

## TA134 – Taller de Sistemas Embebidos – Ingeniería Electrónica – FIUBA

### Guía de Trabajos Prácticos

#### TPF – Grupo 07

Implementar un **Sistema de Control de Portón Levadizo** de Garage (**Producto Mínimo Viable**), que cuente con **dos modos de operación**, en los que:

- **Monitoree y Controle** la/el **apertura/cierre** del Portón, en **forma segura** (modo **Normal**)
- Permita **leer y modificar** los **parámetros** de operación del **Sistema de Control** (modo **Set Up**)

#### Formalismos:

- El **grupo** deberá:
  - **Elegir** una fecha de **Evaluación Integradora** e inscribirse
  - **Completar** con antelación a la fecha elegida, un **cuestionario** en el Campus del curso
  - **Presentarse** en la fecha elegida con el **PMV armado y funcionando**, así como con los **Entregables** subidos al **repositorio** correspondiente
  - **Responder** preguntas de los docentes sobre la implementación

#### Parámetros de Operación:

- **Tiempo de espera** para **cerrar** (automáticamente), el Portón [**Segundos**]
- **Tiempo de permanencia** debajo del Portón [**Segundos**]

#### Sensores:

- Portón **Abierto** (use **un Dip Switch**)
  - Para **detectar** Portón **completamente abierto**
- Portón **Cerrado** (use **un Dip Switch**)
  - Para **detectar** Portón **completamente cerrado**
- Barrera **Infrarroja** (use **dos Dip Switch**)
  - Para **detectar** vehículos **debajo** del Portón
- **Abrir/Detener/Cerrar** Portón (use **un Pulsador**)
  - Para **ordenar** apertura/detención/cierre del Portón
- **Configuración/Enter/Next/Escape** (use **dos Pulsadores**)
  - Para **cambiar** de modo **Normal** a modo **Set UP** y **configurar** **parámetros** de operación del **Sistema de Control**
- **Sensores de Temperatura** (use **un LM35** y el **interno** del uC)
  - Para **medir** la temperatura ambiente y del uC

#### Actuadores:

- **Motor** (use **dos Leds**)
  - Un led **titilando**, indica que el motor gira abriendo el Portón
    - Sino estará **apagado**
  - Un led **titilando**, indica que el motor gira cerrando el Portón
    - Sino estará **apagado**
- **Semáforo** de Garage con **Sirena** (use **dos Leds** y **un Buzzer** activo – auto oscilante)
  - Un led **encendido**, indica que el Portón está completamente cerrado
    - Sino estará **apagado**
  - Un led **encendido**, indica que el Portón no está completamente cerrado
    - Sino estará **apagado**

- Una sirena **ululando**, indica que excedió el tiempo de permanencia debajo del Portón
  - Sino estará **apagada**
- **LCD Display** (use **uno de 2 líneas x16 caracteres** – en modo **4 bits**)
  - Para **mostrar** el **estado** del Sistema, de sus **parámetros** de operación y **temperaturas**
    - Cuando el **Sistema de Control** está en operación (**Normal**)
  - Para **mostrar** menús de **configuración**
    - Cuando el **Sistema de Control** está en configuración (**Set Up**)

#### Entregables:

- **Repositorio** GitHub conteniendo:
  - **Descripción** del **comportamiento** del **Sistema de Control** (PDF)
  - **Esquema** eléctrico y vista del **cableado**, como los del libro del TP3 (PDF)
  - **Calculo y Medición** de consumo eléctrico (PDF)
  - **Listado** de los ítems implementados y no implementados (PDF)
  - **Medición** de **WCET** (Worst-case execution time en **microsegundos**) de cada tarea y **Cálculo** del **Factor de Uso** de CPU (PDF)
- **Proyecto STM32** para placa **NUCLEO-F103RB**, con las siguientes **tareas/bibliotecas/archivos**:
  - **task\_sensor** (task\_sensor\_attributes)
    - **Descripción** de la tarea (txt) => **Dip Switches & Pulsadores**
    - **Modelo** Itemis Create de la tarea para un solo sensor (.scm & .PNG)
  - **task\_temperature** (task\_adc\_attributes)
    - **Descripción** de la tarea (txt) => **ADC**
  - **task\_set\_up** (task\_set\_up\_attributes, task\_set\_up\_inteface)
    - **Descripción** de la tarea (txt) => **Parámetros de Operación & LCD Display**
    - **Modelo** Itemis Create de la tarea (.scm & .PNG)
  - **task\_normal** (task\_normal\_attributes, task\_normal\_inteface)
    - **Descripción** de la tarea (txt) => **Monitoreo & Control** del **Portón**
    - **Modelo** Itemis Create de la tarea (.scm & .PNG)
  - **task\_actuator** (task\_actuator\_attributes, task\_actuator\_inteface)
    - **Descripción** de la tarea (txt) => **Leds & Buzzer**
    - **Modelo** Itemis Create de la tarea para un solo actuador (.scm & .PNG)
  - **display**
    - **Descripción** de la biblioteca (txt)
  - **Actualizar** archivos:
    - **Descripción** del proyecto (**readme.txt**)
    - **Descripción** de la aplicación (**app.txt**)

**IMPORTANTE:** Se pretende una implementación compatible con lo visto desde la primera clase del curso.

Es decir, una solución del tipo **Ejecutor Cíclico de Tareas**, de **Estructurada**, **Modular** (Escrutar, Procesar, Actuar), **Bare Metal** (sin Sistema Operativo), **Super-Loop** (polling & interrupts, código no-bloqueante), con **modelado de tareas** (diagrama de estado), del tipo **Event-Triggered System**.