Document Version	Department
V3.3.1	Technology Department
Document Number	**Confidentiality Level**
SS-TC-TM-M-006	Internal Disclosure

Scishine UHF RFID Reader Universal Communication Protocol Version 3.3.1

Prepared by		**Date**	
Approved by		**Date**	
Authorized by		**Date**	

红宇科技

Revision record

Date	Revision Version	Section Number	Change Description	Author

Contents

1 Summary

This paper is divided into 2 logic layers - application layer and transport layer, each layer has its own encoding rules, but both must follow the following principles:

- The protocol data represented by binary byte sequence encoding.
- The multi-byte integer uses big-endian encoding, the data length remained unchanged.
- Byte string directly copies string content.

The application layer and transport layer are described respectively as follows.

2 Transport Layer Protocol

Transmission layer defines the transmission data integrity and transmission target recognition between the reader and the host computer. The transport layer protocol data unit is called frame (Frame) in this text. The structure of transport layer protocol frames are listed in the following table:

No.	Field Tags	Field Type	Code Length	Field Range	Field Description
1	head	Byte string	1 B	0xAA	Preamble Flags
2	src	Integer	1 B		Source address code
3	dest	Integer	1 B		Destination address code
4	plsize	Integer	2 B	[0, 1000]	The number of bytes of payload data encoding
5	payload	Byte string	plsize B		Load (application layer protocol data unit)
6	crc	Integer	2 B		Calibration range: head to crc field (not including head and crc) See Appendix checksum calculation method
7	tail	Byte string	1 B	0x55	Frame end mark

Because of the transport layer protocol associated with the actual communication network, the reader has different frames structure when the reader communicate with the host computer in different communication network, namely actual frame structure field set is a subset of the table. This chapter will describe the transmission frame structure in different communication network.

2.1 RS232

RS232 transmission is one-to-one transmission, which does not involve the multi work station network. The RS232 interface parameter is: baud rate 57600, 8 data bits, parity bit N, 1 stop bit. If special circumstances need to use other baud rate, the actual product shall be noted.

2.1.1 Transmission Frame Structure

No.	Field Tags	Field Type	Code Length	Field Range	Field Description
1	head	Byte string	1 B	0xAA	Preamble Flags
2	dst	Integer	1 B		
3	src	Integer	1 B		
4	plsize	Bitfield	2 B（16bits）		b15: MSB, crc field ignores indication bit, 0 indicates the transmission frame contains crc field, 1 indicates it does not contain the crc; b14-13: Reserved; b12-0: coded payload data bytes, [0, 1000].
5	payload	Byte string	plsize B		Load
6	crc	Integer	2 B	[0, 65535]	Calibration range: head to crc16 fields, excluding head and crc16 fields. Checksum calculation

					method, see the Appendix. If plsize b15 is 1, then the field is not in transmission frames appear.
7	tail	Byte string	1 B	0x55	Frame end mark

2.1.2 Transmission Encoding Supplement

In transmission frames of actual coding process, in order to avoid the same with the head and tail features byte string interfere with the transmission frame decoding of party identification occurring between the head and tail. The characteristics of bytes between head and tail field series (0xAA and 0x55) need to be escaped, provisions escape to 0xFF. Specific processing rules are as follows:

1） When coding, before you need to insert an escape byte 0xFF at the sequence of bytes to be transmitted in each 0xAA, 0x55 and 0xFF byte forming escape coding sequence of bytes, and then sent;

2） When decoding, encode the escape according to the rules, restore the actual data sequence, and then analysis frame.

2.1.3 Transmission Error Handling

In the decoding process of the transmission frame, it may find the incomplete transmission frame, which includes no end, time out, check error, etc. The decoder adopts the discarding strategy for the transmission frame of the error, and does not do any other processing.

2.2 RS485

RS485 transmission is a single management station to multi workstations network transmission. Management station for all workstations have polling operation, and workstations can not initiate the transfer. This agreement about RS485 interface parameter is: 57600 baud rate, 8 data bits, parity bit N, 1 stop bit. If special circumstances need to use other baud rate, the actual product shall be noted.

2.2.1 Transmission Frame Structure

No.	Field Tags	Field Type	Code Length	Field Range	Field Description
1	head	Byte string	1 B	0xAA	Preamble Flags
2	dst	Bitfield	1 B		bit7: 232 on 485 existing communication protocols (half-duplex with a CRC checksum) compatible flag, constant is 1; bit6 ~ 0: destination node number, 0 is station, 127 indicates the broadcast address (station sent broadcast to all stations)
3	src	Integer	1 B	[0, 127)	Source node number, 0 indicates the management station.
4	plsize	Integer	2 B	[0, 1000]	The number of bytes of payload data encoding
5	payload	Byte string	plsize B		Load
6	crc	Integer	2 B	[0, 65535]	Calibration range: head to crc16 fields, excluding head and crc16 fields See Appendix checksum calculation method
7	tail	Byte string	1 B	0x55	Frame end mark

2.2.2 Transmission Encoding Supplement

The same as RS232.

2.2.3 Transmission Error Handling

The same as RS232.

3 Application Layer Protocol

3.1 Application Layer Protocol Data Unit

The logical structure and encoding plan of the application layer protocol data unit (PDU) are as follows:

Field No.	Field Tags	Field Type	Code Length	Field Range	Field Description
1	cmd	Integer	8 bits	[0, 255]	PDU carries application command word, and specific values can see the chapter: Application Command Set.
2	errno	Integer	8 bits	[-128, 127]	Command processing error codes. errno = -128 indicates that the PDU is a command request, and the value of errno itself has no practical significance; errno> -128 indicates that the PDU is the command response, and errno value itself has practical implications. See Appendix: Error code table.
3	argtab	Byte string	n B		PDU carries the command data (parameter list), the number of bytes n is determined by the transport layer plsize (n=plsize-2). The internal structure is determined by the specific CMD. Storage cmd associated with the specific request (errno == -128) or answer (errno> -128) parameter table.

3.2 Applications Command Set

Order is that between a host computer and an adapter, information exchange in order to complete an application function. A complete command includes a

request PDU and an answer PDU, which have the same CMD PDU fields, but the errno values of the two PDU have different definitions: the PDU of the request errno is constant -128, and the errno value of the response is bigger than -128 and it needs to be defined according to the actual application. To the special command without respond, it only have the request for PDU and haven't the response PDU. When a command is executed with error, if without the special instructions, it means that there is no response or the response parameter list is empty.

Command By The Host Computer：

0x00 : connect — connection

- Function:

Establish command interaction context between the upper computer and the reader. Only by setting up an interactive context, the reader can respond to the work command (except:connect, netaddr_get, netaddr_set).

- Request parameters:

No.	Field Name	Type	Length	Range	Description
1	password	Byte string	4B		Connection password, full 0x00 indicates no password to connect
2	customNo	Integer	4B	[0, ffffffff)	Custom number, link custom equipment, 0 represents the common connection

- Response parameters:

No.	Field Name	Type	Length	Range	Description
1	version	Integer bitfield	4B		The firmware version number of the device, within the structure: major version number (1B) + minor version number (1B) + compiler No. (2B), expressed as 0x01000110 1.0.272
2	devSn	Integer	4B	(0, 0xffffffff)	Device serial number, 10 in decimal format is as follows: yy (2) mm (2) sn (6)

3	model	Ascii string	<=16B		Product Model

- Return value:

> 0 - response parameters are valid, but the custom number or password does not match with the device, the connection is unsuccessful, the device does not respond to other commands except connect, netaddr_get and outside netaddr_set.

SSE_SUCCESS

0x01 : config_get — working parameter query

- Function:

Get the work parameter values.

- Request parameters:

No.	Field Name	Type	Length	Range	Description
1	name	Integer	2 B	[0, 65535]	Parameter index and the valid values for the name are also found in the Appendix.

- Response parameters:

No.	Field Name	Type	Length	Range	Description
1	value	Byte string	n		The actual type of the value, length, range is determined by the name. Name and the corresponding value format can be found in the Appendix.

- Return value:

SSE_SUCCESS

SSE_CMD_INV

SSE_FAIL

0x02 : config_begin — work parameter configuration start

- Function:

Start parameter configuration process. During the configuration process,

can not execute commands except parameter query, set, and disconnect command. The RF section is turned off.

- Request parameters: Empty.
- Response parameters: Empty
- Return value:

  SSE_SUCCESS

  SSE_CMD_INV

  SSE_FAIL

0x03 : config_set — working parameter setting

- Function:

  Set a new value for the specified parameters.

- Request parameters:

No.	Field Name	Type	Length	Range	Description
1	name	Integer	2 B	[0, 65535]	Parameter index
2	value	Byte string	n B		The actual type of the value, length, range is determined by the name. The name and the corresponding value format can be found in the Appendix.

- Response parameters: Empty.
- Return value:

  SSE_SUCCESS

  SSE_CMD_INV

  SSE_FAIL

  SSE_INPUT_INV

0x04 : config_end — work parameter configuration end

- Function:

End the parameter configuration process, save the configuration changes. After that, it can execute other commands normally.

- Request parameters:

No.	Field Name	Type	Length	Range	Description
1	act	Integer	1B	{0,1}	0 - abandon parameter setting 1-- save parameter setting

- Response parameters: Empty.
- Return value:

> 0 - successful termination of the configuration process, but the save fails, and lose the change of configuration.

SSE_SUCCESS

SSE_CMD_INV

- Description

If the command does not respond, you should be retried several times. To ensure the success of configuration chang, after this command executed, read the parameters to verify by config_get command is necessary.

0x2E : tag_inventory_auto — start the automatic label inventory

- Function:

Notify the device to enter automatic inventory mode. The device will automatically perform continuous inventory cycle, until tag_inventory_stop command. The command parameters are not permanent.

- Request parameters:

No.	Field Name	Type	Length	Range	Description
1	report	Boolean	1B	{0,1}	1 initiatively report after the inventory (not supported, if you pass this parameter may cause communication failure) 0 storing inventory after

					waiting inquiry

- Response parameters: Empty.
- Return value:

  SSE_SUCCESS

  SSE_FAIL

  SSE_CMD_INV

  SSE_INPUT_INV

0x2F : tag_inventory_stop — stop automatic label inventory

- Function:

  Notice the equipment to exit from the automatic inventory work mode.
- Request parameters: Empty.
- Response parameters: Empty.
- Return value:

  SSE_SUCCESS

  SSE_FAIL

  SSE_CMD_INV

0x30 : tag_inventory_query — inventory tag query

- Function:

  查询盘存到的标签 ID。如果设备不处于自动盘存状态，则设备将即时执行最小 Q 值（由用户设置）的盘存周期。Query the tag ID. If the equipment is not in the automatic inventory state, the equipment will be executed minimum Q value (set by the user) of the inventory cycle.
- Request parameters: empty.
- Response parameters:

No.	Field Name	Type	Length	Range	Description
1	tagCnt	Integer	1 B	[0, 255]	tagList the tag ID number of records
2	tagIDList	record			tagID list.

		string			List: = record *; record: = tagID See Appendix tagID format

- Return value:

  SSE_SUCCESS

  SSE_FAIL

  SSE_CMD_INV

0x31 : tag_access_read — read tag data

- Function:

  Read the data on the specified partition, specified length and specified location of the specified tag.

- Request parameters:

No.	Field Name	Type	Length	Range	Description
1	accPwd	Byte string	4B		Password
3	bank	Integer	1B	[0,3]	Partition number
4	offset	Integer	EBV	>=0	Start Offset (Unit 2B)
5	words	Integer	1B	>0	Read length (Unit 2B)
6	tagID	Byte string	>=1B		Format can be found in the appendix. The empty of tagID indicates the operation is empty.

- Response parameters:

No.	Field Name	Type	Length	Range	Description
1	data	Byte string	Command specifies length		成功读取的数据字节串 The data byte strings which read successfully.
2	tagID	Byte string	>=1B		Format can be found in the appendix. If the request tagID is empty, response tag's tagID, otherwise response empty tagID.

- Return value:

SSE_SUCCESS

SSE_INPUT_INV

SSE_CMD_INV

SSE_FAIL

SSE_MEM_OVR

SSE_MEM_LCK

SSE_TAG_PWR

0x32 : tag_access_write — write tag partition data

- Function:

Write data in the specified partition, specified length and specified location of the specified tag. If it fails, the data in specified area is unpredictable.

- Request parameters:

No.	Field Name	Type	Length	Range	Description
1	accPwd	Byte string	4B		Password
3	bank	Integer	1B	[0, 3]	Partition number
4	offset	Integer	EBV	>=0	Start Offset (Unit 2B)
5	words	Integer	1B	>0	Read length (Unit 2B)
6	data	Byte string	words * 2 B		To write data
7	tagID	Byte string	>=1B		Format can be found in the appendix. The empty of the label indicates that operate any tag.

- Response parameters:

No.	Field Name	Type	Length	Range	Description
1	tagID	Byte string	>=1B		Format can be found in the appendix. If the request tagID is empty, response tag's tagID, otherwise response empty tagID.

- Return value:

  SSE_SUCCESS

  SSE_INPUT_INV

  SSE_CMD_INV

  SSE_FAIL

  SSE_MEM_OVR

  SSE_MEM_LCK

  SSE_TAG_PWR

0x33 : tag_access_lock — tag data access lock

- Function:

  Lock the data access method in the specified partition of the tag.

- Request parameters: Empty.

No.	Field Name	Type	Length	Range	Description
1	accPwd	Byte string	4B		Password
2	lckData	Byte string	3B		Operands, including the target area and the target area with operations, concrete structure can be found in the Appendix
3	tagID	Byte string	>1B	Not empty	Format can be found in the appendix.

- Response parameters: Empty.

- Return value:

  SSE_SUCCESS

  SSE_FAIL

  SSE_INPUT_INV

  SSE_CMD_INV

  SSE_MEM_OVR

  SSE_MEM_LCK

  SSE_TAG_PWR

0x34 : tag_access_kill — tag disablement

- Function:

  Lock the data access method in the specified partition of the tag.

- Request parameters:

No.	Field Name	Type	Length	Range	Description
1	accPwd	Byte string	4B		Access Password
2	killPwd	Byte string	4B		kill Password
3	tagID	Byte string	*		Format can be found in the appendix.

- Response parameters: Empty.
- Return value:

  SSE_SUCCESS

  SSE_FAIL

  SSE_INPUT_INV

  SSE_CMD_INV

  SSE_TAG_PWR

0xEE : connpwd_set — connection password setting

- Function:

  Set a new connection password for the device.

- Request parameters:

No.	Field Name	Type	Length	Range	Description
1	oldPwd	Byte string	4B		Original link password of the device(see the connect command)
2	newPwd	Byte string	4B		The new connection password

- Response parameters: Empty.
- Return value:

  SSE_SUCCESS

SSE_FAIL

SSE_INPUT_INV

SSE_CMD_INV

0xEF : disconnect — disconnect

- Function:

Disconnect the working relationship between the host computer and the device. After this command is executed, except connect, netaddr_get, netaddr_set outside, the rest of the command can not respond.

- Request parameters: Empty.
- Response parameters: Empty.
- Return value:

SSE_SUCCESS

SSE_FAIL

3.3 Application Command General Call Process

(Omission)

4 Appendix

4.1 Error Code Table

Error code identification	Code value	Error Description
……	>0	After the successful execution of the command, the definition of special meaning return value
SSE_SUCCESS	0	The command completed successfully
……	-1~ -49	Reserved
SSE_CMD _INV	-50	Unsupported command
SSE_INPUT _INV	-51	Invalid input parameter
……	-52~ -79	Reserved
SSE_TAG_MEM_OVR	-80	Tag storage position out of bounds
SSE_TAG_MEM_LCK	-81	Tag storage area is locked
SSE_TAG_PWR	-82	Tags energy is not enough
……	-83~-109	Reserved
……..	-110~-126	Custom error code reserved.
SSE_FAIL	-127	Command fails (for unknown reasons or without specify the reasons)

4.2 Reader Operating Parameters Table

The parameters listed in the following table are possible generic parameters, which may be increased or decreased in different types of reader. As appropriate, it is better to set up an independent parameter list according to the model.

Grade	Parameter name	Type	Length	Range	Description
1	invtRspTm	Integer	2B	[-1, 32767]	The maximum response time inventory, a major role in the automatic inventory controls the inventory efficiency and command response balance, in milliseconds. -1 (<0) means that only an inventory of finished or the label ID cache is full, response command (not recommended the design value). > = 0 means any one of the

					following three conditions are satisfied can respond to commands: one end inventory, tag ID cache is full, the arrival response time.
2	invtBufTm	Integer	1B	[0, 31]	标签盘存刷新时间，秒。缺省值0，表示每次识别即时刷新。本参数用于减少标签识别信息的传输数据量 （暂不支持） Label inventory refresh time, in seconds. The default value is 0, which means each time identifying immediate refresh. This parameter is used to reduce transfer data amount of tag identification information. (Not supported)
3	invtSess	Integer	1B	[0,3]	EPC session number is used in the inventory
4	invtQMax	Integer	1B	[0, 15]	Maximum inventory Q. 0 turn off automatic inventory, >0 start automatic inventory and as the maximum value of Q
5	invtQMin	Integer	1B	[0, 4]	Instant Inventory (or automatic inventory minimum) value of Q
6	invtQMinTries	Integer	1B	[0, 15]	At the end stage of an inventory process, in order to dish out the missing tag as much as possible, the number of times the Q value set by repeatedly invtQMin reforming inventory
7	invtIdleMin	Integer	1B	[0, 255]	Minimum idle time between two inventory process, ms, in order to cool the PA.
8	invtIdleFract	Integer	1B	[1, 255]	Inventory of idle time for inventory work time points, PA for cool to prevent overheating. Such as: 8 indicates that the working time and the idle time ratio is 8:1.
21	rfAntMask	Bitfield	1B		Antenna channel selection bit field, each representing an

					antenna. Set 1 means that the antenna is enabled, 0 means the antenna is not enabled. Such as 0x05 means zeroth, 2nd antenna is enabled
22	rfGain	Integer	1B	[0,31]	RF gain level
23	rfCenter	Integer	4B	>0	RF carrier center frequency, KHz
24	rfDeviation	Integer	2B	>0	Floating carrier frequency value, KHz
25	rfStep	Integer	2B	>0	Carrier frequency adjustment step, KHz
253	beepEnable	Integer	1B	{0, 1}	Enable or disable beep
255	linkHoldSec	Integer	1B	>=0	Link detection interval (seconds) 0 indicates no detection (not supported)

4.3 tagID Internal Format

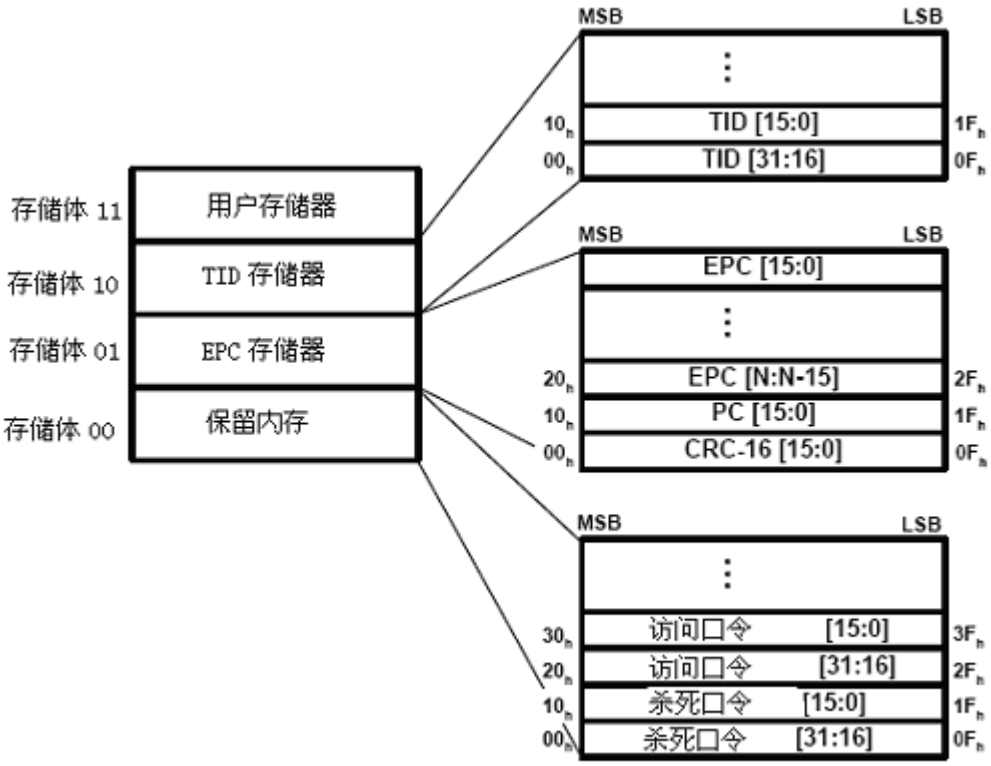
No.	**Field Tags**	**Field Type**	**Code length**	**Field range**	**Field Description**
1	STAT	Integer	1 B		Bit 0 - 5: tagID number of bytes; Bit 6: Status field select flag (1 means enabled); Bit 7: Reserved, set to 0
2	tagID	Integer	2 B		PC + XPC + EPC. PC: PC field which described in EPC standards XPC: XPC field which described in EPC standard. When the PC XPC flag is set, this field length is 2B, conversely 0B; EPC: EPC code
3	status	Bitfield	1B		State for tag identification (optional). Bit0-2: Identification Antenna No. Bit3-7: Reserved (set to 0)

4.4 EBV-encoded Integer

EBV is a data structure which can express the extension data. In each byte,

the l~7 bits store valid data, the high bit is used as an indicator of extension of the instructions. 1 means there are follow byte, 0 means that the byte is the end of EBV.

4.5 Tag Storage Structure



4.6 Parameter Format of lckData

Operating masks									
Kill password		**Access password**		**UII memory**		**TID memory**		**User memory**	
19	18	17	16	15	14	13	12	11	10
Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write
9	8	7	6	5	4	3	2	1	0
Pwd read/ write	Perma lock	Pwd read/ write	Perma lock	Pwd write	Perma lock	Pwd write	Perma lock	Pwd write	Perma lock
Operation code									
Pwd-write		**Permalock**		**Description**					

0	0	The corresponding data segment can be written in OPEN or SECURED state
0	1	In the OPEN or SECURED state, the corresponding data segment can be permanently written, and the corresponding data segment can not be locked.
1	0	The corresponding data segment can be written in SECURED state, OPEN state can not be written
1	1	The corresponding data segment is not written in any state
Pwd-read/write	**Permalock**	**Description**
0	0	The corresponding data segment can be read and written in OPEN or SECURED state
0	1	The corresponding data segment can be read and written in OPEN or SECURED state, the corresponding data segment can not be locked.
1	0	The corresponding data segment can be read and written in SECURED state, the OPEN state can not read and write.
1	1	The corresponding data segment is not read and write in any state

4.7 CRC Verification And Calculation Method

```
unsigned short crc16_calc( unsigned char* data, int len)
{
   int i,j;
   unsigned char    n_tmp;
   unsigned short crcBase = 0xffff;
   for( i = 0; i < len; i++)
   {
       n_tmp = *data++;
      for (j = 0;j < 8;j++)
      {
         if((crcBase & 0x8000) ^ ((n_tmp & 0x80) << 8))
            crcBase = ( (crcBase << 1) & 0xffff) ^ 0x1021;
         else
            crcBase = (crcBase << 1) & 0xffff;
         n_tmp <<= 1;
      }
   }
```

```
    return ~crcBase;
}
```