

Marvelmind 장치와의 데이터 교환을 위한 하드웨어 인터페이스 및 프로토콜

버전 2026.02.24
펌웨어 v7.000 이상에 적용

www.marvelmind.com

목차

1.	Connection to Marvelmind devices.....	4
1.1	UART and other interfaces for Super-Beacon.....	5
1.2	UART and other interfaces for beacon Mini-RX.....	6
1.3	UART and other interfaces for beacon Mini-TX-2.....	7
1.4	UART and other interfaces for Modem HW v5.1.....	8
1.5	UART and other interfaces for Super-Modem.....	9
1.6	UART and other interfaces for Modem HW v4.9.....	10
1.7	UART and other interfaces for Industrial-TX, Industrial-RX, Industrial Super-Beacon...	11
1.8	UART and SPI interfaces for beacon HW v4.9.....	12
1.9	UART and SPI interfaces for beacon HW v4.5.....	13
2.	Protocols of communication via UART.....	14
2.1	'Marvelmind' protocol for streaming.....	14
2.2	Protocol of reading/writing data from/to user device.....	41
2.3	NMEA0183 communication protocol.....	45
3.	Protocols of communication via USB (virtual UART).....	54
3.1	'Marvelmind' protocol for streaming.....	54
3.2	Protocol of reading/writing data from/to user device.....	55
3.3	NMEA0183 communication protocol.....	56
3.4	Protocol of data exchange with modem via USB interface.....	57
4.	Protocols of communication via RS-485.....	84
4.1	'Marvelmind' protocol for streaming.....	84
4.2	Protocol of reading/writing data from/to user device.....	85
4.3	NMEA0183 communication protocol.....	86
5.	Protocols of communication via SPI.....	87
5.1	Packet with hedgehog location.....	87
5.2	Other data via SPI.....	88
6.	Protocols of communication via I ² C.....	89
6.1	Compass emulation for drones with PX4.....	89
6.2	Other data via I ² C.....	90
7.	Protocols of communication via UDP (Wi-Fi).....	91
7.1	Packet with hedgehog location.....	92
7.1.1.	Packet with hedgehog location with real-time timestamps (firmware v7.200+).....	94
7.2.	Packet with stationary beacons locations.....	95
7.3.	Packet with raw IMU data.....	96
7.3.1.	Packet with raw IMU data with real-time timestamps (firmware v7.200+).....	97

7.4.	Packet with raw distances data.....	98
7.4.1.	Packet with raw distances data with real-time timestamps (firmware v7.200+).....	99
7.5.	Packet with IMU fusion data.....	100
7.5.1.	Packet with IMU fusion data with real-time timestamps (firmware v7.200+).....	101
7.6.	Packet with telemetry data.....	102
7.7.	Packet with quality and extended location data.....	103
7.8.	Packet with telemetry of all beacons.....	104
7.9.	NMEA0183 protocol.....	105
8.	Protocols of communication via CAN.....	106
8.1.	'Marvelmind' protocol of streaming.....	107
8.2.	NMEA0183 communication protocol.....	108
9.	Format of dashboard csv log file.....	109
9.1.	Format of csv log file (dashboard version V7.000+).....	110
9.2.	Previous format of csv log (dashboard before V7.000 or modem HW v4.9).....	124
10.	Marvelmind API.....	125
10.1.	Installation for Windows.....	126
10.2.	Installation for Linux.....	127
10.3.	Check connection to API.....	128
10.4.	Marvelmind API library description.....	129
10.5.	Description of C example for Marvelmind API.....	187
10.6.	Device types.....	195
11.	Sending user data from/to user devices.....	196
12.	Contacts.....	198
	Appendix 1. Calculating CRC-16.....	199
	Appendix 2. Format of error reply from modem.....	200

1. Marvelmind 장치에 연결

Marvelmind 장치(Modem 또는 이동형 비콘(Hedgehog))와 통신하려면 다음 인터페이스 중 하나를 통해 외부 장치(로봇, 드론, AGV 등)에 연결해야 합니다:

1. CDC 클래스 USB 장치(Windows에서는 가상 COM 포트, Linux에서는 ttyACM 또는 ttyUSB)로 USB 호스트에 연결합니다. Windows에서는 Modem과 동일한 드라이버가 필요합니다. Linux에서는 필요한 드라이버가 Linux 커널에 통합되어 있으므로 별도 드라이버가 필요하지 않습니다. 이 인터페이스에서는 실제 RS-232를 사용하지 않으므로, 호스트에서 열리는 시리얼 포트의 파라미터(보드레이트, 비트 수, 패리티 등)는 임의의 값으로 설정할 수 있습니다.
2. UART에 연결합니다 — 스트리밍용은 핀에 2선 납땜, 양방향 통신용은 3선 납땜이 필요합니다. 아래 하드웨어 인터페이스 그림을 참조하십시오. UART 송신기의 논리 레벨은 CMOS 3.3V입니다. 기본 보드레이트는 500 kbps이며, Dashboard에서 다음 목록 중에서 설정할 수 있습니다: 4.8, 9.6, 19.2, 38.4, 57.6, 115.2, 500 kbps. 데이터 형식: 8 bit, 패리티 없음, 정지 비트 1.
3. SPI에 연결합니다. Marvelmind 장치는 SPI 슬레이브 장치로 동작합니다. SPI 파라미터: SPI 모드 0, 각 바이트 내 MSB가 먼저 전송됩니다. 연결은 SCK 속도 최대 8 MHz까지 테스트되었습니다. 고속(500 kHz 초과)에서는 배선 연결 품질에 주의하십시오.
4. RS-485에 연결합니다(Super-Modem 또는 Industrial Super-Beacon 전용).
5. I2C에 연결합니다(Super-Beacon 전용).
6. Wi-Fi를 통한 UDP에 연결합니다(Super-Modem의 경우) 또는 임의의 네트워크 연결(Dashboard의 경우).
7. CAN에 연결합니다(Industrial Super-Beacon의 경우, Super-Modem은 요청 시 제공).

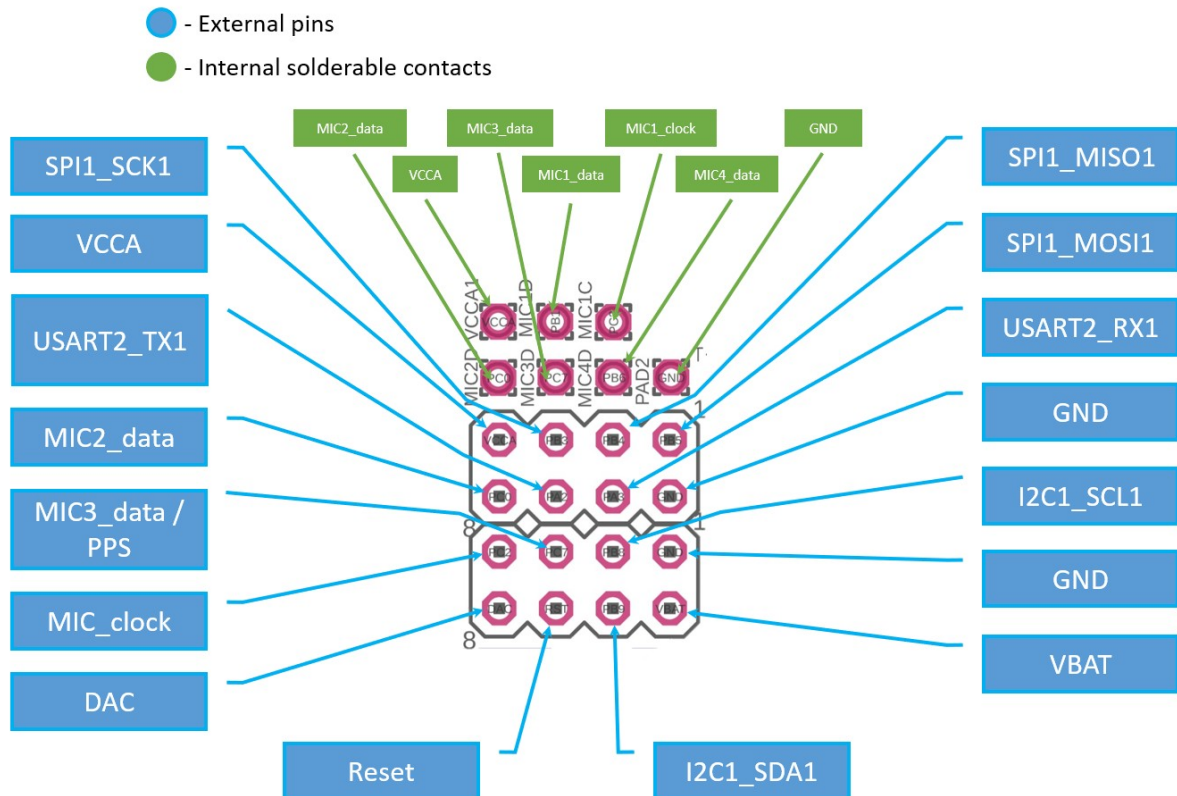
연결 설정 요약:

인터페이스	비트레이트	기타 설정
USB (가상 UART)	UART 비트레이트는 적용되지 않으며, 시리얼 포트 속도는 임의의 값으로 설정할 수 있습니다. USB 풀 스피드(12 Mbit/s)를 통한 데이터 전송	UART 설정 해당 없음
UART	4.8, 9.6, 19.2, 38.4, 57.6, 115.2, 500 Kbit/s Dashboard에서 선택 가능	데이터 비트 8비트, 정지 비트 1비트, 패리티 없음
SPI	8 Mbit/s까지 테스트 완료	SPI 모드 0
RS-485	4.8, 9.6, 19.2, 38.4, 57.6, 115.2, 500 Kbit/s Dashboard에서 선택 가능(UART와 동일)	데이터 비트 8비트, 정지 비트 1비트, 패리티 없음
I2C	최대 400 Kbit/s	
UDP	네트워크 연결 속도에 따름	

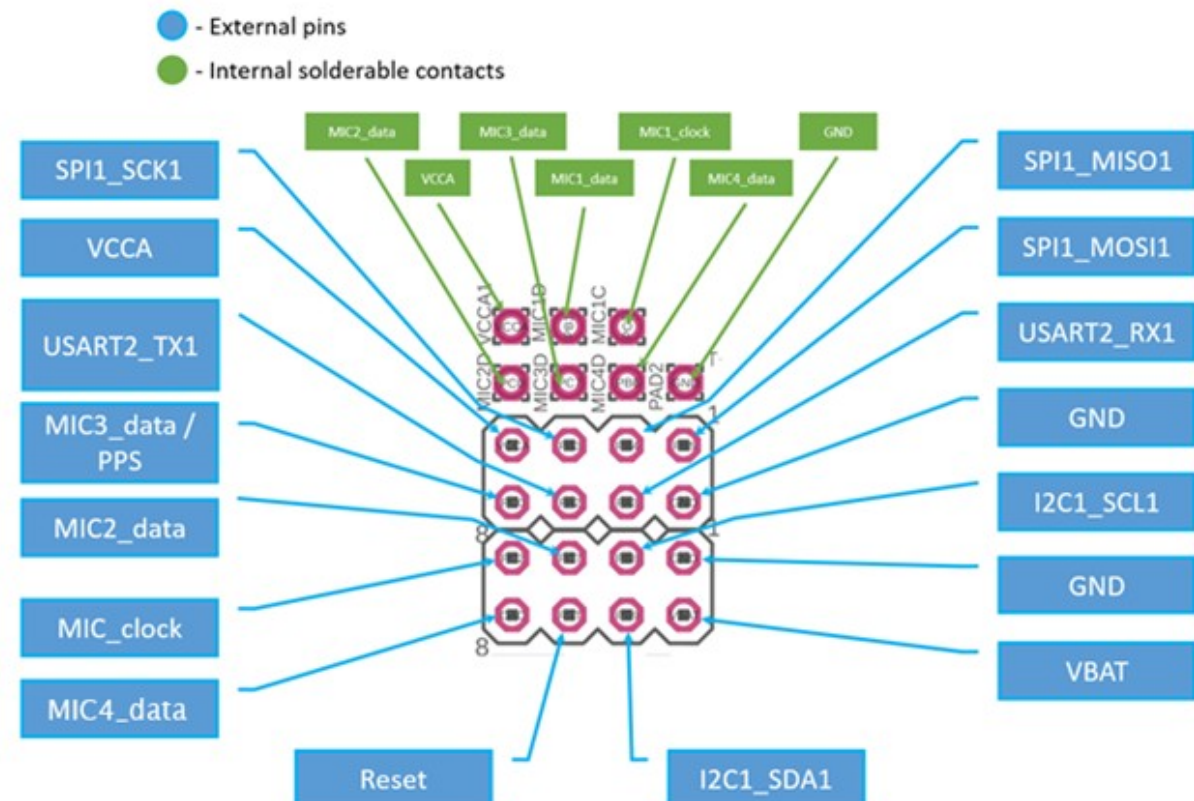
CAN	125 Kbit/s	표준 프레임
-----	------------	--------

1.1 Super-Beacon의 UART 및 기타 인터페이스

Super-Beacon의 4x4 핀아웃:



Super-Beacon-2 및 Super-Beacon-3의 4x4 핀아웃:



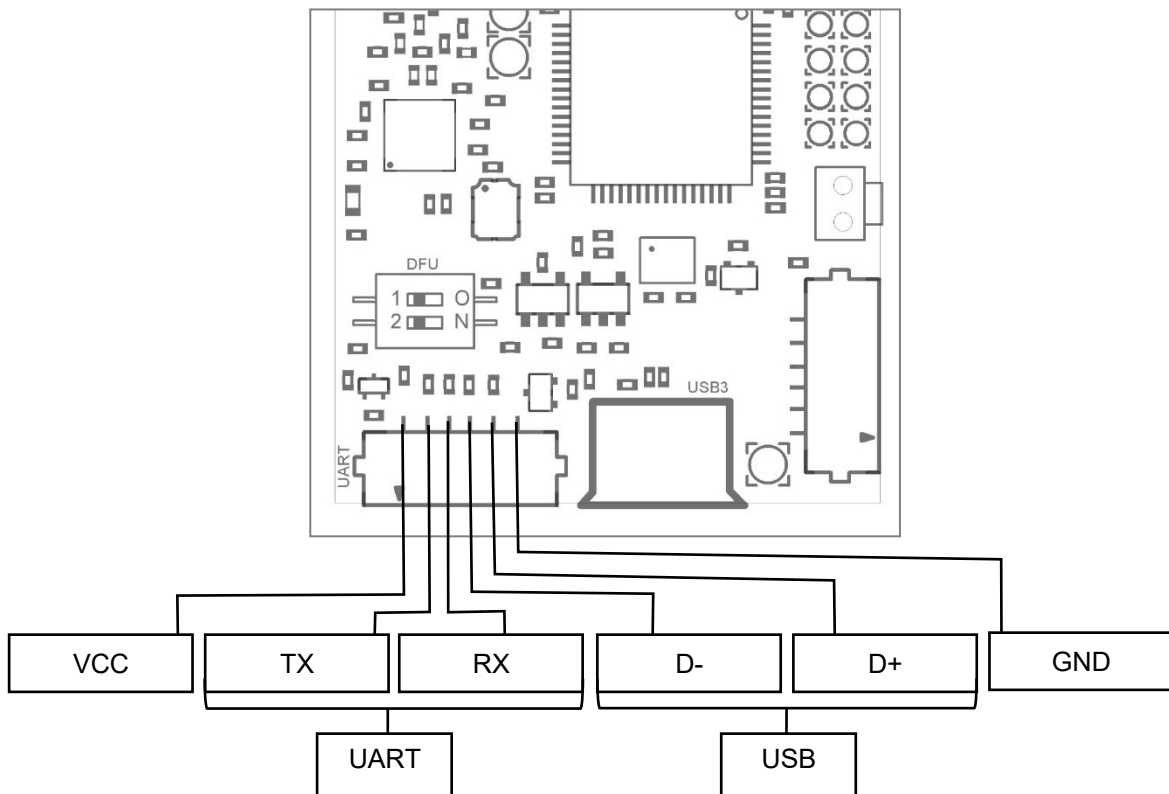
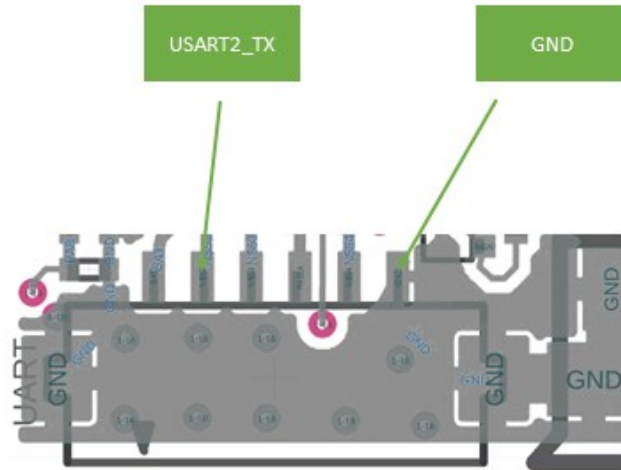
1.2 비콘 Mini-RX의 UART 및 기타 인터페이스



올바르게 납땜할 수 있다고 확신하는 경우에만 사용하십시오. DIP 스위치로 비콘의 전원을 끄는 것을 잊지 마십시오. 납땜이 잘못되어 비콘이 손상된 경우, Marvelmind 팀은 이에 대한 책임을 지지 않습니다.

비콘 Mini-RX에서 UART 데이터 스트리밍을 수신하려면 보드의 핀에 납땜해야 합니다.

● - Internal solderable contacts

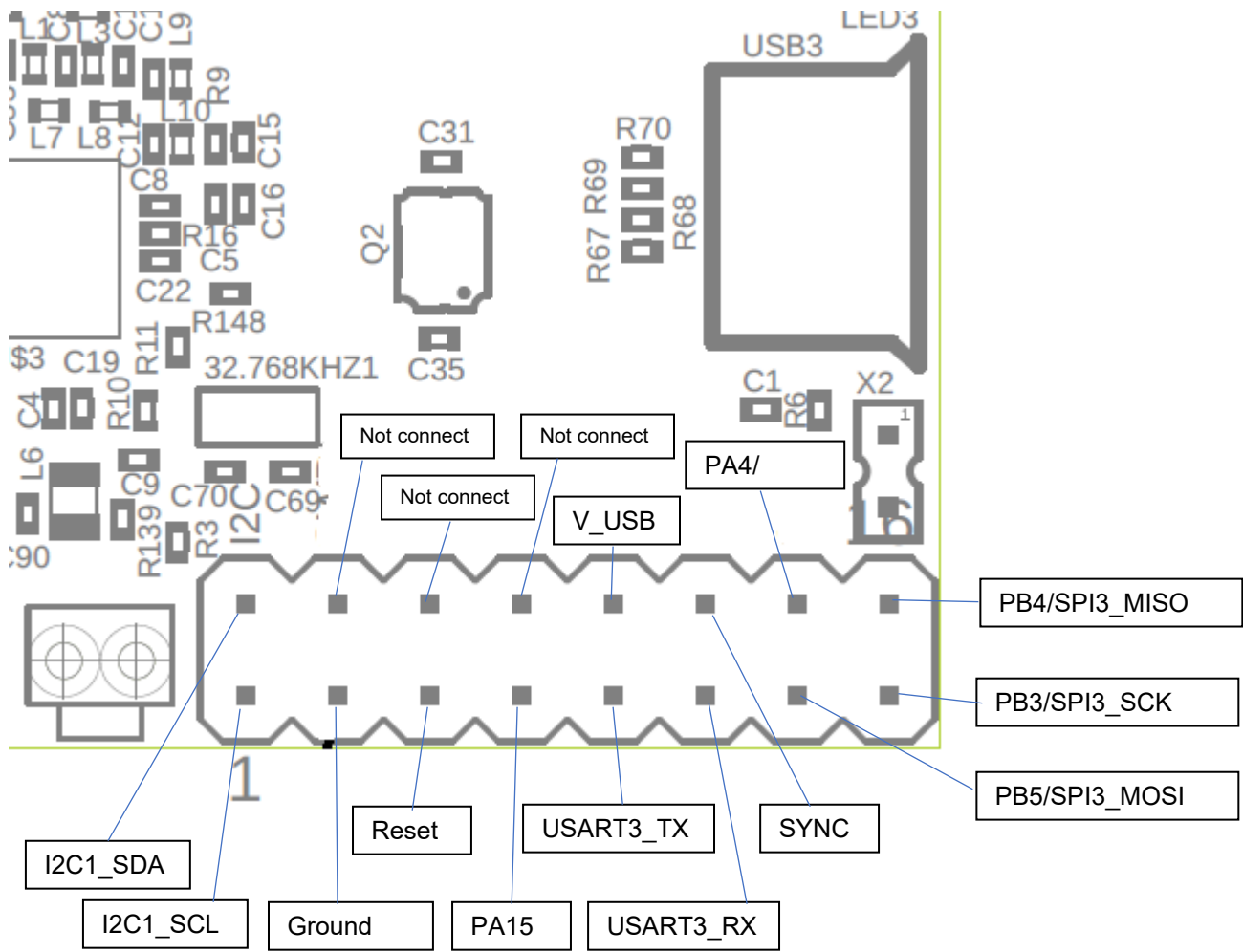


1.3. 비콘 Mini-TX-2의 UART 및 기타 인터페이스

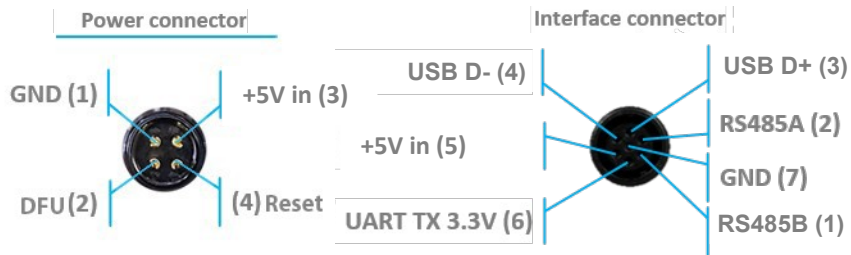
비콘 Mini-TX-2는 비콘 Mini-RX와 동일한 핀아웃을 가진 커넥터를 갖추고 있습니다.

[UART 케이블은 Mini-TX-2 연결에 사용할 수 있습니다.](#)

1.4. Modem HW v5.1의 UART 및 기타 인터페이스



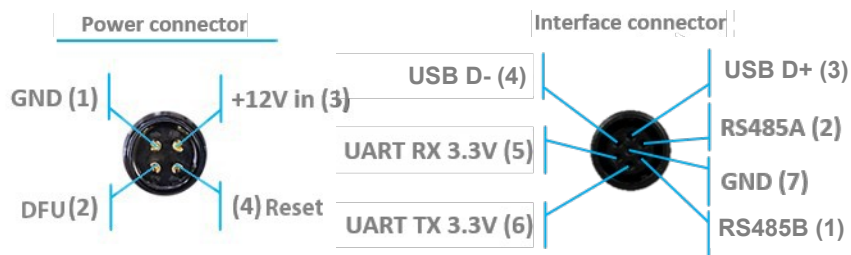
1.5. Super-Modem의 UART 및 기타 인터페이스



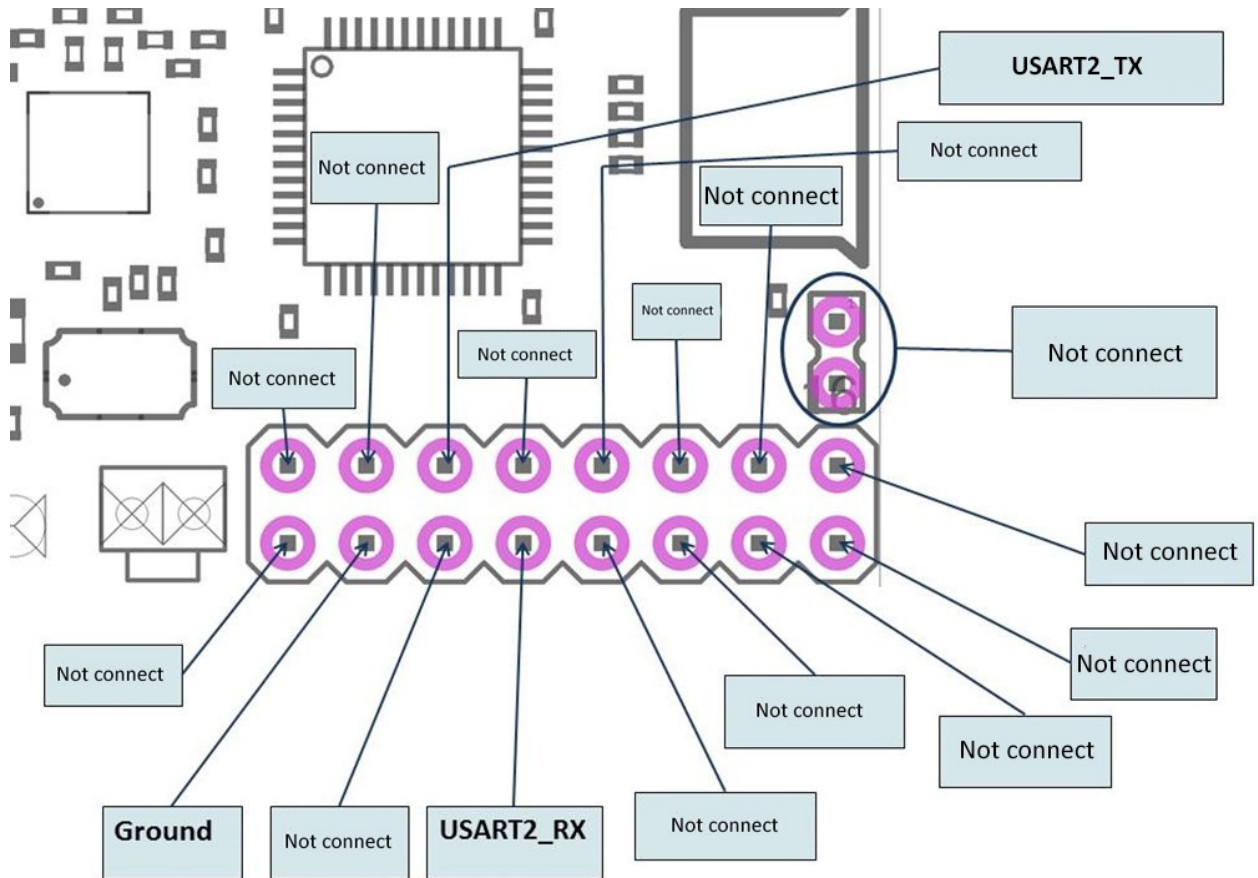
또한 Super-Modem에는 온보드 Wi-Fi 인터페이스가 포함되어 있습니다. Wi-Fi 연결 구성은 UDP 챕터에 설명되어 있습니다.



- 새 버전의 Super-Modem (2023년 6월부터)은 +5V 전원 공급만 지원합니다. 이 버전에는 +12V 전원 공급 컨버터를 사용하지 마십시오. 비콘이 손상될 수 있습니다!



1.6. Modem HW v4.9의 UART 및 기타 인터페이스

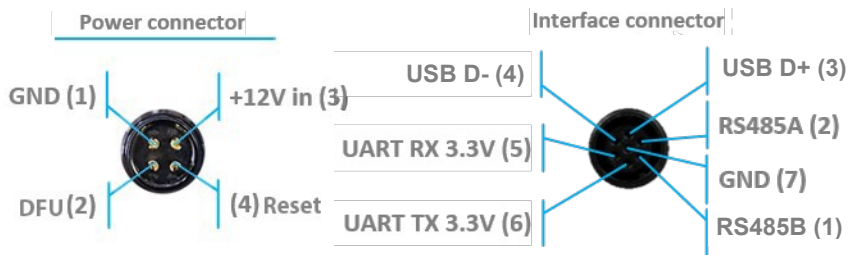


1.7. Industrial-TX, Industrial-RX, Industrial Super-Beacon의 UART 및 기타 인터페이스

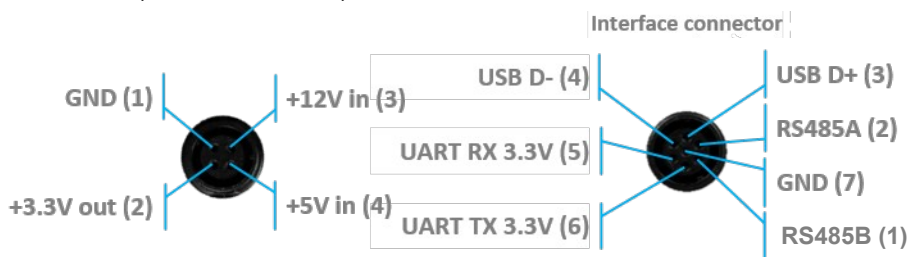


- Industrial-TX, Industrial-RX, Industrial Super-Beacon의 버전 2 및 3 (2022년 6월부터)은 +5V 전원 공급만 지원합니다. 이 버전에는 +12V 전원 공급 컨버터를 사용하지 마십시오. 비콘이 손상될 수 있습니다!

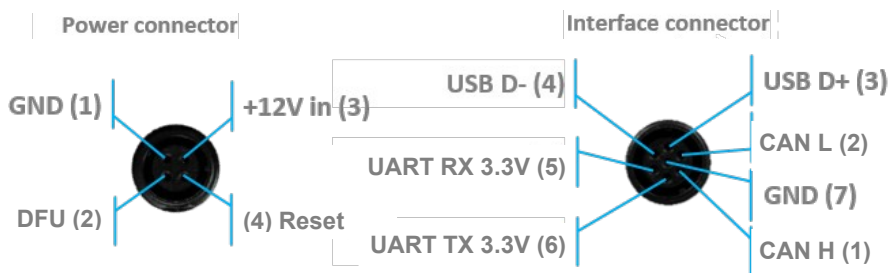
- 이 버전에는 UART RX가 없지만, 이제 Interface 커넥터를 전원 공급 장치로 사용할 수 있습니다.



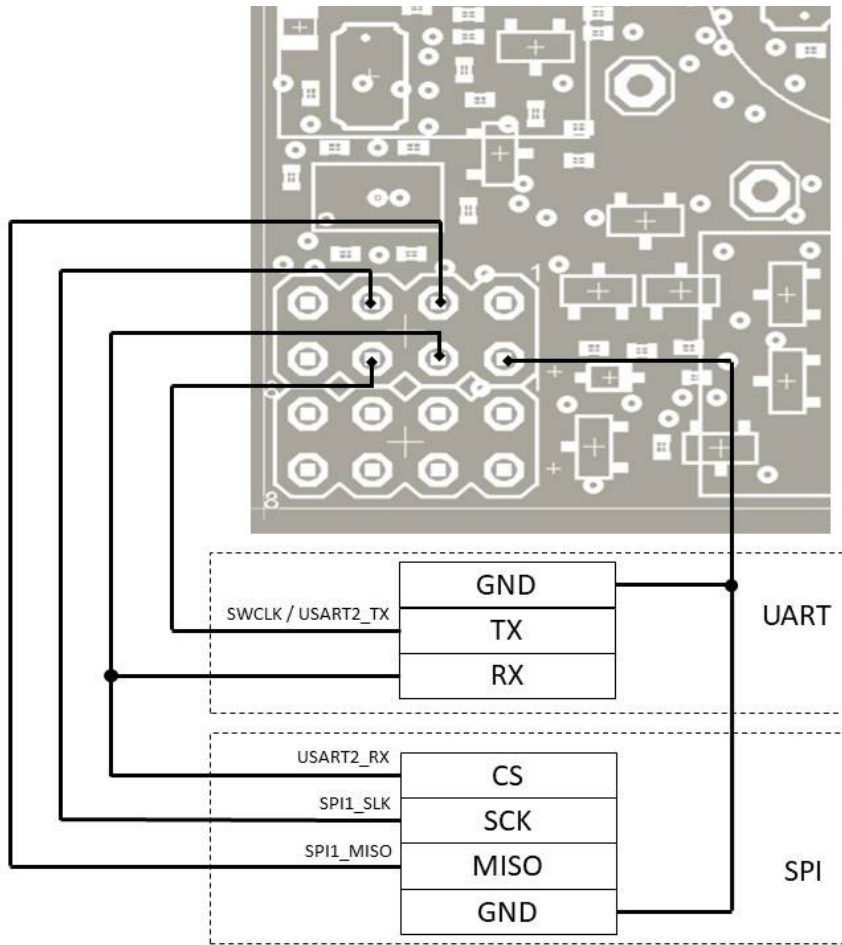
RS485 수정 핀아웃 (2019년 9월 이전)



CAN 수정 핀아웃

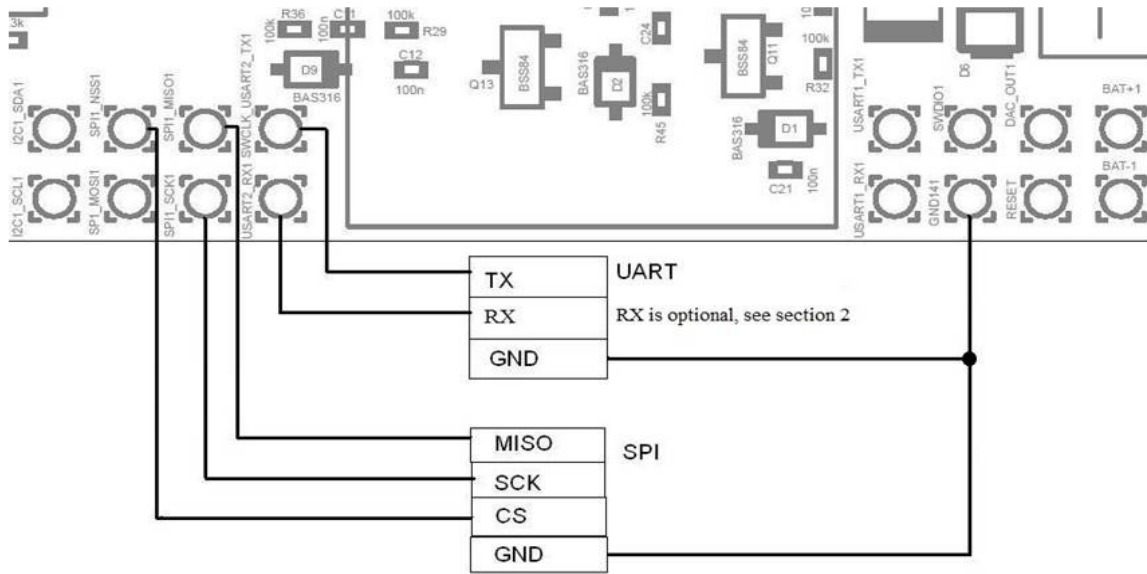


1.8. 비콘 HW v4.9용 UART 및 SPI 인터페이스



참고: 보시다시피 UART RX와 SPI CS는 동일한 공유 핀을 사용합니다. 이 핀의 기능(UART 수신기, SPI 칩 선택 또는 기타)은 Dashboard의 '인터페이스' 섹션에 있는 'PA15 pin function' 매개변수를 통해 선택할 수 있습니다.

1.9. 비콘 HW v4.5용 UART 및 SPI 인터페이스



2. UART를 통한 통신 프로토콜

2.1 스트리밍을 위한 'Marvelmind' 프로토콜

모든 스트리밍 패킷은 동일한 일반 구조를 가집니다:

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	목적지 주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	세부 사항 참조
4	1	uint8_t	전송되는 데이터의 바이트 수	N
5	N	N 바이트	데이터 필드 코드에 따른 페이로드 데이터	
5+N	2	uint16_t	CRC-16 (부록 1 참조)	

소프트웨어 버전 v7.200부터 실시간 타임스탬프가 기본적으로 활성화됩니다. 이는 패킷 0x0081, 0x0083, 0x0084, 0x0085가 각각 패킷 0x0011, 0x0003, 0x0004, 0x0005 대신 스트리밍 출력됨을 의미합니다.

이전 소프트웨어와의 호환성을 위해 로컬 타임스탬프를 사용하는 이전 스트리밍 형식이 필요한 경우, Dashboard의 장치 설정에서 이 옵션을 비활성화할 수 있습니다:

Interfaces	(-) collapse
UART speed, bps	500000
Protocol on UART/USB output	Marvelmind
Raw distances data	disabled
Quality and extended location data	disabled
Telemetry stream	disabled
Telemetry interval, sec (1..255)	n/a
User payload packets number (0..31)	0
Alarm pin function	MMSW0006 required
Alarm pin mode	n/a
PB5 pin function	License SW v7.1 require
Streaming mode	License SW v7.1 require
Debugging data	disabled
SPI data output	n/a
Stream realtime timestamps	enabled

2.1.1 Hedgehog 좌표 패킷

이 패킷은 새로운 좌표가 측정되거나 측정에 실패할 때마다 전송됩니다.

2.1.1.1 이동형 비콘의 mm 해상도 좌표 및 실시간 타임스탬프가 포함된 패킷 (펌웨어 V7.200+)

지원 하드웨어:

Super-Beacon: 지원됨

Industrial Super-Beacon: 지원됨

Modem HW5.1: 지원됨

Super-Modem: 지원됨

Mini-RX (Badge, Helmet 등): UART 케이블 사용 시 지원됨

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: UART 케이블 사용 시 지원됨

Modem HW4.9: 지원됨

Beacon HW4.9: 지원됨

Beacon HW4.5: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	목적지 주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0081
4	1	uint8_t	전송되는 데이터의 바이트 수	N
5	8	int64_t	타임스탬프 – Unix 시간 - 1970.01.01 00:00:00부터의 밀리초 수. Modem 및 Dashboard와 모든 장치가 동기화한 시간.	
13	4	int32_t	Beacon의 X 좌표, mm	
17	4	int32_t	Beacon의 Y 좌표, mm	
21	4	int32_t	Beacon의 Z 좌표(높이), mm	
25	1	uint8_t	플래그 바이트: Bit 0: 1 - 좌표를 사용할 수 없음. X, Y, Z 필드의 데이터를 사용하지 않아야 함. Bit 1: 타임스탬프 단위 표시자 (참고 사항 참조) Bit 2: 1 - 사용자 버튼이 눌림 (V5.23+) Bit 3: 1 - 사용자 장치에 업로드할 데이터가 있음, 섹션 2 참조 (V5.34+) Bit 4: 1 - 사용자 장치에서 데이터를 다운로드하려 함, 섹션 2 참조 (V5.34+) Bit 5: 1 - 두 번째 사용자 버튼이 눌림 (V5.74+) Bit 6: 1 - 다른 Hedgehog에 대한 데이터 (이 패킷을 전송하는 Hedgehog와 동일하지 않음) Bit 7: - 1 - 지오펜싱 구역 외부	
26	1	uint8_t	Hedgehog의 주소	
27	2	uint16_t	비트 0...11: XY 평면에서 Hedgehog 쌍의 방향, 데시디그리 (0...3600) 비트 12: 1 - 좌표가 비콘 쌍의 중심에 대해 지정	

			됨; 0 - 지정된 Hedgehog에 대한 좌표 비트 13: 1 - 방향을 적용할 수 없음 비트 14...15: 예약됨 (0)	
29	2	uint16_t	초음파 송신 시점부터 현재 시간까지 경과한 시간, 밀리초 (V5.88+)	
31	M=N-26		선택적 데이터 - 목록 참조	
31+M	2	uint16_t	CRC-16 (부록 1 참조)	

모바일 비콘 위치 패킷의 선택적 데이터에는 다음 구조가 포함될 수 있습니다:

- 속도 데이터 (7바이트). Dashboard의 모바일 비콘 설정 인터페이스 섹션에서 활성화해야 합니다

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	데이터 필드 코드 = 1은 속도 벡터를 의미함	1
1	2	int16_t	X축 방향 속도, mm/sec	
3	2	int16_t	Y축 방향 속도, mm/sec	
5	2	int16_t	Z축 방향 속도, mm/sec	

2.1.1.2 이동 비콘의 cm 해상도 좌표 패킷

더 이상 사용되지 않는 패킷으로, 0x0081로 대체되었습니다. 해당 구식 패킷의 세부 정보는 여기를 참조하십시오.

2.1.1.3 이동 비콘의 mm 해상도 좌표 패킷

더 이상 사용되지 않는 패킷으로, 0x0081로 대체되었습니다. 해당 구식 패킷의 세부 정보는 여기를 참조하십시오.

2.1.2 고정 비콘의 좌표 패킷

이 패킷은 맵이 동결(frozen)된 상태일 때 전송되며, 10초마다 반복됩니다.

2.1.2.1 고정 비콘의 mm 해상도 좌표 패킷, 데이터 코드 0x0012 (펌웨어 V5.35 이상)

지원 하드웨어:

- Super-Beacon: 지원됨
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): UART 케이블로 지원됨
- Mini-TX: 현재 HW 버전에서는 지원되지 않음
- Mini-TX-2: UART 케이블로 지원됨
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원됨
- Beacon HW4.5: 지원됨

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0012
4	1	uint8_t	전송 데이터의 바이트 수	1+N*14
5	1	uint8_t	패킷 내 beacon 수	N
6	1	N*14 바이트	N개 beacon의 데이터	
6+N*14	2	uint16_t	CRC-16 (부록 1 참조)	

N개 beacon 각각의 데이터 구조 형식:

오프셋	크기 (바이트)	타입	설명
0	1	uint8_t	Beacon 주소
1	4	int32_t	Beacon의 X 좌표, mm
5	4	int32_t	비콘의 Y 좌표, mm
9	4	int32_t	비콘의 Z 좌표(높이), mm
13	1	uint8_t	비트 0: 1 = 위치 적용 불가 비트 1...7: 예약됨

2.1.2.2 고정 비콘의 cm 해상도 좌표 패킷, 데이터 코드 0x0002.

사용되지 않는 패킷으로, 0x0012로 대체되었습니다. 사용되지 않는 패킷의 세부 사항은 여기를 참조하십시오.

2.1.3 원시 관성 센서 데이터 패킷

이 패킷은 새로운 관성 센서 데이터가 수신될 때 전송됩니다.

2.1.3.1 실시간 타임스탬프가 포함된 원시 관성 센서 데이터 패킷, 데이터 코드 0x0083 (펌웨어 V7.200+)

이 패킷은 새로운 관성 센서 데이터가 수신될 때 전송됩니다.

지원 하드웨어:

Super-Beacon: 지원됨, 100 Hz ('Raw inertial sensors data' 활성화 시)

Industrial Super-Beacon: 지원됨, 100 Hz ('Raw inertial sensors data' 활성화 시)

Modem HW5.1: 지원됨, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)

Super-Modem: 지원됨, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)

Mini-RX (Badge, Helmet 등): 지원됨, 100 Hz ('Raw inertial sensors data' 활성화 시)
UART 케이블 사용 시

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: 지원됨, 100 Hz ('Raw inertial sensors data' 활성화 시)

Modem HW4.9: 지원됨, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)

Beacon HW4.9: 지원됨, 100 Hz ('Raw inertial sensors data' 활성화 시)

Beacon HW4.5: 지원됨, 100 Hz ('Raw inertial sensors data' 활성화 시)

타임스탬프에 관한 주의사항을 참조하십시오.

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0083
4	1	uint8_t	전송되는 데이터의 바이트 수	
5	36		데이터 패킷 (하단 참조)	
41	2	uint16_t	CRC-16 (부록 1 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	2	int16_t	가속도계, X축, 1 mg/LSB	
2	2	int16_t	가속도계, Y축, 1 mg/LSB	
4	2	int16_t	가속도계, Z축, 1 mg/LSB	
6	2	int16_t	자이로스코프, X축, 0.0175 dps/LSB	
8	2	int16_t	자이로스코프, Y축, 0.0175 dps/LSB	
10	2	int16_t	자이로스코프, Z축, 0.0175 dps/LSB	
12	2	int16_t	나침반, X축, 1100 LSB/Gauss	
14	2	int16_t	나침반, Y축, 1100 LSB/Gauss	

16	2	int16_t	나침반, Z축, 980 LSB/Gauss
18	1	uint8_t	비콘의 주소
19	5	5 bytes	예약됨 (0)
24	8	int64_t	타임스탬프 – Unix 시간 - 1970.01.01 00:00:00부터의 밀리초 수. Modem 및 Dashboard와 모든 장치가 동기화한 시간.
32	1	uint8_t	플래그: 비트 0: 1 = 가속도계 데이터 없음 비트 1: 1 = 자이로스코프 데이터 없음 비트 2: 1 = 나침반 데이터 없음 비트 3...7 – 예약됨 (0)
33	3	3바이트	예약됨

참고: 나침반 데이터는 IMU가 탑재된 HW v4.9 비콘에서만 사용 가능합니다.

2.1.3.2 원시 관성 센서 데이터 패킷, 데이터 코드 0x0003

더 이상 사용되지 않는 패킷으로, 0x0083으로 대체되었습니다. 해당 구식 패킷의 세부 정보는 여기를 참조하십시오.

2.1.4 원시 거리 데이터 패킷

2.1.4.1 실시간 타임스탬프가 포함된 원시 거리 데이터 패킷, 데이터 코드 0x0084 (펌웨어 V7.200 이상)

이 패킷은 새로운 좌표가 측정되거나 측정에 실패할 때마다, 이동 비콘의 좌표 패킷 이후에 전송됩니다.

설정의 '인터페이스' 섹션에서 "원시 거리 데이터" 옵션이 활성화된 경우에만 사용 가능합니다.

지원 하드웨어:

Super-Beacon: 지원됨

Industrial Super-Beacon: 지원됨

Modem HW5.1: 지원됨

Super-Modem: 지원됨

Mini-RX (Badge, Helmet 등): UART 케이블과 함께 지원됨

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: UART 케이블과 함께 지원됨

Modem HW4.9: 지원됨

Beacon HW4.9: 지원됨

Beacon HW4.5: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0084
4	1	uint8_t	전송 데이터의 바이트 수	
5	36		데이터 패킷 (아래 참조)	
41	2	uint16_t	CRC-16 (부록 1 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Hedgehog 주소	
1	6		거리 항목 1	
7	6		거리 항목 2	
13	6		거리 항목 3	
19	6		거리 항목 4	
25	8	int64_t	타임스탬프 - 비콘 초음파 방출의 유닉스 시간, 1970.01.01 00:00:00 기준 밀리초 수. Modem 및 Dashboard와 모든 장치가 동기화된 시간.	

29	2	uint16_t	초음파 방출 후 현재 시간까지 경과한 시간, 밀리초 (V5.89+)	
31	1	uint8_t	예약됨	

거리 항목의 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	비콘의 주소 (항목이 채워지지 않은 경우 0)	
1	4	uint32_t	비콘까지의 거리, mm	
5	1	uint8_t	비트 0: 1 = 거리 적용 불가 비트 1...7: 예약됨 (0)	

2.1.4.2 실시간 타임스탬프가 포함된 후보 데이터와 함께 원시 거리의 패킷, 데이터 코드 0x0094 (펌웨어 V8.431+)

이 패킷은 새로운 좌표가 측정되거나 측정에 실패할 때마다 모바일 비콘의 좌표가 포함된 패킷 이후에 전송됩니다.

설정의 'Interfaces' 섹션에서 "extra raw distances data" 옵션이 활성화된 경우에만 사용 가능합니다.

지원 하드웨어:

- Super-Beacon: 지원됨
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 요청 시 지원
- Super-Modem: 요청 시 지원
- Mini-RX (Badge, Helmet 등): UART 케이블로 지원
- Mini-TX: 지원되지 않음
- Mini-TX-2: 요청 시 지원
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

타임스탬프에 관한 주의사항을 참조하십시오.

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0094
4	1	uint8_t	전송되는 데이터의 바이트 수	
5	80		데이터 패킷 (하단 참조)	
85	2	uint16_t	CRC-16 (부록 1 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Hedgehog 주소	
1	17		거리 항목 1	
18	17		거리 항목 2	
35	17		거리 항목 3	
52	17		거리 항목 4	
69	8	int64_t	타임스탬프 - 비콘 초음파 방출의 유닉스 시간, 1970.01.01 00:00:00부터의 밀리초 수. Modem 및 Dashboard와 모든 장치가 동기화한 시간.	
77	2	uint16_t	초음파 방출부터 현재 시간까지 경과한 시간, 밀리초 (V5.89+)	
79	1	uint8_t	예약됨	

거리 항목의 형식

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	비콘의 주소 (항목이 채워지지 않은 경우 0)	
1	1	uint8_t	거리 후보의 수	
2	4	uint32_t	비콘까지의 거리 첫 번째 후보, mm	
6	1	uint8_t	첫 번째 후보의 품질, %	
7	4	uint32_t	비콘까지 거리의 두 번째 후보, mm	
11	1	uint8_t	두 번째 후보의 품질, %	
12	4	uint32_t	비콘까지 거리의 세 번째 후보, mm	
16	1	uint8_t	세 번째 후보의 품질, %	

2.1.4.3 원시 거리 데이터 패킷, 데이터 코드 0x0004

더 이상 사용되지 않는 패킷으로, 0x0084로 대체되었습니다. 더 이상 사용되지 않는 패킷에 대한 자세한 내용은 여기를 참조하십시오.

2.1.5 처리된 IMU 데이터 패킷

이 패킷은 새로운 관성 센서 데이터가 사용 가능할 때 전송됩니다.

2.1.5.1 실시간 타임스탬프가 포함된 처리된 IMU 데이터 패킷, 데이터 코드 0x0085 (펌웨어 V7.200 이상)

이 패킷은 새로운 관성 센서 데이터가 사용 가능할 때 전송됩니다.

지원 하드웨어:

Super-Beacon: 지원, 100 Hz ('Processed IMU data' 활성화 시)

Industrial Super-Beacon: 지원, 100 Hz ('Processed IMU data' 활성화 시)

Modem HW5.1: 지원, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)

Super-Modem: 지원, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)

Mini-RX (Badge, Helmet 등): 지원됨, 100 Hz ('Processed IMU data' 활성화 시)

Mini-Rx용 UART 케이블 사용

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: 지원됨, 100 Hz ('Processed IMU data' 활성화 시)

Modem HW4.9: 지원됨, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)

Beacon HW4.9: 지원됨, 100 Hz ('Processed IMU data' 활성화 시)

Beacon HW4.5: 지원됨, 100 Hz ('Processed IMU data' 활성화 시)

타임스탬프에 관한 참고 사항을 확인하십시오.

오프셋	크기 (bytes)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0085
4	1	uint8_t	전송 데이터 바이트 수	
5	46		데이터 패킷 (하단 참조)	
51	2	uint16_t	CRC-16 (부록 1 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	타입	설명	값
0	4	int32_t	비콘의 X 좌표 (융합), mm	
4	4	int32_t	비콘의 Y 좌표 (융합), mm	
8	4	int32_t	비콘의 Z 좌표 (융합), mm	
12	2	int16_t	회전 쿼터니언의 w 필드 (각도)	
14	2	int16_t	회전 쿼터니언의 x 필드 (각도)	
16	2	int16_t	회전 쿼터니언의 y 필드 (각도)	
18	2	int16_t	회전 쿼터니언의 z 필드 (각도)	
20	2	int16_t	비콘의 X축 속도 (융합), mm/s	
22	2	int16_t	비콘의 Y축 속도 (융합), mm/s	
24	2	int16_t	비콘의 Z축 속도 (융합), mm/s	
26	2	int16_t	비콘의 X축 가속도, mm/s ²	

28	2	int16_t	비콘의 Y축 가속도, mm/s ²	
30	2	int16_t	비콘의 Z축 가속도, mm/s ²	
32	1	uint8_t	비콘의 주소	
33	1	1 바이트	예약됨 (0)	
34	8	int64_t	타임스탬프 - Unix 시간 - 1970.01.01 00:00:00부터의 밀리초 수. Modem 및 Dashboard와 모든 장치가 동기화된 시간.	
42	1	uint8_t	플래그: 비트 0: 1 = 위치 데이터 없음 비트 1: 1 = 쿼터니언 데이터 없음 비트 2: 1 = 속도 데이터 없음 비트 3: 1 = 가속도 데이터 없음	
43	3	3 바이트	예약됨 (0)	

참고: 쿼터니언은 10000 값으로 정규화됩니다.

2.1.5.2 처리된 IMU 데이터 패킷, 데이터 코드 0x0005

더 이상 사용되지 않는 패킷으로, 0x0085로 대체되었습니다. 더 이상 사용되지 않는 패킷에 대한 자세한 내용은 여기를 참조하십시오.

2.1.6 원격 측정 데이터 패킷 (데이터 코드 0x0006)

이 패킷은 설정의 'Interfaces' 섹션에서 "Telemetry stream" 옵션이 활성화된 경우, 위치 업데이트 후 전송됩니다.

지원 하드웨어:

Super-Beacon: 지원됨

Industrial Super-Beacon: 지원됨

Modem HW5.1: 지원됨 (펌웨어 V7.000 이상)

Super-Modem: 지원됨 (펌웨어 V7.000 이상)

Mini-RX (Badge, Helmet 등): UART 케이블과 함께 지원됨

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: UART 케이블과 함께 지원됨

Modem HW4.9: 지원되지 않음

Beacon HW4.9: 지원됨

Beacon HW4.5: 지원됨

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0006
4	1	uint8_t	전송되는 데이터의 바이트 수	
5	16		데이터 패킷 (하단 참조)	
21	2	uint16_t	CRC-16 (부록 1 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	타입	설명	값
0	2	uint16_t	배터리 전압, mV	
2	1	int8_t	RSSI, dBm	
3	1	uint8_t	비콘의 주소	
4	12		예약됨 (0)	

2.1.7 품질 및 확장 위치 데이터 패킷 (데이터 코드 0x0007)

이 패킷은 설정의 'Interfaces' 섹션에서 "Quality and extended location data" 옵션이 활성화된 경우, 위치 업데이트 후 전송됩니다.

지원 하드웨어:

Super-Beacon: 지원

Industrial Super-Beacon: 지원

Modem HW5.1: 지원

Super-Modem: 지원

Mini-RX (Badge, Helmet 등): UART 케이블 사용 시 지원

Mini-TX: 현재 HW 버전에서는 미지원

Mini-TX-2: UART 케이블 사용 시 지원

Modem HW4.9: 지원 (품질 필드만 해당)

Beacon HW4.9: 지원 (품질 필드만 해당)

Beacon HW4.5: 지원 (품질 필드만 해당)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0007
4	1	uint8_t	전송 데이터 바이트 수	
5	16		데이터 패킷 (하단 참조)	
21	2	uint16_t	CRC-16 (부록 1 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	장치 주소	
1	1	uint8_t	측위 품질, %	
2	1	uint8_t	0 = 지오펜싱 구역 알람 없음 1...255 - 지오펜싱 구역 인덱스 이 필드는 MMSW0005 라이선스가 필요합니다.	
3	13		예약됨 (0)	

2.1.8 사용 중단된 기능

여기에는 구버전 소프트웨어에서 사용되었으며 기본적으로 다른 패키지로 대체된 더 이상 사용되지 않는 패키지가 설명되어 있습니다.

2.1.8.1 이동형 비콘의 cm 해상도 좌표가 포함된 패킷

사용 중단된 패킷. 0x0081로 대체되었습니다.

지원 하드웨어:

Super-Beacon: 지원됨

Industrial Super-Beacon: 지원됨

Modem HW5.1: 지원됨

Super-Modem: 지원됨

Mini-RX (Badge, Helmet 등): UART 케이블을 통해 지원됨

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: UART 케이블로 지원됨

Modem HW4.9: 지원됨

Beacon HW4.9: 지원됨

Beacon HW4.5: 지원됨

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	목적지 주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0001
4	1	uint8_t	전송되는 데이터의 바이트 수	0x10
5	4	uint32_t	Timestamp – 비콘 초음파 방출의 내부 시간으로, 최근 웨이크업 이벤트 시점부터 밀리초 단위로 표시됩니다. 참고 사항을 확인하십시오.	
9	2	int16_t	비콘의 X 좌표, cm	
11	2	int16_t	비콘의 Y 좌표, cm	
13	2	int16_t	Z 좌표, 비콘의 높이, cm	
15	1	uint8_t	플래그 바이트: Bit 0: 1 - 좌표를 사용할 수 없음. X, Y, Z 필드의 데이터는 사용하지 않아야 합니다. Bit 1: timestamp 단위 표시기 (참고 사항 참조) Bit 2: 1 - 사용자 버튼이 눌림 (V5.23+) Bit 3: 1 - 사용자 장치로 업로드할 데이터가 있음, 섹션 2 참조 (V5.34+) Bit 4: 1 - 사용자 장치에서 데이터를 다운로드하려 함, 섹션 2 참조 (V5.34+) Bit 5: 1 - 두 번째 사용자 버튼이 눌림 (V5.74+) Bit 6: 1 - 다른 Hedgehog의 데이터 (이 패킷을 전송하는 Hedgehog과 동일하지 않음) Bit 7: - 예약됨 (0)	
16	1	uint8_t	Hedgehog의 주소	
17	2	uint16_t	비트 0...11: XY 평면에서 Hedgehog 쌍의 방향, 데시도(decidegrees) (0...3600) 비트 12: 1 - 좌표가 비콘 쌍의 중심에 대해 제공됨; 0 - 지정된 비콘에 대한 좌표 비트 13: 1 - 방향 적용 불가	

			비트 14...15: 예약됨 (0)	
19	2	uint16_t	초음파 발신 시점부터 현재 시간까지 경과 시간, 밀리초 (V5.88+)	
21	2	uint16_t	CRC-16 (부록 1 참조)	

2.1.8.2 모바일 비콘의 mm 해상도 좌표 패킷

더 이상 사용되지 않는 패킷. 0x0081로 대체됨.

지원 하드웨어:

Super-Beacon: 지원됨

Industrial Super-Beacon: 지원됨

Modem HW5.1: 지원됨

Super-Modem: 지원됨

Mini-Rx (Badge, Helmet 등): UART 케이블을 사용하여 지원됨

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: UART 케이블을 사용하여 지원됨

Modem HW4.9: 지원됨

Beacon HW4.9: 지원됨

Beacon HW4.5: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	목적지 주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0011
4	1	uint8_t	전송되는 데이터의 바이트 수	N
5	4	uint32_t	Timestamp - 비콘 초음파 방출의 내부 시간으로, 최근 웨이크업 이벤트 시점으로부터 밀리초 단위로 표시됩니다. 참고 사항을 확인하십시오.	
9	4	int32_t	비콘의 X 좌표, mm	
13	4	int32_t	비콘의 Y 좌표, mm	
17	4	int32_t	Z 좌표, 비콘의 높이, mm	
21	1	uint8_t	플래그 바이트: 비트 0: 1 - 좌표를 사용할 수 없음. X, Y, Z 필드의 데이터는 사용하지 않아야 합니다. 비트 1: timestamp 단위 표시자 (참고 사항 확인) 비트 2: 1 - 사용자 버튼이 눌림 (V5.23+) 비트 3: 1 - 사용자 장치에 업로드할 데이터가 있음, 섹션 2 참조 (V5.34+) 비트 4: 1 - 사용자 장치에서 데이터를 다운로드하려 함, 섹션 2 참조 (V5.34+) 비트 5: 1 - 두 번째 사용자 버튼이 눌림 (V5.74+) 비트 6: 1 - 이 패킷을 전송하는 Hedgehog가 아닌 다른 Hedgehog의 데이터 비트 7: -1 - 지오펜싱 구역 이탈	
22	1	uint8_t	Hedgehog의 주소	
23	2	uint16_t	비트 0...11: XY 평면에서 Hedgehog 쌍의 방향, 데시도(decidegrees) (0...3600) 비트 12: 1 - 좌표가 비콘 쌍의 중심에 대해 제공됨; 0 - 지정된 Hedgehog에 대한 좌표	

			비트 13: 1 - 방향을 적용할 수 없음 비트 14...15: 예약됨 (0)	
25	2	uint16_t	초음파 방출 시점부터 현재 시간까지 경과된 시간, 밀리초 (V5.88+)	
27	M= N-22		선택적 데이터 - 목록 참조	
27+M	2	uint16_t	CRC-16 (부록 1 참조)	

참고: V5.20 이전 펌웨어 버전의 경우 타임스탬프는 1/64초 단위이며 타임스탬프 단위 표시자(플래그 바이트의 비트 1)는 0입니다. 버전 5.20 이상의 경우 타임스탬프는 밀리초 단위이며 타임스탬프 단위 표시자는 1입니다.

2.1.8.3 고정형 비콘의 cm 해상도 좌표가 포함된 패킷, 데이터 코드 0x0002.

더 이상 사용되지 않는 패킷. 0x0012로 대체됨.

지원 하드웨어:

- Super-Beacon: 지원됨
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): UART 케이블로 지원됨
- Mini-TX: 현재 HW 버전에서는 지원되지 않음
- Mini-TX-2: UART 케이블로 지원됨
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원됨
- Beacon HW4.5: 지원됨

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 타입	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0002
4	1	uint8_t	전송 데이터의 바이트 수	1+N*8
5	1	uint8_t	패킷 내 beacon 수	N
6	1	N*8 바이트	N개 beacon에 대한 데이터	
6+N*8	2	uint16_t	CRC-16 (부록 1 참조)	

N개 beacon 각각에 대한 데이터 구조 형식:

오프셋	크기 (바이트)	타입	설명
0	1	uint8_t	Beacon 주소
1	2	int16_t	비콘의 X 좌표, cm
3	2	int16_t	비콘의 Y 좌표, cm
5	2	int16_t	비콘의 Z 좌표 (높이), cm
7	1	uint8_t	예약됨 (0)

2.1.8.4 원시 관성 센서 데이터 패킷, 데이터 코드 0x0003

더 이상 사용되지 않는 패킷. 0x0083으로 대체됨.

지원 하드웨어:

Super-Beacon: 지원, 100 Hz ('Raw inertial sensors data' 활성화 시)
 Industrial Super-Beacon: 지원, 100 Hz ('Raw inertial sensors data' 활성화 시)
 Modem HW5.1: 지원, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)

Super-Modem: 지원, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)

Mini-RX (Badge, Helmet 등): 지원, 100 Hz ('Raw inertial sensors data' 활성화 시)
 UART 케이블 사용 시

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: 지원, 100 Hz ('Raw inertial sensors data' 활성화 시)

Modem HW4.9: 지원, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)

Beacon HW4.9: 지원, 100 Hz ('Raw inertial sensors data' 활성화 시)

Beacon HW4.5: 지원됨, 100 Hz ('Raw inertial sensors data' 활성화 시)

타임스탬프에 관한 참고 사항을 확인하십시오.

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 타입	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0003
4	1	uint8_t	전송 데이터의 바이트 수	
5	32		데이터 패킷 (하단 참조)	
37	2	uint16_t	CRC-16 (부록 1 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	2	int16_t	가속도계, X축, 1 mg/LSB	
2	2	int16_t	가속도계, Y축, 1 mg/LSB	
4	2	int16_t	가속도계, Z축, 1 mg/LSB	
6	2	int16_t	자이로스코프, X축, 0.0175 dps/LSB	
8	2	int16_t	자이로스코프, Y축, 0.0175 dps/LSB	
10	2	int16_t	자이로스코프, Z축, 0.0175 dps/LSB	
12	2	int16_t	나침반, X축, 1100 LSB/Gauss	
14	2	int16_t	나침반, Y축, 1100 LSB/Gauss	
16	2	int16_t	나침반, Z축, 980 LSB/Gauss	
18	1	uint8_t	비콘의 주소	
19	5	5 bytes	예약됨 (0)	
24	4	uint32_t	타임스탬프, ms	

28	1	uint8_t	플래그: 비트 0: 1 = 가속도계 데이터 사용 불가 비트 1: 1 = 자이로스코프 데이터 사용 불가 비트 2: 1 = 나침반 데이터 사용 불가 비트 3...7 - 예약됨 (0)
29	3	3 bytes	예약됨

참고: 나침반 데이터는 IMU가 탑재된 HW v4.9 비콘에서만 사용 가능합니다.

2.1.8.5 원시 거리 데이터 패킷, 데이터 코드 0x0004

더 이상 사용되지 않는 패킷입니다. 0x0084로 대체되었습니다.

지원 하드웨어:

Super-Beacon: 지원됨

Industrial Super-Beacon: 지원됨

Modem HW5.1: 지원됨

Super-Modem: 지원됨

Mini-RX (Badge, Helmet 등): UART 케이블을 통해 지원됨

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: UART 케이블을 통해 지원됨

Modem HW4.9: 지원됨

Beacon HW4.9: 지원됨

Beacon HW4.5: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0004
4	1	uint8_t	전송되는 데이터의 바이트 수	
5	32		데이터 패킷 (하단 참조)	
37	2	uint16_t	CRC-16 (부록 1 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Hedgehog의 주소	
1	6		거리 항목 1	
7	6		거리 항목 2	
13	6		거리 항목 3	

19	6		거리 항목 4	
25	4	uint32_t	타임스탬프 - 비콘 초음파 방출의 내부 시간, 최근 웨이크업 이벤트 시점부터 밀리초 단위 (V5.89+)	
29	2	uint16_t	초음파 방출부터 현재 시간까지 경과된 시간, 밀리초 단위 (V5.89+)	
31	1	uint8_t	예약됨	

거리 항목의 형식

오프셋	크기 (bytes)	타입	설명	값
0	1	uint8_t	비콘의 주소 (항목이 채워지지 않은 경우 0)	
1	4	uint32_t	비콘까지의 거리, mm	
5	1	uint8_t	비트 0: 1 = 거리 적용 불가 비트 1...7: 예약됨 (0)	

2.1.8.6 처리된 IMU 데이터 패킷, 데이터 코드 0x0005

더 이상 사용되지 않는 패킷. 0x0085로 대체됨.

지원 하드웨어:

- Super-Beacon: 각도 지원, 100 Hz ('Processed IMU data' 활성화 시)
- Industrial Super-Beacon: 각도 지원, 100 Hz ('Processed IMU data' 활성화 시)
- Modem HW5.1: 지원, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)
- Super-Modem: 지원, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)
- Mini-RX (Badge 등): 각도 지원, 100 Hz ('Processed IMU data' 활성화 시)
Mini-Rx용 UART 케이블 포함
- Mini-TX: 현재 HW 버전에서는 지원되지 않음
- Mini-TX-2: 각도 지원, 100 Hz ('Processed IMU data' 활성화 시)
- Modem HW4.9: 지원, 시스템 업데이트 속도 ('IMU via modem' 활성화 시)
- Beacon HW4.9: 지원, 100 Hz ('Processed IMU data' 활성화 시)
- Beacon HW4.5: 지원, 100 Hz ('Processed IMU data' 활성화 시)

타임스탬프에 관한 참고 사항을 확인하십시오.

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 타입	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0005
4	1	uint8_t	전송되는 데이터의 바이트 수	
5	42		데이터 패킷 (하단 참조)	
47	2	uint16_t	CRC-16 (부록 1 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	4	int32_t	beacon의 X 좌표 (융합), mm	
4	4	int32_t	beacon의 Y 좌표 (융합), mm	
8	4	int32_t	beacon의 Z 좌표 (융합), mm	
12	2	int16_t	회전 쿼터니언의 W 필드 (각도)	
14	2	int16_t	회전 쿼터니언의 X 필드 (각도)	
16	2	int16_t	회전 쿼터니언의 Y 필드 (각도)	
18	2	int16_t	회전 쿼터니언의 Z 필드 (각도)	
20	2	int16_t	beacon의 X 속도 (융합), mm/s	
22	2	int16_t	Beacon의 Y축 속도 (융합), mm/s	
24	2	int16_t	Beacon의 Z축 속도 (융합), mm/s	
26	2	int16_t	Beacon의 X축 가속도, mm/s ²	
28	2	int16_t	Beacon의 Y축 가속도, mm/s ²	

30	2	int16_t	Beacon의 Z축 가속도, mm/s ²	
32	1	uint8_t	Beacon의 주소	
33	1	1 byte	예약됨 (0)	
34	4	uint32_t	타임스탬프, ms	
38	1	uint8_t	플래그: Bit 0: 1 = 위치 데이터 없음 Bit 1: 1 = 쿼터니언 데이터 없음 비트 2: 1 = 속도 데이터 없음 비트 3: 1 = 가속도 데이터 없음 비트 4...7 - 예약됨 (0)	
39	3	3바이트	예약됨 (0)	

참고: 쿼터니언은 10000 값으로 정규화됨

2.2 사용자 장치로부터/로의 데이터 읽기/쓰기 프로토콜

2.2.1 사용자 장치에서 데이터 전송

지원 하드웨어:

- Super-Beacon: 지원됨
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): UART 케이블로 지원됨
- Mini-TX: 현재 HW 버전에서는 지원되지 않음
- Mini-TX-2: UART 케이블로 지원됨
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원됨
- Beacon HW4.5: 지원됨

사용자 장치가 **Marvelmind** 시스템을 통해 데이터를 전송해야 하는 경우, 다음 프레임을 전송해야 합니다:

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	목적지 주소	0x00
1	1	uint8_t	패킷 타입	0x49
2	2	uint16_t	패킷 내 데이터 코드	0x0200
4	1	uint8_t	전송 데이터 바이트 수	N
5	N	N 바이트	페이로드 데이터	
5+N	2	uint16_t	CRC-16 (부록 1 참조)	

데이터는 Hedgehog의 Dashboard 설정 'Interfaces' 섹션에서 'User payload data size'로 정의된 크기의 단위로 라디오를 통해 모뎀으로 전송됩니다. 이러한 단위의 전송 속도는 Hedgehog의 업데이트 속도와 동일합니다. Hedgehog의 버퍼 크기는 128바이트입니다. 버퍼 오버플로를 방지하기 위해 이 점에 유의하십시오.

2.2.2 사용자 장치에 데이터 쓰기

이 패킷은 Marvelmind 장치(Modem 또는 모바일 비콘)에서 사용자 장치로 전송됩니다.

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	목적지 주소	0xff
1	1	uint8_t	패킷 타입	0x4a
2	2	uint16_t	패킷 내 데이터 코드	0x0200... 0x02ff
4	1	uint8_t	전송 데이터의 바이트 수	N
5	N	N 바이트	페이로드 데이터	
5+N	2	uint16_t	CRC-16 (부록 1 참조)	

이 명령에서 0x200부터 0x2ff까지의 데이터 코드는 예약되어 있습니다.

사용자 장치가 요청을 성공적으로 처리한 경우, 다음 형식으로 응답을 전송해야 합니다:

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Hedgehog의 주소 (스트리밍의 0x0001 또는 0x0011 패킷에서 확인 가능)	
1	1	uint8_t	패킷 유형	0x4a
2	2	uint16_t	패킷 내 데이터 코드	0x0200... 0x02ff
4	2	uint16_t	CRC-16 (부록 1 참조)	

사용자 장치가 요청을 처리하지 못한 경우, 다음 형식으로 응답을 전송합니다:

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Hedgehog의 주소 (스트리밍의 0x0001 패킷에서 가져옴)	
1	1	uint8_t	패킷 유형	0xca
2	2	uint16_t	요청된 데이터의 코드	0x0200... 0x02ff
4	1	uint8_t	오류 코드 (참고 사항 참조)	1
5	2	uint16_t	CRC-16 (부록 1 참조)	

다음 섹션에서는 특정 데이터 쓰기 요청에 대해 설명합니다.

참고: 사용자 장치가 Hedgehog의 요청을 처리할 수 없는 경우, 다음 오류 코드 중 하나를 포함한 응답을 전송해야 합니다:

- 1 - 요청에서 "패킷 유형" 필드를 알 수 없음
- 2 - 요청에서 "데이터 코드" 필드를 알 수 없음
- 3 - 요청의 페이로드 데이터가 올바르지 않음
- 6 - 장치가 사용 중이며 현재 요청된 데이터를 검색할 수 없음

2.2.2.1 이동 경로 쓰기 요청

지원 하드웨어:

- Super-Beacon: 지원됨
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 주문형
- Super-Modem: 주문형
- Mini-RX (Badge, Helmet 등): UART 케이블을 통해 지원됨
- Mini-TX: 현재 HW 버전에서는 지원되지 않음
- Mini-TX-2: UART 케이블로 지원됨
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원됨
- Beacon HW4.5: 지원됨

이 패킷에는 기본 이동 명령 하나가 포함되어 있습니다. **Marvelmind** 장치는 경로 내 모든 기본 이동 명령을 순서대로 하나씩 전송합니다.

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	목적지 주소	0xff
1	1	uint8_t	패킷 유형	0x4a
2	2	uint16_t	패킷 내 데이터 코드	0x201
4	1	uint8_t	전송 데이터의 바이트 수	0x0c
5	12	12 바이트	페이로드 데이터	
17	2	uint16_t	CRC-16 (부록 1 참조)	

페이로드 데이터 형식:

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	기본 동작 유형: 0- 전진 1- 후진 2- 우회전 (시계 방향) 3- 좌회전 (반시계 방향) 4- 일시 정지 5- 처음부터 프로그램 반복 6- 지정된 지점으로 이동 7- 속도 설정	
1	1	uint8_t	이 기본 동작의 인덱스 (0이 첫 번째)	
2	1	uint8_t	기본 동작의 총 수	
3	2	int16_t	동작의 파라미터: 유형 0; 1 - 이동 거리, cm 유형 2; 3 - 회전 각도, 도(degree) 유형 4: 일시 정지 시간, ms 유형 6: X 목표 좌표, cm 유형 7: 속도, %	
5	2	int16_t	동작의 파라미터:	

			유형 6: Y 목표 좌표, cm	
7	2	int16_t	동작의 파라미터: 유형 6: Z 목표 좌표, cm	
9	3	3바이트	예약됨 (0)	

2.2.2.2 구역 쓰기 요청

지원 하드웨어:

- Super-Beacon: 지원됨
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 요청 시 지원
- Super-Modem: 요청 시 지원
- Mini-RX (Badge, Helmet 등): UART 케이블과 함께 지원됨
- Mini-TX: 현재 HW 버전에서는 지원되지 않음
- Mini-TX-2: UART 케이블과 함께 지원됨
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원됨
- Beacon HW4.5: 지원됨

이 패킷에는 구역 목록 시퀀스의 항목 하나가 포함됩니다. **Marvelmind** 장치는 구역 목록에 대한 모든 명령을 순차적으로 전송합니다.

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	목적지 주소	0xff
1	1	uint8_t	패킷 유형	0x4a
2	2	uint16_t	패킷 내 데이터 코드	0x202
4	1	uint8_t	전송 데이터의 바이트 수	0x25
5	37	37 바이트	페이로드 데이터	
42	2	uint16_t	CRC-16 (부록 1 참조)	

페이로드 데이터 형식:

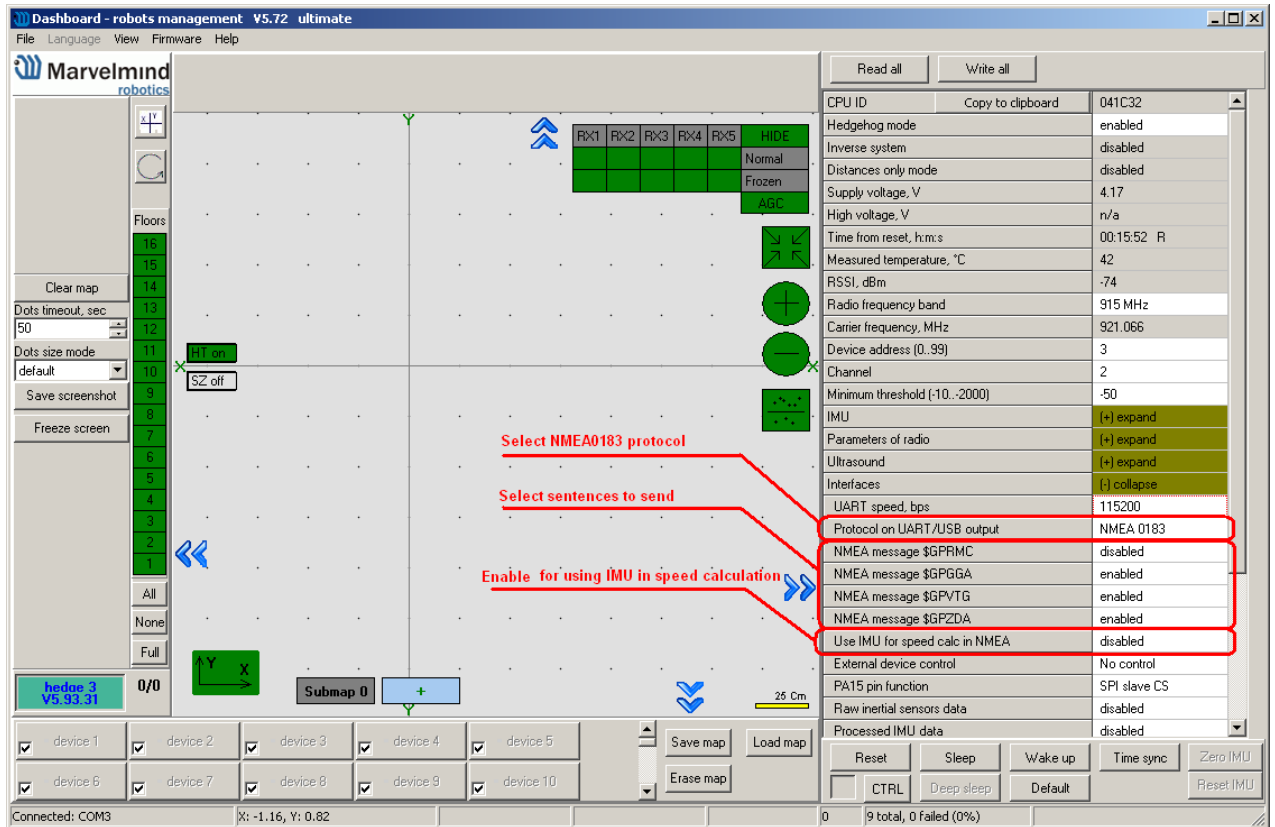
오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	구역의 인덱스	
1	1	uint8_t	구역 다각형의 포인트 수 (N)	
2	1	uint8_t	이 패킷의 첫 번째 포인트 인덱스: M=0...N-1	
3	1	uint8_t	플래그: 비트 0: 1 = 서비스 불가 구역 비트 1: 1 = 주행 불가 구역 비트 2: 1 = 반전 구역 비트 3: 1 = 활성화 구역 비트 4...7: 예약됨 (0)	
4	1	uint8_t	구역 수	
5	32	4x8 바이트	구역 다각형의 최대 4개 포인트 (아래 참조)	

페이로드 데이터 형식:

오프셋	크기 (바이트)	타입	설명	값
0	4	int32_t	포인트의 X 좌표, mm	
4	4	int32_t	포인트의 Y 좌표, mm	

2.3 NMEA0183 통신 프로토콜

모바일 비콘은 UART 및 USB (가상 UART) 인터페이스를 통해 일부 NMEA0183 문장을 출력할 수 있습니다. NMEA 프로토콜은 다음 스크린샷과 같이 Dashboard를 통해 장치에서 활성화해야 합니다:



장치는 업데이트된 위치를 수신할 때마다 활성화된 모든 메시지를 전송합니다.

모바일 비콘(Hedgehog)에서 NMEA 데이터를 수신하려면, 다음 인터페이스 중 하나를 통해 외부 장치(로봇, 드론, AGV 등)에 연결해야 합니다:

1. CDC 클래스의 USB 장치(Windows에서는 가상 COM 포트, Linux에서는 ttyUSB 또는 ttyACM)로 USB 호스트에 연결합니다. Windows에서는 Modem과 동일한 드라이버가 필요합니다. Linux에서는 대부분의 경우 필요한 드라이버가 Linux 커널에 통합되어 있으므로 드라이버가 필요하지 않습니다. 해당 인터페이스에서는 실제 RS-232를 사용하지 않으므로, 호스트에서 열리는 시리얼 포트의 파라미터(보드레이트, 비트 수, 패리티 등)는 임의로 설정할 수 있습니다.
2. Hedgehog의 UART에 연결 - 핀에 2개의 와이어를 납땜해야 합니다. 아래의 비콘 인터페이스 그림을 참조하십시오. 위치 데이터를 출력하려면 GND와 USART2_TX, 2개의 와이어만 연결하면 충분합니다. UART 송신기의 로직 레벨은 CMOS 3.3V입니다. 기본 보드레이트는 500 kbps이며, Dashboard에서 다음 목록 중에서 구성할 수 있습니다(위 그림의 "UART speed, bps" 파라미터 참조): 4.8, 9.6, 19.2, 38.4, 57.6, 115.2, 500 kbps. 데이터 형식: 8비트, 패리티 없음, 정지 비트 1개.

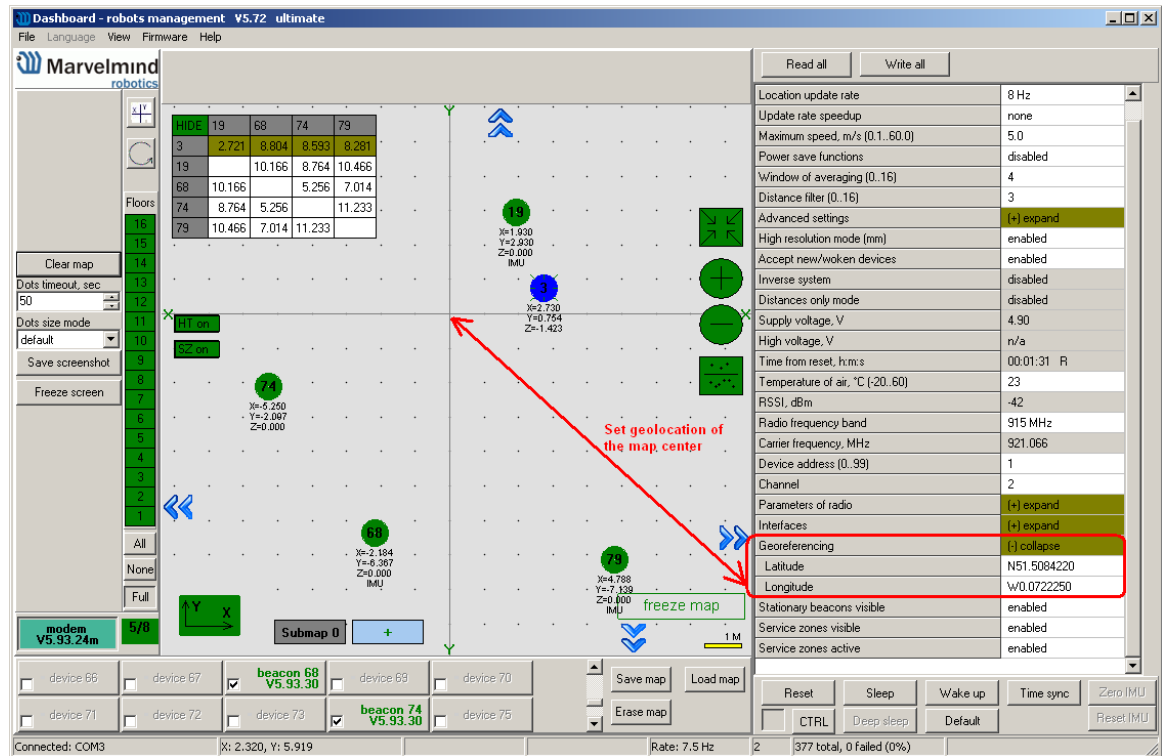
13.11.

2.3.1 좌표 변환을 위한 일반 규칙

Marvelmind 시스템은 직교 데카르트 좌표계 (x, y, z) 형태로 위치를 측정하며, 대부분의 경우 z는 높이를 나타냅니다. GPS 좌표로 변환하기 위해 다음 규칙이 사용됩니다:

- z축은 위쪽 방향을 가리키며, z 좌표는 해발 고도를 의미합니다;
- y축은 북쪽 방향을 가리키므로, y는 위도에 해당합니다;
- x축은 동쪽을 향하므로 x는 경도입니다.
- 점 (x= 0, y= 0)은 지리 참조 점에 따른 GPS 좌표를 가집니다 (기본값: 북위 0°, 서경 0°).

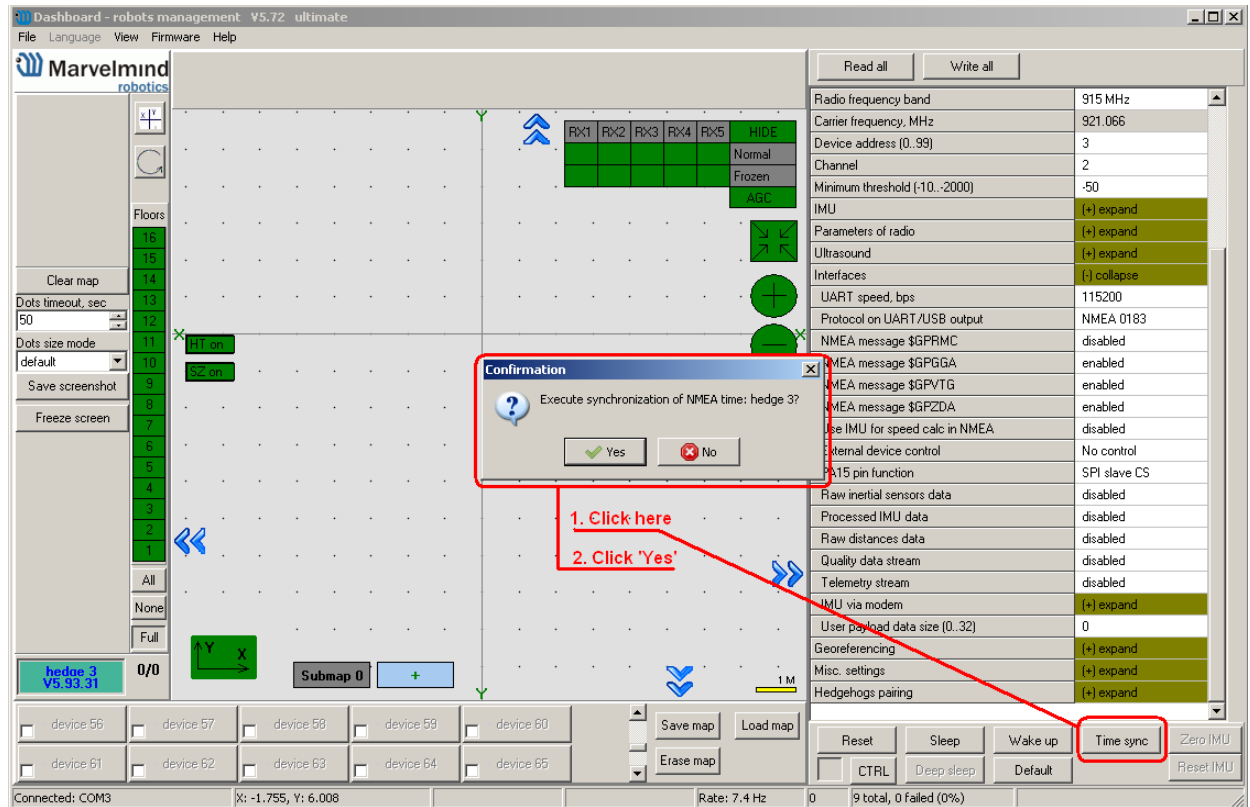
지리 참조 좌표는 스크린샷에 표시된 방법으로 설정할 수 있습니다.



GPS 좌표는 지정된 지리 참조 점과 WGS-84 지구 모델에 따라 계산됩니다. 상세 내용은 다음과 같습니다. $Lat = Lat_ref + y * 9.013373$ 여기서 Lat - 위도, 마이크로도 Lat_ref - 지리 참조 위도, 마이크로도 y - Marvelmind 시스템의 y 좌표, 미터 $Long = Long_ref + x * 8.98315 / \cos(Lat_ref / 1000000)$ Long - 경도, 마이크로도 Long_ref - 지리 참조 경도, 마이크로도 Lat_ref - 지리 참조 위도, 마이크로도 x - Marvelmind 시스템의 x 좌표, 미터

2.3.2 시간에 관한 일반 사항

전원이 켜진 후, 모바일 비콘은 2016.08.01 00:00:00부터 시간을 계산하기 시작합니다. 사용자는 다음 스크린샷에 표시된 방법으로 컴퓨터 시계와 시간을 동기화할 수 있습니다.



2.3.3 "NMEA0183" 메시지 구현 설명

NMEA 0183 메시지는 ASCII로 인코딩된 텍스트 프레임으로, 쉼표로 구분된 여러 부분으로 구성되며 줄 끝으로 종료됩니다. 줄 끝 앞에는 모든 메시지가 '*' 기호로 종료되고, 그 뒤에 NMEA 0183 표준에 따라 계산된 체크섬의 두 기호가 이어집니다.

NMEA 0183 메시지의 각 부분은 특정 매개변수를 나타냅니다.

아래에 지원되는 모든 메시지 및 매개변수 필드에 대한 설명이 나와 있습니다.

메시지 형식은 2002년 1월 1일자 NMEA 0183 표준 버전 3.01에서 가져왔습니다.

1. \$GPRMC - 권장 최소 특정 GNSS 데이터

지원 하드웨어:

Super-Beacon: 지원됨

Industrial Super-Beacon: 지원됨

Modem HW5.1: 지원됨 (SW V7.000부터)

Super-Modem: 지원됨 (SW V7.000부터)

Mini-RX (Badge, Helmet 등): UART 케이블과 함께 지원됨

Mini-TX: 현재 HW 버전에서는 지원되지 않음

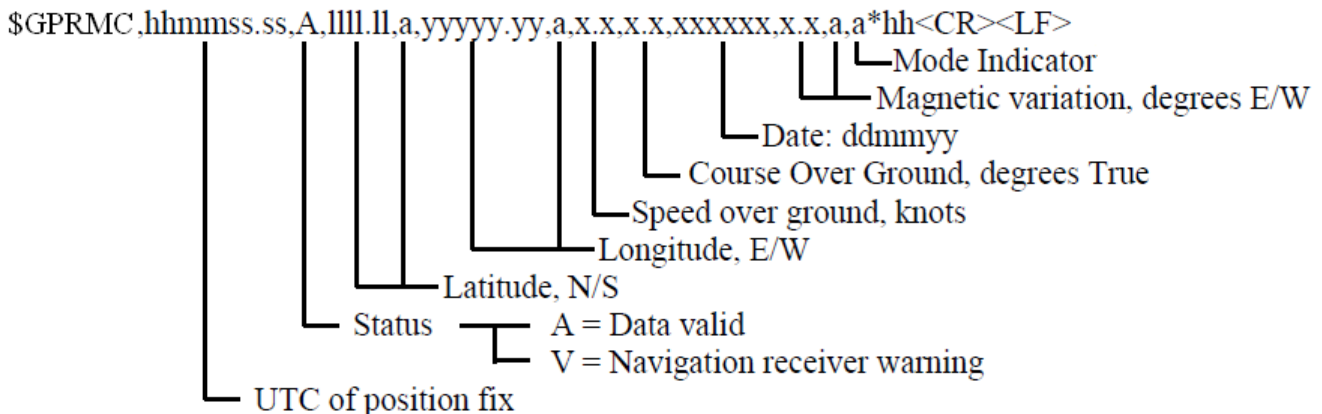
Mini-TX-2: UART 케이블과 함께 지원됨

Modem HW4.9: 지원되지 않음

Beacon HW4.9: 지원됨

Beacon HW4.5: 지원됨

NMEA 0183 표준의 일반 형식:



필드 구현 설명:

1.1. '\$GPRMC' - 메시지 유형 지정

1.2. 'hhmmss.ss' - UTC 위치 수정 시각

일반 규칙에 따라, 시각은 기본값인 2016.01.01부터 계산되거나 컴퓨터 시계와 동기화됩니다.

1.3. 'A' - 상태

마지막 위치 업데이트가 성공적으로 완료된 경우 'A' 값이 전송됩니다.

마지막 위치 업데이트에서 오류가 발생한 경우 'V' 값이 전송됩니다.

1.4. 'llll.ll,a' - 위도, N/S

일반 규칙에 따라(위 참조), 위도는 지리 참조 위치에 대한 Y 좌표에 해당합니다. 위도는 분의 소수점 이하 6자리로 표시되며, 이는 2 mm 이하의 분해능을 제공합니다.

1.5. 'yyyy.yy,xxxxxx,x.x' - 경도, E/W

일반 규칙에 따라(위 참조), 경도는 지리 참조 위치에 대한 X 좌표에 해당합니다. 경도는 분의 소수점 이하 6자리로 표시되며, 이는 2 mm 이하의 분해능을 제공합니다.

1.6. 'x.x' - 대지 속도, 노트

Marvelmind 시스템은 좌표를 측정하며, 속도는 일부 필터링을 적용하여 좌표의 동적 변화로부터 계산됩니다. 선택적으로, 속도 계산에 IMU 용합을 사용할 수 있습니다.

1.7. 'xxxxxx' - 날짜: ddmmyy

일반 규칙에 따라, 시각은 기본값인 2016.01.01부터 계산되거나 컴퓨터 시계와 동기화됩니다.

1.8. 'x.x,a' - 자기 편차

이 파라미터 값은 항상 null 필드입니다.

1.9. 'a' - 모드 표시기

마지막 위치 업데이트가 성공한 경우 'A' 값(자율 모드)이 전송됩니다.

마지막 위치 업데이트에서 오류가 발생한 경우 'N' 값(데이터 유효하지 않음)이 전송됩니다.

2. \$GPGGA - 글로벌 위치 확인 시스템 고정 데이터

지원 하드웨어:

Super-Beacon: 지원

Industrial Super-Beacon: 지원

Modem HW5.1: 지원 (SW V7.000부터)

Super-Modem: 지원 (SW V7.000부터)

Mini-RX (Badge, Helmet 등): Mini-Rx용 UART 케이블을 사용하여 지원

Mini-TX: 현재 HW 버전에서는 지원되지 않음

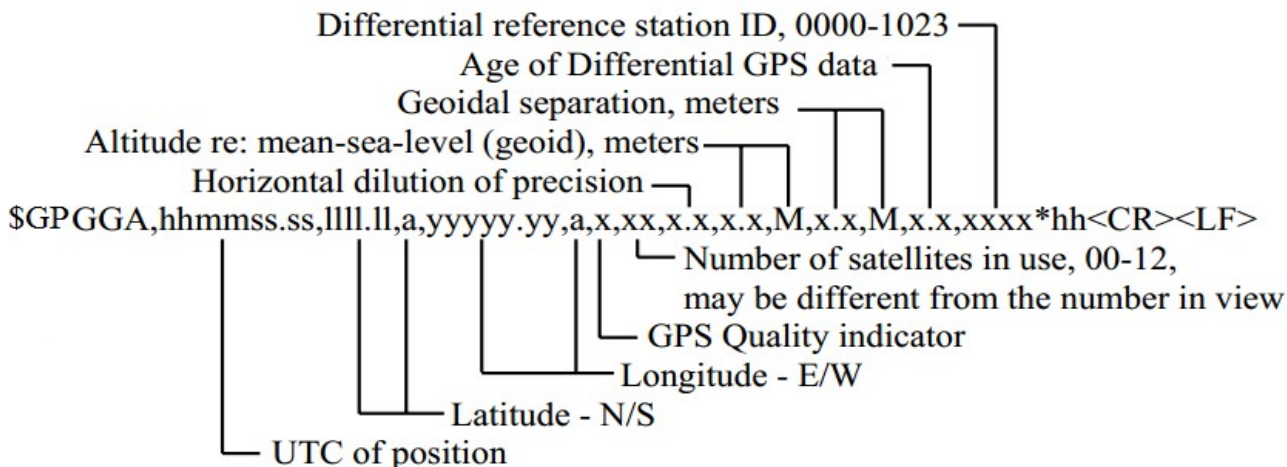
Mini-TX-2: 지원

Modem HW4.9: 지원되지 않음

Beacon HW4.9: 지원

Beacon HW4.5: 지원

NMEA 0183 표준의 일반 형식:



필드 구현 설명:

2.1. '\$GPGGA' - 메시지 유형 지정

2.2. 'hhmmss.ss' - UTC 위치 고정

일반 규정에 따라 시간은 기본값 2016.01.01부터 카운트되거나 컴퓨터 시계와 동기화됩니다.

2.3. 'llll.ll,a' - 위도, N/S

일반 규정에 따라(위 참조), 위도는 지리참조 위치에 대한 Y 좌표에 해당합니다. 위도는 분(minutes)의 소수 자릿수 6자리로 표시되며, 이는 2 mm 이하의 분해능을 제공합니다.

2.4. 'yyyyy.yyyyyy,a' - 경도, E/W

일반 규정에 따라(위 참조), 경도는 지리참조 위치에 대한 X 좌표에 해당합니다. 경도는 분(minutes)의 소수 자릿수 6자리로 표시되며, 이는 2 mm 이하의 분해능을 제공합니다.

2.5. 'x' – GPS 품질 지시자

마지막 위치 업데이트가 성공한 경우 '1' (GPS SPS 모드, 유효한 Fix) 값이 전송됩니다.

마지막 위치 업데이트에서 오류가 발생한 경우 '0' (Fix를 사용할 수 없거나 유효하지 않음) 값이 전송됩니다.

2.6. 'xx' – 사용 중인 위성 수

현재 구현에서는 항상 '08'입니다.

2.7. 'x.x' – 수평 정밀도 저하율(HDOP)

현재 구현에서는 항상 '1.2'입니다.

2.8. 'x.x, M' – 평균 해수면(지오이드) 기준 고도, 단위: m

이는 일반 규정에 따른 Z 좌표에 해당합니다.

2.9. 'x.x, M' – 지오이드 분리, 단위: m

항상 '0.0, M' 값이 전송됩니다.

2.10. 'x.x' – 차등 GPS 데이터의 경과 시간

이 파라미터 값은 항상 null 필드이며, DGPS는 사용되지 않습니다.

2.11. 'xxxx' – 차등 기준국 ID

이 파라미터 값은 항상 null 필드입니다.

3. \$GPVTG - 지상 진행 방위 및 지상 속도

지원 하드웨어:

Super-Beacon: 지원

Industrial Super-Beacon: 지원

Modem HW5.1: 지원 (SW V7.000부터)

Super-Modem: 지원 (SW V7.000부터)

Mini-RX (Badge, Helmet 등): Mini-Rx용 UART 케이블을 사용하여 지원

Mini-TX: 현재 HW 버전에서는 지원되지 않음

Mini-TX-2: 지원

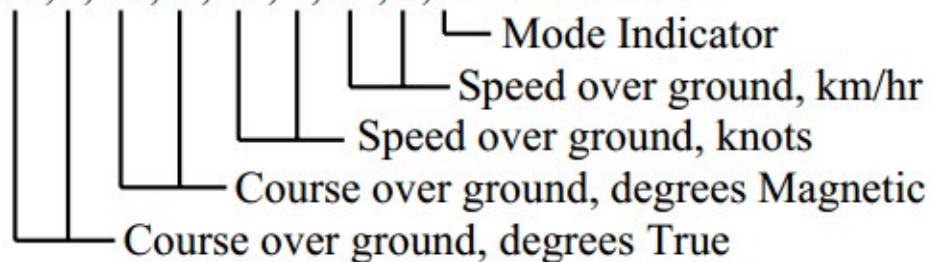
Modem HW4.9: 지원되지 않음

Beacon HW4.9: 지원

Beacon HW4.5: 지원

NMEA 0183 표준의 일반 형식:

\$GPVTG,x.x,T,x.x,M,x.x,N,x.x,K,a*hh<CR><LF>



필드 구현 설명:

3.1. '\$GPVTG' – 메시지 유형 지정

3.2. 'x.x, T' – 지상 진행 방위, 진북 기준 각도

NMEA 표준에 따르면, 방위는 속도 벡터와 북쪽 방향 사이의 각도입니다. 위의 일반 규약에서 명시된 바와 같이, Y축이 북쪽 방향으로 사용됩니다.

3.3. 'x.x, M' – 지상 진행 방위, 자북 기준 각도

현재 구현에서 자북 방위는 진북 방위와 동일합니다.

3.4. 'x.x, N' – 지상 속도, 노트(knots)

Marvelmind 시스템은 좌표를 측정하며, 속도는 일부 필터링을 적용하여 좌표의 동적 변화로부터 계산됩니다. 선택적으로 속도 계산에 IMU 용합을 사용할 수 있습니다.

3.5. 'x.x, K' – 대지 속도, km/hr

다른 단위로 표현된 동일한 속도입니다.

3.6. 'a' – 모드 표시기

마지막 위치 업데이트가 성공적으로 완료된 경우 'A' 값(자율 모드)이 전송됩니다.

마지막 위치 업데이트에서 오류가 발생한 경우 'N' 값(데이터 유효하지 않음)이 전송됩니다.

4. \$GPZDA – 시간 및 날짜

지원 하드웨어:

Super-Beacon: 지원됨

Industrial Super-Beacon: 지원됨

Modem HW5.1: 지원됨 (SW V7.000부터)

Super-Modem: 지원됨 (SW V7.000부터)

Mini-RX (Badge, Helmet 등): UART 케이블을 통해 지원됨

Mini-TX: 현재 HW 버전에서는 지원되지 않음

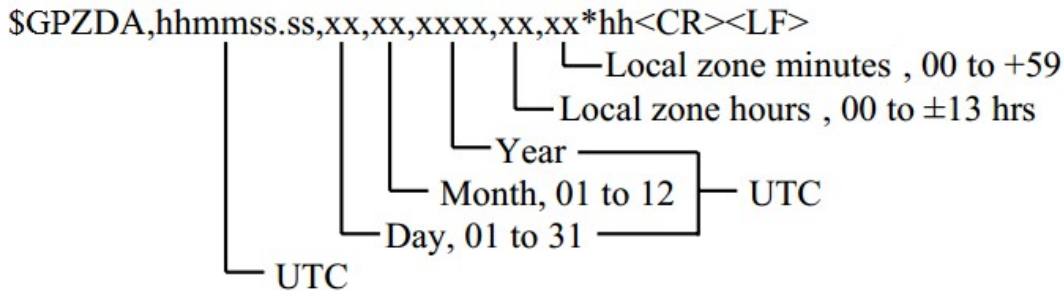
Mini-TX-2: UART 케이블을 통해 지원됨

Modem HW4.9: 지원되지 않음

Beacon HW4.9: 지원됨

Beacon HW4.5: 지원됨

NMEA 0183 표준의 일반 형식:



일반적인 규약에 따라, 시간은 기본값인 2016.01.01부터 계산되거나 컴퓨터 시계와 동기화됩니다.

필드 구현에 대한 설명:

4.1. '\$GPZDA' – 메시지 유형 지정

4.1. 'hhmmss.ss' – UTC

시간(시, 분, 초).

4.2. 'xx' – 일, 01~31

일.

4.3. 'xx' – 월, 01~12

월.

4.4. 'xxxx' – 연도

연도.

4.4. 'xx' – 현지 시간대 시

현지 시간대는 항상 "00"시입니다.

4.5. 'xx' – 현지 시간대 분
현지 시간대는 항상 "00"분입니다.

5. \$GPHDT – 방위각(Heading)

지원 하드웨어:

Super-Beacon: 지원됨

Industrial Super-Beacon: 지원됨

Modem HW5.1: 지원됨 (SW V7.000부터)

Super-Modem: 지원됨 (SW V7.000부터)

Mini-RX (Badge, Helmet 등): UART 케이블로 지원됨

Mini-TX: 지원되지 않음

Mini-TX-2: UART 케이블로 지원됨

Modem HW4.9: 지원되지 않음

Beacon HW4.9: 지원되지 않음

Beacon HW4.5: 지원되지 않음

이 패킷의 스트리밍을 활성화하려면 MMSW0002 라이선스가 필요합니다.

NMEA 0183 표준의 일반 형식:

```
$GPHDT,x.x,T*hh<CR><LF>  
└──Heading, degrees True
```

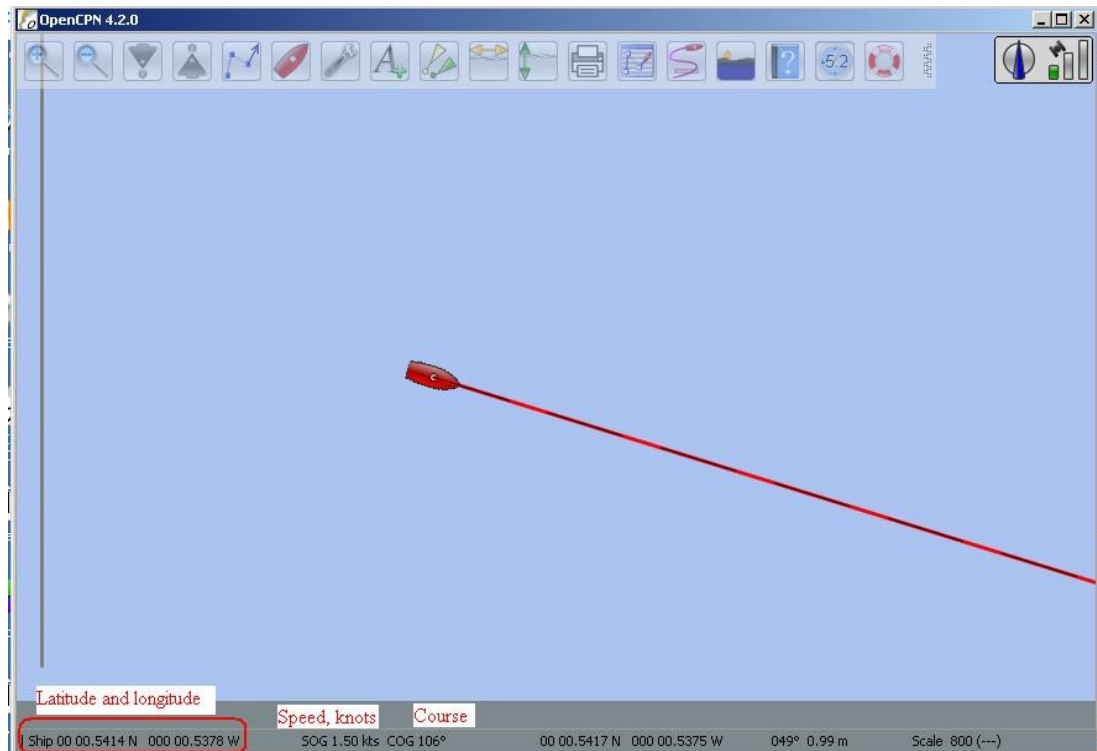
5.1. '\$GPHDT' – 메시지 유형 지정

5.2. 'x.x, T' – 방위각, 진북 기준 도(degree)

이것은 페어드 비콘 또는 페어드 마이크로폰 기능을 자이로스코프와 융합하여 계산된 방위각입니다.

2.3.4 NMEA 데이터 수신 예시

다음 스크린샷은 MS Windows 운영 체제의 컴퓨터에서 실행 중인 OpenCPN 소프트웨어에 USB(가상 COM 포트)를 통해 연결된 모바일 비콘으로부터 수신된 데이터의 예시입니다.



3. USB(가상 UART)를 통한 통신 프로토콜

3.1 스트리밍을 위한 'Marvelmind' 프로토콜

UART 해당 섹션에 설명된 모든 패킷은 USB(가상 UART)를 통해서도 사용할 수 있습니다. 이 데이터는 'UART Cable for Mini-Rx' 없이도 Mini-TX 및 Mini-RX에서 사용할 수 있습니다. Marvelmind 장치가 이 프로토콜에 따른 요청을 수신하면 스트리밍이 5초 동안 중단됩니다.

3.2 사용자 장치와의 데이터 읽기/쓰기 프로토콜

UART 해당 섹션에 설명된 모든 패킷은 USB(가상 UART)를 통해서도 사용할 수 있습니다. 이 데이터는 'UART Cable for Mini-Rx' 없이도 Mini-TX 및 Mini-RX에서 사용할 수 있습니다.

3.3 NMEA0183 통신 프로토콜

UART에 해당하는 섹션에 설명된 모든 패킷은 USB(가상 UART)를 통해서도 사용할 수 있습니다.

이 데이터는 'UART Cable for Mini-Rx' 없이도 Mini-TX 및 Mini-RX에서 사용할 수 있습니다.

Marvelmind 장치가 이 프로토콜에 따른 요청을 수신하면 스트리밍이 5초 동안 중단됩니다.

3.4 USB 인터페이스를 통한 Modem과의 데이터 교환 프로토콜

이 프로토콜은 Dashboard 소프트웨어 및 다음 장에 설명된 Marvelmind API에서 사용됩니다.

Modem은 CDC 클래스의 USB 장치(Windows에서는 가상 COM 포트, Linux에서는 ttyUSB 또는 ttyACM)로 USB 호스트에 연결됩니다.

이 인터페이스에서는 실제 RS-232를 사용하지 않으므로, 호스트에서 설정하는 시리얼 포트 파라미터(보드레이트, 비트 수, 패리티 등)는 어떤 값이든 상관없습니다.

데이터는 바이너리 형식입니다.

USB를 통해 연결된 장치의 «네트워크 주소»는 0xff입니다.

멀티바이트 숫자는 하위 바이트부터 전송됩니다(리틀 엔디안 형식).

3.4.1 최신 좌표 패킷 읽기 (펌웨어 V5.13 이상)

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

요청 프레임 형식 (호스트에서 **Modem**으로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x03
2	2	uint16_t	패킷 내 데이터 코드	0x4110
4	2	uint16_t	접근 모드	0x0000
6	2	uint16_t	CRC-16 (부록 1 참조)	0xc004

응답 프레임 형식 (**Modem**에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 타입	0x03
2	1	uint8_t	전송 데이터 바이트 수	0x64
3	100 (0x64)	100 바이트	데이터 구조 (아래 참조)	
103	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

데이터 필드 형식 (**100** 바이트)

오프셋	크기 (바이트)	설명
0	96 (6*16)	Modem이 수신한 마지막 6개의 좌표 구조 (아래 참조)
96	1	플래그 바이트: 비트 0...1: 예약됨 비트 2: 1 = 사용자 데이터 사용 가능 (섹션 12 참조) 비트 3...7: 예약됨
97	3	예약됨

좌표 구조 형식 (**16** 바이트)

오프셋	크기 (바이트)	설명
0	1	장치 주소
1	4	X 좌표, mm (int32_t)
5	4	Y 좌표, mm (int32_t)

9	4	Z 좌표, mm (int32_t)
13	1	플래그 바이트: 비트 0: 1 – 유효한 좌표 없음 (Dashboard의 빨간색 모드) 비트 1: 1 – 고정된 맵에서의 임시 이동 비콘 (파란색 모드) 비트 2: 1 – 비콘이 Hedgehog 위치 측정에 사용됨
14	2	예약됨 (0)

3.4.2 Modem 구성 읽기/쓰기

3.4.2.1 Modem 구성 읽기 (펌웨어 V5.30+)

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x03
2	2	uint16_t	패킷 내 데이터 코드	0x5000
4	2	uint16_t	액세스 모드	0x0000
6	2	uint16_t	CRC-16 (부록 1 참조)	0x0550

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem의 주소	0xff
1	1	uint8_t	패킷 타입	0x03
2	1	uint8_t	데이터 전송 바이트 수	0x30
3	0x30	구조체	데이터 구조체 (아래 참조)	
0x33	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답의 형식은 부록 2에 설명되어 있습니다.

3.4.2.2 Modem 설정 쓰기

경고! Modem 설정을 쓰려면 먼저 설정을 읽고, 다음 섹션에 설명된 데이터 필드를 구성한 후 쓰기를 수행해야 합니다. 구조체의 다른 바이트는 변경하지 마십시오. 변경 시 Modem의 동작이 저하될 수 있습니다.

요청 프레임의 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem의 주소	0xff
1	1	uint8_t	패킷 타입	0x10
2	2	uint16_t	패킷 내 데이터 코드	0x5000
4	2	uint16_t	액세스 모드	0x0000
6	1	uint8_t	데이터 전송 바이트 수	0x30
7	0x30	structure	데이터 구조 (아래 참조)	
0x37	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x10
2	2	uint16_t	데이터 코드	0x5000
4	2	uint16_t	예약됨	
6	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답의 형식은 부록 2에 설명되어 있습니다.

3.4.2.3 Modem 구성 데이터 구조

데이터 구조의 많은 필드는 설명되어 있지 않습니다. 해당 필드를 변경하지 마십시오! 이 필드들은 **Dashboard** 프로그램에서 시스템 조정에 사용되며, 무단으로 변경할 경우 **Modem**의 작동이 저하될 수 있습니다.

오프셋	크기 (바이트)	유형	설명
0	20	20 바이트	미설명
20	1	int8_t	공기 온도 설정 V_t (부호 있음). Temperature is (V_t+23) °C
21	1	uint8_t	지도 좌표 $X=0, Y=0$ 을 가져야 하는 Beacon의 주소
22	4	4 바이트	미설명
26	1	uint8_t	지도 좌표 $X>0, Y=0$ 을 가져야 하는 Beacon의 주소
27	1	uint8_t	지도 좌표 $Y>0$ 을 가져야 하는 Beacon의 주소
28	1	uint8_t	제어 플래그: 비트 0: 미설명 비트 1: 1 - 모바일 Beacon 이동 필터링 활성화 비트 2: 미설명 비트 3: 1 - 고해상도 모드 (출력 좌표를 cm 대신 mm 단위로 표시) 비트 4: 미설명 비트 5: 1 = 전체 맵 미러링 비트 6: 1 = 절전 모드 (절전 모드는 모든 서브맵이 동결된 경우에만 작동) 비트 7: 설명 없음
29	2	2 바이트	설명 없음
31	1	uint8_t	N, Hedgehog 좌표 검색의 최대 주파수를 결정 $F(N) = 2^{(N-1)}$ Hz, $N = 0...4$, $F(5) = 12$ Hz, $F(6) = 16$ Hz, $F(7) = 16+$ (최대)
32	16	16 바이트	설명 없음

3.4.3 서브맵 구성 읽기/쓰기

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

3.4.3.1 서브맵 구성 읽기 (펌웨어 V5.30+)

요청 프레임 형식 (호스트에서 **Modem**으로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x03
2	2	uint16_t	패킷 내 데이터 코드	0x60XX (XX는 서브맵 번호)
4	2	uint16_t	액세스 모드	0x0000
6	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (**Modem**에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 타입	0x03
2	1	uint8_t	데이터 전송 바이트 수	0x50 (80)
3	80	구조체	데이터 구조 (아래 참조)	
83	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답의 형식은 부록 2에 설명되어 있습니다.

3.4.3.2 서브맵 구성 쓰기 (펌웨어 V5.30+)

경고! 서브맵 구성을 쓰려면 구성을 먼저 읽고, 다음 섹션에 설명된 데이터 필드를 설정한 후 쓰기를 수행해야 합니다. 구조에서 다른 바이트는 변경하지 마십시오. 변경 시 Modem의 동작이 저하될 수 있습니다.

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 타입	0x10
2	2	uint16_t	패킷 내 데이터 코드	0x60XX (XX는 서브맵 번호)
4	2	uint16_t	액세스 모드	0x0000
6	1	uint8_t	데이터 전송 바이트 수	0x50 (80)
7	80	structure	데이터 구조 (아래 참조)	
87	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x10
2	2	uint16_t	데이터 코드	0x5000
4	2	uint16_t	예약됨	
6	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

3.4.3.3 서브맵 구성 데이터 구조

데이터 구조의 많은 필드는 설명되어 있지 않습니다. 해당 필드를 변경하지 마십시오! 이 필드들은 Dashboard 프로그램에서 시스템 조정에 사용되며, 무단 변경 시 Modem의 작동이 저하될 수 있습니다.

오프셋	크기 (바이트)	유형	설명
0	1	uint8_t	서브맵 구성을 위한 시작 비콘의 주소
1	1	uint8_t	제어 워드: 비트 0: 1 - 서브맵 고정 (서브맵 동결) 비트 1: 1 - 비콘이 Hedgehog보다 높은 위치에 있음 비트 2...4: 설명 없음 비트 5: 1 - 서브맵 미러링 비트 6...7: 설명 없음
2	1	uint8_t	거리 제한: 비트 0...6: 수동 거리 제한 (비트 7 = 1인 경우) 비트 7: 0 - 자동 제한, 1 = 수동
3	13	13 바이트	설명 없음
16	2	int16_t	서브맵의 X 이동, cm
18	2	int16_t	서브맵의 Y 이동, cm
20	2	uint16_t	서브맵 회전, 센티도(centidegrees)
22	58	58 바이트	설명 없음

3.4.4 장치 절전/깨우기

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	장치 주소	0x01...0xfe
1	1	uint8_t	패킷 유형	0x10
2	2	uint16_t	패킷 내 데이터 코드	0xb006
4	2	uint16_t	액세스 모드	웨이브용 : 0x0002 기타 : 0x0001
6	1	uint8_t	데이터 전송 바이트 수	0x08
7	1	uint8_t	패스워드, 바이트 0	0x2d
8	1	uint8_t	패스워드, 바이트 1	0x94
9	1	uint8_t	비밀번호, 바이트 2	0x5e
10	1	uint8_t	비밀번호, 바이트 3	0x81
11	1	uint8_t	명령: 0 - 표준 슬립 1 - 딥 슬립 (HW 리셋 시에만 깨어남) 2 - 표준 슬립에서 깨어나기 3...255 - 예약됨	0...2
12	3	3 바이트	예약됨	
15	2	uint16_t	CRC-16 (부록 1 참조)	

깨우기 명령에 대한 응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	장치 주소	0x01...0xfe
1	1	uint8_t	패킷 타입	0x10
2	2	uint16_t	데이터 코드	0xb006
4	2	uint16_t	예약됨	
6	2	uint16_t	CRC-16 (부록 참조)	

슬리핑 명령에 대한 응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	장치 주소	0x01...0xfe
1	1	uint8_t	패킷 유형 (Modem 응답)	0x7f
2	2	uint16_t	데이터 코드	0xb006
4	2	uint16_t	예약됨	
6	2	uint16_t	바이트 0...5의 CRC-16 (부록 1 참조)	
8	1	uint8_t	장치 주소	0x01...0xfe
9	1	uint8_t	패킷 유형	0x10
10	2	uint16_t	데이터 코드	0xb006
12	2	uint16_t	예약됨	
14	2	uint16_t	바이트 8...13에 대한 CRC-16(부록 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

3.4.5 장치 주소 설정

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	장치 주소	0x01...0xfe
1	1	uint8_t	패킷 타입	0x10
2	2	uint16_t	패킷 내 데이터 코드	0x0101
4	2	uint16_t	액세스 모드	0x0000
6	1	uint8_t	데이터 전송 바이트 수	0x02
7	1	uint8_t	데이터 항목 코드 (주소)	0x00
8	1	uint8_t	장치의 새 주소	
9	2	uint16_t	CRC-16 (부록 1 참조)	

3.4.6 측정된 원시 거리 읽기

지원 하드웨어:

- Super-Beacon: 미지원
- Industrial Super-Beacon: 미지원
- Modem HW5.1: 지원
- Super-Modem: 지원
- Mini-RX (Badge, Helmet 등): 미지원
- Mini-TX: 미지원
- Mini-TX-2: 미지원
- Modem HW4.9: 지원
- Beacon HW4.9: 미지원
- Beacon HW4.5: 지원되지 않음

이 명령은 두 가지 모드로 접근할 수 있습니다:

- 데이터 코드 0x4000 사용 - 마지막 8개의 거리 읽기. 응답 프레임에는 요청 시점으로부터 마지막으로 측정된 8개의 거리가 포함됩니다
- 데이터 코드 0x4001 사용 - 모든 거리를 프레임 단위로 읽기. 다음 요청에 대한 응답 프레임에는 저장된 다음 8개의 측정 거리가 포함됩니다. 거리 테이블이 모두 전송되면 처음부터 다시 시작됩니다

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 타입	0x03
2	2	uint16_t	패킷 내 데이터 코드	0x4000 또는 0x4001
4	2	uint16_t	액세스 모드	0x0000
6	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 타입	0x03
2	1	uint8_t	전송 데이터 바이트 수	0x28
3	40 (0x28)	40 바이트	데이터 구조 (아래 참조)	
43	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답의 형식은 부록 2에 설명되어 있습니다.

데이터 필드 형식 (40 바이트)

오프셋	크기 (바이트)	설명
0	32 (8*4)	8개의 원시 거리 구조 (아래 참조)
32	8	예약됨

거리 구조 형식 (4 바이트)

오프셋	크기 (바이트)	설명

0	1	초음파 수신기 주소
1	1	초음파 송신기 주소
2	2	장치 간 측정 거리, mm (uint16_t)

3.4.7 비콘 상태 읽기 (펌웨어 V5.33+)

지원 하드웨어:

- Super-Beacon: 미지원
- Industrial Super-Beacon: 미지원
- Modem HW5.1: 지원
- Super-Modem: 지원
- Mini-RX (Badge, Helmet 등): 미지원
- Mini-TX: 미지원
- Mini-TX-2: 미지원
- Modem HW4.9: 지원
- Beacon HW4.9: 미지원
- Beacon HW4.5: 미지원

요청 프레임 형식 (호스트에서 modem으로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	장치 주소	0x01...0xfe
1	1	uint8_t	패킷 유형	0x03
2	2	uint16_t	패킷 내 데이터 코드	0x0003
4	2	uint16_t	접근 모드	0x0002
6	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	장치 주소	0x01...0xfe
1	1	uint8_t	패킷 유형	0x03
2	1	uint8_t	데이터 전송 바이트 수	0x20
3	32	32 바이트	데이터 구조 (아래 참조)	
35	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

데이터 필드 형식:

오프셋	크기 (바이트)	유형	설명
0	4	uint32_t	리셋 또는 웨이크업 이후 동작 시간 (초)
4	1	uint8_t	R, 라디오 RSSI 레지스터 값 (수신 신호 강도 표시기). $R > 128$ 인 경우, $RSSI (dBm) = (R - 256) / 2 - 74$ $R \leq 128$ 인 경우, $RSSI (dBm) = (R / 2) - 74$
5	1	uint8_t	설명 없음
6	1	int8_t	Measured temperature V_t (signed). Temperature is $(V_t + 23) ^\circ C$
7	2	uint16_t	비트 0...11: 전원 공급 전압, mV 비트 12...13: 설명 없음 비트 14: 1: 저전력, 장치가 짧은 시간 후 슬립 상태로 진입 비트 15: 1: 매우 저전력, 장치가 짧은 시간 후 딥 슬립 상태로 진입
9	23	23 바이트	설명 없음

		E	
--	--	---	--

3.4.8 Marvelmind 로봇 제어 명령

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

3.4.8.1 로봇 제어 명령

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	로봇 주소	0x01...0xfe
1	1	uint8_t	패킷 유형	0x10
2	2	uint16_t	패킷 내 데이터 코드	0x1000
4	2	uint16_t	액세스 모드	0x0001
6	1	uint8_t	데이터 전송 바이트 수	0x10
7	16 (0x10) 바이트	uint8_t	로봇 제어 데이터 (아래 참조)	
23	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형 (modem 응답)	0x7f
2	2	uint16_t	데이터 코드	0x1000
4	2	uint16_t	예약됨	
6	2	uint16_t	바이트 0...5의 CRC-16 (부록 1 참조)	
8	1	uint8_t	로봇 주소	0x01...0xfe
9	1	uint8_t	패킷 유형 (로봇 응답)	0x10
10	2	uint16_t	데이터 코드	0x1000
12	2	uint16_t	reserved	
14	2	uint16_t	바이트 8...13의 CRC-16 (부록 1 참조)	

오류 응답의 형식은 부록 2에 설명되어 있습니다.

로봇 제어 데이터 형식:

오프셋	크기 (바이트)	유형	설명
0	1	uint8_t	제어 모드: 0 - 제어 없음 (대기 모드) 1 - 모터 전력 제어

			2 - 속도 제어 3 - 이동 프로그램 쓰기 4 - 이동 프로그램 일시 정지 5 - 일시 정지 후 이동 계속
1	1	uint8_t	작업 코드: 0 - 앞으로 이동 1 - 뒤로 이동 2 - 시계 방향 회전 3 - 반시계 방향 회전 4 - 지정 시간 동안 일시 정지 (모드 3용) 5 - 처음부터 이동 프로그램 반복 (모드 3용) 6 - 좌표로 지정된 지점으로 이동 (모드 3용) 7 - 이동 속도 설정 (모드 3용)
2	1	uint8_t	제어 바이트 1: 모드 1용: 모터 전원 켜기, % 모드 2용: 이동 속도, % 모드 3용: 프로그램 단계 번호 (0부터 시작)
3	2	Int16_t	프로그램용 데이터 (모드 3): 동작 코드 0 또는 1: 이동 거리, cm 동작 코드 2 또는 3: 회전 각도, degrees 동작 코드 4: 일시 정지 시간, ms 동작 코드 6: 이동 목표 지점의 X 좌표, cm 동작 코드 7: 이동 속도, %
5	1	uint8_t	모드 3의 경우: 프로그램의 총 단계 수.
6	2	int16_t	프로그램의 추가 데이터 (모드 3): 동작 코드 6: 이동 목표의 Y 좌표, cm
8	2	int16_t	동작 코드 6: 이동 목표의 Z 좌표, cm
10	6	6 bytes	예약됨 (0)

이 복잡한 명령에 대한 몇 가지 설명.

로봇 제어 구조의 바이트 0에 지정된 세 가지 주요 로봇 제어 모드가 있습니다:

- 전력 제어 (모드 1)
- 속도 제어 (모드 2)
- 프로그램에 의한 이동 (모드 3)

모드 1과 모드 2는 일반적으로 테스트 목적으로 사용됩니다. 모드 1에서 로봇은 선택된 모터 출력으로 전진, 후진, 좌회전 또는 우회전합니다. 모드 2에서 로봇은 동일한 동작을 수행하지만 선택된 속도를 유지하도록 출력을 조정합니다. 출력 또는 속도는 구조의 바이트 2에 설정되며, 이동 유형은 바이트 1에 설정됩니다.

모드 4와 모드 5는 프로그램 실행 중 이동을 일시 정지하고 일시 정지 후 이동을 재개하기 위한 특수 명령입니다.

복잡한 궤적을 따라 이동하기 위한 주요 모드는 모드 3입니다.

이 모드를 통해 로봇에 기본 동작의 시퀀스를 프로그래밍할 수 있으며, 이러한 동작들의 조합으로 궤적이 구성됩니다. 시퀀스의 각 항목은 이 유형의 명령 하나로 전송되어야 합니다. 각 명령은 로봇 제어 구조의 바이트 2에 현재 단계 번호를, 바이트 5에 총 단계 수를 포함해야 합니다.

로봇 제어 구조의 바이트 1에는 기본 이동의 유형이 지정됩니다. 기본 이동의 매개변수는 "프로그램 데이터" 필드 (바이트 3...4)와 "프로그램 추가 데이터" 필드 (바이트 6...7)에 지정됩니다.

사용 가능한 기본 동작은 다음과 같습니다:

- 주어진 거리만큼 전진;
- 주어진 거리만큼 후진;

- 주어진 각도만큼 시계 방향으로 회전;
- 주어진 각도만큼 반시계 방향으로 회전;
- 주어진 시간만큼 일시 정지;
- 첫 번째 항목부터 이동 프로그램 재시작 (반복 이동용);
- **Marvelmind** 내비게이션 시스템 좌표의 주어진 지점 (x, y)으로 이동;
- 이동 속도 변경.

로봇은 프리미티브 시퀀스를 수신한 후 프로그램 실행을 시작합니다. 프로그램 실행이 완료되면 로봇은 정지합니다. 단, 프로그램에 동작 코드 5 (처음부터 반복)에 해당하는 항목이 포함된 경우, 정지 명령을 수신하거나 새 프로그램이 업로드될 때까지 해당 루프가 무한 반복 실행됩니다.

3.4.8.2 로봇 정지

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (bytes)	유형	설명	값
0	1	uint8_t	로봇 주소	0x01...0xfe
1	1	uint8_t	패킷 유형	0x10
2	2	uint16_t	패킷 내 데이터 코드	0x403
4	2	uint16_t	액세스 모드	0x0001
6	1	uint8_t	데이터 전송 바이트 수	0x04
7	4 bytes	4 bytes	예약됨 (0)	0
11	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형 (Modem 응답)	0x7f
2	2	uint16_t	데이터 코드	0x403
4	2	uint16_t	예약됨	
6	2	uint16_t	바이트 0...5의 CRC-16 (부록 1 참조)	
8	1	uint8_t	로봇 주소	0x01...0xfe
9	1	uint8_t	패킷 유형 (로봇 응답)	0x10
10	2	uint16_t	데이터 코드	0x403
12	2	uint16_t	예약됨	
14	2	uint16_t	바이트 8...13의 CRC-16 (부록 1 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

이 명령은 로봇의 이동 실행 또는 이동 프로그램을 즉시 종료합니다. 로봇은 정지하고 새로운 명령을 기다립니다.

3.4.9 장치 제어 설정 읽기/쓰기 (펌웨어 V6.01 이상)

지원되는 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

3.4.9.1 장치 제어 설정 읽기

요청 프레임 형식 (호스트에서 **Modem**으로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	장치 주소 (beacon/modem)	0x01...0xfe 또는 0xff
1	1	uint8_t	패킷 유형	0x03
2	2	uint16_t	패킷 내 데이터 코드	0x1201
4	2	uint16_t	접근 모드	0x0001
6	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (**Modem**에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem의 주소	0xff
1	1	uint8_t	패킷 유형	요청이 Modem으로 전송된 경우 0x03 요청이 beacon으로 전송된 경우 0x7f
2	1	uint8_t	데이터 전송의 바이트 수	0x10
3	16	구조체	데이터 구조 (섹션 9.3 참조). 요청이 Modem (0xff)으로 전송된 경우에만 해당	
11	2	uint16_t	바이트 0...10의 CRC-16 (부록 1 참조)	
요청이 beacon으로 전송된 경우 다음 데이터가 수신됨				
13	1	uint8_t	장치 주소	0x01...0xfe
14	1	uint8_t	패킷 유형	0x03
15	1	uint8_t	데이터 전송 바이트 수	0x08/0x10
16	16	structure	데이터 구조 (아래 참조).	
32	2	uint16_t	바이트 12...22의 CRC-16 (부록 1 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

3.4.9.2 장치 제어 설정 쓰기

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	장치 주소 (beacon/modem)	0x01...0xfe 또는 0xff
1	1	uint8_t	패킷 타입	0x10
2	2	uint16_t	패킷 내 데이터 코드	0x1201
4	2	uint16_t	액세스 모드	0x0001
6	1	uint8_t	데이터 전송 바이트 수	0x10
7	16	구조체	데이터 구조 (아래 참조)	
23	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	요청이 Modem으로 전송된 경우 0x03 요청이 비콘으로 전송된 경우 0x7f
2	2	uint16_t	데이터 코드	0x1201
4	2	uint16_t	reserved	
6	2	uint16_t	바이트 0..5의 CRC-16 (부록 1 참조)	
요청이 비콘(주소 0x01...0xfe)으로 전송된 경우 다음 데이터가 수신됩니다				
8	1	uint8_t	장치 주소	0x01...0xfe
9	1	uint8_t	패킷 유형	0x10
10	2	uint16_t	데이터 코드	0x1201
12	2	uint16_t	reserved	
14	2	uint16_t	바이트 8..13의 CRC-16 (부록 1 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

3.4.9.3 제어 설정 페이로드 데이터 형식

오피셋	크기 (바이트)	타입	설명
0	1	uint8_t	플래그: 비트 0...5: 미정의, 항상 0이어야 함! 비트 6: 0 - 고정 비콘 모드, 1 - Hedgehog 모드 비트 7: 예약됨 (0)
1	1	uint8_t	UART 전송 속도 설정: 0: 500000 bps (기본값) 1: 4800 bps 2: 9600 bps 3: 19200 bps 4: 38400 bps 5: 57600 bps 6: 115200 bps 7...255: 예약됨
2	1	uint8_t	예약됨 (0)
3	1	uint8_t	비트 0...3: 라디오 프로파일: 0: 38.4 kbps 1: 150 kbps 2: 500 kbps 3...7: 예약됨 비트 4...6: 라디오 대역: 0: 433 MHz 1: 868 MHz 2: 915 MHz 3: 315 MHz 4...7: 예약됨 비트 7: 예약됨
4	1	uint8_t	UART/USB 출력 유형: 0: Marvelmind 프로토콜 1: NMEA0183
5	1	uint8_t	NMEA0183 모드에서 전송할 NMEA 프레임의 마스크: 비트 0: 1 - \$GPRMC 프레임 전송 비트 1: 1 - \$GPGGA 프레임 전송 비트 2: 1 - \$GPVTG 프레임 전송 비트 3: 1 - \$GPZDA 프레임 전송 비트 4...7: 예약됨 (0)
6	1	uint8_t	이 Hedgehog에서 Modem으로 전송하는 사용자 페이로드 데이터의 바이트 수 (0...32)
7	1	uint8_t	'IMU via modem' 모드에서 Modem으로 전송할 IMU 데이터의 마스크: 비트 0: IMU 융합 위치 비트 1: 쿼터니언 비트 2: 속도 비트 3: 가속도 비트 4: 원시 가속도계 비트 5: 원시 자이로 비트 6: 원시 나침반 비트 7: 0 = IMU 융합 전송, 1 = 원시 IMU 전송

8	1	uint8_t	비트 0...6: 스트리밍 텔레메트리 간격 (0 = 스트림 없음) 비트 7: 예약됨 (0)
9	1	uint8_t	비트 0: 속도 계산에 IMU 사용 비트 1...7 - 예약됨 (0)
10	6	6 바이트	예약됨 (0)

경고! 라디오로 연결된 비콘의 라디오 프로파일을 변경하면 라디오 연결이 끊어집니다. 프로파일을 전환해야 하는 경우, 모든 비콘에 대해 라디오 프로파일을 하나씩 순서대로 전환한 다음 **Modem**의 라디오 프로파일을 전환하십시오. 모든 비콘은 몇 초 후에 새 라디오 프로파일에서 사용 가능해야 합니다.

3.4.10 네트워크 내 장치 목록 읽기 (펌웨어 V6.01+)

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem의 주소	0xff
1	1	uint8_t	패킷 타입	0x03
2	2	uint16_t	패킷 내 데이터 코드	0x31xx (xx는 장치 그룹 번호)
4	2	uint16_t	접근 모드	0x0000
6	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem의 주소	0xff
1	1	uint8_t	패킷 타입	0x03
2	1	uint8_t	전송 데이터의 바이트 수	0x72
3	1	uint8_t	네트워크 내 장치의 총 수 (K)	
4	112	112 바이트	네트워크 내 장치 정보의 0...16 구조체, 아래 설명 참조	
116	1	uint8_t	예약됨	0x00
117	2	uint16_t	CRC-16 (부록 1 참조)	

요류 응답 형식은 부록 2에 설명되어 있습니다.

네트워크 내 장치 데이터 형식 (7바이트)

오프셋	크기 (바이트)	설명

	트)	
0	1	장치 주소 (0x01...0xfe)
1	1	펌웨어 메이저 버전
2	1	펌웨어 마이너 버전
3	1	비트 0...5: 장치 유형: 10: 휠 로봇 12: 크롤러 로봇 22: Beacon HW V4.5 23: Beacon HW V4.5 (Hedgehog 모드) 24: Modem (HW V4.5/4.9) 30: Beacon HW V4.9 31: Beacon HW V4.9 (Hedgehog 모드) 32: Mini-RX beacon 36: Mini TX 비콘 (HW V5.07) 37: Industrial-TX 비콘 41: Industrial-RX 비콘 42: Super-Beacon 43: Super-Beacon (Hedgehog 모드) 44: Industrial Super-Beacon 45: Industrial Super-Beacon (Hedgehog 모드) 비트 6: 1 - 이 주소를 가진 장치가 두 개 이상 존재함 비트 7: 1 - 슬리핑 모드
4	1	펌웨어 두 번째 마이너 버전

3.4.11 펌웨어 버전 읽기

지원되는 하드웨어:

- Super-Beacon: 지원됨
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원됨
- Mini-TX: 지원됨
- Mini-TX-2: 지원됨
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원됨
- Beacon HW4.5: 지원됨

요청 프레임 형식 (호스트에서 **Modem**으로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	장치 주소	0xff
1	1	uint8_t	패킷 타입	0x03
2	2	uint16_t	패킷 내 데이터 코드	0xfe00
4	2	uint16_t	접근 모드	0x0000
6	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (**Modem**에서 호스트로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x03
2	1	uint8_t	전송 데이터의 바이트 수	0x08
3	1	uint8_t	펌웨어 마이너 버전	
4	1	uint8_t	펌웨어 주 버전	
5	3	3 바이트	예약됨	
8	1	uint8_t	장치 유형 ID	
9	2	uint16_t	예약됨	
11	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

3.4.12 사용자 데이터 읽기

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

요청 프레임 형식 (호스트에서 **Modem**으로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x03
2	2	uint16_t	패킷 내 데이터 코드	0x0004
4	2	uint16_t	접근 모드	0x0000
6	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (**Modem**에서 호스트로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x03
2	1	uint8_t	전송되는 데이터 바이트 수	0x84
3	1	uint8_t	총 사용자 데이터 크기	
4	3	3 bytes	예약됨 (0)	
7	128	uint8_t	Hedgehog의 사용자 데이터	
135	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답의 형식은 부록 2에 설명되어 있습니다.

Hedgehog의 사용자 데이터는 다음 구조를 가진 레코드의 시퀀스입니다:

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Hedgehog의 주소	H
1	1	uint8_t	Hedgehog의 사용자 데이터 바이트 수	M
2	M	uint8_t	Hedgehog H로부터의 M 바이트 데이터	

3.4.13 수동 장치 위치 쓰기

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

요청 프레임 형식 (호스트에서 Modem으로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	비콘의 주소	0x01...0xfe
1	1	uint8_t	패킷 유형	0x10
2	2	uint16_t	패킷 내 데이터 코드	0x5003
4	2	uint16_t	액세스 모드	0x0002
6	1	uint8_t	데이터 전송 바이트 수	0x20
7	32	structure	데이터 구조 (아래 참조)	
39	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Modem의 주소	0xff
1	1	uint8_t	패킷 유형	0x10
2	2	uint16_t	데이터 코드	0x5003
4	2	uint16_t	예약됨	
6	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

데이터 구조 형식:

오프셋	크기 (바이트)	유형	설명	값
0	4	int32_t	X 좌표, mm	
4	4	int32_t	Y 좌표, mm	
8	4	int32_t	Z 좌표, mm	
12	1	uint8_t	설명 없음	0xff
13	4	int32_t	설명 없음	0
17	4	int32_t	설명 없음	0
21	4	int32_t	설명 없음	0
25	1	uint8_t	설명 없음	0x02
26	6	6 bytes	예약됨	0

3.4.14 비콘 간 수동 거리 쓰기

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원됨
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원됨
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

요청 프레임 형식 (호스트에서 모뎀으로)

오프셋	크기 (bytes)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x10
2	2	uint16_t	패킷 내 데이터 코드	0x4003
4	2	uint16_t	액세스 모드	0x0000
6	1	uint8_t	데이터 전송 바이트 수	0x10
7	16	structure	데이터 구조 (아래 참조)	
23	2	uint16_t	CRC-16 (부록 1 참조)	

응답 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem 주소	0xff
1	1	uint8_t	패킷 유형	0x10
2	2	uint16_t	데이터 코드	0x4003
4	2	uint16_t	예약됨	
6	2	uint16_t	CRC-16 (부록 1 참조)	

오류 응답 형식은 부록 2에 설명되어 있습니다.

데이터 구조 형식:

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	첫 번째 비콘의 주소	
1	1	uint8_t	두 번째 비콘의 주소	
2	4	uint32_t	비콘 간 거리, mm	
6	10	10 바이트	예약됨	0

4. RS-485를 통한 통신 프로토콜

4.1 스트리밍을 위한 'Marvelmind' 프로토콜

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

UART 해당 섹션에 설명된 모든 패킷은 RS-485를 통해서도 사용할 수 있습니다.

이 데이터는 RS-485 하드웨어가 내장된 Super-Modem 및 Industrial Super-Beacon에서만 사용할 수 있습니다.

4.2 사용자 장치와의 데이터 읽기/쓰기 프로토콜

지원되는 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 요청 시 지원
- Modem HW5.1: 지원되지 않음
- Super-Modem: 요청 시 지원
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

UART에 해당하는 섹션에 설명된 모든 패킷은 요청 시 구현할 수 있습니다.

이 데이터는 RS-485 하드웨어가 내장된 Super-Modem 및 Industrial Super-Beacon에서만 사용할 수 있습니다.

4.3 NMEA0183 통신 프로토콜

지원되는 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨 (SW V7.000부터)
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

UART에 해당하는 섹션에 설명된 모든 패킷은 RS-485를 통해서도 사용할 수 있습니다.

이 데이터는 RS-485 하드웨어가 내장된 Super-Modem 및 Industrial Super-Beacon에서만 사용할 수 있습니다.

5. SPI를 통한 통신 프로토콜

5.1 Hedgehog 위치 정보가 포함된 패킷

지원되는 하드웨어:

- Super-Beacon: 지원됨
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 요청 시 지원
- Super-Modem: 지원되지 않음
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원됨
- Beacon HW4.5: 지원됨

Super-Beacon, Beacon HW4.9 및 Beacon HW4.5는 SPI 슬레이브 장치로 작동할 수 있으며, Hedgehog 위치 데이터가 포함된 패킷 읽기를 지원합니다. Modem HW5.1은 하드웨어 SPI를 지원하며, 소프트웨어 지원은 요청 시 추가할 수 있습니다.

5.2 SPI를 통한 기타 데이터

지원되는 하드웨어:

- Super-Beacon: 요청 시 지원
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 요청 시 지원
- Super-Modem: 지원되지 않음
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

챕터 2에 설명된 기타 데이터 패킷 지원은 Super-Beacon 및 Modem HW5.1에 대해 요청 시 추가될 수 있습니다

6. I2C를 통한 통신 프로토콜

6.1 PX4가 탑재된 드론을 위한 나침반 에뮬레이션

지원 하드웨어:

- Super-Beacon: 지원됨
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원되지 않음
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

페어링된 Super-Beacon은 I2C를 통해 PX4에 연결된 보다 안정적이고 정밀한 나침반으로 작동할 수 있습니다.

이를 위해 MMSW0003 라이선스를 구매하셔야 합니다.

6.2 I2C를 통한 기타 데이터

지원 하드웨어:

Super-Beacon: 요청 시 지원

Industrial Super-Beacon: 미지원

Modem HW5.1: 요청 시 지원

Super-Modem: 미지원

Mini-RX (Badge, Helmet 등): 미지원

Mini-TX: 미지원

Mini-TX-2: 미지원

Modem HW4.9: 미지원

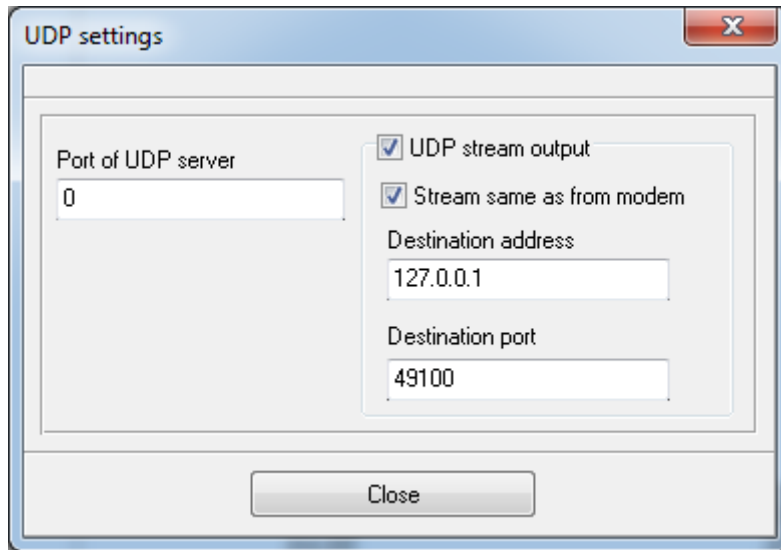
Beacon HW4.9: 미지원

Beacon HW4.5: 미지원

2장에 설명된 기타 데이터 패킷에 대한 지원은 Super-Beacon 및 Modem HW5.1에 대해 요청 시 추가될 수 있습니다

7. UDP (Wi-Fi)를 통한 통신 프로토콜

Dashboard 소프트웨어는 Dashboard가 실행 중인 PC의 네트워크 인터페이스를 통해 UDP로 데이터를 전송할 수 있습니다. 목적지 IP 주소/포트는 File/UDP settings 메뉴를 통해 조정할 수 있습니다:



Super-Modem은 Wi-Fi를 내장하고 있으며, 이동 비콘의 위치 및 아래에 설명된 기타 데이터를 스트리밍할 수 있습니다.

Wi-Fi 네트워크 및 UDP 스트리밍 설정은 Dashboard의 Super-Modem 설정을 통해 가능합니다:

Parameters of radio	(+) expand
Ultrasound	(+) expand
Interfaces	(+) expand
Georeferencing	(+) expand
Wi-Fi/UDP settings	(-) WirelessNet-80
Wi-Fi	enabled
Wi-Fi network name	WirelessNet-80
Wi-Fi network password	-----
Show password	disabled
<input checked="" type="checkbox"/> Wi-Fi reconnect timeout, sec (10..65000)	120
Static IP	disabled
Static IP address	n/a
Router IP address	n/a
Wi-Fi RSSI, dBm	-62
Own IP address	192.168.100.12
UDP destination IP address	192.168.100.23
UDP destination port (0..65535)	49100
UDP port for API (0..65535)	49213
Stationary beacons visible	enabled
Service zones visible	enabled
Service zones active	enabled

7.1 Hedgehog 위치 패킷

지원 하드웨어/소프트웨어:

- Super-Beacon: 미지원
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

패킷 형식

오프셋	크기	유형	설명	값
0	1	uint8_t	Beacon의 주소	
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0011
4	1	uint8_t	데이터 크기 (바이트)	N
5	4	uint32_t	타임스탬프 - 좌표 수신 시점에서 Dashboard/Super-Modem 실행 후 경과 시간 (밀리초)	
9	4	int32_t	비콘의 X 좌표, mm	
13	4	int32_t	비콘의 Y 좌표, mm	
17	4	int32_t	비콘의 Z 좌표, mm	
21	1	uint8_t	플래그 바이트: 비트 0: 1 - 좌표 사용 불가. X, Y, Z 필드의 데이터를 사용하지는 않습니다. 비트 1: 타임스탬프 단위 표시자 (참고 사항 참조) 비트 2...6: 예약됨 (0) 비트 7: -1 - 지오펜싱 구역 외부	
22	1	uint8_t	예약됨 (0)	
23	2	uint16_t	비트 0...11: XY 평면에서 Hedgehog 쌍의 방향, 소수점 각도 (0...3600) 비트 12: 1 - 좌표가 비콘 쌍의 중심에 대해 주어짐; 0 - 지정된 Hedgehog에 대한 좌표 비트 13...15: 예약됨 (0)	
25	2	예약됨 (0)		
27	M=N-22		선택적 데이터 필드 - 목록 참조	
27+M	2	예약됨 (0)		

참고: V6.290 이전 버전의 Dashboard 및 Super-Modem의 경우 타임스탬프는 1/64초 단
위이며 타임스탬프 단위 표시자(플래그 바이트의 비트 1)는 0입니다. V6.290 이상 버전의 경
우 타임스탬프는 밀리초 단위이며 타임스탬프 단위 표시자는 1입니다.

모바일 비콘 위치 패킷의 선택적 데이터에는 다음 구조가 포함될 수 있습니다:

- 속도 데이터 (7바이트). Dashboard의 모바일 비콘 설정 인터페이스 섹션에서 활성화해야 합니다.

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	데이터 필드 코드 = 1은 속도 벡터를 의미함	1
1	2	int16_t	X축 방향 속도, mm/sec	
3	2	int16_t	Y축 방향 속도, mm/sec	
5	2	int16_t	Z축 방향 속도, mm/sec	

7.1.1. 실시간 타임스탬프가 포함된 Hedgehog 위치 패킷 (펌웨어 v7.200 이상)

지원 하드웨어/소프트웨어:

- Super-Beacon: 미지원
- Industrial Super-Beacon: 미지원
- Modem HW5.1: 미지원
- Super-Modem: 지원
- Mini-RX (Badge, Helmet 등): 미지원
- Mini-TX: 미지원
- Mini-TX-2: 미지원
- Modem HW4.9: 미지원
- Beacon HW4.9: 미지원
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

패킷 형식

오프셋	크기	유형	설명	값
0	1	uint8_t	Beacon의 주소	
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0081
4	1	uint8_t	데이터 크기 (바이트)	N
5	8	int64_t	Timestamp – 비콘 초음파 발신의 Unix 시간, 1970.01.01 00:00:00부터의 밀리 초 수. Modem 및 Dashboard와 모든 장치에 의해 동기화된 시간.	
13	4	int32_t	비콘의 X 좌표, mm	
17	4	int32_t	비콘의 Y 좌표, mm	
21	4	int32_t	비콘의 Z 좌표, mm	
25	1	uint8_t	플래그 바이트: 비트 0: 1 - 좌표를 사용할 수 없음. X, Y, Z 필드의 데이터를 사용해서는 안 됨. 비트 1: timestamp 단위 표시자 (비교 참조) 비트 2...6: 예약됨 (0) 비트 7: -1 - 지오펜싱 구역 이탈	
26	1	uint8_t	예약됨 (0)	
27	2	uint16_t	비트 0...11: XY 평면에서의 Hedgehog 쌍 방향, 0.1도 단위 (0...3600) 비트 12: 1 - 비콘 쌍의 중심에 대한 좌표; 0 - 지정된 Hedgehog에 대한 좌표 비트 13...15: 예약됨 (0)	
29	2	예약됨 (0)		
31	M=N-26		선택적 데이터 필드 - 목록 참조	
31+M	2	예약됨 (0)		

7.2. 고정 비콘 위치 포함 패킷

지원되는 하드웨어/소프트웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

패킷 형식

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 타입	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0012
4	1	uint8_t	전송 데이터 바이트 수	1+N*14
5	1	uint8_t	패킷 내 비콘 수	N
6	1	N*14 바이트	N개 비콘에 대한 데이터	

N개 비콘 각각에 대한 데이터 구조 형식:

오프셋	크기 (바이트)	타입	설명
0	1	uint8_t	비콘의 주소
1	4	int32_t	비콘의 X 좌표, mm
5	4	int32_t	비콘의 Y 좌표, mm
9	4	int32_t	비콘의 Z 좌표, mm
13	1	uint8_t	예약됨 (0)

7.3. 원시 IMU 데이터가 포함된 패킷

지원되는 HW/SW:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Beacon의 주소	
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0003
4	1	uint8_t	전송되는 데이터의 바이트 수	
5	2	int16_t	가속도계, X축, 1 mg/LSB	
7	2	int16_t	가속도계, Y축, 1 mg/LSB	
9	2	int16_t	가속도계, Z축, 1 mg/LSB	
11	2	int16_t	자이로스코프, X축, 0.0175 dps/LSB	
13	2	int16_t	자이로스코프, Y축, 0.0175 dps/LSB	
15	2	int16_t	자이로스코프, Z축, 0.0175 dps/LSB	
17	2	int16_t	나침반, X축, 1100 LSB/Gauss	
19	2	int16_t	나침반, Y축, 1100 LSB/Gauss	
21	2	int16_t	나침반, Z축, 980 LSB/Gauss	
23	1	uint8_t	비콘의 주소	
24	5	5 바이트	예약됨 (0)	
29	4	uint32_t	타임스탬프, ms	
33	8	8 바이트	예약됨	

참고: 나침반 데이터는 IMU가 탑재된 HW v4.9 비콘에서만 사용 가능합니다.

7.3.1. 실시간 타임스탬프가 포함된 원시 IMU 데이터 패킷 (펌웨어 v7.200 이상)

지원되는 하드웨어/소프트웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Beacon의 주소	
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0083
4	1	uint8_t	전송 데이터 바이트 수	
5	2	int16_t	가속도계, X축, 1 mg/LSB	
7	2	int16_t	가속도계, Y축, 1 mg/LSB	
9	2	int16_t	가속도계, Z축, 1 mg/LSB	
11	2	int16_t	자이로스코프, X축, 0.0175 dps/LSB	
13	2	int16_t	자이로스코프, Y축, 0.0175 dps/LSB	
15	2	int16_t	자이로스코프, Z축, 0.0175 dps/LSB	
17	2	int16_t	나침반, X축, 1100 LSB/Gauss	
19	2	int16_t	나침반, Y축, 1100 LSB/Gauss	
21	2	int16_t	나침반, Z축, 980 LSB/Gauss	
23	1	uint8_t	비콘의 주소	
24	5	5 바이트	예약됨 (0)	
29	8	int64_t	타임스탬프 - 유닉스 시간, 1970.01.01 00:00:00부터의 밀리초 수. Modem 및 Dashboard와 모든 장치가 동기화한 시간.	
37	8	8 바이트	예약됨	

참고: 나침반 데이터는 IMU가 탑재된 HW v4.9 비콘에서만 사용 가능합니다.

7.4. 원시 거리 데이터가 포함된 패킷

지원 하드웨어/소프트웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

패킷 형식

오피셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0004
4	1	uint8_t	전송 데이터의 바이트 수	
5	32		데이터 패킷 (하단 참조)	

데이터 패킷 형식

오피셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Hedgehog의 주소	
1	6		거리 항목 1	
7	6		거리 항목 2	
13	6		거리 항목 3	
19	6		거리 항목 4	
25	4	uint32_t	타임스탬프 - 비콘 초음파 방출의 내부 시간, 최근 웨이크업 이벤트 시점으로부터의 경과 시간 (밀리초 단위) (V5.89+)	
29	2	uint16_t	초음파 방출 시점부터 현재 시간까지 경과된 시간, 밀리초 (V5.89+)	
31	1	uint8_t	예약됨	

거리 항목의 형식

오피셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	비콘 주소 (항목이 채워지지 않은 경우 0)	
1	4	uint32_t	비콘까지의 거리, mm	

5	1	uint8_t	예약됨 (0)	
---	---	---------	---------	--

7.4.1. 실시간 타임스탬프가 포함된 원시 거리 데이터 패킷 (펌웨어 v7.200+)

지원 HW/SW:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0084
4	1	uint8_t	전송되는 데이터의 바이트 수	
5	32		데이터 패킷 (아래 참조)	

데이터 패킷 형식

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	Hedgehog 주소	
1	6		거리 항목 1	
7	6		거리 항목 2	
13	6		거리 항목 3	
19	6		거리 항목 4	
25	8	int64_t	타임스탬프 - 비콘 초음파 방출의 유닉스 시간, 1970.01.01 00:00:00 기준 밀리초 수 모든 장치가 Modem 및 Dashboard와 동기화된 시간.	
33	2	uint16_t	초음파 방출부터 현재 시간까지 경과된 시간, 밀리초 (V5.89+)	
35	1	uint8_t	예약됨	

거리 항목의 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Beacon의 주소 (항목이 채워지지 않은 경우 0)	

1	4	uint32_t	Beacon까지의 거리, mm	
5	1	uint8_t	예약됨 (0)	

7.5. IMU 융합 데이터가 포함된 패킷

지원되는 HW/SW:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Beacon의 주소	
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0005
4	1	uint8_t	전송 데이터의 바이트 수	
5	4	int32_t	비콘의 X 좌표 (fusion), mm	
9	4	int32_t	비콘의 Y 좌표 (fusion), mm	
13	4	int32_t	비콘의 Z 좌표 (fusion), mm	
17	2	int16_t	회전 쿼터니언의 W 필드	
19	2	int16_t	회전 쿼터니언의 X 필드	
21	2	int16_t	회전 쿼터니언의 Y 필드	
23	2	int16_t	회전 쿼터니언의 Z 필드	
25	2	int16_t	Beacon의 속도 X (융합), mm/s	
27	2	int16_t	Beacon의 속도 Y (융합), mm/s	
29	2	int16_t	Beacon의 속도 Z (융합), mm/s	
31	2	int16_t	Beacon의 가속도 X, mm/s ²	
33	2	int16_t	Beacon의 가속도 Y, mm/s ²	
35	2	int16_t	Beacon의 가속도 Z, mm/s ²	
37	1	uint8_t	Beacon의 주소	
38	1	1 byte	예약됨 (0)	
39	4	uint32_t	타임스탬프, ms	
43	4	4 bytes	예약됨 (0)	

참고: 쿼터니언은 10000 값으로 정규화됩니다.

7.5.1. 실시간 타임스탬프가 포함된 IMU 융합 데이터 패킷 (펌웨어 v7.200 이상)

지원되는 하드웨어/소프트웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

타임스탬프에 관한 참고 사항을 확인하십시오.

패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Beacon의 주소	
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0085
4	1	uint8_t	전송 데이터의 바이트 수	
5	4	int32_t	Beacon의 X 좌표 (융합), mm	
9	4	int32_t	Beacon의 Y 좌표 (융합), mm	
13	4	int32_t	Beacon의 Z 좌표 (융합), mm	
17	2	int16_t	회전 쿼터니언의 W 필드	
19	2	int16_t	회전 쿼터니언의 X 필드	
21	2	int16_t	회전 쿼터니언의 Y 필드	
23	2	int16_t	회전 쿼터니언의 Z 필드	
25	2	int16_t	비콘의 속도 X (융합), mm/s	
27	2	int16_t	비콘의 속도 Y (융합), mm/s	
29	2	int16_t	비콘의 속도 Z (융합), mm/s	
31	2	int16_t	비콘의 가속도 X, mm/s ²	
33	2	int16_t	비콘의 가속도 Y, mm/s ²	
35	2	int16_t	비콘의 가속도 Z, mm/s ²	
37	1	uint8_t	비콘의 주소	
38	1	1 바이트	예약됨 (0)	
39	8	int64_t	타임스탬프 – 유닉스 시간, 1970.01.01 00:00:00 기준 밀리초 수 Modem 및 Dashboard와 모든 장치가 동기화한 시간	
47	4	4 바이트	예약됨 (0)	

참고: 쿼터니언은 10000 값으로 정규화됩니다.

7.6. 텔레메트리 데이터가 포함된 패킷

지원되는 HW/SW:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

패킷 형식

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	비콘의 주소	
1	1	uint8_t	패킷 타입	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0006
4	1	uint8_t	전송되는 데이터의 바이트 수	
5	2	uint16_t	배터리 전압, mV	
7	1	int8_t	RSSI, dBm	
8	13		예약됨 (0)	

7.7. 품질 및 확장 위치 데이터가 포함된 패킷

지원되는 하드웨어/소프트웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

패킷 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	주소	
1	1	uint8_t	패킷 유형	0x47
2	2	uint16_t	패킷 내 데이터 코드	0x0007
4	1	uint8_t	전송 데이터의 바이트 수	
5	1	uint8_t	장치 주소	
6	1	uint8_t	측위 품질, %	
7	1	uint8_t	0 = 지오펜싱 구역 알람 없음 1...255 - 지오펜싱 구역 인덱스 이 필드는 MMSW0005 라이선스가 필요합니다.	
8	13		예약됨 (0)	

7.8. 모든 비콘의 텔레메트리 패킷

지원 하드웨어/소프트웨어:

- Super-Beacon: 미지원
- Industrial Super-Beacon: 미지원
- Modem HW5.1: 미지원
- Super-Modem: 지원 (SSM 펌웨어)
- Mini-RX (Badge, Helmet 등): 미지원
- Mini-TX: 미지원
- Mini-TX-2: 미지원
- Modem HW4.9: 미지원
- Beacon HW4.9: 미지원
- Beacon HW4.5: 미지원
- Dashboard 소프트웨어: 요청 시 제공

패킷 형식

오프셋	크기 (바이트)	타입	설명	값
0	1	uint8_t	주소	0xff
1	1	uint8_t	패킷 유형	0x48
2	2	uint16_t	패킷 내 데이터 코드	0x2001
4	2	uint16_t	전송 데이터의 바이트 수	
6	N*10		N번째 Beacon의 텔레메트리 (아래 표 참조)	

Beacon 텔레메트리 항목의 형식

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Beacon의 주소	
1	2	uint16_t	전원 전압, mV	
3	1	int8_t	RSSI, dBm	
4	4	uint32_t	마지막 데이터 업데이트 이후 경과 시간, sec	
8	2	uint16_t	예약됨 (0)	

7.9. NMEA0183 프로토콜

지원되는 하드웨어/소프트웨어:

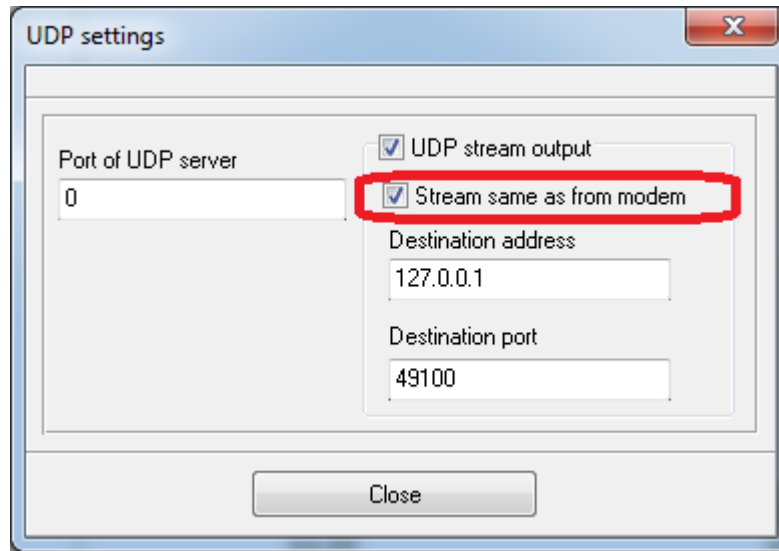
- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원되지 않음
- Modem HW5.1: 지원되지 않음
- Super-Modem: 지원됨
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음
- Dashboard 소프트웨어: 지원됨

패킷 형식

오피셋	크기 (바이트)	유형	설명	값
0	N	N 바이트	NMEA0183 메시지 (여기에서 설명 참조)	
N	1	uint8_t	모바일 비콘의 주소	

참고 사항:

- Super-Modem은 Dashboard의 설정에서 Super-Modem에 대한 인터페이스 섹션에서 NMEA0183 프로토콜이 선택된 경우 UDP를 통해 NMEA0183 메시지를 스트리밍합니다
- Dashboard는 모뎴에 대한 설정의 인터페이스 섹션에서 NMEA0183 프로토콜이 선택되고, Dashboard의 Settings/UDP Settings 메뉴에서 아래 스크린샷에 표시된 옵션이 선택된 경우 NMEA0183 메시지를 스트리밍합니다:



8. CAN을 통한 통신 프로토콜

지원 하드웨어:

- Super-Beacon: 지원되지 않음
- Industrial Super-Beacon: 지원됨
- Modem HW5.1: 지원되지 않음
- Super-Modem: 요청 시 제공
- Mini-RX (Badge, Helmet 등): 지원되지 않음
- Mini-TX: 지원되지 않음
- Mini-TX-2: 지원되지 않음
- Modem HW4.9: 지원되지 않음
- Beacon HW4.9: 지원되지 않음
- Beacon HW4.5: 지원되지 않음

CAN 하드웨어 지원은 요청에 따라 Super-Modem 및 Industrial Super-Beacon에 설치할 수 있습니다. CAN이 설치된 경우 RS-485는 사용할 수 없습니다.

CAN 파라미터:

전송 속도: 125 kbps.

프레임 형식: 표준.

8.1. 스트리밍의 'Marvelmind' 프로토콜

UART 스트리밍에 관한 해당 챕터에서 설명된 패킷은 CAN 프레임 ID 0x10으로 CAN을 통해서도 전송됩니다. 각 CAN 프레임은 1~8바이트의 데이터를 포함할 수 있습니다. 데이터 바이트 수는 CAN 프레임의 DLC 필드에 지정됩니다.

데이터는 원시 스트림으로 전송되므로, CAN 프레임에는 하나의 데이터 패킷의 끝과 다음 패킷의 시작이 포함될 수 있습니다. 사용자는 여러 CAN 프레임을 수신하여 해당 데이터 필드를 버퍼에 저장하고, UART에서 수신한 데이터와 동일한 방식으로 처리해야 합니다.

8.2. NMEA0183 통신 프로토콜

UART 스트리밍에 관한 해당 챕터에서 설명된 패킷은 CAN 프레임 ID 0x11로 CAN을 통해서도 전송됩니다. 각 CAN 프레임은 1~8바이트의 데이터를 포함할 수 있습니다. 데이터 바이트 수는 CAN 프레임의 DLC 필드에 지정됩니다.

데이터는 원시 스트림으로 전송되므로, CAN 프레임에는 하나의 데이터 패킷의 끝과 다음 패킷의 시작이 포함될 수 있습니다. 사용자는 여러 CAN 프레임을 수신하여 해당 데이터 필드를 버퍼에 저장하고, 다음과 동일한 방식으로 처리해야 합니다.

9. Dashboard csv 로그 파일 형식

Dashboard는 고정 및 이동 비콘의 위치와 기타 데이터를 Dashboard 디렉터리 내 'log' 폴더에 위치한 csv 로그 파일에 저장합니다. v7.000 버전부터 로그 형식이 변경되었습니다. 이전 형식은 Modem HW v4.9에 대해서만 유지됩니다.

9.1. csv 로그 파일 형식 (Dashboard 버전 V7.000+)

Dashboard 버전 V7.000+의 csv 로그 파일에서 각 이벤트는 하나의 csv 라인으로 로그에 기록되며, 이벤트 유형에 따라 라인 형식이 다릅니다. 동시에, 라인의 시작 부분은 모든 라인 유형에서 동일합니다.

다음은 csv 로그 파일에서 발췌한 몇 가지 라인의 예시입니다:

```
T2021_11_04__173001_581,user,41,17,14,4.675,2.714,0.250,2,975,100
T2021_11_04__173001_581,user,41,17,15,4.665,2.708,0.250,2,975,114
T2021_11_04__173001_581,user,41,17,26,4.073,1.987,0.250,2,3462,128
T2021_11_04__173001_581,user,41,17,27,4.075,1.987,0.250,2,3462,141
T2021_11_04__173001_581,user,41,17,28,3.588,1.979,0.250,2,3496,155
T2021_11_04__173001_581,user,41,17,29,3.592,1.978,0.250,2,3496,169
T2021_11_04__173001_701,user,43,15,nl
T2021_11_04__173001_728,user,43,27,nl
T2021_11_04__173001_756,user,43,29,nl
```

라인의 공통 부분은 첫 3개의 필드로 구성됩니다:

- "T2021_11_04__173001_581" - 해당 라인 데이터의 타임스탬프: 2021.11.04, 17:30:01.581;
- "user" - 사용자 이름 (향후 사용을 위해 예약됨). 향후 버전의 Dashboard에서는 사용자 로그인 기능을 지원할 예정입니다;
- "41" - 라인 유형의 ID. 라인 유형이 다르면 이후 필드의 형식도 다릅니다.

데이터 필드에는 다음과 같은 공통 특수 코드가 있습니다:

- "nl" - 라이선스 없음. 해당 필드를 채우려면 특정 라이선스가 필요합니다;
- "na" - 해당 없음. 해당 필드에 관련 데이터가 없습니다. 예를 들어, 이동형 비콘의 위치를 성공적으로 파악하지 못한 경우 X, Y, Z 좌표 필드에 "na"가 표시됩니다.

다음 장에서는 다양한 라인 유형에 대한 설명을 다룹니다.

9.1.1. 라인 유형 ID 01 – 맵 파일 링크

이 라인은 맵 파일이 자동으로 저장되거나 사용자가 'Save map' 버튼을 눌러 저장될 때 기록됩니다.

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	01 - 라인 유형 ID (맵 파일 링크)
3	해당 시점에 저장된 맵 파일의 이름

9.1.2. 라인 유형 ID 41 – Marvelmind 프로토콜 스트리밍 레코드

이 라인은 'Interfaces' 섹션의 모뎀 프로토콜 설정이 'Marvelmind'일 때 기록됩니다.

[Marvelmind 프로토콜은 다양한 유형의 레코드를 가지며, 이는 다음 하위 챕터에서 설명하는 로그 파일의 각기 다른 라인에 해당합니다.](#)

9.1.2.1. Hedgehog 위치 (41 17) 또는 (41 129)

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	41 - 라인 유형 ID (Marvelmind 프로토콜 스트리밍)
3	17 (0x0011) – Hedgehog 위치에 대한 데이터 코드 129 (0x0081) – Hedgehog 위치에 대한 데이터 코드 (실시간 타임스탬프 스트리밍 포함)
4	Hedgehog 주소
5	Hedgehog X 좌표, 미터
6	Hedgehog Y 좌표, 미터
7	Hedgehog Z 좌표, 미터
8	플래그: 비트 0: 1 - 좌표 사용 불가. X, Y, Z 필드의 데이터를 사용하지 않아야 합니다. 비트 1...6: 예약됨 비트 7: - 1 - 지오펜싱 구역 외부
9	요각 및 플래그: 비트 0...11: Hedgehog 쌍의 요각, 데시도(0...3600) 비트 12: 1 - 좌표가 비콘 쌍의 중심에 대해 제공됨; 0 - 지정된 Hedgehog에 대한 좌표
10	시간 이동, ms. 초음파 방출부터 해당 줄의 위치 계산까지 경과된 시간

9.1.2.2. 고정 비콘 위치 (41 18)

줄의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	41 - 줄 유형 ID (Marvelmind 프로토콜 스트리밍)
3	18 (0x0012) - 고정 비콘 위치에 대한 데이터 코드
4	고정 비콘 주소
5	비콘 x 좌표, 미터
6	비콘 y 좌표, 미터
7	비콘 z 좌표, 미터
8	예약 필드

9.1.2.3. Hedgehog에서 고정 비콘까지의 원시 거리 (41 4) 또는 (41 132)

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	41 - 라인 유형 ID (Marvelmind 프로토콜 스트리밍)
3	4 (0x0004) - 원시 거리에 대한 데이터 코드 132 (0x0084) - 원시 거리에 대한 데이터 코드 (실시간 타임스탬프 스트리밍 포함)
4	Hedgehog의 주소
5	N - 라인의 거리 수
6	N개의 거리 하위 레코드 (2*N 필드), 아래 참조
6+N*2+1	시간 편차, ms. 초음파 송출부터 거리 측정까지 경과된 시간

거리 하위 레코드의 필드:

0	고정 비콘의 주소
1	고정 비콘까지의 거리

9.1.2.4. 원시 IMU 데이터 (41 3) 또는 (41 131)

이 줄은 MMSW0005 라이선스가 필요합니다.

줄의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	41 - 줄 유형 ID (Marvelmind 프로토콜 스트리밍)
3	3 (0x0003) - 원시 IMU 데이터의 데이터 코드 131 (0x0083) - 원시 IMU 데이터의 데이터 코드 (Hedgehog에 대한 실시간 타임스탬프 활성화)
4	Hedgehog의 주소
5	가속도계, X축, 1 mg/LSB
6	가속도계, Y축, 1 mg/LSB
7	가속도계, Z축, 1 mg/LSB
8	자이로스코프, X축, 0.0175 dps/LSB
9	자이로스코프, Y축, 0.0175 dps/LSB
10	자이로스코프, Z축, 0.0175 dps/LSB
11	나침반, X축, 1100 LSB/Gauss
12	나침반, Y축, 1100 LSB/Gauss
13	나침반, Z축, 980 LSB/Gauss

9.1.2.5. IMU 융합 데이터 (41 5) 또는 (41 133)

이 라인에는 MMSW0005 라이선스가 필요합니다.

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	41 - 라인 유형 ID (Marvelmind 프로토콜 스트리밍)
3	5 (0x0005) - IMU 융합 데이터의 데이터 코드 133 (0x0085) - IMU 융합 데이터의 데이터 코드 (Hedgehog에 대해 실시간 타임스탬프 활성화됨)
4	Hedgehog의 주소
5	비콘의 x 좌표 (융합), 미터
6	비콘의 y 좌표 (융합), 미터
7	비콘의 z 좌표 (융합), 미터
8	회전 쿼터니언의 w 필드
9	회전 쿼터니언의 x 필드
10	회전 쿼터니언의 y 필드
11	회전 쿼터니언의 z 필드
12	Beacon의 속도 X (융합), mm/s
13	Beacon의 속도 Y (융합), mm/s
14	Beacon의 속도 Z (융합), mm/s
15	Beacon의 가속도 X, mm/s ²
16	Beacon의 가속도 Y, mm/s ²
17	Beacon의 가속도 Z, mm/s ²

9.1.2.6. 텔레메트리 데이터 (41 6)

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	41 - 라인 유형 ID (Marvelmind 프로토콜 스트리밍)
3	6 (0x0006) - 텔레메트리 데이터의 데이터 코드
4	Beacon의 주소
5	공급 전압, V
6	RSSI, dBm

9.1.2.7. 품질 및 확장 위치 데이터 (41 7)

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	41 - 라인 유형 ID (Marvelmind 프로토콜 스트리밍)
3	7 (0x0007) - 품질 및 확장 위치 데이터의 데이터 코드
4	Hedgehog의 주소
5	위치 품질, %
6	지오펜싱 구역 번호 (이 필드는 MMSW0005 라이선스 필요)

9.1.3. 라인 유형 ID 42 – NMEA0183 스트리밍 레코드

이 라인은 MMSW0005 라이선스가 필요합니다.

이 라인은 'Interfaces' 섹션의 Modem 프로토콜 설정이 'NMEA0183'인 경우에 기록됩니다.

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	42 - 라인 유형 ID (NMEA0183 프로토콜 스트리밍)
3	Hedgehog의 주소
4, 5 등	NMEA0183 형식에 따른 필드 순서 (NMEA0183 레코드는 심표로 구분된 값 형식도 사용함)

9.1.4. 라인 유형 ID 43 – Hedgehog를 통해 전송되는 사용자 페이로드 데이터

이 라인에는 MMSW0005 라이선스가 필요합니다.

이 라인은 Hedgehog의 설정 인터페이스 섹션에서 0이 아닌 페이로드 데이터 크기가 활성화되어 있고, 사용자 장치가 Hedgehog의 USB 또는 UART를 통해 데이터를 전송할 때 기록됩니다.

또한, 페이로드 데이터는 일부 Marvelmind 장치(예: 로봇 v100 및 Boxie)에서도 사용 가능합니다.

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	43 - 라인 유형 ID (사용자 페이로드)
3	Hedgehog의 주소
4, 5 등	십표로 구분된 페이로드 데이터 바이트 시퀀스 (각 필드는 1바이트)

9.1.4.1. 로봇 v100용 페이로드 텔레메트리 데이터

라인의 일반 형식은 사용자 페이로드 데이터 형식에 해당합니다.

데이터 바이트(라인의 네 번째 필드부터 시작)는 아래에 설명된 형식의 데이터 레코드를 구성합니다.

멀티바이트 값은 하위 바이트부터 배치됩니다 (리틀 엔디안 형식).

로봇 v100 텔레메트리 레코드 N3:

오프셋	크기 (바이트)	타입	설명	값
0	2	uint16_t	레코드 ID	0x3003
2	2*12		12개 LiDAR의 거리 측정값 (LiDAR당 2바이트) 각 LiDAR 데이터의 형식은 다음과 같습니다: Bit 0...11 – LiDAR의 거리 측정값, mm Bit 12...15 – 거리 측정 상태 상태 = 0 – 거리 측정 완료 상태 <> 0 – 거리 미측정	
26	1	uint8_t	LiDAR 전체 상태: Bit0: 1 – LiDAR 읽기 성공 0 – LiDAR 읽기 실패 Bit 1...7 – 예약됨 (0)	
27	1	uint8_t	로봇 상태: 0: 로봇이 정상적으로 정지됨 1: 로봇이 알람에 의해 정지됨 2: 로봇이 자율 이동 중 3: 로봇이 충전 중	
28	1	uint8_t	RV - 로봇 배터리 전압.	

			V= (RV/10) + 20 Volts	
29	2	int16_t	로봇 공급 전류, x10 mA 값이 음수인 경우, 해당 전류로 로봇 배터리가 충전 중임.	N
31	1	uint8_t	잔여 배터리 용량, %	
32	2	uint16_t	로봇 공급 전력, Watt	
34	2	uint16_t	왼쪽 바퀴 속도, mm/s	
36	2	uint16_t	오른쪽 바퀴 속도, mm/s	
37	1	uint8_t	왼쪽 모터 출력, %	
38	1	uint8_t	오른쪽 모터 출력, %	
39	2	uint16_t	목표 속도, mm/s	
41	2	int16_t	로봇 X 좌표, cm	
43	1	uint8_t	상태 플래그: 비트 0...3 - 예약됨 비트 4: 1 - 초음파 추적 오류 비트 5...7 - 예약됨	
44	2	int16_t	로봇 Y 좌표, cm	

로봇 v100 텔레메트리 레코드 N4:

오프셋	크기 (바이트)	타입	설명	값
0	2	uint16_t	레코드 ID	0x3004
2	2*12		12개 LiDAR의 거리 값 (LiDAR당 2바이트) 각 LiDAR 데이터의 형식은 다음과 같습니다: 비트 0...11 - LiDAR 측정 거리, mm 비트 12...15 - 거리 측정 상태 상태 = 0 - 거리 측정 완료 상태 <> 0 - 거리 미측정	
26	1	uint8_t	LiDAR 전체 상태: 비트 0: 1 - LiDAR 읽기 성공 0 - LiDAR 읽기 실패 비트 1...7 - 예약됨 (0)	
27	1	uint8_t	로봇 상태: 0: 로봇이 정상적으로 정지됨 1: 로봇이 알람에 의해 정지됨 2: 로봇이 자율 이동 중 3: 로봇이 충전 중	
28	1	uint8_t	이동 프로그램에서 현재 항목의 인덱스	
29	1	uint8_t	이동 프로그램의 총 항목 수	
30	1	uint8_t	예약됨	
31	1	uint8_t	LiDAR에 의한 장애물 방향: 0 - 없음 1 - 전방 2 - 좌측 3 - 우측	
32	1	uint8_t	LiDAR에 의한 최소 경보 거리, x2cm	
33	1	uint8_t	경보를 발생시킨 현재 측정된 LiDAR 거리, x2cm	
34	3		예약됨	

36	2	int16_t	로봇 X 좌표, cm	
38	2	int16_t	로봇 Y 좌표, cm	

9.1.4.2. 로봇 Boxie의 페이로드 텔레메트리 데이터

라인의 일반 형식은 사용자 페이로드 데이터 형식에 해당합니다.

데이터 바이트(라인의 네 번째 필드부터 시작)는 아래에 설명된 형식의 데이터 레코드를 구성합니다.

멀티바이트 값은 하위 바이트부터 배치됩니다(리틀 엔디안 형식).

로봇 Boxie 텔레메트리 레코드 N1:

오 프 셋	크 기 (바 이 트)	타입	설명	값
0	2	uint16_t	레코드 ID	0x3101
2	2*12		12개 LiDAR의 거리값 (LiDAR당 2바이트) 각 LiDAR 데이터의 형식은 다음과 같습니다: 비트 0...11 - LiDAR의 거리, mm 비트 12...15 - 거리 측정 상태 상태 = 0 - 거리 측정됨 상태 <> 0 - 거리 측정되지 않음	
26	1	uint8_t	일반 LiDAR 상태: 비트0: 1 - LiDAR 읽기 성공 0 - LiDAR 읽기 실패 비트 1...7 - 예약됨 (0)	
27	1	uint8_t	로봇 상태: 0: 로봇이 정상적으로 정지됨 1: 로봇이 알람에 의해 정지됨 2: 로봇이 자율 이동 중 3: 로봇이 충전 중	
28	1	uint8_t	RV - 로봇 배터리 전압, x100 mV. 예를 들어, 값 118은 11.8V를 의미함	
29	2	int16_t	로봇 공급 전류, x10 mA 예를 들어, 값 123은 1.230 A를 의미함	
31	1	uint8_t	예약됨	
32	1	uint8_t	왼쪽 모터 출력, %	
33	1	uint8_t	오른쪽 모터 출력, %	
34	2	uint16_t	왼쪽 바퀴 속도, mm/s	
36	2	uint16_t	오른쪽 바퀴 속도, mm/s	
37	2	int16_t	왼쪽 모터 오도메트리 이동 경로, cm	
39	2	int16_t	오른쪽 모터 오도메트리 이동 경로, cm	
41	2	int16_t	로봇 X 좌표, cm	
43	1	uint8_t	상태 플래그: 비트 0...1 - 예약됨 비트 2: 1 - 이동 프로그램 실행 중 비트 2: 1 - 이동이 일시 정지됨 비트 4: 1 - 초음파 추적 오류 비트 5...7 - 예약됨	
44	2	int16_t	로봇 Y 좌표, cm	

로봇 Boxie 텔레메트리 레코드 N3:

오프셋	크기 (바이트)	타입	설명	값
0	2	uint16_t	레코드 ID	0x3103
2	2	int16_t	각도 제어 PID 조절기의 현재 "P" 값	
4	2	int16_t	각도 제어 PID 조절기의 현재 "I" 값	
6	2	int16_t	각도 제어 PID 조절기의 현재 "D" 값	
8	2	int16_t	EKF 필터를 사용하여 계산된 로봇 X 좌표, cm	
10	2	int16_t	EKF 필터를 사용하여 계산된 로봇 Y 좌표, cm	
12	2	int16_t	예약됨	
13	1	uint8_t	일반 LiDAR 상태: Bit0: 1 – LiDAR 읽기 성공 0 – LiDAR 읽기 실패 Bit 1...7 – 예약됨 (0)	
14	1	uint8_t	로봇 상태: 0: 로봇이 정상적으로 정지됨 1: 로봇이 알람에 의해 정지됨 2: 로봇이 자율 주행 중 3: 로봇이 충전 중	
15	1	uint8_t	현재 이동 단계의 인덱스 (첫 번째 경유점은 0, 두 번째는 1 등)	
16	1	uint8_t	현재 프로그램의 총 이동 단계 수	
17	1	uint8_t	이동 플래그: Bit 0: 1 – "Run forever" 옵션 Bit 1...7 - 예약됨	
18	1	uint8_t	0 – LiDAR에 의한 알람 없음 1...12 - 알람을 발생시킨 LiDAR의 인덱스	
19	1	uint8_t	LiDAR에 의한 최소 알람 거리, x2cm	
20	1	uint8_t	알람을 발생시킨 현재 측정된 LiDAR 거리, x2cm	
21	2	int16_t	로봇의 페어링된 비콘에 의한 각도, 단위: 도(degrees)	
23	1	uint8_t	로봇의 목표 속도 (사용자 설정), cm/s	
24	2	int16_t	로봇 X 좌표, cm	
26	2	int16_t	로봇 Y 좌표, cm	
28	2	int16_t	Reserved	
30	2	int16_t	Reserved	
32	2	int16_t	원하는 방향과 로봇의 현재 방향 사이의 편차 각도, 도(degrees)	
34	1	uint8_t	Reserved	
35	2	int16_t	융합된 로봇 방향 각도, 도(degrees)	
36	5	5 bytes	Reserved	

9.1.5. 라인 타입 ID 44 – Dashboard 실시간 플레이어 위치

이 라인은 MMSW0005 라이선스가 필요합니다.

이 라인은 실시간 플레이어가 활성화된 경우 Hedgehog에 대해 기록됩니다. 실시간 플레이어는 100 Hz 위치 데이터를 제공합니다.

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	44 - 라인 유형 ID (실시간 플레이어 위치)
3	Hedgehog의 주소
4	예약 필드
5	Hedgehog X 좌표, 미터(m)
6	Hedgehog Y 좌표, 미터(m)
7	Hedgehog Z 좌표, 미터(m)

9.1.6. 라인 유형 ID 55 – 간략 비콘 텔레메트리

이 라인에는 SSM(Super Super-Modem)으로부터 수신된 비콘 텔레메트리가 포함됩니다.

라인의 필드:

N	필드 설명
0	타임스탬프 (공통 필드)
1	사용자 이름 (공통 필드)
2	55 - 라인 유형 ID (간략 텔레메트리)
3	비콘의 주소
4	비콘 전원 전압의 하위 바이트: V0
5	비콘 전원 전압의 상위 바이트: V1 비콘 전원 전압은 $V = V0 + V1 * 256$ mV
6	2의 보수 코드로 표현된 RSSI: R0 R0이 128 미만인 경우, $RSSI = R0$ dBm ($R0 \geq 128$)인 경우 $RSSI = R0 - 256$ dBm
7	예약됨
8	예약됨

9.2. 이전 형식의 csv 로그 (V7.000 이전의 Dashboard 또는 Modem HW v4.9)

다음은 이전 형식의 로그 파일을 나타내는 그림입니다:

Format of CSV file recorded by dashboard

```
1608733078625,0,2911360,60,0.805,1.160,1.000,50,1.784,58,1.519,255,0,0,0,0,
1608733078656,31,2911391,60,0.805,1.160,1.000,50,1.784,58,1.519,255,0,0,0,0,
1608733078656,0,2911391,60,0.805,1.160,1.000,50,1.784,58,1.519,255,0,0,0,0,
1608733078671,15,2911406,60,0.805,1.160,1.000,50,1.784,58,1.519,255,0,0,0,0,
1608733078671,0,2911406,60,0.805,1.160,1.000,50,1.784,58,1.519,255,0,0,0,0,
1608733078687,16,2911422,60,0.805,1.160,1.000,50,1.784,58,1.519,255,0,0,0,0,
1608733078687,0,2911422,60,0.805,1.160,1.000,50,1.784,58,1.519,255,0,0,0,0,
1608733078843,156,2911578,60,0.806,1.159,1.000,50,1.784,58,1.519,255,0,0,0,0,
1608733078843,0,2911578,60,0.806,1.159,1.000,50,1.784,58,1.519,255,0,0,0,0,
1608733078859,16,2911594,60,0.806,1.159,1.000,50,1.778,58,1.528,255,0,0,0,0,
1608733078859,0,2911594,60,0.806,1.159,1.000,50,1.778,58,1.528,255,0,0,0,0,
1608733078890,31,2911625,60,0.807,1.159,1.000,50,1.778,58,1.528,255,0,0,0,0,
1608733078890,0,2911625,60,0.807,1.159,1.000,50,1.778,58,1.528,255,0,0,0,0,
```

Case specific parameters: robot state, some users telemetry or so on

32-bit status word.
Bit 0: 1 = geofencing zone alarm
Bit 1...31: reserved

Separator. 255 means end of raw distances list

Addresses of stationary beacons and distances from hedgehog to them (meters)

Coordinates of hedgehog: X,Y,Z (meters)

Address of hedgehog

Time from running dashboard, ms

Time from previous record, ms

Time from 1 january 1970 (Unix time), ms

10. Marvelmind API

Marvelmind API 라이브러리는 Marvelmind Dashboard 소프트웨어에서 사용되며 사용자 소프트웨어에 대한 인터페이스를 제공합니다. API는 MS Windows용 동적 링크 라이브러리(DLL) 및 Linux(x86 및 ARM 플랫폼)용 공유 라이브러리 형태로 제공됩니다. API는 USB(가상 직렬 포트)를 통해 Modem에 연결되며 Modem과의 통신 프로토콜을 구현합니다.

API 라이브러리 외에도, 소프트웨어 패키지에는 API 테스트에 사용되었으며 모든 API 함수 호출을 포함하는 C 예제 소프트웨어가 포함되어 있습니다.

해당 예제는 사용자 소프트웨어 개발의 기초로 활용하거나, API 라이브러리 인터페이스(파일 'marvelmind_api.c')를 다른 프로그래밍 언어로 포팅하는 데 사용할 수 있습니다.

테스트 환경:

1. MS Windows 10; CPU: Intel Core i5
2. Ubuntu 20.04; CPU: Intel Core i5
3. Raspbian (2018-11-13-raspbian-stretch-full); 플랫폼: Raspberry Pi 3 Model B+

10.1. Windows 설치 방법

- Marvelmind API 소프트웨어 패키지를 다운로드합니다. Dashboard API와 예제 소프트웨어를 사용할 디렉터리에 복사합니다. Windows 버전의 예제는 미리 빌드된 실행 파일과 함께 제공되므로, API 소프트웨어 패키지의 'windows' 디렉터리에 있는 'mm_api_example.exe'를 즉시 실행할 수 있습니다.

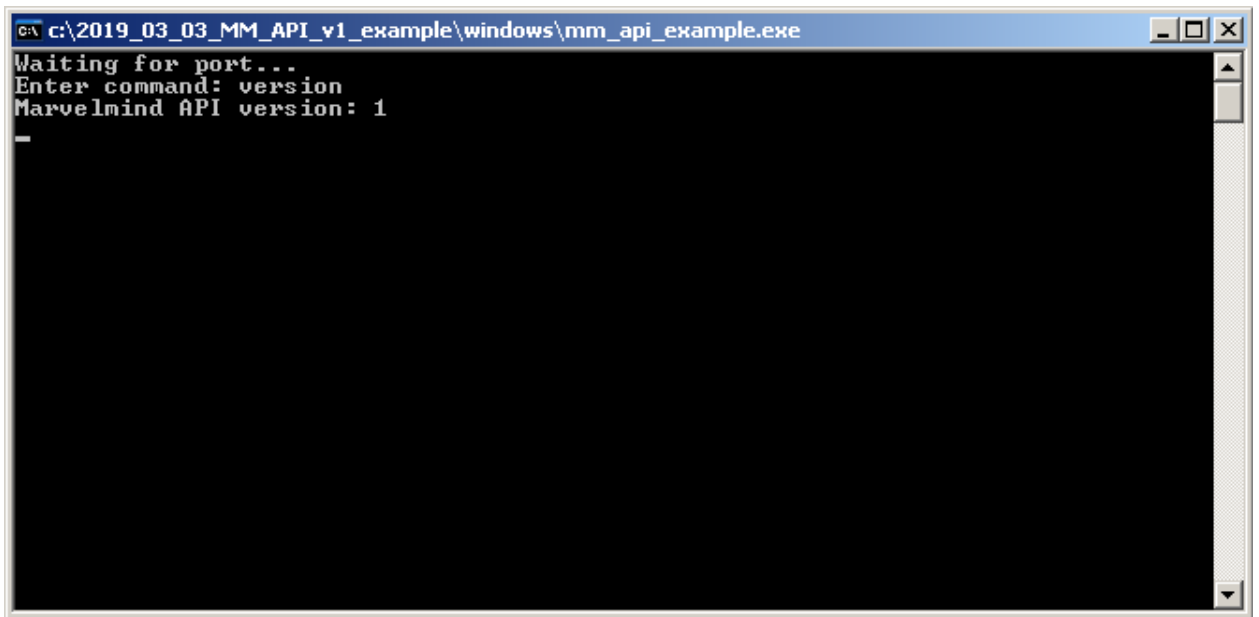
10.2. Linux 설치 방법

- Marvelmind API 소프트웨어 패키지를 다운로드합니다. Dashboard API를 사용할 디렉터리에 복사합니다. Linux 버전은 x86(Intel 또는 AMD CPU 기반의 대부분의 노트북)과 arm(예: Raspberry Pi와 같은 싱글보드 컴퓨터) 두 가지 하드웨어 플랫폼을 위해 제공됩니다.
- 사용 중인 플랫폼에 맞는 라이브러리 libdashapi.so를 /usr/local/lib 디렉터리에 복사하려면, libdashapi.so가 있는 디렉터리에서 열린 터미널에서 `sudo cp libdashapi.so /usr/local/lib` 명령을 실행합니다. 이후 터미널에서 `sudo ldconfig`를 실행합니다.
- 사용자가 직렬 포트에 접근할 수 있도록 dialout 그룹에 추가하여 권한을 부여해야 할 수 있습니다:
 - 터미널에서 실행: `sudo adduser $USER dialout`
 - /etc/udev/rules.d 디렉토리에 다음 내용을 포함한 "99-tty.rules" 파일을 추가하십시오:

```
#Marvelmind serial port rules
KERNEL=="ttyACM0",GROUP="dialout",MODE="666"
```
- 예제 소프트웨어를 빌드하십시오 – 패키지에 포함된 'source' 디렉토리에서 열린 터미널에서 'make all'을 실행하십시오
- 터미널에서 './mm_api_example'을 입력하여 예제를 실행하십시오

10.3. API 연결 확인

예제 소프트웨어를 실행한 후, 터미널에서 "space" 버튼을 누르고 'version' 명령을 입력한 다음 엔터를 누르십시오. 예제 소프트웨어가 API 버전을 출력하면 API 라이브러리와 통신할 수 있는 것입니다.



```
c:\2019_03_03_MM_API_v1_example\windows\mm_api_example.exe
Waiting for port...
Enter command: version
Marvelmind API version: 1
-
```

10.4. Marvelmind API 라이브러리 설명

API는 MS Windows용 동적 링크 라이브러리(DLL) 및 Linux(x86 및 ARM 플랫폼)용 공유 라이브러리로 제공됩니다. 이 라이브러리는 컴퓨터의 USB 포트에 연결된 modem을 통해 Marvelmind 시스템을 모니터링하고 제어하기 위한 함수 집합을 포함합니다. 본 문서의 이 섹션에는 이러한 모든 함수에 대한 설명이 포함되어 있습니다.

다양한 프로그래밍 언어와의 호환성을 높이기 위해, 복잡한 데이터 구조의 대부분은 메모리에 대한 비형식 포인터를 통해 전달됩니다. 함수 설명에는 메모리 풀 내 각 데이터 필드의 오프셋이 포함되어 있습니다. 예제 소프트웨어의 'marvelmind_api.c' 파일에서 메모리 풀과 c 구조체 필드 간의 데이터 이동 구현을 확인할 수 있습니다.

설명에서 매개변수의 타입은 c 문법으로 표시됩니다. 다음은 타입에 대한 설명입니다:

타입	크기 (바이트)	설명
bool	1	불리언 타입. 0은 false를 의미하며, 0이 아닌 값은 true를 의미합니다
uint8_t	1	부호 없는 정수 값, 0...255
int8_t	1	2의 보수 형식의 부호 있는 정수 값, -128...127
uint16_t	2	부호 없는 정수 값, 0...65535
int16_t	2	2의 보수 형식의 부호 있는 정수 값, -32768...32767
uint32_t	4	부호 없는 정수 값, 0...4294967295
int32_t	4	2의 보수 형식의 부호 있는 정수 값, -2147483648...2147483647
void *	4/8	메모리 포인터 (메모리 내 주소). 32비트 플랫폼의 경우 4바이트, 64비트 플랫폼의 경우 8바이트.

각 함수 설명에는 해당 함수를 사용할 수 있는 API 버전 집합이 포함되어 있습니다. 새로운 API 버전은 Marvelmind 시스템의 새로운 기능을 위한 더 많은 함수를 지원할 예정입니다. 현재 Dashboard에서 사용 가능한 모든 기능이 API를 통해 제공되는 것은 아니므로, 추가적인 API 함수가 필요하신 경우 info@marvelmind.com으로 문의하시기 바랍니다.

지원되는 함수 목록:

함수	API 버전	필요한 라이선스
Marvelmind API 라이브러리 버전 가져오기	V1+	없음
마지막 오류 가져오기	V6+	없음
시리얼 포트 열기 시도	V1+	없음
지정된 이름으로 시리얼 포트 열기 시도	V2+	없음
UDP 포트 열기 시도	V9+	없음
시리얼 포트 닫기	V1+	없음
Marvelmind 장치의 버전 및 CPU ID 가져오기	V1+	없음
장치 목록 가져오기	V1+	none
장치 깨우기	V1+	none
장치를 슬립 모드로 전환	V1+	none
비콘에서 텔레메트리 데이터 가져오기	V1+	none
최신 위치 데이터 가져오기	V1+	none
최신 위치 데이터 가져오기 (각도 포함)	V3+	none
비콘의 위치 설정	V3+	MMSW0005
비콘 간 거리 설정	V4+	MMSW0005
최신 원시 거리 데이터 가져오기	V1+	none
Hedgehog의 높이 가져오기	V4+	none
Hedgehog의 높이 설정	V4+	MMSW0005
서브맵 내 고정 비콘의 높이 가져오기	V4+	none
서브맵 내 고정 비콘의 높이 설정	V4+	MMSW0005
위치 업데이트 속도 설정 가져오기	V1+	없음
위치 업데이트 속도 설정	V1+	MMSW0005
서브맵 추가	V1+	MMSW0005
서브맵 삭제	V1+	MMSW0005
서브맵 고정	V1+	MMSW0005
서브맵 고정 해제	V1+	MMSW0005
서브맵 설정 가져오기	V1+	없음
서브맵 설정 지정	V1+	MMSW0005
맵 동결	V4+	MMSW0005
맵 동결 해제	V4+	MMSW0005
비콘의 초음파 설정 가져오기	V1+	none
비콘의 초음파 설정 지정	V1+	MMSW0005
맵 삭제	V1+	MMSW0005
장치를 기본 설정으로 초기화	V1+	MMSW0005
비콘을 축에 연결	V2+	MMSW0005
Modem의 구성 메모리 덤프 읽기	V3+	MMSW0005
Modem의 구성 메모리 덤프 쓰기	V3+	MMSW0005
Modem에서 공기 온도 설정값 가져오기	V3+	none
Modem에 공기 온도 설정값 지정	V3+	none
장치 소프트웨어 리셋	V3+	none
비콘 실시간 플레이어 설정 가져오기	V6+	없음
비콘 실시간 플레이어 설정 지정	V6+	MMSW0005
지오레퍼런싱 설정 가져오기	V6+	없음
지오레퍼런싱 설정 지정	V6+	MMSW0005
위치 업데이트 모드 가져오기	V6+	없음
위치 업데이트 모드 지정	V6+	MMSW0005
위치 업데이트 명령	V6+	MMSW0005

비콘의 지오펜싱 알람 상태 지정	V9+	MMSW0005 MMSW0006
일반 사용자 페이로드 데이터 전송	V9+	MMSW0005
일반 사용자 페이로드 데이터 수신	V9+	MMSW0005
수동 거리 측정 명령 전송	V9+	MMSW0011
Modem에서 스트리밍 데이터 수신	V9+	none
장치 유형이 Modem인지 확인	V1+	none
장치 유형이 고정 beacon인지 확인	V1+	없음
장치 유형이 Hedgehog인지 확인	V1+	없음

10.4.1. Marvelmind API 라이브러리 버전 가져오기

API 라이브러리의 버전을 읽습니다. 이 버전의 라이브러리에서 필요한 함수를 사용할 수 있는지 확인하는 데 필요합니다.

함수 이름: `mm_api_version`

C 선언: `bool mm_api_version(void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

유형	설명
bool	true – 함수가 성공적으로 실행됨 false – 실행 중 오류 발생

매개변수:

유형	설명
void *	채울 데이터에 대한 포인터

포인터를 통해 반환되는 데이터의 구조체

타입	설명
uint32_t	API 라이브러리 버전

10.4.2. 마지막 오류 가져오기

API 라이브러리의 마지막 작업 상태를 읽어 오류 원인을 구분합니다.

함수 이름: `mm_get_last_error`

C 선언: `bool mm_get_last_error(void *pdata);`

사용 가능한 API 버전: V6+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

타입	설명
void *	채울 데이터에 대한 포인터

포인터를 통해 반환되는 데이터의 구조체.

타입	설명
uint32_t	마지막 작업의 상태: 0: 작업이 성공적으로 실행됨 1: 통신 오류 2: 시리얼 포트 열기 오류 3: 라이선스가 필요함

10.4.3. 시리얼 포트 열기

Marvelmind 장치(Modem 또는 beacon)가 USB(가상 시리얼 포트)를 통해 연결된 포트를 엽니다. API가 모든 시리얼 포트를 검색하여 Marvelmind 장치에 해당하는지 여부를 확인하므로 시리얼 포트 이름을 지정할 필요가 없습니다.

함수 이름: mm_open_port

C 선언: bool mm_open_port ();

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 포트가 열림 false - 실행 중 오류 발생

매개변수: 없음

10.4.4. 지정된 이름으로 시리얼 포트 열기

Marvelmind 장치(Modem 또는 beacon)가 USB(가상 시리얼 포트)를 통해 연결된 포트를 엽니다. 함수는 지정된 이름의 포트를 열려고 시도합니다.

함수 이름: mm_open_port_by_name

C 선언: `bool mm_open_port_by_name(void *pdata);`

사용 가능한 API 버전: V2+

필요한 라이선스: 없음

반환 값:

유형	설명
bool	true – 함수가 성공적으로 실행되었으며, 포트가 열림 false – 실행 중 오류 발생

매개변수:

유형	설명
void *	시리얼 포트 이름에 대한 포인터 – 널(zero)로 종료되는 ASCII 문자 시퀀스 (ASCIIZ)

10.4.4.1. UDP 포트 열기

USB 대신 UDP를 통해 Super-Modem과의 통신을 설정할 수 있습니다.

함수 이름: mm_open_port_udp

C 언어 선언: bool mm_open_port_udp(void *pdata);

사용 가능한 API 버전: V9+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며 UDP 포트가 열림 false - 실행 오류

매개변수:

타입	설명
void *	UDP 설정 구조체에 대한 포인터 (아래 참조)

포인터가 가리키는 데이터 구조:

타입	설명
uint16_t	연결할 UDP 포트
uint16_t	통신 타임아웃, ms
uint16_t	예약됨
최대 255바이트	IP 주소 - 널(zero)로 종료되는 ASCII 문자 시퀀스 (ASCIIZ)

IP 주소와 UDP 포트는 Super-Modem의 설정과 일치해야 합니다 (아래 스크린샷 참조).

Setting	Value
Wi-Fi/UDP settings	(-) faberge_LTE_2.4GHz
Wi-Fi	enabled
Wi-Fi network name	faberge_LTE_2.4GHz
Wi-Fi network password	*****
Show password	disabled
<input checked="" type="checkbox"/> Wi-Fi reconnect timeout, sec (10..65000)	120
Static IP	disabled
Static IP address	n/a
Router IP address	n/a
Wi-Fi RSSI, dBm	-69
Own IP address	192.168.1.102
UDP destination IP address	192.168.1.102
UDP destination port (0..65535)	49100
UDP port for API (0..65535)	49213

10.4.5. 시리얼 포트 닫기

mm_open_port 함수에 의해 이전에 열린 경우 포트를 닫습니다.

함수 이름: mm_close_port

C 언어 선언: `bool mm_close_port ();`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 포트가 닫혔습니다 false - 실행 중 오류 발생

매개변수: 없음

10.4.6. Marvelmind 장치의 버전 및 CPU ID 가져오기

버전 및 CPU ID를 읽습니다. 버전에는 펌웨어 버전 및 장치 하드웨어 유형에 관한 정보가 포함됩니다. CPU ID는 해당 장치 항목의 고유 ID입니다.

함수 이름: `mm_get_device_version_and_id`

C 언어 선언: `bool mm_get_device_version_and_id (uint8_t address, void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

유형	설명
bool	true – 함수가 성공적으로 실행되었으며, 버전 및 CPU ID 데이터가 검색되었습니다 false – 실행 중 오류 발생

매개변수:

유형	설명
uint8_t	Marvelmind 장치의 주소 (1...254)
void *	채워질 데이터에 대한 포인터

포인터를 통해 반환되는 데이터의 구조:

타입	설명
uint8_t	펌웨어의 주 버전 (예: 버전 V6.07a의 경우 "6")
uint8_t	펌웨어의 부 버전 (예: 버전 V6.07a의 경우 "7")
uint8_t	펌웨어의 두 번째 부 버전 (예: 버전 V6.07a의 경우 "1")
uint8_t	장치 유형 ID (부록 참조).
uint8_t	펌웨어 옵션 (추후 정의 예정).
uint32_t	CPU ID. 이 값을 16진수로 출력하면 Dashboard 및 장치 스티커에 표시된 형식의 CPU ID를 확인할 수 있습니다.

10.4.7. 장치 목록 가져오기

Modem에 알려진 Marvelmind 장치 목록을 읽습니다. 이 목록에는 슬립 상태인 장치를 포함하여 Modem 네트워크에 무선으로 연결된 모든 장치의 목록이 포함됩니다.

함수 이름: `mm_get_devices_list`

C 언어 선언: `bool mm_get_devices_list (void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환 값:

타입	설명
bool	true – 함수가 성공적으로 실행되었으며, 장치 목록이 검색됨 false – 실행 오류

매개변수:

타입	설명
void *	채워질 데이터에 대한 포인터

포인터를 통해 반환되는 데이터 구조:

타입	설명
uint8_t	목록에 포함된 다음 장치의 수 (N)
N*9 바이트	다음 표에 설명된 N개 장치 구조체의 시퀀스

목록에서 각 장치의 구조체:

타입	설명
uint8_t	장치의 주소
bool	true = 중복된 주소 - 동일한 주소를 가진 장치가 2개 이상 발견됨 false = 중복되지 않은 주소
bool	true = 장치가 슬립 상태임 false = 장치가 슬립 상태가 아님
uint8_t	펌웨어의 메이저 버전 (예: 버전 V6.07a의 경우 "6")
uint8_t	펌웨어의 마이너 버전 (예: 버전 V6.07a의 경우 "7")
uint8_t	펌웨어의 두 번째 마이너 버전 (예: 버전 V6.07a의 경우 "1")
uint8_t	장치 유형 ID (부록 참조).
uint8_t	펌웨어 옵션 (TBD).
uint8_t	플래그: 비트 0: 1 – 장치 연결 완료 – 장치가 연결을 확인함 0 – 장치로부터 확인 대기 중 (Dashboard의 'Connecting...'과 같은 상태). 비트 1...7 - TBD

10.4.8. 장치 깨우기

지정된 장치를 깨우는 명령을 전송합니다. 깨우기 명령이 전송되고 해당 장치가 존재하는 경우, 장치는 몇 초 내에 Modem에 연결되며 장치 목록에 나타납니다.

함수 이름: `mm_wake_device`

C 언어 선언: `bool mm_wake_device (uint8_t address);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

유형	설명
bool	true - 함수가 성공적으로 실행되었으며, 깨우기 명령이 전송됨 false - 실행 중 오류 발생

매개변수:

타입	설명
uint8_t	1...254 - 깨울 Marvelmind 장치의 주소 0 - 모든 장치 깨우기

10.4.9. 장치를 슬립 상태로 전환

기존 장치를 슬립 상태로 전환합니다.

함수 이름: `mm_send_to_sleep_device`

C 선언: `bool mm_send_to_sleep_device (uint8_t address);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 슬립 명령이 전송됨 false - 실행 오류

매개변수:

타입	설명
uint8_t	1...254 - 슬립 상태로 전환할 Marvelmind 장치의 주소 0 - 모든 장치를 슬립 상태로 전환

10.4.10. Beacon에서 텔레메트리 데이터 가져오기

Marvelmind beacon의 텔레메트리 데이터를 읽습니다.

함수 이름: `mm_get_beacon_telemetry`

C 선언: `bool mm_get_beacon_telemetry (uint8_t address, void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true – 함수가 성공적으로 실행되었으며 텔레메트리가 검색됨 false – 실행 중 오류 발생

매개변수:

타입	설명
uint8_t	Marvelmind 비콘의 주소 (1...254)
void *	채울 데이터에 대한 포인터

포인터를 통해 반환되는 데이터 구조:

타입	설명
uint32_t	비콘의 동작 시간, 초 (리셋 또는 웨이크업 이후 경과 시간)
int8_t	RSSI, dBm – 무선 신호 강도
int8_t	측정된 온도, °C
uint16_t	공급 전압, mV
16 바이트	예약됨 (0)

10.4.11. 최신 위치 데이터 가져오기

Modem에서 최신 업데이트된 좌표 패키지를 읽습니다. 또한 사용자 페이로드 데이터가 있는 경우 이를 함께 읽습니다.

함수 이름: `mm_get_last_locations`

C 언어 선언: `bool mm_get_last_locations(void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환 값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 위치 데이터가 검색되었음 false - 실행 오류

매개변수:

타입	설명
void *	채워질 데이터에 대한 포인터

포인터를 통해 반환된 데이터 구조:

타입	설명
18*6 bytes	마지막으로 업데이트된 좌표의 6개 18바이트 데이터 구조체, 아래 표 참조
bool	true - 읽을 수 있는 새로운 원시 거리 데이터가 있음
5 바이트	TBD
uint8_t	사용자 페이로드 데이터 크기 (M)
M 바이트	사용자 페이로드 데이터

각 위치 데이터 항목의 구조:

타입	설명
uint8_t	장치 주소 (1...254) 0 - 이 데이터 항목은 채워지지 않음
uint8_t	헤드 인덱스 (TBD)
int32_t	X 좌표, mm
int32_t	Y 좌표, mm
int32_t	Z 좌표, mm
uint8_t	상태 플래그 (추후 결정)
uint8_t	측위 품질, 0...100%
uint8_t	추후 결정
uint8_t	추후 결정

10.4.12. 최신 위치 데이터 가져오기 (각도 포함)

Modem에서 최신 업데이트된 좌표 패키지를 읽습니다 (쌍을 이룬 비콘의 각도 포함). 사용자 페이로드 데이터가 있는 경우 함께 읽습니다.

함수 이름: mm_get_last_locations2
 C 선언: bool mm_get_last_locations2(void *pdata);
 사용 가능한 API 버전: V3+
 필요한 라이선스: 없음

반환값:

유형	설명
bool	true - 함수가 성공적으로 실행되었으며, 위치 데이터가 검색됨 false - 실행 오류

매개변수:

유형	설명
void *	채워질 데이터에 대한 포인터

포인터를 통해 반환되는 데이터 구조:

유형	설명
20*6 bytes	마지막으로 업데이트된 좌표의 6개 20-byte 데이터 구조, 아래 표 참조
bool	true - 읽을 수 있는 새로운 원시 거리 데이터가 있음
5 bytes	TBD
uint8_t	사용자 페이로드 데이터 크기 (M)
M bytes	사용자 페이로드 데이터

각 위치 데이터 항목의 구조:

유형	설명
uint8_t	장치 주소 (1...254) 0 - 이 데이터 항목은 채워지지 않음
uint8_t	헤드 인덱스 (TBD)
int32_t	X 좌표, mm
int32_t	Y 좌표, mm
int32_t	Z 좌표, mm
uint8_t	상태 플래그 (TBD)
uint8_t	측위 품질, 0...100%
uint8_t	TBD
uint8_t	TBD
uint16_t	비트 0...11 - 회전 각도 (1/10도 단위, 페어링된 비콘 기능이 활성화된 경우) 비트 12 - 1 = 각도를 사용할 수 없음 비트 13...15 - 예약됨

10.4.13. 비콘 위치 설정

지정된 비콘의 위치를 수동으로 설정합니다.

함수 이름: `mm_set_beacon_location`

C 선언: `bool mm_set_beacon_location (uint8_t address, void *pdata);`

사용 가능한 API 버전: V3+

필요한 라이선스: MMSW0005

반환값:

타입	설명
bool	true – 함수가 성공적으로 실행되었으며, 위치가 업데이트됨 false – 실행 오류

매개변수:

타입	설명
uint8_t	비콘의 주소
void *	위치 데이터가 담긴 버퍼에 대한 포인터

포인터가 가리키는 데이터 구조 (함수 호출 전에 채워져야 함):

타입	설명
int32_t	비콘의 새 X 좌표, mm
int32_t	비콘의 새 Y 좌표, mm
int32_t	비콘의 새 Z 좌표, mm

10.4.14. 비콘 간 거리 설정

비콘 간 거리의 수동 설정.

함수 이름: `mm_set_beacons_distance`

C 선언: `bool mm_set_beacons_distance (void *pdata);`

사용 가능한 API 버전: V4+

필요한 라이선스: MMSW0005

반환값:

타입	설명
bool	true – 함수가 성공적으로 실행되었으며, 거리가 기록됨 false – 실행 오류

매개변수:

유형	설명
void *	거리 데이터가 포함된 버퍼에 대한 포인터

포인터에 의한 데이터 구조 (함수 호출 전에 채워져야 함):

유형	설명
uint8_t	첫 번째 비콘의 주소
uint8_t	두 번째 비콘의 주소
int32_t	비콘 간 거리, mm

10.4.15. 최신 원시 거리 데이터 가져오기

Modem에서 최신 업데이트된 원시 거리 패킷을 읽습니다.

함수 이름: mm_get_last_distances

C 언어 선언: bool mm_get_last_distances(void *pdata);

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 원시 거리 데이터가 검색됨 false - 실행 중 오류 발생

매개변수:

타입	설명
void *	채워질 데이터에 대한 포인터

포인터를 통해 반환되는 데이터 구조:

타입	설명
uint8_t	원시 거리 데이터 항목의 수 (N) 요청당 최대 원시 거리 수는 16개입니다: $N \leq 16$
9*N 바이트	마지막 원시 거리의 N개 9바이트 데이터 구조체, 아래 표 참조

각 원시 거리 데이터 항목의 구조:

타입	설명
uint8_t	초음파 RX 장치의 주소 (1...254) 0 - 이 데이터 항목은 채워지지 않음
uint8_t	RX Head 인덱스 (TBD)
uint8_t	초음파 TX 장치의 주소 (1...254) 0 - 이 데이터 항목은 채워지지 않음
uint8_t	TX Head 인덱스 (TBD)
uint32_t	TX 장치에서 RX 장치까지의 거리, mm
uint8_t	TBD

10.4.16. Hedgehog의 높이 가져오기

모바일 비콘(Hedgehog)의 높이를 반환합니다.

함수 이름: mm_get_hedge_height

C 언어 선언: bool mm_get_hedge_height (uint8_t address, void *pdata);

사용 가능한 API 버전: V4+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 높이가 반환됨 false - 실행 중 오류 발생

매개변수:

타입	설명
uint8_t	Hedgehog의 주소
void *	높이 데이터가 저장된 버퍼에 대한 포인터

포인터가 가리키는 데이터 구조:

타입	설명
int32_t	Hedgehog의 높이, mm

10.4.17. Hedgehog의 높이 설정

모바일 비콘(Hedgehog)의 높이를 설정합니다.

함수 이름: `mm_set_hedge_height`

C 선언: `bool mm_set_hedge_height (uint8_t address, void *pdata);`

사용 가능한 API 버전: V4+

필요한 라이선스: MMSW0005

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 높이가 변경됨 false - 실행 중 오류 발생

매개변수:

타입	설명
uint8_t	Hedgehog의 주소
void *	높이 데이터가 담긴 버퍼에 대한 포인터

포인터가 가리키는 데이터 구조 (함수 호출 전에 채워져야 함):

타입	설명
int32_t	Hedgehog의 높이, mm

10.4.18. 서브맵에서 고정 비콘의 높이 가져오기

서브맵에서 고정 비콘의 높이를 반환합니다.

함수 이름: `mm_get_beacon_height`

C 언어 선언: `bool mm_get_beacon_height (uint8_t address, void *pdata);`

사용 가능한 API 버전: V4+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 높이가 반환됨 false - 실행 오류

매개변수:

타입	설명
uint8_t	Beacon의 주소
void *	높이 데이터가 담긴 버퍼에 대한 포인터

포인터로 참조되는 데이터 구조:

타입	설명
uint8_t	Submap ID, 함수 호출 전에 반드시 입력되어야 함
int32_t	Beacon의 높이, mm

10.4.19. Submap 내 고정형 Beacon의 높이 설정

Submap 내 고정형 Beacon의 높이를 설정합니다.

함수 이름: `mm_set_beacon_height`

C 언어 선언: `bool mm_set_beacon_height (uint8_t address, void *pdata);`

사용 가능한 API 버전: V4+

필요한 라이선스: MMSW0005

반환 값:

타입	설명
<code>bool</code>	<code>true</code> – 함수가 성공적으로 실행됨, 높이가 변경됨 <code>false</code> – 실행 오류

매개변수:

타입	설명
<code>uint8_t</code>	비콘의 주소
<code>void *</code>	높이 데이터가 담긴 버퍼에 대한 포인터

포인터가 가리키는 데이터 구조 (함수 호출 전에 채워져야 함):

타입	설명
<code>uint8_t</code>	서브맵 ID
<code>int32_t</code>	비콘의 높이, mm

10.4.20. 위치 업데이트 속도 설정 가져오기

Modem에서 위치 업데이트 속도 설정을 읽어옵니다.

함수 이름: `mm_get_update_rate_setting`

C 선언: `bool mm_get_update_rate_setting (void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true – 함수가 성공적으로 실행되었으며, 업데이트 속도가 검색됨 false – 실행 중 오류 발생

매개변수:

타입	설명
void *	채워질 데이터에 대한 포인터

포인터를 통해 반환되는 데이터 구조:

타입	설명
uint32_t	위치 업데이트 속도 설정 (mHz 단위). 따라서 1 Hz의 경우 1000, 16 Hz의 경우 16000, 0.05 Hz 모드의 경우 50이 반환됩니다.

10.4.21. 위치 업데이트 속도 설정 지정

Modem에 위치 업데이트 속도 설정을 씁니다.

함수 이름: `mm_set_update_rate_setting`

C 선언: `bool mm_set_update_rate_setting (void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: MMSW0005

반환값:

타입	설명
bool	true – 함수가 성공적으로 실행되었으며, 업데이트 속도가 변경됨 false – 실행 중 오류 발생

매개변수:

타입	설명
void *	데이터에 대한 포인터

포인터에 의한 데이터 구조 (함수 호출 전에 채워져야 함):

유형	설명
uint32_t	mHz 단위의 위치 업데이트 속도 설정. 1 Hz의 경우 1000, 16 Hz의 경우 16000, 0.05 Hz 모드의 경우 50이 반환됩니다. 시스템은 다음 시리즈에서 지정된 업데이트 속도에 가장 가까운 값을 사용합니다: 0.05 Hz, 0.1 Hz, 0.2 Hz, 0.5 Hz, 1 Hz, 2 Hz, 4 Hz, 8 Hz, 12 Hz, 16 Hz, 16+Hz.

10.4.22. 서브맵 추가

새 서브맵을 추가합니다.

함수 이름: `mm_add_submap`

C 언어 선언: `bool mm_add_submap (uint8_t submapId);`

사용 가능한 API 버전: V1+

필요한 라이선스: MMSW0005

반환 값:

유형	설명
<code>bool</code>	<code>true</code> - 함수가 성공적으로 실행되었으며, 서브맵이 추가됨 <code>false</code> - 실행 중 오류 발생

매개변수:

유형	설명
<code>uint8_t</code>	추가할 Submap ID (0...254)

10.4.23. Submap 삭제

기존 Submap을 삭제합니다.

함수 이름: mm_delete_submap

C 언어 선언: `bool mm_delete_submap (uint8_t submapId);`

사용 가능한 API 버전: V1+

필요한 라이선스: MMSW0005

반환값:

유형	설명
bool	true – 함수가 성공적으로 실행되었으며, Submap이 제거되었습니다. false – 실행 중 오류 발생

매개 변수:

유형	설명
uint8_t	삭제할 Submap ID (0...254)

10.4.24. Submap 고정

Submap을 고정합니다.

함수 이름: `mm_freeze_submap`

C 선언: `bool mm_freeze_submap (uint8_t submapId);`

사용 가능한 API 버전: V1+

필요한 라이선스: MMSW0005

반환값:

타입	설명
<code>bool</code>	<code>true</code> – 함수가 성공적으로 실행되었으며, 서브맵이 동결됨 <code>false</code> – 실행 중 오류 발생

매개변수:

타입	설명
<code>uint8_t</code>	동결할 서브맵 ID (0...254)

10.4.25. 서브맵 동결 해제

서브맵의 동결을 해제합니다.

함수 이름: `mm_unfreeze_submap`

C 선언: `bool mm_unfreeze_submap (uint8_t submapId);`

사용 가능한 API 버전: V1+

필요한 라이선스: MMSW0005

반환 값:

타입	설명
<code>bool</code>	<code>true</code> - 함수가 성공적으로 실행됨, submap이 동결 해제됨 <code>false</code> - 실행 오류

매개 변수:

타입	설명
<code>uint8_t</code>	동결 해제할 submap ID (0...254)

10.4.26. submap 설정 가져오기

Modem에서 submap 설정을 읽어옵니다.

함수 이름: mm_get_submap_settings

C 언어 선언: bool mm_get_submap_settings (uint8_t submapId , void *pdata);

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환 값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 서브맵 설정이 검색됨 false - 실행 중 오류 발생

매개변수:

유형	설명
uint8_t	서브맵 ID (0...254)
void *	채워질 데이터에 대한 포인터

포인터를 통해 반환되는 데이터 구조:

유형	설명
uint8_t	삼변측량 시작 beacon
uint8_t	beacon 시작 세트, beacon 1
uint8_t	beacon 시작 세트, beacon 2
uint8_t	비콘 시작 세트, 비콘 3
uint8_t	비콘 시작 세트, 비콘 4
bool	true = 3D 내비게이션 활성화됨
bool	true = 서브맵이 Z 좌표에만 사용됨
bool	true = 수동 제한 거리 false = 자동 제한 거리
uint8_t	최대 거리, 미터 (수동 제한 거리의 경우)
int16_t	서브맵 X 이동값, cm
int16_t	서브맵 Y 이동값, cm
int16_t	서브맵 Z 이동값, cm
uint16_t	서브맵 회전, 센티도(centidegrees)
int16_t	평면 회전 쿼터니언, W (쿼터니언은 10000으로 정규화됨)
int16_t	평면 회전 쿼터니언, X
int16_t	평면 회전 쿼터니언, Y
int16_t	평면 회전 쿼터니언, Z
int16_t	서비스 구역 두께, cm
int16_t	2D 모드에서의 Hedgehog 높이
bool	true = 서브맵이 고정됨(frozen)
bool	true = 서브맵이 잠김(locked)
bool	true = 고정형 비콘이 이동형 비콘보다 높은 위치에 있음
bool	true = 서브맵이 미러링됨
4 bytes	서브맵 내 beacon 주소 목록 (0 = 없음)
8 bytes	인접 서브맵 ID 목록 (255 = 없음)
uint8_t	서비스 구역 다각형 꼭짓점 수 (P)
P*4 bytes	서비스 구역 다각형 꼭짓점 구조체 목록 (아래 참조)

서비스 구역 다각형 꼭짓점 구조체:

타입	설명
int16_t	X, cm
int16_t	Y, cm

10.4.27. 서브맵 설정 지정

Modem에 서브맵 설정을 기록합니다.

함수 이름: mm_set_submap_settings

C 선언: `bool mm_set_submap_settings (uint8_t submapId , void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: MMSW0005

반환 값:

타입	설명
bool	true – 함수가 성공적으로 실행되었으며, 서브맵 설정이 변경됨 false – 실행 오류

매개변수:

타입	설명
uint8_t	서브맵 ID (0...254)
void *	기록할 데이터에 대한 포인터 ('get submap settings' 함수 참조)

10.4.28. Freeze map

서브맵을 고정합니다.

함수 이름: `mm_freeze_map`

C 언어 선언: `bool mm_freeze_map ();`

사용 가능한 API 버전: V4+

필요한 라이선스: MMSW0005

반환값:

타입	설명
bool	true – 함수가 성공적으로 실행됨, 맵이 동결됨 false – 실행 오류

10.4.29. 맵 동결 해제

서브맵을 동결합니다.

함수 이름: `mm_unfreeze_map`

C 선언: `bool mm_freeze_map ();`

사용 가능한 API 버전: V4+

필요한 라이선스: MMSW0005

반환값:

타입	설명
bool	true – 함수가 성공적으로 실행됨, 맵 동결이 해제됨 false – 실행 오류

10.4.30. 비콘의 초음파 설정 가져오기

지정된 비콘에서 초음파 설정을 읽습니다.

함수 이름: `mm_get_ultrasound_settings`

C 언어 선언: `bool mm_get_ultrasound_settings (uint8_t address , void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며 초음파 설정이 검색됨 false - 실행 오류

매개변수:

타입	설명
uint8_t	비콘의 주소 (1...254)
void *	채워질 데이터에 대한 포인터

포인터를 통해 반환되는 데이터 구조:

타입	설명
uint16_t	초음파 TX 주파수 (DSP RX 전용 비콘에는 해당 없음)
uint8_t	TX 주기 수 (DSP RX 전용 비콘에는 해당 없음)
bool	true = RX에 AGC 사용 false = RX에 수동 게인 사용
uint16_t	수동 게인 값 (0...4000)
bool	true = 센서 RX1이 일반 모드에서 활성화됨
bool	true = 센서 RX2가 일반 모드에서 활성화됨
bool	true = 센서 RX3이 일반 모드에서 활성화됨
bool	true = 센서 RX4가 일반 모드에서 활성화됨
bool	true = 센서 RX5가 일반 모드에서 활성화됨
bool	true = Sensor RX1이 frozen 모드에서 활성화됨
bool	true = Sensor RX2가 frozen 모드에서 활성화됨
bool	true = Sensor RX3이 frozen 모드에서 활성화됨
bool	true = Sensor RX4가 frozen 모드에서 활성화됨
bool	true = Sensor RX5가 frozen 모드에서 활성화됨
uint8_t	DSP RX 필터의 인덱스 (DSP 비콘에만 해당) 0 = 19 kHz 1 = 25 kHz 2 = 31 kHz 3 = 37 kHz 4 = 45 kHz

10.4.31. 비콘의 초음파 설정 지정

지정된 비콘에 초음파 설정을 씁니다.

함수 이름: `mm_set_ultrasound_settings`

C 선언: `bool mm_set_ultrasound_settings (uint8_t address, void *pdata);`

사용 가능한 API 버전: V1+

필요한 라이선스: MMSW0005

반환값:

유형	설명
bool	true – 함수가 성공적으로 실행되었으며, 초음파 설정이 변경됨 false – 실행 오류

매개변수:

유형	설명
uint8_t	비콘의 주소 (1...254)
void *	기록할 데이터에 대한 포인터 ('초음파 설정 가져오기' 함수 참조).

10.4.32. 맵 삭제

Modem의 맵 삭제 – 모든 서브맵 제거 (서브맵 0 제외), 서브맵 0을 초기 상태로 재설정, 네트워크에서 연결된 모든 비콘 제거.

함수 이름: mm_erase_map

C 선언: bool mm_erase_map ();

사용 가능한 API 버전: V1+

필요한 라이선스: MMSW0005

반환 값:

타입	설명
bool	true – 함수가 성공적으로 실행되었으며, 맵이 삭제됨 false – 실행 중 오류 발생

매개변수: 없음

10.4.33. 장치를 기본 설정으로 초기화

장치를 기본 설정(라디오, 초음파 등)으로 초기화합니다.

함수 이름: `mm_set_default_settings`

C 언어 선언: `bool mm_set_default_settings (uint8_t address);`

사용 가능한 API 버전: V1+

필요한 라이선스: MMSW0005

반환 값:

타입	설명
bool	true – 함수가 성공적으로 실행되었으며, 장치가 기본 설정으로 초기화됨 false – 실행 오류

매개변수:

유형	설명
uint8_t	장치의 주소 (1...254) 255 – USB를 통해 연결된 장치를 기본값으로 초기화

10.4.34. 비콘을 축에 연결

선택한 비콘이 축 위에 위치하도록 맵을 이동합니다.

함수 이름: mm_beacons_to_axes

C 선언: bool mm_beacons_to_axes (uint8_t address_0, uint8_t address_x, uint8_t address_y);

사용 가능한 API 버전: V2+

필요한 라이선스: MMSW0005

반환값:

유형	설명
bool	true - 함수가 성공적으로 실행되었으며 맵이 이동됨 false - 실행 오류

매개변수:

유형	설명
uint8_t	address_0 - 중심(X=0, Y=0)에 위치해야 하는 beacon의 주소
uint8_t	address_x - X축 방향(Y=0)에 위치해야 하는 beacon의 주소
uint8_t	address_y - Y축 양의 방향(Y>0)에 위치해야 하는 beacon의 주소

10.4.35. Modem 구성 메모리 덤프 읽기

Modem의 구성 메모리 덤프를 읽습니다. Modem의 설정 및 저장된 맵을 저장할 수 있습니다.

함수 이름: `mm_read_flash_dump`

C 언어 선언: `bool mm_read_flash_dump(uint32_t offset, uint32_t size, void *pdata);`

사용 가능한 API 버전: v3 이상

필요한 라이선스: MMSW0005

반환값:

유형	설명
bool	true - 함수가 성공적으로 실행되었으며 덤프를 읽었음 false - 실행 오류

매개변수:

타입	설명
uint32_t	offset - 구성 메모리 시작 지점으로부터의 오프셋, 바이트 단위
uint32_t	size - 읽을 데이터 크기, 바이트 단위
void *	pdata - 데이터 수신을 위한 사용자 버퍼에 대한 포인터

10.4.36. Modem 구성 메모리의 덤프 쓰기

Modem 구성 메모리에 데이터 덤프를 씁니다. Modem의 설정 및 맵을 복원할 수 있습니다.

함수 이름: `mm_write_flash_dump`

C 언어 선언: `bool mm_write_flash_dump(uint32_t offset, uint32_t size, void *pdata);`

사용 가능한 API 버전: V3+

필요한 라이선스: MMSW0005

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 덤프가 기록됨 false - 실행 오류

매개변수:

타입	설명
uint32_t	offset - 구성 메모리 시작 지점으로부터의 오프셋, 바이트 단위 올바른 동작을 위해 offset은 4096바이트 페이지에 정렬되어야 합니다 (값 0, 4096, 8192 등).
uint32_t	size - 쓸 데이터의 크기, 바이트 단위
void *	pdata - 데이터가 담긴 사용자 버퍼에 대한 포인터

참고: 구성을 쓴 후에는 새 설정을 적용하고 덮어쓰기를 방지하기 위해 Modem의 소프트웨어 리셋(`mm_reset_device(255)`)을 권장합니다.

10.4.37. 장치 재시작 (소프트 리셋)

지정된 장치에 대해 소프트웨어 리셋을 실행합니다.

함수 이름: `mm_reset_device`

C 언어 선언: `bool mm_reset_device (uint8_t address);`

사용 가능한 API 버전: V3+

필요한 라이선스: 없음

반환값:

타입	설명
<code>bool</code>	<code>true</code> - 함수가 성공적으로 실행되었으며, 장치가 재설정 중 <code>false</code> - 실행 오류

매개변수:

유형	설명
<code>uint8_t</code>	장치 주소 (1...254) 255 - USB를 통해 연결된 장치의 소프트웨어 재설정

10.4.38. Modem에서 공기 온도 설정 읽기

Modem에서 공기 온도 설정값(섭씨)을 읽습니다.

함수 이름: mm_get_air_temperature

C 선언: `bool mm_get_air_temperature (void *pdata);`

사용 가능한 API 버전: V3+

필요한 라이선스: 없음

반환값:

유형	설명
bool	true - 함수가 성공적으로 실행되었으며, 온도가 반환됨 false - 실행 중 오류 발생

pdata 포인터를 통해 반환되는 데이터 구조:

타입	설명
int8_t	공기 온도, 섭씨(Celsius) 단위

10.4.39. Modem에 공기 온도 설정 쓰기

Modem에 공기 온도 설정(섭씨 단위)을 구성합니다.

함수 이름: mm_set_air_temperature

C 언어 선언: bool mm_set_air_temperature (void *pdata);

사용 가능한 API 버전: V3 이상

필요한 라이선스: 없음

반환 값:

타입	설명
bool	true - 함수가 성공적으로 실행되었으며, 온도가 기록됨 false - 실행 중 오류 발생

사용자가 pdata 포인터를 통해 제공해야 하는 데이터 구조:

유형	설명
int8_t	공기 온도, 섭씨(°C)

10.4.40. Beacon 실시간 플레이어 설정 가져오기

Beacon의 실시간 플레이어 설정을 읽습니다.

함수 이름: mm_get_realtime_player_settings

C 선언: bool mm_get_realtime_player_settings (uint8_t address, void *pdata);

사용 가능한 API 버전: V6+

필요한 라이선스: 없음

반환값:

유형	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

유형	설명
uint8_t	address - 비콘의 주소 (1...254)
void *	pdata - 채워질 데이터에 대한 포인터

포인터를 통해 반환되는 데이터의 구조:

타입	설명
bool	true = 실시간 플레이어가 활성화됨
uint8_t	처리할 실시간 플레이어 순방향 샘플 수
uint8_t	처리할 실시간 플레이어 역방향 샘플 수
uint8_t	예약됨 (0)
uint8_t	예약됨 (0)

10.4.41. 비콘 실시간 플레이어 설정 지정

비콘의 실시간 플레이어 설정을 구성합니다.

함수 이름: mm_set_realtime_player_settings

C 선언: bool mm_set_realtime_player_settings (uint8_t address, void *pdata);

API 버전 지원: V6+

필요한 라이선스: MMSW0005

반환 값:

타입	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

타입	설명
uint8_t	address - 비콘의 주소 (1...254)
void *	pdata - 쓸 데이터에 대한 포인터 ('Get beacon real-time player settings' 함수 참조)

10.4.42. 지리 참조 설정 가져오기

지리 참조 설정(Marvelmind 맵의 점 (X=0, Y=0)의 지리적 위치)을 읽습니다.

함수 이름: mm_get_georeferencing_settings

C 언어 선언: bool mm_get_georeferencing_settings (void *pdata);

API 버전 지원: V6+

라이선스 필요: 없음

반환 값:

타입	설명
bool	true - 함수 실행 성공 false - 실행 오류

매개변수:

타입	설명
void *	pdata - 채울 데이터에 대한 포인터

포인터를 통해 반환된 데이터의 구조:

타입	설명
int32_t	위도, x10-7 도
int32_t	경도, x10-7 도

10.4.43. 지리참조 설정 지정

Marvelmind 맵의 점(X=0, Y=0)의 지리적 위치에 대한 지오레퍼런싱 설정을 구성합니다.

함수 이름: mm_set_georeferencing_settings

C 선언: `bool mm_set_georeferencing_settings (void *pdata);`

사용 가능한 API 버전: V6+

필요한 라이선스: MMSW0005

반환 값:

유형	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

유형	설명
void *	pdata - 쓸 데이터에 대한 포인터('Get georeferencing settings' 함수 참조)

10.4.44. 위치 업데이트 모드 가져오기

모바일 비콘의 현재 위치 업데이트 모드를 읽습니다.

함수 이름: `mm_get_update_position_mode`

C 선언: `bool mm_get_update_position_mode (void *pdata);`

사용 가능한 API 버전: V6+

필요한 라이선스: 없음

반환 값:

타입	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

타입	설명
void *	pdata - 채워질 데이터에 대한 포인터

포인터를 통해 반환되는 데이터 구조:

타입	설명
uint8_t	모바일 비콘의 위치 업데이트 모드: 0 - 자동 위치 업데이트 (기본 모드) 1 - 다음 업데이트 주기에 사용자 요청에 의한 위치 업데이트 2 - 사용자 요청 시 즉시 위치 업데이트
7 바이트	예약됨 (0)

10.4.45. 위치 업데이트 모드 설정

모바일 비콘의 위치 업데이트 모드를 설정합니다.

함수 이름: `mm_set_update_position_mode`

C 선언: `bool mm_set_update_position_mode (void *pdata);`

사용 가능한 API 버전: V6+

필요한 라이선스: MMSW0005

반환 값:

타입	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

타입	설명
void *	pdata - 쓸 데이터에 대한 포인터 (함수 'Get mode of updating positions' 참조)

10.4.46. 위치 업데이트 명령

모바일 비콘의 위치 업데이트 명령을 전송합니다 (업데이트 모드가 자동이 아닌 경우).

함수 이름: mm_set_update_position_command

C 언어 선언: bool mm_set_update_position_command (void *pdata);

사용 가능한 API 버전: V6+

필요한 라이선스: MMSW0005

반환 값:

타입	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

타입	설명
void *	pdata - 쓸 데이터에 대한 포인터

포인터가 가리키는 데이터 구조:

타입	설명
8 바이트	예약됨 (0)

10.4.47. 비콘의 지오펜싱 알람 상태 설정

Super-Beacon의 알람 핀에 알람 상태를 설정하는 명령을 전송합니다.

핀의 알람 상태는 Dashboard 설정의 'Interfaces' 섹션에 있는 'Alarm pin mode' 설정을 통해 지정할 수 있습니다 (MMSW0006 라이선스가 활성화된 경우).

함수 이름: mm_set_alarm_state

C 언어 선언: bool mm_set_alarm_state (uint8_t address, void *pdata);

사용 가능한 API 버전: V9+

필요한 라이선스: MMSW0005, MMSW0006

반환값:

유형	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

유형	설명
uint8_t	address - 비콘의 주소 (1...254)
void *	pdata - 쓸 데이터에 대한 포인터

포인터에 의한 데이터 구조:

타입	설명
uint8_t	알람 핀 모드: 0 - 핀이 지오펜싱 상태에 따라 자동으로 제어됨 1 - 핀이 수동으로 제어됨 - 알람 상태 없음 2 - 핀이 수동으로 제어됨 - 알람 상태
uint8_t	지오펜싱 구역 인덱스 - 비콘이 알람 상태에서 스트리밍할 지오펜싱 구역의 번호
6 바이트	예약됨 (0)

10.4.48. 일반 사용자 페이로드 데이터 전송

일반 사용자 페이로드 데이터를 전송합니다. API가 Modem에 연결된 경우, 데이터는 지정된 모바일 비콘의 UART/USB 포트를 통해 전송됩니다. API가 모바일 비콘에 연결된 경우, 데이터는 Modem의 UART/USB 포트를 통해 전송됩니다. 수신된 데이터는 수신 API 함수, Arduino 예제, ROS 및 기타 소프트웨어를 통해 원격 측에서 사용할 수 있습니다.

함수 이름: mm_send_user_payload_data

C 선언: bool mm_send_user_payload_data (uint8_t address, void *pdata);

사용 가능한 API 버전: V9+

필요한 라이선스: MMSW0005

반환 값:

타입	설명
bool	true - 함수 실행 성공 false - 실행 오류

매개변수:

타입	설명
uint8_t	address - Modem에 API가 연결된 경우 비콘의 주소 (1...254) API가 비콘에 직접 연결된 경우 해당 없음
void *	pdata - 쓸 데이터에 대한 포인터

포인터가 가리키는 데이터 구조:

타입	설명
uint8_t	전송할 데이터 크기
256 bytes	전송할 일반 데이터 버퍼

10.4.49. 일반 사용자 페이로드 데이터 가져오기

전송 API 함수, Arduino, ROS 또는 기타 사용자 소프트웨어에 의해 전송된 일반 사용자 페이로드 데이터를 수신합니다. API가 Modem에 연결된 경우, 이 함수는 모바일 비콘의 UART/USB 포트를 통해 전송된 데이터를 수신할 수 있습니다. API가 모바일 비콘에 연결된 경우, 이 함수는 Modem을 통해 전송된 데이터를 수신할 수 있습니다.

함수 이름: mm_get_user_payload_data

C 언어 선언: bool mm_get_user_payload_data (void *pdata);

사용 가능한 API 버전: V9+

필요한 라이선스: MMSW0005

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

타입	설명
void *	pdata - 수신할 데이터에 대한 포인터

포인터로 수신된 데이터의 구조:

타입	설명
uint8_t	address - beacon의 주소
int64_t	데이터 전송 타임스탬프 - 01.01.1970부터의 밀리초 수 (Unix 시간)
uint8_t	전송할 데이터 크기
256 bytes	수신된 데이터의 버퍼

10.4.50. 수동 거리 측정을 위한 명령 전송

시스템 내 지정된 beacon에서 다른 beacon까지의 거리를 측정하는 명령을 전송합니다. 현재 소프트웨어 버전에서는 IA (Inverse architecture)에서 지원됩니다.

함수 이름: mm_send_distances_measurement_command

C 언어 선언: bool mm_send_distances_measurement_command (void *pdata);

사용 가능한 API 버전: V9+

필요한 라이선스: MMSW0011

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 오류

매개변수:

유형	설명
void *	pdata - 전송할 데이터에 대한 포인터

포인터에 의한 데이터 구조:

유형	설명
uint8_t	모드: 0 - 자동 1 - 수동 (이 명령에 의해)
uint8_t	비콘의 주소
uint32_t	측정할 최대 거리, mm
8 바이트	예약됨

10.4.51. Modem에서 스트리밍 데이터 가져오기

이전에 설명한 형식으로 Modem의 스트리밍 데이터를 읽습니다.

함수 이름: mm_get_stream_data

C 언어 선언: bool mm_get_stream_data (void *pdata);

사용 가능한 API 버전: V9+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true - 함수가 성공적으로 실행됨 false - 실행 중 오류 발생

매개변수:

타입	설명
void *	pdata - 전송할 데이터에 대한 포인터

포인터에 의한 데이터 구조:

타입	설명
uint8_t	이 응답의 스트림 레코드 수 (0...16)
138*16 바이트	138 바이트씩 16개의 스트리밍 레코드 (아래 참조)
8 바이트	예약됨

스트림 레코드 구조:

타입	설명
uint8_t	레코드 크기, 바이트
uint8_t	레코드 타입. Dashboard 로그 파일의 'line type'과 동일한 값. 예를 들어, 41은 Marvelmind 프로토콜 데이터를 의미함
8 바이트	예약됨
128 바이트	스트림 레코드 데이터

10.4.52. 장치 타입이 Modem인지 확인

지정된 장치 타입이 Modem에 해당하는지 확인합니다.

함수명: `mm_device_is_modem`

C 선언: `bool mm_device_is_modem (uint8_t deviceType);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

타입	설명
bool	true – 지정된 장치 유형이 Modem에 해당함

매개변수:

타입	설명
uint8_t	확인할 장치 유형

10.4.53. 장치 유형이 고정 비콘인지 확인

지정된 장치 유형이 고정 비콘에 해당하는지 확인합니다.

함수 이름: mm_device_is_beacon

C 선언: `bool mm_device_is_beacon (uint8_t deviceType);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

유형	설명
bool	true – 지정된 장치 유형이 고정 비콘에 해당함

매개변수:

유형	설명
uint8_t	확인할 장치 유형

10.4.54. 장치 유형이 Hedgehog인지 확인

지정된 장치 유형이 Hedgehog에 해당하는지 확인합니다.

함수 이름: mm_device_is_hedgehog

C 선언: `bool mm_device_is_hedgehog (uint8_t deviceType);`

사용 가능한 API 버전: V1+

필요한 라이선스: 없음

반환값:

유형	설명
bool	true – 지정된 장치 유형이 Hedgehog에 해당함

매개변수:

타입	설명
uint8_t	확인할 장치 유형

10.5. Marvelmind API C 예제 설명

C 예제는 Marvelmind API 테스트에 사용되며, 사용자 애플리케이션 구축의 기초로 활용할 수 있습니다.

C 예제는 콘솔 애플리케이션입니다. 다음 플랫폼에서 테스트되었습니다:

- CPU: Intel Core 2 Duo, OS: MS Windows XP;
- CPU: Intel Core i5, OS: Linux Ubuntu 16.04;
- Raspberry Pi 3 Model B+, OS: Raspbian (2018-11-13-raspbian-stretch-full)

Windows 플랫폼에서는 CodeBlocks IDE를 사용하여 예제를 빌드하였으므로, 예제에 CodeBlocks 프로젝트 파일이 포함되어 있습니다.

Linux 플랫폼에서는 make 유틸리티를 사용하여 예제를 빌드하였으므로, 예제에 이를 위한 makefile이 포함되어 있습니다.

예제에는 다음 모듈이 포함되어 있습니다:

파일명	설명
main.c	main() 함수가 포함된 모듈. 예제 함수를 호출하고 간단한 명령줄 인터페이스를 구현합니다.
marvelmind_example.c marvelmind_example.h	marvelmindStart() – 예제 초기화 marvelmindFinish() – API 작업 완료 후 호출 marvelmindCycle() – 메인 루프에서 자주 호출됨 또한, 모듈에는 사용자가 입력한 명령을 처리하기 위한 여러 함수가 포함되어 있습니다.
marvelmind_api.c marvelmind_api.h	marvelmindAPILoad() – API 라이브러리 로드 marvelmindAPIFree() – API 라이브러리가 사용한 메모리 해제 API 라이브러리와 통신에 관한 모든 함수.
marvelmind_devices.c marvelmind_devices.h	Modem에서 'get devices list' 명령 호출을 통해 수신된 비콘 목록을 지원합니다. 각 비콘에는 해당 위치 및 다른 비콘까지의 거리 데이터가 포함됩니다.
marvelmind_pos.c marvelmind_pos.h	최신 위치 데이터 및 최신 원시 거리 데이터를 읽습니다. 이 데이터를 장치 목록에서 업데이트합니다.
marvelmind_utils.c marvelmind_utils.h	다른 모듈에서 사용되는 일부 보조 함수.

예제 동작 방식:

1. 성공할 때까지 시리얼 포트 열기 시도
2. 포트가 열리면 프로그램은 USB를 통해 연결된 장치의 버전을 읽습니다. 연결된 장치가 Modem인 경우 프로그램은 다음 단계를 계속 실행합니다.
3. Modem에 연결되면 프로그램은 1 Hz 속도로 장치 목록을 읽습니다. 읽어온 장치 목록은 marvelmind_devices.c 모듈에 현재 저장된 목록과 비교되며, 변경 사항이 감지되면 marvelmind_devices.c의 목록이 업데이트됩니다. 모든 변경 사항은 콘솔에 출력됩니다.
4. Modem에 연결되면 프로그램은 20 Hz 속도로 최신 위치 데이터를 읽습니다. 새로운 원시 거리 데이터 플래그가 설정되어 있으면 프로그램은 최신 원시 거리 데이터를 읽습니다. 프로그램은 위치 및 거리 데이터를 marvelmind_devices.c의 장치 목록에 있는 데이터와 비교하고, 변경된 경우 데이터를 업데이트합니다. 변경된 모든 데이터는 콘솔에 출력됩니다.
5. 프로그램이 최신 위치 데이터를 10회 연속으로 가져오지 못하면 포트를 닫고 1단계로 돌아가 포트를 다시 열기를 시도합니다. 포트 재열기는 Modem이 USB에서 분리되었다가 다시 연결된 경우를 처리하기 위해 필요합니다.

6. 사용자가 'space' 키를 누르면 프로그램은 'Enter command: ' 메시지를 표시하고 사용자 명령을 기다립니다. 대부분의 API 함수는 사용자 명령을 통해 호출됩니다. 자세한 내용은 아래를 참조하십시오.

사용자 명령:

프로그램 실행 중 사용자가 'space' 키를 누르면 프로그램은 'Enter command: ' 메시지를 표시합니다. 사용자는 키보드로 명령을 입력하고 엔터 키를 눌러야 합니다.

아래 표에는 모든 사용자 명령의 형식이 포함되어 있습니다:

명령 그룹	설명
API 버전	명령 형식: version 동작: API 라이브러리의 버전을 출력합니다.
프로그램 종료	명령 형식: quit 동작: 프로그램 실행을 종료합니다.
슬립/웨이크	명령 형식: wake <address> 동작: 웨이크 명령을 실행합니다. 예시: wake 5 - 장치 5를 깨우는 명령 전송 wake 0 - 모든 장치를 깨우는 명령 전송
	명령 형식: sleep <address> 동작: 슬립 명령 전송을 실행합니다. 예시: sleep 5 - 장치 5를 슬립 상태로 전환 sleep 0 - 모든 장치를 슬립 상태로 전환
기본값	명령 형식: default <address> 동작: 기본 설정으로 초기화 명령을 실행합니다. 예시: default 5 - 장치 5의 기본 설정 적용
텔레메트리 읽기	명령어 형식: tele <address> 동작: 비콘의 텔레메트리 데이터를 읽고 출력합니다. 예시: tele 5 - 비콘 5의 텔레메트리 읽기 및 출력
서브맵 명령어	명령어 형식: submap add <submapId> 동작: 지정된 서브맵 ID로 서브맵을 추가하는 명령어를 실행합니다. 예시: submap add 1 - 서브맵 1 추가
	명령어 형식:

	<p>submap delete <submapId> 동작: 지정된 서브맵 ID로 서브맵을 삭제하는 명령어를 실행합니다. 예시: submap delete 1 - 서브맵 1 삭제</p>
	<p>명령어 형식: submap freeze <submapId> 동작: 지정된 서브맵 ID를 가진 서브맵을 고정(freeze)하는 명령어를 실행합니다. 예시: submap freeze 0 - 서브맵 0 고정(freeze)</p>
	<p>명령어 형식: submap unfreeze <submapId> 동작: 지정된 서브맵 ID를 가진 서브맵의 고정을 해제(unfreeze)하는 명령어를 실행합니다. 예시: submap unfreeze 0 - 서브맵 0 고정 해제(unfreeze)</p>
	<p>명령어 형식: submap get <submapId> 동작: 지정된 서브맵 ID를 가진 서브맵의 설정을 가져오는 명령어를 실행합니다. 예시: submap get 0 - 서브맵 0의 설정을 가져와 출력</p>
	<p>명령어 형식: submap testset <submapId> 동작: 지정된 submap ID를 가진 submap의 설정을 지정하는 명령어를 실행합니다. 프로그램은 명령어 테스트를 위해 미리 정의된 설정을 기록합니다. 예제 코드를 참조하십시오. 예제: submap testset 0 - submap 0의 설정 수정</p>
Map 명령어	<p>명령어 형식: map erase 동작: map 삭제 명령어를 실행합니다. 예제: map erase - Modem의 map 삭제</p>
	<p>명령어 형식: map freeze 동작: map 고정 명령어를 실행합니다. 예제: map freeze - map 고정</p>

	<p>명령어 형식: map unfreeze 동작: 맵 고정 해제 명령을 실행합니다. 예시: map unfreeze - 맵 고정 해제</p>
업데이트 속도 명령	<p>명령어 형식: rate get 동작: 업데이트 속도 설정 읽기 명령을 실행합니다. 예시: rate get - 업데이트 속도 설정 읽기 및 출력</p>
	<p>명령어 형식: rate set <value> 동작: 업데이트 속도 설정 변경 명령을 실행합니다. 값은 Hz 단위로 입력합니다. 예시: rate set 0.5 - 업데이트 속도 0.5 Hz로 설정</p>
초음파 명령	<p>명령어 형식: usound get <address> 동작: 지정된 비콘의 초음파 설정을 읽기 실행합니다. 예시: usound get 5 - 비콘 5의 초음파 설정을 읽고 출력</p>
	<p>명령어 형식: usound testset <address> 동작: 지정된 비콘의 초음파 설정을 쓰기 실행합니다. 프로그램은 명령어 테스트를 위해 미리 정의된 설정을 기록합니다. 예제 코드를 참조하십시오. 예시: usound testset 5 - 비콘 5의 초음파 설정 수정</p>
축 연결 명령어	<p>명령어 형식: axes <address_0> <address_x> <address_y> 동작: 비콘을 축에 연결하는 명령어를 실행합니다. 예시: axes 3 4 5 - 비콘 3을 X=0, Y=0으로 설정; 비콘 4는 X축 방향(Y=0), 비콘 5는 X축 위쪽(Y>0)</p>
Modem에서 구성 메모리 덤프 읽기	<p>명령어 형식: read_dump <offset> <size> 동작: Modem 구성 메모리 읽기 덤프 명령을 실행합니다. 예시: read_dump 0 1000 - 구성 메모리 시작 부분부터 첫</p>

	1000 바이트를 읽습니다
구성 메모리 테스트 덤프를 Modem에 씁니다	명령 형식: write_dump_test <offset> <size> 동작: Modem 구성 메모리 쓰기 덤프 명령을 실행합니다. 예시: write_dump_test 0 1000 - 구성 메모리 시작 부분부터 첫 1000 바이트를 테스트 패턴으로 채웁니다
장치의 소프트웨어 리셋	명령 형식: reset <address> 동작: 소프트웨어 리셋 명령을 실행합니다. 예시: reset 255 - USB를 통해 연결된 장치에 대해 소프트웨어 리셋을 실행합니다
공기 온도 명령	명령 형식: temperature get 동작: Modem에서 공기 설정의 온도 읽기 실행 예시: temperature get 공기 설정의 초음파 온도를 읽어 출력
	명령 형식: temperature set <value> 동작: Modem에 공기 설정의 온도 쓰기 실행 예시: temperature set 30 공기 설정의 온도를 섭씨 30 도로 설정
Beacon 위치 설정	명령 형식: setloc <address> <X> <Y> <Z> 동작: Beacon의 위치 설정 명령 실행. X, Y, Z는 미터 단위의 좌표입니다. 예시: setloc 12 1.51 3.45 2.0 - Beacon 12의 위치를 X= 1.51 m, Y= 3.45 m, Z= 2.0 m로 설정
Beacon 간 거리 설정	명령어 형식: setdist <address1> <address2> <distance> 동작: 비콘 간 거리 설정 명령을 실행합니다. Address1과 address2는 비콘의 주소입니다. Distance는 미터 단위의 거리입니다. 예시: setdist 12 13 16.5 - 비콘 12와 13 사이의 거리를 16.5 미터로 설정합니다

<p>높이 명령어</p>	<p>명령어 형식: height_h get <address> 동작: Hedgehog 높이 가져오기 명령을 실행합니다. Address는 Hedgehog의 주소입니다. 예시: height_h get 15 - Hedgehog 15의 높이를 읽고 출력합니다</p>
	<p>명령어 형식: height_h set <address> <height> 동작: Hedgehog 높이 설정 명령을 실행합니다. Address는 Hedgehog의 주소입니다. Height는 미터 단위입니다. 예시: height_h set 15 2.5 - Hedgehog 15의 높이를 2.5 미터로 설정합니다</p>
	<p>명령어 형식: height_b get <address> <submap_id> 동작: 고정 비콘 높이 조회 명령을 실행합니다. Address는 비콘의 주소입니다. Submap_id는 비콘이 속한 서브맵의 ID입니다. 예시: height_b get 12 0 - 서브맵 0에 있는 고정 비콘 12의 높이를 읽고 출력합니다.</p>
	<p>명령어 형식: height_b set <address> <submap_id> <height> 동작: 고정 비콘 높이 설정 명령을 실행합니다. Address는 Hedgehog의 주소입니다. Submap_id는 비콘이 속한 서브맵의 ID입니다. 높이 단위는 미터(m)입니다. 예시: height_b set 12 0 5.1 - 서브맵 0에 있는 비콘 12의 높이를 5.1m로 설정합니다.</p>
<p>실시간 플레이어 명령어</p>	<p>명령어 형식: rtp get <address> 동작: 실시간 플레이어 설정 조회 명령을 실행합니다. Address는 비콘의 주소입니다. 예시: rtp get 15 - 비콘 15의 실시간 플레이어 설정을 읽고 출력합니다.</p>
	<p>명령어 형식: rtp testset <address> 동작: 실시간 플레이어 설정 명령을 실행합니다. Address는 비콘의 주소입니다. 프로그램은 명령 테스트를 위해 미리 정의된 일부 설정을 작성합니다. 예제 코드를 참조하십시오.</p>

	<p>예제: rtp testset 15 - 비콘 15에 대한 실시간 플레이어 테스트 설정 구성</p>
지오레퍼런싱 명령	<p>명령 형식: georef get 동작: 지오레퍼런싱 설정 가져오기 명령을 실행합니다. 예제: georef get - 지오레퍼런싱 설정을 읽고 출력합니다</p>
	<p>명령 형식: georef set <latitude> <longitude> 동작: 지오레퍼런싱 설정 명령을 실행합니다. 예제: georef set 10 20 - 위도 10도, 경도 20도의 지오레퍼런싱을 저장합니다</p>
업데이트 모드 명령	<p>명령 형식: update_mode get 동작: 위치 업데이트 모드 가져오기 명령을 실행합니다. 예시: update_mode get - 위치 업데이트 모드를 읽고 출력합니다</p>
	<p>명령 형식: update_mode set <mode> 동작: 위치 업데이트 모드 설정 명령을 실행합니다. 예시: update_mode set 0 - 위치 업데이트의 자동 모드를 설정합니다</p>
	<p>명령 형식: update 동작: 위치 업데이트 명령을 실행합니다. 예시: update - 현재 모드에 따라 모바일 비콘의 위치를 업데이트합니다</p>
지오펜싱 알람 상태 설정	<p>명령 형식: alarm <address> <mode> <zone> 동작: 지오펜싱 알람 상태 설정 명령을 실행합니다. 예시: alarm 10 2 5 - 지오펜싱 구역 번호 5로 비콘 n10에 지오펜싱 알람 신호 설정</p>
사용자 페이로드 명령	<p>명령 형식: payload read <address> 동작: 사용자 페이로드 데이터 가져오기 명령을 실행합니다. 예시: payload read - 임의의 비콘/Modem에서 사용자 페이로드 읽기 예시: payload read 10 - 비콘 n10에서 사용자 페이로드 읽기</p>

	<p>명령 형식: payload write <address> 동작: 사용자 페이로드 데이터 전송 명령을 실행합니다. 예시: payload write 10 - 비콘 n10에 테스트 페이로드 데이터 쓰기 테스트 패턴은 100부터 시작하는 40바이트입니다: 100, 101, ..., 139</p>
수동 거리 측정 명령	<p>명령어 형식: distance <manual/auto> <address> <max distance> 동작: 수동 거리 측정 명령을 실행합니다. 예시: distance manual 10 5 - 비콘 10에서 다른 비콘까지의 거리 측정, 최대 거리 5 m 예시: distance manual 10 - 비콘 10에서 다른 비콘까지의 거리 측정, 최대 거리 30 m (기본값) 예시: distance auto - 자동 거리 측정 모드로 복귀</p>

10.6. 장치 유형

특정 장치에 대한 'Device type ID' 값 목록은 다음과 같습니다:

Device type ID	장치 설명
22	Beacon HW V4.5
23	Beacon HW V4.5 (Hedgehog 모드)
24	Modem HW V4.9
30	Beacon HW V4.9
31	Beacon HW V4.9 (Hedgehog 모드)
32	Beacon Mini-RX
36	Beacon Mini-TX
37	Beacon-TX-IP67
41	Beacon industrial-RX
42	Super-Beacon
43	Super-Beacon (hedgehog 모드)
44	Industrial Super-Beacon
45	Industrial Super-Beacon (hedgehog 모드)
46	Super-Modem
48	Modem HW V5.1

장치 유형 ID는 장치 목록 및 장치 버전 읽기 명령에서 확인할 수 있습니다.

11. 사용자 장치로/사용자 장치로부터 사용자 데이터 전송

Marvelmind은 Marvelmind 시스템을 통해 사용자 데이터를 전송하는 다양한 방법을 지원합니다:

- modem의 UART 또는 USB를 통해 데이터를 전송하고, 모바일 beacon의 UART 또는 USB를 통해 수신
- 모바일 beacon의 UART 또는 USB를 통해 데이터를 전송하고, modem의 UART 또는 USB를 통해 수신

Super-Modem은 UDP를 통한 사용자 데이터 전송 및 수신도 지원합니다.

데이터 전송 프로토콜은 본 문서의 이전 섹션에 설명되어 있습니다:

- 사용자 장치로의 데이터 전송 및 사용자 장치로부터의 데이터 전송 프로토콜;
- 데이터 전송 및 수신을 위한 API 함수

Marvelmind은 통신을 위한 다양한 소프트웨어 예제를 제공합니다:

예제	Arduino (UART)	PC / Raspberry Pi (USB)				PC / Raspberry Pi (UDP Super-Modem)
		API	C	Python	ROS/ROS2	C 예제
사용자 장치 → beacon←	+	+	+	+	+	n/a
Modem → 사용자 장치 →	+	+	+	+	+	+
사용자 장치 → beacon→	+	+	-	-	+	n/a
Modem → 사용자 장치 ←	+	+	-	-	+	+

예제 전체 목록:

- 사용자 데이터 송수신을 위한 Arduino 예제는 Marvelmind 소프트웨어 패키지의 '01_Common_Indoor_positioning_SW/ 06_Examples/ arduino' 폴더에 있습니다. 'hedgehog_sample_uart_user_data_receive_v2'는 사용자 데이터 수신용이며, 'hedgehog_sample_uart_user_data_send_v2'는 사용자 데이터 송신용입니다.
- API 통신 예제는 Marvelmind 소프트웨어 패키지의 '01_Common_Indoor_positioning_SW/ 05_API/example_source' 폴더 (소스 코드)와 '01_Common_Indoor_positioning_SW/ 05_API/example_bin_win32' 폴더(Windows용 바이너리)에 있습니다. 데이터 전송 및 수신은 본 문서에 설명된 방법에 따라 호출할 수 있습니다.
- 스트리밍 데이터 수신을 위한 C 예제는 Marvelmind 소프트웨어 패키지의 '01_Common_Indoor_positioning_SW/ 06_Examples/ c' 폴더에 있습니다. 또한 이 예제는 GitHub 저장소에서도 확인할 수 있습니다. 이 예제는 모바일 beacon 또는 Modem으로부터 수신된 사용자 데이터를 포함한 모든 데이터를 단순히 출력합니다.
- 스트리밍 데이터 수신을 위한 Python 예제는 Marvelmind 소프트웨어 패키지의 '01_Common_Indoor_positioning_SW/ 06_Examples/ python' 폴더에 있습니다. 또한 이 예제는 GitHub 저장소에서도 확인할 수 있습니다. 이 예제는 사용자 데이터를 포함하여 모바일 비콘 또는 Modem으로부터 수신된 모든 데이터를 단순히 출력합니다.
- 스트리밍 데이터 수신을 위한 ROS 패키지 예제는 Marvelmind 소프트웨어 패키지의 '01_Common_Indoor_positioning_SW/ 06_Examples/ ROS' 폴더에 있습니다. 또한 이 패키

지는 저장소에서도 확인할 수 있습니다. ROS 패키지를 통해 API로 사용자 데이터를 수신하고 전송할 수 있습니다. 자세한 내용은 문서를 참조하십시오.

- 스트리밍 데이터 수신을 위한 ROS-2 패키지 예제는 **Marvelmind** 소프트웨어 패키지의 '01_Common_Indoor_positioning_SW/ 06_Examples/ ROS2' 폴더에 있습니다. 또한 이 패키지는 저장소에서도 확인할 수 있습니다. ROS 패키지를 통해 API로 사용자 데이터를 수신하고 전송할 수 있습니다. 자세한 내용은 문서를 참조하십시오.
- UDP를 통한 데이터 수신을 위한 C 예제는 **Marvelmind** 소프트웨어 패키지의 '01_Common_Indoor_positioning_SW/ 06_Examples/ c' 폴더에 있습니다. 이 예제는 사용자 데이터를 포함하여 **Super-Modem** 또는 **Dashboard**로부터 UDP를 통해 수신된 모든 데이터를 단순히 출력합니다. UDP를 통한 사용자 데이터 전송은 USB 대신 UDP를 통해 **Super-Modem**에 연결하는 데 API가 사용되는 경우 API를 통해 수행할 수 있습니다.

12. 연락처

추가 지원이 필요하신 경우, info@marvelmind.com으로 문의 사항을 보내주십시오.

부록 1. CRC-16 계산

체크섬에는 CRC-16이 사용됩니다. N바이트 프레임의 마지막 두 바이트는 프레임의 처음 (N-2)바이트에 적용된 CRC-16으로 채워집니다. 데이터를 확인하려면 N바이트의 전체 프레임에 CRC-16을 적용하면 되며, 결과 값은 0이어야 합니다.

다음은 'C'로 구현된 알고리즘입니다:

```
typedef uint16_t ModbusCrc;

typedef union {
    uint16_t w;
    struct{
        uint8_t lo;
        uint8_t hi;
    } b;
    uint8_t bs[2];
} Bytes;

static ModbusCrc modbusCalcCrc(const void *buf, uint16_t length)
{
    uint8_t *arr = (uint8_t *)buf;
    Bytes crc;

    crc.w = 0xffff;

    while(length--){
        char i;
        bool odd;

        crc.b.lo ^= *arr++;
        for(i = 0; i < 8; i++){
            odd = crc.w & 0x01;
            crc.w >>= 1;
            if (odd)
                crc.w ^= 0xa001;
        }
    }
    return (ModbusCrc) crc.w;
}
```

부록 2. Modem의 오류 응답 형식

오류 프레임 형식 (Modem에서 호스트로)

오프셋	크기 (바이트)	유형	설명	값
0	1	uint8_t	Modem의 주소	0xff
1	1	uint8_t	패킷 유형	
2	1	uint8_t	오류 코드	
3	2	uint16_t	CRC-16 (부록 1 참조)	

오류 패킷의 유형은 요청 프레임의 패킷 유형에 상위 비트를 추가한 것입니다. 예를 들어, 요청의 패킷 유형이 0x03인 경우, 오류 패킷의 유형은 0x83이 됩니다.

오류 코드는 다음 중 하나일 수 있습니다:

- 1 - 요청에서 알 수 없는 패킷 유형
- 2 - 요청에서 알 수 없는 데이터 코드
- 3 - 요청의 데이터 필드 오류
- 6 - 장치가 사용 중
- 10 - 원격 장치로부터의 오류 메시지
- 11 - 원격 장치로부터의 응답 시간 초과