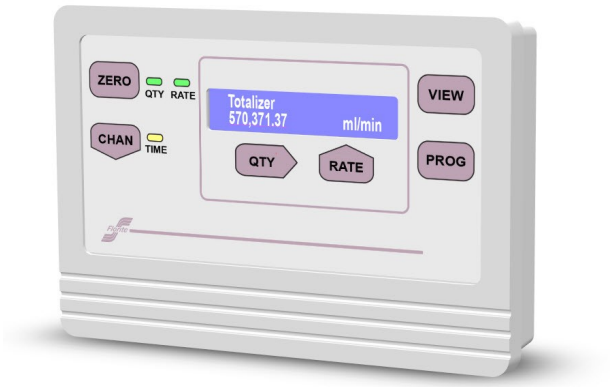
Florite

Process Control by Design

Serial Communication Protocol

700 Series



Florite International, Inc.
500 and 700 Series Instrument Serial Communication Protocol

D10147-011817

700 SERIAL COMMUNICATION PROTOCOL

TABLE OF CONTENTS

1.0 REVISION HISTORY

Effective Date	Revision Enhancement
20Aug1999	1. Initial formats written
22Aug1999	1. Summary form of initial format released for review 2. Cowan reservations on hex use for sum not ascii
28Aug1999	1. Full message structure document complete to Cowan
30Aug1999	1. Revise format include flash instigation by host
01Sep1999	1. Revised sections 3.0, 6.0, 13.0
02Sep1999	1. Message check sum as two ascii-hex characters 2. Comma prelimit all info fields plus check sum
04Sep1999	1. Modified sections 13.1 and 13.2.
10Sep1999	1. Revised sections 7.0, 8.0, 11.1-11.3
22Sep1999	1. Revised sections 6.0, 7.0, 8.0
05Oct1999	1. Revised sections 8.0, 10.0,11.2,11.3
09Oct1999	1. Added notes to sections 13.1 and 13.2
11Nov1999	1. Added null alarm frame fill character 2. Changed value report prefix fill char from space to zero
13Jan2000	1. Added Section 16 - protocol checksum algorithm 2. Added Section 17 - model 540/560/750 old message format
24Feb2001	1. Added rom checksum command/response, modified section 11 command responses for 750 and 990 versions.
24Mar2001	1. Corrected and clarified section 6.0 cmd selections, 10.2 report information format, 11.0 Type 4 Format, 16.6 ascii char convert, 13.2 unit initiated record transfer. Incorporated the exact 990 cmmd operation for comparison to the 750 to highlight differences.

28Mar2001	1. Exchanged msg type with addr extension in applicable messages. Host software implementations shall be capable of accepting messages with either address frame format as follows: AZ,ADR.XTN,TYP, or AZ,ADR,TYP,.XTN,
13Apr2001	1. Incorporated ADR.XTN described above in 750 and 990 code. 2. Section 10.0 - significant host and field unit remote scenario description. 3. Added <DLE><STX> prior to type 0-3 msg start and <DLE><ETX> at record set to delineate multiple record(s) send completion as follows: <DLE><STX> AZ,ADR.XTN,TYP.....<sum><cr><lf> <DLE><ETX> DLE print as right arrow, STX print as happy face, ETX print as a heart. 4. Added AZH and AZS commands. AZH to suspend unsolicited record(s), and AZS to enable unsolicited record(s) send. Applicable to remote communication operation. Field unit assumes send enabled after connect unless field unit otherwise receives an AZH command. 5. Corrected by adding ? symbol at end of each 750 AZP programming line.
31Apr2001	1. Added section 10.3 detail AZH and AZS commands operation.
28Apr2001	1. Corrected section 10.3 number of dle/stx, dle,etx from three to two. 2. Clarification section 11.3 AZJ command 990 response
08Apr2003	1. Added section 18.0, added upgrades to various sections.
18Aug2003	1. Added AZOx output On/Off direct control operation. 2. Added AZZ4 set factory default command. 3. Added AZK networked/non-networked solicited report value block in type 4 frame format. 4. Added Rate Alarm Delay and Input Excitation to AZP programming command. 5. Deleted 900 Series AZJ programmed value request response.
<dd mmm yr> 1.	

2.0 PURPOSE

The purpose of his document is to detail and describe the second generation serial operating commands and responses between a Host computer software program and Florite monitors and and process control equipment.

3.0 COMMAND AND MESSAGE STRUCTURE

3.1 The protocol is established to service need for error detection and correction methods for transferring sensitive value information, and accommodating field unit object code upgrade from a remote host computer.

3.2 Elements

3.2.1 Pre-limit Frame - ascii sentinel indicating start of message

3.2.2 Information Frame - shall containing comma pre-limited fields which start with the first character immediately following the message pre-limiter and include all succeeding characters up to and including a comma which immediately precedes the first ascii-hex character of the checksum.

3.2.3 Check Frame - is two ascii-hex characters which are the 256 Modulo 2's compliment negated sum of all information frame characters which is used to test the message validity.

3.2.4 Delimit Frame - ascii sentinel indicating end of message

3.3 Transfer - All messages are serial half duplex send/response having a four (4) second response default.

3.4 Mastering - the protocol initiator or originator is the master. The master is responsible for managing the communication link connect status. The telecommunication (telcom) link master is the party that places the call and is first to transmit messages.

4.0 UNIT ADDRESSING

Each field unit shall have a unique XXXXX address (ADR) and .X sub-address (XTN) which represents an ascii numeric string which may be omitted when operating within an un-networked unit environment but must be used in multiunit networks to differentiate the units. The command format is characterized as free form variable entry. This method is expeditious and economic in terms of getting the desired result. Commands minimally require a pre-limiter, body and delimiter. Command arguments are single ascii alpha non-case sensitive characters. The symbol XXXXX in this text interchangeably means either the unit address XXXXX or unit address with sub-channel extension XXXXX.XX.

HOST SEND

```
AZ     xxxxx  .       x       K       <cr>
|      |      |       |       |       |
|      |      |       |       |       + -- delimiter
|      |      |       |       |
|      |      |       |       +-- cmmd (menu for example)
|      |      |       |
|      |      |       +-- port number (1 ascii numeric char)
|      |      |
|      |      +-- subaddress (ascii numeric char 0-9)
|      |
|      +-- addr (up to 5 ascii numeric chars < 65536)
|
+-- prelimiter
```

AZ	<argument>	<cr>	Non-network alternative
AZ .X		<cr>	Set multi-channel sub-address to X
AZ XXXXX	<argument>	<cr>	Networked alternative
AZ XXXXX.X	<argument>	<cr>	Networked multi-channel alternative
AZ		<cr>	Stop repetitive command operation

UNIT RESPOND - in accordance with the specific argument.

5.0 RESET STATE COMMAND

This string terminates any command presently in process and resets the command state machine to the initial ready state.

HOST SEND

<esc>AZ<cr>

UNIT RESPOND

<none>

6.0 TERMINAL OPERATOR COMMAND SELECTIONS

The command menu provides available operator commands, make, model and code revision information. Commands which are not for operator use are not displayed.

6.1 Basic process and formats

HOST SEND

AZ XXXXX M <cr>

750 RESPONSE

<cr><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf>
FLORITE COMMANDS - 750MAX68 01.01.13 F000
<cr><lf><lf>
[M] Command Select Menu <cr><lf>
[V] Display All Values <cr><lf>
[K] Output Accumulated Values <cr><lf>
[J] Output Programmed Values <cr><lf>
[I] Product Identification <cr><lf>
[P] Program New Values <cr><lf>
[Z] Reset Accumulated Values<cr><lf>

6.2 The unit type 750MAX68 indicates that the cpu type is a 6800 specifying the firmware assembly type. The line may alternatively be 750MAX11 indicating an 68HC11 type cpu. The cpu type is important for determining which assembly object code type is necessary for reprogramming.

7.0 VIEW PROGRAMMED AND ACCUMULATED VALUES

All programmed and measured values.

HOST SEND

AZ XXXXX V <cr>

750 RESPONSE

```
<cr><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf>
     PROGRAMMED VALUES
<cr><lf><lf>
     Network Address  = 00000.0             Answer Rings      = 010
     Primary Phone    = 0000018002287776    Secondary Phone   = 0000000000000000
     Security         = Off                 Report            = Off
     Measure Units    = gal                 Rate Time Base    = min
     Rate Alarm Type  = HiLo                Relay Operation   = Normal
<cr><lf><lf>
     Quantity 1 Limit = 00000000.00 gal     Quantity 2 Limit  = 00000000.00 gal
     Low Rate Limit   =  0000000.00 gal/m   High Rate Limit   =  0000000.00 gal/m
     Time Limit       =         0168 hrs    Meter Constant    =  0000015715 p/gal
<cr><lf><lf>
     Date-Time        = 21Feb01 14:11:15    Report Start      = 02Dec00 12:00:00
     Next Report Time = 02Dec00 14:46:50    Report Frequency  = 000 minutes
<cr><lf><lf>
     ACCUMULATED VALUES
<cr><lf><lf>
   Quantity 1        Quantity 2        Process Rate        Peak Rate         Time
________________  ________________  __________________  __________________  _________
 00000007.38 gal   00000007.38 gal    0000000.00 gal/m    0000000.36 gal/m  0098 hrs <cr>
```

NOTE:

* Last line terminated with <cr> and reprints every 1.0 seconds.
* Reprint may be terminated by sending an AZ<cr> string.

8.0 PROGRAM OPERATING VALUES

HOST SEND

AZ XXXXX P <cr>

750 RESPONSE

```
<cr><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf><lf>
<lf>
     PROGRAM VALUES
<cr><lf>
     Quantity 1 Limit   (xxxxxxxx.xx) = 00000000.00 gal ?
<cr><lf>
```

Quantity 2 Limit (xxxxxxxx.xx) = 00000000.00 gal ?
<cr><lf>
Time Limit (xxxx hrs) = 0168 hrs ?
<cr><lf>
Meter Constant (xxxxxx) = 015715 p/gal ?
<cr><lf>
Rate Time Base (0=m 1=s 2=h) = 0 ?
<cr><lf>
Relay Type (0=Norm 1=Rev) = 0 ?
<cr><lf>
Low Rate Limit (xxxxxxx.xx) = 0000000.00 gal/m ?
<cr><lf>
High Rate Limit (xxxxxxx.xx) = 0000000.00 gal/m ?
<cr><lf>
Network Address (xxxxx) = 00000 ?
<cr><lf>
Rate Alarm Type =Peak 1=HiLo) = 1 ?
<cr><lf>
Measure Units (uuu) = gal ?
<cr><lf>
Pri Phone (xxxxxxxxxxxxxxxx) = 0000018002287776 ?
<cr><lf>
Sec Phone (xxxxxxxxxxxxxxxx) = 0000000000000000 ?
<cr><lf>
Answer Rings (xxx) = 010 ?
<cr><lf>
Date-Time (##Mon## ##:##:##) = 21Feb01 14:12:55 ?
<cr><lf>
Report Start (##Mon## ##:##:##) = 02Dec00 12:00:00 ?
<cr><lf>
Report Frequency (### unit) = 000 minutes ?
<cr><lf>
<cr><lf>

NOTES:

* Receipt of <cr> at the end of a program value line causes no line value change.
* Receipt of <esc> causes immediate programming termination.

9.0 ZERO ACCUMULATED VALUES

This command resets selected accumulated values to zero

HOST SEND

AZ XXXXX Zn <cr>

n=0	quantity 1
n=1	quantity 2
n=2	quantity 1, quantity 2, time
n=3	time
n=4	set factory defaults

UNIT RESPOND

<none>

10.0 MESSAGE TYPE 0-3 FRAME FORMAT

These message types are characterized for use in transferring value information initiated by field unit to be received by the host. These messages are sent unsolicited by the field unit to send numeric and status information. These messages are instigated at the field on the occurrence of one or more alarms (type 0), scheduled report (type 1), test dial-out installation (type 2), and maintenance action acknowledgement (type 3).

10.1 RECORD SEND AND ACKNOWLEDGMENT

The field unit dials and connects with the host and waits 10 sec to insure the host modem error correction negotiation is completed which usually requires a modem dependant 5 sec period.

Prior to the expiration of 10 sec, the host may send an AZH command causing the field unit to suspend sending its unsolicited message until the the host thereafter sends an AZS command releasing the hold condition. The AZH and AZS are equivalent to flow control operators XOF and XON respectively. Once the suspend command AZH is issued, the host has become the line master and the field unit will stay on line indefinitely until the host disconnects.

The field unit sends the message that originally caused it to dial-in to the host. Four seconds are allowed for the host to receive the information record(s) and acknowledge correct receipt. The host evaluates the record(s) checksum(s) and returns an acknowledge to the unit in the form of AZXXXXXN<cr> negative acknowledge (NAK) that one or more records were incorrect, or an AZXXXXXA<cr> positive acknowledge (ACK) indicating that all record(s) were received correctly.

On receipt of a NAK or no host response in 4 sec following sending of the record(s), the field unit resends all

record(s) and waits for another reponse from the host. The field unit will try to resend its record(s) and receive an ACK host response 3 times, after which it will disconnect from the line and reinitiate the send process 24 hours later.

The field unit intrepretes AZA as positive confirmation that the message was received correctly and no further action by either host or field unit is thereafter required. The field unit thereafter disconnects from the host.

Following a host ACK - the field unit will remain on the line for 4 sec waiting for the host to send a valid AZ command indicating that the host desires to continue the session. If an AZ command is not received after the 4 sec period - the field will disconnect from the line.

If the host sends a valid AZ command within the 4 sec period after the field units unsolicited record(s) have been correctly received by the host, the host now becomes the link master and the field unit will stay on the line waiting for the host to disconnect, or communicate as the host so desires. Under these circumstances, the field unit will disconnect from the line if no characters are received from the host in any 30 sec period. In any case - the field unit will disconnect from the line after 8 minutes from the time it connected with the host.

10.2 MESSAGE DESCRIPTION

750 MESSAGE
AZ,ADR.XTN,TYP,QTY1,QTY2,RAT,PEK,HRS,Q,C,R,T,<sum><cr><lf>

10.3 Message type 0-3 transmissions are prefixed with <DLE><STX> and terminated with <DLE><ETX> to delineate the entire message record set.

FRAME	DESCRIPTION	FORMAT	CONDITION
PRELIMIT			
<DLE><STX>	start sentinal	2 ascii ctrl chars	start record set
AZ	record prelimit2	char ascii alpha	start record
INFO			
ADR	unit addr	5 char ascii numeric	base address
.XTN	sub-address	ascii numeric	sub-address
TYP	msg type	1 char ascii numeric	
	0 - alarm		unsolicited alarm msg
	1 - report		unsolicited report msg
	2 - test	[QTY]+[RATE] keys	demand install dial test
	3 - action	[RST 2] key	demand action ack (secure)
	4 - info	I,K,J,C	solicited info response
	5 - ctrl		Unsolicited control service status
	6		<reserved>
	7		<reserved>

	8		<reserved>
	9 - code		object code request/download

MEASUREMENTS

QTY1	accum qty 1	ascii numeric	
QTY2	accum qty 2	ascii numeric	
RAT	immediate rate	ascii numeric	signed data, +, - or space (+)
PEK	peak rate	ascii numeric	signed data, +, - or space (+)
HRS	accum time	ascii numeric	

ALARMS

Q	qty 1 alarm	ascii alpha	qty 1 alarm
C	qty 2 alarm	ascii alpha	qty 2 alarm
R	rate alarm	ascii alpha	high/low rate alarm
T	time alarm	ascii alpha	service time alarm
X	<default>	ascii alpha	indicates frame has no alarm

CHECK

<sum>	check sum	two ascii-hex bytes	2's comp negate of mod256 sum

DELIMIT

<cr><lf>	record delimit	2 ascii ctrl chars	end of record
<DLE><ETX>	end sentinal	2 ascii ctrl chars	end record set

750 EXAMPLE

AZ,00909.0,0,00000988.93,00162871.43,+0000003.27,+0000345.67,00022,Q,X,R,X, <sum><cr><lf>

10.3 SUSPEND TYPE 0-3 MESSAGE COMMANDS

10.3.1 This command is applicable to unsolicited field unit type 0-3 messages to be communicated to the host. The AZH command acts as an XOF to suspend the field units message.

HOST SEND
AZH<cr>

UNIT RESPOND
<none>

10.3.2 This command is applicable to unsolicited field unit type 0-3 messages to be communicated to the host. The AZS command acts as an XON to enable the field unit to send the message which was previously suspended by the host using the AZH command.

HOST SEND
AZS<cr>

UNIT RESPOND

<none>

10.4 CONTROL COMMAND - OUTPUT PORT

This command enables serial On/Off output port operation when the Control Function is set for Manual allowing direct control of the port output as follows:

HOST SEND

AZ.PO0<cr>	Non-networked port P Off
AZ.PO1<cr>	On
AZ XXXXX.P O0<cr>	Networked port P Off
AZ XXXXX.P O1	On

11.0 MESSAGE TYPE 4 FRAME FORMAT

These message types are characterized as informational only and are for use in transferring value information initiated by the host to be sent to the field unit which produces a response back to the host. The field unit response checksum may be investigated by the host for validity. Should the packet message be incorrect - the host may command the unit to resend the message.

11.1 UNIT IDENTIFICATION

HOST SEND

AZ XXXXX I <cr>	request unit identification

UNIT RESPOND

AZ,ADR,4,A,B,C,D,<sum><cr><lf>

AZ	pre-limit
,00000	address
,4	function
,FLORITE	make
,750MAX68	model
,01.01.13	object date code - yr-mon-day
,F000	start vector block address (else F800)
,<sum>	2's comp negate of mod256 sum
<cr><lf>	de-limit

750 EXAMPLE

AZ,00000,4,FLORITE,750MAX11,01.01.13,F000,<sum><cr><lf>

11.2 SEND ACCUMULATED VALUES

750 HOST SEND

AZ XXXXX K <cr>

AZ K <cr>	Send all non-networked input ports having comport set for Report in standard message type 4 block frame format.
AZ XXXXX K <cr>	Send all networked input ports having comport set for Report in standard message type 4 block frame format.

RESPONSE

AZ,ADR.XTN,4,A,B,C,D,E,<sum><cr><lf>

AZ	prelimit

,00000	address
.0	subaddress
,4	msg function
,00000007.38	qty 1
,00000007.38	qty 2
,- 0000000.00	rate/value
,+0000000.36	peak rate/value
,00098	service time
,<sum>	2's comp negate of mod256 sum
<cr><lf>	delimit

750 EXAMPLE

AZ,00000.0,4,00000000.00,00000000.00,- 0000050.00,- 0000049.90,00024,<sum><cr><lf>

11.3 SEND PROGRAMMED VALUES

HOST SEND

AZ XXXXX J <cr>

750 RESPONSE

AZ,ADR.XTN,4,A,B,C,D,E,F,G,H,I,J,K,L,M,N,P,R,<sum><cr><lf>

AZ	prelimit
,00000	unit address
.0	unit sub-address
,4	msg function
,00000000.00	qty 1 alarm limit
,00000000.00	qty 2 alarm limit
,0168	time alarm limit (hrs)
,0000015715	digital meter constant
,0	rate time base
,0000000.00	low rate alarm limit
,0000000.00	high rate alarm limit
,00000	unit address
,1	rate alarm type (high-low/peak)
,gal795:=	measure units - configuration options

configuration options

fixed measure units

0	gal
1	ltr
2	ozs
3	ml

Relay

4	Reverse

5	Normal	
	Report	
6	On	
7	Off	
	Security	
8	On	
9	Off	
	Error Control	
:	Off	
;	On	
	Code Version	
<	Standard	(old protocol)
=	Maximum	(new protocol)
	Batch Control	
>	Off	
?	On	
	Dose Control	
@	Off	
A	On	
	Meter Constant Granulation	
B	10^0	(xxxxxxxxxx)
C	10^-1	(xxxxxxxxx.x)

,0000018002287776	pri phone
,0000000000000000	sec phone
,010	ansr rings
,21Feb01 14:12:12	date-time
,02Dec00 12:00:00	report start date-time
,000 minutes	report freq (000=report off)
,<sum>	2's comp negate of mod256 sum
<cr><lf>	delimit

750 EXAMPLE

AZ,00000.0,4,00000000.00,00000000.00,0168,0000015715,0,0000000.00,0000000.00,00000,1,gal795:=,
0000018002287776,0000000000000000,010,21Feb01 14:12:12,02Dec00 12:00:00,000 minutes,
<sum><cr><lf>

11.4 SEND ROM CHECKSUM

The ROM checksum interrogate command may be used at any time and is particularily useful following flash download to verify each byt actually sent - and not just the whole rom where portions may NOT have been downloaded or address spaces not actually used. The cksm value is returned as 3 byt straight add/accumulate converted to ascii-hex. An 8 bit 32K rom with all "FF" byts would result in a cksm of $7F8000 (maximum). Three address block segments are summed as follows:

From	To

START_CODE $F000 or $F800	START_CODE_END-1 last start code seg byt
MAIN_CODE $8000	MAIN_CODE_END-1 last main code seg byt
VECTORS (6303 or HC11)	UP THRU $0000 - including byt at address $0000-1

HOST SEND

AZ XXXXX C <cr>

RESPONSE

AZ,ADR,4,A,<sum><cr><lf>

AZ	prelimit
ADR	address
4	msg type
A	six ascii-hex char rom check sum
<sum>	2's comp negate of mod256 sum
<cr><lf>	delimit

EXAMPLE

AZ,00000,4,SSSSSS,<sum><cr><lf>

12.0 UNIT-TO-UNIT CLONE COMMANDS

These commands are unsed to send and receive binary ram register information between a pair of interconnected monitor units which are designated READER unit and FIELD unit.

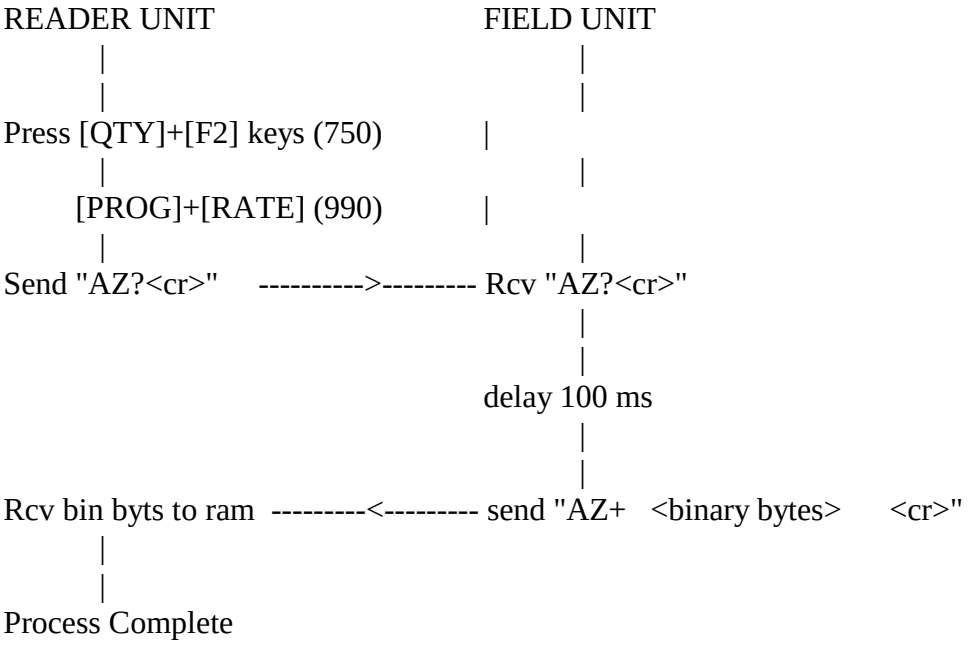
READER SEND

AZ ? <cr>

FIELD UNIT RESPONSE

AZ + <binary bytes> <cr>

TRANSFER PROCESS



13.0 MESSAGE TYPE 9 FRAME FORMATS - <SUSPENDED>

This message type is used to transfer object code from a host computer to a field unit. The field unit programs the host records into its instruction memory. Either the host or the field unit may independently initiate a request for object download.

13.1 HOST INITIATED RECORD TRANSFER - <SUSPENDED>

Object record transfer may be initiated by the host in several ways as follows:

13.1.1 Direct connect with the unit and sending the initiation string; or

13.1.2 Send initiation string within four seconds following receipt of a type 0 or 1 message; or

13.1.3 Send initiation string within thirty seconds following receipt of a type 2 or 3 message.

HOST SEND

AZ XXXXX L40 <cr> host send object records command

NOTE: the XXXXX address may be omitted for single non-networked unit

UNIT RESPOND

AZ XXXXX A <cr> unit rcv ready for first object record

NOTE: unit shall respond with XXXXX address

HOST SEND

<object record>

UNIT RESPOND

AZ XXXXX A <cr> record rcv CORRECT - send next record

AZ XXXXX N <cr> record rcv ERROR - resend last record

NOTE: unit shall respond with XXXXX address

13.2 UNIT INITIATED RECORD TRANSFER - <SUSPENDED>

Object record transfer may be initiated by a field unit after connection with the host by sending the initiation string requesting object download.

UNIT SEND

<AZI msg> Request host to send object records with ident msg type 9

750 VERSION EXAMPLE

AZ,XXXXX,9,FLORITE,750MAX11,01.08.22,F800,<sum><cr><lf>

HOST RESPOND

AZ XXXXX L40 <cr> download available and may proceed

NOTE: the XXXXX address shall always be mod 2^16 ascii numeric - all five digits representing a value from 0 to 65535.

UNIT SEND

AZ XXXXX A <cr> unit rcv ready for first object record

NOTE: unit shall respond with XXXXX address

HOST SEND

<object record>

UNIT RESPOND

AZ XXXXX A <cr> record rcv correct - send next record

AZ XXXXX N <cr> record rcv ERROR - resend last record

NOTE: unit shall respond with XXXXX address

14.0 MESSAGE TYPE 9 OBJECT RECORD FORMAT - <SUSPENDED>

The message type 9 object code record format shall be standard Intel Hex enabling direct compatibility with existing assembly hex files. In the context of this message type, bytes are binary, characters are ascii.

FORMAT

record : 40 E160 00 0A5465737420616E6420736574202446....
.....46202D3E2000504153530D0A0A004641 DC <cr><lf>

short line : 06 E180 00 494C0D0A0A00 E3 <cr><lf>

vectors : 12 FFEE 00 E000E000E000E000E000E000E000E000E000E000 21 <cr><lf>

load end : 00 0000 01 FF <cr><lf>

DATA RECORD [:] [count] [addr] [load type] [data] [cksm] [cr] [lf]

start	1 char	':'
count	2 chars	data byte count
addr hi	2 chars	'x','x'
addr lo	2 chars	'x','x'
type	2 chars	'0','0'
data	2 chars	'x','x'
cksm	2 chars	mod256 2's comp sum of count, addr, type, data
end	2 chars	'cr','lf'

END RECORD [:] [00] [addr] [load type] [cksm] [cr] [lf]

start	1 char	':'
count	2 chars	'0','0'
addr hi	2 chars	'x','x' (typically '0','0')
addr lo	2 chars	'x','x' (typically '0','0')
type	2 chars	'0','1'
cksm	2 chars	mod256 2's comp sum of count, addr, type
end	2 chars	'cr','lf'

15.0 FLASH WRITE CONDITIONS AND RESTRICTIONS - <SUSPENDED>

15.1 CONDITIONS

15.1.1 Entire code initially written in flash rom before installed in pcb

15.1.2 Process receives record and writes bytes to specified load address

15.1.3 Records shall be contiguous and contain 64 (40H) bytes or less

15.1.4 Loader executes from flash rom - relocatable record writer executes from cpu ram

15.1.5 One requested hex file record is serviced at a time
unit ACK = record good - send next record
unit NCK = record bad - resend last record

15.1.6 Unit error checks written record before ack - will retry write once will thereafter abandon loading senario and disconnecting from HOST.

15.2 RESTRICTIONS

1. Assembler responsible for pretransmit flash page record alignment

2. Records in the following ranges are protected - flash rewrite ignored:

START_CODE thru START_CODE_END	boot sector
FLASH_DATA thru FLASH_DATA_END	little data block

3. Successive records shall start at an address greater than the previous record end address

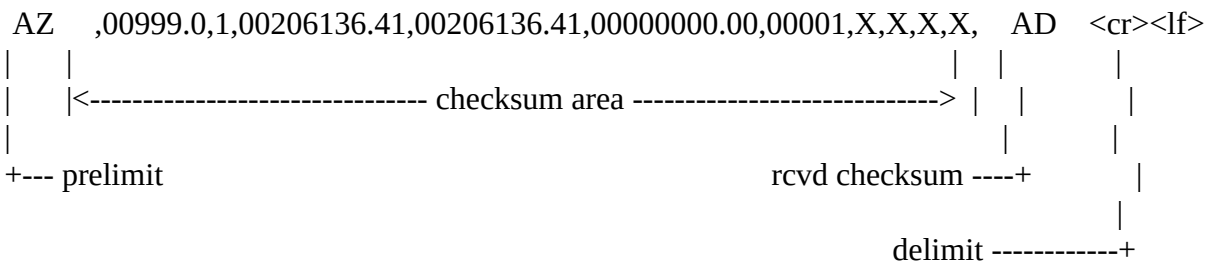
4. No data from subsequent records is allowed to reside in the same flash page as any previously programmed record

5. Loader receives record and writes to flash at the specified start address without regard to the record end address

16.0 RECORD CHECKSUM ALGORITHM

16.1 Checksum Parse and Calculate Algorithm - Modulo 256 Ascii String

Example Record String



16.2 Checksum Ascii-Decimal Conversion:

'AD' = [(10 * 16) + 13] = 173 (decimal)

NOTE: 'FF' (255 decimal) is largest received checksum which is the modulus minus one.

Example string above sums (modulo 256) to 83, so that:

83 + 173 = 256, which becomes zero when 256 is subtracted.

16.3 Computing Requirements:

16.3.1. Numeric calculations performed in base 10 (decimal)
16.3.2. Add and subtract functions available
16.3.3. Accumulate in some pre-designated register space
16.3.2. Ability to test numeric magnitude (like a trial subtract)

16.4 Accumulation Process:

Sum string char values beginning with char immediately following prelimit string, up to, but not including, delimit string. Checksum ok if accumulate is zero. The beauty of this technique is that the checksum sent to you when added to the present sum will cause your accumulated sum to end up zero is the string is valid (checksum=ok). Great hua?

Basically - ya keep adding up the values until detecting the delimit string, or upon reaching an accumulated value of 256 or more as follows:

If value is now 256 or greater - just subtract 256 from present sum
and keep chugging more summing.
ELSE
If CR char detected - get NEXT received char:

Char is LF - goto exit and test accumulated value is zero:
Value zero -> send ACK string to remote unit and
process string contents.

ELSE
Goto error exit, ignore string, send NAK string to remote unit.
ELSE
Char NOT LF - goto error exit, ignore string, send NAK string to remote unit. Q.E.D. - there it is.

16.5 Processing Logic Implementation:

```
        ; preset accumulator and extract pointer
sum_0:
        * zero -> accumulate reg
        * indx -> char immediately following prelimit string

        ; accumulate decimal string char sum
sum_1:
        * get char
        * if char not 13 goto sum_2                     ; <cr> delimit char?
        * indx+1 -> indx
        * get char
        * if char 10 goto sum_3 else goto sum_error

sum_2:
        * gosub convert -> return char decimal [value]
        * {accumulate(n-1) + [value]} -> accumulate(n)
        * indx <- indx+1                                ; point next char
        * test [accumulate - 256]                       ; modulo ovrflo?

                * if negative goto sum_1
                        ELSE
                * {accumulate(n) - 256} -> accumulate(n)
                * goto sum_1

        ; determine checksum validity
sum_3:
        * test [accumulate - 0]
                * if result not zero -> goto sum_error

        * <exit to sum ok services>

sum_error:
        * <exit to sum error services>

        ; convert ascii hex char -> decimal and return [value]
convert:
        * <lookup/calculate decimal value via Value Convert Table>
        RETURN
```

16.6 Ascii Value Conversion Table

recd ascii print char	rcvd hex value	decimal value	notes
A	41	65	prelimit(0)
Z	5A	90	prelimit(1)
,	2C	44	format
.	2E	46	format
0-9	30-39	48-57	numeric
C	43	67	alpha
Q	51	81	alpha
R	52	82	alpha
H	48	72	alpha
L	4C	76	alpha
T	54	84	alpha
X	58	88	alpha
cr	0D	13	delimit(0)
lf	0A	10	delimit(1)
dle	10	16	escape
stx	02	2	start text
etx	03	3	end text

17.0 750/560/540 CLASSIC MESSAGE FORMAT - <SUSPENDED 09.04.06>

17.1 Remote Serial Message Formats

17.1.1 Unsolicited Report

<CR><LF><AAAAA>	network address
<CR><LF><BEL>	prelim
<CR><LF><QQQQQQQQ.QQ>	quantity 1 value
<CR><LF><CCCCCCCC.CC>	quantity 2 value
<CR><LF><BEL>	delim

17.1.2 Alarm(s) Report

<CR><LF><AAAAA>	network address
<CR><LF>	
<CR><LF><max hrs alarm>	exceeded service time limit
<CR><LF>	
<CR><LF><max qty alarm>	exceeded quantity 1 limit
<CR><LF>	
<CR><LF><low flow alarm>	exceeded flow rate limit
<CR><LF>	
<CR><LF><max cyc alarm>	exceeded quantity 2 limit
<CR><LF>	

17.1.3 Dial Installation Test

<CR><LF><AAAAA>	network address
<CR><LF>	

17.2 Local Serial Message Formats

17.2.1 AZK<CR> Command - Output accumulated values

<BEL><CR>	prelim
<LF><QQQQQQQQ.QQ><CR>	quantity 1 value
<LF><CCCCCCCC.CC><CR>	quantity 2 value
<LF><FFFFFFFF.FF><CR>	flow rate
<LF><GGGGGGGG.GG><CR>	average flow rate
<LF><HHHHH><CR>	service time
<LF><BEL>	delim

17.2.2 AZJ<CR> Command - Output programmed variables

<BEL><CR>	prelim
<LF><TTTTTTTT.TT><CR>	quantity 1 alarm limit
<LF><CCCCCCCC.CC><CR>	quantity 2 alarm limit
<LF><HHHHH><CR>	time limit
<LF><NNNNN><CR>	meter constant
<LF><YYY.YY><CR>	<unused>
<LF><FFF.FF><CR>	<unused>
<LF><LLL.LL><CR>	low flow rate alarm limit
<LF><HHH.HH><CR>	high flow rate alarm limit
<LF><AAAAA><CR>	network address
<LF><PPPPPPPPPPPPPPP><CR>	phone number
<LF><RRR><CR>	answer rings
<LF><BEL>	delim

17.2.3 AZP<CR> Command - Serial Value Programming

Programming units above supports a multiple selection string which may be used to set one or more function options. For example the string "2579<cr>" selects ozs units, relay normal off, service time alarm, and keypad un-secure.
Star symbol "*" indicates the default selection.

<CR><LF>	Prelimit sentinal
Tot Qty (xxxxxxxx.xx) = ?<BS> <CR><LF>	Total qty alarm limit
Cyc Qty (xxxxxxxx.xx) = ?<BS> <CR><LF>	Cyc qty alarm limit
Time Limit (hrs) = ?<BS> <or> Report Period (hrs) = ?<BS>	Time alarm limit Report period
<CR><LF> Constant (xxxxx) = ?<BS>	Sensor meter constant
<CR><LF> YHZ (xxx.xx) = ?<BS>	Sensor offset constant (always zero)
<CR><LF> FCO (xxx.xx) = ?<BS>	Sensor low linear constant (always zero)

<CR><LF>
Low Flow (xxx.xx) = ?<BS> Low flow alarm limit

<CR><LF>
High Flow (xxx.xx) = ?<BS> High flow alarm limit

<CR><LF>
Address (xxxxx) = ?<BS> Unit network address

<CR><LF>
Flow Rate (0=avg 1=hi-lo) = ?<BS> Flow indicate type (always Hi-Lo)

<CR><LF>
Units (0=gal 1=ltr 2=ozs 3=mltr) = ?<BS> Measure units/report/secure/relay

[0] gal	[4] Relay normal on	[8] Keypad Secure
[1] ltr	[5] Relay normal off*	[9] Keypad Unsecure*
[2] ozs	[6] Report period (hrs)	
[3] mltr*	[7] Service time alarm*	

<CR><LF>
Phone Number (xxxxxxxxxxxxxxxx) = ?<BS> host phone number - right justified left zero filled

<CR><LF>
Answer Rings (xxx) = ?<BS> answer ring count

<CR><LF> delimit sentinel