



UNIVERSIDAD
AUTÓNOMA
DE QUERÉTARO



FACULTAD
DE INGENIERÍA

PRÁCTICA 1: PARPADEO DE LED RGB

MICROCONTROLADORES

RAMÍREZ ÁLVAREZ CARLO IVÁN - 280847

ONTIVEROS MARTÍNEZ BEATRIZ - 244784

TREJO DOMÍNGUEZ NELLY BIBIANA - 242494

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
INGENIERÍA BIOMÉDICA

MICROCONTROLADORES
ING. JOSÉ DE JESÚS SANTANA RAMÍREZ
ENERO,
2023

I. OBJETIVOS

- Parpadeo de un LED en Tiva LaunchPad con un retraso de 100 ms.
- Controlar uno de los LED's integrados en Tiva LaunchPad por uno de los pines GPIO del microcontrolador.

II. MARCO TEÓRICO

a. ¿Qué es OPENOCD?

Open On-Chip Debugger (OpenOCD) tiene como objetivo proporcionar depuración, programación en el sistema y pruebas de exploración de límites para dispositivos de destino integrados.

Lo hace con la ayuda de un *adaptador de depuración*, que es un pequeño módulo de hardware que ayuda a proporcionar el tipo correcto de señalización eléctrica al objetivo que se está depurando. Estos son necesarios ya que el host de depuración (en el que se ejecuta OpenOCD) generalmente no tendrá soporte nativo para dicha señalización, o el conector necesario para conectarse al objetivo.

b. ¿Qué es Peripheral Driver Library?

La biblioteca de controladores de periféricos TivaWare™ de Texas Instruments® es un conjunto de controladores para acceder a los periféricos que se encuentran en la familia Tiva™ de microcontroladores basados en ARM® Cortex™-M. Aunque no son controladores en el sentido puro del sistema operativo (es decir, no tienen una interfaz común y no se conectan a una infraestructura global de controladores de dispositivos), proporcionan un mecanismo que facilita el uso de los periféricos del dispositivo.

c. ¿Qué es un Compilador?

Un compilador es un programa que traduce código fuente escrito en un lenguaje de alto nivel como Java, a un lenguaje legible por la máquina llamado código objeto, lenguaje de destino o incluso lenguaje ensamblador. Por lo tanto, un compilador podría llamarse traductor, pero sus tareas son más amplias porque, como parte de la compilación del programa, también informa de errores al leer el código. ¿Qué es Depuración?

La depuración de programas es el proceso de identificar y corregir errores de programación

d. ¿Cuál es la diferencia principal entre un microcontrolador y un microprocesador?

Un microprocesador es un elemento que realiza operaciones lógico aritméticas. No dispone de entradas y salidas como un microcontrolador. Requiere de más periféricos adicionales para funcionar, como memorias o controladores de bus. Sin embargo, son más veloces al realizar estas operaciones que un microcontrolador.

- e. ¿Cuáles son los tipos principales de arquitecturas (microcontroladores)? Anotar principales características de cada tipo.

La arquitectura de un microcontrolador permite definir la estructura de su funcionamiento, las dos arquitecturas principales usadas en la fabricación de microcontroladores son: arquitectura de *Von Neumann* y arquitectura *Harvard*. Además, estas arquitecturas pueden tener procesadores de tipo *CISC* o de tipo *RISC*.

Arquitectura de Von Neumann

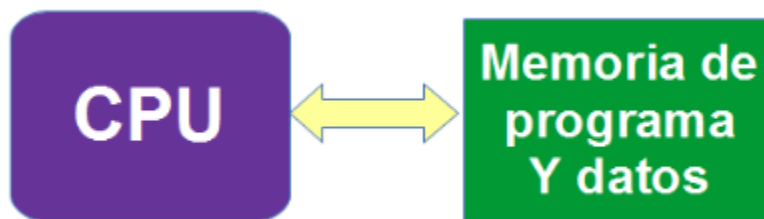


Fig 1. Arquitectura de Von Neumann

En esta arquitectura, los *datos* y las *instrucciones* circulan por el mismo bus ya que estos son guardados en la misma memoria, su principal ventaja es el ahorro de líneas de entrada-salida, pero esto supone una disminución en la velocidad con la que se realizan los procesos.

Arquitectura Harvard



Fig 2. Arquitectura Harvard

A diferencia de la anterior, en la arquitectura Harvard existe una memoria específica para datos y una memoria específica para las instrucciones, de esta forma se usan dos buses bien diferenciados. Con esto se logra trabajar con las dos memorias

simultáneamente y en consecuencia se obtiene mucha más velocidad en la ejecución de los programas.

- f. ¿Cuáles son y en qué consisten las herramientas de GNU Toolchain?

El GNU toolchain agrupa a una serie de proyectos que contienen las herramientas de desarrollo de software producidas por el proyecto GNU. Estos proyectos forman un sistema integrado que es usado para programar tanto aplicaciones como sistemas operativos.

III. MATERIAL

- Un osciloscopio.
- Un multímetro digital.
- Una protoboard.
- Resistencias de carbón de valores distintos.
- Un pulsador.
- Tiva LaunchPad
- Leds Ultrabrillantes
- Transistor 2n2222

IV. DESARROLLO DE LA PRÁCTICA Y RESULTADOS

Experimento 1: Control de la Corriente con un potenciómetro.

Para este experimento, se utiliza uno de los LED's integrados en Tiva LaunchPad. El LED estará controlado por uno de los pines GPIO del microcontrolador Tiva. Para lograr la tasa de parpadeo requerida, el pin GPIO que controla el LED deberá alternarse entre los estados lógico alto y lógico bajo con el retraso requerido de 100 ms. El diagrama de bloques básico para la configuración del experimento se muestra en la Fig. 1. El Tiva LaunchPad tiene un LED RGB incorporado, controlado por los pines GPIO PF1, PF2 y PF3.

Durante la realización de esta práctica pudimos notar que el LED al encenderse y apagarse a una alta velocidad daba una tonalidad blanca.

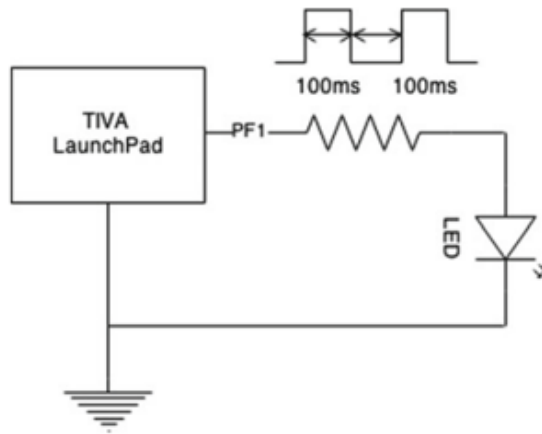
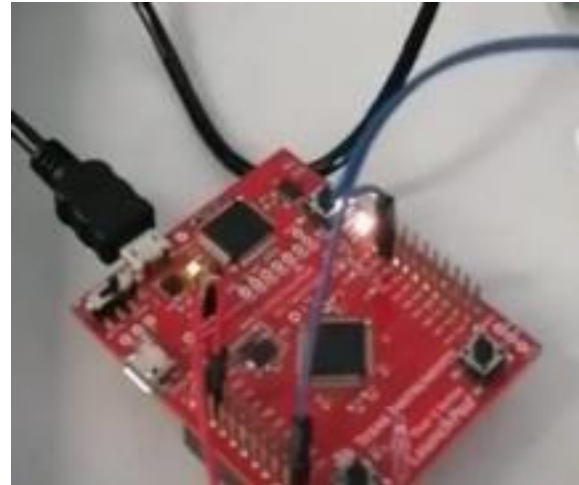


Fig 3. Diagrama de Conexión.



Experimento 2: Arreglos paralelo y mixto.

Para este experimento, solo se requiere un LED, el LED rojo está controlado por PF1. El fragmento esquemático del Tiva LaunchPad (Fig. 2) muestra la configuración alta activa del LED RGB de ánodo común integrado que utiliza transistores NPN. Tal configuración es necesaria porque el microcontrolador Tiva solo es capaz de suministrar hasta 18 mA de corriente como máximo en dos pines ubicados en un lado físico del paquete del dispositivo. Esta limitación de corriente máxima se supera mediante el uso de la configuración de transistor NPN. En esta configuración, la lógica alta encenderá el LED y la lógica baja lo apagará.

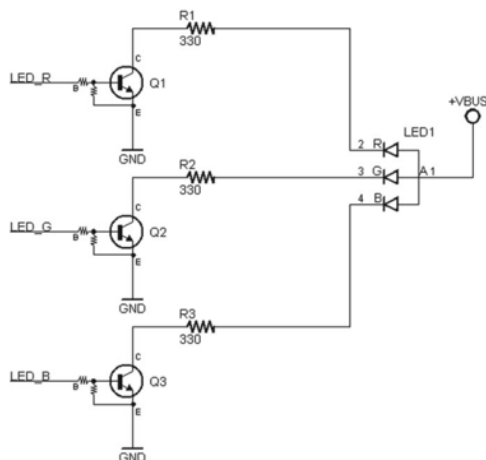
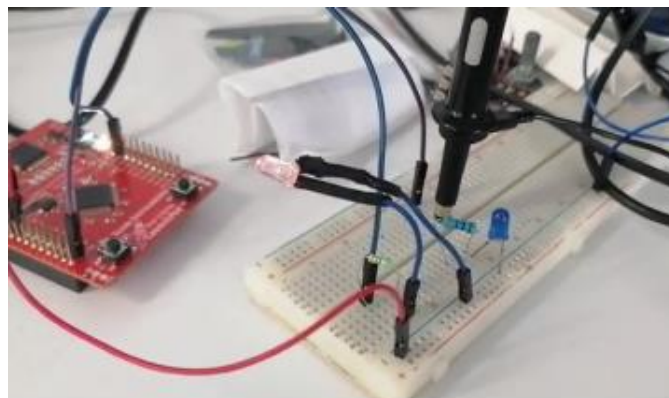
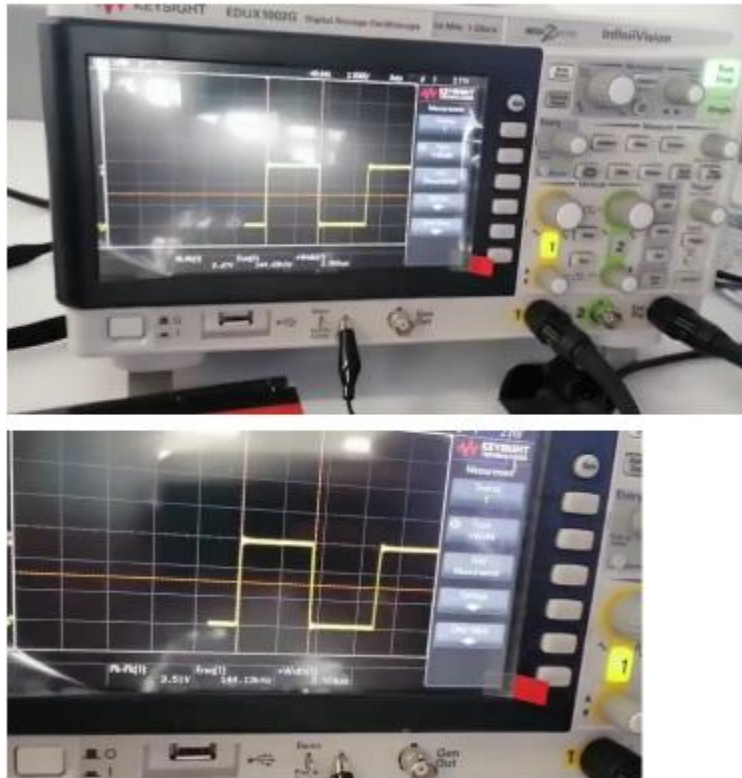


Fig 4. Diagrama de conexión transistor led RGB



Experimento 3: Uso del generador de funciones y del osciloscopio.

Activando los puertos internos de la tiva, donde una cuenta (un ciclo for) equivale a 3.56 ms en la escala del tiempo, lo que da su velocidad y su color.



Código utilizado

```
Getting Started | project0.c | TM4C123GH6PM.h
1 |
2 #include "lib/include.h"
3
4 unsigned long Led;
5
6 void Delay(void){unsigned long volatile time;
7     time = 100000;
8     while(time){
9         time--;
10    }
11 }
12
13 int main(void){
14
15     SYSCTL->RCGC_GPIO |= (1<<5);
16     //SYSCTL->RCGC2 |= 0x00000020;
17     // SYSCTL_RCGC2_R |= 0x00000020; // 1) activate clock for Port F
18     GPIO_PORTF_LOCK_R = 0x4C4F434B; // 2) unlock GPIO Port F
19     GPIO_PORTF_CR_R = 0x1F; // allow changes to PF4-0
20     GPIO_PORTF_AMSEL_R = 0x00; // 3) disable analog on PF
21     GPIO_PORTF_PCTL_R = 0x00000000; // 4) PCTL GPIO on PF4-0
22     GPIO_PORTF_DIR_R = 0x0E; // 5) PF4,PF0 in, PF3-1 out
23     GPIO_PORTF_AFSEL_R = 0x00; // 6) disable alt funct on PF7-0
24     GPIO_PORTF_PUR_R = 0x11; // enable pull-up on PF0 and PF4
25     GPIO_PORTF_DEN_R = 0x1F; // 7) enable digital I/O on PF4-0
```

```
26 while(1){
27     //if(PF4 == 0x0){
28         Led = 0x02;           // reverse value of LED
29         GPIO_PORTF_DATA_R = Led; // write value to PORTF DATA register,toggle led
30         Delay();
31         Led = 0x04;           // reverse value of LED
32         GPIO_PORTF_DATA_R = Led; // write value to PORTF DATA register,toggle led
33         Delay();
34         Led = 0x08;           // reverse value of LED
35         GPIO_PORTF_DATA_R = Led; // write value to PORTF DATA register,toggle led
36         Delay();
37         Led = 0x0A;           // reverse value of LED
38         GPIO_PORTF_DATA_R = Led; // write value to PORTF DATA register,toggle led
39         Delay();
40     // }
41 }
42 }
43
```