

INTERFACING NOKIA COLOUR LCD WITH AVR MICROCONTROLLER



■ ARUN DAYAL UDAI

Adding an elegant-looking colour LCD to a project is a dream for robotics enthusiasts. Usually, colour LCDs are costly and it's difficult to find the technical information for their interfacing.

Using the colour LCD of your old mobile phone could be a solution. Here we describe how to use

the colour LCD of a Nokia handset (model 6100, 7210, 6610, 7250 or 6220) with Philips PCF8833 chipset through ATmega2560 ATMEL AVR microcontroller. These LCDs are readily available and inexpensive even if you buy a new one. The author's prototype is shown in Fig. 1. With this project, you can easily load a colour picture of 132×132 pixels, in 12-bit RGB (red-green-blue) format, to the colour LCD

through your PC's serial port.

ATmega2560 microcontroller

The ATmega2560 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. The AVR core combines a rich instruction set with 32 general-purpose working registers. All the 32 registers are directly connected to the arithmetic logic unit, allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code-efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The microcontroller has 256 kB of in-system programmable Flash with read-while-write capabilities, 4kB EEPROM, 8kB SRAM, 86 general-purpose input/output (I/O) lines, real-time counter, six flexible timers/counters with compare modes and pulse-width modulation (PWM), four USARTs,

a byte-oriented two-wire serial interface, a 16-channel, 10-bit analogue-to-digital converter (ADC) with optional differential input stage with programmable gain, programmable watchdog timer with internal oscillator, a serial peripheral interface (SPI) port, IEEE standard 1149.1 compli-



Fig. 1: Author's prototype

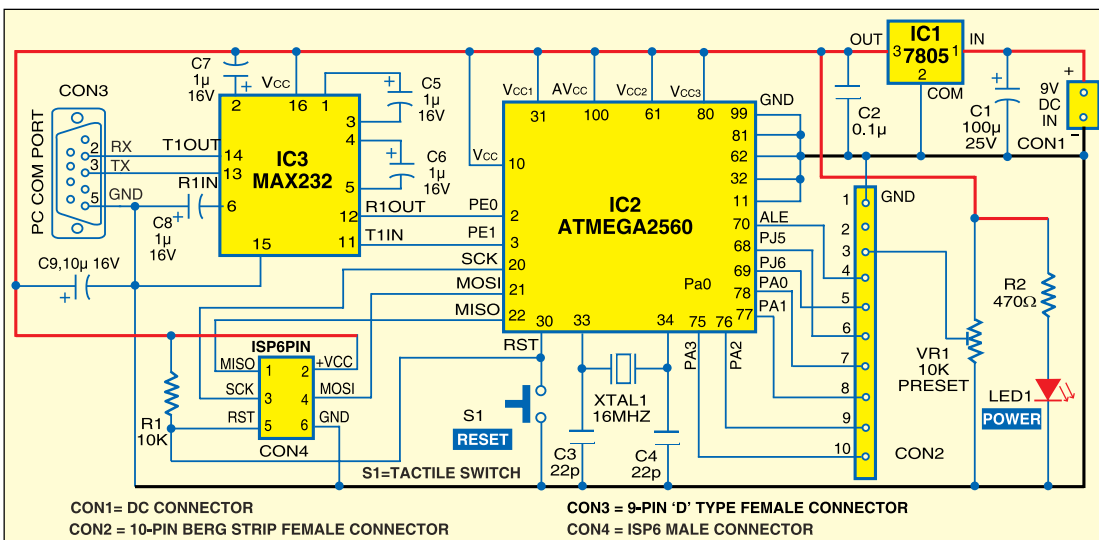


Fig. 2: Circuit for interfacing the Nokia colour LCD with AVR microcontroller

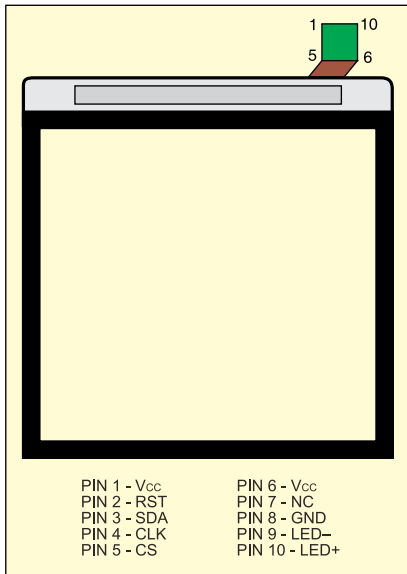


Fig. 3: Front view of Nokia LCD with pin details of the connector

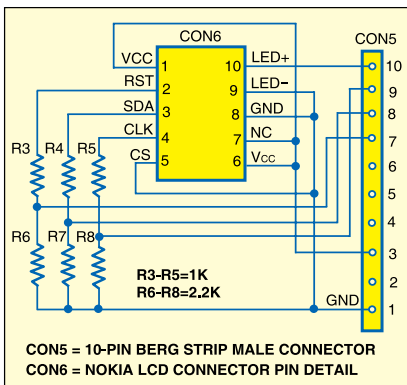


Fig. 4: Circuit for Nokia LCD connector extension

ant JTAG test interface (also used for accessing the on-chip debug system and programming) and six software-selectable power-saving modes.

PCF8833 LCD driver

The PCF8833 is a single-chip, low-power CMOS LCD controller driver, designed to drive colour super-twisted nematic displays of 132 rows and 132 RGB columns. All the necessary functions for the display are provided in a single chip, including the display RAM which has a capacity of 209 kbits. The PCF8833 uses multiple-row addressing technique in order to achieve the best optimal performance at the lowest power consumption. It offers two types of microcontroller interfaces: 8080 sys-

R	R	R	R	G	G	G	G	RED AND GREEN FOR 1ST PIXEL
B	B	B	B	R	R	R	R	BLUE FOR 1ST AND RED FOR 2ND PIXEL
G	G	G	G	B	B	B	B	GREEN AND BLUE FOR 2ND PIXEL

Fig. 5: Standard 12-bit image



Fig. 6: GUI created using VB.net for image transfer from the PC

tem interface and 3-line serial interface.

The PCF8833 communicates with the host using an 8-bit parallel interface or 3-line serial interface. Here a 3-line serial interface is implemented for communication between the microcontroller and the PCF8833 chip. The three lines are chip-select (CS) or enable pin, serial clock (SCLK) and serial data (SD).

Processing the instructions and data sent to the interface does not require the display clock. The display clock and interface clock are independent of each other. The display clock is derived from the built-in oscillator. The serial data pin of PCF8833 is connected to the SDA pin of the microcontroller.

Circuit description

Fig. 2 shows the circuit for interfacing the Nokia colour LCD with Atmega2560 microcontroller. The circuit is powered from a standard 9V DC source. The 9V DC is converted into 5V DC using a 7805 regulator (IC1). The regulated 5V supply powers the circuit including the Atmega2560, MAX232, LCD connector (CON2) and ISP6 connector (CON4). CON3 is a 9-pin, D-type COM port connector used to interface with the PC for picture file transfer to the LCD through the microcontroller (IC2). ISP6 connector is used for programming the microcontroller using STK500 board.

Whenever the program doesn't function properly, you can reset the microcontroller by momentarily pressing reset switch S1.

Pin details of Nokia's 10-pin LCD connector are shown in Fig. 3. A manual implementation of the SPI with the microcontroller is done through SDA, CLK and CS pins on the LCD connector. First, the clock pin is cleared, then the data pin is set or cleared depending on the data bit and the clock pin set. SDA and CLK pins of the LCD connector are connected to pins 77 (PA1) and 76 (PA2) of the microcontroller, respectively.

The 8-bit data is transferred through the SPI in eight clock cycles. CS pin of the LCD chip is permanently made low. Backlight LED pin for the LCD may be connected to a PWM pin to have a dimming effect depending on the duty cycle of PWM.

The LED and Reset pins of the LCD are connected to pins 75 (PA3) and 78 (PA0) of the microcontroller, respectively. When PA0 is made low, it resets the LCD. All the lines PA0 to PA3 is made an output pin by setting the data direction register (DDRA) bits high.

This connector is too small to handle and solder using a normal soldering iron. You can extend the connections to a standard-size berg strip male connector as shown in Fig. 4. The LCD connector extension may be developed with resistors R3 through R5 (1 kilo-ohm each) and R6 through R8 (2.2 kilo-ohms each) forming the voltage divider circuits to obtain 3.3V signals (acceptable by the colour LCD) from 5.5V TTL signals.

Note that this connector extension is made for use with a colour LCD as well as standard character LCD. The LCD receives 3.3V supply through preset VR1. VR1 is also used to adjust the brightness of the colour LCD. Contrast is adjusted through software program.

Make a PCB layout and glue the board on the back side of the LCD module. Now you can easily fit the LCD onto the 10-pin berg strip socket

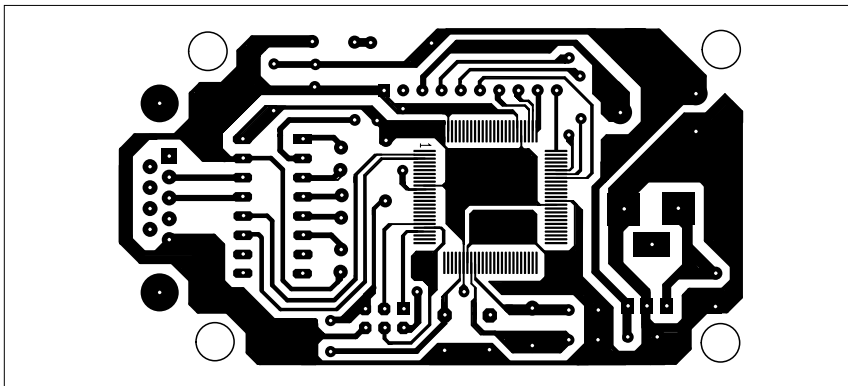


Fig. 7: An actual-size, single-side PCB for interfacing the Nokia colour LCD with the AVR microcontroller circuit

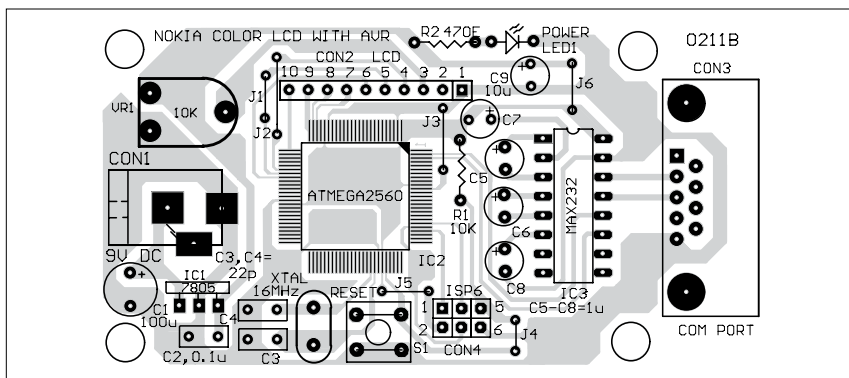


Fig. 8: Component layout for the PCB in Fig. 7

in your ATmega2560 application board.

Software program

The software for the microcontroller is written in 'C' language using the IAR Embedded Workbench integrated development environment. IAR Embedded Workbench is being developed by IAR Systems and ATMEL developers in parallel and hence it generates the optimised code which uses full 'C' coding capabilities of AVR devices. AVR development tools for embedded systems can be downloaded for free from IAR website www.iar.com. For details of IAR Embedded Workbench, you may refer to 'A Beginners' Guide to ATMEL AVR Development' article published in January issue of EFY.

To start the application, click 'IAR Embedded Workbench' icon in 'All Programs' menu of Windows. From the main menu, select Project→Create New Project and start an AVR Studio

compatible C project. Generate the Intel hex file for burning the code into Atmega2560 microcontroller as follows: Project→Options→Linker (under Category)→Extra Output→Generate Extra Output (to click)→Override Default (to click)→Type Project Name. hex→Output Format→intel-standard (to select).

Press 'ok' after all these settings are done. The source code for LCD interfacing (main.c) along with comments for instructions is included in this month's EFY-CD. Import the main.c file from the CD into the editor screen by clicking 'Add Files...' option in Project menu. Click 'Compile' option from Project menu to compile and generate main.hex code.

To send a command, a low bit is sent first before sending an 8-bit command, forming a 9-bit SPI command signal. Similarly, before sending an 8-bit data, a high bit is sent forming a 9-bit SPI data. You may refer to

PARTS LIST

Semiconductors:

IC1	- 7805, 5V regulator
IC2	- ATmega2560 microcontroller
IC3	- MAX232 RS-232 driver
LED1	- 5mm light-emitting diode

Resistors (all 1/4-watt, ±5% carbon):

R1	- 10-kilo-ohm
R2	- 470-ohm
R3-R5	- 1-kilo-ohm
R6-R8	- 2.2-kilo-ohm
VR1	- 10-kilo-ohm preset

Capacitors:

C1	- 100µF, 25V electrolytic
C2	- 0.1µF ceramic
C3, C4	- 22pF ceramic
C5-C8	- 1µF, 16V electrolytic
C9	- 10µF, 16V electrolytic

Miscellaneous:

CON1	- DC connector
CON2	- 10-pin berg strip female connector
CON3	- 9-pin D-type female connector
CON4	- ISP 6-pin male connector
CON5	- 10-pin berg strip male connector
CON6	- 10-pin connector for Nokia LCD
S1	- Tactile switch
	- Nokia 6100 colour LCD module

the datasheet of PCF8833 for various instructions used in the LCD driver chip. The code for this project (main.c) has been programmed as per these instructions.

If you can draw a point with a colour at any given location on the LCD screen, you can implement various graphics algorithms to draw a circle, line, filled rectangle, etc. You can have the AVR microcontroller fitted with a 16.00MHz external crystal oscillator, a serial RS-232 driver connecting a serial port and a reset arrangement to run the code.

Downloading the code into Atmega2560. To program the Atmega2560, connect the 6-wire cable between ISP connectors of the STK500 board (marked as ISP6) and the target board (marked as ISP con). Connect a serial cable from 'RS232 CTRL' connector on the STK500 board to a COM port on the PC. Now start AVR Studio 4.0 without opening any project file.

Optionally, you may proceed as follows: Main Menu→Tools→Program

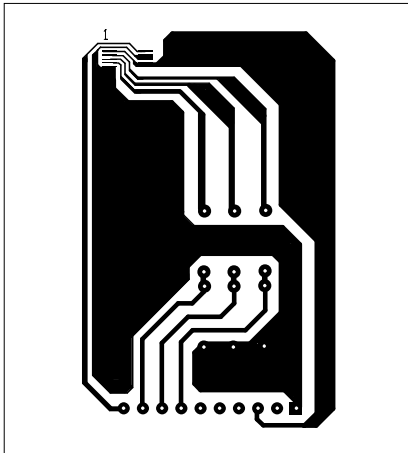


Fig. 9: An actual-size, single-side PCB for the LCD extension circuit

AVR→Select AVR Programmer. Press 'Connect...' after selecting STK500 or AVRISP in the platform window and COM port (say, COM1) in the port window. Next, select Atmega2560 as 'Device,' ISP as 'Programming Mode' and browse your project hex file from Project→Debug→Exe folder. Press 'Program' button on the STK500 dialogue box to burn the hex file into your microcontroller.

Converting the image file. You may convert any image file into a .raw file of 132×132 size, in 12-bit RGB colour, using Adobe Photoshop.

A standard 12-bit image encodes the two pixels in three bytes of data (24 bits) as shown in Fig. 5. In a 12-bit image, each pixel has three colour components of 4-bit value (0 to 15). So each of red, green and blue can vary from 0 to 15. Now, as a single COM port communication or any serial communication has 8-bit data and 12 bits are one-and-a-half byte, the two pixels are transferred so that $12 \times 2 = 24$ -bit value is sent in three successive transfers of 8-bit value. Same is the case when storing these pixels in a file or a memory which stores a byte value only. The two pixels having 24 bits can be stored in three bytes of data.

A front-end (user-interface) program for transferring a picture file to the AVR microcontroller may be designed using a suitable software.

Using GUI to transfer the image.

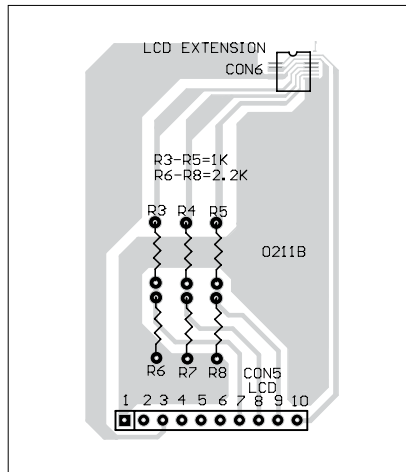


Fig. 10: Component layout for the PCB in Fig. 9

The user-interface program for transferring the picture file (.raw) from your PC to the Nokia LCD is designed in Visual Basic.NET platform. The send_picture.exe file is already included in the EFY-CD of this issue. When you click this file, you will see a GUI window as shown in Fig. 6.

In the first drop-down menu, select the appropriate com port. The image to be transferred should be saved in *.raw format. Select that image using 'Browse' option in the second pane and press 'Send' button to transfer the image to the graphical LCD. Now the image is displayed on the LCD.

Writing a text to the screen is similar to drawing a bitmap with predefined height, width and colour of characters.

Construction

An actual-size, single-side PCB for interfacing the Nokia colour LCD with the AVR microcontroller circuit (Fig. 2) is shown in Fig. 7 and its component layout in Fig. 8. An actual-size, single-side PCB for the LCD extension circuit (Fig. 4) is shown in Fig. 9 and its component layout in Fig. 10.

Mount all the components as shown in the PCB layouts. The Atmega2560 used in the prototype comes in a 100-pin, surface-mount TQFP package. By using surface-mount devices (SMDs) in your project, you don't have to drill holes in the PCB and the board size will be much smaller. But it may

be difficult to handle and solder SMDs without an SMD soldering station. However, with a normal soldering iron, tweezers and some basic knowledge, you can solder SMD components without much problem. You need good eyes, a steady hand and a soldering iron with a small, clean tip.

Make sure that the PCB is clean. You may use copper polish material to remove all kinds of oxidation and acetone (nail polish remover) to remove all kinds of contaminations on the PCB.

On the PCB, figure out the correct location for placing the SMDs. Fixate the SMD at the top right corner and the bottom left corner. Carefully solder SMD pins one by one. SMD components are very sensitive to heat, so allow your SMD to cool down after every step. For more tips on soldering SMD components, you may refer to <http://hem.passagen.se/communication/pcbsmd.html> and www.engadget.com/2006/03/07/how-to-make-a-surface-mount-soldering-iron/ websites.

After soldering the components on the PCBs, insert the LCD connector extension board in the main interfacing PCB using the connector marked 'CON2.' With the LCD screen facing towards you, the leftmost pin of the berg strip is pin 1. Make sure that the orientation of the LCD connector is correct.

Now connect 9V DC source to the circuit and switch on the power supply. LED1 should glow to indicate the presence of power in the circuit. At this point, you will see a test colour band drawn by itself on the LCD screen and then waits for serial port to get 12-bit RGB image from PC at 115.2 kbps baud rate. Now run the send_picture.exe file and send the picture file (.raw) to the LCD from your computer.

EFY note. The source codes of this article (main.c and Visual Basic.NET GUI program) have been included in this month's EFY-CD and are also available on efymag.com website. ●

The author is an assistant professor in Department of Mechanical Engineering at Birla Institute of Technology, Mesra, Ranchi