MITMECHE

Robotic Arm Control

Department of Mechanical Engineering
Massachusetts Institute of Technology

MITMECHE

AsiaUS Business Connection

Workshop TIM THE BEAVER

Outline

- Hardware
  - Mechanical robotic arm
  - Arduino Uno with Shield
  - Raspberry PI
- Software
  - Architecture
  - Arduino driver
  - ROS launch file
  - Xacro/URDF model
  - Joint State Publisher, Robot State Publisher, Rviz
  - Python arm control script

MITMECHE

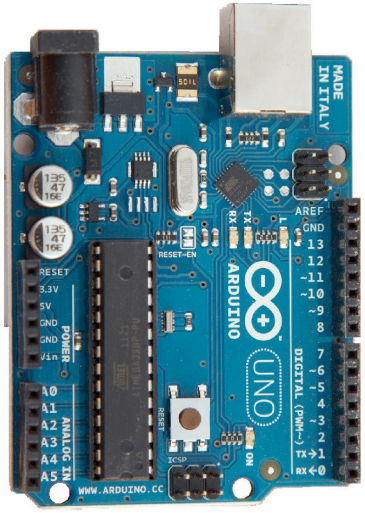
AsiaUS Business Connection

Workshop TiM THE BEAVER

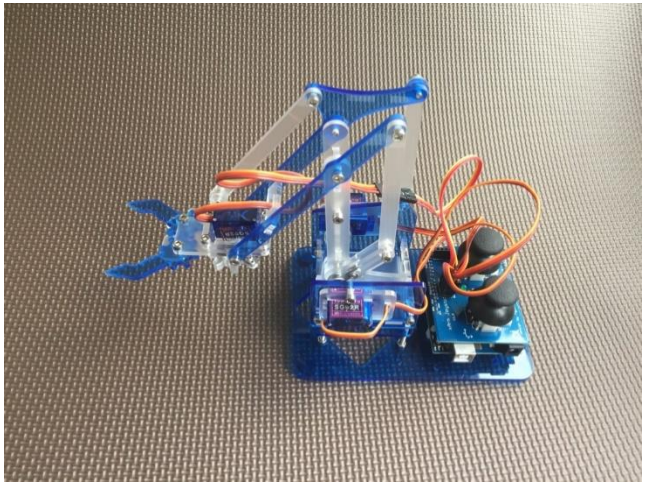
System Overview

Raspberry PI
ROS Visualization

Arduino
Servo Control

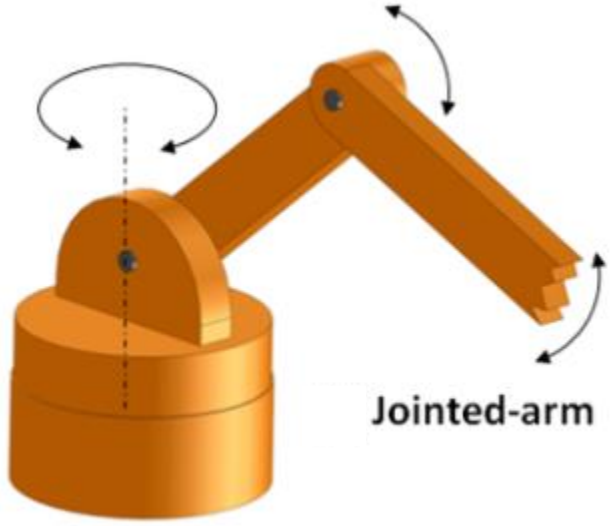


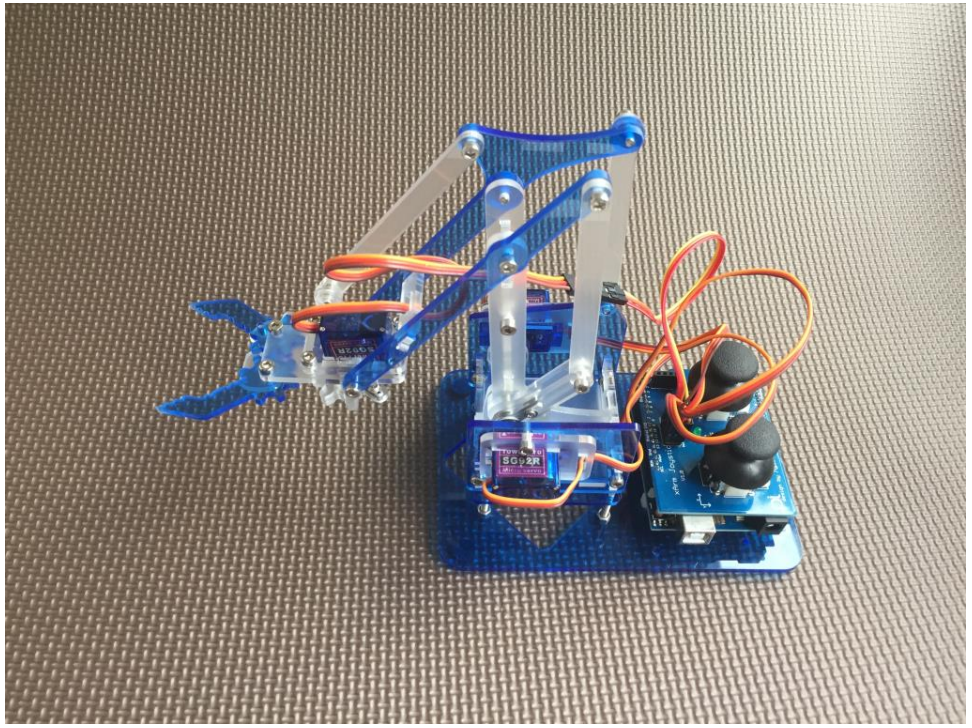
Robotic Arm
Motion Realization



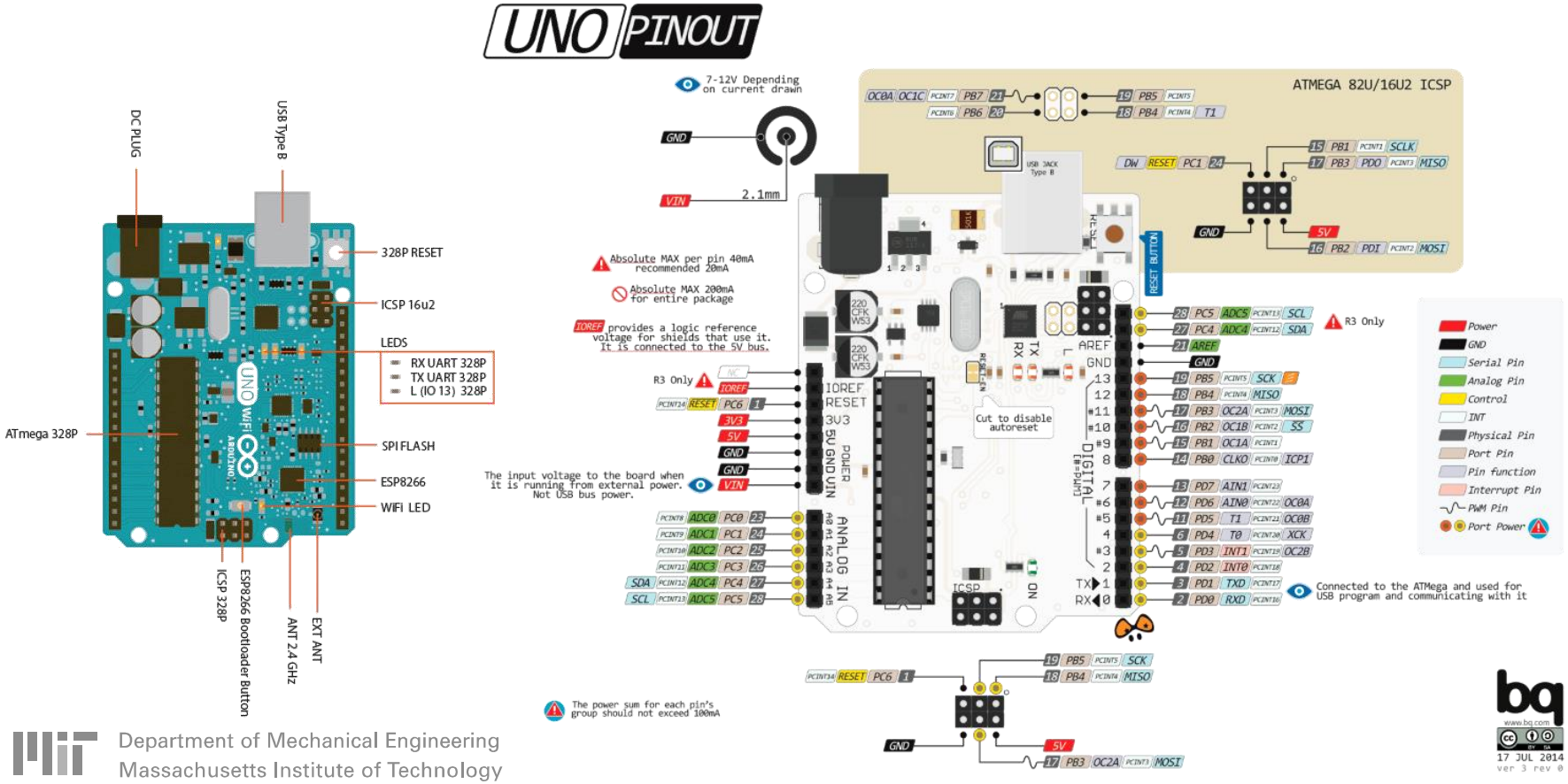
Mechanical Assembly

3 DOF + Gripper

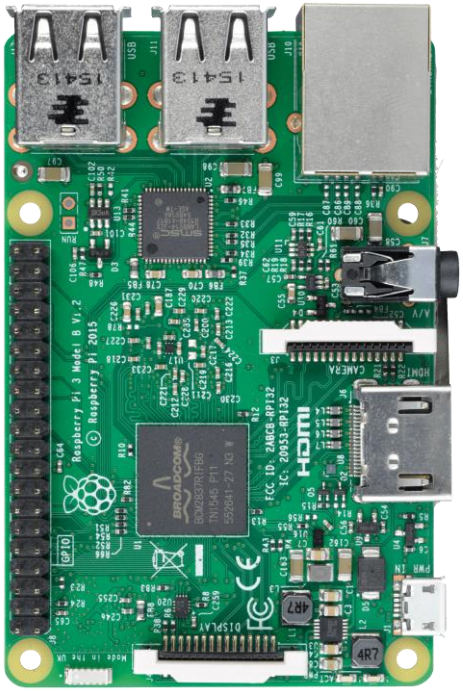


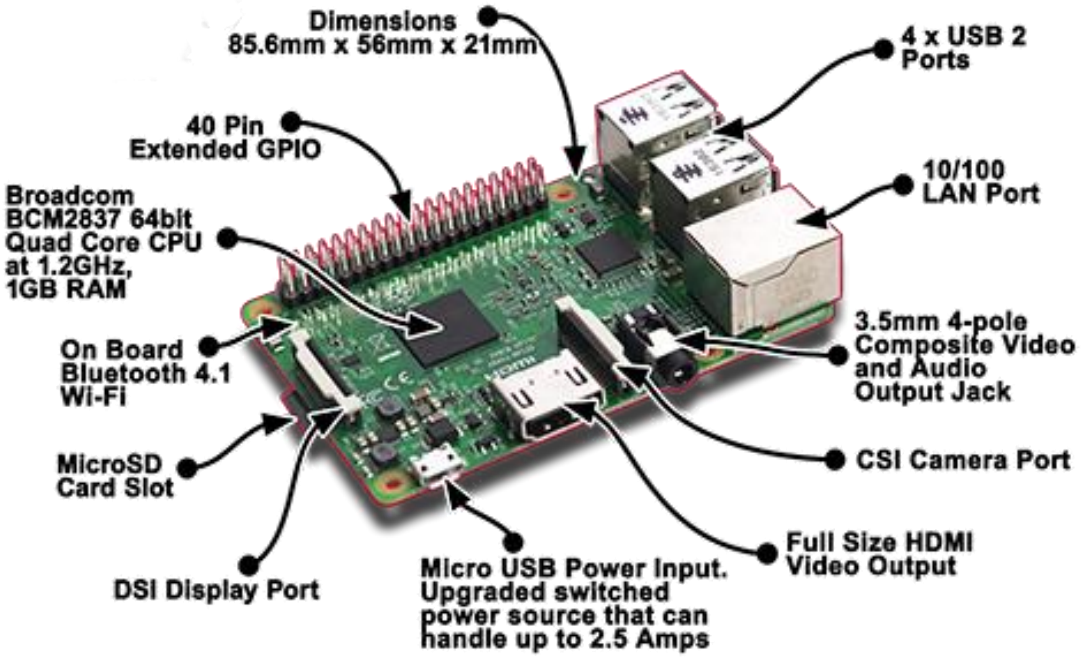


Arduino Uno Board



Raspberry Pi 3





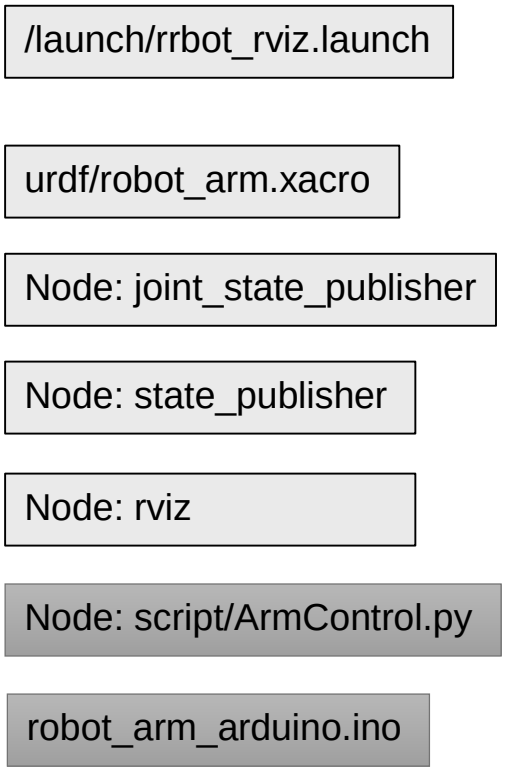
MITMECHE

AsiaUS Business Connection

Workshop TiM THE BEAVER

Software Architecture

- File Function
  - rrbot_rviz.launch: launch file for various nodes
  - Robot_arm_xacro: robot model definition
  - joint_state_publisher: joint angles GUI package
  - state_publisher: robot model management
  - rviz: robot visualization
  - ArmControl.py: custom node
  - robot_arm_arduino.ino: Arduino driver code
- Implementation
  - Edit and upload Arduino code
  - Edit ArmControl.py
  - Change and enable serial port
  - Run rrbot_viz.launch and ArmControl.py



Arduino Driver

- Hardware interface
  - Servos: Pin 11/10/9/5, corresponds to base/left/right/gripper
  - Serial connection (115200 baud rate)
- Function
  - Read serial value from Raspberry PI
  - Parse string into integer values
  - Send command to servo motors
- Implementation
  - String comma separated data conversion to integer
  - Servo motor code

ROS Launch File

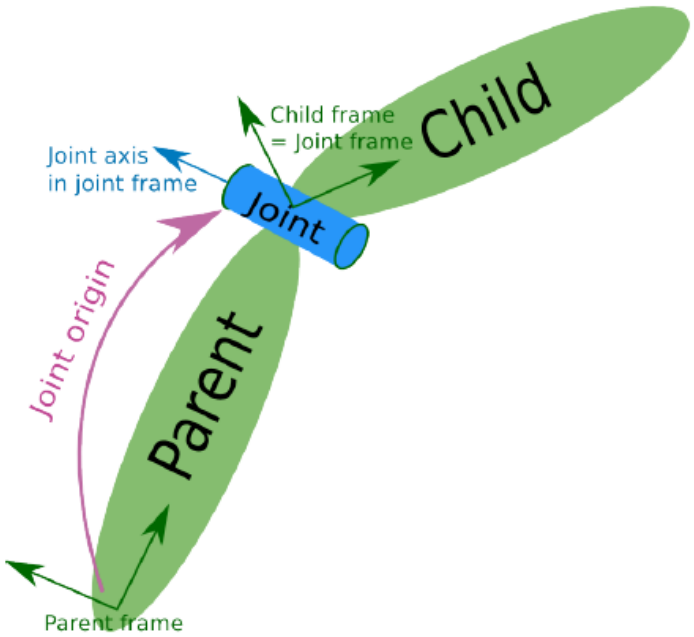
```
<launch>
  <!-- set parameter on Parameter Server -->
  <arg name="model" />
  <param name="robot_description"
    command="$(find xacro)/xacro.py '$(find ros_robotics)/urdf/$(arg model)'" />

  <!-- send joint values from gui -->
  <node name="joint_state_publisher" pkg="joint_state_publisher" type="joint_state_publisher">
    <param name="use_gui" value="TRUE"/>
    <param name="rate" value="0.5"/>
  </node>

  <!-- use joint positions to update tf -->
  <node name="robot_state_publisher" pkg="robot_state_publisher" type="state_publisher"/>

  <!-- visualize robot model in 3D -->
  <node name="rviz" pkg="rviz" type="rviz" args="-d $(find ros_robotics)/urdf.rviz" required="true" />

</launch>
```

MITMECHE

AsiaUS Business Connection

Workshop TIM THE BEAVER

Xacro Model



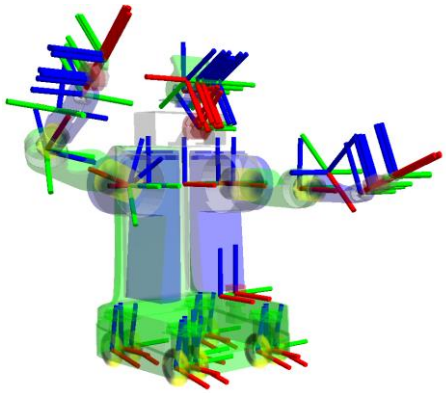

```
<!-- Base Link -->
<link name="base_link">
  <visual>
    <origin xyz="0 0 ${height1/2}" rpy="0 0 0"/>
    <geometry>
<box size="0.8 0.8 ${height1}"/>
    </geometry>
    <material name="red"/>
  </visual>

  <collision>
    <origin xyz="0 0 ${height1/2}" rpy="0 0 0"/>
    <geometry>
<box size="${width} ${width} ${height1}"/>
    </geometry>
  </collision>

  <xacro:default_inertial z_value="${height1/2}" i_value="1.0" mass="1"/>
</link>

<!-- Joint between base Link and rotation Link -->
<joint name="joint_base_rotation" type="revolute">
  <parent link="base_link"/>
  <child link="rotation_link"/>
  <origin xyz="0 0 ${height1 - axle_offset}" rpy="0 0 0"/>
  <axis xyz="0 0 1"/>
  <dynamics damping="${damp}"/>
  <limit effort="100.0" velocity="0.5" lower="-1.57" upper="1.57" />
</joint>
```

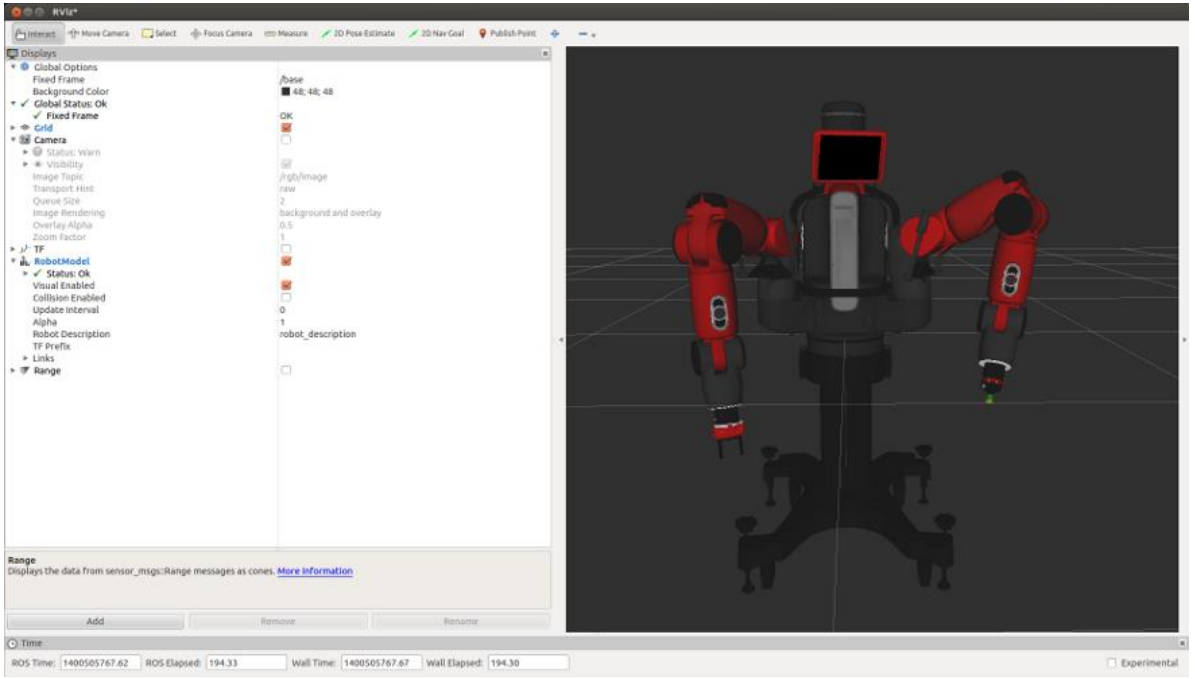
MITMECHE

AsiaUS Business Connection

Workshop TiM THE BEAVER

Joint State, Robot State Publisher and Rviz





MITMECHE

AsiaUS Business Connection

Workshop TiM THE BEAVER

Python Arm Control Script

- Headers
  - #!/usr/bin/env python: declare ROS to run this code with Python
  - rospy: for python ROS operation
  - serial: for serial communication
  - sensor_msgs.msg: obtain JointState message type
- Function
  - main(): define subscriber and keep running
  - callback(msg): compute and send joint angle when message received
- Implementation
  - Angle mapping from Rviz simulation to Arduino servo angle

```
import rospy
import serial
from sensor_msgs.msg import JointState
```

```
serialComm = serial.Serial('/dev/ttyACM2',115200, timeout = 5)
arduino_message = "90,90,90,60"

def main():

    rospy.init_node('ArmControl',anonymous = True)
    rospy.Subscriber('joint_states', JointState, callback)
    rospy.spin()
```

Task Outline

- Edit Code
  - Implement string parsing in Arduino code robot_arm_Arduino.ino
  - Implement angle mapping in Python code ArmControl.py file
- Running Code
  - Copy “ros_robotics” folder into catkin_ws/src
  - Upload Arduino file and connect Arduino to Raspberry PI
  - Run the commands in the readme.txt file

Step 1: compile and run ROS code

```
cd catkin_ws
catkin_make
source devel/setup.sh
echo $ROS_PACKAGE_PATH
roslaunch ros_robotics rrbot_rviz.launch model:=robot_arm.xacro
```

Step 3: run Python code

```
cd catkin_ws
source devel/setup.sh
cd src/ros_robotics/script
python ArmContorl.py
```

Step 2:check serial port

```
ls /dev/ttyACM*
sudo chmod a+rw /dev/ttyACM0
```

⇐ Replace 0 with the number you see and edit ArmControl.py file

Thank You!