❒   514

A prosthetic limb managed by sensors-based electronic system: Experimental results on amputees

F. Gaetani[1], R. de Fazio[2], G. A. Zappatore[3], P. Visconti[4]
[1,3]BionIT Labs Company, Lecce, street G. Antonio Orsini del Balzo, Italy
[2,4]Department of Innovation Engineering, University of Salento, Lecce, Italy

Article Info	ABSTRACT
	Taking the advantages offered by smart high-performance electronic devices, transradial prosthesis for upper-limb amputees was developed and tested. It is equipped with sensing devices and actuators allowing hand movements; myoelectric signals are detected by Myo armband with 8 ElectroMyoGraphic (EMG) electrodes, a 9-axis Inertial Measurement Unit (IMU) and Bluetooth Low Energy (BLE) module. All data are received through HM-11 BLE transceiver by Arduino board which processes them and drives actuators. Raspberry Pi board controls a touchscreen display, providing user a feedback related to prosthesis functioning and sends EMG and IMU data, gathered via the armband, to cloud platform thus allowing orthopedic during rehabilitation period, to monitor users' improvements in real time. A GUI software integrating a machine learning algorithm was implemented for recognizing flexion/extension/rest gestures of user fingers. The algorithm performances were tested on 9 male subjects (8 able-bodied and 1 subject affected by upper-limb amelia), demonstrating high accuracy and fast responses. *This is an open access article under the <u>CC BY-SA</u> license.*

Corresponding Author:

P. Visconti,
Department of Innovation Engineering, University of Salento,
Road to Monteroni, Ecotekne Campus, Lecce, 73100, Italy.
Email: paolo.visconti@unisalento.it

1. INTRODUCTION

This manuscript describes the realization and testing of a transradial prosthesis addressed to upper-limb amputees managed by a wireless armband able to detect, through electrodes placed on the skin, the myoelectric signals produced by muscle contractions. Each movement is associated to the activation of a determined muscular group; thus signals' processing makes possible the prosthetic hand's movements. The prosthesis can actuate 15 degrees of freedom, three for each finger, with only one motor (instead of five-six motors normally used) and uses two servo-motors for wrist flexion/extension and prono-supination; the torque is automatically distributed among fingers which adapt to the grasped object's form, so obtaining an optimal grasp [1, 2]. This allows to simplify the control logic and to save on prosthesis cost, weight and dimensions, besides obtaining low power consumption and reduced noise. Myo armband is a wearable device by Thalmic Labs Inc. with 8 EMG electrodes, a 9-axis IMU (3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer) and BLE transmission module in order to send data to HM11 BLE module connected to control Arduino board for driving and managing actuators so allowing hand and wrist movements. Each prosthesis' fingertip is equipped with temperature and pressure sensors, connected to Arduino board, for providing user a feedback related to the grasped objects, data visualized on a touchscreen display installed into the prosthesis socket. A Raspberry Pi board manages the display and sends EMG and IMU data via Wi-Fi to a dedicate on-cloud platform to be monitored by orthopedic

during user's rehabilitation as well as post-processed by using Wolfram Mathematica.Myo armband is excellent solution for prosthetic applications due to its reliability, ease of use and low cost as shown in Figure 1(a); further fields of use concern medicine, robotics or for remotely controlling flying drones and robot movements[3].

The armband operation is managed by *ARM Cortex M4* processor while data transmission is performed by BLE *NRF51822* chip toward the *HM-11* BLE module mounted on bottom side of the prosthesis control unit. Myo armband is shown disassembled in Figure 1(b)and its control board in Figure 1(c) [4]; two lithium batteries are located into two elements and recharged providing 5V voltage by micro-USB connector.Different research studies were conducted on Myo armband suitability for prosthesis and virtual reality applications; in [5] it was used with the aim to control PeopleBot robot for reaching bio-infected or radioactive environments or for helping disabled people. In [6] Myo armband was used for real time EMG data collection in order to control a robotic wheel-chair based on EMG and electro-encephalography, confirming its suitability instead of expensive professional EMG devices. Other works focused on the improvement of human-machine interaction obtained by using devices similar to Myo armband and pattern recognition's algorithms [7-10]. Figure 2 shows two robotic arms controlled by user wearing the Myo armband; specifically by opening/closing his hand, user controls robotic hand's opening/closing (a), while in Figure 2(b) by two Myo bracelets, user's arm orientation and gestures allow to pick up and manipulate objects.
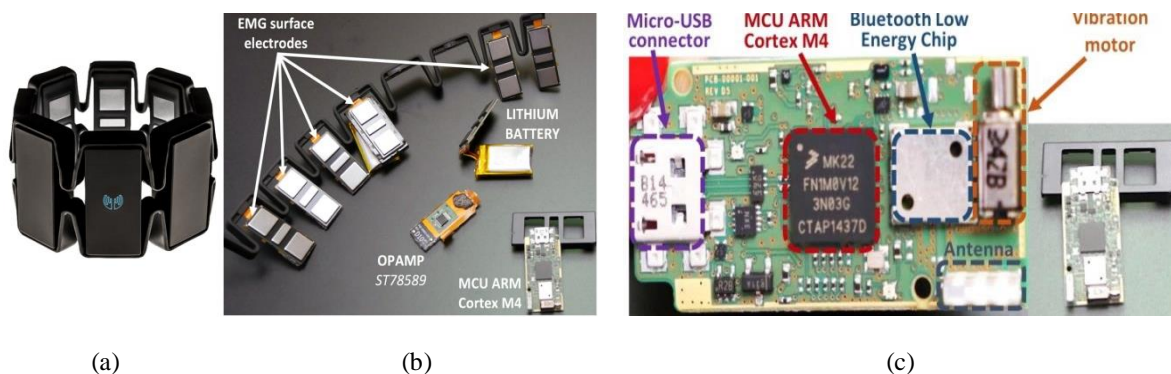

(a)   (b)   (c)

Figure 1. (a) Myo armband, (b) Its 8 EMG electrodes with other components and *ST78589 OA* located into each armband element for amplifying signals provided by electrodes, (c) Control board into armband

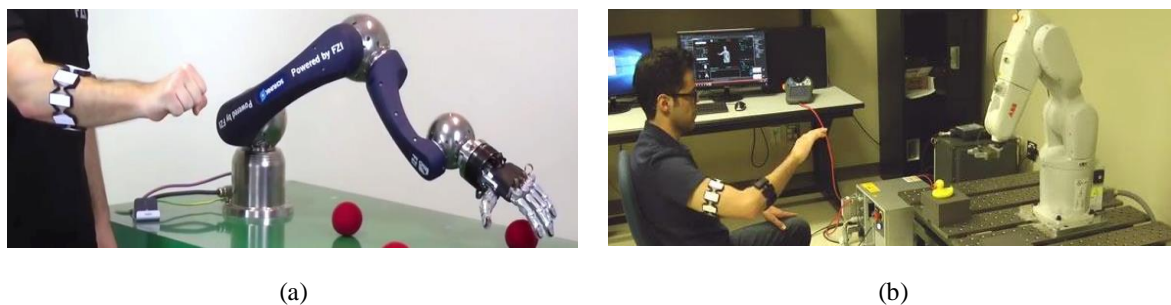(a)   (b)

Figure 2. (a) A robotic arm controlled by user wearing Myo armband (source: forschungszentruminformatik), (b) Industrial robotic arm controlled by two myo bracelets

2.   RESEARCH METHOD
2.1. Devices and circuital solutions to drive and control prosthesis motion

The different electronic modules, realized and used in the prosthetic limb, are detailed below. The electronic section for motor driving and prosthesis control is located in the upper side of the prosthetic hand, while pressure and temperature sensors are placed into the fingertips. A touchscreen display, positioned on the forearm, is employed to manage prosthesis and visualize data or error messages. Figure 3(a) shows a functional scheme with employed modules connected to the driving and control board. The Myo armband is arranged on the upper side of user forearm; it acquires myoelectric signals by EMG electrodes and sends data to the control unit. The Arduino Micro board was chosen for its features suitable for this specific

application: it is light and small so easily insertable into prosthesis, enables the simple performing of all the tasks needed for the right operation of the prosthesis and its power consumption is low, thus extending battery life [11].Figure 3(b) shows the connection scheme of the Arduino board with employed modules, sensors and actuators; the communication with HM-11 BLE module is implemented through serial interface with TX/RX pins, the analog *FSR400* pressure and LM35 temperature sensors are connected to analoginputs of embedded ADC with a capacitor referred to ground for noise and electromagnetic interferences rejection [12].
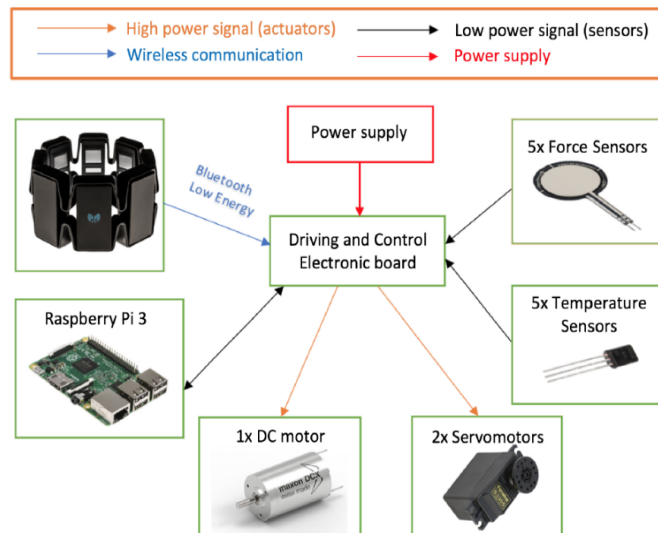

(a)
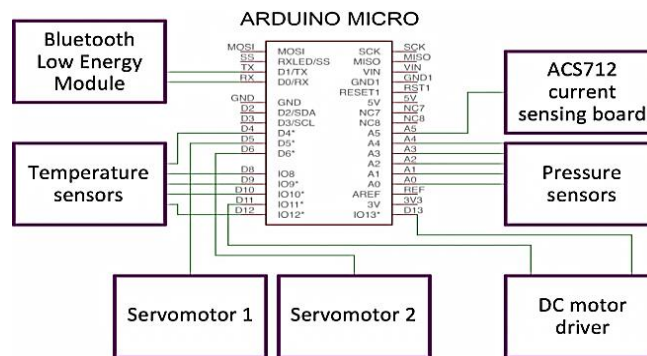
(b)

Figure 3. (a) Used modules: Myo armband, raspberry board, DC motor, two servomotors, temperature and force sensors, all interfaced with Arduino, (b) Connections between Arduino and modules, sensors and actuators

The DC and servo motors are controlled by dedicated output pins for PWM signals generation in order to manage and adjust motors' actuation. ACS712 analog current sensor is employed for measuring the current absorbed by DC motor to adjust the force applied by prosthesis' fingers on the grasped objects.The LM35 temperature sensor are installed in each fingertip and the measured values are reported on the display which represents a feedback on grasped objects' temperature besides over limit temperature warning given by vibration that would not be otherwise detectable by user. In addition, a pressure resistive sensor equips each fingertip (FSR 40 from Interlink Electronics) to acquire pressure that the prosthetics' fingers exert on grasped object; adding these data to current absorbed from DC motor, it is possible to grip any object typology without damaging it. The ACS712 is a Hall effect current sensor and is used to sense the current flowing into its integrated path, as shown in Figure 4(a). The used breakout board is composed of two ICs: the ACS712 and OPA344 to amplify voltage signal provided by ACS712 sensor Figure 4(b).

Temperature and pressure sensors give information on the exerted pressure and grasped object's temperature whereas current's control allows to avoid crushing the tea can or to grip a pen Figure 5(a) and (b) [13]. The *Maxon DCX-19-S* motor, employed to actuate fingers, is interfaced with a controller, provided of power supply unit to feed DC and servo-motors (5-6V), and to *SN754410NE* DC motor driver. Figure 5(c) shows the connection scheme of Arduino board with ACS712 current sensor and DC motor driver

Two *Futaba S3305* servomotors actuates the wrist, one for hand prono-supination and the second for performing wrist flexion/extension; they are powered and driven from the control board by 6V DC supply voltage and two 50Hz PWM signals. A PID regulator, by means of the angular position detected through a rotary encoder, holds the angular position of the motor shaft constant, thus leading to zero the position error even in presence of applied forces (within motors' power and torque limits). The graphs reported in Figure 6 show the rotation speed and absorbed current as functions of exerted torque. The DC motor, used for fingers movement, requires a supply voltage of 6V; the *LT1086* linear voltage regulator receives 7.2 V from lithium-Ion battery and provides the required voltage. In order to obtain a stabilized voltage and reduced noise, high-value electrolytic capacitors were added able to avoid voltage drops during the maximum current absorption. Considering worst case (peak-torque condition), the current absorbed by DC motor and each servomotor is respectively 1.2A and 1.07A; since the control logic allows actuating motors one at a time, the maximum current to be guaranteed is 1.2A and LT1086 satisfies this requirement. Since input and output current are equal ($I_{IN}=I_{OUT}$), the LT1086 efficiency was estimated: $= \dfrac{P_{load/}}{P_{tot}} = \dfrac{I_{out}*V_{out}}{I_{in}*V_{in}} = \dfrac{V_{out}}{V_{in}} = \dfrac{6}{7.2} \cong 0.833 = 83.3\%$.
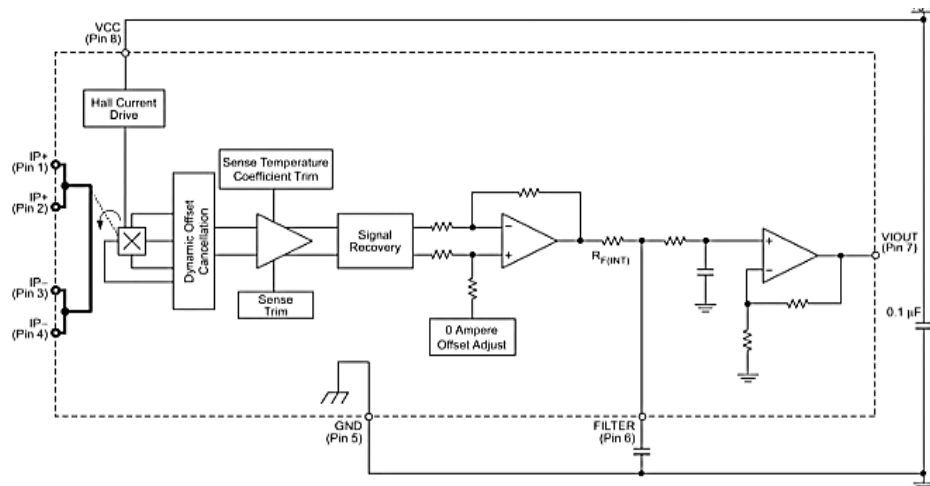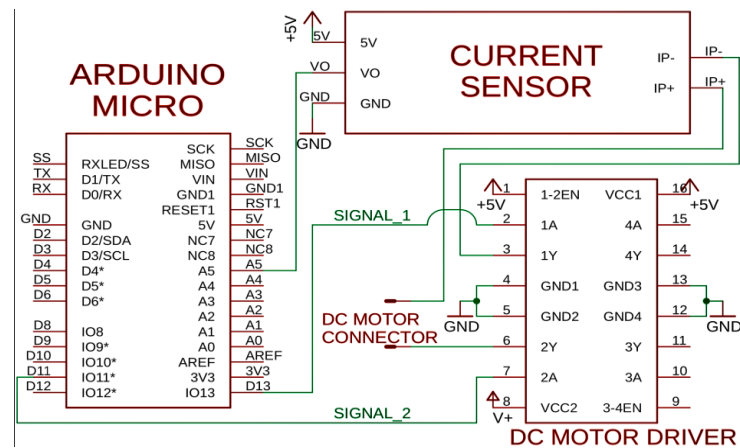

(a)
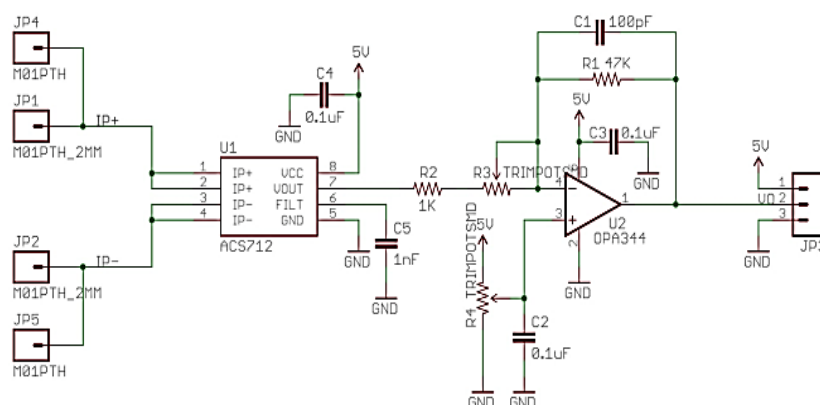
(b)

Figure 4. (a) Inner scheme of ACS712 IC current sensor, (b) Circuital scheme of breakout board
with ACS712 sensor and OPA344 IC that amplifies voltage supplied by ACS712 sensor

(a)

(b)

(c)

Figure 5. (a-b) The control on DC motor absorbed current allows to manage applied force in object grasping, (c) Circuital connections between Arduino board, ACS712 current sensor and SN754410NE motor driver
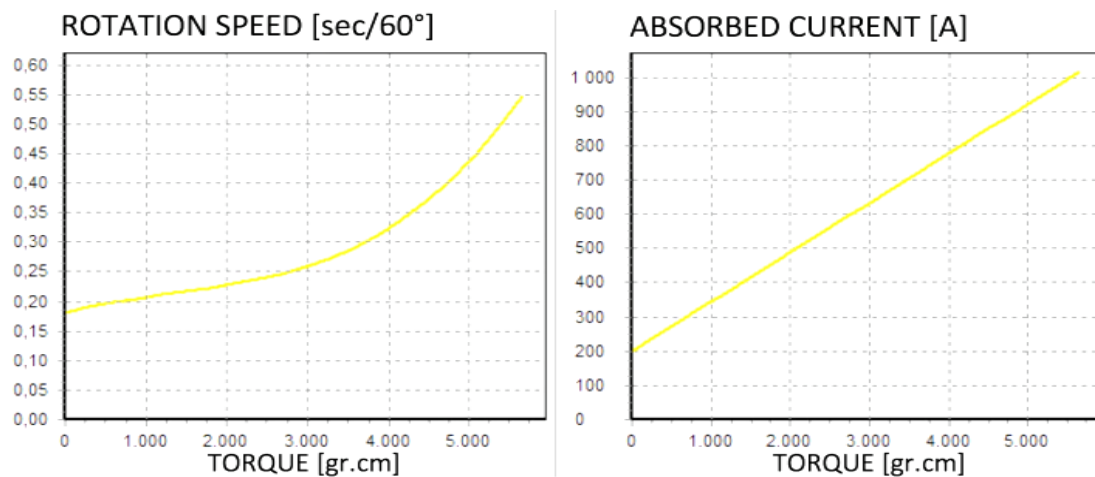
Figure 6. Graphs related to servo-motors rotation speed and absorbed current as functions of exerted torque

Since difference between input and output voltages is low, the LT1086 power dissipation results: $P_{diss} = I_{out} * (V_{in} - V_{out}) = 1.2A * 1.2V = 1.44W$ in case of 1.2A peak current. The junction thermal resistance is 50°C/W (as reported in LT1086 datasheet); therefore, known the power dissipation and working temperature (25°C), the junction temperature results: $T_j = (P_{diss} * R_t) + T_a = (50°C/W * 1.44W) + 25°C = 97°C$.

For correct functioning, the junction temperature should not exceed 150°C, thus a heatsink was not required, but for ensuring a low working temperature and thus long device life-time, a small heatsink was installed.

In order to give user the possibility to control, view and also send via Wi-Fi to the cloud platform the device's usage data, a Raspberry Pi board was placed into an internal compartment [13], it manages a 3.5" LCD touch-screen display to visualize battery level and sensors data, to set prosthesis preferences and to manage Wi-Fi connectivity. The Raspberry board is able to provide remote access to several functionalities, such as technical assistance for software update regarding the operating system or control board firmware; in addition the orthopedic staff can follow the amputee remotely, giving him assistance in real time. Raspberry board requires a supply voltage of 5V, while the used *OttoBock* battery provides 7.2 V; hence a step-down switching voltage regulator is used Figure 7(a) able to provide high current value (up to 2A as required from Raspberry board) with 80% efficiency. It employs LM2596 IC provided with a mosfet which converts the input voltage in a kHz-frequency square wave, then filtered by a low pass filter. Two electrolytic capacitors (C_1, C_2) at circuit's input and output are used for voltage stabilization; V_{R1} trimmer allows to adjust the output voltage while the *schottky* diode (D_1) acts as protection. A disadvantage of switching regulators is the output ripple, critical for supplying analog sensors but negligible in this case (with digital devices).

The touch-screen display connected to Raspberry board is mounted into prosthesis socket; in Figure 7(b) an alert message related to a high temperature detected from the index fingertip is reported. At the screen bottom, the five preset hand poses are shown; when Myo armband recognizes one of these, the corresponding circle lights up, giving user a visual feedback of the performed action, useful during the training phase. On the right side, in addition to information on Wi-Fi/Bluetooth connectivity and battery level, a customizable area is left to be personalized by user. The Arduino board communicates with Raspberry by RS232 serial interface sending data related to EMG electrodes (with 200Hz frequency), to the IMU and to the five pressure (with 50Hz frequency) and temperature sensors (with 1Hz rate); all these data are stored on the SD card of Raspberry board. An executable script was developed by using *Wolfram Mathematica* software installed on Raspberry board to visualize in real-time, after filtering of corrupted data, the forearm muscular activity and signals provided by IMU [14]. Then, processed data are sent on the cloud platform to be remotely monitored by orthopedic [15, 16]. Figure 8(a) reports used boards and devices highlighting connections between them.

Battery autonomy is a critical point for portable devices especially for wearable ones as the realized prosthesis, becoming a fundamental aspect for patient quality of life and comfort. A lithium-Ion battery, model 757B20 produced by OttoBock, was used, featured by 900mAh capacity (i.e. 6.5Wh), 3.5h recharge time, provided voltage of 7.2V, 65g weight and operative temperature 0-50°C Figure 8(b). The power consumption of each powered device is reported following: 7.2W for DC motor, 12.8W for two servomotors, 180mW for Arduino board, 33mW for HM-11 BLE module, 1.5mW and 12.5mW respectively for all temperature and pressure sensors. Raspberry board consumption is 4W, touch-screen display 330mW, 1.44W is related to voltage regulator for motor power supply and 0.87W for switching regulator. It is possible to calculate a weighted average of power consumption (P_{avg}) making the sum of all consumptions multiplied by usage-time percentages obtaining about 1.4W; since battery energy is 6.5Wh, its autonomy results ~4.6 hour.

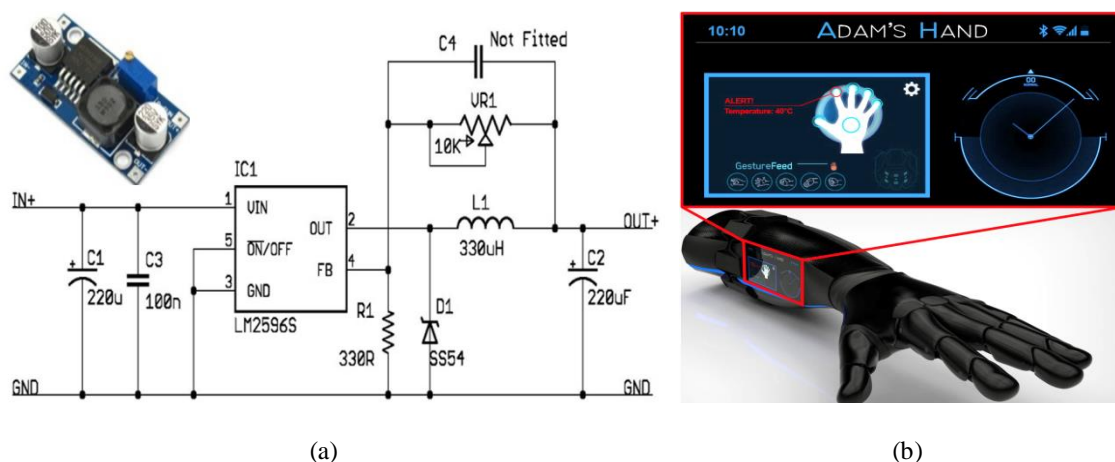(a)   (b)

Figure 7. (a) DC/DC step-down converter used to feed Raspberry Pi board (top left) and the related circuital scheme, (b)touchscreen display reporting sensors data and feedback messages for user

(a)   (b)
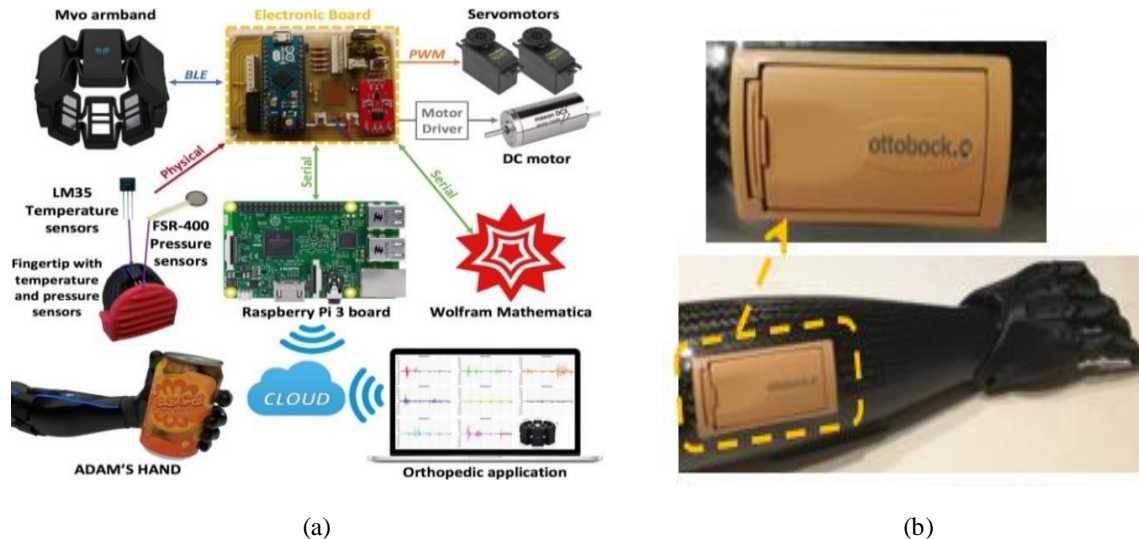
Figure 8. (a) Employed modules and actuators: user rehabilitation is remotely monitored by the orthopedic, (b) Realized prosthesis with the lithium-Ion battery in the zoomed box and its housing into prosthesis socket

2.2. Developed Firmware for data acquisition and prosthesis control

In this section, the methods developed to get correct functioning of prosthetic hand are explained; the firmware employs a library (*MyoBridge*) which integrates numerous methods (*setEMGDataCallBack* for EMG data, *setIMUMotionCallBack* and *setIMUDataCallBack* for IMU data and *setPoseEventCallBack* for poses data) that allow data exchange between Myo armband and Arduino board. The implemented firmware can be divided in three groups: the first one comprises functions that manage poses data (*handlePoseData()*), IMU (*handleIMUData()*) and EMG data (*handleEMGData()*) (as shown in Figure 9(a)); the second group is related to functions that manage pressure (*updatePressure()*) and temperature (*update Temperature()*) sensors data; the last group comprises functions for motors managing (*updateServo1()* and *updateServo2()* for servomotors and *updateDCMotor()* for DC motor) [17]. A further function (*ResetMotors()*) resets the motors to their base position (hand rest). Following the developed functions are analyzed.

− *handlePoseData()* function: the prosthesis movements are managed by exploiting five predefined poses (*spread fingers*, *make fist*, *wrist flexion*, *wrist extension*, *fingers touch*) recognized by the armband thanks to data provided by EMG electrodes and 9-axis IMU [18, 19]. When the armband recognizes a preset pose, a *callback* event is transmitted to Arduino microcontroller and by the *setPoseEventCall Back()* function, performed pose is identified. The *handlePoseData()* function converts *MyoPoseData* into a *MyoPose object*, for managing the wrist extension/flexion and to control hand opening/closing. A further *if* structure, related to *fingers touch* pose, resets the servomotors and DC motor positions, obtaining hand rest position.

− *handleIMUData()* function: the 9-axis IMU provides data related to arm prono-supination, to properly drive the servomotor employed for wrist rotation. When the armband has to provide these data, an interrupt is generated and recognized by Arduino that executes *handleIMUData()* function where the magnetometer values expressed in quaternions are converted into roll, pitch and yaw angles. Also, in the same function, the accelerometer and gyroscope data are stored and transmitted through serial connection to Raspberry.

− *handleEMGData()* function: it is recalled when data related to EMG electrodes are received from Arduino board and then sent to Raspberry board for further processing.

The functions to process analog signals from LM35 temperature and FSR400 pressure sensors are described following. Since Arduino *ADC* has 10bit resolution, values returned by the *analogRead()* function range from 0 to 1023, in order to map these values to the desired interval (e.g 0-500 °C for temperature), the Arduino library's *map()* function was used; since temperature varies slowly, a 1 Hz sampling rate was adopted ensuring both low processing load and power consumption. The *updateTemperature()* function processes LM35 sensors data in order to warn user in case of high temperatures by means of vibration of Myo armband.

− *updatePressure()* function: it processes data of pressure FSR400 resistive sensors with a 50Hz sampling rate of the analog signals. The voltage values obtained from the *map()* function are converted in resistance

by the voltage divider formula. Then, by using the pressure/resistance FSR400 characteristic, the force or pressure exerted by the bionic hand is obtained; Figure 9(b) shows the function's flowchart. Thus the pressure values are used to control the gripping force and to detect any slipping of the grasped object.

Functions for motors managing are described below; the functional diagram is shown in Figure 9(c).

− *updateServo1()* function: it is used to control wrist flexion/extension and is recalled if a contraction of the wrist flexors/extensors muscles, corresponding to *Wave left* or *Wave Right* poses respectively, is recognized by *handlePoseData()* function. The function receives as input a determined angular value, depending on the received pose, and recalls a *Servo* library method providing the *PWM* signal for driving the servomotor.

− *updateServo2()* function: it is used to drive the second servomotor needed to actuate the wrist rotation controlled by the yaw angle got by *handleIMUData()* function. To avoid motor vibrations due to high magnetometer sensitivity, it is executed after set time interval providing an average value of the yaw angle.

− *updateDCMotor()* function: it is recalled whenever the *Make Fist* or *Spread fingers* poses are recognized, so that the motor rotates clockwise closing fingers, or counterclockwise to make the fingers open. If the absorbed current or pressure exerted by fingertips exceeds a preset threshold, the motor is switched off.

− *ResetMotors()* function: it is called whenever the *Fingers Touch* pose is detected by the *handlePoseData()* function; in this case, the *three* motors return to their default position bringing the hand to the rest position. Numerous objects with different forms, dimensions and stiffness were correctly gripped and data related to temperature and pressure acquired by sensors integrated into fingertips, shown on the PC terminal.
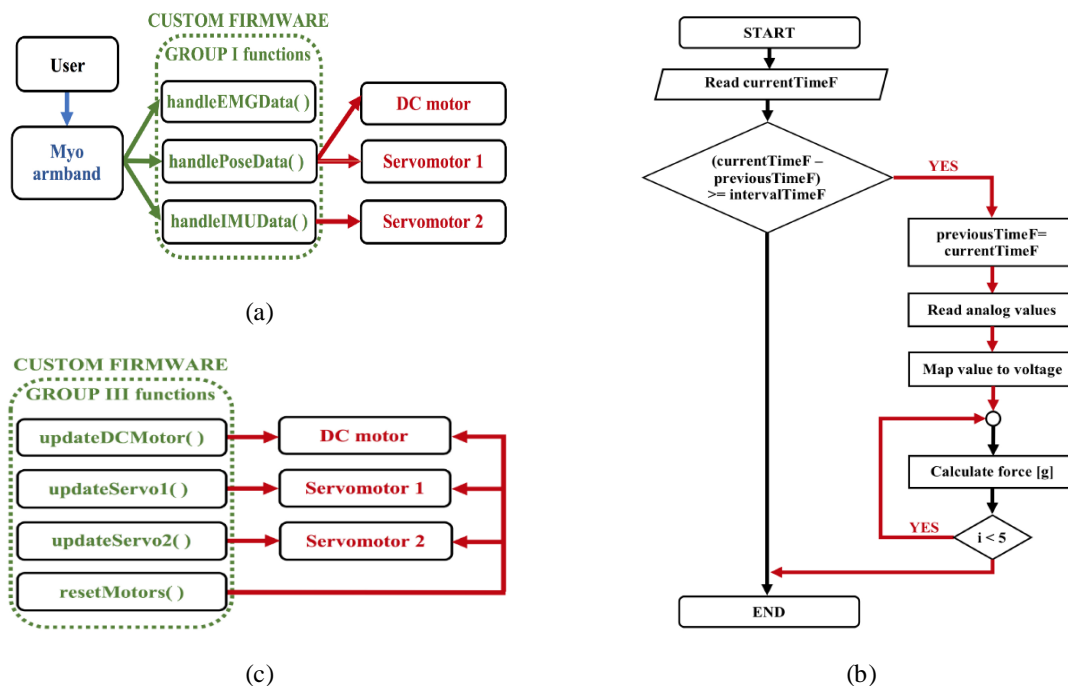

Figure 9. (a) Diagram related to functions implemented to manage the poses, IMU and EMG data, (b) Flowchart of the update Pressure() function, (c) Functional diagram related to motors driving

3. RESULTS AND DISCUSSION
3.1. Real experiments and on subjects

In the carried out experiments, the RAW data from the Myo armband were analyzed and elaborated. The myoelectric control of the prosthetic limb was tested through the hardware and software system, which includes the electronic board, described above, for acquiring the EMG signals from Myo armband and GUI software integrating a machine learning algorithm that enables the recognition of flexion/extension/rest gestures of the user's fingers [20, 21]. The algorithm performances have been tested on 9 male subjects (8 able-bodied and 1 subject affected by congenital upper-limb amelia at the right forearm).

The results obtained by the classification procedure show high accuracy (90.4±2.4%) and fast responses (< 1s for 97.4±2.0%). Consequently, these performances show that the machine learning algorithm is accurate and fast. A GUI software was developed (see Figure 10) for managing the following steps:

- EMG acquisition: the 8 EMG double differential channels are streamed from Myo device to the serial port of PC by means of the electronic board.
- EMG processing: raw signals are band-pass filtered (10-280 Hz, 50 elements FIR digital filter), full-wave rectified and low-pass filtered (1 Hz, 50 elements FIR digital filter) to obtain the signal envelopes.
- Training phase: each subject was asked to perform three different gestures with hand fingers (10 seconds of flexion, 10 seconds of extension, and 10 seconds of rest), while EMG signals were acquired, processed and windowed into 500 ms epochs which were given as input to the machine learning algorithm, in order to obtain the proper weights for flexion/extension/rest gesture classification.
- Gesture recognition: after performing the training phase, EMG signals were acquired, processed and windowed into 500 ms epochs in real time. For each epoch, the machine learning algorithm classifies the hand fingers gesture as flexion, extension or rest [22-24].

For evaluating the algorithm performances, N=8 able-bodied male participants (mean age of 30.30 ± 4.96 years, mean height of 177.75 ± 5.55 cm, mean weight of 74.86 ± 11.31 kg, mean BMI of 23.70 ± 3.42 kg/m2) were recruited. Myo armband was worn by each subject and, after a training phase, they performed a series of 10 flexions and extensions in random manner; for each gesture, the GUI software registered EMG signals and gesture recognition was considered as correctly detected if the algorithm recognizes it within 1sec since the gesture execution. Afterwards, the same test was performed by a subject affected by a congenital upper-limb amelia at the right forearm (male, 27.23 years old, 195 cm, 82 kg, BMI of 21.26 kg/m^2) [24, 25]. The accuracy of gesture recognition was calculated as the percentage of gesture correctly classified. Finally, the results obtained by the classification procedure are shown in the following Table 1.
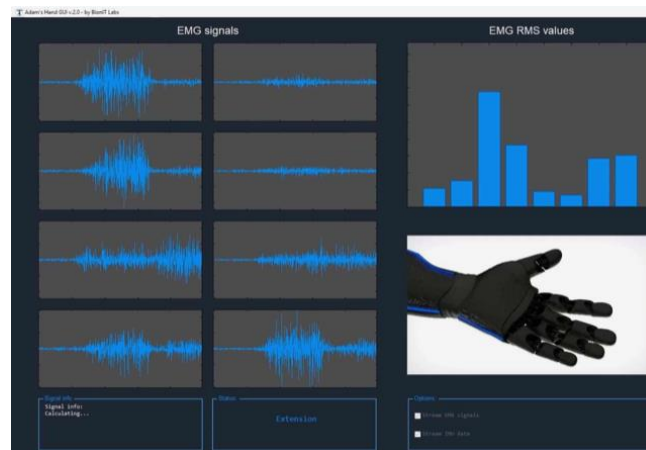

Figure 10. GUI software for EMG recording, processing and gesture recognition of fingers flexion, extension or rest. EMG signals are shown in real time and gesture is classified each 500 ms with no-detectable delay

Table 1. Performances of gesture recognition algorithm to detect flexion, extension and rest of hand fingers by EMG signals analysis (* *Subject 9* affected by congenital upper-limb amelia at the right forearm)

	Recognition accuracy (%)				Time response < 1 s (%)			
	Flex	Ext	Rest	Average	Flex	Ext	Rest	Average
Subject 1	90	100	90	**93,3**	100	100	100	**100,0**
Subject 2	90	80	100	**90,0**	90	100	100	**96,7**
Subject 3	80	90	90	**86,7**	100	90	100	**96,7**
Subject 4	100	90	100	**96,7**	100	90	100	**96,7**
Subject 5	90	80	90	**86,7**	90	100	100	**96,7**
Subject 6	70	90	90	**83,3**	100	100	100	**100,0**
Subject 7	90	100	80	**90,0**	90	100	100	**96,7**
Subject 8	90	90	100	**93,3**	90	100	90	**93,3**
Subject 9*	90	100	90	**93,3**	100	90	100	**96,7**
Mean	**87,8**	**91,1**	**92,2**	**90,4**	**95,6**	**96,7**	**98,9**	**97,0**
St. dev.	**8,3**	**7,8**	**6,7**	**4,2**	**5,3**	**5,0**	**3,3**	**2,0**

4.   CONCLUSION

In this work, the electronic system of a prosthetic limb controlled by wireless myoelectric armband was realized. The prosthesis provides to the user feedback from the surroundings such as objects temperature and grip force applied by fingers on grabbed objects; it was designed in modular way to be adapt to different amputees types, thus satisfying a larger users category. The usage of a single motor to handle the whole hand allows to simplify the prosthesis control as well as to save on its cost and weight. EMG and IMU data are monitored remotely by the orthopedic that follows user rehabilitation by the Wi-Fi connectivity provided by Raspberry. An algorithm based on machine learning was implemented and real experiments conducted: the recognition of hand gestures was tested on 9 subjects (one affected by congenital upper-limb amelia). Results obtained by the classification procedure show high accuracy and fast responses also for amputee subject, meaning that the algorithm works fine also on subject with small not-trained muscles. Future developments will consist in improving the control algorithm by exploiting EMG signal intensity, controlled by user after training procedure, to implement a closed-loop control to be able to apply desired force to grasped objects.

REFERENCES

[1]   G. Zappatore, et al., "Adam's Hand: An underactuated robotic end-effector," in *Advances in Italian Mechanism Science*: *Mechanisms and Machine Science,* vol. 47, pp. 239-246, 2016.

[2]   G. Zappatore, et al., "Analysis of a highly underactuated robotic hand," in *International Journal of Mechanics and Control*, vol. 18, pp. 17-23, Dec. 2017.

[3]   P. Visconti, et al., "Technical features and functionalities of Myo armband: An overview on related literature and advanced applications of myoelectric bracelets mainly focused on arm prostheses," in *International Journal on Smart Sensing and Intelligent Systems*, pp. 1-25, Feb. 2018.

[4]   F. Gaetani, et al., "Design of an Arduino-based platform interfaced by Bluetooth low energy with Myo armband for controlling an under-actuated transradial prosthesis," in *2018 International Conference on IC Design & Technology (ICICDT)*, pp.185-188, 2018.

[5]   G. D. Morais, et al., "Application of Myoarmband system to control a robot interface," in *Proc. of 9th Int. Conf. on Biomedical Engineering Systems and Technologies,* vol. 4, pp. 227-231, 2016.

[6]   G. Kucukyildiz, et al., "Design and implementation of a multi sensor based brain computer interface for a robotic wheelchair," in *Journal of Intelligent and Robotic System*, vol. 87, pp. 247-263, Jan. 2017.

[7]   M. Havlikova, et al., "A Man as the Regulator in Man-Machine Systems," in *Advances in Electrical and Electronic Engineering*, vol. 12, pp. 469-475, Dec. 2014.

[8]   A.V. Le, et al., "Human detection system by fusing depth map-based method and convolutional neural network-based method," in *Advances in Electrical and Electronic Engineering*, vol. 15, pp 648-656, Nov. 2017.

[9]   M. M.-. Vidovic, H. Hwang, S. Amsüss, J. M. Hahne, D. Farina and K. Müller, "Improving the robustness of myoelectric pattern recognition for upper limb prostheses by covariate shift adaptation," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 9, pp. 961-970, Sept. 2016.

[10]   Y. Fang, N. Hettiarachchi, D. Zhou and H. Liu, "Multi-modal sensing techniques for interfacing hand prostheses: A review," in *IEEE Sensors Journal*, vol. 15, no. 11, pp. 6065-6076, Nov. 2015.

[11]   P. Visconti, et al., "Arduino-based solution for in-car-abandoned infants' controlling remotely managed by smartphone application," in *Journal of Communications Software and Systems*, vol. 15, no. 2, pp. 89-100, Jun. 2019.

[12]   P. Visconti, A. Lay-Ekuakille, P. Primiceri, G. Ciccarese and R. de Fazio, "Hardware design and software development for a white LED-Based experimental spectrophotometer managed by a PIC-Based control system," in *IEEE Sensors Journal*, vol. 17, no. 8, pp. 2507-2515, 15 April15, 2017.

[13]   P. Visconti, et al., "Design and testing of electronic control system based on STM X-Nucleo board for detection and wireless transmission of sensors data applied to a single-seat Formula SAE car," in *International Journal of Electronics and Telecommunications*, vol. 65, no. 4, pp. 671-678, Jun. 2019.

[14]   N.A. Nayan, et al., "Development of respiratory rate estimation technique using electrocardiogram and photo-plethysmogram for continuous health monitoring," in *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol.7, no.3, pp. 487-494, Sep. 2018.

[15]   S.A. Hameed, et al., "Application of mobile cloud computing in emergency health care," in *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 8, no. 3, pp. 1088-1095, Sep. 2019.

[16]   J. Too, et al., "Featureless EMG pattern recognition based on convolutional neural network," in *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14 no. 3, pp. 1291-1297, June 2019.

[17]   P. Primiceri, et al., "Hardware and software solution developed in ARM mbed environment for driving and controlling DC brushless motors based on ST X-NUCLEO development boards," *in Int. Journal on Smart Sensing and Intelligent Systems,* vol. 9, no. 3, pp. 1534-1562, Sep. 2016.

[18]   T. Triwiyanto, et al., "Embedded system for upper-limb exoskeleton based on electro-myography control," in *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 17 no. 6, pp. 2992-3002, Dec. 2019.

[19]   J. Too, et al., "Application of gabor transform in the classification of myoelectric signal," *TELKOMNIKA*

(Telecommunication, Computing, Electronics and Control), vol. 17 no. 2, pp. 873-881, April 2019.

[20] M. R. Mohamad Ismail, et al., "Hand motion pattern recognition analysis of forearm muscle using MMG signals," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 8, pp. 533-540, Aug. 2019.

[21] A. M. NajlaIlyana, et al., "Analysis of surface electromyography for hand gesture classification," in *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 15, no. 3, pp. 1366-1373, Sept. 2019.

[22] M. Azlan Abu, et al., "Classification of EMG signal for multiple hand gestures based on neural network," in*Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17 no. 1, pp. 256-263, Jan. 2020.

[23] G.W. Nie, et al, "Deep stair walking detection using wearable inertial sensor via long short-term memory network," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, pp. 238- 246, Feb. 2020.

[24] N. Abdul Malik, et al., "Investigation of lower limb's muscles activity during performance of salat between two age groups," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14,no.2, pp. 608-617, May 2019.

[25] J. Too, et al., "A Detail Study of Wavelet Families for EMG Pattern Recognition,"in *International Journal of Electrical,* vol. 8, no. 6, pp. 4221-4229, Dec. 2018.