J.MARPLE

Communication Protocol

Abstract

This document defines the SLAMBOX Communication Protocol, designed to streamline the interaction between the SLAMBOX device and various host systems, such as PCs, robots, and drones. The protocol outlines the structured methods for transmitting SLAM results, ensuring efficient and reliable communication. It details the specific message formats and configurations necessary for different host systems to effectively receive and utilize data from SLAMBOX.

Table of Contents

1. Overall Structure

1.1. Common Protocols

- The byte order of integers is Big Endian.
- Float is represented in 4 bytes according to IEEE 754 standard.
    - 1 bit for the sign
    - 8 bits for the exponent
    - 23 bits for the fraction
- String's length is indicated in the first byte.
    - Therefore, the maximum length of a String is 255.
- This document is written from the perspective of the sender (server).

1.2. Overall Message structure

2 bytes	2 bytes	2 bytes	Variable	2 byte
Header (0xFA 0x5A)	Mode	Length	Payload	Checksum

- Header: Indicates the beginning of the message.
- Mode: Defines the type of message. The data in the Payload changes accordingly, and details are provided below.
- Length: Defines the length of the Payload. Therefore, the total length of the message is (8+Length) bytes.
- Checksum: The message (Mode, Length, Payload), excluding the Header, is calculated according to the TCP standard(The 8-bit Fletcher Checksum).

The 8-bit Fletcher Checksum

```
CK_A = 0, CK_B = 0
For(I = 0; I < N; I++)
{
    CK_A = CK_A + Buffer[I]
    CK_B = CK_B + CK_A
}
```

1.3. Summary of Message Mode

Mode	Description	Payload Length
0x01 0x01	Push Odometry	348 bytes
0x01 0x02	Push Pointcloud	? bytes
0x10 0x01	Request data status	0 byte
0x11 0x01	Response data status	3 bytes
0x20 0x01	Set Enable/Disable data status	3 bytes
0x30 0x01	Command Restart LIO	0 byte
0x30 0x02	Save PCD	2 bytes
0xA0 0x00	Acknowledge	3 bytes

2. Mode Detailed Messages

2.1. Push

- Continuous transmission at specific intervals

2.1.1. Push Odometry (0x01 0x01)

- Contains Odometry values.
- Payload length: 348 bytes

Data Category	Data Type	Length
Timestamp (second)	uint32	4 bytes
Timestamp (nano second)	uint32	4 bytes
Position x	float	4 bytes
Position y	float	4 bytes
Position z	float	4 bytes
Orientation x	float	4 bytes
Orientation y	float	4 bytes
Orientation z	float	4 bytes

Orientation w	float	4 bytes
Pose covariance	36 float array	144 bytes
Twist linear x	float	4 bytes
Twist linear y	float	4 bytes
Twist linear z	float	4 bytes
Twist angular x	float	4 bytes
Twist angular y	float	4 bytes
Twist angular z	float	4 bytes
Twist covariance	36 float array	144 bytes

2.1.2. Push cloud points (0x01 0x02)

- Contains Cloud point data.
- Payload Length: Variable
- If the Compression flag is 0 -> Uncompressed payload
- If the Compression flag is 1 -> Compressed payload
    - The compression algorithm is performed by zlib

Payload		
Data Category	**Data Type**	**Length**
Compression flag	bool	1 byte
Payload	Compressed or Uncompressed payload	? bytes

Compressed Payload		
Data Category	**Data Type**	**Length**
Uncompressed payload length	uint32	4 bytes
Compressed payload	Compressed byte array	? bytes

Uncompressed payload		
Data Category	**Data Type**	**Length**
Timestamp (second)	uint32	4 bytes
Timestamp (nano second)	uint32	4 bytes
frame_id	String	? bytes
Height	uint32	4 bytes
Width	uint32	4 bytes
Number of [Field]	uint8	1 byte
[Field]	Field array	? bytes
is_bigendian	uint8	1 byte
point_step	uint32	4 bytes
row_step	uint32	4 bytes
Number of [data]	uint32	4 bytes
[data]	uint8 array	? bytes
is_dense	uint8	1 byte

Field		
Data Category	**Data Type**	**Length**
[Field] name	String	? bytes
[Field] offset	uint32	4 bytes
[Field] datatype	uint8	1 byte
[Field] count	uint32	4 bytes

2.2. Request config setting

- Receives requests for specific configuration values
- After receiving a Request, send a Response as defined in section 2.3.

2.2.1. Request Mavlink communication config (0x10 0x01)

- Receives requests for Mavlink Communication configuration values.
- Payload length: 0 byte

2.2.2. Request Serial communication config (0x10 0x02)

- Receives requests for Serial communication configuration values.
- Payload length: 0 byte

2.2.3. Request Ethernet communication config (0x10 0x03)

- Receives Ethernet communication configuration values.
- Payload length: 0 byte

~~2.2.4. Request LiDAR config (0x10 0x10)~~

- ~~Receives LiDAR configuration values.~~
- ~~Payload length: 0 byte~~

2.3. Response config setting

- Send the requested configuration values.

2.3.1. Response Mavlink communication config (0x11 0x01)

- Send Mavlink communication configuration values.
- Payload length: 5 byte

Data Category	Data Type	Length
Enable/Disable	Bool	1 byte
Baudrate	uint32	4 bytes

2.3.2. Response Serial communication config (0x11 0x02)

- Send Serial communication configuration values
- Payload length: 5 byte

Data Category	Data Type	Length
Enable/Disable	Bool	1 byte
Baudrate	uint32	4 bytes

2.3.3. Response Ethernet communication config (0x11 0x03)

- Send Serial communication configuration values
- Payload length: 5 byte

Data Category	Data Type	Length
Enable/Disable	Bool	1 byte
Port number	uint32	4 bytes

~~2.3.4. Response LiDAR config (0x11 0x10)~~

- ~~Send LiDAR configuration values~~
- ~~Payload length: 1 byte~~

~~Data Category~~	~~Data Type~~	~~Length~~
~~LiDAR Type~~	~~uint8~~	~~1 byte~~

- ~~LiDAR Type~~
  - ~~0x01: Livox MID-360~~
  - ~~0x02: Ouster OS0-32~~
  - ~~0x03: Ouster OS0-64~~
  - ~~0x04: Ouster OS0-128~~
  - ~~0x05: Livox AVIA~~

2.4. Set config setting

- Receives requests to change specific configuration values.
- After receiving this protocol, send an Acknowledgement message as defined in section 2.6.

2.4.1. Set Mavlink communication config (0x20 0x01)

- Receives requests to change Mavlink communication configuration values.
- Payload length: 5 bytes
- Payload specification is the same as in 2.3.1.

2.4.2. Set Serial communication config (0x20 0x02)

- Receives requests to change Serial communication configuration values.
- Payload length: 5 bytes
- Payload specification is the same as in 2.3.2.

2.4.3. Set Ethernet communication config (0x20 0x03)

- Receives requests to change Ethernet communication configuration values.
- Payload length: 5 bytes
- Payload specification is the same as in 2.3.3.

~~2.4.4. Set LiDAR config (0x20 0x10)~~

- ~~Receives requests to change LiDAR configuration values.~~
- ~~Payload length: 1 byte~~
- ~~Payload specification is the same as in 2.3.4.~~

2.5. Command

- Receives other commands
- After receiving a command, send an Acknowledge as defined in section 2.6.

~~2.5.1. Restart LIO (0x30 0x01)~~

- ~~Command to restart LIO~~
- ~~Payload length: 0 byte~~

2.5.2. Save PCD (0x30 0x02)

- Command to save PCD file
- Payload length: 2 bytes
- reset flag
  - 0x00: Do not reset Pointcloud buffer
  - 0x01: Reset Pointcloud buffer
- save flag
  - 0x00 : Do not perform PCD file save
  - 0x01 : Perform PCD file save

Data Category	Data Type	Length
Save flag	Bool	1 byte
Reset flag	Bool	1 byte

2.6. Acknowledge

2.6.1. Acknowledge (0xA0 0x00)

- Send status of reception confirmation for messages.
- Payload length: 3 bytes
- (Status) 0x00: Acknowledge OK
- (Status) 0x01: Acknowledge Failed
- (Status) 0x02: Acknowledge Mode Unknown

Data Category	Data Type	Length
Requested mode	Mode	2 byte
Status	Status	1 byte

2.6.2. Ping (0xA0 0x01)

- Request to check if the connection is normal. Upon receiving this message, respond with an Acknowledge as defined in 2.6.1.
- Payload length: 0 byte