ORIGINAL PAPER

WEARABLE GAIT DEVICE FOR LONG-TERM MONITORING

ION CACIULA[1], GIORGIAN MARIUS IONITA[1], HENRI GEORGE COANDA[1],
DINU COLTUC[1], NICOLETA ANGELESCU[1], FELIX ALBU[1,*], DANIELA HAGIESCU[2]

Manuscript received: 17.07.2023; Accepted paper: 24.08.2023;
Published online: 30.09.2023.

Abstract. *This study describes a low-cost and easy to deploy gait monitoring system that uses an ESP32 microcontroller and an ICM-20948 module. The ESP32 microcontroller collects data from the ICM-20948 module and these data are used to train a convolutional neural network (CNN) to classify gait patterns into two categories: normal and pathological. The results show that the system can achieve a high accuracy for binary gait classification, being able to correctly classify 97.05% of the normal gait samples and 84.54% of the pathological gait samples. The power consumption of the devive was measured using a calibrated and dual-acquisition digital multimeter. The estimated operating time was around 12 hours, with a battery capacity of 1800 mAh LiPo type. Therefore, it could be used to track the gait of patients with neurological disorders or to assess the effectiveness of gait rehabilitation treatments.*

Keywords: *Gait monitoring; data acquisition; ESP32 microcontroller; ICM-20948 module; power consumption; convolutional neural networks.*

1. INTRODUCTION

Gait analysis systems are devices that monitor and evaluate how a person stands and walks, and how their posture and movement change over time. These systems can help to detect diseases and their complications early, identify the cause of problems in the muscles or bones, and guide treatment options. Examples of gait measurement methods are video analysis, pressure mats, and sensor devices [1-7].

Image processing-based approaches use camera triangulation, time of flight (ToF) systems, infrared thermography (IRT), and laser beam scanners [1-3].

Foot pressure can be measured by floor sensors that sense pressure or force to capture the subject's footsteps. Some electronic devices (called inertial measurement units, or IMUs) have multiple sensors that can measure pressure and force, as well as acceleration, rotation, position, temperature, and muscle activity [2,4,5].

Wearable sensors are placed on different parts of the patient's body to measure various gait features. Multiple parameters, such as accuracy, ease of use, and portability, demonstrate that portable systems based on body sensors are promising methods for gait analysis [2]. IMUs (inertial measurement units) are devices that have many benefits for gait analysis, being light, small, affordable, easy to carry, and non-intrusive. However, some factors need to be considered when using IMUs for gait analysis, such as how the sensors are placed on the body, which can affect the accuracy of the measurements [6]. Other limitations are provoked

[1] Valahia University of Targoviste, Department of Electronics, Telecommunications and Energy Engineering, 130004 Targoviste, Romania. E-mail: ion.caciula@valahia.ro.
* *Corresponding author: f_albu@yahoo.com.*
[2] Advanced Slisys SRL, 060104 Bucuresti, Romania. E-mail: aslisys.121@gmail.com.

https://doi.org/10.46939/J.Sci.Arts-23.3-c01

by environmental factors [7], magnetic interferences and energy consumption. The advantages of accelerometers and gyroscopes are their small size, low energy consumption, high durability. These advantages makes them suitable for long-lasting daily use. The process of training and testing machine learning models usually involves steps such as data preprocessing, segmentation, feature extraction, classification, and evaluation. Activity recognition and abnormal gait analysis use traditional machine learning classification models, such as support vector machines (SVMs) and K-nearest neighbors (KNNs) [8]. To improve the robustness of machine learning models for smart insole devices, it is important to use more data that is collected in a variety of settings. This will help the models to learn to recognize patterns that are common in different environments.

In this paper, we present a novel system for monitoring patient movement that uses an ESP32 microcontroller and an ICM-20948 module with sensors for motion and orientation. We also explain the architecture of the system for collecting data. A convolutional neural network (CNN) was used to classify the patient data into two gait categories. The main contribution consists of the combination of the implemented hardware and software solution.

The rest of this paper is structured as follows: Section 2 presents the proposed approach and its performances; Section 3 presents the results and a short discussion. Finally, the conclusions and future work are mentioned.

2. THE PROPOSED PROTOTYPE

The acquisition module presented in [8] was modified. The MPU9250 sensor [9] was replaced with the ICM-20948 module, which was developed by the same company, TDK Invensense. The reason is that the company no longer produces the MPU9250 module. There are other modules on the market that are equivalent to the original, but they do not offer adequate software or hardware support. Additionally, these modules have quality issues. The ICM-20948 module is available in both SMD [10] and development board [11] by the company Sparkfun. Four examples of the prototype implementation using external hardware modules are shown in Fig. 1. The block diagram for thise module is shown Fig.2.

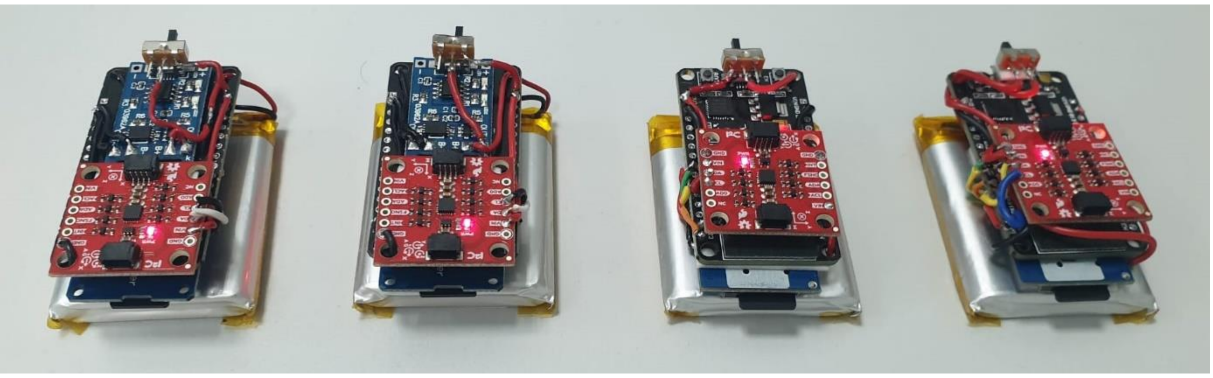


Figure 1. Prototypes that use the ICM-20948 sensor

The following external hardware modules were used:

- ESP32-DevKit 01 development board with ESP32 microcontroller, dual-core@240 MHz, 512 kBytes RAM [12];
- ICM-20948 Sparkfun module with I2C communication [10];

- LiPo external battery, capacity 1800 mAh, voltage 3.7V + charging module and voltage stabilization provided by the battery;
- microSD memory card module with SPI communication;
- switch for power/board programming on the microUSB port of the development board, respectively battery power / battery charging.

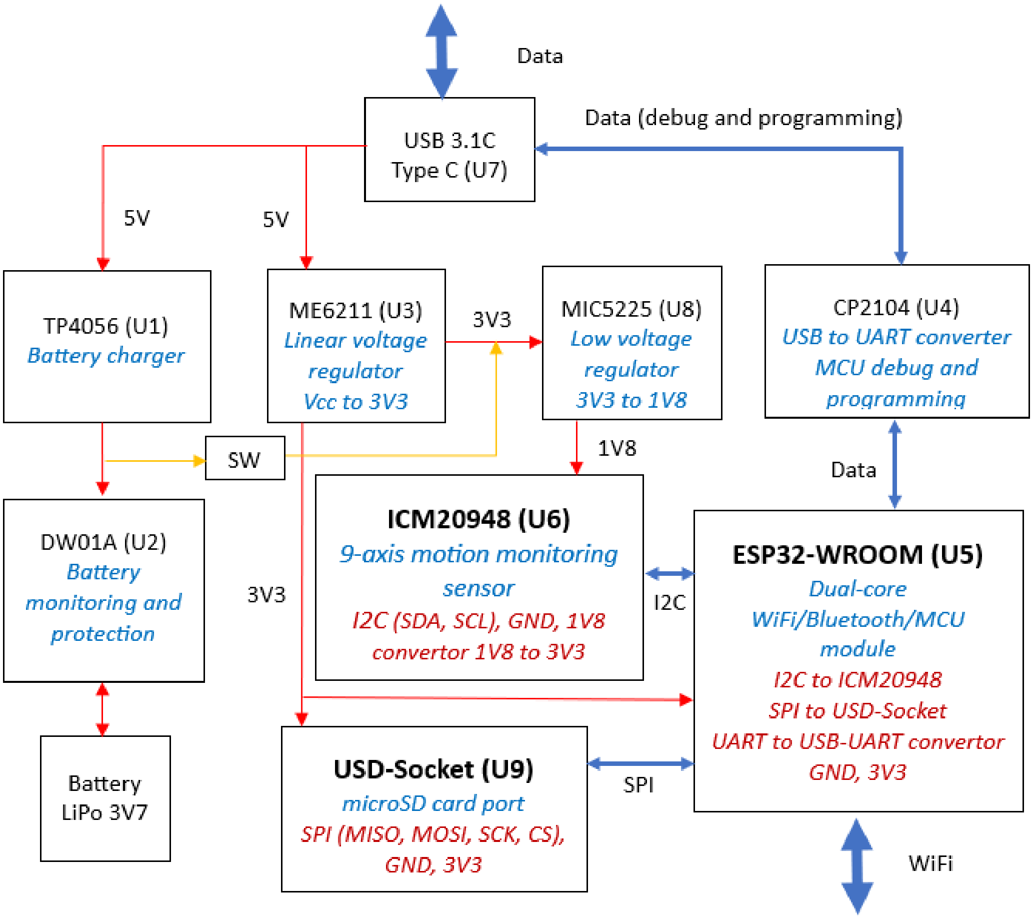


Figure 2. Block diagram of the circuit

The implemented prototype has the advantage of using the same type of support for programming the development board and charging the external battery. The system requires a 3V7 LiPo battery, which can be recharged using the TP4056 (U1) and the USB 3.1C connector. For its protection, a load monitoring circuit using DW01A (U2) is provided (Fig. 2)

Based on the results of tests, simulations, and experimental models, a new PCB variant was designed. The device is small (approximately 4 cm x 5 cm for the PCB) and has the ability to transmit data via WiFi and store it on the card. The electronic diagram shown in Fig. 3 includes the following main elements:

- a USB Type-C connector that makes the device easier to use. This connector can be used for both programming the device and for charging the battery. It requires a supply with a voltage of 5V;
- the TP4056 integrated circuit, which is used to charge the battery [13]. It provides constant voltage and current;
- the DW01A integrated circuit, which provides battery monitoring and protection [14];
- the SW1 switch, which turns on/off battery power;
- the CP2104 USB to UART converter, which is used for device programming [15];

- the ME6211C25M5G linear voltage regulator, which is responsible for producing 3.3V voltage for circuit operation;
- the UMH3N 2-transistor integrated circuit, which is used in device programming operation;
- LED 1: Indicates that the charging operation of the device is in progress;
- LED 2: Indicates the completion of charging the device;
- LED 3: Indicates the presence of 3.3V voltage, which means that the device is turned on;
- ESP32-WROOM: Dual-core WiFi/Bluetooth/MCU module;
- USD-Socket: microSD card port, which only allows micro SD memory cards and has a lock/unlock system by pushing the card;
- MIC5225: Low voltage regulator, used to produce the 1.8V voltage required for the ICM20948 sensor;
- ICM20948: 9-axis motion monitoring sensor. It contains a 3-axis accelerometer, a 3-axis gyroscope, a three-axis magnetometer, and a digital motion processor.

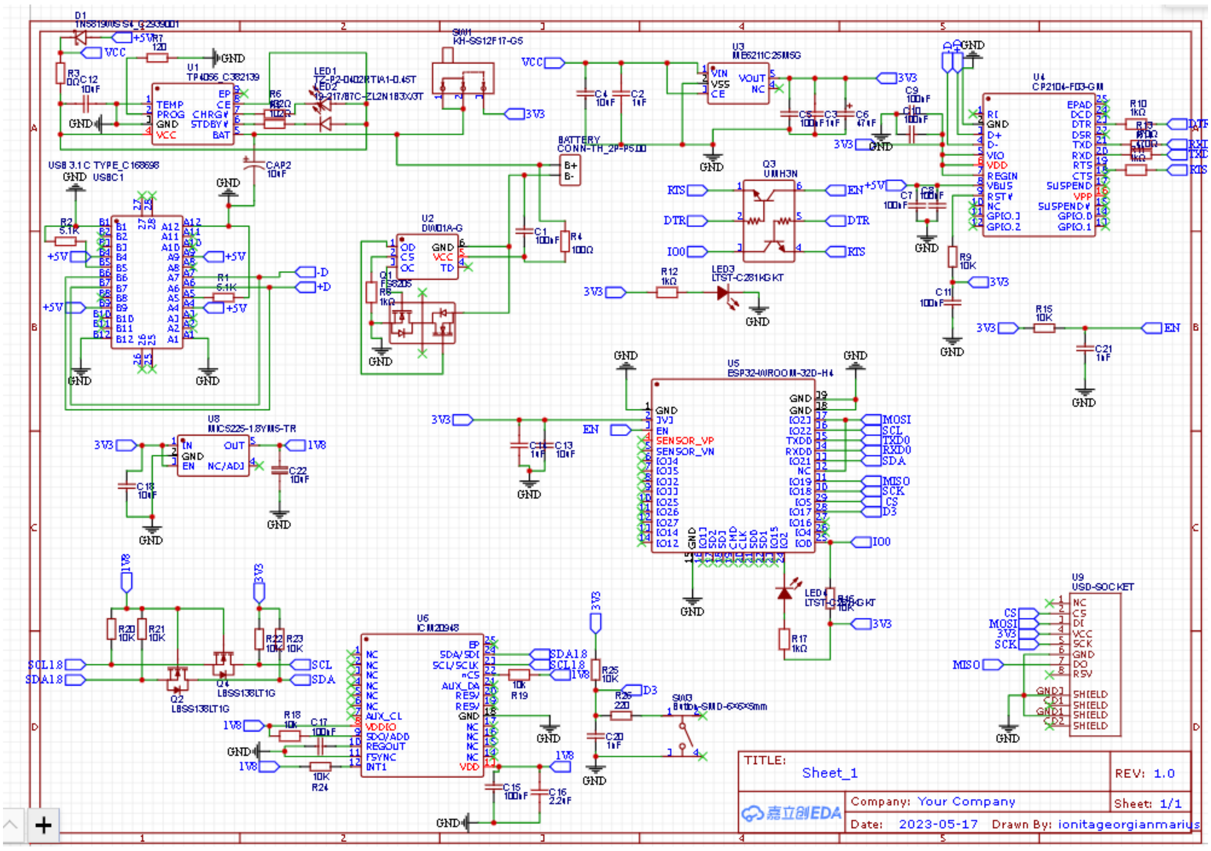


Figure 3. Electronic diagram of the circuit

The electronic circuit was developed in the EasyEDA program, the free web version. The PCB was also made in the same program. Figs. 4 and 5 show the PCB path design of the device (top layer and bottom layer, respectively). Two PCB path thicknesses were used: 0.254 mm and 0.127 mm. The 0.127 mm paths are used only for the ICM-2098 sensor, which has a footprint that allows a maximum thickness of 0.127 mm. The remaining components use a thickness of 0.254 mm, which is easily and safely achievable with modern PCB manufacturing methods.

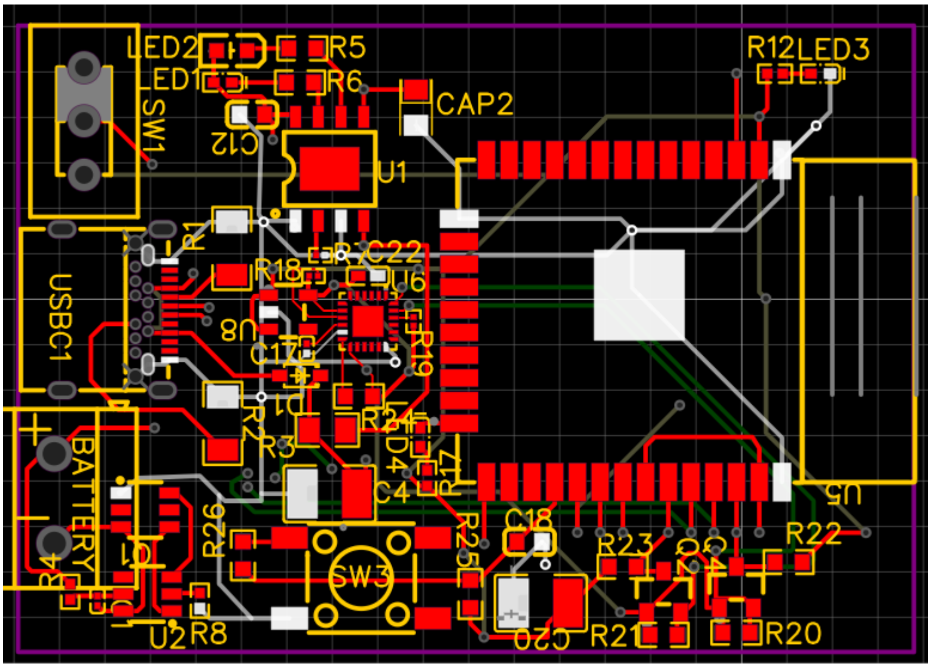


Figure 4. Track representation (Gerber file) – Top layer

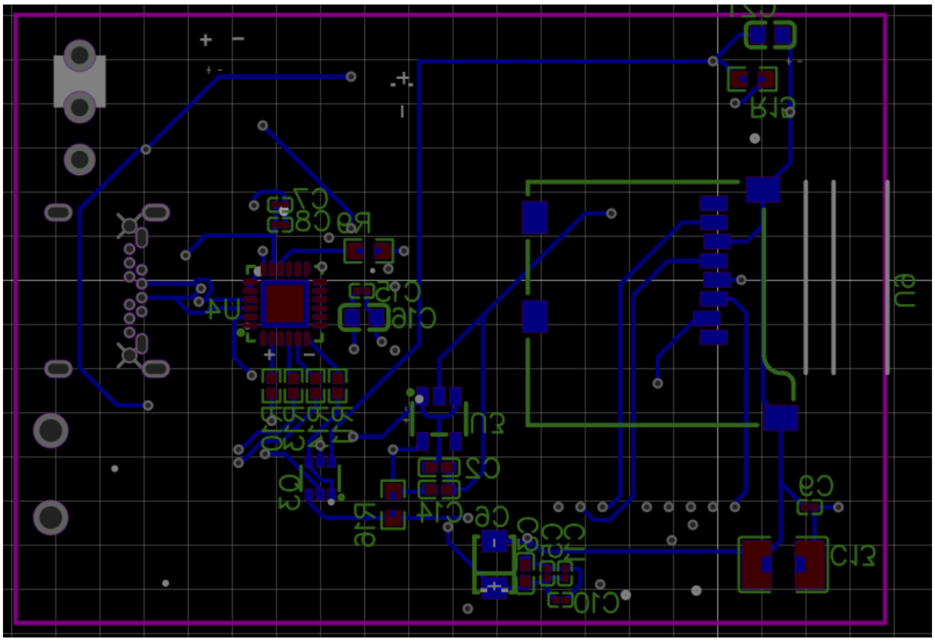
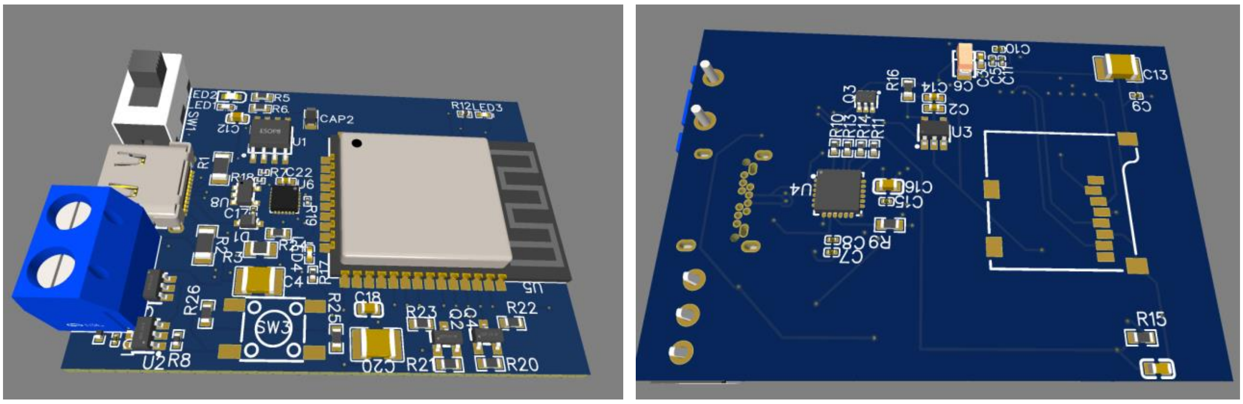


Figure 5. Track representation (Gerber file) – Bottom layer

The 3D representation of the device's PCB can be seen in Fig. 6. It consists of the following components:

- an acceleration sensor, gyroscope, and magnetometer, all in the ICM20948 module;
- an ESP32 WiFi module;
- two buttons;
- a microSD card socket;
- electronic elements for power, programming, and correct operation of the device;
- charging electronics and LiPo 1800 mAh battery protection.



View from above View from below

Figure 6. 3D representation of device PCB

The modified architecture for data collection in this report is shown in Fig.7. Unlike in [8], the hardware has been modified to work only with the SD card module via SPI communication. The system supports microSD cards. The file system for saving data is FAT 32-bit. The ways of connecting to the device's server are the same as those of the previous work [8], but the volume of data transmitted to the server has been greatly reduced.

The complexity and latency of communication between the server and the application using the HTTP protocol was reduced by transmitting parameters using the GET method instead of the POST method. This also reduced the volume of data transmitted by the GET method.

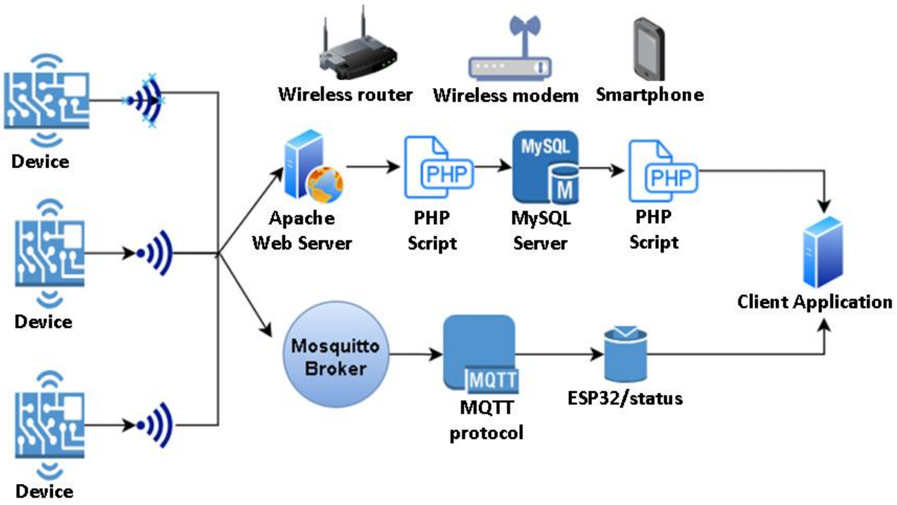


Figure 7. Patient Data Acquisition System Architecture

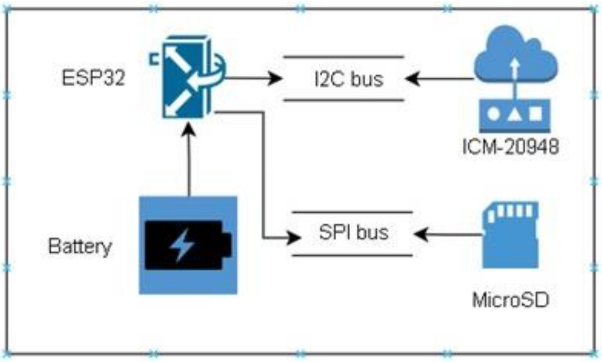


Figure 8. Internal architecture of the data acquisition module from the ICM-20948 sensor

The ESP-32 microcontroller and the ICM-20948 module communicate using the I2C communication protocol (Fig. 8). The I2C protocol is a two-wire protocol that allows multiple devices to share the same bus. This reduces the number of ports required for the three sensors integrated into the ICM-20948 module. The ICM-20948 module is a device that combines a three-axis accelerometer and gyroscope, and a three-axis magnetometer.

The I2C protocol operates at a clock signal frequency of 400 kHz. This is a fast enough frequency to ensure that the data is transferred quickly and accurately. The ESP32 microcontroller has two processing cores, the first core for data acquisition in RAM and the second core for saving data to the microSD memory card.

Data calibration was performed using the MotionCal program. The magnetic field varies with the location of the device. The calibration matrices for the magnetometer after running the program are shown in Eqs. (1) and (2) as follows:

$$M_{bias} = [-5.47 \;\; 6.96 \;\; 32.79] \tag{1}$$

$$M_{A_{inv}} = \begin{bmatrix} 1.044 & -0.003 & 0.012 \\ -0.003 & 0.971 & 0.001 \\ 0.012 & 0.001 & 0.986 \end{bmatrix} \tag{2}$$

For the gyroscope we have the following correction matrix from Eq. (3):

$$G_{offset} = [17.2 \;\; 173.7 \;\; -90.5] \tag{3}$$

The FreeRTOS library was chosen to assign the acquisition task to the first kernel and to save data on the memory card using the second core. The FreeRTOS library is supported by the ESP32 microcontroller. The characteristics of the two tasks are defined in the following table:

Table 1. Assigning tasks using FreeRTOS

Task		**Execution priority**	**Allocated core**
acquireDataTask	Data acquisition from ICM-20948 module Data transmission using the MQTT protocol	configMAX_PRIORITIES-2	0
sendDataTask	Save data on the SD card	configMAX_PRIORITIES-1	1

The highest priority that can be assigned to a task is configMAX_PRIORITIES-1. The first core in the ESP32 microcontroller has index 0 and the second core has index value 1.

The acquireDataTask() acquisition task performs the following operations:

- data collection from the ICM-20948 module;
- correction of acquired data using calibration matrices;
- calculation of yaw, roll, and pitch rotation angles using the Mahony algorithm.

The advantages of using static global allocation in RAM over the dynamic allocation are the following:

- reduces execution time. Static global allocation eliminates the need for memory allocation and freeing operations, which can be time-consuming;
- reduces RAM fragmentation. Static global allocation ensures that all allocated memory is contiguous, which can help to reduce RAM fragmentation;
- allows for estimation of remaining RAM. At the time of application compilation, the amount of RAM that is statically allocated is known. This allows for estimation of the amount of remaining RAM that is available for dynamic allocation.

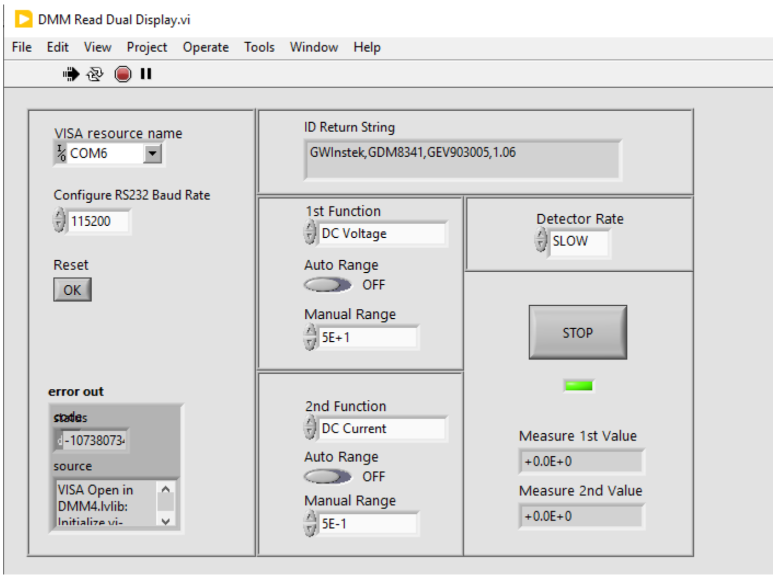
The acquired data will be saved to the memory card in batch mode, with a delimiter used to identify each record. The record block size is 100 records, and each block of records is a variable of type String. Real number values are saved to the nearest 6 decimal places. The code in the save task is run in an infinite loop. To save 500 records from RAM to the memory card, the execution time is less than 3 seconds. This may vary depending on the following factors:

- memory card wear: the more the memory card is used, the longer it may take to save data;
- memory card write time: the write speed of the memory card may vary depending on the controller on the card.
- allocation/reallocation of flash memory blocks: the time it takes to allocate or reallocate flash memory blocks may also vary.

The latency for communicating with the Apache web server using the HTTP protocol (communication port 80) is approximately 300 milliseconds. The wireless communication standards supported by the ESP32 microcontroller are 802.11 b/g/n, 2.4 GHz frequency. The wireless connection for the developed module can be provided by one of the following devices: A smartphone with hotspot function enabled; a 3G/4G/5G internet router; a wireless router connected to the internet. The data is saved on the memory card in text format. This means that the data is saved as a series of characters, rather than as a binary file. This can result in a larger data volume, but it also makes the data more portable and easier to analyze. The data can be imported into a database to reduce the data volume. The text format also allows the data to be imported and analyzed in several programs. This is because most programming languages can read and write text files.

2.1. *ENERGY CONSUPTION MEASUREMENT*

To measure energy consumption, we used the GDM-8341 digital multimeter produced by GWINSTEK. This multimeter allows for simultaneous measurement of voltage and current intensity, and the values can be viewed using the LabView DMM Read Dual Display.vi program. The source code for this program is available, so we modified it to save records in text format. The interface and the modified code are shown in Figs. 9 and 10, respectively.



VISA resource name: COM6

Configure RS232 Baud Rate: 115200

Reset

ID Return String: GWInstek,GDM8341,GEV903005,1.06

1st Function: DC Voltage

Auto Range: OFF

Manual Range: 5E+1

Detector Rate: SLOW

error out

status

-10738073

source

VISA Open in DMM4.lvlib: Initialize.vi-

2nd Function: DC Current

Auto Range: OFF

Manual Range: 5E-1

Measure 1st Value: +0.0E+0

Measure 2nd Value: +0.0E+0

Figure 9. Communication interface with the GDM-8341 multimeter

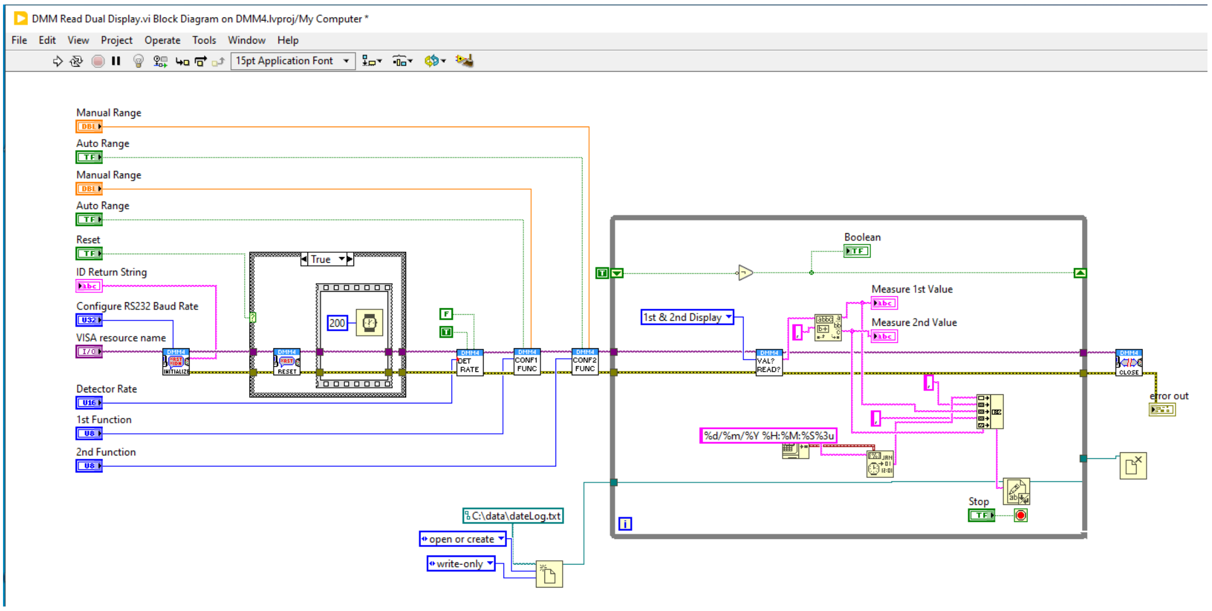
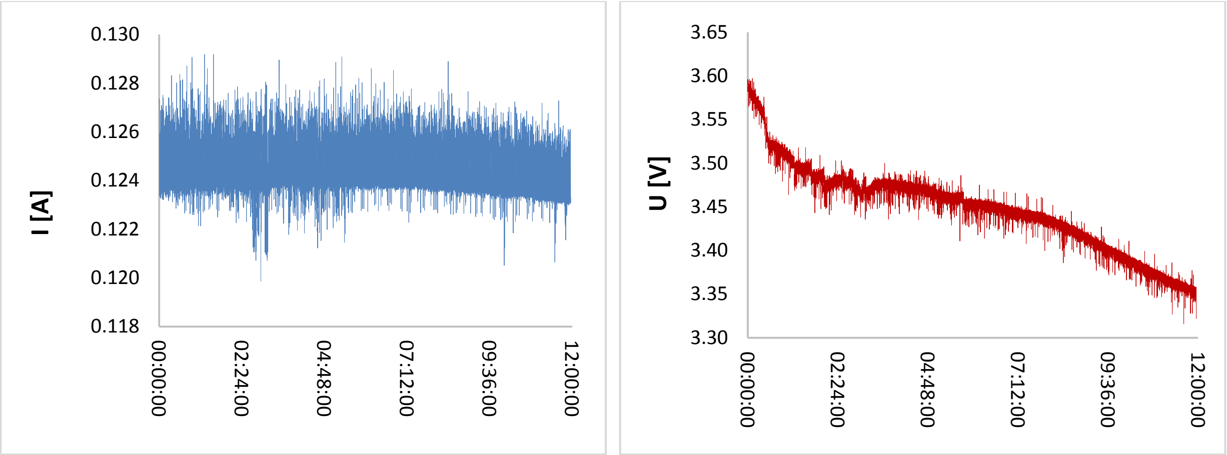


Figure 10. Modified code for communication interface with multimeter GDM-8341

The device was designed to retrieve data from patients for both short-term and long-term monitoring. To test its service life, the experimental model was subjected to a test that monitored the voltage and current after the battery management circuit in normal operation. The device needs a minimum of 120 mA to perform its tasks (calculation, sensor reading, wireless communication, and writing to the card). The experiment started with the device charged and ended when it could no longer perform all its tasks. The experiment results presented in Fig. 11 show that the device operated for 11 hours, 57 minutes, and 11 seconds with an average current of 124.110 mA at a voltage between 3.59 V and 3.32 V. In Fig. 11a, we can observe a relatively stable consumption, and the supply voltage decreases, like the discharge graphs of LiPo batteries. This indicates that the device is operating correctly and is using the battery to its maximum capacity. Calculations show that the device used 5.21262 Wh of energy during the test period. This means that the device can use 78.32% of the battery's nominal capacity during operation. After this value, the device stopped working.



a) Current *vs.* time **b) Voltage *vs.* time**

Figure 11. Monitoring the current and electrical voltage parameters of the device

2.2. DESIGN AND EVALUATION OF GAIT DETECTION METHODS USING CNNS

Next, the use of CNN was investigated for gait type classification. A set of weights for network neurons and a set of weights for convolution filters are obtained, which can be passed on for the classification part. New CPU and graphics card architectures provide support for optimized data types under 32-bit, enabling optimal transmission and deployment of classification modules. The CNN architecture was based on AlexNet. The same dataset from [8] including real data experiments from 20 patients was used for training and validation of the CNN.

Convolutional neural networks have several advantages over other machine learning methods for gait classification. First, CNNs can reduce the number of neurons by using convolution filters. Second, CNNs can be accelerated using hardware such as NVIDIA video cards. Third, CNNs are more flexible than other methods, such as KNN and SVC.

Data from patients were classified into two gait types: pathological and healthy. The pathological gait class was trained and tested on 80,580 records, which corresponds to approximately 9 minutes of data. The dataset was divided into two subsets: the training set and the validation set (80% of the values were used for training the neural network, and 20% were used for validation). In gait classification, CNNs learn to extract features from 1D data, such as the acceleration and tilt data collected from a sensor. The use of a CNN led to the following changes in the architecture of the MLP solution from [8]:

- the input layer was changed to size 1×9×K, where K represents the number of consecutive recordings that will feed the network, and 9 represents the 9 parameters acquired from the sensor: linear acceleration on 3 axes, angular acceleration on 3 axes, respectively tilts on 3 axes determined using yaw, roll and pitch parameters. Convolution layers were added to implement the convolution operation in 1D;
- the activation function was changed from sigmoid to ReLU, which has a lower computational complexity;
- the Dense layer and sigmoid activation function appear in the final layer for binary or multi-class classification. The Dense layer is a fully connected layer that connects all neurons in the previous layer to all neurons in the current layer. The sigmoid activation function is a non-linear function that outputs a value between 0 and 1, which is ideal for binary classification.

The script was written in Python 3.9, and the CNN network was implemented using TensorFlow. The processor architecture on which the Python code ran provided hardware acceleration support by implementing the AVX, AVX2, and AVX-512 instruction set. The hardware configuration for running Python code was as follows:

- Intel i7-11800H@2.5GHz processor (8 physical cores/16 threads) with support for AVX, AVX-2 and AVX-512 instruction set for acceleration;
- 64 GB DDR4 RAM;
- 1 x 2TB M.2 SSD, PCIE 4.0 interface.

3. RESULTS AND DISCUSSION

The proposed CNN architecture and the used running parameters are shown in Table 2. After running the program, a graph with the confusion matrix was generated and shown in Fig. 12. The *Ox* axis shows the predicted category for both categories, and the *Oy* axis shows the correct category. The values on the matrix's main diagonal represent the number of correct predictions of the belonging category. The sum of the values on each line in the confusion matrix represents the number of records for each category. For a performance comparison with previous classification methods, only values from accelerometer and gyroscope sensors were used.

Table 2. CNN's architecture and used global parameters

Layer	No. of neurons / dimension of the convolution filters	Parameters	Value
Input	6	Batch_size	64
Conv1D	32 / 3	Solver	Adam
Conv1D	64 / 3	Beta_1	0.99
Conv1D	32 / 3	Beta_2	0.999
Dense	32	Epsilon	1E-8
Output	2	No. of Epochs	10

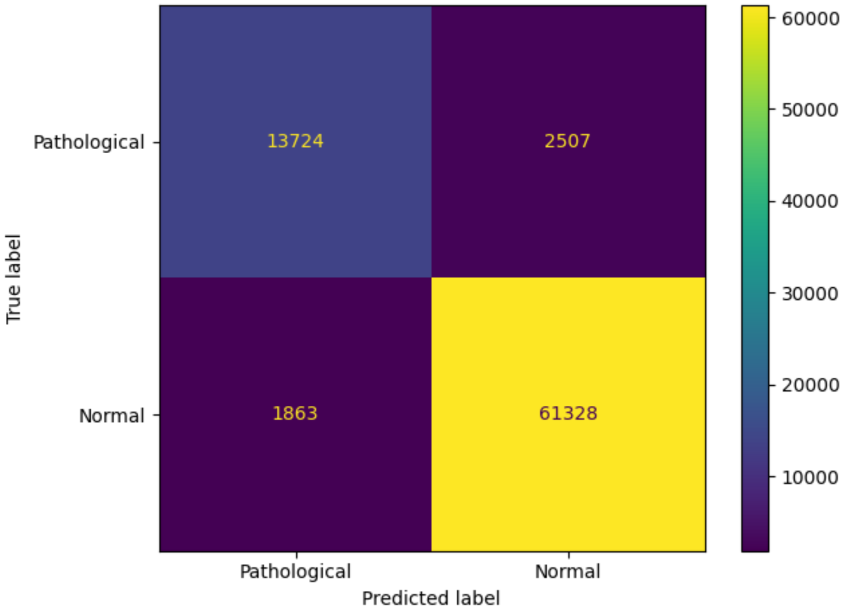


Figure 12. Confusion Matrix

The CNN neural network has a superior performance according to the results obtained and comparable with those reported in [8]. However, it is important to note that the network's performance may vary depending on the dataset, and the convolution filters used. For the validation set, the results were as follows: Pathological gait category: 84.54%; Normal gait category: 97.05%. These results show that the CNN neural network is able to accurately classify gait patterns, with a high accuracy for both pathological and normal gait categories. Other changes from the previous solution were: data transmission using the HTTP protocol has been replaced by saving to a microSD memory card. The POST method of transmitting parameters using the HTTP protocol was replaced by the GET method, and the size of the message to the server was significantly reduced.

4. CONCLUSIONS

A prototype for gait monitoring based on ICM-20948 module has been designed. The data acquired from the ICM-20948 module were calibrated, and the values obtained from the integrated magnetometer in the ICM-20948 module were used to calculate the angles of rotation (yaw, roll, and pitch) on the Ox, Oy, and Oz axes. The algorithm for calculating the rotation angles angles is an optimized version of the Mahony algorithm, which allows the acquisition operation to be performed in lower than 10 milliseconds, thus ensuring the acquisition frequency of 100 samples per second. Our system achieved a good gait classification performance for both normal gait and pathological gait categories. We proved that the proposed gait monitoring system could monitor the gait of patients for about 12 hours, therefore it could be useful for varios gait rehabilitation treatments. Our future work will optimize the presented platform's architecture and test suitable classification algorithms for small and imbalanced datasets [16] on an extended database including data acquired from many more patients.

Acknowledgment: *This work was carried out for the POSTUREC/121690 research project, co-financed through the POC 1.2.1-Innovative technological project program.*

REFERENCES

[1] Biswas, N., Chakrabarti, S., Jones, L.D., Ashili, S., *Materials Today Communications*, **35**, 2023

[2] Ribeiro, N. F., Santos C. P., *IEEE 5th Portuguese Meeting on Bioengineering (ENBENG)*, 1, 2017.

[3] Sethi, D., Bharti, S., Prakash, C., *Artificial Intelligence In Medicine*, **129**, 2022.

[4] Gupta, R., Kumari, S., Senapati, A., Ambasta, R. K., Kumar, P., *Ageing Research Reviews*, 102013, 2023.

[5] Lopes, T.A.M., *MSc thesis Nova School of Science & Technology*, Lisbon, 2021. Available online: https://run.unl.pt/bitstream/10362/141082/1/Lopes_2021.

[6] Küderle, A., Roth, N., Zlatanovic, J., Zrenner, M., Eskofier, B., Kluge, F., *PLoS ONE*, **17**(6), e0269567, 2022.

[7] Preatoni, E., Bergamini, E., Fantozzi, S., Giraud, L., Bustos, A.S.O., Vannozzi,G., Camomilla, V., *Sensors*, **22**(9), 3225, 2022.

[8] Caciula, I., Ionita, G.M., Coanda, H.G., Angelescu, N., Albu, F., Hagiescu, D., *Proceedings of 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI),* 409, 2023. https://doi.org/10.1109/ECAI58194.2023.10194076.

[9] MPU-9250 Product Specification. Available online:
https://invensense.tdk.com/download-pdf/mpu-9250-datasheet/

[10] ICM-20948 World's Lowest Power 9-Axis MEMS MotionTracking™ Device.
Available online: https://www.farnell.com/datasheets/2632662.pdf

[11] SparkFun 9DoF IMU Breakout - ICM-20948 (Qwiic)

[12] ESP32-WROOM-32 Datasheet. Available online:
https://www.espressif.com/sites/default/files/ documentation/esp32-wroom-32_datasheet_en.pdf

[13] TP4056 1A Standalone Linear Li-lon Battery Charger with Thermal Regulation in SOP-8. Available online:
https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf

[14] DW01A Datasheet. Available online:
https://datasheet.lcsc.com/szlcsc/1901091236_PUOLOP-DW01A_ C351410.pdf

[15] CP2104 Single-chip USB-to-UART bridge. Available online:
https://www.silabs.com/documents/public/ data-sheets/cp2104.pdf

[16] Alzubaidi, L. et al., *Journal of Big Data*, **10**, 46, 2023.