ECR6171

9 June 2000

W.E.Hamilton

1 Statement of problem to be solved by ECR6171

ECR6171 is an attempt to get Pelco's TXB-AB translator units respond to addresses greater than 128 on Burle type CCTV systems.

Note

> There is some difficulty in the nomenclature applied to the Burle protocol. In some places its called the "AD" protocol and others the "Burle" protocol. For convenience I am going to call it the Burle protocol in this note.

I have a short e-mail from John Ellenberger, dated 08JUN00, which indicates that there is a potential order for 300 Spectras that will run on a Burle controlled system. In the e-mail he mentions that there is a limit of "**168** devices" that the TXB-AB will support. When I went into Bob Herold's office to talk to him about this, he had a handwritten note (an original copy of ECR6171) on his desk that listed the problem as "**128** devices". This smaller limit seems to be logical as it is power of two (which is usually where digital decodes break down) and was also mentioned by Chuck Harmon, in technical manuals, who seems to have been in a meeting about the problem that occurred earlier (I don't know when the meeting was but according to him there were "all the project managers" present). (The ECR indicates that this problem is also listed in PAR 2000-024, which I don't have a copy of yet.)

In the technical manual for the TXB-AB, there is no mention of a 128 limit, but there is a suggestion that there might be an address limit of 1,024 (or is this a limit of 64? Its not clear.). (CM1492M-C (4/00) page 9) The technical manual is not very specific as to how the Burle protocol works and what its limitations are. In the next section I will attempt to describe how it may work.

2 Analysis of the Burle protocol

Philips has a copy of the description of their protocol available on the web. In the e-mail that I got from Steve Robinson, there was a copy of that specification. Much of this discussion is based on that document.

[1] `$Header: d:/ecr6171/RCS/prob.tex,v 1.2 2000-06-09 11:32:32-07 Hamilton Exp Hamilton $`

Protocol Name	Receiver/Driver and AutoDome Control Code Protocol
Applies to	TC8560 Series and TC700 series
Published by	Philips Communication & Security Systems Inc. in 1997
Communications type	RS-232, 2400 or 9600 baud, 1 stop bit, 8 data bits, no parity, no handshake
Data format	1. Length Byte 2. High Address Byte 3. Low Address Byte 4. Opcode Byte 5. Data Byte 1 6. Data Byte 2 7. Data Byte 3 8. Checksum Byte
Address length	14 bits, spread across two bytes of data, in each byte only bits 6 → 0 are used, bit 7 is always `0` and is not part of the address.
Maximum address allowed	16,384

There is a short table that clearly shows that each byte of the address uses 7 bits and that the most significant bit is never set. (They use the MSB set to `1` to indicate a start of message byte, all other bytes have the MSB set to `0`.) In their protocol addressing all numbers start from `0` while in their "device" addressing all addresses start at `1`, thus to get a protocol address from a device address `1` must be subtracted before encoding the address for transmission. Following is a condensed version of their table, it is important to note the addressing changes that occur near powers of two and to note that 16,384 is that maximum address supported.

Camera Number	Value to Be Encoded	High Order Byte	Low Order Byte
1	0	0x00	0x00
2	1	0x00	0x01
256	255	0x01	0x7F
257	256	0x02	0x00
500	499	0x03	0x73
512	511	0x03	0x7F
513	512	0x04	0x00
1,024	1,023	0x07	0x7F
5,000	4,999	0x27	0x07
9,999	9,998	0x4E	0x0E
16,384	16,383	0x7F	0x7F

3 Analysis of the TXB-AB

The TXB-AB is a small PC board that uses two PIC computer chips to decode and translate the Manchester coded communications signal. (This is a point that the technical manual is mute about. It would be nice if it indicated somewhere or other what the input data format was. From looking at the schematics (SH05-008-00C0) it seems to be Manchester only.

The two PIC chips consist of U1 which is a type 16C57 CPU that appears to be used to translate the Manchester data into 8-bit parallel data, and U2 which is a type 16C73A CPU that does the actual translation from Burle to Pelco "D" protocol. (There are some other chips on the board, however they are only support chips and may be ignored for this discussion.) There is no way to select the address that the board responds to. However there are several jumpers (J1, J2, J3 and J4) with J2 and J4 being marked as "Not Installed". The manual seams to indicate that J1 is used to terminate the Manchester data line and that J3 selects an alternate input protocol. Not too surprisingly what the two protocols are isn't mentioned.

4 Discussion on the Pelco "D" protocol

The Pelco "D" protocol document that I have been using is an unreleased manual that is dated 02MAR99 and is 8 pages long. Putting the "D" protocol information in the same format as I did for the Burle protocol gives us:

Protocol Name	Pelco "D" Protocol
Applies to	?? (But probably includes Intercept, Spectra and Esprit type of units.)
Published by	Not a released document, however my copy is marked 1999 and probably is an internal document.
Communications type	Not specified.
Data format	1. Sync Byte 2. Address Byte 3. Command 1 Byte 4. Command 2 Byte 5. Data Byte 1 6. Data Byte 2 7. Checksum Byte
Address length	8 bits.
Maximum address allowed	256

It should be noted that the maximum address allowed by this protocol will not allow access to 300 cameras.

5 Comments on the TXB-AB source code

I obtained the source code for this project (the source code for U1 is PG51-0022-0124, and the source code for U2 is PG51-0015-0124) and have been giving it a casual look. It does have a reasonable set of comments and probably has a working set of project build files. (I have no experience with the PIC family of computer chips, and can not evaluate if all the required files are present, however I believe that they are.)

With out doing a detailed examination of the code I believe that the code running on U1 breaks down the Manchester formatted data and sends **all** the data over to the U2 chip. Then the U2 chip takes that data and translates it into "D" protocol which is then sent over to the camera.

There is a problem with this process in that, how does the U2 chip deal with two bytes of address coming in and only one going out? Here I have to speculate as to what happens to the first (or high order) byte of address. I believe that it is "thrown away" (It must be thrown away, but which chip does it? U1? or U2? This makes a difference latter on.) thus giving a nice output address that will always fit into the "D" protocol address range.

If my assumption about what happens to the high order address byte is correct, then all we have to do is to figure how to pass its information through to the Spectra. It gets even easier if U2 is the chip to throw it away, as then the software in U1 does not need to be changed.

5.1 First fix

There is a "quick and dirty" way to partially do this and that is to decode the high order byte and add its information into the data being sent out in "D" protocol. This solution will give us an "improved" maximum address of 256 (which is still short of the requested 300). This will require a modification to the software for the TXB-AB units only.

5.2 Second fix

Then there is a more advanced fix that involves doing the first fix and then modifying the Spectra, etc., units so that they speak a modified version of "D" protocol. This would involve using spare bits in SW1 where there appear to be 6 (or maybe 7) spare bits. This would increase the addressing range of "D" protocol from a maximum of 256 to 8,192 (or 16,384 if seven bits are available). I dislike this solution for two reasons: first every Spectra, etc., camera will have to have this modification and second I do not think that we should use up all the "spare" switch positions. This will require modifications to the TXB-AB's software as well as to all Spectra, etc., units.

5.3 Third fix

As an alternate the TXB-AB board could be modified by adding an address switch that it honored for just the higher order part of the address. If this is done then only the TXB-AB boards will require modification as will the software for the TXB-AB. Now the TXB-AB/Spectra combination will be able to support a maximum address of 8,192.

5.4 Fourth fix

This isn't really a fix at all. In reading the technical manual for the TXB-AB unit, I found that they make a statement of "Each of the 16 AD groups has its own communication line that controls 64 devices; hence, each group on the chart contains 64 done addresses." (Page 7)

I am not clear what this means.

- Does it mean that a group (for example: with group 1, address range 1 → 64) only gets addresses in its range?
- Or does it indicate that all addresses are sent to group 1 and that group 1 should only decode addresses in the 1 → 64 range?
- Would address 444 ever be seen in group 1?

- What exactly is a group?
  - Is a group a bunch of physically connected items in a single chain?
  - Or is it a logical group which is spread over many transmission chains?

If we are lucky and only addresses from 1 → 64 appear in a physical chain of group 1 cameras. Then we can use the aliases that occur when we only decode the lower 7 bits of data to get many cameras on the same system. I.e. if address 17 in group 1 is never sent to group 5 cameras, then when address 273 is sent to group 5 it will decode as 17 which can then be sent to the camera.

If groups are handled as I hope they are, then there are no modifications needed and we can address 8,192 cameras with no problems (and no modifications to any hardware or software, but this will probably need to have something done to the technical manual).

6 Comments about the Technical Manual for the TXB-AB

The manual on the TXB-AB provides very little “hard” information about what the TXB-AB does and what the changes made in the Spectra, etc., actually do. I could not find any clear indication of what the input signal was, nor could I determine what the output protocol was. To do this I had to read the source code and the schematics. I do not believe that our customer is well served by this lack of information in our manuals.