INRM301 Communication Protocol

Date：2021-12-13

Version：V1.0

Doc No.：

Confidentiality level：For internal use

深圳市富斯科技有限公司

FLYSKY Technology Co.,Ltd

16/F, huafeng building, 6006 shennan Road, Shenzhen, P.R. China
Fax +86 755 8328 0770 Tel +86 755 8328 2965 ext. 8866

All rights reserved

Disclaimer

Shenzhen Flysky Technology Co., Ltd. reserves the right to make changes to the functionality or design of the product to improving performance. Shenzhen Flysky Technology Co., Ltd. does not assume any joint and several liability arising from use of this product.

Revision History

Revision Date	Version	Revision Content	Revised by	Approved by
2021-12-13	V1.0	New	罗平	

深圳市富斯科技有限公司

Contents

I. Communication Rule

1. Communication Convention

This protocol defines the communication method between the transmitter motherboard and RF module. The communication relationships between the transmitter motherboard and RF module are as follows:

(1) The transmitter motherboard or RF module sends commands, and the recipient makes a response with or without parameters.

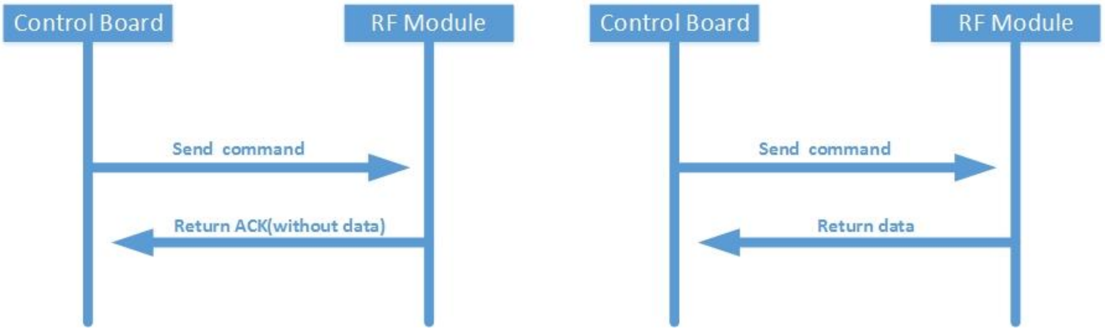


Figure 1 The transmitter motherboard sends a command.

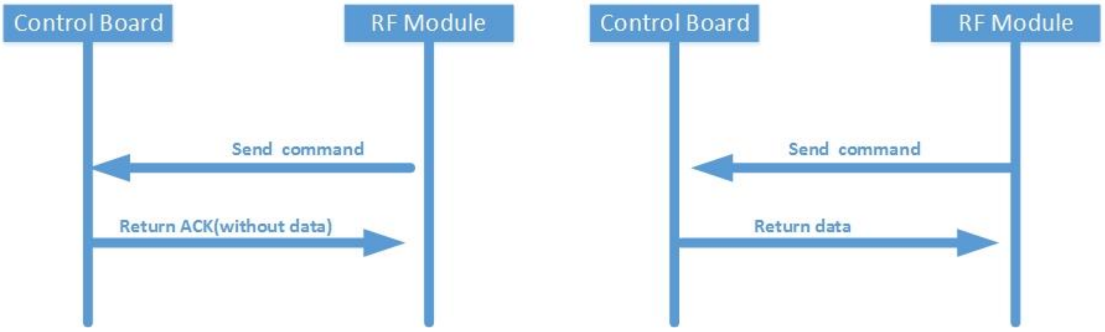


Figure 2 The RF module takes the initiative to send commands.

(2) The transmitter motherboard/RF module sends data, and the recipient does not need to make a response.

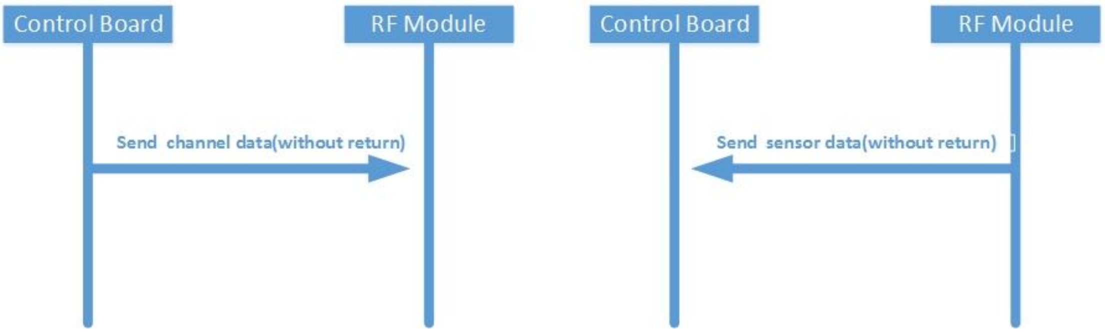

Control Board
RF Module
Send channel data(without return)
Control Board
RF Module
Send sensor data(without return)

Figure 3 Real-time data transmission

2. Communication Rule Description

2.1 Sending Ready Commands

Before transmitter motherboard communicates with RF module, it must send the ready command to RF module first. If RF module returns it is ready, transmitter motherboard can send other commands. If RF module does not return or returns it is not ready, transmitter motherboard cannot send other commands. When RF module is not ready, it will respond with the command format of not ready if it receives commands other than ready command.

2.2 Sending Commands

The sender sends the corresponding command to the recipient, and the recipient first judges whether the received frame is correct. If yes, there are two response methods:

- When parameters need to be returned: the recipient returns the command in the response command format with parameters.
- When parameters do not need to be returned: the recipient returns the command in the response command format without parameters.

If the received frame is incorrect, no response is made. If the sender does not receive a reply from the recipient within 5ms after sending the frame, the sender resends the current frame (up to 5 times). If the sender fails to resend the current frame after five attempts, the current frame is abandoned. If the sender cannot receive the response frame from the recipient, and the recipient can receive information frame sent by the sender, the receiver will only process the same frame only once. A response is still required.

2.3 Real-time Data Transmission

Real-time transmission applies in the following two cases:

- Transmitter motherboard sends channel data to RF module in real-time transmission method. The RF module does not make a response.
- RF module sends sensor data to the transmitter motherboard in real-time transmission method. Transmitter motherboard does not make a response.

2.4 Others

If there is a new frame to be sent when the frame data is being sent, the new frame data shall be sent only after the current frame is sent. (For example, the next frame can be sent after the response frame is completed or it expires)

(1) Frames with the same FrameNumber, PROCOTOLID and CHECKSUM are the same frames.

(2) When multi-byte parameters are sent and received, the low byte is in front (small-end mode).

II. Communication Format

- Communication mode: USART
- Baud rate : 1.5Mbit/s
- Data bits : 8 bit
- Stop bit : 1 bit
- Parity check: None

1. Frame Format

END	Address	Frame Number	Frame Type	Protocol ID	DATA0…DATAn	CHECKSUM	END

2. Information Frame

(1) END: Indicates the head and tail of the frame packet;

(2) Address(char): Indicates the sender's device address (lower 4 bits) and the recipient's device address (higher 4 bits);

Frame Number(char): Used to distinguish data of different frames. For example, the sequence number (0-255) is assigned when a response frame is sent and is automatically added by 1 for each successful sending of frame (if it fails in 5 attempts, the frame number is also automatically added by 1 when a new frame is sent). When the sequence number has reached 255, it should be incremented from 0, and so on. (In special cases, if more than one frame is received at the same time, only the last frame will be answered)

(3)

(4))Frame Type(char): The type of the frame. See the frame type table.

(5) Protocol ID(char): Indicates the protocol ID of the frame. See the frame

深圳市富斯科技有限公司

function table.

(6) DATA0 ~ DATAn: Message content sent or responded to (small-end mode).

(7) CHECKSUM(char) = (Address + Frame Number + Frame Type + Protocal ID + DATA0 + …DATAn)^0xFF。(Note: If replacement characters appear in the frame content, the checksum must be calculated using the original data.)

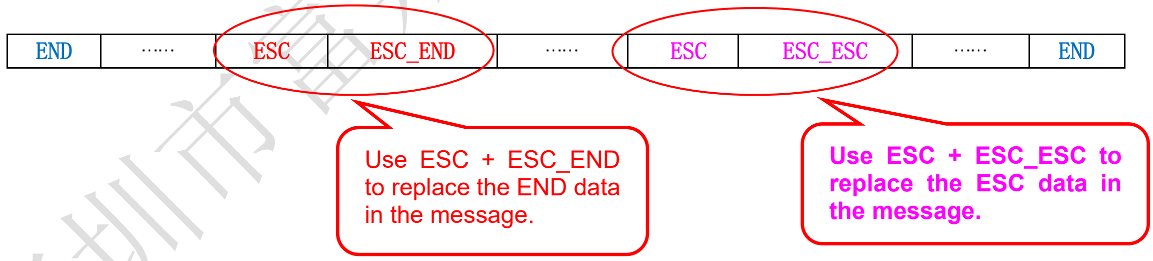
深圳市富斯科技有限公司

3. Special Characters in Communication Protocol

(1) Special Character Table

Special characters	Character value	Description
END	0xC0	Frame header or trailer
ESC	0xDB	If there is a character with the same END or ESC in the transmitted data, the ESC character is sent first.
ESC_END	0xDC	If there is a character with the same END in the data, the ESC character is sent first, and then the ESC_END character is sent afterwards.
ESC_ESC	0xDD	If there is a character with the same ESC in the data, the ESC character is sent first, and then the ESC_ESC character is sent afterwards.

(2) Transmission format, the following format is a frame of data.



Use ESC + ESC_END to replace the END data in the message.

Use ESC + ESC_ESC to replace the ESC data in the message.

III. Protocol Content

END	Address	Frame Number	Frame Type	Protocol ID	DATA0…DATAn	CHECKSUM	END

1. Frame Address

Transmitter address ： 0x01
RF module address： 0x05

2. Frame Type Table

Frame Type	Function Description
0x01	The sender reads data from the recipient.
0x02	With the settings of the recipient performed by the sender, the recipient makes a response with parameters.
0x03	With the settings of the recipient performed by the sender, the recipient makes a response without parameters.
0x05	The sender sends one-way real-time transmission data. The recipient does not make a response.
0x10	Response with parameters (the recipient responds to the sender's command with parameters)
0x20	Response without parameters (the recipient responds to the sender's command without parameters and only returns the received frame ID to the host)

3. Frame Function Table

Protocol ID	Function Description
0x01	RF module ready
0x02	RF module status
0x03	RF module mode setting
0x04	RF module binding parameter configuration
0x06	RF module binding parameter reading
0x07	Transmitter real-time data CH/failsafe
0x09	Receiver returns real time data
0x0C	Command
0x0D	Returns command results
0x0E	RF module radio frequency test
0x20	RF module version information
0x2F	RF module model setting
0x30	Start update of the receiver firmware
0x31	Request receiver firmware
0x32	Update receiver firmware results
0x33	Update mode receiver firmware information

市富斯科技有限公司

4. Communication Frame Content Table

4.1 RF Module Ready

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x01	**Transmitter reads data from RF module.**	
Protocol ID	BIT(7-0)	0x01	**RF module ready**	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x10	**RF module makes response to transmitter with parameters.**	
Protocol ID	BIT(7-0)	0x01	**RF module ready**	
DATA0	BIT(7-0)	X	0x01： **Unready** 0x02：Ready	

深圳市富斯科技有限公

4.2 RF Module Status

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x01	Transmitter reads data from the RF module.	
Protocol ID	BIT(7-0)	0x02	RF module status	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x10	RF module makes response to transmitter with parameters.	
Protocol ID	BIT(7-0)	0x02	RF module status	
DATA0	BIT(7-0)	X	RF module status: 0x01: Hardware error 0x02: Binding 0x03: Synchronizing 0x04: Synchronized 0x05: In standby 0x06: RF module to be updated 0x07: RF module updating 0x08: Update receiver wirelessly 0x09: Wireless receiver update failed 0x0A: RF module radio frequency test 0xFF: Hardware test mode	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0X03	RF module sends message to transmitter.	Notify transmitter actively when RF module status hanges
Protocol ID	BIT(7-0)	0x02	RF module status	
DATA0	BIT(7-0)	X	RF module status: 0x01: Hardware error 0x02: Binding 0x03: Synchronizing 0x04: Synchronized 0x05: In standby 0x06: RF module to be updated 0x07: RF module updating 0x08: Wireless update receiver 0x09: Wireless receiver update failed 0x0A: RF module radio frequency test 0xFF: Hardware test mode	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x20	Transmitter makes response to RF module without parameters.	
Protocol ID	BIT(7-0)	0x02	RF module status	

深圳市富斯科技有限公司

4.3 RF Module Mode Setting

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x02	**With the settings of RF module performed by transmitter, RF module makes a response with parameters.**	Different modes have no priority and can be switched arbitrarily.
Protocol ID	BIT(7-0)	0x03	**RF module state setting**	
DATA0	BIT(7-0)	X	0x01： **Enter standby mode** 0x02：**Enter binding mode (Automatically enter normal communication state after successful binding, except for one-way communication)** 0x03：**Enter normal communication mode** 0x04：**Enter update receiver mode (After update is finished, it automatically enters standby mode. If failed, it enters Wireless receiver update failure state)** 0x55： **Enter hardware test mode (internal)**	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x10	**RF module makes response to transmitter with parameters.**	
Protocol ID	BIT(7-0)	0x03	**RF module state setting**	
DATA0	BIT(7-0)	X	0x01： **Unsuccessful** 0x02： **Successful**	

4.4 RF Module Binding Parameter Configuration

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x02	**With the settings of RF module performed by the transmitter, RF module makes a response with parameters.**	**The module configuration can only be effective after successful binding.**
Protocol ID	BIT(7-0)	0x04	**RF module binding parameter configuration**	

DATA0~N	BIT(7-0)	X	**The binding structure is as follows:**
			typedef struct __attribute__((packed))
			{
			  unsigned char Version;//=0
			  eLNK_EMIStandard EMIStandard;
			  unsigned char IsTwoWay;
			  eDATA_PHYMODE PhyMode;
			  unsigned char SignalStrengthRCChannelNb; //0xFF if not used, 0`18
			  unsigned short FailsafeTimeout;//in unit of ms
			  signed short FailSafe[MAX_RF_CHANNELS_NUMBER];
			  unsigned char FailsafeOutputMode;//TRUE Or FALSE
			  sSES_PWMFrequencyV0 PWMFrequency;
			  eSES_PA_SetAnalogOutput AnalogOutput;
			  eEB_BusType ExternalBusType;
			}sDATA_ConfigV0;
			
			typedef struct __attribute__((packed))
			{
			  unsigned char Version;//=1
			  eLNK_EMIStandard EMIStandard;
			  unsigned char IsTwoWay;
			  eDATA_PHYMODE PhyMode;
			  unsigned char SignalStrengthRCChannelNb;// 0xFF if not used , 0`18
			  unsigned short FailsafeTimeout;//in unit of ms	

深圳市富斯科技有

			signed short FailSafe[MAX_RF_CHANNELS_NUMBER];
			unsigned char FailsafeOutputMode; //TRUE Or FALSE
			eSES_NewPortType NewPortTypes[SES_NPT_NB_MAX_PORTS];
			sSES_PWMFrequenciesAPPV1 PWMFrequenciesV1;
			}sDATA_ConfigV1;
			
			The structure types/enumeration types/macro definitions contained in the binding structure are as follows:
			typedef enum
			{
			CLASSIC_FLCR1_18CH=0,
			CLASSIC_FLCR6_10CH,
			ROUTINE_FLCR1_18CH,
			ROUTINE_FLCR6_8CH,
			ROUTINE_LORA_12CH
			}eDATA_PHYMODE;
			
			typedef enum
			{
			LNK_ES_FREE,
			LNK_ES_CE,
			LNK_ES_FCC
			} eLNK_EMIStandard;
			
			typedef struct __attribute__((packed))
			{
			unsigned short Frequency:15; // From 50 to 400Hz
			unsigned short Synchronized:1; // 1=Synchronize the PWM output to the RF cycle (lower latency but unstable frequency)
			} sSES_PWMFrequencyV0;
			
			typedef enum __attribute__((packed))
			{
			SES_ANALOG_OUTPUT_PWM,
			SES_ANALOG_OUTPUT_PPM
			} eSES_PA_SetAnalogOutput;
			
			typedef enum
			{
			EB_BT_IBUS1,
			EB_BT_IBUS2,
			EB_BT_SBUS1
			} eEB_BusType;
			
			typedef enum
			{
			SES_NPT_PWM,
			SES_NPT_PPM,
			SES_NPT_SBUS,
			SES_NPT_IBUS1_IN,
			SES_NPT_IBUS1_OUT,
			SES_NPT_IBUS2,
			SES_NPT_IBUS2_HUB_PORT,
			SES_NPT_WSTX,
			SES_NPT_WSRX,
			SES_NPT_NONE=0xFF
			} eSES_NewPortType;	

深圳市富斯科技有限公司

			// This structure may be used by main applications to store the PWM parameters in a single convenient structure
			typedef struct __attribute__((packed))
			{
			  unsigned short PWMFrequencies[SES_NB_MAX_CHANNELS]; // One unsigned short per channel, From 50 to 400Hz ,1:1000Hz,2:833Hz
			  unsigned long Synchronized; // 1 bit per channel, 32 channels total
			} sSES_PWMFrequenciesAPPV1;
			
			#define SES_NB_MAX_CHANNELS (32)
			#define MAX_RF_CHANNELS_NUMBER (18)
			#define SES_NPT_NB_MAX_PORTS (4)	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x10	**RF module makes response to transmitter with parameters.**	**Configuration fails in the binding state.**
Protocol ID	BIT(7-0)	0x04	**RF module binding parameter configuration**	
DATA0	BIT(7-0)	X	0x01: Unsuccessful 0x02: Successful	

深圳市富斯科技有限公司

4.5 RF Module Reading of Binding Parameters

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x01	**Transmitter reads data from the RF module.**	
Protocol ID	BIT(7-0)	0x06	**RF module binding parameter reading**	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x10	The **RF module makes response to transmitter with parameters.**	
Protocol ID	BIT(7-0)	0x06	**RF module binding parameter reading**	

DATA0~N	BIT(7-0)	X	typedef union __attribute__((packed))
			{
			unsigned char Version;
			sDATA_ConfigV0 ConfigV0;
			sDATA_ConfigV1 ConfigV1;
			}uDATA_Config;	

深圳市富斯科技有限公司

4.6 Transmitter Real-Time Data CH/Failsafe

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x05	**Transmitter sends one-way real-time transmission data. RF module does not make response.**	
Protocol ID	BIT(7-0)	0x07	**Transmitter real-time data CH/failsafe**	
DATA0	BIT(7-0)	X	0x01： **Real-time data CH** 0x02： **Failsafe (Applied only in one-way communication. In two-way communication, command instructions are used)**	
DATA1	BIT(7-0)	X	**Number of channels**	
DATA2~N	BIT(7-0)	X	**Byte length = 2 (signed short) * number of channels (Channel data range -15000~15000)**	

4.7 Transmitter Real-Time OEM Data

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x05	**Transmitter sends one-way real-time transmission data. RF module does not make response.**	
Protocol ID	BIT(7-0)	0x08	**Transmitter real-time data CH/failsafe**	
DATA0	BIT(7-0)	X	Cammand	
DATA1~N	BIT(7-0)	X	TXPayloadLength	

4.8 Receiver Returning Real-Time Data

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x05	**The RF module sends one-way real-time transmission data. Transmitter does not make response.**	
Protocol ID	BIT(7-0)	0x09	**Receiver returns real-time data**	
DATA0	BIT(7-0)	0x22	**Sensor data command**	
DATA1~10	BIT(7-0)	X	(unsigned char *10) **Data format: length + type + ID + content** **Example：** 0x06 0x56 0x00 0x20 0xF4 0x01 0x06： **The total frame length is 6 bytes.**	

深圳市富斯科技有限公司

			0x56： Frame type and RF module internal information 0x00：ID=0 0x20： RF module temperature=32degC 0x01F4： RF module external power supply voltage = 500 (i.e., 5V)	

4.9 Command

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x02	**With settings of the RF module performed by transmitter, RF module makes a response with parameters.**	**This command is valid only in normal communication mode.**
Protocol ID	BIT(7-0)	0x0C	**Command**	
DATA0-1	BIT(7-0)	X	Code: **Read receiver capability：** 0x7015 **Set PPM/PWM (V0)：** 0x7016 **Set PWM frequency (V0)：** 0x7017 **Set PWM frequency (V1)：** 0x7028 **Set I-BUS/S-BUS (V0)：** 0x7018 **Set I-BUS IN/I-BUS OUT (V0)：** 0x7020 **Set custom interface type (V1)：** 0x7027 **Read receiver version information：** 0x701F **BVD calibration：** 0x702C **Set PPM/I-BUS/S-BUS failsafe ：** 0x702A **Set failsafe：** 0x6011 **Set failsafe time：** 0x6012 **Set RSSI signal output channel：** 0x602B	
DATA2	BIT(7-0)	X	ArgumentLength: **Read receiver capability：** 0 **Set PPM/PWM：** 1 **Set PWM frequency (V0)：** 2 **Set PWM frequency (V1)：** 32+3 **Set I-BUS/S-BUS (V0)：** 1 **Set I-BUS IN/I-BUS OUT (V0)：** 1 **Set custom interface type (V1)：** 4 **Read receiver version information ：** 0 **BVD calibration ：** 8 Set PPM/I-BUS/S-BUS **failsafe：** 1 **Set failsafe：** Channels_Number*2 **Set failsafe time：** 2 **Set RSSI signal output channel：** 1	
DATA3~ArgumentLength+2	BIT(7-0)	X	Argument[ArgumentLength] **Read receiver capability：** 0 Set PPM/PWM： 0：PWM 1：PPM **Set PWM frequency (V0)：** **Value range：** 50-400 **Set PWM frequency (V1)：** Argument[0]：	

深圳市富斯科技有限公

深圳市富斯科技有限公司

			0： The PWM frequency of CH1~CH16 is sent
			1： The PWM frequency of CH17~CH32 is sent.
			Argument[1]~ Argument[ArgumentLength]:
			// This structure may be used by main applications to store the PWM parameters in a single convenient structure
			typedef struct __attribute__((packed))
			{
			  unsigned short PWMFrequencies[SES_NB_MAX_CHANNELS]; // One unsigned short per channel, From 50 to 400Hz ,1:1000Hz,2:833Hz
			  unsigned long Synchronized; // 1 bit per channel, 32 channels total
			} sSES_PWMFrequenciesAPPV1;
			Set I-BUS/S-BUS (V0)：
			0：IBUS1
			1：IBUS2
			2：SBUS1
			Set I-BUS IN/I-BUS OUT (V0)：
			0：I-BUS OUT
			1：I-BUS IN
			Set custom interface type (V1):
			eSES_NewPortType NewPortTypes[4];
			typedef enum
			{
			  SES_NPT_PWM,
			  SES_NPT_PPM,
			  SES_NPT_SBUS,
			  SES_NPT_IBUS1_IN,
			  SES_NPT_IBUS1_OUT,
			  SES_NPT_IBUS2,
			  SES_NPT_IBUS2_HUB_PORT,
			  SES_NPT_WSTX,
			  SES_NPT_WSRX,
			  SES_NPT_NONE=0xFF
			} eSES_NewPortType;
			Read receiver version information： 0
			BVD calibration：
			typedef struct __attribute__((packed))
			{
			  unsigned long ActualInternalVoltage; // Voltage currently supplied to the receiver in unit of 1mV, zero if no calibration needed
			  unsigned long ActualExternalVoltage; // External voltage currently measured by the receiver in unit of 1mV, zero if no calibration needed
			} sSES_CA_CalibrateVoltageMonitorV1;
			Set PPM/I-BUS/S-BUS failsafe：
			0： **Hold last output**
			1： **No output**
			Set failsafe：
			Keep last output： 0x8000
			Set failsafe： -15000 - 15000
			Set failsafe time：
			 >0(ms)
			Set RSSI signal output channel： 0xFF if not	

			used, 0`18	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x10	**RF module makes response to transmitter with parameters.**	
Protocol ID	BIT(7-0)	0x0C	**Command**	
DATA0	BIT(7-0)	X	0x01: Unsuccessful 0x02: Successful	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x03	**RF module sends message to transmitter.**	
Protocol ID	BIT(7-0)	0x0D	**Returns command results**	
DATA0-1	BIT(7-0)	X	Code: **Read receiver capability:** 0x7015 **Set PPM/PWM (V0):** 0x7016 **Set PWM frequency (V0):** 0x7017 **Set PWM frequency** (V1): 0x7028 Set I-BUS/S-BUS(V0): 0x7018 Set I-BUS IN/I-BUS OUT(V0): 0x7020 **Set custom interface type (V1):** 0x7027 **Read receiver version information:** 0x701F **BVD calibration:** 0x702C **Set** PPM/I-BUS/S-BUS **failsafe:** 0x702A **Set failsafe:** 0x6011 **Set failsafe time:** 0x6012 **Set RSSI signal output channel:** 0x602B	
DATA2	BIT(7-0)	X	Result: 0: **Success** 1: **Timeout** 2: **Not supported** 3: **Invalid**	
DATA3	BIT(7-0)	X	ResponseLength(小于 32): **Read receiver capacity:** 32 **Set PPM/PWM (V0):** 0 **Set PWM frequency (V0):** 0 **Set PWM frequency** (V1): 0 Set I-BUS/S-BUS(V0): 0 Set I-BUS IN/I-BUS OUT(V0): 0 **Set custom interface type (V1):** 0 **Read receiver version information** : 14 **BVD calibration** : 4 **Set** PPM/I-BUS/S-BUS **failsafe:** 0 **Set failsafe** : 0 **Set failsafe time:** 0 **Set RSSI signal output channel** : 0	
DATA4~ResponseLength	BIT(7-0)	X	Response[ResponseLength] **Read receiver capability** : V0: typedef struct __attribute__((packed))	

深圳市富斯科技有限公司

深圳市富斯科技有限公司

			{
			unsigned char HasTwoAntennas:1;
			unsigned char HasPWMOutputs:1;
			unsigned char HasPPMOutput:1;
			unsigned char HasExternalWSPort:1;
			unsigned char SupportsIBus1:1;
			unsigned char SupportsIBus2:1;
			unsigned char SupportsSBus:1;
			unsigned char HasDualExternalBusPorts:1;
			
			unsigned char HasDualExternalBusUSARTs:1;
			unsigned char SupportsSVC:1;
			unsigned char Reserved1:6;
			
			unsigned char Reserved2[32-2]; // 256 bits for 256 capabilities
			} sSES_CA_GetCapabilitiesResponseV0;V1:
			typedef struct __attribute__((packed))
			{
			unsigned char NbRCChannels:5;
			unsigned char NbNewPortPorts:3; // From 0 to 4 ports
			
			unsigned char HasTwoAntennas:1;
			unsigned char SupportsSVC:1;
			unsigned char Reserved1:6;
			
			unsigned char Reserved2[32-2]; // 256 bits for 256 capabilities
			} sSES_CA_GetCapabilitiesResponseV1;
			Set PPM/PWM(V0): **None**
			Set PWM **frequency** (V0): **None**
			Set PWM **frequency** (V1): **None**
			Set I-BUS/S-BUS(V0): **None**
			Set I-BUS IN/I-BUS OUT(V0): **None**
			Set custom interface type (V1): **None**
			Read receiver version information :
			typedef struct __attribute__((packed))
			{
			unsigned long ProductNumber;
			unsigned short MainboardVersion;
			unsigned short RFModuleVersion;
			unsigned short BootloaderVersion;
			unsigned short FirmwareVersion;
			unsigned short RFLibraryVersion;
			} sSES_CA_GetVersionResponse;
			BVD **calibration:**
			typedef struct __attribute__((packed))
			{
			unsigned short InternalVoltageCorrection; // 1<<14=1.0
			unsigned short ExternalVoltageCorrection; // 1<<14=1.0
			} sSES_CA_CalibrateVoltageMonitorV1Response;
			if(InternalVoltageCorrection!=0 && \ InternalVoltageCorrection!= 0xffff)
			{
			;//BVD **calibration succeeded**
			}	

			else
			{
			;//BVD calibration failed
			}
			Set PPM/I-BUS/S-BUS failsafe ： None
			Set failsafe ： None
			Set failsafe time： None
			Set RSSI signal output channel： None	

Data sequence	**Bit**	**Value**	**Description**	**Remarks**
Frame Type	BIT(7-0)	0x20	**Transmitter makes response to RF module without parameters.**	
Protocol ID	BIT(7-0)	0x0D	**Returns command results**	

深圳市富斯科技有限公司

5.1 RF Module Version Information Query

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x01	**Transmitter reads data from RF module.**	
Protocol ID	BIT(7-0)	0x1F	**RF module version information**	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x10	**RF module makes response to transmitter with parameters.**	
Protocol ID	BIT(7-0)	0x1F	**RF module version information**	
DATA0~3	BIT(7-0)	X	**Product ID (unsigned long)**	
DATA4~7	BIT(7-0)	X	**Hardware version**（unsigned long)	
DATA8~11	BIT(7-0)	X	Bootloader **version**（unsigned long)	
DATA12~15	BIT(7-0)	X	Firmware **version**（unsigned long)	
DATA16~19	BIT(7-0)	X	RF **version** （unsigned long)	

5.2 RF Module Model Settings

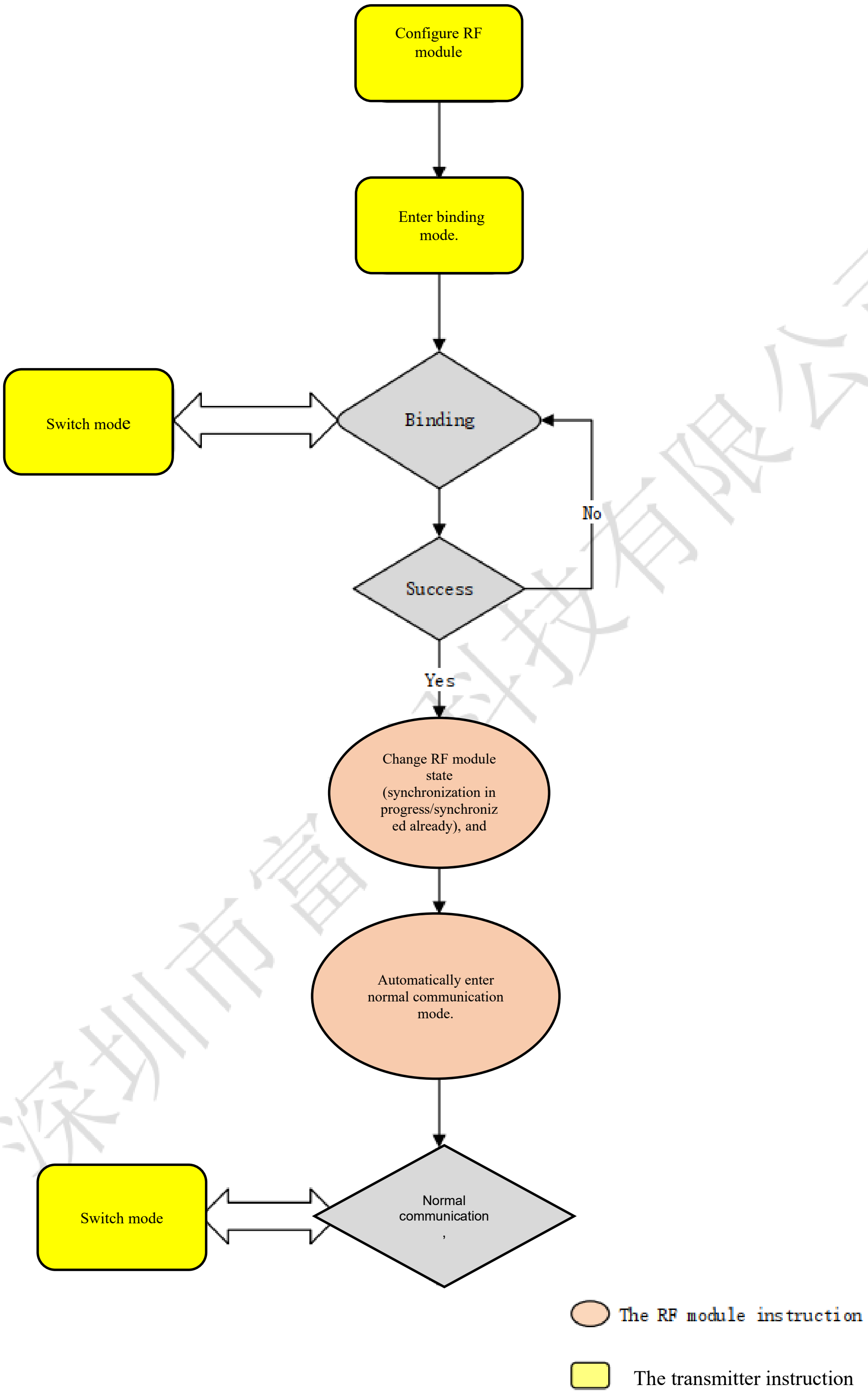
Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x02	**With settings of RF module performed by transmitter, RF module makes a response with parameters.**	Valid when it is in standby mode.
Protocol ID	BIT(7-0)	0x2F	**RF module model setting**	
DATA0	BIT(7-0)	X	**Model setting** (0~19)	

Data sequence	Bit	Value	Description	Remarks
Frame Type	BIT(7-0)	0x10	**RF module makes response to transmitter with parameters.**	
Protocol ID	BIT(7-0)	0x2F	**RF module model setting**	
DATA0	BIT(7-0)	X	0x01：**Unsuccessful** 0x02：**Successful** ；	

深圳 技有限公司

IV. Description of Function Usage

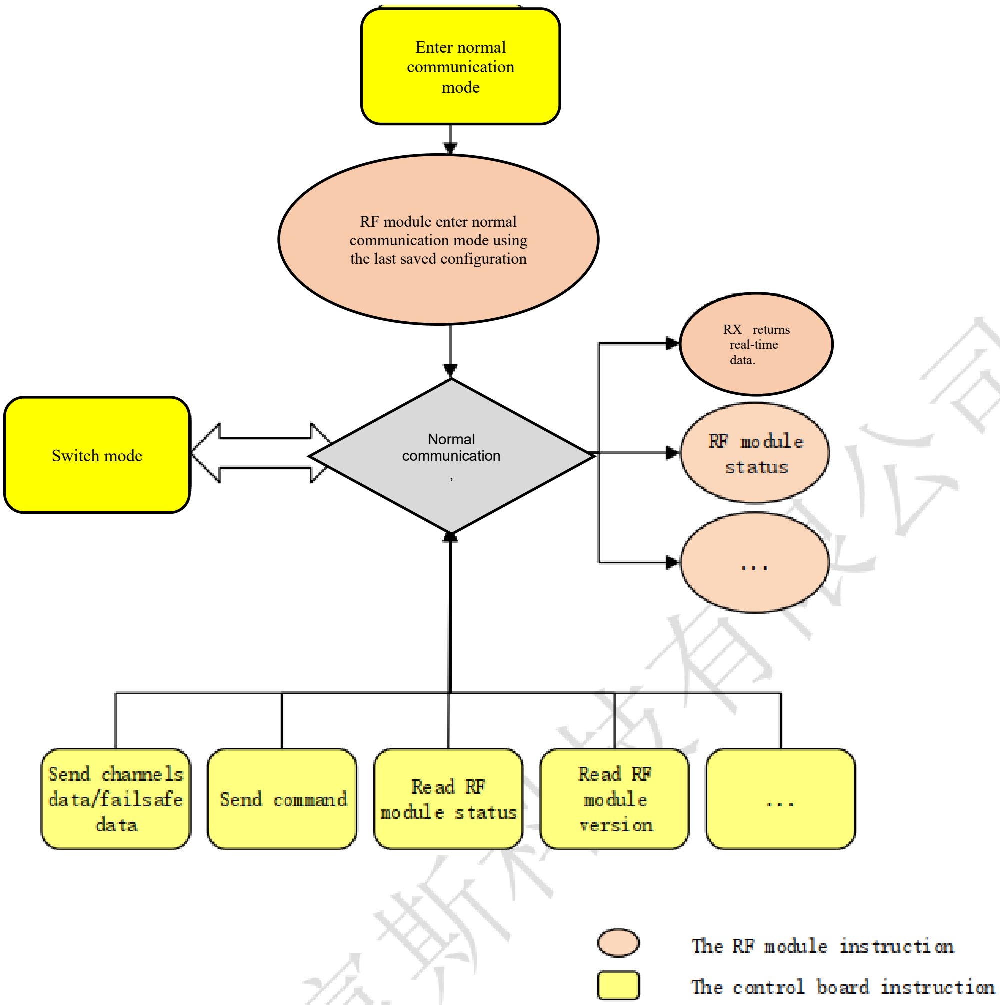
1. Binding


Configure RF module
Enter binding mode.
Switch mode
Binding
No
Success
Yes
Change RF module state (synchronization in progress/synchroniz ed already), and
Automatically enter normal communication mode.
Switch mode
Normal communication
,
The RF module instruction
The transmitter instruction

Binding process

深圳市富斯科技有限公司

2. Normal Communication


Enter normal communication mode
RF module enter normal communication mode using the last saved configuration
Normal communication
Switch mode
RX returns real-time data.
RF module status
...
Send channels data/failsafe data
Send command
Read RF module status
Read RF module version
...
The RF module instruction
The control board instruction

Normal Communication

深圳市富斯科技有限公司