Nokia 3310 LCD

A simple project to test the usage of a Nokia 3310 LCD. A lot of hobby projects have been done based on it - I'm doing this to see if it's suitable to be used in our microprocessor system labs.

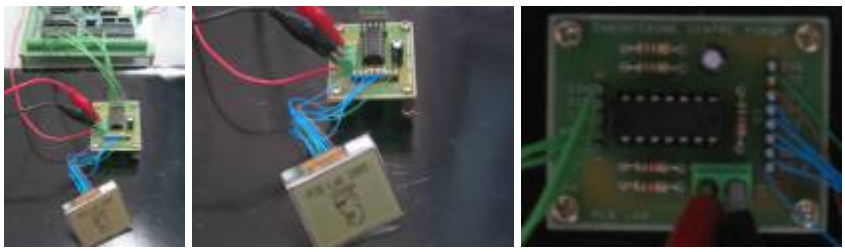
Background



I was looking for a graphical display unit when a friend of mine suggested me to use the LCD display used in mobile phones. To my surprise, I was able to get a Nokia 3310 monochrome display at only RM22! I bought it off a shop in Kuala Perlis. One thing about buying components like this is we don't have any specifications to start with. However, I was in luck when I discovered that this particular LCD has been used extensively by hobbyist! So, I downloaded the specifications for the LCD controller (which happens to be a Phillips LCD controller PCD8544) - you can get it here.

Setup

Like I've said before, the Nokia 3310 LCD is a nice small graphical LCD, suitable for a lot of various projects. It has 84 x 48 pixels resolution and the only external components that you need is a capacitor (1-10 uF) from VOUT to GND. Being used in handphones, we could imagine that it is a low-power device - operating at a voltage range of 2.7V to 3.3V. Having the Phillips LCD controller makes our lives much easier - we need to connect only 8 pins (two of which are, of course, VDD and GND).

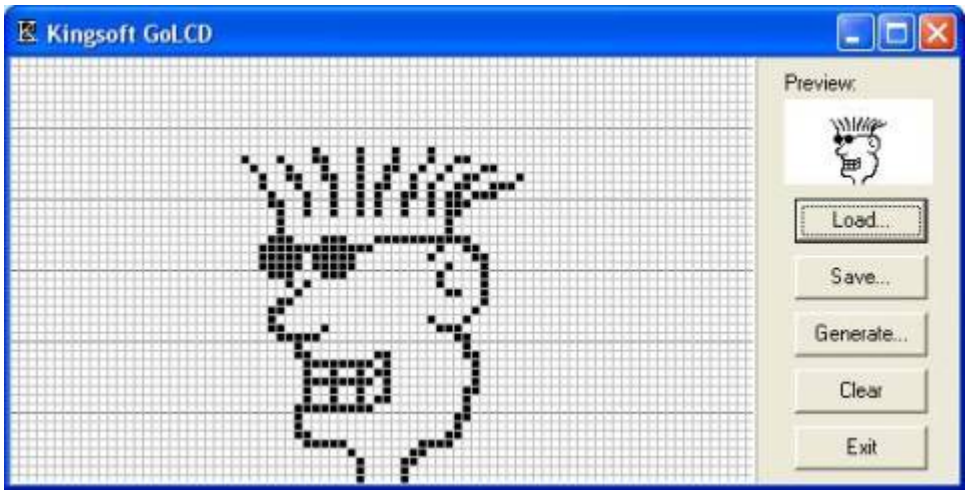


Out of the 8 pins, we need to 'control' only 5 pins (Shoot! This just gets better and better, ain't it? :p). There are a lot of sample projects that you can refer to. Some have used ATMEL's AVR controller and some have even used PC's parallel port. Since I'm quite eager to try this, I used whatever I have available to me - an 8085 trainer board that we use for undergraduate microprocessor course in KUKUM. For I/O interface, it has an 8255 chip that has been assigned a rollback address starting from 80H. I used the bit-set-reset (BSR) mode available on that chip.

I wrote this program in assembly language for 8085 system based on the Nokia LCD library available at MicroSyl.Com - you can also get the connections requirements there. Just to remind you, the Phillips LCD controller can serially receive data @ command at 4Mbit/second. So, in my case, the clocking for data transfer need not have any delays in its routine. Another thing that you might notice that I'm using a 5V supply for my 8085 trainer board. So, I need to have a converter for it to drive the LCD controller at 3.3V - for this purpose, we have fabricated a simple driver board based on an open-collector buffer to do just that - notice that small board in the image above. Big thanks to En. Halim for this!

Results

That 'simple' test program I wrote is basically an animation program (unzip and listen to the mpg file - you can actually hear Spin Doctor's 'Two Princes' in the background :p). There is a software called FastLCD that we can use to convert a bitmap to LCD data format. However, I found it a little bit unsuitable for my use since the program generates c-style data while I'm writing in assembly using C16 Cross Compiler. So, I wrote myself a program I called GoLCD to do that for me - it can generate both c-style and assembly-style data table, but ONLY FOR this particular LCD horizontal format. It also separates the data by labelling the data according to the banks (0-5). You can get GoLCD here. Below is a snapshot of GoLCD.



Update: GoLCD has been rewritten as an open source software - my1golcd.

What's Next

I could have got a more compact code if I compressed the frame data - but, like I've said, this is a simple test project. What's next? I'm in the process of putting together a compact board design based on a more advanced processor/controller (still can't decide which). I'm also considering the old 8086-based system since that processor is still available and relatively cheap. So, let's just wait and see :)

Finally, I would also like to thank En. Rahman for giving that valuable solution of taking the LCD out of its case and solder the connections directly to it. Kudos!

kaz20050408

From:
https://azman.my1matrix.cc/ - **Azman @MY1**

Permanent link:
https://azman.my1matrix.cc/doku.php?id=archive:nok3310lcd&rev=1770522381

Last update: **2026/02/08 03:46**