

Definition of Terms:

1. **SOM** Start of Message: **0AH (10d)**
2. **EOM** End of Message : **0DH (13d)**
3. **SRC** Node Address of **Message Sender**
4. **DST** Serial Number of **Message Receiver**
5. **TYP** Type of Message
6. **CRC** 16 bit Cyclic Redundancy Check

Notes:

1. All Packet data is transmitted as ASCII Hex except SOM & EOM
2. **0000000h** is a Group Address for all Readers (Broadcast)
02h is the Host Node Address (Reserved)
 Reader Node address range

- v1.02, Sept 5, 2006
 - (1) Change Typ 13, Config 2, Bit 5 to Buzz for Latched Relay Activation
 - (2) Change Typ 13, Config 2, Bit 4 to Buzz for Timed Relay Activation
 - (3) Change Typ 13, Config 2, Bit 3 to LED for Latched Relay Activation
 - (4) Change Typ 13, Config 2, Bit 2 to LED for Timed Relay Activation
 - (5) Add Typ 00 comment re: Latch Relay
 - (6) Change Typ 82 response description: Send Firmware Version Number
- v1.03, Sept 11, 2006
 - (1) Change Typ 00, revert back to the Rk-XRM command format
- v1.04, Sept 14, 2006
 - (1) Change Typ 00, Port # 1 = Relay, 2 = Additional output # 1, 3 = Additional output # 2
- v1.04, Sept 15, 2006
 - (1) Change Typ 13, Port # 1 = Relay, 2 = Additional output # 1, 3 = Additional output # 2
- v1.05, Nov 13, 2006
 - (1) Add Typ 04, set card read behavior
- v1.06, Dec 21, 2006
 - (1) Change Typ 81, bit map the input status
- v1.07, Apr 19, 2007
 - (1) Change Typ 81, add bit mapped changed input status
- v1.09, Nov 07, 2007
 - (1) Change Typ 87, Tag is still in the field
 - (2) Add Typ 88, Most recent tag read is no longer in the field
- v1.10, Mar 19, 2008
 - (1) Add Typ 0A, Get Blue and Yellow Input Voltage Readings.
 - (2) Add Typ 8A, response to Typ 0A request.
 - (3) Add Typ 09, Get Settings
 - (4) Add Typ 89, response to Typ 09 request.
 - (5) Add Typ 06, Get Model Request.
 - (6) Add Typ 86, Model Response Message.
 - (7) Change Typ 03, add Enable Class B Supervision.
 - (8) Add Typ 84, Class B Supervision Alert Message.

Packet Types:

Host -> Reader: Timed Operate Relay

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	D-TIME	A-TIME	DTX	ATX	Port #	MN	MN	CRC	CRC	CRC	CRC	CRC	CRC	EOM	
								0	2	0	0	Delay Time	Activation Oj	Multi	Multi	Output Port	Msg #									

Note: Src is 02 only.
 Note: If Activation Time = FF and ATX = F then the Port(s) is latched.
 Note***: 1=relay, 2=output #1, 4=output #2, i.e. 6=outputs #1 and #2, 7=all outputs

Host -> Reader: LED Background State

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20					
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	STE	STE	MN	MN	CRC	CRC	CRC	CRC	EOM					
								0	2	0	1	LED State													

Note: LED State values: Off=0, Green=1, Red=2, Amber=3
 Note: Src is 02 only.

Host -> Reader: LED Foreground State

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	STE	STE	TIME	TIME	TIME	TIME	MN	MN	CRC	CRC	CRC	CRC	EOM	
								0	2	0	2	LED State		LED Time											

Note: Src is 02 only.
 Note: LED State values: Off=0, Green=1, Red=2, Amber=3
 Buzzer: To hear an audible tone at the same time, set bit 4 of the LED State field

Host -> Reader: Set Mode

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20					
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	Mo1	Mo1	MN	MN	CRC	CRC	CRC	CRC	EOM					
								0	2	0	3	Mode 1													

Note: Src is 02 only.

Mo1							
Uid/Enc	RFU	I4Mon	I3Mon	RFU	RFU	RFU	RFU
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- Uid/Enc 0 = Auto Read Encrypted Cards
 1 = Auto Read UID Cards
- I4Mon Blue
 0 = Input 4 Class B Supervision Disablec
 1 = Input 4 Class B Supervision Enabled, see message Type 84
- I3Mon Yellow
 0 = Input 3 Class B Supervision Disablec
 1 = Input 3 Class B Supervision Enabled, see message Type 84

Host -> Reader: Set Card Read Behavior

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	CF1	CF1	CF2	CF2	CF3	CF3	CF4	CF4	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	0	4	Config 1	Config 1	Config 2	Config 2	Config 3	Config 3	Config 4	Config 4	Msg #						

Note: Src is 02 only.

CF1

ARUCR	AO1UCR	AO2UCR	RFU	RFU	RFU	Color (2 bits)	
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Activate Relay Upon Card-Read (ARUCR)
 0 = Disable Activate Relay Upon Card Reac
 1 = Enable Activate Relay Upon Card Reac

Activate Output 1 Upon Card-Read (AO1UCR)
 0 = Disable Activate Output1 Upon Card Reac
 1 = Enable Activate Output1 Upon Card Reac

Activate Output 2 Upon Card-Read (AO2UCR)
 0 = Disable Activate Output2 Upon Card Reac
 1 = Enable Activate Output2 Upon Card Reac

COLOR 0 = LED Flash Color None
 1 = LED Flash Green
 2 = LED Flash Red
 3 = LED Flash Amber

CF2

Activation On-Time, Relay

Relay Time (8 bits)						255 seconds	
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Note: a value of 0 activates the output for one quarter second.

CF3

Activation On-Time, Output 1

Output 1 On-Time (8 bits)						255 seconds	
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Note: a value of 0 activates the output for one quarter second.

CF4

Activation On-Time, Output 2

Output 2 On-Time (8 bits)						255 seconds	
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Note: a value of 0 activates the output for one quarter second.

Host => Reader: Get Serial info

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	0	5	Msg #						

Note: Response is message 82

Host => Reader: Get Model Number

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	0	6	Msg #						

Note: Response is message 82

Host => Reader: Message Acknowledge

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	0	7	Msg #						

Note: Msg # in Ack = message # for which ack applies.

Host => Reader: Get Settings

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	0	9	Msg #						

Note: Response is message 89

Host => Reader: Get Voltage Reading on the Blue and Yellow Wires

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	0	A	Msg #						

Note: Response is message 8A

Host => Reader: Send Tag, Send Host Last presented card TAG I.D.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	0	D	Msg #						

Note: Send last Tag Read

Host => Node: Change Baud Rate

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	BR	BR	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	0	E	Baud Rate	Msg #							

Baud Rate	Configuration Value
9600	09
14400	14
19200	19
28800	28
38400	38
57600	57
####	15

Host -> Reader: Set Configuration

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	IYP	IYP	CF1	CF1	CF2	CF2	CF3	CF3	CF4	CF4	CF5	CF5	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	1	3	Config 1	Config 2	Config 3	Config 4	Config 5	Msg #											

CF1

I4OR	I3OR	I2OR	I1OR	I4AR	I3AR	I2AR	I1AR
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- I4OR Blue
0 = Input 4 Not Operate Relay
1 = Input 4 Operate Relay
- I3OR Yellow
0 = Input 3 Not Operate Relay
1 = Input 3 Operate Relay
- I2OR Brown
0 = Input 2 Not Operate Output #2
1 = Input 2 Operate Output #2
- I1OR Orange
0 = Input 1 Not Operate Output #1
1 = Input 1 Operate Output #1
- I4AR: Blue
0 = Input 4 Not Auto-Report Change
1 = Input 4 Auto-Report Change
- I3AR: Yellow
0 = Input 3 Not Auto-Report Change
1 = Input 3 Auto-Report Change
- I2AR Brown
0 = Input 2 Not Auto-Report Change
1 = Input 2 Auto-Report Change
- I1AR Orange
0 = Input 1 Not Auto-Report Change
1 = Input 1 Auto-Report Change

CF2

RLYPOL	SecurRelay	Buz Latch	Buz Timed	Led Latch	Led Timed	Color (2 bits)	
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- RLYPOL 0 = Relay Normally Open
1 = Relay Normally Closed
- SecurRelay 0 = Disable SecurRelay
1 = Enable SecurRelay
- BUZZ for Latched Relay Activation
0 = No Buzzer during Latched Relay Activator
1 = Buzz during Latched Relay Activator
- BUZZ for Timed Relay Activation
0 = No Buzzer during Timed Relay Activator
1 = Buzz during Timed Relay Activator
- LED for Latched Relay Activation
0 = LED Foreground Not Triggered by Latched Relay Activator
1 = LED Foreground Triggered by Latched Relay Activator
- LED for Timed Relay Activation
0 = LED Foreground Not Triggered by Timed Relay Activator
1 = LED Foreground Triggered by Timed Relay Activator
- COLOR 0 = Foreground Color None
1 = Foreground Green
2 = Foreground Red
3 = Foreground Amber

CF3

Led Flash Duration in seconds

Led On-Time (8 bits)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

CF4

Activation On-Time, Relay

Time (8 bits) 255 seconds							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

CF5

Activation On-Time, Output #1 & #2

Time (8 bits) 255 seconds							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Host -> Reader: ISO 15693 Read Block

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	Opt	Opt	Len	Len	Flg	Flg	Cmd	Cmd	Blk	Blk	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	1	5	0	0	0	3	0	3	2	0	0	0	0	Msg #					

Opt

Bit(s)	Description
0	(ISO 15693) 0 = Request
1	(ISO 15693) 0 = Read; 1 = Write
2	(ISO 15693) 0 = Write TI; 1 = Write Philips
3	Spare - Don't care
4	Spare - Don't care
5	Spare - Don't care
6	Spare - Don't care
7	Spare - Don't care

Flg

Bit(s)	Description
0	Sub-carrier-flag 1 = FSK; 0 = ASK
1	Data-rate-flag 0 = Slow, 1 = Fast
2	Spare - Don't care
3	Spare - Don't care
4	Select Flag
5	Tag address flag
6	Option Flag
7	Spare - Don't care

Host -> Reader: ISO 15693 Write Block

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	Opt	Opt	Len	Len	Flg	Flg	Cmd	Cmd	Blk	Blk	Data	Data	Data	Data	Data	Data	Data	Data
								0	2	1	5	0	2	0	7	4	3	2	1	0	0								

Opt

Bit(s)	Description
0	(ISO 15693) 0 = Request
1	(ISO 15693) 0 = Read; 1 = Write
2	(ISO 15693) 0 = Write TI; 1 = Write Philips
3	Spare - Don't care
4	Spare - Don't care
5	Spare - Don't care
6	Spare - Don't care
7	Spare - Don't care

30	31	32	33	34	35	36
MN	MN	CRC	CRC	CRC	CRC	EOM
Msg #						

Flg

Bit(s)	Description
0	Sub-carrier-flag 0 = FSK; 1 = ASK
1	Data-rate-flag 0 = Slow, 1 = Fast
2	Spare - Don't care
3	Spare - Don't care
4	Select Flag
5	Tag address flag
6	Option Flag
7	Spare - Don't care

Host => Reader: Reset the unit, i.e. Firmware Upgrade

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SOM	DST	DST	DST	DST	DST	DST	DST	SRC	SRC	TYP	TYP	MN	MN	CRC	CRC	CRC	CRC	EOM
								0	2	2	1	Msg #						

Reader To Host Messages

Reader -> Host: Tag Read

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	41	42	43	44	45	46	47	48	49	50	
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	D31	D30	D29	D2	D1	D0	MN	MN	CRC	CRC	CRC	CRC	EOM
			0	2						8	0	128 bits of user tag data, defined by host								Msg #					

Reader -> Host: Input Status

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP			InSt	InSt	MN	MN	CRC	CRC	CRC	CRC	EOM	
			0	2						8	1	3	7	Input Status	Msg #								

Input Status	Description
x 0	All inputs inactive (all high)
x 1	Orange input active (low)
x 2	Brown input active (low)
x 4	Yellow input active (low)
x 8	Blue input active (low)

Input Changed	Description
1 x	Orange input changed
2 x	Brown input changed
4 x	Yellow input changed
8 x	Blue input changed

Reader -> Host: Send Firmware Version Number

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	VN	VN	VN	VN	MN	MN	CRC	CRC	CRC	CRC	EOM	
			0	2						8	2	Version Number								Msg #			

Note: VN = Version Number (in ASCII decimal)

Reader -> Host: Message Acknowledge

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	MN	MN	CRC	CRC	CRC	CRC	EOM
	0	2								8	3	Msg #						

Note: Msg # in Ack = message # for which ack applies.

Reader -> Host: Class B Supervision

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	InChg	InSt					MN	MN	CRC	CRC	CRC	CRC	EOM
	0	2								8	4	Input Status		RFU				Msg #						

Input Changed	Description
1	x Yellow input
2	x Blue input

Input Status	Description	Description
x 0	Blue input voltage = 0, low level	Yellow input voltage = 0, low level
x 1	Blue input voltage = 0, low level	Yellow input voltage = 2.1, mid-low level
x 2	Blue input voltage = 0, low level	Yellow input voltage = 3.0, mid-high level
x 3	Blue input voltage = 0, low level	Yellow input voltage = 5, high level
x 4	Blue input voltage = 2.1, mid-low level	Yellow input voltage = 0, low level
x 5	Blue input voltage = 2.1, mid-low level	Yellow input voltage = 2.1, mid-low level
x 6	Blue input voltage = 2.1, mid-low level	Yellow input voltage = 3.0, mid-high level
x 7	Blue input voltage = 2.1, mid-low level	Yellow input voltage = 5, high level
x 8	Blue input voltage = 3.0, mid-high level	Yellow input voltage = 0, low level
x 9	Blue input voltage = 3.0, mid-high level	Yellow input voltage = 2.1, mid-low level
x A	Blue input voltage = 3.0, mid-high level	Yellow input voltage = 3.0, mid-high level
x B	Blue input voltage = 3.0, mid-high level	Yellow input voltage = 5, high level
x C	Blue input voltage = 5, high level	Yellow input voltage = 0, low level
x D	Blue input voltage = 5, high level	Yellow input voltage = 2.1, mid-low level
x E	Blue input voltage = 5, high level	Yellow input voltage = 3.0, mid-high level
x F	Blue input voltage = 5, high level	Yellow input voltage = 5, high level

Reader => Host: Message Non-Acknowledge - Configuration Message cannot be implemented or is invalid

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	EC	EC	MN	MN	CRC	CRC	CRC	CRC	EOM
	0	2								8	5	Error Code		Msg #						

Note: Error Code = reason for non-acknowledge
Msg # = message # for which nak applies.

Error Code	Description
1	
2	Info on port not available
3	Invalid message length
4	relay number invalid
5	activation time not valid
6	data received is out of range
7	
8	message invalid during idle mode
9	activate control on with no override enabled
0x21	pass thru was attempted with Input/Output mismatch

Reader -> Host: Get Model Response

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	Mod	Mod							MN	MN	CRC	CRC	CRC	CRC	EOM
	0	2								8	6	RFU								Msg #						

Model Number	Description
1	eTag8-SR
2	IO Module

Reader -> Host: Tag is Still in the Field

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	D31	D30	D29	D2	D1	D0	MN	MN	CRC	CRC	CRC	CRC	EOM																									
	0	2								8	7	128 bits of user tag data, defined by host												Msg #																										

Reader -> Host: Most recent Tag Read is no longer in the Field

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	D31	D30	D29	D2	D1	D0	MN	MN	CRC	CRC	CRC	CRC	EOM																									
	0	2								8	8	128 bits of user tag data, defined by host												Msg #																										

Reader -> Host: Return Current Reader Settings

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	BG	BG	MO	MO	C1	C1	C2	C2	C3	C3	C4	C4
	0	2								8	9												

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
S1	S1	S2	S2	S3	S3	S4	S4	S5	S5	OTL	OTA	MN	MN	CRC	CRC	CRC	CRC	EOM
												Msg #						

BG = current background LED setting, set in Typ 01.
 MO = current card read mode, set in Typ 03.
 C1 = current CF1 setting, set in Typ 04.
 C2 = current CF2 setting, set in Typ 04.
 C3 = current CF3 setting, set in Typ 04.
 C4 = current CF4 setting, set in Typ 04.
 S1 = current CF1 setting, set in Typ 13.
 S2 = current CF2 setting, set in Typ 13.
 S3 = current CF3 setting, set in Typ 13.
 S4 = current CF4 setting, set in Typ 13.
 S5 = current CF5 setting, set in Typ 13.

OTL	Description
0	No latched output
1	Relay is latched
2	Output #1 is latched
4	Output #2 is latched

OTA	Description
0	No active output
1	Relay is active
2	Output #1 is active
4	Output #2 is active

Reader -> Host: Voltage Reading of the Blue and Yellow Wires

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	BL	BL	Y	Y	MN	MN	CRC	CRC	CRC	CRC	EOM
	0	2								8	A					Msg #						

BL = Voltage measured at the blue wire, 0 - FF (hex) where FF (hex) = 5 volts
 Y = Voltage measured at the yellow wire, 0 - FF (hex) where FF (hex) = 5 volts

Reader -> Host: ISO 14443 Response

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	Len	Len	Data	Data	Data	Data	MN	MN	CRC	CRC	CRC	CRC	EOM
	0	2								9	4	0	7							Msg #						

Reader -> Host: ISO 15693 Read Response

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
SOM	DST	DST	SRC	SRC	SRC	SRC	SRC	SRC	SRC	TYP	TYP	Len	Len	Fig	Fig	Data	Data	Data	Data	Data	Data	Data	Data
	0	2								9	5	0	7	0	0								

24	25	26	27	28	29	30
MN	MN	CRC	CRC	CRC	CRC	EOM
Msg #						

Fig	Bit(s)	Description
0	(ISO 15693) 0 = No Error, 1 = Error detected	
1	1 = Response Length > 15 Bytes	
2	1 = Tag Not Found	
3	1 = CRC Error	
4	1 = Input Packet Data Len > 20	
5	Spare - Don't care	
6	Spare - Don't care	
7	Spare - Don't care	

Here is an example of our usage:

```

unsigned int lucCRC;
// Index is the quantity of chars in SendBuff (points to next open spot in array)

lucCRC = calc_crc16(&SendBuff[1], Index-1);
CharToHex( &SendBuff[Index], (unsigned char) (lucCRC >> 8) );
CharToHex( &SendBuff[Index+2], (unsigned char) (lucCRC << 8) );
SendBuff[Index+4] = EOM;

```

Here is our code to compute the CRC:

```

unsigned calc_crc16(char *msgpoly,int bytes)
{
  unsigned crc16 = 0;
  char i;
  while (0< bytes--)
  {
    crc16 = crc16 ^ (int) *msgpoly++ << 8;
    for (i=0;i<8;i++)
    {
      if (crc16 & 0x8000) crc16 = crc16 << 1 ^ 0x1020;
      else crc16 = crc16 << 1;
    }
  }
  return (crc16);
}

```