



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD ZACATENCO

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

SECCIÓN DE BIOELECTRÓNICA

“Desarrollo de un sistema de monitoreo del estado de irrigación
sanguínea en extremidades inferiores en humano a través de la
medición de temperatura”

T E S I S

Que presenta

Karina Teresa Castán Aquino

Para obtener el Grado de

Maestra en Ciencias

en la Especialidad de

Ingeniería Eléctrica

Director de la Tesis:

Dr. Pablo Rogelio Hernández Rodríguez



DEDICATORIAS

A mí esposo Olaf

Por todo su apoyo, asesorías y paciencia que me ha brindado a lo largo de este proyecto y en cada momento de mi vida. Esta también es tu tesis mi amor.

A mis padres

Por enseñarme a echarle ganas y seguir adelante en cada proyecto que me proponga.

A mis hermanas

Porque sin su locura sería aburrida la vida.

Karina Castán

Autora



AGRADECIMIENTOS

A mi director de tesis, Dr. Pablo Rogelio Hernández Rodríguez, por sus lecciones y apoyo a lo largo de mi maestría.

Al Ingeniero Eladio Cardiel, por sus tutorías y correcciones brindadas en la realización de mi proyecto de tesis.

A mis sinodales, el Dr. Arturo Vera Hernández por permitirme trabajar en su laboratorio y hacer uso del instrumental como un miembro más de su alumnado. A la Dra. Laura Ivoone Garay Jiménez por sus consejos y asesorías en la realización de mi proyecto de tesis. Al M. en C. Ricardo Arturo Horta por sus aportaciones y sugerencias en la realización de mi tesis.

A mis compañeras y compañeros de laboratorio por hacer más amena mi estadía.

Al pueblo de México que a través del Consejo Nacional de Ciencia y Tecnología (CONACyT) me apoyaron con una beca para la realización de mis estudios de maestría en el CINVESTAV-IPN.



CONTENIDO

LISTA DE FIGURAS	III
LISTA DE TABLAS	V
RESUMEN	VI
ABSTRACT	VII
1. CAPÍTULO 1. INTRODUCCIÓN	1
1.1. PLANTEAMIENTO DEL PROBLEMA	1
1.2. OBJETIVOS	3
1.2.1. <i>Objetivo General</i>	3
1.2.2. <i>Objetivos Particulares</i>	3
2. CAPÍTULO 2. ANTECEDENTES Y ESTADO DEL ARTE	4
2.1. CIRCULACIÓN SANGUÍNEA	4
2.2. CIRCULACIÓN EN LA PIEL	5
2.3. TEMPERATURA CENTRAL Y TEMPERATURA PERIFÉRICA (CUTÁNEA).....	7
2.3.1. <i>Regulación de la Temperatura Corporal</i>	8
2.4. MEDICIÓN DEL FLUJO SANGUÍNEO.....	11
2.5. CORRELACIÓN DE LA TEMPERATURA DE LA PIEL Y LAS OSCILACIONES DEL FLUJO SANGUÍNEO.....	12
2.6. CONTRIBUCIÓN DEL FLUJO SANGUÍNEO A LA RESPUESTA DE LA TEMPERATURA DE LA PIEL DURANTE UNA PRUEBA DE SENSIBILIDAD AL FRÍO	13
2.7. INFLUENCIA DE LAS TEMPERATURAS INTERNAS Y DE LA PIEL SOBRE LA VASODILATACIÓN CUTÁNEA ACTIVA BAJO DIFERENTES NIVELES DE EJERCICIO Y TEMPERATURAS AMBIENTE EN LOS SERES HUMANOS.....	13
2.8. SISTEMA PARA EL ANÁLISIS DE TRANSFERENCIA DE CALOR Y LA MEDICIÓN DE FLUJO SANGUÍNEO PERIFÉRICO.....	14
3. CAPÍTULO 3. DESARROLLO	15
3.1. MÓDULO DE MEDICIÓN PARA LA PC	15
3.1.1. <i>Sensor de Temperatura</i>	15
3.1.2. <i>Circuito de Acondicionamiento</i>	17
3.1.3. <i>Curva de Caracterización de los Sensores</i>	22
3.1.4. <i>Fuente de Alimentación</i>	24
3.1.5. <i>Módulo de Adquisición de Datos</i>	25



3.1.6. Interfaz Gráfica.....	26
3.2. MÓDULO PORTÁTIL	27
3.2.1. Almacenamiento de Datos.....	28
3.2.2. Despliegue de la Temperatura.....	30
3.2.3. Microcontrolador.....	31
4. CAPÍTULO 4. PRUEBAS	34
4.1. INSTRUMENTACIÓN.....	34
4.2. PROTOCOLO DE MEDICIÓN A	35
4.2.1. Sujetos A.....	35
4.2.2. Condiciones del Laboratorio A.....	36
4.2.3. Puntos de Medición A.....	36
4.2.4. Descripción del Protocolo A.....	37
4.2.5. Procesamiento de los Datos A.....	38
4.3. PROTOCOLO DE MEDICIÓN B	39
4.3.1. Sujetos B.....	39
4.3.2. Condiciones del Laboratorio B.....	39
4.3.3. Puntos de Medición B.....	40
4.3.4. Descripción del Protocolo B.....	40
4.3.5. Procesamiento de los Datos B.....	42
5. CAPÍTULO 5. RESULTADOS Y DISCUSIÓN	43
5.1. PROTOCOLO A. CAMBIO DE POSICIÓN DE LA EXTREMIDAD INFERIOR DERECHA.....	43
5.2. PRUEBA B. EJERCICIO FÍSICO EN LAS EXTREMIDADES INFERIORES.	46
6. CAPÍTULO 6.CONCLUSIONES Y PERSPECTIVAS	50
REFERENCIAS.....	51
APÉNDICE A. RESULTADOS DE LA CARACTERIZACIÓN DE LOS SENSORES	53
APÉNDICE B. LIBRERÍAS PARA LA MEMORIA SD	54
APÉNDICE C. LIBRERÍA PARA EL MANEJO DE LA GLCD.....	64
APÉNDICE D. DIAGRAMAS ESQUEMÁTICOS DEL MÓDULO PORTÁTIL	82
APÉNDICE E. CÓDIGO DE PRINCIPAL DEL MÓDULO PORTÁTIL.....	85
APÉNDICE F. ARTÍCULO EN CONGRESO	100



LISTA DE FIGURAS

Figura 2.1. Circulación sanguínea por la piel [3].	6
Figura 2.2. Temperaturas de diversas zonas del cuerpo de un sujeto desnudo, expuesto a temperaturas ambiente, generadas en un calorímetro [1].	8
Figura 2.3. Mecanismos de pérdida de calor del cuerpo [3].	9
Figura 2.4. Efecto de la variación de la temperatura ambiental sobre la conductancia de calor desde el centro del organismo hasta la piel. (Modificado de Benziger TH: Heat and Temperature Fundamentals of Medical Physiology, New York: Dowden, Hutchin & Ross, 1980) [3].	10
Figura 2.5. Medición por fotopleletismografía y termografía en la punta del dedo [13, 16].	13
Figura 2.6. Estructura del sensor de volumen de flujo sanguíneo [6].	14
Figura 3.1. Diagrama a bloques del sistema de adquisición de temperatura para la PC.	15
Figura 3.2. Banda con termistor utilizada para el registro de la temperatura. Izquierda: para medición en tobillo, pantorrilla o muslo. Derecha: para la medición en el hallux.	17
Figura 3.3. Curva característica del termistor 10K3A1B sin linealizar.	17
Figura 3.4. Gráfica donde se muestra los tres puntos para el rango de temperatura a medir	19
Figura 3.5. Etapa de linealización del termistor.	19
Figura 3.6. Etapa de linealización, filtrado y eliminación de offset del termistor.	20
Figura 3.7. Circuito final para el acondicionamiento del sensor de temperatura.	22
Figura 3.8. Sistema para la obtención de la curva de caracterización de los termistores, en donde se observa la tina con agua y el termorregulador, así como el termómetro de mercurio, el sensor y la manguera del enfriador.	23
Figura 3.9. Curvas características finales para cada sensor de temperatura.	24
Figura 3.10. Circuito de la fuente de voltaje para el módulo de la PC.	25
Figura 3.11. Módulo de adquisición de datos NI USB 6008®.	25



Figura 3.12. Panel de control del programa desarrollado en Labview®.	26
Figura 3.13. Diagrama a bloques del programa de control.	27
Figura 3.14. Diagrama a bloques del sistema portátil para la adquisición de la temperatura de la piel.....	28
Figura 3.15. Módulo de lectura para la memoria SD.	29
Figura 3.16. Diagrama esquemático del módulo y el cambiador de nivel para la lectura de la memoria SD.	30
Figura 3.17. Diagrama eléctrico a bloques del módulo portátil.....	32
Figura 3.18. Diagrama de flujo del programa principal del módulo portátil.....	33
Figura 4.1. Puntos de medición en la extremidad inferior derecha para el protocolo A.	36
Figura 4.2. Posición de la arteria tibial posterior en la extremidad izquierda.....	37
Figura 4.3. Esquema de las fases del protocolo experimental A.....	38
Figura 4.4. Posición de la pierna derecha durante la segunda fase de la prueba A..	38
Figura 4.5. Puntos de medición en la extremidad inferior derecha para el protocolo B.	40
Figura 4.6. Esquema de las fases del protocolo experimental B.....	41
Figura 4.7. Posición del sujeto de medición en la tercera fase del protocolo B.....	42
Figura 5.1. Promedios de temperatura para el hallux, tobillo y pantorrilla comparados con el promedio de las velocidades de flujo de todos los sujetos.	43
Figura 5.2. Promedios de temperatura para el hallux, dorso, tobillo y pantorrilla comparados con el promedio de las velocidades de flujo de todos los sujetos.....	47
Figura. B.1.Circuito esquemático de la etapa de acondicionamiento de la señal.....	82
Figura. B.2.Circuito esquemático de la fuente de alimentación del módulo.	83
Figura. B.3.Circuito esquemático principal del módulo portátil.	84



LISTA DE TABLAS

Tabla 2.1. Características de varios tipos de vasos sanguíneos en humanos [1].	5
Tabla 3.1. Valores típicos de resistencia del termistor dados por el fabricante [18]. .	18
Tabla 3.2. Pines de conexión de la memoria SD mediante comunicación SPI.	29
Tabla 3.3. Pines de conexión de la GLCD.	31
Tabla 5.1. Temperatura promedio [°C] en los puntos de medición correspondientes a las posiciones de la pierna en cada sujeto. Las casillas sombreadas indican disminución de temperatura.	45
Tabla 5.2. Velocidad de flujo sanguíneo promedio [cm/s] en los puntos de medición correspondientes a las posiciones de la pierna en cada sujeto. Las casillas sombreadas indican disminución de temperatura.	45



RESUMEN

En la presente tesis se desarrollaron dos sistemas de medición de temperatura de la piel, con el objetivo de determinar si este parámetro puede servir como un indicador que permita monitorear el estado de la irrigación sanguínea en las extremidades inferiores, particularmente en personas con problemas de movilidad y en su caso sometidos a un proceso de rehabilitación.

El primer sistema de medición, para trabajo en laboratorio, cuenta con 4 termistores NTC miniatura y una etapa de acondicionamiento de la respuesta de los sensores, que son adquiridas mediante un módulo de adquisición de datos NI USB 6008® y mostrados en una PC mediante una interfaz gráfica desarrollada en LabView®.

El segundo sistema, para trabajo en campo, cuenta con 4 termistores, una etapa de adquisición de las señales de los sensores de temperatura por medio de un microcontrolador (PIC18F4550), un display gráfico y una tarjeta de memoria SD para almacenar datos y su análisis posterior.

Los sistemas fueron aplicados mediante dos protocolos experimentales. Los resultados del primero, basado en la realización de cambios posturales, mostraron que el 57% de los sujetos presentan una relación directamente proporcional entre la velocidad de flujo sanguíneo y la temperatura de la piel en las extremidades inferiores, mientras que el segundo, en donde se ejecutó una actividad física, no se observó una relación directa entre esos parámetros.



ABSTRACT

In this thesis two skin temperature measurement systems were developed, with the aim of determining whether skin temperature can be used as an indicator to monitor the state of blood irrigation in lower extremities, particularly in people with mobility problems and submitted to a process of rehabilitation.

The first measurement system, for laboratory work, has 4 miniature NTC thermistors and a stage of signal conditioning, which are acquired by a data acquisition card NI USB 6008[®] and displayed on a PC using a graphical interface developed in LabView[®].

The second is a portable system, has 4 thermistors and a temperature acquisition stage. The temperature sensors signals are acquired by a microcontroller (PIC18F4550), shown in a graphical display and stored in an SD memory for later analysis.

The systems were applied using two experimental protocols. The results of the first one, based on performing postural changes, showed that 57% of subjects had a directly proportional relationship between the velocity of blood flow and skin temperature in the lower extremities, while the second, where physical activity was executed, no direct relationship between these parameters was observed.





CAPÍTULO 1. INTRODUCCIÓN

1.1. Planteamiento del Problema

Los problemas de movilidad en una persona a causa de una enfermedad o un accidente pueden ser permanentes o de carácter transitorio. En estas condiciones, la formación de úlceras por presión, riesgos vasculares, hinchazón por retención de líquidos, deterioro en articulaciones, formación de trombos en extremidades inferiores, etc. representan un grave riesgo para su salud.

En consecuencia, resulta de fundamental importancia monitorear que el flujo sanguíneo, en las zonas o puntos más lejanos tal como las extremidades inferiores sea adecuado, ya que la sangre se encarga de llevar los nutrientes, así como el oxígeno a los diferentes tejidos y órganos del cuerpo; a su vez, también se encarga de recoger los desechos y el dióxido de carbono de estos para ser eliminados [1].

El flujo sanguíneo, “volumen de sangre que fluye a través de cualquier tejido en un determinado intervalo de tiempo (en ml/min)” [2], tiene entre otras funciones el transporte y distribución del calor que participa en los mecanismos de control de la temperatura corporal desde el interior del organismo hacia la piel [3].

Existen tres características que describen el flujo sanguíneo de la piel [4]:

1. movimiento físico,
2. transporte de calor,
3. contenido de oxígeno.

Conforme la sangre fluye a través de la piel, ésta transporta calor [4, 5]; “si el flujo cutáneo aumenta, el calor se conduce con mayor eficiencia desde el interior del cuerpo hasta la piel, mientras que cuando el flujo sanguíneo hacia la piel se reduce, la conducción de calor disminuye” [3]. En otras palabras, la cantidad de calor perdido



por el cuerpo se regula en gran medida mediante las variaciones en la cantidad de la sangre que fluye a través de la piel [1]. Por lo tanto, la medición de la temperatura puede ser usada como un indicador local o regional del flujo sanguíneo de la piel [4].

Diversos autores han estudiado el comportamiento y relación entre el flujo sanguíneo y la temperatura de la piel. Uno de esos trabajos es el realizado por los autores A.A. Sagaidachnyi et al., los cuales estudiaron la relación entre las oscilaciones en temperatura y flujo sanguíneo en la punta de los dedos de la mano a través de la ecuación de biotransferencia de calor de Pennes permitiendo una correlación entre estos parámetros de 0.35 a 0.63 en el promedio de los datos [5]. Un trabajo similar fue el realizado por los autores K. Nagata et al., proponiendo que en base a la temperatura de los dedos de las manos es posible calcular el flujo sanguíneo periférico. Su sistema de medición basado en dos tipos de sensores, uno de contacto y otro infrarrojo para la medición de la temperatura de la piel, les permitió calcular el volumen de flujo sanguíneo comparado con un equipo Doppler [6].

En base a los conceptos mencionados, se propone desarrollar un sistema que, mediante la medición de la temperatura superficial de las extremidades inferiores del cuerpo humano, permita hacer una estimación del estado de irrigación de la piel, para detectar y evaluar anomalías en personas con problemas de movilidad, específicamente en usuarios de sillas de ruedas [4].

Pudiendo servir en un futuro como un sistema de monitoreo que permita retroalimentar el funcionamiento de un dispositivo de rehabilitación.



1.2. Objetivos

1.2.1. Objetivo General

Diseñar, desarrollar y evaluar un sistema de medida de la temperatura superficial de las extremidades inferiores del cuerpo humano para determinar de forma indirecta y aproximada el estado de irrigación sanguínea regional.

1.2.2. Objetivos Particulares

- Investigación bibliográfica de la relación entre temperatura y flujo sanguíneo de la piel.
- Diseñar un módulo de laboratorio para la adquisición de la temperatura de la piel así como su interfaz gráfica, basada en un módulo de adquisición.
- Diseñar un módulo portátil para trabajo en campo basado en dispositivos programables.
- Diseñar un sistema de soporte de los sensores y fijación a las extremidades inferiores.
- Plantear dos protocolos experimentales.
- Realizar pruebas en diferentes sujetos.



CAPÍTULO 2. ANTECEDENTES Y ESTADO DEL ARTE

2.1. Circulación Sanguínea

El aparato circulatorio consta de tres componentes interrelacionados: la sangre, el corazón y los vasos sanguíneos [2].

La sangre, tejido conectivo líquido, posee tres funciones generales.

1. Transporte: lleva oxígeno desde los pulmones hacia las células del cuerpo y dióxido de carbono desde las células hacia los pulmones; así como nutrientes, hormonas, calor y productos de desecho para que sean eliminados del cuerpo.
2. Regulación: ayuda a mantener la homeostasis de todos los líquidos corporales, y a regular el pH. Contribuye en el ajuste de la temperatura corporal a través de las propiedades refrigerantes y de absorción de calor del agua presente en el plasma sanguíneo y su flujo variable a través de la piel, donde el excedente de calor puede perderse y ser transferido al medio ambiente.
3. Protección: sus glóbulos blancos nos protegen de las enfermedades, así como los anticuerpos y factores del sistema del complemento. La sangre puede coagularse, lo cual previene su pérdida excesiva del aparato circulatorio tras una lesión.

La sangre debe ser constantemente bombeada a través de los vasos sanguíneos (un sistema cerrado de conductos) de manera tal que pueda alcanzar las células del organismo, siendo el corazón el encargado de esta función [2].



El volumen de sangre que fluye a través de cualquier tejido en un determinado intervalo de tiempo (en ml/min), es conocido como flujo sanguíneo [2]. Éste se debe principalmente a la fuerza proporcionada por el bombeo cardiaco [1].

La velocidad del flujo sanguíneo (en cm/s) se relaciona en forma inversa con el área de sección transversal de los vasos [2]. Existen cinco tipos principales de vasos sanguíneos: las arterias, las arteriolas, los capilares, las vénulas y las venas. La tabla 2.1 muestra las medidas generales de estos vasos sanguíneos en humanos [2].

Vaso	Diámetro de la luz	Grosor de la pared	Todos los vasos de cada tipo	
			Área transversal total aproximada (cm ²)	Porcentaje de volumen sanguíneo contenido ¹
Aorta	2.5 cm	2 mm	4.5	2
Arteria	0.4 cm	1 mm	20	8
Arteriola	30 µm	20 µm	400	1
Capilar	5 µm	1 µm	4500	5
Vénula	20 µm	2 µm	4000	54
Vena	0.5 cm	0.5 mm	40	
Vena Cava	3 cm	1.5 mm	18	

Tabla 2.1. Características de varios tipos de vasos sanguíneos en humanos [1].

¹En vasos sistémicos. Existe 12% adicional en el corazón y 18% en la circulación pulmonar.

2.2. Circulación en la Piel

La sangre que circula en los vasos cutáneos no sólo aporta O₂ y nutrientes, sino que forma parte del sistema termorregulador [7, 8]. La cantidad de calor perdido por el cuerpo se regula en gran medida mediante las variaciones en la cantidad de la sangre que fluye a través de la piel [1].



La piel está vascularizada de una manera generosa en su capa inferior. Especial mención merece el plexo venoso continuo que recibe sangre de los capilares cutáneos (figura 2.1). En la mayoría de las áreas expuestas del cuerpo (manos, pies y orejas) las pequeñas arterias también aportan sangre al plexo a través de anastomosis arteriovenosas provistas de mucho tejido muscular [3].

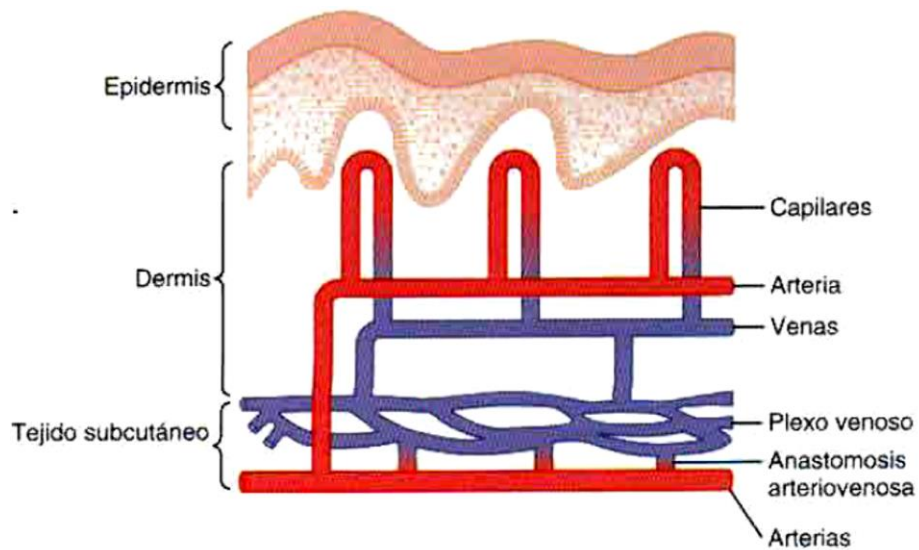


Figura 2.1. Circulación sanguínea por la piel [3].

La velocidad (o volumen por minuto) con la que fluye la sangre hacia la piel por el plexo venoso varía mucho casi un 0% hasta un 30% del gasto cardiaco total [3]. “Si el flujo cutáneo aumenta, el calor se conduce con enorme eficiencia desde el centro del cuerpo hasta la piel, mientras que cuando la tasa de perfusión de la piel se reduce, la conducción de calor es mínima” [3].

En resumen, la piel es un sistema “radiador de calor” con un control eficaz; el flujo sanguíneo de la piel se comporta como el mecanismo más eficiente que transfiere el calor del centro del organismo hacia la piel [3].

Otro mecanismo de regulación de la irrigación cutánea está representado por la hiperemia reactiva [7]; ésta ocasiona el cambio de suministro de sangre a los



diferentes tejidos a través de la vasodilatación al momento de restablecer la circulación en una región después de un periodo de oclusión. Con la oclusión del suministro sanguíneo a una extremidad, las arteriolas cutáneas se dilatan por debajo de la oclusión. Además, al restablecer la circulación la sangre que fluye en los vasos dilatados ocasiona la apariencia enrojecida de la piel. El O_2 se difunde a una distancia corta a través de la piel y la hiperemia reactiva se evita si la oclusión de la circulación en la extremidad se realiza en una atmósfera con 100% de O_2 . Por tanto, la dilatación arteriolar aparentemente se debe al efecto local de la hipoxia [1].

2.3. Temperatura Central y Temperatura Periférica (cutánea)

La temperatura de los tejidos profundos del organismo, o temperatura “central”, permanece constante, con un margen de ± 0.6 °C, excepto cuando se presenta una enfermedad febril [3].

No existe una temperatura central que pueda considerarse normal porque las mediciones efectuadas en muchas personas sanas revelan un intervalo de temperaturas bucales desde 36°C hasta más de 37.5°C [3].

Zonas diversas del cuerpo muestran temperaturas diferentes; la magnitud de la diferencia entre ellas cambia con la temperatura ambiental (fig. 2.2). En general, las extremidades son más frías que el resto del organismo [1].

La temperatura de la piel, a diferencia de la central, aumenta y descende con la temperatura del entorno [3].

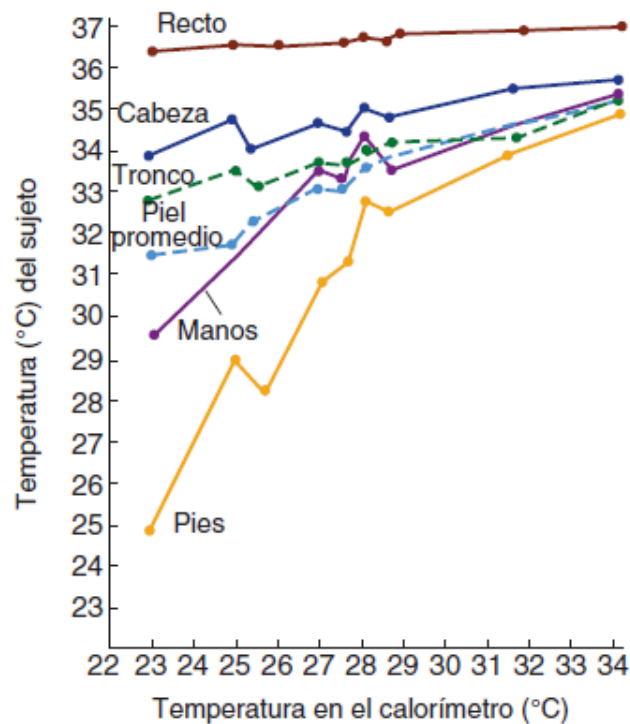


Figura 2.2. Temperaturas de diversas zonas del cuerpo de un sujeto desnudo, expuesto a temperaturas ambiente, generadas en un calorímetro [1].

2.3.1. Regulación de la Temperatura Corporal

La temperatura corporal se regula por el equilibrio entre la producción y la pérdida de calor. Si la tasa de producción calorífica del organismo excede la de la pérdida, se acumula el calor dentro del cuerpo y aumenta la temperatura corporal. En cambio, si la pérdida es mayor, descienden el calor y la temperatura corporal [3].

El calor en el cuerpo es producido por actividad muscular, asimilación del alimento y todos los procesos vitales que contribuyen con la tasa metabólica basal [1].

La figura 2.3 ilustra los distintos métodos por los que el calor se desprende de la piel al entorno [3, 9]:



1. conducción térmica es el intercambio de calor entre las moléculas de dos materiales que están en contacto directo entre sí;
2. convección es la transferencia de calor por el movimiento de un fluido (gas o un líquido) entre áreas de diferente temperatura;
3. radiación es la transferencia de calor en forma de rayos infrarrojos entre un objeto cálido y uno frío sin mediar contacto físico;
4. evaporación debido a la conversión del sudor en vapor; cuando el agua se evapora de la superficie cutánea, por cada gramo evaporado desaparecen 0.58 calorías (kilocalorías) de calor [3].

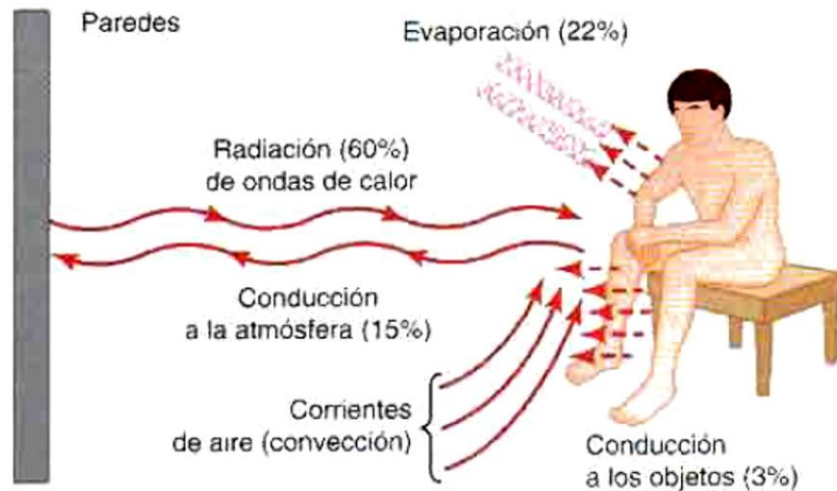


Figura 2.3. Mecanismos de pérdida de calor del cuerpo [3].

Puesto que la conducción térmica se presenta de la superficie de un objeto a la superficie de otro, la temperatura de la piel determina, en gran escala, el grado en que el calor corporal se pierde o se gana. La cantidad de calor que llega a la piel proveniente de tejidos profundos puede variar por cambios del flujo sanguíneo hacia la piel. Cuando se dilatan los vasos cutáneos, la sangre caliente llega a la piel, mientras que en el estado de máxima contracción vascular el calor se mantiene en el centro del cuerpo. La frecuencia a la que el calor es transferido desde los tejidos profundos hacia la piel se llama conductancia tisular [1].



Por otro lado, cuando se enfría la piel de todo el organismo, se desencadenan efectos reflejos que aumentan la temperatura corporal por diversas vías [3, 10]:

- 1) se estimula con intensidad el escalofrío, con lo que aumenta la tasa de producción de calor del organismo;
- 2) se inhibe la sudoración, si es que se había manifestado y
- 3) se induce una vasoconstricción de la piel para reducir la pérdida del calor a través de la misma.

“La figura 2.4 muestra el efecto cuantitativo de la temperatura ambiental sobre la conducción de calor desde el centro del cuerpo hasta la superficie de la piel y luego, la conducción atmosférica: la conductancia del calor se multiplica casi por ocho entre el estado de vasoconstricción plena y el de vasodilatación total” [3].

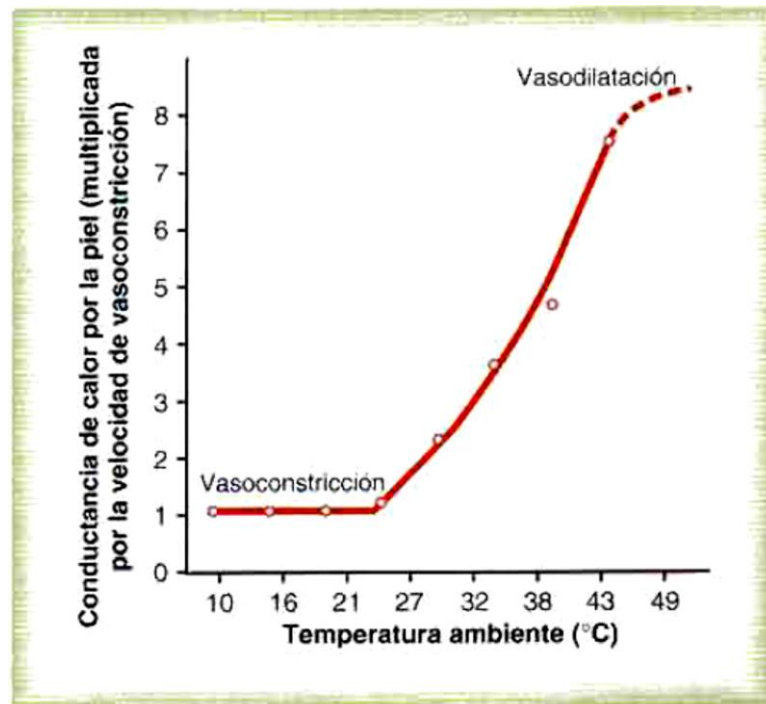


Figura 2.4. Efecto de la variación de la temperatura ambiental sobre la conductancia de calor desde el centro del organismo hasta la piel. (Modificado de Benziger TH: Heat and Temperature Fundamentals of Medical Physiology, New York: Dowden, Hutchin & Ross, 1980) [3].



2.4. Medición del Flujo Sanguíneo

El flujo sanguíneo suele ser registrado tanto de forma invasiva, como no-invasiva [11, 12].

Los métodos invasivos para la medición del flujo sanguíneo gozan de poca aceptación debido a su misma naturaleza y a las complicaciones relacionadas a ésta. Entre los métodos invasivos empleados están [11, 12]:

- método indicador, en donde un indicador es introducido dentro del flujo sanguíneo y las resultantes concentraciones arteriales y venosas del indicador son medidas de acuerdo al llamado “principio Fick”;
- método electromagnético que está enfocado en un vaso sanguíneo expuesto el cual es colocado dentro de un campo magnético transversal. Una diferencia de potencial es inducida en el conductor y medida de forma perpendicular para ambas direcciones, de flujo y campo magnético.

Los métodos no-invasivos más utilizados para la determinación del flujo sanguíneo, comprenden los siguientes tres enfoques [1, 11, 12]:

- método ecocardiográfico, basado en ultrasonido Doppler;
- método de cardiografía de impedancia, introduce una corriente eléctrica y mide el voltaje resultante en dirección axial del tórax;
- método de contorno de presión de pulso, deriva la forma de onda de flujo aórtico de la forma de onda de la presión arterial.

Un método que permite la medición indirecta del flujo regional es la termografía la cual es una técnica que permite calcular la temperatura superficial de los objetos a distancia, siendo necesario para validar el estado de irrigación de la piel alguno de los métodos antes descritos.



Las mediciones del flujo sanguíneo de la piel son usadas para estudiar condiciones clínicas en donde éste se ve afectado por una patología subyacente. Las condiciones principales incluyen: enfermedad vascular periférica, desórdenes vasomotores, selección del nivel de amputación y la evaluación de la eficacia de un amplio rango de tratamientos [4].

Es sabido que la cantidad de calor que se pierde del cuerpo es regulada en gran medida por la variación en el volumen sanguíneo que fluye por la piel [1]. Como se expuso anteriormente, existen diversos autores que han estudiado el comportamiento y relación entre el flujo sanguíneo y la temperatura de la piel en diferentes zonas de cuerpo y con diversos estímulos [6, 13, 14], concluyendo que sí existe una relación entre estos dos parámetros. Estos trabajos se exponen en los apartados siguientes.

2.5. Correlación de la Temperatura de la Piel y las Oscilaciones del Flujo Sanguíneo

La relación entre la temperatura de la piel y las oscilaciones del flujo sanguíneo en la punta de los dedos de la mano en sujetos sanos, ha sido investigada por el autor A.A. Sagaidachnyi et al. [5, 13, 15, 16]. En estos trabajos por medio de la medición de las oscilaciones del flujo sanguíneo por fotoplethysmografía (PPG, *photoplethysmography*) y la medición de las oscilaciones de la temperatura por una cámara de imagen térmica (figura 2.5), buscan encontrar la correlación entre estos dos parámetros, diferenciándose los estudios de acuerdo al tipo de análisis realizado en sus datos obtenidos. En uno de ellos, determinan la relación por medio de los componentes espectrales de los parámetros medidos [13]; en el segundo la encuentran por medio de la simplificación del modelo matemático de Shitzer [15]; en un tercero correlacionan el comportamiento de los parámetros en el dominio de tiempo y frecuencia [16]. En otro estudio buscan obtener la posibilidad de



restauración de las oscilaciones de flujo sanguíneo a través de los datos de temperatura [5].

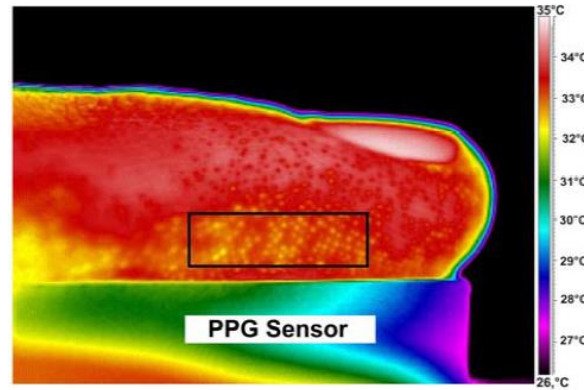


Figura 2.5. Medición por fotopleletismografía y termografía en la punta del dedo [13, 16].

2.6. Contribución del Flujo Sanguíneo a la Respuesta de la Temperatura de la Piel Durante una Prueba de Sensibilidad al Frío

En este estudio tratan de demostrar si en una prueba de sensibilidad al frío, la temperatura de la piel obtenida refleja el retorno del flujo sanguíneo a la extremidad tras una entrada de frío local. Encontrando que el flujo sanguíneo de la piel contribuye a su recalentamiento [14].

2.7. Influencia de las Temperaturas Internas y de la Piel Sobre la Vasodilatación Cutánea Activa Bajo Diferentes Niveles de Ejercicio y Temperaturas Ambiente en los Seres Humanos

Para aclarar la influencia de la temperatura interna y la temperatura de la piel sobre la dilatación cutánea activa durante el ejercicio, este trabajo comparó los umbrales de temperatura del cuerpo en el comienzo de la vasodilatación durante ejercicio ligero y moderado bajo diferentes condiciones de temperatura ambiente [17]. Sus resultados



sugieren que la vasodilatación se ve más afectada por las temperaturas internas que por la temperatura de la piel, así como en el ejercicio con carga moderada que con carga ligera.

2.8. Sistema para el Análisis de Transferencia de Calor y la Medición de Flujo Sanguíneo Periférico

En este trabajo de investigación emplean un sensor (fig. 2.6) para medir la temperatura de la punta de los dedos debido a que el calor emitido por la superficie de la piel depende del volumen y la temperatura del flujo sanguíneo en los capilares. Los resultados de la prueba en los dedos humanos sugieren que este método es útil para estimar el volumen de flujo sanguíneo periférico [6].

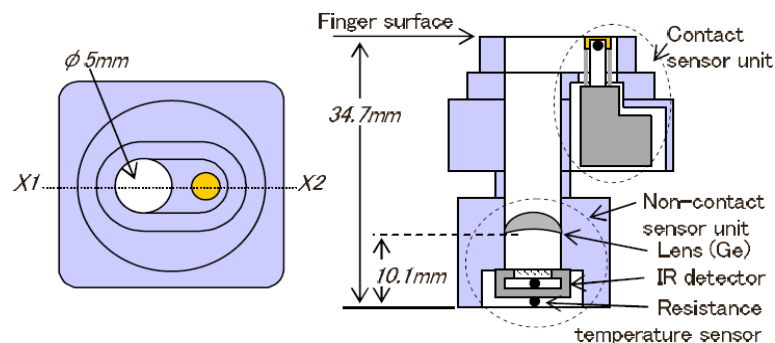


Figura 2.6. Estructura del sensor de volumen de flujo sanguíneo [6].



CAPÍTULO 3. DESARROLLO

Para la obtención de la medida de temperatura de la piel, se proponen dos sistemas de medición, ambos basados en el uso de sensores de contacto, con la diferencia de que uno es empleando una PC para medir en laboratorio y el otro utiliza dispositivos programables para generar un sistema portátil.

3.1. Módulo de Medición para la PC

Como fue mencionado en el capítulo 2, el rango de temperatura de interés reportado para la piel de las extremidades inferiores se encuentra entre 24°C y 35°C. Se considera pertinente asignar una tolerancia para este rango de trabajo, por lo que el sistema trabajará entre 20°C y 40°C.

Los bloques que componen el sistema de medición de laboratorio, a través de la PC, se muestran en la figura 3.1, los cuales son detallados en los apartados siguientes.

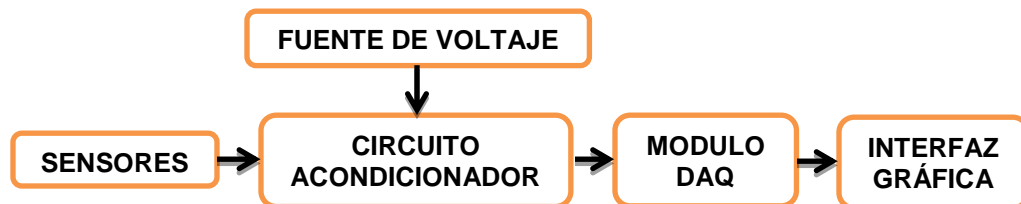


Figura 3.1. Diagrama a bloques del sistema de adquisición de temperatura para la PC.

3.1.1. Sensor de Temperatura

Las mediciones que se pretenden efectuar requieren llevarse a cabo por periodos prolongados, y dado que el sensor de temperatura es de contacto, estos podrían llegar a ser una incomodidad para el sujeto. Debido a esto se buscó un sensor cuyas



dimensiones no provocaran una gran molestia, encontrando útil el termistor miniatura 10K3A1B (Betatherm, USA) con un diámetro de 2.4 mm [18].

Los termistores son dispositivos sensores que basan su funcionamiento en la variación de la resistencia eléctrica que presenta un material con cambios en la temperatura del ambiente con el que tiene contacto. Son fabricados a partir de óxidos de metales de transición (manganeso, cobalto y níquel) [19].

Existen dos tipos según su variación de resistencia, los de coeficiente de temperatura negativo (NTC, *Negative Temperature Coefficient*) y los de coeficiente de temperatura positivo (PTC, *Positive Temperature Coefficient*). El primero, y más común comercialmente hablando, presenta una resistencia eléctrica que disminuye al aumentar la temperatura en tanto que el segundo presenta el efecto contrario [20].

Estos sensores, generalmente, tienen una sensibilidad muy alta ($\sim 200 \Omega/^{\circ}\text{C}$), lo cual ayuda a proporcionar medidas exactas en aplicaciones de menor temperatura [20]. Poseen como desventajas el no ser lineales y están limitados para usos con un rango de temperatura no mayor a 150°C .

El termistor miniatura 10K3A1B es del tipo NTC con un rango de trabajo de 0 a 70°C , una resistencia nominal de $10 \text{ k}\Omega$ para una temperatura de $25^{\circ}\text{C} \pm 0.2^{\circ}\text{C}$, posee una respuesta de 1 segundo típicamente en líquidos y una alta sensibilidad (alrededor de $300 \Omega/^{\circ}\text{C}$).

Para garantizar el contacto entre los termistores miniatura y la piel de la extremidad inferior, se construyó una estructura para montaje y sujeción. La estructura o banda está hecha de etilvinilacetato (goma EVA) y cintas ajustables de velcro, además de acetato preformado esféricamente para montar el termistor y asegurar el contacto entre el dispositivo y la piel, figura 3.2.



Se construyeron 5 bandas, de las cuales una de ellas posee las dimensiones necesarias para colocarse únicamente en el hallux, figura 3.2 derecha.

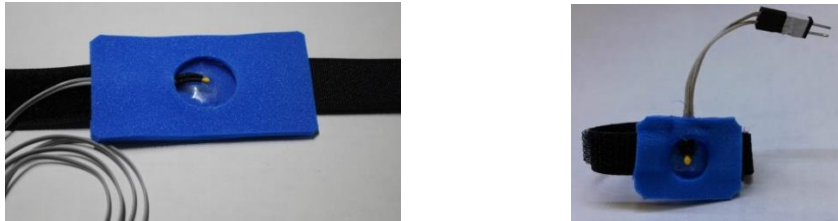


Figura 3.2. Banda con termistor utilizada para el registro de la temperatura. Izquierda: para medición en tobillo, pantorrilla o muslo. Derecha: para la medición en el hallux.

3.1.2. Circuito de Acondicionamiento

Como se mencionó, el comportamiento de los termistores no es lineal, por lo que es necesario llevar a cabo una etapa de linealización antes de poder trabajar con su respuesta.

Los valores de resistencia del termistor obtenidos en el rango de trabajo, 20°C a 40°C, con los que se trabajó para la linealización se muestran en la tabla 3.1, cuya curva característica se encuentra en la figura 3.3.

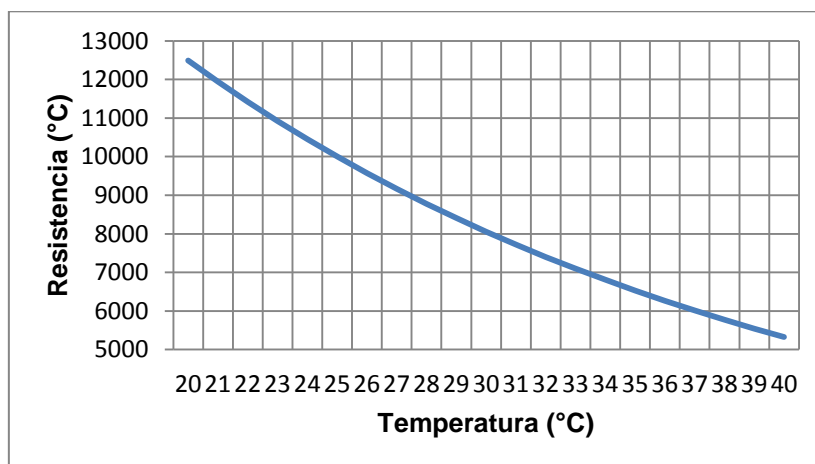


Figura 3.3. Curva característica del termistor 10K3A1B sin linealizar.



Temperatura (°C)	Resistencia (Ω)	Temperatura (°C)	Resistencia (Ω)
20	12490	31	7722
21	11940	32	7404
22	11420	33	7098
23	10920	34	6808
24	10450	35	6532
25	10000	36	6268
26	9574	37	6016
27	9166	38	5776
28	8778	39	5546
29	8408	40	5326
30	8058		

Tabla 3.1. Valores típicos de resistencia del termistor dados por el fabricante [18].

El procedimiento llevado a cabo para linealizar la respuesta del termistor [21], suele utilizar un resistor en paralelo (R_p) cuyo valor es calculado a través del método de los tres puntos (ecuación 3.1), el cual consiste en tomar tres valores de temperatura y sus respectivos valores de resistencia para el rango deseado (figura 3.4). Los tres datos de temperatura a considerar deben ser equidistantes entre sí, esto es, $T_1 - T_2 = T_2 - T_3$.

$$R_p = \frac{R_{T_2}(R_{T_1} + R_{T_3}) - 2R_{T_1}R_{T_3}}{R_{T_1} + R_{T_3} - 2R_{T_2}} \quad (3.1)$$

En donde $R_{T_1}, R_{T_2}, R_{T_3}$, obtenidos de la tabla 3.1, y representados en la figura 3.4, nos dan como resultado una $R_p = 6.19 \text{ k}\Omega$.

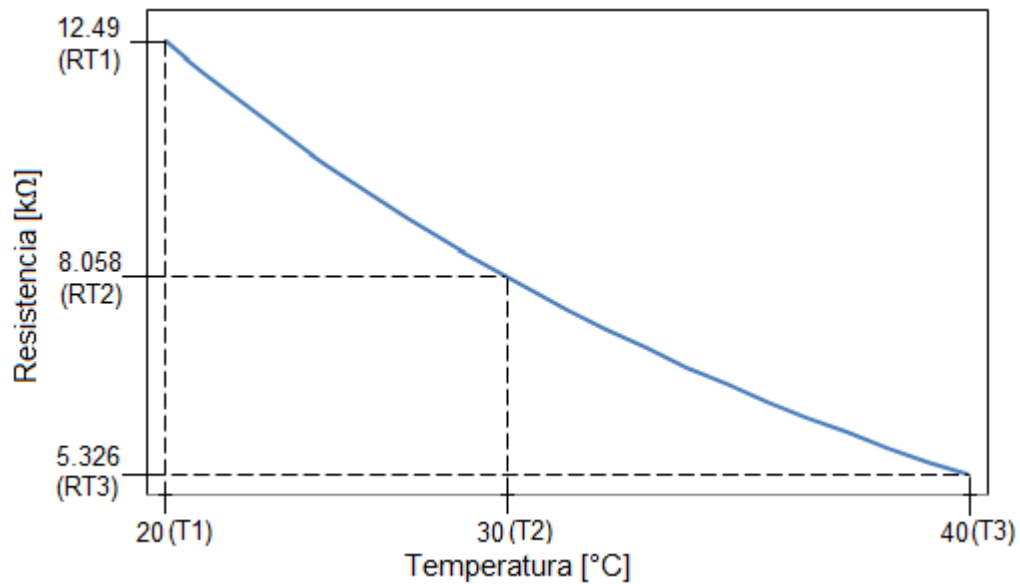


Figura 3.4. Gráfica donde se muestra los tres puntos para el rango de temperatura a medir

Para tener una salida de voltaje y a su vez un coeficiente de temperatura positivo en la respuesta del termistor, se colocó la resistencia R_p en paralelo con la resistencia del termistor en un divisor de voltaje (figura 3.5), en donde R_t representa el termistor. El valor de R_2 se determinó para obtener una sensibilidad de $S = 10 \text{ mV}/^\circ\text{C}$, sin embargo de manera práctica se obtuvo un valor de $S = 10.8 \text{ mV}/^\circ\text{C}$.

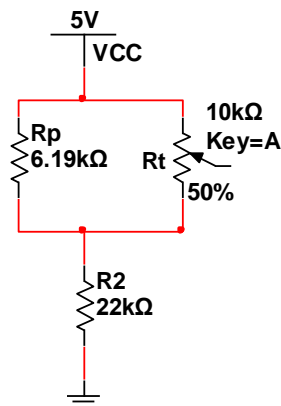


Figura 3.5. Etapa de linealización del termistor.

Como toda señal analógica requiere una etapa de filtrado, se implementó un filtro pasa bajos RC de primer orden con una frecuencia de corte de $f_c = 22 \text{ Hz}$, ajustada



experimentalmente, la cual permite eliminar la señal de 60 Hz sin atenuar la salida del sensor.

La salida en voltaje del divisor se encuentra en el rango de 4.208 V a 4.424 V, es decir, se tiene un offset de 4.0 V, y considerando que en nuestro caso se trabajó con una fuente de alimentación de ± 5.0 V, el amplificar la salida del circuito de linealización sería causante de una saturación en la etapa de amplificación.

Para ello, se corrigió mediante un offset fijo de 3.97 V con la ayuda de un divisor de voltaje, el cual es restado de la señal filtrada mediante el amplificador en modo diferencial; esta etapa se muestra en la figura 3.6.

Se eligió el amplificador operacional de alta precisión OPA4277, debido a sus excelentes prestaciones, tales como, bajo ruido, ultra bajo offset, ultra bajo corrimiento, alto CMRR, entre otras.

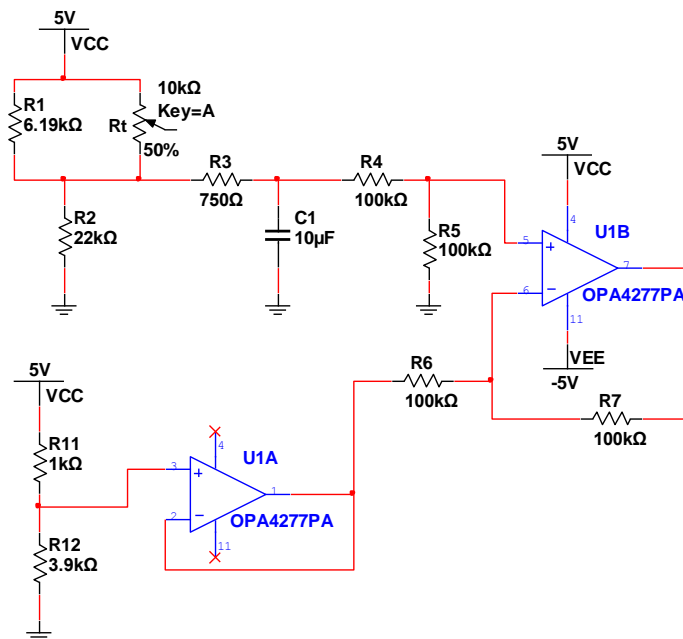


Figura 3.6. Etapa de linealización, filtrado y eliminación de offset del termistor.



Como fue descrito anteriormente, la etapa de linealización nos da una sensibilidad de $10.8 \text{ mV}/^\circ\text{C}$. Con objeto de hacer más significativos los cambios de voltaje en el intervalo de trabajo se decidió implementar una etapa de amplificación. La sensibilidad que se desea obtener es de $100 \text{ mV}/^\circ\text{C}$, para calcular el factor de amplificación A se consideró la ecuación 3.2, en donde $m_1 = 10.8 \text{ mV}/^\circ\text{C}$ y $m_2 = 100 \text{ mV}/^\circ\text{C}$, teniendo como resultado una $A=9.25$.

$$A = \frac{m_2}{m_1} \quad (3.2)$$

Esta amplificación se consideró para el cálculo de las resistencias del amplificador en la configuración de no inversor (ecuación 3.3), proponiendo una $R_1=1 \text{ k}\Omega$, y teniendo como resultado una $R_2=8.25 \text{ k}\Omega$.

$$R_2 = (A - 1)R_1 \quad (3.3)$$

Al efectuar la amplificación no solo la señal de voltaje se ve afectada, también el ruido, por lo que fue necesario aplicar una nueva etapa de filtrado, con una $f_c = 0.1 \text{ Hz}$, toda vez que la respuesta de los sensores es de muy baja frecuencia.

Como etapa final se agregó un seguidor de voltaje para lograr un adecuado acoplamiento de impedancias. El circuito final de linealización, eliminación de offset, filtrado y amplificación se observa en la figura 3.7.

Cada etapa de acondicionamiento es considerada un canal de entrada, el módulo para la PC cuenta con 4 canales permitiendo trabajar con 4 sensores de temperatura a la vez.

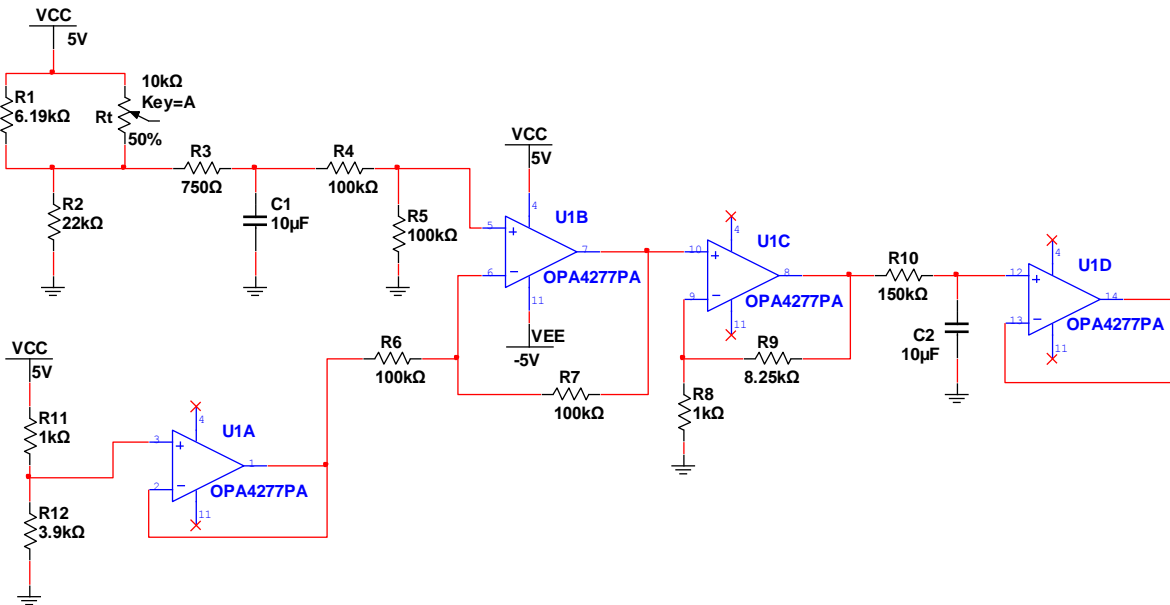


Figura 3.7. Circuito final para el acondicionamiento del sensor de temperatura.

3.1.3. Curva de Caracterización de los Sensores

Con la linealización de la respuesta de los sensores se procedió a obtener la curva de caracterización del sistema. Se obtuvo la curva de caracterización de los 5 sensores colocados en bandas, debido a que el módulo cuenta con 4 canales el sensor utilizado en el hallux es intercambiable en un canal de la tarjeta de adquisición de datos con un sensor colocado en una banda cuyo tamaño permite colocarse en el tobillo o muslo.

Para generar los cambios de temperatura en los sensores, éstos fueron sumergidos de forma individual en un baño de agua cuya temperatura era controlada mediante el Termorregulador TU-20D (Techne[®], USA) y el Enfriador EK20 (Thermo scientific[®], USA), empleando como referencia un termómetro de mercurio con 0.1°C de exactitud (figura 3.8).



El rango de temperatura para la caracterización fue de 20 a 40°C; para cada sensor se realizaron 6 barridos en este rango de temperatura, tres de subida y tres de bajada. Cada 0.5°C se tomó la lectura del termómetro de mercurio relacionándolo con el voltaje de salida del circuito de acondicionamiento medido por el multímetro (Fluke®, USA).



Figura 3.8. Sistema para la obtención de la curva de caracterización de los termistores, en donde se observa la tina con agua y el termorregulador, así como el termómetro de mercurio, el sensor y la manguera del enfriador.

Debido a que no se observó el fenómeno de histéresis durante las mediciones de subida y bajada, los seis registros se promediaron para obtener las curvas características finales, figura 3.9. La tabla con los resultados obtenidos se encuentra en el apéndice A.

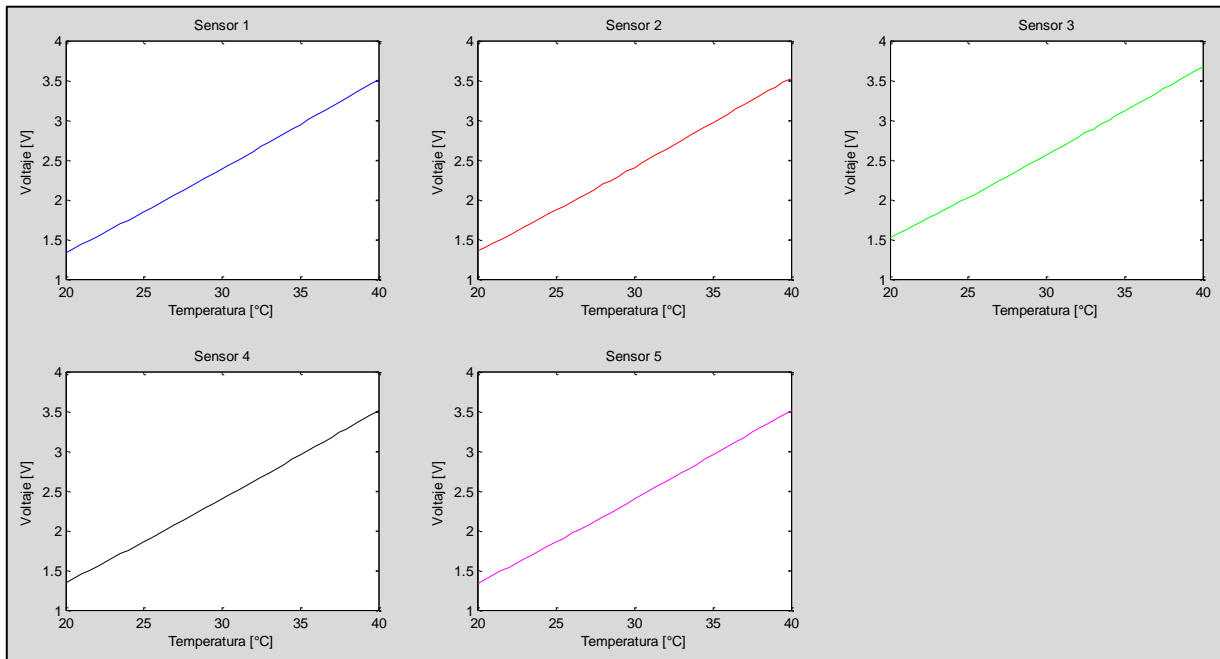


Figura 3.9. Curvas características finales para cada sensor de temperatura.

De cada curva característica se obtuvieron mediante regresión lineal la ecuación 3.4 para los sensores 1 y 5, y las ecuaciones 3.5, 3.6, 3.7 y 3.8 empleadas para el cálculo de la temperatura de los sensores 2, 3 y 4 respectivamente.

$$x = \frac{y + 0.863}{0.109} \quad (3.4)$$

$$x = \frac{y + 0.856}{0.109} \quad (3.5)$$

$$x = \frac{y + 0.6572}{0.1078} \quad (3.6)$$

$$x = \frac{y + 0.839}{0.108} \quad (3.7)$$

Donde x = temperatura ($^{\circ}\text{C}$), y =voltaje obtenido del circuito de caracterización.

3.1.4. Fuente de Alimentación

Debido a que el módulo de adquisición de datos trabaja de 0 a ± 5 V, se implementó una fuente de alimentación con ± 5 V @ 0.5 A. La figura 3.10 muestra el circuito esquemático de la fuente de alimentación.

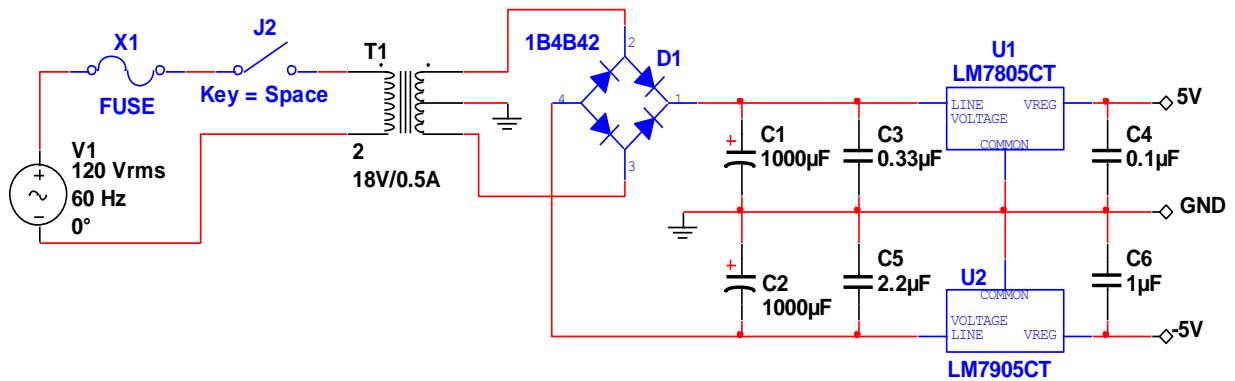


Figura 3.10. Circuito de la fuente de voltaje para el módulo de la PC.

3.1.5. Módulo de Adquisición de Datos

Para el proceso de digitalización de las señales de temperatura y comunicación con la PC del sistema de laboratorio, se empleó el módulo DAQ NI USB-6008® de Texas Instruments figura 3.11.



Figura 3.11. Módulo de adquisición de datos NI USB 6008®.

Este módulo dispone de ocho entradas analógicas sencillas también denominadas *single-ended* de ± 10 V con una resolución de 11 bits, una frecuencia máxima de muestreo de 10 kS/s, dos salidas analógicas, 12 canales digitales de entrada/salida (DIO, *Digital Input Output*) y un contador de 32 bits con una interface USB de alta velocidad.



3.1.6. Interfaz Gráfica

El programa que controla la adquisición, la conversión de los niveles de voltaje del circuito de acondicionamiento a temperatura, el despliegue y almacenamiento de datos se realizó con la ayuda del programa Labview®. La figura 3.12 muestra el panel de control y la figura 3.13 los bloques del programa desarrollado.

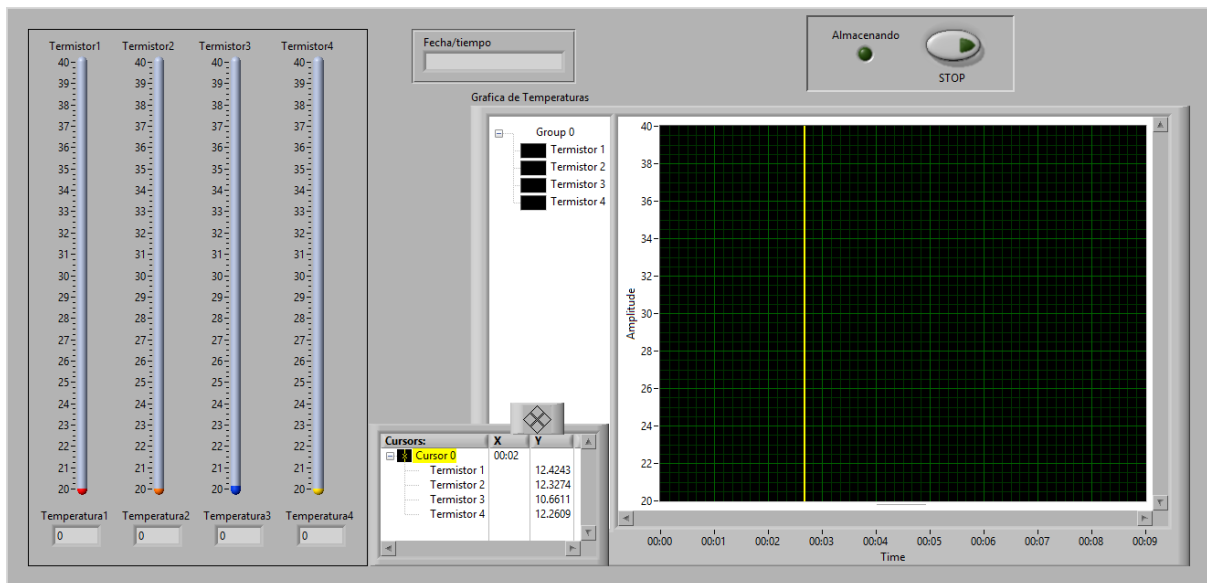


Figura 3.12. Panel de control del programa desarrollado en Labview®.

El programa permite almacenar los datos en un archivo *.TXT desde el inicio de la prueba una vez que se ha designado la ubicación y nombre del archivo.

En la interfaz se aprecian de forma gráfica y numérica los valores de temperatura registrados. A su vez, en el gráfico es posible mover un cursor que permite conocer los valores de temperatura en puntos específicos de éste.

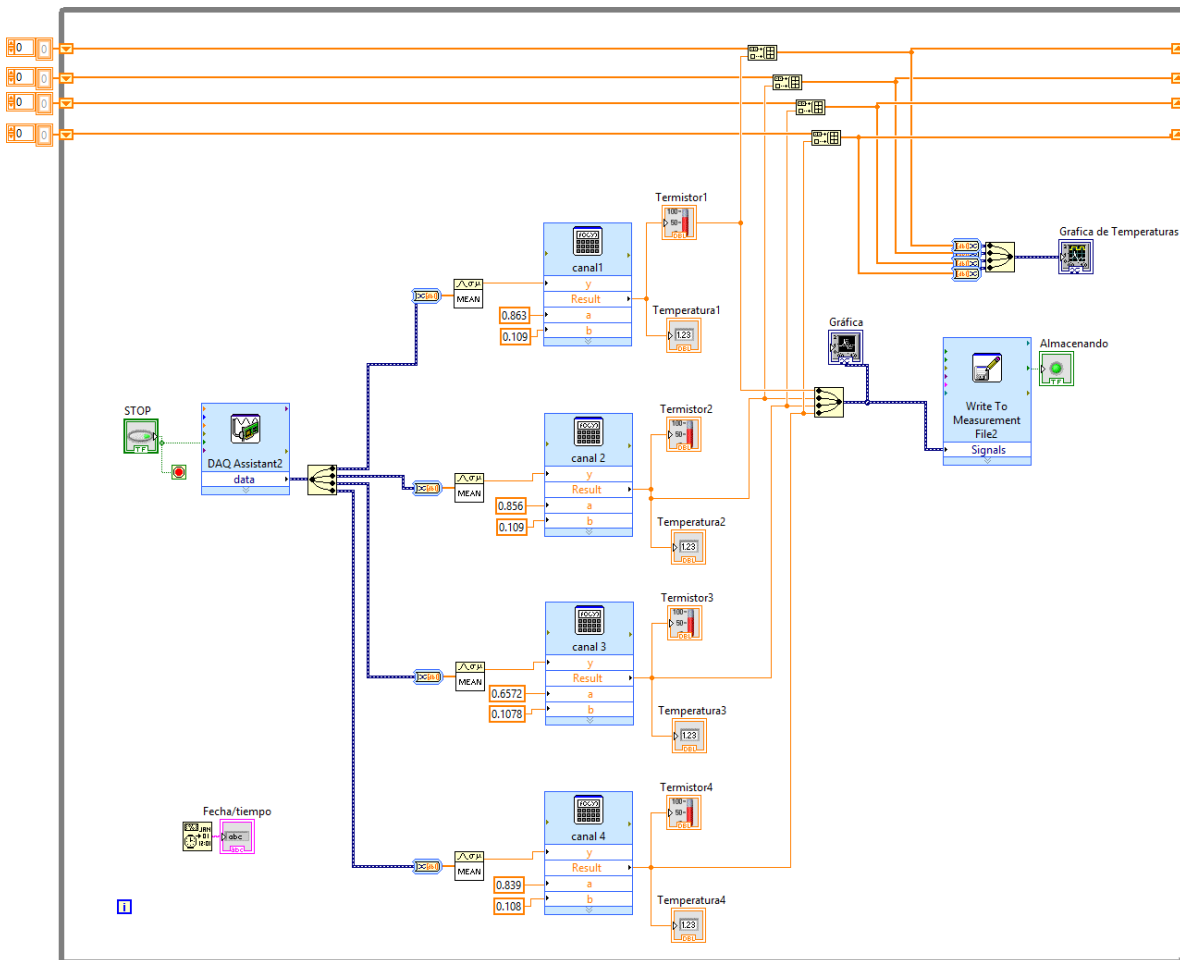


Figura 3.13. Diagrama a bloques del programa de control.

La adquisición de la señal de temperatura se efectuó promediando 128 muestras a una frecuencia de muestreo de 1 kHz, para eliminar fluctuaciones en la señal de entrada y suavizar los cambios en la visualización de los datos en el gráfico.

3.2. Módulo Portátil

Con la intención de contar con un sistema para trabajos de campo se llevó a cabo el diseño de un módulo portátil, el cual trabaja con los mismos termistores miniatura 10K3A1B, cuenta con 4 canales para trabajar en el rango de 20°C a 40°C, una pantalla gráfica de cristal líquido o GLCD (*Graphic Liquid Crystal Display*) de 128x64



pixeles y una memoria SD para el almacenamiento de datos, todos estos entorno al microcontrolador PIC18F4550, figura 3.14. La programación se realizó en lenguaje C y se compiló con el programa PIC C del CCS (*Custom Computer Services, Inc.*).

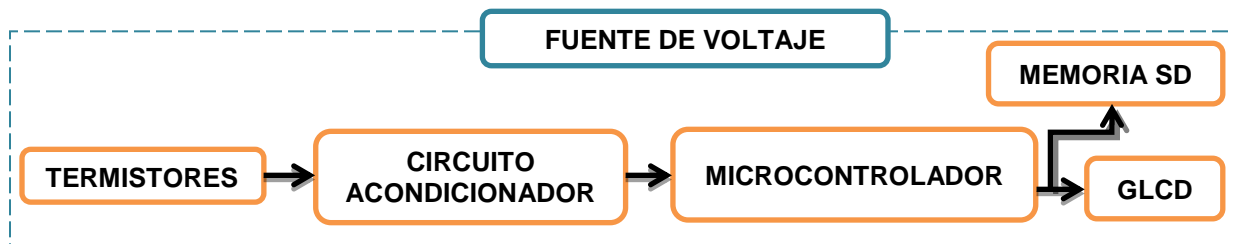


Figura 3.14. Diagrama a bloques del sistema portátil para la adquisición de la temperatura de la piel.

La etapa de acondicionamiento de la señal de los sensores de temperatura es la descrita en el subcapítulo 3.1.2, toda vez que se trata de los mismos termistores miniatura 10K3A1B (Betatherm, USA).

3.2.1. Almacenamiento de Datos

Una de las características importantes de ambos sistemas es el almacenamiento de los datos de temperatura en un archivo *.TXT. En el módulo portátil este almacenamiento se realiza en una memoria SD con una capacidad de 2 GB.

La memoria SD posee dos protocolos de comunicación, el estándar denominado BUS SD que utiliza 4 líneas paralelas para la comunicación y el tipo serial SPI (*Serial Peripheral Interface*). Este último protocolo es fácilmente implementado en los microcontroladores PIC, por tal razón, se decidió trabajar con este dispositivo para realizar la comunicación entre ambos [22].

Las memorias SD poseen 9 pines de conexión, de los cuales, al trabajar con el protocolo SPI 4 líneas son de comunicación: *Data In* (DI), *Data Out* (DO), *Chip Select*



(CS) y *clock* (CLK); 2 son líneas de alimentación, 1 para tierra, y 2 son reservados (RSV), tabla 3.2. El rango de voltaje de operación permitido es de 2.7 V a 3.6 V [22].

PIN	Nombre	Descripción
1	CS	Activación de tarjeta
2	DI (MOSI)	Comandos de datos desde el host
3	VSS	GND
4	VDD	Alimentación (3.3V)
5	SCLK	Reloj
6	VSS	GND
7	DO (MISO)	Datos hacia el host
8	RSV	Reservado
9	RSV	Reservado

Tabla 3.2. Pines de conexión de la memoria SD mediante comunicación SPI.

Para la escritura de la SD se adquirió el módulo de evaluación, figura 3.15, el cual cuenta con un regulador de voltaje de 5 V a 3.3V, resistencias de pull-up para las líneas de datos y el socket para la memoria.



Figura 3.15. Módulo de lectura para la memoria SD.

Debido a que la memoria trabaja con una alimentación máxima de 3.3 V es necesario cambiar el nivel de los datos que van del microcontrolador a la memoria, para este fin se utilizó el buffer SN74LV4T125. La figura 3.16 muestra el diagrama esquemático de la etapa correspondiente al módulo y el cambiador de nivel para su conexión con el PIC18F4550.

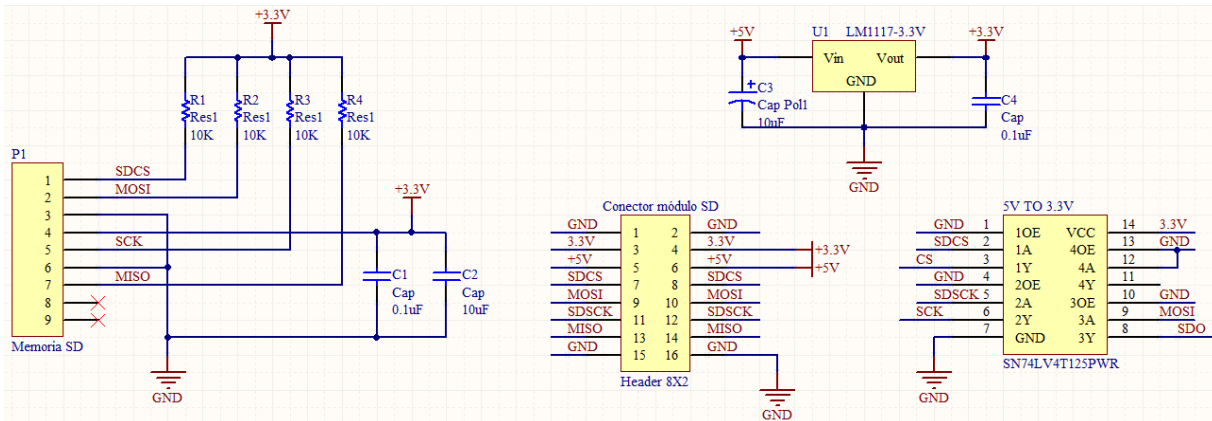


Figura 3.16. Diagrama esquemático del módulo y el cambiador de nivel para la lectura de la memoria SD.

Para generar un archivo tipo *.txt es necesario trabajar en el formato fat16 en la memoria. La librería que trabaja con este formato se obtuvo del foro de programación ucontrol [23], del cual se modificaron los pines de comunicación entre el PIC18F4550 y el módulo de la SD card, así como la frecuencia de trabajo a 10 MHz. Las librerías utilizadas se encuentran en el apéndice B.

3.2.2. Despliegue de la Temperatura

Los datos adquiridos son mostrados en una pantalla gráfica de cristal líquido o GLCD (*Graphic Liquid Crystal Display*) con una resolución de 128x64 pixeles. El envío de los datos para su despliegue es por medio de un bus de 8 bits. Los pines de conexión para el control de la GLCD se muestran en la tabla 3.3 [24].

El programa PIC C cuenta con una librería para el manejo de la GLCD, de la cual se modificaron los pines para la conexión entre la pantalla y el microcontrolador. Esta librería permite la inicialización de la GLCD, el encendido y apagado de un pixel, así como el despliegue de texto y formas geométricas básicas. El código con las modificaciones realizadas se encuentra en el apéndice C.



PIN	SÍMBOLO	DESCRIPCIÓN	FUNCIÓN
1	CSA	Chip Select A	CS1=1, señal para IC1
2	CSB	Chip Select B	CS2=1, señal para IC2
3	VSS	Tierra	GND
4	VDD	Alimentación para el circuito lógico	+5 V
5	V0	Ajuste de contraste de la LCD	
6	D/I	Selección del registro datos/instrucción	D/I=0: registro de instrucción D/I=1: registro de datos
7	R/W	Selección lectura/escritura	R/W=0: escritura del registro R/W=1: lectura del registro
8	E	Señal de habilitación	
9-16	DB0-DB7	Líneas de datos	8 bits
17	RST	Señal de reset	RST=0, apagado de la pantalla
18	VEE	Voltaje de control negativo	-10 V
19	A	Ánodo del led para la luz de fondo	+5 V
20	K	Cátodo del led para la luz de fondo	GND

Tabla 3.3. Pines de conexión de la GLCD.

3.2.3. Microcontrolador

La adquisición, procesamiento y control del sistema se encuentra a cargo del microcontrolador PIC18F4550, el cual cuenta con una memoria flash de 32 kB, 256 bytes de memoria EEPROM, 13 canales Analógicos-Digitales (A/D) de 10 bits con tiempos de adquisición programables, puertos para comunicación por SPI e I²C, protección de código y programación en sitio, entre otras características [25].

La figura 3.17 muestra el diagrama eléctrico a bloques del módulo portátil, en el que se observa la conexión entre el microcontrolador, la GLCD, el circuito de acondicionamiento, la memoria SD y un módulo RTC. Los diagramas esquemáticos correspondientes a estas etapas se muestran en el apéndice D.

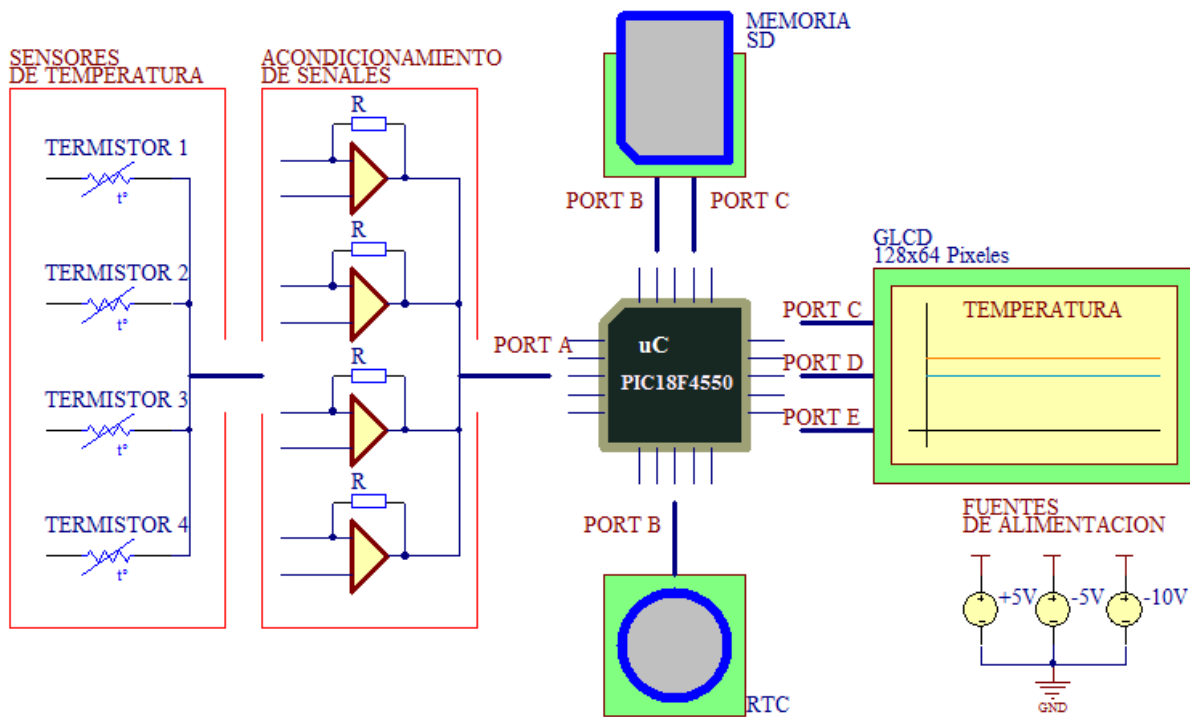


Figura 3.17. Diagrama eléctrico a bloques del módulo portátil.

Para el control del tiempo de muestreo, se agregó un módulo RTC (*Real Time Clock*), el cual se restablece cada vez que se enciende el módulo portátil, iniciando el conteo desde cero y cuyos valores se van adquiriendo conforme se muestrea la señal de los sensores de temperatura.

Toda vez que los sensores están caracterizados para canales específicos, estos y sus entradas correspondientes se encuentran debidamente identificados en ambos sistemas.

El diagrama de flujo del programa principal del módulo portátil se observa en la figura 3.18. El código desarrollado se encuentra en el apéndice E.

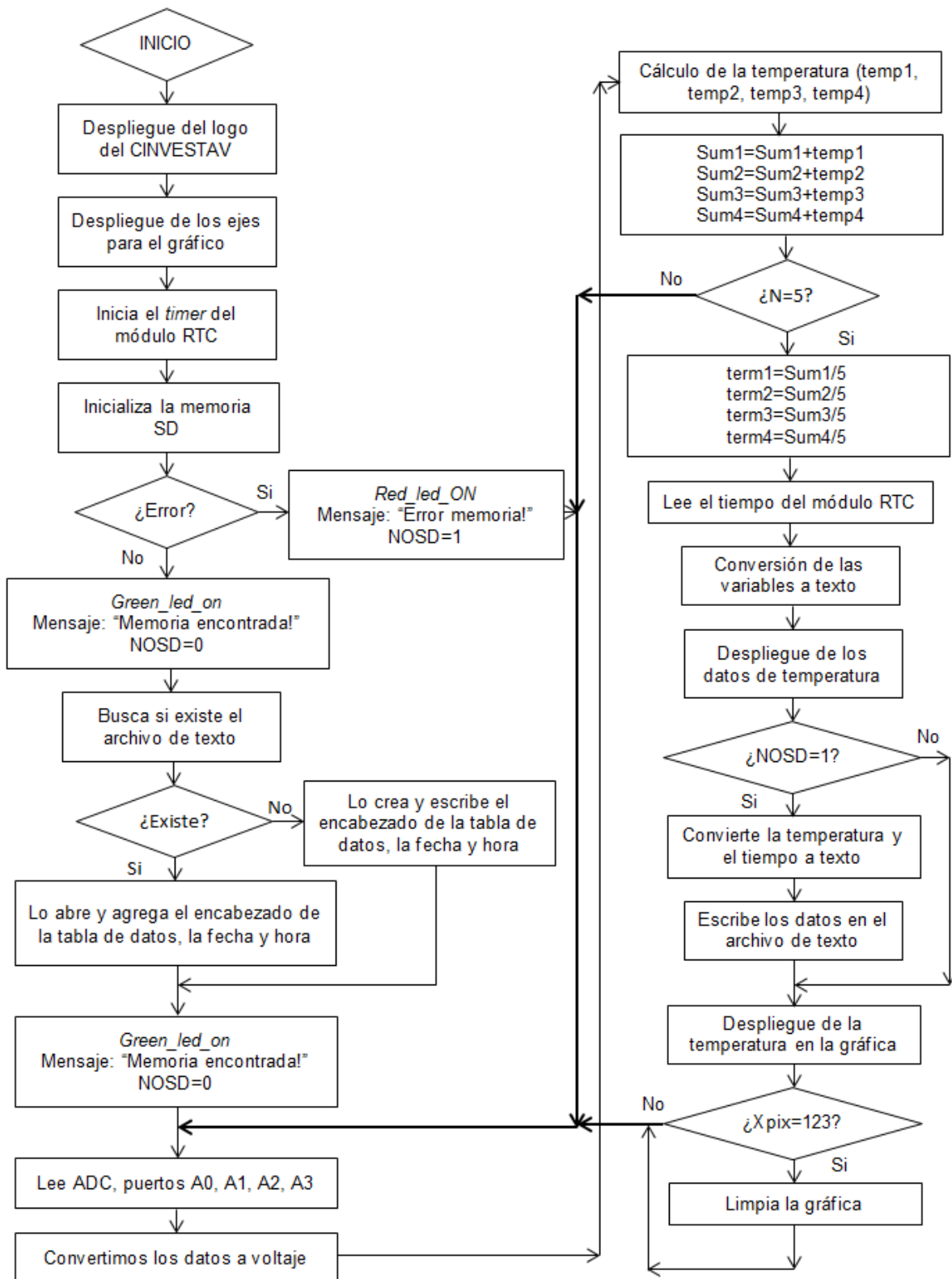


Figura 3.18. Diagrama de flujo del programa principal del módulo portátil.



CAPÍTULO 4. PRUEBAS

En este estudio se buscará estimular el flujo sanguíneo en sujetos de prueba sanos con el objetivo de medir la velocidad de flujo y determinar su relación con la temperatura medida de la piel en las extremidades inferiores.

Existen tres factores principales que afectan el flujo sanguíneo [2]:

- 1) presión arterial,
- 2) resistencia vascular,
- 3) retorno venoso.

La resistencia vascular depende del tamaño de la luz del vaso sanguíneo, la viscosidad de la sangre y el largo total del vaso [2]. De estos factores la luz de los vasos puede ser modificada por la contracción y relajación de los músculos debidos al ejercicio, estimulando a su vez el retorno venoso [2], el cual se ve ayudado con cambios posturales como la elevación de las extremidades

Teniendo estas ideas como base, se proponen dos protocolos experimentales que estimulen el flujo sanguíneo, uno a través de un cambio postural y otro por medio de un ejercicio.

4.1. Instrumentación

El equipo a emplear para obtener la velocidad de flujo sanguíneo en las extremidades inferiores es un flujómetro Doppler Modelo 810-B (Parks Medical Electronics, INC.; Aloha, Oregon USA), el cual entrega un valor en frecuencia que, de acuerdo a la ecuación (4.1), dada por el manual de usuario, permite conocer el dato de interés.



$$Velocidad\ de\ Flujo\ Sanguíneo = \frac{F(1.56 \times 10^5)}{2(8.1 \times 10^6) \cos(\theta)} \quad (4.1)$$

Donde θ es el ángulo del transductor respecto de la dirección del flujo sanguíneo en el vaso, el cual se fijó en 45° para cada medición y F es la diferencia absoluta entre la frecuencia de la señal emitida y la frecuencia de la señal recibida, la cual es dada por el equipo.

Para la medición de la temperatura de la piel se empleará el módulo de PC en la prueba de cambios posturales y el módulo portátil en la prueba con ejercicio físico.

4.2. Protocolo de Medición A

En este protocolo, el sujeto de prueba realizó dos cambios posturales en su extremidad inferior derecha. La hipótesis es que al elevar la extremidad inferior se obtendrá un cambio en la velocidad de flujo sanguíneo el cual conllevará a un cambio en la temperatura de la piel de dicha extremidad.

4.2.1. Sujetos A

Cinco mujeres y nueve hombres, no fumadores, entre 24 y 56 años de edad participaron en las pruebas. A cada uno se le pidió realizar dos cambios de posición y altura de su pierna derecha como estímulo. Todos fueron informados de la naturaleza del estudio y proporcionaron su consentimiento.



4.2.2. Condiciones del Laboratorio A

Las pruebas fueron efectuadas en un laboratorio a una temperatura ambiente de $23^{\circ}\text{C} \pm 0.5$, con una humedad relativa de $55\% \pm 5$.

4.2.3. Puntos de Medición A

Debido a que el intercambio de calor entre el cuerpo y el ambiente se manifiesta significativamente en la región más alejada de las extremidades, se eligió el hallux como primer punto de medición de temperatura; el tobillo, a 4 cm del punto para la medición de velocidad de flujo y la pantorrilla (a 16 cm de separación con respecto al sensor del tobillo) como valor de referencia para la comparación con el punto más alejado (fig. 4.1).

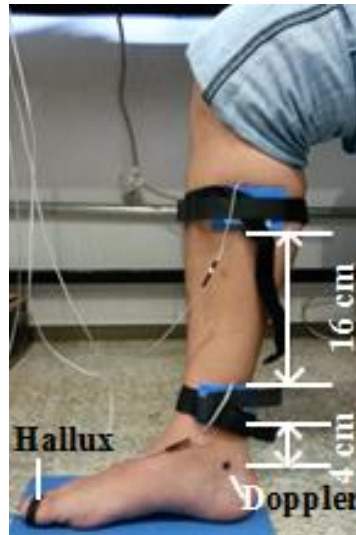


Figura 4.1. Puntos de medición en la extremidad inferior derecha para el protocolo A.

La medición de la velocidad del flujo sanguíneo se realizó en la arteria tibial posterior (fig. 4.2).

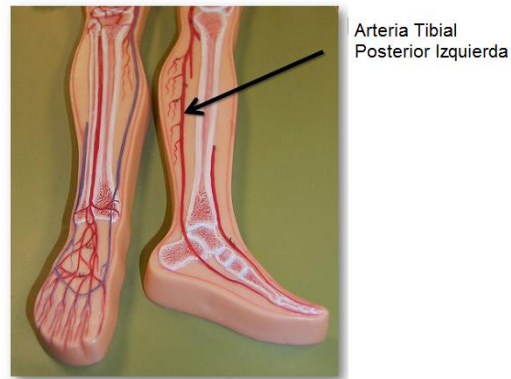


Figura 4.2. Posición de la arteria tibial posterior en la extremidad izquierda.

4.2.4. Descripción del Protocolo A

Cada prueba duró aproximadamente 40 minutos dividida en tres fases (figura 4.3), durante las cuales el sujeto permaneció en posición sedente. Antes de iniciar la prueba, se le pidió a cada sujeto descubrirse la pierna derecha para efectuar un procedimiento de limpieza de las zonas a medir. Después de este procedimiento se inició la primera fase de 10 minutos de adaptación a la temperatura ambiente, con las piernas del sujeto en ángulo recto. Dentro de este tiempo se localizó con el flujómetro Doppler el punto óptimo para la medición en la arteria (marcándose con pintura no tóxica); de igual manera, durante esta fase se colocaron los sensores de temperatura. La segunda fase consistió en pedir a cada voluntario elevar y dejar descansar la pierna derecha sobre un cojín durante 15 min (figura 4.4), quedando a un ángulo aproximado de 20° con respecto al suelo. En la tercera fase se le pidió al sujeto volver a la posición inicial apoyando el pie sobre una base con una altura de 9 cm. A partir del inicio de la segunda y hasta finalizar la tercera fase, se almacenaron y registraron los datos de temperatura y velocidad de flujo sanguíneo con el flujómetro Doppler cada 5 min aproximadamente.

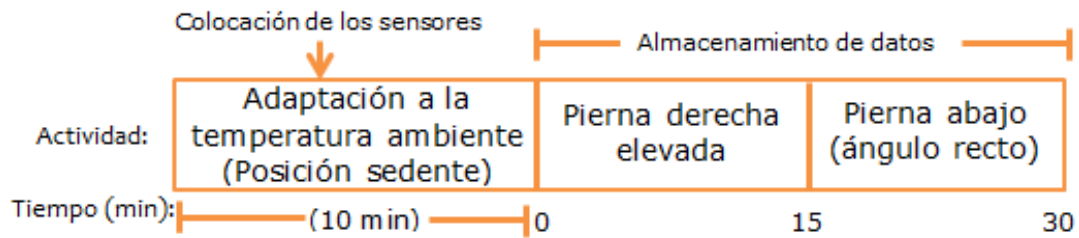


Figura 4.3. Esquema de las fases del protocolo experimental A.

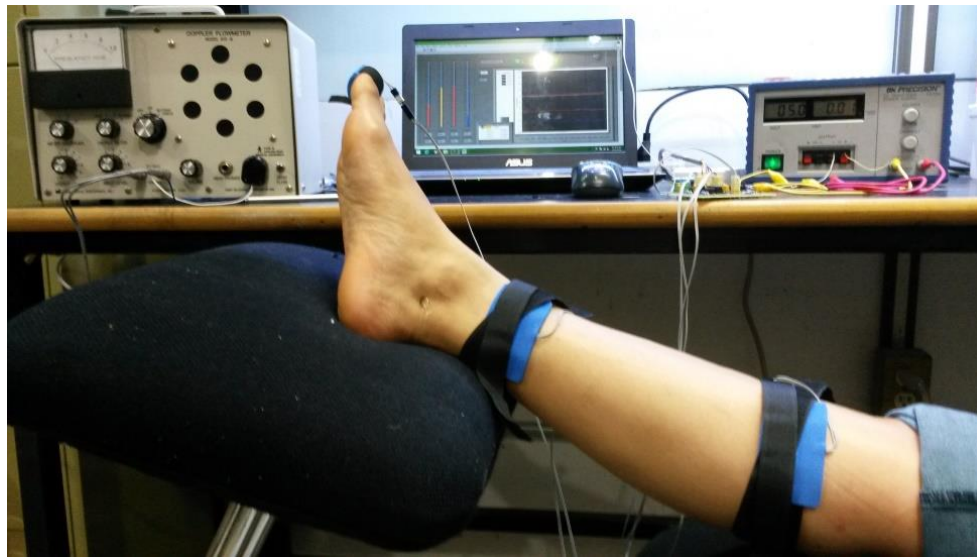


Figura 4.4. Posición de la pierna derecha durante la segunda fase de la prueba A.

4.2.5. Procesamiento de los Datos A

Se realizaron dos tipos de análisis de los datos de temperatura y velocidad de flujo sanguíneos obtenidos en este protocolo. En el primero se promediaron y graficaron los datos para cada punto de medición de todos los sujetos con el fin de observar su tendencia. En el segundo, se promediaron los datos para cada punto de medición por sujeto y fase de la prueba, para observar si existe un aumento o disminución en velocidad de flujo y temperatura en los cambios de posición.



4.3. Protocolo de Medición B

En el segundo protocolo de medición, los sujetos fueron sometidos a un ejercicio físico. La hipótesis planteada es que al someter a un grupo de músculos del cuerpo a un ejercicio en específico, el flujo sanguíneo será estimulado lo cual repercutirá en una modificación de la temperatura de la piel.

Un ejercicio simple y focalizado de las extremidades inferiores es la extensión de talones el cual estimula la pantorrilla. De acuerdo al trabajo de los autores D. Formenti et al. [26] el tiempo máximo que una mujer, atleta o sedentaria, puede hacer la extensión de talones sin fatigarse es de entre 2 a 2.30 min. Por lo que en este protocolo se solicitará a los voluntarios realizar este ejercicio por el límite inferior de 2 min.

4.3.1. Sujetos B

Se reclutaron 15 sujetos de ambos sexos, cinco mujeres y diez hombres, no fumadores, de entre los 24 a 32 años de edad. A todos se les pidió vestir con ropa holgada y que la ingesta de alimentos fuera de al menos 2 horas previas a la medición. Todos fueron informados de la naturaleza del estudio y proporcionaron su consentimiento.

4.3.2. Condiciones del Laboratorio B

Las pruebas fueron efectuadas en un laboratorio a una temperatura ambiente de $20^{\circ}\text{C} \pm 0.5$, con una humedad relativa de $56\% \pm 5$.



4.3.3. Puntos de Medición B

En este protocolo la temperatura de la piel fue medida en la cara interna de la extremidad inferior derecha. El primer punto de medición fue el hallux, el segundo fue el dorso del pie a 3 cm de la cara frontal de la pierna, el tercer punto fue el tobillo, a 3 cm por arriba del punto para la medición de la velocidad de flujo y el cuarto fue la pantorrilla, con 14 cm de separación respecto al sensor del tobillo (fig. 4.5). La medición de velocidad de flujo sanguíneo se efectuó en la arteria tibial posterior.

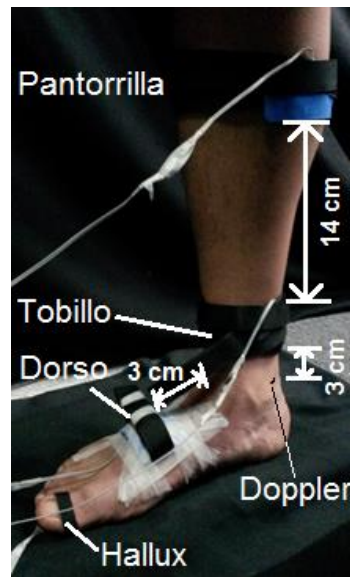


Figura 4.5. Puntos de medición en la extremidad inferior derecha para el protocolo B.

4.3.4. Descripción del Protocolo B

Para llevar a cabo este protocolo, se dividió en tres etapas (fig. 4.6): la primera corresponde al acondicionamiento de la extremidad a la temperatura ambiente, la segunda al ejercicio de extensión de talones y finalmente la tercera a la etapa de recuperación del sujeto. A lo largo de la primera y segunda etapa el sujeto permaneció de pie, con los pies separados a la altura de los hombros, para evitar



movimientos que generaran perturbaciones en la temperatura de la piel previos al ejercicio.

Tomando en cuenta que el cambio en el flujo sanguíneo debido al ejercicio es el interés principal, no se consideró necesario monitorear la recuperación del músculo durante todo el tiempo, que la bibliografía indica es de aproximadamente 30 min [1]. En nuestro caso, propusimos un tiempo de recuperación de 7 min para realizar el registro final de velocidad de flujo y temperatura.

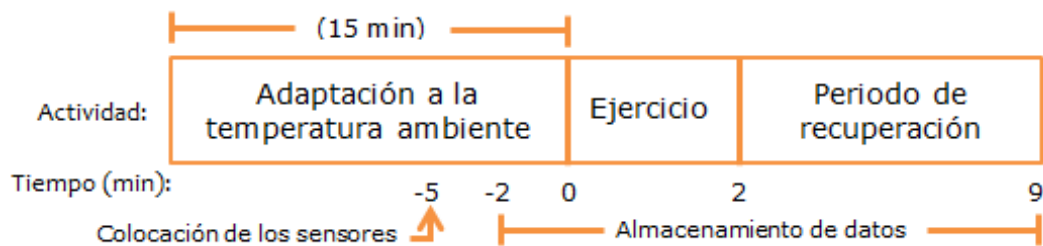


Figura 4.6. Esquema de las fases del protocolo experimental B.

Antes de iniciar la primera etapa se solicitó a los participantes retirarse el calzado y calcetines y colocarse sobre dos superficies, una para cada pie, con una altura aproximada de 3 cm, cubiertas por una tela para evitar pérdidas importantes de calor entre el pie y la superficie de contacto. Posteriormente, se les limpiaron las zonas de medición de velocidad de flujo y temperatura. Una vez realizado esto, dio inicio la etapa de adaptación a la temperatura ambiente de 15 minutos, dentro de este tiempo se localizó con el flujómetro Doppler el punto óptimo para la medición en la arteria tibial posterior. Los sensores de temperatura fueron colocados a los 10 min de iniciada esta etapa.

El registro de los datos de temperatura se llevó a cabo desde 2 min antes de la finalización de la primera etapa para continuar hasta finalizar la tercera. El inicio de la segunda etapa es considerado el tiempo 0 para el registro de los datos, dentro de esta etapa se llevó a cabo el ejercicio de extensión de talones para estimular el flujo



por un periodo de 2 min. El compás del movimiento se definió por un metrónomo para estandarizar el número de repeticiones; los sujetos levantaron en 1 s los talones lo más alto que podían y en el segundo siguiente los bajaron regresando a la posición inicial.

En la tercera fase de recuperación, se solicitó a los sujetos colocarse en una posición semisedente después de realizar la medición de la velocidad de flujo sanguíneo inmediatamente posterior al ejercicio, sin despegar las plantas de los pies del suelo (figura 4.7). Con esto se deseaba evitar dolores posteriores por la falta de movimiento después del ejercicio, sin generar obstrucciones en el retorno venoso debido al ángulo y presión en las piernas ocasionadas por la posición sedente.



Figura 4.7. Posición del sujeto de medición en la tercera fase del protocolo B.

4.3.5. Procesamiento de los Datos B

Los datos de temperatura y velocidad de flujo sanguíneos obtenidos en este protocolo se promediaron para observar la tendencia en temperatura y velocidad de flujo, especialmente durante la fase de ejercicio.



CAPÍTULO 5. RESULTADOS Y DISCUSIÓN

5.1. Protocolo A. Cambio de Posición de la Extremidad Inferior Derecha.

Se graficó el promedio de los datos de temperatura en cada punto de medición contra los promedios de los valores de velocidad de flujo sanguíneo obtenidos durante la segunda y tercera fases de todos los sujetos (figura 5.1), para mostrar la tendencia de cambio entre estos parámetros.

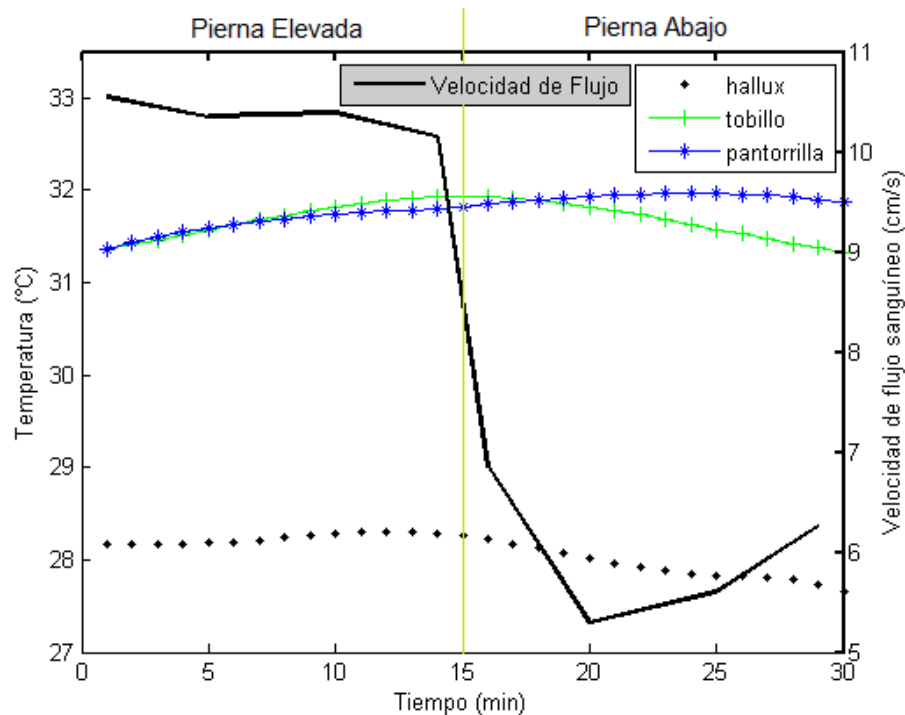


Figura 5.1. Promedios de temperatura para el hallux, tobillo y pantorrilla comparados con el promedio de las velocidades de flujo de todos los sujetos.

De las tres curvas de temperatura se observa que el hallux es aproximadamente 3°C menor en relación al tobillo y pantorrilla, en tanto que entre estos la diferencia de temperatura es menor a 1°C.



Se observa en la figura que cuando la pierna se encuentra elevada la tendencia de la temperatura es ascendente en tanto que los valores de velocidad de flujo sanguíneo están por arriba de los 10 cm/s. Por el contrario, cuando la pierna se encuentra abajo, la tendencia de la temperatura en tobillo y hallux es descendente, en tanto que la velocidad de flujo cae por debajo de los 7 cm/s.

El comportamiento promedio demuestra que el mantener la pierna en una posición elevada, conlleva una velocidad del flujo sanguíneo mayor que en el caso de la pierna abajo (posición normal sedente). De igual forma, la temperatura en el hallux incrementa 0.3°C , y en el tobillo y pantorrilla hasta 0.7°C en el primer intervalo de tiempo. Para el caso de pierna abajo, en el hallux la temperatura desciende aproximadamente 0.6°C , el tobillo regresa a su valor inicial de aproximadamente 31.4°C , y en el caso de la pantorrilla la temperatura continúa incrementándose hasta el minuto 25.

En la tabla 5.1 se observa el promedio de los datos de temperatura cuando la pierna se encuentra elevada y cuando la pierna se encuentra abajo para cada punto de medición por sujeto. Por su parte la tabla 5.2 muestra el promedio de los datos de velocidad de flujo sanguíneo.

De estas tablas, se resaltan los cambios de temperatura y velocidad de flujo entre las dos posturas de análisis por sujeto. Observando que en el hallux 10 de 14 sujetos presentan una disminución de temperatura, casillas sombreadas, así como 9 sujetos en el tobillo. En el caso de la pantorrilla, 11 sujetos tienen un incremento en temperatura en tanto que la velocidad de flujo disminuyó en 11 de los 14 sujetos.

De acuerdo a este protocolo, los resultados preliminares obtenidos demuestran una relación directa entre la temperatura de la piel de los puntos anatómicos definidos de la extremidad inferior y la velocidad de flujo sanguíneo en el 57% (8 de 14) de los sujetos.



Sujeto	Hallux		Tobillo		Pantorrilla	
	Pierna elevada	Pierna abajo	Pierna elevada	Pierna abajo	Pierna elevada	Pierna abajo
1	22.09	21.83	29.21	29.04	30.92	30.91
2	32.95	32.88	31.77	32.12	32.00	32.14
3	33.08	31.39	33.82	33.46	33.33	33.21
4	27.33	27.04	31.87	32.26	31.62	31.37
5	27.70	25.82	30.63	30.41	30.11	30.40
6	29.56	30.12	31.42	31.84	31.48	32.84
7	26.87	27.33	32.61	32.75	32.88	32.90
8	31.16	31.57	32.11	32.46	31.91	32.30
9	25.84	24.27	31.80	31.34	31.86	31.81
10	24.67	23.77	30.78	30.32	30.66	30.76
11	29.92	29.01	33.64	32.85	32.63	32.99
12	22.79	22.19	29.93	29.58	30.51	31.11
13	29.88	31.12	32.76	32.21	32.73	33.10
14	31.50	31.25	31.91	31.61	30.94	31.20

Tabla 5.1. Temperatura promedio [°C] en los puntos de medición correspondientes a las posiciones de la pierna en cada sujeto. Las casillas sombreadas indican disminución de temperatura.

Sujeto	Pierna elevada	Pierna abajo
1	2.32	2.04
2	16.07	10.76
3	5.72	4.36
4	14.44	5.45
5	2.32	1.09
6	14.98	4.49
7	5.86	6.67
8	15.12	9.94
9	16.48	3.00
10	6.81	3.00
11	9.26	3.81
12	7.22	1.91
13	13.62	13.21
14	14.84	14.30

Tabla 5.2. Velocidad de flujo sanguíneo promedio [cm/s] en los puntos de medición correspondientes a las posiciones de la pierna en cada sujeto. Las casillas sombreadas indican disminución de temperatura.



Estos resultados pueden deberse a que al elevar la extremidad inferior se mejora el retorno venoso, lo cual conlleva a una estimulación de la circulación, generando un aumento en la velocidad de flujo sanguíneo y en consecuencia provoca un incremento de la temperatura de la piel en la extremidad.

Por otro lado, al tener la pierna abajo en ángulo recto, la acción de la gravedad podría afectar a este retorno venoso teniendo una disminución en la velocidad de flujo, y en consecuencia un decremento en la temperatura de la piel en la extremidad.

Sin embargo, se considera necesario realizar pruebas con intervalos de tiempo más largos y más casos de estudio; así como observar con más detalle el comportamiento de la pantorrilla para reforzar estos resultados.

De este protocolo se realizó el artículo “Desarrollo de un sistema de monitoreo del estado de irrigación sanguínea en extremidades inferiores en humano a través de la medición de temperatura” presentado en el VI Congreso Latinoamericano de Ingeniería Biomédica (CLAIB 2014) realizado en Paraná, Entre Ríos, Argentina, encontrado en el apéndice F.

5.2. Prueba B. Ejercicio Físico en las Extremidades Inferiores.

Para el análisis de los datos, en esta prueba se efectuó un procedimiento similar al de la prueba anterior, es decir, se graficó el promedio de los datos de temperatura en cada punto de medición de todos los sujetos, obtenidos desde los 2 min de registro previos al ejercicio hasta la finalización del tiempo de reposo, contra los promedios de los valores de velocidad de flujo sanguíneo medidos antes y después del ejercicio y cada minuto durante la etapa de reposo, figura 5.2.



De la gráfica, se observa que el hallux presenta una temperatura aproximadamente 3°C menor respecto a los otros puntos de medición. Su comportamiento indica una tendencia a la baja desde el periodo previo al ejercicio, denominado basal en la gráfica, hasta un minuto posterior al haber concluido la actividad física que es en donde tiende a incrementarse.

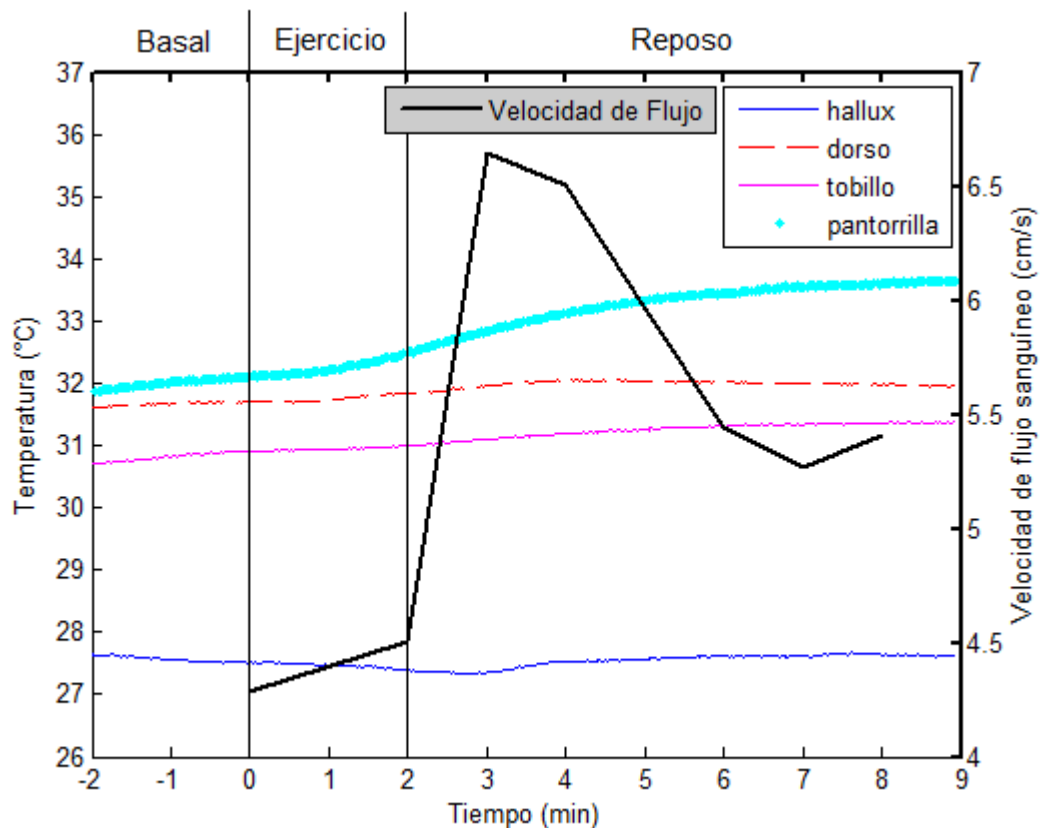


Figura 5.2. Promedios de temperatura para el hallux, dorso, tobillo y pantorrilla comparados con el promedio de las velocidades de flujo de todos los sujetos.

En el caso del dorso y el tobillo, su temperatura muestra un tendencia ascendente a un minuto de haber iniciado el ejercicio, ambos aumentando aproximadamente 0.5°C respecto a la temperatura basal.



La pantorrilla, muestra un retardo en su incremento de temperatura de 1 minuto después de iniciado el ejercicio y continuó incrementándose hasta casi 1.5°C por el resto de la prueba.

Por su parte, la velocidad de flujo sanguíneo no presenta un cambio significativo entre la medición realizada antes y después del ejercicio (no más de 0.5 cm/s). Sin embargo, un minuto después de finalizada la actividad física se observa un aumento por arriba de los 6.5 cm/s .

El comportamiento de la temperatura observada en el hallux puede deberse a que es el punto de mayor intercambio calórico en la extremidad inferior con el medio ambiente y por tanto su temperatura tiende a disminuir, sin embargo, al presentarse el ejercicio que estimula el flujo y la temperatura del hallux comienza a aumentar progresivamente.

Como era de esperarse, el punto que registró un cambio mayor en temperatura fue la pantorrilla debido a que en esta zona se encuentran los músculos sometidos al ejercicio, y en consecuencia es donde se estimula el flujo sanguíneo.

La velocidad de flujo sanguíneo no presenta cambios importantes inmediatamente antes y después del ejercicio, pero un minuto después de este se eleva por arriba de los 6.5 cm/s , en este tiempo los sujetos se encontraban en una posición semisedente, por lo que el incremento en la velocidad pudo deberse al cambio de presión que se generó al liberar las extremidades del peso corporal del sujeto, y consecuentemente a la relajación de los músculos de dicha extremidad. A pesar de este incremento, después del minuto 4 se observa una caída de la velocidad de flujo sanguíneo por debajo de los 5.5 cm/s .

En el 33% de los sujetos el incremento en la velocidad de flujo fue evidente al finalizar el ejercicio, pero en el 67% restante no hubo cambios al inicio y finalización



de la actividad física. Como se observa en la figura 5.2, la temperatura de la piel presenta un retardo en su incremento de 1 min después de iniciado el ejercicio. Un detalle observado respecto a este cambio en la temperatura de la piel es que se presenta antes de que la velocidad de flujo sanguíneo tenga un incremento.

Debido a que las estructuras musculares en donde se tomaron las mediciones participan en el ejercicio de manera importante el incremento de temperatura pudiera deberse más por el calor generado en la actividad muscular que el proveniente del flujo sanguíneo.

Para corroborar si este incremento es debido a la actividad muscular más que al aumento en el flujo sanguíneo, se sugiere realizar pruebas con el sujeto en posición semisedente y pruebas con el sujeto de pie antes y después del ejercicio para comparar estos resultados.



CAPÍTULO 6. CONCLUSIONES Y PERSPECTIVAS

En la presente tesis se diseñaron e implementaron dos sistemas de medición de la temperatura de la piel, el primero para trabajo de laboratorio basado en el uso de un módulo de adquisición de datos y cuyos valores se muestran a través de una interfaz gráfica en una PC, y el segundo para trabajo en campo basado en el microcontrolador PIC18F4550.

Para garantizar el contacto entre los sensores de temperatura y la piel, se diseñó y construyó una banda con el soporte del sensor para su sujeción a la extremidad.

Se plantearon dos protocolos experimentales, uno basado en un cambio postural de la extremidad inferior y otro en base a una actividad física, empleándose en el primer protocolo el sistema para laboratorio y en el segundo el sistema para trabajo en campo. En ambos protocolos la temperatura de la piel y la velocidad de flujo sanguíneo fueron medidas en la extremidad inferior derecha en un grupo de sujetos de prueba.

De acuerdo a los resultados obtenidos en el primer protocolo experimental, se observó una relación directamente proporcional entre la temperatura de la piel y la velocidad de flujo sanguíneo en el 57% de los sujetos participantes.

Por su parte los resultados del segundo protocolo no permitieron establecer una relación directa entre la temperatura de la piel y la velocidad de flujo sanguíneo debido a que el primer parámetro se vio afectado por el calor generado en la actividad muscular.

Como perspectivas se plantean realizar más pruebas similares que refuercen los resultados obtenidos en los dos protocolos documentados, así como posteriormente efectuar mediciones en sujetos en sillas de ruedas.



REFERENCIAS

- [1] W. F. Ganong, “Fisiología Médica”, Mc Graw Hill, 23^a, Edición en Español, 2010, pág. 282-285, 518-522, 536-544, 580-581.
- [2] G. J. Tortora, B. Derrickson, “Principios de anatomía y Fisiología”, Edit. Médica Panamericana, 11^a. Edición, pág.671, 750-752,982.
- [3] A. C. Guyton, J. E. Hall, “Tratado de Fisiología Médica”, Editorial Elsevier, 11^a Edición, 2006, pág.164-170,889-900, 914.
- [4] I. D. Swain and L. J. Grant, “Methods Of Measuring Skin Blood Flow”, Phys. Med. Biol., 1989, Vol. 34, No 2, 151-175.
- [5] A.A. Sagaidachnyi et al, “Correlation of skin temperature and blood flow oscillations”, Saratov Fall Meeting 2011, Proc. of SPIE Vol. 8337, doi: 10.1117/12.925146
- [6] K. Nagata et al, “Heat Transfer Analysis for Peripheral Blood Flow Measurement System”, Review of Scientific Instruments 80, 064902 (2009), doi: 10.1063/1.3155458
- [7]http://mazinger.sisib.uchile.cl/repositorio/lb/ciencias_quimicas_y_farmaceuticas/steinera/parte04/02q.html
- [8] C. D. Haley et al., “Novel, high-amplitude blood-flow oscillations in vasodilating human skin”, Journal of Thermal Biology 29 (2004) 717–723, doi:10.1016/j.jtherbio.2004.08.045
- [9] A. L. Nilsson, “Blood Flow, Temperature, and Heat Loss of Skin Exposed to local Radiative and Convective Cooling”, National Swedish Board for Technical Development, 83-3907, Vol. 88, No. 5 Mayo de 1987.
- [10] N. Charkoudian, “Skin Blood Flow in Adult Human Thermoregulation: How It Works, When It Does Not, and Why”, Mayo Clin Proc. 2003; 78:603-612.
- [11] K. Eugenijus, “Biomedical Signals and Sensors I: Linking Physiological Phenomena and Biosignals”, Ed. Springer, 2012, pág. 215-218.
- [12] R. Tabrizchi, “Methods of blood flow measurement in the arterial circulatory system”, Journal of Pharmacological and Toxicological Methods 44 (2000) 375-384.
- [13] A. A. Sagaidachnyi et al, “Determination of the amplitude and phase relationships between oscillations in skin temperature and photoplethysmography-measured blood



flow in fingertips”, *Physiol. Meas.* 35 (2014) 153–166, doi:10.1088/0967-3334/35/2/153

[14] M. Davey et al, “The contribution of blood flow to the skin temperature responses during a cold sensitivity test”, *Eur J Appl Physiol* (2013) 113:2411–2417, DOI 10.1007/s00421-013-2678-8

[15] A. A. Sagaidachnyi et al, “Skin blood flow as the first time derivative of the temperature: spectral approach to the blood flow estimation in hands”, *Proc. of SPIE Vol. 9031 903108-1*, doi: 10.1117/12.2053110

[16] A.A. Sagaidachnyi et al, “Restoration of finger blood flow oscillations by means of thermal imaging”, 11th International Conference on Quantitative InfraRed Thermography, 11-14 June 2012, Naples Italy

[17] K. Demachi, “The influence of internal and skin temperatures on active cutaneous vasodilation under different levels of exercise and ambient temperatures in humans”, *Int J Biometeorol* (2013) 57:589–596, DOI 10.1007/s00484-012-0586-y

[18] <http://datasheet.octopart.com/10K3A1B-Betatherm-datasheet-529803.pdf>.

[19] http://www.academia.edu/7823988/INFORME_TERMISTORES

[20] www.ni.com/white-paper/7112/es/pdf

[21] R. Quintero, “Método para linealizar la salida de un sensor”, *Ingeniería*, 2003-00-00 vol: 8 nro: 1 pág. 82-85.

[22] C.A. Henao y E.D. Cardona, “Manejo de una memoria SD/MMC con un PIC16F87x”, *Scientia et Technica Año XVI, No 44, Abril de 2010. Universidad Tecnológica de Pereira. ISSN 0122-1701.*

[23] <http://www.ucontrol.com.ar/forosmf/programacion-en-c/manejo-de-memoria-sd-con-ccs-libreria-nivel-hardware/165/>

[24] JHD-63512864E Datasheet, JHD Electronics®

[25] PIC18F4550 Datasheet, Microchip®

[26] D. Formenti, et al.; “Thermal Imaging of Exercise-Associated Skin Temperature Changes in Trained and Untrained Female Subjects”; *Annals of Biomedical Engineering*, Vol. 41, No. 4, April 2013 (©2012) pp. 863-871; DOI: 10.1007/s10439-012-0718-x



APÉNDICE A. Resultados de la Caracterización de los Sensores

Temperatura [°C]	Sensor 1 [V]	Sensor 2 [V]	Sensor 3 [V]	Sensor 4 [V]	Sensor 5 [V]
20	1.335	1.352	1.522	1.347	1.337
20.5	1.385	1.400	1.568	1.398	1.390
21	1.437	1.452	1.617	1.447	1.437
21.5	1.485	1.500	1.667	1.495	1.490
22	1.533	1.553	1.718	1.548	1.540
22.5	1.590	1.603	1.772	1.600	1.593
23	1.642	1.657	1.823	1.652	1.643
23.5	1.690	1.710	1.873	1.703	1.697
24	1.740	1.758	1.923	1.755	1.748
24.5	1.795	1.815	1.975	1.807	1.798
25	1.850	1.865	2.027	1.860	1.855
25.5	1.900	1.913	2.080	1.913	1.905
26	1.950	1.972	2.133	1.963	1.960
26.5	2.008	2.027	2.188	2.017	2.013
27	2.060	2.080	2.240	2.073	2.065
27.5	2.113	2.133	2.295	2.125	2.118
28	2.168	2.190	2.347	2.178	2.172
28.5	2.223	2.243	2.398	2.232	2.227
29	2.277	2.297	2.455	2.285	2.280
29.5	2.332	2.352	2.508	2.340	2.337
30	2.387	2.405	2.562	2.393	2.393
30.5	2.445	2.465	2.618	2.450	2.447
31	2.498	2.518	2.673	2.503	2.503
31.5	2.553	2.575	2.727	2.560	2.557
32	2.607	2.625	2.780	2.617	2.613
32.5	2.663	2.687	2.838	2.672	2.667
33	2.722	2.738	2.890	2.725	2.725
33.5	2.778	2.798	2.948	2.780	2.780
34	2.828	2.853	3.000	2.837	2.837
34.5	2.887	2.907	3.058	2.893	2.893
35	2.940	2.965	3.115	2.948	2.947
35.5	3.002	3.025	3.170	3.005	3.005
36	3.055	3.077	3.225	3.062	3.063
36.5	3.112	3.138	3.280	3.117	3.118
37	3.168	3.190	3.335	3.173	3.175
37.5	3.227	3.250	3.392	3.230	3.233
38	3.278	3.302	3.443	3.282	3.285
38.5	3.340	3.365	3.505	3.340	3.345
39	3.393	3.417	3.558	3.393	3.400
39.5	3.448	3.473	3.615	3.450	3.455
40	3.502	3.525	3.667	3.505	3.510



APÉNDICE B. Librerías para la Memoria SD

//

/// (C) Copyright 2011 www.micros-designs.com.ar ///

/// Este código puede ser usado, modificado y distribuido libremente ///

/// sin eliminar esta cabecera y sin garantía de ningún tipo. ///

/// Esta librería encontrar, crear y/o escribir datos en un archive de

///texto por medio del protocolo FAT16

//

```
#ifndef __FAT_H
```

```
    #define __FAT_H
```

```
#if defined(__PCH__)
```

```
    #include "SDCardSPI.c"
```

```
#else
```

```
    #include "SDCardSPI.h"
```

```
#endif
```

```
#include "GenericTypeDefs.h"
```

```
#include <string.h>
```

```
// #define FAT16_DEBUG
```

```
// #define FAT16_DEBUG_FAT_FindDirectory
```

```
// #define FAT16_DEBUG_FAT_FindFile
```

```
// #define FAT16_DEBUG_FAT_FindFreeEnt
```

```
// #define FAT16_DEBUG_FAT_CreateDirectory
```

```
// #define FAT16_DEBUG_FAT_FindClusFree
```

```
// #define FAT16_DEBUG_FAT_CreateFile
```

```
// #define FAT16_DEBUG_FAT_FindFileDir
```

```
// #define FAT16_DEBUG_FAT_OpenAddFile
```

```
// #define FAT16_DEBUG_FAT_SearchOrderCluster
```

```
#if defined(FAT16_DEBUG)
```

```
    #if defined(__18CXX) || defined (__C30__) || defined (__PIC32MX__)
```

```
        #include <usart.h>
```

```
        #include <stdio.h>
```

```
    #endif
```



```
#endif
```

```
#define FAT_USE_DIRECTORY  
#define FAT_USE_FIND_DIRECTORY  
#define FAT_USE_READ_FILE
```

```
#define DIRECTORIO      0x10  
#define ARCHIVO        0x20  
#define DIRECTORIAIZ   0  
#define DIRECTORIO_RAIZ 0  
#define FAT_ERROR      0x00  
#define FAT_OK         0x01  
#define FAT_READ_COMPLETE 0xFF
```

```
/** \brief Inicia FAT, leyendo primer sector de memoria determinando parámetros del sistema de archivos.-
```

```
    \param Ninguno  
    \return 1 si la inicializacion es correcta, 0 caso contrario.-  
*/
```

```
BOOL FAT_init(void);
```

```
void FAT_UpdateDateTime(UINT8 Day,UINT8 Mth,UINT8 Year,UINT8 Hr,UINT8 Min,UINT8 Sec);
```

```
/** \brief Busca directorio/s retornando posicion para luego trabajar en él.-
```

```
    \param *ptr: Nombre de directorio/s.-  
    \param NClusP: N° de Cluster donde comenzar la busqueda.-  
  
    \return NClus: N° de cluster del último directorio en busqueda.-  
    \return 1 si es exitoso, 0 caso contrario.-  
*/
```

```
BOOL FAT_FindDirectory(UINT8 *ptr,UINT16 NClusP,UINT16 *NClus);
```

```
/** \brief Busca Archivo/Directorio dentro del cluster padre (NClusP).
```



```
\param Name: Nombre del archivo/directorio a buscar.-
\param NClusP: Numero del cluster donde buscar. Se inicia por DirectorioRaiz-> NClusP=0.-
\param Type: Archivo=0x20, Directorio=0x10.-
\return SecFileDir: Sector donde esta definido archivo.-
\return AddDir: Direccion adicional donde se encuentra definido archivo.-
\return NClus: Numero de cluster donde esta ubicado el archivo/directorio.-
\return SizeFile: Tamaño del archivo.-
\return 1 si la inicializacion es correcta, 0 caso contrario.-
*/

BOOL FAT_FindFile(UINT8 *Name,UINT16 NClusP);

BOOL FAT_FindFileDir(UINT8 *Name,UINT16 NClusP,UINT8 Type,UINT32 *SecFileDir,UINT16
*AddDir,UINT16 *NClus,UINT32 *SizeFile);
/** \brief Crea entrada en FAT1 y FAT2.

\param NClus: Numero de cluster donde crear entrada.-
\param EOC: Si es cluster fina se guarda 0xFFFF o sino se direcciona al proximo cluster.-

\return 1 si la inicializacion es correcta, 0 caso contrario.-
*/
BOOL FAT_CreateEntFAT(UINT16 NClus,UINT16 EOC);
/** \brief Busca la estructura de cluster de un archivo entregando el último usado.-

\param NClusInit: N° de Cluster donde comenzar la busqueda.-

\return NClusEnd: N° del último cluster que ocupa el archivo.-
\return 1 si es exitoso, 0 caso contrario.-
*/
BOOL FAT_SearchOrderCluster(UINT16 NClusInit,UINT16 *NClusEnd);

UINT16 FAT_ReadClusFAT(UINT16 NClus);
/** \brief Busca un cluster libre para crear archivo o directorio.
```



En este caso solo se piensa que el archivo no pesará mas de 1 Cluster xD

```
\param NClus: Numero de cluster donde crear entrada en la FATs.-
\return 1 si la inicializacion es correcta, 0 caso contrario.-
*/

UINT16 FAT_FindClusFree(void);
/** \brief Dentro del Directorio(NClus) a crear archivo/directorio busca N° Sector y posicion
dentro del sector para crear la entrada.-

\param NClus: Numero de cluster donde crear entrada en el Directorio.-
\return NSecFree: Numero de sector adicional del Sector Inicial.-
\return AddDir: Direccion adicional dentro del Sector.-
\return 1 si la inicializacion es correcta, 0 caso contrario.-
*/
BOOL FAT_FindFreeEnt(UINT16 NClus,UINT16 *ClusEnt, UINT8 *NSecFree,UINT16 *AddDir);
/** \brief Calcula el Numero de sector dentro de la memoria del Cluster.-

\param NClus: Numero de cluster.-
\return Numero del sector del cluster.-
*/
UINT32 FAT_SecClusN(UINT16 NClus);
/** \brief A partir de Fecha se forma variable para cargar fecha del archivo/directorio.

\param Day: Dia.-
\param Mth: Mes.-
\param Year: Año.-
\return Variable con el formato adecuado para cargar en FAT.-
*/
UINT16 FAT_formDate(UINT8 Day,UINT8 Mth,UINT8 Year);
/** \brief A partir de Hora se forma variable para cargar hora del archivo/directorio.

\param Hour: Hora.-
\param Min: Minuto.-
\param Sec: Segundo.-
\return Variable con el formato adecuado para cargar en FAT.-
```



*/

```
UINT16 FAT_formTime(UINT8 Hour,UINT8 Min,UINT8 Sec);
```

```
/** \brief Crea un directorio. Busca un cluster libre en la FAT, crea la definicion  
y crea la entrada en la FAT.-
```

```
\param NameLong: Nombre largo del directorio.-
```

```
\param NameShort: Nombre corto del directorio.-
```

```
\param NClusP Numero de cluster padre donde se define directorio.-
```

```
\return 0 si ha ocurrido algun error y 1 si ha sido exitoso
```

*/

```
UINT16 FAT_CreateDirectory(UINT8 *NameLong,UINT8 *NameShort,UINT16 NClusP);
```

```
/** \brief Define directorio, solo crea la definicion del nuevo directorio en el directorio seleccionado xD  
(NClusP)
```

```
\param NameLong: Nombre largo del directorio.-
```

```
\param NameShort: Nombre corto del directorio.-
```

```
\param NClusP Numero de cluster padre donde se define directorio.-
```

```
\param NClus: Numero de cluster donde se guarda el directorio nuevo.-
```

```
\param NSecFree: Numero de sector en el cluster padre para crear definicion.-
```

```
\param AddDir: Direccion adicional dentro de NSecFree donde comenzar a crear definicion.-
```

```
\return 0 si ha ocurrido algun error y 1 si ha sido exitoso
```

*/

```
BOOL FAT_DefineDirectory(UINT8 *NameLong,UINT8 *NameShort,UINT16 NClusP,UINT16  
NClus,UINT8 NSecFree,UINT16 AddDir );
```

```
/** \brief Crea un archivo. Busca un cluster libre en la FAT, crea la definicion, crea el archivo  
y crea la entrada en la FAT.-
```

```
\param NameLong: Nombre largo del archivo.-
```

```
\param NameShort: Nombre corto del archivo.-
```

```
\param NClusP: Numero de cluster padre donde se define archivo.-
```

```
\param WriteString: Buffer del archivo a cargar en memoria.-
```

```
\return 0 si ha ocurrido algun error y 1 si ha sido exitoso
```

*/

```
BOOL FAT_CreateFile(UINT8 *NameLong,UINT8 *NameShort,UINT16 NClusP,UINT8 *WriteString);
```



```
/** \brief Define archivo, solo crea la definicion del nuevo archivo en el directorio seleccionado xD  
(NClusP)
```

```
\param NameLong: Nombre largo del archivo.-  
\param NameShort: Nombre corto del archivo.-  
\param NClusP Numero de cluster padre donde se define archivo.-  
\param NClus: Numero de cluster donde se guarda el archivo nuevo.-  
\param NSecFree: Numero de sector en el cluster padre para crear definicion.-  
\param AddDir: Direccion adicional dentro de NSecFree donde comenzar a crear definicion.-  
\param WriteString: Buffer del archivo a cargar en memoria.-
```

```
\return 0 si ha ocurrido algun error y 1 si ha sido exitoso
```

```
*/
```

```
BOOL FAT_DefineFile(UINT8 *NameLong,UINT8 *NameShort,UINT16 NClusP,UINT16 NClus,UINT8  
NSecFree,UINT16 AddDir,UINT8 *WriteString);
```

```
/** \brief Busca un archivo por nombre corto, determina cluster inicial,  
cluster final, calcula tamaño del archivo y mas el adicional  
determina si es necesario usar otro Cluster. Si es necesario  
busca un nuevo cluster libre para adicionar al archivo. Terminado  
el agregado de datos, actualiza estructura de cluster en FAT y  
actualiza tamaño del archivo en la definición.-
```

```
\param *NameShor: Nombre corto del archivo.-  
\param NClusP: Nº de Clsuter donde comenzar la busqueda.-  
\param WriteString: Datos a agregar al archivo.-
```

```
\return 1 si es exitoso, 0 caso contrario.-
```

```
*/
```

```
BOOL FAT_OpenAddFile(UINT8 *NameShort,UINT16 NClusP,UINT8 *WriteString);
```

```
/** \brief Carga nuevos datos al cluster final del archivo y en el nuevo cluster  
si es necesario. Además actualiza tamaño del archivo.-
```

```
\param SecClusEnd: Ultimo cluster del archivo.-  
\param SecClusNew: Nuevo cluster de ser necesario.-  
\param NSecAddWrite: Ultimo sector del cluster final a donde comenzar agregar datos.-  
\param AddDirWrite: Direccion adicional del último sector para comenzar agregar datos.-  
\param SecFileDir: Sector donde esta definido archivo.-
```



```
\param AddDirSec: Direccion adicional donde se encuentra definido archivo.-
\param Size: Nuevo tamaño a cargar.-
\param WriteString: Datos a cargar.-
\return 1 si es exitoso, 0 caso contrario.-
*/
BOOL FAT_WriteAddFile(UINT32 SecClusEnd,UINT32 SecClusNew,UINT16 NSecAddWrite,UINT16
AddDirWrite,
                UINT32 SecFileDir,UINT16 AddDirSec,UINT32 Size,UINT8 *WriteString);

BOOL FAT_OpenFile(UINT8 *NameShort,UINT16 NClusP);
UINT8 FAT_OpenFileAndRead(UINT8 *NameShort,UINT16 NClusP, UINT8 Type, UINT8 *Buffer);
BOOL FAT_GetFile(UINT16 NClus,UINT32 SizeFile,UINT32 *SizeRead );
/** \brief Carga buffer para crear entrada del archivo/directorio. Coloca nombre,
        fecha, hora, etc.

\param Name: Nombre corto del archivo.-
\param Type: Tipo, archivo o directorio.-
\param NClus: Numero de cluster donde estará el archivo/directorio.-
\param SizeFile: Tamaño del archivo.-
\param AddDir: Direccion adicional en el sector para crear archivo/directorio.-
\param BufferFAT: Buffer donde guardar los parametros.-

\return Ninguno.-
*/
void FAT_LoadBuffer(UINT8 *Name,UINT8 Type,UINT16 NClus,UINT32 SizeFile,UINT16
AddDir,UINT8 *BufferFAT);
/** \brief Carga nombre en el Buffer-

\param ptr: Nombre corto del archivo sin tratamiento.-
\param Type: Tipo, archivo o directorio.-
\param ptr2: Nombre del archivo sacando el . y colocando espacios adecuados.-
\param AddDir: Direccion adicional en el sector para crear archivo/directorio.-

\return Ninguno.-
*/
void FAT_LoadName(UINT8 *ptr,UINT8 Type,UINT8 *ptr2,UINT16 AddDir);
```



```
/** \brief Determina el CheckSum en un nombre largo. (Enlace entre nombre corto y largo)
    \param NameShor: Nombre corto.-

    \return CheckSum
*/
UINT8 FAT_CheckSum(UINT8 *NameShort);
/** \brief Carga Buffer con el nombre largo segun especificaciones de FAT.-

    \param Name: Nombre largo del archivo/directorio.-
    \param NameShort: Nombre corto del archivo/directorio.
    \param Type: Tipo, archivo o directorio.-
    \param AddDir: Direccion adicional en el sector para crear archivo/directorio.-
    \param BufferFAT: Buffer donde guardar los parametros.-

    \return AddDir: Direccion adicional despues de haber cargado el buffer.-
*/
void FAT_LongName(UINT8 *Name,UINT8 *NameShort,UINT8 NRow,UINT8 Type,UINT16
*AddDir,UINT8 *BufferFAT);
#endif

////////////////////////////////////
// Encabezado de la librería para manejar el puerto SPI por software.
// Programador: KARY
// Versión: 0.1 Revisión: 0.01
// Fecha: Agosto 2014.
////////////////////////////////////
// Incluimos las definiciones de tipos de datos usados.
#include "GenericTypeDefs.h"
// Definimos los pines utilizados para realizar la comunicación SPI.
#define SD_CS PIN_B2
#define SD_SCK PIN_B1
#define SD_MOSI PIN_C7
#define SD_MISO PIN_B0

// Configura e inicializa el puerto SPI.
```




```
#use
spi(MASTER,BAUD=1000000,SAMPLE_RISE,MSB_FIRST,DI=SD_MISO,DO=SD_MOSI,CLK=SD_
SCK,BITS=8,STREAM=SDCard)

// Inicializa el hardware SPI.
void InitHard_SDCard(void);
// Lee el dispositivo SPI.
UINT8 ReadMedia(void);
// Escribe en el dispositivo SPI.
void WriteMedia(UINT8 data);

////////////////////////////////////
////                                     ////
////   (C) Copyright 2011 www.micros-designs.com.ar   ////
//// Este código puede ser usado, modificado y distribuido libremente ////
//// sin eliminar esta cabecera y sin garantía de ningún tipo.   ////
// Encabezado de la librería para manejar la memoria SD a nivel hardware.
// Programa modificado para trabajar solo con CCS.
////                                     ////
////////////////////////////////////

#ifndef __SDCardSPI_H
#define __SDCardSPI_H

// Incluimos librería para el manejo del puerto SPI.
#include "SoftwareSPI.c"
// Incluimos librería para manejar diferentes tipos de datos.
#include "GenericTypeDefs.h"

// Realiza debugger mediante RS232.
// #define SDCARD_DEBUG

// Definición de comandos.
#define CMD0 0    // Resetea la SD Card.
#define CMD1 1    // Activa proceso de inicialización de SD Card.
#define CMD8 8
#define CMD9 9    // Lectura registro CSD.
```



```
#define CMD10 10 // Lectura registro CID.
#define CMD16 16 // Selecciona largo del bloque para lectura/escritura (1 a 512).
#define CMD17 17 // Lectura de un único bloque.
#define CMD24 24 // Escritura de un único bloque.
#define CMD41 41
#define CMD55 55
#define CMD58 58
#define CMD59 59 // Enciende/Apaga CRC.
#define CMD32 32 // Setea dirección del primer bloque a borrar.
#define CDM33 33 // Setea dirección del último bloque en un continuo rango a borrar.
#define CMD38 38 // Borra todos los sectores entre las direcciones establecidas.

#define BLOCK_SIZE 512 // Tamaño del bloque de lectura/escritura.

// Inicialización de la SD.
BOOL SDCard_init(void);
// Realiza la lectura de un bloque, toma como parámetro
// la dirección del sector y el buffer donde se almacenará
// la lectura. Retorna 0 si hay error, caso contrario retorna 1.
BOOL SDCard_read_block(UINT32 Address,UINT8 *Buffer);
// Escribe un bloque de la memoria SD, toma como parámetro
// la dirección del sector a escribir y el buffer a escribir
// en la memoria. Retorna 0 si hay error, caso contrario retorna 1.
BOOL SDCard_write_block(UINT32 Address,UINT8 *Buffer);
// Realiza la lectura del registro CSD.
// Retorna 0 si hay error, caso contrario retorna 1.
BOOL SDCard_read_CSD(void);
// Realiza la lectura del registro CID.
// Retorna 0 si hay error, caso contrario retorna 1.
BOOL SDCard_read_CID(void);
#endif
```



APÉNDICE C. Librería para el Manejo de la GLCD

```
//////////////////////////////////////////////////////////////////
///          GLCDblue.C          ///
///          ///
/// Esta librería contiene los drivers para el control de una pantalla GLCD ///
/// Este archivo fue modificado para funcionar con la GLCD      ///
/// JHD-63512864E de 128 por 64 pixels por Karina          ///
///          ///
//////////////////////////////////////////////////////////////////
///          ///
/// LCD Pin connections:          ///
/// (These can be changed as needed in the following defines).  ///
/// * 1: Chip Select 1 (CS1)      (C2)          ///
/// * 2: Chip Select 2 (CS2)      (C1)          ///
/// * 3: VSS (GND)                ///
/// * 4: VDD (+5V)                ///
/// * 5: CONTRASTE                ///
/// * 6: D/I (Data/Instrucction)  (E2)        ///
/// * 7: R/W (Read/Write)         (E1)        ///
/// * 8: E (Enable)               (E0)        ///
/// *9-16: BUS DE DATOS           (PUERTO D)   ///
/// *17: RST (Reset)              (C0)        ///
/// *18: VEE (-10V)              ///
/// *19: LEDA (LED anode)        ///
/// *20: LEDC (LED catodo)       ///
///          ///
//////////////////////////////////////////////////////////////////
///          ///
/// glcd_init(mode)              ///
/// * Must be called before any other function.          ///
/// - mode can be ON or OFF to turn the LCD on or off    ///
///          ///
/// glcd_pixel(x,y,color)        ///
/// * Sets the pixel to the given color.                ///
/// - color can be ON or OFF          ///
///          ///
```



```
//// glcd_line(x1,y1,x2,y2,color)          ////
//// * Draws a line from the first point to the second point  ////
//// with the given color.                    ////
//// - color can be ON or OFF                 ////
////                                         ////
//// glcd_rect(x1,y1,x2,y2,fill,color)       ////
//// * Draws a rectangle with upper left point (x1,y1) and lower ////
//// right point (x2,y2).                    ////
//// - fill can be YES or NO                 ////
//// - color can be ON or OFF                 ////
////                                         ////
//// glcd_bar(x1,y1,x2,y2,width,color)        ////
//// * Draws a bar (wide line) from the first point to the    ////
//// second point.                               ////
//// - width is the number of pixels wide     ////
//// - color is ON or OFF                     ////
////                                         ////
//// glcd_circle(x,y,radius,fill,color)       ////
//// * Draws a circle with center at (x,y)      ////
//// - fill can be YES or NO                 ////
//// - color can be ON or OFF                 ////
////                                         ////
//// glcd_text57(x,y,textptr,size,color)      ////
//// * Write the null terminated text pointed to by textptr with ////
//// the upper left coordinate of the first character at (x,y).////
//// Characters are 5 pixels wide and 7 pixels tall.          ////
//// - size is an integer that scales the size of the text   ////
//// - color is ON or OFF                                     ////
//// * Note - The given text is character wrapped. If this   ////
//// function is used on a different size display, then change ////
//// the GLCD_WIDTH define appropriately.                    ////
////                                         ////
//// glcd_fillScreen(color)                      ////
//// * Fills the entire LCD with the given color.            ////
//// - color can be ON or OFF                             ////
////                                         ////
```




```
#endif
```

```
#ifndef GLCD_E
```

```
#define GLCD_E PIN_E0 // Enable
```

```
#endif
```

```
#ifndef GLCD_RST
```

```
#define GLCD_RST PIN_C0 // Reset
```

```
#endif
```

```
BYTE glcd_readByte(BYTE chip);
```

```
void glcd_writeByte(BYTE chip, BYTE data);
```

```
void glcd_fillScreen(int1 color);
```

```
const BYTE TEXT[51][5] = {0x00, 0x00, 0x00, 0x00, 0x00, // SPACE
```

```
    0x00, 0x00, 0x5F, 0x00, 0x00, // !  
    0x00, 0x03, 0x00, 0x03, 0x00, // "  
    0x14, 0x3E, 0x14, 0x3E, 0x14, // #  
    0x24, 0x2A, 0x7F, 0x2A, 0x12, // $  
    0x43, 0x33, 0x08, 0x66, 0x61, // %  
    0x36, 0x49, 0x55, 0x22, 0x50, // &  
    0x00, 0x05, 0x03, 0x00, 0x00, // '  
    0x00, 0x1C, 0x22, 0x41, 0x00, // (  
    0x00, 0x41, 0x22, 0x1C, 0x00, // )  
    0x14, 0x08, 0x3E, 0x08, 0x14, // *  
    0x08, 0x08, 0x3E, 0x08, 0x08, // +  
    0x00, 0x50, 0x30, 0x00, 0x00, // ,  
    0x08, 0x08, 0x08, 0x08, 0x08, // -  
    0x00, 0x60, 0x60, 0x00, 0x00, // .  
    0x20, 0x10, 0x08, 0x04, 0x02, // /  
    0x3E, 0x51, 0x49, 0x45, 0x3E, // 0  
    0x04, 0x02, 0x7F, 0x00, 0x00, // 1  
    0x42, 0x61, 0x51, 0x49, 0x46, // 2  
    0x22, 0x41, 0x49, 0x49, 0x36, // 3  
    0x18, 0x14, 0x12, 0x7F, 0x10, // 4  
    0x27, 0x45, 0x45, 0x45, 0x39, // 5
```



```
0x3E, 0x49, 0x49, 0x49, 0x32, // 6
0x01, 0x01, 0x71, 0x09, 0x07, // 7
0x36, 0x49, 0x49, 0x49, 0x36, // 8
0x26, 0x49, 0x49, 0x49, 0x3E, // 9
0x00, 0x36, 0x36, 0x00, 0x00, // :
0x00, 0x56, 0x36, 0x00, 0x00, // ;
0x08, 0x14, 0x22, 0x41, 0x00, // <
0x14, 0x14, 0x14, 0x14, 0x14, // =
0x00, 0x41, 0x22, 0x14, 0x08, // >
0x02, 0x01, 0x51, 0x09, 0x06, // ?
0x3E, 0x41, 0x59, 0x55, 0x5E, // @
0x7E, 0x09, 0x09, 0x09, 0x7E, // A
0x7F, 0x49, 0x49, 0x49, 0x36, // B
0x3E, 0x41, 0x41, 0x41, 0x22, // C
0x7F, 0x41, 0x41, 0x41, 0x3E, // D
0x7F, 0x49, 0x49, 0x49, 0x41, // E
0x7F, 0x09, 0x09, 0x09, 0x01, // F
0x3E, 0x41, 0x41, 0x49, 0x3A, // G
0x7F, 0x08, 0x08, 0x08, 0x7F, // H
0x00, 0x41, 0x7F, 0x41, 0x00, // I
0x30, 0x40, 0x40, 0x40, 0x3F, // J
0x7F, 0x08, 0x14, 0x22, 0x41, // K
0x7F, 0x40, 0x40, 0x40, 0x40, // L
0x7F, 0x02, 0x0C, 0x02, 0x7F, // M
0x7F, 0x02, 0x04, 0x08, 0x7F, // N
0x3E, 0x41, 0x41, 0x41, 0x3E, // O
0x7F, 0x09, 0x09, 0x09, 0x06, // P
0x1E, 0x21, 0x21, 0x21, 0x5E, // Q
0x7F, 0x09, 0x09, 0x09, 0x76}; // R
```

```
const BYTE TEXT2[44][5]={0x26, 0x49, 0x49, 0x49, 0x32, // S
```

```
0x01, 0x01, 0x7F, 0x01, 0x01, // T
0x3F, 0x40, 0x40, 0x40, 0x3F, // U
0x1F, 0x20, 0x40, 0x20, 0x1F, // V
0x7F, 0x20, 0x10, 0x20, 0x7F, // W
0x41, 0x22, 0x1C, 0x22, 0x41, // X
```



0x07, 0x08, 0x70, 0x08, 0x07, // Y
0x61, 0x51, 0x49, 0x45, 0x43, // Z
0x00, 0x7F, 0x41, 0x00, 0x00, // [
0x02, 0x04, 0x08, 0x10, 0x20, // \
0x00, 0x00, 0x41, 0x7F, 0x00, //]
0x04, 0x02, 0x01, 0x02, 0x04, // ^
0x40, 0x40, 0x40, 0x40, 0x40, // _
0x00, 0x01, 0x02, 0x04, 0x00, // `
0x20, 0x54, 0x54, 0x54, 0x78, // a
0x7F, 0x44, 0x44, 0x44, 0x38, // b
0x38, 0x44, 0x44, 0x44, 0x44, // c
0x38, 0x44, 0x44, 0x44, 0x7F, // d
0x38, 0x54, 0x54, 0x54, 0x18, // e
0x04, 0x04, 0x7E, 0x05, 0x05, // f
0x08, 0x54, 0x54, 0x54, 0x3C, // g
0x7F, 0x08, 0x04, 0x04, 0x78, // h
0x00, 0x44, 0x7D, 0x40, 0x00, // i
0x20, 0x40, 0x44, 0x3D, 0x00, // j
0x7F, 0x10, 0x28, 0x44, 0x00, // k
0x00, 0x41, 0x7F, 0x40, 0x00, // l
0x7C, 0x04, 0x78, 0x04, 0x78, // m
0x7C, 0x08, 0x04, 0x04, 0x78, // n
0x38, 0x44, 0x44, 0x44, 0x38, // o
0x7C, 0x14, 0x14, 0x14, 0x08, // p
0x08, 0x14, 0x14, 0x14, 0x7C, // q
0x00, 0x7C, 0x08, 0x04, 0x04, // r
0x48, 0x54, 0x54, 0x54, 0x20, // s
0x04, 0x04, 0x3F, 0x44, 0x44, // t
0x3C, 0x40, 0x40, 0x20, 0x7C, // u
0x1C, 0x20, 0x40, 0x20, 0x1C, // v
0x3C, 0x40, 0x30, 0x40, 0x3C, // w
0x44, 0x28, 0x10, 0x28, 0x44, // x
0x0C, 0x50, 0x50, 0x50, 0x3C, // y
0x44, 0x64, 0x54, 0x4C, 0x44, // z
0x00, 0x08, 0x36, 0x41, 0x41, // {
0x00, 0x00, 0x7F, 0x00, 0x00, // |



```
0x41, 0x41, 0x36, 0x08, 0x00, // }  
0x02, 0x01, 0x02, 0x04, 0x02};// ~
```

```
// Purpose:   Initialize a graphic LCD. This must be called before any  
//           other glcd function is used.  
// Inputs:   The initialization mode  
//           OFF - Turns the LCD off  
//           ON  - Turns the LCD on  
// Date:    5/28/2003  
void glcd_init(int1 mode)  
{  
    // Initialize some pins  
    output_high(GLCD_RST);  
    output_low(GLCD_E);  
    output_low(GLCD_CS1);  
    output_low(GLCD_CS2);  
  
    output_low(GLCD_DI);           // Set for instruction  
    glcd_writeByte(GLCD_CS1, 0xC0); // Specify first RAM line at the top  
    glcd_writeByte(GLCD_CS2, 0xC0); // of the screen  
    glcd_writeByte(GLCD_CS1, 0x40); // Set the column address to 0  
    glcd_writeByte(GLCD_CS2, 0x40);  
    glcd_writeByte(GLCD_CS1, 0xB8); // Set the page address to 0  
    glcd_writeByte(GLCD_CS2, 0xB8);  
  
    if(mode == ON)  
    {  
        glcd_writeByte(GLCD_CS1, 0x3F); // Turn the display on  
        glcd_writeByte(GLCD_CS2, 0x3F);  
    }  
    else  
    {  
        glcd_writeByte(GLCD_CS1, 0x3E); // Turn the display off  
        glcd_writeByte(GLCD_CS2, 0x3E);  
    }  
}
```



```
glcd_fillScreen(OFF);          // Clear the display
}

// Purpose:   Turn a pixel on a graphic LCD on or off
// Inputs:    x - the x coordinate of the pixel
//            y - the y coordinate of the pixel
//            color - ON or OFF
// Output:    1 if coordinate out of range, 0 if in range
void glcd_pixel(int x, int y, int1 color)
{
  BYTE data;
  BYTE chip = GLCD_CS1; // Stores which chip to use on the LCD

  if(x > 63) // Check for first or second display area
  {
    x -= 64;
    chip = GLCD_CS2;
  }

  output_low(GLCD_DI);          // Set for instruction
  bit_clear(x,7);               // Clear the MSB. Part of an instruction code
  bit_set(x,6);                 // Set bit 6. Also part of an instruction code
  glcd_writeByte(chip, x);      // Set the horizontal address
  glcd_writeByte(chip, (y/8 & 0b10111111) | 0b10111000); // Set the vertical page address
  output_high(GLCD_DI);        // Set for data
  data = glcd_readByte(chip);

  if(color == ON)
    bit_set(data, y%8); // Turn the pixel on
  else // or
    bit_clear(data, y%8); // turn the pixel off
  output_low(GLCD_DI); // Set for instruction
  glcd_writeByte(chip, x); // Set the horizontal address
  output_high(GLCD_DI); // Set for data
}
```



```
glcd_writeByte(chip, data); // Write the pixel data
}

// Purpose: Draw a line on a graphic LCD using Bresenham's
// line drawing algorithm
// Inputs: (x1, y1) - the start coordinate
// (x2, y2) - the end coordinate
// color - ON or OFF
// Dependencies: glcd_pixel()
void glcd_line(int x1, int y1, int x2, int y2, int1 color)
{
    signed int x, y, addx, addy, dx, dy;
    signed long P;
    int i;
    dx = abs((signed int)(x2 - x1));
    dy = abs((signed int)(y2 - y1));
    x = x1;
    y = y1;

    if(x1 > x2)
        addx = -1;
    else
        addx = 1;
    if(y1 > y2)
        addy = -1;
    else
        addy = 1;

    if(dx >= dy)
    {
        P = 2*dy - dx;

        for(i=0; i<=dx; ++i)
        {
            glcd_pixel(x, y, color);
```



```
    if(P < 0)
    {
        P += 2*dy;
        x += addx;
    }
    else
    {
        P += 2*dy - 2*dx;
        x += addx;
        y += addy;
    }
}
else
{
    P = 2*dx - dy;

    for(i=0; i<=dy; ++i)
    {
        glcd_pixel(x, y, color);

        if(P < 0)
        {
            P += 2*dx;
            y += addy;
        }
        else
        {
            P += 2*dx - 2*dy;
            x += addx;
            y += addy;
        }
    }
}
}
```



```
// Purpose: Draw a rectangle on a graphic LCD
// Inputs: (x1, y1) - the start coordinate
//         (x2, y2) - the end coordinate
//         fill - YES or NO
//         color - ON or OFF
// Dependencies: glcd_pixel(), glcd_line()
void glcd_rect(int x1, int y1, int x2, int y2, int fill, int1 color)
{
    if(fill)
    {
        int y, ymax;           // Find the y min and max
        if(y1 < y2)
        {
            y = y1;
            ymax = y2;
        }
        else
        {
            y = y2;
            ymax = y1;
        }

        for(; y<=ymax; ++y)    // Draw lines to fill the rectangle
            glcd_line(x1, y, x2, y, color);
    }
    else
    {
        glcd_line(x1, y1, x2, y1, color); // Draw the 4 sides
        glcd_line(x1, y2, x2, y2, color);
        glcd_line(x1, y1, x1, y2, color);
        glcd_line(x2, y1, x2, y2, color);
    }
}

// Purpose: Draw a bar (wide line) on a graphic LCD
```



```
// Inputs:   (x1, y1) - the start coordinate
//           (x2, y2) - the end coordinate
//           width - The number of pixels wide
//           color - ON or OFF
void glcd_bar(int x1, int y1, int x2, int y2, int width, int1 color)
{
    signed int x, y, addx, addy, j;
    signed long P, dx, dy, c1, c2;
    int i;
    dx = abs((signed int)(x2 - x1));
    dy = abs((signed int)(y2 - y1));
    x = x1;
    y = y1;
    c1 = -dx*x1 - dy*y1;
    c2 = -dx*x2 - dy*y2;

    if(x1 > x2)
    {
        addx = -1;
        c1 = -dx*x2 - dy*y2;
        c2 = -dx*x1 - dy*y1;
    }
    else
        addx = 1;
    if(y1 > y2)
    {
        addy = -1;
        c1 = -dx*x2 - dy*y2;
        c2 = -dx*x1 - dy*y1;
    }
    else
        addy = 1;

    if(dx >= dy)
    {
        P = 2*dy - dx;
```



```
for(i=0; i<=dx; ++i)
{
  for(j=-(width/2); j<width/2+width%2; ++j)
  {
    if(dx*x+dy*(y+j)+c1 >= 0 && dx*x+dy*(y+j)+c2 <=0)
      glcd_pixel(x, y+j, color);
  }
  if(P < 0)
  {
    P += 2*dy;
    x += addx;
  }
  else
  {
    P += 2*dy - 2*dx;
    x += addx;
    y += addy;
  }
}
else
{
  P = 2*dx - dy;

  for(i=0; i<=dy; ++i)
  {
    if(P < 0)
    {
      P += 2*dx;
      y += addy;
    }
    else
    {
      P += 2*dx - 2*dy;
      x += addx;
    }
  }
}
```



```
        y += addy;
    }
    for(j=-(width/2); j<width/2+width%2; ++j)
    {
        if(dx*x+dy*(y+j)+c1 >= 0 && dx*x+dy*(y+j)+c2 <=0)
            glcd_pixel(x+j, y, color);
    }
}
}
```

```
// Purpose:    Draw a circle on a graphic LCD
// Inputs:    (x,y) - the center of the circle
//            radius - the radius of the circle
//            fill - YES or NO
//            color - ON or OFF
void glcd_circle(int x, int y, int radius, int1 fill, int1 color)
{
    signed int a, b, P;
    a = 0;
    b = radius;
    P = 1 - radius;

    do
    {
        if(fill)
        {
            glcd_line(x-a, y+b, x+a, y+b, color);
            glcd_line(x-a, y-b, x+a, y-b, color);
            glcd_line(x-b, y+a, x+b, y+a, color);
            glcd_line(x-b, y-a, x+b, y-a, color);
        }
        else
        {
            glcd_pixel(a+x, b+y, color);
            glcd_pixel(b+x, a+y, color);
        }
    }
}
```




```
    glcd_pixel(x-a, b+y, color);
    glcd_pixel(x-b, a+y, color);
    glcd_pixel(b+x, y-a, color);
    glcd_pixel(a+x, y-b, color);
    glcd_pixel(x-a, y-b, color);
    glcd_pixel(x-b, y-a, color);
}

if(P < 0)
    P+= 3 + 2*a++;
else
    P+= 5 + 2*(a++ - b--);
} while(a <= b);
}

// Purpose:    Write text on a graphic LCD
// Inputs:    (x,y) - The upper left coordinate of the first letter
//            textptr - A pointer to an array of text to display
//            size - The size of the text: 1 = 5x7, 2 = 10x14, ...
//            color - ON or OFF
void glcd_text57(int x, int y, char* textptr, int size, int1 color)
{
    int i, j, k, l, m;           // Loop counters
    BYTE pixelData[5];         // Stores character data

    for(i=0; textptr[i] != '\0'; ++i, ++x) // Loop through the passed string
    {
        if(textptr[i] < 'S') // Checks if the letter is in the first text array
            memcpy(pixelData, TEXT[textptr[i]-' '], 5);
        else if(textptr[i] <= '~') // Check if the letter is in the second array
            memcpy(pixelData, TEXT2[textptr[i]-'S'], 5);
        else
            memcpy(pixelData, TEXT[0], 5); // Default to space

        if(x+5*size >= GLCD_WIDTH) // Performs character wrapping
        {
```



```
x = 0;           // Set x at far left position
y += 7*size + 1; // Set y at next position down
}
for(j=0; j<5; ++j, x+=size) // Loop through character byte data
{
    for(k=0; k<7*size; ++k) // Loop through the vertical pixels
    {
        if(bit_test(pixelData[j], k)) // Check if the pixel should be set
        {
            for(l=0; l<size; ++l) // The next two loops change the
            { // character's size
                for(m=0; m<size; ++m)
                {
                    glcd_pixel(x+m, y+k*size+l, color); // Draws the pixel
                }
            }
        }
    }
}
}
```

```
// Purpose: Fill the LCD screen with the passed in color.
// Works much faster than drawing a rectangle to fill the screen.
// Inputs: ON - turn all the pixels on
// OFF - turn all the pixels off
// Dependencies: glcd_writeByte()
void glcd_fillScreen(int1 color)
{
    int i, j;

    // Loop through the vertical pages
    for(i = 0; i < 8; ++i)
    {
        output_low(GLCD_DI); // Set for instruction
```



```
glcd_writeByte(GLCD_CS1, 0b01000000); // Set horizontal address to 0
glcd_writeByte(GLCD_CS2, 0b01000000);
glcd_writeByte(GLCD_CS1, i | 0b10111000); // Set page address
glcd_writeByte(GLCD_CS2, i | 0b10111000);
output_high(GLCD_DI); // Set for data

// Loop through the horizontal sections
for(j = 0; j < 64; ++j)
{
    glcd_writeByte(GLCD_CS1, 0xFF*color); // Turn pixels on or off
    glcd_writeByte(GLCD_CS2, 0xFF*color); // Turn pixels on or off
}
}
}

// Purpose: Write a byte of data to the specified chip
// Inputs: chipSelect - which chip to write the data to
// data - the byte of data to write

void glcd_writeByte(char chip, BYTE data)
{
    if(chip == GLCD_CS1) // Choose which chip to write to
        output_high(GLCD_CS1);
    else
        output_high(GLCD_CS2);

    output_low(GLCD_RW); // Set for writing
    output_d(data); // Put the data on the port
    output_high(GLCD_E); // Pulse the enable pin
    delay_us(2);
    output_low(GLCD_E);

    output_low(GLCD_CS1); // Reset the chip select lines
    output_low(GLCD_CS2);
}
```



```
// Purpose:   Reads a byte of data from the specified chip
// Outputs:   A byte of data read from the chip
BYTE glcd_readByte(BYTE chip)
{
    BYTE data;           // Stores the data read from the LCD
    if(chip == GLCD_CS1) // Choose which chip to read from
        output_high(GLCD_CS1);
    else
        output_high(GLCD_CS2);

    input_d();           // Set port d to input
    output_high(GLCD_RW); // Set for reading
    output_high(GLCD_E);  // Pulse the enable pin
    delay_us(2);
    output_low(GLCD_E);
    delay_us(2);
    output_high(GLCD_E);  // Pulse the enable pin
    delay_us(2);
    data = input_d();     // Get the data from the display's output register
    output_low(GLCD_E);

    output_low(GLCD_CS1); // Reset the chip select lines
    output_low(GLCD_CS2);
    return data;         // Return the read data
}

#endif
```

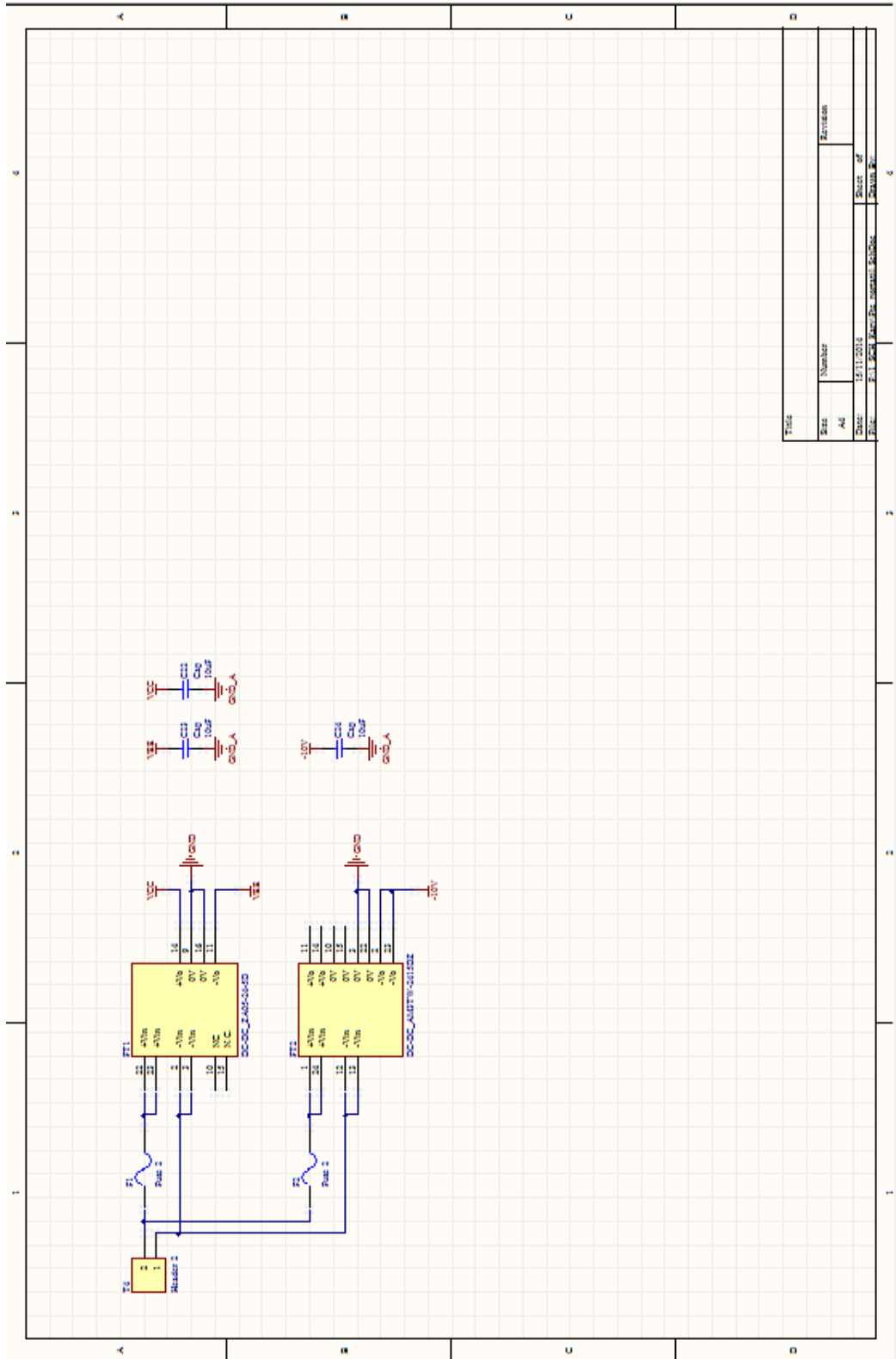



Figura. B.2.Circuito esquemático de la fuente de alimentación del módulo.



APÉNDICE E. Código de Principal del Módulo Portátil

//

//Tercera versión del programa final para el sistema portátil

//Programador: Karina

//Fecha: Noviembre 2014

//

//

//El programa maneja:

//Memoria SD, GLCD, ADC, RTC

//

//Al inicio despliega el logo y busca que la memoria SD esté insertada

//Si la encuentra muestra un mensaje de "Memoria encontrada" y lee el ADC

//realiza la conversión a temperatura y manda el valor a la SD y GLCD

//siendo una lectura->escritura->despliegue...

//a su vez, toma la lectura del tiempo, para establecer en que momento se

//adquirió la temperatura

//

//En el caso de no encontrar la memoria SD insertada, despliega un mensaje de error

//y solo despliega datos a la GLCD

//

//

//Para esta tercera versión se modificó:

// -El despliegue de los datos en la gráfica

// -La posibilidad de escribir hasta 10 archivos .TXT en la memoria SD

// NOTA:

// Para definir el nombre del archivo TXT se almacena en la memoria eeprom un número,

// el cual se lee cada vez que corre el programa, este valor va del 1 al 10,

// por lo que si en algún momento se borran los archivos generados en la SD

// creará un archivo con el nombre del último número que almacenó

// por ejem. si se crearon 5 archivos, y se borran al generar los nuevos

// iniciará en el número 6

//

#include<18F4550.h>

#device adc=10

#fuses HS,NOWDT,NOPROTECT,NOLVP

#use delay(clock=42000000)

//reloj 42Mhz



```
#use rs232(baud=9600,parity=N,xmit=PIN_C6,bits=8) //Línea para el funcionamiento de la librería de la SD
```

```
//LIBRERÍAS
```

```
#include <GLCDblue.c>          //para la GLCD azul  
#include "FAT16.c"            //para la memoria SD.  
#include <string.h>           //para la conversión a string  
#include <DS1307mio2.c>       //libreria RTC
```

```
//DEFINICIÓN DE LOS LEDS PARA LA SD
```

```
#define LED_SD_OK    PIN_B3  
#define LED_SD_ERROR PIN_B4  
// Macros de E/S.  
#define LED_SD_OK_ON  output_high(LED_SD_OK);  
#define LED_SD_OK_OFF output_low(LED_SD_OK);  
#define LED_SD_ERROR_ON output_high(LED_SD_ERROR);  
#define LED_SD_ERROR_OFF output_low(LED_SD_ERROR);
```

```
//VARIABLES GLOBALES
```

```
float term1,term2,term3,term4;    //variables temperatura
```

```
int Xpix;          //variables gráfica  
int Ypix1,Ypix2,Ypix3,Ypix4;  
int AYpix1[87]={0};  
int AYpix2[87]={0};  
int AYpix3[87]={0};  
int AYpix4[87]={0};  
int i,n,rtci;
```

```
char temp[10];      //variables SD
```

```
char Fin[5]="\r\n";
```

```
char tab[3]="\t";
```

```
    //Nombre largo y corto del archivo
```

```
char NombreCortoArchivo[13];
```

```
char NombreLargoArchivo[16];
```

```
char encabezado[40]="Tiempo\tTerm1\tTerm2\tTerm3\tTerm4\r\n";
```



```
int1 NOSD;
int A; //número a guardar

char ho[40]={0.0}; //variables RTC
char mi[10];
char se[10];
BYTE hrs;
BYTE min;
BYTE sec;

//MENSAJES GLCD
char ok[]="Memoria Encontrada!";
char error[]="Error Memoria!";
char s1[15]={0.0};
char s2[15]={0.0};
char s3[15]={0.0};
char s4[15]={0.0};

//-----
//-----IMAGEN DE INICIO-----
const int8 logo[64][16] = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,
0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xFF, 0xFF, 0xC0, 0x00, 0x78,
0xCC, 0x6C, 0x1B, 0xF1, 0xE3, 0xF0, 0xE3, 0x06, 0x00, 0x00, 0x0F, 0xFF,
0xFF, 0xE0, 0x00, 0xFC, 0xCE, 0x66, 0x33, 0xF3, 0xF3, 0xF0, 0xE1, 0x8C,
0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xF0, 0x01, 0x8C, 0xCE, 0x66, 0x33, 0x03,
0x00, 0xC1, 0xB1, 0x8C, 0x00, 0x00, 0x0F, 0xFF, 0xFF, 0xF0, 0x01, 0x80,
0xCF, 0x67, 0x73, 0xF3, 0xE0, 0xC1, 0xB1, 0xDC, 0x00, 0x00, 0x0F, 0xF0,
0x3F, 0xF0, 0x01, 0x80, 0xCD, 0xE3, 0x63, 0xF0, 0xF0, 0xC3, 0x18, 0xD8,
0x00, 0x00, 0x0F, 0x00, 0x01, 0xF0, 0x01, 0x8C, 0xCC, 0xE3, 0x63, 0x00,
0x30, 0xC3, 0xF8, 0xD8, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x70, 0x00, 0xFC,
0xCC, 0xE1, 0xC3, 0xF3, 0xF0, 0xC3, 0xF8, 0x70, 0x00, 0x00, 0x06, 0x60,
```



0x00, 0x70, 0x00, 0x78, 0xCC, 0x61, 0xC3, 0xF1, 0xE0, 0xC6, 0x0C, 0x70,
0x00, 0x00, 0x07, 0xF0, 0x0F, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x7C, 0x0F, 0xF8, 0x0F, 0xF0, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFE, 0x3F, 0xF8,
0x1F, 0xF8, 0x1E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0xFF, 0x7F, 0xF8, 0x1F, 0xFE, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x03, 0xFF, 0xFF, 0xC8, 0x13, 0xFF, 0xFF, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xF9, 0xFC, 0xCF,
0xF3, 0xFF, 0xFF, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00,
0x07, 0xF1, 0xF8, 0x7F, 0xFF, 0x1F, 0x9F, 0xC1, 0xE6, 0x00, 0x18, 0x00,
0x40, 0x80, 0x18, 0x00, 0x07, 0xF1, 0xF0, 0x7F, 0xFE, 0x0F, 0xCF, 0xE1,
0xB0, 0x00, 0x18, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x0F, 0xE3, 0xF0, 0x3F,
0xFE, 0x07, 0xC7, 0xE1, 0xB6, 0x71, 0x99, 0x8D, 0xFE, 0xE7, 0x99, 0x9C,
0x0F, 0xC3, 0xF8, 0x04, 0x30, 0x0F, 0xE3, 0xF1, 0xE6, 0xDB, 0x5B, 0x5A,
0xD9, 0xB6, 0xDB, 0x46, 0x0F, 0xC1, 0xCC, 0x04, 0x20, 0x1B, 0xE3, 0xF1,
0xB6, 0xDB, 0xDB, 0xD8, 0xD9, 0xB6, 0xDB, 0x0E, 0x1F, 0x80, 0xDE, 0x0C,
0x20, 0x33, 0xC1, 0xF1, 0xB6, 0xDB, 0x1B, 0x18, 0xD9, 0xB6, 0xDB, 0x16,
0x1F, 0x80, 0x7E, 0x3C, 0x38, 0x3F, 0x01, 0xF9, 0xB6, 0xDB, 0x5B, 0x5A,
0xD9, 0xB6, 0xDB, 0x56, 0x1F, 0x00, 0x3C, 0x7C, 0x3E, 0x3C, 0x01, 0xF9,
0xE6, 0x71, 0x99, 0x8C, 0x78, 0xE6, 0xD9, 0x9E, 0x3F, 0x00, 0x7E, 0xF0,
0x1F, 0x3C, 0x00, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x3F, 0x00, 0x7B, 0xE0, 0x07, 0xFE, 0x00, 0xF8, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x3F, 0x00, 0x70, 0xC0, 0x03, 0x0E, 0x00, 0xFC,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3E, 0x00, 0xF0, 0x00,
0x00, 0x0F, 0x00, 0x7C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x3E, 0x3F, 0xF0, 0x00, 0x00, 0x0F, 0x00, 0x7C, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x3E, 0x3F, 0xF8, 0x00, 0x00, 0x0F, 0xFC, 0x7C,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3E, 0x3C, 0xFF, 0x00,
0x00, 0xFF, 0xFC, 0x7C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x3F, 0x7C, 0xE3, 0x00, 0x01, 0xFF, 0x3C, 0x7C, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x3F, 0xFF, 0xE3, 0x00, 0x01, 0xCF, 0x3C, 0xFC,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xFF, 0xC3, 0x00,
0x01, 0xC7, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x07, 0xFC, 0x03, 0x80, 0x01, 0xC0, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x3C, 0x01, 0x80, 0x03, 0x80, 0x3E, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3C, 0x01, 0xC0,



0x03, 0x80, 0x3C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x3E, 0x19, 0xC0, 0x03, 0x00, 0x7C, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x1E, 0x3F, 0x80, 0x01, 0xB8, 0x78, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xFF, 0x0F,
0xE0, 0xFE, 0x78, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x1F, 0xFE, 0x0F, 0xF0, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x9F, 0x1F, 0xF1, 0xF3, 0xF8, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xF0,
0x1B, 0xF3, 0xFC, 0x00, 0x04, 0x01, 0x00, 0x00, 0x40, 0x10, 0x00, 0x80,
0x01, 0xFF, 0xE7, 0xF0, 0x0F, 0xFF, 0xFF, 0x00, 0x04, 0x01, 0x00, 0x00,
0x40, 0x00, 0x01, 0x80, 0x01, 0xE3, 0xC1, 0xF8, 0x3F, 0xCF, 0xFF, 0x00,
0x04, 0x39, 0x46, 0x5E, 0xEC, 0xB3, 0x02, 0x80, 0x01, 0xF1, 0x80, 0xFC,
0x3F, 0x07, 0xCF, 0x80, 0x04, 0x05, 0xA9, 0x61, 0x52, 0xD4, 0x82, 0x80,
0x01, 0xF0, 0x00, 0x7C, 0x3E, 0x03, 0x8F, 0x80, 0x04, 0x1D, 0x29, 0x47,
0x52, 0x94, 0x84, 0x80, 0x01, 0xF8, 0x00, 0xCC, 0x32, 0x00, 0x1F, 0x80,
0x04, 0x25, 0x29, 0x49, 0x52, 0x94, 0x87, 0xC0, 0x00, 0xFC, 0x00, 0xC8,
0x33, 0x00, 0x1F, 0x00, 0x04, 0x25, 0xA9, 0x49, 0x52, 0x94, 0x80, 0x80,
0x00, 0xFE, 0x01, 0xF8, 0x1F, 0x00, 0x3F, 0x00, 0x07, 0x9D, 0x46, 0x47,
0x6C, 0x93, 0x00, 0x80, 0x00, 0x7F, 0x01, 0xFC, 0x3F, 0x00, 0x7E, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x81, 0xFF,
0xFF, 0x01, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x1F, 0xC0, 0xFF, 0xFF, 0x03, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF0, 0x3F, 0xFC, 0x07, 0xF0, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFE, 0x18,
0x30, 0x1F, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x03, 0xFF, 0xF0, 0x18, 0xFF, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xFF, 0xF0, 0x1F, 0xFF, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xF0,
0x1F, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x3F, 0xE0, 0x1F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xC0, 0x0F, 0xE0, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x80,
0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00

};



```
//-----  
//-----IMAGEN DEL GRÁFICO-----  
const int8 graf[64][16] = {  
    0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x5C, 0x40, 0x1C, 0xCC, 0xC3, 0x76, 0x6E, 0xA1,  
    0x4C, 0x1B, 0x86, 0x43, 0x29, 0x13, 0x80, 0x00, 0xB0, 0xE0, 0x11, 0x55,  
    0x42, 0x55, 0x4A, 0xC1, 0x54, 0x2A, 0x82, 0xA5, 0x2A, 0xAA, 0x00, 0x00,  
    0x51, 0xF0, 0x11, 0x55, 0x41, 0x65, 0x2A, 0x81, 0x54, 0x2B, 0x04, 0xA5,  
    0x3A, 0x92, 0x00, 0x00, 0x10, 0x40, 0x1C, 0xCC, 0xC3, 0x75, 0x6E, 0x80,  
    0x8C, 0x1B, 0x86, 0x43, 0x09, 0x03, 0x80, 0x00, 0x1C, 0x40, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x03, 0xC3, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,  
    0xFF, 0xFF, 0xFF, 0xC0, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0xE4, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x8C, 0x40, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0xE4, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x24, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xE4, 0x40, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x03, 0xC3, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,  
    0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0x00, 0x40, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xEE, 0x40, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x82, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```




```
0x00, 0x00, 0x00, 0x00, 0x22, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xE2, 0x40, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08,  
0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x0C, 0x00, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,  
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0C,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xB5, 0x76, 0x4A, 0x6C, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x55, 0x54,  
0xEC, 0xA8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x01, 0x15, 0x62, 0x48, 0xA4, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x17, 0x76, 0x68, 0x6C, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00
```

```
};
```

```
//-----
```

```
//Función para el despliegue del logo de inicio
```

```
void glcd_logo()
```

```
{
```

```
    char i,j;
```

```
    signed char k;
```

```
    for( i = 0 ; i < 64 ; i ++ )
```

```
    {
```

```
        for( j = 0 ; j < 16 ; j ++)
```

```
        {
```

```
            for(k=7;k>-1;k--)
```

```
            {
```

```
                if( bit_test(logo[i][j] ,7-k )
```

```
                    glcd_pixel( j*8+k,i, ON );
```

```
            }
```

```
        }
```

```
    }
```



```
}
```

```
//Función para el despliegue de la gráfica
```

```
void glcd_grafica()
```

```
{
```

```
    char i,j;
```

```
    signed char k;
```

```
    for( i = 0 ; i < 64 ; i ++ )
```

```
    {
```

```
        for( j = 0 ; j < 16 ; j ++)
```

```
        {
```

```
            for(k=7;k>-1;k--)
```

```
            {
```

```
                if( bit_test(graf[i][j] , 7-k ))
```

```
                    glcd_pixel( j*8+k,i, ON );
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
//-----
```

```
//Función para la adquisición de los datos y conversión a temperatura
```

```
void adc_temp()
```

```
{
```

```
    int N;
```

```
    int16 a1,a2,a3,a4;          //variables ADC
```

```
    float v1,v2,v3,v4;
```

```
    float t1,t2,t3,t4;
```

```
    float sum1,sum2,sum3,sum4;
```

```
    sum1=0;                    //valores iniciales
```

```
    sum2=0;
```

```
    sum3=0;
```

```
    sum4=0;
```




```
setup_adc_ports(AN0_TO_AN3);    //configuración ADC
setup_adc(ADC_CLOCK_INTERNAL);
delay_us(40);

for(N=1;N<=5;N++){

    set_adc_channel(0);        //TERMISTOR 1
    delay_us(30);
    a1=read_adc();            //lee la entrada del ADC
    v1=5.0*a1/1024.0;         //conversión a tensión
    t1=((v1-1.24)/0.108)+19.0; //cálculo de la temperatura
    sum1=sum1+t1;

    set_adc_channel(1);        //TERMISTOR 2
    delay_us(30);
    a2=read_adc();
    v2=5.0*a2/1024.0;
    t2=((v2-1.255)/0.108)+19.0;
    sum2=sum2+t2;

    set_adc_channel(2);        //TERMISTOR 3
    delay_us(30);
    a3=read_adc();
    v3=5.0*a3/1024.0;
    t3=((v3-1.421)/0.107)+19.0;
    sum3=sum3+t3;

    set_adc_channel(3);        //TERMISTOR 4
    delay_us(30);
    a4=read_adc();
    v4=5.0*a4/1024.0;
    t4=((v4-1.249)/0.107)+19.0;
    sum4=sum4+t4;
}

term1=sum1/5.0;                //resultados
term2=sum2/5.0;
```



```
term3=sum3/5.0;
term4=sum4/5.0;

//ADQUISICIÓN DEL TIEMPO
if(rtci==0){
    ds1307_set_time(00,00,00);    //resetamos el timer
    rtci=1;
}
ds1307_get_time(hrs,min,sec);
}
//-----
//-----PROGRAMA PRINCIPAL-----
void main()
{
    //VALORES INICIALES          ---
    Xpix=10;
    i=0;
    n=0;
    rtci=0;
    delay_ms(1300);            //espera inicial

                                //INICIALIZACIÓN GLCD
    glcd_init(ON);             //Iniciamos la GLCD
    glcd_fillScreen(0);        //Limpiamos la GLCD
    delay_ms(200);

                                //LOGO          ---
    glcd_logo();
    delay_ms(1500);
    glcd_fillScreen(0);        //limpiamos GLCD

                                //IMAGEN DE LA GRÁFICA          ---
    glcd_grafica();           //ejes, gráfica

                                //INICIALIZACIÓN RTC          ---
    ds1307_init();

                                //INICIALIZAMOS LA SD          ---
```



```
        if(SDCard_init()==0){                                //Si NO inicializa la SD
// Enciende el led SD_ERROR, indicando un error al iniciar la memoria.
LED_SD_ERROR_ON
LED_SD_OK_OFF
glcd_text57(20,24,error,1,ON);    //muestra el msj error
delay_ms(1000);
glcd_text57(20,24,error,1,OFF);    //quita el msj error
NOSD=0;
    }

else{                                                        //Si inicializa la SD
// Enciende el led SD_OK, indicando que se inicializó correctamente la memoria.
LED_SD_OK_ON
LED_SD_ERROR_OFF
glcd_text57(14,24,ok,1,ON);        //muestra el mensaje de inicialización
delay_ms(500);
glcd_text57(14,24,ok,1,OFF);      //quita el mensaje de inicialización

//DEFINIMOS EL NOMBRE DEL ARCHIVO TXT                    ---
A=read_eeprom(0x0);
if (A==1||A==2){
    strcpy(NombreCortoArchivo,"Uno.txt");
    strcpy(NombreLargoArchivo,"Uno.txt");
    write_eeprom(0x0,0x2);
}
if (A==3){
    strcpy(NombreCortoArchivo,"Dos.txt");
    strcpy(NombreLargoArchivo,"Dos.txt");
    write_eeprom(0x0,0x3);
}
if (A==4){
    strcpy(NombreCortoArchivo,"Tres.txt");
    strcpy(NombreLargoArchivo,"Tres.txt");
    write_eeprom(0x0,0x4);
}
}
```



```
if (A==4){
    strcpy(NombreCortoArchivo,"Cuatro.txt");
    strcpy(NombreLargoArchivo,"Cuatro.txt");
    write_eeprom(0x0,0x5);
}
if (A==5){
    strcpy(NombreCortoArchivo,"Cinco.txt");
    strcpy(NombreLargoArchivo,"Cinco.txt");
    write_eeprom(0x0,0x6);
}
if (A==6){
    strcpy(NombreCortoArchivo,"Seis.txt");
    strcpy(NombreLargoArchivo,"Seis.txt");
    write_eeprom(0x0,0x7);
}
if (A==7){
    strcpy(NombreCortoArchivo,"Siete.txt");
    strcpy(NombreLargoArchivo,"Siete.txt");
    write_eeprom(0x0,0x8);
}
if (A==8){
    strcpy(NombreCortoArchivo,"Ocho.txt");
    strcpy(NombreLargoArchivo,"Ocho.txt");
    write_eeprom(0x0,0x9);
}
if (A==9){
    strcpy(NombreCortoArchivo,"Nueve.txt");
    strcpy(NombreLargoArchivo,"Nueve.txt");
    write_eeprom(0x0,0xA);
}
if (A==10){
    strcpy(NombreCortoArchivo,"Diez.txt");
    strcpy(NombreLargoArchivo,"Diez.txt");
    write_eeprom(0x0,0x1);
}
```



```
// Inicializa la FAT.
FAT_init();
    delay_ms(500);

if(FAT_FindFile(&NombreCortoArchivo[0],DirectorioRaiz)==0){
    //---DirectorioRaiz==1
    //Si existe el archivo en el directorio principal, entonces se agregan los nuevos datos.
    FAT_OpenAddFile(&NombreCortoArchivo[0],DirectorioRaiz,&encabezado[0]);
}
else{
    //En caso de no existir el archivo lo crea y copia el buffer de datos dentro del nuevo archivo.

FAT_CreateFile(&NombreLargoArchivo[0],&NombreCortoArchivo[0],DirectorioRaiz,&encabezado[0]);
}
LED_SD_OK_OFF
delay_ms(500);
NOSD=1;
}

    //RUTINA PRINCIPAL
while(TRUE){
    adc_temp();

    //DESPLIEGUE TEMPERATURA
    sprintf(s1,"%f",term1);          //conversión de la temp. a cadena
    sprintf(s2,"%f",term2);
    sprintf(s3,"%f",term3);
    sprintf(s4,"%f",term4);
    glcd_text57(98,8,s1,1,ON);
    glcd_text57(98,22,s2,1,ON);
    glcd_text57(98,36,s3,1,ON);
    glcd_text57(98,48,s4,1,ON);

    //MEMORIA SD
    if(NOSD==1){
        sprintf(ho,"%02d",hrs);      //Conversión del tiempo a cadena
        sprintf(mi,"%02d",min);
```



```
    sprintf(se, ":%02d", sec);
    strcat(ho, mi);           //unimos hrs:min:seg
    strcat(ho, se);
    strcat(ho, tab);
        sprintf(temp, "%f", term1);           // Convierte la variable a cadena
    strcat(ho, temp);
    strcat(ho, tab);
        sprintf(temp, "%f", term2);           // Convierte la variable a cadena
    strcat(ho, temp);           // Agrega el valor a la variable
    strcat(ho, tab);
    sprintf(temp, "%f", term3);
    strcat(ho, temp);
    strcat(ho, tab);
    sprintf(temp, "%f", term4);
    strcat(ho, temp);
    strcat(ho, Fin);           // Agrega \n
    strcpy(&ho[0], ho);           // Convierte a puntero
    LED_SD_OK_ON
    //Se agregan los datos en el archivo previamente generado
    FAT_OpenAddFile(&NombreCortoArchivo[0], DirectorioRaiz, &ho[0]);
    // delay_ms(500);
    LED_SD_OK_OFF           //Apaga el led al terminar de escribir los datos
}

//GRAFICA
Ypix1=28-(term1*0.5);           //Cálculo de grados a pixeles
Ypix2=41-(term2*0.5);
Ypix3=54-(term3*0.5);
Ypix4=67-(term4*0.5);

AYpix1[i]=Ypix1;           //Variables con posición pixeles
AYpix2[i]=Ypix2;
AYpix3[i]=Ypix3;
AYpix4[i]=Ypix4;

glcd_pixel(Xpix, Ypix1, on);           //Despliegue de la línea gráfica
```



```
glcd_pixel(Xpix,Ypix2,on);
glcd_pixel(Xpix,Ypix3,on);
glcd_pixel(Xpix,Ypix4,on);

i=i+1;
Xpix=Xpix+1;
delay_ms(200);

if(Xpix==97){
  Xpix=10;
  for(n=0;n<=87;n++){
    glcd_pixel(Xpix,AYpix1[n],off);
    glcd_pixel(Xpix,AYpix2[n],off);
    glcd_pixel(Xpix,AYpix3[n],off);
    glcd_pixel(Xpix,AYpix4[n],off);
    Xpix=Xpix+1;
  }
  Xpix=10;
  n=0;
  i=0;
}

delay_ms(500);
glcd_text57(98,8,s1,1,OFF);           //Borramos los valores anteriores
glcd_text57(98,22,s2,1,OFF);
glcd_text57(98,36,s3,1,OFF);
glcd_text57(98,48,s4,1,OFF);
}
}
```

APÉNDICE F. Artículo en Congreso

K. Castan, E. Cardiel, S. Arias, P. Rogelio Hernández, “*Desarrollo de un sistema de monitoreo del estado de irrigación sanguínea en extremidades inferiores en humano a través de la medición de temperatura*”, VI Congreso Latinoamericano de Ingeniería Biomédica CLAIB 2014, pp. 167-170.

Desarrollo de un sistema de monitoreo del estado de irrigación sanguínea en extremidades inferiores en humano a través de la medición de temperatura

K.T. Castán¹, E. Cardiel¹, S. Arias¹ and P.R. Hernández¹

¹CINVESTAV-IPN/Ingeniería Eléctrica, Sección Bioelectrónica, México, México

Abstract— Mobility problems in people due to an illness or an accident may be permanent or transitory; in both cases monitoring blood flow is very important because a low blood flow could generate secondary risks. In this paper a system based on miniature NTC thermistor for measuring skin temperature was developed, looking for a possible correlation between this parameter with blood flow velocity evaluated in the posterior tibial artery, in order to evaluate the state of blood irrigation in lower limbs. Measurement system has 3 thermistors and a signal conditioner stage; signals were collected by the data acquisition card NI USB 6008® and displayed in a PC by a graphic interface developed in LabView®. For this study, fourteen subjects (five women and nine men) were submitted to positional changes on their right leg to measure temperature in the hallux, ankle and calf. A direct relationship resulted when average temperature data were compared with average measurements of blood flow velocity, suggesting that temperature could be used to monitor the state of blood irrigation in lower limbs.

Palabras claves— Doppler, temperatura de la piel, termistor, velocidad de flujo sanguíneo.

I. INTRODUCCIÓN

Los problemas de movilidad en una persona a causa de una enfermedad o un accidente pueden ser permanentes o de carácter transitorio. En estas condiciones la formación de úlceras por presión, riesgos vasculares, inflamación por retención de líquidos, etcétera, representan un grave riesgo para la salud y eventualmente para la calidad de vida de estas personas.

En consecuencia, resulta de fundamental importancia monitorear que el flujo sanguíneo, sobre todo en las zonas o puntos más lejanos como las extremidades inferiores, sea adecuado, toda vez que la sangre se encarga de llevar nutrientes y oxígeno a los diferentes tejidos y órganos del cuerpo. Por otro lado, también tiene la importante función de recoger los desechos y el dióxido de carbono para ser eliminados [1].

Conforme la sangre fluye a través del tejido cutáneo, se transporta calor, además de los nutrientes [3]. Se ha reportado que si el flujo aumenta, la temperatura de la piel tiende a aumentar. En tanto que si el flujo disminuye, la temperatura también disminuye [3]. Por lo tanto, la medición de la

temperatura puede ser utilizada como un indicador regional del flujo sanguíneo de la piel [2]. En este sentido, existen diversos estudios [4,5], que muestran que la temperatura de la piel puede ser un indicador del estado del flujo sanguíneo.

En el presente artículo, se reporta el desarrollo de un sistema de monitoreo del estado de irrigación sanguínea a través de la mediciones de la velocidad de flujo sanguíneo en la arteria tibial posterior y la temperatura de la piel en extremidades inferiores en humanos.

El sistema de medición propuesto cuenta con tres termistores miniatura NTC como sensores de temperatura, una etapa de acondicionamiento y linealización de la señal; un módulo NI DAQ USB 6008® para digitalizar las señales y enviarlas a la PC y finalmente una interfaz gráfica desarrollada en Labview® para visualizar la información.

Los sensores de temperatura se emplearon para la medición en el hallux, tobillo y cara interna del músculo gastrocnemio (pantorrilla) de la pierna derecha de los sujetos.

Las mediciones de velocidad de flujo sanguíneo fueron realizadas con un flujómetro Doppler modelo 810-B®, con una frecuencia de trabajo de 8.1 MHz.

En el protocolo de medición participaron cinco mujeres y nueve hombres entre 24 y 56 años en posición sedente. Este protocolo fue dividido en tres fases, la primera se refiere al acondicionamiento de la extremidad inferior derecha, mientras que en la segunda y tercera se modificó la posición y altura de dicha extremidad. Los datos de temperatura y velocidad de flujo sanguíneo se registraron durante las dos últimas fases. Los datos de temperatura (hallux, tobillo y pantorrilla) de todos los sujetos fueron promediados y comparados gráficamente con el promedio de los datos de velocidad de flujo sanguíneo.

De acuerdo al protocolo ejecutado y a los resultados se aprecia una relación directa entre la temperatura y la velocidad de flujo sanguíneo en la extremidad inferior.

II. METODOLOGÍA

A. Sistema de Medición

El sistema de medición de la temperatura de la piel fue caracterizado para un rango de 20 a 40°C. Cuenta con tres termistores, una etapa de acondicionamiento de la señal, una

fente de energía, un módulo NI DAQ USB-6008® y una interfaz gráfica de usuario para la visualización de los datos (Fig. 1). Se empleó un flujómetro Doppler modelo 810-B® para la medición de la velocidad de flujo sanguíneo.



Fig. 1 Diagrama a bloques del sistema.

1) Sensores de temperatura.

Los sensores de temperatura utilizados fueron termistores miniatura NTC con un rango de trabajo de 0 a 70°C, una resistencia nominal de 10 kΩ para una temperatura de 25°C con tolerancia de ±0.2°C (@25°C) y 2.4 mm de diámetro, seleccionados por su rápida respuesta y su alta sensibilidad (alrededor de 300 Ω/°C). Para la sujeción con la extremidad inferior, los sensores fueron colocados en un brazaletes hecho a base de etilvinilacetato (goma EVA) y cintas ajustables de velcro (Fig. 2).

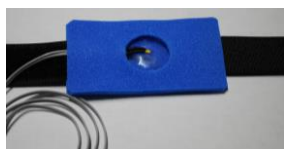


Fig. 2 Brazaletes con termistor utilizado para el registro de la temperatura.

2) Circuito de acondicionamiento.

Debido a que la respuesta de los termistores no es lineal se llevó a cabo un circuito de acondicionamiento (Fig. 3), el cual consta de una etapa de linealización, un eliminador de offset, una amplificación para obtener una sensibilidad de 100 mV/°C y sus respectivas etapas de filtrado.

3) Módulo de adquisición de datos.

Para el proceso de digitalización de las señales de temperatura y posterior comunicación con la PC se empleó el módulo DAQ NI USB-6008® de Texas Instruments.

4) Interfaz gráfica.

El despliegue de los datos adquiridos por el módulo DAQ a la PC se muestra a través de una interfaz de usuario desarrollado en el software Labview 2012® estudiantil. En esta se monitorean en tiempo real las gráficas de los sensores de temperatura, su valor numérico y la selección del almacenamiento de los datos para su análisis posterior.

5) Flujómetro Doppler modelo 810-B®.

El medidor de flujo Doppler trabaja a una frecuencia de 8.1 MHz. Para las mediciones de velocidad de flujo sanguíneo se ajustó manualmente para una mejor lectura con un filtro y un nivel de salida igual a 3, un nivel de entrada de 3.3 aproximadamente y un multiplicador igual a 2.

neo se ajustó manualmente para una mejor lectura con un filtro y un nivel de salida igual a 3, un nivel de entrada de 3.3 aproximadamente y un multiplicador igual a 2.

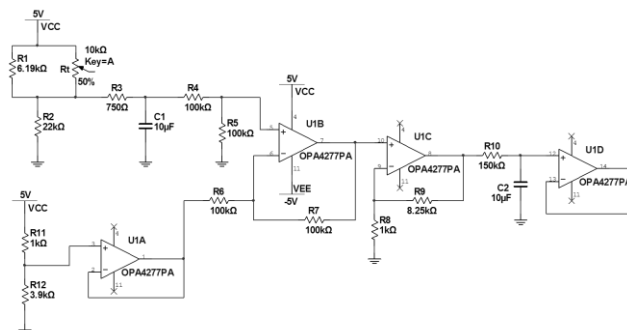


Fig. 3 Circuito de acondicionamiento del termistor.

B. Protocolo de Medición

Sujetos: Cinco mujeres y nueve hombres, no fumadores, entre 24 y 56 años de edad participaron en las pruebas. A cada uno se le pidió realizar dos cambios de posición y altura de su pierna derecha como estímulo. Todos fueron informados de la naturaleza del estudio y proporcionaron su consentimiento.

Condiciones del laboratorio: Las pruebas fueron efectuadas en un laboratorio a una temperatura ambiente de 23°C ±0.5, con una humedad relativa de 55% ±5.

Puntos de medición: Debido a que el intercambio de calor entre el cuerpo y el ambiente se manifiesta significativamente en la región más alejada de las extremidades, se eligió el hallux como primer punto de medición de temperatura; el tobillo, a 4 cm del punto para la medición de velocidad de flujo y la pantorrilla (a 16 cm de separación con respecto al sensor del tobillo) como valor de referencia para la comparación con el punto más alejado (Fig. 4). La medición de la velocidad del flujo sanguíneo se realizó en la arteria tibial posterior.

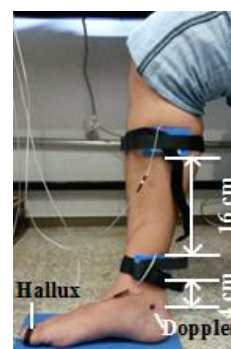


Fig. 4 Puntos de medición en la extremidad inferior derecha.

Protocolo experimental: Cada prueba duró aproximadamente 40 minutos dividida en tres fases (Fig. 5), durante las cuales el sujeto permaneció en posición sedente. Antes de iniciar la prueba se le pidió a cada sujeto descubrirse la pierna derecha para efectuar un procedimiento de limpieza de las zonas a medir. Finalizado esto, se inició la primera fase de 10 minutos de adaptación a la temperatura ambiente, con las piernas del sujeto en ángulo recto. Dentro de este tiempo se localizó con el flujómetro Doppler el punto óptimo para la medición en la arteria (marcándose con pintura no tóxica), de igual manera durante esta fase se colocaron los sensores de temperatura. La segunda fase consistió en pedir a cada voluntario en elevar y dejar descansar la pierna derecha sobre un cojín durante 15 minutos (Fig. 6), quedando a un ángulo aproximado de 20° con respecto al suelo. En la tercera fase se le pidió al sujeto volver a la posición inicial apoyando el pie sobre una base con una altura de 9 cm. A partir del inicio de la segunda y hasta finalizar la tercera fase, se almacenaron y registraron los datos de temperatura y velocidad de flujo sanguíneo con el flujómetro Doppler cada 5 minutos aproximadamente.

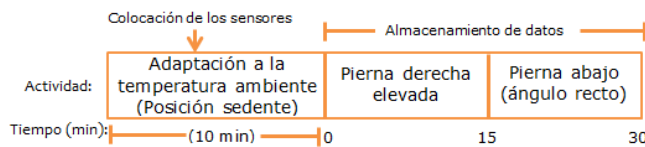


Fig. 5 Esquema de las fases del protocolo experimental.

Cálculo de la velocidad de flujo sanguíneo: Este dato está determinado por la ecuación 1 [6]. Donde θ es igual al ángulo de transductor con la piel, el cual se procuró fuera de 45° para cada medición y F es la diferencia absoluta entre la frecuencia de la señal emitida y la frecuencia de la señal recibida, dado por el mismo equipo.

$$\text{Velocidad de Flujo Sanguíneo} = \frac{F \cdot 1.56 \times 10^5}{2.8.1 \times 10^6 \cdot \cos(\theta)} \quad (1)$$

Procesamiento de los datos: La adquisición de la señal de temperatura se efectuó promediando 128 muestras a una frecuencia de muestreo de 1 kHz.

Con la ayuda del software Matlab® 2012b se obtuvo una gráfica que muestra el promedio de los datos de temperatura de todos los sujetos para cada punto de medición versus los promedios de los valores de velocidad de flujo sanguíneo obtenidos durante la segunda y tercera fase. Con objeto de observar si se presentaba un aumento o disminución en el valor de la temperatura para cada punto de medición, se determinó el promedio de los datos de temperatura y velocidad de flujo sanguíneo para la segunda y tercera fase de la prueba para cada sujeto.

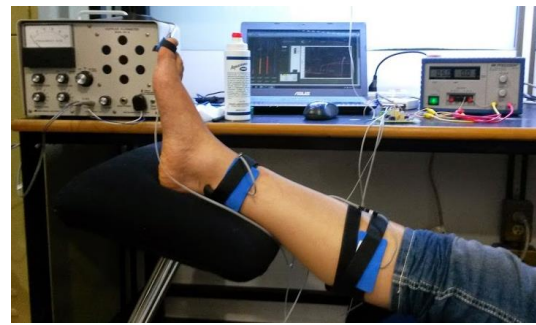


Fig. 6 Posición de la pierna derecha durante la segunda fase de la prueba.

III. RESULTADOS Y DISCUSIÓN

De las tres curvas de temperatura correspondientes a las obtenidas con la pierna elevada, mostradas en la gráfica de la Fig. 7, se observa que el hallux es aproximadamente 4°C menor en relación al tobillo y pantorrilla, en tanto que entre estos se obtuvo una diferencia de temperatura más estrecha (menor a 1°C). Dentro de estos primeros 15 minutos, la tendencia de temperatura es ascendente, sobre todo para el caso del hallux y tobillo; asimismo los valores de velocidad de flujo sanguíneo son los más altos.

Por el contrario, considerando el intervalo de tiempo restante (pierna abajo), el tobillo y hallux presentan una tendencia negativa, mientras que la temperatura de la pantorrilla continua incrementándose. Por su parte la velocidad de flujo descende pero en los minutos 20-30 comienza a ascender nuevamente.

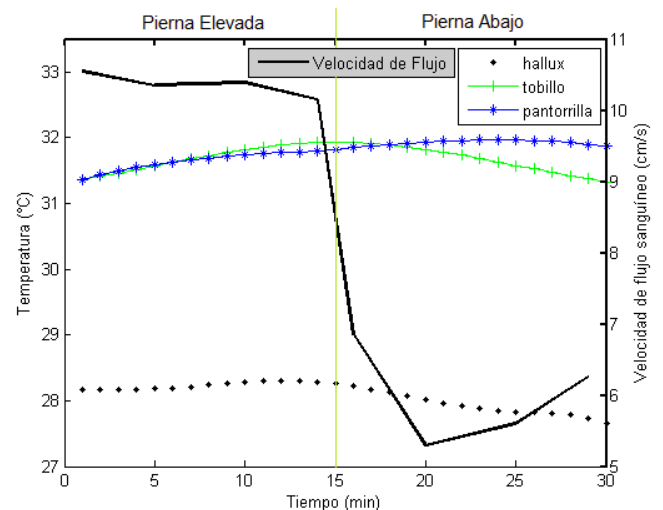


Fig. 7 Promedios de temperatura para el hallux, tobillo y pantorrilla comparados con el promedio de las velocidades de flujo de todos los sujetos.

Tabla 1 Temperatura promedio [°C] en los puntos de medición correspondientes a las posiciones de la pierna en cada sujeto.

Sujeto	Hallux		Tobillo		Pantorrilla	
	Pierna elevada	Pierna abajo	Pierna elevada	Pierna abajo	Pierna elevada	Pierna abajo
1	22.09	21.83	29.21	29.04	30.92	30.91
2	32.95	32.88	31.77	32.12	32	32.14
3	33.08	31.39	33.82	33.46	33.33	33.21
4	27.33	27.04	31.87	32.26	31.62	31.37
5	27.7	25.82	30.63	30.41	30.11	30.4
6	29.56	30.12	31.42	31.84	31.48	32.84
7	26.87	27.33	32.61	32.75	32.88	32.9
8	31.16	31.57	32.11	32.46	31.91	32.3
9	25.84	24.27	31.8	31.34	31.86	31.81
10	24.67	23.77	30.78	30.32	30.66	30.76
11	29.92	29.01	33.64	32.85	32.63	32.99
12	22.79	22.19	29.93	29.58	30.51	31.11
13	29.88	31.12	32.76	32.21	32.73	33.1
14	31.5	31.25	31.91	31.61	30.94	31.2

El comportamiento promedio de las mediciones demuestra que el mantener la pierna en una posición elevada, conlleva una velocidad del flujo sanguíneo mayor que en el caso de la pierna abajo (posición normal sedente). De igual forma la temperatura en el hallux incrementa 0.3°C, y en el tobillo y pantorrilla hasta 0.7°C en el primer intervalo de tiempo. Para el caso de pierna abajo, en el hallux desciende aproximadamente 0.6°C, el tobillo regresa a su valor inicial, y en el caso de la pantorrilla la temperatura continúa incrementándose.

Considerando las mediciones obtenidas de los sujetos de manera individual durante el cambio de posición, tabla 1, se observa que 10 de 14 sujetos, casillas en gris, presentan un decremento en temperatura en hallux. En el caso del tobillo 9 sujetos incrementan su temperatura mientras que solamente 5 se mantienen prácticamente estables. Por otro lado, los datos registrados en la pantorrilla indican que solamente 3 sujetos presentaron un decremento en la temperatura, y los 11 restantes sufrieron un incremento. De la tabla 2 se observa que 11 de los sujetos presentan una disminución en la velocidad de flujo sanguíneo.

El comportamiento descrito anteriormente puede deberse a que, no obstante, que la elevación de la pierna pareciera ser un impedimento para un adecuado flujo sanguíneo hacia la extremidad inferior, por efecto de la gravedad, resulta ser más un beneficio para el retorno venoso, resultando estimulada la circulación y generando un aumento en la velocidad de flujo sanguíneo y esto a su vez se manifiesta en un incremento de la temperatura en esta extremidad. Por el contrario, al tener la pierna en ángulo recto, la acción de la gravedad afecta a este retorno venoso teniendo una disminución en la velocidad de flujo, y en consecuencia un decremento en la temperatura de la extremidad.

Tabla 2 Velocidad de flujo sanguíneo promedio [cm/s] en los puntos de medición correspondientes a las posiciones de la pierna en cada sujeto.

Sujeto	Pierna elevada	Pierna abajo
1	2.32	2.04
2	16.07	10.76
3	5.72	4.36
4	14.44	5.45
5	2.32	1.09
6	14.98	4.49
7	5.86	6.67

Sujeto	Pierna elevada	Pierna abajo
8	15.12	9.94
9	16.48	3
10	6.81	3
11	9.26	3.81
12	7.22	1.91
13	13.62	13.21
14	14.84	14.3

IV. CONCLUSIONES

De acuerdo a este protocolo los resultados preliminares obtenidos demuestran una relación directa entre la temperatura de la piel en la extremidad inferior y la velocidad de flujo sanguíneo en la mayoría de los sujetos. Sin embargo se considera necesario realizar pruebas con intervalos de tiempo más largos y más casos de estudio; así como observar con más detalle el comportamiento de la pantorrilla para reforzar esta conclusión.

AGRADECIMIENTOS

Los autores agradecen al CONACyT por su aportación para la realización de este artículo, y a todos los voluntarios que participaron en las pruebas.

CONFLICTO DE INTERÉS

Los autores declaran que no tienen conflicto de interés.

REFERENCIAS

- Ganong W F (2006) Fisiología Médica. El Manual moderno, 18ª Edición en español, 2006, pp.272-275, 668.
- Charkoudian N. (2003) Skin Blood Flow in Adult Human Thermoregulation: How It Works, When It Does Not, and Why, Mayo Clin Proc. 2003; 78:603-612.
- Guyton A C, Hall J E (2006) Tratado de Fisiología Médica. Ed. Elsevier
- Vuksanović V., et. al. (2008) Nonlinear Relationship between Level of Blood Flow and Skin Temperature for Different Dynamics of Temperature Change. Biophys J. 94(10):L78-80. DOI: 10.1529/biophysj.107.127860
- Lohman III E. B., et. al. (2012) A comparison of whole body vibration and moist heat on lower extremity skin temperature and skin blood flow in healthy older individuals, Med Sci Monit, 2012; 18(7): CR415-424
- Prks Medical Electronics, Inc. Model 810-B Doppler Flowmeter Operating Manual.

Author: Karina Teresa Castán
 Institute: CINVESTAV-IPN, Sección Bioelectrónica
 Street: Avenida IPN 2508, Zacatenco
 City: México
 Country: México
 Email: kcastan@cinvestav.mx