Laboratorio de Sistemas Embebidos Avanzados
Práctica No. 2
Reloj Digital Empleando Teclado Matricial y Display de Cristal Líquido

Profesor Teoría: Ricardo Andrés Velásquez V. (randres.velasquez@udea.edu.co)
Profesor Laboratorio: Luis Germán García M. (german.garcia@udea.edu.co)

Diciembre 15, 2021

Fecha de entrega:	Enero 26 de 2022
Medio de entrega:	http://www.ingeniaudea.co
Sustentación:	Enero 26 de 2022
Valor Práctica:	10% del curso

1 Introducción

En esta práctica de laboratorio, el objetivo principal será el desarrollo de un par de controladores o *drivers* para el manejo tanto de un teclado matricial como de un display de cristal líquido (LCD), empleando las entradas y salidas de propósito general (GPIO) y el módulo de TIMER disponibles en el MCU RP2040 del sistema de desarrollo Raspberry Pi Pico.

Como aplicación, el grupo de trabajo desarrollará un reloj digital con alarma mediante el uso del módulo RTC del MCU RP2040.

Para la implementación, el grupo de estudiantes podrá emplear el sistema de desarrollo Raspberry Pi Pico junto con la SDK de C/C++, además de librerías propias.

El esquema del sistema a implementar se muestra en la Fig. 1.

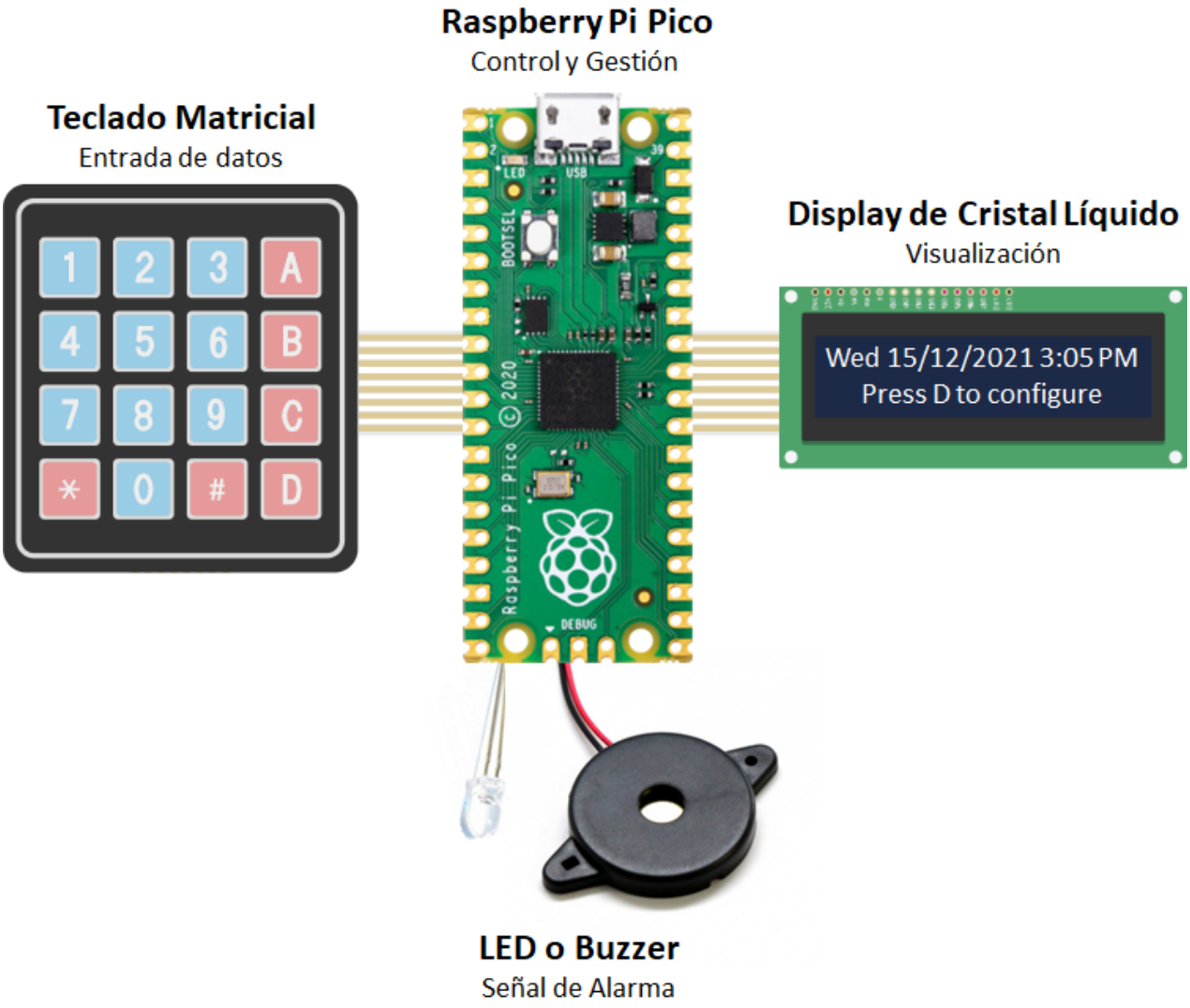

Fig. 1: Esquema del sistema a diseñar

2 Consideraciones

Las consideraciones para el desarrollo de los drivers para el teclado y el LCD se muestran a continuación:

a. Utilice entradas y salidas de propósito general GPIO para manejar las filas y columnas del teclado. Tenga en cuenta que algunas de ellas se deben configurar como interrupciones. Emplee también líneas de GPIO para manipular el LCD.

b. Active resistores de pull-up o pull-down en los GPIOs según corresponda para la correcta lectura de filas y columnas del teclado matricial.

c. Haga uso del módulo de TIMER para esperar a que la señal generada al presionar una tecla se estabilice. Haga uso también de este módulo para el manejo de tiempos del LCD o utilice la función sleep_us para tiempos de espera en microsegundos. Una estrategia recomendada para evitar utilizar la función sleep_us es leer la bandera de busy del LCD.

Las consideraciones para el desarrollo del reloj digital con alarma se muestran a continuación:

a. Habrá dos modos de operación: configuración y normal.
b. En modo configuración se podrá establecer mediante teclado matricial, con visualización en LCD:
   (a) La fecha y hora actual (Ej. Diciembre 15 de 2021, 3PM).
   (b) Una alarma para una fecha y hora determinada.
c. En modo normal, el sistema:
   (a) Visualizará la fecha y hora actual (Ej. Diciembre 15 de 2021, 3PM) en el display de cristal líquido.
   (b) Activará una señal sonora o luminosa cuando la fecha y hora de la alarma coincida con la fecha y hora actual. La señal de alarma se desactivará transcurridos 60 segundos o cuando el usuario presione una tecla específica.

Consideraciones generales:

a. La programación se deberá realizar empleando el SDK de C/C++ para la Raspberry Pi Pico y librerías propias. Es mandatorio utilizar los módulos TIMER, GPIO y RTC para el desarrollo de la aplicación, en el cual se haga uso de interrupciones (trate de evitar el uso de la función sleep_us a menos que no encuentre otra solución). El módulo UART puede ser empleado a través de las funciones de la librería de entrada y salida estándar STDIO (printf, getchar, etc.) para la realización de pruebas de verificación.
b. Haga uso de programación con metodología de eventos (interrupciones + *polling*).
c. Las funciones de los drivers que realice, tanto para el teclado matricial como para el LCD, no deben bloquear el flujo de ejecución a la espera de una acción por parte del usuario. Por ejemplo, si realiza una función llamada GetKey() para obtener la tecla presionada por el usuario, dicha función no deberá quedarse a la espera de que el usuario presione una determinada tecla.

3 Entrega

Número de integrantes por grupo: máximo dos.
La entrega de la práctica puede hacerse hasta la fecha límite dada en esta guía. Para la entrega, cree un archivo comprimido que incluya los archivos fuente de su programa en lenguaje C/C++ (archivos de extensión .c, .cpp y .h) y súbalo a la plataforma del curso (IngeniaUdeA). El nombre del archivo comprimido debe tener el siguiente formato:
p2_primerapellidointegrante1_primerapellidointegrante2.zip.
Ejemplo: si el primer apellido de ambos integrantes es **Velásquez** y **García**, respectivamente, entonces el archivo debe ser nombrado así: *p2_velasquez_garcia.zip*.

4 Evaluación

La evaluación de la práctica se divide en dos partes, funcionamiento (60%) y sustentación (40%). Cada grupo de trabajo deberá sustentar la práctica en un tiempo de 20 minutos, 10 minutos para mostrar el funcionamiento mediante implementación real y 10 minutos para sustentar el diseño. Durante la sustentación, el profesor hará entre dos (2) y tres (3) preguntas a cada uno de los integrantes del grupo de trabajo.

5 Referencias

a. Embedded System Design de Peter Marwedel. Kluwer Academic, Springer, 2006.
b. Programming Embedded Systems in C and C++ de Michael Barr. O'Reilly.
c. Documentación SDK para Raspberry Pi Pico
https://raspberrypi.github.io/pico-sdk-doxygen/
d. Getting Started with Pico Manual
https://datasheets.raspberrypi.org/pico/getting-started-with-pico.pdf
e. Raspberry Pi Pico Datasheet
https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf
f. RP2040 Microcontroller Datasheet
https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf
g. Raspberry Pi Pico C/C++ SDK
https://datasheets.raspberrypi.com/pico/raspberry-pi-pico-c-sdk.pdf