

LABORATORIO DE MICROCONTROLADORES

Reporte de la práctica 5

Profesor Matías Vázquez Piñón

Equipo 3

Estefanía Oseguera Salazar A01351832

Leslye Vanessa Franco Aguilera A01235714

19 de abril del 2022

Introducción

En la presente práctica se trabaja en el desarrollo del firmware para la simulación del principio básico del juego Whack-a-mole, un juego arcade en el que los jugadores usan un mazo para golpear topos de juguete que aparecen en ubicaciones aleatorias para enviarlos de vuelta a sus escondites. Al hacer la relación entre los elementos de este juego y el hardware empleado en esta práctica, se hizo uso de elementos como la placa Curiosity y la PCB elaborada en las primeras prácticas, en donde cada topo es simulado por un led que enciende al momento de *salir de su escondite* y el mazo es simulado por un botón que, al ser presionado *golpea al topo*.

El desarrollo del firmware para esta aplicación contiene varias secciones:

- **Libraries:** Donde incluimos los recursos a usar en el programa.

```
#include "device_config.h"
#include <math.h>
#include <xc.h>
```

- **Directives:** Aquí se definió la frecuencia del cristal, que es necesario para el uso correcto de la función “delay” que se usa en este programa.

```
#define _XTAL_FREQ 1000000
```

- **Port Names:** En esta sección se asignan nombres a los puertos para darle mejor legibilidad al programa.
- **Function declaration:** Aquí incluimos el “prototipo” de las funciones que generamos para utilizar en el main del programa, estas funciones se desarrollaron en la última parte del script, pero declaramos su existencia (por así decirlo) antes del main. En este caso utilizamos una función para inicializar los puertos y otra más para hacer el efecto Knight rider, que aparece cuando el usuario logra “pegarle al topo”.

```
void portsInit( void );
void rider(void);
```

- **Main:** En el programa principal lo que se hace es que se tiene un ciclo infinito, en el que se está generando un número random entre los números con base = 2 y exponente que va del 0 al 7 para encender uno de los LEDs a la vez aleatoriamente, y a su vez se encuentra constantemente revisando el estado de los botones por medio de un condicional que verifica si el número de LED encendido es igual al número de botón que se encuentre presionado, para que de ser así se ejecute la función rider() que desplegará el Knight rider effect que indica que el usuario ha presionado el botón correspondiente al LED encendido en ese momento, después de esta condicional el programa continúa con el encendido aleatorio de LEDs hasta que se vuelva a presentar un match.
- **Functions:** Aquí desarrollamos las funciones a utilizar en el programa.

```
void portsInit( void ){
    ANSELA = 0b00000000;           // Set Port A as digital port
    TRISA = 0b00000000;           // Set Port A as output
    PORTA = 0b00000000;           //Set initial value on outputs = 0

    ANSELD = 0b00000000;          // Set Port D as digital port
    TRISD = 0b11111111;          // Set Port D as inputs
}
```

Es así como al integrar el desarrollo del código con el hardware preparado anteriormente se logra realizar simulaciones del principio básico de diferentes procesos, en este caso adaptado a un juego Arcade. De aquí la importancia de saber cómo relacionar elementos y, sobre todo, conocer el funcionamiento básico para realizar la programación y pruebas experimentales de firmware para corregir posibles errores lógicos del diseño.

Link a github:

<https://github.com/Estef06/Equipo3-Microcontroladores/tree/main/Lab05>

Link al video:

https://drive.google.com/drive/folders/1tjay3ajw3W2b_zK06ZfbDE0OjEAdwJHw?usp=sharing

Fotos de evidencia



