Revision History

|  |  |  |
| --- | --- | --- |
| Date | Ver | Remark |
| 2018.11.09 | V1.0.0 | First formal version for PXX2.0 file. |
| 2019.02.12 | V1.0.1 | Change the length of RXID and Password to 8 bytesAdd the RX\_NUM Byte when BIND |
| 2019.02.12 | V1.0.2 | Update REGISTER frames |
| 2019.02.20 | V1.0.3 | Change the length of PPM frame to 24 channelsAdd Tx channel mode (8/16/24)Add bind without Password mode (share mode)Add RX PWM output period  |
| 2019.02.22 | V1.0.4 | Add GET HW INFO frameRename RX\_TELEM\_ID to RX\_UID (0-15)Rename RX\_ID to RX\_NAME |
| 2019.02.25 | V1.0.5 | Update GET HW INFO frameSwap RX\_NUM and RX\_UID in BIND |
| 2019.03.14 | V1.0.6 | Add SHARE frame |
| 2019.03.15 | V1.0.7 | Add LOOP\_ID in REGISTER frame (temporary hack to avoid implement a random delay on BIND) |
| 2019.03.15 | V1.0.8 | Add RESET frame |
| 2019.04.23 | V1.0.9 | Add OTA frame |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

FrSky PXX2 Protocol

# Introduction

PXX2 is based on a physical UART, including TX and RX. It supports duplex communication and the baud rate between Radio and TX Modules can reach 450kbps. Based on the duplex communication port, Radio and TX Modules can communicate in real-time.

RADIO

TX MODULE

RX/TX

TX/RX

Figure-1

# Physical Interface

Below is hardware drawing (figure-2) for PXX2 physical interface.

​​

Figure-2

**Note**: Only PXX2\_TX/RX, GND, VCC5.0 were used in V1.0, other pins are reserved.

# Physical Layer

## Interface Settings

450kbps, 8N1, no parity.

## Frame structure

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1 | …… | DATA N | CRC16H | CRC16L |
| 0x7E |  |  |  |  |  |  |  |  |  |

Table-3.2

Frame Head: 0x7E.

LEN: Length of the frame (byte), Head/LEN/CRC bytes are not included.

TYPE\_C: Class of data, please refer to the Table-4.01

TYPE\_ID: Please refer to the Table-4.01

DATA0~DATA N: Data.

N: Max size of data packet, no more than 57 (N<=57).

CRC16: The sum of the bytes up to (but not including) CRC16H added to the 16-bit value CRC16H:CRC16L = 0xFFFF

# Frame definitions

## Summary

|  |  |  |
| --- | --- | --- |
| TYPE\_C | TYPE\_ID | REMARK |
| 0x00 | … | Discard frame |
| 0x01(TX Modules) | 0x01 | REGISTER |
| 0x02 | BIND |
| 0x03 | CHANNELS |
| 0x04 | TX SETTINGS |
| 0x05 | RX SETTINGS |
| 0x06 | HARDWARE INFORMATION |
| 0x07 | SHARE |
| 0x08 | RESET |
| … | reserved |
| 0xFE | S.PORT TELEMETRY |
| 0xFF | DATA (user defined format) |
| 0x02(TX Tools) | … | reserved |
| 0x01 | POWER METER |
| 0x02 | SPECTRUM ANALYSER |
| … | reserved |
| 0xFE (OTA Updates) | … | reserved |
| 0x01 | Update the TX |
| 0x02 | Update the RX |
| …. | reserved |
| 0xFF |  |  |
|  |  |  |

## Register receiver

1. Radio to TX in register mode

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA 0 | CRCH | CRCL |
| 0x7E | 0x03 | 0x01 | 0x01 | 0x00 | 0xFF | 0xFD |

2. TX returns the id received

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~8 | DATA17 | CRCH | CRCL |
| 0x7E | 0x0C | 0x01 | 0x01 | 0x00 | RX\_NAME(initial) | LOOP\_ID |  |  |

3. Radio to TX : the user can set here the RX\_NAME and Password

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~8 | DATA9~16 | DATA17 | CRCH | CRCL |
| 0x7E | 0x14 | 0x01 | 0x01 | 0x01 | RX\_NAME(T) | PASSWORD(T) | LOOP\_ID |  |  |

**Note**: LOOP\_ID is only used for BIND answer. It avoids that 2 receivers answer in the same time on plane power up

 It will be removed once a better solution is found to avoid this BIND answer collision

4. TX returns value to Radio

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~8 | DATA9~16 | CRCH | CRCL |
| 0x7E | 0x13 | 0x01 | 0x01 | 0x01 | RX\_NAME(R) | PASSWORD(R) |  |  |

**Note**: Register is successful if RX\_NAME(T) is equal to RX\_NAME(R) and PASSWORD(T) is equal to PASSWORD(R)

## Bind receiver

1. Radio to TX Modules in bind mode

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA8 | CRCH | CRCL |
| 0x7E | 0x0B | 0x01 | 0x02 | 0x00 | PASSWORD(T) |  |  |

2. TX returns the id received

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA8 | CRCH | CRCL |
| 0x7E | 0x0B | 0x01 | 0x02 | 0x00 | RX\_NAME(R) |  |  |

3. Radio selects one receiver to bind (the RX\_NAME is used to select the receiver which is bind)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~8 | DATA9 | DATA10 | CRCH | CRCL |
| 0x7E | 0x0D | 0x01 | 0x02 | 0x01 | RX\_NAME(T) | Table 4.3‑1 | RX\_NUM |  |  |

**Note: RX\_NUM is used for Model Match feature**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| **R9M** | **LBT mode**0: 8ch with telemetry1: 16ch with telemetry2: 16ch without telemetry3: unused | **FLEX mode**0: Flex 9151: Flex 868 | unused | RX\_UID(T) |
| **ISRM**(ACCST mode) | **0: CH1~8****1: CH9~16** | **0: Telemetry OFF****1: Telemetry ON** |  |  |  |

Table 4.3‑1

4. TX returns the device (RX) information.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~8 | DATA9 | DATA10 | CRCH | CRCL |
| 0x7E | 0x0D | 0x01 | 0x02 | 0x01 | RX\_NAME(T) | Table 4.3‑1 | RX\_NUM |  |  |

**Note**: Bind is successful if RX\_NAME(R) is equal to RX\_NAME(T)

## Reset receiver

**Radio to TX**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1 | CRCH | CRCL |
| 0x7E | 0x04 | 0x01 | 0x08 | RX\_UID | Reset flag:0x01 : RESET BIND 0x02 : RESET SETTINGS0x04 : RESET NAME + LOOP\_ID...0xFF : RESET ALL (factory defaults) |  |  |

**TX to Radio (only sent when**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | CRCH | CRCL |
| 0x7E | 0x03 | 0x01 | 0x08 | RX\_UID |  |  |

## Channel frame

|  |  |  |  |
| --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID |
| 0x7E | Len (depending on channels count) | 0x01 | 0x03 |

**In 8CH / 16CH modes only the 8 / 16 first channels are sent in this frame**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DATA0 | DATA1 | DATA2 | DATA3 | DATA4 | DATA5 | DATA6 | DATA7 |
| Flag0 | Flag1 | ppm1 | ppm1/2 | ppm2 | ppm3 | ppm3/4 | ppm4 |
|  |  |  |  |  |  |  |  |
| DATA8 | DATA9 | DATA10 | DATA11 | DATA12 | DATA13 | DATA14 | DATA15 |
| ppm5 | ppm5/6 | ppm6 | ppm7 | ppm7/8 | ppm8 | ppm9 | ppm9/10 |
|  |  |  |  |  |  |  |  |
| DATA16 | DATA17 | DATA18 | DATA19 | DATA20 | DATA21 | DATA22 | DATA23 |
| ppm10 | ppm11 | ppm11/12 | ppm12 | ppm13 | ppm13/14 | ppm14/15 | ppm15 |
|  |  |  |  |  |  |  |  |
| DATA24 | DATA25 | DATA26 | DATA27 | DATA28 | DATA29 | DATA30 | DATA31 |
| ppm15/16 | ppm16 | ppm17 | ppm17/18 | ppm18 | ppm19 | ppm19/20 | ppm20 |
|  |  |  |  |  |  |  |  |
| DATA32 | DATA33 | DATA34 | DATA35 | DATA36 | DATA37 |  |  |
| ppm21 | ppm21/22 | ppm22 | ppm23 | ppm23/24 | ppm24 |  |  |

Table-103A

**Flag0 (DATA0):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| RANGE CHECK | SET FAILSAFE | RX\_NUM |

**Note: RX\_NUM was given on BIND for Model Match feature**

**Flag1 (DATA1) for 2.4GHz (ISRM family)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| **TX RF Protocol：**0x00 = ACCESS0x01 = ACCST - D160x02 = ACCST - LR12 | unused | unused | unused | unused |

**Flag1 (DATA1) for 900MHz (R9M family)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| **TX RF Protocol：**0x00 = FCC0x01 = EU0x02 = FLEX | unused | unused | unused | unused |

## GET / SET TX Parameters

**Radio to TX**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1 | DATA2 | CRCH | CRCL |
| 0x7E |  | 0x01 | 0x04 | **Flag0** | **Flag1***(only sent on SET)* | Power (in dBM)*(only sent on SET)* |  |  |

**Flag0 (DATA0):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 0: Response freedom1: No response if no received cmd | 0: GET1: SET | unused | unused | unused | unused | unused | unused |

**Flag1 (DATA1):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| unused | unused | unused | unused | External antenna | unused | unused | unused |

Correspondance table for common power values (dBm => mW conversion):

|  |  |
| --- | --- |
| 10 | 10mW |
| 14 | 25mW |
| 20 | 100mW |
| 23 | 200mW |
| 27 | 500mW |
| 30 | 1000mW |

**TX to Radio**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1 | DATA2 | CRCH | CRCL |
| 0x7E |  | 0x01 | 0x04 | **Flag0** | **Flag1** | Power-127~127dbm |  |  |

**Flag0 (DATA0):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 0: Response freedom1: No response if no received cmd | reserved | unused | unused | unused | unused | unused | unused |

**Flag1 (DATA1):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| **TX RF Protocol：**- X16 = 0x00- ... | External antenna | unused | unused | unused |

## GET / SET RX Parameters

**Radio to TX**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1 | DATA2~DATAx | CRCH | CRCL |
| 0x7E |  | 0x01 | 0x05 | **Flag0** | **Flag1***(only sent on SET)* | Data2: Pin 1 channelData9: Pin 8 channel...*(only sent on SET)* |  |  |

**Flag0 (DATA0):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 0: Response freedom1: No response if no received cmd | 0: GET1: SET | unused | RX\_UID |

**Flag1 (DATA1):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 0: Telemetry enable1: Telemetry disable | reserved | PWM period:0 = 18ms1 = 9ms | unused | unused | unused | unused |

**TX to Radio**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1 | DATA2~DATAx | CRCH | CRCL |
| 0x7E |  | 0x01 | 0x05 | **Flag0** | **Flag1** | Data2: Pin 1 channelData9: Pin 8 channel... |  |  |

**Flag0 (DATA0):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 0: Response freedom1: No response if no received cmd | unused | unused | unused | RX\_UID |

**Flag1 (DATA1):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 0: Telemetry enable1: Telemetry disable | 1: RX Settings READONLY0: RX Settings READWRITE | PWM period:0 = 18ms1 = 9ms | unused | unused | unused | unused |

Note: RX Settings (BIT6) can only be read (so that in SHARE mode, the user won’t be able to modify these settings)

## GET Hardware Information

**Radio to TX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | CRCH | CRCL |
| 0x7E | 0x03 | 0x01 | 0x06 | RX\_UID:- 0 ~ 15 : Receivers- 0xFF : TX module |  |  |

**TX to Radio**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1 | DATA2~3 | DATA4~DATA5 | DATA6 | CRCH | CRCL |
| 0x7E | 0x09 | 0x01 | 0x06 | RX\_UID:- 0~15: RX- 0xFF: TX  | RECEIVER / MODULEModel ID | HARDWARE VERSION | SOFTWARE VERSION | VARIANT |  |  |

**R9M / R9 Variants:**

|  |  |
| --- | --- |
| 0x00 | --- |
| 0x01 | FCC |
| 0x02 | EU (LBT) |
| 0x03 | FLEX |

**HARDWARE / SOFTWARE VERSION format**

|  |  |  |
| --- | --- | --- |
| BYTE1 | BYTE2 – BIT4~7  | BYTE2 – BIT0~3  |
| Major version | Minor version | Revision |

**MODULE Model ID list**

|  |  |
| --- | --- |
| 0x00 | --- |
| 0x01 | XJT |
| 0x02 | ISRM |
| 0x03 | ISRM-PRO |
| 0x04 | ISRM-S |
| 0x05 | R9M |
| 0x06 | R9MLite |
| 0x07 | R9MLite-PRO |
| 0x08 | ISRM-N |

**RECEIVER Model ID list:**

|  |  |
| --- | --- |
| 0x00 | --- |
| 0x01 | X8R |
| 0x02 | RX8R |
| 0x03 | RX8R-PRO |
| 0x04 | RX6R |
| 0x05 | RX4R |
| 0x06 | G-RX8 |
| 0x07 | G-RX6 |
| 0x08 | X6R |
| 0x09 | X4R |
| 0x0A | X4R-SB |
| 0x0B | XSR |
| 0x0C | XSR-M |
| 0x0D | RXSR |
| 0x0E | S6R |
| 0x0F | S8R |
| 0x10 | XM |
| 0x11 | XM+ |
| 0x12 | XMR |
| 0x13 | R9 |
| 0x14 | R9-SLIM |
| 0x15 | R9-SLIM+ |
| 0x16 | R9-MINI |
| 0x17 | R9-MM |
| 0x18 | R9-STAB |
| 0x19 | R9-MINI-O |
| 0x1A | R9-MM-O |
|  |  |
|  |  |

## Share command

1. Radio to TX Modules in bind mode

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | CRCH | CRCL |
| 0x7E |  | 0x01 | 0x07 | RX\_UID(T) |  |  |

Note: RX\_UID(T) = 0x0F means share all receivers

2. TX returns one frame per receiver

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | CRCH | CRCL |
| 0x7E |  | 0x01 | 0x07 | RX\_UID(R) |  |  |

## Telemetry frame

Telemetry frame are frames coming from internal TX / RX telemetry or from external sensors.

They are also used in PUSH mode (The TX sends a frame to a receiver, specified with its RX\_UID)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1 | DATA2 | DATA3~4 | DATA5~8 | CRCH | CRCL |
| 0x7E | 0x0B | 0x01 | 0xFE | Flag0 | PhysicalID | PrimID | AppID | Value |  |  |

**Note: See S.PORT protocol for PrimID and DATA description**

**Flag0 (DATA0):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| unused | unused | unused | unused | RX\_UID |

## Power Meter command

**Radio to TX**

|  |  |  |  |
| --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID |
| 0x7E | 0x07 | 0x02 | 0x01 |

|  |  |
| --- | --- |
| DATA0 | DATA1~4 |
| Bit 7: 0: Response freedom1: no response if no received cmdBit 6: 1: write0: readBit 5~0: unused | Freq (Hz)Low endian mode |

**TX to Radio**

|  |  |  |  |
| --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID |
| 0x7E | 0x07 | 0x02 | 0x01 |

|  |  |  |
| --- | --- | --- |
| DATA0 | DATA1~4 | DATA5~6 |
| Bit 7: 0: response freedom1: no response if no received cmdBit 6:1: write0: readBit 5~0: unused | Frequency (Hz)Low endian mode | Power (-327.67 => +327.67dbm)Low endian mode |

## Spectrum Analyser command

Frequency range: 2400MHz – 2485MHz

Maximum span: 80MHz

**Radio to TX**

|  |  |  |  |
| --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID |
| 0x7E | 0x0F | 0x02 | 0x02 |

|  |  |  |  |
| --- | --- | --- | --- |
| DATA0 | DATA1~4 | DATA5~8 | DATA9~12 |
| Bit 7:0: Response freedom1: no response if no received cmdBit 6:1: write0: readBit 5~0: unused | Center frequency (Hz)Low endian mode | Frequency span (Hz)Low endian mode | Bandwidth (Hz)Low endian mode |

**TX to Radio**

|  |  |  |  |
| --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID |
| 0x7E | 0x08 | 0x02 | 0x02 |

|  |  |  |
| --- | --- | --- |
| DATA0 | DATA1~4 | DATA5 |
| Bit 7: 0: response freedom1: no response if no received cmdBit 6:1: write0: readBit 5:Bit 4~0: unused | Current frequency (Hz)Low endian mode | Current Power(dbm)Power ( -127 ~ 127dbm) |

## OTA Update

The user first selects the right FRK file to update. The radio then starts BIND mode to reach the receiver(s).

1. Radio to TX Modules in bind mode

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA8 | CRCH | CRCL |
| 0x7E | 0x0B | 0x01 | 0x02 | 0x00 | PASSWORD(T) |  |  |

2. TX returns the id received

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA8 | CRCH | CRCL |
| 0x7E | 0x0B | 0x01 | 0x02 | 0x00 | RX\_NAME(R) |  |  |

The radio will now show a popup with all returned receivers and the user will choose one.

3. Radio to TX (to retrieve the current RX version)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA8 | CRCH | CRCL |
| 0x7E | 0x0B | 0x01 | 0x02 | 0x02 | RX\_NAME |  |  |

4. TX to Radio

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA8 | DATA9 | DATA10~11 | DATA12~DATA13 | DATA14 |
| 0x7E | 0x11 | 0x01 | 0x02 | 0x02 | RX\_NAME | RECEIVER / MODULEModel ID | HARDWARE VERSION | SOFTWARE VERSION | VARIANT |

|  |  |
| --- | --- |
| CRCH | CRCL |
|  |  |

The radio will display the version information and ask confirmation to the user.

5. Radio to TX (to start the update process)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA8 | CRCH | CRCL |
| 0x7E | 0x0B | 0xFE | 0x02 | 0x00 | RX\_NAME |  |  |

Then the binary is transmitted until the EOF. Each frame carries 32 bytes of data.

6. TX to Radio (acknowledge update process)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA8 | CRCH | CRCL |
| 0x7E | 0x0B | 0xFE | 0x02 | 0x00 | RX\_NAME |  |  |

Then the binary is transmitted until the EOF. Each frame carries 32 bytes of data.

7. Radio to TX

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA4 | DATA5~36 | CRCH | CRCL |
| 0x7E | 0x27 | 0xFE | 0x02 | 0x01 | ADDRESSLow endian mode | DATA(32 bytes) |  |  |

8. TX to Radio

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | DATA1~DATA4 | CRCH | CRCL |
| 0x7E | 0x07 | 0xFE | 0x02 | 0x01 | ADDRESSLow endian mode |  |  |

9. Radio to TX (EOF)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | CRCH | CRCL |
| 0x7E | 0x03 | 0xFE | 0x02 | 0x02 |  |  |

10. TX to Radio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HEAD | LEN | TYPE\_C | TYPE\_ID | DATA0 | CRCH | CRCL |
| 0x7E | 0x26 | 0xFE | 0x02 | 0x02 |  |  |