THE PHANTOMIX PINCHER ROBOTIC ARM

TECHNICAL SURVEY

FEBRUARY 27, 2017

ALI Mohamed
Danie Jianah SONIZARA

Contents

INTRODUCTION

Is there life without hands? Yes, certainly, but it would look very different if we didn't have hands. A hand is the main tool used in manipulation of objects with various sizes and different weights. Human structure is the largest and the best source of inspiration for researchers and engineers in robotics, as they are trying to create a universal robotic arm to be used for most tasks, with a hand that could not look or work differently than the human hand.

To build the perfect robotic arm researchers and engineers replace bones, tendons, muscles and nerves with metal, plastic and electricity. In addition to all these things, engineers is building a mechanism with the same dimensions, proportions and senses. If replication of the arm is almost complete, there is still a long way for researchers and engineers to solve the interaction between brain and arm for a robot.

Nowadays, robotic arms are employed in a large scale design generally to replicate a human arm especially in an environment not suited for humans which have a different variety commercially available. Some of them are excellent in accuracy and repeatability. In this paper, we do a survey on a robotic arm and described the different parameters in order to set it up. It will give you a technical introduction to the PhantomX Pincher robotic arm.
To do so, as a first step we will see a general overview of the robotic arm following by the overview of the PhantomX Pincher Arm.
In a second time, we will see how to assemble, to test and to debug the robotic arm.
And finally before developing some applications, we will show how to interface the robotic arm with ROS.

I. General overview of the robotic arm

1. Description

The goal of this survey is to the study of this robotic arm in order to understand its operation and to get familiar with.

Robot arms have been developing since 1960's, and those are widely used in industrial factories such as welding, painting, assembly, transportation, etc. Nowadays, the robot arms are indispensable for automation of factories. Moreover, applications of the robot arms are not limited to the industrial factory but expanded to living space or outer space. The robot arm is an integrated technology, and its technological elements are actuators, sensors, mechanism, control and system, etc.

A Robot arm is known manipulator, in Robotics a manipulator is a device used to manipulate materials without direct contact. The applications were originally for dealing with material or situation which would be dangerous for human to work directly with like Radioactive or Biological hazards materials, using robotic arms, or they were used in inaccessible places.

In more recent developments they have been used in diverse range of applications including welding automation, robotically-assisted surgery and in space. It is an arm-like mechanism that consists of a series of segments, usually sliding or jointed called cross-slides, which grasp and move objects with a number of degrees of freedom. The set of joints separated in space by the arm links. The joints are where the motion in the arm occurs. In basic, a robot arm consists of the parts: base, joints, links, and a gripper. The base is the basic part over the Arm, It may be fix or active. The joint is flexible and joins two separated links. The link is fix and supports the gripper. The last part is a gripper. The gripper is used to hold and move the objects. Figure-1 shows these parts for the 5 DOF Robot arm we use in this report.

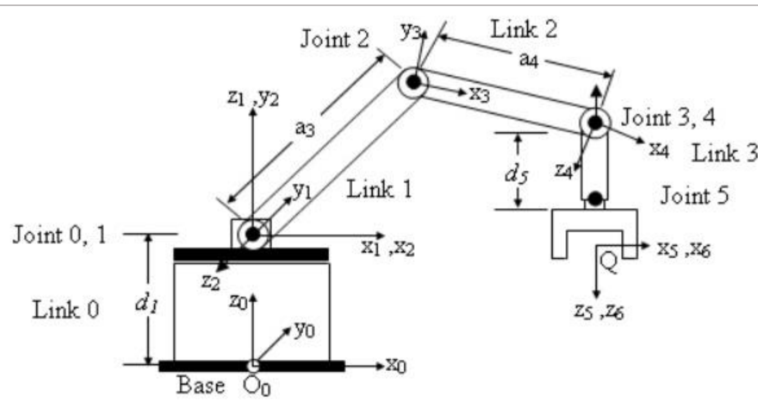


Figure 1: Basic robot arm 5DOF with 1 Base, 4 Joints, 3 Links and the last part a gripper.

2. Characteristics

2.1. Axis

Axis are used for movement indication, one use for a line, two for a plane and three for a point at anywhere in space. Roll pitch and yaw control are the main factors of a robotic arm axis, use for full control. Figure 2 shows an exemple of a six-axis robotic arm

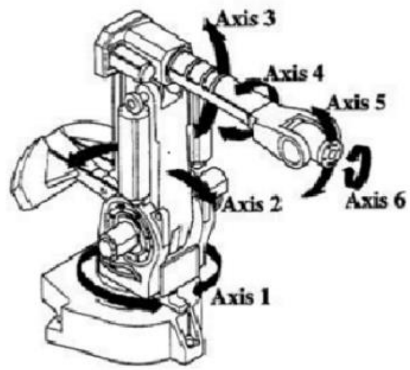


Figure 2: A six-axis-robotic arm

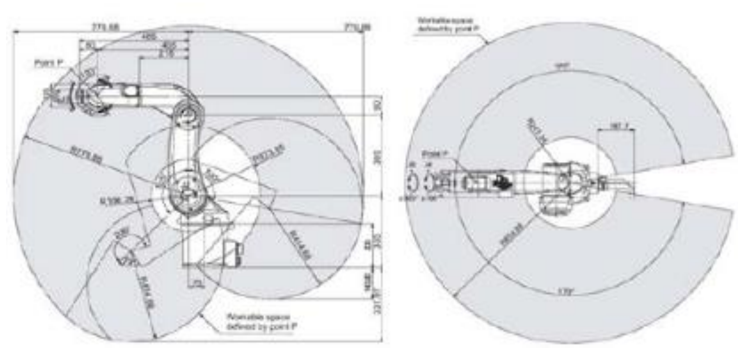


Figure 3: Side dimension working envelope and top dimension working envelope

2.2. Degree of freedom

A primary criteria that defines a robotic arm is the number of degrees of freedom (DOF), each degree of freedom representing a motion a robot can perform. For a robotic arm to perform a motion it requires a joint which allows either rotational or translational movement, and of course an actuator to power that section, therefore the number of DOF can be easily determined by establishing the number of actuators in a robotic arm. The more DOF the better, as the robotic arm can perform a greater range of movements.

2.3. Working space

The robotic arm can cover circle in space is a working envelope of a robotic arm. Envelope means the range can cover by arm or range of movement. If we use the arm in different possible directions (forward and backward, up and down) then one 3D shape is an envelope, it depend on a robotic arm and number of axis, the axis can manipulate the range of motion. Figure 3 shows Side dimension working envelope and top dimension working envelope. A robot can only work in their working envelope.

2.4. Kinematic

Robot kinematics is use for finding the movement of multi-axis and multi-degree of freedom. In the kinematic analysis of manipulator position, there are two separate problems to solve direct kinematics and inverse kinematics. Direct kinematics involves solving the forward transformation equation to find the location of the hand in terms of the angles and displacements between the links. Inverse kinematics involves solving the inverse transformation equation to find the relationships between the links of the manipulator from the location of the hand in space.

II. Overview of the PhantomX Pincher Arm

1. Description

The PhantomX Pincher Arm is a programmable robotic arm designed by Trossen Robotics. This full robotic kit allows you to grab, to move small objects located in its area.

It is a **5DOF (5 Degrees of Freedom)** robotic arm consisting of 5 joints allowing rotation, lifting or displacement movements.

The PhantomX Pincher robotic arm must therefore be equipped with 5 Dynamixel AX-12A servomotors. Please note that if the Trossen Robotics KIT-RK-PINCHER robotic arm is programmable via an open-source ROS library, this robotic kit is only a hardware platform with no software.

The PhantomX Pincher Arm can be used as a standalone robotic arm platform, or as an addition to the Turtlebot ROS robot platform. It is a tool especially used for education or for the creation of robotic projects.

The PhantomX Pincher is also designed to complement the TurtleBot and TurtleBot 2 robotic kits, enabling you to add a new possibility for interaction with your environment to your programmable robot.

Figure 4: The PhantomX Pincher Robot arm

2. Technical specifications

The complete arm has a weight of 550 grams and has a horizontal reach of 28 centimeters. Vertically fully extended it has a reaching distance of 21 centimeters and comes equipped with a parallel gripper. The robot has a sturdy ABS plastic structure and is also equipped with bearings for its rotating base and various mounting brackets for accessories, sensors, etc. The gripper has a 500 gram holding strength while the wrist has a lifting strength of 250 grams. The robotic arm itself can hold up to 100 grams, 40 grams when fully extended.

Technical specifications of the PhantomX Pincher robotic arm

Weight	550 g
Vertical Reach	35 cm
Horizontal reach	21 cm
Gripper Strength	500 g (holding)
Wrist Lift Strength	250 g
	25 cm / 40 g
Strength	20 cm / 70 g
	15 cm / 100g

The robot requires a 12 VDC 2 A power supply, included with the kit. Also included is a FTDI programming cable and interestingly enough, a bottle of Turbo-Lock thread locker to prevent parasitic movements and vibrations of the arm. This may be a good addition for any robot employing screws and nuts for fixtures.

Product Features	**Kit includes**
- AX-12A Dynamixel Actuactuors - Solid Needle Roller Bearing Base - Rugged ABS construction - Arbotix Robocontroller for Onboard Processing - Custom Parallel Gripper - Mounting Brackets for cameras & sensors	- 5xAX-12A Dynamixel Actuactuors - PhanthomX Pincher arm Hardware & Frame Kit - Arbotix Robocontroller - 12V 5amp power supply - FTDI 5V Programming cable - 2ml Bottle of Turbot-Lock

Figure 5: AX-12A Dynamixel Actuactuors

Figure 6: Arbotix Robocontroller

The robot is controlled by an ArbotiX controller board, which is designed for controlling the advanced AX-12A Dynamixel actuators. There is no bundled or optional software for this robotic arm kit.

III. Setting up the arm

1. Install Software

The PhantomX Pincher robotic arm currently does not come with supported software so it is important to download and install the following software before assembling the arm.

1.1.Arduino IDE

The Arduino IDE (Integrated Developer Environment) is an application that you can use to program and interact with Arduino based microcontrollers like the ArbotiX-M.

Here we will download the Arduino 1.0.6 in which the ArbotiX-M libraries have been thoroughly tested. In our case we download the Arduino for Linux 64bits. You can find the different Arduino IDE 1.0.6 compatible to your computer here.

1.2.FTDI drivers

Now you will need to install FTDI drivers. These drives will allow your FTDI-USB cable function properly. Some modern Operating Systems either have these drivers or can automatically find them. We personally didn’t install it because the FTDI drivers were already installed in UBUNTU. If you don't have the drivers or you are unsure, you can find the FTDI drivers here, and a guide to installing them here.

Then connect the Arbotix Robocontroller to the computer.

The orientation of the FTDI cable is very important. So please be carefull and follow the example in the figure 7.

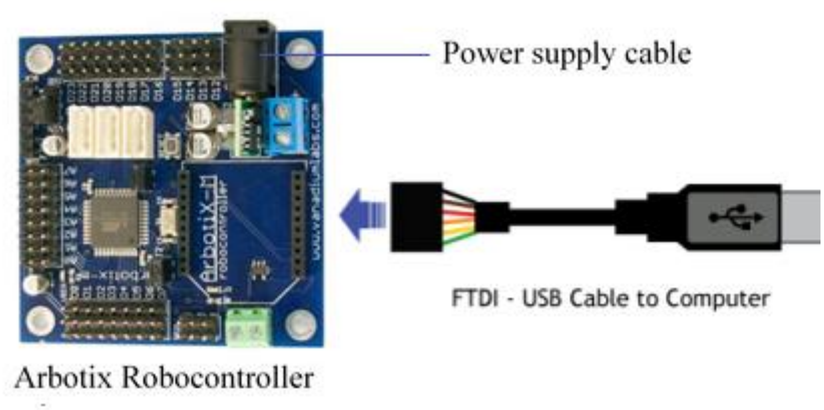


Figure 7: Connecting the Arbotix Robocontroller to the computer.

1.3.Arbotix libraries and Hardware files

For the Arduino IDE to support the ArbotiX-M, we'll need to add files into the Arduino user folder. Click here to download the needed ArbotiX-M Library and Hardware Files. In this .zip file, there will be 3 folders named *Hardware, libraries and ArbotiX Sketches* that we have to move inside the 'Arduino' user folder.

Once the installation finished you have to program the Arbotix-M robocontroller to blink by picking the Arbotix board from the boards menu.

Select the proper board:

```
Tools->Board ->ArbotiX
```

Now pick the serial port. Go to:

```
Tools ->Serial Port
```

And pick the serial port for the FTDI device.

Once you have set the board and serial port, you can open the 'ArbotixBlink' sketch.

```
File -> Sketchbook -> ArbotiX Sketches -> Test Sketchs -> ArbotiXBlink
```

When the Arduino IDE displays 'Done Uploading' the user LED should blink on and off in a 1 second interval. This shows that the ArbotiX-M Board is programmed.

1.4. JAVA

We need to install Java to make the DynaManager software run. If you do not have Java installed on your system, it is available here.

1.5.DynaManager

Once Java installed, you will need to download the DynaManager Software in order to ID the servomotor

For the DYNAManager to work, you will need the ROS.ino sketch loaded onto your ArbotiX. New ArbotiX boards come pre-loaded with the ROS sketch. If you have already loaded another program onto your ArbotiX, you can find the ROS sketch here.

```
File -> Sketchbook -> ArbotiX Sketches -> ROS
```

Before you set your servo IDs, here are some things to keep in mind:

- You can only ID one servo at a time.
- Do not plug multiple servos into the ArbotiX when using the DYNAManger.
- Make sure you label your servos as you ID them! This will make the assmebly phase go much faster

To set up the ID, it is necessary to connect the Dynamixel Actuactors to robocontroller as it shows the figure below.

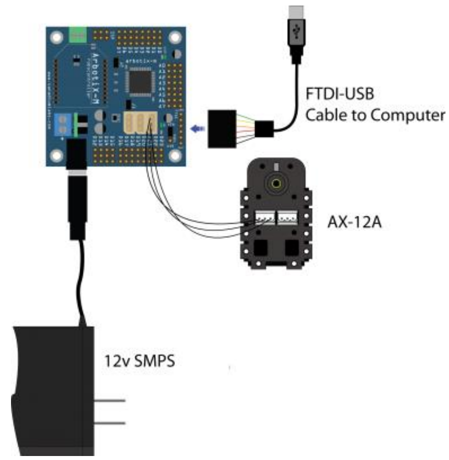
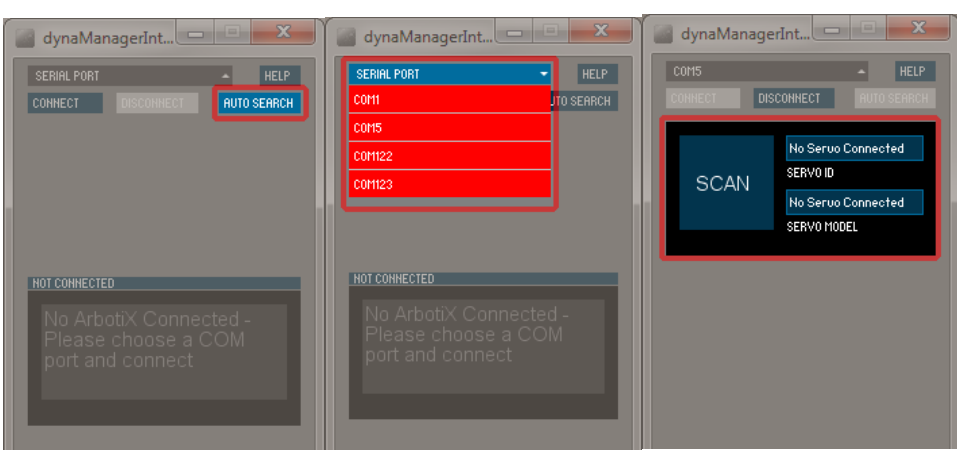


Figure 8: connecting the Dynamixel Actuactuors

Once the connection between the robocontroller and the dynamyxel actuactuor done, set up the servo motor by using DynaManager.

- Launch the dynamanager.
- The easiest way to connect to the ArbotiX is to lick on 'Auto Search'. This will scan all of your available serial ports for an active ArbotiX running ROS.
- If you have problems with the Auto Search Function, manually pick the serial port that corresponds to your FTDI device, and hit 'connect'.
- Once connected to the ArbotiX, the Servo Scan panel will appear.
- Click the 'Scan' button to begin the servo scan. While scanning the 'Scan' button will change color and the text 'Scanning' will show in the Servo ID field. New AX servos should be found almost instantly, and new MX servos will take a 5-10 seconds to find. The scanner will scan from IDs 0-252 at both 1000000 baud and 57600 baud. A full scan takes about a minute. If the scan completes and your servo is not found, make sure that your servo is powered and connected to the ArbotiX.



1 Once a servo is found, the servo's ID and model number will appear. The 'Set Servo ID' and 'Test Servo' panels will now appear.

2 Enter in the ID that you would like to set the servo to. In the assembly guide for your robot, you will find a diagram of which servo should get which ID.

3 Click the 'Set ID/Baud' button. Note: New servos come pre-set to an ID of '1'. However it is still a good idea to 'Set' your first servo to an ID of 1, to make sure that your Baud rate is set correctly.

4 The 'Servo Scan' panel should now update to reflect your newly ID'ed servo (if it does not, click 'Scan' again. Now you can use the 'Test Servo' panel to move your servo horn to make sure it is working properly. Click and hold anywhere along the blue ring and the servo should move to that position.

5 Once you are done, click the 'Center Servo' button to set the servo horn back to its centered position. This is very important, as the assembly guide for your robot will assume your servo horn is centered. Assembling your robot with mis-aligned horns will cause it to malfunction.

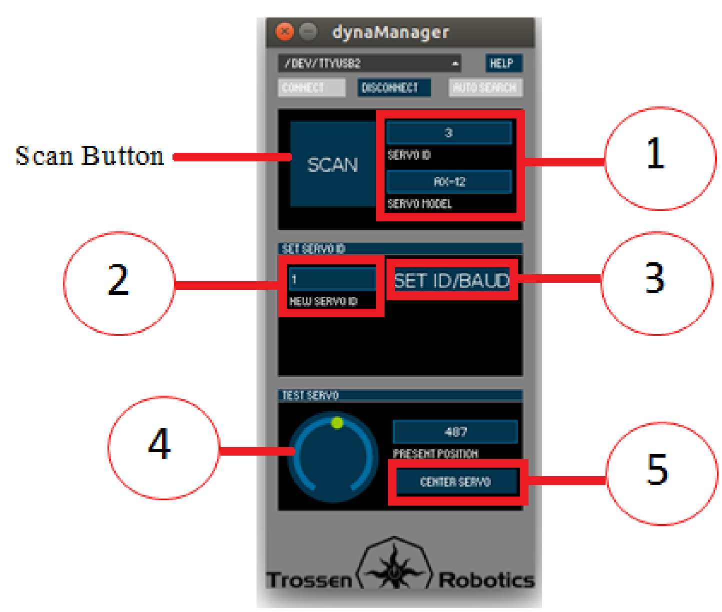


Figure 10: Setting up the ID

2. Assembling the robotic arm

Before assembling your arm, make sure that you have all the component inside the kit. You can refer to the checklist proposed in the Appendix 1.

Please find here the guide which help you to build your arm.

Make note of the servo IDs and their positions. Each servo has a # written on the back of it. Be sure you assemble the servos in the correct order so that your arm will work!

Figure 9: Order to position the

Through the following step we assembly the Phantom Pincher Robotic Arm:

- ➔ Build the base
- ➔ Build the arm
- ➔ Build the gripper assembly
- ➔ Attach the arm
- ➔ Wire the arm

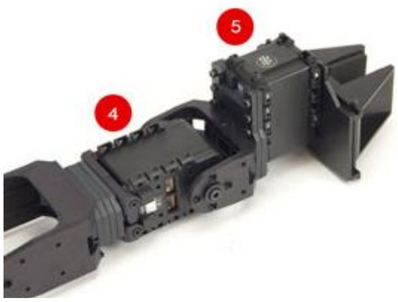
The following images help you see the final results for each parts.

1- Build the base

2- Build the arm

3- Build the gripper assembly

4- Attach the arm



5- Wire the arm

3. Test and Debug

3.1.Testing

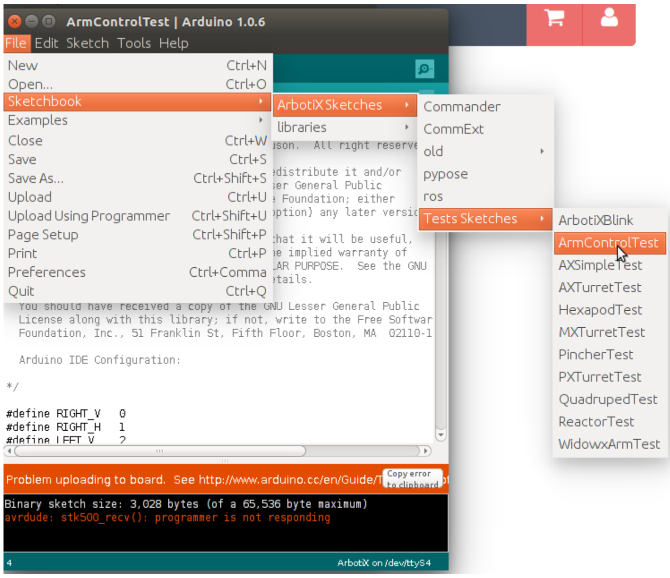
To test the robotic arm, we have to open the Arduino IDE and open:

```
File -> Sketchbook -> ArbotiX Sketches -> Tests Sketches ->PincherTest
```

Once the program is loaded, make sure that your robot is powered from a 12v power supply, and that the power jumper is set to 'VIN'.

Once the robot is programmed and powered, it will move to a 'Center Position' where each servo horn is at its centered position. You will need to wait for ~10 seconds as the robot starts the build check. The robot should move in the exact way as shown in this video. If nothing happen, check the debugging below.

You can also run the ArmControlTest, to make sure your arm work.



3.2.Debugging

No Serial Communications
Check that your ArbotiX is powered correctly Make sure that you are applying a 12v power source and that the power jumper is set to 'Vin'
Re-load the test firmware onto your ArbotiX and make sure you see an 'Upload Successful' message
Check that your FTDI cable is oriented properly
Make sure you are opening the serial monitor on the same port as you programmed the board.
Check that your Servos are set to the correct IDs. Having multiple servos set to the same ID can cause problems with the serial monitor.

"Voltage levels below 10v, please charge battery."
Usually this error is very straightforward - you may get this error if you batteries are low, or you are using a power supply blow 10v.
Check your power and your power jumpers
This error can also be caused if there is a problem with servo #1. If servo #1 cannot be found, the system cannot check the voltage. Problems with servo # 1 are most commonly No Servo ID # 1If there is no servo # 1 present, then the system voltage will not be read
In both cases, you can solve this problem by correctly setting the IDs on your servos.

"ERROR! Servo ID(s) are missing from Scan"
Check to make sure that each servo's cables are firmly seated. Inspect each cable for breaks or tears
Watch each servo as you apply power to the robocontroller. The red led on the servo should light up if it is receiving power
Run through the Servo ID guide and test each servo to make sure that it has the correct ID. Also make sure that each servo works in its full range of motion

IV. ROS Interfacing

In order to be integrated to ROS, we have to download an Arbotix package, a complete package of turtlebot arm, named turtlebot_arm that we can found on GitHub.

In our Workspace:

```
cd ~ros/catkin_ws/src
```

We install the arbotix package using the command:

```
sudo apt-get install ros-indigo-arbotix
```

Then, we can check that the arm is effectively connected to the computer.
Connect the PhantomX Pincher arm through USB port to and make sure that you have read, write and execute access on the USP port using this command:

```
sudo chmod 777 /dev/ttyUSB0
```

Make sure that the arm is connected on port *ttyUSB0* to the computer. Then, we execute the following command:

```
arbotix_terminal
```

You should see the following:

```
ArbotiX Terminal --- Version 0.1
Copyright 2011 Vanadium Labs LLC
>>
```

We check that your servos are all active using the `ls` command

```
ArbotiX Terminal --- Version 0.1
Copyright 2011 Vanadium Labs LLC
>> ls
1    2    3    4    5 .... .... .... ....
.... .... .... .... .... .... .... .... ....
```

This means that all the five servos are active and recognized.
But f after using `ls` command you have:

```
>> ls
.... .... .... .... .... .... .... .... ....

.... .... .... .... .... .... .... .... ....
```

Put again the `ls` command, sometimes it take time to run.

V. Applications

In the arbotix package directory inside the ***turtlebot_arm_bringup*** folder we change the ***yaml*** file to ***new_arm.yaml*** and we write the code below:

```
port: /dev/ttyUSB0
read_rate: 15
write_rate: 25
joints: {
arm_shoulder_pan_joint: {id: 1, neutral: 205, max_angle: 180, min_angle: -60, max_speed: 90},
arm_shoulder_lift_joint: {id: 2, max_angle: 150, min_angle: -150, max_speed: 90},
arm_elbow_flex_joint: {id: 3, max_angle: 150, min_angle: -150, max_speed: 90},
arm_wrist_flex_joint: {id: 4, max_angle: 100, min_angle: -100, max_speed: 90},
gripper_joint: {id: 5, max_speed: 90},
}
controllers: {
arm_controller: {type: follow_controller, joints: [arm_shoulder_pan_joint, arm_shoulder_lift_joint,  arm_elbow_flex_joint,
arm_wrist_flex_joint], action_name: arm_controller/follow_joint_trajectory, onboard: False}
}
```

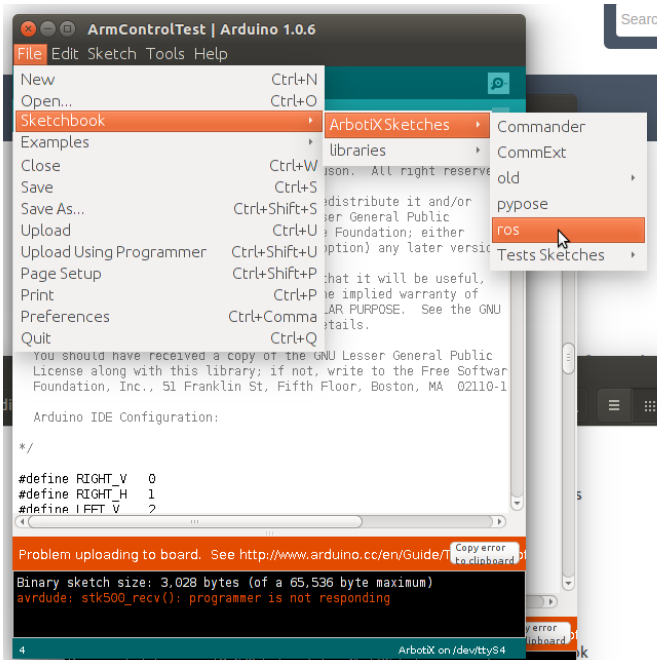
Inside the same folder, ***turtlebot_arm_bringup,*** we create a new launch file named ***new_arm.launch*** and we put the code below:

```
<launch>

<node name="arbotix" pkg="arbotix_python" type="arbotix_driver" output="screen">
<rosparam file="$(find turtlebot_arm_bringup)/config/turtlebot_arm.yaml"
command="load" />
</node>

</launch>
```

Then launch the ***new_arm.launch*** launch file:

```
roslaunch turtlebot_arm_bringup new_arm.launch
```

Make sure to run ROS through Arduino at the same time.

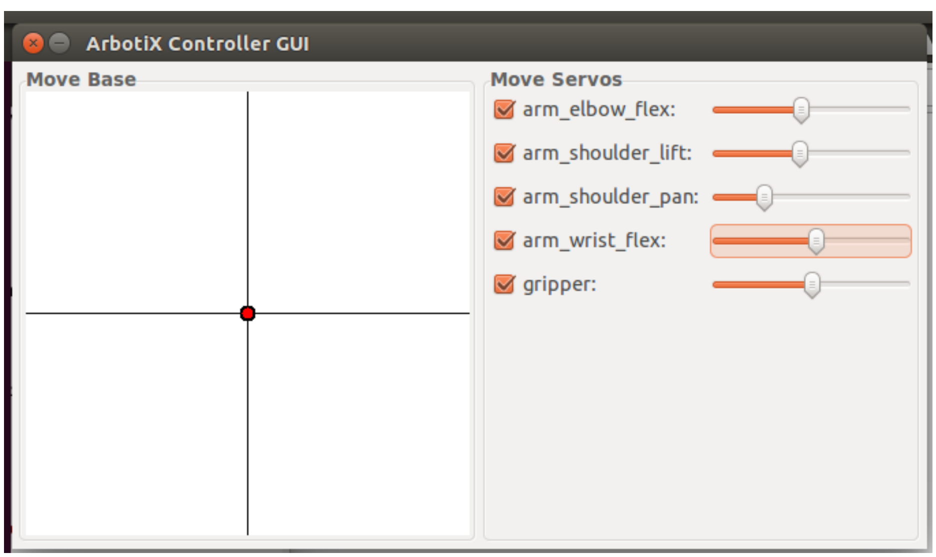


And finally, in another terminal, run the ***gui-controller.***

```
arbotix_gui
```

Now you can play with your arm.

Notice that you have to check all the Servos move.



References

- ✓ https://www.smashingrobotics.com/robotic-hands-with-the-nearest-approach-to-the-human-hand/
- ✓ https://www.smashingrobotics.com/an-overview-of-affordable-robotic-arm-kits-part-1/
- ✓ https://www.smashingrobotics.com/an-overview-of-affordable-robotic-arm-kits-part-2/
- ✓ https://www.smashingrobotics.com/overview-of-commercially-available-professional-robotic-manipulators-part-1/
- ✓ http://www.trossenrobotics.com/p/PhantomX-Pincher-Robot-Arm.aspx
- ✓ https://www.generationrobots.com/fr/402036-bras-robotique-programmable-phantomx-pincher-sans-servomoteurs.html
- ✓ http://www.generationrobots.com/media/PhantomX_Pincher/PhantomX_Pincher_Arm_Quickstart.pdf
- ✓ http://www.trossenrobotics.com/productdocs/assemblyguides/phantomx-basic-robot-arm.html
- ✓ https://www.arduino.cc/en/Main/OldSoftwareReleases
- ✓ http://www.ftdichip.com/Drivers/VCP.htm
- ✓ http://www.ftdichip.com/Support/Documents/InstallGuides.htm
- ✓ http://learn.trossenrobotics.com/arbotix/arbotix-quick-start.html
- ✓ https://code.google.com/archive/p/arbotix/wikis/GettingSetup.wiki
- ✓ https://www.generationrobots.com/fr/402036-bras-robotique-programmable-phantomx-pincher-sans-servomoteurs.html
- ✓ http://makezine.com/projects/build-a-turtlebot-robotic-arm-trossen-robotics-version/
https://www.youtube.com/watch?v=o0JtXuj7HmA
- ✓ https://www.youtube.com/watch?v=ISpqtMfSGBc
- ✓ http://edu.gaitech.hk/turtlebot/turtlebot-arm-pincher.html
- ✓ https://www.researchgate.net/publication/303749615_Survey_of_Robotic_Arm_and_Parameters

Appendix

Appendix 1: Checklist

PHANTOMX PINCHER ROBOT ARM - PARTS CHECKLIST

HARDWARE KIT

- 1x Pincher Hardware & Frames Kit
- 1x Nut & Bolt Hardware Kit

SERVOS & CABLES

- 5x AX Dynamixel Servos (verify ID programmed)

ELECTRONICS PACK

- Arbotix
- FTDI Cable
- 12v 2amp Power Supply
- 5x 200mm 3P Cables

PHANTOMX PINCHER ROBOT ARM (NO SERVO) - PARTS CHECKLIST

HARDWARE KIT

- 1x Pincher Hardware & Frames Kit
- 1x Nut & Bolt Hardware Kit

ELECTRONICS PACK

- Arbotix
- FTDI Cable
- 12v 2amp Power Supply
- 5x 200mm 3P Cables

PHANTOMX PINCHER ROBOT ARM (BAREBONES) - PARTS CHECKLIST

HARDWARE KIT

- 1x Pincher Hardware & Frames Kit
- 1x Nut & Bolt Hardware Kit