NEC NG-87243-001 LCD initialisation sequence

v1.0 4 Feb 2021

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 : RS, E, D0, D1, D2, D3, Vcc, GND

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

There are also 3 pins for controlling the front panel red and green LEDs.

Once the controller is in 4 bit mode, bytes are sent to the controller as two successive nibbles, with two E clock pulses 39uS apart, most significant nibble first. Generally a 39uS delay between nibbles forming one byte and a 72uS delay between successive command bytes.

LCD initialisation:

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
```

3*0* Function set interface to be 8 bits long

```
8mS delay
```

3*0* 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
```

3*0* Function set interface to be 8 bits long

Now chip is definitely in 8 bit mode, set it to 4 bit mode!

2*0* 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
45d
53d
54d
20d
20d
20d
20d
20d
50d
55d
53d
48d
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
45d
58d
54d
3dd
46d
31d
32d
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
31d
36d
2bd
42d
20d
30d
```

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
45d
53d
54d

...
```