Proyecto 2
Laboratorio de Microcontroladores
Temporizador y GPIOs Parte 2 creación semáforo

Marco Antonio Montero Chavarría Carné: A94000
Francisco Molina Carné: B14194

20 de septiembre de 2015

1. Problema

Desarrollar un "semáforo inteligente"simplificado utlizando los leds, la botonera, y los temporizadores del microcontrolador STM32F4,e los cuatro leds disponibles en el microcontrolador los leds LD3 y LD5 representarán el semáforo vehicular, mientras que los LD4 y LD6 serían el semáforo peatonal. El botón B1 (user_button) sería el puerto con el que el usuario solicitará la activación de las luces peatonales. La asignación de funcionalidades de los leds es la siguiente:

- LD3:paso de vehículos.
- LD5:vehículos detenidos.
- LD4:paso de peatones.
- LD6:peatones deternidos

El botón se puede presionar inclusive antes que se acaben los 10s del funcionamiento de LD3, pero debería permanecer encendido hasta que terminen los 10s. Si no se presiona el botón LD3 debería permancer encendido indefinidamente. La temporización de los leds se debe realizar utilizando uno de los temporizadores del microcontrolador. Se recomienda utilizar el Advanced Timer 1 con interrupciones de tiempo habilitados. Como innovación se podrían agregar leds externos para representar adecuadamente con colores verde y rojo ambos semáforos, o agregar una luz amarilla externa con su respectivo control adicional. En cualquiera de los se deberían utilizar resistencias de pull-up para los leds alimentados por el mismo micro-controlador para garantizar la seguridad de la tarjeta.

2. Solución Propuesta

Crear un código en C que sea capaz de cumplir los requisitos del proyecto y que una vez cargado en el stm32f4, el mismo simule el comportamiento del semáforo. Se utilizará además el código creado en el primer proyecto y a partir del mismo, se agregerá lo necesario para llegar un código final acorde al problema.

3. Procedimiento

Se utilizó el archivo glove.c en la carpeta scr_glove de opencoroco que fue modificado en el proyecto 1. // Sobre el código existente se realizaron los siguientes cambios en el método del temporizador:

```
void tim1_up_tim10_isr(void)
{
  //Clear the update interrupt flag
  timer_clear_flag(TIM1,TIM_SR_UIF);


static int counter = 0;
static int counter1 = 0;
static int counter2 = 0;
counter +=1 ;
counter1 +=1 ;
counter2 +=1 ;
int button;
button = gpio_get(GPIOA,GPIO0);
static int bandera=0;
static int banderaseg=0;
if(button == 0)
{

                if(bandera == 0)//Revise si llego el boton
                {
                        gpio_set(GPIOD,GPIO12);//Vehic Verde
                        gpio_set(GPIOD,GPIO13);//Peatonal Rojo
                        gpio_clear(GPIOD,GPIO15);//Peatonal Verde
                        gpio_clear(GPIOD,GPIO14);//Vehic Rojo
                }
                else if(banderaseg == 1)//Revise si llego el boton y pasaron 10
                {
                                if(counter >=3000)// 3 segs, parpadee
                                {
                                        if(counter >=4000)// 1 segs, ponga en ro
                                        {
                                                if(counter >=14000)// 10 segs, p
```

```
                {
                        if(counter >=17000)// 3
                        {
                                if(counter >=180
                                {
                                bandera=0;
                                banderaseg=0;
                                counter2=0;
                                }
                                else{
                                        gpio_set
                                        gpio_set
                                        gpio_cle
                                        gpio_cle
                                }
                        }
                        else{
                                if(counter1 >=50
                                {
                                gpio_toggle(GPIO
                                gpio_toggle(GPIO
                                counter1=0;
                                }
                        }
                }
                else{
                        gpio_set(GPIOD,GPIO15);
                        gpio_set(GPIOD,GPIO14);
                        gpio_clear(GPIOD,GPIO12)
                        gpio_clear(GPIOD,GPIO13)
                }
        }
        else{
                gpio_set(GPIOD,GPIO14);
                gpio_set(GPIOD,GPIO13);
                gpio_clear(GPIOD,GPIO12);
                gpio_clear(GPIOD,GPIO15);
        }
}
else{
        if(counter1 >=500)
        {
        gpio_toggle(GPIOD,GPIO12);
        gpio_toggle(GPIOD,GPIO13);
        counter1=0;
        }
```

```
                        }
                }
                else if( counter2>=10000)
                {
                        banderaseg=1;
                        counter= 0;
                }

}
else
{
bandera=1;
counter=0;
}
}
```

Esta definición añadida permite además de cumplir con los tiempos requeridos por el problema, encender los leds de acorde a los mismos. Los leds se definieron de la siguiente manera:

```
GPIO12 //Vehiculo Verde
GPIO13 //Peatonal Rojo
GPIO15 //Peatonal Verde
GPIO14 //Vehiculo Rojo
```

Los if's anidados permiten incluir los retardos de tiempo necesarios entre cada una de las funciones del semáforo, como lo son el temporizador de 10 seg previo al contacto del botón para pedir paso peatonal. Este temporizador espera 10 segundos y se limpia cada vez que hay un contacto en el botón. Si el bóton llega antes de que se cumplan los 10 seg, el método funciona de tal manera que el cambio de luces se realiza hasta que se cumplan esos 10 seg mínimos.// Además se pone una mecánica de parpadeo a los leds para indicar que van a cambiar de estado, esto se logra mediante los if's que poseen un contador interno con los métodos toggle dentro de sus llaves.// Como innovación se realizó un circuito en una protoboard que se conectara a los pines definidos para los leds y así ver el circuito funcionando de una manera más agradable.// En este ambito se podría trabajar aún más, quizas añadir un tercer led para el semáforo de vehículos que sea de color amarillo.

4. Conclusiones

El proyecto a pesar de tener sus dificultades iniciales, se logró con exito. Más aún se pudo hacer una implementación en una protoboard de la cual se obtiene una grabación que será añadida a la carpeta del problema o se brindará un link para el video para que el lector pueda observarlo.//

Cabe destacar que el proyecto deja una base sólida para futuros proyectos en especial el uso de temporizadores, ya que son extremadamente útiles para

definir comportamientos reales con retardos.