

# Achieving Assistive Technology Systems Based on IoT Devices in Cloud Computing

Davide Mulfari, Antonio Celesti, Maria Fazio, Massimo Villari, Antonio Puliafito<sup>1</sup>  
e-mail: {dmulfari,acelesti,mfazio,mvillari,apulafito}@unime.it

<sup>1</sup>DICIEAMA, Faculty of Engineering, University of Messina

## Abstract

This paper discusses how programmable embedded IoT systems may be a challenge for the development of Assistive Technologies (ATs), with particular reference to the Human-Computer Interaction (HCI). Nowadays, users with disabilities access computer-based devices by using tailored AT solutions equipped with specialized software components closely tied to the physical device. We propose to decouple the AT tools from the computers and/or ICT (Information and Communication Technology) systems. The main advantage of such a solution is the high flexibility for users with disabilities in using their personal AT equipments to access/control different types of computers and ICT systems. To avoid manual driver installation and software configuration on each available computer, the IoT component makes use of Cloud technology and services to customize the AT environment according to the user's needs and preferences.

**Keywords:** Assistive Technology, Embedded Systems, Cloud Computing, Internet of Things, Human Computer Interaction, Users with disabilities.

Received on 01 December 2014, accepted on 20 January 2015, published on 23 February 2015

Copyright © 2015 Davide Mulfari, *et al.*, licensed to ICST. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/cs.1.1.e4

## 1. Introduction

People with disabilities interact with computing environments by using tailored Assistive Technology [1] (AT) equipments. In the ICT (Information and Communication Technology) field, these tools include hardware and software components to perform I/O (Input Output) actions. For instance, users with motor disabilities can rely on special joysticks, keyboards, or sensors as alternative pointing/input devices, or they can use universal access software options enclosed in the computer's operating system (OS) in order to simulate mouse movements by pushing specific keys on the keyboard.

In general, different types of AT equipments include both software and hardware tools. For instance, eye tracking tools use one or more cameras to track the user's pupils using visible or infrared light and they also require a tailored computer application in order to process data coming from cameras. In other contexts, AT tools consist of specialized pieces of software, such as screen reader and screen magnification

computer programs. These considerations may be easily applied to any AT solutions based on sensors [2] (e.g., Brain Computer Interface (BCI) techniques [3], Electromyography (EMG)[4] and Inertial Measurement Unit (IMU) systems [5]), which rely on a given hardware platform and a dedicated software in order to work properly.

In this paper we consider a smart environment, where users interact with several computing devices and platforms. To this aim, he/she has to adapt each device to his/her needs and preferences. Thus, the concrete usage of any AT tool requires a specific configuration process in order to meet the end user's needs. This task has to be performed by an AT expert, who works close to the person with a disability in order to properly adapt the computing environment to its user. The configuration task has to be carried out on any used computer, and it can become a complicated task to accomplish in public sites, such as Internet point or laboratory, where the user doesn't have full access to the system to install software and drivers and to properly configure the ICT system. A much more difficult situation occurs every time a person with a disability needs to interact with a computing

\*Corresponding author. Email: [mfazio@unime.it](mailto:mfazio@unime.it)

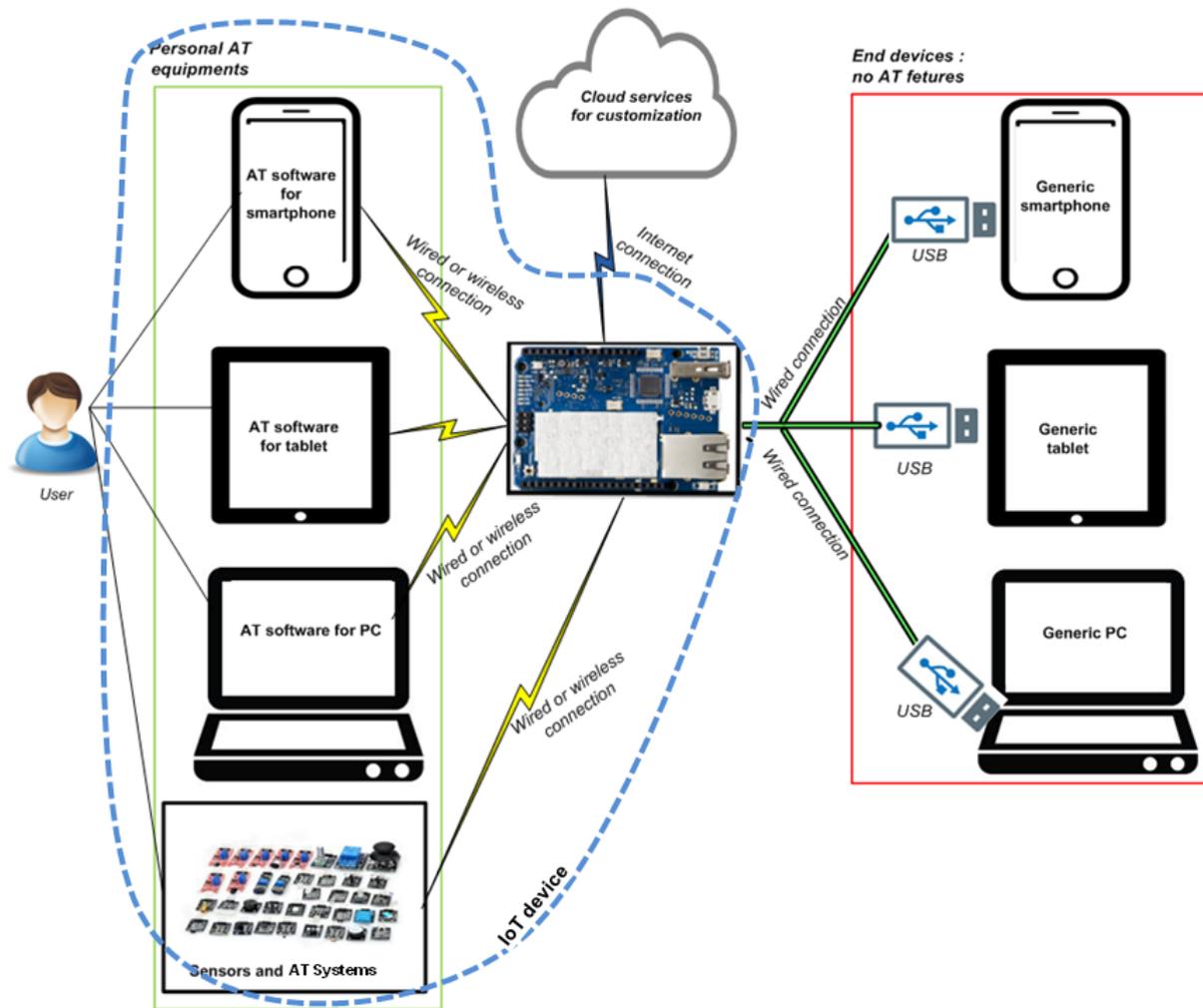


Figure 1. Reference scenario for the proposed solution.

environment that does not support his/her personal assistive solutions.

This paper aims to address the aforementioned issues. Specifically, we propose the usage of embedded systems to interface sensors and existing AT equipments to whichever computer-based devices, as shown in Figure 1. Our solution allows a person with a disability to use his/her own personalized AT tool everywhere for interacting with many computer-based devices. Indeed, it decouples the AT components from the devices on which the user has to operate. Specifically, the embedded system becomes a smart input device working on any computing device (e.g., traditional personal computers (PCs), tablets, smartphones) supporting a mouse and/or a keyboard compatible with the HID (human interface device) standard protocol. The usage of such a protocol is a key feature because it allows all commonly-used operating systems (Microsoft Windows, Mac OS X, Linux, Android) to recognize the hardware peripheral without any specialized driver.

The connection between an external assistive tool, (i.e., a software running on a particular computing platform) and the embedded device requires dedicated pieces of software interfaces called plugins. We exploit Cloud capabilities to store the plugins, that will be deployed on demand into the Linux operating system running on the embedded device. This feature allows an AT expert to remotely support a user with a disability by adjusting the software environment running on the embedded system used by the person to access to a computer

Summarizing, the main contribution of this work is giving a person with disabilities access to multiple ICT systems:

- reducing dependence on AT experts;
- reducing dependence on location;
- giving access to information and communication systems not equipped for interacting with AT devices.

The rest of the paper is organized as follows. Section 2 contains related works. In Section 3, we describe the whole architecture for interconnecting users to ICT

systems through the IoT device. Section 4 presents a prototype of the proposed system. Finally, conclusions are drawn in Section 5.

## 2. Related works

Nowadays, some works on the usage of embedded systems and IOT microcontroller boards in order to arrange HCI (human computer interface) systems for users with disabilities are available in literature.

In [6], the authors discuss the development of an alternative computer input system that allows a person with quadriplegia to move a computer's cursor and activate left and right click button inputs. This solution employs a head mounted Inertial Measurement Unit (IMU) with 9 DoF (Degrees of Freedom) to track head movements and correlate these motions to computer cursor movements. A Sip-Puff transducer detects the pressure of the air blown through to a vinyl tube as analog voltages, which are then interpreted over time to trigger left and right click events. Finally, an Arduino Due board is used to interpret and process these inputs and send mouse commands to the PC via a USB connection.

Another kind of HCI equipment [7] is an augmentative communication system allowing people with severe motor disabilities to use a computer. Specifically, an interface adapted to detect eye winks has been implemented and the system uses these signals, after the processing, to control a mouse or a virtual keyboard. This proposed equipment consists of three primary components: the interface, the processing system based on an Arduino Leonardo hardware platform and the dedicated software installed on the user's computer.

Novel input and output devices designed for power wheelchair users are presented in [8]: these peripherals are mainly based on the Makey Makey microcontroller board, which is an Arduino based device for making tangible user interfaces [9].

The major benefit of such a platform is that it implements the HID protocol, which allows it to send keyboard and mouse events to a computer without installing drivers or other pieces of software. The user can connect everyday objects and natural materials to Makey Makey so as to create a tangible user interface that controls any software running on a computer. In the AT field, an interesting Makey Makey based application allows developers to build a computer input device composed of a wheelchair headrest on which conductive threads are utilized to incorporate a number of inputs- the number of inputs would vary depending on how severe the user's case of paralysis is, and it can be changed by simply attaching/detaching the leads to the headrest.

## 3. Proposed Architecture

Considering the reference scenario in Figure 1, the end user can use different types of AT equipments (e.g., PCs, smartphones and tablets with AT software, or sensors or specific AT devices). Each of them executes tailored pieces of AT software that have been already customized according to the specific user's needs. Our solution allows us to interface such assistive tool with an embedded system in order to create a general purpose IoT device able to access/control many different ICT system, also the ones that do not support AT features. Currently, the control of ICT systems is based on standard Input/Output peripherals, such as mouse and keyboards. Thus, the IoT device in our scenario has a dedicated micro-controller chip able to appear as a standard USB HID input peripheral. This feature is strategic, since the HID protocol over a USB connection can simulate mouse/keyboard input events without requiring a specialized software on the accessed ICT system.

Furthermore a key component of the proposed embedded system is a dedicated processor able to run a Linux OS. It is responsible for handling wired or wireless (such as bluetooth or wifi) network interfaces in order to process data coming from the user's personal AT equipments.

The interaction between an external AT tool and the embedded platform strongly depends on the specific type of AT tool. To make our solution suitable for different types of user's needs, we included a plugin-based interface able to interact with different AT tools in our platform. Each plugin implements specific functionalities for the particular assistive systems in use, and is deployed on top of the Linux-enabled processor. Also, it generally requires a customization process that takes account of the particular assistive tool and preferences of the disabled person. For example, if the end user adopts his/her personal smartphone as AT tool, the plugin has to retrieve the specialized software libraries able to communicate with the given user's assistive environment and set up the responsivity level of the system according to the user ability.

In general, the aforementioned configuration process is not trivial at all. Our solution leverages the Cloud Computing technology together with the GIT version control to update and deploy software components and settings into the user's embedded system. Software components include a full image of the OS for the IoT device, so we can fully customize the behaviour of the board at its boot time. For instance, this approach enables us to set up and initialize just the suitable plugins in order to meet the end user's personal demands.

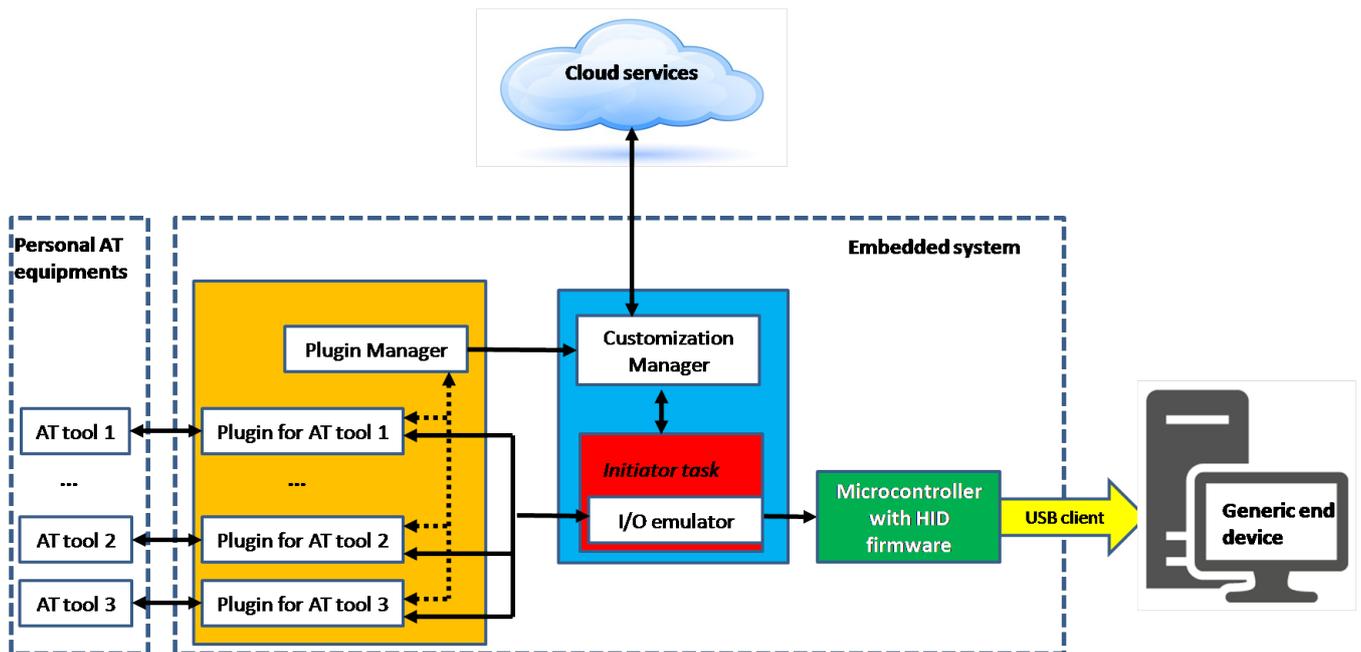


Figure 2. Block diagram of the proposed software infrastructure.

Figure 2 shows the main components of the software architecture deployed on our embedded systems. They are:

- **Plugins for external AT tools.** These are pieces of software interfaces between the user's personal equipments and our system. To achieve this, plugins need to manage network interfaces and services (e.g., a serial port over a bluetooth connection) to acquire data coming from external device. Since an AT equipment can consist of a sensor connected to our board by using GPIO (general-purpose input/output) extensions, plugins have to handle such peripherals.
- **Plugin Manager.** It orchestrates data processed by the plugins and activates them suitably.
- **Customization Manager.** This software module is responsible for getting the embedded system's OS and the software from remote Cloud services.
- **Initiator Task.** It loads the software and the operating system on the IoT device.
- **I/O emulator.** Such a software module handles the communication between the Linux processor and the microcontroller chip by means of a software serial port.
- **HID firmware on a dedicated microcontroller.** This component allows our IoT embedded system to appear as standard HID mouse/keyboard for the connected computer.

#### 4. Prototype Description

The software architecture discussed in Section 3 can be deployed into several IoT boards and systems available on the market. In this paper, we focus our attention on a Linux embedded board equipped with both a processor and a microcontroller chip with HID capabilities. First, an Atmega32u4 chip is required to appear as a native mouse (or keyboard) when the device is attached to a generic computer. More specifically, the same microcontroller executes a custom version of the Arduino bootloader that supports the Mouse / Keyboard libraries. Another main hardware component of our HCI system is a second processor allowing to boot a Linux embedded operating system equipped with a Python runtime. Therefore we use a standard Python high - level programming language to acquire and to process data from external devices and sensors. Additionally, our embedded system needs to manage a network connection to access the Internet, and a USB host port able to interface it with external peripherals.

To fulfill the above requirements, we have developed our prototype on an Arduino Yun board. Arduino Yun is a microcontroller board based on the Atmega32u4 and the Atheros AR9331 processor which supports a Linux distribution based on OpenWrt named Linino. The board has built-in Ethernet and WiFi support, a USB host port, micro-SD card slot, 20 digital input/output pins and a micro USB connection for connecting a computer. Moreover, differently from other Arduino boards (e.g., Leonardo, Fio, Nano,...), Arduino Yun is able to communicate through the Linux distribution

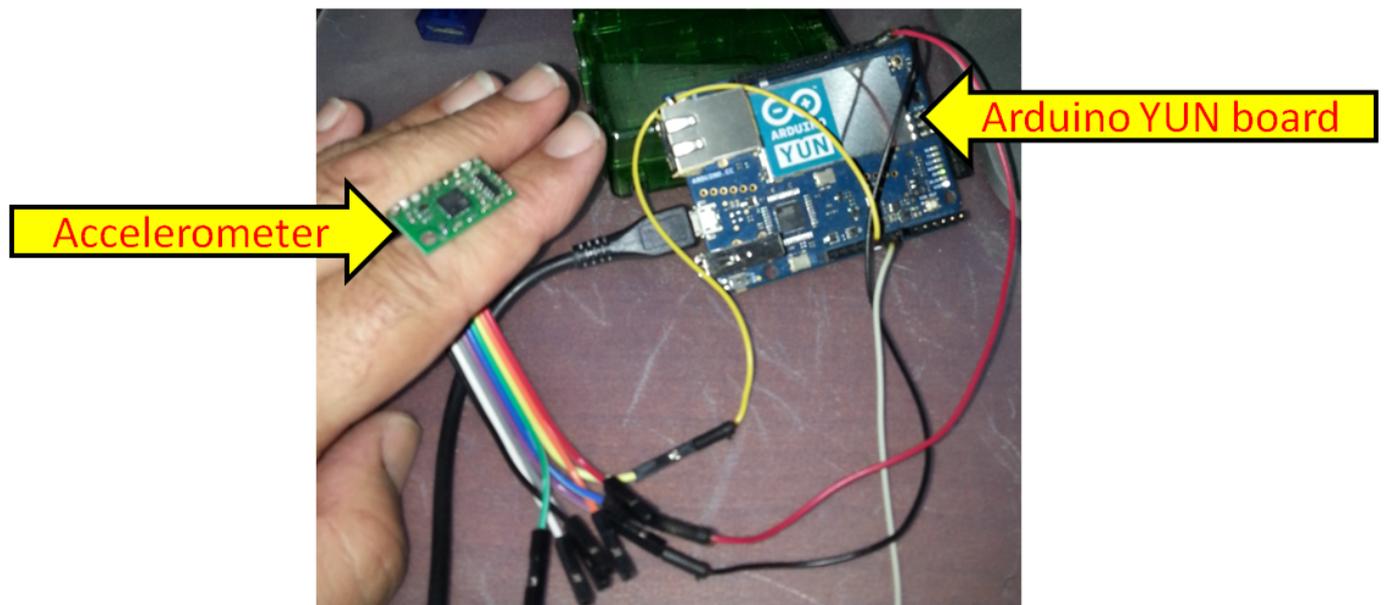


Figure 3. Arduino Yun and an accelerometer-based sensor node placed on a finger.

onboard, offering a powerful networked computer with the ease of Arduino. In addition to basic Linux commands (such as cURL), the Atheros processor is able to execute shell and Python scripts for robust interactions.

From a technical point of view, we have interfaced the Yun board with several kinds of sensor node, available on the market. Each sensor node consists of an accelerometer, a dedicated microcontroller, and an interface which enables the communication with a computing device, like an embedded system. This interface can be a wireless (bluetooth or zigbee) or wired. Sensor nodes can be powered by using a coin cell battery or a rechargeable battery. So, these devices are very small and non invasive; for example, they can easily be attached on the head. Furthermore, this hardware solution does not impair the natural movement of the body: examples of possible placements of the sensor nodes are given in figures 3 and 4.

Here we focus on head movements, since the head is the highest body segment and in consequence, the last affected part in spinal cord injury patients. Additionally, many people are unable to use a standard computer mouse because of disabilities affecting their hands or arms. Head controls offer one alternative by allowing people to use head movements to control the computer cursor. Human head-movements can be used to control any other two dimensional device. By using an accelerometer attached at the head, we are able to



Figure 4. A wireless accelerometer-based sensor node attached on the head. In this case, it is connected to the Arduino Yun wirelessly.

detect four kinds of movements, that are left and right roll and up and down pitch, as shown in the Figure 5.

#### 4.1. The mouse emulator

According to the proposed mouse emulator, the movements made with the hand to move the mouse pointer are substituted by head movements. Specifically, up and down vertical movement (pitch) is used for up down displacement while roll movements are used for left right displacement. So the end user can fully



**Figure 5.** On the left, there is a roll head movement. On the right, there is a pitch head movement.

control a computer: for example, he/she can use an on-screen keyboard. Such a solution provides also a novel interaction method for controlling a computing environment through very intuitive body movements given that a sensor node can detect measurement of inclination changes less than  $1.0^\circ$ . This is a very important feature which can be a key element for independence and autonomy of people with disabilities: in these cases, enabling alternative and simple interaction with a standard personal computer becomes critical for these kinds of users.

Since the Linino OS supports a Python runtime, in this work we have implemented the software infrastructure depicted in Figure 2 by means of this high-level programming language. Also, we have created a Bash script to execute the `sysupgrade` utility in order to perform the Initiator activities. Specifically, by using a `git push` command, we are able to download the operating system image with tailored plugins software from a remote virtual machine over the Cloud.

## 5. Conclusion

The present paper describes a solution aimed at supporting the interaction with computing devices for people with disabilities in smart IoT environments. Generally, these types of users interact with computer-based platforms by means of tailored AT pieces of software that are closely tied to the particular physical device on which they run.

By using embedded platforms, based on Linux embedded systems and IoT devices, it is possible to interface these user's personal AT equipments in order to control other different types of computers without having to install any additional software on them. Specifically the proposed system exploits a USB client connection to interface a generic end-device, while it uses the HID protocol to emulate the mouse cursor actions and movements on the controlled computer.

Currently the interaction between our platform and external AT equipments requires the development of dedicated pieces of software interfaces, called plugins. Each plugin is able to process data from a specific AT tool. Nowadays, some plugins for AT running on an Android smartphone have been developed.

In future works, we plan to extend our system by developing plugins for others kind of AT equipments and sensors. In addition, new features will be added in next versions of our system and we will be able to increase the performance of the proposed input device.

**Acknowledgement.** The research leading to the results presented in this paper has received funding from the Project "Design and Implementation of a Community Cloud Platform aimed at SaaS services for on-demand Assistive Technology".

## References

- [1] ISO (2007) 9999-2007: *Assistive Products for Persons with Disability - Classification and Terminology*.
- [2] MARIANO, D., FREITAS, A., LUIZ, L., SILVA, A., PIERRE, P. and NAVES, E. (2014) An accelerometer-based human computer interface driving an alternative communication system. In *Biosignals and Biorobotics Conference (2014): Biosignals and Robotics for Better and Safer Living (BRC)*, 5th ISSNIP-IEEE (IEEE): 1–5.
- [3] POLI, R., CINEL, C., MATRAN-FERNANDEZ, A., SEPULVEDA, F. and STOICA, A. (2013) Towards cooperative brain-computer interfaces for space navigation. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces, IUI '13* (New York, NY, USA: ACM): 149–160. doi:10.1145/2449396.2449417, URL <http://doi.acm.org/10.1145/2449396.2449417>.
- [4] ZHANG, D., WANG, Y., CHEN, X. and XU, F. (2011) Emg classification for application in hierarchical fes system for lower limb movement control. In *Proceedings of the 4th International Conference on Intelligent Robotics and Applications, ICIRA'11* (Berlin, Heidelberg: Springer-Verlag): 162–171. doi:10.1007/978-3-642-25486-4\_17, URL [http://dx.doi.org/10.1007/978-3-642-25486-4\\_17](http://dx.doi.org/10.1007/978-3-642-25486-4_17).
- [5] XIONG, A., CHEN, Y., ZHAO, X., HAN, J. and LIU, G. (2011) A novel hci based on emg and imu. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on: 2653–2657*. doi:10.1109/ROBIO.2011.6181705.
- [6] ELDER, T., MARTINEZ, M. and SYLVESTER, D. (2014) Alternate computer input device for individuals with quadriplegia .
- [7] CARRERA, A., ALONSO, A.A., DE LA ROSA, R., AGUIAR, J.M. *et al.* (2013) Biomechanical signals human-computer interface for severe motor disabilities. *E-Health Telecommunication Systems and Networks* 2(04): 65.
- [8] CARRINGTON, P., HURST, A. and KANE, S.K. (2014) Wearables and chairables: inclusive design of mobile input and output techniques for power wheelchair users. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (ACM): 3103–3112.
- [9] COLLECTIVE, B.M. and SHAW, D. (2012) Makey makey: improvising tangible and nature-based user interfaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction* (ACM): 367–370.