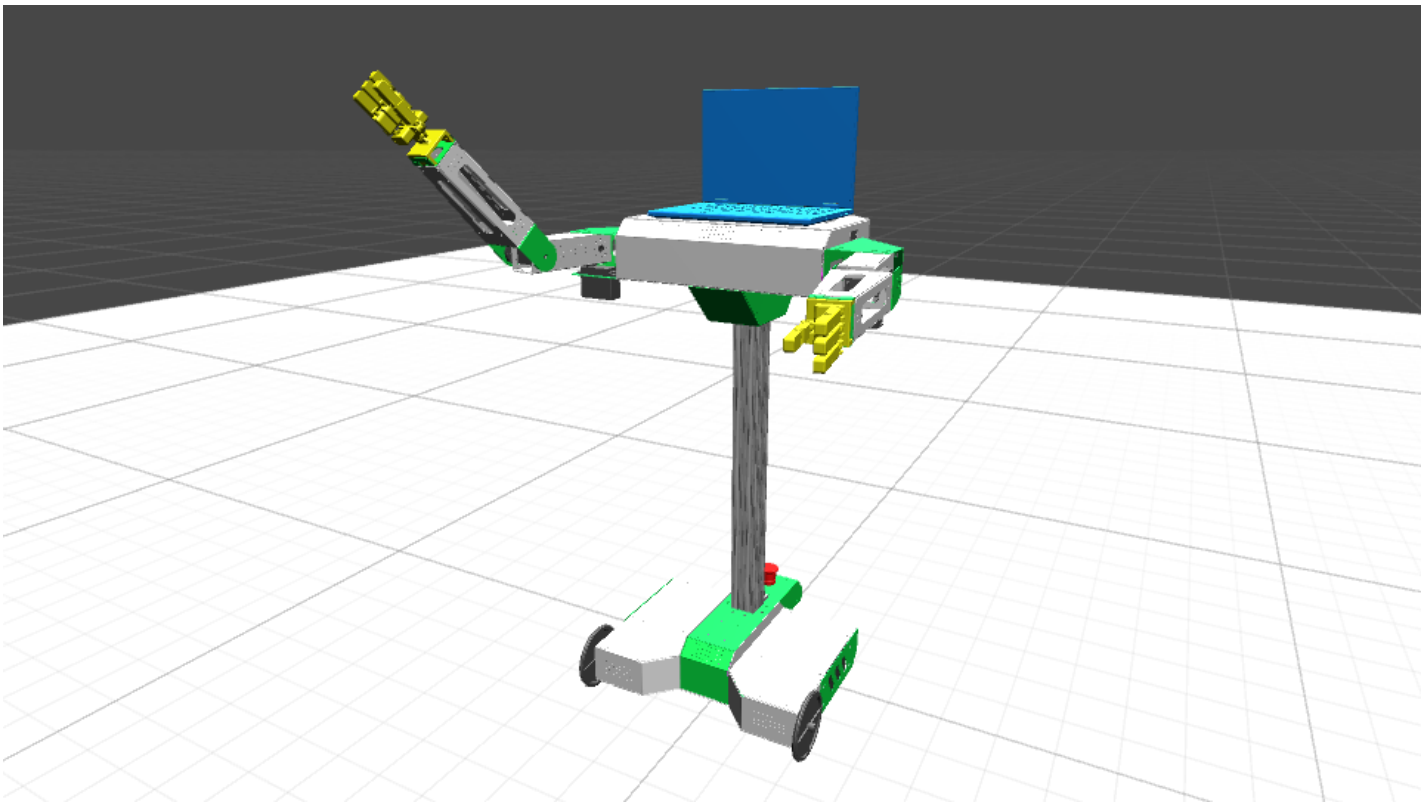
Megamark Unity Library Documentation

The Choitek Megamark is an advanced full-size multipurpose mobile manipulator robotics platform for students, artists, educators and researchers alike. In our mission to make the platform as open, friendly and compatible as possible, we've made the platform compatible with a variety of different languages and frameworks. This document covers all the commands of the Megamark Unity framework, located in the `Megamark.cs` file.



Using the Megamark Library in a Unity Project

To use the Megamark Library, import the `UnityMegamark.unitypackage` into your Unity project. If the import fails for whatever reason, open the provided Unity Megamark Example project and copy the Megamark Controller folder into your Unity project. Make sure you have `.NET 2.0` enabled instead of `.NET 2.0 Subset` in `File->Build Settings->Player Settings->Other Settings`. Now, drag and drop the Megamark.prefab in the Prefabs folder from Project Assets into your scene.

And voila! You should now be able to control all the actuators of the Choitek Megamark robot with Unity.

Note that you will have to change `'COM3'` to your COM Port. This is something like "/dev/ttyACM0" on Linux and Mac. If you don't know what your COM Port is, plug in the USB to the Arduino Mega 2560 and follow these instructions: https://www.arduino.cc/en/Guide/Troubleshooting#toc1

List of Commands in the Megamark Library

The following is a list of all default commands that can be sent to the Megamark robot in Unity:

- `void reset()`
- `void setWheelVelocities(float leftSpeed, float rightSpeed)`
- `void rotateLeftShoulder(float angle)`
- `void rotateRightShoulder(float angle)`
- `void rotateLeftElbow(float angle)`
- `void rotateRightElbow(float angle)`
- `setLeftGrippers(l1,l2,l3,l4,l5,l6)`
- `setRightGrippers(r1,r2,r3,r4,r5,r6)`
- `int getLeftLaser()`
- `int getRightLaser()`
- `void exit()`

Note there are a few additional commands that are hidden in the `Megamark.py` file for reading, writing and formatting data for communication with Megamark robots. We recommend not modifying them if you are a beginner as they are automatically handled by the Megamark library in real time.

`reset()`

This command does not take any arguments, and resets the robot's pose to all default positions. This means wheel velocities are set to zero, shoulders are set to 90°, elbows are set to 0°, and all grippers are set to 90°.

`setWheelVelocities(float leftSpeed, float rightSpeed)`

This takes two arguments, one to control the speed of the robot's left wheel and one to control the speed of the robot's right wheel. The Megamark robot's wheels have a 5-inch (127 mm) diameter and are spaced 17.2-inches (437 mm) laterally. That maximum forward speed of the robot's wheels are 0.2 meters/second, and have a maximum backward speed of -0.2 meters/second. To make the robot go forward, set both velocities positive. To make the robot go backward, set both velocities negative. To make the robot turn left, set the left wheel negative and the right wheel positive. To make the robot turn right, set the left wheel positive and the right wheel negative.

`rotateLeftShoulder(float angle)`

This takes one argument to control the angle of the robot's left shoulder. This ranges from 0 to 120 degrees (the default is straight out at 90 degrees).

`rotateRightShoulder(float angle)`

This takes one argument to control the angle of the robot's right shoulder. This ranges from 0 to 120 degrees (the default is straight out at 90 degrees).

`rotateLeftElbow(float angle)`

This takes one argument to control the angle of the robot's left elbow. This ranges from -60 to 60 degrees (the default is straight out at 0 degrees).

`rotateRightElbow(float angle)`

This takes one argument to control the angle of the robot's right elbow. This ranges from -60 to 60 degrees (the default is straight out at 0 degrees).

`setLeftGrippers(l0,l1,l2,l3,l4,l5)`

This takes anywhere from 1 to 6 arguments to control up to 6 servos on the robot's left gripper respectively, zero-indexed. For example, `l0` corresponds to left servo 1, `l1` corresponds to left servo 2, and so forth. Each servo ranges from 0 to 180 degrees (the default is straight out at 90 degrees).

`setRightGrippers(r0,r1,r2,r3,r4,r5)`

This takes anywhere from 1 to 6 arguments to control up to 6 servos on the robot's right gripper respectively, zero-indexed. For example, `r0` corresponds to right servo 1, `r1` corresponds to right servo 2, and so forth. Each servo ranges from 0 to 180 degrees (the default is straight out at 90 degrees).

`getLeftLaser()`

This takes in zero arguments and returns the value of the robot's left laser rangefinder (normally in millimeters, though this will vary depending on the exact kind of laser rangefinder used). Note that a laser rangefinder must be attached to the robot's left rangefinder pin, otherwise unusable data will be returned.

`getRightLaser()`

This takes in zero arguments and returns the value of the robot's right laser rangefinder (normally in millimeters, though this will vary depending on the exact kind of laser rangefinder used). Note that a laser rangefinder must be attached to the robot's right rangefinder pin, otherwise unusable data will be returned.

`Disconnect()`

This command destroys the connection to the Megamark robot if it is currently connected. We recommend using this command before you quit your program to ensure any allocated data is properly cleared.