SKYETEK™ PROTOCOL V2 GUIDE

VERSION 100112

COPYRIGHT INFORMATION:

Version 100112

TECHNICAL SUPPORT AND CONTACT INFORMATION

SkyeTek, Inc.

1525 Market Street. Suite 200

Denver, CO 80202

http://www.skyetek.com

SALES:

sales@skyetek.com

TECHNICAL SUPPORT:

support@skyetek.com

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 About this Document

This document serves to show how the Skyetek Protocol V2 (STPV2) can be used to design RFID applications using the M1 and M1 mini skyemodules. Familiarity with this protocol will allow the user to communicate with a variety of tags without detailed knowledge of each tags operation and encryption algorithyms.

1.1 Revision History

Revision	Author	Change
100112	Ryan Smith	Updated document formatting.

Table 1-1: Revision History

2 Dictionary of Terms

3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
API	Application Programming Interface
DES	Data Encryption Standard
GPIO	General Purpose Input/Output
HID	Human Interface Device
HMAC	Hash-based message authentication code
I^2C	Inter-integrated Circuit
LSB	Least Significant Bit
MD5	Message-Digest Algorithm
MSB	Most Significant Bit
NC	No Connect
PRNG	Pseudo-Random Number Generator
RoHS	Reduction of Hazardous Substances
SHA	Secure Hash Algorithm
SPI	Serial Programming Interface
SSEL	Slave Select
STP V3	SkyeTek Protocol Version 3
TTL	Transistor-transistor Logic

3 Protocol Description

The SkyeTek Protocol defines the data exchange between a host controller and a SkyeTek RFID radio module. It specifies how a host controller can address, configure and command a radio module in order to read and write to RFID tags and smart labels. This document describes the format of messages exchanged between a SkyeTek radio module and a host. The content of these messages is dependent on the type of radio module, and this document must be used in conjunction with the Reference Guide of the specific radio module.

The SkyeTek Protocol specifies only the data-communication layer. The hardware-interface layer is a function of the hardware settings and method of connection of the radio module. See the individual product's Reference Guide for hardware interface specifications.

The SkyeTek Protocol supports ASCII and Binary formats. The host initiates every request or response sequence, and determines which format is used (ASCII or Binary).

Figure 3-1: ASCII Request and Response

- CR is the ASCII code for a Carriage Return (CR = CHR$(13) = 0x0D [the ENTER key]).
- LF is the ASCII code for a Line Feed (LF = CHR$(10) = 0x0A).
- To allow a simple terminal program to communicate with any radio module, the ASCII format transmits each byte of request and response data as two ASCII characters (corresponding to the HEX values of the upper and lower nibbles of each byte, respectively).
- For example, if the single-byte FLAGS field within a request has the hex value 0x1A then the host sends two characters:
  ASCII '1' followed by an ASCII 'A'.

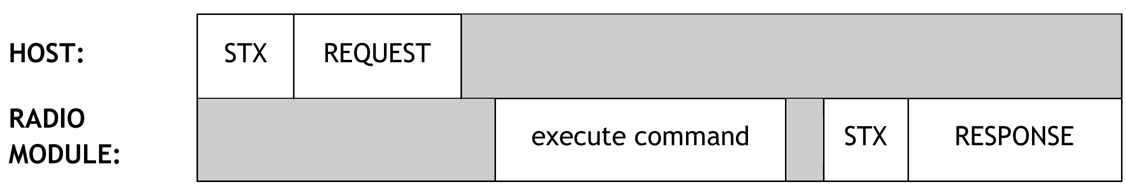

Figure 3-2: Binary Request and Response

- STX is the ASCII code for a Start of Transmission (STX = CHR$(2) = 0x02).
- For Binary format, each byte must be sent within 10ms of the preceding byte such that a 10ms delay marks the end of transmission.

4 Description of the Request Fields

Mandatory Fields

Optional Fields

FLAGS	COMMAND	RID	TAG TYPE	TID	AFI	STARTING BLOCK	NUMBER OF BLOCKS	DATA	CRC
8 bits	8 bits	8 bits	8 bits	64 bits	8 bits	8 bits	8 bits	n*8 bits	16 bits

Figure 4-1: ASCII Request Message

request										
MSG LEN	FLAGS	COMMAND	RID	TAG TYPE	TID	AFI	STARTING BLOCK	NUMBER OF BLOCKS	DATA	CRC
8 bits	8 bits	8 bits	8 bits	8 bits	64 bits	8 bits	8 bits	8 bits	n*8 bits	16 bits

Figure 4-2: Binary Request Message

4.1 MSG LEN Field (Required for Binary)

The MSG LEN field specifies how many bytes are in the host request, not including the <STX> and not including the MSG LEN field.

For example: <STX> 0x05 0x20 0x14 0x00 0xF1CA

4.2 FLAGS Field (Required)

The bits of the FLAGS field specify protocol options and command options.

FLAGS (LSByte)							
RID_F	TID_F	CRC_F	AFI_F	RF_F	LOCK_F	INV_F	LOOP_F
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Figure 4-3: FLAGS Field

RID_F = 1 The RID field is present in the request of any command.
RID_F = 0 The RID field is not present in the request.

TID_F = 1 The TID field is present in the request of any tag command.
TID_F = 0 The TID field is not present in the request.

CRC_F = 1 The CRC field is present in the request of any command.
CRC_F = 0 The CRC field is not present in the request.

AFI_F = 1 The AFI field is present in the request (valid only for the SELECT_TAG command).
AFI_F = 0 The AFI field is not present in the request.

RF_F = 1 The RF transmitter stays on after the command has been executed.
RF_F = 0 Turns off the RF transmitter after the command has been executed.

LOCK_F = 1 Locks the tag block(s). Use a WRITE_TAG command with DATA field not present.
LOCK_F = 0 All other commands.

INV_F = 1 Selects the tag command to inventory all tags in the field, outputs all Tag IDs detected.
INV_F = 0 Selects the tag command to select only one tag, outputs only first Tag ID detected.

LOOP_F = 1 Repeats the tag command. Gives a response for every pass.
LOOP_F = 0 Single execution of the tag command results in a one time response is pass or fail.

4.3 COMMAND Field (Required)

The COMMAND field specifies the command *type* and the command *target* of a host request.

COMMAND Type				COMMAND Target			
reserved (set to 0)	Write_Bit	Read_Bit	Sel_Bit	reserved (set to 0)	Tag_Bit	Sys_Bit	Mem_Bit
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Figure 4-4: COMMAND Field

Each command specifies exactly one *type* and exactly one *target*. The value of the COMMAND field in a request has exactly two bits set—one bit in the upper nibble, and one bit in the lower nibble. The following table shows the list of commands that the SkyeTek Protocol supports.

COMMAND	
Value	Name
0x14	SELECT_TAG
0x21	READ_MEM

0x22	READ_SYS
0x24	READ_TAG
0x41	WRITE_MEM
0x42	WRITE_SYS
0x44	WRITE_TAG

4.4 Tag Type Field (Required for Tag Command)

The optional TAG TYPE field is present in a request when the command is a Tag Command (for example, Tag_Bit is 1 in the COMMAND field). The TAG TYPE field specifies the RFID tag type with which the radio module attempts to communicate.

4.5 RID Field

RID is the Reader ID field. A host controller can specify to which radio module a request is directed. The optional RID field is present in a request when RID_F=1.

4.6 TID Field

TID stands for Tag ID Field. A host controller can specify to which tag a request is directed. The optional TID field is present in a request when TID_F=1. The TID_F is only valid in a Tag Command request.

4.7 AFI Field

AFI stands for Application Field Identifier. This field can be present only in a SELECT_TAG command request. ICode1 and ISO15693 tag types support AFI. When a request contains AFI, only the tags with matching AFI answer to the request.

4.8 STARTING BLOCK Field

The STARTING BLOCK field specifies the starting tag block number to read or write.

The STARTING BLOCK field is not present in a request containing a SELECT command.

A BLOCK of data corresponds to 1 byte when reading or writing to the radio module memory. A BLOCK of data, when reading or writing to the tag memory, varies depending on the tag type.

4.9 NUMBER OF BLOCKS Field

The NUMBER OF BLOCKS field specifies the number of tag blocks to read or write.

The NUMBER OF BLOCKS field is not present in a request containing a SELECT tag command. The NUMBER OF BLOCKS is always 0x01 for a system command.

4.10 DATA Field

The DATA field contains the values to write during a WRITE command. The optional DATA field is not present in the request of a SELECT command.

4.11 CRC Field

The CRC field is optional (present if CRC_F is 1) when the format is ASCII mode. The CRC field is required (CRC_F must be set to 1) when the format is Binary mode. The CRC field is a two-byte field calculated by the host controller according to the REQUEST fields present. The CRC is calculated over all the fields in the request or response not including the initiator and terminator bytes (STX, CR, LF, etc.).

The CRC_CCITT polynomial, $x^{16} + x^{12} + x^5 + 1$ = 0x8408, is used in a forward CRC calculation with a preset value of 0x0000.

```
// *dataP is a pointer to the byte array over which the CRC is to be calculated
// n is the number of bytes in the array pointed to by *dataP
//
unsigned int crc16( unsigned char * dataP, unsigned char n )
{
      unsigned char i, j;                    // byte counter, bit counter
      unsigned int crc_16;                   // calculation
      crc_16 = 0x0000;                       // PRESET value
      for (i = 0; i < n; i++)                // check each byte in the array
      {
             crc_16 '= *dataP++;             //
             for (j = 0; j < 8; j++)         // test each bit in the byte
             {
                    if(crc_16 & 0x0001 )     //
                    {
                           crc_16 >>= 1;
                           crc_16 '= 0x8408;  // POLYNOMIAL x'16 + x'12 + x'5 + 1
                    }
                    else
                    {
                           crc_16 >>= 1;
                    }
             }
      }
      return( crc_16 );                      // returns calculated crc (16 bits)
}
```

Figure 4-5: C Language Implementation of the CRC Calculation

CRC Calculation Example:

A simple host request (ASCII mode) to get the Tag ID of an ISO15693 tag is <CR><001401><CR>, where FLAGS=0x00, COMMAND=0x14, TAG TYPE=0x01. (without CRC).

If CRC_F is set in the FLAGS field, then the request becomes <CR><201401E043><CR>, where FLAGS=0x20, COMMAND=0x14, TAG TYPE=0x01, CRC=0xE043.

A possible radio module response is <LF><14E00700000147637A1AA2><CR><LF>, where RESPONSE CODE=0x14, RESPONSE DATA=0xE00700000147637A, CRC=0x1AA2.

5 Description of the RESPONSE Fields

RESPONSE CODE	RID	TAG TYPE	RESPONSE DATA	CRC
8 bits	8 bits	8 bits	n * 8 bits	16 bits

Figure 5-1: ASCII Response Message

RESPONSE					
MSG LEN	RESPONSE CODE	RID	TAG TYPE	RESPONSE DATA	CRC
8 bits	8 bits	8 bits	8 bits	n * 8 bits	16 bits

Figure 5-2: Binary Response Message

Required fields: RESPONSE CODE

All other fields are optional, and their use depends on the value of the RESPONSE CODE field.

5.1 MSG LEN Field (Required for Binary)

The MSG LEN field specifies how many bytes are in the response, not including the MSG LEN byte and not including the <STX> byte. This field is *always* present in binary mode. This field is *never* present in ASCII mode. See the example in the section 0.

5.2 RESPONSE CODE Field (Required)

RESPONSE CODES	
RESPONSE CODE	Description
0x14	SELECT _TAG pass
0x1C	SELECT _TAG LOOP activate
0x94	SELECT _TAG fail
0x9C	SELECT _TAG LOOP cancel
0x21	READ_MEM pass
0x22	READ_SYS pass
0x24	READ_TAG pass
0XA1	READ_MEM fail
0xA2	READ_SYS fail
0xA4	READ_TAG fail
0x32	EVENT_report
0xC2	EVENT_error
0x41	WRITE_MEM pass
0x42	WRITE_SYS pass
0x44	WRITE_TAG pass
0xC1	WRITE_MEM fail
0xC2	WRITE_SYS fail
0xC4	WRITE_TAG fail
0x80	Non ASCII character in request
0x81	BAD CRC
0x82	FLAGS do not match COMMAND
0x83	FLAGS do not match TAG TYPE
0x84	Unknown COMMAND
0x85	Unknown TAG TYPE
0x86	Invalid STARTING BLOCK
0x87	Invalid NUMBER OF BLOCKS
0x88	Invalid Message Length

Figure 5-3: RESPONSE CODE List

5.3 RID Field

The RID field is only present in a response when the request has FRID_Fis 1.

5.4 TAG TYPE Field

The TAG TYPE field is only present in a response when the request is SELECT_TAG with TAG_TYPE is AUTO. See Figure 7 for the list of TAG TYPE codes.

5.5 RESPONSE DATA Field

The RESPONSE DATA field is present in a response when the response contains data requested by the host such as a TagID, Tag Block data, or system parameter data.

5.6 CRC Field

The CRC field is present in the response to any request in which the CRC_F is set. The CRC calculation for a response is the same as for a request.

Previously in this document, a CRC calculation example presented the request field description where the radio module response using CRC was <LF><14E00700000147637A**1AA2**><CR><LF>

The host can verify the response two different ways.

5.6.1 Matching the CRC

The host can verify the response by calculating the CRC over 0x14E00700000147637A to get 0x**1AA2**.

5.6.2 Calculating 0x0000 Using the RESPONSE CRC With LSByte First

The host can verify the response by calculating the CRC over 0x14E00700000147637AA21A to get 0x0000. (The CRC, which is sent MSByte in the response from the radio module, is first used in the calculation LSByte).

6 Description of the Commands

6.1 Description of the System Commands (Sys_Bit = 1 in COMMAND Field)

The system commands that allow reading and writing of system parameters are internal to a radio module. System parameters govern certain operating characteristics of a radio module. See the Reference Guide of the individual product for system parameters that are supported.

6.1.1 READ_SYS

The host controller reads data from the system parameters of a radio module.

READ							
MSG LEN	FLAGS	COMMAND	RID	STARTING BLOCK	NUMBER OF BLOCKS	DATA	CRC
8 bits	8 bits	0x22	8 bits	8 bits	0x01	n*8 bits	16 bits

Figure 6-1: Request for READ_SYS

The STARTING BLOCK field specifies the starting system parameters to read.
The NUMBER OF BLOCKS field specifies the number of system parameters to read.
The DATA field is only present during special system commands.

FLAGS							
RID_F		CRC_F					
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Figure 6-2: FLAGS in Request for READ_SYS

When RID_F is 1, the optional RID field is included in the request to specify a radio module.
When CRC_F is 1, the optional CRC field is included in the request.

Response				
MSG LEN	RESPONSE CODE	RID	RESPONSE DATA	CRC
8 bits	8 bits	8 bits	n * 8 bits	16 bits

Figure 6-3: Response to READ_SYS

If the RESPONSE CODE field indicates a successful READ operation (RESPONSE CODE is 0x22), then the response includes the RESPONSE DATA field containing the NUMBER OF BLOCKS bytes that were successfully read from the system parameters.
If the RESPONSE CODE field indicates an error condition, then the RESPONSE DATA field is not present in the response.
If the CRC_F is set in the request, then the CRC field is included in the response.
If the RID_F is set in the request, then the RID field is included in the response.

6.1.2 WRITE_SYS

The host controller writes data to the system parameters of a radio module.

WRITE_SYS							
MSG LEN	FLAGS	COMMAND	RID	STARTING BLOCK	NUMBER OF BLOCKS	DATA	CRC
8 bits	8 bits	0x42	8 bits	8 bits	0x01	n * 8 bits	16 bits

Figure 6-4: Request for Write_SYS

The STARTING BLOCK field specifies the starting system parameter to write.

The NUMBER OF BLOCKS field specifies the number of system parameters to write.

The DATA field specifies the system parameter value(s) to write. The DATA field must contain the number of bytes specified in the NUMBER OF BLOCKS field.

The CRC field is required for all WRITE_SYS commands.

FLAGS							
RID_F		CRC_F					
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Figure 6-5: FLAGS in Request for WRITE_SYS

When RID_F is 1, the optional RID field is included in the request to specify a radio module.

It is required that CRC_F is 1. The CRC field is included in each request of a WRITE_SYS command.

Response			
MSG LEN	RESPONSE CODE	RID	CRC
8 bits	8 bits	8 bits	16 bits

Figure 6-6: Response to WRITE_SYS

The RESPONSE DATA field is not present in the response.

Because the CRC is mandatory in the request, it is always included in the response.

If the RID_F is set in the request, then the RID field is included in the response.

6.2 Description of the Memory Commands (Mem_Bit = 1 in COMMAND Field)

The memory commands access standard and extended memory areas of radio modules.

The optional memory areas and features are unique to each product model. See the Reference Guide of the individual product for memory features and memory commands.

System Parameters				
Address	Description	Supported Commands		
		SELECT	READ	WRITE
0x00-0x7F	System Parameters (EEPROM and RAM)		√	√

Figure 6-7: Memory Space of Radio Modules

6.2.1 READ_MEM

The host controller reads data from the memory of a radio module.

READ_MEM						
MSG LEN	FLAGS	COMMAND	RID	STARTING BLOCK	NUMBER OF BLOCKS	CRC
8 bits	8 bits	0x21	8 bits	8 bits	0x01	16 bits

Figure 6-8: Request for READ_MEM

The STARTING BLOCK field specifies the starting memory address to read.

The NUMBER OF BLOCKS field specifies the number of memory addresses to read.

FLAGS							
RID_F		CRC_F					
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Figure 6-9: FLAGS in Request for READ_MEM

When RID_F is 1, the optional RID field is included in the request to specify a radio module.

When CRC_F is 1, in the FLAGS field of a request the optional CRC field is present.

Response				
MSG LEN	RESPONSE CODE	RID	RESPONSE DATA	CRC
8 bits	8 bits	8 bits	n * 8 bits	16 bits

Figure 6-10: RESPONSE to READ_MEM

If the RESPONSE CODE field indicates a successful READ operation (RESPONSE CODE is 0x21), then the response includes the RESPONSE DATA field containing the bytes that were successfully read from the memory address or addresses.

If the RESPONSE field indicates an error condition, then the RESPONSE DATA field is not present in the response.

If the CRC_F is set in the request, then the CRC field is included in the response.

If the RID_F is set in the request, then the RID field is included in the response.

6.2.2 WRITE_MEM

The host controller writes data to the EEPROM of a radio module.

WRITE_MEM							
MSG LEN	FLAGS	COMMAND	RID	STARTING BLOCK	NUMBER OF BLOCKS	DATA	CRC
8 bits	8 bits	0x41	8 bits	8 bits	8 bits	n * 8 bits	16 bits

Figure 6-11: Request for WRITE_MEM

The STARTING BLOCK field specifies the starting memory address to write.

The NUMBER OF BLOCKS field specifies the number of memory addresses to write.

The DATA field specifies the memory value to write. The DATA field must contain the number of bytes specified in the NUMBER OF BLOCKS field.

The CRC field is required for all WRITE_MEM commands.

FLAGS							
RID_F		CRC_F					
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Figure 6-12FLAGS in Request for WRITE_MEM

When RID_F is 1, the optional RID field is included in the request to specify a radio module.

It is required that CRC_F is 1, which ensures that the CRC field is included in each request of a WRITE_MEM command.

Response			
MSG LEN	RESPONSE CODE	RID	CRC
8 bits	8 bits	8 bits	16 bits

Figure 6-13: RESPONSE to WRITE_MEM

The RESPONSE DATA field is not present in the response.

Because the CRC is mandatory in the request, it is always included in the response.

If the RID_F is set in the request, then the RID field is included in the response.

6.3 Description of the Tag Commands (Tag_Bit = 1 in COMMAND Field)

The tag commands operate on the tag (or tags) of type TAG TYPE which are present in the RF field of the radio module.

6.3.1 SELECT_TAG

The SELECT_TAG command can be used several different ways, depending on the FLAGS field bits.

SELECT_TAG							
MSG LEN	FLAGS	COMMAND	RID	TAG TYPE	TID	AFI	CRC
8 bits	8 bits	0x14	8 bits	8 bits	64 bits	8 bits	16 bits

Figure 6-14: Request for SELECT_TAG

FLAGS							
RID_F	TID_F	CRC_F	AFI_F	RF_F		INV_F	LOOP_F
bit 7	bit 6	bit 5	bit 4	bit 3		bit 1	bit 0

Figure 6-15: FLAGS in Request for SELECT_TAG

Response					
MSG LEN	RESPONSE CODE	RID	TAG TYPE	RESPONSE DATA	CRC
8 bits	8 bits	8 bits	8 bits	n * 8 bits	16 bits

Figure 6-16: Response to SELECT_TAG

If the RESPONSE CODE field indicates a successful READ_TAG operation then the

RESPONSE DATA field is present and contains the unique TID of the tag. The size of the TID depends on the TAG TYPE and version.

If the TAG TYPE is AUTO in the request, then the TAG TYPE field is present in the response.

If the RID_F is set in the request, then the RID field is included in the response.

6.3.2 WRITE_TAG

The WRITE_TAG command writes data to a tag.

Request									
MSG LEN	FLAGS	COMMAND	RID	TAG TYPE	TID	STARTING BLOCK	NUMBER OF BLOCKS	DATA	CRC
8 bits	8 bits	0x44	8 bits	8 bits	64 bits	8 bits	8 bits	n*8 bits	16 bits

Figure 6-17: Request for WRITE_TAG

FLAGS							
RID_F	TID_F	CRC_F		RF_F	LOCK_F		
bit 7	bit 6	bit 5		bit 3	bit 2		

Figure 6-18: FLAGS in Request for WRITE_TAG

When TID_F is 1, the optional TID field is included in the request to specify the tag to write.

When LOCK_F is 1, the DATA field is not present in the request

The STARTING BLOCK field specifies the least significant block of tag memory to write.

The NUMBER OF BLOCKS field specifies the number of tag blocks to write. The DATA field specifies the tag block value(s) to write.

When NUMBER OF BLOCKS field is greater than one, the DATA field is organized with lowest block first, followed by next higher block, and so on.

Response			
MSG LEN	RESPONSE CODE	RID	CRC
8 bits	8 bits	8 bits	16 bits

Figure 6-19: Response to WRITE_TAG

The RESPONSE DATA field is not present in the RESPONSE of a WRITE_TAG command.

If the RID_F is set in the request, then the RID field is included in the response.

6.3.3 READ_TAG

The host reads data from the tag(s) in proximity of the RF field of a radio module.

READ_TAG								
MSG LEN	FLAGS	COMMAND	RID	TAG TYPE	TID	STARTING BLOCK	NUMBER OF BLOCKS	CRC
8 bits	8 bits	0x24	8 bits	8 bits	64 bits	8 bits	8 bits	16 bits

Figure 6-20: Request for READ_TAG

FLAGS							
RID_F	TID_F	CRC_F		RF_F		INV_F	
bit 7	bit 6	bit 5		bit 3		bit 1	

Figure 6-21: FLAGS in request for READ_TAG

When RID_F is 1, the optional RID field is included in the request to specify the radio module.

When TID_F is 1, the optional TID field is included in the request to specify the tag to read.

When CRC_F is 1, the CRC field is included in the request.

When RF_F is 1, the RF transmitter remains active after execution of the Read command.

INV_F is 1 the radio module performs an "inventory read" of the specified blocks of all tags in the field.

RESPONSE				
MSG LEN	RESPONSE CODE	RID	RESPONSE DATA	CRC
8 bits	8 bits	8 bits	N * 8 bits	16 bits

Figure 6-22: RESPONSE to READ_TAG

If the RESPONSE CODE field indicates a successful READ_TAG operation (RESPONSE CODE is 0x24) then the response message includes the RESPONSE DATA field containing the NUMBER OF BLOCKS that were successfully read from the tag. The size of each block depends on the TAG TYPE and version.

If the RESPONSE CODE field indicates an error condition, the RESPONSE DATA field is not present in the response.

If the RID_F is set in the request, then the RID field is included in the response.