brachI/Oplexus

User Manual
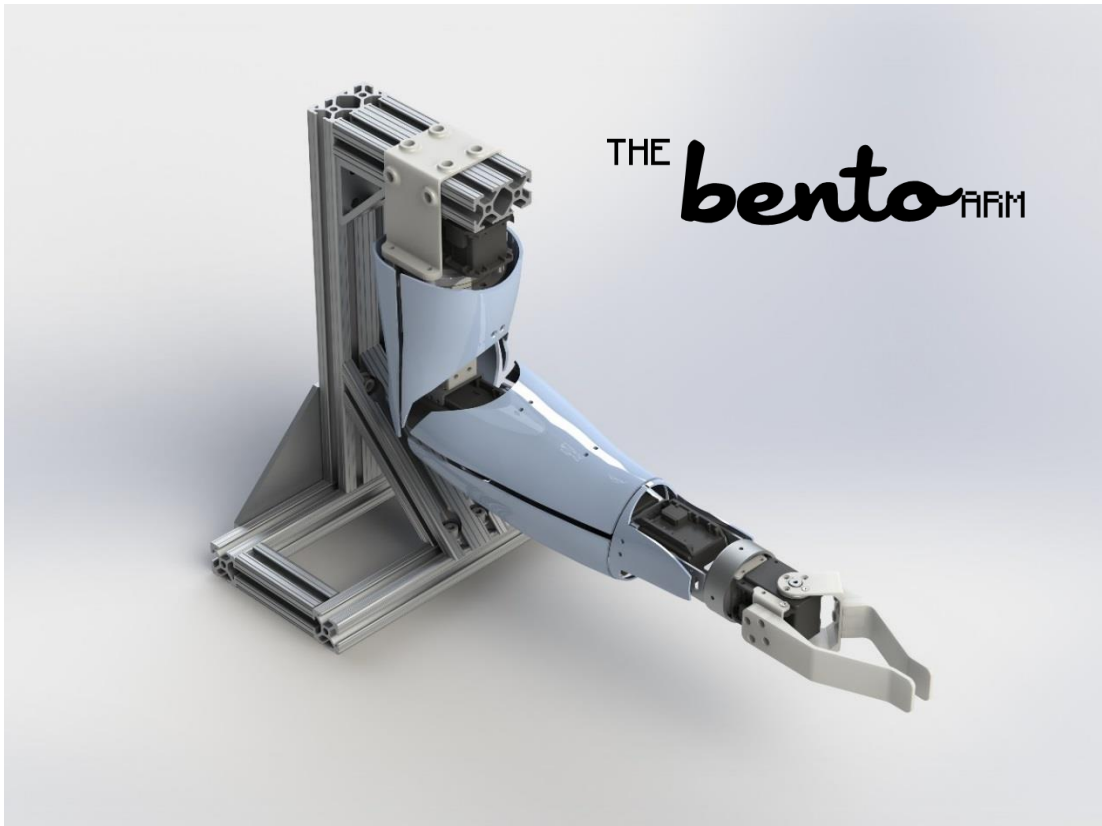
Michael R. Dawson, Helen Zhao

August 16, 2021

UNIVERSITY OF ALBERTA   BLINC   GLENROSE REHABILITATION · HOSPITAL FOUNDATION   amii

Table of Contents

THE bento ARM

brachI/Oplexus User Manual

1.1 Introduction

This guide will help you from start to finish with installing and operating the brachI/Oplexus software. The purpose of the software is to act as a digital nerve center for connecting human interfaces to robotic arms. The latest release of this software includes support for controlling the physical or virtual version of our open source robotic platform - The Bento Arm - with an Xbox 360 controller, MYO armband via muscle signals, or keyboard. Future releases will include additional human interfaces and robotic arms.

NOTE: brachI/Oplexus is pronounced 'brack-I-O-plexus' and is inspired by the anatomical term 'brachial plexus' which is the main network of nerves that connects the brain and spinal cord to your arm.

1.2 System Requirements

Supported Operating Systems
- Windows 10

Hardware Requirements
- 1.6 GHz or faster processor
- 2 GB of RAM
- 9 MB of available hard disk space

NOTE: brachI/Oplexus can work on slower systems (i.e. Tablets in the 1.4 GHz/2GB RAM) range, but delays may be longer and performance may be degraded (i.e. tabs may be less responsive and take longer to render and the motors may jump more noticeably when they stop themselves).

Supported I/O (input/output) devices
- Xbox 360 controller (wired)
  - Rock Candy Wired Controller for Xbox 360 (037-010)
  - Afterglow Wired Controller for Xbox 360 (PL-3702)
- Myo Gesture Control Armband
- Keyboard
- The Bento Arm v3
  - Open source files for 3D printing and assembling the Bento Arm
  - Open source files for the Virtual Bento Arm
- 12V/12.5A Power Supply
  - Inventus Power MWA150012A-12A
- Power Harness Cable for Dynamixel motors
  - Open source instructions for creating the cables can be found in the Bento Arm Assembly Guide section 1.5 page 11
- USB Extender Cable
- ROBOTIS USB2dynamixel or U2D2
  - Manual available here

1.3 Installation Instructions

1.3.1 Install brachIOplexus software

1. Download the latest release from github:
   https://github.com/blincdev/brachIOplexus/releases
2. Extract the installation files to a temporary folder
3. Double click on 'setup.exe' to start the installation.
4. If it is not already installed on your system, the setup will prompt you to install the 'Visual C++ 2015-2019 Redistributable (x86)' which is one of the required libraries. Click 'Install' to continue. If it is already installed then you can skip to step X
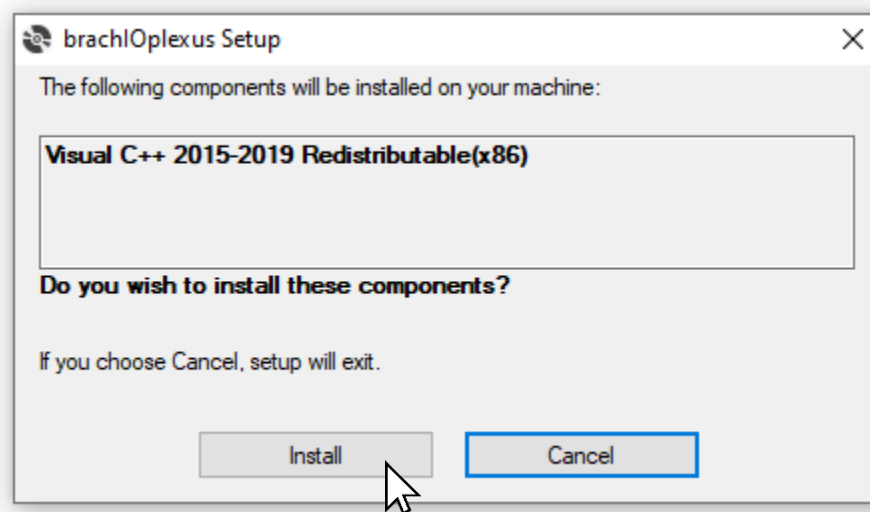

5. The Microsoft Visual Studio C++ 2015-2019 Redistributable (x86) installer will launch. Click 'Install" to continue. NOTE: an internet connection is required for this step in order to access the redistributable. If you prefer you can also pre-install the redistributable from the Microsoft website.
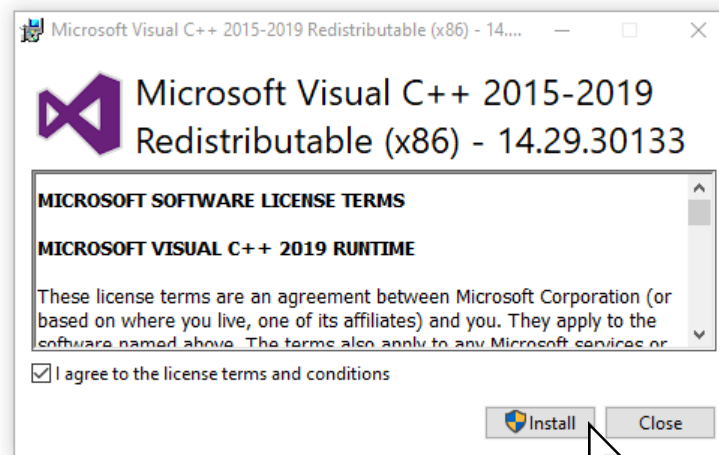
6. It may prompt with a user account control window asking 'Do you want to allow the following program to make changes to this computer. Program Name: VC_redist.x86.exe' Click 'yes'.
7. After installing it should say that setup was successful and it may ask you to restart the computer.
    a. If a restart is required make sure to save and close any other files or programs that may be open and then click 'Restart'. If it did not prompt for a restart then you skip to step b).
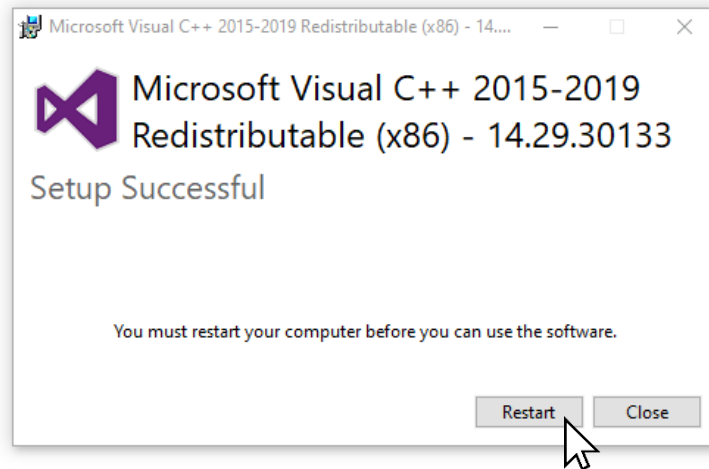
    b. If everything installs correctly it should now prompt you that the redistributable setup has been successful. Click 'Close'.
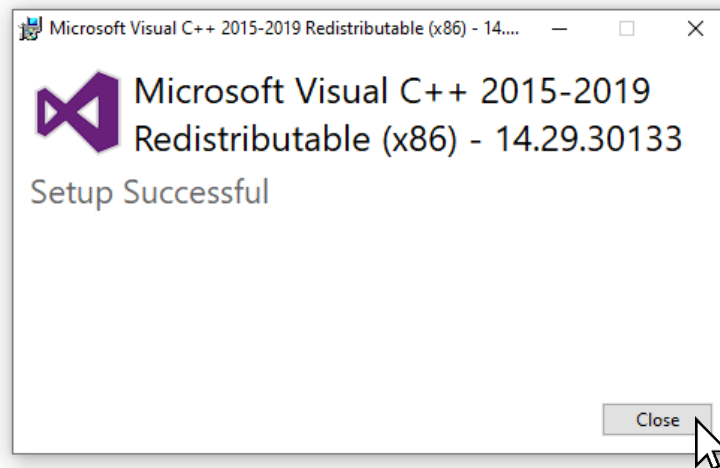
8. The rest of the installation can now resume. A window should pop up with a security warning for the brachI/Oplexus software. This is expected since we are not at this time able to register as a known publisher. Click 'Install' to continue.
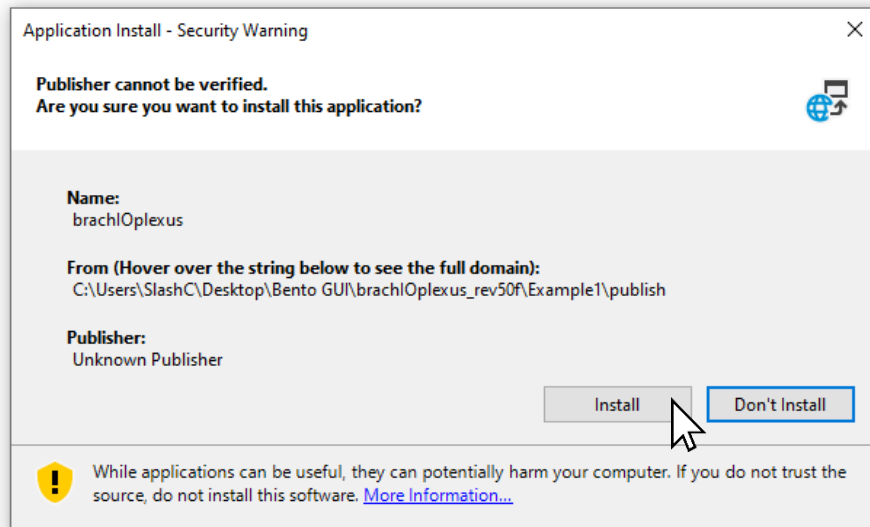
9. The installer should now finish up and launch the program. If the program opens successfully and looks similar to the image below then the installation has completed successfully!
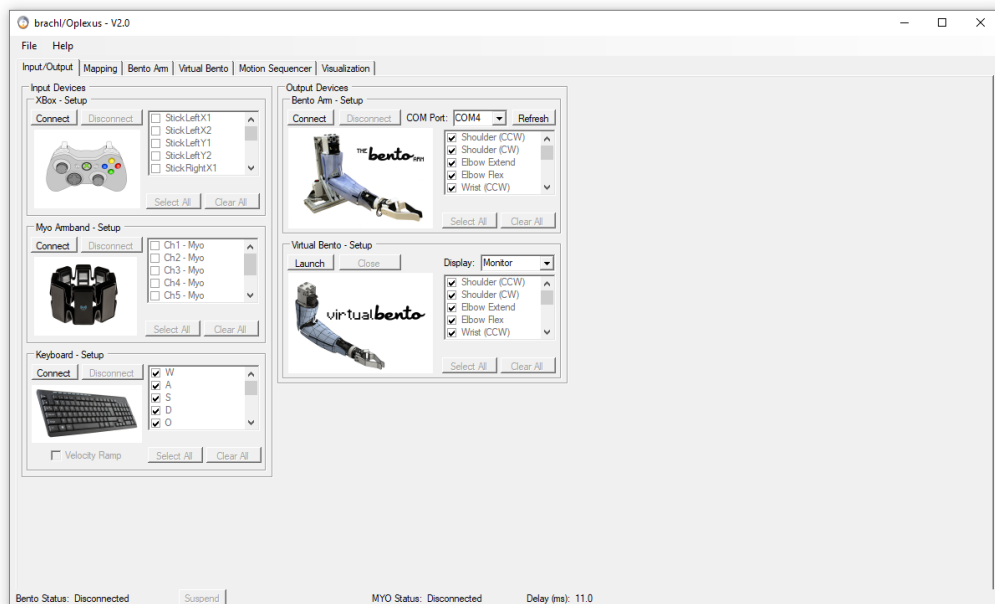
10. To open the brachI/Oplexus program in the future you can either go 'start → all programs → BLINCdev → brachIOplexus' or double click on the shortcut on your desktop labelled 'brachIOplexus'.
11. For instructions on operating the software please see section 2.

6

1.3.2 Install Xbox Drivers

1. Plug the Xbox controller into the USB port on your computer
2. Windows should automatically try to install drivers for the controller
3. Open the brachI/Oplexus software and click on the '`Connect`' button in the '`Xbox – Setup`' groupbox.
   a. If the '`Connect`' button greys out and the '`Xbox`' groupbox becomes active then the drivers have installed correctly and you should be able to press the buttons on the controller and see the feedback in the program. You can skip ahead to installing the USB2dynamixel drivers.
4. If the '`Connect`' button stays greyed out then the drivers are not pre-installed on your system and you will need to download additional drivers for Windows. NOTE: This typically only happens if you are running Windows 7.
5. You can download the Microsoft drivers for Windows 7 64-bit from the following link: Xbox 360 Accessories Software 1.2.
6. Open the downloaded executable and click '`Yes`' to approve the installation, if asked
7. Click '`Run`'. The Xbox 360 program will install the necessary drivers onto your computer. After it installs it may prompt you to restart the computer.
8. Now try opening up brachI/Oplexus software and see if the xbox controller will connect.
   a. If the controller works then you can skip to the next section
9. If the controller still does not work then run through the following procedure to manually install the driver
   a. Press the windows button in the bottom left of the desktop and typing '`device manager`' in the search bar. Once it appears click on it to open it.
   b. In device manager right click on '`rocky candy gamepad for xbox 360`' or similar and select '`properties`'
   c. Click on '`update driver`; and select '`browse my computer for driver software`'
   d. Select '`Let me pick from a list of device drivers on my computer`'
   e. Select '`Microsoft Common Controller for Windows Class`' and click next
   f. Select '`XBOX 360 Controller For Windows Version: 6.1.7600.16385 [21/06/2006]`' and click 'Next'
   g. An Update Driver Warning will pop up asking if you want to continue installing the driver. Click '`Yes`'
   h. It will then let you know that Windows has successfully updated your driver software. Click '`Close`'
   i. The controller should now work!

1.3.3 Install MYO connect

1. Download MYO connect.
2. Follow the comprehensive instructions in their installer to install the drivers and MYO connect software.
3. After installation right click on the MYO connect icon in the system tray and click 'Preferences…'
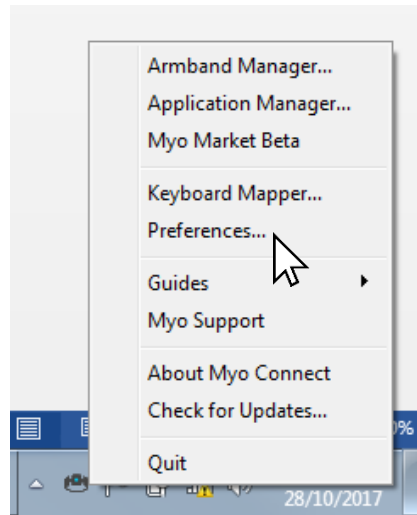
4. Under the general tab disable the 'Myo Quick Launch Menu' item in order to prevent software from inadvertently being launched while you are using the MYO with brachI/Oplexus. Click the 'x' button to close the preferences window.
5. Right click on the MYO connect icon and select 'Armband Manager…'
6. Under the general tab disable 'Show Myo gestures'
7. The setup for MYO connect is now complete.
   NOTE: You will need to make sure that MYO connect is open and that it is connected to the MYO armband in order for the signals to get into brachI/Oplexus

1.3.4 Install USB2dynamixel & U2D2 Drivers

The USB2dynamixel or the U2D2 controller can be used to communicate with the Bento Arm. You can proceed to step 1 (USB2dynamixel) or step 2 (U2D2) depending on which device you are using:

1. USB2dynamixel Driver Installation
   a. The easiest way to install the drivers for the USB2dynamixel is to install the latest version of their RoboPlus software. This Dynamixel Wizard part of the RoboPlus software is also used for the initial Bento Arm setup, so it is handy to have. The latest version can be found by going to their downloads site. The latest version to date is v1.1.3.0 and can be downloaded here
   b. Detailed instructions for installing the RoboPlus software can be found here
   c. Once the software has installed connect the USB2dynamixel to the computer using a USB extender cable
   d. Skip ahead to step 3 to continue.

2. U2D2 Driver Installation
    a. Connect the U2D2 to the computer using a USB extender cable.
    b. Download the latest virtual com port drivers from the FTDI website. The latest version is v2.12.36.4 and can be downloaded here.
    c. Extract the drivers to their own folder by right clicking on the zip file and selecting 'Extract All…'
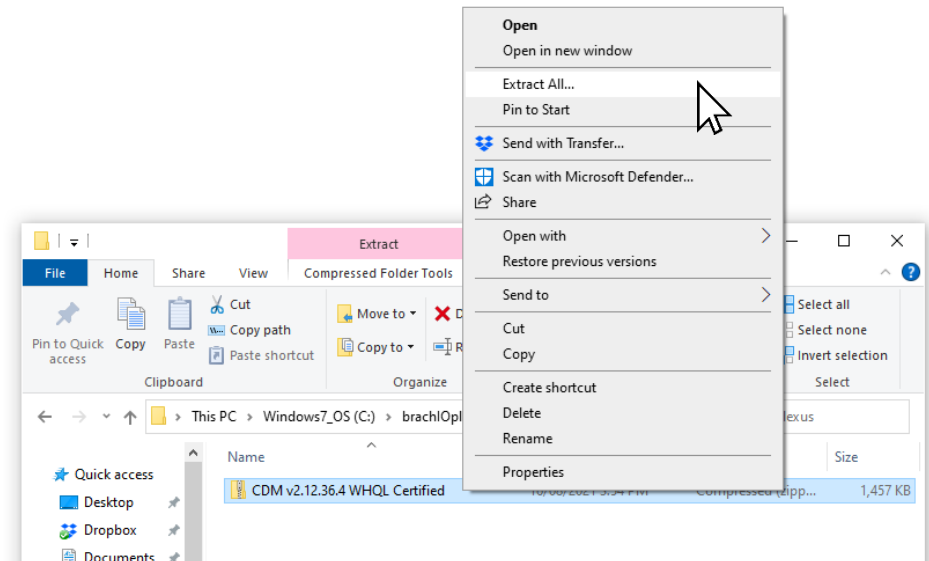
    d. Press the windows button in the bottom left of the desktop and typing 'device manager' in the search bar. Once it appears click on it to open it.
    e. Right click on 'USB ←> Serial Converter' and select 'Update Driver'
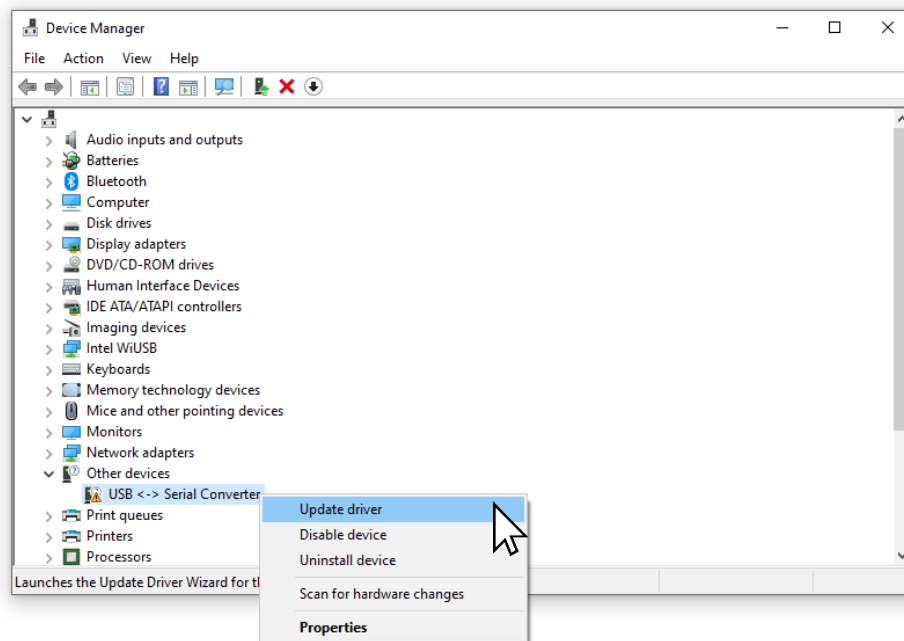
f. Select '`Browse for my computer for drivers`' and enter in the location of the folder where you extracted the drivers. Click '`Next`' to continue.
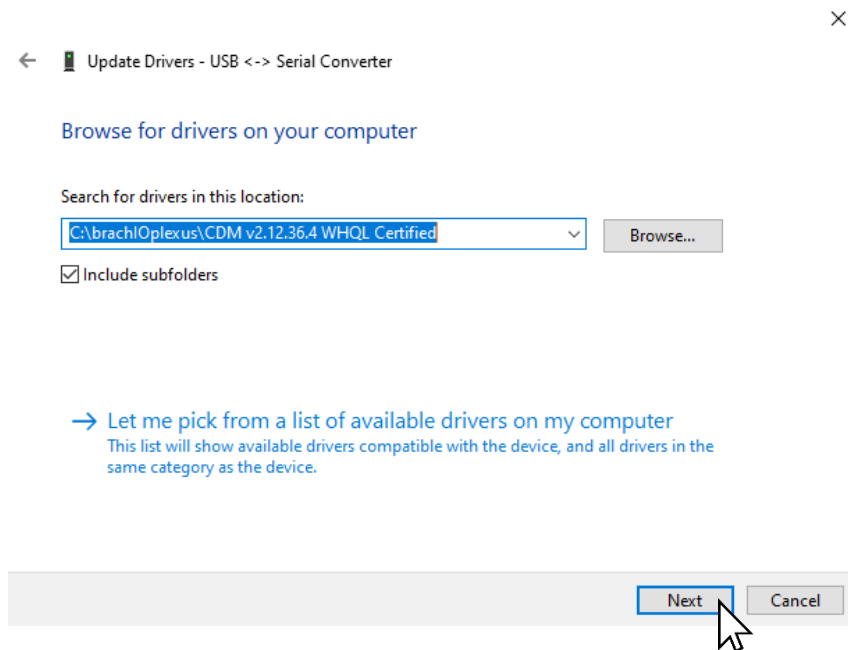
g. The driver should install and then a window should open indicating that '`Windows has successfully updated your drivers`'. Click '`Close`' to continue.

h. Open device manager again and right click on '`USB Serial Port`' and select '`Update Driver`'

i. Select '`Browse for my computer for drivers`' and enter in the location of the folder where you extracted the drivers. Click '`Next`' to continue.

j. The driver should install and then a window should open indicating that '`Windows has successfully updated your drivers`'. Click '`Close`' to continue.
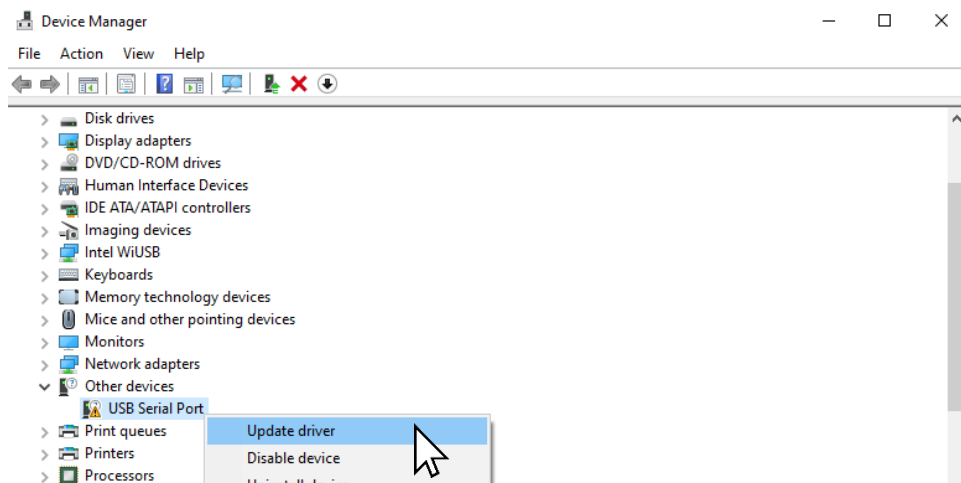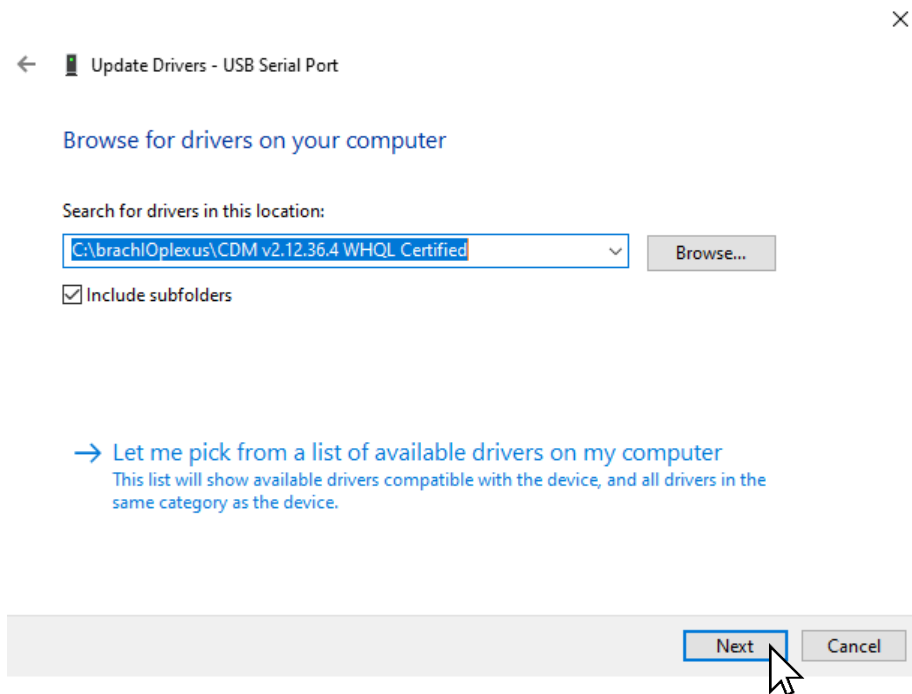
k. Both of the required drivers should now be installed.

3. Press the windows button in the bottom left of the desktop and typing '`device manager`' in the search bar. Once it appears click on it to open it.

4. In device manager click on '`Ports (COM &LPT)`' and right click on '`USB Serial Port (COMX)`' where X is some number and select '`Properties`'

5. A properties window should pop up. Click on the '`Port Settings`' tab and then click on the '`Advanced…`' button

6. An advanced settings window should pop up. Click on the drop down for '`Latency Timer (msec)`' and change it from 16 to 1.

   a. Note: You should only have to do this once per computer. When you connect the same USB2dynamixel or U2D2 it should remember the previous settings and keep the setting at 1ms.

7. Click '`Ok`' to close the advanced settings window then click '`Ok`' again to close the properties window. You can also close the device manager by clicking on the '`x`' button in the top right of the window

8. The USB2dynamixel/U2D2 is now ready to go!

9. If you have trouble with any of the above you can refer to the following youtube video that has a screen cap of the process.

1.4 Connecting the Hardware

This section will guide you through how to connect the input and output devices that you would like to use with brachI/Oplexus.

1.4.1 Input Devices

1. <u>Xbox</u> → connect it to a USB port on the computer
2. <u>Keyboard</u> → connect it to a USB port on the computer or an integrated laptop keyboard will also work
3. <u>MYO Armband</u>
    a. If the MYO armband is turned off you will need to turn it on by connecting it to a USB port on a computer using a micro USB cable.
    b. After it turns on (LEDs will light up) unplug the cable from the computer and MYO armband
    c. Connect the MYO dongle to one of the USB ports on the computer. If you have trouble with the MYO armband disconnecting later you might consider using a USB extender cable to extend the dongle away from the computer or laptop which often helps reduce interference.
    d. Slide the MYO armband onto one of the operator's arms making sure that the status LED and micro USB port face towards the hand. The LED should also face upwards in the same direction as the back of the hand. For best signals the band should sit over top of the flexor/extensor muscles on the forearm. If the operator flexes their wrist upwards the muscles can often be seen visibly or felt by touching the arm. The orientation of the MYO is somewhat arbitrary, but if you would like to use the pre-made profiles in brachI/Oplexus please follow the instructions above. Here is an example for how it should look when placed on the left forearm:
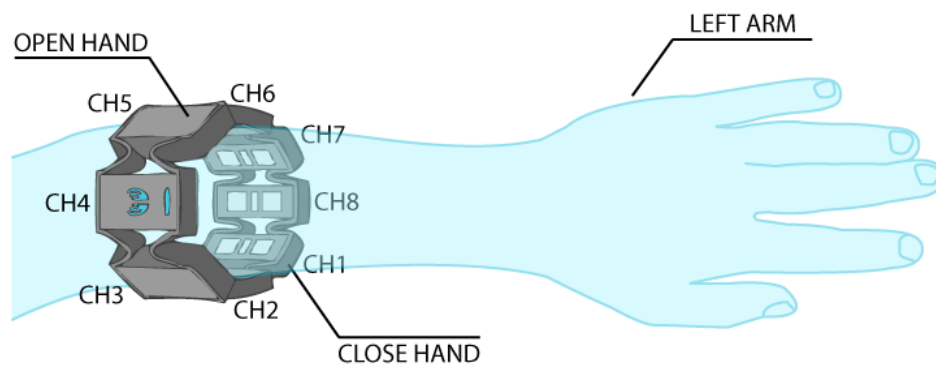
    e. If MYO connect is not already open you will need to open it now. Once it is open it should appear in the system tray (see section 1.3.3 for screenshot)
    f. Right click on the MYO connect icon in the system tray and select 'Armband Manager…'
    g. In the Myo Armband Manager click ensure the Myo Armband is connected by clicking the 'Connect' button.

h. Once connected the Armband should vibrate. Sometimes it vibrates intermittently for a while. To stop this vibration the operator should flex their wrist upwards for a few seconds to synch the MYO

i. The MYO armband is now ready for use.

j. NOTE: It can take 3-5 minutes for the muscle sensors aka electromyography (EMG) sensors to warm up. You may want to wait to setup gains/thresholds until after the signals have warmed up. You can tell when they've warmed up because the resting signal (i.e. when the muscle is not contracting) will stop decreasing appreciably as the skin/electrode impedance stabilizes from the skin sweating a bit. Once the warm-up period is complete the signal to noise ratio is usually improved and the signals may be a bit smoother.

1.4.2 Output Devices

1. Detailed instructions for how to connect the Bento Arm to its power supply and to the computer via the USB2dynamixel can be found in section 1.7 page 16, of the Bento Arm Assembly Guide. The guide will walk you through how to make sure that each dynamixel servo is set with the appropriate unique ID and a baud rate of 1000000 bps. It is also important to ensure that the servos are using the Dynamixel 1.0 protocol.
2. As a reference here is the recommended connection order:
   a. Connect the USB2dynamixel to the computer via the USB extender cord
   b. Connect the power to the Actuators via the power harness or if you have an electronics enclosure press the power button
   c. Connect the USB2dynamixel to the Actuators via the power harness.
   d. NOTE: when disconnecting please follow the reverse order (c, b, a)

1.5 Running the software

NOTE: The guide below is quite detailed and contains a description of every feature in the GUI. If you would just like to skip ahead and get the arm moving and learn the details later please check out our brief run-through video on youtube.

1. Open the brachI/Oplexus software by double clicking the shortcut on the desktop or by navigating to it via the Windows start menu.
2. Connect to the input/output devices that you would like to use by clicking on the corresponding '`Connect`' buttons in the '`Input/Output`' tab.
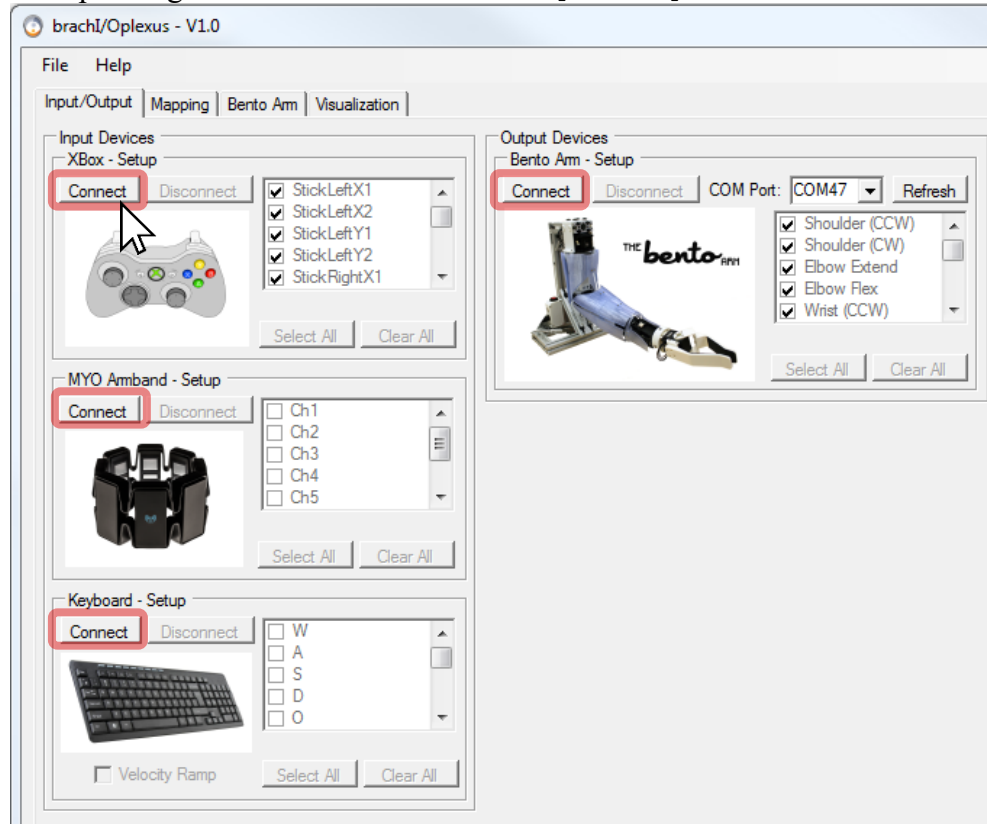
a. <u>Xbox</u> → Click the '`Connect`' button in the '`Xbox – Setup`' groupbox in order to connect to the Xbox controller. If the controller is plugged in and the drivers are installed then the '`Connect`' button should become greyed out.

b. <u>MYO Armband</u> → Click the '`Connect`' button in the '`MYO Armband – Setup`' groupbox in order to connect to the MYO Armband. If the controller is plugged and the drivers are installed then the '`Connect`' button should become greyed out. The '`MYO Status`' label on the status bar at the bottom of the GUI should now display '`Searching for MYO…`' or '`Connected`'. If it says '`Searching for MYO…`' you may need to go back to MYO Connect and ensure it is connected properly there.

c. <u>Keyboard</u> → Click the '`Connect`' button in the '`Keyboard – Setup`' groupbox in order to connect to the keyboard. If the keyboard connects properly then the '`Connect`' button should become greyed out.

   i. Check the '`Velocity Ramp`' checkbox if you would like to be able to give analog-like inputs using the keyboard by tapping the buttons. If you prefer the keyboard inputs to be on/off or digital-like then leave the checkbox unchecked.

   ii. NOTE: When you connect to the keyboard it will disable keyboard input on all of the controls in the GUI, so as to prevent inadvertent changes to mappings. You can re-enable the keyboard input by clicking the '`Disconnect`' button for the keyboard.

d. <u>Bento Arm</u> → Click the '`Connect`' button in the '`Bento Arm – Setup`' groupbox in order to connect to the USB2dynamixel. The COM port should auto-select as long as the USB2dynamixel was connected in advance of starting the program. If you connected it afterwards you can manually select it after clicking the '`Refresh`' button. If at least one Dynamixel Actuator is connected on the bus with an ID between 1 and 5 and bps set to 1000000 bps then the '`Bento Arm`' groupbox should become enabled. The '`Bento Status`' label on the status bar at the bottom of the GUI should now display '`Connected / Torque Off`'.

3. The '`Delay (ms)`' in the status bar at the bottom of the GUI should be about 2 to 5 ms while disconnected from the Bento Arm and about 3 to 7 ms when connected to the Bento Arm. This is about how long it takes for a background thread to read from the input devices, query a feedback packet from the actuators, and send a command packet to the actuators. If it is much greater -- then the latency timer in the USB2dynamixel driver may not have been set correctly. Refer to step 4 in section 1.3.4 and try setting it again to 1 ms. Also, we recommend not running any other programs or processes in the background while running brachI/Oplexus as it may effect performance and increase the loop delay.
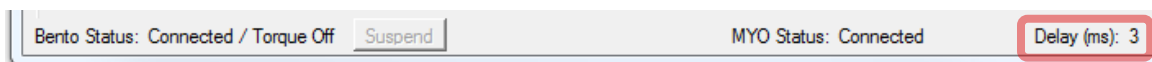
4. Select the buttons, channels, or movements that you would like to use as part of your custom mapping. These are the items that will populate into the input output combo boxes in the 'Mapping' tab. If you are unsure which ones you want to use at this time just click the 'Select All' buttons for each connected device.
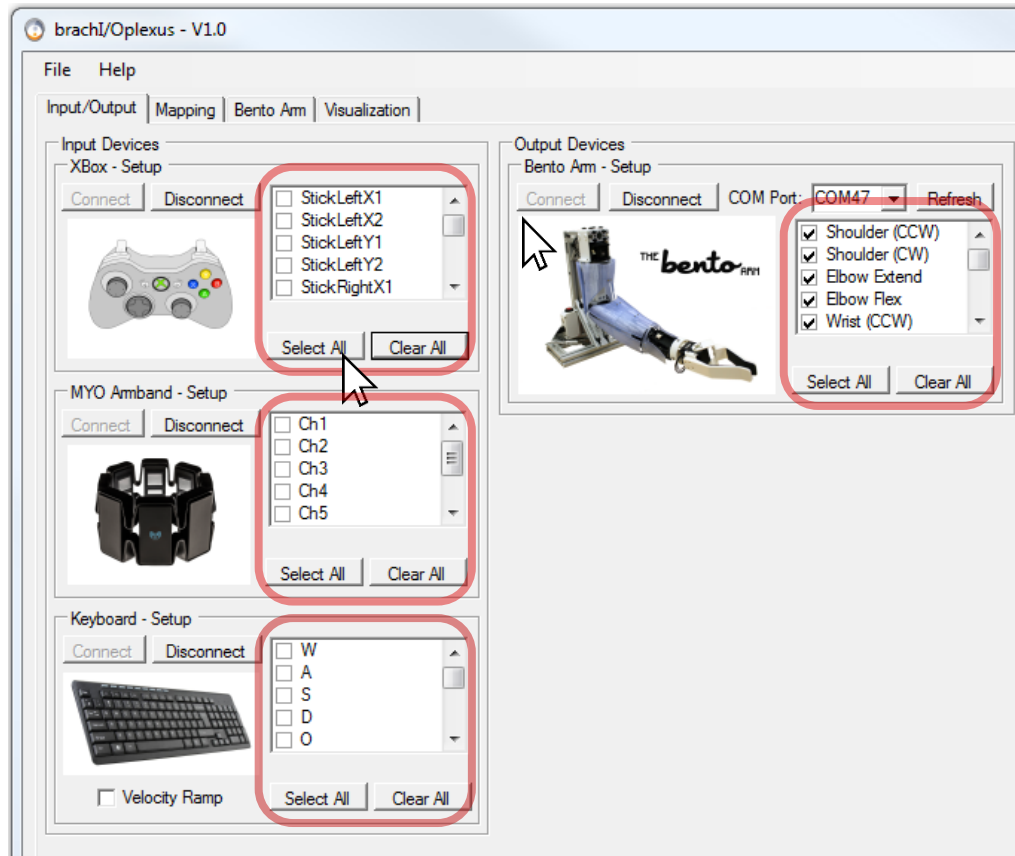
5. If you are unsure which buttons or channels correspond to different inputs on the controllers then you can click on the 'Visualization' tab to view the raw feedback from each connected controller. This is especially useful for the Xbox controller which has a lot of buttons and axes. NOTE: In the future this tab will allow you to visualize feedback from input and output devices in a pop out window and also include adjustable data logging features.
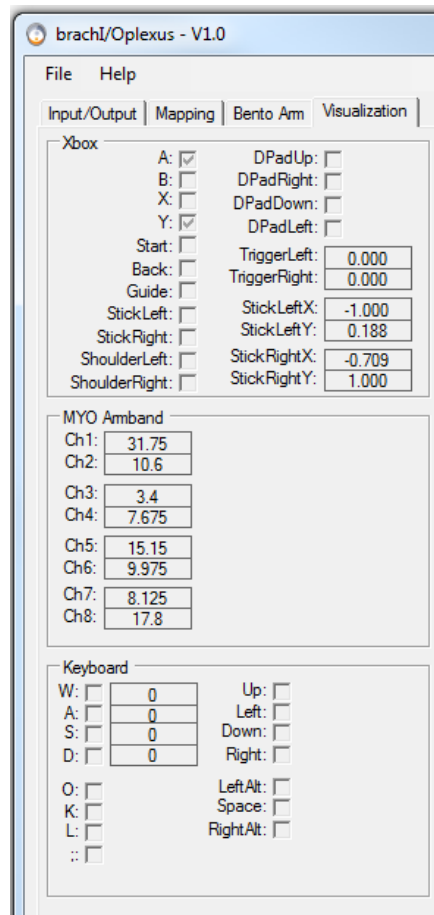
6. Click on the 'Mapping' tab in order to adjust or create custom mappings. For each degree of freedom you will need to select an input device, output device and mapping. Up to 6 degrees of freedom (DoF) can be mapped simultaneously in this version of brachI/Oplexus.
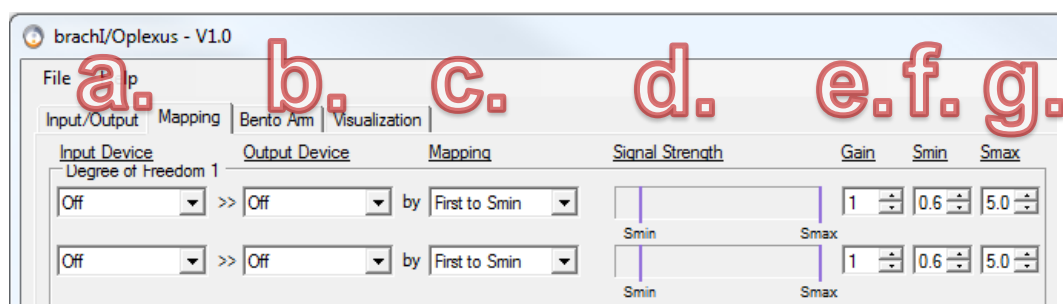
a. <u>Input device</u> → These are the items that you selected from the connected input devices in the '`Input/Output`' tab.

b. <u>Output device</u> → These are the items that you selected from the connected output devices in the '`Input/Output`' tab. You can also use this combobox to flip the direction of movement by swapping the items.

c. <u>Mapping</u> → In this release the only mapping available is '`First to Smin`' also known as "first past the post" which is a common linear proportional mapping used in robotics for mapping between analog input signals and the joint velocity of an actuator. When both channels are below their minimum thresholds (Smin) the actuator holds its position and does not move. When one channel exceeds its Smin the actuator will start moving in the corresponding direction at a velocity proportional to the input signal. The larger the input signal the faster the joint will move. If the signal exceeds the maximum threshold (Smax) it will not move faster than the maximum velocity specified for that actuator. Once the input signal drops below the minimum threshold it will stop moving. The first past the post comes into play in that for a given channel pair grouped in a degree of freedom only one of them can actually move the actuator at a time (i.e. you can't move an actuator in the clockwise and counter clockwise simultaneously). In the case where both signals are above their minimum threshold – the signal that got their first will be the one that takes precedence and actually causes the actuator to move in the correspondingly mapped direction. Future releases of brachI/Oplexus will include additional mappings for controlling joint velocities and joint positions of our supported robots.

d. <u>Signal Strength</u> → This is the signal output from the item selected in the input device (varies from 0 to 5) and appears as a blue bar graph. If the signal is maxed out above 5 then the bar graph will be colored red.

e. <u>Gain</u> → This is a gain multiplier that can be used to boost or reduce the signal strength. If the signal strength is too high you can reduce it by setting the gain to be a smaller value. If it is too weak then you can increase the gain to increase the signal strength. A good target is that the analog input signal should be able to get close to the maximum threshold at its maximum excursion (for a joystick) or at a comfortable contraction level (for EMG sensors).

f. <u>Smin</u> → the minimum threshold below which the actuator does not move and above which it will move. The Smin threshold acts as a deadband to help prevent inadvertent movements when the input signals are noisy or do not rest at 0. A good target is to have it sitting greater than the resting signal that occurs when the joystick is at neutral or when an operator wearing EMG sensors is not contracting their muscle.

g. <u>Smax</u> → the maximum threshold above which the actuator will not move any faster than its maximum joint velocity. Typically this is left at a value of 5, but in some cases it can be decreased as an alternative to increasing the gain in order to improve the proportional range of the mapping.

h. Here is an example of a mapping from two EMG signals to the hand close/hand open movements on the Bento Arm. In this example since Ch1 is mapped to Hand Close and its signal strength is above its Smin the hand would be closing at about 33% of its maximum speed. Also, notice how the Smin value is set above the resting value for Ch5 in order to prevent unintentional movements.
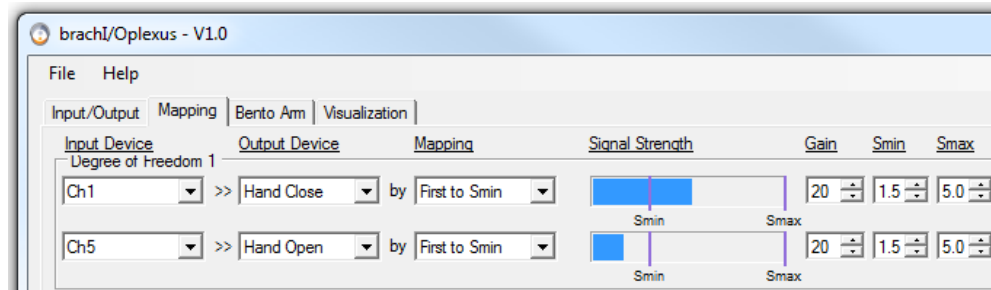
7. In the mapping tab you can also set up a sequential switching list that uses a trigger signal to sequentially switch between different robot movements that are mapped to a single pair of input signals. This is most useful when you have limited discrete input signals measured by EMG sensors and are trying to do direct mappings (aka conventional EMG control) that normally only allow you to control a single DoF on the robotic arm at a time. Enabling the sequential switch allows you to control up to 5 DoF sequentially using just two EMG channels from the MYO armband. In the example below the output device mapped to Degree of Freedom 1 would cycle from Shoulder to Elbow to Wrist Rotate to Wrist Flex to Hand to Shoulder each time the channel pair in Degree of Freedom 1 are co-contracted.
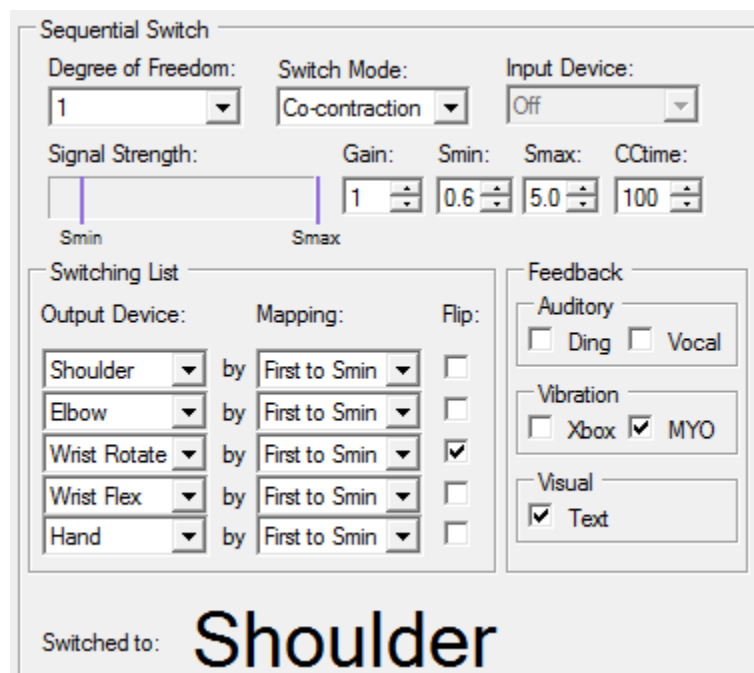
a. <u>Degree of Freedom</u> →The pair of channels or buttons that are being used to move a joint on the robot. In most of the preset profiles this is set to the first two channels in Degree of Freedom 1.

b. <u>Switch Mode</u> → The method used for switching between which joint the channel pair are mapped to.

    i. <u>Button press</u> → The sequential switching is triggered with a button press as soon as it crosses the minimum threshold (Smin).

    ii. <u>Co-contraction</u> → The sequential switching is triggered by activating both of the signals in the channel pair so that they cross their respective maximum thresholds within a certain time window (see CCtime) after one of them has crossed their respective minimum threshold.

c. <u>Input Device</u> → When the 'Button Press' mode is selected this is where you select the button or channel that you want to use for triggering.

d. <u>Signal Strength, Gain, Smin, Smax</u> → These are the signal parameters for the input device selected in c. and have the same purpose as described in 6. d. through g.. They are only relevant when the 'Switch Mode' is set to 'Button Press'.

e. <u>CCtime</u> → This is the amount of time in milliseconds (usually 100-200ms) that the co-contraction algorithm waits after a signal has reached minimum threshold to decide whether a joint should be moved or a switching event should occur. If both signals rise up and exceed their maximum thresholds sometime during CCtime then a switching event will immediately occur and the output device will cycle to the next joint on the switching list. If the other signal does not rise up then no switching event will occur and the arm will move the currently active joint.

f. <u>Output Device</u> → These are the output movements in the switching list that will be sequentially cycled through (also known as the "switching list").

g. <u>Mapping</u> → The type of mapping for each output device. The only option in this release is 'First to Smin' which was previously described in 6.c.

h. <u>Flip</u> → If the direction of movement is backwards you can use this checkbox to flip it the other way.

i. <u>Feedback</u> → Feedback signals can be enabled or disabled to provide information about when the output device has switched and to what output device it has switched to.

    i. <u>Ding</u> → Makes a ding noise every time the switching signal is triggered.

    ii. <u>Vocal</u> → Plays an audio file verbally naming the corresponding output device as it cycles through the switching list.

    iii. <u>Xbox</u> → Briefly activates the vibrating motor on the Xbox controller every time a switching signal is triggered.

    iv. <u>MYO</u> → Briefly activates the vibrating motor on the MYO armband every time a switching signal is triggered.

v. <u>Text</u> → Visually displays the output device that is currently active in the switching list at the bottom of the groupbox beside the label 'Switched to'. In the example above the current output device is 'Shoulder'.

8. Profiles
   a. Once you have created a custom mapping you can save it for future use by going 'File → Save Profile As' and then entering descriptive profile name in the dialog box.
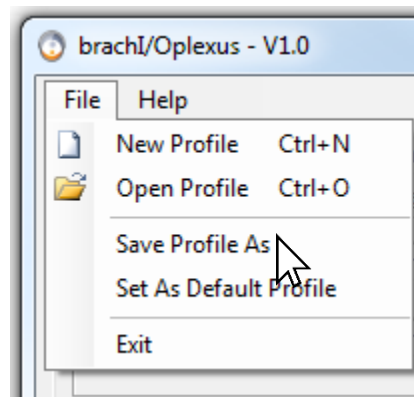

   b. You can also load custom mappings when you first open the program or connect to devices by going 'File → Open Profile'.
   c. If there is a particular mapping that you use most frequently you can have it load it at startup by going 'File → Set As Default Profile'.
   d. You can load an empty profile by going 'File → New Profile'.
   e. We have created a number of common mappings that you can use as starting points. Some of these mappings also have accompanying diagrams that can be accessed through the 'Help' menu at the top of the screen by selecting 'Mapping Diagrams'.
      i. <u>Default</u> – the default profile that is loaded on startup
      ii. <u>Empty</u> – an empty profile
      iii. <u>Kb_multi</u> – multi joint control of Bento Arm using the keyboard
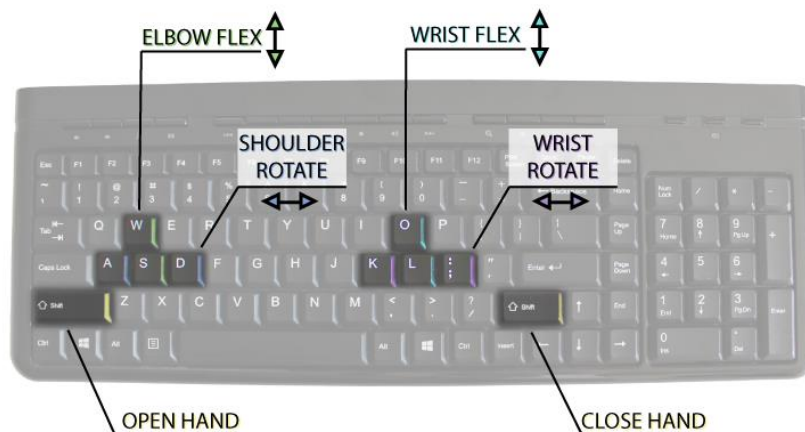
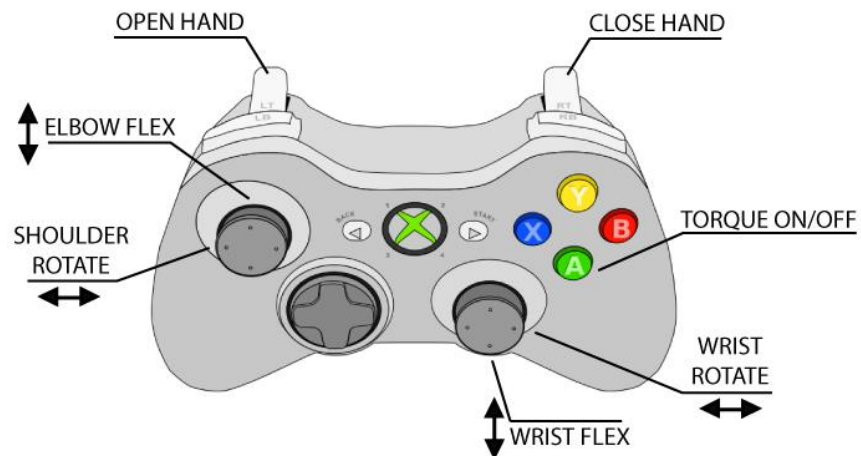iv. Kb_sequential – sequential control of Bento Arm using the keyboard.
v. MYO_sequential_left – sequential control of the Bento Arm using the muscle signals from the MYO armband on the left arm.
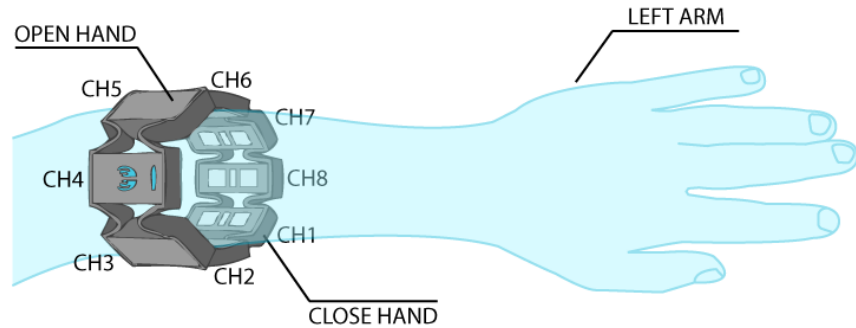

vi. MYO_sequential_right – sequential control of the Bento Arm using the muscle signals from the MYO armband on the right arm.
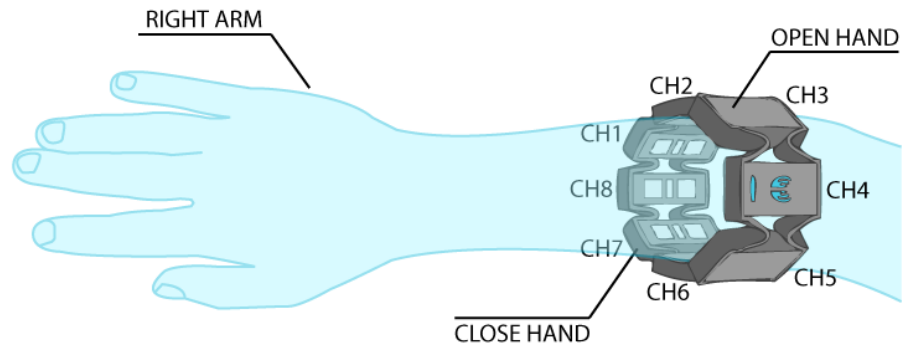
vii. Xbox_multi - multi joint control of Bento Arm using an Xbox controller.

viii. <u>Xbox_sequential</u> – sequential control of the Bento Arm using an Xbox controller.

    f. NOTE: If you uninstall brachI/Oplexus make sure to make a backup of any profiles you have saved or changed as they will be removed with the program.

9. You can also customize the display of the mapping tab by hiding individual degrees of freedom or the sequential switching groupbox. This feature is useful if you are only using some of the features and want to simplify the display to only show the elements in use. To hide a degree of freedom in the mapping tab you can right click on it and then to make it appear again you can left or right click in the area where it used to be.

10. Click on the 'Bento Arm' tab in order to get the arm moving and set up joint limits.
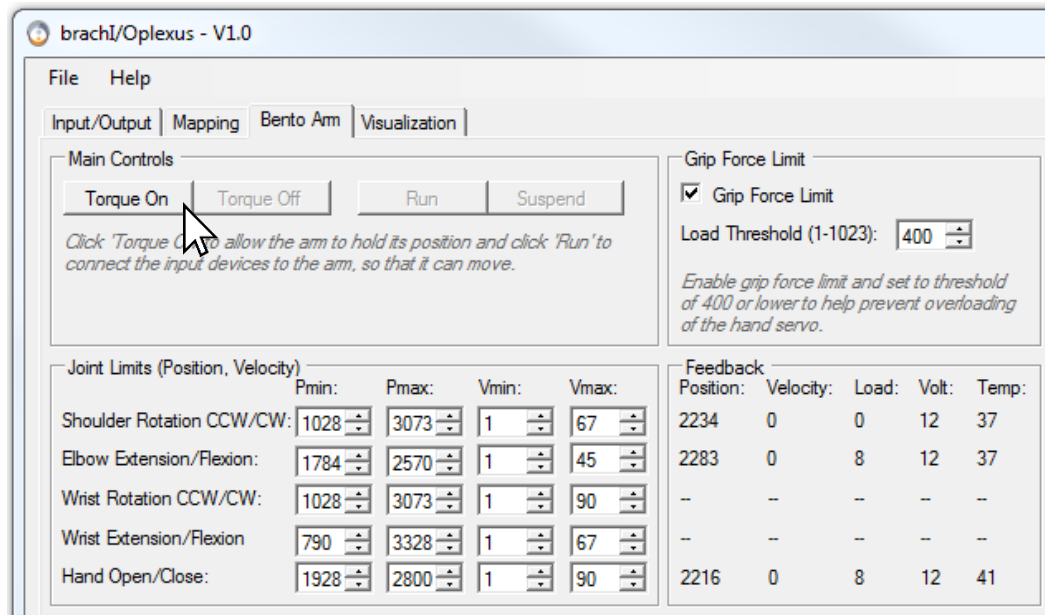
    a. With the arm in the "*Torque Off*" state passively position the Bento Arm into the desired starting position and then click the 'Torque on' button to allow the Bento Arm to hold its position. NOTE: you should never try to passively move the arm while it is in the "*Torque on*" state as it can overload or damage the actuators.

    b. After clicking 'Torque on' the Bento Arm will default into its 'Run' state that will connect the input devices to the arm so that it will be able to move immediately.

    c. If you want to passively move or pose the Bento Arm, first place your hand under the gripper to support it and then click the 'Torque Off' button. Be sure to support the arm before pressing this button as otherwise it could fall into whatever surface is below it and be damaged by impact.

Passively move the arm to the new desired position and then click 'Torque On' again.

d. If you want to disconnect the input devices from the Bento Arm click the 'Suspend' button. The Bento Arm will not be able to move while it is in the "*Suspend*" state, but will still hold its current position with the torque on. This is useful if a control input is giving errant signals and you want to temporarily disable the arm from moving without having to hit the power switch or turn the torque off which would cause the arm to drop to the table.

e. NOTE: For safety reasons the Bento Arm will auto-suspend in certain situations such as when loading profiles, changing input devices, or enabling/disabling sequential switching. After you have ensured that none of the signals are maxed out you can click the 'Run' button to set the Bento Arm back to its "*Run*" state, so that it can move again.

f. The 'Grip force limit' for the gripper can be enabled and set to a threshold of 400 or lower to help prevent overloading of the hand servo. A setting of 400 is equivalent to a grip force of about 5 or 6 N on the chopsticks gripper. This setting is enabled by default.

g. The joint limits of the Bento Arm can be fine-tuned using the controls in the 'Joint Limits' group box. NOTE: You will be able to adjust the joint limits to values within the limits programmed into their registers. If you want to be able to move outside of the register limits you will need to use the Dynamixel Wizard software as described in section 1.7 page 16, of the Bento Arm Assembly Guide. Adjusting the register limits is not generally recommended because it can result in the Bento Arm hitting itself or the stand or moving at unsafe speeds.
    i. Pmin → The position limit in the CW direction
    ii. Pmax → The position limit in the CCW direction
    iii. Vmin → The minimum velocity that the actuator will be able to move once the signal passes the minimum threshold Smin. NOTE: the lowest possible speed is 1.
    iv. Vmax → The maximum velocity that the actuator will be able to move once the signal equals or exceeds the maximum threshold Smax. NOTE: this value is limited to 200 for safety reasons.

h. Feedback from each actuator on the Bento Arm is provided in the 'Feedback' group box
    i. Position → the present position of the actuator (should be between Pmin and Pmax)
    ii. Velocity → the present velocity of the actuator (negative velocities are in the CW direction and positive velocities are in the CCW direction.
    iii. Load → the present load on the actuator (varies from 0-1023). If the load exceeds 400 for a duration of about 10 seconds then the actuator will overload and automatically shut down.
    iv. Volt → The present voltage that the actuator is seeing in volts. Should normally read 12 volts

v. <u>Temp</u> → The present temperature of the actuator in degrees. If the temperature exceeds 70 degrees the actuator will overload and automatically shut down.

i. The status bar shows the present state of the Bento Arm and has a button for quickly switching between the Run/Suspend state from any tab. Error messages are also displayed if a servo overloads or overheats

Bento Status: Torque On / Running   Suspend   Error - ID5 is overloaded
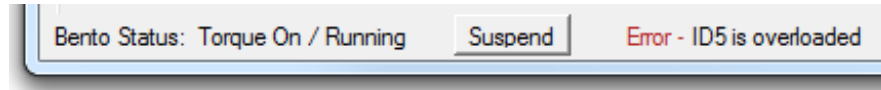
j. If a Dynamixel actuator overloads or overheats while you are using the Bento Arm it will automatically go into a shutdown state where it turns off the torque. You will know it is in this state because of an error message in the status bar and also the LED on the actuator will flash red. To re-enable the actuator we recommend the following procedure:

    i. Ensure you have removed the cause of the overload.

    ii. Place your hand under the gripper to support the arm and then click the 'Torque Off' button and lay it gently on the table.

    iii. Unplug the USB2dynamixel from the power harness.

    iv. Cycle the power to the actuators by unplugging them from the power harness and then plugging them back in.

    v. Plug the USB2dynamixel back into the power harness.

    vi. Pose the Bento Arm in the desired position and click the 'Torque On' button.

    vii. The Bento Arm can now move again!

    viii. NOTE: If the actuator has overheated let it cool down to about 40-50 deg C before re-enabling it.

11. When you want to shutdown the program you can click the 'x' in the top right corner or go 'file → exit'. We recommend supporting the arm and clicking the 'Torque Off' button before closing the program otherwise it will continue to hold torque even after the program is closed.

12. The details for disconnecting the Bento Arm are covered in section 1.4. As a reference here is the recommended disconnection order:

    e. Disconnect the USB2dynamixel to the Actuators via the power harness.

    f. Disconnect the power to the Actuators via the power harness or if you have an electronics enclosure press the power button off

    g. Disconnect the USB2dynamixel to the computer via the USB extender cord

13. A link to this help manual is also available while using the program by clicking 'Help' and then 'User Manual'

1.6 Modifying the Software

This section describes how you can make changes to the software such as adding additional controllers and robots or modifying the mappings.

1.6.1 Development Environment

The main development environment used in this project was Visual Studio Express 2015, but the project was also able to load successfully in Visual Studio 2010. Visual Studio Express 2015 for Windows Desktop can be freely downloaded by individual developers, open source contributors, academic researchers, and small organizations from the visual studio website: https://www.visualstudio.com/vs/visual-studio-express/

As a first step you will need to download visual studio express in order to view or make changes to the C# project files.

1.6.2 Libraries and Interfaces

The brachI/Oplexus software uses the following open source libraries and interfaces to communicate with input and output devices. Consider reviewing their documentation if you want to learn more about how these interfaces work:

- XInputDotNet
  - C# wrapper around XInput, works with any Mono or .NET application (eg. Unity3D)
  - Used in brachI/Oplexus to communicate with Xbox controllers
  - https://github.com/speps/XInputDotNet
- DynamixelSDK
  - Software development library that builds control and feedback packets to communicate with Dynamixel Actuators
  - Used in brachI/Oplexus to communicate with the Bento Arm
  - https://github.com/ROBOTIS-GIT/DynamixelSDK
- MyoSharp
  - C# wrapper for the MYO Armband
  - Used in brachI/Oplexus to communicate with the MYO armband.
  - https://github.com/tayfuzun/MyoSharp
- Simple Moving Average Algorithm
  - A moving average algorithm implemented in C# with a circular list
  - Used in brachI/Oplexus to help smooth the EMG signals from the MYO armband. The variable called 'window' in the initialization region can be increased to improve smoothing with the trade-off that it will also create a time delay.
  - https://www.codeproject.com/Articles/17860/A-Simple-Moving-Average-Algorithm

Note: If you have already followed the installation instructions for brachI/Oplexus you will not need to download these libraries or install additional drivers in order to run the source code
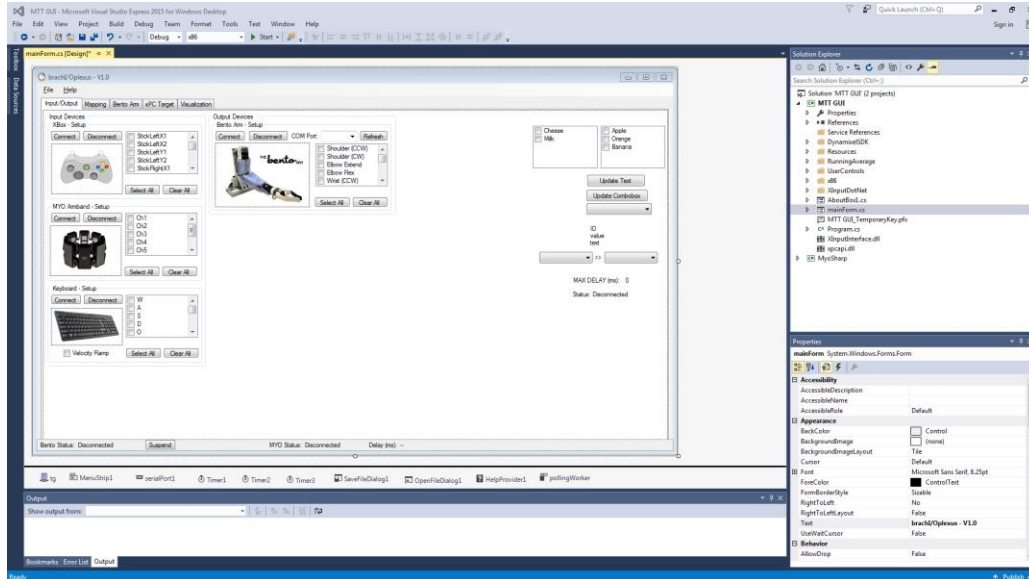
1.6.3 Source code

1. To download the source code please visit our github page and click the 'clone or download' button to download the latest master branch:
   https://github.com/blincdev/brachIOplexus

2. Before extracting the files you should unblock the downloaded file by right clicking on 'brachIOplexus-master.zip' and then selecting 'properties'. The properties window should open and then in the 'General' tab you should make sure 'Unblock' is checked as seen in the following image and then click the 'Apply' and 'OK' buttons. The file should now be unblocked. If you skip this step, the solution will still open, but you may not be able to load the designer view.
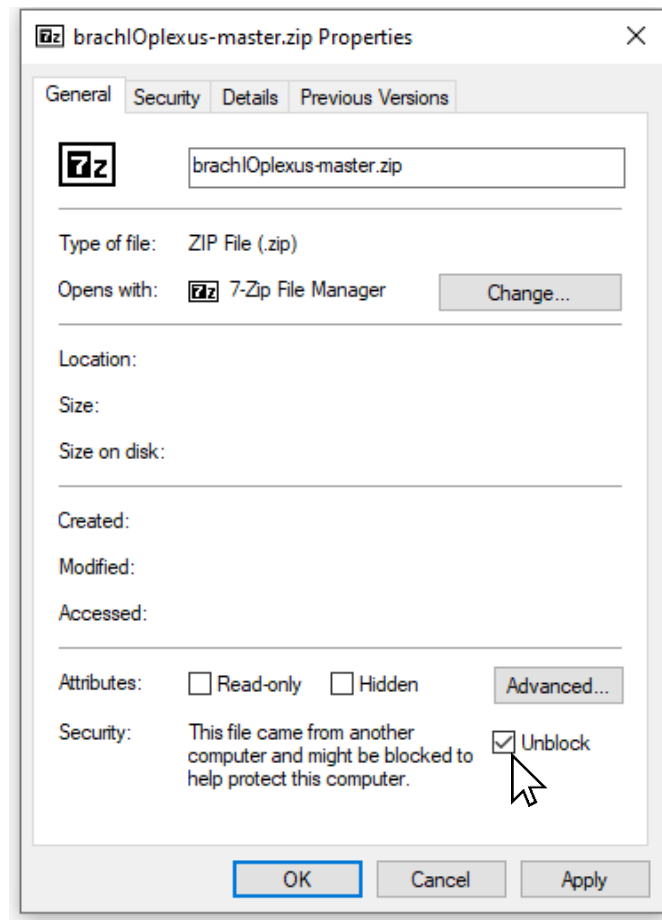
3. Extract the files to a directory and then double click on 'MTT GUI.sln' to open the C# project in Visual Studio Express 2015. NOTE: Do not place the folder with the software in a filepath that contains any unusual characters (i.e. underscores or spaces seem okay, but dashes or tilds in the file path will make it so the designer view will not open properly). Also, it works best if you double click on the solution file in windows explorer to open visual studio and the project rather than opening visual studio and then trying to load the project.

4. The project should open and look similar to the following image:

5. If the 'mainForm.cs' tabs are not already open you can right click on mainForm.cs in the 'Solution Explorer' and select 'View code' to view the source code. To view the graphical user interface right click again on mainForm.cs and select 'View Design'.
6. To run the program you can click on the 'Start' button at the top of the screen that has a big green play button beside it.
7. If you are interested in contributing to the development of the program or have ideas for future features please contact us through our website: https://blincdev.ca