

CIIC 4082 – Arquitectura de Computadoras II
Tarea Programación MSP430FR6989 Launchpad
Conversión entre Sistema Decimal y Binario con Punto Fijo
Semestre Agosto-Diciembre 2022

Requisitos generales

Se requiere diseñar un sistema para cambiar la representación de valores entre el sistema decimal y la representación en binario con punto fijo. El sistema se implementará utilizando un Launchpad de Texas Instruments con el MSP430FR6989 y el código estará escrito en lenguaje del ensamblador con el IAR EW.

Al iniciar el sistema aparecerá escrito en la pantalla F(P.Q) para representar el nombre de la aplicación. La Figura 1 muestra cómo luciría el nombre en la pantalla. Al presionar el botón S1 aparecerá el nombre (no apellido) del primer integrante del equipo. El orden de los nombres se seleccionará por el orden alfabético por apellidos de los integrantes. Al presionar S1 nuevamente aparecerá el nombre del segundo integrante y así sucesivamente cada vez que se presione S1. El nombre siempre se comenzará a escribir comenzando por el primer carácter a la izquierda del despliegue. Si el nombre tiene menos de seis letras los espacios que no se utilicen aparecerán en blanco. Si tiene más de seis letras entonces sólo se escribirán los primeros seis caracteres. La Figura 2 muestra cómo luciría el nombre de Pepito. Si de los integrantes asignados para trabajar en su equipo hay uno o más que no están participando no incluya el(los) nombre(s) en esta parte de la aplicación.

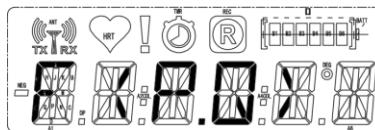


Figura 1. Despliegue mostrando nombre de la aplicación

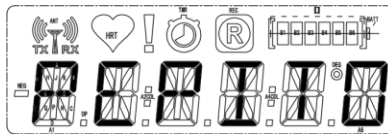


Figura 2. Despliegue mostrando el nombre de Pepito

Cuando se presione el botón S2 la aplicación presentará un menú que le permitirá al usuario configurar cómo se realizarán las conversiones. Tiene que poder indicar lo siguiente:

- Números con signo o sin signo
- Cambio de decimal a binario o de binario a decimal
- Si se entrará un valor decimal hay que indicar si utilizará 1 o 2 dígitos enteros
 - La cantidad de dígitos decimales será $5 - \text{totalDigitosEnteros}$.
 - Si se entrará un valor decimal sin signo entonces el espacio correspondiente al primer caracter en el despliegue no se utilizará y aparecerá en blanco
- Si se entrará un valor binario hay que indicar cuántos bits se utilizarán para la parte entera (p).
 - El número de bits para la parte fraccional se determinará restando $6 - p$.

La Figura 3 presenta un ejemplo de una opción en donde se pregunta al usuario si el valor será con signo (signed) o sin signo (unsigned) y el usuario indicó que se utilizará signo.

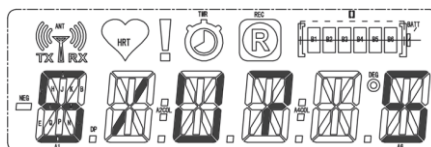


Figura 3: Ejemplo entrada para indicar uso de signo

Todos los números que el usuario entre en decimal se escribirán en uno de los siguientes dos formatos:

- a) sii.ddd
- b) si.dddd

La *s* representa el signo o su ausencia. Para el negativo se escribirá $-$. Para positivo se escribirá $+$. Si el número no tiene signo entonces no utilizará esta posición y estará en blanco. *i* es un dígito de la parte entera. El punto es un punto que aparecerá automáticamente en la posición correspondiente dependiendo de la opción que seleccione el usuario. *d* es un dígito de la parte decimal del número. Todos los valores binarios se escribirán con 6 bits. La Figura 4 muestra un ejemplo de cómo luciría en pantalla el valor -34.567 .

Los diseñadores determinarán cómo se presentará y manejará el menú. Tienen que tomar en cuenta que sólo cuentan con 6 caracteres en el despliegue y que toda entrada por parte del usuario se realizará con los botones S1 y S2. El uso de la aplicación tiene que ser lo más intuitivo posible.

Luego de configurar cómo se realizará la conversión, el sistema proveerá para que el usuario escriba el valor al que desea realizarle la conversión. Luego de realizar la conversión el programa regresará al menú principal al presionar un botón.

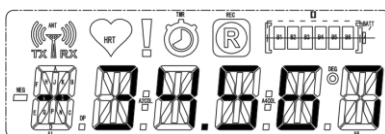


Figura 4. Despliegue mostrando un -34.567

La forma en que se presentará el menú y la forma en que se manejará tiene que diseñarla cada equipo de trabajo. Cada equipo también diseñará el procedimiento para que el usuario escriba los valores. Tome en consideración que el proceso debe ser lo más simple posible, reducir al mínimo el número de veces que el usuario tenga que presionar los botones y facilitar manejar las equivocaciones al entrar un dígito. El equipo le presentará al instructor del curso su propuesta para el menú y entrada de datos. El instructor indicará si acepta el diseño o si hay que hacerle modificaciones. La interfaz y funcionalidad que resulten de este proceso serán las que se implementarán con el diseño. Si fuera necesario realizar modificaciones (porque se encontró que hay fallas de diseño) a la interfaz se necesita la autorización del instructor para implementar la misma. Note que puede tomar tres días laborables para recibir contestación sobre una presentación de diseño o solicitud de cambio. No obtendrán puntos trabajos

que no tengan una propuesta aprobada por el instructor ni trabajos cuya interfaz no funcione según establecido en la propuesta aprobada. Solicitud de cambios a la interfaz o su operación luego de aprobada la misma puede conllevar la pérdida de puntos que en general aumentarán dependiendo de la magnitud del cambio y de cuánto tiempo haya transcurrido desde que recibieron la descripción de la tarea.

Requisitos técnicos

La estructura del código será modular haciendo uso apropiado de subrutinas. Para cada proceso de la aplicación se dedicará al menos una subrutina. Por ejemplo: una subrutina para leer un valor, una para enviar un string de caracteres a la pantalla, una para convertir un string a un valor decimal, etc. Cada subrutina realizará una y sólo una tarea. Cada subrutina tiene que estar documentada indicando: propósito, pre-condiciones, post-condiciones, valores y argumentos que recibe, autor, fecha de creación. Todos los elementos de la documentación tienen que ser correctos y apropiados. Asegúrese de estudiar qué requiere cada uno de los elementos de la documentación.

Los nombres de las etiquetas con los que se hará referencia a las subrutinas tienen que indicar claramente cuál es el propósito de las mismas. Por ejemplo: "leeValor", "presentaMenu".

Bajo ninguna circunstancia la ejecución pasará de una subrutina a otra si no es por medio de una instrucción CALL o porque se ejecute un RET al llegar al final de una subrutina. Un bloque de código que no comience con una etiqueta y termine con un RET no se considera una subrutina. No es aceptable utilizar bloques que no sean subrutinas para modularizar el código (con la posible excepción del programa principal), tener subrutinas que realicen más de una tarea o realizar en el programa principal tareas que no sean distribuir el trabajo a las subrutinas. No cumplir con los requisitos anteriores es equivalente a que el programa no funcione.

Entregables

Reporte escrito del proyecto

- Página de presentación
- Descripción general del trabajo
- Proceso para la solución del problema

- Tabla con tareas realizadas por cada integrante del equipo
- Tabla con por ciento (estimado) del total del proyecto trabajo realizado por cada integrante (la suma de todos tiene que ser igual a 100)
- Instrucciones para operación del sistema: Instrucciones claras, detalladas y con ejemplos de cómo se maneja la aplicación de forma que el usuario pueda realizar los cambios de formatos numéricos que desea. Presuma que para evaluar su aplicación se pedirá la asistencia de personas ajenas al curso a quienes se les proveerá el sistema con su aplicación y copia impresa de sus instrucciones. Si las personas no pueden realizar las tareas de cambio de temperatura con su aplicación se considerará que las instrucciones no son adecuadas.
- El archivo del reporte estará en formato .docx o .doc
- El nombre del archivo del reporte será Grupo##TareaFPQ en donde ## es el número del equipo
- Con la posible excepción de la página de presentación y títulos de secciones utilizará letra Verdana de tamaño 12.

Workspace generado por IAR. Todos los archivos y folders del workspace mas un archivo tipo texto con el código del programa estarán comprimidos en un archivo con el nombre Grupo##TareaFPQ

Video mostrando

- el uso y manejo del menú para configurar cómo se realizará el cambio de representación de los valores
- el procedimiento para entrar los valores
- resultados del cambio de un sistema a otro para representar los valores
- cómo regresar al menú luego de mostrar el resultado

El vídeo tendrá una duración máxima de 4 minutos con voz fácil de entender y a velocidad normal. El vídeo no explicará código, sólo lo que se indica arriba (con varios ejemplos). El vídeo estará en formato mp4 e incluirá voz como parte de la explicación. El nombre del archivo del vídeo será Grupo##TareaFPQ. Antes de hacer la toma final del vídeo pruebe con diferentes resoluciones y seleccione la más baja que permita apreciar lo que quiere mostrar. De esta forma el archivo del vídeo no será más grande de lo necesario. Subirá el vídeo a la nube (por ejemplo, Google Drive) e incluirá un enlace al mismo desde la primera página del reporte. El enlace tiene que ser público y tiene que funcionar durante al menos todo el semestre. Si no se puede acceder al vídeo con el enlace será equivalente a no haber entregado el vídeo. Presuma que subir su vídeo puede tomar varias horas. Así que separa al menos 8 horas para subir su trabajo.

Auto Evaluación y Evaluación de los Pares: Completará y enviará con su trabajo la hoja electrónica para la auto evaluación y evaluación de los pares. La evaluación de los pares se utiliza como factor para ajustar la puntuación de cada integrante del equipo. Si no envía la misma se entenderá que está de acuerdo con la evaluación que envíe el resto del equipo. El nombre del archivo será Grupo##EvalSuNombre en donde su nombre será su primer nombre, seguido de su primer apellido sin utilizar espacios en blanco, puntos, acentos o tildes. Por ejemplo. Si su nombre es Rincón Mayagüez Añasco y su grupo es el 21 el nombre del archivo será Grupo21EvalRinconMayaguez.

Entrega del trabajo: Todos los archivos requeridos se comprimirán en uno de tipo .zip o .rar con el nombre Grupo##TareaFPQ. Ése será el archivo que cada estudiante entregará como evidencia de su trabajo.

Evaluación

Fase I

[00] Propuesta de interfaz

[10] Interfaz funciona correctamente y es fácil de manejar

[05] Tiempo de respuesta de botones es apropiado

Fase II

[10] Reporte

[10] Documentación del código

[10] Estructura del código, rutinas con uno y sólo un objetivo

[50] Sistema funciona de forma correcta en presentación, operación y resultados

[10] Vídeo cumple con lo requerido

[05] Se entregan todos los archivos requeridos y los nombres y formato de los mismos cumplen con los requisitos

Presentaciones individuales o grupales (a ser determinado) con defensa del trabajo. La defensa puede ser oral, escrita o ambas.

Puede haber una prueba escrita o práctica sobre conocimientos y destrezas relacionadas con la tarea.

* Si no puede presentar o defender su trabajo no obtendrá puntos en el mismo