

MW Motor Controller DC Mobile Application



Application Notes



※ 어플리케이션 노트에 포함된 정보는 정확하고 신뢰성이 있는 내용입니다. 그러나 출판 당시 발견되지 않은 오류가 있을 수 있으니 사용자는 자신의 제품 검증을 수행하시기 바라며, 전적으로 어플리케이션 노트에 포함된 정보에 의존하지 마시기 바랍니다.

1	소개	1
2	제어기 및 주변 부품 선정 안내	2
2.1	제어기 선정	2
2.2	커넥터 및 액세서리	3
2.2.1	MoonWalker MW DSub Cable	3
2.2.2	MoonWalker MW Encoder Line Driver	4
2.3	배터리 선정	5
2.3.1	배터리의 종류	5
2.3.2	배터리 선정시 고려사항	5
2.4	모터 선정	7
2.4.1	모터 선정시 고려사항	7
2.4.2	모터 용량 및 주요 사양 도출	8
3	모터 연결과 전압·전류 구동	10
3.1	제어기 단자	10
3.2	기본 연결	11
3.2.1	배터리 및 전원 연결	11
3.2.2	PC 연결	12
3.2.3	마이크로컨트롤러 연결	13
3.2.4	제어기 전원 투입	13
3.2.5	PC에서 USB 드라이버 설치	13
3.3	UI 유틸리티 다운로드와 실행	14
3.3.1	USB(VCP) 통신 설정과 연결	14
3.3.2	제품 버전 및 ID 확인	15
3.4	모터의 전압 구동	16
3.4.1	모터의 특성 설정	17
3.4.2	모터의 전압 구동 테스트	18

3.5	모터의 토크(전류) 구동	19
3.5.1	전류 제어기 파라미터 설정	19
3.5.2	전류 제어 테스트	20
4	엔코더 연결과 위치·속도 구동	22
4.1	증분 엔코더의 사용	22
4.1.1	광학식 증분 엔코더의 원리	22
4.1.2	모터에 엔코더 결합	23
4.1.3	엔코더 신호선 연결	24
4.1.4	엔코더 방향 확인	25
4.1.5	엔코더 노이즈 대책	25
4.2	모터의 속도 구동	26
4.2.1	속도 제어기 파라미터 설정	27
4.2.2	속도 제어 테스트	28
4.3	모터의 위치 구동	29
4.3.1	위치 제어기 파라미터 설정	29
4.3.2	위치 제어 테스트	30
5	이동로봇에 제어기 적용	31
5.1	2륜 차동 구동 로봇	31
5.1.1	로봇 기구부	31
5.1.2	로봇 배선	32
5.1.3	제어기 파라미터 설정	33
5.1.4	제어기 이득 동조	35
5.1.5	로봇의 주행 테스트	35
5.2	4륜 스키드 조향 로봇	36
5.2.1	로봇 기구부	36
5.2.2	로봇 배선	37
5.2.3	제어기 설정 및 테스트	37
5.3	조이스틱으로 로봇 조종	37
5.3.1	조이스틱 연결 구성	38

5.3.2	아날로그 입력 채널의 파라미터 설정	38
5.3.3	캘리브레이션 파라미터 설정	40
5.3.4	로봇의 조종	41
5.4	RC 조종기로 로봇 조종	42
5.4.1	RC 조종기 연결 구성	43
5.4.2	펄스 입력 채널의 파라미터 설정	44
5.4.3	캘리브레이션 파라미터 설정	45
5.4.4	로봇의 조종	45
6	아날로그·펄스 입력 포트의 사용	47
6.1	절대 엔코더의 사용	47
6.1.1	모터에 절대 엔코더의 연결	47
6.1.2	아날로그 출력형 절대 엔코더	48
6.1.3	PWM 출력형 절대 엔코더	49
6.1.4	절대 엔코더 테스트	51
6.2	외부 위치 센서 피드백	51
6.2.1	피드백 관련 파라미터 설정	52
6.2.2	스케일 변환 파라미터 및 제어기 이득 설정	52
6.2.3	위치 센서 피드백 테스트	54
7	디지털 입력 포트의 사용	56
7.1	비상 정지 버튼의 사용	56
7.1.1	비상 정지 버튼 배선	57
7.1.2	디지털 입력 채널의 파라미터 설정	57
7.1.3	비상 정지 버튼의 동작	58
7.2	정지 스위치의 사용	58
7.2.1	발 페달 스위치 배선	58
7.2.2	디지털 입력 채널의 파라미터 설정	59
7.2.3	발 페달 스위치의 동작	59
7.3	홈 센서의 사용	60
7.3.1	홈 센서 배선	61

7.3.2	디지털 입력 채널의 파라미터 설정	61
7.3.3	홈 센서의 동작	62
7.4	리미트 센서의 사용	63
7.4.1	리미트 센서 배선	63
7.4.2	디지털 입력 채널의 파라미터 설정	64
7.4.3	리미트 센서의 동작	64
7.5	홈 센서와 리미트 센서의 사용	65
7.5.1	홈 센서와 리미트 센서 배선	65
7.5.2	디지털/아날로그 입력 채널의 파라미터 설정	66
7.5.3	홈 센서와 리미트 센서의 동작	67
7.6	근접 센서 및 포토 인터럽터의 사용	67
7.6.1	근접 센서 배선	68
7.6.2	포토 인터럽터 배선	69
8	디지털 출력 포트의 사용	71
8.1	모터 브레이크 사용	72
8.1.1	모터 브레이크 배선	72
8.1.2	디지털 출력 채널의 파라미터 설정	73
8.1.3	모터 브레이크 작동 확인	74
8.2	다이나믹 브레이크	74
8.2.1	다이나믹 브레이크 배선	75
8.2.2	디지털 출력 채널의 파라미터 설정	76
8.2.3	다이나믹 브레이크 작동 확인	76
8.3	회생 브레이크	77
8.3.1	회생 브레이크 배선	77
8.3.2	디지털 출력 채널의 파라미터 설정	78
8.3.3	회생 브레이크 작동 확인	79
8.4	제어기에 냉각팬 사용	80
8.4.1	냉각팬 배선	80
8.4.2	디지털 출력 채널의 파라미터 설정	81
8.4.3	냉각팬 작동 확인	82

9	스크립트 작성과 응용	84
9.1	모터 구동 파형 생성과 테스트	84
9.1.1	파형 생성 개요	84
9.1.2	스크립트 작성	85
9.1.3	빌드 및 다운로드	86
9.1.4	구동 테스트	87
9.2	원점 복귀	89
9.2.1	스크립트 작성	90
9.2.2	빌드 및 다운로드	91
9.2.3	구동 테스트	91
9.3	컨베이어 구동	92
9.3.1	컨베이어 개요	92
9.3.2	스크립트 작성	94
9.3.3	구동 테스트	95
9.4	모터 온도 측정과 냉각	96
9.4.1	모터 온도 측정과 냉각 회로 구성	96
9.4.2	I/O 채널의 파라미터 설정	97
9.4.3	스크립트 작성	98
9.4.4	구동 테스트	100
9.5	2륜 차동 구동 로봇 운전	101
9.5.1	2륜 차동 구동 로봇 개요	101
9.5.2	2륜 차동 구동 로봇 기구학	102
9.5.3	스크립트 작성	104
9.5.4	구동 테스트	105
9.6	전방향 로봇 운전	106
9.6.1	전방향 로봇 개요	106
9.6.2	전방향 로봇 기구학	107
9.6.3	전방향 로봇 배선	109
9.6.4	스크립트 작성	110
9.6.5	구동 테스트	112

10	관련 자료	113
11	문서 변경 이력	114

1 소개

어플리케이션 노트는 제어기 및 주변 부품 선정부터 제어기의 기능 사용 방법 그리고 응용에 대한 정보를 제공합니다. 사용자는 각 절의 순서대로 모든 내용을 습득하거나 필요한 정보만 선택해서 습득하면 됩니다.

어플리케이션 노트는 다음과 같은 순서로 구성되어 있습니다:

2장. 제어기 및 주변 부품 선정 안내:

제어기와 배터리 그리고 모터를 선정하는 방법에 대해 설명합니다.

3장. 모터 연결과 전압 전류 구동:

제어기에 전원과 모터를 연결하고 UI 유틸리티로 PC에 연결하는 방법에 대해 설명합니다. 그리고 전압과 토크(전류) 구동을 위한 설정 및 테스트 방법에 대해 설명합니다.

4장. 엔코더 연결과 위치속도 구동:

증분 엔코더에 대한 전반적인 내용과 속도와 위치 구동을 위한 설정 및 테스트 방법에 대해 설명합니다.

5장. 이동로봇에 제어기 적용:

2륜 차동 구동 로봇과 4륜 스키드 조향 로봇에 설정 및 테스트 방법에 대해 설명합니다. 그리고 조이스틱과 RC 조종기를 이용해서 로봇을 조종하는 방법과 설정에 대해 설명합니다.

6장. 아날로그펄스 입력 포트의 사용:

절대 엔코더 값과 외부 위치 센서값을 피드백 받는 방법과 설정에 대해 설명합니다.

7장. 디지털 입력 포트의 사용:

비상 정지 버튼, 발 페달 스위치, 홈 센서와 리미트 센서 그리고 근접센서 및 포토 인터럽터를 사용하기 위한 외부 회로 구성과 설정에 대해 설명합니다.

8장. 디지털 출력 포트의 사용:

모터 브레이크, 다이내믹 브레이크, 회생 브레이크, 제어기 냉각팬을 사용하기 위한 외부 회로 구성과 설정에 대해 설명합니다.

9장. 스크립트 작성과 응용:

스크립트 작성 방법과 원점 복귀, 컨베이어 구동, 모터 온도 측정 및 냉각, 연진자, 각종 모바일 로봇 구동에 대한 스크립트 예제 소스를 제공합니다.

※ 제어기 모델에 따라 전원 및 모터 그리고 외부 장치 연결 구성 방법이 다를 수 있으니, 제품별 데이터시트와 사용자 매뉴얼에서 모터 및 전원 연결과 외부 장치 구성 방법에 대한 내용을 숙지하시기 바랍니다.

2 제어기 및 주변 부품 선정 안내

본 장에서는 사용자의 응용(application)에 따라 구매 가능한 DC모터 제어기 모델과 DC 모터, 주변 부품들에 대하여 소개합니다.

2.1 제어기 선정

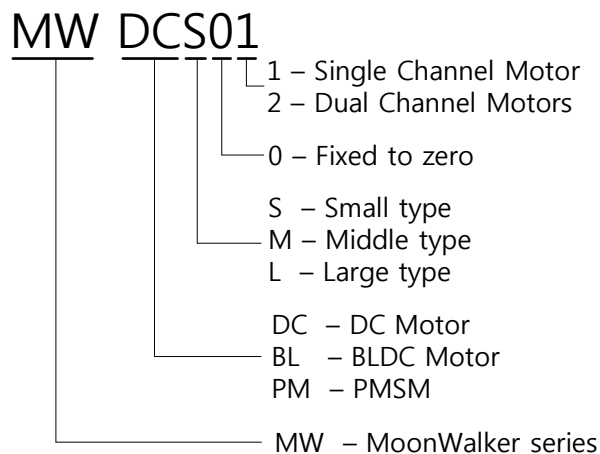
(주)엔티렉스에서 DC모터의 위치, 속도, 전류(토크), 전압 제어를 위해 MoonWalker로 명명된 소형/중형/대형 싱글/듀얼 채널 제어기 시리즈를 제공합니다. 다음 표 2-1에서 제어기 모델별 주요 기능을 요약하고 있습니다.

표 2-1 MoonWalker 제어기 스펙

Controller Model	Operation Voltage	Current		Num. Motors	IO Port			
		Max	Surge		D-OUT	D-IN	A-IN	Pul-IN
MDC24D100S	8~30 VDC	3A	<10A	1	-	-	-	-
MDC24D100D	8~30 VDC	3A	<10A	2	-	-	-	-
MDC24D200S	8~30 VDC	10A	<20A	1	2	2	1	1
MDC24D200D	8~30 VDC	10A	<20A	2	2	2	2	2
MDC24D500S	8~50 VDC	80A	<150A	1	2	2	1	1
MDC24D500D	8~50 VDC	40A	<150A	2	2	2	2	2

- MoonWalker 소개: <http://www.ntrexgo.com/moonwalker-%EC%86%8C%EA%B0%9C>
- MoonWalker 판매: <http://www.devicemart.co.kr/goods/brand.php?seq=1475>

다음은 제어기 모델 명에 대한 세부 설명입니다:



MoonWalker 제어기는 모터 수에 따라 싱글 채널과 듀얼 채널로, 용량에 따라 S(Small), M(Middle), L(Large)로 구분됩니다.

2.2 커넥터 및 액세서리

제어기 모델별로 다음과 같은 엔코더 커넥터와 통신 커넥터, 액세서리들을 부가적으로 연결하여 사용할 수 있습니다.

표 2-2 제어기 모델에 따른 엔코더 커넥터 및 액세서리

	엔코더 커넥터	통신 커넥터 (RS232,CAN)	액세서리
MDC24D100S		[YEONHO]SMH200-03	-
MDC24D100D			-
MDC24D200S	[YEONHO]SMH200-04	-	- MoonWalker MW DCM01 Cable
MDC24D200D		-	- MoonWalker MW DSub Cable
MDC24D500S	[Molex] 43025-0600	-	- MoonWalker MW Encoder Line Driver
MDC24D500D		-	

- 관련 액세서리 판매: <http://www.devicemart.co.kr/goods/list.php?category=006011007>
- DS1033-15M(3열) 판매: <http://www.devicemart.co.kr/goods/view.php?seq=7460>

2.2.1 MoonWalker MW DSub Cable

MoonWalker MW DSub Cable은 주로 많이 사용하는 RS232 커넥터와 RC소켓으로 구성된 케이블입니다. 사용자는 제품구매와 동시에 RS232통신과 RC 조종기를 바로 사용하길 원하면 MoonWalker MW DSub Cable 액세서리를 구매하면 됩니다.

만약 다른 핀(CAN 또는 I/O 핀)을 사용하고 싶다면 MoonWalker MW DSub Cable을 분해해서 사용하고자 하는 핀에 와이어를 납땜한 후 사용합니다. 또는, DS1033-15M(3열) 커넥터를 구매해서 사용해도 됩니다.

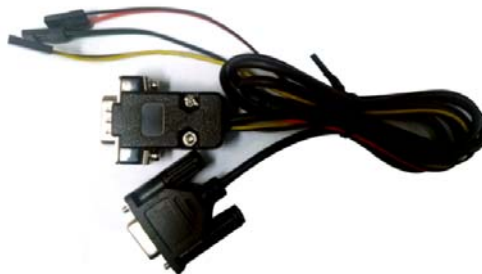


그림 2-1 MoonWalker MW DSub Cable

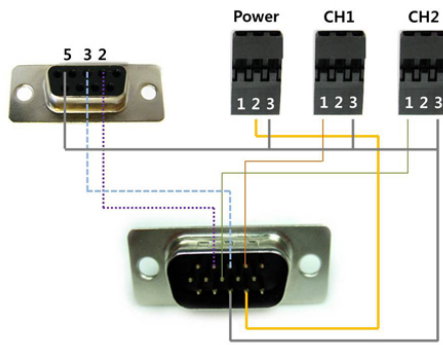


그림 2-2 MoonWalker MW DSub Cable 배선도

2.2.2 MoonWalker MW Encoder Line Driver

로타리 엔코더 출력(회로) 방식에는 오픈 콜렉터(Open Collector), 전압 출력(Voltage Output), 토템 폴(Totem Pole), 콤플리멘탈(Complemental or Push-Pull Output), 라인 드라이버 출력방식(Line Driver)이 있습니다.

이 중에서 라인 드라이버 출력방식은 A/A; B/B; Z/Z와 같이 출력상이 각각 +와 - 상 1조로 이루어져 있기 때문에 고속 응답과 노이즈 특성이 강하며, 주로 장거리 전송에 사용됩니다. 이러한 라인 드라이버 출력방식은 주로 AC Servo System, DC Servo System, 로봇, NC 공작기계 같은 분야에서 많이 사용됩니다.

MoonWalker MW Encoder Line Driver는 26LS32 IC 칩을 사용한 라인드라이버 방식의 제품으로, 제 여기에 라인 드라이버 출력방식을 가진 엔코더를 연결하기 위해 사용합니다.



그림 2-3 MoonWalker MW Encoder Line Driver

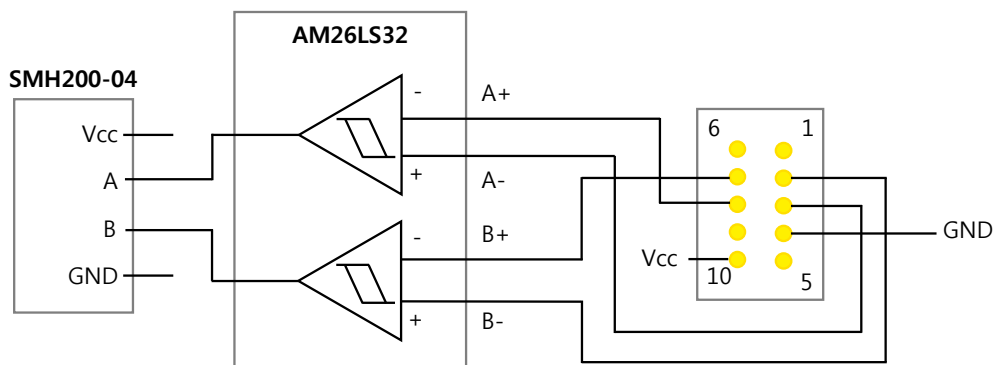


그림 2-4 MoonWalker MW Encoder Line Driver 회로도

2.3 배터리 선정

사용자는 제어기와 구동할 모터의 전압과 구동전류 그리고 배터리의 스펙을 고려하여 전체 제품 구성에 맞는 크기와 충분한 출력을 내는 배터리를 선정하시기 바랍니다.

2.3.1 배터리의 종류

배터리는 일반적으로 연(납)축전지(Lead), 리튬인산철(LiFePO₄), 리튬폴리머(Li-Po) 배터리를 많이 사용합니다.

연(납)축전지:

연(납)축전지는 대용량과 다양한 크기로 제작할 수 있으며, 충전 효율(방전/충전=70%)이 우수합니다. 또한, 과충전 저항력이 강하며, (밀폐형이 아닌 이상) 쉽게 충전상태를 확인할 수 있는 장점이 있습니다. 그래서 아직까지 안정된 품질, 신뢰성 그리고 경제성 때문에 많은 곳에서 사용하고 있습니다. 그러나 비 친환경적(중금속)이고 작은 크기의 제작이 어려우며 충전상태 지속성이 낮은(자가방전 월 5~20%) 단점을 가지고 있습니다.

리튬인산철 배터리:

리튬인산철 배터리는 극판을 인산철로 사용하는 리튬이온 계열의 배터리로 가장 안전하고 가볍고 친환경적입니다. 또한, 긴 충방전 수명을 가지고 있고(충방전 수명 2,000회) 고용량 충전이 가능하며 낮은 자가 방전율(월 1%의 자가방전)을 가지고 있습니다. 그러나 연(납)축전지에 비해 가격이 비싼 단점이 있습니다.

참고로 리튬인산철 배터리는 연(납)축전지에 비해 가격이 비싸나 3배 긴 수명과 1.8배 더 많은 실질 출력 전력량을 고려하면 오히려 더 저렴해질 수도 있습니다.

리튬폴리머 배터리:

리튬폴리머 배터리는 높은 에너지 저장 밀도(같은 크기에 더 큰 용량을 가짐)를 가지고 뛰어난 온도 특성을 가지고 있으나 전해질이 액체로 흘러나오거나 폭발할 위험 가능성도 있습니다.

※ 리튬폴리머를 사용 시 리튬폴리머 전용 보호회로를 사용하시기 바랍니다.

2.3.2 배터리 선정시 고려사항

배터리를 선정할 때, 많은 것을 고려해야 하지만 기본적으로 배터리의 데이터시트에서 기재된 전압, 전류, 평균/최대 방전율(C) 등을 주의 깊게 살펴봐야 합니다. 예를 들어 로봇을 구동하기 위해 아래 스펙과 같은 리튬 폴리머 배터리 사용한다고 가정합니다.

리튬 폴리머 배터리:

- 전압 - 11.1V
- 전류 - 1200mA

- 평균/최대 방전율(C) - 10C/15C
- 무게 - 150g

로봇을 구동하기 위한 출력은 W(전압*전류)의 의미로 해석해서 계산하며, 우리는 데이터시트에서 제공하는 정보를 통해 다음과 같은 사실을 알 수 있습니다.

- 1시간 동안 11.1V와 1.2A를 사용. 즉 11W로 1시간 동안 로봇을 구동 가능 (물론 11W로 150g 배터리를 싣고 구동되는 로봇은 없음)
- 10C의 방전율을 가지는 배터리이므로 12A(1.2A*10)까지는 안정적으로 공급할 수 있고, 약 120W(11.1V*12A)로 6분(1C=배터리 전류용량/1h)동안 연속 구동이 가능
- 만약 위에 최대값을 넘어가는 전류를 사용하거나 제품의 업체에서 지정한 15C/15초를 넘기면 배터리는 영구 파손이 됨

우리는 위와 같은 사실을 파악한 후 제품 전체 구성에 맞는 배터리 용량을 선정해야 합니다.

또한, 위와 같은 사실 이외에 추가적으로 아래의 항목들도 고려해서 배터리를 선정해야 합니다.

배터리 장시간 사용:

배터리를 장시간 사용할 경우 배터리 용량, 방전 전류, 방전 시간의 관계식을 통해서 산출된 수치의 2~3배의 용량을 가진 배터리로 선정해야 합니다. 그리고 사용기간, 방전전류에 대해서는 산출된 시간, 전류의 1/2 ~ 1/3의 수치로 사용해야 합니다.

배터리 방전전류와 방전시간:

배터리는 방전되는 전류에 따라 방전시간이 달라집니다. 예를 들어 1시간에 10A씩 흘러보낼 때 방전시간이 2시간이 걸리는 배터리가 있다고 가정합니다. 만약 이 배터리를 1시간에 20A씩 흘러보낸다면 방전시간은 1/2배인 1시간으로 줄어드는 게 아니라 배터리 종류와 방전 전류 크기에 따라 방전시간은 2배 이상으로 줄어들기 때문에 이를 장시간 연속 사용 시 고려해야 합니다.

제품의 무게와 무게중심:

구동하고자 하는 제품의 무게와 무게중심에 따라 필요한 출력량(W)은 달라집니다. 따라서 사용자는 무게에 따른 출력량을 테스트를 통해서 알아보는 것이 정확합니다. (똑같은 무게의 2륜 모바일 로봇(지상)과 쿼드콥터(하늘)가 있다고 가정할 때 이 두 제품을 움직이기 위한 출력량은 무게중심에 따라 달라질 수 있습니다)

배터리 테스트 필요:

배터리 방전전류에 따른 시간은 사용자 환경(배터리 완충상태, 주변온도 등)에 따라 달라지기 때문에 데이터시트 정보가 아닌 직접 배터리를 테스트해서 알아보는 것이 정확합니다.

사용자고자 하는 용량의 배터리가 없다면:

사용자가 원하는 용량의 배터리가 없으면 직접 완충된 배터리를 직렬과 병렬로 연결해서 사용합니다.

아래 디바이스마트 홈페이지에서는 다양한 배터리를 구매하실 수 있습니다.

- 디바이스마트 배터리 판매: <http://www.devicemart.co.kr/goods/list.php?category=003002>

※주의※ 본 어플리케이션 노트는 기본적인 배터리 선정 방법에 대해 기술되어 있습니다. 사용자는 자신의 제품 검증을 수행하시기 바라며, 전적으로 어플리케이션 노트에 포함된 정보에 의존하지 마시기 바랍니다.

※주의※ 배터리의 잘못된 구성으로 인한 제품 손상 및 파손은 책임지지 않습니다.

2.4 모터 선정

DC 모터는 출력이 몇 와트인지 또는 몇 마력인지, 토크와 부하중량이 얼마인지 고려해서 선정해야 합니다. 본 절은 모터 선정시 고려사항과 모터 용량 및 주요 사양 도출 방법에 대해서 설명합니다.

2.4.1 모터 선정시 고려사항

먼저 모터를 선정하기 전에 사용자는 모터의 출력, 전압(정격전압), 토크, 회전수를 결정합니다. 그리고 난 후 모터의 데이터시트 정보와 입력량, 출력량, 효율 기본 공식을 가지고 정격 부하(Rated torque), 정격회전수(Rated torque), 정격 전류(Rated current), 기동토크(Starting torque), 무부하 회전수(No load speed) 값을 계산한 후 이에 맞는 모터를 선정합니다.

아래는 모터를 선정하기 전에 사용자가 결정해야 할 항목들입니다.

모터의 전원(정격전압) 결정:

DC 모터는 일반적으로 DC12V와 DC24V 모터를 많이 사용합니다. 제품의 전체 구성에 따라 모터의 정격전압을 결정하겠지만 일반적으로 배터리를 사용하거나 많은 토크를 필요로 할 경우 모터의 전류량 때문에 보통은 DC24V를 사용합니다. (전류량이 높으면 배터리 소모가 많고 전기 회로의 용량이 커져야 하며, DC24V 모터보다 DC12V 모터가 전류량은 2배 정도 높습니다.)

※ 일반적으로 250W 미만은 DC12V를 250W 이상은 DC24V 모터를 사용합니다.

모터의 회전수 결정:

통상 모터의 회전수는 분당 회전수인 rpm 단위를 사용하며, 대부분의 모터가 rpm모터로 되어 있습니다.

모터의 회전수는 속도 조절이 필요한 경우 최고 속도(rpm)의 약 10~30% 정도 올려서 선정하고 만약 속도 조절할 범위가 넓은 경우에는 모터의 토크를 넉넉히 잡아서 선정합니다.

모터의 토크와 와트(Watt) 결정:

통상 모터의 토크는 kg.cm 단위를 사용하며, 이는 1kg 무게의 물체를 반지름이 1cm 도르레로 끌어올리는 힘이라고 생각하시면 됩니다.

모터의 토크는 필요한 모터의 토크보다 40% 정도 올려서 선정하고 만약 속도 가변이 필요한 경우에는 100% 정도 올려서 선정하기를 권장합니다. (저속에서 무리 없이 운전해야 할 경우)

모터의 와트는 토크와 전류량에 비례합니다. 이는 사용하고자 하는 모터의 데이터시트에 토크표를 참조하시기 바랍니다.

※ 사용자는 꼭 회전수와 토크를 결정하고 모터를 선정해야 합니다. 예를 들어, 모터 출력이 100W(백와트)라고 하면 $10,000 \text{ (rpm)} \times 0.974 \text{ (kg.cm)} = 1000 \text{ (rpm)} \times 9.74 \text{ (kg.cm)}$ 가 되지만, 모터의 회전수는 1000 ~ 10,000 rpm의 범위가 되기 때문에 모터의 선정 범위가 넓어지게 됩니다.

모터의 운전 특성 결정:

속도 가변이 필요한지, 장시간 운전이 필요한지, 정지 시 미끄러짐이 없어야 하는지, 좌우 회전 전환이 필요한지, 환경에 따른 여러 가지 필요한 동작들을 결정합니다.

예를 들어 빈번한 좌우 회전 및 급출발 제동이 필요할 경우 기어 헤드가 견딜 수 있는 허용 토크를 넉넉히 잡아서 모터를 선정해야 합니다. 오르막길이나 큰 토크가 필요로 하는 경우에는 필요한 모터의 토크에서 40~100%까지 올려서 모터를 선정해야 합니다. 이는 사용자 제품 구성 환경에 따라 달라지기 때문에 주변환경을 잘 고려해야 합니다.

모터의 치부 및 크기:

모터의 크기가 제한이 되는지 제한이 없는지 결정하시고 제한이 있다면 모터를 선정할 때 고려해야 합니다.

2.4.2 모터 용량 및 주요 사양 도출

위와 같은 항목들이 결정된 후 모터의 데이터시트와 아래 기본 공식을 통해서 모터의 정격 토크(Rated torque), 정격회전수(Rated torque), 정격 전류(Rated current), 기동 토크(Starting torque), 무부하 회전수(No load speed) 값을 계산합니다. 그리고 이에 맞는 모터를 최종적으로 선정합니다:

- 입력 량 (Wi) = 전압 (V) x 전류 (A)
- 출력 량 (Wo) = $0.01027 \times \text{회전수(rpm)} \times \text{토크 (kg.cm)}$
- 모터 효율 (%) = 입력량 (Wi) / 출력량 (Wo)

모터는 부하(Load)의 변동에 따라 흐르는 전류량(Current)도 변하게 되므로, 같은 모터에서도 공차(Tolerance)가 커서 다르게 표시될 수 밖에 없습니다. 모터의 데이터시트에 있는 모터 성능 곡선(Motor performance curve)을 보면, 일반적으로 모터의 최고출력(Max. output)은 기동토크(Starting torque)의 50% 위치, 모터의 최고 효율(Max. Efficiency)은 기동토크의 대략 15% 위치가 되는데, 가능한 최고 효율구간에서 사용하는 것이 바람직합니다.

※ 제품마다 모터 성능이 다르니 꼭 데이터시트를 숙지하시기 바랍니다.

예를 들어 아래와 같은 특성을 가진 40W 모터를 선정한다고 가정합니다. (위에서 이야기한 대로 모터를 선정하기 전에 미리 모터의 정격전압과 정격부하 회전 수는 사용자가 결정해야 합니다)

150W 모터 선정:

- 정격 전압 - 24 V
- 정격 토크 - 5 kg.cm
- 정격 회전수 - 3,000 rpm

위와 같이 정했다면 다음과 같이 계산 할 수 있습니다.

- 정격 출력(Rated Output) - $0.01027 \times 3000 \times 5 = 154.05 \text{ W}$
- 기동 토크(Starting Torque) - $5 / 0.15 = 33.33 \text{ kg.cm}$
- 무부하 회전수(No Load Speed) - $(33.33 \times 3000) / (33.33 - 5) = \text{약 } 3,530 \text{ rpm}$
- 최고 출력(Max. Efficiency Output) - $0.01027 \times (3530 / 2) \times (33.33 / 2) = 302.08 \text{ W}$

보통 DC 모터의 최고 효율은 모터 종류에 따라 40~85% 정도이며, PMDC(영구자석형) 모터라면 효율이 65% 정도입니다. 만약 영구자석형 모터라고 가정했을 때 모터에 인가되는 전력과 전류는 다음과 같습니다:

- 모터에 인가되는 전력 - $154.05 / 0.65 = 237 \text{ W}$
- 정격 전류 - $237 / 24 = 9.875 \text{ A}$

위와 같은 계산을 통해서 최종적으로 150W급 DC모터 선정에 있어 필요한 스펙은 아래와 같으며 이에 만족하는 모터를 선정하시면 됩니다.

- 정격 전압 - 24 V
- 정격 전류 - 9.875 A
- 정격 토크 - 5 kg.cm
- 기동 토크 - 33.33 kg.cm
- 정격 회전수 - 3,000 rpm
- 무부하 회전수 - 3,530 rpm

※ 경우에 따라 모터의 속도를 조절해서 사용하려면 속도 조절기나, 감속기가 연결된 모터를 사용하시면 됩니다.

※주의※ 본 어플리케이션 노트는 기본적인 모터의 선정 방법에 대해 기술되어 있습니다. 모터는 제조사마다 모터 선정 방법이 다를 수 있기 때문에 사용자는 제조사 홈페이지에서 모터의 데이터 시트와 설명서를 숙지한 후 모터를 선정하시기 바랍니다. 또한, 어플리케이션 노트에 포함된 정보에 의존하지 마시고 전체 제품 구성에 맞게 선정 또는 사용하시기 바랍니다.

3 모터 연결과 전압·전류 구동

이전 장의 제어기 및 부품 선정 안내에 따라 제어기와 모터, 배터리를 준비하였다면 서로 연결하여 구동해 봐야 합니다.

이번 장에서는 배터리(또는 파워서플라이)와 모터를 제어기에 연결하고 PC에서 USB로 제어기에 연결하여 Motor Control UI 유틸리티로 모터를 구동해보는 가장 기본적인 내용을 다룹니다.

이번 장에서 필요한 준비물은 다음과 같습니다:

- Windows OS가 설치된 노트북 또는 PC
- MoonWalker DC 모터 제어기
- DC 모터
- 배터리 or 파워서플라이
- Mini A 타입의 USB 케이블
- 약간의 전선 (허용 Ampere 확인)
- 일(-)자와 십(+)자 드라이버
- 인두, 납

3.1 제어기 단자

그림 3-1과 그림 3-2에서와 같이 제어기에 모터와 전원을 연결하는 단자는 스크류 터미널 블록 (Screw Terminal Block) 혹은 전선(Wire)으로 구성됩니다.

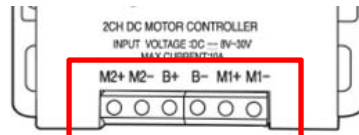


그림 3-1 스크류 터미널 블록 연결 (예: MW-MDC24D200D)

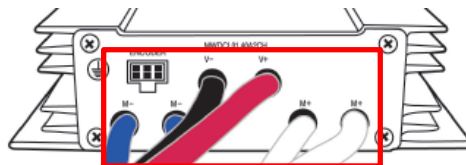


그림 3-2 전선 연결 (예: MW-MDC24D500D)

제어기에 배터리를 연결하는 단자는 B+(적색 전선)와 B-(흑색 전선)로 표시됩니다. 그리고 모터를 연결하는 단자는 M+(백색 전선)와 M-(청색 전선)로 표시됩니다. 듀얼 채널 제어기의 경우 모터를 연결하는 단자는 M1+와 M1- 및 M2+와 M2-로 표시됩니다.

그리고 PC나 마이크로컨트롤러와 통신은 위해 USB, RS-232, CAN 포트를 가지고 있습니다. 하지만 일부 모델(MW-MDC24D200S)의 경우 USB 포트가 제공되지 않습니다.

3.2 기본 연결

그림 3-3은 모터를 구동하기 위해 제어기에 기본적으로 연결되는 배터리와 모터, 노트북(또는 PC)에 대한 배선도입니다. 이 배선도는 제어기를 사용하는 단순한 예제일 뿐이며, 제어기의 응용에 따라 추가적인 회로 구성이 필요합니다.

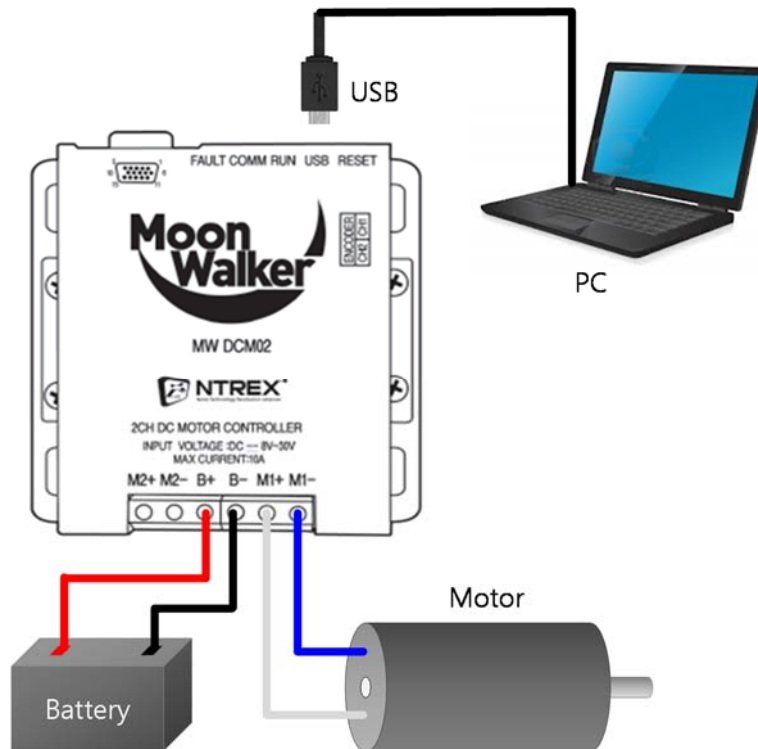


그림 3-3 제어기와 배터리, 모터, USB 연결

※주의※ 그림 3-3의 배선도에서는 안전을 위한 구성이 반영되어 있지 않습니다. 사용자는 제어기의 사용 환경에 따라 적절한 안전사항이 반영되어야 합니다.

3.2.1 배터리 및 전원 연결

제어기에서 전원과 모터를 연결하는 스크류 터미널 블록의 경우, 그림 3-4에서와 같이 십자(+) 또는 일자(-) 드라이버를 사용하여 나사를 풀어준 후, 5mm정도의 피복을 벗긴 전원 선과 모터 선을 삽입하고 나사를 조입니다.



그림 3-4 스크류 터미널 블록에 전선 연결

전원과 모터를 연결하기 위해 전선이 사용되는 경우는 배터리 또는 모터의 단자에 맞는 터미널을 전선에 고정하여 연결하거나 직접 납땜으로 연결합니다. 이 때 배선이 여러 갈래로 분기 되거나 전선의 길이를 연장해야 할 때는 링형 또는 포크형 터미널을 전선에 고정한 후 터미널 블록에 연결합니다.



그림 3-5 터미널 블록을 사용한 전선의 분기와 연장

※주의※ 전원 및 모터를 배선할 때는 전선의 허용전류가 모터에 흐르는 최대 전류보다 높은 전선을 사용하여야 합니다. 그렇지 않은 경우, 전선에서 발열 및 화재의 위험이 있습니다.

3.2.2 PC 연결

PC에서 USB 포트를 통해 제어기에 연결할 때는 USB 케이블을 사용합니다. 케이블의 PC 쪽은 가장 일반적인 USB A 타입 커넥터고 제어기 쪽은 Mini A 혹은 B 타입의 커넥터입니다.



그림 3-6 Mini A/B 타입의 USB 케이블

일부 모델(MW-MDC24D200S)의 경우 USB 포트가 제공되지 않습니다. 이 때는 그림 3-7과 같이 USB2Serial 컨버터를 사용하여 제어기의 RS-232 포트에 연결합니다.

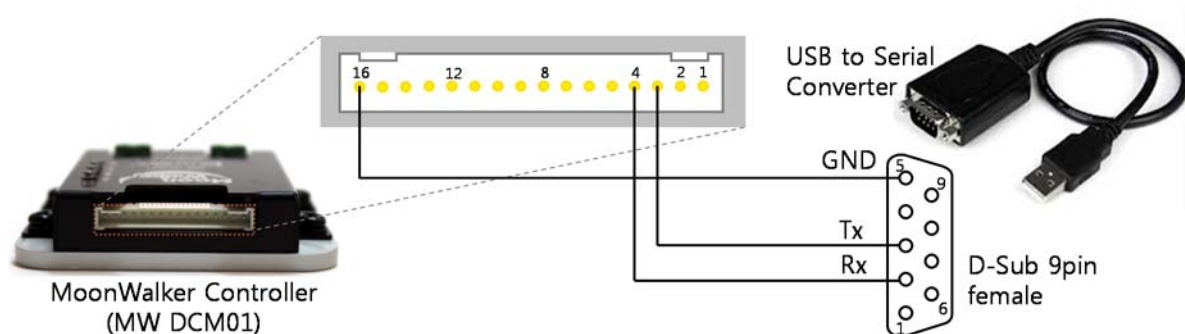


그림 3-7 MW-MDC24D200S 모델과 USB2Serial 컨버터의 연결

3.2.3 마이크로컨트롤러 연결

최근 들어 Arduino, Beaglebone black, Raspberry Pi 등 다양한 단일보드 마이크로컨트롤러들이 다양하게 활용되고 있습니다. 이러한 마이크로컨트롤러는 보통 TTL 레벨의 UART 통신 포트를 가지고 있기 때문에, 그림 3-8에서와 같이 제어기의 RS-232 포트에 연결하기 위해서는 RS-232 to TTL Converter가 필요합니다.

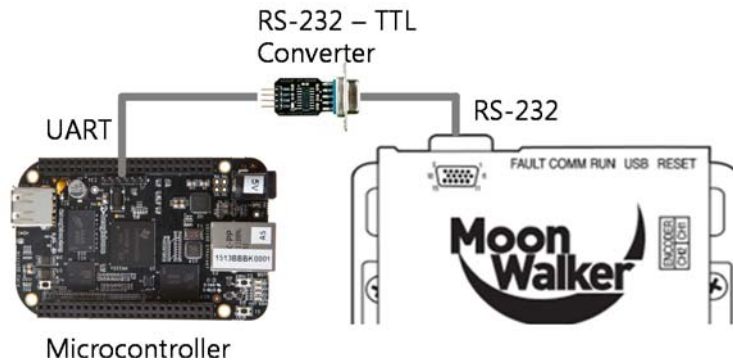


그림 3-8 마이크로컨트롤러 연결

RS-232 to TTL Converter는 온라인 전자부품 쇼핑몰에서 구입할 수 있습니다:

- RS-232 to TTL Converter: <http://www.devicemart.co.kr/goods/view.php?seq=1067177>

3.2.4 제어기 전원 투입

제어기에 전원(배터리 혹은 파워서플라이)을 연결하였다면, 스위치를 켜서 제어기에 전원을 투입합니다. 제어기가 정상적으로 구동되는 상황에서는 RUN(청색) LED가 1초 주기로 깜박입니다.

만약 모든(RUN, FAULT, COMM) LED가 꺼져있거나 켜져 있는 경우에는 제어기가 정상적으로 시작되지 않은 경우입니다. 이런 경우에는 사용자 매뉴얼의 "1.3.2 표시등" 내용을 참고하여 적절히 대응합니다.

3.2.5 PC에서 USB 드라이버 설치

PC와 제어기를 USB 케이블로 연결하면 PC에는 자동으로 제어기의 USB 드라이버와 VCP 드라이버가 설치됩니다. 그림 3-9은 드라이버 설치가 완료되었을 때 표시되는 윈도우 메시지입니다.

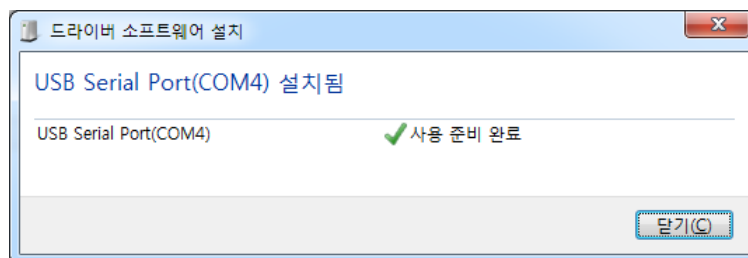


그림 3-9 USB 드라이버 설치 완료

제어기의 USB 드라이버가 정상적으로 설치되면 장치관리자의 포트(COM & LPT) 그룹에 'USB Serial Port(COM4)'와 같은 이름으로 표시 됩니다.

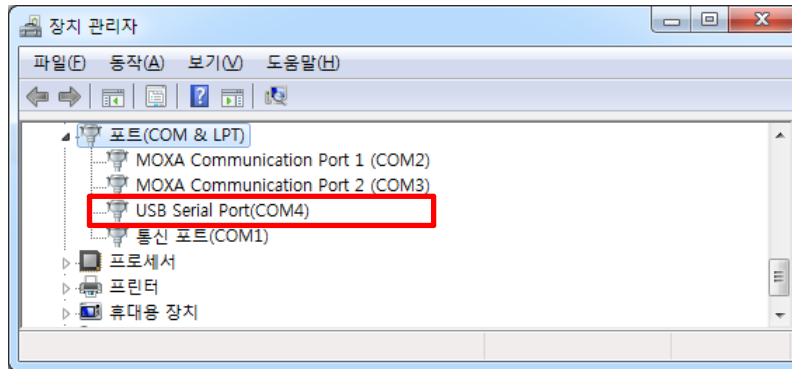


그림 3-10 장치관리자에 추가된 가상 COM 포트 드라이버

드라이버가 제대로 설치되지 않는 경우, 아래 페이지에서 USB 드라이버를 다운로드 받아 수동으로 설치해야 합니다:

- USB 드라이버 다운로드: <http://www.ftdichip.com/Products/ICs/FT232R.htm>

3.3 UI 유틸리티 다운로드와 실행

제어기가 정상적으로 시작되고 PC에 제어기의 USB 드라이버가 제대로 설치되었다면, (주)엔티렉스 연구소 홈페이지에서 최신 버전의 Motor Control UI 유틸리티를 다운로드 받습니다.

- Motor Control UI 유틸리티 다운로드: <http://www.ntrexgo.com/archives/19482>

MoonWalker Motor Control UI Program 다운로드 받기

Motor Control UI v1.02

다운로드 받은 파일은 별도의 설치 과정을 필요로 하지 않습니다. 다운로드 받은 zip 파일의 압축을 풀고 MotorControlUI.exe 파일을 실행합니다. 실행 파일은 Windows XP/7/8 32bit/64bit에서 실행 가능합니다.

3.3.1 USB(VCP) 통신 설정과 연결

제어기에서 통신과 관련된 제품 출하시 초기 설정 값은 다음과 같습니다:

- Device ID: 1
- CAN 통신 속도: 1Mbps
- Serial 포트 통신 속도: 115200bps

먼저 시리얼 포트에 연결된 제어기를 검색하기 위한 시리얼 통신 설정이 필요합니다. [Port Config.] 버튼을 눌러 그림 3-11과 같이 Port Configuration 대화상자를 실행합니다.

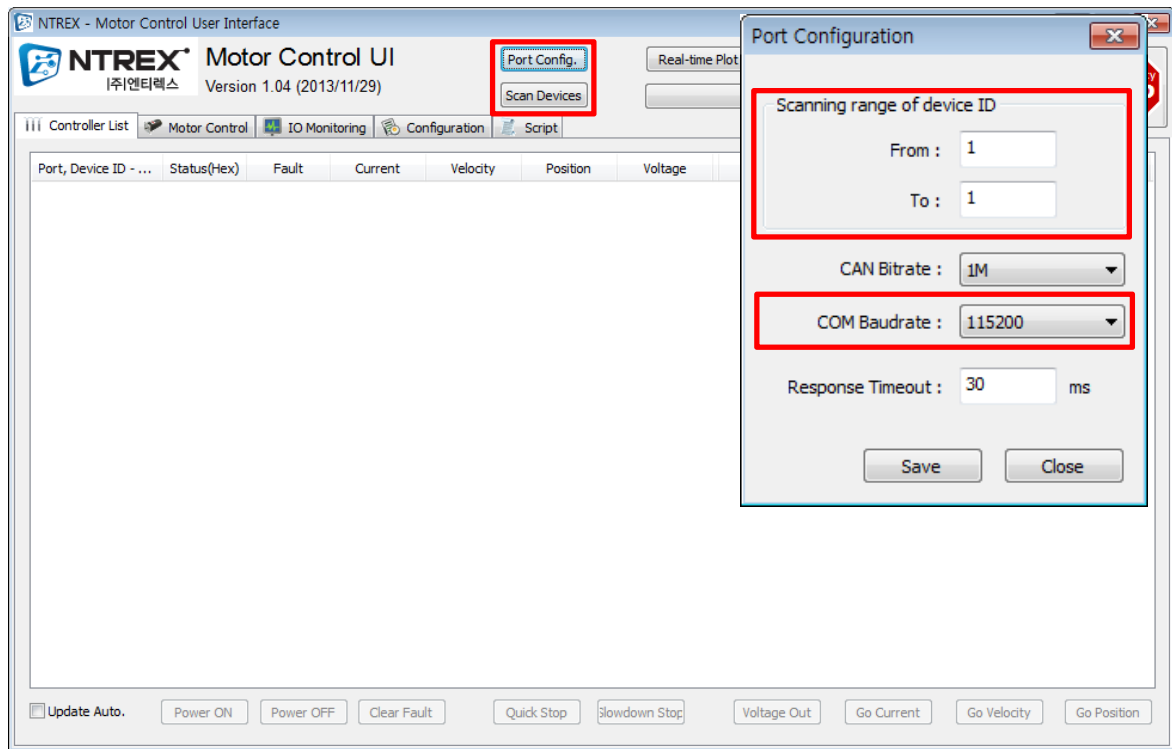


그림 3-11 통신 설정 방법

Port Configuration 대화상자에서 제어기를 검색하기 위한 장치 ID의 범위와 시리얼 통신 속도를 설정하고 [Save] 버튼을 눌러서 설정 값을 저장합니다. 그리고 [Close] 버튼을 눌러 대화상자를 닫습니다.

UI 유틸리티 헤더에서 [Scan Devices] 버튼을 눌러 USB와 RS-232, CAN 포트에 연결된 제어기들을 검색합니다. 만일 한 개 이상의 제어기가 검색되었다면, UI 유틸리티는 제일 먼저 검색된 제어기에 자동으로 연결됩니다.

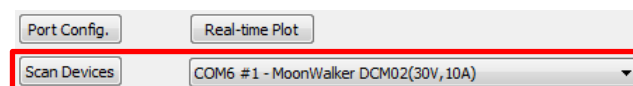


그림 3-12 연결된 제어기의 표시

PC를 사용하여 연결하는 경우 대부분 USB로 제어기에 연결하기 때문에, 그림 3-12에서와 같이 USB 포트의 가상 COM 포트(VCP; Virtual COM Port)를 통해 제어기와 통신하게 됩니다.

3.3.2 제품 버전 및 ID 확인

제어기와 통신이 되면 Configuration 탭에 Product Information과 Serial/CAN Communication 그룹에서 제품 버전과 통신에서 사용할 제어기 장치 ID 그리고 통신속도를 확인합니다. 그림 3-13에서와 같이 UI 유틸리티가 제어기와 정상적으로 통신하고 있다면, 이러한 값들이 올바르게 표시될 것입니다.

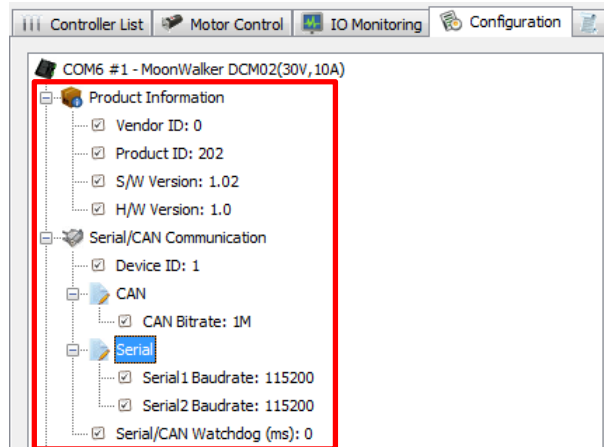


그림 3-13 제품 버전 및 ID 확인

만일 제어기의 장치 ID를 변경하거나 USB, RS-232, CAN 포트의 통신 속도를 변경하고자 할 때는, Configuration 탭의 Serial/CAN Communication 그룹에서 해당 항목을 원하는 값으로 설정하면 됩니다.

※ UI 유틸리티의 Serial/CAN Communication에서 변경한 값을 제어기에 다운로드 하고 플래시 메모리에 저장한 후, 제어기를 재시작 해야 변경된 값이 반영됩니다.

3.4 모터의 전압 구동

DC모터는 가해지는 전압에 비례하는 속도로 회전합니다. 모터의 역기전력 상수(K_e)를 알면 모터에 가해지는 전압(Voltage)에 따른 회전 속도(Rotation Speed)를 알 수 있습니다:

$$Rotation\ Speed = Voltage / K_e.$$

간단한 예로, 그림 3-14와 같이 파워서플라이에 DC모터를 연결하고 전압을 조절하면 전압에 따라 모터의 회전 속도가 비례하여 변하는 것을 확인할 수 있습니다.



그림 3-14 파워서플라이로 DC 모터의 전압 구동

제어기에서도 가장 간단히 DC 모터를 구동해 볼 수 있는 방법이 개루프(Open loop) 전압 구동입니다. 전압 구동은 모터의 위치나 속도를 측정하기 위한 추가적인 센서(엔코더, 홀센서, 포텐서

미터 등)를 필요로 하지 않고, 단지 제어기에 모터만 연결되어있다면 즉시 구동해 볼 수 있습니다. 구동에 앞서 먼저 DC 모터의 특성을 제어기의 파라미터로 설정하는 과정이 선행되어야 합니다.

3.4.1 모터의 특성 설정

Motor Control UI 유틸리티의 Configuration 탭으로 이동합니다. Motor Characteristics 그룹에는 모터의 특성을 설정하고 제어기의 각종 한계 값과 폴트 감지 조건을 설정합니다(그림 3-15 참조).

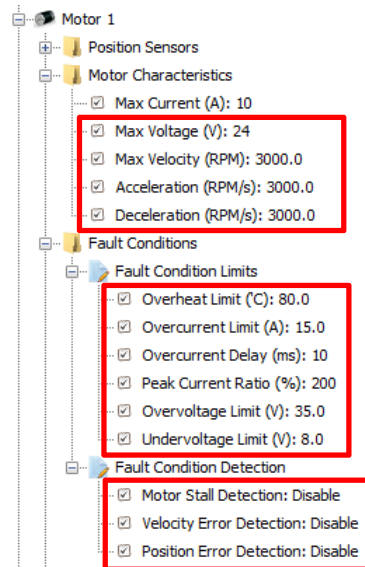


그림 3-15 모터 특성 설정

그림 3-15에서와 같이 모터를 구동하는 최대 전압(Max Voltage), 모터의 최대 속도(Max Velocity), 모터의 가속도(Acceleration)와 감속도(Deceleration) 항목의 값을 설정합니다. 자세한 설정 방법은 제어기 매뉴얼에서 "5.1.1 최대 전류와 최대 전압"을 참조하기 바랍니다.

전압 구동에서 모터의 최대 회전 속도와 모터의 회전 가감속도를 설정하는 이유는 **전압 출력 모드에서도 속도 제어 모드에서도 같이 사다리꼴 모양의 속도 프로파일을 사용**하기 위해서입니다. 자세한 설정 방법은 제어기 매뉴얼에서 "5.1.2 가속도와 감속도"를 참조하기 바랍니다.

Fault Condition Limits 그룹은 모터의 폴트를 감지하는 조건을 설정하는 파라미터들로, 모터와 제어기의 사양에 따라 과열 한계(Overheat Limit)과 과전류 한계(Overcurrent Limit), 과전류가 흐르는 시간(Overcurrent Delay), 순간 최대 전류(Peak Current Ratio), 과전압 한계(Overvoltage Limit), 저전압 한계(Undervoltage Limit) 항목의 값을 설정합니다. 자세한 설정 방법은 제어기 매뉴얼에서 "5.3 모터와 제어기 보호"를 참조하기 바랍니다.

그리고 Fault Condition Detection 그룹에서 모든 항목을 Disable로 설정해 둡니다.

모든 설정이 끝나면 Configuration 탭의 우측에 있는 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

3.4.2 모터의 전압 구동 테스트

모터의 특성 설정을 완료하였다면, UI유틸리티의 Motor Control 탭에서 마우스로 Voltage 바를 움직이면서 모터가 가해진 전압에 비례하여 회전속도가 증감하는지 확인합니다. 이 때 그림 3-16에서와 같이 Motor Control Status 그룹에서 Motor Power ON LED가 켜져 있어야 합니다.

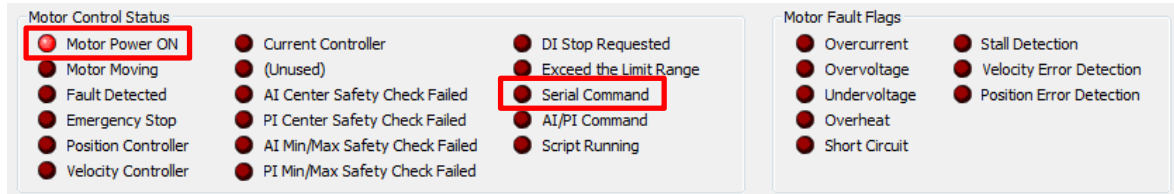


그림 3-16 모터의 제어 상태

만일 Motor Fault Flags 그룹의 LED 중 하나 이상이 켜져 있고 Motor Power ON LED가 꺼져 있는 경우에는 [Clear Fault] 버튼을 눌러 폴트 상태를 해제한 후 [Power ON] 버튼을 눌러 모터를 Power ON 상태로 만들어야 합니다. **Power ON 상태가 되지 않으면 모터는 제어가 불가능합니다.**

그리고 마우스로 Voltage 바를 움직일 때 Motor Control Status 그룹에서 Serial Command LED가 잠시 동안 켜지는지 확인합니다. 이 LED가 켜지는 것은 USB나 RS-232 포트로 명령이 수신되고 있는 것을 표시합니다.

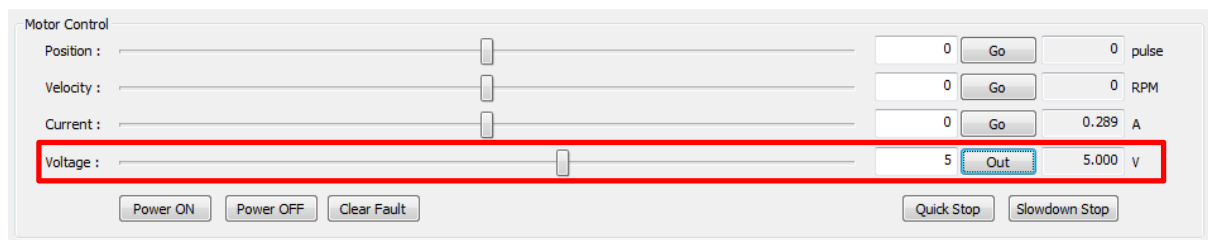


그림 3-17 전압 구동 테스트

Motor Control 탭에서 Voltage 바를 움직여 모터에 양(+)의 전압이 가해질 때 모터가 정회전 하는지 확인합니다. 반대로 음(-)의 전압이 가해질 때 모터가 역회전 하는지 확인합니다. 만일 그렇지 않다면, 그림 3-18과 같이 모터의 +극과 -극에 연결된 배선을 교차하여 반대로 연결합니다.

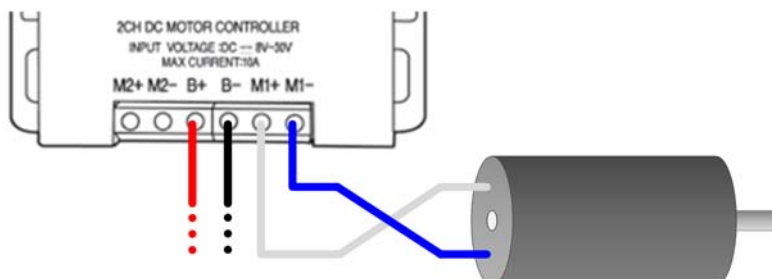


그림 3-18 모터 단자에 교차 연결된 배선

※ 모터에 가해지는 전압의 극성과 반대로 모터가 회전할 때는 모터의 +단자와 -단자의 결선을 바꾸어 줍니다. 결선을 바꾸면 모터의 회전 방향은 반대가 됩니다.

3.5 모터의 토크(전류) 구동

DC 모터에서 토크는 직접적으로 전류에 관련됩니다. 따라서 전류를 제어하는 것은 토크를 제어하는 것과 같습니다. 모터의 토크 상수(Kt)를 알면 토크(Torque)와 전류(Current) 간에는 다음과 같이 변환이 가능합니다:

$$Torque = Kt \times Current.$$

제어기 내부에는 모터에 흐르는 전류를 측정하는 전류 센서가 내장되어 있습니다. 그래서 제어기는 외부에 전류 센서를 연결하지 않고도 펄스 전류 제어가 가능합니다.

※ 모터의 전압 구동과 전류(토크) 제어는 추가적인 센서를 필요로 하지 않습니다.

3.5.1 전류 제어기 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭으로 이동합니다. 그림 3-19에서와 같이 Motor Characteristics 그룹에서 모터에 흘릴 수 있는 최대 전류를 제한하기 위해 Max Current 항목의 값을 설정합니다. 이 항목은 모터의 데이터시트에서 정격 전류 값을 참고하여 설정합니다.

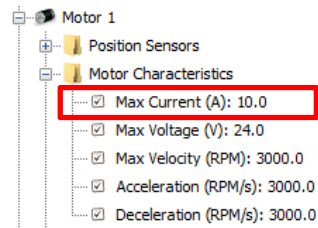


그림 3-19 최대 전류 설정

그리고 그림 3-20과 같이 Current Controller Gain 그룹에서 전류 제어기의 이득을 나타내는 Proportional Gain(Kp), Integral Gain(Ki), Feed-forward Gain(Kff) 항목의 값을 설정합니다. 이득 설정에 대한 자세한 설정 방법은 제어기 매뉴얼에서 “7.4 모터제어기 이득 조정”을 참조하기 바랍니다.

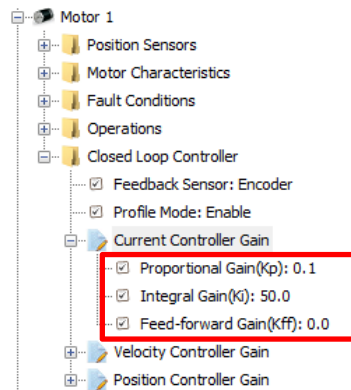


그림 3-20 전류 제어기 이득 설정

UI 유틸리티에서 전류 제어기의 이득을 변경한 후에는 [Write configurations] 버튼을 눌러 제어기

에 적용되도록 합니다. 그리고 "3.5.2 전류 제어 테스트"를 참조하여 전류 제어의 응답성을 테스트합니다. 원하는 전류제어 성능이 나올 때까지 상기 과정을 반복합니다.

모든 설정이 끝나면 [Save to Flash] 버튼을 클릭해서 변경한 내용을 제어기의 플래시 메모리에 영구히 저장합니다.

3.5.2 전류 제어 테스트

전류 제어기의 이득을 변경한 후 전류 제어기를 테스트 하려면 Motor Control 탭에서 Current 바를 움직여 봅니다. 모터에 흐르는 전류는 모터에 가해지는 토크(회전 가속도)와 비례합니다. 따라서, 모터에 약한 전류를 흘리더라도 속도가 계속 누적되어 최고 속도로 회전할 수 있습니다.

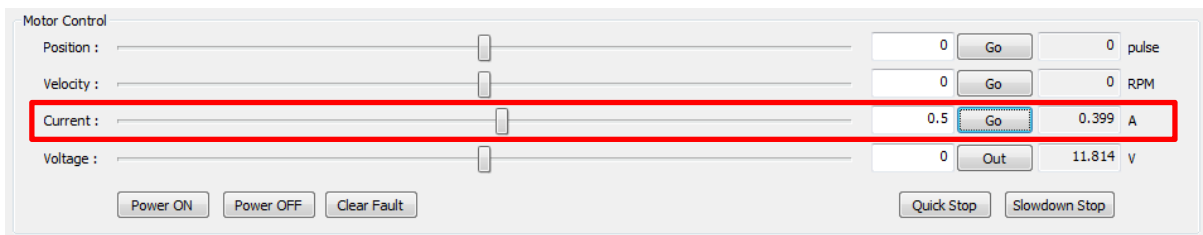


그림 3-21 전류 제어 테스트

모터에 흐르는 전류가 오버슈트와 진동 없이 사용자가 내린 전류 명령을 추종하는지 확인합니다. 전류 명령과 실제 모터에 흐르는 전류의 시간순 변화를 살펴보기 위해 UI 유틸리티 헤더의 [Real-time Plot] 버튼을 눌러 Real-time Plot 창을 표시합니다.

Real-time Plot 창이 실행되면 그림 3-22에서와 같이 선택된 오브젝트의 값을 시간에 따른 그래프로 볼 수 있습니다. 그래프는 최대 8개의 오브젝트를 표시하며, 각각에 대한 채널과 스케일을 지정 가능합니다.

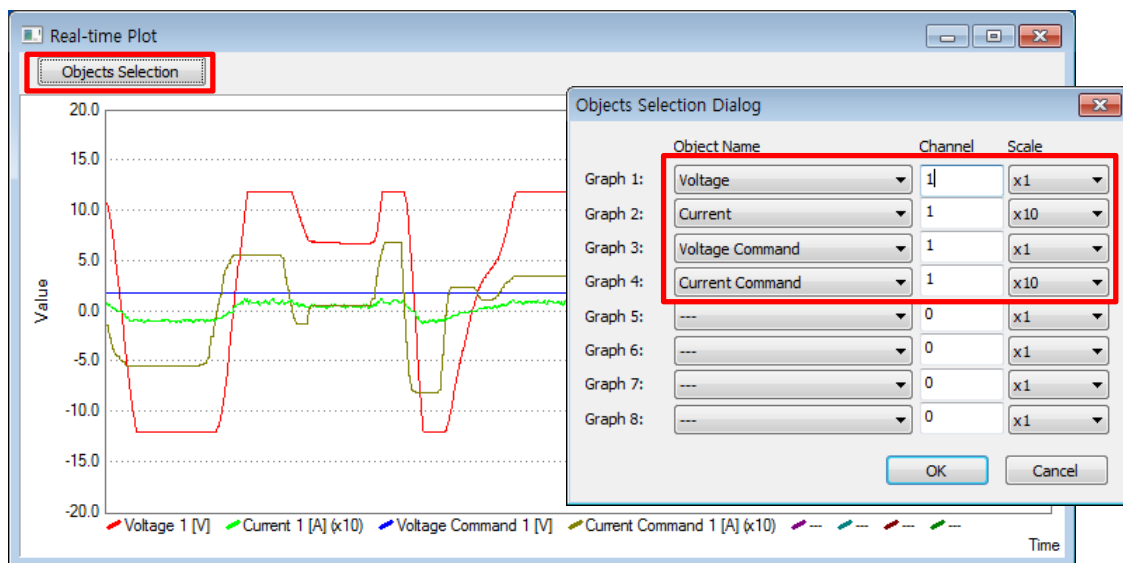


그림 3-22 Real-time Plot 창과 오브젝트 선택 대화상자

이 창에서 [Object Selection] 버튼을 눌러 Object Selection Dialog를 실행하고 모니터링 할 오브젝

트를 그림 3-22에서와 같이 선택합니다.

Real-time Plot 창에서 그래프로 표시되는 '**Current Command**' (목표 전류) 오브젝트를 '**Current**' (센서에서 측정된 전류) 오브젝트가 잘 추종하는지 확인합니다. 만일 목표 전류를 잘 추종하지 못하고 측정된 전류에서 오버슈트가 발생하거나 진동할 때는 전류 제어기의 이득을 바꿔가며 테스트를 반복합니다.

이때 주의할 점은, 전류 제어를 테스트 할 때는 **모터 회전축을 회전하지 못하도록 고정**하고 테스트 해야 합니다. 아래 그림에서 왼쪽 그래프는 모터의 회전축을 고정하지 않았을 때로, 목표 전류 (황색 그래프)를 현재 전류(녹색 그래프)가 잘 추종하지 못하고 있습니다. 이는 모터의 회전에 의해 발생하는 역기전력이 전류를 흐르지 못하도록 방해하기 때문입니다. 오른쪽 그래프는 모터의 회전축을 고정하여(손으로 잡고 있는 상황) 회전하지 못하도록 한 상태에서 테스트한 그래프입니다.

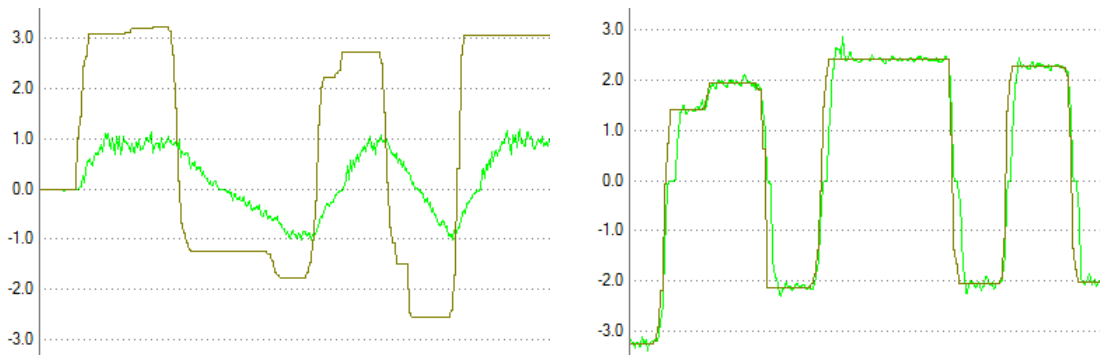


그림 3-23 전류제어기에서 전류 추종 그래프

모터의 인덕턴스 값이 작고 모터의 전류 제어 특성이 나쁠 때는 다음과 같이 제어기와 모터를 연결하는 배선에 직렬로 인덕턴스가 큰 코일을 삽입하여 모터의 전류 응답성을 느리게 해 봅니다.

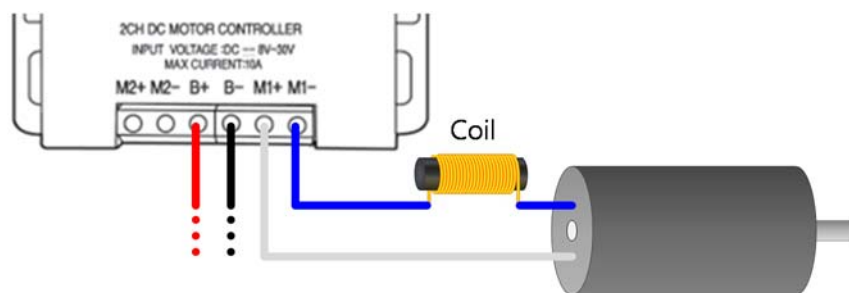


그림 3-24 모터 연결에 코일 사용

그림 3-24에서와 같이 모터와 직렬로 코일을 사용하는 것은 Bipolar PWM Switching 방식에서 인덕턴스가 낮은 모터를 사용할 때도 유용합니다. 모터의 인덕턴스가 낮은 경우 Idle current가 높아 (모터는 돌지 않는데 전류는 계속해서 흐름) 에너지를 낭비하고 모터에 열을 발생시킵니다. 이러한 경우, 코일을 사용하여 Idle current를 낮출 수 있습니다.

4 엔코더 연결과 위치·속도 구동

DC 모터로 정밀한 위치나 속도 제어를 하려면 엔코더는 필수적으로 사용되어야 합니다. 보통 엔코더 없이 모터에 가해지는 전압만으로도 어느 정도 속도를 계산할 수 있지만, 부하가 바뀔에 따라 속도가 일정하지 않아 계산 속도와 실제 속도 간 오차가 발생합니다. 따라서 **정밀한 위치/속도 제어를 위해서는 엔코더를 사용한 피드백 제어가 필수** 요구됩니다.

이번 장에서는 엔코더가 결합된 모터를 사용하여 폐루프(Closed-loop) 위치 제어와 속도 제어에 대해 설명합니다.

이번 장에서 필요한 준비물은 다음과 같습니다:

- 3장에서의 준비물
- 광학식 증분 엔코더
- 엔코더 연결 커넥터와 핀
- 제어기 엔코더 단자 연결 커넥터
- 약간의 엔코더 신호선 (4 연선)
- 압착기

※ 엔코더를 사용한 폐루프 위치·속도 제어에 앞서, "3 모터 연결과 전압·전류 구동"에서 다룬 내용이 제어기에 기본적으로 구성되어 있어야 합니다.

4.1 증분 엔코더의 사용

증분 엔코더(Incremental Encoder)는 모터의 회전각 변화량을 전기적인 신호(pulse)로 변환하여 출력하는 센서입니다. 위치나 각도의 증가량과 방향을 읽어 모터의 회전 위치와 회전 속도를 측정합니다.

※ 제어기의 엔코더 단자에 연결 가능한 엔코더는 A와 B상 신호를 출력하는 증분 엔코더입니다. 제어기에 절대 엔코더를 연결하는 방법은 "6 아날로그·펄스 입력 포트의 사용"에서 다룹니다.

4.1.1 광학식 증분 엔코더의 원리

광학식 증분 엔코더(Incremental Optical Encoder)는 구조가 간단하고 가격이 싸며 출력선의 수도 적어 모터의 폐루프 위치·속도 제어에 널리 이용되고 있습니다. 아래 그림 4-1은 광학식 증분 엔코더의 구조를 보여줍니다.

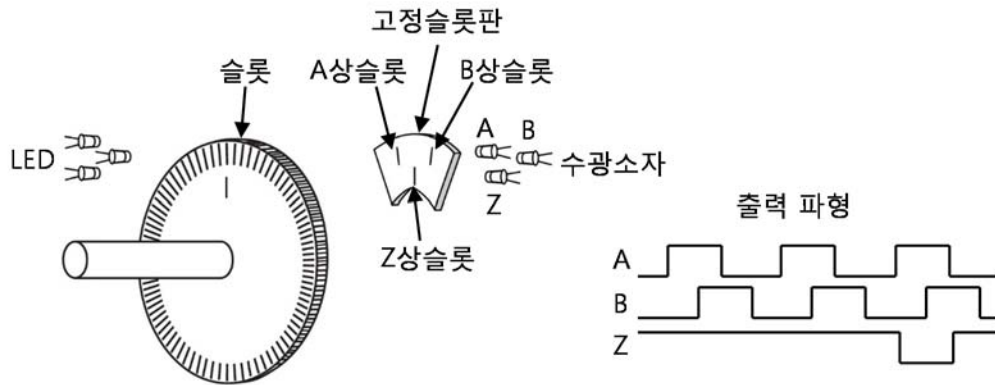


그림 4-1 광학식 증분 엔코더의 구조

LED로부터 투사된 광선은 회전 디스크의 슬릿을 통과한 뒤 고정 슬릿판의 A, B, Z에 해당하는 각각의 슬릿을 통과하여 A, B, Z의 수광소자에서 검출 됩니다. 고정 슬릿판의 A, B 슬릿은 90도의 위상차를 가지도록 배치되어 있으며, 전기적 신호의 출력도 90도의 위상차를 가지는 구형파가 됩니다. Z 신호는 회전 디스크가 한 바퀴 회전할 때 한 번 검출됩니다.

엔코더의 출력 펄스는 절대 위치를 나타내지 않고 회전한 각도에 비례하는 펄스수가 얻어지기 때문에 펄스 카운터에 의해 위치를 알게 됩니다. 만일 전원이 끊어지면 카운터의 위치를 잃어버리기 때문에 사용에 주의해야 합니다.

4.1.2 모터에 엔코더 결합

모터와 엔코더를 따로 구하여 결합하는 것보다, 모터에 엔코더가 결합된 제품을 구매하는 것이 편리합니다. 일부 전자부품이나 로봇관련 온라인 쇼핑몰에서 해상도가 낮은 엔코더가 장착된 모터들이 판매되고 있습니다. 만일 로봇의 바퀴 구동부에서와 같이 감속기+모터+엔코더 일체형이 필요하다면 이러한 모터를 사용하여도 위치나 속도 제어에 문제가 되지 않습니다.

만일 엔코더의 해상도가 높고 모터와 감속기의 선택 폭이 넓어야 한다면 전문 모터 메이커에서 구매해야 합니다:

사용자의 목적에 부합되는 사양의 모터와 엔코더 일체형 제품이 없을 때는 부득이하게 모터와 엔코더를 따로 구매하여 결합하여야 합니다. 이때 모터는 엔코더를 연결할 수 있도록 그림 4-2에서와 같이 회전축이 모터 뒤로 돌출되어 있어야 합니다. 그리고 엔코더는 뒤로 돌출된 회전축의 지름을 고려하여 구매하여야 합니다.



그림 4-2 엔코더 연결이 가능한 모터

그림 4-3에서 오른쪽과 같이 모터에 증분 엔코더를 연결할 때 **감속기(Gear Box)를 거쳐 연결하는 구조는 되도록 사용하지 않는 것이 좋습니다.** 이렇게 사용하면 엔코더의 해상도가 감속비 만큼 낮아지고 기어의 백래시(backlash)로 인해 모터의 속도제어가 어렵습니다. 하지만 절대 엔코더와 같은 경우에는 주로 감속기를 거쳐 최 종단에 연결되는 경우가 많습니다.

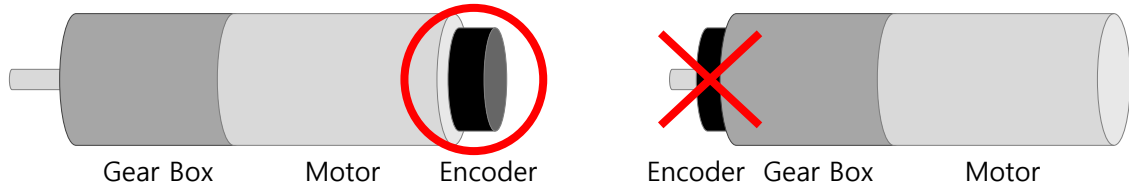


그림 4-3 모터와 엔코더의 연결 방법 비교

4.1.3 엔코더 신호선 연결

이 장에서 모터의 위치와 속도 구동에 사용한 엔코더는 US Digital사의 E8P OEM Optical Kit Encoder 입니다. 이 엔코더는 단일 5V 전원을 사용하고 A와 B상을 출력합니다.

제어기와 엔코더를 연결할 때는 그림 4-4에서와 같이 4가닥의 연선을 사용하여 가능한 짧게 배선 합니다. 그리고 배선이 전원 선이나 모터 선과 겹치지 않도록 분리하고, 분리가 불가능할 때는 쉴드 선을 사용합니다.

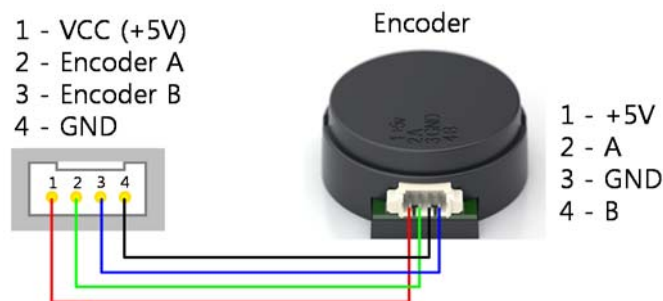


그림 4-4 제어기와 엔코더 배선

또한, 제어기 엔코더 단자의 핀 맵과 엔코더의 핀 맵을 확인 후 정확히 연결하여야 합니다. 제어기 모델에 따라 엔코더를 연결하는 단자가 서로 다릅니다.

엔코더와 제어기간 배선 길이가 길어질 때는 그림 4-5에서와 같이 라인 드라이버를 사용합니다. 라인 드라이버는 엔코더 신호를 증폭하여 노이즈를 억제하는 효과가 있습니다.

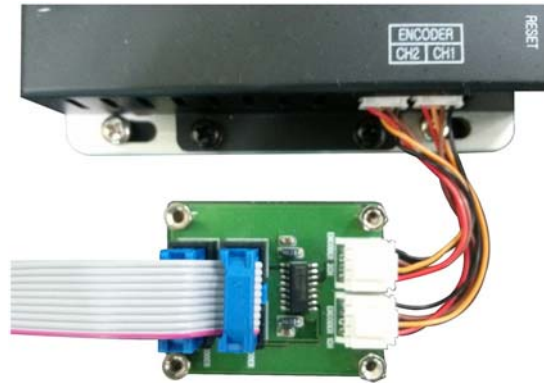


그림 4-5 라인드라이버를 사용한 엔코더 배선

4.1.4 엔코더 방향 확인

제어기와 모터, 엔코더, 전원을 모두 연결하고 제어기에 전원을 투입합니다. 그리고 PC에서 제어기로 USB 연결하고 Motor Control UI 유틸리티를 실행합니다. 헤더에서 [Scan Devices] 버튼을 눌러 연결된 제어기를 검색하고 첫 번째로 검색된 제어기와 통신을 시작합니다.

Motor Control 탭에서 Voltage 바를 오른 쪽으로 움직여 모터에 약간의 양(+) 전압을 가해 봅니다. 그러면 모터는 정방향으로 회전하고 엔코더 카운트 값도 증가하게 됩니다. 만일 엔코더 카운트 값이 감소한다면 그림 4-6과 같이 엔코더의 A와 B상 연결을 서로 바꿔 줍니다. 그러면 엔코더의 카운트 방향이 반대로 바뀌게 됩니다.

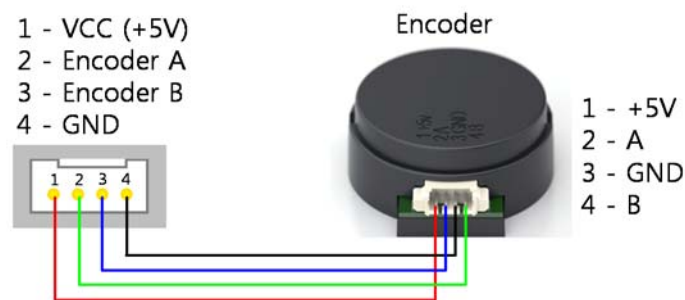


그림 4-6 엔코더의 A와 B상을 바꿔 배선

또 다른 방법으로 모터의 +와 -단자 연결을 바꾸는 방법도 있습니다. 그러면 모터의 회전 방향이 바뀌면서 엔코더 카운트 방향과 일치하게 됩니다.

※ 만일 양의 속도 명령에 대해 모터가 반대로 회전하는(음의 모터 속도가 검출되는) 경우는 모터의 +, -극 연결을 바꾸거나 엔코더의 A, B 상의 연결을 바꿔줍니다.

4.1.5 엔코더 노이즈 대책

Motor Control 탭에서 Voltage 바를 움직여 모터에 약간의 전압을 가한 상태를 유지합니다. 그러면 모터는 일정한 속도로 회전하는데, 이때 Real-time Plot 창을 실행하여 제어기의 '**Velocity**' 오

브렉트를 모니터링 합니다. 만일 이 때 그림 4-7과 같이 제어기의 속도 그래프(녹색)가 일정하지 않고 튀는 현상이 나타날 수 있습니다. 이는 주로 엔코더 및 배선에 노이즈가 유입되어 나타나는 현상입니다.

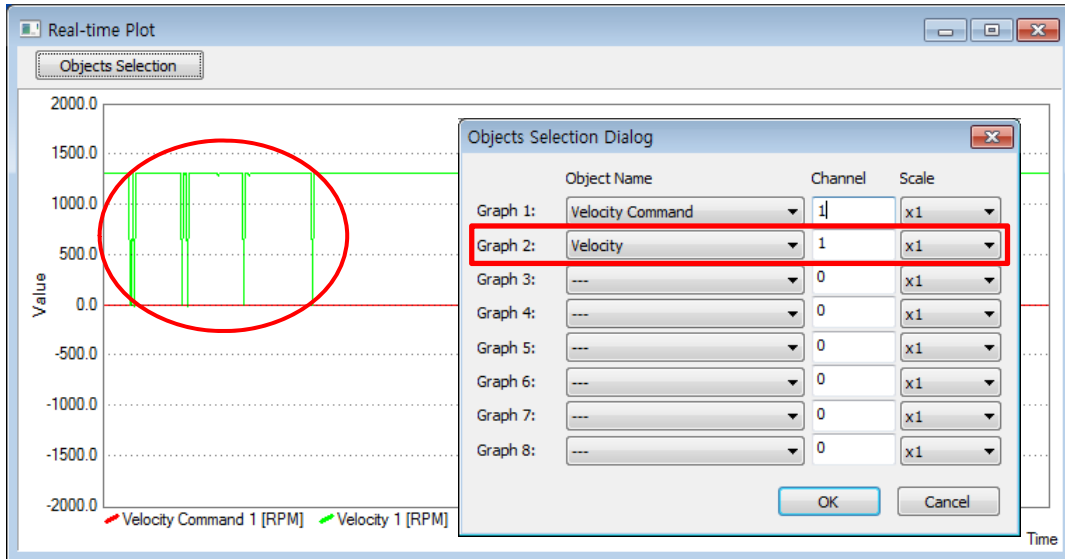


그림 4-7 엔코더 신호에 노이즈가 유입되었을 때의 속도 그래프

이런 경우, 엔코더와 제어기간 배선 길이를 최단거리로 하고 모터나 전력 선과 엔코더 선이 겹치지 않도록 분리 배선 합니다. 그리고 그림 4-8에서와 같이 페라이트코어를 엔코더 선에 감아 어느 정도 노이즈를 감소하는 효과를 볼 수 있습니다.



그림 4-8 엔코더 신호선에 페라이트코어 감기

4.2 모터의 속도 구동

제어기에 연결된 모터와 엔코더가 정상적으로 동작하는 것을 확인하였다면, PC에서 Motor Control UI 유틸리티를 사용하여 모터의 속도제어 테스트를 진행합니다.

이 절에서는 모터의 속도제어와 관련된 파라미터를 설정하고 속도 제어기의 이득을 설정한 후 모터를 속도제어 모드로 구동해 보는 것에 대해 설명합니다.

4.2.1 속도 제어기 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 모터의 속도 구동을 위한 파라미터를 설정합니다.

그림 4-9에서 엔코더의 분해능(Encoder Resolution)과 사다리꼴 속도 프로파일 사용을 위한 최대 회전 속도(Max Velocity), 모터의 운용 환경과 모터에 연결된 부하를 고려하여 모터의 회전 가속도(Acceleration)와 감속도(Deceleration)항목의 값을 설정합니다.

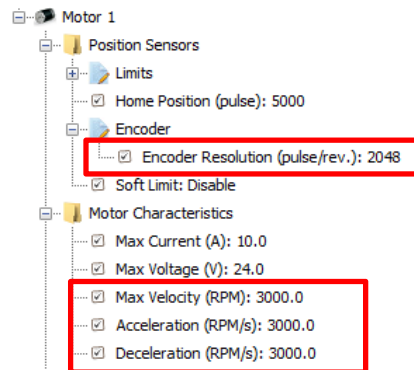


그림 4-9 엔코더 분해능과 모터 특성 파라미터 설정

엔코더 분해능과 모터의 최대 회전 속도는 데이터시트를 참조합니다. 정확한 엔코더 분해능을 모를 때는 모터를 Power OFF 한 상태에서 손으로 한 바퀴 돌려서 카운트 되는 펄스 수를 측정하면 됩니다.

※주의※ 엔코더 분해능은 엔코더 펄스 수를 4배한 값으로 설정해야 합니다. 만약 올바른 값을 설정하지 않는다면, 제어기는 모터의 회전 속도를 제대로 계산하지 못하게 됩니다.

Closed Loop Controller 그룹에서 Profile Mode 항목을 Enable로 설정합니다. 그리고 속도 제어기의 Feedback Sensor 항목을 Encoder로 선택합니다. 속도 프로파일 설정에 대한 자세한 내용은 사용자 매뉴얼의 "7.4.2 속도 제어기 PI 이득 조정"을 참고합니다.

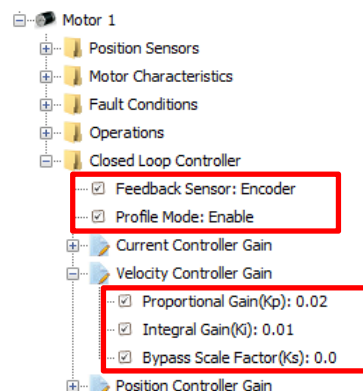


그림 4-10 속도 제어기 이득 설정

Velocity Controller Gain 그룹에서 속도 제어기의 Proportional gain(Kp)와 Integral Gain(Ki)를 설정

합니다. Bypass Scale Factor(Ks)는 0으로 둡니다. 속도 제어기의 이득 설정에 대한 자세한 내용은 사용자 매뉴얼의 "7.4.2 속도 제어기 PI 이득 조정"을 참고합니다.

모든 설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

4.2.2 속도 제어 테스트

Motor Control UI 유틸리티의 헤더에서 [Real-time Plot] 버튼을 눌러 Real-time Plot 창을 실행합니다. 그리고 모니터링 오브젝트로 '**Velocity**'와 '**Velocity Command**'를 선택합니다. 그러면 그림 4-11과 같이 속도 명령과 모터의 실제 속도를 시간에 따라 그래프로 모니터링 할 수 있습니다.

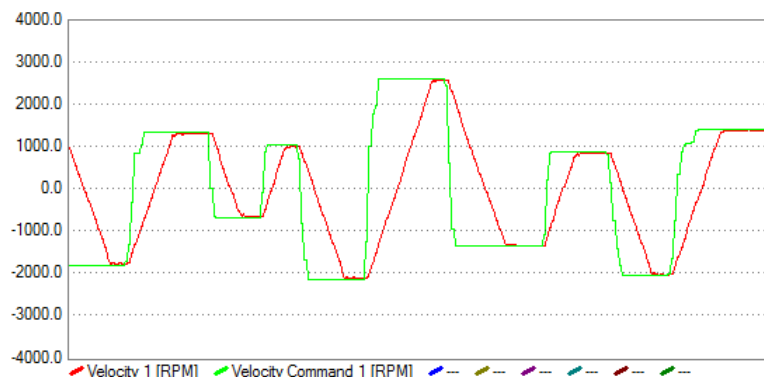


그림 4-11 속도 명령과 모터의 실제 속도 그래프

그리고 그림 4-12에서와 같이 Motor Control 탭에서 Velocity 바를 움직이면서 모터의 속도가 속도 명령을 따라가는지 그림 4-11의 그래프를 통해 확인합니다.



그림 4-12 속도 제어 테스트

※주의※ 만일 모터의 회전 방향과 엔코더의 카운트 방향이 불일치 할 경우, 모터에 아주 작은 속도 명령이 내려지더라도 페루프 속도 제어기가 오차를 증폭시켜 모터는 최고 속도로 회전하게 되고 제어가 불가능한 위험한 상황을 초래할 수 있습니다.

이 때는 Motor Control UI 유틸리티의 우측 상단 **[Emergency STOP]** 버튼을 눌러 모터를 정지하거나 Motor Control 탭의 [Power OFF] 버튼, 혹은 [Quick Stop] 버튼을 눌러 모터를 정지할 수 있습니다. [Declaration Stop] 버튼은 모터를 정지할 수 없습니다.

또한, Configuration 탭에서 Profile Mode 항목을 Enable 혹은 Disable로 바꾸어서 속도 제어를 테스트 해 봅니다. 그림 4-11은 프로파일 모드를 켜고 테스트 한 그래프 입니다.

4.3 모터의 위치 구동

모터의 속도 구동이 잘 된다면, 모터의 위치 구동은 어려움 없이 수행할 수 있습니다.

이 절에서는 모터를 위치 제어하기 위한 관련 파라미터들을 설정하고 위치 제어기의 이득을 설정한 후 모터를 위치제어 모드로 구동해 보는 것에 대해 설명합니다.

4.3.1 위치 제어기 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭으로 이동합니다. 그리고 모터의 위치 구동과 관련된 파라미터들을 설정합니다.

먼저 Limits 그룹에서 Min Position과 Max Position 항목에 모터가 이동 가능한 최대/최소 범위를 설정합니다. 그리고 Soft Limit 항목('Use Soft Limit' 오브젝트)를 Enable 혹은 Disable로 설정합니다. 이 항목을 Enable로 설정하면 설정한 최대/최소 범위를 벗어날 경우 Exceed the Limit Range 상태를 표시하고 벗어난 방향으로의 모터 구동은 불가능하게 됩니다. 소프트 리미트에 대한 자세한 내용은 사용자 매뉴얼의 "11.4.5 use_soft_limit - Use Soft Limit"를 참조하기 바랍니다.

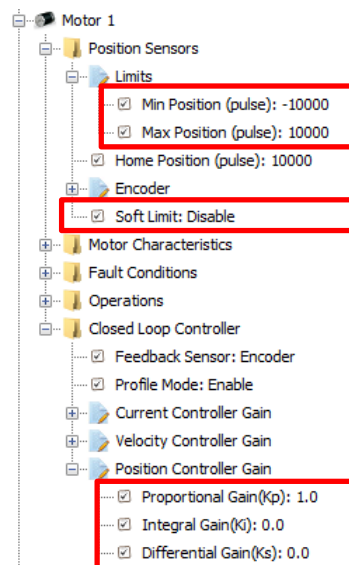


그림 4-13 위치 센서 파라미터 및 위치 제어기 이득 설정

그리고 Position Controller Gain 그룹에서 위치 제어기의 이득으로 Proportional Gain(Kp), Integral Gain(Ki), 'Differential Gain(Kd) 항목의 값을 설정합니다. 위치 제어기의 이득 설정에 대한 자세한 내용은 사용자 매뉴얼의 "7.4.3 위치 제어기 PID 이득 조정"을 참조하기 바랍니다.

모든 설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

4.3.2 위치 제어 테스트

Motor Control UI 유틸리티의 헤더에서 [Real-time Plot] 버튼을 눌러 Real-time Plot 창을 실행합니다. 그리고 모니터링 오브젝트로 '**Velocity**'와 '**Position**', '**Position Command**'를 선택합니다. 그러면 그림 4-14와 같이 각각의 오브젝트를 시간변화에 따른 그래프로 모니터링 할 수 있습니다.

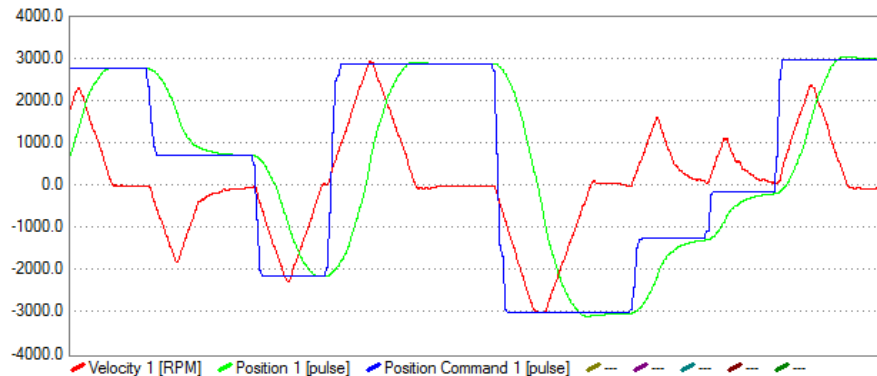


그림 4-14 위치 명령과 모터의 실제 위치, 속도 그래프

그리고 그림 4-15에서와 같이 Motor Control 탭에서 Position 바를 움직이면서 모터의 위치가 위치 명령을 잘 추종하는지 그림 4-14의 그래프를 통해 확인합니다.



그림 4-15 위치 제어 테스트

그림 4-10에서와 같이 Profile Mode 항목이 Enable로 설정된 경우, 제어기는 목적 위치로 이동하기 위해 사다리꼴 모양의 속도 프로파일을 사용합니다. 그림 4-14의 속도 그래프(청색 그래프)는 속도 프로파일이 적용된 그래프입니다.

위치 제어에서 소프트 리미트 기능을 사용하기 위해 그림 4-13에서 Soft Limit 항목('Use Soft Limit' 오브젝트)을 Enable로 설정합니다. 그리고 Velocity나 Current, Voltage 바 중 하나를 움직여 리미트 범위를 벗어나도록 모터를 구동해 봅니다. 만일 모터의 위치가 Min Position을 벗어나면 음(-)의 방향으로 움직이는 명령은 더 이상 실행되지 않습니다. 이 때 양(+)의 방향으로 움직이는 명령은 실행 됩니다. 반대로 Max Position을 벗어나면 양(+)의 방향으로 움직이는 명령이 실행되지 않고 양(+)의 방향으로 움직이는 명령만 실행됩니다.

5 이동로봇에 제어기 적용

이전 장에서 제어기에 모터를 연결하고 전압, 전류, 속도, 위치 제어를 수행하였습니다. 이번 장에서는 이동로봇의 구동부에 듀얼 채널 제어기를 적용하는 방법에 대해 설명합니다.

이번 장에서 필요한 준비물은 다음과 같습니다:

- 3장에서의 준비물
- 듀얼 채널 MoonWalker 제어기
- 2륜 로봇 플랫폼(엔코더와 감속기가 연결된 두 개의 모터로 구성)
- 4륜 로봇 플랫폼(엔코더와 감속기가 연결된 네 개의 모터로 구성)
- 포텐서미터가 내장된 2채널 아날로그 조이스틱
- 2채널 이상의 RC 조종기와 RC 수신기
- 압착기, 스트리퍼, 인두 등 배선 자재와 공구

5.1 2륜 차동 구동 로봇

2륜 차동 구동(Two wheel differential drive) 로봇은 두 개의 독립적으로 구동되는 바퀴와 하나 또는 두 개의 자유롭게 움직이는 (볼)캐스터를 가지는 구조입니다. 이러한 구조는 설계와 제작이 용이하고 제어 방법도 단순하기 때문에 실내용 이동 로봇 플랫폼의 제작에 가장 많이 응용되는 구조입니다.

이 절에서는 2륜 차동 구동 로봇에 대해 간략히 소개하고 듀얼 채널 MoonWalker 제어기를 사용하여 로봇을 제작하고 주행해 보는 것에 대해 설명합니다.

5.1.1 로봇 기구부

2륜 차동 구동 로봇은 기구적으로 가장 단순한 형태의 이동 로봇으로, 그림 5-1과 같이 좌우 바퀴에 각각 구동을 위한 모터가 연결되고 앞이나 뒤에서 (볼)캐스터가 로봇을 지지하는 구조입니다.

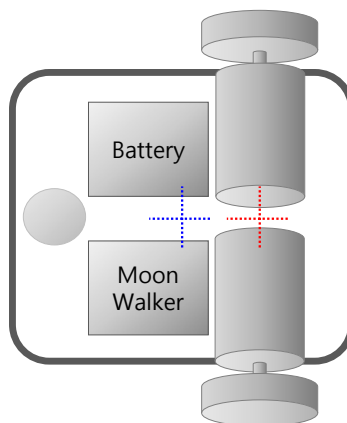


그림 5-1 두 구동 바퀴와 하나의 (볼)캐스터를 가지는 로봇

이러한 3점 지지 구조는 로봇의 중심(상기 그림에서 청색 점선 십자가)이 회전 중심(상기 그림에서 적색 점선 십자가)과 일치하지 않아 회전반경이 커지는 문제가 있습니다. 하지만 기구부 제작이 간단하기 때문에 이동로봇 베이스로 가장 많이 사용되는 구조입니다.

회전 중심이 로봇의 중심과 일치하지 않는 문제를 해결하기 위해 그림 5-2와 같이 로봇의 앞뒤로 (볼)캐스트가 있고 중앙에 구동 바퀴가 배치되는 구조를 사용하기도 합니다.

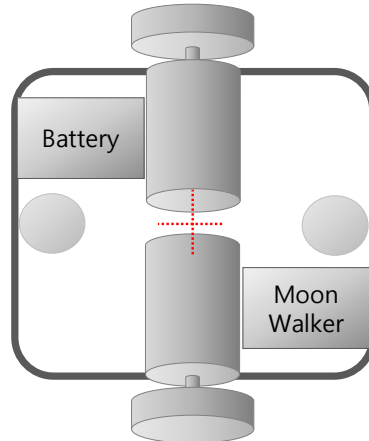


그림 5-2 바퀴가 중앙에 있는 2륜 차동 구동 로봇 플랫폼 구조

하지만 이러한 4점 지지 구조의 로봇은 바닥이 조금이라도 평평하지 않으면 네 바퀴 중 하나가 지면과 접촉하지 않게 되어 구동력을 잃어버릴 수 있습니다. 그래서 바퀴 축 그리고/혹은 (볼)캐스터에 서스펜션을 달기도 하지만, 이렇게 설계한 경우 운행도중 로봇의 몸체가 기울어지거나 흔들리게 됩니다.

5.1.2 로봇 배선

그림 5-3과 같이 제어기와 배터리, 모터, 엔코더 간에 배선합니다. 이 때, 전력선들이 엔코더 신호선과 동일한 경로를 따라 배선되지 않도록 주의합니다. 그리고 엔코더 선은 최대한 짧게 배선합니다.

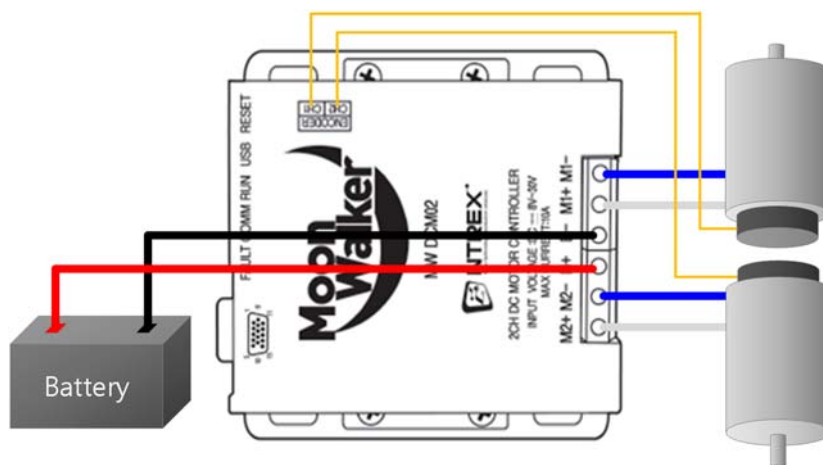


그림 5-3 2륜 차동 구동 로봇의 배선도

여기서 제공하는 배선도는 2륜 차동 구동 로봇의 가장 기본적인 내용만을 포함합니다. 로봇의 안전한 운행과 편의를 위해 전원 스위치, 긴급 정지 버튼, 퓨즈, 배터리 충전회로 등 부가적인 장치 구성이 필요합니다. 사용자는 로봇의 사용 목적 및 운행 환경에 맞추어 적절한 회로를 구성하기 바랍니다.

※주의※ 로봇에 장착된 제어기와 다른 장치들이 전원 라인을 공유하고 있다면, 이를 분리하여야 합니다. 만일 하나의 배터리에 모터 제어기와 센서가 연결되어 있는 경우 제어기에서 발생한 과전압으로 인해 센서가 파괴될 수도 있습니다.

이런 현상은 정지한 로봇을 외력으로 밀어서 움직이거나 로봇을 급 감속 할 때 발생합니다. 또한 전원 스위치가 꺼져 있더라도 모터의 발전으로 인해 센서 등의 기타 장비에 과전압이 걸릴 수 있습니다. 이러한 문제의 가장 좋은 해결책은 모터 제어기가 다른 장치들과 전원이 분리되도록 전용 배터리를 사용하는 것입니다. 만일 이렇게 사용하는 것이 불가능할 때는 **공유되는 배터리와 다른 장치들 간에 과전압 보호 장치를 사용**해야 합니다.

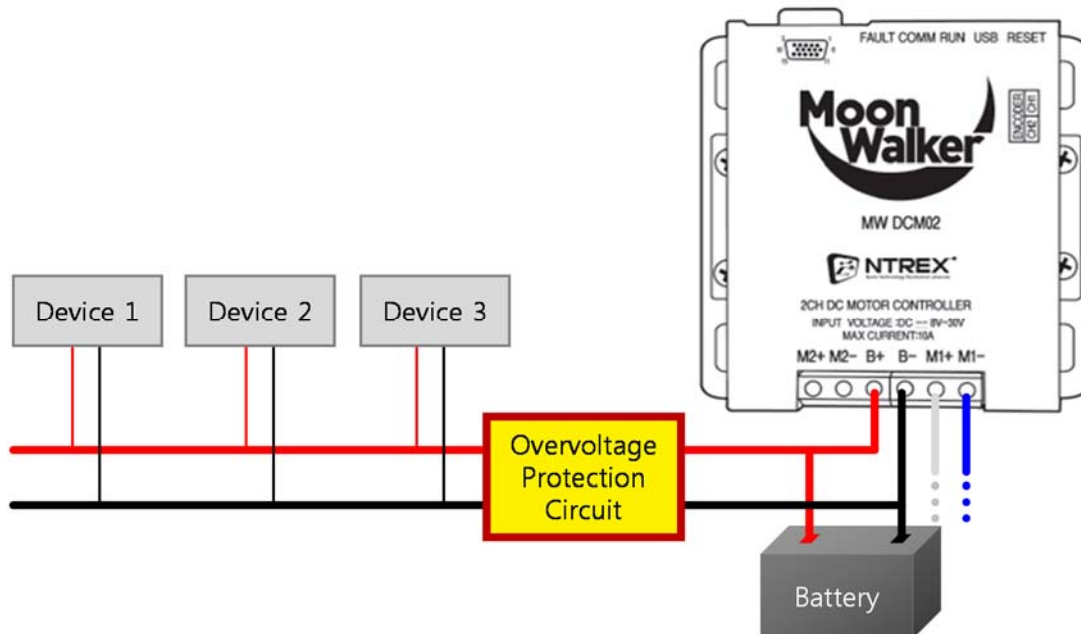


그림 5-4 과전압 보호 회로의 사용

제어기와 배터리, 모터, 엔코더 등 모두 연결되면 제어기에 전원을 인가합니다. 그리고 PC와 제어기간에 USB 포트로 연결 합니다. 제어기에 배터리, 모터, 엔코더 등을 연결하기 위해 "3 모터 연결과 전압·전류 구동"과 "4 엔코더 연결과 위치·속도 구동"을 참고하기 바랍니다.

5.1.3 제어기 파라미터 설정

모터와 엔코더의 기본적인 설정은 "4 엔코더 연결과 위치·속도 구동"과 "5 이동로봇에 제어기 적용"의 설명을 따릅니다. 여기서는 2륜 구동 로봇과 관련된 파라미터 설정에 대해서만 다룹니다.

Motor Control UI 유틸리티에서 Configuration 탭으로 이동합니다. 그리고 Joystick/RC Control and

Safety 그룹에서 Control Mixing 항목을 Mixing Mode 1/2/3/4 중 하나로 설정합니다. 믹싱 모드에 대한 자세한 내용은 사용자 매뉴얼의 "9.6.3 신호의 믹싱"을 참조하기 바랍니다.

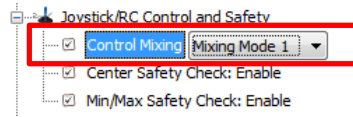


그림 5-5 믹싱 모드 파라미터 설정

로봇의 좌우 바퀴의 반지름(Wheel Radius), 좌우 바퀴간 거리(Axle Length), 바퀴의 감속비(Gear Ratio) 항목의 값을 설정합니다. 로봇의 기구적 특성 파라미터에 대한 자세한 내용은 사용자 매뉴얼의 "10.8 모바일 로봇 속성"을 참조하기 바랍니다.

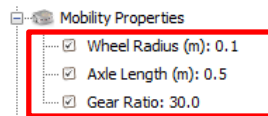


그림 5-6 이동로봇 기구적 특성 파라미터 설정

로봇에 장착된 모터의 특성을 반영하여 Motor 1그룹의 각종 파라미터를 변경합니다. Motor 1에 대한 설정이 끝나면, 그림 5-7에서와 같이 [Copy Config. Mot.1->Mot.2] 버튼을 클릭해서 Motor 1에서 설정한 값들을 Motor 2로 복사합니다. 이제 좌우 모터에 대한 설정 값들이 서로 일치하게 됩니다. 보통 이동로봇의 좌우 바퀴에는 같은 사양의 모터를 사용하기 때문에 파라미터들도 서로 동일하게 설정합니다.

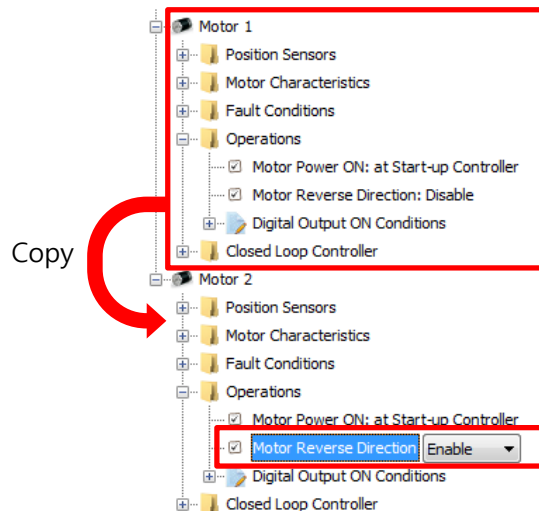


그림 5-7 2륜 구동 로봇 파라미터 설정

마지막으로 **Motor 1에서 Motor Reverse Direction 항목을 Disable로 설정하였다면, Motor 2에서는 Enable로 설정합니다.** Motor 1과 Motor 2에서 이 둘을 반대로 설정해야 로봇 바퀴의 구동 방향을 서로 일치시킬 수 있습니다.

그리고 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

5.1.4 제어기 이득 동조

전류, 속도, 위치 제어기의 이득을 동조할 때는 기본적으로 실험을 반복하면서 수작업으로 최적의 이득을 찾게 됩니다. 이 때 로봇을 지면으로부터 들어올려 바퀴가 자유롭게 회전할 수 있는 상태에서 이득을 동조하는 것이 편리합니다. 하지만 이렇게 동조한 이득은 로봇의 부하, 관성, 기어 비율 등을 반영하지 않기 때문에 실제 로봇을 운행할 때는 원하는 성능을 만족하지 않을 수도 있습니다.

보통 이동로봇은 사용자가 내리는 속도 명령에 따라 움직입니다. 따라서, 모터제어기에서 전류 제어기와 속도 제어기의 이득만 설정합니다.

5.1.5 로봇의 주행 테스트

이동 로봇에 대한 설정이 완료되었다면 그림 5-8에서와 같이 Motor Control 탭의 Motor Selection 그룹에서 Channel 1과 Channel 2 중 하나를 선택하여 좌우 모터를 바꿔가며 속도 제어기를 테스트 해 봅니다. 이 탭에서는 하나의 모터에 대해서만 테스트 가능하기 때문에 두 개의 모터를 동시에 테스트 하기에는 불편한 점이 있습니다.

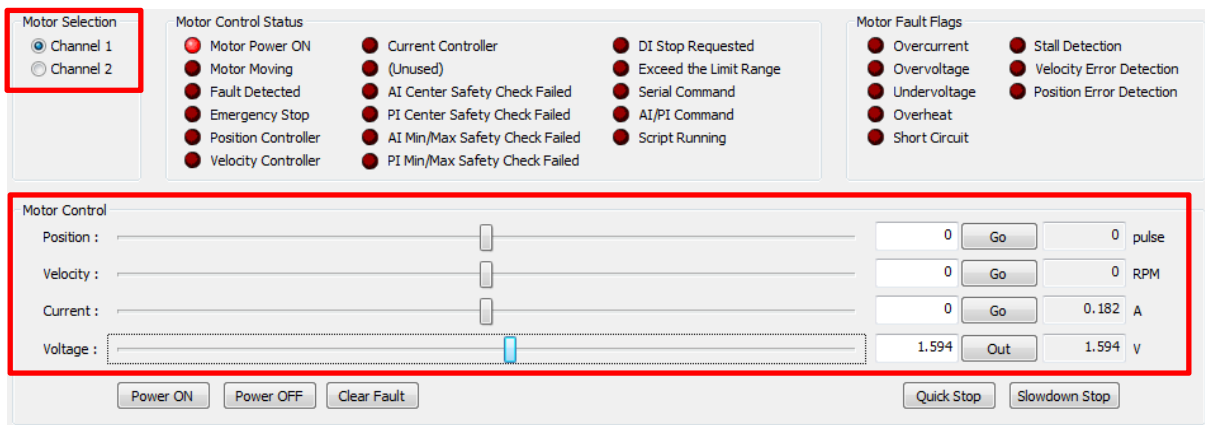


그림 5-8 모터 선택 및 개별 구동

그림 5-9와 같이 Controller List 탭에서는 두 모터에 동시에 속도 명령을 내릴 수 있습니다. 좌측 하단의 Update Auto 체크박스를 체크하고 모터의 1채널과 2채널을 모두 체크합니다. 그리고 Voltage나 Velocity에 원하는 전압이나 속도 값을 적고 [Voltage Out] 버튼이나 [Go Velocity] 버튼으로 모터를 전압이나 속도 구동해 봅니다.

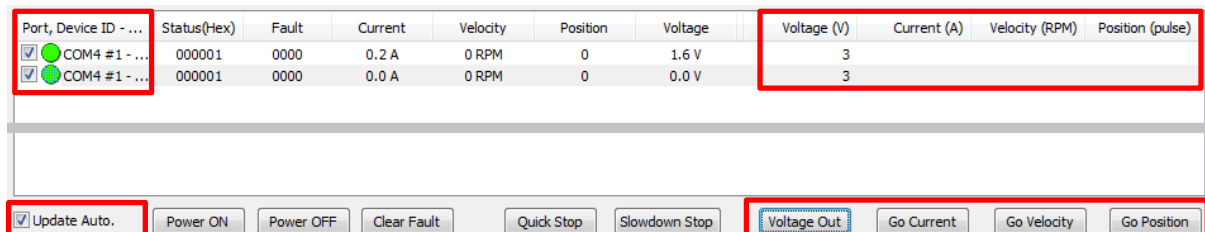


그림 5-9 모터 동시 구동

상기에서와 같이, 이동로봇의 구동 명령을 내리는데 Motor Control UI 유틸리티를 사용하는 것은 상당히 불편합니다. 로봇을 조종하는 좀 더 일반적인 방법으로 "5.3 조이스틱으로 로봇 조종"과 "5.4 RC 조종기로 로봇 조종"에서 조이스틱과 RC 조종기를 사용하는 방법에 대해 소개합니다.

5.2 4륜 스키드 조향 로봇

4륜 스키드 조향(Four Wheels Skid Steering) 로봇은 4개의 구동 바퀴를 가지는 구조로, 설계와 제작이 용이하고 제어 방법도 단순하기 때문에 군사, 탐사, 험지주행 등 실외 이동 로봇 플랫폼으로 주로 사용됩니다.

이러한 로봇은 회전 시 4개의 구동 바퀴에서 바퀴의 구름 방향과 직각으로 미끄러짐이 발생하기 때문에 아스팔트와 같은 바퀴와 지면간 마찰력이 큰 장소에서는 운용이 불가능합니다. 하지만 자갈, 모래, 흙 등 어느 정도의 슬립이 발생할 수 있는 장소에서 사용되며, 보통 사용자의 원격 조종에 의해 운용됩니다.

이 절에서는 4륜 스키드 조향 로봇에 대해 간략히 소개하고 듀얼 채널 MoonWalker 제어를 사용하여 로봇을 제작하고 주행해 보는 것에 대해 설명합니다.

5.2.1 로봇 기구부

그림 5-10과 같이, 4륜 스키드 조향 로봇은 가장 간단한 형태의 실외용 이동 로봇으로 앞 좌우 바퀴와 뒤 좌우 바퀴에 각각 모터가 연결되는 구조입니다.

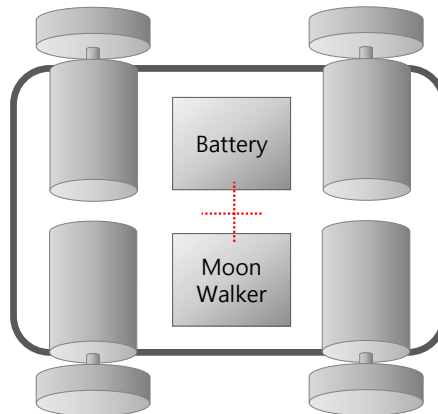


그림 5-10 4륜 구동로봇 플랫폼 구조

이 구조는 로봇의 중심이 회전 중심(상기 그림에서 적색 점선 십자가)과 일치함으로 아커만식 조향을 채택한 로봇에 비해 회전반경이 아주 짧은 장점이 있습니다. 하지만 조향 시 4개의 바퀴에서 슬립이 발생해야하기 때문에 바퀴와 지면간 마찰력이 큰 장소에서는 운행이 불가능합니다.

또한 4개의 바퀴가 항상 지면에 닿아야 하기 때문에 4개의 바퀴 축에 서스펜션을 장착하게 됩니다. 이로 인해 로봇의 몸체가 기울어지거나 흔들려 센서의 측정값을 보정하는데 어려움이 있습니다.

5.2.2 로봇 배선

그림 5-11과 같이 제어기와 배터리, 모터, 엔코더를 배선합니다. 이 때, 전력선들이 엔코더 신호선과 동일한 경로를 따라 배선되지 않도록 주의합니다. 그리고 엔코더 선은 최대한 짧게 배선합니다.

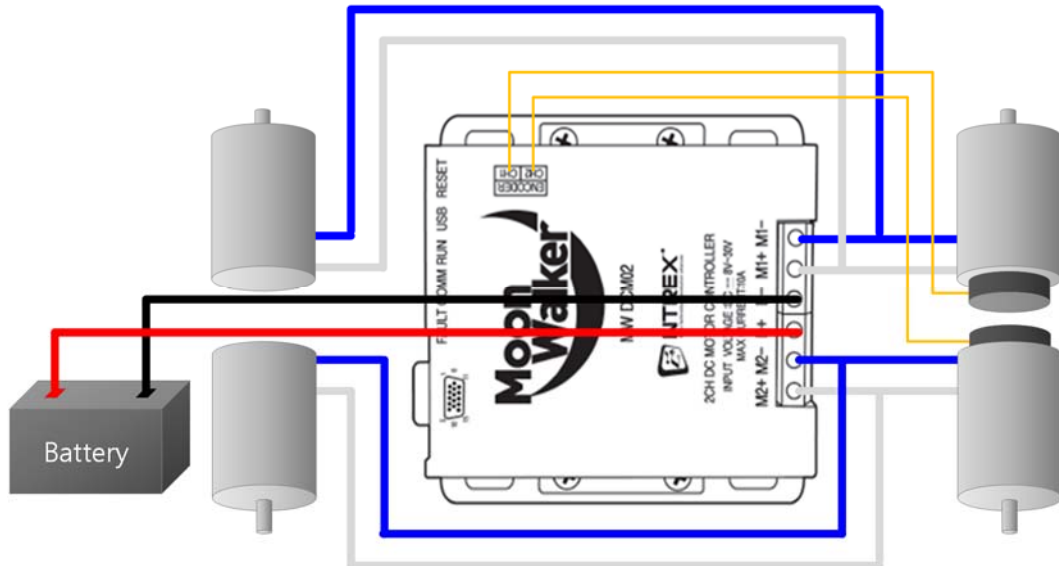


그림 5-11 4륜 스키드 조향 로봇의 배선

4개의 모터를 제어기에 연결하기 위해서는 그림 5-11과 같이 앞뒤 모터를 하나로 묶어 병렬로 연결하고 엔코더는 앞 쪽 좌우에만(혹은 뒤 쪽 좌우에만) 연결합니다. 이러한 구성은 모터의 구동력을 증가하기 위해 모터 두 개를 병렬로 연결하는 것과 같습니다. 이 경우, 제어기에서는 한 모터의 최대 전류보다 2배의 전류를 흘릴 수 있습니다.

모터에서 발생하는 토크가 밸런스를 이루기 위해서는 4개의 모터가 동일한 종류로 구성되어야 합니다. 그리고 앞쪽 모터를 기준으로 속도제어가 되기 때문에 앞뒤 모터간 속도의 불일치가 발생할 수 있습니다. 하지만 앞뒤 바퀴가 지면에 닿아 있기 때문에 앞뒤 두 모터가 기구적으로 구속된 효과를 냅니다.

5.2.3 제어기 설정 및 테스트

Motor Control UI 유틸리티로 로봇에 장착된 제어기의 파라미터를 설정하고 테스트 하는 것은 "5.1 2륜 차동 구동 로봇"의 내용과 동일하니 이 절을 참조하시기 바랍니다.

5.3 조이스틱으로 로봇 조종

이 절에 들어가기에 앞서 "5.1 2륜 차동 구동 로봇"이나 "5.2 4륜 스키드 조향 로봇"에서 설명한

대로 로봇에 제어기를 설치하고 관련 파라미터가 설정되어 있어야 합니다.

이 절에서는 2축 아날로그 조이스틱을 제어기에 연결하여 로봇을 조종하는 것에 대해 설명합니다.

아날로그 조이스틱 내부에는 2방향 조종을 위해 포텐서미터가 두 개 설치되어 있는 제품을 사용해야 합니다. 로봇의 조종에 사용한 조이스틱은 다음과 같습니다:

- 아날로그 조이스틱: (주)엔티렉스 조이스틱 200-M4

또한 제어기에 조이스틱을 연결하기 위해 제어기에는 2개 이상의 아날로그 입력 포트가 있어야 합니다. 제어기 모델에 따라 아날로그 입력 포트의 수는 다르기 때문에, 제품의 데이터시트를 참조하기 바랍니다.

5.3.1 조이스틱 연결 구성

그림 5-12과 같이 2채널 조이스틱의 포텐서미터를 제어기의 아날로그 입력 포트에 연결합니다.

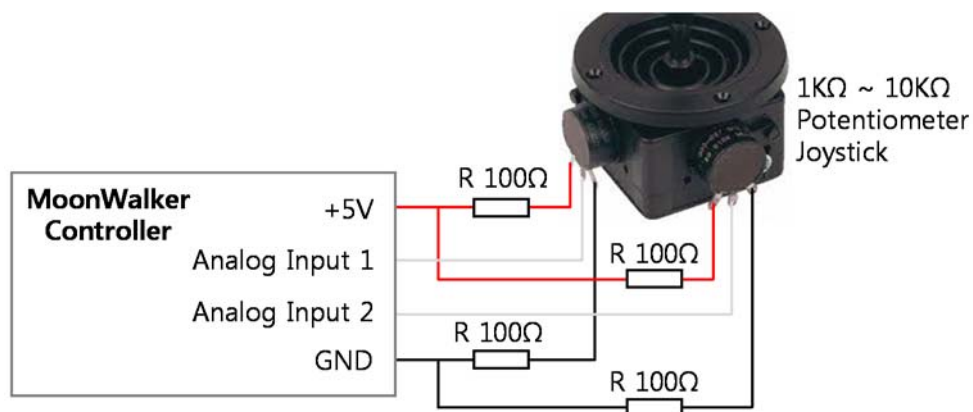


그림 5-12 제어기와 Joystick(아날로그 입력) 연결

조이스틱을 선정할 때는, 포텐서미터의 저항 값이 1KΩ에서 10KΩ 사이의 조이스틱을 사용하면 됩니다. 그리고 포텐서미터에 공급되는 +5V와 GND 배선에 각각 100Ω 정도의 저항을 직렬로 연결합니다. 이 저항은 Min/Max 안전 검사에 사용되는데, 아날로그 입력 값이 사용자가 설정한 최대 최소 범위를 벗어나는 경우 이상 상황으로 감지하고 모터를 안전하게 정지하게 됩니다. 이 기능을 사용하기 위해서는, Motor Control UI 유틸리티에서 캘리브레이션 작업을 통해 아날로그 입력의 최대, 최소 범위를 조정해야 합니다. 자세한 내용은 사용자 매뉴얼의 "10.8.3 min_max_safety - Min/Max Safety"을 참조하기 바랍니다.

5.3.2 아날로그 입력 채널의 파라미터 설정

Motor Control UI 유틸리티의 **우측 상단의 [Emergency STOP] 버튼을 눌러 제어기의 모든 모터를 Power OFF 상태로 만듭니다.** 이는 아날로그 입력 채널을 캘리브레이션 할 때 로봇의 모터가 따라 움직이는 것을 방지하기 위함입니다.



그림 5-13 Motor Control UI 유틸리티의 긴급 정지 버튼

그림 5-14과 같이 Configuration 탭의 Joystick/RC Control and Safety 그룹에서 **Control Mixing** 항목을 **Mixing Mode 1/2/3/4 중 하나로 선택** 합니다. 그리고 Center Safety Check('Center Safety' 오브젝트) 항목을 Enable로 설정합니다. 이는 아날로그 입력 값이 센터에서 시작되지 않을 때는 모터를 구동할 수 없도록 합니다. 그리고 Min/Max Safety Check('Min/Max Safety' 오브젝트) 항목을 Enable로 설정합니다. 이는 조이스틱이 고장 나거나 단선으로 인해 잘못된 값을 읽는 것을 방지합니다. 자세한 내용은 사용자 매뉴얼의 "10.8.2 center_safety - Center Safety"와 "10.8.3 min_max_safety - Min/Max Safety"을 참조하기 바랍니다.

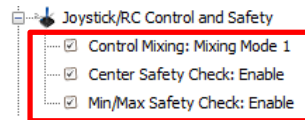


그림 5-14 Joystick/RC 조종의 믹싱과 안전검사 설정

아날로그 입력 채널을 활성화 하기 위해 Analog Input Channel 1과 Analog Input Channel 2에서 Enable 항목을 체크합니다. 그리고 아날로그 입력 채널로 모터의 전압/속도 명령을 매핑하기 위해 Input Channel Mapping 서브 그룹에서 Target 항목을 각각 Motor Channel 1과 Motor Channel 2로 선택하고 Action 항목을 Command: Voltage 또는 Command: Velocity 중 하나로 선택합니다.

※ 믹싱 기능(Control Mixing 항목)은 Position과 Current 명령에 대해서는 동작하지 않습니다.

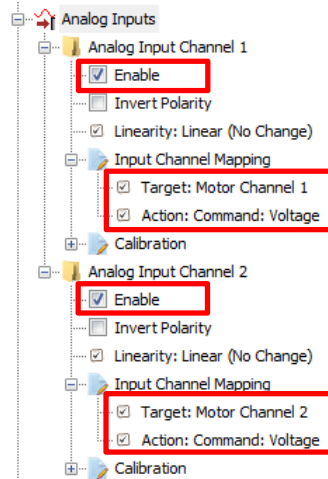


그림 5-15 아날로그 입력 채널의 설정

모든 설정이 끝나면 [Write configurations] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 합니다.

※ 다음 단계인 캘리브레이션 작업을 진행하기 위해서는 아날로그 입력 채널이 활성화(Enable 항목 체크) 되어야 합니다.

5.3.3 캘리브레이션 파라미터 설정

아날로그 입력 신호(범위: 0 ~ 5V)를 MCU에서 A/D 변환하여 읽은 값은 0에서 4095 범위의 원시 값(raw value)이 됩니다. 이 값은 정규화 과정을 거쳐 -1과 1 사이의 값으로 변환됩니다.

캘리브레이션은 원시 값을 정규화 된 값으로 변환하기 위한 파라미터를 찾는 과정입니다.

캘리브레이션 하고자 하는 아날로그 입력 채널에서 Calibration 그룹을 마우스로 클릭 하면 오른쪽에 [...] 버튼이 나타납니다. 그리고 이 버튼을 클릭합니다.

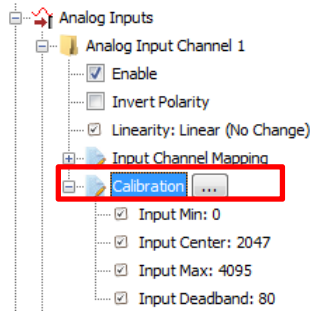


그림 5-16 아날로그 입력 채널에서 캘리브레이션 선택

그러면 그림 5-17과 같이 선택된 아날로그 입력 채널에 대해 Calibration 대화상자가 나타나고 대한 캘리브레이션을 진행할 수 있게 됩니다.

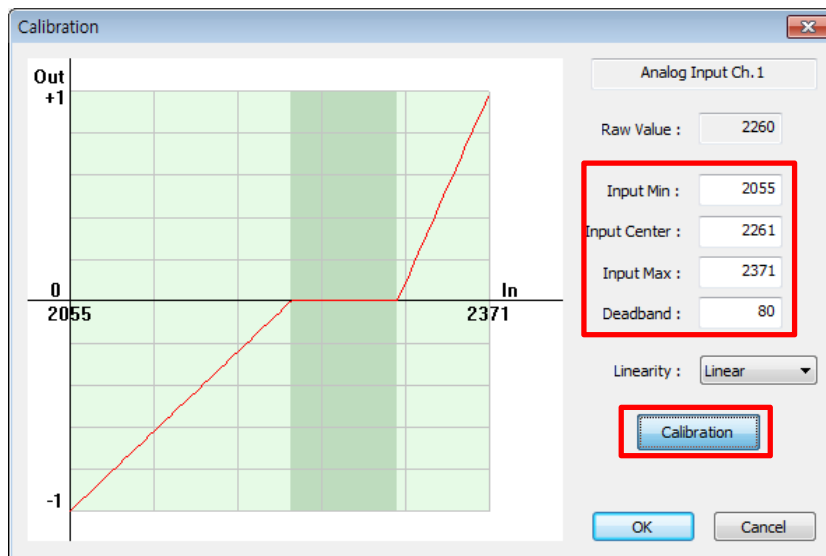


그림 5-17 Calibration 대화상자

대화상자에서 [Calibration] 버튼을 클릭한 후 조이스틱을 상하 좌우 최대 범위로 움직여서 Input Min 값과 Input Max 값을 찾습니다. 그리고 조이스틱을 중앙에 두고 [Calibration] 버튼을 한 번 더 눌러 Input Center 값을 설정합니다. 캘리브레이션에 대한 자세한 내용은 사용자 매뉴얼에서 "16.8.18 Calibration"을 참고하기 바랍니다.

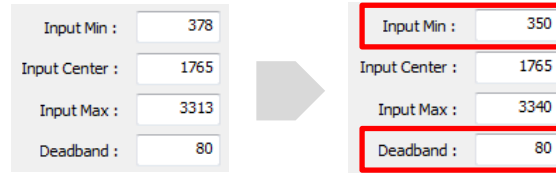


그림 5-18 Input Min, Input Max 범위 확장

※ 캘리브레이션을 완료한 후, 그림 5-18과 같이 대화상자 오른 편의 Input Min 값과 Input Max 값의 범위를 약간 넓혀야 합니다. Min/Max Safety Check가 Enable로 설정된 경우, 입력 값이 Min이나 Max 값 근처에 있을 때 Min/Max 안전 검사에 실패할 가능성이 있습니다.

마지막으로 Configuration 탭에서 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

아날로그 입력 채널 1에 대한 캘리브레이션 과정이 끝나면 아날로그 입력 채널 2에 대해서도 위와 동일한 방법으로 캘리브레이션 과정을 진행합니다.

5.3.4 로봇의 조종

Configuration 탭에서 파라미터의 설정이 끝나면 그림 5-19와 같이 I/O Monitoring 탭에서 Analog Input 창이 활성화 되어있고 조이스틱의 조종 값이 표시되는지 확인합니다.

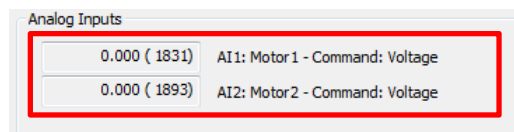


그림 5-19 아날로그 입력 채널의 입력 값 모니터링

그림 5-19에서와 같이 하나의 편집 박스에 두 숫자가 표시되는데, 괄호 바깥의 수는 -1과 1 범위를 가지는 정규화 된 값이고 괄호 안의 수는 원시 값(범위: 0 ~ 4095)입니다. 정규화 된 값에 'Max Velocity'나 'Max Voltage' 오브젝트의 값이 곱해져 속도 명령이나 전압 명령으로 변환됩니다.

이제 조이스틱으로 로봇을 조정해 볼 것입니다. 이 때 로봇이 갑자기 움직여 위험한 상황을 초래할 수도 있기 때문에, 지면에서 로봇의 바퀴를 들어올려 바퀴가 자유롭게 움직일 수 있도록 합니다.

Motor Control 탭에서 Channel 1과 Channel 2의 모터에 대해 [Power ON] 버튼을 눌러 두 모터 모두 Power ON 상태로 만듭니다.

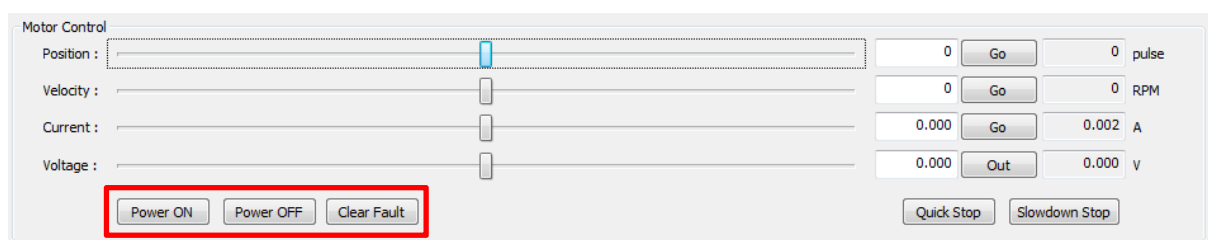


그림 5-20 모터 Power ON

이제 조이스틱을 움직여서 모터가 정상적으로 동작하는지 확인합니다. 이 때 Motor Control UI 유틸리티의 헤더에서 [Real-time Plot] 버튼을 눌러 Real-time Plot 창을 실행하고, 그래프 오브젝트로 두 개의 '**AI Converted Value**'를 선택하고 Channel 1과 2를 각각 설정합니다. 그리고 '**Voltage**'(속도 명령일 때는 '**Velocity**')를 선택하여 실제로 모터에 명령으로 내려가는 값을 모니터링 합니다.

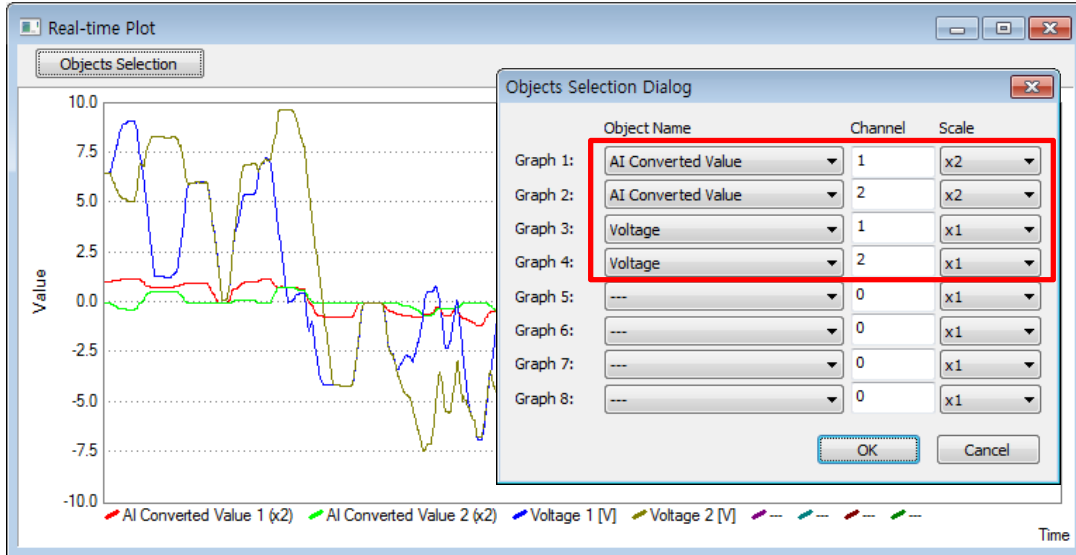


그림 5-21 아날로그 입력 채널과 명령 모니터링

그림 5-21의 그래프에서 AI Converted Value 1/2는 조이스틱에서 입력된 값을 정규화하여 표시합니다. 그리고 Voltage 1/2는 실제 제어기에 내려지는 전압 명령을 표시합니다. 전압 명령이 아날로그 입력 값을 그대로 추정하지 않는 것을 볼 수 있는데, 이는 아날로그 입력 채널 1과 2의 값이 믹싱 기능에 의해 섞였기 때문에 나타나는 현상입니다.

로봇이 원하는 대로 조종된다면 이제 바닥에 내려 놓고 다시 테스트 해 봅니다. 이때 로봇의 관성에 의해 제어기의 이득을 새로 설정해야 할 수도 있습니다.

5.4 RC 조종기로 로봇 조종

제어기에는 조이스틱뿐만 아니라 펄스 입력 포트에 RC 수신기를 연결해서 RC 조종기로 로봇을 원격 조종할 수 있습니다.

이 절에서는 RC 수신기의 PWM 신호에서 펄스 폭(Pulse Width)을 캡처하여 로봇을 조종하는 방법에 대해 설명합니다.

이 절에서 로봇의 조종에 사용한 RC 조종기와 수신기는 다음과 같습니다:

- RC 조종기: Hitec Optic-6 Sports (2.4GHz)
- RC 수신기: Hitec Optima-6 (2.4GHz)

RC 조종기와 수신기는 2개 이상의 신호 채널을 가지고 있어야 합니다. 그리고 제어기에 RC 수신

기를 연결하기 위해 제어기는 2개 이상의 펄스 입력 포트가 있어야 합니다. 제어기 모델에 따라 펄스 입력 포트의 수가 다르기 때문에, 제품의 데이터시트를 참조하기 바랍니다.

5.4.1 RC 조종기 연결 구성

그림 5-22과 같이 RC 수신기와 제어기의 펄스 입력 포트를 연결합니다. 제어기 MW-MDC24D200D와 MW-MDC24D500S, MW-MDC24D500D 모델에서는 'MoonWalker MW DSub Cable'을 사용하면 제어기와 RC 수신기 연결을 간단히 할 수 있습니다. "2 제어기 및 주변 부품 선정 안내"를 참고하기 바랍니다.

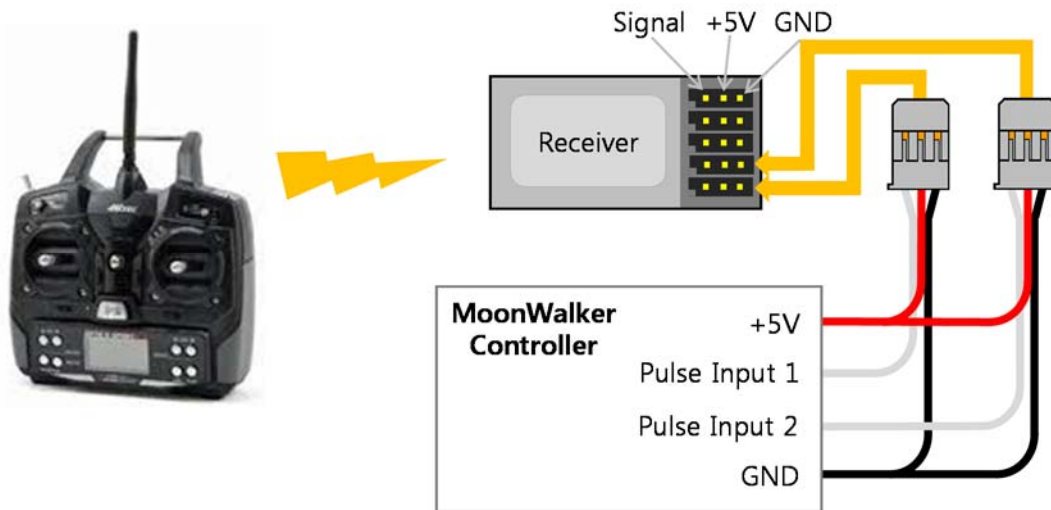


그림 5-22 제어기와 RC 조종기 연결

RC 케이블은 전원 공급과 제어 신호를 위해 다음과 같이 세 가닥의 선을 사용합니다:

- GND 선: 검정 or 갈색
- +5V 선: 적색 (4.8 ~ 6V)
- Signal 선: 백색 or 황색 (3 ~ 5V)

RC 송신기는 무선 신호를 출력하고, 수신기는 무선 신호를 수신하여 펄스 폭(pulse width) 제어 신호를 만들어냅니다.

RC 수신기의 신호 주기는 20ms를 기준으로 약 ± 1 의 오차를 허용합니다. 그리고 펄스 폭 동작 범위는 0.8ms와 2.2ms 사이가 되며, 1.5ms(조종기의 중앙)일 경우 모터는 정지 상태가 됩니다. 그리고 1ms(조종기의 최소)일 경우 모터는 역방향 최고 속도/전압 명령이 가해지고, 2ms(조종기의 최대)일 경우 모터는 정방향 최고 속도/전압 명령이 가해집니다 (그림 5-23 참조).

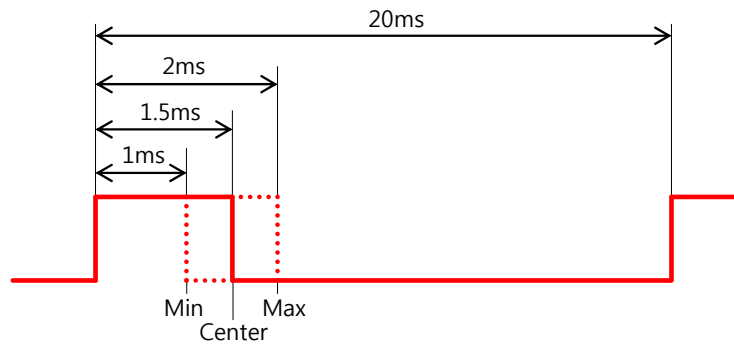


그림 5-23 RC 수신기의 PWM 출력 신호

만일 RC 조종기와 수신기간에 연결이 이루어지지 않으면, 조종기와 수신기의 Crystal 주파수가 일치하는지 확인하고 RC 수신기의 신호 단자에서 위와 같은 신호가 발생하는지 오실로스코프로 확인하기 바랍니다.

5.4.2 펄스 입력 채널의 파라미터 설정

Motor Control UI 유틸리티의 **우측 상단의 [Emergency STOP] 버튼을 눌러 제어기의 모든 모터를 Power OFF 상태로 만듭니다.** 이는 펄스 입력 채널을 캘리브레이션 할 때 로봇의 모터가 따라 움직이는 것을 방지하기 위함입니다.

그림 5-24과 같이 Configuration 탭의 Joystick/RC Control and Safety 그룹에서 **Control Mixing 항목을 Mixing Mode 1/2/3/4 중 하나로 선택** 합니다. 그리고 Center Safety Check('Center Safety' 오브젝트) 항목을 Enable로 설정합니다. 이는 펄스 입력 값이 센터에서 시작되지 않을 때는 모터를 구동할 수 없도록 합니다. 그리고 Min/Max Safety Check('Min/Max Safety' 오브젝트) 항목을 Enable로 설정합니다. 이는 RC 수신기와 제어기간 단선으로 인해 잘못된 값을 읽는 것을 방지합니다. 자세한 내용은 사용자 매뉴얼의 "10.8.2 center_safety - Center Safety"와 "10.8.3 min_max_safety - Min/Max Safety"를 참조하기 바랍니다.

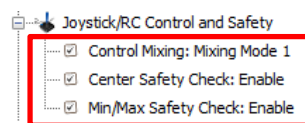


그림 5-24 Joystick/RC 조종의 믹싱과 안전검사 설정

펄스 입력 채널을 활성화 하기 위해 그림 5-25에서와 같이 Pulse Input Channel 1과 Pulse Input Channel 2 그룹에서 Enable 항목을 체크합니다. 그리고 펄스 입력 채널을 모터의 전압/속도 명령으로 매핑하기 위해 Input Channel Mapping 그룹에서 Target 항목을 각각 Motor Channel 1과 Motor Channel 2로 선택하고 Action을 Command: Voltage 또는 Command: Velocity 중 하나로 선택합니다.

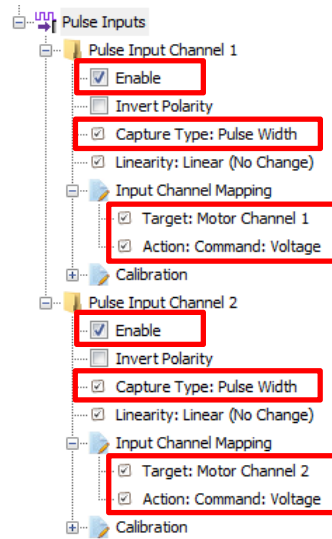


그림 5-25 펄스 입력 채널의 설정

모든 설정이 끝나면 [Write configurations] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 합니다.

만약 펄스 입력 채널을 UI에서 활성화 하고 제어기에 변경된 값을 저장하지 않으면 다음 단계인 캘리브레이션 작업을 진행할 수 없습니다.

5.4.3 캘리브레이션 파라미터 설정

캘리브레이션에 대한 설명은 "5.3.3 캘리브레이션 파라미터 설정"을 참고하기 바랍니다.

5.4.4 로봇의 조종

Configuration 탭에서 파라미터의 설정이 끝나면 그림 5-26과 같이 I/O Monitoring 탭에서 Pulse Input 항목이 활성화 되어있고 RC 조종기의 값이 표시되는지 확인합니다.

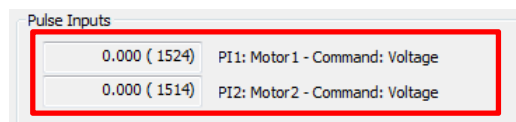


그림 5-26 펄스 입력 채널의 값 모니터링

그림 5-26에서와 같이 하나의 편집 박스에 두 숫자가 표시되는데, 괄호 바깥의 수는 -1과 1 범위를 가지는 정규화 된 값이고 괄호 안의 수는 원시 값(펄스 폭을 마이크로 초 단위로 측정한 값)입니다. 정규화 된 값에 'Max Velocity'나 'Max Voltage' 오브젝트의 값이 곱해져 속도 명령이나 전압 명령으로 변환됩니다.

이제 RC 조종기로 로봇을 조정해 볼 것입니다. 이 때 로봇이 갑자기 움직여 위험한 상황을 초래할 수도 있기 때문에, 지면에서 로봇의 바퀴를 들어 올려 바퀴가 자유롭게 움직일 수 있도록 합니다.

니다.

Motor Control 탭에서 채널 1과 채널 2의 모터에 대해 [Power ON] 버튼을 눌러 두 모터 모두 Power ON 상태로 만듭니다.

이제 RC 조종기의 스틱을 움직여서 로봇의 바퀴가 RC 조종기의 조종에 따라 회전하는지 확인합니다. 이 때 Motor Control UI 유틸리티의 헤더에서 [Real-time Plot] 버튼을 눌러 Real-time Plot 창을 실행하고 그래프 오브젝트로 두 개의 '**PI Converted Value**'를 선택하고 각각 Channel 1과 Channel 2를 설정합니다. 그리고 '**Voltage**'(속도 명령일 때는 '**Velocity**')를 선택하여 실제로 모터에 명령으로 내려가는 값을 모니터링 합니다.

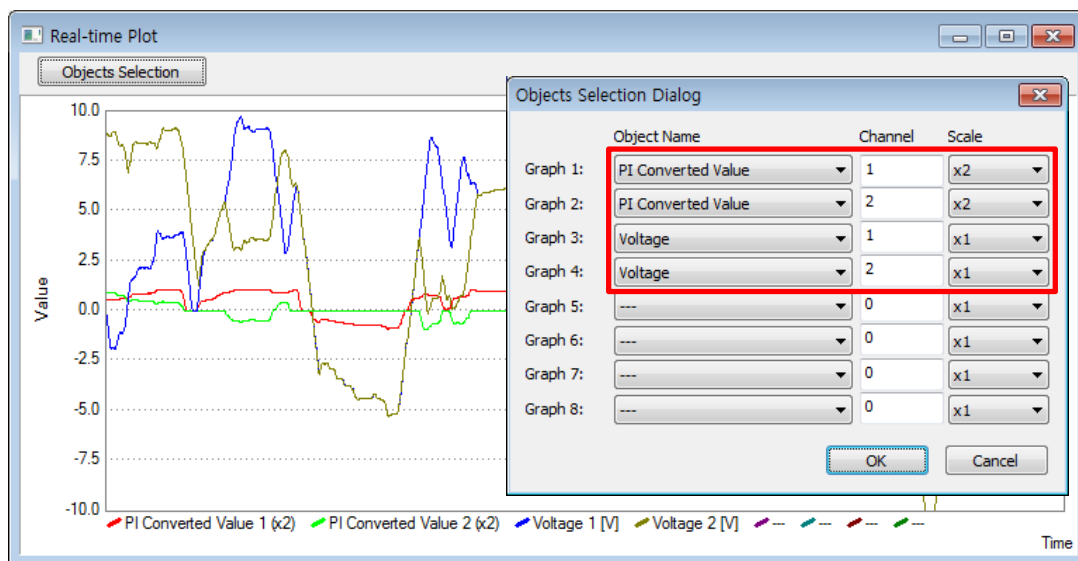


그림 5-27 펄스 입력 채널과 명령 모니터링

그림 5-27의 그래프에서 AI Converted Value 1/2는 조이스틱에서 입력된 값을 정규화하여 표시합니다. 그리고 Voltage 1/2는 실제 제어기에 내려지는 전압 명령을 표시합니다. 전압 명령이 아날로그 입력 값을 그대로 추정하지 않는 것을 볼 수 있는데, 이는 아날로그 입력 채널 1과 2의 값이 믹싱 기능에 의해 섞였기 때문에 나타나는 현상입니다.

로봇이 원하는 대로 조종된다면 이제 바닥에 내려 놓고 다시 테스트 해 봅니다. 이때 로봇의 관성에 의해 제어기의 이득을 새로 설정해야 할 수도 있습니다.

6 아날로그·펄스 입력 포트의 사용

제어기의 아날로그 입력 포트에는 전압을 출력하는 아날로그 센서를 연결해서 사용하거나 포텐서 미터(Potentiometer)나 타코미터(Tachometer) 같은 아날로그 신호를 출력하는 센서를 연결해서 사용합니다. 그리고 써미스터를 연결해서 외부 온도를 감지할 수도 있습니다.

펄스 입력 포트에는 RC 조종기와 같이 펄스를 기반으로 하는 장치를 연결할 수 있습니다. 펄스 입력으로 펄스 폭(Pulse Width), 주파수(Frequency), 듀티비(Duty cycle) 중 하나를 사용자가 선택하여 캡처할 수 있습니다. 각 캡처 타입에 따른 입력 범위는 다음과 같습니다:

- Pulse Width: 0 ~ 50000 μ s
- Frequency: 20 ~ 20kHz
- Duty Cycle: 0 ~ 1000%

이번 장에서는 제어기의 아날로그/펄스 입력 포트를 활용하는 사례들에 대해 소개합니다.

이번 장에서 소개하는 내용을 진행하기 위해서는 다음 준비물이 필요합니다:

- 아날로그 신호 출력 절대 엔코더
- 펄스 신호 출력 절대 엔코더

6.1 절대 엔코더의 사용

절대 엔코더(Absolute Encoder)는 증분 엔코더(Incremental encoder)와 달리 전원이 꺼지더라도 절대 위치 값을 유지할 수 있으며 전원이 공급되지 않는 상태에서 위치 이동이 발생하여도 전원 투입 후 곧바로 현재의 위치정보를 읽어올 수 있습니다.

여기서는 US Digital사의 자기식 절대 엔코더 MA3 Miniature Absolute Magnetic Shaft Encoder를 사용하여 모터의 절대 위치를 읽어오는 응용을 구성해 보겠습니다.

MA3 절대 엔코더는 PWM 출력과 아날로그 출력의 두 가지 타입이 있습니다.

표 6-1 MA3 시리즈 핀 아웃

Pin	MA3-A(아날로그 출력)	MA3-P10/12(PWM 출력)
1	+5V DC power	+5V DC power
2	Analog output	PWM output
3	Ground	Ground

6.1.1 모터에 절대 엔코더의 연결

모터에 절대 엔코더를 연결할 때는, 그림 6-1과 같이 모터의 감속기를 거친 말단에 연결하여 최종 회전각을 측정합니다.

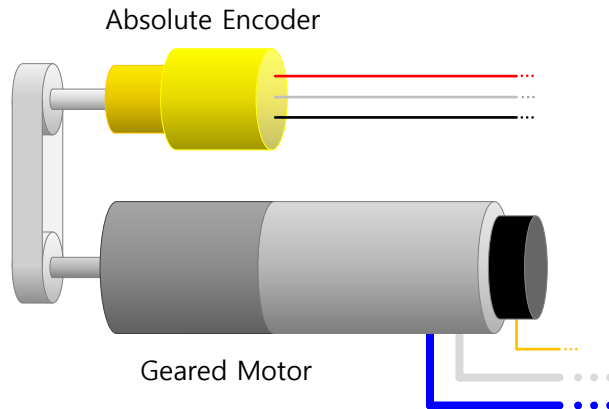


그림 6-1 모터와 절대 엔코더의 기구적 연결

절대 엔코더가 출력단의 절대 위치를 측정하지만 감속기와 벨트 등 몇 단계의 기구적 운동 전달 과정을 거치게 되어, 모터의 회전 위치에 대한 정밀도가 떨어집니다. 따라서 절대 엔코더를 위치 제어기나 속도 제어기의 피드백으로 사용하는 것은 좋지 않은 방법입니다.

※ 절대 엔코더는 제어기에 전원이 켜지면 처음 한번 값을 읽어 모터의 위치를 파악하고, 이후 위치 제어기나 속도 제어기의 피드백으로는 증분 엔코더를 사용하는 것이 좋습니다.

절대 엔코더는 보통 회전 범위가 제한되어 있습니다. 일반적으로 사용되는 Single turn 절대 엔코더는 0° 와 360° 범위에서 측정하기 때문에 회전 각도는 측정하지만 회전 수는 측정할 수 없게 됩니다. 360° 이상을 측정하기 위해서는 Multi turn 절대 엔코더를 사용하여야 합니다.

6.1.2 아날로그 출력형 절대 엔코더

그림 6-2에서와 같이 제어기의 아날로그 입력 포트에 아날로그 출력형 절대 엔코더 MA3-A를 연결합니다.

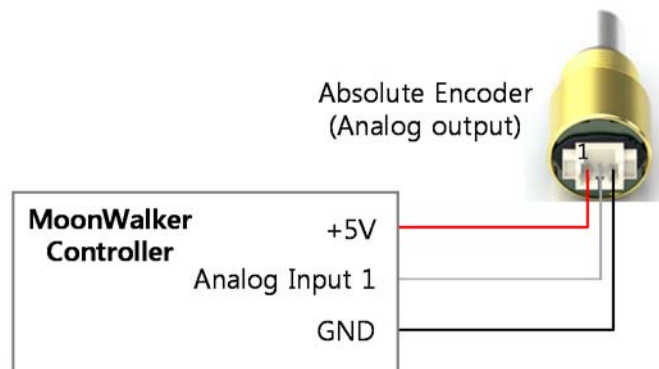


그림 6-2 아날로그 입력 포트에 절대 엔코더 연결

Motor Control UI 유틸리티의 Configuration 탭에서 그림 6-3와 같이 아날로그 입력 채널의 파라미터를 설정합니다. 아날로그 입력 채널을 활성화 하기 위해 Analog Input Channel 1 그룹의 Enable 항목을 체크합니다. 그리고 아날로그 입력 채널의 매핑 기능은 사용하지 않도록 설정합니다.

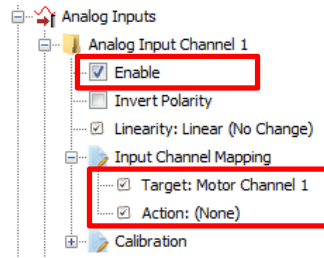


그림 6-3 아날로그 입력 채널의 설정

모든 설정이 끝나면 [Write configurations] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 합니다.

만약 아날로그 입력 채널을 UI에서 활성화 하고 제어기에 변경된 값을 저장하지 않으면 다음 단계인 캘리브레이션 작업을 진행할 수 없습니다. 캘리브레이션에 대한 설명은 "5.3.3 캘리브레이션 파라미터 설정"을 참고하여 진행합니다.

※ 캘리브레이션을 완료한 후 **Input Min**과 **Input Max** 항목의 값은 그대로 둡니다. 하지만 **Input Center** 항목의 값은 $(\text{Input Min} + \text{Input Max})/2$ 로 바꾸어 줍니다. 그리고 **Input Deadband** 항목의 값은 0으로 설정합니다.

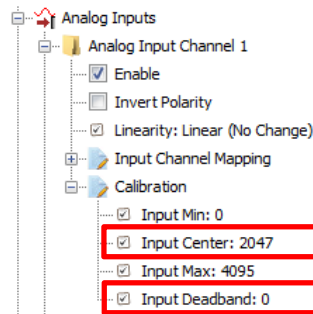


그림 6-4 캘리브레이션 값 조정

6.1.3 PWM 출력형 절대 엔코더

그림 6-5에서와 같이 제어기의 펄스 입력 포트에 PWM을 출력하는 절대 엔코더 MA3-P10/12를 연결합니다.

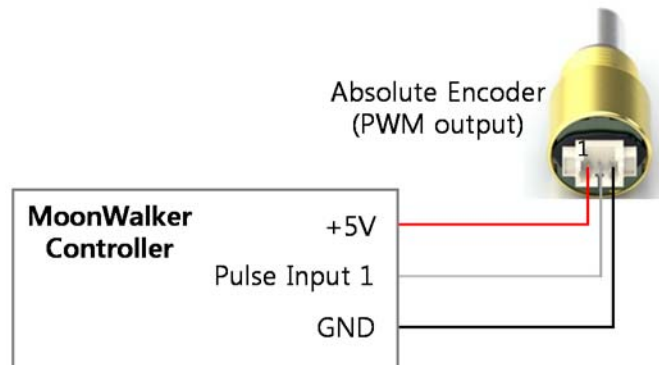


그림 6-5 펄스 입력 포트에 절대 엔코더 연결

PWM을 출력하는 절대 엔코더를 사용할 때는 PWM Frequency와 Minimum/Maximum Pulse Width를 살펴봐야 합니다. MA3-P10와 MA3-P12 두 제품의 PWM Frequency는 975Hz와 244Hz로 위치 제어기의 동작 주파수인 100Hz보다 크기 때문에 문제가 없습니다. 하지만 Minimum Pulse Width는 두 제품 모두 1μs로 제어기의 펄스 입력 채널에서 인식하는 최소 펄스 폭인 10μs보다 짧기 때문에, PWM 펄스 폭이 0에 가까워지면 제어기가 PWM 신호를 제대로 인식 하지 못할 수도 있습니다.

Motor Control UI 유틸리티의 Configuration 탭에서 그림 6-6과 같이 펄스 입력 채널의 파라미터를 설정합니다. 펄스 입력 채널을 활성화 하기 위해 Pulse Input Channel 1 그룹의 Enable 항목을 체크합니다. 그리고 Capture Type 항목을 Pulse Width로 설정합니다. 그리고 펄스 입력 채널의 맵핑 기능은 사용하지 않도록 설정합니다.

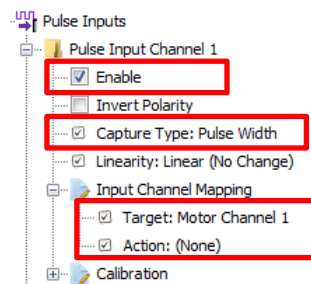


그림 6-6 펄스 입력 채널의 설정

모든 설정이 끝나면 [Write configurations] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 합니다.

만약 펄스 입력 채널을 UI에서 활성화 하고 제어기에 변경된 값을 저장하지 않으면 다음 단계인 캘리브레이션 작업을 진행할 수 없습니다. 캘리브레이션에 대한 설명은 "5.3.3 캘리브레이션 파라미터 설정"을 참고하여 진행합니다.

※ 캘리브레이션을 완료한 후 Input Min과 Input Max 항목의 값은 그대로 둡니다. 하지만 Input Center 항목의 값은 $(\text{Input Min} + \text{Input Max})/2$ 로 바꾸어 줍니다. 그리고 Input Deadband 항목의 값은 0으로 설정합니다.

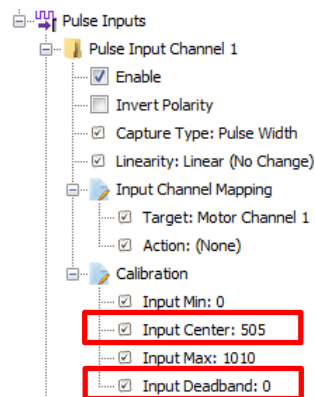


그림 6-7 캘리브레이션 값 조정

6.1.4 절대 엔코더 테스트

그림 6-8와 같이 I/O Monitoring 탭에서 Analog Input 또는 Pulse Input 그룹의 해당 값이 올바르게 들어오는지 확인합니다.

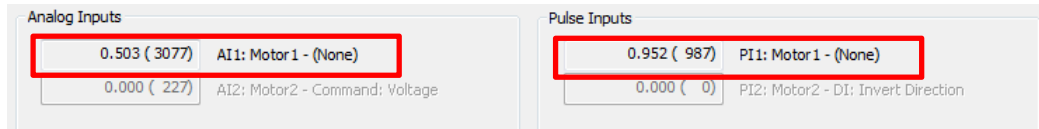


그림 6-8 아날로그/펄스 입력 채널 모니터링

다음 식에 의해 모터의 위치('Position' 오브젝트)를 종단의 상대 각(Relative Angle)으로 바꿀 수 있습니다.

$$Relative\ Angle = Position / (Encoder\ Resolution \times Gear\ Ratio)$$

여기서 Encoder Resolution은 증분 엔코더가 1회전하였을 때 카운트 되는 펄스 수이고 Gear Ratio는 모터 감속기의 감속비입니다.

그리고 다음 식에 의해 절대 엔코더의 정규화 된 값(Normalized Value)을 종단의 절대 각(Absolute Angle)로 바꿀 수 있습니다.

$$Absolute\ Angle = (Max\ Angle - Min\ Angle) / 2 \times Normalized\ Value + (Max\ Angle + Min\ Angle) / 2$$

여기서 Max Angle은 종단의 최대 각(정규화 된 값이 1일 때)이고 Min Angle은 종단의 최소 각(정규화 된 값이 -1일 때)입니다.

모터를 구동하여 절대 엔코더를 회전함에 따라 Relative Angle 값 Absolute Angle 값이 일정한 오프셋 값을 유지하면서 변하는지 살펴봅니다.

6.2 외부 위치 센서 피드백

이전 절에서 절대 엔코더를 사용하는 방법에 대해 살펴보았습니다. 이번 절에서는 절대 엔코더를 위치 제어기의 피드백 값으로 사용하는 방법에 대해 설명합니다.

앞 절에서 절대 엔코더를 위치 제어기의 피드백으로 사용하지 말 것을 당부하였으나, 부득이하게 절대 엔코더를 위치 제어기의 피드백으로 사용해야 할 경우에 이 절을 참고하기 바랍니다.

※주의※ 포텐서미터, 절대 엔코더와 같은 위치기반 센서를 위치 제어기의 피드백으로 사용하면 모터의 제어 성능이 크게 떨어집니다.

6.2.1 피드백 관련 파라미터 설정

여기서 절대 엔코더는 명령 입력 장치로 사용되지 않기 때문에, 그림 6-9와 같이 Center 안전 검사와 Min/Max 안전 검사 기능을 Disable 하고 믹싱 기능도 Separate로 설정합니다.

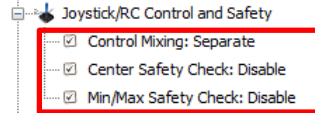


그림 6-9 안전 검사 기능 Disable

그림 6-11과 같이 아날로그/펄스 입력 채널의 매핑을 해당 모터에 대한 위치 피드백으로 설정합니다.

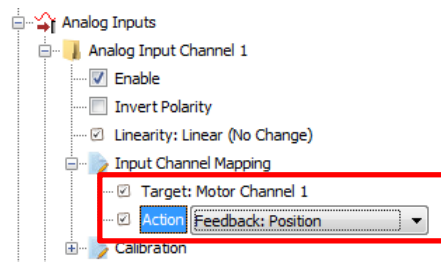


그림 6-10 아날로그 입력 채널의 매핑 설정

그림 6-11과 같이 피드백 센서로 Potentiometer를 선택합니다. 피드백 센서로 Potentiometer를 선택하면 제어기가 아날로그나 펄스 입력 포트에 연결된 위치 센서로부터 피드백을 받게 됩니다.

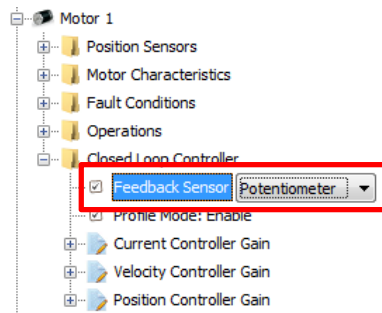


그림 6-11 Feedback Sensor 선택

6.2.2 스케일 변환 파라미터 및 제어기 이득 설정

아날로그/펄스 입력 채널에서 변환과정을 거쳐 읽은 '**AI Converted Value**'/'**PI Converted Value**' 오브젝트의 값은 -1과 1 사이의 정규화 된 값입니다. 이 값을 위치제어기의 피드백 값으로 사용하기 위해서는 '**Min Position**'과 '**Max Position**' 오브젝트의 범위로 스케일 변환합니다. 이에 대한 자세한 내용은 사용자 매뉴얼의 "7.2.5 위치 피드백 스케일 변환"을 참고하기 바랍니다.

제어기에 내려지는 위치 명령과 피드백 값의 스케일 변환을 위해 Position Sensors 그룹의 항목들을 그림 6-12와 같이 설정합니다. 이때 **가상의 엔코더**를 상상하고 엔코더의 해상도와 최대/최소 위치 값을 설정합니다.

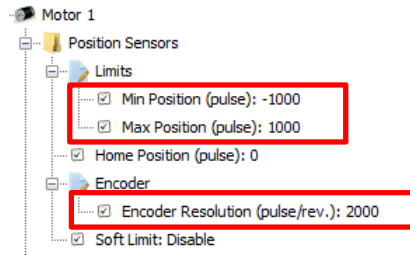


그림 6-12 위치 센서 관련 파라미터 설정

위치 제어에서 엔코더의 해상도(상기 그림에서 Encoder Resolution 항목)는 사용되지 않습니다. 하지만 **엔코더의 해상도가 설정되어 있어야 모터의 회전속도를 계산할 수 있게 됩니다. 따라서 이 값은 꼭 설정되어야 합니다.**

상기와 같이 설정된 경우, 위치 센서가 최소 위치에 있을 때(아날로그 입력 값이 -1일 때) pulse로 환산된 모터의 위치는 -1000이 됩니다. 그리고 위치 센서가 최대 위치에 있을 때(아날로그 입력 값이 1일 때) pulse로 환산된 모터의 위치는 1000이 됩니다. 그리고 모터의 위치가 2000pulse (Encoder Resolution에서 설정한 값)만큼 변하면 모터는 1회전 하였다고 판단합니다.

그림 6-13와 같이 Motor Characteristics 그룹에서 속도 프로파일을 생성하기 위한 최고 속도, 가속도, 감속도 값을 설정합니다.

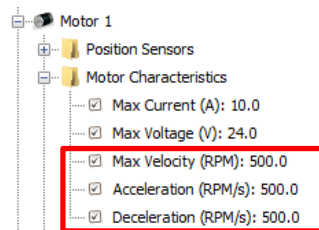


그림 6-13 속도 프로파일 관련 파라미터 설정

마지막으로 그림 6-14와 같이 Current Controller Gain 그룹과 Velocity Controller Gain 그룹, Position Controller Gain 그룹에서 위치 제어기와 속도 제어기, 전류제어기의 이득을 설정합니다.

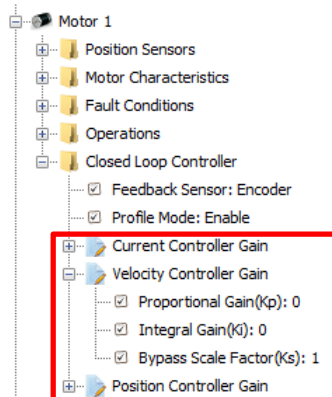


그림 6-14 위치, 속도, 전류 제어기의 이득 설정

전류 제어기의 이득 설정은 "3.5 모터의 토크(전류) 구동"절을 참고하기 바랍니다.

속도 제어기는 사용하지 않을 것이므로 Proportional Gain(Kp)와 Integral Gain(Ki) 항목의 값을 0으

로 설정합니다. 그리고 위치 제어기의 출력을 바이패스 시키기 위해 Bypass Scale Factor 항목을 1로 설정합니다. 여기서 속도 제어기를 사용하지 않는 이유는, 절대 엔코더가 모터의 속도를 계산할 만큼 충분한 해상도를 제공하지 않으며 위치 값에 포함된 미미한 노이즈도 차분으로 속도를 계산했을 때 크게 증폭되기 때문입니다.

위치 제어기의 이득 설정은 "4.3 모터의 위치 구동"을 참고하기 바랍니다.

모든 설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

6.2.3 위치 센서 피드백 테스트

그림 6-15와 같이 I/O Monitoring 탭에서 Analog Inputs 그룹에서 절대 엔코더의 값을 모니터링합니다.

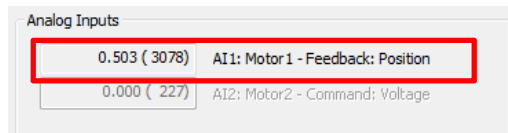


그림 6-15 아날로그 입력 값 모니터링

그리고 그림 6-16과 같이 Motor Control 탭에서 피드백 값을 모니터링 할 수 있습니다. 피드백 값은 절대 엔코더의 신호를 아날로그나 펄스 입력 채널로부터 읽어 -1과 1사이의 정규화 된 값을 표시합니다. 이때 화면에 표시하기 위해 사용하는 단위는 퍼밀(‰) 입니다.

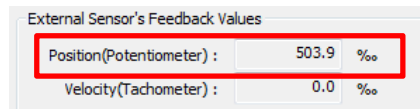


그림 6-16 피드백 값 모니터링

마지막으로 Position 바를 움직여 모터의 위치제어가 진동이나 오버슈트 없이 잘 되는지 확인합니다. 그림 6-17과 같이 Real-time Plot 창에서 '**Velocity**', '**Position**', '**Position Command**' 오브젝트를 모니터링 하면 다음과 같습니다.

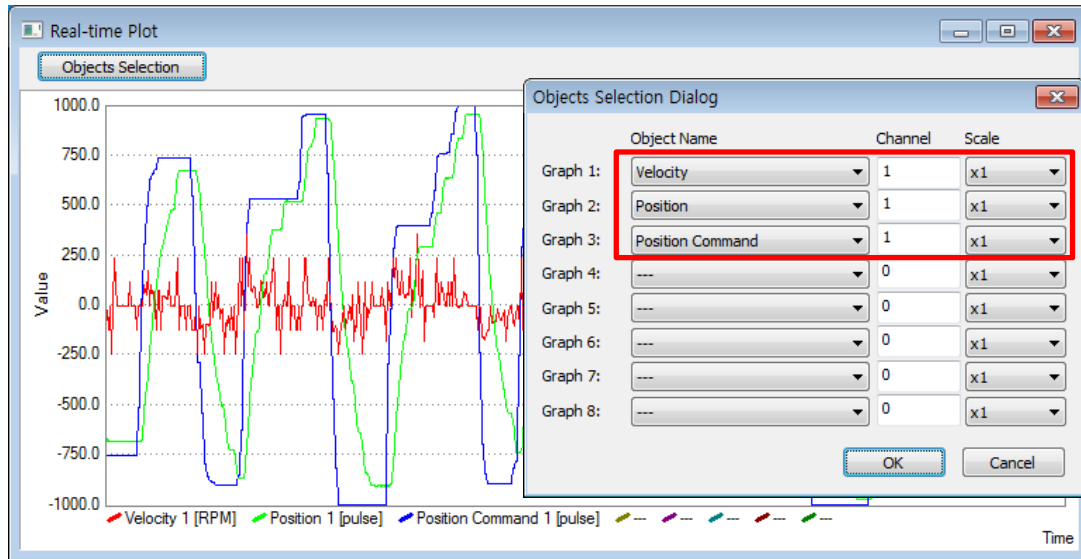


그림 6-17 절대 엔코더를 사용한 위치제어 모니터링

상기 그래프에서 '**Velocity**' 오브젝트에 대한 그래프(적색 그래프)는 노이즈가 심한 것을 볼 수 있습니다. 이와 같은 이유로, 외부 위치 센서 피드백으로 위치 제어 시 속도 제어를 사용할 수 없습니다.

7 디지털 입력 포트의 사용

제어기의 디지털 입력 포트는 근접센서나 터치센서와 같은 외부 디지털 입력 신호를 받아서 모터 정지 또는 정/역방향 전환, 리미트 등의 기능을 수행하는데 사용합니다.

디지털 입력은 외부의 신호가 MCU에 직접 입력되지 않도록 입력 버퍼를 가지고 있으며, 입력 신호의 전압 범위는 0 ~ 5V입니다. 그리고 디지털 입력 단자와 입력 버퍼 사이에 직렬로 22Ω 저항이 연결되어 있습니다.

이번 장에서는 제어기의 디지털 입력 포트를 활용하는 사례들에 대해 소개합니다.

이번 장에서 소개하는 내용을 진행하기 위해서는 다음 준비물이 필요합니다:

- 비상 정지 버튼
- 발 페달 스위치
- 1~10KΩ 저항
- 마이크로 스위치 or 근접 센서, 포토 인터럽터
- 근접 센서를 사용하는 경우 전원 분리를 위한 포토커플러

7.1 비상 정지 버튼의 사용

비상 정지 버튼(Emergency Stop Switch)은 제어기의 오작동이나 위험한 상황에서 모터의 전원을 차단하기 위해 사용합니다. 비상 정지 버튼은 그림 7-1과 같이 버섯형 누름 버튼 스위치 방식으로, 운전자가 위험한 순간에 신속하게 누를 수 있도록 눈에 잘 띄도록 설치해야 합니다.



그림 7-1 Emergency Stop Button

이 절에서는 비상 정지 버튼으로 모터의 전원을 차단하는 것에 대해 설명합니다. 비상 정지 버튼에 추가적으로 다이내믹 브레이크 회로를 사용한다면 비상시 모터를 긴급정지 하는데 도움이 됩니다.

※주의※ 사고의 위험이 있는 곳에서 비상 정지 기능을 사용할 때에는, 정지 신호가 마이크로프로세서나 제어기 등 전자회로를 거치지 않고 반드시 전용 정지신호 배선에 의한 하드와이어 방식으로 구성되어야 합니다.

7.1.1 비상 정지 버튼 배선

그림 7-2에서와 같이 제어기의 디지털 입력 포트에 비상 정지 버튼을 연결합니다. 이 때, 비상 정지 버튼은 누르지 않은 상태에서 회로가 연결된 상황이 되도록 결선합니다. 그리고 풀업 저항을 신호선과 +5V 사이에 연결합니다.

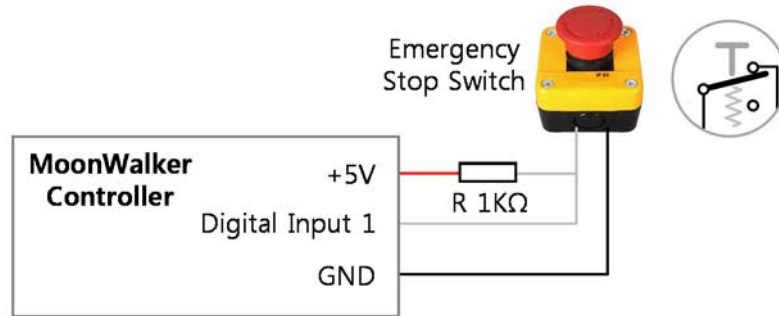


그림 7-2 제어기에 비상 정지 버튼의 연결

만일 비상 정지 신호가 제어기에 입력되면 모든 조건에서 즉시 정지해야 합니다. 비상 정지 후 정상 기능 복귀는 수동으로 복귀시킬 때까지 회로 복귀가 불가능한 구조이어야 합니다. 만일 여러 개의 비상정지 스위치가 설치된 경우, 하나의 비상 스위치 작동 시 비상 정지 기능이 동작해야 하며, 작동된 모든 비상정지 스위치가 복귀되기 전에는 비상 정지 기능이 복구되지 않아야 합니다.

※주의※ 비상 정비 버튼은 긴급한 순간에 동작해야 함으로, 버튼으로 연결 회로가 단절되었을 때 동작하도록 구성해야 합니다. 만일 반대로 회로를 구성한 경우, 버튼의 고장이나 단선으로 인해 긴급한 순간에 비상 정지 기능을 사용하지 못할 경우가 발생할 수 있습니다.

7.1.2 디지털 입력 채널의 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 디지털 입력 채널의 파라미터를 설정합니다. Digital Input Channel 1 그룹에서 Enable 항목을 체크하고 Input channel Mapping 서브 그룹에서 Target을 Motor Channel 1로 선택하고 Action을 Emergency Stop으로 선택합니다.

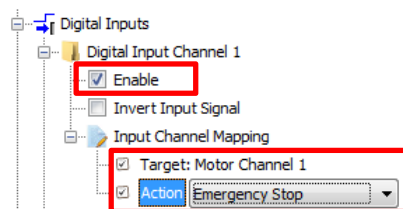


그림 7-3 디지털 입력 채널 설정

설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

7.1.3 비상 정지 버튼의 동작

UI 유틸리티의 IO Monitoring 탭에서 Digital Inputs 그룹의 해당 LED가 비상 정지 버튼을 누르고 땔 때마다 켜지고 꺼지는지 확인합니다.

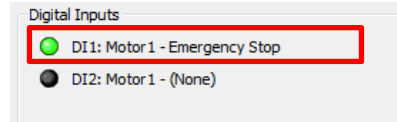


그림 7-4 디지털 입력 채널 모니터링

그리고 Motor Control 탭의 Motor Control Status 그룹에서 Emergency Stop 상태가 비상 정지 버튼의 상태에 연동하는지 확인합니다.

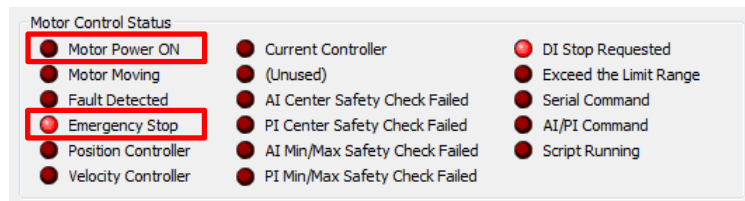


그림 7-5 모터 제어 상태 모니터링

Motor Power ON LED가 켜진 상태에서 비상 정지 버튼이 눌리면 LED는 꺼진 상태가 됩니다. 이 때 비상 정지 버튼을 떼어도(해제 하여도) Motor Power ON LED가 다시 켜지지는 않습니다. 모터가 Power ON 상태로 바뀌려면 [Power ON] 버튼을 눌러야 합니다.

7.2 정지 스위치의 사용

정지 스위치는 모터 운용 중 사용자의 요청에 따라 구동을 일시적으로 멈춰야 할 때 사용합니다.

이 절에서는 사용자가 작업 중 발 페달 스위치(Foot Pedal Switch)를 밟고 모터를 정지하는 방법에 대해 설명합니다.

7.2.1 발 페달 스위치 배선

그림 7-6에서와 같이 제어기의 디지털 입력 포트에 발 페달 스위치를 연결합니다. 이 때, 발 페달 스위치는 밟지 않은 상태에서 회로가 연결된 상황이 되도록 결선합니다. 그리고 풀업 저항을 신호선과 +5V 사이에 연결합니다.

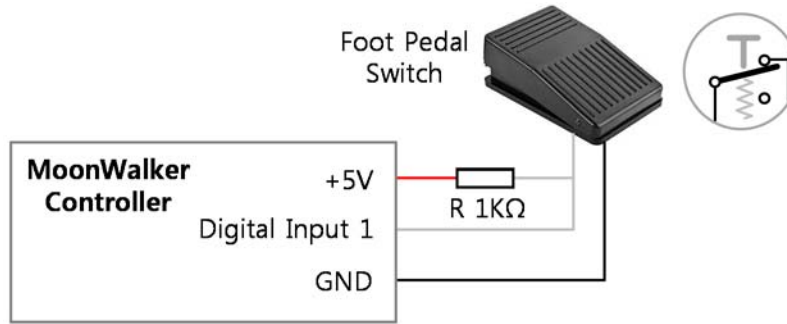


그림 7-6 제어기에 발 페달 스위치의 연결

※주의※ 발 페달 스위치 역시 비상 정비 버튼과 마찬가지로, 스위치 연결 회로가 단절되었을 때 동작하도록 구성해야 합니다. 모터의 정지 기능 역시 중요한 순간에 모터를 정지하기 위해 사용될 수 있습니다.

7.2.2 디지털 입력 채널의 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 디지털 입력 채널의 파라미터를 설정합니다. Digital Input Channel 1 그룹에서 Enable 항목을 체크하고 Input channel Mapping 서브 그룹에서 Target을 Motor Channel 1으로 선택하고 Action을 Quick Stop 또는 Declaration Stop 중 하나로 선택합니다.

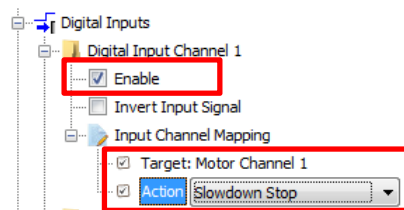


그림 7-7 디지털 입력 채널 설정

설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

7.2.3 발 페달 스위치의 동작

Motor Control UI 유틸리티의 IO Monitoring 탭에서 Digital Inputs 그룹의 해당 LED가 발 페달 스위치를 누르고 뗄 때마다 켜지고 꺼지는지 확인합니다.

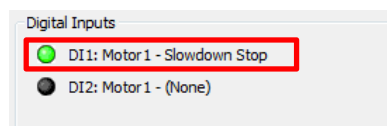


그림 7-8 디지털 입력 채널 모니터링

그리고 Motor Control 탭의 Motor Control Status 그룹에서 DI Stop Requested 상태가 발 페달 스위치의 상태에 연동하는지 확인합니다.

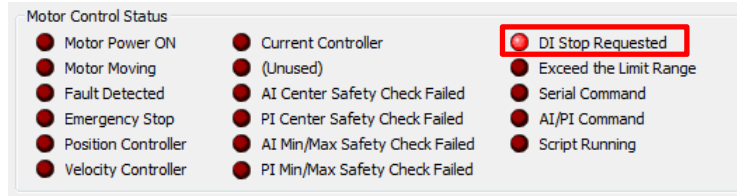


그림 7-9 모터 제어 상태 모니터링

현재 구동 중인 모터가 디지털 입력의 정지 요청에 의해 멈춘 경우, 정지 요청이 해제되어도 이전 명령이 재개되지는 않습니다. 새로운 구동 명령이 입력되어야 모터는 구동 됩니다.

그리고 디지털 입력의 정지 요청이 1로 지속되는 상태에서 모터를 움직이는 명령이 내려지더라도 명령은 수행되지 않습니다. 디지털 입력 값이 0이 될 때까지 모터는 움직일 수 없게 됩니다.

7.3 홈 센서의 사용

홈 센서는 모터의 홈 위치를 설정하기 위해 사용됩니다. 홈 위치 설정과정은 다음과 같습니다: 1) 홈 위치를 찾기 위해 홈 센서가 있는 위치로 모터에 속도 명령을 내립니다. 2) 홈 센서가 감지되면 모터의 위치에 홈 위치(Home Position)를 복사하여 기준 위치를 설정합니다. 3) 이를 확인하고 모터를 정지합니다.

홈 센서를 사용할 때는 기구부를 그림 7-10과 같이 홈 위치를 기준으로 센서 값이 1인 영역(그림에서 **Sensing Range**로 표시)과 0인 영역(그림에서 **Unsensing Range**로 표시)을 정확히 구분하도록 설계하는 것이 중요합니다. 그래서 홈 센서 값이 1일 때와 0일 때 모터가 어느 영역에 있는지를 알고 구동 방향(Forward or Reverse)을 결정할 수 있어야 합니다.

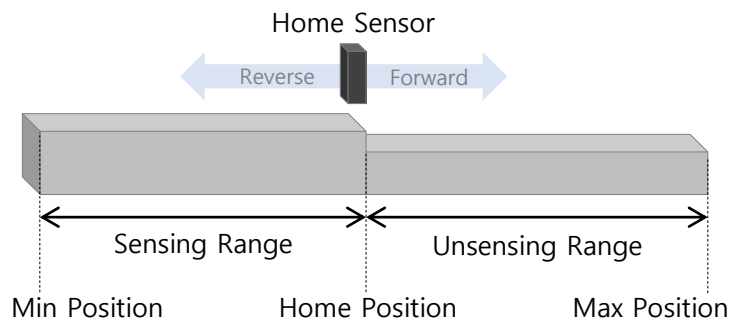


그림 7-10 홈 센서의 감지 위치

그림 7-10에서와 같이 기준 위치를 찾기 위해 홈 센서를 움직여야 한다면, 센서의 값이 0일 때 (홈 센서는 Unsensing Range에 있음) Reverse 방향으로 이동해야 하고 1일 때(홈 센서는 Sensing Range에 있음) Forward 방향으로 이동해야 합니다.

홈 위치는 센서 값이 0에서 1로 바뀌는 순간에 설정됩니다. 따라서, 만일 현재 홈 센서가 **Sensing Range**에 있다면 먼저 **Unsensing Range**로 이동해야 합니다. 그리고 홈 위치 설정과정을 진행해야 합니다.

7.3.1 홈 센서 배선

그림 7-11에서와 같이 제어기의 디지털 입력 포트에 홈 센서로 사용되는 마이크로 스위치를 연결합니다. 이 때, 마이크로 스위치는 누르지 않은 상태에서 켜진 상황이 되도록 연결합니다. 그리고 풀업 저항을 신호선과 +5V 사이에 연결합니다.

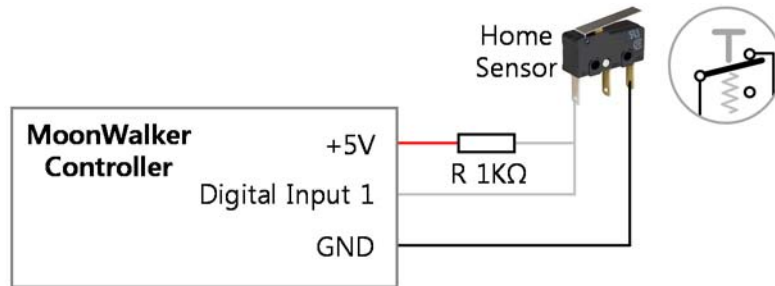


그림 7-11 제어기에 마이크로 스위치의 연결

홈 센서로 접촉식 마이크로 스위치 외에도 비접촉식 근접 센서나 포토 인터럽터가 사용될 수도 있습니다. 근접 센서나 포토 인터럽터를 사용하기 위해서는 추가적인 회로 구성이 필요합니다. "7.6 근접 센서 및 포토 인터럽터의 사용" 절을 참고하기 바랍니다.

7.3.2 디지털 입력 채널의 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 디지털 입력 채널의 파라미터를 설정합니다. 먼저 Position Sensors 그룹에서 Home Position 항목에 홈 위치를 설정합니다. 홈 위치는 홈 센서가 작동하였을 때 모터의 위치로 설정되는 기준 값입니다.

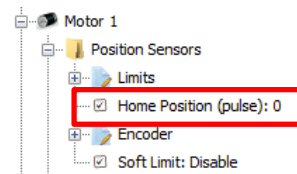


그림 7-12 홈 위치의 설정

그리고 Digital Input Channel 1 그룹에서 Enable 항목을 체크하고 Input channel Mapping 서브 그룹에서 Target을 Motor Channel 1으로 선택하고 Action을 Load Home Counter로 선택합니다.

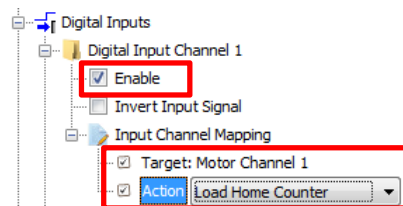


그림 7-13 디지털 입력 채널 설정

설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을

제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

7.3.3 홈 센서의 동작

Motor Control UI 유틸리티의 IO Monitoring 탭에서 Digital Inputs 그룹의 해당 LED가 마이크로 스위치를 손으로 누르고 뿔 때마다 켜지고 꺼지는지 확인합니다.

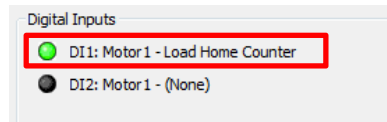


그림 7-14 디지털 입력 채널 모니터링

그리고 Motor Control 탭에서 모터가 홈 센서가 있는 위치로 이동하도록 약간의 전압 출력을 내 보내거나 느린 속도 명령(그림 7-15에서는 100RPM)을 내립니다. 그러면 그림 7-15에서와 같이 모터는 홈 위치로 저속으로 이동하고 홈 센서가 감지되면 **'Home Position'** 오브젝트의 값을 **'Position'** 오브젝트(모터의 위치, 엔코더 카운터의 값)에 복사합니다. 홈 위치 설정이 끝나면 [Quick Stop] 버튼을 눌러 모터를 중지시킵니다.

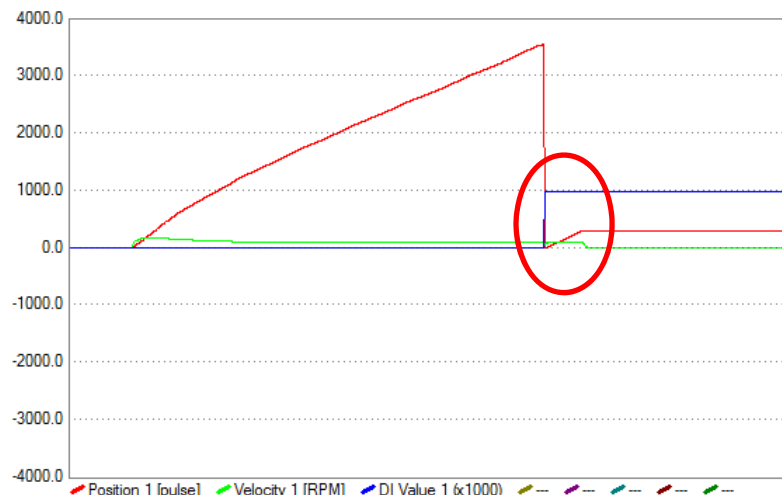


그림 7-15 홈 센서의 감지와 홈위치 설정

그림 7-15의 그래프는 홈 센서가 감지되었을 때 모터의 위치가 **'Home Position'** 오브젝트의 값 (여기서는 0)으로 설정되는 것을 볼 수 있습니다.

※ 홈 센서로 모터 제어기의 원점 복귀 기능을 쉽게 사용하기 위해서는 상기 과정을 스크립트 프로그래밍으로 구현해야 합니다. 이에 대한 상세한 내용은 "9.2 원점 복귀" 절을 참고하기 바랍니다.

※주의※ 홈 위치 설정 과정에서 주의를 기울이지 않으면 모터와 주변 기계 장치간 충돌이 발생할 수 있습니다.

7.4 리미트 센서의 사용

'Use Soft Limit' 오브젝트가 1로 설정된 경우, 모터가 구동 중에 사용자가 설정한 '**Min Position**'과 '**Max Position**' 오브젝트의 범위를 벗어나면 모터와 주변 기계 장치간 충돌을 방지하기 위해 모터를 정지시키게 됩니다. 그리고 또 다른 안전을 위해 그림 7-16와 같이 '**Min Position**'과 '**Max Position**' 범위를 조금 벗어난 곳에 정방향/역방향 리미트 센서를 연결하고 모터가 리미트 센서의 범위를 벗어나지 못하도록 제어 합니다.

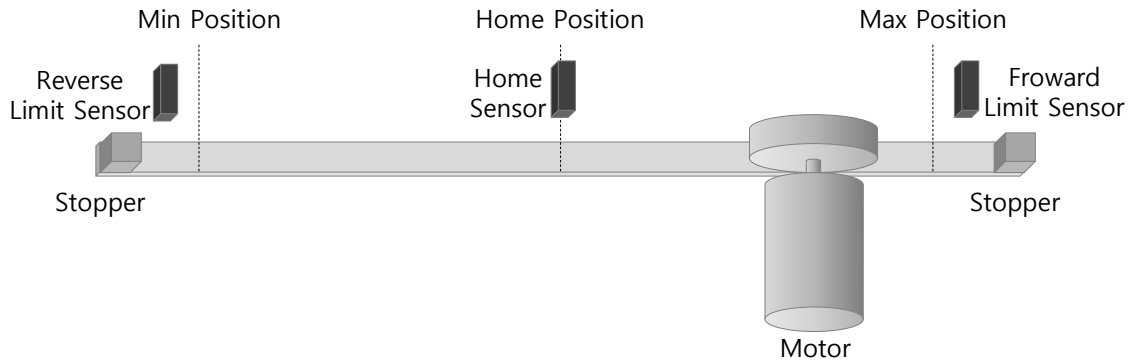


그림 7-16 리미트 센서의 사용

이 절에서는 정방향과 역방향 리미트 센서를 사용하여 모터의 구동 범위를 제한하는 방법에 대해 설명합니다.

7.4.1 리미트 센서 배선

그림 7-17에서와 같이 제어기의 디지털 입력 포트에 리미트 센서로 사용할 두 개의 마이크로 스위치를 연결합니다. 이 때, 마이크로 스위치는 눌리지 않은 상태에서 켜진 상황이 되도록 연결합니다. 그리고 풀업 저항을 신호선과 +5V 사이에 연결합니다.

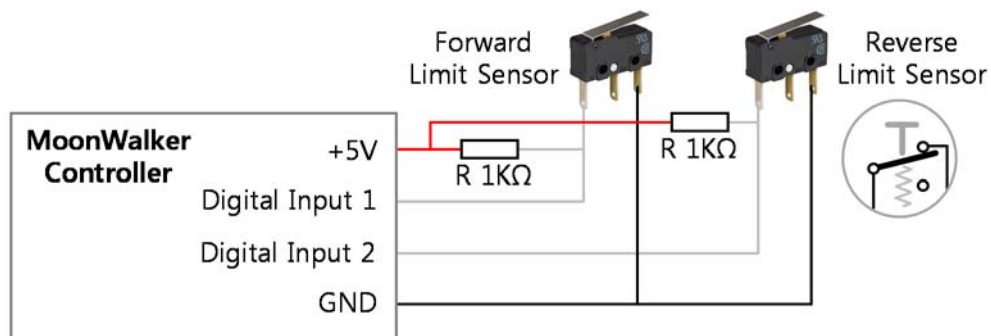


그림 7-17 제어기에 마이크로 스위치의 연결

리미트 센서로 접촉식 마이크로 스위치 외에도 비접촉식 근접 센서나 포토 인터럽터가 사용될 수도 있습니다. 근접 센서나 포토 인터럽터를 사용하기 위해서는 추가적인 회로 구성이 필요합니다. "7.6 근접 센서 및 포토 인터럽터의 사용" 절을 참고하기 바랍니다.

7.4.2 디지털 입력 채널의 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 디지털 입력 채널의 파라미터를 설정합니다. 그림 7-18과 같이 Digital Input Channel 1과 Digital Input Channel 2 그룹에서 Enable 항목을 체크하고 Input channel Mapping 서브 그룹에서 Target을 Motor Channel 1으로 선택하고 Action을 Forward Limit Switch와 Reverse Limit Switch로 선택합니다.

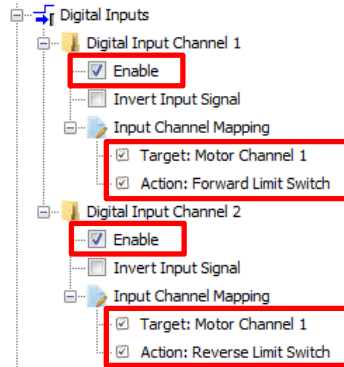


그림 7-18 디지털 입력 채널 설정

설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

7.4.3 리미트 센서의 동작

Motor Control UI 유틸리티의 IO Monitoring 탭에서 Digital Inputs 그룹의 해당 LED가 마이크로 스위치를 누르고 뿔 때마다 켜지고 꺼지는지 확인합니다.

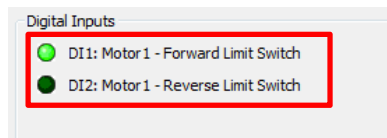


그림 7-19 디지털 입력 채널 모니터링

그리고 Motor Control 탭의 Motor Control Status 그룹에서 Exceed the Limit Range 상태가 마이크로 스위치의 상태에 연동하는지 확인합니다.

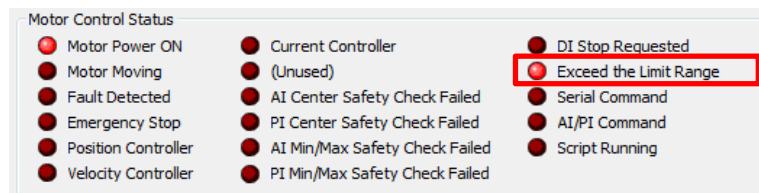


그림 7-20 모터 제어 상태

만일 모터에 위치, 속도, 전류, 전압 명령을 내리고 있다면, 정방향 리미트 센서가 켜질 경우 이 스위치가 꺼지기 전까지 모터는 정방향으로 구동할 수 없으며 역방향으로만 구동이 가능합니다. 반대로 역방향 리미트 센서가 켜질 경우 이 스위치가 꺼지기 전까지 모터는 역방향으로 구동할

수 없으며 정방향으로만 구동이 가능합니다.

※주의※ 리미트 센서의 테스트 과정에서 주의를 기울이지 않으면 모터와 주변 기계 장치간 충돌이 발생할 수 있습니다.

7.5 홈 센서와 리미트 센서의 사용

보유한 제어기에서 제공하는 디지털 입력 채널의 수가 2개 이고 연결해야 할 센서가 3개 이상일 때는 남은 아날로그 입력 채널에 디지털 신호를 출력하는 센서를 연결할 수 있습니다.

이 절은 디지털 신호를 출력하는 센서를 아날로그 입력 채널에서 사용하는 방법을 설명하는데 목적이 있습니다.

7.5.1 홈 센서와 리미트 센서 배선

그림 7-21에서와 같이 제어기의 디지털 입력 채널 1과 2에 각각 정방향과 역방향 리미트 센서로 사용되는 마이크로 스위치를 연결합니다. 그리고 아날로그 입력 채널 1에 홈 센서로 사용되는 마이크로 스위치를 연결합니다.

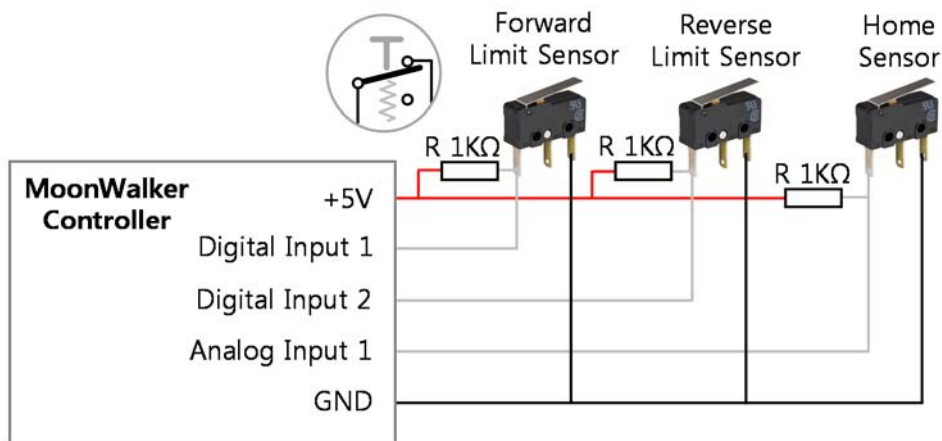


그림 7-21 제어기에 리미트센서와 홈 센서 연결

리미트 센서와 홈 센서로 접촉식 마이크로 스위치 외에도 비접촉식 근접 센서나 포토 인터럽터가 사용될 수도 있습니다. 근접 센서나 포토 인터럽터를 사용하기 위해서는 추가적인 회로 구성이 필요합니다. "7.6 근접 센서 및 포토 인터럽터의 사용" 절을 참고하기 바랍니다.

7.5.2 디지털/아날로그 입력 채널의 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 디지털 입력 채널과 아날로그 입력 채널의 파라미터를 설정합니다.

그림 7-23에서와 같이 Digital Input Channel 1과 Digital Input Channel 2 그룹에서 Enable 항목을 체크하고 Input channel Mapping 서브 그룹에서 Target을 Motor Channel 1으로 선택하고 Action을 Forward Limit Switch와 Reverse Limit Switch로 선택합니다.

그리고 Analog Input Channel 1 그룹에서 Enable 항목을 체크하고 Input Channel Mapping 서브 그룹에서 Target을 Motor Channel 1으로 선택하고 Action을 DI: Load Home Counter로 선택합니다.

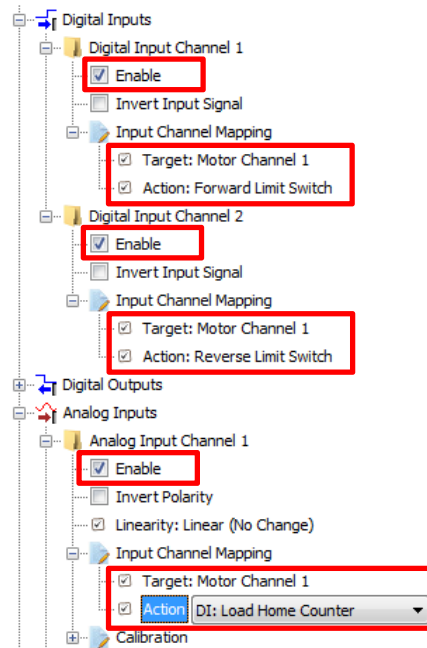


그림 7-22 디지털/아날로그 입력 채널 설정

Analog Input Channel 1에서 캘리브레이션 파라미터를 설정해야 하는데, Calibration 대화상자를 사용하지 않고, 그림 7-23에서와 같이 직접 입력 합니다: Input Min = 0, Input Center = 2047, Input Max = 4095, Input Deadband = 0.

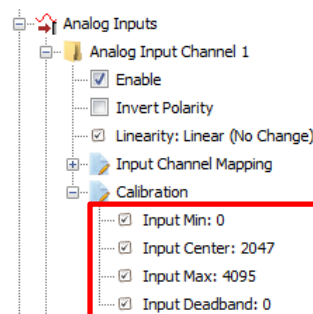


그림 7-23 캘리브레이션 파라미터 입력

설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

7.5.3 홈 센서와 리미트 센서의 동작

Motor Control UI 유틸리티의 IO Monitoring 탭에서 Digital Inputs 그룹의 해당 LED가 마이크로 스위치를 누르고 뿔 때마다 켜지고 꺼지는지 확인합니다.

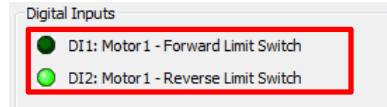


그림 7-24 디지털 입력 채널 모니터링

그리고 Analog Inputs 그룹의 해당 입력이 활성화되고 마이크로 스위치를 누르고 뿔 때마다 올바른 값이 입력되는지 확인합니다.

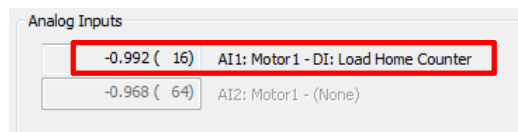


그림 7-25 아날로그 입력 채널 모니터링

아날로그 입력 채널의 값을 디지털로 변환하는 조건은 다음과 같습니다: **아날로그 값이 0보다 같거나 작을 때는 디지털 값 0으로 변환되고 아날로그 값이 0보다 클 때는 디지털 값 1로 변환됩니다.**

Motor Control 탭의 Motor Control Status 그룹에서는 Exceed the Limit Range 상태가 마이크로 스위치의 상태에 연동하는지 확인합니다.

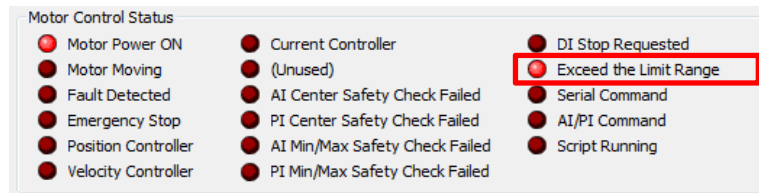


그림 7-26 모터 제어 상태

그리고 모터를 구동하여 '**Position**' 오브젝트의 값을 '**Home Position**'이 아닌 위치로 옮겨 두고, 홈 센서(마이크로 스위치)를 손으로 눌렀을 때 '**Position**' 오브젝트에 '**Home Position**'이 설정되는지 확인합니다.

이후 테스트 과정은 "7.3.3 홈 센서의 동작"과 "7.4.3 리미트 센서의 동작"의 내용을 참고하여 진행합니다.

7.6 근접 센서 및 포토 인터럽터의 사용

리미트 센서로 접촉식 마이크로 스위치 외에도 비접촉식 근접 센서나 포토 인터럽터가 사용될 수도 있습니다.

근접 센서:

근접 센서는 검출 물체의 접근을 비접촉으로 검출할 수 있는 센서로 고주파를 이용하는 고주파형 발진형, 자석을 이용한 자기형 그리고 전기장을 이용하는 정전용량형으로 크게 나눌 수 있습니다.

근접 센서는 비접촉으로 검출이 가능하고 수명이 길며, 좋지 않은 환경에서도 사용이 가능합니다. 그리고 반복 정밀도가 매우 높아 위치결정용 센서로 가장 적합하며, 빠른 주파수응답을 가지고 있습니다. 그러나 금속 이외에 검출이 불가능하고 검출거리가 매우 짧은 단점이 있습니다.



그림 7-27 근접 센서

포토 인터럽터:

포토 인터럽터는 크게 투과형과 반사형으로 구분할 수 있습니다. 투과형 포토 인터럽터는 발광소자와 수광소자를 일정한 간격을 두고 마주보게 놓아 그 사이의 물체의 유무와 위치를 감지하는 방식입니다. 그리고 반사형 포토 인터럽터라고 불리는 것으로 발광소자와 수광소자를 평면상 혹은 각도를 두고 발광소자로부터 나온 광을 물체에 반사시켜 그 반사광을 수광소자에서 검출하는 방식입니다.

포토 인터럽터는 주호 회전체의 회전검출이나 물체의 위치검출에 많이 사용되고 있습니다.

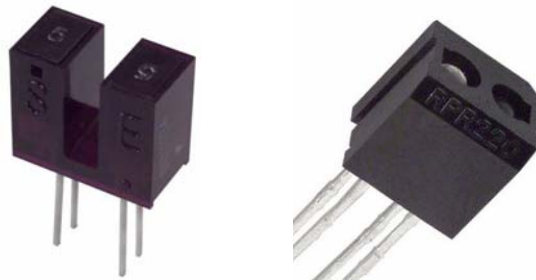


그림 7-28 투과형과 반사형 포토 인터럽터

본 절은 리미트 센서 대신 근접 센서를 이용한 방법에 대해 설명합니다.

7.6.1 근접 센서 배선

그림 7-29와 같이 제어기의 디지털 입력 포트에 두 개의 근접 센서를 연결합니다. 이 때, 근접 센서 전원단과 제어기의 5V 전원단을 분리하기 위해 포토커플러를 사용해서 디지털 입력 포트와 근접센서 신호선을 연결합니다.

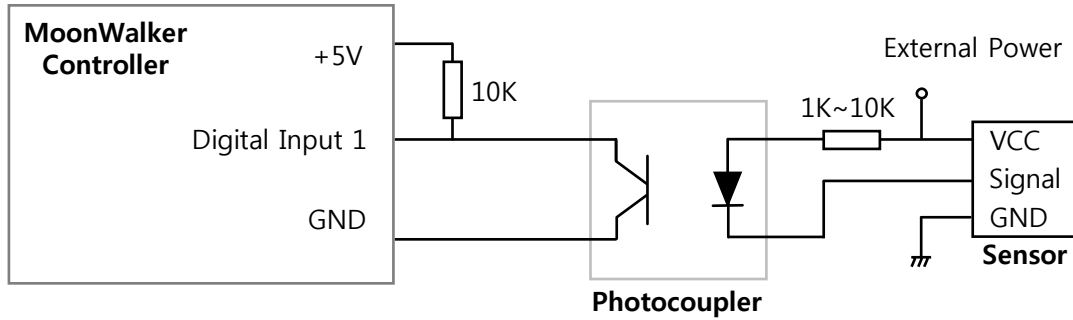


그림 7-29 제어기에 근접 센서의 연결

그리고 근접 센서의 전원은 별도의 외부 전원에 연결하시기 바랍니다.

※ 일반적으로 많이 사용하는 근접 센서의 전원은 12~24V 입니다.

7.6.2 포토 인터럽터 배선

수, 발광부가 한 번에 구성된 포토 인터럽터는 사용자의 환경에 따라 다양하게 외부 회로를 구성할 수 있습니다.

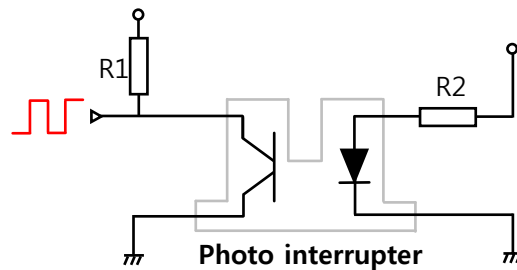


그림 7-30 포토 인터럽터의 대표적인 회로

그림 7-30은 포토 인터럽터의 대표적인 회로이며, 왼쪽이 수광부, 오른쪽이 발광부입니다.

포토 인터럽터는 발광부에서 발산한 빛이 트랜지스터를 Turn-on시키면 R1 밑에 연결된 신호 포트에 Low 신호가 출력됩니다. 그리고 중간에 차폐막(ex 막대기)이 들어오게 되면 빛이 차단되어 트랜지스터가 Turn-off되며 신호 포트에 High 신호가 출력됩니다.

그림 7-31은 그림 7-30을 바탕으로 제어기에 포토 인터럽터를 연결하는 방법입니다.

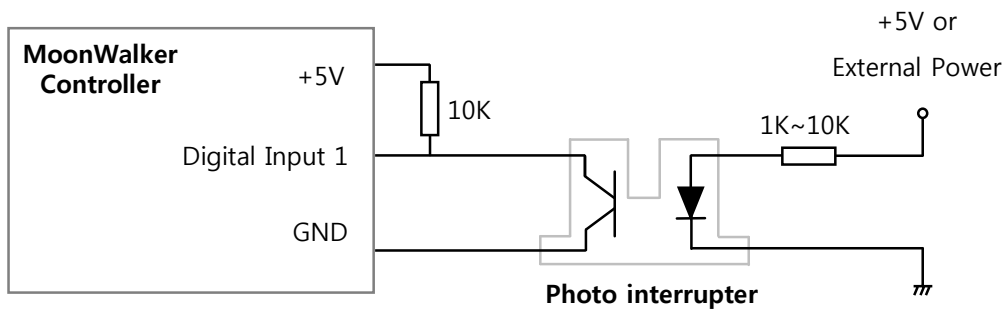


그림 7-31 제어기에 포토 인터럽터의 연결 1

만약 포토 인터럽트의 전원이 5V인 경우 제어기에 공급하는 5V 전원에 연결해서 사용하고 그 외에는 다른 외부 전원을 사용하시기 바랍니다.

또한, 포토 인터럽트는 그림 7-32와 같은 회로 구성을 통해 또 하나의 트랜지스터를 구동하는 신호로 사용할 수 있습니다. 이때 저항 R1, R2, R3는 구성 회로에 맞게 저항 값을 결정하면 됩니다. 그림 7-32 제어기에 포토 인터럽터의 연결 2

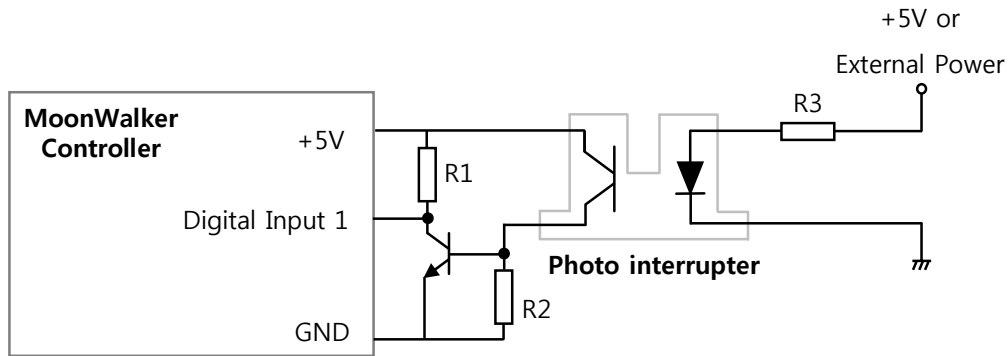


그림 7-32 제어기에 포토 인터럽터의 연결 2

이 외에 포토 인터럽터를 가지고 검출된 신호를 트리거 신호로 활용할 수 있습니다. IC I/O 포트에 입력해서 해석할 수 있습니다. 그뿐만 아니라 만약 사용자가 큰 신호를 드라이빙해야 할 경우 트랜지스터 뒷단에 릴레이를 구동하시면 작은 포토 신호를 이용해서 상용전원을 사용하는 제품도 제어가 가능합니다.

※ 포토 인터럽터는 제품과 제조사에 따라 핀 구조와 핀 개수가 다를 수도 있습니다. 그러나 포토 커플러와 같은 광전자 소자 중 하나이기 때문에 위에 내용은 참조해서 포토 인터럽터 제품에 맞는 주변 회로를 구현하시기 바랍니다.

8 디지털 출력 포트의 사용

제어기의 디지털 출력 포트는 모터의 구동 상태(Power ON/OFF 상태, 회전 방향, 과전압 상태) 및 FET의 과열 상태를 외부에 알리는데 사용합니다.

디지털 출력 포트를 통해 브레이크, LED, 부저, 경고등, 팬 등 다양한 장치들을 구동 가능합니다. 그림 8-1과 같이 디지털 출력 포트 말단에는 마이크로프로세서의 출력 신호에 비해 높은 전력의 부하를 구동하기 위해 MOSFET가 사용됩니다.

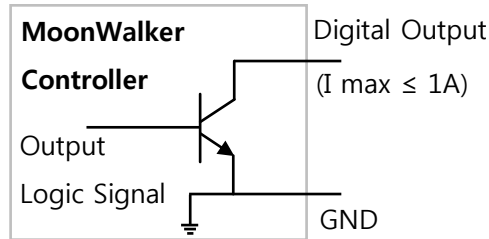


그림 8-1 디지털 출력 포트의 FET

다음은 디지털 출력 포트를 구동하는 MOSFET 소자 정보입니다:

- Continuous drain source voltage : 60V
- On-state resistance : 500mΩ
- Nominal load current (VIN = 5V) : 1.3A

MOSFET는 60V에 1.3A에서 구동할 수 있는 Open Drain MOSFET 출력으로, 활성화된 경우 출력은 GND로 연결됩니다. 따라서 부하의 한쪽 끝은 출력에 연결하고 다른 한쪽 끝은 24V 배터리 같은 양의 전압 출력에 연결해야 합니다.

※ ZXMS6004FF MOSFET 데이터시트에는 60V, 1.3A까지 구동할 수 있다고 기재되어 있지만, 사용자와 제어기의 안전을 위해 50V, 1A까지 사용하시길 권장합니다.

이 장에서는 디지털 출력 포트를 활용하는 사례들에 대해 소개합니다.

이 장에서 소개하는 내용을 진행하기 위해서는 다음 준비물이 필요합니다:

- 브레이크가 장착된 DC 모터
- 냉각 팬
- 저항: 100Ω, 1K ~ 10KΩ
- 제동 저항: 세라믹 저항 (5Ω)
- 50V, 10A 이하의 릴레이
- 20A 이상의 N 또는 P Channel MOSFET
- 10A 이상의 다이오드

8.1 모터 브레이크 사용

엘리베이터 혹은 산업용 장비에서 모터의 전원이 갑자기 끊어지거나 중단되었을 때 모터 및 연결된 장치 그리고 주변의 안전을 위해 모터 브레이크를 작동해야 합니다.

이 절에서는 제어기의 정전 상황이나 모터 Power OFF 상태에서 모터 브레이크를 사용하는 방법에 대해 설명합니다.

8.1.1 모터 브레이크 배선

브레이크가 연결된 서보 모터를 사용하여 그림 8-2와 같이 제어기와 모터, 모터 브레이크를 배선합니다.

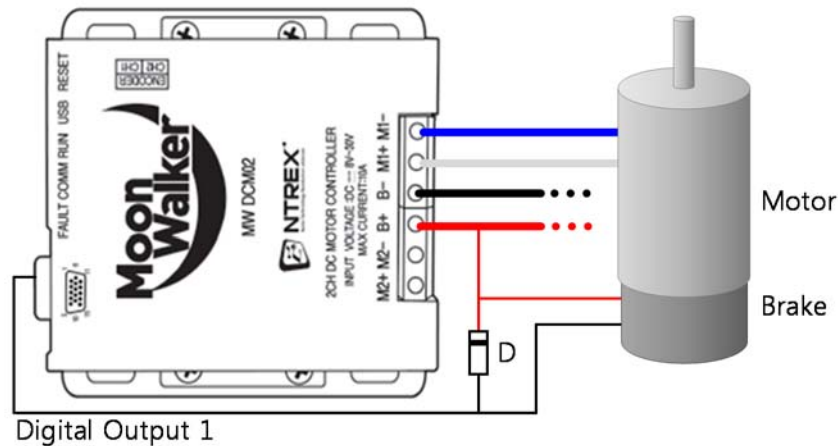


그림 8-2 제어기에 브레이크 연결

디지털 출력 포트에 내장된 FET가 브레이크를 직접 구동할 수 있는 용량일 때는 부가적인 회로를 구성하지 않고 디지털 출력 포트에 직접 연결합니다. 그리고 브레이크의 구동 전압 범위가 제어기에 공급되는 전압을 포함할 때는 제어기의 전원을 사용하도록 합니다.

모터 브레이크는 내부에 인덕턴스 성분을 가지는 유도성 부하(Inductive loads)입니다. 유도성 부하가 스위칭 할 때 유기된 고전압 스파크에 의해 FET가 파괴될 수 있으니 보호 다이오드를 사용하기 바랍니다.

※주의※ 배터리 음극 연결 단자(B-)와 I/O의 GND 단자를 서로 연결하지 않도록 주의합니다.

만일 제어기의 디지털 출력이 브레이크를 구동하기에 충분하지 않을 때는 다음과 같이 간단한 릴레이 구동 회로를 사용합니다.

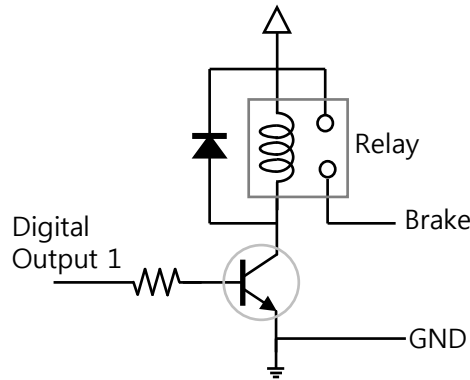


그림 8-3 브레이크 구동을 위한 릴레이 사용

8.1.2 디지털 출력 채널의 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 디지털 출력 채널의 파라미터를 설정합니다. Digital Output Channel 1 그룹에서 Enable 항목을 체크하고 Output channel Mapping 서브 그룹에서 Source를 Motor Channel 1으로 선택하고 Status를 Motor Power ON (Brake Release)로 선택합니다.

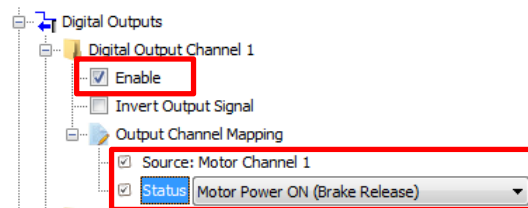


그림 8-4 디지털 출력 채널 설정

그리고 Digital Output ON Conditions 그룹에서 Brake ON Delay 항목을 설정합니다. 이 항목은 모터가 Power OFF 상태가 되었을 때 브레이크를 ON 하기까지의 지연 시간을 조절합니다.

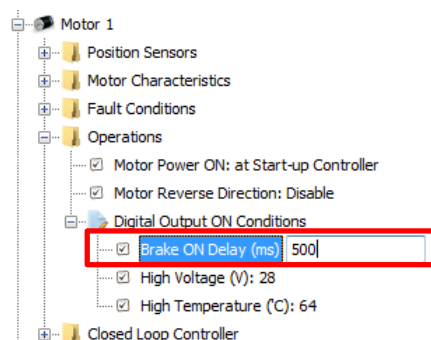


그림 8-5 Brake ON Delay 설정

상기와 같이 설정할 경우 모터 Power ON시 브레이크는 즉시 OFF가 되고 모터 Power OFF시 500ms 후 브레이크는 ON이 됩니다.

설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

8.1.3 모터 브레이크 작동 확인

Motor Control UI 유틸리티의 IO Monitoring 탭에서 Digital Outputs 그룹의 해당 LED가 모터를 Power ON/OFF 할 때마다 켜지고 꺼지는지 확인합니다.

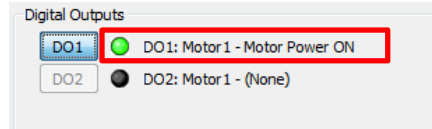


그림 8-6 디지털 출력 채널 상태

그리고 모터의 브레이크가 켜지고 꺼지는지 확인합니다. 감속기가 연결되지 않은 소형 모터의 경우 브레이크가 꺼지면 손으로 약간의 힘을 주어 모터 축을 돌릴 수 없고 브레이크가 켜지면 손으로 돌릴 수 있습니다.

※주의※ 모터 브레이크를 테스트할 때, 브레이크가 잠긴 상태에서 모터를 구동하면 브레이크가 파손될 수 있습니다.

※주의※ 모터를 정지하기 위해 모터 브레이크를 사용하지 않아야 합니다. 모터 정지에 반복적으로 사용하면 브레이크가 파손될 수 있습니다.

8.2 다이نام릭 브레이크

모터가 고속 회전 중 제어가 전원이 끊어지거나 폴트가 발생하여 모터에 전력 공급이 중단되었을 때(Motor Power OFF) 모터는 프리런(Free Run)상태가 되어 정지까지 긴 시간이 필요합니다. 결국에는 이동량이 커져 다른 기구물과 충돌할 수도 있습니다.

그림 8-7은 간단한 다이نام릭 브레이크(Dynamic Brake) 회로를 보여줍니다. 스위치가 ON 상태일 때는 배터리가 모터에 전력을 공급하여 모터를 회전하게 됩니다. 하지만 스위치가 OFF 상태가 되면 모터의 단자간을 저항으로 단락(short-circuit) 시킵니다. 그리고 모터의 회전 에너지를 저항에서 열로 소비시킴으로 모터를 급속히 정지시킵니다.

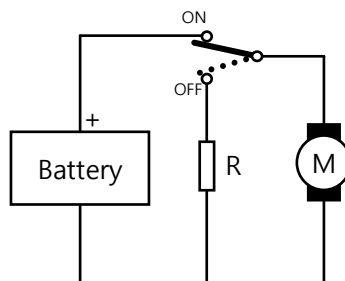


그림 8-7 다이نام릭 브레이크 기본 회로

다이نام릭 브레이크는 다른 기계적 브레이크와 달리 모터가 회전하고 있을 때만 기능합니다. **모터가 정지하고 있을 때는 브레이크 능력이 현저히 떨어집니다.**

※주의※ 다이نام믹 브레이크를 엘리베이터와 같은 상하 운동 기구물의 제동과 유지에 사용하면 안 됩니다. 상하 운동 기구물의 경우 기계적 브레이크와 병행하여 사용해야 합니다.

8.2.1 다이نام믹 브레이크 배선

다이نام믹 브레이크를 사용하기 위해 그림 8-8과 같이 릴레이(Relay)와 부하 저항(Shunt Load Resistor)을 사용하여 설계합니다.

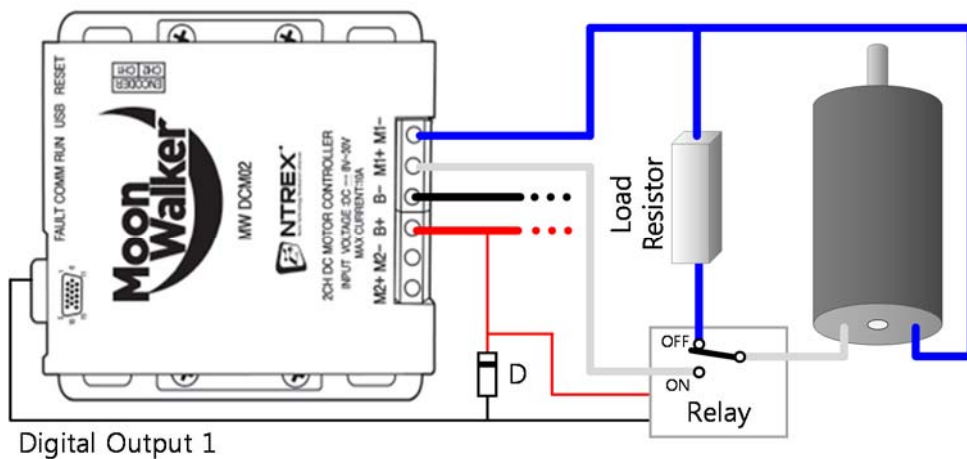


그림 8-8 다이نام믹 브레이크 회로 구성

디지털 출력 포트에 내장된 FET는 소형 릴레이(Relay)를 직접 구동할 수 있는 용량이기 때문에 부가적인 회로를 구성하지 않고 릴레이를 디지털 출력 포트에 직접 연결합니다. 이 때 릴레이는 인덕턴스 성분을 가지는 소자이기 때문에 소자 양단에 다이오드를 역방향으로 삽입합니다.

모터가 Power ON 되어 디지털 출력이 켜지면, 릴레이는 ON 상태가 되어 모터와 제어기간 연결 회로가 만들어집니다. 하지만 디지털 출력이 꺼지거나 제어기 및 릴레이에 전원이 공급되지 않으면 OFF 상태가 되어 모터의 단자간을 부하 저항으로 단락 시키게 됩니다.

부하 저항 값이 낮을수록 다이نام믹 브레이크의 성능은 좋아집니다. 하지만 저항에 흐르는 전류(I)는 저항 값(R)이 낮아질수록 반비례하여 증가합니다($I=V/R$). 또한 저항에서 소모되는 전력(P)도 저항 값이 낮아질 수로 증가합니다($P=V^2/R$). 결국 저항 값이 낮아지면 많은 전력을 소모하게 됩니다. 저항에서 소모되는 전력은 열 에너지로 전환되기 때문에, 이 열 에너지가 저항의 온도를 상승시키게 되는데, 저항의 허용 온도를 초과하면 저항이 타버리게 됩니다.

부하 저항의 용량은 다음과 같이 계산하면 됩니다: 배터리에서 10V의 전압을 가해 모터를 회전하면, 모터에서는 10V의 역기전력이 발생합니다. 이때 스위치를 OFF 하면 부하 저항(여기서는 5Ω이라 가정할 때)에는 2A의 전류가 흐릅니다($I = V/R$). 그러면 부하 저항의 소비 전력은 20W가 됩니다($P=VI$). 부하 저항을 선정할 때는 최대 허용전력이 소비 전력보다 큰 저항을 선정하면 됩니다.

※주의※ 다이نام믹 브레이크 작동 시, 부하 저항에서 고열이 발생하고 화재의 위험이 있습니다.

8.2.2 디지털 출력 채널의 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 디지털 출력 채널의 파라미터를 설정합니다.

그림 8-9에서와 같이 Digital Output Channel 1 그룹에서 Enable 항목을 체크합니다. 그리고 Output channel Mapping 서브 그룹에서 Source를 Motor Channel 1으로 선택하고 Status를 Motor Power ON(Brake Release)로 선택합니다.

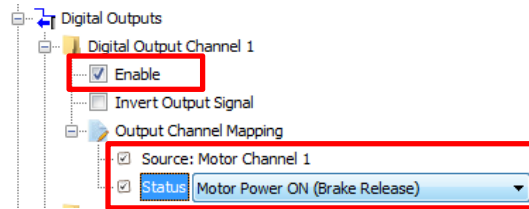


그림 8-9 디지털 출력 채널 설정

그림 8-10에서와 같이 Digital Output ON Conditions 그룹에서 Brake ON Delay 항목을 설정합니다. 이 항목은 모터가 Power OFF 되었을 때 브레이크를 ON 하기까지의 지연 시간을 의미합니다.

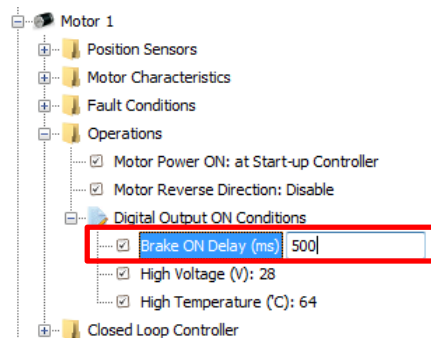


그림 8-10 다이내믹 브레이크 옵션 설정

설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

상기와 같이 설정할 경우 모터 Power ON시 다이내믹 브레이크는 즉시 OFF가 되고 모터 Power OFF시 500ms 후 다이내믹 브레이크는 ON이 됩니다.

8.2.3 다이내믹 브레이크 작동 확인

Motor Control UI 유틸리티의 IO Monitoring 탭에서 Digital Outputs 그룹의 해당 LED가 모터를 Power ON/OFF 할 때마다 켜지고 꺼지는지 확인합니다.

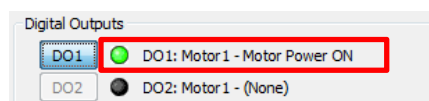


그림 8-11 디지털 출력 채널 상태

그리고 이에 연동하여 릴레이가 켜지고 꺼지는지 확인합니다. 작동 실험을 위해 모터를 고속으로 회전 시킵니다. 그리고 모터를 Power OFF 합니다. 그러면 500ms 후에 다이내믹 브레이크가 작동하여 모터는 빠르게 정지합니다.

그림 8-12의 왼쪽 그래프는 다이내믹 브레이크가 작동하지 않았을 때이고, 오른쪽 그래프는 작동했을 때의 그래프입니다. 다이내믹 브레이크가 작동하면 모터의 정지 시간이 훨씬 짧아지는 것을 볼 수 있습니다 (그림에서 적색 원은 모터 Power OFF 시점을 표시).

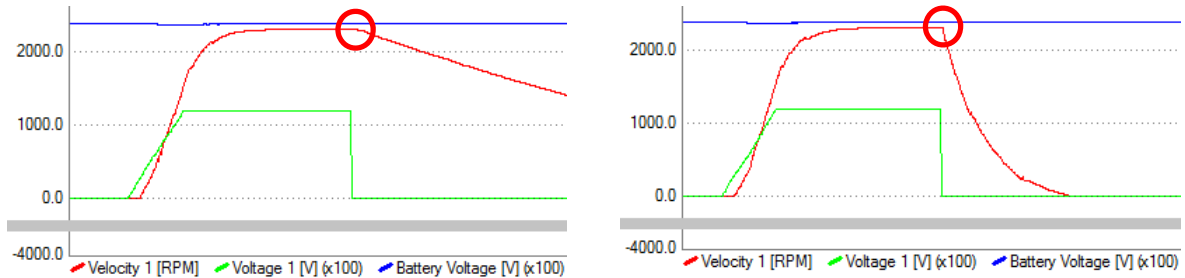


그림 8-12 모터의 속도 그래프

※주의※ 다이내믹 브레이크는 비상시에 회전을 정지시키는 보조 기능입니다. 일반 운전시에 모터를 정지하는 용도로 사용하지 않아야 합니다.

8.3 회생 브레이크

속도 명령보다 모터의 실제 회전속도가 높을 때(감속시, 상하축의 하강시 등)는 FET의 스위칭에 의해 모터에서 발생하는 역기전력을 증폭하여 전원의 전압보다 높은 전압을 생성하게 됩니다. 이때 전원으로 파워서플라이가 사용되는 경우 회생전류를 수용하지 못하여 전원의 전압은 높아지고 모터는 감속이 제대로 이루어지지 않게 됩니다.

상기와 같은 경우, 회생 브레이크(Regenerative Brake)가 작동하여 모터 및 부하가 가지는 회전운동에너지를 회생저항으로 소비함으로써 모터는 제동력을 얻게 됩니다.

8.3.1 회생 브레이크 배선

회생 브레이크를 사용하기 위해 그림 8-13과 같이 브레이크 저항(Braking Resistor)과 FET로 구성되는 회로를 설계합니다.

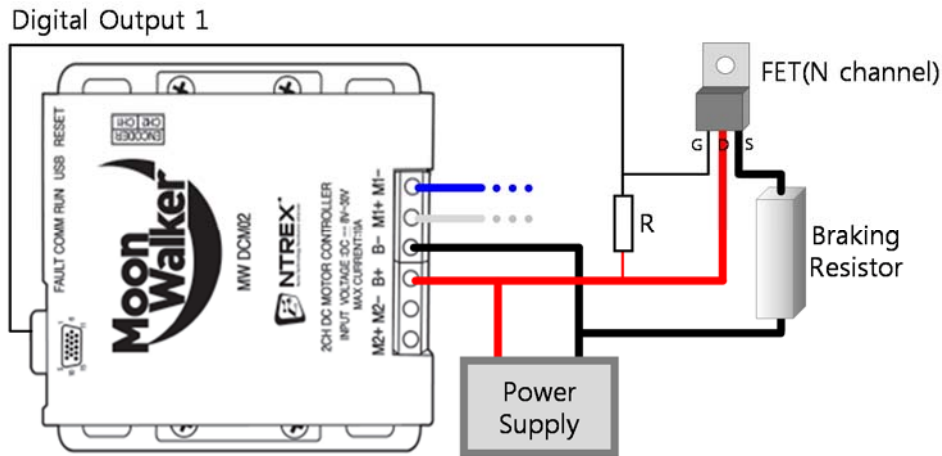


그림 8-13 회생 브레이크를 사용하기 위한 회로

회생회로는 모터에 부하로 작용하므로 브레이크 저항(Braking Resistor)의 에너지 소비율에 따라 회생 브레이크의 성능이 달라집니다. 큰 회생 에너지를 소비시켜야 할 경우에는 저항 값이 낮고 큰 와트의 브레이크 저항을 사용하도록 설계합니다.

※주의※ 감속시 또는 상하측의 하강시 브레이크 저항에서 고열이 발생할 수 있습니다. 또한 FET의 고장으로 회로가 단락된 경우, 브레이크 저항에서 고열의 발생과 화재 위험이 있습니다.

8.3.2 디지털 출력 채널의 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 디지털 출력 채널의 파라미터를 설정합니다. 그림 8-14와 같이 Digital Output Channel 1 그룹에서 Enable 항목을 체크하고 Output channel Mapping 서브 그룹에서 Source를 Motor Channel 1으로 선택하고 Status를 High Voltage (Brake Resistor ON)으로 선택합니다.

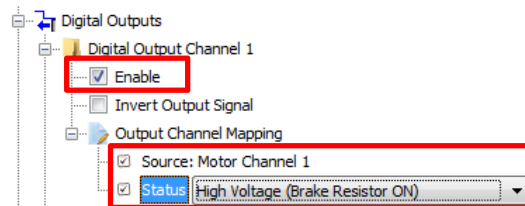


그림 8-14 회생 브레이크 사용을 위한 디지털 출력 채널 설정

그리고 그림 8-15와 같이 Digital Output On Conditions 그룹에서 High Voltage 항목에 회생 브레이크가 동작할 전압 값을 설정합니다.

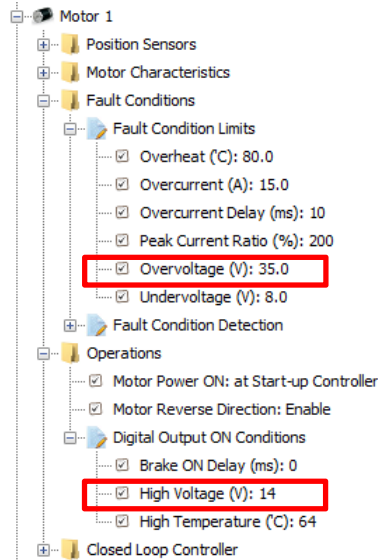


그림 8-15 모터의 High Voltage 설정

'High Voltage' 오브젝트는 설정한 값 주변에서 회생 브레이크가 반복적으로 ON/OFF 되는 것을 막기 위해 $\pm 0.5V$ 의 작동 편차가 있습니다. 예를 들어 그림 8-15와 같이 'High Voltage' 오브젝트를 14V로 설정할 경우 전원의 전압('Voltage' 오브젝트)이 14.5V 이상이면 회생 브레이크가 작동하고 13.5V 이하로 내려가면 작동을 멈춥니다. 그리고 13.5V와 14.5V 범위에서는 이전 상태를 유지하게 됩니다.

또한, 'High Voltage'는 폴트(Fault) 발생 조건이 되는 'Overvoltage Limit' 오브젝트 설정 값보다 낮은 값을 설정해야 합니다.

모든 설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

8.3.3 회생 브레이크 작동 확인

그림 8-15에서와 같이 High Voltage 항목의 값을 설정할 경우 배터리 전압이 14.5V 이상 되면 그림 8-16과 같이 I/O Monitoring 탭에서 해당 LED가 켜지는지 확인합니다. 해당 LED가 켜지면 회생 브레이크가 동작하여 전원의 전압 상승을 억제합니다. 그리고 모터는 제동력을 얻게 됩니다.

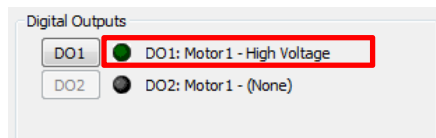


그림 8-16 디지털 출력 채널 모니터링

실제 회생 브레이크의 작동을 실험하기 위해 전원으로 파워서플라이를 사용합니다. 그리고 모터에 플라이휠(Fly Wheel)과 같이 회전 관성이 큰 부하를 연결합니다. 여기서 전원 전압은 12V이고 'High Voltage' 오브젝트를 14V로, 'Overvoltage Limit' 오브젝트를 35V로 설정합니다.

관성이 큰 부하가 연결된 모터를 고속으로 회전하다가 갑자기 정지 시키면 그림 8-17의 왼쪽 그

래프와 같이 회생전류로 인해 전원의 전압이 치솟습니다. 이때 회생 브레이크를 사용하면 오른쪽 그래프와 같이 전압 상승이 억제되고 모터가 급속히 정지하는 것을 볼 수 있습니다.

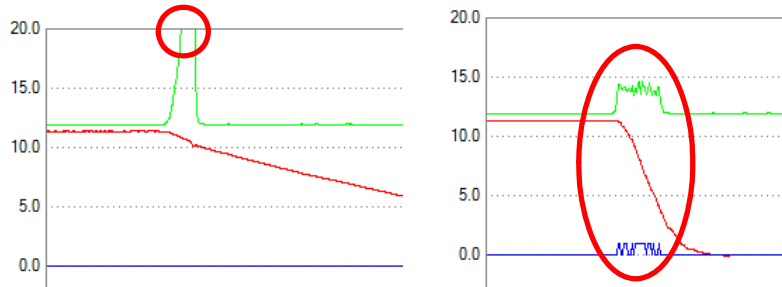


그림 8-17 회생 브레이크 사용/비사용시의 전원 전압 및 모터의 회전 속도 비교

그림 8-17의 그래프에서 녹색은 전원의 전압이고 적색은 모터의 속도 입니다. 청색은 디지털 출력 채널 1의 값입니다.

그림 8-17의 왼쪽은 Digital Output Channel 1의 Enable을 체크하지 않은 상태(회생 브레이크를 사용하지 않음)에서 모터의 속도를 갑자기 줄였을 때의 그래프 입니다. 그림 8-17의 왼쪽에서와 같이 Overvoltage 폴트가 나면서 모터 Power OFF 상태가 됩니다. 그리고 모터는 천천히 속도를 줄여 정지하게 됩니다.

오른쪽 그림은 Digital Output Channel 1의 Enable을 체크한 상태(회생 브레이크를 사용)에서 모터의 속도를 갑자기 줄였을 때의 그래프 입니다. 디지털 출력 채널이 켜지면서 브레이크 저항이 전원단 회로를 단락시켜 전압 상승을 억제합니다. 그리고 모터를 단시간에 감속하는 것을 볼 수 있습니다.

8.4 제어기에 냉각팬 사용

장시간 모터를 구동하거나 높은 온도 환경에서 모터를 구동할 때 제어기는 열로 인해 고장 또는 파손될 가능성이 있습니다. 따라서 한계 온도('Overheat Limit' 오브젝트)에 도달하기 전에 외부에서 냉각팬으로 제어기의 방열판을 식혀야 합니다.

8.4.1 냉각팬 배선

제어기 전원을 사용하여 냉각팬을 구동하기 위해 그림 8-18과 같이 간단한 회로를 구성합니다.

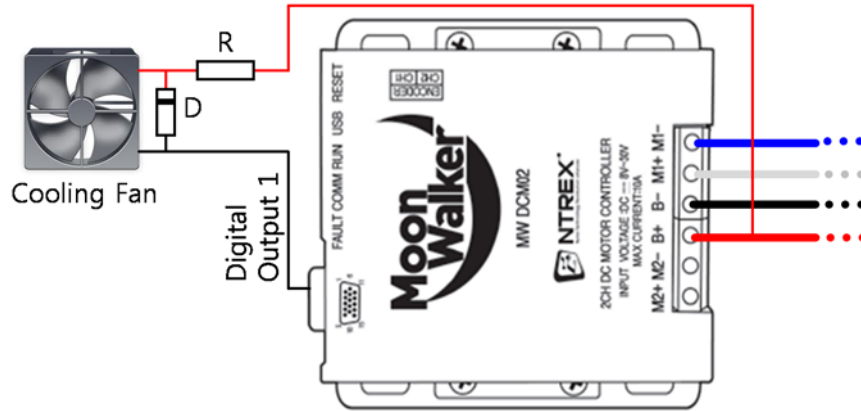


그림 8-18 냉각팬 사용을 위한 외부 회로(제어기 전원 사용)

냉각팬의 정격 전압과 소비 전류를 고려하여 저항 R의 값을 정합니다. 여기서는 정격 전압 12V와 정격 전류 0.13A의 냉각팬을 사용합니다. 이 경우 저항 R에 100에서 150Ω 사이의 값을 사용하면 됩니다.

제어기 전원을 사용하지 않고 외부 전원(그림에서 Battery로 표시)을 따로 구성할 경우에는 그림 8-19과 같이 회로를 구성합니다. 이 때는 I/O 인터페이스의 GND 단자와 외부 전원의 음극(-) 단자는 서로 연결되어야 합니다.



그림 8-19 냉각팬 사용을 위한 외부 회로(독립 전원 사용)

냉각팬은 인덕턴스 성분을 가지는 부하이기 때문에 부하 양단에 다이오드를 역방향으로 삽입합니다.

8.4.2 디지털 출력 채널의 파라미터 설정

Motor Control UI 유틸리티의 Configuration 탭에서 디지털 출력 채널의 파라미터를 설정합니다. 그림 8-20와 같이 Digital Output Channel 1 그룹에서 Enable 항목을 체크하고 Output channel Mapping 서브 그룹에서 Source를 Motor Channel 1으로 선택하고 Status를 High Temperature (Cooling Fan ON)으로 선택합니다.

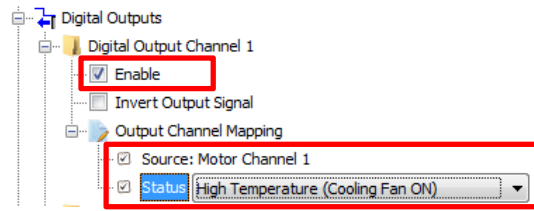


그림 8-20 냉각팬 사용을 위한 디지털 출력 채널 설정

그리고 그림 8-21과 같이 Digital Output On Conditions 그룹에서 High temperature 항목에 냉각팬이 동작할 온도를 설정합니다.

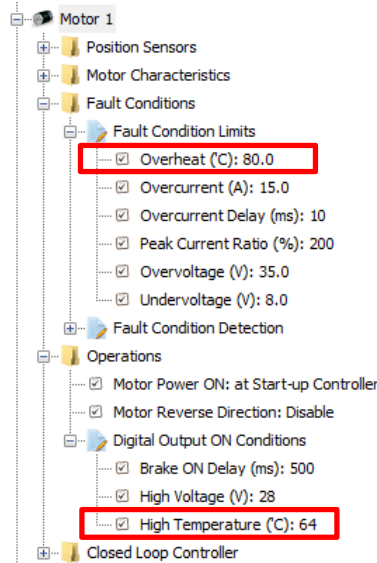


그림 8-21 냉각팬 사용을 위한 파라미터 설정

'High Temperature' 오브젝트는 설정 값 주변에서 냉각팬이 반복적으로 ON/OFF 되는 것을 막기 위해 $\pm 0.5^{\circ}\text{C}$ 의 작동 편차가 있습니다. 예를 들어 그림 8-21과 같이 'High Temperature' 항목을 64°C 로 설정할 경우 방열판의 온도가 64.5°C 이상이면 냉각팬이 작동하고 63.5°C 이하로 내려가면 작동을 멈춥니다. 그리고 63.5°C 와 64.5°C 범위에서는 이전 상태를 유지하게 됩니다.

또한, 'High Temperature'는 폴트(Fault) 발생 조건이 되는 'Overheat Limit' 설정 값보다 낮은 값을 설정해야 합니다.

모든 설정이 끝나면 [Write configurations] 버튼과 [Save to Flash] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 하고 제어기의 플래시메모리에 영구 저장합니다.

8.4.3 냉각팬 작동 확인

그림 8-21에서와 같이 High Temperature 항목의 값을 64°C 로 설정할 경우 제어기의 방열판 온도가 64.5°C 이상 되면 그림 8-22과 같이 I/O Monitoring 탭에서 해당 LED가 켜지는지 확인합니다. 해당 LED가 켜지면 냉각팬이 동작하여 방열판의 온도 상승을 억제합니다.

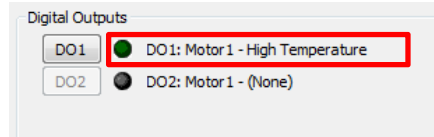


그림 8-22 디지털 출력 채널 모니터링

그리고 그림 8-23와 같이 Controller Monitoring에서 FET와 방열판의 온도를 모니터링할 수 있습니다. 이 값을 확인하도록 합니다.

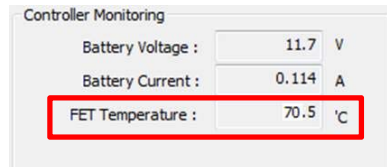


그림 8-23 FET 및 방열판 온도 모니터링

※주의※ 사용자는 자신의 제품 검증을 수행하시기 바라며, 전적으로 어플리케이션 노트에 포함된 정보에 의존하지 마시기 바랍니다.

9 스크립트 작성과 응용

스크립트를 작성하고 실행하는 것은 제어기의 강력한 기능 중 하나입니다. 스크립트는 제어기의 파라미터 설정만으로 불가능한 다양한 작업을 수행할 수 있도록 합니다.

본 장에서는 스크립트 언어를 사용하여 제어기의 활용을 극대화 할 수 있는 스크립트 예제들에 대해 소개합니다.

제어기가 제공하는 Mini-C 스크립트 언어와 사용 방법에 대해서는 사용자 매뉴얼에서 "14 Mini-C 스크립트 언어" 장을 참조하기 바랍니다.

9.1 모터 구동 파형 생성과 테스트

이 절에서는 모터에 일반적인 주기적 파형을 가하여 모터의 구동 특성을 테스트 할 수 있도록, 스크립트로 각종 파형을 생성하고 모터에 전압, 전류, 속도, 위치 명령을 내리는 스크립트를 작성 합니다.

9.1.1 파형 생성 개요

스크립트에서 사용할 주기적 파형은 다음과 같습니다(출처: 위키백과, 파형):

- 정현파 (sine wave): $\sin(2 \pi t)$
- 구형파 (square wave): $\text{saw}(t) - \text{saw}(t - \text{duty})$.
- 삼각파 (triangle wave): $(t - 2 \text{floor}((t + 1)/2))(-1)^{\text{floor}((t + 1)/2)}$.
- 톱니파 (sawtooth wave): $2(t - \text{floor}(t)) - 1$.

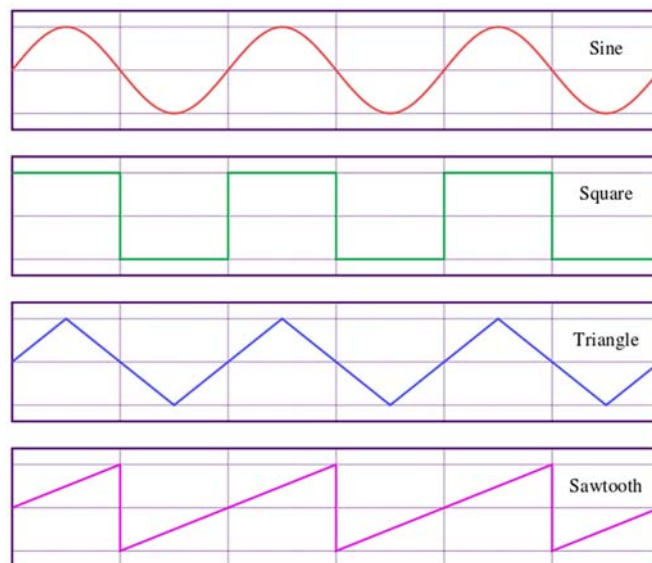


그림 9-1 파형의 종류(출처: 위키백과)

그림 9-1에서와 같은 목록의 주기적 파형 중 하나를 생성하고 이를 모터의 전압, 전류, 속도, 위치 명령 중 하나로 인가합니다. 그리고 Real-time Plot에서 목표 명령에 대한 모터의 현재 측정 값을 시간의 흐름에 따라 비교함으로써, 제어기 및 모터의 구동 파라미터를 조정하고 최적의 설정 값을 찾는데 도움을 줄 수 있습니다.

9.1.2 스크립트 작성

```
/*
    SCRIPT Example 1: ex1_wave_test.scr
    정현파, 구형파, 삼각파, 톱니파를 만들어 모터의 전압, 전류, 속도, 위치 제어 테스트

    NTREX 기술연구소, 2014.1.10
*/

motor_channel      = 1;                // 채널 1의 모터 선택
period             = 5000;             // 5000ms 주기
wave_type          = 3;                // 파형 선택
motor_command      = 1;                // 모터 구동 방식

setv(_command, motor_channel, 2);      // Clear Fault Flags
setv(_command, motor_channel, 1);      // Motor Power ON

// 위치, 속도, 전류, 전압에 대한 진폭(amplitude) 설정
max_position = 10000;
max_velocity = 1000;
max_current = 1;
max_voltage = 12;

while (1) {
    sleep (10);
    t = clock ();
    v = 0;

    // 정현파, 구형파, 삼각파, 톱니파 중 하나 생성
    if (wave_type == 1) {                // Sine wave
        v = sin(2*_PI*t/period);
    }
    else if (wave_type == 2) {           // Square wave
        v = sin(2*_PI*t/period);
        if (v >= 0) v = 1;
        else      v = -1;
    }
    else if (wave_type == 3) {           // Triangle wave
        v = 4*(t%period)/period - 2;
        if (-1 >= v)    v = -2 - v;
        else if (1 <= v) v = 2 - v;
    }
    else if (wave_type == 4) {           // Sawtooth wave
        v = 2*(t%period)/period - 1;
    }
}
```

```

}

// 모터의 전압, 전류, 속도, 위치 명령 내보냄
if (motor_command == 1) {          // Voltage command
    setv(_voltage_command, motor_channel, max_voltage*v);
}
else if (motor_command == 2) {     // Current command
    setv(_current_command, motor_channel, max_current*v);
}
else if (motor_command == 3) {     // Velocity command
    setv(_velocity_command, motor_channel, max_velocity*v);
}
else if (motor_command == 4) {     // Position command
    setv(_position_command, motor_channel, max_position*v);
}
}
}

```

9.1.3 빌드 및 다운로드

Motor Control UI 유틸리티에서 Script 탭을 선택했을 때 보이는 스크립트 작성 화면은 그림 9-2 와 같습니다. 소스코드 편집 창에서 상기 스크립트 코드를 입력합니다. 스크립트의 작성, 빌드 및 다운로드 방법에 대한 자세한 내용은 사용자 매뉴얼의 "15장 프로그램의 작성과 실행"을 참고하기 바랍니다..

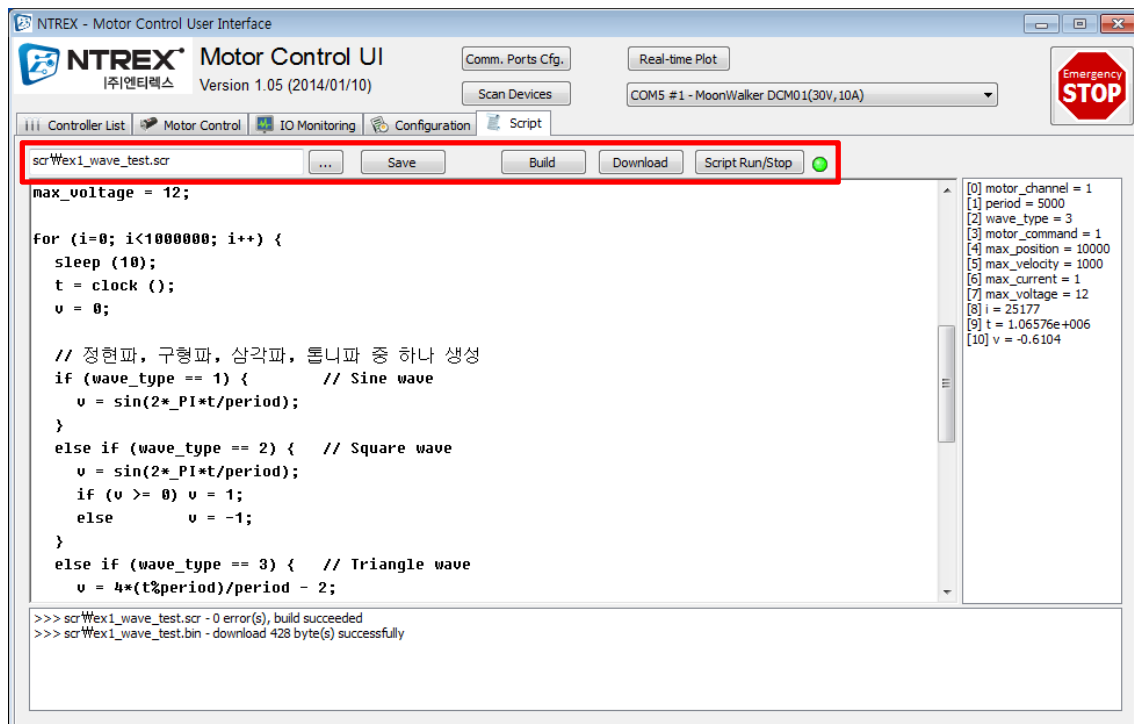


그림 9-2 스크립트 작성 화면

소스코드의 입력이 끝나면, 좌측 상단 에디트 박스에 파일 이름을 입력하고 [Save] 버튼을 눌러 작성한 내용을 파일로 저장합니다. 그리고 [Build] 버튼을 눌러 Error 없이 컴파일 되는지 확인함

니다. 빌드를 성공하면 [Download] 버튼을 눌러 바이트코드를 제어기로 다운로드 합니다.

다운로드가 완료되면 [Script Run/Stop] 버튼을 눌러 제어기에 다운로드 된 스크립트를 실행/중단 할 수 있습니다.

9.1.4 구동 테스트

스크립트에서 설정 변수가 다음과 같을 때,

```
motor_channel      = 1;           // 채널 1의 모터 선택
period             = 5000;        // 파형 주기: 5000ms
wave_type          = 1;           // 파형 선택: 구형파
motor_command      = 1;           // 모터 구동 방식: 전압 구동
```

제어기의 전압 명령과 모터에 실제 가해지는 전압은 그림 9-3과 같습니다.

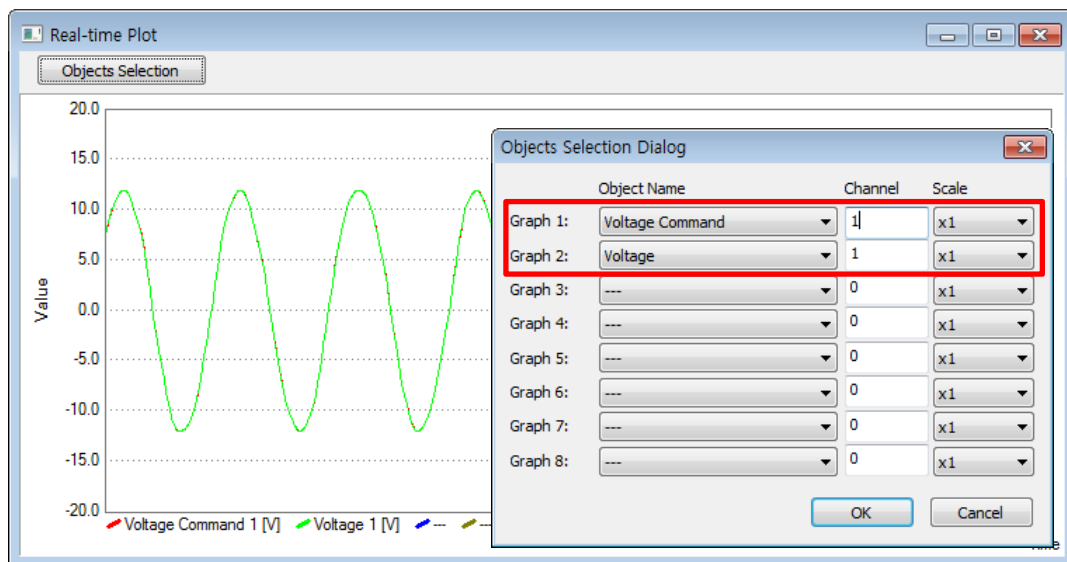


그림 9-3 전압 명령에 정현파를 가했을 때

모터에 구형파(square wave) 전압 명령을 내보내기 위해 스크립트의 소스코드를 "wave_type = 2;"로 수정합니다. 그리고 다시 빌드 하고 다운로드 하여 실행하면, 제어기의 전압 명령과 모터에 실제 가해지는 전압은 그림 9-4와 같이 표시됩니다.



그림 9-4 구형파를 전압 명령에 가했을 때

여기서 전압 명령(그래프에서 'Voltage Command')을 모터에 실제 가해지는 전압(그래프에서 'Voltage')이 즉시 추종하지 못하는데, 이는 모터 파라미터 중 '**Profile Mode**' 오브젝트를 1로 설정하여 전압 출력에 사다리꼴 프로파일이 적용되었기 때문입니다.

모터에 삼각파(triangle wave) 전압 명령을 내리기 위해 스크립트의 소스코드를 `"wave_type = 3;"`로 고칩니다. 그리고 `"period = 5000"`, `"period = 3000"`, `"period = 2000"`, `"period = 1000"` 순으로 주기를 변경하면서 테스트했을 때의 결과는 그림 9-5와 같습니다.

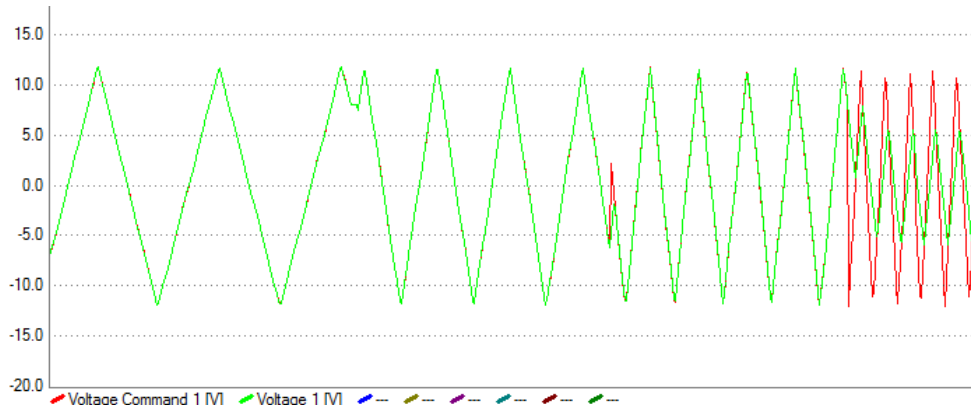


그림 9-5 삼각파를 전압 명령에 가했을 때

모터에 3초 주기의 톱니파(sawtooth wave) 전압 명령을 내리기 위해 스크립트의 소스코드를 `"wave_type = 4;"`, `"period = 3000;"`로 수정하고 테스트했을 때의 결과는 그림 9-6과 같습니다.

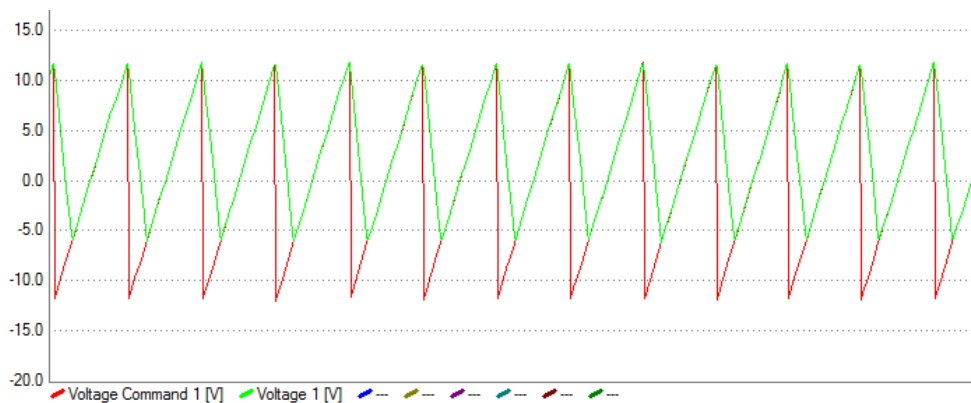


그림 9-6 톱니파를 전압 명령에 가했을 때

모터에 5초 주기의 정현파(sine wave) 전류 명령을 내리기 위해 스크립트의 소스코드를 `"wave_type = 1;"`, `"period = 5000;"`, `"motor_command = 2;"`로 수정하고 테스트했을 때 전류 명령과 모터에 실제로 흐르는 전류는 그림 9-7과 같습니다.

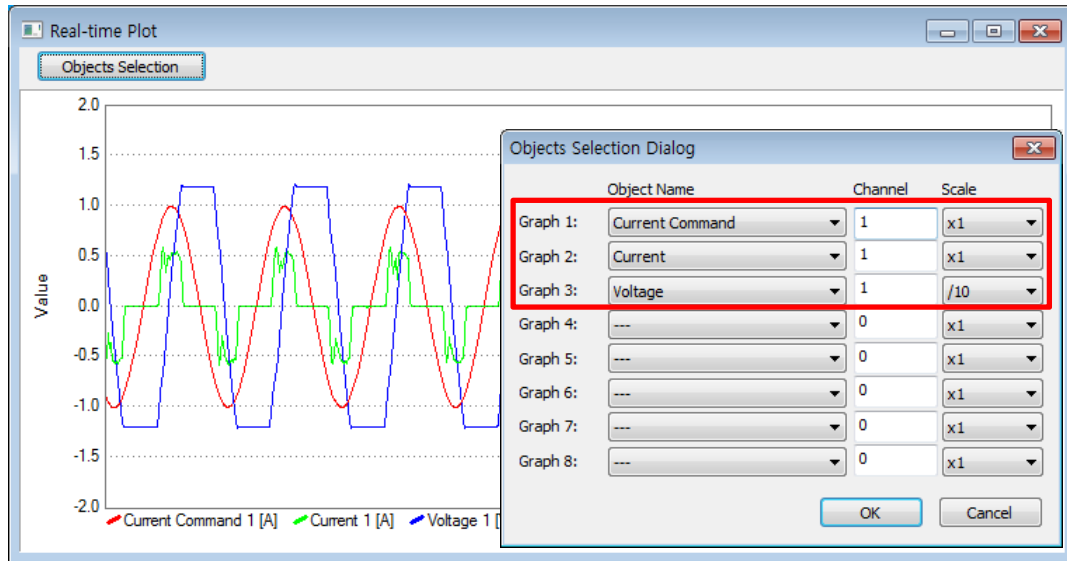


그림 9-7 정현파를 전류 명령에 가했을 때

상기 예제 스크립트를 활용하여 몇 가지 파형에 대해 모터의 명령과 응답을 연속적으로 테스트 할 수 있습니다. 이는 수작업으로 실험을 반복하면서 위치, 속도, 전류 제어기의 이득을 동조하는 과정을 쉽게 할 수 있도록 합니다.

수동으로 위치, 속도, 전류 제어기의 이득을 동조하기 전에, 스크립트를 실행하고 그림 9-7에서와 같이 Real-time Plot 창을 띄워 둡니다. 그리고 UI 유틸리티의 Configuration 탭에서 제어기의 이득을 바꾸었을 때, 모터의 명령과 응답을 Real-time Plot 창을 통해 바로 관찰할 수 있습니다.

이와 같이 스크립트를 활용하면 제어기의 이득 변경 → 명령 생성 → 응답성 확인 과정을 동시에 할 수 있게 됩니다.

9.2 원점 복귀

원점이란 기구물의 동작 기준이 되는 고정된 임의의 지점으로, 보통 홈 센서가 감지된 위치로부터 오프셋을 알면 기구적 원점을 찾을 수 있게 됩니다.

충분 엔코더를 사용하는 제어기에서 전원이 꺼졌다 켜지면 모터의 현재 위치를 잃어버리게 됩니다. 이 때 모터를 위치제어 하기 위해서 제일 먼저 수행하는 과정이 원점 복귀 입니다.

원점 복귀는 먼저 홈 센서가 있는 위치로 이동 합니다. 그리고 홈 센서가 감지되는 지점에서 기구적 원점과의 오프셋 값을 모터 위치에 설정합니다. 홈 센서를 사용하는 방법에 대해서는 "7.3 홈 센서의 사용" 절의 내용을 참고하기 바랍니다.

이 절에서는 홈 센서를 사용한 원점 복귀 기능을 스크립트로 작성합니다.

9.2.1 스크립트 작성

```
/*
    SCRIPT Example 2: ex2_homing.scr
    홈 센서를 사용하여 모터의 원점을 찾는 스크립트

    NTREX 기술연구소, 2014.1.10
*/

motor_channel = 1;           // 모터 선택
homing_direction = 1;        // 1 - 홈 센서 감지 영역이 모터의 정방향에 있을 때,
                             // -1 - 홈 센서 감지 영역이 모터의 역방향에 있을 때
homing_speed = 10;           // 원점복귀 속도 (단위: RPM)
homing_voltage = 3;          // 원점복귀 전압, 만일 homing_speed가 0일 때는 전압 구
동

setv (_command, motor_channel, 2);    // Clear Fault Flags
setv (_command, motor_channel, 1);    // Motor Power ON

// Digital Input 1을 Enable 한다.
di_enable = getv (_di_enable, 0);
setv (_di_enable, 0, di_enable | 1);

// 만일, 지금 홈 센서가 감지되는 영역에 있다면 홈 센서를 벗어나야 한다.
// 홈 센서가 감지되지 않으면, 다음 과정으로 넘어간다.
home_sensor = getv(_di_value, 1);
if (home_sensor) {
    if (homing_speed > 0) {
        setv (_velocity_command, motor_channel, -
homing_direction*homing_speed);
    }
    else {
        setv (_voltage_command, motor_channel, -
homing_direction*homing_voltage);
    }

    while (1) {
        sleep (10);
        home_sensor = getv(_di_value, 1);
        if (!home_sensor) break;
    }
    sleep (1000); // 홈 센서를 약간 지나쳐 멈추도록 1초 지연
    setv (_command, motor_channel, 6);    // 정지 명령
}

// 홈 센서로 이동 시작
if (homing_speed > 0) {
    setv (_velocity_command, motor_channel,
homing_direction*homing_speed);
}
else {
    setv (_voltage_command, motor_channel,
```

```

homing_direction*homing_voltage);
}

// 홈 센서가 감지되면 정지
while (1) {
    sleep (10);
    home_sensor = getv(_di_value, 1);
    if (home_sensor) {
        setv (_command, motor_channel, 6);        // 정지 명령
        break;
    }
}

// 홈 센서가 더이상 작동하지 않도록 Digital Input 1을 Disable 한다.
di_enable = getv (_di_enable, 0);
setv (_di_enable, 0, di_enable & ~1);

```

9.2.2 빌드 및 다운로드

빌드 및 다운로드 방법은 "9.1.3 빌드 및 다운로드"을 참고하기 바랍니다.

9.2.3 구동 테스트

원점 복귀 과정은 홈 센서의 감지 여부에 따라 달라집니다. 만일 지금 홈 센서가 감지되고 있다면, 홈 센서가 감지되지 않는 영역으로 이동 후 원점 복귀 과정을 시작합니다. 홈 센서가 감지되고 있지 않다면, 바로 원점 복귀 과정을 시작합니다.

먼저, 홈 센서가 감지되지 않는 영역에 모터를 위치시키고 스크립트를 실행해 봅니다. 그리고 원점 복귀 과정을 완료하는지 확인합니다. 원점 복귀를 완료하면 스크립트는 종료 됩니다.

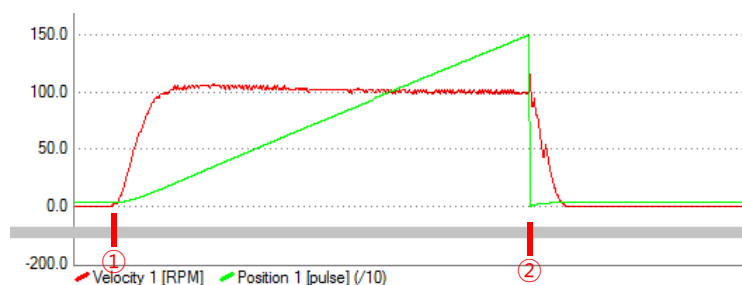


그림 9-8 홈 센서 비감지 영역에서의 원점복귀

그림 9-8에서 원점 복귀 과정을 설명하자면, ①에서 스크립트의 원점 복귀를 위한 이동이 시작됩니다. 그리고 ②에서 홈 센서를 감지 하여 원점이 설정되고 원점복귀 과정은 끝납니다.

다음으로, 홈 센서가 감지되는 영역에 모터를 위치시키고 스크립트를 실행해 봅니다. 그리고 원점 복귀 과정을 완료하는지 확인합니다.

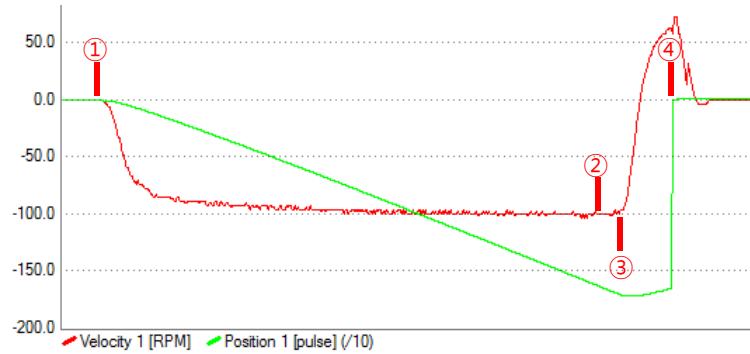


그림 9-9 홈 센서 감지 영역에서의 원점복귀

원점 복귀 과정이 정상적으로 완료되면, 모터는 홈 센서를 만나는 순간 정지하고, 'Home Position' 오브젝트에 설정한 값이 모터의 위치('Position' 오브젝트)에 복사됩니다.

그림 9-9에서 원점 복귀 과정을 설명하자면, ①에서 현재 홈 센서가 감지되고 있기 때문에, 홈 센서가 감지되지 않는 영역으로의 이동이 시작됩니다. 홈 센서가 더 이상 감지되지 않는 위치가 ②번 위치입니다. 그리고 1초 후 ③에서 다시 홈 센서를 감지하기 위해 반대 방향으로 이동을 시작합니다. 마지막으로 ④에서 홈 센서를 감지하여 원점이 설정되고 원점복귀 과정은 끝납니다.

9.3 컨베이어 구동

컨베이어(conveyor)는 주로 공장이나 물류센터에서 재료나 화물을 운반하는 용도로 사용하는 기계 장치입니다. 보통 고무로 만들어진 벨트를 모터로 회전시켜서 구동하게 됩니다.

이 절에서는 지정된 구간을 왕복 운동하는 컨베이어를 제어하는 스크립트를 작성합니다.

9.3.1 컨베이어 개요

그림 9-10에서와 같이 사용자가 푸쉬 버튼(Move Forward Button과 Move Backward Button)을 눌러 컨베이어 구동을 시작하고 물건을 이송합니다. 그리고 벨트의 양 끝에 설치된 리미트 센서(Reverse Limit Sensor, Forward Limit Sensor)에 물건이 감지될 때 멈추는 간단한 왕복 운동하는 장치를 제작합니다.

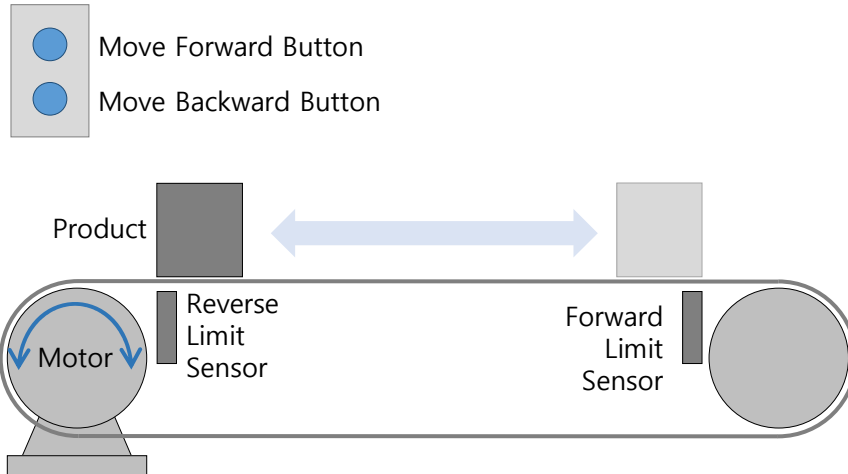


그림 9-10 왕복 운전하는 컨베이어 벨트

그리고 그림 9-11와 같이 제어기의 디지털 입력 포트와 아날로그 입력 포트에 푸쉬 버튼과 마이크로 스위치를 연결하여 회로를 구성합니다.

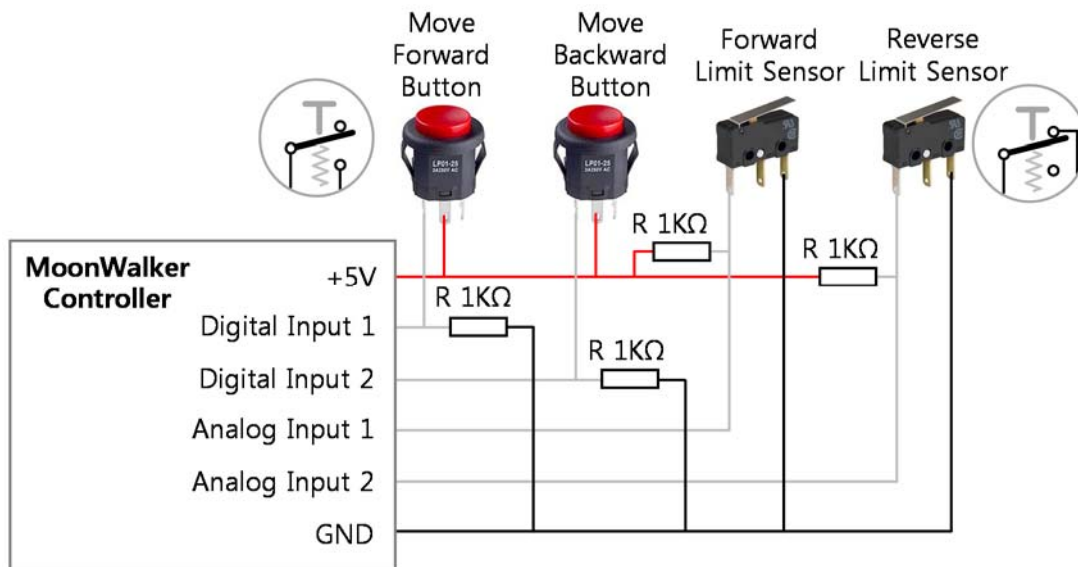


그림 9-11 디지털 입력 포트와 아날로그 입력 포트 배선

푸쉬 버튼과 마이크로 스위치가 연결된 디지털 입력 포트와 아날로그 입력 포트는 스크립트에서 직접 읽어 모터의 동작을 제어해야 함으로, 그림 9-12, 그림 9-13과 같이 **매핑 기능을 사용하지 않도록 Action 항목을 모두 '(None)'으로 선택**합니다. 단지 각 입력 채널을 활성화 하도록 Enable 항목만 체크해 둡니다.

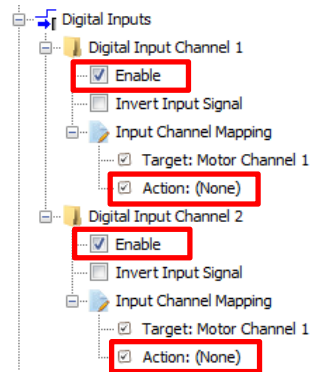


그림 9-12 디지털 입력 채널 설정

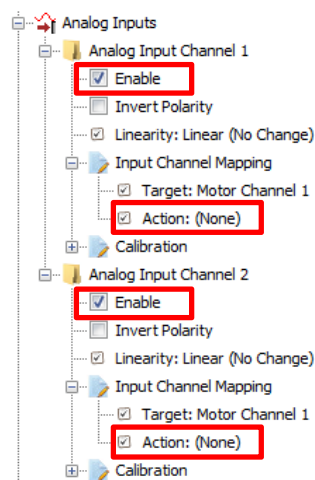


그림 9-13 아날로그 입력 채널 설정

9.3.2 스크립트 작성

```

/*
    SCRIPT Example 3: ex3_conveyor.scr
    사용자가 버튼을 눌러 컨베이어 벨트의 물건을
    다른 쪽 끝으로 옮기는 간단한 스크립트 예제

    NTREX 기술연구소, 2014.1.10
*/

motor_channel = 1;
motor_voltage_out = 5;

setv(_command, motor_channel, 2);    // Clear Fault Flags
setv(_command, motor_channel, 1);    // Motor Power ON

mfb_count = 0;
mbb_count = 0;
rls_count = 0;
fls_count = 0;

```

```
while (1) {
    sleep (10);

    move_forward_button = getv(_di_value, 1);
    move_backward_button = getv(_di_value, 2);
    reverse_limit_switch = getv(_ai_value, 1);
    forward_limit_switch = getv(_ai_value, 2);

    if (move_forward_button) mfb_count++;
    else mfb_count = 0;

    if (move_backward_button) mbb_count++;
    else mbb_count = 0;

    if (reverse_limit_switch > 2047) rls_count++; // 입력 범위: 0 ~ 4095
    else rls_count = 0;

    if (forward_limit_switch > 2047) fls_count++; // 입력 범위: 0 ~ 4095
    else fls_count = 0;

    if (rls_count > 10) { // 10ms 이상 Reverse Limit Switch 가 작동함
        setv (_voltage_command, motor_channel, 0);
    }
    else if (fls_count > 10) { // 10ms 이상 Forward Limit Switch 가 작동함
        setv (_voltage_command, motor_channel, 0);
    }
    else if (mfb_count == 10) { // 10ms 이상 Move Forward Button 이 눌림
        setv (_voltage_command, motor_channel, motor_voltage_out);
    }
    else if (mbb_count == 10) { // 10ms 이상 Move Backward Button 이 눌림
        setv (_voltage_command, motor_channel, -motor_voltage_out);
    }
}
```

9.3.3 구동 테스트

컨베이어 벨트에 물건을 올립니다. 그리고 'Move Forward Button'을 눌러 컨베이어가 'Forward Limit Sensor'에서 물건이 감지될 때까지 이동하는지 확인합니다. 그리고 'Move Backward Button'을 눌렀을 때는 'Reverse Limit Sensor'에서 물건이 감지될 때까지 이동하는지 확인합니다.

9.4 모터 온도 측정과 냉각

제어기 내부의 FET와 방열판의 온도를 측정하고 한계 온도 이상 올라가면 팬을 구동하여 제어기의 온도를 낮추는 기능에 대해서 "8.4 제어기에 냉각팬 사용" 절에서 설명했습니다.

이 절에서는 모터의 온도를 측정하는 써미스터 센서를 모터 표면에 설치하고 설정한 온도 이상이 되면 제어기의 출력을 낮추거나 냉각팬을 켜는 기능을 가지는 스크립트를 작성합니다.

9.4.1 모터 온도 측정과 냉각 회로 구성

모터의 온도 측정을 위해 써미스터를 사용합니다. 써미스터는 아래와 같은 특성을 가진 10KΩ NTC를 사용하길 권장합니다(여기서는 NTC-103F397F를 사용함). 이때 외부에 10KΩ 풀다운(Pull-Down) 저항을 추가하여야 합니다. 풀다운 저항은 선정된 써미스터의 저항 수치에 따라 적절하게 선정합니다.

※ 써미스터 모델, 브랜드, 저항 정밀도 그리고 사용자 환경에 따라서 아날로그 전압 값이 달라질 수 있습니다.

냉각 팬 사용:

써미스터(NTC-103F397F)와 냉각팬을 그림 9-14에서와 같이 연결하여 모터의 온도 측정과 냉각 회로를 구성합니다. 스크립트는 써미스터에서 측정된 온도가 특정 값 이상으로 올라가면 냉각팬을 작동하여 모터의 열을 식힙니다.

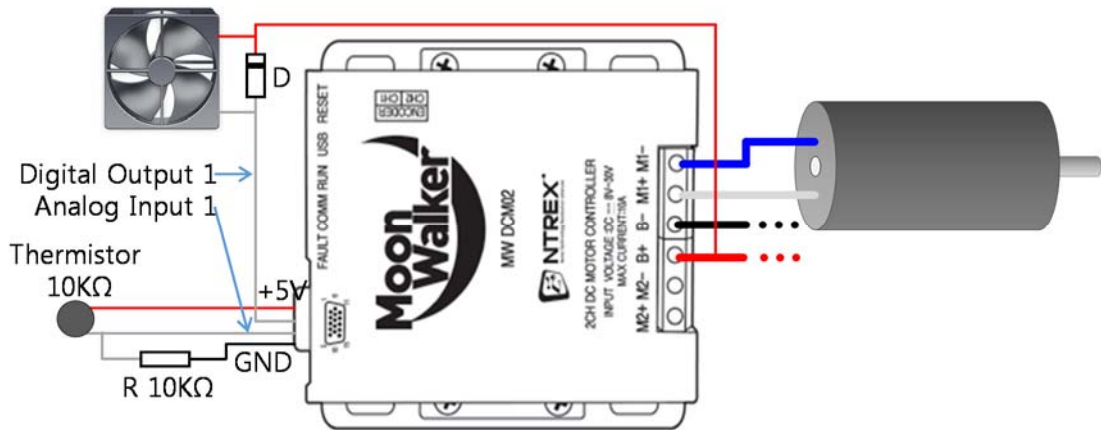


그림 9-14 모터 냉각 회로 구성

상기와 같이 써미스터 회로를 구성하였을 때, 아날로그 입력 채널에서 읽은 원시 값(raw value)으로부터 써미스터의 저항은 다음과 같이 계산합니다: 먼저 원시 값을 전압(Voltage)으로 바꿉니다.

$$\text{Voltage} = \text{Value} \times 5 / 4095$$

그리고 전압으로부터 써미스터의 저항($R_{\text{thermistor}}$)을 계산합니다($R_{\text{pull_dn}}$ 은 풀다운 저항, 10K).

$$R_{thermistor} = R_{pull_dn} \times (5 - Voltage) / Voltage$$

써미스터는 모터의 온도를 측정할 수 있도록 모터 표면에 설치되어야 합니다. 그리고 냉각팬도 모터에 송풍할 수 있도록 설치되어야 합니다.

최대 전류 제한:

또 다른 방법으로 써미스터의 측정 온도가 특정 값 이상으로 올라가면 모터에 흘릴 수 있는 최대 전류를 제한하여 모터의 출력을 낮춤으로 모터의 온도 상승을 방지할 수도 있습니다.

만일 모터 권선의 저항이 R이고 흐르는 전류가 I라면 모터에서 발생하는 열은 I^2R 이 됩니다. 따라서 I를 1/2로 낮추면 발생하는 열은 1/4로 낮아지게 됩니다. 대신 모터의 출력 토크도 1/2로 낮아지게 됩니다.

모터의 최대 전류 제한 실험을 위해 그림 9-15과 같이 DC Motor 2에 조절 가능한 부하 저항 (Load Resistor)를 연결합니다. 그리고 제어기에 연결된 DC Motor 1과는 커플링으로 연결합니다.

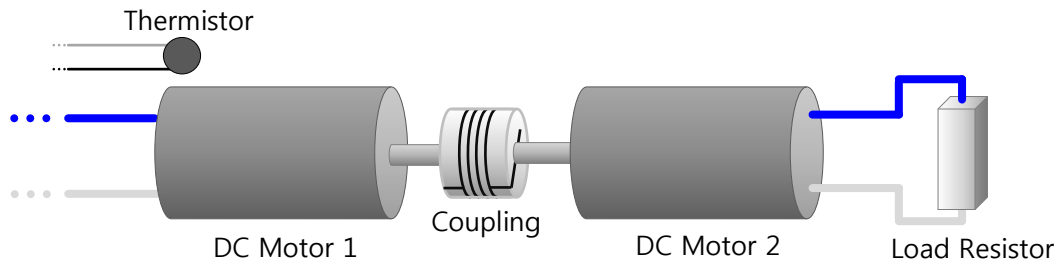


그림 9-15 테스트 모터에 부하 연결

그림 9-15에서 부하 저항을 조절하여 DC Motor 2의 브레이크 능력을 조절하고 결국 DC Motor 1에서 부하를 조절하게 됩니다.

9.4.2 I/O 채널의 파라미터 설정

써미스터가 연결된 아날로그 입력 채널을 활성화 하기 위해 Analog Input Channel 1 그룹에서 Enable을 체크합니다. 그리고 스크립트에서 아날로그 입력 포트의 원시 값(raw value)을 직접 읽기 때문에 매핑은 사용하지 않도록 설정합니다. 그리고 캘리브레이션도 사용하지 않기 때문에 설정하지 않아도 됩니다.

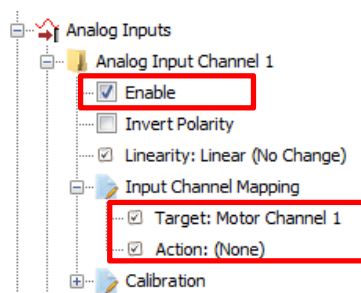


그림 9-16 아날로그 입력 채널의 설정

냉각 팬 사용:

냉각팬이 연결된 디지털 출력 채널을 활성화 하기 위해 Digital Output Channel 1 그룹에서 Enable을 체크합니다. 그리고 스크립트에서 디지털 출력 포트를 직접 제어하기 때문에 매핑 기능은 사용하지 않도록 설정합니다.

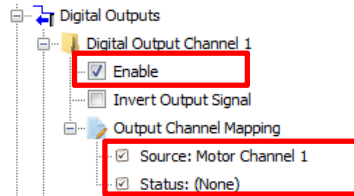


그림 9-17 디지털 출력 채널의 설정

모든 설정이 끝나면 [Write configurations] 버튼을 눌러 UI에서 변경한 값들을 제어기로 다운로드 합니다.

최대 전류 제한:

여기서는 추가 설정사항이 없습니다.

9.4.3 스크립트 작성

냉각 팬 사용:

/*

SCRIPT Example 3: ex3_motor_cooling.scr

써미스터에서 측정되는 온도가 특정 값 이상 되면 냉각 팬을 작동하여 모터를 냉각

테스트에 사용한 써미스터: NTC103F397F

0 'C	-	32.600Kohm
10 'C	-	19.885Kohm
20 'C	-	12.489Kohm
30 'C	-	8.058Kohm
40 'C	-	5.329Kohm
50 'C	-	3.606Kohm
60 'C	-	2.492Kohm
70 'C	-	1.756Kohm
80 'C	-	1.260Kohm
90 'C	-	0.920Kohm
100 'C	-	0.682Kohm

NTREX 기술연구소, 2014.1.10

*/

```
R_pull_dn = 10000;           // Pull down Resistance: 10Kohm
threshold = 2492;           // 2,492ohm 일 때 60도
```

```
while (1) {
    sleep (10);
```

```
v = getv(_ai_value, 1);
v *= 5/4095; // 전압으로 환산
R_th = R_pull_dn*(5 - v)/v; // 써미스터 저항으로 환산

setv (_temp_value, 1, R_th);
if (R_th < threshold) {
    setv (_do_value, 1, 1);
}
else {
    setv (_do_value, 1, 0);
}
}
```

최대 전류 제한:

```
/*
SCRIPT Example 4: ex4_motor_current.scr
써미스터에서 측정되는 온도가 특정 값 이상이 되면
모터에 흐르는 최대 전류를 제한하여 모터의 출력을 낮춤
테스트에 사용한 써미스터: NTC103F397F
    0'C - 32.600Kohm
    10'C - 19.885Kohm
    20'C - 12.489Kohm
    30'C - 8.058Kohm
    40'C - 5.329Kohm
    50'C - 3.606Kohm
    60'C - 2.492Kohm
    70'C - 1.756Kohm
    80'C - 1.260Kohm
    90'C - 0.920Kohm
    100'C - 0.682Kohm

NTREX 기술연구소, 2014.1.10
*/

motor_max_current = 5; // 최대 5A를 흘릴 수 있는 모터
R_pull_dn = 10000; // Pull down Resistance: 10Kohm
threshold1 = 3606; // 3,606ohm일 때 50도 --> 최대 전류의 75% 흘림
threshold2 = 2492; // 2,492ohm일 때 60도 --> 최대 전류의 50% 흘림
threshold3 = 1756; // 1,756ohm일 때 70도 --> 최대 전류의 25% 흘림
multiply_prev = 0;

while (1) {
    sleep (10);

    v = getv(_ai_value, 1);
    v *= 5/4095; // 전압으로 환산
    R_th = R_pull_dn*(5 - v)/v; // 써미스터 저항으로 환산

    multiply = 1;
```

```

if (R_th < threshold3) multiply = 0.25;
else if (R_th < threshold2) multiply = 0.50;
else if (R_th < threshold1) multiply = 0.75;

// 화면 표시를 위해 temp_value2 오브젝트에 써미스터 저항 값을 복사해 둠
setv (_temp_value, 2, R_th);
if (multiply != multiply_prev) {
    setv (_max_current, 1, multiply*motor_max_current);
    // 화면 표시를 위해 temp_value1 오브젝트에 최대 전류 값을 복사해 둠
    setv (_temp_value, 1, multiply*motor_max_current*1000);

    multiply_prev = multiply;
}
}

```

9.4.4 구동 테스트

냉각 팬 사용:

제어기에 첫 번째 스크립트를 빌드하여 다운로드 하고 실행합니다.

스크립트는 화면 표시를 위해 'Temp Value' 오브젝트에 써미스터의 저항 값을 기록합니다. Motor Control UI 유틸리티의 IO Monitoring 탭에서 Temp Values 오브젝트를 모니터링 해 봅니다.

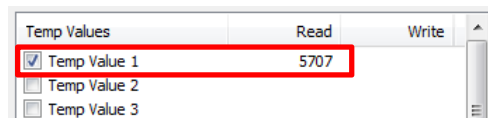


그림 9-18 써미스터의 저항 값 모니터링

그리고 모터를 구동하다 보면 모터에 열이 축적되어 모터의 온도가 점점 높아지게 됩니다. 모터의 온도가 스크립트에서 설정한 임계치(threshold)에 도달하면 냉각 팬이 동작하는지 확인합니다.

만일 모터에서 발생하는 열이 적을 때는, 테스트를 위해 열풍기와 같은 도구를 사용하여 모터에 장착된 써미스터를 가열해 봅니다(그림 9-19 참조, Temp Value 1은 써미스터의 저항 값).

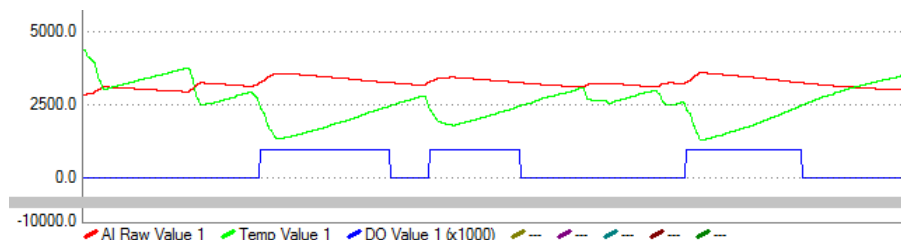


그림 9-19 모터의 온도와 냉각팬의 작동 그래프

최대 전류 제한:

두 번째 스크립트는 모터의 온도가 올라감에 따라 모터에 흘릴 수 있는 최대 전류를 낮춥니다. 먼저 제어기에 속도 명령을 내려 모터를 회전합니다. 그리고 모터에서 측정된 온도에 따라 모터의 출력이 떨어지는지 확인합니다.

스크립트는 화면 표시를 위해 'Temp Value' 오브젝트에 최대 전류를 mA 단위로 기록합니다. Motor Control UI 유틸리티의 IO Monitoring 탭에서 Temp Values 오브젝트를 모니터링 해 봅니다.

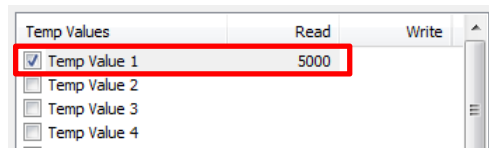


그림 9-20 최대 전류 모니터링

만일 모터에서 발생하는 열이 적을 때는, 테스트를 위해 열풍기와 같은 도구를 사용하여 모터에 장착된 써미스터를 가열해 봅니다(그림 9-21 참조, Temp Value 1은 모터의 최대 전류, Temp Value 2는 써미스터의 저항 값).

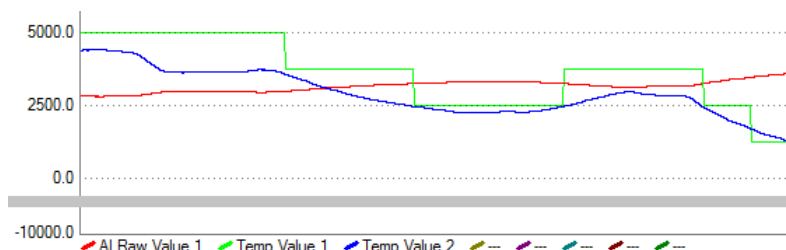


그림 9-21 모터의 온도와 모터의 최대 전류 제한 그래프

9.5 2륜 차동 구동 로봇 운전

2륜 차동 구동(Two wheel differential drive) 로봇은 구조가 간단하고 제작이 용이하며, 주로 실내 주행 용도로 사용되는 로봇입니다.

이 절에서는 2륜 차동 구동 로봇에 속도 주행 명령을 내리고 데드레쿠닝(dead-reckoning)으로 로봇의 위치를 추정하는 스크립트를 작성합니다.

9.5.1 2륜 차동 구동 로봇 개요

2륜 차동 구동 로봇의 구조 및 제어기 연결과 설정에 대해서는 "5.1 2륜 차동 구동 로봇" 절을 참고하기 바랍니다. 그리고 로봇을 RC 조종기로 조종하기 위해서는 "5.4 RC 조종기로 로봇 조종" 절을 참고하기 바랍니다. 이 절에서는 그림 9-22과 같이 엔코더와 감속기가 장착된 모터에 바퀴가 연결된 일체형 모듈을 사용하여 로봇을 구성하였습니다.

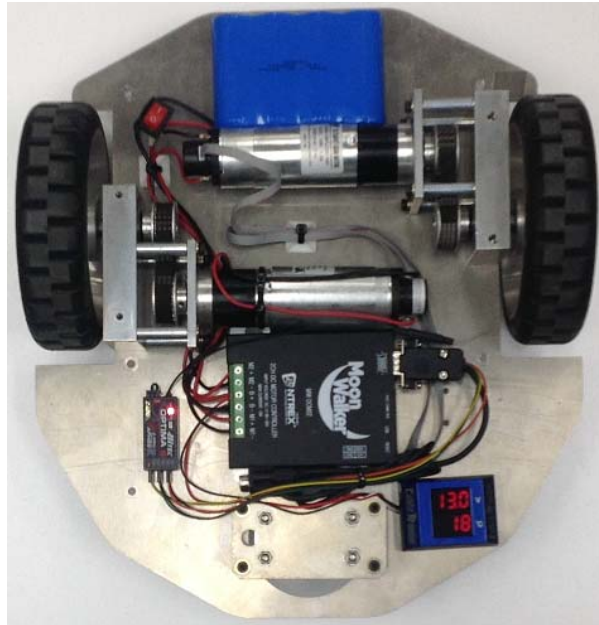


그림 9-22 2륜 차동 구동 로봇 실사

다음은 2륜 차동 구동 로봇의 제작에 사용되는 부품 목록입니다:

- 모터 바퀴 일체형 모듈: NTRexLAB 모바일 플랫폼용 바퀴셋
- 배터리: 12.8V 3000mAh 리튬인산철 배터리
- 전원 스위치: 로커 스위치
- 모터 제어기: MW-MDC24D200D
- 기타: 캐스터

추가로 모터 바퀴 일체형 모듈과 배터리, 제어기를 고정할 프레임을 가공해야 합니다.

9.5.2 2륜 차동 구동 로봇 기구학

2륜 차동 구동 로봇에 내려지는 구동 명령은 로봇 좌표계에서 전후진 속도와 회전 속도가 됩니다. 이 두 속도 명령을 좌우 바퀴에 분배하기 위해서는 플랫폼의 역기구학(inverse kinematics)을 구해야 합니다. 그리고 좌우 바퀴의 속도로부터 로봇의 이동 위치를 추정하기 위해서는 정기구학(forward kinematics)을 구해야 하고 전역 좌표계에서 데드레쿠닝(dead-reckoning) 해야 합니다.

역기구학과 정기구학을 구하기 위해 2륜 차동 구동 플랫폼의 구조를 그림 9-23과 같이 정합니다.

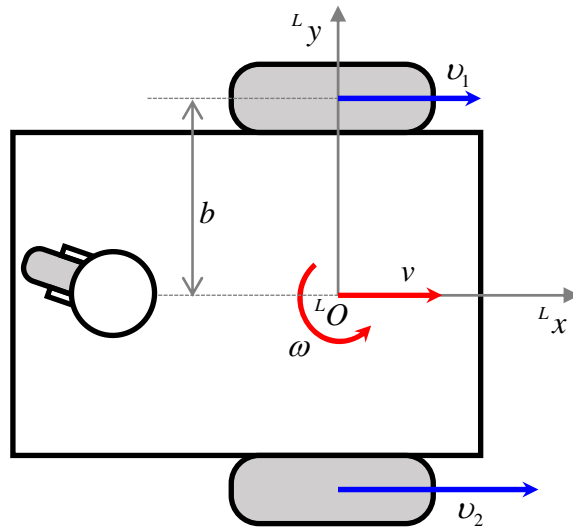


그림 9-23 2륜 차동 구동 로봇 구조

사용자가 로봇에 내린 전진속도(v)와 회전속도(ω) 명령은 다음 역기구학 식에 의해 좌우 바퀴의 속도(v_1, v_2)로 분배됩니다.

$$v_1 = v - b\omega$$

$$v_2 = v + b\omega$$

여기서 b 는 로봇 중심(${}^L O$)에서 바퀴까지 거리입니다.

바퀴의 속도(v)로부터 모터의 회전 속도(ω_m)는 다음과 같이 계산할 수 있습니다.

$$\omega_m = \frac{g}{r} v$$

여기서 r 은 바퀴의 반지름('Wheel Radius' 오브젝트)이고 g 는 감속기의 감속비('Gear Ratio' 오브젝트) 입니다.

로봇이 이동할 때 일정한 시간 간격(Δt)으로 엔코더의 변화량을 측정하여 좌우 바퀴의 속도(v_1, v_2)를 계산하고, 다시 다음 정기구학 식에 의해 로봇의 전후진 속도와 회전속도(v, ω)를 계산합니다.

$$v = \frac{(v_2 + v_1)}{2}$$

$$\omega = \frac{(v_2 - v_1)}{2b}$$

그리고 로봇의 속도를 오일러 적분(Euler integration)하여 전역 좌표계에서 로봇의 위치와 방위(x, y, θ)를 업데이트 합니다. 이를 데드레쿠닝(dead-reckoning) 이라 합니다.

$$x \leftarrow x + v\Delta t \cos \theta$$

$$y \leftarrow y + v\Delta t \sin \theta$$

$$\theta \leftarrow \theta + \omega\Delta t$$

이 방법은 로봇의 위치에 오차가 누적되어 로봇의 이동거리가 길어질수록 추정 위치의 정확도가 떨어집니다.

9.5.3 스크립트 작성

스크립트에서는 'Temp Value' 오브젝트를 통해 로봇의 구동 명령을 수신하거나 RC 조종기를 통해 구동 명령을 수신합니다. 작성된 스크립트 내에서 주석을 풀거나 묶어서 둘 중 하나의 구동 명령 입력을 선택합니다.

```
/*
    차동 바퀴형 이동 로봇을 전진속도와 각속도로 제어하고
    로봇의 위치(x, y, theta)를 dead-reckoning으로 추정함

    NTREX 기술연구소, 2014.1.10
*/

r = getv (_wheel_radius, 0);
b = getv (_axle_length, 0);
g = getv (_gear_ratio, 0);
enc = getv (_encoder_ppr, 1);

enc1_p = getv (_position, 1);
enc2_p = getv (_position, 2);

rps2rpm   = 60/(2*_PI);
pulse2met = 2*_PI*r/g/enc;

x = 0;
y = 0;
th = 0;

while (1) {
    sleep (10);

    // 아래 주석을 풀면 temp_value1과 temp_vale2로부터 로봇 구동 명령을 받아들임
    // v = getv (_temp_value,1)/1000;
    // w = getv (_temp_value,2)/1000;

    // 아래 주석을 풀면 펄스 입력 채널 1,2로부터 로봇 구동 명령을 받아들임
    v = getv (_pi_converted_value, 1);
    w = getv (_pi_converted_value, 2);

    // 전진 속도와 각속도를 로봇의 오른쪽 바퀴와 왼쪽 바퀴의 속도로 변환
```



```

vl = g/r*(v - w*b/2);
vr = g/r*(v + w*b/2);

// 로봇의 좌우 모터에 RPM 단위의 속도 명령을 내림
setv (_velocity_command, 1, vl*rps2rpm);
setv (_velocity_command, 2, vr*rps2rpm);

// 좌우 바퀴의 엔코더 값을 읽음
enc1 = getv (_position, 1);
enc2 = getv (_position, 2);
de1 = enc1 - enc1_p;
de2 = enc2 - enc2_p;
enc1_p = enc1;
enc2_p = enc2;

de1 *= pulse2met;
de2 *= pulse2met;

// 엔코더 변위를 로봇의 위치 변위 x, y, th로 변환
s = (de2 + de1)/2;
a = (de2 - de1)/b;

x += s*cos(th);
y += s*sin(th);
th += a;

setv (_temp_value,4,x*1000);
setv (_temp_value,5,y*1000);
setv (_temp_value,6,th*1000);
}

```

9.5.4 구동 테스트

Motor Control UI 유틸리티의 IO Monitoring 탭에서 Temp Values 항목의 값을 모니터링 합니다. Temp Values 항목의 Read 칼럼 값은 '**Temp Value**' 오브젝트의 현재 값을 표시합니다.

Temp Values	Read	Write
<input type="checkbox"/> Temp Value 1		
<input type="checkbox"/> Temp Value 2		
<input type="checkbox"/> Temp Value 3		
<input checked="" type="checkbox"/> Temp Value 4	-3265	
<input checked="" type="checkbox"/> Temp Value 5	-4683	
<input checked="" type="checkbox"/> Temp Value 6	4065	
<input type="checkbox"/> Temp Value 7		
<input type="checkbox"/> Temp Value 8		

그림 9-24 로봇의 위치 모니터링

스크립트는 '**Temp Value**' 오브젝트의 4,5,6 채널에 각각 로봇의 위치(x, y)와 방위를 mm 단위와 milli rad 단위로 표시합니다.

만일 로봇에 RC 조종기가 연결되어 있다면, 펄스 입력 채널로부터 로봇 구동 명령을 받아들이도

록 스크립트를 수정하여 빌드하고 다운로드하여 실행 합니다. 그리고 로봇을 PC와 분리하고 RC 조종기로 조종해 봅니다.

9.6 전방향 로봇 운전

실내용 로봇으로 주로 사용되는 2륜 차동 구동 로봇은 평면에서 이동 능력이 전후진과 회전으로 제한되어(좌우 이동이 불가능) 유연한 구동이 어렵고, 목적지에 특정한 자세로 도착하기 위해서는 제어가 복잡해 집니다. 이런 점을 극복하기 위해 여러 전방향(omni-directional) 이동로봇 플랫폼들이 개발되었으며, 특히 메카넘 휠(Mecanum wheel)을 사용하는 플랫폼이 잘 알려져 있습니다.

메카넘 플랫폼은 모든 방향(전후, 좌우, 회전)으로 움직임이 가능하기 때문에 좁은 공간에서도 자유롭게 이동 및 회전이 가능합니다.

이 절에서는 메카넘 휠로 만든 전방향 로봇에 속도 명령을 내리고 데드레쿠닝(dead-reckoning)으로 로봇의 위치를 추정하는 스크립트를 작성합니다.

9.6.1 전방향 로봇 개요

메카넘 휠은 그림 9-25과 같이 바퀴 주변에 비스듬히 작은 아이들 롤러를 여러 개 장착하여 바퀴가 회전하는 방향이 아닌 대각선 방향으로 움직이도록 고안된 바퀴입니다.



그림 9-25 메카넘 휠

전방향 로봇에는 그림 9-26과 같이 4개의 메카넘 휠이 장착됩니다. 이 로봇은 2차원 평면 공간에서 주행의 제약이 없이 전후, 좌우, 회전의 3자유도 운동이 가능하여 직관적인 조작이 가능합니다.

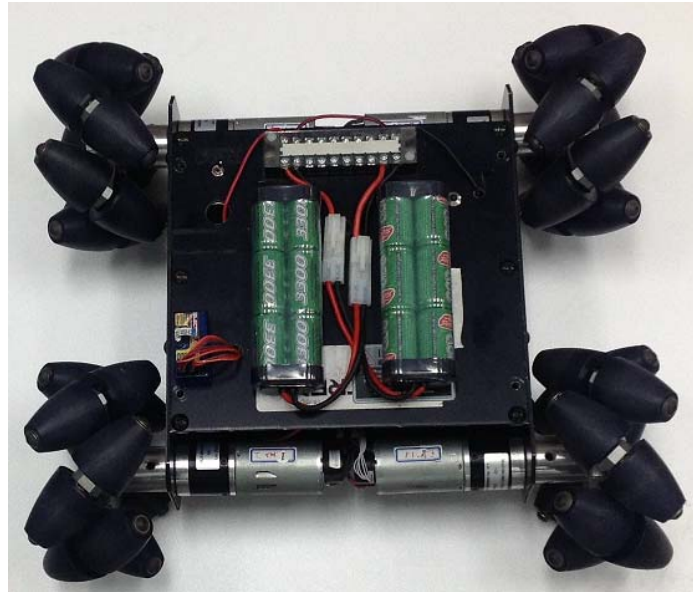


그림 9-26 메카넴 휠을 사용하여 제작된 메카넴 플랫폼

메카넴 휠 자체는 구조가 다소 복잡하여 설계와 제작이 쉽지 않습니다. 시중에 판매되는 메카넴 휠을 구매하여 플랫폼을 구성하는 것이 좋습니다.

네 개의 메카넴 휠을 각각 따로 구동하기 위해 4개의 모터가 필요합니다. 그리고 각 모터는 각 바퀴에 직결 됨으로 다른 로봇에 비해 제작이 용이합니다. 하지만 전방향 로봇을 제대로 구동하기 위해서는 네 바퀴가 항상 지면에 닿아 있어야 하므로 구동축에 서스펜션과 같은 장치가 필요합니다.

9.6.2 전방향 로봇 기구학

전방향 로봇에 내려지는 구동 명령은 로봇 좌표계에서 전후진 속도와 좌우 속도, 회전 속도가 됩니다. 이 세 속도 명령을 4개의 바퀴에 분배하기 위해서는 플랫폼의 역기구학(inverse kinematics)을 구해야 합니다. 그리고 4개의 바퀴 속도로부터 로봇의 이동 위치를 추정하기 위해서는 정기구학(forward kinematics)을 구해야 하고 전역 좌표계에서 데드레쿠닝(dead-reckoning) 해야 합니다.

역기구학과 정기구학을 구하기 위해 메카넴 플랫폼의 구조를 그림 9-27에서와 같이 정합니다.

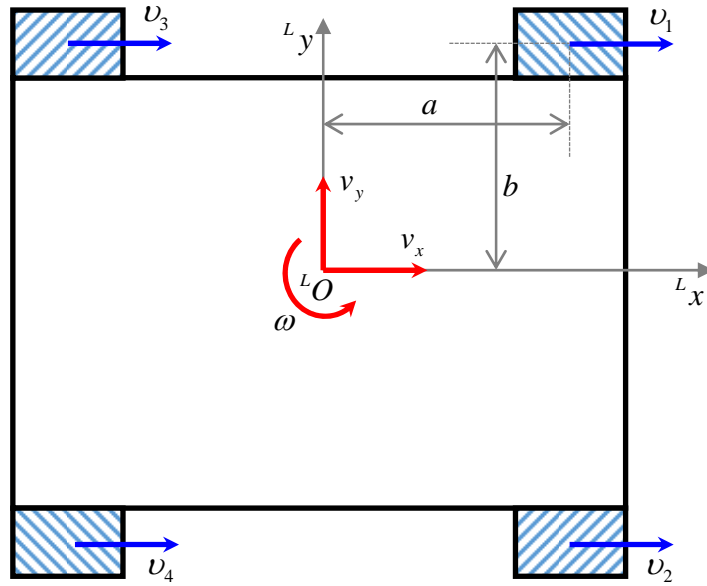


그림 9-27 메카넘 플랫폼의 구조

사용자가 로봇에 내린 전후진 속도와 좌우 속도, 회전 속도(v_x, v_y, ω) 명령은 다음 역기구학 식에 의해 4 바퀴의 속도(v_1, v_2, v_3, v_4)로 분배됩니다.

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -a-b \\ 1 & 1 & a+b \\ 1 & 1 & -a-b \\ 1 & -1 & a+b \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

여기서 a 와 b 는 로봇 중심에서 바퀴까지의 수평 거리와 수직 거리입니다.

바퀴의 속도(v)로부터 모터의 회전 속도(ω_m)는 다음과 같이 계산할 수 있습니다.

$$\omega_m = \frac{g}{r} v$$

여기서 r 은 메카넘 휠의 반지름이고 g 는 감속기의 감속비입니다.

로봇이 이동할 때 일정한 시간 간격(Δt)으로 엔코더의 변화량을 측정하여 4 바퀴의 속도(v_1, v_2, v_3, v_4)를 계산하고, 다시 다음 정기구학 식에 의해 로봇의 속도(v_x, v_y, ω)를 계산합니다.

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -c & c & -c & c \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}, \quad c = \frac{1}{a+b}$$

로봇 좌표계로 표시되는 로봇의 속도(v_x, v_y, ω)는 전역 좌표계에서의 속도($^w v_x, ^w v_y, \omega$)로 다음과 같이 변환됩니다 (θ 는 전역 좌표계에서 로봇의 방향).

$$\begin{bmatrix} {}^w v_x \\ {}^w v_y \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

그리고 로봇의 속도를 오일러 적분(Euler integration)하여 전역 좌표계에서 로봇의 위치와 방위 (x, y, θ) 를 업데이트 합니다.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \leftarrow \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \Delta t \begin{bmatrix} {}^w v_x \\ {}^w v_y \\ \omega \end{bmatrix}$$

9.6.3 전방향 로봇 배선

전방향 로봇은 4개의 모터를 사용하므로 2개의 듀얼채널 제어기가 사용됩니다. 그림 9-28과 같이 이 중 하나는 마스터가 되고 하나는 슬레이브가 됩니다.

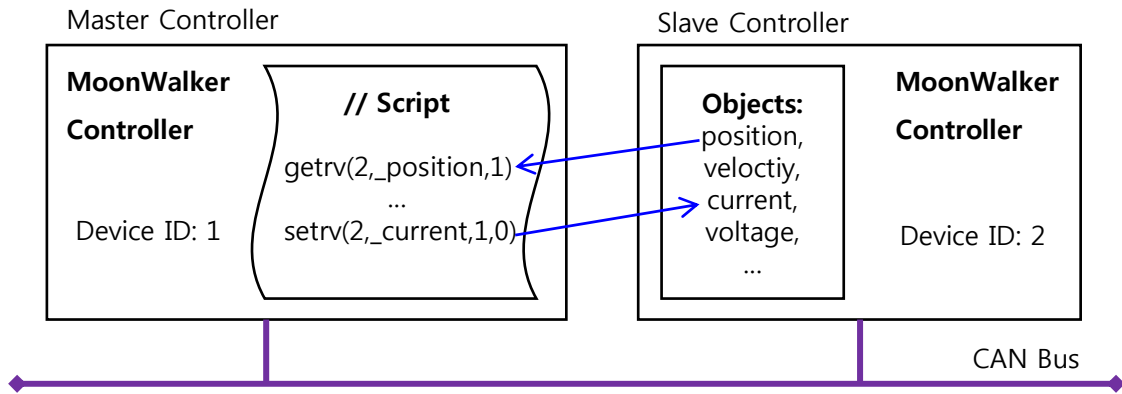


그림 9-28 제어기의 마스터 슬레이브 연결 구조

마스터와 슬레이브 제어기는 CAN으로 연결하고 마스터의 장치 ID를 1로, 슬레이브의 장치 ID를 2로 설정합니다. 마스터는 스크립트 프로그램으로 슬레이브의 오브젝트를 액세스 하여 전체 시스템을 제어하게 됩니다.

그림 9-29와 같이 제어기와 배터리, 모터, RC 수신기를 배선합니다. 이 때, 전력선들이 엔코더 신호선과 동일한 경로를 따라 배선되지 않도록 주의합니다.

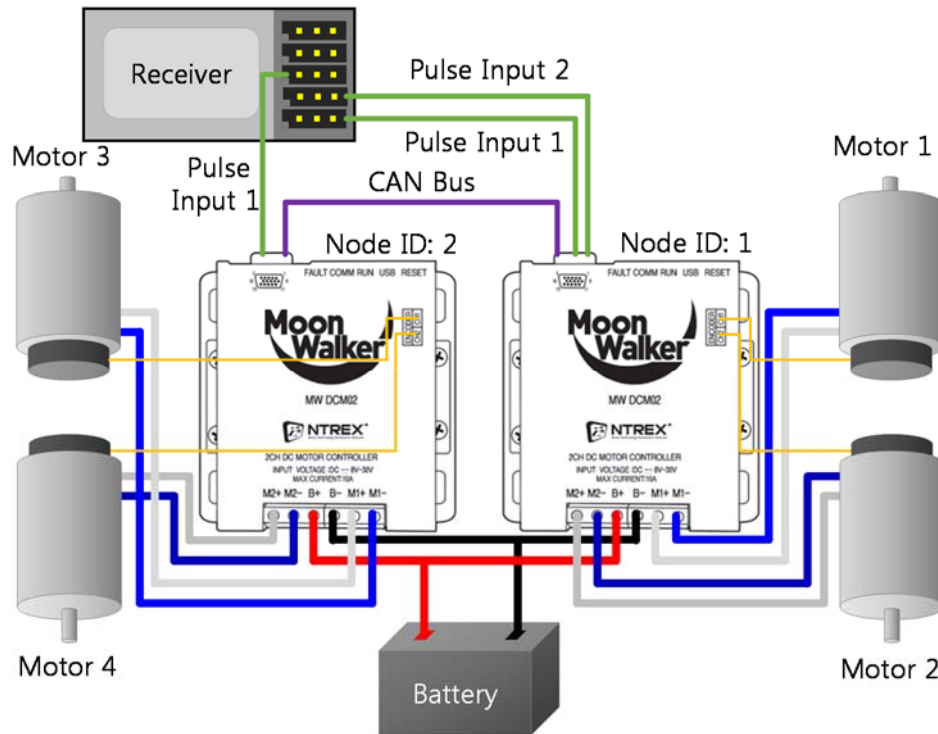


그림 9-29 메카넘 플랫폼 구성

듀얼 채널 제어기는 최대 2개의 펄스 입력 채널이 있습니다. 하지만 전후, 좌우, 회전 속도 명령을 받아 들이기 위해서는 3개의 펄스 입력 채널이 필요합니다. 그래서 전후, 좌우 속도 명령은 마스터 제어기에서 받아들이고, 회전 명령은 슬레이브 제어기에서 받아들이도록 RC 수신기를 연결합니다.

9.6.4 스크립트 작성

다음은 메카넘 플랫폼에 사용된 두 개의 제어기 중 마스터에서 실행되는 스크립트 예제입니다.

```
/*
    SCRIPT Example 7: ex7_4wheel_omni.scr
    메카넘 플랫폼 로봇을 x, y-속도와 각속도로 제어하고
    로봇의 위치(x, y, theta)를 dead-reckoning으로 추정함

    NTREX 기술연구소, 2014.1.10
*/

// 로봇의 위치
x = 0;
y = 0;
th = 0;

a = 0.235/2; // 앞뒤 바퀴간 거리 / 2
b = 0.315/2; // 좌우 바퀴간 거리 / 2
```

```
ab = a+b;

r = getv (_wheel_radius, 0);          // 바퀴 반지름
// b = getv (_axle_length, 0);
g = getv (_gear_ratio, 0);           // 감속비

vel_scale = 1;          // x, y 속도에 곱해지는 스케일, 최고 1m/s
ang_scale = 2;          // 회전 속도에 곱해지는 스케일, 최고 2rad/s

enc = getv (_encoder_ppr, 1);         // 엔코더 해상도
enc1_p = getv (_position, 1);
enc2_p = getv (_position, 2);
enc3_p = getrv (2, _position, 1);
enc4_p = getrv (2, _position, 2);

met2rpm  = 60/(2*_PI)*g/r; // 속도명령(m/s)을 RPM으로 바꾸는 scale factor
pulse2met = 2*_PI*r/g/enc; // 엔코더 펄스를 meter로 바꾸는 scale factor

while (1) {
    sleep (1);
    // Pulse Input으로부터 로봇의 구동 명령을 읽음(vel_x, vel_y, vel_theta)
    vel_x = getv(_pi_converted_value, 1);
    vel_y = getv(_pi_converted_value, 2);
    vel_theta = -getrv(2, _pi_converted_value, 1);

    // 구동 명령의 스케일 변환
    vx = vel_scale*vel_x;
    vy = vel_scale*vel_y;
    w  = ang_scale*vel_theta;

    // 전방향 로봇의 각 바퀴 속도를 계산
    v1 = vx - vy - ab*w;
    v2 = vx + vy + ab*w;
    v3 = vx + vy - ab*w;
    v4 = vx - vy + ab*w;

    // 각 바퀴에 속도 명령을 내림
    setv (_velocity_command, 1, v1*met2rpm);
    setv (_velocity_command, 2, v2*met2rpm);
    setrv (2, _velocity_command, 1, v3*met2rpm);
    setrv (2, _velocity_command, 2, v4*met2rpm);

    // 네 바퀴의 엔코더 값을 읽음
    enc1 = getv (_position, 1);
    enc2 = getv (_position, 2);
    enc3 = getrv (2, _position, 1);
    enc4 = getrv (2, _position, 2);

    de1 = enc1 - enc1_p;
    de2 = enc2 - enc2_p;
    de3 = enc3 - enc3_p;
    de4 = enc4 - enc4_p;
```

```

enc1_p = enc1;
enc2_p = enc2;
enc3_p = enc3;
enc4_p = enc4;

de1 *= pulse2met;
de2 *= pulse2met;
de3 *= pulse2met;
de4 *= pulse2met;

// 엔코더 변위를 로봇의 위치 변위 x, y, th로 변환
lvx = ( de1 + de2 + de3 + de4)/4;
lvy = (-de1 + de2 + de3 - de4)/4;
lw  = (-de1 + de2 - de3 + de4)/(4*ab);

x += cos(th)*lvx - sin(th)*lvy;
y += sin(th)*lvx + cos(th)*lvy;
th+= lw;

setv (_temp_value,4,x*1000);
setv (_temp_value,5,y*1000);
setv (_temp_value,6,th*1000);
}

```

9.6.5 구동 테스트

Motor Control UI 유틸리티의 IO Monitoring 탭에서 Temp Values 항목의 값을 모니터링 합니다. 'Temp Values'의 Read 칼럼 값은 '**Temp Value**' 오브젝트의 현재 값을 표시합니다.

Temp Values	Read	Write
<input type="checkbox"/> Temp Value 1		
<input type="checkbox"/> Temp Value 2		
<input type="checkbox"/> Temp Value 3		
<input checked="" type="checkbox"/> Temp Value 4	2526	
<input checked="" type="checkbox"/> Temp Value 5	1260	
<input checked="" type="checkbox"/> Temp Value 6	4065	
<input type="checkbox"/> Temp Value 7		
<input type="checkbox"/> Temp Value 8		

그림 9-30 로봇의 위치 모니터링

스크립트는 '**Temp Value**' 오브젝트의 4,5,6 채널에 각각 로봇의 위치(x, y)와 방위를 mm 단위와 milli rad 단위로 표시합니다.

마지막으로 로봇에 RC 조종기를 연결하고 로봇을 PC와 분리하여 RC 조종기로 조종해 봅니다.

10 관련 자료

아래 홈페이지에서 MoonWalker 관련 문서와 예제 그리고 동영상 자료를 확인하실 수 있습니다.

- 엔티렉스:
<http://www.ntrexgo.com/>
- 디바이스마트:
<http://www.devicemart.co.kr/>
- MoonWalker 소개:
<http://www.ntrexgo.com/moonwalker-%EC%86%8C%EA%B0%9C>
- MoonWalker 판매페이지:
<http://devicemart.co.kr/goods/brand.php?seq=1475>
- MoonWalker UI Utility:
<http://www.ntrexgo.com/archives/19482>
- MoonWalker 자료:
<http://www.ntrexgo.com/archives/category/ntrex-lab/moonwalker>
- MoonWalker 예제:
http://www.ntrexgo.com/archives/category/ntrex-lab/moonwalker_applications
- MoonWalker 액세서리:
http://www.ntrexgo.com/archives/category/moonwalker_accessory

11 문서 변경 이력

Data	Version	Charges
2014. 01. 13	1.00	- Application Notes 1 첫 출시
2014. 02. 04	1.10	- Application Notes 1을 Application Notes로 이름 변경 - Application Notes 내용 수정 및 추가