Burle and AD Command Details

28 December 2000

TABLE OF CONTENTS

Section **Page**

[1] $Header: d:/ecr6171/RCS/commands.tex,v 1.12 2000-11-14 09:25:11-08 Hamilton Exp Hamilton $

LIST OF FIGURES

1 Burle command details

1.1 Burle Commands

There are two basic formats of Burle commands. For convenience I am calling these: Format K and Format M.

Format K commands are those used primarily with Burle keyboards. Such as their "Autodome Controller", type number LTC-5136/50.

Format M commands are those used primarily with Burle matrices. Such as an LTC-8801/60 matrix that I have in my cube. (I also have a TC-8501 matrix that sends similar commands to the LTC-8801/60.)

A full list of the Burle equipment that was used for testing is listed in Section C, page C-1. American Dynamics equipment is listed in Section D, page D-1.

There were two documents that were used in understanding the content of Burle commands. One of these is titled "Receiver/Driver and AutoDome® Control Code Protocol" by Philips and dated 1997. The other was an Excel spread sheet of information about the commands. By far the most useful was the Philips document.

1.1.1 Common characteristics of the two Burle command types

Although each of the different sources of Burle commands sends out commands that differ in several respects, the following apply to all versions so far examined: (See Figure 1, page 9 for an example screen shot with the decodes shown for first full byte of a Format K command from an LTC-5136 keyboard. See Figure 2, page 10 for an example screen shot with the decodes shown for the start of a Format M command from a TC-8501 matrix and Figure 3, page 11 for a Format M command from an LTC-8801 matrix.)

1. Bit rate is about 31.25 KHz with a baud rate of about 62.5 KHz.
2. Minimum transition time between any high and low is 16 μs.
3. Time duration for any data bit 32 μs. (Except for the sync and anti-sync pulses, which are longer.)
4. Data is in BI-ϕ-L format.
5. Data is sent as "three state" data. Note this is not the same as "tri-state" data format as used with digital logic. The three states of data line are:

   A. Quiescent, i.e. no data.
   B. High.
   C. Low.

[2] `$Header: d:/ecr6171/RCS/commands.inc,v 1.23 2000-11-27 14:22:48-08 Hamilton Exp Hamilton $`

It is important to note that when the data line is in either high or low the voltage value does not convey the logic value, *per sé* but rather the direction of a transition does. For convenience high's and low's are designated as `1`'s and `0`'s in this write up.

6. The actual value of any given bit is determined by the direction of a data transition "event" which **always** occur at 32 μs (plus 16 μs for the first transition) points after the sync pulse ends. This means that, if at the sample time a signal is going positive then that bit is a logical `1`, if it is going negative then that bit is a logical `0`. (Or $t = 16 + (n \times 32)$ where $t =$ time of transition, $n =$ which transmitted bit number.)

   For example the first bit time of interest is 16 μs $(= 16 + (0 \times 32))$ after the sync pulse, the second is 48 μs $(= 16 + (1 \times 32))$ and the third is 80 μs $(= 16 + (2 \times 32))$, etc.

7. The data format conforms to what is ordinarily called "Manchester coded" data. The data **is not** in "Differential Manchester coded" format.

8. A sync signal is **always** present and consists of the data line going low for 48 μs and then high for another 48 μs. This makes up a 96 μs sync signal. (Unless the input wires are reversed, in which case the sync pulse, and all others, are complemented.) (See Figure 6, page 17 parts A and B.)

9. The duration of the preamble and sync pulse varies due to the variable number of preamble pulses that each source sends out. A preamble pulse is usually a pair of 16 μs transitions which take a total of 32 μs.

10. When decoding a command, all timing is based on the trailing edge of the sync pulse.

11. Commands are variable in length in all formats.

    A. Have either seven or eight bytes of information.

    B. The data bytes are 10 or 11 bits long. (10 for Format M, 11 for Format K.)

    C. The data bytes always start with a `1` or start bit.

    D. The data bytes always have an even parity bit as bit 10 of the byte. (Of course the parity may be odd with the start bit ignored. But then it's only a little bit.)

    E. A Format K data byte **always** has a stop bit `0` as its last bit which is not present in Format M data.

    F. There are unknown commands that the matrices may send out that are longer than those used to control Spectras. In general these are ignored. (See Section 1.4.4.1, page 35 and Section 1.4.4.2, page 36.)

12. In the first interpreted data byte, bit 0 is always set to a `1`. All other data bytes have bit 0 set to a `0`. Since the data arrives in "reverse bit order" the least significant bit arrives at the TXB-B immediately after the start bit.

Start times of command bytes in Format M data. In μs after the end of the sync pulse. Time is of the transition for the "s" (or start) bit of the byte. ($T = 16 + (n \times (32 \times 10))$ where T is the time of the start of the byte, n is the byte number and it must be remembered that each byte is 10 bits long.)

Byte	Start
1	16 μs
2	336 μs
3	656 μs
4	976 μs
5	1,296 μs
6	1,616 μs
7	1,936 μs
8	2,256 μs

By using the above information it is possible to uniquely and automatically identify several important characteristics of the data.

Polarity of wiring The polarity of the data line may be easily determined by examining the polarity order of the sync pulse. If it is negative (-) and then positive (+), this is considered to be "normal" wiring. If however, the polarity of the sync pulse is reversed, positive (+) then negative (-) it is "reversed" and all input bits will be complemented upon receipt. (For examples of a normal sync pulse, see: Figure 4, page 15 or Figure 6, page 17.)

The input circuitry of the TXB-AB, and TXB-B, primarily consists of a transformer and a MAX-485 to clean up the data. The MAX-485's output is not specified for inputs in a 200 mv wide guard band around 0 Volts[3]. On devices from MAXIM it appears that the output is **always** high when the input is at, or near, 0 Volts. However the output from a Linear Technology equivalent part an (LTC-485) is **either** high or low. (See Figure 13, page 27 for the output of the MAX-485 along with its input for the start of each command type.)

Pelco recently (August 2000) had several failures of "good" TXB-ABs that were using Linear Technology parts. (Of 19 new TXB-ABs received fresh from stock, five did not work with either a TC-8501 or an LTC-8801 matrix.) Having implemented the "Polarity detection" logic outlined above, it appears that the new TXB-B will not be affected by the inconsistent default output level of the LTC-485 (neé MAX-485) type of chip. This is important, since when the behavior of a part is unspecified, the manufacturer may change it at any time by making a "harmless" improvement in another area. It also allows Pelco to buy chips from alternate vendors.

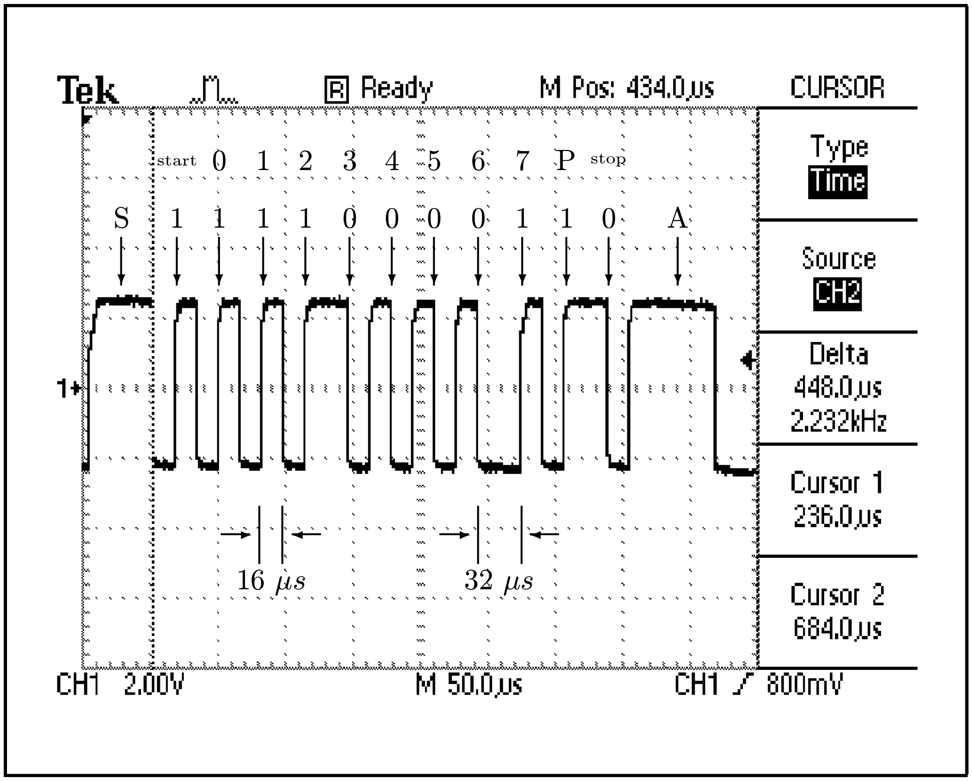
[3] The specification sheet for the MAX-485 says: "Receiver Output: If A > B by 200 mV, RO will be high; if A < B by 200 mV, RO will be low." A and B are the two input pins (6 and 7), RO is the output pin (1). On measuring two working TXB-ABs, it was found that the input voltage diffrence was between 75 and 80 mv.

Source of the command

All command bytes in Format M consist of a start bit, eight data bits and a parity bit. All command bytes in Format K consist of a start bit, eight data bits, a parity bit and a stop bit. (For detailed matrix decodes, see Section 1.4.2.1, page 31 and Section 1.4.2.2, page 31. For a detailed keyboard decode, see Section 1.4.2.3, page 32.)

This is convenient since all start bits are ones (`1`) and all stop bits are zeros (`0`). Thus if after detecting the parity bit a one (`1`) follows then it **must** be a start bit of another byte and the command is in Format M. If the bit following a parity bit is a zero (`0`) then it is a stop bit for that byte and the command is in Format K.

Being able to detect the command format by the end of the first command byte allows us to get rid of a jumper (as is currently needed) and all problems that getting it in the wrong position entail. In the current design the first command received is used to set parts of the mode of the TXB-AB and is then thrown out. This will no longer happen with the TXB-B.

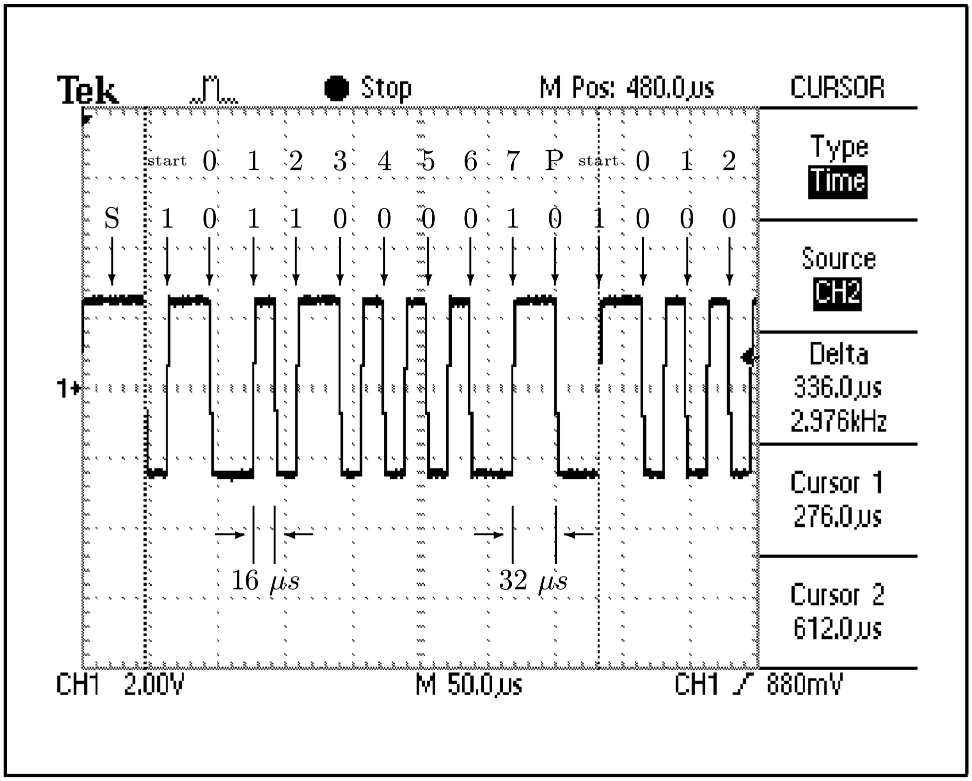


K5.PS 08-31-00 3:19p $RCSfile: k5.inc,v $

Figure 1. Format K, LTC-5136 first byte decoded from the data line

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 137.

[4] $Header: d:/ecr6171/RCS/k5.inc,v 1.8 2000-09-11 08:00:31-07 Hamilton Exp Hamilton $

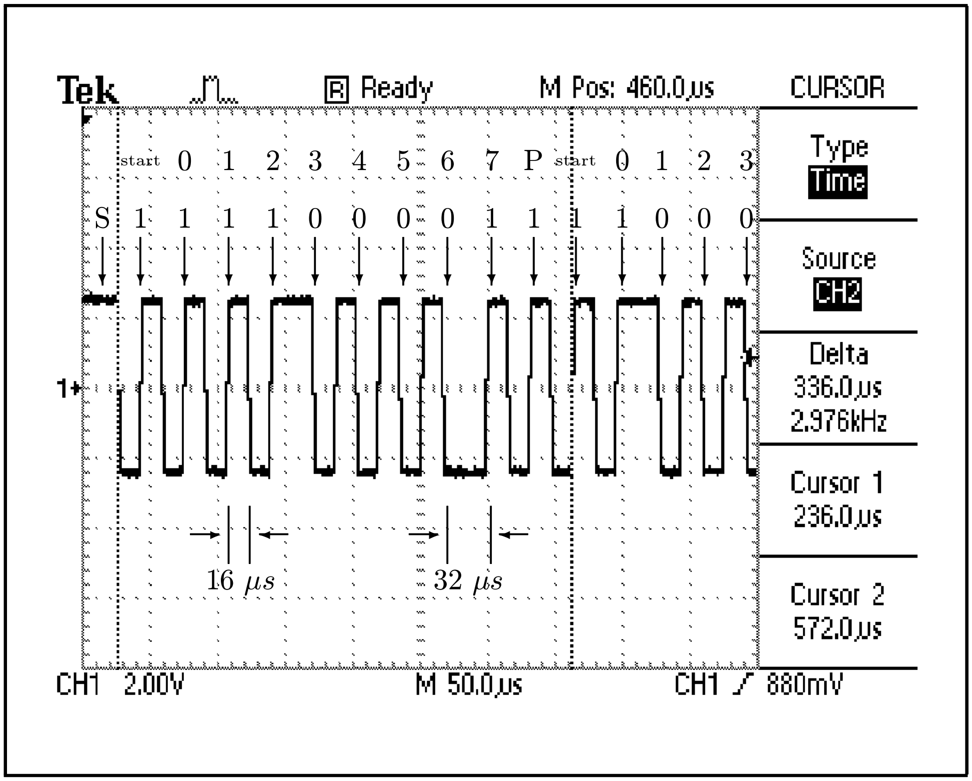


08.PS 09-01-00 8:57a $RCSfile: o8.inc,v $

Figure 2. Format M, TC-8501 first few bits decoded from the data line

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 9.

[5] `$Header: d:/ecr6171/RCS/o8.inc,v 1.7 2000-09-11 08:00:34-07 Hamilton Exp Hamilton $`



N6.PS 09-01-00 8:28a $RCSfile: n6.inc,v $

Figure 3. Format M, LTC-8801 first few bits decoded from the data line

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 137.

[6]`$Header: d:/ecr6171/RCS/n6.inc,v 1.7 2000-09-11 08:00:33-07 Hamilton Exp Hamilton $`

Sending Oscilloscope pictures to a computer

With a Tektronix TDS220 (and other similar Tektronix Oscilloscopes) configure the printer port as follows:

- Hard Copy Setup:
  - Layout Portrait
  - Format EPSIMAGE
  - Port RS232
- RS-232 setup:
  - Baud 19,200
  - Flow control None
  - EOL string CR/LF
  - Parity None

Computer setup.

- Boot the computer into "real DOS" not "DOS in a window" if using Kermit to collect data.
- On the computer set it up to receive serial data.
- Use a "null modem" or "Laplink" type cable to connect the two together.
- Using this configuration, it will take 1 to 2 minuits to transfer the data from the oscilloscope to the picture.

These pictures were collected using a 100 MHz Pentium computer running Kermit under DOS.

1.1.1.1 Format K Command Differences

A typical command may be seen in Figure 5, page 16 and Figure 4, page 15 which are pictures of the first byte of raw data as it appears on the data line. It should be pointed out that the input circuitry of the TXB-AB/TXB-B cleans up the data so that only logic `0`s and `1`s are ever present at the input to U1. (A fully decoded example of the first byte of a Format K command is shown in Figure 1, page 9. For a view of the cleaned up data see Figure 13, page 27.)

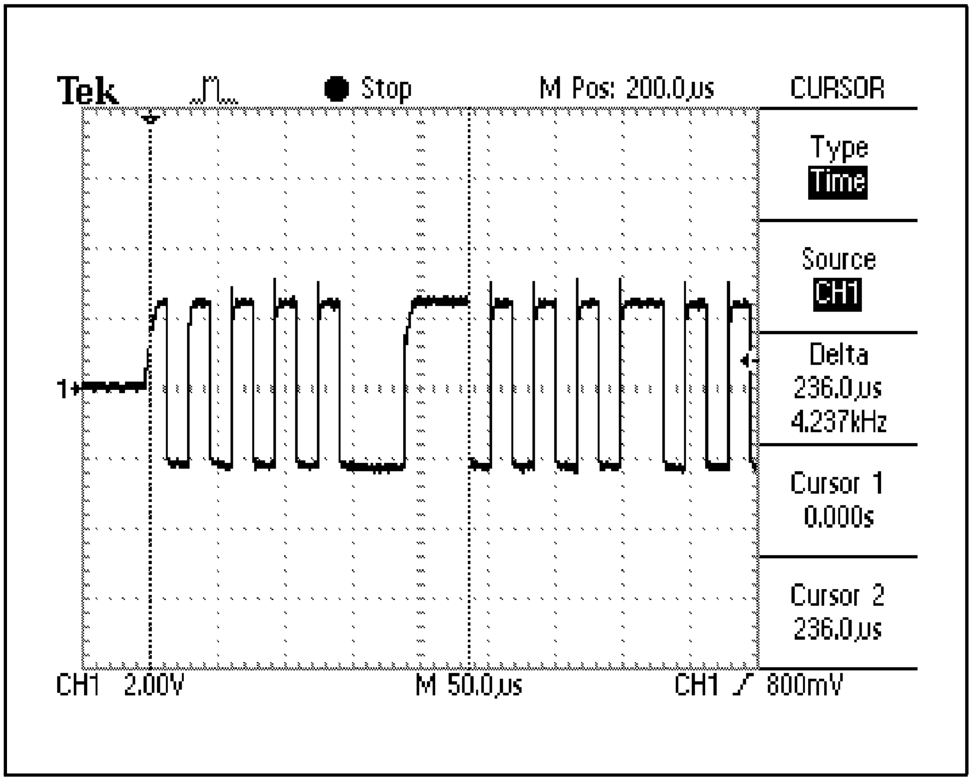
1. The first pulse of the preamble is positive going and about 14 μs long. (See Figure 4, page 15.)
2. There is a preamble of four pulses before the sync pulse. (See Figure 4, page 15.)
3. The duration of the preamble and sync pulse is 236 μs. (See Figure 4, page 15.)
4. In Format K each byte is sent with a following "quescent period" such that each byte takes up about 1.1 ms of time. With 728 μs of that being when each byte is being sent with a following dead period of about 368 μs. (This may be seen in Figure 5, page 16.)
5. A full eight byte command takes about 8 ms to be sent. (See Figure 5, page 16.)
6. There are two different times taken to send each command. When a command is initially sent, it is followed by a "short" (18 ms) interval and then by a "long" 46 ms interval when commands repeat. Most commands are always sent twice. The long interval only starts to occur if the key is held down to enable an "auto-repeat" type of action. (See Figure 8, page 19.
7. Each byte includes a start bit, eight bits of data, an even parity bit and a stop bit. (See Figure 1, page 9.)
8. Each byte is trailed by a positive starting "anti-sync" pulse that is high for 64 μs and low for 72 μs. The data line then returns to the quiescent level until the next command/data byte is sent. (This pulse may be seen in Figure 6, page 17.)
9. Each data byte consists of:

   A. A starting quiescent level.
   B. A preamble sequence.
   C. A sync pulse (low for 48 μs and high for 48 μs).
   D. A start bit (`1`).
   E. A "first byte of the message" bit. (Set to `1` on the first data byte, on all others it is set to `0`.)
   F. Eight data bits (either `0` or `1`).
   G. An even parity bit (either `0` or `1`).
   H. A stop bit (`0`).
   I. A trailing "anti-sync" pulse.
   J. An ending quiescent level.

K. This would look like this: "`Q+ppppSsddddddddPtAQ`" with:

- `+` A positive going first pulse.
- `A` An anti-sync pulse.
- `d` Eight data pulses (`0` or `1`).
- `P` An even parity pulse (`0` or `1`).
- `p` A preamble pulse.
- `Q` The quiescent level of the three state data line.
- `S` The sync pulse (low for 48 μs and high for 48 μs).
- `s` A start pulse (`1`).
- `t` A stop pulse (`0`).

1.1.1.2 Typical oscilloscope screen shots for Format K commands

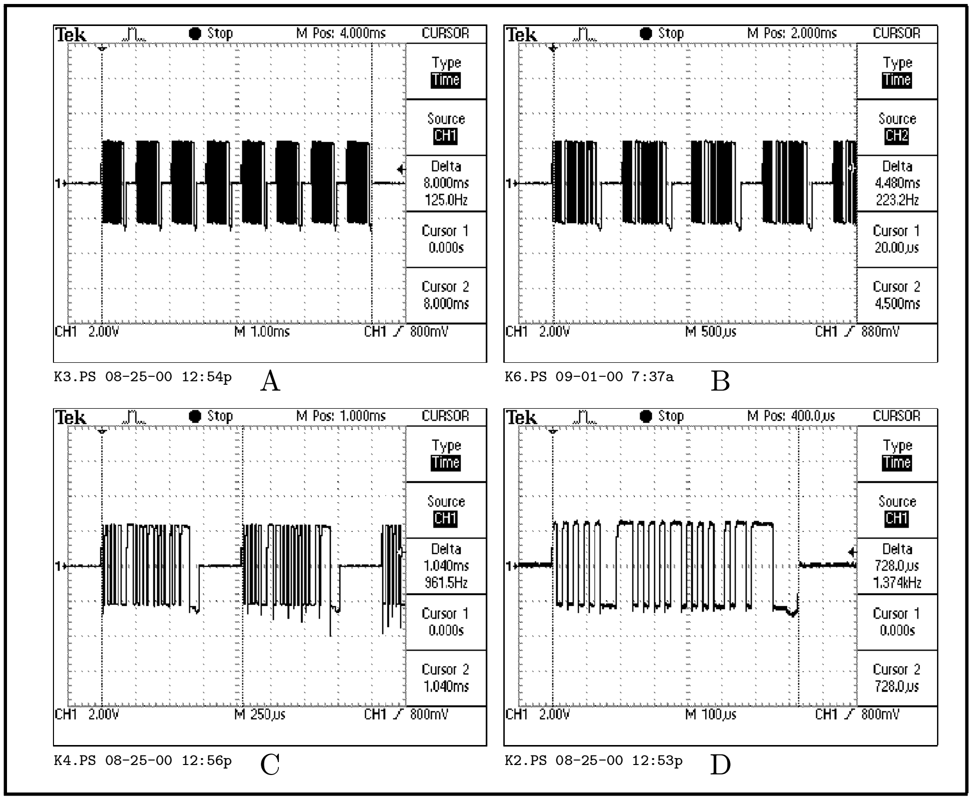


K1.PS 08-25-00 12:51p $RCSfile: k1.inc,v $

Figure 4. Format K, LTC-5136 details of first byte

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 9.

[7]$Header: d:/ecr6171/RCS/k1.inc,v 1.5 2000-09-11 08:00:31-07 Hamilton Exp Hamilton $

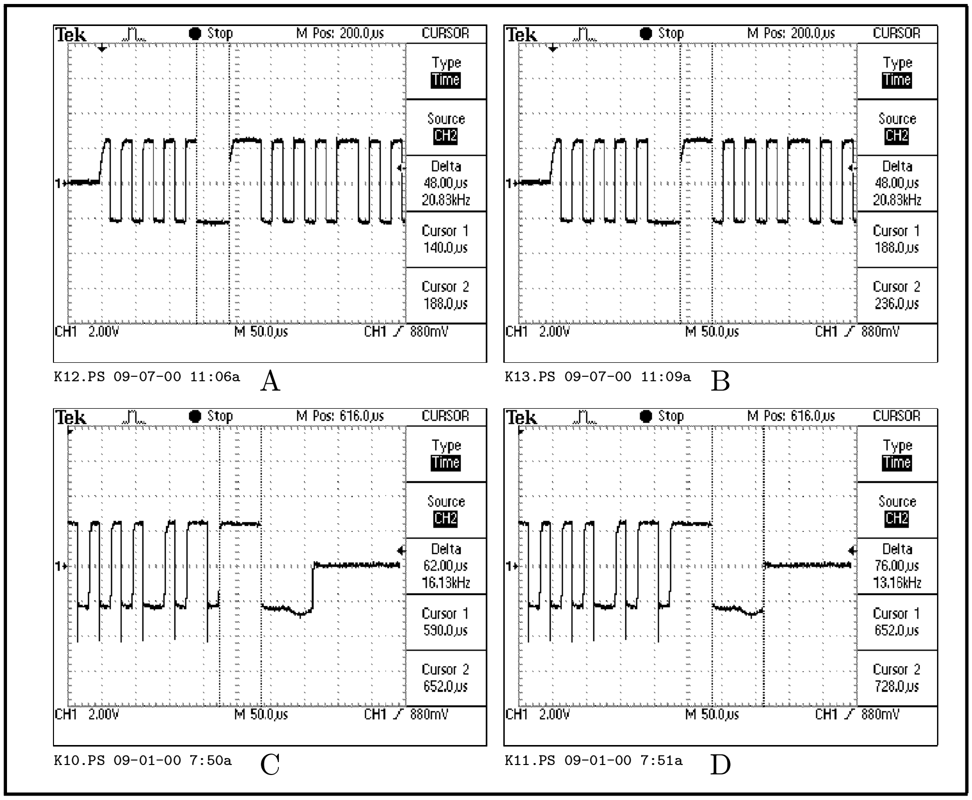


K6.PS, K3.PS, K4.PS, K2.PS $RCSfile: kstart.inc,v $

Figure 5. Format K, LTC-5136 Typical command

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 9.
Figure A	Full command.
Figure B	Start of command.
Figure C	First two command bytes.
Figure D	First command byte.

[8] $Header: d:/ecr6171/RCS/kstart.inc,v 1.6 2000-11-27 14:22:49-08 Hamilton Exp Hamilton $

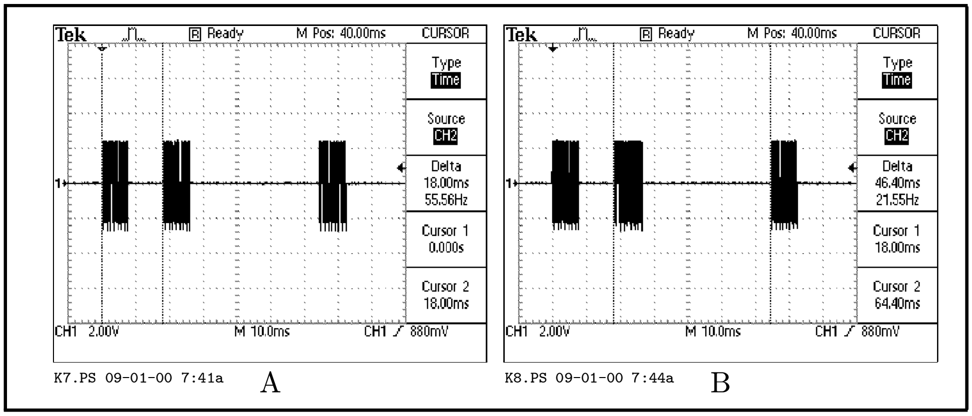


K12.PS, K13.PS, K10.PS, K11.PS $RCSfile: kstrtend.inc,v $

Figure 6: Format K, LTC-5136 Various timing information for the start and end of a command byte.

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 9.
Figure A	Start of all sync pulses. (Applies to all formats.)
Figure B	End of all sync pulses. (Applies to all formats.)
Figure C	Start of an anti-sync pulse.
Figure D	End of an anti-sync pulse.

[9] $Header: d:/ecr6171/RCS/kstrtend.inc,v 1.5 2000-11-27 14:22:49-08 Hamilton Exp Hamilton $

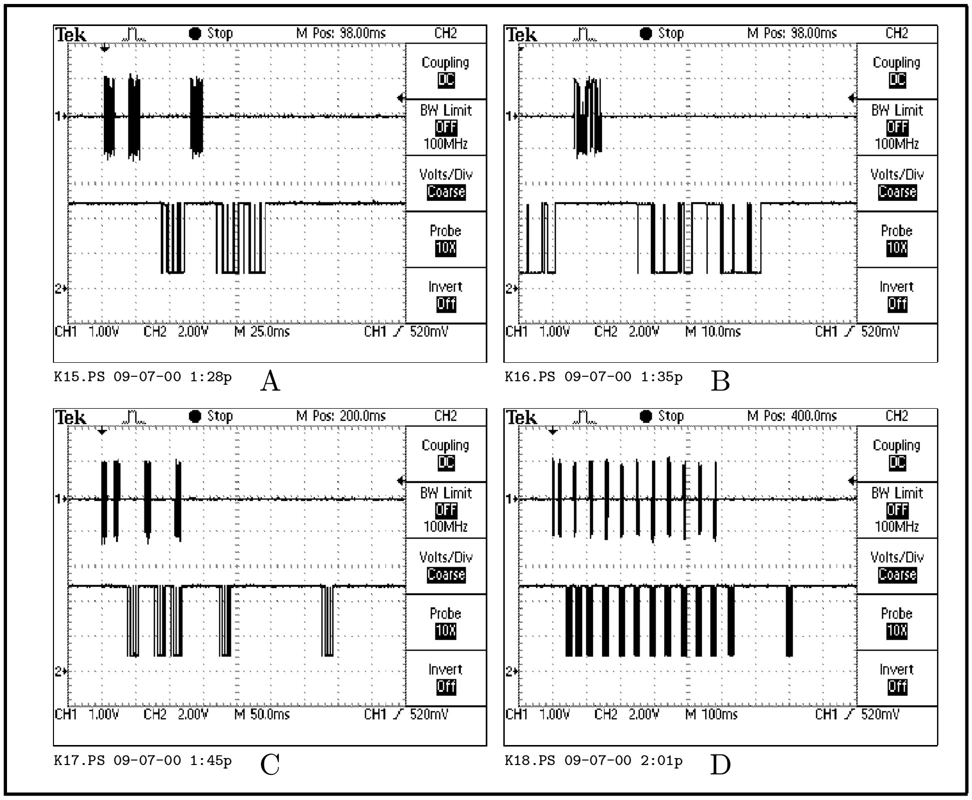


K7.PS, K8.PS $RCSfile: krepcmns.inc,v $

Figure 7. Format K, LTC-5136 what happens with most commands

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 9.
Figure A	Timing details for the first repeated command.
Figure B	Timing details for the second repeated command.

[10] $Header: d:/ecr6171/RCS/krepcmns.inc,v 1.3 2000-09-11 08:00:30-07 Hamilton Exp Hamilton $



K15.PS, K16.PS, K17.PS, K18.PS $RCSfile: kreprobs.inc,v $

Figure 8. Format K, LTC-5136 what happens with “fast” repeating commands

Item	Setting/Use
Trace 1	Raw data line
Trace 2	Output of U2 pin 18, the data received from a Spectra.
Command	Open Iris sent to Camera 9.
Figure A	Typical single button push.
Figure B	Details of center of Figure A.
Figure C	Slightly longer button push.
Figure D	Long button push.

[11] `$Header: d:/ecr6171/RCS/kreprobs.inc,v 1.3 2000-09-11 08:00:30-07 Hamilton Exp Hamilton $`

1.1.1.3 Format M Command Differences

A typical command may be seen in Figure 10, page 23 and Figure 12, page 25 which are pictures of the first byte of raw data as it appears on the data line for old (TC-8501) new (LTC-8801) types of matrices. It should be pointed out that the input circuitry of the TXB-AB/TXB-B cleans up the data so that only logic `0`s and `1`s are ever present at the input to U1. (See Figure 13, page 27.)

1. On old matrices the first pulse is positive going and on new matrices the first pulse is negative going. (Figure 9, page 22 and Figure 11, page 24)
2. There is a preamble of five pulses before the sync pulse. (Figure 9, page 22 and Figure 11, page 24)
3. The duration of the preamble and sync pulse is 278 μs on a TC-8501 type system and it is 256 μs on an LTC-8801 type system. (Figure 9, page 22 and Figure 11, page 24)
4. In Format M all bytes are sent in a single block with only one sync pulse.
5. There are two different times taken to send each command for each matrix types:
   - TC-8501 Command duration is 2.6 ms for a seven byte command. (Figure 10, page 23)
   - LTC-8801 Command duration is 2.9 ms for a seven byte command. (Figure 12, page 25)
6. There are two different repetition rates for commands that are sent continuously which varies depending on the matrix type:
   - TC-8501 Commands are resent every 53 ms. (Figure 10, page 23)
   - LTC-8801 Commands are resent every 32.8 ms. (Figure 12, page 25)
7. The anti-sync pulse for a TC-8501 matrix is high for 56 μs and is shown in Figure 10, page 23. For an LTC-8801 it is high for 64 μs and is shown in Figure 12, page 25.
8. Each byte includes a start bit, eight bits of data and an even parity bit.
9. Each full command is trailed by a single stop bit (`0`). I.e. there is only one stop bit for the full command *vs* one for each byte in Format K.
10. The full message consists of:
    A. A starting quiescent level.
    B. A preamble sequence.
    C. A sync pulse (low for 48 μs and high for 48 μs).
    D. Then the requisite number of information bytes (7 or 8 depending on the command). Within each of the information bytes the following holds:
       a. A start bit of `1`.

   b. A "first byte of the message" bit. (Set to `1` on the first data byte, on all others it is set to `0`.)
   c. Eight data bits.
   d. An even parity bit.
   e. A stop bit for the **full** message. Which is always `0`.
   f. A trailing long high which is either 64 or 56 μs long.

E. An ending quiescent level.

F. A full seven byte message would look like this: "`Q±ppppS sddddddddP sddddddddP sddddddddP sddddddddP sddddddddP sddddddddP sddddddddPA tAQ`" with:

   ± Either a positive or negative going first pulse which varies by matrix type.

   `A` An anti-sync pulse.

   `d` Eight data pulses (either `0` or `1`).

   `p` A preamble pulse.

   `P` An even parity pulse (either `0` or `1`).

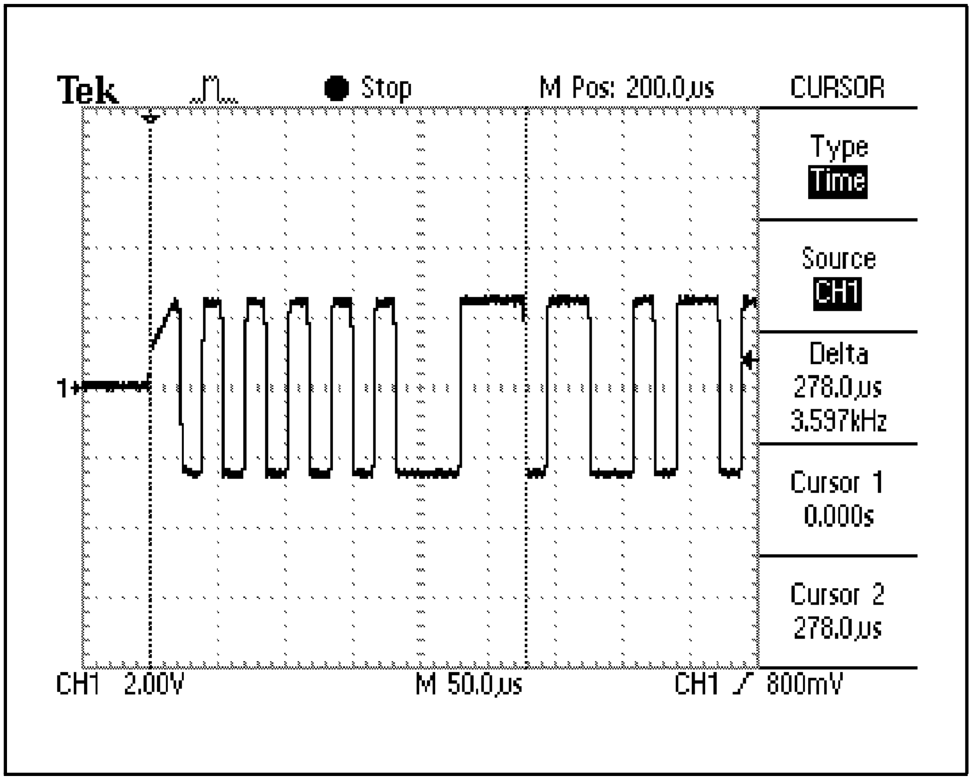
   `Q` The quiescent level of the data line.

   `S` The sync pulse (low for 48 μs and high for 48 μs).

   `s` A start pulse (`1`).

   `t` A stop pulse (`0`).

1.1.1.4 Typical oscilloscope screen shots for Format M commands, from a TC-8501 Matrix)

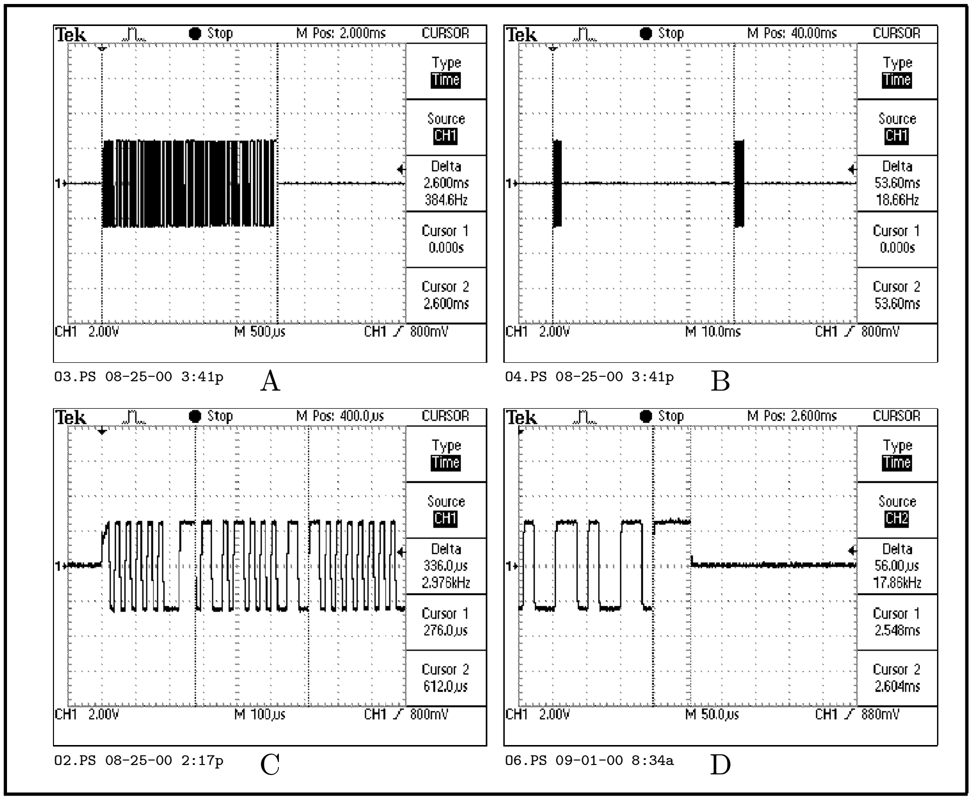


01.PS 08-25-00 2:15p $RCSfile: o1.inc,v $

Figure 9. Format M, TC-8501 details of first byte

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 9.

[12]$Header: d:/ecr6171/RCS/o1.inc,v 1.5 2000-09-11 08:00:34-07 Hamilton Exp Hamilton $



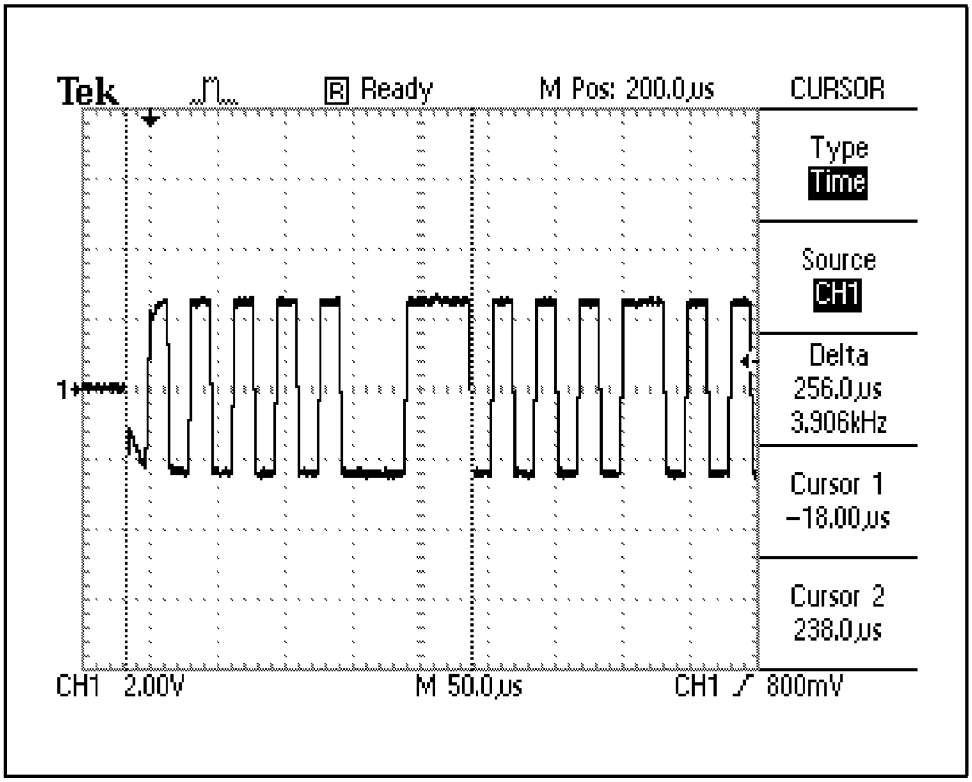
O3.PS, O4.PS, O2.PS, O6.PS $RCSfile: mocmnds.inc,v $

Figure 10. Format M, TC-8501 Basic command information

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 9.
Figure A	Typical command.
Figure B	Repeated command.
Figure C	First few bytes.
Figure D	Timing of anti-sync pulse.

[13] `$Header: d:/ecr6171/RCS/mocmnds.inc,v 1.5 2000-09-11 08:00:30-07 Hamilton Exp Hamilton $`

1.1.1.5 Typical oscilloscope screen shots for Format M commands, from an LTC-8801 Matrix

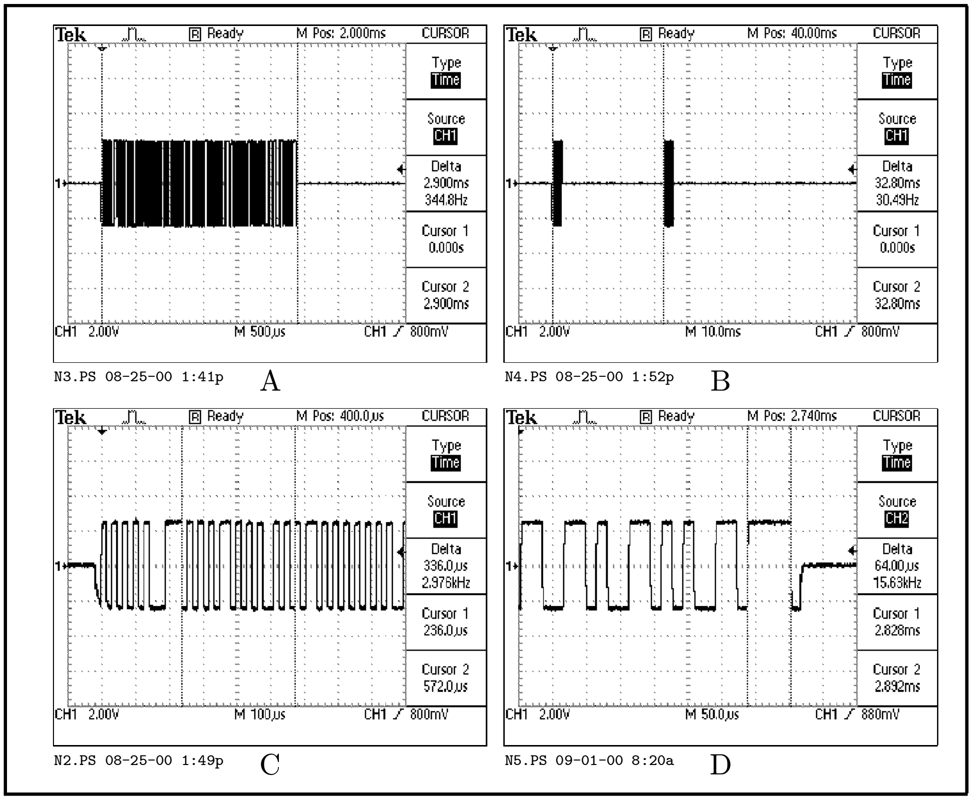


N1.PS 08-25-00 1:46p $RCSfile: n1.inc,v $

Figure 11. Format M, LTC-8801 details of the command's start

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 137.

[14]$Header: d:/ecr6171/RCS/n1.inc,v 1.5 2000-09-11 08:00:33-07 Hamilton Exp Hamilton $



N3.PS, N4.PS, N2.PS, N5.PS $RCSfile: mncmnds.inc,v $

Figure 12. Format M, TC-8801 Basic command information

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 137.
Figure A	Typical command.
Figure B	Repeated command.
Figure C	First few bytes.
Figure D	Timing of anti-sync pulse.

[15]$Header: d:/ecr6171/RCS/mncmnds.inc,v 1.5 2000-09-11 08:00:30-07 Hamilton Exp Hamilton $

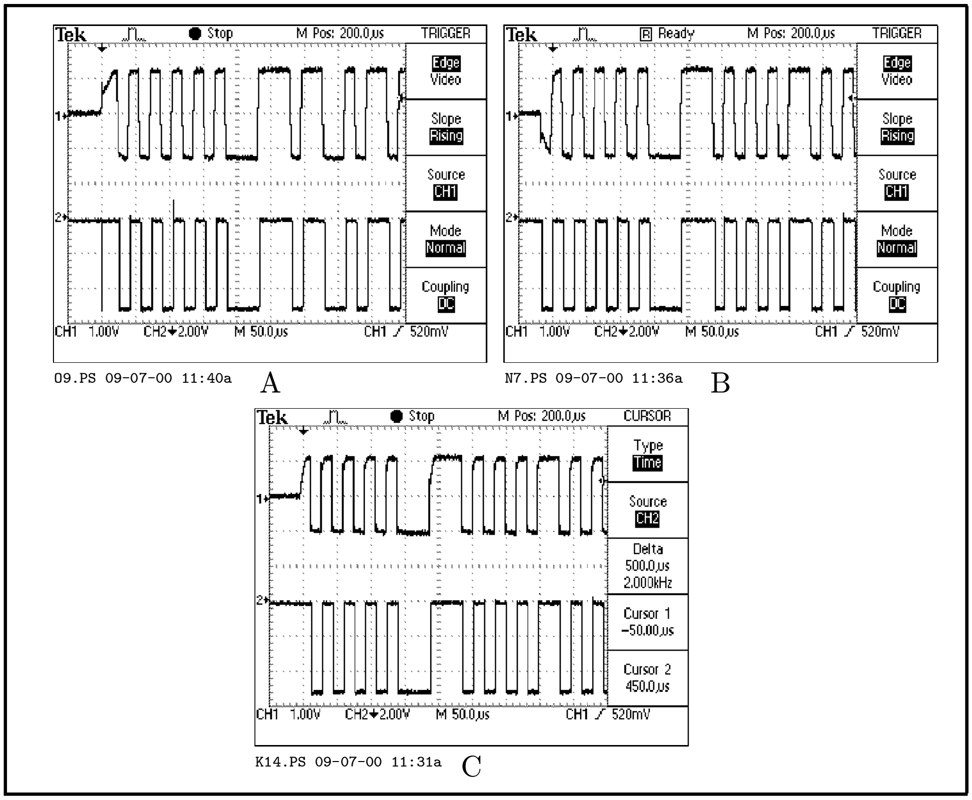
1.2 Time to communicate with a Spectra

The time that a TXB-B takes to send a protocol "D" message to a Spectra may be calculated as follows:

1. Each command is seven bytes in length.
   - A. Sync
   - B. Address
   - C. Command 1
   - D. Command 2
   - E. Data 1
   - F. Data 2
   - G. Checksum
2. Each byte is 10 bits long.
   - A. A start bit
   - B. Eight data bits
   - C. A stop bit
3. Giving a total of 70 bits (7 bytes $\times$ 10 bits).
4. At 2400 baud, each bit takes .4167 ms to send. ($\frac{1}{2400} = .4167\text{ms}$)
5. Therefore each command takes 29.17 ms to send. ($\frac{70}{2400} = 29.17\text{ms}$)

1.3 Outputs of the MAX-485 input circuity

The MAX-485 used in the input logic has the following outputs. In each case the output is shown along with its input signal. It is important to notice what happens to the first "little" pulse of either + or −, when it goes through the MAX-485.



O9.PS, N7.PS, K14.PS $RCSfile: inputckt.inc,v $

Figure 13. Capabilities of the TXB-B input circuit

Item	Setting/Use
Trace 1	Raw data line
Trace 2	Output of U3 pin 1, the data sent to U1.
Command	Open Iris sent to Camera ....
Figure A	Format M, from a TC-8501 matrix to Camera 9.
Figure B	Format M, from a LTC-8801 matrix to Camera 137.
Figure C	Format K, from a LTC-5136 keyboard to Camera 9.

[16] `$Header: d:/ecr6171/RCS/inputckt.inc,v 1.5 2000-11-27 14:22:48-08 Hamilton Exp Hamilton $`

1.4 Typical Burle Commands

This data was taken when commanding a camera at address 9 or 137. 137 is used with the new Matrix, 9 is used with the old Matrix.

In the following tables these are the abbreviations that are used:

`+`	First pulse goes from quiescent to high.
`-`	First pulse goes from quiescent to low.
`0`	Manchester coded data transition from high to low.
`1`	Manchester coded data transition from low to high.
`A`	Anti-Sync pulse. On an LTC-8801 matrix, this is high for 64 μs (Figure 12, page 25). On an TC-8501 matrix, this is a trailing long pulse high for 56 μs (Figure 10, page 23).
`P`	Preamble pulse.
`Q`	Quiescent level of the data line.
`S`	Sync pulse, low for 48 μs and then high for 48 μs.

1.4.1 Joystick maximum upper left corner

1.4.1.1 Old Matrix

This command reads out as:[17]

`Q +PPPPPS`

`1011000010 1000000001 1000100000 1001000000 1000000001 1010100001 1001110000 0AQ`

Byte	Use	Raw Binary	Reversed Hex
—	Quiescent	Q	—
0	Preamble, sync	+ P P P P P S	—
1	1st byte, length	1 0 1 1 0 0 0 0 1 0	0x86
2	High address	1 0 0 0 0 0 0 0 0 1	0x00
3	Low address	1 0 0 0 1 0 0 0 0 0	0x08
4	OpCode	1 0 0 1 0 0 0 0 0 0	0x04
5	Data 1	1 0 0 0 0 0 0 0 0 1	0x00
6	Data 2	1 0 1 0 1 0 0 0 0 1	0x0A
7	Checksum	1 0 0 1 1 1 0 0 0 0	0x1C or 0x9C
?	Unknown	0 A Q	—

1.4.1.2 New Matrix

This command reads out as[18]:

`Q -PPPPPS`

`1111000011 1100000000 1000100000 1000100000 1111100001 1000111101 1010100001 1100101000 0A0Q`

Byte	Use	Raw Binary	Reversed Hex
—	Quiescent	Q	—
0	Preamble, sync	- P P P P P S	—
1	1st byte, length	1 1 1 1 0 0 0 0 1 1	0x87
2	High address	1 1 0 0 0 0 0 0 0 0	0x01
3	Low address	1 0 0 0 1 0 0 0 0 0	0x08
4	OpCode	1 0 0 0 1 0 0 0 0 0	0x08
5	Data 1	1 1 1 1 1 0 0 0 0 1	0x0F
6	Data 2	1 0 0 0 1 1 1 1 0 1	0x78
7	Data 3	1 0 1 0 1 0 0 0 0 1	0x0A
8	Checksum	1 1 0 0 1 0 1 0 0 0	0x29 or 0x129
?	Unknown	0 A 0 Q	—

[17] Oscilloscope screen shots are from 24AUG00, pages 5 → 8.
[18] Oscilloscope screen shots are from 24AUG00, pages 1 → 4.

1.4.1.3 LTC-5136

This command reads out as:[19]

```
Q +PPPPS 11110000110 AQ
Q +PPPPS 10000000010 AQ
Q +PPPPS 10001000000 AQ
Q +PPPPS 10001000000 AQ
Q +PPPPS 11111000010 AQ
Q +PPPPS 10001111010 AQ
Q +PPPPS 10101000010 AQ
Q +PPPPS 10001010010 AQ
```

Byte	Use	Raw Binary	Reversed Hex
—	Quiescent	Q	—
1	1^{st} byte, length	1 1 1 1 0 0 0 0 1 1 0	0x87
2	High address	1 0 0 0 0 0 0 0 0 1 0	0x00
3	Low address	1 0 0 0 1 0 0 0 0 0 0	0x08
4	OpCode	1 0 0 0 1 0 0 0 0 0 0	0x08
5	Data 1	1 1 1 1 1 0 0 0 0 1 0	0x0F
6	Data 2	1 0 0 0 1 1 1 1 0 1 0	0x78
7	Data 3	1 0 1 0 1 0 0 0 0 1 0	0x0A
8	Checksum	1 0 0 0 1 0 1 0 0 1 0	0x28 or 0x128

[19]Oscilloscope screen shots are from 25AUG00, pages 2 → 9.

1.4.2 Iris command

1.4.2.1 Old Matrix

This command reads out as: `Q +PPPPPS`
`1011000010 1000000001 1000100000 1001000000 1000000001 1010000000 1001010001 0AQ`

Byte	Use	Raw Binary	Reversed Hex
—	Quiescent	Q	—
0	Preamble, sync	+ P P P P P S	—
1	1^{st} byte, length	1 0 1 1 0 0 0 0 1 0	0x86
2	High address	1 0 0 0 0 0 0 0 0 1	0x00
3	Low address	1 0 0 0 1 0 0 0 0 0	0x08
4	OpCode	1 0 0 1 0 0 0 0 0 0	0x04
5	Data 1	1 0 0 0 0 0 0 0 0 1	0x00
6	Data 2	1 0 1 0 0 0 0 0 0 0	0x02
7	Checksum	1 0 0 1 0 1 0 0 0 1	0x14 or 0x94
?	Unknown	0 A Q	—

1.4.2.2 New Matrix

This command reads out as: `Q -PPPPPS`
`1111000011 1000000001 1000000001 1000100000 1000000001 1010000000 10000000011 100010001 0A0Q`

Byte	Use	Raw Binary	Reversed Hex
—	Quiescent	Q	—
0	Preamble, sync	- P P P P P S	—
1	1^{st} byte, length	1 1 1 1 0 0 0 0 1 1	0x87
2	High address	1 0 0 0 0 0 0 0 0 1	0x00
3	Low address	1 0 0 0 0 0 0 0 0 1	0x00
4	OpCode	1 0 0 0 1 0 0 0 0 0	0x08
5	Data 1	1 0 0 0 0 0 0 0 0 1	0x00
6	Data 2	1 0 1 0 0 0 0 0 0 0	0x02
7	Data 3	1 0 0 0 0 0 0 0 0 1	0x00
8	Checksum	1 1 0 0 0 1 0 0 0 1	0x11 or 0x91
?	unknown	0 A 0 Q	—

1.4.2.3 LTC-5136

This command reads out as:[20]

```
Q +PPPPS 11110000110 AQ
Q +PPPPS 10000000010 AQ
Q +PPPPS 11000000000 AQ
Q +PPPPS 10001000000 AQ
Q +PPPPS 10000000010 AQ
Q +PPPPS 10010000000 AQ
Q +PPPPS 10000000010 AQ
Q +PPPPS 10010100010 AQ
```

Byte	Use	Raw Binary	Reversed Hex
—	Quiescent	Q	—
1	1^{st} byte, length	1 1 1 1 0 0 0 0 1 1 0	0x87
2	High address	1 0 0 0 0 0 0 0 0 1 0	0x00
3	Low address	1 1 0 0 0 0 0 0 0 0 0	0x01
4	OpCode	1 0 0 0 1 0 0 0 0 0 0	0x08
5	Data 1	1 0 0 0 0 0 0 0 0 1 0	0x00
6	Data 2	1 0 0 1 0 0 0 0 0 0 0	0x04
7	Data 3	1 0 0 0 0 0 0 0 0 1 0	0x00
8	Checksum	1 0 0 1 0 1 0 0 0 1 0	0x14 or 0x94

[20] Oscilloscope screen shots are from 27JUN00, page 37.

1.4.3 Shot 1 Command

1.4.3.1 Old Matrix

This command reads out as:[21]

```
Q +PPPPPS
1011000010 1000000001 1000100000 1111000000 1101000001 1100000000
1110110001 0AQ
```

Byte	Use	Raw Binary	Reversed Hex
—	Quiescent	Q	—
0	Preamble, sync	+ P P P P P S	—
1	1^{st} byte, length	1 0 1 1 0 0 0 0 1 0	0x86
2	High address	1 0 0 0 0 0 0 0 0 1	0x00
3	Low address	1 0 0 0 1 0 0 0 0 0	0x08
4	OpCode	1 1 1 1 0 0 0 0 0 0	0x07
5	Data 1	1 1 0 1 0 0 0 0 0 1	0x05
6	Data 2	1 1 0 0 0 0 0 0 0 0	0x01
7	Checksum	1 1 1 0 1 1 0 0 0 1	0x1B or 0x9B
?	unknown	0 A Q	—

1.4.3.2 New Matrix

This command reads out as:[22]

```
Q -PPPPS
1011000010 1100000000 1000100000 1111000000 1101000001 1100000000
1001110000 0AQ
```

Byte	Use	Raw Binary	Reversed Hex
—	Quiescent	Q	—
0	Preamble, sync	- P P P P P S	—
1	1^{st} byte, length	1 0 1 1 0 0 0 0 1 0	0x86
2	High address	1 1 0 0 0 0 0 0 0 0	0x01
3	Low address	1 0 0 0 1 0 0 0 0 0	0x08
4	OpCode	1 1 1 1 0 0 0 0 0 0	0x07
5	Data 1	1 1 0 1 0 0 0 0 0 1	0x05
6	Data 2	1 1 0 0 0 0 0 0 0 0	0x01
7	Checksum	1 0 0 1 1 1 0 0 0 0	0x1C or 0x9C
?	unknown	0 A Q	—

[21] Oscilloscope screen shots are from 24AUG00, pages 12 → 14.
[22] Oscilloscope screen shots are from 24AUG00, pages 9 → 11.

1.4.3.3 LTC-5136

This command reads out as:[23]

```
Q +PPPPS 10110000100 AQ
Q +PPPPS 10000000010 AQ
Q +PPPPS 10000000010 AQ
Q +PPPPS 11110000000 AQ
Q +PPPPS 11010000010 AQ
Q +PPPPS 11000000000 AQ
Q +PPPPS 11100100000 AQ
```

Byte	Use	Raw Binary	Reversed Hex
—	Quiescent	Q	—
1	1st byte, length	1 0 1 1 0 0 0 0 1 0 0	0x86
2	High address	1 0 0 0 0 0 0 0 0 1 0	0x00
3	Low address	1 0 0 0 0 0 0 0 0 1 0	0x00
4	OpCode	1 1 1 1 0 0 0 0 0 0 0	0x07
5	Data 1	1 1 0 1 0 0 0 0 0 1 0	0x05
6	Data 2	1 1 0 0 0 0 0 0 0 0 0	0x01
7	Checksum	1 1 1 0 0 1 0 0 0 0 0	0x13 or 0x93

[23]Oscilloscope screen shots are from 10JUL00, pages 13, 15 and 16.

1.4.4 Unknown pulse trains

1.4.4.1 Old matrix system, first unknown pulse group

This command reads out as:[24]

`Q +PPPPPS`

`1110100011 1100101101 1001111001 1000000001 1101000001 1010000000 1101000001 1100000000 1001100001 1010000000 1000100000 1110010101 0AQ`

Byte	Raw Binary	Reversed Hex
—	Q	—
0	+ P P P P P S	—
1	1 1 1 0 1 0 0 0 1 1	0x8B
2	1 1 0 0 1 0 1 1 0 1	0x69
3	1 0 0 1 1 1 1 0 0 1	0x3C
4	1 0 0 0 0 0 0 0 0 1	0x00
5	1 1 0 1 0 0 0 0 0 1	0x05
6	1 0 1 0 0 0 0 0 0 0	0x02
7	1 1 0 1 0 0 0 0 0 1	0x05
8	1 1 0 0 0 0 0 0 0 0	0x01
9	1 0 0 1 1 0 0 0 0 1	0x0C
10	1 0 1 0 0 0 0 0 0 0	0x02
11	1 0 0 0 1 0 0 0 0 0	0x08
12	1 1 1 0 0 1 0 1 0 1	0x153 or 0x53
?	0 A Q	—

[24]Oscilloscope screen shots are from 21AUG00, pages 3 → 14.

1.4.4.2 Old matrix system, second unknown pulse group

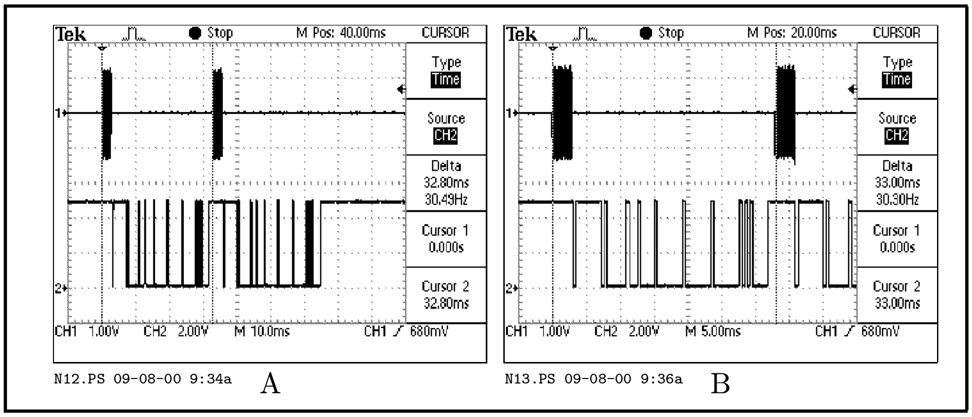
This command reads out as:[25]

```
Q +PPPPPS
1110010011 1101001101 1000100101 1000000001 1000000001 1000000001 1000000001
1000000001 1000000001 1000000001 1000000001 1000000001 1000000001 1000000001
1000000001 1000000001 1000000001 1000000001 1000000001 1000000100 0AQ
```

Byte	Raw Binary	Reversed Hex
—	Q	—
0	+ P P P P P S	—
1	1 1 1 0 0 1 0 0 1 1	0x93
2	1 1 0 1 0 0 1 1 0 1	0x65
3	1 0 0 0 1 0 0 1 0 1	0x48
4	1 0 0 0 0 0 0 0 0 1	0x00
5	1 0 0 0 0 0 0 0 0 1	0x00
6	1 0 0 0 0 0 0 0 0 1	0x00
7	1 0 0 0 0 0 0 0 0 1	0x00
8	1 0 0 0 0 0 0 0 0 1	0x00
9	1 0 0 0 0 0 0 0 0 1	0x00
10	1 0 0 0 0 0 0 0 0 1	0x00
11	1 0 0 0 0 0 0 0 0 1	0x00
12	1 0 0 0 0 0 0 0 0 1	0x00
13	1 0 0 0 0 0 0 0 0 1	0x00
14	1 0 0 0 0 0 0 0 0 1	0x00
15	1 0 0 0 0 0 0 0 0 1	0x00
16	1 0 0 0 0 0 0 0 0 1	0x00
17	1 0 0 0 0 0 0 0 0 1	0x00
18	1 0 0 0 0 0 0 0 0 1	0x00
19	1 0 0 0 0 0 0 0 0 1	0x00
20	1 0 0 0 0 0 0 1 0 0	0x40 or 0x140
?	0 A Q	—

[25]Oscilloscope screen shots are from 21AUG00, pages 16 → 36.

1.5 Misc command information

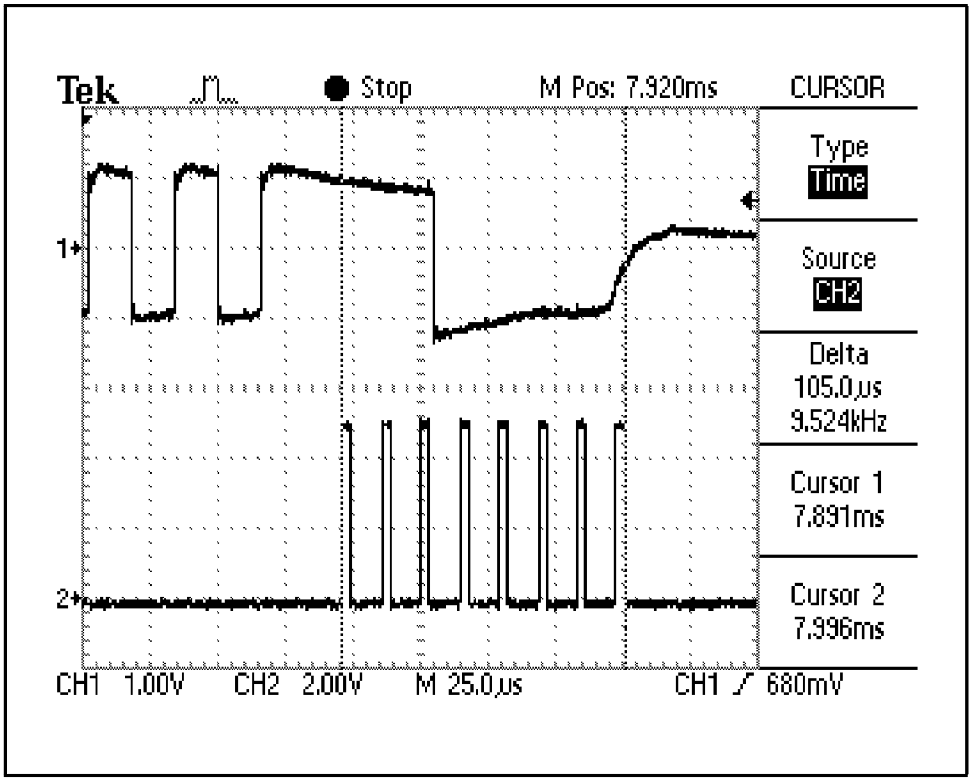


N12.PS, N13.PS $RCSfile: munexpe2.inc,v $

Figure 14. Format M, LTC-8801 Timing of commands for an Open iris command

Item	Setting/Use
Trace 1	Raw data line
Trace 2	Output of U1 pin 8, the data sent to a Spectra.
Command	Open Iris sent to Camera 1.
Figure A	Details of first command pair.
Figure B	Expanded details of Figure A.

[26] `$Header: d:/ecr6171/RCS/munexpe2.inc,v 1.2 2000-11-27 14:22:49-08 Hamilton Exp Hamilton $`



K22.PS 09-08-00 10:03a $RCSfile: k22.inc,v $

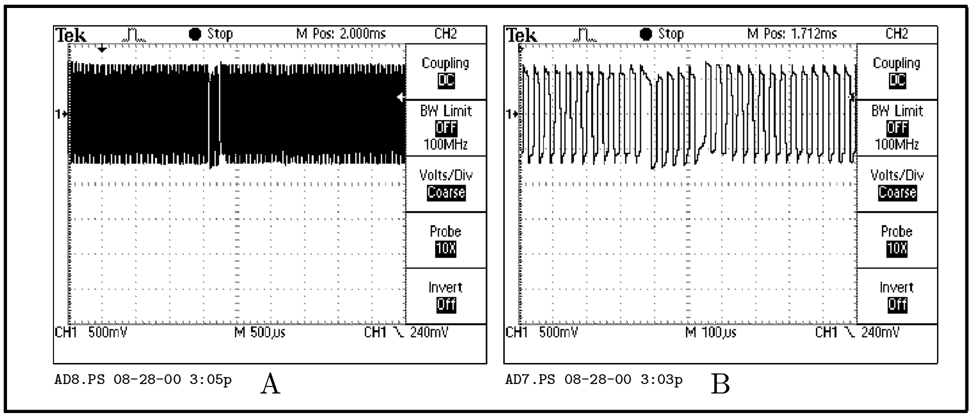
Figure 15: Format K, LTC-5136 timing of sending an 8 byte command between U1 and U2 with a TXB-AB/TXB-AD.

Item	Setting/Use
Trace 1	Raw data line
Trace 2	U2-4 (`dt`) to Spectra.
Command	Call 1, Camera 1, Monitor 1.

[27] `$Header: d:/ecr6171/RCS/k22.inc,v 1.4 2000-11-27 14:22:49-08 Hamilton Exp Hamilton $`

APPENDIX A

A American Dynamics, command data

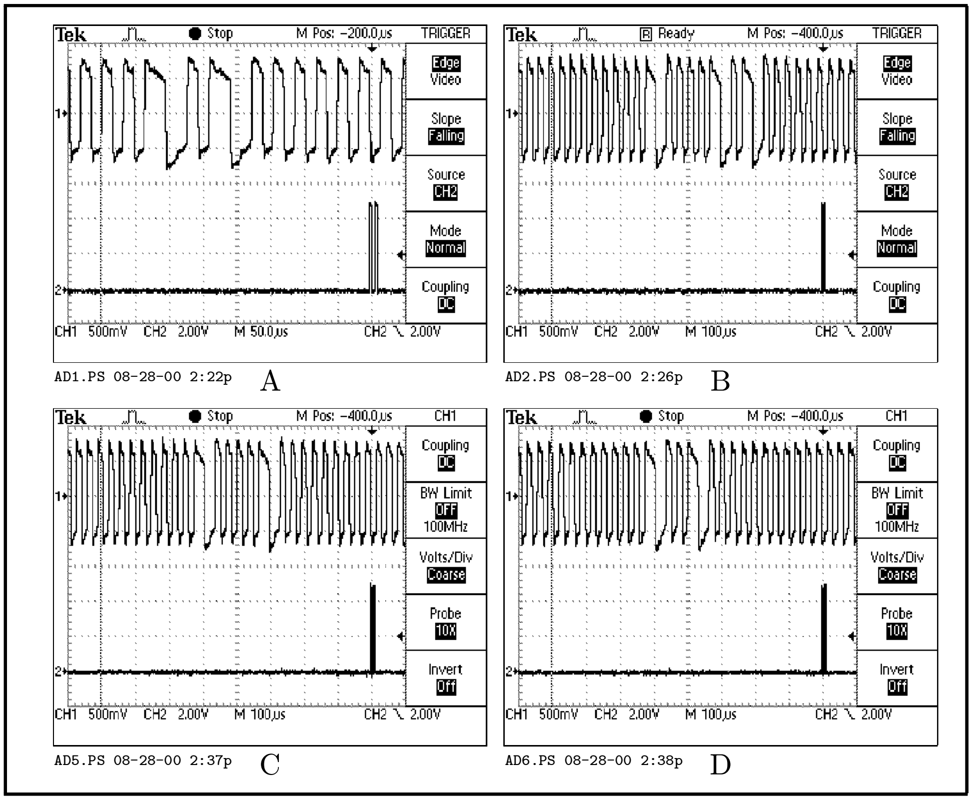


AD8.PS, AD7.PS $RCSfile: adcmnds.inc,v $

Figure A-1. Format AD, Typical commands, unknown

Item	Setting/Use
Trace 1	Raw data line
Command	Unknown.
Figure A	Normal speed, the "glitch" is the command.
Figure B	Glitch from Figure A.

[28]$Header: d:/ecr6171/RCS/adcmnds.inc,v 1.9 2000-11-27 14:22:47-08 Hamilton Exp Hamilton $



AD1.PS, AD2.PS, AD5.PS, AD6.PS $RCSfile: adcmnds.inc,v $

Figure A-2. Format AD, Typical commands from a TXB-AB/TXB-AD.

Item	Setting/Use
Trace 1	Raw data line
Trace 2	Output of U2 pin 4, `dt`.
Command	Open Iris sent to Camera ..., Monitor 1.
Figure A	Camera 9.
Figure B	Same as Figure A, diffrent horizontal resolution.
Figure C	Camera 1.
Figure D	Iris Close, Camera 1.

APPENDIX B

B Burle commands

In testing the TXB-B there were four types of Burle equipment used. These were:

1. Matrices

   A. An older TC-8501 matrix with a capacity of 64(?) input video lines. Most references to this will have an "m" and "o" in their name, it will be called "old" or it will be called "TC-8501".

   B. A new LTC-8801 matrix with a capacity of 256(?) input video lines. Most references to this will have an "m" and "n" in their name, it will be called "new" or it will be called "LTC-8801".

2. Keyboards

   A. An "AutoDome Controller" LTC-5136. May address up to 9,999 cameras. Most references to this will have a "k" in their name, be called an "LTC-5136".

   B. A "Virtual Keyboard" LTC-5138. May address up to 9,999 cameras. Most references to this will have a "v" in their name or be called "LTC-5138". Extended commands that are available from this keyboard are shown in Section B.3, page B-6.

B.1 Common command information

1. Commands are either seven or eight bits long.

2. The first byte is always the number of bytes following the first and always has the most significant bit set as a "first byte" marker.

3. The second and third bytes are the address. This value is "zero based", i.e. `0x00:0x00` = Camera 1. The range is `0x00:0x00` (1) to `0x4E:0x0E` (9999_{10}).

   A. The second byte is the upper half of the address and has a range of from 0 to 127_{10}, i.e. it is a seven bit value since it is not a "first byte".

   B. The third byte is the lower half of the address and has a range of from 0 to 127_{10}, i.e. it is a seven bit value since it is not a "first byte".

4. The fourth byte is the operation code (opcode). Only opcodes 2, 4, 5, 6, 7 and 8 have been observed.

5. The last byte is the arithmetic checksum of all the preceding bytes, including the first byte. Note that the most significant bit is cleared as this is not a "first byte".

[29] `$Header: d:/ecr6171/RCS/bc.inc,v 1.6 2000-11-27 14:22:47-08 Hamilton Exp Hamilton $`

6. In these tables, when a range is given as `0x00`, this indicates that zero is the slowest rate that the event may occur at. Note that movement, etc., may be happening.

7. When hexadecimal values are given for bytes 5, 6 or 7 (Data bytes 1, 2 and 3). These are used to indicate a single bit that is set. When actions are combined, then more that a single bit in a byte may be set. I.e. with an LTC-8801 matrix, if the joy stick in the upper left position. Byte 7 becomes `0x0A` and if the zoom control is also simultaneously turned to the right it would become `0x2A`.

8. In the command information tables:

—	=	Field not present, used to indicate seven byte commands that do not have an eighth byte.
AUX	=	Auxiliary #, "one based", i.e. `0x01` = 1, etc., range is `0x01` (1) to `0x08` (8) using bits 0, 1, 2 and 3.
AZ	=	Azimuth (pan) rate, range is `0x00` (0) to `0x78` (15_{10}) using bits 3, 4, 5 and 6. Usage of bits 0, 1 and 2 is unknown.
EL	=	Elevation (tilt) rate, range is `0x00` (0) to `0x0F` (15_{10}) using bits 0, 1, 2 and 3.
SH	=	Shot #, "one based", i.e. `0x01` = 1, etc. For the TC-8501 matrix the maximum value of a shot (preset) is `0x3F` (63_{10}). For the LTC-8801 matrix the maximum value of a shot (preset) is 128_{10} (`0xFF`, = 127_{10}, `0x00` = 128_{10}).
ZM	=	Zoom rate, range is `0x00` (0) to `0x07` (7) using bits 4, 5 and 6. Usage of bits 0, 1, 2 and 3 is for elevation.

9. Phrases similar to "Lower Zoom" indicate that either the lower part of the Zoom button was pushed, or the lower of two buttons was pushed.

10. Phrases similar to "Zoom Left" indicate that the joystick knob was being turned in the left direction, i.e. counterclockwise.

B.2 Command Information

This section details the contents of bytes 4, 5, 6 and when present 7, of commands from various sources.

Keyboard format commands consist of a block multiple single bytes of data which lasts 7 ms to 8 ms for camera control commands. These repeat at either a 18 ms or a 46 ms rate.

Matrix format commands consist of a single block of data which lasts 2.7 ms or 2.9 ms for camera control commands. These repeat at either a 53.6 ms or a 32.8 ms rate. (See Figure B-1, page B-5.)

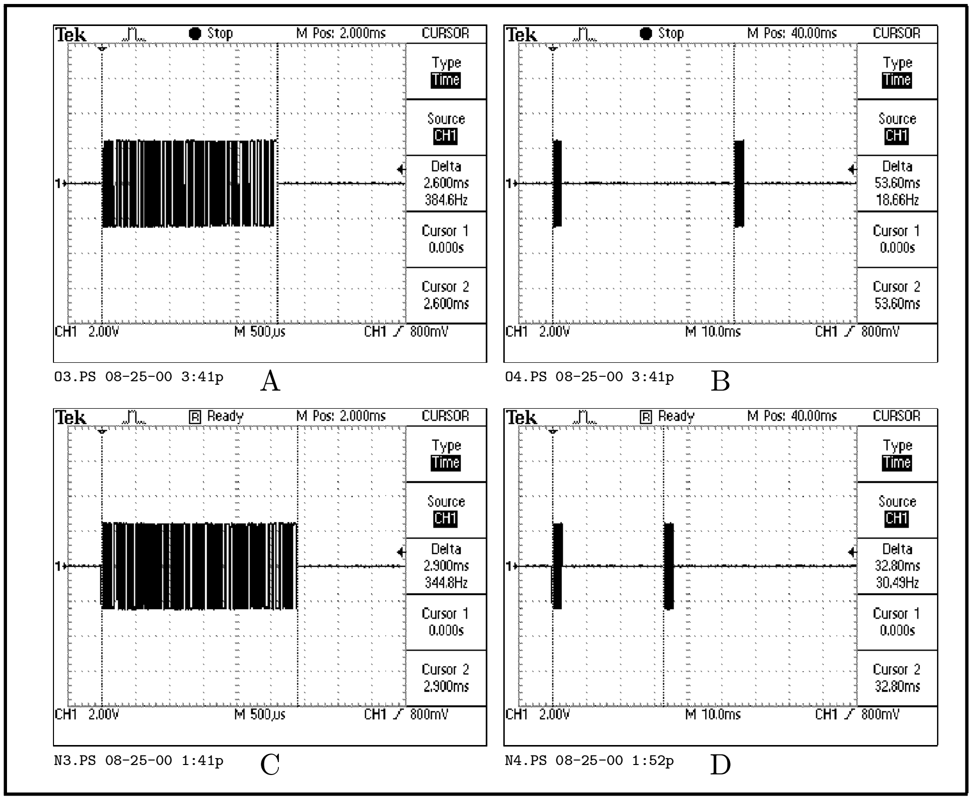
1. The LTC-5136 keyboard is a small camera controller with a variable speed joystick.

2. The LTC-5138 keyboard is software that runs on a PC under Windows 9x.
3. The TC-8501 matrix is an older fixed speed system.
4. The LTC-8801 matrix is a newer variable speed system.

Command Source	Command	4 Opcode	5 Data 1	6 Data 2	7 Data 3
LTC-5136	Upper Focus	0x08	0x00	0x01	0x00
LTC-5138		0x02	0x00	0x02	—
TC-8501		0x06	0x00	0x04	—
LTC-8801		0x08	0x00	0x00	0x40
LTC-5136	Lower Focus	0x08	0x00	0x00	0x40
LTC-5138		0x02	0x02	0x00	—
TC-8501		0x06	0x00	0x08	—
LTC-8801		0x08	0x00	0x01	0x00
LTC-5136	Upper Iris	0x08	0x00	0x04	0x00
LTC-5138		0x02	0x01	0x00	—
TC-8501		0x06	0x00	0x10	—
LTC-8801		0x08	0x00	0x02	0x00
LTC-5136	Lower Iris	0x08	0x00	0x02	0x00
LTC-5138		0x02	0x00	0x01	—
TC-8501		0x06	0x00	0x20	—
LTC-8801		0x08	0x00	0x04	0x00
LTC-5136	Joy Stick Left	0x08	*EL*	*AZ*	0x02
LTC-5138		0x05	*EL*	*AZ*	0x02
TC-8501		0x04	0x00	0x02	—
LTC-8801		0x08	*EL*	*AZ*	0x02
LTC-5136	Joy Stick Right	0x08	*EL*	*AZ*	0x01
LTC-5138		0x05	*EL*	*AZ*	0x01
TC-8501		0x04	0x00	0x01	—
LTC-8801		0x08	*EL*	*AZ*	0x01
LTC-5136	Joy Stick Down	0x08	*EL*	*AZ*	0x04
LTC-5138		0x05	*EL*	*AZ*	0x04
TC-8501		0x04	0x00	0x04	—
LTC-8801		0x08	*EL*	*AZ*	0x04
LTC-5136	Joy Stick Up	0x08	*EL*	*AZ*	0x08
LTC-5138		0x05	*EL*	*AZ*	0x08
TC-8501		0x04	0x00	0x08	—

Continued on the next page.

Continued from the previous page.					
Command Source	Command	4 Opcode	5 Data 1	6 Data 2	7 Data 3
LTC-8801		0x08	*EL*	*AZ*	0x08
LTC-5136	Zoom Right	0x08	*ZM*	0x00	0x20
LTC-5138	Upper Zoom	0x02	0x00	0x04	—
TC-8501	Upper Zoom	0x06	0x00	0x01	—
LTC-8801	Zoom Right	0x08	*ZM*	0x00	0x20
LTC-5136	Zoom Left	0x08	*ZM*	0x00	0x10
LTC-5138	Lower Zoom	0x02	0x04	0x00	—
TC-8501	Lower Zoom	0x06	0x00	0x02	—
LTC-8801	Zoom Left	0x08	*ZM*	0x00	0x10
LTC-5136	Shot	0x07	0x05	*SH*	—
LTC-5138		0x07	0x05	*SH*	—
TC-8501		0x07	0x05	*SH*	—
LTC-8801		0x07	0x05	*SH*	—
LTC-5136	Set	0x07	0x04	*SH*	—
LTC-5138		0x07	0x04	*SH*	—
TC-8501		0x07	0x04	*SH*	—
LTC-8801		0x07	0x04	*SH*	—
LTC-5138	Aux On	0x07	0x01	*AUX*	—
LTC-5138	Aux Off	0x07	0x02	*AUX*	—



O3.PS, O4.PS, N3.PS, N4.PS $RCSfile: typlcmnd.inc,v $

Figure B-1. Format M, Typical matrix command information

Item	Setting/Use
Trace 1	Raw data line
Command	Open Iris sent to Camera 9 (Figs A, B), Camera 137 (Figs C, D).
Figure A	Typical TC-8501 single button push.
Figure B	Repeat rate for a TC-8501 matrix.
Figure C	Typical LTC-8801 single button push.
Figure D	Repeat rate for an LTC-8801 matrix.

[30] `$Header: d:/ecr6171/RCS/typlcmnd.inc,v 1.2 2000-09-20 16:07:40-07 Hamilton Exp Hamilton $`

B.3 Extended Brule Commands from the LTC-5138

B.3.1 By Functional Group

B.3.1.1 General Commands

Command	Option	Opcode	Data 1	Data 2
AutoDome Master Reset	Set	0x07	0x74	0x03 (3_{10})
Camera Reset	Set	0x07	0x01	0x28 (40_{10})
Command Lock	Set	0x07	0x04	0x67 (103_{10})
Command Unlock	Set	0x07	0x04	0x68 (104_{10})
Playback Continous	On	0x07	0x01	0x32 (50_{10})
	Off	0x07	0x02	0x32 (50_{10})
Playback Single	On	0x07	0x01	0x33 (51_{10})
	Off	0x07	0x02	0x33 (51_{10})
Resume Playback	On	0x07	0x01	0x34 (52_{10})
Record	On	0x07	0x01	0x64 (100_{10})
	Off	0x07	0x02	0x64 (100_{10})
Resume Record	On	0x07	0x01	0x65 (101_{10})
	Off	0x07	0x02	0x65 (101_{10})
Erase Record to End	Set	0x07	0x34	0x74 (116_{10})
Software Version	Display	0x07	0x01	0x42 (66_{10})
On-Screen Display	On	0x07	0x01	0x3C (60_{10})
	Off	0x07	0x02	0x3C (60_{10})
On-Screen Display Adjust*	On	0x07	0x01	0x3D (61_{10})
	Off	0x07	0x02	0x3D (61_{10})
Pre-Position Title Set*	On	0x07	0x01	0x3E (62_{10})
	Off	0x07	0x02	0x3E (62_{10})
Zone Title Set*	On	0x07	0x01	0x3F (63_{10})
	Off	0x07	0x02	0x3F (63_{10})
*On screen positioning commands	U	0x04	0x00	0x08 (8_{10})
	R	0x04	0x00	0x01 (1_{10})
	D	0x04	0x00	0x04 (4_{10})
	L	0x04	0x00	0x02 (2_{10})
	Prev/Back On	0x06	0x00	0x08 (8_{10})
	Next/Back Off	0x06	0x00	0x04 (4_{10})
	Clear	0x06	0x00	0x10 (16_{10})

[31]`$Header: d:/ecr6171/RCS/bec.inc,v 1.5 2000-11-27 14:22:47-08 Hamilton Exp Hamilton $`

B.3.1.2 Pan & Tilt Commands

Command	Option	Opcode	Data 1	Data 2
Auto-Scan	ON	0x07	0x01	0x01 (1_{10})
	OFF	0x07	0x02	0x01 (1_{10})
Auto-Pan	ON	0x07	0x01	0x02 (2_{10})
	OFF	0x07	0x02	0x02 (2_{10})
Auto-Pan Left Limit	SET	0x07	0x04	0x65 (101_{10})
	SHOW	0x07	0x05	0x65 (101_{10})
Auto-Pan Right Limit	SET	0x07	0x04	0x66 (102_{10})
	SHOW	0x07	0x05	0x66 (102_{10})
Set Auto Scan/Pan Speed	INCREASE	0x07	0x09	0x0E (14_{10})
		0x07	0x08	0x0E (14_{10})
	DECREASE	0x07	0x0A	0x0E (14_{10})
		0x07	0x08	0x0E (14_{10})
Pre-Position Tour	ON	0x07	0x01	0x08 (8_{10})
	OFF	0x07	0x02	0x08 (8_{10})
Set Tour Period	INCREASE	0x07	0x09	0x0F (15_{10})
		0x07	0x08	0x0F (15_{10})
	DECREASE	0x07	0x0A	0x0F (15_{10})
		0x07	0x08	0x0F (15_{10})
Home position	RECALIBRATE	0x07	0x04	0x6E (110_{10})
	SHOW	0x07	0x05	0x6E (110_{10})
Return Home Upon Inactivity	ON	0x07	0x01	0x09 (9_{10})
	OFF	0x07	0x02	0x09 (9_{10})
Auto-Speed	ENABLE	0x07	0x01	0x10 (16_{10})
	DISABLE	0x07	0x02	0x10 (16_{10})
Auto-Pivot	ENABLE	0x07	0x01	0x12 (18_{10})
	DISABLE	0x07	0x02	0x12 (18_{10})
Pivot Callup	SHOW	0x07	0x05	0x6F (111_{10})

B.3.1.3 Lens Commands

Command	Option	Opcode	Data 1	Data 2
Iris Control	AUTO	0x07	0x01	0x03 (3_{10})
	MANUAL	0x07	0x02	0x03 (3_{10})
Focus Control	AUTO	0x07	0x01	0x04 (4_{10})
	MANUAL	0x07	0x02	0x04 (4_{10})
Auto Iris ALC Adjust	PEAK	0x07	0x09	0x0A (10_{10})
		0x07	0x08	0x0A (10_{10})
	AVERAGE	0x07	0x0A	0x0A (10_{10})
		0x07	0x08	0x0A (10_{10})
Auto Iris Level Adjust	INCREASE	0x07	0x09	0x0B (11_{10})
		0x07	0x08	0x0B (11_{10})
	DECREASE	0x07	0x09	0x0B (11_{10})
		0x07	0x08	0x0B (11_{10})
Auto Focus Activation	ON	0x07	0x01	0x0C (12_{10})
	OFF	0x07	0x02	0x0C (12_{10})
Auto Iris Activation	ON	0x07	0x01	0x0D (13_{10})
	OFF	0x07	0x02	0x0D (13_{10})
Spot Focus	ON	0x07	0x01	0x11 (17_{10})
	OFF	0x07	0x02	0x11 (17_{10})
Zoom Range Limit	ON	0x07	0x01	0x13 (19_{10})
	OFF	0x07	0x02	0x13 (19_{10})
Digital Zoom	ENABLE	0x07	0x01	0x50 (80_{10})
	DISABLE	0x07	0x02	0x50 (80_{10})
Field of View Optimize	ON	0x07	0x01	0x2D (45_{10})
	OFF	0x07	0x02	0x2D (45_{10})

B.3.1.4 Camera Commands

Command	Option	Opcode	Data 1	Data 2
Backlight Compensation	On	0x07	0x01	0x14 (20_{10})
	Off	0x07	0x02	0x14 (20_{10})
High-Light	On	0x07	0x01	0x15 (21_{10})
	Off	0x07	0x02	0x15 (21_{10})
White Balance	Auto	0x07	0x01	0x1E (30_{10})
	Manual	0x07	0x02	0x1E (30_{10})
Fixed White Balance	Indoor	0x07	0x01	0x23 (35_{10})
	Outdoor	0x07	0x02	0x23 (35_{10})
Line Lock Phase Delay Adjust	Increase	0x07	0x09	0x29 (41_{10})
		0x07	0x08	0x29 (41_{10})
	Decrease	0x07	0x0A	0x29 (41_{10})
		0x07	0x08	0x29 (41_{10})
Sync Mode	Line-Lock	0x07	0x01	0x2A (42_{10})
	Internal	0x07	0x02	0x2A (42_{10})
AGC	On	0x07	0x01	0x2B (43_{10})
	Off	0x07	0x02	0x2B (43_{10})
Vertical Aperture Adjust	Increase	0x07	0x09	0x2C (44_{10})
		0x07	0x08	0x2C (44_{10})
	Decrease	0x07	0x0A	0x2C (44_{10})
		0x07	0x08	0x2C (44_{10})

B.3.2 By preset number used

Command	Option	Opcode	Data 1	Data 2
Erase Record to End (*Allows you to erase the remaining sequence of recorded AutoDome commands from the spot where you are at.*)	SET	0x07	0x34	0x74 (116_{10})
AutoDome Master Reset (*Restores factory default settings. (AutoDomes)*)	SET	0x07	0x74	0x03 (3_{10})
Auto-Scan (*AutoDomes: Activates 360 degrees of continuous scan. Receiver Drivers: Activates panning between previously set limit stops.*)	ON	0x07	0x01	0x01 (1_{10})
	OFF	0x07	0x02	0x01 (1_{10})
Auto-Pan (*AutoDomes: Activates panning between previously set limit stops. Receiver Drivers: Activates aux 2 output*)	ON	0x07	0x01	0x02 (2_{10})
	OFF	0x07	0x02	0x02 (2_{10})
Iris Control (*Selects automatic or manual light compensation.*)	AUTO	0x07	0x01	0x03 (3_{10})
	MANUAL	0x07	0x02	0x03 (3_{10})
Focus Control (*Selects automatic or manual focus.*)	AUTO	0x07	0x01	0x04 (4_{10})
	MANUAL	0x07	0x02	0x04 (4_{10})
Pre-Position Tour (*Sequences the AutoDome through selected 'Tour' pre-positions.*)	ON	0x07	0x01	0x08 (8_{10})
	OFF	0x07	0x02	0x08 (8_{10})
Return Home Upon Inactivity (*Returns the AutoDome to its home position following a period of inactivity.*)	ON	0x07	0x01	0x09 (9_{10})
	OFF	0x07	0x02	0x09 (9_{10})

Continued on the next page.

Continued from the previous page.

Command	Option	Opcode	Data 1	Data 2
Auto Iris ALC Adjust (*Sets the auto iris sensitivity for light and dark areas in a scene.*)	PEAK	0x07	0x09	0x0A (10_{10})
		0x07	0x08	0x0A (10_{10})
	AVERAGE	0x07	0x0A	0x0A (10_{10})
		0x07	0x08	0x0A (10_{10})
Auto Iris Level Adjust (*Sets the video output level for the automatic iris.*)	INCREASE	0x07	0x09	0x0B (11_{10})
		0x07	0x08	0x0B (11_{10})
	DECREASE	0x07	0x09	0x0B (11_{10})
		0x07	0x08	0x0B (11_{10})
Auto Focus Activation (*Activates the auto focus when the camera is moved.*)	ON	0x07	0x01	0x0C (12_{10})
	OFF	0x07	0x02	0x0C (12_{10})
Auto Iris Activation (*Activates the auto iris when the camera is moved.*)	ON	0x07	0x01	0x0D (13_{10})
	OFF	0x07	0x02	0x0D (13_{10})
Set Auto Scan/Pan Speed (*Sets auto pan/scan speed.*)	INCREASE	0x07	0x09	0x0E (14_{10})
		0x07	0x08	0x0E (14_{10})
	DECREASE	0x07	0x0A	0x0E (14_{10})
		0x07	0x08	0x0E (14_{10})
Set Tour Period (*Set the duration a pre-position is held during a 'Tour'.*)	INCREASE	0x07	0x09	0x0F (15_{10})
		0x07	0x08	0x0F (15_{10})
	DECREASE	0x07	0x0A	0x0F (15_{10})
		0x07	0x08	0x0F (15_{10})
Auto-Speed (*Provides a gradual acceleration to maximum speed when using a fixed speed pan-tilt controller.*)	ENABLE	0x07	0x01	0x10 (16_{10})
	DISABLE	0x07	0x02	0x10 (16_{10})
Spot Focus (*Limits the auto focus operation to 3 seconds duration after a camera stops moving.*)	ON	0x07	0x01	0x11 (17_{10})

Continued on the next page.

Continued from the previous page.				
Command	Option	Opcode	Data 1	Data 2
	OFF	0x07	0x02	0x11 (17_{10})
Auto-Pivot (*Enables Auto-Pivot feature.*)	ENABLE	0x07	0x01	0x12 (18_{10})
	DISABLE	0x07	0x02	0x12 (18_{10})
Zoom Range Limit (*Limits the maximum wide angle of the zoom lens (AutoDomes with tele-converters only.)*)	ON	0x07	0x01	0x13 (19_{10})
	OFF	0x07	0x02	0x13 (19_{10})
Backlight Compensation (*Enhances operation in high contrast scenes.*)	ON	0x07	0x01	0x14 (20_{10})
	OFF	0x07	0x02	0x14 (20_{10})
High-Light (*Enhances operation in bright light scenes.*)	ON	0x07	0x01	0x15 (21_{10})
	OFF	0x07	0x02	0x15 (21_{10})
White Balance (*Continuously adjusts for the best color reproduction.*)	AUTO	0x07	0x01	0x1E (30_{10})
	MANUAL	0x07	0x02	0x1E (30_{10})
Fixed White Balance (*Sets the white balance for indoor or outdoor applications.*)	INDOOR	0x07	0x01	0x23 (35_{10})
	OUTDOOR	0x07	0x02	0x23 (35_{10})
Camera Reset (*Restores factory default settings.*)	SET	0x07	0x01	0x28 (40_{10})
Line Lock Phase Delay Adjust (*Adjusts camera synchronization for multiple camera systems.*)	INCREASE	0x07	0x09	0x29 (41_{10})
		0x07	0x08	0x29 (41_{10})
	DECREASE	0x07	0x0A	0x29 (41_{10})
		0x07	0x08	0x29 (41_{10})
Sync Mode (*Selects an internal or line-lock camera synchronization.*)	LINE-LOCK	0x07	0x01	0x2A (42_{10})
	INTERNAL	0x07	0x02	0x2A (42_{10})
AGC (*Provides the camera with additional low light sensitivity.*)	ON	0x07	0x01	0x2B (43_{10})
				Continued on the next page.

Continued from the previous page.

Command	Option	Opcode	Data 1	Data 2
	OFF	0x07	0x02	0x2B (43_{10})
Vertical Aperture Adjust (*Adjusts camera vertical detail enhancement.*)	INCREASE	0x07	0x09	0x2C (44_{10})
		0x07	0x08	0x2C (44_{10})
	DECREASE	0x07	0x0A	0x2C (44_{10})
		0x07	0x08	0x2C (44_{10})
Field of View Optimize (*Coordinates the tilt and zoom positions for optimal viewing.*)	ON	0x07	0x01	0x2D (45_{10})
	OFF	0x07	0x02	0x2D (45_{10})
Playback Continous (*Continuously repeats a recorded sequence of AutoDome commands.*)	ON	0x07	0x01	0x32 (50_{10})
	OFF	0x07	0x02	0x32 (50_{10})
Playback Single (*Plays a recorded sequence of AutoDome commands once.*)	ON	0x07	0x01	0x33 (51_{10})
	OFF	0x07	0x02	0x33 (51_{10})
Resume Playback (*Resumes playback of a recorded sequence of AutoDome commands.*)	ON	0x07	0x01	0x34 (52_{10})
On-Screen Display (*Enable on-screen text display.*)	ON	0x07	0x01	0x3C (60_{10})
	OFF	0x07	0x02	0x3C (60_{10})
On-Screen Display Adjust* (*Sets position and brightness of on-screen text display.*)	ON	0x07	0x01	0x3D (61_{10})
	OFF	0x07	0x02	0x3D (61_{10})
Pre-Position Title Set* (*Allows you to program pre-position titles.*)	ON	0x07	0x01	0x3E (62_{10})
	OFF	0x07	0x02	0x3E (62_{10})
Zone Title Set* (*Allows you to program zone titles.*)	ON	0x07	0x01	0x3F (63_{10})
	OFF	0x07	0x02	0x3F (63_{10})

Continued on the next page.

Command	Option	Opcode	Data 1	Data 2
Continued from the previous page.				
Software Version (*Displays the current software revision.*)	DISPLAY	0x07	0x01	0x42 (66_{10})
Digital Zoom (*Enables/Disables digital zoom feature.*)	ENABLE	0x07	0x01	0x50 (80_{10})
	DISABLE	0x07	0x02	0x50 (80_{10})
Record (*Stores approximately two minutes of recorded sequence of AutoDome commands.*)	ON	0x07	0x01	0x64 (100_{10})
	OFF	0x07	0x02	0x64 (100_{10})
Resume Record (*Allows you to add to an existing record of AutoDome commands.*)	ON	0x07	0x01	0x65 (101_{10})
	OFF	0x07	0x02	0x65 (101_{10})
Auto-Pan Left Limit (*Sets left pan limit.*)	SET	0x07	0x04	0x65 (101_{10})
	SHOW	0x07	0x05	0x65 (101_{10})
Auto-Pan Right Limit (*Sets right pan limit.*)	SET	0x07	0x04	0x66 (102_{10})
	SHOW	0x07	0x05	0x66 (102_{10})
Command Lock (*Limits Keyboard programming capabilities.*)	SET	0x07	0x04	0x67 (103_{10})
Command Unlock (*Restores full keyboard programming capabilities.*)	SET	0x07	0x04	0x68 (104_{10})
Home position (*Recalibrates an AutoDome and returns it to the home position.*)	RECALIBRATE	0x07	0x04	0x6E (110_{10})
	SHOW	0x07	0x05	0x6E (110_{10})
Pivot Callup (*Executes a single auto pivot.*)	SHOW	0x07	0x05	0x6F (111_{10})
*On screen positioning commands	U	0x04	0x00	0x08 (8_{10})
	R	0x04	0x00	0x01 (1_{10})
	D	0x04	0x00	0x04 (4_{10})
	L	0x04	0x00	0x02 (2_{10})

Continued on the next page.

Continued from the previous page.				
Command	Option	Opcode	Data 1	Data 2
	PREV/BACK On	0x06	0x00	0x08 (8_{10})
	NEXT/BACK Off	0x06	0x00	0x04 (4_{10})
	CLEAR	0x06	0x00	0x10 (16_{10})

APPENDIX C

C Burle equipment available for testing

C.1 "New" Burle Matrix System

Model	Quantity	Name
LTC-8801/60	1	Matrix card cage
TC8805	1	Matrix power supply
LTC-8553/00	1	Variable speed keyboard
TC8821VIM	1	32 channel video input card
TC8834VOM	1	4 channel video output card
TC8810A	1	CPU card
LTC-8059/00	1	PC based support software

C.2 "Old" Burle Matrix System

Model	Quantity	Name
TC-8501	1	Matrix card cage
TC-8505	1	Matrix power supply
TC-8550	1	Keyboard
TC-8520VIM	8	8 channel video input card
TC-8332VOM	4	4 channel video output card
???	1	"Microcomputer Adapter" CPU card
TC-8568	1	Code Distribution Unit

C.3 "KBD-300" equivalent hardware

Model	Quantity	Name
LTC-5136/50	1	Autodome Controller, Philips (PAL)
LTC-5138/50	1	Virtual Keyboard

1. 2 small black boxes ($2\frac{1}{4}'' \times 4\frac{1}{4}'' \times 1\frac{1}{4}''$) marked "`303-2746-501`" with screw terminals on one end and a power connector and RJ-11 connector on the other end.
2. 2 TC220PS power supplies, with 240 V AC in and 15 V DC out.
3. 2 RJ-11 cables "silver" wired with 6 conductors.

   "Silver" wired indicates that wireing is like this when each RJ-11 connector is viewed from the bottom:

[32] `$Header: d:/ecr6171/RCS/bparts.inc,v 1.7 2000-11-27 14:22:47-08 Hamilton Exp Hamilton $`

One end	The other end
BL	BL
YE	YE
GR	GR
RE	RE
BK	BK
WH	WH

The alternate wiring “black” is like this:

One end	The other end
BL	WH
YE	BK
GR	RE
RE	GR
BK	YE
WH	BL

I have given up on trying to call one of these configurations as “streight through” and the other as “flipped” because it depends on how the ends are viewed.

4. 1 DB-9-F to RJ-11 adapter. 2 = GN, 3 = RE and 5 = BK on the DB-9 going to pins 3 (GN), 4 (RE) and 5 (BK) (when viewed from the front) on the RJ-11. (All other pins are either empty or cut off.)

APPENDIX D

D American Dynamics equipment available for testing

D.1 Unit from "the line" Matrix System

Model	Quantity	Name
AD2150D16-5	1	Matrix 32 in, 5 out
AD2078	1	Keyboard
HP0082	1	Connector block
TP8082	1	Power Supply

APPENDIX E

E Burle/Philips protocol

Note

Most of this information is copied from a Philips application note about their protocol, dated 1997.

The original title of the application note was "Receiver/Driver and AutoDome Control Code Protocol", with a sub-title of "TC8560 Series, TC700 Series".

E.1 Introduction

The following paragraphs document the control code data format used by Philips CSS control systems to communicate with OnSite Receiver/Drivers and AutoDome cameras. This document does not cover other forms of status type data which is also generated by certain control system.

Control code data is sent in message packets. For certain functions, commands must be sent repetitively at a rate of 20 Hz to maintain a smooth operational response of the device being driven by the receiver/driver.

Default RS-232 parameters for receiver/drivers are:

1. 2400 or 9600 baud, 1 stop bit, 8 data bits, no parity, and no handshake.

The format for the Receiver/Driver control message packet is as follows:

1. Length Byte
2. Receiver/Driver High Order Address Byte
3. Receiver/Driver Low Order Address Byte
4. Opcode Byte
5. Data Byte 1
6. Data Byte 2
7. Data Byte 3 (used with Opcodes 5 and 8 only)
8. Checksum Byte

Each packet begins with a **Length byte** that specifies the number of bytes in the remainder of the packet (the length byte itself is not included in this number). The most significant bit (MSB) of the length byte is always set to 1. The MSB must be zero for ALL other bytes of the message packet. For the functions below, only control commands with lengths of 6 and 7 bytes are used, therefore, the length byte (with the MSB set) will be either 86 hex or 87 hex.

The message packet contains a **Receiver/Driver** address number encoded using a 14 bit binary. (Including an address value that permits the data to be broadcast to all receiver/driver sites but only the site sent to a matching address will respond.) This address number is sent using 2 bytes of the message packet. The binary value corresponds to the logical camera number of the camera site being controlled minus 1. For example, camera 1 should be encoded with all 14 data bits reset to zero. Camera number has its Least Significant Bit (LSB) set to 1 and all other its set to zero. The High Order Address byte consists of the upper 7 bits of the 14 bit binary camera number. The Low Order Address Byte consists of the lower 7 bits

[33] `$Header: d:/ecr6171/RCS/bproto.inc,v 1.4 2000-11-27 14:22:47-08 Hamilton Exp Hamilton $`

of the 14 bit binary camera number. In all cases, the MSB of each byte is not counted as part of the address number and must always bee reset to zero so it will not be confused with a Length byte. Since the use of a 14 bit binary number provides a camera number range from 1 to 16384, the corresponding Receiver/Driver Address data bytes would take the from of 0000 hex to 7F7F hex.

Address Byte examples are shown below:

Camera Number	Value to Be Encoded	14-bit Binary Value	High Order Byte (Hex)	Low Order Byte (Hex)
1	0	0000000 0000000	00	00
2	1	0000000 0000001	00	01
256	255	0000001 1111111	01	7F
257	256	0000010 0000000	02	00
500	499	0000011 1110011	03	73
512	511	0000011 1111111	03	7F
513	512	0000100 0000000	04	00
1024	1023	0000111 1111111	07	7F
5000	4999	0100111 0000111	27	07
9999	9998	1001110 0001110	4E	0E
16384	16383	1111111 1111111	7F	7F

The **Opcode** byte along with the 2 or 3 data bytes determine the actual function to be executed by the receiver/driver. Currently 7 different opcodes are supported. Opcodes 2, 3, 4, 6 and 7 are associated with 2 data bytes while opcodes 5 and 8 are associated with 3 data bytes.

The format of three **Data Bytes** depend on the opcode. THe opcode functions and their corresponding data formats are shown in Table 1. In all cases, a 1 written into a bit position initiates the specified action. If conflicting bits are set (e.g. Pan Left and Pan Right), the action is undefined, but the receiver/driver will resolve the conflict with no damage. If the receiver/driver receives a command while still processing a previous command, the old command will be aborted, and the new one executed.

The **Checksum** is computed as the sum of all previous bytes (including the length byte) in the message, modulo 128 (logical AND sum with 7F hex).

E.2 Opcode Descriptions

E.2.1 Opcode 2

Is used to activate fixed speed pan/tilt/zoom functions for an indefinite period. A logic one will activate the indicated function, which will remain active until explicitly turned off. The functions can be turned off by a command with a different opcode, or by opcode 2 with zero in the associated data bit position.

E.2.2 Opcode 3

Is revered to as a "poor man's preposition" because it can be used to operate any pan/tile/zoom (even those without preposition capability) to approximate positions by moving for a specified time in the desired direction. The duration of the function is specified using a 6-bit data value where the time is specified in units of half-seconds. This provides a time range of 1/2 second duration (all bits reset to zero) to 32 seconds (all bits set to 1). Note that the actual duration of the function may only approximate the specified time due to conditions at the receiver/driver site.

E.3 Opcode 4

Activates pan/tilt/zoom functions at a fixed speed determined by the Receiver/Driver or AutoDome. This opcode causes the specified function to be activated for at least 50 ms, so the command must be issued at a frequency of not less than 20 Hz for smooth operation.

E.4 Opcode 5

Is used to activate variable speed functions for an indefinite period. A logic one will activate the indicated function, which will remain active until explicitly turned off. This function can be turned off by a command with a different opcode, or by opcode 5 with a zero in the associated data bit position.

E.5 Opcode 6

Activates iris/focus/zoom function at a fixed speed determined by the Receiver/Driver or AutoDome. This opcode causes the specified function to be activated for at least 50 ms, so the command must be issued at a frequency of not less than 20 Hz for smooth operation.

E.6 Opcode 7

Activates preposition or auxiliary functions as shown in Table 2. The numeric data consists of a 10 bit binary number. The upper 3 bits of this 10 bit number along with the desired function is sent as Data Byte 1. The lower 7 bits of the 10 bit number is sent as Data Byte 2. The Auxiliary On and Off commands are sometimes issued repetitively to control level adjustment type functions. For those functions that use repetitive issuance of the command, the repetition rate is 20 Hz.

E.7 Opcode 8

Activates pan, tilt, zoom, focus and iris functions. It provides for variable speed control over pan/tilt/-zoom functions. The pan and tilt functions require a speed value of 0 to 15 (0 to F hex); 0 is the slowest speed and 15 is the fastest speed. The zoom function requires a speed value of 0 to 7; 0 is the slowest speed and 7 is the fastest speed. This opcode causes the specified function to be activated for at least 50 ms, so the command must be insured at a frequency of not less than 20 Hz for smooth operation.

Table 1, Coding of Data Bytes

OPCODE	DATA BYTE	BIT							
		7	6	5	4	3	2	1	0
2 (Indefinite Activation)	1	0	0	0	Pan L	Tilt U	Zoom O	Foc N	Iris 0
	2	0	0	0	Pan R	Tilt D	Zoom I	Foc F	Iris C
3 (PoorMan's Prepos)	1	0	Time (value-1 = number of half seconds)						Foc F
	2	0	Foc N	Zoom I	Zoom O	Tilt U	Tilt D	Pan L	Pan R
4 (Pan/Tilt/-Zoom)	1	0	0	0	0	0	0	0	Foc F
	2	0	Foc N	Zoom I	Zoom O	Tilt U	Tilt D	Pan L	Pan R
5 (Indef Act Var Speed Pan/Tilt/-Lens)	1	0	Zoom Speed			Tilt Speed			
	2	0	Pan Speed				Iris O	Iris C	Foc F
	3	0	Foc N	Zoom I	Zoom O	Tilt U	Tilt D	Pan L	Pan R
6 (Lens Control)	1	0	0	0	0	0	0	0	0
	2	0	0	Iris O	Iris C	Foc F	Foc N	Zoom I	Zoom O
7 (Aux/Prepos)	1	0	Numeric Data MSB's			Function Code (See table 2)			
	2	0	Numeric Data LSB's						
8 (Variable Speed Pan/Tilt/-Lens)	1	0	Zoom Speed			Tilt Speed			
	2	0	Pan Speed				Iris O	Iris C	Foc F
	3	0	Foc N	Zoom I	Zoom O	Tilt U	Tilt D	Pan L	Pan R

Abbreviations used in Table 1:

Foc N = Focus Near	Tilt U = Tilt Up
Foc F = Focus Far	Tilt D = Tilt Down
Iris C = Iris Close (Iris Darker)	Zoom I = Zoom In
Iris O = Iris Open (Iris Brighter)	Zoom O = Zoom Out
Pan R = Pan Right	
Pan L = Pan Left	

Opcode 4 note: Although bits 1, 2, and 3 for Data Byte 1 are shown as zeros in Table 1 above, certain existing Philips CSS controllers generate commands where these bits are set to 1. Either format will be accepted by the receiver/driver device and it has no affect on the desired function.

Table 2: Opcode 7 Functions

Function Code	Function	Numeric Data
0000	Reserved	Undefined
0001	Auxiliary ON	Auxiliary #
0010	Auxiliary OFF	Auxiliary #
0011	Auxiliary TOGGLE	Auxiliary #
0100	Pre-position SET	Pre-position #
0101	Pre-position SHOW	Pre-position #
0110	*<not specified>*	—
0111	Reserved	Undefined
1000*	Cancel Latching Aux	Don't Care
1001*	Latching AUX ON	Auxiliary #
1010*	Latching AUX OFF	Auxiliary #
1011	Reserved	Undefined
1100	Reserved	Undefined
1101	Reserved	Undefined
1110	Reserved	Undefined
1111	Reserved	Undefined

* Latching auxiliary ON and OFF functions are used to activate the auxiliary function until explicitly deactivated using the Cancel Latching Aux command.

E.8 Sample Message Packets

E.8.1 Camera Address 1, Pan Left

	Bit Contents										
	7	6	5	4	3	2	1	0	Binary	Decimal	Hex
Length Byte	1	0	0	0	0	1	1	0	1000 0110	134	86
High Order Address Byte	0	0	0	0	0	0	0	0	0000 0000	0	00
Low Order Address Byte	0	0	0	0	0	0	0	0	0000 0000	0	00
Opcode Byte	0	0	0	0	0	1	0	0	0000 0100	4	04
Data Byte 1	0	0	0	0	0	0	0	0	0000 0000	0	00
Data Byte 2	0	0	0	0	0	0	1	0	0000 0010	2	02
Checksum Byte	0	0	0	0	1	1	0	0	0000 1100	12	0C

E.8.2 Camera Address 62, Pan Right and Tilt Down at Medium Speed

	Bit Contents										
	7	6	5	4	3	2	1	0	Binary	Decimal	Hex
Length Byte	1	0	0	0	0	1	1	1	1000 0111	135	87
High Order Address Byte	0	0	0	0	0	0	0	0	0000 0000	0	00
Low Order Address Byte	0	0	1	1	1	1	0	1	0011 1101	61	3D
Opcode Byte	0	0	0	0	1	0	0	0	0000 1000	8	08
Data Byte 1	0	0	0	0	1	0	0	0	0000 1000	8	08
Data Byte 2	0	1	0	0	0	0	0	0	0100 0000	64	40
Data Byte 3	0	0	0	0	0	1	0	1	0000 0101	5	05
Checksum Byte	0	0	0	0	0	0	0	0	0000 1100	25	19

APPENDIX F

F D Protocol

This section covers the “D” Protocol. This protocol is used between matrix switching systems and receiver/drivers.

F.1 Physical Layer

Receivers use serial RS-422/RS-485 as the basic communications mechanism.

Connection mechanisms vary and are dependent on the equipment. Please refer to the manual for the specific equipment concerned.

F.2 Byte Format

The data format for a character is 1 start bit, 8 data bits, and 1 stop bit, no parity.

F.3 Message Format

Each message consists of 7 bytes as follows:

Byte	Function
1	Synchronization character
2	Receiver address
3	Command byte 1
4	Command byte 2
5	Data byte 1
6	Data byte 2
7	Checksum

F.4 Messages

F.4.1 Byte 1, Sync byte

The synchronization character is always `0xFF` (8 data bits, all ones).

F.4.2 Byte 2, Address byte

The receiver address is any 8 bit binary value from `0x01` through `0xFF`. Note that address zero (`0x00`) is invalid.

[34]`$Header: d:/ecr6171/RCS/dproto.inc,v 1.5 2000-11-27 14:22:48-08 Hamilton Exp Hamilton $`

F.4.3 Bytes 3, 4, 5, and 6 as PTZ commands

The commands are divided up into two groups. The first group are commands that control the pan, tilt, and zoom functions (are called PTZ functions).

F.4.3.1 Byte 3, Command 1

Bit	Function
7	Sense
6	Reserved, 0
5	Reserved, 0
4	Auto scan, manual scan
3	Camera on, camera off
2	Iris close
1	Iris Open
0	Focus Near

The sense bit (command 1, bit 7) indicates the meaning of bits 3 and 4. If the sense bit is on, and bits 3 and 4 are on, the command will enable auto-scan and turn the camera on. If the sense bit is off and bits 3 and 4 are on the command will enable manual scan and turn the camera off. Of course, it either bit 3 or 4 are off then no action will be taken for those features. And both bits do not need to be turned on simultaneously.

The reserved bits, 5 and 6, should be set to 0.

F.4.3.2 Byte 4, Command 2

Bit	Function
7	Focus Far
6	Zoom wide
5	Zoom telephoto
4	Tilt down
3	Tilt up
2	Pan left
1	Pan right
0	Always 0 in the PTZ commands

F.4.3.3 Byte 5, Data 1

Contains the pan speed. Legal values are `0x00` (slow) to `0x3F` (fast) and `0x40` turbo. Turbo speed is the maximum speed the unit can run at and is considered separately because the step from high speed to turbo speed is not smooth. That is, going from one speed to the next usually looks smooth and will provide for smooth motion with the exception of going into and out of turbo speed.

F.4.3.4 Byte 6, Data 2

Contains the tilt speed. Legal values are `0x00` (slow) to `0x3F` (fast).

F.4.3.5 Usage summary of bytes 3, 4, 5 and 6

	7	6	5	4	3	2	1	0
Command 1 Alt cmnd 1	Sense	0	0	Auto/ Manual	On/ Off	Close	Open	Near
Command 2	Far	Wide	Telephoto	Down	Up	Left	Right	Always 0
Data 1	Pan speed, 0 → `0x3F`, with `0x40` = Turbo speed							
Data 2	Tilt speed, 0 → `0x3F`							

F.4.4 Bytes 3, 4, 5, and 6 as Extended commands

The extended commands (non-PTZ) always have bit 0 of byte 4 set to 1. The values in the quick reference table, below, are all in hexadecimal.

Command	Byte 3	Byte 4	Byte 5	Byte 6	Spectra?
Set preset	0x00	0x03	0x00	0x01 to 0x20	Y
Clear preset	0x00	0x05	0x00	0x01 to 0x20	Y
Go to preset	0x00	0x07	0x00	0x01 to 0x20	Y
Flip	0x00	0x07	0x00	0x21	Y
Zero pan position	0x00	0x07	0x00	0x22	Y
Set preset, part 2	0x00	0x03	0x00	0x23 to 0x42	Y
Clear preset, part 2	0x00	0x05	0x00	0x23 to 0x42	Y
Go to preset, part 2	0x00	0x07	0x00	0x23 to 0x42	Y
Set aux	0x00	0x09	0x00	0x01-0x08	Y
Clear aux	0x00	0x0B	0x00	0x01-0x08	Y
Remote reset	0x00	0x0F	0x00	0x00	Y
Set zone start	0x00	0x11	0x00	0x01-0x08	Y
Set zone end	0x00	0x13	0x00	0x01-0x08	Y
Write character to screen	0x00	0x15	0x00-0x27	ASCII char	Y

Continued on the next page.

Continued from the previous page.

Command	Byte 3	Byte 4	Byte 5	Byte 6	Spectra?
Clear screen	0x00	0x17	0x00	0x00	Y
Alarm acknowledge	0x00	0x19	0x00	Alarm #, 1 based	Y
Zone scan on	0x00	0x1B	0x00	0x00	Y
Zone scan off	0x00	0x1D	0x00	0x00	Y
Pattern start	0x00	0x1F	0x00	0x00-0x02	Y
Pattern stop	0x00	0x21	0x00	0x00-0x02	Y
Run pattern	0x00	0x23	0x00	0x00-0x02	Y
Zoom lens speed	0x00	0x25	0x00	0x00-0x03 (slow - fast)	Y
Focus lens speed	0x00	0x27	0x00	0x00-0x03 (slow - fast)	Y
Reset camera to factory defaults	0x00	0x29	0x00	0x00	N
Focus auto/off/on	0x00	0x2B	0x00	0x00-0x02	N
Iris auto/off/on	0x00	0x2D	0x00	0x00-0x02	N
AGC auto/off/on	0x00	0x2F	0x00	0x00-0x02	N
Backlight compensation off/on	0x00	0x31	0x00	0x01-0x02	N
White balance auto/off	0x00	0x33	0x00	0x01-0x02	N
Enable device phase delay mode	0x00	0x35	0x00	0x00	N
Set shutter speed	0x00	0x37	any	any	N
Adjust line lock phase delay	0x00-0x01	0x39	any	any	N
Adjust white balance (R-B)	0x00-0x01	0x3B	any	any	N
Adjust white balance (M-G)	0x00-0x01	0x3D	any	any	N
Adjust gain	0x00-0x01	0x3F	any	any	N
Adjust auto iris level	0x00-0x01	0x41	any	any	N
Adjust auto-iris peak level	0x00-0x01	0x43	any	any	N
Query	0x00	0x45	any	any	Y

Continued on the next page.

Continued from the previous page.					
Command	Byte 3	Byte 4	Byte 5	Byte 6	Spectra?

Note that Query may only be used in a point to point application. A device being queried will respond to any address. Therefore, if more than one device hears this command, they will all simultaneously respond and the replies will be messed up.

The response to this style of command consists of four bytes. Byte 1 is the sync byte (`0xFF`), byte 2 is the receiver address, byte 3 is alarm information, and byte 4 is the check sum.

The alarm information consists of the values `0xB0` through `0xB7` representing alarms 1 through 8 respectively.

F.4.5 Byte 7, Checksum

Contains the 8 bit sum of bytes 2 through 6.

F.5 Creating Labels

Many devices have the ability to display labels on the video. Labels that identify the preset or zone being scanned are common. There is a special technique to establish a label that is associated with either a preset or a zone. First, send the label to the receiver/driver using the "Write char. to Screen" command. After the label is up on the screen set the preset/zone. That will establish the label and associate it with the preset/zone.

F.6 Examples

Command to send	Command						
	Sync	Add	Cmnd1	Cmnd2	Data1 Pan	Data2 Tilt	Cksm
Receiver 1, Camera on	0xFF	0x01	0x88	0x00	0x00	0x00	0x89
Receiver 1, Camera off	0xFF	0x01	0x08	0x00	0x00	0x00	0x09
Receiver 2, Left $\frac{1}{2}$ speed	0xFF	0x02	0x00	0x04	0x00	0x20	0x26
Receiver 2, Stop	0xFF	0x02	0x00	0x00	0x00	0x00	0x02
Receiver 10, Camera on, Focus Far, Left, Turbo Speed	0xFF	0x0A	0x88	0x82	0x40	0x00	0x54

F.6.1 Calculating a checksum

Using the last of the examples, we get:

	0x0A	0000 1010	
	0x88	1000 1000	
1	Subtotal	1001 0010	0x92
	0x82	1000 0020	
2	Subtotal	0001 0100	0x14
	0x40	0400 0000	
3	Subtotal	0101 0100	0x54
	0x00	0000 0000	
4	Total	0101 0100	0x54

Note in step 2 that when adding modulo 256 results in having the high order bit "disappearing".

F.7 Responses

Devices that utilize the D protocol may generate a response.

F.7.1 General Responses

The general response to a received command consist of information about the current alarms present at the remote unit. The format of the response message is as follows:

Sync	Address	Alarm information	Checksum

The alarm information is a bit encoded byte which indicates which bytes are active. The format of the alarm information byte is:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Alarm 8	Alarm 7	Alarm 6	Alarm 5	Alarm 4	Alarm 3	Alarm 2	Alarm 1

The address is the address of the unit responding.

Active alarms are indicated by having the appropriate bit set (`1`). Inactive alarms have their bits reset (`0`).

The checksum is the sum, modulo 256, of the address and alarm information bytes.

F.7.2 Query response

The response to a query command is:

Sync	Address	Part Number (15 bytes)	Checksum

The address is the address of the unit responding.

The part number is the ASCII text string containing the program number (`PRG...`) of the device being queried.

The checksum is the sum, modulo 256, of the original query command's checksum, the address and the 15-byte part number.

F.8 Command Descriptions

Most of the commands are straight forward. The following explains some of the commands.

1. Adjust auto-iris level `0x46`

   If byte 1 is 0, byte 3 and byte 4 are the high and low bytes respectively of an unsigned 16-bit number which the new level. If byte 1 is 1, byte 3 and byte 4 are the high and low bytes respectively of a signed 16-bit number which is the change to the current level. The level is limited internally to the range from 0 to 255 (FF hex). If an attempt is made to set or change the level to a value outside this range, the gain will be set to the appropriate end of the range.

2. Adjust auto-iris peak value `0x43`

   If byte 1 is 0, byte 3 and byte 4 are the high and low bytes respectively of an unsigned 16-bit number which the new peak value. If byte 1 is 1, byte 3 and byte 4 are the high and low bytes respectively of a signed 16- bit number which is the change to the current peak. The peak is limited internally to the range from 0 to 255 (FF hex). If an attempt is made to set or change the peak to a value outside this range, the peak will be set to the appropriate end of the range.

3. Adjust gain `0x3F`

   If byte 1 is 0, byte 3 and byte 4 are the high and low bytes respectively of an unsigned 16-bit number which is the new gain. If byte 1 is 1, byte 3 and byte 4 are the high and low bytes respectively of a signed 16-bit number which is the change to the current gain. The gain is limited internally to the range from 0 to 448 (1C0 hex). If an attempt is made to set or change the gain to a value outside this range, the gain will be set to the appropriate end of the range.

4. Adjust line lock phase delay `0x39`

   If byte 1 is 0, byte 3 and byte 4 are the high and low bytes respectively of an unsigned 16-bit number which is the new phase delay. If byte 1 is 1, byte 3 and byte 4 are the high and low bytes respectively of a signed 16-bit number which is the change to the current phase delay. The phase delay is the delay between the zero crossing of the AC power waveform and the line lock signal sent to the camera. It is in units of 1.085 microseconds. The phase delay is limited internally to the range from 0 to 32767. If an attempt is made to set or change the delay to a value outside this range, the delay will be set to the appropriate end of the range. This command disables device phase delay mode.

5. Adjust white balance (M-G) `0x3D`

   If byte 1 is 0, byte 3 and byte 4 are the high and low bytes respectively of an unsigned 16-bit number which is the new magenta-green white balance value. If byte 1 is 1, byte 3 and byte 4 are the high and low bytes respectively of a signed 16-bit number which is the change to the current magenta-green white balance value. The balance value is limited internally to the range from 192 (C0 hex) to 768 (300 hex). If an attempt is made to set or change the balance to a value outside this range, the balance will be set to the appropriate end of the range. This command turns off auto white balance.

6. Adjust white balance (R-B) `0x3B`

   If byte 1 is 0, byte 3 and byte 4 are the high and low bytes respectively of an unsigned 16-bit number which is the new red-blue white balance value. If byte 1 is 1, byte 3 and byte 4 are the high and low bytes respectively of a signed 16-bit number which is the change to the current red-blue white balance value. The balance value is limited internally to the range from 192 (C0 hex) to 768 (300 hex). If an attempt is made to set or change the balance to a value outside this range, the balance will be set to the appropriate end of the range. This command turns off auto white balance.

7. AGC auto/on/off `0x2F`

   If byte 4 is 0 the device automatically controls whether AGC (automatic gain control) is on or off (default), If byte 4 is 1, AGC is turned off (manual gain). If byte 4 is 2, AGC is turned on. Other values are ignored. Sending an adjust gain command turns AGC off.

8. Auto focus auto/on/off `0x2B`

   If byte 4 is 0 the device automatically controls whether auto focus is on or off (default). If byte 4 is 1, auto focus is turned off. If byte 4 is 2, auto focus is turned on. Other values are ignored.

9. Auto iris auto/on/off `0x2D`

   If byte 4 is 0 the device automatically controls whether auto iris is on or off (default). If byte 4 is 1, auto iris is turned off. If byte 4 is 2, auto iris is turned on. Other values are ignored.

10. Auto white balance on/off `0x33`

    If byte 4 is 1, auto white balance is turned on (default). If byte 4 is 2, auto white balance is turned off. Other values are ignored. Sending an adjust white balance command turns auto white balance off.

11. Backlight compensation on/off `0x31`

    If byte 4 is 1, backlight compensation is turned off (default). If byte 4 is 2, backlight compensation is turned on. Other values are ignored.

12. Enable device phase delay mode `0x36`

    When device phase delay is set, the phase delay is set by the device (there may be a manual adjustment). Sending an adjust line lock phase delay command will disable device phase delay mode.

13. Reset camera to defaults `0x29`

    Resets the camera to its default condition, except that the current phase delay is not changed.

14. Set shutter speed `0x37`

    Byte 3 and byte 4 are the high and low bytes respectively of 1 divided by the shutter speed. The shutter speed is limited internally to the range from 1/60 second (NTSC) or 1/50 second (PAL) to 1/30000 second, corresponding to a sent number range from 60 (or 50) to 30000. If the sent number is, 0 the shutter speed is reset to its default value (1/60 or 1/50 second). If the sent number is 1, the shutter speed is moved to the next faster speed in the shutter speed table (below). If the sent number is 2, the shutter speed is set to the next slower speed in the table.

    Valid values are in the range 60-30000.

    Sending the value 1 will increment the shutter speed to the next higher value in the shutter speed table. Sending a 2 will decrement to the next lower value in the shutter speed table.

Shutter Speeds	
$\frac{1}{60}$ NTSC	$\frac{1}{1000}$
$\frac{1}{50}$ PAL	$\frac{1}{2000}$
$\frac{1}{100}$	$\frac{1}{4000}$
$\frac{1}{120}$	$\frac{1}{10000}$
$\frac{1}{250}$	$\frac{1}{30000}$
$\frac{1}{500}$	—

APPENDIX G

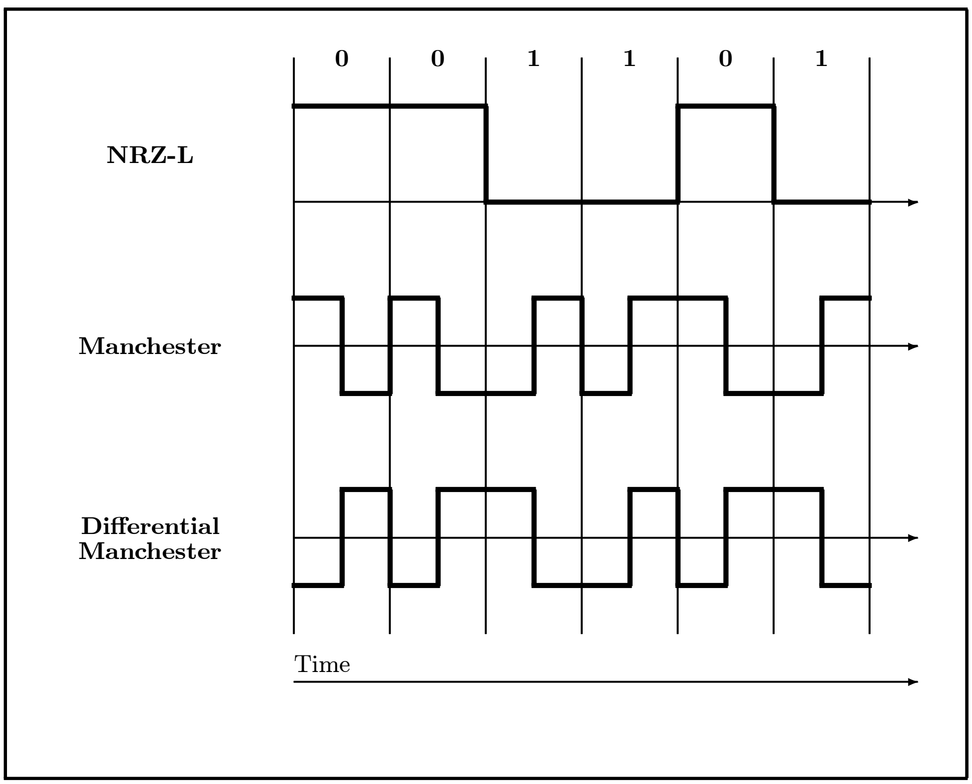
G Manchester Data Encoding

Manchester encoding[36] is characterized by being "self clocking". This is done by having a data transition at least once per bit period. There are two common ways to implement a Manchester coding system. One is called Manchester and the other is called Differential Manchester.

With Manchester encoding there is a transition of the data at the middle of each bit period. The mid-bit transition serves as a clocking mechanism and also as data: A high-to-low transition represents a zero (0), and a low-to-high transition represents a one (1).

With Differential Manchester, the mid-bit transition is used only to provide clocking. The encoding of a zero (0) is represented by the presence of a transition at the beginning of a bit period, and a one (1) is represented by the absence of a transition at the beginning of a bit period.

Differential Manchester has the added advantage of employing differential encoding. In differential encoding, the signal is decoded by comparing the polarity of one signal element with its preceding element, rather than determine the absolute value of a signal element. One benefit of this scheme is that it may be more reliable to detect a transition in the presence of noise than to compare a value to a threshold. Another benefit is that with a complex transmission layout, it is easy to lose the sense of the polarity of the signal. For example, on a multipoint twisted-pair line, if the leads from an attached device to the twisted pair are accidentally inverted, all ones and zeros for non-differential encoding will be inverted and the sense of the message will be lost. This does not happen with differential encoding.



$RCSfile: biphase.inc,v $

[35] `$Header: d:/ecr6171/RCS/manchest.inc,v 1.3 2000-10-13 16:22:25-07 Hamilton Exp Hamilton $`

[36] This information was copied from: `http://www.gre.ac.uk/ cm34/teaching/icn99/itcnlec06.html`.

This page intentionally left blank

APPENDIX H

Index