universidade de aveiro theoria poiesis praxis

Departamento de Eletrónica, Telecomunicações e Informática
Ano letivo 2019/20

41489 – Sistemas de Instrumentação Eletrónica / Electronic Instrumentation Systems

Digital Data Aquisition System

Objective

This work aims at developing a digital data acquisition system, using the chipKIT Max32 platform, from Digilent. This system should acquire (read) an analogue signal produced by a bench top signal generator and produce an analogue output, corresponding to some processing on the input signal.

Non-functional specifications

- The input signal will be acquired by using the PIC32 ADC peripheral.
- The analogue inputs will be protected against over-voltage.
- The analogue output will be generated by means of a PWM signal and a low-pass filter.
- The filter that implements the relationship from input to output should be defined as a C function that can be adapted as needed.
- The system sampling frequency should be programmable. “Being programmable” means that the code should contain some section (e.g., a code segment, a value definition or a function call) that clearly defines the sampling frequency in Hz.

Functional specifications

- The input signal voltage will be in the range [0 V, 3.3 V] (maximal range).
- The system sampling frequency will be in the range [100 Hz, 500 Hz].
- The base frequency of the PWM signal will be 2 000 Hz and the resolution will be no less that 256 steps.

Methodology proposal

The following methodology may help you when developing the system. It is taken for granted that the basic issues related to programming the chipKIT platform (installing the IDE and compiler, downloading and running code in the platform, ...) are solved.

1. Choose one of the available timers. Start by programming the timer to count at a slow rate, that you can check by watching one of the LEDs LD4 or LD5 going on and off. The timer should be working by polling, not interrupt (check the status of the interrupt flag in the main program and reset it when the counter reaches max count). Modify the counting rate and check that results are as expected (if you double the count, the frequency should reduce to half, and so on). This will give some grip on how to program the timer frequency;

2. Create a function that programs the required register values that control the timer. This will encapsulate all the register programming in a code block, and you will only have to provide the necessary input;

3. Program the ADC converter to read the input channel and output the corresponding voltage. Do it firstly by polling (no interrupts), at a slow rate (e.g. once per second) and output the voltage value read from the input (0...3.3V) through the USART. Verify if it operates as desired by using diverse input voltage levels (e.g. through a potentiometer). The successful completion of this step assures a correct signal acquisition;

4. Join the timer and ADC input functions. Verify the correct operation. The successful completion of this step assures a correct signal acquisition at the required rate;

5. Create a function that sets the PWM module for generating a PWM signal with a frequency of 2 000 Hz. Create another function that accepts as input a Duty-cycle value. The PWM configuration should allow at least 256 steps for the Duty-Cycle;

6. Test the functions of Step 5 with several Duty-Cycle values. For that, examine the PWM signal with the oscilloscope. The successful completion of this step assures that a correct output signal is generated;

7. Associate the ADC value (Step 4) with the Duty-cycle function (Step 6). The system should be working properly.