

Laboratorio de Sistemas Embebidos Avanzados

Práctica No. 4

Monitoreo y Control Remoto de Variables Físicas

Profesor Teoría: Ricardo Andrés Velásquez V. (randres.velasquez@udea.edu.co)

Profesor Laboratorio: Luis Germán García M. (german.garcia@udea.edu.co)

Febrero 16, 2022



Fecha de entrega: Marzo 9 de 2022

Medio de entrega: <http://www.ingeniaudea.co>

Sustentación: Marzo 9 de 2022

Valor Práctica: 10% del curso

1 Introducción

En esta práctica de laboratorio, el grupo de estudiantes diseñará e implementará un sistema de monitoreo y control remoto de variables físicas. Entre los objetivos de esta práctica están:

- Emplear periféricos apropiados para la captura de magnitudes físicas y realizar el reporte de las mismas a un celular con Android o iOS.
- Controlar algunos periféricos conectados al MCU desde el propio celular.
- Emplear comunicación Bluetooth entre el celular y el MCU para el intercambio de información.

- d. Emplear mecanismos de consulta (comando) / respuesta con verificación de check-sum para el intercambio de información.
- e. Realizar la programación del MCU mediante el uso del lenguaje de programación C para sistemas microcontrolados con recursos limitados, empleando la metodología de manejo de eventos orientada al bajo consumo y máquinas de estados finitos.
- f. Hacer uso de los diferentes fundamentos de electrónica tales como circuitos electrónicos, digitales y otros para el desarrollo del hardware del sistema.
- g. Emplear el sistema de desarrollo Raspberry Pi Pico junto con la SDK de C/C++, además de librerías propias, para la implementación.

2 Diseño e Implementación

Al grupo de estudiantes se le encarga la labor de desarrollar un sistema embebido con la capacidad de monitorear y controlar distintas variables físicas mediante el uso de celular, microcontrolador y periféricos apropiados. El desarrollo de esta práctica será llevado a cabo en varias fases:

2.1 Fase 1: Monitoreo y Control

Desarrolle una serie de módulos tanto de hardware como de software para:

- a. Sensar dos variables o magnitudes físicas: temperatura, presión, luminosidad, humedad, etc.
- b. Manipular la intensidad de un led empleando la técnica de PWM. Utilice un LED externo de alta luminosidad.
- c. Manipular la velocidad de una hélice que está conectada a un motor DC empleando la técnica de PWM.

Tenga en cuenta las siguientes restricciones de diseño en sus módulos de software:

- a. Para cada periférico empleado en el MCU, cree su respectivo módulo de software (x.c y x.h) que incluya funciones de inicialización, operación y servicio de interrupción.
- b. Vincule los distintos módulos de software mediante el uso de la metodología de eventos, la cual podrá estar en un módulo aparte.
- c. La función main solo podrá tener llamados a funciones de inicialización y ciclo while infinito con el llamado a la función de control de eventos.
- d. Dentro de la función donde se manejan los eventos, sólo podrá existir un llamado a la instrucción en ensamblador WFI.

Emplee la terminal serial que ha utilizado en los laboratorios previos para llevar a cabo pruebas preliminares de esta primera etapa.

2.2 Fase 2: Comandos de Monitoreo y Control

La interfaz de usuario del sistema estará en una aplicación a ser ejecutada en un celular basado en Android o iOS. En esta segunda fase, el grupo de estudiantes creará varios módulos de software para establecer la comunicación con la aplicación empleando Bluetooth. Para el intercambio de información, realice lo siguiente:

- a. Emplee el módulo Bluetooth 4.0 basado en el chip CC2541 o compatible, el cual soporta comunicación UART, para el intercambio de información con el celular. Este módulo también se conoce como HM10 BLE 4.0.
- b. Cree un módulo de software (x.c y x.h) para establecer comunicación serial con el módulo Bluetooth a través de uno de los módulos UART del MCU. Este módulo debe incluir funciones de inicialización, recepción de comandos desde el celular, envío de respuestas al celular y servicios de interrupción.
- c. Diseñe el módulo de comunicación serial de manera tal que el intercambio de información a través de la UART se realice empleando interrupciones (uso de FIFOs es imprescindible). Recuerde integrar este módulo con el controlador de eventos para monitorear y controlar las variables establecidas en la fase 1.

La comunicación entre el MCU y el celular se realizará mediante un esquema maestro / esclavo. Todas las solicitudes de transferencia de información inician en la aplicación del celular (maestro), obteniendo una respuesta desde el MCU (esclavo). La aplicación solicitará el reporte de una variable o la modificación del estado de determinado periférico mediante una trama con el siguiente formato:

Formato de trama: Encabezado (1-byte), Monitoreo o Control (1-byte), Valor (1-byte) y Check-Sum (1-byte). Cada elemento de la trama se describe a continuación:

- a. **Encabezado:** corresponde al caracter '\$'.
- b. **Monitoreo o control:** corresponde a un byte compuesto de dos campos: el bit más significativo para indicar monitoreo (0) o control (1); los 7 bits restantes para representar el identificador de la variable a sensar o periférico a controlar (0x00 - Temperatura, 0x01 - Aceleración, 0x40 - LEDR, 0x41 - Motor, etc.).
- c. **Valor:** corresponde a un valor entre 0 y 255 usado para modificar el estado de determinado periférico (control). En el caso de un comando de monitoreo, Valor será 0 (0x00).
- d. **Check-Sum:** corresponde a la suma de los bytes anteriores de la trama módulo 256.

Ejemplo de trama: 0x24C1280D

Encabezado \$ = 0x24; Control Motor: 0x80 OR 0x41 = 0xC1; Porcentaje PWM 40% = 0x28; Check.sum: (0x24 + 0xC1 + 0x28) MOD 256 = 0x0D

Por otro lado, el MCU responderá a la aplicación mediante una trama similar a la anterior. Las únicas consideraciones serían:

- a. Valor (Monitoreo), corresponde a un valor entre 0 y 255 usado para indicar el valor de la variable reportada (monitoreo).
- b. Valor (Control), corresponde al valor establecido en el periférico seleccionado siempre que la operación se haya realizado correctamente.

Tenga en cuenta que el Check-Sum se generará tanto en el transmisor (con el resto de caracteres de la trama a enviar) como en el receptor (con el resto de caracteres que se recibieron). Si el Check-Sum generado en el receptor es distinto al que se recibe en la trama, el comando o respuesta será ignorado. Establezca un mecanismo de time-out de manera que, si tanto la aplicación del celular como el MCU no han recibido el siguiente carácter de la trama en los siguientes X segundos, retorne al punto donde se pueda enviar / recibir un nuevo comando.

Emplee la terminal serial que ha utilizado en los laboratorios previos para llevar a cabo pruebas preliminares de esta segunda etapa.

2.3 Fase 3: Aplicación en Android o iOS

Diseñe una interfaz de usuario que permita monitorear o controlar todas las variables del sistema que estableció en la fase 1. Luego, agregue a la aplicación la capacidad de establecer comunicación a través de Bluetooth. Finalmente, establezca el algoritmo de intercambio de información entre la aplicación y el MCU para completar el diseño del sistema.

Realice pruebas para comprobar el funcionamiento de todo el sistema.

2.4 Componentes del desarrollo

- a. Microcontrolador RP2040.
- b. Dispositivo de comunicación Bluetooth. En el laboratorio estará disponible el módulo Bluetooth 4.0 basado en el chip CC2541 o usted puede adquirir uno. Usted será responsable de entender como manipular este módulo (Comandos AT están disponibles por si quiere modificar la configuración del módulo).
- c. Sensores apropiados junto con el respectivo circuito electrónico de acondicionamiento.
- d. Para la creación de la aplicación en Android, puede utilizar cualquier entorno o framework de desarrollo: Android Studio, NativeScript, React Native, Flutter, etc.

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 (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:

p4_primerapellidointegrante1_primerapellidointegrante2.zip.

Ejemplo: si el primer apellido de ambos integrantes es **Velásquez** y **García**, respectivamente, entonces el archivo debe ser nombrado: *p4_velasquez_garcia.zip*.

4 Evaluación

La evaluación de la práctica se divide en dos partes, funcionamiento (50%) y sustentación (50%). Cada grupo de trabajo deberá sustentar la práctica en un tiempo de 15 minutos, 7 minutos para mostrar el funcionamiento y 8 minutos para sustentar el diseño. Durante la sustentación, el profesor hará entre una (1) y dos (2) preguntas a cada uno de los integrantes del grupo de trabajo. Entre las preguntas que se podrían hacer estarían las siguientes:

- a. Indicar las ventajas y desventajas de la metodología de eventos.
- b. Explicar la configuración de un módulo particular del MCU.
- c. Explicar la forma como generó la trama de intercambio de datos.
- d. Explicar el módulo hecho para manejar el dispositivo de comunicación Bluetooth.
- e. Indicar bajo que condiciones, el MCU del sistema entra en modo de bajo consumo.
- f. Explicar cualquier parte del código escrito en lenguaje C.
- g. Realizar cambios al código escrito en lenguaje C.

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>
- h. Bluetooth V.4.0 BLE-CC41-A
<https://img.banggood.com/file/products/20150104013145BLE-CC41-A%20Specification.pdf>