NEC NG-87243-001 LCD initialisation sequence

These were recovered from an NEC DTR-16D-1A phone manufactured around 2003 to 2006.
LCD controller compatible with Hitachi HD44780.

LCD 16 pin interface: led, led, led, RS, E, D0, D1, D2, D3, Vcc, Gnd, Contrast?, Vcc, -, Gnd, -

LED K	Green LED cathode
Common	LED anodes
LED K	Red LED cathode
RS	Register Select, low for commands, high for data
E	Enable, clocks in each 4 byte nibble on the trailing edge.
D0 - D3	Data lines. These are actually lines D4 through D7 as far as the HD44780 is concerned – the other 4 data-lines aren't brought out to the interface, and so communication is via 4 bit mode.
Vcc	3.3V, logic power supply.
GND	0V
Contrast	Varies from 1.7V to 1.9V, appears to be contrast control
?	Seems to be tied high
-	no connection
GND	Seems to be tied low, R/W maybe?
-	no connection

Once the controller is in 4 bit mode, bytes are sent to the controller as two successive nibbles, with two E clock pulses 27uS apart, most significant nibble first. E clock pulses are 12uS duration, data clocked in on the falling (trailing) edge. Generally there is a 39uS delay between nibbles forming one byte and a 72uS delay between successive command bytes.

LCD initialisation:

(each byte is a command, unless designated with a "d" as data, ie RS was high)
Typical initialisation sequence for Hitachi HD44780, starts off assuming it is unknown whether the chip is in 8 bit or 4 bit mode. The first 3 bytes are written with a single E clock pulse each (the MPU is only asserting 4 bits of data each time. It assumes the other 4 data-lines (which aren't brought out to the interface) are all tied low.

```
3   3   3   2 28 08 01 06 0C
   30         Function set interface to be 8 bits long
```

8mS delay

```
   30         Function set interface to be 8 bits long
```
If the chip happened to have been in 4 bit mode already, it has now received the command 33 which would also set it to 8 bit mode

194uS delay

```
   30         Function set interface to be 8 bits long
```
Now chip is definitely in 8 bit mode, set it to 4 bit mode!
```
   20         Function set interface to be 4 bits long
   28         Function set 4bits, 2 rows of character, 5x7 font
   08         Set display OFF, cursor OFF, blink OFF
   01         Clear display
```

06	Set increment
0C	Set display ON, cursor OFF, blink OFF

2S delay

Phone test sequence:

0C	Set display ON, cursor OFF, blink OFF
80	Set display RAM address 0

"TEST   PUSH=   "

54d	T
45d	E
53d	S
54d	T
20d	
20d	
20d	
20d	
20d	
50d	P
55d	U
53d	S
48d	H
3Dd	=
20d	
20d	
20d	
20d	
20d	
20d	
20d	
20d	
20d	
20d	

A0	Set display RAM address 32

"   NEXT=F12   "

20d	
20d	
20d	
20d	
20d	
20d	
20d	
20d	
20d	
4Ed	N
45d	E
58d	X
54d	T
3Dd	=
46d	F
31d	1
32d	2

```
20d
20d
20d
20d
20d
20d
20d

C0              Set display RAM address 64
"                        L16+B 0"
20d
20d
20d
20d
20d
20d
20d
20d
20d
20d
20d
20d
20d
20d
20d
20d
20d
4Cd        L
31d        1
36d        6
2Bd        +
42d        B
20d
30d        0
```

Store 8 programmable character patterns

```
40              Set character generator address 0
04d         . . . . . ▉ . .
0Ed         . . . . ▉▉▉ .
15d         . . ▉ . ▉ . ▉
04d         . . . . . ▉ . .
04d         . . . . . ▉ . .
04d         . . . . . ▉ . .
04d         . . . . . ▉ . .
00d         . . . . . . . .

04d         . . . . . ▉ . .
04d         . . . . . ▉ . .
04d         . . . . . ▉ . .
04d         . . . . . ▉ . .
15d         . . ▉ . ▉ . ▉
0Ed         . . . . ▉▉▉ .
04d         . . . . . ▉ . .
```

```
00d    . . . . . . . .

08d    . . . . ▓ . . .
04d    . . . . ▓ . . .
00d    . . . . . . . .
0Ed    . . . ▓▓▓ . .
11d    . . ▓ . . . ▓ .
1Fd    . . ▓▓▓▓▓ .
11d    . . ▓ . . . ▓ .
00d    . . . . . . . .

02d    . . . . . . ▓ .
04d    . . . . . ▓ . .
00d    . . . . . . . .
0Ed    . . . ▓▓▓ . .
11d    . . ▓ . . . ▓ .
1Fd    . . ▓▓▓▓▓ .
11d    . . ▓ . . . ▓ .
00d    . . . . . . . .

04d    . . . . ▓ . . .
0Ad    . . . ▓ . ▓ . .
00d    . . . . . . . .
0Ed    . . . ▓▓▓ . .
11d    . . ▓ . . . ▓ .
1Fd    . . ▓▓▓▓▓ .
11d    . . ▓ . . . ▓ .
00d    . . . . . . . .

05d    . . . . ▓ . ▓ .
0Ad    . . . ▓ . ▓ . .
00d    . . . . . . . .
0Ed    . . . ▓▓▓ . .
11d    . . ▓ . . . ▓ .
1Fd    . . ▓▓▓▓▓ .
11d    . . ▓ . . . ▓ .
00d    . . . . . . . .

04d    . . . . ▓ . . .
0Ad    . . . ▓ . ▓ . .
1Fd    . . ▓▓▓▓▓ .
10d    . . ▓ . . . . .
1Ed    . . ▓▓▓▓ . .
10d    . . ▓ . . . . .
1Fd    . . ▓▓▓▓▓ .
00d    . . . . . . . .

05d    . . . . ▓ . ▓ .
0Ad    . . . ▓ . ▓ . .
00d    . . . . . . . .
0Ed    . . . ▓▓▓ . .
11d    . . ▓ . . . ▓ .
11d    . . ▓ . . . ▓ .
1Ed    . . ▓▓▓▓ . .
```

```
00d        ........
```

After this point the display and programmable characters appear to be refreshed over and over while it waits for input (key-presses to test the phone):

```
0C          Set display ON, cursor OFF, blink OFF
80          Set display RAM address 0
```
"TEST"
```
54d         T
45d         E
53d         S
54d         T
```

```
...
```