The App interface for Michelangelo prosthesis

Contents

About this interface

This interface enables you to control the OttoBock Michelangelo hand via a standard UDP connection. UDP is a very simple networking protocol that allows transmission of data packets (in this context, commands to/from the hand) between two applications, either, on same or different PCs. The Michelangelo hand can perform different movements like *Grasping* (either with Lateral or Palmar grasp), *Wrist Rotation* (both Pronation and Supination) and *Wrist Flexion/Extension*. To accomplish these movements, the prosthesis has different on-board motors, which move the motor-shaft/actuator to the commanded position.

With this interface, the Michelangelo hand can be controlled in two different modes,

- *Velocity control*: This mode allows you to control the velocity of individual motors i.e. closing/opening velocity of grasping or rotation velocity for Pronation/Supination.
- *Position Control*: This mode allows you to control the position of individual motor-shafts i.e. it allows you to pre-shape a grasp with a specific amount of closure or rotate to a specific angle w.r.t the neutral position.

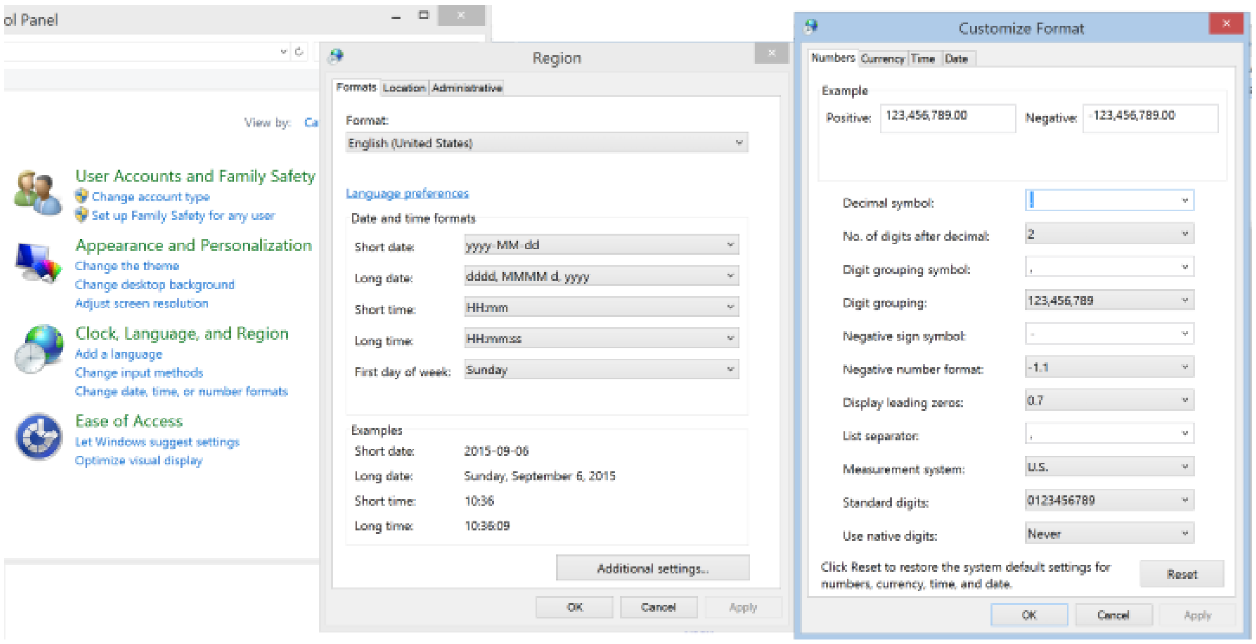
The Michelangelo hand also has on-board prosthesis sensors. With this interface, it is also possible to receive the sensor data from prosthesis; via UDP connection. The interface provides data from the following prosthesis sensors,

- *EMG*: Up to 8 EMG sensors can be connected to the prosthesis. The interface can stream EMG data at either 100 Hz or 1000 Hz.
- *Force*: The grasping force exerted on any grasped object can be measure by using the on-board Force sensor.
- *Actuator Position*: The rotation and flexion angles w.r.t the neutral position are available via the interface.
- *Internal controller states*: The states of the embedded prosthesis controller that is used to drive the prosthesis as programmed by the therapists;

Installation

Before connecting the hardware, the drives for the Bluetooth dongle must be installed. These drives are available under the folder *…/Install/Drivers/USB Drivers 3.0* (you needs to unzip the *USB Driver 3.0* folder, before using it). Open the folder *USB Drivers 3.0* by double-clicking it and run the file *setup.exe.* Then click on 'Continue' until the installation process is over. Only after the installation is complete, connect the available Bluetooth dongle to your USB port.

Additionally, you should also check, that the system number format is set to use the "."- as a decimal point; and not "," (otherwise the calibration files will fail to load). You can easily check this by going to the *“Control Panel”* and then to *“Change date, time or number formats”* and then under *Additional Settings* as shown in Figure 1.



Format: English (United States)

Date and time formats

Short date: yyyy-MM-dd
Long date: dddd, MMMM d, yyyy
Short time: HH:mm
Long time: HH:mm:ss
First day of week: Sunday

Examples
Short date: 2015-09-06
Long date: Sunday, September 6, 2015
Short time: 10:36
Long time: 10:36:09

Example
Positive: 123,456,789.00 Negative: -123,456,789.00

Decimal symbol:
No. of digits after decimal: 2
Digit grouping symbol: ,
Digit grouping: 123,456,789
Negative sign symbol: -
Negative number format: -1.1
Display leading zeros: 0.7
List separator: ,
Measurement system: U.S.
Standard digits: 0123456789
Use native digits: Never

Click Reset to restore the system default settings for numbers, currency, time, and date.

Figure 1: Checking the System Number Format

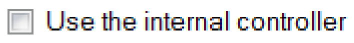
Operation modes and prosthesis types

1. Two types of available prosthesis:
    a. Gripper + Rotation Unit: This prosthesis can grasp using two different grip-types, namely, Palmar (or Power) and Lateral (or Flat) grip. It also has a rotation unit that allows Pronation and Supination of the hand.
    b. Gripper + Wrist Unit: This prosthesis permits grasping using Palmar and Lateral grasp. It has a wrist unit with 2 full degrees of freedom with which the hand can perform Pronation, Supination, Flexion and Extension (unlike the previous hand, which only supports Pronation and Supination).

2. There are two different types of USB Bluetooth dongles available, namely, *FAST (BaudRate: 460800)* and *SLOW (BaudRate:115200)*. Further, the prosthetic hands are fitted with *controllers* that support transmission at either 100 Hz or 1000 Hz. The SLOW-Dongle can only connect with the 100Hz-Controller. The FAST-Dongle can connect with both the 100Hz and 1000Hz-Controller.

3. Internal Controller: The controller connected with the prosthesis supports the *standard* state-of-the-art (SoA) control as programmed by the therapist via the OttoBock AxonSoft software package. The usefulness of this mode is reflected in the fact that if you need the SoA control algorithm, you don’t need to program it yourself, rather, you can use the OttoBock software suite

to program the internal prosthesis controller in many different operation modes. If you need to use the prosthesis in this mode, you must check the "Use the internal controller" checkbox on the GUI, before pressing the *Connect* button.


☐ Use the internal controller

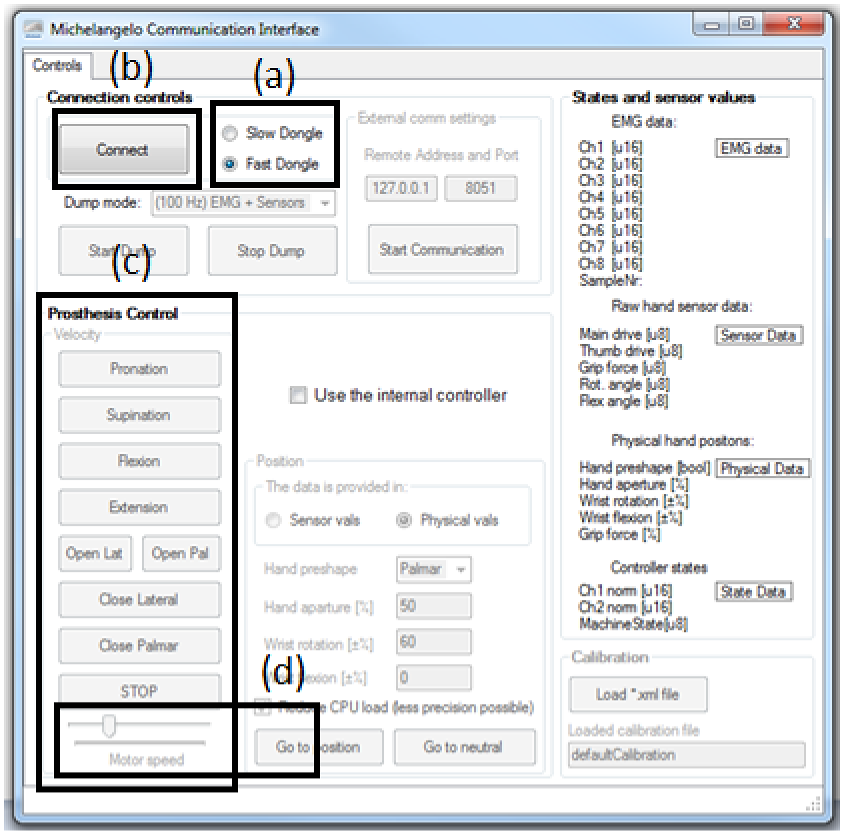
In this mode, you will not be able to send commands to the prosthesis from the GUI (i.e., it will not respond to them). But, you can still log and stream the data @100 Hz from the prosthesis (e.g. sensor data, EMG data or controller-state). You can collect this data from the provided interface (via UDP communication) and store it for further analysis.

Connecting to the prosthesis

1. Turn ON the prosthesis:
   Press the button (on the prosthesis) for a long time until you here two short beeps twice. So, the sound that you shall hear will be like "...beep, beep ... beep, beep..." (thus, in total 4 beeps). If you hear only the first two beep (like: "...beep, beep,...") then the Bluetooth is not active. After you here the first two beeps you need to wait for the next two beeps (before releasing). The light will continuously light blue.

2. Open the GUI:

   From the provided root folder of the application, go to DLLs/… and click on the application "MichelangeloGUI" (the .exe file). You shall see the User Interface depicted in Figure 2. To connect with the prosthesis, follow the steps mentioned below,



Dump mode: (100 Hz) EMG + Sensors

Remote Address and Port: 127.0.0.1 8051

Slow Dongle [✓ Fast Dongle]

Sensor vals [✓ Physical vals]

Hand preshape: Palmar

Hand aparture [%]: 50

Wrist rotation [±%]: 60

Wrist flexion [±%]: 0

Loaded calibration file: defaultCalibration

Figure 2: The view of the GUI when started

a. Connect the Bluetooth Dongle to your computer. Based on the type of Dongle you have (i.e. either SLOW or FAST), on the GUI, you need to select the correct dongle-type by clicking on the corresponding RadioButton shown in Figure 2-box (a) (i.e. click on either 'Slow Dongle' or 'Fast Dongle' RadioButton).

b. Then press the "Connect" button and wait until the program connects to the Prosthesis. The text of the "Connect" button will change to "Disconnect" and hereafter the same button shall be used to disconnect the prosthesis from the program.

c. The panel on the left hand side in Figure 2, marked as (c), can be used to send '*velocity commands'* to the prosthesis. You can control the velocity (or speed) of any motor using the slider "Motor speed" (represented via the box (d)).

3. Controlling the prosthesis in the Velocity mode (available in both 100 and 1000 Hz mode, but only if the option "use internal controller" is not checked)
Now you should be able to move the prosthesis by clicking buttons. For example, if the *Motor Speed* slider is set to 20% and if you press the *Pronation* button, then the prosthesis will start pronating with 20% of the maximum motor speed. To stop pronating, click on the 'STOP' button.

4. Loading the Calibration File: The prosthesis sends sensor data to our GUI, but this sensor data is *'raw'*. The calibration file is used to convert raw sensor data to human readable normalized sensor data. To load the calibration file click on the *"Load *.xml file"* (see Figure 3). Under the file path …/ MichaelAngeloHand/Calibration/…, you will find four Calibration files. Select the appropriate file name based on the prosthesis you have (and click Open). In case that you don't care for the position control or sensor data (e.g., you just want to read the EMG, or to use velocity control, use the defaultCalibration.xml)

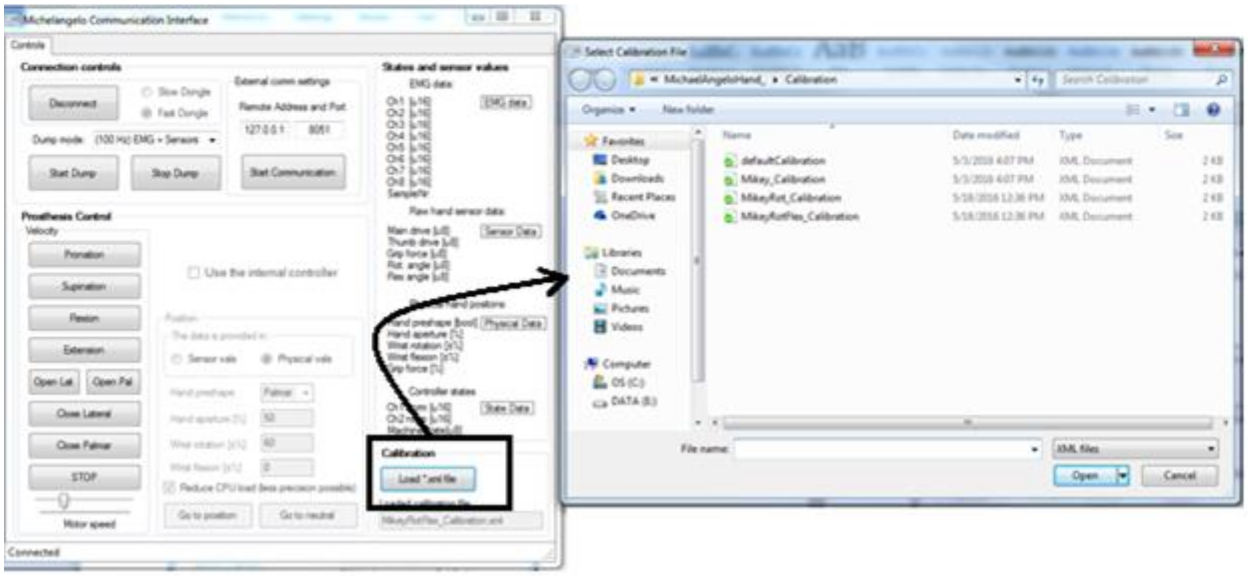


Figure 3: Loading the Calibration File

5. Configure the DUMP Mode:

   Under the *Connect/Disconnect* button, there is a *drop-down* menu labeled "*Dump Mode*". There are two dump modes available as shown in Figure 4,

   a. (100Hz)EMG+Sensors: This dump mode will stream EMG data +Sensor data at 100 Hz.
   b. (1KHz)EMG: This dump mode will not transmit Sensor data. But, it will transmit EMG data at a very high rate of 1kHz. This mode is useful when one needs to process raw EMG signals.
   c. To start the dump, select the required dump mode and click "Start Dump".

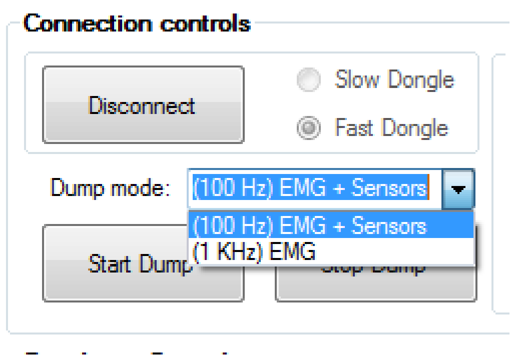
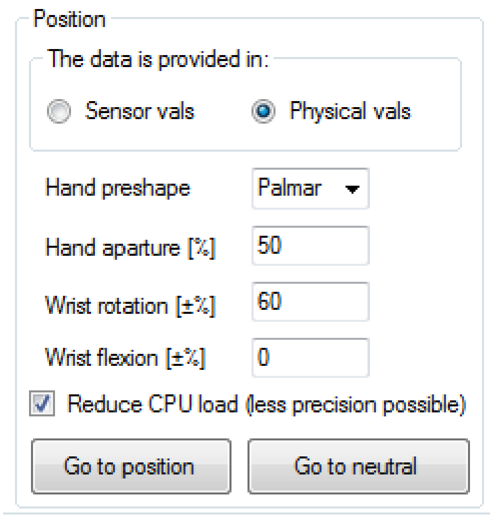


Figure 4: Dump Modes

6. Controlling Position of the Hand (available only if the 100 Hz mode is selected, the Dump is started and if the option “use internal controller” is not checked):

   The *position group-box* in the GUI is used to control the position of individual motors on the prosthesis (position group-box of the GUI has been shown in Figure 5). This option allows you to control either the ‘Sensor vals’ or the ‘Physical vals’. The Physical values are easy to interpret for humans, as they represent normalized sensor value in range [-100, 100]. In the position group-box, you can set the grip-type (either Palmar or Lateral), the hand aperture (in range [0, 100]) and the wrist position as:

   - Wrist rotation [±%]; where, a positive position value in range [0, 100] represents Pronation (clockwise displacement from 0) and a negative value in range [-100, 0] represents Supination (anti-clockwise displacement from 0).
   - Wrist flexion [±%]: where, a positive position value in range [0, 100] represents Extension and a negative value in range [-100, 0] represents Flexion (where, -100 means full flexion of the wrist).


Position

The data is provided in: Sensor vals [✓ Physical vals]

Hand preshape: Palmar

Hand aparture [%]: 50

Wrist rotation [±%]: 60

Wrist flexion [±%]: 0

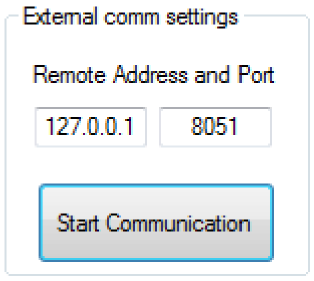
[✓] Reduce CPU load (less precision possible)

Figure 5: Position group-box in the GUI used to control the position of individual prosthesis motors.

Interfacing with the prosthesis from an external program

The biggest advantage of this interface is that it allows you to control and read the data from the prosthesis via UDP port opened in an external program (e.g. Matlab, C/C++, Python, etc.). To establish a UDP connection from an external program, do as follows:

a. Under the "External comm settings" group-box enter the (IP address, Port) at which your would like to send commands from your external code. As shown in Figure 6, the default IP and listening UDP port are set to 127.0.01 and 8051; it is advisible/convenient to directly use these settings.
b. Click on the "Start Communication" button, to control the hand from your external program (written in Matlab, C/C++, Python, etc.).


External comm settings

Remote Address and Port: 127.0.0.1 8051

Figure 6: External Communication Settings.

If you would like to receive sensor (e.g., grasping force, hand position) or EMG data from the Prosthesis. you should click on the "Start Dump" button. The interface will send sensor data on port 127.0.0.1/8052 (i.e. your program should listen for sensor data UDP packets at port 8052).

The Packet Coding Scheme:

1. For the Sender (i.e. to control the prosthesis via UDP packets):

   Three modes are supported as follows,

   a. "*Velocity Mode*": it has 9 unsigned-byte
      i. `byte0 = 1: to indicate velocity-control mode`
      ii. `byte1 = Palmar Grip Closing command in range [0, 255]`
      iii. `byte2 = Palmar Grip Opening command in range [0, 255]`
      iv. `byte3 = Lateral Grip Closing command in range [0, 255]`
      v. `byte4 = Lateral Grip Opening command in range [0, 255]`
      vi. `byte5 = Pronation Velocity in range [0, 255]`
      vii. `byte6 = Supination Velocity in range [0, 255]`
      viii. `byte7 = Flexion Velocity in range [0, 255]`
      ix. `byte8 = Extension Velocity in range [0, 255]`

   b. "*Position Control Mode*": it has 5 or 8 signed-bytes
      i. `signed_byte0 = 2: to indicate position-control mode`
      ii. `signed_byte1 = Grip Type, 0 - Palmar, 1 - Lateral`
      iii. `signed_byte2 = Grip Closure in range [0, 100]`
      iv. `signed_byte3 = Supination/Pronation, range [-100, 100]`
      v. `signed_byte4 = Flexion/Extension, range [-100, 100]`

      If there are only 5 bytes, then the speed will be always maximal. Otherwise:

      vi. `signed_byte5 = Maximum speed of the Grip, range [0, 100]`
      vii. `signed_byte6 = Maximum speed of the Rotation, range [0, 100]`
      viii. `signed_byte7 = Maximum speed of the Flexion, range [0, 100]`

   c. "Neutral Position": The hand will come back to the neutral position. It has only one byte
      i. `Byte0 = 0`

2. For the Receiver

   a. For the Dump mode *(1kHz) EMG data*: we get 18 bytes in a single UDP packet
      i. `byte0 and byte1 = 1st channel of EMG to convert in Uint16, range [0, 856]`
      ii. `byte2 and byte3 = 2nd channel of EMG to convert in Uint16, range [0, 856]`
      …
      iii. `byte14 and byte15 = 8th channel of EMG to convert in Uint16, range [0, 856]`
      iv. `byte16 = Counter`

   b. For the dump mode *(100Hz) Sensor + EMG Data*: we get 35bytes is a UDP packet.

____raw sensor data_____

      i. `byte0 = Main Drive in Uint8, range [0, 255]`

ii. `byte1 = Thumb Drive in Uint8, range [0, 255]`
iii. `byte2 = Rotation Angle in Uint8, range [0, 255]`
iv. `byte3 = Flexion Angle in Uint8, range [0, 255]`
v. `byte4 = Force in Uint8, range [0, 255]`

____normalized sensor data (human readable) ____

vi. `byte5 = Grasp Type in Int8, 0 for palmar and 1 for lateral`
vii. `byte6 = Aperture in Int8, range [-100, 100]`
viii. `byte7 = Pronation/Supination in Int8, range [-100, 100]`
ix. `byte8 = Flexion/Extension in Int8, range [-100, 100]`
x. `byte9 = Force in Int8, range [-100, 100]`

____Controller State, when in Co-Contraction Mode____

xi. `byte10 and byte11 = First Control Channel in Uint16, range [0, 856]`
xii. `byte12 and byte13 = Second Control Channel in Uint16, range [0,856];` These last two are normalized EMG
xiii. `byte14 and byte15 = Machine State in Uint16`

____Raw EMG data____

xiv. `byte16 and byte17 =` 1st `channel of EMG to convert in Uint16, range [0, 856]`
xv. `byte18 and byte19 =` 2nd `channel of EMG to convert in Uint16, range [0, 856]`
…
xxi. `byte30 and byte31 =` 8th `channel of EMG to convert in Uint16, range [0, 856]`

____Overhead for data management____

xxii. `byte32 and byte33 = two counters`
xxiii. `byte34 = Boolean which says if the hand has reached the command position`

A Sample Python Script

A python script is attached with the document named "MHand_Python.py". This script allows you to send triangular/square waves to the prosthesis in both velocity and position mode. It also receives data back from the prosthesis and plots the results. If you do not have experience with python, first, install *Anaconda with Python 3.5* (www.continuum.io/downloads). After the installation is complete, open *Spyder* it is a very simply IDE for Python programming (just like the editor in Matlab or Visual Studio for C/C++). Open the python script in *Spyder* and run the program (after appropriately connecting the prosthesis with the provided interface).

Appendix I: Prosthesis Technical specifications

These are the specifications, as provided by Ottobock. Use this table for conversion from normalized to physical values.

Hand		
	Length: Middle finger to proximal hand	180 mm
	Length: Middle finger to proximal including passive wrist	225 mm
	Length: Middle finger to proximal including active wrist	232 mm
	Gripping force in Opposition Mode	70 N (commercial), 100 N (research)
	Gripping force in Lateral Mode	60 N
	Gripping force in Neutral Mode	15 N
	Average speed of gripping	325 mm/sec
	Aperture	~11 cm in palmar, ~7cm in lateral
Rotation unit (commercial)		
	Rotation of the wrist	From -160° to 160°
	Speed of the rotation	Max. 25 rpm
Flexion unit (experimental)		
	Gripping force in Opposition Mode	45° in extension; 70° in flexion