Measurement of Muscle Signals and their Processing Using AI

Martin Uhrina, Michal Kubaščík, Ján Kapitulík

***Abstract*—**The article focuses on monitoring and processing muscle activity signals using the MyoWare 2.0 sensor and the ESP32-P4 microcontroller. The MyoWare 2.0 sensor provides detailed electromyographic (EMG) data that can be processed in real time to analyze muscle performance, fatigue, or movement patterns. The ESP32-P4 microcontroller, equipped with advanced connectivity and built-in AI support, is used for data processing. Analog signals from the sensor are converted to digital form using an ADC and subsequently filtered to reduce noise. The microcontroller enables AI models, such as convolutional and recurrent neural networks, to classify gestures and monitor muscle fatigue. Processed data can be shared via Wi-Fi to cloud platforms like Google Firebase or ThingSpeak, locally using Bluetooth, or visualized on OLED or LCD displays. This approach integrates signal monitoring with artificial intelligence, offering practical applications in rehabilitation, robotics, and IoT.

***Keywords*—**MyoWare 2.0, ESP32-P4, EMG signals, artificial intelligence, TensorFlow Lite, signal processing, Edge AI, IoT, Bluetooth, cloud platforms, OLED display

I. INTRODUCTION

Muscle activity is the fundamental driver of movement and communication in our body. The ability to capture and analyze the electrical signals generated by muscles can be applied in areas such as rehabilitation, sports performance, and the development of smart prosthetic technologies. These signals, recorded through electromyography (EMG), can reveal not only the intensity of muscle activity but also information about its condition, fatigue tolerance, and coordination with other muscles. Modern technology allows us not only to collect this data but also to process it efficiently.

The MyoWare 2.0 sensor is an advanced yet affordable tool for measuring muscle activity. With its compact design and easy integration with microcontrollers like the ESP32-P4, it allows for the integration of this data into artificial intelligence (AI)-driven applications. The ESP32-P4 is a next-generation microcontroller that combines high computational power with advanced peripherals. Thanks to its ability to process analog signals from the MyoWare sensor, apply filters to remove noise, and deploy AI models directly on the hardware, it represents an ideal platform for applications utilizing muscle signals. By combining artificial intelligence with EMG data, we can:

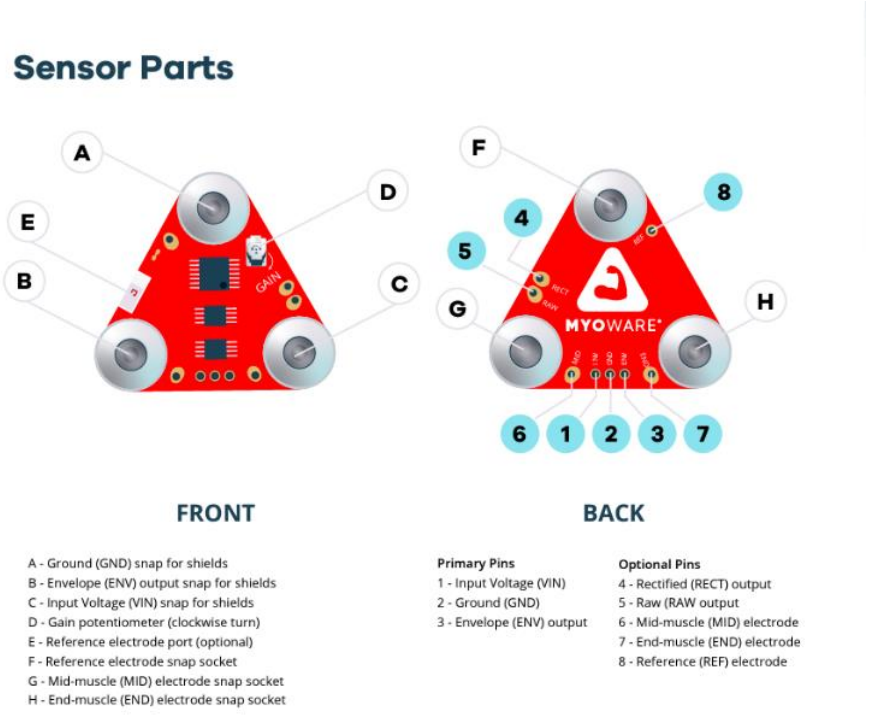
- Recognize complex movements or gestures for controlling robotic systems.
- Monitor muscle fatigue and prevent overloading during physical exertion.
- Analyze patient progress in rehabilitation in real-time.
- Create smart IoT devices that respond to muscle activity.

II. SENSOR MYOWARE 2.0

It is a sensor designed to detect and process surface electromyographic (sEMG) signals generated during muscle activity. When the brain communicates with muscles, it generates electrical signals that activate motor units, creating measurable electrical potentials. These

Martin Uhrina, University of Žilina, Slovak Republic (e-mail: uhrina@stud.uniza.sk)
Michal Kubaščík, University of Žilina, Slovak Republic (e-mail: michal.kubascik@fri.uniza.sk)
Ján Kapitulík, University of Žilina, Slovak Republic (e-mail: jan.kapitulik@fri.uniza.sk)

electrical signals are very weak, typically ranging from 2 to 10 mV in voltage. To give an idea of how such a signal works, we can use the example of bicep contraction. When the arm is extended and the bicep is relaxed, the electrical potential in the muscle will be very small, approximately 2 mV. However, when the muscle is contracted, the brain sends a more intense signal to the muscle, increasing the electrical potential in the muscle to about 10 mV.

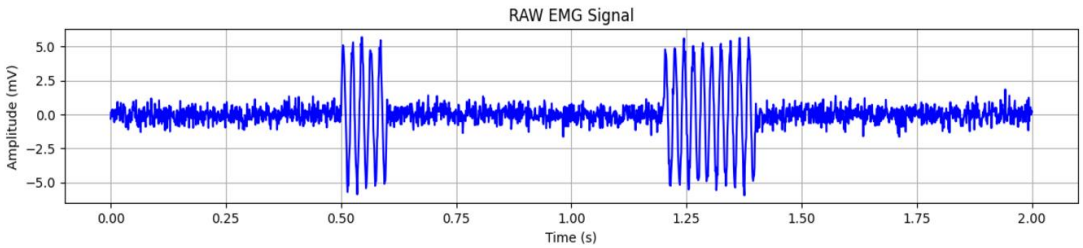


Picture 1 Sensor parts

The signal is measured by connecting three electrodes (MID, END, REF) to the muscle. The MID and END electrodes are used to measure the differential signal and are attached to a specific muscle, while the REF electrode provides a reference point located on a neutral part of the body, such as a bone.

A. RAW

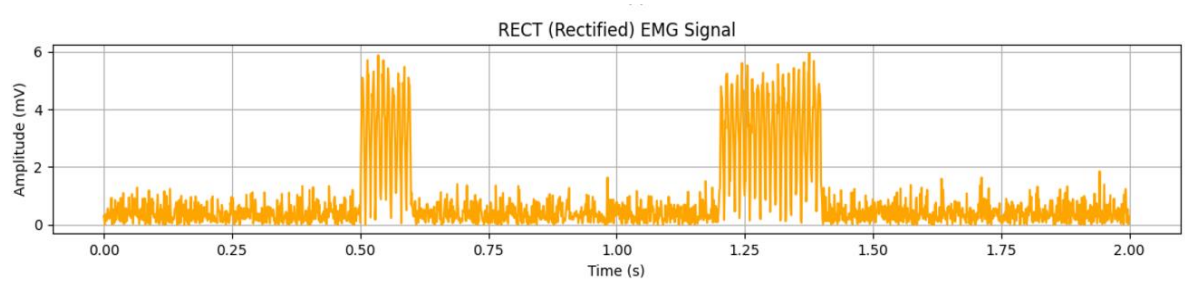
The raw output represents a 200-fold amplified and continuously monitored electrical signal from the muscles. It contains all the details about muscle activity, including high-frequency fluctuations and noise. This is an unprocessed signal. It typically ranges from 20 to 500 Hz, which corresponds to the physiological characteristics of sEMG signals. Such a signal can be used for detailed analysis of muscle activity or for frequency-based movement analysis.



Picture 2 Raw signal

B. RECT

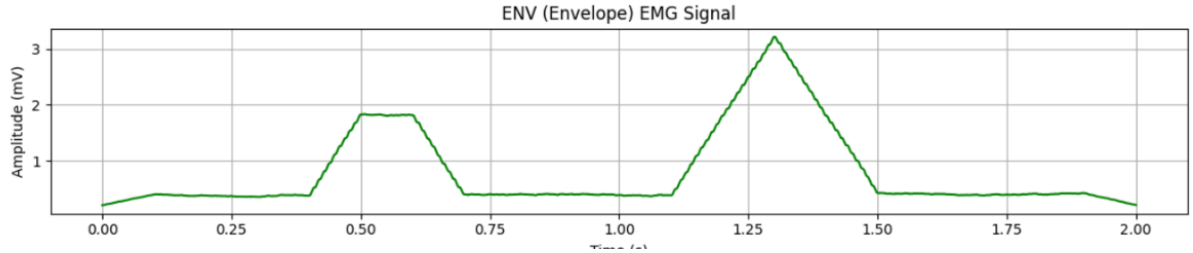
This signal represents the processed RAW signal, where all negative values are transformed into positive ones. This process is called full-wave rectification and ensures that the signal is easier to process. It is suitable for use in projects where advanced signal filtering is not required.



Picture 3 Rect signal

C.ENV

This output, the smoothed signal, represents the amplitude of muscle activity over time. Its calculation involves filtering high frequencies and applying a smoothing algorithm to remove minor fluctuations. It has a low frequency of approximately 3.6 Hz, which makes real-time processing easier. The amplification of this output can be adjusted using the potentiometer on the sensor. The amplification is calculated using the formula G = 200 * R / 1kΩ. Since this output is suitable for real-time applications, it can be used in systems such as robotic control, prosthetic devices, or monitoring patient progress during rehabilitation.



Picture 4 ENV signal

III. USE OF ESP32-P4 FOR MONITORING MUSCLE ACTIVITY

To process the electrical signals from the MyoWare 2.0 sensor, we can use the ESP32-P4 microcontroller, a powerful device designed for applications that require rapid data analysis, advanced connectivity options, and artificial intelligence support. This microcontroller offers a wide range of communication interfaces, such as Wi-Fi 6, Bluetooth 5.0, SPI, UART, and I2C, ensuring flexibility when integrating into various systems. The ESP32-P4 not only allows for efficient processing of signals from the MyoWare 2.0 sensor but also facilitates sharing them across different platforms, making it ideal for modern applications in healthcare, sports, and IoT.

A. Sensor and Microcontroller Integration

For proper collaboration between the MyoWare 2.0 sensor and the ESP32-P4 microcontroller, their connection needs to be set up correctly. The MyoWare sensor provides three types of analog outputs (RAW, RECT, ENV), which can be connected to the analog input of the ESP32-P4. The microcontroller can then receive and process these signals.

The sensor can be powered directly from the microcontroller via either the 3.3 V or 5 V output, depending on the requirements.

Once the signal is received at the analog input of the ESP32-P4, it is converted to a digital form using an ADC (Analog-to-Digital Converter). This process allows the microcontroller to further process the acquired data. The ESP32-P4 can apply programmable filters to remove noise and highlight the desired signal. For example, a low-pass filter isolates the ENV signal, while a high-pass filter can remove certain interferences.

To read data from the ADC on the ESP32-P4, we can use the driver/adc library, which allows for the configuration of ADC channels, real-time reading of analog signals, and continuous

sampling using DMA. For more advanced signal analysis and processing, such as filtering, libraries like ESP-DSP or CMSIS-DSP can be used, offering a wide range of tools for digital signal processing.

B. Processing Measured Data Using AI

The ESP32-P4 microcontroller has built-in support for artificial intelligence, enabling signal processing directly on the device using the Edge AI method. This signal processing approach reduces reliance on cloud services, thereby speeding up data processing.

After the signal is received via the ADC converter, preprocessing is required. First, filters are applied to clean the data of noise, and then the signal values are adjusted to an appropriate range for processing by an AI model. Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) can be used for such signal processing. Convolutional networks are ideal for recognizing patterns, gestures, or movements, while recurrent networks are more suitable for monitoring muscle fatigue or analyzing time-series data.

To implement AI models on the ESP32-P4 microcontroller, libraries such as TensorFlow Lite for Microcontrollers and uTensor can be used. These libraries enable the deployment of pre-trained AI models directly onto the microcontroller, which is ideal for devices with limited computational power. uTensor, like TensorFlow Lite, is designed to efficiently run models on low-power devices. Additionally, libraries like ESP-DSP and CMSIS-DSP can be utilized.

AI models can identify various muscle movements, such as hand flexion, finger contraction, or grip, making them ideal for controlling robotic devices. Furthermore, they can be used to detect muscle fatigue, such as by monitoring the drop in signal amplitude during exercise or other physical activities.

C. Sharing Measured Data

The measured signal and processed data can be shared across various platforms for better representation. Using Wi-Fi, these data can be sent to cloud storage services like Google Firebase, AWS IoT Core, ThingSpeak, or to a private server. This allows for long-term archiving, easy access, and later analysis.

For local communication with mobile applications, the Bluetooth interface can be used. ESP-IDF provides the esp_ble_gatts library, which allows the creation of a Bluetooth server for sending real-time data.

For local display, we can use a screen connected via I2C or SPI, where we can display the measured signals or processed results. For working with OLED displays, libraries like Adafruit SSD1306 or U8g2 can be used, which are powerful and provide flexible graphical display options.

Table 1 Methods of Sharing

Method of Sharing	Advantages	Disatvantages
Cloud(Wi-Fi)	Global access to data	Requires internet connect
Bluetooth	Local sharing(mobile app)	Limited range (approx. 10 meters)
OLED/LCD display	Direct display from the device	Limited space for detailed data

IV. Conclusion

The integration of advanced sensing technologies, such as the MyoWare 2.0 sensor, with the versatile capabilities of the ESP32-P4 microcontroller represents a significant step forward in muscle signal monitoring and analysis. This combination allows for real-time data processing and the application of artificial intelligence directly on embedded devices, reducing reliance on external computing resources. By leveraging machine learning models, it becomes possible to classify gestures, detect muscle fatigue, and enable innovative applications in fields such as

rehabilitation, robotics, and IoT.

Additionally, the flexibility in sharing measured and processed data—whether via cloud storage, Bluetooth communication, or direct visualization on a display—ensures usability in diverse environments. Despite the challenges of limited hardware resources, the use of optimized libraries like TensorFlow Lite and ESP-DSP demonstrates the feasibility of deploying powerful AI tools on embedded platforms. Future work could focus on expanding the range of AI models, improving signal accuracy, and integrating the system into more comprehensive healthcare and robotic systems, further unlocking the potential of electromyographic analysis in practical applications.

REFERENCES

[1] "SparkFun 9DoF IMU Breakout - MPU9250," *SparkFun Electronics*. [Online]. Available: https://www.sparkfun.com/products/21265. [Accessed: Dec. 6, 2024].

[2] "Myoware Muscle Sensor," *Myoware*. [Online]. Available: https://myoware.com/products/muscle-sensor/. [Accessed: Dec. 6, 2024].

[3] "SparkFun DEV-18977 Product Overview," *Mouser Electronics*. [Online]. Available: https://www.mouser.com/pdfDocs/Productoverview-SparkFun-DEV-18977.pdf. [Accessed: Dec. 6, 2024].

[4] Espressif Systems, "ESP-DSP," GitHub Repository. [Online]. Available: https://github.com/espressif/esp-dsp. [Accessed: Dec. 6, 2024].

[5] "SparkFun 9DoF IMU Breakout - LSM9DS1," *SparkFun Electronics*. [Online]. Available: https://www.sparkfun.com/products/21269. [Accessed: Dec. 6, 2024].

[6] Espressif Systems, "ESP32-P4 Function EV Board User Guide," *Espressif Systems*. [Online]. Available: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32p4/esp32-p4-function-ev-board/user_guide.html. [Accessed: Dec. 6, 2024].

[7] Espressif Systems, "ESP32-P4," *Espressif Systems*. [Online]. Available: https://www.espressif.com/en/news/ESP32-P4. [Accessed: Dec. 6, 2024].

[8] Espressif Systems, "ESP-TFLite Micro," GitHub Repository. [Online]. Available: https://github.com/espressif/esp-tflite-micro. [Accessed: Dec. 6, 2024].

[9] Eloquent Arduino, "TensorFlow Lite on ESP32 for TinyML," *Eloquent Arduino*. [Online]. Available: https://eloquentarduino.com/posts/tensorflow-lite-tinyml-esp32. [Accessed: Dec. 6, 2024].