



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

---

**MASTER EN INGENIERÍA INDUSTRIAL**  
**ESCUELA DE INGENIERÍAS INDUSTRIALES**  
**UNIVERSIDAD DE VALLADOLID**

**TRABAJO FIN DE MÁSTER**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA  
MONITORIZACIÓN DE LA FRECUENCIA CARDÍACA Y DE LA  
SATURACIÓN DE OXÍGENO EN SANGRE**

Autor: D. Pedro José Vera Gallego  
Tutor: Dña. María Isabel Del Valle González

Valladolid, Septiembre, 2020



---

**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

---



## RESUMEN

El presente proyecto se enfoca en la selección de un dispositivo capaz de obtener la medición del ritmo cardíaco y la saturación de oxígeno en sangre en tiempo real. Se realizará el diseño e implementación del software necesario para la monitorización de dichos valores, con una precisión suficientemente alta y disminuyendo el coste frente a otros dispositivos comerciales.

Se especificará el hardware empleado y la configuración necesaria para el correcto funcionamiento del código implementado, así como el coste material del mismo.

## ABSTRACT

The following project is focused on the selection of a device capable of obtaining the heart rate measurement and the oxygen saturation in blood in real time. Also, Will be done the necessary software design and implementation to monitor such values with high accuracy and decreasing the cost in relation to other commercial devices.

The hardware and necessary configuration for the precise operation of the implemented code will be explained, as well as its material cost.



---

**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

---



## CONTENIDO

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS .....	1
1.1. Introducción.....	1
1.2. Objetivos .....	2
CAPÍTULO 2. ANTECEDENTES Y MARCO HISTÓRICO .....	3
CAPÍTULO 3. SENSORES DE PULSO Y SATURACIÓN DE OXÍGENO .....	5
3.1. Bases fisiológicas.....	5
3.1.1. Pulso o ritmo cardíaco.....	5
3.1.2. Oxígeno en sangre.....	5
3.2. Principio de funcionamiento .....	6
3.2.1. Espectrofotometría .....	6
3.2.2. Pletismografía .....	7
3.3. Sistemas de tratamiento de datos .....	9
3.3.1. Protocolo I2C .....	9
3.3.2. Protocolo USART y UART .....	10
3.3.3. Puerto paralelo y puerto serie (USB) .....	10
3.4. Tipos de sensores en función de la disposición de emisor y receptor.....	11
3.4.1. Transmisión .....	11
3.4.2. Reflexión.....	12
CAPÍTULO 4. SENSORES MIKROE.....	13
CAPÍTULO 5. HARDWARE EMPLEADO .....	19
5.1. Sensor elegido .....	19
5.2. Placa de desarrollo.....	22
5.3. Microcontrolador.....	24
5.4. Displays LCD y GLCD .....	25
CAPÍTULO 6. SOFTWARE EMPLEADO.....	27
6.1. IDE. MikroC PRO for PIC .....	27
6.2. Programador MikroProg Suite for PIC .....	29
CAPÍTULO 7. DESARROLLO DEL SOFTWARE .....	31
7.1. Librerías necesarias .....	32
7.2. Algoritmo para el cálculo del ritmo cardíaco.....	35



7.3. Algoritmo para la saturación de oxígeno en sangre .....	44
CAPÍTULO 8. RESULTADOS OBTENIDOS. ....	47
CAPÍTULO 9. DISEÑO DE UN HARDWARE PROPIO .....	51
CAPÍTULO 10. CONCLUSIÓN.....	55
CAPÍTULO 11. LÍNEAS FUTURAS .....	57
CAPÍTULO 12. ESTUDIO ECONÓMICO .....	59
CAPÍTULO 13. BIBLIOGRAFÍA.....	61
<b>ANEXOS</b>	
ANEXO 1. DESARROLLO DEL CÓDIGO EN MIKROC PRO FOR PIC.....	63



## CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS

### 1.1. Introducción

En los últimos años, la ingeniería y la biomedicina han ido de la mano en la investigación de nuevas tecnologías que faciliten el trabajo del personal sanitario. Ya sea aportando nuevos métodos de tratamiento para distintas patologías, o mejorando el material actual, para que sea más efectivo.

Así mismo, en el ámbito de la ingeniería electrónica, más en concreto en la electrónica de instrumentación, se ha innovado para buscar una mayor precisión en los aparatos de medida. La investigación de nuevos componentes que sean más precisos, más pequeños o más baratos, es un campo en continua evolución.

Algunas de las constantes vitales que se suelen medir con este tipo de instrumentos electrónicos son la tensión, presión arterial, la presión ocular, la saturación de oxígeno en sangre o la frecuencia cardíaca. Siendo estos dos últimos apartados, en los que se centrará el presente proyecto.

En la situación actual en que nos encontramos este año 2020, este tipo de sensores, los pulsioxímetros, están siendo muy útiles a la hora de detectar la neumonía, una de las principales causas de muerte de los pacientes con coronavirus. Pues muchos pacientes de coronavirus no muestran sensación de ahogo ni dificultad para respirar a pesar de padecer un caso de neumonía avanzada. Durante los últimos meses, varios médicos han afirmado que medir la saturación de oxígeno en sangre, para detectar a tiempo cualquier anomalía, es esencial para diagnosticar una neumonía.

Un alto número de pacientes que acudieron a urgencias durante la pandemia del coronavirus, no experimentaba la sensación de ahogo que caracteriza a una neumonía, esto se denomina hipoxia silenciosa. Sin embargo, si se presenciaban otros síntomas como tos y fiebre. Es aquí donde aparece la importancia del pulsioxímetro, ya que, tras realizar la medida de saturación de oxígeno en sangre, se detecta que los niveles del mismo son muy inferiores a la normalidad.

Algunos neumólogos, como el doctor David De la Rosa, han notificado que algunos pacientes de coronavirus, acudían al hospital con una saturación de oxígeno próxima al 90%, cuando los valores normales se encuentran entre 95 y 98%, pero sin sensación de ahogo. Dichos pacientes respiraban con una frecuencia más rápida de lo normal. Lo normal es respirar unas veinte veces por minuto, estos pacientes respiraban aproximadamente unas treinta veces por minuto.

Una posible reducción de la curva de contagios, podría haber sido una inversión en pulsioxímetros. Cuando un paciente detectara síntomas, entregándole un pulsioxímetro, enseñándole a usarlo y contactando con él varias veces al día, se evitaría el desplazamiento de estos pacientes, en caso de no ser indispensable. [1]



## 1.2. Objetivos

El principal objetivo del presente trabajo consiste en desarrollar un dispositivo similar a los que se emplean en el ámbito médico para medir el ritmo cardíaco y la saturación de oxígeno en sangre con suficiente precisión. Valorando distintos tipos de dispositivos de detección, todos ellos basados en la técnica de la pulsioximetría por fotodetección.

Una vez elegido un dispositivo, se procederá a realizar el software necesario para configurar, tratar y mostrar los valores obtenidos. Para lo cual, se empleará un PIC programado mediante el software "MikroC PRO for PIC".

Con objeto de mostrar los valores y realizar una correcta calibración, se configurará y programará la placa "EasyPIC PRO v7", para controlar el dispositivo de medición y los displays LCD para visualizar los datos.

Se presentarán los valores obtenidos, en comparación con valores que proporcionan otros sensores comerciales similares, como pueden ser los pulsioxímetros de dedo empleados en enfermería, o los sensores integrados en relojes inteligentes y pulseras de actividad, que están muy en auge entre las personas deportistas, para realizar un control del ritmo cardíaco en todo momento, mientras se realiza una actividad que puede ser más o menos intensa. Con este seguimiento, se pueden evitar paradas cardíacas producidas por un ejercicio demasiado intenso para el aparato circulatorio del deportista.

En último lugar, se pretende realizar el diseño de un dispositivo propio, basado en el sensor seleccionado para los apartados anteriores, con el objetivo de que se minimicen los costes de hardware.





## CAPÍTULO 2. ANTECEDENTES Y MARCO HISTÓRICO

La primera toma de contacto con los dispositivos de detección de pulso y saturación de oxígeno, la llevó a cabo Karl Matthes en 1935, mediante un dispositivo que trabajaba con las longitudes de onda de la luz roja y la luz verde, aunque él mismo, unos años después descubriría que empleando luz roja e infrarroja se obtenían mejores resultados.

En el año 1940 el científico Squire descubrió que las compresiones del tejido arterial causaban modificaciones en la transmisión de luz roja e infrarroja, permitiendo calcular la saturación de oxígeno en sangre. Lo que posteriormente, fue usado por Wood en 1949, para calcular la saturación absoluta en función de la densidad de luminosa y la presión, pero este método fue ineficiente, ya que las fotoceldas y emisores de luz eran muy inestables.

Shaw, en 1964, empleó un equipo de medición de gran tamaño y un gran coste material, para la tecnología de la época, que consistía en emitir ocho longitudes de onda distintas. Este método no era muy preciso y se limitaba a funciones pulmonares.

Fue en 1974, cuando Takuo Aoyagi y Michio Kishi, en busca de realizar la medición de forma no invasiva, utilizaron la relación entre la absorción de las longitudes de onda roja e infrarroja. Sus modelos, ideas e instrumentos se adaptaron al uso médico, se mejoraron y comercializaron en torno al año 1980. A partir de esta fecha, diversas empresas comenzaron a investigar y comercializar distintos tipos de pulsioxímetros. [2]

Con estos avances tecnológicos, se mejoró mucho el método de adquisición de la saturación de oxígeno en sangre, hasta el momento medido mediante gasometría arterial, un sistema invasivo que consiste en inyectar gases en el punto de medición, un método más laborioso, lento e, incluso, peligroso si se practica mal. Además, la aparición de la pulsioximetría permitía realizar una monitorización continua del paciente.

Finalmente, en el año 2009, se desarrolló el primer pulsioxímetro inalámbrico, que se coloca en la yema de los dedos y que permite enviar los datos del paciente vía bluetooth al sistema de monitorización. [3]



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES



## CAPÍTULO 3. SENSORES DE PULSO Y SATURACIÓN DE OXÍGENO

### 3.1. Bases fisiológicas

#### 3.1.1. Pulso o ritmo cardíaco

El ritmo cardíaco es el número de contracciones que realiza el corazón por unidad de tiempo. El modo de comprobar dichas contracciones, es contar las veces que bombea el corazón en un minuto. Para ello, se deben observar las arterias más cercanas al exterior, por ejemplo, la yugular, las muñecas o los dedos.

La frecuencia cardíaca media en un hombre de mediana edad está entre 60 y 80 pulsaciones por minuto. Para una mujer de mediana edad, las pulsaciones se encontrarían unas diez pulsaciones por encima, entre 70 y 90 pulsaciones por minuto.

Poder controlar el ritmo cardíaco con precisión en cualquier momento y de forma rápida, es esencial para comprobar las constantes vitales y tener una rápida capacidad de reacción frente a cualquier anomalía cardíaca. [4]

#### 3.1.2. Oxígeno en sangre

El oxígeno es transportado por la sangre para ser llevado a través del cuerpo, tanto disuelto físicamente en la sangre, como combinado químicamente con hemoglobina.

La hemoglobina es una molécula compleja con un átomo de hierro ferroso en su centro, que permite la unión de una molécula de oxígeno. Como la hemoglobina tiene una estructura tetramérica, puede combinarse con cuatro moléculas de oxígeno.

La hemoglobina es un pigmento rojo contenido en los hematíes de la sangre de los seres vertebrados. Capta el oxígeno que proviene de los alveolos pulmonares, además transportan el dióxido de carbono de nuevo a los pulmones, para que estos lo eliminen del organismo. El color de la hemoglobina depende de la cantidad de oxígeno contenido, siendo de un color rojo más intenso, cuanto más oxigenada está, y de un color más oscuro, púrpura, cuando es hemoglobina desoxigenada. Este es el principio de funcionamiento de los captadores. [5]



### 3.2. Principio de funcionamiento

El funcionamiento de los sensores pulsométricos, se basa en dos principios fundamentales, la espectrofotometría y la pletismografía.

Mediante la espectrofotometría, se ha definido que tanto la hemoglobina como la oxihemoglobina, tienen distintos niveles de absorción de la luz roja e infrarroja.

La pletismografía será la herramienta a utilizar para la detección del ritmo cardíaco, ya que se basa en la absorción de luz, variable con el volumen de sangre arterial.

#### 3.2.1. Espectrofotometría

La espectrofotometría es un método científico que se utiliza para estudiar la cantidad de luz absorbida por una sustancia química. Basándose en el principio de absorción de la radiación electromagnética en la zona visible del espectro, así como en la ultravioleta y la infrarroja. Al receptor llega la muestra de la que se mide la cantidad de luz absorbida en función de la longitud de onda empleada. Dependiendo del tipo de estructura molecular, característica de cada sustancia química, la absorción varía, siguiendo las leyes de Lambert y Lambert-Beer.

La ley de Lambert determina la relación directamente proporcional entre la iluminación recibida por una superficie y el producto de la intensidad de la fuente por el coseno del ángulo que forma la normal a dicha superficie con la dirección del haz de luz y a su vez es inversamente proporcional al cuadrado de la distancia entre la superficie y la fuente.

Matemáticamente se expresaría como se muestra en la ecuación 1.

$$\Delta A = \Delta A' \cdot \cos \alpha \quad \text{Ec. 1}$$

Donde  $\Delta A$  es la porción de superficie iluminada y  $\Delta A'$  la superficie aparente vista desde un cierto ángulo  $\alpha$  respecto a la normal formada con  $\Delta A$  (en este proyecto el ángulo entre el fotodiodo y la luz emitida por los diodos).

$$\Delta \Omega = \frac{\Delta A' \cdot \cos \alpha}{r^2} \quad \text{Ec. 2}$$

$\Delta \Omega$  representa el ángulo sólido en que  $\Delta A'$  es visto desde un punto a una distancia  $r$ .



$$\Delta\Phi = I \cdot \Delta\Omega = I \cdot \frac{\Delta A' \cdot \cos\alpha}{r^2} \quad \text{Ec. 3}$$

El flujo de radiación  $\Delta\Phi$  emitido por el ángulo sólido  $\Delta\Omega$  se divide por el área para finalmente dar la ecuación de Lambert (Ec. 4).

$$E = \frac{\Delta\Phi}{\Delta A'} = \frac{I \cdot \cos\alpha}{r^2} \quad \text{Ec. 4}$$

Por otro lado, la ley de Lambert-Beer determina que cuanto mayor concentración en una solución, mayor será la absorbancia de luz. Existen dos expresiones, una para gases y otra para líquidos. Ya que nos interesa la absorbancia de la sangre, la expresión para líquidos es la mostrada en la siguiente ecuación (Ec. 5).

$$\frac{I_1}{I_0} = 10^{-\alpha l} = 10^{-\epsilon l c} \quad \text{Ec.5}$$

Donde  $I_1$  e  $I_0$  son las intensidades saliente y entrante respectivamente y  $\alpha l = \epsilon l c$  es la absorbancia, cuyos componentes son el coeficiente de absorción ( $\alpha$ ), la longitud atravesada por la luz ( $l$ ), el factor de calibración ( $\epsilon$ ) y la concentración del absorbente en el medio ( $c$ ). [5]

### 3.2.2. Pletismografía

El método con el que se consigue ambas incógnitas, frecuencia cardíaca y oxígeno en sangre, es el mismo. Consiste en emitir alternadamente dos luces con distintas longitudes de onda, una roja y otra infrarroja, desde un emisor hasta un fotodetector. Estas ondas atraviesan tanto la sangre oxigenada o no, como la piel, hueso, tendones y las diferentes capas de músculo de la parte del cuerpo donde se tome la medida. Por lo que interesa hacerlo en un lugar donde se minimicen, como puede ser en la yema de los dedos, en la muñeca o incluso en la oreja. [6]

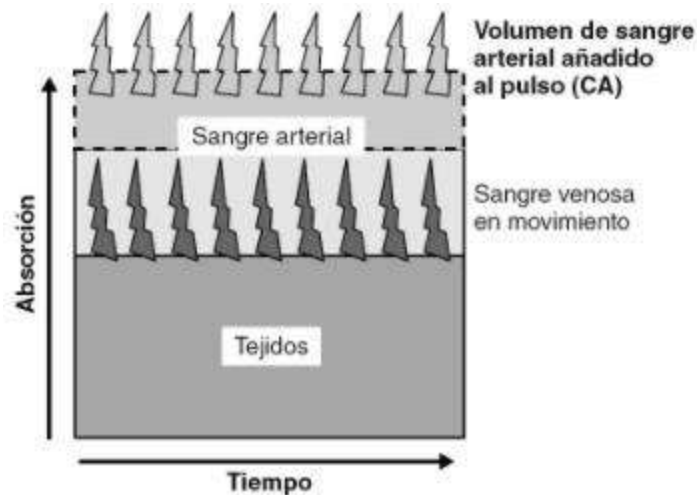


Figura 1. Representación de la absorción de la sangre. [7]

Hay dos factores que afectan a la relación entre las señales luminosas captadas en la pletismografía y el volumen de sangre arterial. El primero es la atenuación de la luz, ya que parte de la luz emitida por los LED será reflejada y otra parte será transmitida a través del cuerpo humano, lo que supone que parte de la luz se refleje en los tejidos más profundos y se ilumine la superficie de la arteria, lo que complica la lectura del sensor, siendo más crítico cuanto menor sea el diámetro de la arteria.

Sin embargo, el segundo factor es la reflexión de la luz en la pared de la arteria. La luz reflejada en la pared más cercana al emisor, hace que más luz llegue al sensor cuanto mayor diámetro tiene la arteria, ya que estará más cerca de este. Por lo tanto, este factor contrarresta al primero.

En conclusión, la forma de onda captada por el sensor dependerá de estos dos factores, que a su vez dependerán de la posición del fotodetector respecto al emisor. Por ejemplo, al alejar el detector del emisor, la reflexión domina el proceso de lectura. [7]

### 3.3. Sistemas de tratamiento de datos

#### 3.3.1. Protocolo I<sup>2</sup>C

Este es un protocolo síncrono, que sirve para interconectar dispositivos mediante bus serie, minimizando las conexiones y tarjetas empleadas, disminuyendo el efecto del ruido, pero a costa de disminuir la velocidad con respecto a un bus paralelo.

Su origen se remonta a principios de los años 80, cuando la compañía Philips comenzó a utilizarlo para la interconexión entre CPU y los dispositivos periféricos.

Está formado por dos hilos, uno de datos, SDA, y otro de reloj, SCL, además de una referencia común a masa, funcionando como maestro-esclavo, con una dirección única por cada dispositivo y pudiendo trabajar como multimaestro. El o los dispositivos maestros inician la transferencia de datos, generando las señales de reloj e indicando cuando termina la transferencia.

El maestro suele ser un microcontrolador, que mediante los puertos de entrada y salida permite controlar la velocidad del bus, inicio y final de las comunicaciones, así como las señales de ACK de dato recibido o enviado. [8]

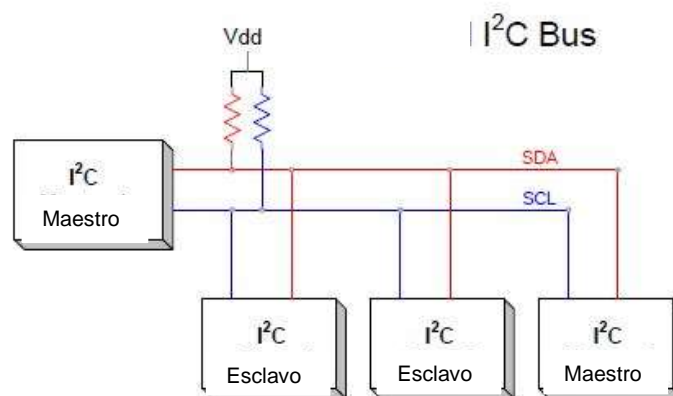


Figura 2. Ejemplo conexión I<sup>2</sup>C. [8]

### 3.3.2. Protocolo USART y UART

Las siglas de estos significan “Universal Asynchronous Receiver/Transmitter” y “Universal Synchronous Asynchronous Receiver/Transmitter” respectivamente. Como sus nombres indican, el UART permite solo trabajar en asíncrono y el USART tanto síncrono como asíncrono.

Con este protocolo, que trabaja con niveles lógicos TTL (transistor-transistor logic), facilita el trabajo a los microcontroladores, ya que no necesitan drivers para convertir los voltajes de entrada.

Los buses que se comunican con estos protocolos deben estar compuestos por una pista TX, otra RX y una de tierra GND. Es un bus para comunicación entre dos dispositivos, debido a que si se introducen más, puede haber problemas de comunicación. Al conectar el pin RX de un primer dispositivo con el TX de un segundo dispositivo, el segundo dispositivo transmitirá información y el primero la recibirá. [9]

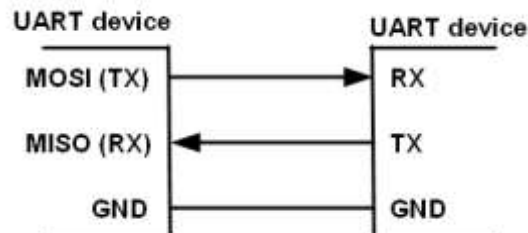


Figura 3. Conexión protocolo UART. [9]

### 3.3.3. Puerto paralelo y puerto serie (USB)

Un puerto paralelo está formado por un conjunto de líneas de entrada y salida, proporcionando una configuración de puerta paralela para la conexión con los diferentes buses de otros dispositivos, Microcontroladores o microprocesadores.

Los datos son transmitidos byte a byte, es decir, se envían los 8 bits a la vez.

El caso del puerto serie, es más complejo. Los PIC que incorporan este protocolo, llevan implementado un sistema de multiplicación de frecuencia de oscilación PLL, para obtener una frecuencia de reloj de 96 MHz, el cual se divide para asignar 48 MHz al funcionamiento del USB y permite emplear los otros 48 MHz en el funcionamiento del microcontrolador.



Los microcontroladores con este puerto pueden presentar aplicaciones como:

- HID. Siglas de Human Interface Device, hace referencia a un tipo de interfaces que toman entradas provenientes de humanos y pueden entregar salidas preparadas para la comprensión de las personas.
- CDC. Microchip permite la emulación de uno o varios puertos COM virtuales.
- Propósito general (USB Hardware Driver). En cada caso, microcontrolador o CPU que trabaje con una conexión USB, debe disponer de las librerías dinámicas necesarias para establecer estas conexiones y un firmware desarrollado en el PIC, que permita la transmisión de datos. [10]

### 3.4. Tipos de sensores en función de la disposición de emisor y receptor

#### 3.4.1. Transmisión

Cuando el fotodiodo se sitúa en el lado opuesto a los emisores de luz, es decir, entre el emisor y receptor se coloca el medio a medir (en el caso de un pulsioxímetro, el dedo), el sensor se basará en la transmisión de la luz a través de los tejidos.

La ventaja de este tipo de disposición es que la señal que llega al fotodetector es más fuerte, además es más sencilla de tratar, ya que la ganancia que necesitará para ser bien detectada es menor.

Sin embargo, este tipo de disposición se emplea en sensores que necesiten gran precisión, pero que no se desplacen demasiado. Además, este tipo de sensores debe colocarse en partes del cuerpo que no sean demasiado anchas. Suele ser en dedos, el lóbulo de la oreja o en el puente de la nariz. [11]

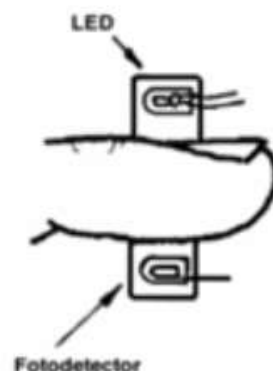


Figura 4. Fotodetector en transmisión. [11]



### 3.4.2. Reflexión

Si el fotodiodo está en el mismo plano que los emisores de luz, el sensor detectará la luz reflejada por el medio a medir, en el momento que más sangre circule, se reflejará más luz, ya que menos luz es transmitida a través de la sangre.

La gran ventaja de estos sensores, es que permiten situarlos en cualquier parte del cuerpo, facilitando la medición con pulseras o bandas que permiten mayor movimiento.

Como desventajas, esta disposición hace que la señal sea más débil, ya que mayor cantidad de luz es transmitida a través del cuerpo humano, de la que es reflejada. Además, la medida de la señal en este tipo de sensores puede ser muy confusa y complicada de tratar, ya que no solo se debe amplificar la señal reflejada, sino que hay que filtrar la posible luz que incida directamente de los emisores sin pasar por el cuerpo. [11]

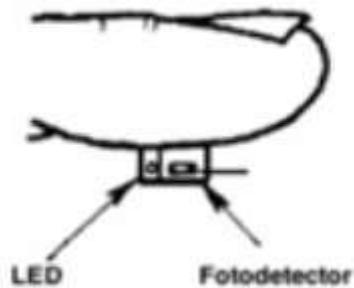


Figura 5. Fotodetector en reflexión. [11]



## CAPÍTULO 4. SENSORES MIKROE

Aunque hay una gran variedad de dispositivos en el mercado, el presente proyecto se centrará en los sensores de Mikroe, sensores ya preparados para emplearlos en microcontroladores PIC, como el que disponemos y utilizaremos para tratar los datos de pulso y saturación de oxígeno obtenidos. Dichos sensores incorporan una librería con las funciones necesarias para enviar y recibir datos, así como controlarlos, almacenarlos y convertirlos en unidades comprensibles.

Todos los sensores de Mikroe, se basan en la misma estructura y forma de detección. En la parte superior se presenta el sistema de iluminación y detección de diodos LED y fotodiodos. A diferencia de otros sensores, estos tienen los diodos emisores y el receptor en el mismo plano, siendo suficiente con poner la yema del dedo pegada al detector, el cual capta la luz reflejada en la superficie del dedo.

Todos los sensores que se presentan a continuación poseen detección de frecuencia cardíaca, pero no todos detectan oximetría, además cada uno de ellos tiene una electrónica y una configuración distinta, basándose todos en el protocolo de comunicación I2C.

### Heart rate click

Este sensor cuenta con el integrado "MAX30100", el cual posee dos LED, uno rojo y otro infrarrojo, un fotodiodo y un sistema de filtrado analógico para minimizar los ruidos y proporcionar lecturas precisas. Dicho sistema, incorpora filtros de tiempo discreto y de control automático de nivel, para variar la ganancia en caso de que se introduzca luz ambiental o ruido a 50/60 Hz, evitando así que estas interferencias hagan que la lectura se aleje del valor real.

Una de las principales características de este dispositivo es su bajo consumo energético, ya que se puede poner en modo espera y la energía utilizada es mínima.

La longitud de onda del led rojo es de 660 nm y la del infrarrojo de 880 nm, la luz reflejada es detectada por el fotodiodo y muestreada por un convertidor analógico/digital de 16 bits, cuya frecuencia de muestreo se puede ajustar entre 50 Hz y 1 kHz.

Para poder calibrar las mediciones, el integrado cuenta con un sensor de temperatura que permite compensar los cambios ambientales. Dicho sensor tiene una precisión de  $\pm 1$  °C y trabaja en el rango de -20 °C a 70 °C.

Las medidas se almacenan en un búfer FIFO de 16 palabras de tamaño. Mientras este tenga espacio, el microcontrolador puede realizar otras tareas, cuando el búfer se

llena, manda una interrupción y envía los datos del búfer a la entrada del microcontrolador.

Su precio unitario es de 16,09 €. [12]

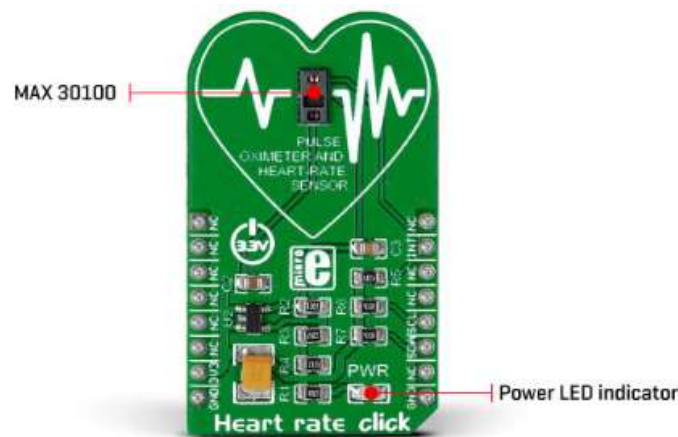


Figura 6. Sensor Heart rate click.

### Heart rate 2 click

Esta placa está basada en el circuito MAXM86161, mediante un sistema de cancelación de luz ambiental para permitir que el sensor funcione en condiciones de alta luz, un convertidor analógico/digital sigma-delta y un filtro discreto de tiempo, realiza muestreos ópticos con rangos de corriente programables entre 4 y 32 micro amperios, una resolución de 19 bits y velocidades también programables entre 8 y 4096 muestras por segundo.

Para disminuir el consumo energético y ampliar la vida útil de la batería, el sensor dispone de una función de proximidad óptica, que hace que la potencia sea mínima cuando no hay una señal luminosa reflejada y captada por el fotodiodo.

Permite alimentación a 5 o 3,3 V y su precio unitario es de 22,02 €. [13]

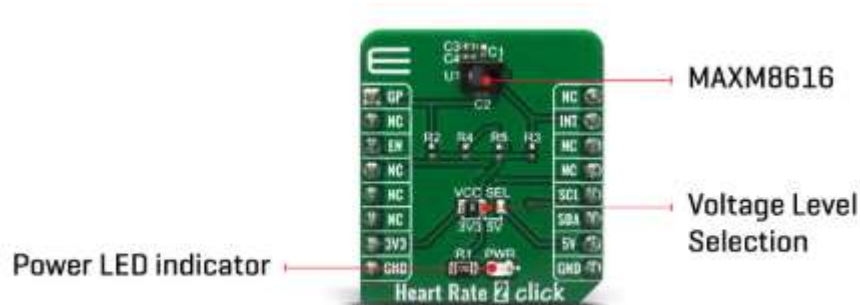


Figura 7. Sensor Heart rate 2 click.

### Heart rate 3 click

Este sensor emplea el integrado “SFH7050”, el cual dispone de tres LED y un fotodiodo, separados por una línea de luz para evitar la diafonía óptica. Los LED son uno verde, otro rojo y otro infrarrojo.

Además, incluye un integrado “AFE4404” que permite configurar el dispositivo para distintos tipos de medición y condiciones exteriores, como pueden ser el tono de piel de la persona o la iluminación respectivamente. Esto aporta gran precisión a este sensor, pero incrementa bastante el precio, elevándose hasta los 33,03 €. [14]



Figura 8. Sensor Heart rate 3 click.

### Heart rate 4 click

Este dispositivo emplea un integrado muy similar al del “Heart rate click”, un integrado MAX30101. La principal diferencia de este es que emplea tres LED en vez de dos, además del LED rojo e infrarrojo, incluye un LED verde, todos ellos programables para una alimentación entre 0 y 50 mA.

Incluye el mismo sistema de espera que el anterior, pero en este caso no emplea el “AFE4404”, por lo que el precio es inferior, siendo su coste unitario de 19,48 €. [15]



Figura 9. Sensor Heart rate 4 click.

### Heart rate 5 click

La placa de este sensor emplea el integrado “SFH7072”, de la familia del que usa el dispositivo “Heart rate 3 click”. Como diferencia, este circuito tiene dos LED verdes, uno rojo, uno infrarrojo y dos fotodiodos. Además, en este caso, se emplea un amplificador con ganancia programable que permite variaciones de voltaje lineales en el convertidor analógico/digital de 22 bits. Como en otros dispositivos, también se incluyen filtros de ruido y de cancelación de luz ambiental.

Para lograr un mayor rechazo del ruido en el amplificador en modo común, se mantiene la señal de lectura de forma diferencial en todo el canal del receptor.

Este sensor posee un switch de selección de modo, con dos polos que deben colocarse en el mismo lado, ambos a la derecha o ambos a la izquierda. Este switch conecta los extremos de cada uno de los fotodiodos para habilitarlos. En caso de estar a la izquierda, conectarán el fotodiodo de banda ancha, para las lecturas de frecuencia

cardíaca, y con ambos a la derecha, se habilita el fotodiodo de banda infrarroja para las lecturas de saturación de oxígeno en sangre.

La tarjeta dispone de un pin de interrupción para evitar que el sensor esté buscando datos todo el tiempo, así, cuando finaliza el ciclo de muestreo, este pin se activa con un uno lógico, permitiendo leer los cuatro registros de salida de datos.

Además, incluye también el integrado “AFE4404”, para configurar los distintos tipos de medición. Este integrado se puede sincronizar con un reloj interno, a 4MHz por defecto, o con un reloj externo, pudiendo subir la frecuencia hasta 60 MHz, pero manteniendo el divisor interno entre 4 y 6 MHz.

El precio unitario de este sensor es de 27,10. [16]

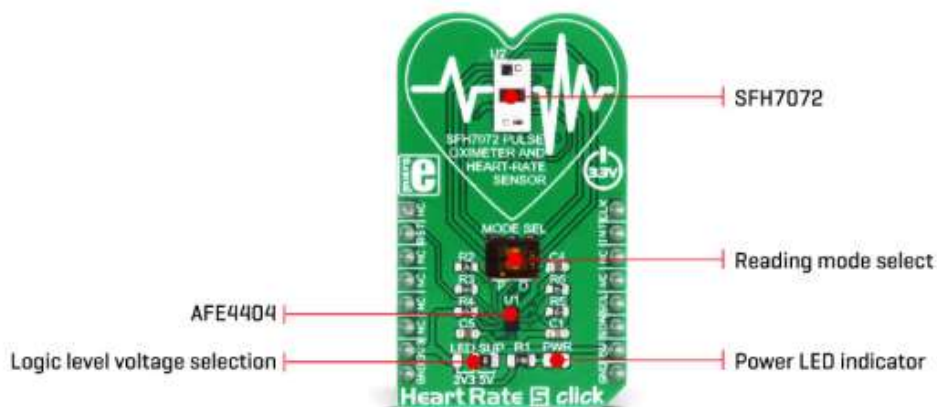


Figura 10. Sensor Heart rate 5 click.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES



## CAPÍTULO 5. HARDWARE EMPLEADO

### 5.1. Sensor elegido

Entre los sensores comentados en el apartado anterior, se ha seleccionado el dispositivo “Heart rate 5 click”, ya que posee una precisión mayor que el resto y dispone de un selector de modo de trabajo, saturación de oxígeno o pulso, lo que permite realizar un código independiente para cada modo.

Se va a emplear la librería que proporciona el fabricante, ya que dispone de las suficientes funciones para realizar la lectura de los registros en los que se almacenan los datos obtenidos por los fotodetectores.

El diseño de este dispositivo, está preparado para conectar mediante mikroBUS. Este modo de conexión es un estándar que permite una amplia capacidad de transmisión de datos, mediante diferentes protocolos (UART, I<sup>2</sup>C, SPI), empleando un mínimo número de pines. [17]

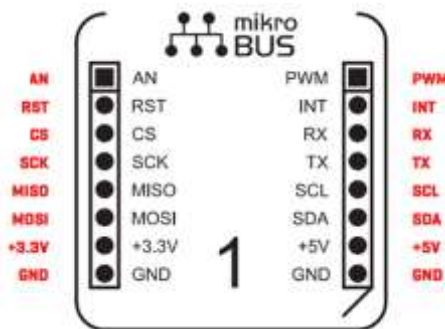


Figura 11. Módulo mikroBUS.

Los pines mostrados en la Figura 11 se refieren a:

- AN: pin analógico.
- RST: reset, reinicio.
- CS: selección de chip SPI.
- SCK: reloj de SPI.
- MISO: entrada de maestro, salida de esclavo.
- MOSI: salida de maestro, entrada de esclavo.
- +3.3V: alimentación a 3.3V.
- GND: referencia a tierra.

- PWM: salida PWM.
- INT: interrupción del dispositivo Hardware.
- RX: pin de recepción UART.
- TX: pin de transmisión UART.
- SCL: reloj en protocolo I<sup>2</sup>C.
- SDA: pin de datos en protocolo I<sup>2</sup>C.
- +5V: alimentación a 5V.

Para comprender mejor como funciona este sensor, se muestra en la Figura 12 el esquema del circuito integrado “AFE4044”, ya que es el que se encarga de gestionar la iluminación de los LED y la lectura de los fotodiodos.

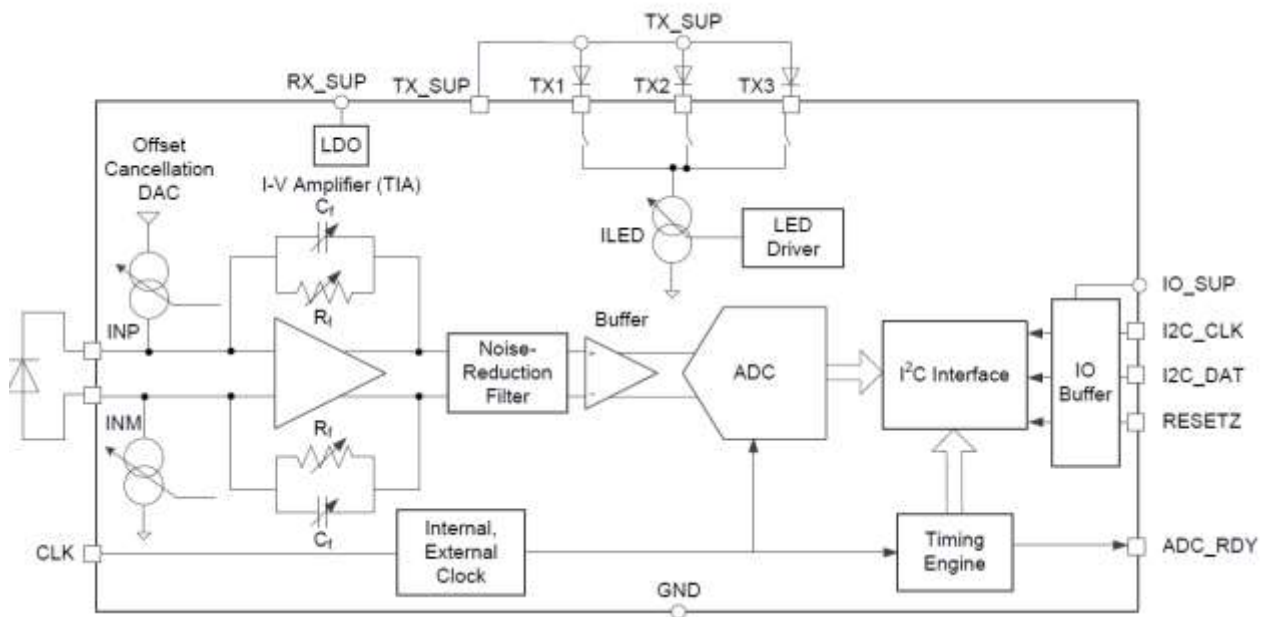


Figura 12. Esquema del circuito integrado AFE4044.

En la Figura 12 se diferencian las siguientes partes:

- En la parte superior, podemos observar el circuito que se encarga de gestionar la iluminación de los LED. Está formado por un controlador o driver, que manda la señal a ILED, para que se cierre el circuito que ilumina el LED1, LED2 o LED, con las salidas TX1, TX2 y TX3 respectivamente.

- Del fotodetector, llega la señal al circuito de cancelación de offset, necesario para minimizar la diferencia entre las intensidades de la entrada inversora y la no inversora, en el esquema se representan por INP e INM.
- Un circuito de amplificación operacional. Da la ganancia necesaria a la señal, para ser diferenciada y convertida correctamente.
- Tras amplificar la señal recibida, esta llega a una etapa de filtrado. Este filtro se puede desglosar en cuatro más pequeños, ya que hay uno para el LED1, otro para el LED2 y para ambos también un filtro de luz ambiental, como se muestra en la Figura 13.
- Una vez filtrada, la señal está lista para ser transformada en la etapa de conversión analógico/digital, la cual enviará mediante el protocolo I<sup>2</sup>C, los datos a un buffer, conectado al pin correspondiente en la placa. [18]

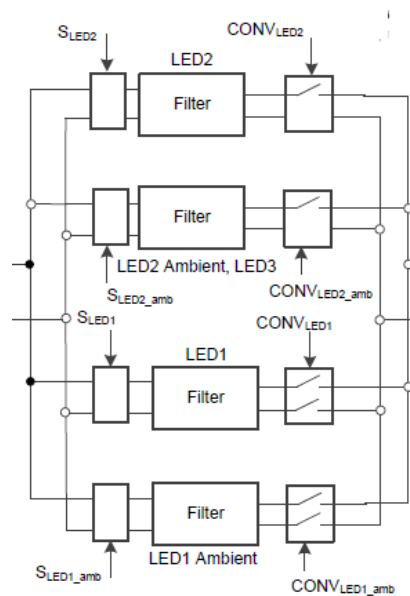


Figura 13. Detalle del sistema de filtrado de ruido.

## 5.2. Placa de desarrollo

Para realizar el montaje se va a emplear una placa de desarrollo “EasyPIC PRO v7” de la marca Mikroelektronika. Esta placa, dispone de 3 puertos mikroBUS como el mencionado anteriormente, marcados en amarillo en la Figura, que nos permite conectar nuestro sensor e incluso añadir 2 dispositivos más. Además, incluye un espacio preparado para conectar una pantalla LCD de 2x16 y una pantalla GLCD de 128x64, conectados adecuadamente a los pines del microcontrolador. Estos espacios son los marcados con un recuadro rojo en la Figura 14.

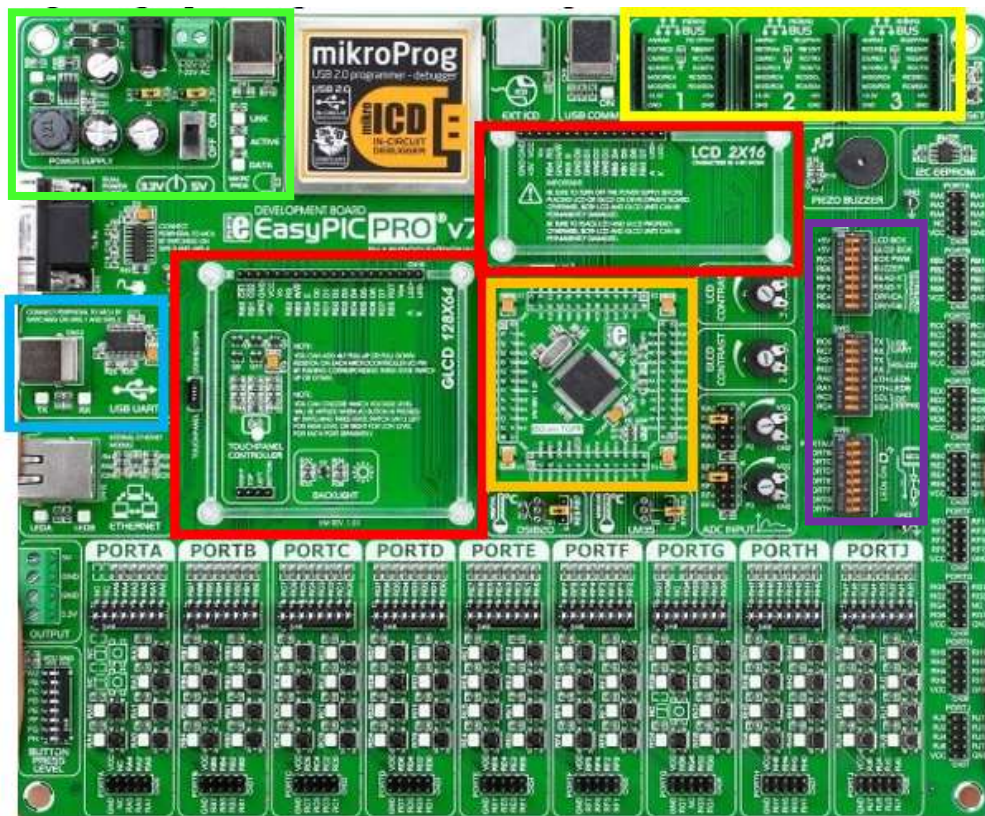


Figura 14. Placa de desarrollo EasyPIC PRO v7.

En una de sus esquinas, marcada en la Figura 14 en verde claro, se encuentra el módulo de alimentación, el cual está preparado para la conexión directa mediante una fuente de alimentación alterna o continua (entre 7-23V y 9-32 V respectivamente). También dispone de una conexión mediante Jack. En este módulo, se incluye un interruptor de encendido y apagado, además de dos jumpers para seleccionar la

alimentación de la placa, 3.3V o 5V. Por último, otra forma de alimentar la placa es mediante el puerto USB del ordenador, conectando al puerto designado para transferir el programa al microcontrolador, mediante “MikroProg”. [19]



Figura 15. Módulo de alimentación de la placa EasyPIC PRO v7.

Otro de los módulos que vamos a emplear es el puerto USB UART, el cual nos permitirá enviar datos al ordenador y mostrarlos por el terminal UART de MikroC. En la Figura 15 se muestra con un recuadro azul.

En morado se marca la zona de configuración de la placa, la cual consta de 24 interruptores para seleccionar el modo en que trabajaran los diferentes puertos del microcontrolador.

Por último, con un recuadro naranja, se muestra en la Figura 14 el espacio reservado para la colocación del microcontrolador PIC. Este debe estar integrado en una placa TQFP de 80 pines. TQFP se refiere a un estándar de Microchip, que consiste en un empaquetamiento cuadrado y fino. El resto de módulos no serán empleados en este proyecto, aunque son interesantes para posibles líneas futuras, pudiendo emplear los módulos de temperatura para determinar las condiciones en que se están tomando las medidas, o el módulo ethernet para enviar los datos a distintos dispositivos.

La configuración de esta placa es muy importante, ya que, en caso de no realizarse correctamente, los datos transmitidos entre microcontrolador y dispositivos, pueden ser erróneos. Para esto, se deben posicionar los interruptores de configuración como se muestra en la imagen de la Figura 16. Es decir, todos en off, exceptuando el 1 y 3 del switch 4, que equivalen a la alimentación del LCD y a la señal de PWM respectivamente, en caso de emplear el GLCD habría que activar el 2. En el switch 5 habría que habilitar el primer y segundo interruptor, que corresponden al TX y RX del puerto UART. [19]



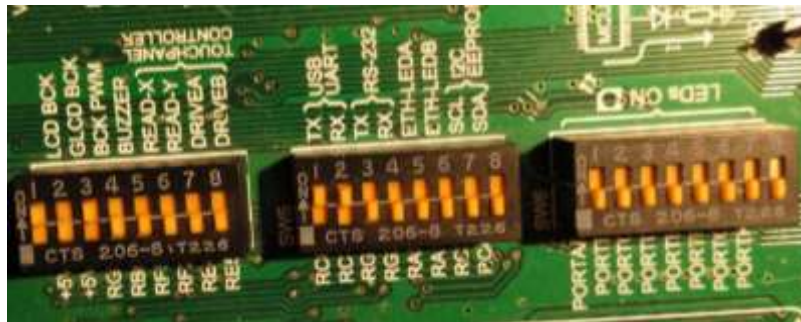


Figura 16. Módulo de configuración de la placa EasyPIC PRO v7.

### 5.3. Microcontrolador

El microcontrolador empleado en el proyecto es el PIC 18F87K22 de Microchip. Un microcontrolador de 80 pines, para el cual ya está preparada la placa de desarrollo “EasyPIC PRO v7”.



Figura 17. Microcontrolador PIC 18F87K22.

Este microcontrolador de 8 bits se adapta perfectamente a aplicaciones con una alimentación continua, de no muy alta energía. Además, dispone de una unidad de medición del tiempo de carga (CTMU), la cual permite desarrollar aplicaciones en las que se requiere una medición muy precisa del tiempo, como es el caso de la medición del pulso.

Permite altos rangos de voltaje y temperatura, por lo que es lo suficientemente robusto para trabajar en condiciones no ideales, siendo su memoria flash y su memoria EEPROM excelentes para emplearlo en aplicaciones industriales.



Incluye un convertidor analógico/digital de 12 bits, 3 comparadores H/W RTCC, 10 canales CCP/ECCP, 11 temporizadores y 69 entradas y salidas generales.

#### **5.4. Displays LCD y GLCD**

Se mostrará el pulso actualizado en un LCD de 2x16 y la gráfica de la señal se plasmará en una pantalla GLCD de 128x64. Ambas pantallas están integradas en la placa de desarrollo y disponen de librerías propias, facilitadas por el fabricante, para realizar una correcta configuración y comunicación.

Ambas pantallas emplean el protocolo I<sup>2</sup>C, al igual que el sensor, por lo que se debe prestar atención a la configuración y evitar que coincidan las direcciones.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES



## CAPÍTULO 6. SOFTWARE EMPLEADO

### 6.1. IDE. MikroC PRO for PIC

IDE, hace referencia al entorno integrado de desarrollo, es decir, el sistema de software que mediante la combinación de herramientas en una misma interfaz gráfica, facilita el diseño de aplicaciones al desarrollador. Entre esas herramientas, suelen aparecer un editor de código fuente y herramientas para la compilación automática local del software. [20]

Para realizar un código que trate la señal y la convierta en un dato numérico que permita leer el pulso y la saturación de sangre, será necesario el IDE “mikroC PRO for PIC”, el cual dispone de un compilador potente y una interfaz bastante sencilla de utilizar. Este IDE incluye una gran cantidad de librerías necesarias para gestionar los distintos dispositivos conectados a la placa de desarrollo.

En la Figura 18 se muestra una captura de la pantalla principal del editor, donde se puede seleccionar el proyecto a abrir o crear un proyecto nuevo desde cero. Además, se puede ver en la parte inferior de la pantalla una ventana de mensajes, en la que se mostrará la información referente al código y a la compilación del mismo.

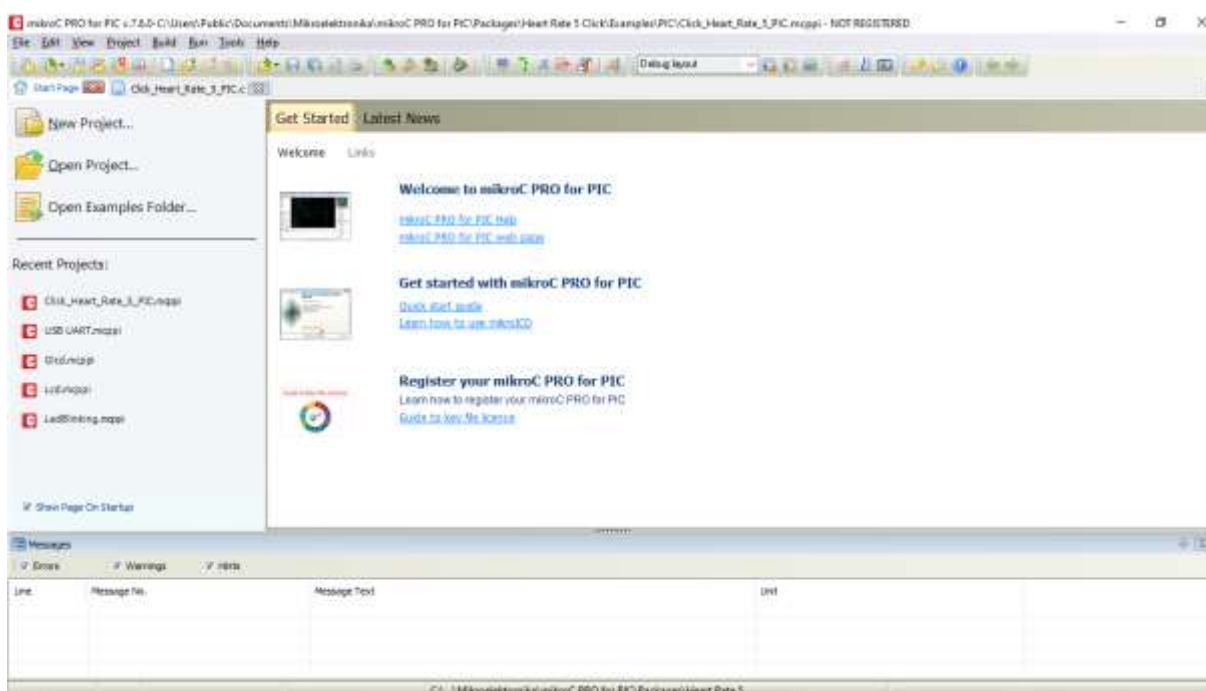


Figura 18. Interfaz del editor mikroC PRO for PIC.

Al abrir un proyecto aparece una nueva pestaña con la extensión .c. En ella se muestra la ventana de edición del código principal.

```
StartPage Lcd.c
* NOTES:
  - Turn on Lcd backlight switch SW4.1. (board specific)
*/

// Lcd module connections
sbit LCD_RS at LATB4_bit;
sbit LCD_EN at LATB5_bit;
29 sbit LCD_D4 at LATB0_bit;
30 sbit LCD_D5 at LATB1_bit;
sbit LCD_D6 at LATB2_bit;
sbit LCD_D7 at LATB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
40 // End Lcd module connections

sbit LCD_BCK at LATG3_bit;
sbit LCD BCK Direction at TRISG3 bit;
```

Figura 19. Editor del programa mikroC PRO for PIC.

Una vez terminado el código, se debe compilar mediante el icono representado con un engranaje verde y otro amarillo habilitado en la barra de tareas, que se muestra en la Figura 20.



Figura 20. Herramientas de compilación y programación.

En la ventana de mensajes se indicará si la compilación ha sido finalizada con éxito o, si en cambio, existe algún error, en cuyo caso aparecerá un mensaje en rojo indicando el error pertinente.

## 6.2. Programador MikroProg Suite for PIC

Es necesario un software para configurar y cargar el código en el microcontrolador PIC. Para ello se va a emplear la interfaz de mikroC, “mikroProg Suite”.

En primer lugar, se requiere indicar la familia del PIC que se desea programar y microcontrolador en concreto que se va a emplear. Después, esta aplicación permite seleccionar un archivo hexadecimal de la carpeta del proyecto compilado, aunque también se puede omitir este paso pinchando sobre el icono de programación, en el editor. Este icono se representa con una flecha y un integrado en la Figura 20.

En la ventana de configuración, mostrada en la Figura 21, se puede seleccionar las condiciones de programación, como la frecuencia, el uso de oscilador interno o externo, etc.

Una vez configurado, pulsando en el botón de “load” se procede a cargar el programa. El avance de la transmisión se puede ver en la barra de carga situada en la esquina inferior izquierda.

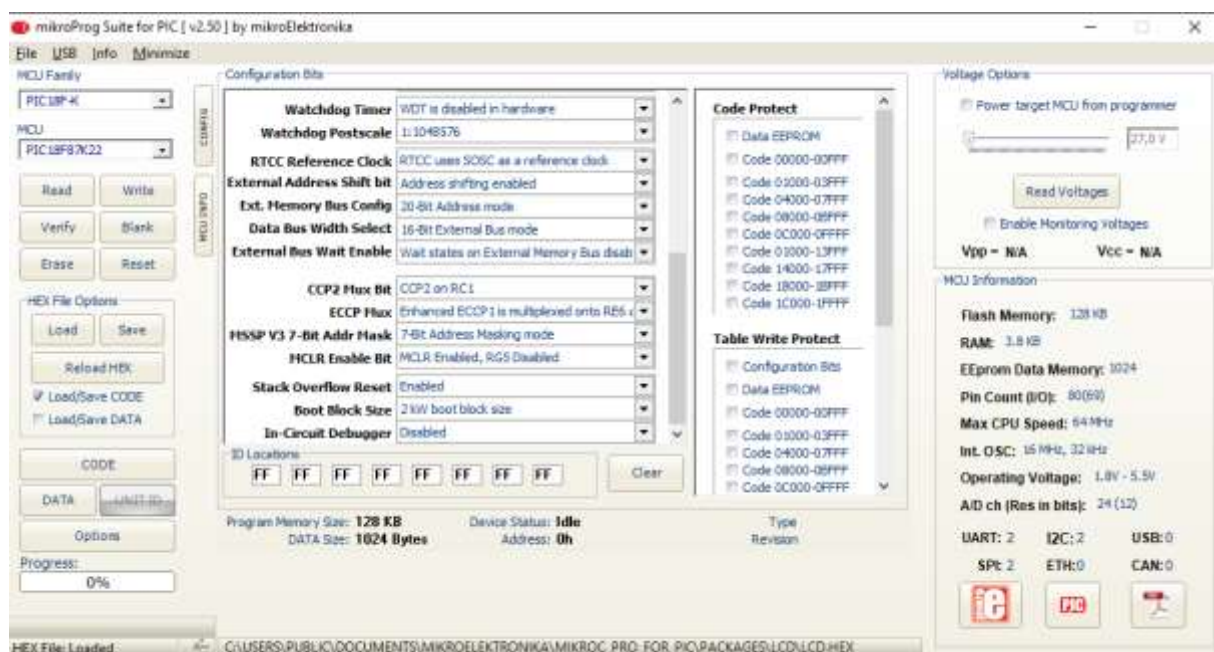


Figura 21. Interfaz del software mikroProg Suite for PIC.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

## CAPÍTULO 7. DESARROLLO DEL SOFTWARE

El primer paso para el desarrollo es configurar la frecuencia del reloj externo. Ya que es la frecuencia a la que trabajarán los dispositivos conectados al microcontrolador.

En este proyecto se debe asignar una frecuencia de 64 MHz, ya que es la frecuencia a la que trabaja el display LCD y emplear frecuencias muy distintas haría que se muestren valores falsos por su pantalla. Además, esta frecuencia se adapta al dispositivo de medición "Heart rate 5 click", ya que emplea un circuito de conversión que transforma la frecuencia de entrada, de hasta 60 MHz, en un rango de entre 4 y 6 MHz, necesarios para el correcto trabajo del integrado "AFE4404".

Para determinar que algoritmo será necesario en la transformación de la señal captada por los fotodetectores y obtener los datos numéricos del pulso y el oxígeno en sangre, es preciso obtener los valores analógicos de corriente almacenados en los registros correspondientes del integrado "AFE4404".

Debido a que el sensor elegido dispone de un interruptor de selección de modo, entre saturación de oxígeno y ritmo cardíaco, se han desarrollado dos aplicaciones, una para cada caso, siendo necesaria una señal externa para seleccionar el código a ejecutar. Dicha señal será una línea del puerto H configurada como entrada digital, conectada a un switch de la placa de desarrollo, aunque también sería posible realizar la lectura del registro correspondiente del "AFE4404".

La distribución del código, será en distintas funciones y el código principal llamará a la función correspondiente al cálculo del ritmo cardíaco o de la saturación de oxígeno en sangre, en función de la entrada, la cual se determinará por la línea menos significativa del puerto H, que se controla desde el módulo correspondiente del puerto H, mediante el interruptor 8 y que será almacenada en la variable "modo", activando el algoritmo de ritmo cardíaco en caso de ser 0 o el de SPO2 si es 1. En la Figura 22 se muestra el interruptor correspondiente.

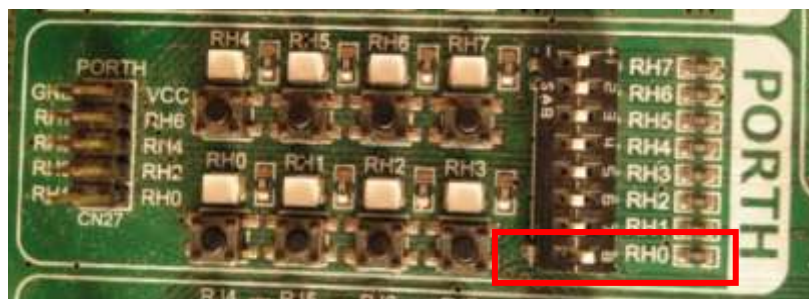


Figura 22. Módulo línea H.

El diagrama de flujo correspondiente al código principal es el de la Figura 23.

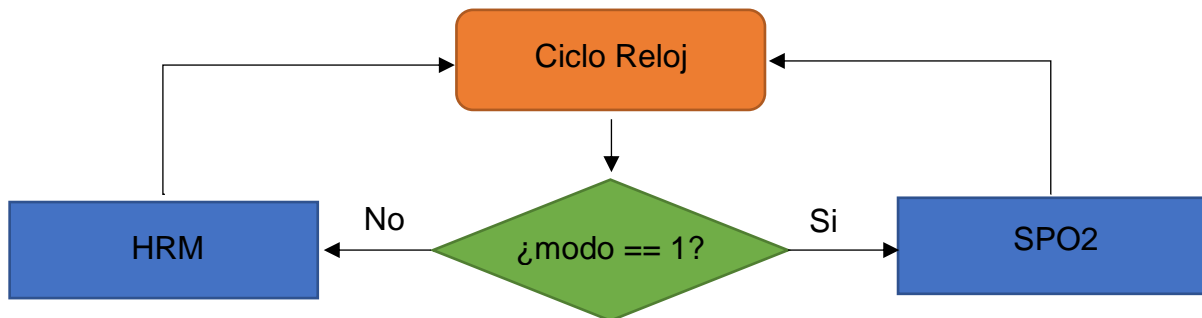


Figura 23. Diagrama de flujo global, selección de modo.

### 7.1. Librerías necesarias

El fabricante proporciona una librería para realizar este código en MikroC, en él se incluyen las siguientes funciones:

- `heartrate5_i2cDriverInit (T_HEARTRATE5_P gpio, T_HEARTRATE5_P i2c, uint8_t slave)`. Esta función es el driver que inicializa la conexión I<sup>2</sup>C del sensor con el microcontrolador. Donde el primero término del paréntesis se refiere al puerto gpio, el segundo al puerto I<sup>2</sup>C, en nuestro caso, estos puertos son los correspondientes pines del módulo de mikroBUS, y el último término es la dirección correspondiente al sensor.
- `heartrate5_hwReset`. Función que resetea los registros del sensor.
- `heartrate5_init`. Se utiliza para reestablecer los valores por defecto, es decir, para inicializar el sensor.
- `heartrate5_swReset`. Función que resetea el programa instalado en el microcontrolador.



- `heartrate5_getAled1val`. Devuelve el valor analógico obtenido de la conversión de la lectura del registro 2Ch del integrado “AFE4404”, que corresponde al valor de lectura de la luz proveniente del LED1.
- `heartrate5_getAled2val`. Como el anterior, este corresponde al LED2 y leído es el 2Ah.
- `heartrate5_getLed1_Aled1val`. Obtiene el valor analógico de la señal ambiental que capta el fotodiodo del LED1, albergado en el registro 2Dh.
- `heartrate5_getLed2_Aled2val`. Obtiene la señal ambiental captada por el fotodiodo del LED2, registro 2Eh.
- `heartrate5_getAled2val_led3val`. Devuelve el valor del LED habilitado, el valor del LED2 o el valor del LED3. Esta función solo se emplea para la saturación de oxígeno.
- `heartrate5_readReg (uint8_t regAddr)`. Con esta función se puede seleccionar el registro que se desea leer. No será necesaria para el algoritmo, pero es útil para realizar comprobaciones de las funciones de lectura. [21]

Además, serán necesarias las librerías correspondientes para comunicarse con el display LCD y el módulo mikroBUS de la tarjeta de desarrollo.

Las funciones a utilizar por el módulo mikroBUS serán:

- `mikrobus_gpiolnit (T_mikrobus_soc bus, T_mikrobus_pin pin, T_gpio_dir dir)`. Función para inicializar el protocolo gpio. Donde el primer término del paréntesis es una estructura formada por las 6 direcciones del bus gpio, el segundo es un entero que indica el pin correspondiente al módulo mikroBUS utilizado, y por último se encuentra la dirección del módulo.
- `mikrobus_i2cInit (T_mikrobus_soc bus, const uint32_t *cfg)`. Inicializa el bus I<sup>2</sup>C, y como en el caso anterior, el primer término son las direcciones de los registros del bus y en este caso, el segundo es un puntero para configurar la comunicación con el módulo UART.





Para comunicarse con el LCD se emplea las siguientes variables globales, que determinan los bits y direcciones a definir en el programa. [22]

- LCD\_D4: bit 4 de datos.
- LCD\_D5: bit 5 de datos.
- LCD\_D6: bit 6 de datos.
- LCD\_D7: bit 7 de datos.
- LCD\_RS: pin de señal para seleccionar registro (de datos o instrucción)
- LCD\_EN: pin de habilitación de señal.
- LCD\_D4\_Direction: dirección del pin 4 de datos.
- LCD\_D5\_Direction: dirección del pin 5 de datos.
- LCD\_D6\_Direction: dirección del pin 6 de datos.
- LCD\_D7\_Direction: dirección del pin 7 de datos.
- LCD\_RS\_Direction: dirección el pin de selección de registro.
- LCD\_EN\_Direction: dirección del pin de habilitación de señal.

Las funciones que interesan en este proyecto son:

- Lcd\_Init. Inicializa las variables del LCD. Pone a cero las variables anteriormente comentadas.
- Lcd\_Out (char row, char colum, char \*text). Permite mostrar por pantalla el contenido de la cadena \*text. Los dos términos iniciales sirven para seleccionar en que espacio del Lcd comenzar a escribir, línea superior o inferior y el número de columna respectivamente.
- Lcd\_Out\_Cp (char \*text). Escribe el texto en la posición actual del cursor.
- Lcd\_Cmd (char ut\_char). Envía un comando al LCD. Aunque la lista de comandos es larga, solo se emplearán el comando que limpia el display (\_LCD\_CLEAR) y el comando que elimina el cursor (\_LCD\_RETURN\_HOME).
- Lcd\_Chr (char row, char colum, char out\_char). Se emplea para escribir un único carácter en una posición determinada.
- Lcd\_Chr\_Cp (char out\_char). Como la anterior, escribe un único carácter en el LCD, pero en este caso lo hace en la posición actual del cursor.





## 7.2. Algoritmo para el cálculo del ritmo cardíaco

Para conocer cómo llega la señal del fotodetector se va a emplear el programa de ejemplo que facilita el fabricante junto a la librería.

Este ejemplo simplemente emplea las funciones de inicialización y configuración del sensor "Heart rate 5 click", además de las funciones que proporcionan la señal ambiente del LED 2 y la señal directa del LED 1.

El código consta de un bucle con una función para mostrar gráficamente los datos obtenidos. Pero que se consulte el dato de lectura del LED 1 depende de la señal ambiente del LED 2, esta señal será 0 cuando se coloque el dedo sobre el sensor, por lo tanto, solo se leerá la señal detectada del LED 1 cuando el dedo esté sobre el sensor, de esta manera se evita que el código se ejecute continuamente, aunque no se quiera medir nada.

Cuando se coloca el dedo sobre el sensor, en cada ciclo de reloj, el microcontrolador mandará la señal obtenida del fotodiodo correspondiente al LED 1 mediante el puerto UART al ordenador, para mostrar en cualquier aplicación que muestre los datos del puerto serie en una gráfica. Además de mandar el valor de la señal, como dato en el eje de ordenadas, envía el valor del contador, que va aumentando en cada ciclo de reloj, como dato en el eje de abscisas.

Empleando como reloj externo una frecuencia de 64 MHz, la gráfica obtenida tiene demasiado ruido, ya que, al tomar muchas lecturas del fotodiodo, la diferencia entre un punto y el siguiente, aunque es mínima, producen muchas oscilaciones. En esta oscilación, influye tanto el circuito interno del sensor, como las condiciones ambientales. Por ejemplo, la temperatura o la iluminación exterior, que al ser corriente alterna tiene una frecuencia de 50 Hz.

La gráfica obtenida con este ejemplo es la de la Figura 24, como se puede observar, cada periodo corresponde a un pulso. El pico tras el que se produce la caída con una mayor pendiente, sería el momento en que más sangre pasa por el dedo, es decir, el movimiento de sístole del corazón, mientras que el punto más bajo, cuando termina la caída con mayor pendiente, sería el movimiento de diástole.

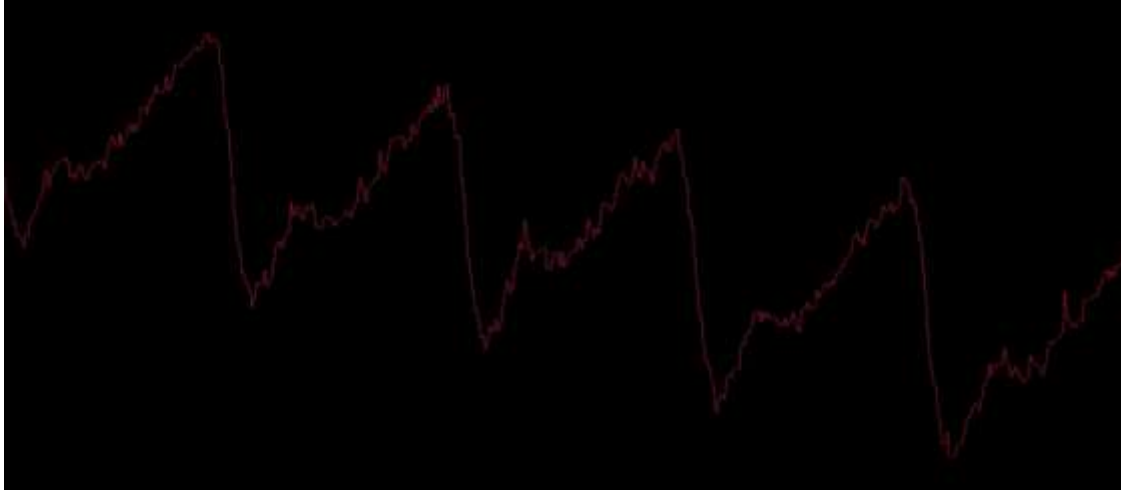


Figura 24. Gráfica de valores obtenidos, sin tratar.

Aunque con estos valores se podría realizar el cálculo del pulso, se decide realizar una media de un conjunto de valores, para conseguir una gráfica más suavizada.

El número de muestras que se ha seleccionado para realizar la media, es de 20. Como se muestra en la Figura, los picos están más definidos, por lo que es más sencillo detectarlos mediante código.

También podemos observar en la Figura 25, que los primeros segundos de lectura no son coherentes, por lo tanto, se debe realizar un filtrado mediante código, que deseche este tipo de valores a la hora de mostrar por pantalla.

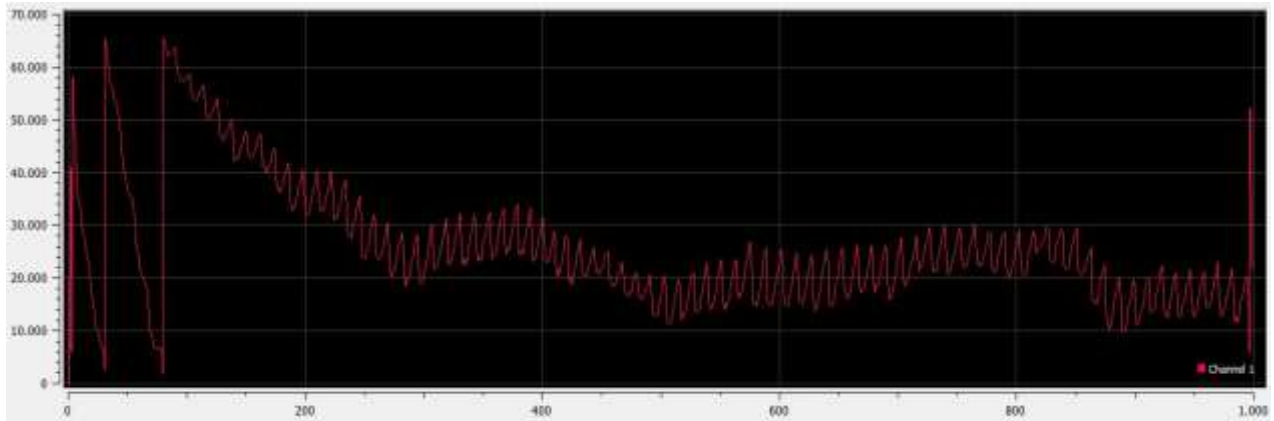


Figura 25. Gráfica suavizada de valores obtenidos.



Para determinar el pulso, se ha decidido tomar seis picos sucesivos durante el primer minuto, lo que equivale a seis latidos del corazón. Dividiendo entre seis el tiempo que discurre entre el primer pico y el séptimo, a partir de este, se obtiene el periodo medio de cada pulso. Para obtener las pulsaciones por minuto, simplemente se deben dividir los 60 segundos entre el periodo. Como el contador está en microsegundos y cada 3,9 microsegundos se activa el contador, se debe multiplicar por 3,9 antes de dividir por los 60000 microsegundos.

Si se amplía la gráfica, se observan los puntos que forman cada pico.

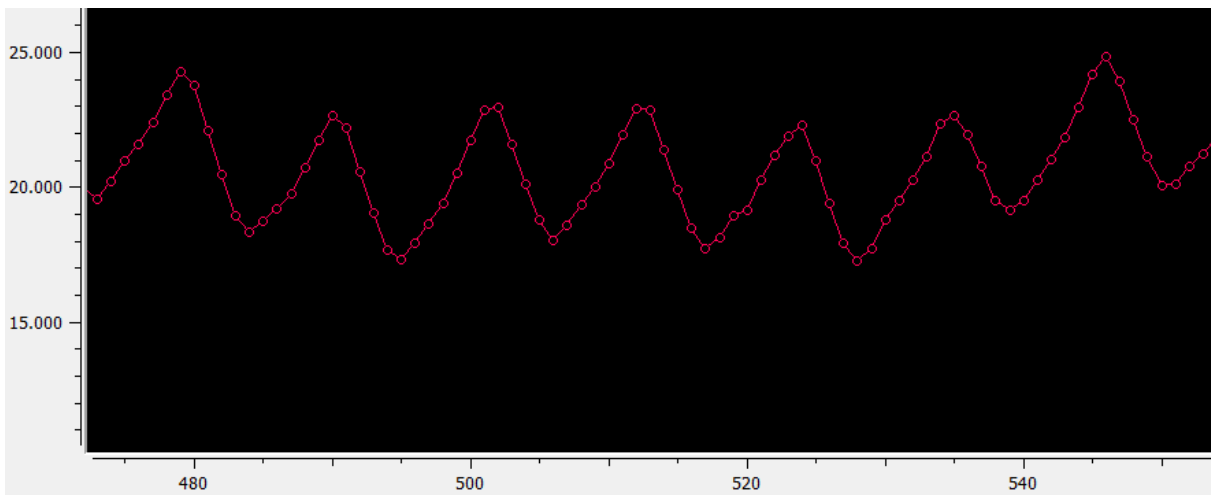


Figura 26. Gráfica ampliada de valores obtenidos.

Como no se obtienen los mismos valores en cada pulsación, no se puede determinar el máximo absoluto, sino que se deben tener en cuenta los datos relativos. Esto se refiere a que para conocer el instante en que el pico llega al máximo, se debe realizar una comparación del último valor leído con el inmediatamente anterior y con los siguientes. En el algoritmo, se ha decidido tomar los cuatro valores siguientes al pico, los cuales deben ser menores que el pico. De esta manera, si el valor anterior es menor y los cuatro valores siguientes son menores, se determina y almacena el instante en que se produce un pico, cuando se repita esta secuencia, se obtiene el segundo valor de pico y se realiza la diferencia entre ambos, lo que representa el periodo.



Para comprender esta secuencia mejor, se muestra el diagrama de flujo en la Figura 27, donde el término “descenso” representa un entero que se incrementa cada ciclo en que el dato disminuye respecto al valor de pico, en caso de que aumente, se reinicia esta variable. El término “último” representa el último valor leído por el sensor. En cuanto al término “anterior” es el valor inmediatamente anterior. Cuando se inicia el ciclo, se almacena en la variable “pico” el valor del contador de tiempo, y en “valor\_pico” la lectura del sensor. Una vez terminado el ciclo, se almacena el siguiente pico y se realiza la diferencia entre el instante actual y el tiempo del pico anterior.

Se empleará también una variable “flag” para indicar el fin o el inicio de la cuenta de puntos descendentes. Esta variable será 0 por defecto, pasando a valer 1 cuando se define un pico.

Este diagrama de flujo corresponde a la detección de picos en la señal, algoritmo que se engloba dentro de la obtención de dato.

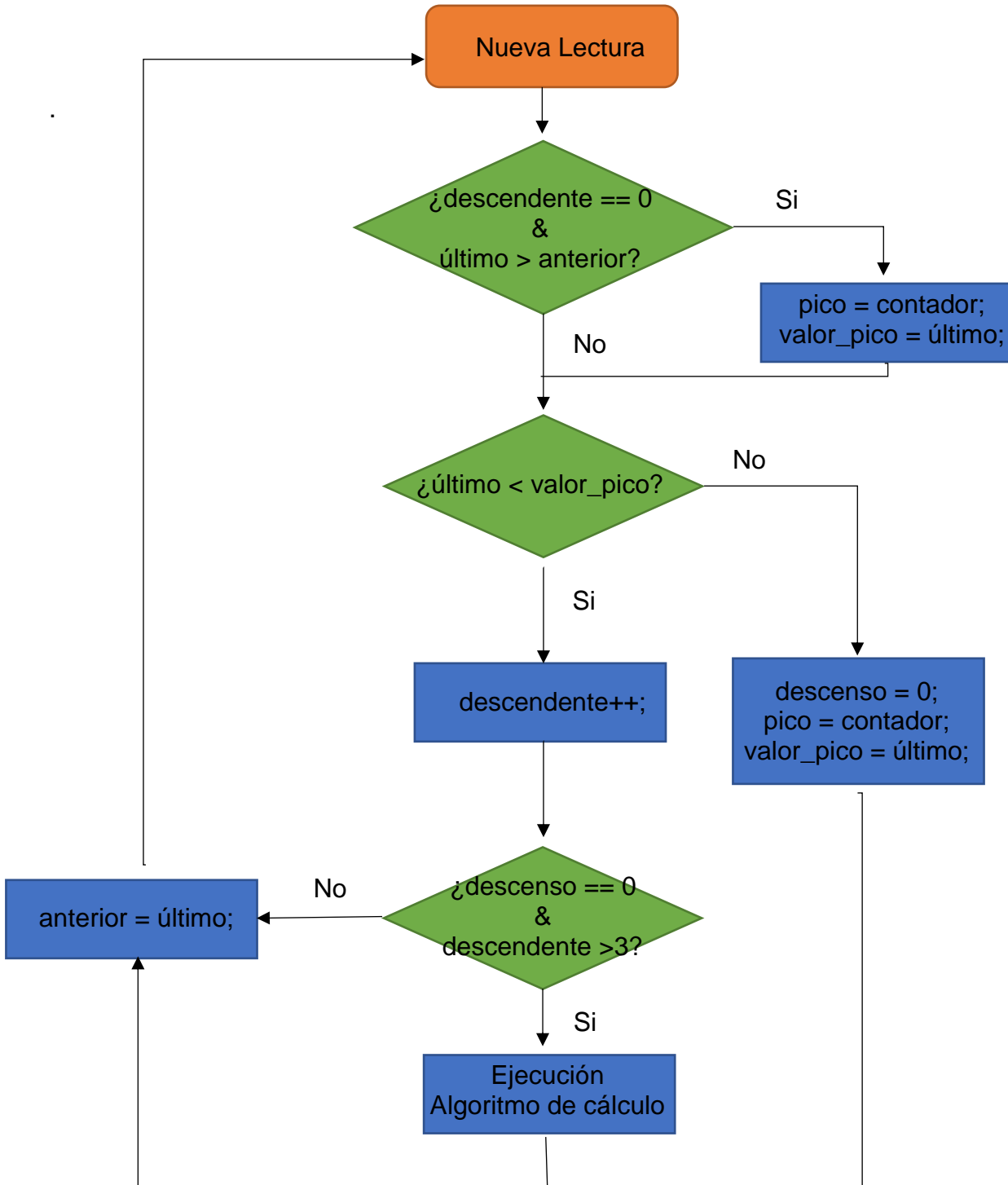


Figura 27. Diagrama de flujo del algoritmo de detección



El diagrama anterior comienza con la obtención de una nueva lectura, esta lectura no es directamente la última lectura del sensor, sino que como ya se mencionó anteriormente, si se detecta la presencia de un dedo o parte del cuerpo a medir, se obtiene una medida cada ciclo de reloj y se almacena en una variable "sensorValor" a la vez que se incrementa un contador "nm". Cuando el contador alcanza un valor definido como "divisiones", se realiza el valor medio como la división de "sensorValor" entre "nm".

Cuando no se detecta dedo u otra parte del cuerpo, se inicializan las variables y se inicia un temporizador, que en caso de llegar a dos minutos, ordenará limpiar el LCD y mostrar un mensaje indicativo con el texto "Coloque el dedo sobre los LED". Para esto, son necesarias dos variables que funcionarán como flags. La primera es la variable "mostrado", la cual se establece a 1 cuando el algoritmo de cálculo se activa y envía datos al LCD. La otra variable es "detección", esta se activará en el primer ciclo en que se detecte presencia de dedo, para lo que es necesario que tanto la variable "mostrado" como "detección" sean 0.

Si no se detecta presencia de dedo, en cada ciclo de reloj irá aumentando un temporizador. En caso de llegar a dos minutos, si la variable "mostrado" es 1, quiere decir que el LCD tiene texto escrito, con lo cual, se limpiará el texto actual y se volverá a mostrar el mensaje indicativo. Ambas variables, se vuelven a poner a 0 tras esta acción.

La variable "detección" aunque influye en esta tarea, ha sido incluida por otro motivo. Su objetivo principal, como ya se ha comentado, es hacer saber al microcontrolador que el dedo está sobre el sensor. Cuando pasa a ser 1, se limpia el LCD y se muestra un texto con el mensaje "Midiendo...".

Lo explicado anteriormente se puede comprender mejor con el diagrama de la Figura 28.

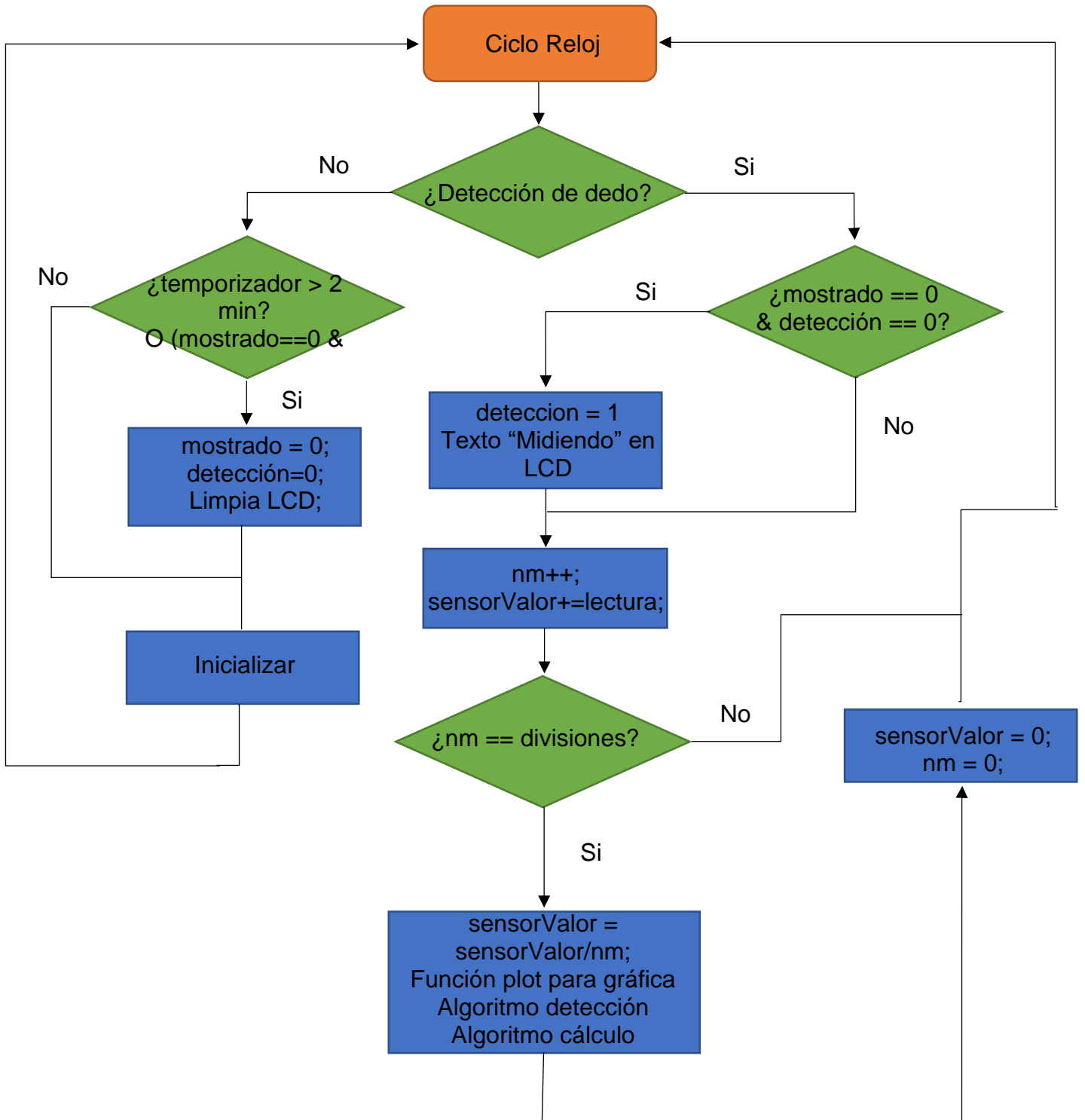


Figura 28. Diagrama de flujo del programa principal.



El algoritmo de cálculo, se ha ajustado y calibrado para este sensor, en condiciones de reposo. Para ello, se ha decidido tomar inicialmente seis pulsos, aunque pueda perder algo de precisión inicialmente, es interesante realizar los primeros cálculos con menos pulsaciones, ya que disminuye el tiempo, de tal manera, que en los primeros segundos se obtiene una primera estimación del pulso. Transcurrido un minuto con el dedo sobre el sensor, la medida de frecuencia cardíaca se realizará con quince pulsos, ya que la obtención de datos se habrá estabilizado e interesa que sea más constante y no fluctúe demasiado. En caso de no hacerse de esta manera, un ritmo cardíaco que fluctúe entre 60 y 64, por ejemplo, estaría modificando el valor a mostrar continuamente, mientras de esta manera, se obtendría un valor medio.

Esta parte del código, requiere conocer el tiempo que ha discurrido entre pico y pico, para lo que recibe el tiempo en que se ha producido el nuevo pico y lo almacena en "pico", así calcula el tiempo como la diferencia de este valor menos el pico anterior, denominado "primer\_pico", y lo almacena en la variable "actual". Inicialmente el tiempo de primero pico será cero, pero en siguientes ciclos, tras el cálculo, se almacena en "primer\_pico" el valor de "pico".

El valor de actual se acumula en una variable "suma", del cual se realiza la media cuando la variable "contador" que aumenta en cada ciclo, alcanza el número definido en la variable "pulsos". Como se indicó anteriormente, "pulsos" es seis durante el primer minuto y después será quince. Para obtener las pulsaciones por minuto, como el dato "suma" no equivale a milisegundos, sino que son ciclos del contador, que realiza un ciclo cada 3,9 milisegundos, se debe multiplicar por 3,9 el valor obtenido. Este nuevo dato, es el periodo, y para conocer el número de veces que se produce en un minuto, simplemente se debe dividir 60000 milisegundos entre dicho valor.

El dato de pulsaciones debe ser un número entero, pero como se realiza una división y no tiene por qué ser exacta, se va a almacenar en una variable float. Este valor debe pasarse a entero, pero si directamente se cambiase el tipo de variable, almacenándose en un int, el valor sería truncado, ya que esta transformación elimina la parte racional. De modo que, la forma más sencilla de realizar un redondeo será sumando 0,5 al valor y cambiando de tipo de variable a int. Por ejemplo, si se hubiera obtenido 54,7, esta suma lo dejaría en 55,2 y el cambio de variable devolvería un valor de 55 pulsaciones por minuto.

Por último, para realizar un último filtro, evitando posibles valores extraños obtenidos por el sensor, se incluyen tres variables que almacenen tres valores, de modo que el que se envíe al LCD para ser mostrado sea el segundo. Este segundo valor, para ser válido, no debe alejarse demasiado del primero y del tercero, para lo que se define una constante "variación" que se ha calibrado para que no aparezcan valores extraños y a su vez no tarde demasiado en devolver un valor. Además, se establecen unos límites, para evitar mostrar valores extraños. Un mínimo de 40 y un máximo de 190.

Como en los casos anteriores, se muestra el diagrama de flujo para una mejor comprensión.



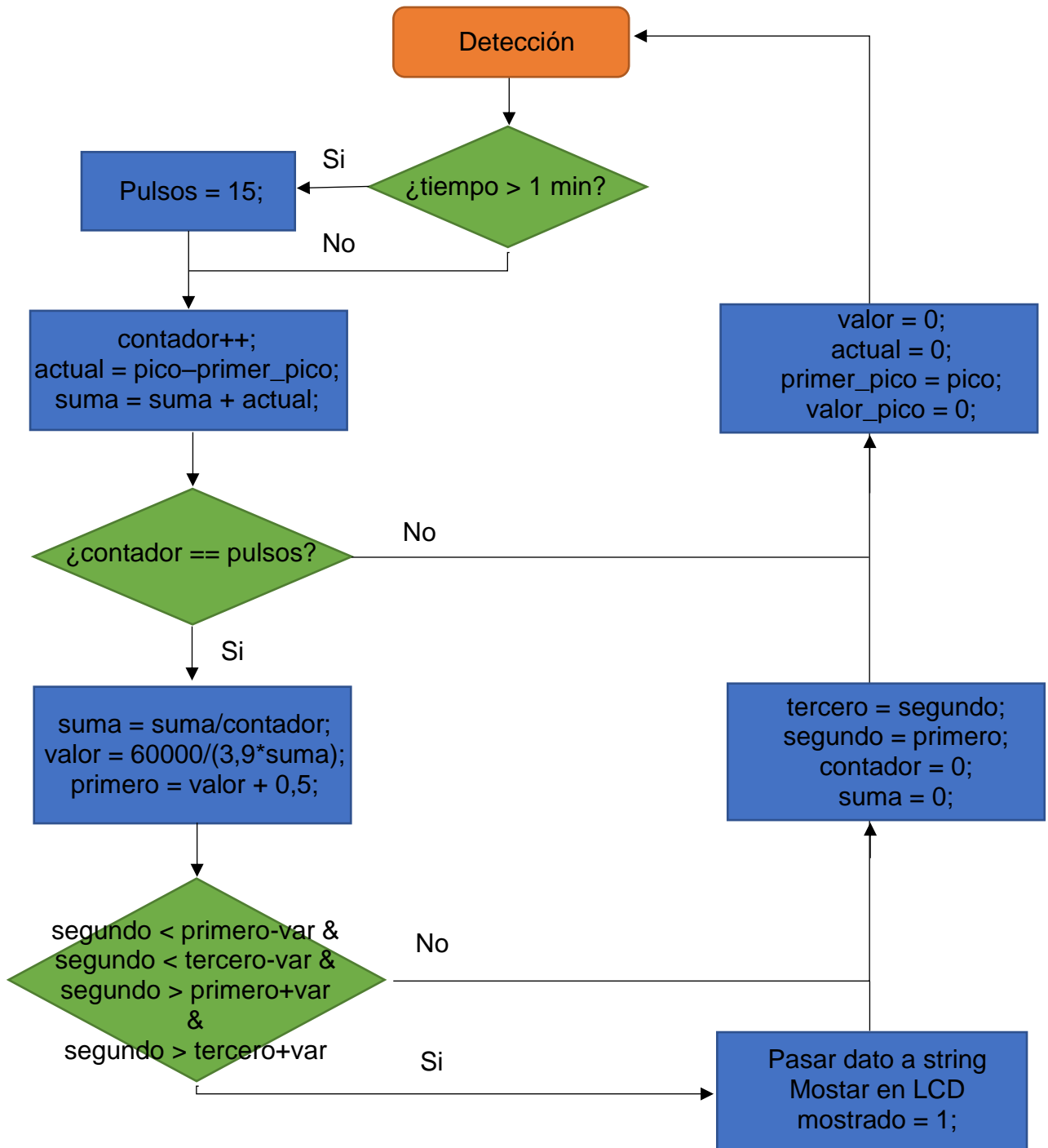


Figura 29. Diagrama de flujo del algoritmo de cálculo.



El dato calculado se convierte en una cadena de caracteres y se envía a la función LCD, donde se limpia el texto actual y se escribe "BPM: más el dato calculado.

Para realizar pruebas, antes de configurar el LCD, se empleó el terminal USART para mostrar el valor calculado, enviando los datos mediante el puerto serie.

Por último, para mostrar la gráfica de valores leídos por el sensor, se envían el contador de ciclos y el valor actual, a la función plot. Esta función, está preparada para enviar los datos mediante UART al puerto serie, donde pueden ser mostrados por una aplicación gráfica, como "SerialPlot". Pero también se ha realizado una función para mostrar la gráfica en la pantalla GLCD de la placa de desarrollo. No se incluye en el código principal, ya que puede dar problemas de comunicación con el display LCD, pero se explicará más adelante.

### 7.3. Algoritmo para la saturación de oxígeno en sangre

Como ya se ha comentado en líneas anteriores, el dispositivo empleado en este proyecto permite conocer la saturación de oxígeno en sangre. Para ello, emplea los dos fotodetectores de los que dispone. El dispositivo devolverá la intensidad luminosa reflejada que llega al fotodetector infrarrojo por un lado y al fotodetector de verde por otro.

El modo en que se va a realizar el cálculo emplea la ley de Lambert-Beer, que relaciona la luz absorbida por la sangre para cada longitud de onda, según la ecuación.

$$R = \frac{I_{\lambda 1}}{I_{\lambda 2}} \quad \text{Ec. 6}$$

Este método es muy sencillo de implementar, ya que simplemente se debe conocer la intensidad captada por los fotodetectores en un mismo momento. De esta manera se obtiene el ratio definido por Lambert-Beer para obtener el porcentaje de saturación de oxígeno en sangre mediante la curva de calibración teórica. Esta curva de calibración, también fue definida por Lambert-Beer, y se muestra en la Figura 30.

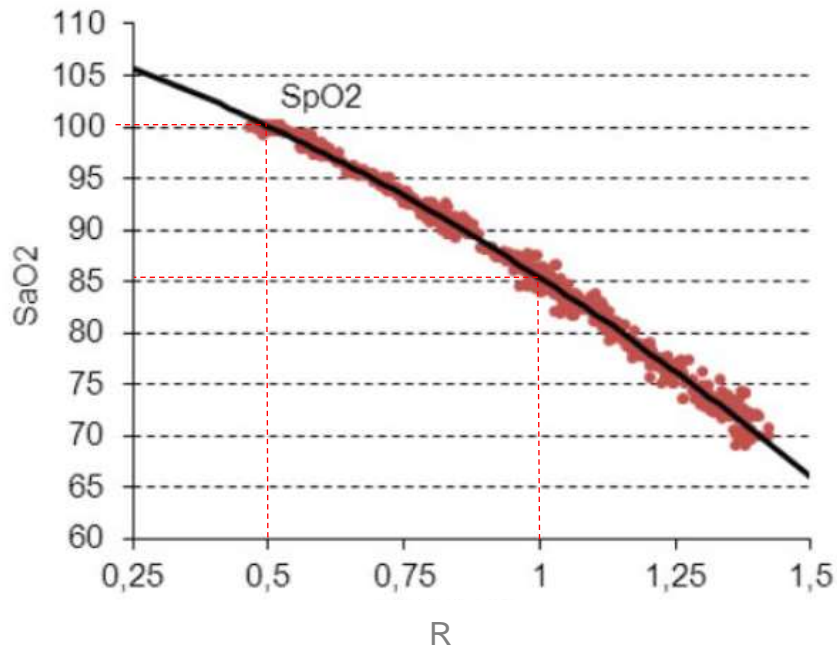


Figura 30. Curva de calibración, ratio-saturación.

Se puede obtener una aproximación lineal de la curva, mediante la ecuación de la recta.

$$SpO2(\%) = b - a \cdot R \quad \text{Ec. 7}$$

De la gráfica podemos obtener dos parejas de valores, con los que calcular las constantes b y a. Para un ratio  $R=0,5$  se obtendría aproximadamente un 100% de saturación, mientras que para un ratio  $R=1$  se obtiene un 85%. Resolviendo el sistema de dos ecuaciones con dos incógnitas, despejamos a y b. [5]

$$100 = a - b \cdot 0,5 \quad \text{Ec. 8}$$

$$85 = a - b \cdot 1 \quad \text{Ec. 9}$$

El resultado de las constantes es  $a=115$  y  $b=30$ , por lo que la ecuación de la recta es la que se muestra a continuación.



$$SpO2(\%) = 115 - 30 \cdot R \quad \text{Ec. 10}$$

Para implementar esta función, solo será necesario conocer el valor de R, el cual es posible conocer mediante el dispositivo “Heart rate 5 click”, empleando la librería anteriormente mencionada.

Colocando el switch del sensor al lado derecho, se habilitará el LED infrarrojo, cuyo valor reflejado puede obtenerse leyendo el fotodiodo correspondiente, con la función “heartrate5\_getAled2val\_led3val” devuelve el valor de lectura en el fotodiodo infrarrojo y mediante “heartrate5\_getAled1val” la lectura del LED verde.

Dividiendo la lectura del LED verde entre la del LED infrarrojo, se obtiene el valor del ratio a aplicar directamente en la ecuación. En la Figura 31 se muestra el diagrama de flujo correspondiente a este código.

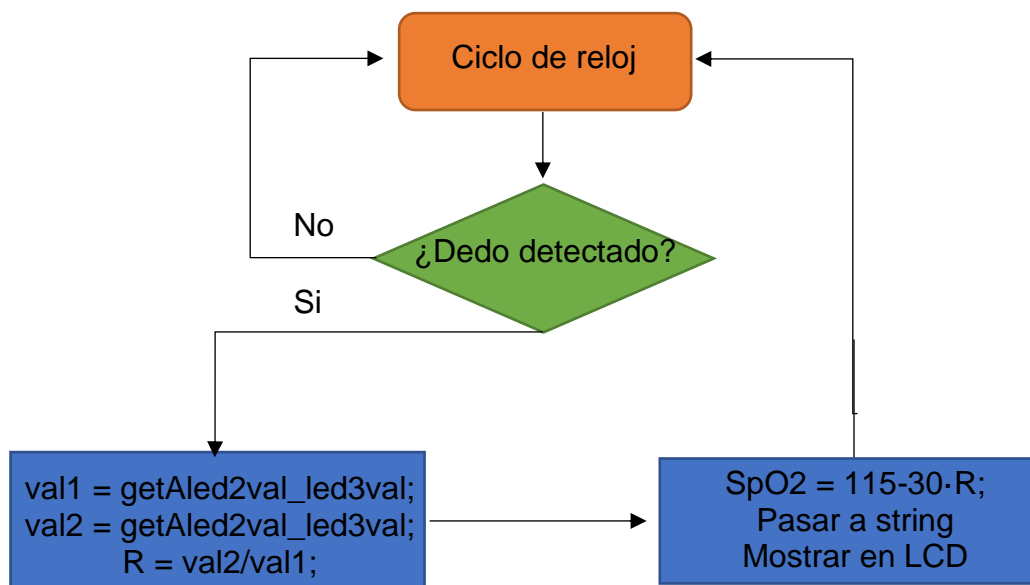


Figura 31. Diagrama de flujo del algoritmo para SpO2.



## CAPÍTULO 8. RESULTADOS OBTENIDOS.

Los resultados obtenidos, mostrados en el display LCD, han sido comprobados mediante otros sensores comerciales, más concretamente, con un pulsioxímetro de dedo que funciona con una configuración de transmisión y un reloj inteligente con detector en configuración de reflexión.

Algunas capturas de los resultados obtenidos se muestran a continuación.



*Figura 32. Imagen de funcionamiento previo a la medida.*

La Figura 32 corresponde al instante inicial, antes de que se coloque el dedo sobre el sensor, aunque también se mostrará este texto cuando haya transcurrido un minuto desde la retirada del dedo.

Una vez colocado el dedo sobre el sensor, el texto mostrado en el display será el que se muestra en la Figura 33.



Figura 33. Imagen durante la detección.

Por último, en la Figura 34 se muestra el resultado obtenido tras la medición.



Figura 34. Imagen de la medida realizada.



La imagen de la Figura 35, compara los valores medidos simultáneamente con el sistema diseñado y con un reloj inteligente. En ella, el ritmo cardíaco varía ligeramente entre ambos. Estas diferencias, pueden ser debidas al número de pulsos que emplea cada uno de ellos para el cálculo de pulsaciones por minuto. Otra de las principales diferencias, es el lugar donde se toma la medida, ya que la muñeca es más ancha y puede dar lugar a mayores errores de lectura, además se debe tener en cuenta el redondeo que realiza el código desarrollado en este proyecto, que puede no ser el mismo que emplea el dispositivo comercial.



Figura 35. Comparación de lecturas de BPM.

A continuación, se muestran otras tres nuevas medidas, en este caso, en imágenes separadas para obtener una mejor calidad, pero tomadas al mismo tiempo. La primera de las medidas se ha tomado en el dedo izquierdo mediante un pulsioxímetro comercial, empleado en enfermería. Este instrumento dispone una configuración en transmisión, la cual permite una exactitud mayor, como se ha indicado en anteriores capítulos. Además, al medir las pulsaciones en el brazo contrario, es posible que las mediciones sean ligeramente distintas.



Figura 36. Medida con pulsioxímetro por transmisión.





En esta segunda imagen, se muestra la medida del dispositivo diseñado, en la que se puede comprobar que el valor es de 70 ppm, dos menos que en la Figura 36.



Figura 37. Medida con el sistema desarrollado.

La imagen de la Figura 38 corresponde a la medida con el reloj inteligente, el cual emplea un detector similar al del presente proyecto, ya que dispone una configuración en reflexión, con dos diodos LED verdes. En esta captura la medida coincide con la de la Figura 37, aunque en las pruebas realizadas se ha comprobado que el valor puede variar ligeramente. Siendo la máxima diferencia captada de 3 pulsaciones por minuto, cuando la medición se ha estabilizado.



Figura 38. Medida con reloj inteligente.



## CAPÍTULO 9. DISEÑO DE UN HARDWARE PROPIO

Lo que se pretende en este apartado es realizar un diseño propio del dispositivo empleado, el cual minimice los costes del Hardware a emplear. Para ello, se propone emplear los mismos componentes que incorpora el sensor de Mikroe, de manera que el Hardware final sea compatible con el código ya preparado.

Se indicarán los componentes necesarios y sus precios unitarios, además del esquema del montaje a realizar.

	Componente	Fabricante	Precio unitario
	SFH7072	OSRAM Opto Semiconductors Inc.	3,22 €
	AFE4404	Texas Instruments	4,07 €
	Placa epoxy doble cara. 100x160mm	Bungard	3,11 €
	Condensador SMD 100 nF $\pm 10\%$	Kemet	0,32 €
	Condensador SMD 100 nF $\pm 10\%$	Kemet	0,14 €
	2 x Resistencia SMD 4K7 OHM $\pm 1\%$	Vishay/Dale	0,31 €
	2 x Resistencia SMD 470 OHM $\pm 1\%$	Vishay/Dale	0,18 €
	Resistencia SMD 470K OHM $\pm 1\%$	Yageo	0,09 €



	2 x 8 Pines	TE Connectivity	0,68 €
	Interruptor SMT SMD 2 filas, 2 posiciones	Nidec Copal	1,58 €

Tabla 1. Componentes para la fabricación del sensor.

El total en material asciende a 13,70 €.

Habría que tener en cuenta también los productos necesarios para la fabricación de la placa, desde los químicos, para eliminar el cobre sobrante de la PCB, hasta las herramientas necesarias para su producción. Además, sería necesaria una aisladora para fijar las pistas en la PCB, herramientas de corte y taladrado, así como equipo para soldar componentes, en especial, soldadura de superficie SMD.

El montaje constará de una placa de doble cara, que permita optimizar el espacio, con pistas por ambos lados de la misma.

Es esencial la colocación de los pines de conexión, J1 y J2 en el esquema de la Figura 35, ya que deben encajar en los módulos mikroBUS. El resto de componentes se pueden colocar a gusto del diseñador, siendo aconsejable su disposición en la cara superior, para que los diodos LED sean accesibles, en la Figura 35 el componente U2.

El componente U3 corresponde al switch de selección de modo. Con ambos interruptores en la primera posición, 2-3 y 5-6, se trabaja en modo detección de ritmo cardíaco. En la segunda posición, 2-1 y 5-4, el modo de trabajo es el de saturación de oxígeno en sangre.

En la Figura 35 se puede observar que las conexiones indicadas anteriormente, habilitan uno de los dos fotodiodos del componente U2.

Un jumper servirá para seleccionar el voltaje de trabajo, 3,3V o 5V. En el presente proyecto se puentó este jumper para trabajar siempre con 3,3V.

El montaje con la resistencia R1 y el LED D1 sirve para indicar que el sensor está alimentado, mediante la iluminación de este diodo.

Del integrado "AFE4404", U1 en la Figura 35, sale tanto la línea de datos como la de reloj, que se conecta al pin correspondiente de J2, para ser transmitido al módulo mikroBUS.

El pin ADC\_RDY será el INT de mikroBUS y los pines TX del protocolo USART, se enrutan por medio de un BUS al detector U2.

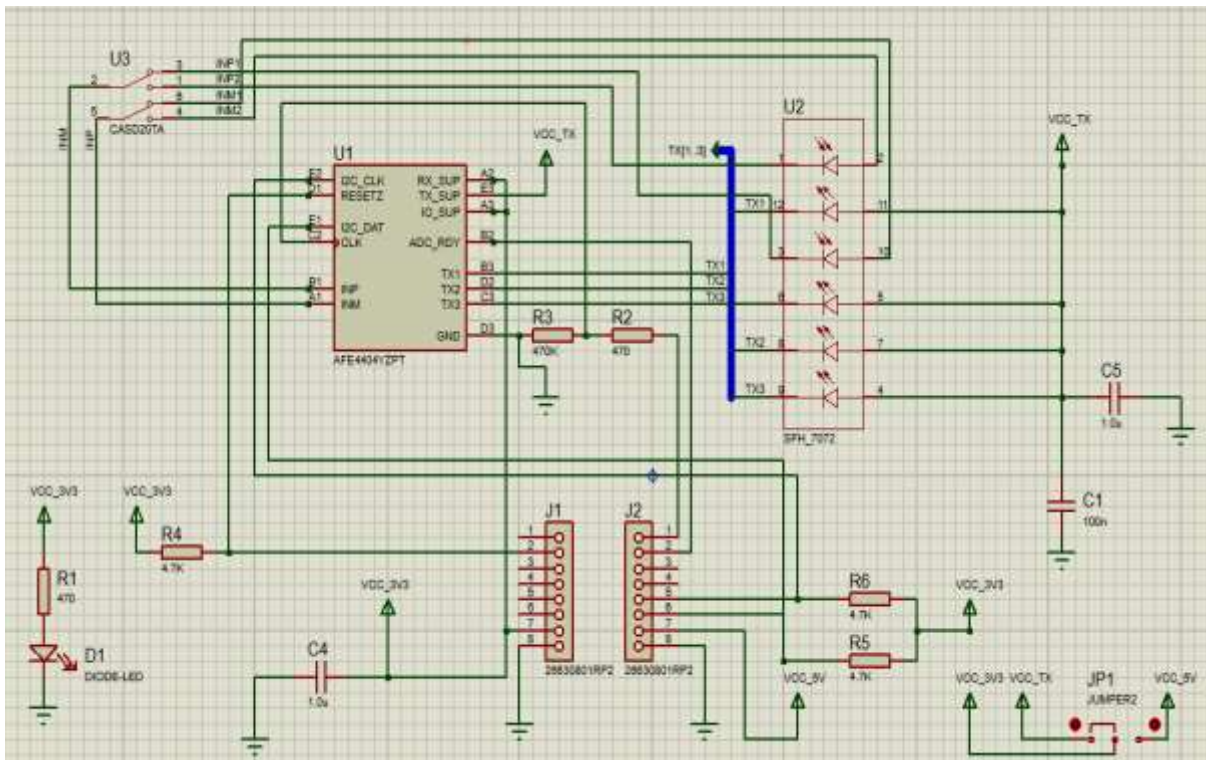


Figura 39. Esquema del montaje de los componentes del sensor.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES



## CAPÍTULO 10. CONCLUSIÓN

El sistema realizado, emplea un sensor biométrico, que permite la obtención de saturación de oxígeno en sangre y ritmo cardíaco, empleando una configuración en reflexión, la cual es más sencilla de colocar a la hora de tomar mediciones sobre el cuerpo humano, ya que permite su colocación en partes anchas del cuerpo, como la muñeca. Esto favorece la monitorización en actividades aeróbicas, como puede ser el ejercicio físico al aire libre.

Los datos obtenidos, se han comparado con sensores comerciales. En concreto con pulsioxímetros de bajo coste empleados en enfermería, que funcionan por transmisión, así como sensores incorporados en pulseras de actividad y relojes inteligentes, los cuales trabajan con una configuración en reflexión. Los resultados obtenidos, son muy similares, incluso se han obtenido valores más precisos en el dispositivo diseñado, esto se ha podido comprobar mediante la captación manual del pulso. Sin embargo, el dispositivo es ligeramente más lento, a costa de precisar más en la medición.

Las principales aportaciones propias de este proyecto, son el desarrollo de los algoritmos para el cálculo del ritmo cardíaco y la saturación de oxígeno en sangre. Estos han sido implementados de tal forma que la lectura sea legible y no varíe en cada instante, ya que la realidad es que los latidos del corazón no son constantes cuando una persona está en reposo, aunque a lo largo de un minuto, sin movimiento, las mediciones no variarán más de dos o tres pulsaciones por encima o por debajo.

Este sensor se ha conectado a un microcontrolador PIC, mediante una placa de desarrollo, la cual facilita la conexión, configuración y calibración. Tras estas pruebas, se podría proponer integrar el microcontrolador PIC, en una PCB junto al resto de componentes que conforman el sensor, haciendo de este dispositivo una buena alternativa a otros sensores comerciales, ya que su precisión es alta y su coste de fabricación no es muy elevado.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES



## CAPÍTULO 11. LÍNEAS FUTURAS

Como posibles líneas futuras se plantean los siguientes puntos.

- Sobre el código actual, realizar un estudio más exhaustivo en cada tipo de paciente, ya que es posible que el sensor se comporte de diferente manera en personas con distinto tipo de piel.
- Adaptar la placa de desarrollo para permitir mostrar simultáneamente el dato numérico de ritmo cardíaco y saturación de oxígeno, junto a la gráfica de la señal del ritmo cardíaco en el display GLCD.
- Diseño del circuito impreso PCB y fabricación del mismo. Revelando las pistas y soldando los componentes, que en su mayoría requieren soldadura superficial SMD.
- Modificación del diseño, para no necesitar un interruptor de selección de modo, sino que se realizará una medición alterna en cada ciclo, mostrando simultáneamente saturación de oxígeno y ritmo cardíaco. Esto se podría conseguir mediante una señal de habilitación desde un pin digital, siendo necesario algún componente físico que sustituya al switch, es decir, que diferencie la señal de los dos fotodetectores, podría ser un relé, que con la señal digital enviada desde el microcontrolador seleccione la señal del fotodiodo infrarrojo o la del LED verde. Almacenando estos valores en cada ciclo, se pueden realizar los algoritmos en ciclos alternos.
- Prescindir de la placa de desarrollo "EasyPIC PRO v7". Mediante la incorporación a la PCB fabricada, de un módulo de alimentación, que podría ser mediante una batería de 3,3V. Requeriría integrar el microcontrolador en el montaje, soldando a la placa el PIC empleado. Además, sería necesaria un pequeño display LCD, donde mostrar los resultados. El PIC se puede programar con el archivo hexadecimal (.HEX) que genera el compilador y empleando algún hardware que permita grabar este archivo en el microcontrolador.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES





## CAPÍTULO 12. ESTUDIO ECONÓMICO

### COSTE DEL PROYECTO REALIZADO

#### Material necesario para la realización del proyecto

<u>Material</u>	<u>Coste</u>
Placa de desarrollo EasyPIC PRO v7	126,17 €
Sensor Heart rate 5 click	27,10 €
Licencia del software MikroC PRO for PIC	227,29 €
<b>Total material</b>	<b>380,56 €</b>

#### TAREAS REALIZADAS

<u>Tarea</u>	<u>Horas empleadas</u>	<u>Coste</u>
Estudio y selección del sensor a emplear	15	450 €
Familiarización con el software MikroC	10	300 €
Análisis de los componentes del sensor	10	300 €
Análisis de la lectura del sensor, sin tratar	20	600 €
Desarrollo del código necesario	150	4500 €
Calibración del sistema	50	1500 €
Redacción y documentación	60	1800 €
<b>Total trabajo</b>	<b>315</b>	<b>9450 €</b>

---

**TOTAL** **9830,56 €**



**Extras** (Fuera del trabajo realizado)

Material Coste

**Total material indicado en el Capítulo 9** **13,70 €**

Tarea Horas empleadas Coste

Estudio de los componentes necesarios 10 300 €

Diseño de la PCB del sensor 30 900 €

Fabricación de la PCB 10 300 €

Pruebas de funcionamiento y calibración 20 600 €

**Total trabajo** **80** **2100€**

---

**TOTAL EXTRAS** **2113,70 €**



## CAPÍTULO 13. BIBLIOGRAFÍA

- [1] The Objective, María Hernández (2020). "El pulsioxímetro, una herramienta clave para detectar la silenciosa neumonía por coronavirus". [Online] [recuperado 07/09/2020]
- [2] John W. Severinghaus, MD, and Yoshiyuki Honda, MD "History of blood gas analysis. VII. Pulse oximetry".
- [3] Tekne Awards Announced. "Star Tribune". [Online] [recuperado 20/07/2020]
- [4] David Lopz Serrano (2017). "Diseño e implementación de un pulsioxímetro". Proyecto de grado. Universidad Politécnica de Valencia.
- [5] Hershel Raff ML. Fisiología médica. "Un enfoque por aparatos y sistemas" [Online] [recuperado 25/07/2020].
- [6] Jorge Alarcó Pérez (2015). "Medida del nivel de saturación de oxígeno en sangre: desarrollo de un pulsioxímetro de bajo coste y comparativa con otros sistemas existentes". Proyecto de grado. Universidad Politécnica de Madrid;
- [7] Wang HY, Qi LW, Wang CZ, Li P (2011). "Bioactivity enhancement of herbal supplements by intestinal microbiota focusing on ginsenosides". Am J Chin Med.
- [8] Eduardo Carletti. "Comunicación - Bus I2C. Descripción y funcionamiento".
- [9] (1996). Universal asynchronous receiver/transmitter (UART) slave device containing an identifier for communication on a one-wire bus. [Online] [recuperado 28/07/2020]
- [10] Jhon Tapia (2008). "Comunicación en Microcontroladores PIC". [Online] [recuperado el 07/09/2020]
- [11] Angulo Duato (2014). "Diseño e implementación de un pulsómetro digital basado en la fotopletismografía". Proyecto de grado. Universidad Politécnica de Valencia.
- [12] Características del sensor "Heart rate click", <https://www.mikroe.com/heart-rate-click>. [Online] [recuperado 15/06/2020]
- [13] Características del sensor "Heart rate 2 click", <https://www.mikroe.com/heart-rate-2-click>. [Online] [recuperado 15/06/2020]
- [14] Características del sensor "Heart rate 3 click", <https://www.mikroe.com/heart-rate-3-click>. [Online] [recuperado 16/06/2020]
- [15] Características del sensor "Heart rate 4 click", <https://www.mikroe.com/heart-rate-4-click>. [Online] [recuperado 16/06/2020]
- [16] Características del sensor "Heart rate 5 click", <https://www.mikroe.com/heart-rate-5-click>.



- rate-5-click. [Online] [recuperado 16/06/2020]
- [17] MikroElektronika (2015). Standard specifications mikroBUS™ standard specifications.
- [18] Datasheet AFE4404 Ultra-Small, Integrated AFE for Wearable, Optical, Heart-Rate Monitoring and Bio-Sensing. <https://download.mikroe.com/documents/datasheets/afe4404.pdf> [Online] [recuperado 17/06/2020].
- [19] MikroElektronika. Dual Power Supply mikroBUS™ sockets The ultimate PIC® board Fast USB 2.0 programmer.
- [20] Red Hat. "El concepto de IDE". [Online] [recuperado el 07/09/2020]
- [21] MikroElektronika. Librería "Heart rate 5 click".
- [22] MikroElektronika. Librería "mikroSDK". Funciones LCD.



## ANEXOS

### ANEXO 1. DESARROLLO DEL CÓDIGO EN MIKROC PRO FOR PIC

/\*

Pulsioxímetro con el sensor Heart Rate 5 click

Fecha : agosto 2020  
Autor : Pedro Vera Gallego

Test configuration PIC :

MCU : P18F87K22  
Dev. Board : EasyPIC PRO v7  
PIC Compiler ver : v7.6.0.0

---

Description :

La aplicación consta de varias partes:

- Inicialización del sistema - Inicializa GPIO, I2C y las estructuras log para comunicar con UART.
- Inicializar variables - ReEstablece el valor por defecto de las variables.
- Inicialización de la aplicación - Configura el microcontrolador para la comunicación e inicializa el dispositivo click. También envía la señal de inicio del temporizador y llama a la función Inicializar, que inicializa las variables.
- plot - Prepara y envía por puerto serie los datos, para ser mostrados gráficamente.
- glcd - Mapea los valores recibidos, para mostrar la gráfica GLCD
- LCD - Limpia el texto que haya en el LCD y escribe el dato enviado.
- Calculo - Realiza la medida del pulso mediante el periodo entre picos.
- Detect - Algoritmo para diferenciar los picos de la señal
- Aplicacion 1 - Procesa los datos recibidos por el sensor y se envían, bien al puerto serie para observar en gráfica, o bien al LCD, previa detección y cálculo.
- Aplicación 2 - Realiza el cálculo de la saturación de oxígeno en sangre.

\*/



```
#include "Click_Heart_Rate_5_types.h"
#include "Click_Heart_Rate_5_config.h"
#include "Timer_Interrupt.h"

#define muestras 4
#define ref_descenso 3
#define divisiones 20
#define variacion 6

float lecturas[muestras], sum;
long int ptr;
float ultimo, valor_pico, anterior, pico, primer_pico;
float actual, valor, suma;
int descenso, deteccion;
int descendente, modo;
int nm, contador;
int i, pulsos;
int primero, segundo, tercero;
uint32_t sensorValor, temporizador;
char textVal[20];
int mostrado;

// Lcd module connections
sbit LCD_RS at LATB4_bit;
sbit LCD_EN at LATB5_bit;
sbit LCD_D4 at LATB0_bit;
sbit LCD_D5 at LATB1_bit;
sbit LCD_D6 at LATB2_bit;
sbit LCD_D7 at LATB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
// End Lcd module connections

sbit LCD_BCK at LATG3_bit;
sbit LCD_BCK_Direction at TRISG3_bit;

char txt1[] = "BPM:";
char txt2[] = "Coloque el dedo";
```



```
char txt3[] = "sobre los LED";  
char txt4[] = "Midiendo...";  
char txt5[] = "SpO2(%):";
```

```
///  
// glcd pinout settings  
char GLCD_DataPort at PORTD;  
  
sbit GLCD_CS1 at RB0_bit;  
sbit GLCD_CS2 at RB1_bit;  
sbit GLCD_RS at RB2_bit;  
sbit GLCD_RW at RB3_bit;  
sbit GLCD_EN at RB4_bit;  
sbit GLCD_RST at RB5_bit;  
  
sbit GLCD_CS1_Direction at TRISB0_bit;  
sbit GLCD_CS2_Direction at TRISB1_bit;  
sbit GLCD_RS_Direction at TRISB2_bit;  
sbit GLCD_RW_Direction at TRISB3_bit;  
sbit GLCD_EN_Direction at TRISB4_bit;  
sbit GLCD_RST_Direction at TRISB5_bit;*/  
  
void sistemalni()  
{  
    //Se configura el módulo de MikroBUS 2  
    mikrobus_gpiolnit( _MIKROBUS2, _MIKROBUS_INT_PIN, _GPIO_INPUT );  
    mikrobus_gpiolnit( _MIKROBUS2, _MIKROBUS_RST_PIN, _GPIO_OUTPUT );  
    mikrobus_i2cInIt( _MIKROBUS2, &_HEARTRATE5_I2C_CFG[0] );  
    mikrobus_logInIt( _LOG_USBUART, 57600 );  
    Delay_ms( 100 );  
}  
  
void Inicializar_variables()  
{  
    ms_counter = 0;  
    sum = 0;  
    ptr = 0;  
    nm = 0;  
    sensorValor = 0;  
    for (i = 0; i < muestras; i++)  
    {  
        lecturas[i] = 0;  
    }  
    ultimo = 0;
```



```
    actual = 0;
    anterior = 0;
    valor = 0;
    descenso = 0;
    descendente = 0;
    primer_pico = 0;
    valor_pico = 0;
    pulsos = 6;
    primero = 0;
    segundo = 0;
    tercero = 0;
}

void aplicacionIni()
{
    //Inicializa el sensor
    heartrate5_i2cDriverInit((T_HEARTRATE5_P)&_MIKROBUS1_GPIO,
(T_HEARTRATE5_P) &_MIKROBUS1_I2C, HR5_ADDR);

    heartrate5_hwReset();
    Delay_ms(100);
    heartrate5_init();
    InitTimer();
    Delay_ms(500);
    Inicializar_variables();
}

void plot(uint32_t ms_count, uint32_t medida)
{
    WordToStr(medida,textVal);
    Ltrim(textVal);
    mikrobus_logWrite(textVal, _LOG_TEXT);
    mikrobus_logWrite(", ", _LOG_TEXT);
    WordToStr(ms_count,textVal);
    Ltrim(textVal);
    mikrobus_logWrite(textVal, _LOG_LINE);
}

void glcd(uint32_t medida)
{
    unsigned pos_x, pos_y;
```





//mapeo de valores, para convertirlos entre 0 y 127 para el eje x y 0 y 63 para el y

```
pos_x++;
if(pos_x > 127)
{
    Glcd_Init();
    Glcd_Fill(0x00);
    pos_x = 0;
}
pos_y = (63*medida)/63000;
Glcd_Dot(pos_x,pos_y,1);
}

void LCD(uint32_t valor_mostrar)
{
    Lcd_Cmd(_LCD_CLEAR);           // limpiar display
    Lcd_Cmd(_LCD_CURSOR_OFF);     // Cursor off

    // Si es modo 1 se muestra BPM, si no SpO2
    if(modo == 1)
    {
        Lcd_Out(1,5,txt1);
    }
    else
    {
        Lcd_Out(1,2,txt5);
    }
    Lcd_Out(1,10,valor_mostrar);
}

void calculo()
{
    // Cuando la bajada pase el número de puntos (ref_descenso) más bajos que el pico
    //se toma el valor de la muestra, que es la diferencia entre el pico actual y el anterior

    descenso = 1;
    contador++;
    actual = pico - primer_pico;
    suma = suma + actual;

    //A partir de 60 segundos, se empieza a hacer la media con 15 pulsos
    if(ms_counter > (60000/3.9))
```



```
{
  pulsos = 15;
}

if(contador == pulsos)
{
  suma = suma/contador;
  //cuenta los picos en 1 minuto, ya que el sensor realiza interrupciones cada 3.9ms
  valor = 60000/(3.9*suma);
  //se añade 0.5 para realizar el redondeo, ya que trunca los decimales
  primero = (int)(valor + 0.5);

  //comprobación de que el valor a mostrar está entre unos límites mayor y menor,
  respecto a los valores adyacentes, así como máximo y mínimo
  if((segundo >= primero-variacion) && (segundo <= primero+variacion) &&
(segundo >= tercero-variacion) && (segundo <= tercero+variacion) && (segundo > 40)
&& (segundo < 190))
  {
    intToStr(segundo,textVal);
    Ltrim(textVal);
    LCD(textVal);
    mostrado = 1;
  }

  //Se almacenan los valores anteriores para descartar en caso de ser un valor muy
distinto
  tercero = segundo;
  segundo = primero;
  contador = 0;
  suma = 0;
}

valor = 0;
actual = 0;
primer_pico = pico;
valor_pico=0;
}

void detect()
{
  sensorValor = sensorValor/nm;
```



```
// Se guarda la nueva lectura en un array para realizar una media
sum -= lecturas[ptr];           //Se sustrae el valor antiguo de la suma
sum += sensorValor;           //Se añade el nuevo valor a la suma
lecturas[ptr] = sensorValor;   //Actualiza el valor antiguo
ultimo = sum / muestras;       //Media de los últimos valores (muestras)

//Muestra los datos del sensor en una gráfica
//plot(ms_counter,ultimo);
/*glcd(ultimo);*/

//si es el pico de la curva, fall_count será 0 y el valor actual será mayor que el
anterior
if((descendente == 0) && (ultimo > anterior))
{
    pico = ms_counter;
    valor_pico = ultimo;
}

//comprobación si el siguiente punto es decreciente
if (ultimo < valor_pico)
{
    descendente++;
    if (descenso == 0 && descendente > ref_descenso)
    {
        calculo();
    }
}
else
{
    // La curva está ascendiendo
    descenso = 0;
    descendente = 0;
    pico = ms_counter;
    valor_pico = ultimo;
}

anterior = ultimo;
ptr++;
ptr %= muestras; //Restaura a 0 ptr si alcanza el valor definido
nm = 0;
sensorValor = 0;
}
```



//Aplicación para el cálculo del pulso

```
void aplicacion1()
{
    if(heartrate5_getLed2_aled2val() < 10)
    {
        //Se pone a cero el temporizador de no detección
        temporizador = 0;

        //mostrar en LCD texto indicativo
        if(mostrado == 0 && deteccion == 0)
        {
            Lcd_Cmd(_LCD_CLEAR);           // limpiar display
            Lcd_Out(1,4,txt4);
            deteccion = 1;
        }

        if(flag==1)
        {
            sensorValor += heartrate5_getAled1val();
            nm++;
            flag=0;
        }
        if(nm==divisiones)
        {
            detect();
        }
    }
    else
    {
        //Se aumenta el temporizador de no detección
        temporizador++;
        //En caso de superar 1 minuto sin detectar presencia, se inicializa sensor y LCD
        if((temporizador > 60000*3.9 && mostrado == 1) || (mostrado == 0 &&
            deteccion == 1))
        {
            deteccion = 0;
            mostrado = 0;
            //mostrar en LCD texto indicativo
        }
    }
}
```



```
Lcd_Cmd(_LCD_CLEAR);           // limpiar display
Lcd_Cmd(_LCD_CURSOR_OFF);     // Cursor off
Lcd_Out(1,1,txt2);
Lcd_Out(2,1,txt3);
}

//Se inicializa todo de nuevo
Inicializar_variables();

}
}

//Aplicación para el cálculo de la saturación de oxígeno

void aplicacion2()
{
    if(heartrate5_getLed2_aled2val() < 10)
    {
        //Se calcula el ratio R
        valor = heartrate5_getAled1val()/heartrate5_getAled2val_led3val();
        //Se calcula la saturación en porcentaje, con la ecuación de la recta
        valor = 115 - (30*valor);
        //Se trunca el valor
        actual = (int)(valor + 0.5);
        //Si el valor es distinto al anterior, se modifica el display
        if(primeros != actual)
        {
            intToStr(actual,textVal);
            Ltrim(textVal);
            LCD(textVal);
            primeros = actual;
        }
    }
    else
    {
        //Se aumenta el temporizador de no detección
        temporizador++;
        //En caso de superar 1 minuto sin detectar presencia, se inicializa sensor y LCD
        if((temporizador > 60000*3.9 && mostrado == 1) || (mostrado == 0 &&
            deteccion == 1))
        {
            deteccion = 0;
        }
    }
}
```



```
        mostrado = 0;
        //mostrar en LCD texto indicativo
        Lcd_Cmd(_LCD_CLEAR);           // limpiar display
        Lcd_Cmd(_LCD_CURSOR_OFF);     // Cursor off
        Lcd_Out(1,1,txt2);
        Lcd_Out(2,1,txt3);
    }

    Inicializar_variables();
}

void main()
{
    sistemaIni();
    aplicacionIni();

    deteccion = 0;

    ANCON0 = 0;           // Condifura los puertos como digitales
    ANCON1 = 0;
    ANCON2 = 0;

    TRISH0_bit = 1;      //Define el puerto H como entrada

    if(Button(&PORTH, 0, 1, 1)){
        modo = 1;
    }
    else modo = 0;

    //Inicializa LCD
    LCD_BCK_Direction = 0;
    LCD_BCK = 1;
    Lcd_Init();
    /*Glcd_Init();*/

    //mostrar en LCD texto indicativo
    Lcd_Cmd(_LCD_CLEAR);           // limpiar display
    Lcd_Cmd(_LCD_CURSOR_OFF);     // Cursor off
    Lcd_Out(1,1,txt2);
    Lcd_Out(2,1,txt3);
}
```



```
while (1)
{
  if(modos == 1)
  {
    aplicacion1();
  }
  else
  {
    aplicacion2();
  }
}
```