



# STELLA B2

## MOBILE ROBOT / PLATFORM

### USER MANUAL

<펌웨어 버전 2.00>



## < 머릿말 >

먼저, 저희 (주)엔티렉스의 STELLA B2 모델을 구입해 주셔서 감사합니다. 본 제품을 사용하기 전에, 본 사용 설명서에 명시되어 있는 제품에 관한 내용과 주의사항 및 설치 방법을 반드시 숙지하신 후에 사용하여 주십시오. 본 사용 설명서를 다 읽으신 후에도 언제나 참고할 수 있도록 잘 보이는 곳에 보관하여 주십시오. 본 사용 설명서에 명시되어 있지 않은 내용은 언제든지 저희 엔티렉스 로봇 연구소 (<http://www.ntrexlabs.co.kr>)에 문의 바랍니다.

## < 목차 >

1. STELLA B2 개요.....	P.3
2. 안전에 관한 유의 사항.....	P.15
3. 충전 및 작동 방법.....	P.17
4. RS232 구동 명령 커맨드 .....	P.22
5. CAN 구동 명령 커맨드 .....	P.45
6. 제품의 고장 진단.....	P.84
7. 제품의 보증.....	P.85

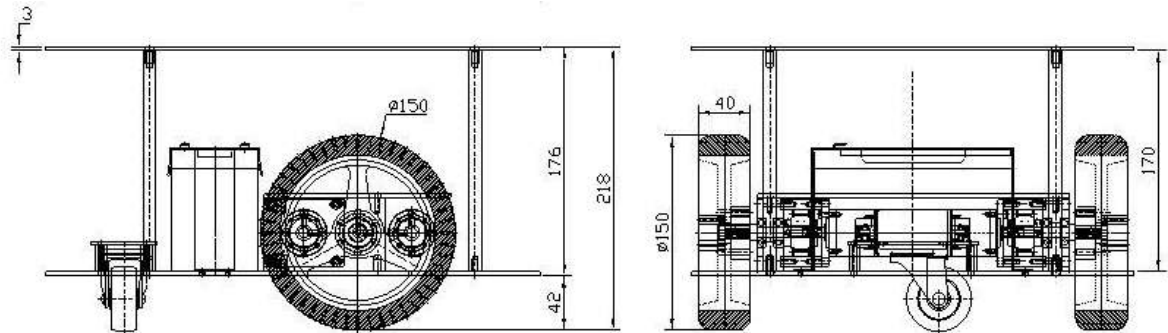
## 1. STELLA B2 개요

● 'STELLA'는 실내환경에서 각종 주행 알고리즘 개발 및 테스트를 하는데 있어서 활용이 가능한 이동형 로봇 플랫폼입니다. 다양한 명령어로 인해 사용자의 활용도가 증가하도록 구성되었습니다. 또한 보다 정확한 주행성능을 확보하였습니다.

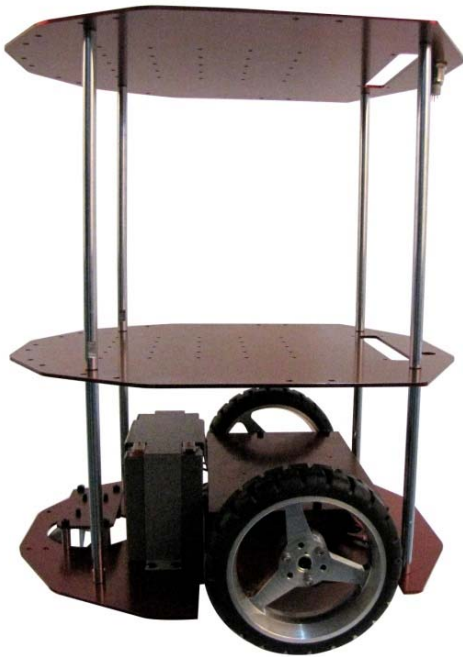
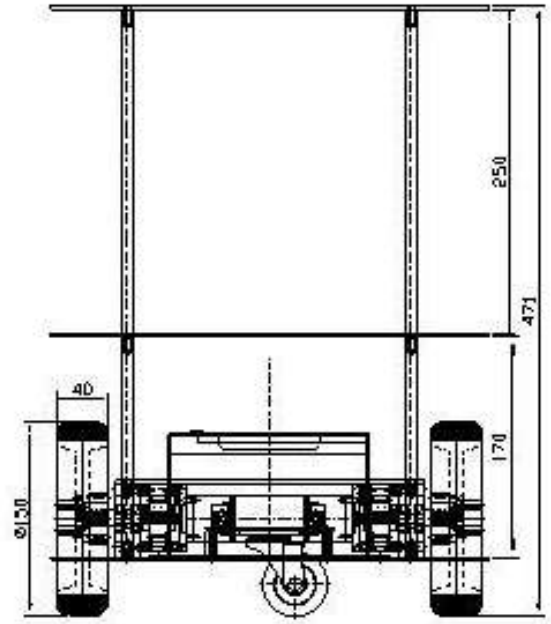
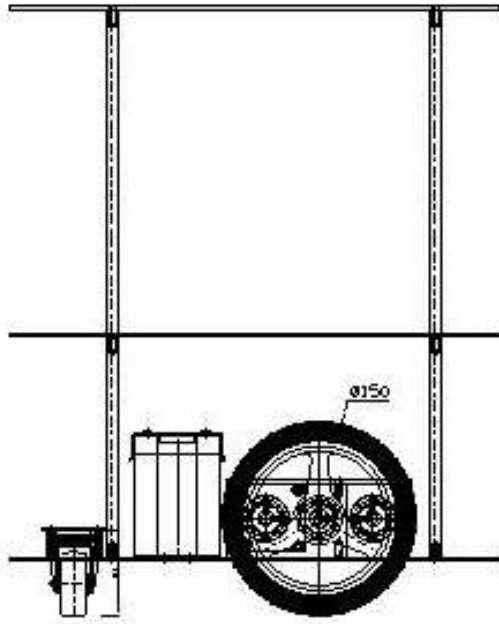
### ● STELLA B2 특징

- 속도 명령 : 각 휠은 지시 받은 속도로 주행합니다.
- 위치 명령 : 로봇의 관점으로 지시 받은 회전각도, 이동거리 만큼 회전 및 주행시킬 수 있습니다. (적재 하중이 없는 상태에서 1M 이동시 약 5cm 내외의 오차범위(5%)를 가지고 있으며, 제자리에서 360° 회전 시 약 5° 내외의 오차범위를 가지고 있습니다)
- 다양한 파라미터 설정이 가능합니다. (p, I, d, il, scale factor 등...)
- 기본 모델인 STELLA B2에 다음과 같이 옵션으로 중간 판을 추가로 올릴 수도 있습니다. (노트북이나 부가적인 장비를 장착할 수 있는 여유 공간이 많습니다)

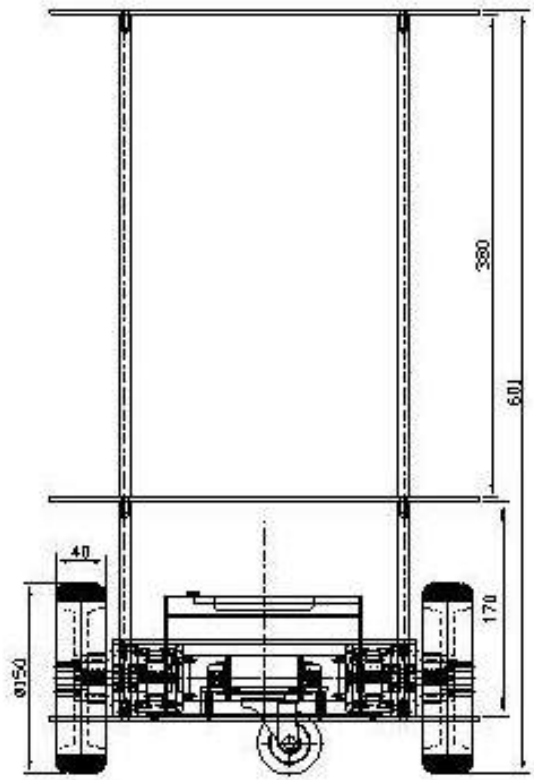
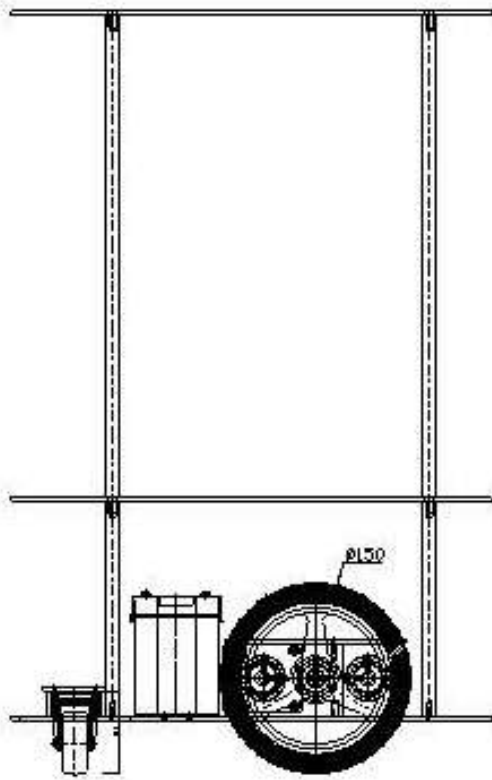
### ● STELLA B2 치수 및 외관



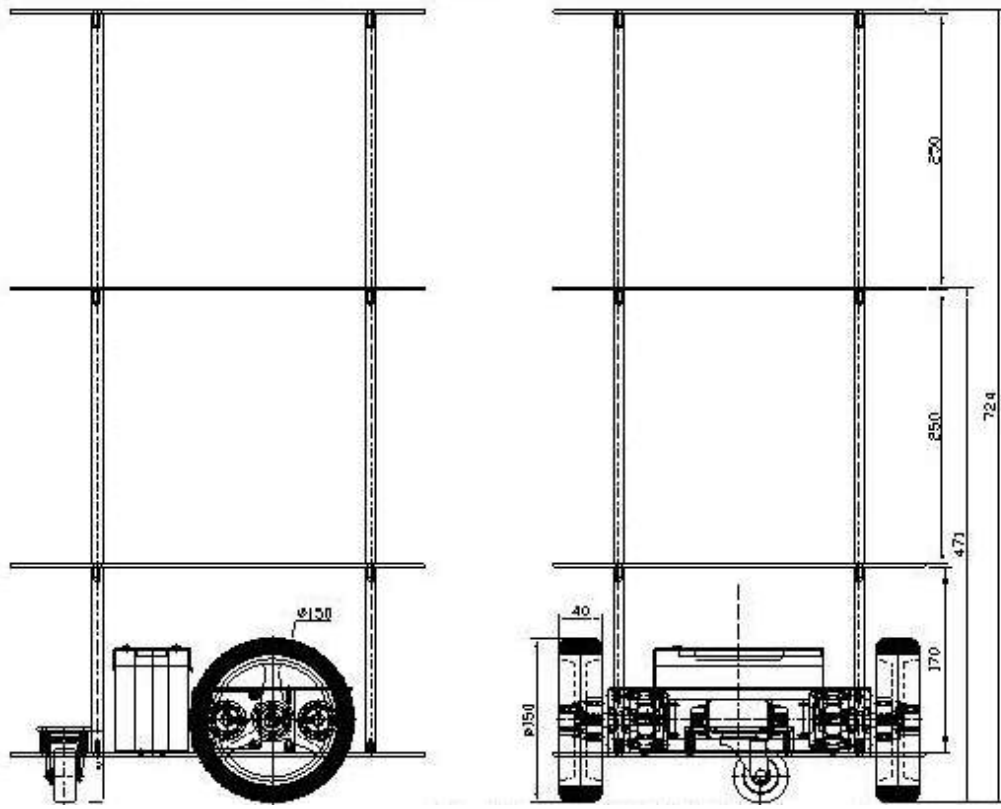
(STELLA B2) - 기본형



(STELLA B2에 250mm의 높이로 상판을 추가한 모습) - 옵션1

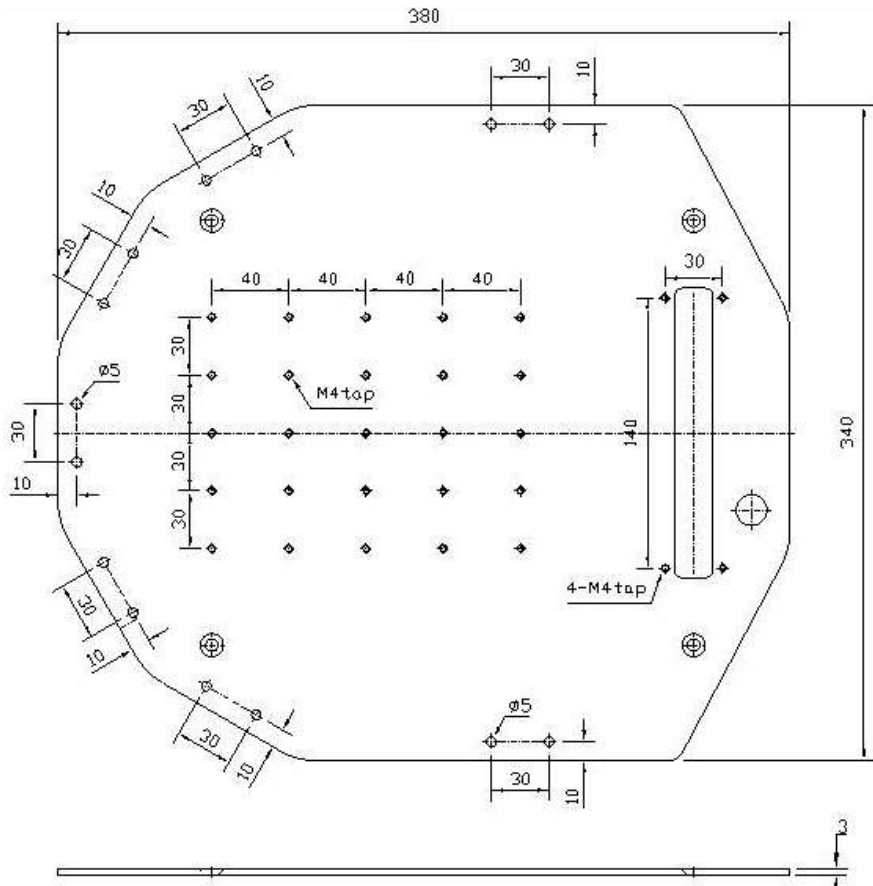


(STELLA B20에 380mm의 높이로 상판을 추가한 모습) - 옵션2

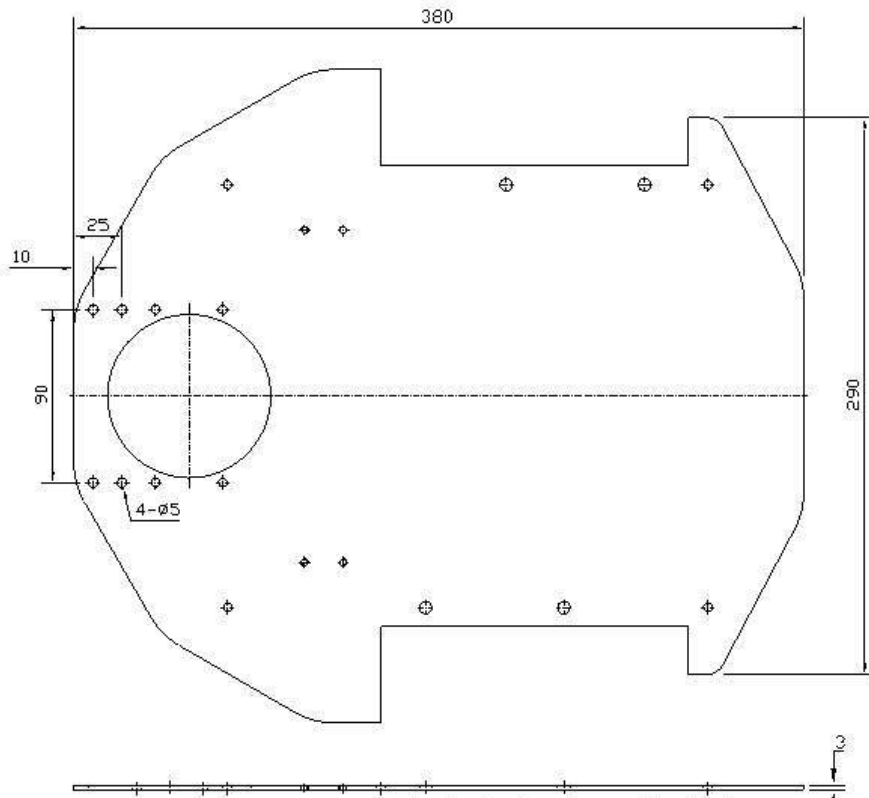




(STELLA B2에 250mm의 높이로 상판2단 추가한 모습) - 옵션3



(STELLA B2의 상판 치수)

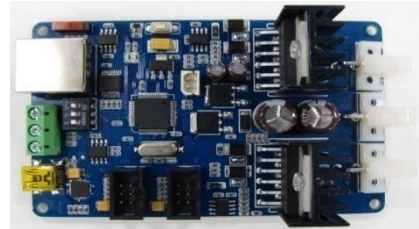
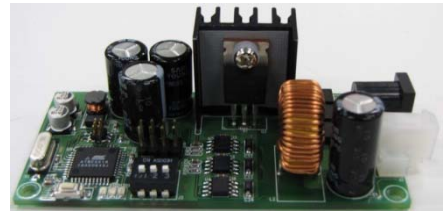




(STELLA B2의 하판 치수)

● STELLA B2 구성품

① STELLA B2 (배터리, 충전보드, 모션드라이브가 탑재되어있음)



② USB 케이블



③ 충전 어댑터



#### ④ 충전 어댑터 A/C전원 케이블



#### ● STELLA B2 제원

- 배터리 : 12V(7A) 연속전지
- 재 질 : 알루미늄, 스틸(기동 봉)
- 무 게 : 8.5Kg
- 적재용량 : 7Kg
- 속 도 : 5.2km/h (Max)
- 바퀴둘레 : 47.25cm
- 두 바퀴간의 거리 : 28.9cm
- 모 터 : 10w DC Motor (감속 14:1)
- Encoder CPR : 256
  
- 인터페이스 : RS232통신 제어
- 알람 기능 : 과부하, 과전압, 저전압 알람
- 동작 온도 : 0°C ~ 50°C (동결 되지 않는 곳)
- 습도 : 85% 이하 (이슬이 맺히지 않는 곳)

#### ● STELLA B2에 사용된 모터 스펙

##### - Motor Technical Data - 1

Rated voltage	12V
No-load speed	3000r/min
No-load current	150mA
Rated speed	2420r/min
Rated torque	40mN.m
Output power	10W
Rated current	1.3A
Stall torque	180mN.m
Stall current	4.4A

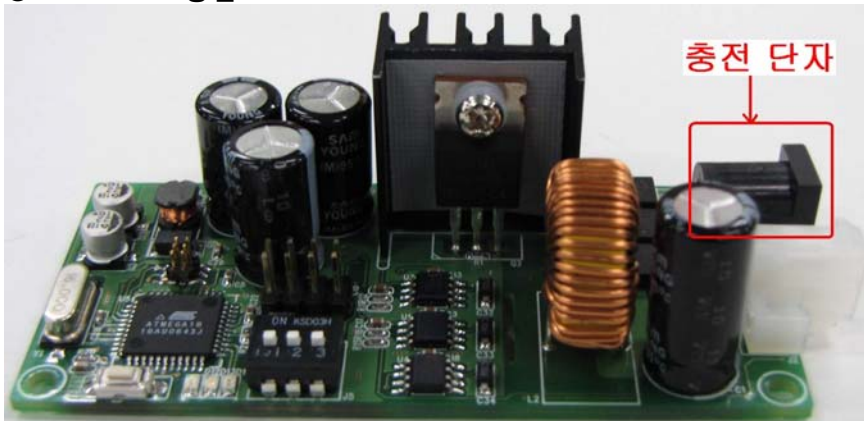
- Motor Technical Data - 2

Terminal resistance	4.6 $\Omega$
Terminal inductance	2.067mH
Torque constant	0.0307 (Nm/A)
B-EMF constant	0.0313 (V rad/sec)

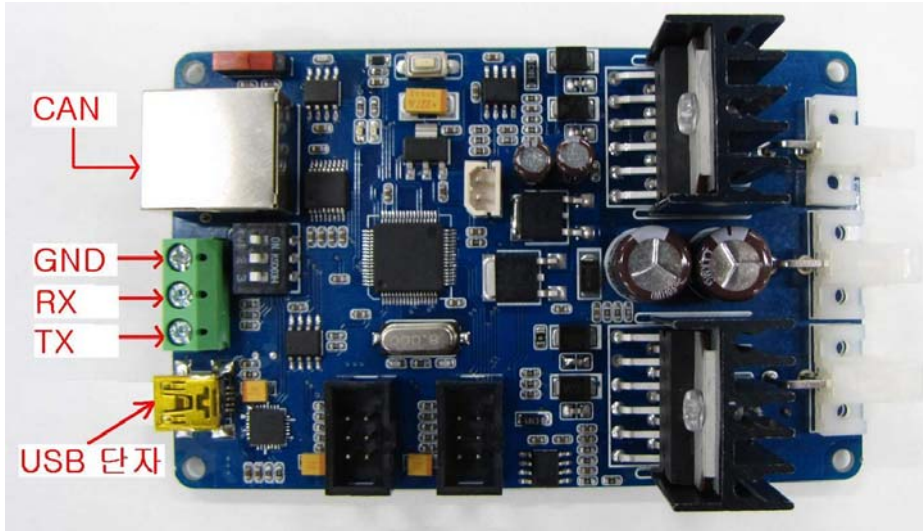
- Gear motor Technical Data

Reduction ratio	1:14
Number of gear trains	2
No-load speed	214r/min
Rated speed	172r/min
Rated torque	0.5N.m
Max.permmissible load in a short time	3N.m

● STELLA B2 충전보드



● STELLA B2 모션 드라이브 보드



- USB 단자를 통해 PC와의 연결을 손쉽게 할 수 있습니다.

(Silicon Laboratories에서 제작한 USB to UART Bridge 칩인 'CP2102'를 사용했기 때문에 해당 드라이버를 PC에 설치 하셔야 합니다. 드라이버는 <http://www.silabs.com>에서 다운로드 받으실 수 있습니다)

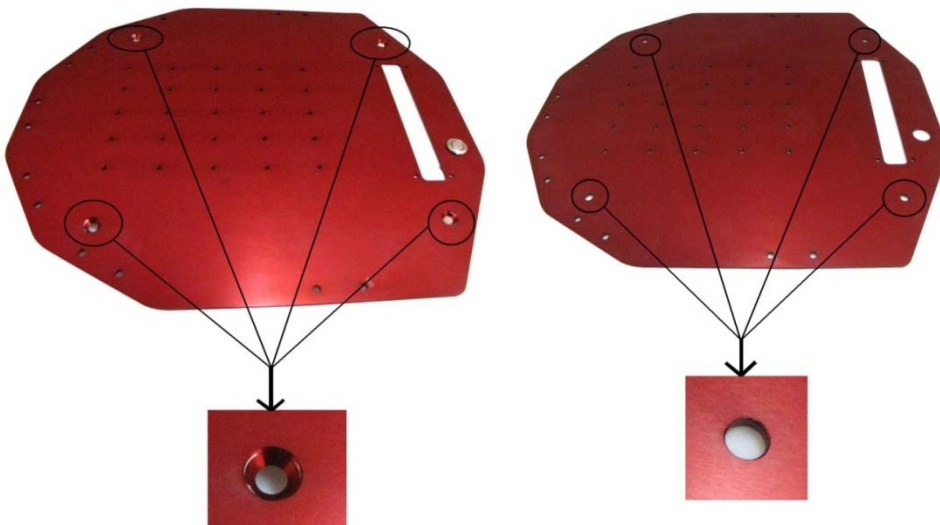
- GND, RX, TX를 통해 다른 임베디드 장비와 RS232통신이 가능합니다. (모션 드라이브의 RX, TX는 라인드라이버를 거친 시그널 입니다. TTL 레벨이 아닙니다)

- CAN 통신(RJ45단자)으로 STELLA B2를 구동시킬 수 있습니다.

● STELLA B2 옵션형 조립방법 (중간 판은 옵션으로 별도판매 제품입니다)

- STELLA B2은 기본형 모양으로 조립되어 배송됩니다. 중간판 및 상판 조립은 사용자가 하도록 구성되어있습니다.

- STELLA B2의 상판과 중간 판은 기둥(봉)이 연결되는 부분이 다릅니다. 아래 그림처럼 홈 모양으로 상판과 중간 판을 구별하면 됩니다..

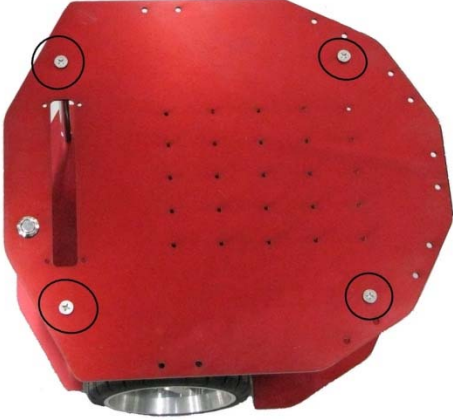


(상판 기둥 홈모양)

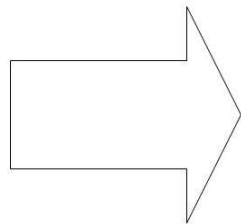
(중간판 기둥 홈 모양)

- STELLA B2 옵션형 조립순서

① 'STELLA B2'의 TOP의 접시머리볼트(4개)를 풀습니다.



② 접시머리볼트가 박혀있던 부분에 동봉된 기둥(봉)을 홈에 맞춰서 끼운 후 시계방향으로 돌려줍니다.



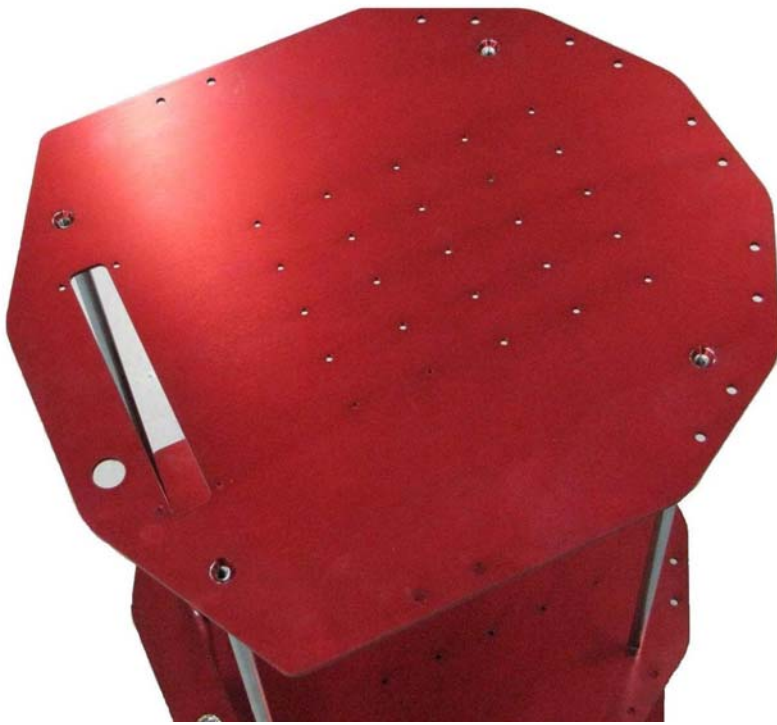
③ 스패너를 기둥(봉)의 홈에 끼운 후 시계방향으로 조여서 기둥(봉)을 단단히 고정시킵니다.



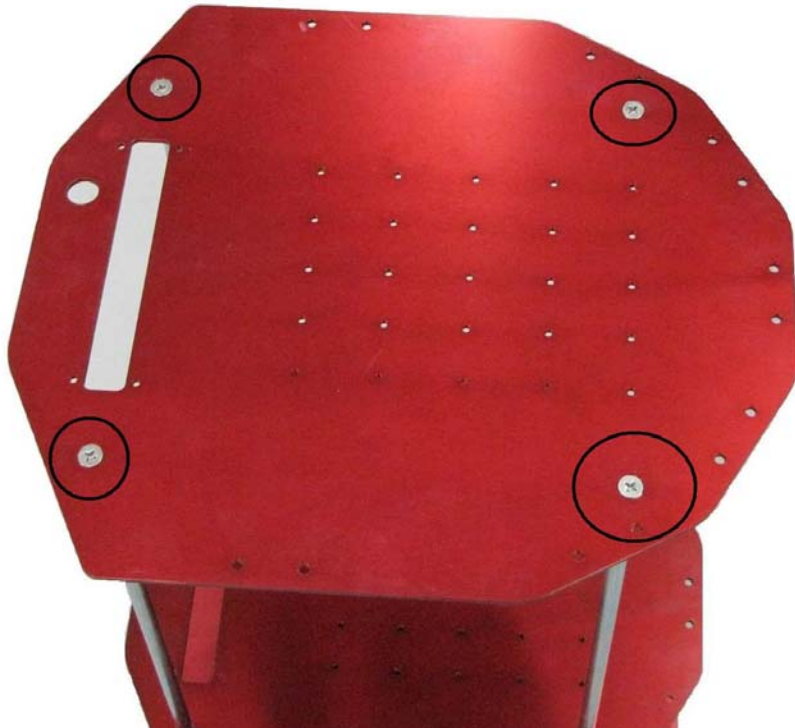
④ 기둥(봉) 4개를 모두 조립한 모습



⑤ 상판을 기둥(봉) 홀에 맞춰 올려놓습니다





⑥ 처음 풀어놓았던 나사(접시머리)로 상판을 고정시켜줍니다.



## 2. 안전에 관한 유의 사항.

제품을 사용하기 전에 안전상 유의점을 충분히 숙지해야 사고나 화재를 예방 할 수 있습니다. 명시되어 있는 사항을 꼭 숙지하신 후 사용 하여 주십시오.

 **경고** 부적절한 사용에 의해 위험한 상황이 발생할 수 있으며, 인명상의 사망이나 중상 또는 재산상의 피해를 입을 수도 있습니다..

 **주의** 부적절한 사용에 의해 위험한 상황이 발생할 수 있으며, 인명상의 경상 이나 재산상의 피해를 입을 수도 있습니다.

 **경고**

- 폭발성 물질이나 인화성 물질 또는 부식성 물질, 물이 닿는 장소에서는 사용하지 마십시오.



화재나 제품 파손의 원인이 될 수 있습니다.

- 전원이 인가된 상태에서 이동하거나 설치, 접속, 점검 작업을 하지 마십시오. 감전 사고나 기기 파손의 원인이 될 수 있습니다.
- 반드시 설치나 접속 과정은 본 사용 설명서를 충분히 습득한 전문가에 의해 시행 되어야 합니다. 화재나 감전의 원인이 될 수 있습니다.
- 제품 및 제품의 부속품을 임의로 분해, 개조하지 마십시오. 제품의 오작동으로 인해 감전 사고나 화재가 발생 할 수 있습니다.
- 탑재된 모션 드라이버의 전원은 반드시 정격 범위 내의 전원을 사용하여 주십시오. 화재의 원인이 될 수 있습니다.
- 설치 시 케이블을 무리하게 당기거나 구부릴 경우 피복이 벗겨 질 수 있으니 유의 해 주십시오. 화재의 원인이 될 수 있습니다.
- 점검이나 수리는 반드시 저희 엔티렉스의 기술 개발 연구소에 문의하여 주십시오. 전문적인 기술자 외에 점검이나 수리는 허용하지 않습니다. 제품의 고장 및 사고의 원인이 될 수 있습니다.

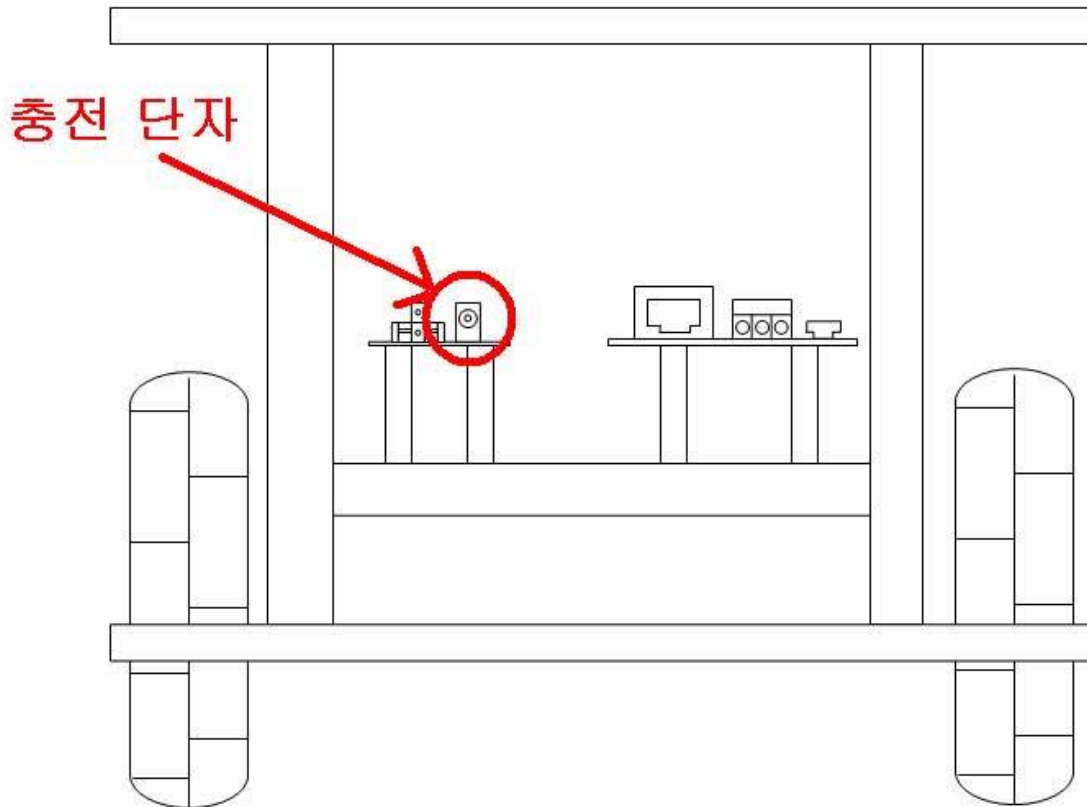


- 모션 드라이버의 한계 값을 초과해서 사용하지 마십시오. 제품의 고장이나 부상의 원인이 될 수 있습니다.

### 3. 충전 및 작동 방법

#### ■ 충전

STELLA B2는 충전을 위한 보드가 탑재되어 있으며 이 충전보드를 이용해 배터리 충전이 가능합니다. 아래 그림과 같이 STELLA B2 후면에서 왼쪽에 위치한 보드가 충전보드이며 이 보드에 충전을 위한 단자(검은색)가 있습니다.

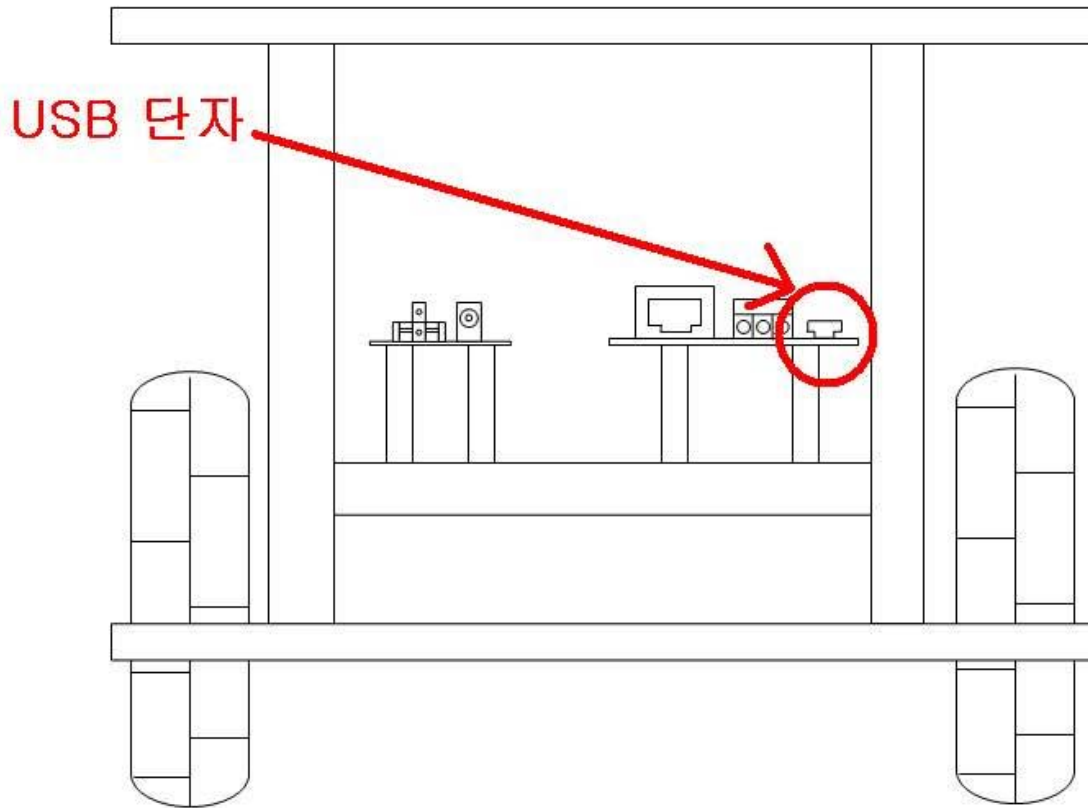


**충전은 반드시 아래의 순서대로 하시기 바랍니다.**

1. STELLA B2의 전원(상판 스위치)을 OFF 시킵니다.
2. 충전 어댑터의 A/C 전원 케이블을 확인한다. 만약 케이블이 전원 콘센트에 꽂혀있다면 전원을 콘센트에서 분리시킵니다.
3. 충전 어댑터의 동그란 부분을 위 그림의 충전단자에 꽂습니다.
4. 충전 어댑터의 A/C 전원 케이블을 전원 콘센트에 꽂습니다.
5. 충전이 시작되면 충전보드의 붉은색 LED가 점등되며, 충전이 완료되면 충전보드의 파란색 LED가 점등되면서 붉은색 LED는 OFF 됩니다.

#### ■ 작동 방법

- ① STELLA B2는 탑재되어 있는 모션 드라이브의 USB단자를 통해 구동 명령을 입력 받습니다. 아래 그림과 같이 STELLA B2 후면에서 오른쪽에 위치한 보드가 모션 드라이브이며 이 보드에 명령 입력을 위한 USB단자가 있습니다.

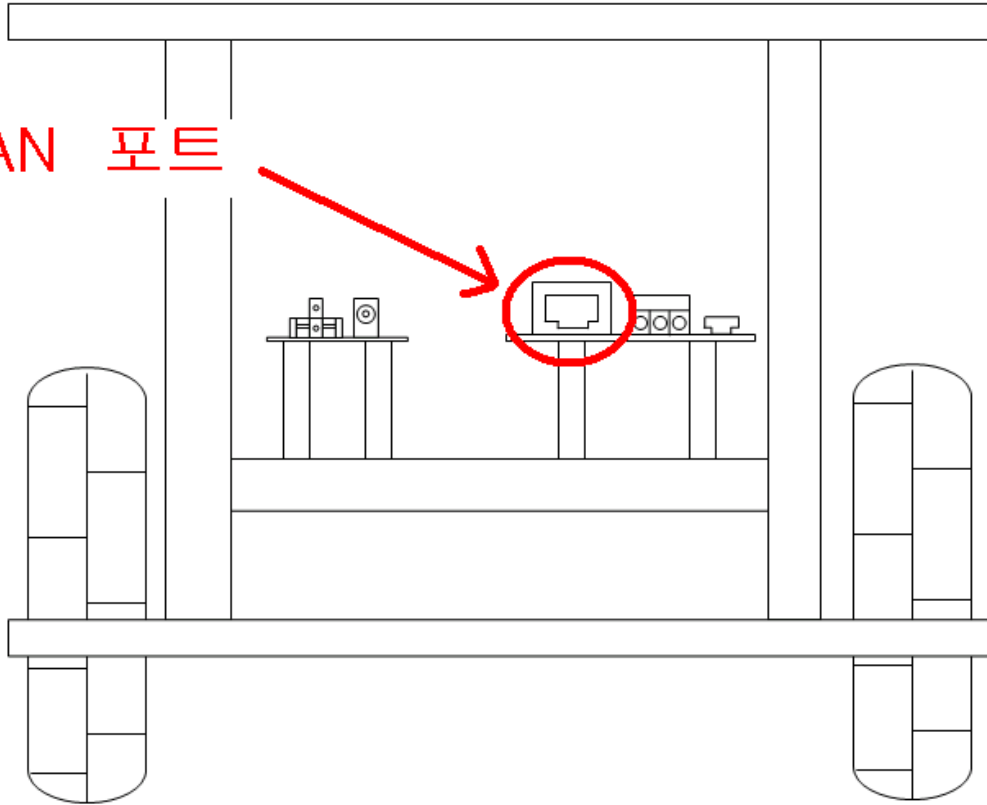


이 USB단자를 통해 PC와 연결하여 구동 명령을 입력할 수 있습니다. 구동 명령은 속도명령, 위치명령 및 로봇의 파라미터 설정/읽기 등이 있습니다. 구동 명령에 관한 내용은 다음장의 '구동 명령 커맨드'를 참조하시기 바랍니다.

② USB단자옆의 초록색의 3포트 단자는 RS232포트로 다른 임베디드 보드와 통신을 할 수 있습니다. (임베디드 보드에서 보내는 명령어를 받아들일 수 있습니다.)

③ CAN통신을 통해 STELLA B2를 구동시킬 수 있습니다.  
(CAN통신은 **펌웨어 버전 2.00**부터 새롭게 추가된 기능입니다.)

CAN 포트





<CAN 포트(RJ45 커넥터)>

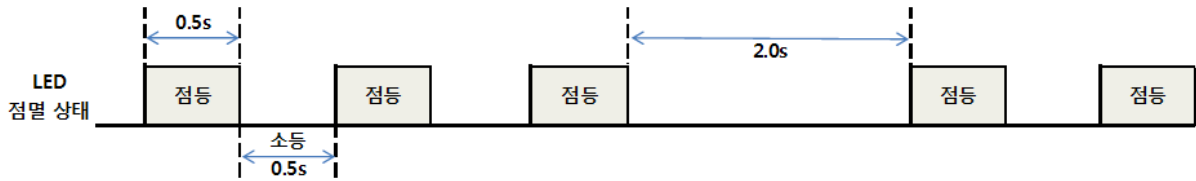
Pin	Signal	Description
1	CAN_H	CAN High bus line (Dominant high)
2	CAN_L	CAN Low bus line (Dominant low)
3	CAN_GND	CAN Ground (Common Ground)
4	N.C	No connection
5	N.C	No connection
6	N.C	No connection
7	CAN_GND	CAN Ground (Common Ground)
8	N.C	No connection

<CAN통신(RJ45) pin map>

■ ALARM 출력

1. 다음과 같은 경우에 ALARM 이 발생하며 ALARM LED(모션드라이브의 붉은색 LED) 가 점멸 됩니다. 이때 LED 의 점멸 횟수와 로봇의 현재 상태를 확인하는 명령으로 알람 내용을 확인할 수 있습니다.
2. 0.5 초 점등, 0.5 초 소등으로 규정 횟수 점멸 후, 약 2 초의 간격을 두고 점멸을 반복합니다.
3. ALARM 발생시에는 모터 구동에 대한 명령을 받아들이지 않습니다.

알람 내용	ALARM LED 점멸 회수	명령어를 통한 상태 응답 값	원인
과부하 경고	3	STX 1 ETX STX 2 ETX STX 3 ETX	제한전류를 초과하는 전류량이 2초 이상 지속될 때
과전압 경고	4	STX 4 ETX	인가되는 전압이 DC15 V를 초과했을 때
저전압 경고	5	STX 5 ETX	인가되는 전압이 DC10 V 미만일 때



## 경고

ALARM이 발생하면 원인을 제거한 후에 로봇을 구동하시기 바랍니다. 원인을 제거하지 않고 운전을 계속하면 오작동, 장치 파손의 원인이 됩니다.

## 4. RS232 구동 명령 커맨드

- **RS232C** 방식의 제어 모드입니다.
- **USB** 케이블을 이용해 **PC**와 연결하면 **PC**에서 가상으로 **COM**포트를 생성하여 줍니다.
- 통신 환경은 다음과 같습니다.  
**BaudRate : 115200**  
**Data bits : 8**  
**Parity : 없음**  
**Stop bits : 1**  
 흐름제어 : 없음
- 모든 명령코드 앞에는 반드시 **ASCII 'STX'(0x02)**를 끝에는 **'ETX'(0x03)**를 붙여서 전송해야 합니다.
- 명령코드는 크게 구동에 관한 명령과 데이터(파라미터)를 설정하는 것과 데이터를 읽어오는 코드로 나뉘어집니다.
- 구동에 관한 명령은 **'C'(Command)**로 시작하며 데이터를 설정하는 명령은 **'S'(Set)**로 시작됩니다. 또한 데이터를 읽어오는 코드는 **'G'(Get)**로 시작합니다.
- 명령코드에 사용되는 문자는 모두 대문자입니다.
- 명령 전송 후 다음 명령을 전송할 때까지 충분한 시간간격(약 **20~ 100ms**)을 확보하여 주

시기 바랍니다.

- 단위가 [rad/sec], [rad/sec^2]인 것은 휠이 아닌 모터기준의 속도입니다.

- **Scale Factor**

- 모바일 로봇/플랫폼은 양쪽 휠이 완전히 동일한 회전속도를 내는 것이 좋습니다. 하지만 같은 회사에서 생산된 같은 종류의 모터라 하더라도 각각의 모터들은 회전속도에 조금씩 차이가 있으며 하나의 모터에 대한 정/역회전 속도 또한 차이가 있습니다.

이러한 양쪽 휠의 회전속도의 차이를 보정하기 위해 **STELLA B2**은 회전 및 주행에 대한 **Scale Factor**를 사용합니다. 이 **Scale Factor**는 제어기에서 산출된 출력 값을 더 높여 주는 가중치 역할을 하는 것입니다.

$$\text{가중치} = (\text{Scale Factor} / 1000)$$

예를 들어 **Scale Factor**가 1인 경우의 가중치는 0.001입니다. 이 가중치가 제어기의 출력을 0.001 만큼 출력을 높여주는 것입니다.

$$\text{모터에 입력하는 출력 값} = (\text{제어기 출력 값}) * (1 + 0.001)$$

- **STELLA B2**는 다음과 같은 명령 지원합니다

No	Description
1	속도명령1 (F : Forward, B : backward) (000 ~ 270rad/sec) (좌,우 모터 중에 어느 한쪽만 구동하거나 좌/우 모터를 같은 방향으로 같은 속도로 구동할 때 사용하는 명령)
2	속도명령2 (F : Forward, B : backward) (000 ~ 270rad/sec) (한번의 명령으로 좌,우 모터를 각각 속도와 방향을 다르게 구동 시킬 수 있습니다)
3	속도명령3 (F : Forward, B : backward) (000 ~ 270rad/sec) (한번의 명령으로 좌,우 모터를 각각 속도와 방향을 다르게 구동 시킬 수 있습니다, 위의 '2'번 명령과 다른 점은 이 명령은 전송하면 현재 위치와 플랫폼의 현재 상태를 알려줍니다.)
4	속도명령4 (F : Forward, B : backward) (000 ~ 270rad/sec) (한번의 명령으로 좌,우 모터를 각각 속도와 방향을 다르게 구동 시킬 수 있습니다, 위의 '2'번 명령과 다른 점은 이 명령은 전송하면 엔코더 데이터와 플랫폼의 현재 상태를 알려줍니다.)
5	위치 명령 (A: Angle (0-360 degree), D: Distance [000 ~ 999cm] T: Time [0 ~ 255 second])
6	정지 명령 (1 ~ 3)
7	리셋 명령
8	초기화 명령
9	상태 값 읽어오기 ( return '0' ~ '7')
10	현재속도 읽어오기 [rad/sec]
11	현재 위치 읽어오기 [meter]
12	회전 스케일 팩터 읽어오기
13	주행 스케일 팩터 읽어오기
14	속도 가/감속 값 읽어오기 [rad/sec^2]
15	속도/위치 제어기 p, I, d, il 값 읽어오기



<b>16</b>	설정된 휠 반지름 읽어오기 [단위 <b>meter</b> ]
<b>17</b>	설정된 휠 사이의 너비 읽어오기 [단위 <b>meter</b> ]
<b>18</b>	펌웨어 버전정보 읽어오기
<b>19</b>	<b>CAN ID</b> 읽어오기
<b>20</b>	엔코더 데이터 읽어오기
<b>21</b>	단위 : <b>mA (max 4000mA)</b>
<b>22</b>	회전 스케일 팩터 값 입력 ( <b>000 ~ 999</b> )
<b>23</b>	주행 스케일 팩터 값 입력 ( <b>000 ~ 999</b> )
<b>24</b>	속도 가/감속 값 입력 ( <b>0.0 ~ 0.9</b> ) [rad/sec^2]
<b>25</b>	속도/위치 제어기 <b>p, I, d, i1</b> 값 입력 ( <b>000.000 ~ 999.999</b> )
<b>26</b>	<b>Wheel Radius</b> 설정 [단위 <b>meter</b> ]
<b>27</b>	두 휠 사이의 너비 설정 [단위 <b>meter</b> ]
<b>28</b>	<b>CAN ID</b> 설정하기 ( <b>001 ~ 255</b> )

※ 이 박스의 내용은 다음 페이지에서 참고하시기 바랍니다

**[L / R / A]**

'L' , 'R' , 'A ' 중에 한 문자가 들어감을 의미합니다

**[F / B]**

'F' , 'B' 중에 한 문자가 들어감을 의미합니다

**[V / P]**

'V' , 'P' 중에 한 문자가 들어감을 의미합니다

**[A / D]**

'A' , 'D' 중에 한 문자가 들어감을 의미합니다

**[L / R]**

'L' , 'R' 중에 한 문자가 들어감을 의미합니다

**[P / I / D / L]**

'P' , 'I' , 'D' , 'L' 중에 한 문자가 들어감을 의미합니다

**[M / S]**

'M' , 'S' 중에 한 문자가 들어감을 의미합니다.

**%**

'0' ~ '9' 까지의 숫자가 들어가는 자리임을 의미합니다

펌웨어 버전 **2.00** 으로 표시된 명령은 펌웨어 버전 **v2.00**에 새로이 추가된 명령들입니다.

No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	byte	구현	필요여 부전
1	STX	C	V	[L/R/A]	[F/B]	%	%	%	ETX										9	0	
2	STX	C	V	[F/B]	%	%	%	[F/B]	%	%	ETX								12	0	2.00
3	STX	C	V	[F/B]	%	%	%	[F/B]	%	%	P	ETX							13	0	2.00
4	STX	C	V	[F/B]	%	%	%	[F/B]	%	%	E	ETX							13	0	2.00
5	STX	C	P	A	[L/R]	%	%	%	D	[F/B]	%	%	T	%	%	%	ETX		18	0	
6	STX	C	S	T	O	P	%	ETX											8	0	
7	STX	C	R	E	S	E	T	[L/R/A]	ETX										9	0	
8	STX	C	I	N	I	T	ETX												7	0	
9	STX	G	S	T	A	T	E	ETX											8	0	
10	STX	G	V	E	L	O	C	I	T	Y	[L/R/A]	ETX							12	0	
11	STX	G	P	O	S	I	T	I	O	N	[L/R/A]	ETX							12	0	2.00
12	STX	G	S	F	A	[L/R]	ETX												7	0	
13	STX	G	S	F	D	[L/R]	ETX												7	0	
14	STX	G	V	[A/D]	[L/R]	ETX													6	0	
15	STX	G	G	[N/P]	[P/I/D/L]	[L/R]	ETX												7	0	
16	STX	G	R	ETX															4	0	
17	STX	G	W	ETX															4	0	
18	STX	G	V	E	R	S	I	O	N	ETX									10	0	2.00
19	STX	G	C	A	N	I	D	[M/S]	ETX										9	0	2.00
20	STX	G	E	N	C	ETX													6	0	2.00
21	STX	S	L	A	%	%	%	ETX											9	0	
22	STX	S	S	F	A	[L/R]	%	ETX											10	0	
23	STX	S	S	F	D	[L/R]	%	ETX											10	0	
24	STX	S	V	[A/D]	[L/R]	%	.	ETX											9	0	
25	STX	S	G	[N/P]	[P/I/D/L]	[L/R]	%	ETX		.	%	%	ETX						14	0	
26	STX	S	R	%	.	%	%	ETX											10	0	
27	STX	S	W	%	.	%	%	ETX											10	0	
28	STX	S	C	A	N	I	D	[M/S]	%	%	ETX								12	0	2

- **STELLA B2** 명령 오류 피드백 - **STELLA B2**의 모션 드라이브는 명령 오류에 대해서 **Error Code**를 피드백 합니다. 이 **Error Code**로 오류 원인을 파악할 수 있습니다. 명령대로 동작하지 않을 때 이런 **Error Code**가 발생하지 않았는지 확인하시기 바랍니다.

Error Code	설명
<b>08</b>	입력한 위치 명령이 모터 한계속도 값을 넘어서는 명령일 경우 발생
<b>09</b>	명령 문자열에 문자가 더 들어갔거나 덜 들어간 경우
<b>10 or 02</b>	명령 문자열이 올바르지 않은 경우
<b>11</b>	명령 문자열중 방향 데이터가 올바르지 않은 경우
<b>12</b>	입력 데이터가 범위를 벗어나는 경우

예) 명령 문자열이 올바르지 않은 경우

STX	e	r	r	o	r	1	0	ETX
-----	---	---	---	---	---	---	---	-----

### ※ STELLA B2의 모션 제어기에 설정된 파라미터 데이터들의 기본 값

(전원을 넣거나 초기화 명령을 보내면 모션 드라이브는 아래의 **DEFAULT PARAMETER**들로 설정됩니다)

- 휠 반지름 = 0.0752 [meter]
- 두 휠 사이의 너비 = 0.29 [meter]
- velocity controller p = 30.0
- velocity controller i = 200.0
- velocity controller d = 0.005
- velocity controller il = 20.0

- velocity controller acc = 0.4 [rad/sec^2]
- velocity controller dec = 0.1 [rad/sec^2]
- position controller p = 4.473
- position controller i = 0.015
- position controller d = 7.040
- position controller il = 100.0
- limit current = 2500mA
- limit min voltage = 10V
- limit max voltage = 15V
- 왼쪽 회전 스케일팩터 = 000
- 오른쪽 회전 스케일팩터 = 000
- 왼쪽 주행 스케일팩터 = 000
- 오른쪽 주행 스케일팩터 = 000

## ■ COMMAND 명령

### 1. 속도 명령1

- STELLA B2 를 Forward/Backward 로 한계속도(모터기준 270rad/sec, 휠 기준 5.2km/h) 안에서 구동 시킬 수 있습니다.

$$\ast (((270/(2*\pi))*60*60)/14)*0.4725)/1000 = 5.2 \text{ km/h}$$

- 속도명령에서 속도단위는 휠이 아닌 Motor 기준의 회전속도 입니다.
- 속도 설정 범위 : "000" ~ "270" [rad/sec]
- 명령코드 :

양쪽 휠 모두 Forward 방향으로 100rad/sec 의 속도로 구동 - CVAF100

STX	C	V	A	F	1	0	0	ETX
-----	---	---	---	---	---	---	---	-----

양쪽 휠 모두 Backward 방향으로 100rad/sec 의 속도로 구동 - CVAB100

STX	C	V	A	B	1	0	0	ETX
-----	---	---	---	---	---	---	---	-----

왼쪽 휠 Forward 방향으로 100rad/sec 의 속도로 구동 - CVLF100

STX	C	V	L	F	1	0	0	ETX
-----	---	---	---	---	---	---	---	-----

왼쪽 휠 Backward 방향으로 100rad/sec 의 속도로 구동 - CVLB100

STX	C	V	L	B	1	0	0	ETX
-----	---	---	---	---	---	---	---	-----

오른쪽 휠 Forward 방향으로 100rad/sec 의 속도로 구동 - CVRF100

STX	C	V	R	F	1	0	0	ETX
-----	---	---	---	---	---	---	---	-----

오른쪽 휠 Backward 방향으로 100rad/sec 의 속도로 구동 - CVRB100

STX	C	V	R	B	1	0	0	ETX
-----	---	---	---	---	---	---	---	-----

## 2. 속도 명령2

- STELLA B2 를 Forward/Backward 로 한계속도(모터기준 270rad/sec, 휠 기준 5.2km/h) 안에서 구동 시킬 수 있습니다.

※  $\frac{(((270/(2*\pi))*60*60)/14)*0.4725}{1000} = 5.2 \text{ km/h}$

- 이 명령은 '속도명령 1'과 다르게 좌/우 모터의 속도와 방향을 각각 구동 시킬 수 있습니다.

- 속도명령에서 속도단위는 휠이 아닌 Motor 기준의 회전속도 입니다.

- 속도 설정 범위 : "000" ~ "270" [rad/sec]

- 명령코드 :

오른쪽 모터는 Forward 방향으로 100rad/sec 의 속도로 구동하고, 왼쪽 모터는 Backward 방향으로 150rad/sec 의 속도로 구동하는 명령

STX	C	V	F	1	0	0	B	1	5	0	ETX
-----	---	---	---	---	---	---	---	---	---	---	-----

오른쪽 모터는 Backward 방향으로 050rad/sec 의 속도로 구동하고, 왼쪽 모터는 Forward 방향으로 100rad/sec 의 속도로 구동하는 명령

STX	C	V	B	0	5	0	F	1	0	0	ETX
-----	---	---	---	---	---	---	---	---	---	---	-----

## 3. 속도 명령3

- STELLA B2 를 Forward/Backward 로 한계속도(모터기준 270rad/sec, 휠 기준 5.2km/h)

안에서 구동 시킬 수 있습니다.

$$\ast (((270/(2*\pi))*60*60)/14)*0.4725)/1000 = 5.2 \text{ km/h}$$

- 이 명령은 '속도명령 2'와 거의 같습니다. 다른 점은 이 명령이 다른 속도명령과 다른 점은 현재 위치와 플랫폼의 상태를 피드백 해준다는 것입니다.

- 속도명령에서 속도단위는 휠이 아닌 Motor 기준의 회전속도 입니다.

- 속도 설정 범위 : "000" ~ "270" [rad/sec]

- 명령코드 :

오른쪽 모터는 Forward 방향으로 100rad/sec 의 속도로 구동, 왼쪽 모터는 Backward 방향으로 050rad/sec 의 속도로 구동시키고 두 휠의 이동량과 플랫폼의 상태를 읽는 명령

STX	C	V	F	1	0	0	B	0	5	0	P	ETX
-----	---	---	---	---	---	---	---	---	---	---	---	-----

오른쪽 모터는 Backward 방향으로 050rad/sec 의 속도로 구동, 왼쪽 모터는 Forward 방향으로 100rad/sec 의 속도로 구동하는 명령시키고 두 휠의 이동량과 플랫폼의 상태를 읽는 명령

STX	C	V	B	0	5	0	F	1	0	0	P	ETX
-----	---	---	---	---	---	---	---	---	---	---	---	-----

예 ) 위 명령에 대한 응답코드(응답 값의 단위는 [meter]이며 소수점 셋째 자리까지 표현합니다. 또한 응답 최대값은 9999.999meter 입니다.)

STX	F	0	0	0	3	.	1	4	1	B	0	0	6	.	2	8	2	S	7	ETX
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----

-> 오른쪽 휠 은 Forward 방향으로 3.141m 이동했고 왼쪽 휠 은 Backward 방향으로 6.282m 이동했으며 플랫폼의 상태는 모터가 회전하고 있음을 나타냄  
(17 번째 문자 'S' 다음의 18 번째 문자가 모터의 상태를 나타내는 문자입니다)

#### 4. 속도 명령4

- STELLA B2 를 Forward/Backward 로 한계속도(모터기준 270rad/sec, 휠 기준 5.2km/h) 안에서 구동 시킬 수 있습니다.

$$\ast (((270/(2*\pi))*60*60)/14)*0.4725)/1000 = 5.2 \text{ km/h}$$

- 이 명령은 '속도명령 2'와 거의 같습니다. 다른 점은 이 명령이 다른 속도명령과 다른 점은 엔코더 데이터와 플랫폼의 상태를 피드백 해준다는 것입니다.

- 속도명령에서 속도단위는 휠이 아닌 Motor 기준의 회전속도 입니다.

- 속도 설정 범위 : "000" ~ "270" [rad/sec]

- 명령코드 :

오른쪽 모터는 Forward 방향으로 100rad/sec 의 속도로 구동, 왼쪽 모터는 Backward 방향으로 050rad/sec 의 속도로 구동시키고 두 휠의 이동량과 플랫폼의 상태를 읽는 명령

STX	C	V	F	1	0	0	B	0	5	0	E	ETX
-----	---	---	---	---	---	---	---	---	---	---	---	-----

오른쪽 모터는 Backward 방향으로 050rad/sec 의 속도로 구동, 왼쪽 모터는 Forward 방향으로 100rad/sec 의 속도로 구동하는 명령시키고 엔코더 데이터와 플랫폼의 상태를 읽는 명령

STX C V B 0 5 0 F 1 0 0 E ETX

- 응답코드 :  
앞부분이 오른쪽 모터에 대한 엔코더 데이터이며 뒷부분이 왼쪽 모터에 대한 엔코더 데이터 입니다. 각 엔코더 데이터 표현에는 10 바이트(방향 1 바이트, 데이터 9 바이트)를 사용하고 있습니다.

STX F 0 0 0 0 9 3 3 1 5 B 0 0 0 0 9 3 2 7 0 ETX

-> 오른쪽 모터는 Forward 방향으로 93315pulse 만큼 이동했고, 왼쪽 모터는 Backward 방향으로 93270pulse 만큼 이동했음을 알 수 있습니다

읽은 엔코더 펄스로부터 이동량 계산하기 :

$$\text{이동 량 [meter]} = (((2 \times \pi) / (\text{엔코더 CPR} \times 4)) * \text{읽은 엔코더 데이터}) / \text{기어 비}) * \text{휠 반지름[meter]}$$

STELLA B2 의 엔코더 CPR : 256

STELLA B2 모터의 기어비 : 13.71

$$\text{오른쪽 휠의 이동 량} = (((2 \times 3.14) / (256 \times 4)) * 93315) / 13.71) * 0.0752$$

$$= \text{약 } 3.14 \text{ [meter]}$$

$$\text{왼쪽 휠의 이동 량} = (((2 \times 3.14) / (256 \times 4)) * 93270) / 13.71) * 0.0752$$

$$= \text{약 } 3.14 \text{ [meter]}$$

### 5. 위치 명령

- STELLA B2 를 원하는 위치로 이동 및 회전시킬 수 있습니다.
- 위치 명령에는 회전각도, 이동거리, 이동시간이 필요합니다.
- 회전각 설정 범위 : "000" ~ "360" [degree]
- 이동거리 설정 범위 : "000" ~ "999" [centi meter]
- 이동시간 설정 범위 : "000" ~ "255" [second]
- 명령코드 :

회전하지 않으면서 Forward 방향으로 5초동안 1m 전진 -

CPAR000DF100T005

STX C P A R 0 0 0 D F 1 0 0 T 0 0 5 ETX  
또는

CPAL000DF100T005

STX C P A L 0 0 0 D F 1 0 0 T 0 0 5 ETX

제자리에서 2초동안 오른쪽으로 90도 회전

CPAR090DF000T002



STX C P A R 0 9 0 D F 0 0 0 T 0 0 2 ETX  
또는

CPAR090DB000T002

STX C P A R 0 9 0 D B 0 0 0 T 0 0 2 ETX  
제자리에서 2초동안 왼쪽으로 90도 회전

CPAL090DF000T002

STX C P A L 0 9 0 D F 0 0 0 T 0 0 2 ETX  
또는

CPAL090DB000T002

STX C P A L 0 9 0 D B 0 0 0 T 0 0 2 ETX

## 6. 정지 명령

- 모터 정지 명령입니다.
- 정지모드 입력 범위 : '1' ~ '3'
- 명령코드 :

양쪽 휠 모두 정지(Motor Free) - CSTOP1

STX C S T O P 1 ETX

※ CSTOP1 명령 후에 리셋 또는 초기화 명령을 주거나 전원을 재입력 해주기 전까지는 구동 명령을 받아들이지 않습니다.

양쪽 휠 모두 정지(Motor Hold) - CSTOP2

STX C S T O P 2 ETX

양쪽 휠 모두 감속하며 정지(Motor Hold) - CSTOP3

STX C S T O P 3 ETX

## 7. 리셋 명령

- 사용자가 설정한 파라미터 데이터로 리셋되며 파라미터들을 따로 설정하지 않았으면 기본 값들로 설정된다.
- 명령코드 :

양쪽 휠 모두 리셋 - CRESETA

STX C R E S E T A ETX

오른쪽 휠 리셋 - CRESETR



STX C R E S E T R ETX

왼쪽 휠 리셋 - CRESETL

STX C R E S E T L ETX

## 8. 초기화 명령

- 유저가 설정한 파라미터 데이터가 아닌 제품 출하시의 기본 값들로 설정된다. 그 외의 데이터들은 0으로 초기화 된다.

- 명령코드 :

CINIT

STX C I N I T ETX

## ■ GET 명령

### 9. STELLA B2의 현재 상태 값 읽어오기

- 응답 값을 확인하여 STELLA B2의 상태를 확인할 수 있습니다.

- 명령코드 :

GSTATE

STX G S T A T E ETX

- 응답코드 :

0 - 정상 상태

STX 0 ETX

1 - 오른쪽 휠 과부하 상태

STX 1 ETX

2 - 왼쪽 휠 과부하 상태

STX	2	ETX
-----	---	-----

3 - 양쪽 휠 과부하 상태

STX	3	ETX
-----	---	-----

4 - 과전압 상태 (인가 전압 15V 이상)

STX	4	ETX
-----	---	-----

5 - 저전압 상태 (인가 전압 10V 미만)

STX	5	ETX
-----	---	-----

6 - 목표위치 도달 상태 (위치 명령시에만 나타나는 상태임)

STX	6	ETX
-----	---	-----

7 - 휠 회전 중인 상태 (어떤 명령에 의해 휠이 움직이고 있는 상태)

STX	7	ETX
-----	---	-----

※ 과부하, 과전압, 저전압은 그 상태가 2초이상 지속되었을 때 알람이 발생하며 휠을 정지상태로 만들며 이때는 휠 구동 명령에 반응하지 않습니다. 이 세가지 상태에 대한 알람이 발생하면 STELLA B2의 전원을 OFF 한 후 알람 상황을 제거해 주고 전원을 다시 입력하시기 바랍니다.

또한 이 세가지 알람에 대해서는 모션 드라이브의 LED로도 확인이 가능합니다.

과전류 알람일 때 붉은색 LED가 주기적으로 3번 점멸

과전압 알람일 때 붉은색 LED가 주기적으로 4번 점멸

저전압 알람일 때 붉은색 LED가 주기적으로 5번 점멸

위치 명령에 의해 목표위치에 도달했을 때에는 모션 드라이브의 파란색 LED가 점등되며 STELLA B2가 구동 중이거나 위치 명령이 아닐 경우에는 파란색 LED는 OFF 되어있습니다.

## 10. STELLA B2의 현재 속도 읽어오기

- 응답 값을 확인하여 STELLA B2의 속도를 확인할 수 있습니다.
- 응답 속도 값은 모터의 회전속도[rad/sec]를 나타냅니다.
- 명령코드 :

왼쪽 모터 속도 읽어오기 - GVELOCITYL

STX G V E L O C I T Y L ETX

오른쪽 모터 속도 읽어오기 - GVELOCITYR

STX G V E L O C I T Y R ETX

- 응답코드 :

STX F 1 0 0 . 0 ETX

-> Forward 방향으로 100.0 rad/sec 의 속도로 회전하고 있음.

STX B 1 2 3 . 4 ETX

-> Backward 방향으로 123.4 rad/sec 의 속도로 회전하고 있음.

- 명령코드 :

양쪽 모터 속도 읽어오기 - GVELOCITYA

STX G V E L O C I T Y A ETX

- 응답코드 :

STX F 1 0 0 . 0 B 2 0 0 . 0 ETX

-> 오른쪽 모터는 Forward 방향으로 100.0 rad/sec 의 속도로 회전하고 있으며 왼쪽 모터는 Backward 방향으로 200rad/sec 의 속도로 회전하고 있음.

## 11. STELLA B2의 이동한 거리 읽어오기

- 응답 값을 확인하여 STELLA B2 가 이동한 거리를 확인할 수 있습니다.

- 응답 값은 STELLA B2 각 휠의 이동 량을 나타냅니다.

- 펌웨어버전 2.00 이전버전에서는 소수점 첫째자리 까지만 응답하며 2.00 이 후 버전에서만 소수점 셋째 자리 까지 응답합니다.

- 응답값의 단위는 [meter]입니다.

- 이 명령으로 읽을 수 있는 최대 이동거리는 9999.999 meter 입니다.

(펌웨어 버전 2.00 부터 소수점 셋째 자리 까지 표현합니다. 이전 버전에서는 소수점 첫째 자리 까지만 표현)

- 명령 코드 :

왼쪽 휠 이동량 읽어오기 - GPOSITIONL

STX G P O S I T I O N L ETX

오른쪽 휠 이동량 읽어오기 - GPOSITIONR

STX G P O S I T I O N R ETX

- 응답 코드 :

STX F 0 0 0 3 . 1 4 1 ETX

-> 현재 Forward 방향으로 3.141m 만큼 이동했음.

STX B 0 0 0 6 . 2 8 2 ETX

-> 현재 Backward 방향으로 6.282m 만큼 이동했음.

- 명령 코드 :

양쪽 휠 이동 량 읽어오기 - GPOSITIONA

STX G P O S I T I O N A ETX

- 응답 코드 :

STX F 0 0 0 3 . 1 4 1 B 0 0 0 6 . 2 8 2 ETX

-> 현재 오른쪽 휠은 Forward 방향으로 3.141m 이동했고 왼쪽 휠은 Backward 방향으로 6.282m 만큼 이동했음.

## 12 ~ 13. Scale Factor 값 읽어오기

- 응답 값을 확인하여 STELLA B2 의 Scale Factor 값을 확인할 수 있습니다.

- 명령 코드 :

왼쪽 휠 회전 스케일 팩터 읽어오기 - GSFAL

STX G S F A L ETX

오른쪽 휠 회전 스케일 팩터 읽어오기 - GSFAR

STX G S F A R ETX

왼쪽 휠 주행 스케일 팩터 읽어오기 - GSFDL

STX G S F D L ETX

오른쪽 휠 주행 스케일 팩터 읽어오기 - GSFDR

STX G S F D R ETX

- 응답코드 :

STX 0 . 0 0 1 ETX

-> 스케일 팩터 값은 0.001

가중치 =  $(1 + 0.001) = 1.001$

#### 14. 가/감속 값 읽어오기

- 응답 값을 확인하여 STELLA B2의 속도 명령 시 이용되는 가/감속 값을 확인할 수 있습니다.

- 응답 값은 모터의 회전 속도에 관한 값이며 단위는 [rad/sec<sup>2</sup>]입니다.

- 명령 코드 :

왼쪽 모터 가속 값 읽어오기 - GVAL

STX G V A L ETX

왼쪽 모터 감속 값 읽어오기 - GVDL

STX G V D L ETX

오른쪽 모터 가속 값 읽어오기 - GVAR

STX G V A R ETX

오른쪽 모터 감속 값 읽어오기 - GVDR

STX G V D R ETX

- 응답코드 :

STX 0 . 4 ETX

-> 0.4 [rad/sec<sup>2</sup>]

#### 15. STELLA B2의 속도/위치 제어기의 P, I, D, IL값 읽어오기

- 응답 값을 확인하여 STELLA B2의 속도/위치 제어기에서 이용되는 P, I, D, IL 값을 확인할 수 있습니다.

- 명령 코드 :

속도 제어기의 왼쪽 모터 P GAIN 읽어오기 - GGVPL

STX G G V P L ETX

속도 제어기의 왼쪽 모터 I값 읽어오기 - GGVI

STX G G V I L ETX

속도 제어기의 왼쪽 모터 D값 읽어오기 - GGVDL

STX G G V D L ETX

속도 제어기의 왼쪽 모터 IL값 읽어오기 - GGULL

STX G G V L L ETX

속도 제어기의 오른쪽 모터 P값 읽어오기 - GGVPR

STX G G V P R ETX

속도 제어기의 오른쪽 모터 I값 읽어오기 - GGVIR

STX G G V I R ETX

속도 제어기의 오른쪽 모터 D값 읽어오기 - GGVDR

STX G G V D R ETX

속도 제어기의 오른쪽 모터 IL값 읽어오기 - GGVLR

STX G G V L R ETX

위치 제어기의 왼쪽 모터 P값 읽어오기 - GGPLL

STX G G P P L ETX

위치 제어기의 왼쪽 모터 I값 읽어오기 - GGPIIL

STX G G P I L ETX

위치 제어기의 왼쪽 모터 D값 읽어오기 - GGDDL

STX G G P D L ETX

위치 제어기의 왼쪽 모터 IL값 읽어오기 - GGPLL

STX G G P L L ETX

위치 제어기의 오른쪽 모터 P값 읽어오기 - GGPPR

STX G G P P R ETX

위치 제어기의 오른쪽 모터 I값 읽어오기 - GGPIR

STX G G P I R ETX

위치 제어기의 오른쪽 모터 D값 읽어오기 - GGDDR

STX G G P D R ETX

위치 제어기의 오른쪽 모터 IL값 읽어오기 - GGPLR

STX G G P L R ETX

- 응답코드 :

STX 0 3 0 . 0 0 0 ETX

-> 설정된 게인값은 030.000 이다.

### 16. 설정된 휠 반지름 값 읽어오기

- 응답 값을 확인하여 STELLA B2 제어기에 설정되어있는 휠 반지름 값을 확인할 수 있습니다.

- 응답 값의 단위는 [meter]입니다.

- 명령 코드 :

GR

STX G R ETX

- 응답코드 :

STX 0 . 0 7 5 2 ETX

-> 설정된 휠 반지름은 0.0752m(7.52cm)이다.

### 17. 설정된 두 휠 사이의 너비 읽어오기

- 응답 값을 확인하여 STELLA B2 제어기에 설정되어있는 두 휠 사이의 너비 값을 확인할 수 있습니다.

- 응답 값의 단위는 [meter]입니다.

- 명령 코드 :

GW

STX G W ETX

- 응답코드 :

STX 0 . 2 8 9 0 ETX

-> 설정된 두 휠 사이의 너비는 0.289m(28.9cm)이다.

**18. 펌웨어 버전 정보 읽어오기**

- 응답 값을 확인하여 STELLA B2 펌웨어 버전을 확인할 수 있습니다.
- 명령 코드 :

GVERSION

STX G V E R S I O N ETX

- 응답코드 :

STX 2 . 0 0 ETX

-> 펌웨어 버전은 2.00 이다

**19. CAN통신 ID 읽어오기**

- 응답 값을 확인하여 STELLA B2 에 설정된 CAN 통신에 사용되는 ID 를 확인할 수 있습니다.
- CAN MASTER ID 는 스텔라가 응답 값을 보내는 목적지 ID 를 의미합니다.
- 명령 코드 :

CAN 통신에 사용되는 MASTER ID 읽어오기 - GCANIDM

STX G C A N I D M ETX

CAN 통신에 사용되는 모션 드라이버 자신의 ID 읽어오기 - GCANIDS

STX G C A N I D S ETX

- 응답코드 :

STX 2 5 5 ETX

-> 명령에 대한 응답 CAN ID 는 '255' 이다

**20. 엔코더 데이터 읽어오기**

- 응답 값을 확인하여 모터에 부착된 엔코더의 데이터를 확인할 수 있습니다.
- 응답 가능한 최대값은 999,999,999 입니다.
- 명령 코드 :

엔코더 데이터 읽어오기 - GENC

STX G E N C ETX



- 응답코드 :

앞부분이 오른쪽 모터에 대한 엔코더 데이터이며 뒷부분이 왼쪽 모터에 대한 엔코더 데이터 입니다. 각 엔코더 데이터 표현에는 10 바이트(방향 1 바이트, 데이터 9 바이트)를 사용하고 있습니다.

STX F 0 0 0 0 9 3 3 1 5 B 0 0 0 0 9 3 2 7 0 ETX

-> 오른쪽 모터는 STELLA B2 에 처음 전원을 입력한 지점으로부터 Forward 방향으로 93315pulse 만큼 회전(이동)했고, 왼쪽 모터는 STELLA B2 에 처음 전원을 입력한 지점으로부터 Backward 방향으로 93270pulse 만큼 회전(이동)했음을 알 수 있습니다

※ 읽은 엔코더 펄스로부터 이동 량 계산하기

이동 량 [meter] =

$$(((2 \times \pi) / (\text{엔코더 CPR} \times 4)) * \text{읽은 엔코더 데이터} / \text{기어비}) * \text{휠 반지름[meter]}$$

STELLA B2 의 엔코더 CPR : 256

STELLA B2 모터의 기어비 : 13.71

$$\text{오른쪽 휠의 이동 량} = (((2 \times 3.14) / (256 \times 4)) * 93315) / 13.71 * 0.0752$$

$$= \text{약 } 3.14 \text{ [meter]}$$

$$\text{왼쪽 휠의 이동 량} = (((2 \times 3.14) / (256 \times 4)) * 93270) / 13.71 * 0.0752$$

$$= \text{약 } 3.14 \text{ [meter]}$$

## ■ SET 명령

### 21. 허용 제한 전류 값 설정하기

- 단위는 [mA]이며 최대 4000mA 까지 설정할 수 있습니다.

- 설정 범위 : "0000" ~ "4000"

- 이 허용 제한 전류 값 이상의 전류가 2 초이상 흐르면 과부하 알람이 발생합니다.

- 명령 코드 :

허용 제한 전류 값을 2500mA(2.5A)로 설정하기 - SLA2500

STX S L A 2 5 0 0 ETX

### 22 ~ 23. Scale Factor 값 설정하기

- 설정 범위 : "000" ~ "999"

- 명령 코드 :

왼쪽 회전 스케일 팩터 설정하기 - SSFAL000

STX S S F A L 0 0 0 ETX

왼쪽 주행 스케일 팩터 설정하기 - SSFDL000

STX S S F D L 0 0 0 ETX

오른쪽 회전 스케일 팩터 설정하기 - SSFAR000

STX S S F A R 0 0 0 ETX

오른쪽 주행 스케일 팩터 설정하기 - SSFDR000

STX S S F D R 0 0 0 ETX

## 24. 가/감속 값 설정하기

- 단위는  $[\text{rad}/\text{sec}^2]$ 이며, 모터의 속도에 대한 가/감속 값입니다.
- 입력 범위 : "0.0" ~ "9.9"
- 명령 코드 :

왼쪽 모터 가속 값 0.4  $\text{rad}/\text{sec}^2$ 으로 설정하기 - SVAL0.4

STX S V A L 0 . 4 ETX

왼쪽 모터 감속 값 0.1  $\text{rad}/\text{sec}^2$ 으로 설정하기 - SVDL0.1

STX S V D L 0 . 1 ETX

오른쪽 모터 가속 값 0.4  $\text{rad}/\text{sec}^2$ 으로 설정하기 - SVAR0.4

STX S V A R 0 . 4 ETX

오른쪽 모터 감속 값 0.1  $\text{rad}/\text{sec}^2$ 으로 설정하기 - SVDR0.1

STX S V D R 0 . 1 ETX

## 25. 속도/위치 제어기의 P, I, D, IL값 설정하기

- 입력 범위 : "000.000" ~ "999.999"
- 명령 코드 :

속도 제어기의 왼쪽 모터 P값 30.0으로 설정하기 - SGVPL030.000

STX S G V P L 0 3 0 . 0 0 0 ETX

속도 제어기의 왼쪽 모터 I값 200.0으로 설정하기 - SGVIL200.000

STX S G V I L 2 0 0 . 0 0 0 ETX

속도 제어기의 왼쪽 모터 D값 0.005로 설정하기 - SGVDL000.005

STX S G V D L 0 0 0 . 0 0 5 ETX

속도 제어기의 왼쪽 모터 IL값 20.0으로 설정하기 - SGVLL020.000

STX S G V L L 0 2 0 . 0 0 0 ETX

속도 제어기의 오른쪽 모터 P값 30.0으로 설정하기 - SGVPR030.000

STX S G V P R 0 3 0 . 0 0 0 ETX

속도 제어기의 오른쪽 모터 I값 200.0으로 설정하기 - SGVIR200.000

STX S G V I R 2 0 0 . 0 0 0 ETX

속도 제어기의 오른쪽 모터 D값 0.005로 설정하기 - SGVDR000.005

STX S G V D R 0 0 0 . 0 0 5 ETX

속도 제어기의 오른쪽 모터 IL값 20.0으로 설정하기 - SGVLR020.000

STX S G V L R 0 2 0 . 0 0 0 ETX

위치 제어기의 왼쪽 모터 P값 4.472로 설정하기 - SGPPL004.472

STX S G P P L 0 0 4 . 4 7 2 ETX

위치 제어기의 왼쪽 모터 I값 0.015로 설정하기 - SGPIL000.015

STX S G P I L 0 0 0 . 0 1 5 ETX

위치 제어기의 왼쪽 모터 D값 7.039로 설정하기 - SGPDL007.039

STX S G P D L 0 0 7 . 0 3 9 ETX

위치 제어기의 왼쪽 모터 IL값 100.0으로 설정하기 - SGP LL100.000

STX S G P L L 1 0 0 . 0 0 0 ETX

위치 제어기의 오른쪽 모터 P값 4.472로 설정하기 - SGPPR004.472

STX S G P P R 0 0 4 . 4 7 2 ETX

위치 제어기의 오른쪽 모터 I값 0.015로 설정하기 - SGPIR000.015

STX S G P I R 0 0 0 . 0 1 5 ETX

위치 제어기의 오른쪽 모터 D값 7.039로 설정하기 - SGPDR007.039

STX S G P D R 0 0 7 . 0 3 9 ETX

위치 제어기의 오른쪽 모터 IL값 100.0으로 설정하기 - SGPLR100.000

STX S G P L R 1 0 0 . 0 0 0 ETX

## 26. 휠 반지름 값 설정하기

- 입력 단위는 [meter] 입니다.
- 입력 범위 : "0.0000" ~ "0.1000"
- 명령 코드 :

휠 반지름 값을 0.0742m(7.42cm)로 설정하기 - SR0.0742

STX S R 0 . 0 7 4 2 ETX

## 27. 두 휠 사이의 너비 설정하기

- 입력 단위는 [meter] 입니다.
- 입력 범위 : "0.0000" ~ "0.4000"
- 명령 코드 :

두 휠 사이의 너비 값을 0.289m(28.9cm)로 설정하기 - SW0.2890

STX S W 0 . 2 8 9 0 ETX

## 28. CAN ID 설정하기

- 입력 범위 : "001" ~ "255"
- MASTER는 STELLA가 응답 값을 보내는 목적지를 의미합니다.
- 명령 코드 :

MASTER ID 를 "255"로 설정하기

STX S C A N I D M 2 5 5 ETX

스텔라 ID 를 "011"로 설정하기

## 5. CAN 구동 명령 커맨드

- CAN 통신 제어 모드로서 **STELLA B2** 펌웨어 버전 **2.00**에 탑재되었습니다.
- 모션드라이브의 **RJ45** 포트를 이용해 **CAN** 통신을 합니다
- 통신 환경은 다음과 같습니다.  
**CAN2.0B - Extended CAN**  
**Fixed BaudRate : 250Kb/s**  
 모션 드라이버의 **CAN ID : 11** (제품 출하 시 기본값)  
**Fixed Master MASTER ID : 255** (제품 출하 시 기본값)
- **CAN ID**(마스터와 모션드라이버)는 사용자가 변경한 후 리셋을 해도 기본값(제품 출하 시의 기본값)으로 변경되지 않고 사용자가 입력한 **ID**를 계속 보존하여 사용합니다.
- 명령코드는 크게 구동에 관한 명령과 데이터(파라미터)를 설정하는 것과 데이터를 읽어오는 코드로 나뉘어집니다.
- **CAN** 통신에서 이용되는 **sbyte** 데이터 중 첫 번째 데이터는 명령이나 응답에 대한 코드를 나타냅니다.
- **CAN** 통신으로 주고받는 데이터는 모두 **헥사코드(16진수)** 형태입니다.  
 따라서, 데이터를 주고받을 때에 **10진수**를 **16진수**로, 혹은 **16진수**를 **10진수**로 변환하는 과정이 필요할 수 있습니다.
- 명령 전송 후 다음 명령을 전송할 때까지 충분한 **시간간격(약 10-20ms)**을 확보하여 주시기 바랍니다.
- 설명 중 ' >> ' 로 표기된 것은 비트 쉬프트 연산을 의미하며 ' | '는 비트 **OR** 연산을 의미합니다

다.

- 단위가 [rad/sec], [rad/sec^2]인 것은 휠이 아닌 모터기준의 속도입니다.

- **Scale Factor**

- 모바일 로봇/플랫폼은 양쪽 휠이 완전히 동일한 회전속도를 내는 것이 좋습니다. 하지만 같은 회사에서 생산된 같은 종류의 모터라 하더라도 각각의 모터들은 회전속도에 조금씩 차이가 있으며 하나의 모터에 대한 정/역회전 속도 또한 차이가 있습니다.

이러한 양쪽 휠의 회전속도의 차이를 보정하기 위해 **STELLA B2**은 회전 및 주행에 대한 **Scale Factor**를 사용합니다. 이 **Scale Factor**는 제어기에서 산출된 출력 값을 더 높여 주는 가중치 역할을 하는 것입니다.

$$\text{가중치} = (\text{Scale Factor} / 1000)$$

예를 들어 **Scale Factor**가 1인 경우의 가중치는 0.001입니다. 이 가중치가 제어기의 출력을 0.001만큼 출력을 높여주는 것입니다.

$$\text{모터에 입력하는 출력 값} = (\text{제어기 출력 값}) * (1 + 0.001)$$

- **STELLA B2**는 다음과 같은 **CAN** 명령을 지원합니다

No	Description
1	속도명령1 (0 ~ 270 rad/sec)
2	속도명령2 (0-270 rad/sec) (한번의 명령으로 좌/우 모터 별개 구동)
3	속도명령3 (0-270 rad/sec) (한번의 명령으로 좌/우 모터 별개 구동 및 휠의 이동 량과 플랫폼의 상태 요청)
4	위치 명령 (Angle 0-360 degree, Distance 000 ~ 999[cm] Time 0-255[second])
5	정지 명령 (1 ~ 3)
6	리셋 명령
7	초기화 명령
8	상태 값 읽어오기 ( return 0 ~ 7)
9	현재속도 읽어오기 (rad/sec)
10	현재 위치 읽어오기 (centimeter)
11	회전 스케일 팩터 읽어오기
12	주행 스케일 팩터 읽어오기
13	속도 가/감속 값 읽어오기 (rad/sec^2)
14	속도/위치 제어기 p, I, d, il 값 읽어오기
15	설정된 휠 반지름 읽어오기 (단위: 1/100 centimeter)
16	설정된 휠 사이의 너비 읽어오기 (단위: 1/100 centimeter)
17	모션드라이버 펌웨어 버전 읽어오기
18	엔코더 데이터 읽어오기
19	전류제한 값 설정하기 (단위: mA, 최대값: 4000mA)
20	회전 스케일 팩터 값 입력 (범위: 000 ~ 999)
21	주행 스케일 팩터 값 입력 (범위: 000 ~ 999)

<b>22</b>	속도 가/감속 값 입력 (범위: <b>0.0 ~ 9.9</b> , 단위: <b>rad/sec^2</b> )
<b>23</b>	속도/위치 제어기 <b>p, I, d, i1</b> 값 입력 (범위: <b>000.000 ~ 999.999</b> )
<b>24</b>	<b>Wheel Radius</b> 설정 (범위: <b>0000 ~ 1000</b> , 단위: <b>1/100 centimeter</b> )
<b>25</b>	두 휠 사이의 너비 설정 (범위: <b>0000 ~ 4000</b> , 단위: <b>1/100 centimeter</b> )

- **STELLA B2** 명령 오류 피드백 - **STELLA B2**의 모션 드라이브는 명령 오류에 대해서 **Error Code**를 피드백 합니다. 이 **Error Code**로 오류 원인을 파악할 수 있습니다. 명령대로 동작하지 않을 때 이런 **Error Code**가 발생하지 않았는지 확인하시기 바랍니다.

**Error Code** 설명

- 08 (0x08)** 입력한 위치 명령이 모터 한계속도 값을 넘어서는 명령일 경우 발생
- 10 (0x0a)** 명령이 올바르지 않은 경우
- 11 (0x0b)** 방향 데이터가 올바르지 않은 경우
- 12 (0x0c)** 입력 데이터가 범위를 벗어나는 경우

예) 범위를 벗어나는 데이터를 입력 할 경우

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x91      0x0c            0            0            0            0            0            0**

**Data ①** 에러를 나타내는 코드 : 0x91

**Data ②** 에러 내용을 나타내는 코드 (0x0c => 범위를 벗어나는 데이터를 입력 할 경우)

※ **STELLA B2**의 모션 제어기에 설정된 파라미터 데이터들의 기본 값

(전원을 넣거나 초기화 명령을 보내면 모션 드라이브는 아래의 **DEFAULT PARAMETER**들로 설정됩니다)

- 휠 반지름 = 0.0752 [meter]
- 두 휠 사이의 너비 = 0.29 [meter]
- velocity controller p = 30.0
- velocity controller i = 200.0
- velocity controller d = 0.005
- velocity controller il = 20.0
- velocity controller acc = 0.4 [rad/sec<sup>2</sup>]
- velocity controller dec = 0.1 [rad/sec<sup>2</sup>]
- position controller p = 4.473
- position controller i = 0.015
- position controller d = 7.034
- position controller il = 100.0
- limit current = 2.5A
- limit min voltage = 10V
- limit max voltage = 15V
- 왼쪽 회전 스케일 팩터 = 000
- 오른쪽 회전 스케일 팩터 = 000
- 왼쪽 주행 스케일 팩터 = 000
- 오른쪽 주행 스케일 팩터 = 000



## ■ COMMAND 명령

### 1. 속도명령

- STELLA B2 를 Forward/Backward 로 한계속도(모터기준 270rad/sec, 휠 기준 5.2km/h) 안에서 구동 시킬 수 있습니다.

※  $\frac{(((270/(2*\pi))*60*60)/14)*0.4725}{1000} = 5.2 \text{ km/h}$

- 속도명령에서 속도단위는 휠이 아닌 Motor 기준의 회전속도 입니다.

- 속도 단위는 [rad/sec] 입니다.

- 속도 설정 범위 : 000 ~ 270 (0x0000 ~ 0x010e)

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0x01)	구동할 모 터	구동 방향	속도데이터 (상위 8bit)	속도데이터 (하위 8bit)	0	0	0

Data ① 명령 코드 : 0x01 (속도명령)

Data ② 구동할 모터 : 0x00 ~ 0x02

(0x00 = 양쪽 모터, 0x01 = 오른쪽 모터, 0x02 = 왼쪽 모터)

Data ③ 구동 방향 :

Forward 방향 : 0x46

Backward 방향 : 0x42

Data ④ 속도데이터 (상위 8bit)

Data ⑤ 속도데이터 (하위 8bit)

: 예를 들어 속도데이터 100 은 16 진수로 0x0064 이며 이 16 진수의 상위

8bit 는 '0x00', 하위 8 비트는 '0x64' 입니다

Data ⑥ ~ ⑧ : 의미 없음

예 1) 양쪽 휠 모두 Forward 방향으로 100rad/sec 의 속도로 구동

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x01	0x00	0x46	0x00	0x64	0	0	0

예 2) 양쪽 휠 모두 Backward 방향으로 100rad/sec 의 속도로 구동

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x01	0x00	0x42	0x00	0x64	0	0	0

예 3) 왼쪽 휠 Forward 방향으로 100rad/sec 의 속도로 구동

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x01	0x02	0x46	0x00	0x64	0	0	0

예 4) 왼쪽 휠 Backward 방향으로 100rad/sec 의 속도로 구동

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x01	0x02	0x42	0x00	0x64	0	0	0

**0x01      0x02      0x42      0x00      0x64      0      0      0**

예 5) 오른쪽 휠 Forward 방향으로 100rad/sec의 속도로 구동

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x01      0x01      0x46      0x00      0x64      0      0      0**

예 6) 오른쪽 휠 Backward 방향으로 100rad/sec의 속도로 구동

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x01      0x01      0x42      0x00      0x64      0      0      0**

## 2. 속도명령2

- 이 명령은 '속도명령1'과 다르게 좌/우 모터의 속도와 방향을 각각 구동 시킬 수 있습니다.  
 (즉, 한번의 명령으로 좌/우 모터를 다르게 구동 시킬 수 있습니다.)

- STELLA B2 를 Forward/Backward 로 한계속도(모터기준 270rad/sec, 휠 기준 5.2km/h) 안에서 구동 시킬 수 있습니다.

※  $(((270/(2*\pi))*60*60)/14)*0.4725)/1000 = 5.2 \text{ km/h}$

- 속도명령에서 속도단위는 휠이 아닌 Motor 기준의 회전속도 입니다.

- 속도 단위는 [rad/sec] 입니다.

- 속도 설정 범위 : 000 ~ 270 (0x0000 ~ 0x010e)

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0x06)	오른쪽 모터 구 동 방향	오른쪽 모 터 속도데 이터 (상위 8bit)	오른쪽 모 터 속도데 이터 (하위 8bit)	왼쪽 모 터 구동 방향	왼쪽 모터 속도데이터 (상위 8bit)	왼쪽 모터 속도데이터 (하위 8bit)	0

**Data ①** 명령 코드 : 0x06 (속도명령 2)

**Data ②** 오른쪽 모터 구동 방향

Forward 방향 : 0x46

Backward 방향 : 0x42

**Data ③** 오른쪽 모터 속도데이터 상위 8 비트

**Data ④** 오른쪽 모터 속도데이터 하위 8 비트

**Data ⑤** 왼쪽 모터 구동 방향

Forward 방향 : 0x46

Backward 방향 : 0x42

**Data ⑥** : 왼쪽 모터 속도데이터 상위 8 비트

**Data ⑦** : 왼쪽 모터 속도데이터 하위 8 비트

**Data ⑧** : 의미 없음

예 1) 오른쪽 모터는 **Forward** 방향으로 **050rad/sec** 의 속도로, 왼쪽 모터는 **Backward** 방향으로 **100rad/sec** 의 속도로 구동

오른쪽 모터 구동 방향은 **Forward** 방향이므로 **Data②** 는 **0x46** 이고

오른쪽 모터 속도 **050** 은 **16** 진수로 **0x0032** 이므로

오른쪽 모터 속도데이터의 상위 8 비트는 **0x00**

오른쪽 모터 속도데이터의 하위 8 비트는 **0x32**

왼쪽 모터 구동 방향은 **Backward** 방향이므로 **Data ⑤** 는 **0x42** 이고

왼쪽 모터 속도 **100** 은 **16** 진수로 **0x0064** 이므로

왼쪽 모터 속도데이터의 상위 8 비트는 **0x00**

왼쪽 모터 속도데이터의 하위 8 비트는 **0x64**

가 됩니다.

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x06</b>	<b>0x46</b>	<b>0x00</b>	<b>0x32</b>	<b>0x42</b>	<b>0x00</b>	<b>0x64</b>	<b>0</b>

예 2) 오른쪽 모터는 **Backward** 방향으로 **100rad/sec** 의 속도로, 왼쪽 모터는 **Forward** 방향으로 **050rad/sec** 의 속도로 구동

오른쪽 모터 구동 방향은 **Backward** 방향이므로 **Data②** 는 **0x42** 이고

오른쪽 모터 속도 **100** 은 **16** 진수로 **0x0064** 이므로

오른쪽 모터 속도데이터의 상위 8 비트는 **0x00**

오른쪽 모터 속도데이터의 하위 8 비트는 **0x64**

왼쪽 모터 구동 방향은 **Forward** 방향이므로 **Data ⑤** 는 **0x46** 이고

왼쪽 모터 속도 **050** 은 **16** 진수로 **0x0032** 이므로

왼쪽 모터 속도데이터의 상위 8 비트는 **0x00**

왼쪽 모터 속도데이터의 하위 8 비트는 **0x32**

가 됩니다.

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x06</b>	<b>0x42</b>	<b>0x00</b>	<b>0x64</b>	<b>0x46</b>	<b>0x00</b>	<b>0x32</b>	<b>0</b>

### 3. 속도명령3

- 이 명령은 좌/우 모터를 각각 다르게 구동시킬 수 있다는 점에서는 '속도명령2'와 같습니다. 다른 점은 이 속도명령3 을 전송할 때마다 휠의 이동 량과 플랫폼의 상태를 피드백 해준다는 것입니다.

(휠의 이동량과 플랫폼의 상태를 확인하기 위해 추가로 명령전송을 할 필요가 없다는 것이 이 명령의 이점입니다)

- STELLA B2를 Forward/Backward 로 한계속도(모터기준 270rad/sec, 휠 기준 5.2km/h) 안에서 구동 시킬 수 있습니다.

$$\ast (((270/(2\ast\pi))\ast 60\ast 60)/14)\ast 0.4725)/1000 = 5.2 \text{ km/h}$$

- 속도명령에서 속도단위는 휠이 아닌 Motor 기준의 회전속도입니다.

- 속도 단위는 [rad/sec] 입니다.

- 속도 설정 범위 : 000 ~ 270 (0x0000 ~ 0x010e)

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0x07)	오른쪽 모터 구 동 방향	오른쪽 모 터 속도데 이터	오른쪽 모 터 속도데 이터	왼쪽 모 터 구동 방향	왼쪽 모터 속도데이터 (상위 8bit)	왼쪽 모터 속도데이터 (하위 8bit)	0

Data ① 명령 코드 : 0x07 (속도명령 3)

Data ② 오른쪽 모터 구동 방향

Forward 방향 : 0x46

Backward 방향 : 0x42

Data ③ 오른쪽 모터 속도데이터 상위 8 비트

Data ④ 오른쪽 모터 속도데이터 하위 8 비트

Data ⑤ 왼쪽 모터 구동 방향

Forward 방향 : 0x46

Backward 방향 : 0x42

Data ⑥ : 왼쪽 모터 속도데이터 상위 8 비트

Data ⑦ : 왼쪽 모터 속도데이터 하위 8 비트

Data ⑧ : 의미없음

예 1) 오른쪽 모터는 Forward 방향으로 050rad/sec의 속도로, 왼쪽 모터는 Backward 방향으로 100rad/sec의 속도로 구동하고 피드백을 요청하는 명령

오른쪽 모터 구동 방향은 Forward 방향이므로 Data②는 0x46 이고

오른쪽 모터 속도 050은 16진수로 0x0032 이므로

오른쪽 모터 속도데이터의 상위 8 비트는 0x00

오른쪽 모터 속도데이터의 하위 8 비트는 0x32

왼쪽 모터 구동 방향은 Backward 방향이므로 Data ⑤는 0x42 이고

왼쪽 모터 속도 100은 16진수로 0x0064 이므로

왼쪽 모터 속도데이터의 상위 8 비트는 0x00

왼쪽 모터 속도데이터의 하위 8 비트는 0x64

가 됩니다.

Data ① Data ② Data ③ Data ④ Data ⑤ Data ⑥ Data ⑦ Data ⑧

**0x06      0x46      0x00      0x32      0x42      0x00      0x64      0**

예 2) 오른쪽 모터는 Backward 방향으로 100rad/sec 의 속도로, 왼쪽 모터는 Forward 방향으로 050rad/sec 의 속도로 구동

오른쪽 모터 구동 방향은 Backward 방향이므로 Data② 는 0x42 이고

오른쪽 모터 속도 100 은 16 진수로 0x0064 이므로

오른쪽 모터 속도데이터의 상위 8 비트는 0x00

오른쪽 모터 속도데이터의 하위 8 비트는 0x64

왼쪽 모터 구동 방향은 Forward 방향이므로 Data ⑤ 는 0x46 이고

왼쪽 모터 속도 050 은 16 진수로 0x0032 이므로

왼쪽 모터 속도데이터의 상위 8 비트는 0x00

왼쪽 모터 속도데이터의 하위 8 비트는 0x32

가 됩니다.

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x06</b>	<b>0x42</b>	<b>0x00</b>	<b>0x64</b>	<b>0x46</b>	<b>0x00</b>	<b>0x32</b>	<b>0</b>

- 응답 데이터 구성

응답값의 단위는 [millimeter]이며 응답 가능한 최대값은 8388607[mm] 입니다.

(각 휠의 이동 량에 대한 데이터 구성은 **7 비트의 상위데이터, 8 비트의 중위데이터, 8 비트의 하위데이터**로 구성되어있습니다. 그리고 **상위데이터의 최상위 비트는 방향**을 나타

내는 비트로 1 은 Backward 방향, 0 은 Forward 방향을 나타냅니다)

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
응답코드 <b>(0x93)</b>	방향데이터 <b>1bit</b> 와 오 른쪽 휠 이 동량 상위 <b>7bit</b>	오른쪽 휠 이동량 중위 <b>8bit</b>	오른쪽 휠 이동량 하위 <b>8bit</b>	방향데이터 <b>1bit</b> 와 왼쪽 휠 이 동량 상위 <b>7bit</b>	왼쪽 휠 이동량 중위 <b>8bit</b>	왼쪽 휠 이동량 하위 <b>8bit</b>	플랫폼의 상태 알림 데이터

응답 예 ) 현재 오른쪽 휠은 Backward 방향으로 59cm 이동했고 왼쪽 휠은 Forward 방향으로 259cm 만큼 이동했으며 플랫폼의 상태는 모터 회전중임을 나타낼 때

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x93</b>	<b>0x80</b>	<b>0x00</b>	<b>0x3b</b>	<b>0x00</b>	<b>0x01</b>	<b>0x03</b>	<b>0x07</b>

**Data ①** 응답 코드 : 0x93 (속도명령 3 에 대한 응답코드)

**Data ②** : 오른쪽 휠 이동량의 상위 7bit 와 함께 방향에 대한 데이터를 갖고 있다 (0x80)

0x80(16 진수) = 0b **1** 0 0 0 0 0 0 0 (2 진수)

방향을 나타내는 최상위 비트가 1 이므로 Backward 방향을 나타낸다.

**Data ③** : 오른쪽 휠 이동 량 데이터의 중위 8bit (0x00)

**Data ④** : 오른쪽 휠 이동 량 데이터의 하위 8bit (0x3b)

-> Data ②의 7 비트와 Data ③, ④의 각 8 비트를 합치면 0x00003b 이며 이는 10 진수로 59 를 나타낸다.

**Data ⑤** : 왼쪽 휠 이동 량의 상위 7bit 와 함께 방향에 대한 데이터를 가지고 있다. (0x00)

0x00(16 진수) = 0b 0 0 0 0 0 0 0 0 (2 진수)

방향을 나타내는 최상위 비트가 0 이므로 Forward 방향을 나타낸다.

**Data ⑥** : 왼쪽 휠 이동 량 데이터의 중위 8bit (0x01)

**Data ⑦** : 왼쪽 휠 이동 량 데이터의 하위 8bit (0x03)

-> Data ⑤의 7 비트와 Data ⑥, ⑦의 각 8 비트를 합치면 0x000103 이며 이는 10 진수로 259 를 나타낸다.

**Data ⑧** 플랫폼의 상태 코드 : 0x07 (모터 회전 중)

#### 4. 위치 명령

- STELLA B2 를 원하는 위치로 이동 및 회전시킬 수 있습니다.

- 위치 명령에는 회전각도, 이동거리, 이동시간이 입력으로 필요합니다

- 회전각도 범위 : 000 ~ 360 (0x0000 ~ 0x0168) [degree]

- 이동거리 범위 : 000 ~ 999 (0x0000 ~ 0x03e7) [centimeter]

- 이동시간 범위 : 000 ~ 255 (0x00 ~ 0xff) [second]

- CAN 데이터 구성 :

Data ① 명령코드 (0x02)	Data ② 회전 방향	Data ③ 회전각도 상위 8bit	Data ④ 회전각도 하위 8bit	Data ⑤ 구동 방향	Data ⑥ 이동거리 상위 8bit	Data ⑦ 이동거리 하위 8bit	Data ⑧ 이동 시간
--------------------------	-----------------	---------------------------	---------------------------	-----------------	---------------------------	---------------------------	-----------------

**Data ①** 명령 코드 : 0x02 (위치명령)

**Data ②** 회전 방향 :

왼쪽으로 회전 : 0x4c

오른쪽으로 회전 : 0x52

**Data ③** 회전각도 (상위 8bit)

**Data ④** 회전각도 (하위 8bit)

: 예를 들어 회전각도 90[°]는 16 진수로 0x005a 이며 이 16 진수의 상위 8bit 는 '0x00'이고 하위 8 비트는 '0x5a'입니다.

**Data ⑤** 구동 방향 :

Forward 방향으로 이동 : 0x46

Backward 방향으로 이동 : 0x42

**Data ⑥** 이동거리 (상위 8bit) : 100[cm] = 0x0064 의 상위 8bit -> 0x00

**Data ⑦** 이동거리 (하위 8bit) : 100[cm] = 0x0064 의 하위 8bit -> 0x64

**Data ⑧** 이동시간 : 5 [second]

예 1) 회전하지 않으면서 Forward 방향으로 5 초동안 1m 전진

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x02	0x52	0x00	0x00	0x46	0x00	0x64	0x05

또는

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x02	0x4C	0x00	0x00	0x46	0x00	0x64	0x05

예 2) 제자리에서 2초동안 오른쪽으로 90도 회전

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x02	0x52	0x00	0x5A	0x46	0x00	0x00	0x02

또는

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x02	0x52	0x00	0x5A	0x42	0x00	0x00	0x02

예 3) 제자리에서 2초동안 왼쪽으로 90도 회전

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x02	0x4C	0x00	0x5A	0x46	0x00	0x00	0x02

또는

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x02	0x4C	0x00	0x5A	0x42	0x00	0x00	0x02

## 5. 정지 명령

- 바퀴 정지 명령

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0x03)	정지모드	0	0	0	0	0	0

Data ① 명령 코드 : 0x03 (정지 명령)

Data ② 정지모드 : 0x01 ~ 0x03

0x01 : Motor Free 정지 (이 명령 후에는 초기화 명령이나 전원재입력 전  
까지 다른 명령을 받아들이지 않습니다)

0x02 : Motor Hold (순간정지)

0x03 : Motor Hold (감속하며 정지)

Data ③ ~ ⑧ : 의미 없음

예 1) 양쪽 휠 모두 정지(Motor Free) - CSTOP1

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
--------	--------	--------	--------	--------	--------	--------	--------

**0x03      0x01      0      0      0      0      0      0**

예 2) 양쪽 휠 모두 순간 정지(Motor Hold) - CSTOP2

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x03      0x02      0      0      0      0      0      0**

예 3) 양쪽 휠 모두 감속하며 정지(Motor Hold) - CSTOP3

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x03      0x03      0      0      0      0      0      0**

## 6. 리셋 명령

- 사용자가 설정한 파라미터 데이터로 리셋되며 파라미터들을 따로 설정하지 않았으면 기본 값들로 설정됩니다.

- CAN 데이터 구성 :

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**명령코드   리셋모드      0      0      0      0      0      0**  
**(0x04)**

**Data ①** 명령 코드 : 0x04 (리셋 명령)

**Data ②** 리셋모드 : 0x00 ~ 0x02

0x00 : 양쪽 휠에 대한 리셋

0x01 : 오른쪽 휠에 대한 리셋

0x02 : 왼쪽 휠에 대한 리셋

**Data ③ ~ ⑧** : 의미 없음

예 1) 양쪽 휠 모두 리셋

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x04      0x00      0      0      0      0      0      0**

예 2) 오른쪽 휠 리셋

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x04      0x01      0      0      0      0      0      0**

예 3) 왼쪽 휠 리셋

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x04      0x02      0      0      0      0      0      0**



## 7. 초기화 명령

- 유저가 설정한 파라미터 데이터가 아닌 제품 출하시의 기본 값들로 설정됩니다.

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0x05)	0	0	0	0	0	0	0

Data ① 명령 코드 : 0x05 (초기화 명령)

Data ② ~ ⑧: 의미 없음

예) 초기화

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x05	0	0	0	0	0	0	0

## ■ GET 명령

### 8. STELLA B2의 현재 상태 값 읽어오기

- 응답 값을 확인하여 STELLA B2의 상태를 확인할 수 있습니다.

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
--------	--------	--------	--------	--------	--------	--------	--------

명령코드            0            0            0            0            0            0            0  
**(0x40)**

**Data ①** 명령 코드 : 0x40 (상태 읽기 명령)

**Data ② ~ ⑧** : 의미 없음

예) 상태확인

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x40	0	0	0	0	0	0	0

- 응답 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
응답코드	상태코드	0	0	0	0	0	0
<b>(0x80)</b>							

**Data ①** 응답 코드 : 0x80 (상태 읽기 명령에 대한 응답임을 나타내는 코드)

**Data ② ~ ⑧** : 의미 없음

응답 예 1) 0x00- 정상 상태

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x80	0x00	0	0	0	0	0	0

응답 예 2) 0x01 - 오른쪽 휠 과부하 상태

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x80	0x01	0	0	0	0	0	0

응답 예 3) 0x02 - 왼쪽 휠 과부하 상태

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x80	0x02	0	0	0	0	0	0

응답 예 4) 0x03 - 양쪽 휠 과부하 상태

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x80	0x03	0	0	0	0	0	0

응답 예 5) 0x04 - 과전압 상태 (인가 전압 15V 이상)

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x80	0x04	0	0	0	0	0	0

응답 예 6) 0x05 - 저전압 상태 (인가 전압 10V 미만)

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x80	0x05	0	0	0	0	0	0

응답 예 7) 0x06 - 목표위치 도달 상태 (위치 명령시에만 나타나는 상태임)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x80	0x06	0	0	0	0	0	0

응답 예 8) 0x07 - 휠 회전 중인 상태 (휠이 움직이고 있는 상태)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x80	0x07	0	0	0	0	0	0

※ 과부하, 과전압, 저전압은 그 상태가 2초이상 지속되었을 때 알람이 발생하며 휠을 정지상태로 만들며 이때는 휠 구동 명령에 반응하지 않습니다. 이 세가지 상태에 대한 알람이 발생하면 STELLA B2의 전원을 OFF 한 후 알람 상황을 제거해 주고 전원을 다시 입력하시기 바랍니다.

또한 이 세가지 알람에 대해서는 모션 드라이브의 LED로도 확인이 가능합니다.

과전류 알람일 때 붉은색 LED가 주기적으로 3번 점멸

과전압 알람일 때 붉은색 LED가 주기적으로 4번 점멸

저전압 알람일 때 붉은색 LED가 주기적으로 5번 점멸

위치 명령에 의해 목표위치에 도달했을 때에는 모션 드라이브의 파란색 LED가 점등되며 STELLA B2가 구동 중이거나 위치 명령이 아닐 경우에는 파란색 LED는 OFF 되어있습니다.

## 9. STELLA B2의 현재 속도 읽어오기

- 응답 값을 확인하여 STELLA B2의 속도를 확인할 수 있습니다.

- 응답 속도 값은 모터의 회전속도를 나타내며 단위는 rad/sec 입니다.

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0x41)	0	0	0	0	0	0	0

Data ① 명령 코드 : 0x41 (속도 읽기 명령)

Data ② ~ ⑧ : 의미 없음

예) 모터 속도 읽어오기

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x41	0	0	0	0	0	0	0

- 응답 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
--------	--------	--------	--------	--------	--------	--------	--------

응답코드 <b>(0x81)</b>	오른쪽 휠 구동방향	오른쪽 모 터 속도 상위 <b>8bit</b>	오른쪽 모 터 속도 하위 <b>8bit</b>	왼쪽 휠 구동방향	왼쪽 모터 속도 상위 <b>8bit</b>	왼쪽 모터 속도 하위 <b>8bit</b>	<b>0</b>
-----------------------	---------------	---------------------------------	---------------------------------	--------------	-------------------------------	-------------------------------	----------

응답 예) 오른쪽 모터는 Forward 방향으로 100.0 rad/sec 의 속도로 회전하고 있으며 왼쪽 모  
터는 Backward 방향으로 200rad/sec 의 속도로 회전하고 있음

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x81</b>	<b>0x46</b>	<b>0x00</b>	<b>0x64</b>	<b>0x42</b>	<b>0x00</b>	<b>0xc8</b>	<b>0</b>

**Data ①** 응답 코드 : 0x81 (속도 읽기 명령에 대한 응답코드)

**Data ②** 오른쪽 휠 구동방향 : 0x46 -> Forward 방향, 0x42 -> Backward 방향

**Data ③** 오른쪽 모터 속도 데이터의 상위 8bit (0x00)

**Data ④** 오른쪽 모터 속도 데이터의 하위 8bit (0x64)

-> 0x0064 는 10 진수로 100 [rad/sec]

**Data ⑤** 왼쪽 휠 구동방향 : 0x46 -> Forward 방향, 0x42 -> Backward 방향

**Data ⑥** 왼쪽 모터 속도 데이터의 상위 8bit (0x00)

**Data ⑦** 왼쪽 모터 속도 데이터의 하위 8bit (0xc8)

-> 0x00c8 는 10 진수로 200 [rad/sec]

**Data ⑧** : 의미 없음

## 10. STELLA B2의 이동한 거리 읽어오기

- 응답 값을 확인하여 STELLA B2 가 이동한 거리를 확인할 수 있습니다.

- 응답 값은 STELLA B2 각 휠의 이동량을 나타냅니다.

- 응답 값의 단위는 [millimeter]이며 응답 가능한 최대값은 8388607[mm] 입니다.

- CAN 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
명령코드 <b>(0x42)</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Data ①** 명령 코드 : 0x42 (이동량 읽기 명령)

**Data ② ~ ⑧** : 의미 없음

예) 이동량 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x42</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

- 응답 데이터 구성

(각 휠의 이동량에 대한 데이터 구성은 7 비트의 상위데이터, 8 비트의 중위데이터, 8 비트  
의 하위데이터로 구성되어있습니다. 그리고 상위데이터의 최상위 비트는 방향을 나타내

는 비트로 1 은 Backward 방향, 0 은 Forward 방향을 나타냅니다)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
응답코드 (0x82)	방향데이터 1bit 와 오 른쪽 휠 이 동량 상위 7bit	오른쪽 휠 이동량 중위 8bit	오른쪽 휠 이동량 하위 8bit	방향데이터 1bit 와 왼쪽 휠 이 동량 상위 7bit	왼쪽 휠 이 동량 중위 8bit	왼쪽 휠 이 동량 하위 8bit	0

응답 예 ) 현재 오른쪽 휠은 Backward 방향으로 59cm 이동했고 왼쪽 휠은 Forward 방향으로 259cm 만큼 이동했음

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x82	0x80	0x00	0x3b	0x00	0x01	0x03	0

Data ① 응답 코드 : 0x82 (이동량 읽기 명령에 대한 응답코드)

Data ② : 오른쪽 휠 이동량의 상위 7bit 와 함께 방향에 대한 데이터를 갖고있다 (0x80)

$$0x80(16 \text{ 진수}) = 0b \underline{1} 0 0 0 0 0 0 0 (2 \text{ 진수})$$

방향을 나타내는 최상위 비트가 1 이므로 Backward 방향을 나타낸다.

Data ③ : 오른쪽 휠 이동량 데이터의 중위 8bit (0x00)

Data ④ : 오른쪽 휠 이동량 데이터의 하위 8bit (0x3b)

-> Data ② 의 7 비트와 Data ③, ④ 의 각 8 비트를 합치면 0x00003b 이며 이는 10 진수로 59 를 나타낸다.

Data ⑤ : 왼쪽 휠 이동량의 상위 7bit 와 함께 방향에 대한 데이터를 갖고있다 (0x00)

$$0x00(16 \text{ 진수}) = 0b \underline{0} 0 0 0 0 0 0 0 (2 \text{ 진수})$$

방향을 나타내는 최상위 비트가 0 이므로 Forward 방향을 나타낸다.

Data ⑥ : 왼쪽 휠 이동량 데이터의 중위 8bit (0x01)

Data ⑦ : 왼쪽 휠 이동량 데이터의 하위 8bit (0x03)

-> Data ⑤ 의 7 비트와 Data ⑥, ⑦ 의 각 8 비트를 합치면 0x000103 이며 이는 10 진수로 259 를 나타낸다.

Data ⑧ : 의미 없음

## 11. 회전 Scale Factor 값 읽어오기

- 응답 값을 확인하여 STELLA B2 의 회전 Scale Factor 값을 확인할 수 있습니다.

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0x43)	0	0	0	0	0	0	0

Data ① 명령 코드 : 0x43 (회전 스케일팩터 읽기 명령)

Data ② ~ ⑧ : 의미 없음

예 ) 회전 스케일팩터 읽어오기

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
--------	--------	--------	--------	--------	--------	--------	--------

**0x43      0      0      0      0      0      0      0**

- 응답 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
응답코드 <b>(0x83)</b>	오른쪽 휠 회전 스케 일팩터 상위 <b>8bit</b>	오른쪽 휠 회전 스케 일팩터 하위 <b>8bit</b>	왼쪽 휠 회 전 스케일 팩터 상위 <b>8bit</b>	왼쪽 휠 회 전 스케일 팩터 하위 <b>8bit</b>	<b>0</b>	<b>0</b>	<b>0</b>

응답 예 )

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x83</b>	<b>0x00</b>	<b>0x01</b>	<b>0x00</b>	<b>0x02</b>	<b>0</b>	<b>0</b>	<b>0</b>

오른쪽 휠 회전 스케일팩터 = (Data ② << 8) | (Data ③)  
= 0x0001 (10 진수로는 1 을 나타냄)

가중치 = (1 + (스케일팩터 / 1000)) = 1.001

이 가중치의 의미는 STELLA B2 가 회전할 때 오른쪽 휠에 약 0.1%만큼 출력을 더 높인다는 의  
미입니다.

왼쪽 휠 회전 스케일팩터 = (Data ④ << 8) | (Data ⑤)  
= 0x0002 (10 진수로는 2 를 나타냄)

가중치 = (1 + (스케일팩터 / 1000)) = 1.002

이 가중치의 의미는 STELLA B2 가 회전할 때 왼쪽 휠에 약 0.2%만큼 출력을 더 높인다는 의미  
입니다.

## 12. 주행 Scale Factor 값 읽어오기

- 응답 값을 확인하여 STELLA B2 의 회전 Scale Factor 값을 확인할 수 있습니다.

- CAN 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
명령코드 <b>(0x44)</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Data ①** 명령 코드 : 0x44 (주행 스케일팩터 읽기 명령)

**Data ② ~ ⑧** : 의미 없음

예 ) 주행 스케일팩터 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x44</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

- 응답 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
응답코드 (0x84)	오른쪽 휠 주행 스케일 일팩터 상위 Sbit	오른쪽 휠 주행 스케일 일팩터 하위 Sbit	왼쪽 휠 주 행 스케일 팩터 상위 Sbit	왼쪽 휠 주 행 스케일 팩터 하위 Sbit	0	0	0

응답 예 )

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x84	0x00	0x01	0x00	0x02	0	0	0

$$\begin{aligned} \text{오른쪽 휠 주행 스케일팩터} &= (\text{Data ②} \ll 8) | (\text{Data ③}) \\ &= 0x0001 \text{ (10 진수로는 1 을 나타냄)} \end{aligned}$$

$$\text{가중치} = (1 + (\text{스케일팩터} / 1000)) = 1.001$$

이 가중치의 의미는 STELLA B2 가 주행할 때 오른쪽 휠에 약 0.1%만큼 출력을 더 높인다는 의미입니다.

$$\begin{aligned} \text{왼쪽 휠 주행 스케일팩터} &= (\text{Data ④} \ll 8) | (\text{Data ⑤}) \\ &= 0x0002 \text{ (10 진수로는 2 를 나타냄)} \end{aligned}$$

$$\text{가중치} = (1 + (\text{스케일팩터} / 1000)) = 1.002$$

이 가중치의 의미는 STELLA B2 가 주행할 때 왼쪽 휠에 약 0.2%만큼 출력을 더 높인다는 의미입니다.

### 13. 가/감속 값 읽어오기 [단위 : rad/sec^2]

- 응답 값을 확인하여 STELLA B2 의 속도 명령시 이용되는 가/감속 값을 확인할 수 있습니다.

- 응답 값은 모터의 회전 속도에 관한 가/감속 값이며 단위는 rad/sec^2 입니다

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0x45)	대상 모터	0	0	0	0	0	0

Data ① 명령 코드 : 0x44 (주행 스케일팩터 읽기 명령)

Data ② 대상 모터 : 0x01 (오른쪽 모터), 0x02 (왼쪽 모터)

Data ③ ~ ⑧ : 의미 없음

예 1) 오른쪽 모터 가감속 데이터 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x45</b>	<b>0x01</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

예 2) 왼쪽 모터 가감속 데이터 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x45</b>	<b>0x02</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

- 응답 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
응답코드 <b>(0x85)</b>	대상 모터	가속값 정 수부	<b>0x2e'</b>	가속값 소 수부	감속값 정 수부	<b>0x2e</b>	감속값 소 수부

응답 예 )

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x85</b>	<b>0x01</b>	<b>0x00</b>	<b>0x2e</b>	<b>0x04</b>	<b>0x00</b>	<b>0x2e</b>	<b>0x01</b>

**Data ①** 응답 코드 : 0x85 (가감속 읽기 명령)

**Data ②** 대상 모터 : 0x01 (오른쪽 모터)

**Data ③** 가속값 정수부 : 0x00 = 0 (10 진수)

**Data ④** : 항상 0x2e 데이터를 갖고 있지만 데이터 자체가 의미를 갖고 있지 않음  
(단지, 정수부와 소수부 구분을 위해 사용된 것임)

**Data ⑤** : 가속값 소수부 : 0x04 = 4 (10 진수)

-> 가속값 = 0.4 [rad/sec<sup>2</sup>]

**Data ⑥** : 감속값 정수부 : 0x00 = 0 (10 진수)

**Data ⑦** : 항상 0x2e 데이터를 갖고 있지만 데이터 자체가 의미를 갖고 있지 않음  
(단지, 정수부와 소수부 구분을 위해 사용된 것임)

**Data ⑧** : 감속값 소수부 : 0x01 = 1 (10 진수)

-> 감속값 = 0.1 [rad/sec<sup>2</sup>]

#### 14. STELLA B2의 속도/위치제어기의 P/I/D/IL 계인 읽어오기

- 응답 값을 확인하여 STELLA B2의 각각의 속도/위치 제어기에서 이용되는 P/I/D/IL 계인 값을 확인할 수 있습니다.

- CAN 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
명령코드	대상 모터	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Data ①** 명령 코드 :





- 0x47 (속도 제어기 P 게인 읽기 명령)
- 0x48 (속도 제어기 I 게인 읽기 명령)
- 0x49 (속도 제어기 D 게인 읽기 명령)
- 0x4a (속도 제어기 IL 게인 읽기 명령)
- 0x4b (위치 제어기 P 게인 읽기 명령)
- 0x4c (위치 제어기 I 게인 읽기 명령)
- 0x4d (위치 제어기 D 게인 읽기 명령)
- 0x4e (위치 제어기 IL 게인 읽기 명령)

**Data ②** 대상 모터 :

0x01 (오른쪽 모터), 0x02 (왼쪽 모터)

**Data ③ ~ ⑧** : 의미 없음

예 1-1) 오른쪽 모터 속도 제어기 P 게인 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x47	0x01	0	0	0	0	0	0

예 1-2) 오른쪽 모터 속도 제어기 I 게인 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x48	0x01	0	0	0	0	0	0

예 1-3) 오른쪽 모터 속도 제어기 D 게인 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x49	0x01	0	0	0	0	0	0

예 1-4) 오른쪽 모터 속도 제어기 P 게인 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x4A	0x01	0	0	0	0	0	0

예 2-1) 왼쪽 모터 속도 제어기 P 게인 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x47	0x02	0	0	0	0	0	0

예 2-2) 왼쪽 모터 속도 제어기 I 게인 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x48	0x02	0	0	0	0	0	0

예 2-3) 왼쪽 모터 속도 제어기 D 게인 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

**0x49      0x02      0      0      0      0      0      0**

예 2-4) 왼쪽 모터 속도 제어기 P 계인 읽어오기

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x4A      0x02      0      0      0      0      0      0**

예 3-1) 오른쪽 모터 위치 제어기 P 계인 읽어오기

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x4B      0x01      0      0      0      0      0      0**

예 3-2) 오른쪽 모터 위치 제어기 I 계인 읽어오기

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x4C      0x01      0      0      0      0      0      0**

예 3-3) 오른쪽 모터 위치 제어기 D 계인 읽어오기

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x4D      0x01      0      0      0      0      0      0**

예 3-4) 오른쪽 모터 위치 제어기 P 계인 읽어오기

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x4E      0x01      0      0      0      0      0      0**

예 4-1) 왼쪽 모터 위치 제어기 P 계인 읽어오기

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x4B      0x02      0      0      0      0      0      0**

예 4-2) 왼쪽 모터 위치 제어기 I 계인 읽어오기

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x4C      0x02      0      0      0      0      0      0**

예 4-3) 왼쪽 모터 위치 제어기 D 계인 읽어오기

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x4D      0x02      0      0      0      0      0      0**

예 4-4) 왼쪽 모터 위치 제어기 P 계인 읽어오기

**Data ①   Data ②   Data ③   Data ④   Data ⑤   Data ⑥   Data ⑦   Data ⑧**  
**0x4E      0x02      0      0      0      0      0      0**

- 응답 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
응답코드	응답 모터	게인값 정수부 상위 8비트	게인값 정수부 하위 8비트	정수부와 소수부 구분점 (0x2e)	게인값 소수부 상위 8비트	게인값 소수부 하위 8비트	0

**Data ①** 응답 코드 :

- 0x87 (속도 제어기 P 게인 응답)
- 0x88 (속도 제어기 I 게인 응답)
- 0x89 (속도 제어기 D 게인 응답)
- 0x8a (속도 제어기 IL 게인 응답)
- 0x8b (위치 제어기 P 게인 응답)
- 0x8c (위치 제어기 I 게인 응답)
- 0x8d (위치 제어기 D 게인 읽기 명령)
- 0x8e (위치 제어기 IL 게인 읽기 명령)

**Data ②** 응답 모터 :

- 0x01 (오른쪽 모터), 0x02 (왼쪽 모터)

**Data ③** : 게인값 정수부의 상위 8 비트

**Data ④** : 게인값 정수부의 하위 8 비트

$$\rightarrow \text{게인값 정수부} = ((\text{Data ③} \ll 8) | \text{Data ④})$$

**Data ⑤** : 게인값 정수부와 소수부 구분점

이 데이터는 0x2e 로 고정이지만 의미를 갖는 데이터가 아니다.

**Data ⑥** : 게인값 소수부의 상위 8 비트

**Data ⑦** : 게인값 소수부의 하위 8 비트

$$\rightarrow \text{게인값 소수부} = ((\text{Data ⑥} \ll 8) | \text{Data ⑦})$$

**Data ⑧** : 의미 없음

응답 예 1) 오른쪽 모터 위치제어기의 p 게인값에 대한 응답 (응답 P 게인 값 = 4.472)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x8b	0x01	0x00	0x04	0x2e	0x01	0xd8	0

**Data ①** 응답 코드 : 0x8b (위치제어기 P 게인 값 응답임을 나타내는 코드)

**Data ②** 대상 모터 : 0x01 (오른쪽 모터)

**Data ③** 게인 값 정수부 상위 8 비트 : 0x00

**Data ④** 게인 값 정수부 하위 8 비트 : 0x04

$$\begin{aligned} \rightarrow \text{게인 값 정수부} &= ((\text{Data ③} \ll 8) | \text{Data ④}) \\ &= (0x00 \ll 8) | 0x04 \\ &= 0x0004 \\ &= 10 \text{ 진수로 '4'} \end{aligned}$$

**Data ⑤** : 항상 0x2e 데이터를 갖고 있지만 데이터 자체가 의미를 갖고 있지 않음  
(단지, 정수부와 소수부 구분을 위해 사용된 것임)

**Data ⑥** : 개인 값 소수부 상위 8 비트 : 0x01

**Data ⑦** : 개인 값 소수부 상위 8 비트 : 0xd8

$$\begin{aligned} \text{-> 개인 값 소수부} &= ((\text{Data ⑥} \ll 8) \mid \text{Data ⑦}) \\ &= (0x01 \ll 8) \mid 0xd8 \\ &= 0x01d8 \\ &= 10 \text{ 진수로 '472'} \end{aligned}$$

개인 값 정수부는 '4'이고 소수부는 '472' 이므로 개인 값은 4.472임을 알 수 있음.

**Data ⑧** : 의미없음

### 15. 설정된 휠 반지름 값 읽어오기

- 응답 값을 확인하여 STELLA B2 제어기에 설정되어있는 휠 반지름 값을 확인할 수 있습니다.

- 응답 데이터의 단위는 [1/100 centimeter]입니다.

- CAN 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
명령코드 (0x4f)	0	0	0	0	0	0	0

**Data ①** 명령 코드 : 0x4f (설정된 휠 반지름 읽기)

**Data ② ~ ⑧** : 의미 없음

예) 설정된 휠 반지름 읽기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x4f	0	0	0	0	0	0	0

- 응답 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
응답코드 (0x8f)	휠 반지름 상위 8bit	휠 반지름 하위 8bit	0	0	0	0	0

**Data ①** 응답 코드 : 0x8f (반지름 응답)

**Data ②** : 설정된 휠 반지름의 상위 8 비트

**Data ③** : 설정된 휠 반지름의 하위 8 비트

$$\text{-> 휠 반지름} = (\text{Data ②} \ll 8) \mid \text{Data ③}$$

**Data ④ ~ ⑧** : 의미 없음

응답 예) 설정된 반지름이 7.52cm(기본설정값)인 경우

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x8f	0x02	0xf0	0	0	0	0	0

**Data ①** 응답 코드 : 0x8f (반지름 응답)

**Data ②** 설정된 휠 반지름의 상위 8 비트 : 0x02

**Data ③** 설정된 휠 반지름의 하위 8 비트 : 0xf0

$$\begin{aligned} \rightarrow \text{휠 반지름} &= (\text{Data ②} \ll 8) | \text{Data ③} \\ &= (0x02 \ll 8) | 0xf0 \\ &= 0x02f0 \\ &= 10 \text{ 진수로 } 752 \text{ [단위 : } \mathbf{1/100cm} \text{]} \\ &= 7.52 \text{ cm} \end{aligned}$$

**Data ④ ~ ⑧** : 의미 없음

### 16. 설정된 두 휠 사이의 너비 읽어오기

- 응답 값을 확인하여 STELLA B2 제어기에 설정되어있는 두 휠 사이의 너비 값을 확인할 수 있습니다.

- 응답 코드의 단위는 [1/100 centi meter]입니다.

- CAN 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
명령코드 (0x50)	0	0	0	0	0	0	0

**Data ①** 명령 코드 : 0x50 (설정된 두 휠 사이의 너비 읽기)

**Data ② ~ ⑧** : 의미 없음

예) 설정된 두 휠 사이의 너비 값 읽어오기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
0x50	0	0	0	0	0	0	0

- 응답 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
응답코드 (0x90)	휠 사이 너비 상위 8bit	휠 사이 너비 하위 8bit	0	0	0	0	0

**Data ①** 응답 코드 : 0x90 (휠 사이의 너비 응답)

**Data ②** : 설정된 두 휠 사이의 너비 상위 8 비트

**Data ③** : 설정된 두 휠 사이의 너비 하위 8 비트

$$\rightarrow \text{너비} = (\text{Data ②} \ll 8) | \text{Data ③}$$

**Data ④ ~ ⑧** : 의미 없음

응답 예) 두 휠 사이의 너비값이 29cm(기본설정값)인 경우

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x90	0x0b	0x54	0	0	0	0	0

Data ① 응답 코드 : 0x90 (너비 응답)

Data ② 설정된 휠 반지름의 상위 8 비트 : 0x0b

Data ③ 설정된 휠 반지름의 하위 8 비트 : 0x54

$$\begin{aligned}
 \text{-> 너비} &= (\text{Data ②} \ll 8) \mid \text{Data ③} \\
 &= (0x0b \ll 8) \mid 0x54 \\
 &= 0x0b54 \\
 &= 10 \text{ 진수로 } 2900 \text{ [단위 : } \mathbf{1/100\text{cm}}] \\
 &= \mathbf{29 \text{ cm}}
 \end{aligned}$$

Data ④ ~ ⑧ : 의미 없음

### 17. STELLA B2 모션드라이버의 펌웨어 버전 읽어오기

- 응답 값을 확인하여 STELLA B2 모션드라이버의 펌웨어 버전을 확인할 수 있습니다.

- 펌웨어 버전은 정수 한자리, 소수점 두 자리로 표현됩니다. (예 : 2.00)

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0x52)	0	0	0	0	0	0	0

Data ① 명령 코드 : 0x52 (펌웨어 버전 읽기)

Data ② ~ ⑧ : 의미 없음

예) 설정된 휠 반지름 읽기

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x52	0	0	0	0	0	0	0

- 응답 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
응답코드 (0x92)	펌웨어 버 전정보 정수부	정수부와 소수부 구 분점 (0x2e)	펌웨어 버 전정보 소수부	0	0	0	0

Data ① 응답 코드 : 0x92 (펌웨어 버전 응답)

Data ② : 펌웨어 버전정보 정수부

Data ③ : 항상 0x2e 데이터를 갖고있지만 데이터 자체가 의미를 갖고 있지 않음  
(단지, 정수부와 소수부 구분을 위해 사용된 것임)

**Data ④** : 펌웨어 버전정보 소수부

**Data ⑤ ~ ⑧** : 의미 없음

응답 예) 펌웨어 버전이 2.00 인 경우

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x92</b>	<b>0x02</b>	<b>0x2e</b>	<b>0x00</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Data ①** 응답 코드 : 0x90 (너비 응답)

**Data ②** 펌웨어 버전정보 정수부 : 0x02 (10 진수로 '2')

**Data ③** 0x2e : 정수부 소수부 구분점

**Data ④** 펌웨어 버전정보 소수부 : 0x00 (10 진수로 '0')

$$\begin{aligned} \text{-> 펌웨어 버전 : (Data ②) + (Data ③ / 100)} \\ &= 2 + 0.00 \\ &= 2.00 \end{aligned}$$

**Data ⑤ ~ ⑧** : 의미 없음

## 18. 엔코더 데이터 읽어오기

- 응답 값을 확인하여 엔코더 데이터를 확인 할 수 있습니다.

- CAN 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
명령코드 <b>(0x54)</b>	대상 모터	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Data ①** 명령 코드 : 0x54 (엔코더 데이터 읽기)

**Data ②** 대상 모터 :

0x01 : 오른쪽 모터

0x02 : 왼쪽 모터

**Data ③ ~ ⑧** : 의미 없음

예) 오른쪽 엔코더 데이터 읽기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x54</b>	<b>0x01</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

예) 왼쪽 엔코더 데이터 읽기

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0x54</b>	<b>0x02</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

- 응답 데이터 구성 : (엔코더 데이터 표현에는 4 바이트(32 비트)를 사용함)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
응답코드 (0x94)	응답 모 터	엔코더 데이 터 방향(부 호)	엔코더 데이 터 (32~25 비트)	엔코더 데이 터 (24~17 비트)	엔코더 데이 터 (16~9 비트)	엔코더 데이 터 (8~1 비트)	0

Data ① 응답 코드 : 0x94 (엔코더 데이터 응답)

Data ② 응답 모터 :

0x01 : 오른쪽 모터

0x02 : 왼쪽 모터

Data ③ : 엔코더 방향(부호)데이터

0x42 : STELLA B2 가 처음 전원이 켜졌던 지점으로부터 모터가 Backward 방  
향

0x46 : STELLA B2 가 처음 전원이 켜졌던 지점으로부터 모터가 Forward 방  
향

Data ④ : 엔코더 데이터 (32 비트 데이터 중에 32~25 비트)

Data ⑤ : 엔코더 데이터 (32 비트 데이터 중에 24~17 비트)

Data ⑥ : 엔코더 데이터 (32 비트 데이터 중에 16~9 비트)

Data ⑦ : 엔코더 데이터 (32 비트 데이터 중에 8~1 비트)

Data ⑧ : 의미 없음

응답 예 ) 오른쪽 모터의 엔코더 데이터가 Forward 방향으로 93324pulse 만큼 이동했음을 나  
타내는 경우

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0x94	0x01	0x46	0x00	0x01	0x6c	0x8c	0

Data ① 응답 코드 : 0x94 (엔코더 데이터 응답)

Data ② 0x01 : 오른쪽 모터

Data ③ 0x46 : 처음 전원이 켜졌던 지점으로부터 Forward 방향으로 회전(이동)해 있는  
상태임.

Data ④ ~ Data ⑦ : 이 4 바이트는 엔코더 카운트 데이터를 나타냄.

$$\text{엔코더 데이터} = (\text{Data ④} \ll 24) | (\text{Data ⑤} \ll 16) | (\text{Data ⑥} \ll 8) |$$

Data ⑦

$$= (0x00 \ll 24) | (0x01 \ll 16) | (0x6c \ll 8) | (0x8c)$$

$$= 0x00016c8c$$

$$= 10 \text{ 진수로 } 93324$$

Data ⑧ : 의미 없음

※ 읽은 엔코더 펄스로부터 이동량 계산하기

$$\text{이동량 [meter]} =$$



$$\left( \left( \left( 2 \times \pi \right) / \left( \text{엔코더 CPR} \times 4 \right) \right) \times \text{읽은 엔코더 데이터} \right) / \text{기어비} \times \text{휠 반지름[meter]}$$

**STELLA B2의 엔코더 CPR : 256**

**STELLA B2 모터의 기어비 : 13.71**

회전(이동)량 =  $\left( \left( \left( 2 \times 3.14 \right) / \left( 256 \times 4 \right) \right) \times 93324 \right) / 13.71 \times 0.0752$

= 약 3.14 [meter]

## ■ SET 명령

### 19. 허용 제한 전류값 설정하기

- 단위는 [mA]이며 기본 설정값은 2500mA 입니다.
- 최대 4000mA 까지 설정할 수 있습니다.
- 설정 범위 : 0000 ~ 4000 (0x0000 ~ 0x0fa0)
- 이 허용 제한 전류값 이상의 전류가 2 초이상 흐르면 과부하 알람이 발생합니다.
- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0xc0)	제한전류 상위 8비트	제한전류 하위 8비트	0	0	0	0	0

**Data ①** 명령 코드 : 0xc0 (제한 전류값 설정)

**Data ②** : 제한 전류값의 상위 8 비트 데이터

**Data ③** : 제한 전류값의 하위 8 비트 데이터

**Data ④ ~ ⑧** : 의미 없음

예 ) 허용 제한 전류값을 2500mA(2.5A)로 설정하기

2500 = 0x09c4

상위 8 비트 데이터 : 0x09

하위 8 비트 데이터 : 0xc4							
Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc0	0x09	0xc4	0	0	0	0	0

## 20. 회전 Scale Factor 값 설정하기

- 회전 스케일팩터는 기본적으로 설정된 값은 '000'입니다.
- 스케일팩터의 의미에 대해서는 'CAN 구동명령 커맨드' 장의 첫 페이지를 참고하시기 바랍니다.
- 설정범위 : 000 ~ 999 (0x0000 ~ 0x03e7)

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0xc1)	오른쪽 스 케일팩터 상위 8 비트	오른쪽 스케 일팩터 하위 8 비트	왼쪽 스케일 팩터 상위 8 비트	왼쪽 스케일 팩터 하위 8 비트	0	0	0

**Data ①** 명령 코드 : 0xc1 (회전 스케일팩터 설정)

**Data ②** : 오른쪽 모터의 회전 스케일팩터 상위 8 비트 데이터

**Data ③** : 오른쪽 모터의 회전 스케일팩터 하위 8 비트 데이터

**Data ④** : 왼쪽 모터의 회전 스케일팩터 상위 8 비트 데이터

**Data ⑤** : 왼쪽 모터의 회전 스케일팩터 하위 8 비트 데이터

**Data ⑥ ~ ⑧** : 의미 없음

예 ) 오른쪽 모터 회전 스케일팩터는 '123'으로 설정하고 왼쪽 모터 회전 스케일팩터는 '456'으로 설정하기

오른쪽 모터 회전 스케일팩터 123 = 0x007b

상위 8 비트 데이터 : 0x00

하위 8 비트 데이터 : 0x7b

왼쪽 모터 회전 스케일팩터 456 = 0x01c8

상위 8 비트 데이터 : 0x01

하위 8 비트 데이터 : 0xc8

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc1	0x00	0x7b	0x01	0xc8	0	0	0

## 21. 주행 Scale Factor 값 설정하기

- 주행 스케일팩터는 기본적으로 설정된 값은 '000'입니다.

- 스케일팩터의 의미에 대해서는 'CAN 구동명령 커맨드' 장의 첫페이지를 참고하시기 바랍니다.

- 설정범위 : 000 ~ 999 (0x0000 ~ 0x03e7)

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0xc2)	오른쪽 스케일팩터 상위 8 비트	오른쪽 스케일팩터 하위 8 비트	왼쪽 스케일팩터 상위 8 비트	왼쪽 스케일팩터 하위 8 비트	0	0	0

Data ① 명령 코드 : 0xc2 (주행 스케일팩터 설정)

Data ② : 오른쪽 모터의 주행 스케일팩터 상위 8 비트 데이터

Data ③ : 오른쪽 모터의 주행 스케일팩터 하위 8 비트 데이터

Data ④ : 왼쪽 모터의 주행 스케일팩터 상위 8 비트 데이터

Data ⑤ : 왼쪽 모터의 주행 스케일팩터 하위 8 비트 데이터

Data ⑥ ~ ⑧ : 의미 없음

예 ) 오른쪽 모터 주행 스케일팩터는 123 으로 설정하고 왼쪽 모터 주행 스케일팩터는 456 으로 설정하기

오른쪽 모터 주행 스케일팩터 123 = 0x007b

상위 8 비트 데이터 : 0x00

하위 8 비트 데이터 : 0x7b

왼쪽 모터 주행 스케일팩터 456 = 0x01c8

상위 8 비트 데이터 : 0x01

하위 8 비트 데이터 : 0xc8

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc2	0x00	0x7b	0x01	0xc8	0	0	0

## 22. 가/감속 값 설정하기

- 모터의 속도에 대한 가/감속 값이며 단위는 rad/sec<sup>2</sup> 입니다.

- 기본적으로 설정되어있는 값은 가속: 0.4rad/sec<sup>2</sup>, 감속: 0.1rad/sec<sup>2</sup> 입니다.

- 설정 범위 : 0.0 ~ 9.9 (0x00.0x00 ~ 0x09.0x09)

- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0xc3)	대상 모터	가속값 정수부	0x2e	가속값 소수부	감속값 정수부	0x2e	감속값 소수부

Data ① 명령 코드 : 0xc3 (가/감속 데이터 설정)

Data ② 대상 모터 : 0x01(오른쪽 모터), 0x02(왼쪽 모터)

**Data** ③ 가속값의 정수부

**Data** ④ 0x2e : 정수부와 소수부 구분 점으로 데이터 자체가 가지고 있는 의미는 없음

**Data** ⑤ 가속값의 소수부

**Data** ⑥ 감속값의 정수부

**Data** ⑦ 0x2e : 정수부와 소수부 구분 점으로 데이터 자체가 가지고 있는 의미는 없음

**Data** ⑧ 감속값의 소수부

예 ) 오른쪽 모터의 가속값은 0.4 [rad/sec^2] 감속값은 0.1 [rad/sec^2]으로 설정하기

대상 모터는 0x01 (오른쪽 모터)

가속값 0.4 의 정수부 : 0 (16 진수로 0x00)

가속값 0.4 의 소수부 : 4 (16 진수로 0x04)

감속값 0.1 의 정수부 : 0 (16 진수로 0x00)

감속값 0.1 의 소수부 : 1 (16 진수로 0x01)

<b>Data</b> ①	<b>Data</b> ②	<b>Data</b> ③	<b>Data</b> ④	<b>Data</b> ⑤	<b>Data</b> ⑥	<b>Data</b> ⑦	<b>Data</b> ⑧
0xc3	0x01	0x00	0x2e	0x01	0x00	0x2e	0x04

### 23. 속도/위치 제어기의 P, I, D, IL값 설정하기

- 게인값의 설정 범위 : 000.000 ~ 999.999 (0x0000.0x0000 ~ 0x03e7.0x0327)

- CAN 데이터 구성 :

<b>Data</b> ①	<b>Data</b> ②	<b>Data</b> ③	<b>Data</b> ④	<b>Data</b> ⑤	<b>Data</b> ⑥	<b>Data</b> ⑦	<b>Data</b> ⑧
명령코드	대상 모터	게인값 정수 부 상위 8 비트	게인값 정수 부 하위 8 비트	0x2e	게인값 소 수부 상위 8 비 트	게인값 소 수부 하위 8 비 트	0

**Data** ① 명령 코드 :

0xc5 (속도제어기 P 게인 설정 명령)

0xc6 (속도제어기 I 게인 설정 명령)

0xc7 (속도제어기 D 게인 설정 명령)

0xc8 (속도제어기 IL 게인 설정 명령)

0xc9 (위치제어기 P 게인 설정 명령)

0xca (위치제어기 I 게인 설정 명령)

0xcb (위치제어기 D 게인 설정 명령)

0xcc (위치제어기 IL 게인 설정 명령)

**Data** ② 대상 모터 :

0x01(오른쪽 모터), 0x02(왼쪽 모터)

**Data** ③ 게인값 정수부의 상위 8 비트 데이터

**Data** ④ 게인값 정수부의 하위 8 비트 데이터

**Data** ⑤ 0x2e : 정수부와 소수부 구분점으로 데이터 자체가 가지고있는 의미는 없음

**Data ⑥** 게인값 소수부의 상위 8 비트 데이터

**Data ⑦** 게인값 소수부의 하위 8 비트 데이터

**Data ⑧** 0x00 : 의미없음

예 1-1) 오른쪽 모터 속도제어기 P 게인 '030.000'으로 설정하기

- 속도제어기의 P 게인 설정이므로 명령코드 **Data ① = 0xc5**

- 오른쪽 모터 설정이므로 **Data ② = 0x01**

- 게인값 '030.000' 에서 정수부 '030'은 16 진수로 '0x001e'

(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x1e)

소수부 '000'은 16 진수로 0x0000

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x00)

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0xc5</b>	<b>0x01</b>	<b>0x00</b>	<b>0x1e</b>	<b>0x2e</b>	<b>0x00</b>	<b>0x00</b>	<b>0</b>

예 1-2) 오른쪽 모터 속도제어기 I 게인 '200.000'으로 설정하기

- 속도제어기의 I 게인 설정이므로 명령코드 **Data ① = 0xc6**

- 오른쪽 모터 설정이므로 **Data ② = 0x01**

- 게인값 '200.000' 에서 정수부 '200'은 16 진수로 '0x00c8'

(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0xc8)

소수부 '000'은 16 진수로 0x0000

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x00)

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0xc6</b>	<b>0x01</b>	<b>0x00</b>	<b>0xc8</b>	<b>0x2e</b>	<b>0x00</b>	<b>0x00</b>	<b>0</b>

예 1-3) 오른쪽 모터 속도제어기 D 게인 '000.005'으로 설정하기

- 속도제어기의 D 게인 설정이므로 명령코드 **Data ① = 0xc7**

- 오른쪽 모터 설정이므로 **Data ② = 0x01**

- 게인값 '000.005' 에서 정수부 '000'은 16 진수로 '0x0000'

(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x00)

소수부 '005'은 16 진수로 0x0005

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x05)

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0xc7</b>	<b>0x01</b>	<b>0x00</b>	<b>0x00</b>	<b>0x2e</b>	<b>0x00</b>	<b>0x05</b>	<b>0</b>

예 1-4) 오른쪽 모터 속도제어기 IL 게인 '020.000' 으로 설정하기

- 속도제어기의 IL 게인 설정이므로 명령코드 **Data ① = 0xc8**

- 오른쪽 모터 설정이므로 **Data ② = 0x01**

- 게인값 '020.000' 에서 정수부 '020'은 16 진수로 '0x0014'

(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x14)

소수부 '000'은 16 진수로 0x0000

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x00)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc8	0x01	0x00	0x14	0x2e	0x00	0x00	0

예 2-1) 왼쪽 모터 속도제어기 P 게인 '030.000'으로 설정하기

- 속도제어기의 P 게인 설정이므로 명령코드 Data ① = 0xc5

- 왼쪽 모터 설정이므로 Data ② = 0x02

- 게인값 '030.000' 에서 정수부 '030'은 16 진수로 '0x001e'

(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x1e)

소수부 '000'은 16 진수로 0x0000

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x00)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc5	0x02	0x00	0x1e	0x2e	0x00	0x00	0

예 2-2) 왼쪽 모터 속도제어기 I 게인 '200.000'으로 설정하기

- 속도제어기의 I 게인 설정이므로 명령코드 Data ① = 0xc6

- 왼쪽 모터 설정이므로 Data ② = 0x02

- 게인값 '200.000' 에서 정수부 '200'은 16 진수로 '0x00c8'

(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0xc8)

소수부 '000'은 16 진수로 0x0000

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x00)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc6	0x02	0x00	0xc8	0x2e	0x00	0x00	0

예 2-3) 왼쪽 모터 속도제어기 D 게인 '000.0050'으로 설정하기

- 속도제어기의 D 게인 설정이므로 명령코드 Data ① = 0xc7

- 왼쪽 모터 설정이므로 Data ② = 0x02

- 게인값 '000.005' 에서 정수부 '000'은 16 진수로 '0x0000'

(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x00)

소수부 '005'은 16 진수로 0x0005

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x05)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc7	0x02	0x00	0x00	0x2e	0x00	0x05	0

예 2-4) 왼쪽 모터 속도제어기 IL 게인 '020.000' 으로 설정하기

- 속도제어기의 IL 게인 설정이므로 명령코드 Data ① = 0xc8
- 왼쪽 모터 설정이므로 Data ② = 0x02
- 게인값 '020.000' 에서 정수부 '020'은 16 진수로 '0x0014'  
(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x14)
- 소수부 '000'은 16 진수로 0x0000

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x00)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc8	0x02	0x00	0x14	0x2e	0x00	0x00	0

예 3-1) 오른쪽 모터 위치제어기 P 게인 '004.473' 으로 설정하기

- 위치제어기의 P 게인 설정이므로 명령코드 Data ① = 0xc9
- 오른쪽 모터 설정이므로 Data ② = 0x01
- 게인값 '004.473' 에서 정수부 '004'은 16 진수로 '0x0004'  
(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x04)
- 소수부 '473'은 16 진수로 0x01d9

(소수부 상위 8 비트 = 0x01, 소수부 하위 8 비트 = 0xd9)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc9	0x01	0x00	0x04	0x2e	0x01	0xd9	0

예 3-2) 오른쪽 모터 위치제어기 I 게인 '000.015' 으로 설정하기

- 위치제어기의 I 게인 설정이므로 명령코드 Data ① = 0xca
- 오른쪽 모터 설정이므로 Data ② = 0x01
- 게인값 '000.015' 에서 정수부 '000'은 16 진수로 '0x0000'  
(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x00)
- 소수부 '015'은 16 진수로 0x000f

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x0f)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xca	0x01	0x00	0x00	0x2e	0x00	0x0f	0

예 3-3) 오른쪽 모터 위치제어기 D 게인 '007.040' 으로 설정하기

- 위치제어기의 D 게인 설정이므로 명령코드 Data ① = 0xcb
- 오른쪽 모터 설정이므로 Data ② = 0x01
- 게인값 '007.040' 에서 정수부 '007'은 16 진수로 '0x0007'  
(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x07)
- 소수부 '040'은 16 진수로 0x0028

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x28)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xcb	0x01	0x00	0x07	0x2e	0x00	0x28	0

예 3-4) 오른쪽 모터 위치제어기 IL 계인 '100.000' 으로 설정하기

- 위치제어기의 IL 계인 설정이므로 명령코드 Data ① = 0xcc
- 오른쪽 모터 설정이므로 Data ② = 0x01
- 계인값 '100.000' 에서 정수부 '100'은 16 진수로 '0x0064'  
(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x64)
- 소수부 '000'은 16 진수로 0x0000

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x00)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xcc	0x01	0x00	0x64	0x2e	0x00	0x00	0

예 4-1) 왼쪽 모터 위치제어기 P 계인 '004.473' 으로 설정하기

- 위치제어기의 P 계인 설정이므로 명령코드 Data ① = 0xc9
- 왼쪽 모터 설정이므로 Data ② = 0x02
- 계인값 '004.473' 에서 정수부 '004'은 16 진수로 '0x0004'  
(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x04)
- 소수부 '473'은 16 진수로 0x01d9

(소수부 상위 8 비트 = 0x01, 소수부 하위 8 비트 = 0xd9)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xc9	0x02	0x00	0x04	0x2e	0x01	0xd9	0

예 4-2) 왼쪽 모터 위치제어기 I 계인 '000.015' 으로 설정하기

- 위치제어기의 I 계인 설정이므로 명령코드 Data ① = 0xca
- 왼쪽 모터 설정이므로 Data ② = 0x02
- 계인값 '000.015' 에서 정수부 '000'은 16 진수로 '0x0000'  
(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x00)
- 소수부 '015'은 16 진수로 0x000f

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x0f)

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xca	0x02	0x00	0x00	0x2e	0x00	0x0f	0

예 4-3) 왼쪽 모터 위치제어기 D 계인 '007.040' 으로 설정하기

- 위치제어기의 D 계인 설정이므로 명령코드 Data ① = 0xcb
- 왼쪽 모터 설정이므로 Data ② = 0x02
- 계인값 '007.040' 에서 정수부 '007'은 16 진수로 '0x0007'  
(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x07)
- 소수부 '040'은 16 진수로 0x0028
- (소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x28)



<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0xcb</b>	<b>0x02</b>	<b>0x00</b>	<b>0x07</b>	<b>0x2e</b>	<b>0x00</b>	<b>0x28</b>	<b>0</b>

예 4-4) 왼쪽 모터 위치제어기 IL 계인 '100.000' 으로 설정하기

- 위치제어기의 IL 계인 설정이므로 명령코드 Data ① = 0xcc
- 왼쪽 모터 설정이므로 Data ② = 0x02
- 계인값 '100.000' 에서 정수부 '100'은 16 진수로 '0x0064'  
(정수부 상위 8 비트 = 0x00, 정수부 하위 8 비트 = 0x64)
- 소수부 '000'은 16 진수로 0x0000

(소수부 상위 8 비트 = 0x00, 소수부 하위 8 비트 = 0x00)

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0xcc</b>	<b>0x02</b>	<b>0x00</b>	<b>0x64</b>	<b>0x2e</b>	<b>0x00</b>	<b>0x00</b>	<b>0</b>

#### 24. 휠 반지름 값 설정하기

- STELLA B2 의 휠 반지름 값을 설정합니다.
- 이 명령어는 STELLA 의 **휠 을 변경했을 경우에만** 사용하시기 바랍니다.
- 단위는 1/100 [cm]
- CAN 데이터 구성 :

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
명령코드 <b>(0xcd)</b>	설정할 반 지름 값 상위 8 비트	설정할 반지 름 값 하위 8 비트	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Data ①** 명령 코드 : 0xcd (휠 반지름 설정)

**Data ②** : 설정할 반지름 값의 상위 8 비트

**Data ③** : 설정할 반지름 값의 하위 8 비트

**Data ④ ~ ⑧** : 의미없음

예 ) 휠 반지름값을 0.0752m(7.52cm)로 설정하기

$$7.52 \text{ [cm]} = 752 \text{ [1/100 cm]}$$

$$752 = 0x02f0$$

상위 8 비트 데이터 : 0x02

하위 8 비트 데이터 : 0xf0

<b>Data ①</b>	<b>Data ②</b>	<b>Data ③</b>	<b>Data ④</b>	<b>Data ⑤</b>	<b>Data ⑥</b>	<b>Data ⑦</b>	<b>Data ⑧</b>
<b>0xcd</b>	<b>0x02</b>	<b>0xf0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

25. 두 휠 사이의 너비 설정하기

- STELLA B2 의 두 휠 사이의 너비 값을 설정합니다.
- 이 명령어는 STELLA 의 **휠 을 변경했을 경우에만** 사용하시기 바랍니다.
- 단위는 1/100 [cm]
- CAN 데이터 구성 :

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
명령코드 (0xce)	설정할 너 비 값 상위 8 비트	설정할 너비 값 하위 8 비트	0	0	0	0	0

- Data ①** 명령 코드 : 0xce (두 휠 사이의 너비 설정)
- Data ②** : 설정할 두 휠 사이의 너비 값의 상위 8 비트
- Data ③** : 설정할 두 휠 사이의 너비 값의 상위 8 비트
- Data ④ ~ ⑧** : 의미없음

예 ) 휠 반지름 값을 0.29m(29.0cm)로 설정하기

$$29.0 \text{ [cm]} = 2900 \text{ [1/100 cm]}$$

$$2900 = 0x0b54$$

상위 8 비트 데이터 : 0x0b

하위 8 비트 데이터 : 0x54

Data ①	Data ②	Data ③	Data ④	Data ⑤	Data ⑥	Data ⑦	Data ⑧
0xce	0x0b	0x54	0	0	0	0	0

## 6. 제품의 고장 진단

	확인 사항	조치 사항
전원이 들어 오지 않을 경우	전원선의 결선이나 또는 잘못된 배선 등을 확인하여 주십시오.	결선을 다시 하거나 잘못된 배선을 다시 배선 하여 주십시오.
바퀴가 회전하지 않을 경우	배터리의 방전 여부를 확인하여 주십시오. 입력 신호의 결선이나 또는 잘못된 배선 등을 확인하여 주십시오. 배터리의 방전 여부를 확인하여 주십시오. 보호 기능에 의해 알람 기능이 작동 하였는지 확인 하여 주십시오.	배터리를 충전하거나 교환하여 주십시오. 결선을 다시 하거나 잘못된 배선을 다시 배선 하여 주십시오.. 배터리를 충전하거나 교환하여 주십시오.. 알람의 원인을 제거 한 뒤 전원을 다시 입력 하여 주십시오.
바퀴는 회전하나 이상한 동작을 할 경우	명령 커맨드가 올바른지 확인 하여 주십시오 모터와 모션드라이브에 연결되어있는 엔코더 케이블이 빠져있는지 확인하여 주십시오 바퀴가 움직이다가 저전압 알람이 발생.	명령 커맨드를 올바르게 입력 하십시오 케이블을 올바르게 꽂아주십시오. 배터리의 방전을 부족으로 인한 전압 강하로 인해 경고가 발생 할 수 있습니다. 배터리를 교체하여 주십시오.

## 7. 제품의 보증

- 저희 ㈜엔티렉스 STELLA B2의 보증 기간은 구입일로부터 6개월입니다.
- 보증 기간 내에 정상 사용 중 고장 발생시 무상 수리를 받으실 수 있습니다.
- 보증 기간 경과 후 정상 사용 중 고장 발생시 유상 수리를 받으실 수 있습니다.  
- 수리 비용 별도 발생 -
- 사용자의 고의/부주의에 의한 고장 발생시 수리가 가능한 경우(보증기간 관계 없이) 유상 수리를 받으실 수 있습니다.  
- 수리 비용 별도 발생 -
- 기타 궁금한 내용은 아래 ㈜엔티렉스 로봇 연구소로 문의 바랍니다.



인천 광역시 남구 주안동 5-38

(주)엔티렉스 로봇 연구소

web site : <http://ntrexlabs.co.kr>

E-mail : [lab@ntrex.co.kr](mailto:lab@ntrex.co.kr)

## MEMO





인천 광역시 남구 주안동 5-38

(주)엔티렉스 로봇 연구소

web site : <http://ntrexlabs.co.kr>

E-mail : [lab@ntrex.co.kr](mailto:lab@ntrex.co.kr)

