

Laboratory Exercise 1

Getting Started with Linux

This is an introductory exercise in using a Linux operating system on Intel's Cyclone V SoC devices. The exercise uses the DE1-SoC development and education board, which is described on the Intel FPGA University Program website. Linux runs on the ARM processor that is part of the Cyclone V SoC device. During the operating system boot process, Linux programs the Cyclone V FPGA with the *DE1-SoC Computer* system. This system instantiates components inside the FPGA to make it easy to use the peripherals built into the DE1-SoC board, such as the LEDs, seven-segment displays, switches, and pushbuttons. For a detailed description of this computer system, please refer to the document *DE1-SoC Computer System with ARM*.

Part I

Read and complete Sections 1, 2, and 3 of the tutorial *Using Linux on the DE1-SoC*. These sections will guide you through setting up the Linux microSD card, running the Linux OS on the DE1-SoC board, and communicating with the board from a host computer.

Part II

Section 3.3 of the *Using Linux on the DE1-SoC* tutorial involves implementing a program that increments the LEDR lights on the DE1-SoC board. In this part you are to write another program that controls the red lights in a different manner, described below.

Your program should turn on one LEDR light at a time. First, the rightmost light $LEDR_0$ should be on, then $LEDR_1$, then $LEDR_2$, and so on. When you get to the leftmost light $LEDR_9$, the direction should be reversed. Only one LEDR light is ever on at one time. The effect should be a single light sweeping from right-to-left, then left-to-right, and so on. Use a delay so that the light moves at some reasonable speed. To implement the delay, you can use a Linux system function, such as `nanosleep(...)`. Documentation for the `nanosleep` function can be found by searching for it on the Internet.

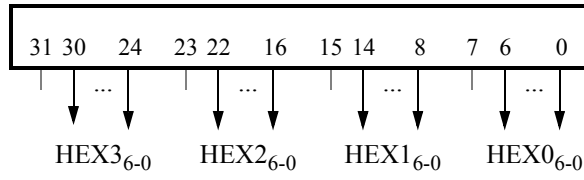
Part III

Section 3.4 of the tutorial *Using Linux on the DE1-SoC* shows how to implement a device driver using interrupts for the KEY pushbuttons on the DE1-SoC board. In this part you are to extend the functionality of that device driver.

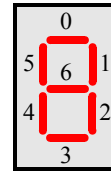
The existing KEY device driver initializes the LEDR port to $(1000000000)_2$, so that the leftmost light is on, and then increments this value whenever a KEY is pressed. You are to augment this code to display the count of pushbutton presses on the seven-segment display *HEX0*. You only need to display the decimal value corresponding to the four least-significant bits of the count, and wrap around to 0 when it reaches 9. Always leave the leftmost LEDR light set to 1 as a visual indicator that the device driver is present in the kernel. The count values displayed on the LEDR port should cycle through 1000000000, 1000000001, 1000000010, ..., 100001001, 1000000000, etc., and the corresponding digits displayed on *HEX0* should be 0, 1, 2, ..., 9, 0, respectively. A diagram of the seven-segment display ports in the DE1-SoC Computer is given below.

Address

0xFF200020

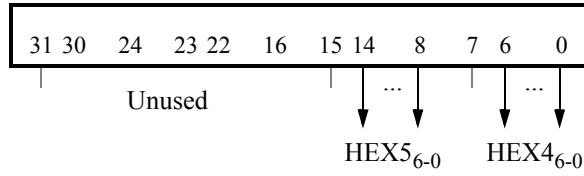


Data register



Segments

0xFF200030



Data register

Copyright © Intel Corporation.