Lab Session 5

Peripheral programming

Up to now we have been focusing on the CPU part of the ARM microprocessor. As we know, the chip used in the Beaglebone is a System On a Chip (SOC) and has additional logic units apart from the CPU. One of these units is a hardware timer and is an example of a relatively simple logic unit that can take a lot of computational load off the CPU. Instead of using 100% CPU time to time things to great precision, we can use a Time/Counter unit that is under the control of a CPU.

For this lab we will be writing C code to manage a Timer unit. The example chosen is a simple timing operation where we will time the interval between successive key presses on a keyboard. But before diving into the actual coding, please have a look at the DMTimer section in the AM335X Technical Reference Manual. This document explains the functionality of the SOC parts and is about 5000 pages ! The DMTimer section starts on page 4436. You will notice that there are actually several timer units and they can perform different types of tasks. **In your report**, please provide a brief summary of the features on the DMTimer unit.

Import the "EEEN301 Lab5 Student" project into your workspace and take a look at the various files. Lab5.c contains 'main' and you will notice that most of the code has been provided for you. **Your task** is to fill in the blanks "your code here". Now, fortunately for you, we have a set of C functions that perform the typical tasks we want to do with the timer such as setting things up and starting and stopping. These can be found in dmtimer.c. A set of functions like these is often referred to as an Application Programmers Interface (API) or Driver. These are often provided by manufacturers to help speed up your development.

Before launching into writing code, I suggest that you use your earlier UART program to test that you have your hardware and terminal program configured correctly.

Question: What is base frequency of the Timer and where is it sourced from?