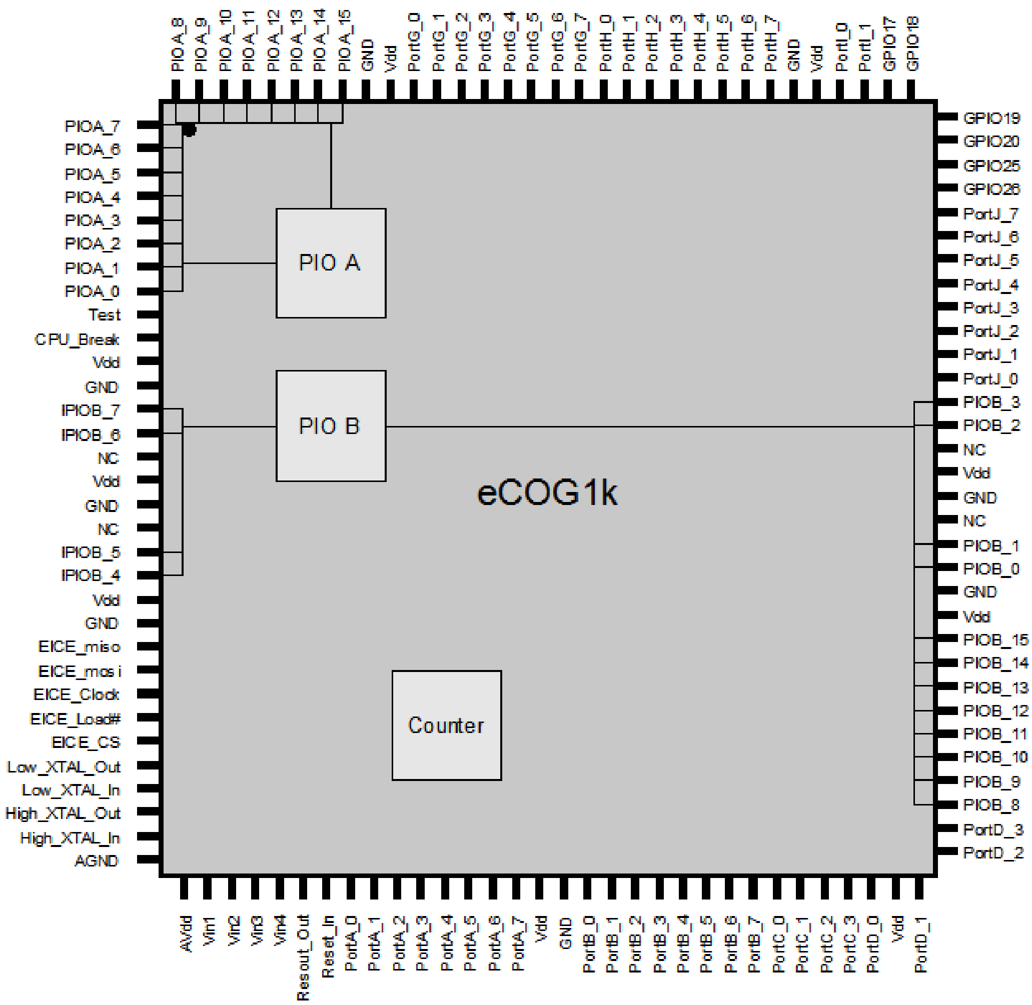
AN022 – Direct Drive Software for a Static LCD

Version 1.1

This application note describes a software driver for a static (non-multiplexed) LCD that runs on the eCOG1k microcontroller and requires no additional drive hardware.



Revision History

Version	Date	Notes
V1.0	08/04/2005	First release
V1.1	01/06/2006	Expanded with sample application to drive all 4 digits of the LCD

Contents

1 Introduction

Liquid crystal displays are useful in microcontroller applications for basic communication with the user. Normally, LCD drivers are implemented in hardware, either by inclusion of extra driver ICs or sometimes through dedicated on-chip hardware providing the necessary drive signals. Solutions using an external LCD driver IC are very flexible and powerful. On-chip solutions are usually targeted only to a particular type of LCD and can consume a large number of dedicated pins on the microcontroller.

This application note describes a software driver for a static (non-multiplexed) LCD that runs on the eCOG1k microcontroller and requires no additional drive hardware. An elapsed time minute/second counter application example is provided to demonstrate how a 4-digit 7-segment static display can be driven directly from the eCOG1k using parallel I/Os and one general purpose I/O on the development board. The driver is implemented using interrupt service routines and is configured to use minimal power. This is achieved using the eCOG1k sleep mode and minimal system clocks.

2 Glossary

A table of abbreviations used in this document.

eCOG1	Cyan Technology target micro controller
GPIO	General Purpose Input/Output
ISR	Interrupt Service Routine
LCD	Liquid Crystal Display
PIO	Parallel Input/Output

3 Driving Static LCDs

When driving LCDs, it is important to ensure that the DC component of the drive signal applied to the LCD is minimised. This is to avoid electrolysis of the liquid crystal materials, which would otherwise eventually result in the decay of the LCD and malfunction of the display. This requires an alternating voltage between the backplane(s) and the segments of the display with an average DC value of zero. The contrast of the LCD segment is related to the RMS voltage applied across the segment. Simple static LCDs have only one backplane electrode and a single electrode for each segment, resulting in eight connections for a single seven-segment display.

Driving static displays is accomplished simply by applying a clock signal input to the backplane (typically 32Hz) with a 50% duty cycle. A segment is then turned off by applying an in phase signal to the segment or turned on by applying a 180° phase shifted signal to the segment. This is described graphically in Figure 1. As power consumption in LCD displays is low, it is possible to drive the segment and backplane directly from GPIO lines of a typical microcontroller.

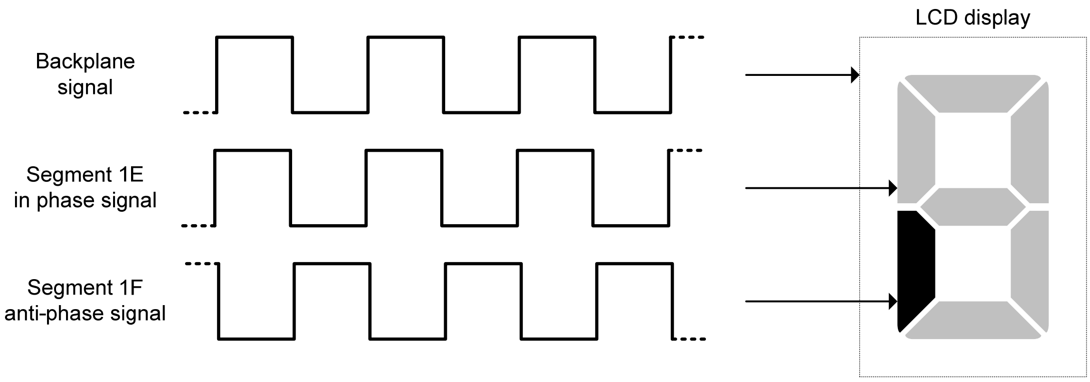


Figure 1. Static LCD drive.

4 Hardware

4.1 Seven Segment Display

The seven-segment display used in this application is the Varitronix VI402 static LCD. The device consists of a 4-digit 7-segment display with decimal points and colon, in a forty-pin package. This particular LCD is not multiplexed and has a single backplane connection.

4.2 eCOG1k Connections

The LCD input pins are connected directly to the outputs of the eCOG1k. All 32 segments of the LCD are driven by two 16-pit parallel I/O ports (PIO A and PIO B), while the backplane signal is driven by one general purpose I/O pin (GPIO 17). On the eCOG1k device, a further 22 GPIOs are freely available for other purposes. Figure 2 illustrates the physical connections between the processor and the display unit.

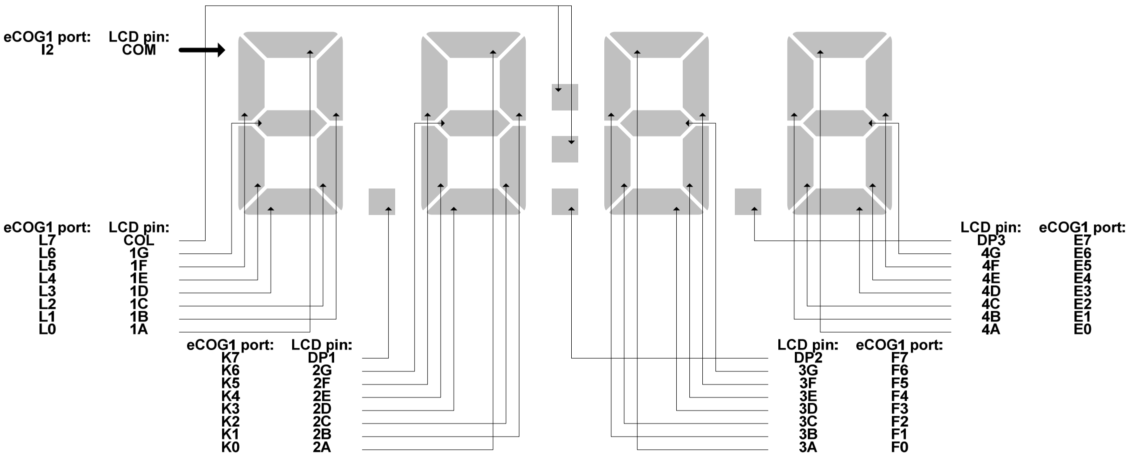


Figure 2. eCOG1k to VI402 static LCD physical connections.

5 Example Software

5.1 Overview

The example software is designed to drive the four digits and colon part of the display. The software uses the minimum number of peripherals and the eCOG1k sleep mode to obtain minimum power consumption. An elapsed time minute/second counter is displayed on the LCD in the format MM:SS. The counter continuously increments at a one-second interval, looping back to zero when an hour has elapsed.

5.2 Software Operation

The software is designed to run entirely from interrupts and at a low processor clock speed of 16 kHz to minimise power consumption. The high frequency oscillator, high frequency PLL and low frequency PLL are disabled during initialisation. Following initialisation, the processor is put to sleep and wake up occurs only on timer interrupts. When the interrupt service routine finishes executing, the processor returns back to the sleep state. Using this configuration the processor consumes approximately 140 µA during operation.

Ports L and K are configured as PIO B and used to drive the minutes display (and the colon) on LCD digits 1 and 2 respectively. Ports F and E are configured as PIO A and used to drive the seconds display on LCD digits 3 and 4 respectively. The LCD backplane signal is driven by Port_I I2, which is configured as GPIO 17. A timer interrupt is initialised to run at 64 Hz to generate a 32 Hz LCD frame rate, while counter 1 is initialised to generate a 1 Hz interrupt to update the LCD display value.

The counter 1 ISR simply increments and keeps track of the elapsed time counter. The operation of the timer ISR is slightly more complex. It is important to ensure a true 50% duty cycle on the LCD inputs to remove any DC component. However, differences in the code execution time could produce fluctuations in the time required to generate the LCD waveform. This in turn could disturb the 50% duty cycle, resulting in a net DC voltage component across the LCD segments. In order to avoid this, the output digit display values are updated with data generated by the previous interrupt, before calculating the output signals for the next one.

When a timer interrupt occurs, the stored digit display values and backplane signal are output. The encoded digit display values for the next interrupt are then generated and stored. The backplane signal is toggled on every interrupt.

6 Further Enhancements

More complicated LCDs, with a large number of segments are often manufactured with more than one backplane. Traditionally this requires more complex LCD driving waveforms with each signal being driven to different levels. Multiplexed LCDs can be driven in a similar way to the static LCD described in this application note, but the details are beyond the scope of this document.