



REAL TIME SYSTEM AND INTERNET OF THINGS FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

SaR-PaM (Self-Balancing Robot with Path Memorization)

GROUP 15

Bryan Herdianto	2306210885
Daffa Bagus Dhiananto	2306250756
Nabiel Harits Utomo	2306267044
Tjokorde Gde Agung Abel Putra	2206059736

PREFACE

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa karena dengan rahmat dan karunia-Nya, laporan proyek akhir berjudul “SaR-PaM (Self-Balancing Robot with Path Memorization)” ini dapat diselesaikan dengan baik. Proyek ini merupakan bagian dari rangkaian praktikum Real Time System and Internet of Things dan bertujuan untuk merancang sebuah robot self-balancing berbasis ESP32 yang mampu menjaga kestabilan menggunakan sensor IMU dan kendali PID, sekaligus dapat merekam serta mengulangi jalur pergerakan melalui mekanisme path memorization. Selain itu, sistem juga menyediakan dua jenis metode komunikasi, yaitu kontrol melalui WebSocket (wifi) untuk jangkauan lebih luas dan kontrol melalui Bluetooth untuk interaksi jarak dekat.

Penyelesaian proyek ini tidak terlepas dari dukungan dan bantuan berbagai pihak. Kami mengucapkan terima kasih kepada seluruh asisten laboratorium Digital FTUI yang telah memberikan arahan, bimbingan, serta evaluasi selama proses praktikum dan pengembangan proyek ini. Kami juga berterima kasih kepada rekan-rekan kelompok yang telah bekerja sama secara aktif dalam merancang, mengimplementasikan, menguji, dan mendokumentasikan sistem hingga proyek ini dapat terselesaikan.

Kami menyadari bahwa laporan ini masih memiliki kekurangan dan keterbatasan. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun demi peningkatan kualitas pada proyek maupun dokumentasi di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi bagi pengembangan teknologi di bidang robotika dan Internet of Things.

Depok, December 07, 2025

Group 15

TABLE OF CONTENTS

CHAPTER 1.....	4
INTRODUCTION.....	4
1.1 PROBLEM STATEMENT.....	4
1.3 ACCEPTANCE CRITERIA.....	5
1.4 ROLES AND RESPONSIBILITIES.....	5
1.5 TIMELINE AND MILESTONES.....	5
CHAPTER 2.....	7
IMPLEMENTATION.....	7
2.1 HARDWARE DESIGN AND SCHEMATIC.....	7
2.2 SOFTWARE DEVELOPMENT.....	7
2.3 HARDWARE AND SOFTWARE INTEGRATION.....	8
CHAPTER 3.....	9
TESTING AND EVALUATION.....	9
3.1 TESTING.....	9
3.2 RESULT.....	9
3.3 EVALUATION.....	10
CHAPTER 4.....	11
CONCLUSION.....	11

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Pengembangan robot self-balancing menghadapi beberapa tantangan yang berkaitan dengan kestabilan dan kemampuan navigasi. Robot dua roda secara alami tidak stabil dan membutuhkan sistem kendali yang cepat dan akurat agar dapat mempertahankan keseimbangannya. Tanpa pengolahan data sensor yang baik dan algoritma kontrol yang tepat, robot akan mudah jatuh, kehilangan arah, atau gagal merespons perubahan lingkungan.

Selain itu, kebanyakan robot self-balancing hanya mampu bergerak secara manual dan tidak memiliki kemampuan untuk mengingat atau mengulangi jalur yang telah dilalui. Hal ini membatasi fungsionalitas robot dalam skenario yang memerlukan navigasi berulang, misalnya demonstrasi edukasi, eksplorasi ruangan, atau aplikasi semi-otonom.

Di sisi lain, kebutuhan fleksibilitas kontrol juga menjadi perhatian. Bluetooth menyediakan kemudahan interaksi jarak dekat, tetapi memiliki keterbatasan jangkauan. Sebaliknya, wifi memberikan jangkauan yang lebih luas namun membutuhkan mekanisme komunikasi yang lebih stabil dan real-time. Kombinasi kedua metode komunikasi ini perlu dirancang agar robot dapat dikendalikan dalam berbagai kondisi operasional.

Dengan demikian, permasalahan yang ingin diselesaikan melalui proyek ini adalah bagaimana merancang sebuah robot self-balancing berbasis ESP32 yang:

- Mampu menjaga kestabilan menggunakan pembacaan sensor IMU dan kendali PID,
- Dapat merekam dan mengulangi jalur pergerakan,
- Mendukung kontrol jarak dekat melalui Bluetooth dan kontrol jarak jauh melalui WebSocket (wifi),
- Serta mampu beroperasi secara real-time melalui pengelolaan multitasking menggunakan FreeRTOS.

1.2 PROPOSED SOLUTION

Solusi yang diusulkan untuk menyelesaikan permasalahan yang telah diidentifikasi adalah merancang sebuah self-balancing robot berbasis ESP32 yang mengintegrasikan sensor IMU, kendali PID, mekanisme penyimpanan jalur, serta dua jenis metode komunikasi (wifi dan Bluetooth) di dalam satu sistem yang dikelola menggunakan FreeRTOS.

Robot ini menggunakan sensor IMU untuk membaca orientasi dan sudut kemiringan secara real-time. Data tersebut kemudian diproses oleh algoritma PID guna mengatur kecepatan dan arah putaran kedua motor sehingga robot dapat mempertahankan keseimbangannya. Untuk mendukung navigasi, robot dilengkapi kemampuan path memorization, yaitu mekanisme pencatatan pergerakan motor yang memungkinkan robot mengingat dan mengulangi jalur yang pernah dilalui.

Sistem komunikasi dirancang dengan dua pendekatan: Bluetooth sebagai kontrol jarak dekat yang responsif, serta WebSocket (wifi) untuk kontrol jarak jauh dengan jangkauan lebih luas. Kedua metode komunikasi ini berjalan bersamaan dalam arsitektur FreeRTOS, di mana masing-masing fungsi penting, pembacaan sensor, kontrol motor, komunikasi, dan path processing, dijalankan dalam task terpisah sehingga operasi robot tetap deterministik dan tidak saling mengganggu.

Dengan pendekatan tersebut, solusi ini tidak hanya memungkinkan robot untuk menjaga keseimbangan, tetapi juga memberikan fleksibilitas mode operasi dan kemampuan navigasi yang lebih cerdas, sekaligus memenuhi kebutuhan proyek untuk menerapkan konsep RTOS dan IoT secara terintegrasi.

1.3 ACCEPTANCE CRITERIA

Kriteria penerimaan dari proyek ini adalah sebagai berikut:

1. Sistem harus mampu menjaga keseimbangan robot secara stabil menggunakan sensor IMU dan kontrol PID.
2. Sistem harus mampu beroperasi dalam empat mode: mode stabil, mode penyimpanan jalur, mode kontrol WebSocket, dan mode kontrol Bluetooth.
3. Robot harus mampu merekam jalur pergerakan dan mengulang kembali jalur tersebut secara konsisten.

4. Sistem harus mampu menerima perintah kontrol melalui WebSocket menggunakan koneksi wifi dengan respons yang baik.
5. Sistem harus mampu menerima input kontrol secara real-time melalui Bluetooth dengan koneksi yang stabil.
6. Arsitektur berbasis FreeRTOS harus memastikan bahwa task pembacaan IMU, komputasi PID, dan pengendalian motor berjalan tepat waktu tanpa kegagalan.
7. Sistem harus memiliki input melalui sensor atau komunikasi, serta output melalui aktuator berupa motor driver.
8. Proyek harus menggunakan minimal enam modul praktikum sesuai dengan ketentuan yang diberikan.
9. Sistem harus dapat didemonstrasikan secara penuh menggunakan rangkaian fisik pada saat presentasi.

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Programmer	Membuat rangkaian elektrik, kerangka 3D, dan kode	Bryan Herdianto
Project Manager	Membuat laporan, PPT, dan dokumentasi proyek	Daffa Bagus Dhiananto
Project Manager	Membuat laporan, PPT, dan dokumentasi proyek	Nabiel Harits Utomo
Project Manager	Membuat dokumentasi	Tjokorde Gde Agung Abel Putra

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

No	Task / Milestone	Dec 4	Dec 5	Dec 6	Dec 7	Dec 8	Status
1	Hardware Design Completion						Completed
2	Software Development						Completed
3	Hardware & Software Integration						Completed
4	Final Assembly & Testing						Completed

Table 2. Timeline Gantt Chart

Milestone	Description	Status
Hardware Design Completion	Finalisasi desain rangka, pemasangan motor driver, IMU, ESP32, baterai, serta wiring seluruh komponen hardware.	Completed
Software Development	Implementasi FreeRTOS tasks, pembacaan IMU, kontrol PID, kontrol motor, path memorization, serta integrasi komunikasi WebSocket dan Bluetooth.	Completed
Hardware & Software Integration	Sinkronisasi pembacaan IMU, PID balancing, dan kontrol motor dengan input dari WebSocket/Bluetooth, serta pengujian integrasi awal seluruh sistem.	Completed
Final Assembly & Testing	Final tuning sistem balancing, debugging, pengujian replay jalur, uji	Completed

	operasional semua mode, dan penyusunan dokumentasi akhir.	
--	---	--

Table 3. Timeline and Milestones

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

Hardware design pada sistem SaR-PaM (Self-Balancing Robot with Path Memorization) mencakup perancangan keseluruhan rangkaian elektronik yang digunakan untuk menjaga kestabilan robot, mengontrol motor, membaca sensor, serta menyediakan koneksi nirkabel. Desain hardware berfokus pada integrasi sensor IMU, driver motor, mikrokontroler, modul komunikasi, serta sistem catu daya.

Robot ini menggunakan sensor IMU (Inertial Measurement Unit) sebagai sumber data utama untuk pengendalian keseimbangan. IMU mengukur percepatan dan kecepatan sudut, kemudian data ini diolah oleh mikrokontroler untuk menjaga posisi robot menggunakan algoritma PID. Dua motor DC dengan gearbox dihubungkan ke motor driver yang mampu memberikan arus cukup besar untuk menggerakkan robot secara stabil.

Selain itu, SaR-PaM dilengkapi modul wifi untuk komunikasi berbasis WebSocket dan modul Bluetooth/BLE untuk kontrol jarak dekat. Integrasi kedua modul ini memungkinkan robot memiliki dua metode kendali:

- Kontrol Bluetooth, untuk mode mengikuti pengguna (follow-me) atau kendali manual jarak dekat.
- Kontrol wifi, untuk kendali jarak jauh dan pengiriman data secara real-time melalui WebSocket.

Sistem catu daya menggunakan baterai Li-ion/Li-Po yang terhubung ke modul power management guna menjaga stabilitas suplai tegangan ke motor dan mikrokontroler.

2.2 SOFTWARE DEVELOPMENT

Software untuk sistem SaR-PaM dikembangkan menggunakan Arduino IDE dengan dukungan library ESP32. Arsitektur perangkat lunak disusun menggunakan FreeRTOS, yang telah terintegrasi pada ESP32, sehingga setiap proses utama dapat dijalankan secara paralel. Pendekatan multitasking ini memastikan robot tetap responsif saat melakukan balancing, menerima perintah pengguna, dan menyimpan jalur pergerakan secara bersamaan.

Terdapat beberapa modul utama dalam pengembangan software:

- **IMU Processing & PID Balancing Module**

Modul ini bertanggung jawab untuk:

- Membaca data gyro dan accelerometer dari IMU.
- Melakukan filter (Complementary / Kalman).
- Menghitung sudut kemiringan.
- Menjalankan algoritma PID untuk menjaga keseimbangan robot.

Proses ini berjalan dalam task real-time dengan interval tetap menggunakan software timer, sehingga balancing tidak terganggu walaupun modul lain sedang aktif.

- **Path Memorization Module**

Modul ini mencatat:

- Kecepatan motor
- Arah rotasi
- Waktu tempuh

Data tersebut disimpan dalam buffer atau queue sehingga robot dapat mengulangi jalur yang pernah ditempuh (path replaying mode). Modul ini berjalan paralel dengan motor control tanpa mempengaruhi stabilitas robot.

- **WebSocket Control Module (wifi)**

Modul ini mengelola koneksi wifi untuk remote-control jarak jauh. Fungsinya meliputi:

- Menerima instruksi dari user melalui WebSocket.
- Mengirim telemetry (sudut kemiringan, output PID, kecepatan motor, status mode).

Karena jangkauan wifi lebih luas dibandingkan Bluetooth, mode ini cocok untuk kendali jarak jauh.

- **Bluetooth Control Module**

Digunakan untuk kontrol lokal atau mode follow-me. Dalam modul ini yaitu:

- Menerima instruksi arah (forward, backward, turn).
- Mengirim status singkat seperti battery level atau mode aktif.

Bluetooth dipilih karena latensi rendah dan tidak bergantung pada jaringan wifi.

- **Motor Control Module**

Modul ini menghasilkan sinyal PWM untuk mengatur:

- Kecepatan motor kiri dan kanan
- Arah rotasi

PWM dijalankan melalui software timer sehingga nilai output selalu stabil meskipun beban multitasking tinggi.

- **Telemetry System**

Sistem telemetri membaca dan mengirimkan informasi seperti:

- Sudut IMU
- Output PID
- Kecepatan motor
- Level baterai
- Tegangan sistem

Data tersebut dapat dikirim melalui WebSocket atau Bluetooth tergantung mode.

- **Software Timer Implementation**

Software timer digunakan untuk menjaga:

- Frekuensi update PID
- Interval pengiriman telemetry
- Keseragaman PWM motor

Dengan timer, sistem tidak bergantung pada loop utama yang sering berubah-ubah durasinya karena multitasking.

- **Dual Control Input Support**

Dengan implementasi arsitektur di atas, software mendukung dua metode kendali utama untuk Bluetooth Control:

- Kontrol lokal, latensi rendah
- Cocok untuk mode follow-me
- Tidak memerlukan jaringan wifi

Untuk wifi (WebSocket) Control:

- Kontrol jarak jauh
- Mengirim data telemetry secara real-time
- Lebih stabil untuk monitoring dan otomasi

Kedua modul dijalankan secara independen sehingga pengguna dapat memilih mode sesuai skenario penggunaan.

2.3 HARDWARE AND SOFTWARE INTEGRATION

Integrasi hardware dan software pada SaR-PaM dilakukan melalui sinkronisasi antara pembacaan sensor, pengendalian motor, dan komunikasi nirkabel secara simultan. Data dari IMU dibaca oleh task sensor, lalu dikirim melalui queue ke task PID untuk menjaga keseimbangan. Task motor kemudian menerima output PID untuk menentukan kecepatan dan arah motor.

Saat robot berjalan, data posisi dan kecepatan direkam untuk fitur path memorization, sehingga robot mampu mengulang jalur yang pernah dilewati. Pada mode kontrol jarak jauh, komunikasi WebSocket digunakan untuk mengirim perintah dan menerima data secara real-time tanpa mengganggu stabilitas robot.

Dengan desain hardware yang terintegrasi dan sistem software multitasking yang terstruktur, SaR-PaM dapat menjalankan empat mode operasinya secara efektif:

- Mode Stabil, menjaga robot tetap tegak.
- Mode Mengingat Jalur, menyimpan dan mengulang lintasan.
- Kontrol WebSocket, kendali jarak jauh berbasis wifi.
- Kontrol Bluetooth, kendali manual jarak dekat.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

Pengujian dilakukan untuk memastikan bahwa seluruh fungsi sistem SaR-PaM berjalan sesuai dengan spesifikasi, terutama pada kemampuan robot untuk bergerak stabil, menerima perintah melalui wifi maupun Bluetooth, serta sistem switching mode dan manajemen energi. Beberapa pengujian yang dilakukan meliputi:

- Pengujian Stabilitas Gerak Robot Dua Roda
 - Mengamati performa balancing saat robot bergerak maju, mundur, dan berbelok.
 - Menguji respon terhadap perubahan kecepatan mendadak.
 - Menilai kemampuan robot mempertahankan sudut keseimbangan dengan gangguan eksternal kecil.
 - Pengujian Kontrol wifi
 - Mengirimkan perintah gerak (forward, backward, turn left/right) melalui WebSocket.
 - Mengukur delay respon kontrol.
 - Menguji konsistensi perintah dalam jangkauan wifi berbeda.
 - Pengujian Kontrol Bluetooth
 - Kendali lokal menggunakan aplikasi smartphone.
 - Menguji delay dan stabilitas koneksi.
 - Mengamati apakah balancing tetap berjalan selama kontrol Bluetooth aktif.
 - Pengujian Mode Switching
- Pengujian dilakukan pada dua skenario:
- Switch dari wifi ke Bluetooth, wifi dimatikan otomatis untuk menghemat energi.

- Switch dari Bluetooth ke wifi, bluetooth dinonaktifkan saat wifi aktif agar konsumsi daya lebih efisien.
- Pengujian Power Management
 - Memantau konsumsi daya saat wifi aktif, Bluetooth aktif, dan saat mode switching.
 - Mencatat voltage drop saat robot bergerak pada beban tinggi.

Semua pengujian dilakukan dalam berbagai kondisi lingkungan dan jarak kontrol untuk mendapatkan hasil yang representatif.

3.2 RESULT

Berikut adalah result dari pengujian projek kami.

No.	Pengujian	Parameter	Status
1	Stabilitas robot roda dua	Kestabilan PID dan balancing tilt agar robot tidak jatuh	PASS
2	Kontrol Wifi	Delay respon yang cepat dan akurasi perintah yang sesuai	PASS
3	Kontrol Bluetooth	Latensi dan kestabilan koneksi yang tidak mudah terputus	PASS
4	Mode switching	Keberhasilan berpindah mode Wifi ke Bluetooth tanpa gangguan	PASS
5	Power management	Pengurangan konsumsi daya setelah modul dimatikan	PASS
6	Path memorization	Jalur yang di save dapat dijalankan kembali oleh robot	PASS

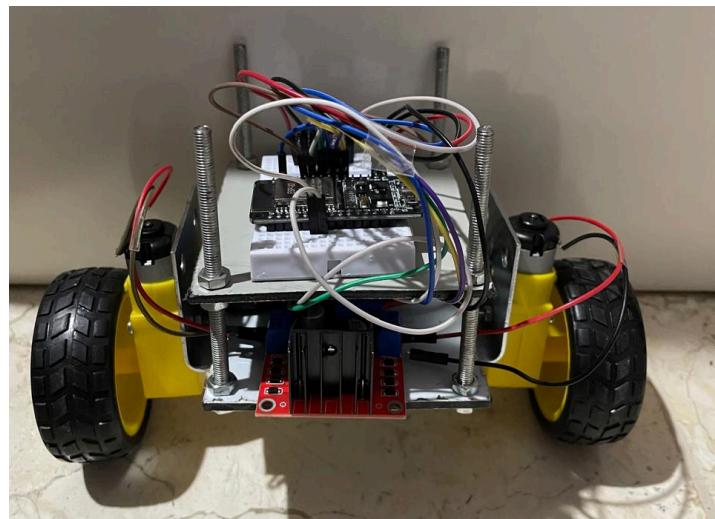


Fig 2. Testing Result

Gambar menunjukkan proses pengujian SaR-PaM pada berbagai mode kontrol. Dalam foto tersebut terlihat robot sedang diuji untuk bergerak maju dan berbelok sambil menjaga stabilitas menggunakan balancing system. Pengujian juga melibatkan verifikasi apakah robot tetap stabil saat perintah dikirim melalui wifi maupun Bluetooth. Gambar ini mendokumentasikan kondisi robot saat menjalankan skenario uji, termasuk stabilitas gerak, respon terhadap perintah, dan performa mode switching.

3.3 EVALUATION

Berdasarkan pengujian yang telah dilakukan, beberapa evaluasi dapat disimpulkan sebagai berikut:

- **Robot stabil meskipun menggunakan dua roda**

Sistem balancing berbasis IMU dan PID mampu menjaga robot tetap tegak saat bergerak. Robot dapat mempertahankan stabilitas meskipun terdapat perubahan kecepatan maupun manuver belok.

- **Robot dapat dikontrol melalui wifi**

Kontrol melalui WebSocket berjalan dengan baik. Delay relatif kecil dan robot merespons perintah secara real-time. Mode ini efektif untuk kontrol jarak jauh karena jangkauan wifi lebih luas.

- **Robot dapat dikontrol melalui Bluetooth**

Bluetooth memberikan latensi yang lebih rendah sehingga kontrol lokal terasa lebih responsif. Robot tetap stabil ketika menerima perintah melalui Bluetooth.

- **Sistem dapat melakukan switching mode Wifi dan Bluetooth**

Mode dapat diganti tanpa mengganggu proses balancing.

- Ketika wifi aktif, Bluetooth berhasil dinonaktifkan untuk menghemat energi.
- Ketika Bluetooth diaktifkan, wifi dinonaktifkan sehingga konsumsi daya berkurang dan interferensi berkurang.

Fitur switching ini mendukung penggunaan robot dalam berbagai kondisi operasional.

CHAPTER 4

CONCLUSION

Pada penelitian dan pengembangan sistem kontrol robot berbasis ESP32 ini, telah berhasil dibangun sebuah platform kecil yang dapat dikendalikan melalui dua mode komunikasi, yaitu WiFi (MQTT) dan Bluetooth. Sistem dirancang dengan mempertimbangkan efisiensi energi, stabilitas gerak, serta kemampuan switching antar mode kontrol secara dinamis. Berdasarkan rangkaian pengujian yang telah dilakukan, dapat disimpulkan beberapa poin utama berikut:

- **Robot dapat bergerak stabil meskipun menggunakan konfigurasi dua roda.**
Implementasi pengendalian motor menggunakan PWM dan tuning kecepatan yang tepat memungkinkan robot mempertahankan arah gerak yang konsisten baik saat maju, mundur, maupun ketika melakukan manuver belok.
- **Sistem berhasil mendukung kontrol jarak jauh melalui koneksi WiFi berbasis MQTT.**
Robot mampu menerima perintah dari frontend secara real-time dan mengirimkan data telemetry seperti kecepatan, sudut kemudi, dan status baterai secara kontinu dengan latensi rendah.
- **Kontrol melalui Bluetooth dapat berfungsi dengan baik sebagai mode lokal.**
Menggunakan aplikasi pada smartphone, pengguna dapat mengendalikan robot secara langsung dengan respons yang cepat dan stabil.
- **Fitur hybrid control mode berhasil diimplementasikan.**
Sistem dapat melakukan switching antara WiFi dan Bluetooth secara otomatis maupun manual. Ketika salah satu mode tidak digunakan, modul komunikasi tersebut dapat dinonaktifkan untuk menghemat konsumsi daya.

Secara keseluruhan, sistem yang dibangun telah memenuhi seluruh acceptance criteria yang ditetapkan. Implementasi multitasking menggunakan FreeRTOS juga terbukti mampu

menjaga responsivitas robot meskipun beberapa modul berjalan secara paralel. Dengan demikian, proyek ini dapat dinyatakan berhasil dan dapat dikembangkan lebih lanjut, misalnya dengan menambahkan obstacle detection, autonomous navigation, atau peningkatan stabilitas mekanik.

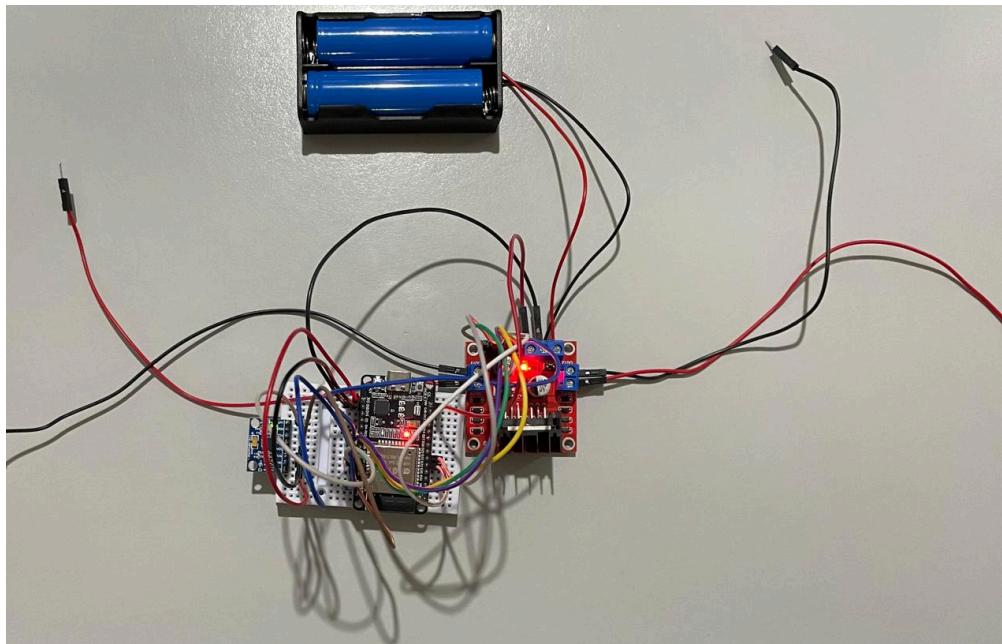
REFERENCES

- [1] "Module 1 - Introduction to SMP with RTOS," Digilab DTE. [Online]. Available: <https://learn.digilabdte.com/books/internet-of-things/chapter/module-1-introduction-to-smp-with-rtos> (Accessed: Dec. 7, 2025).
- [2] "Module 2 - Task Management," Digilab DTE. [Online]. Available: <https://learn.digilabdte.com/books/internet-of-things/chapter/module-2-task-management> (Accessed: Dec. 7, 2025).
- [3] "Module 3 - Memory Management & Queue," Digilab DTE. [Online]. Available: <https://learn.digilabdte.com/books/internet-of-things/chapter/module-3-memory-management-queue> (Accessed: Dec. 7, 2025).
- [4] "Module 4 - Deadlock & Synchronization," Digilab DTE. [Online]. Available: <https://learn.digilabdte.com/books/internet-of-things/chapter/module-4-deadlock-syncronization> (Accessed: Dec. 7, 2025).
- [5] "Module 5 - Software Timer," Digilab DTE. [Online]. Available: <https://learn.digilabdte.com/books/internet-of-things/chapter/module-5-software-timer> (Accessed: Dec. 7, 2025).
- [6] "Module 6 - Bluetooth & BLE," Digilab DTE. [Online]. Available: <https://learn.digilabdte.com/books/internet-of-things/chapter/module-6-bluetooth-ble> (Accessed: Dec. 7, 2025).
- [7] "Module 7 - MQTT, HTTP, WIFI," Digilab DTE. [Online]. Available: <https://learn.digilabdte.com/books/internet-of-things/chapter/module-7-mqtt-http-wifi> (Accessed: Dec. 7, 2025).
- [8] "Module 8 - Power Management," Digilab DTE. [Online]. Available: <https://learn.digilabdte.com/books/internet-of-things/chapter/module-8-power-management> (Accessed: Dec. 7, 2025).
- [9] Instructables, "DIY ESP32 Wifi Self Balancing Robot - B-Robot ESP32 Arduino Programing," Instructables, Sep. 19, 2021. [Online]. Available: <https://www.instructables.com/DIY-ESP32-Wifi-Self-Balancing-Robot-B-Robot-ESP32/> (Accessed: Dec. 7, 2025).
- [10] "Bluetooth® Low Energy (Bluetooth LE)," espressif. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/bluetooth/bt_le.html (Accessed: Dec. 7, 2025).

[11] "Bluetooth® Architecture," espressif. [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/bt-architecture/index.html> (Accessed: Dec. 7, 2025).

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation



