	INSTRUCTIONS FOR MANAGING ALEx RS WITH AN EXTERNAL COMPUTER	**IS_00_01**
		Pag. 1 di 13
		Rev. 00

WEARABLE ROBOTICS

INSTRUCTIONS FOR MANAGING ALEx RS WITH AN EXTERNAL COMPUTER

SUMMARY

1. SCOPE

The purpose of this instruction is to explain how to connect an external computer to the ALEx RS device bypassing the high-level computer.

In this case, ALEx RS will be controlled by the external computer and will **lose its CE marking**.

3. REFERENCES

N/A

4. HARDWARE CONFIGURATION

ALEx-RS (Arm Light Exoskeleton Rehab Station) is a complete platform, combining robotic and virtual reality technologies, expressly conceived for providing therapeutic exercises for the motor and functional rehabilitation of the upper limbs.

ALEx-RS is composed of 11 macro- components, 6 of which have a hardware core and 5 are purely software. These components are not devices accessories intended to be used in combination with ALEx-RS but are embedded with the device. Moreover, relating the software, there are not variants that can be marketed.
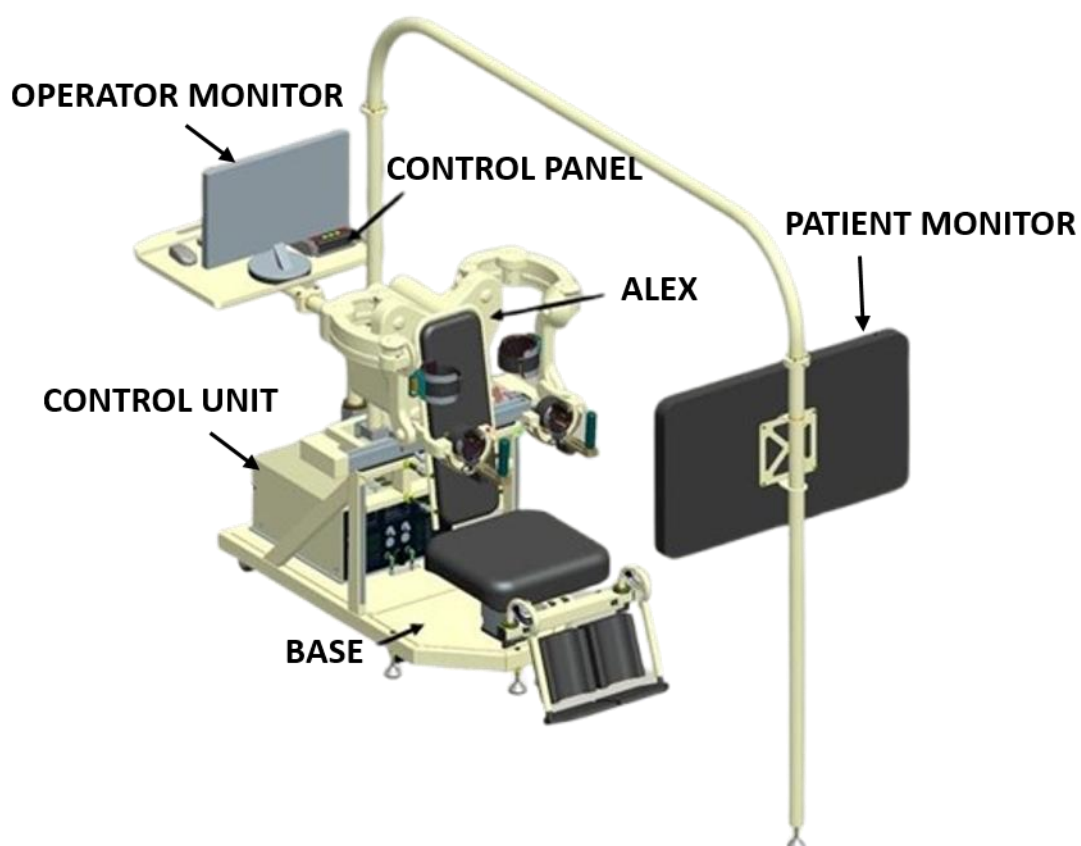

Figure 1: CAD view of ALEx RS

More in detail, the macro-components having a hardware core are the following:

1. **ALEx**

The bilateral arm exoskeleton, composed by two independent and symmetrical arm exoskeletons.

2. **Base**

The holding frame equipped with motorized linear guides.

3. **Control Unit**

It provides power supply for all system components and allows the implementation and management of all system functionalities.

4. **Control Panel**

It provides the system status display and allows the position adjustments of the seat and the two exoskeletons composing ALEx.

5. **Acquisition Board (ACQ)**

It provides the acquisition of the ALEx sensors (joint absolute position sensors and handle pressure sensors). It has been expressly developed for the application and for being embedded into the mechanics of ALEx.

6. **Motor Driver Board (MD)**

It provides the modulation of the electric energy supplied to the ALEx motors. It has been expressly developed for this application and for being embedded into the mechanics of ALEx.

The purely software macro-components are the following:

• **Games**

The module that generates the Serious Games.

• **GUI**

The module that implements the Graphical User Interface to access system functionalities.

• **Device Control**

The module allowing the exoskeleton low level control.

• **Device Interface**

The interface between the Device Control and the Graphical User Interface.

• **Device Test & Settings**

The module allowing the setting of the device parameters and the execution of the production final tests.

Figure 2: Physical model of ALEx RS. The macro-components are depicted as a box with a light green background.

5. WIRING FOR EXTERNAL CONNECTION

IMPORTANT: THE EXTERNAL COMPUTER MUST HAVE AN INTEL-TYPE PROCESSOR.

The first step in connecting an external computer to ALEx RS is to remove the protective plate highlighted in the figure below. To remove the plate, simply unscrew the four Allen-head screws that secure it to the control unit (CU).

LAN PORT

Figure 3: how to connect an external computer to the CU.

After removing the plate, the LAN port will be accessible. The external computer can now be connected via a simple LAN cable.

6. ALEx RS HIGH-LEVEL COMPUTER CONFIGURATION

This section specifies all the configuration steps of the ALEx RS high-level computer to allow access from outside.

1. Switch on the ALEx RS.
2. Connect a USB stick (minimum capacity 32GB) to one of the ports on the operator monitor.
3. Copy and paste the 'Release' folder in 'C:\' on the USB stick.
4. Create the windows bridge by following the steps in this video up to minute 4:10 (https://www.youtube.com/watch?v=1qxUYKO_4WA).
5. Close the GUI and all black windows in the background except the window with the name 'Power312'.

7. EXTERNAL COMPUTER CONFIGURATION

This section describes the steps to configure the external computer that is to manage ALEx RS.

1. Paste the 'Release' folder on the external computer.
2. Set the IP address of the IPv4 protocol to 10.24.4.215 and subnet mask to 255.255.255.0
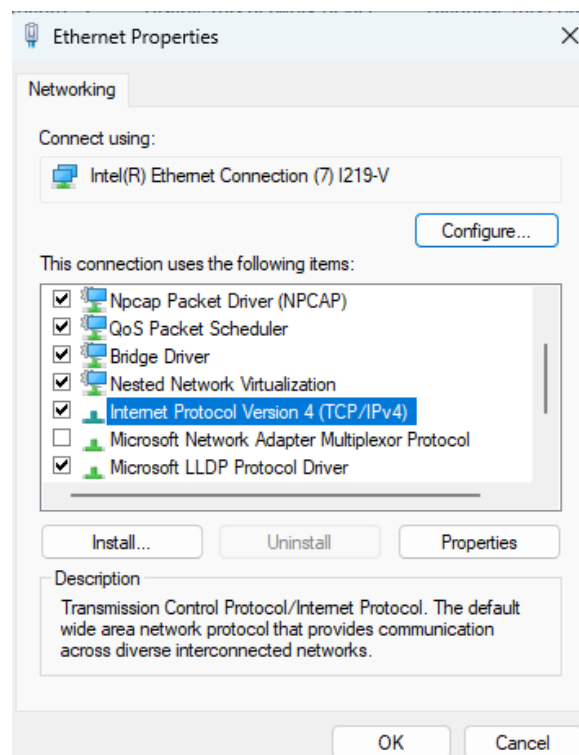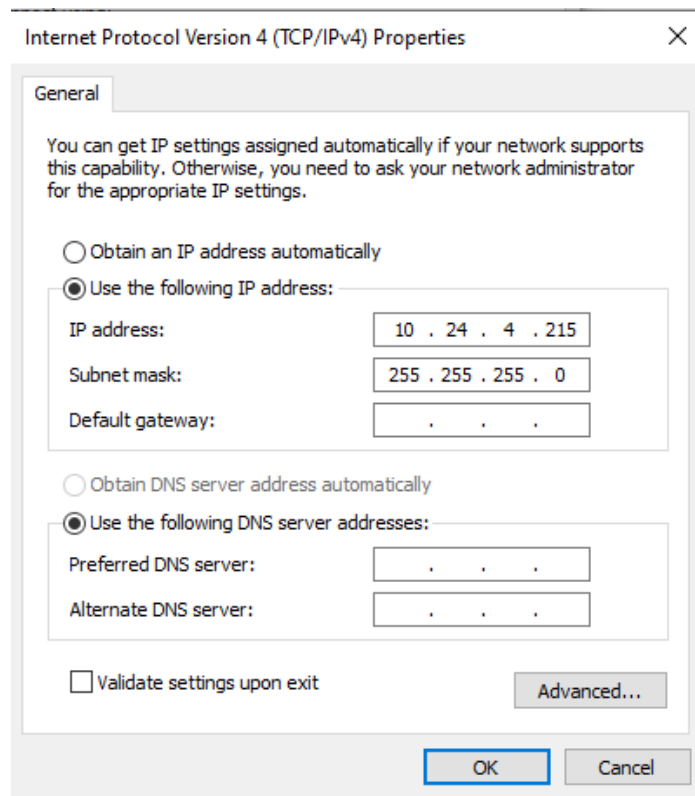

Figure 4: Ethernet properties

Figure 5: IPv4 properties

3. Open the 'Release\dll\' folder and double-click on the file 'Lancia_tutto_32.bat.
4. Close the black window in the background named 'POWER312'.
5. <mark>Check that the label in the top left-hand corner of the GUI changes from 'disconnected' to 'connected'.</mark>

At this point, if the label is 'connected', the external computer is able to manage ALEx RS operation.

TIP: we recommend using the GUI to manage the basic functionality of ALEx RS (e.g., turning on and off exoskeletons).

8. INFORMATION FOR CUSTOM SOFTWARE DEVELOPMENT

Before you start developing custom software for ALEx RS you need to know that there are two main ways:

1. using shared memories (safer)
2. by sending high-frequency UDP packets (most dangerous)

8.1 SHARED MEMORY USE

Using shared memories to manage ALEx RS is the <mark>simplest and most secure method because it allows access to data in real time and sends commands in a structured manner</mark>.
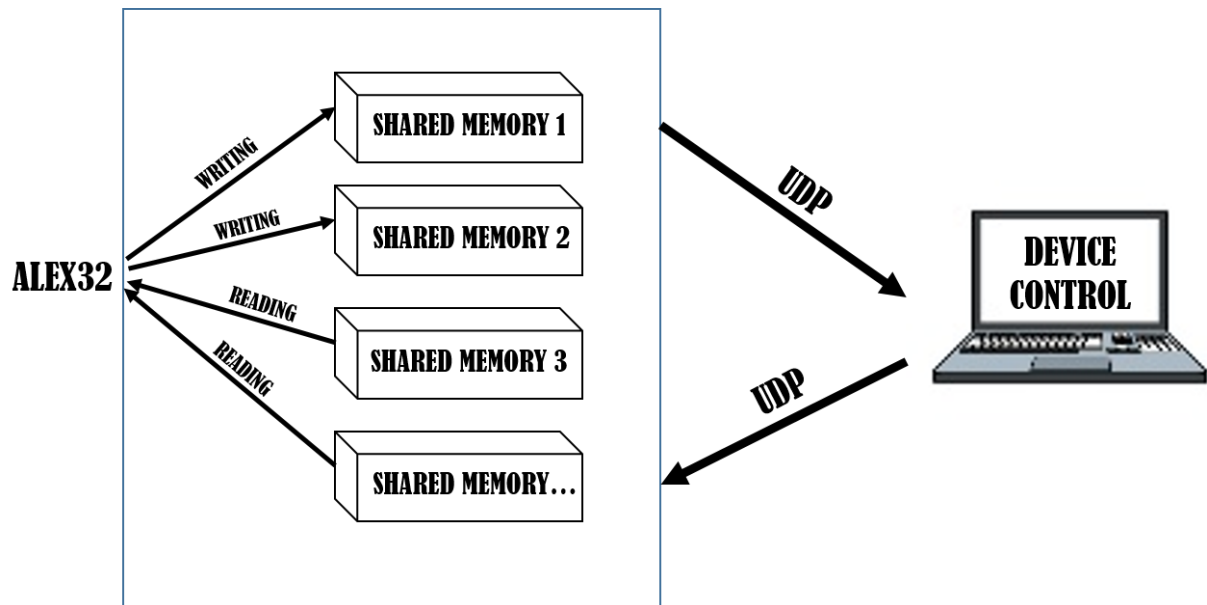
	INSTRUCTIONS FOR MANAGING ALEx RS WITH AN EXTERNAL COMPUTER	**IS_00_01**
		Pag. 8 di 13
		Rev. 00

WEARABLE ROBOTICS

Figure 6: ALEX32 Software

Management of packets sent and received from device control and shared memories is delegated to the ALEX32 software that runs in the background.

The ALEX32 software creates the shared memories, which are divided into two types: memories used to read data in real time and memories used to send commands.

This software acts as a bridge between the shared memories and the device control; it is responsible for writing to the shared memories what it receives from UDP packets and sending UDP packets from what is written to the shared memories.

The software developer user must, therefore, exclusively devote himself to writing or reading the data mapped in the shared memories.

8.2 USE UDP PACKAGES

Handling ALEx RS by sending UDP packets is a riskier method because the ALEX32 software is bypassed, increasing complexity and the possibility of error on the part of the developer user.

The developer user is responsible for creating the UDP packet by perfectly following the structure that the device control expects and sending it at a frequency of not less than 1kHz.

The developer user must also create a receiver that can decode UDP packets sent by device control to read data in real time.

The main difficulty lies in encoding and decoding UDP packets which, if not done correctly, can generate wrong commands (or no command) in sending or reading data that is not correct in receiving.

8.2.1 UDP PACKAGE STRUCTURE

UDP packets are divided into two types: Data In and Data Out.

Data In type packets are those that the device control sends in broadcast and contain all the (high-frequency) information related to kinematics, forces and torques of exoskeletons.

Data Out type packets are sent to the device control and contain parameters and commands to be applied to the exoskeletons.

Below are in detail the facilities of both types of packages.

vuol dire che è un array che contiene 8 variabili di tipo float.
è di 32 byte, quindi 4 byte ciascuno

Data In   inviati dall'exo

UDP Port = 25003

Packet size = 304 bytes

Packet description

Byte offset	Byte size	Data type	Arm	Description	Unit
0	4	Float[1]	--	Counter	--
4	4	Float[1]		Timer	second
8	32	Float[8]	Right	Angular position of the joints (all the joints)	rad
40	32	Float[8]		Angular speed of the joints	rad/s
72	16	Float[4]		Torque applied on the joints (only acted joints)	Nm
88	12	Float[3]		Handle position	m
100	12	Float[3]		Handle speed	m/s
112	12	Float[3]		Force applied on the handle	N
124	16	Float[4]		Target angular position of the joint (only acted joints)	rad
140	12	Float[3]		Target position of handle	m
152	4	Float[1]		Pressure applied on the handle	0 … 1
156	148	Float[37]	Left	Same variables of the right arm	
304					

rispetto quale sistema di riferimento ?

Data Out   input per l'exo

UDP Port = 26003

Packet size = 648 bytes

Packet description

Byte offset	Byte size	Data type	Arm	Description	Min	Max	No effect	Unit
0	4	Float[1]	--	Counter				--
4	4	Float[1]		Timer				second
8	12	Float[3]	Right	Force to be applied on the handle	Modulus of vector < 50		0	N
20	16	Float[4]		Torque to be applied on the joints (only acted joints)	-50	50	0	Nm
36	12	Float[3]		EE position control: Handle required position	--	--	--	m
48	12	Float[3]		EE position control: not used	--	--	--	--
60	132	Float[33]		EE position control: Impedance control parameters (see table below)	--	--	--	--
192	4	Float[1]		EE position control: maximum handle speed	0.001	0.5		m/s
196	4	Float[1]		EE position control: maximum handle force	0.001	50		N
200	16	Float[4]		Joint position control: Joint required position (only acted joints)	--	--		rad
216	16	Float[4]		Joint position control: not used	--	--		--

232	64	Float[16]		Joint position control: Impedance control parameters (see table below)	--	--		--
296	16	Float[4]		Joint position control: maximum joint speed	0.001	3.5		rad/s
312	16	Float[4]		Joint position control: maximum joint torque	0.001	25		Nm
328	320	Float[80]	Left	Same variables of the right arm				
648								

EE position control: Impedance control parameters

Byte offset	Byte size	Data type	Description	Min	Max	No effect	Unit
0	12	Float[3]	Positive region: Stiffness	0	1500	0	N / m
12	12	Float[3]	Positive region: Viscosity	0	150	0	N s / m
24	12	Float[3]	Positive region: Viscosity (absolute)	0	150	0	N s / m
36	12	Float[3]	Positive region: Stiffness modification speed	0	10000	10000	N s / m
48	12	Float[3]	Negative region: Stiffness	0	1500	0	N / m
60	12	Float[3]	Negative region: Viscosity	0	150	0	N s / m
72	12	Float[3]	Negative region: Viscosity (absolute)	0	150	0	N s / m
84	12	Float[3]	Negative region: Stiffness modification speed	0	10000	10000	N s / m
96	36	Float[9]	EE impedance rotation matrix	--	--	Identity matrix	--
132							

Joint position control: Impedance control parameters

Byte offset	Byte size	Data type	Description	Min	Max	No effect	Unit
0	16	Float[4]	Stiffness	0	200	0	Nm / rad
16	16	Float[4]	Viscosity	0	20	0	Nm s / rad
32	16	Float[4]	Viscosity (absolute)	0	20	0	Nm s / rad
48	16	Float[4]	Stiffness modification speed	0	10000	10000	Nm s / rad
64							