TAREA # 3

MICROPROCESDORES Y MICROCONTROLADORES

SAMUEL PEÑA MORONTA 20170570-10131492

```
//============================================================================
/* SAMUEL P. MORONTA
   20170570-10131492

   TAREA            : # 3

   PLANTEAMIENTO    : Crear un programa que contenga varias secuencias de conteo, donde
                      mismo sea capaz de crear un corrimiento de los pines desde o a 15
                      y de 15 a 0 para la primera secuencia. Para la segunda secuencia
                      el programa debe de mover los pines del centro hacia los laterales
                      y de los laterales hacia el centro, es decir [15-8 || 8-0]

   Objetivo:          Comprender de forma mas especifica el funcionamiento de los
                      de los perifericos, los registros de control de relog, las entradas
                      y salidad de los de proposito general (GPIO).

*/
//============================================================================
#include "stm32f10x.h"

/*Prototipo de funcion*/
void readReg(void);
void delay(void);
void moveIz(void);
void moveDr(void);
void moveCenter(void);
//============================================================================
    /*Arreglo de numeros para mover los pines desde el centro
      hacia los laterales y de los laterales hacia el centro
    */
    int arrNum[8] = {
        0b0000000110000000,
        0b0000001001000000,
        0b0000010000100000,
        0b0000100000010000,
        0b0001000000001000,
        0b0010000000000100,
        0b0100000000000010,
        0b1000000000000001,
    };
//============================================================================
// main function
//============================================================================
int main(void)
{
    /*Declaracion de variables necesarias*/
    volatile uint32_t dly;
    unsigned int typePin = 0;

    /*Llamado para lectura de los registros para entada y salida*/
    readReg();
    /*Si es '1' = true, me mantengo en el bloque de codigo*/
    while(1) {
        /*Sin el pin es '1', el tipo de pin que tengo pulsado es 'typePin 1'*/
        if(GPIOB -> IDR & 0x00000001) {
            typePin = 1;//2
            /*Bucle do-while para que el bloque de instrucciones se ejecute al menos una vez*/
```

```
        do{
            /*Llamado de las funciones para corrimiento de derecha e izquieda, primera secuencia*/
            moveIz();
            moveDr();
            /*Si tengo '1', es decir ya he pulsado por segunda vez el pin, el programa no realizará
              nada, hasta cuando entre a la condicion de que este sea diferente de pin '1' y sera mi
              segunda secuencia. Desde el registro B, accedo al registro de entrada de datos.
            */
            if(GPIOB -> IDR & 0x00000001){

                if(typePin == 2) {
                    typePin = 3;
                }
            }else{
                if(!(GPIOB-> IDR & 0x00000001)) {
                    typePin = 2;
                }
            }
            /*El bloque de instrucciones se va a ejecutar mientras tenga un '1' o
              si ya he presionado por segunda vez el pin, para que no haga nada
             */
        }while(typePin == 1 || typePin == 2);
    }
    /*Entrara en el do-while, siempre y cuando se cumpla la condicion de que
      he pulsado el pin 3 veces, por lo que entrara a la segunda secuencia
    */
    if(typePin == 3 || typePin == 4){
        do{
            /*Muevo del centro hacia los laterales y de los laterales hacia el centro*/
            moveCenter();
            /* Aqui se vuelve a repetir la logica de los selecciones de pin,
                para el asunto de cambio de secuancia
            */
            if(GPIOB->IDR & 0x00000001){
                if(typePin == 4) {
                    typePin = 1;
                    GPIOA ->ODR = 0b1000000000000001;
                }
            }
            else{
                if(!(GPIOB -> IDR & 0x00000001)) {
                typePin = 4;
                }
            }
            /*El bloque de instrucciones se va a ejecutar mientras tenga un '1' o
              si ya he presionado por tercera vez el pin, para que no haga nada
             */
        }while(typePin == 3 || typePin == 4);
    }
}
/*
    Funcion  : redReg
    Objetivo : preparar los registros RCC (Registros de control de relog)
               Configutando el puerto A como salida  a 2 Mghz, tanto de
               entrada alto como bajo, dado que se utilizaran todos los pines
    Parametro: Funcion Void sin recibir parametros
    Retorno  : no retorna(void)
*/
void readReg(){
    RCC->APB2ENR |= RCC_APB2ENR_IOPAEN;
    RCC->APB2ENR |= RCC_APB2ENR_IOPBEN;
```

```
    GPIOB->CRL = 0x4;
    GPIOA->CRL = 0x22222222; //2
    GPIOA->CRH = 0x22222222;
    /*Registro de lectura*/
    GPIOA->ODR = 0b1;
}
/*
    Funcion  : delay
    Objetivo : Crear un retrazo para los corrimientos de los pines
    Parametro: Funcion Void sin recibir parametros
    Retorno  : no retorna(void)
*/
void delay(){
    volatile uint32_t dly;
    for(dly = 0; dly < 75000; dly++);
}
/*
    Funcion  : movIz
    Objetivo : Corrimiento de los pines hacia la izquieda
    Parametro: Funcion Void sin recibir parametros
    Retorno  : no retorna(void)
*/
void moveIz(){
    for(int i = 0; i<15; i++) {
        GPIOA ->ODR = GPIOA->ODR<<1;
        delay();
    }
}
/*
    Funcion  : movIz
    Objetivo : Corrimiento de los pines hacia la derecha
    Parametro: Funcion Void sin recibir parametros
    Retorno  : no retorna(void)
*/
void moveDr(){
    for(int i = 0; i < 15; i++) {
        GPIOA ->ODR = GPIOA->ODR>>1;
        delay();
     }
}
/*
    Funcion  : movIz
    Objetivo : Mover los pines del centro hacia los laterales
               Y de los laterales hacia el centro
    Parametro: Funcion Void sin recibir parametros
    Retorno  : no retorna(void)
*/
void moveCenter(){

    for(int i = 0; i < 7; i++) {
        /*En cada corrida del bucle voy guardando el
          elemento correspondiente en el arreglo de numeros
        */
        GPIOA->ODR = arrNum[i];
        delay();
    }
    for(int j = 7; j>=0; j--) {
        GPIOA->ODR = arrNum[j];
        delay();
    }
}
```