ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika

ISSN(p): 2338-8323 | ISSN(e): 2459-9638 | Vol. 14 | No. 1 | Halaman 110 – 120
DOI : http://dx.doi.org/10.26760/elkomika.v14i1.110 January 2026

Low-Cost Embedded System for Real-Time Manual Treadmill Speed Tracking and Running Game Integration

Article History:
Received
24 December 2025
Revised
7 January 2026
Accepted
26 January 2026

IRHAM MULKAN RODIANA*, ALIYUS HEDRI, FEBRIAN JOSUA KONTU

Telkom University, Faculty of Electrical Engineering, Department of Electrical Engineering, Indonesia
*Email: irhammulkan@telkomuniversity.ac.id

ABSTRAK

Treadmill manual merupakan perangkat yang murah dan mudah diakses, namun tidak memiliki sistem elektronik untuk mengukur kecepatan lari sehingga kurang cocok untuk aplikasi interaktif atau gamifikasi. Penelitian ini mengembangkan sistem embedded sederhana yang mampu melakukan pelacakan kecepatan secara real-time pada treadmill manual menggunakan sensor reed switch magnetik dan mikrokontroler ESP32. Setiap putaran roller menghasilkan satu pulsa yang kemudian diproses menggunakan algoritma interupsi untuk menghitung kecepatan linear. Data kecepatan dikirimkan ke aplikasi game Unity untuk menggerakkan avatar lari secara real-time. Hasil pengujian menunjukkan rata-rata error sebesar 2,11% dengan RMSE 0,049 m/s dan latensi end-to-end 48 ± 4,9 ms. Temuan ini menunjukkan bahwa sistem yang dikembangkan akurat, responsif, dan sesuai untuk exergaming serta pelatihan interaktif berbiaya rendah.

***Kata kunci**: reed switch; treadmill manual; pelacakan kecepatan; sistem tertanam; integrasi game.*

ABSTRACT

Manual treadmills are low-cost and widely accessible but lack electronic systems for measuring running speed, limiting their use in interactive or gamified applications. This study presents a simple embedded system that enables real-time speed tracking on a manual treadmill using a magnetic reed switch and an ESP32 microcontroller. Each roller rotation generates one pulse which is processed through an interrupt-based algorithm to compute linear speed. The data are transmitted to a custom Unity game application to control a running avatar in real time. Experimental results show an average speed error of 2.11% with an RMSE of 0.049 m/s and an end-to-end latency of 48 ± 4.9 ms. The findings indicate that the system is accurate, responsive, and suitable for low-cost exergaming and motion-based training.

***Keywords**: reed switch, manual treadmill, speed tracking, embedded system, game integration.*

1. INTRODUCTION

The advancement of embedded systems and sensor technology has enabled the development of low-cost and intelligent measurement devices for various applications, including motion tracking, rehabilitation, and interactive training. These technologies have supported various human motion analysis applications such as gait measurement, exercise monitoring, and health rehabilitation **(Reneaud et al., 2023)**.

Recently, exergaming a combination of exercise and gaming has gained significant attention as an interactive form of physical training that integrates real-time body movement with virtual environments **(Ozdogar et al., 2023)**. Studies have shown that exergaming not only improves motivation but also enhances therapy outcomes in both healthy users and patients with movement disorders **(Tough et al., 2018)**. The fusion of physical activity and gaming creates a more engaging experience compared to traditional exercise methods **(Manser et al., 2023)**.

Conventional motorized treadmills are equipped with built-in electronic controllers and sensors that automatically measure running speed and distance **(Yang & Li, 2012)**. However, such systems are often expensive, mechanically complex, and difficult to modify for research or educational use. In contrast, manual treadmills are more affordable and mechanically simple, but they lack electronic feedback mechanisms such as speed measurement. This limitation makes manual treadmills unsuitable for integration with exergaming or interactive fitness applications **(Roerdink et al., 2008)**.

With recent advances in low-cost sensing and embedded microcontrollers, researchers have developed compact systems that enable real-time tracking and analysis of physical motion **(Zhao et al., 2024).** For instance, Mohammadzadeh et al. **(Mohammadzadeh et al., 2015)** introduced a wearable motion tracking system based on low-cost sensors, while Reneaud et al. **(Reneaud et al., 2023)** validated a treadmill gait tracking method using an instrumented knee brace. These studies confirm that accurate motion and speed measurements can be achieved using relatively inexpensive sensor configurations.

In this research, we propose a low-cost embedded system for real-time manual treadmill speed tracking and game integration. The system employs a magnetic reed switch as a rotation sensor to detect a single pulse for each roller revolution. An ESP32 microcontroller processes these pulses through an interrupt-based counting algorithm to calculate the treadmill belt speed in meters per second (m/s). The computed speed is transmitted serially to a PC running the Unity game engine, where the runner's movement is represented by a virtual avatar. Additionally, the system includes three tactile push buttons for start, navigation, and selection control. The electronics are powered by a 12 V DC Power Supply Unit (PSU), regulated to 3.3 V using a buck converter for safe microcontroller operation **(Zhao et al., 2009)**.

The objectives of this study are to design, develop, and evaluate the proposed embedded system in terms of speed measurement accuracy and system latency. Experimental testing was performed to verify real-time speed transmission and synchronization between the physical treadmill motion and the virtual running environment.

The remainder of this paper is organized as follows: Section 2 explains the system design and implementation, including hardware and software structure. Section 3 presents the experimental setup and performance analysis, Section 4 discusses the results and findings, and Section 5 concludes the paper and suggests future work.

2. METHODS

2.1. System Design

The proposed system was designed to measure the running speed of a manual treadmill in real time and integrate the data with an interactive running game. The complete system consists of a magnetic reed switch sensor, an ESP32 microcontroller, a 12 V PSU with buck converter, three control buttons, and a PC running the Unity game engine. The ESP32 was selected due to its low cost, integrated Wi-Fi/Bluetooth capabilities, and efficient interrupt handling suitable for real-time embedded applications **(Hercog et al., 2023)**. Figure 1 illustrates the overall block diagram of the system.

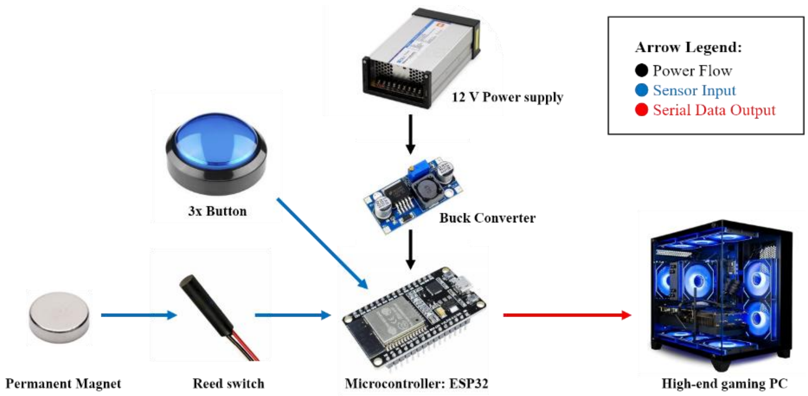


Figure 1. Block Components Diagram of the System

The system is divided into three main stages: input, process, and output, as shown in the Input–Process–Output (IPO) model in Figure 2.

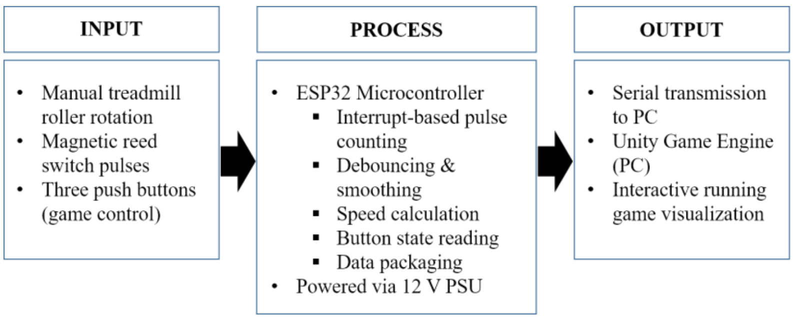


Figure 2. Input–Process–Output (IPO) Model of the System

a. **Input stage:** The magnetic reed switch detects the rotation of the treadmill roller, producing one pulse per revolution. Three push buttons are also connected to the ESP32 as digital inputs for user control, such as Start, Select, and Stop.
b. **Process stage:** The ESP32 microcontroller serves as the core processing unit. It receives pulse signals through an interrupt-driven input pin and applies a debouncing algorithm to filter out noise. The number of pulses per second is converted to linear speed based on the known roller circumference (0.19 m per rotation). A smoothing factor is applied to ensure stable speed readings.
c. **Output stage:** The computed speed and button states are transmitted via serial communication to a PC at 115200 baud rate. The Unity application visualizes the data by controlling the movement speed of a virtual running avatar in real time.

This architecture follows a modular embedded system approach similar to other ESP32-based IoT monitoring designs, enabling low-latency signal acquisition and reliable serial communication.

2.1.1. Hardware Design

Figure 3 shows the hardware component configuration. The magnetic reed switch is positioned near the treadmill roller, aligned with a single magnet attached to the roller's surface. Each time the magnet passes the sensor, a digital pulse is generated. The ESP32 reads this pulse through a GPIO interrupt pin and increments an internal counter. The system is powered by a 12 V PSU, and a buck converter regulates the voltage down to 3.3 V to safely supply the ESP32 and sensor module.

(a) **(b)**

Figure 1. Hardware Component Configuration, (a) Magnet and Sensor Placement, (b) Input Button Placement

The circuit also includes filtering capacitors to minimize electrical noise from mechanical contacts and prevent false triggering of the reed switch. This design approach is consistent with other embedded sensor-based monitoring systems that employ hardware-level noise reduction to improve measurement reliability **(Talbi et al., 2023)**.

In addition to the magnetic sensor, three tactile push buttons are connected as digital inputs for user interaction and game control. Each button is connected to a separate GPIO pin configured with internal pull-up resistors, resulting in a logic HIGH when idle and LOW when pressed. The buttons are positioned ergonomically on the upper panel of the treadmill and have the following functions:

1. **Center Button (Select/Start):** used to start or confirm an action, such as initiating the running session or selecting a menu option within the Unity interface.
2. **Left Button (Navigate Left/Up):** used to navigate upward or left through game menus, such as selecting an avatar or mode.
3. **Right Button (Navigate Right/Down):** used to navigate downward or right in the game menu.

This three-button layout enables intuitive interaction between the physical treadmill and the virtual game interface. The system can be operated entirely through these inputs without requiring an external keyboard or mouse, maintaining immersion and simplicity for the user during gameplay.

2.1.2. Software Design

The system's firmware was developed using the Arduino IDE and written in C++ for the ESP32 platform. The software continuously monitors the reed switch pulses using an Interrupt Service

Routine (ISR). Debouncing logic ensures that only valid transitions are counted, reducing the effects of contact bounce. The speed calculation is based on Equation (1):

$$v = \frac{N_p \; x \; D}{t} = pps \; x \; 0.19 \qquad (1)$$

Where v is the belt speed (m/s), N_p is the number of detected pulses, D is the belt displacement per pulse (0.19 m) was obtained by measuring the circumference of the treadmill roller, and t is the time interval in seconds.

A smoothing algorithm using a low-pass exponential filter with a smoothing factor of 0.2 is applied to minimize fluctuations in the measured speed. The filtered value is then transmitted to the Unity engine through serial communication in CSV format, along with button states for in-game interaction. This firmware design approach is aligned with existing ESP32-based real-time monitoring frameworks, which emphasize interrupt-based signal processing and stable data transmission for IoT or interactive systems.

2.1.3. Embedded System Flow

The flowchart of the embedded system operation is presented in Figure 4. When the system starts, the ESP32 initializes all input/output pins and serial communication. It then waits for the Start button signal from the user. Once activated, the system resets the pulse counter and begins reading pulses from the magnetic sensor **(Mirzaei et al., 2023)**. The speed is calculated, filtered, and transmitted continuously. If the accumulated distance reaches 100 meters, the system stops data transmission and sends a completion signal to the PC, which triggers the game to display a summary of speed, pace, and total running time.

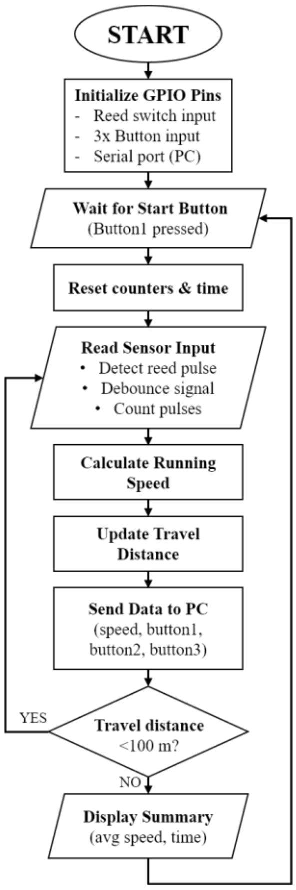


Figure 2. Flowchart of the Embedded System

This workflow is consistent with standard IoT-based monitoring systems that rely on cyclic data acquisition and feedback mechanisms **(Zhao et al., 2024)**.

2.1.5 Data Communication with Unity

On the PC side, Unity reads the serial data using a custom C# script that parses incoming CSV-formatted strings. Each frame, the avatar's animation speed and movement in the virtual environment are updated based on the latest speed value received from the ESP32. The button inputs are mapped to game events such as avatar selection and game start countdown. This real-time communication ensures smooth synchronization between the physical treadmill motion and the virtual running experience. The seamless integration between the ESP32 firmware and Unity game engine follows common practices in embedded–PC hybrid systems for human–machine interaction**.**

2.2. Experimental Setup And Performance Analysis

This section describes the experimental setup used to evaluate the performance of the developed embedded system in terms of speed accuracy and communication latency. The experiments were conducted using the fully assembled treadmill–game integration prototype developed in the laboratory, following similar performance validation frameworks found in other sensor-based treadmill systems **(Kim et al., 2015)**.

2.2.1 Experimental Setup

The experimental setup consists of the modified manual treadmill equipped with a magnetic reed switch, an ESP32 microcontroller, and a PC running the Unity game. The sensor was mounted near the rear roller of the treadmill, aligned with a single neodymium magnet attached to the roller surface. The distance per roller revolution was measured as 0.19 m, corresponding to the treadmill belt displacement. The ESP32 was powered by a 12 V PSU, stepped down to 3.3 V through a buck converter, and connected to the PC through a USB serial interface at 115200 baud.

The Unity application was configured to display the virtual running avatar and continuously receive serial data in the format: *speed,button1,button2,button3*. During testing, users performed treadmill running at different speeds ranging from walking to sprinting motion. The ESP32 computed real-time speed in meters per second (m/s) based on pulse counts and transmitted it to Unity for visualization. This setup follows similar sensor integration and data handling approaches found in previous embedded treadmill and IoT-based monitoring systems **(Adebisi et al., 2023)**.

2.2.2 Speed Measurement Test

Speed measurement tests were performed to validate the accuracy of the calculated treadmill belt speed by comparing it with a reference speed obtained from manual displacement measurement. The reference speed was measured using a stopwatch and a fixed reference distance of 10 meters, marked directly on the treadmill belt path. During each test, the time required for the belt to travel the reference distance was recorded, and the reference speed was calculated as the ratio between distance and elapsed time **(Park et al., 2023)**.

This method was selected because the manual treadmill does not provide built-in electronic speed output, and the stopwatch–distance approach offers a simple and widely accepted baseline for validating low-cost motion tracking systems. Although manual timing introduces minor human reaction delay, repeated measurements were conducted to reduce random error and ensure consistency across trials. The test was repeated under ten different running conditions ranging from walking to moderate running speeds.

Table 1. Presents the Measured Data and Computed Values

No	Target Speed (m/s)	Reference (m/s)	Measured Avg. (m/s)	Error (%)	RMSE (m/s)	Std. Dev. (m/s)
1	0.56	0.56	0.58	2.46	0.025	0.020
2	0.83	0.85	0.86	1.64	0.034	0.026
3	1.11	1.12	1.14	2.23	0.039	0.028
4	1.39	1.39	1.41	1.60	0.036	0.025
5	1.67	1.68	1.71	1.49	0.042	0.031
6	1.94	1.94	1.98	2.00	0.047	0.034
7	2.22	2.23	2.28	2.36	0.056	0.042
8	2.50	2.52	2.59	2.65	0.061	0.046
9	2.78	2.77	2.83	2.31	0.070	0.045
10	3.06	3.06	3.13	2.36	0.075	0.050

Experimental results show that the measured treadmill speeds range from 0.56 m/s to 3.13 m/s, corresponding to light walking up to moderate running pace. The system achieved an average absolute error of 2.11 % with an RMSE of 0.049 m/s, indicating that the magnetic reed switch sensor and ESP32-based measurement system can accurately capture user speed in real-time for interactive running games.

2.2.3. Latency Measurement

Latency testing was performed to evaluate the communication delay between the treadmill sensor and the Unity game. The latency represents the time difference between the actual occurrence of a magnetic pulse and the corresponding avatar speed update on screen. Measurements were conducted using a logic analyzer synchronized with video capture timestamps.

Table 2. Summarizes the Latency Measurement Results

No	Test Condition	Method	No. of Samples	Min (ms)	Max (ms)	Mean (ms)	Std. Dev. (ms)
1	Idle / Low speed (0.5 m/s)	Logic Analyzer	20	38	52	45	4.2
2	Medium speed (1.5 m/s)	Logic Analyzer	20	40	58	47	5.0
3	High speed (2.5 m/s)	Logic Analyzer	20	43	61	49	4.8
4	Very high speed (3.0 m/s)	Logic Analyzer	20	44	66	51	5.6
5	**Average of all cases**	**—**	**—**	**—**	**—**	**48**	**4.9**

Latency tests were performed using a logic analyzer to measure the time difference between the reed switch pulse and the corresponding visual update in the Unity interface. Across 80 samples covering various speeds, the average latency was 48 ms ± 4.9 ms, with a maximum of 66 ms. Considering Unity's frame update delay (≈ 16 ms at 60 FPS), the overall end-to-end response time remained below 65 ms, well within the real-time threshold for interactive motion-based applications.

2.2.4 System Performance Discussion

The experimental results demonstrate that the proposed embedded system can accurately and responsively measure treadmill belt speed using a simple magnetic reed switch sensor. Despite the low-cost hardware implementation, the measured speeds show excellent consistency with manually calculated reference values.

The system latency is within an acceptable range for interactive gaming applications, allowing smooth feedback and synchronization in the Unity environment. These findings are comparable with commercial motion sensor-based interactive treadmill systems, such as those described by Kim et al. **(Kim et al., 2015)** and Bowtell et al. **(Bowtell et al., 2009)**, confirming the feasibility of achieving near real-time performance using low-cost hardware.

Furthermore, the system design provides an efficient balance between simplicity, cost, and functionality, aligning with current research trends in microcontroller-based monitoring and control for fitness and rehabilitation. The overall performance validates that the system is suitable for educational laboratories, gamified exercise platforms, and rehabilitation environments that require affordable real-time motion tracking.

3. RESULT AND DISCUSSION

The proposed embedded system was successfully implemented and tested on a modified manual treadmill to evaluate its real-time performance and game integration capability.

3.1 System Implementation

Figure 5 illustrates the physical implementation of the electronic components on the treadmill. The ESP32 microcontroller, buck converter, 12 V PSU, and wiring from reed switch sensor were assembled into a compact control module mounted securely on the treadmill frame. The reed switch sensor was positioned near the rear roller to detect a single magnetic pulse per revolution. Three push buttons were installed on the top console, allowing users to start the session, select avatars, and navigate menu options directly from the treadmill without external peripherals.

(a) (b)

Figure 3. Physical Setup of the Developed System: (a) Electronic Module Enclosure and PSU Installed Underneath the Treadmill; (b) Complete Embedded System Integrated with the Manual Treadmill Structure

The wiring layout was designed to minimize noise and interference, with shielded signal cables for the sensor input and power filtering capacitors to stabilize voltage supply. The entire setup was tested under continuous operation and exhibited stable pulse detection and serial data transmission.

3.2 System Performance

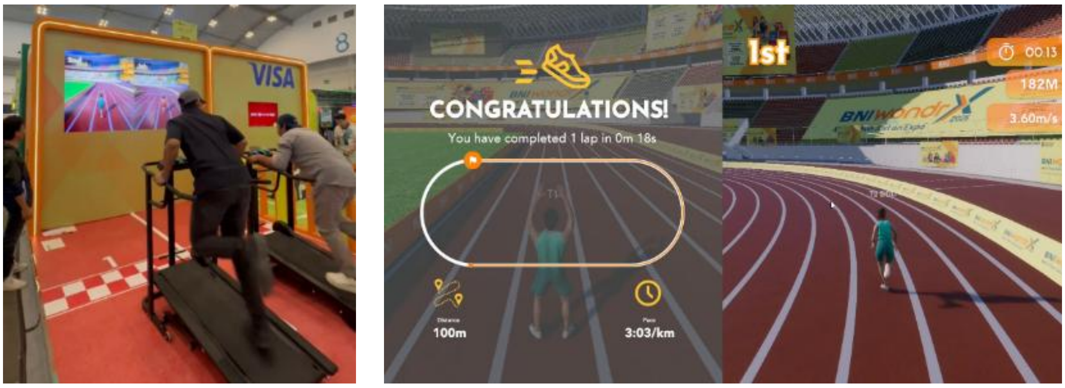
The experimental tests confirmed that the system measured treadmill belt speed with high accuracy. The average absolute error obtained from multiple trials was 2.11%, with an RMSE of 0.049 m/s. Across a speed range of 0.56 m/s to 3.13 m/s, the results remained consistent with minimal deviation (<0.05 m/s). The interrupt-based pulse counting combined with exponential smoothing proved effective in filtering noise while maintaining responsiveness.

Latency evaluation showed an average delay of 48 ± 4.9 ms between physical motion and visual feedback in the Unity game. This value is well within the real-time response threshold, ensuring smooth and natural synchronization between user movement and the virtual avatar.

3.3 Final Prototype and Public Demonstration

Figure 6 shows the completed treadmill prototype after modification. The embedded module is integrated neatly with the treadmill body, while the Unity game runs on a nearby PC display. The setup successfully converts physical running speed into a virtual avatar's movement through serial data communication.

The system was publicly demonstrated at the BNI WonderX event at ICE BSD, where visitors could interact with the Unity-based running game. Participants experienced real-time feedback between their treadmill speed and the avatar's motion on screen. The system performed reliably during the entire event, showcasing the feasibility and robustness of the proposed low-cost embedded design.



(a) (b)

Figure 4. Final Prototype of the Manual Treadmill–Game Integration: (a) Participant Using the Treadmill; (b) Game Interface Showing Real-time Speed Visualization

4. CONCLUSION

This study presented the design and implementation of a low-cost embedded system for real-time treadmill speed tracking and game integration using a magnetic reed switch sensor and an ESP32 microcontroller. The system successfully converts roller rotation pulses from a manual treadmill into accurate linear speed data, which is transmitted to a PC for visualization in the Unity game engine. Experimental results demonstrated that the system achieved an average absolute error of 2.11% and an RMSE of 0.049 m/s, confirming the accuracy of the proposed measurement approach across a wide range of running speeds from 0.56 m/s to 3.13 m/s. The average latency of 48 ± 4.9 ms ensures real-time responsiveness between physical motion and virtual avatar updates, maintaining smooth gameplay synchronization. The integration of three push buttons for user input further enhances interactivity, allowing players to navigate menus and control gameplay directly from the treadmill interface. The proposed design effectively bridges physical exercise and digital gaming, demonstrating that a manual treadmill can be transformed into an interactive fitness device through simple embedded hardware and open-source software.

This system is particularly suitable for educational laboratories, rehabilitation training, and gamified exercise environments, where cost efficiency, modularity, and real-time feedback are essential. Future development will focus on incorporating wireless communication, higher-resolution sensors, and biometric feedback to enhance usability and expand its application to broader interactive fitness platforms.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to ICE BSD and BNI WonderX for providing full financial support for this research project. This work was fully funded and made possible through their sponsorship and collaboration. The authors also acknowledge Arthatronic Studio for providing the workshop facilities, tools, and technical support necessary for the development and assembly of the treadmill–game integration system. The final prototype was featured and demonstrated at the BNI WonderX 2025 event, showcasing its real-time interactive performance to the public. The authors greatly appreciate the support and opportunities provided by all contributing organizations, which enabled the successful completion and exhibition of this research.

REFERENCES

Adebisi, O. I., Ogundare, A. B., Erinosho, T. C., Sonola, M. O., & Adesanu, A. R. (2023). Development of a microcontroller and resistive touchscreen-based speed monitoring and control system for DC motor. *International Journal of Advances in Applied Sciences*, *12*(4). https://doi.org/10.11591/ijaas.v12.i4.pp350-360

Bowtell, M. V., Tan, H., & Wilson, A. M. (2009). The consistency of maximum running speed measurements in humans using a feedback-controlled treadmill, and a comparison with maximum attainable speed during overground locomotion. *Journal of Biomechanics*, *42*(15). https://doi.org/10.1016/j.jbiomech.2009.07.024

Hercog, D., Lerher, T., Truntič, M., & Težak, O. (2023). Design and Implementation of ESP32-Based IoT Devices. *Sensors*, *23*(15). https://doi.org/10.3390/s23156739

Kim, J., Gravunder, A., & Park, H. S. (2015). Commercial Motion Sensor Based Low-Cost and Convenient Interactive Treadmill. *Sensors (Basel, Switzerland)*, *15*(9). https://doi.org/10.3390/s150923667

Manser, P., Adcock-Omlin, M., & de Bruin, E. D. (2023). Design Considerations for an Exergame-Based Training Intervention for Older Adults With Mild Neurocognitive Disorder: Qualitative Study Including Focus Groups With Experts and Health Care Professionals and Individual Semistructured In-depth Patient Interviews. *JMIR Serious Games*, *11*. https://doi.org/10.2196/37616

Mirzaei, M., Ripka, P., & Grim, V. (2023). An eddy current speed sensor with a novel configuration of longitudinal and transversal coils. *Sensors and Actuators A: Physical*, *352*. https://doi.org/10.1016/j.sna.2023.114201

Mohammadzadeh, F. F., Liu, S., Bond, K. A., & Nam, C. S. (2015). Feasibility of a Wearable, Sensor-based Motion Tracking System. *Procedia Manufacturing*, *3*. https://doi.org/10.1016/j.promfg.2015.07.128

Ozdogar, A. T., Ertekin, O., Kahraman, T., Dastan, S., & Ozakbas, S. (2023). Effect of exergaming in people with restless legs syndrome with multiple sclerosis: A single-blind randomized controlled trial. *Multiple Sclerosis and Related Disorders*, *70*. https://doi.org/10.1016/j.msard.2022.104480

Park, Y. H., Lee, H. B., & Kim, G. W. (2023). Crack Monitoring in Rotating Shaft Using Rotational Speed Sensor-Based Torsional Stiffness Estimation with Adaptive Extended Kalman Filters. *Sensors*, *23*(5). https://doi.org/10.3390/s23052437

Reneaud, N., Zory, R., Guérin, O., Thomas, L., Colson, S. S., Gerus, P., & Chorin, F. (2023). Validation of 3D Knee Kinematics during Gait on Treadmill with an Instrumented Knee Brace. *Sensors*, *23*(4). https://doi.org/10.3390/s23041812

Roerdink, M., Coolen, B. (H)., Clairbois, B. (H)E, Lamoth, C. J. C., & Beek, P. J. (2008). Online gait event detection using a large force platform embedded in a treadmill. *Journal of Biomechanics*, *41*(12). https://doi.org/10.1016/j.jbiomech.2008.06.023

Talbi, K., El Ougli, A., Tidhaf, B., & Zrouri, H. (2023). Low-cost real-time internet of things-based monitoring system for power grid transformers. *International Journal of Electrical and Computer Engineering*, *13*(3). https://doi.org/10.11591/ijece.v13i3.pp2579-2588

Tough, D., Robinson, J., Gowling, S., Raby, P., Dixon, J., & Harrison, S. L. (2018). The feasibility, acceptability and outcomes of exergaming among individuals with cancer: A systematic review. In *BMC Cancer* (Vol. 18, Issue 1). https://doi.org/10.1186/s12885-018-5068-0

Yang, S., & Li, Q. (2012). Inertial sensor-based methods in walking speed estimation: A systematic review. In *Sensors (Switzerland)* (Vol. 12, Issue 5). https://doi.org/10.3390/s120506102

Zhao, L., Guo, X., Pan, Y., Jia, S., Liu, L., Daoud, W. A., Poechmueller, P., & Yang, X. (2024). Triboelectric gait sensing analysis system for self-powered IoT-based human motion monitoring. *InfoMat*, *6*(5). https://doi.org/10.1002/inf2.12520

Zhao, W., Han, Y., Wu, H., & Zhang, L. (2009). Weighted distance based sensor selection for target tracking in wireless sensor networks. *IEEE Signal Processing Letters*, *16*(8). https://doi.org/10.1109/LSP.2009.2022151