HOVIS Lite

DRC 레지스터 & 프로토콜



Dongbu Robot

Dongbu Robot

Hovis Lite DRC Register & Protocol Manual

초판 발행일 : 2011년 11월 18일
펴낸곳 : ㈜동부로봇
주소 : 경기도 부천시 원미구 약대동 부천테크노파크 11층
내용문의 : 032-329-5551
FAX : 032-329-5569

본 교재는 ㈜동부로봇에서 저작한 고유 제작물로서 사전 승인없이 무단 복사, 복제 및 전제를 금합니다.

㈜동부로봇 All rights reserved

목차

PART 01 Register

비휘발성 Register Map ---------- 05

휘발성 Register Map ---------- 07

Register 상세 설명 ---------- 09

PART 02 Protocol

프로토콜 형식 ---------- 22

Command Set ---------- 24

명령 상세 설명 ---------- 28

PART 03 부록

ADC Lookup Table ---------- 52

PART 01

Register

비휘발성 Register Map

제어기의 레지스터는 Non-Volatile(비휘발성) Register와 Volatile(휘발성) Register로 구분되며, 제어기의 현재 상태와 구동에 대한 Data로 구성되어 있습니다. 사용자는 프로토콜의 명령을 통해서 Register의 Data를 읽거나 변경함으로써 제어기를 제어할 수 있고, DR-Visual Logic 프로그램을 통하여 로봇의 행동을 프로그램할 때도 사용할 수 있습니다.

Non-Volatile Register (EEP Register, 비휘발성) Map

비휘발성 레지스터는 전원이 꺼진 상태에서도 데이터가 지워지지 않는 영역으로, 제어기 동작의 기본적인 설정 값들이 저장됩니다. 비휘발성 레지스터의 값은 전원이 켜진 순간 휘발성 레지스터의 해당 영역으로 복사됩니다. 비휘발성 레지스터의 값을 변경하면 그 당시에는 동작에 직접적인 영향을 주지 않지만, 제어기를 Reboot 하거나 다시 전원을 켤 때에는 변경된 값이 휘발성 레지스터로 복사되면서 동작에 영향을 미치게 됩니다.

- **Address**

  Address는 Register의 주소로서, Register를 읽거나 쓰기 위해서는 Packet에 해당 레지스터의 주소를 지정해주어야 합니다.

- **Default**

  공장에서 최초 출하 시의 설정 값입니다. Rollback 명령을 사용하면 비휘발성 레지스터를 Default 값으로 되돌릴 수 있습니다.

- **Valid Range**

  레지스터가 가질 수 있는 데이터의 유효 범위입니다. 유효 범위를 초과하여 입력된 경우에는 휘발성 메모리에 복사되는 과정에서 에러가 발생하며, 이 때 휘발성 메모리의 데이터는 유효 범위 내의 값으로 변경됩니다.

- **RW**

  RO(Read Only)는 읽기만 가능하고 쓰기는 할 수 없는 Register를 의미합니다. RO Register에 Write할 경우 에러가 발생합니다. RO Register는 제어기의 모델 명이나 Firmware 버전 같은 고유한 값이거나, 제어기의 센서 정보가 들어있는 항목입니다. RW는 읽기와 쓰기가 모두 가능한 Register입니다.

  ※ e(Reg_Name) : 비휘발성 Register(EEP Register)의 Reg_Name 항목을 지칭합니다.
  ※ r(Reg_Name) : 휘발성 Register(RAM Register)의 Reg_Name 항목을 지칭합니다.

PART 01

Register

HOVIS

비휘발성 Register Map

Addr	구분	Bytes	Default	Valid Range	RW	Comments
0	Model No1	1	0x05	–	RO	제어기 모델 표시
1	Model No2	1	0x54	–	RO	
2	Version1	1	0x01	–	RO	Firmware version
3	Version2	1	0x22	–	RO	
4	Baud Rate	1	0x10	8page참고	RW	PC–제어기, 제어기–서보 간 통신 속도
5	Special Function	1	0x00	0x00~0xF3	RW	특수용도로 DRC를 사용하는 Flag
6	Reserved	1	0x00	–	–	
7	ID	1	0xFD	0x00~0xFD	RW	제어기 ID(0xFE: Broadcasting ID로 사용 가능, ID 할당은 불가)
8	Ack Policy	1	0x01	0x00~0x02	RW	Policy에 의해 패킷에 응답
9	Torque Off Policy	1	0x2B	0x00~0x7F	RW	Policy에 의해 서보의 Torque를 품
10	Alarm LED Policy	1	0x7F	0x00~0x7F	RW	Policy에 의해 경보 LED 점멸
11	Status Check Policy	1	0x01	0x00~0x01	RW	서보의 각도 값을 체크할 지 여부
12	Min. Voltage	1	0x5F	0x00~0xFE	RW	최소 허용 전압(0x5F : 7.1V)
13	Max. Voltage	1	0x88	0x00~0xFE	RW	최대 허용 전압(0x88 : 10.0V)
14	Max.Temperature	1	0xDF	0x00~0xFE	RW	최대 허용 온도(0xDF : 85 ° C)
15	Remocon Channel	1	0x61	0x61~0x6A	RW	적외선 리모콘의 채널 코드
16	Servo Ack Wait Tick	1	0x04	0x00~0xFE	RW	Servo Ack를 기다릴 최소시간 (0x04 : 6.4ms)
17	Zigbee Ack Wait Tick	1	0x50	0x00~0xFE	RW	Zigbee Ack를 기다릴 시간 (0x50 : 128ms)
18	LED Blink Period	1	0xBB	0x00~0xFE	RW	경보 LED 점멸 주기 (0xBB : 약 300ms)
19	ADC Fault Check Period	2	0x0138	0x0000 ~ 0x7FFF	RW	온도/전압 Error 판정 주기 (0x0138 : 약 500ms)
21	Packet Garbage Check Period	2	0x007D	0x0000 ~0x7FFF	RW	오류 패킷 판정 주기 (0x7D : 약 200ms)

0번부터 6번 주소까지는 제어기의 기본적인 정보와 통신에 관련된 Data이며, 7번부터 22번 주소까지는 제어기의 기능과 관련된 설정 Data입니다. 7번부터 22번 주소까지의 Data는 제어기를 부팅할 때 휘발성 레지스터에 복사됩니다.

휘발성 Register Map

Volatile Register(RAM Register MAP)

휘발성 레지스터는 제어기 동작 설정이나 제어기의 상태, 센서 값을 저장하고 있으며, 휘발성 레지스터에 저장된 동작 설정이 제어기 동작에 직접적인 영향을 미칩니다., 제어기가 꺼졌다 켜지거나 Reboot 되면 초기화 됩니다. 휘발성 레지스터를 변경하여 제어기의 동작 설정을 변경했더라도 제어기가 Reboot되면 비휘발성 레지스터의 값으로 초기화됩니다.

Addr	구분	Bytes	Valid Range	RW	Comments
0	ID	1	0x00~0xFD	RW	
1	Ack Policy	1	0x00~0x02	RW	
2	Torque Off Policy	1	0x00~0x7F	RW	
3	Alarm LED Policy	1	0x00~0x7F	RW	
4	Status Check Policy	1	0x00~0x01	RW	
5	Min. Voltage	1	0x00~0xFE	RW	
6	Max. Voltage	1	0x00~0xFE	RW	비휘발성 레지스터 영역에서 복사되어 비휘발성 레지스터 영역과 동일하게 초기화 됨.
7	Max. Temperature	1	0x00~0xFE	RW	
8	Remocon Channel	1	0x61~0x6A	RW	
9	Servo Ack Wait Tick	1	0x00~0xFE	RW	
10	Zigbee Ack Wait Tick	1	0x00~0xFE	RW	
11	LED Blink Period	1	0x00~0xFE	RW	
12	ADC Fault Check Period	2	0x0000~0x7FFF	RW	
14	Packet Garbage Check Period	2	0x0000~0x7FFF	RW	
16	Status Error	1	0x00~0x7F	RW	상태 에러, 11 페이지 참조
17	Error Codes [0] ~ [4]	5	50 페이지 참조	RW	상태 에러 코드 중 최근 5개
22	LED Control	1	0x00~0x07	RW	Task 실행 시 LED 값 (0x01:Red, 0x02:Green, 0x04: Blue)
23	User Timer Tick	1	0x00~0xFF	RW	사용자 설정 가능한 타이머(100[ms]/tick)
24	Connected Program	1	0~3	RW	현재 연결된 PC 프로그램
25	Zigbee Channel	1	11~26	RW	Zigbee 의 통신채널
26	Zigbee PANID	2	0x0000~0xFFFF	RW	Zigbee가 속한 WPAN ID
28	Zigbee SADDR	2	0x0000~0xFFFF	RW	Zigbee의 ID
30	Zigbee DSTADDR	2	0x0000~0xFFFF	RW	상대 Zigbee의 ID
32	Zigbee ACKREQ	1	0~2	RW	RF 통신시 Ack를 요구할 것인지 여부
33	Zigbee BACKOFF	1	0~2	RW	RF 통신시 Random Delay 적용 여부
34	Servo Count	1	0~32	RO	현재 연결된 모터의 수
35	Servo ID[0]~[32]	33	0x00~0xFE	RO	각 모터의 ID (0xFE는 모터가 없다는 것을 의미)

Addr	구분	Bytes	Valid Range	RW	Comments
68	Playing Motion	1	0x00~0x01	RO	Motion이 실행 중인지 여부
69	Playing Task	1	0, 1, 3	RO	Task가 실행 중인지 여부
70	Charger Connected	1	0~1	RO	충전기가 연결 되었는지 여부
71	Buzzer Scale	1	0x00~0x25	RO	버저 음의 음높이
72	Buzzer Time	1	0~192	RO	버저 음의 남은 시간(6.4[ms]/tick)
73	Button Status	1	0x00~0x3F	RO	버튼의 상태
74	Remocon Length	1	0~240	RO	리모콘 버튼이 눌린 시간(125[ms]/tick)
75	Remocon Data	1	0x00~0x1D,0xFE	RO	리모콘 버튼 번호
76	Input Voltage Value	1	0x00~0xFE	RO	입력 전압의 Raw Data, 8bit
77	Temperature Value	1	0x00~0xFE	RO	현재 온도의 Raw Data, 8bit
78	Light Sensor Value	1	0x00~0xFE	RO	빛 센서 값의 Raw Data, 8bit
79	ADC Port 1 Sensor Type	1	0~3	RO	ADC 포트1에 연결된 센서 종류
80	ADC Port 2 Sensor Type	1	0~3	RO	ADC 포트2에 연결된 센서 종류
81	ADC Port 1 Sensor Value	2	0x0000~0xFFFF	RO	ADC 포트1에 연결된 센서 출력 값
83	ADC Port 2 Sensor Value	2	0x0000~0xFFFF	RO	ADC 포트2에 연결된 센서 출력 값
85	ACC/GYRO Connected	1	0~1	RO	가속도/자이로 센서 연결 여부
86	ACC X Value	2	-4096~4095	RO	가속도 센서 X축 Raw Data, 13bit
88	ACC Y Value	2	-4096~4095	RO	가속도 센서 Y축 Raw Data, 13bit
90	ACC Z Value	2	-4096~4095	RO	가속도 센서 Z축 Raw Data, 13bit
92	GYRO X Value	2	-32768~32767	RO	자이로 센서 X축 Raw Data, 16bit
94	GYRO Y Value	2	-32768~32767	RO	자이로 센서 Y축 Raw Data, 16bit
96	GYRO Z Value	2	-32768~32767	RO	자이로 센서 Z축 Raw Data, 16bit
98	Sound Detection Flag	1	0~250	RO	소리감지횟수
99	Sound Direction	1	-2~2	RO	소리를 감지한 방향(-는 왼쪽, +는 오른쪽)
100	Touch Status	1	0~1	RO	연결된 버저 모듈의 터치 상태값
101	Tick	2	0~60000	RO	System tick, 1.6[ms]/INT
103	DRT-HWW1 Connected	1	0~1	RO	DRT-HWW1이 연결되었는지 여부
104	DRC-004TO Connected	1	0~1	RO	DRC-004TO가 연결되었는지 여부
105	Reserved	1	-	-	예비용
106	Servo Status Error & Detail [0]~[31]	64	0x00~0x80 * 64	RO	연결된 서보의 상태값
170	Servo Position[0]~[31]	64	0x0000~0x7FFF	RO	연결된 서보의 Position
234	DRT-HWWI Status Error	1	0x00~0x80	RO	연결된 DRT-HWW1의 상태에러
235	DRT-HWWI Status Detail	1	0x00~0x7F	RO	연결된 DRT-HWW1의 상세상태
236	DRT-004TO Status Error	1	0x00~0x80	RO	연결된 DRT-004TO의 상태에러
237	DRT-004TO Status Detail	1	0x00~0x7F	RO	연결된 DRT-004TO의 상세상태

Register 상세 설명

Model No 1, Model No 2(EEP Register의 0, 1번 Address)

DRC의 모델명을 2바이트의 Binary로 표시한 것입니다. 사용자가 바꿀 수 없습니다.

Version 1, Version 2(EEP Register의 2, 3번 Address)

DRC 내부에 들어가 있는 현재 펌웨어의 버전입니다. 펌웨어 버전이 최신이 아니라면 홈페이지에서 다운로드 받아서 업데이트 하여야 합니다. 사용자가 바꿀 수 없습니다.

Baud Rate(EEP Register의 4번 Address)

PC와 DRC 사이, 그리고 DRC와 DRS 사이의 UART 통신 속도를 설정하는 데이터입니다. 값에 따른 통신 속도는 아래 표와 같습니다. 기본 설정 속도는 115,200 bps이며, 표에 없는 값이 레지스터에 들어 있으면 기본 설정 속도인 115,200 bps로 설정됩니다.

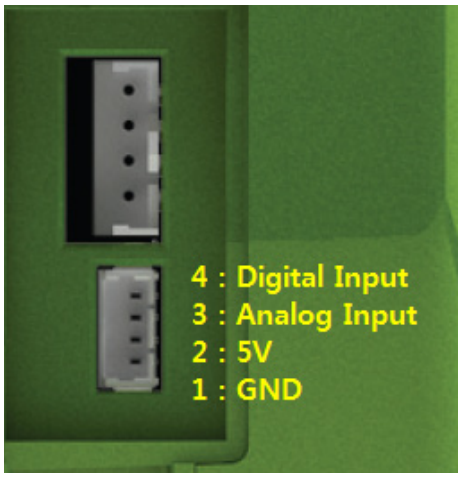
Baud Rate	Register Value
57,600	34
115,200	16
200,000	9
250,000	7
400,000	4
500,000	3
666,667	2

Special Function(EEP Register의 5번 Address)

DRC-005T를 보통과 다른 특수 용도로 쓰고 싶을 때 사용하는 레지스터입니다. 각 비트별로 1, 0의 값을 넣어서 사용할지 여부를 결정할 수 있으며, 초기값은 0x00(모두 사용 안함)입니다. 비트에 따른 용도는 아래와 같습니다.

비트	값	모드
0	0x01	사용자 센서 모드
1	0x02	TTL 통신 모드
2	0x04	Reserved
3	0x08	Reserved
4	0x10	Reserved
5	0x20	Reserved
6	0x40	Reserved
7	0x80	Reserved

*** 사용자 센서 모드 :** 사용자가 임의의 센서를 달아서 사용할 수 있는 모드입니다. DRC-005T에는 양 옆에 센서가 연결되는 4핀 포트가 있는데, 이 포트는 평상시에는 제한된 종류의 센서만을 사용할 수 있습니다. 사용자 센서 모드를 사용하면 이 포트에 5V를 전원으로 하는 아날로그 센서 하나와 디지털 센서 하나를 달 수 있습니다. 포트는 양 옆에 총 2개가 있으므로, 총 아날로그 센서 2개와 디지털 센서 2개를 달 수 있는 것입니다. 아날로그 센서의 값은 각각 r(ADC Port 1 Sensor Value), r(ADC Port 2 Sensor Value)에 10비트(0~1023)로 나타나며, 디지털 센서의 값은 각각 r(ADC Port 1 Sensor Type), r(ADC Port 2 Sensor Type)에 0~1로 나타납니다.
센서 포트의 핀맵은 그림과 같습니다.



*** TTL 통신 모드 :** 현재 PC와 DRC-005T의 통신 레벨은 RS232C로, ±5~10V의 레벨을 사용합니다. 하지만 RS232C 레벨이 아닌 3.3V의 TTL 레벨로 DRC-005T를 PC처럼 제어하고 싶은 경우도 있을 것입니다. 이런 경우에는 DRC-005T를 TTL 통신 모드로 설정하고 Zigbee가 장착되는 핀을 사용하면 DRC-005T과 TTL 레벨의 신호로 통신할 수 있습니다. 사용하실 핀은 아래와 같습니다.

ID(EEP Register의 7번 Address, RAM Register의 0번 Address)

DRC의 ID입니다. 253(0xFD)가 기본 설정이며, 여러 대의 DRC를 서로 ID를 다르게 하면 DRS 여러 개를 제어하듯 같은 통신 라인에 여러 대의 DRC를 연결하여 제어할 수 있습니다. 같은 통신 라인에 서로 ID가 같은 두 대 이상의 DRC가 연결되어 있다면, 통신 시 오동작할 가능성이 큽니다.

ACK Policy(EEP Register의 8번 Address, RAM Register의 1번 Address)

PC에서 DRC로 요청 패킷(Request Packet)을 수신 시, 응답 패킷(ACK Packet)을 보낼지 여부에 대한 설정 Data입니다.

- 0 : 어떠한 요청 패킷에도 응답하지 않습니다.
- 1 : Read Command와 몇몇 응답을 요구하는 패킷에 대해서만 응답합니다.
- 2 : 모든 패킷에 대하여 응답합니다.

※ STAT 요청 패킷에 대해서는 ACK Policy에 무관하게 항상 ACK Packet을 보냅니다.

※ REMOCON 요청 패킷에 대해서는 ACK Policy에 무관하게 항상 응답하지 않습니다.

※ 패킷의 pID가 254(0xFE, Broadcast pID)인 경우 응답하지 않습니다. 단, STAT 명령에는 응답합니다.

※ 각 요청 패킷의 ACK Policy에 따른 응답 여부는 22page의 명령 상세 설명에서 더 자세히 다룹니다.

Torque Off Policy(EEP Register의 9번 Address, RAM Register의 2번 Address)

상태 에러가 발생 시 제어기에 연결된 서보 모터의 토크를 풀 것인지 여부에 대한 정책을 결정합니다.

- (r(Torque Off Policy) & r(Status Error))이 참일 경우 연결된 모든 서보 모터의 토크가 풀려, 서보 모터가 더 이상 움직이지 않게 됩니다.
- r(Status Error)의 에러 상태를 해제해야 다시 서보 모터를 Torque On 상태로 변경할 수 있습니다.

※ &는 Bitwise AND 연산자로, A & B 연산을 수행할 시 A와 B의 각 비트를 비교하여 A, B 모두에서 1인 비트만 결과가 1이 됩니다. 예시) 00101110 & 10110110 의 연산 결과는 00100110이 됩니다.

Alarm LED Policy(EEP Register의 10번 Address, RAM Register의 3번 Address)

상태 에러가 발생 시 제어기의 경보 LED를 점멸할 것인지 여부에 대한 정책을 결정합니다.

- (r(Alarm LED Policy) & r(Status Error))이 참일 경우 제어기의 TX, RX, Spare LED가 점멸하며, 경보 LED의 점멸 주기는 r(LED Blink Period)에 의해 결정됩니다.
- 경보 LED가 점멸 중일 때는 TX, RX, Spare LED의 본래 기능은 무시 됩니다.
- TX, RX, Spare LED가 원래대로 정상 동작하기 위해서는 r(Status Error)의 에러 상태를 해제해야 합니다.

Status Check Policy(EEP Register의 11번 Address, RAM Register의 4번 Address)

제어기가 지속적으로 자신에 연결된 서보의 현재 위치 값을 읽어올지 여부를 설정합니다. r(Status Check Policy)가 1로 설정되어 있으면, 제어기는 자신과 연결된 서보의 위치 값과 서보의 상태를 지속적으로 r(Servo Status Error & Status Detail[0]~[31])과 r(Servo Position[0]~[31])에 업데이트 합니다. r(Status Check Policy)가 0로 설정되어 있으면, 이런 업데이트를 수행하지 않습니다.

Minimum Voltage(EEP Register의 12번 Address, RAM Register의 5번 Address)

최소 입력 전압의 Raw Data를 의미합니다. DRC의 입력 전압 r(Input Voltage Value)가 r(Minimum Voltage) 미만일 경우 r(Status Error)에 “Exceed Input Voltage Limit”(0번 비트)이 설정되며, r(Error Codes[0]~[4])에 0x01(전압이 너무 낮음)이 추가 됩니다.

- Default 설정치는 0x5F(약 7.1V)입니다. 실제 전압 값과의 관계는 환산표(page 48)를 참조 바랍니다.

Maximum Voltage(EEP Register의 13번 Address, RAM Register의 6번 Address)

최대 입력 전압의 Raw Data를 의미합니다. DRC의 입력 전압 r(Input Voltage Value)가 r(Maximum Voltage)를 초과할 경우 r(Status Error)에 “Exceed Input Voltage Limit”(0번 비트)이 설정되며, r(Error Codes[0]~[4])에 0x02(전압이 너무 높음)이 추가 됩니다.

- Default 설정치는 0x88(약 10.0V)입니다. 실제 전압 값과의 관계는 환산표(page 48)를 참조 바랍니다.

Maximum Temperature(EEP Register의 14번 Address, RAM Register의 7번 Address)

최대 동작 온도의 Raw Data를 의미합니다. DRC의 온도 r(Temperature Value)가 r(Maximum Temperature)를 초과할 경우 r(Status Error)에 "Exceed Temperature Limit"(1번 비트)이 설정되며, r(Error Codes[0]~[4])에 0x03(온도가 너무 높음)이 추가 됩니다.

- Default 설정치는 0xDF(약 85℃)입니다. 실제 온도 값과의 관계는 환산표(page 49)를 참조 바랍니다.

Remocon Channel(EEP Register의 15번 Address, RAM Register의 8번 Address)

리모컨의 채널을 의미합니다. 리모컨의 채널 값은 0x61부터 0x6A까지 10개의 채널 설정이 가능하며, 현재 사용하는 리모컨의 채널과 r(Remocon Channel)이 서로 일치해야 리모컨을 인식할 수 있습니다.

Servo Ack Wait Tick(EEP Register의 16번 Address, RAM Register의 9번 Address)

DRC에 연결된 서보에 요청해 컷을 보낸후에는, 응답패킷을 받기 까지 약간의 시간을 기다려야 합니다. DRC는 돌아올 응답패킷의 크기를 예측해서 그만큼에 비례하는 시간을 기다리는데, 이 시간이 지나가도록 응답패킷이 모두 들어오지 않으면 응답이 오지 않은 것으로 판단합니다. Servo Ack wait Tick 은 가장 짧은 응답패킷 (9 byte)이 예상될때 기다릴 시간을 의미하며 응답 패킷이 길어질수록 더 긴 시간을 기다립니다. 1tick은 1.6ms를 의미하며, Default 설정치는 0x50(약 128ms)입니다.

Zigbee Ack Wait Tick (EEP Register의 17번 Address, RAM Register의 10번 Address)

DRC에 연결된 Zigbee 모듈에 응답 패킷(ACK Packet)을 필요로 하는 요청 패킷을 보낸 후, 응답 패킷을 받기 까지 기다리는 최대 시간입니다. 최대 시간까지 응답 패킷이 돌아오지 않는다면, 응답이 오지 않은 것으로 판단합니다. 1tick은 1.6ms를 의미하며, Default 설정치는 0x50(약 128ms)입니다.

LED Blink Period(EEP Register의 18번 Address, RAM Register의 11번 Address)

상태 에러 발생 시 r(Alarm LED Policy)에 따라 LED가 점멸할 때, 경보 LED의 점멸 주기를 나타냅니다. LED는 r(LED Blink Period)만큼 켜져 있고, r(LED Blink Period)만큼 꺼져 있는 것을 계속 반복합니다. 1tick은 1.6ms를 의미하며, Default 설정치는 0xBB(약 300ms)입니다.

ADC Fault Check Period(EEP Register의 19번 Address, RAM Register의 12번 Address)

입력 전압과 온도의 오류 판정 주기입니다. 입력 전압과 온도가 허용 범위를 벗어나 r(ADC Fault Check Period)에 해당하는 시간만큼 지속될 경우 오류로 판정합니다. 1tick은 1.6ms를 의미하며, Default 설정치는 0x0138(약 500ms)입니다.

Packet Garbage Check Period(EEP Register의 21번 Address, RAM Register의 14번 Address)

미완성 패킷 판정 주기입니다. 패킷 형식에 어긋나는 데이터를 받았거나, 완성된 패킷을 미처 다 받지 못한 상태가 r(Packet Garbage Check Period)만큼 유지되면 미완성 패킷을 삭제하고 r(Status Error)의 "Invalid Packet"(2번 비트)이 설정합니다. r(Error Codes[0]~[4])에는 패킷을 어디서 받았느냐에 따라서 0x41(Zigbee 모듈의 응답 패킷이 미완성) 혹은 0x51(Servo의 응답 패킷이 미완성) 혹은 0x61(PC측 요청 패킷이 미완성)이 추가 됩니다.

Status Error(RAM Register의 16번 Address)

제어기 상태의 오류 내역을 나타냅니다. 총 7개의 비트로 각 오류의 상태 값을 나타냅니다. r(Alarm LED Policy)와 r(Torque Off Policy)도 아래의 오류 내역과 동일한 구조이며, r(Alarm LED Policy)에서 1로 설정된 비트에 해당하는 오류가 발생 시에는 경보 LED가 점멸하고, r(Torque Off Policy)에서 1로 설정된 비트에 해당하는 오류가 발생 시에는 모든 서보의 토크를 해제합니다.

비트	값	내역
0	0x01	Exceed Input Voltage limit
1	0x02	Exceed Temperature limit
2	0x04	Invalid Packet
3	0x08	Servo Missing
4	0x10	EEP REG distorted
5	0x20	Servo Status Error
6	0x40	Flash Data Distorted
7	0x80	Reserved

Error Codes[0]~[4](RAM Register의 17번 Address)

에러 발생 시 상세 에러 코드를 나타냅니다. 총 5바이트의 공간에 마지막으로 발생한 5가지의 에러가 저장됩니다. 에러가 발생하면 해당하는 에러 코드를 [0]에 저장하고, 기존 [0]~[3]에 저장되어 있던 에러 코드를 [1]~[4]로 한 바이트씩 밀어냅니다. 에러 코드의 상세 내역은 에러 코드 목록(page 50)을 참조 바랍니다.

LED Control(RAM Register의 22번 Address)

Task를 실행하고 있을 때 LED를 제어하는 레지스터입니다. 총 0x00~0x07까지의 값을 가질 수 있으며, 각 비트가 1일 때는 LED를 켜고, 0일 때는 LED를 끕니다. 각 비트가 제어하는 LED는 표와 같습니다. Task 실행 중이 아닐 때에는 LED Control은 의미를 가지지 않으며 항상 0입니다.

비트	값	LED
0	0x01	TX(Red)
1	0x02	RX(Green)
2	0x04	Spare(Blue)

User Timer Tick(RAM Register의 23번 Address)

사용자가 컨트롤할 수 있는 Timer로, 0이 아닌 값을 쓰면 100ms마다 1씩 줄어듭니다. Task 실행 시에 일정 시간만큼 Delay를 주는 용도로 사용합니다.

Connected Program(RAM Register의 24번 Address)

현재 어떤 프로그램으로 PC와 연결되어 통신하고 있는지 알려주는 레지스터 입니다.

- 0 : 프로그램과 연결되지 않았음
- 1 : HerkuleX Manager와 연결된 상태
- 2 : DR-SIM과 연결된 상태
- 3 : DR-Visual Logic과 연결된 상태

Zigbee channel (RAM Registor의 25번 Address)

Zigbee 모듈이 현재 통신하고 있는 주파수 채널입니다. 처음 구입하였을 때 공장초기값은 15이며, 11~26까지 설정이 가능합니다. Zigbee 모듈이 연결되지 않은 상태라면 이 레지스터는 0의 값을 가집니다.

Zigbee PANID(RAM Register의 26번 Address)

Zigbee 모듈이 현재 접속해 있는 WPAN(Wireless Personal Area Network)의 ID를 나타냅니다. Zigbee 모듈을 처음 구입하여 DRC와 연결하였을 때 공장 초기값은 0xBADA이며, Zigbee 모듈이 연결되지 않은 상태라면 이 레지스터는 0xFFFF의 값을 가집니다.

Zigbee SADDR(RAM Register의 28번 Address)

Zigbee 모듈은 2바이트의 Short Address와 8바이트의 Long Address를 가집니다. DRC는 그 중 Short Address를 사용합니다. 이 Address는 한 WPAN내에서 각 Zigbee 모듈을 구분할 수 있도록 합니다. Zigbee 모듈을 처음 구입하여 DRC와 연결하였을 때 공장 초기값은 0xBEAD이며, Zigbee 모듈이 연결되지 않은 상태라면 이 레지스터는 0xFFFF의 값을 가집니다.

Zigbee DSTADDR(RAM Register의 30번 Address)

Zigbee 모듈을 통해서 무선으로 WPAN내의 다른 모듈로 패킷을 전송할 때, 패킷을 받을 Zigbee 모듈의 Short Address를 뜻합니다. Zigbee 모듈을 처음 구입하여 DRC와 연결하였을 때 공장 초기값은 0xBEAD이며, Zigbee 모듈이 연결되지 않은 상태라면 이 레지스터는 0xFFFF의 값을 가집니다.

※ 0xFFFF로 설정하고 패킷을 전송하면, Broadcasting이 되어 같은 WPAN 내의 모든 Zigbee 모듈이 패킷을 받게 됩니다.

Zigbee ACKREQ(RAM Register의 32번 Address)

Zigbee 모듈에서 다른 모듈로 무선 신호를 보낼 때, 다른 무선 통신 장비에 의해 방해를 받거나 장해물이 있다면 통신이 원활하게 이루어지지 않을 수도 있습니다. 이 때, 받는 쪽의 모듈이 받은 패킷에 대한 응답 패킷(ACK Packet)을 보내도록 요청한다면, 보내는 입장에서는 응답 패킷을 받지 못할 경우 재전송하면 되므로 통신의 신뢰성을 높일 수 있습니다. 그러나 응답 패킷을 받게 할 경우 신뢰성이 높아지는 대신에 통신에 필요한 시간은 그만큼 증가하기 때문에, 100ms 이하의 짧은 주기로 계속 패킷을 보내야 하는 경우에는 적합하지 않습니다. r(Zigbee ACKREQ)가 1일 경우에는 통신 시에 응답 패킷을 받도록 하고, 0일 경우에는 응답 패킷을 받지 않도록 합니다. Zigbee 모듈에 저장된 공장 초기값은 1이며, Zigbee 모듈이 연결되지 않은 상태라면 이 레지스터는 2의 값을 가집니다.

Zigbee BACKOFF(RAM Register의 33번 Address)

Zigbee 모듈에서 다른 모듈로 무선 신호를 보낼 때, 주변에 WIFI가 있거나 다른 Zigbee 모듈이 통신 중이라면 사용하려는 주파수 대역을 이미 다른 기기가 사용하고 있을 수 있습니다. 이 경우 통신을 할 수 없고 기존의 통신이 끝난 후에야 통신을 할 수 있습니다. r(Zigbee BACKOFF)가 1일 경우에는 이런 경우 랜덤한 시간 동안 기다린 후에 다시 통신을 시도합니다. r(Zigbee ACKREQ)와 마찬가지로, 신뢰성이 높아지는 대신 통신에 필요한 시간은 증가합니다. r(Zigbee BACKOFF)가 0일 경우에는 기다림 없이 바로 다시 통신을 시도합니다. Zigbee 모듈에 저장된 공장 초기값은 1이며, Zigbee 모듈이 연결되지 않은 상태라면 이 레지스터는 2의 값을 가집니다.

Servo Count(RAM Register의 34번 Address)

현재 제어기에 연결된 서로 다른 ID의 서보 모터 개수를 나타냅니다. 최대 32개까지 서보를 연결할 수 있으며, 32개를 넘는 서보 모터가 연결된 경우 r(Status Error)에 “Servo Status Error”(5번 비트)이 설정되며, r(Error Codes[0]~[4])에 0x33(너무 많은 서보가 연결됨)이 추가 됩니다

Servo ID[0]~[32](RAM Register의 35번 Address)

현재 제어기에 연결된 서보 모터의 ID들을 저장하는 33바이트의 공간입니다. Servo ID[0]부터 Servo ID[r(Servo Count)-1]까지 총 r(Servo Count) 바이트에 모터의 ID가 각각 저장되며, 그 외의 공간에는 0xFE(Broadcasting ID)가 저장됩니다. 서보 모터의 최대 개수는 32개이지만, Servo ID[r(Servo Count)]에 0xFE가 저장되는 규칙을 r(Servo Count)가 32일 때에도 만족시키기 위해서 33바이트의 공간을 사용합니다.

Playing Motion(RAM Register의 68번 Address)

현재 DRC에 저장되어 있는 Motion을 실행하고 있는지 여부를 나타내는 Flag입니다. 실행 중일 때에는 1, 그렇지 않을 때에는 0입니다.

Playing Task(RAM Register의 69번 Address)

현재 DRC에 저장되어 있는 Task를 실행하고 있는지 여부를 나타내는 Flag입니다. 실행 중일 때에는 1, 디버깅 모드로 실행 중일 때에는 3, 그렇지 않을 때에는 0입니다.

Charger Connected(RAM Register의 70번 Address)

현재 DRC의 DC잭을 통해 충전기가 연결 되었는지 여부를 나타내는 Flag입니다. DC잭이 연결되었을 때에는 1, 그렇지 않을 때에는 0입니다.

Buzzer Scale(RAM Register의 71번 Address)

현재 실행중인 버저 음의 음 높이를 나타냅니다. 총 3옥타브 대역의 버저 음을 반음 단위로 낼 수 있습니다. 버저가 울리고 있지 않을 때에는 0으로 유지됩니다. 표에서 음 높이 앞의 숫자는 옥타브를 나타냅니다.

값	음 높이	값	음 높이	값	음 높이	값	음 높이
0	쉼표	10	3라	20	4솔	30	5파
1	3도	11	3라#	21	4솔#	31	5파#
2	3도#	12	3시	22	4라	32	5솔
3	3레	13	4도	23	4라#	33	5솔#
4	3레#	14	4도#	24	4시	34	5라
5	3미	15	4레	25	5도	35	5라#
6	3파	16	4레#	26	5도#	36	5시
7	3파#	17	4미	27	5레	37	6도
8	3솔	18	4파	28	5레#		
9	3솔#	19	4파#	29	5미		

Buzzer Time(RAM Register의 72번 Address)

현재 실행중인 버저 음의 남은 시간을 나타냅니다. 1tick은 6.4ms에 해당합니다. 버저 멜로디를 만들거나 Task에서 음표를 실행하거나 하는 경우에 사용 가능한 음 길이의 수는 총 10개가 있습니다. 예를 들어 8분 음표를 실행하면, Buzzer Time에는 24라는 데이터가 쓰여지고 이 값은 6.4ms마다 1씩 감소하여 0이 될 때까지 총 153.6ms만큼 버저가 울리게 됩니다.

Button Status(RAM Register의 73번 Address)

6가지 버튼의 상태를 의미합니다. 버튼 하나당 한 비트로 상태를 나타내며, 눌린 상태가 1, 눌리지 않은 상태가 0입니다. 가령 OK버튼과 Left 버튼이 동시에 눌렸을 경우, r(Button Status)는 0x12가 됩니다.

비트	값	Button
0	0x01	Mode
1	0x02	OK
2	0x04	Up
3	0x08	Down
4	0x10	Left
5	0x20	Right

Remocon Length(RAM Register의 74번 Address)

리모컨 버튼을 얼마나 길게 누르고 있는지를 나타내는 레지스터입니다. 평소에는 0이나, 리모컨 버튼 신호가 들어온 순간부터 약 125ms마다 1씩 증가합니다. 예를 들어, 리모컨 버튼을 3초 동안 누르고 있었다면 r(Remocon Length)의 값은 24가 될 것입니다. r(Remocon Length)의 값은 최대 240까지 이므로, 최대 30초까지 누른 시간을 인식할 수 있습니다.

Remocon Data(RAM Register의 75번 Address)

현재 눌러져 있는 리모컨 버튼의 Key 값입니다. 리모컨 버튼에는 고유의 Key 값이 할당되어 있습니다. 리모컨 신호가 들어오지 않는 때에는 254(0xFE)의 값을 가집니다.

Input Voltage Value(RAM Register의 76번 Address)

입력 전압의 ADC(Anlog-to-Digital Conversion)값, 즉 Raw Data를 나타냅니다. 실제 전압 값과의 관계는 환산표(page 48)를 참조 바랍니다.

Temperature Value(RAM Register의 77번 Address)

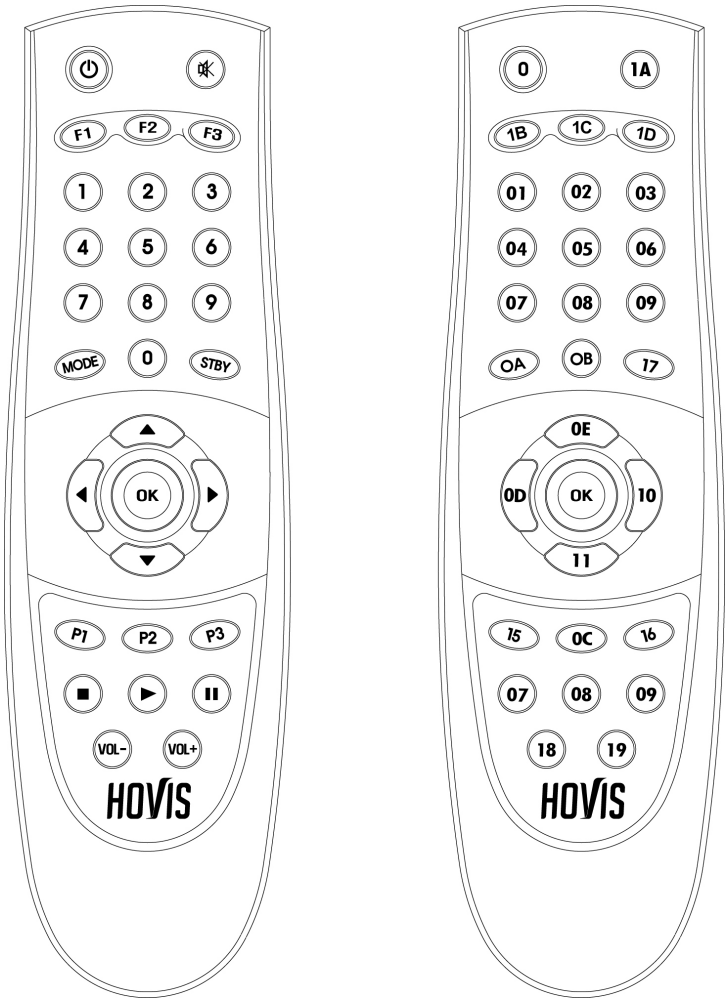
현재 온도의 ADC(Anlog-to-Digital Conversion)값, 즉 Raw Data를 나타냅니다. 실제 온도 값과의 관계는 환산표(page 49)를 참조 바랍니다.

Light Sensor Value(RAM Register의 78번 Address)

DRC에 부착되어 있는 조도 센서로 들어오는 빛의 양을 나타냅니다. r(Light Sensor Value)의 크기가 클수록 더 밝은 환경에서 DRC가 동작하고 있다는 뜻입니다.

Hovis 리모컨의 키맵은 왼쪽 그림과 같습니다. IR Receive 모듈의 Data가 왼쪽의 키 숫자에 대응됩니다.
가령 왼쪽 위의 전원 버튼을 누르면 Data 00이 DRC로 들어오게 됩니다. 이 때 사용자는 IR Receive 모듈에서 Data를 0으로 설정해서 Switch 모듈의 입력으로 연결하면, 전원 버튼을 눌렀을 때 로봇이 어떤 행동을 할 것인지를 프로그래밍 할 수 있습니다.

리모컨에는 채널이 있어서, 리모컨의 채널과 DRC에 설정 된 채널이 같아야만 리모컨 데이터를 받을 수 있습니다.
리모컨의 채널은 1~0의 숫자 + OK를 동시에 누름으로써 설정할 수 있으며, DRC의 채널은 MPSU Ram Data의 RmcChannel 값을 바꾸어 설정할 수 있습니다. 리모컨의 숫자에 대응되는 RmcChannel 값은 아래와 같습니다.



리모컨 버튼	RmcChannel 값
0+OK	97(0x61)
1+OK	98(0x62)
2+OK	99(0x63)
3+OK	100(0x64)
4+OK	101(0x65)
5+OK	102(0x66)
6+OK	103(0x67)
7+OK	104(0x68)
8+OK	105(0x69)
9+OK	106(0x6A)

ADC Port 1 Sensor Type(RAM Register의 79번 Address)

현재 ADC Port 1 로 연결된 센서의 종류를 나타냅니다.

- 0 : 센서가 연결되지 않았음을 뜻합니다.
- 1 : 아날로그 적외선 거리 센서(PSD)가 연결 되었음을 뜻합니다.
- 2 : 디지털 거리 센서가 연결 되었음을 뜻합니다.
- 3 : DRX-0001M이 연결 되었음을 뜻합니다.

ADC Port 2 Sensor Type(RAM Register의 80번 Address)

현재 ADC Port 2로 연결된 센서의 종류를 나타냅니다.

- 0 : 센서가 연결되지 않았음을 뜻합니다.
- 1 : 아날로그 적외선 거리 센서(PSD)가 연결 되었음을 뜻합니다.
- 2 : 디지털 거리 센서가 연결 되었음을 뜻합니다.
- 3 : DRX-0001M이 연결 되었음을 뜻합니다.

ADC Port 1 Sensor Value(RAM Register의 81번 Address)

현재 ADC Port 1로 연결된 센서의 값을 나타냅니다.

- r(ADC Port 1 Sensor Type)이 0일 경우 : 센서가 연결되지 않았으므로 0입니다.
- r(ADC Port 1 Sensor Type)이 1일 경우 : 측정된 거리(cm 단위)가 3~40의 값으로 표시 됩니다.
- r(ADC Port 1 Sensor Type)이 2일 경우 : 디지털 거리 센서의 출력 값이 0이나 1로 표시 됩니다. 10cm보다 물체가 멀리 있으면 1, 가까이 있으면 0의 값이 출력됩니다.
- r(ADC Port 1 Sensor Type)이 3일경우 : 센서가 아니므로 0입니다.

ADC Port 2 Sensor Value(RAM Register의 83번 Address)

현재 ADC Port 2로 연결된 센서의 값을 나타냅니다.

- r(ADC Port 2 Sensor Type)이 0일 경우 : 센서가 연결되지 않았으므로 0입니다.
- r(ADC Port 2 Sensor Type)이 1일 경우 : 측정된 거리(cm 단위)가 3~40의 값으로 표시 됩니다.
- r(ADC Port 2 Sensor Type)이 2일 경우 : 디지털 거리 센서의 출력 값이 0이나 1로 표시 됩니다. 10cm보다 물체가 멀리 있으면 1, 가까이 있으면 0의 값이 출력됩니다.
- r(ADC Port 2 Sensor Type)이 3일경우 : 센서가 아니므로 0입니다.

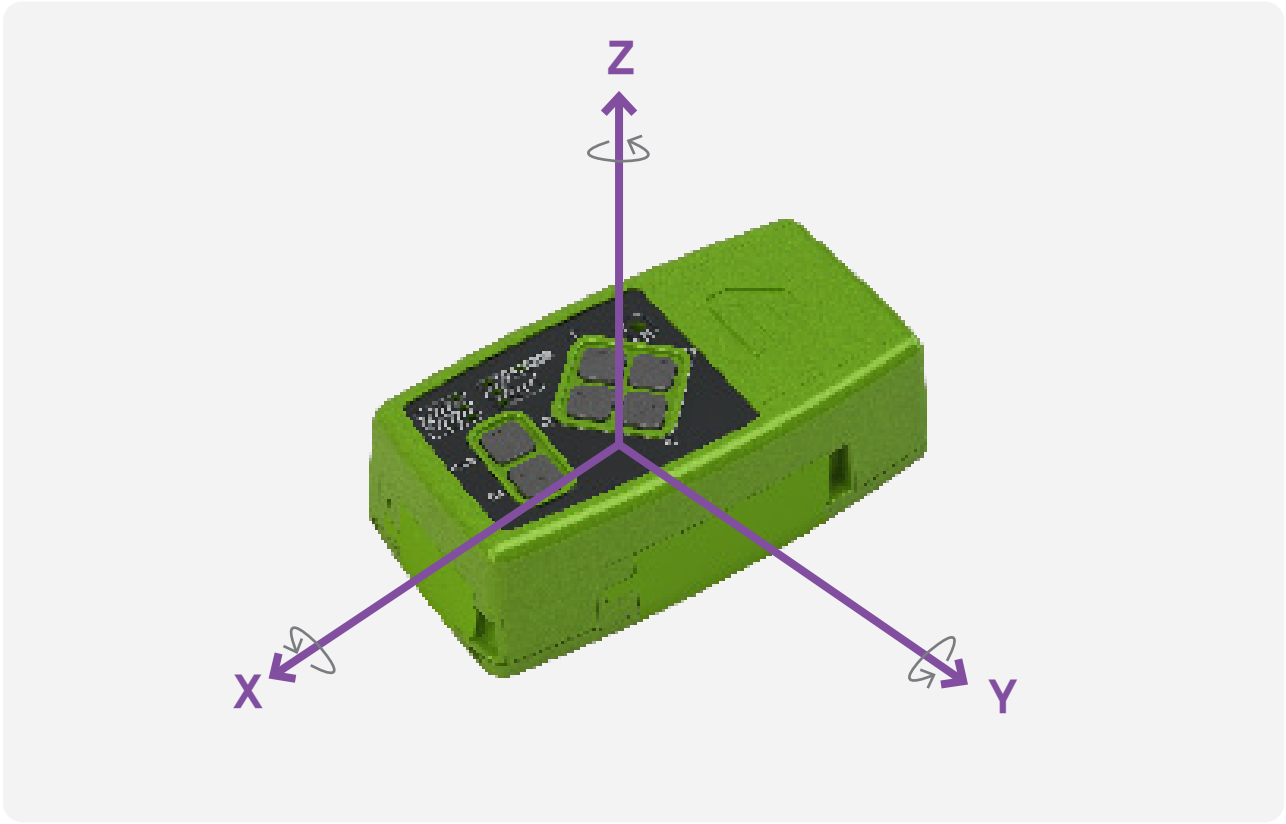
Acc/Gyro Connected(RAM Register의 85번 Address)

가속도/자이로 센서 모듈이 장착되었는지를 나타내는 Flag입니다. 모듈이 장착되었다면 1이 되고, 장착되지 않은 상태라면 0의 값을 가집니다.

Acc X Value, Acc Y Value, Acc Z Value(RAM Register의 86번, 88번, 90번 Address)

가속도 센서의 X, Y, Z축 값의 Raw Data입니다. 가속도 센서는 현재 제어기에 작용하는 가속도 값을 측정합니다. 가속도 센서 각 축의 방향은 아래 그림과 같습니다. 가속도 센서의 각 축은 –4096~4095 의 값을 가질 수 있으며, 256이 1g(중력가속도, 9.8m/s2)에 해당합니다. 따라서 Raw Data를 g의 단위로 바꾸는 계산 식은 아래와 같습니다.

- 가속도(g) = (Raw Data) / 256



Gyro X Value, Gyro Y Value, Gyro Z Value(RAM Register의 92, 94, 96번 Address)

자이로 센서의 X, Y, Z축 값의 Raw Data입니다. 자이로 센서는 현재 제어기가 돌아가는 회전 속도를 측정합니다. 오른손 엄지를 가속도 센서의 축의 방향으로 놓고, 오른손의 나머지 손가락을 접을 때의 방향이 해당 축의 (+) 회전 방향입니다. 다르게 말하면, 축 방향에서 원점을 내려다볼 때 반시계 방향이 (+) 회전 방향입니다. 자이로 센서의 각 축은 –32768~32767의 값을 가질 수 있으며, 약 16.38 이 1°/s(1초당 1° 회전하는 속도)에 해당합니다. 따라서 Raw Data를 °/s의 단위로 바꾸는 계산 식은 아래와 같습니다.

- 각속도(°/s) = (Raw Data) / 32768 X 2000

Sound Detection Flag(RAM Register의 98번 Address)

소리를 감지한 횟수 여부를 나타냅니다. DRC가 소리를 감지하면 r(Sound Detection Flag)가 1증가하고, 그 후 1초내로 다시 소리가 감지되지 않으면 0이 됩니다. 1초내로 또 다시 소리가 감지되었다면, 다시 r(Sound Detection Flag)는 1증가하고, 그 후 1초 동안 소리가 감지되지 않으면 0이 됩니다.

Sound Direction(RAM Register의 99번 Address)

r(Sound Detection Flag)가 1이상 일 때, 가장 늦게 감지한 소리의 방향을 나타내는 레지스터입니다. 값은 –2~2의 범위를 가집니다. 가운데 방향을 기준으로 잡았을 때, 각 값의 의미는 아래와 같습니다.

- –2 : 왼쪽 약 90°에서 소리가 났음을 의미합니다.
- –1 : 왼쪽 약 45°에서 소리가 났음을 의미합니다.
- 0 : 가운데 방향에서 소리가 났음을 의미합니다.
- 1 : 오른쪽 약 45°에서 소리가 났음을 의미합니다.
- 2 : 오른쪽 약 90°에서 소리가 났음을 의미합니다.

r(Sound Detection Flag)가 0일 때는 0의 값을 가집니다.

Tick(RAM Register의 101번 Address)

DRC 전반의 모든 시간에 관련된 작동의 기준이 되는 타이머의 Tick입니다. 0부터 시작하여 60000이 되면 다시 0으로 바뀝니다.

Servo Position[0]~[31](RAM Register의 170번 Address)

현재 연결되어 있는 서보 모터의 위치가 저장되는 2*32바이트짜리 공간입니다. r(Servo ID[n])에 해당하는 ID를 가진 서보의 위치 값이 r(Servo Position[n])에 저장되며, r(Status Check Policy)가 1이라면 실시간으로 계속 업데이트 됩니다. r(Servo Count) 이상의 공간은 0으로 채워집니다.

Touch Status(RAM Register의 100번 Address)

머리 LED 모듈 DRT–HWW10이 연결되었을 경우, 머리 터치 여부를 나타냅니다. 머리에 터치 인식이 될 경우 1이며, 아닐 경우 0입니다. DRT–HWW10이 연결되지 않은 경우에는 0으로 유지됩니다.

Tick(RAM Register의 101번 Address)

수정사항 없음

DRT–HWW1 Connected(RAM Register의 103번 Address)

DRT–HWW10이 연결 되었는지 여부를 나타냅니다. DRT–HWW10이 연결되었다면 1, 연결되지 않았다면 0입니다.

DRC–004TO Connected(RAM Register의 104번 Address)

DRC–004TO가 연결 되었는지 여부를 나타냅니다. DRC–004TO가 연결되었다면 1, 연결되지 않았다면 0입니다.

Servo Status Error & Detail[0]~[31](RAM Register의 106번 Address)

현재 연결되어 있는 서보 모터의 상태 값이 저장되는 2*32바이트짜리 공간입니다. r(Servo ID[n])에 해당하는 서보 모터의 RAM Register에 있는 r(Status Error), r(Status Detail) 값이 r(Servo Status Error & Detail[n])에 저장되며, r(Status Check Policy)가

1이라면 실시간으로 계속 업데이트 됩니다. 이를 통해서 현재 연결된 서보 모터의 상태를 지속적으로 체크할 수 있습니다. r(Servo Count)이상, 즉 연결된 서보 개수 이상의 공간은 0으로 채워집니다. 또한, 연결되었던 서보가 통신이 되지 않아 업데이트가 되지 않을 경우에는 r(Servo Status Error & Detail[n])의 첫 바이트(Status Error에 해당) 값이 0x80으로 바뀌어 통신이 되지 않음을 표시합니다.

Servo Position[0]~[31](RAM Register의 170번 Address)

현재 연결되어 있는 서보 모터의 위치가 저장되는 2*32바이트짜리 공간입니다. r(Servo ID[n])에 해당하는 서보 모터의 RAM Register에 있는 r(Calibrated Position) 값이 r(Servo Position[n])에 저장되며, r(Status Check Policy)가 1이라면 실시간으로 계속 업데이트 됩니다. 이를 통해서 현재 연결된 서보 모터의 위치 값을 지속적으로 체크할 수 있습니다. r(Servo Count)이상, 즉 연결된 서보 개수 이상의 공간은 0으로 채워집니다.

DRT-HWW1 Status Error(RAM Register의 234번 Address)

DRT-HWW1의 r(Status Error)값이 저장됩니다. r(Status Check Policy)가 1이라면 실시간으로 계속 업데이트 됩니다. 이를 통해서 현재 연결된 DRT-HWW1의 상태 에러를 지속적으로 체크할 수 있습니다. r(DRT-HWW1 Connected)가 0이라면, 즉 DRT-HWW1이 연결되지 않았다면 계속 0으로 유지됩니다. 또한, 연결되었던 DRT-HWW1이 통신이 되지 않아 업데이트가 되지 않을 경우에는 0x80으로 바뀌어 통신이 되지 않음을 표시합니다.

DRT-HWW1 Status Detail(RAM Register의 235번 Address)

DRT-HWW1의 r(Status Detail)값이 저장됩니다. r(Status Check Policy)가 1이라면 실시간으로 계속 업데이트 됩니다. 이를 통해서 현재 연결된 DRT-HWW1의 상세 상태를 지속적으로 체크할 수 있습니다. r(DRT-HWW1 Connected)가 0이라면, 즉 DRT-HWW1이 연결되지 않았다면 계속 0으로 유지됩니다.

DRC-004TO Status Error(RAM Register의 236번 Address)

DRC-004TO의 r(Status Error)값이 저장됩니다. r(Status Check Policy)가 1이라면 실시간으로 계속 업데이트 됩니다. 이를 통해서 현재 연결된 DRC-004TO의 상태 에러를 지속적으로 체크할 수 있습니다. r(DRC-004TO Connected)가 0이라면, 즉 DRC-004TO가 연결되지 않았다면 계속 0으로 유지됩니다. 또한, 연결되었던 DRC-004TO가 통신이 되지 않아 업데이트가 되지 않을 경우에는 0x80으로 바뀌어 통신이 되지 않음을 표시합니다.

DRC-004TO Status Detail(RAM Register의 237번 Address)

DRC-004TO의 r(Status Detail)값이 저장됩니다. r(Status Check Policy)가 1이라면 실시간으로 계속 업데이트 됩니다. 이를 통해서 현재 연결된 DRC-004TO의 상세 상태를 지속적으로 체크할 수 있습니다. r(DRC-004TO Connected)가 0이라면, 즉 DRC-004TO가 연결되지 않았다면 계속 0으로 유지됩니다.

프로토콜 형식

개요

DRC를 제어하기 위한 패킷은 크게 두 가지로 나뉩니다. 하나는 PC에서 DRC로 보내는 요청 패킷(Request Packet)이며 다른 하나는 DRC에서 PC로 돌려주는 응답 패킷(ACK Packet)입니다.

설정

통신을 하기 위한 설정은 아래와 같습니다.

Baud Rate : 57,600 / 115,200 / 0.2M / 0.25M / 0.4M / 0.5M / 0.667M

Data Bit : 8

Stop Bit : 1

Parity : None

Flow Control : None

※ PC에 장착된 통신 Port나 USB to Serial Cable의 통신 속도는 Hardware 또는 Driver에 의해서 제한될 수 있습니다. 통신이 안 되는 경우 Baud Rate를 확인해야 합니다. DRC의 공장 초기값은 115,200bps입니다.

패킷 구조

구분	Header		Packet Size	pID	CMD	Check Sum1	Check Sum2	Optional Data
value	0xFF	0xFF	7~223	0~0xFE	세부 참조	세부 참조	세부 참조	세부 참조
bytes	1	1	1	1	1	1	1	MAX 216

1. Header(2바이트)

통신 패킷의 시작을 알립니다. 0xFF, 0xFF의 2바이트로 이루어져 있습니다.

2. Packet Size(1바이트)

Header부터 Optional Data까지 Packet의 총 바이트 수를 의미합니다. Packet Size의 최대 값은 223이므로 Packet Size가 223보다 크면 패킷 에러가 발생합니다.

3. pID(1바이트)

제어할 DRC의 ID를 뜻합니다. pID가 254(0xFE)일 경우 패킷을 받은 모든 DRC가 제어 대상이 되므로 사용에 주의해야 합니다. pID가 254보다 크면 패킷 에러가 발생합니다.

※ 레지스터의 r(ID)와 구분하기 위해, 패킷 내부의 ID 영역은 pID로 표기합니다.

4. CMD(1바이트)

요청 패킷에서는 DRC가 수행할 명령, 응답 패킷에서는 DRC가 받은 명령을 의미합니다. PC에서 DRC로보내는 요청 패킷의 명령은 총 14가지가 있습니다. 응답 패킷은 13가지의 명령이 있으며, 요청 패킷과 구별하기 위하여 요청 패킷의 CMD에 0x40을 Bitwise OR연산 합니다. 예를 들어, 요청 패킷의 EEP_WRITE(0x11) CMD에 대한 응답 패킷의 CMD는 0x51이 됩니다. CMD의 목록은 page 20의 Command Set을, 각 CMD에 대한 자세한 설명은 page 22의 명령 상세 설명을 참조하면 됩니다.
이외에도 DRC에 연결된 서보를 향한 요청 패킷 9가지를 보낼 수 있습니다. DRC는 패킷을 검사한 후 서보로 그 패킷을 전달해주며, 서보로부터 응답 패킷이 오면 다시 그것을 PC로 전달해줍니다. 서보의 요청 패킷과 응답 패킷에 대한 내용은 서보 매뉴얼을 참조하시기 바랍니다.

5. Check Sum1, Check Sum2(2바이트)

Check Sum1, 2는 패킷이 손상 없이 전송되었는지 확인하기 위한 2바이트의 공간입니다. Optional Data가 n바이트 있을 때, Check Sum의 계산식은 아래와 같습니다.

Check Sum1 = (Packet Size ^ pID ^ CMD ^ Data[0] ^ … ^ Data[n–1]) & 0xFE
Check Sum2 = (~(Packet Size ^ pID ^ CMD ^ Data[0] ^ … ^ Data[n–1])) & 0xFE

※ ~는 Bitwise NOT 연산자로, ~A 연산을 수행할 시 A 의 모든 비트를 반전합니다. 예시) ~(01101101)의 연산 결과는 10010010이 됩니다.
※ ^는 Bitwise AND 연산자로, A ^ B 연산을 수행할 시 A와 B의 각 비트를 비교하여 서로 다른 비트만 결과가 1이 됩니다.
예시) 00101110 ^ 10110110 의 연산 결과는 10011000이 됩니다.

6. Optional Data(0~216바이트)

CMD의 종류에 따라서 달라지는 부가적인 데이터입니다. 각 CMD에 따른 Optional Data의 내용은 page 22의 명령 상세 설명을 참조하면 됩니다.

Protocol

Command Set

프로토콜 구조의 CMD부분에 들어가는 Command들의 목록입니다. DRC를 제어하기 위하여 요청 패킷(Request Packet)에는 14가지의 CMD가 존재하며, 이에 응답하는 응답 패킷(ACK Packet)에는 13가지의 CMD가 존재합니다. PC에서 DRC로 요청 패킷을 보내면 DRC는 수신한 패킷에 따른 명령을 수행한 후, 그 결과나 자신의 상태를 응답 패킷의 형태로 PC에 되돌려줍니다. 각 요청 패킷과 응답 패킷의 형태와 형식은 page 22의 명령 상세 설명을 참조하시기 바랍니다.

1. 요청 패킷(PC to DRC)

항목	Cmd	Remark
EEP_WRITE	0x11	EEP Register Address 위치에서 Length개의 값 변경
EEP_READ	0x12	EEP Register Address 위치에서 Length개의 값 요청
RAM_WRITE	0x13	RAM Register Address 위치에서 Length개의 값 변경
RAM_READ	0x14	RAM Register Address 위치에서 Length개의 값 요청
CON_CHECK	0x15	현재 제어기에 어떤 ID의 Servo가 연결되어 있는지 스캔
PLAY_MOTION	0x16	저장되어 있는 Motion을 실행
PLAY_TASK	0x17	저장되어 있는 Task를 실행
PLAY_BUZZ	0x18	저장되어 있는 머리 LED와 Buzzer를 실행
STAT	0x19	제어기의 에러 상태와 가장 최근에 발생한 에러 코드를 요청
ROLLBACK	0x1A	공장 초기화값으로 모든 설정 변수를 변경 변경된 설정은 전원이 꺼진후 다시 기동될때 적용
REBOOT	0x1B	재부팅을 요청
ZIGBEE	0x1C	제어기에 연결된 Zigbee와 관련하여 제어 명령을 전송
REMOCON	0x1D	리모컨 데이터를 보냄
SERVO_FW_UPDATE	0x1E	Servo의 F/W를 업데이트하는 모드에 진입

2. 응답 패킷(DRC to PC)

항목	Cmd	Remark
EEP_WRITE	0x51	r(Status Error)와 r(Status Error Codes[0]) 회신 r(Ack Policy)가 All일 경우에 응답
EEP_READ	0x52	EEP Register Address 위치에서 Len개의 값 회신 r(Ack Policy)가 Read Only, All일 경우에 응답
RAM_WRITE	0x53	r(Status Error)와 r(Status Error Codes[0]) 회신 r(Ack Policy)가 All일 경우에 응답
RAM_READ	0x54	RAM Register Address 위치에서 Len개의 값 회신 r(Ack Policy)가 Read Only, All일 경우에 응답
CON_CHECK	0x55	스캔에 따른 결과로 발견된 서보의 ID들을 회신 r(Ack Policy)가 Read Only, All일 경우에 응답
PLAY_MOTION	0x56	r(Status Error)와 r(Status Error Codes[0]) 회신 r(Ack Policy)가 All일 경우에 응답
PLAY_TASK	0x57	세부 Instruction에 따라서 응답 여부와 형식이 다름(34page 참조)
PLAY_BUZZ	0x58	r(Status Error)와 r(Status Error Codes[0]) 회신 r(Ack Policy)가 All일 경우에 응답
STAT	0x59	r(Status Error)와 r(Status Error Codes[0]) 회신 r(Ack Policy)와 무관하게 항상 응답
ROLLBACK	0x5A	r(Status Error)와 r(Status Error Codes[0]) 회신 r(Ack Policy)가 All일 경우에 응답
REBOOT	0x5B	r(Status Error)와 r(Status Error Codes[0]) 회신 r(Ack Policy)가 All일 경우에 응답
ZIGBEE	0x5C	세부 Instruction에 따라서 응답 여부와 형식이 다름(43page 참조)
REMOCON	–	응답 패킷이 없음.
SERVO_FW_UPDATE	0x5E	r(Status Error)와 r(Status Error Codes[0]) 회신 r(Ack Policy)가 All일 경우에 응답

명령 상세 설명 – EEP_WRITE

1–1. EEP_WRITE – 요청 패킷(0x11)

구분	Packet Size	pID	CMD	Data[0]	Data[1]	Data[2]	···	Data[Length+1]
Value	7+2+Length	0~0xFE	0x11	Address	Length	EEP Data[0]	···	EEP Data[Length–1]

EEP Register의 Address 위치로부터 Length개의 값을 변경합니다. Optional Data에는 Address와 Length, 그리고 Length 개 만큼의 데이터가 들어가므로 Optional Data의 길이는 (2+Length)바이트가 됩니다. 따라서 Packet Size는 기본 7바이트에 (2+Length)바이트를 더해 (9+Length)바이트가 됩니다.
DRC는 이 패킷을 받으면, 비휘발성 레지스터의 주소 Address번부터 (Address+Length–1)번까지의 값을 EEP Data[0]부터 EEP Data[Length–1]의 값으로 변경합니다.

※ DRC의 작동에 직접적으로 영향을 미치는 것은 휘발성 레지스터이므로, EEP_WRITE로 변경한 값은 바로 작동에 반영되지 않습니다. REBOOT CMD를 보내거나 전원을 다시 켜서 다시 부팅 과정을 거쳐야 휘발성 레지스터에 반영됩니다.

예시

■ r(ID)가 253인 DRC의 e(Alarm LED Policy)를 0x3F로 변경하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]	Data[2]
Val–ue	0xFF	0xFF	0x0A(10)	0xFD	0x11	0xD2	0x2C	0x0A	0x01	0x3F

※ CS1, CS2는 각각 Check Sum1, Check Sum2의 약자입니다.
e(Alarm LED Policy)의 주소가 100이고 1바이트짜리 데이터이므로 Address는 10, Length는 1, 그리고 EEP Data[0]은 0x3F가 됩니다. Packet Size는 (9+Length)이므로 100이 되며, page 19의 계산식에 따라 Check Sum1과 Check Sum2가 계산됩니다.

1–2. EEP_WRITE – 응답 패킷(0x51)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	r(ID)	0x51	r(Status Error)	r(Error Codes[0])

r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다. Optional Data의 크기는 2바이트로 고정 되어 있기 때문에 Packet Size는 9바이트로 고정됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x11에 0x40을 Bitwise OR 연산한 0x51이 됩니다.

응답 조건

EEP_WRITE는 r(ACK Policy)가 2(모든 패킷에 대하여 응답)일 경우에만 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 r(ACK Policy)가 2라도 응답하지 않습니다.

예시

- r(ID)가 253인 DRC의 e(Alarm LED Policy)를 0x3F로 변경하는 패킷을 받은 후의 응답 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x51	0xA4	0x5A	0x00	0x00

현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

명령 상세 설명 – EEP_READ

2–1. EEP_READ – 요청 패킷(0x12)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	0~0xFE	0x12	Address	Length

EEP Register의 Address 위치로부터 Length개의 값을 읽습니다. Optional Data에는 Address와 Length가 들어가므로 Optional Data의 길이는 2바이트가 됩니다. 따라서 Packet Size는 기본 7바이트에 2바이트를 더해 9바이트가 됩니다.
DRC는 이 패킷을 받으면, 비휘발성 레지스터의 주소 Address번부터 (Address+Length–1)번까지의 값을 응답 패킷을 통해서 보내줍니다.

예시

- r(ID)가 253인 DRC의 e(Min Voltage), e(Max Voltage), e(Max Temperature)를 읽어오는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x12	0xE8	0x16	0x0C	0x03

e(Min Voltage)의 주소가 120이고 그 이후로 3바이트를 읽어야 하므로 Address는 12, Length는 30이 됩니다. Packet Size는 90이며, page 19의 계산식에 따라 Check Sum1과 Check Sum2가 계산됩니다.

2–2. EEP_READ – 응답 패킷(0x52)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2+Length+2	r(ID)	0x52	Address	Length

구분	Data[2]	···	Data[Length+1]	Data[Length+2]	Data[Length+3]
Value	EEP Data[0]	···	EEP Data[Length–1]	r(Status Error)	r(Error Codes[0])

비휘발성 레지스터의 주소 Address번부터 (Address+Length–1)번까지의 값을 EEP Data[0]부터 EEP Data[Length–1]에 실어서 보냅니다. 또한 r(Status Error)와 r(Error Codes[0])의 값도 반환합니다. Optional Data에는 Address와 Length, 그리고 Length 개 만큼의 데이터, 또한 r(Status Error)와 r(Error Codes[0])이 들어가므로 Optional Data의 길이는 (2+Length+2)바이트가 됩니다. 따라서 Packet Size는 기본 7바이트에 (4+Length)바이트를 더해 (11+Length)바이트가 됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x12에 0x40을 Bitwise OR 연산한 0x52이 됩니다.

응답 조건

EEP_READ는 r(ACK Policy)가 1(Read 명령일 때만 응답), 2(모든 패킷에 대하여 응답)일 경우에 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 응답하지 않습니다.

예시

■ r(ID)가 253인 DRC의 e(Min Voltage), e(Max Voltage), e(Max Temperature)를 읽어오는 패킷에 대한 응답 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x0E(14)	0xFD	0x52	0xA6	0x58	0x0C	0x03

구분	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]
Value	0x5F	0x88	0xDF	0x00	0x00

Address 12번부터 3바이트의 데이터를 Data[2]~Data[4]에 실어서 보냅니다. Data[2]에는 e(Min Voltage), Data[3]에는 e(Max Voltage), Data[4]에는 e(Max Temperature)가 각각 실려 있습니다. 그리고 Data[5]와 Data[6]에 현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

명령 상세 설명 – RAM_WRITE

3-1. RAM_WRITE – 요청 패킷(0x13)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]	Data[2]	…	Data[Length+1]
Value	7+2+Length	0~0xFE	0x13	Address	Length	RAM Data[0]	…	RAM Data[Length-1]

RAM Register의 Address 위치로부터 Length개의 값을 변경합니다. Optional Data에는 Address와 Length, 그리고 Length 개 만큼의 데이터가 들어가므로 Optional Data의 길이는 (2+Length)바이트가 됩니다. 따라서 Packet Size는 기본 7바이트에 (2+Length)바이트를 더해 (9+Length)바이트가 됩니다.

DRC는 이 패킷을 받으면, 휘발성 레지스터의 주소 Address번부터 (Address+Length-1)번까지의 값을 RAM Data[0]부터 RAM Data[Length-1]의 값으로 변경합니다.

예시

- r(ID)가 253인 DRC의 r(Status Error)와 r(Error Codes[0]~[4])를 모두 0x00로 변경하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x0F(15)	0xFD	0x13	0xF6	0x08	0x10	0x06

구분	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
Value	0x00	0x00	0x00	0x00	0x00	0x00

r(Status Error)의 주소가 16이고, 그 이후로 6바이트의 값을 변경해야 하므로 Address는 16, Length는 6, 그리고 RAM Data[0]~RAM Data[5]는 0x00이 됩니다. Packet Size는 (9+Length)이므로 15가 되며, page 19의 계산식에 따라 Check Sum1과 Check Sum2가 계산됩니다.

※ r(Status Error)와 r(Error Codes[0]~[4])는 쓰고 읽기가 모두 가능한 레지스터이지만, MPSU의 현재 상태를 나타내기 때문에 임의로 값을 수정할 수 없습니다. 이 6바이트의 레지스터에는 현재 값을 그대로 쓰거나, 모두 0x00을 쓰거나 둘 중 하나만 할 수 있습니다. 그렇지 않으면 r(Status Error)에 "Invalid Packet"(2번 비트)이 설정되며, r(Error Codes[0]~[4])에 0x73(r(Status Error)와 r(Error Codes[0]~[4])에 잘못된 쓰기 명령)가 추가됩니다.

3–2. RAM_WRITE – 응답 패킷(0x53)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	r(ID)	0x53	r(Status Error)	r(Error Codes[0])

r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다. Optional Data의 크기는 2바이트로 고정 되어 있기 때문에 Packet Size는 9바이트로 고정됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x13에 Bitwise OR 연산한 0x53이 됩니다.

응답 조건

RAM_WRITE는 r(ACK Policy)가 2(모든 패킷에 대하여 응답)일 경우에만 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 r(ACK Policy)가 2라도 응답하지 않습니다.

예시

- r(ID)가 253인 DRC의 r(Status Error)와 r(Error Codes[0]~[4])를 모두 0x00로 변경하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x53	0xA6	0x58	0x00	0x00

현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

명령 상세 설명 – EEP_READ

4-1. RAM_READ – 요청 패킷(0x14)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	0~0xFE	0x14	Address	Length

RAM Register의 Address 위치로부터 Length개의 값을 읽습니다. Optional Data에는 Address와 Length가 들어가므로 Optional Data의 길이는 2바이트가 됩니다. 따라서 Packet Size는 기본 7바이트에 2바이트를 더해 9바이트가 됩니다.
DRC는 이 패킷을 받으면, 휘발성 레지스터의 주소 Address번부터 (Address+Length-1)번까지의 값을 응답 패킷을 통해서 보내줍니다.

예시

- r(ID)가 253인 DRC의 r(Min Voltage), r(Max Voltage), r(Max Temperature)를 읽어오는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x14	0xE6	0x18	0x05	0x03

r(Min Voltage)의 주소가 50이고 그 이후로 3바이트를 읽어야 하므로 Address는 5, Length는 30이 됩니다. Packet Size는 90이며, page 00의 계산식에 따라 Check Sum1과 Check Sum2가 계산됩니다.

4–2. RAM_READ – 응답 패킷(0x54)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2+Length+2	r(ID)	0x54	Address	Length

구분	Data[2]	···	Data[Length+1]	Data[Length+2]	Data[Length+3]
Value	RAM Data[0]	···	RAM Data[Length–1]	r(Status Error)	r(Error Codes[0])

휘발성 레지스터의 주소 Address번부터 (Address+Length–1)번까지의 값을 RAM Data[0]부터 RAM Data[Length–1]에 실어서 보냅니다. 또한 r(Status Error)와 r(Error Codes[0])의 값도 반환합니다. Optional Data에는 Address와 Length, 그리고 Length 개 만큼의 데이터, 또한 r(Status Error)와 r(Error Codes[0])이 들어가므로 Optional Data의 길이는 (2+Length+2)바이트가 됩니다. 따라서 Packet Size는 기본 7바이트에 (4+Length)바이트를 더해 (11+Length)바이트가 됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x14에 0x40을 Bitwise OR 연산한 0x54이 됩니다.

응답 조건

RAM_WRITE는 r(ACK Policy)가 2(모든 패킷에 대하여 응답)일 경우에만 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 r(ACK Policy)가 2라도 응답하지 않습니다.

예시

- r(ID)가 253인 DRC의 r(Min Voltage), r(Max Voltage), r(Max Temperature)를 읽어오는 패킷에 대한 응답 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x54	0xA8	0x56	0x05	0x03

구분	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]
Value	0x5F	0x88	0xDF	0x00	0x00

Address 5번부터 3바이트의 데이터를 Data[2]~Data[4]에 실어서 보냅니다. Data[2]에는 r(Min Voltage), Data[3]에는 r(Max Voltage), Data[4]에는 r(Max Temperature)가 각각 실려 있습니다. 그리고 Data[5]와 Data[6]에 현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

명령 상세 설명 – CON_CHECK

5–1. CON_CHECK – 요청 패킷(0x15)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]	···	Data[Length]
Value	7+1+Length	0~0xFE	0x15	Length	ID[0]	···	ID[Length–1]

현재 DRC에 ID[0]~ID[Length–1]의 ID를 가지는 서보가 연결되어 있는지 확인합니다. Optional Data에는 Length와 Length개 만큼의 ID가 들어가므로 Optional Data의 길이는 (1+Length)바이트가 됩니다. 따라서 Packet Size는 기본 7바이트에 (1+Length)바이트를 더해 (8+Length)바이트입니다.
DRC는 이 패킷을 받으면, ID[0]부터 ID[Length–1]까지 서보들에 통신을 시도하고, 이 중에서 실제로 연결되어 통신에 성공한 개수와 ID를 응답 패킷을 통해 보내줍니다.

※ Length가 0인 경우에는 0~253까지 모든 ID를 스캔합니다.

예시

■ r(ID)가 253인 DRC에 ID가 0, 1, 2, 3, 4인 서보가 연결되었는지 확인하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]
Value	0xFF	0xFF	0x0D(13)	0xFD	0x15	0xE4	0x1A	0x05

구분	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
Value	0x00	0x01	0x02	0x03	0x04

서보의 개수가 5개이므로 Length는 5가 되고, Data[1]부터 Data[5]에 각각 검사할 ID인 0, 1, 2, 3, 4가 각각 들어갑니다. Packet Size는 (8+Length)이므로 13이 되며, page 19의 계산식에 따라 Check Sum1과 Check Sum2가 계산됩니다.

5–2. CON_CHECK – 응답 패킷(0x55)

구조

구분	Packet Size	pID	CMD	Data[0]
Value	7+1+Length+2	r(ID)	0x55	Length

구분	Data[1]	···	Data[Length]	Data[Length+1]	Data[Length+2]
Value	ID[0]	···	ID[Length–1]	r(Status Error)	r(Error Codes[0])

요청 패킷을 받아서 ID스캔을 수행한 후, 실제로 통신이 되었던 서보의 개수를 Length에, 그 ID값들을 ID[0]부터 ID[Length–1]에 실어서 보냅니다. 또한 r(Status Error)와 r(Error Codes[0])의 값도 반환합니다. Optional Data에는 Length, 그리고 Length개 만큼의 ID, 또한 r(Status Error)와 r(Error Codes[0])이 들어가므로 Optional Data의 길이는 (1+Length+2)바이트가 됩니다. 따라서 Packet Size는 기본 7바이트에 (3+Length)바이트를 더해 (10+Length)바이트가 됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x15에 0x40을 Bitwise OR 연산한 0x55이 됩니다.

응답 조건

CON_CHECK는 r(ACK Policy)가 1(Read 명령일 때만 응답), 2(모든 패킷에 대하여 응답)일 경우에 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 응답하지 않습니다.

예시

■ r(ID)가 253인 DRC에 ID가 0, 1, 2, 3, 4인 서보가 연결되었는지 확인하는 패킷을 받았는데, 검사 결과 실제로는 0, 1, 2만 연결되어 있는 경우의 응답 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]
Value	0xFF	0xFF	0x0D(13)	0xFD	0x55	0xA4	0x5A	0x03

구분	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
Value	0x00	0x01	0x02	0x00	0x00

ID 스캔 결과 0, 1, 2만 연결되어 있으므로, Length를 나타내는 Data[0]는 3이 되며 Data[1]~Data[3]에 0, 1, 2가 차례로 들어갑니다. 그리고 Data[4]와 Data[5]에 현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다. Packet Size는 (10+Length)이므로 13이 됩니다.

명령 상세 설명 – PLAY_MOTION

6-1. PLAY_MOTION – 요청 패킷(0x16)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	0~0xFE	0x16	Motion No.	Motion Ready Flag

현DR-SIM을 사용하여 DRC에 모션을 저장하면, 모션은 0부터 127까지의 번호를 가지게 됩니다. PLAY_MOTION 패킷은 현재 DRC에 저장된 모션을 실행하며, Motion No.는 실행할 모션의 번호를 나타냅니다. Motion Ready Flag는 모션 준비 자세를 취할지 여부를 결정합니다. Motion Ready Flag를 1로 해서 패킷을 보내면, 해당 모션의 첫 프레임만을 느리게 실행하게 됩니다. 실행하려는 모션을 Motion Ready Flag를 1로 해서 한번 요청한 후, 조금 기다렸다가 0으로 해서 다시 요청하면, 걷기 등의 모션을 실행할 때 급격한 자세 변화로 인한 넘어짐과 모터 손상을 예방할 수 있습니다. 또한, 모션 번호를 254(0xFE)로 해서 패킷을 보내면 현재 실행중인 모션이 중지됩니다.

Motion No.와 Motion Ready Flag에 따른 동작을 정리하면 아래와 같습니다.

Motion No	Motion Ready Flag	동작
0~127	0	모션 실행
0~127	1	모션의 첫 프레임만 실행
254	0~1	모션 정지

예시

■ r(ID)가 253인 DRC에서 모션 1번을 실행하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x16	0xE2	0x1C	0x01	0x00	0x00

모션 번호가 1번이므로 Motion No.에는 1이 들어가며 Motion Ready Flag은 0입니다.

■ r(ID)가 253인 DRC에서 모션 2번을 첫 프레임만 실행하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x16	0xE0	0x1E	0x02	0x01

모션 번호가 2번이므로 Motion No.에는 2가 들어가며 Motion Ready Flag은 1입니다.

6–2. PLAY_MOTION – 응답 패킷(0x56)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	r(ID)	0x56	r(Status Error)	r(Error Codes[0])

r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다. Optional Data의 크기는 2바이트로 고정 되어 있기 때문에 Packet Size는 9바이트로 고정됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x16에 Bitwise OR 연산한 0x560이 됩니다

응답 조건

PLAY_MOTION은 r(ACK Policy)가 2(모든 패킷에 대하여 응답)일 경우에만 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 r(ACK Policy)가 2라도 응답하지 않습니다.

예시

- r(ID)가 253인 DRC의 2번 모션을 실행하는 것에 대한 응답 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x56	0xA2	0x5C	0x00	0x00

현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

명령 상세 설명 – PLAY_TASK

7-1. PLAY_TASK – 요청 패킷(0x17)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+1	0~0xFE	0x17	Instruction	Motion Ready Flag

DR-Visual Logic을 사용하여 DRC에 저장한 Task를 실행합니다. PLAY_TASK는 Instruction에 따라서 4개의 명령으로 나뉘는데, Instruction에 따라서 수행하는 일이 다릅니다.

- Instruction이 0일 때에는 일반 모드로 Task를 실행합니다.
- Instruction이 1일 때에는 디버깅 모드로 Task를 실행합니다.
- Instruction이 2일 때에는 Task를 한 Step만 진행한 후 멈춥니다. 이 Instruction은 디버깅 모드일 때에만 의미가 있습니다.
- Instruction이 254일 때에는 Task를 정지합니다. 일반 모드와 디버깅 모드에 관계 없이 Task가 정지됩니다.

예시

- r(ID)가 253인 DRC에서 저장된 Task를 실행하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]
Value	0xFF	0xFF	0x08(8)	0xFD	0x17	0xE2	0x1C	0x00

Task 실행이므로 Instruction은 0입니다.

- r(ID)가 253인 DRC에서 저장된 Task를 디버깅 모드로 실행하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]
Value	0xFF	0xFF	0x08(8)	0xFD	0x17	0xE2	0x1C	0x01

디버깅 모드에서 Task 실행이므로 Instruction은 1입니다.

- r(ID)가 253인 DRC가 디버깅 모드일 때, Task를 한 Step 진행하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]
Value	0xFF	0xFF	0x08(8)	0xFD	0x17	0xE0	0x1E	0x02

디버깅 모드 중에 Task를 한 Step 진행하므로 Instruction은 2입니다.

7–2. PLAY_TASK – 응답 패킷(0x57)

구조 – 디버깅 응답 패킷

구분	Packet Size	pID	CMD	Data[0]	Data[1]	Data[2]	Data[3]
Value	7+4	r(ID)	0x57	Program Counter L	Program Counter H	r(Status Error)	r(Error Codes[0])

구조 – 상태 응답 패킷

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	r(ID)	0x57	r(Status Error)	r(Error Codes[0])

PLAY_TASK의 응답 패킷은 Instruction에 따라서 두 종류로 나뉩니다.
디버깅 응답 패킷은 현재 Task의 어떤 부분을 실행하고 있는지 Program Counter L과 Program Counter H의 두 바이트를 통해서 알려줍니다. 이 정보를 바탕으로 DR–Visual Logic에서 Task를 디버깅할 때 현재 어느 위치의 코드가 실행 중인지 알 수 있습니다. 디버깅 응답 패킷에는 추가로 r(Status Error)와 r(Error Codes[0])의 값도 포함됩니다. Optional Data의 크기가 4바이트로 고정되기 때문에 Packet Size는 11바이트입니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x17에 Bitwise OR 연산한 0x57이 됩니다.
상태 응답 패킷은 r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다. Optional Data의 크기는 2바이트로 고정 되어 있기 때문에 Packet Size는 9바이트로 고정됩니다.
디버깅 응답 패킷은 디버깅에 관련된 상황일 때, 그리고 상태 응답 패킷은 나머지 상황일 때 사용됩니다. Instruction에 따라서 어떤 응답 패킷을 보내는지는 아래와 같습니다.

Instruction	성공 시	실패 시
0	상태	상태
1	디버깅	상태
2	디버깅	상태
254	상태	상태

Task 실행(Instruction 0)이나 Task 중지(Instruction 254)같은 경우에는 상태 응답 패킷으로 응답합니다. 디버깅 모드로 Task 실행(Instruction 1)이나 한 Step 진행 후 멈춤(Instruction 2)과 같이 디버깅과 관련된 Instruction에는 디버깅 응답 패킷으로 응답합니다. 그러나 이미 Task가 실행 중인데 Instruction 1을 보냈다거나, Task가 실행 중이 아닌데 Instruction 2를 보냈다거나 등과 같이 명령이 제대로 수행되지 않은 경우에는 상태 응답 패킷으로 응답합니다.

응답 조건

상태 응답 패킷을 보내야 할 경우에는 r(ACK Policy)가 2(모든 패킷에 대하여 응답)일 경우에만 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 r(ACK Policy)가 2라도 응답하지 않습니다. 디버깅 응답 패킷을 보내야 할 경우에는 r(ACK Policy)가 1(Read 명령일 때만 응답), 2(모든 패킷에 대하여 응답)일 경우에 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 응답하지 않습니다.

예시

■ r(ID)가 253인 DRC에서 저장된 Task를 실행하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x57	0xA2	0x5C	0x00	0x00

요청 패킷의 Instruction이 0이므로, 현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

■ r(ID)가 253인 DRC에서 저장된 Task를 디버깅 모드로 실행하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]	Data[2]	Data[3]
Value	0xFF	0xFF	0x0B(11)	0xFD	0x57	0xAA	0x54	0x0B	0x00	0x00	0x00

요청 패킷의 Instruction이 1이므로, 디버깅 응답 패킷을 보냅니다. Data[0]과 Data[1]에는 현재 프로그램 카운터의 위치가 들어가 있습니다. 디버깅을 시작한 상태에서 현재 위치는 0x000B입니다. Data[2]와 Data[3]은 현재 상태와 가장 최근의 에러 코드를 나타냅니다.

■ r(ID)가 253인 DRC가 디버깅 모드일 때, Task를 한 Step 진행하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]	Data[2]	Data[3]
Value	0xFF	0xFF	0x0B(11)	0xFD	0x57	0x86	0x78	0x26	0x00	0x00	0x00

요청 패킷의 Instruction이 2이므로, 디버깅 응답 패킷을 보냅니다. Data[0]과 Data[1]를 보면 프로그램 카운터의 현재 위치는 0x0026임을 알 수 있습니다. Data[2]와 Data[3]은 현재 상태와 가장 최근의 에러 코드를 나타냅니다.

명령 상세 설명 – PLAY_BUZZ

8-1. PLAY_BUZZ – 요청 패킷(0x18)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	0~0xFE	0x18	Reserved	Buzz No.

DRC에 저장된 Buzzer를 실행합니다. Buzzer의 번호는 1부터 63까지가 가능하며, Data[1]의 Buzz No.에 실행할 버저의 번호를 담아서 패킷을 보내게 됩니다. Data[0]은 다른 데이터가 들어가기 위해서 Reserve 되어 있는 공간이므로 무조건 0을 넣으면 됩니다.

예시

■ r(ID)가 253인 DRC에서 5번 버저를 실행하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x18	0xE8	0x16	0x00	0x05

5번 버저를 실행해야 하므로 Data[1]에는 5가 들어갑니다.

8-2. PLAY_BUZZ – 응답 패킷(0x58)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	r(ID)	0x58	r(Status Error)	r(Error Codes[0])

r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다. Optional Data의 크기는 2바이트로 고정 되어 있기 때문에 Packet Size는 9바이트로 고정됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x18에 Bitwise OR 연산한 0x58이 됩니다.

응답 조건

PLAY_Buzz은 r(ACK Policy)가 2(모든 패킷에 대하여 응답)일 경우에만 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 r(ACK Policy)가 2라도 응답하지 않습니다.

예시

- r(ID)가 253인 DRC에 보낸 5번 버저 실행 요청 패킷에 대한 응답 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x58	0xAC	0x52	0x00	0x00

현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

명령 상세 설명 – STAT

9–1. STAT – 요청 패킷(0x19)

구조

구분	Packet Size	pID	CMD
Value	7+2	0~0xFE	0x18

DRC의 현재 상태를 요청합니다. DRC는 이 패킷을 받으면 r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다.

예시

■ r(ID)가 253인 DRC에 STAT 명령을 요청하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2
Value	0xFF	0xFF	0x07(7)	0xFD	0x19	0xE2	0x1C

9–2. STAT – 응답 패킷(0x59)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	r(ID)	0x59	r(Status Er–ror)	r(Error Codes[0])

r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다. Optional Data의 크기는 2바이트로 고정 되어 있기 때문에 Packet Size는 9바이트로 고정됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x19에 Bitwise OR 연산한 0x59이 됩니다.

응답 조건

STAT은 r(ACK Policy)와 관계 없이 모든 경우에 응답합니다. 요청 패킷의 pID가 254(Broadcasting ID)일 경우에도 응답합니다.

예시

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x59	0xAC	0x52	0x00	0x00

현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

명령 상세 설명 – ROLLBACK

10-1. ROLLBACK – 요청 패킷(0x1A)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	0~0xFE	0x1A	ID Skip	Baud Skip

DRC에 저장되어 있는 공장 초기 값으로 비휘발성 레지스터를 초기화합니다. 초기화된 비휘발성 레지스터는 다음에 전원을 켜거나 리셋을 건 후부터 동작에 영향을 미치게 됩니다. Data[0]과 Data[1]의 ID Skip, Baud Skip은 각각 초기화 시 e(ID)와 e(Baud Rate)를 초기화 대상에서 제외할 것인지 여부를 결정합니다. ID Skip이 1이면 e(ID)를 초기화하지 않고, Baud Skip이 1이면 e(Baud Rate)를 초기화하지 않습니다.

예시

■ r(ID)가 253인 DRC를 e(ID)만 빼고 공장 초기화 하는 패킷에 대한 응답패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x1A	0xEE	0x10	0x01	0x00

e(ID)만을 빼고 초기화 하려는 패킷이므로 ID Skip은 1, Baud Skip은 0이 됩니다.

■ r(ID)가 253인 DRC를 e(ID), e(Baud Rate)만 빼고 공장 초기화 하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x1A	0xEE	0x10	0x01	0x01

e(ID)와 e(Baud Rate)를 빼고 초기화 하려 하므로 ID Skip은 1, Baud Skip도 1이 됩니다.

10–2. ROLLBACK – 응답 패킷(0x5A)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	r(ID)	0x5A	r(Status Error)	r(Error Codes[0])

r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다. Optional Data의 크기는 2바이트로 고정 되어 있기 때문에 Packet Size는 9바이트로 고정됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x1A에 Bitwise OR 연산한 0x5A이 됩니다.

응답 조건

ROLLBACK은 r(ACK Policy)가 2(모든 패킷에 대하여 응답)일 경우에만 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 r(ACK Policy)가 2라도 응답하지 않습니다.

예시

- r(ID)가 253인 DRC를 e(ID)만 빼고 공장 초기화 하는 패킷에 대한 응답패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x5A	0xAE	0x50	0x00	0x00

현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

명령 상세 설명 – REBOOT

11-1. REBOOT – 요청 패킷(0x1B)

구조

구분	Packet Size	pID	CMD
Value	7	0~0xFE	0x1B

DRC에 SW 리셋을 요청합니다. DRC는 이 패킷을 받으면 자신을 리셋 하여 초기 부팅을 다시 진행합니다.

예시

■ r(ID)가 253인 DRC에 REBOOT 명령을 요청하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2
Value	0xFF	0xFF	0x07(7)	0xFD	0x1B	0xE0	0x1E

11-2. REBOOT – 응답 패킷(0x5B)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]
Value	7+2	r(ID)	0x5B	r(Status Er–ror)	r(Error Codes[0])

r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다. Optional Data의 크기는 2바이트로 고정 되어 있기 때문에 Packet Size는 9바이트로 고정됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x1B에 Bitwise OR 연산한 0x5B이 됩니다.

응답 조건

REBOOT은 r(ACK Policy)가 2(모든 패킷에 대하여 응답)일 경우에만 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 r(ACK Policy)가 2라도 응답하지 않습니다.

예시

■ r(ID)가 253인 DRC에 보낸 REBOOT 요청 패킷에 대한 응답 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x5B	0xAE	0x50	0x00	0x00

현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

명령 상세 설명 – ZIGBEE

12-1. ZIGBEE – 요청 패킷(0x1C)

구조

구분	Packet Size	pID	CMD	Data[0]
Value	7+1	0~0xFE	0x1C	Instruction

DRC에 장착되어 있는 Zigbee 모듈에 관련된 제어 명령을 수행합니다. ZIGBEE는 Instruction에 따라서 6개의 명령으로 나뉩니다. 휘발성 레지스터 맵에는 r(Zigbee PANID), r(Zigbee SADDR), r(Zigbee DSTADDR), r(Zigbee ACKREQ), r(Zigbee BACKOFF) 등 5가지 항목(총 8바이트)의 Zigbee와 관련된 레지스터가 있습니다. 각 레지스터는 Zigbee 모듈 내부에 저장된 속성값에 대응됩니다. Zigbee를 통한 무선 통신과 연결 케이블을 통한 유선 통신은 동시에 이루어질 수 없으므로 주의해야 합니다.

- Instruction이 0일 때에는 Zigbee 모듈의 속성값을 휘발성 레지스터로 읽어옵니다.
- Instruction이 1일 때에는 휘발성 레지스터의 속성값을 Zigbee 모듈로 덮어씁니다.
- Instruction이 2일 때에는 Zigbee 모듈의 속성값을 출고 상태로 초기화 합니다.
- Instruction이 3일 때에는 Zigbee 모듈을 리셋합니다.
- Instruction이 4일 때에는 유선 통신 모드(연결 케이블과 COM PORT를 통한 통신)로 전환합니다.
- Instruction이 5일 때에는 무선 통신 모드(Zigbee를 통한 통신)로 전환합니다.

예시

- r(ID)가 253인 DRC에서 Zigbee의 속성값을 읽어오는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]
Value	0xFF	0xFF	0x08(8)	0xFD	0x1C	0xE8	0x16	0x00

모듈로부터 RAM으로 속성값을 읽어오는 것이므로 Instruction은 0입니다.

■ r(ID)가 253인 DRC에서 Zigbee 모듈을 출고 상태로 초기화하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]
Value	0xFF	0xFF	0x08(8)	0xFD	0x1C	0xEA	0x14	0x02

출고 상태로 초기화이므로 Instruction은 2입니다.

■ r(ID)가 253인 DRC를 무선 통신으로 전환하는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]
Value	0xFF	0xFF	0x08(8)	0xFD	0x1C	0xEC	0x12	0x05

무선 통신 모드 전환의 Instruction은 5입니다.

12–2. ZIGBEE – 응답 패킷(0x1C)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]	Data[2]
Value	7+3	r(ID)	0x5C	Success	r(Status Error)	r(Error Codes[0])

ZIGBEE의 응답 패킷에는 해당 명령의 성공 여부를 나타내는 Success라는 항목이 Data[0]에 실립니다. Success는 Zigbee 명령이 성공했을 시에 1, Zigbee 모듈이 장착되어 있지 않거나 통신 상의 문제로 실패했을 시에 0이 됩니다. Optional Data에는 추가로 r(Status Error)와 r(Error Codes[0])의 값도 포함됩니다. Optional Data의 크기가 3바이트로 고정되기 때문에 Packet Size는 10바이트입니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x1C에 Bitwise OR 연산한 0x5C이 됩니다.

응답 조건

ZIGBEE는 r(ACK Policy)가 1(Read 명령일 때만 응답), 2(모든 패킷에 대하여 응답)일 경우에 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 응답하지 않습니다.

예시

■ r(ID)가 253인 DRC에서 Zigbee의 속성값을 읽어오는 패킷에 대한 응답 패킷(Zigbee 장착 시)

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]	Data[2]
Value	0xFF	0xFF	0x0A(10)	0xFD	0x5C	0xAA	0x54	0x01	0x00	0x00

Zigbee가 장착되어 있으므로 통신이 정상적으로 이루어졌을 것이고, Success의 값은 1입니다.

예시

■ r(ID)가 253인 DRC에서 Zigbee를 출고 상태 초기화하는 패킷에 대한 응답 패킷(Zigbee 장착 시)

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]	Data[2]
Value	0xFF	0xFF	0x0A(10)	0xFD	0x5C	0xAA	0x54	0x01	0x00	0x00

마찬가지로 Zigbee 값이 정상적으로 초기화 되었고, Success의 값은 1입니다.

■ r(ID)가 253인 DRC를 무선 통신으로 전환하는 패킷에 대한 응답 패킷(Zigbee 미 장착 시)

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]	Data[2]
Value	0xFF	0xFF	0x0A(10)	0xFD	0x5C	0xAA	0x54	0x00	0x00	0x00

Zigbee가 장착되어 있지 않으므로, 무선 통신 전환 시도는 실패합니다. 따라서 Success의 값은 0입니다.

명령 상세 설명 – REMOCON

13-1. REMOCON – 요청 패킷(0x1D)

구조

구분	Packet Size	pID	CMD	Data[0]	Data[1]	Data[2]
Value	7+3	0~0xFE	0x1D	Channel	Length	Data

DRC에는 리모컨 리시버 모듈을 붙일 수 있어서, IR 리모컨으로 제어 명령을 내릴 수 있습니다. 하지만 IR 리모컨이 없는 환경이나 Zigbee 모듈을 통한 무선 환경에서 IR 리모컨 대신에 패킷을 통해 제어를 할 수 있도록 하는 것이 REMOCON 명령입니다. Data[0]에는 리모컨의 Channel(0x61~0x6A)가 들어가며, Data[1]에는 리모컨 버튼을 누른 시간에 해당하는 Length(0~240, 1당 125ms)가 들어가며, Data[2]에는 리모컨 버튼의 키 Data가 들어갑니다. DRC가 리모컨 값을 받으면 Channel과 r(Remocon Channel)을 비교한 후, 일치한다면 약 250ms 동안 r(Remocon Length)값과 r(Remocon Data)값을 각각 Length, Data로 바꿉니다. 250ms 후에는 r(Remocon Length)값과 r(Remocon Data)값이 다시 각각 0과 254로 돌아가므로, REMOCON 요청 패킷을 사용할 때에는 125ms의 주기로 Length 값을 1씩 증가시키면서 보내는 것을 권장합니다.

예시

■ 모든 DRC(Broadcasting)에 0x61의 채널로 1초 동안 버튼 0x21이 눌렸음을 알리는 요청 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]	Data[2]
Value	0xFF	0xFF	0x0A(10)	0xFE	0x1D	0xA0	0x5E	0x61	0x08	0x21

모든 DRC를 향해서 보내는 것이므로 pID는 0xFE이며, 채널이 0x61이므로 Data[0]에는 0x61이 들어갑니다. Length 값은 1당 125ms이므로, 1초는 8에 해당됩니다. 따라서 Data[1]에는 8의 값이 들어가며, Data[2]에는 리모컨 키 값인 0x21이 들어갑니다.

13-2. REMOCON – 응답 패킷(0x1D)

REMOCON 명령은 응답 패킷이 없습니다.

명령 상세 설명 – SERVO_FW_UPDATE

14-1. SERVO_FW_UPDATE – 요청 패킷(0x1E)

구조

구분	Packet Size	pID	CMD
Value	7	0~0xFE	0x1Z

DRC에 연결된 서보의 펌웨어(Firmware)를 업데이트해야 할 때 사용하는 패킷입니다. 서보의 펌웨어를 업데이트할 때는 평소에 사용하는 프로토콜과 다른 것을 사용하기 때문에, SERVO_FW_UPDATE 요청 패킷을 보내 특수한 모드로 넘어가야 합니다. 이 모드에서는 PC와 제어기 간의 통신은 일체 할 수 없고, 오로지 PC와 서보가 직접적으로 연결된 것처럼 동작합니다.

예시

■ r(ID)가 253인 DRC를 서보 펌웨어 업데이트 모드로 바꾸는 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2
Value	0xFF	0xFF	0x09(9)	0xFD	0x1E	0xE4	0x1A

14-2. SERVO_FW_UPDATE – 응답 패킷(0x5E)

구조

구분	Packet Size		pID	CMD	Data[0]	Data[1]
Value	7+2	r(ID)	0x5E	r(Status Error)	r(Error Codes[0])	0xA0

r(Status Error)와 r(Error Codes[0])의 값을 응답 패킷에 실어서 반환합니다. Optional Data의 크기는 2바이트로 고정 되어 있기 때문에 Packet Size는 9바이트로 고정됩니다. pID는 현재 응답하는 DRC의 r(ID)가 실리게 되며, CMD는 요청 패킷 CMD 0x1E에 Bitwise OR 연산한 0x5E이 됩니다.

응답 조건

SERVO_FW_UPDATE은 r(ACK Policy)가 2(모든 패킷에 대하여 응답)일 경우에만 응답합니다. 단, 요청 패킷의 pID가 254(Broadcasting ID)일 경우에는 r(ACK Policy)가 2라도 응답하지 않습니다.

예시

■ r(ID)가 253인 DRC를 서보 펌웨어 업데이트 모드로 바꾸는 요청 패킷에 대한 응답 패킷

구분	Header		Packet Size	pID	CMD	CS1	CS2	Data[0]	Data[1]
Value	0xFF	0xFF	0x09(9)	0xFD	0x5E	0xAA	0x54	0x00	0x00

현재 상태와 가장 최근의 에러 코드를 실어서 보냅니다. 에러가 없는 상태이므로 둘 다 0x00입니다.

ADC Lookup Table – Voltage

ADC		VIN	ADC		VIN	ADC		VIN	ADC		VIN
Decimal	HEX		Decimal	HEX		Decimal	HEX		Decimal	HEX	
0	0	0.000	64	40	4.722	128	80	9.444	192	C0	14.167
1	1	0.074	65	41	4.796	129	81	9.518	193	C1	14.240
2	2	0.148	66	42	4.870	130	82	9.592	194	C2	14.314
3	3	0.221	67	43	4.944	131	83	9.666	195	C3	14.388
4	4	0.295	68	44	5.017	132	84	9.740	196	C4	14.462
5	5	0.369	69	45	5.091	133	85	9.813	197	C5	14.536
6	6	0.443	70	46	5.165	134	86	9.887	198	C6	14.609
7	7	0.516	71	47	5.239	135	87	9.961	199	C7	14.683
8	8	0.590	72	48	5.313	136	88	10.035	200	C8	14.757
9	9	0.664	73	49	5.386	137	89	10.109	201	C9	14.831
10	A	0.738	74	4A	5.460	138	8A	10.182	202	CA	14.905
11	B	0.812	75	4B	5.534	139	8B	10.256	203	CB	14.978
12	C	0.885	76	4C	5.608	140	8C	10.330	204	CC	15.052
13	D	0.959	77	4D	5.681	141	8D	10.404	205	CD	15.126
14	E	1.033	78	4E	5.755	142	8E	10.477	206	CE	15.200
15	F	1.107	79	4F	5.829	143	8F	10.551	207	CF	15.273
16	10	1.181	80	50	5.903	144	90	10.625	208	D0	15.347
17	11	1.254	81	51	5.977	145	91	10.699	209	D1	15.421
18	12	1.328	82	52	6.050	146	92	10.773	210	D2	15.495
19	13	1.402	83	53	6.124	147	93	10.846	211	D3	15.569
20	14	1.476	84	54	6.198	148	94	10.920	212	D4	15.642
21	15	1.549	85	55	6.272	149	95	10.994	213	D5	15.716
22	16	1.623	86	56	6.345	150	96	11.068	214	D6	15.790
23	17	1.697	87	57	6.419	151	97	11.141	215	D7	15.864
24	18	1.771	88	58	6.493	152	98	11.215	216	D8	15.938
25	19	1.845	89	59	6.567	153	99	11.289	217	D9	16.011
26	1A	1.918	90	5A	6.641	154	9A	11.363	218	DA	16.085
27	1B	1.992	91	5B	6.714	155	9B	11.437	219	DB	16.159
28	1C	2.066	92	5C	6.788	156	9C	11.510	220	DC	16.233
29	1D	2.140	93	5D	6.862	157	9D	11.584	221	DD	16.306
30	1E	2.214	94	5E	6.936	158	9E	11.658	222	DE	16.380
31	1F	2.287	95	5F	7.010	159	9F	11.732	223	DF	16.454
32	20	2.361	96	60	7.083	160	A0	11.806	224	E0	16.528
33	21	2.435	97	61	7.157	161	A1	11.879	225	E1	16.602
34	22	2.509	98	62	7.231	162	A2	11.953	226	E2	16.675
35	23	2.582	99	63	7.305	163	A3	12.027	227	E3	16.749
36	24	2.656	100	64	7.378	164	A4	12.101	228	E4	16.823
37	25	2.730	101	65	7.452	165	A5	12.174	229	E5	16.897
38	26	2.804	102	66	7.526	166	A6	12.248	230	E6	16.970
39	27	2.878	103	67	7.600	167	A7	12.322	231	E7	17.044
40	28	2.951	104	68	7.674	168	A8	12.396	232	E8	17.118
41	29	3.025	105	69	7.747	169	A9	12.470	233	E9	17.192
42	2A	3.099	106	6A	7.821	170	AA	12.543	234	EA	17.266
43	2B	3.173	107	6B	7.895	171	AB	12.617	235	EB	17.339
44	2C	3.247	108	6C	7.969	172	AC	12.691	236	EC	17.413
45	2D	3.320	109	6D	8.043	173	AD	12.765	237	ED	17.487
46	2E	3.394	110	6E	8.116	174	AE	12.839	238	EE	17.561
47	2F	3.468	111	6F	8.190	175	AF	12.912	239	EF	17.635
48	30	3.542	112	70	8.264	176	B0	12.986	240	F0	17.708
49	31	3.615	113	71	8.338	177	B1	13.060	241	F1	17.782
50	32	3.689	114	72	8.411	178	B2	13.134	242	F2	17.856
51	33	3.763	115	73	8.485	179	B3	13.207	243	F3	17.930
52	34	3.837	116	74	8.559	180	B4	13.281	244	F4	18.003
53	35	3.911	117	75	8.633	181	B5	13.355	245	F5	18.077
54	36	3.984	118	76	8.707	182	B6	13.429	246	F6	18.151
55	37	4.058	119	77	8.780	183	B7	13.503	247	F7	18.225
56	38	4.132	120	78	8.854	184	B8	13.576	248	F8	18.299
57	39	4.206	121	79	8.928	185	B9	13.650	249	F9	18.372
58	3A	4.280	122	7A	9.002	186	BA	13.724	250	FA	18.446
59	3B	4.353	123	7B	9.076	187	BB	13.798	251	FB	18.520
60	3C	4.427	124	7C	9.149	188	BC	13.872	252	FC	18.594
61	3D	4.501	125	7D	9.223	189	BD	13.945	253	FD	18.668
62	3E	4.575	126	7E	9.297	190	BE	14.019	254	FE	18.741
63	3F	4.648	127	7F	9.371	191	BF	14.093	255	FF	18.815

ADC Lookup Table – Temperature

ADC		VIN	ADC		VIN	ADC		VIN	ADC		VIN
Decimal	HEX		Decimal	HEX		Decimal	HEX		Decimal	HEX	
0	0	−80.57	64	40	−1.34	128	80	25.00	192	C0	56.99
1	1	−72.89	65	41	−0.89	129	81	25.41	193	C1	57.67
2	2	−64.26	66	42	−0.44	130	82	25.82	194	C2	58.36
3	3	−58.84	67	43	0.01	131	83	26.24	195	C3	59.05
4	4	−54.80	68	44	0.46	132	84	26.65	196	C4	59.76
5	5	−51.55	69	45	0.90	133	85	27.07	197	C5	60.48
6	6	−48.81	70	46	1.34	134	86	27.49	198	C6	61.21
7	7	−46.43	71	47	1.78	135	87	27.91	199	C7	61.96
8	8	−44.32	72	48	2.21	136	88	28.33	200	C8	62.71
9	9	−42.41	73	49	2.64	137	89	28.75	201	C9	63.48
10	A	−40.68	74	4A	3.07	138	8A	29.18	202	CA	64.27
11	B	−39.08	75	4B	3.50	139	8B	29.60	203	CB	65.06
12	C	−37.59	76	4C	3.93	140	8C	30.03	204	CC	65.88
13	D	−36.20	77	4D	4.35	141	8D	30.46	205	CD	66.71
14	E	−34.89	78	4E	4.77	142	8E	30.89	206	CE	67.55
15	F	−33.66	79	4F	5.19	143	8F	31.32	207	CF	68.41
16	10	−32.49	80	50	5.61	144	90	31.76	208	D0	69.29
17	11	−31.37	81	51	6.03	145	91	32.20	209	D1	70.19
18	12	−30.31	82	52	6.45	146	92	32.64	210	D2	71.11
19	13	−29.29	83	53	6.86	147	93	33.08	211	D3	72.05
20	14	−28.31	84	54	7.27	148	94	33.52	212	D4	73.01
21	15	−27.36	85	55	7.68	149	95	33.97	213	D5	74.00
22	16	−26.45	86	56	8.09	150	96	34.42	214	D6	75.01
23	17	−25.57	87	57	8.50	151	97	34.87	215	D7	76.04
24	18	−24.72	88	58	8.91	152	98	35.33	216	D8	77.10
25	19	−23.89	89	59	9.32	153	99	35.78	217	D9	78.19
26	1A	−23.09	90	5A	9.72	154	9A	36.24	218	DA	79.31
27	1B	−22.31	91	5B	10.13	155	9B	36.71	219	DB	80.46
28	1C	−21.54	92	5C	10.53	156	9C	37.17	220	DC	81.65
29	1D	−20.80	93	5D	10.94	157	9D	37.64	221	DD	82.87
30	1E	−20.08	94	5E	11.34	158	9E	38.11	222	DE	84.13
31	1F	−19.37	95	5F	11.74	159	9F	38.59	223	DF	85.44
32	20	−18.68	96	60	12.14	160	A0	39.07	224	E0	86.78
33	21	−18.00	97	61	12.55	161	A1	39.55	225	E1	88.17
34	22	−17.34	98	62	12.95	162	A2	40.04	226	E2	89.62
35	23	−16.69	99	63	13.35	163	A3	40.53	227	E3	91.12
36	24	−16.05	100	64	13.75	164	A4	41.02	228	E4	92.67
37	25	−15.42	101	65	14.15	165	A5	41.52	229	E5	94.29
38	26	−14.81	102	66	14.54	166	A6	42.02	230	E6	95.98
39	27	−14.20	103	67	14.94	167	A7	42.52	231	E7	97.75
40	28	−13.61	104	68	15.34	168	A8	43.03	232	E8	99.59
41	29	−13.02	105	69	15.74	169	A9	43.55	233	E9	101.53
42	2A	−12.45	106	6A	16.14	170	AA	44.07	234	EA	103.57
43	2B	−11.88	107	6B	16.54	171	AB	44.59	235	EB	105.71
44	2C	−11.32	108	6C	16.94	172	AC	45.12	236	EC	107.98
45	2D	−10.76	109	6D	17.34	173	AD	45.65	237	ED	110.38
46	2E	−10.22	110	6E	17.74	174	AE	46.19	238	EE	112.93
47	2F	−9.68	111	6F	18.13	175	AF	46.74	239	EF	115.65
48	30	−9.15	112	70	18.53	176	B0	47.29	240	F0	118.57
49	31	−8.62	113	71	18.93	177	B1	47.84	241	F1	121.72
50	32	−8.10	114	72	19.33	178	B2	48.40	242	F2	125.12
51	33	−7.59	115	73	19.73	179	B3	48.97	243	F3	128.83
52	34	−7.08	116	74	20.13	180	B4	49.54	244	F4	132.89
53	35	−6.58	117	75	20.54	181	B5	50.12	245	F5	137.38
54	36	−6.08	118	76	20.94	182	B6	50.71	246	F6	142.40
55	37	−5.59	119	77	21.34	183	B7	51.30	247	F7	148.06
56	38	−5.10	120	78	21.74	184	B8	51.90	248	F8	154.56
57	39	−4.62	121	79	22.15	185	B9	52.51	249	F9	162.13
58	3A	−4.14	122	7A	22.55	186	BA	53.13	250	FA	171.18
59	3B	−3.66	123	7B	22.96	187	BB	53.75	251	FB	182.34
60	3C	−3.19	124	7C	23.36	188	BC	54.38	252	FC	196.72
61	3D	−2.72	125	7D	23.77	189	BD	55.02	253	FD	216.58
62	3E	−2.26	126	7E	24.18	190	BE	55.67	254	FE	247.46
63	3F	−1.80	127	7F	24.59	191	BF	56.33	255	FF	310.08
63	3F	4.648	127	7F	9.371	191	BF	14.093	255	FF	18.815

Error Code 상세 명세

Status Error Flag	Error Code	상세
Exceed Input Voltage limit	0x01	전압이 너무 낮음
	0x02	전압이 너무 높음
Exceed Temperature limit	0x03	온도가 너무 높음
Servo Missing	0x11	상태 자동 확인에 의해 서보 레지스터 읽던 중 서보 응답 없음
	0x12	Task 실행에 의해 서보 레지스터 읽던 중 서보 응답 없음
EEP REG distorted	0x21	EEPROM의 모델명이 잘못 됨
	0x22	EEPROM의 ID가 잘못 됨
	0x23	EEPROM 데이터가 손상됨
Servo Status Error	0x31	서보 상태 에러
	0x33	너무 많은 서보가 제어기에 연결됨
Invalid Packet	0x41	Zigbee Ack가 온전히 오지 않거나 Noise가 들어옴
	0x42	Zigbee Ack에서 Check Sum Error
	0x43	Zigbee Ack에서 Unknown Command
	0x44	Zigbee Ack가 들어왔으나 ID가 0xFC가 아님
	0x45	Zigbee Ack에서 사이즈가 너무 큰 패킷이 들어옴
	0x46	Zigbee Ack에서 명령과 안 맞는 사이즈의 패킷이 들어옴
	0x47	Zigbee Ack가 오지 않음
	0x51	Servo Ack가 온전히 오지 않거나 Noise가 들어옴
	0x52	Servo Ack에서 Check Sum Error
	0x53	Servo Ack에서 Unknown Command
	0x54	Servo Ack에서 Invalid ID의 패킷이 들어옴
	0x56	Servo Ack에서 사이즈가 너무 큰 패킷이 들어옴
	0x57	Servo Ack에서 명령과 안 맞는 사이즈의 패킷이 들어옴
	0x58	Servo Ack를 받는 UART 버퍼가 꽉참
	0x59	Servo에 보낼 패킷을 저장하는 버퍼가 꽉참
	0x61	PC측 패킷이 온전히 오지 않거나 Noise가 들어옴
	0x62	PC측 패킷에서 Check Sum Error
	0x63	PC측 패킷에서 Unknown Command
	0x64	PC측 패킷에서 Invalid ID의 패킷이 들어옴
	0x66	PC측 패킷에서 사이즈가 너무 큰 패킷이 들어옴
	0x67	PC측 패킷에서 명령에 안 맞는 사이즈의 패킷이 들어옴
	0x68	PC측 패킷을 받는 UART 버퍼가 꽉참
	0x71	EEP/RAM의 WRITE/READ 명령이 레지스터 범위를 벗어남
	0x72	RAM_WRITE에서 잘못된 값을 씀
	0x73	RAM_WRITE에서 Status에 잘못된 값을 씀
	0x74	CON_CHECK에서 패킷에 잘못된 ID가 들어가 있음
	0x75	PLAY_MOTION에서 잘못된 모션 번호가 들어가 있음
	0x76	PLAY_TASK에서 잘못된 Instruction이 들어가 있음
	0x77	REMOCON에서 잘못된 Channel이나 Length가 들어가 있음
	0x78	ZIGBEE에서 잘못된 Instruction이 들어가 있음
	0x79	PLAY_BUZZ에서 잘못된 버저 번호가 들어가 있음

Error Code 상세 명세

Status Error Flag	Error Code	상세
Flash Data Distorted	0x81	없는 Motion을 실행하라고 함
	0x82	Motion 데이터에 문제 있음
	0x83	Motion 데이터의 축 수와 현재 축 수가 다름
	0x84	다음 프레임까지 시간이 음수인 프레임을 만남
	0x85	너무 많은 Repeat 명령이 중첩되었음(최대 3개까지 가능)
	0x91	Task 데이터에 문제 있음
	0x92	계산 식 처리 중 이상
	0x93	프로그램 스택 오버플로우
	0x94	MPSU RAM에 Load 중 잘못된 레지스터 주소
	0x95	MPSU RAM에 Load 중 잘못된 레지스터 길이
	0x96	Servo RAM에 Load 중 잘못된 레지스터 주소
	0x97	Servo RAM에 Load 중 잘못된 레지스터 길이
	0x98	Servo RAM에 Load 중 잘못된 ID
	0x99	MPSU RAM에서 Read 중 잘못된 레지스터 길이
	0x9A	Servo RAM에서 Read 중 잘못된 레지스터 길이
	0x9B	Servo RAM에서 Read 중 잘못된 ID
	0xA1	Motion 명령에 범위 벗어난 값
	0xA2	Motion Ready 명령에 범위 벗어난 값
	0xA3	서보 제어 명령에 범위 벗어난 값
	0xA5	제어기 LED 제어 명령에 범위 벗어난 값
	0xA6	버저 멜로디 명령에 범위 벗어난 값
	0xA7	버저 음표 명령에 범위 벗어난 값
	0xB2	없는 Buzzer를 실행하라고 함

Dongbu Robot

www.hovis.co.kr/guide

www.dongburobot.com → 고객지원 → 서비스용 → 자료실

www.dongburobot.com → QuickMenu → 서비스용 → 자료실

Tel. 032-329-5551
Fax. 032-329-5569
Homepage : www.dongburobot.com