



OBJETIVOS

- Entender y aplicar los flujos de programación clásicos de sistemas embebidos: polling, interrupciones y polling+interrupciones
- Comparar los ambientes de desarrollo de software disponibles para el MCU del curso: Arduino, SDK-Python y SDK-C y el desempeño que se puede obtener con cada uno de ellos
- Desarrollar habilidades en el manejo del SDK-C y el manejo de interrupciones

DESCRIPCIÓN

En esta práctica desarrollaremos en un dispositivo Generador Digital de Señales (GDS). El dispositivo GDS genera 4 formas de onda diferentes: seno, triangular, diente de sierra y cuadrada. El usuario puede por medio de un pulsador seleccionar el tipo de forma de onda que genera el GDS. Así cada que se presiona el botón el sistema pasa de una forma de onda a la siguiente de forma secuencial. La amplitud, el nivel DC (offset) y la frecuencia de la señal se introduce por medio de un teclado matricial de 4x4. Para ingresar la amplitud, el usuario presionara la tecla A seguida del valor de voltaje deseado en milivoltios y la tecla D para finalizar. De forma similar para ingresar el nivel DC, el usuario presionara la tecla B seguida del valor de voltaje DC deseado en milivoltios y la tecla D para finalizar. Finalmente, para ingresar la frecuencia, es necesario presionar la tecla C seguida del valor de frecuencia en Hertz y la tecla D para finalizar. Los valores actuales de Amplitud, Nivel DC y Frecuencia junto con el tipo de forma de onda actual serán impresos cada segundo a través de la interfaz serial o USB a una herramienta tipo terminal. La Amplitud deberá poder variarse entre 100mV y 2500mV, el Nivel DC entre 50mV y 1250mV. La frecuencia debe poder variarse entre 1 Hz y 12000000 Hz. Por defecto el dispositivo GDS inicia generando una señal de tipo senoidal con una amplitud de 1000 mV, Nivel DC de 500mV y frecuencia de 10Hz. La señal generada y sus características deberán desde luego poder ser corroboradas por un instrumento de medida, multímetro u osciloscopio.

El dispositivo GDS será implementado múltiples veces utilizando diferentes flujos y ambientes de programación disponibles para la Raspberry pi pico. Los flujos y ambientes para los que implementaremos el GDS son:

1. Arduino con polling
2. Micro Python con polling
3. C usando estrategia de polling
4. C usando estrategia de solo Interrupciones
5. C usando polling + interrupciones

El propósito principal de la práctica es contrastar los diferentes flujos de programación y ambientes de desarrollo y estimar la eficiencia de cada uno de estos para generar las señales. Para ser lo más justos posibles en esta práctica restringiremos el uso de librerías a las estrictamente proporcionadas oficialmente y nos abstendremos de usar librerías de terceros. Esto quiere decir que por ejemplo en Arduino estamos restringidos al API oficial de Arduino, en Micro Python al SDK de Python y en C al SDK de C. Si bien la eficiencia de un programa puede medirse de formas diversa, en esta práctica utilizaremos como medida de la eficiencia del flujo de programa y ambiente de desarrollo respectivo, la frecuencia máxima que cada uno de los flujos logre generar cada una de las señales de forma estable.

PROCEDIMIENTO

1. Instalar las herramientas de desarrollo necesarias para llevar a cabo la práctica: Arduino, Micro Python y Visual Studio Code + el SDK de C.
2. Divida el firmware de esta aplicación en módulos funcionales (Generación de forma de onda, Interfaz de usuario).
3. Desarrollar y probar la aplicación de medición de frecuencia en los 5 flujos de programación antes descritos: “Arduino + polling”, “Micro-python + polling”, “C + polling”, “C + interrupciones” y “C + polling + interrupciones”.
4. Documentar en doxygen cada una de las 5 implementaciones.
5. Comparar la facilidad de programación entre los 5 flujos de diseño. Para esto asigne una calificación de 1 a 10 al nivel de facilidad de cada flujo de diseño, siendo 1 muy difícil y 10 muy fácil.
6. Medir la frecuencia máxima que se puede alcanzar con cada forma de onda en cada una de las 5 implementaciones.
7. Comparar el tamaño del programa y el uso de memoria RAM en cada flujo de diseño.
8. Escriba un reporte donde consigne el diagrama de flujo de las 5 implementaciones, discuta las principales diferencias entre los ambientes y flujos de programación utilizados, y los resultados obtenidos en los numerales 5, 6 y 7.
9. Suba los 5 códigos fuente y el reporte en un archivo comprimido a la plataforma Moodle antes de las 23:59 horas del 19/04/2023. Marque el archivo de entrega con los apellidos de los integrantes del grupo. Ej. P03_gomez_velasquez.zip

EVALUACIÓN

- Funcionamiento (50%) – Generador digital de señales funcionando correctamente en los 5 flujos de programación. El éxito de la práctica depende de las 5 implementaciones en los diferentes flujos, no hacer una o más de las soluciones tendrá una penalización en la nota.
- Sustentación (20%) – Respuesta adecuada a las preguntas relacionadas con el desarrollo de la práctica y la comprensión de los flujos de programación aplicados.
- Código fuente (20%) – Correcta estructuración de los diferentes flujos de programación y su documentación en doxygen.
- Reporte (10%) – Uso de la plantilla de reportes del laboratorio, redacción, conclusiones.