

HOME CARE ASSISTANT

Communication Protocol

V3.9

Sep 10th 2025

HISTORY

Ref	Date	Change Reason
v1.0	9/1/2022	Creation
V1.4	11/4/2022	Add Type and Cmd
V1.5	11/21/2022	Add Neg FallDetect
V1.6	11/25/2022	Add Soft reset
V2.1	8/24/2023	Unified description
V2.2	9/1/2023	Unified description
V2.5	11/10/2023	Add presence detection
V2.7	2/2/2024	Add clear room interference, fix errors Room interference study finish
V2.8	2/4/2024	Change Learning room layout position
V2.9	2/6/2024	Add clear room interference remarks
V3.1	2/19/2024	Add ML turn on/off description
V3.3	2/19/2024	Add Room Configuration
V3.4	5/20/2024	Change WiFi signal from 0xFFFF to 0x0414
V3.5	1/11/2025	Update 0x1C position information, add motion detection (firmware version >= 2.8.1.44)
V3.6	2/18/2025	Add num of moving target (firmware version >= 2.8.1.45)
V3.7	6/4/2025	Add descriptions and examples
V3.8	6/25/2025	Add head counting (0x00 0x27)
V3.9	9/10/2025	Update examples

Contents

1. Introduction	1
1.1. Purpose	1
1.2. Updates and maintenance	1
2. Technical Specifications	1
2.1. Frame Format	1
2.2. Data Format Specifications	2
2.3. Communication Direction	2
3. Detailed Command Set Description	2
3.1. Radar ID (0x00 0x01)	4
3.2. Set Installation Height (0x00 0x02)	5
3.3. Get Installation Height (0x00 0x03)	6
3.4. Set Buffer Time (0x00 0x04)	7
3.5. Get Buffer Time (0x00 0x05)	8
3.6. Set Device working range (0x00 0x06)	8
3.7. Get Device working range (0x00 0x07)	9
3.8. Fall detection (0x00 0x09)	10
3.9. Intrusion alert (0x00 0x0a)	11
3.10. Moving heatmap (0x00 0x0b)	12
3.11. Enable collect Heatmap (0x00 0x0c)	12
3.12. Get collect Heatmap status (0x00 0x0d)	13
3.13. Enable Intrusion detections (0x00 0x0e)	14
3.14. Get Intrusion detections (0x00 0x0f)	15
3.15. Learning room layout position (0x00 0x10)	15
3.16. Negative fall detection (0x00 0x11)	16
3.17. Register (0x00 0x12)	17
3.18. Get Firmware version (0x00 0x13)	18
3.19. The radar is reset (0x00 0x14)	19
3.20. Enable Machine Learning (0x00 0x15)	19
3.21. Get Machine Learning (0x00 0x16)	20
3.22. Fall Alert Elimination (0x00 0x17)	21
3.23. Presence Detection (0x00 0x18)	21
3.24. Start room interference study (0x00 0x19)	22
3.25. Room interference study finish (0x00 0x1A)	23
3.26. Room interference clear (0x00 0x1B)	24
3.27. Position and motion detection (0x00 0x1C)	24
3.28. Fall detection Triggered (0x00 0x1D)	25
3.29. Set Fall detection mode (0x00 0x1E)	26
3.30. Get Fall detection mode (0x00 0x1F)	27

3.31. Set Room Configuration (0x00 0x24)	28
3.32. Get Room Configuration (0x00 0x25)	30
3.33. Head counting (0x00 0x27)	31
3.34. Firmware Upgrade	32
3.34.1. Send upgrade parameters (0x00 0x21)	32
3.34.2. Send firmware content in segments (0x00 0x22)	33
3.34.3. Send upgrade completed (0x00 0x23)	34
3.35. Soft reset the sensor (0x04 0x11)	35
3.36. WiFi Signal dBm (0xFF 0xFF)	35

1. Introduction

1.1. Purpose

This protocol aims to standardize the communication interface between the Home Care Assistant and the server, ensuring standardized and reliable data exchange. The primary objectives include:

- Defining a unified communication frame structure and data format
- Providing a complete command set and functional descriptions
- Supporting rapid integration for various application scenarios
- Ensuring system maintainability and extensibility

1.2. Updates and maintenance

We may update this document without previous announcement because of works related to product optimization. If you think your version is out of date, please consult factory to acquire the latest version.

2. Technical Specifications

This is a TCP protocol. This section provides detailed specifications for the communication interface and protocol format.

2.1. Frame Format

Field	Length	Type	Description	Example Value
Magic	1 byte	Hex	Fixed value: 0x12	12
Ver	1 byte	Hex	Fixed value: 0x01	01
Type	1 byte	Enum	Frame type: 0x00: response frame 0x01: request frame 0x02: one-way request frame	01
Cmd	1 byte	Enum	Command type: 0x00: Heartbeat 0x01: Request command 0x02: Response command	01
Req id	4 bytes	int	auto increment from 1 (for request-response matching)	00 00 00 01

Timeout	2 bytes	short	Timeout duration 0-32768	10 27
Content Len	4 bytes	int	Data field length, see section 3	00 00 00 06
Function Code	2 bytes	Hex	Command function identifier (see section 3)	00 02
Data Content	N bytes	Hex	Command-specific data (actual content, N=data length value, see section 3)	3F E6 66 66

Table 1 Basic Frame Format

2.2. Data Format Specifications

- Integer/Floating-point numbers: Big-endian (MSB first), e.g., 1.8 meters → 3F E6 66 66.

- **Status codes:**

0x00 0x00 0x00 0x00: Failed

0x00 0x00 0x00 0x01: Succeed

- **Firmware upgrade:** Segmented transmission (≤ 240 bytes per frame), with CRC16 verification

2.3. Communication Direction

Symbol	Description
↑	radar to host/server
↓	host to radar/server

Table 2 Communication Direction Symbols

3. Detailed Command Set Description

Function Code	Function Description	Remarks
0x00 0x01	Radar ID	Upload every second until a response is received and the connection is considered successful. Radar type: 0x00 (Home Care Assistant), 0x01 (Sleep Sensor)
0x00 0x02	InstallationHeight	Command to set InstallationHeight 1.4m~2.2m
0x00 0x03		Command to get InstallationHeight
0x00 0x04	BufferTime	Command to set the BufferTime to trigger the fall detection alert 30~300 seconds
0x00 0x05		Command to read Home Care Assistant fall detection alert report BufferTime

0x00 0x06	Device working range	Command to set Home Care Assistant working range 1.0~7.0m
0x00 0x07		Command to read Home Care Assistant working range
0x00 0x09	Fall detection	Fall detection alert with fall position, X, Y
0x00 0x0a	Intrusion alert	
0x00 0x0b	Moving heatmap data	The sdk has a heatmap parsing example. For more information, please contact our technical staff.
0x00 0x0c	Enable collect Heatmap	Enable/Disable collect Heatmap 0: Disabled 1: Enable
0x00 0x0d		Read collect Heatmap status
0x00 0x0e	Intrusion detections	Enable/Disable Intrusion detections 0: Disabled 1: Enable
0x00 0x0f		Read Intrusion detection status
0x00 0x10	Learning room layout position	Start/Stop sending room layout (X,Y) 0x01 Start 0x03 Stop If the server does not send a stop command, Home Care Assistant will automatically stop after sending 1000 times
0x00 0x11	Negative fall detection	NegativeFallAlert with fall position, X, Y
0x00 0x12	Register	When the WiFi or the Tcp Socket is resumed from disconnection, the radar will send a registration message to the server automatically to re-establish the mapping relationship. Radar type: 0x00 (Home Care Assistant), 0x01 (Sleep Sensor)
0x00 0x13	Firmware version	Command to request radar firmware version
0x00 0x14	The radar is reset	Once the user resets the radar, the radar will send a reset command to the server(Only work in firmware version <= 1.8.X.X.)
0x00 0x15	Machine Learning	To turn on/off machine learning. The machine learning algorithm is to further reduce the false alerts of fall detection for better user experiences. However, it may result in lower accuracy in certain scenarios. 0: turn on 1: turn off
0x00 0x16		To request the status of whether machine learning is enabled.
0x00 0x17	FallAlertElimination	FallAlertElimination
0x00 0x18	Presence Detection	Upload presence detection range bin and energy
0x00 0x19	Start room interference study	To help to improve the accuracy and user

		experience
0x00 0x1A	Room interference study finish	Response to server to finished study
0x00 0x1B	Room interference clear	To clear room interference learning results (Only work in firmware version >= 2.8.1.9.)
0x00 0x1C	Position and motion detection	Upload user position X, Y, Z and motion detection Upload target position X, Y, Z and motion detection (firmware >= 2.8.1.44) Upload num of moving targets (firmware >= 2.8.1.45)
0x00 0x1D	Fall detection Triggered	Sent immediately after a fall event occurs 0: Failed 1: Succeed
0x00 0x1E	Fall detection mode	Command to set the fall detection mode 0: High sensitivity mode 1: Low false alert mode
0x00 0x1F		Command to get the fall detection mode
0x00 0x24	Room Configuration	Command to set room configuration install type 0: corner, 1: wall num of area Area Type 0 :Door, 1:Sofa, 2:Bed XY[8] x1,y1,x2,y2,x3,y3,x4,y4 Limit[3] detect boundary boundary default value corner install val1=0 or val2=0 wall install (val1=0 and val2 =0) or (val3 = 0)
0x00 0x25		Command to get room configuration
0x00 0x21	Send upgrade parameters	Upgrade command three steps: 1. Send upgrade parameters, radar response; 2. Send firmware content in segments; 3. Send upgrade completed
0x00 0x22	Send firmware content in segments	
0x00 0x23	Send upgrade completed	
0x04 0x11	Soft reset the sensor	Soft reset Home Care Assistant
0xFF 0xFF	WiFi Signal dBm	Upload WiFi signal dBm

Table 3 Radar Module Command Set List

3.1. Radar ID (0x00 0x01)

● Command Description

Status information automatically sent by the radar at regular intervals: Upload every second until a response is received and the connection is considered successful. The radar uploads data once per minute after the server confirms successful connection.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x14	0x00 0x01	↑

Data Content Structure:

Field	Type	Description	Remarks
radar type	u8	Radar type	0x00: Home Care Assistant 0x01: Sleep Sensor
firmware version	U32	Firmware version	
radar id	int13	Radar unique identifier ID	

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 14 00 01 [radar type] + [firmware version] + [radar id]

● Request Example

Home Care Assistant: 00 , firmware version: 2.8.2.3 , ID: 12CC163F000300100800000000

12 01 01 01 00 00 00 01 27 10 00 00 00 14 00 01 00 02 08 02 03 12 CC 16 3F 00 03 00 10 08 00 00 00 00

● Response Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x01	↓

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code	0x00 0x00 0x00 0x01:Acknowledgement

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 01 [Status code]

● Response Example (Success)

12 01 00 02 00 00 00 01 0000 00 00 00 06 00 01 00 00 00 01

3.2. Set Installation Height (0x00 0x02)

● Command Description

Command to set the radar's installation height. Recommended height range: 1.4m~2.2m.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x02	↓

Data Content Structure:

Field	Type	Description	Remarks
InstallationHeight	Float	The sensor installation height	1.4m~2.2m

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 02 [InstallationHeight]

- **Request Example (1.8m)**

12 01 01 01 00 00 00 02 27 10 00 00 00 06 00 02 3F E6 66 66

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x02	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01:Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 02 [Status code]

- **Response Example (Success)**

12 01 00 02 00 00 00 02 00 00 00 00 06 00 02 00 00 00 01

3.3. Get Installation Height (0x00 0x03)

- **Command Description**

Command to get the radar installation height.

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x03	↓

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 03 00 00 00 00

- **Request Example**

12 01 01 01 00 00 00 01 00 00 00 00 06 00 03 00 00 00 00

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x03	↑

Data Content Structure:

Field	Type	Description	Remarks
InstallationHeight	Float	The sensor installation height	1.4m~2.2m

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 03 [InstallationHeight]

● Response Example (1.8m)

12 01 00 02 00 00 00 02 00 00 00 00 06 00 03 3F E6 66 66

3.4. Set Buffer Time (0x00 0x04)

● Command Description

Command to set the buffer time (30~300 seconds) to trigger the fall detection alert. Prevents false alarms from brief falls (e.g., quickly sitting down). If the user resumes standing within the buffer time, the alarm is canceled.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x04	↓

Data Content Structure:

Field	Type	Description	Remarks
BufferTime	Unit32	Fall alert trigger buffer time	Range: 30~300 seconds, e.g.: 30 s , 00 00 00 1E 60 s , 00 00 00 3C 300 s , 00 00 01 2C

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 04 [BufferTime]

● Request Example (Set to 60 seconds)

12 01 01 01 00 00 00 02 00 00 00 00 06 00 04 00 00 00 3C

● Response Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x04	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 04 [Status code]

- **Response Example (Success)**

12 01 00 02 00 00 00 02 00 00 00 00 06 00 04 00 00 00 01

3.5. Get Buffer Time (0x00 0x05)

- **Command Description**

Command to read Sensor fall detection alert report BufferTime.

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x05	↓

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 05 00 00 00 00

- **Request Example**

12 01 01 01 00 00 00 01 00 00 00 00 06 00 05 00 00 00 00

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x05	↑

Data Content Structure:

Field	Type	Description	Remarks
BufferTime	Unit32	Fall alert trigger buffer time	Range: 30~300 seconds, e.g.: 30 s, 00 00 00 1E 60 s, 00 00 00 3C 300 s, 00 00 01 2C

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 05 [BufferTime]

- **Response Example (30 s)**

12 01 00 02 00 00 00 02 00 00 00 00 06 00 05 00 00 00 1E

3.6. Set Device working range (0x00 0x06)

- **Command Description**

Command to set Sensor working range (effective detection range):1.0~7.0m.

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
------	-----	----------------	---------------	-----------

0x01	0x01	0x06	0x00 0x06	↓
------	------	------	-----------	---

Data Content Structure:

Field	Type	Description	Remarks
Working range	Float	Effective detection distance	Range: 1.0~7.0m, e.g.: 1.0m, 3F 80 00 00 4.0m, 40 80 00 00 7.0m, 40 E0 00 00

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 06 [Working range]

● Request Example (4.0m)

12 01 01 01 00 00 00 01 07 D0 00 00 00 06 00 06 40 80 00 00

● Response Frame Format (Req id matches request sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x06	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 06 [Status code]

● Response Example (success)

12 01 00 02 00 00 00 01 07 D0 00 00 00 06 00 06 00 00 00 01

3.7. Get Device working range (0x00 0x07)

● Command Description

Command to read Sensor working range.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x07	↓

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 07 00 00 00 00

● Request Example

12 01 01 01 00 00 00 01 00 00 00 00 06 00 07 00 00 00 00

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x07	↑

Data Content Structure:

Field	Type	Description	Remarks
Working range	Float	Effective detection distance	Range: 1.0~7.0m, e.g.: 1.0m, 3F 80 00 00 4.0m, 40 80 00 00 7.0m, 40 E0 00 00

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 07 [Working range]
```

- **Response Example (7m)**

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 07 40 E0 00 00
```

3.8. Fall detection (0x00 0x09)

- **Command Description**

Radar actively reports fall detection alert with fall position (X/Y axis floating-point numbers). Requires acknowledgement (similar to heartbeat mechanism). If no response is received, the radar will retry reporting up to 6 times at 10-second intervals.

- **Alert Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x0A	0x00 0x09	↓

Data Content Structure:

Field	Type	Description	Remarks
Fall position	Float * 2	Fall position: 4-byte X + 4-byte Y coordinates	Unit: meters (floating-point) Coordinate system: Origin (0,0) at radar position X: horizontal axis Y: vertical axis

Complete structure of the request frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 0A 00 09 [Fall position]
```

- **Alert Example (X=3.0m, Y=5.0m)**

```
12 01 01 01 00 00 00 01 00 00 00 00 00 0A 00 09 40 40 00 00 40 A0 00 00
```

- **Response Frame Format (Req id matches alert sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x09	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code	0x00 0x00 0x00 0x01:Acknowledgement

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 09 [Status code]

● Response Example

12 01 00 02 00 00 00 01 00 00 00 00 06 00 09 **00 00 00 01**

3.9. Intrusion alert (0x00 0x0a)

● Command Description

Radar actively reports intrusion detection alerts: triggered when intrusion behavior is detected.

● Alert Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x0a	↑

Complete structure of the alert frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 0A **00 00 00 00**

● Alert Example

12 01 01 01 00 00 00 01 00 00 00 00 06 00 0A **00 00 00 00**

● Response Frame Format (Req id matches alert sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x0a	↓

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code	0x00 0x00 0x00 0x01:Acknowledgement

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 0A [Status code]

● Response Example

12 01 00 02 00 00 00 01 00 00 00 00 06 00 0A **00 00 00 01**

3.10. Moving heatmap (0x00 0x0b)

- **Command Description**

Radar actively reports moving heatmap data (requires SDK for parsing).

- **Data Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	Variable (N bytes)	0x00 0x0b	↑

Data Content Structure:

Field	Type	Description	Remarks
Heat map data	Variable (N bytes)	The sdk has a heatmap parsing example	For more information, please contact our technical staff.

Complete structure of the data frame

```
12 01 01 01 [Req id] [Timeout] [Content Len] 00 0B [Heat map data]
```

- **Frame Example**

```
12 01 01 01 00 00 00 01 00 XX XX 00 00 00 XX 00 0B 12 01 01 01 00 00 00 0A 27 10 00 00 06 02 00 0B
31 7B 06 00 01 00 0C 6F 09 09 70 03 00 01 00 FF 33 10 0E 31 03 0D 32 03 FF 36 05 03 35 03 00 34 07
00 01 00 F6 36 03 00 01 00 01 34 09 FE 34 06 00 01 00 09 33 07 00 02 00 06 34 07 00 01 00 03 73 03 00
01 00 06 33 04 04 34 08 00 01 00 07 33 03 FF 34 07 00 01 00 09 34 04 00 05 00 04 72 06 FD 70 11 03 6B
10 05 6A
```

- **Response Frame Format (Req id matches data frame sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	Variable (N bytes)	0x00 0x0b	↓

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code	0x00 0x00 0x00 0x01:Acknowledgement

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] [Content Len] 00 0B [Status code]
```

- **Response Example**

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 0B 00 00 00 01
```

3.11. Enable collect Heatmap (0x00 0x0c)

- **Command Description**

Controls heatmap collection: Enables/disables collect heatmap.

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x0c	↓

Data Content Structure:

Field	Type	Description	Remarks
Enable collect Heatmap	Unit32	Heatmap function control flag	0x00 00 00 00: Disabled 0x00 00 00 01: Enabled

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 0C [Enable collect Heatmap]

● Request Example (Enable)

12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 0C 00 00 00 01

● Response Frame Format (Req id matches request sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x0c	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 0C [Status code]

● Response Example (Success)

12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 0C 00 00 00 01

3.12. Get collect Heatmap status (0x00 0x0d)

● Command Description

To request the status of whether collect heatmap is enabled.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x0d	↓

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 0D 00 00 00 00

● Request Example

12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 0D 00 00 00 00

● Response Frame Format (Req id matches request sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x0d	↑

Data Content Structure:

Field	Type	Description	Remarks
Enable collect Heatmap	Unit32	Control flag	0: Disabled 1: Enabled

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 0D [Enable collect Heatmap]
```

● Response Example (Disable)

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 0D 00 00 00 00
```

3.13. Enable Intrusion detections (0x00 0x0e)

● Command Description

Controls intrusion detection function: Enables/disables intrusion detections.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x0e	↓

Data Content Structure:

Field	Type	Description	Remarks
Enable Intrusion detections	Unit32	Intrusion detection control flag	0x00 00 00 00: Disabled 0x00 00 00 01: Enabled

Complete structure of the request frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 0E [Enable Intrusion detections]
```

● Request Example (Enable)

```
12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 0E 00 00 00 01
```

● Response Frame Format (Req id matches request sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x0e	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 0E [Status code]

- **Response Example (Success)**

12 01 00 02 00 00 00 01 00 00 00 00 06 00 0E 00 00 00 01

3.14. Get Intrusion detections (0x00 0x0f)

- **Command Description**

To request the status of whether Intrusion detection is enabled.

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x0f	↓

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 0F 00 00 00 00

- **Request Example**

12 01 01 01 00 00 00 01 00 00 00 00 06 00 0F 00 00 00 00

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x0f	↑

Data Content Structure:

Field	Type	Description	Remarks
Enable Intrusion detections	Unit32	Intrusion detection control flag	0x00 00 00 00: Disabled 0x00 00 00 01: Enabled

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 0F [InstallationHeight]

- **Response Example (Enable)**

12 01 00 02 00 00 00 01 00 00 00 00 06 00 0F 00 00 00 01

3.15. Learning room layout position (0x00 0x10)

- **Command Description**

Controls room layout learning: Starts/stops radar from sending room layout coordinate data (X,Y).

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x0E	0x00 0x10	↓

Data Content Structure:

Field	Type	Description	Remarks
Room layout	Unit32	Start/Stop sending room layout (X,Y)	0x00 0x00 0x00 0x01: Start 0x00 0x00 0x00 0x03: Stop If the server does not send a stop command, Fall Detection Sensor will automatically stop after sending 1000 times
Reserve	Unit32	8-byte reserved field	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

Complete structure of the request frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 0E 00 10 [Room layout] 00 00 00 00 00 00 00 00
```

- **Request Example (Start transmission)**

```
12 01 01 01 00 00 00 01 00 00 00 00 00 0E 00 10 00 00 00 01 00 00 00 00 00 00 00 00
```

- **Data Report Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x0E	0x00 0x10	↑

Data Content Structure:

Field	Type	Description	Remarks
Sending data tag	Unit32	Data transmission tag, once per second	0x00 0x00 0x00 0x02
X	Float	Target position X-coordinate	Coordinate system: Origin at radar installation position X: horizontal axis Y: vertical axis
Y	Float	Target position Y-coordinate	

Complete structure of the report frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 0E 00 10 [Sending data tag] [X] [Y]
```

- **Response Example (X=1.5m, Y=2.0m)**

```
12 01 00 02 00 00 00 01 00 00 00 00 00 0E 00 10 00 00 00 02 3F C0 00 00 40 00 00 00
```

3.16. Negative fall detection (0x00 0x11)

- **Command Description**

Radar actively reports negative fall alerts: Reports negative fall events with fall position (X/Y axis floating-point numbers).

- **Alert Frame Format**

Type	Cmd	Content Length	Function Code	Direction
------	-----	----------------	---------------	-----------

0x01	0x01	0x0A	0x00 0x11	↑
------	------	------	-----------	---

Data Content Structure:

Field	Type	Description	Remarks
X	Float	Fall position X-coordinate	Coordinate system: Origin at radar installation position X: horizontal axis Y: vertical axis
Y	Float	Fall position Y-coordinate	

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 0A 00 11 [NegFall position]

● Alert Example (X=3.0m, Y=5.0m)

12 01 01 01 00 00 00 01 00 00 00 00 00 0A 00 11 40 40 00 00 40 A0 00 00

● Response Frame Format (Req id matches alert sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x11	↓

Data Content Structure:

字段名	类型	说明	备注
Status code	Unit32	Status code	0x00 0x00 0x00 0x01:Acknowledgement

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 11 [Status code]

● Response Example

12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 11 00 00 00 01

3.17. Register (0x00 0x12)

● Command Description

When the WiFi or the Tcp Socket is resumed from disconnection, the radar will send a registration message to the server automatically to re-establish the mapping relationship.

If no response is received, the radar should retry at intervals (e.g., every 10 seconds, up to 6 times).

● Registration Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x10	0x00 0x12	↑

Data Content Structure:

Field	Type	Description	Remarks
radar type	u8	Radar type	0x00: Home Care Assistant 0x01: Sleep Sensor
radar id	int13	Radar unique identifier ID	

Complete structure of the register frame

12 01 01 01 [Req id] [Timeout] 00 00 00 10 00 12 [radar type] + [radar id]

- **Register Example (Home Care Assistant , ID="12CC163F000300100800000000")**

12 01 01 01 00 00 00 01 00 00 00 00 10 00 12 00 12 CC 16 3F 00 03 00 10 08 00 00 00 00

- **Response Frame Format (Req id matches registration frame sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x12	↓

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code	0x00 0x00 0x00 0x01:Registration success

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 12 [Status code]

- **Response Example**

12 01 00 02 00 00 00 01 00 00 00 00 06 00 12 00 00 00 01

3.18. Get Firmware version (0x00 0x13)

- **Command Description**

Gets firmware version: Command to request radar firmware version

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x13	↓

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 13 00 00 00 00

- **Request Example**

12 01 01 01 00 00 00 01 00 00 00 00 06 00 13 00 00 00 00

- **Response Frame Format**

Type	Cmd	Content Length	Function Code	Direction
------	-----	----------------	---------------	-----------

0x00	0x02	0x06	0x00 0x13	↑
------	------	------	-----------	---

Data Content Structure:

Field	Type	Description	Remarks
firmware version	Unit32	Firmware version	

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 13 [firmware version]

● Response Example (1.8.2.0)

12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 13 00 01 08 02

3.19. The radar is reset (0x00 0x14)

● Command Description

Once the user resets the radar, the radar will send a reset command to the server(Only work in firmware version <= 1.8.X.X.). One-way notification, no response required.

● Notification Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x14	↓

Complete structure of the notification frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 14 0D 0D 0D 0D

● Notification Example

12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 14 0D 0D 0D 0D

.Note: When the server fails to receive this reset message from the radar, the firmware version should be checked.

3.20. Enable Machine Learning (0x00 0x15)

● Command Description

To turn on/off machine learning. The machine learning algorithm is to further reduce the false alerts of fall detection for better user experiences. However, it may result in lower accuracy in certain scenarios.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x15	↓

Data Content Structure:

Field	Type	Description	Remarks
machine learning	Unit32	Machine learning control flag	0x00 0x00 0x00 0x00: Turn off 0x00 0x00 0x00 0x01: Turn on

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 15 [machine learning]

● Request Example (Turn on)

12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 15 00 00 00 01

● Response Frame Format (Req id matches request sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x15	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 15 [Status code]

● Response Example (Success)

12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 15 00 00 00 01

3.21. Get Machine Learning (0x00 0x16)

● Command Description

To request the status of whether machine learning is enabled.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x16	↓

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 16 00 00 00 00

● Request Example

12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 16 00 00 00 00

● Response Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x16	↑

Data Content Structure:

Field	Type	Description	Remarks
machine learning	Unit32	Machine learning control flag	0x00 0x00 0x00 0x00: Turn off 0x00 0x00 0x00 0x01: Turn on

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 16 [machine learning]

● Response Example (Turn off)

12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 16 00 00 00 01

3.22. Fall Alert Elimination (0x00 0x17)

● Command Description

Fall alert elimination: When the radar detects the person has stood up after sending a fall alert, it actively notifies the alert cancellation.

● Notification Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x17	↑

Complete structure of the notification frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 17 00 00 00 00

● Frame Example

12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 17 00 00 00 00

● Response Frame Format (Req id matches notification frame sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x17	↓

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code	0x00 0x00 0x00 0x01:Acknowledgement

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 17 [Status code]

● Response Example

12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 17 00 00 00 01

3.23. Presence Detection (0x00 0x18)

● Command Description

Presence detection data report: Upload presence detection range bin and energy.

● Data Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0xD6	0x00 0x18	↑

Data Content Structure:

Field	Type	Description	Remarks
presence	Unit32	Presence status	0: empty. 1: occupied
presence_range	Float	Detection range (meters)	
presence_en	Float	Energy value	
raw data	uint8 * 200	Original ADC data	

Complete structure of the data frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 D6 00 18 [presence] [presence_range] [presence_en] [raw data]
```

● Request Example

Presence detected: **1**, detection range: **4.5m**, energy value **100** , raw data (may not be parsed)

```
12 01 01 01 00 00 00 04 27 10 00 00 00 D6 00 18 00 00 00 01 40 90 00 00 42 C8 00 00 00 00 00 04 01 02
02 02 01 06 03 03 04 03 01 01 03 04 04 07 0D 0E 09 08 05 02 08 01 03 03 04 07 05 04 01 05 06 05 01 00
01 02 07 0A 03 03 02 05 04 03 04 03 05 08 0A 04 03 0D 11 03 16 1A 25 30 29 31 18 0D 0D 08 13 05 14
16 22 2C 29 3D 4E 48 3B 24 06 07 05 08 19 24 13 05 08 0B
```

● Response Frame Format (Req id matches report frame sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x18	↓

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code	0x00 0x00 0x00 0x01:Acknowledgement

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 18 [Status code]
```

● Response Example

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 18 00 00 00 01
```

3.24. Start room interference study (0x00 0x19)

● Command Description

Start room interference studying: The radar will actively learn and filter interference sources within the environment, to help to improve the accuracy and user experience. Note: Make sure all personnel exit the radar's detection range before initiating the learning process. This

prevents human activity from being misidentified as interference, which could compromise radar detection performance.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x19	↓

Complete structure of the request frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 19 00 00 00 00
```

● Request Example

```
12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 19 00 00 00 00
```

● Response Frame Format (Req id matches request sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x19	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Response to server to start study

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 19 [Status code]
```

● Response Example (To **start study**)

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 19 00 00 00 01
```

Note: Power failure during learning requires retriggering **0x19**.

3.25. Room interference study finish (0x00 0x1A)

● Command Description

Response to server to finished study: Actively reports studying completion to server. One-way notification, no response required.

● Notification Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x1A	↑

Complete structure of the notification frame

```
12 01 001 01 [Req id] [Timeout] 00 00 00 06 00 1A 00 00 00 00
```

- **Notification Frame Example**

```
12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 1A 00 00 00 00
```

3.26. Room interference clear (0x00 0x1B)

- **Command Description**

To clear room interference learning results (Only work in firmware version $\geq 2.8.1.9$).

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x1B	↓

Complete structure of the request frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 1B 00 00 00 00
```

- **Request Example**

```
12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 1B 00 00 00 00
```

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x1B	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 1B [Status code]
```

- **Response Example (Success)**

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 1B 00 00 00 01
```

Note: Check firmware version if clearing fails.

3.27. Position and motion detection (0x00 0x1C)

- **Command Description**

Basic position report: Upload user position X, Y, Z and motion detection data every 10 seconds.

Enhanced position report (firmware $\geq 2.8.1.44$): Upload target position X, Y, Z and motion detection

Multi-target report (firmware $\geq 2.8.1.45$): Upload num of moving targets.

One-way notification, no response required.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x02	0x01	0x0E 0x1E (firmware $\geq 2.8.1.44$) 0x22 (firmware $\geq 2.8.1.45$)	0x00 0x1C	↑

Data Content Structure:

Field	Type	Description	Remarks
position_x	Float	X-axis coordinate (meters)	
position_y	Float	Y-axis coordinate (meters)	
position_z	Float	Z-axis coordinate (meters)	
motion_x	Float	Moving target X coordinate	firmware $\geq 2.8.1.44$
motion_y	Float	Moving target Y coordinate	
motion_z	Float	Moving target Z coordinate	
motion_snr	Float	Signal-to-noise ratio (dB)	
target_count	UInt32	Number of moving targets	firmware $\geq 2.8.1.45$

Complete structure of the request frame

```
12 01 02 01 [Req id] [Timeout] [Content Len] 00 1C [position_x] [position_z] [motion_x] [motion_y]
[motion_z] [motion_snr] [target_count]
```

● Request Example (Basic position report X=1.5, Y=2.0, Z=0.8)

```
12 01 02 01 00 00 00 01 00 00 00 00 00 0E 00 1C 3F C0 00 00 40 00 00 00 3F 4C CC CD
```

● Request Example (Enhanced position report X=1.5, Y=2.0, Z=0.8, Moving target X=1.5, Moving target Y=2.0, Moving target Z=1.8, SNR = 60)

```
12 01 02 01 00 00 00 01 00 00 00 00 00 1E 00 1C 3F C0 00 00 40 00 00 00 3F 4C CC CD 3F C0 00 00
40 00 00 00 3F 4C CC CD 42 70 00 00
```

● Request Example (Enhanced position report X=1.5, Y=2.0, Z=0.8, Moving target X=1.5, Moving target Y=2.0, Moving target Z=1.8, SNR = 60, Multi-target report, 3 targets)

```
12 01 02 01 00 00 00 01 00 00 00 00 00 22 00 1C 3F C0 00 00 40 00 00 00 3F 4C CC CD 3F C0 00 00
40 00 00 00 3F 4C CC CD 42 70 00 00 00 00 00 03
```

3.28. Fall detection Triggered (0x00 0x1D)

● Command Description

Sent immediately after a fall event occurs.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x02	0x01	0x0A	0x00 0x1D	↓

Data Content Structure:

Field	Type	Description	Remarks
X	Float	Fall position X-coordinate (m)	Coordinate system: Origin (0,0) at radar position X: horizontal axis Y: vertical axis
Y	Float	Fall position Y-coordinate (m)	

Complete structure of the request frame

```
12 01 02 01 [Req id] [Timeout] 00 00 00 06 00 1D [X] [Y]
```

● Request Example (Trigger alarm coordinates X=1m , Y=3m)

```
12 01 02 01 00 00 00 01 00 00 00 00 00 0A 00 1D 3F 80 00 00 40 40 00 00
```

● Response Frame Format (Req id matches request sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x1D	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 1D [Status code]
```

● Response Example (Success)

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 1D 00 00 00 01
```

3.29. Set Fall detection mode (0x00 0x1E)

● Command Description

Command to set the fall detection mode: High sensitivity mode, Low false alert mode.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x1E	↓

Data Content Structure:

Field	Type	Description	Remarks
-------	------	-------------	---------

fall detection mode	Unit32	Fall detection mode	0: High sensitivity mode 1: Low false alert mode
---------------------	--------	---------------------	---

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 1E [fall detection mode]

- Request Example (Set to **low false alert mode**)

12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 1E 00 00 00 01

- Response Frame Format (Req id matches request sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x1E	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 1E [Status code]

- Response Example (**Success**)

12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 1E 00 00 00 01

3.30. Get Fall detection mode (0x00 0x1F)

- Command Description

Command to get the fall detection mode.

- Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x1F	↓

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 1F 00 00 00 00

- Request Example

12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 1F 00 00 00 00

- Response Frame Format (Req id matches request sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x1F	↑

Data Content Structure:

Field	Type	Description	Remarks
fall detection mode	Unit32	Fall detection mode	0: High sensitivity mode 1: Low false alert mode

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 1F [fall detection mode]

● Response Example (Low false alert mode)

12 01 00 02 00 00 00 02 00 00 00 00 00 06 00 1F 00 00 00 01

3.31. Set Room Configuration (0x00 0x24)

● Command Description

Command to set room configuration: Configure installation type/area coordinates/detection boundaries.

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0xCA	0x00 0x24	↓

Data Content Structure:

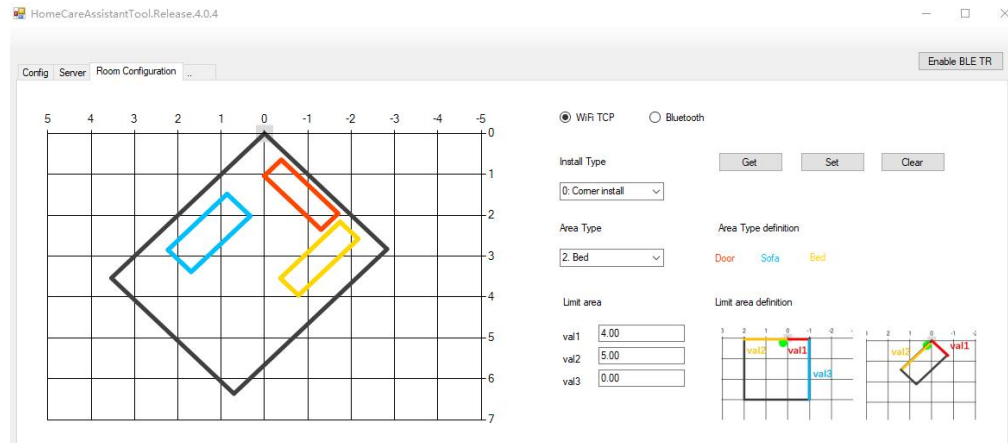
Field	Type	Description	Remarks
install type	U32	Installation type	0: corner, 1: wall
num of area	U32	Number of configured areas	0-5
Area_1 Type	U32	Area 1 type	0 :Door, 1:Sofa, 2:Bed
Area_1_XY[8]	float	Area 1 coordinates	Quadrilateral vertex coordinates(to be set in clockwise or counterclockwise order) x1,y1,x2,y2,x3,y3,x4,y4
Area_2 Type	U32	Area 2 type	
Area_2_XY[8]	float	Area 2 coordinates	
Area_3 Type	U32	Area 3 type	
Area_3_XY[8]	float	Area 3 coordinates	
Area_4 Type	U32	Area 4 type	
Area_4_XY[8]	float	Area 4 coordinates	
Area5 Type	U32	Area 4 type	
Area_5_XY[8]	float	Area 4 coordinates	
Limit[3]	float	Detection boundary	boundary default value: : 0 indicates no boundary is set. corner install val1=0 or val2=0 wall install (val1=0 and val2 =0) or

(val3 = 0)

Complete structure of the request frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 CA 00 24 [install type] [num of area] [Area 1 Type]
[Area 1_XY[8]] [Area 2 Type] [Area 2_XY[8]] [Area 3 Type] [Area 3_XY[8]] [Area 4 Type]
[Area 4_XY[8]] [Area 5 Type] [Area 5_XY[8]] [Limit[3]]
```

● Request Example (Conner-mounted + 3 areas of different types)



```
12 01 01 01 9B 00 00 00 10 27 00 00 00 CA 00 24 00 00 00 00 00 00 03 00 00 00 00 3C 14 1F B9 3F
84 F6 BC BE C5 39 16 3F 25 00 6E BF D9 58 0A 3F FA 89 FB BF A6 E1 85 40 16 80 40 00 00 00 01 3F
5D 82 1A 3F BD AC 7E 40 0E C9 3F 40 36 3E F7 3F D8 D4 51 40 58 9E 0E 3E A8 0B 7E 40 01 35 55 00
00 00 02 C0 0A 54 3E 40 25 0C 9C BF 48 13 54 40 7D 5C 05 BE BA 48 C6 40 62 A0 48 BF DF 31 03 40
0A 50 E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 40 80 00 00 40 A0 00 00 00 00 00 00
```

C code struct

```
#define MAX_NUM 5
typedef enum CareSensor_Configuration_Area_Type_e
{
    CareSensor_Configuration_Area_Type_Door = 0,
    CareSensor_Configuration_Area_Type_Sofa = 1,
    CareSensor_Configuration_Area_Type_Bed = 2,
} CareSensor_Configuration_Area_Type;
typedef struct CareSensor_Configuration_Type_t
{
    CareSensor_Configuration_Area_Type type;
    float location[8];
} CareSensor_Configuration_Type;
typedef struct CareSensor_Room_Configuration_t
{
    uint32_t installType;
    uint32_t num;
    CareSensor_Configuration_Type config[MAX_NUM];
    float limit[3];
} CareSensor_Room_Configuration;
```

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x24	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 24 [Status code]
```

- **Response Example (Success)**

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 24 00 00 00 01
```

3.32. Get Room Configuration (0x00 0x25)

- **Command Description**

Command to get room configuration.

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x25	↓

Complete structure of the request frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 25 00 00 00 00
```

- **Request Example**

```
12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 25 00 00 00 00
```

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0xCA	0x00 0x25	↑

Data Content Structure:

Field	Type	Description	Remarks
install type	U32	Installation type	0: corner, 1: wall
num of area	U32	Number of configured areas	0-5
Area_1 Type	U32	Area 1 type	0 :Door, 1:Sofa, 2:Bed
Area_1_XY[8]	float	Area 1 coordinates	Quadrilateral vertex coordinates x1,y1,x2,y2,x3,y3,x4,y4
Area_2 Type	U32	Area 2 type	

Area_2_XY[8]	float	Area 2 coordinates	
Area_3 Type	U32	Area 3 type	
Area_3_XY[8]	float	Area 3 coordinates	
Area_4 Type	U32	Area 4 type	
Area_4_XY[8]	float	Area 4 coordinates	
Area5 Type	U32	Area 4 type	
Area_5_XY[8]	float	Area 4 coordinates	
Limit[3]	float	Detection boundary	boundary default value corner install val1=0 or val2=0 wall install (val1=0 and val2 =0) or (val3 = 0)

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 CA 00 25 [install type] [num of area] [Area_1 Type]
[Area_1_XY[8]] [Area_2 Type] [Area_2_XY[8]] [Area_3 Type] [Area_3_XY[8]] [Area_4 Type]
[Area_4_XY[8]] [Area_5 Type] [Area_5_XY[8]] [Limit[3]]
```

● Response Example (Wall-mounted , 3 areas of different types)

```
12 01 00 02 00 00 00 02 00 00 00 00 00 CA 00 25 00 00 00 01 00 00 00 03 00 00 00 00 3C 14 1F B9 3F
84 F6 BC BE C5 39 16 3F 25 00 6E BF D9 58 0A 3F FA 89 FB BF A6 E1 85 40 16 80 40 00 00 00 01 3F
5D 82 1A 3F BD AC 7E 40 0E C9 3F 40 36 3E F7 3F D8 D4 51 40 58 9E 0E 3E A8 0B 7E 40 01 35 55 00
00 00 02 C0 0A 54 3E 40 25 0C 9C BF 48 13 54 40 7D 5C 05 BE BA 48 C6 40 62 A0 48 BF DF 31 03 40
0A 50 E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 40 80 00 00 40 A0 00 00 00 00 00 00
```

3.33. Head counting (0x00 0x27)

● Command Description

The radar actively reports the total number of moving targets every 10 seconds. Each report contains 100 frames of data, and the frame is refreshed every 100ms.

● Alarm Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x0A	0x00 0x27	↓

Detailed Data Field Structure:

Field	Type	Description	Remarks
head counting number	UInt8 * 100	head counting data: 100 frames of alarm data within 10 seconds	Frame interval unit: 100ms

Complete Alarm Frame Structure:

```
12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 27 [head counting number]
```

- **Alarm Frame Example(100ms: 1 target detected, 200ms: 0,1,0.....) :**

```
12 01 02 01 00 00 00 0E 27 10 00 00 00 66 00 27 01 00 01 00 01 01 01 00 00 00 00 01 01 01 01 01 00 01
01 01 01 01 01 01 00 01 01 01 01 01 01 00 00 01 00 01 01 01 02 02 01 01 01 01 01 01 01 01 02 01 00
01 01 01 01 00 01 01 01 01 01 02 01 01 01 01 01 02 01 01 01 01 01 01 01 02 01 01 01 01 01 02 01 01
01 01 01 01 01 01 01 01 01 01 01 01 01 01
```

- **Response Frame Format (Req id matches the sequence number of the alarm frame)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x27	↑

Detailed Data Field Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code	0x00 0x00 0x00 0x01: Acknowledgement

Complete Response Frame Structure:

```
12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 09 [Status code]
```

- **Response Example**

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 27 00 00 00 01
```

3.34. Firmware Upgrade

The upgrade command consists of three steps:

3.34.1. Send upgrade parameters (0x00 0x21)

- **Command Description**

Upgrade initialization: Notify the upgrade and sending the upgrade firmware length.

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x21	↓

Data Content Structure:

Field	Type	Description	Remarks
Firmware size	U32	Firmware size (bytes)	Example: 524,292 bytes

Complete structure of the request frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 21 [Firmware size]
```

- **Request Example (524292)**

```
12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 21 00 08 00 04
```

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
------	-----	----------------	---------------	-----------

0x00	0x02	0x06	0x00 0x21	↑
------	------	------	-----------	---

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 21 [Status code]

● Response Example (Success)

12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 21 00 00 00 01

3.34.2. Send firmware content in segments (0x00 0x22)

● Command Description

Send firmware content in fragments: Every frame checks CRC16 (crc only checks firmware content, without header, length, flag bits). Firmware content up to 240 bytes

● Request Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	Variable (N)	0x00 0x22	↓

Data Content Structure:

Field	Type	Description	Remarks
Firmware content	UInt8	Firmware content ($N \leq 240$)	240 bytes or remaining bytes in the last frame
crc check	uint16	CRC16 checksum	Each frame is CRC16-verified (checksum only for firmware content, excluding headers/length/flags)

Complete structure of the request frame

12 01 01 01 [Req id] [Timeout] [Content Len] 00 22 [Firmware size]

● Request Example (240bytes)

12 01 01 01 9B 00 00 00 10 27 00 00 03 EC 00 22 01 20 83 90 01 46 B3 F1 58 00 07 D0 17 38 09 D0 40 1E 03 D0 08 38 18 BF 0A 22 04 D1 10 22 02 E0 FF FF 0F 00 08 22 70 2B 81 98 00 F0 B3 80 43 F6 60 76 B8 46 30 40 20 38 B9 46 00 F0 8D 80 20 38 75 D0 C0 38 5C D0 B0 F5 80 70 42 D0 B0 F5 C0 60 31 D0 B0 F5 00 60 20 D0 B0 F5 80 50 0F D0 B3 F1 58 00 72 D0 0C 38 00 F0 8C 80 40 1F 00 F0 89 80 80 1F 6A D0 80 1F 68 D0 C0 1E 66 D0 94 E0 B3 F1 58 00 62 D0 0C 38 7C D0 40 1F 7A D0 80 1F 5C D0 80 1F 5A D0 C0 1E 58 D0 86 E0 B3 F1 58 00 54 D0 0C 38 6E D0 40 1F 6C D0 80 1F 4E D0 80 1F 4C D0 C0 1E 4A D0 78 E0 B3 F1 58 00 2F D0 0C 38 2D D0 40 1F 2B D0 80 1F 29 D0 80 1F 27 D0 C0 1E 25 D0 6A E0 B3 F1 58 00 0A D0 0C 38 52 D0 40 1F 50 D0 80 1F 04 D0 80 1F 18 BF B0 F1 03 00 5C D1 E4 1C 24 F0 03 04 24 1D 14 F8 04 8C 4F F0 00 09 52 E0 B3 F1 58 00 0A D0 0C 38 08 D0 40 1F 06 D0 80 1F 04 D0 80 1F 18 BF B0 F1 03 00 45 D1 E4 1D 24 F0 07 04 08 34 A4 F1 08 00 90 E8 00 03 3B E0 B3 F1 58 00 0A D0 0C 38 24 D0 40 1F 22 D0 80 1F 04 D0 80 1F 18 BF B0 F1 03 00 2E D1 E4 1C 24 F0 03 04 24 1D 54 F8 04 8C D0 E7 B3 F1 58 00 0A D0 0C 38 0F D0 40 1F 0D D0 80 1F 04 D0 80 1F 18 BF B0

```
F1 03 00 19 D1 E4 1C 24 F0 03 04 24 1D 34 F8 04 8C BB E7 E4 1C 24 F0 03 04 24 1D 54 F8 04 5C 4F
EA E5 79 07 E0 E4 1C 24 F0 03 04 24 1D 4F F0 00 09 54 F8 04 5C A8 46 90 94 C8 17 4C 46 45 46 54
EA 00 06 45 EA 01 00 08 BF 00 28 81 98 02 D1 01 09 C0 F0 43 81 64 2B 18 BF 69 2B 08 D1 00 2C 06
DA 6D 42 4F F0 00 00 60 EB 04 04 01 26 8E 96 85 A8 00 90 21 46 28 46 1B F0 13 FC 83 99 40 1C 9D
F8 10 32 81 42 85 99 29 DA 78 2B 03 D0 58 2B 18 BF 70 2B 11 D1 81 98 01 09 0F D3 70 2B 08 BF 78
23 85 98 30 21 03 70 85 98 40 1E 85 90 01 70 85 98 9D F8 10 32 40 1E 85 90 81 98 6F 2B 40 F0 0D 81
01 09 85 99 C0 F0 09 81 4A 78 30 2A 00 F0 05 81 30 20 08 70 85 98 40 1E 85 90 BF E0 30 22 0A 70 85
99 49 1E 85 91 C8 E7 E4 1D 24 F0 07 04 08 34 90 94 18 46 A0 F1 61 01 1A 29 14 ED 02 0B 38 BF 20
38 98 42 39 46 08 BF 01 21 59 EC 10 8B 47 F6 F0 74 47 F6 F0 70 04 EA 19 44 A0 42 4D 46 14 D1 91
9E 28 46 00 00 4F F0 00 0E 4E
```

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	Variable (N)	0x00 0x22	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

```
12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 22 [Status code]
```

- **Response Example (Success)**

```
12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 22 00 00 00 01
```

3.34.3. Send upgrade completed (0x00 0x23)

- **Command Description**

Notify the radar that the upgrade is complete (if interrupted, the firmware can be resent to restart the upgrade). Triggers the radar to verify and apply the new firmware.

- **Request Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x00 0x23	↓

Complete structure of the request frame

```
12 01 01 01 [Req id] [Timeout] 00 00 00 06 00 23 00 00 00 00
```

- **Request Example**

```
12 01 01 01 00 00 00 01 00 00 00 00 00 06 00 23 00 00 00 00
```

- **Response Frame Format (Req id matches request sequence number)**

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x00 0x23	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 00 23 [Status code]

● Response Example (Success)

12 01 00 02 00 00 00 01 00 00 00 00 00 06 00 23 00 00 00 01

3.35. Soft reset the sensor (0x04 0x11)

● Command Description

Soft reset the Sensor: Resets the system without hardware reboot. After reset, wait 5 seconds to reconnect. Non-persistent configurations will be cleared.

● Notification Frame Format

Type	Cmd	Content Length	Function Code	Direction
0x01	0x01	0x06	0x04 0x11	↓

Complete structure of the notification frame

12 01 01 01 [Req id] [Timeout] 00 00 00 06 04 11 00 00 00 00

● Notification Frame Example

12 01 01 01 00 00 00 01 00 00 00 00 00 06 04 11 00 00 00 00

● Response Frame Format (Req id matches the notification sequence number)

Type	Cmd	Content Length	Function Code	Direction
0x00	0x02	0x06	0x04 0x11	↑

Data Content Structure:

Field	Type	Description	Remarks
Status code	Unit32	Status code (success/failure)	0x00 0x00 0x00 0x00: Failed 0x00 0x00 0x00 0x01: Succeed

Complete structure of the response frame

12 01 00 02 [Req id] [Timeout] 00 00 00 06 04 11 [Status code]

● Response Example (Success)

12 01 00 02 00 00 00 01 00 00 00 00 00 06 04 11 00 00 00 01

3.36. WiFi Signal dBm (0xFF 0xFF)

● Command Description

Upload WiFi signal strength (dBm). Unidirectional notification; no response required.

- **Report Frame Format**

Type	Cmd	Content Length	Function Code	Direction
0x02	0x01	0x06	0xFF 0xFF	↑

Complete structure of the report frame

```
12 01 02 01 [Req id] [Timeout] 00 00 00 06 FF FF 00 00 00 00
```

- **Report Frame Example**

```
12 01 02 01 9B 00 00 00 10 27 00 00 00 06 FF FF FF FF FF C6
```