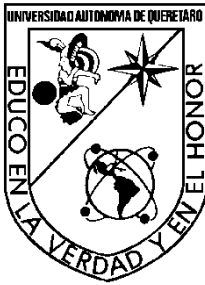
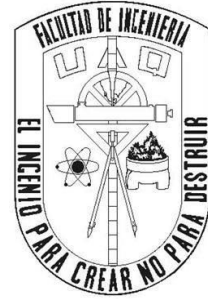


Ingeniería Biomédica



**UNIVERSIDAD AUTÓNOMA DE
QUERÉTARO**



INGENIERÍA BIOMÉDICA

PRACTICAS DE LABORATORIO

Periodo: agosto-diciembre 2022

Docente: Ing. José de Jesús Santana Ramírez

Ayudante: Ricardo Carrillo Guzmán

Alumnos:

Yessica Sosa Zepeda

Jessica Itzel Ugalde Pérez

Práctica 1

Experimento 1

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 aTiva LaunchPad tiene un LED RGB incorporado, controlado por los pines GPIO PF1, PF2 y PF3.

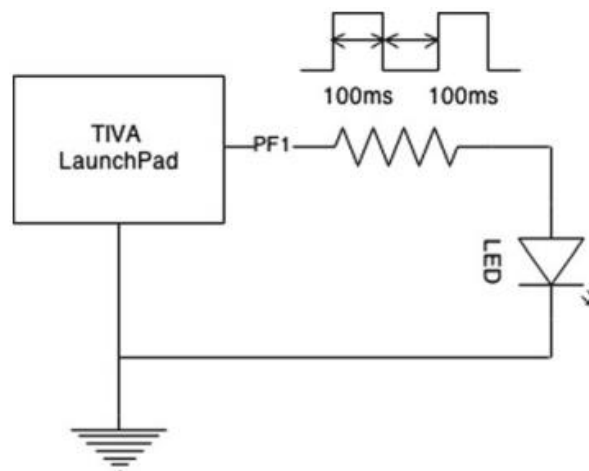


Fig 1. Diagrama de Conexión.

```
int main(void)
{
    volatile uint32_t ui32Loop;
    volatile uint32_t valor;
    //
    // Enable the GPIO port that is used for the on-board LED.
    //
    SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOF;

    //
    // Do a dummy read to insert a few cycles after enabling the peripheral.
    //
    ui32Loop = SYSCTL_RCGC2_R;

    //
    // Enable the GPIO pin for the LED (PF3). Set the direction as output, and
    // enable the GPIO pin for digital function.
    //

    //Puerto F
    GPIO_PORTF_DIR_R = 0x02 ; // configura el Port F como Salida en el pin
    GPIO_PORTF_DEN_R = 0x02 ;

    //
    // Loop forever.
    //
    valor = 133000;

    while(1)
    {
        //
        // Turn on the LED.
        //
        GPIO_PORTF_DATA_R |= 0x02 ;

        //
        // Delay for a bit.
        //
        for(ui32Loop = 0; ui32Loop < valor; ui32Loop++)
        {
        }

        //
        // Turn off the LED.
        //
        GPIO_PORTF_DATA_R &= ~(0x02);

        //
        // Delay for a bit.
        //
        for(ui32Loop = 0; ui32Loop < valor; ui32Loop++)
        {
        }
    }
}
```

Fig. 2 Código en VS code

Para realizar lo solicitado en el experimento 1, se hicieron unas pequeñas modificaciones en el archivo blinky.c, donde agregamos una variable llamada valor, con la que controlamos el tiempo entre los estados alto y bajo.

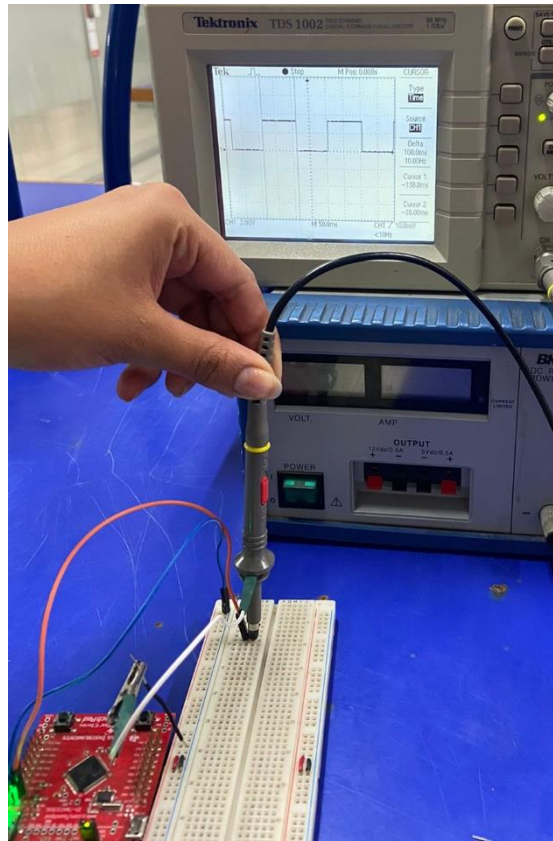


Fig. 3 Evidencia salida del pin PF3 vista en el osciloscopio.

Para el Experimento 1, lo primero que hicimos fue ver directamente la salida del pin PF3 en el osciloscopio, simplemente conectando este pin a la protoboard, así como un pin a tierra.

Comprobamos en el osciloscopio que la salida cumplía con el retraso requerido de 100 ms.

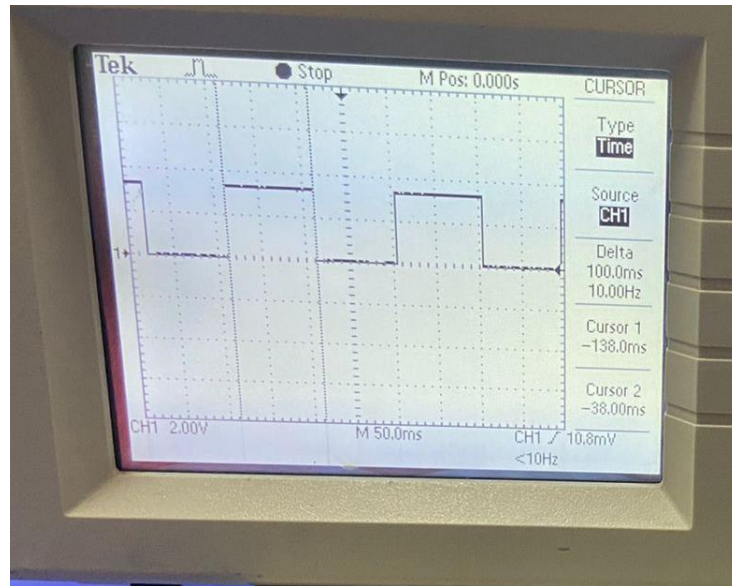


Fig. 4 Evidencia del osciloscopio

En la imagen anterior, podemos ver detalladamente la señal generada con un retraso de 100ms.

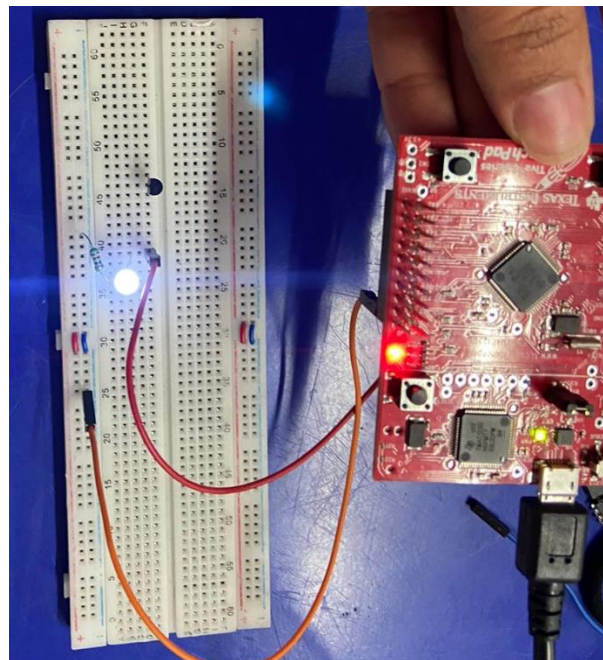


Fig. 5 Conexión con led

Para el experimento 1, seguimos el diagrama de conexión indicado, simplemente conectamos el pin PF3 al ánodo del led, mientras que el cátodo lo conectamos a tierra con su respectiva resistencia de 330 Ohms.

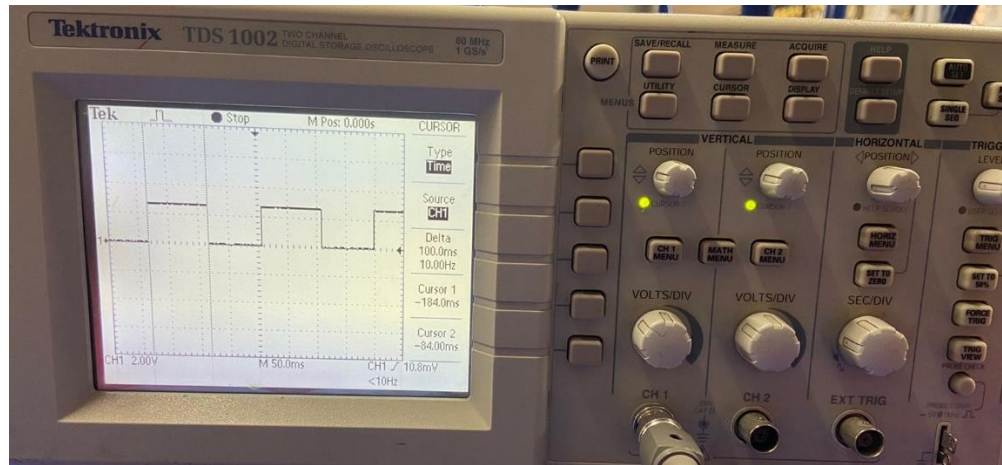


Fig. 6 Evidencia osciloscopio

Después de realizar la conexión indicada, utilizamos el osciloscopio para comprobar el retraso de 100 ms.

Experimento 2

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.

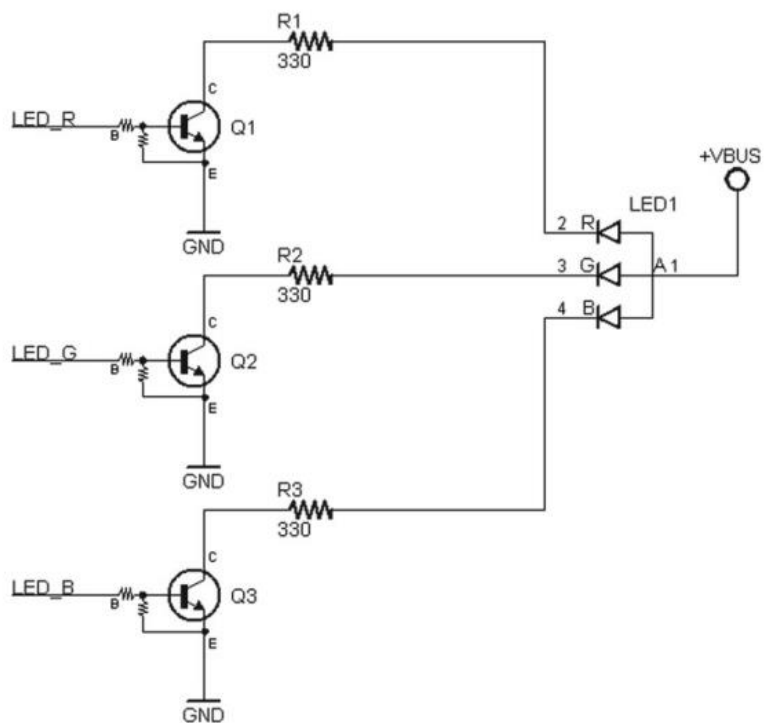


Fig. 2. Diagrama de conexión transistor led RGB

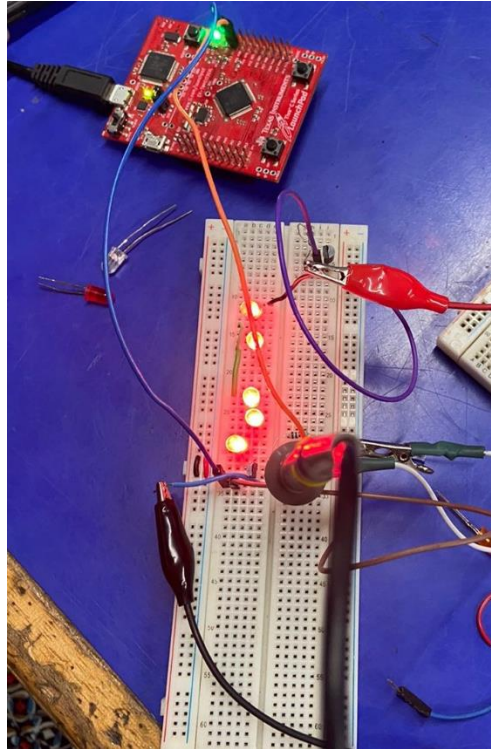


Fig. 7 Circuito experimento 2

Para el experimento 2, conectamos 5 leds en serie, dichos leds conectados a un transistor 2N222A, en el pin del transistor colector, se conectó al primer led, la base se conectó al pin PF3 de la tiva y el emisor directamente a tierra.

No colocamos resistencia en la conexión en serie de los leds, ya que la caída de voltaje por led era de aproximadamente 2 volts, alimentamos el circuito con una fuente de 10Volts y de esta manera evitamos utilizar una resistencia para los leds.

La única resistencia requerida, fue una de 1kOhm que se conectó a la base del transistor.

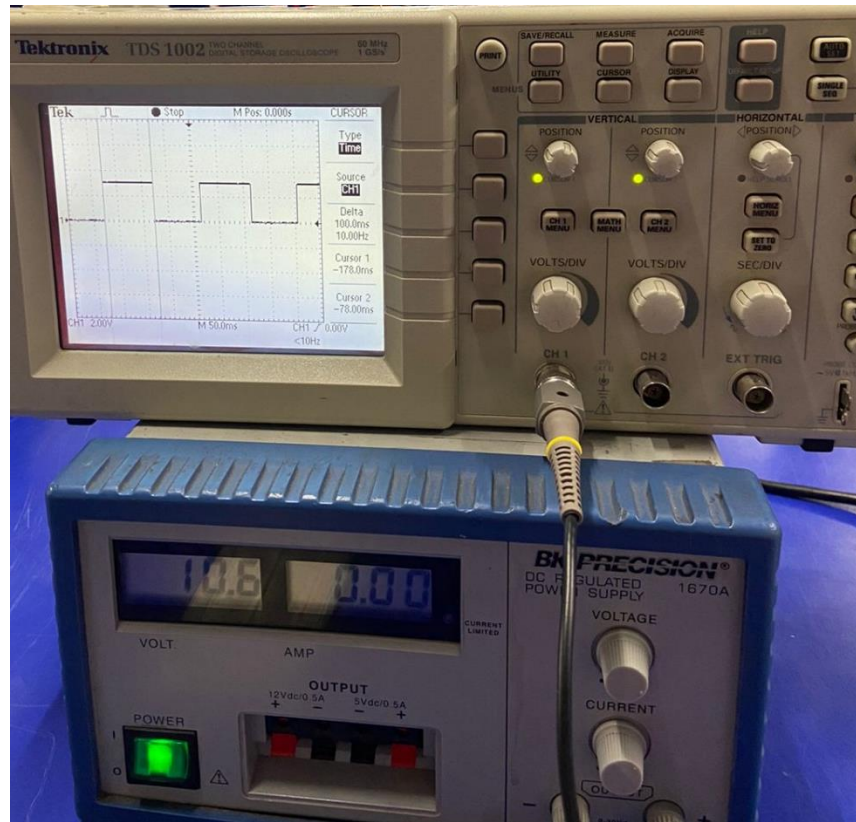


Fig. 8. Evidencia osciloscopio

Por último, alimentamos el circuito con una fuente suministrando 10Volts, y observamos la salida en el osciloscopio, comprobando así, el retraso requerido de 100ms.

Objetivo(s): Parpadea un LED en Tiva LaunchPad con un retraso de 100 ms.

Actividad complementaria:

Investigar:

- ¿Qué es OPENOCD?

La herramienta OpenOCD, que significa "Open On-Chip Debugger", logra la depuración, programación y "Boundary scanning" para dispositivos de destino integrados.

Es una herramienta que nos permite comunicarnos con nuestro hardware y poder programarlo. Puede conectarse con un programador/debugger y permite pasar un programa compilado a un microcontrolador.

- ¿Qué es Peripheral Driver Library?

Peripheral Driver Library es una biblioteca de controladores periféricos TivaWare de Texas Instruments, que cumple la función de acceder a los periféricos que hay en la familia Tiva de microcontroladores basados en ARM Cortex-M.

- ¿Qué es un Compilador?

En el área de la programación, el compilador traduce un programa escrito en cierto tipo de lenguaje de programación, como código fuente, a otro lenguaje alto, medio o de bajo nivel, como C o C++.

Básicamente se puede decir que, para crear un programa, es necesario un compilador, el cual también es un programa informático.

- ¿Qué es Depuración?

La depuración en pocas palabras significa quitar errores del código de programación, aunque hay varias maneras para hacerlo, como examinar el código para encontrar errores tipográficos o usar un analizador de código, también puede depurarse mediante un generador de perfiles o simplemente usar un depurador, que es una herramienta especializada que se asocia a la aplicación en ejecución y permite inspeccionar el código.

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

La diferencia principal es que los microcontroladores tienen periféricos externos RAM, ROM, EEPROM integrados, en cambio para los microprocesadores es necesario usar circuitos externos. El microcontrolador es mucho más compacto comparado con un microprocesador, debido a que los periféricos del microcontrolador están en un solo chip.

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

Gracias a la arquitectura de un microcontrolador es posible definir la estructura de su funcionamiento.

Existen dos arquitecturas principales para los microcontroladores:

-Arquitectura de Von Neumann

-Arquitectura Harvard

En la arquitectura Von Neumann los datos y las instrucciones pasan por el mismo bus porque se guardan en la misma memoria.

Una ventaja es que se pueden ahorrar líneas de entrada-salida.

En la arquitectura Harvard, existe una memoria específica para los datos y otra para las instrucciones, utilizando así, dos buses distintos, dando como resultado una ejecución mucho más rápida.

- ¿Cuáles son y en qué consisten las herramientas de GNU Toolchain?
GNU toolchain es una serie de proyectos con varias herramientas de desarrollo de software, por el proyecto GNU. Dichos proyectos forman un sistema integrado, utilizado para programar aplicaciones o sistemas operativos.
Algunos proyectos son:
GNU make: automatización de la estructura y de la compilación.
GNU Compiler Collection (GCC): compiladores para varios lenguajes.
GNU Binutils: enlazador, ensamblador y otras herramientas.
GNU Debugger (GDB): un depurador interactivo.
GNU build system (autotools): Autoconf, Autoheader, Automake, Libtool - generadores de makefiles.

Material:

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

Conclusión

Se logró llevar a cabo correctamente los experimentos 1 y 2, donde pudimos comprobar el retraso de los 100ms requeridos al parpadear tanto el led integrado de la Tiva, un led externo y una conexión de leds en serie empleando un transistor, mediante el uso de un osciloscopio.

Visual Studio Code, es una herramienta que nos permite realizar los ajustes pertinentes para los códigos de programación que se deseen, empleando también la Tiva y los componentes necesarios

Desarrollo de la práctica:

Experimento 1: Control de la Corriente con un potenciómetro

(arreglo en serie).

Solicite la revisión del experimento.

Experimento 2: Arreglos paralelo y mixto.

Solicite la revisión del experimento.

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

Solicite la revisión del experimento.

Experimento 4: Uso del capacitor y desfase entre señales.

Solicite la revisión del experimento.

HOJA DE REVISIÓN (Práctica 1)

Favor de llenar con tinta azul (sólo los nombres de los integrantes).

Integrante	No. Experimento				Cuestionario
	1	2	3	4	

Firma del encargado de la práctica

Referencias

- <https://tech.tribalyte.eu/blog-programa-o-compiler>
- <https://docs.microsoft.com/es-es/visualstudio/debugger/what-is-debugging?view=vs-2022>
- <http://sherlin.xbot.es/microcontroladores/introduccion-a-los-microcontroladores/arquitectura-de-microcontroladores>
- <https://spiegato.com/es/que-es-la-depuracion>