Sensormatic Protocols

17 July 2002

TABLE OF CONTENTS

Section **Page**

[1] $Header: d:/txb-s422/RCS/sproto.tex,v 1.27 2002-06-12 15:50:19-07 Hamilton Exp Hamilton $

APPENDIX A

APPENDIX B

APPENDIX C

APPENDIX D

APPENDIX E

APPENDIX F

APPENDIX G

LIST OF FIGURES

Figure **Page**

Note

In the Sensormatic publication: “Communication Protocols and Cable Networks, How They are Different and When to Use Them, 8000-2573-19”, they mention three different types of networks that are used by Sensormatic. The following table was extracted from that document.

	SensorNet	**RS422**	**Manchester**
Cable type	1 unshielded twisted pair	2 shielded twisted pair	1 shielded twisted pair
Wire gauge	22 AWG	22 AWG	18 AWG
Maximum length	1 Km (3,281 ft.)	1 Km (3,281 ft.)	1.5 Km (5,000 ft.)
Connection	Non-polarized	Polarized	Polarized
Data rate	230.4 Kbits/sec	4.8 Kbits/sec	31 Kbits/sec
Maximum devices per line	Depends on device loading.	Depends on device loading.	Depends on device loading.

In addition the above “physical” connection methods, there are several sub-types of information that are carried. These different types are slowly getting understood, however there are many gaps in what is known.

Miscellaneous information relating to Sensormatic’s protocols, consists of:

1. SensorNet protocol. Supports a maximum of 32 devices on a line. As of 16MAY01, nothing else is known about it.
2. Manchester protocol. Supports a maximum of 3 devices on a line. As of 16MAY01, this has never been encountered.
3. The “RS-422/RS-485” protocol. Supports a maximum of 10 devices on a line.

We have an old document that describes the RS-422/RS-485”protocol. It was copied from a multiply Xeroxed and FAXed document that is titled “RS-422/RS-485 Communications Protocols”, with a sub-title of “For SpeedDome® Ultra IV and AD DeltaDome® Camera Domes, and later versions”.

On the bottom of each page is says “8000-2694-01 Rev 0” and “User’s guide”. (I recently got a newer revison “Rev. A” and have used it to update this document.)

From data acquired in the field (Fresno and Clovis Sears) and in the lab (from an AD2083/06 and SIM58) there appear to be several variations from this described protocol.

The basis of this document, the one that you are reading now, was copied from the Sensormatic original.

Starting in March 2001, various additions, amplifications and changes have added to the this basic document. Almost none of these improvements are indicated.

This page intentionally left blank

1 Sensormatic RS-422/RS-485 Protocol

1.1 About this Guide

This guide explains the RS-422/RS-485 asynchronous communications protocols and how to develop a controlling protocol interface for SpeedDome® UltraDome IV and AD DeltaDome® camera domes, and later versions.

> **Pro-to-col** *n* An agreed-upon format for transmitting data between two devices. The protocol determines the following:
>
> - The type of error checking to be used.
> - Data compression method, if any.
> - How the sending device will indicate that it has finished sending a message.
> - How the receiving device will indicate that it has received a message.

Communication protocols, such as RS-422 and RS-485 allow two network-compatible devices, such as video controllers and remote camera domes to exchange data.

1.1.1 If you need assistance

Call Customer Support at: 1-800-543-9740. (*Note that this is for a Sensormatic phone number.*)

1.2 Types of domes used to verify this protocol

Sensormatic/American Dynamics types of domes used to examine this protocol consist of (more details are in Appendix A, page A-1:

1. Sensormatic Dome #1. This is a working single circuit board type of color camera dome that does not display any configuration information on power up.
2. Sensormatic Dome #2. This is a semi-working (positioning works, camera displays vertical colored bars) double circuit board type of color camera dome. When powered up there is no display of configuration information.
3. Sensormatic Dome #3. This is a non-functioning double circuit board type of ??? (color or B&W) camera dome.
4. Sensormatic Dome #4. This is a new model American Dynamics UltraDome type of color dome. When powered up it displays: "`BOOT VER 0103 MAIN VER 0303`" and "`22X OPTICAL ZOOM`" in double high characters. In normal height characters there are the results of passing (or otherwise, ours always passes) the "`COMM. LOOPBACK`", "`CAMERA LOOPBACK`" and "`MOTOR CIRCUIT`" tests.

[2] `$Header: d:/txb-s422/RCS/sproto.inc,v 1.80 2002-05-31 11:58:22-07 Hamilton Exp Hamilton $`

5. Sensormatic Dome #5. This is working “new” single circuit board type of B+W camera dome. This dome does not display any configuration information when it is initially powered up.

6. As specials when either an “**SC**” or an “**SF**” indicator is used that indicates that this command information was verified at either the Sears Clovis or Fresno store. Usually the exact type of dome is unknown.

1.3 Protocol Characteristics

The RS-422/RS-485 protocol characteristics are listed in Table 1, Page 8.

Table 1. Protocol characteristics

Data Rate	4.8 kbits/sec
Data Format	Start bits: 1 Data bits: 8 Parity bits: 0 Stop bits: 1
Format	3 bytes: Dome Address Command Checksum
Max devices per line	10 (Depends on device loading)
Cable type	2 shielded, twisted pair.
Wire gauge	22 AWG*
Max. length	1 km (3.28?)
Connection	Polarized
Data and Power Connectors Color-Code Conventions	Color Designation Black 24 VAC Red Ground White 24 VAC Orange RS-422 Data in High (+) Green RS-422 Data in Low (—) Yellow RS-422 Data out High (+) Brown RS-422 Data out Low (—)

∗ Sensormatic composite cable is recommended. This cable also contains wires for power and video. In another cable is substituted, cable ??? colors may be different.

[3] $Header: d:/txb-s422/RCS/sprotoch.inc,v 1.2 2001-11-30 15:12:06-08 Hamilton Exp Hamilton $

1.3.1 RS-422/RS-485 Commands

1. The majority of dome control communications is by 3-byte data packets consisting of Dome Address, Command, and Checksum. The Dome Address range if from 1_{10} to 99_{10} (`0x01` to `0x63`), depending on the type of control system used. The Checksum is calculated by subtracting the sum of the bytes from `0x100` and using the least significant byte of the results. The dome acknowledges ACK a command by sending its 1-byte address within 25 milliseconds.

   Three byte commands have this format:

Dome Address	Command Code	Checksum

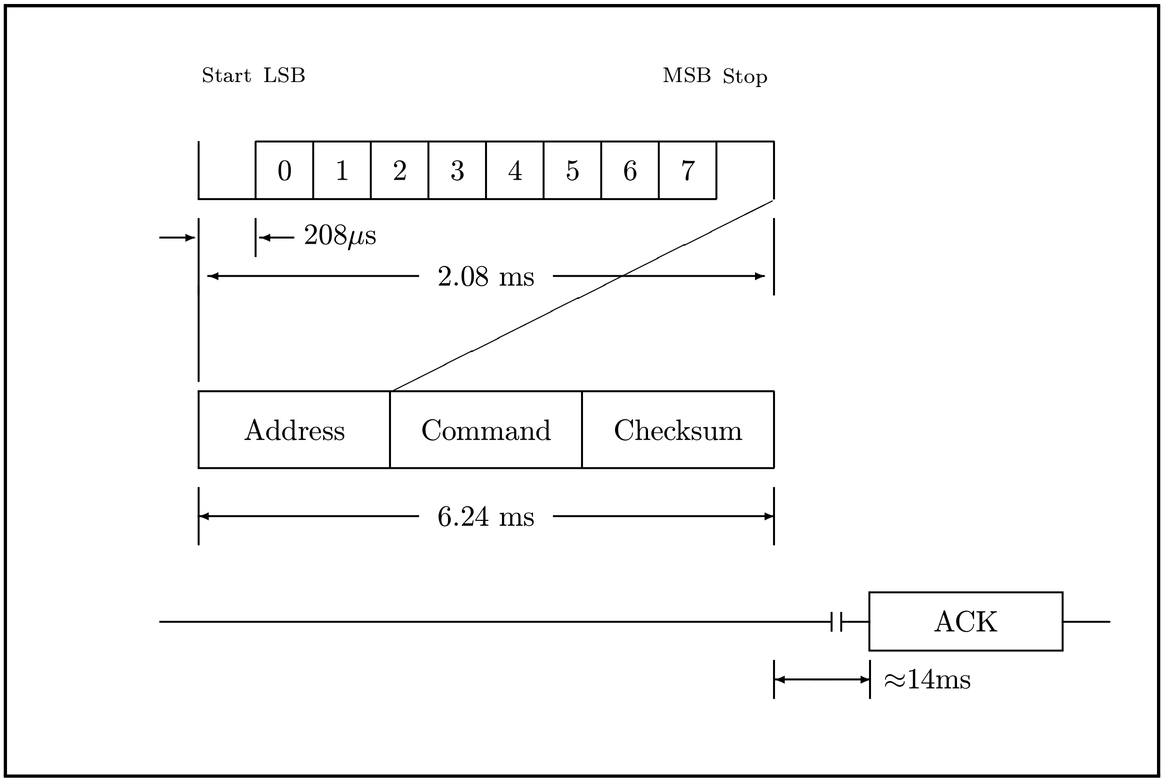
2. The domes also recognizes multiple-byte data commands. These commands consist of Dome Address, Command, 1-N Data, and Checksum bytes.

   Multiple byte commands have this format:

Dome Address	Command Code	Data	Checksum

3. Pan and Tilt speeds shown in °/sec in the tables are nominal speeds with a wide-angle zoom setting. The Zoom Adjusted Program™ (ZAP) feature automatically adjusts pan and tilt speeds scaled in proportion to zoom positions.

4. Most responses are one byte in length and consist of the dome's address.



$RCSfile: 3bytfmt.inc,v $

Figure 1. 3-byte format

1.4 Controller to Dome Communications

Values that are a single number were all generated with an American Dynamics AD2050 matrix driving an AD2083/02 translator or with special purpose software such as `sim58.bas` or `seqtest.bas`.

1.4.1 Standard 3-byte Format

Table 2. Commands from the controller to the domes

Command name	Description	Code	Response
SUNKNOWN 80 `S_UNKNOWN80`	**Unknown** three byte command. “Part of the power up sequence. Is sent by the controller to a dome.” (See Section 1.8.9.2, page 48)	`0x80` 128	Table 5, Page 27
—	Causes a three byte message to be sent from the dome		Section 1.5.2, page 27
PAN LEFT `S_PAN_LEFT`	Pan left (24°/sec) until Pan Right or Pan Stop (See Section 1.8.12.1, page 51)	`0x81` 129	ACK
PAN RIGHT `S_PAN_RIGHT`	Pan right (24°/sec) until Pan Left or Pan Stop (See Section 1.8.12.1, page 51)	`0x82` 130	ACK
PAN STOP `S_PAN_STOP`	Stop panning (See Section 1.8.12.1, page 51)	`0x83` 131	ACK
TILT UP `S_TILT_UP`	Tilt up until Tilt Down or Tilt Stop (See Section 1.8.12.1, page 51)	`0x84` 132	ACK
TILT DOWN `S_TILT_DOWN`	Tilt Down until Tilt Up or Tilt Stop	`0x85` 133	ACK
TILT STOP `S_TILT_STOP`	Stop tilting (See Section 1.8.12.1, page 51)	`0x86` 134	ACK
FOCUS NEAR `S_FOCUS_NEAR`	Focus near until Focus Far or Focus Stop	`0x87` 135	ACK
FOCUS FAR	Focus far until Focus Near or Focus Stop	`0x88`	ACK

Continued on the next page.

Table 2. Commands to domes, continued

Continued from the previous page.

Command name	Description	Code	Response
S_FOCUS_FAR		136	
FOCUS STOP S_FOCUS_STOP	Stop Focus	0x89 137	ACK
ZOOM IN S_ZOOM_IN	Zoom in until Zoom Out or Zoom Stop (See Section 1.8.12.1, page 51)	0x8A 138	ACK
ZOOM OUT S_ZOOM_OUT	Zoom out until Zoom In or Zoom Stop (See Section 1.8.12.1, page 51)	0x8B 139	ACK
ZOOM STOP S_ZOOM_STOP	Stop zoom (See Section 1.8.12.1, page 51)	0x8C 140	ACK
FAST S_FAST	Increase pan and tilt speeds to 48^o/sec until Fast Stop (See Section 1.8.4.1, page 38 and Section 1.8.4.2, page 39)	0x8D 141	ACK
FASTEST S_FASTEST	Increase pan and tilt speeds to 96^o/sec until Fast Stop (See Section 1.8.4.1, page 38 and Section 1.8.4.2, page 39)	0x8E 142	ACK
FAST STOP S_FAST_STOP	Stop fast/fastest speeds (back to normal 24^o/sec) (See Section 1.8.4.2, page 39)	0x8F 143	ACK
IRIS OPEN S_IRIS_OPEN	Opens iris (manual iris mode/lightens iris Preference™ offset (auto iris mode) until Iris Close or Iris Stop (See Section 1.8.8, page 46)	0x90 144	ACK
IRIS CLOSE S_IRIS_CLOSE	Closes iris (manual iris mode/darkens iris Preference™ offset (auto iris mode) until Iris Open or Iris Stop	0x91 145	ACK
IRIS STOP S_IRIS_STOP	Stop iris offset adjustment (also stops V-Phase Adjust) (See Section 1.8.8, page 46)	0x92 146	ACK
ALL STOP	Stop all movement	0x93	ACK

Continued on the next page.

Table 2. Commands to domes, continued

Continued from the previous page.

Command name	Description	Code	Response
S_ALL_STOP		147	
DOME TYPE S_GET_DOME_TYPE	Request dome type (See Section 1.8.6, page 44)	0x94 148	Table 5, Page 27
—	Causes the dome type to be sent from the dome to the controller		Section 1.5.2, page 27
ALARM STATUS S_GET_ALARMS	Request status of alarm inputs (See Section 1.8.1, page 33)	0x95 149	Table 5, Page 27
—	Causes the alarm status to be sent from the dome to the controller		Section 1.5.2, page 27
SUNKNOWN 96 S_UNKNOWN96	**Unknown**	0x96 150	Table 5, Page 27
—	Causes information to be sent from the dome to the controller		Section 1.5.2, page 27
SEND ACK S_ACK	ACKnowledge sent to dome in responce to asynchronous commands. (No ACK.) (See Section 1.8.1, page 33)	0x97 151	—
START SUSPEND XMIT S_STARTSUSPEND	Start of temporary no transmit period. (No ACK.) (See Section 1.8.2.1, page 35, and Section 1.8.9.1, page 46)	0x98 152	—
END SYSPEND XMIT S_ENDSUSPEND	End of temporary no transmit period. (No ACK.) (See Section 1.8.9.1, page 46)	0x99 153	—
FASTER S_FASTER	Increase pan and tilt speeds to 72^o/sec until Fast Stop	0x9A 154	ACK
FASTER STOP S_FASTER_STOP	Stop faster speeds (back to normal 24^o/sec)	0x9B 155	ACK

Continued on the next page.

Table 2. Commands to domes, continued

Continued from the previous page.

Command name	Description	Code	Response
DEFINE BOUNDARY S_DEFINE_BOUNDARY	Start boundary definition. This command is followed by dome movement commands and tour Mark Boundary commands. (No ACK.)	0x9C 156	— Table 5, Page 27
—	Is also information sent from the dome to the controller		Section 1.5.2, page 27
MARK BOUNDARY S_MARK_BOUNDARY	Marks the current position as a boundary	0x9D 157	Table 5, Page 27
—	Is also information sent from the dome to the controller		Section 1.5.2, page 27
ON AIR S_ON_AIR	Set On Air status to tell the dome to send the asynchronous boundary crossing command (As shown in Table 4, Page 25)	0x9E 158	ACK
ON AIR RESET S_ON_AIR_RESET	Reset On Air status (See Section 1.8.4.2, page 39)	0x9F 159	ACK
DEFINE PATTERN 1 S_DEFINE1	Start defining Pattern 1 (See Section 1.8.12.1, page 51)	0xA0 160	ACK
DEFINE PATTERN 2 S_DEFINE2	Start defining Pattern 2	0xA1 161	ACK
DEFINE PATTERN 3 S_DEFINE3	Start defining Pattern 3	0xA2 162	ACK
NEW PATTERN S_NEW_PATTERN	Accept the new pattern as the current pattern and delete the old pattern. (See Section 1.8.12.1, page 51)	0xA3 163	ACK
DUMP DOME MEMORY S_DUMP_DOME_MEMORY	Dump dome memory, (See Section 1.8.9.2, page 48)	0xA4 164	Multiple bytes
—	Causes a memory dump to be sent from the dome to the controller		Section 1.5.2, page 28

Continued on the next page.

Table 2. Commands to domes, continued

Continued from the previous page.

Command name	Description	Code	Response
REQUEST POSITION S_GET_POSITION	Request Dome position Coordinates (See Section 1.8.9.1, page 46)	0xA5 165	Table 5, Page 28
—	Causes a 12 byte block of position/camera information to be sent from the dome to the controller		Section 1.5.2, page 28
GOTO POSITION S_GOTO_POSITION	Goto absolute position (**Multiple-byte format**, Table 1.4.2, page 20). Or for RC58 systems goto zone start, See capture record in Section 1.8.10, page 49).	0xA6 166	Table 3, Page 20
MARK TARGET 1 S_MARK1	Mark the current position as Target 1 (See Section 1.8.9.2, page 48)	0xA8 168	ACK
MARK TARGET 2 S_MARK2	Mark the current position as Target 2 (See Section 1.8.9.1, page 46)	0xA9 169	ACK
MARK TARGET 3 S_MARK3	Mark the current position as Target 3	0xAA 170	ACK
MARK TARGET 4 S_MARK4	Mark the current position as Target 4	0xAB 171	ACK
GOTO PATTERN 1 S_GOTO_PAT1	Go to the start of pattern 1 (See Section 1.14, page 77)	0xAC 172	ACK
GOTO PATTERN 2 S_GOTO_PAT2	Go to the start of pattern 2 (See Section 1.14, page 78)	0xAD 173	ACK
GOTO PATTERN 3 S_GOTO_PAT3	Go to the start of pattern 3 (See Section 1.14, page 78)	0xAE 174	ACK
GOTO PATTERN 4 S_GOTO_PAT4	Go to the start of pattern 4 (See Section 1.14, page 78)	0xAF 175	ACK
RUN PATTERN 1 S_RUN1	Run Pattern 1 (See Section 1.8.12.1, page 51)	0xB0 176	ACK
—	Is a boundary #1 crossing message which is sent from the dome to the controller		Section 1.5.1, page 25

Continued on the next page.

Table 2. Commands to domes, continued

Continued from the previous page.

Command name	Description	Code	Response
RUN PATTERN 2 `S_RUN2`	Run Pattern 2	`0xB1` 177	ACK
—	Is a boundary #2 crossing message which is sent from the dome to the controller		Section 1.5.1, page 25
RUN PATTERN 3 `S_RUN3`	Run Pattern 3	`0xB2` 178	ACK
—	Is a boundary #3 crossing message which is sent from the dome to the controller		Section 1.5.1, page 25
RUN NEW PATTERN `S_RUN_NEW`	Run a newly defined pattern to review it before accepting it to replace the previous pattern.	`0xB3` 179	ACK
—	Is a boundary #4 crossing message which is sent from the dome to the controller		Section 1.5.1, page 25
GO TO TARGET 1 `S_GOTO1`	Go to preset position called Target 1 Section 1.8.11.2, page 51	`0xB4` 180	ACK
—	Is a boundary confusion message which is sent from the dome to the controller		Section 1.5.1, page 26
GO TO TARGET 2 `S_GOTO2`	Go to preset position called Target 2	`0xB5` 181	ACK
—	Is a pattern done message which is sent from the dome to the controller		Section 1.5.1, page 26
GO TO TARGET 3 `S_GOTO3`	Go to preset position called Target 3	`0xB6` 182	ACK
GO TO TARGET 4 `S_GOTO4`	Go to preset position called Target 4	`0xB7` 183	ACK
PATTERN END	Tells the dome to stop recording (defining) a pattern	`0xB8`	ACK

Continued on the next page.

Table 2. Commands to domes, continued

Continued from the previous page.

Command name	Description	Code	Response
S_PATTERN_END	(See Section 1.8.12.1, page 51)	184	
MARK TARGET 5 S_MARK5	Mark the current position as Target 5	0xB9 185	ACK
MARK TARGET 6 S_MARK6	Mark the current position as Target 6	0xBA 186	ACK
MARK TARGET 7 S_MARK7	Mark the current position as Target 7	0xBB 187	ACK
GO TO TARGET 5 S_GOTO5	Go to preset position called Target 5	0xBC 188	ACK
GO TO TARGET 6 S_GOTO6	Go to preset position called Target 6	0xBD 189	ACK
GO TO TARGET 7 S_GOTO7	Go to preset position called Target 7	0xBE 190	ACK
VARIABLE SPEED S_VARIABLE_SPEED	Variable speed control (*New command.* **Multiple-byte format**, Table 1.4.2, page 20) Byte 3 = `0x81` Pan Left, `0x82` Pan Right, `0x84` Tilt Up, `0x85` TIlt Down. Byte 4 = Speed 1 → 100^o/sec.	0xC0 192	ACK
— —	— Is a dome power up message which is sent from the dome to the controller	0xC1	Section 1.5.1, page 26
PROPORTIONAL SPEED S_PROP_SPEED	Proportional speed pan or tilt movement commands (**Multiple-byte format**, Table 1.4.2, page 21)	0xC3 195	Table 1.4.2, page 21
GET CONFIG BUFFER	Allows access to the data stored in the configuration buffer. "Part of the power up sequence. Is sent by the dome to the controller." Is also sent from the controller to the dome to read out parts/all of the internal configuration buffer.	0xC4	ACK/- Section 1.13, page 73

Continued on the next page.

Table 2. Commands to domes, continued

Continued from the previous page.

Command name	Description	Code	Response
`S_GET_CONFIG_BUFFER`	(See Section 1.8.9.2, page 48 and Section 1.14, page 81) (For typical data sent back by a dome see Section 1.13, page 73.)	196	

SET CONFIG BUFFER
`S_SET_CONFIG_BUFFER`
	Allows the contents of the configuration buffer to be changed.
	New command, no data yet.
		`0xC5`
		197
			none

PEEL `S_PEEL`	Run default "Apple Peel" pattern for a spiral view of everything (only supported by SpeedDome, UltraDome IV and DeltaDome, or later products)	`0xC6` 198	ACK
SOFTWARE VERSION `S_SOFTWARE_VERSION`	Get software version number from dome (See Section 1.8.7, page 46)	`0xC9` 201	Table 5, Page 28
—	Causes a software configuration message to be sent from the dome to the controller		Section 1.5.2, page 27
— —	— Is an alarm message which is sent from the dome to the controller, indicates condition of alarm input 0.	`0xD0`	Section 1.5.1, page 26
— —	— Is an alarm message which is sent from the dome to the controller, indicates condition of alarm input 1.	`0xD1`	Section 1.5.1, page 26
— —	— Is an alarm message which is sent from the dome to the controller, indicates condition of alarm input 2.	`0xD2`	Section 1.5.1, page 26
— —	— Is an alarm message which is sent from the dome to the controller, indicates condition of alarm input 3.	`0xD3`	Section 1.5.1, page 26

Continued on the next page.

Table 2. Commands to domes, continued

Continued from the previous page.			
Command name	Description	Code	Response
OUTPUT S_OUTPUT0	Set output drivers Clears all active drivers (See Section 1.14, page 82)	0xEx 0xE0 224	ACK
S_OUTPUT1	Driver #1 (See Section 1.14, page 82)	0xE1 225	
S_OUTPUT2	Driver #2 (See Section 1.14, page 82)	0xE2 226	
S_OUTPUT3	Driver #3 (See Section 1.14, page 82)	0xE4 228	
S_OUTPUT4	Driver #4 (See Section 1.14, page 82)	0xE8 232	
TERMINATE PATTERN S_TERM_PAT	Terminate the currently running pattern	0xF0	Table 5, Page 28

1.4.2 Multiple-byte Format

Multiple byte Commands, part 1

Command Name	Description	Code	Dome Response
GOTO POSITION S_GOTO_POSITION	Go to absolute position Byte 1 Dome address Byte 2 Command Byte 3, 4 Pan position Byte 5, 6 Tilt position Byte 7, 8 Zoom position Byte 9, 10 Digital Zoom Byte 11 Iris Offset Byte 12 Electronic zoom limit Byte 13 Checksum Pan position is in steps, max step value = 16384 Other ranges are unknown.	0xA6	ACK
VARIABLE SPEED S_VARIABLE_SPEED	Variable speed pan or tilt movement commands Byte 1 Dome address Byte 2 command 0xC0 Byte 3 S_PROP_LEFT 0x81 = pan left S_PROP_RIGHT 0x82 = pan right S_PROP_UP 0x84 = tilt up S_PROP_DOWN 0x85 = tilt down Byte 4 Speed $1 \rightarrow 100_{10} = 1 \rightarrow 100_{10}{}^{o}/\mathrm{sec}$ The eight observed values, per axis, with the rather limited AD2083/02 are: Pan: 0x04, 0x06, 0x0A, 0x0F, 0x18, 0x21, 0x2D, 0x5A Tilt: 0x03, 0x05, 0x09, 0x0D, 0x14, 0x18, 0x21, 0x2D The RC216 has at least 30 different values of variable speeds that it uses. See Section E, page E-1 for a partial listing. Byte 5 Checksum	0xC0	ACK

Multiple byte Commands, part 2

Command Name	Description	Code	Dome Response
SUNKNOWN C1	Length is 12, usage is unknown `0x06 0xC1 0x12`? `0x06 0xE8 0x12`? `0x06 0xE8 0x12`? `0x06 0xC1 0x39`?	`0xC1`	—
PROPORTIONAL SPEED S_PROP_SPEED	Proportional speed pan or tilt movement commands Byte 1 Dome address Byte 2 `S_PROP_LEFT` `0x81` = pan left `S_PROP_RIGHT` `0x82` = pan right `S_PROP_UP` `0x84` = tilt up `S_PROP_DOWN` `0x85` = tilt down Byte 3 Speed $1 \rightarrow 100_{10} = 1 \rightarrow 100_{10}{}^{o}$/sec Byte 4 Checksum	`0xC3`	ACK
GET CONFIGURATION BUFFER S_GET_CONFIG_BUFFER	Read the contents of the configuration buffer Byte 1 Dome address Byte 2 `S_GET_CONFIG_BUFFER` `0xC4` Byte 3 Starting byte Byte 4 Number of bytes Byte 5 Checksum	`0xC4`	ACK

1.4.3 Combination Commands

These are standard three byte commands.

The following commands are generated by pressing a combination of keys on the video manager controller, in some systems, a single key or menu selection will cause the controller to send the combination commands.

1.4.3.1 Reset Auto Iris and Resume Auto Focus

Simultaneously press and hold IRIS OPEN and IRIS CLOSE keys. This restores the iris setpoint to the factory setting, and returns to Auto Focus after a manual focus operation.

(*This sequence of commands may not be generated with an AD2050—AD2078—AD2083/02 set of hardware, if you are using the keyboard buttons. When* IRIS OPEN *is depressed then depressing* IRIS CLOSE *results in no action. However note that with the AD2083/02 this sequence is generated for both* RESET IRIS *and* RETURN TO AUTO IRIS/AUTO FOCUS *command actions.*)

An example is shown in Section 1.14, page 81.

IRIS OPEN	addr	`0x90`, `S_IRIS_OPEN`	cksm
IRIS CLOSE	addr	`0x91`, `S_IRIS_CLOSE`	cksm
IRIS STOP	addr	`0x92`, `S_IRIS_STOP`	cksm

1.4.3.2 Increase V-Phase Delay

Press and hold the FASTER key and press the IRIS OPEN key (The IRIS STOP command (Table 2, Page 11 through Page 19) stops the Phase increase.)

FASTER	addr	`0x9A`, `S_FASTER`	cksm
IRIS OPEN	addr	`0x90`, `S_IRIS_OPEN`	cksm

V-Phase will slowly increase delay until ...

IRIS STOP	addr	`0x92`, `S_IRIS_STOP`	cksm
FASTER STOP	addr	`0x9B`, `S_FASTER_STOP`	cksm

(An alternate to this is offered for AD2083/02 units in Section 1.14, page 81 and Section 1.14, page 81 which supports a different method of changing V-Phasing. In the test case, I only used dome #4. It might work with other domes too.)

1.4.3.3 Decrease V-Phase Delay

Press and hold the FASTER key and press the IRIS CLOSE key (The IRIS STOP command (Table 2, Page 11 through Page 19) stops the Phase decrease.)

FASTER	addr	`0x9A`, `S_FASTER`	cksm
IRIS CLOSE	addr	`0x91`, `S_IRIS_CLOSE`	cksm

V-Phase will slowly decrease delay until ...

IRIS STOP	addr	`0x92`, `S_IRIS_STOP`	cksm
FASTER STOP	addr	`0x9B`, `S_FASTER_STOP`	cksm

(An alternate to this is offered for AD2083/02 units in Section 1.14, page 81 and Section 1.14, page 81 which supports a different method of changing V-Phasing. In the test case, I only used dome #4. It might work with other domes too.)

1.4.3.4 Clear pattern and reset to default "Apple Peel" pattern

With the SpeedDome an "Apple Peel" pattern consists of:

1. Opening the lens up to maximum wide angle.
2. Moving the lens until it is horizontal.
3. Doing three revs of the dome, after each full rev move the camera in tilt about $\frac{1}{3}$ the distance from horizontal to vertical (about 30^o).
4. Then repeat starting at step 2 above until manually stopped.

DEFINE PATTERN 1	addr	0xA0, S_DEFINE1	cksm
		or	
DEFINE PATTERN 2		0xA1, S_DEFINE2	cksm
		or	
DEFINE PATTERN 3		0xA2, S_DEFINE3	cksm
FASTEST	addr	0x8E, S_FASTEST	cksm
PATTERN END	addr	0xB8, S_PATTERN_END	cksm
NEW PATTERN	addr	0xA3, S_NEW_PATTERN	cksm
FAST STOP	addr	0x8F, S_FAST_STOP	cksm

Examples of this are shown in Section 1.14, page 79, Section 1.14, page 79, Section 1.14, page 80 and Section 1.8.13.3, page 62.

1.4.3.5 Pan 180° from Current Position "Flip"

1.4.3.5.1 With an AD2083/02 translator

Press and hold the FAST key and press the FASTEST key. This is done automatically when 40 CALL is entered from the keyboard. (40 CALL is the way American Dynamics initiates a "goto preset" operation.)

FAST	addr	0x8D, S_FAST	cksm
FASTEST	addr	0x8E, S_FASTEST	cksm
FAST STOP	addr	0x8F, S_FAST_STOP	cksm

See Section 1.8.4.1, page 38 for an example.

1.4.3.5.2 With an RC58 type controller

With SensorVision^{TM} type systems: press and hold the RAIL LEFT key and press the RAIL RIGHT key. **The order is important!**

RAIL LEFT	addr	0x8D, S_FAST	cksm
RAIL RIGHT	addr	0x8E, S_FASTEST	cksm

See Section 1.8.4.2, page 39 for an example.

1.4.3.5.3 With an RC216 type controller

With a TouchTracker select a camera and hit the "D" key.

FOCUS STOP	addr	0x89, S_FOCUS_STOP	cksm
RAIL LEFT	addr	0x8D, S_FAST	cksm
RAIL RIGHT	addr	0x8E, S_FASTEST	cksm

See Section 1.8.4.3, page 39 for an example.

1.4.3.6 Repeat Pattern

Press and hold the FASTER key and initiate a pattern. The pattern repeats until cancelled by any movement command.

FASTER	addr	0x9A, S_FASTER	cksm
RUN PATTERN	addr	0xBx, S_RUN1 → 3	cksm
x = 0 → 2 (0 = pat 1)			
FASTER STOP	addr	0x9B, S_FASTERSTOP	cksm

With the AD2083/02 the sequence is slightly different. See Section 1.14, page 78, Section 1.14, page 78, Section 1.14, page 78 or Section 1.14, page 78 for examples. The primary difference is that the AD2083/02 sends an undocumented command of 0xAC (0xAD, 0xAE or 0xAF) in addition to these commands. The AD2083/02 also supports four patterns.

With the RC216 there are other changes to this sequence of commands. See Section 1.8.13.2, page 62 for an example.

1.4.3.7 Reset the Dome

Press and hold the FASTER key and, in sequence, press and hold the ZOOM OUT, FOCUS FAR and IRIS OPEN keys.

FASTER	addr	0x9A, S_FASTER	cksm
ZOOM OUT	addr	0x8B, S_ZOOM_OUT	cksm
FOCUS FAR	addr	0x88, S_FOCUS_FAR	cksm
IRIS OPEN	addr	0x90, S_IRIS_OPEN	cksm

An example of this done with an AD2083/02 and dome #4 is shown in Section 1.14, page 81. In this example is also shown the communications between the dome and the controller following a reset. I.e. the dome goes through its full power up sequence.

1.4.3.8 Start on-screen menu

Start on-screen configuration mode and display the menu on the video display. (*Does not work on domes #2 and #5. Does work on dome #4.*)

IRIS OPEN	addr	0x90, S_IRIS_OPEN	cksm
FOCUS NEAR	addr	0x87, S_FOCUS_NEAR	cksm
		or	
FOCUS FAR		0x88, S_FOCUS_FAR	cksm
ZOOM OUT	addr	0x8B, S_ZOOM_OUT	cksm

See Section 1.8.3.1, page 37 for an example using an AD2083/02 and dome 4.

1.4.3.9 Exit the on-screen menu

Quick exit the on-screen configuration menu mode and save the changes.

IRIS OPEN	addr	0x90, S_IRIS_OPEN	cksm
FOCUS NEAR	addr	0x87, S_FOCUS_NEAR	cksm
		or	
FOCUS FAR		0x88, S_FOCUS_FAR	cksm

In Section 1.8.3.1, page 37 for an example that is different from this which is using an AD2083/02 and dome 4.

1.5 Dome to Controller Communications

1.5.1 Asynchronous and Dome Response Messages

Asynchronous messages are in standard three byte format.

When appropriate, the dome sends an asynchronous message listed in Table 4, Page 25 or when requested sends a response message listed in Table 5, Page 26.

Upon power-up or reset, the dome **On Air status** is reset, it can be set or reset with the ON AIR or ON AIR RESET command. This status is used for the BOUNDARY CROSSING messages.

Response messages have a multiple byte length format. Each response message length varies from four to 12 bytes in length. Each of these messages starts with an address and ends with a checksum. None indicates the message type in byte 2.

Table 4. Asynchronous Dome Messages

Response Code	Description of Dome Asynchronous Messages
0xBx	BOUNDARY CROSSING (sent by dome if pan boundary was crossed and "on-the-air" status is true (being watched). 0xBx: x is low nibble S_BOUNDARY0CROSSED, 1, Boundary crossing #1 0xB0 S_BOUNDARY1CROSSED, 2, Boundary crossing #2 0xB1 S_BOUNDARY2CROSSED, 3, Boundary crossing #3 0xB2 S_BOUNDARY3CROSSED, 4, Boundary crossing #4 0xB3 Dome expects an ACK from the controller (0x97/151_{10}) or dome tries twice more. See Section 1.10.1, page 64

Continued on the next page.

Table 4. Asynchronous Dome Messages, continued

Response Code	Description of Dome Asynchronous Messages
0xB4 S_BOUNDARY_CONFUSION	BOUNDARY CONFUSION (sent by dome if problem defining boundaries)
0xB5 S_PATTERN_DONE	PATTERN DONE (sent by dome when it completes a pattern); dome expects an ACK from controller (0x97/151_{10}) or dome tries twice more. See Section 1.8.12.2, page 59
0xC1 S_POWERED_UP	DOME POWERED UP (sent by dome to indicate it has powered up and is on line) See Section 1.8.1, page 33, Section 1.14, page 81
0xDx	DOME ALARM (sent by dome if its input switches change state): 0xDx: x is the low nibble S_DOME_ALARM0, Bit 0, Switch 1 (0 = on, 1 = off) 0xD0 S_DOME_ALARM1, Bit 1, Switch 2 (0 = on, 1 = off) 0xD1 S_DOME_ALARM2, Bit 2, Switch 3 (0 = on, 1 = off) 0xD2 S_DOME_ALARM3, Bit 3, Switch 4 (0 = on, 1 = off) 0xD3 Dome expects an ACK from the controller (0x97/151_{10}) or dome tries twice more.

1.5.2 Dome Response Messages

As shown in Table 2, Page 11 through Page 19, some controller commands are responded to with an ACK followed by data. Table 5, Page 26 lists these data responses and provides examples. They are in similar format to the other commands.

Table 5. Dome Response Messages

Command Description	Code Sent	Data Received
SUNKNOWN 80	0x80	0xC4 UNKNOWN RESPONSE. Is sent from the dome to the controller during power up when the dome receives an 0x80 command. (See Section 1.8.9.2, page 48 and Section 1.14, page 81.)

Continued on the next page.

Table 5. Dome Response Messages, continued

Command	Code	Data
Description	Sent	Received
S_UNKNOWN80		Byte 1, Dome address Byte 2, Unknown use, 0xC4 Byte 3, Checksum
REQUEST DOME TYPE	0x94	0xEx, or 0xFx Changes from E to F following receiving a preset command.
S_GET_DOME_TYPE		x = 5 or 8
		Byte 1, Dome address Byte 2, Dome type, 0xF5 Byte 3, Checksum
REQUEST ALARM STATUS S_GET_ALARMS	0x95	0x: x = low nibble Byte 1, Dome address Byte 2, Active alarms, 0x00 Byte 3, Checksum
		bit 0, Alarm input 1 bit 1, Alarm input 2 bit 2, Alarm input 3 bit 3, Alarm input 4
		1 = Off/Open 0 = On/Closed
UNKNOWN 96	0x96	0xC0
S_UNKNOWN_96		Byte 1, Dome address Byte 2, Dome type, 0xC0 Byte 3, Checksum
UNKNOWN 9C	0x9C	0xC4
S_UNKNOWN_9C		Byte 1, Dome address Byte 2, Dome type, 0xC4 Byte 3, Checksum
MARK BOUNDRY	0x9D	0xBx
S_MARK_BOUNDARY		Boundry ID from Table 4, 25. Including "S_BOUNDARY_CONFUSION". Byte 1, Dome address Byte 2, Dome type, 0xBx

Continued on the next page.

Table 5. Dome Response Messages, continued

Command Description	Code Sent	Data Received
		Byte 3, Checksum
DUMP DOME MEMORY	0xA4	A whole bunch of data, several hundred bytes or more. Format is unknown.
S_DUMP_DOME_MEMORY		Byte 1, Dome address Byte 2 — N, Data Byte $N + 1$, Checksum
REQUEST DOME POSITION S_GET_POSITION	0xA5	LTZZEEPP Byte 1, Dome address Byte 2, Iris xuning, 0x80 Byte 3, Zoom Limit, 0x35 Byte 4, 5, Tilt position, 0x41, 0x33 Byte 6, 7, Zoom position, 0x10, 0xDC Byte 8, 9, Electronic zoom, 0x00, 0x00 Byte 10, 11, Pan position, 0x6C, 0xB4 Byte 12, Checksum
REQUEST SOFTWARE VERSION # S_SOFTWARE_VERSION	0xC9	AA AA BB BB CC CC AAAA-BBBB-CCCC Example: 0x01, 0x08, 0x06, 0x07, 0x01, 0x25, 0x07, 0x02, 0x04, 0xF6 Sensormatic Part number assigned to the executable code.
		See Section 1.8.7, page 46
TERMINATE PATTERN	0xF0	0xB4.
S_TERM_PAT		Byte 1, Dome address Byte 2, 0xB4 Byte 3, Checksum

1.6 Dome Configuration Menu

The dome configuration utility provides a text overlay menu for setting the camera dome features. The utility provides settings relating to camera functions, alarms, text display, privacy zones, directional indicators, and password protection. Some items supplement similar features that may be available through the controller. Access the menu by a keystroke combination on the camera controller.

To start the dome configuration utility, do the following:

1. Select the dome that you need to configured. Refer to the controller operation instructions for specific information.

2. Press and hold IRIS OPEN (S_IRIS_OPEN), press and hold a FOCUS button (NEAR (S_FOCUS_-

NEAR) or FAR (S_FOCUS_FAR)), then press ZOOM OUT (S_ZOOM_OUT) (Zoom Wide). (See Section 1.4.3.8, page 24)

See Section 1.8.3.1, page 37 for an example of this procedure.

The following type of menu appears on the monitor: (this one is from a SpeedDome, UltraDome.)

```
  DOME CONFIGURATION MENU

 PAN/TILT/ZOOM/SHUTTER/OPTS
   CAMERA / LENS FUNCTIONS
 ALARMS / AREAS / HOME / PZ
   ON-SCREEN TEXT DISPLAY
    LANGUAGE / PASSWORD

  RESET TO FACTORY SETTINGS
    QUIT WITHOUT SAVING
   EXIT AND SAVE CHANGES
```

1.6.1 Working with the Dome Configuration Utility

Once the Dome Configuration Menu is displayed, you can select a menu item, then modify the settings you want to change. The controls used with the utility are Pan/Tilt (Tracker Ball or Joystick), FOCUS NEAR, FOCUS FAR, ZOOM IN (Zoom Tele), ZOOM OUT (Zoom Wide), IRIS OPEN, and IRIS CLOSE. For combination keystrokes, press and hold each button in sequence, then release it.

The following table summarizes the controller commands used with the configuration utility.

If you want to ...	Use ...
Start the configuration utility (See Section 1.4.3.8, page 24)	IRIS OPEN (S_IRIS_OPEN), FOCUS NEAR (S_FOCUS_FAR), or FOCUS FAR (S_FOCUS_NEAR), then ZOOM OUT (S_ZOOM_OUT)
Move the highlight bar	Pan/Tilt
Select the highlighted item on the screen.	FOCUS FAR (S_FOCUS_FAR)
Increase the value of the selected setting or displays the sent choice for the setting.	ZOOM IN (S_ZOOM_IN)
Decrease the value of the selected setting or displays the sent choice for the setting.	ZOOM OUT (S_ZOOM_OUT)
During naming, move the cursor to the right of the current character in the name	ZOOM IN (S_ZOOM_IN)
During naming, move the cursor to the left of the current character in the name	ZOOM OUT (S_ZOOM_OUT)
Save changes and exit the utility from any screen (See Section 1.4.3.9, page 25)	IRIS CLOSE (S_IRIS_CLOSE), then FOCUS (S_FOCUS_FAR or S_FOCUS_NEAR)

Note

Where no specific FOCUS button is listed either FOCUS NEAR (S_FOCUS_NEAR) or FOCUS FAR (S_FOCUS_FAR) may be used.

Where no specific ZOOM button is listed, either a ZOOM IN (S_ZOOM_IN) or ZOOM OUT (S_ZOOM_OUT) command may be used.

The functions are performed by the appropriate commands shown in Table 2, Page 11 through 19 and Table 3, Page 20.

1.7 Boundaries and Patterns

Definitions

Area	Programmed start and end points of the dome's field of view around its pan axis. each area is a part of a circular viewing area that extends around the dome. The areas can be different sizes.
Boundary	A separation between areas.
Pattern	A sequential series of pan, tilt, zoom, and focus movements from a single programmable dome. You "teach" the dome a combination of these movements that can be replayed automatically.

Boundaries and Patterns are enhanced features and are not required for basic camera and dome operations.

1.7.1 Boundaries

The DEFINE BOUNDARY, MARK BOUNDARY, ON AIR, and ON AIR RESET commands (`0x9C` through `0x9F` (`S_DEFINE_BOUNDARY`, `S_MARK_BOUNDARY`, `S_ON_AIR` and `S_ON_AIR_RESET`)) support the Boundary feature of the SpeedDome, UltraDome and DeltaDome series of domes. They are not necessary if this feature is not used, or if the internal boundary feature supported by the dome configuration the Utility Menu is used instead of the external controller text overlay boundary feature.

To establish the boundaries the DEFINE BOUNDARY command (`0x9C` (`S_DEFINE_BOUNDARY`)) is sent to the dome, followed by movement commands to position the dome at the boundaries. The MARK BOUNDARY command (`0x9C` (`S_MARK_BOUNDARY`)) is sent when the dome is pointing at each of the four boundaries. To change a boundary positions the process is started over with the DEFINE BOUNDARY command.

When the dome's On Air status (`0x9E` (`S_ON_AIR`)) is set, it will asynchronously transmit the BOUNDARY CROSSING response (see Table 4, Page 25) when a boundary is crossed.

1.7.2 Patterns

You can define up to three patterns for each dome that can be stored into the dome's memory. Patterns are established by first sending the DEFINE PATTERN command (`0xA0` (`S_DEFINE1`), `0xA1` (`S_DEFINE2`) and `0xA2` (`S_DEFINE3`)) to the dome. This puts the dome into a mode where it will record the movement commands.

Note: To maintain reproducible pattern accuracy, only 3-byte non-proportional commands can be used.

A dome can store up to a total of 99 movement commands in its memory for the three patterns. When defining a pattern, the on-screen display of the dome shows the remaining available memory.

Initially all patterns default to the "Apple Peel" or spiral pattern. This pattern provides a view of the entire viewable area of the dome. This default pattern does not use dome memory space.

- To indicate that a pattern recording is finished, a PATTERN END command (`0xB8` (`S_PATTERN_END`)) is sent to the dome. To review the new pattern, the RUN NEW PATTERN command (`0xB3` (`S_RUN_NEW`)) is sent to the dome. To replace the previous pattern with the new pattern, the NEW PATTERN command (`0xA3` (`S_NEW_PATTERN`)) is sent to the dome.

- To play patterns the RUN PATTERN commands (`0xB0` (`S_RUN1`) through `0xB2` (`S_RUN3`)) are sent to the dome.

- To clear a pattern and return to the default "Apple Peel" command, send the CLEAR PATTERN combination command (`0xA0` (`S_DEFINE1`), `0xA1` (`S_DEFINE2`) or `0xA2` (`S_DEFINE3`)) (`0x8E` (`S_FASTEST`)) (`0xB8` (`S_PATTERN_END`)) (`0xA3` (`S_NEW_PATTERN`)) (`0x8F` (`S_FAST_STOP`)).

1.8 Typical dome/controller conversations

Much of the following command sequences are from dome #1 which is set for address `6`. Most of the protocol information was collected at Pelco using an AD2083/02 and came from a Breakout dump. The camera/dome has the following characteristics:

It is a working single circuit board type of color camera dome that does not display any configuration information on power up.

Manufacturer	`Sensormatic`
Model #	`SpeedDome 2000 Outdoor`
Reg M/N	`SU SD`
Part of	`0100-0760-05`
Rev.	`A0`
Serial #	`1507495`
Mfd.	`4QTR95`
Bar Code	`*1507495*`
Paper Label	`RAS586LS`
2^{nd} Paper Label	`RAS586LS REV. C`
	`SC 04/28/99 T.33`

It has three EPROMs: (U4's part number is what gets returned when a software revision request is made.)

U4	`0701-0081`
	`U4`
	`VER 0.316`

U8	`0701-0077`
	`U8`
	`VER 03.12`

U50	`3133-0013-01`
	`U50`
	`VER 1.00`

In reading the hexadecimal log of each command sequence in the samples that follow. It should be remembered that each command byte is preceded with either a "`1`", for input data or a "`0`", for output data. Input and output are referred to the controller (or head end), i.e. input data comes from a dome, output data goes to a dome.

There appears to be something "special" associated with dome address 64_{10} (40_{16}). Several special messages are sent to this address during a command sequence to set a preset when using an AD2083/02. The unknown command types are `0x98` and `0x99`. The manual for the AD2083/02 indicates that address 64_{10} (40_{16}) is "not used". (See Section 1.8.9.1, page 46 for an example of setting a preset with an AD2083/02.) By analyzing, and guessing a lot, it appears that the unknown commands `0x98` and `0x99` are used as a "global stop transmission" (`0x98`) and "global restart transmission" (`0x99`) commands. They are normally sent just before, and after, the controller requests information from a dome. Part of this logic is that I have never detected a "BOUNDARY CROSSING" message from a dome during the time that global messages are disabled.

All types of controllers studied so far "POLL" domes constantly. Some do it twice a second others once a second. As these POLL messages, and their responses, have little meaning on the operation of the protocol, they are usually deleted.

In each extracted section of protocol conversation, I have attempted to indicate the approximate timing of various events. As I don't use the timer in Breakout, I have had to use the one second (or one half second) "tick" provided by the POLL messages. Thus the first five characters on each line are the time tag of that event in `MM.SS` format. (I had to limit its width so that the line would not be too long.) The second field is the event number modulo 1000, again so that the line will not get too wide.

1.8.1 Typical dome power up sequence

1.8.1.1 With an 2083/02A controller

This is from dome #5 when connected to an AD2083/02A. The AD2083/02A is powered up and this is a record of what happens when the dome is turned on.

1. In line 2 the dome indicates that it has just powered up.
2. In lines 3 → 4 the controller and dome then ACK each other.
3. In lines 5 → 7 two undocumented messages/commands are exchanged which is followed by an ACK from the controller.
4. In line 8 an unexpected message is generated which is probably an indication as to which alarms (input points) are active.
5. In line 9 is the request for the alarm information.

```
1 <30JAN02A>
2 06:53,   841: 1-05 1-C1 1-3A          < 5: Dome Power Up Message>
3 06:53,   842: 0-05 0-97 0-64          < 5: ACK to dome>
4 06:53,   843: 0-05 0-80 0-7B          < 5: Unknown 80>
5 06:53,   844: 1-05                    < 5: ACK>
6 06:54,   846: 1-05 1-C4 1-37          < 5: Unknown Response C4>
7 06:54,   847: 0-05 0-97 0-64          < 5: ACK to dome>
8 06:54,   848: 0-05 0-95 0-66          < 5: Request Alarm Status>
9 06:54,   849: 1-05 1-00 1-FB          < 5: Dome Alarm status>
```

1.8.1.2 With an RC216 controller at Sears Clovis

Note with a "real" RC216, i.e. one with multiple cameras installed and at a currently working location, how the power up sequence changes.

Notes

1. In line 2 the dome sends a power up message of `0xC1`.
2. In line 3 the controller responds with a dome acknowledge of `0x97`.
3. Then in lines 4 → 9 the controller continues polling.
4. In lines 10, 11 the controller polls address 44 (`0x2C`) and gets a response with a type code of `0xF5`.
5. For lines 12 → 23 the controller continues on on its own.
6. In line 24 the controller sends out a poll which is out of sequence, i.e. it should be to address 56 but is to the newly powered up dome at address 44.
7. In line 25 the dome acknowledges the poll.
8. In line 26 the controller requests the dome's software rev code.
9. The software rev is returned on line 28.
10. Next an ON AIR (`0x9E`) command is sent to the dome and the dome replies in line with its current boundary ID. (`0xB0`).
11. Now in line 31 the dome is placed OFF AIR, to which the dome sends an ACK in line 32.
12. As a last step in the power up sequence, the controller requests the status of the alarms (0x95) to which the dome replies that no alarms are active (`0x00`).

```
 1 <07FEB02A.CAP>
 2 09:31,  2091: 1-2C 1-C1 1-13          <44: Dome Power Up Message>
 3 09:31,  2092: 0-2C 0-97 0-3D          <44: ACK to dome>
 4 09:31,  2093: 0-26 0-94 0-46          <38: Poll, Dome type query>
 5 09:31,  2094: 0-27 0-94 0-45          <39: Poll, Dome type query>
 6 09:32,  2095: 0-28 0-94 0-44          <40: Poll, Dome type query>
 7 09:32,  2096: 0-29 0-94 0-43          <41: Poll, Dome type query>
 8 09:33,  2097: 0-2A 0-94 0-42          <42: Poll, Dome type query>
 9 09:33,  2098: 0-2B 0-94 0-41          <43: Poll, Dome type query>
10 09:34,  2099: 0-2C 0-94 0-40          <44: Poll, Dome type query>
11 09:34,  2100: 1-2C 1-F5 1-DF          <44: Dome Query Response>
12 09:34,  2101: 0-2D 0-94 0-3F          <45: Poll, Dome type query>
13 09:35,  2102: 0-2E 0-94 0-3E          <46: Poll, Dome type query>
14 09:35,  2103: 0-2F 0-94 0-3D          <47: Poll, Dome type query>
15 09:36,  2104: 0-30 0-94 0-3C          <48: Poll, Dome type query>
16 09:36,  2105: 0-0C 0-97 0-5D          <12: ACK to dome>
17 09:36,  2106: 0-31 0-94 0-3B          <49: Poll, Dome type query>
18 09:37,  2107: 0-32 0-94 0-3A          <50: Poll, Dome type query>
19 09:37,  2108: 0-33 0-94 0-39          <51: Poll, Dome type query>
20 09:38,  2109: 0-34 0-94 0-38          <52: Poll, Dome type query>
21 09:38,  2110: 0-35 0-94 0-37          <53: Poll, Dome type query>
22 09:39,  2111: 0-36 0-94 0-36          <54: Poll, Dome type query>
23 09:39,  2112: 0-37 0-94 0-35          <55: Poll, Dome type query>
24 09:40,  2113: 0-2C 0-94 0-40          <44: Poll, Dome type query>
25 09:40,  2114: 1-2C 1-F5 1-DF          <44: Dome Query Response>
26 09:40,  2115: 0-2C 0-C9 0-0B          <44: Get software ID>
27 09:40,  2116: 1-2C                    <44: ACK>
```

```
09:40,  2117: 1-2C 1-C9 1-06 1-07 1-01 1-00 1-01 1-03 1-16 1-E3 <44: Software ID Response>
09:40,  2118: 0-2C 0-9E 0-36          <44: On Air>
09:40,  2119: 1-2C 1-B0 1-24          <44: Boundary Crossing 1>
09:40,  2120: 0-2C 0-9F 0-35          <44: Reset On Air>
09:40,  2121: 1-2C                    <44: ACK>
09:40,  2122: 0-2C 0-95 0-3F          <44: Request Alarm Status>
09:40,  2123: 1-2C 1-00 1-D4          <44: Dome Alarm status>
```

1.8.2 Typical controller power up sequences

1.8.2.1 Typical RC216 power up sequence with no domes attached

When powered up, the RC216 waits for four minutes and then sends the below sequence of ON AIR and OFF AIR commands to camera #1. Following a short delay of 2 → 4 seconds it then sends a typical full block of "POLL"s to cameras 1 → 96_{10}. If camera #1 isn't available then it starts to send a POLL command out every $\frac{1}{2}$ second. (It is unclear what happens if camera #1 is there.)

Note that during the initial block of "POLL"s that each command is sent three times, where as the AD2083/02 only sends each command out once. And also note that when the initial POLLing is completed, that camera #1 is skipped and that $\frac{1}{2}$ second POLLing starts at camera id #2. All following $\frac{1}{2}$ second POLLing sequences start a one and continue on to 96_{10}.

There are no "skipped" camera IDs, i.e. it POLLs camera address 64_{10} even though 64_{10} is a reserved camera ID. (The VM96 consists of several items, the actual controller is an RC216H.)

```
08:52, 590: 0-01 0-9E 0-61          < 1: On Air>
08:52, 591: 0-01 0-9E 0-61          < 1: On Air>
08:52, 592: 0-01 0-9E 0-61          < 1: On Air>
08:52, 593: 0-01 0-9F 0-60          < 1: Reset On Air>
08:52, 594: 0-01 0-9F 0-60          < 1: Reset On Air>
08:52, 595: 0-01 0-9F 0-60          < 1: Reset On Air>
08:52, 596: 0-40 0-98 0-28          <64: Start Temp No Xmit>
08:52, 597: 0-40 0-98 0-28          <64: Start Temp No Xmit>
08:52, 598: 0-40 0-98 0-28          <64: Start Temp No Xmit>

08:52, 600: 0-01 0-94 0-6B          < 1: Poll, Dome type query>
08:52, 601: 0-01 0-94 0-6B          < 1: Poll, Dome type query>
08:52, 602: 0-01 0-94 0-6B          < 1: Poll, Dome type query>
08:52, 603: 0-02 0-94 0-6A          < 2: Poll, Dome type query>
08:53, 604: 0-02 0-94 0-6A          < 2: Poll, Dome type query>
08:53, 605: 0-02 0-94 0-6A          < 2: Poll, Dome type query>
<280 deleted poll commands>
09:39, 885: 0-60 0-94 0-0C          <96: Poll, Dome type query>
09:40, 886: 0-60 0-94 0-0C          <96: Poll, Dome type query>
09:40, 887: 0-60 0-94 0-0C          <96: Poll, Dome type query>
09:40, 888: 0-02 0-94 0-6A          < 2: Poll, Dome type query>
09:40, 889: 0-03 0-94 0-69          < 3: Poll, Dome type query>
09:41, 890: 0-04 0-94 0-68          < 4: Poll, Dome type query>
```

1.8.2.2 Typical RC216 power up sequence with domes attached

```
<08JUN01D.CAP>
00:28, 362: 0-02 0-94 0-6A           < 2: Poll, Dome type query>
00:28, 363: 1-02 1-F5 1-09           < 2: Dome Query Response>
00:28, 364: 0-03 0-94 0-69           < 3: Poll, Dome type query>
00:29, 365: 1-03 1-F5 1-08           < 3: Dome Query Response>

00:37, 398: 0-14 0-94 0-58           <20: Poll, Dome type query>
00:37, 399: 0-14 0-94 0-58           <20: Poll, Dome type query>
00:37, 400: 0-14 0-94 0-58           <20: Poll, Dome type query>
00:37, 401: 0-15 0-94 0-57           <21: Poll, Dome type query>
00:38, 402: 0-15 0-94 0-57           <21: Poll, Dome type query>
00:38, 403: 0-15 0-94 0-57           <21: Poll, Dome type query>

00:49, 472: 1-0E 1-B3 1-3F           <14: Boundry Crossing 4>
01:15, 641: 0-0E 0-97 0-5B           <14: ACK to dome>

01:15, 642: 0-02 0-94 0-6A           < 2: Poll, Dome type query>
01:16, 643: 1-02 1-F5 1-09           < 2: Dome Query Response>
01:16, 644: 1-0C 1-B2 1-42           <12: Boundry Crossing 3>
01:16, 645: 0-0C 0-97 0-5D           <12: ACK to dome>

01:17, 655: 0-02 0-94 0-6A           < 2: Poll, Dome type query>
01:17, 656: 1-02 1-F5 1-09           < 2: Dome Query Response>
01:17, 657: 0-02 0-C9 0-35           < 2: Get software ID>
01:17, 658: 1-02 1-C9 1-06 1-07 1-01 1-25 1-07 1-02 1-04 1-F5 < 2: Software ID Response>
01:17, 659: 0-02 0-9E 0-60           < 2: On Air>
01:17, 660: 1-02                     < 2: ACK>
01:17, 661: 0-02 0-9F 0-5F           < 2: Reset On Air>
01:17, 662: 1-02                     < 2: ACK>
01:17, 664: 0-02 0-95 0-69           < 2: Request Alarm Status>
01:17, 665: 1-02 1-0F 1-EF           < 2: Dome Alarm status>
01:17, 667: 0-02 0-9E 0-60           < 2: On Air>
01:17, 668: 1-02                     < 2: ACK>
```

1.8.2.3 Typical AD2083/02 power up sequence with domes attached

When powered up, the AD2083/02 sends a full block of "POLL"s to cameras 1 → 99_{10}. It then starts to send a POLL command out every second. There are no "skipped" camera IDs, i.e. it POLLs camera address 64_{10} even though 64_{10} is a reserved camera ID. (As each POLL command is sent once, starting at one and continuing on to 99_{10}, I couldn't see anything to be gained by listing it out. So I didn't.)

1.8.2.4 Typical RC58 power up sequence

Unknown. The only one of these that I have encountered is on a customer site and I have not yet powered it down and back up.

1.8.3 Typical dome menu sequence

1.8.3.1 From an AD2083/02 translator

Note that all of these commands are "normal", it's just the order that they are given in that makes them special. And then too it is important to note that the first steps for starting menu mode, involve holding three buttons down **simultaneously**.

Notes

1. In lines 2 → 12 menu mode is accessed. Here note that three buttons are depressed simultaneously and that no "STOP" commands are sent during the sequence. It appears that the order of releasing the buttons has no effect, as long at the order in depressing, and holding them, is maintained.
2. In lines 13 → 36 the menu bar is moved down a couple of lines. Eventually it gets to the exit line. (Several "down" movements have been omitted here.)
3. In lines 37 → 38 is a POLL and its response.
4. In lines 39 → 42 the exit item is selected.
5. In lines 43 → 58 a sub-menu appears here and the next option is selected.
6. In lines 59 → 72 the last item is selected and menu mode is exited.

```
 1 <21JUN01A.CAP>
 2 01:52, 165: 0-06 0-90 0-6A           < 6: Iris Open>
 3 01:52, 166: 1-06                     < 6: ACK>
 4 01:53, 168: 0-06 0-88 0-72           < 6: Focus Far>
 5 01:53, 169: 1-06                     < 6: ACK>
 6 01:54, 171: 0-06 0-8B 0-6F           < 6: Zoom Out>
 7 01:54, 172: 1-06                     < 6: ACK>
 8 01:54, 173: 0-06 0-8C 0-6E           < 6: Zoom Stop>
 9 01:54, 174: 1-06                     < 6: ACK>
10 01:55, 177: 0-06 0-89 0-71           < 6: Focus Stop>
11 01:55, 178: 1-06                     < 6: ACK>
12 01:55, 179: 0-06 0-92 0-68           < 6: Iris Stop>
13 02:04, 189: 0-06 0-C0 0-85 0-05 0-B0 < 6: Var Speed Tilt Down  5/sec>
14 02:04, 190: 1-06                     < 6: ACK>
15 02:04, 191: 0-06 0-86 0-74           < 6: Tilt Stop>
16 02:04, 192: 1-06                     < 6: ACK>
17 02:09, 198: 0-06 0-C0 0-85 0-05 0-B0 < 6: Var Speed Tilt Down  5/sec>
18 02:09, 199: 1-06                     < 6: ACK>
19 02:09, 200: 0-06 0-86 0-74           < 6: Tilt Stop>
20 02:09, 201: 1-06                     < 6: ACK>
21 02:09, 202: 0-06 0-C0 0-85 0-05 0-B0 < 6: Var Speed Tilt Down  5/sec>
22 02:09, 203: 1-06                     < 6: ACK>
23 02:09, 204: 0-06 0-86 0-74           < 6: Tilt Stop>
24 02:09, 205: 1-06                     < 6: ACK>
25 02:10, 207: 0-06 0-C0 0-85 0-05 0-B0 < 6: Var Speed Tilt Down  5/sec>
26 02:10, 208: 1-06                     < 6: ACK>
27 02:10, 209: 0-06 0-86 0-74           < 6: Tilt Stop>
```

```
02:10, 210: 1-06                     < 6: ACK>
02:10, 211: 0-06 0-C0 0-85 0-05 0-B0 < 6: Var Speed Tilt Down  5/sec>
02:10, 212: 1-06                     < 6: ACK>
02:10, 213: 0-06 0-86 0-74           < 6: Tilt Stop>
02:10, 214: 1-06                     < 6: ACK>
02:11, 216: 0-06 0-C0 0-85 0-03 0-B2 < 6: Var Speed Tilt Down  3/sec>
02:11, 217: 1-06                     < 6: ACK>
02:11, 218: 0-06 0-86 0-74           < 6: Tilt Stop>
02:11, 219: 1-06                     < 6: ACK>
02:40, 250: 0-06 0-94 0-66           < 6: Poll, Dome type query>
02:41, 251: 1-06 1-F5 1-05           < 6: Dome Query Response>
02:50, 261: 0-06 0-88 0-72           < 6: Focus Far>
02:50, 262: 1-06                     < 6: ACK>
02:50, 263: 0-06 0-89 0-71           < 6: Focus Stop>
02:50, 264: 1-06                     < 6: ACK>
02:54, 269: 0-06 0-C0 0-85 0-05 0-B0 < 6: Var Speed Tilt Down  5/sec>
02:54, 270: 1-06                     < 6: ACK>
02:54, 271: 0-06 0-86 0-74           < 6: Tilt Stop>
02:54, 272: 1-06                     < 6: ACK>
02:55, 274: 0-06 0-C0 0-84 0-05 0-B1 < 6: Var Speed Tilt Up    5/sec>
02:55, 275: 1-06                     < 6: ACK>
02:55, 276: 0-06 0-C0 0-84 0-2D 0-89 < 6: Var Speed Tilt Up   45/sec>
02:55, 277: 1-06                     < 6: ACK>
02:55, 278: 0-06 0-86 0-74           < 6: Tilt Stop>
02:55, 279: 1-06                     < 6: ACK>
02:56, 281: 0-06 0-C0 0-82 0-06 0-B2 < 6: Var Speed Pan Right  6/sec>
02:56, 282: 1-06                     < 6: ACK>
02:56, 283: 0-06 0-C0 0-82 0-5A 0-5E < 6: Var Speed Pan Right 90/sec>
02:56, 284: 1-06                     < 6: ACK>
02:56, 285: 0-06 0-83 0-77           < 6: Pan Stop>
02:56, 286: 1-06                     < 6: ACK>
02:56, 287: 0-06 0-C0 0-81 0-2D 0-8C < 6: Var Speed Pan Left  45/sec>
02:56, 288: 1-06                     < 6: ACK>
02:56, 289: 0-06 0-C0 0-81 0-5A 0-5F < 6: Var Speed Pan Left  90/sec>
02:56, 290: 1-06                     < 6: ACK>
02:57, 292: 0-06 0-83 0-77           < 6: Pan Stop>
02:57, 293: 1-06                     < 6: ACK>
03:07, 304: 0-06 0-8A 0-70           < 6: Zoom In>
03:07, 305: 1-06                     < 6: ACK>
03:07, 306: 0-06 0-8C 0-6E           < 6: Zoom Stop>
03:07, 307: 1-06                     < 6: ACK>
03:09, 310: 0-06 0-88 0-72           < 6: Focus Far>
03:09, 311: 1-06                     < 6: ACK>
03:09, 312: 0-06 0-89 0-71           < 6: Focus Stop>
03:09, 313: 1-06                     < 6: ACK>
```

1.8.4 Typical dome flip sequence

1.8.4.1 From an AD2083/02 translator

```
            0-06 0-8D 0-6D       <FAST from controller>
```

```
 2            1-06                 <ACK from dome to controller message>
 3            0-06 0-8E 0-6C       <FASTEST from controller>
 4            1-06                 <ACK from dome to controller message>
```

1.8.4.2 From an RC58 controller

This is an example of a flip command sequence which was captured from an RC58 type controller.

Notes

1. In lines 2 → 3 the dome is selected and the controller sends an ON AIR command.
2. In lines 4 → 11 that the RAIL LEFT (FAST) and RAIL RIGHT (FASTEST) buttons were depressed in the correct order and that RAIL LEFT was held down while RAIL RIGHT was depressed. Then note that the controller sends two FAST STOP commands when the buttons are released.
3. In lines 12 → 19 I had it flip back to its original position.
4. In lines 20 → 21 I selected another dome and the controller automatically sent a RESET ON AIR command.

```
 1 <29MAR01A.CAP>
 2 05:46, 956: 0-05 0-9E 0-5D          < 5: On Air>
 3 05:46, 957: 1-05                    < 5: ACK>
 4 05:50, 970: 0-05 0-8D 0-6E          < 5: Fast (Rail left)>
 5 05:50, 971: 1-05                    < 5: ACK>
 6 05:51, 974: 0-05 0-8E 0-6D          < 5: Fastest (Rail right)>
 7 05:51, 975: 1-05                    < 5: ACK>
 8 05:51, 976: 0-05 0-8F 0-6C          < 5: Fast Stop>
 9 05:51, 977: 1-05                    < 5: ACK>
10 05:51, 978: 0-05 0-8F 0-6C          < 5: Fast Stop>
11 05:51, 979: 1-05                    < 5: ACK>
12 05:57, 993: 0-05 0-8D 0-6E          < 5: Fast (Rail left)>
13 05:57, 994: 1-05                    < 5: ACK>
14 05:57, 995: 0-05 0-8E 0-6D          < 5: Fastest (Rail right)>
15 05:57, 996: 1-05                    < 5: ACK>
16 05:57, 997: 0-05 0-8F 0-6C          < 5: Fast Stop>
17 05:57, 998: 1-05                    < 5: ACK>
18 05:58,   1: 0-05 0-8F 0-6C          < 5: Fast Stop>
19 05:58,   2: 1-05                    < 5: ACK>
20 06:16,  50: 0-05 0-9F 0-5C          < 5: Reset On Air>
21 06:16,  51: 1-05                    < 5: ACK>
```

1.8.4.3 From an RC216 controller

This is an example of a flip command sequence which was captured from an RC216 type controller.

```
 1 <14JAN02A.CAP>
```

```
08:26,  2576: 0-06 0-89 0-71           < 6: Focus Stop>
08:26,  2577: 1-06                     < 6: ACK>
08:26,  2578: 0-06 0-8D 0-6D           < 6: Fast (Rail left)>
08:26,  2579: 1-06                     < 6: ACK>
08:27,  2582: 0-06 0-8E 0-6C           < 6: Fastest (Rail right)>
08:27,  2583: 1-06                     < 6: ACK>
```

1.8.5 Typical variable speed dome commands

1.8.5.1 From an AD2083/02 translator

```
<20MAR01C.CAP>
01:51, 287: 0-06 0-C0 0-82 0-06 0-B2 < 6: Var Speed Pan Right  6/sec>
01:51, 288: 1-06                     < 6: ACK>
01:51, 289: 0-06 0-C0 0-85 0-03 0-B2 < 6: Var Speed Tilt Down  3/sec>
01:51, 290: 1-06                     < 6: ACK>
01:52, 292: 0-06 0-C0 0-82 0-5A 0-5E < 6: Var Speed Pan Right 90/sec>
01:52, 293: 1-06                     < 6: ACK>
01:52, 294: 0-06 0-83 0-77           < 6: Pan Stop>
01:52, 295: 1-06                     < 6: ACK>
01:52, 296: 0-06 0-86 0-74           < 6: Tilt Stop>
01:52, 297: 1-06                     < 6: ACK>
01:52, 298: 0-06 0-C0 0-81 0-0F 0-AA < 6: Var Speed Pan Left  15/sec>
01:52, 299: 1-06                     < 6: ACK>
01:52, 300: 0-06 0-C0 0-81 0-5A 0-5F < 6: Var Speed Pan Left  90/sec>
01:52, 301: 1-06                     < 6: ACK>
01:53, 303: 0-06 0-83 0-77           < 6: Pan Stop>
01:53, 304: 1-06                     < 6: ACK>
01:53, 305: 0-06 0-C0 0-85 0-18 0-9D < 6: Var Speed Tilt Down 24/sec>
01:53, 306: 1-06                     < 6: ACK>
01:53, 307: 0-06 0-C0 0-81 0-04 0-B5 < 6: Var Speed Pan Left   4/sec>
01:53, 308: 1-06                     < 6: ACK>
01:53, 309: 0-06 0-C0 0-85 0-2D 0-88 < 6: Var Speed Tilt Down 45/sec>
01:53, 310: 1-06                     < 6: ACK>
01:53, 311: 0-06 0-83 0-77           < 6: Pan Stop>
01:53, 312: 1-06                     < 6: ACK>
01:54, 314: 0-06 0-86 0-74           < 6: Tilt Stop>
01:54, 315: 1-06                     < 6: ACK>
01:54, 316: 0-06 0-C0 0-84 0-03 0-B3 < 6: Var Speed Tilt Up    3/sec>
01:54, 317: 1-06                     < 6: ACK>
01:54, 318: 0-06 0-C0 0-84 0-21 0-95 < 6: Var Speed Tilt Up   33/sec>
01:54, 319: 1-06                     < 6: ACK>
01:54, 320: 0-06 0-C0 0-84 0-2D 0-89 < 6: Var Speed Tilt Up   45/sec>
01:54, 321: 1-06                     < 6: ACK>
01:54, 322: 0-06 0-94 0-66           < 6: Poll, Dome type query>
01:55, 323: 1-06 1-E8 1-12           < 6: Dome Query Response>
```

1.8.5.2 From an RC58 controller

The RC58 does not have a “real” variable speed capability. Instead it has four fixed speeds of operation. These are “normal”, “fast”, “faster” and “fastest”.

```
<22MAR01A.CAP>
<Polls deleted>
10:09, 696: 0-06 0-81 0-79           < 6: Pan Left>
10:09, 697: 1-06                     < 6: ACK>
10:09, 698: 0-06 0-81 0-79           < 6: Pan Left>
10:09, 699: 1-06                     < 6: ACK>
10:09, 700: 0-06 0-81 0-79           < 6: Pan Left>
10:09, 701: 1-06                     < 6: ACK>
<Polls deleted>
10:13, 714: 0-06 0-9A 0-60           < 6: Faster>
10:13, 715: 1-06                     < 6: ACK>
10:13, 716: 0-06 0-9A 0-60           < 6: Faster>
10:13, 717: 1-06                     < 6: ACK>
10:13, 718: 0-06 0-9A 0-60           < 6: Faster>
10:13, 719: 1-06                     < 6: ACK>
<Polls deleted>
10:17, 732: 0-06 0-83 0-77           < 6: Pan Stop>
10:17, 733: 1-06                     < 6: ACK>
10:17, 734: 0-06 0-83 0-77           < 6: Pan Stop>
10:17, 735: 1-06                     < 6: ACK>
10:17, 736: 0-06 0-83 0-77           < 6: Pan Stop>
10:17, 737: 1-06                     < 6: ACK>
10:17, 738: 0-06 0-9B 0-5F           < 6: Faster Stop>
10:17, 739: 1-06                     < 6: ACK>
10:17, 740: 0-06 0-9B 0-5F           < 6: Faster Stop>
10:17, 741: 1-06                     < 6: ACK>
10:17, 742: 0-06 0-9B 0-5F           < 6: Faster Stop>
10:17, 743: 1-06                     < 6: ACK>
<Polls deleted>
```

Sending multiple fast's, faster's and fastest's:

```
<22MAR01A.CAP>
<Polls deleted>
12:43, 305: 0-06 0-81 0-79           < 6: Pan Left>
12:43, 306: 1-06                     < 6: ACK>
12:43, 307: 0-06 0-81 0-79           < 6: Pan Left>
12:43, 308: 1-06                     < 6: ACK>
12:43, 309: 0-06 0-81 0-79           < 6: Pan Left>
12:43, 310: 1-06                     < 6: ACK>
<Polls deleted>
12:44, 314: 0-06 0-8D 0-6D           < 6: Fast (Rail left)>
12:44, 315: 1-06                     < 6: ACK>
12:44, 316: 0-06 0-8D 0-6D           < 6: Fast (Rail left)>
12:44, 317: 1-06                     < 6: ACK>
12:44, 318: 0-06 0-8D 0-6D           < 6: Fast (Rail left)>
12:44, 319: 1-06                     < 6: ACK>
<Polls deleted>
12:46, 326: 0-06 0-8F 0-6B           < 6: Fast Stop>
12:46, 327: 1-06                     < 6: ACK>
12:46, 328: 0-06 0-8F 0-6B           < 6: Fast Stop>
12:46, 329: 1-06                     < 6: ACK>
12:46, 330: 0-06 0-8F 0-6B           < 6: Fast Stop>
```

```
12:46, 331: 1-06                     < 6: ACK>
<Polls deleted>
12:47, 335: 0-06 0-9A 0-60           < 6: Faster>
12:47, 336: 1-06                     < 6: ACK>
12:47, 337: 0-06 0-9A 0-60           < 6: Faster>
12:47, 338: 1-06                     < 6: ACK>
12:47, 339: 0-06 0-9A 0-60           < 6: Faster>
12:47, 340: 1-06                     < 6: ACK>
<Polls deleted>
12:48, 344: 0-06 0-9B 0-5F           < 6: Faster Stop>
12:48, 345: 1-06                     < 6: ACK>
12:48, 346: 0-06 0-9B 0-5F           < 6: Faster Stop>
12:48, 347: 1-06                     < 6: ACK>
12:48, 348: 0-06 0-9B 0-5F           < 6: Faster Stop>
12:48, 349: 1-06                     < 6: ACK>
<Polls deleted>
12:49, 353: 0-06 0-8E 0-6C           < 6: Fastest (Rail right)>
12:49, 354: 1-06                     < 6: ACK>
12:49, 355: 0-06 0-8E 0-6C           < 6: Fastest (Rail right)>
12:49, 356: 1-06                     < 6: ACK>
12:49, 357: 0-06 0-8E 0-6C           < 6: Fastest (Rail right)>
12:49, 358: 1-06                     < 6: ACK>
<Polls deleted>
12:51, 365: 0-06 0-8F 0-6B           < 6: Fast Stop>
12:51, 366: 1-06                     < 6: ACK>
12:51, 367: 0-06 0-8F 0-6B           < 6: Fast Stop>
12:51, 368: 1-06                     < 6: ACK>
12:51, 369: 0-06 0-8F 0-6B           < 6: Fast Stop>
12:51, 370: 1-06                     < 6: ACK>
<Polls deleted>
12:52, 374: 0-06 0-83 0-77           < 6: Pan Stop>
12:52, 375: 1-06                     < 6: ACK>
12:52, 376: 0-06 0-83 0-77           < 6: Pan Stop>
12:52, 377: 1-06                     < 6: ACK>
12:52, 378: 0-06 0-83 0-77           < 6: Pan Stop>
12:52, 379: 1-06                     < 6: ACK>
<Polls deleted>
```

It is also possible to get these “speed steps” in the tilt axis.

```
<22MAR01A.CAP>
<Polls deleted>
14:38, 806: 0-06 0-85 0-75           < 6: Tilt Down>
14:38, 807: 1-06                     < 6: ACK>
14:38, 808: 0-06 0-85 0-75           < 6: Tilt Down>
14:38, 809: 1-06                     < 6: ACK>
14:38, 810: 0-06 0-85 0-75           < 6: Tilt Down>
14:38, 811: 1-06                     < 6: ACK>
<Polls deleted>
14:40, 818: 0-06 0-9A 0-60           < 6: Faster>
14:40, 819: 1-06                     < 6: ACK>
14:40, 820: 0-06 0-9A 0-60           < 6: Faster>
14:40, 821: 1-06                     < 6: ACK>
```

```
14:40, 822: 0-06 0-9A 0-60           < 6: Faster>
14:40, 823: 1-06                     < 6: ACK>
<Polls deleted>
14:42, 830: 0-06 0-9B 0-5F           < 6: Faster Stop>
14:42, 831: 1-06                     < 6: ACK>
14:42, 832: 0-06 0-9B 0-5F           < 6: Faster Stop>
14:42, 833: 1-06                     < 6: ACK>
14:42, 834: 0-06 0-9B 0-5F           < 6: Faster Stop>
14:42, 835: 1-06                     < 6: ACK>
<Polls deleted>
14:43, 839: 0-06 0-86 0-74           < 6: Tilt Stop>
14:43, 840: 1-06                     < 6: ACK>
14:43, 841: 0-06 0-86 0-74           < 6: Tilt Stop>
14:43, 842: 1-06                     < 6: ACK>
14:43, 843: 0-06 0-86 0-74           < 6: Tilt Stop>
14:43, 844: 1-06                     < 6: ACK>
<Polls deleted>
```

1.8.5.3 From an RC216 controller

Note that the variable speeds are closer together and that boundary crossing responses come from the dome.

```
<30MAY01A.CAP>
00:08, 473: 0-08 0-C0 0-81 0-01 0-B6 < 8: Var Speed Pan Left   1/sec>
00:08, 474: 1-08                     < 8: ACK>
00:08, 475: 1-08                     < 8: ACK>
00:08, 476: 0-08 0-C0 0-84 0-02 0-B2 < 8: Var Speed Tilt Up    2/sec>
00:08, 477: 1-08                     < 8: ACK>
00:08, 478: 0-08 0-C0 0-81 0-07 0-B0 < 8: Var Speed Pan Left   7/sec>
00:08, 479: 1-08                     < 8: ACK>
00:08, 480: 0-08 0-C0 0-84 0-06 0-AE < 8: Var Speed Tilt Up    6/sec>
00:08, 481: 1-08                     < 8: ACK>
00:08, 482: 1-08                     < 8: ACK>
00:08, 483: 0-08 0-C0 0-81 0-09 0-AE < 8: Var Speed Pan Left   9/sec>
00:08, 484: 1-08                     < 8: ACK>
00:08, 485: 0-08 0-C0 0-84 0-08 0-AC < 8: Var Speed Tilt Up    8/sec>
00:08, 486: 1-08                     < 8: ACK>
00:08, 487: 0-08 0-C0 0-81 0-0B 0-AC < 8: Var Speed Pan Left  11/sec>
00:08, 488: 1-08                     < 8: ACK>
00:08, 489: 0-08 0-C0 0-84 0-07 0-AD < 8: Var Speed Tilt Up    7/sec>
00:08, 490: 1-08                     < 8: ACK>
00:08, 491: 0-08 0-C0 0-81 0-0E 0-A9 < 8: Var Speed Pan Left  14/sec>
00:08, 492: 0-08 0-C0 0-84 0-08 0-AC < 8: Var Speed Tilt Up    8/sec>
00:09, 494: 0-08 0-C0 0-81 0-11 0-A6 < 8: Var Speed Pan Left  17/sec>
00:09, 495: 1-08                     < 8: ACK>
00:09, 496: 1-08                     < 8: ACK>
00:09, 497: 0-08 0-C0 0-84 0-09 0-AB < 8: Var Speed Tilt Up    9/sec>
00:09, 498: 0-08 0-C0 0-84 0-0A 0-AA < 8: Var Speed Tilt Up   10/sec>
00:09, 499: 1-08                     < 8: ACK>
```

```
00:09, 500: 0-08 0-C0 0-81 0-14 0-A3 < 8: Var Speed Pan Left  20/sec>
00:09, 501: 1-08                     < 8: ACK>
00:09, 502: 1-08                     < 8: ACK>
00:09, 503: 0-08 0-C0 0-84 0-0B 0-A9 < 8: Var Speed Tilt Up   11/sec>
00:09, 504: 1-08                     < 8: ACK>
00:09, 505: 1-08                     < 8: ACK>
00:09, 506: 0-08 0-C0 0-84 0-0D 0-A7 < 8: Var Speed Tilt Up   13/sec>
00:09, 507: 0-08 0-C0 0-84 0-0F 0-A5 < 8: Var Speed Tilt Up   15/sec>
00:09, 508: 1-08                     < 8: ACK>
00:09, 510: 1-08                     < 8: ACK>
00:09, 511: 0-08 0-C0 0-84 0-10 0-A4 < 8: Var Speed Tilt Up   16/sec>
00:09, 512: 1-08                     < 8: ACK>
00:09, 513: 0-08 0-C0 0-81 0-20 0-97 < 8: Var Speed Pan Left  32/sec>
00:09, 514: 0-08 0-C0 0-84 0-11 0-A3 < 8: Var Speed Tilt Up   17/sec>
00:09, 515: 1-08                     < 8: ACK>
00:09, 516: 0-08 0-C0 0-84 0-12 0-A2 < 8: Var Speed Tilt Up   18/sec>
00:09, 518: 1-08                     < 8: ACK>
00:09, 519: 0-08 0-C0 0-84 0-14 0-A0 < 8: Var Speed Tilt Up   20/sec>
00:09, 520: 1-08                     < 8: ACK>
00:09, 521: 0-08 0-C0 0-81 0-40 0-77 < 8: Var Speed Pan Left  64/sec>
00:10, 523: 1-08                     < 8: ACK>
00:10, 524: 0-08 0-C0 0-84 0-15 0-9F < 8: Var Speed Tilt Up   21/sec>
00:10, 525: 1-08 1-B0 1-48           < 8: Boundry Crossing 1>
00:10, 526: 0-08 0-97 0-61           < 8: ACK to dome>
00:10, 527: 0-08 0-C0 0-84 0-16 0-9E < 8: Var Speed Tilt Up   22/sec>
00:10, 528: 1-08                     < 8: ACK>
00:10, 529: 0-08 0-C0 0-81 0-50 0-67 < 8: Var Speed Pan Left  80/sec>
00:10, 530: 1-08                     < 8: ACK>
00:10, 531: 1-08                     < 8: ACK>
```

1.8.6 Typical polling sequences

1.8.6.1 Typical polling sequence at wrap around with an AD2083/02

This example was taken from an AD2083/02's data capture record. Other controllers have a similar pattern with only the point where wrap around occurs changing. For an RC58 it's 58, for an AD2083/02 it's 99 (63_{16}), etc.

```
<20MAR01C.CAP>
00:06,  42: 0-60 0-94 0-0C           <96: Poll, Dome type query>
00:07,  43: 0-61 0-94 0-0B           <97: Poll, Dome type query>
00:08,  44: 0-62 0-94 0-0A           <98: Poll, Dome type query>
00:09,  45: 0-63 0-94 0-09           <99: Poll, Dome type query>

00:10,  47: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
00:11,  48: 0-02 0-94 0-6A           < 2: Poll, Dome type query>
00:12,  49: 0-03 0-94 0-69           < 3: Poll, Dome type query>
00:13,  50: 0-04 0-94 0-68           < 4: Poll, Dome type query>
```

1.8.6.2 Typical RC58 polling sequence

This data was collected at a customer site. The customer had several cameras active and this example has been edited to remove extraneous information. Most of the extraneous information was the constant presence of "POLL"ing messages.

In this sequence it should be noted that camera/dome `0x37` is not installed. However the RC58 still attempts to detect if a camera is there by sending "queries" (or DOME TYPE requests). (A difference between the RC58 and AD2083/02 output signals is also evident here, is that the command to camera `0x37` is sent three times by an RC58 and only gets sent once by an AD2083/02.)

Now note that when the RC58 gets a camera response (from camera 0x34). That response indicates which dome/camera type is active. This is followed by a `0x97` (SEND ACK) message which is used to acknowledge the reply that came from the dome.

```
<29MAR01B.CAP>
00:39, 133: 0-37 0-94 0-35          <55: Poll, Dome type query>
00:40, 134: 0-37 0-94 0-35          <55: Poll, Dome type query>
00:40, 135: 0-37 0-94 0-35          <55: Poll, Dome type query>
00:40, 136: 0-38 0-94 0-34          <56: Poll, Dome type query>
00:41, 137: 1-38 1-F5 1-D3          <56: Dome Query Response>
00:41, 138: 0-38 0-97 0-31          <56: ACK to dome>
```

1.8.6.3 Polling domes with an RC58

Note that the POLL command is repeated three times as is its response from the dome. Also note that the POLLing sequence wraps from `0x3A` (58_{10}) to `0x01` (1_{10}).

```
<29MAR01B.CAP>
00:41, 141: 0-39 0-94 0-33          <57: Poll, Dome type query>
00:42, 142: 0-39 0-94 0-33          <57: Poll, Dome type query>
00:42, 143: 0-39 0-94 0-33          <57: Poll, Dome type query>
00:42, 144: 0-3A 0-94 0-32          <58: Poll, Dome type query>
00:43, 145: 0-3A 0-94 0-32          <58: Poll, Dome type query>
00:43, 146: 0-3A 0-94 0-32          <58: Poll, Dome type query>

00:43, 148: 0-01 0-94 0-6B          < 1: Poll, Dome type query>
00:44, 149: 1-01 1-F5 1-0A          < 1: Dome Query Response>
00:44, 150: 0-01 0-97 0-68          < 1: ACK to dome>
00:44, 151: 0-02 0-94 0-6A          < 2: Poll, Dome type query>
00:45, 152: 0-02 0-97 0-67          < 2: ACK to dome>
```

1.8.6.4 Typical RC216 polling sequence at wrap around

```
<30MAY01A.CAP>
01:25, 116: 0-5F 0-94 0-0D          <95: Poll, Dome type query>
01:26, 117: 0-60 0-94 0-0C          <96: Poll, Dome type query>

```

```
5 01:26, 119: 0-01 0-94 0-6B          < 1: Poll, Dome type query>
6 01:27, 120: 0-02 0-94 0-6A          < 2: Poll, Dome type query>
7 01:27, 121: 1-02 1-F5 1-09          < 2: Dome Query Response>
```

1.8.7 Typical "software version" responses

These responses were from domes #4 and then #5. They were obtained by sending a request software version (S_SOFTWARE_VERSION, 0xC9) command with a specially written GWBASIC program. (I couldn't figure out how to get the American Dynamics equipment to do it.)

1. Dome #4 EPROM is marked "0701-2507-0303".

```
1       0-06 0-C9 0-31
2       1-06 1-C9 1-06 1-07 1-01 1-25 1-07 1-03 1-03 1-F1
```

2. Dome #5 EPROM is marked "0701-0081-0316".

```
1       0-06 0-C9 0-31
2       1-06 1-C9 1-06 1-07 1-01 1-00 1-81 1-03 1-16 1-89
```

1.8.8 Typical "button" commands

This is for IRIS OPEN, all the other single button commands generate a similar command sequence. These include: ZOOM IN, ZOOM OUT, FOCUS NEAR, FOCUS FAR, IRIS OPEN, IRIS CLOSE. By generating a similar sequence it is meant that on button down one command is generated and on button up a STOP command is generated.

```
1 <20MAR01C.CAP>
2 01:11, 162: 0-06 0-90 0-6A          < 6: Iris Open>
3 01:11, 163: 1-06                    < 6: ACK>
4 01:11, 164: 0-06 0-92 0-68          < 6: Iris Stop>
5 01:11, 165: 1-06                    < 6: ACK>
```

1.8.9 Typical dome set preset sequence

1.8.9.1 With an AD2083/02 translator

Notes

1. In lines 2 → 4 the controller sends three 0x98 commands to address 0x40 (64_{16}). This may be to stop all the domes from sending the controller any messages.
2. In line 5 the controller requests the dome's position.
3. In lines 6 → 13 the dome acknowledges the position request and sends its current position. Then the controller acknowledges receipt of the current position message.
4. In lines 14 → 37 the above procedure is repeated twice more.
5. In lines 38 → 40 the controller sends three 0x99 commands to address 0x40 (64_{16}). This may be to tell all domes to restart sending any of their normal messages to the controller.

6. From this it is possible that address `0x40` (64_{16}) is used as a “broadcast” address that all devices are expected to “listen” to.

```
<20MAR01C.CAP>
01:15, 172: 0-40 0-98 0-28           <64: Start Temp No Xmit>
01:15, 173: 0-40 0-98 0-28           <64: Start Temp No Xmit>
01:15, 174: 0-40 0-98 0-28           <64: Start Temp No Xmit>
01:15, 175: 0-06 0-A5 0-55           < 6: Request Dome Position>
01:15, 176: 1-06                     < 6: ACK>
01:15, 177: 1-06 1-00 1-00 1-01 1-DA 1-0B 1-5E 1-15 1-01 1-34 1-43 1-29 < 6: Current Dome
Position>
                                Iris xunning    = 0x00
                                Zoom limit      = 0x00
                                Tilt Position   = 0x01 0xDA
                                Zoom Position   = 0x0B 0x5E
                                Electronic Zoom = 0x15 0x01
                                Pan Position    = 0x34 0x43
01:15, 178: 0-40 0-98 0-28           <64: Start Temp No Xmit>
01:15, 179: 0-40 0-98 0-28           <64: Start Temp No Xmit>
01:15, 180: 0-40 0-98 0-28           <64: Start Temp No Xmit>
01:15, 181: 0-06 0-A5 0-55           < 6: Request Dome Position>
01:15, 182: 1-06                     < 6: ACK>
01:15, 183: 1-06 1-00 1-00 1-01 1-DA 1-0B 1-5E 1-15 1-01 1-34 1-43 1-29 < 6: Current Dome
Position>
                                Iris xunning    = 0x00
                                Zoom limit      = 0x00
                                Tilt Position   = 0x01 0xDA
                                Zoom Position   = 0x0B 0x5E
                                Electronic Zoom = 0x15 0x01
                                Pan Position    = 0x34 0x43
01:15, 184: 0-40 0-98 0-28           <64: Start Temp No Xmit>
01:15, 185: 0-40 0-98 0-28           <64: Start Temp No Xmit>
01:15, 186: 0-40 0-98 0-28           <64: Start Temp No Xmit>
01:15, 187: 0-06 0-A5 0-55           < 6: Request Dome Position>
01:15, 188: 1-06                     < 6: ACK>
01:15, 189: 1-06 1-00 1-00 1-01 1-DA 1-0B 1-5E 1-15 1-01 1-34 1-43 1-29 < 6: Current Dome
Position>
                                Iris xunning    = 0x00
                                Zoom limit      = 0x00
                                Tilt Position   = 0x01 0xDA
                                Zoom Position   = 0x0B 0x5E
                                Electronic Zoom = 0x15 0x01
                                Pan Position    = 0x34 0x43
01:15, 190: 0-40 0-99 0-27           <64: End Temp No Xmit>
01:15, 191: 0-40 0-99 0-27           <64: End Temp No Xmit>
01:15, 192: 0-40 0-99 0-27           <64: End Temp No Xmit>
```

The position information is sent three times. Its format is as follows:

Format of a POSITION response		
#	Byte	Use
1	0x06	Address
2	0x00	Iris Information
3	0x00	Zoom Limit
4, 5	0x06, 0x25	Tilt Position
6, 7	0x26, 0x6B	Zoom Position
8, 9	0x28, 0x3A	Electronic Zoom
10, 11	0x33, 0x0E	Pan Position
12	0x9B	Checksum

1.8.9.2 With an RC58 controller

This data was collected at a customer site. The customer had several cameras active and this example has been edited to remove extraneous information. Most of the extraneous information was the constant presence of "POLL"ing messages. The location of the deleted "POLL"s may be deduced by "skips" in the time or/and message count columns.

Notes

1. In lines 2 → 7 the selected dome is placed ON AIR and has an undefined command of 0x80 sent to it. The 0x80 command is given a conventional ACK (0x01, being dome number 1) and then responds with an unknown response of 0xC4 and a BOUNDARY CROSSING 3 message of 0xB2. At this point it is unlikely that the dome is actually moving, so the generation of the BOUNDARY CROSSING command probably is used to indicate which area "boundary" ("zone" for Pelco) the dome is pointing in.
2. In lines 8 → 29 the dome is moved around some.
3. In lines 30 → 31 the dome is told to MARK this target.
4. In lines 32 → 34 the controller might be telling all domes to operate in "silent" mode. (The use of command 0x98 sent to address 0x40 seams to be a command sequence to cause all domes to temporarily halt sending messages to the controller.)
5. In line 35 an unknown command of 0xA4 is sent to the dome. This appears to cause the dome to do a memory dump of some type which shown in lines 36 → 46. (More dump details are in Section 1.11, page 68.) The 104 bytes of response that are generated by sending an 0xA4 have an unknown format and use.
6. In lines 47 → 49 a 0x99 command is sent three times to address 0x40 which might release silent mode for all domes.

```
1 <29MAR01A.CAP>
2 52:15,   8: 0-01 0-9E 0-61           < 1: On Air>
3 52:15,   9: 1-01                     < 1: ACK>
4 52:15,  10: 0-01 0-80 0-7F           < 1: Unknown 80>
```

```
52:15,  11: 1-01                     < 1: ACK>
52:16,  14: 1-01 1-C4 1-3B           < 1: Unknown Response C4>
52:16,  15: 0-01 0-97 0-68           < 1: ACK to dome>
52:16,  16: 1-01 1-B2 1-4D           < 1: Boundry Crossing 3>
52:16,  17: 0-01 0-97 0-68           < 1: ACK to dome>
52:47,  99: 0-01 0-82 0-7D           < 1: Pan Right>
52:47, 100: 1-01                     < 1: ACK>
52:47, 101: 0-01 0-83 0-7C           < 1: Pan Stop>
52:47, 102: 1-01                     < 1: ACK>

52:48, 107: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
52:49, 108: 1-01 1-F5 1-0A           < 1: Dome Query Response>
52:49, 109: 0-01 0-97 0-68           < 1: ACK to dome>
52:52, 116: 0-01 0-82 0-7D           < 1: Pan Right>
52:52, 117: 1-01                     < 1: ACK>
52:54, 124: 0-01 0-83 0-7C           < 1: Pan Stop>
52:54, 125: 1-01                     < 1: ACK>
52:55, 128: 0-01 0-82 0-7D           < 1: Pan Right>
52:55, 129: 1-01                     < 1: ACK>
52:56, 132: 0-01 0-83 0-7C           < 1: Pan Stop>
52:56, 133: 1-01                     < 1: ACK>
53:01, 146: 0-01 0-85 0-7A           < 1: Tilt Down>
53:01, 147: 1-01                     < 1: ACK>
53:02, 150: 0-01 0-86 0-79           < 1: Tilt Stop>
53:02, 151: 1-01                     < 1: ACK>
53:21, 198: 0-01 0-A8 0-57           < 1: Mark Target 1>
53:21, 199: 1-01                     < 1: ACK>
53:21, 200: 0-40 0-98 0-28           <64: Start Temp No Xmit>
53:21, 201: 0-40 0-98 0-28           <64: Start Temp No Xmit>
53:21, 202: 0-40 0-98 0-28           <64: Start Temp No Xmit>
53:21, 203: 0-01 0-A4 0-5B           < 1: Dump Dome Memory>
53:21, 204: 1-01 1-66 1-0E 1-80 1-0D 1-56 1-1F 1-50 1-1A 1-5F <Bytes  1-10>
            1-08 1-5F 1-C0 1-00 1-00 1-00 1-00 1-00 1-00 1-00 <Bytes 11-20>
            1-00 1-00 1-00 1-00 1-00 1-00 1-0A 1-A7 1-21 1-7B <Bytes 21-30>
            1-1C 1-25 1-36 1-1D 1-1F 1-00 1-00 1-0A 1-A7 1-21 <Bytes 31-40>
            1-7B 1-1C 1-25 1-34 1-68 1-D6 1-07 1-86 1-73 1-05 <Bytes 41-50>
            1-E2 1-19 1-03 1-6D 1-90 1-00 1-94 1-6C 1-0E 1-00 <Bytes 51-60>
            1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 <Bytes 61-70>
            1-DF 1-FF 1-22 1-0E 1-01 1-00 1-00 1-00 1-00 1-00 <Bytes 71-80>
            1-00 1-00 1-00 1-00 1-00 1-DF 1-FF 1-21 1-0E 1-02 <Bytes 81-90>
            1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 <Bytes 91-100>
            1-DF 1-FF 1-20 1-6F < 1: Dome memory dump msg length = 104>
53:21, 205: 0-40 0-99 0-27           <64: End Temp No Xmit>
53:21, 206: 0-40 0-99 0-27           <64: End Temp No Xmit>
53:21, 207: 0-40 0-99 0-27           <64: End Temp No Xmit>
```

1.8.10 Goto start of a zone

When using an RC58 type matrix there is an unexpected use for commands with an opcode of `0xA6`. This is used to send the dome to the start of a “zone”. Very little is known about this command and it has only been captured at Sears Fresno. It is generated by hitting the ZONE button on the console followed by a number of 1, 2, 3 or 4.

```
<15JAN02A.CAP>
00:00,  2204: 0-37 0-A6 0-23           <55: Go To Absolute Position>
```

1.8.11 Typical dome call preset sequence

1.8.11.1 With an AD2083/02 translator

```
<20MAR01C.CAP>
01:22, 215: 0-06 0-A6 0-3D 0-6E 0-43 0-F6 0-0B 0-EC 0-00 0-00 0-3E 0-55 0-E6 < 6: Goto Position>
                                Pan Position          = 0x3D 0x6E
                                Tilt Position         = 0x43 0xF6
                                Zoom Position         = 0x0B 0xEC
                                Digital Zoom          = 0x00 0x00
                                Iris Offset           = 0x3E
                                Electronic Zoom Limit = 0x55
01:22, 216: 1-06                     < 6: ACK>
01:22, 217: 0-06 0-A6 0-3D 0-6E 0-43 0-F6 0-0B 0-EC 0-00 0-00 0-3E 0-55 0-E6 < 6: Goto Position>
                                Pan Position          = 0x3D 0x6E
                                Tilt Position         = 0x43 0xF6
                                Zoom Position         = 0x0B 0xEC
                                Digital Zoom          = 0x00 0x00
                                Iris Offset           = 0x3E
                                Electronic Zoom Limit = 0x55
01:22, 218: 1-06                     < 6: ACK>
01:22, 219: 0-06 0-A6 0-3D 0-6E 0-43 0-F6 0-0B 0-EC 0-00 0-00 0-3E 0-55 0-E6 < 6: Goto Position>
                                Pan Position          = 0x3D 0x6E
                                Tilt Position         = 0x43 0xF6
                                Zoom Position         = 0x0B 0xEC
                                Digital Zoom          = 0x00 0x00
                                Iris Offset           = 0x3E
                                Electronic Zoom Limit = 0x55
01:22, 220: 1-06                     < 6: ACK>
```

Format of a GOTO command		
#	Byte	Use
1	0x06	Address
2	0xA6	Command
3, 4	0x33, 0x0E	Pan Position
5, 6	0x06, 0x25	Tilt Position
7, 8	0x26, 0x6B	Zoom Position
9, 10	0x28, 0x3A	Digital Zoom
11	0x00	Iris Offset
12	0x00	Electronic Zoom Limit
13	0xF5	Checksum

1.8.11.2 With an RC58 controller

This data was collected at a customer site.

```
<29MAR01A.CAP>
15:58, 120: 0-01 0-B7 0-48           < 1: Go To Target 4>
15:58, 121: 1-01                     < 1: ACK>
```

1.8.11.3 With an RC216 controller

```
<30MAY01A.CAP>
10:10, 222: 0-10 0-9E 0-52           <16: On Air>
10:10, 223: 1-10                     <16: ACK>
10:10, 224: 0-10 0-A6 0-36 0-C1 0-4B 0-16 0-10 0-E4 0-00 0-00 0-80 0-15 0-69 <16: Goto
Position>
                                Pan Position          = 0x36 0xC1
                                Tilt Position         = 0x4B 0x16
                                Zoom Position         = 0x10 0xE4
                                Digital Zoom          = 0x00 0x00
                                Iris Offset           = 0x80
                                Electronic Zoom Limit = 0x15
10:10, 225: 0-59 0-94 0-13           <89: Poll, Dome type query>
10:11, 226: 1-10                     <16: ACK>
10:11, 227: 1-10 1-B0 1-40           <16: Boundry Crossing 1>
10:11, 228: 0-10 0-97 0-59           <16: ACK to dome>
10:11, 229: 0-5A 0-94 0-12           <90: Poll, Dome type query>
10:11, 230: 0-5B 0-94 0-11           <91: Poll, Dome type query>
10:12, 231: 1-10 1-B3 1-3D           <16: Boundry Crossing 4>
10:12, 232: 0-10 0-97 0-59           <16: ACK to dome>
```

1.8.12 Typical dome pattern preset sequence

1.8.12.1 With an RC58 controller

This data was collected at a customer site. The customer had several cameras active and this example has been edited to remove extraneous information. Most of the extraneous information was the constant presence of “POLL”ing messages.

Notes

1. In lines 2 → 9 the controller initiates the standard startup sequence for beginning to work with a dome. I.e. puts it ON AIR and determines what BOUNDARY it is inside of.
2. In lines 10 → 149 I moved the dome around a bit.
3. In lines 150 → 151 I sent a DEFINE PATTERN 1 command.
4. In lines 152 → 284 I moved the dome around some more.
5. In lines 285 → 286 I send a PATTERN END command.
6. In lines 287 → 334 I had the dome run the pattern. Note the various BOUNDARY CROSSING replies from the dome while running the test pattern.
7. In lines 335 → 336 I told the dome to accept the pattern.
8. In lines 337 → 364 the controller might “silence” all other domes and appears to request a data dump from the dome. (More dump details are in Section 1.11, page 68.) It then appears to release all other domes.

```
 1 <29MAR01A.CAP>
 2 00:44, 458: 0-01 0-9E 0-61          < 1: On Air>
 3 00:44, 459: 1-01                    < 1: ACK>
 4 00:44, 460: 0-01 0-80 0-7F          < 1: Unknown 80>
 5 00:44, 461: 1-01                    < 1: ACK>
 6 00:45, 464: 1-01 1-C4 1-3B          < 1: Unknown Response C4>
 7 00:45, 465: 0-01 0-97 0-68          < 1: ACK to dome>
 8 00:45, 466: 1-01 1-B2 1-4D          < 1: Boundry Crossing 3>
 9 00:45, 467: 0-01 0-97 0-68          < 1: ACK to dome>
10 01:17, 545: 0-01 0-82 0-7D          < 1: Pan Right>
11 01:17, 546: 1-01                    < 1: ACK>
12 01:21, 558: 0-01 0-83 0-7C          < 1: Pan Stop>
13 01:21, 559: 1-01                    < 1: ACK>
14 01:24, 569: 0-01 0-82 0-7D          < 1: Pan Right>
15 01:24, 570: 1-01                    < 1: ACK>
16 01:26, 577: 1-01 1-B3 1-4C          < 1: Boundry Crossing 4>
17 01:26, 578: 0-01 0-97 0-68          < 1: ACK to dome>
18 01:27, 582: 0-01 0-83 0-7C          < 1: Pan Stop>
19 01:27, 583: 1-01                    < 1: ACK>
20 01:29, 590: 0-01 0-84 0-7B          < 1: Tilt Up>
21 01:29, 591: 1-01                    < 1: ACK>
22
23 01:30, 596: 0-01 0-94 0-6B          < 1: Poll, Dome type query>
24 01:31, 597: 1-01 1-F5 1-0A          < 1: Dome Query Response>
25 01:31, 598: 0-01 0-97 0-68          < 1: ACK to dome>
26 01:31, 599: 0-01 0-86 0-79          < 1: Tilt Stop>
27 01:31, 600: 1-01                    < 1: ACK>
28 01:34, 607: 0-01 0-82 0-7D          < 1: Pan Right>
```

```
29 01:34, 608: 1-01                     < 1: ACK>
30 01:37, 617: 0-01 0-83 0-7C           < 1: Pan Stop>
31 01:37, 618: 1-01                     < 1: ACK>
32 01:38, 620: 1-01 1-B0 1-4F           < 1: Boundry Crossing 1>
33 01:38, 621: 0-01 0-97 0-68           < 1: ACK to dome>
34 01:38, 622: 0-01 0-82 0-7D           < 1: Pan Right>
35 01:38, 623: 1-01                     < 1: ACK>
36 01:39, 627: 0-01 0-83 0-7C           < 1: Pan Stop>
37 01:39, 628: 1-01                     < 1: ACK>
38 01:42, 636: 0-01 0-85 0-7A           < 1: Tilt Down>
39 01:42, 637: 1-01                     < 1: ACK>
40 01:43, 640: 0-01 0-86 0-79           < 1: Tilt Stop>
41 01:43, 641: 1-01                     < 1: ACK>
42 01:44, 644: 0-01 0-85 0-7A           < 1: Tilt Down>
43 01:44, 645: 1-01                     < 1: ACK>
44 01:44, 646: 0-01 0-86 0-79           < 1: Tilt Stop>
45 01:44, 647: 1-01                     < 1: ACK>
46 01:46, 652: 0-01 0-85 0-7A           < 1: Tilt Down>
47 01:46, 653: 1-01                     < 1: ACK>
48 01:46, 654: 0-01 0-86 0-79           < 1: Tilt Stop>
49 01:46, 655: 1-01                     < 1: ACK>
50 01:49, 662: 0-01 0-84 0-7B           < 1: Tilt Up>
51 01:49, 663: 1-01                     < 1: ACK>
52 01:49, 664: 0-01 0-86 0-79           < 1: Tilt Stop>
53 01:49, 665: 1-01                     < 1: ACK>
54 01:50, 669: 0-01 0-84 0-7B           < 1: Tilt Up>
55 01:50, 670: 1-01                     < 1: ACK>
56 01:50, 671: 0-01 0-86 0-79           < 1: Tilt Stop>
57 01:50, 672: 1-01                     < 1: ACK>
58 01:51, 676: 0-01 0-84 0-7B           < 1: Tilt Up>
59 01:51, 677: 1-01                     < 1: ACK>
60 01:51, 678: 0-01 0-86 0-79           < 1: Tilt Stop>
61 01:51, 679: 1-01                     < 1: ACK>
62 01:53, 686: 0-01 0-85 0-7A           < 1: Tilt Down>
63 01:53, 687: 1-01                     < 1: ACK>
64 01:54, 691: 0-01 0-86 0-79           < 1: Tilt Stop>
65 01:54, 692: 1-01                     < 1: ACK>
66 01:56, 697: 0-01 0-82 0-7D           < 1: Pan Right>
67 01:56, 698: 1-01                     < 1: ACK>
68 02:02, 712: 1-01 1-B1 1-4E           < 1: Boundry Crossing 2>
69 02:02, 713: 0-01 0-97 0-68           < 1: ACK to dome>
70 02:14, 744: 1-01 1-B2 1-4D           < 1: Boundry Crossing 3>
71 02:14, 745: 0-01 0-97 0-68           < 1: ACK to dome>
72 02:20, 763: 0-01 0-83 0-7C           < 1: Pan Stop>
73 02:20, 764: 1-01                     < 1: ACK>
74 02:22, 771: 0-01 0-82 0-7D           < 1: Pan Right>
75 02:22, 772: 1-01                     < 1: ACK>
76 02:23, 776: 0-01 0-83 0-7C           < 1: Pan Stop>
77 02:23, 777: 1-01                     < 1: ACK>
78 02:27, 790: 0-01 0-85 0-7A           < 1: Tilt Down>
79 02:27, 791: 1-01                     < 1: ACK>
80 02:27, 792: 0-01 0-86 0-79           < 1: Tilt Stop>
```

```
02:27, 793: 1-01                     < 1: ACK>
02:28, 797: 0-01 0-85 0-7A           < 1: Tilt Down>
02:28, 798: 1-01                     < 1: ACK>
02:28, 799: 0-01 0-86 0-79           < 1: Tilt Stop>
02:28, 800: 1-01                     < 1: ACK>

02:28, 802: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
02:29, 803: 1-01 1-F5 1-0A           < 1: Dome Query Response>
02:29, 804: 0-01 0-97 0-68           < 1: ACK to dome>
02:31, 809: 0-01 0-81 0-7E           < 1: Pan Left>
02:31, 810: 1-01                     < 1: ACK>
02:31, 811: 0-01 0-83 0-7C           < 1: Pan Stop>
02:31, 812: 1-01                     < 1: ACK>
02:56, 873: 0-01 0-82 0-7D           < 1: Pan Right>
02:56, 874: 1-01                     < 1: ACK>
03:03, 889: 1-01 1-B3 1-4C           < 1: Boundry Crossing 4>
03:03, 890: 0-01 0-97 0-68           < 1: ACK to dome>
03:03, 891: 0-01 0-83 0-7C           < 1: Pan Stop>
03:03, 892: 1-01                     < 1: ACK>
03:03, 893: 0-01 0-81 0-7E           < 1: Pan Left>
03:03, 894: 1-01                     < 1: ACK>
03:05, 900: 1-01 1-B2 1-4D           < 1: Boundry Crossing 3>
03:05, 901: 0-01 0-97 0-68           < 1: ACK to dome>
03:06, 905: 0-01 0-83 0-7C           < 1: Pan Stop>
03:06, 906: 1-01                     < 1: ACK>
03:10, 918: 0-01 0-81 0-7E           < 1: Pan Left>
03:10, 919: 1-01                     < 1: ACK>
03:10, 920: 0-01 0-83 0-7C           < 1: Pan Stop>
03:10, 921: 1-01                     < 1: ACK>
03:14, 933: 0-01 0-82 0-7D           < 1: Pan Right>
03:14, 934: 1-01                     < 1: ACK>
03:14, 935: 0-01 0-83 0-7C           < 1: Pan Stop>
03:14, 936: 1-01                     < 1: ACK>
03:17, 945: 0-01 0-85 0-7A           < 1: Tilt Down>
03:17, 946: 1-01                     < 1: ACK>
03:17, 947: 0-01 0-86 0-79           < 1: Tilt Stop>
03:17, 948: 1-01                     < 1: ACK>
03:18, 952: 0-01 0-85 0-7A           < 1: Tilt Down>
03:18, 953: 1-01                     < 1: ACK>
03:18, 954: 0-01 0-86 0-79           < 1: Tilt Stop>
03:18, 955: 1-01                     < 1: ACK>
03:20, 962: 0-01 0-84 0-7B           < 1: Tilt Up>
03:20, 963: 1-01                     < 1: ACK>
03:20, 964: 0-01 0-86 0-79           < 1: Tilt Stop>
03:20, 965: 1-01                     < 1: ACK>
03:21, 969: 0-01 0-84 0-7B           < 1: Tilt Up>
03:21, 970: 1-01                     < 1: ACK>
03:21, 971: 0-01 0-86 0-79           < 1: Tilt Stop>
03:21, 972: 1-01                     < 1: ACK>
03:21, 973: 0-01 0-84 0-7B           < 1: Tilt Up>
03:21, 974: 1-01                     < 1: ACK>
03:22, 978: 0-01 0-86 0-79           < 1: Tilt Stop>
```

```
03:22, 979: 1-01                     < 1: ACK>
03:22, 980: 0-01 0-84 0-7B           < 1: Tilt Up>
03:22, 981: 1-01                     < 1: ACK>
03:22, 982: 0-01 0-86 0-79           < 1: Tilt Stop>
03:22, 983: 1-01                     < 1: ACK>
03:23, 987: 0-01 0-84 0-7B           < 1: Tilt Up>
03:23, 988: 1-01                     < 1: ACK>
03:23, 989: 0-01 0-86 0-79           < 1: Tilt Stop>
03:23, 990: 1-01                     < 1: ACK>

03:26,   1: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
03:27,   2: 1-01 1-F5 1-0A           < 1: Dome Query Response>
03:27,   3: 0-01 0-97 0-68           < 1: ACK to dome>
03:29,   8: 0-01 0-85 0-7A           < 1: Tilt Down>
03:29,   9: 1-01                     < 1: ACK>
03:29,  10: 0-01 0-86 0-79           < 1: Tilt Stop>
03:29,  11: 1-01                     < 1: ACK>
03:41,  40: 0-01 0-A0 0-5F           < 1: Define Pattern 1>
03:41,  41: 1-01                     < 1: ACK>
03:51,  67: 0-01 0-82 0-7D           < 1: Pan Right>
03:51,  68: 1-01                     < 1: ACK>
04:04,  98: 1-01 1-B3 1-4C           < 1: Boundry Crossing 4>
04:04,  99: 0-01 0-97 0-68           < 1: ACK to dome>
04:13, 125: 0-01 0-83 0-7C           < 1: Pan Stop>
04:13, 126: 1-01                     < 1: ACK>
04:15, 132: 0-01 0-84 0-7B           < 1: Tilt Up>
04:15, 133: 1-01                     < 1: ACK>
04:21, 152: 0-01 0-86 0-79           < 1: Tilt Stop>
04:21, 153: 1-01                     < 1: ACK>
04:24, 163: 0-01 0-82 0-7D           < 1: Pan Right>
04:24, 164: 1-01                     < 1: ACK>

04:24, 166: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
04:25, 167: 1-01 1-F5 1-0A           < 1: Dome Query Response>
04:25, 168: 0-01 0-97 0-68           < 1: ACK to dome>
04:42, 206: 1-01 1-B0 1-4F           < 1: Boundry Crossing 1>
04:42, 208: 1-01 1-B0 1-4F           < 1: Boundry Crossing 1>
04:42, 209: 0-01 0-97 0-68           < 1: ACK to dome>
04:48, 227: 0-01 0-83 0-7C           < 1: Pan Stop>
04:48, 228: 1-01                     < 1: ACK>
04:50, 233: 0-01 0-85 0-7A           < 1: Tilt Down>
04:50, 234: 1-01                     < 1: ACK>
04:50, 235: 0-01 0-86 0-79           < 1: Tilt Stop>
04:50, 236: 1-01                     < 1: ACK>
04:53, 244: 0-01 0-85 0-7A           < 1: Tilt Down>
04:53, 245: 1-01                     < 1: ACK>
04:53, 246: 0-01 0-86 0-79           < 1: Tilt Stop>
04:53, 247: 1-01                     < 1: ACK>
04:54, 250: 0-01 0-85 0-7A           < 1: Tilt Down>
04:54, 251: 1-01                     < 1: ACK>
04:54, 252: 0-01 0-86 0-79           < 1: Tilt Stop>
04:54, 253: 1-01                     < 1: ACK>
```

```
04:55, 256: 0-01 0-85 0-7A           < 1: Tilt Down>
04:55, 257: 1-01                     < 1: ACK>
04:56, 260: 0-01 0-86 0-79           < 1: Tilt Stop>
04:56, 261: 1-01                     < 1: ACK>
04:59, 268: 0-01 0-82 0-7D           < 1: Pan Right>
04:59, 269: 1-01                     < 1: ACK>
04:59, 270: 0-01 0-83 0-7C           < 1: Pan Stop>
04:59, 271: 1-01                     < 1: ACK>
05:00, 274: 0-01 0-82 0-7D           < 1: Pan Right>
05:00, 275: 1-01                     < 1: ACK>
05:01, 279: 0-01 0-83 0-7C           < 1: Pan Stop>
05:01, 280: 1-01                     < 1: ACK>
05:06, 295: 0-01 0-82 0-7D           < 1: Pan Right>
05:06, 296: 1-01                     < 1: ACK>

05:22, 344: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
05:23, 345: 1-01 1-F5 1-0A           < 1: Dome Query Response>
05:23, 346: 0-01 0-97 0-68           < 1: ACK to dome>
05:33, 371: 0-01 0-83 0-7C           < 1: Pan Stop>
05:33, 372: 1-01                     < 1: ACK>
05:35, 377: 0-01 0-84 0-7B           < 1: Tilt Up>
05:35, 378: 1-01                     < 1: ACK>
05:40, 389: 0-01 0-86 0-79           < 1: Tilt Stop>
05:40, 390: 1-01                     < 1: ACK>
05:42, 396: 0-01 0-82 0-7D           < 1: Pan Right>
05:42, 397: 1-01                     < 1: ACK>
05:47, 412: 1-01 1-B1 1-4E           < 1: Boundry Crossing 2>
05:47, 413: 0-01 0-97 0-68           < 1: ACK to dome>
06:11, 473: 0-01 0-83 0-7C           < 1: Pan Stop>
06:11, 474: 1-01                     < 1: ACK>
06:13, 481: 0-01 0-85 0-7A           < 1: Tilt Down>
06:13, 482: 1-01                     < 1: ACK>
06:19, 501: 0-01 0-86 0-79           < 1: Tilt Stop>
06:19, 502: 1-01                     < 1: ACK>

06:20, 507: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
06:21, 508: 1-01 1-F5 1-0A           < 1: Dome Query Response>
06:21, 509: 0-01 0-97 0-68           < 1: ACK to dome>
06:21, 510: 0-01 0-82 0-7D           < 1: Pan Right>
06:21, 511: 1-01                     < 1: ACK>
06:59, 601: 1-01 1-B2 1-4D           < 1: Boundry Crossing 3>
06:59, 602: 0-01 0-97 0-68           < 1: ACK to dome>
07:13, 642: 0-01 0-83 0-7C           < 1: Pan Stop>
07:13, 643: 1-01                     < 1: ACK>
07:15, 650: 0-01 0-84 0-7B           < 1: Tilt Up>
07:15, 651: 1-01                     < 1: ACK>
07:17, 658: 0-01 0-86 0-79           < 1: Tilt Stop>
07:17, 659: 1-01                     < 1: ACK>

07:18, 664: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
07:19, 665: 1-01 1-F5 1-0A           < 1: Dome Query Response>
07:19, 666: 0-01 0-97 0-68           < 1: ACK to dome>
```

```
07:22, 673: 0-01 0-8B 0-74           < 1: Zoom Out>
07:22, 674: 1-01                     < 1: ACK>
07:24, 682: 0-01 0-8C 0-73           < 1: Zoom Stop>
07:24, 683: 1-01                     < 1: ACK>
07:27, 691: 0-01 0-8A 0-75           < 1: Zoom In>
07:27, 692: 1-01                     < 1: ACK>
07:33, 706: 0-01 0-8C 0-73           < 1: Zoom Stop>
07:33, 707: 1-01                     < 1: ACK>
07:39, 722: 0-01 0-84 0-7B           < 1: Tilt Up>
07:39, 723: 1-01                     < 1: ACK>
07:40, 727: 0-01 0-86 0-79           < 1: Tilt Stop>
07:40, 728: 1-01                     < 1: ACK>
07:40, 729: 0-01 0-84 0-7B           < 1: Tilt Up>
07:40, 730: 1-01                     < 1: ACK>
07:41, 734: 0-01 0-86 0-79           < 1: Tilt Stop>
07:41, 735: 1-01                     < 1: ACK>
07:46, 748: 0-01 0-8A 0-75           < 1: Zoom In>
07:46, 749: 1-01                     < 1: ACK>
07:49, 756: 0-01 0-8C 0-73           < 1: Zoom Stop>
07:49, 757: 1-01                     < 1: ACK>
07:55, 771: 0-01 0-82 0-7D           < 1: Pan Right>
07:55, 772: 1-01                     < 1: ACK>
08:11, 818: 0-01 0-83 0-7C           < 1: Pan Stop>
08:11, 819: 1-01                     < 1: ACK>
08:14, 829: 0-01 0-8B 0-74           < 1: Zoom Out>
08:14, 830: 1-01                     < 1: ACK>

08:16, 838: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
08:17, 839: 1-01 1-F5 1-0A           < 1: Dome Query Response>
08:17, 840: 0-01 0-97 0-68           < 1: ACK to dome>
08:24, 857: 0-01 0-8C 0-73           < 1: Zoom Stop>
08:24, 858: 1-01                     < 1: ACK>
08:25, 862: 0-01 0-82 0-7D           < 1: Pan Right>
08:25, 863: 1-01                     < 1: ACK>
08:29, 873: 0-01 0-83 0-7C           < 1: Pan Stop>
08:29, 874: 1-01                     < 1: ACK>
08:31, 879: 0-01 0-85 0-7A           < 1: Tilt Down>
08:31, 880: 1-01                     < 1: ACK>
08:36, 892: 0-01 0-86 0-79           < 1: Tilt Stop>
08:36, 893: 1-01                     < 1: ACK>
08:37, 897: 0-01 0-85 0-7A           < 1: Tilt Down>
08:37, 898: 1-01                     < 1: ACK>
08:38, 902: 0-01 0-86 0-79           < 1: Tilt Stop>
08:38, 903: 1-01                     < 1: ACK>

09:14, 999: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
09:15,   0: 1-01 1-F5 1-0A           < 1: Dome Query Response>
09:15,   1: 0-01 0-97 0-68           < 1: ACK to dome>
09:38,  57: 0-01 0-B8 0-47           < 1: Pattern End>
09:38,  58: 1-01                     < 1: ACK>

10:12, 148: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
```

```
10:13, 149: 1-01 1-F5 1-0A           < 1: Dome Query Response>
10:13, 150: 0-01 0-97 0-68           < 1: ACK to dome>
10:25, 179: 0-01 0-B0 0-4F           < 1: Run Pattern 1>
10:25, 180: 1-01                     < 1: ACK>
10:26, 183: 1-01 1-B3 1-4C           < 1: Boundry Crossing 4>
10:26, 184: 0-01 0-97 0-68           < 1: ACK to dome>
10:34, 204: 1-01 1-B2 1-4D           < 1: Boundry Crossing 3>
10:34, 205: 0-01 0-97 0-68           < 1: ACK to dome>
10:46, 233: 1-01 1-B1 1-4E           < 1: Boundry Crossing 2>
10:46, 234: 0-01 0-97 0-68           < 1: ACK to dome>
10:56, 261: 1-01 1-B0 1-4F           < 1: Boundry Crossing 1>
10:56, 262: 0-01 0-97 0-68           < 1: ACK to dome>
11:02, 280: 1-01 1-B3 1-4C           < 1: Boundry Crossing 4>
11:02, 281: 0-01 0-97 0-68           < 1: ACK to dome>
11:07, 297: 1-01 1-B2 1-4D           < 1: Boundry Crossing 3>
11:07, 298: 0-01 0-97 0-68           < 1: ACK to dome>

11:10, 309: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
11:11, 310: 1-01 1-F5 1-0A           < 1: Dome Query Response>
11:11, 311: 0-01 0-97 0-68           < 1: ACK to dome>
11:19, 331: 1-01 1-B1 1-4E           < 1: Boundry Crossing 2>
11:19, 332: 0-01 0-97 0-68           < 1: ACK to dome>
11:29, 354: 1-01 1-B0 1-4F           < 1: Boundry Crossing 1>
11:29, 355: 0-01 0-97 0-68           < 1: ACK to dome>
11:36, 374: 1-01 1-B3 1-4C           < 1: Boundry Crossing 4>
11:36, 375: 0-01 0-97 0-68           < 1: ACK to dome>
11:39, 383: 1-01 1-B2 1-4D           < 1: Boundry Crossing 3>
11:39, 384: 0-01 0-97 0-68           < 1: ACK to dome>
11:51, 414: 1-01 1-B1 1-4E           < 1: Boundry Crossing 2>
11:51, 415: 0-01 0-97 0-68           < 1: ACK to dome>
12:01, 443: 1-01 1-B0 1-4F           < 1: Boundry Crossing 1>
12:01, 444: 0-01 0-97 0-68           < 1: ACK to dome>
12:08, 466: 1-01 1-B3 1-4C           < 1: Boundry Crossing 4>
12:08, 467: 0-01 0-97 0-68           < 1: ACK to dome>

12:08, 469: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
12:09, 470: 1-01 1-F5 1-0A           < 1: Dome Query Response>
12:09, 471: 0-01 0-97 0-68           < 1: ACK to dome>
12:12, 478: 1-01 1-B2 1-4D           < 1: Boundry Crossing 3>
12:12, 479: 0-01 0-97 0-68           < 1: ACK to dome>
12:24, 508: 1-01 1-B3 1-4C           < 1: Boundry Crossing 4>
12:24, 509: 0-01 0-97 0-68           < 1: ACK to dome>
12:29, 522: 1-01 1-B2 1-4D           < 1: Boundry Crossing 3>
12:29, 523: 0-01 0-97 0-68           < 1: ACK to dome>
12:41, 552: 1-01 1-B1 1-4E           < 1: Boundry Crossing 2>
12:41, 553: 0-01 0-97 0-68           < 1: ACK to dome>
12:45, 563: 0-01 0-A3 0-5C           < 1: Accept New Pattern>
12:45, 564: 1-01                     < 1: ACK>
12:45, 565: 0-40 0-98 0-28           <64: Start Temp No Xmit>
12:45, 566: 0-40 0-98 0-28           <64: Start Temp No Xmit>
12:45, 567: 0-40 0-98 0-28           <64: Start Temp No Xmit>
12:45, 568: 0-01 0-A4 0-5B           < 1: Dump Dome Memory>
```

```
341 12:45, 569: 1-01 1-CA 1-0E 1-80 1-0D 1-56 1-1F 1-50 1-1A 1-5F  <  1- 10>
342             1-08 1-5F 1-C0 1-00 1-00 1-0B 1-54 1-1F 1-50 1-1A  < 11- 20>
343             1-5F 1-3A 1-EC 1-93 1-00 1-00 1-07 1-E0 1-1F 1-50  < 21- 30>
344             1-1A 1-5F 1-2A 1-FA 1-0D 1-00 1-00 1-0D 1-6C 1-1F  < 31- 40>
345             1-50 1-1A 1-5F 1-0E 1-F3 1-9E 1-07 1-86 1-73 1-05  < 41- 50>
346             1-E2 1-19 1-03 1-6D 1-90 1-00 1-94 1-6C 1-72 1-00  < 51- 60>
347             1-00 1-00 1-0E 1-10 1-1F 1-50 1-1A 1-5F 1-08 1-9D  < 61- 70>
348             1-10 1-28 1-20 1-7C 1-30 1-81 1-50 1-9C 1-10 1-A6  < 71- 80>
349             1-21 1-01 1-41 1-09 1-51 1-0B 1-41 1-13 1-51 1-17  < 81- 90>
350             1-41 1-18 1-51 1-1B 1-41 1-1D 1-51 1-21 1-11 1-2D  < 91-100>
351             1-21 1-2F 1-11 1-32 1-21 1-35 1-11 1-4A 1-21 1-B4  <101-110>
352             1-31 1-BA 1-51 1-CD 1-11 1-D5 1-22 1-44 1-42 1-4D  <111-120>
353             1-52 1-67 1-12 1-6D 1-23 1-34 1-33 1-3B 1-53 1-43  <121-130>
354             1-A3 1-58 1-B3 1-5D 1-93 1-69 1-B3 1-82 1-33 1-98  <131-140>
355             1-53 1-9C 1-33 1-9E 1-53 1-9F 1-93 1-B2 1-B3 1-C0  <141-150>
356             1-13 1-D4 1-24 1-14 1-A4 1-1F 1-B4 1-45 1-14 1-4C  <151-160>
357             1-24 1-5A 1-44 1-61 1-54 1-73 1-44 1-7A 1-54 1-7C  <161-170>
358             1-F5 1-64 1-CA 1-0E 1-01 1-00 1-00 1-00 1-00 1-00  <171-180>
359             1-00 1-00 1-00 1-00 1-00 1-DF 1-FF 1-21 1-0E 1-02  <181-190>
360             1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00  <191-200>
361             1-DF 1-FF 1-20 1-A7 < 1: Dome memory dump msg length = 204>
362 12:45, 570: 0-40 0-99 0-27           <64: End Temp No Xmit>
363 12:45, 571: 0-40 0-99 0-27           <64: End Temp No Xmit>
364 12:45, 572: 0-40 0-99 0-27           <64: End Temp No Xmit>
```

1.8.12.2 With an AD2083/02 translator

In this example I have combined three different actions:

1. Clear out a pattern.
2. Program a pattern
3. Run the pattern two or three ways.

Mostly because they all appeared to be related.

Notes

1. This example uses an UltraDome, in my numbering scheme it is dome #4.
2. The only information deleted from this data capture is the constant POLLing of other domes.
3. In lines 2 → 17 the old pattern is cleared by sending a "41 SET SHOT" command from the AD2083/02.
4. In lines 19 → 20 the dome is told to start accepting commands to store as a pattern.
5. In lines 21 → 25 the new pattern is entered. This pattern is to have the dome move to the right for four seconds and then stop.

6. In lines 27 → 28 the dome is told to end the pattern with a “33 SET SHOT” command from the AD2083/02.
7. In lines 30 → 31 the dome is told to accept the pattern with a “34 SET SHOT” command from the AD2083/02.
8. In lines 33 → 46 the new pattern is run using a “`21 call shot`” command from the AD2083/02, twice. Note that on lines 42 → 43 that the dome gets and responds to a POLL message.
9. In lines 48 → 55 the dome is told to automatically the pattern with a “25 CALL SHOT” command from the AD2083/02. (There was no obvious repeating of the pattern, but it did do the pattern once.)
10. Finally in lines 57 → 60 there is a set of ZOOM OUT/ZOOM STOP commands that resulted from my hitting the zoom button.

```
 1 <20JUN01D.CAP>
 2 00:23,  72: 0-06 0-A0 0-5A          < 6: Define Pattern 1>
 3 00:23,  73: 1-06                    < 6: ACK>
 4 00:23,  74: 1-06                    < 6: ACK>
 5 00:23,  75: 0-06 0-8E 0-6C          < 6: Fastest (Rail right)>
 6 00:23,  76: 0-06 0-B8 0-42          < 6: Pattern End>
 7 00:23,  77: 1-06                    < 6: ACK>
 8 00:23,  78: 0-06 0-B8 0-42          < 6: Pattern End>
 9 00:23,  79: 1-06                    < 6: ACK>
10 00:23,  80: 0-06 0-A3 0-57          < 6: Accept New Pattern>
11 00:23,  81: 0-06 0-8F 0-6B          < 6: Fast Stop>
12 00:23,  82: 0-06 0-8F 0-6B          < 6: Fast Stop>
13 00:23,  83: 0-06 0-8F 0-6B          < 6: Fast Stop>
14 00:23,  84: 1-06                    < 6: ACK>
15 00:23,  85: 1-06                    < 6: ACK>
16 00:23,  86: 1-06                    < 6: ACK>
17 00:23,  87: 1-06                    < 6: ACK>
18 00:31,  96: 0-06 0-A0 0-5A          < 6: Define Pattern 1>
19 00:31,  97: 1-06                    < 6: ACK>
20 00:32,  99: 0-06 0-82 0-78          < 6: Pan Right>
21 00:32, 100: 1-06                    < 6: ACK>
22 00:36, 105: 0-06 0-83 0-77          < 6: Pan Stop>
23 00:36, 106: 1-06                    < 6: ACK>
24 00:42, 113: 0-06 0-B8 0-42          < 6: Pattern End>
25 00:42, 114: 1-06                    < 6: ACK>
26 00:46, 119: 0-06 0-A3 0-57          < 6: Accept New Pattern>
27 00:46, 120: 1-06                    < 6: ACK>
28 00:54, 129: 0-06 0-B0 0-4A          < 6: Run Pattern 1>
29 00:54, 130: 1-06                    < 6: ACK>
30 01:08, 145: 1-06 1-B5 1-45          < 6: Pattern Done>
31 01:08, 146: 0-06 0-97 0-63          < 6: ACK to dome>
32 01:32, 171: 0-06 0-B0 0-4A          < 6: Run Pattern 1>
33 01:32, 172: 1-06                    < 6: ACK>
34
```

```
01:38, 180: 0-06 0-94 0-66          < 6: Poll, Dome type query>
01:39, 181: 1-06 1-F5 1-05          < 6: Dome Query Response>
01:45, 188: 1-06 1-B5 1-45          < 6: Pattern Done>
01:45, 189: 0-06 0-97 0-63          < 6: ACK to dome>
01:52, 197: 0-06 0-9A 0-60          < 6: Faster>
01:52, 198: 1-06                    < 6: ACK>
01:52, 199: 1-06                    < 6: ACK>
01:52, 200: 0-06 0-AC 0-4E          < 6: Go To Pattern 1 Start>
01:52, 201: 1-06                    < 6: ACK>
01:52, 202: 0-06 0-B0 0-4A          < 6: Run Pattern 1>
01:52, 203: 1-06                    < 6: ACK>
01:52, 204: 0-06 0-9B 0-5F          < 6: Faster Stop>
02:34, 247: 0-06 0-8B 0-6F          < 6: Zoom Out>
02:34, 248: 1-06                    < 6: ACK>
02:34, 249: 0-06 0-8C 0-6E          < 6: Zoom Stop>
02:34, 250: 1-06                    < 6: ACK>
```

1.8.13 Typical dome pattern call sequence

1.8.13.1 With an RC216 controller

This data was collected at a customer site. The customer had several cameras active and this example has been edited to remove extraneous information. Most of the extraneous information was the constant presence of "POLL"ing messages.

```
<30MAY01A.CAP>
06:34, 623: 0-11 0-9E 0-51          <17: On Air>
06:34, 624: 1-11                    <17: ACK>
06:34, 625: 0-11 0-9A 0-55          <17: Faster>
06:34, 626: 1-11                    <17: ACK>
06:34, 627: 0-11 0-B0 0-3F          <17: Run Pattern 1>
06:34, 628: 1-11                    <17: ACK>
06:34, 629: 0-11 0-9B 0-54          <17: Faster Stop>
06:34, 630: 1-11                    <17: ACK>
06:35, 632: 1-11 1-B0 1-3F          <17: Boundry Crossing 1>
06:35, 633: 0-11 0-97 0-58          <17: ACK to dome>
06:47, 658: 1-11 1-B1 1-3E          <17: Boundry Crossing 2>
06:47, 659: 0-11 0-97 0-58          <17: ACK to dome>

07:10, 723: 0-11 0-94 0-5B          <17: Poll, Dome type query>
07:11, 724: 1-11 1-F5 1-FA          <17: Dome Query Response>
07:11, 727: 1-11 1-B2 1-3D          <17: Boundry Crossing 3>
07:12, 730: 0-11 0-97 0-58          <17: ACK to dome>
07:33, 773: 1-11 1-B3 1-3C          <17: Boundry Crossing 4>
07:33, 774: 0-11 0-97 0-58          <17: ACK to dome>

07:58, 842: 0-11 0-94 0-5B          <17: Poll, Dome type query>
07:59, 843: 1-11 1-F5 1-FA          <17: Dome Query Response>
07:59, 844: 1-11 1-B0 1-3F          <17: Boundry Crossing 1>
07:59, 845: 0-11 0-97 0-58          <17: ACK to dome>
08:10, 870: 1-11 1-B1 1-3E          <17: Boundry Crossing 2>
```

```
27 08:10, 871: 0-11 0-97 0-58          <17: ACK to dome>
28 08:33, 919: 1-11 1-B2 1-3D          <17: Boundry Crossing 3>
29 08:33, 920: 0-11 0-97 0-58          <17: ACK to dome>
30
31 08:46, 963: 0-11 0-94 0-5B          <17: Poll, Dome type query>
32 08:47, 964: 1-11 1-F5 1-FA          <17: Dome Query Response>
33 08:56, 985: 1-11 1-B3 1-3C          <17: Boundry Crossing 4>
34 08:56, 986: 0-11 0-97 0-58          <17: ACK to dome>
35 09:21,  38: 1-11 1-B0 1-3F          <17: Boundry Crossing 1>
36 09:21,  39: 0-11 0-97 0-58          <17: ACK to dome>
37
38 09:32,  74: 1-11 1-B1 1-3E          <17: Boundry Crossing 2>
39 09:32,  75: 0-11 0-97 0-58          <17: ACK to dome>
40 09:33,  78: 0-11 0-9F 0-50          <17: Reset On Air>
41 09:33,  79: 1-11                    <17: ACK>
```

1.8.13.2 Initiating a continous pattern with an RC216 controller

```
1 <14JAN02A.CAP>
2 10:35,  2932: 0-06 0-9E 0-5C          < 6: On Air>
3 10:35,  2933: 1-06                    < 6: ACK>
4 10:35,  2934: 0-06 0-9A 0-60          < 6: Faster ("Fast")>
5 10:35,  2935: 1-06                    < 6: ACK>
6 10:35,  2936: 0-06 0-B0 0-4A          < 6: Run Pattern 1>
7 10:35,  2937: 1-06                    < 6: ACK>
8 10:35,  2938: 0-06 0-9B 0-5F          < 6: Faster Stop ("Fast" button up)>
9 10:35,  2939: 1-06                    < 6: ACK>
```

1.8.13.3 Setting a pattern with a dome and an RC216

Notes

1. Line 2 starts the pattern definition sequence and is followed by an ACK from the dome.
2. As usual there are many POLLs that will usually be ignored.
3. Lines 8 ⟶ 26 consists of many motion commands that comprise the pattern being entered into the SPectra.
4. In line 29, 30 the pattern is ended and ACKed.
5. In line 33, 34 an attempt is made to run the new pattern is made. As the Spectra does not have this capability the, already saved, pattern is run.
6. Eventually in line 50, 51 a PAN STOP command is sent from the controller to stop the “test pattern run”.
7. In line 53, 54 an ACCEPT NEW PATTERN is sent from the controller. (This is ignored by the Spectra.)

What happens if the new pattern is not accepted?

8. In line 55 the controller issues a TEMP NO XMIT command so that the following DUMP DOME MEMORY command will not be messed up by asynchronous dome messages.
9. In line 56 the controller requests a memory dump.
10. In line 57 the Spectra sends out a “canned” memory dump which was copied from dome #5 (a SpeedDome).
11. In line 58 the controller reeneables asynchronous messages.
12. And on line 59 the dome (Spectra) has its memory dump ACKnowledged.

```
 1 <07FEB02A.CAP>
 2 28:05,  7274: 0-2C 0-A0 0-34          <44: Define Pattern 1>
 3 28:05,  7275: 1-2C                    <44: ACK>
 4 28:05,  7276: 0-12 0-94 0-5A          <18: Poll, Dome type query>
 5 28:06,  7277: 0-13 0-94 0-59          <19: Poll, Dome type query>
 6 28:06,  7278: 0-0C 0-97 0-5D          <12: ACK to dome>
 7 28:06,  7279: 0-14 0-94 0-58          <20: Poll, Dome type query>
 8 28:07,  7280: 0-2C 0-82 0-52          <44: Pan Right>
 9 28:07,  7281: 1-2C                    <44: ACK>
10 28:07,  7282: 0-15 0-94 0-57          <21: Poll, Dome type query>
11 28:07,  7283: 0-16 0-94 0-56          <22: Poll, Dome type query>
12 28:08,  7284: 0-2C 0-8E 0-46          <44: Fastest (Rail right)>
13 28:08,  7285: 1-2C                    <44: ACK>
14 28:08,  7286: 0-0E 0-97 0-5B          <14: ACK to dome>
15 28:08,  7287: 0-2C 0-83 0-51          <44: Pan Stop>
16 28:08,  7288: 1-2C                    <44: ACK>
17 28:08,  7289: 0-17 0-94 0-55          <23: Poll, Dome type query>
18 28:08,  7290: 0-2C 0-8B 0-49          <44: Zoom Out>
19 28:08,  7291: 1-2C                    <44: ACK>
20 <100 movement, poll and acks deleted. All motion commands are fixed speed types>
21 28:22,  7392: 1-2C                    <44: ACK>
22 28:22,  7393: 0-34 0-94 0-38          <52: Poll, Dome type query>
23 28:23,  7394: 0-0E 0-97 0-5B          <14: ACK to dome>
24 28:23,  7395: 0-35 0-94 0-37          <53: Poll, Dome type query>
25 28:23,  7396: 0-36 0-94 0-36          <54: Poll, Dome type query>
26 28:24,  7397: 0-2C 0-83 0-51          <44: Pan Stop>
27 28:24,  7398: 1-2C                    <44: ACK>
28 28:24,  7399: 0-37 0-94 0-35          <55: Poll, Dome type query>
29 28:24,  7400: 0-2C 0-B8 0-1C          <44: Pattern End>
30 28:24,  7401: 1-2C                    <44: ACK>
31 <4 polls deleted>
32 28:26,  7406: 0-0C 0-97 0-5D          <12: ACK to dome>
33 28:26,  7407: 0-2C 0-B3 0-21          <44: Run New Pattern>
34 28:26,  7408: 1-2C                    <44: ACK>
35 <16 polls deleted>
36 28:34,  7425: 0-0D 0-97 0-5C          <13: ACK to dome>
37 <13 polls deleted>
38 28:41,  7439: 0-0D 0-97 0-5C          <13: ACK to dome>
39 28:41,  7440: 0-0E 0-97 0-5B          <14: ACK to dome>
40 28:41,  7441: 0-59 0-94 0-13          <89: Poll, Dome type query>
41 28:41,  7442: 0-11 0-97 0-58          <17: ACK to dome>
42 <6 polls deleted>
```

```
28:44,  7449: 0-60 0-94 0-0C           <96: Poll, Dome type query>

28:45,  7451: 0-01 0-94 0-6B           < 1: Poll, Dome type query>
28:45,  7452: 1-01 1-F5 1-0A           < 1: Dome Query Response>
<7 polls deleted>
28:49,  7460: 0-11 0-97 0-58           <17: ACK to dome>
<4 polls deleted>
28:51,  7465: 0-2C 0-83 0-51           <44: Pan Stop>
28:51,  7466: 1-2C                     <44: ACK>
<10 polls deleted>
28:56,  7477: 0-2C 0-A3 0-31           <44: Accept New Pattern>
28:56,  7478: 1-2C                     <44: ACK>
28:56,  7479: 0-40 0-98 0-28           <64: Start Temp No Xmit>
28:56,  7480: 0-2C 0-A4 0-30           <44: Dump Dome Memory>
28:56,  7481: 1-2C 1-66 1-0E 1-80 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-72 1-00 1-00 1-00
1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00
1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00
1-00 1-00 1-0E 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-DF 1-FF 1-22 1-0E 1-00 1-00
1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-DF 1-FF 1-21 1-0E 1-00 1-00 1-00 1-00 1-00 1-00 1-00
1-00 1-00 1-00 1-00 1-DF 1-FF 1-20 1-47           <44: Dome memory dump msg length = 104>
28:56,  7482: 0-40 0-99 0-27           <64: End Temp No Xmit>
28:56,  7483: 0-0C 0-97 0-5D           <12: ACK to dome>
```

1.9 Unique RC58 controller command sequences

1.9.1 Changing monitors

This is done with the line monitor to select camera 6 when using camera 1.

```
<22MAR01A.CAP>
15:55, 641: 0-01 0-9F 0-60           < 1: Reset On Air>
15:55, 642: 1-01                     < 1: ACK>
15:55, 643: 0-01 0-9F 0-60           < 1: Reset On Air>
15:55, 644: 1-01                     < 1: ACK>
15:55, 645: 0-01 0-9F 0-60           < 1: Reset On Air>
15:55, 646: 1-01                     < 1: ACK>
15:55, 647: 0-06 0-9E 0-5C           < 6: On Air>
15:55, 648: 1-06                     < 6: ACK>
15:55, 649: 0-06 0-9E 0-5C           < 6: On Air>
15:55, 650: 1-06                     < 6: ACK>
15:55, 651: 0-06 0-9E 0-5C           < 6: On Air>
15:55, 652: 1-06                     < 6: ACK>
```

1.10 Unique RC216 controller command sequences

1.10.1 Boundary Crossing

In the two following examples, note that boundary crossing messages are generated by the domes. Also note that the same boundary crossing ID is used several times in a row. (*Does this always happen? Even though the boundary crossing was acknowledged?*)

In the first example, note the response when a dome gets POLLed. It immediately gives out its type code and then says that it has crossed a boundary.

```
<30MAY01A.CAP>
00:39, 884: 0-03 0-94 0-69           < 3: Poll, Dome type query>
00:40, 885: 1-03 1-F5 1-08           < 3: Dome Query Response>
00:40, 886: 0-04 0-94 0-68           < 4: Poll, Dome type query>
00:40, 887: 1-04 1-F5 1-07           < 4: Dome Query Response>
00:40, 888: 1-03 1-B1 1-4C           < 3: Boundry Crossing 2>
00:40, 889: 0-03 0-97 0-66           < 3: ACK to dome>
```

In the second example note that each boundary crossing message is sent out multiple times.

```
<30MAY01A.CAP>
01:44, 196: 1-0B 1-B3 1-42           <11: Boundry Crossing 4>
01:44, 197: 0-0B 0-97 0-5E           <11: ACK to dome>
01:44, 198: 1-0B 1-B3 1-42           <11: Boundry Crossing 4>
01:44, 199: 0-0B 0-97 0-5E           <11: ACK to dome>
01:44, 200: 1-0B 1-B3 1-42           <11: Boundry Crossing 4>
01:44, 201: 0-0B 0-97 0-5E           <11: ACK to dome>
```

1.10.2 Changing domes and then goto'ing a preset

```
<30MAY01A.CAP>
12:06, 679: 0-0A 0-9F 0-57           <10: Reset On Air>
12:06, 680: 1-0A                     <10: ACK>
12:06, 681: 0-0F 0-9E 0-53           <15: On Air>
12:06, 682: 1-0F                     <15: ACK>
12:06, 683: 0-0F 0-9E 0-53           <15: On Air>
12:06, 684: 1-0F                     <15: ACK>
12:06, 685: 0-0F 0-A6 0-39 0-2C 0-48 0-D6 0-07 0-84 0-00 0-00 0-80 0-15 0-A8 <15: Goto
Position>
                                Pan Position          = 0x39 0x2C
                                Tilt Position         = 0x48 0xD6
                                Zoom Position         = 0x07 0x84
                                Digital Zoom          = 0x00 0x00
                                Iris Offset           = 0x80
                                Electronic Zoom Limit = 0x15
12:06, 686: 1-0F                     <15: ACK>
12:07, 688: 1-0F 1-B0 1-41           <15: Boundry Crossing 1>
12:07, 689: 0-0F 0-97 0-5A           <15: ACK to dome>
12:08, 692: 0-0F 0-9F 0-52           <15: Reset On Air>
12:08, 693: 1-0F                     <15: ACK>
12:08, 694: 0-10 0-9E 0-52           <16: On Air>
12:08, 695: 1-10                     <16: ACK>
12:08, 696: 0-10 0-9E 0-52           <16: On Air>
12:08, 697: 1-10                     <16: ACK>
12:08, 698: 0-10 0-A6 0-36 0-C1 0-4B 0-16 0-10 0-E4 0-00 0-00 0-80 0-15 0-69 <16: Goto
Position>
                                Pan Position          = 0x36 0xC1
                                Tilt Position         = 0x4B 0x16
                                Zoom Position         = 0x10 0xE4
                                Digital Zoom          = 0x00 0x00
                                Iris Offset           = 0x80
                                Electronic Zoom Limit = 0x15
12:08, 699: 1-10                     <16: ACK>
12:08, 701: 1-10 1-B3 1-3D           <16: Boundry Crossing 4>
12:08, 702: 0-10 0-97 0-59           <16: ACK to dome>
12:09, 704: 0-10 0-9E 0-52           <16: On Air>
12:09, 705: 1-10                     <16: ACK>
12:09, 706: 0-10 0-A6 0-36 0-C1 0-4B 0-16 0-10 0-E4 0-00 0-00 0-80 0-15 0-69 <16: Goto
Position>
                                Pan Position          = 0x36 0xC1
                                Tilt Position         = 0x4B 0x16
                                Zoom Position         = 0x10 0xE4
                                Digital Zoom          = 0x00 0x00
                                Iris Offset           = 0x80
                                Electronic Zoom Limit = 0x15
12:09, 707: 1-10                     <16: ACK>
12:09, 709: 1-10 1-B3 1-3D           <16: Boundry Crossing 4>
12:09, 710: 0-10 0-97 0-59           <16: ACK to dome>
```

1.10.3 Response to a manual “on air” command

```
<30MAY01A.CAP>
02:30, 401: 0-05 0-9F 0-5C           < 5: Reset On Air>
02:30, 402: 1-05                     < 5: ACK>
02:30, 403: 0-06 0-9E 0-5C           < 6: On Air>
02:30, 404: 1-06                     < 6: ACK>
02:31, 406: 1-06 1-B1 1-49           < 6: Boundry Crossing 2>
02:31, 407: 0-06 0-97 0-63           < 6: ACK to dome>
```

1.10.4 Response to manually selecting a dome and then doing a goto

```
<30MAY01A.CAP>
11:57, 600: 0-10 0-9E 0-52           <16: On Air>
11:57, 601: 1-10                     <16: ACK>
11:57, 602: 0-10 0-A6 0-16 0-10 0-E4 0-00 0-00 0-80 0-15 0-69 0-36 0-C1 0-4B <16: Goto
Position>
                                Pan Position          = 0x16 0x10
                                Tilt Position         = 0xE4 0x00
                                Zoom Position         = 0x00 0x80
                                Digital Zoom          = 0x15 0x69
                                Iris Offset           = 0x36
                                Electronic Zoom Limit = 0xC1
11:57, 603: 1-10                     <16: ACK>
11:58, 606: 1-10 1-B3 1-3D           <16: Boundry Crossing 4>
11:58, 607: 0-10 0-97 0-59           <16: ACK to dome>
11:58, 608: 0-10 0-94 0-5C           <16: Poll, Dome type query>
11:58, 609: 1-10 1-F5 1-FB           <16: Dome Query Response>
11:59, 614: 0-10 0-9E 0-52           <16: On Air>
11:59, 615: 1-10                     <16: ACK>
```

```
11:59, 616: 0-10 0-A6 0-36 0-C1 0-4B 0-16 0-10 0-E4 0-00 0-00 0-80 0-15 0-69 <16: Goto
Position>
                                Pan Position          = 0x36 0xC1
                                Tilt Position         = 0x4B 0x16
                                Zoom Position         = 0x10 0xE4
                                Digital Zoom          = 0x00 0x00
                                Iris Offset           = 0x80
                                Electronic Zoom Limit = 0x15
11:59, 617: 1-10                     <16: ACK>
11:59, 618: 1-10 1-B3 1-3D           <16: Boundry Crossing 4>
11:59, 619: 0-10 0-97 0-59           <16: ACK to dome>
12:00, 622: 0-10 0-9F 0-51           <16: Reset On Air>
12:00, 623: 1-10                     <16: ACK>
```

1.10.5 Running a pattern

```
<30MAY01A.CAP>
11:55, 574: 0-11 0-9E 0-51           <17: On Air>
11:55, 575: 1-11                     <17: ACK>
11:55, 576: 0-11 0-9E 0-51           <17: On Air>
11:55, 577: 1-11                     <17: ACK>
11:55, 578: 0-11 0-B0 0-3F           <17: Run Pattern 1>
11:55, 579: 1-11                     <17: ACK>
11:56, 582: 1-11 1-B3 1-3C           <17: Boundry Crossing 4>
11:56, 583: 0-11 0-97 0-58           <17: ACK to dome>
11:56, 586: 1-11 1-B0 1-3F           <17: Boundry Crossing 1>
11:56, 587: 0-11 0-97 0-58           <17: ACK to dome>
11:56, 588: 0-11 0-9F 0-50           <17: Reset On Air>
11:56, 589: 1-11                     <17: ACK>
```

1.10.6 Unremembered command sequence

```
<30MAY01A.CAP>
11:21, 450: 0-0A 0-89 0-6D           <10: Focus Stop>
11:21, 451: 1-0A                     <10: ACK>
11:21, 452: 0-0A 0-89 0-6D           <10: Focus Stop>
11:21, 453: 1-0A                     <10: ACK>
11:21, 454: 0-0A 0-8D 0-69           <10: Fast (Rail left)>
11:21, 455: 1-0A                     <10: ACK>
11:21, 456: 0-0A 0-8E 0-68           <10: Fastest (Rail right)>
11:21, 457: 1-0A                     <10: ACK>
11:22, 459: 1-0A 1-B0 1-46           <10: Boundry Crossing 1>
11:22, 460: 0-0A 0-97 0-5F           <10: ACK to dome>
11:22, 462: 1-0A 1-B3 1-43           <10: Boundry Crossing 4>
11:22, 463: 0-0A 0-97 0-5F           <10: ACK to dome>
```

1.11 Contents of a Dome Memory dump

A DUMP DOME MEMORY command (0xA4), on a SpeedDome 2000 (Dome #5) generates the following response on power up (i.e. there is nothing in dome memory):

Note that the various “Length” fields do not include themselves, nor does the overall Message Length include the Source Address field.

Memory Dump of a SpeedDome 2000 (Dome #5)

Source address	1		1	0x01
Message length	2		1	0x66 (102_{10})
Field length	3		1	0x0E (14_{10})
Data?	4	16	13	0x80 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x72 0x00 0x00 0x00 or 0x80, 8 0x00s, 0x72 and 3 more 0x00s
Field checksum	17		1	0x00
Data?	18	58	41	0x00 (41 0x00s)
Field length	59		1	0x0E (14_{10})
Data?	60	72	13	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xDF 0xFF or 11 0x00s and then two data? bytes
Field checksum	73		1	0x22
Field length	74		1	0x0E (14_{10})
Data?	75	87	13	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xDF 0xFF or 11 0x00s and then two data? bytes
Field checksum	88		1	0x21
Field length	89		1	0x0E (14_{10})
Data?	90	101	13	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xDF 0xFF or 11 0x00s and then two data? bytes
Field checksum	103		1	0x20
Message checksum	104		1	0x72

Memory Dump of a SpeedDome 2000 (Dome #5)									
1	2	3	4	5	6	7	8	9	10
1-01	1-66	1-0E	1-80	1-00	1-00	1-00	1-00	1-00	1-00
11	12	13	14	15	16	17	18	19	20
1-00	1-00	1-72	1-00	1-00	1-00	1-00	1-00	1-00	1-00
21	22	23	24	25	26	27	28	29	30
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
31	32	33	34	35	36	37	38	39	40
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
41	42	43	44	45	46	47	48	49	50
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
51	52	53	54	55	56	57	58	59	60
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-0E	1-00
61	62	63	64	65	66	67	68	69	70
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
71	72	73	74	75	76	77	78	79	80
1-DF	1-FF	1-22	1-0E	1-00	1-00	1-00	1-00	1-00	1-00
81	82	83	84	85	86	87	88	89	90
1-00	1-00	1-00	1-00	1-00	1-DF	1-FF	1-21	1-0E	1-00
91	92	93	94	95	96	97	98	99	100
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
101	102	103	104	—	—	—	—	—	—
1-DF	1-FF	1-20	1-72	Dome memory dump msg length = 104					

A DUMP DOME MEMORY command (0xA4), on a UltraDome (Dome #4) generates the following response on power up (i.e. there is nothing in dome memory):

Note that the various "Length" fields do not include themselves, nor does the overall Message Length include the Source Address field.

Memory Dump of a UltraDome (Dome #4)				
Source address	1		1	0x04
Message length	2		1	0x95 (149_{10})
Data?	3	58	56	56 0x00s
Field length	59		1	0x0E (14_{10})
Data?	60	72	13	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xDF 0xFF or 11 0x00s and then two data? bytes of 0xDF 0xFF
Field checksum	73		1	0x22
Field length	74		1	0x0E (14_{10})
Data?	75	87	13	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xDF 0xFF or 11 0x00s and then two data? bytes of 0xDF 0xFF
Field checksum	88		1	0x21
Field length	89		1	0x3E (62_{10})
Data?	90	150	61	0x02, 58 0x00s and then two data? bytes of 0xDF 0xFF
Field checksum	151		1	0x20
Message checksum	152		1	0x0D

Memory Dump of a UltraDome (Dome #4)									
1	2	3	4	5	6	7	8	9	10
1-04	1-95	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
11	12	13	14	15	16	17	18	19	20
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
21	22	23	24	25	26	27	28	29	30
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
31	32	33	34	35	36	37	38	39	40
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
41	42	43	44	45	46	47	48	49	50
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
51	52	53	54	55	56	57	58	59	60
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-0E	1-00
61	62	63	64	65	66	67	68	69	70
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
71	72	73	74	75	76	77	78	79	80
1-DF	1-FF	1-22	1-0E	1-01	1-00	1-00	1-00	1-00	1-00
81	82	83	84	85	86	87	88	89	90
1-00	1-00	1-00	1-00	1-00	1-DF	1-FF	1-21	1-3E	1-02
91	92	93	94	95	96	97	98	99	100
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
101	102	103	104	105	106	107	108	109	110
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
111	112	113	114	115	116	117	118	119	120
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
121	122	123	124	125	126	127	128	129	130
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
131	132	133	134	135	136	137	138	139	140
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00
141	142	143	144	145	146	147	148	149	150
1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-00	1-DF	1-FF
151	152	—	—	—	—	—	—	—	—
1-20	1-0D	4: Dome memory dump msg length = 152							

1.12 Unique RC32 controller command sequences

Getting the RC32 up and working was quite easy, however the commands that it sends is a cause for concern and because **none of them match any known command**. (Even though the checksum is always correct!)

1.12.1 Controller Power up sequence with an RC32

```
<02AUG01B.CAP>
00:00,  35: 0-01 0-F1 0-0E          < 1: Undefined command 1 Len = 3>
00:00,  36: 1-01                    < 1: ACK>
00:00,  37: 0-01 0-F1 0-0E          < 1: Undefined command 2 Len = 3>
00:00,  38: 1-01                    < 1: ACK>
00:00,  39: 0-01 0-F1 0-0E          < 1: Undefined command 3 Len = 3>
00:00,  40: 1-01                    < 1: ACK>
00:00,  41: 0-02 0-E0 0-1E          < 2: Aux 1,2,3,4 off>
00:00,  42: 0-02 0-E0 0-1E          < 2: Aux 1,2,3,4 off>
00:00,  43: 0-02 0-E0 0-1E          < 2: Aux 1,2,3,4 off>

00:00,  50: 0-05 0-F1 0-0A          < 5: Undefined command 4 Len = 3>
00:00,  51: 0-05 0-F1 0-0A          < 5: Undefined command 5 Len = 3>
00:00,  52: 0-05 0-F1 0-0A          < 5: Undefined command 6 Len = 3>

00:00, 131: 0-20 0-E0 0-00          <32: Aux 1,2,3,4 off>
00:00, 132: 0-20 0-E0 0-00          <32: Aux 1,2,3,4 off>
00:00, 133: 0-20 0-E0 0-00          <32: Aux 1,2,3,4 off>
00:00, 134: 0-01 0-B8 0-47          < 1: Pattern End>
00:00, 135: 1-01                    < 1: ACK>
00:00, 136: 0-01 0-B8 0-47          < 1: Pattern End>
00:00, 137: 1-01                    < 1: ACK>
00:00, 138: 0-01 0-B8 0-47          < 1: Pattern End>
00:00, 139: 1-01                    < 1: ACK>
```

1.12.2 Dome Power up sequence with an RC32

```
<02AUG01B.CAP>
00:00, 143: 1-01 1-C1 1-3E          < 1: Dome Power Up Message>
00:00, 144: 0-70                    <112: ACK., cksum bad = 0x0070, msg length = 1>
00:00, 145: 1-01 1-C1 1-3E          < 1: Dome Power Up Message>
00:00, 146: 0-70                    <112: ACK., cksum bad = 0x0070, msg length = 1>
00:00, 147: 1-01 1-C1 1-3E          < 1: Dome Power Up Message>
00:00, 148: 0-70                    <112: ACK., cksum bad = 0x0070, msg length = 1>
```

[4] $Header: d:/txb-s422/RCS/rc32.inc,v 1.2 2001-11-30 15:12:04-08 Hamilton Exp Hamilton $

1.13 Dump of the configuration buffer from a Delta Dome

There are two “new” commands that were listed in the most recent version of Sensormatic’s protocol document (8000-2694-01). These involve working with a dome’s internal “configuration” buffer. The documented contents are as follows:

Offset	Description	Length
0	Eeprom version number	6
6	Flash version number	6
12	Reserved	6
18	Device type	4
22	Device serial number	10
32	Date of manufacture	3

The first command (`0xC4`) is supposed to dump portions of the buffer. I never got “good” results with it and decided that the capability of being able to read various parts of the configuration buffer will probably be added in in future Sensormatic products. The following annotated data capture illustrates some of the problems that I had with the data.

```
<29MAY02A.CAP>
00:01,    49: 0-06 0-C9 0-31           < 6: Get software ID>
00:01,    50: 1-06 1-C9 1-06 1-07 1-01 1-25 1-07 1-03 1-06 1-EE < 6: Software ID Response>

00:04,    65: 0-06 0-C4 0-00 0-00 0-36           < 6: Undefined command 5 Len = 5>
00:04,    66: 1-06                                dome address
              1-C4                                command type
              1-36                                message length
              1-07 1-01 1-25 1-07 1-01 1-03       EEPROM version
              1-07 1-01 1-25 1-07 1-03 1-06       flash revision (U4)
              1-53 1-44 1-55 1-20 1-35 1-20       "SDS 5 "
              1-22 1-83 1-01 1-05                 device type (first two bytes)
              1-00 1-00 1-00 1-00 1-00 1-00 1-04 1-46 1-64 1-59  serial number
              1-07 1-30 1-01                      Date of manfacture
                                                  unknown/undocmented follows:
              1-01 1-00 1-22 1-83 1-21 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00
1-00 1-00 1-00 1-79           < 6: Unknown Response C4 msg length = 58>

00:07,    99: 0-06 0-C4 0-00 0-06 0-30           < 6: Undefined command 6 Len = 5>
00:07,   100: 1-06
              1-C4
              1-30
              1-07 1-01 1-25 1-07 1-03 1-06
              1-53 1-44 1-55 1-20 1-35 1-20
              1-22 1-83 1-01 1-05 1-00 1-00
              1-00 1-00 1-00 1-00
              1-04 1-46 1-64 1-59 1-07 1-30 1-01 1-01 1-00 1-22
              1-83 1-21 1-00

              1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-00 1-B7          <
6: Unknown Response C4 msg length = 52>
```

In first attemp in this listing note that I requested “`0`” bytes of data and actually got 54 bytes in. It does look as though the first part is valid data, but then what is the rest? In the second try I used the example from the protocol manual and here everything is offset by `6` bytes. Because of these anomalies, I decided to not implement it at all in the TXB-S422.

I have decided that it is too dangerous to attempt to **change** the contents of the internal configuration buffer as I might mess something up quite seriously. Thus there is no example of what happens when I try it.

1.14 Dome commands generated by an AD2083/02 translator

This table of commands and their responses was generated with an AD2083/02 and dome #5. (This information has not been rechecked at the project end. Thus there may be some important items left out or misinterpreted.)

Function	AD2078 Key(s)	Command(s) Generated	
Set preset 1	1 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 1	1 CALL	`0-06 0-A6 0-FF 0-1A 0-FF 0-00 0-FF`	
		`0-00 0-FF 0-10 0-FF 0-49 0-E6`	
	ACK	`1-06`	
Set preset 2	2 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 2	2 CALL	`0-06 0-A6 0-FF 0-00 0-FF 0-00 0-FF`	
		`0-FF 0-FF 0-40 0-FF 0-00 0-1A`	
	ACK	`1-06`	
Set preset 3	3 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 3	3 CALL	`0-06 0-A6 0-FF 0-00 0-FF 0-40 0-FF`	
		`0-00 0-FF 0-0A 0-FF 0-82 0-8D`	
	ACK	`1-06`	
Set preset 4	4 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 4	4 CALL	`0-06 0-A6 0-FF 0-7F 0-FF 0-02 0-FF`	
		`0-00 0-FF 0-20 0-FF 0-00 0-B8`	
	ACK	`1-06`	
Set preset 5	5 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 5	5 CALL	`0-06 0-A6 0-FF 0-00 0-FF 0-80 0-FF`	
		`0-20 0-FF 0-FF 0-FF 0-00 0-BA`	
	ACK	`1-06`	
Set preset 6	6 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 6	6 CALL	`0-06 0-A6 0-FF 0-00 0-FF 0-02 0-FF`	
		`0-00 0-FF 0-00 0-FF 0-00 0-57`	
	ACK	`1-06`	
Set preset 7	7 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 7	7 CALL	`0-06 0-A6 0-FF 0-00 0-FF 0-F7 0-FF`	
		`0-00 0-FF 0-00 0-FF 0-20 0-42`	
	ACK	`1-06`	
Set preset 8	8 SET	Similar to Section 1.8.9.1, page 46	

Continued on the next page.

Continued from the previous page.

Function	AD2078 Key(s)	Command(s) Generated	
Goto preset 8	8 CALL	0-06 0-A6 0-FF 0-00 0-DF 0-00 0-FF	
		0-00 0-FF 0-00 0-FF 0-5F 0-1A	
	ACK	1-06	
Set preset 9	9 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 9	9 CALL	0-06 0-A6 0-FF 0-00 0-FF 0-00 0-FF	
		0-00 0-FF 0-00 0-FF 0-40 0-19	
	ACK	1-06	
Set preset 10	10 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 10	10 CALL	0-06 0-A6 0-3D 0-6E 0-43 0-F6 0-0B	
		0-EC 0-00 0-00 0-3E 0-55 0-E6	
	ACK	1-06	
Set preset 11	11 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 11	11 CALL	0-06 0-A6 0-06 0-EE 0-0A 0-C0 0-11	
		0-D7 0-26 0-87 0-00 0-00 0-01	
	ACK	1-06	
Set preset 12	12 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 12	12 CALL	0-06 0-A6 0-00 0-FB 0-FF 0-02 0-FF	
		0-00 0-FD 0-00 0-FF 0-00 0-5D	
	ACK	1-06	
Set preset 13	13 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 13	13 CALL	0-06 0-A6 0-FF 0-84 0-FF 0-05 0-FF	
		0-00 0-FF 0-79 0-FF 0-00 0-57	
	ACK	1-06	
Set preset 14	14 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 14	14 CALL	0-06 0-A6 0-FF 0-04 0-FF 0-00 0-FF	
		0-00 0-FF 0-41 0-FF 0-46 0-CE	
	ACK	1-06	
Set preset 15	15 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 15	15 CALL	0-06 0-A6 0-FF 0-02 0-FF 0-EF 0-FF	
		0-70 0-FF 0-00 0-FF 0-08 0-F0	
	ACK	1-06	
Set preset 16	16 SET	Similar to Section 1.8.9.1, page 46	
Goto preset 16	16 CALL	0-06 0-A6 0-FF 0-00 0-FF 0-06 0-FF	
		0-02 0-FF 0-14 0-FF 0-FE 0-3F	

Continued on the next page.

Continued from the previous page.

Function	AD2078 Key(s)	Command(s) Generated
	ACK	`1-06`
Define Pattern #1	17 SET	`0-06 0-A0 0-5A`
	ACK	`1-06`
Goto Pattern #1	17 CALL	`0-06 0-AC 0-4E`
	ACK	`1-06`
Define Pattern #2	18 SET	`0-06 0-A1 0-59`
	ACK	`1-06`
Goto Pattern #2	18 CALL	`0-06 0-AD 0-4D`
	ACK	`1-06`
Define Pattern #3	19 SET	`0-06 0-A2 0-58`
	ACK	`1-06`
Goto Pattern #3	19 CALL	`0-06 0-AE 0-4C`
	ACK	`1-06`
There is no Define Pattern #4		
Goto Pattern #4	20 CALL	`0-06 0-AF 0-4B`
	ACK	`1-06`
Run Pattern #1	21 CALL	`0-06 0-B0 0-4A`
	ACK	`1-06`
Run Pattern #2	22 CALL	`0-06 0-B1 0-49`
	ACK	`1-06`
Run Pattern #3	23 CALL	`0-06 0-B2 0-48`
	ACK	`1-06`
Run Pattern #4	24 CALL	`0-06 0-B3 0-47`
	ACK	`1-06`
Auto Repeat Pattern #1 (See Section 1.4.3.6, page 24)	25 CALL	`0-06 0-9A 0-60 <Faster>`
		`1-06 <ACK>`
		`0-06 0-AC 0-4E <Unknown AC>`
		`1-06 <ACK>`
		`0-06 0-B0 0-4A <Run Pattern 1>`
		`1-06 <ACK>`
		`0-06 0-9B 0-5F <Faster Stop>`

Continued on the next page.

Continued from the previous page.

Function	AD2078 Key(s)	Command(s) Generated
		1-06 <ACK>
Auto Repeat Pattern #2 (See Section 1.4.3.6, page 24)	26 CALL	0-06 0-9A 0-60 <Faster> 1-06 <ACK> 0-06 0-AD 0-4D <Unknown AD> 1-06 <ACK> 0-06 0-B1 0-49 <Run Pattern 2> 1-06 <ACK> 0-06 0-9B 0-5F <Faster Stop> 1-06 <ACK>
Auto Repeat Pattern #3 (See Section 1.4.3.6, page 24)	27 CALL	0-06 0-9A 0-60 <Faster> 1-06 <ACK> 0-06 0-AE 0-4C <Unknown AE> 1-06 <ACK> 0-06 0-B2 0-48 <Run Pattern 3> 1-06 <ACK> 0-06 0-9B 0-5F <Faster Stop> 1-06 <ACK>
Auto Repeat Pattern #4 (See Section 1.4.3.6, page 24)	28 CALL	0-06 0-9A 0-60 <Faster> 1-06 <ACK> 0-06 0-AF 0-4B <Unknown AF> 1-06 <ACK> 0-06 0-B3 0-47 <Run New Pattern> 1-06 <ACK> 0-06 0-9B 0-5F <Faster Stop> 1-06 <ACK>
End Pattern Definition	33 SET ACK	0-06 0-B8 0-42 1-06
Accept (save) New Pattern	34 SET ACK	0-06 0-A3 0-57 1-06
Flip	40 CALL	0-06 0-8D 0-6D

Continued on the next page.

Continued from the previous page.

Function	AD2078 Key(s)	Command(s) Generated
(See Section 1.4.3.5, page 23)	ACK ACK	1-06 0-06 0-8E 0-6C 1-06
Clear Pattern #1 (See Section 1.4.3.4, page 22)	41 SET	0-06 0-A0 0-5A <Define Pattern 1> 1-06 <ACK> 0-06 0-8E 0-6C <Fastest (Rail right)> 1-06 <ACK> 0-06 0-B8 0-42 <Pattern End> 1-06 <ACK> 0-06 0-A3 0-57 <Accept New Pattern> 1-06 <ACK> 0-06 0-8F 0-6B <Fast Stop> 1-06 <ACK>
Clear Pattern #2 (See Section 1.4.3.4, page 22)	42 SET	0-06 0-A1 0-59 <Define Pattern 2> 1-06 <ACK> 0-06 0-8E 0-6C <Fastest (Rail right)> 1-06 <ACK> 0-06 0-B8 0-42 <Pattern End> 1-06 <ACK> 0-06 0-A3 0-57 <Accept New Pattern> 1-06 <ACK> 0-06 0-8F 0-6B <Fast Stop> 1-06 <ACK>
Clear Pattern #3 (See Section 1.4.3.4, page 22)	43 SET	0-06 0-A2 0-58 <Define Pattern 3> 1-06 <ACK> 0-06 0-8E 0-6C <Fastest (Rail right)>

Continued on the next page.

Continued from the previous page.

Function	AD2078 Key(s)	Command(s) Generated

		`1-06 <ACK>`
		`0-06 0-B8 0-42 <Pattern End>`
		`1-06 <ACK>`
		`0-06 0-A3 0-57 <Accept New Pattern>`
		`1-06 <ACK>`
		`0-06 0-8F 0-6B <Fast Stop>`
		`1-06 <ACK>`

Set Switch 1 Normally Closed	51 SET	`0-06 0-95 0-65`
		`1-06 1-00 1-FA`

Set Switch 1 Normally Open	52 SET	`0-06 0-95 0-65`
		`1-06 1-00 1-FA`

Set Switch 2 Normally Closed	53 SET	`0-06 0-95 0-65`
		`1-06 1-00 1-FA`

Set Switch 2 Normally Open	54 SET	`0-06 0-95 0-65`
		`1-06 1-00 1-FA`

Set Switch 3 Normally Closed	61 SET	`0-06 0-95 0-65`
		`1-06 1-00 1-FA`

Set Switch 3 Normally Open	62 SET	`0-06 0-95 0-65`
		`1-06 1-00 1-FA`

Set Switch 4 Normally Closed	63 SET	`0-06 0-95 0-65`
		`1-06 1-00 1-FA`

Set Switch 4 Normally Open	64 SET	`0-06 0-95 0-65`
		`1-06 1-00 1-FA`

Reset Dome (See Section 1.4.3.7, page 24)	66 CALL, 66 SET	`0-06 0-9A 0-60 <Faster>`
		`1-06 <ACK>`

Continued on the next page.

Continued from the previous page.

Function	AD2078 Key(s)	Command(s) Generated

		`0-06 0-8B 0-6F <Zoom Out>`
		`1-06 <ACK>`
		`0-06 0-88 0-72 <Focus Far>`
		`1-06 <ACK>`
		`0-06 0-90 0-6A <Iris Open>`
		`1-06 <ACK>`

		`1-06 1-C1 1-39 <Undefined message>`
		`0-06 0-97 0-63 <ACK to dome>`
		`0-06 0-80 0-7A <Unknown 80>`
		`1-06 <ACK>`
		`1-06 1-C4 1-36 <Unknown Response C4>`
		`0-06 0-97 0-63 <ACK to dome>`
		`0-06 0-95 0-65 <Undefined command>`
		`1-06 1-00 1-FA <Undefined message>`

Enter V-Phase Adjust Mode	67 SET	`0-06 0-9A 0-60`
	ACK	`1-06`
To increase V-Phase use IRIS OPEN, (See Section 1.4.3.2, page 22) To decrease V-Phase use IRIS CLOSE, (See Section 1.4.3.3, page 22)		
Exit V-Phase Adjust Mode	68 SET	`0-06 0-9B 0-5F`
	ACK	`1-06`

Return to Auto-Iris/Auto Focus	69 CALL	`0-06 0-90 0-6A < 6:  Iris Open>`
		`1-06 < 6:  ACK>`
		`0-06 0-91 0-69 < 6:  Iris Close>`
		`1-06 < 6:  ACK>`

Reset iris (See Section 1.4.3.1, page 21)	69 SET	`0-06 0-90 0-6A <Iris Open>`
		`1-06 <ACK>`
		`0-06 0-91 0-69 <Iris Close>`
		`1-06 <ACK>`

Set Auxiliary 1 off	1 OFF	`0-06 0-E0 0-1A`

Continued on the next page.

Continued from the previous page.

Function	AD2078 Key(s)	Command(s) Generated
	ACK	`1-06`
Set Auxiliary 1 on	1 ON	`0-06 0-E1 0-19`
	ACK	`1-06`
Set Auxiliary 2 off	2 OFF	`0-06 0-E0 0-1A`
	ACK	`1-06`
Set Auxiliary 2 on	2 ON	`0-06 0-E2 0-18`
	ACK	`1-06`
Set Auxiliary 3 off	3 OFF	`0-06 0-E0 0-1A`
	ACK	`1-06`
Set Auxiliary 3 on	3 ON	`0-06 0-E4 0-16`
	ACK	`1-06`
Set Auxiliary 4 off	70 CALL	`0-06 0-E0 0-1A`
	ACK	`1-06`
Set Auxiliary 4 on	71 CALL	`0-06 0-E8 0-12`
	ACK	`1-06`

APPENDIX A

A Available types of Sensormatic domes

Currently on hand Sensormatic/American Dynamics domes consist of:

A.1 Sensormatic Dome #1

This is a working single circuit board type of color camera dome that does not display any configuration information on power up.

Manufacturer	Sensormatic
Model #	SpeedDome 2000 Outdoor
Reg M/N	SU SD
Part of	0100-0760-05
Rev.	A0
Serial #	1507495
Mfd.	4QTR95
Bar Code	*1507495*
Paper Label	RAS586LS
2nd Paper Label	RAS586LS REV. C SC 04/28/99 T.33

Table A-1. Identifying marks on Dome #1

It has three EPROMs:

U4	0701-0081 U4 VER 0.316
U8	0701-0077 U8 VER 03.12
U50	3133-0013-01 U50 VER 1.00

Table A-2. EPROM information from Dome #1

Dome response to SSOFTWAREVERSION (0xC9):

[5] $Header: d:/txb-s422/RCS/dometype.inc,v 1.14 2001-12-12 14:08:06-08 Hamilton Exp Hamilton $

```
0-06 0-C9 0-31
1-06 1-C9 1-06 1-07 1-01 1-00 1-81 1-03 1-16 1-89
```

Dome response to SGETDOMETYPE (`0x94`): `0xF8`

A.2 Sensormatic Dome #2

This is a semi-working (positioning works, camera displays vertical colored bars) double circuit board type of color camera dome. When powered up there is no display of configuration information.

Manufacturer	Sensormatic
Model #	RA486LP
Part #	01RA486LP
Rev.	D0
Serial #	555790
Mfd.	2QTR92
Bar Code	None

Table A-3. Identifying marks on Dome #2

Data from the EPROM label:

```
0701-0006-0620
CHECKSUM 800E
U 10 S/W VER 6.2
```

Reading the EPROM chip with an EPROM reader, the following information was obtained:

```
Copyright Sensormatic Electronics Corporation - 1993
All rights Reserved
SpeedDome Designer Lawrence R. Mills
```

By peeling the label back some we found the following:

Chip Type:	NM27C512Q
Chip Speed:	200 ns
Chip date code:	G9336

Table A-4. EPROM information from Dome #2

Or that the EPROM was probably made the 36th week (or August) of 1993.

Dome response to SSOFTWAREVERSION (0xC9):

```
0-06 0-C9 0-31
1-06 1-C9 1-06 1-07 1-01 1-00 1-06 1-06 1-20 1-F7
```

Dome response to SGETDOMETYPE (0x94): 0xE5

A.3 Sensormatic Dome #3

This is a non-functioning double circuit board type of ??? (color or B&W) camera dome.

Manufacturer	`Sensormatic`
Model #	`RA485LP`
Part of	`0100-0197-01`
Rev #	`M0`
Serial #	`749172`
Mfd	`3QTR93`
Bar Code	`*749172*`

Table A-5. Identifying marks on Dome #3

Data from the EPROM label:

```
0701-0006-0610
CHECKSUM F8F0
U 10  S/W VER 6.1
```

When the chip was read on the EPROM reader it was found to say:

```
Copyright Sensormatic Electronics Corporation - 1993
All rights Reserved
SpeedDome Designer Lawrence R. Mills
```

Peeling the label back a little bit revealed the EPROM's date code was:

Chip Type:	NM27C512Q
Chip Speed:	200 ns
Chip date code:	B9324

Table A-6. EPROM information from Dome #3

Or that the EPROM was probably made the 24th week (or June) of 1993.

A.4 Sensormatic Dome #4

This is a new model American Dynamics Ultra Dome type of color dome. When powered up it displays: "BOOT VER 0103 MAIN VER 0303" and "22X OPTICAL ZOOM" in double high characters. In normal height characters there are the results of passing (or otherwise, ours always passes) the "COMM. LOOPBACK", "CAMERA LOOPBACK" and "MOTOR CIRCUIT" tests.

Manufacturer	American Dynamics
Reg Id	UP SDU
P/N	0100 2283-25
Rev	B1
S/N	4867379
Bar Code	000525-0925
EPROM	0701-2507-0303 U-4 CS=C27A

Table A-7. Identifying marks on Dome #4

Dome response to SSOFTWAREVERSION (0xC9):

```
0-06 0-C9 0-31
1-06 1-C9 1-06 1-07 1-01 1-25 1-07 1-03 1-03 1-F1
```

Dome response to SGETDOMETYPE (0x94): 0xF5

A.5 Sensormatic Dome #5

This is working “new” single circuit board type of B+W camera dome. This dome does not display any configuration information when it is initially powered up.

Manufacturer	Sensormatic
Model #	SpeedDome 2000 Outdoor
Reg M/N	SU SDO
Rev	B0
Part of	0100-761-03
Mfd	4QTR97
Serial #	2380062
Bar Code	*2380062*

Table A-8. Identifying marks on Dome #5

It has three EPROMs:

U4	0701-0081 U4 VER 0316
U8	0701-0077 U8 VER 03.12
U50	3133-0013-01 U50 VER 1.00

Table A-9. EPROM information from Dome #5

Dome response to SSOFTWAREVERSION (0xC9):

```
0-06 0-C9 0-31
1-06 1-C9 1-06 1-07 1-01 1-00 1-81 1-03 1-16 1-89
```

Dome response to SGETDOMETYPE (0x94): 0xF8

A.6 Sensormatic Dome #6

This is a new model American Dynamics Ultra Dome type of color dome. When powered up it displays: “`BOOT VER 0103 MAIN VER 0306`” and “`22X OPTICAL ZOOM`” in double high characters. In normal height characters there are the results of passing (or otherwise, ours always passes) the “`COMM. LOOPBACK`”, “`CAMERA LOOPBACK`” and “`MOTOR CIRCUIT`” tests.

Manufacturer	`American Dynamics`
Reg Id	`UP SDU`
P/N	`0100 2283-21`
Rev	`C0`
S/N	`4466459`
Bar Code	`010718-0593`

EPROM	`0701-2507-0306`
	`U-4`
	`CS=FA82`

Table A-10. Identifying marks on Dome #6

Dome response to SSOFTWAREVERSION (`0xC9`):

```
0-06 0-C9 0-31
1-06 1-C9 1-06 1-07 1-01 1-00 1-81 1-03 1-16 1-8A
```

Dome response to SGETDOMETYPE (`0x94`): `0xF8`

A.7 Sensormatic Dome #SF1 “MiniDome”

This is an older Sensormatic Dome which obtained from Sears Fresno. It is a non-working specimen that has the following written on it.

Serial #	126938
	Anti Theft Alarm 441X

Table A-11. Identifying marks on Dome #SF1

Hand written on the side, with a ball point pen(?), there is a note that says: “Good Camera Slow $\boxed{\texttt{2 RPM}}$.”

A.8 Sensormatic Dome #SF2 “MiniDome”

This non-working older Sensormatic Dome was obtained from Sears Fresno.
“Tested by Sensormatic 8.8.90 Rev C”

Mfg Date	4QTR84
Serial #	136315
Model #	RA410
Paper Label	RA410 Rev E
	RC 4/10/92 T06

Table A-12. Identifying marks on Dome #SF2

A.9 Sensormatic Dome #SF3 “MiniDome”

This non-working older Sensormatic Dome was obtained from Sears Fresno.

Mfg Date	0684
Serial #	128404
Model #	RA511
	Anti Theft Alarm 441X

Table A-13. Identifying marks on Dome #SF3

APPENDIX B

B Differences between various domes and matrix types

B.1 Known dome differences

Note

1. In the below discussions, those values in SANS SERIF font are either the names of various commands or are the marking on the console keys. Values in `teletype font` are actual hexadecimal values or file names.
2. In all the following entries, where Spectra is mentioned, it is safe to assume that an Esprit will work the same way. When there are differences in operation, they will be clearly indicated.
3. Included dome types in this list:

   There are at least four different types of Sensormatic domes. Three by Sensormatic and one or two (Spectra/Esprit) by Pelco. Either a Spectra/Esprit should “look the same”[7] Sensormatic controller when a TXB-S422 is installed. If the Sensormatic system has an American Dynamics model AD2091 Manchester Code translator installed, a Pelco TXB-AB/TXB-AD may be installed for control. This write-up does not discuss Pelco domes with a TXB-AB or TXB-AD installed in them.
4. Alternate dome IDs:

   A. **SpeedDome, two board model:** Dome #2.

   B. **SpeedDome 2000, UniCard model:** Domes #1 or/and #5 (5 is the most commonly used dome as it was bought new for this project.)

   C. **DeltaDome/UltraDome:** Dome #4.

   D. **Spectra with a TXB-S422 installed:** Any old Spectra.

   E. Dome #3, a two board SpeedDome has never worked.

   F. Dome #6, a DeltaDome/UltraDome has not been used due to availability problems. (And of course it hadn't been acquired until the project was more than half done.)

This is a short list of the known differences between the various types of domes.

1. **Command Response Timing:**

   This information was obtained by using a storage oscilloscope with its storage set to “infinite” and a STOPALL command sent to it for about 30 seconds. The data in parentheses following each item below refers to the oscilloscope picture that shows this information.

[6] `$Header: d:/txb-s422/RCS/diffs.inc,v 1.8 2002-02-06 15:57:23-08 Hamilton Exp Hamilton $`

[7] However the Esprit may not have the same pan and tilt speeds and in future models of the Spectra/Esprit the speeds may change.

A. **SpeedDome, two board model:** This dome sends an ACKnowledgement to the controller starting as soon as 3.2 ms following receiving a command. The latest that it sends an ACK is 20.7 ms after receiving a command. (`S34.PS` and `S35.PS`)

B. **SpeedDome 2000, UniCard model:** This dome sends an ACKnowledgement to the controller starting as soon as 2.7 ms following receiving a command. The latest that it sends an ACK is 15.7 ms after receiving a command. (`S36.PS` and `S37.PS`)

C. **DeltaDome/UltraDome:** This dome sends an ACKnowledgement to the controller starting as soon as 5.3 ms following receiving a command. The latest that it sends an ACK is 11.5 ms after receiving a command. (`S38.PS` and `S39.PS`)

D. **Spectra with a TXB-S422 installed:** This dome sends an ACKnowledgement to the controller starting at 33.2 ms following receiving a command. The latest that it sends an ACK is 38.5 ms after receiving a command. (`S40.PS` and `S41.PS`)

The TXB-S422 "bit bangs" data out to the Spectra, while doing this it can not do anything else and that is what causes this long delay. A Sensormatic AUX # command generates four individual commands from the TXB-S422 to "its" Spectra. Thus the minimum time to send an ACK to the controller is slightly over 120 ms.

2. **Responses to increased speed commands in fixed speed systems:**

A. **Response to a FASTER (or RAIL RIGHT, `0x8E`) command:**

In Appendix G of the "Training Manual for SensorVision Programmable Video Management Systems" it indicates that this key should change the default speed of 28^o/sec by two times.

a. **SpeedDome, two board model:** If the dome is moving when it receives a FASTER command it increases the dome's speed to about 56^o/sec. If the dome is not moving then nothing happens.

b. **SpeedDome 2000, UniCard model:** If the dome is moving when it receives a FASTER command it increases the dome's speed to about 56^o/sec. If the dome is not moving then nothing happens.

c. **DeltaDome/UltraDome:** If the dome is moving when it receives a FASTER command it moves down in tilt until it gets to the maximum down position and then does a FLIP followed by moving to maximum tilt up, when it stops motion. If the dome is not moving then the dome does a FLIP.

d. **Spectra with a TXB-S422 installed:** If the dome is moving when it receives a FASTER command it increases the dome's speed to about 56^o/sec. If the dome is not moving then the dome does a FLIP.

B. **Response to a FAST (or FAST, `0x9A`) command:**

In Appendix G of the "Training Manual for SensorVision Programmable Video Management Systems" it indicates that this key should change the default speed of 28^o/sec by three times.

a. **SpeedDome, two board model:** If the dome is moving when it receives a FAST command it increases the dome's speed to about 83^o/sec. If the dome is not moving then nothing happens.

b. **SpeedDome 2000, UniCard model:** If the dome is moving when it receives a FAST command it increases the dome's speed to about 83^o/sec. If the dome is not moving then nothing happens.

c. **DeltaDome/UltraDome:** If the dome is moving when it receives a FAST command it moves down in tilt until it gets to the maximum down position and then does a FLIP followed by moving to maximum tilt up, when it stops motion. If the dome is not moving then nothing happens.

d. **Spectra with a TXB-S422 installed:** If the dome is moving when it receives a FAST command it increases the dome's speed to about 83^o/sec. If the dome is not moving then nothing happens.

C. **Response to a FASTEST (or RAIL LEFT, `0x8E`) command:**

In Appendix G of the "Training Manual for SensorVision Programmable Video Management Systems" it indicates that this key should change the default speed of 28^o/sec by four times.

a. **SpeedDome, two board model:** If the dome is moving when it receives a FASTEST command it increases the dome's speed to about 111^o/sec. If the dome is not moving then nothing happens.

b. **SpeedDome 2000, UniCard model:** If the dome is moving when it receives a FASTEST command it increases the dome's speed to about 111^o/sec. If the dome is not moving then nothing happens.

c. **DeltaDome/UltraDome:** If the dome is moving when it receives a FASTEST command it moves down in tilt until it gets to the maximum down position and then does a FLIP followed by moving to maximum tilt up, when it stops motion. If the dome is not moving then nothing happens.

d. **Spectra with a TXB-S422 installed:** If the dome is moving when it receives a FASTEST command it increases the dome's speed to about 144^o/sec. If the dome is not moving then nothing happens.

3. **Motion timeouts:**

A. **SpeedDome, two board model:** None.

B. **SpeedDome 2000, UniCard model:** None.

C. **DeltaDome/UltraDome:** 30 seconds.

D. **Spectra with a TXB-S422 installed:** 15 seconds.

4. **Flip:**

A. **SpeedDome, two board model:** Yes.

B. **SpeedDome 2000, UniCard model:** Yes.

C. **DeltaDome/UltraDome:** Yes.

D. **Spectra with a TXB-S422 installed:** Yes.

5. **Auto flip:**

A. **SpeedDome, two board model:** No.

B. **SpeedDome 2000, UniCard model:** No.

C. **DeltaDome/UltraDome:** Yes.

D. **Spectra with a TXB-S422 installed:** Yes.

6. **Auto focus:**

A. **SpeedDome, two board model:** No.

B. **SpeedDome 2000, UniCard model:** No.

C. **DeltaDome/UltraDome:** Yes.

D. **Spectra with a TXB-S422 installed:** Yes.

7. Dome response to a POLL (or REQUEST DOME TYPE, `0x94`) command:

A. **SpeedDome, two board model:** This type of dome sends a response of `0xE5` following initial power up and then changes to `0xF5` after a while for no apparent reason. (The change appears to occur after about five POLLs have been received. It is not "time" based as the dome may be turned on and left alone for ten minuets and it will not change.)

B. **SpeedDome 2000, UniCard model:** This type of dome sends a response of `0xE8` following initial power up and then changes to `0xF8` after a while for no apparent reason. (The change appears to occur after about five POLLs have been received. It is not "time" based as the dome may be turned on and left alone for ten minuets and it will not change.)

C. **DeltaDome/UltraDome:** This type of dome sends a response of `0xF5` following initial power up and does not appear to ever change.

D. **Spectra with a TXB-S422 installed:** This type of dome sends a response of `0xF8` or `0xF5` following initial power up and never changes. The exact response type is controlled by switch position 4. (On = `0xF5`, off (default) = `0xF8`.)

8. Actions related to presets (or QUICKVIEWS):

A. **Response to a PRESET SET (or REQUEST DOME POSITION `0xA5`) command:**

a. **SpeedDome, two board model:** This dome sends a 12 byte message containing various information as to where it is pointing and the current camera settings.

b. **SpeedDome 2000, UniCard model:** This dome sends a 12 byte message containing various information as to where it is pointing and the current camera settings.

c. **DeltaDome/UltraDome:** This dome sends a 12 byte message containing various information as to where it is pointing and the current camera settings.

d. **Spectra with a TXB-S422 installed:** This dome sends a 12 byte message copied from dome #5 (item B, above) with the current preset # number in some of the fields. Note that this makes it impossible to actually tell the controller where the dome is pointing. This is because there is no way in the current version of D Protocol for a dome to convey its direction and camera parameters to an outside source.

B. **Response of the dome to a** PRESET CALL **(or** GOTO ABSOLUTE POSITION `0xA6`**) command:**

a. **SpeedDome, two board model:** Unknown.

b. **SpeedDome 2000, UniCard model:** Unknown.

c. **DeltaDome/UltraDome:** The dome takes in the supplied pointing and camera parameter information and uses this information to control where/how the unit will point.

d. **Spectra with a TXB-S422 installed:** The dome takes the supplied pointing and camera parameter information. It uses the preset ID that was placed into the PRESET SET command's response as a preset #.

C. **Dome response to a** MARK TARGET # **(**`0xA8`**,** `0xA9`**,** `0xAA`**,** `0xAB`**,** `0xB9`**,** `0xBA` **and** `0xBC`**) commands:**

a. **SpeedDome, two board model:** The current location information is saved in the dome. It appears that it is saved in semi-long term RAM, i.e. the information stays in memory during a short power down cycle, but isn't there is the unit is powered down overnight.

b. **SpeedDome 2000, UniCard model:** The current location information is saved in the dome. It appears that it is saved in semi-long term RAM, i.e. the information stays in memory during a short power down cycle, but isn't there is the unit is powered down overnight.

c. **DeltaDome/UltraDome:** The current pointing information and camera information is saved in non-volatile memory. The data is still there after the dome is turned off overnight.

d. **Spectra with a TXB-S422 installed:** Sets the indicated preset#, either 1, 2, 3, 4, 5, 6 or 7, in non-volatile memory.

D. **Dome response to a** GOTO TARGET # **(**`0xB4`**,** `0xB5`**,** `0xB6`**,** `0xB7`**,** `0xBC`**,** `0xBD` **and** `0xBE`**) commands:**

a. **SpeedDome, two board model:** If the data has been loaded since the last power cycle, then a normal preset move is executed. See above for length of data storage.

b. **SpeedDome 2000, UniCard model:** If the data has been loaded since the last power cycle, then a normal preset move is executed. See above for length of data storage.

c. **DeltaDome/UltraDome:** Goes to the indicated preset # with data from the non-volatile memory.

d. **Spectra with a TXB-S422 installed:** Goes to the indicated preset#, either 1, 2, 3, 4, 5, 6 or 7, with data from the non-volatile memory.

9. **Response to a MEMORY DUMP (`0xA4`) command:**

   A. **SpeedDome, two board model:** 104 bytes of information. On initial power up the message is shorter and gets longer when patterns are loaded in.

   B. **SpeedDome 2000, UniCard model:** 104 bytes of information. In some cases this gets up to 6,415 byte messages. On initial power up the message is shorter and gets longer when presets and patterns are loaded in.

   C. **DeltaDome/UltraDome:** 104 bytes of information. On initial power up the message is shorter and gets longer when patterns are loaded in.

   D. **Spectra with a TXB-S422 installed:** 104 bytes of information. This was copied from a memory dump captured from dome #5 (item B, above) when it had no motion information loaded into it. The size, and content, never change.

10. **Response to a REQUEST SOFTWARE VERSION (`0xC9`) command:**

    A. **SpeedDome, two board model:** This dome sends a 10 byte response that holds its current software version ID.

    B. **SpeedDome 2000, UniCard model:** This dome sends a 10 byte response that holds its current software version ID.

    C. **DeltaDome/UltraDome:** This dome sends a 10 byte response that holds its current software version ID.

    D. **Spectra with a TXB-S422 installed:** This dome sends a 10 byte response that was copied from the one generated by dome #5 (item B, above).

11. DEFINE BOUNDARY (`0x9C`):

    A. **SpeedDome, two board model:** This dome replys with an ACK followed by a `0xC4` message.

    B. **SpeedDome 2000, UniCard model:** This dome replys with an ACK followed by a `0xC4` message.

    C. **DeltaDome/UltraDome:** This dome replys with an ACK followed by a `0xC4` message.

    D. **Spectra with a TXB-S422 installed:** This dome replys with an ACK followed by a `0xC4` message.

12. MARK BOUNDARY (`0x9D`):

A. **SpeedDome, two board model:** This dome will send a BOUNDARY CONFUSION (0xB4) response if it is not in DEFINE BOUNDARY mode. If it is in DEFINE BOUNDARY mode then it sends an ACK, and a BOUNDARY CROSSING # (0xB0) message.

B. **SpeedDome 2000, UniCard model:** This dome will send a BOUNDARY CONFUSION (0xB4) response if it is not in DEFINE BOUNDARY mode. If it is in DEFINE BOUNDARY mode then it sends an ACK, and a BOUNDARY CROSSING # (0xB0) message.

C. **DeltaDome/UltraDome:** This dome will send a BOUNDARY CONFUSION (0xB4) response if it is not in DEFINE BOUNDARY mode. If it is in DEFINE BOUNDARY mode then it sends an ACK, and a BOUNDARY CROSSING # (0xB0) message.

D. **Spectra with a TXB-S422 installed:** This dome always sends a BOUNDARY CROSSING 0 (0xB0) message.

13. ALARM STATUS **(0x95):**

A. **SpeedDome, two board model:** 0x00.

B. **SpeedDome 2000, UniCard model:** 0x00.

C. **DeltaDome/UltraDome:** 0x0F.

D. **Spectra with a TXB-S422 installed:** 0x00.

14. **Is the first command received, processed?**

A. **SpeedDome, two board model:** Yes, no commands are lost.

B. **SpeedDome 2000, UniCard model:** Yes, no commands are lost.

C. **DeltaDome/UltraDome:** The first command is always lost. This is done so that the dome may recognize the "network type". The results of detecting the first command are used in setting the network type. The first command is normally a POLL command and thus "losing" it isn't noticed.

D. **Spectra with a TXB-S422 installed:** Yes, no commands are lost.

B.2 Known matrix differences

Note

1. In the below discussions, those values in SANS SERIF font are either the names of various commands or are the marking on the console keys. Values in `teletype font` are actual hexadecimal values or file names.
2. In all the following entries, where Spectra is mentioned, it is safe to assume that an Esprit will work the same way. When there are differences in operation, they will be clearly indicated.
3. Included matrix types, or matrix equivalents, in this list:

   There are at least four different types of Sensormatic/American Dynamics matrices.

   Two or four by Sensormatic and one by American Dynamics.

4. **Matrix types:**
   - A. **VM1:** this an older type of matrix and is probably a group name that covers an RC20, RC32 and an RC58. Sears Fresno has a working RC58 installed and most of the information shown here came from examining it. (This unit may not be in current production.)
   - B. **AD2083/02A/AD1995:** this is an medium older unit that was made by American Dynamics. It is controlled by an AD2078 keyboard. The AD1996 CPU is used to drive the AD2083/02A code translator. The AD1996 CPU is supposed to be similar to the AD2050 matrix that we used to have[8].
   - C. **VM96:** this is a newer type of matrix and probably consists of an RC216. Sears Clovis has a working matrix of this type and Pelco has a partial on of these. Most of the information about the operation of a RC216 came from examining the one at Sears Clovis.
   - D. **TouchTracker:** marketing has a Sensormatic TouchTracker that works similarly to a KBD-300. However it appears to communicate with a SpeedDome/-UltraDome using other than the RS422 protocol that is used by the TXB-S422.

This is a short list of the known differences between the various types of matrices.

1. **Maximum # cameras supported:**
   - A. **VM1:** 20 (RC20), 32 (RC32) or 58 (RC58).
   - B. **AD2083/02A:** Unknown, it might any large number in steps of 98 cameras. The POLLing sequence from the AD2083/02A starts at 1 and goes to 99_{10}, but it will not communicate with unit 64_{10}. Address 64_{10} appears to be a “broadcast” channel and no camera may use this as a control address.

     Some of the experments to determine what the address range of the AD2083/02A were:
     - a. An attempt was made to “encourage” the AD2083/02A to work on address 64_{10} by setting the dome to address 64_{10}, waiting for POLL cycle to pass through 64_{10}. The AD2083/02A did ACK the POLL, but still refused to control the dome.
     - b. The dome was powered (dome #5) down and back up while at address 64_{10}. The AD2083/02A did properly process the power up sequence, but still there was no control of the dome available.
     - c. The same dome’s address was changed to 100_{10} and power cycled. The AD2083/02A did not process the power up sequence.
     - d. Last the same dome was placed at address 0_{10} and power cycled. The AD2083/02A did correctly process the power up sequence but the AD2078 keyboard would not accept a camera number of 0_{10}.

[8] Pelco east wanted the AD2050 back so that they could “look at its construction” when building a new Pelco matrix.

C. **VM96:** 95. (The POLLing sequence from the VM96 starts at 1 and goes to 96_{10}, but it might not communicate with unit 64_{10}. Address 64_{10} appears to be a "broadcast" channel and no camera may use this as a control address.

2. POLL **rate:**

   A. **VM1:** Once per second.

   B. **AD2083/02A:** Once per second.

   C. **VM96:** Twice per second.

3. **Dome speeds available:**

   A. **VM1:** Four. There are the FingerStick "default", RAIL RIGHT 2X, FAST 3X and RAIL LEFT 4X, speeds available.

   B. **AD2083/02A:** Eight. These are under the control of a joy stick on the AD2078 keyboard.

   C. **VM96:** Unclear. At low pan and tilt speeds the speed command changes in steps of 1^o/sec. At higher speeds it changes in steps of 4 and then 8^o/sec. There may be as many as 50 or 60 different speeds available. (Determining the exact number is difficult as we do not have a TouchTracker to use with our RC216 and so an exact count is not possible to get. This information came from the system at Sears Clovis.)

4. **When will the matrix communicate with a dome?**

   This is important as the AD2083/02A may indicate that the dome isn't "there", but a Sensormatic matrix will control the dome immediately. (Note, to get presets/patterns to work the dome **must** have working **two-way communications** with the matrix, **and this does not necessarily happen if the dome is only being run in pan/tilt.**)

   A. **VM1:** Any time that a dome is addressed.

   B. **AD2083/02A:** Only after the matrix has "seen" either an answer to a POLL (which may take up to a minute and a half to be requested/received) or a dome initiated power up sequence.

   C. **VM96:** Any time that a dome is addressed.

5. **When turned on, how long is it until the matrix is working?**

   A. **VM1:** Unknown, the only unit I have access to (and that is at Sears Fresno) is not too reliable and I haven't attempted to turn it off and then back on.

   B. **AD2083/02A:** One to two seconds.

   C. **VM96:** Four minuets. This is a Lynx (Linux) based matrix and it takes awhile to get it working.

6. **Other pecularities:**

   A. **VM1:** None that are known of.

   B. **AD2083/02A:** None that are known of.

   C. **VM96:** If a TouchTracker is not installed, requires that a PC type keyboard be installed, or it will not boot up. It is unclear if it will boot up if a TouchTracker is installed[9].

[9] We don't have one to allow testing of this.

APPENDIX C

C VM96 controller data

Pelco has two VM96 controllers. There are some differences between them and these differences are summarized here.

It appears that Sensormatic combines different pieces of equipment into a full system. In reality Pelco has two RC216s. However as a RC216 is the major part of a VM96 system, I have chosen to start referring to these two systems as "VM96"s. Older references were always to the RC216 only.

C.1 VM96 #1

Lynx preboot version	`5.1`
LynxOS 386/AT Version	`2.2.1`
LynxOS creation date	`Wed Apr 12 09:02:18 1995`
File system creation date	`Wed Oct 9 21:44:55 1996`
VM96 software rev	`0701-0051-0300`

C.2 VM96 #2

Lynx preboot version	`5.1`
LynxOS 386/AT Version	`2.2.1`
LynxOS creation date	`Wed Apr 12 09:02:18 1995`
File system creation date	`Wed Jul 9 14:13:20 1997`
VM96 software rev	`0701-0051-0310`

[10] `$Header: d:/txb-s422/RCS/vm96data.inc,v 1.2 2002-06-12 15:50:25-07 Hamilton Exp Hamilton $`

APPENDIX D

D Dome responses to commands

This data was collected with domes $1, #2 and #5 using `seqtest.bas` in sequential mode. Two, or more, complete runs were done when the each dome had been turned off overnight and was freshly turned on. The original data is available as:

Dome	Files
#1	06DEC01C.CAP 06DEC01C.DOC
#2	06DEC01D.CAP 06DEC01D.DOC 07DEC01A.CAP 07DEC01A.DOC 07DEC01B.CAP 07DEC01B.DOC
#4	11DEC01B.CAP 11DEC01B.DOC
#5	06DEC01A.CAP 06DEC01A.DOC 10DEC01B.CAP 10DEC01B.DOC

Notes

- `0xA6` (Go To Absolute Position) didn't get a response as it is "long" (multibyte) command and the commands following it were interperted as data. Thus commands `0xA7` (an unimplemented command, which should not get a response) and `0xA8` (Mark Target 1), unexpectedly generated no response.
- The phrase "ACK" indicates that the dome responded with the standard acknowledment of just its ID. While the phrase "1, 2, 5 = ACK" indicates that domes #1, #2 and #5 generated a standard acknowledgement.
- Dome #3 is inoperable and has never responded to any command.
- Dome #6 "hung up" on high number commands (i.e. those in the range of `0xF0` ⟶ `0xFF`) and results of its testing have been ignored.

Cmnd	Use	Dome Response
1, 2, 4, 5 = For commands `0x00` ⟶ `0x7F` no response of any kind was observed, neither is any of these defined.		
`0x80`	Undefined	1, 2, 4, 5 = ACK `0xC4` UNKNOWN RESPONSE, three times

Continued on the next page.

[11] `$Header: d:/txb-s422/RCS/dr.inc,v 1.6 2001-12-17 16:27:48-08 Hamilton Exp Hamilton $`

Continued from the previous page.

Cmnd	Use	Dome Response
0x81	PAN LEFT	1, 2, 4, 5 = ACK
0x82	PAN RIGHT	1, 2, 4, 5 = ACK
0x83	PAN STOP	1, 2, 4, 5 = ACK
0x84	TILT UP	1, 2, 4, 5 = ACK
0x85	TILT DOWN	1, 2, 4, 5 = ACK
0x86	TILT STOP	1, 2, 4, 5 = ACK
0x87	FOCUS NEAR	1, 2, 4, 5 = ACK
0x88	FOCUS FAR	1, 2, 4, 5 = ACK
0x89	FOCUS STOP	1, 2, 4, 5 = ACK
0x8A	ZOOM IN	1, 2, 4, 5 = ACK
0x8B	ZOOM OUT	1, 2, 4, 5 = ACK
0x8C	ZOOM STOP	1, 2, 4, 5 = ACK
0x8D	FAST (RAIL LEFT)	1, 2, 4, 5 = ACK
0x8E	FASTEST (RAIL RIGHT)	1, 2, 4, 5 = ACK
0x8F	FAST STOP	1, 2, 4, 5 = ACK
0x90	IRIS OPEN	1, 2, 4, 5 = ACK
0x91	IRIS CLOSE	1, 2, 4, 5 = ACK
0x92	IRIS STOP	1, 2, 4, 5 = ACK
0x93	STOP ALL	1, 2, 4, 5 = ACK
0x94	POLL, DOME TYPE QUERY	1, 2, 4, 5 = DOME QUERY RESPONSE, 1 = 0xE8; 2 = 0xE5, 0xF5; 4, 5 = 0xF8
0x95	REQUEST ALARM STATUS	1, 2, 4, 5 = 0x00 DOME ALARM STATUS
0x96	Undefined	1, 2, 4, 5 = 0xC0 Undefined message
0x97	ACK TO DOME	1, 2, 4, 5 = None, none expected
0x98	START TEMP NO XMIT	1, 2, 4, 5 = None, none expected
0x99	END TEMP NO XMIT	1, 2, 4, 5 = None, none expected
0x9A	FASTER	1, 2, 4, 5 = ACK
0x9B	FASTER STOP	1, 2, 4, 5 = ACK
0x9C	DEFINE BOUNDARY	1, 2, 4, 5 = ACK, 0xC4 UNKNOWN RESPONSE, three times
0x9D	MARK BOUNDARY	1, 2, 4, 5 = ACK, 0xB0 BOUNDRY CROSSING 1, three times; sometimes 2 = 0xB4 BOUNDARY CONFUSION.

Continued on the next page.

Continued from the previous page.

Cmnd	Use	Dome Response
`0x9E`	ON AIR	1, 2, 4, 5 = ACK
`0x9F`	RESET ON AIR	1, 2, 4, 5 = ACK
`0xA0`	DEFINE PATTERN 1	1, 2, 4, 5 = ACK
`0xA1`	DEFINE PATTERN 2	1, 2, 4, 5 = ACK
`0xA2`	DEFINE PATTERN 3	1, 2, 4, 5 = ACK
`0xA3`	ACCEPT NEW PATTERN	1, 2, 4, 5 = ACK
`0xA4`	DUMP DOME MEMORY	1, 2, 4, 5 = A whole bunch of data
`0xA5`	REQUEST DOME POSITION	1, 2, 4, 5 = ACK and 12 bytes of position data, three times. Position data only once for #2.
`0xA6`	GO TO ABSOLUTE POSITION	No response, see notes
`0xA7`	Undefined	None
`0xA8`	MARK TARGET 1	No response, see notes
`0xA9`	MARK TARGET 2	1, 5 = ACK. Sometimes 2 = ACK and 12 bytes of position data, once.
`0xAA`	MARK TARGET 3	1, 5 = ACK.
`0xAB`	MARK TARGET 4	1, 5 = ACK
`0xAC`	GO TO PATTERN 1 START	1, 2, 4, 5 = ACK, 2 = No ACK if no pattern is defined.
`0xAD`	GO TO PATTERN 2 START	1, 2, 4, 5 = ACK, 2 = No ACK if no pattern is defined.
`0xAE`	GO TO PATTERN 3 START	1, 2, 4, 5 = ACK, 2 = No ACK if no pattern is defined.
`0xAF`	GO TO PATTERN 4 START	1, 2, 4, 5 = ACK, 2 = No ACK if no pattern is defined.
`0xB0`	RUN PATTERN 1	1, 2, 4, 5 = ACK, 2 = No ACK if no pattern is defined.
`0xB1`	RUN PATTERN 2	1, 2, 4, 5 = ACK, 2 = No ACK if no pattern is defined.
`0xB2`	RUN PATTERN 3	1, 2, 4, 5 = ACK, 2 = No ACK if no pattern is defined.
`0xB3`	RUN NEW PATTERN	1, 2, 4, 5 = ACK
`0xB4`	GO TO TARGET 1	1, 2, 4, 5 = ACK, 2 = No ACK if no target is defined.
`0xB5`	GO TO TARGET 2	1, 2, 4, 5 = ACK, 2 = No ACK if no target is defined.

Continued on the next page.

Continued from the previous page.

Cmnd	Use	Dome Response
`0xB6`	GO TO TARGET 3	1, 2, 4, 5 = ACK, 2 = No ACK if no target is defined.
`0xB7`	GO TO TARGET 4	1, 2, 4, 5 = ACK, 2 = No ACK if no target is defined.
`0xB8`	PATTERN END	1, 2, 4, 5 = ACK
`0xB9`	MARK TARGET 5	1, 2, 4, 5 = ACK
`0xBA`	MARK TARGET 6	1, 2, 4, 5 = ACK
`0xBB`	MARK TARGET 7	1, 2, 4, 5 = ACK
`0xBC`	GO TO TARGET 5	1, 2, 4, 5 = ACK
`0xBD`	GO TO TARGET 6	1, 2, 4, 5 = ACK
`0xBE`	GO TO TARGET 7	1, 2, 4, 5 = ACK
`0xBF`	Undefined	1, 2, 4, 5 = ACK
`0xC0`	Undefined	1, 2, 4, 5 = None
`0xC1`	Undefined	1, 2, 4, 5 = None
`0xC2`	Undefined	1, 2, 4, 5 = None
`0xC3`	Undefined	1, 5 = ACK
`0xC4`	Undefined	1, 5 = None. 2 = Unknown 25 byte response sent once.
`0xC5`	Undefined	1, 5 = None. 2 = ACK
`0xC6`	Undefined	1, 5 = ACK
`0xC7`	Undefined	1, 5 = ACK, 2 = `0xC1` DOME POWER UP MESSAGE, three times.
`0xC8`	Undefined	1, 2, 4, 5 = ACK
`0xC9`	GET SOFTWARE ID	1, 2, 4, 5 = 10 bytes of software ID data
1, 2, 4, 5 = For commands `0xCA` ⟶ `0xDC` an ACK was received. However none of these is defined.		
`0xDD`	Undefined	1, 5 = ACK. 2 = `0xC1` DOME POWER UP MESSAGE
`0xDE`	Undefined	1, 2, 4, 5 = ACK
`0xDF`	Undefined	1, 2, 4, 5 = ACK
`0xE0`	AUX 1,2,3,4 OFF	1, 2, 4, 5 = ACK
`0xE1`	AUX 1 ON, 2,3,4 OFF	1, 2, 4, 5 = ACK
`0xE2`	AUX 2 ON, 1,3,4 OFF	1, 2, 4, 5 = ACK
`0xE3`	AUX 1,2 ON, 3,4 OFF	1, 2, 4, 5 = ACK

Continued on the next page.

Continued from the previous page.

Cmnd	Use	Dome Response
`0xE4`	AUX 3 ON, 1,2,4 OFF	1, 2, 4, 5 = ACK
`0xE5`	AUX 1,3, ON, 2,4 OFF	1, 2, 4, 5 = ACK
`0xE6`	AUX 2,3, ON, 1,4 OFF	1, 2, 4, 5 = ACK
`0xE7`	AUX 1,2,3 ON, 4 OFF	1, 2, 4, 5 = ACK
`0xE8`	AUX 4 ON, 1,2,3, OFF	1, 2, 4, 5 = ACK
`0xE9`	AUX 1,4 ON, 2,3 OFF	1, 2, 4, 5 = ACK
`0xEA`	AUX 2,4 ON, 1,3 OFF	1, 2, 4, 5 = ACK
`0xEB`	AUX 1,2,4 ON, 3 OFF	1, 2, 4, 5 = ACK
`0xEC`	AUX 3,4 ON, 1,2 OFF	1, 2, 4, 5 = ACK
`0xED`	AUX 1,3,4 ON, 2 OFF	1, 2, 4, 5 = ACK
`0xEE`	AUX 2,3,4 ON, 1 OFF	1, 2, 4, 5 = ACK
`0xEF`	AUX 1,2,3,4 ON	1, 2, 4, 5 = ACK
`0xF0`	TERMINATE PATTERN	1, 2, 4, 5 = ACK, `0xB5` PATTERN DONE, three times
1, 2, 4, 5 = For commands `0xF1` ⟶ `0xFF` an ACK was received. However none of these is defined.		

APPENDIX E

E Speeds of an RC216 controller

Here is a listing of the detected variable speed values from an RC216 controller. In this listing different speeds were detected for pan and tilt. In some cases there are some missing speed values, these were probably caused when the operator “skiped over” them.

Left	Right	Up	Down	^{o}/sec
0x81	0x82	0x84	0x85	---
0x01	0x01	0x01	0x01	1
0x02	0x02	0x02	0x02	2
0x03	0x03	0x03	0x03	3
0x04	0x04	0x04	0x04	4
0x05	0x05	PT	PT	5
0x06	0x06	0x06	PT	6
0x07	0x07	0x07	PT	7
0x08	0x08	0x08	0x08	8
0x09	0x09	0x09	0x09	9
0x0A	0x0A	0x0A	0x0A	10
0x0B	0x0B	0x0B	0x0B	11
0x0C	0x0C	0x0C	0x0C	12
0x0D	0x0D	PT	PT	13
0x0E	0x0E	0x0E	PT	14
0x0F	0x0F	0x0F	0x0F	15
0x10	0x10	0x10	0x10	16
0x11	0x11	0x11	0x11	17
0x12	0x12	0x12	PT	18
0x13	PT	PT	PT	19
0x14	0x14	0x14	0x14	20
PM	PM	PT	0x15	21
0x16	0x16	0x16	0x16	22
PM	PM	0x17	PT	23
0x18	0x18	0x18	0x18	24
PM	PM	PT	PT	25
PT	PT	0x1A	PT	26
PM	PM	0x1B	PT	27
0x1C	0x1C	0x1C	0x1C	28

Continued on the next page.

[12] $Header: d:/txb-s422/RCS/speeds.inc,v 1.5 2002-05-10 10:59:10-07 Hamilton Exp Hamilton $

[13] $Header: d:/txb-s422/RCS/rc216spd.inc,v 1.2 2002-05-10 11:29:56-07 Hamilton Exp Hamilton $

Continued from the previous page.

Left	Right	Up	Down	^{o}/sec
0x81	0x82	0x84	0x85	---
PM	PM	0x1E	PT	30
0x20	0x20	0x20	0x20	32
PM	PM	0x22	PT	34
0x24	0x24	0x24	0x24	36
0x28	0x28	0x28	0x28	40
PM	PM	0x2C	0x2C	44
0x30	0x30	0x30	0x30	48
PM	PM	0x32	0x32	50
PM	PM	0x34	0x34	52
0x38	0x38	0x38	0x38	56
0x40	0x40	PM	PM	64
0x48	0x48	PM	PM	72
0x50	0x50	PM	PM	80
0x58	0x58	PM	PM	88
0x63	0x63	PM	PM	99

Note

- PM is used to indicate that this speed value is Probably Missing, as it has never been detected as comming from an RC216.
- PT is used to indicate that this speed value is Probably There, even though it was not detected in this sample data capture.
- In the higher pan speeds that these speeds appear to increase in steps of $8_{10}{}^o$/sec (0x08) with a limit of $99_{10}{}^o$/sec (0x63).
- In the higher tilt speeds that these speeds appear to increase in steps of $2_{10}{}^o$/sec or $4_{10}{}^o$/sec with a limit of 56^o/sec 0x38).
- At lower speeds the increments are quite small, i.e. steps of 1^o/sec and 2^o/sec are common.
- Data is based on speed values found in the 14JAN02A.CAP file, which was collected at Sears Clovis on their RC216 (VM96) system.

APPENDIX F

F How to process “breakout” (.CAP) files

30MAR01 Revised 14JUN01, 30JUL01

How to process Breakout’s capture (.CAP) files of Sensormatic protocol data for use.

1. Some how or other collect the data.

2. Run it through DB. (DB stands for Dump Breakout capture files. Or is Dumb Bunny?)

DB’s options are:

/2 To cause DB to put out two CR-LFs following a data direction change. /A To cause DB to put out ASCII equivalents of the hex data. /R To cause DB to reverse the logic for direction identifiers.

3. Then run it through CRIT. (CRIT stands for put CRs in IT.) This puts most of the new line delimiters into the .DB file.

4. Now run it through NBL. (NBL stands for No Blank Lines.) If the blank lines are not eliminated, the file will have a lot of them.

5. And then run it through MARK1, which will put a blank line in front of polls to ID 01. Be sure to name the output file .DOC.

6. Now for the fun part, get the editor going and "clean up" all problems in the data.

7. When the data has been cleaned up run ID1. (ID1 stands for IDentify, version 1.) This will stick text descriptions after each identified command.

8. An automated method of doing steps 2 thorough 5 is in YY.BAT. (If it has been determined that the data directions are not reversed, run YYR.BAT.)

9. At this point you should have a file of several thousand lines of data that you can analyze further.

[14] $Header: d:/txb-s422/RCS/00howto.txt,v 1.2 2001-07-31 10:22:12-07 Hamilton Exp Hamilton $

APPENDIX G

Index