 메시지윈도우가 발생하며 확인 입력을 기다립니다. }
```

(3) 제어문
<if문>
:조건문으로써 if(){...} else if(){...} else {...} 구조를 갖습니다. else if() {...} 및 else {...} 는 생략될 수 있으며 else if() {...} 구조는 여러 개 반복될 수 있습니다.
-형식-

```
if( 평가수식1 )
{
	//...평가수식1 참일때 실행하는 코드 기술
}
else if( 평가수식2 )
{
	//...평가수식2 참일때 실행하는 코드 기술
}
else
{
	//...평가수식1.2 모두 거짓일때 실행하는 코드 기술
}
```

<while문>
:반복문으로써 while() {...} 구조를 갖습니다.
-형식-

```
while(평가수식)
{
        //... 평가수식이 참일 때 반복실행할 코드를 기술합니다.
}
```

※ while문이 무한루프일 경우 전체적인 실행이 데드락상태일 수 있으므로 주의하세요.

(4) 일반 실행문

거의 대부분의 문장은 아래와 같은 단순한 실행문입니다. 실행문은 일반적인 수식문(표현식:EXPRESSION)으로 이해할 수 있습니다.

```
예> X = A + R * SIN(DEG)          // X, A, R, DEG의 변수가 선언되었다고 가정합니다.
```

여기서 +, * 등은 연산자이며 SIN() 함수(삼각함수)가 됩니다.
다양한 연산자와 함수가 제공되며 연산의 우선순위를 위해 (괄호)를 사용할 수 있습니다.

※주의> - 연산자는 부호를 표현할 경우 괄호()를 사용하는 것이 안전합니다. 예를 들면 { X = 3 * -2 }는 비록 띄어쓰기를 했지만 내부적으로 곱하기 연산자와 빼기연산자가 나란히 있어 오류를 발생합니다. 이때 X = 3 * (-2)로 표현 해 주어야 합니다.
특히 다음의 비교 연산시에 주의할 필요가 있습니다. { X >= -2 }는 X가 -2 이상인가를 평가하는 수식이지만 실제 >= (보다 크다) 와 - 모두 연산자로 취급되므로 2개의 연산자가 나란히 올 수 없게 됩니다. 그러므로 이들 평가수식역시 괄호를 써서 { X >= (-2) }로 표현해 주어야 합니다.
비교연산자인 ==, !=, >, < , >=, <= 은 연산의 결과가 거짓이면 0.0(실수)의 값을 리턴하게 됩니다.

{ }코드블록내에 문법오류가 발생되면 메시지가 출력되고 코드블록은 실행되지 않게 됩니다.
평가결과값은 0.0을 리턴하게 되어 IF문과 WHILE문에서 사용될 경우 거짓을 리턴하는 것과 같은 결과를 발생합니다.

※참고> 하나의 라인(명령줄)에 여러 명령들을 나열할 때 세미콜론 ';' 으로 구분할 수 있습니다.

(5) 주석설명문

C언어와 비슷한 한줄 주석문 // 과 다줄 주석문 /* ... */ 지원하며 다중주석문은 /* /* ... */ */ 과 같이 내포된 다중주석문을 지원합니다.
내장된 인터프린터는 한 줄 씩 명령을 읽어 처리하는 라인인터프리터이기 때문에 여러줄의 설명문을 포함할 때에는 다줄 주석문 형식으로 표현하는 것이 실행 속도에 유리합니다. 한 줄 주석문을 여러줄로 기입하게 되면 한줄 한줄을 각각 주석명령으로 읽어들여, 한꺼번에 여러줄의 주석설명문을 읽어들이는 것보다 불리합니다.

10.1.2. 제어문

제어문은 아래와 같이 조건문과 분기분 및 반복문으로 나눌 수 있습니다.

(1) 조건문 : IF 문

형식 > IF (평가식) 참일때 다음번 코드블럭을 실행합니다. (IF { 평가식 } 혹은 IF [평가식] 모두 가능)

```
예>     {  VAR X ; X = 12 }
        IF ( X > 10 )  { PRINT X }

예>     IF ( X > 10 ) G01 X100 Y100 F200
```

(2) 분기문 : GOTO 문

형식 > GOTO N100 혹은 GOTO 100 // 100번 문장으로 분기

```
예>     IF ( X > 100 ) GOTO N100
        GOTO N400
N100    { Y = 100 }
N400
```

(3) 반복문 : WHLIE 문

형식> WHILE (평가식)   (WHILE { 평가식 } 혹은 WHILE [평가식] 모두 가능)

```
       ...
       END
```

평가식이 참일 때 WHILE 과 END사이의 코드가 실행됩니다.

```
예>
        {       // 간단한 원그리기
                VAR DEG, R, X, Y ;
                DEG = 0.0; R = 100;
        }
        F3600 // 속도를 미리 설정합니다.
        WHILE ( DEG = DEG + 1.0; DEG <= 360.0 ) // 최종수식 DEG <= 360.0 이 평가됩니다.
                {
                        X = R * COS(DEG)
                        Y = R * SIN(DEG)
                }
                G01 X(X) Y(Y)
        END
```

10.1.3. 매크로기능으로 헬리컬보간 만들기

아래는 짧은 직선을 이어서 간단한 헬리컬형상대로 보간동작을 실행하도록 작성된 코드입니다.

```
G90 G15 G49 G54 G64 G50 G69 G80
{       // 변수 선언
        VAR R, DEG, PITCH, L
        VAR X, Y, Z, CENTER_X, CENTER_Y, CENTER_Z
        VAR CNT

        // 변수 초기화
        R = 50 ; DEG = 0; PITCH = 5; L = 0 ; CNT = 0
        CENTER_X = 80;
        CENTER_Y = 80;
        CENTER_Z = 20
}
G00 X(CENTER_X) Y(CENTER_Y) Z(CENTER_Z) ; // 급속이송 명령
F3000 //속도설정
// 순환
WHILE ( DEG = 0.0; CNT < 10 ) // 최종평가식은 CNT < 10 입니다.
        WHILE ( DEG < 360 )     // 내포된 다중 반복문
                {
                        X = CENTER_X + R * COS (DEG);
                        Y = CENTER_Y + R * SIN (DEG);
                        Z = CENTER_Z + L;

                        DEG = DEG + 1;
                        L = PITCH*CNT + PITCH * DEG/360;
                }
                G01 X(X) Y(Y) Z(Z) ; // 직선보간 명령
        END
        { CNT = CNT + 1 }
END
// 프로그램 종료
M30
```

10.2. V스크립트언어 연산자 및 함수들

{ }코드블록내에서 사용할 수 있는 스크립터 명령에는 다양한 연산자 및 함수들로 구성됩니다.지원함수에 대한 최신정보는 별도의 “V스크립터 매뉴얼"을 통해 참조하실 수 있습니다. 여기서는 기본적인 연산자와 함수들에 대해 요약합니다.

10.2.1. 연산자

종류	기호	우선순위	설명
대입연산자	=	10 (가장낮음)	A = B // B값을 A에 복사
산술연산자	+	4	덧셈 A = B + C
	-	4	뺄셈 A = B - C
	/	3	곱셈 A = B * C
	*	3	나눗셈 A = B / C
	%	3	나머지계산 (나눈 후 정수 몫을 제외한 나머지값) A = B % C
승수연산	^	2	A = B ^ C ; // 2^3 => 8
괄호	( )	0 (가장높음)	연사자의 우선순위에 관계없이 ( )안이 먼저 실행됩니다.
비트 연산자	|	5	비트 OR 연산 (16비트정수값으로 치환하여 비트연산을 수행합니다.)
	&	5	비트 AND 연산
	~	2	비트 NOT(반전)연산
	$	5	비트 XOR 연산
			(OR, AND, NOT, XOR 함께지원)
	<<	5	왼쪽 시프트
	>>	5	오른쪽 시프트
논리 연산자	||	7	논리 OR 연산 (논리연산대상자가 0.0의 값이면 거짓, 아니면 참을 기준합니다.) (결과가 참이면 1.0을 거짓이면 0.0을 리턴합니다.)
	&&	7	논리 AND연산
	!	2	논리 NOT연산
	$$	7	논리 XOR연산
비교 연산자	==	6	A == B (A와 B값이 동일할 때 참(1.0), 아닐때 거짓(0.0)을 리턴)
	!=	6	A != B (A와 B값이 서로 다를 때 참, 같을 때 거짓을 리턴)
	>	6	A > B (A가 B보다 클 때 참, 작거나 같을 때 거짓을 리턴)
	>=	6	A >= B (A가 B보다 크거나 같을 때 참, 작을 때 거짓을 리턴)
	<	6	A < B (A가 B보다 작을 때 참, 크거나 같을 때 거짓을 리턴)
	<=	6	A <= B (A가 B보다 작거나 같을 때 참, 작을 때 거짓을 리턴)
			(EQ,NE,GT,LT,GE,LE 함께지원)

※ 연산자 우선순위 값이 작을수록 우선순위가 높아 먼저 실행됩니다.

<연산종류에 따른 일반적인 우선순위>
함수,() > !, ^ > * / % > 기타산술연사, 비트연산 > 비교연산 > 논리연산 > 대입연산 =

10.2.2. 수학함수

내부적은 PI(파이=3.141592654...)와 E(자연대수=2.718281828...)가 상수(고정된 수)로 내정되어 있습니다.

종류	기호	설명
삼각함수	SIN(),sin()	**대문자함수는 각도단위이며, 소문자함수는 라디안단위입니다.**
	COS(),cos()	
	TAN(),tan()	
	ASIN(),asin()	
	ACOS(),acos()	
	ATAN(),atan()	
	ATAN2(),atan2()	
지수로그승수	LN(), ln()	자연대수 e에 대한 로그입니다.
	LOG(), log()	상용LOG입니다.
	EXP(), exp()	
	POW(A,B), pow()	A^B 와 동일합니다.
변환함수	ABS(), abs()	절대값을 리턴합니다.
	SQRT(), sqrt()	제곱근값을 리턴합니다.
	RAD(), rad()	각도를 라디안값으로 변환합니다.
	DEG(), deg()	라디안을 각도값으로 변환합니다.
	ROUND(), round()	반올림합니다.
	CEIL(), ceil()	소수점이하를 올림합니다.
	FLOOR(), floor()	소수점이하를 버립니다.
	MOD(A), mod()	A 의 소수 부분을 구합니다.
	FUP(A)	소수점이하를 올림합니다. CEIL()과 동일합니다.
	FIX(A)	소수점이하를 버립니다. FLOOR()와 동일합니다.
	BCD(A)	정수값 A를 BCD 코드로 변환합니다. (16비트까지 유효)
	BIN(A)	BCD코드 A를 정수값 A로 변환합니다.(16비트까지 유효)
	INT(A)	A값을 가장 가까운 정수로 변환합니다.
기타함수		
	min(A,B)	A,B 중 보다 작은수를 리턴합니다.
	max(A,B)	A,B 중 보다 큰수를 리턴합니다.
	minabs(A,B)	A,B 중 절대값이 보다 작은수를 리턴합니다.
	maxabs(A,B)	A,B 중 절대값이 보다 큰수를 리턴합니다.
	absmin(A,B)	A,B 중 절대값이 보다 작은수의 절대값을 리턴합니다.
	absmax(A,B)	A,B 중 절대값이 보다 큰수의 절대값을 리턴합니다.
	minmax(A,min,max)	A가 min과 max범위내의 경계처리된 값으로 변환되어 리턴합니다.
	range(A,min,max)	A가 min과 max범위안에 있을 때 1, 아니면 0을 리턴합니다.
	tolerance(A,cent,tol)	A가 cent값을 기준으로 +/- tol 공차이내에 있으면 1, 아니면 0

10.2.3. PLC 기능지원

종류	기호	설명
동작제어	looptime(s)	Background.code실행속도를 s초단위로 설정합니다.

10.2.4. PLC 기능지원 : M 메모리 관련함수

종류	기호	설명
M 메모리	run_M()	M메모리동작을 활성화합니다. 일반적으로 Background.code 의 마지막에 호출합니다.
	M(i)	i번째 M메모리값을 얻습니다.
	setM(i)	i번째 M메모리값을 1로 설정합니다.
	clrM(i)	i번째 M메모리값을 0으로 설정합니다.
	togM(i)	i번째 M메모리값을 토글합니다.
	putM(i,v)	i번째 M메모리값을 v값으로 설정합니다.
	pM(i)	I번째 메모리값의 상승에지를 검출합니다.
	nM(i)	I번째 메모리값의 하강에지를 검출합니다.
	cM(i)	I번째 메모리값의 변화를 검출합니다.

10.2.5. PLC 기능지원 : 타이머 관련함수

종류	기호	설명
타이머	run_T()	타이머 동작을 활성화 합니다. 일반적으로 Background.code 의 마지막에 호출합니다. PLC타이머는 100개까지 사용가능합니다.
	startT(i, time)	i번째 타이머를 time설정값으로 시작합니다.
	setT(i, time)	i번째 타이머의 설정시간을 time으로 설정합니다.
	getT(i)	i번째 타이머의 현재시간값을 얻습니다.
	stopT(i)	i번째 타이머의 동작일 정지시킵니다.
	restartT(i)	i번째 타이머의 동작을 재개시킵니다.
	resetT(i)	i번째 타이머를 리셋합니다.
	TON(i)	i번째 타이머의 ON동작결과를 얻습니다.
	TOFF(i)	i번째 타이머의 OFF동작결과를 얻습니다.

10.2.6. PLC 기능지원 : 입출력 관련함수

종류	기호	설명
버퍼링된 입출력	read_port()	전체입력을 읽어서 버퍼링합니다. 일반적으로 Background.code 의 선두에서 호출합니다.
	write_port()	버퍼링된 출력을 장치로 전송합니다. 일반적으로 Background.code 의 마지막에 호출합니다.
	X(i)	i번째 입력값을 얻습니다.
	pX(i)	i번째 입력의 상승에지를 검출합니다.
	nX(i)	i번째 입력의 하강에지를 검출합니다.
	cX(i)	i번째 입력의 변화를 검출합니다.
	Y(i)	i번째 출력값을 얻습니다.
	pY(i)	i번째 출력의 상승에지를 검출합니다.
	nY(i)	i번째 출력의 하강에지를 검출합니다.
	cY(i)	i번째 출력의 변화를 검출합니다.
	setY(i)	i번째 출력값을 1로 설정합니다.
	clrY(i)	i번째 출력값을 0으로 설정합니다.
	togY(i)	i번째 출력값을 토글합니다.
	putY(i,v)	i번째 출력값으로 v값으로 설정합니다.
일괄포트입출력확장	Xw(i) / Yw(i)	w 접미사는 word라는 의미로 하나이 신호핀이 아니라 포트입출력 전체를 의미합니다. (Xw(), Yw(), setYw(), clrYw(), putYw() 함수로 확장적용)
추가장비입출력확장	PX(n,i)/PY(n,i)	n= 입출력장치별 인덱스 n=0) 모션보드 입출력 n=1,2) 확장 PIX48 입출력 i= 입출력포트번호 (pPX(),nPX(),cPX(),pPY(),nPY(),cPY(), setPY(), clrPY(), putPY(), togPY() 함수로 확장적용)
추가장비 일괄포트 입출력확장		w 접미사는 word라는 의미로 하나이 신호핀이 아니라 포트입출력 전체를 의미합니다. (PXw(), PYw(), setPYw(), clrPYw(), putPYw() 함수로 확장적용)

10.2.7. 시간 관련함수

종류	기호	설명
	delay(t)	t는 초단위. 시간을 지연합니다.
	wait(t)	t는 초단위.내부적으로 메세지를 처리하면서 대기합니다.
	getmsec()	1/1000초 단위로 프로그램이 시작된 이후의 msec시간을 얻습니다.
	getyear()	연도를 얻습니다.
	getmonth()	달수를 얻습니다.
	getday()	월 단위 일수를 얻습니다.
	getweekday()	주일단위 일수를 얻습니다.
	getyearday()	연단위 일수를 얻습니다.
	gethour()	일단위 시간을 얻습니다.
	getminute()	시간단위 분을 얻습니다.
	getsecond()	분단위 초를 얻습니다.

10.2.8. 메모리 관련함수

종류	기호	설명
	copy_array(d,s,n)	배열 전송 d = destination array s = source array n = array size
	set_array(d,n,v)	배열 설정 d = destination array n = array size v = set value
	shift_block(n,e,b,m,r)	시스템메모리블록을 시프트 및 로테이트합니다. n : 데이터블록 이동방식 n=0) 메모리블록 상향 시프트 n=1) 메모리블록 하향 시프트 e ; 블록 크기 b ; 블록 수 m ; 블록 배열메모리 r ; 인입, 인출 단위블록 메모리
	alloc(s)	s 바이트 크기만큼의 메모리 할당
	free(mp)	할당된 메모리포인터 해제
	load_file_alloc(fname)	fname 의 파일을 로딩하여 메모리 할당한 포인터리턴.

10.2.9. 문자열 관련함수

V스크립터 언어는 문자열변수가 따로 존재하지 않으며 모두 동일하게 VAR선언문을 사용합니다.
아래와 같이 일반적인 변수를 선언하여 문자열로 초기화 할 수 있습니다.

```
VAR str1, str2, str3;
str1 = "Motion";
str2 = "Master";
str3 = strcat(str1, str2); // str3 은 “MotionMaster”의 문자열이 저장된 테이블위치가 반환됩니다.
```

실제 문자열변수는 문자열이 등록된 내부테이블 인덱스값입니다. 문자열은 최대 64kbyte이내에서 작성됩니다.(모션마스터버전에 따라 달라질 수 있습니다.)
단독의 코드블록 내에서 선언된 문자열은 코드블록을 벗어나면 내부 문자열테이블에서 사라지게 되므로 유효하지 않게 됩니다. 단 if()문과 while()문의 제어코드블록{}에서 선언된 문자열은 해당 코드블록을 벗어나도 사라지지 않고 계속 유효하게 됩니다.

```
VAR str1, str2, str3;
{
        str1 = "Motion";
}
if(1)
{
        str2 = "Master";
}
str3 = strcat(str1, str2); // str1은 유효하지 않습니다. str2는 유효합니다.
```

<문자열 지원 함수>

종류	기호	설명
	strcat(s1,s2)	문자열s1, s2를 결합한다.
	strcmp(s1,s2)	문자열s1과 s2를 비교한다.
	strlen(s)	문자열 s의 길이를 얻는다.
	tolower(c)	소문자 바꾸기
	toupper(c)	대문자 바꾸기
	strlwr(s)	소문자열 바꾸기
	strupr(s)	대문자열 바꾸기
	sputc(s,i,c)	s문자열의 i번째 문자를 c로 대체한다.(i가 음수면 끝부터 반대로)
	sgetc(s,i)	s문자열의 i번째 문자를 리턴한다. (i가 음수면 끝부터 반대로)
	ftos(r)	실수r을 문자열로 변환
	itos(i)	정수i를 문자열로 변환
	stof(s)	문자열s를 실수로 변환
	stoi(s)	문자열s를 정수로 변환
	sprintf_str(fmt,s)	sprintf_str("string print %s", "Hellow");
	sprintf_real(fmt,r)	sprintf_real("real print %8.3lf", 123.456);
	sprintf_int(fmt,i)	sprintf_int("int print %ld", 123456);
	sprintf_char(fmt,c)	sprintf_int("int print %c", c);
	sgetkeystr(src,key)	src스트링에서 key=str 에 해당하는 str 을 리턴한다.

10.2.10. 파일 관련함수

종류	기호	설명
	save_datafile(fn,m,s)	fn(파일명)로 지정된 파일에 memory(배열혹은시스템메모리)의 데이터를 size개수만큼 저장합니다.load_datafile()함수와 쌍으로 사용됩니다.
	load_datafile(fn,m,s)	fn(파일명)로 지정된 파일의 내용을 memory(배열혹은시스템메모리)에 size개수만큼 로딩합니다. save_datafile()함수와 쌍으로 사용됩니다.
	load_file_alloc(fn)	fn(파일명)로 지정된 파일의 내용을 내부메모리에 할당받은 포인터를 리턴합니다.
	logfile(title,context)	로그파일에 title문자열과 context문자열의 로그를 기록합니다.
	fopen(fn,m)	fn=파일명, m=오픈모드(“r","w","r+","w+"), 반환=파일포인트
	fputc(fp, c)	fp로 지정된 파일에 c문자 기록
	fgetc(fp)	fp로 지정된 파일에 문자얻기
	fputs(fp, s)	fp로 지정된 파일에 문자열쓰기
	fgets(fp)	fp로 지정된 파일에 문자열읽기
	fgetl(fp)	fp로 지정된 파일에 1줄 읽기
	fprintf_str(fp,fmt,s)	fp로 지정된 파일에 형식문자열쓰기
	fprintf_real(fp,fmt,r)	fp로 지정된 파일에 형식 실수수치쓰기
	fprintf_int(fp,fmt,i)	fp로 지정된 파일에 형식 정수수치쓰기
	feof(fp)	파일 끝 검출
	fseek(fp,m,offset)	파일포인터 위치이동
	fsize(fp)	파일크기
	fclose(fp)	파일닥기
	pathfileexist(f)	파일/경로 존재검출
	deletefile(f)	파일삭제
	makedir(d)	폴더생성
	removedir(d)	폴더삭제
	splite_pathname(p,n)	p로 지정된 경로명을 n에 따라서 드라이브명(n=1), 경로명(n=2), 파일명(n=3), 확장자명(n=4), 드라이브+경로명(n=5), 파일명+확장자(n=6)으로 분해.

10.2.11. 사운드 관련함수

종류	기호	설명
	sound(soundfile)	soundfile은 경로명이 포함된 wav파일명 입니다.
	volume(v)	마스터볼륨으로써 0 ~ 100(%)를 설정합니다.
	beep(hz,t)	비프음(주파수,시간)

10.2.12. 그래프 관련함수

종류	기호	설명
	RGB(r,g,b)	r,g,b 는 각각 1바이트 red, green, blue 16진 칼라값입니다.
	rgb(c)	c는 4바이트 16진 칼라값 입니다.

10.2.13. 공통다이얼로그함수

종류	기호	설명
	file_dialog(fmt,m)	파일열기 다이얼로그입니다. fmt는 열기파일의 확장자형식(“*.nc")이고 m은 다이얼로그 생성방식입니다. 0이면 쓰기모드, 1이면 읽기모드, 2이면 현재폴더를 옮기지 않는 쓰기모드, 3이면 현재폴더를 옮기지 않는 읽기모드입니다.
	color_dialog(m,c)	색상선택 다이얼로그입니다. m은 다이얼로그 생성방식입니다. 0이면 디폴트칼라 c가 유효합니다. 1은 디폴트칼라를 무시합니다.
	keypad_dialog(s,m)	숫자입력용 다이얼로그입니다. s는 초기문자열이며 m은 다이얼로그 종류입니다.

10.2.14. 메시지박스 함수

종류	기호	설명
	messagebox(n,p,t1,t2)	n=0~6 : 메시지박스종류 0=CNC상태메시지, 1=OK메시지, 2=OK/CANCEL, 3=ABORT/RETRY/IGNORE, 4=YES/NO, 5=YES/NO/CANCEL/ 6=RETRY/CANCEL p=0~4 : 표시아이콘타입 0=알림/1=경고/2=정보/3=질의/4=오류, (단, CNC상태메세지일 경우 0=알림/1=주의/2=알람/3=메시지삭제 t1=창타이틀, t2=메시지 반환값 : 0=NO, 1=OK/YES, 2=CANCEL, 3=RETRY, 4=IGNORE, -1=ABORT,

10.2.15. 내부 인터프리터 관련함수

종류	기호	설명
	libpath(path)	call() 및 include()함수에 대한 라이브러리 경로를 설정합니다.
	get_libpath()	현재 라이브러리 경로명을 얻습니다.
	get_defaultlibpath()	디폴트로 설정된 라이브러리 경로명을 얻습니다.
	get_programpath()	현재 프로그램이 실행중인 경로명을 얻습니다.
	include(codefile)	codefile 소스를 포함시켜 해석실행합니다.
	call(codefile)	codefile 소스를 호출하여 해석실행합니다.
	compile(codestr)	codestr 문자열을 해석실행합니다.
	return(n)	현재 코드해석을 종료시킵니다.
	at(v)	v로 지정된 변수명의 변수의 시스템메모리 위치(인덱스)값을 반환합니다. (get_varindex(v)함수와 동일)
	reset_arg()	call 호출에 한 인자전달을 리셋(없앤다)
	push_arg(arg)	call 호출전에 순차적으로 실행하여 arg값을 call 실행파일에 전달한다.
	pop_arg()	call 호출에 의해 실행되는 코드내에서 push_arg()함수의 역순서로 전달된 arg값을 리턴한다.

10.2.16. 시스템 명령 및 기타함수

종류	기호	설명
시스템 외부실행	system(cmd)	cmd 문자열의 시스템명령을 실행합니다.
핫키 등록	register_hotkey(id,mod,key,str)	id = 0~199 (핫키등록번호) mod = 0x01(alt키), 0x02(ctrl키), 0x04(shift키) key = 숫자영문 대문자키코드(visual c++의 VK_ 코드) str = 핫키등록스트링

10.2.17. DLL 관련함수

윈도우즈 정규DLL함수를 호출합니다.

- DLL 함수 프로토타입> double DLL_Func(void *pSysData[], double *p1, double *p2);
- pSysData[]는 포인터배열로써 pSysData[0]은 시스템메모리맵 포인터입니다. 그 외 pSysData[] 포인터는 예약되어 있으므로 사용자가 임의적으로 사용해서는 안 됩니다 , 시스템 메모리맵은 또다시 배정도실수의 포인터배열로 (double *sysMemoryMap[]) 로 선언되어 있습니다. 이 포인터배열의 인덱스는 최대 0~99,999 까지 사용될 수 있습니다.
- p1, p2 사용자 파라메터 입니다.

종류	기호	설명
	load_dll(dllfile)	dll파일 로딩, 핸들값을 리턴합니다. 0 이면 오류
	exe_dll(h,func,p1,p2)	dll함수 실행함수 h = dll 핸들 func = 실행함수명 p1,p2 = 사용자 임의 입력 파라메터
	free_dll(h)	dll파일 해제

10.2.18. 시스템 메모리 데이터 액세스

종류	함수명	설명
시스템메모리	WINC_PUT_DATA_REAL(i,f)	시스템 메모리 I번지에 실수값 f를 기록합니다.
	WINC_GET_DATA_REAL(i,f)	시스템 메모리 I번지내용을 실수값으로 읽습니다.
	WINC_PUT_DATA_INT(i,d)	시스템 메모리 I번지에 정수값 d를 기록합니다.
	WINC_GET_DATA_INT(i)	시스템 메모리 I번지내용을 정수값으로 읽습니다.
	WINC_PUT_DATA_HEX(i,h)	시스템 메모리 I번지에 부호없는 정수값 h를 기록합니다.
	WINC_GET_DATA_HEX(i)	시스템 메모리 I번지내용을 부호없는 정수값으로 읽습니다.
	WINC_PUT_DATA_BOOL(i,b)	시스템 메모리 I번지에 논리값 b를 기록합니다.
	WINC_GET_DATA_BOOL(i)	시스템 메모리 I번지내용을 논리값으로 읽습니다.
	WINC_PUT_DATA_STR(i,"str")	시스템 메모리 I번지에 문자열 str을 기록합니다.
	WINC_GET_DATA_STR(i)	시스템 메모리 I번지내용을 문자열값으로 읽습니다.
	WINC_PUT_DATA_BIT(i, n, b);	시스템 메모리 i번지의 n번째 비트를 b값으로 기록합니다.
	WINC_GET_DATA_BIT(i,n);	시스템 메모리 i번지의 n번째 비트를 읽습니다.

10.2.19. 모션 지원함수

<기초 변환>

종류	함수명	설명
축변환	WINC_TO_AXIS(n)	축 인덱스값 n을 축 지정값으로 변환합니다. n=0 이면 X축으로 0x01 값을 리턴하며, n=1 이면 Y축으로 0x02 값을 리턴합니다. 리턴값을 c언어방식으로 표현하면 return (0x01 << n); 이 됩니다.
	WINC_TO_AXISNO(axis)	axis로 설정된 축 지정값을 축인덱스값으로 변환시킵니다. axis=0x01 이면 0을 리턴하며, axis=0x02 이면 1을 리턴합니다. LSB로부터 첫번째로 만나는 비트1의 위치를 리턴합니다.

<명령실행방식 제어>

종류	함수명	설명
즉시실행명령	WINC_ENTER_IMCMD()	이후 명령들을 즉시실행명령으로 처리합니다.
	WINC_LEAVE_IMCMD()	즉시실행명령방식을 종료합니다.
큐실행명령	WINC_ENTER_QUEUECMD()	이후 명령들을 큐실행명령으로 처리합니다.

	WINC_LEAVE_QUEUECMD()	큐실행명령방식을 종료합니다.
		기본적으로 자동동작은 큐명령실행으로 처리됩니다. 만약 즉시실행명령을 사용할 경우 아래의 순서로 사용해야 합니다.
		WINC_LEAVE_QUEUECMD() // 먼저 큐명령실행모드를 닫습니다.
		WINC_ENTER_IMCMD() // 즉시실행명령을 시작합니다.
		.... // 명령을 기록합니다.
		WINC_LEAVE_IMCMD() // 즉시실행명령을 종료합니다.
		WINC_ENTER_QUEUECMD() // 다시 원래대로 큐명령실행모드로 되돌립니다.

<일반입출력>

종류	함수명	설명
GIO 입출력	WINC_SET_GIO(p)	출력포트 p를 SET 합니다. (p는 16진수)
	WINC_CLR_GIO(p)	출력포트 p를 CLR 합니다. (p는 16진수)
	WINC_GET_GIO(a)	입력포트 값을 리턴합니다.(a=0x01, 리턴값은 16진수)
EIO 입출력	WINC_ENABLE_EIO(enable)	enable = 0 이면 확장IO 연결을 DISABLE합니다. enable = 1 이면 확장IO 연결을 ENABLE합니다.
	WINC_GET_EIO(ch)	확장IO가 ENABLE되었을 때 연결된 IO보드채널의 입력포트값을 리턴합니다.
	WINC_SET_EIO(ch)	확장IO가 ENABLE되었을 때 연결된 IO보드채널에 출력포트값을 설정합니다.

<서보입출력>

종류	함수명	설명
서보온	WINC_SERVO_ON(axis,val[])	axis축지정, val[]는 최대 8배열로 해당축 서보의 on/off(0/1)값 지정합니다. (X6V3엔진 모션컨트롤러는 통합된 하나의 SERVO ON신호출력만 가능합니다. 이때 축지정은 0xff로 전체지정합니다.)
서보리셋	WINC_SERVO_RESET(axis,val[])	axis축지정, val[]는 최대 8배열로 해당축 서보의 on/off(0/1)값 지정합니다. (X6V3엔진 모션컨트롤러는 통합된 하나의 SERVO RESET신호출력만 가능합니다. 이때 축지정은 0xff로 전체지정합니다.)

<카운터, 엔코더>

종류	함수명	설명
카운터(업)	WINC_GET_CNT(ch)	ch 채널의 카운터값을 읽습니다.
	WINC_SET_CNT(ch, value)	ch 채널의 카운터값을 value값을 설정합니다.
엔코더	WINC_ENABLE_ENC(ch,enable)	ch = { 0, 1 } 범위로 GIO의 엔코더 채널 enable = 0 이면 비활성화 1 이면 활성화 됩니다.
	WINC_GET_ENC(ch)	ch = {0,1} 채널번호 해당 채널의 엔코더카운터값을 읽어서 리턴합니다.

<PWM 출력 제어>

종류	함수명	설명
PWM	WINC_SET_PWM(ch, period, duty)	ch = {0, 1} 채널번호 period = 주기결정 카운터값 duty = 출력비 결정 카운터값 x6v3엔진기준> 실제주기 = 20kHz / period 해당 채널의 pwm파형을 출력합니다.

<ADC 입력>

종류	함수명	설명
ADC	WINC_GET_ADC(ch)	ch = {0,1,2,3} 채널번호 해당 채널의 adc값을 읽어서 리턴합니다.

10.2.20. 모션 지원함수 2

<주 동작제어>

종류	함수명	설명
자동시작	WINC_START()	자동가공을 시작합니다.
일시정지	WINC_PAUSE()	일시정지시킵니다.
동작취소	WINC_CANCEL()	일시정지혹은 정지된 동작을 완전히 취소합니다.
자동종료	WINC_END()	자동가공을 종료합니다.
비상정지	WINC_ESTOP()	비상정지시킵니다.
원점복귀	WINC_HOME(axis)	axis=0이면 전체 시퀀스 원점복귀를 실행합니다. 자동동작 중일때는 허용되지 않습니다.
리셋	WINC_RESET()	소프트리셋동작을 수행합니다.
종료	WINC_EXIT()	프로그램을 종료시킵니다.
알람 리셋	WINC_ALARM_RESET()	알람상태를 리셋시킵니다.
명령큐클리어	WINC_CLR_QUEUE()	현재 실행대기 중인 명령큐를 삭제합니다.
머신록	WINC_MACHLOCK(b)	b=1 : 머신록 동작으로 진입합니다. b=0 : 머신록 동작을 해제합니다. (반드시 1과 0동작을 한쌍으로 사용하여야 합니다.)
싱글블록	WINC_SINGLEBLOCK(b)	b=1 : 싱글블록 설정입니다. b=0 : 싱글블록 해제입니다.
블록스킵	WINC_BLOCKSKIP(b)	b=1 : 블록스킵을 설정합니다. b=0 : 블록스킵을 해제합니다.
옵셔널스톱	WINC_MO1STOP(b)	b=1 : 옵셔널스톱을 설정합니다. b=0 : 옵셔널스톱을 해제합니다.
알람 해제	WINC_RELEASE_ALARM(b)	b=1 : 현재 알람상태이면 일시적으로 알람을 해제시킵니다. b=0 : 해제된 알람에서 원래상태로 복귀합니다. (반드시 1과 0동작을 한쌍으로 사용하여야 합니다.)
구성설정저장	WINC_SAVE_CONFIG(p, b)	지정된 파일(p)에 현재구성설정을 저장합니다. b=1 이면 종료시점에서 저장되어야 하는 기본적인 구성들만 저장되며, b=0 이면 모든 정보를 저장합니다. 지정된 파일경로 p가 “” 로 아무런 경로가 없다면 내정된 구성설정파일에 덮어쓰게 됩니다.

<테이블 등록>

종류	함수명	설명
워크테이블 등록	WINC_REGISTER_WORKTABLE(n,a,p)	워크번호 n, 축 a, 위치배열 p (G52=52, G53=53, G54=54, G55=55...G59=59, G54.00=1(혹은 5400), G54.99=10(혹은 5499))
매크로 테이블 등록	WINC_REGISTER_MACROTABLE(n,v,m,p)	매크로 테이블의 n번째 위치에 유효성여부(v), 매크로이름(m), 파일경로(p) 정보를 등록합니다.
자동프로그램 테이블 등록	WINC_REGISTER_AUTOTABLE(n,t,p,s,c,g)	자동프로그램테이블의 n번째 위치에 프로그램타이틀(t), 파일경로(p), 셋카운터(s), 현재카운터(c), 불량카운터(g) 정보를 등록합니다.

<기본다이얼로그>

종류	함수명	설명
다이얼로그	WINC_DIALOG(n)	모션마스터에서 사용하는 다이얼로그 생성 n=0 : 데이터 관리 테이블 다이얼로그 n=1 : 시스템 메모리 테이블 다이얼로그 n=2 : 홈위치 관리 테이블 다이얼로그 n=3 : 워크 테이블 다이얼로그 n=4 : 공구 테이블 다이얼로그 n=5 : 매크로 테이블 다이얼로그 n=6 : 자동프로그램 테이블 다이얼로그 n=8 : 연결구성 초기화 다이얼로그 n=9 : 알람리스트 다이얼로그 n=10: 네트워크 통신설정 오픈

<파일 로딩>

종류	함수명	설명
파일로딩	WINC_LOAD_FILE_FROM_TABLE(n)	n번째 자동프로그램테이블에 등록된 파일을 로딩합니다.
	WINC_LOAD_FILE("file")	“file”로 지정된 파일을 로딩합니다.
	WINC_RELOAD()	현재 로딩된 파일을 리로딩합니다.

<실행요청>

종류	함수명	설명
실행요청	WINC_RUN_NCCODE("motion.nc");	"motion.nc" 파일 실행을 요청합니다. (자동동작상태가 아닐때)
	WINC_RUN_NCSTR("G21G90");	G코드 문자열 실행을 요청합니다. (자동동작상태가 아닐때)

<조그동작>

종류	함수명	설명
조그동작	WINC_JOG_ENABLE(b)	조그/MPG동작 여부를 설정합니다.
	WINC_JOG_MODE(m)	m=0 : 스텝조그동작 m=1 : 연속조그동작
	WINC_JOG_MOVE(axis,dir)	axis : 축지정값 dir = 0 : 축정지 dir = 1 : 증가방향 동작 dir = -1 : 감소방향 동작
	WINC_JOG_MASTER_SPEED(p)	p = 0~100% 값의 비율의 조그 한계동작속도 비율설정
	WINC_JOG_SPEED(p)	p = 0~100% 값의 비율로 조그 메인동작속도 비율설정
	WINC_JOG_AXIS_SPEED(p)	p = 0~100% 값의 비율의 조그 축별동작속도 비율설정
	WINC_JOG_MAIN_SPEED_AXIS(axis)	메인동작속도에 영향 받는 축 지정
	WINC_JOG_MULTIPLIER(m)	m = 1,10,100,1000 에서 배율선택

<피드 오버라이딩>

종류	함수명	설명
	WINC_FEED_OVERRIDE(p)	0~100 지정
	WINC_G00_OVERRIDE(p)	0~100 지정
	WINC_SPINDLE_OVERRIDE(p)	0~100 지정

<툴번호지정>

종류	함수명	설명
	WINC_SET_TOOLNO(curT, preT);	curT = -1 이면 지정변화 없음 preT = -1 이면 자동지정 preT = -2 이면 지정변화 없음

<시스템 메모리 접근>

종류	함수명	설명
시스템메모리	WINC_PUT_DATA_REAL(i,f)	시스템 메모리 I번지에 실수값 f를 기록합니다.
	WINC_GET_DATA_REAL(i,f)	시스템 메모리 I번지내용을 실수값으로 읽습니다.
	WINC_PUT_DATA_INT(i,d)	시스템 메모리 I번지에 정수값 d를 기록합니다.
	WINC_GET_DATA_INT(i)	시스템 메모리 I번지내용을 정수값으로 읽습니다.
	WINC_PUT_DATA_HEX(i,h)	시스템 메모리 I번지에 부호없는정수값 h를 기록합니다.
	WINC_GET_DATA_HEX(i)	시스템 메모리 I번지내용을 부호없는저수값으로 읽습니다.
	WINC_PUT_DATA_BOOL(i,b)	시스템 메모리 I번지에 논리값 b를 기록합니다.
	WINC_GET_DATA_BOOL(i)	시스템 메모리 I번지내용을 논리값으로 읽습니다.
	WINC_PUT_DATA_STR(i,"str")	시스템 메모리 I번지에 문자열 str을 기록합니다.
	WINC_GET_DATA_STR(i)	시스템 메모리 I번지내용을 문자열값으로 읽습니다.
	WINC_PUT_DATA_BIT(i, n, b);	시스템 메모리 i번지의 n번째 비트를 b값으로 기록합니다.
	WINC_GET_DATA_BIT(i,n);	시스템 메모리 i번지의 n번째 비트를 읽습니다.

<평션키동작>

종류	함수명	설명
좌표계	WINC_FUNCKEY(n)	n=1 이면 F1키 동작을 실행합니다. ... n=8 이면 F8키 동작을 실행합니다.

<축지정, 축교환명령>

종류	함수명	설명
축지정	WINC_ASSIGN_AXIS(ncaxis, bdaxis)	NC의 axis축을 모션보드의 축 bdaxis로 지정합니다.
축교환	WINC_EXCHANGE_AXIS(axis1, axis2)	axis1축과 axis2축을 서로 바꿉니다.

<좌표 처리>

종류	함수명	설명
좌표계	WINC_GET_WORKCOORD_NO()	현재좌표계 방식을 얻습니다. 53=G53, 54=G54...
	WINC_SET_WORKCOORD_NO(n)	n으로 지정된 방식으로 현재 좌표계를 설정합니다. G53: n=53, G54: n=54...
	WINC_SET_WORKCOORD(axis,pos[])	현재좌표계의 좌표를 pos로 설정합니다. axis = 0 이면 전체 워크좌표를 설정합니다. * 완전히 정지한 상태에서 사용해야하며, 로컬좌표, 기계좌표를 변화시키지 않으며, G54 등의 공작물좌표계의 원점을 바꿉니다. 이때 회전, 스케일 등은 적용되지 않으므로 반드시 이들 옵션들을 해제한 후 사용합니다.
	WINC_GET_MACHPOS(pos[])	현재 기계좌표값을 pos배열에 기록합니다.
	WINC_SET_MACHPOS(axis, pos[])	pos배열에 기록된 값으로 현재좌표값을 설정합니다. axis=0이면 전체 축의 좌표값을 설정합니다.
	WINC_UPDATE_NCPOS()	현재 기계위치값으로 NC위치를 새롭게 갱신합니다.
	WINC_TO_NCPOS(mPos[], ncPos[])	mPos[]의 기계위치값을 NC위치값으로 바꾸어 ncPos[]에 저장합니다.
	WINC_TO_NCPOS_COMPH(mPos[], ncPos[])	mPos[]의 기계위치값을 공구장보정이 적용된 NC위치값으로 바꾸어 ncPos[]에 저장합니다.
	WINC_TO_MACH(ncPos[], mPos[]);	ncPos[]의 NC위치값을 기계위치값으로 바꾸어 mPos[]에 저장합니다.

10.2.21. 모션 지원함수 3

<컨트롤러 명령>

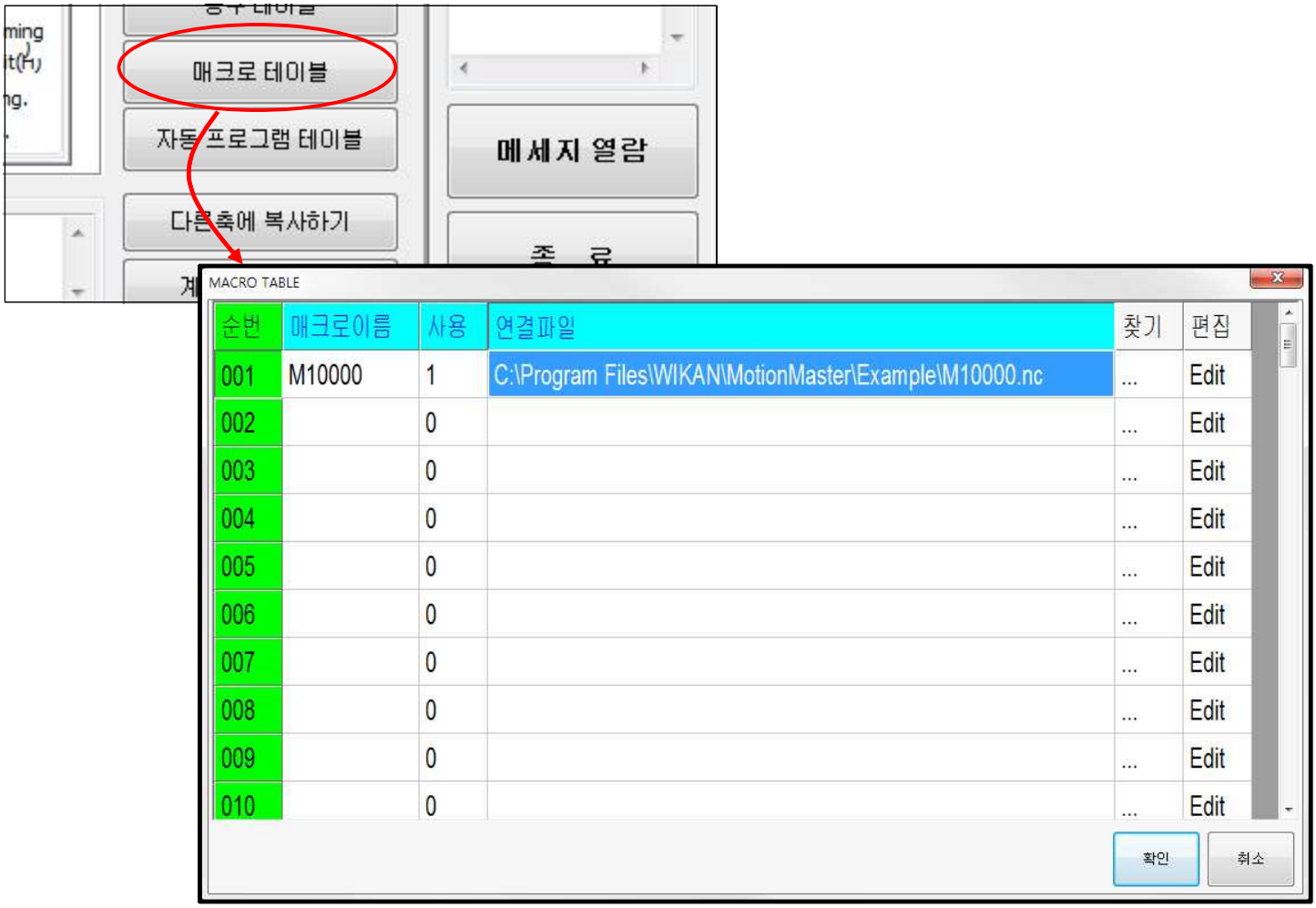
종류	함수명	설명
	WINC_CMD_MI()	다중보간 ON/OFF명령
	WINC_CMD_CLV()	선속도일정 설정 명령
	WINC_CMD_STEP()	스텝실행 ON/OFF명령
	WINC_CMD_CEXE()	연속실행 ON/OFF명령
	WINC_CMD_SYNC()	동기실행 ON/OFF명령
	WINC_CMD_DELAY()	지연명령
	WINC_CMD_WAIT()	sync신호입력대기명령
	WINC_CMD_ENABLE_EXE()	실행허가명령
	WINC_CMD_RESET()	리셋명령
	WINC_CMD_PAUSE()	일시정지명령
	WINC_CMD_RESUME()	동작재개명령
	WINC_CMD_CANCEL()	취소명령
	WINC_CMD_ALRESET()	알람리셋명령
	WINC_CMD_CLRQUEUE()	명령큐클리어명령
	WINC_VELMODE()	속도모드설정
	WINC_STARTVEL()	시작속도설정
	WINC_ACC()	가속도설정
	WINC_DEC()	감속도설정
	WINC_ACC2()	2차가속도설정
	WINC_DEC()	2차감속도설정
	WINC_DRIVEVEL()	구동속도설정
	WINC_CURVEL()	현재속도설정
	WINC_LINE()	직선보간명령
	WINC_CWARC()	CW 원호보간명령
	WINC_CCWARC()	CCW 원호보간명령
	WINC_CWCIRCLE()	CW 원호명령
	WINC_CCWCIRCLE()	CCW 원호명령
	WINC_IMOVE()	독립이송명령
	WINC_CMOVE()	연속이송명령
	WINC_SPINDLE()	스핀들회전명령
	WINC_PID_CTRL()	PID제어기 동작명령
	WINC_UPDATE_LASTPOS()	최종위치 실시간 갱신명령
	WINC_IS_ENDOF_MOTION()	모션명령 종료 체크명령
	WINC_WAIT_ENDOF_MOTION()	모션동작 종료 대기명령

11. 외부 매크로 호출

외부 매크로 호출 기능은 외부 파일로 작성된 실행코드를 읽어서 하나의 G코드 혹은 M코드로 사용하는 기능입니다. 반복적인 특정동작을 실행하는 코드파일을 작성한 후 매크로테이블에서 코드파일과 연결할 G/M코드(Gxxxx/Mxxxx)를 지정하여서 사용하면 편리합니다. 이때 코드 이름의 선두는 항상 G 혹은 M으로 시작하며 다음은 숫자값(0~99999)로 지정할 수 있습니다.

11.1. 외부 매크로 호출 테이블

CNC구성화면의[매크로테이블]버튼을 누르면 아래와 같은 매크로테이블이 나타납니다. 매크로테이블은 호출 매크로 코드와 실제 동작코드파일을 서로 연결해 줍니다.



11.2. 외부 매크로 호출 사용법

외부매크로 호출의 사용법은 호출하는 쪽(부모코드)과 호출되는 쪽(자식코드)으로 나누어 볼 수 있습니다.
우선 부모코드에서는 아래와 같이 단순히 지정된 매크로 코드이름을 기입하면 됩니다.

11.2.1. 호출 하는 쪽(부모코드)

```
G90 G15 G49 G54 G64 G50 G69 G80
...
{
        VAR X, Y, Z;    // 선언된 변수는 현재 코드파일내에서만 유효하며
                        // 다음에 호출되는 매크로파일 코드내에는 영향을 주지않습니다.
        X = 100; Y = 200; Z = 300;
}
M100.00 X(X) Y100 Z100  ; M100.00 은 M10000과 동일합니다. M10000을 외부 매크로 이름으로 가정합니다.
                        ; 인자는 X100 Y100 Z100 입니다.
...
..
M30  // 프로그램 종료
```

11.2.2. 호출 되는 쪽(자식코드) : M10000.nc파일

```
// 인자를 받습니다.
{
        VAR X, Y, Z // 매크로코드내의 변수선언은 현재 코드파일내에서만 유효합니다.
        X = #31023 // X인자
        Y = #31024 // Y인자
        Z = #31025 // Z인자

        #100 = X
        #101 = Y
        #102 = Z
}
IF { #100 < 0 } RETURN
G00 X#100 Y#101 Z#102
RETURN  // M30을 사용하면 전체프로그램을 종료시킵니다.
        //  RETURN을 사용하면 부모코드로 돌아갑니다.
```

11.2.3. 매크로 내의 매크로 호출

A매크로파일에서 B매크로파일을 호출 할 수 있으며 B매크로파일에서 다시 C매크로파일을 호출할 수 있습니다.
이러한 계층적인 호출은 최대 8레벨까지만 가능합니다.
특정 매크로내에서 자신이 다시 호출되면 기본적인 내정동작 있으면 해당동작을 수행하지만 없다면 아무 동작을 실행하지 않습니다. (재귀호출 불가)

11.2.4. 외부 매크로 인자전달 예제

아래와 같이 해당 하나는 인자들의 #번지를 통해서 X,Y,Z 과 같은 인자(PARAMETER)들을 간단하게 입력받을 수 있습니다. 해당 인자에 대한 #번지주소는 이전 장의 “시스템 메모리 맵”의 “인자 Parameter"편을 참조하실 수 있습니다.

여기서는 직교좌표 G01동작을 극좌표로 바꾸어 동작하는 새로운 G01을 만들 동작시켜봅니다.
아래의 순서대로 작성한 후 실행해 봅니다.

(1) 호출되는 쪽 파일(외부 매크로코드 : G01.nc 파일) 작성

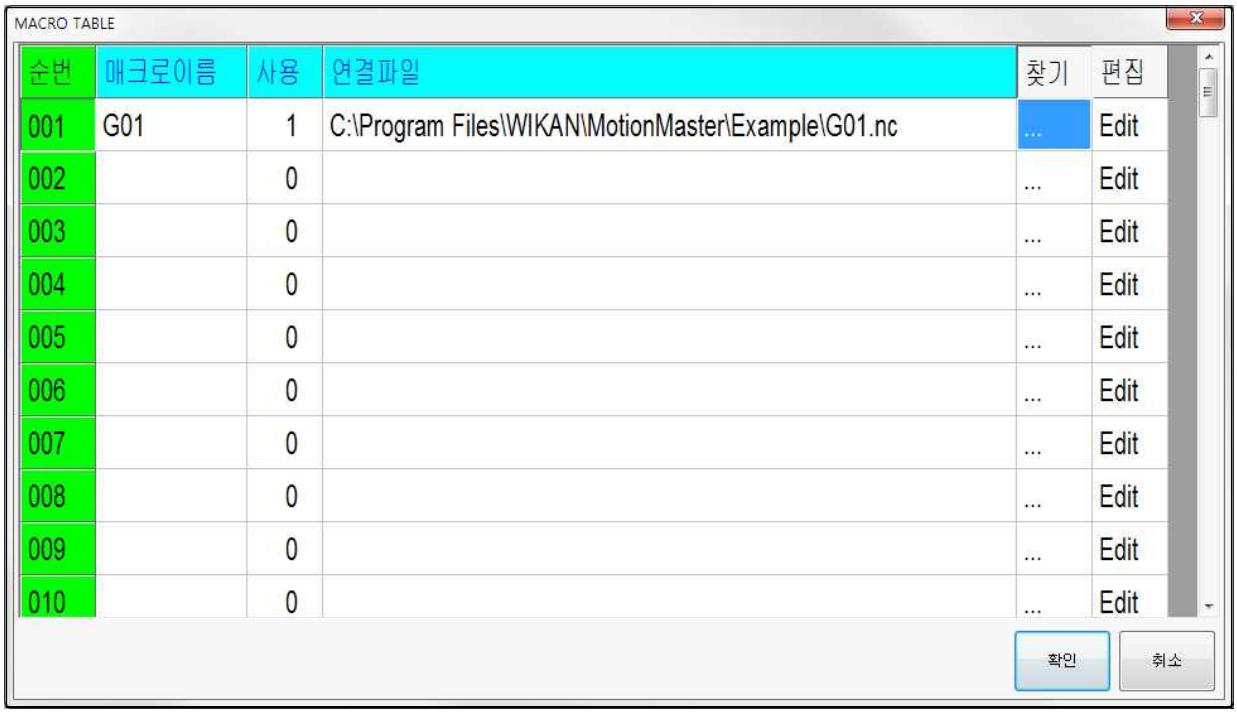
```
{
        // 각 변수들을 시스템메모리 인자영역으로 지정합니다.
        VAR P_X:#31023, P_Y:#31024, P_Z:#31025;
        VAR X, Y, Z;
        X = P_X*COS(P_Y)
        Y = P_X*SIN(P_Y)
        Z = P_Z
}
G01 X(X) Y(Y) Z(Z)
RETURN
```

(2) 호출하는 쪽 파일(Main.nc 파일) 작성
G01코드가 매크로정의 되어 G01.nc파일로 연결되었다고 가정합니다.

```
G90 G15 G49 G54 G64 G50 G69 G80
F3000
G00 X0 Y0 // 초기위치 이동
G01 X100 Y0
G01 X100 Y90
G01 X100 Y180
G01 X100 Y270
G01 X100 Y360
G00 X0 Y0
M30 // 프로그램 종료
```

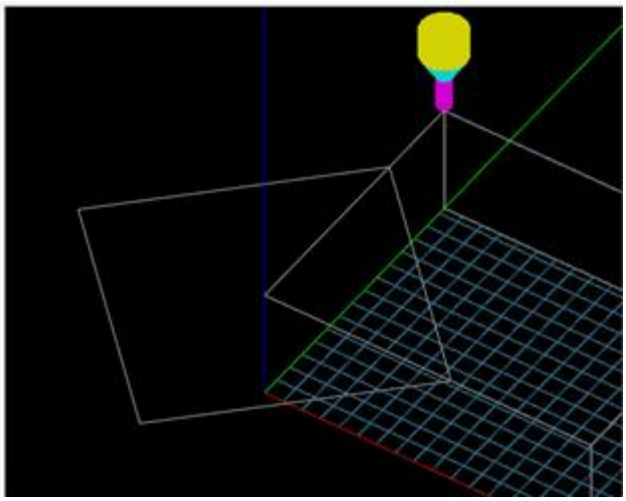
(3) 매크로테이블 등록 (호출되는 파일 : G01.nc 파일)

CNC CONFIG화면의 MACRO테이블에 아래와 같이 G01을 등록합니다.



(4) 프로그램오픈 및 코드실행 (호출하는 파일 : Main.nc 파일)

Main.nc파일을 오픈해서 로딩해보면 아래와 같은 마름모형상이 나타납니다.



12. 시스템 메모리

시스템 메모리는 CNC운영을 담당하는 WINC의 주요 제어정보 메모리영역과 임의적으로 사용할 수 있는 사용자 메모리 영역을 포함합니다.
0번지 부터 최대 99999번지까지 총 100,000개의 메모리단위로 구성됩니다(모션마스터 버전에 따라 확장될 수 있습니다). 이러한 메모리는 기본적으로 8바이트 배정도부동소수점변수(실수변수)로 구성되어 있습니다. 그러나 때로는 정수, 부울 및 문자열과 같은 속성으로 내용이 저장되어 있을 수 있습니다. 배정도실수가 아닌 다른 형태의 데이터일 경우 단순히 #1000 = 33; 과 같은 매크로확장문법이나 V스크립터문법으로 액세스할 수 없으며 데이터의 형에 따라 아래의 함수 중 하나를 사용해야 합니다.
<자료형 : 읽는 함수 / 기록 함수> 형태로 표현
- 배정도 실수 : WINC_GET_DATA_REAL() / WINC_PUT_DATA_REAL()
- 4바이트 정수자료형(부호있는정수) : WINC_GET_DATA_INT() / WINC_PUT_DATA_INT()
- 4바이트 HEX자료형(부호없는정수) : WINC_GET_DATA_HEX() / WINC_GET_DATA_HEX()
- 바이트 : WINC_GET_DATA_BYTE() / WINC_PUT_DATA_BYTE()
- 2바이트 워드 : WINC_GET_DATA_WORD() / WINC_PUT_DATA_WORD()
- 1바이트 부울 : WINC_GET_DATA_BOOL() / WINC_PUT_DATA_BOOL()
- 1비트 : WINC_GET_DATA_BIT() / WINC_PUT_DATA_BIT()
- 문자열 : WINC_GET_DATA_STR() / WINC_PUT_DATA_STR()

8바이트 배정도 실수가 아닌 경우 시스템 메모리 테이블표의 비고 난에 “부울”, “정수” “문자열”등의 데이터 자표형이 별도로 표기됩니다.

여기서는 CNC시스템 메모리영역 #30000 ~ #49999 영역의 내용을 살펴봅니다.
이들 시스템데이터의 변경은 즉각적으로 CNC동작에 반영될 수도 있으며, 단순히 읽기만 되는 영역도 존재합니다. CNC시스템 메모리영역을 읽고 쓸 때에서는 CNC시스템동작에 지장이 없도록 최대한 안전한 코드에 의해 실행되어야 합니다.

12.1. 시스템 메모리 그룹

총 100,000개의 메모리는 아래와 같은 응용영역으로 나누어서 사용됩니다. 절대적인 영역구분은 아니지만 #30000~49999 번지의 내용은 CNC시스템에서 반드시 사용되며, #50000 이후는 추후 티칭과 다른 응용용도로 사용될 수 있으므로 사용자에 의한 임의적인 사용영역은 0000번지에서 최대 19999번지까지의 영역이 안전합니다.

번지 영역	그룹
#00000 ~ #9999	사용자 임시변수 영역 (사용자가 임의적으로 사용할 수 있는 영역)
#10000 ~ #19999	사용자 시스템 정의 영역 (사용자가 임의적으로 사용할 수 있는 영역)
#20000 ~ #29999	CNC 위저드 기본응용영역
#30000 ~ #39999	CNC 시스템 파라메터 영역
#40000 ~ #49999	CNC 확장 시스템 파라메터 영역
#50000 ~ #99999	티칭 등 응용프로그램 확장 데이터 영역

12.2. 좌표계

번지	설명	비고
#30000 ~ #30005	현재 기계좌표	각 번지 순서대로 X,Y,Z,A,B,C 축에 대한 내용입니다.
#30010 ~ #30015	현재 워크좌표	
#30020 ~ #30025	현재 WORK 원점 ORG	
#30030 ~ #30035	현재 LOCAL 원점 ORG	
#30040	코드명령 최종 기계좌표	
#30050	코드명령 최종 NC코드 좌표	
#30060	단위 스케일	
#30070	회전	
#30080	회전중심	
#30090	스케일	
#30100	스케일중심	

12.3. 자동동작

번지	설명	비고
#30110	G37 연산결과값	
#30111	자동동작 작업시간	1/1000초단위
#30112	자동동작 설정카운터	
#30113	자동동작 실행카운터	

12.4. 스캔모드

번지	설명	비고
#30114	스캔모드상태 0 = 실행모드 1 = 비실행모드(그래픽처리등)	비실행모드 :현재코드를 읽는 이유가 실행이 아닌 그래픽처리 등의 목적일 때.

12.5. 종료동작

번지	설명	비고
#30115	M30 코드에 의한 종료대기 1 = 종료대기상태	이 메모리값이 1일 때, 0으로 전환시키면 자동프로그램은 강제종료.

12.6. MDI실행상태

번지	설명	비고
#30116	1 = MDI 혹은 수동 NC코드 실행중 0 = 실행없음	
#30117	예약(MDI실행상태)	(카운터) MDI 1회 실행시마다 3회 증가 +1 :준비단계 +2 :실행단계 +3 :종료단계

12.7. 메인쓰레드 실행속도제어

번지	설명	비고
#30118	전체 처리속도(초당 처리수)	최소 100에서 2000 이내 (일반적으로 #30119의 1~2배값)
#30119	실시간 처리속도(초당 처리수)	최소 100에서 1000 이내 (PC성능에 따라 조절해줍니다.)

12.8. FEED / 스핀들 RPM

번지	설명	비고
#30120	G00 FEED_RATE	오버라이딩 적용안 된 값
#30121	FEED_RATE	오버라이딩 적용안 된 값
#30122	SPINDLE RPM	오버라이딩 적용안 된 값
#30123	SPINDLE M03/M04/M05(CW/CCW/STOP)	0=STOP(M05),1=CW(M03),-1=CCW(M04)
#30124	SPINDLE RPM 설정용 버퍼	
#30125	G00 피드 오버라이딩	
#30126	G01/G02/G03 피드 오버라이딩	
#30127	스핀들 오버라이딩	
#30128	현재 피드	
#30129	현재 RPM	

12.9. 잔량좌표/목표좌표

번지	설명	비고
#30130	현재 잔량좌표	
#30140	현재 목표좌표	

12.10. 마스터 인에이블

번지	설명	비고
#30150	AUTO 마스터 인에이블	사용자 응용영역
#30151	MDI 마스터 인에이블	
#30152	JOG 마스터 인에이블	
#30153	MPG 마스터 인에이블	
#30154	HOME 마스터 인에이블	
#30155	EDIT 마스터 인에이블	
#30156	CNC 명령허용 마스터 인에이블	(읽기)
#30157	UIJOG 마스터 인에이블	
#30158	조이스틱 인에이블	
#30159	사용자 인터페이스 인에이블	사용자 응용영역

12.11. 자동시작라인

번지	설명	비고
#30160	ENABLE 자동시작라인	(부울)
#30161	자동시작라인번호	(정수)

12.12. 커스텀 기본동작 지원

번지	설명	비고
#30162	자동모드(/~수동모드)	
#30163		
#30164		
#30165	커스텀화면 종류	
#30166`	설정화면 진입 플래그	
#30167	팝업화면 생성 플래그	
#30168	커스텀화면 생성 플래그	
#30169	전원초기 시작플래그	

12.13. CNC 동작 상태

CNC머신의 전체 기계제어상태를 표시하는 메모립니다. (읽기만 가능)

번지	설명	비고
#30170	동작상태값	(동작상태값)
#30171	전원인가초기상태	해당상태=1, 아니면=0, (10)
#30172	연결대기상태	해당상태=1, 아니면=0, (12)
#30173	전원ON 리셋 진입상태	해당상태=1, 아니면=0, (15)
#30174	전원ON 리셋 초기화 상태	해당상태=1, 아니면=0, (20)
#30175	소프트 리셋상태	해당상태=1, 아니면=0, (50)
#30176	수동대기상태	해당상태=1, 아니면=0, (60)
#30177	자동대기상태	해당상태=1, 아니면=0, (110)
#30178	자동시작진입상태	해당상태=1, 아니면=0, (120)
#30179	자동기본동작상태	해당상태=1, 아니면=0, (130)
#30180	자동오류상태	해당상태=1, 아니면=0, (190)
#30181	자동종료상태	해당상태=1, 아니면=0, (180)
#30182	자동일시정지상태	해당상태=1, 아니면=0, (140)
#30183	자동작업취소상태	해당상태=1, 아니면=0, (160)
#30184	싱글블록대기상태	해당상태=1, 아니면=0, (145)
#30185	프로그램종료진입상태	해당상태=1, 아니면=0, (200)
#30186	머신록 동작상태	해당상태=1, 아니면=0
#30187	싱글블록 상태	해당상태=1, 아니면=0
#30188	옵션스톱 상태	해당상태=1, 아니면=0
#30189	알람 해제상태	해당상태=1, 아니면=0
#30190	AllHome 동작상태	해당상태=1, 아니면=0
#30191	알람상태	해당상태=1, 아니면=0
#30192	동작대기 전상태 (전원투입~수동대기 전상태까지)	해당상태=1, 아니면=0
#30193	수동대기 상태 (전원투입~자동대기 상태까지)	해당상태=1, 아니면=0
#30194	자동동작 상태 (자동대기 후 상태~ : 자동동작중)	해당상태=1, 아니면=0
#30195	임의축 Home 동작상태	해당상태=1, 아니면=0
#30196	모션일시중지상태	해당상태=1, 아니면=0 (2=M01옵셔널 스톱동작상태)
#30197	자동프로그램 로딩중	해당상태=1, 아니면=0
#30198	자동프로그램 실행완료(정상종료)	해당상태=1, 아니면=0
#30199	머신 동작중	해당상태=1, 아니면=0

12.14. 기계 원점

기계 원점테이블의 내용과 동일합니다.

번지	설명	비고
#30200	제1 기계원점	각 번지순서대로 X,Y,Z,A,B,C 위치 값입니다.
#30210	제2 기계원점(공구원점)	
#30220	제3 기계원점	
#30230	제4 기계원점	

12.15. 원점복귀 완료상태

번지	설명	비고
#30240	All Home 동작 완료상태	리셋 동작 후 완료상태 여부 0 = 완료없음 1 = 하드웨어 리셋 후 완료상태 2 = 소프트웨어 리셋 후 완료상태
#30241	개별 X축 Home 동작 완료상태	
#30242	개별 Y축 Home 동작 완료상태	
#30243	개별 Z축 Home 동작 완료상태	
#30244	개별 A축 Home 동작 완료상태	
#30245	개별 B축 Home 동작 완료상태	
#30246	개별 C축 Home 동작 완료상태	
#30247	개별 S축 Home 동작 완료상태	
#30248	개별 S2축 Home 동작 완료상태	
#30249	원점복귀완료 전 CNC명령허용상태	0 = 허용안함, 1 = 허용

12.16. 현재시간

번지	설명	비고
#30250	년	
#30251	월	
#30252	일	
#30253	주일	
#30254	연일	
#30255	시	
#30256	분	
#30257	초	

12.17. 동작 시간 및 이동거리 관리

번지	설명	비고
#30260	프로그램 총 동작시간 sec	
#30261	프로그램 동작시간 sec	
#30262	프로그램 동작시간 min	
#30263	프로그램 동작시간 hour	
#30265	머신 누적 총 사용시간 sec	
#30266	머신 동작 누적 총 사용시간 sec	
#30267	머신 자동동작 누적 총동작시간 sec	
#30268	머신 누적 사용시간 min	
#30269	머신 누적 사용시간 hour	
#30270	자동 총 동작시간 sec	
#30271	자동 동작시간 sec	
#30272	자동 동작시간 min	
#30273	자동 동작시간 hour	
#30274	예상시간 총 시간 sec	
#30275	잔여시간 총 sec	
#30276	잔여시간 sec	
#30277	잔여시간 min	
#30278	잔여시간 hour	
#30280	스핀들1 누적사용시간 sec	출력포트지정시간 동작가능
#30281	스핀들1 현재사용시간	
#30282	스핀들1 누적사용시간 min	
#30283	스핀들1 누적사용시간 hour	
#30284	스핀들1 사용시간 제어명령	-1=사용시간리셋, 0=동작없음, 그외는 지정값으로 시간설정 (현재사용시간은 zero세팅)
#30285	스핀들2 누적사용시간	사용안함
#30286	스핀들2 총 사용시간	사용안함
#30290	기계 총 이동거리	
#30291	기계 G01/G02/G03 총 절삭거리	
#30292	자동동작 총 이동거리	
#30293	자동동작 총 커팅거리	

12.18. 스캔타임

번지	설명	비고
#30300	메인쓰레드 스캔타임	
#30301	메인쓰레드 스캔타임 평균	
#30310	명령스캔 카운터	
#30311	명령스캔 초당평균	
#30320	실행속도 최대 락(LOCK)	부울 (내부적으로 자동동작) (G코스 실행실시간성을 최대로 높일 때 1, 아니면 0)

12.19. 동작버전

번지	설명	비고
#30330	모션마스터 버전정보	Vxx.xx
#30331	모션마스터 갱신정보	Rxx.xx
#30340 ~ #30346	모션보드 하드웨어 정보	

12.20. 원점복귀 동작정보

번지	설명	비고
#30347	현재(전체) 원점복귀 스텝	
#30348	현재(전체+개별) 원점복귀 동작축	
#30349	현재(개별) 원점복귀동작 동작축	

12.21. 동작카운터

번지	설명	비고
#30350	자동 누적 동작카운터	
#30351	자동 누적 OK 동작카운터	
#30352	자동 누적 NG 동작카운터	
#30355	자동 동작카운터	
#30356	자동 OK 동작카운터	
#30357	자동 NG 동작카운터	
#30358	자동 동작카운터 활성화유무	

12.22. 축 엔코더

번지	설명	비고
#30360 ~	각 축별 엔코더 기계좌표값	
#30370 ~	각 축별 엔코더 NC좌표값	
#30380 ~	각 축별 엔코더 옵셋값	
#30390 ~	예약	

12.23. 타이머 플래그

번지	설명	비고
#30400	0.01초 토글 플래그	
#30401	0.05초 토글 플래그	
#30402	0.1초 토글 플래그	
#30403	0.2초 토글 플래그	
#30404	0.5초 토글 플래그	
#30405	1초 토글 플래그	
#30406	1.5초 토글 플래그	
#30407	2초 토글 플래그	
#30408	5초 토글 플래그	
#30409	10초 토글 플래그	
<타이머 50개>		
#30450	타이머 0번	
~	<쓰기동작> 0 = 타이머리셋 -1 = 타이머 리셋 & 시작 -2 = 타이머 홀드 -3 = 타이머 재시작 0 이상값 = 의미없음. <읽기동작> 음수는 타이머명령임. 0 이상 수치값은 타이머 시간진행값 타이머 시간은 0.001초 단위	
#30499	타이머 49번	

12.24. 알람 코드

번지	설명	비고
#30500	프로세싱 알람	
#30501	명령알람	
#30502	통신상태 알람	
#30503	서보알람	
#30504	비상정지 알람	
#30505	원점복귀 알람	
#30506	리미트 신호 알람	
#30507	소프트 리미트 알람	
#30508	보조장치 통신상태 알람	
#30509	위치정보 알람	
#30510	워치독 알람	
#30511	파라메터 설정알람	
#30512	보간동작 보상제어 알람	
~		
#30549	예약	

12.25. 자동프로그램파일정보

번지	설명	비고
#30600 ~ #30679	자동프로그램 파일명(경로포함)	
#30680 ~ #30699	자동프로그램 파일명	
#30700	자동프로그램 현재 테이블번호	
#30701 ~ #30749	자동프로그램 주석문	
#30750	자동프로그램 로딩중	
#30751	자동프로그램 로딩완료	
#30760 ~	자동프로그램 테이블 동작속성	
#30760	동작 활성화	
#30761	동작 상태값	
#30762	자동 사이클 딜레이	
#30763	자동 사이클 카운터 프리셋	
#30764	설정 사이클 카운터	
#30765	현재 사이클 카운터	
#30770	자동테이블 로딩시작지령	
#30771	자동테이블실행 동작중	
#30772	자동테이블 번호 강제지령	적용 후 자동 클리어(0.0)
#30780	전체블록수	실제라인수가 아닌 해석블록수
#30781	실행블록	실행라인수이 아닌 실행블록수
#30782	실행%	
#30783	소스라인	
#30784	소스실행라인	

12.26. 조그 동작정보

번지	설명	비고
#30800	조그 동작 ENABLE	부울
#30801	조그 마스터 스피드	
#30802	조그 메인 스피드 적용 축	정수
#30803	조그 메인 스피드	
#30804	조그 X축 스피드	
#30805	조그 Y축 스피드	
#30806	조그 Z축 스피드	
#30807	조그 A축 스피드	
#30808	조그 B축 스피드	
#30809	조그 C축 스피드	
#30810	조그 이동 배율 스위치	정수 (x1, x10, x100, x1000 등의 이동배율설정)
#30811	조그 프로브 기능	부울
#30812	조그 동작모드	정수 0=스텝, 1=연속
#30813	조그 스텝이동 마스터 배율	1.0 (내부 Mpg동작에서 사용)
#30814	조그 합성 제한 주파수	100,000 (100kPPS최대)
#30815	조그 ENABLE SWITCH	부울
#30816	예약(조그 동작 마스터모드)	
#30817	조그 Z안전위치 동작기능	부울
#30818	조그 Z안전위치	
#30819	조그 Z안전위치 비교방식	정수
#30820	조그/MPG X축 증가방향 이동지시	부울
#30821	조그/MPG X축 감소방향 이동지시	부울
#30822	조그/MPG Y축 증가방향 이동지시	부울
#30823	조그/MPG Y축 감소방향 이동지시	부울
#30824	조그/MPG Z축 증가방향 이동지시	부울
#30825	조그/MPG Z축 감소방향 이동지시	부울
#30826	조그/MPG A축 증가방향 이동지시	부울
#30827	조그/MPG A축 감소방향 이동지시	부울
#30828	조그/MPG B축 증가방향 이동지시	부울
#30829	조그/MPG B축 감소방향 이동지시	부울
#30830	조그/MPG C축 증가방향 이동지시	부울
#30831	조그/MPG C축 감소방향 이동지시	부울
#30832	예약	
#30833	예약	
#30834	조그 X축 최종 스피드	마스터배율 * 조그X축스피드 (read)
#30835	조그 Y축 최종 스피드	마스터배율 * 조그Y축스피드 (read)
#30836	조그 Z축 최종 스피드	마스터배율 * 조그Z축스피드 (read)
#30837	조그 A축 최종 스피드	마스터배율 * 조그A축스피드 (read)
#30838	조그 B축 최종 스피드	마스터배율 * 조그B축스피드 (read)
#30839	조그 C축 최종 스피드	마스터배율 * 조그C축스피드 (read)

12.27. MPG 동작정보

번지	설명	비고
#30840	전체 Enable	부울
#30841	동작상태값	0 <= 40 :초기화 150 = 연속동작상태 250 = 스텝동작상태
#30842	휠동작상태	0 = 동작안함, 1 = 동작함(1초지연)
#30845	Wheel Enable	부울
#30846	Wheel Interface	정수
#30847	Wheel Counter	정수
#30848	Wheel PPS	
#30849	Wheel Speed 배수	
#30850	Axis Enable	부울
#30851	Axis Interface	정수
#30852	Axis	정수
#30853	Axis Invert	부울
#30854	Axis Sel	정수
#30855	Multiplier Enable	부울
#30856	Multiplier Interface	정수
#30857	Multiplier	정수
#30858	Multiplier Invert	부울
#30859	Multiplier Sel	정수
#30860	EStop Enable	부울
#30861	EStop Interface	정수
#30862	EStop Invert	부울
#30863	EStop Signal	정수
#30865	Enable Enable	부울
#30866	Enable Interface	부울
#30867	Enable Invert	부울
#30868	Enable Signal	정수
#30870	X축 동작방향 반전	부울
#30871	Y축 동작방향 반전	부울
#30872	Z축 동작방향 반전	부울
#30873	A축 동작방향 반전	부울
#30874	B축 동작방향 반전	부울
#30875	C축 동작방향 반전	부울

12.28. JOG 추가 정보

번지	설명	비고
#30900 ~30909	JOG 축동작 ENABLE	부울

12.29. 인자 Parameter(X,Y,Z,I,J,K...)

<인자 값 저장>

번지	설명	비고
#31000	A	
#31001	B	
#31002	C	
#31003	D	
#31004	E	
#31005	F	
#31006	G	정의안됨
#31007	H	
#31008	I	
#31009	J	
#31010	K	
#31011	L	
#31012	M	정의안됨
#31013	N	정의안됨
#31014	O	정의안됨
#31015	P	
#31016	Q	
#31017	R	
#31018	S	
#31019	T	
#31020	U	
#31021	V	
#31022	W	
#31023	X	
#31024	Y	
#31025	Z	

<어드레스 지정유무> 매크로호출시에만 유효 (0:지정없음, 1:지정됨)

번지	설명	비고
#31100	A	
#31101	B	
#31102	C	
#31103	D	
#31104	E	
#31105	F	정의안됨
#31106	G	정의안됨
#31107	H	
#31108	I	
#31109	J	
#31110	K	
#31111	L	
#31112	M	정의안됨
#31113	N	정의안됨
#31114	O	정의안됨
#31115	P	
#31116	Q	
#31117	R	
#31118	S	정의안됨
#31119	T	정의안됨
#31120	U	
#31121	V	
#31122	W	
#31123	X	
#31124	Y	
#31125	Z	

<값 지정유무> 매크로호출시에만 유효 (0:지정없음, 1:지정됨)

번지	설명	비고
#31150	A	
#31151	B	
#31152	C	
#31153	D	
#31154	E	
#31155	F	정의안됨
#31156	G	정의안됨
#31157	H	
#31158	I	
#31159	J	
#31160	K	
#31161	L	
#31162	M	정의안됨
#31163	N	정의안됨
#31164	O	정의안됨
#31165	P	
#31166	Q	
#31167	R	
#31168	S	정의안됨
#31169	T	정의안됨
#31170	U	
#31171	V	
#31172	W	
#31173	X	
#31174	Y	
#31175	Z	

12.30. 패턴출력 인자

번지	설명	비고
#31200	패턴 인자수	
#31201	패턴 최대수	
#31202	패턴 출력반전	
	패턴정보 최대 100개	
#31210	ON 패턴길이	
#31211	OFF 패턴길이	
반복		
~#31399		

12.31. 조이스틱 정보

번지	설명	비고
조이스틱1		
#31500	ENABLE	사용자 프로그램사용
#31501	INSTALL 상태	
#31505	눌러진 버튼 수	
#31506	버튼 정보	(16진수)
#31507	버튼 정보	실수
#31508	새롭게 눌러진 버튼 정보	실수
#31509	새롭게 떼어진 버튼 정보	실수
#31510	X	
#31511	Y	
#31512	Z	
#31513	U	
#31514	V	
#31515	R	
#31516	POV	
#31517	변화이전 POV	
#31520 ~ #31549	내부조이스틱드라이브 전역변수영역	
조이스틱2		
#31600	ENABLE	사용자프로그램사용
#31601	INSTALL 상태	
##31605 이하 동일 데이터 구조		

12.32. 전용 IO M코드 실행

번지	설명	비고
#31700 ~ #31799	전용 IO 출력제어	정수값 1이면 동작, 동작후 바로 0으로 설정됨.
#31703	M3	
#31704	M4	
#31705	M5	
#31708	M8	
#31709	M9	
#31710	M10	
#31711	M11	
#31712	M12	
#31713	M13	
#31714	M14	
#31715	M15	
#31738	M38	
#31739	M39	
#31768	M68	
#31769	M69	

12.33. 전용IO 상태

번지	설명	비고
#31800 ~ #31899	실시간 출력 상태	실시간 모니터링속도는 모션보드 상태 타이머 동작HZ와 동일.
#31800	(M03,M04/M05)SPINDLE ON	
#31801	(M03) SPINDLE CW	
#31802	(M04) SPINDLE CCW	
#31803	(M08/M09)COOLANT	
#31804	(M10/M11)CLAMP	
#31805	(M14/M15)BLOWER	
#31806	(M38/M39)5축 CLAMP	
#31807	(M68/M69)메인 CHUCK	
#31900 ~ #31999	실시간 입력 상태	실시간 모니터링속도는 모션보드 상태 타이머 동작HZ와 동일
#31900	프로브입력상태	

12.34. 워크좌표계 원점

번지	설명	비고
#32000 ~		읽기만 유효
#32520 ~	G52 ORG	
#32540 ~	G54 ORG	
#32550 ~	G55 ORG	
#32560 ~	G56 ORG	
#32570 ~	G57 ORG	
#32580 ~	G58 ORG	
#32590 ~	G59 ORG	
~#32999		

12.35. CNC연산옵션 (공구번호,보정,절대-상대,극,평면,사이클,...)

번지	설명	비고
#33000	현재공구번호	Txx의 (xx -1)값을 갖는다.
#33001	이전공구번호	최근 Txx명령 이전의 툴번호값
#33002	현재 D 테이블번호	
#33003	현재 H 테이블번호	
#33004	현재 D 보정값	
#33005	현재 H보정값	
#33006	좌측경보정 유무	G41,G40
#33007	우측경보정 유무	G42,G40
#33008	+장보정 유무	G43,G49
#33009	-장보정 유무	G44,G49
#33020	최종 모션 G코드	
#33021	최종 모달 모션 G코드	
#33022	최종 M코드	
#33023	인치단위계 ON표시	G20,G21
#33024	기계좌표계 ON표시	G53
#33025	워크좌표계 ON표시	G54,G55...
#33026	로컬좌표계 ON표시	G52
#33027	상대좌표 ON표시	G90,G91
#33028	극좌표 ON표시	G15,G16
#33029	스케일 ON표시	G51,G50
#33030	회전 ON표시	G68,G69
#33031	워크좌표계 번호	G54=54, G55=55,...
#33032	평면XY ON표시	G17
#33033	평면YZ ON표시	G18
#33034	평면ZA ON표시	G19
#33035	블록 스킵(BLOCK SKIP)	읽기전용
#33036 ~ #33058		
#33039	모션동작축	
#33050	M코드 실행결과리턴 0 =정상, 1=오류	
#33060	사이클동작 코드	G73,G74...,G86,G87, G80
#33061	리지드탭동작on/off	읽기전용

12.36. 공구테이블

번지	설명	비고
#33100~ #33199	D	경보정
#33200~	D.WEAR	경마모
#33300~	H	길이보정
#33400~	H.WEAR	길이마모
#33500~	특성1	
#33600~	특성2	
#33700~	특성3	
#33800~	offset.X	공구옵셋좌표
#33900~	offset.Y	
#34000~	offset.Z	
#34100~	offset.A	
#34200~	offset.B	
#34300~	offset.C	
#34400~	Pos.X	공구교환 주요 위치정보
#34500~	Pos.Y	
#34600~	Pos.Z	
#34700~	Pos.A	
#34800~	Pos.B	
#34900~	Pos.C	
#35000~	Use time (sec)	공구 사용시간
#35100~	Life time (sec)	공구 수명시간
#35200~	Use count	공수 사용횟수
#35300~	설치유무	
#35400~	존재유무(센서)	
#35500~#35599	예약	
#35600~#35999	공구특성문자열 공구당 32바이트씩 할당 4개의 연속된 시스템메모리사용 35600~35603 = 공구1 문자열	string

12.37. 모션보드 메인 상태

번지	설명	비고
#36000	모션보드 링크상태	
#36001	모션보드 동기입력(SYNC)신호상태	
#36002	모션보드 동기명령	
#36003	모션보드 대기명령	
#36004	모션보드 지연명령	
#36005	모션보드 일시정지상태	
#36006	모션보드 보간동작완료상태	
#36007	모션보드 모션동작상태	
#36008	모션보드 실행가능상태	
#36009	모션보드 알람상태	
#36020,#36021	모션명령실행id, 모션명령대기id	
#36022	준비된 모션명령여부 (연속모션시)	
#36023,#36024	현재모션 최대펄스수,실행펄스수	
#36025	현재모션 실행%	
#36026,#36027	엔진명령큐,시스템명령큐 명령수	
#36028,#36029	스캔타임,지연타임	
#36030	선속도일정제어 ON	
#36031	연속실행제어 ON	
#36032	다중보간제어 ON	
#36033	스텝실행제어 ON	
#36034	종점보상제어 ON	
#36040	ELX36 연결상태	
#36041	PIX48 #1 및 확장보드 연결상태	
#36042	PIX48 #2 및 확장보드 연결상태	
#36043	PIX48 #3 및 확장보드 연결상태	
#36044	PIX48 #4 및 확장보드 연결상태	

12.38. 모션보드 축별 동작상태

번지	설명	비고
#36100	X축동작상태	0=정지 / 1=가속 / 2=정속 /3=감속
#36101	X축동작방향	0=증가CW,1=감소CCW
#36102	X축홈동작상태	1=원점복귀동작상태
#36103	X축 홈센서상태	1=센서감지
#36104	X축 리미트 L 상태	1=센서감지
#36105	X축 리미트 H 상태	1=센서감지
#36106	X축 서보알람 상태	1=알람, 0=정상
#36107	X축 서보온 상태	1=ON, 0=OFF
#36108	X축 동작완료 상태	1=동작완료, 0=동작중
#36109	X축 알람상태	1=알람, 0=정상
#36110 ~ #36119	Y축 축별 동작상태	X축 상태값 비고참조
#36120 ~ #36129	Z축 축별 동작상태	X축 상태값 비고참조
#36130 ~ #36139	U축 축별 동작상태	X축 상태값 비고참조
#36140 ~ #36149	V축 축별 동작상태	X축 상태값 비고참조
#36150 ~ #36159	W축 축별 동작상태	X축 상태값 비고참조
#36160 ~ #36169	S1축 축별 동작상태	X축 상태값 비고참조
#36170 ~ #36179	S2축 축별 동작상태	X축 상태값 비고참조
#36180 ~ #36187	축 엔코더 카운터 정보 값	
#36190 ~ #36197	축 피드백 정보값	

12.39. PIX48 입출력

번지	설명	비고
#36200	GIN	비트구조, 읽기전용
#36201 ~ #36232	GIN.0 ~ GIN.31	읽기전용 (실수형)
#36233	GIN	실수형
#36234	입력Filter	
#36235	확장장치 상태값(실수형)	1: 연결 정상 및 키스캔동작 off 3: 연결 정상 및 키스캔동작 on
#36240 ~ #36243	ADC0 ~ ADC3	읽기전용
#36250	GOUT	비트구조, 읽기전용
#36251 ~ #36282	GOUT.0 ~ GOUT.31	읽기전용 (실수형)
#36283	GOUT	실수형
#36284	GOUT SET 명령	비트구조
#36285	GOUT CLR 명령	비트구조
#36286	GOUT TOGGLE 명령	비트구조
#36287	GOUT SET 명령	실수형
#36288	GOUT CLR 명령	실수형
#36289	GOUT TOGGLE 명령	실수형
#36290	MPG COUNTER	읽기전용 (최대 16비트 카운터값)
#36291	MPG AXIS	읽기전용
#36292	MPG MULTIPLY	읽기전용
#36293	MPG ENABLE	읽기전용
#36294	MPG ESTOP	읽기전용
#36300 ~ #36303	ENC 입력채널 0~3	(최대 32비트카운터값)
#36310	PWM0 주기값	
#36311	PWM0 듀티값	
#36312 ~ 36331	PWM1 ~ PWM15 (주기, 듀티)값 배열	
#36350	매트릭스키 코드 동작명령	0=명령없음, 1=동작ON, 2=동작OFF
#36351 ~ #36358	매트릭스키 스캔코드 (8코드)	스캔별 입력포트(8핀)값
#36359		
#36360	매트릭스 키이벤트 변환테이블 주소	0 = 이벤트사용안함, (윈도우 가상키코드 VK_xx값 사용)
#36361 ~ #36368	매트릭스 키이벤트 번호 (최대 8개)	스캔코드 번역키코드 0 ~ 64 값
#36369	스크립트 호출용 키이벤트번호	키테이블에 변환된 코드값 (키테이블변환코드값이 음수일 경우 State_PIX48_MatrixKey_Post.code 호출)
#36370 ~ #36379	예약	
#36200 ~ #36399	PIX48 #2 영역	
#36400 ~ #36699	확장장치 예약영역	

12.40. 모션보드 입출력

번지	설명	비고
<일반입출력>		
#36700	IN	비트구조, 읽기만 가능
#36705	IN	(실수값)
#36710	OUT	비트구조, 읽기만 가능
#36715	OUT	(실수값)
#36720	CLEAR OUT	비트구조, 쓰기용 해당비트가 1이면 clear output 쓰기후 자동 0으로 변화됨
#36730	SET OUT	비트구조, 쓰기용 해당비트가 0이면 set output 쓰기후 자동 0으로 변화됨
<서보입출력>		
#36740~#36747	축별 서보 입력상태	
#36750~#36757	축별 서보 출력상태	
<특수입출력>		
#36800	ADC 채널 0	
#36801	ADC 채널 1	
#36802	ADC 채널 2	
#36803	ADC 채널 3	
#36810	데이터 변환된 ADC 채널 0	WINC_Adc0.dat 파일에 의해 변환
#36811	데이터 변환된 ADC 채널 1	WINC_Adc1.dat 파일에 의해 변환
#36812	데이터 변환된 ADC 채널 2	WINC_Adc2.dat 파일에 의해 변환
#36813	데이터 변환된 ADC 채널 3	WINC_Adc3.dat 파일에 의해 변환
#36820	GIO 카운터 0	
#36821	GIO 카운터 1	
#36840	GIO 엔코더 카운터 0	
#36842	GIO 엔코더 카운터 1	
<축별 엔코더>		
#36860	각 축별 엔코더카운터	
#36870	각 축별 엔코더Z 변화 카운터	

12.41. PC 화면정보

번지	설명	비고
#37000	현재화면번호	0 = 모니터링화면 1 = 엔진구성화면 2 = CNC 운전화면 3 = CNC 구성화면 4 = 3D 프린터 구성화면 5 = CUSTOM 화면
#37001	탭화면번호	0 = 기본 설정탭 1 = 작동 탭 2 = 조그 / MPG탭
#37002	탭화면변경	0 = 변경없음 1 = 기본설정탭 2 = 작동탭 3 = 조그 / MPG탭
#37003	프로그램 포워드 활성화 유무	0 = 없음, 1 = 포워드활성화됨.
#37005	메인화면 설정제어	0 = 변경없음 1 = 모니터링화면. ... 6 = CUSTOM화면
#37010	메인프레임 SHOW 여부	실수 0=안보임, 1=보임
#37020	메인뷰 SHOW 여부	
#37030	탭 화면 SHOW 여부	
#37031	탭 화변 SHOW 제어	-1=탭화면감추기, 1=탭화면보이기
#37040	모니터링화면 SHOW 여부	
#37050	엔진구성화면 SHOW 여부	
#37060	CNC 운전화면 SHOW 여부	
#37070	CNC 구성화면 SHOW 여부	
#37080	3D프린터 구성화면 SHOW 여부	
#37090	CUSTOM 화면 SHOW 여부	
#37140	LOADING 화면 SHOW 여부	
#37150	메시지화면 SHOW 여부	
#37151	메시지화면 타임설정	정수 (0.1초단위) 10 = 1초 의미
#37160	PopUp 커스텀 화면1 SHOW여부	
#37170	PopUp 커스텀 화면2 SHOW여부	
#37180	PopUp 커스텀 화면3 SHOW여부	
#37190 ~ #37191	JOG화면위치(X,Y)	
#37199	백그라운드 UI 실행제어	0 = 디폴트실행(10Hz) 1 ~ 10 (1~10Hz실행) >999 (실행off)
#37200	백그라운드 UI 실행상태	0 = 실행안함 1 = 실행중
#37201 ~ #37219	F1~F19기능키 램프	
#37225 ~ #37229	장치구성저장 진행정보	

12.42. 드래그 앤 드롭 파일명정보

번지	설명	비고
#37300 ~#37399	드래그 앤 드롭 파일명	

12.43. PC장치 마우스 정보

번지	설명	비고
#37500	위치X	
#37501	위치Y	
#37502	L버튼상태	
#37503	R버튼상태	
#37504	M버튼상태	

12.44. PC장치 키보드 정보

번지	설명	비고
#37520	영문,숫자키코드	눌러진 코드
숫자키패드		눌러지면 1, 아니면 0
#37521	NUMLOCK 키	
#37522	/ 키	
#37523	* 키	
#37524	- 키	
#37525	+ 키	
#37526	Enter키	
#37530 ~ #37539	0 키 ~ 9키	
펑션키		
#37541	F1 키	
...	F2,F3...	
#37550	F10 키	
#37551	F11 키	
#37552	F12 키	
기타특수키		
#37560	왼쪽 Ctrl키	
#37561	오른쪽 Ctrl키	
#37562	왼쪽 Shift키	
#37563	오른쪽 Shift키	
#37564	Alt키	
#37565	Caplock키	
#37566	Tab키	
#37567	Escape키	
#37568	Backspace키	
#37569	Space키	
#37570	Insert키	
#37571	Delete키	
#37572	Home키	
#37573	End키	
#37574	PageUp키	
#37575	PageDown키	
#37576	Pause키	
#37577	Scroll Lock키	
방향키		
#37590	Left키	
#37591	Right키	
#37592	Up키	
#37593	Down키	
단축키(HOT KEY)정보		
#37595	WINC 등록 HotKey ID	
#37596	윈도우 HotKey ID	
#37597	Key 1	
#37598	Key 2	
#37700 ~ #37799	매트릭스 키보드 이벤트변환테이블1	100key

12.45. 커스텀 UI 인자전달용

번지	설명	비고
#37800~#37999	전체영역	
#37800~#37899	예약	
#37900~#37910	워크좌표설정용 UI 버퍼 (피드,RPM)	
#37900	워크설정좌표 x	
#37901	워크설정좌표 Y	
#37902	워크설정좌표 Z	
#37903	워크설정좌표 A	
#37904	워크설정좌표 B	
#37905	워크설정좌표 C	
#37906	워크설정좌표 S	
#37907	워크설정좌표 S2	
#37908	피드	
#37909	RPM	
#37920~#37929	조그동작모드 UI 버퍼	
#37920	조그메인속도저장용	UserKey_Jog_UI.code에서 사용
#37921	조그현재동작모드저장용	UserKey_Jog_UI.code에서 사용

0

12.46. CNC 설정구성

번지	설명	비고
#38000 ~ #38099	예약	
G01,G02,G03 피딩동작		
#38110	예약	
#38111	START VEL	
#38112	MAX VEL	
#38113	ACC	
#38114	DEC	
#38117	START DELAY	
G00급속이송		
#38140	예약	
#38141	START VEL	
#38142	MAX VEL	
#38143	ACC	
#38144	DEC	
#38147	START DELAY	
스핀들		
#38180 ~~#38199	예약	
속도동작 옵션		
#38200~#38205	내부연산 예약	
#38206	잔여거리 강제설정	0 혹은 음수 =사용안함
#38207	잔여거리 적용비율	0.1~1000 [%]
#38208	잔여거리 자동가감속	부울 0=적용안함, 1=적용
#38209	직선조합아크 연산방식	부울 0=평균모드, 1=피크모드
#38210	CV 경계각	
#38211	교차각 계산 적용축	2진수(0x01 = X축, 0X02 = Y축....)
#38212	교차각 선속도일정비율	
#38213	CV시작각	
#38214	원심력허용배율	
#38215	직선조합 원심력 계수	(평균타임)
#38216	모션최대명령수	
#38217	모션최대명령수 피크 오버라이딩	
#38218	속도상승필터	
#38219	속도강하필터	
사이클 동작옵션		
#38220	펙킹 거리	
#38221	공구각도	
공구경보정 옵션		
#38230	Over length limit	
이송속도 제한치	CLV OFF시	X6V3엔진F/W마다 다를수 있음.
#38300	단일축 이송속도제한 PPS	
#38301	2축 동시 이송속도제한 PPS	
#38302	3축 동시 이송속도제한 PPS	
#38303	4축 동시 이송속도제한 PPS	
#38304	5축 동시 이송속도제한 PPS	
#38305	6축 동시 이송속도제한 PPS	
#38306	7축 동시 이송속도제한 PPS	
#38307	8축 동시 이송속도제한 PPS	

번지	설명	비고
제어축 축맵핑	기계축에 대한 실제 제어기축번호	-0 이면 지정안됨(읽기전용)
#38500 ~ #38507		
#38510	사용축	정수값(읽기전용)
#38511	사용 제어기축	정수값(읽기전용)
#38520~	기계 동작하한점(소프트리미트하한)	(읽기전용)
#38530~	기계 동작상한점(소프트리미트상한)	(읽기전용)
#38540~	백래쉬	(읽기전용)
#38550~	조그속도 하한	
#38560~	조그속도 상한	
#38570~	MPG 연속 속도 상한	
#38700~	원점복귀 옵셋이동량	

12.47. 자동동작 연산참조

번지	설명	비고
#39000	연산 시작 머신좌표	
#39010	연산 종점 머신좌표	
#39020	보정연산 머신 종점 좌표	
#39040	연산 시작 워크좌표	
#39050	연산 종점 워크좌표	
#39060	현재해석 명령 모션타입	
#39061	현재해석 명령 모션코드	
#39062	현재해석 명령 이동거리	
#39063	현재해석 명령 전체 이동거리	
#39064	현재해석 명령 FEED 누적거리	
#39065	현재해석 명령 G00 누적거리	
#39066	현재해석 명령 G01 누적거리	
#39067	현재해석 명령 G02 누적거리	
#39068	현재해석 명령 G03 누적거리	
#39073~#39078	스캔해석 명령 누적거리	
#39100	현재명령ID	
#39700~39800 <예약>		

12.48. 공구교환

공구교환동작은 모션마스터의 내부적기능이 아니라, 사용자 M코드생성등의 외부 매크로기능에 의해 구현됨으로 아래에 정의된 메모리의 기능들은 사용자에 의해서 재구성되어 사용될 수 있습니다. 추후 모션마스터가 제공하는 공구교환 외부 매크로 예제의 경우 아래 정의된 기능 중심으로 메모리를 운영하게 됩니다.

번지	설명	비고
#33000	현재공구 지정번호	
#33001	이전공구 지정번호	
#33100~ #35799	공구테이블	
#34400~	공구교환용 주요위치정보 X	
#34500~	공구교환용 주요위치정보 Y	
#34600~	공구교환용 주요위치정보 Z	
#34700~	공구교환용 주요위치정보 A	
#34800~	공구교환용 주요위치정보 B	
#34900~	공구교환용 주요위치정보 C	
#39800	ATC ENABLE	
#39801	ATC TYPE	0 = XY테이블, 1 = 로터리
#39802	ATC EXTRA1 TYPE	
#39803	ATC EXTRA2 TYPE	
#39804	ATC 동작알림	0 = 동작없음(시작시), 1 = 동작중,
#39805	ATC 동작단계번호	
#39806	교환할 다음공구번호	사용자인터페이스에 필요할경우사용
#39807	현재공구복귀 동작알림	0 = 동작없음, 1 = 동작중
#39808	다음공구교환 동작알림	0 = 동작없음, 1 = 동작중
#39809	ATC 동작 알람	
#39810	공구척 잠금상태	
#39811	공구자동교환 완료상태	
#39812	공구수동조작 허가상태	
#39813	교환동작중 공구번호	
#39814	최종 리턴할 공구번호	
#39840	Z축 안전위치	
#39841	교환 동작 축	
#39842	교환 동작에 사용되지 않는 축	
#39843	교환 동작 스텝수	
#39844	교환 동작 현재스텝번호	
#39845	교환 동작 속도	분당 이송FEED
#39846	교환 안정 대기시간	초단위
#39847	교환 안정 대기시간2	
#39848	교환안정 대기알람시간	
#39850	공구교환 동작 변위정보1	
#39860	공구교환 동작 변위정보2	
#39870	공구교환 동작 변위정보3	
#39880	공구교환 동작 변위정보4	
#39890	공구교환 동작 변위정보5	

12.49. 공구측정

공구측정동작 역시 모션마스터의 내부적기능이 아니라, 사용자 M코드 생성등의 외부 매크로기능에 의해 구현됨으로 아래에 정의된 메모리의 기능들은 사용자에 의해서 재구성되어 사용될 수 있습니다. 추후 모션마스터가 제공하는 공구측정 외부 매크로 예제들의 경우 아래 정의된 기능 중심으로 메모리를 운영하게 됩니다.

번지	설명	비고
#39950	공구자동측정 ENABLE	
#39954	공구측정 동작상태	0 = 동작없음, 1=동작중
#39955	공구자동 측정값반환	
#39956	측성센서 측정상태	
#39959	공구측정알람발생	측정허용공차범위 벗어났음.
#39960	공구측정 센서번호	프로브 전용IO지정으로 결정
#39961	공구측정 클린출력 IO번호	1~32번까지 (음수는 신호반전) 0은 사용안함
#39969	공구측정센서 기준 옵셋	
#39970	공구측정센서 기준위치	
#39971	공구측정 범위(공차)	
#39972	사용자 공구측정 알람공차1	
#39973	사용자 공구측정 알람공차2	
#39974	사용자 공구측정 알람공차3	
#39975	공구측정속도	
#39980 ~ #39985	공구측정용 좌표 X,Y,Z.A,B,C	측정 안전대기좌표
#39990 ~ #39995	측정 정지 기계좌표값 반환 X,Y,Z,A,B,C	

12.50. MDI 스트링 영역

번지	설명	비고
#40000~#40099	MDI 스트링 영역(800바이트)	

12.51. 로그 파일명

번지	설명	비고
#40100~#40199	현재로그파일명	
#40200~#40299	이전로그파일명	
#40300	현재로그파일번호	

12.52. 시간 스트링 영역

번지	설명	비고
#40310	현재일자/현재 시간 표시방식	=0 > 2017.05.25 (일자) 02:30:12 (24시간) = 1> 2017Y 05M 25D 02h 30m 12s (24시간) = 2> 2017년 05월 25일 02시 30분 12초 (24시간) = 10,11,12 > 각, 0,1,2, 의 12시간 표시방식 (AM/PM, 오전/오후)
#40311	프로그램 사용 시간 표시방식	
#40312	자동동작 시간 표시방식	
#40318	예측 잔여시간 표시방식	
#40320~#40324	현재 일자 스트링	
#40325~#40329	현재 시간 스트링	
#40330~#40339	프로그램 사용시간 스트링	
#40340~#40349	자동동작 사용시간 스트링	
#40370~#40379		
~#40399	예약	

12.53. 자동작업테이블 영역

번지	설명	비고
#41000~#46000		

12.54. 3D GRAPH 사용자그리기 속성영역

번지	설명	비고
#60000 ~60099	<3D DRAW GROUP 0번>	100개의 데이터가 하나의 사용자 3D 그리기 속성그룹 이며 최대 40개의 그룹을 갖는다.
#60000	ENABLE 0=그리기 없음; 1 = 메인창, 2 = 커스텀창 4 = 1번째팝업창, 8 = 2번째팝업창	1,2,4,8 을 조합해서 그리기창의 OR 가능
#60001	그리기 방식 0 = 일반 3D 그리기	
#60002	사용자 그리기 메모리 시작번지	
#60003	사용자 그리기 메모리 종료번지	
#60004	예약	
#60005	부모 GROUP (0이면 없음) (단, 자신보다 낮은 그룹번호사용)	로봇관절과 같은 링크형태의 드로잉 표현시 사용
#60006	예약	
#60007	예약	
#60008	원호그리기 분할 각도	0 이하이면 기본각
#60009	원호그리기 분할 수	0 이하이면 기본분할수
<기준옵셋속성>		
#60010	#60010 = 기준좌표 X	
	#60011 = 기준좌표 Y	
	#60012 = 기준좌표 Z	
<이동속성>		
#60015	#60015 = 이동 X	
	#60016 = 이동 Y	
	#60017 = 이동 Z	
<임의축회전#1>		
#60020	#60020 = 회전각도	
	#60021 = 회전축벡터X	
	#60022 = 회전축벡터Y	
	#60023 = 회전축벡터Z	
<임의축회전#2>		
#60025	#60025 = 회전각도	
	#60026 = 회전축벡터X	
	#60027 = 회전축벡터Y	
	#60028 = 회전축벡터Z	
<임의축회전#3>		
#60030	#60030 = 회전각도	
	#60031 = 회전축벡터X	
	#60032 = 회전축벡터Y	
	#60033 = 회전축벡터Z	
<스케일>		
#60035	#60035 = X축 스케일	
	#60036 = Y축 스케일	
	#60037 = Z축 스케일	
<변환행렬>		
#60040~#60055	현재 모델변환행렬 [4x4]	
#60060~#60075	최종 모델변환행렬 [4x4]	
#60080~#60089	STL메모리할당 영역	
#60090~#60099	예약	
#60100 ~ #63999	그룹 1 ~ 그룹 39 예약	

※3D변환순서는 아래와 같습니다.

스케일변환 ▶임의축 회전변환(#1→#2→#3) ▶(기준옵셋+이동)변환 ▶그리기코드변환

12.55. 메인 3D 그래프 기본속성 영역

번지	설명	비고
#64000	공구 표현 연산 설정	0=없음, 1=아래 연산조건 따름.
#64005~#64007	공구 옵셋이동량 X,Y,Z	
#64010~#64012	공구 제1회전축 X,Y,Z	
#64013	공구 제1회전각 맵핑축	0=없음, 1=X,2=Y,3=Z,...
#64014	공구 제1회전각 배율	
#64015	공구 제1회전각 옵셋	
#64020~#64022	공구 제2회전축 X,Y,Z	
#64023	공구 제2회전각 맵핑축	0=없음, 1=X,2=Y,3=Z,...
#64024	공구 제2회전각 배율	
#64025	공구 제2회전각 옵셋	
#64030~#64032	공구 제3회전축 X,Y,Z	
#64033	공구 제3회전각 맵핑축	0=없음, 1=X,2=Y,3=Z,...
#64034	공구 제3회전각 배율	
#64035	공구 제3회전각 옵셋	
#64040~#64042	공구 스케일 X,Y,Z	
#64050~#64053	G00 칼라속성 R,G,B,A	A=0이면 그리기 없음
#64055~#64058	G01 칼라속성 R,G,B,A	A=0이면 그리기 없음
#64060~#64062	G02 칼라속성 R,G,B	A기능 없음
#64065~#64067	G03 칼라속성 R,G,B	A기능 없음

12.56. 3D GRAPH 사용자그리기 영역

#65000~#69999 영역의 5000개의 데이터메모리 영역은 앞장의 “3D GRPAH 사용자그리기 속성영역”에 의해서 그려질 내용에 대한 명령들을 기록할 수 있는 공간입니다. 물론 사용자3D그리기 기능을 사용하지 않는다면 임의적인 기억 공간으로 사용할 수도 있습니다. 아래에 "3D GRAPH 사용자그리기 속성영역“과 연관되어 이 영역의 어떻게 사용되며, 기본적인 그리기 명령은 무엇인지 확인해 봅니다. (일반적인 3D그리기명령은 OPENGL 기본함수에 기반된 동작을 수행합니다.) 보다 자세한 그리기 명령의 사용예는 별도의 매뉴얼을 참조하실 수 있습니다.

먼저 특정 그리기그룹(1~50)의 속성영역에서 그리기 시작번지와 그리기 종료변지 속성을 #65000~#69999 번지이내로 지정합니다. 이때 종료번지는 시작번지보다 같거나 커야합니다.
아래에 그리기그룹(1) 에 대해서 파라메터를 설정하고 그리기 영역을 #65000~#65100으로 지정한 것으로 가정할 때 메모리 내용을 임의로 설정하여 살펴봅니다.

<사용자그리기 그룹1 영역>

번지	값	설명
#60000	1	메인모션창에 사용자그리기 표현
#60001	0	그리기모드 0
#60002	65000	그리기명령시작번지
#60003	65099	그리기명령종료번지

<사용자그리기 예> 간단한 직선그리기

번지	값	명령 혹은 인자	설명
#65000	1	그리기시작(NOP)	명령
#65001	2	칼라설정	명령
#65002	1.0	Red	파라메터>0.0~1.0 사이값
#65003	1.0	Green	파라메터>0.0~1.0 사이값
#65004	1.0	Blue	파라메터>0.0~1.0 사이값
#65005	21	OPENGL 라인그리기	명령>
#65006	0.0	라인그리기시작X	파라메터1> 시작점 X
#65007	0.0	라인그리기시작Y	파라메터2> 시작점 Y
#65008	0.0	라인그리기시작Z	파라메터3> 시작점 Z
#65009	100.0	라인그리기종료X	파라메터4> 종점 X
#65010	100.0	라인그리기종료Y	파라메터5> 종점 Y
#65011	100.0	라인그리기종료Z	파라메터6> 종점 Z
#65012	0	그리기 종료	명령>

<기본 그리기명령 종류>

- 기초 및 OPENGL기반 명령셋-.

명령	코드	파라메터	설명
Start	1	-	무효명령(NOP)
End	0	-	그리기 종료명령
Color	2	R,G,B	색 지정명령
Load Identity	10	-	기초행렬로딩
Push Matrix	11	-	현재모델뷰행렬 스택에 푸쉬
Pop Matrix	12	-	현재모델뷰행렬 스택에서 팝
Load Matrix	13	[4x4] 행렬	현재모델뷰행렬 로딩
Multiple Matrix	14	[4x4] 행렬	현재모델뷰행렬과의 곱
Translation	15	이동 [X,Y,Z]	이동변환
Rotate	16	회전축[X,Y,Z], 회전각 Angle	회전변환
Scale	17	스케일[X,Y,Z]	스케일변환
Line	21	시작정점[X,Y,Z],종료정점 [X,Y,Z]	선분 그리기
Line Strip	22	정점수 N / 정점배열 XYZ[]	연속 선분 그리기
Triangle	23	정점3개 [X1,Y1,Z1],[X2,Y2,Z2],[X3,Y3,Z3]	삼각형 그리기
Triangle Strip	24	정점수 N, 정점배열 XYZ[]	연속 삼각형 그리기
Triangle Fan	25	정점수 N, 정점배열 XYZ[]	확산 삼각형 그리기
Arc	26	중심정점 [X,Y,Z] 반경r, 시작각도A1, 아크각도 A2 세그먼트 수 M	원호 그리기
Solid Arc	27	중심정점 X,Y,Z 반경r, 시작각도 A1, 아크각도 A2 세그먼트 수 M	채워진 원호 그리기
Text	45	문자열 수 N 문자열(“아스키코드”)	문자열 그리기

- STL 파일그리기 명령셋 -

명령	코드	파라메터	설명
Load STL	50	STL파일이 로딩된 내부메모리포인터를 저장하고 있는 시스템메모리위치	p=load_file_alloc(“파일명”)함수에 의해 할당받은 메모리포인터 p를 저장하고 있는 시스템메모리위치

- G코드형식의 그리기 명령셋-

명령	코드	파라메터	설명
MoveTo	100	이동위치 X,Y,Z	현재 위치(CP) 이동명령
LineTo	101	직선 종료위치 X,Y,Z	직선변위 그리기기
ArcCW	102	원호평면 A 원호중심 X,Y,Z 원호종점 X,Y,Z	원호평면 A> 0=XY평면, 1=ZX평면, 2=YZ평면 원호그리기 세그먼트수는 #60009의 원호그리기분할수값으로 결정
ArcCCW	103	원호평면 A 원호중심 X,Y,Z 원호종점 X,Y,Z	원호평면 A> 0=XY평면, 1=ZX평면, 2=YZ평면 원호그리기 세그먼트수는 #60009의 원호그리기분할수값으로 결정

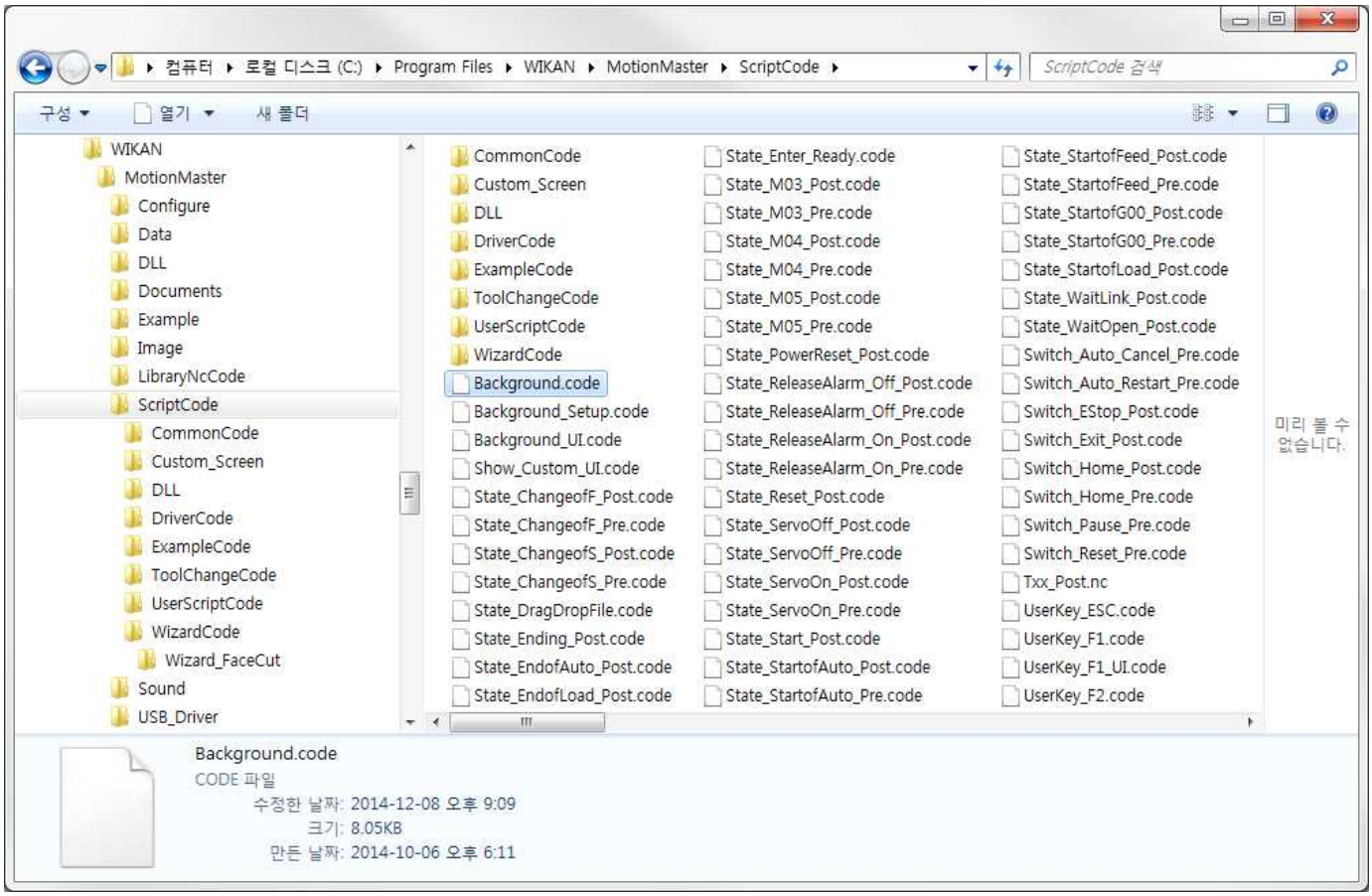
13. 사용자 자동화 기능

모션마스터는 다양한 방식의 사용자 커스터마이징 기능을 제공합니다. 모션마스터에서 제공되는 사용자 커스터마이징은 전체적인 화면을 재구성하여 완전히 새롭게 만드는 방식 뿐만 아니라 CNC운전 화면을 그대로 사용하면서 사용자의 동작기능을 추가하는 비교적 단순한 방식도 가능합니다. 또한 및 사용자전용의 팝업윈도우를 만들어 실행하는 방식도 지원합니다. 아래의 3가지 방법을 통해서 모션마스터의 커스터마이징을 구현할 수 있습니다.

(1) CNC 운영과 직접적으로 연관된 스크립트 코드(파일)의 변경
(2) 사용자가 직접 작성할 수 있는 팝업창 생성기능(위저드기능)
(3) 사용자전용의 메인화면을 생성할 수 있는 CUSTOM화면 기능

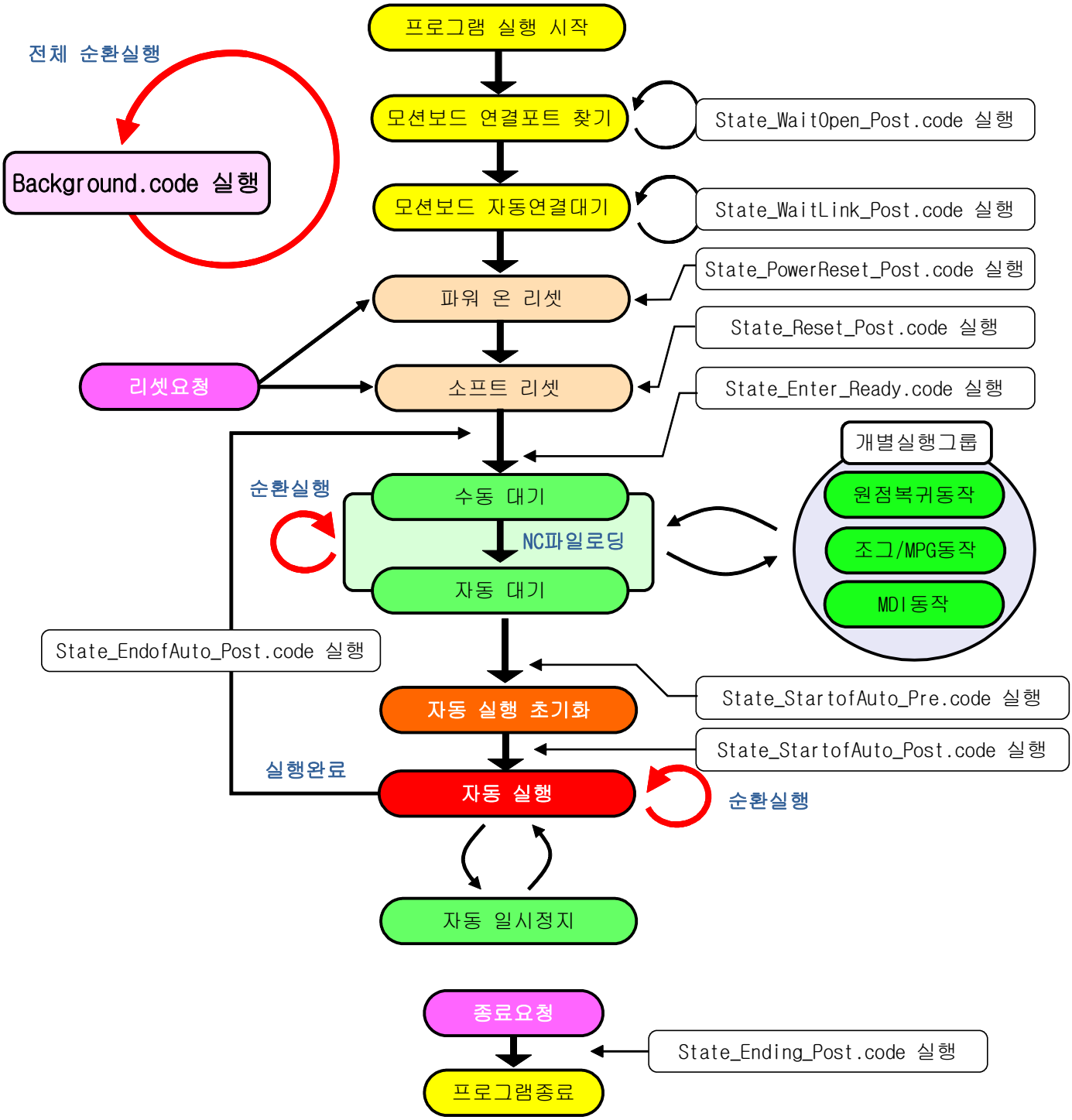
사용자 자동화기능은 모션마스터에 내장된 실시간 실행 V스크립터 인터프리터에 의해 구현됩니다. 실시간으로 텍스트파일을 한줄 씩 읽어 처리해 내는 간단한 명령처리기인 V스크립트 인터프리터는 CNC운영의 전반에 사용되어 기계동작상태와 버튼등의 상태변화시에 내부적으로 지정된 스크립트파일을 읽어들여 실행하게 됩니다. 스크립트파일 대부분은 내용을 갖지 않으며 사용자에 의해 프로그램 될 수 있습니다. 모션마스터를 사용자 기계시스템에 맞게 변경하기 위해서는 모션마스터가 다양한 외부 스크립트파일들과 어떠한 방식으로 서로 관계되는지 알 필요가 있습니다. 외부스크립터파일은 프로그램 설치폴더의 ScriptCodeW라는 하위폴더내에 포함되어 있습니다. ScriptCodeW폴더는 관리를 위해 다시 여러개의 하위폴더로 나누어집니다. 또한 사용자의 전용코드가 추가될 때 새로운 하위폴더를 생성하는 것이 전체 스크립터코드파일들을 관리하는데 편리할 것입니다.

<ScriptCode 폴더구성>



13.1. 모션마스터 CNC운영 전체흐름과 관련된 외부스크립터파일

아래에는 모션마스터프로그램의 전체제어흐름과 관련되어 실행되는 연관 스크립터파일을 표현하고 있습니다.
특히 주목할 만한 부분은 전체순환실행 속에서 실행되는 Backgournd.code 스크립트파일입니다. V스크립터언어는 PLC장치와 같은 입출력상태제어와 PLC타이머 등의 함수를 포함하고 있으며 직렬통신과 이더넷통신 함수가 지원되므로 다양한 주변장치 제어프로그램을 작성하여 실행시킬 수 있습니다.



13.2. 제어입력 혹은 제어명령 변화와 관련된 스크립트파일 요약

<표> 제어스위치 및 버튼입력 혹은 제어명령에 의한 제어상태변화로 실행되는 스크립트 실행파일 목록

실행이벤트	관련 스크립트파일
재시작 명령발생(RESTART)	Switch_Auto_Restart_Pre.code
일시정지 명령발생(PAUSE/STOP)	Switch_Pause_Pre.code
자동 취소 명령발생(CANCEL)	Switch_Auto_Cancel_Pre.code
원점복귀 명령발생(HOME)	Switch_Home_Pre.code (원점복귀명령전) Switch_Home_Post.code (원점복귀명령완료후)
리셋 명령발생(RESET)	Switch_Reset_Pre.code
비상정지 명령발생(ESTOP)	Switch_EStop_Post.code
프로그램 종료명령발생(EXIT)	Switch_Exit_Post.code
사용자 기능키입력 발생	UserKey_Fx.code (일반적으로 제어코드 실행부분) UserKey_Fx_UI.code (일반적으로 화면코드 실행부분) (x는 1~8까지)
ESC키 입력발생	UserKey_ESC.code
사용자 키에 의한 M코드실행	UserKey_Mxx.code (일반적으로 제어코드 실행부분) UserKey_Mxx_UI.code (일반적으로 화면코드 실행부분) (xx는 정수값, M3,M4,M5,M8M9,M10M11,M14M15,M38M39)
조그운전모드전환키 입력발생	UserKey_Jog_UI.code
동작탭전환키 입력발생	UserKey_OpTab_UI.code
공구교환 메뉴 입력키 발생	UserKey_ToolChangeMain.code (일반적으로 제어코드 실행부분) UserKey_ToolChangeMain_UI.code (일반적으로 화면코드 실행부분)
서보 ON 명령	State_ServoOn_Pre.code / State_ServoOn_Post.code
서보 OFF 명령	State_ServoOff_Pre.code / State_ServoOff_Post.code
자동시작명령	State_StartofAuto_Pre.code / State_StartofAuto_Post.code
절삭시작명령	State_StartofFeed_Pre.code / State_StartofFeed_Post.code
급속이송시작명령	State_StartofG00_Pre.code / State_StartofG00_Post.code
NC파일 로딩명령	State_StartofLoad_Post.code
스핀들 ON/OFF 명령	State_M03_Pre.code / State_M03_Post.code State_M04_Pre.code / State_M04_Post.code State_M05_Pre.code / State_M05_Post.code
Feed 변화명령	State_ChangeofF_Pre.code / State_ChangeofF_Post.code
RPM 변화명령	State_ChangeofS_Pre.code / State_ChangeofS_Post.code
드래그 앤 드롭 발생	State_DragDropFile.code

※모션 마스터 프로그램 버전에 따라 실행스크립트파일 목록이 추가될 수 있으며, 일부 스크립트파일에 대해서 실행되지 않을 수도 있습니다.

13.3. 화면제어와 관련된 스크립트파일

일반적으로 UI로 끝나는 스크립트 파일은 화면제어명령함수가 포함된 스크립트파일입니다.
화면제어함수는 질의메세지 함수와 같이 실행을 홀딩하고 대기하는 명령이 존재할 수 있으므로 CNC 주동작에 영향을 미칠 가능성이 있어서 분리되었습니다. UI로 끝나는 스크립트파일들은 실제 실행하는 주체가 CNC 주동작코어제부가 아닌 화면제어부분에서 실행되어 사용자의 실수에도 비교적 안전할 수 있습니다. 다만, CNC 시스템 메모리를 액세스할 경우 CNC 주동작처리부와 크리티컬섹션 문제(공유자원에 대한 읽기/쓰기문제)를 발생시킬 수 있습니다. 그러나 CNC 주동작처리와 관련된 시스템메모리를 단순히 화면상에 표시하는 정도의 기능이다면 별 문제가 되지 않습니다.

우선 화면을 제어하는 스크립트파일이라고 특별히 다른 코드의 종류는 아니며 다 같은 동일한 스크립터 실행파일입니다. 다만 모션마스터프로그램의 사용자 화면제어 버튼에 의해 동작되는 스크립터 파일명을 따로 정해둔 것 뿐입니다. 다음의 화면을 생성하는 코드들은 스크립터가 실행되는 곳 어디에서나 사용가능한 코드입니다. 그러므로 NC코드파일내에서도 스크립터코드를 삽입하여 팝업 윈도우를 생성하고 메뉴입력을 대기하는 동작도 가능합니다.

여기서는 동작탭(설정탭)의 [CUSTOM]화면(사용자정의화면)버튼 입력에 의해 실행되는 파일과 내용을 간단하게 살펴봅니다.

<실행파일명> Show_Custom_UI.code

Show_Custom_UI.code 내용 중 일부

```
... 생략

VAR ScreenType:#30165;
ScreenType = 1;

... 생략

if( ScreenType == 0 )
{
        //-----------------------------
        // 800x600 밀링타입 001 화면
        //-----------------------------
        call("Custom_ScreenWWScreen_800x600_Mill_001.code");
}
else if( ScreenType == 1 )
{
        //-----------------------------
        // 1024x768 밀링타입 001 화면
        //-----------------------------
        call("Custom_ScreenWWScreen_1024x768_Mill_001.code");
}
... 생략
```

ScreenType 이라는 시스템메모리변수(30165번지)의 값에 따라 Custom_ScreenW하위폴더에 있는 화면그리기 스크립트코드파일(Screen_800x600_Mill_001.code 혹은 Screen_1024x768_Mill_001.code)을 호출하여 실행하는 코드로 구성되어 있습니다.

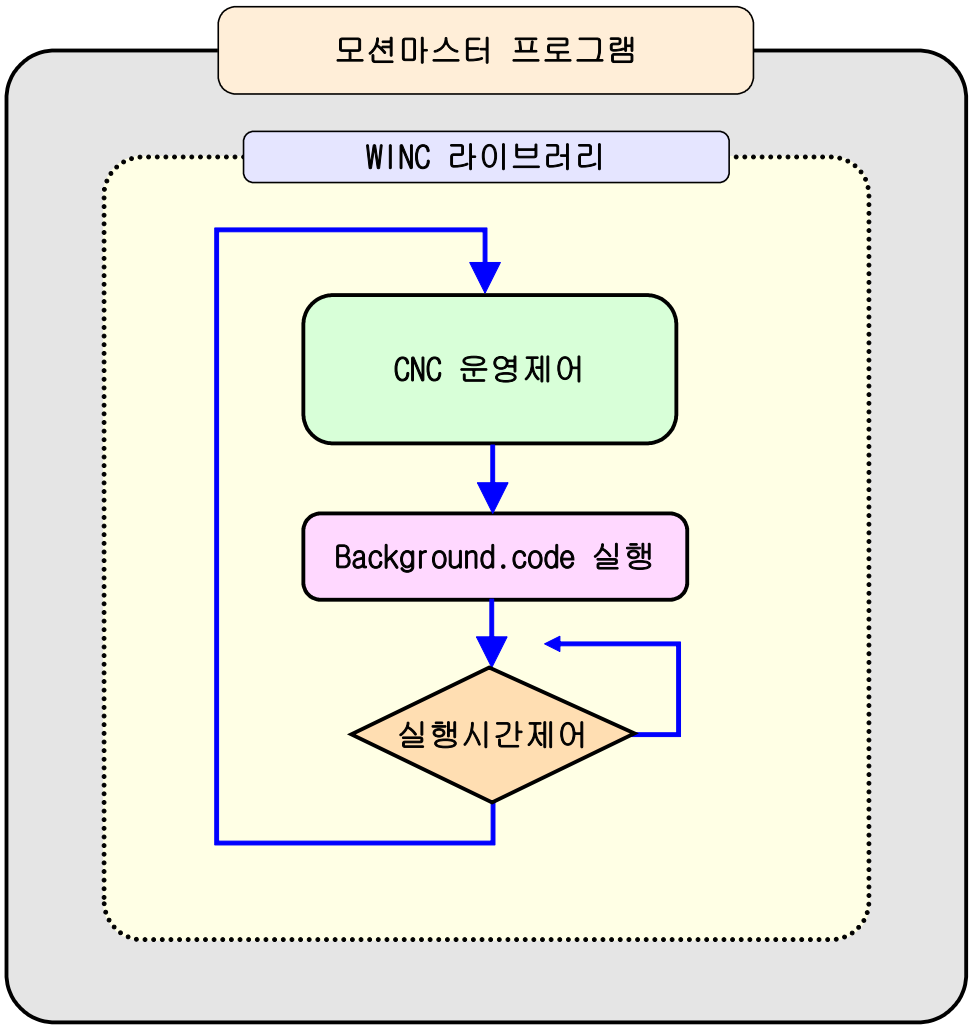
※ 화면을 생성하는 스크립트함수에 대한 자세한 설명은 별도의 “V스크립터 매뉴얼”을 참조하세요.

13.4. 배경제어 스크립트 파일 (Background.code)

Background.code 스크립트 파일은 배경제어용 코드파일로써 모션마스터프로그램이 모션보드와 연결된 이후 부터 항상 배경에서 실시간으로 실행되는 파일입니다. 실시간 실행주기는 시스템메모리 30119번지(메인쓰레드 실시간 처리수)와 Background.code내의 looptime()함수에 의해서 조절됩니다.

모션마스터프로그램의 CNC운영을 맡고 있는 부분을 내부적으로 WINC(위엔씨)라 불리는 내부 응용 라이브러리입니다. WINC는 기본적으로 독립된 쓰레드에 의해 동작되며 지속적으로 순환루프를 돌며 CNC 머신상태에서 따라서 CNC 동작제어를 수행합니다. 이 순환 루프속에서 Background.code 스크립트 파일이 실행되게 됩니다. 이 순환루프의 실행시간을 조절하는 시스템메모리영역이 30118 및 30119번지입니다. 30118번지는 초당 최대 순환처리횟수를 지정하고 30119번지는 실시간 응답을 유지해야하는 순환처리횟수를 설정합니다. 일반적으로 30118값은 30119값의 정수배로 지정합니다. 이러한 메인쓰레드의 실행속도는 내부판단에 의해 임의적으로 달라질 수 있으므로 절대적인 것은 아닙니다.

아래에는 WINC 내부응용라이브러리가 CNC운영제어와 Background.code의 실행을 순환처리 하는 내용을 그림으로 간략히 표현합니다.



모션마스터프로그램내에서 실시간으로 실행되는 Background.code 파일은 기본적으로 사용자가 임의적으로 응용할 수 있는 몇가지 예제코드들이 비활성화 상태로 포함되어 있습니다. 이들 예제는 PLC제어함수의 응용, 입력포트에 의한 조그동작 예제, 속도오버라이딩 예제 등 프로그램버전에 따라 추가될 수 있습니다. 이들은 if문을 통해서 아래와 같이 구성되어져 있습니다.

```
/*
        코드예제 설명문
*/
if(0)
{
        //예제 코드 리스트
        ....
}
```

if(0) 코드에 의해 비활성화 되었습니다.
if(1) 로 편집하게 되면 중괄호 { ... } 내부 스크립트문장들이 실행하게 됩니다.

다음은 Background.code 파일에서 주요한 기본코드구조를 살펴봅니다.

```
/*-------------------------------------------------
        PLC운용 시작 처리부 (항상 필요한 부분)
-------------------------------------------------*/
read_port();    // 입력버퍼링
looptime(0.02); // 0.02초마다 실행

/*-------------------------------------------------
        사용자 프로그램 영역 시작 ~
-------------------------------------------------*/

/*-------------------------------------------------
        ~ 사용자 프로그램 영역 종료
-------------------------------------------------*/
/*-------------------------------------------------
        PLC운영 종료처리부 (항상 필요한 부분)
-------------------------------------------------*/
run_M();        // M메모리 운영
run_T();        // PLC타이머 운영
write_port();   // 출력버퍼링
return (0);
```

항상 코드의 시작부분에 있어 사용자응용코드와 예제코드의 동작을 도와주는 코드부분입니다.

read_port()함수는 모션마스터프로그램과 연결된 장치들의 입력신호를 버퍼링하는 함수입니다.

looptime()함수는 Background.code의 실행시간주기를 설정하는 함수입니다.

사용자 프로그램코드나 예제코드가 작성되는 영역입니다.

항상 마지막에 작성되는 코드로써 PLC타이머명령, IO입출력 등의 운영처리를 담당하는 함수로 구성됩니다.
마지막에 스크립터 1회 실행완료를 알리는 return()함수로 끝납니다.

상기 코드구성에서 보듯이 Backgound.code 파일의 내부는 크게 초기화시작코드와 사용자작성코드, 그리고 종료코드로 구성됩니다. 사용자는 사용자작성코드 영역내에서 임의로 프로그램 할 수 있습니다.

Background.code 파일이 편집되어 수정되면 CNC운영화면의 [리셋]버튼을 눌러 주면 리셋 초기화 단계에서 Background.code 파일을 다시 로딩하여 수정된 내용을 실행하게 됩니다.
리셋 후 처음 Background.code 내용이 실행되기 전에 먼저 Background_Setup.code라는 스크립터파일을 찾아서 한번

실행하게 됩니다. Background_Setup.code실행이 완료되면 이후부터 실행시간에 맞춰 Background.code 파일을 순환해서 실행하게 됩니다.

아래에 Background_Setup.code 스크립터파일 내용을 보여줍니다.
별다른 코드는 없으며 단순히 PLC타이머 0번을 1초로 초기화 시작하는 코드가 있습니다. 이 내용은 사용자의 코드에 맞게 추가되거나 변경시킬 수 있습니다.

```
// Background_Setup.code 파일은
// Motion Master 프로그램이 실행될 때 단 한번만 실행됩니다.

// 0번 타이머 시작
startT(0, 1.0);
```

부록I. 지원 NC코드 요약

A. G코드

<기본동작>

코드	기능	예제	비고
G00	급속이송	G00 X100 Y100	
G01	직선보간	G01 X100 Y100	
G02	CW원보호간	G02 X100 Y100 I50 J0 G02 X100 Y100 R50	
G03	CCW원호보간	G03 X100 Y100 I50 J0 G03 X100 Y100 R50	
G04	드웰 혹은 EXACT STOP	G04 X1.0 ; 1초 휴식 G04 P1000 ; 1초 휴식 G04 ; EXACT STOP기능	

<데이터테이블입력>

코드	기능	예제	비고
G10	데이터 입력 L=2 : 워크좌표계테이블 L=10 : 공구테이블 경,장 값 L=11 : 공구테이블 경,장 마모값 P= 테이블 인덱스	G10 L2 P1 X100 Y100 Z100 ; G54좌표계입력 G10 L2 P1 XYZ : G54 워크좌표계원점을 현재 최종위치로 설정 G10 L2 P2 X100 Y100 Z100 ; G55좌표계입력 G10 L10 P1 D20 H30 ; 1번공구경 20, 길이30 으로 설정 G10 L11 P1 D0.1 H0.1 ; 1번 공구경마모 0.1, 길이마모 0.1 로 설정	L은 데이터 테이블 참조값 P는 테이블 인덱스 참조값

<좌표계 및 단위선택>

코드	기능	예제	비고
G15	극좌표 취소		
G16	극좌표 설정		
G17	XY평면선택		
G18	ZX평면선택		
G19	YZ평면선택		
G20	inch단위계입력		
G21	mm단위계입력		

<원점 복귀>

코드	기능	예제	비고
G27	원점복귀	G28참조	알람발생형
G28	원점복귀	G28 X10 Y10 ; 10,10로 급송이동 후 X,Y축 동시원점복귀 G28 X ; X축만 원점복귀 G28 ; 모든 축 원점복귀	
G30	제1,제2 원점복귀	G30 P2	제 2원점복귀

<공구측정 및 보정>

코드	기능	예제	비고
G37	공구측정	G37 Z-200 H3 P100 F300 : 현재위치에서 Z축으로 최대 -200mm까지 300 FEED로 움직이면서 probe센서를 감지함. 감지 후 정지하고 계산결과값을 H3 테이블에 기록, 그리고 원래위치로 복귀(단일 축에서만 적용) ========== 변경 R184> G37 Z-10 P0 L5 Q0 H3 ; 현재위치에서 Z축으로 최대 -15mm(Z-10과 L5) 까지 상대이동하면서 프로브신호가 발생되면 정지하고, 원래위치로 복귀. ; Z-10mm를 기준으로 여유값 +-L5mm 이내에 프로브가 감지되지 않으면 알람 메모리설정 (축지정은 X,Y,Z,A,B,C 중 하나만 지정) ; L값을 지정하지 않으면 최대 -10mm까지만 이동하며 프로브가 감지되지 않아도 오류발생없음. ; 최종결과값은 ((기준위치 Z지정값으로부터 변위) + P값(옵셋)) ; L값이 음수이면 최종계산값 부호반전 ; Q값이 0 이면 H[3]테이블에 최종결과값 기록 ; Q값이 1이면 H.Wear[3]테이블에 최종결과값 기록 ; Q값이 2이면 현재 워크좌표계의 동작축 원점설정 * 특징 G90,G91, 공작물좌표계, 및 공구장보정과 무관하게 기계좌표로 동작됨.	(1) 연산결과 = Z기준변위 + P옵셋 (2) 연산결과값은 시스템메모리 #39955 에 저장 (3) 최종 센서가 감지된 기계좌표는 시스템메모리 #39990 ~ #39995 에 저장 (4) 센서감지영역에서 센서신호가 발견되지 않으면 시스템메모리 #39959 에 1이 저장됩니다. 정상작동에서는 0 이 저장됩니다.
G43	+공구길이보정	G43 H2	
G44	-공구길이보정	G44 H2	
G49	공구길이 보정취소		

< 스케일 및 회전>

코드	기능	예제	비고
G50	스케일취소		
G51	스케일	G51 X100Y100 P5.0 ; 100,100위치를 기준으로 x5배율 G51 X100Y100 I-1.0 J-2.0 ; 100,100위치를 기준으로 X는 -1배율, Y는 -2배율	
G68	회전	G68 X100Y100 R45 ; 100,100위치를 기준으로 45도 회전	
G69	회전취소		

<작업 좌표계 설정>

코드	기능	예제	비고
G52	로컬좌표계	G53 G52 X100 Y100 ; 기계좌표계 100,100위치를 영점으로 설정 G54 G52 X100 Y100 ; G54공작물좌표계 100,100위치를 영점으로 설정	기계좌표계,공작물좌표계모두에 적용
G53	기계좌표계지정		
G54~G59	공작물좌표계 지정		

<피드 절삭모드>

코드	기능	예제	비고
G61/G62 G63/G64	절삭모드 지정	G61 ; Exact Cutting 모드 / G62 : Soft Cutting 모드 G63 : Hard Cutting 모드 / G64 : Normal Cutting모드	

<사이클 동작>

코드	기능	예제	비고
G73	고속펙킹사이클	사이클동작 부록을 참조하세요.	
G74	역탭핑사이클	사이클동작 부록을 참조하세요.	
G76	정밀보링사이클	사이클동작 부록을 참조하세요.	
G80	고정사이클취소	사이클동작 부록을 참조하세요.	
G81	드릴 사이클	사이클동작 부록을 참조하세요.	
G82	카운터보어 사이클	사이클동작 부록을 참조하세요.	
G84	심공드릴 사이클	사이클동작 부록을 참조하세요.	
G84	탭핑 사이클	사이클동작 부록을 참조하세요.	
G85	보링 사이클	사이클동작 부록을 참조하세요.	
G86	보링 사이클2 (황삭)	사이클동작 부록을 참조하세요.	
G87	역보링 사이클	사이클동작 부록을 참조하세요.	
G89	드웰적용 G85	사이클동작 부록을 참조하세요.	
G98	가공초기점복귀	사이클동작 부록을 참조하세요.	
G99	가공시작점복귀	사이클동작 부록을 참조하세요.	

<절대/상대좌표기입 설정>

코드	기능	예제	비고
G90	절대좌표기입		
G91	상대좌표기입		

B. M코드

1. 기본 M코드

코드	기능	부가설명	비고
M00	프로그램 실행 일시중지		
M01	선택적 프로그램 일시중지		
M02	프로그램 종료		
M30	프로그램 종료		
M03	스핀들 CW회전		
M04	스핀들 CCW회전		
M05	스핀들 정지		
M08	절삭유공급		
M09	절삭유차단		
M10	클램프ON		
M11	클래프OFF		
M14	스핀들 블로워(클린)ON		
M15	스핀들 블로워(클린)OFF		
M17	서보 ON		
M18	서보 OFF		
M38	5축 클램프 ON		
M39	5축 클램프 OFF		
M68	메인 척 닫힘		
M69	메인 척 열림		
M29	리지드 탭		G80으로 해제
M98	서브루틴 호출	M98 P1000 L3 ; 01000번으로 지정된 서브프로그램을 3번 호출합니다. 서브프로그램은 하나의 파일내부에 존재해야합니다.	
M99	서브루틴 복귀(리턴)		

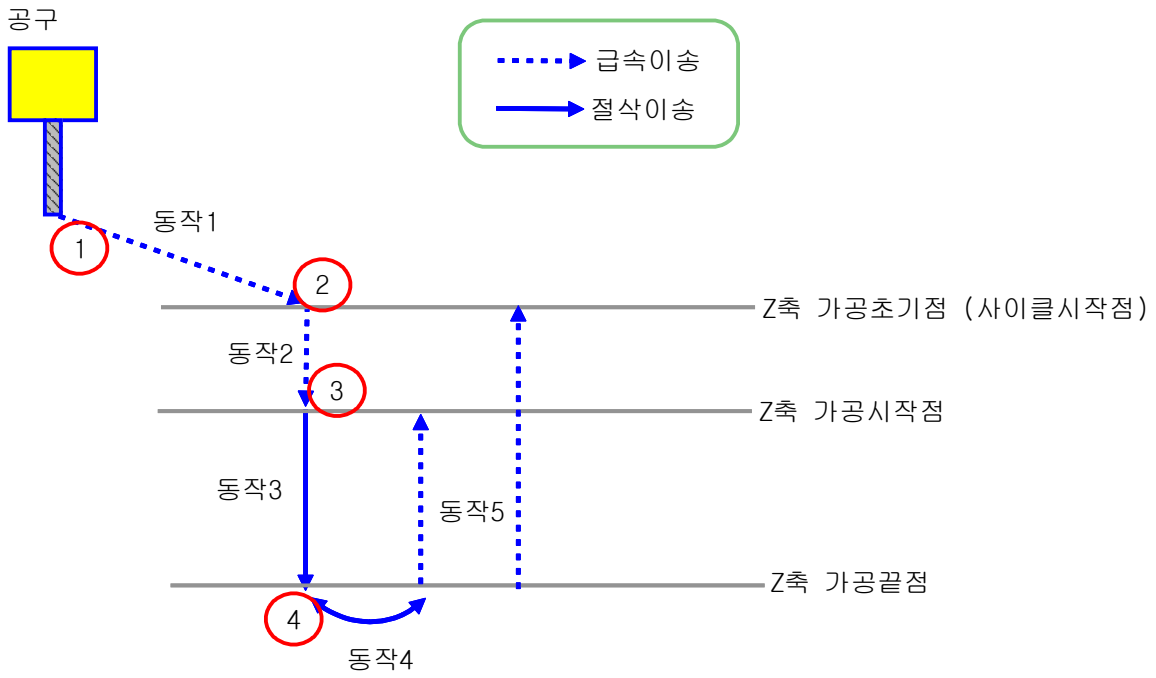
2. 응용 M코드

코드	기능	부가설명	비고
M500	프로그램 처음부터 다시실행	(무한루프)	
M501	M01 옵셔널 스위치해제시 M500실행	M01 옵셔널 스위치 동작시 대기(PAUSE)	
M502	M01 옵셔널 스위치해제서 M500실행	M01 옵셔널 스위치 동작시 프로그램 종료(END)	
M503	출력포트 ON (P=포트번호)	M503 P3 ; 출력포트 3번 ON	포트번호는 0~ 모션보드포트지원값까지
M504	출력포트 OFF (P=포트번호)	M504 P3 ; 출력포트 3번 OFF	포트번호는 0~ 모션보드포트지원값까지
M505	출력포트 SET (P=포트값) ; P값을 이진수값으로 변경한 후 해당 비트값이 1인 출력포트를 ON 합니다.	M505 P16 ; 출력포트 3번 ON	
M506	출력포트 CLEAR (P=포트값) ; P값을 이진수값으로 변경한 후 해당 비트값이 1인 출력포트를 OFF 합니다.	M506 P16 ; 출력포트 3번 OFF	
M507	입력포트 ON대기 (P=포트번호, L=대기시간)	M507 P3 L5.0 ; P3번 포트의 입력이 ON될 때까지 대기합니다. 대기최대시간은 5초입니다. 대기시간을 초과하면 다음 실행블록으로 넘어갑니다. (대기실패시 #33050=1)	L값이 음수이면 대기시간 초과시 알람이 발생합니다.
M508	입력포트 OFF대기 (P=포트번호, L=대기시간)	M508 P3 L5.0 ; P3번 포트의 입력이 ON될 때까지 대기합니다. 대기최대시간은 5초입니다. 대기시간을 초과하면 다음 실행블록으로 넘어갑니다.(대기실패시 #33050=1)	L값이 음수이면 대기시간 초과시 알람이 발생합니다.
M509	SYNC입력 대기 (L=대기시간)	외부동기입력신호(SYNC)를 기다립니다.	L값이 음수이면 대기시간 초과시 알람이 발생합니다.
M510	모션동작 완료대기 (L=대기시간)	-모든 모션동작이 완료되기를 기다립니다. (대기실패시 #33050=1)	L값이 음수이면 대기시간 초과시 알람이 발생합니다.
M511	시스템메모리값 비교불일치 동작대기 (L=시스템메모리주소) (P=시스템메모리 비교값)	M511 L1000 P0 ; 1000번지 내용이 0일 때까지 대기	
M512	시스템메모리값 비교일치 동작대기 (L=시스템메모리주소) (P=시스템메모리 비교값)	M512 L1000 P0 ; 1000번지 내용이 0일 아닐때까지 대기	
M513	축 교환명령	M513 Y A P1 ; Y축과 A축 교환, 단 Y축이 P로 지정된 컨트롤러 축번호와 동일하지 않을때만 교환동작	
M514	축 지정명령	M514 Y P3 ; Y축을 컨트롤러 3번째 축으로 지정	
M515	명령동기화 명령(SYNC)	이전 보간동작이 완료된 후 다음명령이 실행되도록	
M516	강제 다중보간 시작명령		
M517	강제 다중보간 종료명령		
M518	스핀들 원점복귀명령 (완료대기형)	M518 L10 F100; 스핀들 원점복귀 후 F100속도로 스핀들 10도 위치로 이동한다.	
M519	자동카운터 증감	M519 ; 자동카운터 1증가 M519 L-5; 자동카운터 5감소 M519 P100; 자동카운터 100설정	
M520	모션 GIO 패턴출력 START		엔진 V01.03 이후
M521	모션 GIO 패턴출력 END	M521 : 패턴종료 M521 P0 : 패턴종료시 출력 0 M521 P1 : 패턴종료시 출력 1	엔진 V01.03 이후
M522	CLV 각도 연산축 설정	M522 XYZ ; XYZ축의 벡터각도에 의한 CLV 계산	
M523	CLV 동작 각도 지정	M523 P25 : 25도 각도로 지정	
M524	패킹거리Packing Distance	M524 P1.0 : 패킹거리는 P어드레스로 지정	
M525	사이클 공구각도 Tool Angle	M525 P45.0 : 보링등 사이클코드의 공구각도를 P어드레스로 설정	
M526~M579			
M580	현재 코드블록 이동거리계산 시작지정		
M581	M580에서 현재 코드블록까지의 이동거리 계산.		

부록II. 사이클 동작

A. 고정사이클 동작 개요

<고정사이클 기본 5개 동작>



(1) 가공 초기점 급속이송 동작 (① → ②)
(2) 가공 시작점 급속이송 동작 (② → ③)
(3) 가공 (③ → ④)
(4) 드웰동작 (④)
(5) 복귀 동작 (G98: ④ → ②)
(G99: ④ → ③)

대부분의 사이클 동작은 상기 기본동작과 유사합니다. 사이클 동작이 G98로 지정되었다면 복귀시 가공초기점으로 복귀 하며, G99로 지정되었다면 복귀시 가공시작점으로 복귀합니다. 사이클동작은 모달 동작입니다.

※ 모달동작 > G코드명령이 동일그룹(유사한 명령의 그룹)내에서 유효성을 갖는 명령을 말합니다. (자세한 내용은 CNC관련 서적을 참조해 주세요)
예를 들면 아래의 G00코드 역시 모달코드 이므로 두 종류의 코드들이 동일하게 동작을 수행합니다.

G00 X100 Y100 X200 Y200	G00 X100 Y100 G00 X200 G00 X200

B. 드릴 사이클(G81)

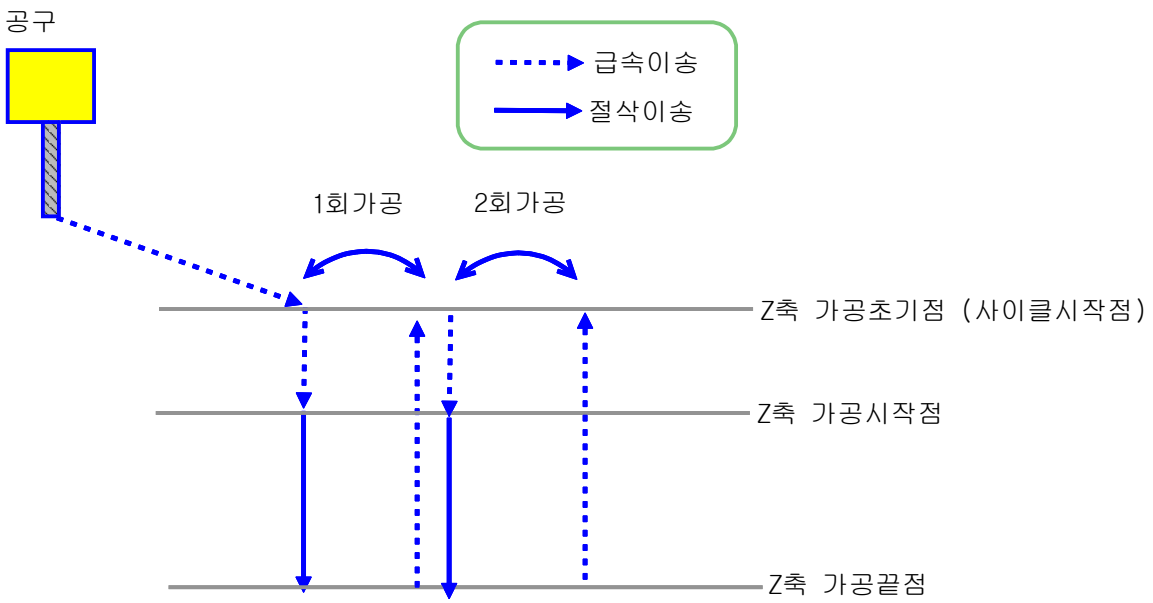
드웰동작이 없는 기본적인 고정사이클 5개동작을 수행합니다.

예> 아래의 코드를 프로그램하여 실행하며 봅니다.

```
S1000 M03 ; 스핀들 회전
G04 X1.0; 스핀들 회전안정 지연
G00 X0 Y0 Z50 ; 초기위치코드
G98 G81 X100 Y100 Z20 R30 F1000 K2 ;  사이클코드 2회 반복(K2)
G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.
G00 X0 Y0 Z50; 원위치복귀
M05 ; 스핀들 정지
M30 ; 프로그램 종료
```

<사이클코드에 대한 동작설명입니다.>

① 가공초기점 100,100,50로 급속이송합니다.
② 가공시작점 Z좌표값 30으로 급속이송합니다.
③ 가공끝점 Z좌표값 20으로 절삭이송합니다.
④ 가공끝점에 도달하면 복귀위치 Z50으로 급속이송한 후 종료합니다. (G98 가공초기점복귀)

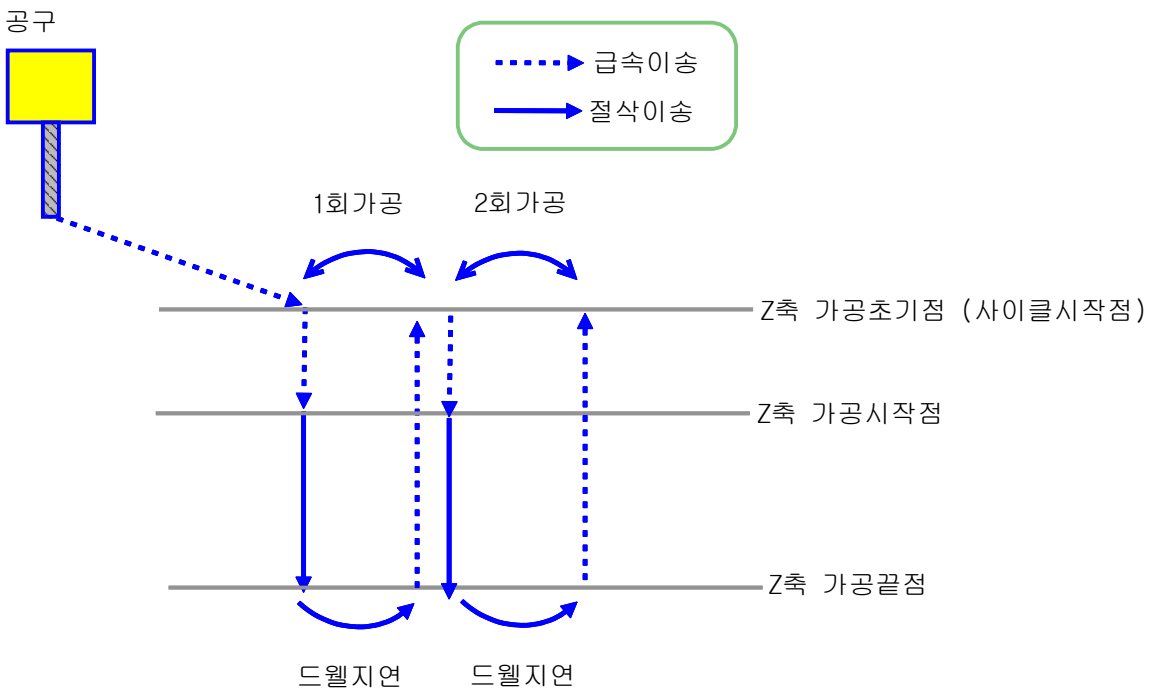


C. 카운터 보어 사이클 (G82)

드릴 사이클에서 드웰동작이 추가됩니다. 드웰값은 P값으로 msec단위로 주어집니다.

예> 아래의 코드를 프로그램하여 실행하며 봅니다.
S1000 M03 ; 스핀들 회전
G04 X1.0; 스핀들 회전안정 지연
G00 X0 Y0 Z50 ; 초기위치코드
G98 G82 X100 Y100 Z20 R30 P1000 F1000 K2 ; 사이클코드 2회 반복(K2)
G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.
G00 X0 Y0 Z50; 원위치복귀
M05 ; 스핀들 정지
M30 ; 프로그램 종료

<사이클코드에 대한 동작설명입니다.>
① 가공초기점 100,100,50로 급속이송합니다.
② 가공시작점 Z좌표값 30으로 급속이송합니다.
③ 가공끝점 Z좌표값 20으로 절삭이송합니다.
④ 드웰시간 1.000초(P1000) 지연합니다.
⑤ 가공끝점에 도달하면 복귀위치 Z50으로 급속이송한 후 종료합니다. (G98 가공초기점복귀)



D. 심공 드릴 사이클 (G83)

깊은 구멍을 위한 반복동작 사이클입니다.

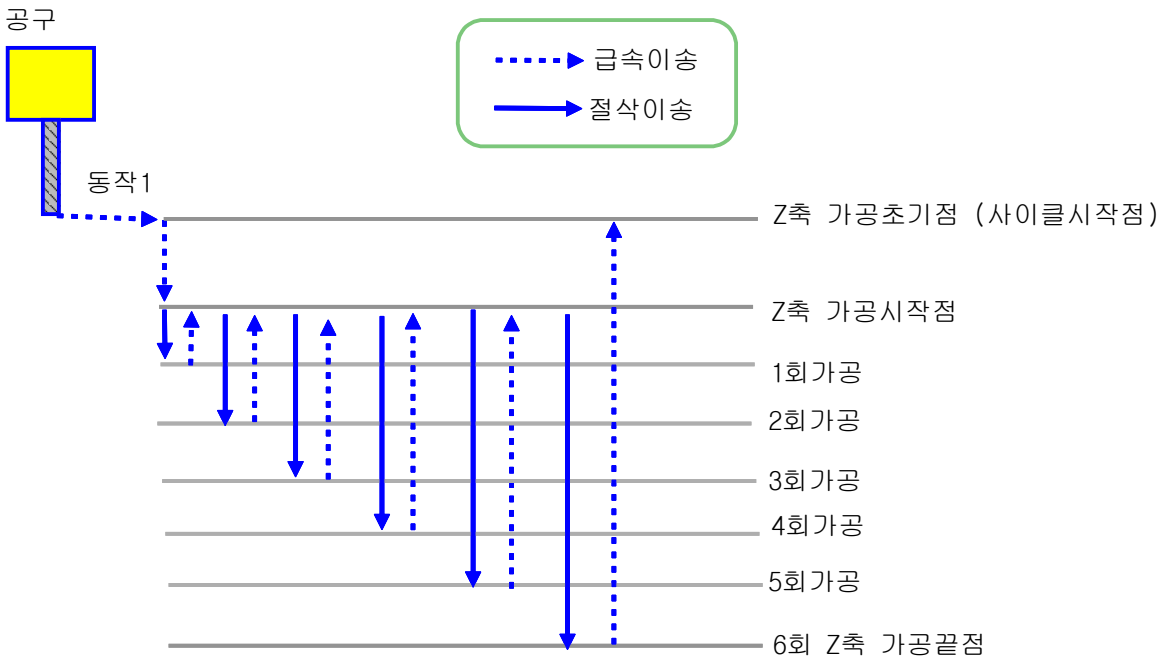
예> 코드를 프로그램하여 실행하며 봅니다.

```
S1000 M03 ; 스핀들 회전
G04 X1.0; 스핀들 회전안정 지연
G00 X0 Y0 Z50 ; 초기위치코드
G98 G83 X100 Y100 Z0 R30 Q5 F1000;  사이클코드
G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.
G00 X0 Y0 Z50; 원위치복귀
M05 ; 스핀들 정지
M30 ; 프로그램 종료
```

<사이클코드부분에 대한 동작설명입니다.>

① 가공초기점 100,100,50로 급속이송합니다.
② 가공시작점 Z좌표값 30으로 급속이송합니다.
③ 1회 가공깊이이 5mm만큼(Q5) Z축을 더 깊이 이동시켜 절삭합니다.
④ 가공시작점 으로 급속이송합니다.
⑤ 가공끝점에 도달할 때까지 ③④ 동작을 반복합니다. (반복회수 6=(R30-Z0)/Q5입니다.)
⑥ 가공끝점 도달하면 복귀위치 Z50으로 급속이송하며 종료합니다. (G98 가공초기점복귀)

프로그램한 후 동작을 확인해보면 아래와 같이 내리락 오르락을 6회 반복한 후 복귀하는 것을 확인할 수 있습니다.



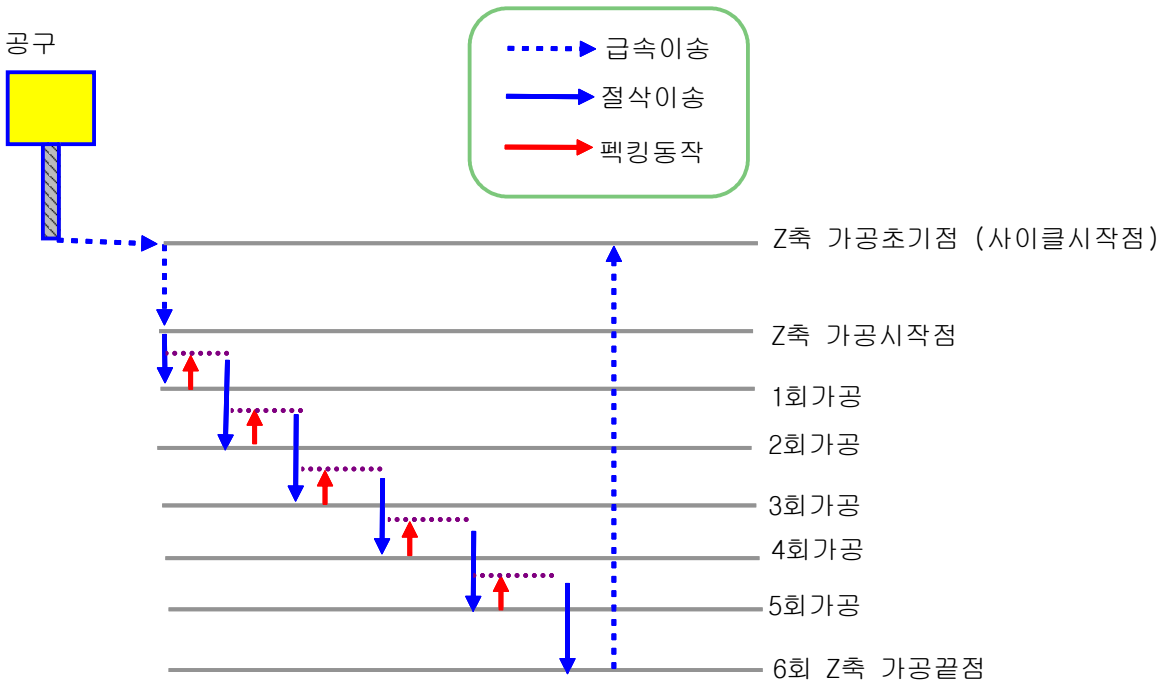
E. 고속 펙킹 드릴 사이클 (G73)

가공 칩의 배출이 용일할 때에는 보다 빠르게 깊은 구멍 가공을 할 수 있는 동작 사이클입니다.
이사이클을 동작시키기 위해서 “CNC구성설정화면”의 "Cycle파라메터 “ Pecking dist값을 설정합니다. 보통 1~2mm 정도 설정해 놓습니다.

예> 코드를 프로그램하여 실행하며 봅니다.
S1000 M03 ; 스핀들 회전
G04 X1.0; 스핀들 회전안정 지연
G00 X0 Y0 Z50 ; 초기위치코드
G98 G73 X100 Y100 Z0 R30 Q5 F1000; 사이클코드
G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.
G00 X0 Y0 Z50; 원위치복귀
M05 ; 스핀들 정지
M30 ; 프로그램 종료

<사이클코드부분에 대한 동작설명입니다.>
① 가공초기점 100,100,50로 급속이송합니다.
② 가공시작점 Z좌표값 30으로 급속이송합니다.
③ 1회 가공깊이이 5mm만큼(Q5) Z축을 더 깊이 이동시켜 절삭합니다.
④ Pecking dist로 설정된 이동량만큼 Z축을 위로 급속이송합니다.
⑤ 가공끝점에 도달할 때까지 ③④ 동작을 반복합니다. (반복회수 6=(R30-Z0)/Q5입니다.)
⑥ 가공끝점 도달하면 복귀위치 Z50으로 급속이송하며 종료합니다. (G98 가공초기점복귀)

프로그램한 후 동작을 확인해보면 G83동작과 비슷하지만 올라오는 거리가 Pecking dist 파라메터 설정값만큼 인 것을 확인할 수 있습니다.



F. 탭핑 사이클 (G84)

오른나사를 기준으로 하는 탭핑가공 사이클입니다. 탭핑가공을 위해서는 기본적으로 스핀들기계장치가 서보형 동작기기일 때 바르게 동작합니다.

예> 아래의 코드를 프로그램하여 실행하며 봅니다.

S1000 M03 ; 스핀들 회전

G04 X1.0; 스핀들 회전안정 지연

G00 X0 Y0 Z50 ; 초기위치코드

G98 G84 X100 Y100 Z20 R30 F1000 ; 사이클코드

;※ F값은 rpm x p (p는 탭의 피치값)으로 설정합니다. 여기서는 탭의 피치가 1mm로 가정하여 F1000 이 설정되었습니다.

G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.

G00 X0 Y0 Z50; 원위치복귀

M05 ; 스핀들 정지

M30 ; 프로그램 종료

<사이클코드부분에 대한 동작설명입니다.>

① 가공초기점 100,100,50로 급속이송합니다.

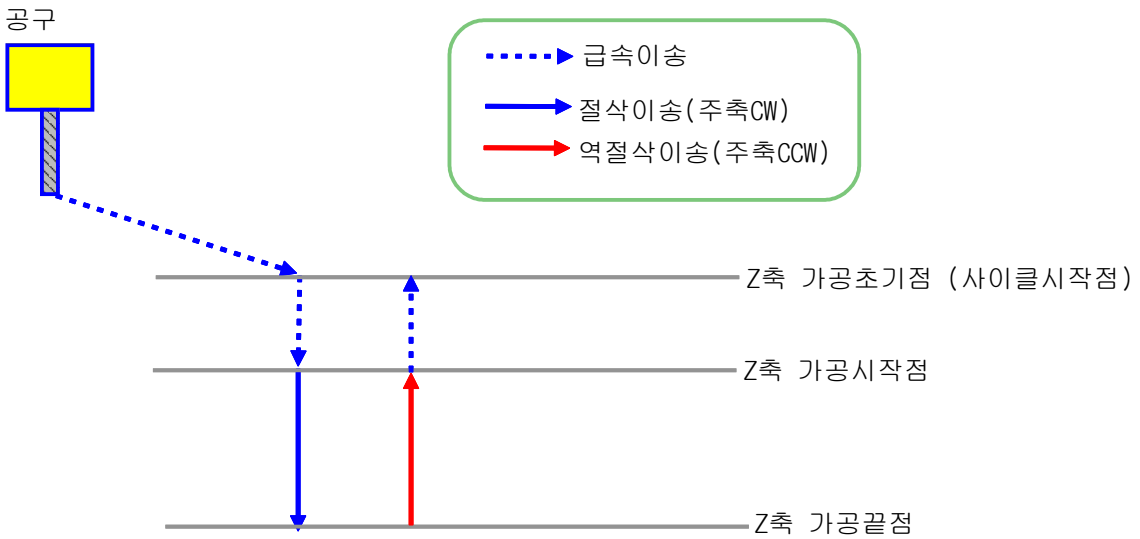
② 가공시작점 Z30 까지 급속이송합니다.

③ 가공끝점 Z20까지 주축 정회전 절삭이송합니다.

④ 가공끝점에 도달하면 주축은 역회전하며 가공시작점 Z30까지 절삭합니다.

⑤ 주축은 다시 정회전하며 복귀위치 Z50으로 급속이송한 후 종료합니다.(G98 가공초기점복귀)

드릴사이클과 비슷하지만 가공끝점에서 스핀들은 역회전하면서 R점까지 역가공을 실행합니다. R점으로 복귀완료 후 스핀들은 다시 정회전하게 됩니다.



※ 실제 탭가공은 RIGID TAP모드(리지드탭모드)와 같이 주축과 Z축의 직선보간으로 동작되어야만 고속으로 정확한 탭가공을 할 수 있습니다. RIGID탭 기능을 사용하려면 주축장치가 서보형장치이면서 펄스구동형이어야 합니다. 그리고 M05코드로 스핀들을 정지시키며, M03/M04코드를 사용하지 않습니다. M29코드를 사이클코드 이전에 삽입하여 G84코드가 리지드탭 사이클동작을 수행하도록 합니다. G80코드를 만나면 리지드탭모드는 해제됩니다.

G. 역탭핑 사이클 (G74)

역탭핑 사이클은 G84 탭핑사이클과 스핀들의 회전방향만 반대이며 다른 동작은 동일합니다.

예> 아래의 코드를 프로그램하여 실행합니다.

S1000 M04 ; 스핀들 역회전시작

G04 X1.0; 스핀들 회전안정 지연

G00 X0 Y0 Z50 ; 초기위치코드

G98 G74 X100 Y100 Z20 R30 F1000 ; 사이클코드

;※ F값은 rpm x p (p는 탭의 피치값)으로 설정합니다. 여기서는 탭의 피치가 1mm로 가정하여 F1000 이 설정되었습니다.

G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.

G00 X0 Y0 Z50; 원위치복귀

M05 ; 스핀들 정지

M30 ; 프로그램 종료

<사이클코드부분에 대한 동작설명입니다.>

① 가공초기점 100,100,50로 급속이송합니다.

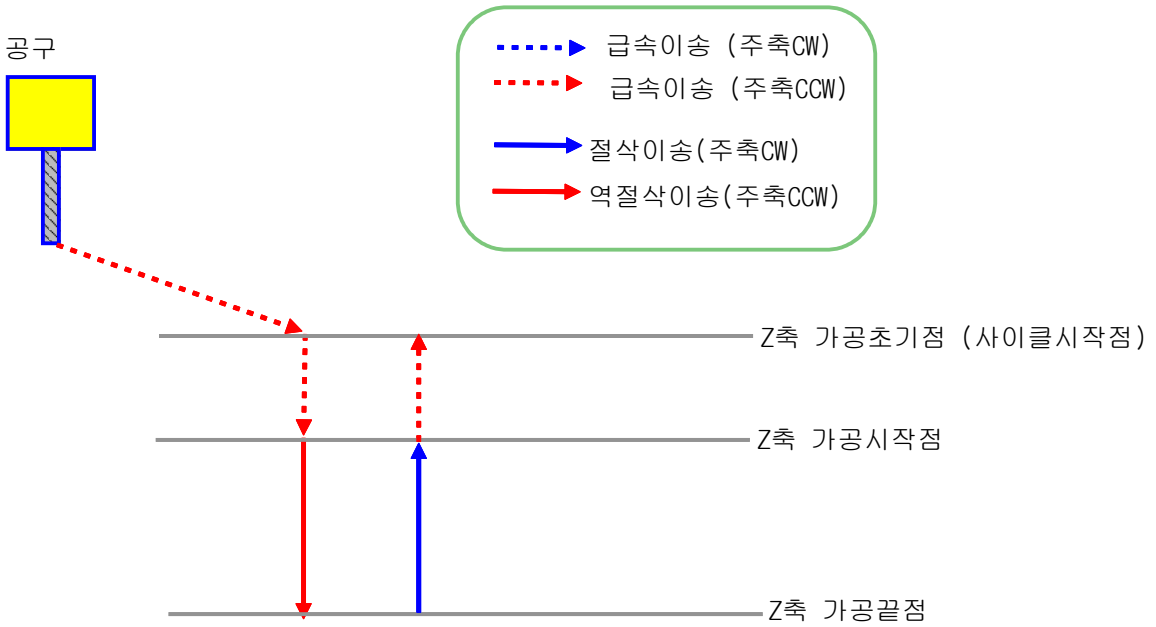
② 가공시작점 Z30 까지 급속이송합니다.

③ 가공끝점 Z20까지 주축 역회전 절삭이송합니다.

④ 가공끝점에 도달하면 주축은 정회전하며 가공시작점 Z30까지 절삭합니다.

⑤ 주축은 다시 역회전하며 복귀위치 Z50으로 급속이송한 후 종료합니다.(G98 가공초기점복귀)

탭핑사이클과 반대로 가공끝점까지 역회전하면서 가공한 후 정회전하며 R점까지 가공을 실행합니다. R점으로 복귀 완료 후 스핀들은 다시 역회전하게 됩니다.



※ 역탭핑사이클의 리지드탭모드도 G84 탭핑사이클과 동일한 방식으로 실행됩니다. G84의 리지드탭모드편을 참조하세요.

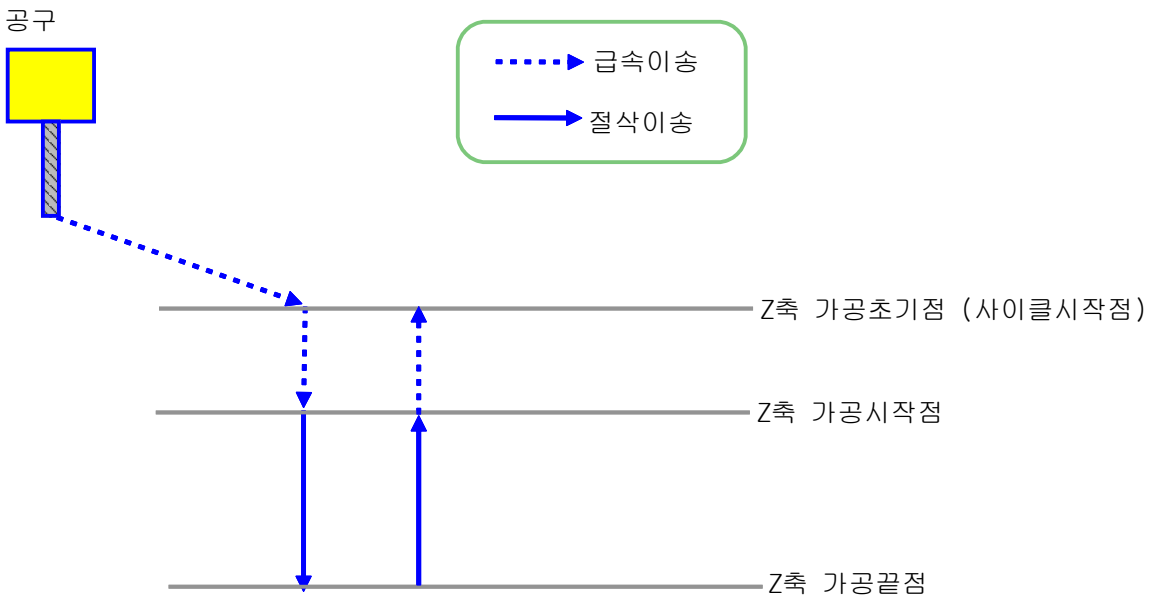
H. 보링 사이클 (G85)

보링은 미리 뚫려있는 구멍을 가공하는 작업입니다.

예> 아래의 코드를 프로그램하여 실행하며 봅니다.

G00 X0 Y0 Z50 ; 초기위치코드
G98 G85 X100 Y100 Z0 R30 F500 ; 사이클코드
G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.
G00 X0 Y0 Z50; 원위치복귀
M30 ; 프로그램 종료

<사이클코드부분에 대한 동작설명입니다.>
① 가공초기점 100,100,50로 급속이송합니다.
② 가공시작점 Z30 까지 급속이송합니다.
③ 가공끝점 Z20까지 절삭이송합니다.
④ 가공시작점 Z30까지 올라가며 다시 절삭합니다.
⑤ 복귀위치 Z50으로 급속이송한 후 종료합니다.(G98 가공초기점복귀)



I. 드웰이 추가된 보링 사이클 (G89)

G85 보링사이클에서 P값으로 드웰시간을 추가할 수 있는 사이클동작입니다.
그 외 모든 기능은 G85와 동일함으로 G85코드 설명을 참조하세요.
아래에 G89코드를 이용한 사이클코드의 변경부분입니다.
G98 G85 X100 Y100 Z0 R30 P1000 F500 ; Z축 가공끝점에서 P1000=1초동안 드웰타임(휴지기간)을 갖습니다.

J. 보링 사이클 2 (G86)

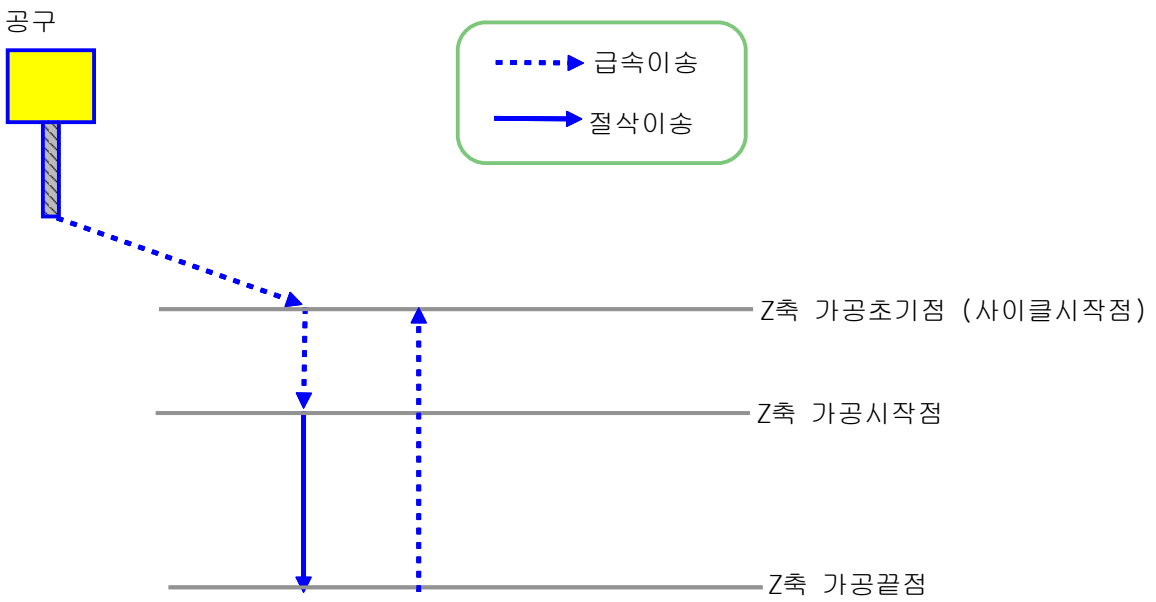
G85보링 사이클과 유사하며 가공끝점에 도달하면 복귀점까지 바로 급속이송으로 올라가는 동작을 수행합니다. 흔히 **황삭 보링사이클**로 분류합니다.

예> 아래의 코드를 프로그램하여 실행하며 봅니다.

```
G00 X0 Y0 Z50 ; 초기위치코드
G98 G86 X100 Y100 Z0 R30 F500 ;  사이클코드
G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.
G00 X0 Y0 Z50; 원위치복귀
M30 ; 프로그램 종료
```

<사이클코드부분에 대한 동작설명입니다.>

① 가공초기점 100,100,50로 급속이송합니다.
② 가공시작점 Z30 까지 급속이송합니다.
③ 가공끝점 Z20까지 절삭이송합니다.
④ 복귀위치 Z50으로 급속이송한 후 종료합니다.(G98 가공초기점복귀)



K. 정밀 보링 사이클(G76)

일반보링사이클에서는 공구가 가공끝점에 도달하면 다시 가공초기점이나 시작점으로 복귀하므로 공구의 이송흔적을 남기게 됩니다. 정밀보링사이클은 공구의 날방향과 반대방향으로 후퇴한 후 복귀하여 날의 이송흔적을 남지 않게 동작하는 방식입니다. (흔히 정밀보링과 뒤에 나올 백보링사이클은 외날 보링가공에 많이 사용합니다.)

예> 아래의 코드를 프로그램하여 실행하며 봅니다.

G00 X0 Y0 Z50 ; 초기위치코드

G98 G76 X100 Y100 Z0 R30 Q5 F500 ; 사이클코드, Q5 로써 5mm만큼 공구를 후퇴시키는 동작이 추가됩니다.

G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.

G00 X0 Y0 Z50; 원위치복귀

M30 ; 프로그램 종료

<사이클코드부분에 대한 동작설명입니다.>

① 가공초기점 100,100,50로 급속이송합니다.

② 가공시작점 Z30까지 급속이송합니다.

③ 가공끝점 Z0까지 절삭이송합니다.

④ 주축정위치 정지를 실행합니다.

※ Cycle파라메터 "Enable Spindle Orientation "을 참조하세요.

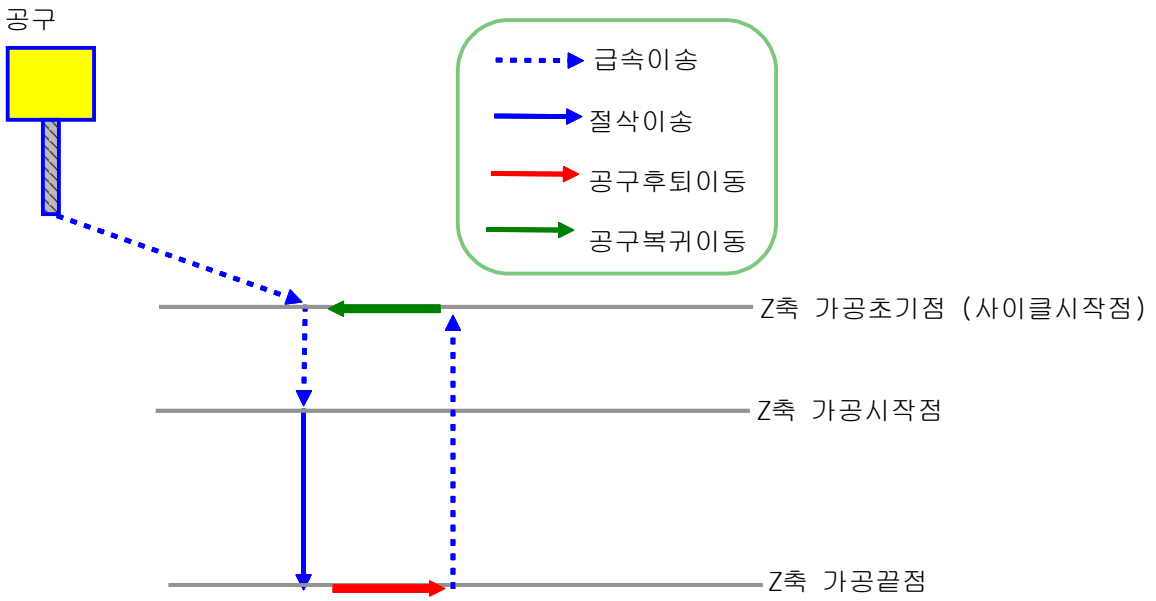
⑤ Q5로 지정된 거리만큼 공구를 후퇴합니다.

※ Cycle파라메터 Tool Angle값의 역방향이 공구후퇴각이 됩니다.

⑥ 복귀위치 Z50으로 급속이송합니다.(G98 가공초기점복귀)

⑦ Q5로 지정된 거리만큼 공구를 진입하고 스핀들정회전 후 종료합니다.

※ Cycle파라메터 Tool Angle값이 공구진입각이 됩니다.



L. 백보링 사이클(G87)

정밀보링사이클과 반대방향인 아래에서 위로 공구가 이송하면서 보링하는 동작입니다.

예> 아래의 코드를 프로그램하여 실행하며 봅니다.

G00 X0 Y0 Z50 ; 초기위치코드

G98 G87 X100 Y100 Z30 R0 Q5 F500 ; 사이클코드, Z보다 R값이 더 아래에 있습니다.

G80 G00 Z60; Z축을 피하면서 사이클을 종료합니다.

G00 X0 Y0 Z50; 원위치복귀

M30 ; 프로그램 종료

<사이클코드부분에 대한 동작설명입니다.>

① 가공초기점 100,100,50로 급속이송합니다.

② 주축 정위치 제어를 실행합니다.

※ Cycle파라메터 "Enable Spindle Orientation "을 참조하세요.

③ Q5로 지정된 거리만큼 공구를 후퇴합니다.

※ Cycle파라메터 Tool Angle값의 역방향이 공구후퇴각이 됩니다.

④ 가공시작점 Z0까지 급속이송합니다.

⑤ Q5로 지정된 거리만큼 공구를 진입합니다.

※ Cycle파라메터 Tool Angle값이 공구진입각이 됩니다.

⑥ 가공끝점 Z30으로 올라가면서 절삭합니다.

⑦ 주축정위치 정지를 실행합니다.

※ Cycle파라메터 "Enable Spindle Orientation "을 참조하세요.

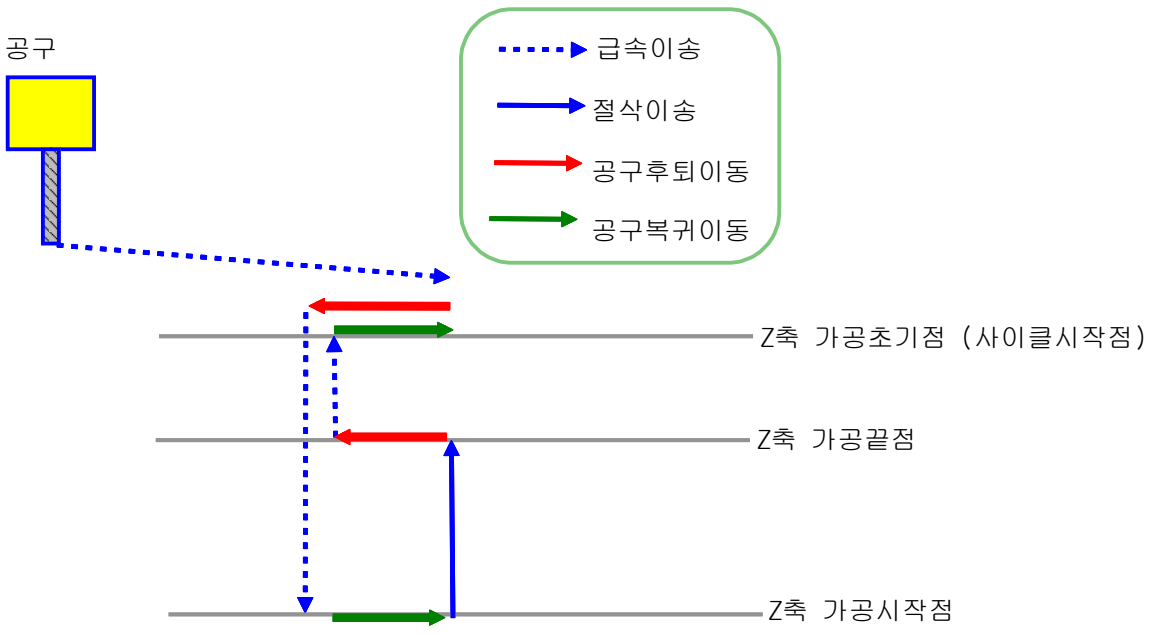
⑧ Q5로 지정된 거리만큼 공구를 공구를 후퇴합니다.

※ Cycle파라메터 Tool Angle값의 역방향이 공구후퇴각이 됩니다.

⑨ 복귀위치 Z50으로 급속이송합니다.(G98 가공초기점복귀)

⑩ Q5로 지정된 거리만큼 공구를 공구를 진입하고, 주축은 회전하며 종료합니다.

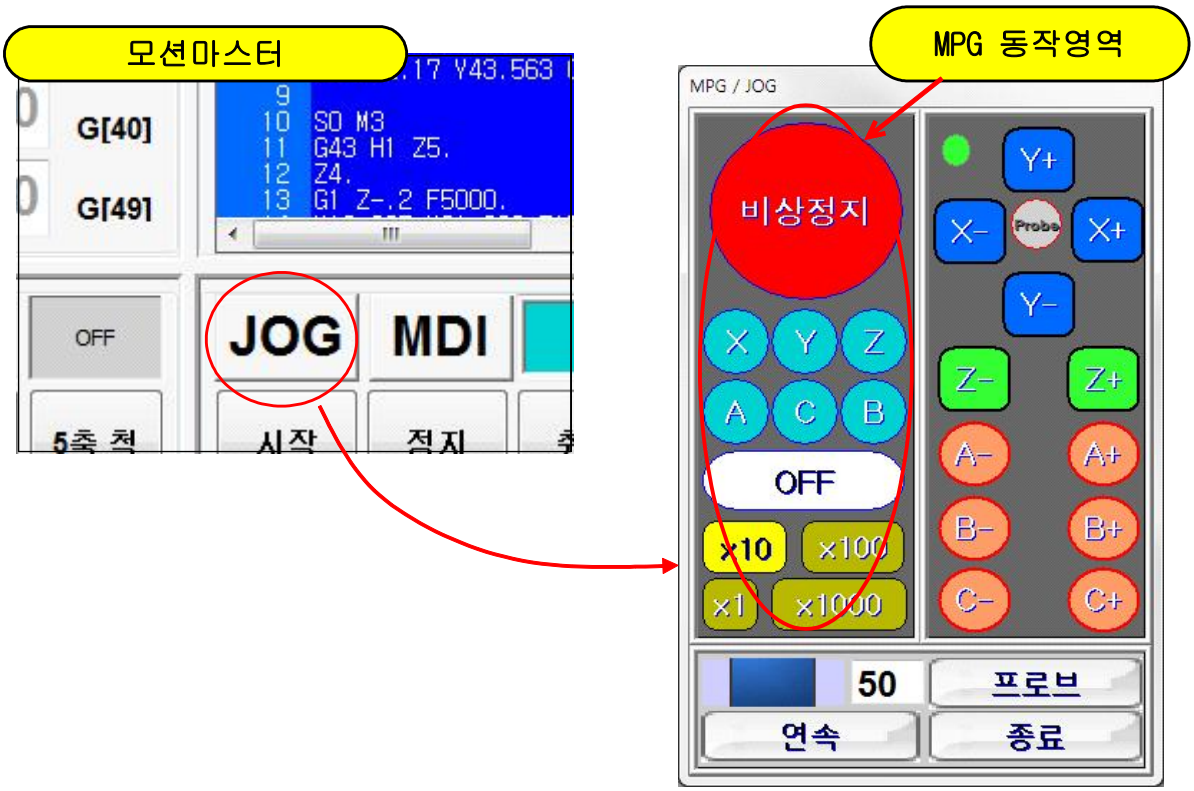
※ Cycle파라메터 Tool Angle값이 공구진입각이 됩니다.



부록III. MPG 연결

A. 모션마스터 MPG 동작특징

모션마스터의 [조그]버튼을 눌러주면 조그/MPG기능제어를 위한 화면이 나타납니다. 이 화면은 모션마스터의 JOG옵션에 따라 다양한 모양으로 나타날 수 있지만 기본적인 동작은 동일합니다.
모션마스터의 MPG동작은 아래의 특징으로 요약될 수 있습니다.
(1) JOG키 입력과 병행된 동작가능
(2) 다양한 MPG의 입력에 대응하는 기능별 동작기능 설정가능
(3) 다양한 휠(WHEEL)의 동작방식 설정가능



여기서는 조그키입력 기능이 아닌 MPG 동작영역의 기능이 바르게 동작하기 위한 CNC구성 설정을 살펴봅니다.
먼저 기본적으로 MPG장치는 아래의 3가지 방식으로 연결될 수 있습니다.

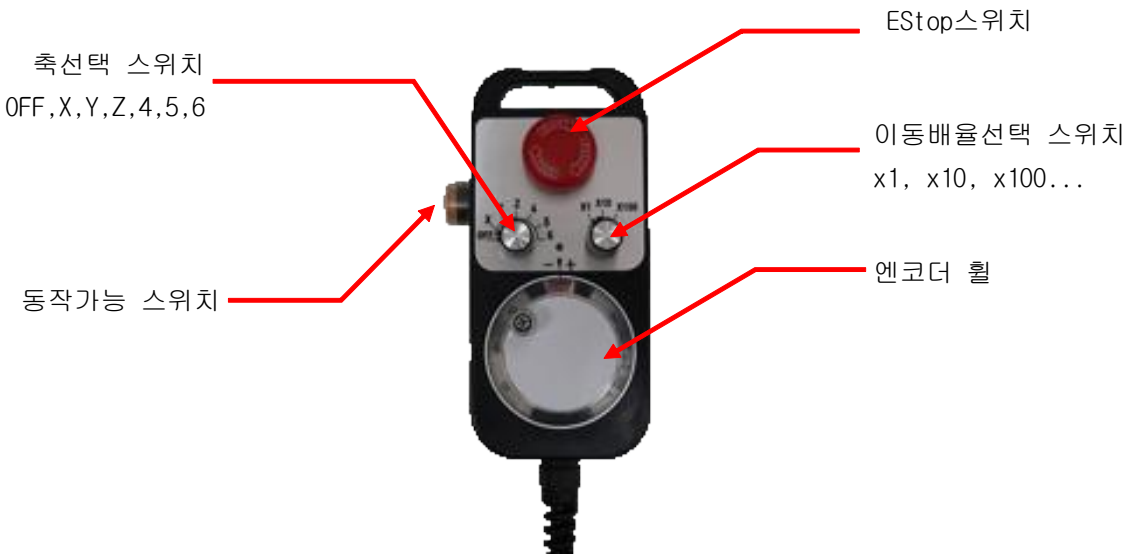
(1) 모션보드의 GIO 입력을 이용하는 MPG연결방식
(2) 모션보드의 확장연결신호(EXT-LINK : EIO확장연결)에 의한 연결방식
(3) COM포트를 이용한 통신방식의 MPG연결방식

현재 모션마스터 V01.R13.208에서는 (1)과 (2)방식만을 지원합니다.
특히 GIO입력을 이용할 경우 GIO의 입력핀의 사용량을 줄이기 위해 아날로그입력 핀을 이용해서 축(AXIS)과 이동배율(x1, x10, x100...)을 선택할 수 있는 기능을 포함하고 있습니다.

B. MPG의 구성요소

일반적인 MPG는 자동동작이 아닌 상태에서 수동으로 기계의 축을 하나씩 움직여 주는 역할을 합니다.
이를 위해서 아래와 같은 기본적은 구성을 갖고 있습니다.

(1) 엔코더 휠 (Encoder Wheel)
(2) 축선택 스위치 (Axis Selector)
(3) 이동배율선택 스위치 (Movement Multiplier)
(4) 동작가능 스위치 (Enable Switch)
(5) 비상정지 스위치 (EStop Switch)



MPG마다 각 구성요소들의 형태에 차이가 있지만, 대체적 5가지 정도의 구성요소를 갖습니다.
제품마다 각 구성요소에서 출력되는 스위칭 신호가 다를 수 있으며
모션마스터프로그램은 5가지 구성요소들의 다양신호와 연결될 수 있는 MPG SETUP 윈도우를 제공합니다.
다음에 간단히 MPG SETUP윈도우 화면에 대한 전체구성과 MPG의 실제 구성요소간에 비교해봅니다.

C. MPG Setup 윈도우

아래는 모션마스터프로그램의 CNC 동작화면에서 CNC 구성설정화면을 선택한 후 나타난 CNC구성설정화면 중 일부를 보여줍니다.

[MPG]버튼을 누르면 아래와 같은 MPG SETUP 윈도우가 표시되며 각 항목별 구성은 MPG의 실제 구성요소들과 대응합니다.

< MPG SETUP 윈도우 >

다음 절에서 MPG의 각 구성요소들의 신호들과 연결 및 설정을 살펴봅니다.

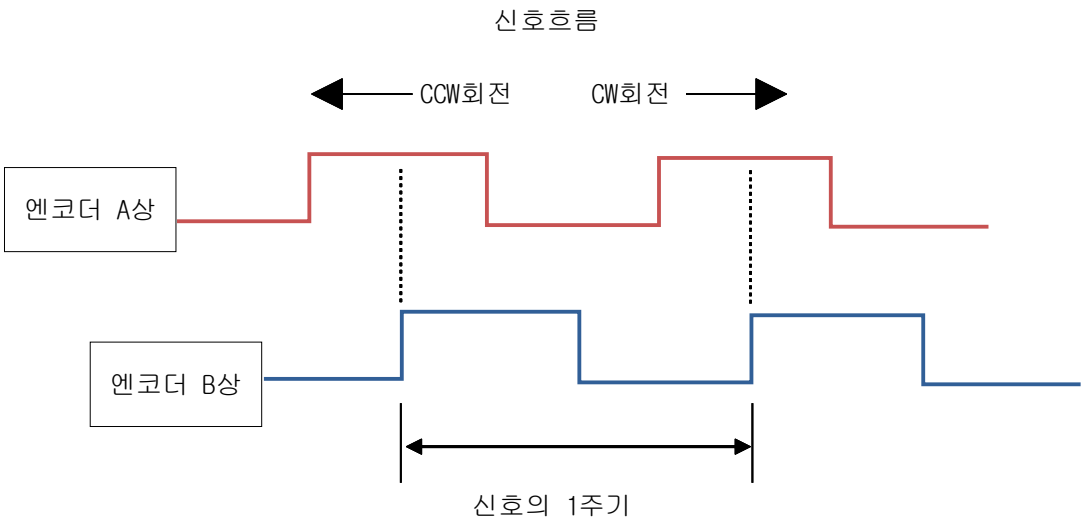
D. MPG 구성요소들의 신호와 연결방식 및 설정

아래의 5가지 MPG 스위치 요소들과 이들의 신호연결 및 MPG 셋업화면의 설정을 살펴봅니다.
(1) 엔코더 휠 (Encoder Wheel)
(2) 축선택 스위치 (Axis Selector)
(3) 이동배율선택 스위치 (Movement Multiplier)
(4) 동작가능 스위치 (Enable Switch)
(5) 비상정지 스위치 (EStop Switch)

E. 엔코더 휠 (ENCODER WHEEL)

손잡이가 달린 휠 형태의 엔코더는 일반적으로 광학식 INCREMENTAL(증분식) 엔코더와 동일한 신호형태를 갖고 있습니다. 즉 기본적으로 90도의 위상차를 갖는 A상과 B상의 신호가 출력됩니다.

이들 위상의 형식은 아래와 같이 단순합니다.



엔코더의 각 위상신호 A,B는 제품의 종류마다 크게 아래의 3가지 전기적 신호형태를 갖습니다.
(1) 차동 라인 드라이브 형태의 신호
(2) 오픈 콜렉터 형태의 신호
(3) 토템폴 형태의 신호

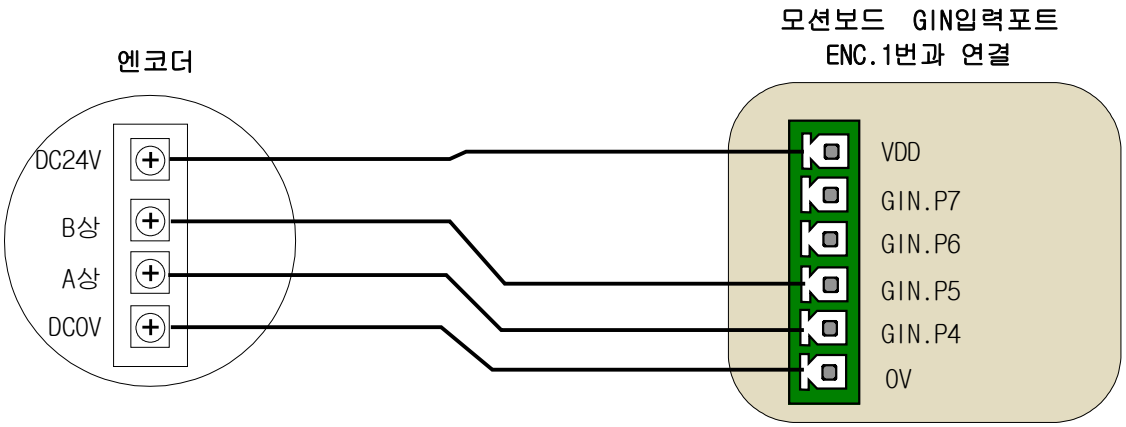
1. 연결

X6V3엔진을 사용하는 모션컨트롤러의 MPG의 엔코더휠의 신호는 아래의 3가지 경로 중 하나를 택해서 연결할 수 있습니다.
(1) GIO입력의 ENC 0번 채널 (GIO입력핀 2번/3번)
(2) GIO입력의 ENC 1번 채널 (GIO입력핀 4번/5번)
(3) Extend Link통신에 의한 전용 MPG연결보드 <= 해당제품 매뉴얼을 참조하세요.
이들 연결방식 중 (2)을 방식을 기준으로 설명합니다. 모션보드의 GIO입력은 포토커플러에 의해 DC24전원을 기준으로 신호를 입력받도록 설계되어 있습니다. BLX6보드와 트레이닝보드 모두 기준전원 DC24V에 의한 포토커플러를 통

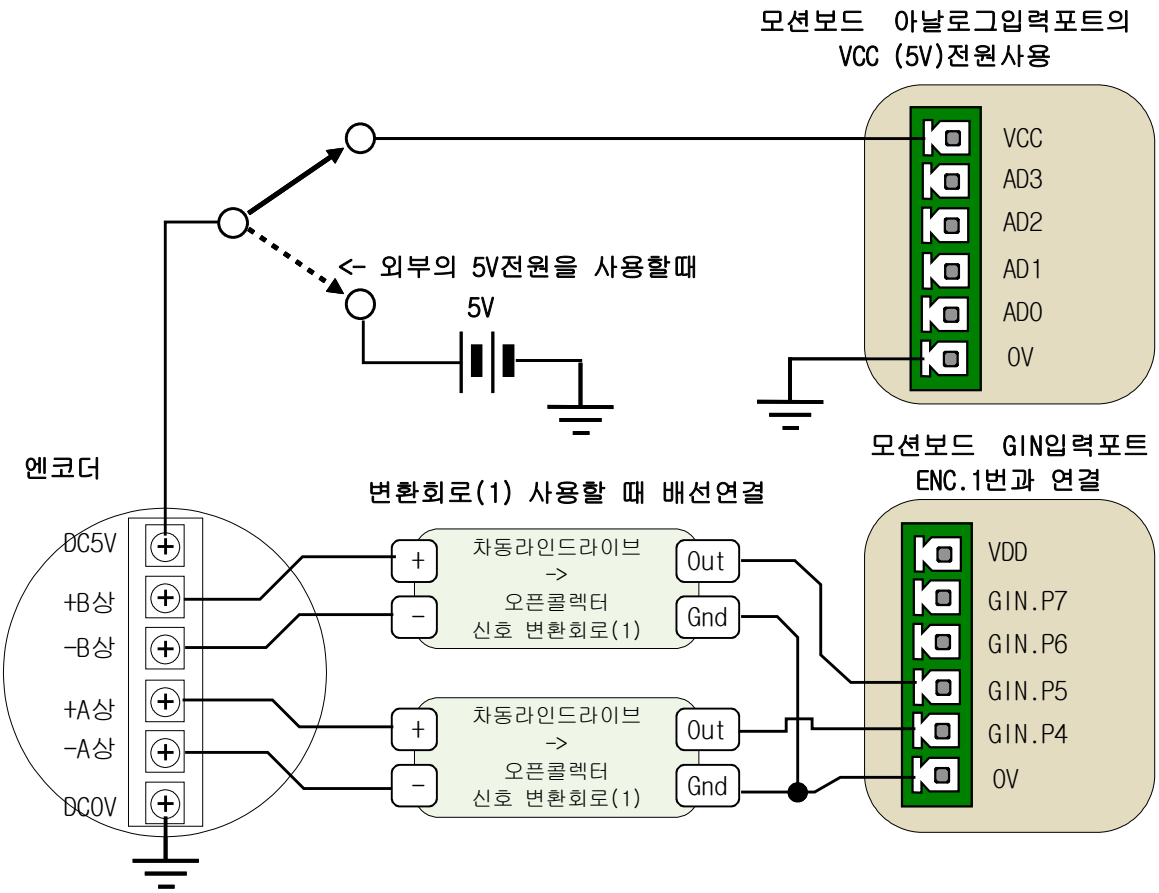
해 신호를 입력받습니다. 그러므로 엔코더가 DC24V로 동작되며, 신호가 오픈콜렉터 형태이거나 토템폴 형태일 경우에는 직접적으로 연결하여 동작할 수 있습니다.

반면에 차동 라인 드라이브 형태의 신호를 출력하는 엔코더의 경우 반드시 DC5V의 입력을 받으며 출력신호 역시 5V영역이므로 직접적으로 모션보드에 의해 입력받을 수 없으며 간단한 포토커플러 회로나 트랜지스터 회로를 통해서 신호를 변환하여 입력받을 수 있습니다. 아래의 (2)번 내용은 엔코더의 라인드라이브신호를 포토커플러회로를 구성하여 신호를 변환시킨 후 연결하는 예를 보여줍니다.

(1) DC24V입력용 오픈콜렉터(혹은 토템폴) 출력형태의 엔코더와 모션보드(BLX6, 트레이닝보드)의 연결



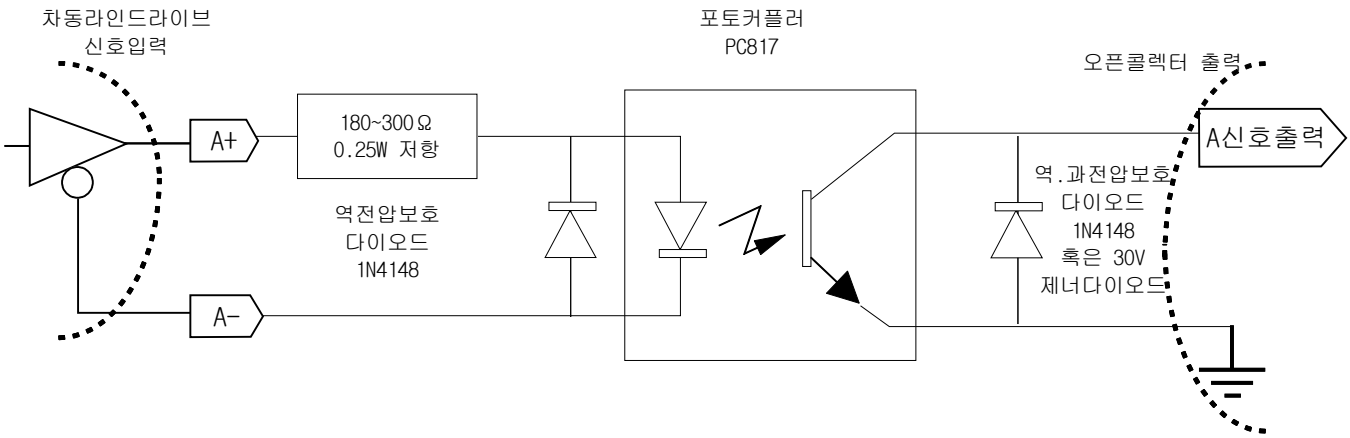
(2) 차동라인드라이브 출력형태의 엔코더와 모션보드(BLX6, 트레이닝보드)의 연결



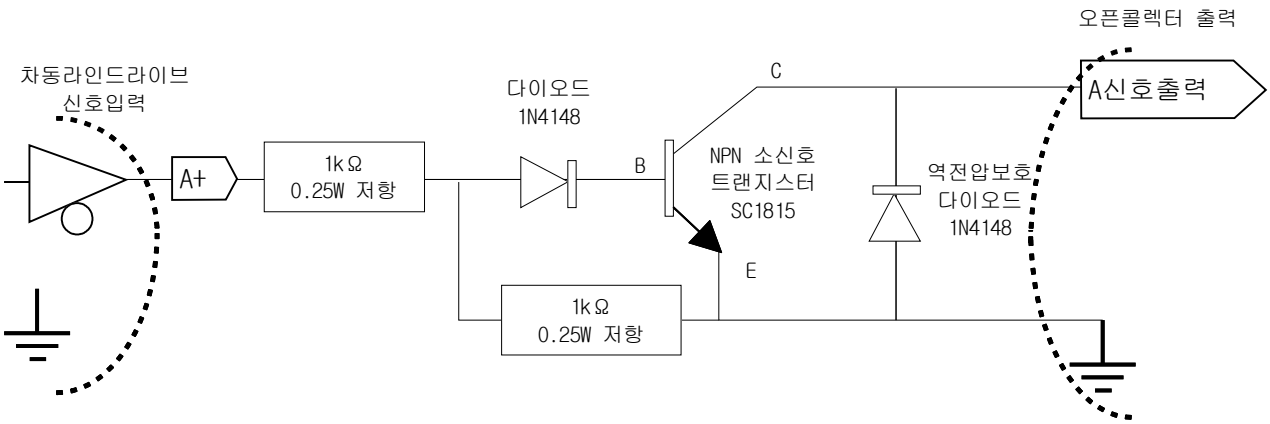
다음은 차동라인드라이브신호를 오픈콜렉터신호로 변환해 주는 회로 예를 보여줍니다.

※차동라인 드라이브 -> 오픈콜렉터신호 변환 회로

변환회로(1) 포토커플러를 이용한 신호변환

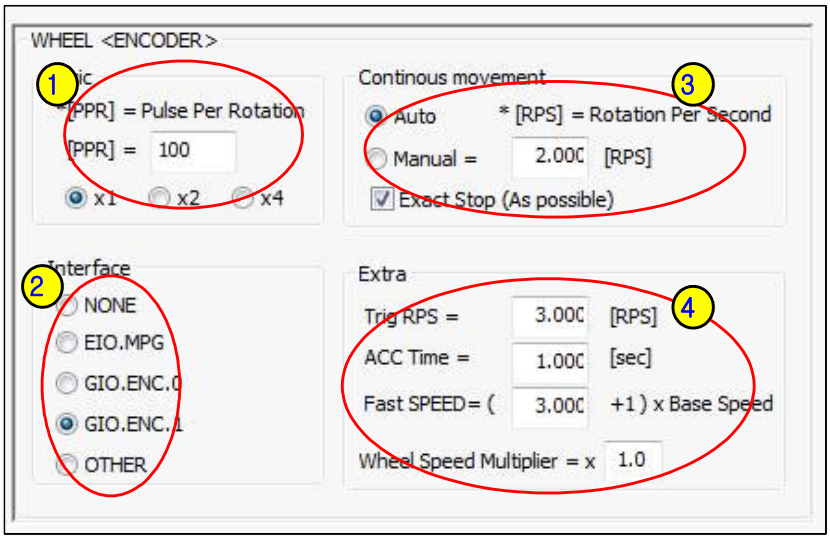


변환회로(2) NPN 트랜지스터를 이용한 신호변환



2. MPG SETUP 윈도우 설정

엔코더휠의 동작속성설정은 아래와 같이 4가지의 구성설정으로 이루어집니다. 하나씩 살펴봅니다.



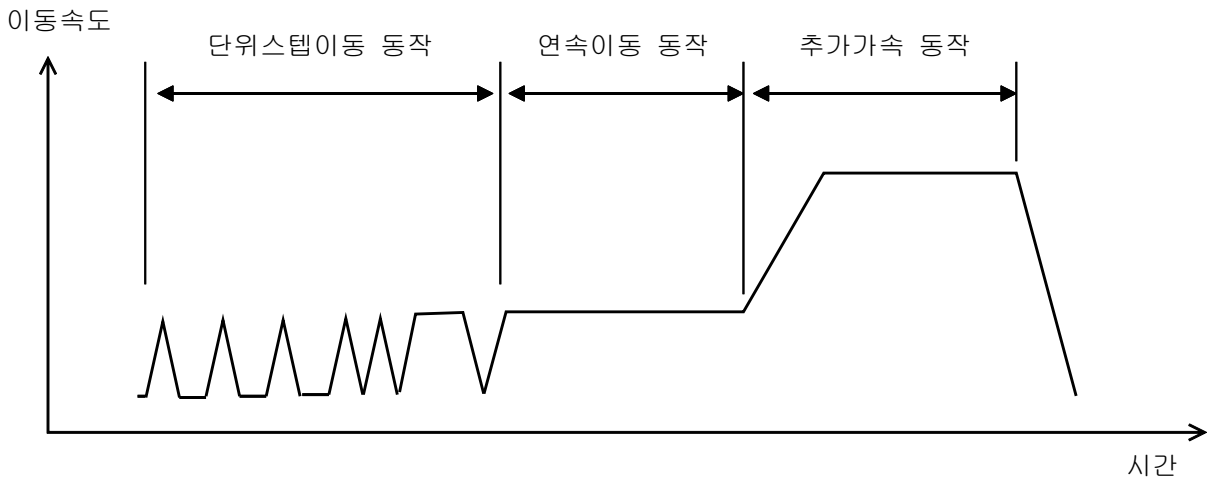
(1) 기본구성 (엔코더의 PPR 설정)
PPR은 엔코더가 1회전당 출력하는 펄스수를 의미합니다. (Pulse Per Rotation)
대부분 100[PPR] 혹은 25[PPR]로 구성되며 해당 엔코더에 맞는 PPR을 입력합니다.

INCREMENTAL 위상신호방시의 엔코더는 기본적 1체배 2체배 4체배 동작을 실행할 수 있습니다.
상기 100[PPR]의 설정은 1체배 방식을 기준으로 한 펄스수입니다.
엔코더 휠을 동작할 때 1체배방식 그대로 동작할 수 있으며, 2체배 혹은 4체배 방식으로 동작시킬 수 있습니다.
4체배 방식의 경우 같은 휠의 동작일 때 4배 많은 펄스를 발생할 수 있습니다.

(2) 연결방식
상기 배선연결예는 GIO.ENC.1 번채널을 이용하는 연결입니다.

(3) 연속 이동동작 설정
모션마스터프로그램은 MPG의 엔코더휠의 움직임에 대해서 자동으로 가감속하는 AUTO방식으로 동작됩니다. 그러나 Manual 방식으로 설정하게 되면 여러가지 설정에 의한 아래의 동작형태 동작될 수 있습니다.
모션마스터는 MPG동작은 MPG펄스를 그대로 서보드라이브에 전송하지 않으며 해당하는 축의 설정된 가감속도에 맞게 가감속하여 출력합니다.

모션마스터의 Manual MPG동작을 타임그래프로 간단히 표현할 수 있습니다.



연속이동 동작의 기준은 엔코더휠의 회전동작속도와 단위스텝의 이동시간과 관련됩니다.
이때 단위스텝이동과 연속 이동동작의 전환을 자동(Auto)로 설정할 수도 있고, 수동(Manual)으로 설정할 수 있습니다. 수동으로 설정할 경우 엔코더 휠의 회전동작속도[RPS]를 설정하게 되면 설정된 RPS보다 높은 회전동작에서 연속이동동작을 실행하게 됩니다. (RPS는 1초당 회전수를 의미합니다. Rotation Per Second)

(4) 추가 가속동작
연속회전 동작시 엔코더휠을 특정회전수보다 빨리 회전시키면 추가적으로 가속하여 동작할 수 있습니다.
추가적이 가속동작은 아래의 3가지 파라메터에 의해 실행됩니다.

- 추가 가속동작의 시작하기 위한 엔코더휠의 회전동작 속도[RPS]
  : 기본적으로 3.0으로 설정합니다.(초당3회전이상 동작시킬 때)
- 가속시간 설정 : 추가가속을 실행을 위한 가속도 결정
  : 기계 관성부하와 작동의 안전에 맞게 적당히 설정합니다.
- 가속 한계 속도배율설정

: 연속동작의 기준속도에 추가적으로 몇 배율로 동작할 것인지 결정합니다.
기계 관성부하와 작동의 안전에 맞게 적당히 설정합니다.

※속도배율값을 0.0으로 설정하면 추가 가속동작을 실행하지 않는 것과 동일합니다.

- Exact Stop 옵션은 MPG휠 동작시 가능하면 배율이동의 배수만큼 이동하도록 조절하는 옵션입니다.

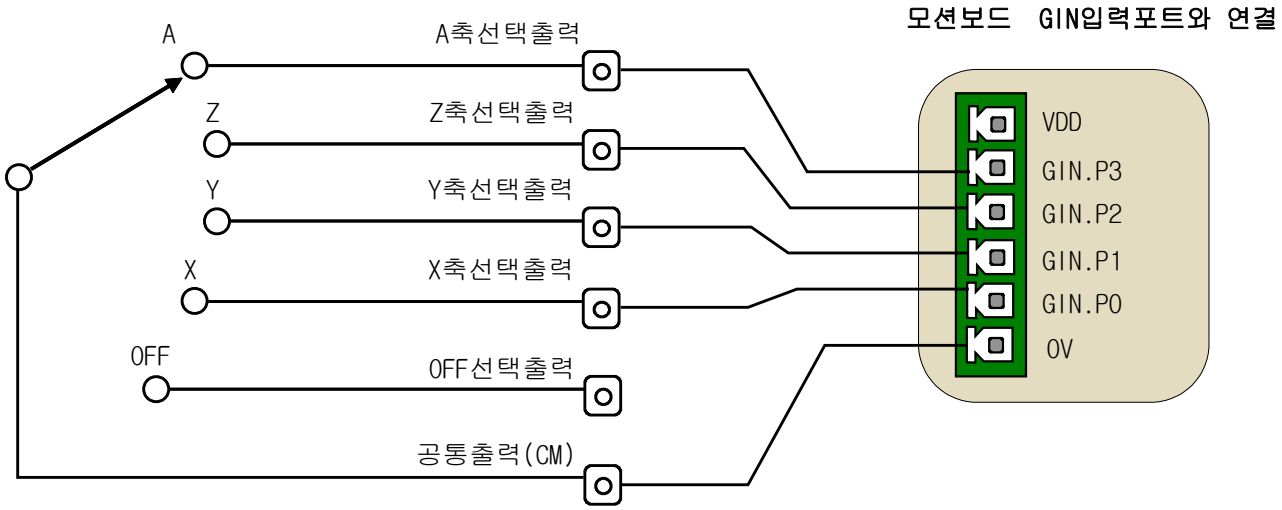
F. 축 선택 및 배율선택 스위치

※ 축선택 스위치와 배율선택스위치는 기능만 다르며, 전기적 연결과 설정이 동일함으로 축선택 스위치를 기준으로만 설명됩니다.

일반적으로 축선택스위치와 이동배율선택스위치는 기본적으로 2가지 형태의 스위치를 주로 사용합니다.
(1) SELECTOR 접점 출력 스위치
(2) BCD 접점 출력 스위치
모션마스터프로그램은 이들 접점의 수를 줄이기 위해 아날로그 입력핀(ADC)을 이용하여 축선택과 배율선택기능을 제공합니다. 이들 각 스위치형태에 따른 연결과 설정들을 살펴봅니다.

1. SELECTOR 스위치 접점방식과 연결

SELECTOR 스위치접점 방식은 선택 축 수 만큼의 입력접점이 요구되므로 비교적 많은 입력접점을 사용합니다.
아래에 X,Y,Z,A 4개의 축 선택에 대한 GIO입력 연결방식을 보여주며, GIO 입력이 아닌 ADC(아날로그) 입력에 의해 입력신호를 1개로써 사용하는 방법도 제시합니다.

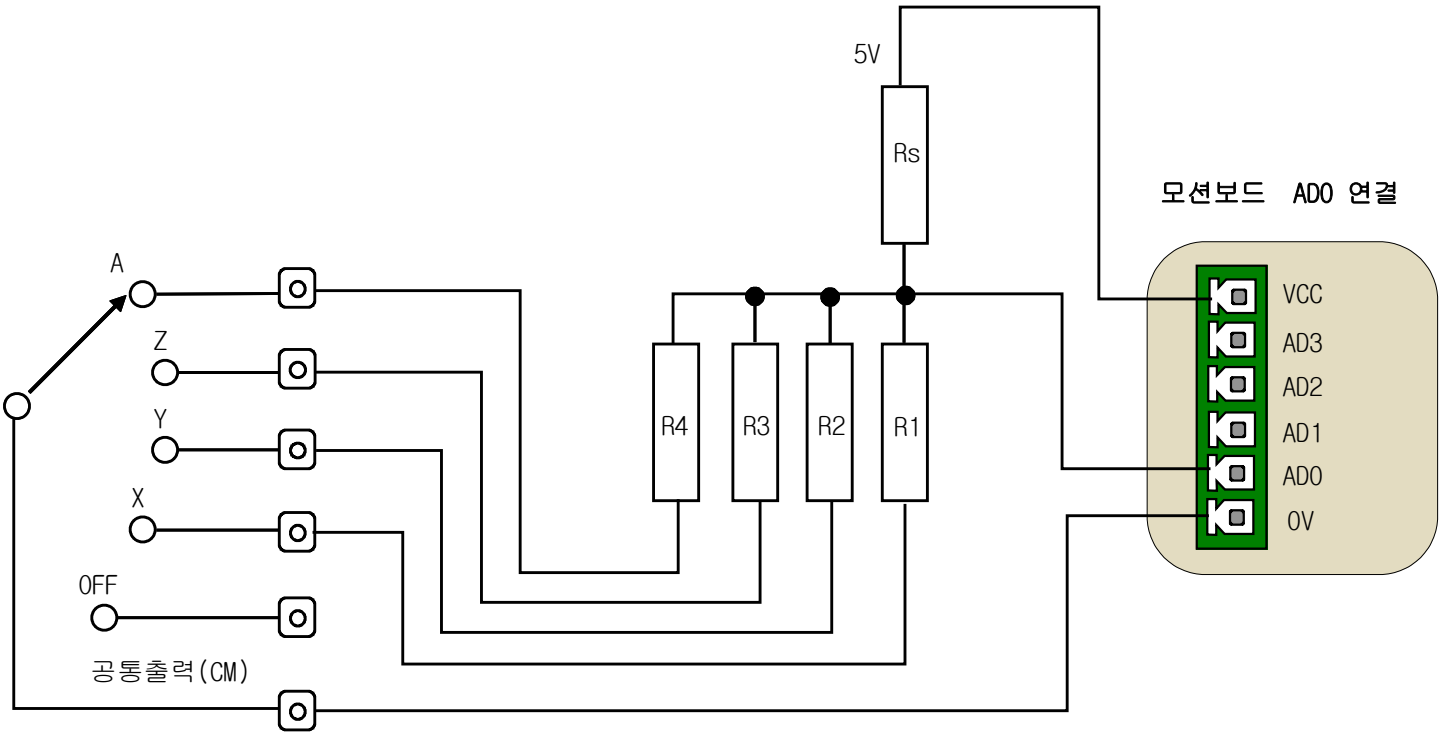


- 출력접점 동작표 -

출력 / 선택	OFF	X	Y	Z	A
OFF 선택	1	0	0	0	0
X 선택	0	1	0	0	0
Y 선택	0	0	1	0	0
Z 선택	0	0	0	1	0
A 선택	0	0	0	0	1

0 : 공통출력(CM)과 연결안됨
1 : 공통출력(CM)과 연결됨

2. SELECTOR 스위치와 ADC입력의 연결

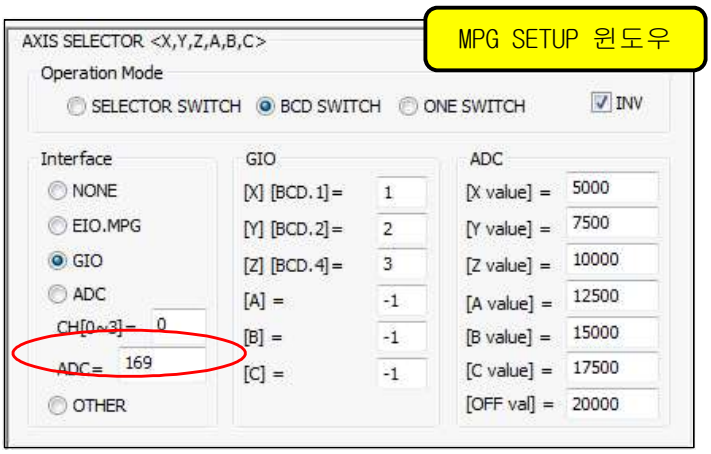


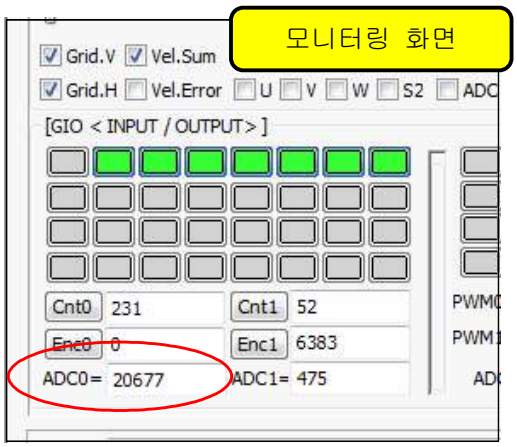
Rs, R1, R2, R3, R4는 각 축선택 스위칭마다 0~5V사이의 전압이 골고루 분포될 수 있도록 값을 정합니다.
Rs값은 1k옴 이상값을 정하며, 각 저항이 10k옴 이하의 범위내에서 설정합니다.

예> 아래와 같은 저항을 선택하면 AD0는 우측의 전압 근사값이 표현됩니다.

Rs = 1.0k
R1 = 0.5k AD0 = 5[V] x 0.5k / (1.0k + 0.5k) = 1.67 [V]
R2 = 1.0k AD0 = 5[V] x 1.0k / (1.0k + 1.0k) = 2.50 [V]
R3 = 2.5k AD0 = 5[V] x 2.5k / (1.0k + 2.5k) = 3.57 [V]
R4 = 9.0k AD0 = 5[V] x 9.0k / (1.0k + 9.0k) = 4.50 [V]

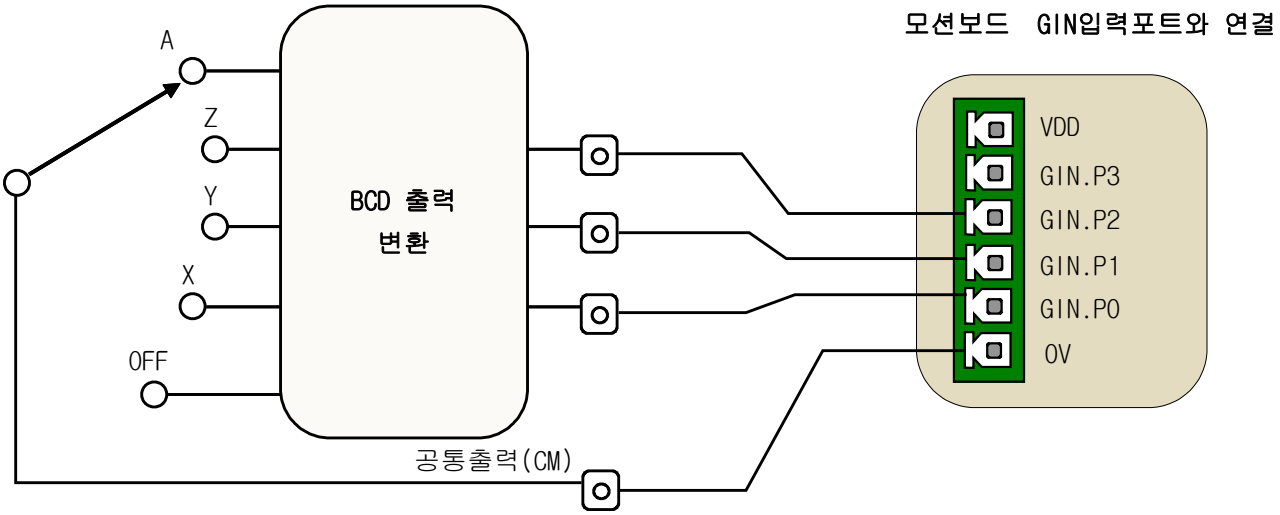
각 축 선택에 대한 AD값은 아래와 같이 MPG SETUP 윈도우에서 직접 확인 할 수 있으며 또한 모니터링화면의 ADC0 값표시창에서도 확인할 수 있습니다. 각 축에 대한 ADC값은 MPG설정에서 사용됩니다.





3. BCD 스위치 접점방식과 연결

BCD접점 출력은 SELECTOR 접점의 1:1접점 방식이 아니라 BCD코드에 의한 접점이 출력됩니다.
BCD코드는 각 접점을 2진법으로 취하고 십진수 0,1,2,...9 까지만을 표현하는 코드입니다.
축수가 최대 6개로 가정하면 2진수 접점은 최소 3개를 필요로 하며 축수를 3개로 가정한다면 접점은 최소 2개를 필요로 합니다.



- 출력접점 동작표 -

출력 선택	BCD.4	BCD.2	BCD.1
OFF 선택	0	0	0
X 선택	0	0	1
Y 선택	0	1	0
Z 선택	0	1	1
A 선택	1	0	0

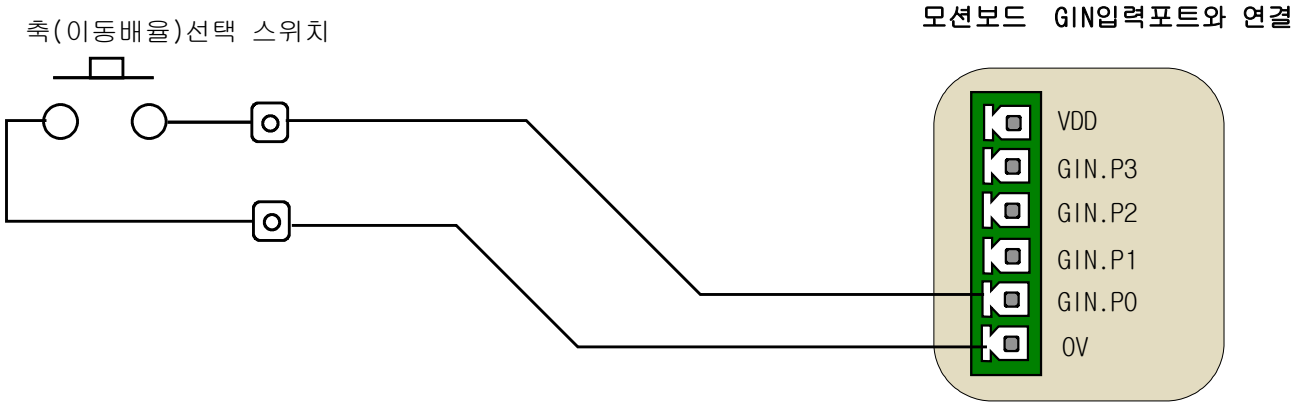
0 : 공통출력(CM)과 연결안됨
1 : 공통출력(CM)과 연결됨

4. ONE SWITCH 방식의 연결

축선택 동작과 이동배율선택 동작에 대한 GIO입력핀 사용의 최소화를 위해 하나의 스위치 입력(버튼동작)을 통해서 다양한 축의 선택과 이동배율을 선택할 수 있는 연결방식입니다.

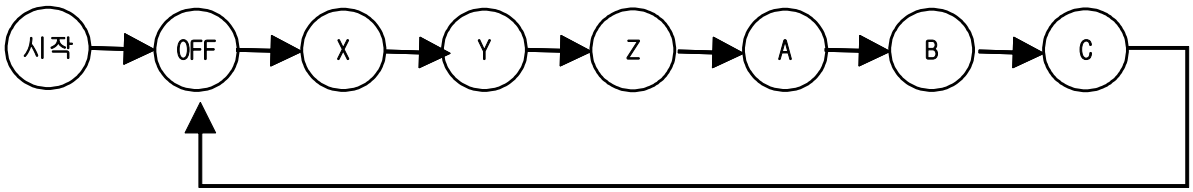
축선택의 경우 X축에 대한 GIO입력핀이 사용되고
이동배율선택의 경우 (이동1배율)x1 에 대한 GIO입력핀이 사용됩니다.

각 GIO입력핀이 상승엣지일 때 축(이동배율)은 다음 축(이동배율)로 바뀌게 됩니다.

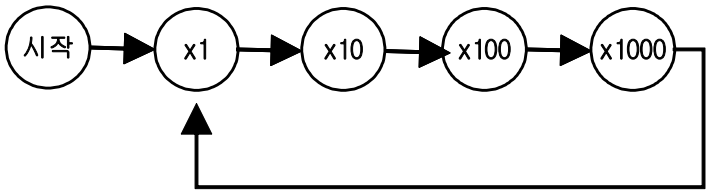


아래는 6축에 대한 각 축의 선택 변환순서와 이동배율의 선택 변환순서를 보여줍니다.

(1) 축의 선택에 대한 변환순서



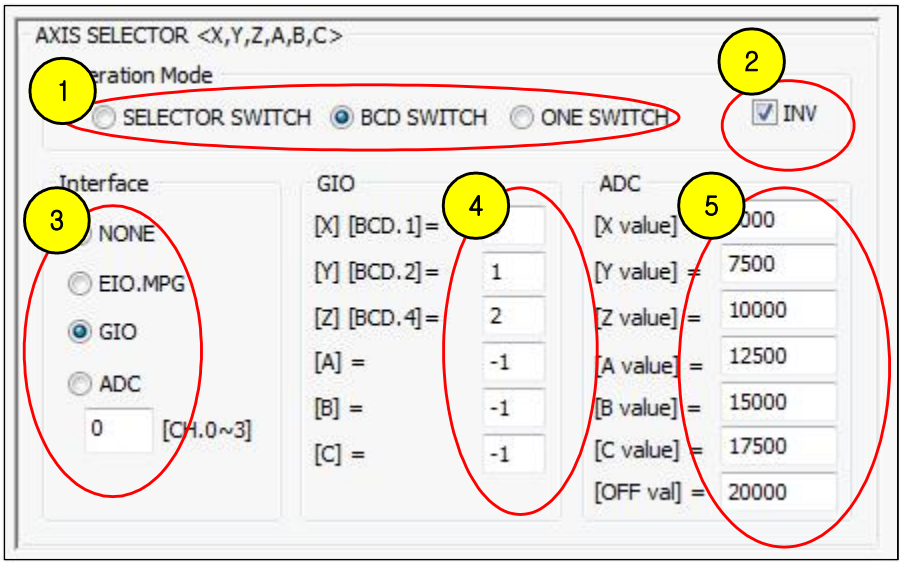
(1) 이동배율의 선택에 대한 변환순서



5. 축선택(이동배율선택) 설정

축선택스위치연결과 이동배율선택에 관련된 MPG 셋업윈도우의 설정내용이 동일함으로 축선택스위치 연결과 관련된 MPG설정내용을 기준으로 설명합니다.

축선택스위치과 관련되어 아래의 총 5가지 항목에 대한 설정을 필요로 합니다. 여기서 GIO연결과 ADC연결 중 하나만을 선택함으로 실제 총 4개의 선택사항이 필요합니다.
각 설정별로 살펴봅니다.



(1) 스위치 동작방식 설정 (Operation Mode)
- 셀렉터 스위치(Selector switch)형태의 출력방식
- BCD 스위치형태의 출력방식
- 단일 스위치형태의 출력방식

(2) 신호반전 여부
일반적으로 스위치동작은 0V(GND)와 연결되어 동작되며, 신호의 형태나 모션보드의 입력방식에 따라 신호를 반전시켜 입력으로 사용할 경우, 입력된 신호를 반전시켜 적용하도록 지시합니다.

(3) 모션보드 연결방식
- NONE : 연결없음
- EIO.MPG : 전용 Ext_Link통신(확장통신)기능을 사용한 MPG연결시
- GIO : GIO입력을 사용한 축 선택 기능연결
- ADC : ADC채널의 입력값을 사용한 축 선택 기능 연결

(4) GIO입력 연결방식일 때 각 스위치 접점신호의 입력포트 결정
해당하는 신호에 연결된 입력포트번호를 설정합니다. 현재 0~7번까지 사용가능하며, -1로 설정하며 해당 접점기능이 없음을 나타냅니다.

(5) ADC입력방식으로 축선택스위치기능을 사용할 때 각 축이 선택될 때마다 표현되는 ADC값을 기록합니다.
0보다 작은 음의 값을 사용하면 해당 축의 선택신호를 사용하지 않음을 나타냅니다.
각 기록값들은 가장 인접된 기록값과 비교되어 그 중간값까지를 유효한 입력값으로 적용합니다.(신호여유값 자동계산).

즉 X축 ADC 신호값을 2000 으로 설정하고 +방향으로 가장 인접된 임의축(Z축으로 가정)의 ADC 신호값이 3000 이고 -방향으로 가장 인접된 임의축(Y축으로 가정)의 ADC 신호값이 1000일 경우 아래와 같이 자동으로 X축의 신호범위값이 설정됩니다.

(Y축 선택영역) 1500 < X 축선택 ADC 신호범위 < 2500 (Z축선택영역)

G. 기타 연결 설정

엔코더휠, 축선택스위치, 이동배율선택 스위치와의 연결과 설정이 끝났다면, 추가적인 옵션으로 아래의 3가지 설정을 필요로 할 수 있습니다. 이들 설정은 MPG장치마다 있거나 없을 수 있습니다.

(1) 비상정지 스위치
(2) 휠 동작 ENABLE(가능) 스위치
(3) 휠 동작에 의한 축이동 반전옵션

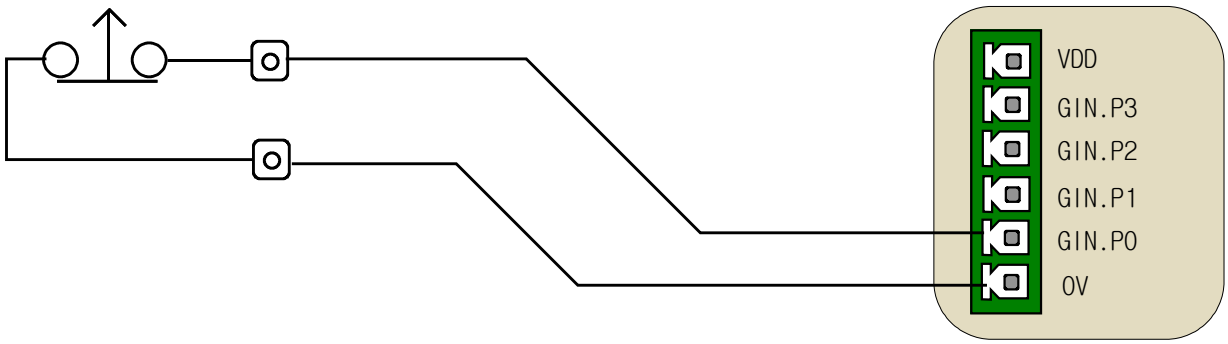
1. 비상정지(ESTOP) 스위치 연결 및 설정

위칸 모션보드는 ESTOP입력신호를 GIO입력신호와 관계없이 별도의 신호로 제공됩니다. BLX6보드와 모션트레이닝보드에서 제공되는 ESTOP신호는 기본적으로 A접점형식으로 ESTOP신호입력을 받아들입니다.

MPG의 비상정지신호가 A접점이면 전용의 ESTOP입력신호를 그대로 사용할 수 있습니다. 그러나 이 신호가 아닌 별도의 ESTOP기능 신호를 GIO 입력포트에 할당하고 한다면 아래와 같이 사용할 수 있습니다.

비상정지스위치

모션보드 GIN입력포트와 연결



상기와 같이 연결되었다면 아래와 같이 설정하여 동작할 수 있습니다.

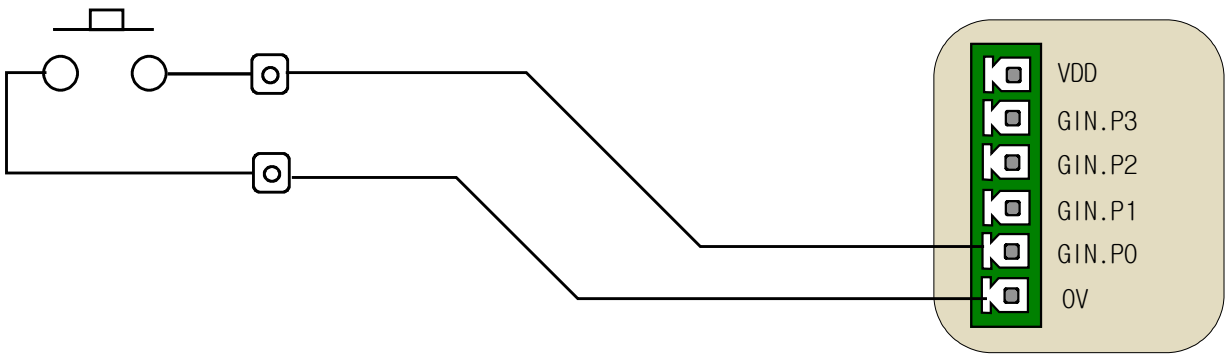
2. 동작가능(ENABLE) 스위치

동작가능 스위치는 MPG의 동작전체 혹은 엔코더휠의 동작을 금지하거나 허가하는 용도로 사용됩니다.
모션마스터 V13.R179버전은 MPG의 엔코더휠의 동작을 금지하며 허가하는 용도로 동작합니다.

ENABLE스위치의 연결은 아래와 같이 GIO입력포트 중 하나에 할당할 수 있습니다.

동작가능 스위치

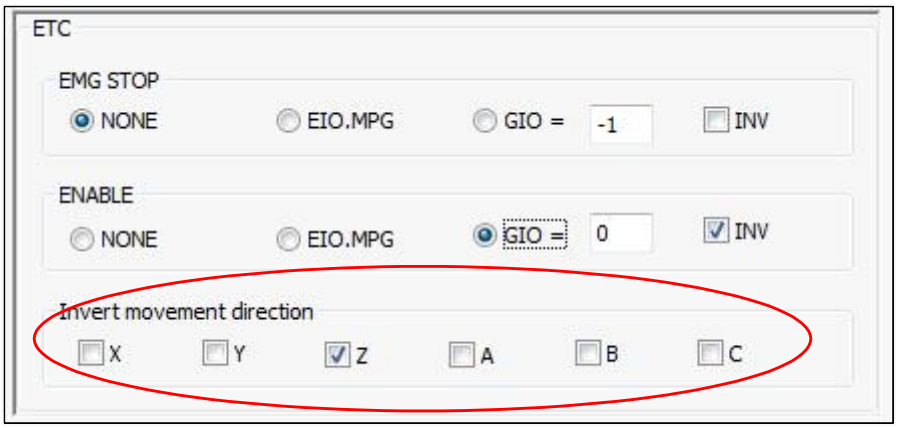
모션보드 GIN입력포트와 연결



상기와 같이 연결한 후 아래와 같이 설정하여 동작할 수 있습니다.

3. 휠 동작의 축이동 방향반전 옵션

기계장치의 특성마다 휠의 회전방향과 해당 선택축의 이동방향이 서로 반대방향의 느낌이 발생할 수 있습니다. 이때 휠의 회전방향에 대한 축의 진행방향을 반전시키고자 한다면 해당 축에 체크를 설정하여 사용합니다.



-끝-