| | |
|---|---|
| | الجـمـهوريـــة الجـــزائريـــة الديمقـراطيـة الشعبيـــة |
| | People's Democratic Republic of Algeria |
| | وزارة التـعليـم العـالـي والبحث العـلمـي |
| | Ministry of Higher Education and Scientific Research |
| | جـــامـعـــة عبد الحميـد بـن باديـــس – مستـغانـــم |
| | Abdel Hamid Ibn Badis University – Mostaganem |
| | كليـــة العلـــوم والتكنولـوجـــيا |
| | Faculty of Sciences and Technology |
| | قـسم الهندسة الكهربائية |
| | Department of Electrical Engineering |


**N° d'ordre : M2-ESE/GE/2025**

# *MEMOIRE DE FIN D'ETUDES*

# *DE MASTER ACADEMIQUE*

**Filière : Génie Electrique**

**Spécialité : Electronique des systèmes embarqués**

# Thème

**Système embarqué intelligent pour la surveillance des signes vitaux avec un robot compagnon de santé**


Présenté par :

**-ABDELOUHAB IBRAHIM**


Soutenu le /12/ 2025 devant le jury composé de :

| | | | |
|---|---|---|---|
| Président(e) : | HENNI SidAhmed | MCA | Université de Mostaganem |
| Examinateurs : | AARIZOU Meriem | MCB | Université de Mostaganem |
| | Nom et Prénom | Grade | Université de Mostaganem |
| Encadrant (e) : | ABDERRAHMANE Abdelkader | MCB | Université de Mostaganem |

**Année universitaire 2024 / 2025**

# Acknowledgments:

I would like to thank Mr.Abdehahmen Abdelkader, Mr.ABED MANSOUR and the members of the research laboratory of signals and systems (LSS) of the faculty of sciences and technology (FST) for their welcome, support And technical assistance during the finalization of this master's thesis project.

# Table des matières

# CHAPTER I:

# GENERALITY

# I.1 Introduction:

Modern healthcare systems worldwide face unprecedented challenges driven by demographic aging, the rising burden of non-communicable diseases (NCDs), and constrained clinical resources. According to the United Nations, the global population aged 65 and older is projected to increase from 10% in 2022 to 16% by 2050, representing nearly 1.6 billion people (Figure 1.1), thereby intensifying demand for chronic care and long-term support services [1]. Concurrently, NCDs—including cardiovascular diseases, diabetes, chronic respiratory conditions, and cancer—account for 74% of all global deaths, with the majority requiring continuous monitoring rather than episodic interventions [2].

Traditional healthcare delivery models—centered on in-person clinic visits, hospital admissions, and scheduled check-ups—are increasingly misaligned with these epidemiological and demographic realities. These models are inherently reactive, capturing only transient snapshots of patient health while often missing critical physiological trends that evolve between appointments. Geographic barriers further limit access for rural and underserved populations, while elderly individuals frequently resist institutionalization due to preferences for aging in place. Moreover, the escalating cost of frequent in-person care renders it economically unsustainable for both individuals and healthcare systems [3].

The COVID-19 pandemic acted as a catalyst, accelerating the adoption of remote care models and exposing the limitations of facility-dependent care. Telemedicine emerged as a critical tool to maintain care continuity during lockdowns and social distancing mandates. Post-pandemic evaluations indicate that well-implemented telemedicine programs can significantly reduce emergency department utilization—by 30% to 50% in select cohorts of patients with chronic conditions—while preserving or even improving clinical outcomes [4,5]. This crisis underscored the strategic importance of digital health infrastructure and positioned connected care technologies as essential components of resilient, future-ready healthcare systems.

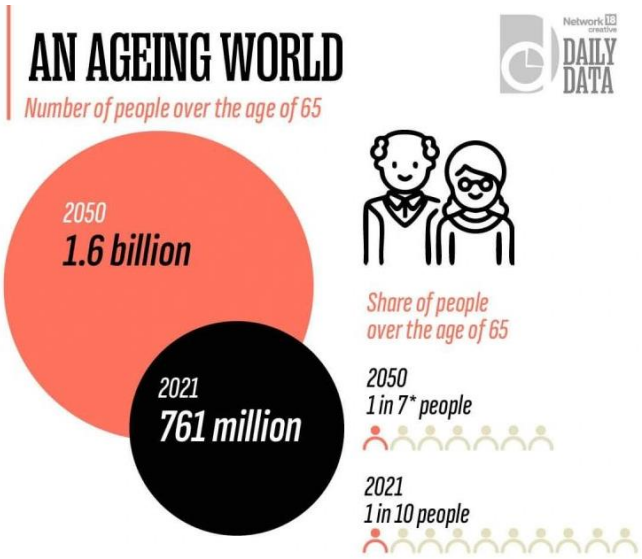



**Figure 1.1**: Global risk weighs: the aging population crisis.

## I.2 The evolution toward preventive and continuous care:

The healthcare paradigm is undergoing a fundamental transformation—from reactive, hospital-centered interventions toward proactive, community-based, and continuous care models. This shift is driven by the need to intercept clinical deterioration before it escalates into emergencies requiring costly hospital resources. Continuous physiological monitoring enables early detection of critical events, such as atrial fibrillation or acute decompensated heart failure, allowing timely intervention that can prevent hospitalization [6].

Evidence from systematic reviews indicates that structured remote monitoring programs can reduce all-cause mortality by approximately 20% and hospital admissions by 25–40% in high-risk populations with chronic conditions—though outcomes vary significantly based on patient selection, technology fidelity, and care team responsiveness. This transition is enabled by the convergence of miniaturized biosensors, embedded intelligence, wireless connectivity, and data-driven analytics, unified under the framework of the Internet of Medical Things (IoMT). The IoMT integrates wearables, smart home devices, and assistive platforms—including robotic systems—into cohesive ecosystems that support aging in place and chronic disease self-management.

Economic imperatives further reinforce this evolution. Remote monitoring pathways have been shown to reduce per-patient annual costs by 20–50% compared to traditional care, primarily through avoided hospitalizations and optimized resource allocation.

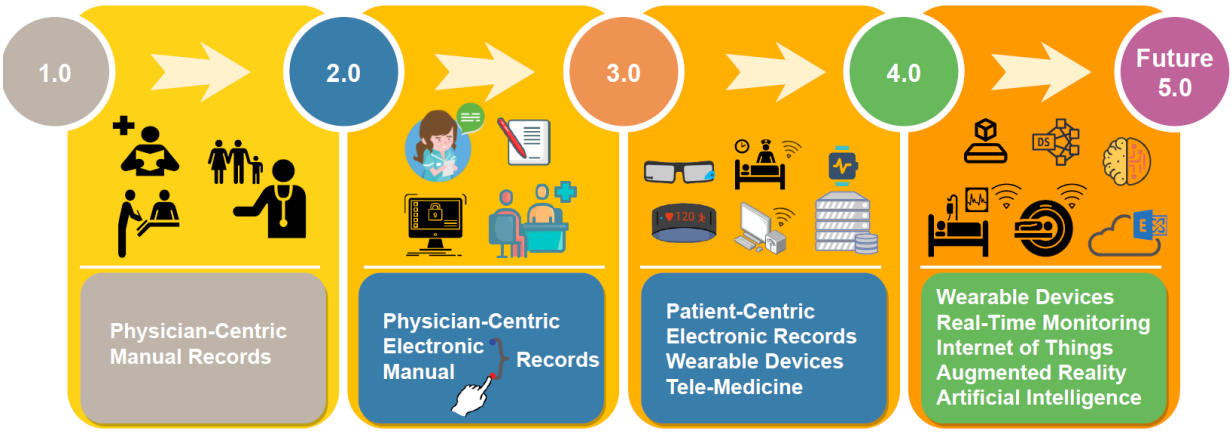



**Figure 1.2**: Evolution of healthcare delivery models.

## I.3 Telemedicine: Definition and Core Capabilities:

Telemedicine is formally defined by the World Health Organization as "the delivery of healthcare services, where distance is a critical factor, by all healthcare professionals using information and communication technologies for the exchange of valid information for diagnosis, treatment, prevention of disease and injuries, research, evaluation, and the continuing education of healthcare providers".

Contemporary telemedicine encompasses four interdependent capabilities:

1. **Teleconsultation**: Real-time (synchronous) or store-and-forward (asynchronous) clinical interactions via audio, video, or messaging platforms.
2. **Remote patient monitoring (telemonitoring)**: Automated, continuous, or periodic transmission of physiological and behavioral data from patients to care teams.
3. **Mobile health (mHealth)**: Use of mobile devices for symptom logging, medication adherence support, health education, and patient engagement.

4. **Specialist tele-services**: Remote interpretation of diagnostic data, including teleradiology, telepathology, and teledermatology.

Rather than operating in isolation, these components form an integrated digital care continuum. For example, anomaly alerts generated by telemonitoring sensors can trigger a teleconsultation, during which clinicians may request high-resolution imaging reviewed via teleradiology—all coordinated through a unified mHealth interface.

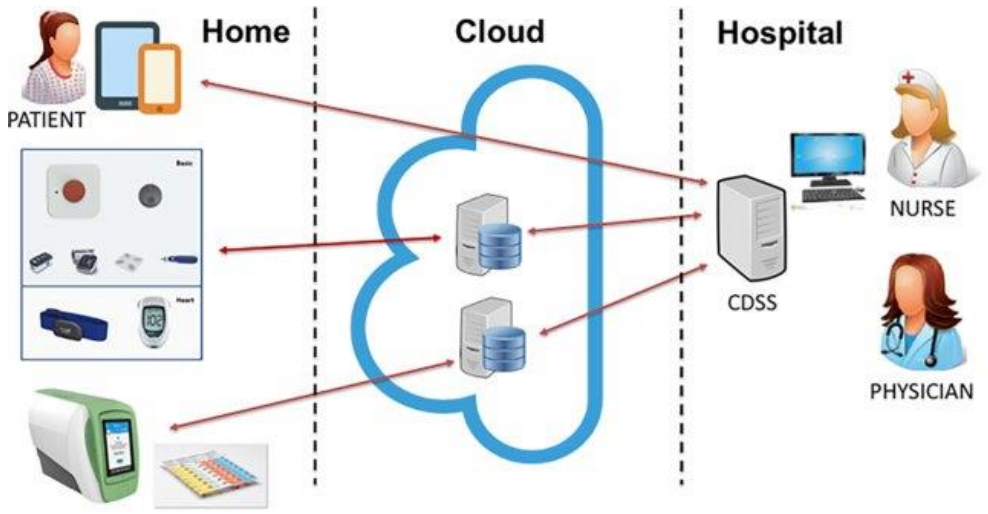



**Figure 1.3**: Functioning of the telemonitoring service.

## I.4 Telemonitoring: Continuous Vital Signs Surveillance:

Telemonitoring constitutes the physiological backbone of modern telemedicine, enabling longitudinal observation of patients outside clinical settings. It is defined as the use of digital technologies to capture and transmit symptom reports and objective physiological data—including heart rate, blood pressure, oxygen saturation ($SpO_2$), and weight—directly to healthcare providers for clinical assessment.

Key monitored domains include:

1. **Cardiovascular**: Heart rate, blood pressure, arrhythmia detection (e.g., via ECG or PPG)
2. **Respiratory**: $SpO_2$, respiratory rate, breath sounds
3. **Metabolic**: Blood glucose, weight trends, hydration markers

4. **Functional**: Physical activity, gait, posture, fall events

Robust clinical evidence supports its efficacy. A landmark meta-analysis of 17 randomized controlled trials involving 3,740 heart failure patients demonstrated that telemonitoring significantly reduced all-cause mortality (relative risk [RR] = 0.80) and heart failure–related hospitalizations (RR = 0.71) [6]. Comparable benefits have been observed in diabetes management (average HbA1c reduction of 0.5–1.0%) and chronic obstructive pulmonary disease (COPD), where telemonitoring reduced exacerbations by approximately 35%.

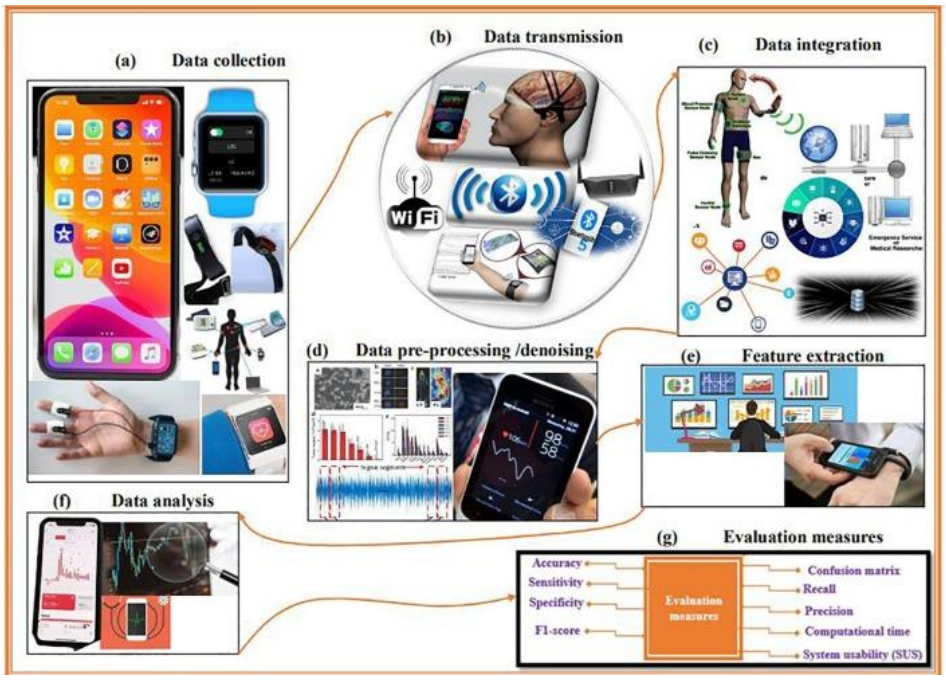



**Figure 1.4**: System architecture for mobile and wearable devices health monitoring.

# I.5 Telemonitoring System Architectures:

Modern telemonitoring systems adopt a **layered architecture** that balances real-time responsiveness, energy efficiency, data security, and clinical usability. This architecture typically comprises four functional tiers:

1. **Edge Layer**: Miniaturized wearable or implantable sensors (e.g., PPG, ECG, accelerometers) acquire raw physiological signals directly from the patient. This layer emphasizes low-power operation, noise resilience, and local preprocessing to reduce bandwidth requirements.
2. **Gateway Layer**: A local computing node—such as a smartphone, dedicated hub, or mobile robot—aggregates data from multiple sensors, performs initial validation and feature extraction, and manages secure communication. This layer often hosts lightweight machine learning models for anomaly detection.
3. **Cloud Layer**: Centralized servers store longitudinal data, execute advanced analytics (e.g., predictive modeling, population health trends), and provide web-based dashboards for clinicians and patients.
4. **Action Layer**: Clinical or automated responses are triggered based on analytical outputs—ranging from SMS alerts and teleconsultation scheduling to robotic assistance or emergency service dispatch.

This design enables **edge-cloud synergy**: time-critical decisions (e.g., fall detection) occur locally for low latency, while non-urgent trend analysis leverages cloud-scale computing. Security is integrated at each layer via authentication, encryption, and role-based access control, aligning with healthcare data protection standards [7].

## I.6 Clinical Evidence and Real-World Impact:

The clinical efficacy of telemonitoring has been evaluated across numerous randomized controlled trials (RCTs), with the strongest evidence in heart failure (HF), diabetes, chronic obstructive pulmonary disease (COPD), and hypertension management.

A comprehensive meta-analysis of 17 RCTs involving 3,740 HF patients confirmed that structured telemonitoring significantly reduces **all-cause mortality** (relative risk [RR] = 0.80) and **HF-related hospitalizations** (RR = 0.71) [7]. In diabetes, telemonitoring of glucose and activity data yields average **HbA1c reductions of 0.5–0.7%** and a **25–30% reduction in emergency**

**department visits**. For COPD, remote symptom and $SpO_2$ tracking reduces exacerbation-related admissions by **approximately 35–39%**, while home blood pressure telemonitoring achieves **systolic reductions of 8–10 mmHg**.

However, outcomes are highly implementation-dependent. The **TEN-HMS trial**, often cited for its reported 45% reduction in a composite endpoint, was **stopped early after interim analysis**, which may have inflated effect sizes—a known bias in prematurely terminated trials [8]. Real-world deployments demonstrate more modest but scalable benefits: the **U.S. Veterans Health Administration's Home Telehealth Program**, serving over 50,000 patients, reports **25% fewer hospitalizations** and average annual savings of **$3,000–$4,000 per patient** through avoided admissions and optimized care coordination [8].

## I.7 Technological Enablers of Modern Telemonitoring:

The clinical success of telemonitoring rests on four interdependent technological pillars:

1. **Miniaturized Biosensors**: Low-cost, high-fidelity sensors such as photoplethysmography (PPG) modules, single-lead ECG patches, and continuous glucose monitors enable unobtrusive, long-term data capture.
2. **Low-Power Wireless Connectivity**: Bluetooth Low Energy (BLE 5.0) supports short-range wearable-to-gateway links, while cellular LPWAN technologies (e.g., NB-IoT, LTE-M) enable direct-to-cloud transmission in rural or robot-based systems.
3. **Edge Computing**: On-device signal processing and lightweight AI (e.g., peak detection, motion artifact rejection) reduce reliance on cloud infrastructure, improving latency and privacy.
4. **AI/ML Analytics**: Supervised and unsupervised models detect anomalies (e.g., atrial fibrillation), predict clinical deterioration, and personalize care pathways using multimodal physiological and behavioral data.

**Photoplethysmography (PPG)** exemplifies this convergence: employed in >80% of consumer wearables, it enables non-invasive estimation of heart rate and $SpO_2$ using optical principles. When integrated with microcontrollers like the **ESP32**—featuring dual-core 240 MHz processing,

520 KB SRAM, and built-in Wi-Fi/BLE [9]—PPG systems achieve medical-grade performance at a component cost below €5, democratizing access to continuous monitoring in low-resource settings.

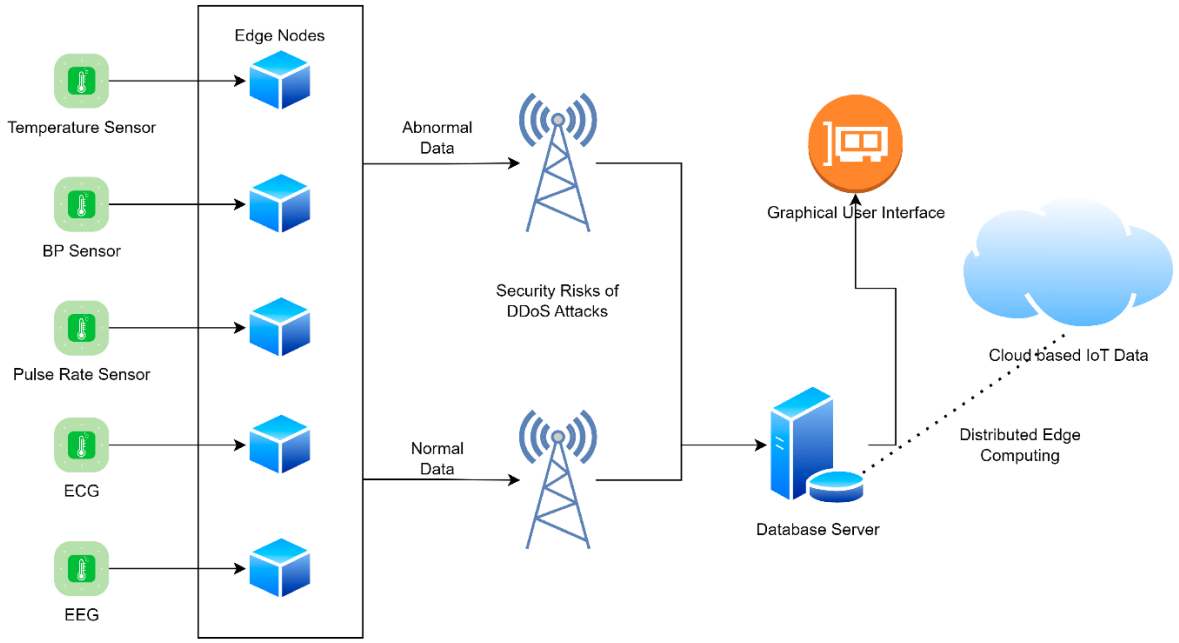



**Figure 1.5**: Telemonitoring technology stack.

## I.8 Wearable Health Devices: From Fitness Trackers to Medical Monitors:

Wearable health devices have evolved from basic activity trackers into sophisticated platforms capable of clinically relevant physiological monitoring. These systems—typically wrist-worn, ring-based, or patch-form—integrate biosensors, low-power microcontrollers, and wireless interfaces to enable continuous, unobtrusive data collection. Global shipments now exceed **1 billion units annually**, reflecting strong consumer and clinical adoption [10].

The transition from consumer wellness tools to **medical-grade devices** hinges on three pillars: **analytical validity** (sensor accuracy against reference standards), **clinical validity** (association with health outcomes), and **regulatory clearance** (e.g., FDA 510(k), CE Class IIa). While early wearables relied on accelerometers for step counting, modern devices incorporate **photoplethysmography (PPG)**, single-lead ECG, temperature, and inertial sensors to estimate heart rate, oxygen saturation ($SpO_2$), heart rate variability (HRV), and arrhythmias.

Notably, the **Apple Heart Study** (n = 419,091) demonstrated that smartwatch-based atrial fibrillation (AFib) detection is feasible at scale, though with **modest positive predictive value (PPV ≈ 0.34)** due to low disease prevalence in the general population—highlighting the risk of overinterpreting sensitivity alone [10]. Nonetheless, when deployed in targeted high-risk cohorts, wearable AFib screening has led to **earlier diagnosis and anticoagulation initiation**, reducing stroke risk.

Common form factors include:

1. **Wrist-worn devices** (e.g., Apple Watch, Fitbit): Highest adoption; integrate PPG + ECG + IMU
2. **Rings** (e.g., Oura, Ultrahuman): Enable 24/7 wear with multi-day battery life
3. **Chest patches** (e.g., Zio XT): Offer clinical-grade ECG for prolonged monitoring
4. **Ear/finger sensors**: Provide higher signal fidelity but limited wearability


**Figure 1.6**: Wearable health device ecosystem.

# I.9 Photoplethysmography (PPG): The Optical Heart of Wearables:

Photoplethysmography (PPG) is the dominant optical sensing modality in consumer and medical wearables, enabling non-invasive estimation of **heart rate**, **$SpO_2$**, and **respiratory rate** through light absorption dynamics in perfused tissue. A PPG signal comprises:

- **DC component**: Baseline absorption from static tissue, venous blood, and non-pulsatile arterial volume
- **AC component**: Pulsatile variation caused by cardiac-induced arterial expansion

The ratio of AC to DC at specific wavelengths (typically **green ~525 nm** for heart rate, **red ~660 nm** and **infrared ~940 nm** for $SpO_2$) allows derivation of physiological parameters. Green light offers high perfusion contrast in superficial capillaries (ideal for wrist wearables), while red/IR penetrates deeper and is less affected by motion—critical for $SpO_2$ accuracy [10].

PPG's success stems from its **low cost (<€1 per sensor in volume)**, **minimal power consumption**, and **ease of miniaturization**—making it ideal for battery-constrained wearables. However, its performance is highly sensitive to **motion artifacts**, **skin tone**, and **fit quality**, necessitating advanced signal processing and multi-sensor fusion for clinical reliability.

## I.10 PPG Sensor Architectures and Configurations:

Wearable PPG systems employ two primary optical geometries:

1. **Transmission PPG** (used in finger/earlobe clips):
    - Light is emitted through thin tissue and detected on the opposite side
    - Offers **high signal-to-noise ratio (SNR)** and is the **clinical standard for $SpO_2$** (±1% accuracy in controlled settings)
    - Limited to anatomical sites with low optical density
2. **Reflection PPG** (used in wristbands, smartwatches):
    - Emitter (LED) and photodetector are placed **adjacent on the same surface**
    - Enables wearability on arbitrary body sites but suffers from **lower SNR** and **higher motion artifact susceptibility** (accounting for ~80% of errors)
    - Modern implementations use **multi-wavelength LEDs** (green + red + IR) and **closely spaced detectors** to improve robustness

Integrated sensor modules like the **MAX30102** (or MAX30105) combine dual/triple LEDs, a photodiode, a **high-resolution (18-bit) analog-to-digital converter (ADC)**, and $I^2C$ communication in a 2.0 × 2.5 mm package—enabling 100+ Hz sampling for real-time heart rate and $SpO_2$ estimation. When paired with an **inertial measurement unit (IMU)** such as the MPU6050, motion artifacts can be adaptively compensated via **sensor fusion algorithms** [10].

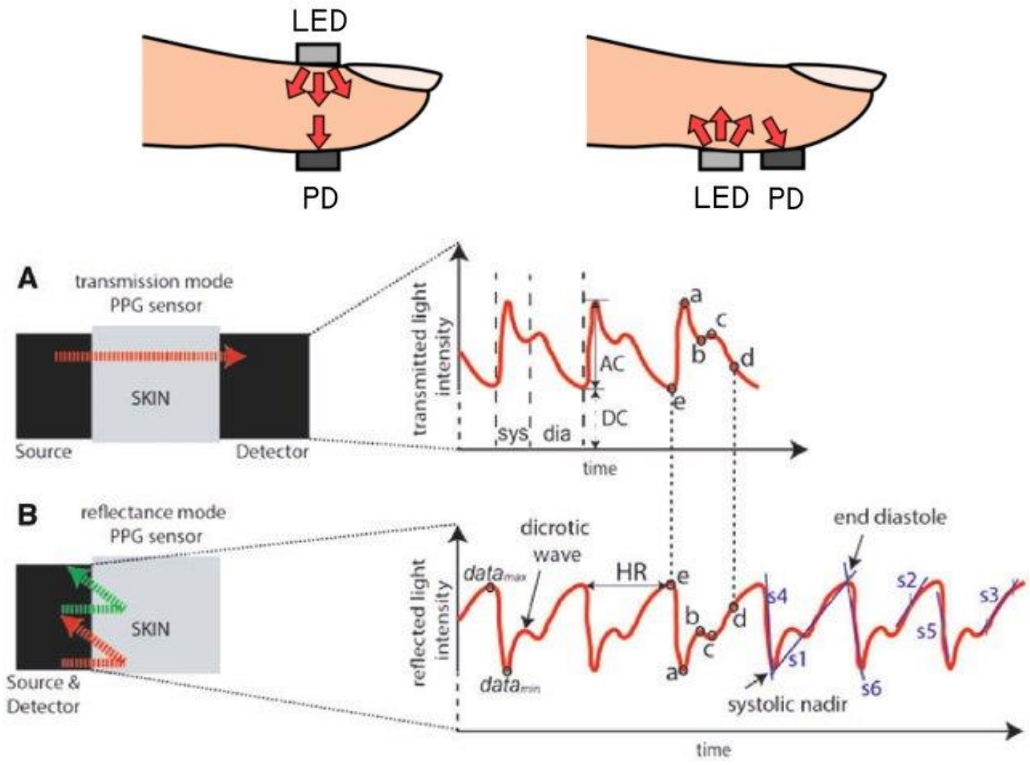



**Figure 1.7**: Comparison of transmission vs. reflection PPG geometries, including light paths and typical applications

# I.11 PPG Signal Processing Pipeline:

Raw photoplethysmography (PPG) signals are highly susceptible to noise from motion, ambient light, and perfusion variability, necessitating a robust multi-stage processing pipeline to extract clinically meaningful features. A typical embedded implementation includes:

1. **Preprocessing**: Bandpass filtering (typically 0.5–8 Hz) removes baseline drift (from respiration or vasomotion) and high-frequency noise (e.g., powerline interference). Adaptive filters may replace fixed thresholds in dynamic conditions.
2. **Signal Quality Assessment**: A Signal Quality Index (SQI) is computed using metrics such as kurtosis, entropy, or autocorrelation to flag unreliable segments. Only high-quality windows proceed to feature extraction.

3. **Peak Detection**: Systolic peaks are identified using adaptive thresholding, derivative-based methods, or lightweight deep learning models (e.g., CNNs trained on annotated PPG datasets).

4. **Feature Extraction**: Inter-beat intervals (IBIs) yield instantaneous heart rate (HR) and heart rate variability (HRV); the AC/DC ratio at red and infrared wavelengths enables $SpO_2$ estimation via the modified Beer-Lambert law.

5. **Physiological Validation**: Outputs undergo plausibility checks (e.g., HR constrained to 40–220 bpm) and artifact rejection before transmission.

On resource-constrained platforms like the ESP32, this pipeline often leverages fixed-point arithmetic, 10–20 ms sliding windows, and real-time FFT or zero-crossing methods to achieve sub-10 ms latency. Advanced implementations may offload complex tasks (e.g., HRV spectral analysis) to a gateway or robot-level processor to balance accuracy and power consumption [11].

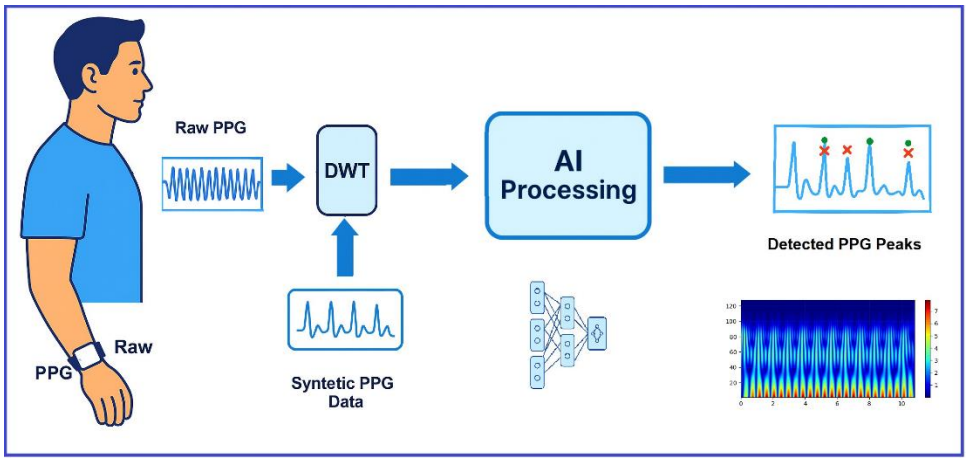



**Figure 1.8**: Complete PPG processing chain.

## I.12 Wearable PPG Performance and Limitations:

While PPG enables non-invasive monitoring, its real-world accuracy is context-dependent and often overstated in marketing materials. Independent validation studies report the following

performance against clinical references (ECG for HR, arterial blood gas or certified pulse oximeters for $SpO_2$):

- **Resting heart rate**: ±2–3 bpm error in controlled settings (comparable to medical devices)
- **Exercise heart rate**: ±5–8 bpm error due to motion artifacts, especially during high-intensity or irregular arm movements
- **$SpO_2$**: ±1–2% error within 90–100% saturation range under ideal conditions; accuracy degrades significantly below 90% or during motion
- **HRV**: Generally unreliable during ambulation due to misclassified peaks; suitable only for resting-state analysis

Critical limitations include:

1. **Motion artifacts**: The dominant error source (accounting for >70% of inaccuracies), as limb acceleration generates optical signals mimicking cardiac pulsations.
2. **Skin pigmentation**: Melanin absorbs green light, reducing signal amplitude in darker skin tones and increasing HR error by up to 2× compared to lighter skin—a well-documented bias with ethical implications [11].
3. **Fit and contact pressure**: Loose wear induces motion noise; excessive pressure causes local vasoconstriction, attenuating the PPG waveform.
4. **Ambient light interference**: Solved partially via modulated LED drive currents and

# I.13 Beyond PPG: Multimodal Wearable Sensing:

To overcome PPG's inherent limitations, modern wearables employ **sensor fusion**, combining complementary modalities to enhance robustness and expand clinical utility:

1. **Inertial Measurement Units (IMUs)**: 3-axis accelerometers and gyroscopes enable motion characterization, activity classification (e.g., walking vs. falling), and adaptive PPG artifact correction. For example, IMU data can trigger re-weighting of HR estimates during high-motion epochs.

2. **Temperature sensors**: Detect fever trends or circadian shifts; when fused with elevated HR and reduced $SpO_2$, they may flag early infection.
3. **Single-lead ECG**: Available in Apple Watch and Samsung Galaxy devices, ECG provides gold-standard rhythm confirmation for atrial fibrillation, resolving PPG's ambiguity in arrhythmia detection.
4. **Galvanic Skin Response (GSR)**: Measures electrodermal activity as a proxy for sympathetic nervous system activation, useful for stress and cognitive load monitoring.

The synergy of **PPG + IMU** alone can improve HR accuracy during ambulation from ~92% to >97%. Similarly, **PPG + temperature + activity** enables energy expenditure and respiratory infection risk models that surpass single-sensor approaches. This multimodal paradigm mirrors hospital-grade monitoring while remaining feasible in low-cost, wrist-worn form factors—laying the groundwork for intelligent, context-aware embedded systems [11].

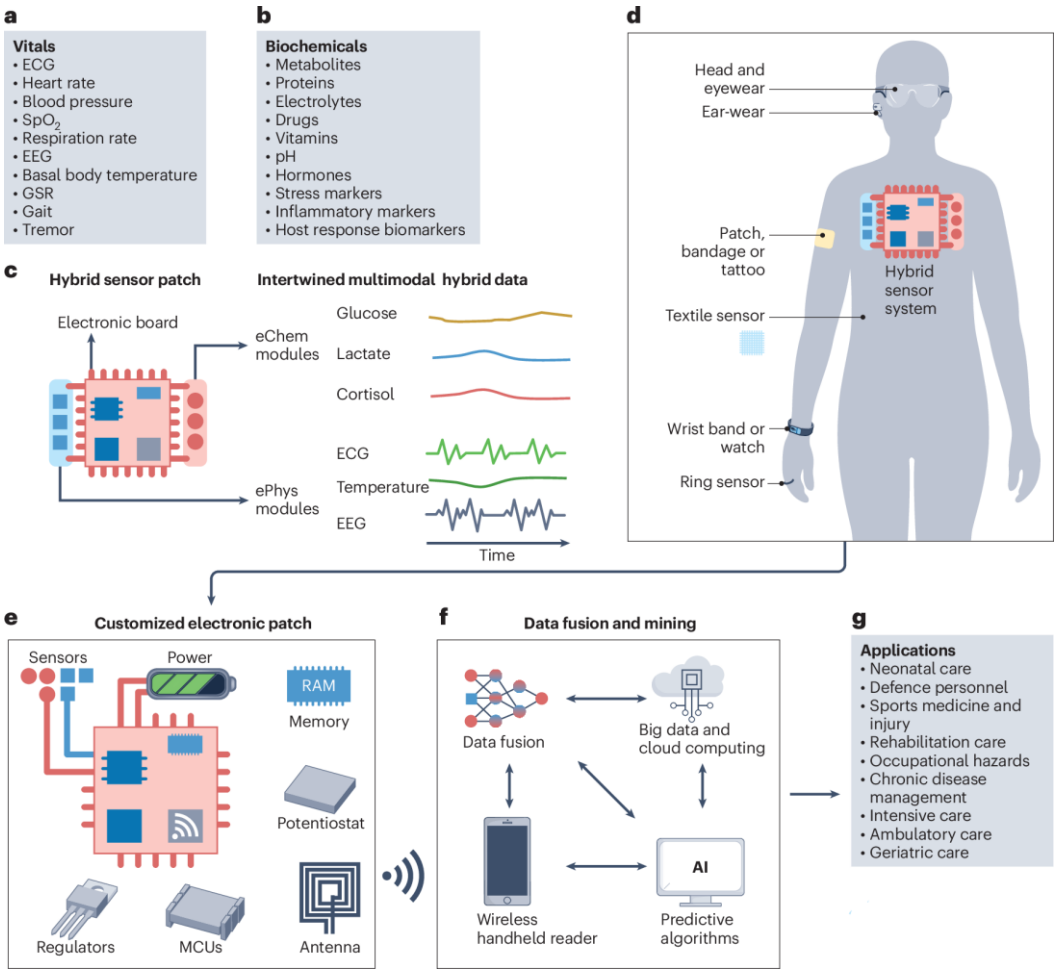



**Figure 1.9**: Multimodal wearable sensor fusion.

# I.14 Intelligent Embedded Systems in Healthcare:

Embedded systems are specialized computing platforms designed for dedicated functions under strict constraints of power, size, real-time performance, and reliability—making them ideal for medical and wearable applications. Unlike general-purpose computers, embedded systems integrate processing, memory, sensing, and communication into compact, often single-chip architectures (System-on-Chip, SoC) optimized for specific clinical tasks.

Key characteristics that distinguish embedded systems in healthcare include:

1. **Real-time determinism**: Predictable response latencies enabled by Real-Time Operating Systems (RTOS) such as FreeRTOS, critical for applications like arrhythmia detection or insulin delivery.
2. **Ultra-low power operation**: Achieved through deep-sleep modes (<1 μA), duty cycling, and dynamic voltage scaling—essential for multi-day wearable autonomy.
3. **Physical compactness**: Modern microcontrollers like the ESP32 occupy a $5 \times 5$ mm footprint, enabling seamless integration into wristbands, patches, or implantable devices.
4. **IoMT connectivity**: Native support for medical-grade wireless protocols (BLE 5.0, Wi-Fi, NB-IoT) ensures secure, interoperable data flow within the Internet of Medical Things ecosystem.

These systems power a continuum of healthcare technologies—from **pacemakers** and **glucose monitors** to **surgical robots** and **smart hospital beds**—demonstrating their versatility and clinical indispensability [12].

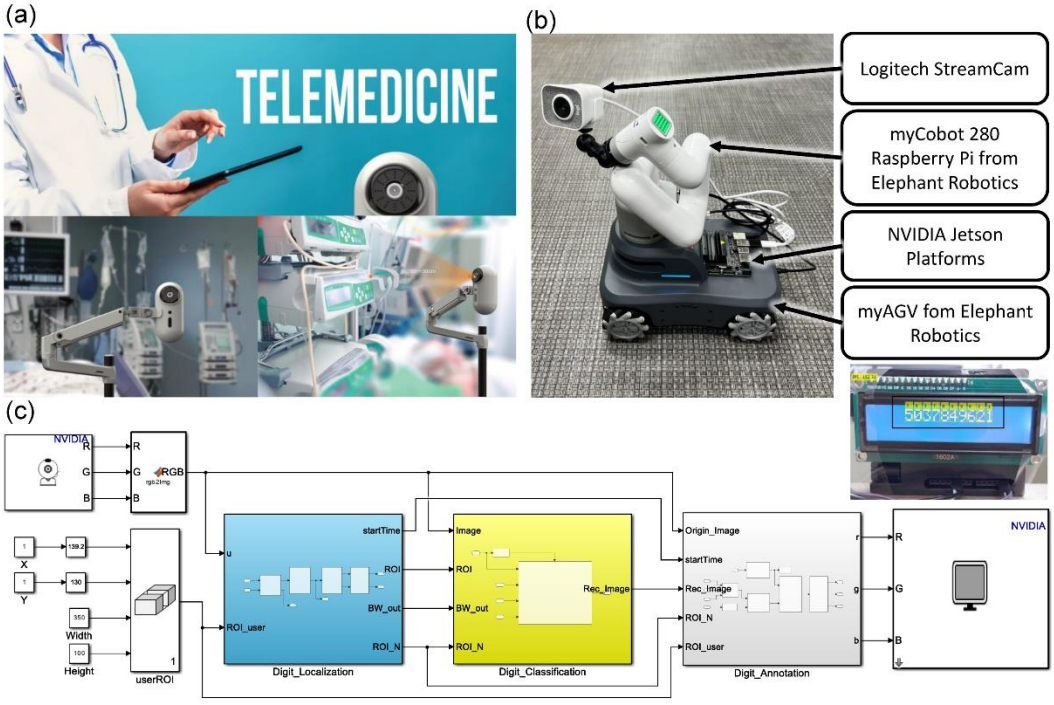



**Figure 1.10**: Embedded systems in healthcare.

## I.15 Microcontrollers in Wearable Health Devices:

The **ESP32 family** exemplifies the modern intelligent microcontroller, striking an optimal balance between performance, connectivity, and cost for wearable health applications. Its key specifications include:

- **Dual-core Xtensa LX6 processor** running at up to **240 MHz** with **520 KB SRAM**
- **Integrated wireless**: IEEE 802.11 b/g/n Wi-Fi and Bluetooth 5.0 (including BLE)
- **Rich peripherals**: 18-channel 12-bit SAR ADC, multiple I²C/SPI/UART interfaces, capacitive touch sensors
- **Power efficiency**: **10 µA** in deep-sleep mode, scaling to **~170 mA** during BLE/Wi-Fi transmission
- **Low cost**: **€2–5** in volume, enabling scalable deployment in resource-constrained settings

Firmware development for such platforms typically leverages lightweight stacks like **Arduino Core** or **MicroPython**, structured as concurrent FreeRTOS tasks.

The architecture supports **end-to-end latency under 50 ms** for anomaly detection and alerting—sufficient for non-critical but timely interventions in home-based care [12].

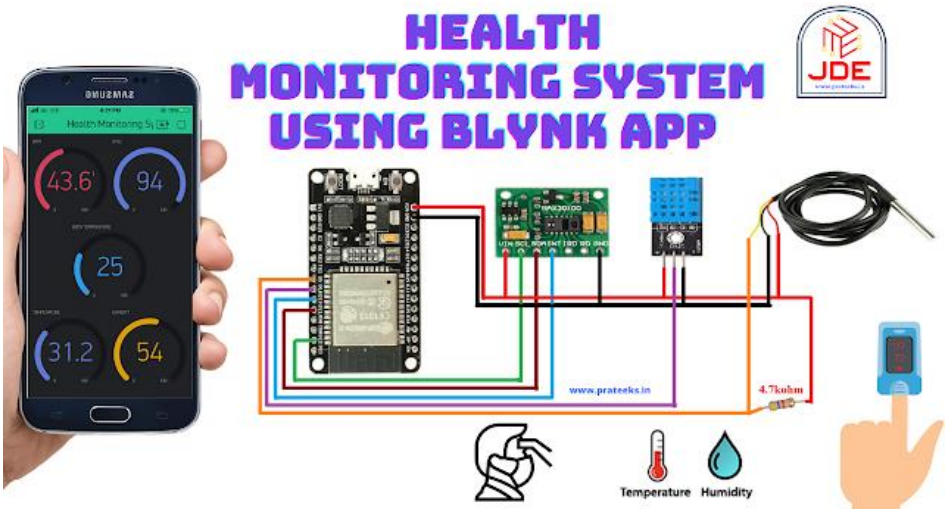



**Figure 1.11**: ESP32 block diagram highlighting healthcare-relevant peripherals.

# I.16 Edge AI on Embedded Platforms:

The emergence of **TinyML** (Tiny Machine Learning) has enabled the deployment of neural networks directly on microcontrollers, shifting intelligence from the cloud to the edge. Frameworks like **TensorFlow Lite Micro** compress models to **<1 MB** and execute inferences in **<1–5 ms** on ESP32-class hardware through techniques like **8-bit integer quantization**, which reduces memory by 4× and accelerates execution by 2–3×.

Key healthcare applications of Edge AI include:

1. **PPG peak detection**: 1D CNNs achieve **>98% beat detection accuracy** even under motion, outperforming threshold-based methods.
2. **Activity and fall classification**: IMU + PPG fusion with lightweight ML models delivers **>95% accuracy** in distinguishing walking, sitting, and falling.
3. **Arrhythmia screening**: Binary classifiers detect atrial fibrillation from short PPG segments with **sensitivity >97%**, reducing unnecessary cloud alerts.
4. **Motion artifact compensation**: Autoencoders or recurrent networks reconstruct clean PPG signals in real time, improving $SpO_2$ reliability during ambulation.

By processing raw signals locally, Edge AI **reduces cloud transmissions by up to 90%**, enhances **user privacy** (sensitive data never leaves the device), and ensures **offline functionality** during network outages—critical for rural or emergency contexts [12].

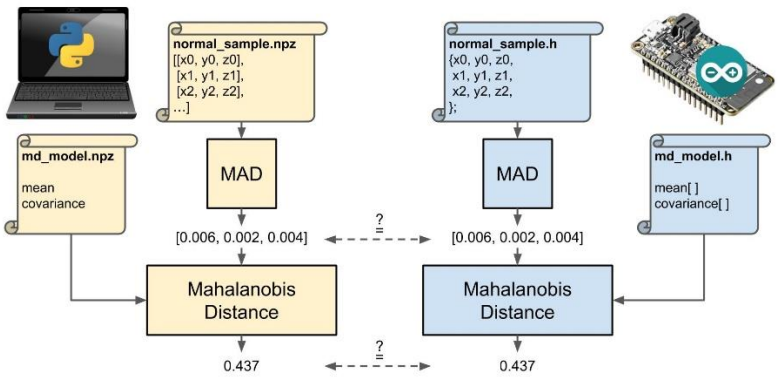



**Figure 1.12**: Edge AI pipeline on ESP32.

# I.17 Embedded System Architectures for Medical Reliability:

Medical embedded systems must satisfy far stricter requirements than consumer electronics, prioritizing **safety**, **predictability**, and **fault tolerance** over raw performance. Design for clinical use follows principles codified in international standards such as **IEC 60601-1** (medical electrical equipment safety) and **ISO 13485** (quality management), which mandate rigorous risk management, electromagnetic compatibility (EMC), and defined "essential performance" thresholds that must be maintained even under fault conditions.

Key architectural strategies to ensure reliability include:

1. **Watchdog timers**: Reset the system automatically if firmware hangs.
2. **Brown-out detection**: Prevent erratic behavior during low-voltage conditions by entering a safe shutdown state.
3. **Error-correcting code (ECC) memory**: Mitigates single-bit flips from cosmic radiation or electrical noise.
4. **Dual-core lockstep (in safety-critical SoCs)**: Compares outputs from redundant cores to detect silent failures.
5. **Failsafe modes**: Upon sensor disconnection or signal loss, the system defaults to a known-safe state (e.g., stop actuation, log error, alert user).

Power management is equally critical in wearable and home-care contexts. Achieving **7–30 days of battery autonomy** typically relies on **ultra-low duty cycling**—e.g., 1% active time (sensing + BLE transmission) and 99% deep-sleep—combined with **dynamic voltage scaling** (240 MHz during processing → 40 kHz during idle). While energy harvesting (solar, thermoelectric) remains experimental for medical wearables, it is increasingly explored in implantable and robotic platforms [13].

## I.18 General Overview of Healthcare Robots:

Healthcare robots are categorized by their primary role in the care continuum:

1. **Logistics robots** (e.g., TUG by Aethon): Transport medications, lab samples, and linens in hospitals, reducing human exposure and operational burden.
2. **Rehabilitation robots** (e.g., Lokomat, exoskeletons): Deliver consistent, high-intensity physical therapy for stroke or spinal cord injury patients.
3. **Socially assistive robots (SARs)**: Provide companionship, cognitive stimulation, and behavioral prompting without physical contact—examples include PARO (therapeutic seal) and ElliQ.

In **home and long-term care**, SARs are particularly relevant. They support medication adherence, encourage physical activity, and detect deviations from daily routines (e.g., missed meals, prolonged inactivity). When integrated with wearable biosensors, they gain **physiological context**, enabling responses like:

*"Your heart rate is elevated. Would you like to sit down and take deep breaths?"*

Critically, SARs function as **local IoMT hubs**, aggregating data from wearables, environmental sensors (door/window, bed occupancy), and voice interfaces. Their **physical embodiment**—movement, expressive lights, directional speech—makes alerts more noticeable than smartphone notifications, especially for elderly users with sensory impairments. However, **wheeled platforms struggle with common domestic obstacles** (rugs, thresholds, cords), a limitation that motivates alternative morphologies like **quadrupedal robots**, which offer superior stability on uneven surfaces [13].

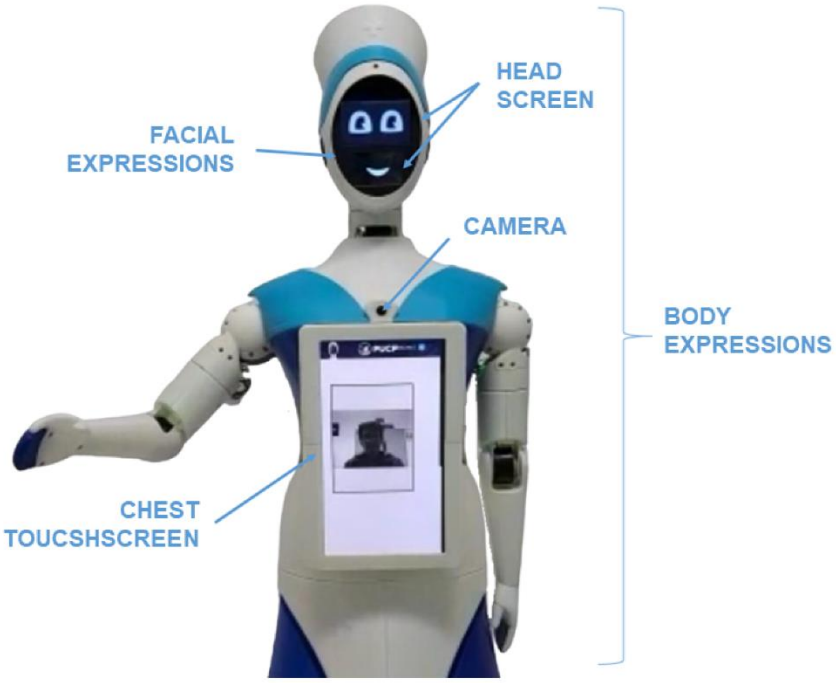



**Figure 1.13**: Elements for human–robot interaction.

## I.19 Ethical and Privacy Issues in Connected Healthcare:

The integration of continuous biosensing, AI analytics, and embodied robotics introduces profound ethical and legal challenges. Health data—especially longitudinal vital signs, behavior logs, and audio/video streams—are among the most sensitive personal information, demanding **privacy by design** and **security by default**.

Regulatory frameworks such as the **EU General Data Protection Regulation (GDPR)** and the **U.S. Health Insurance Portability and Accountability Act (HIPAA)** require:

- **Data minimization**: Collect only what is strictly necessary.
- **Purpose limitation**: Use data only for predefined, consented purposes.
- **User rights**: Enable access, correction, and deletion of personal data.
- **Technical safeguards**: Implement end-to-end encryption, secure boot, and authenticated over-the-air (OTA) updates.

Embedded systems can mitigate privacy risks through **edge-first architectures**: raw PPG, video, or audio remain on the local device (wristband or robot), and only **derived indicators** (e.g., "HR = 112 bpm, quality = high") are transmitted. This approach aligns with the **"least privilege" principle** and reduces attack surface.

Ethical deployment further requires:

1. **Transparency**: Clear explanation of what is monitored and why.
2. **Granular consent**: Allow users to disable specific sensors (e.g., keep HR monitoring but turn off camera).
3. **Bias awareness**: Ensure PPG and AI models are validated across diverse skin tones, ages, and activity levels.
4. **Human oversight**: Robots should never replace clinical judgment; alerts must be triaged by qualified personnel.

These principles are not optional add-ons but **foundational to trust, adoption, and regulatory compliance** in digital health [13].

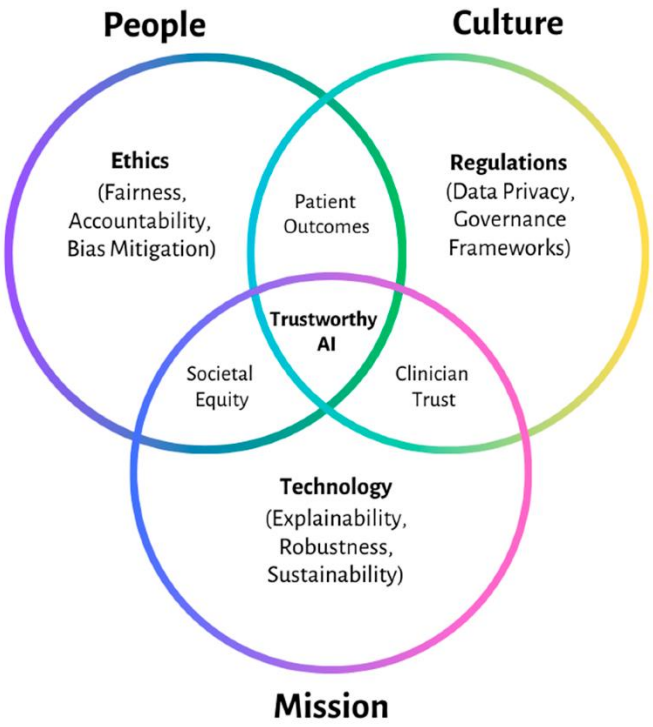



**Figure 1.14**: Ethical design framework for embedded healthcare systems.

# II. CHAPTER II:

# DESIGN

# II.1 Introduction :

This chapter describes the hardware and software used in the project and explains how each part contributes to the overall system. It outlines the physical components that enable operation and the software that controls, processes, and manages the system's functions. Together, they form the foundation that makes the project work effectively.

# II.2 Biomedical Sensors:

## II.2.1 MAX30102: Heart Rate and SpO2 Sensor:

The MAX30102 is an integrated pulse oximetry and heart-rate monitor biosensor module. It includes internal LEDs, photodetectors, optical elements, and low-noise electronics with ambient light rejection. The MAX30102 provides a complete system solution to ease the design-in process for mobile and wearable devices.

The MAX30102 operates on a single 1.8V power supply and a separate 3.3V power supply for the internal LEDs. Communication is through a standard $I^2C$-compatible interface. The module can be shut down through software with zero standby current, allowing the power rails to remain powered at all times.

### Key Specifications:

- Heart-Rate Monitor and Pulse Oximeter Biosensor in LED Reflective Solution
- Tiny 5.6mm x 3.3mm x 1.55mm 14-Pin Optical Module
    - Integrated Cover Glass for Optimal, Robust Performance
- Ultra-Low Power Operation for Mobile Devices
    - Programmable Sample Rate and LED Current for Power Savings
    - Low-Power Heart-Rate Monitor (< 1mW)
    - Ultra-Low Shutdown Current (0.7µA, typ)
- Fast Data Output Capability
    - High Sample Rates
- Robust Motion Artifact Resilience

- High SNR
- -40°C to +85°C Operating Temperature Range

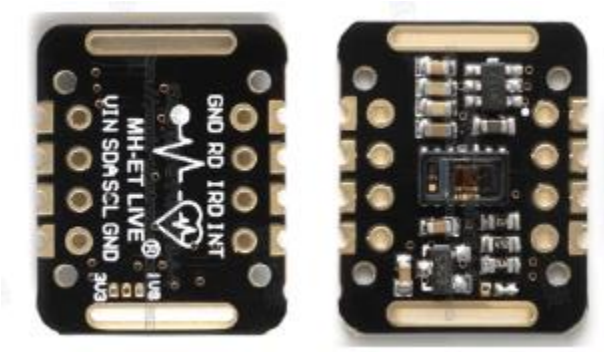



**Figure 2.15**: MAX30102 module

## II.2.2 MAX30205: Temperature Sensor:

The MAX30205 temperature sensor accurately measures temperature and provide an overtemperature alarm/interrupt/shutdown output. This device converts the temperature measurements to digital form using a high-resolution, sigma-delta, analog-to-digital converter (ADC). Accuracy meets clinical thermometry specification of the ASTM E1112 when soldered on the final PCB. Communication is through an $I^2C$-compatible 2-wire serial interface.

The $I^2C$ serial interface accepts standard write byte, read byte, send byte, and receive byte commands to read the temperature data and configure the behavior of the open-drain overtemperature shutdown output.

The MAX30205 features three address select lines with a total of 32 available addresses. The sensor has a 2.7V to 3.3V supply voltage range, low 600µA supply current, and a lockup-protected $I^2C$-compatible interface that make them ideal for wearable fitness and medical applications.

This device is available in an 8-pin TDFN package and operates over the 0°C to +50°C temperature range.

**Key Specifications:**

- High Accuracy and Low-Voltage Operation Aids Designers in Meeting Error and Power Budgets
    - 0.1°C Accuracy (37°C to 39°C)
    - 16-Bit (0.00390625°C) Temperature Resolution
    - 2.7V to 3.3V Supply Voltage Range
- One-Shot and Shutdown Modes Help Reduce Power Usage
- 600µA (typ) Operating Supply Current
- Digital Functions Make Integration Easier into Any System
    - Selectable Timeout Prevents Bus Lockup
    - Separate Open-Drain OS Output Operates as Interrupt or Comparator/Thermostat Output

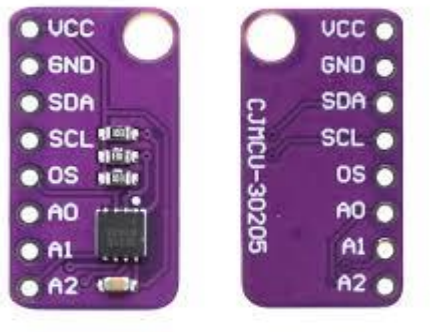



**Figure 2.16**: MAX30205 module

# II.3 Microcontrollers and Processing Units:

## II.3.1 ESP 12-E:

ESP-12E is a member of 'ESP-XX' series. Although all of them are based on ESP8266 SoC, they differ in on output pins, flash memory and antenna type. These modules numbered from ESP-01 to ESP-15 and are best in performance and cost. Many engineers use these modules to setup a

wireless communication between two applications. For data sharing and IoT, you will find these modules Ideal.


**Figure 2.17**: **ESP 12E module**

## Key Specifications:

- **Processor**: Tensilica L106 32-bit RISC, 80/160MHz
- **RAM**: 160KB total (80KB available for user programs after WiFi stack)
- **Flash**: 4MB (ESP-12E variant) for program and data storage
- **WiFi**: IEEE 802.11 b/g/n, 2.4GHz, supporting STA/AP/STA+AP modes
- **Peripherals**: GPIO, I²C, SPI, UART, ADC, PWM, I²S

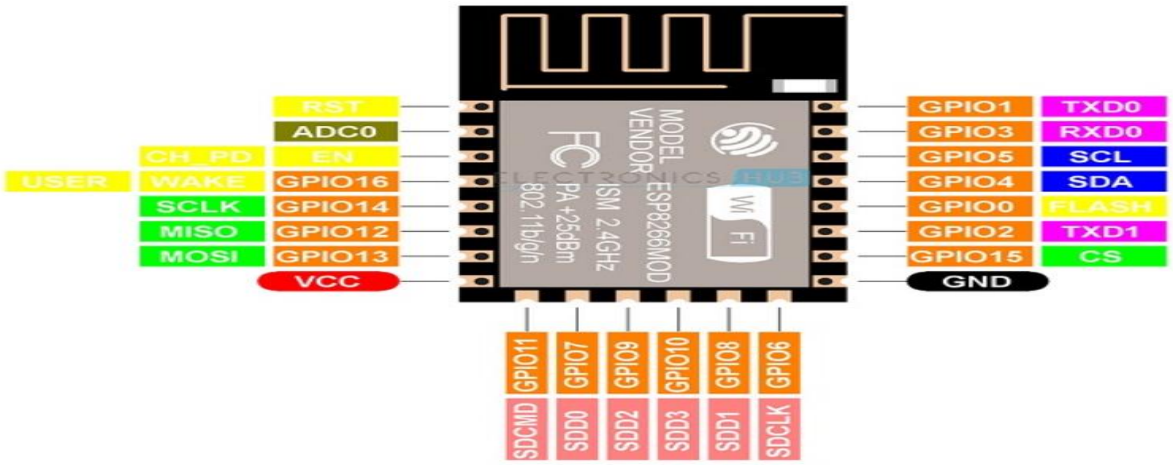



**Figure 2.18**: ESP 12E pins header

## II.3.2 Raspberry Pi Zero 2 W:

Raspberry Pi Zero 2 W is the latest product in Raspberry Pi's most affordable range of single-board computers. The successor to the breakthrough Raspberry Pi Zero W, Raspberry Pi Zero 2 W is a form factor–compatible drop-in replacement for the original board.

The board incorporates a quad-core 64-bit Arm Cortex-A53 CPU, clocked at 1GHz. At its heart is a Raspberry Pi RP3A0 system-in-package (SiP), integrating a Broadcom BCM2710A1 die with 512MB of LPDDR2 SDRAM.

The upgraded processor provides Raspberry Pi Zero 2 W with 40% more single-threaded performance, and five times more multi-threaded performance, than the original single-core Raspberry Pi Zero.

Raspberry Pi Zero 2 W offers 2.4GHz 802.11 b/g/n wireless LAN and Bluetooth® 4.2, along with support for Bluetooth® Low Energy, and modular compliance certification.The board has a microSD card slot, a CSI-2 camera connector, a USB On-The-Go (OTG) port, and an unpopulated footprint for a HAT-compatible 40-pin GPIO header. It is powered via a micro USB socket.

Video output is via a mini HDMI port; composite video output can easily be made available via test points, if needed. Sharing the same form factor as the original Raspberry Pi Zero, Raspberry Pi Zero 2 W fits inside most existing Raspberry Pi Zero cases.

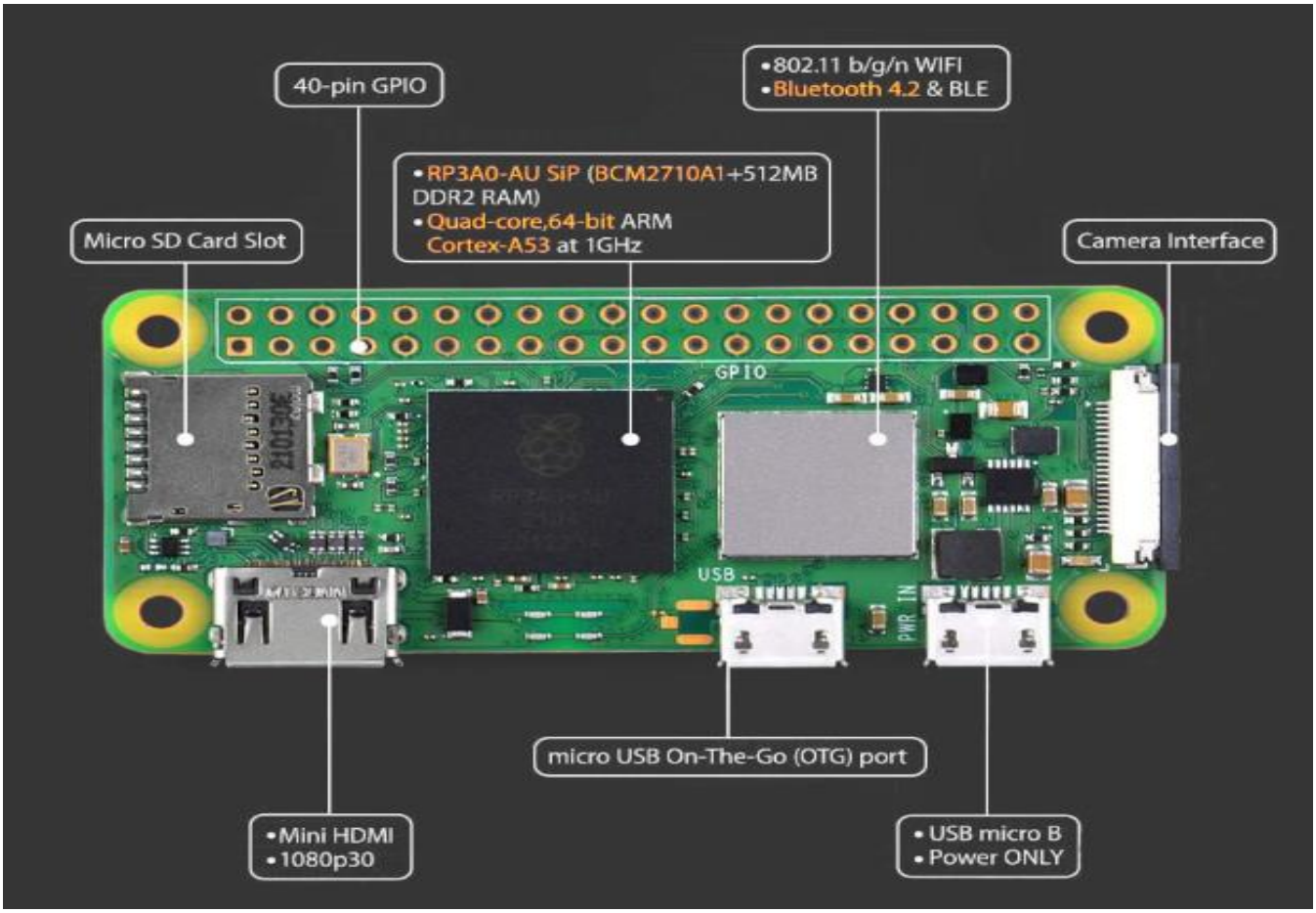



**Figure 2.19**: **Raspberry PI zero 2w module**

# Key Specifications:

- **Processor**: Broadcom BCM2710A1, quad-core ARM Cortex-A53 @ 1GHz
- **RAM**: 512MB LPDDR2 SDRAM
- **Storage**: microSD card (class 10 recommended, 16-32GB typical)
- **Connectivity**:
    - WiFi 802.11 b/g/n (2.4GHz)
    - Bluetooth 4.2 / Bluetooth Low Energy (BLE)
- **Video Output**: Mini-HDMI port
- **Interfaces**:

- 40-pin GPIO header (same as other Pi models)
- USB 2.0 (micro-USB OTG)
- Camera Serial Interface (CSI)

- **Power**: 5V via micro-USB, 100-400mA depending on load

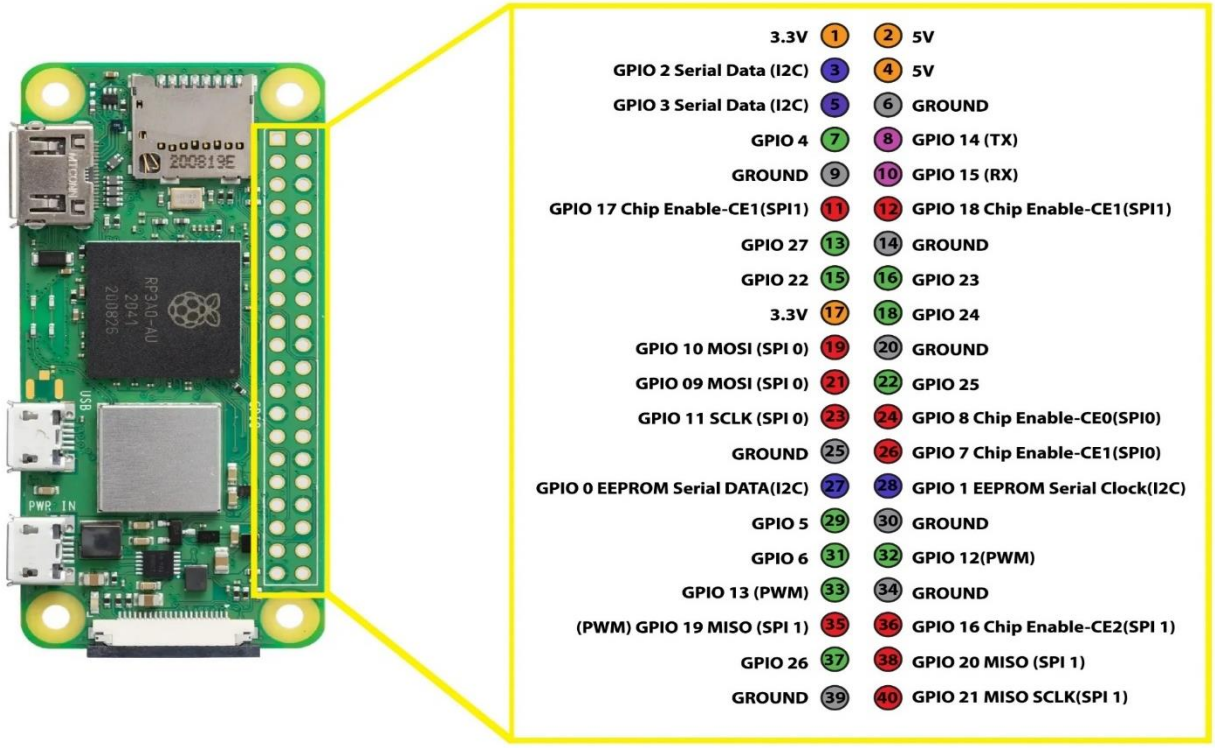



**Figure 2.20**: Raspberry PI zero 2w pins headers

## II.3.3 ESP 8266 (additional) :

The ESP8266EX microcontroller integrates a Tensilica L106 32-bit RISC processor, which achieves extra-low power consumption and reaches a maximum clock speed of 160 MHz. The Real-Time Operating System (RTOS) and Wi-Fi stack allow about 80% of the processing power to be available for user application programming and development.

## Key Specifications:

- **Processor**: The ESP8266 is powered by a 32-bit RISC microprocessor core based on the Tensilica Diamond Standard 106Micro, running at 80 MHz or 160 MHz.
- **Memory**: It includes 32 KiB instruction RAM, 32 KiB instruction cache RAM, 80 KiB user-data RAM, and 16 KiB ETS system-data RAM.

- **Wi-Fi**: It supports IEEE 802.11 b/g/n Wi-Fi with integrated TR switch, balun, LNA, power amplifier, and matching network.
- **GPIO Pins**: The chip has 17 GPIO pins, which can be used for various input/output operations.
- **Interfaces**: It supports SPI, I²C (software implementation), I²S interfaces with DMA, UART, and a 10-bit ADC.

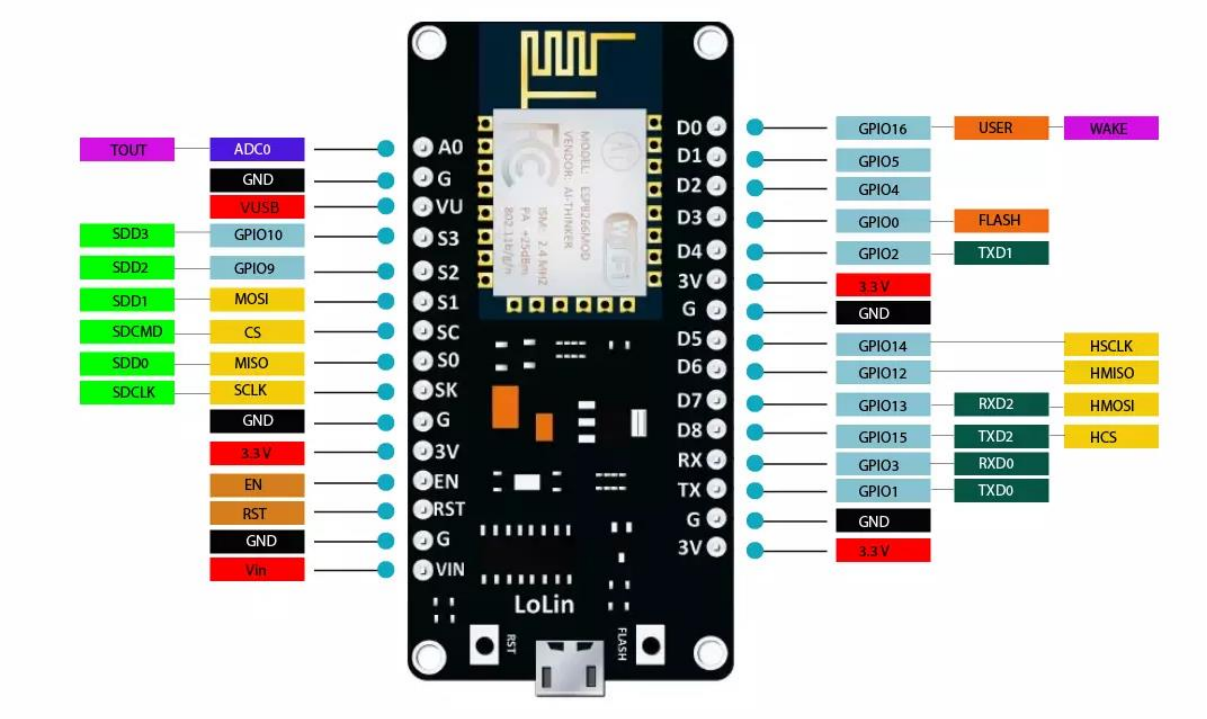



**Figure 2.21:** ESP8266 modules and pins headers

# II.4 Modules:

## II.4.1 SG-90 Micro Servo:

A servo motor is a closed-loop system that uses position feedback to control its motion and final position. The SG90 Servo Motor is a popular, compact, and lightweight servo motor widely used in electronics, robotics, and DIY projects. It is known for its low cost and ease of use, especially with microcontrollers like Arduino, Raspberry Pi, and ESP boards.

### Key Specifications:

- Operating Voltage is +5V typically

- Torque: 2.5kg/cm
- Operating speed is 0.1s/60°
- Gear Type: Plastic
- Rotation : 0°-180°
- Weight of motor : 9gm
- Package includes gear horns and screws

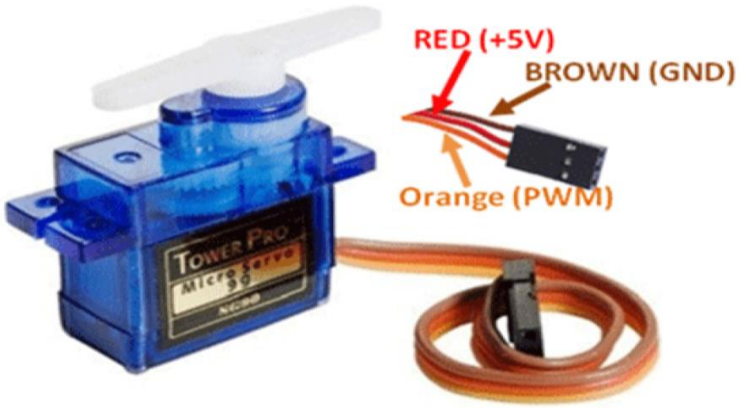



**Figure 2.22: SG-90 microservo**

## II.4.2 PCA9685 16-Channel 12-bit PWM/Servo Driver:

The PCA9685 is a highly integrated PWM (Pulse Width Modulation) controller manufactured by NXP Semiconductors, specifically designed to control multiple servos, LEDs, or other PWM-driven devices from a single I²C interface. This chip is particularly popular in robotics applications where numerous servo motors need independent control without consuming multiple microcontroller GPIO pins or timer resources.

The device operates as an I²C-bus controlled 16-channel LED controller optimized for Red/Green/Blue/Amber (RGBA) color backlighting applications, but its most common use in robotics is for servo motor control. Each of the 16 outputs can be independently controlled with 12-bit resolution (4096 steps), providing smooth, precise control of servo positions or LED brightness.

## Key Specifications:

- **Supply Voltage (VDD)**: 2.3V to 5.5V (logic supply)
- **Output Voltage (V+)**: Up to 6V (separate power rail for outputs)
- **Logic Levels**: Compatible with 3.3V and 5V systems
- **Output Current**: 25mA maximum per output (LED drive mode)
- **Total Output Current**: 400mA maximum for all outputs combined
- **Operating Temperature**: -40°C to +85°C (industrial grade)

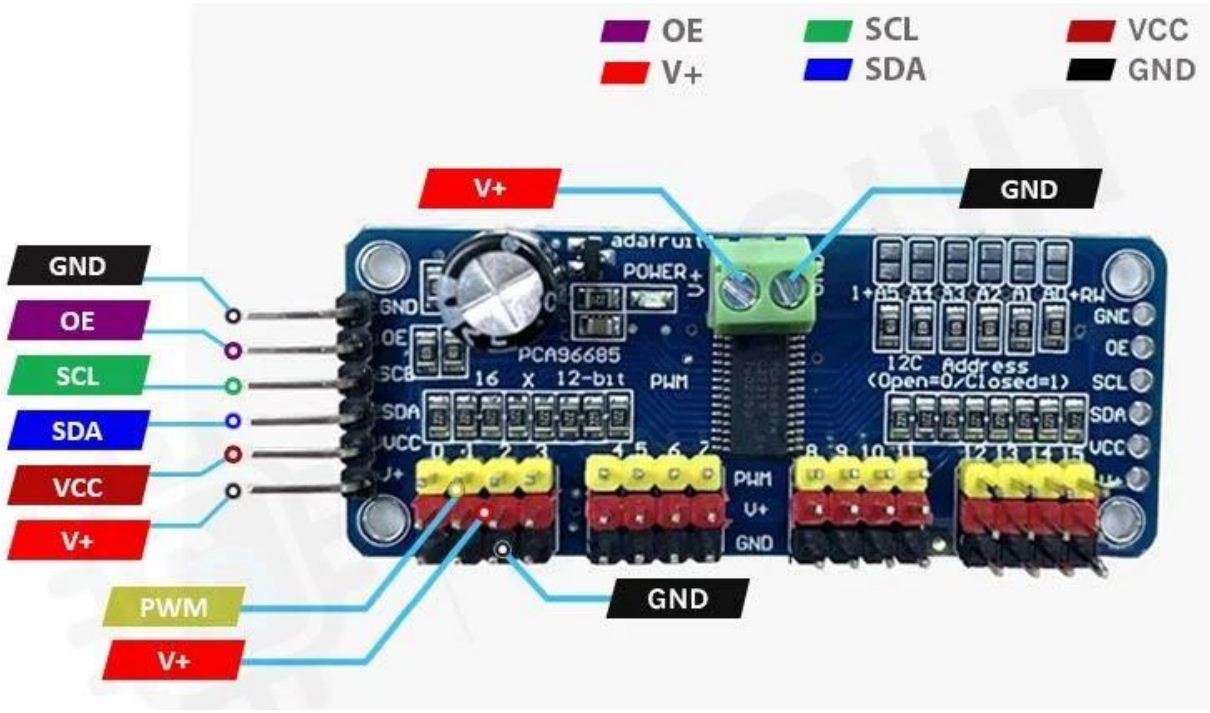



**Figure 2.23:** PCA9685 16-Channel 12-bit PWM/Servo Driver

## II.4.3 Raspberry Pi Camera REV 1.3 :

The Raspberry Pi Camera Rev 1.3 features a 5MP sensor, captures HD video, and is compatible with Raspberry Pi models via the CSI interface.

## Key Specifications:

- **Sensor:** OmniVision OV5647

- **Maximum Photo Resolution:** 5MP (2592 x 1944 pixels)
- Video Resolution: Supports 1080p at 30 frames per second (fps), 720p at 60 fps, and 640x480 at 90 fps.
- **Lens Viewing Angle:** Approximately 72.4°.
- **Interface:** Connects via the Camera Serial Interface (CSI) on Raspberry Pi boards, making it easy to integrate into various projects.
- **Image Quality:** Capable of capturing sharp images and HD video, making it suitable for a variety of applications, including photography and video recording.

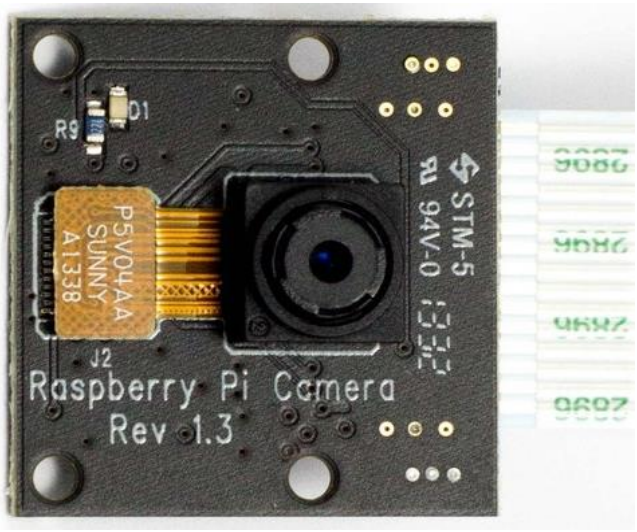



**Figure 2.24:** PI camera rev 1.3 modules

## II.4.4 18650 Battery 3.7V:

An 18650 battery is a rechargeable lithium-ion battery that measures 18mm in diameter and 65mm in length. It typically has a voltage of 3.6 to 3.7 volts and a capacity ranging from 1800mAh to 3500mAh. These batteries are widely used in various applications, including laptops, flashlights, power tools, and electric vehicles due to their high energy density and reliability.

## Key Specifications:

- **Dimensions:**

  A cylindrical cell about **18 mm in diameter** and **65 mm in length** (that's where "18650" comes from).
- **Nominal Voltage:**

  Usually **3.6 V or 3.7 V** depending on chemistry.
- **Full-Charge Voltage:**

  Typically **4.20 V** when fully charged.
- **Minimum Safe Voltage (cut-off):**

  Normally **2.5 V to 3.0 V**.

  Going below this can damage the cell.
- **Capacity:**

  Most cells range from **1800 mAh to 3500 mAh**.

  High-quality high-capacity cells reach around **3300–3500 mAh**.
- **Energy per cell:**

  Roughly **8–12 Wh**, depending on voltage and capacity.
- **Weight:**

  Usually between **44 g and 48 g**.
- **Discharge Rate (maximum continuous current):**

  Varies widely by model: **5 A to 35 A**.

  High-drain cells can deliver much more current but usually have lower capacity.
- **Chemistry Types (common variants):**

- **NMC (INR)** – balanced performance and safety
- **IMR (LMO)** – high-drain, safer
- **ICR ($LiCoO_2$)** – high capacity, lower discharge capability
- **$LiFePO_4$** – very safe and long-life but lower voltage

- **Cycle Life:**

  Typically **300–800 cycles**, depending on depth-of-discharge and temperature.

**Figure 2.25:** 18650 battery

## II.4.5 SY8205 DC/DC Step-Down:

A compact DC to DC power supply module that can accept a wide input voltage range of 7 to 30V (< 24V recommended) with a high current output capability. The module will step down the input voltage to a fixed stable 5V output. Input and output connections are via large solderable pads and also includes a grid of pads in an arrangement for powering multiple servos (max number of servos will be dependant on power requirements of each servo), or just for convenient connection of multiple 5V devices. Should you not require the additional pads these can be broken off leaving the main large output pads.

## Key Specifications:

- **Module type:** SY8205 + servo pads
- **Input voltage:** 7V to 30V (< 24V recommended)
- **Output Voltage:** 5V
- **Max output current** (peak or with active cooling @ 5V): 5A
- **Max output current** (continuous @ 5V): 3.7A (see table)
- **Dimensions:** 43.5 (31.5 with servo pads removed) x 20.2 x 5.0mm


**Figure 2.26:** SY8502 step down module

# II.5 Softwares:

## II.5.1 Arduino IDE:

The Arduino IDE 2 features a new sidebar, making the most commonly used tools more accessible, is an open-source project that is free for anyone to download. You can contribute to the project through **donations**, or by reporting issues to **our GitHub repository**.


**Figure 2.27:** Arduino IDE interface

- **Verify / Upload** - compile and upload your code to your Arduino Board.
- **Select Board & Port** - detected Arduino boards automatically show up here, along with the port number.
- **Sketchbook** - here you will find all of your sketches locally stored on your computer. Additionally, you can sync with the **Arduino Cloud**, and also obtain your sketches from the online environment.
- **Boards Manager** - browse through Arduino & third party packages that can be installed. For example, using a MKR WiFi 1010 board requires the **Arduino SAMD Boards** package installed.
- **Library Manager** - browse through thousands of Arduino libraries, made by Arduino & its community.
- **Debugger** - test and debug programs in real time.
- **Search** - search for keywords in your code.
- **Open Serial Monitor** - opens the Serial Monitor tool, as a new tab in the console.

## II.5.2 Raspberry Pi OS (32-bit):

The Raspberry Pi Foundation makes and keeps Raspberry Pi OS (formerly Raspbian) up to date. It is the official operating system for Raspberry Pi single-board computers. The 32-bit Desktop version of this Debian-based Linux distribution is the full-featured version that is optimized for Raspberry Pi hardware. It provides a complete desktop computing environment that is good for education, development, and real-world uses like robotics and embedded systems.

The 32-bit Desktop version is specifically compiled and optimized for 32-bit ARM processors (ARMv7 architecture with NEON SIMD extensions), ensuring efficient execution on the Raspberry Pi's hardware:

- **Hardware Acceleration**: Optimized video drivers leverage the VideoCore GPU for graphical acceleration
- **Power Management**: Tuned for ARM's power-efficient architecture

- **Peripheral Support**: Pre-configured drivers for Raspberry Pi-specific hardware (GPIO, camera interface, audio, etc.)


**Figure 2.28:** Raspberry PI OS user interface

## *Programming and Development:*

- **Thonny IDE:** Python development environment optimized for beginners but suitable for professional development
- **Geany:** Lightweight, fast text editor with IDE features
- **Terminal Emulator:** Command-line access (LXTerminal)
- **Python 3:** Pre-installed with extensive standard library

## Utilities

- **Task Manager:** Monitor system resources and running processes
- **File Manager:** Browse and manage files with graphical interface
- **Calculator:** Basic and scientific calculations
- **Screenshot Tool:** Capture screen images

## **Interface Options** (Critical for Hardware Integration):

- **Camera**: Enable Pi Camera interface (CSI)
- **SSH**: Enable secure shell remote access
- **VNC**: Enable VNC remote desktop server
- **SPI**: Enable Serial Peripheral Interface
- **I²C**: Enable Inter-Integrated Circuit bus (required for PCA9685)
- **Serial Port**:
    - Serial console: Disable (to free UART for data communication)
    - Serial hardware: Enable (to activate UART peripheral)
- **1-Wire**: Enable for temperature sensors
- **Remote GPIO**: Allow remote control of GPIO pins

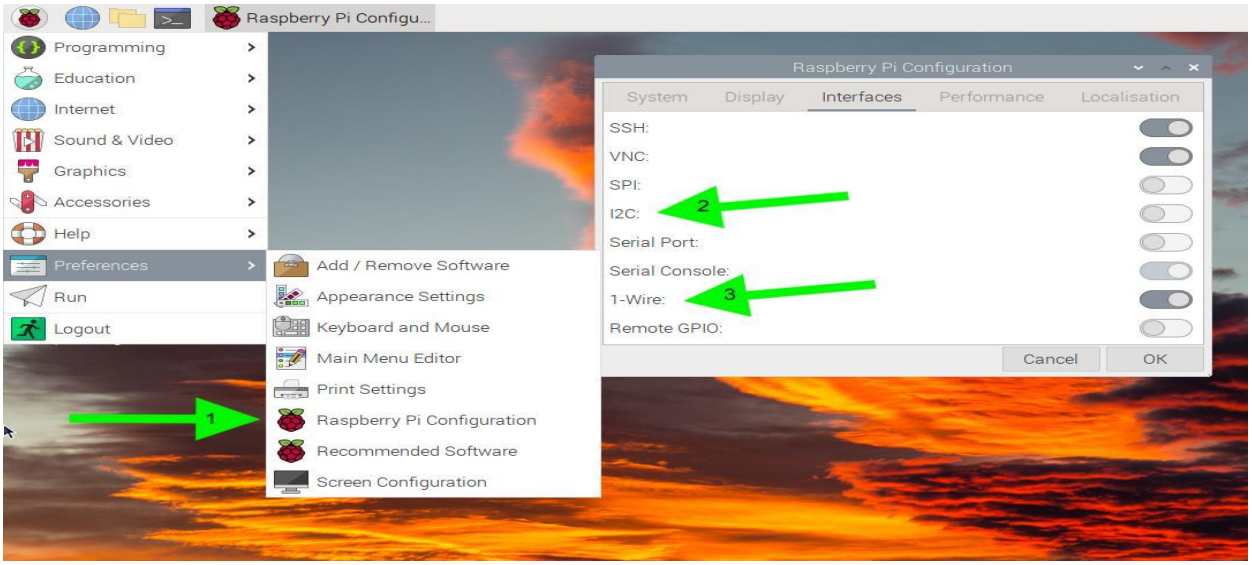



**Figure 2.29:** Raspberry PI OS menue

## II.6 Conclusion:

In this chapter, the hardware and software components work together to deliver a reliable and efficient system. The hardware provides the physical foundation for data collection, processing, and output, while the software manages control, automation, and user interaction. By integrating both elements effectively, the project achieves its intended functionality, improves performance, and ensures ease of use and scalability for future enhancements.

# III. CHAPTER III:

# IMPLEMENTATION

# III.1 Introduction

The Implementation of System chapter presents the practical realization of the design concepts outlined in the previous sections of this study. It describes how the proposed system was developed, the tools and technologies used, and the procedures followed to transform theoretical models into a functional solution. This chapter also highlights the step-by-step approach adopted during development, including system setup, module integration, interface creation, and database configuration.

Furthermore, it explains how different components of the system work together to achieve the intended objectives. The implementation phase is crucial because it validates the feasibility of the design and demonstrates how conceptual ideas are translated into an operational system. By detailing the development environment, programming methodologies, and testing mechanisms, this chapter provides a comprehensive understanding of how the final system was built and prepared for deployment.

# III.2 Implementation of health monitor:

## III.2.1 Functional Description:

The Health Monitoring System is an intelligent embedded solution designed to continuously measure and transmit vital signs using compact biomedical sensors and an ESP12E Wi-Fi module. The system collects physiological data—specifically heart rate, blood oxygen saturation ($SpO_2$), and body temperature—using the MAX30102 and MAX30205 sensors. After processing these signals, the ESP12E uploads the data to the Blynk IoT platform, where users can monitor their health parameters in real time.

## III.2.2 Data Processing:

The ESP12E:

- Reads raw sensor values through $I^2C$.
- Applies filtering algorithms (e.g., moving average or low-pass filtering) to stabilize readings.
- Calculates:

    - Heart rate from PPG peaks
    - $SpO_2$ using ratio of red/IR intensities
    - Temperature directly from MAX30205 output
- Validates data to avoid false readings (sensor noise, motion artifacts).

## III.2.3 Wireless Connectivity:

- The ESP12E connects to the local Wi-Fi network.
- Ensures secure communication with the **Blynk Cloud**.
- Periodically uploads:
    - Heart rate (BPM)
    - $SpO_2$ (%)
    - Body temperature (°C)
- Data transmission rate can be adjusted based on power consumption or real-time requirements.

## III.2.4 Cloud Monitoring (Blynk Platform):

The Blynk mobile app/dashboard provides:

- **Real-time visualization** of vital signs using widgets (gauges, charts, indicators).
- **Historical data logs** for trend analysis.
- **Alerts and notifications** when values exceed normal ranges (e.g., fever or low oxygen saturation).
- **Remote access**, allowing caregivers or users to check vital signs anytime.

## III.2.5 Hardware Architecture:

The **ESP-12E** is just a *bare ESP8266 module* soldered onto a small PCB.
It **does not include**:

- USB-to-serial converter
- Voltage regulator

- Auto-reset (EN/RESET, GPIO0) circuitry
- Easy programming pins
- Power protection or filtering

That means the ESP-12E **cannot connect directly to a computer**.

we use esp8266 board as a programmer because it has:

- USB-to-serial chip (CH340 or CP2102)
- Auto-programming circuitry (reset + GPIO0 control)
- 3.3V regulator
- USB connector
- Exposed pins on headers

A NodeMCU can be used in two main ways. In normal operation, it flashes and runs its own firmware by uploading code directly through the USB interface. Alternatively, it can be used as a programmer for another ESP-12E module.

This approach is popular because the NodeMCU board already has the necessary wiring in place, provides a stable 3.3 V power supply, automatically manages the ESP boot-mode signals, costs roughly the same as a dedicated USB-to-serial adapter, and is very easy to set up and use.

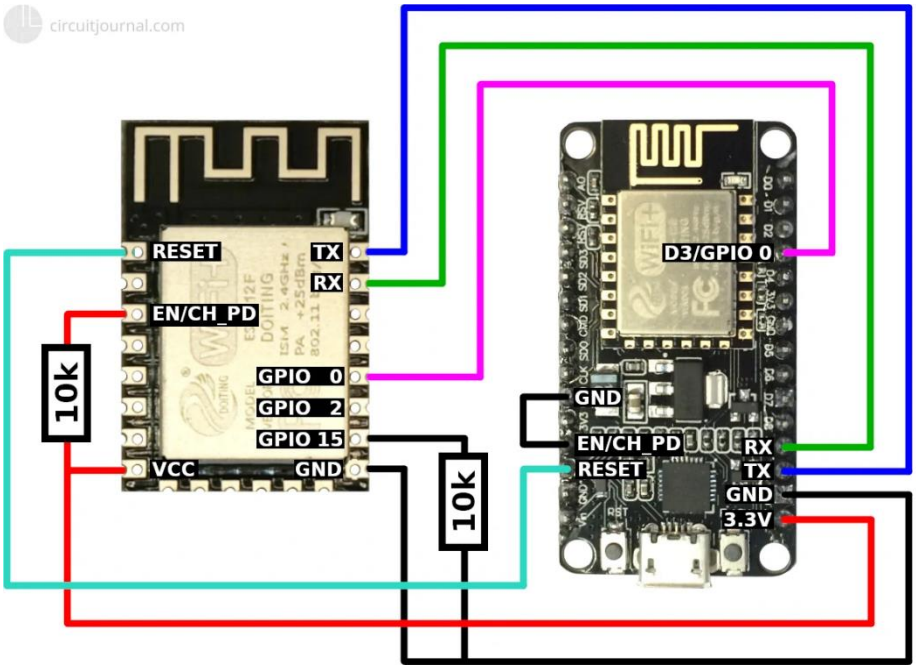



**Figure 3.30:** ESP12E programmer with ESP8266.

### *III.2.5.1 Pin Connections (ESP12E → MAX30102 & MAX30205):*

**Shared Connections (I²C bus):**

- ESP12E GPIO4 (SDA) → SDA (MAX30102) **and** → SDA (MAX30205)
- ESP12E GPIO5 (SCL) → SCL (MAX30102) **and** → SCL (MAX30205)

**Power:**

- ESP12E 3.3V → VIN/VCC (MAX30102)
- ESP12E 3.3V → VCC (MAX30205)
- ESP12E GND → GND (MAX30102)
- ESP12E GND → GND (MAX30205)

**Additional pins:**

- A0/A1/A2 (MAX30205) → to **GND** to set I²C address
- OS(MAX30205) → to 3.3V

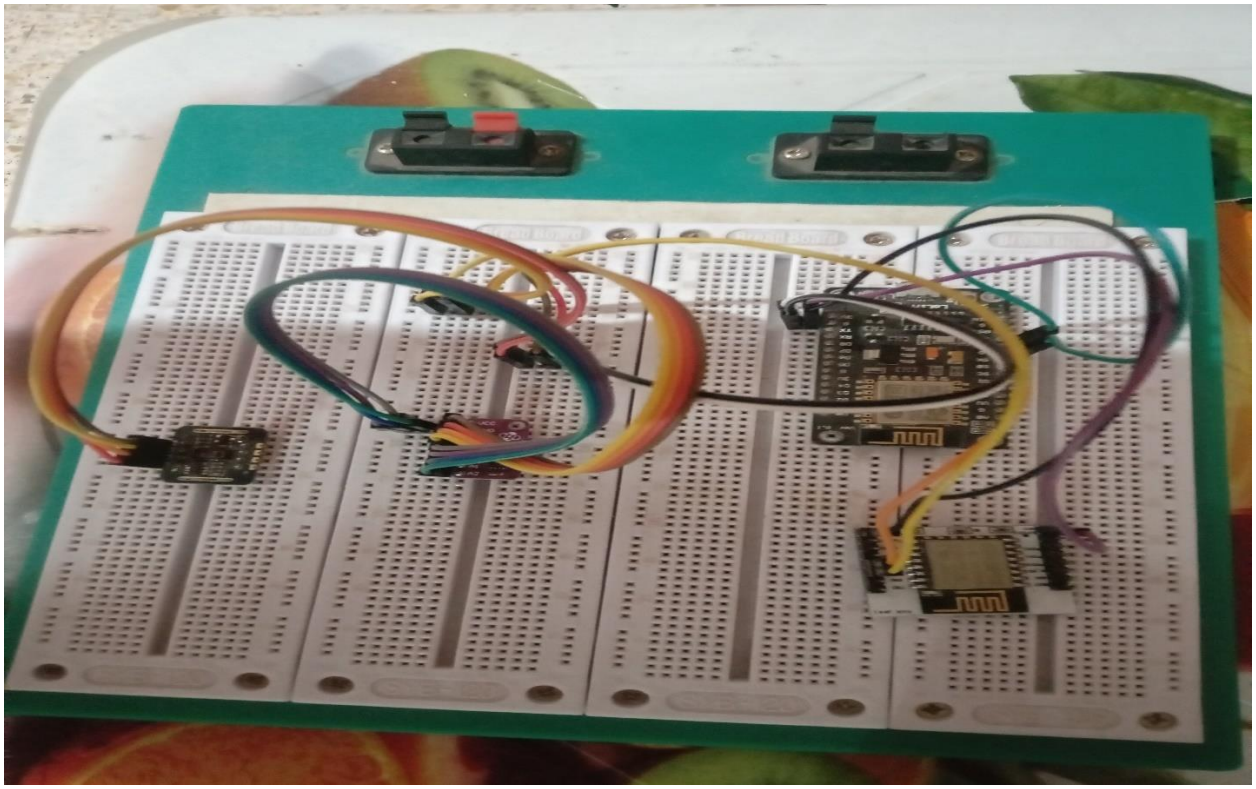
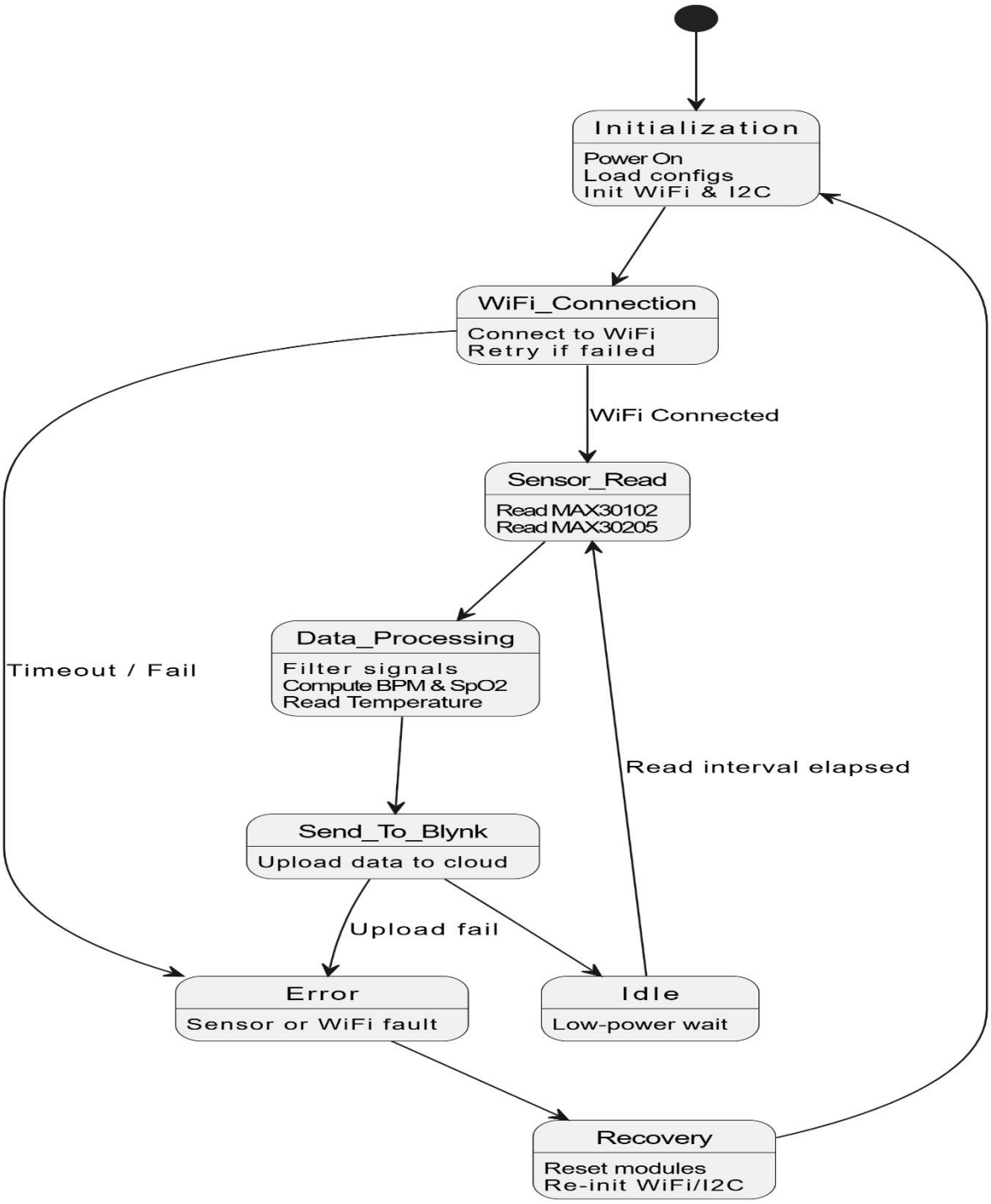


**Figure 3.30:** health monitor wiring

## III.2.6 UML modeling:

### *III.2.6.1 State machine diagram:*

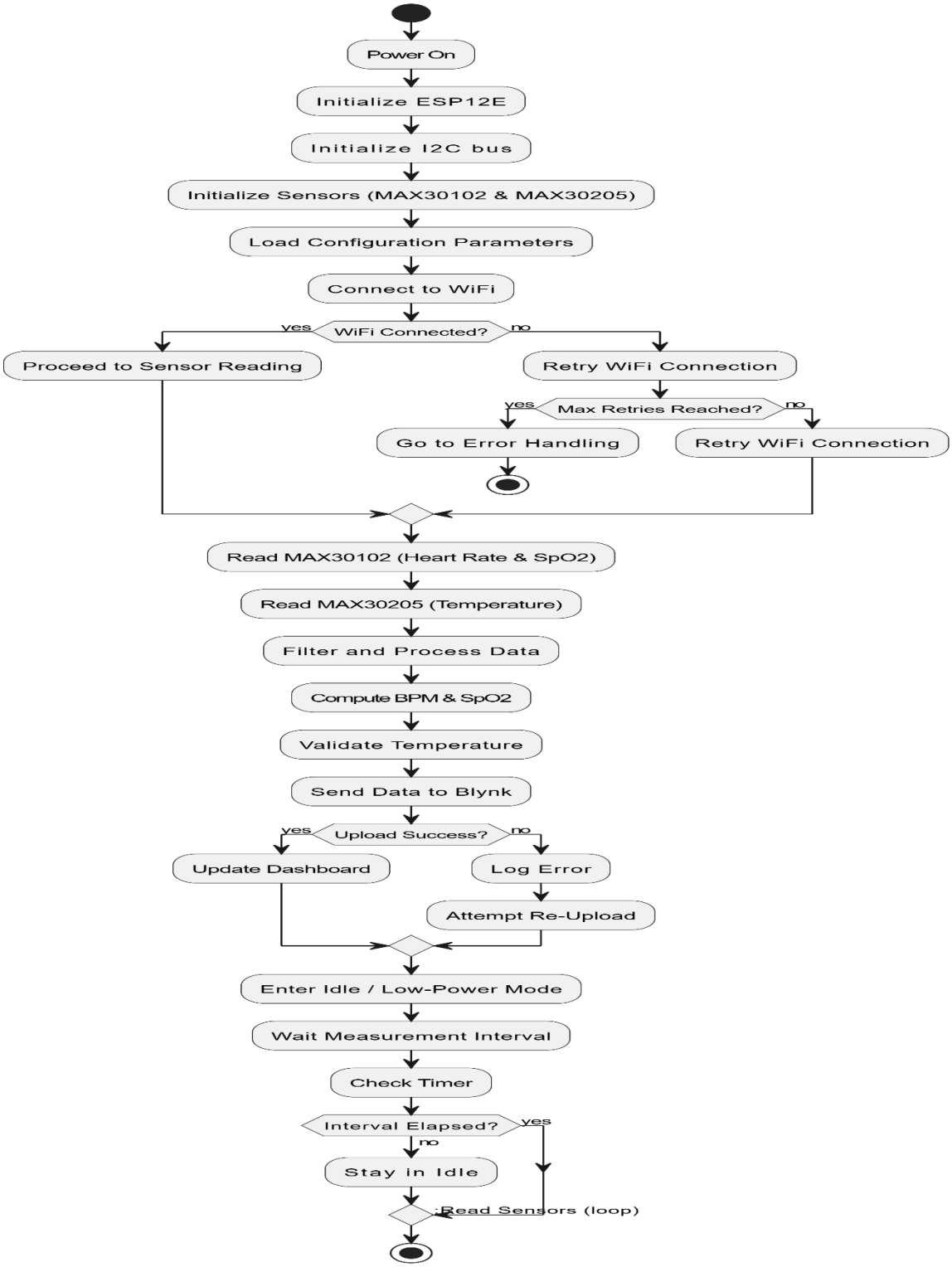


This sequence represents the system's state machine, beginning with the Initialization state where the ESP12E powers up, loads its configuration parameters, prepares peripherals, and establishes I²C communication with the MAX30102 and MAX30205 sensors. Once initialized, the system transitions to the WiFi_Connection state, attempting to connect to the configured network; a successful connection advances the state machine to Sensor_Read, while repeated failures redirect it to the Error state.

In the Sensor_Read state, the ESP12E gathers raw PPG data from the MAX30102 and temperature data from the MAX30205, then moves into the Data_Processing state, where the measurements are filtered, analyzed, and validated. After processing, the state machine enters the Send_To_Blynk state, transmitting the prepared values to the Blynk cloud platform.

A successful upload transitions the system to the Idle state, where it remains in a low-power wait mode until the next measurement interval, after which it loops back to Sensor_Read. If any critical problem occurs—such as Wi-Fi disconnection, I²C errors, or sensor faults—the state machine enters the Error state, which triggers the Recovery state. During Recovery, the system resets key modules, reinitializes communication interfaces, and returns to Initialization, ensuring continuous and reliable operation.

### *III.2.6.2 Activities diagram:*



The system begins by powering on the ESP12E module and allowing all voltage rails to stabilize. Once powered, the ESP12E initializes its GPIOs, ADCs, and timers, ensuring that all boot pins—EN, GPIO0, GPIO2, and GPIO15—are correctly configured. The $I^2C$ bus is then set up, with the ESP12E configuring the $I^2C$ peripheral, establishing the clock speed, and confirming that the bus is ready for communication.

Next, the sensors are initialized: the MAX30102 is configured with its LED currents, sample rate, and FIFO settings, while the MAX30205 is verified for proper addressing and used to obtain a baseline temperature reading. After hardware setup, the system loads configuration parameters such as Wi-Fi credentials, the Blynk token, measurement intervals, and alert thresholds.

The ESP12E then attempts to connect to the Wi-Fi network, retrying if necessary, and shifting into an error-handling routine if repeated failures occur. Once connected, the device begins reading sensor data—collecting PPG signals from the MAX30102 to derive raw heart-rate and $SpO_2$ values, and obtaining precise body temperature from the MAX30205. The data is then filtered and processed using smoothing algorithms, peak detection for BPM calculation, ratio computation for $SpO_2$, and validation of temperature readings to ensure they fall within a realistic human range.

After processing, the resulting values are formatted and transmitted to the Blynk cloud dashboard, with errors logged and retransmission attempted if uploads fail. The ESP12E then enters an idle or low-power state until the next measurement cycle, conserving energy when operating on battery. A timer manages the delay until the next interval, ensuring consistent periodic monitoring. Once the interval expires, the process returns to the sensor-reading stage, repeating continuously to provide uninterrupted real-time health data.

## III.2.7 Blynk IOT:

**Blynk** is an **Internet of Things (IoT) platform** that allows you to build web and mobile applications to control hardware remotely. Essentially, it acts as a bridge between hardware devices (like Arduino, ESP32, Raspberry Pi) and your smartphone or tablet**,** is a drag–and–drop programming system for the Internet of Things (IoT) that makes it much easier to develop IoT applications. Not only does it enable building programs using a simple drag–and–drop interface, but it also standardizes the connection of devices such as sensors and motors, ensuring that the necessary drivers are in place. In this way, it simplifies both the programming and hardware setup.

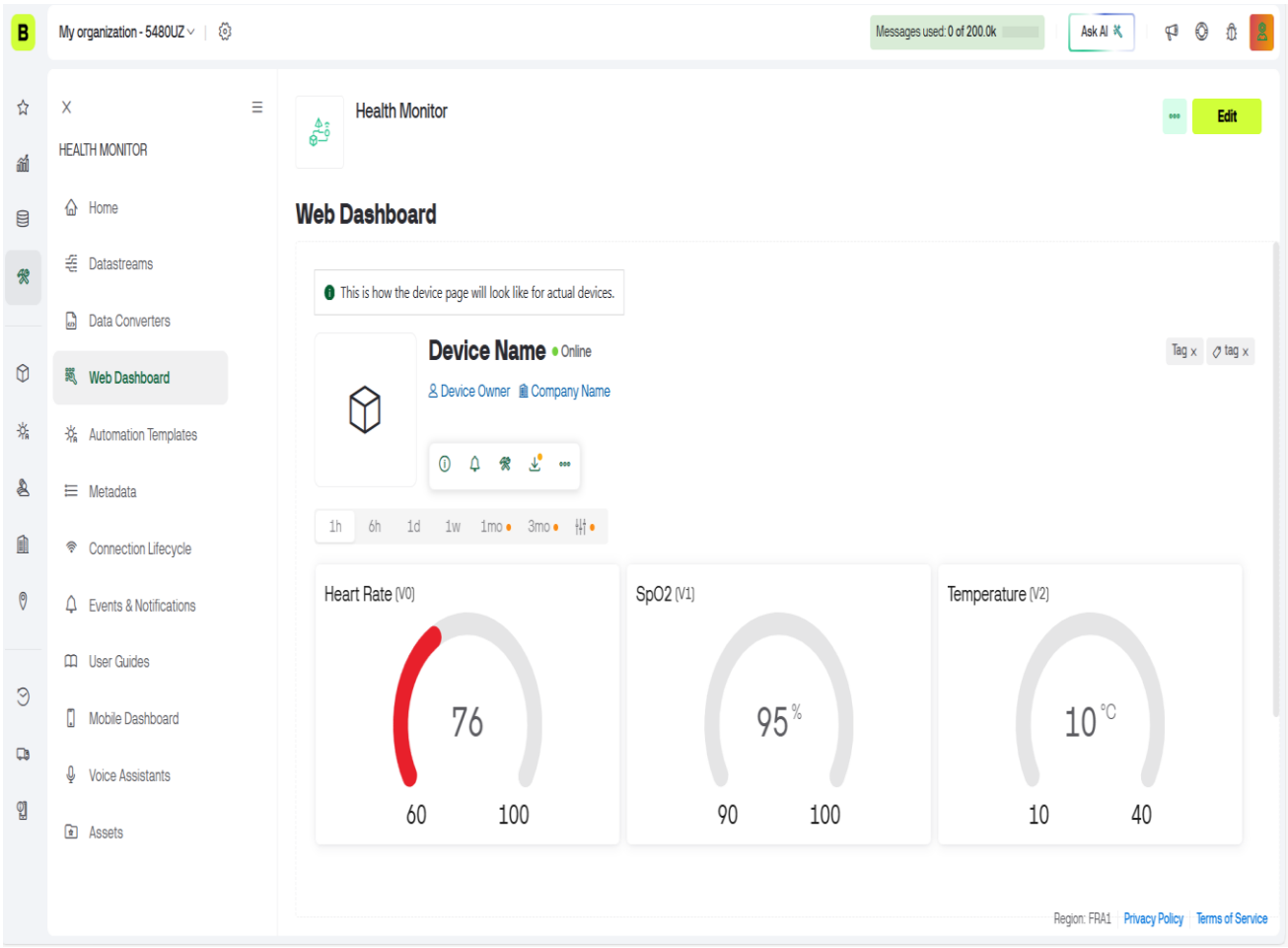
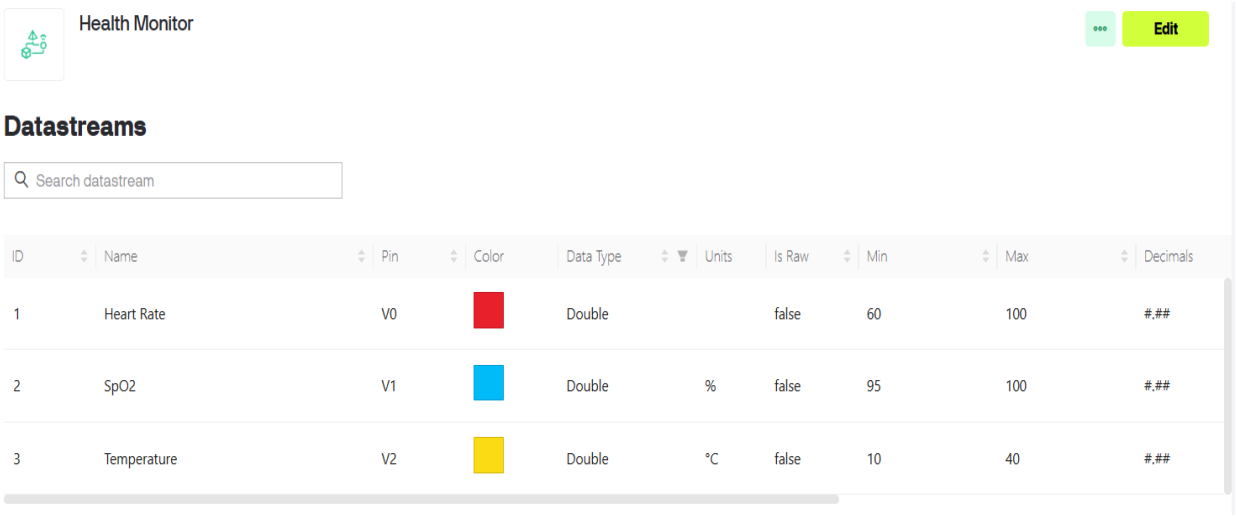



**Figure 3.31:** Blynk web dashboard.

In Blynk, **Data Streams** (also called Virtual Pins) are the channels through which data flows between your hardware (ESP-12E/Raspberry Pi) and the Blynk Cloud/App. Each data stream has a unique identifier (V0, V1, V2, etc.) and can be configured for specific data types, update frequencies, and value ranges.




| ID | Name | Pin | Color | Data Type | Units | Is Raw | Min | Max | Decimals |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Heart Rate | V0 | | Double | | false | 60 | 100 | #.## |
| 2 | SpO2 | V1 | | Double | % | false | 95 | 100 | #.## |
| 3 | Temperature | V2 | | Double | °C | false | 10 | 40 | #.## |

**Figure 3.32:** Data stream of sensors readings in the system.

## V0 - Heart Rate (BPM):

- **Data Type**: Integer
- **Range**: 30 - 220 BPM
- **Update Frequency**: Every 2-5 seconds
- **Direction**: Device → App (read-only)
- **Widget**: Gauge, Value Display, or SuperChart
- **Purpose**: Display current heart rate
- **Alert Thresholds**:
    - Low: < 50 BPM (bradycardia warning)
    - High: > 100 BPM (tachycardia warning)
- **Units**: BPM (Beats Per Minute)

## V1 - Blood Oxygen Saturation (SpO2):

- **Data Type**: Integer
- **Range**: 70 - 100 %

- **Update Frequency**: Every 2-5 seconds
- **Direction**: Device → App (read-only)
- **Widget**: Gauge, Value Display, Level
- **Purpose**: Display oxygen saturation level
- **Alert Thresholds**:
    - Critical: < 90% (hypoxemia)
    - Warning: < 95%
    - Normal: ≥ 95%
- **Units**: % (percentage)

## V2 - Body Temperature (°C):

- **Data Type**: Double/Float
- **Range**: 30.0 - 45.0 °C
- **Update Frequency**: Every 5-10 seconds
- **Direction**: Device → App (read-only)
- **Widget**: Gauge, Value Display, Level
- **Purpose**: Display body temperature
- **Alert Thresholds**:
    - Hypothermia: < 35.0°C
    - Normal: 36.1 - 37.2°C
    - Fever: > 38.0°C
    - High Fever: > 39.0°C
- **Units**: °C (Celsius) or °F (Fahrenheit)
- **Decimal Precision**: 1 decimal place

## Benefits of Blynk Integration:

- **Remote Monitoring**: Caregivers/family can monitor vital signs from anywhere
- **Historical Data**: Cloud storage of trends and patterns
- **Instant Alerts**: Push notifications for concerning values
- **Remote Control**: Adjust settings, trigger measurements, control robot
- **Multi-User Access**: Multiple family members/healthcare providers can access

- **Cross-Platform**: iOS, Android, and Web access
- **Easy Setup**: No need to develop custom mobile app
- **Data Export**: Can export historical data for medical records

## III.2.8 Coding:

### 1. WIFI setup:

First, the ESP8266 loads the WiFi credentials into its WiFi stack and powers on the WiFi radio to enable the module. The device is then set to STATION mode, operating as a client rather than an access point. It begins scanning for available networks that match the configured SSID, and once the target network is detected, the ESP8266 initiates the connection process.

```
Serial.print("Connecting to WiFi");
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("\nWiFi connected!");
Serial.print("IP: ");
Serial.println(WiFi.localIP());
```

### 2. Blynk setup:

The **Blynk.begin** function initializes the connection to the Blynk server using the provided authentication token, **WiFi SSID**, and password. After the connection is established, two timer tasks are set up: the first timer calls **sampleAndProcessData** every 1 second to collect and process sensor data, while the second timer calls **sendToBlynk** every 5 seconds to transmit the processed data to the Blynk server.

```
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

// Timer tasks
timer.setInterval(1000L, sampleAndProcessData); // collect + process data every 1s
timer.setInterval(5000L, sendToBlynk);          // send data every 5s
```

## 3. MAX30205 reading:

To select the temperature register on the MAX30205, the master first sends a START condition on the I²C bus, followed by the device address (0x48) with the write bit (0), forming the full address byte 0x90. After waiting for an acknowledgment (ACK) from the sensor, the master sends the register address 0x00 to select the temperature register and again waits for an ACK.

The master then terminates the write transaction by sending a STOP condition. To read the temperature, the master issues a repeated START condition, followed by the device address with the read bit (1), forming the full address byte 0x91, and waits for an ACK from the MAX30205. It then reads the most significant byte (MSB) of the temperature data, sending an ACK to indicate that it will continue reading. Next, it reads the least significant byte (LSB) and sends a NACK to signal that no further data is needed. Finally, the master ends the transaction by sending a STOP condition.

```
void readTemperature() {
  Wire.beginTransmission(MAX30205_ADDRESS);
  Wire.write(0x00); // Temperature register
  Wire.endTransmission();
  Wire.requestFrom(MAX30205_ADDRESS, 2);
  if (Wire.available() == 2) {
    uint8_t msb = Wire.read();
    uint8_t lsb = Wire.read();
    int16_t raw = (msb << 8) | lsb;
    temperature = raw / 256.0f;
  }
}
```

## 4. MAX30102 reading:

The function **maxim_heart_rate_and_oxygen_saturation** processes the raw IR and Red LED data collected from the MAX30102 sensor. It analyzes the IR buffer to determine heart-rate peaks and uses both IR and Red buffers to calculate the blood oxygen saturation ($SpO_2$). The results are stored in the provided variables, where **spo2** and **heartRate** contain the computed values, and **validSPO2** and **validHeartRate** indicate whether each measurement is reliable. This function essentially converts raw photoplethysmography (PPG) samples into meaningful heart-rate and $SpO_2$ readings.

```
// Collect 100 samples (~4 sec worth of data)
const int samples = 100;

for (int i = 0; i < samples; i++) {
  while (!particleSensor.check()) ; // Wait for new data
  redBuffer[i] = particleSensor.getRed();
  irBuffer[i] = particleSensor.getIR();
}

// Run MAXIM algorithm (from the working version)
maxim_heart_rate_and_oxygen_saturation(irBuffer, samples, redBuffer, &spo2, &validSPO2, &heartRate, &validHeartRate);
```

# III.3 Implementation of medical assistant:

## III.3.1 Functional Description:

The Health Companion Robot is an intelligent, interactive robotic platform designed for continuous monitoring of vital physiological parameters (heart rate, blood oxygen saturation, and body temperature) combined with physical embodiment capabilities for enhanced user interaction and engagement.

The system integrates biomedical sensing, embedded computing, servo-based articulation, and cloud connectivity to provide a comprehensive health monitoring solution suitable for elderly care, chronic disease management, and independent living support.

The robot comprises two primary processing units working in coordinated fashion: an ESP-12E microcontroller module dedicated to real-time sensor data acquisition and preprocessing, and a Raspberry Pi Zero 2W application processor responsible for high-level decision making, servo control, user interaction, and cloud communication. This distributed architecture ensures deterministic sensor timing while enabling sophisticated behavioral capabilities through the robotic platform.

## III.3.2 Hardware Architecture:

### *III.3.2.1 Battery Configuration (2× 18650 in Series)*

The robot's primary power source consists of two 18650 lithium-ion rechargeable batteries connected in series configuration. Each 18650 cell has a nominal voltage of 3.7V and typical

capacity ranging from 2500mAh to 3500mAh depending on the specific cell model selected. When connected in series (positive terminal of first cell to negative terminal of second cell), the configuration produces a nominal voltage of 7.4V with the same capacity as a single cell.

The voltage characteristics of this 2S (two-series) battery pack are as follows: fully charged voltage reaches 8.4V (2× 4.2V per cell), nominal operating voltage is 7.4V (2× 3.7V), and minimum safe discharge voltage is approximately 6.0V (2× 3.0V per cell). Discharging below 3.0V per cell risks permanent damage to the lithium-ion chemistry. With a typical 3000mAh capacity, this battery pack provides approximately 22.2 watt-hours of energy (7.4V × 3000mAh), which translates to 1.5 to 2.5 hours of continuous robot operation depending on servo activity and computational load.

The series connection is accomplished by connecting the positive (+) terminal of the first 18650 cell to the negative (-) terminal of the second cell. The remaining free positive terminal becomes the battery pack's positive output, while the remaining free negative terminal becomes the pack's negative output. This configuration delivers higher voltage while maintaining the same current capacity, which is ideal for powering 5V systems through a buck converter with minimal conversion losses.

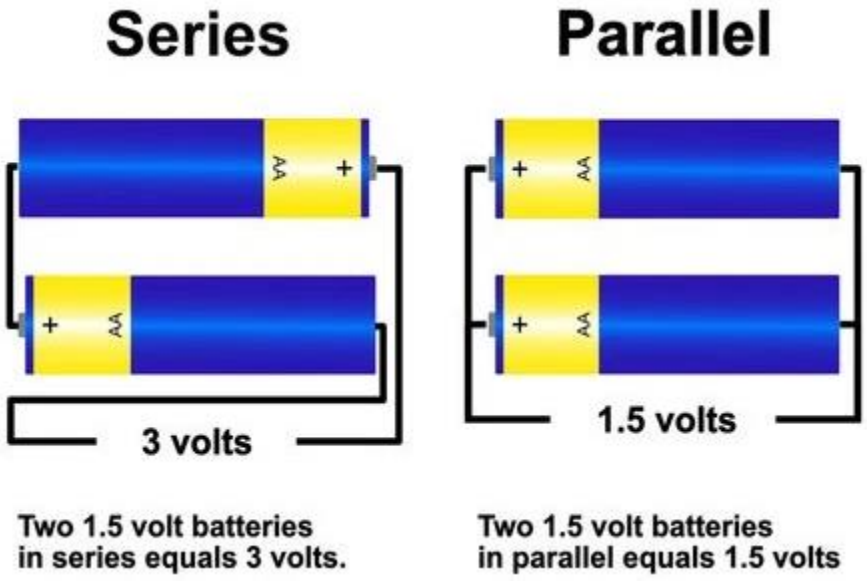



**Figure 3.33:** battery montage types

### *III.3.2.2 SY8205 DC-DC Buck Converter:*

The SY8205 is a high-efficiency synchronous buck (step-down) DC-DC converter that regulates the variable battery voltage (6.0V to 8.4V) down to a stable 5V output required by the Raspberry Pi, servos, and camera. This integrated circuit employs synchronous rectification, meaning it uses a MOSFET instead of a diode for the rectification phase, achieving conversion efficiencies typically exceeding 90-95% depending on load conditions.

The converter accepts an input voltage range of 4.5V to 18V, making it fully compatible with our 2S battery configuration across its entire discharge range. It provides a fixed 5V output with ±2% accuracy and can deliver up to 5A continuous current, sufficient for the robot's peak power demands. The SY8205 operates at a switching frequency of approximately 1.4MHz, which allows for smaller external inductor and capacitor components compared to lower-frequency converters.

The Raspberry Pi Zero 2W receives regulated 5V power from the SY8205 converter through one of two possible connection methods. The primary method uses the micro-USB port labeled "PWR IN" on the board, which internally connects to the 5V power rail through a protection circuit. Alternatively, power can be supplied directly to GPIO header Pin 2 or Pin 4 (both are 5V pins), bypassing the micro-USB connector entirely. When using GPIO power injection, careful attention must be paid to polarity, as incorrect connection can permanently damage the board. Ground connects to any of the multiple GND pins on the GPIO header (Pins 6, 9, 14, 20, 25, 30, 34, or 39).

The PCA9685 requires two separate power supplies: a logic supply (VDD) for the chip's internal circuitry and I²C communication, and a servo power supply (V+) for driving the servo motors' power pins. This dual-supply architecture isolates the noise-prone servo motors from the sensitive digital logic.

### 3.3.2.3 I²C Connection to PCA9685:

The Raspberry Pi Zero 2W communicates with the PCA9685 servo driver through its hardware I²C interface (I²C1), which is available on GPIO pins 2 and 3 of the 40-pin header. This is a two-

wire serial communication protocol consisting of a data line (SDA) and a clock line (SCL), both of which require pull-up resistors to function correctly.

GPIO Pin 3 (physical pin 3 on the header) serves as SDA (Serial Data), carrying bidirectional data between the Raspberry Pi (master) and PCA9685 (slave). GPIO Pin 5 (physical pin 5 on the header) serves as SCL (Serial Clock), providing the timing reference generated by the Raspberry Pi as the bus master. Both signals operate at 3.3V logic levels, which is compatible with the PCA9685's VDD logic supply.

The connection requires four wires between the Raspberry Pi and PCA9685 module:

- **SDA wire**: Raspberry Pi GPIO2 (Pin 3) connects to PCA9685 SDA terminal
- **SCL wire**: Raspberry Pi GPIO3 (Pin 5) connects to PCA9685 SCL terminal
- **Power wire**: Raspberry Pi 3.3V (Pin 1) connects to PCA9685 VDD terminal
- **Ground wire**: Raspberry Pi GND (Pin 6 or any GND pin) connects to PCA9685 GND terminal

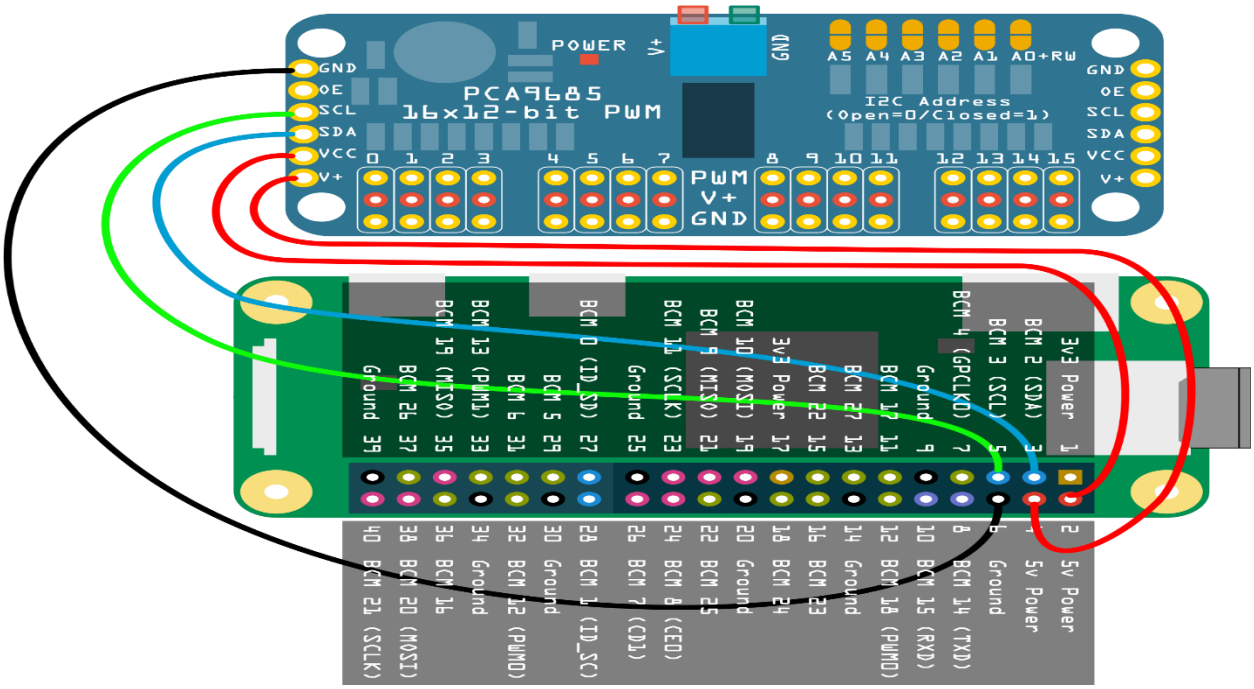



**Figure 3.34:** Wiring the PCA9685 to Raspberry Pi Zero.

### 3.3.2.4 CSI Connection to Pi Camera Rev 1.3:

The Pi Camera V2 Revision 1.3 connects to the Raspberry Pi Zero 2W through the dedicated CSI (Camera Serial Interface) port using a 15-pin flat flexible cable (FFC). This is a high-speed differential serial interface specifically designed for camera modules, supporting the MIPI CSI-2 protocol with two data lanes capable of transferring up to 1 Gbps of image data.

The CSI connector on the Raspberry Pi Zero 2W is located between the HDMI port and the micro-USB ports, identifiable as a thin horizontal slot with a pull-up plastic locking tab. To install the camera cable, the locking tab is gently lifted upward (not pulled out, just lifted approximately 2-3mm), the FFC is inserted with the blue side (where contacts are visible) facing upward toward the USB ports, and the tab is pressed down firmly to lock the cable in place. Proper insertion ensures the cable sits evenly in the connector with no gaps on either side.

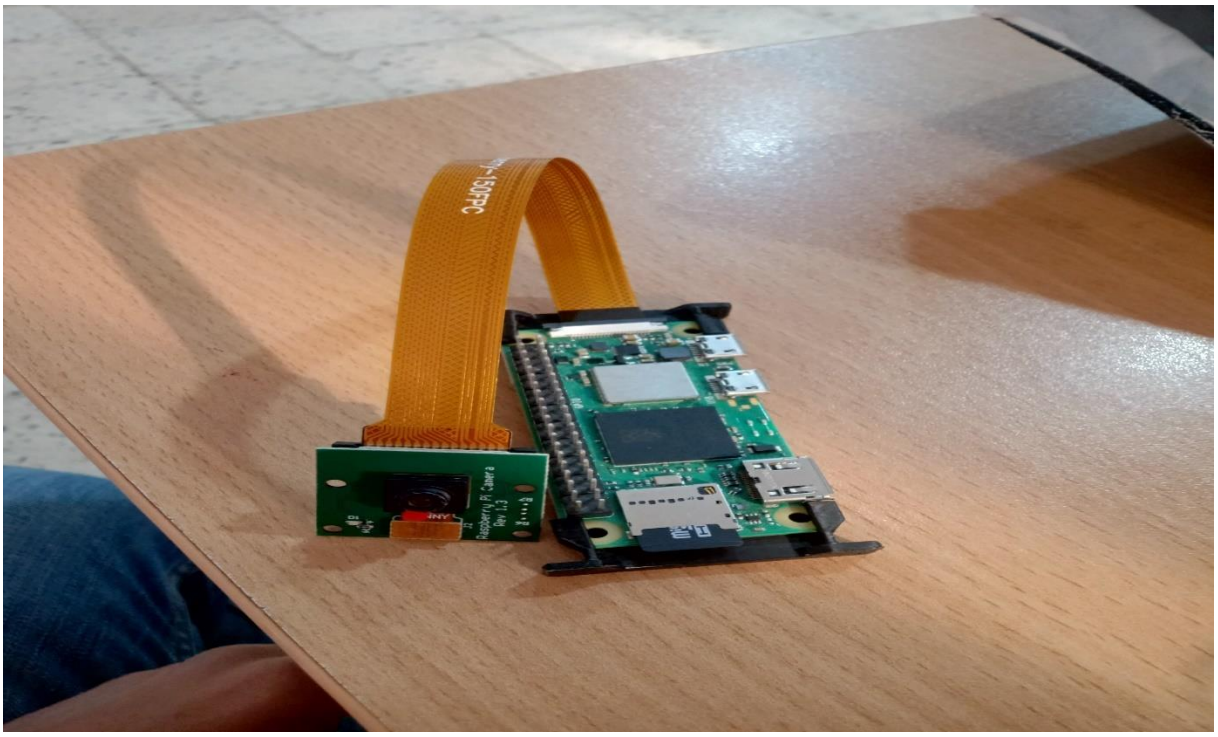


**Figure 3.35:** wiring the PI camera to PI zero 2w

### 3.3.2.5 PWM Output Configuration:

The PCA9685 provides 16 independent PWM output channels numbered 0 through 15. Each channel consists of a three-pin header labeled with the channel number (0-15), with pins arranged as Signal (PWM output), V+ (power pass-through), and GND (ground pass-through). For this robot application, eight of these channels (0-7) are utilized for the eight SG90 servos.

The PWM signals are generated by the PCA9685's internal 12-bit counter driven by a 25MHz oscillator. For servo control, the PWM frequency is configured to 50Hz (20ms period), which is the standard frequency expected by hobby servos. Within each 20ms cycle, the servo signal pin must receive a pulse whose width determines the servo's position: 1ms pulse commands 0° position, 1.5ms commands 90° (center), and 2ms commands 180°. The 12-bit counter provides 4096 discrete steps across the 20ms period, yielding approximately 4.88µs resolution per step—more than sufficient for smooth servo positioning.

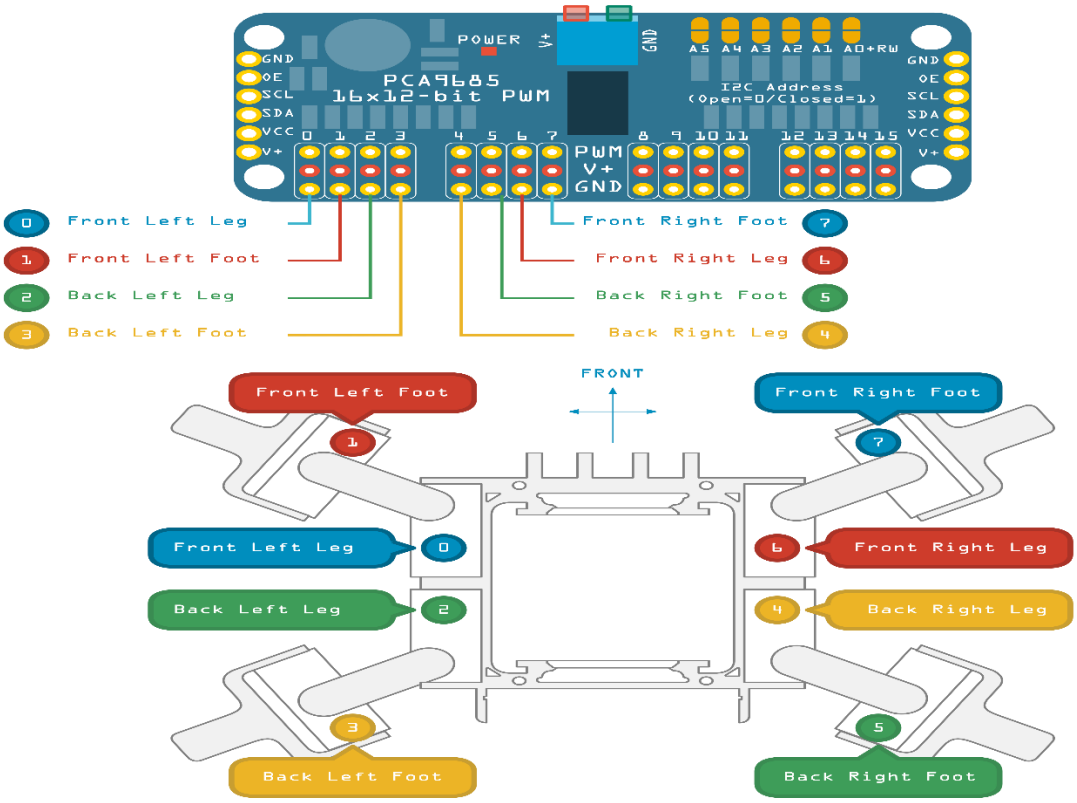
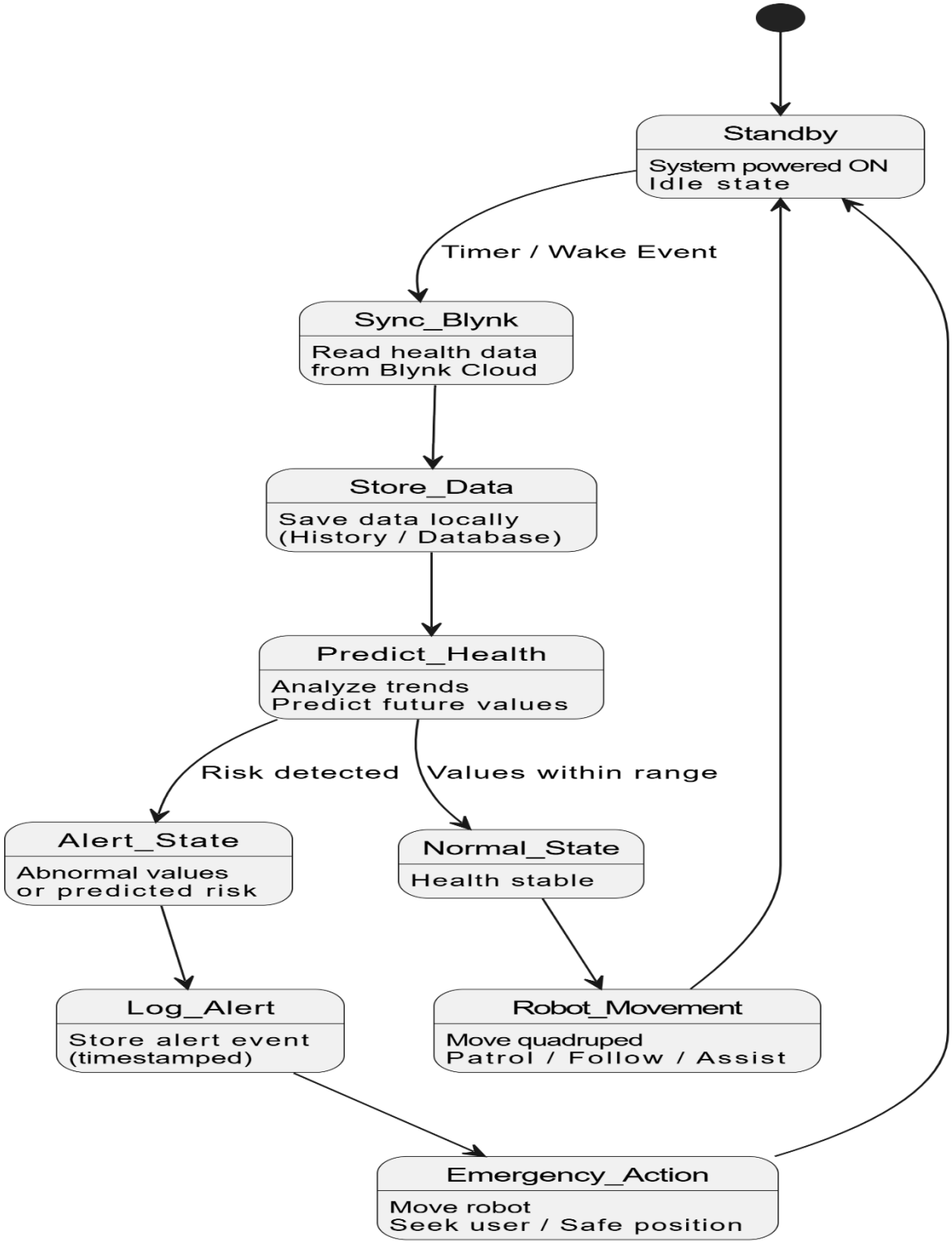



**Figure 3.36:** Wiring the Servos to the PCA9685 board .

## III.3.3 UML Modeling:

### *III.3.3.1 State machine diagram:*

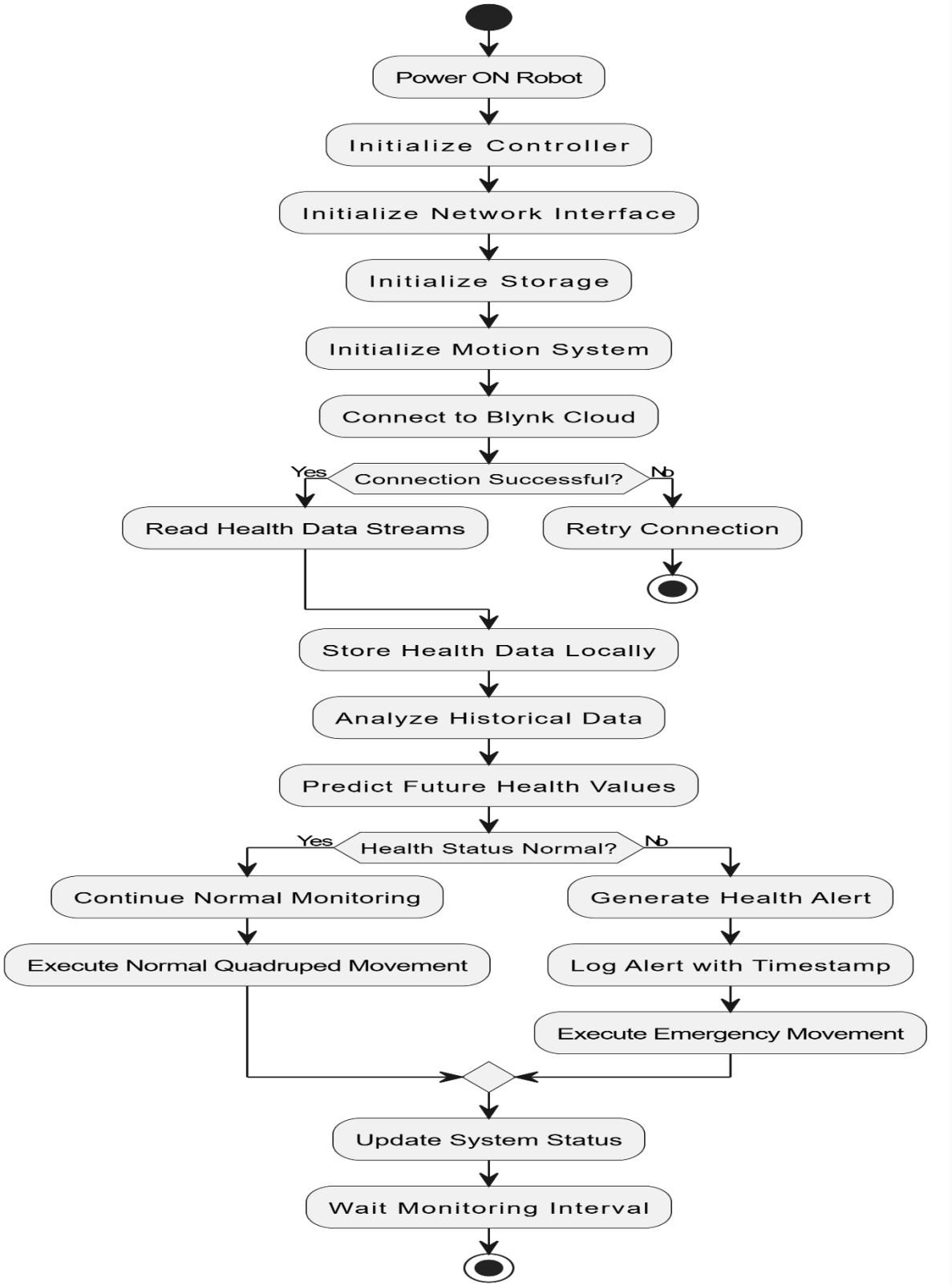


This description explains the operation of the robot using a state machine model. After power-up, the system enters the Standby state, where all subsystems are initialized and ready, but no active processing occurs. In this state, the robot waits for a scheduled health-check event while maintaining readiness and minimizing power consumption. When triggered, the state machine transitions to Sync_Blynk, during which the robot connects to the Blynk cloud platform and retrieves the latest available health data, including heart rate, $SpO_2$, body temperature, and any enabled historical values, ensuring that decisions are based on the most recent information. The system then moves to the Store_Data state, where the retrieved data is saved locally with timestamps, historical trends, and previous alert records to support long-term monitoring and predictive analysis.

Next, in the Predict_Health state, the robot analyzes the stored data to identify trends such as rising temperature, falling $SpO_2$, or abnormal heart-rate patterns, applying simple predictive methods like moving averages, regression, or lightweight machine-learning models to classify health risk levels. If the predicted health status is within safe limits, the state machine enters the Normal_State, where no alerts are generated and the robot proceeds with routine behavior. This leads to the Robot_Movement state, in which the quadruped robot performs movement tasks such as patrolling the nearby area, following the user, maintaining proximity for continuous monitoring, or navigating to a charging station, providing physical presence and assistance without verbal interaction.

If abnormal or potentially dangerous health conditions are detected or predicted, the system transitions to the Alert_State, preparing a safety response. The robot then enters the Log_Alert state to record alert details—including the alert type, sensor values, prediction results, and timestamps—either locally or in the cloud for medical traceability and caregiver review. Following this, the Emergency_Action state is activated, during which the robot executes predefined physical responses such as moving toward the user, positioning itself in a safe zone, or activating visual indicators like LEDs to attract attention, without using speech or audio feedback. After completing the required movement or alert response, the state machine returns to the Standby state, where the robot awaits the next monitoring cycle.

### *III.3.3.2 Activities diagram:*



This description represents the system behavior as an **activity diagram** for the health companion robot. When the robot is powered on, all hardware components receive power and the system begins its startup sequence to prepare for monitoring, analysis, and movement operations. The robot first initializes its main controller by loading firmware, setting up timers and internal variables, and establishing default operational states. It then initializes the network interface by enabling Wi-Fi, loading network credentials, and preparing cloud communication protocols. Local storage is also initialized to support long-term monitoring, including the storage of health data history, prediction results, and alert logs.

Next, the quadruped motion system is initialized by configuring motor drivers, calibrating servos, and setting gait parameters to ensure stable and controlled movement. Once the hardware and subsystems are ready, the robot connects to the Blynk cloud platform to access real-time and historical health data, retrying the connection if necessary to maintain reliable operation. After a successful connection, the robot retrieves the latest health data streams, including heart rate, blood oxygen level, and body temperature, and stores the collected data locally with timestamps to enable trend analysis, prediction modeling, and alert tracking.

The system then analyzes the historical health data to detect trends, identify abnormal variations, and compare measurements against safe thresholds. Based on this analysis, the robot predicts future health conditions using techniques such as moving averages, regression, or lightweight machine-learning models to identify potential risks early. A decision is then made regarding the user's health status: if the predicted state is normal, no alert is generated and the robot proceeds with routine quadruped movement behaviors such as following the user, patrolling the nearby area, or maintaining a monitoring presence.

If an abnormal condition is detected or predicted, the system generates a health alert, logs the alert details—including the type, health values, prediction results, and timestamp—for traceability and medical review, and executes predefined emergency movement actions such as moving toward the user or positioning itself visibly in a designated alert location, without using speech or sound. After completing either the normal or emergency actions, the robot updates its

internal system status and enters a waiting phase, pausing until the next scheduled monitoring interval to ensure continuous and efficient health monitoring.

## III.3.4 Coding:

### 1. Blynk configuration:

```
BLYNK_TOKEN = "8dhfVCtpMv4aZXTSwqNOXs5ldEInusJj"
BASE_URL = f"http://blynk-cloud.com/{BLYNK_TOKEN}"

VP_HR = "V0"
VP_SPO2 = "V1"
VP_TEMP = "V2"
```

Define Blynk device identity so the Pi knows which ESP12E device data it is allowed to read, and ensures the Pi reads heart rate, $SpO_2$, and temperature correctly

### 2. Initialize servo controller (PCA9685):

```
i2c = busio.I2C(SCL, SDA)
pca = PCA9685(i2c)
pca.frequency = 50
```

Connects the Pi to the PCA9685 board via $I^2C$, Enables stable control of 8 SG90 servos at correct PWM frequency.

### 3. Read health data from Blynk Cloud:

```
def get_blynk(vpin):
    try:
        r = requests.get(f"{BASE_URL}/get/{vpin}", timeout=6)
        if r.status_code == 200:
            data = r.json()
            if isinstance(data, list) and data:
                return float(data[0])
    except Exception as e:
        print("Blynk error:", e)
    return None
```

Sends HTTP requests to Blynk Cloud to retrieve sensor data. This is the ONLY part where the Pi receives data from the ESP.

### 4. Predict future health values:

```
def predict_next(values):
    if len(values) < 3:
        return None
    x = np.arange(len(values))
    y = np.array(values)
    model = np.polyfit(x, y, 1)
    poly = np.poly1d(model)
    return float(poly(len(values)))
```

Uses simple linear regression to estimate future health readings. Allows early detection of worsening conditions.

### 5. Detect abnormal health conditions

```
def check_health(hr, spo2, temp):
    issues = []

    if hr is not None and (hr < HR_MIN or hr > HR_MAX):
        issues.append(f"HR out of range ({hr})")

    if spo2 is not None and spo2 < SPO2_MIN:
        issues.append(f"Low SpO2 ({spo2})")

    if temp is not None and (temp < TEMP_MIN or temp > TEMP_MAX):
        issues.append(f"Temp abnormal ({temp})")

    return issues
```

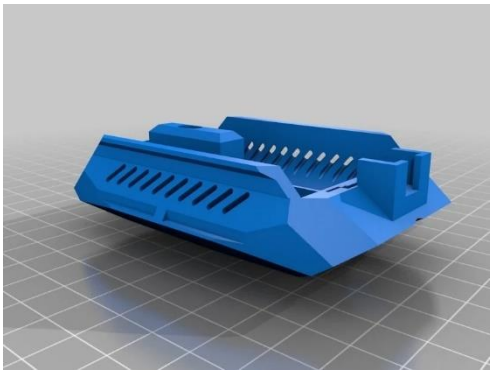
Compares current and predicted values to medical thresholds. Generates warnings without using physical alerts

## III.3.5 3D Modeling and Visualization:

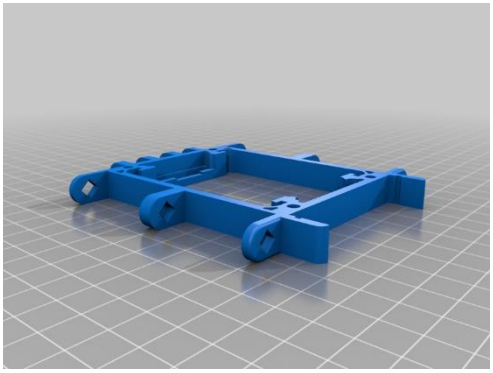
The SMARS Quadruped is a modification that transforms the SMARS robot into a four-legged walking robot using servos and smart design.

### Printable Parts:

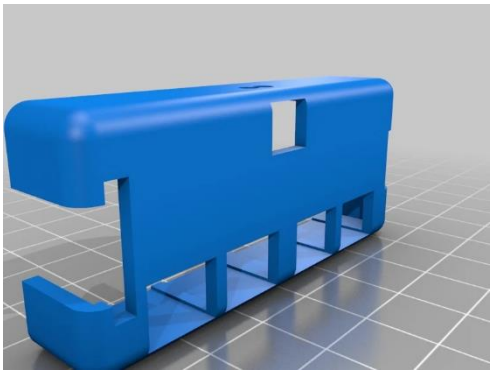
- **1x Quad Power Shell**



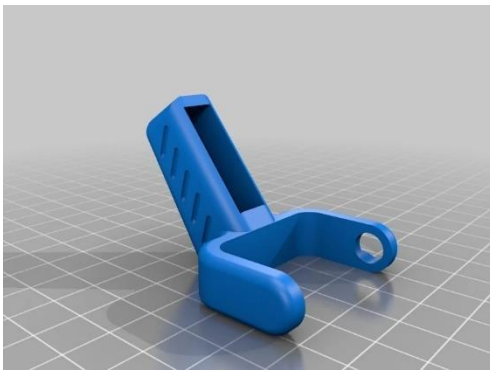
- **1x Frame** (1.5 Pi for chassis_S, 2 Pi for 4WD chassis)



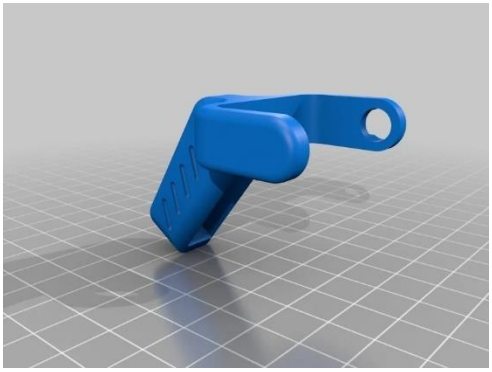
- **1x Servo Driver Cover**



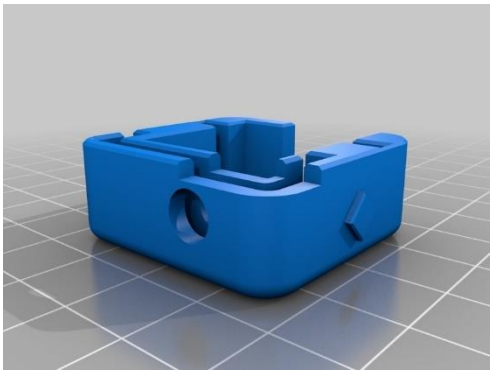
- **2x Foot**



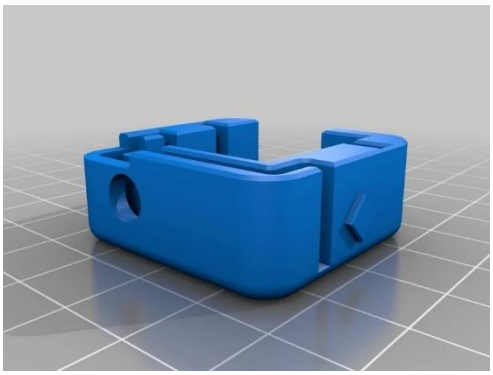
- **2x Foot_M**



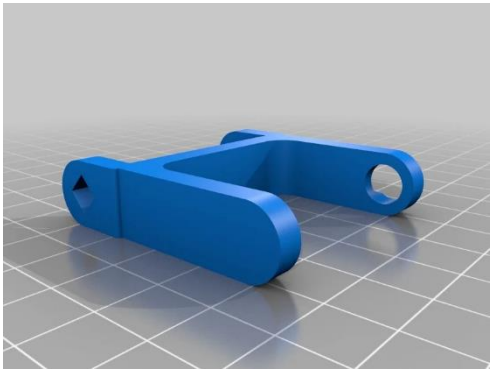
- **4x Servo Case**



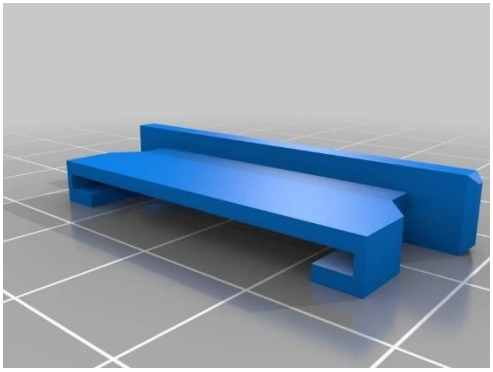
- **4x Servo Case_M**



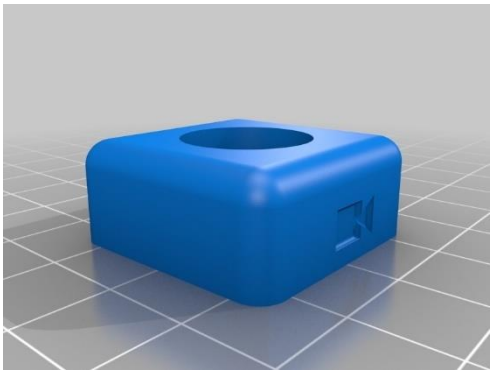
- **4x Servo Arm**



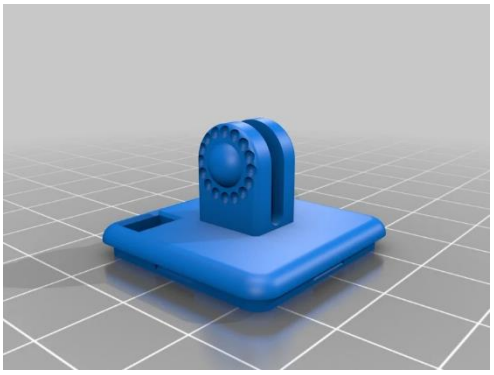
- **2x Pi Holder V2** (if using a Raspberry Pi)



- **Top_camera**



- **Bottom_camera**



**Fit the servo into the servo holder:**

The SG90 size servos will fit snuggly into the servo holder. You can push the servo wire into the channel within the holder to neatly guide the wire. If the type of servo you have has a wire that comes directly out of the bottom, you can easily clip away some of the channel wall so that the wire fits in.

**Fit the servo holders into the frame:**

The servo holder can be pushed into the frame so that the wires and the servo are locked into place. The small diamond shaped lugs on the servo holders will locate into the holes in the frame and click into place when pushed in.

**Mount the servo holder in the servo arm:**

Insert the servo horn onto the servo and screw in securely to the servo arm.

**Mount the foot on the servo holder:**

Adding the foot to the servo servo holder is simple, make sure the servo can rotate in both directions before you screw in the servo horn. You can check the precise angle of these later using the test.py program.

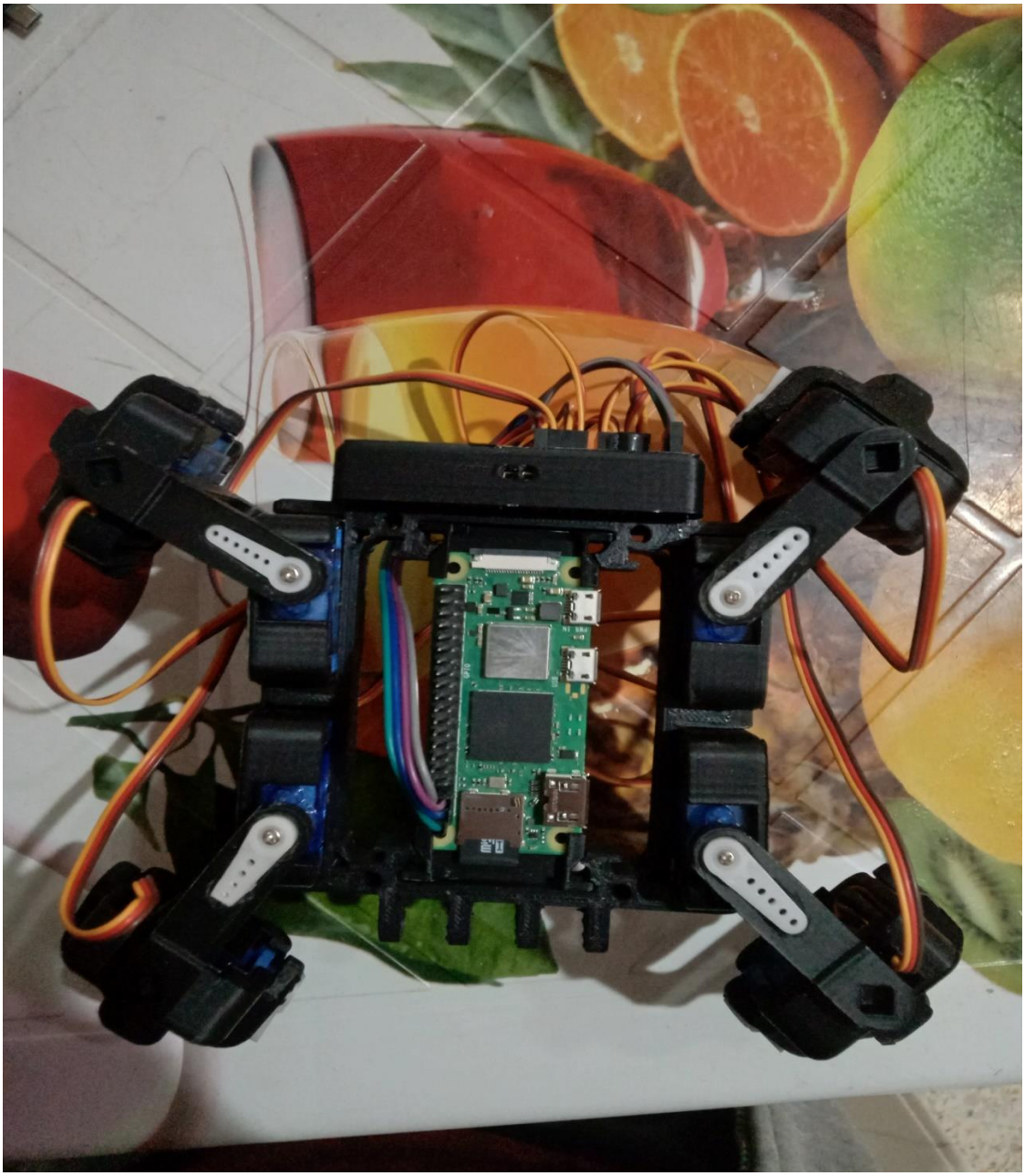


**Figure 3.37:** assembling of the robot

## III.4 Conclusion:

This chapter has presented the complete implementation process of the proposed system, detailing how the design specifications were translated into a fully functional application. It discussed the development environment, tools, technologies, and methodologies used to construct the system and integrate its various components effectively. Each module was implemented and tested to ensure proper functionality and seamless interaction within the system.

The successful implementation demonstrates that the proposed design is feasible and capable of meeting the defined requirements. This chapter confirms that the system operates as intended and provides a solid foundation for further testing, evaluation, and future enhancements. The outcomes of this phase serve as a critical step toward validating system performance and readiness for real-world deployment.

# IV. CONCLUSION AND FUTURE DEVELOPMENT

# IV.1 Conclusion:

This project presents a comprehensive and innovative approach to addressing one of healthcare's most pressing challenges: providing continuous, non-intrusive vital sign monitoring combined with meaningful human-robot interaction for elderly care, chronic disease management, and independent living support. By integrating biomedical sensing technology with robotic embodiment and cloud connectivity, the system transcends traditional health monitoring devices to create a truly interactive health companion that serves both functional and emotional needs.

The intelligent embedded system successfully combines multiple technological domains into a cohesive, practical solution. At its core, the dual-processor architecture strategically distributes computational responsibilities: the ESP-12E microcontroller handles real-time sensor data acquisition from the MAX30102 pulse oximetry sensor and MAX30205 clinical thermometer with deterministic timing guarantees, while the Raspberry Pi Zero 2W manages high-level processing, servo control for robotic articulation, camera-based interaction, and cloud communication through the Blynk IoT platform. This separation of concerns ensures reliable physiological measurements while enabling sophisticated behavioral capabilities.

The sensor subsystem provides medical-grade monitoring of three critical vital signs—heart rate, blood oxygen saturation (SpO2), and body temperature—using compact, integrated sensors that require minimal user interaction. The MAX30102's photoplethysmography technology delivers accurate cardiovascular measurements through simple finger placement, while the MAX30205 provides clinical-accuracy temperature readings. The embedded signal processing algorithms extract meaningful health metrics from raw sensor data, implementing filtering, peak detection, and the Maxim integrated algorithm for robust SpO2 calculation even in the presence of motion artifacts and ambient light interference.

The robotic platform transforms abstract health data into physical, relatable interaction through eight SG90 micro servos controlled via the PCA9685 PWM driver. The servo allocation—spanning head pan/tilt for attention direction, sensor arm positioning for automated measurement, display adjustment for user-centered viewing, and dual-arm articulation for expressive gestures—provides sufficient degrees of freedom for meaningful social interaction

without excessive complexity. The Pi Camera V2 adds visual perception capabilities, enabling face recognition for personalized interaction, gesture-based control, telemedicine video conferencing, and environmental awareness.

Cloud connectivity through the Blynk platform extends the system's reach beyond the immediate user to caregivers, family members, and healthcare providers. Real-time vital sign streaming, historical trend visualization, configurable alerts, and remote control capabilities transform the robot from a standalone device into a node in a comprehensive care ecosystem. The carefully designed data stream architecture (Virtual Pins V0-V35) provides intuitive access to sensor readings, system status, and control functions through mobile applications and web dashboards accessible from anywhere with internet connectivity.

The power system architecture, based on dual 18650 lithium-ion cells in series configuration regulated by an efficient SY8205 buck converter, provides 1.5-2.5 hours of continuous operation in a compact, portable form factor. The careful power budget analysis and distribution strategy ensures stable 5V supply to all subsystems even during peak servo current demands, with adequate bulk capacitance preventing voltage droop during transients.

## IV.2 Future Development:

**Advanced Sensing:** Integration of additional biomedical sensors such as electrocardiogram (ECG) for detailed cardiac monitoring, continuous blood pressure measurement through pulse transit time or photoplethysmography-based estimation, respiration rate through impedance pneumography or camera-based chest movement analysis, galvanic skin response for stress/anxiety detection, and environmental sensors (air quality, temperature, humidity) for context-aware health assessment.

**Machine Learning and AI:** Implementation of on-device or cloud-based machine learning models for personalized baseline learning (understanding individual normal ranges), anomaly detection beyond simple thresholds (recognizing subtle multi-parameter patterns), predictive analytics forecasting health deterioration before acute events, activity recognition from camera

data, emotion recognition from facial expressions and voice, and natural language processing for conversational interaction.

**Enhanced Interaction Modalities:** Voice-based interaction with wake-word detection and natural language understanding, advanced gesture recognition for touchless control, emotion-aware responses adapting behavior to user mood, proactive interaction initiating check-ins rather than only responding to commands, and gamification encouraging engagement and adherence through achievements and challenges.

**Mobility and Navigation:** Adding wheeled or tracked locomotion with obstacle avoidance for autonomous room-to-room movement, following the user around the home, returning to charging dock independently, and navigating to optimal positions for interaction. This significantly expands utility but adds complexity, cost, and safety considerations.

# ANNEXE

## Problem to be solved:

Many elderly, chronically ill, or physically vulnerable individuals lack continuous, real-time health monitoring and immediate assistance in their daily environments. Traditional medical monitoring systems are often **stationary, intrusive, expensive, or dependent on constant human supervision**, making them unsuitable for home use. As a result, **early warning signs of health deterioration (such as abnormal heart rate, temperature, oxygen saturation, or falls)** can go undetected, leading to delayed medical intervention and increased health risks.

Additionally, patients may experience **loneliness, poor medication adherence, and difficulty communicating health issues**, while caregivers face challenges in providing round-the-clock monitoring.

Therefore, the core problem is the **absence of an intelligent, autonomous, and user-friendly system** that can:

- Continuously monitor vital signs in real time
- Detect anomalies and emergencies early
- Provide immediate feedback or alerts
- Assist users emotionally and physically through interaction
- Reduce the burden on caregivers and healthcare providers

An intelligent embedded system integrated into a **health companion robot** is needed to address these gaps by offering **continuous monitoring, intelligent decision-making, and supportive human–robot interaction in a home environment**.

## Customers Segments:

- **Elderly Individuals Living Alone**: Seniors who require continuous health monitoring, fall detection, and companionship to maintain safety and independence at home.
- **Patients with Chronic Diseases**: Individuals suffering from conditions such as heart disease, diabetes, or respiratory disorders who need regular vital sign monitoring and early warning alerts.

- **Home Caregivers and Family Members**: Relatives or informal caregivers who need real-time health updates and alerts to ensure the safety of loved ones remotely.
- **Hospitals and Healthcare Providers**: Medical institutions seeking remote patient monitoring solutions to reduce hospital visits, readmissions, and workload on healthcare staff.
- **Nursing Homes and Assisted Living Facilities**: Organizations that require scalable monitoring systems to track multiple residents' vital signs and improve care efficiency.
- **Health Insurance and Telemedicine Companies**: Companies interested in preventive healthcare solutions that lower long-term medical costs through early detection and continuous monitoring.

# Value Proposition:

- **Continuous Real-Time Health Monitoring**: Provides 24/7 monitoring of vital signs such as heart rate, body temperature, $SpO_2$, and activity levels, enabling early detection of health abnormalities.
- **Early Emergency Detection and Alerts**: Automatically identifies critical conditions (falls, abnormal vitals) and instantly alerts caregivers or healthcare providers, reducing response time.
- **Personalized Health Assistance**: Adapts to individual user health profiles, offering reminders for medication, hydration, and daily activities based on personal needs.
- **Reduced Caregiver Burden**: Minimizes the need for constant human supervision by providing reliable autonomous monitoring and reporting.
- **Enhanced Emotional Support and Companionship**: Acts as a friendly companion that interacts with users, reducing loneliness and improving mental well-being.
- **Cost-Effective Home Healthcare Solution**: Lowers healthcare costs by reducing hospital visits, preventing emergencies, and enabling efficient remote care management.

# Channels:

- **Direct Sales to Healthcare Institutions**
  Supplying the health companion robot directly to hospitals, clinics, nursing homes, and assisted living facilities.
- **Online Sales Platforms**
  Company website and e-commerce platforms for individual customers and families to purchase or subscribe to the system.
- **Partnerships with Telemedicine Providers**
  Integration and distribution through telehealth companies offering remote patient monitoring services.
- **Medical Device Distributors**
  Leveraging existing healthcare equipment distributors to reach a wider market efficiently.
- **Insurance and Healthcare Programs**
  Distribution through insurance providers or government health programs as part of preventive care packages.
- **Exhibitions, Conferences, and Demonstrations**
  Showcasing the system at healthcare technology expos, medical conferences, and community health events to attract institutional buyers.

## Customer Relationships:

- **Personalized Onboarding and Setup**
  Guided installation, calibration, and user training to ensure correct use of the robot and monitoring system.
- **Ongoing Technical Support**
  24/7 customer support via phone, chat, or app to resolve technical issues and ensure system reliability.
- **Long-Term Care Monitoring Relationship**
  Continuous engagement through regular health reports, alerts, and system updates for users and caregivers.

- **Automated Notifications and Reminders**
  Proactive communication such as medication reminders, health alerts, and maintenance notifications.
- **Subscription-Based Service Model**
  Regular interaction through subscription plans that include software updates, cloud services, and data analytics.
- **Feedback and Improvement Loop**
  Collecting user and caregiver feedback to improve features, personalize services, and enhance user satisfaction over time.

## Revenue Streams:

- **Device Sales**
  One-time purchase revenue from selling the health companion robot and embedded monitoring hardware.
- **Subscription Fees**
  Monthly or annual subscriptions for cloud services, data storage, analytics, and remote monitoring features.
- **Service and Maintenance Contracts**
  Fees for technical support, system maintenance, repairs, and hardware upgrades.
- **Licensing to Healthcare Providers**
  Licensing the monitoring software and AI algorithms to hospitals, clinics, and telemedicine companies.
- **Customization and Integration Services**
  Revenue from customizing the system for specific medical needs or integrating it with existing healthcare IT systems.
- **Partnership and Insurance Reimbursement Programs**
  Revenue through partnerships with insurance companies or government health programs that reimburse or subsidize the system.

## Key Resources:

- **Embedded Hardware Components**
  Sensors (heart rate, $SpO_2$, temperature, motion), microcontrollers, communication modules, and robotic actuators.
- **Software and AI Algorithms**
  Embedded firmware, data processing software, machine learning models for anomaly detection, and decision-making.
- **Robotic Platform and Design**
  Physical robot structure, mobility system, human–robot interaction interfaces (voice, display, gestures).
- **Cloud Infrastructure and Data Systems**
  Servers, databases, and analytics platforms for remote monitoring, data storage, and secure access.
- **Skilled Development Team**
  Engineers and specialists in embedded systems, robotics, AI, healthcare technology, and cybersecurity.
- **Medical and Regulatory Expertise**
  Clinical knowledge, certifications, and compliance resources to meet healthcare standards and regulations.

# Key Activities:

- **Design and Development of Embedded Systems**
  Developing and integrating sensors, microcontrollers, and firmware for accurate vital sign monitoring.
- **AI and Data Analytics Development**
  Creating algorithms for health data analysis, anomaly detection, and predictive alerts.
- **Robot Design and Integration**
  Designing the robotic platform, mobility, and human–robot interaction features.
- **System Testing and Validation**
  Performing functional testing, clinical validation, and reliability testing to ensure accuracy and safety.

- **Manufacturing and Assembly**
  Producing, assembling, and quality-checking hardware components and robotic units.
- **Deployment, Maintenance, and Updates**
  Installing systems, providing maintenance, software updates, and continuous performance improvements.

## Key Partenerships:

- **Healthcare Providers and Hospitals**
  For clinical validation, pilot deployments, and integration into healthcare workflows.
- **Medical Device and Sensor Manufacturers**
  To supply reliable, certified sensors and embedded hardware components.
- **AI and Software Technology Partners**
  Collaboration for advanced data analytics, machine learning models, and cloud services.
- **Telemedicine and Remote Care Companies**
  Integration with telehealth platforms for remote monitoring and virtual consultations.
- **Insurance Companies and Government Health Agencies**
  Partnerships for reimbursement models, subsidies, and large-scale healthcare programs.
- **Manufacturing and Robotics Firms**
  For robot production, assembly, scalability, and cost-efficient manufacturing.

## Cost Structure:

- **Hardware and Sensor Costs**
  Expenses for embedded sensors, microcontrollers, robotic actuators, and other physical components.
- **Software Development and Maintenance**
  Costs for AI algorithms, embedded firmware, mobile apps, cloud platforms, and regular software updates.
- **Manufacturing and Assembly**
  Production, assembly, quality control, and testing of the robotic system.

- **Research and Development (R&D)**
  Investment in new features, system improvements, clinical trials, and technology innovation.
- **Operational and Support Costs**
  Customer service, technical support, installation services, and ongoing maintenance.
- **Regulatory Compliance and Certification**
  Costs associated with medical certifications, safety compliance, and adherence to healthcare regulations.

# Références:

[1] United Nations, Department of Economic and Social Affairs, Population Division. *World Population Ageing 2022*. ST/ESA/SER.A/476. Available: https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/wpp2022_summary_of_results.pdf

[2] World Health Organization (WHO). *Noncommunicable Diseases Fact Sheet*. June 2022. Available: https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases

[3] Lygidakis, Charilaos, Clodagh McLoughlin, and Kunal Patel. "Achieving universal health coverage: Technology for innovative primary health care education." (2016).

[4] Bashshur, Rashid L., Joel D. Howell, Elizabeth A. Krupinski, Kathryn M. Harms, Noura Bashshur, and Charles R. Doarn. "The empirical foundations of telemedicine interventions in primary care." *Telemedicine and e-Health* 22, no. 5 (2016): 342-375.

[5] Jayakody, A., Bryant, J., Carey, M., Hobden, B., Dodd, N. and Sanson-Fisher, R., 2016. Effectiveness of interventions utilising telephone follow up in reducing hospital readmission within 30 days for individuals with chronic disease: a systematic review. *BMC health services research*, *16*(1), p.403.

[6] Inglis, Sally C., Robyn A. Clark, Riet Dierckx, David Prieto-Merino, and John GF Cleland. "Structured telephone support or non-invasive telemonitoring for patients with heart failure." *Heart* 103, no. 4 (2017): 255-257.

[7] Inglis, Sally C., Robyn A. Clark, Riet Dierckx, David Prieto-Merino, and John GF Cleland. "Structured telephone support or non-invasive telemonitoring for patients with heart failure." *Cochrane Database of Systematic Reviews* 10 (2015).

[8] Chaudhry, S. I., et al. “Telemonitoring for Patients with Chronic Heart Failure.” *New England Journal of Medicine*, vol. 363, no. 18, 2010, pp. 1702–1711.

https://doi.org/10.1056/NEJMoa1005993

→ *Primary source for real-world VA program impact and TEN-HMS limitations (trial stopped early)*

[9] Espressif Systems. *ESP32 Technical Reference Manual*, v4.9, 2024. https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf, *Authoritative source for ESP32 specs (clock speed, memory, peripherals)*

[10] Pereira, S. L., et al. "Accuracy of Wearable Devices for Estimating Heart Rate, Heart Rate Variability, and Blood Oxygen Saturation: Systematic Review of Validation Studies." *NPJ Digital Medicine*, vol. 3, no. 1, 2020, Article 129. https://doi.org/10.1038/s41746-020-0284-y

[11] Radha, M., et al. "A comparative study of motion artifact reduction algorithms on wrist-worn PPG signals during daily activities." *Physiological Measurement*, vol. 42, no. 8, 2021, 085006. https://doi.org/10.1088/1361-6579/ac1a5a

[12] Banbury, C. R., et al. "TinyML: Efficient Deep Learning on Microcontrollers." *Proceedings of Machine Learning Research (PMLR)*, vol. 166, 2022, pp. 1–30. https://proceedings.mlr.press/v166/banbury22a.html

[13] Almalki, M., et al. "Privacy, Security, and Ethical Challenges in the Integration of IoT and AI in Digital Health: A Systematic Review." *Journal of Medical Internet Research (JMIR) Medical Informatics*, vol. 10, no. 4, 2022, e35470. https://doi.org/10.2196/35470