Byte and Message Timings

30 June 2010

Eric Hamilton

Contents

List of Figures

[1] `$Header: d:/Binder2/ByteTime/RCS/ByteTime.tex,v 1.7 2010-06-29 14:43:06-07 Hamilton Exp Hamilton $`

List of Tables

1 Message Durations

1.1 How to use the tables

A serial byte of data consists of several parts:

1. The electrical format of the data is what is called "Non-Return to Zero" or NRZ. This means that there is no return to a base value in between each bit time. Thus the only way that an end of bit may be detected is to maintain accurate and careful timing during the decode of the bits in a data byte. (In the figures that accompany this note, there are arrows to show where each bit ends and starts. On a "real" oscilloscope there are no helpful arrows.)

2. Each byte starts out with a "start" bit. The duration of a start bit is $\frac{1}{baudrate}$ when baud rate is defined as number of equal length bits per second.

3. Following the start bit are a variable number of data bits. Most protocols used at Pelco use eight "data" bits per byte. Other protocols use as few as five and as many as twelve. (Other lengths are legal.)

4. After the data bits there may, or may not, be a "parity" bit. If a parity bit is included in the byte, then it has four different values:

   (a) Odd, this bit will cause all "1" bits in the byte to be have an odd count of 1s.

   (b) Even, this bit will cause all "1" bits in the byte to be have an even count of 1s.

   (c) Mark, in this case the bit will always be "on". Some systems will define "on" to be high and others will define it to be low.

   (d) Space, in this case the bit will always be "off". Some systems will define "off" to be low and others will define it to be high.

   Note that this means that there are five types of parity when "none" is included. (And six if parity is to be ignored, as on receive it is ignored and on transmit it may be anything at all.)

5. At the end of each byte there will be a "stop" bit(s). At Pelco one stop bit is common. Other systems use many different durations which range from 1.00 bit times to over 2.00 bit times. At one time 1.42 was common with older Tele-Type machines. (One of the UARTs in use at Pelco offers the following stop bit durations: 0.563, 0.625, 0.688, 0.750, 0.813, 0.875, 0.938, 1.000, 1.563, 1.625, 1.688, 1.750, 1.813, 1.875, 1.938 and 2.00. (Most UARTs, including programs running under Windows on a PC, only offer 1.0, 1.5 and 2.0 for stop bit durations.)

6. The contents of the duration tables are as follows:

   (a) All timing values in the tables are in milliseconds (ms). I.e $\frac{1}{1000}^{th}$s of a second.

   (b) Full byte timings are given for 10, 11 and 12 bit bytes. As was outlined above these are the three most common byte lengths used at Pelco. 10 bit bytes are normally used with D Protocol for PTZ control. When parity is added in, then the byte length in bits will become 11 bits long. If two stop bits are specified, then the two preceding values become 11 and 12 respectively.

   (c) The duration of a single bit is not specified, but rather the value for a full 10 bit byte may be easily divided by 10 to get a bit's duration. (This was part of an effort to save paper.)

   (d) Full message timings are given for all message lengths up to eight byte messages. At Pelco messages lengths vary from 3 $\rightarrow$ 18 and sometimes longer.

7. The figures that accompany this short note were all taken of commands being sent in RS-422 electrical format. (Except for Figure 8, page 19 and Figure 9, page 19 which are pictures of RS-485 byte.)

8. In RS-422 format, there are two signals, a "+" and a "–" version of the data. The receiver examines the input data and reports out a "1" or "0" based on the difference in the two signals.

[2] `$Header: d:/Binder2/ByteTime/RCS/ByteTime.inc,v 1.10 2010-06-29 14:43:06-07 Hamilton Exp Hamilton $`

9. On the accompanying figures the upper trace, trace 1, is the "+" signal and the lower trace, trace 2, is the "–" signal.

10. The receiver has a decision threshold of 200 mV. I.e. any signal that is greater than 200 mV different is reported out as a "1" while ones that are less are reported out as "0" logical value.

11. It should be noted that when "nothing" is being sent, that the quiescent value is an asserted high, for the "+" signal and a low for the "–" signal.

12. When a command starts, a start bit is sent that changes the value on the line to the "active" condition (and the receiver will output a logical "1") and then the rest of the character follows.

13. When a character ends, a stop bit (and the receiver will output a logical "0") is sent. It just so happens that the stop bit has the same value as the line does when it is in quiescent state. I.e. when nothing is being sent, the system always is sending stop bits.

14. Sometimes there are small timing differences between the receiver and the transmitter. Since infinite stop bits may be interpreted the same as a single stop bit, to get the most reliable possible transmission of data, it is best to send data with two stop bits and receive data with one stop bit. That way if there are any timing differences, there is a built in "safety factor" of about ±10% of the baud rate. No Pelco equipment implements this logic. The only down side to implementing this logic is that messages take about 10% longer to send. At the normal rate of sending message, i.e. when ever the operator fools around with a joy stick, the additional 10% is insignificant.

1.2 The Data Pictures

A Byte without Parity As shown in Figure 1, page 15, a D Protocol Sync byte is sent at 9600 baud which consists of one start bit, eight data bits (which are all "1"s), no parity bit and one stop bit (10 total bits) is shown. Note that the time from the start of the byte to the end of the byte is about 1.040 ms. (A digital oscilloscope was used to make the capture and it quantizes the data so that values shown in the readout are not exact, even though they may appear to be so.)

Bytes with Parity In Figure 2, page 15 another D Protocol Sync byte is shown, except that this time the byte has an "odd" parity bit. Likewise in Figure 3, page 16 a D Protocol Sync of all data "1"s is shown that has even parity.

A Message with No Parity In Figure 4, page 17 a seven byte D Protocol Stop command without parity is shown.

Messages with Parity In Figure 5, page 17 a seven byte D Protocol Zoom In command is shown with no parity. And in Figure 6, page 18 a seven byte D Protocol Zoom In command with odd parity is shown. While in Figure 7, page 18 a D Protocol Zoom In command with even parity is shown.

RS-485 Byte In Figure 8, page 19 a typical RS-485 byte is shown. Note the time to acquire the data line and the release of it after the byte has been sent. Note also that the sending unit is sending two stop bits. (The character being sent is an `ACK` (`0x06`) byte.) The character was sent by a Panasonic system running at 9600 baud.

RS-485 Message In Figure 9, page 19 a typical RS-485 message is shown. Note that the handling of the data line is the same as in the previous example. I.e. the line is seized, used and the released.

1.3 The Byte Duration Tables

This is a collection of the predicted durations of bytes and messages for various baud rates. The format of all messages is 1 start bit, 8 data bits, 0 or 1 parity bits, 1 or 2 stop bits. Giving byte lengths of 10, 11 and 12 bits. Durations, and start/end times, of each byte in a D Protocol or P Protocol messages are listed. Following these tables there are two sets of full message/reply durations at various baud rates.

300 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	33.333	36.667	40.000
2	66.667	73.333	80.000
3	100.000	110.000	120.000
4	133.333	146.667	160.000
5	166.667	183.333	200.000
6	200.000	220.000	240.000
7	233.333	256.667	280.000
8	266.667	293.333	320.000

Table 1: Timings for data sent at 300 baud

600 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	16.667	18.333	20.000
2	33.333	36.667	40.000
3	50.000	55.000	60.000
4	66.667	73.333	80.000
5	83.333	91.667	100.000
6	100.000	110.000	120.000
7	116.667	128.333	140.000
8	133.333	146.667	160.000

Table 2: Timings for data sent at 600 baud

1200 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	8.333	9.167	10.000
2	16.667	18.333	20.000
3	25.000	27.500	30.000
4	33.333	36.667	40.000
5	41.667	45.833	50.000
6	50.000	55.000	60.000
7	58.333	64.167	70.000
8	66.667	73.333	80.000

Table 3: Timings for data sent at 1200 baud

2400 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	4.167	4.583	5.000
2	8.333	9.167	10.000
3	12.500	13.750	15.000
4	16.667	18.333	20.000
5	20.833	22.917	25.000
6	25.000	27.500	30.000
7	29.167	32.083	35.000
8	33.333	36.667	40.000

Table 4: Timings for data sent at 2400 baud

4800 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	2.083	2.292	2.500
2	4.167	4.583	5.000
3	6.250	6.875	7.500
4	8.333	9.167	10.000
5	10.417	11.458	12.500
6	12.500	13.750	15.000
7	14.583	16.042	17.500
8	16.667	18.333	20.000

Table 5: Timings for data sent at 4800 baud

9600 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	1.042	1.146	1.250
2	2.083	2.292	2.500
3	3.125	3.438	3.750
4	4.167	4.583	5.000
5	5.208	5.729	6.250
6	6.250	6.875	7.500
7	7.292	8.021	8.750
8	8.333	9.167	10.000

Table 6: Timings for data sent at 9600 baud

14400 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	0.694	0.764	0.833
2	1.389	1.528	1.667
3	2.083	2.292	2.500
4	2.778	3.056	3.333
5	3.472	3.819	4.167
6	4.167	4.583	5.000
7	4.861	5.347	5.833
8	5.556	6.111	6.667

Table 7: Timings for data sent at 14400 baud

19200 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	0.521	0.573	0.625
2	1.042	1.146	1.250
3	1.563	1.719	1.875
4	2.083	2.292	2.500
5	2.604	2.865	3.125
6	3.125	3.438	3.750
7	3.646	4.010	4.375
8	4.167	4.583	5.000

Table 8: Timings for data sent at 19200 baud

28800 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	0.347	0.382	0.417
2	0.694	0.764	0.833
3	1.042	1.146	1.250
4	1.389	1.528	1.667
5	1.736	1.910	2.083
6	2.083	2.292	2.500
7	2.431	2.674	2.917
8	2.778	3.056	3.333

Table 9: Timings for data sent at 28800 baud

38400 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	0.260	0.286	0.313
2	0.521	0.573	0.625
3	0.781	0.859	0.938
4	1.042	1.146	1.250
5	1.302	1.432	1.563
6	1.563	1.719	1.875
7	1.823	2.005	2.188
8	2.083	2.292	2.500

Table 10: Timings for data sent at 38400 baud

57600 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	0.174	0.191	0.208
2	0.347	0.382	0.417
3	0.521	0.573	0.625
4	0.694	0.764	0.833
5	0.868	0.955	1.042
6	1.042	1.146	1.250
7	1.215	1.337	1.458
8	1.389	1.528	1.667

Table 11: Timings for data sent at 57600 baud

115200 baud	10 bit bytes	11 bit bytes	12 bit bytes
1	0.087	0.095	0.104
2	0.174	0.191	0.208
3	0.260	0.286	0.313
4	0.347	0.382	0.417
5	0.434	0.477	0.521
6	0.521	0.573	0.625
7	0.608	0.668	0.729
8	0.694	0.764	0.833

Table 12: Timings for data sent at 115200 baud

1.4 Bit/Byte/Message durations in μs

Message durations in μs at 300 baud							
	1	7	8	9	10	11	12
300	3333.333	23333.333	26666.667	30000.000	33333.333	36666.667	40000.000
4	13333.333	93333.333	106666.667	120000.000	133333.333	146666.667	160000.000
7	23333.333	163333.333	186666.667	210000.000	233333.333	256666.667	280000.000
18	60000.000	420000.000	480000.000	540000.000	600000.000	660000.000	720000.000

Message durations in μs at 600 baud							
	1	7	8	9	10	11	12
600	1666.667	11666.667	13333.333	15000.000	16666.667	18333.333	20000.000
4	6666.667	46666.667	53333.333	60000.000	66666.667	73333.333	80000.000
7	11666.667	81666.667	93333.333	105000.000	116666.667	128333.333	140000.000
18	30000.000	210000.000	240000.000	270000.000	300000.000	330000.000	360000.000

Message durations in μs at 1200 baud							
	1	7	8	9	10	11	12
1200	833.333	5833.333	6666.667	7500.000	8333.333	9166.667	10000.000
4	3333.333	23333.333	26666.667	30000.000	33333.333	36666.667	40000.000
7	5833.333	40833.333	46666.667	52500.000	58333.333	64166.667	70000.000
18	15000.000	105000.000	120000.000	135000.000	150000.000	165000.000	180000.000

Message durations in μs at 1800 baud							
	1	7	8	9	10	11	12
1800	555.556	3888.889	4444.444	5000.000	5555.556	6111.111	6666.667
4	2222.222	15555.556	17777.778	20000.000	22222.222	24444.444	26666.667
7	3888.889	27222.222	31111.111	35000.000	38888.889	42777.778	46666.667
18	10000.000	70000.000	80000.000	90000.000	100000.000	110000.000	120000.000

Message durations in μs at 2400 baud							
	1	7	8	9	10	11	12
2400	416.667	2916.667	3333.333	3750.000	4166.667	4583.333	5000.000
4	1666.667	11666.667	13333.333	15000.000	16666.667	18333.333	20000.000
7	2916.667	20416.667	23333.333	26250.000	29166.667	32083.333	35000.000
18	7500.000	52500.000	60000.000	67500.000	75000.000	82500.000	90000.000

Message durations in μs at 3600 baud							
	1	7	8	9	10	11	12
3600	277.778	1944.444	2222.222	2500.000	2777.778	3055.556	3333.333
4	1111.111	7777.778	8888.889	10000.000	11111.111	12222.222	13333.333
7	1944.444	13611.111	15555.556	17500.000	19444.444	21388.889	23333.333
18	5000.000	35000.000	40000.000	45000.000	50000.000	55000.000	60000.000

[3] `$Header: d:/Binder2/ByteTime/RCS/Tus.inc,v 1.1 2010-06-29 12:16:18-07 Hamilton Exp Hamilton $`

Message durations in μs at 4800 baud							
	1	7	8	9	10	11	12
4800	208.333	1458.333	1666.667	1875.000	2083.333	2291.667	2500.000
4	833.333	5833.333	6666.667	7500.000	8333.333	9166.667	10000.000
7	1458.333	10208.333	11666.667	13125.000	14583.333	16041.667	17500.000
18	3750.000	26250.000	30000.000	33750.000	37500.000	41250.000	45000.000

Message durations in μs at 9600 baud							
	1	7	8	9	10	11	12
9600	104.167	729.167	833.333	937.500	1041.667	1145.833	1250.000
4	416.667	2916.667	3333.333	3750.000	4166.667	4583.333	5000.000
7	729.167	5104.167	5833.333	6562.500	7291.667	8020.833	8750.000
18	1875.000	13125.000	15000.000	16875.000	18750.000	20625.000	22500.000

Message durations in μs at 14400 baud							
	1	7	8	9	10	11	12
14400	69.444	486.111	555.556	625.000	694.444	763.889	833.333
4	277.778	1944.444	2222.222	2500.000	2777.778	3055.556	3333.333
7	486.111	3402.778	3888.889	4375.000	4861.111	5347.222	5833.333
18	1250.000	8750.000	10000.000	11250.000	12500.000	13750.000	15000.000

Message durations in μs at 19200 baud							
	1	7	8	9	10	11	12
19200	52.083	364.583	416.667	468.750	520.833	572.917	625.000
4	208.333	1458.333	1666.667	1875.000	2083.333	2291.667	2500.000
7	364.583	2552.083	2916.667	3281.250	3645.833	4010.417	4375.000
18	937.500	6562.500	7500.000	8437.500	9375.000	10312.500	11250.000

Message durations in μs at 28800 baud							
	1	7	8	9	10	11	12
28800	34.722	243.056	277.778	312.500	347.222	381.944	416.667
4	138.889	972.222	1111.111	1250.000	1388.889	1527.778	1666.667
7	243.056	1701.389	1944.444	2187.500	2430.556	2673.611	2916.667
18	625.000	4375.000	5000.000	5625.000	6250.000	6875.000	7500.000

Message durations in μs at 38400 baud							
	1	7	8	9	10	11	12
38400	26.042	182.292	208.333	234.375	260.417	286.458	312.500
4	104.167	729.167	833.333	937.500	1041.667	1145.833	1250.000
7	182.292	1276.042	1458.333	1640.625	1822.917	2005.208	2187.500
18	468.750	3281.250	3750.000	4218.750	4687.500	5156.250	5625.000

Message durations in μs at 57600 baud							
	1	7	8	9	10	11	12
57600	17.361	121.528	138.889	156.250	173.611	190.972	208.333
4	69.444	486.111	555.556	625.000	694.444	763.889	833.333
7	121.528	850.694	972.222	1093.750	1215.278	1336.806	1458.333
18	312.500	2187.500	2500.000	2812.500	3125.000	3437.500	3750.000

Message durations in μs at 115200 baud							
	1	7	8	9	10	11	12
115200	8.681	60.764	69.444	78.125	86.806	95.486	104.167
4	34.722	243.056	277.778	312.500	347.222	381.944	416.667
7	60.764	425.347	486.111	546.875	607.639	668.403	729.167
18	156.250	1093.750	1250.000	1406.250	1562.500	1718.750	1875.000

Byte/Bit Times in μs							
Baud	1	7	8	9	10	11	12
300	3333.333	23333.333	26666.667	30000.000	33333.333	36666.667	40000.000
600	1666.667	11666.667	13333.333	15000.000	16666.667	18333.333	20000.000
1200	833.333	5833.333	6666.667	7500.000	8333.333	9166.667	10000.000
1800	555.556	3888.889	4444.444	5000.000	5555.556	6111.111	6666.667
2400	416.667	2916.667	3333.333	3750.000	4166.667	4583.333	5000.000
3600	277.778	1944.444	2222.222	2500.000	2777.778	3055.556	3333.333
4800	208.333	1458.333	1666.667	1875.000	2083.333	2291.667	2500.000
9600	104.167	729.167	833.333	937.500	1041.667	1145.833	1250.000
14400	69.444	486.111	555.556	625.000	694.444	763.889	833.333
19200	52.083	364.583	416.667	468.750	520.833	572.917	625.000
28800	34.722	243.056	277.778	312.500	347.222	381.944	416.667
38400	26.042	182.292	208.333	234.375	260.417	286.458	312.500
57600	17.361	121.528	138.889	156.250	173.611	190.972	208.333
115200	8.681	60.764	69.444	78.125	86.806	95.486	104.167

Table 13: Byte/Bit Times in μs

1.5 Bit/Byte/Message durations in ms

Message durations in ms at 300 baud

	1	7	8	9	10	11	12
300	3.333	23.333	26.667	30.000	33.333	36.667	40.000
4	13.333	93.333	106.667	120.000	133.333	146.667	160.000
7	23.333	163.333	186.667	210.000	233.333	256.667	280.000
18	60.000	420.000	480.000	540.000	600.000	660.000	720.000

Message durations in ms at 600 baud

	1	7	8	9	10	11	12
600	1.667	11.667	13.333	15.000	16.667	18.333	20.000
4	6.667	46.667	53.333	60.000	66.667	73.333	80.000
7	11.667	81.667	93.333	105.000	116.667	128.333	140.000
18	30.000	210.000	240.000	270.000	300.000	330.000	360.000

Message durations in ms at 1200 baud

	1	7	8	9	10	11	12
1200	0.833	5.833	6.667	7.500	8.333	9.167	10.000
4	3.333	23.333	26.667	30.000	33.333	36.667	40.000
7	5.833	40.833	46.667	52.500	58.333	64.167	70.000
18	15.000	105.000	120.000	135.000	150.000	165.000	180.000

Message durations in ms at 1800 baud

	1	7	8	9	10	11	12
1800	0.556	3.889	4.444	5.000	5.556	6.111	6.667
4	2.222	15.556	17.778	20.000	22.222	24.444	26.667
7	3.889	27.222	31.111	35.000	38.889	42.778	46.667
18	10.000	70.000	80.000	90.000	100.000	110.000	120.000

Message durations in ms at 2400 baud

	1	7	8	9	10	11	12
2400	0.417	2.917	3.333	3.750	4.167	4.583	5.000
4	1.667	11.667	13.333	15.000	16.667	18.333	20.000
7	2.917	20.417	23.333	26.250	29.167	32.083	35.000
18	7.500	52.500	60.000	67.500	75.000	82.500	90.000

Message durations in ms at 3600 baud

	1	7	8	9	10	11	12
3600	0.278	1.944	2.222	2.500	2.778	3.056	3.333
4	1.111	7.778	8.889	10.000	11.111	12.222	13.333
7	1.944	13.611	15.556	17.500	19.444	21.389	23.333
18	5.000	35.000	40.000	45.000	50.000	55.000	60.000

[4] `$Header: d:/Binder2/ByteTime/RCS/Tms.inc,v 1.1 2010-06-29 12:16:12-07 Hamilton Exp Hamilton $`

Message durations in ms at 4800 baud							
	1	7	8	9	10	11	12
4800	0.208	1.458	1.667	1.875	2.083	2.292	2.500
4	0.833	5.833	6.667	7.500	8.333	9.167	10.000
7	1.458	10.208	11.667	13.125	14.583	16.042	17.500
18	3.750	26.250	30.000	33.750	37.500	41.250	45.000

Message durations in ms at 9600 baud							
	1	7	8	9	10	11	12
9600	0.104	0.729	0.833	0.938	1.042	1.146	1.250
4	0.417	2.917	3.333	3.750	4.167	4.583	5.000
7	0.729	5.104	5.833	6.563	7.292	8.021	8.750
18	1.875	13.125	15.000	16.875	18.750	20.625	22.500

Message durations in ms at 14400 baud							
	1	7	8	9	10	11	12
14400	0.069	0.486	0.556	0.625	0.694	0.764	0.833
4	0.278	1.944	2.222	2.500	2.778	3.056	3.333
7	0.486	3.403	3.889	4.375	4.861	5.347	5.833
18	1.250	8.750	10.000	11.250	12.500	13.750	15.000

Message durations in ms at 19200 baud							
	1	7	8	9	10	11	12
19200	0.052	0.365	0.417	0.469	0.521	0.573	0.625
4	0.208	1.458	1.667	1.875	2.083	2.292	2.500
7	0.365	2.552	2.917	3.281	3.646	4.010	4.375
18	0.938	6.563	7.500	8.438	9.375	10.313	11.250

Message durations in ms at 28800 baud							
	1	7	8	9	10	11	12
28800	0.035	0.243	0.278	0.313	0.347	0.382	0.417
4	0.139	0.972	1.111	1.250	1.389	1.528	1.667
7	0.243	1.701	1.944	2.188	2.431	2.674	2.917
18	0.625	4.375	5.000	5.625	6.250	6.875	7.500

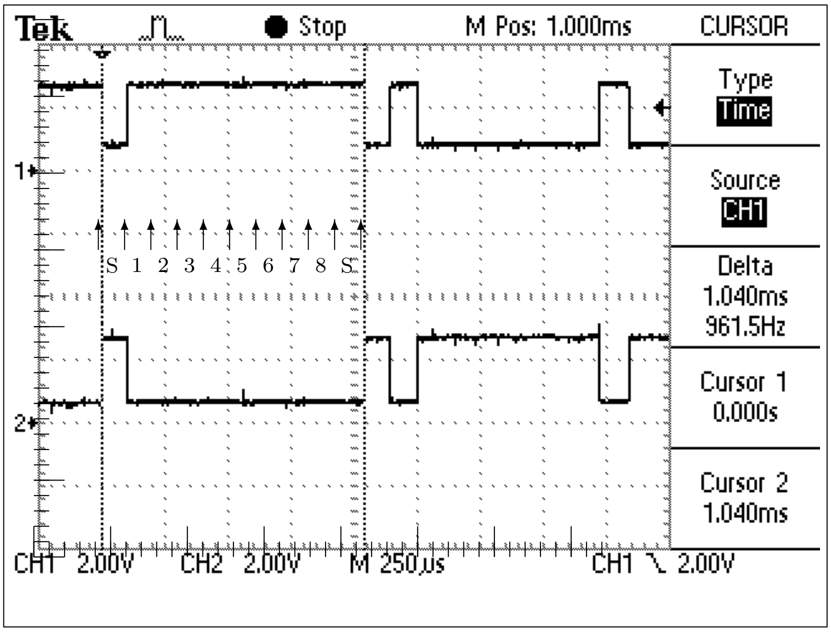
Message durations in ms at 38400 baud							
	1	7	8	9	10	11	12
38400	0.026	0.182	0.208	0.234	0.260	0.286	0.313
4	0.104	0.729	0.833	0.938	1.042	1.146	1.250
7	0.182	1.276	1.458	1.641	1.823	2.005	2.188
18	0.469	3.281	3.750	4.219	4.688	5.156	5.625

Message durations in ms at 57600 baud							
	1	7	8	9	10	11	12
57600	0.017	0.122	0.139	0.156	0.174	0.191	0.208
4	0.069	0.486	0.556	0.625	0.694	0.764	0.833
7	0.122	0.851	0.972	1.094	1.215	1.337	1.458
18	0.313	2.188	2.500	2.813	3.125	3.438	3.750

Message durations in ms at 115200 baud							
	1	7	8	9	10	11	12
115200	0.009	0.061	0.069	0.078	0.087	0.095	0.104
4	0.035	0.243	0.278	0.313	0.347	0.382	0.417
7	0.061	0.425	0.486	0.547	0.608	0.668	0.729
18	0.156	1.094	1.250	1.406	1.563	1.719	1.875

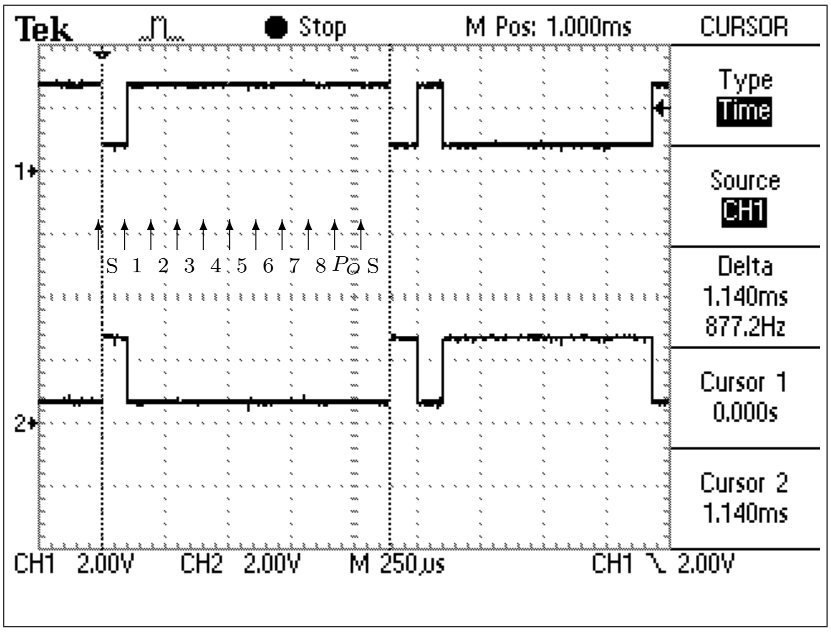
Byte/Bit Times in ms							
Baud	1	7	8	9	10	11	12
300	3.333	23.333	26.667	30.000	33.333	36.667	40.000
600	1.667	11.667	13.333	15.000	16.667	18.333	20.000
1200	0.833	5.833	6.667	7.500	8.333	9.167	10.000
1800	0.556	3.889	4.444	5.000	5.556	6.111	6.667
2400	0.417	2.917	3.333	3.750	4.167	4.583	5.000
3600	0.278	1.944	2.222	2.500	2.778	3.056	3.333
4800	0.208	1.458	1.667	1.875	2.083	2.292	2.500
9600	0.104	0.729	0.833	0.938	1.042	1.146	1.250
14400	0.069	0.486	0.556	0.625	0.694	0.764	0.833
19200	0.052	0.365	0.417	0.469	0.521	0.573	0.625
28800	0.035	0.243	0.278	0.313	0.347	0.382	0.417
38400	0.026	0.182	0.208	0.234	0.260	0.286	0.313
57600	0.017	0.122	0.139	0.156	0.174	0.191	0.208
115200	0.009	0.061	0.069	0.078	0.087	0.095	0.104

Table 14: Byte/Bit Times in ms



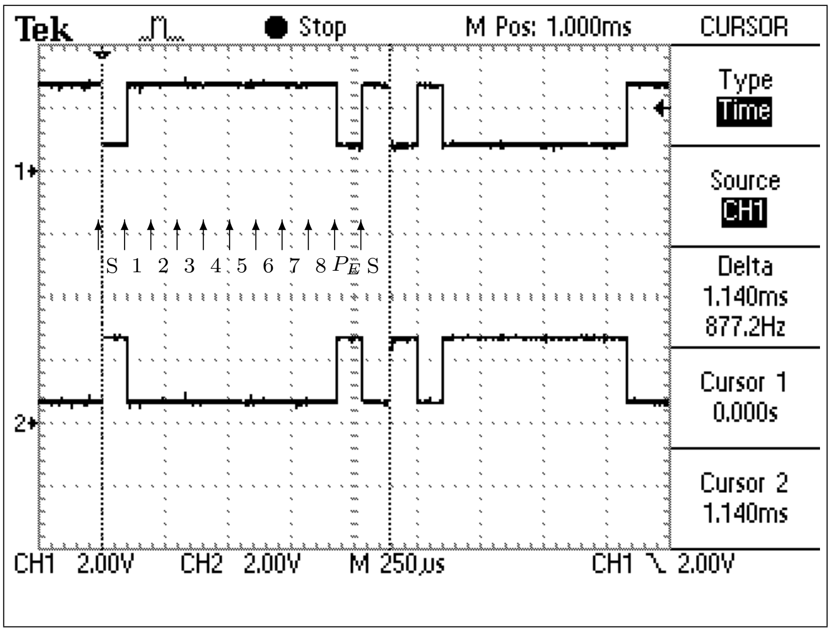
ByteTime004.ps 04/19/2007 09:11 AM

Figure 1: Sync byte of a D Protocol message, 1 Start, 8 data bits of all 1s, No parity, 1 Stop



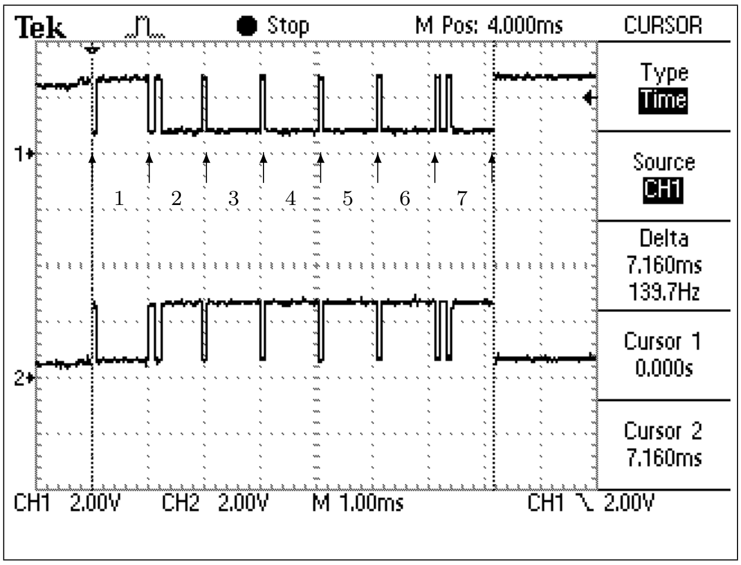
ByteTime001.ps 04/19/2007 09:11 AM

Figure 2: Sync byte of a D Protocol message, 1 Start, 8 data bits of all 1s, Odd parity, 1 Stop



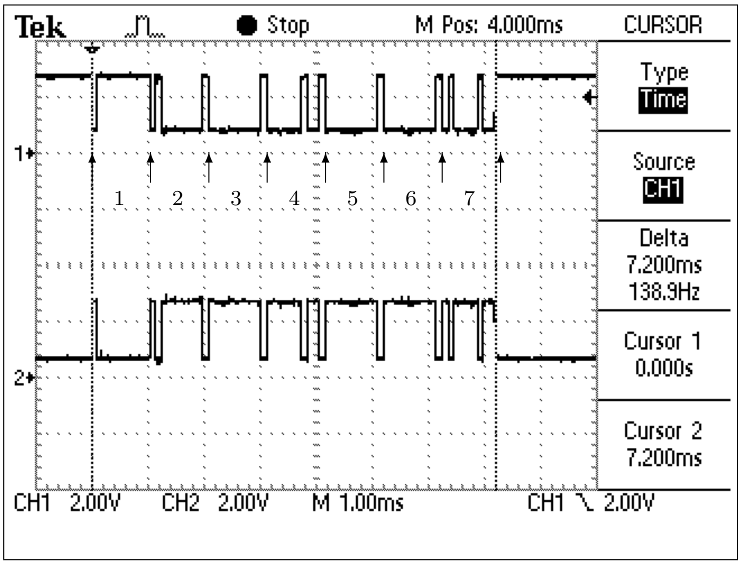
ByteTime002.ps 04/19/2007 09:11 AM

Figure 3: Sync byte of a D Protocol message, 1 Start, 8 data bits of all 1s, Even parity, 1 Stop



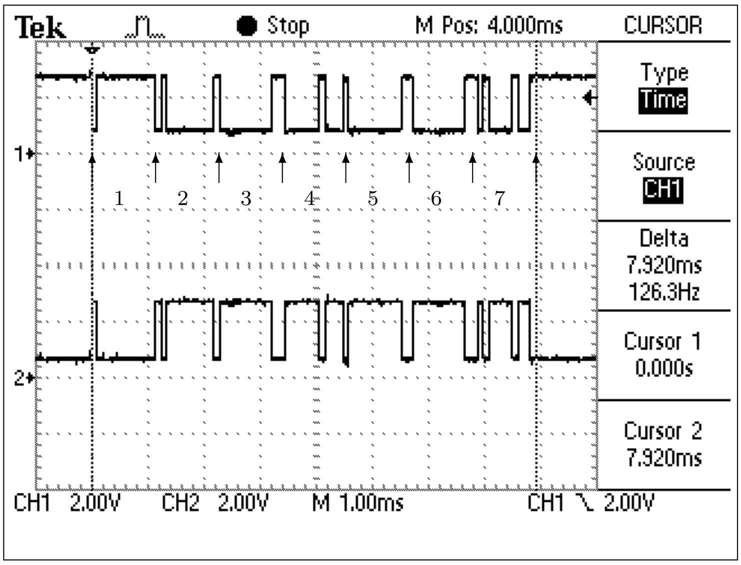
ByteTime009.ps 04/19/2007 09:11 AM

Figure 4: Full D Protocol Stop message of 1 Start, 8 data, No parity, 1 Stop



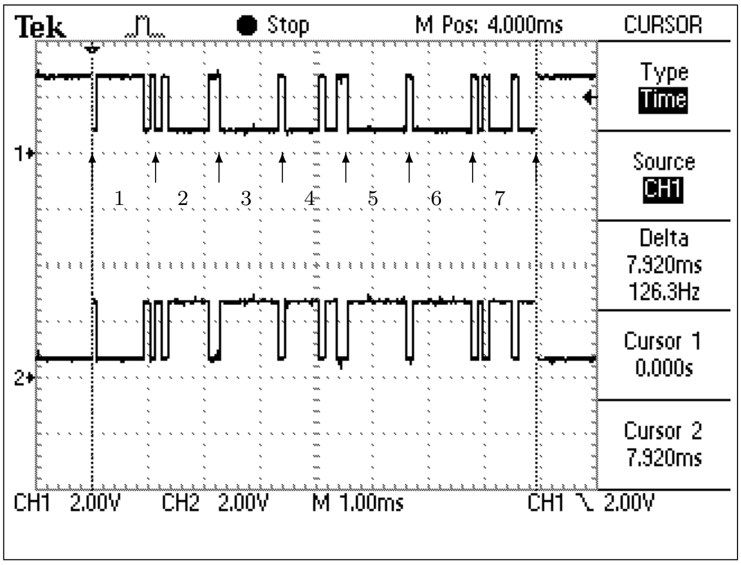
ByteTime005.ps 04/19/2007 09:11 AM

Figure 5: Full D Protocol Zoom In message of 1 Start, 8 data, No parity, 1 Stop



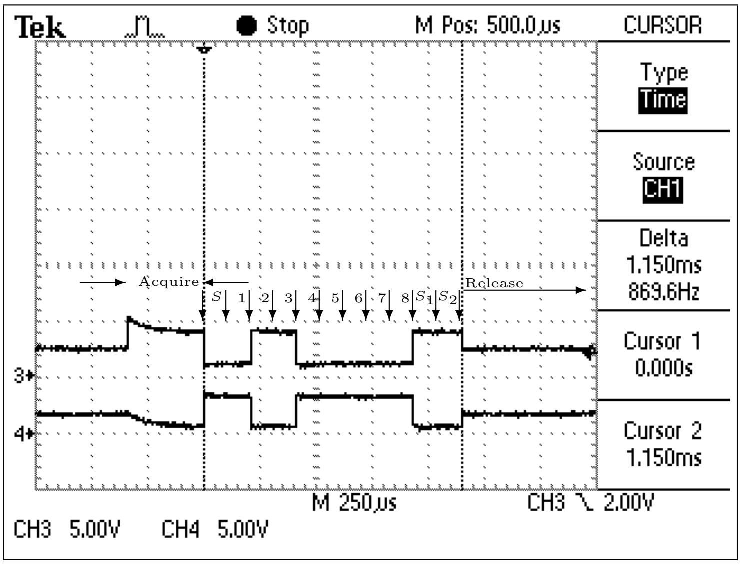
ByteTime007.ps 04/19/2007 09:11 AM

Figure 6: Full D Protocol Zoom In message of 1 Start, 8 data, Odd parity, 1 Stop



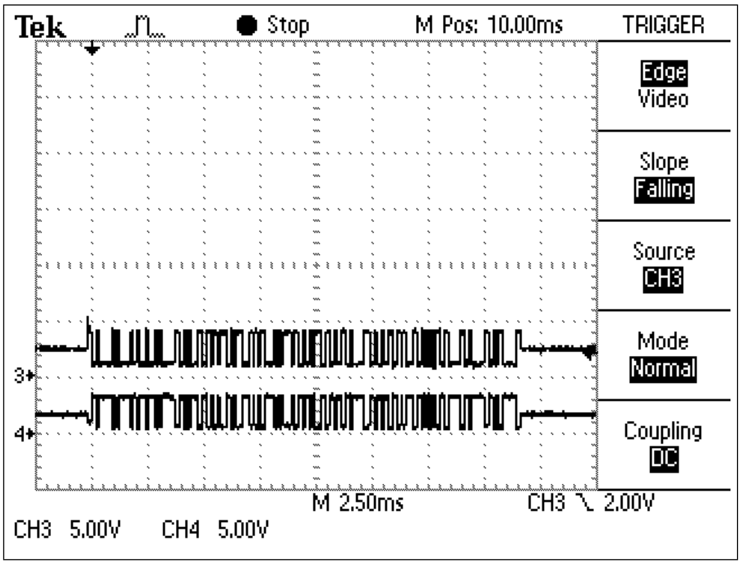
ByteTime006.ps 04/19/2007 09:11 AM

Figure 7: Full D Protocol Zoom In message of 1 Start, 8 data, Even parity, 1 Stop



23apr07a009.ps 04/23/2007 07:37 AM

Figure 8: Single byte of RS-485 data, 1 Start, 8 data, 2 Stops



23apr07a011.ps 04/23/2007 07:37 AM

Figure 9: Full 18 Byte message RS-485 data

2 Parity and character bit definations

Communications messages are built up of several bits. These bits are sent asynchronisouly as a NRZ (Non Return to Zero) bit stream.

There are four attached scope pictures of various character streams. These are:

1. Figure 10, page 23 shows the letter "`c`" followed by a carriage return and a line feed for three bytes of data being sent at 1200 baud. This byte string was sent using a USB/RS-232 converter.

2. Figure 11, page 23 shows the letter "`c`" followed by a carriage return and a line feed for three bytes of data being sent at 115200 baud. This byte string was sent using a USB/RS-232 converter.

3. Figure 12, page 24 shows the numbers "`12345`" followed by a carriage return and a line feed for seven bytes of data being sent at 1200 baud. This byte string was sent using a USB/RS-232 converter.

4. Figure 13, page 24 shows the numbers "`12345`" followed by a carriage return and a line feed for seven bytes of data being sent at 115200 baud. This byte string was sent using a USB/RS-232 converter.

In the above scope pictures the letter `c` (a lower case "c") was chosen because it was convenient. The carriage return/line feed pair were appended by the operating system. And the string `12345` was chosen because that would give a seven byte string which is the same as is used in D Protocol.

2.1 Contents of a character

Each character consists of:

1. The RS-232 standard uses negative true logic.

2. A "start" bit goes from the "idle", or negative, to the positive level for one bit time.

3. "Data" bits are non-return to zero (NRZ) encoded, so there is no gap between them.

4. "Parity" bits, when present, are identical to the data bits, except for their special use of being "parity" bits.

5. "Stop" bits are NRZ bits too.

6. During the "idle" time between characters, the line is held at the "stop" bit level. Thus there are close to an infinite number of "stop" bits being sent between bunches of characters.

7. There is a special case in asynchronous communication, which is ignored in Pelco's PTZ units, and that is the "break" state. "Break" consists of a long series of NRZ bits at the same level as the "start" bit. It is unclear how many bit times are needed for a "break", however it is usually longer than a second or so.

8. The bits of a character are sent out with the LSB first. Well actually it is the "start" bit that goes out first, followed by the data bits, parity and stop bits.

9. All of this is shown in Figure 10, page 23 which is a lower case `c` followed by a CR/LF sequence.

10. Pelco uses RS-422/RS-485 to control our PTZ units. In these two standards, the signal is sent as a normal and as a complemented value. Thus, depending on which line is being observed, the values may be high or low true.

1. One start bit,

2. Eight data bits, but the data bits may vary from $5 \rightarrow 8$ with 8 being the most common for use in CCTV control applications.

3. One parity bit of which there are five choices:

   (a) No Parity or None Parity, in this case the parity bit is not sent reducing the total number of bits sent by one.

   Starting with an original byte of `0110 0011`, a lower case `c`, we will send in serial:

   Note there are no gaps in the byte. Gaps are shown here for clarity only.

[5] `$Header: d:/Binder2/ByteTime/RCS/parity.inc,v 1.1 2010-06-30 07:04:21-07 Hamilton Exp Hamilton $`

   i. `1100 0110`, reversing the bit order.
   ii. `0011 1001`, complementing the bits.
   iii. `1 0011 1001`, adding in a start bit.
   iv. `1 0011 1001 00`, adding in two stop bits.

(b) Odd parity, a single bit that when added to the character provides for an odd total number of data+parity bits being active in the byte.

   Using the same original byte as was used for No Parity, we now get:

   i. `1100 0110`, reversing the bit order.
   ii. `0011 1001`, complementing the bits.
   iii. `1 0011 1001`, adding in a start bit.
   iv. `1 0011 1001 1`, adding in an odd parity bit.
   v. `1 0011 1001 1 00`, adding in two stop bits.

   Note that the parity bit does not include the "start" bit in its calculations.

(c) Even Parity, a single bit that when added to the character provides for an even total number of data+parity bits being active in the byte.

   Using the same original byte as was used for No Parity, we now get:

   i. `1100 0110`, reversing the bit order.
   ii. `0011 1001`, complementing the bits.
   iii. `1 0011 1001`, adding in a start bit.
   iv. `1 0011 1001 0`, adding in an odd parity bit.
   v. `1 0011 1001 0 00`, adding in two stop bits.

(d) Mark Parity, a single bit that is added to the character as a "`1`" value.

   Using the same original byte as was used for No Parity, we now get:

   i. `1100 0110`, reversing the bit order.
   ii. `0011 1001`, complementing the bits.
   iii. `1 0011 1001`, adding in a start bit.
   iv. `1 0011 1001 1`, adding in a "mark" parity bit.
   v. `1 0011 1001 1 00`, adding in two stop bits.

(e) Space Parity, a single bit that is added to the character as a "`0`" value.

   Using the same original byte as was used for No Parity, we now get:

   i. `1100 0110`, reversing the bit order.
   ii. `0011 1001`, complementing the bits.
   iii. `1 0011 1001`, adding in a start bit.
   iv. `1 0011 1001 0`, adding in a "space" parity bit.
   v. `1 0011 1001 0 00`, adding in two stop bits.

(f) Then there is a special case of "ignore" parity, in which provision is provided to tolerate a parity bit on receive, and no parity bit is generated on transmit.

4. One or two stop bits. Depending on the application each character has a stop period between characters. The stop state is the level that is sent between messages, i.e. stop bits are being sent when "nothing" is being sent. Thus a UART has to be able to work correctly with stop bit counts varying from 1 to slightly less than ∞ bits. Traditional stop bit times provide for non-integer times with older values of 1.42 being a common value. The current closest value for this is 1.5 bit times for a stop bit and is supported by Windows (and most other operating systems).

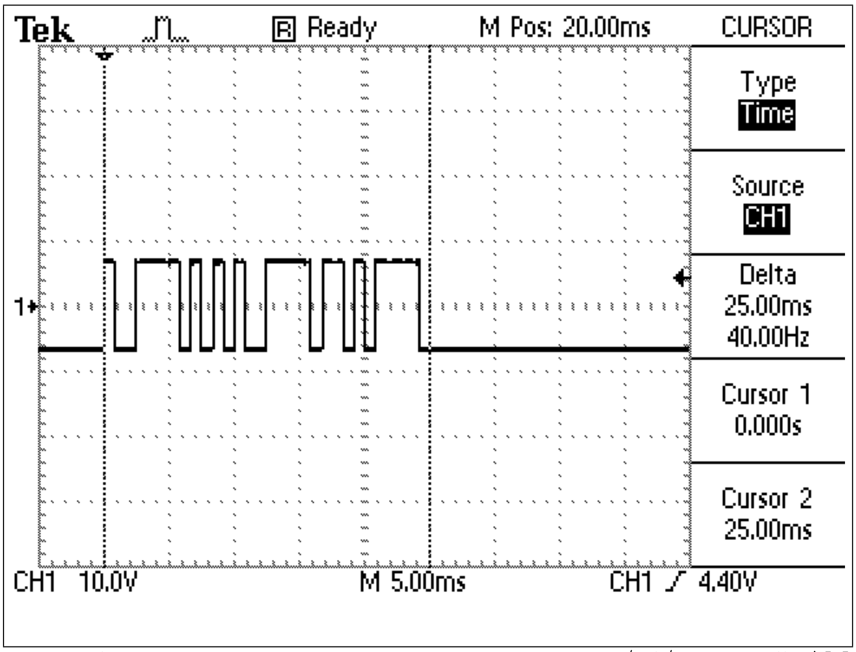
Using the above rules for counting character bits we get, 10 bit times minimum (10 = 1 start + 8 data + 0 parity + 1 stop bits) to 12 bit times maximum (10 = 1 start + 8 data + 1 parity + 2 stop bits), when bytes are in a message and many, many "stop" bit times in between messages.

The Table 13, page 10 lists timing, in μs, and Table 14, page 14, in ms, for various numbers of bits/bytes/messages at various baud rates.

2.2 Scope pictures of byte strings

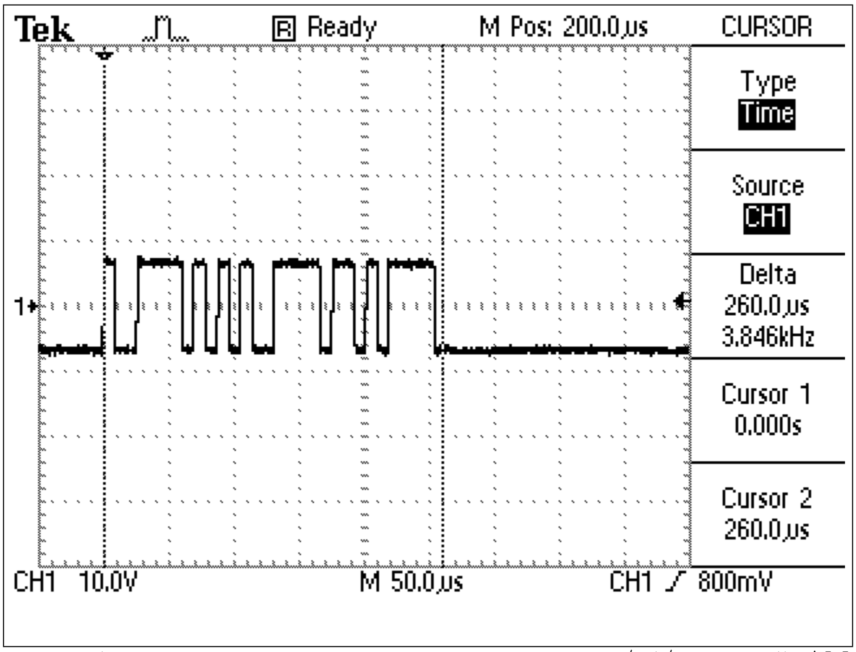
Note

These pictures were taken using a Tektronics model TDS-224 digital oscilloscope on March 14^{th}, 2010.



10mar14a001.ps 03/14/2010 11:47 AM

Figure 10: **com3: A 1200, 8,** `C`



10mar14a006.ps 03/14/2010 11:47 AM

Figure 11: **com3: A 115200, 8,** `C`

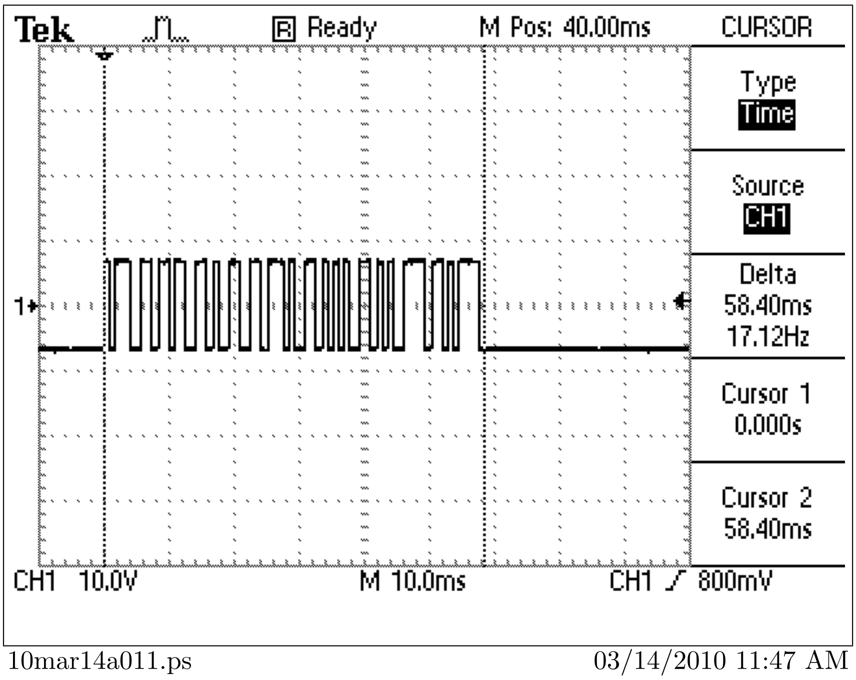


Figure 12: **com3: A 1200, 8,** `12345`

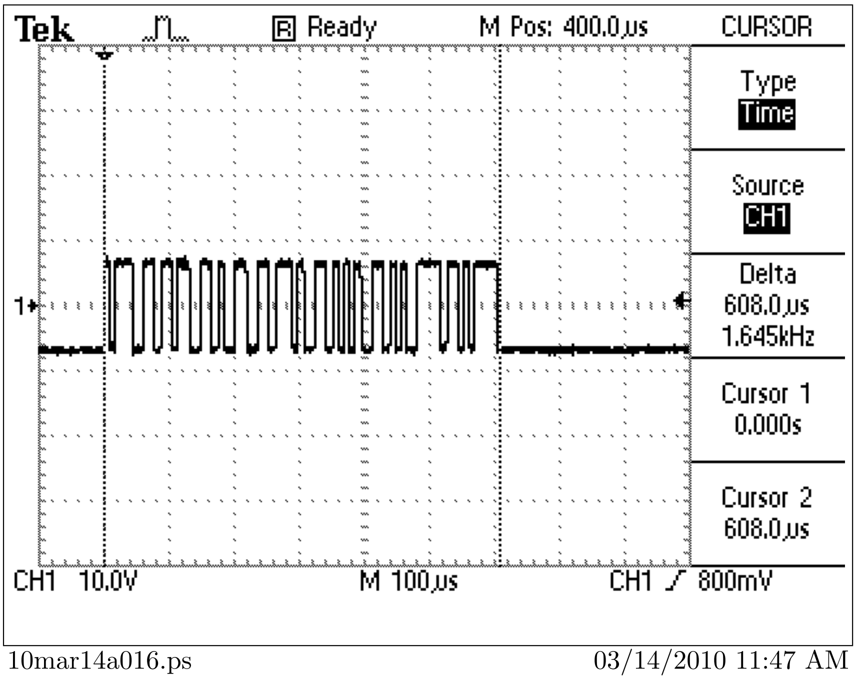


Figure 13: **com3: A 115200, 8,** `12345`

Index