

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE BOGOTÁ D.C.

ELECTRÓNICA DIGITAL II

Proyecto Final: VitalSense

Autores:

María J. Morales Villacres David S. Díaz Rivera Julián A. Mancipe Muñoz
mmoralesvi@unal.edu.co ddiazri@unal.edu.co jumancipem@unal.edu.co

Departamento de Ingeniería Eléctrica y Electrónica
Facultad de Ingeniería
Universidad Nacional de Colombia, Bogotá

Índice

1	Introducción	5
2	Planteamiento del problema	5
3	Análisis PESTAL	6
3.1	Político	6
3.1.1	Oportunidad — Impulso gubernamental a la telesalud.	6
3.1.2	Amenaza — Exigencias regulatorias estrictas para dispositivos médicos.	6
3.2	Económico	6
3.2.1	Oportunidad — Creciente demanda por soluciones de monitoreo domiciliario.	6
3.2.2	Amenaza — Vulnerabilidad a factores macroeconómicos e impuestos de importación.	6
3.3	Social	6
3.3.1	Oportunidad — Transformaciones demográficas y cambios en la cultura de autocuidado.	6
3.3.2	Amenaza — Brechas sociales y limitaciones en alfabetización digital.	7
3.4	Tecnológico	7
3.4.1	Oportunidad — Avances en hardware y sensores biomédicos.	7
3.4.2	Amenaza — Dependencia tecnológica y desafíos de precisión.	7
3.5	Ambiental	7
3.5.1	Oportunidad — Sostenibilidad y eficiencia energética.	7
3.5.2	Amenaza — Riesgos asociados a materiales y consumo energético.	7
3.6	Legal	8
3.6.1	Oportunidad — Propiedad intelectual y protección de desarrollos.	8
3.6.2	Amenaza — Cumplimiento normativo y manejo de datos sensibles.	8
4	Antecedentes	8
5	Possibles soluciones	9
6	Objetivos	9
6.1	Objetivo principal	9
6.2	Objetivos secundarios	9
7	Solución propuesta	9
8	Roles	10
9	Desarrollo de la solución	10
9.1	Protocolos de comunicación	10
9.1.1	Protocolo I2C (Inter-Integrated Circuit)	10
9.1.2	Protocolo UART (Universal Asynchronous Receiver/Transmitter)	11
9.2	Sensores	13
9.2.1	Sensor TARJ MAX30102	13
9.2.2	Sensor GY-906	14
9.2.3	Modulo bluetooth HC-05	16
9.2.4	Buzzer	17
9.3	Descripción del hardware	18
9.4	Variables principales del sistema	19
9.4.1	Señales de entrada	19
9.4.2	Señales de salida	19
9.4.3	Variables de estado	19
9.5	Solución en Software	19
9.5.1	Configuración de registros del sensor <i>MAX30102</i>	20
9.5.2	Protocolo de transmisión de datos	20
9.5.3	Implementación de Drivers y Protocolos I2C	21

9.5.4	Funciones Principales del Sistema	21
9.5.5	Controlador del sensor térmico <i>MLX90614</i>	21
9.5.6	Lógica de procesamiento (<code>HR_ProcessSample</code>)	22
9.6	Aplicación	22
10	Posibles mejoras y alcance	26
11	Conclusiones	27
11.1	Notas	27

1. Introducción

En los últimos años, la conciencia colectiva sobre la importancia del cuidado de la salud ha crecido de forma notable. Cada vez más personas buscan monitorear de manera constante ciertos indicadores vitales como el pulso cardíaco, la saturación de oxígeno en la sangre y la temperatura corporal. Las razones son muchas: desde quienes practican ejercicio de forma constante o profesional, hasta quienes simplemente desean llevar un registro básico de su estado de salud como parte de una rutina preventiva.

Este interés creciente ha impulsado el desarrollo y la comercialización de dispositivos especializados capaces de brindar este tipo de información en tiempo real. Sin embargo, como suele suceder con muchas tecnologías, el auge de este mercado también ha traído consigo un incremento considerable en los precios. A pesar de los avances en precisión, comodidad y conectividad, estos dispositivos se vuelven cada vez más inaccesibles para una parte importante de la población. Entonces surge la pregunta: ¿qué opciones tienen aquellas personas que desean cuidar de su salud, pero no pueden acceder a estos productos por su costo? ¿Existe una manera más económica, pero igualmente funcional, de obtener información vital sobre el estado del cuerpo?

2. Planteamiento del problema

En Colombia, las enfermedades cardiovasculares se han consolidado como la principal causa de muerte, representando en 2022 aproximadamente el 23.5 % del total de fallecimientos registrados en el país. Solo ese año, se reportó un aumento del 35 % en la mortalidad por este tipo de patologías en comparación con 2021, superando las 70 000 muertes anuales relacionadas con enfermedades como la hipertensión arterial, los infartos y los accidentes cerebrovasculares [1].

Además, en 2023 se estimó una prevalencia de 10.87 casos de hipertensión arterial por cada 100 habitantes, mientras que la diabetes mellitus y la enfermedad renal crónica presentaron tasas de 3.82 y 1.91 casos respectivamente [2]. Estudios académicos, como una revisión sistemática publicada por *Acta Médica Colombiana*, revelan que alrededor del 24 % de los adultos colombianos sufren de hipertensión, una cifra alarmante que tiende a crecer con la edad y varía por género y región geográfica [3].

Estos datos muestran que un alto porcentaje de la población está en riesgo permanente de sufrir complicaciones cardiovasculares, muchas de las cuales pueden prevenirse si se cuenta con un seguimiento adecuado y frecuente de variables vitales como la presión arterial, la saturación de oxígeno y la temperatura corporal. Sin embargo, el acceso a dispositivos que permitan este tipo de monitoreo sigue siendo limitado por factores como el costo, la disponibilidad y el desconocimiento tecnológico.

En paralelo, el mercado colombiano de dispositivos médicos tipo *wearable* ha mostrado un crecimiento acelerado. En 2023 se estimaron ingresos de USD 222 millones, y se proyecta que para 2030 estos superen los USD 1 233 millones, con una tasa de crecimiento anual (CAGR) de 27.8 % [4]. No obstante, aunque la demanda ha aumentado, la penetración de estos dispositivos sigue siendo baja: en 2019, por ejemplo, solo el 1.6 % de los hogares colombianos contaban con *wearables*, y se espera que para 2025 las ventas alcancen apenas 35 100 unidades a nivel nacional [5].

Esto evidencia una brecha entre la necesidad real de monitoreo de salud y el acceso efectivo a la tecnología que lo permite. Allí es donde cobra valor una solución como la propuesta: un sistema de bajo costo, basado en sensores y tecnologías abiertas, que pueda ofrecer seguimiento básico de indicadores vitales a usuarios que no pueden acceder a *wearables* comerciales. Este tipo de soluciones no solo se alinea con los objetivos de salud pública del país, sino que también representa una oportunidad para democratizar el acceso a herramientas preventivas, mejorar el autocuidado y reducir la presión sobre el sistema de salud.

3. Análisis PESTAL

3.1. Político

3.1.1. Oportunidad — Impulso gubernamental a la telesalud.

El Ministerio de Salud y Protección Social ha consolidado políticas que favorecen la expansión de la Telemedicina, especialmente a partir de la Ley 2015 de 2020, la cual formaliza y regula la prestación de servicios a distancia. Este marco institucional, alineado con la transformación digital del sistema de salud, promueve iniciativas orientadas al monitoreo remoto de pacientes y a la implementación de tecnologías de apoyo clínico. En este contexto, el proyecto se ve fortalecido por la posibilidad de articularse con programas estatales de innovación en salud digital, así como por el acceso potencial a convocatorias de financiación pública destinadas a soluciones que mejoren la continuidad y la accesibilidad del servicio.

3.1.2. Amenaza — Exigencias regulatorias estrictas para dispositivos médicos.

Pese al avance normativo favorable, subsisten barreras regulatorias significativas. El Instituto Nacional de Vigilancia de Medicamentos y Alimentos (INVIMA) mantiene lineamientos rigurosos para el registro y certificación de dispositivos médicos, procesos que suelen ser costosos, prolongados y técnicamente exigentes. Estas condiciones representan un desafío para la eventual comercialización de dispositivos biomédicos emergentes. No obstante, para efectos de este proyecto académico se aclara que el sistema corresponde a un prototipo de investigación y no a un producto médico certificado, por lo que no está sujeto a los requisitos regulatorios aplicables a dispositivos destinados al mercado.

3.2. Económico

3.2.1. Oportunidad — Creciente demanda por soluciones de monitoreo domiciliario.

El aumento sostenido en los costos de la atención hospitalaria ha incentivado a las instituciones prestadoras de salud a buscar alternativas que reduzcan la presión sobre los servicios presenciales. En este contexto, los sistemas de monitoreo remoto representan una opción económicamente atractiva para EPS, clínicas privadas y pacientes con enfermedades crónicas que requieren seguimiento continuo. La transición hacia modelos de atención más eficientes y costo-efectivos abre un nicho de mercado potencial para tecnologías que faciliten el control domiciliario de variables biomédicas. Asimismo, el entorno económico nacional ofrece mecanismos de financiación que pueden respaldar el desarrollo del proyecto, incluyendo recursos del Sistema General de Regalías, convocatorias de MinCiencias, programas de innovación de iNNpulsa Colombia y fondos de capital semilla promovidos por instituciones universitarias, lo cual favorece la sostenibilidad inicial de la iniciativa.

3.2.2. Amenaza — Vulnerabilidad a factores macroeconómicos e impuestos de importación.

A pesar del potencial de mercado, la ejecución del proyecto enfrenta riesgos asociados a la volatilidad económica. La dependencia del dólar para la adquisición de componentes tecnológicos —como la tarjeta Zybo Z7 y diversos sensores biomédicos— incrementa significativamente el costo de producción ante fluctuaciones en la tasa de cambio. A esto se suman los impuestos de importación, la inflación y los sobrecostos logísticos, que pueden elevar el presupuesto total y limitar la escalabilidad del prototipo. Estas condiciones imponen restricciones financieras que podrían afectar la continuidad del desarrollo y dificultar la futura expansión hacia versiones más avanzadas del sistema.

3.3. Social

3.3.1. Oportunidad — Transformaciones demográficas y cambios en la cultura de autocuidado.

El envejecimiento progresivo de la población colombiana ha incrementado la necesidad de sistemas de seguimiento continuo, especialmente en adultos mayores que presentan mayores riesgos clínicos y requieren supervisión frecuente. A este fenómeno se suma la alta prevalencia de enfermedades crónicas como hipertensión, diabetes y patologías cardiovasculares, que demandan mecanismos de control oportuno para prevenir complicaciones. Paralelamente, la amplia penetración de teléfonos inteligentes y el avance de una cultura de prevención en salud

han favorecido la apropiación de soluciones tecnológicas por parte de los ciudadanos. Este entorno social configura un escenario propicio para la adopción de dispositivos biomédicos conectados con aplicaciones móviles, facilitando su uso como herramientas de apoyo al autocuidado y al seguimiento clínico remoto.

3.3.2. Amenaza — Brechas sociales y limitaciones en alfabetización digital.

A pesar de las condiciones favorables, persisten desafíos sociales que pueden dificultar la adopción del dispositivo. Las desigualdades históricas entre áreas urbanas y rurales limitan el acceso equitativo a servicios de salud y a tecnologías de monitoreo, lo que restringe el alcance del sistema en comunidades apartadas. Adicionalmente, la baja alfabetización digital en buena parte de la población adulta mayor representa un obstáculo para la comprensión y el uso adecuado de dispositivos tecnológicos. A ello se suma la desconfianza hacia equipos no certificados o poco conocidos, fenómeno común en poblaciones vulnerables y en usuarios acostumbrados a la atención presencial. Estas barreras podrían disminuir la aceptación inicial del prototipo y requieren estrategias de acompañamiento y pedagogía para garantizar su apropiación efectiva.

3.4. Tecnológico

3.4.1. Oportunidad — Avances en hardware y sensores biomédicos.

El uso de plataformas como la tarjeta Zybo Z7, que combina un SoC FPGA con un procesador ARM, permite el procesamiento de datos en tiempo real y ofrece una gran flexibilidad para el desarrollo de algoritmos personalizados. Esta característica es especialmente relevante para sistemas de monitoreo biomédico, donde la velocidad de adquisición y el procesamiento eficiente son críticos. Adicionalmente, la rápida evolución de sensores biomédicos de bajo costo —como PPG, SpO y medidores de presión arterial— favorece la escalabilidad del prototipo y la mejora continua en la precisión de las mediciones. La convergencia de estas tecnologías ofrece un marco favorable para la innovación y la implementación de dispositivos de monitoreo remoto accesibles y funcionales.

3.4.2. Amenaza — Dependencia tecnológica y desafíos de precisión.

No obstante, la adopción de plataformas importadas, como la tarjeta Zybo Z7 y otros componentes de Xilinx, genera dependencia tecnológica que encarece la producción y limita la capacidad de fabricación local. Esta condición puede dificultar la sostenibilidad y escalabilidad del proyecto en el mediano plazo. Por otro lado, la medición precisa de parámetros críticos, como la presión arterial, constituye un desafío técnico significativo. Garantizar la exactitud de estas mediciones requiere algoritmos robustos de procesamiento de señal y calibración continua, factores que demandan experiencia especializada y recursos adicionales para cumplir con los estándares de confiabilidad en entornos biomédicos.

3.5. Ambiental

3.5.1. Oportunidad — Sostenibilidad y eficiencia energética.

El diseño del dispositivo incorpora estrategias de sostenibilidad desde su concepción, contemplando la correcta gestión de residuos al final de su vida útil. Como aparato electrónico, se propone que los componentes sean reciclables siempre que sea posible, cumpliendo con las disposiciones locales sobre la gestión de Residuos de Aparatos Eléctricos y Electrónicos (RAEE). Además, al ser un dispositivo portátil, el desarrollo del hardware y software se orienta a optimizar la eficiencia energética, implementando modos de bajo consumo que prolonguen la autonomía de la batería y reduzcan la demanda de energía durante su operación. De manera indirecta, el sistema contribuye a disminuir la huella de carbono, al reducir la necesidad de desplazamientos frecuentes hacia centros de salud y fomentar el monitoreo remoto de pacientes.

3.5.2. Amenaza — Riesgos asociados a materiales y consumo energético.

A pesar de estas medidas, el uso de componentes electrónicos importados y de cierta complejidad tecnológica implica potenciales desafíos ambientales. La producción y eventual disposición de estos materiales pueden generar impactos negativos si no se gestionan adecuadamente, incluyendo la liberación de metales y sustancias tóxicas durante el reciclaje informal o inadecuado. Asimismo, la eficiencia energética depende de un diseño

óptimo y de la correcta utilización por parte del usuario; fallas en la programación o en el uso cotidiano podrían incrementar el consumo eléctrico y disminuir los beneficios ambientales esperados. Estas consideraciones subrayan la necesidad de un enfoque integral de sostenibilidad que abarque tanto el diseño como la operación y disposición final del dispositivo.

3.6. Legal

3.6.1. Oportunidad — Propiedad intelectual y protección de desarrollos.

El desarrollo del dispositivo ofrece la posibilidad de generar propiedad intelectual, incluyendo software HDL, firmware, aplicaciones en Python y diseño de hardware. Estos activos pueden ser registrados y protegidos legalmente, sentando las bases para futuras innovaciones y asegurando ventajas competitivas en un eventual escalamiento. La correcta protección de estos desarrollos fomenta la consolidación de conocimientos técnicos y tecnológicos en el contexto académico y profesional, promoviendo el reconocimiento formal de las aportaciones del proyecto.

3.6.2. Amenaza — Cumplimiento normativo y manejo de datos sensibles.

El prototipo, aunque actualmente con fines académicos, debe contemplar el cumplimiento de la legislación vigente en materia de datos personales y seguridad digital. La Ley 1581 de 2012 establece la necesidad de garantizar la seguridad, confidencialidad y el consentimiento informado para el tratamiento de datos sensibles, incluyendo información de salud. De igual manera, la Ley 527 de 1999 regula la firma digital y los mensajes de datos, aspectos críticos para la transmisión y almacenamiento electrónico de la información recolectada. Aunque el dispositivo no requiere registro sanitario en esta fase de investigación, se reconoce que este sería un requisito indispensable en un escenario de comercialización futura, y su cumplimiento sería esencial para garantizar la legalidad y confianza en el sistema.

4. Antecedentes

Las Enfermedades No Transmisibles (ENT), como la hipertensión y los padecimientos cardiovasculares, son una de las principales causas de morbilidad en Colombia. Su manejo efectivo requiere de una monitorización constante y preventiva de signos vitales, que tradicionalmente se limita a entornos clínicos.

Ante este desafío de salud pública, y en el contexto del impulso gubernamental a la Telesalud (Ley 2015 de 2020), surgen las tecnologías portátiles de monitorización como una solución viable. Estas permiten el seguimiento continuo del paciente en su vida diaria, facilitando un modelo de medicina preventiva.

Tecnológicamente, la disponibilidad de plataformas embebidas potentes y asequibles, como la tarjeta Zybo Z7 (SoC FPGA + ARM), y de sensores biomédicos, hace posible el desarrollo de prototipos avanzados. Sin embargo, integrar múltiples sensores (PPG para frecuencia cardíaca/ SpO_2 , temperatura y presión) en un sistema confiable, con una comunicación robusta y un procesamiento de señales preciso, representa una oportunidad de desarrollo e innovación en el campo de la electrónica digital.

Este proyecto se plantea como respuesta a estas necesidades y oportunidades. Surge de la identificación de una brecha entre la tecnología disponible y la accesibilidad a herramientas de monitorización confiables y asequibles. Los antecedentes de este desarrollo se enmarcan en la convergencia de:

- Una necesidad de salud pública: La alta prevalencia de ENT que requieren monitorización constante.
- Un impulso político y regulatorio: La promoción de la Telesalud por parte del gobierno colombiano.
- Una oportunidad tecnológica: La disponibilidad de plataformas de hardware potentes (Zybo Z7) y sensores asequibles que permiten implementar soluciones complejas en el ámbito académico.

Por lo tanto, este proyecto no solo tiene como objetivo cumplir con los requisitos académicos de la asignatura de Electrónica Digital II, sino también sentar las bases técnicas para el desarrollo de un prototipo funcional que

contribuya al ecosistema de innovación en salud digital en Colombia, explorando la viabilidad de integrar múltiples bioseñales en un sistema portátil de bajo costo y comunicarlo de forma inalámbrica para su visualización y análisis.

5. Posibles soluciones

Una posible solución a esta problemática consiste en el diseño y desarrollo de un sistema de monitoreo de signos vitales que utilice componentes electrónicos de bajo costo y tecnologías ampliamente disponibles. El sistema buscará medir variables como la saturación de oxígeno en sangre, la frecuencia cardíaca y la temperatura corporal, permitiendo a los usuarios obtener información relevante sobre su estado de salud en tiempo real. Para ello, se contempla el uso de sensores como el MAX30102, que, mediante comunicación I2C, permitirá recopilar datos precisos sobre las condiciones fisiológicas del usuario. Esta información será transmitida mediante una interfaz UART y conectividad Bluetooth, facilitando su integración con una aplicación móvil. Dicha aplicación será desarrollada en MIT App Inventor, una plataforma accesible y visual que permitirá mostrar los datos recolectados de forma clara e intuitiva en dispositivos móviles, incorporando funciones como alertas en caso de que se detecten valores fuera de los rangos normales. De esta manera, se busca ofrecer una solución funcional, sencilla y de bajo costo que permita a más personas acceder al monitoreo de parámetros vitales básicos, promoviendo el autocuidado de la salud con ayuda de la tecnología.

6. Objetivos

6.1. Objetivo principal

Desarrollar un módulo de monitorización biomédica portátil que, usando la plataforma Zybo Z7, adquiera y procese señales de PPG (frecuencia cardíaca y SpO_2), temperatura corporal y presión arterial, transmitiendo los datos mediante UART/Bluetooth a una aplicación en App Inventor para visualización y generación de alertas.

6.2. Objetivos secundarios

- Implementar los protocolos I2C y UART en los sensores y el módulo Bluetooth.
- Desarrollar una aplicación en App Inventor que muestre datos en tiempo real y emita notificaciones configurables.
- Realizar pruebas experimentales con el dispositivo.

7. Solución propuesta

Una posible solución a esta problemática consiste en el diseño y desarrollo de un sistema de monitoreo de signos vitales que utilice componentes electrónicos de bajo costo y tecnologías ampliamente disponibles. El sistema buscará medir variables como la saturación de oxígeno en sangre, la frecuencia cardíaca y la temperatura corporal, permitiendo a los usuarios obtener información relevante sobre su estado de salud en tiempo real. Para ello, se contempla el uso de sensores como el MAX30105, que, mediante comunicación I2C, permitirá recopilar datos precisos sobre las condiciones fisiológicas del usuario. Esta información será transmitida mediante una interfaz UART y conectividad Bluetooth, facilitando su integración con una aplicación móvil. Dicha aplicación será desarrollada en MIT App Inventor, una plataforma accesible y visual que permitirá mostrar los datos recolectados de forma clara e intuitiva en dispositivos móviles, incorporando funciones como alertas en caso de que se detecten valores fuera de los rangos normales. De esta manera, se busca ofrecer una solución funcional, sencilla y de bajo costo que permita a más personas acceder al monitoreo de parámetros vitales básicos, promoviendo el autocuidado de la salud con ayuda de la tecnología.

8. Roles

1. **Equipo de trabajo:** Diseño del hardware, integración de sensores, desarrollo de los protocolos de comunicación, pruebas, documentación y desarrollo del software.
2. **Docente:** Acompañamiento, guía, retroalimentación y criterios de evaluación.
3. **Proveedor de componentes:** Suministro de sensores, módulos Bluetooth, cables y materiales de montaje.
4. **Sujeto de prueba:** Participar en pruebas controladas para validar el funcionamiento del equipo.

9. Desarrollo de la solución

9.1. Protocolos de comunicación

9.1.1. Protocolo I2C (Inter-Integrated Circuit)

El protocolo I2C (Inter-Integrated Circuit) es un estándar de comunicación en serie desarrollado por Philips Semiconductor (ahora NXP) que permite la interconexión de múltiples dispositivos utilizando solo dos líneas: SDA (Serial Data Line) para la transmisión de datos y SCL (Serial Clock Line) para la sincronización. Es un protocolo ampliamente utilizado en sistemas embebidos, sensores, pantallas LCD y microcontroladores debido a su simplicidad y eficiencia en la comunicación entre dispositivos. I2C funciona bajo un esquema maestro-esclavo, donde un dispositivo maestro controla la comunicación y los esclavos responden según su dirección única asignada. Soporta velocidades de transmisión desde 100 kHz (modo estándar) hasta varios MHz en modos más avanzados; su capacidad de conectar múltiples dispositivos en el mismo bus sin requerir muchas conexiones físicas lo hace ideal para aplicaciones de bajo consumo y dispositivos integrados [6].

La comunicación en I2C sigue una secuencia bien definida controlada por un dispositivo maestro que inicia y gestiona la transmisión de datos hacia uno o más dispositivos esclavos. La secuencia es la siguiente:

- **Estado de reposo (IDLE):** El bus se encuentra en reposo cuando SCL y SDA están en alto, no hay comunicación activa en el bus y los dispositivos están en espera.
- **Condición de inicio (START):** El maestro genera una transición de alto a bajo en SDA mientras SCL está en alto. Esto indica el inicio de una comunicación y que el bus está siendo reservado por el maestro.
- **Envío de dirección del esclavo (ADDRESS):** El maestro envía una dirección de 7 bits o 10 bits (según el estándar) del esclavo con el que quiere comunicarse. A esta dirección se le añade un bit de lectura/escritura (0 para escritura, 1 para lectura), formando un byte de dirección completo. El maestro envía este byte de dirección bit a bit, con el bit más significativo (MSB) primero, sincronizado con la línea SCL.
- **Confirmación (ACK/NACK):** Después de que el maestro envía el byte de dirección, el esclavo cuya dirección coincide responde con un bit de reconocimiento (ACK) bajando la línea SDA durante el noveno ciclo de reloj. Si el esclavo no está presente, está ocupado o no reconoce la dirección, no bajará la línea SDA, resultando en un NACK (No Acknowledge). En caso de NACK, la comunicación se detiene o el maestro puede intentar nuevamente.
- **Transmisión de datos (DATA):** Una vez que se establece la conexión y se recibe un ACK, el maestro o el esclavo (dependiendo del bit de lectura/escritura enviado en la fase de dirección) envía datos de 8 bits. Cada byte de datos se transmite bit a bit, comenzando con el bit más significativo (MSB), y el receptor (ya sea el maestro o el esclavo) envía un ACK después de cada byte para indicar que el dato fue recibido correctamente. Este proceso se repite para cada byte de datos que se desee transferir.
- **Condición de parada (STOP):** El maestro genera una transición de bajo a alto en SDA mientras SCL está en alto. Esto indica el final de la comunicación, libera el bus y lo regresa al estado de reposo, permitiendo que otros dispositivos inicien nuevas comunicaciones.

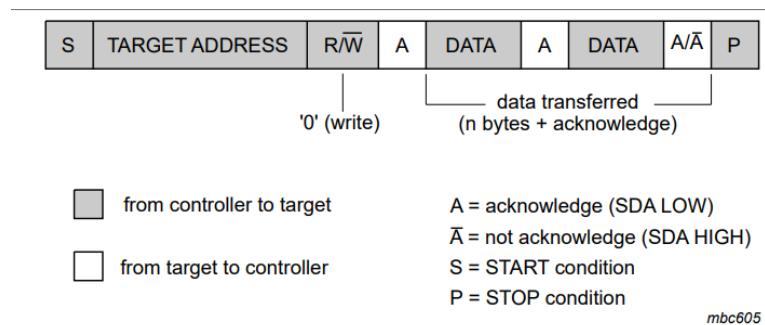


Figura 1: Estructura protocolo I2C. [6]

Además de las operaciones básicas de lectura y escritura, el protocolo I2C soporta funcionalidades avanzadas como la repetición de la condición de inicio (Repeated START condition) para realizar múltiples operaciones sin liberar el bus, lo que es útil en secuencias de lectura-modificación-escritura. También permite la conexión de múltiples maestros en el mismo bus, manejando las colisiones mediante un mecanismo de arbitraje basado en la detección de niveles bajos en las líneas SDA, asegurando que solo un maestro transmita a la vez. La versatilidad del I2C lo convierte en una solución robusta y eficiente para la comunicación de bajo nivel en una amplia gama de aplicaciones electrónicas, por su capacidad de controlar varios dispositivos esclavos con solo dos buses de información [6].

9.1.2. Protocolo UART (Universal Asynchronous Receiver/Transmitter)

El protocolo UART (Universal Asynchronous Receiver/Transmitter) es un estándar de comunicación serie por hardware que se caracteriza por su naturaleza asíncrona, lo que significa que no requiere una señal de reloj compartida entre el dispositivo transmisor y el receptor para la sincronización. Es uno de los protocolos de comunicación de dispositivo a dispositivo más utilizados, comúnmente empleado en sistemas embebidos, microcontroladores y computadoras para la comunicación a nivel de hardware, su simplicidad radica en el uso de solo dos líneas de comunicación: Tx (Transmisor) y Rx (Receptor), lo que reduce la complejidad del cableado y el costo de implementación. Para que la comunicación sea exitosa, tanto el dispositivo transmisor como el receptor deben operar a la misma velocidad de baudios, que es la tasa de transferencia de información. Una discrepancia de hasta el 10 % en esta velocidad generalmente se considera tolerable antes de que surjan errores en el manejo de datos [7].

La transmisión de datos en UART ocurre en forma de paquetes, los cuales incluyen un bit de inicio, una trama de datos, un bit de paridad opcional y bits de parada.

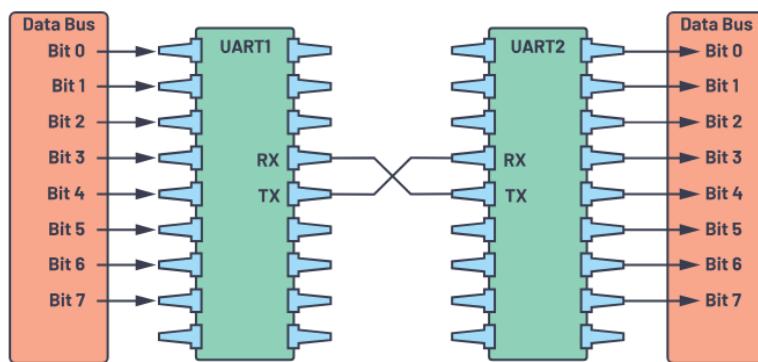


Figura 2: UART con bus de datos. [?]

La secuencia de comunicación en UART se desarrolla de la siguiente manera:

- **Estado de reposo:** La línea de transmisión se mantiene en un estado alto cuando no hay comunicación activa.

- **Bit de inicio (Start Bit):** Para iniciar la transferencia de datos, el UART transmisor baja la línea de transmisión (de alto a bajo) durante un ciclo de reloj. El UART receptor detecta esta transición y entiende que un nuevo paquete de datos está a punto de ser recibido, comenzando a leer los bits a la frecuencia de la velocidad de baudios preestablecida.
- **Trama de datos (Data Frame):** Despues del bit de inicio, se envían los bits de datos reales. La trama de datos puede tener una longitud de 5 a 8 bits. Si se utiliza un bit de paridad, la trama de datos suele ser de hasta 8 bits; si no hay bit de paridad, puede ser de 9 bits. Los datos se envían típicamente con el bit menos significativo (LSB) primero.
- **Bit de paridad (Parity Bit):** Este es un bit opcional que se utiliza para la detección de errores. Se añade al final de la trama de datos e indica si el número de bits '1' en la trama de datos es par (paridad par) o impar (paridad impar). El UART receptor calcula la paridad de los datos recibidos y la compara con el bit de paridad. Si no coinciden, se detecta un error en la transmisión.
- **Bits de parada (Stop Bits):** Para señalar el final del paquete de datos, el UART transmisor lleva la línea de transmisión de nuevo a un estado alto durante una o dos duraciones de bit. Esto le indica al UART receptor que la transmisión del paquete ha terminado y que puede prepararse para el siguiente.

Start Bit (1 bit)	Data Frame (5 to 9 Data Bits)	Parity Bits (0 to 1 bit)	Stop Bits (1 to 2 bits)
------------------------	------------------------------------	-------------------------------	------------------------------

Figura 3: Paquete UART. [7]

Pasos detallados de la transmisión UART:

1. El UART transmisor recibe datos en paralelo desde un bus de datos.
2. A continuación, el UART transmisor añade el bit de inicio, el bit de paridad y el/los bit(s) de parada a la trama de datos.
3. El paquete completo se envía en serie desde el UART transmisor al UART receptor. El receptor muestrea la línea de datos a la velocidad de baudios preconfigurada.
4. El UART receptor descarta el bit de inicio, el bit de paridad y el/los bit(s) de parada.
5. Finalmente, el UART receptor convierte los datos seriales de nuevo a paralelo y los transfiere al bus de datos en su extremo.

El protocolo UART ofrece varias ventajas significativas, entre las que destaca su **simplicidad**, ya que solo requiere dos cables (Tx y Rx) para la comunicación, lo que reduce drásticamente el cableado y, por ende, el costo de implementación. Su **naturaleza asíncrona** elimina la necesidad de una señal de reloj compartida entre los dispositivos, simplificando aún más el cableado; además, su **velocidad es configurable**, permitiendo ajustar la tasa de baudios a diversas velocidades según las necesidades específicas de la aplicación. Para una mayor seguridad, la implementación de un **protocolo de trama** con cabeceras, finalizadores y CRC (Cyclic Redundancy Checking) puede añadir una capa de protección adicional, asegurando la comunicación con el dispositivo correcto y detectando posibles alteraciones accidentales en los datos [7].

Sin embargo, el UART también presenta algunas **desventajas**, pues la ausencia de una señal de reloj compartida implica que ambos dispositivos deben estar de acuerdo en la velocidad de baudios, y cualquier discrepancia significativa puede provocar errores de comunicación. Asimismo, su diseño lo **limita a un esquema de comunicación punto a punto**, es decir, típicamente soporta solo un dispositivo maestro y un dispositivo esclavo, lo que lo hace inadecuado para redes con múltiples dispositivos, por lo que a pesar de su amplio uso, se ha señalado que no siempre está "totalmente optimizado", especialmente en lo que respecta a una implementación rigurosa del protocolo de trama [?].

En cuanto a sus **aplicaciones**, el protocolo UART es muy versátil; se utiliza comúnmente en **depuración**, permitiendo la captura de mensajes de un sistema para la detección temprana de errores durante el desarrollo. En el ámbito de la **fabricación**, es valioso para el rastreo a nivel de función, registrando eventos para monitorear

funcionalidades y alertar a los operadores; también facilita las **actualizaciones de software** para hardware dinámico, siendo una herramienta crucial para el soporte a clientes. Finalmente, es fundamental en **pruebas y verificación** de productos antes de que salgan del proceso de fabricación, garantizando la calidad final [7].

9.2. Sensores

9.2.1. Sensor TARJ MAX30102

El **MAX30102** es un sensor integrado para la medición de *oximetría de pulso (SpO_2)* y *frecuencia cardíaca*. Incluye LEDs Rojo e Infrarrojo, fotodetector, óptica integrada, amplificadores de bajo ruido y filtros que reducen el efecto de la luz ambiente. Se comunica mediante el bus I^2C y posee un *buffer FIFO* interno para almacenar muestras.



Figura 4: Sensor TARJ MAX30102.

Especificaciones Eléctricas

- Tensión de alimentación interna del chip: **1.8V**.
- Tensión típica en módulos breakout: **3.3V–5V**.
- Interfaz de comunicación: **I^2C** .
- Dirección I²C de 7 bits: **0x57**.
- LED Rojo e IR con corriente programable.

Pines del Módulo Breakout

- **VIN / VCC**: Alimentación (3.3V–5V).
- **GND**: Tierra.
- **SDA**: Línea de datos I²C.
- **SCL**: Línea de reloj I²C.
- **INT**: Interrupción (opcional).

Arquitectura Interna y Funcionamiento El MAX30102 incorpora los siguientes bloques funcionales:

- **LED Drivers:** Controlan la corriente de los LEDs Rojo e IR.
- **Fotodetector:** Recoge la luz reflejada por el tejido.
- **Amplificadores y Filtros:** Eliminan ruido y luz ambiente.
- **ADC:** Conversión analógica-digital de alta resolución.
- **FIFO:** Memoria interna para almacenar múltiples muestras.

Modos de Operación El registro **MODE_CONFIG** permite seleccionar:

- **Modo HR:** Solo LED IR para medir frecuencia cardíaca.
- **Modo SpO₂:** LED Rojo + IR para oxigenación.
- **Multi-LED Mode:** Ciclos de muestreo avanzados.

Parámetros Configurables

- **Tasa de muestreo (SPS):** entre 50 y 400 muestras/s.
- **Resolución del ADC:** 15–18 bits según configuración.
- **Corriente de LEDs:** Ajustable según la piel y condiciones de luz.
- **Duración del pulso del LED:** Influye en la energía óptica emitida.

Consideraciones Prácticas

- Evitar movimiento durante la medición.
- Cubrir el sensor para evitar luz ambiente.
- Corrientes altas de LED generan calentamiento.
- No es un dispositivo médico certificado.

9.2.2. Sensor GY-906

El **GY-906** es un módulo basado en el sensor **MLX90614**, un termómetro infrarrojo sin contacto capaz de medir la temperatura de un objeto mediante radiación térmica. El sensor está compuesto por:

- Un termopila infrarroja para detectar radiación IR.
- Un ADC de alta resolución.
- Un procesador digital interno (DSP) con algoritmos de corrección.
- Una interfaz de comunicación **SMBus/I²C**.

Puede medir:

- **Temperatura ambiente (Ta).**
- **Temperatura de objeto (To)** sin contacto.



Figura 5: Sensor GY-906.

Especificaciones Principales

- Rango de medición del objeto: **-70°C a +380°C**.
- Rango de medición ambiente: **-40°C a +125°C**.
- Resolución: **0.02°C**.
- Precisión típica: $\pm 0,5^\circ\text{C}$ (según rango).
- Distancia recomendada: **5 cm – 10 cm**.
- Protocolo: **SMBus** compatible con I²C.
- Dirección I²C por defecto: **0x5A**.

Pines del Módulo

- **VIN**: 3.3V–5V (el módulo incluye regulador).
- **GND**: Tierra.
- **SDA**: Datos I²C.
- **SCL**: Reloj I²C.

Funcionamiento Interno El sensor funciona mediante la detección de radiación infrarroja emitida por objetos.

El MLX90614 utiliza:

- **Termopila IR** para captar energía térmica.
- **Conversión ADC** de alta resolución.
- **Procesamiento digital** para compensación:
 - Corrección de temperatura ambiente.
 - Corrección de no linealidad.
 - Corrección de emisividad (configurable).

Consideraciones de Uso

- Mantener una distancia constante para lecturas estables.
- Evitar superficies reflectivas.
- Permitir tiempo de estabilización térmica al encender.
- No exponer directamente al sol o fuentes térmicas intensas.

9.2.3. Modulo bluetooth HC-05

El **HC-05** es un módulo Bluetooth clásico (Bluetooth 2.0 + EDR) diseñado para comunicación serie inalámbrica entre microcontroladores y dispositivos móviles. Soporta los modos **Maestro** y **Esclavo**, y permite su configuración mediante comandos **AT**.

Es uno de los módulos de comunicación más utilizados para proyectos con Arduino, ESP32, PIC y sistemas embebidos.

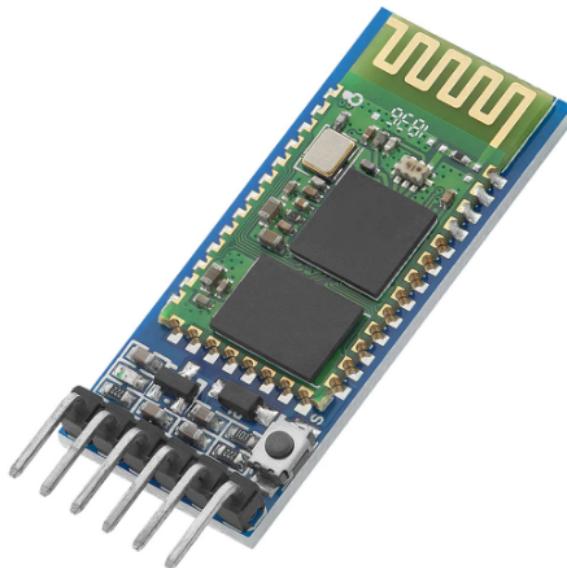


Figura 6: Modulo bluetooth HC-05.

Características Técnicas

- Versión Bluetooth: **2.0 + EDR**.
- Protocolos soportados: **Serial Port Profile (SPP)**.
- Frecuencia de operación: **2.4 GHz**.
- Velocidad de comunicación UART: 9600 bps por defecto (configurable).
- Modos de trabajo: **Maestro** o **Esclavo**.
- Alcance típico: **10 metros**.
- Alimentación en la placa breakout: **3.6V a 6V** (regulada).
- Niveles lógicos UART: **3.3V**.

Pines del Módulo (Breakout)

- **VCC:** 3.6V – 6V.
- **GND:** Tierra.
- **TXD:** Transmisión UART del módulo.
- **RXD:** Recepción UART (3.3V).
- **STATE:** Indica estado de conexión (alto cuando está enlazado).
- **EN / KEY:** Activa el modo de comandos AT (alto = AT Mode).

Funcionamiento Básico El HC-05 crea un enlace serie inalámbrico entre dos dispositivos:

- En modo **Esclavo**: espera a que otro dispositivo se conecte.
- En modo **Maestro**: inicia una conexión hacia otro módulo o dispositivo.

La comunicación se realiza mediante UART:

MCU \leftrightarrow TX/RX HC-05 \leftrightarrow Bluetooth

Proceso de Comunicación UART El flujo típico de comunicación es:

1. Configurar UART del microcontrolador a 9600 bps.
2. Enviar datos por TX hacia RX del HC-05.
3. El HC-05 transmite estos datos por Bluetooth.
4. Si está emparejado, el dispositivo remoto recibe datos instantáneamente.

9.2.4. Buzzer

Para representar la alarma del sistema, se optó por utilizar un módulo zumbador. El funcionamiento de este dispositivo consiste en aplicarle una señal cuadrada con una frecuencia entre 2, kHz y 5, kHz. En el proyecto, se implementó un driver en Verilog encargado de generar dicha señal cuadrada. Este driver recibe como entradas la señal de reloj del procesador y una señal de control, la cual es gestionada por la solución implementada en el software desarrollado.

9.3. Descripción del hardware

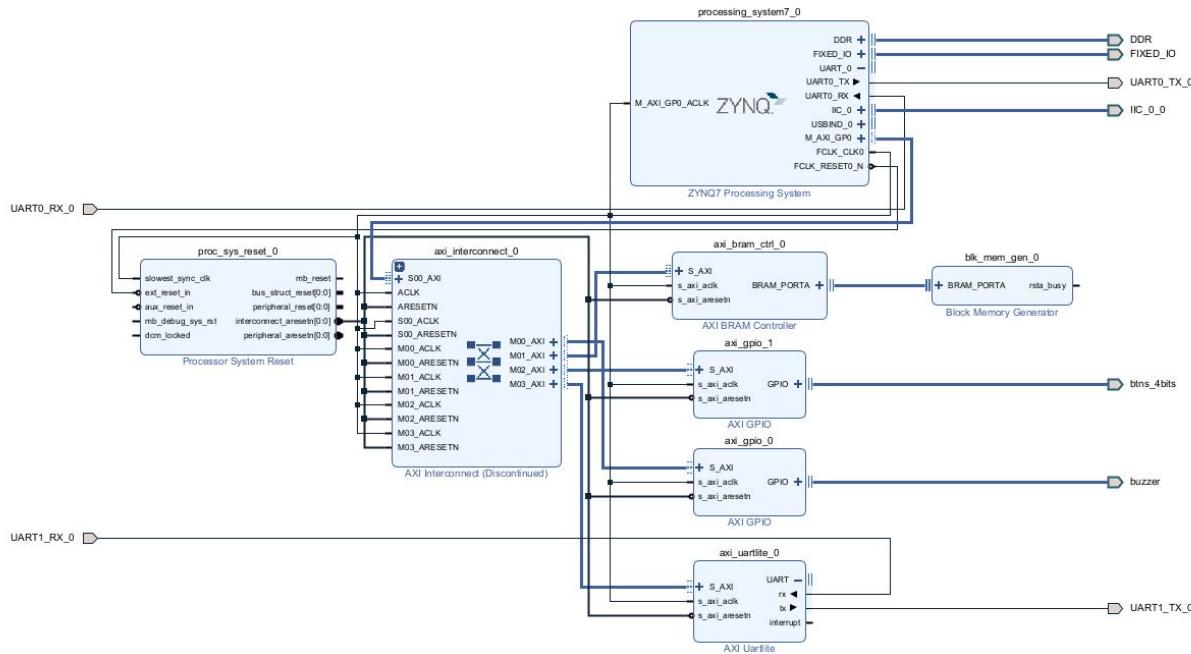


Figura 7: Diagrama de bloques en Vivado.

El diseño del hardware del sistema se basó en la plantilla del procesador utilizada en las prácticas de laboratorio, la cual incluía el *ZYNQ Processing System (PS)* y un módulo *AXI GPIO* destinado únicamente al control básico de LEDs y la lectura de switches. A partir de esta base, se realizaron diversas modificaciones y ampliaciones para implementar la funcionalidad completa del sistema propuesto.

Como se muestra en la Figura 7, el diseño fue extendido mediante la incorporación de periféricos adicionales conectados al bus AXI. En primer lugar, se añadieron módulos *AXI GPIO* dedicados a la lectura de botones físicos y al control de un actuador tipo buzzer, reemplazando el uso de indicadores visuales simples por elementos de interacción y alerta funcionales.

Adicionalmente, a diferencia del diseño de laboratorio, en este proyecto se habilitaron múltiples interfaces de comunicación serial. El controlador **UART0**, integrado en el PS del ZYNQ, se utilizó para la transmisión de información hacia una consola de monitoreo en un PC, permitiendo la visualización detallada y la depuración del sistema en tiempo real. Por otra parte, se incorporó un bloque **AXI UART Lite** en la lógica programable (PL), el cual se destinó a la comunicación con un módulo Bluetooth, empleando un protocolo de datos compacto orientado a aplicaciones móviles.

Asimismo, se activó el controlador **I²C** del ZYNQ PS para permitir la comunicación directa con los sensores externos utilizados en el sistema, específicamente el oxímetro MAX30102 y el sensor térmico MLX90614. Esta interfaz reemplaza el uso exclusivo de señales digitales simples presente en el laboratorio, permitiendo la adquisición de datos fisiológicos de forma precisa y periódica.

En conjunto, estas modificaciones transforman el diseño base de laboratorio en una arquitectura de hardware más robusta y funcional, orientada a la adquisición, procesamiento y transmisión de datos biomédicos, manteniendo la estructura AXI y los mecanismos de reloj y reinicio del sistema original.

9.4. Variables principales del sistema

9.4.1. Señales de entrada

- *red* e *ir*: valores crudos de intensidad de luz roja e infrarroja reflejada por el dedo del usuario. Estas señales son adquiridas desde el sensor MAX30102 mediante el bus I²C y constituyen la base para el cálculo del ritmo cardíaco y la saturación de oxígeno.
- *GPIO_BTNS_DATA*: señal lógica proveniente del bloque de botones físicos. Se utiliza como disparador para iniciar una nueva sesión de medición y reiniciar los contadores del sistema.
- *MLX_REG_TOBJ1* (almacenado en *object*): temperatura superficial del objeto detectado (piel del usuario), expresada en grados Celsius.
- *MLX_REG_TA* (almacenado en *ambient*): temperatura ambiente detectada por el sensor térmico, utilizada como referencia para validar las condiciones de medición.

9.4.2. Señales de salida

- *GPIO_BUZZER_DATA*: señal digital que activa el zumbador cuando el promedio de pulsaciones por minuto (*avg_bpm*) supera el umbral de seguridad establecido (120 BPM).
- *buffer (UART PS)*: trama de datos en formato legible transmitida hacia el PC. Incluye el número de muestra, BPM, SpO₂ y temperaturas.
- *bt_buffer (UART Lite)*: trama compacta destinada al módulo Bluetooth, estructurada como BPM|SpO₂|Temp para facilitar su análisis en la aplicación móvil.

9.4.3. Variables de estado

- *measurement_active*: bandera lógica que indica si el sistema se encuentra en una sesión activa de medición.
- *measurements_taken*: contador de muestras válidas transmitidas durante la sesión actual (máximo 150).
- *is_beat_rising*: variable interna del algoritmo DSP que indica si la señal cardíaca se encuentra en fase ascendente.
- *avg_bpm* y *avg_spo2*: valores fisiológicos estimados mediante promedios móviles para mejorar la estabilidad frente al ruido.

9.5. Solución en Software

En síntesis, el flujo del código implementado para el monitor de signos vitales se basa principalmente en la importación de las librerías del *Board Support Package (BSP)* de Xilinx y las bibliotecas estándar de C. Estas fueron:

```
#include "xparameters.h"
#include "xuartps.h"
#include "xuartlite.h"
#include "xiicps.h"
#include "xil_printf.h"
#include "sleep.h"
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "xil_types.h"
```

Posteriormente se realizaron las definiciones de hardware mediante el mapeo de memoria directo para los periféricos, estableciendo las direcciones base para la UART del sistema de procesamiento (PS), la UART Lite (PL) y el controlador I²C. A su vez, se declararon las variables globales críticas para el procesamiento digital de señales (DSP) y el control de estado:

```

XIicPs IicInstance;
XUartPs UartPs;
XUartLite UartLite;

// Variables DSPfloat
current_dc_ir = 0; float current_dc_red = 0; float avg_bpm = 0; float avg_spo2 = 0;

// Variables de Controlint
measurement_active = 0; int measurements_taken = 0;

```

El flujo principal (*main*) inicializa los controladores de comunicación, configura los sensores mediante comandos I²C específicos y entra en un bucle infinito. Dentro de este bucle, el sistema lee continuamente los datos crudos del sensor óptico, verifica el estado de los botones GPIO para iniciar sesiones de medición y gestiona el reporte de datos hacia el PC y el módulo Bluetooth, activando actuadores (Buzzer) según los umbrales fisiológicos definidos. A continuación, se describe la implementación de los controladores y la lógica de los sensores.

9.5.1. Configuración de registros del sensor *MAX30102*

A diferencia de una configuración por defecto, el sistema sobrescribe los registros internos del sensor de oximetría para adaptarlos a las condiciones de lectura del dedo. En la función *Max30102_Init_Config* se envían comandos específicos mediante I²C para definir la tasa de muestreo, el ancho de pulso y la corriente de los LED. La Tabla 1 detalla los valores hexadecimales cargados en los registros de control y su interpretación física.

MAPA DE CONFIGURACIÓN DEL MAX30102			
Registro	Dirección	Valor (Hex)	Configuración resultante
<i>Mode Config</i>	0x09	0x40	Reset: reinicio por software del sensor.
<i>FIFO Config</i>	0x08	0x4F	Promedio: 4 muestras por lectura. Rollover: activado (sobrescribe datos antiguos).
<i>Mode Config</i>	0x09	0x03	Modo SpO₂: habilita ambos LED (rojo e IR).
<i>SpO₂ Config</i>	0x0A	0x27	Rango ADC: 4096 nA. Tasa de muestreo: 100 Hz. Ancho de pulso: 411 μ s (mayor resolución).
<i>LED1 Pulse</i>	0x0C	0x24	Corriente LED rojo: \approx 7.2 mA (valor 36 decimal).
<i>LED2 Pulse</i>	0x0D	0x24	Corriente LED IR: \approx 7.2 mA (valor 36 decimal).

Tabla 1: Valores de configuración escritos en los registros del MAX30102 durante la inicialización.

9.5.2. Protocolo de transmisión de datos

El sistema actúa como una unidad de transmisión que empaqueta la información fisiológica procesada para dos receptores distintos: una consola de monitoreo (PC) y una aplicación móvil vía Bluetooth. El formato de los datos varía según el canal de comunicación utilizado.

- **Canal UART PS (PC):** diseñado para ser legible por humanos en una terminal serial. La trama incluye etiquetas descriptivas y el estado del contador de muestras.

Formato: "[NNN/150] BPM: XXX | SpO2: XX% | Amb: XX.X | Obj: XX.X\r\n"
 Ejemplo: "[045/150] BPM: 78 | SpO2: 98% | Amb: 24.5 | Obj: 36.2"

- **Canal UART Lite (Bluetooth):** se implementó un protocolo compacto delimitado por el carácter *pipe* (|) para facilitar el análisis de la trama en la aplicación móvil sin transmitir bytes innecesarios.

Formato: "BPM|SpO2|TempObjeto\r\n"
 Ejemplo: "78|98|36.20"

Esta diferenciación permite optimizar el ancho de banda en la conexión inalámbrica, mientras se mantiene una salida detallada y orientada a depuración en la conexión cableada.

9.5.3. Implementación de Drivers y Protocolos I2C

A diferencia de utilizar librerías externas abstractas, el control de los sensores se implementó mediante funciones de escritura y lectura directa sobre el bus I2C, utilizando la instancia `XIicPs`. Para asegurar la correcta comunicación, se definieron las direcciones esclavas predeterminadas de los sensores y los registros de configuración clave, como se muestra a continuación:

DIRECCIONES Y REGISTROS I2C DE LOS SENSORES			
Sensor	Dirección (Hex)	Registro Clave	Descripción
MAX30102	0x57	0x09	Configuración de Modo (SpO_2 : Red + IR)
MAX30102	0x57	0x0C / 0x0D	Amplitud de pulso LED Rojo e IR
MAX30102	0x57	0x07	Lectura de la FIFO de datos
MLX90614	0x5A	0x06	Temperatura Ambiente (T_a)
MLX90614	0x5A	0x07	Temperatura del Objeto (T_{obj})

Tabla 2: Mapa de direcciones y registros principales utilizados en el bus I2C.

9.5.4. Funciones Principales del Sistema

Para modularizar el código, se diseñaron funciones específicas encargadas de la inicialización, la adquisición de datos y el procesamiento matemático. Las funciones más relevantes se describen en la siguiente tabla:

FUNCIONES DE CONTROL Y PROCESAMIENTO	
Función	Descripción
<code>int Max30102_Init_Config(void);</code>	Envía la secuencia de comandos I2C para resetear y configurar los LEDs y la FIFO del oxímetro.
<code>int Max30102_ReadLatestRedIR(u32 *red, u32 *ir);</code>	Lee 6 bytes de la FIFO del sensor y reconstruye los valores de 18 bits de los canales Rojo e IR.
<code>float MLX90614_ReadTemp(u8 RegAddr);</code>	Solicita la temperatura al registro especificado y convierte el dato crudo a grados Celsius.
<code>void HR_ProcessSample(u32 red, u32 ir);</code>	Algoritmo DSP: elimina la componente DC, detecta picos y calcula BPM y SpO_2 en tiempo real.
<code>void Reset_Session(void);</code>	Reinicia los contadores y promedios para comenzar una nueva sesión de 150 muestras.

Tabla 3: Funciones principales implementadas en el archivo `main.c`.

9.5.5. Controlador del sensor térmico **MLX90614**

Al igual que con el oxímetro, la comunicación con el termómetro infrarrojo se implementó de forma directa sobre el bus I²C. Este sensor opera accediendo a su memoria RAM interna para leer los datos de temperatura calculados por su DSP interno.

Para su gestión, se definieron los accesos a las direcciones de memoria específicas del dispositivo, las cuales permiten diferenciar entre la temperatura del chasis del sensor (ambiente) y la temperatura del cuerpo al que apunta (objeto).

REGISTROS DE MEMORIA MLX90614			
Tipo	Macro en código	Dirección RAM	Descripción
Dirección	<code>MLX_I2C_ADDR</code>	0x5A	Dirección esclava del dispositivo
Registro	<code>MLX_REG_TA</code>	0x06	Temperatura ambiente (sensor)
Registro	<code>MLX_REG_TOBJ1</code>	0x07	Temperatura del objeto (paciente)

Tabla 4: Mapa de registros RAM utilizados para la lectura térmica.

La adquisición de datos se centralizó en la función `MLX90614_ReadTemp`, la cual no solo gestiona la transacción I²C (lectura de dos bytes), sino que también realiza la conversión lineal necesaria para traducir el valor crudo de 16 bits a grados Celsius.

$$T (\text{ }^{\circ}\text{C}) = (\text{Datos}_{\text{raw}} \times 0,02) - 273,15 \quad (1)$$

Esta función fue diseñada para ser genérica, aceptando como argumento la dirección del registro (`RegAddr`), lo que permite reutilizar el mismo bloque de código tanto para medir la temperatura ambiente como la corporal.

9.5.6. Lógica de procesamiento (`HR_ProcessSample`)

La función `HR_ProcessSample` implementa el algoritmo de detección de latidos y estimación de parámetros fisiológicos. El procesamiento se realiza en las siguientes etapas:

1. **Filtrado por media móvil:** se aplica un filtro de suavizado de tamaño 4 sobre la señal infrarroja para reducir el ruido de alta frecuencia.
2. **Eliminación de componente DC:** se separa la componente continua de la señal pulsátil mediante un filtro pasa-altos digital:

$$IR_{AC} = IR_{raw} - IR_{smooth} \quad (2)$$

3. **Detección de latidos y cálculo de BPM:** se detectan cruces por umbral dinámico. Al identificar un flanco de subida válido, se calcula el intervalo entre latidos:

$$BPM_{inst} = \frac{60000}{\Delta t (\text{ms})} \quad (3)$$

4. **Cálculo de SpO₂:** se obtiene la relación de radios entre las componentes AC y DC de ambas longitudes de onda:

$$R = \frac{(Red_{AC}/Red_{DC})}{(IR_{AC}/IR_{DC})} \quad (4)$$

$$SpO_2 = 104 - 17R + Offset_{cal} \quad (5)$$

9.6. Aplicación

A continuación se puede observar el desarrollo tanto de la parte visual como de la programación en bloques de la aplicación. Todo esto se ha realizado haciendo uso del programa *MIT APP INVENTOR*.

Para la parte visual, se pensaron inicialmente cuatro pantallas: la de inicio, otra que permita seleccionar el sensor que quiere ver y, otras dos para cada sensor. Sin embargo, durante el desarrollo del proyecto se finalizó la aplicación con tres pantallas: la de inicio, una que permite visualizar los tres indicadores provenientes de los sensores y una de ayuda para que el usuario sepa como se debe conectar el Bluetooth correctamente. Estas se pueden ver en la siguientes imágenes:

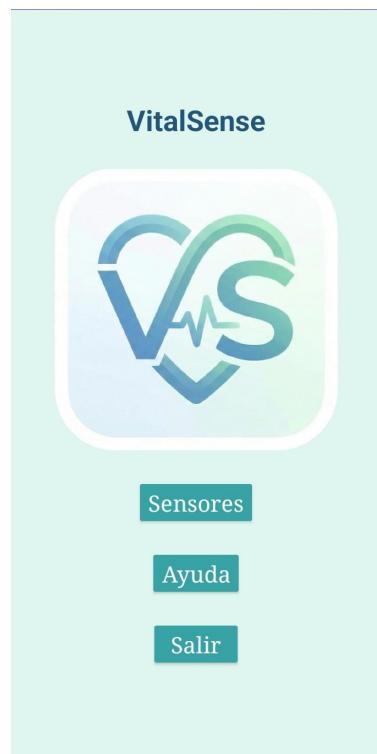


Figura 8: Pantalla de inicio.



Figura 9: Pantalla que permite visualizar los indicadores transmitidos desde la FPGA.



Figura 10: Pantalla que permite ver las indicaciones para una buena conexión del bluetooth.

En figura 9 se podrá apreciar que se desplegará en la pantalla un texto de confirmación cuando el bluetooth haya sido correctamente emparejado con el celular.

Por otro lado, la aplicación aún requiere de un mayor desarrollo, ya que no todas las alertas han sido definidas e implementadas, en el momento solo el indicador de pulso cardíaco tiene alertas definidas mediante las cuales se despliega una notificación en la mitad de la pantalla mostrandole al usuario el posible problema al cual se va a enfrentar o que puede ser normal habiéndose cumplido ciertas condiciones.

El código elaborado para la aplicación consta de una comunicación de las pantallas entre sí, también la configuración del modulo bluetooth y la recepción de los datos transmitidos desde la FPGA. Asimismo, se puede ver en la figura 9 que se incorporaron en una sola pantalla las tres mediciones y se mantuvo la graficación del valor del pulso cardíaco.

La programación elaborada para la aplicación se puede ver en las siguientes imágenes debido a que en *APP INVENTOR* se programa mediante bloques:

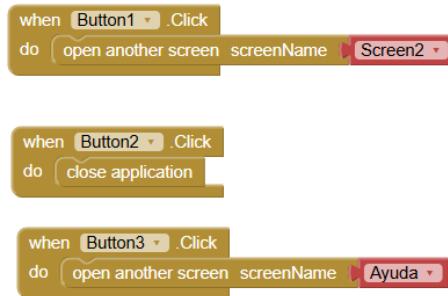


Figura 11: Programación de la pantalla de inicio.

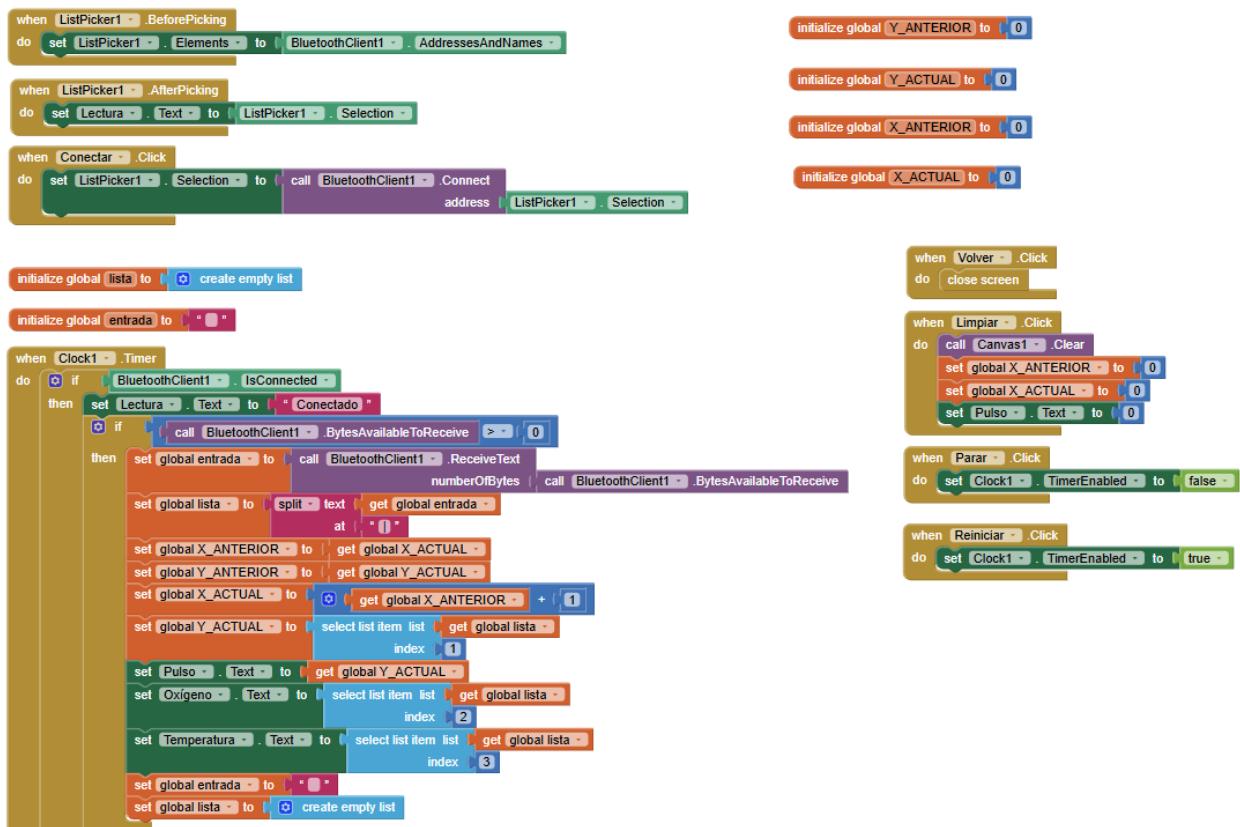


Figura 12: Programación en bloques para la separación de la cadena de texto proveniente de la FPGA, con esto luego se puede refrescar los valores en la aplicación.

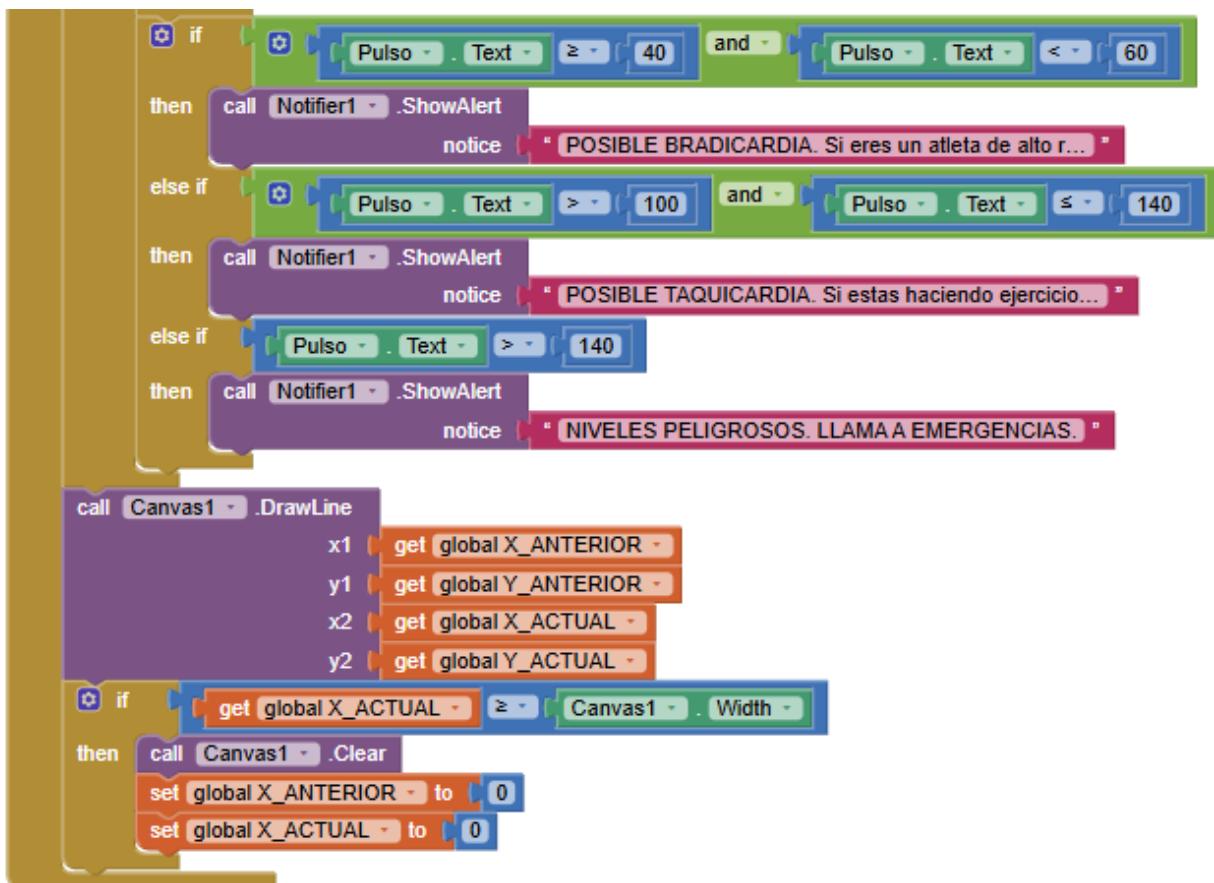


Figura 13: Continuación de la programación en bloques para la impresión de los indicadores.

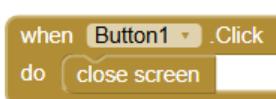


Figura 14: Programación de la pantalla que ayuda a las persona a conectar correctamente el bluetooth.

10. Posibles mejoras y alcance

Una primera línea de mejora se relaciona con la interfaz y funcionalidad de la aplicación móvil. Si bien la aplicación desarrollada en MIT App Inventor cumple con la visualización básica de los datos, su usabilidad podría fortalecerse mediante la inclusión de un sistema de almacenamiento de datos que permitiría llevar un historial de mediciones, lo cual es especialmente útil para identificar tendencias y apoyar procesos de seguimiento médico.

Desde el punto de vista de la calidad y confiabilidad de las mediciones, se propone fortalecer los mecanismos de calibración de los sensores utilizados. Sensores como el MAX30102 y el MLX90614 pueden verse afectados por condiciones ambientales, movimiento del usuario o variaciones fisiológicas, por lo que resulta conveniente incluir rutinas de calibración inicial y procedimientos de validación periódica. De igual forma, la incorporación de algoritmos de filtrado digital y compensación de ruido permitiría mejorar la estabilidad de las señales adqui-

ridas, aumentando la precisión de los parámetros estimados y reduciendo lecturas erráticas.

Desde una perspectiva de escalabilidad y diseño modular, el proyecto podría evolucionar hacia una arquitectura que permita la integración de nuevos sensores biomédicos sin modificaciones profundas en el hardware principal. Por ejemplo, la incorporación futura de sensores de presión arterial o actividad física ampliaría el alcance del sistema y lo haría más competitivo frente a soluciones comerciales. Esta modularidad también facilitaría la adaptación del prototipo a diferentes contextos de uso, como el monitoreo domiciliario o la investigación académica.

Una posible ampliación del alcance del proyecto consiste en aprovechar la comunicación serial existente entre la plataforma Zybo Z7 y un computador para el procesamiento avanzado de los datos adquiridos. Mediante la recepción de los datos biomédicos a través de un puerto serial, es posible desarrollar aplicaciones en entornos de alto nivel como Python o MATLAB que permitan el análisis, almacenamiento y visualización de la información. Esto facilitaría la generación de gráficas en tiempo real, tablas históricas y reportes estadísticos, contribuyendo a un seguimiento más detallado de las variables fisiológicas medidas.

Adicionalmente, el procesamiento de datos en un computador permitiría consolidar y fortalecer el sistema de alertas, superando el esquema actual basado únicamente en umbrales. A través de algoritmos de análisis temporal, sería posible implementar alertas más robustas basadas en tendencias, variaciones sostenidas o la combinación de múltiples parámetros biomédicos. Estas alertas podrían registrarse, visualizarse gráficamente y adaptarse al perfil del usuario, mejorando la capacidad del sistema para identificar condiciones anómalas de manera temprana.

11. Conclusiones

Se concluye que el proyecto VitalSense ha superado exitosamente su fase de prototipado, alcanzando un estado de producto terminado y funcional. La integración entre la plataforma FPGA Zybo Z7, los sensores biomédicos y la interfaz móvil ha demostrado ser robusta y estable en su entrega final. A diferencia de las etapas preliminares, el sistema ahora garantiza:

- Cobertura Total de Mediciones: Se logró la visualización precisa y en tiempo real de todas las variables fisiológicas propuestas (pulso, saturación de oxígeno y temperatura corporal), superando las limitaciones iniciales de código en la interfaz de usuario.
- Sistema de Alertas Activo: El módulo de notificaciones y el sistema de alarma (buzzer) operan correctamente ante valores anómalos, validando la utilidad del dispositivo como herramienta preventiva real.
- Fiabilidad en la Comunicación: La transmisión de datos vía Bluetooth (UART) se mantiene estable y sin pérdidas, asegurando que la experiencia de monitoreo remoto sea fluida y confiable para el usuario final.

El proyecto responde de manera efectiva a una necesidad crítica en Colombia, donde las enfermedades cardiovasculares son la principal causa de muerte (representando el 23.5 % de los fallecimientos en 2022). VitalSense se posiciona como una alternativa de democratización tecnológica, diseñada para cerrar la brecha entre la necesidad de monitoreo constante y el alto costo de los dispositivos wearables comerciales, cuya penetración en los hogares colombianos es muy baja (1.6 % en 2019).

11.1. Notas

Para entender de manera mas clara y sencilla el funcionamiento del dispositivo, puede visitar el siguiente link de video: <https://youtu.be/zWynaepTANG?si=0jQUQD8bfIupDY81>

Para poder visualizar el código fuente desarrollado diríjase a la siguiente carpeta compartida:<https://drive.google.com/drive/folders/1UaNeZ2RTmQH4VWrQL8jNrd3dFxMQBd1c?usp=sharing>

Referencias

- [1] L. Gómez, “Colombia registró un aumento del 35% en muertes por enfermedades cardiovasculares durante el último año,” *Infobae*, 29 de septiembre de 2023. [En línea]. Disponible en: <https://www.infobae.com/colombia/2023/09/29/colombia-registro-un-aumento-del-35-en-muertes-por-enfermedades-cardiovasculares-durante-el-ultimo-año>. [Accedido: 1 de Septiembre del 2025].
- [2] ACHC, “Situación de la enfermedad renal crónica, la hipertensión arterial y la diabetes mellitus en Colombia en el 2023 – ACHC,” *Revista Hospitalaria*, 1 de noviembre de 2024. [En línea]. Disponible en: <https://revistahospitalaria.org/cifras-del-sector/situacion-de-la-enfermedad-renal-cronica-la-hipertension-arterial-y-la-diabetes-mellitus-en-colombia>. [Accedido: 1 de Septiembre del 2025].
- [3] M. S. Zurique Sánchez, C. P. Zurique Sánchez, P. A. Camacho López, M. Sanchez Sanabria y S. C. Hernández Hernández, “Prevalencia de hipertensión arterial en Colombia,” *Acta Médica Colombiana*, 30 de septiembre de 2019. [En línea]. Disponible en: <https://www.actamedicacolombiana.com/ojs/index.php/actamed/article/view/1293>. [Accedido: 19 de Septiembre del 2025].
- [4] Grand View Horizon, “Colombia Wearable Medical Devices Market Size & Outlook,” *Grand View Research*, 2023. [En línea]. Disponible en: <https://www.grandviewresearch.com/horizon/outlook/wearable-medical-devices-market/colombia>. [Accedido: 1 de Septiembre del 2025].
- [5] N. Larocca, “Venta de wearables en Colombia crecerá hasta 35 mil unidades en 2025,” *DPL News*, 10 de junio de 2021. [En línea]. Disponible en: <https://dplnews.com/venta-de-wearables-en-colombia-crecerá-hasta-35-mil-unidades-en-2025/>. [Accedido: 1 de Septiembre del 2025].
- [6] NXP Semiconductors, *I2C-bus specification and user manual*, Rev. 7.0. NXP Semiconductors, 1 de octubre de 2021. [En línea]. Disponible en: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>. [Accedido: 1 de Julio del 2025].
- [7] Analog Devices, “UART: A Hardware Communication Protocol,” Analog Dialogue. [En línea]. Disponible en: <https://www.analog.com/en/resources/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>. [Accedido: 1 de Julio del 2025].
- [8] LibDriver, *MLX90614 Full-Featured Driver Library for General-Purpose MCU and Linux*, GitHub repository, Disponible en: <https://github.com/libdriver/mlx90614>, Accedido el: 3 de noviembre de 2025.
- [9] LibDriver, *MAX30102 Full-Featured Driver Library for General-Purpose MCU and Linux*, GitHub repository, Disponible en: <https://github.com/libdriver/max30102>, Accedido el: 3 de noviembre de 2025.