

Laboratorio de microprocesadores

Trabajo Práctico Integrador 2022

- 1) Crear un repositorio remoto de Git llamado lm2022-<nombre>-<apellido>, por ejemplo: lm2022-federico-roux. El repositorio debe tener una rama principal denominada “develop”. Cada uno de los siguientes ejercicios debe ser resuelto en el mismo repositorio (organizar las carpetas de una manera entendible) y desarrollado en una rama individual. Una vez terminados, serán mergeados a la rama “develop” mediante el mecanismo Pull Request.
- 2) Escribir un script de Python que permita generar un archivo llamado <sgn.h>, que contenga 64 valores enteros de una señal tipo rampa, entre 0 y 3300, distribuidos uniformemente. Los valores deben estar separados por coma para poder ser incluido como header de C/C++, por ejemplo:

sgn.h:

```
0, 52, 105, 156, ...
```

En C/C++ se puede utilizar de la siguiente manera (revisar que esto se cumpla en un programa de ejemplo en C++):

```
uint16_t sgn[] = {  
#include <sgn.h>  
}
```

- 3) Realizar una máquina de estados (desarrollada como una clase C++) que tenga dos estados, ACUMULAR y PROMEDIAR.
 - a) En el estado ACUMULAR se debe sumar una muestra de la señal generada en el punto anterior a un acumulador e incrementar un contador. Al cumplirse N llamados, debe pasar al estado PROMEDIAR.
 - b) En el estado PROMEDIAR, debe imprimir en pantalla el promedio de las N muestras acumuladas y regresar al estado ACUMULAR, reiniciando el acumulador en cero.

Escribir un programa main.cpp que pruebe el correcto funcionamiento de la máquina de estados.

- 4) Utilizando como base el programa en freeRTOS visto en clase (MAIN_BLINKY), crear una tarea nueva llamada “**adc_task**”, y un semáforo binario denominado “**adc_sem**”. La tarea **adc_task** debe estar bloqueada por el semáforo **adc_sem**. Una vez que se desbloquee, debe llamar a la máquina de estados desarrollada en el punto anterior. El semáforo debe desbloquearse en el callback del timer.