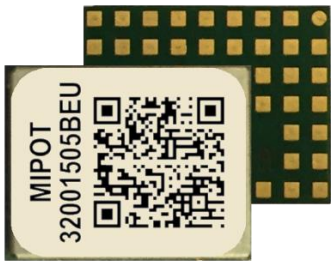
Wireless Protocol Modules MiP Series 3200150xFxx

Command Reference



Description

This document provides list of commands that the 3200150xFxx implement and the description of their use.

Contents

1. Communication protocol

1.1. Byte Order

Multiple byte values are transmitted in little endian order with least significant byte first (LSB).

1.2. Message Structure

The structure of the messages is the following:

HEADER	CMD	LENGTH	PAYLOAD (n Bytes)	CHECKSUM

Where:

HEADER = 0xAA

CMD = Command code to the module (32001505Fxx) / to the radio stack (32001506Fxx) , see the following table

LENGTH = Payload length

CHECKSUM = 2's complement on one byte of the sum of all preceding bytes

Each command from the host invokes an answer from the module (32001505Fxx)/from the radio stack (32001506Fxx) in the same format.

The answer to the host has the CMD field equal to host request OR 0x80.

1.3. Message Types

There are three types of messages:

Commands: sent from the host to the module to request an information or an action (32001505Fxx case).

sent from the user application running on the M4 core to the radio stack running on the M0+ core to request an information or an action (32001506Fxx case).

Replies: sent from the module to the host as direct reply to a command, their command code is equal to the host request (<cmd> OR 0x80) (32001505Fxx case).

sent from the radio stack to the user application as direct reply to a command, their command code is equal to the host request (<cmd> OR 0x80) (32001506Fxx case).

Indications: messages sent from the module to the host that are sent without prior action from the host, triggered by events on the radio interface. (e.g.: a received transmission) (32001505Fxx case).

messages sent from the radio stack to the user application that are sent without prior action from the host, triggered by events on the radio interface. (E.g.: a received transmission) (32001506Fxx case).

2. Command Set Description

Current document describes only commands needed to move between stacks loaded into 3200150xFxx module.

To get information about stack specific commands please refer to their own reference guides as listed below:

Stack	Document
LoRaModem	3200150xDEU_Command_Reference
LoRaWan	3200150xBxx_Command_Reference

List of the implemented commands:

Command (CMD)	Value	Description
CHANGE_STACK_CMD	0x2A	Select the working stack without saving
GET_STACK_CMD	0x2B	Get currently used stack
CHG_STACK_SAVE_CMD	0x32	Select stack and save parameter in EEPROM
GET_SAVED_STACK_CMD	0x33	Reads saved stack value

2.1. CHANGE_STACK_CMD (0x2A)

This command performs a 'runtime' stack change without saving the new setting.

Host: 0xAA, 0x2A, 0x01, Stack, cks

Reply: 0xAA, 0xAA, 0x01, Status, cks

Stack: 0x00: LoRaModem, 0x01: LoRaWan

Status: 0x00: Success, 0xFF: Fail

2.2. GET_STACK_CMD (0x2B)

This command retrieves the code of the currently used stack.

Host: 0xAA, 0x2B, 0x00, 0x2B

Reply: 0xAA, 0xAB, Length, Status, Stack, cks

Stack: 0x00: LoRaModem, 0x01: LoRaWan

Status: 0x00: Success, 0xFF: Fail

If Status is Fail then Length is 0x01 and Stack is omitted.

2.3. CHG_STACK_SAVE _CMD (0x32)

This command performs a stack change and saves the new setting in EEPROM.

Host: 0xAA, 0x32, 0x02, 0x90, Stack, cks

Reply: 0xAA, 0xB2, 0x01, Status, cks

Stack: 0x00: LoRaModem, 0x01: LoRaWan

Status: 0x00: Success, 0xFF: Fail

2.4. GET_SAVED_STACK_CMD (0x33)

This commands reads from EEPROM the stored code of the stack to be used at startup. It may differ from the currently used stack.

Host: 0xAA, 0x33, 0x02, 0x90, 0x01, 0x90

Reply: 0xAA, 0xB3, 0x02, Status, Stack, cks

Stack: 0x00: LoRaModem, 0x01: LoRaWan

Status: 0x00: Success, 0xFF: Fail

3. Module Configuration

3.1. Module parameters

Parameter	Description	Address	Range	Default	Notes
Stack	Code of the stack to be used at startup	0x90	0-1	0	

4. Revision History

Revision	Date	Description
0.1	25.11.2024	First version