INGENIERÍA DE SISTEMAS ELECTRÓNICOS

Curso 2019-2020
BLOQUE 2

Ignacio Blasco Hernandez
Jorge Rosales Martín
GRUPO V1V2V3 - PUESTO 3

Departamento de Ingeniería Telemática y Electrónica
Universidad Politécnica de Madrid

Índice del documento

1 OBJETIVOS DEL BLOQUE

1.1 Resumen de los objetivos del bloque

El objetivo de este bloque es la realización de un sistema electrónico de mediana complejidad para medir la temperatura a través de una RTD, y procesar la temperatura a través del microcontrolador LPC1768 para mostrarlo en la pantalla LCD de la mbed, enviarlo a través del puerto serie al ordenador, mostrarlo en la página web y guardarlo en la memoria Flash del micro. Para la realización de lo mencionado anteriormente se han logrado los siguientes objetivos de aprendizaje:

-Capacidad de realizar la especificación, implementación, documentación y puesta a punto de equipos y sistemas electrónicos, considerando tanto los aspectos técnicos como las normativas reguladoras correspondientes.
-Capacidad para realizar diseño de circuitos y sus pruebas a través de programas como el Orcad Pspice sin disponer de todos los componentes.
-Capacidad para calcular los errores de medida generados en un sistema electrónico y su repercusión en las medidas.
-Capacidad de diseñar circuitos de electrónica analógica y digital, de conversión analógica-digital y viceversa para aplicaciones de telecomunicación y computación.
-Capacidad de búsqueda y selección de información, de razonamiento crítico y de elaboración y defensa de argumentos dentro del área.
- Habilidades para la utilización de las Tecnologías de la Información y las Comunicaciones.

1.2 Resumen de los objetivos de las fases realizadas

A continuación, describiremos qué objetivos se han alcanzado en cada fase y cómo se han alcanzado dichos objetivos.

1.2.1 Objetivos de la fase 1

El objetivo de la practica 1 es diseñar un circuito analógico y realizar mediante componentes reales disponibles en el mercado una propuesta de implementación.

Para ello se han diseñado tres subsistemas analógicos. Primero un sistema de regulación de alimentación para tener una tensión regulada y lo más similar posible a los valores teóricos y reducir con ello los errores. Segundo se ha diseñado un circuito de acondicionamiento de la señal para obtener la medida de la tensión en la RTD y pasar la medida al microcontrolador. Se ha escogido la medida a tres hilos debido a los beneficios económicos y de calidad de la medida con respecto a las otras opciones posibles. Y por último se ha diseñado un circuito de conversión voltaje a frecuencia. Además del diseño de todo el circuito se ha realizado el cálculo teórico de errores introducidos por los cables y de las tensiones para los distintos rangos de temperatura.

Por último, se han realizado las simulaciones de los subsistemas analógicos, se ha realizado una comparación de los errores a partir de los valores teóricos calculados y se ha ajustado de manera simulada las características del circuito para conseguir una precisión de +/- 0,3 ºC.

Con todo lo que se ha comentado anteriormente que se ha desarrollado, los objetivos de aprendizaje que se han alcanzado son: Capacidad de diseño y simulación de circuitos electrónicos a través del programa Orcad Pspice, capacidad de calcular los errores del circuito, capacidad para búsqueda y selección de información a la hora de encontrar los componentes adecuados para el diseño de los circuitos electrónicos.

1.2.2 Objetivos de la fase 2

En la fase 2 se ha procedido al diseño software de todo el circuito para cumplir todos los requisitos de medida y presentación de las medidas. Para ello se ha conseguido realizar los objetivos de: presentar las medidas en el LCD del microcontrolador y realizarlo periódicamente cada 3 segundos, guardar la medida en la memoria Flash del microprocesador con el formato de hora y temperatura correspondiente, enviar la información de la medida y fecha de la medida a través del puerto serie RS232 al ordenador, poder seleccionar el modo de medida a través del cliente web y mostrar el color del RGB correspondiente a la temperatura medida.

Con todas las acciones realizadas previamente se han conseguido alcanzar los siguientes objetivos de aprendizaje: Se ha adquirido la capacidad de construir una aplicación que cumple con los requisitos pedidos.

1.2.3 Objetivos de la fase 3

En esta fase a la funcionalidad desarrollada en la fase 2 se ha añadido la simulación de la señal de entrada analógica a través del uso del DAC integrado en el microcontrolador. De manera que con el empleo del joystick se puede controlar el nivel de la señal simulada comprendido entre los niveles correspondientes determinados en las fases anteriores.

Con la realización de lo anteriormente comentado se han conseguido alcanzar los siguientes objetivos de aprendizaje: Capacidad para generar una señal analógica mediante el uso del DAC y de la medida de una señal mediante el ADC.

1.2.4 Objetivos de la fase 4

Para la fase 4 se generado una señal cuadrada para simular mediante el convertidor de V/F la medida de la temperatura a partir de la frecuencia. Para ellos se ha escogido el pin correspondiente para la simulación de la señal cuadrada y se ha unido al pin de entrada. Además, se ha implementado la funcionalidad de los joysticks para poder aumentar o disminuir la frecuencia de la señal.

Con todo lo que se ha realizado en esta fase se han alcanzado los siguientes objetivos de aprendizaje: Se adquirido la capacidad de búsqueda de información y diseño de circuitos para elegir el pin correspondiente para generar la señal cuadrada, también se ha adquirido la capacidad para implementar y poner a punto un circuito electrónico.

1.2.5 Objetivos de la fase 5

En esta fase final recogiendo y aplicando todas las funcionalidades y aplicaciones realizadas en las fases anteriores, se ha logrado recibir comandos desde el ordenador a través del puerto serie RS232. Al recibir estos comandos, donde se indica la temperatura a simular, se ha conseguido generar las señales correspondientes para simular esa temperatura tanto en voltios como en frecuencia.

Gracias a lo mencionado anteriormente se han logrado alcanzado los siguientes objetivos de aprendizaje: Uso de la UART, tanto recepción como envío de información, además se han adquirido los conocimientos necesarios para poder configurar un programa mediante este periférico, de modo que, mediante el envío de distintos comandos, sería posible configurar el sistema de una manera o de otra.

1.3 Acrónimos utilizados

Identifique los acrónimos usados en su documento.

RTC	Real Time Clock
SNTP	Simple Network Time Protocol
DAC	Digital to Analog Conversor
V/F	Voltage to Frecuency Conversor
ADC	Analog to Digital Conversor
LCD	Liquid cristal display
UART	Universal Asynchronous Receiver-Transmitter
RTD	Resistance Temperature Detector
AI	Amplificador de Instrumentación
RGB	Led red, green, blue
SW	Switch (Pulsador)
PWM	Pulse width mudulation

1.4 Tiempo empleado en la realización de cada una de las prácticas.

En cuanto el tiempo empleado, se han empleado aproximadamente 55 horas, entre los dos, de las cuales, se han empleado 15h (Este tiempo se deriva de pensar el circuito, buscar los componentes, simular y realizar la memoria) para la realización de la fase 1, 25h para la realización de las fases 2 y 3 (Al tener que desarrollar la aplicación casi en su totalidad se ha necesitado más tiempo). 5 horas para la realización de la fase 4 y otras 5 para la realización de la fase 5 (en estas dos últimas fases al ser relativamente más sencillas que las anteriores, no nos hemos encontrado con demasiadas dificultades, y su realización ha sido bastante efectiva). Por último, se han empleado aproximadamente 5 horas para la realización de la memoria

1.5 Bibliografía utilizada

[RD1] Manual del Lpc1768: https://www.nxp.com/docs/en/user-guide/UM10360.pdf

[RD2] Documentación de CMSIS: https://arm-software.github.io/CMSIS_5/Core/html/index.html

[RD3] Ejemplos de la placa MCB1000

[RD4] Tutorial para el Timer
http://www.ocfreaks.com/lpc1768-timer-programming-tutorial/

[RD5] Tutorial para la UART
http://www.ocfreaks.com/lpc1768-uart-programming-tutorial/
https://www.electronicshub.org/how-to-program-uart-in-lpc1768/

[RD6] Tutoriales para el LPC1768:
https://www.exploreembedded.com/wiki/index.php?search=LPC1768&title=Special%3ASearch

[RD7] Esquemático del LPC1768:
https://os.mbed.com/media/uploads/chris/lpc1768-refdesign-schematic.pdf

[RD8] Esquemáticos de la placa de expansión:
https://os.mbed.com/media/uploads/chris/mbed-014.1_b.pdf

1.6 Autoevaluación.

Autoevalúese de forma individual comprobando los objetivos de aprendizaje indicados en la guía de la asignatura. Si ha encontrado dificultades en la realización de alguna de las prácticas, indíquelo en este apartado.

Autoevaluación estudiante A: Ignacio Blasco Hernández

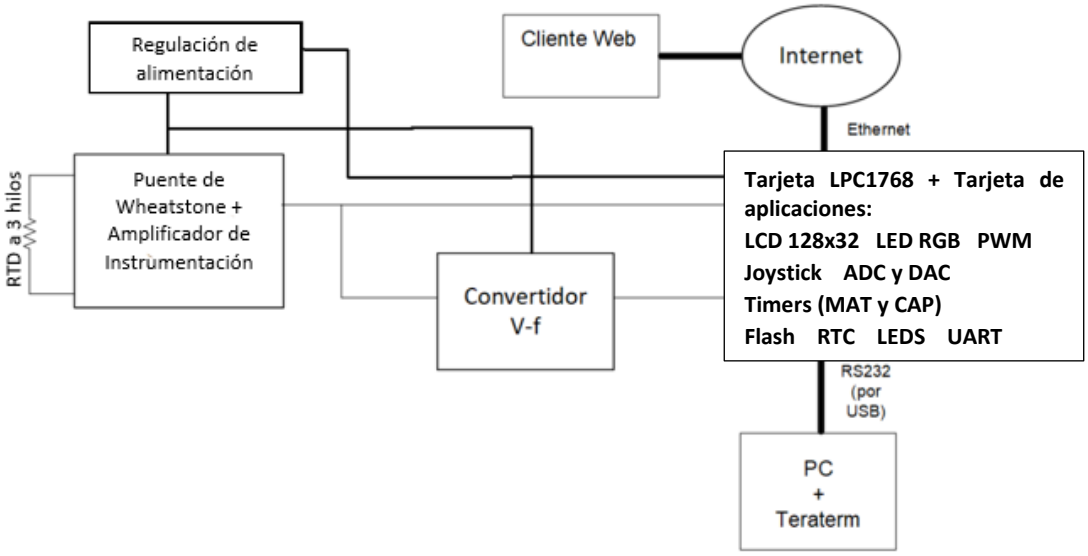
Personalmente creo que he alcanzado algunos de los objetivos de aprendizaje que se indican en la guía, creo que soy capaz de realizar un proyecto cumpliendo con las especificaciones requeridas, y de elaborar documentación técnica. Además, creo que he mejorado mi capacidad para analizar y solucionar problemas encontrados, así como mi capacidad para escribir código funcional y robusto.
Además, he suplido los problemas que encontré durante la realización del bloque 1 y he sido capaz de solventar todas las dificultades que he encontrado en este bloque de manera autónoma.
Por último, creo que he sido capaz de junto con mi compañero, diseñar un sistema hardware y software que cumple con las especificaciones que se nos han pedido.

Autoevaluación estudiante B: Jorge Rosales Martín

En este bloque de la asignatura pienso objetivamente que he alcanzado los objetivos de aprendizaje propuestos para esta parte de la asignatura. La búsqueda de información en datasheets, manuales y videos explicativos para poder afianzar los conocimientos de cada elemento del circuito ha sido clave en este proyecto. También se ha mejorado la capacidad de trabajo colaborativo junto a mi compañero para poder solventar cualquier duda o imprevisto que ha surgido.
Para finalizar, he conseguido el objetivo principal, según mi opinión, de esta asignatura y es poder realizar desde cero un proyecto buscando los materiales necesarios entre los que se encontraban disponibles en el mercado, y realizando un proyecto que cumple con todas las funcionalidades.

2 RECURSOS UTILIZADOS DEL MICROCONTROLADOR

2.1 Diagrama de bloques hardware del sistema completo



2.2 Justificación de las soluciones adoptadas.

A continuación, se van a describir las soluciones que se han escogido para diseñar el circuito teniendo en cuenta la imposibilidad de realizar el diseño físico del circuito total tal como se describe en la fase 1.

De la tarjeta NXP LPC1768 se han utilizado los siguientes recursos:

- Para poder dotar de sistema operativo, conexión ethernet y pines de entrada/salida es necesario activar las siguientes opciones en las opciones del menú Manage Run-Time Environment del Keil:
    - CMSIS->CORE
    - CMSIS->RTOS2(API)-> Keil RTX5
    - CMSIS Driver->Ethernet MAC (API) -> Ethernet Mac
    - CMSIS Driver->Ethernet PHY (API) -> DP83838C (Driver Ethernet del LPC1768)
    - CMSIS Driver->SPI->SSP
    - DEVICE->GPDMA
    - DEVICE->GPIO
    - DEVICE->PIN
    - DEVICE->Startup
    - NETWORK-> CORE
    - NETWORK-> INTERFACE->ETH (1)
    - NETWORK->SERVICE->SNTP
    - NETWORK->SERVICE -> WEB SERVER COMPACT
    - NETWORK->SOCKET-> TCP
    - NETWORK->SOCKET->UDP

- El RTC se ha configurado para generar una interrupción cada segundo, poniendo el registro LPC_RTC->CIIR =0x01, esto produce que cada vez que se incrementa en 1 la cuenta de los segundos, se genera una interrupción: Por otro lado, se ha configurado una alarma cada minuto, es decir, cada vez que los segundos pasan a 0, se activa la alarma, esto se consigue con la siguientes líneas;

LPC_RTC-> AMR =0xFE; (Para que solo compare los segundos) y PC_RTC->ALSEC=0 (valor de los segundos a comparar).

- Es necesario aumentar el número de hilos que vienen por defecto ya que se utilizan más de los indicados, para esto, en el archivo RTX_config.h, se ha marcado el checkbox Thread Configuration->Object specific memory allocation y aumentar el número de Number of user thread lo que sea necesario, en nuestro caso, 11 (no se usan todos, se pone 11 por comodidad). También puede resultar necesario aumentar la memoria predeterminada.
- Por otro lado, para configurar la conexión a internet, es necesario configurar en el archivo RTE_Device.h seleccionando el checkbox ENET(Ethernet Media Interface), y desplegando el menú, seleccionar en MIIM (Management Data Interface)->ENET_MDC Pin el P1_16 y en MIIM (Management Data Interface)->ENET_MDIO Pin el P1_17. A continuación, en el archivo Net_Config_ETH_0.h indicar una dirección Mac, IPv4->IP address, IPv4->subnet mask válidos. Y deshabilitar la función Dynamic Host Configuration Protocol, para utilizar una dirección ip fija. Conviene destacar que a la hora de conectarlo al router puede ser de utilidad marcar esta opción, para que el router le asigne una dirección valida, y, con una aplicación (fing, por ejemplo) que escanee las direcciones ip que hay conectadas a la red, muestre la que asigna al Lpc1768.
- Sntp (No es un componente físico, pero por simplicidad lo tomamos como un periférico), es necesario configurar una dirección ip válida para que devuelva la petición. Esto se puede hacer o bien en el archivo Net_Config_SNTP_Client.h, escribiendo la dirección Ip, o escribiendo la siguiente sentencia:

  const NET_ADDR4 ntp_server = {NET_ADDR_IP4, 0, 130, 206, 3, 166}
- Se va a utilizar el DAC integrado en el microcontrolador para realizar la simulación de la señal analógica de entrada. Para ello conectamos la salida del DAC a la entrada del ADC, pin 18 a pin 17 del microprocesador. De esta manera tendremos un buffer circular del código de la simulación de varias medidas de temperatura para que mediante la pulsación del joystick arriba o abajo se vayan aumentando o disminuyendo las temperaturas simuladas.
- Se ha utilizado el ADC integrado en el microcontrolador para recibir la medida de la temperatura (que en este caso será una simulación realizada a través del DAC) y pasarla a una señal digital.
- El LED RGB se ha configurado para para una temperatura igual o superior a 25 ºC luzca de color rojo y cuando sea inferior de azul, aumentando la intensidad de los colores según la temperatura medida.
- Se va a utilizar el joystick para que realice ciertas acciones según el tipo de pulsación que se produzca. De manera que se van a utilizar las pulsaciones arriba y abajo para aumentar o disminuir la temperatura simulada, si esta medida a través de la tensión se modificaría la tensión simulada y si esta simulada a través de la frecuencia se modificaría la frecuencia. Para ello se ha realizado la configuración reutilizando el código de SBM.
- Se va a utilizar el código usado en la asignatura de SBM para representar en la pantalla LCD la temperatura medida y la frecuencia o la tensión generadas para la simulación de la medida.
- También se utiliza un Timer para simular el convertidor de voltaje a frecuencia de manera que se varía la frecuencia según las pulsaciones del joystick. Y según la frecuencia del Timer se pasa a la temperatura simulada.

3 SOFTWARE

3.1 Descripción del funcionamiento de la aplicación completa.

A continuación, se va a realizar la explicación de la aplicación al completo con las distintas tareas implementadas, la sincronización realizada entre los componentes del proyecto y las interrupciones manejadas.

La aplicación consta de cuatro partes principales que son:

- Main.c encargado de la inicialización del de la aplicación, en concreto inicializa el kernel de SO, los leds y el adc, además de iniciar un hilo (app_main) que dará paso al resto del programa. Si bien es cierto que la inicialización de los leds y del adc se podría haber hecho más adelante, se ha puesto ahí por simplicidad.
- HTTP_Server.c, contiene hilos encargados de la realización de todas las tareas de la aplicación y de la comunicación de los periféricos.
- HTTP_Server_CGI.c, fichero encargado de leer y escribir valores en la página web.
- El resto de los ficheros son de periféricos (lcd.c, lcd.h, adc.c, adc.h, led.c, led. h, …) que contienen funciones de inicialización, configuración, lectura y escritura, en resumen, funciones que permiten utilizar dichos periféricos.

Ya que las tareas se encuentran principalmente en fichero HTTP_Server.c, vamos a describir detalladamente el funcionamiento de las tareas más importantes que se han implementado en este segundo bloque, omitiendo la explicación de las partes ya explicadas en el bloque 1. El hilo principal es app_main() desde el que se inicializan el resto de los hilos, y que se arranca desde el fichero main.c tras iniciar el core del sistema operativo.

Las tareas que se realizan, ordenadas según su línea de aparición, son:

- Display(), este hilo se utiliza para, primeramente, inicializar el LCD y guardar la dirección MAC y la dirección IP en la memoria Flash (Dirección 0x78000). Después escribe en la pantalla la temperatura y dependiendo de si el modo de recepción de la temperatura está en tensión o en frecuencia representará la tensión o frecuencia correspondiente. Además, se aprovecha para realizar la transmisión a través del puerto serie para transmitir la temperatura y la fecha en la que se ha realizado, cabe destacar que para hacer el envío de la trama con el timestamp se utiliza una función auxiliar que devuelve un array de caracteres cuyo contenido tiene el siguiente formato (entre paréntesis el número de dígitos, incluido decimales):

  Dia(2),Mes(2), Año4), Hora(2), Min(2), Seg(2), signo(1), temperatura(3+1punto), formato(1)

  Donde signo será 1 cuando la temperatura sea negativa y cero en caso contrario, y formato será 0 en caso de adquisición en tensión y 1 en caso de adquisición en frecuencia.

- SW_Thread(), en este hilo se han concentrado muchas de las funcionalidades desarrolladas a lo largo del bloque. Aparte de realizar la inicialización del joystick también se realiza la habilitación e inicialización del DAC y del timer que se encargarán de realizar la simulación de las señales analógicas de entrada tanto en tensión como en frecuencia. También se realiza la gestión de pulsaciones y eliminación de los rebotes, para cada pulsación se gestiona:
  - UP y DOWN: Aumentar o disminuir el valor de la tensión o la frecuencia, según el formato en el que se encuentre la medida. Se recorre un buffer circular con distintos valores predeterminados, tanto de tensión como de frecuencia.
  - LEFT Y RIGHT: Aumentamos/disminuimos y actualizamos la frecuencia de la medida de la temperatura, se recorre un buffer circular con 6 tiempos predefinidos de 0.5s, 1s, 2s, 3s, 4s, 5s.

- CENTER: Guardamos el valor de la temperatura y la fecha de la medida en la memoria Flash. Si ya hemos realizado diez escrituras de datos, borramos la primera medida y la sobrescribimos con los nuevos datos (conocido como buffer circular).
  El formato que se almacena es el siguiente (entre paréntesis el número de bytes):
  Dia(1),Mes(1), Año(2), Hora(1), Min(1), Seg(1), signo(1), temperatura(2), formato(1)
  Por simplicidad, como hay espacio suficiente, el signo y el formato ocupan 1 byte cada uno, si hubiera problemas de espacio, sería conveniente unirlos en un byte, por otro lado, como notación se ha utilizado un 1 para representar que la temperatura es negativa (signo =0x01), y un 0 para temperaturas positivas, además 0x00 en formato cuando la adquisición sea tensión y 0x01 cuando la adquisición sea en frecuencia.

- rec_Uart(), en esta tarea se reciben los datos de la trama que se envían desde el PC a través del puerto serie RS232. En esta trama se recibe la temperatura y con ella se pasa a la tensión y frecuencia correspondiente y se actualiza los valores simulados de las señales de entrada. Por último, se envía una trama indicando la temperatura, tensión y frecuencia que se ha actualizado. Para recibir y leer la trama correctamente es necesario que se encuentre en este formato:
  "SIMULA Temp=-XX.X ºC" o "SIMULA Temp= XX.X ºC"
  Además, se ha añadido una función que permite borrar el buffer en caso de que la trama sea errónea, o que el buffer (por haber metido un carácter sin querer) este desalineado, esto se hace pulsando la tecla r/R. De manera que al enviar la trama la tensión se hace de manera errónea bastará con pulsar la r y enviar la trama de nuevo, también vale enviar la trama con una r/R al principio, y sin espacio de separación:
  "rSIMULA Temp= XX.X ºC" o "RSIMULA Temp= XX.X ºC"

Para la sincronización de todas las partes del proyecto se han empleado señales de sincronización. No se va a realizar la explicación de señales de sincronización procedentes del bloque 1. A continuación, explicaremos cuales son las señales y el uso que las ha dado:

- ➔ En la tarea Display se necesitaba una sincronización para poder saber cuándo actualizar la información a mostrar por la pantalla LCD. Para ello se usó un osThreadFlagsWait (0x01U, osFlagsWaitAny, osWaitForeve y osThreadFlagsSet (TID_Display, 0x01) para esperar recibir la señal y enviarla. Esto se realizará cada vez que se modifique la temperatura o el modo de obtener la medida.
- ➔ Desde la tarea SW_Thread(), aparte de esperar a recibir la señal de la pulsación, cada vez que se realiza una pulsación se envía una señal al hilo Display() para actualizar lo que se muestra por pantalla.
- ➔ Para la sincronización de la recepción de una trama a través del puerto serie se utiliza una señal que se enviará, cuando se produzca la interrupción generada al enviar la trama, al hilo rec_UART() que realizará el análisis de los datos enviados desde el PC una vez recibida la señal. Si la trama es correcta enviará una señal al hilo Display para que represente la nueva temperatura.

A continuación, se va a analizar los ficheros de los periféricos que se han empleado en este proyecto, se va a omitir la explicación de los periféricos ya analizados durante la memoria del bloque 1. Por lo tanto, los ficheros que se han utilizado son:

- Uart.c y Uart.h, librería que permite inicializar la UART y recoge las funciones necesarias para poder realizar transmisiones y recepciones de caracteres a través del puerto serie RS232.
- DAC.C Y DAC.H, librería que recoge la inicialización del DAC y las funciones necesarias para poder configurar una tensión simulada a la salida del DAC.
- Timer.c() y Timer.h, librería donde se recoge la inicialización del Timer y las funciones a través de las cuales se puede obtener la frecuencia a la que está funcionando o establecer la nueva frecuencia de funcionamiento.
- RGB.c y RGB.h, librería que recogen las funciones necesarias para que, a través del uso del PWM, se encienda el led RGB en azul o en rojo con distintas intensidades.

Para la temporización se han usado los osThreadFlagsWait (0x80U, osFlagsWaitAny, 100) a modo de Timeout para esperar un cierto tiempo (100ms en este caso) sin que paralice la ejecución del resto de los hilos ya que no es bloqueante.
Por último, se han utilizado dos timers periódicos que llevan el control de la hora y del refresco de la pantalla, en concreto, se utiliza un timer que lanza una petición al servidor SNTP para obtener los segundos en epoch y otro timer (Timer_LCD_func) cuya función es obtener la medida correspondiente y mandar una señal al hilo de Display para que haga sus funciones. Este último timer, se destruye y se vuelve a crear con un tiempo distinto cada vez que se pulsa el botón derecho o el izquierdo con el fin de variar la frecuencia con la que se obtiene la medida.

En este proyecto se manejan cuatro tipos de interrupciones. Primero tenemos las interrupciones generadas por las pulsaciones del joystick (**EINT3_IRQHandler** ()) donde realizamos el tratamiento de los rebotes y enviamos la señal al hilo SW_Thread para realizar las acciones correspondientes a cada tipo de pulsación.
Después tenemos las interrupciones generadas por la UART. Estas interrupciones se generan cada vez que se recibe una trama a través del puerto serie RS232 desde el PC. Dentro de la interrupción (**UART0_IRQHandler ()**) se almacenan todos los caracteres recibidos y cuando se ha recibido toda la trama se envía la señal al hilo rec_Uart() donde se realiza la conversión de la trama. En este punto, cabe destacar que cuando se recibe una trama, se responde con otra que tiene el siguiente formato:
T=xx.x; V=y.yyyy; F=zzzzzzz, donde la tensión tendrá una resolución de 4 decimales ya que el ADC tiene una resolución de 12 bits, y por tanto el escalón mas pequeño es de 0.0008V, de este modo obtenemos la máxima precisión posible, la frecuencia será sin decimales hay que está expresada en Hz, y la frecuencia con la que trabaja el convertidor este entorno a los kHz.

También utilizamos la interrupción generada por el Timer de la simulación del convertidor V/F que se genera cada vez que pasa el tiempo establecido y donde realizamos la limpieza del flag de interrupción y se calcula el tiempo que ha pasado desde la anterior interrupción, de manera que, se obtiene el periodo de la señal que se quiere medir. Cabe destacar que para simular la señal cuadrada de frecuencia variable y para capturarla, se han utilizado el timer 3 en modo match (MAT3.0, pin 28) que cada vez que llegue al fin de cuenta invertirla la salida del pin 30 (modo toggle) y por otro lado, se ha utilizado el timer 2 en modo captura (CAP2.0, pin 30) para que genere una interrupción cada vez que se detecte un flanco de subida.
Por último, tenemos la interrupción generada por el RTC que mantiene la misma funcionalidad que la explicada en el primer bloque de la asignatura.
En cuanto a la configuración de la UART, se ha configura con un baudrate de 9800, 8 bits, 1 bit de stop y sin paridad, además se genera una interrupción cada vez que se recibe un carácter, cuando el buffer de lectura está lleno, se indica enviando una señal al htt_server de que se ha recibido la trama completa, para que este lea el buffer.

A continuación, se explicará cómo se ha configurado la web y como se hace el cambio en la adquisición de la medida.
La nueva página se ha creado en base a la ya existente del LCD, dado que no se iba a utilizar, se ha escrito sobre este fichero, Lcd.cgi. Lo que se ha hecho es reordenar la disposición de los elementos y cambiar los títulos y etiquetas de éstos. Además, se han añadido un menú desplegable que permite seleccionar entre tensión y humedad, esta parte del código se ha obtenido del fichero leds.cgi, que tal y como estaba codificado resultaba perfecto para esta aplicación. Por otro lado, se ha añadido el código necesario para que la pagina se refresque continuamente de manera que el valor mostrado en el LCD sea el mismo que el de la web. Este código es el mismo que el que se usó en la página de la fecha y hora datetime.cgi.
Por otro lado, en el fichero HTTP_Server_CGI, se ha añadido el código necesario para poder enviar los valores de temperatura, tensión y frecuencia a la web y para recibir la selección del menú sobre el modo de adquisición.

Por último, se comentarán como se han realizado todas las conversiones de, tanto de tensión como de frecuencia.

Obtención de la medida de tensión a través del ADC:

Lo primero es obtener el valor de medido a través del canal 2 del ADC mediante la función medida_adc, como este valor indica un nivel de tensión entre 0 y 3.3V con una resolución de 12 bits, es necesario obtener la tensión que representa dicho nivel, para esto utilizamos la siguiente relación: $medidaTension = Medida_{ADC} * \left(\frac{3.3}{2^{12}}\right)$, a continuación para obtener la temperatura a la que equivale dicha tensión se realiza el siguiente proceso;

1) Restar Offset y dividir por la ganancia: $\frac{medidaTension - Voff(0.4)}{ganancia(13.923)} = V_{AI}\ tensión\ a\ la\ entrada\ del\ AI$

2) Resistencia de la RTD: $V_{AI} = V_D - V_N \rightarrow V_D = V_{AI} + V_N \rightarrow \frac{2.5*RTD}{RTD+1000} = V_{AI} + \frac{2.5*923}{1923} \rightarrow RTD = \frac{1000}{\frac{1}{\frac{V_{AI}}{2.5}+\frac{923}{1923}}-1}$

3) Temperatura: $RTD = 1000(1 + 0.00385 * T) \rightarrow T = \frac{\left(\frac{RTD}{1000}-1\right)}{0.00385}$

Por otro lado, para obtener la medida de temperatura a través de la frecuencia se utiliza la siguiente función que devuelve un valor de frecuencia en KHz, y aplicando la siguiente relación que permite convertir la frecuencia obtenida en un valor de tensión: $medidaV\_Freqz = \left(\frac{Medida_{TIMER}}{1000} - 1\right) * \frac{3.3}{0.8}$ (Esta fórmula viene del apartado donde se explica el convertidor V/F

Por último, una vez se tiene el valor de tensión se aplican los pasos 1), 2) y 3) explicado anteriormente. Por otro lado, cabe destacar que la lectura del periodo de la señal introduce un error en la ganancia, que se solventa multiplicando la medida del periodo por 2.78027465, que elimina el error. La obtención de este valor viene tras medir distintos periodos, observamos que presentaban un error, y al dividir el valor ideal por el real, todas las medidas tenían la misma relación de 2. 78027465.

$\frac{T_{ideal}}{T_{real}} = relacion = 2.78027465$

Por otro lado, cuando se recibe la trama por la UART, al recibir la temperatura o al pulsar los botones que permiten cambiar la temperatura, es necesario realizar el proceso contrario al explicado anteriormente. En concreto, se realiza lo siguiente:

1) Obtención de la RTD: $RTD = (1000.0 * (1.0 + (0.00385 * Temperatura)))$;

2) Obtención de la tensión a la entrada de AI:

$$V_D - V_N = tension = 2.5 * ((RTD/(RTD + 1000.0)) - (923.0/1923.0))$$

3) Obtención de la tensión a la salida del AI: $V_O = (V_D - V_N) * 13.923) + 0.4$

4) Calculo del código binario para el ADC: $Codigo_{DAC} = V_o * \frac{1024}{3.3}$ (El DAC es de 10 bits)

5)Calculo del periodo para el timer: $\frac{1}{1000000.0*\left(0.1+\left(V_O*\left(\frac{0.8}{3.3}\right)\right)\right)} * 10^9$ (en ns)

Por último, para ajustar la intensidad en el RGB

1) Para ajustar la intensidad cuando está en rojo se utiliza la siguiente formula:

$$255 * \left(1 - \frac{temperatura * 2 - 30}{100}\right)$$

2) Para ajustar la intensidad cuando está en azul:

$$255 * \left(1 - \frac{temperatura * (-1.6) + 58.9}{100}\right)$$

Ambas están calculadas de manera que el máximo de intensidad 90% se de para los valores extremos de temperatura 60ºC y -20ºC y que el mínimo, 20% se de para los 25ºc en rojo y 24.9ºC en azul. De modo que para 60ºC y -20ºC tenemos que:

ROJO MAX: $255\left(1 - \frac{60*2-30}{100}\right) = 25.5 \rightarrow 100 - \frac{25.5}{255}x100 = 90\%$

AZUL MAX: $255\left(1 - \frac{-20*-1.6+58.9}{100}\right) = 23.2 \text{->} 100 - \frac{23.2}{255}x100 = 91\%$

El 90% obtenido corresponde a una señal PWM con un con un ciclo de trabajo del 10% a nivel alto, cabe destacar el que RGB es activo a nivel bajo y por tanto se encenderá durante el nivel bajo del PWM

Para los mínimos de intensidad tenemos que:

ROJO MIN: $255\left(1-\frac{25*2-30}{100}\right)=204\rightarrow\ 100-\frac{204}{255}x100=20\%$

AZUL MIN: $255\left(1-\frac{25*-1.6+58.9}{100}\right)=207$->$100-\frac{207}{255}x100=19\%$

Si bien es cierto que el azul oscila entre 91% y 19%, creemos que esta diferencia es inapreciable y por tanto lo damos como bueno.

Para terminar ese apartado, la siguiente tabla indica las conexiones y pines que hay que realizar para poder simular la tensión y la frecuencia y leerlas de manera correcta.

Elemento	Pin	Función	Conexión
ADC	AD0.2 DIP17	Leer un valor de tensión	Conectar estos dos pines con un cable
DAC	AOUT DIP18	Poner un valor de tensión a elección del usuario	
Timer2	CAP2.0 DIP 30	Contar el tiempo que transcurre entre flancos de subida	Conectar estos dos pines con un cable
Timer3	MAT3.0 DIP 28	Generar un reloj de frecuencia variable a selección del usuario	

3.2 Descripción de los proyectos de Keil de cada una de las prácticas.

Proyecto	*Funcionalidad*	*Origen*
Fase 2	En este proyecto se implementan todas las funcionalidades SW descritas en el enunciado. Se representan las medidas en la pantalla LCD, se simula una temperatura fija, se guarda la información en la memoria Flash a partir de la pulsación del botón central del joystick, se envía la información a través del puerto serie RS232 al PC, se enciende el RGB con el color (azul o rojo) e intensidad correspondientes según la temperatura simulada y, por último, desde el cliente WEB se puede escoger el método de medida (frecuencia o tensión).	Proyecto final del bloque 1
Fase 3	En este proyecto se ha añadido la señal de entrada analógica simulada al proyecto de la fase 2. A partir del DAC se simulan las tensiones correspondientes a las temperaturas medidas y se conecta al ADC. Se representa en la pantalla LCD la tensión actual.	Proyecto de la fase 2
Fase 4	En esta fase se ha implementado la simulación del convertidor de voltaje a frecuencia. Se ha adaptado el proyecto para poder elegir a través del cliente WEB la manera de obtener la temperatura (tensión/frecuencia) y se ha procedido a añadir la representación en el LCD la frecuencia cuando está en ese formato de medida.	Proyecto de la fase 3
Fase 5	Añade a la funcionalidad de la fase 4 la recepción de tramas procedentes del PC a través del puerto serie RS232. Con la trama recibida se recoge la temperatura y se introduce en las señales simuladas de tensión y frecuencia. Y, por último, se envía una trama con la nueva temperatura, tensión y frecuencia.	Parte del proyecto de la fase 4

4 DEPURACION Y TEST

4.1 Plan de pruebas del sistema.

A continuación, vamos a realizar una planificación de las pruebas que deberíamos realizar para poder validar todas las funcionalidades realizadas por el programa de cada práctica.

4.1.1 Plan de pruebas del sistema completo desarrollado en la fase 2.

Durante la realización de esta fase se ha implementado todos los requisitos software y tenemos que comprobar que todos se realizan de manera correcta. Primero empezamos comprobando si se realiza bien la medida de la temperatura con los valores de tensión correspondiente. Para comprobar este funcionamiento introducimos un valor de tensión correspondiente a una cierta temperatura y comprobamos que al realizar la conversión corresponde con esa temperatura.

Después comprobamos si se visualizan correctamente los datos en la pantalla LCD, para ello al tener un valor constante de tensión correspondiente a una temperatura podemos observar que se representa en la pantalla la temperatura correcta con el formato correspondiente.

Para comprobar que se guardan los datos en la memoria Flash comprobamos, desde la ventana memory 1 del debugger, los valores que se encuentran en la dirección de memoria que hemos elegido (0x78000 (sector 29)) cada vez que pulsamos el botón central del joystick. Para que la prueba fuese satisfactoria debería coincidir con el formato que queremos de datos (DD/MM/AAAA hh:mm:ss ±TT.T V/f). También comprobamos que tras escribir 10 datos en la memoria el siguiente dato se guarda en la posición donde se guardó el primer dato, con esto comprobamos que se trata de un buffer circular.

Para comprobar si se reciben datos correctamente en el PC a través del puerto serie RS232 se procede a abrir una comunicación serial con el puerto COM3 en el programa Tera Term. AL ejecutar el programa se debería visualizar los datos correctos con el formato (DD/MM/AAAA hh:mm:ss ±TT.T V/f) y en el tiempo especificado de media

Por otro lado, también vamos a comprobar el funcionamiento del RGB, para ello se van a introducir distintos valores simulados de temperatura y se debería observar que para temperaturas mayores o iguales a 25 ºC el RGB es rojo y para temperaturas menores el RGB se encuentra azul.

En cuanto al funcionamiento del joystick ya se ha comentado las pruebas de funcionamiento de la pulsación central. Para comprobar el funcionamiento de la pulsación del joystick se va a analizar en el debugger a mirando en el watch1 la variable de la frecuencia de refresco (*time*) y se va a realizar la pulsación hacia izquierda o derecha. El resultado esperado sería que la frecuencia aumentase al pulsar derecha y disminuyese al realizar la pulsación izquierda. También se comprobaría si al llegar al límite tanto de izquierda como de derecha y se realiza una nueva pulsación, la frecuencia fuera la correcta.

4.1.2 Plan de pruebas del sistema completo desarrollado en la fase 3.

En esta fase se ha implementado la simulación de la señal analógica de entrada mediante el uso de el DAC integrado en el microprocesador. La salida del DAC irá conectado a la entrada del ADC y se puede incrementar o decrementar el nivel de tensión con las pulsaciones del joystick hacia arriba o hacia abajo. La tensión generada deberá visualizarse en la pantalla LCD.

Para comprobar el correcto funcionamiento de la generación de la tensión en el DAC se introduce un código correspondiente a una temperatura, y se mira con el wacth 1, dentro del debugger, si la variable de tensión a la salida del DAC corresponde con el de la tensión correspondiente a ese código. También se comprueba que al representar la temperatura en la pantalla LCD corresponde a la que deseábamos simular.

Para aumentar la temperatura mediante la pulsación del joystick se ha creado un array de códigos

correspondientes a un rango de tensiones para poder simularlas. Para comprobar si tenemos el funcionamiento correcto de la pulsación del joystick mostraremos por la pantalla del LCD la tensión simulada. Al realizar la pulsación UP del joystick se debería aumentar la tensión de la posición del array donde se encontraba a la tensión que se encontraba en la siguiente posición del array.

4.1.3 Plan de pruebas del sistema completo desarrollado en la fase 4.

En la fase 4 se pretende realizar la simulación de la señal digital periódica generada por el convertidor de V/F, para ello se han creado dos timers y un array con distintas frecuencias elegidas, al aumentar o disminuir de temperatura con las pulsaciones UP y DOWN del joystick se irá variando la frecuencia del Timer con una de las del array. Se ha unido un timer al otro para que mida el tiempo que pasa por cada vez que el otro timer hace match.

Para comprobar el correcto funcionamiento de l convertidor de voltaje a frecuencia se escoge una frecuencia determinada en el timer 1 y se comprueba en el debugger si el periodo que sale en el segundo timer corresponde a la del primero. Otra manera de comprobar si funciona correctamente es introducir una frecuencia correspondiente y comprobar si muestra en la pantalla LCD la misma frecuencia. Para comprobar si se realiza correctamente el aumento de la frecuencia a través del joystick se va a realizar una pulsación hacia arriba del joystick y se va a comprobar que en la pantalla del LCD se muestra la siguiente frecuencia del array de frecuencias creado. Para saber si realiza correctamente el cambio de frecuencia a temperatura, introducimos una frecuencia sabiendo que nos debería salir X temperatura y comprobamos si en la pantalla LCD se muestra la temperatura correspondiente.

4.1.4 Plan de pruebas del sistema completo desarrollado en la fase 5.

En esta fase se va a introducir la recepción de tramas a través del puerto serie RS232 desde el PC para establecer una nueva temperatura. Se devolverá una trama con la temperatura, tensión y frecuencia actualizada.

Para comprobar que las tramas se reciben correctamente se ha realizado el análisis en el debugger del array recibido y se ha comprobado que al enviar la trama se reciben correctamente y en el orden adecuado los caracteres enviados. Para comprobar que se actualiza la temperatura se deberá mostrar la temperatura enviada desde el PC y la tensión/frecuencia correspondiente en la pantalla LCD. Cuando se envía una trama desde el PC se deberá comprobar si se recibe desde el programa donde tengamos abierta la comunicación serie (Tera Term en este caso) la temperatura, tensión y frecuencia correspondientes en el formato correcto (“T=xx.x; V=y.yyyy; F=zzz.zzzz”).

4.2 Pruebas realizadas.

Ejecutadas todas las pruebas comentadas en el plan de pruebas, podemos afirmar que el sistema funciona correctamente. El único punto que tenemos que aclarar se encuentra en la fase 5. El envío de tramas desde el PC a través del puerto serie RS232 se realiza correctamente si se introduce la trama en el formato adecuado. El programa podría sufrir un mal funcionamiento si la trama no se envía con el formato: “Simula Temp= XX.X ºC” o “Simula Temp=-XX.X ºC”. Este comportamiento no esta contemplado ya que en el enunciado se especifica el envío de tramas con ese formato único. Para evitar que el buffer pueda tener caracteres indeseados se ha implementado el envío de un comando (r) para limpiar el buffer y poder enviar la trama correctamente.

5. Sistema analógico

En este apartado se incluye la memoria realizada en la fase 1, con una corrección en el apartado 5.3.4 en la que se indica en rojo.

5.1 Diseño del circuito.

Para este segundo bloque de la asignatura tenemos que realizar un medidor de temperatura a través de una RTD, y procesar la temperatura a través del microcontrolador LPC1768 para mostrarlo a través de la pantalla LCD de la mbed AppBoard, enviarse la temperatura al PC a través del puerto serie RS232, mostrarlo en la página Web y guardarlo en la memoria Flash del micro. Para la realización de lo descrito anteriormente hemos pensado en el siguiente diseño:

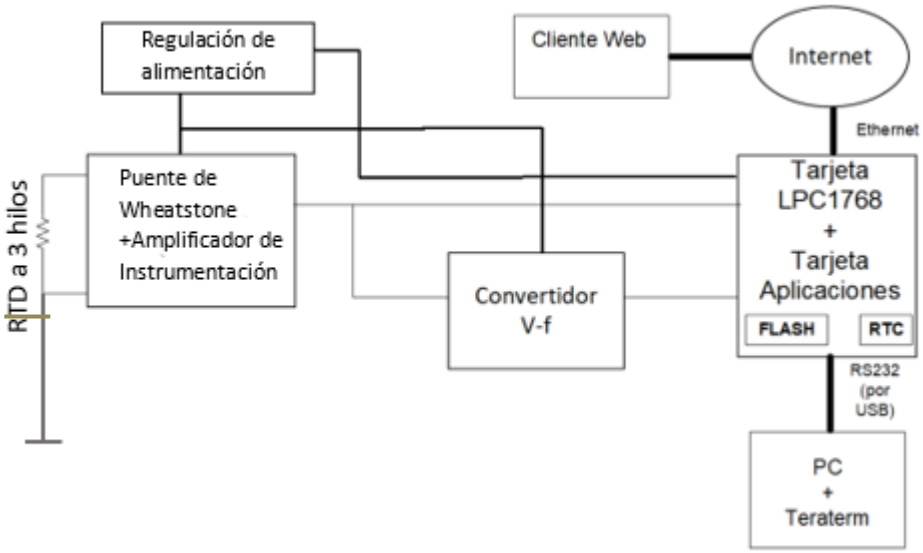


Ilustración 1 Diagrama de bloques del sistema completo

Para poder alimentar todo el circuito y poder tener una tensión regulada, constante y lo más próxima a los valores teóricos utilizados para los cálculos del circuito, hemos decidido utilizar la salida de 5V del micro no regulada para obtener a través del circuito TLV75533P_T la tensión de 3,3V necesaria. A partir de esta tensión hemos realizado dos divisores de tensión para obtener las tensiones de 2,5V, utilizada como tensión de referencia en el puente de Wheatstone, y la de 0,4V, utilizada para el offset.

Para la parte de la obtención de la temperatura hemos seleccionado el siguiente circuito:

Hemos decidido realizar la medida de la RTD a 3 hilos debido a que considerábamos que era la mejor opción. Por una parte, dentro de todas las opciones que barajábamos (medida a 2,3 y 4 hilos) esta opción

Medida de temperatura

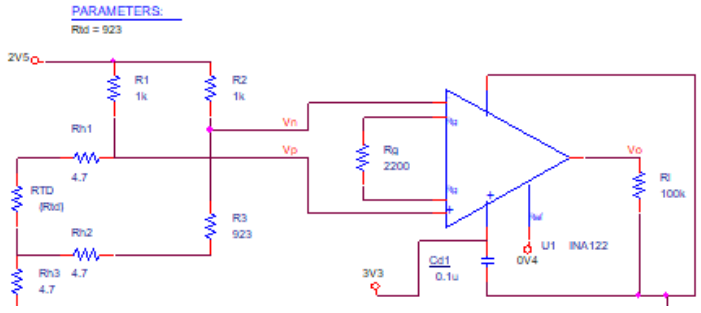


Ilustración 2 Circuito para la obtención de la medida de temperatura

no es la que más error evita, ya que lo ideal sería realizarlo con una medida a 4 hilos, pero tampoco introduce un error que vaya a suponer un peligro para la correcta de medida del circuito. Sin embargo, en la relación calidad/precio, siendo la calidad el menor error posible, la medida es la que sale ganando y lo que ha hecho que nos acabemos decantando por ella.

Para tratar la temperatura que obtenemos de la RTD tenemos la opción de tratar en voltios la medida y en el microprocesador obtener la temperatura medida. Pero para cuando la RTD se encuentre a distancias considerables que provoquen que nuestra medida a 3 hilos de la resistencia RTD genere un error considerable, hemos tenido que realizar la conversión de los voltios a frecuencia a la salida de nuestro acondicionador de señal.

Para afrontar esta conversión hemos seguido las mismas pautas por las que nos hemos guiado con el resto del circuito, funcionalidad junto a economía, es por ello por lo que nos hemos decantado por la elección del circuito AD7740YRT (Grado Y). Este circuito se adecua perfectamente a nuestro circuito y se explicará detalladamente su funcionamiento en el apartado 2.3.

5.2 Cálculos teóricos.

Utilizando la PT1000 dentro del rango de temperaturas de -20ºC a 60ºC obtenemos los siguientes valores de la RTD:

$$RTD = R_0 * (1 + \alpha T)$$

Siendo $R_0 = 1000\Omega$ y α=0,00385 $\Omega/\Omega/$ºC tenemos los siguientes valores de RTD:

T (ºC)	RTD (Ω)
-20	923
-15	942,25
-10	961,5
-5	980,75
0	1000
5	1019,25
10	1038,5
15	1057,75
20	1077
25	1096,25
30	1115,5
35	1134,75
40	1154
45	1173,25
50	1192,5
55	1211,75
60	1231

Ilustración 3 Cálculos teóricos RTD

Para poder tener 0V para la temperatura de -20ºC hemos elegido unas resistencias de 1k, 1k y 920 en el puente de Wheatstone dispuestas tal como se muestra en el circuito del apartado 1.
Realizamos los cálculos con la resistencia RTD medida a 2 hilos y obtenemos los siguientes valores:

T (ºC)	RTD	Vd-Vn2	Vo(2H)	Error Cables (mV)	Error cables ºC
-20	923	0,006323995	0,489110845	0,089110845	2,380048623
-15	942,25	0,019084616	0,668919582	0,089440241	2,388846416
-10	961,5	0,031595967	0,845215895	0,089780004	2,397921103
-5	980,75	0,043865282	1,018101708	0,090129349	2,407251691
0	1000	0,055899519	1,187675037	0,090487537	2,416818486
5	1019,25	0,067705368	1,35403018	0,090853877	2,426603009
10	1038,5	0,07928927	1,517257889	0,09122772	2,436587915
15	1057,75	0,090657425	1,677445534	0,091608455	2,446756919
20	1077	0,101815806	1,834677263	0,09199551	2,457094732
25	1096,25	0,112770165	1,989034145	0,092388349	2,467586993
30	1115,5	0,123526048	2,140594312	0,092786465	2,478220214
35	1134,75	0,1340888	2,289433089	0,093189384	2,488981723
40	1154	0,144463576	2,435623123	0,09359666	2,499859612
45	1173,25	0,154655351	2,579234494	0,094007875	2,510842693
50	1192,5	0,164668924	2,720334836	0,094422635	2,521920451
55	1211,75	0,174508928	2,858989434	0,09484057	2,533083003
60	1231	0,184179836	2,995261331	0,095261331	2,544321059

Ilustración 4 Valores teóricos a dos hilos

Por lo que descartamos inmediatamente el empleo de esta medida debido a la gran cantidad de error introducida por los cables. Es por ello que decidimos realizar las medidas a 3 hilos.
Para calcular el valor de Vp-Vn con la resistencia medida a 3 hilos tenemos:

$$V_p = V_A + (V_{REF} - V_A)\left(\frac{RTD + R_H}{R_1 + RTD + R_H}\right)$$

Donde Va es el valor de la tensión entre la RTD y la masa, Rh es la resistencia de los hilos que hemos supuesto de 4,7Ω y R1 es la resistencia de valor 920Ω.

$$V_n = V_A + (V_{REF} - V_A)\left(\frac{R + R_H}{R + R_h + R_1}\right)$$

$$V_A = \frac{V_{REF}}{(R_1 + R_h + RTD)//(R_1 + R + R_h)}$$

Por lo que:

$$V_d = V_p - V_n = (V_{REF} - V_A)\left(\frac{RTD + R_H}{R_1 + RTD + R_H} - \frac{R + R_H}{R + R_h + R_1}\right)$$

Y la tensión $V_O(3hilos)$ será:

$$\boldsymbol{V_O(3hilos) = A_d V_d = A_d(V_{REF} - V_A)\left(\frac{RTD + R_H}{R_1 + RTD + R_H} - \frac{R + R_H}{R + R_h + R_1}\right)}$$

De las fórmulas anteriores obtenemos los siguientes valores:

T (ºC)	RTD	Vd-Vn	Vo(ideal)	Vd-Vn3	Vo(3H)	Error Cables(V)	Error ºC
-20	923,00	0,0000	0,4000	0,0000	0,4000	0,0000	0,0000
-15	942,25	0,0129	0,5795	0,0128	0,5798	0,0003	0,0105
-10	961,50	0,0255	0,7554	0,0253	0,7561	0,0007	0,0214
-5	980,75	0,0379	0,9280	0,0375	0,9290	0,0010	0,0325
0	1000,00	0,0501	1,0972	0,0496	1,0986	0,0014	0,0439
5	1019,25	0,0620	1,2632	0,0614	1,2649	0,0017	0,0556
10	1038,50	0,0737	1,4260	0,0730	1,4281	0,0021	0,0676
15	1057,75	0,0851	1,5858	0,0843	1,5883	0,0025	0,0797
20	1077,00	0,0964	1,7427	0,0955	1,7456	0,0029	0,0921
25	1096,25	0,1074	1,8966	0,1064	1,8999	0,0033	0,1046
30	1115,50	0,1183	2,0478	0,1172	2,0515	0,0037	0,1173
35	1134,75	0,1290	2,1962	0,1278	2,2003	0,0041	0,1302
40	1154,00	0,1394	2,3420	0,1381	2,3465	0,0045	0,1432
45	1173,25	0,1497	2,4852	0,1483	2,4901	0,0049	0,1563
50	1192,50	0,1598	2,6259	0,1583	2,6312	0,0053	0,1696
55	1211,75	0,1697	2,7641	0,1682	2,7699	0,0057	0,1829
60	1231,00	0,1795	2,9000	0,1779	2,9062	0,0062	0,1963

Ilustración 5 Valores teóricos 3 hilos

5.3 Simulaciones y explicación de los módulos empleados.

5.3.1 Alimentación y tensiones de referencia.

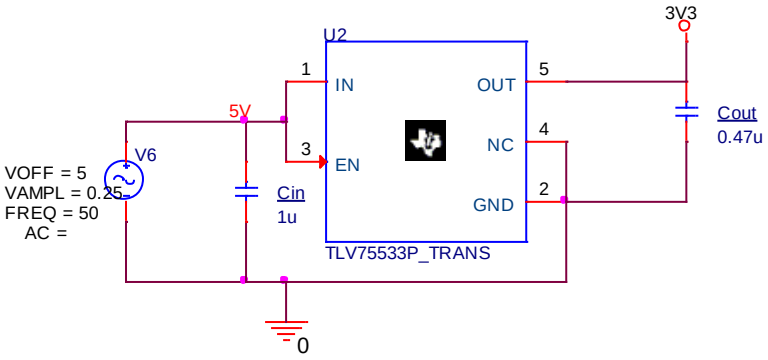


Ilustración 6 Circuito regulador de tensión

Mediante el TLV75533P_T podemos obtener una tensión regulada de 3.3V a partir de una entada de 5V no regulada.

En cuanto al uso de los condensadores Cin y Cout, el fabricante recomienda en la página 15 del datasheet, colocar un condensador de entrada Cin de 1uF y un condensador cerámico de salida Cout de 0.47uF.

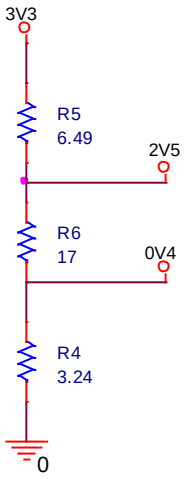


Ilustración 7 Obtención de tensiones de referencia (Circuito)

Para obtener las tensiones necesarias de 0.4V (para el offset) y 2.5 (para el puente), utilizamos el siguiente divisor de tensión.

El valor de las resistencias viene de las siguientes relaciones.

$2.5V = \frac{3.3V*(R_6+R_4)}{R_6+R_4+R_5}$ y $0.4V = \frac{2.5V*R_4}{R_6+R_4}$

Empleando resistencias con tolerancias de 0.1% para conseguir una mayor precisión en estos valores de tensión.

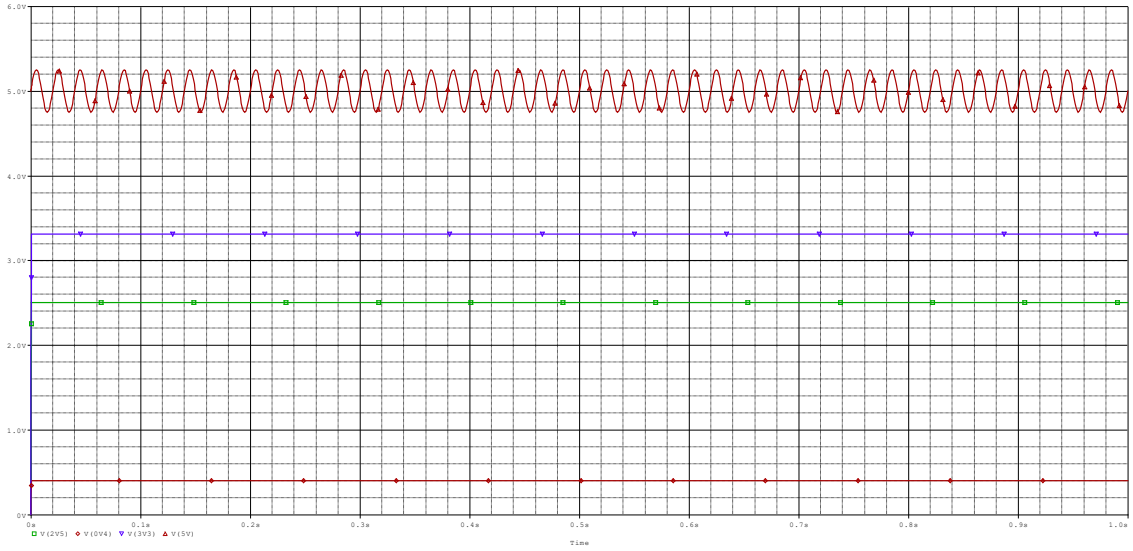


Ilustración 8 Resultados simulación, alimentación y tensiones de referencia

Tabla resultados numéricos de tensiones de referencia y alimentación.

Señal:	Valor (V)
V(2V5)	2.5063
V(0V4)	401.202m
V(3V3)	3.3099
V(5V)	5.2235

Ilustración 9 Tabla resultados numéricos de tensiones de referencia y alimentación.

5.3.2 Puente de Wheatstone.

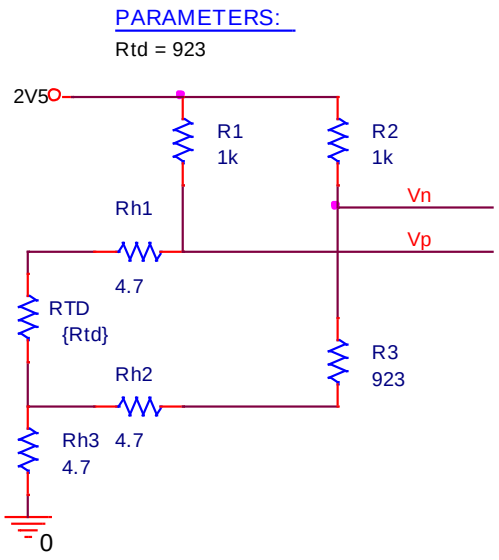


Ilustración 10 Circuito puente de WheatStone

Por simplicidad se han utilizado resistencias de 1kΩ para R1 y R2 y se ha utilizado una R3 de 923Ω para conseguir que a 0ºC la diferencia entre Vp y Vn sea 0V.

Realizando una simulación paramétrica en función del valor óhmico de la RTD como se muestra a continuación, obtenemos las siguientes medidas.

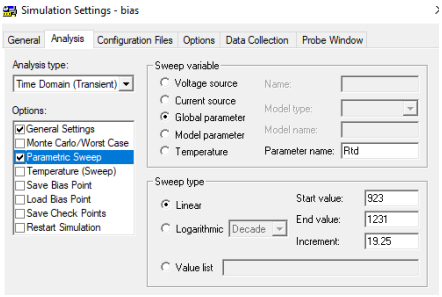


Ilustración 11 Parámetros de simulación

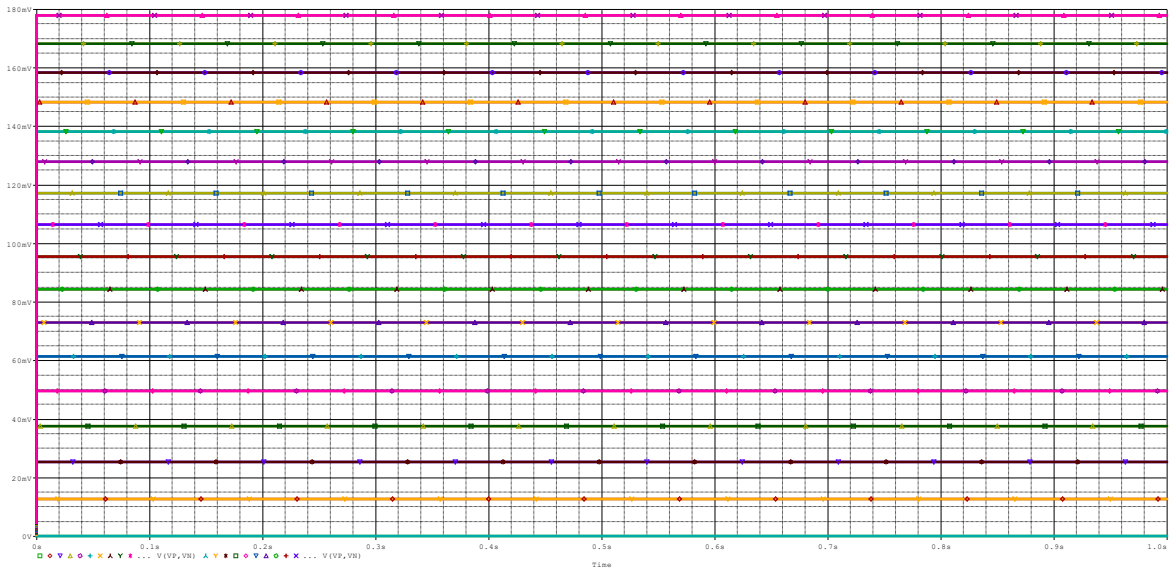


Ilustración 12 Resultados de simulación, puente de WheatStone

Valor de Temperatura (ºC)	Valor de RTD (Ω)	Diferencia de tensión (V) SIMULADO	Diferencia de tensión (mV) IDEAL	ERROR (mV)	ERROR (ºC)
-20	923	-2.8571n	0	-0,0028	1,35901E-06
-15	942.25	12.761m	0,0129	-0,139	6,61169E-05
-10	961.5	25.272m	0,0255	-0,228	0,000108451
-5	980.75	37.542m	0,0379	-0,358	0,000170287
0	1000	49.576m	0,0501	-0,524	0,000249246
5	1019.25	61.382m	0,0620	-0,618	0,000293959
10	1038.5	72.966m	0,0737	-0,734	0,000349135
15	1057.75	84.334m	0,0851	-0,766	0,000364356
20	1077	95.493m	0,0964	-0,907	0,000431425
25	1096.25	106.447m	0,1074	-0,953	0,000453305
30	1115.5	117.203m	0,1183	-1,097	0,0005218
35	1134.75	127.766m	0,1290	-1,234	0,000586966
40	1154	138.141m	0,1394	-1,259	0,000598857
45	1173.25	148.333m	0,1497	-1,367	0,000650229
50	1192.5	158.346m	0,1598	-1,454	0,000691611
55	1211.75	168.187m	0,1697	-1,513	0,000719675
60	1231	177.858m	0,1795	-1,642	0,000781035

Ilustración 13 Resultados numéricos diferencia de tensión en el puente (simulado vs ideal)

La gráfica anterior muestra la diferencia de tensión entre los puntos Vp y Vn. Por otro lado, a continuación, se muestra una tabla que compara los valores teóricos con los obtenidos mediante la simulación.

5.3.3 Amplificador de instrumentación.

A continuación, se muestra el amplificador de instrumentación INA122, se ha escogido este componente frente a otros dado que permite la alimentación no simétrica y además trabaja bien a baja tensiones.

Para la lección de la Rg se ha utilizado la siguiente fórmula, que encontramos en el datasheet el componente (Cuyo enlace se encuentra al final del documento.): $G = 5 + 200k\Omega/\mathrm{R_G}$

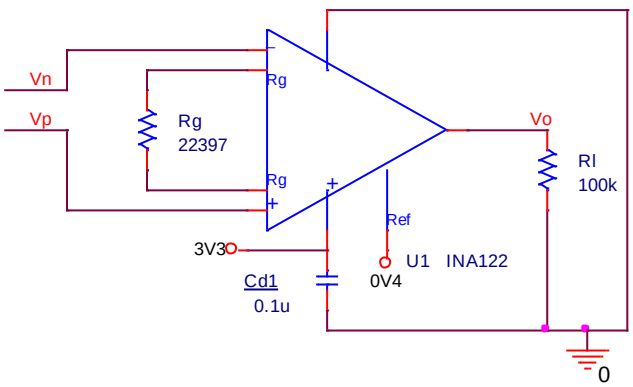


Ilustración 14 Circuito amplificador de instrumentación

Para el cálculo de la ganancia se ha utilizado como requerimiento en que los valores máximo y mínimo de salida no entren en saturación. Para ello tras simular el componente encontramos que satura para valores inferiores a 0.4V y superiores a 3V. Por ello, se ha añadido una señal de 0.4V al terminal Ref para solventar la saturación para el mínimo valor de tensión, esta señal actúa de la siguiente manera: $V_o = G(V_p - V_n) + V_{ref}$.

Ahora toca calcular el valor de la ganancia de la siguiente manera: $\frac{V_o - V_{ref}}{V_{pmax} - V_n} = G$ para esto encontramos un valor de G ideal de 13,92926299

V/V que supone una Rg de 22398.26Ω. Dado que este valor no es comercial, lo aproximamos al valor posible comercial más cercano, además se han escogido resistencias d 0.1% de tolerancia para reducir lo máximo posible el error de ganancia.

Los valores de Rg escogidos son la suma de 2 resistencias: $R_{G1} = 22k\Omega y\ R_{G2} = 397\Omega$.

Rg=22397 Ω

Hechos estos ajustes, se simula el circuito y se obtiene el siguiente resultado, donde se visualiza el valor de Vo para los distintos valores de la Rtd.

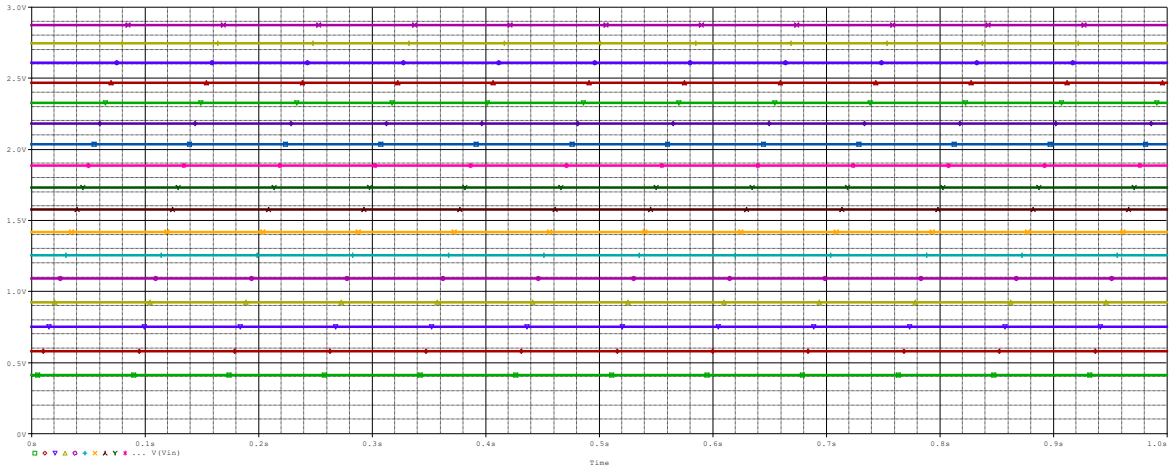


Ilustración 15 Resultados de simulación a la salida del amplificador (ERROR DE GANANCIA)

La siguiente tabla muestra los valores obtenidos, y con los compara con los valores ideales.

Valor de Temperatura (ºC)	Valor de RTD (Ω)	Tensión a la salida (V) SIMULADO	Tensión a la salida (V) IDEAL	ERROR (V)	ERROR (ºC)
-20	923	409.307m	0,4000	0,00931	0,00000
-15	942.25	577.681m	0,5816	-0,00388	0,00000
-10	961.5	751.946m	0,7596	-0,00761	0,00000
-5	980.75	922.854m	0,9341	-0,01125	0,00000
0	1000	1.0905	1,1053	-1,09437	-0,23
5	1019.25	1.2549	1,2732	-1,26064	-0,26
10	1038.5	1.4163	1,4379	-1,42377	-0,30
15	1057.75	1.5747	1,5996	-1,58385	-0,33
20	1077	1.7301	1,7583	-1,74096	-0,37
25	1096.25	1.8827	1,9140	-1,89519	-0,40
30	1115.5	2.0325	2,0669	-2,04661	-0,43
35	1134.75	2.1796	2,2171	-2,19529	-0,46
40	1154	2.3242	2,3646	-2,34132	-0,49
45	1173.25	2.4661	2,5094	-2,48476	-0,52
50	1192.5	2.6056	2,6517	-2,62569	-0,55
55	1211.75	2.7427	2,7916	-2,76416	-0,58
60	1231	2.8727	2,9290	-2,90028	-0,61

Ilustración 16 Resultados de simulación, salida del amplificador (ERROR DE GANANCIA)

Tras observar que con ese valor de ganancia se introduce mucho error, de modo que se ha variado y probado con distintos valores de ganancia hasta llegar al que menor error introduce. Esto se ha conseguido

poniendo una Rg de 22kΩ. Obtenemos los siguientes resultados mediante la simulación (se incluye el puente de Wheatstone):
Nota: la nueva G es 14,09090909

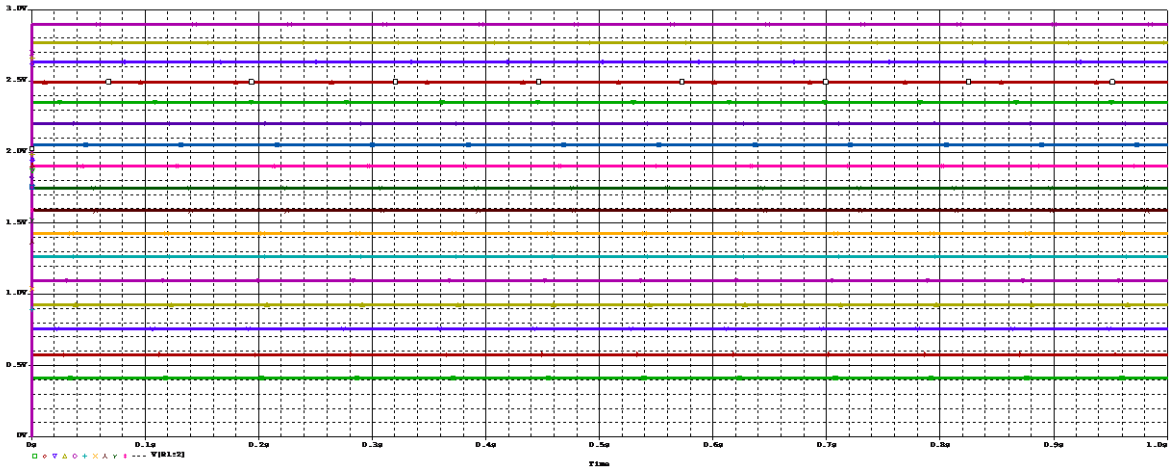


Ilustración 17 Resultados de simulación a la salida del amplificador (Ganancia corregida)

Valor de Temperatura (ºC)	Valor de RTD (Ω)	Tensión a la salida (V) SIMULADO	Tensión a la salida (V) IDEAL	ERROR (V)	ERROR (ºC)
-20	923	0,4088	0,4000	0,0088	0,2818
-15	942.25	0,5785	0,5816	-0,0010	-0,0315
-10	961.5	0,7543	0,7596	-0,0011	-0,0349
-5	980.75	0,9268	0,9341	-0,0012	-0,0370
0	1000	1,0960	1,1053	-0,0012	-0,0380
5	1019.25	1,2620	1,2732	-0,0012	-0,0376
10	1038.5	1,4248	1,4379	-0,0012	-0,0394
15	1057.75	1,5847	1,5996	-0,0011	-0,0364
20	1077	1,7416	1,7583	-0,0011	-0,0346
25	1096.25	1,8956	1,9140	-0,0010	-0,0335
30	1115.5	2,0468	2,0669	-0,0010	-0,0323
35	1134.75	2,1954	2,2171	-0,0008	-0,0270
40	1154	2,3413	2,3646	-0,0007	-0,0233
45	1173.25	2,4846	2,5094	-0,0006	-0,0201
50	1192.5	2,6254	2,6517	-0,0005	-0,0164
55	1211.75	2,7638	2,7916	-0,0003	-0,0112
60	1231	2,8928	2,9290	-0,0072	-0,2304

Ilustración 18 Resultados de simulación, salida del amplificador (ganancia corregida)

El amplificador quedaría de la siguiente manera una vez se ha ajustado la ganancia:

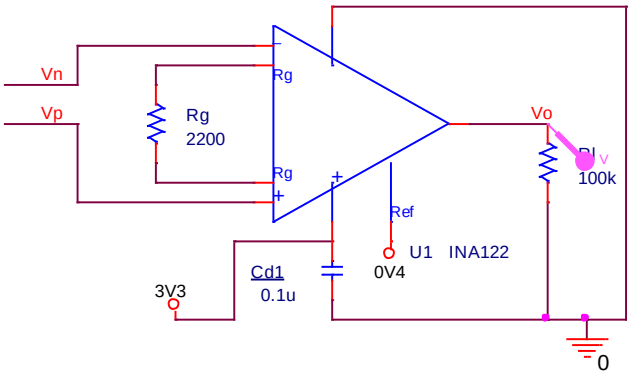


Ilustración 19 Circuito amplificador de instrumentación con ganancia corregida

5.3.4 Convertidor Voltaje frecuencia.

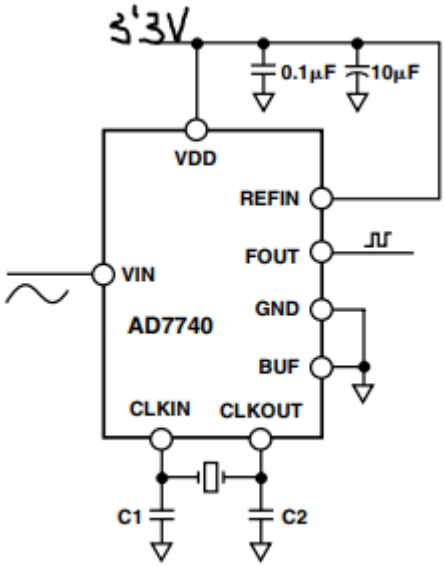


Ilustración 20 Circuito convertidor tensión frecuencia (imagen del datasheet)

Se ha empleado el componente AD7740YRT (Grado Y) que soporta temperaturas de entre -40ºC y 105ºC y, además, que permite una salida amplia en frecuencia con una alimentación entre 3V y 5V, perfecto para nuestro sistema.

Montando el siguiente circuito podemos obtener en la salida fout una señal cuya frecuencia se corresponde con la siguiente fórmula:

$$F_{OUT} = 0.1 * f_{CLKIN} + 0.8\left(\frac{V_{IN}}{V_{REF}}\right) * f_{CLKIN}$$

Dejando el pin REFIN desconectado, hacemos que V_{REF} valga 2.5V, por otro lado, empleando un oscilador de 1MHz y condensadores C1 y C2 de 30pF

De este modo, tendremos a la salida

$$F_{OUT-MIN} = 0.1 * 1MHz * 0.8\left(\frac{0.4}{2.5}\right) * 1MHz = 196.97KHz$$

$$F_{OUT-MAX} = 0.1 * 1MHz * 0.8\left(\frac{3}{2.5}\right) * 1MHz = 827.27KHz$$

Corrección:

Debido a una errata que se detectó durante la realización de la fase 3, nos gustaría comentar lo siguiente: Como se observa en la figura 20, el terminal REFIN esta conectado a los 3.3V, y no se deja al aire, como se indica en el apartado, haciendo que valga 2.5V, esto lleva a que las fórmulas para la obtención de fmax y fmin queden de la siguiente manera.

$$F_{OUT-MIN} = 0.1 * 1MHz * 0.8\left(\frac{0.4}{3.3}\right) * 1MHz = 196.96KHz$$

$$F_{OUT-MAX} = 0.1 * 1MHz * 0.8\left(\frac{2.9}{3.3}\right) * 1MHz = 803.03KHz$$

5.3.5 Simulación del sistema completo.

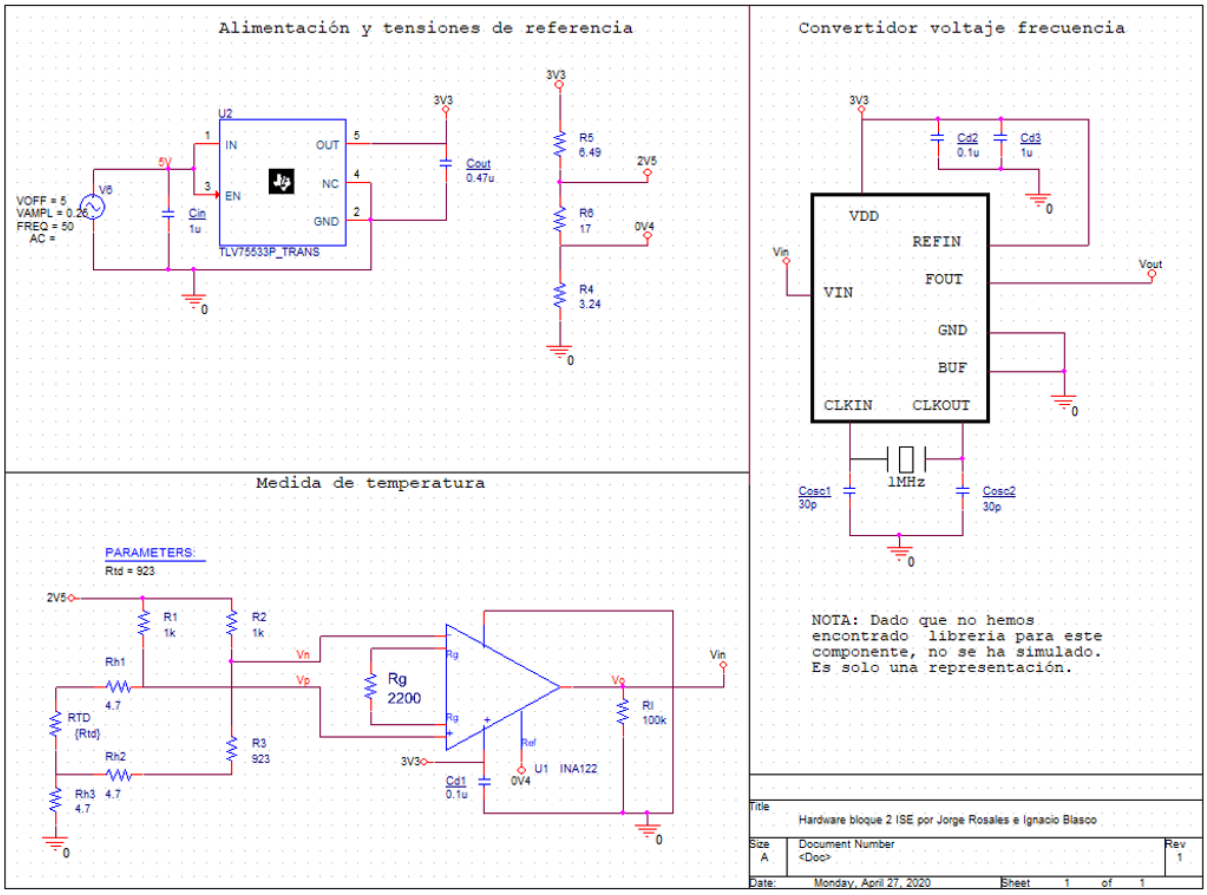


Ilustración 21 Esquema completo del sistema propuesto

La imagen anterior muestra el sistema completo.

Ejecutando la simulación encontramos los siguientes resultados.

- Tensiones de referencia:

Ilustración 22 Resultados simulación, alimentación y tensiones de referencia

Señal:	Valor (V)
V(2V5)	2.4936
V(0V4)	399.208m
V(5V)	5.2449
V(3V3)	3.3099

Ilustración 23 resultados numéricos de tensiones de referencia y alimentación.

- Diferencia de tensión entre Vp y Vn:

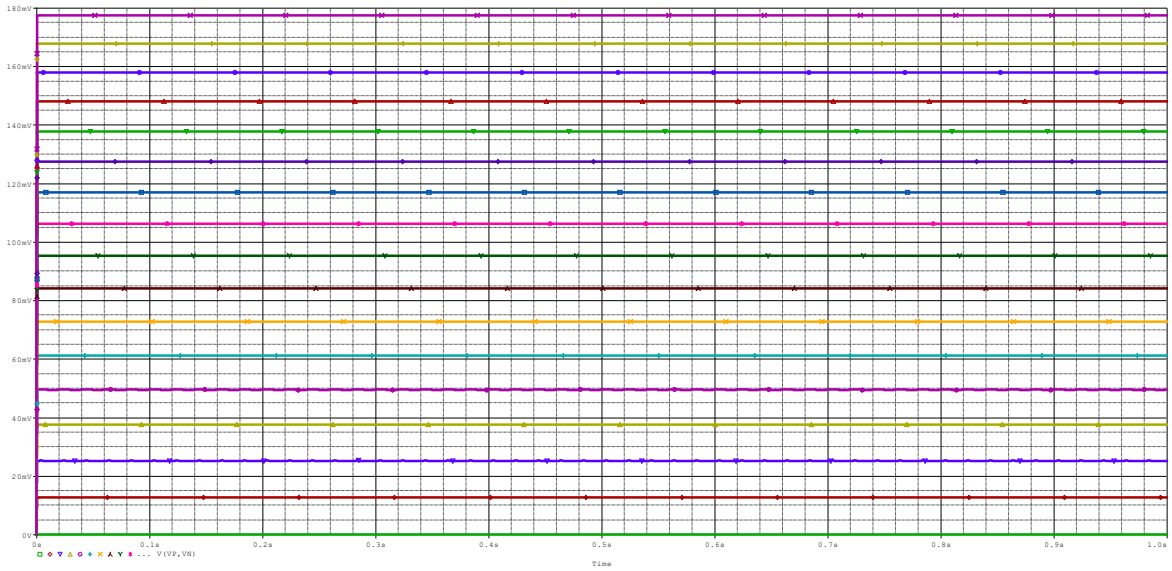


Ilustración 24 Simulación completa del puente de Wheatstone

Valor de Temperatura (ºC)	Valor de RTD (Ω)	Diferencia de tensión (V) SIMULADO	Diferencia de tensión (V) IDEAL	ERROR (mV)	ERROR (ºC)
-20	923	-2.8513n	0	-2.8513n	0,001285434
-15	942.25	12.727m	12.885m	-0,158	7,12302E-05
-10	961.5	25.207m	25.517m	-0,31	0,000139755
-5	980.75	37.445m	37.904m	-0,459	0,000206928
0	1000	49.450m	50.052m	-0,602	0,000271396
5	1019.25	61.227m	61.969m	-0,742	0,000334511
10	1038.5	72.784m	73.660m	-0,876	0,000394922
15	1057.75	84.126m	85.133m	-1,007	0,00045398
20	1077	95.258m	96.393m	-1,135	0,000511685
25	1096.25	106.188m	107.446m	-1,258	0,000567137
30	1115.5	116.921m	118.298m	-1,377	0,000620785
35	1134.75	127.461m	128.955m	-1,494	0,000673531
40	1154	137.814m	139.421m	-1,607	0,000724474
45	1173.25	147.984m	149.701m	-1,717	0,000774065
50	1192.5	157.977m	159.801m	-1,824	0,000822303
55	1211.75	167.798m	169.725m	-1,927	0,000868738
60	1231	177.450m	179.478m	-2,028	0,000914271

Ilustración 25 Resultados numéricos de la salida del puente (Simulación vs Ideal)

- Tensión a la salida del amplificador:

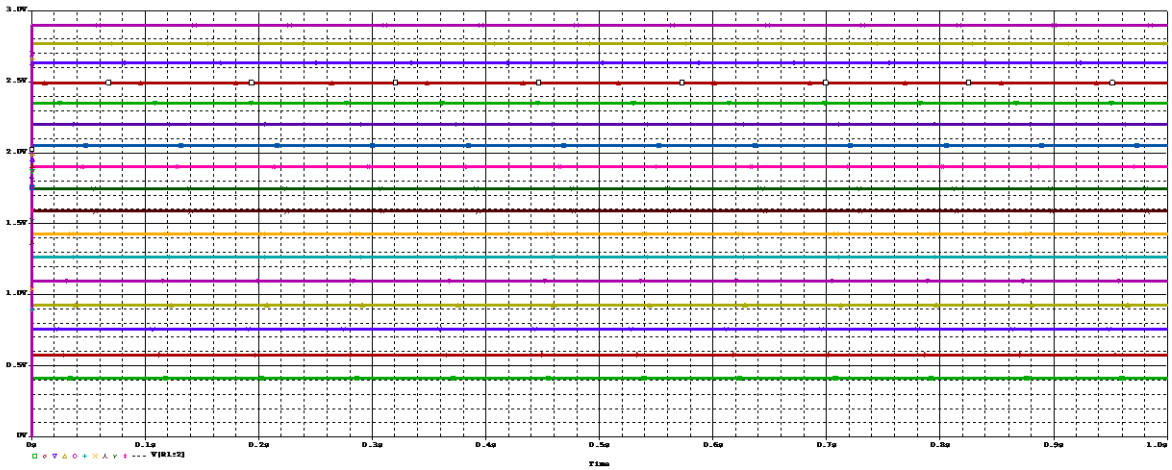


Ilustración 26 Simulación del sistema completo salida del amplificador de instrumentación

Valor de Temperatura (ºC)	Valor de RTD (Ω)	Tensión a la salida (V) SIMULADO	Tensión a la salida (V) IDEAL	ERROR (V)	ERROR (ºC)
-20	923	0,4088	0,4000	0,0088	0,2818
-15	942.25	0,5785	0,5816	-0,0010	-0,0315
-10	961.5	0,7543	0,7596	-0,0011	-0,0349
-5	980.75	0,9268	0,9341	-0,0012	-0,0370
0	1000	1,0960	1,1053	-0,0012	-0,0380
5	1019.25	1,2620	1,2732	-0,0012	-0,0376
10	1038.5	1,4248	1,4379	-0,0012	-0,0394
15	1057.75	1,5847	1,5996	-0,0011	-0,0364
20	1077	1,7416	1,7583	-0,0011	-0,0346
25	1096.25	1,8956	1,9140	-0,0010	-0,0335
30	1115.5	2,0468	2,0669	-0,0010	-0,0323
35	1134.75	2,1954	2,2171	-0,0008	-0,0270
40	1154	2,3413	2,3646	-0,0007	-0,0233
45	1173.25	2,4846	2,5094	-0,0006	-0,0201
50	1192.5	2,6254	2,6517	-0,0005	-0,0164
55	1211.75	2,7638	2,7916	-0,0003	-0,0112
60	1231	2,8928	2,9290	-0,0072	-0,2304

Ilustración 27 Resultados numéricos salida del amplificador de instrumentación (Simulación vs Ideal)

5.4 Posibles problemas y soluciones para el calibrado.

Dada la tolerancia de los componentes y de los errores de ganancia y no linealidad introducidos por los componentes utilizados, y dado que no se permite la calibración manual mediante potenciómetros, proponemos para hacer un ajuste correcto en caso de que la medida no sea precisa, introducir al programa el valor de las señales de referencia de 2.5V y 0.4V (Mediante 2 entradas ADC) para que mediante el uso de cinco resistencias de valores conocidos, el sistema pueda calibrarse mediante ajustes internos (operaciones de conversión), pudiendo solventar en cierta medida los errores introducidos por el hardware.

Además, al realizar la conversión se tomaría en cuanta el valor real de las señales de 2.5V y 0.4V para que los errores debido a posibles diferencias en estas señales de referencia no afecten en absoluto (por ejemplo, si en vez de valer 2.5 valiese 2.3). Como anotación se han utilizado resistencias de baja tolerancia para reducir al máximo este problema.
En cuanto a los errores de no linealidad, tanto en el INA122 como en el AD7740, los errores máximos son del ±0.012%, y el error de ganancia aproximado en el IN122, es como máximo 0.1%. Para solventar el posible error de ganancia que se pueda encontrar, se puede sustituir la resistencia Rg por una resistencia de 20kΩ y un potenciómetro multivuelta de 5kΩ, de modo que, se pueda ajustar la ganancia para el menor error posible. Esto se consigue midiendo las 5 resistencias de valor fijo mencionadas anteriormente, y con los valores obtenidos en el microprocesador más los valores de tensión de las señales de 0.4V y 2.5V, se puede calcular la ganancia real. Se ajusta el potenciómetro para que los valores de tensión correspondientes a las 5 resistencias se asemejen lo máximo posible a sus valores ideales. Por otro lado, también se podría sustituir la resistencia R5 (superior del divisor de tensión) para ajustar la señal de 2.5V de manera precisa y tener una buena señal de referencia.
Como nota, una vez ajustados los 2.5V, la señal de 0.4V también se ajustaría dada la relación de resistencias que presenta y la baja tolerancia de estas.

5.5 Componentes empleados

Lista con los componentes empleados y precio.

- 1 Resistencia PT1000
- 2 resistencias de 1kΩ, 1 Resistencia de 100kΩ, 1Resistencia de 20kΩ, 1 resistencia de 397Ω, 1 resistencia de 6,49Ω, 1 resistencia de 17Ω y una resistencia de 3,24Ω.
- 1 potenciómetro multivuelta de 5k Ω
- 2 condensadores de 0,1 uF, 1 condenador de 0,47 uF, 3 condensadores de 1 uF y 2 condensadores de 30 pF.
- 1 oscilador de cuarzo de 1MHz
- 1 amplificador de instrumentación INA 122.
- 1 conversor de voltaje a frecuencia AD7740YRT (Grado Y)
- 1 regulador de tensión TLV75533P_T Tabla de precios:

Componente	Precio	Enlace
PT1000 3 hilos	13.46€	Mouser-> REF: 485-3984
1kΩ	0,324 €	Mouser-> 594-MCT0603MD1001BP5
100kΩ	0,594 €	Mouser-> 756-HPWC2512-100KJT1
2000Ω	0,117 €	Mouser-> 284-APC0603B20K0N
Pot 5k Ω	3,42 €	Mouser-> 72-TS63Y-5K
379Ω	0,392 €	Mouser-> 660-RN732ATTD3790B50
64.9Ω	0,252	Mouser-> 71-CRCW020164R9FKED

17Ω (10+7)	0,234 €	Mouser->756-PCF1206R-10RBT1 Mouser->756-PCF1206R-10RBT1
3.24Ω (2+1+0.24)	0,101 € 0,111 € 0,486 €	Mouser-> 660-RK73H1ETTP2R00F Mouser-> 660-RK73B1ERTTP1R0J Mouser-> 710-561070332092
0.1uF	0,03 €	Mouser-> 710-885012105001
0.47uF	0,13 €	Mouser-> 710-860160672001
1uF	0,324 €	Mouser-> 80-EDH105M050S9BAA
30pF	0,09	Mouser-> 603-AC0402JRNP9BN300
Oscilador de cuarzo 1MHz	0.50€	https://www.planetaelectronico.com/filtro-resonador-1mhz-2pin-p-20079.html
INA 122U	6,98 €	Mouser-> 595-INA122U
AD7740YRT (Grado Y)	2,61 €	Mouser-> 584-AD7740YRTZ-R7
TLV75533P	0,477 €	Mouser-> 595-TLV75533PDBVR

Ilustración 28 Componentes, precio y enlace a compra

5.6 Enlaces a datasheets de interés.

Amplificador de instrumentación: https://www.ti.com/lit/ds/symlink/ina122.pdf?ts=1587981836974
Regulador LDO: http://www.ti.com/lit/ds/symlink/tlv755p.pdf?ts=1587982738364
Convertidor V/F: https://www.mouser.com/datasheet/2/609/adi_ad7740-1215214.pdf

6 OTROS ASPECTOS

En este punto nos gustaría comentar la posibilidad de ajustar el sistema mediante la UART para corregir los distintos errores de ganancia, offset, y no linealidad introducidos por el AI. Bastaría con enviar un comando indicando la ganancia y offset que se tuviera realmente. Y definir en el código como variables la ganancia y el offset de manera que se pudiera mediante el comando enviando cambiar estos valores para ajustar la precisión del sistema.