

PX-1000 DES version 1983 - serial output

TEXT-output

header: 600 baud, 8data bits, 1 stop bit

text: 600 baud, 7data bits, 2 stop bit

text 1 = **Hello world!**

```
06 C0 00 F8 C0 E6 0C <-- header 06 C0 = text #1 00 F8 C0 E6 = size + type
28 48 65 6C 6C 6F 20 77 3D <-- block #1 (8 byte + parity)
 ( H e l l o SP w PA

6F 72 6C 64 21 0D 20 20 39 <-- block #2 (8 byte + parity)
 o r l d ! CR SP SP PA

00 00 <-- end
NU NU
```

SP = space

CR = carriage return

PA = parity (xor over 8byte)

NU = NUL character

CRYPTO-output (DES)

header: 600 baud, 8data bits, 1 stop bit

text: 600 baud, 8data bits, 1 stop bit

text = **Hello word!**

key = **123456**

```
06 C0 00 00 E0 E6 0C <-- header
5A 0E 33 F0 53 81 CB 41 CF <-- block #1 (8 byte + parity)
CI CI CI CI CI CI CI CI PA

39 C0 B7 C7 9A B2 CD 08 64 <-- block #2 (8 byte + parity)
CI CI CI CI CI CI CI CI PA

00 00 <-- end
NU NU
```

CI = cipher

DUMP-output

header: 600 baud, 8 data bits, 1 stop bit
text: 1200 baud, 7 data bits, 2 stop bit
text #1 = a
text #2 = ab
text #3 = abc
text #4 = abcd
text #5 = abcde
text #6 = abcdef
text #7 = abcdefg
text #8 = abcdefgh
text #9 = abcdefghi
text #10 = abcdefghij

```
06 C0 00 1E 00 80 0C      <- header text 1 00 1E 00 80 = size + type
28 61 0D 20 20 20 20 20 64  <- text 1
( a CR SP SP SP SP SP PA

18 C0 00 60 C0 80 0C      <- header text 2
28 61 62 0D 20 20 20 20 26  <- text 2
( a b CR SP SP SP SP PA

1E 00 00 66 00 80 0C      <- header text 3
28 61 62 63 0D 20 20 20 65  <- text 3
( a b c CR SP SP SP PA

60 C0 00 78 00 80 0C      <- header text 4
28 61 62 63 64 0D 20 20 21  <- text 4
( a b c d CR SP SP PA

66 00 00 7E C0 80 0C      <- header text 5
28 61 62 63 64 65 0D 20 64  <- text 5
( a b c d e CR SP PA

78 00 00 80 C0 80 0C      <- header text 6
28 61 62 63 64 65 66 0D 22  <- text 6
( a b c d e f CR PA

7E C0 00 86 00 80 0C      <- header text 7
28 61 62 63 64 65 66 67 48  <- text 7 block #1
( a b c d e f g PA
20 20 20 20 20 20 20 20 2D  <- text 7 block #2
SP SP SP SP SP SP SP SP PA

80 C0 00 98 00 80 0C      <- header text 8
28 61 62 63 64 65 66 67 48  <- text 8 block #1
( a b c d e f g PA
68 0D 20 20 20 20 20 20 65  <- text 8 block #2
h CR SP SP SP SP SP SP PA

86 00 00 9E C0 80 0C      <- header text 9
28 61 62 63 64 65 33 76 3B  <- text 9 block #1
( a b c d e f g PA
68 69 0D 20 20 20 20 20 2C  <- text 9 block #2
h i CR SP SP SP SP SP PA

98 00 00 E0 00 80 0C      <- header text 10
28 61 62 63 64 65 66 67 48  <- text 10 block #1
( a b c d e f g PA
68 69 6A 0D 20 20 20 20 66  <- text 10 block #2
h i j CR SP SP SP SP PA

00 00      <- end (600 baud)
NU NU
```

Coding of text number in first 2 Bytes of the header

text	B 1	B 2
1	06	C0
2	18	C0
3	1E	00
4	60	C0
5	66	00
6	78	00
7	7E	C0
8	80	C0
9	86	00
10	98	00
11	9E	C0
12	E0	00
13	E6	C0
14	F8	C0
15	FE	00
16	00	E0

text	B 1	B 2
17	06	80
18	18	80
19	1E	E0
20	60	80
21	66	E0
22	78	E0
23	7E	80
24	80	80
25	86	E0
26	98	E0
27	9E	80
28	E0	E0
29	E6	80
30	F8	80
31	FE	E0
32	00	E1

text	B 1	B 2
33	06	03
34	18	03
35	1E	C3
36	60	03
37	66	C3
38	78	C3
39	7E	03
40	80	03
41	86	C3
42	98	C3
43	9E	03
44	E0	C3
45	E6	03
46	F8	03
47	FE	C3
48	00	E0

text	B 1	B 2
49	06	F8
50	18	F8
51	1E	E0
52	60	F8
53	66	E0
54	78	E0
55	7E	F8
56	80	F8
57	86	E0
58	98	E0
59	9E	F8
60	E0	E0
61	E6	F8
62	F8	F8
63	FE	E0
64	00	7E

structure of
1st byte code

text	B 1	B 2
65	06	0C
66	18	0C
67	1E	CC
68	60	0C
69	66	CC
70	78	CC
71	7E	0C
72	80	0C
73	86	CC
74	98	CC
75	9E	0C
76	E0	CC
77	E6	0C
78	F8	0C
79	FE	CC
80	00	86

text	B 1	B 2
81	06	E6
82	18	E6
83	1E	86
84	60	E6
85	66	86
86	78	86
87	7E	E6
88	80	E6
89	86	86
90	98	86
91	9E	E6
92	E0	86
93	E6	E6
94	F8	E6
95	FE	86
96	00	78

text	B 1	B 2
97	06	CF
98	18	CF
99	1E	0F

HEX	BIN
06	00000110
18	00011000
1E	00011110
60	01100000
66	01100110
78	01111000
7E	01111110
80	10000000
86	10000110
98	10011000
9E	10011110
E0	11100000
E6	11100110
F8	11111000
FE	11111110
00	00000000

Duplicates are marked read.