



## PdM - Práctica 2

### Ejercicio 2

#### Resumen

Implementar mediante el modelo de Máquina de Estados Finitos (MEF) el antirrebote de teclas y la lógica de funcionamiento de un semáforo.

#### Condiciones de funcionamiento

Dependencias externas:

- `arduino.h`
- Usar `delay` no bloqueantes

#### Objetivos

1. Implementar un módulo de software en los archivos `teclas.c` y `teclas.h`, que encapsule el funcionamiento de dos teclas (pulsadores).

Dentro del módulo se debe implementar:

- a. Una MEF que resuelva el antirrebote por software con dos funciones asociadas, una para inicializar la MEF y otra para actualizar el estado de la MEF.
- b. Los estados de la MEF se deben implementar con un tipo de dato enumerativo (`enum`).
- c. Una función `buttonPressed(int16_t tecla)` que resuelva las acciones correspondientes al flanco descendente.  
La función puede aceptar un argumento que indique qué tecla se presionó.
- d. Una función `buttonReleased(int16_t tecla)` que resuelva las acciones correspondientes al flanco ascendente.  
La función puede aceptar un argumento que indique qué tecla se liberó.

2. Implementar una única MEF que permita controlar las dos teclas de forma independiente. La función de actualización de la MEF debe recibir como argumento **por referencia** una estructura con los siguientes campos:

- a. Nombre de la tecla
- b. *Delay* no bloqueante asociado a la tecla en particular
- c. Estado de la MEF asociado a esa tecla en particular



Por ejemplo:

```
actualizarMEF(dbn_t *dataTecla1); //prototipo de función para actualizar MEF
```

```
typedef struct { // estructura para controlar el antirrebote de una tecla
    int16_t tecla;
    int16_t delay;
    estadoMEF estado
} dbn_t;
```

3. Implementar un controlador de semáforos dentro de un módulo de software con archivos **semaforo.c** y **semaforo.h**.
  - a. El controlador se debe implementar como una MEF con dos funciones asociadas, una para inicializar la MEF y otra para actualizar el estado de la MEF.
  - b. Los estados de la MEF se deben implementar con un tipo de dato enumerativo (`enum`).
  - c. El semáforo debe tener los siguientes modos de operación:
    - i. **Normal**. Secuencia de leds: verde 3 s, verde y amarillo 500 ms, rojo 2 s, verde nuevamente....
    - ii. **Desconectado**. Led amarillo intermitente cada 500 ms.
    - iii. **Alarma**. Led rojo intermitente cada 1 s.
  - d. alguna de las dos teclas (pulsadores) debe ser capaz de cambiar la base de tiempo de la condición de operación normal (0.5x, 1x, 2x).
  - e. alguna de las dos teclas (pulsadores) debe permitir cambiar el modo de operación del semáforo a normal. Con cada flanco ascendente se debe ciclar el modo en el siguiente orden: normal, desconectado, alarma, normal, etc...
4. Usar la comunicación UART en un módulo de software con archivos **uart.c** y **uart.h**. El módulo debe implementar funciones para informar:
  - Cambios en la MEF antirrebote (qué tecla y en qué estado está la MEF).
  - Cambios en la MEF del controlador de semáforos (modo de funcionamiento)

## Fecha de entrega:

**15 de junio de 2023**

¡Éxitos en esta actividad!