Raspberry PI Robot Arm with MATLAB Workshop Guide

Introduction

This guide offers all the necessary instructions for preparing and conducting a workshop using the custom-made and self-designed robot kits developed by Mathworks Student Ambassadors, Samuel Zeitler and Paola Canuto. This workshop was conducted in 2023 at TU Munich for a group of approximately 25 Engineering students, despite a higher number expressing interest in participating. We received positive feedback from many students, particularly appreciating the opportunity to explore MATLAB's hardware interaction capabilities and the valuable robotics knowledge they either gained or reinforced during the workshop.

We hope that this guide will inspire the reader to organize a similar workshop at TU Munich or other universities. Additionally, we encourage readers to utilize the provided hardware and software to cultivate their own ideas. The subsequent section will present an overview of all the resources necessary to get started.

Overview of the Hardware

At the time of writing, 8 robot kits have been assembled. Each kit includes a fully functional robotic arm that can be directly connected to a laptop or PC running MATLAB. The following image showcases a completely assembled robot kit.

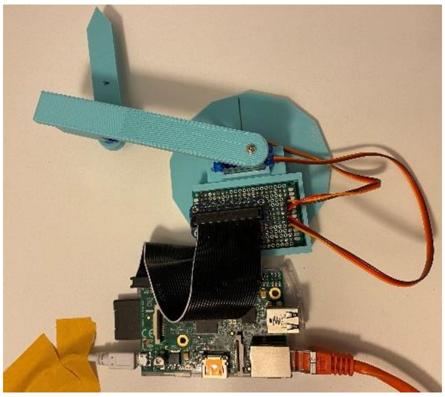


Figure 1: Robot Kit (Assembled)

For packaging and storage, the 8 robot kits come fully disassembled. The following parts compose one robot kit:

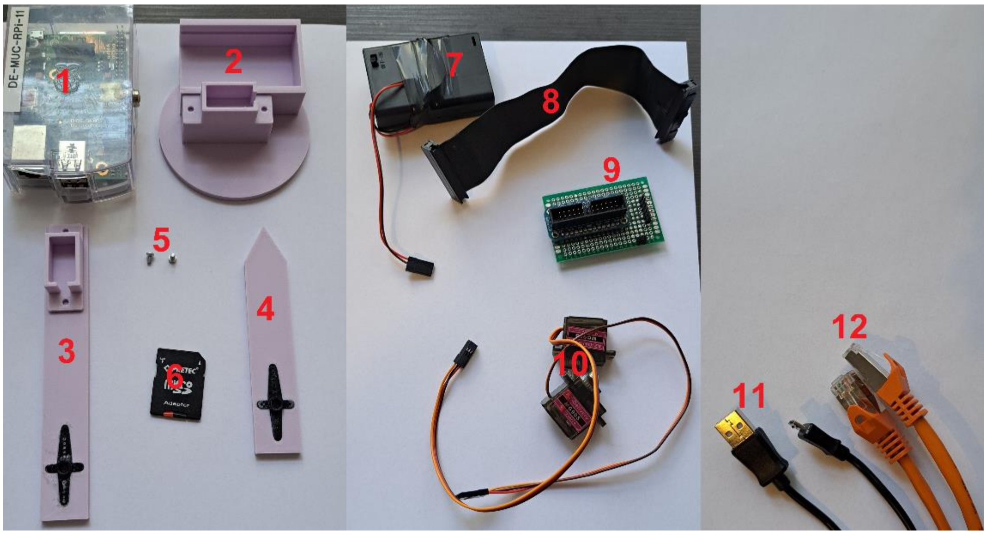


Figure 2: Part list

Part Number	Part Name	Description/Note
1	Raspberry Pi Model B (2012)	Any Raspberry Pi will do
2	3D Printed Base Plate	Fix to table with double sided tape
3	3D Printed First Segment	
4	3D Printed Second Segment	Pointy Endeffector
5	2 Screws	Used for mounting the servos
6	SD Card with Adapter	
7	4.5 V Battery Pack	3xAAA Batteries with OFF Switch: Power for Servos
8	Connector Cable	
9	Selfmade Breakout PCB	
10	2x Servo MG90S	Cheap from Amazon
11	USB to Micro USB Cable	Power for Rasberry Pi
12	Ethernet Cable	Connection PC – Raspberry Pi

Assembly

1. Start by ensuring all required parts are available. Begin assembly by fitting both servos into their respective 3D printed holders. If a servo tends to slip from its holder, wrap some electrical tape around the bottom of the servo to secure a tight fit.
2. Then, align the second arm segment onto the metal horn of the servo in the first segment. The black plastic portion at the bottom of the second segment should snugly fit onto the ridged metal horn. Secure this connection by inserting a screw from the top of the first segment.

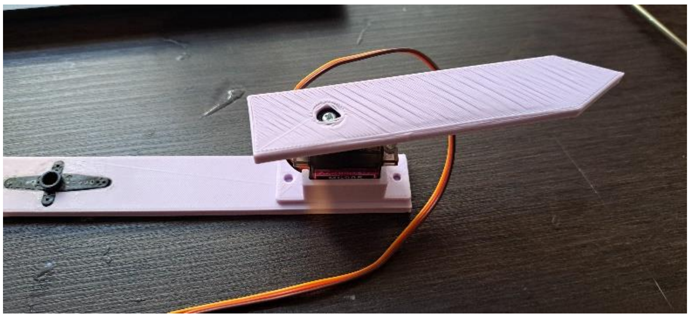


Figure 3: Connected first and second segment

3. Next, attach the connected arm segments to the servo mounted on the base plate, using the same method as before.
4. Insert the connector cable into the breakout PCB, then slide the breakout PCB into its holder on the base plate. Ensure the connectors are facing towards the closed side of the PCB holder.

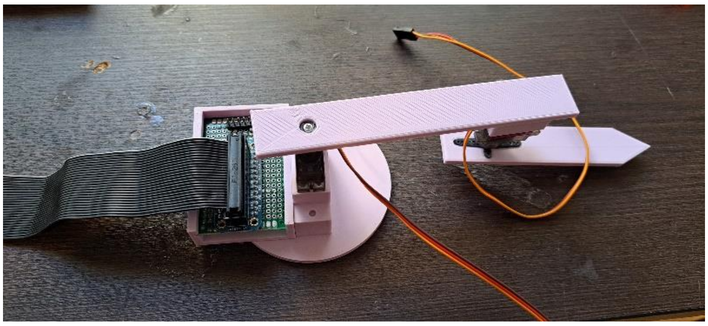


Figure 4: PCB Mounted with Connector Cable

5. To insert the connector cable into the Pi, you'll first need to remove the cover. Refer to the provided pictures to ensure the connector cable is inserted correctly and not in the wrong direction.

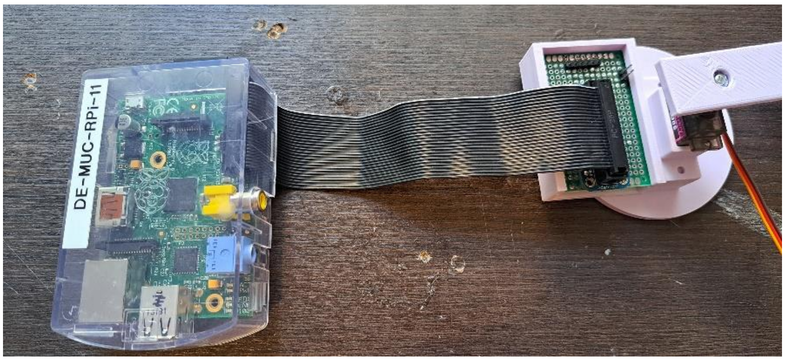


Figure 5: Raspberry Pi connected with PCB

6. Connect all the wires from the robot and the battery pack into the breakout PCB. Ensure the wires are plugged into the correct slots and in the proper orientation. There are symbols on the PCB to help guide your wiring: 'F' signifies the "Front Servo Yellow Cable"; 'R' represents the "Rear Servo Yellow Cable"; '+' denotes the "Plus Pole Red Cable"; and '-' stands for the "Minus Pole Black/Brown Cable".

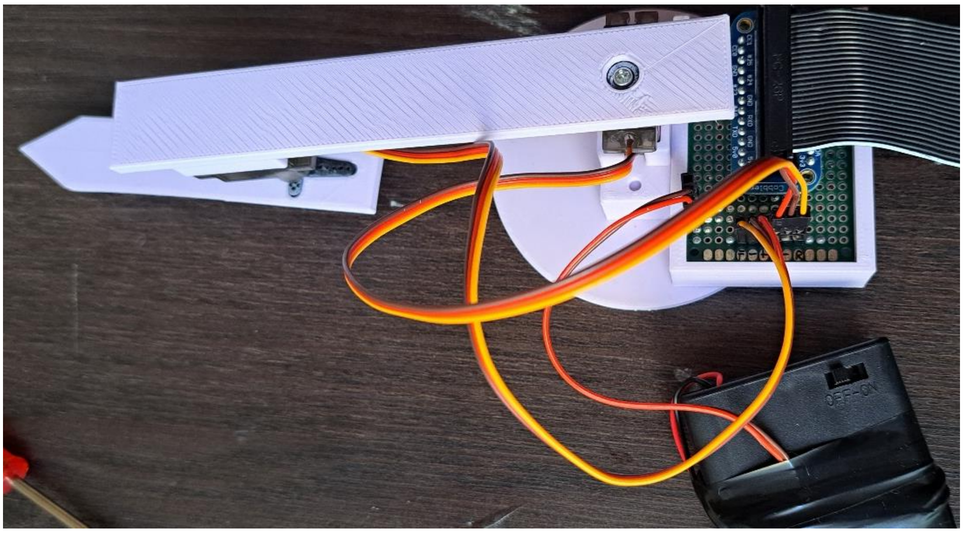


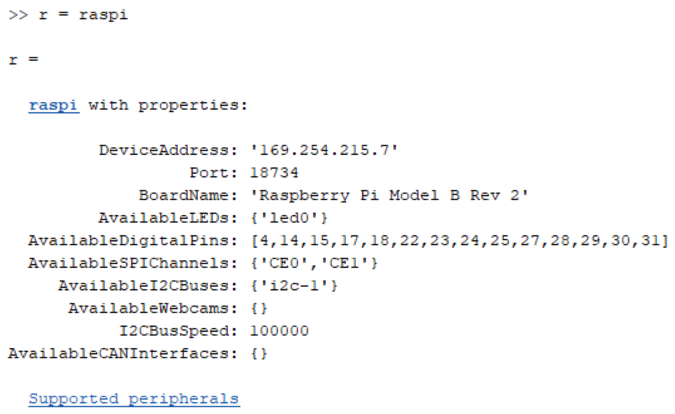
Figure 6: Breakout PCB correct connections

7. Connect the Raspberry Pi to the PC using a USB and Ethernet cable. Always connect the Ethernet cable first; if the USB is connected first, the PC might not recognize the Raspberry Pi. If this happens, disconnect and then reconnect the USB, ensuring the Ethernet is connected first. Ethernet to USB adapters are available if the host machine doesn't have an Ethernet port.
8. A successful connection will cause all the LEDs on the Raspberry Pi to light up. If there's an issue with the SD card or the connection, only the red LED will illuminate. In this case, try to restart the Raspberry Pi and verify the correctness of the Ethernet connection.

PI Software Setup

In order to ensure a successful boot and communication between your Raspberry Pi and MATLAB, a specific MATLAB Raspberry Pi image must be installed on the Pi's SD card. This image has already been installed on the provided SD cards. You can proceed by inserting any of these SD cards into any Raspberry Pi and connecting it to your host machine. Remember to plug in the Ethernet cable before powering up the Raspberry Pi. If the image is properly installed and the connection is accurate, all the LEDs on the Raspberry Pi should light up.

To find out the IP address of your Raspberry Pi, enter 'r = raspi' in the MATLAB command window. This command will establish a connection between your host machine and the Raspberry Pi, displaying its IP address as a result.



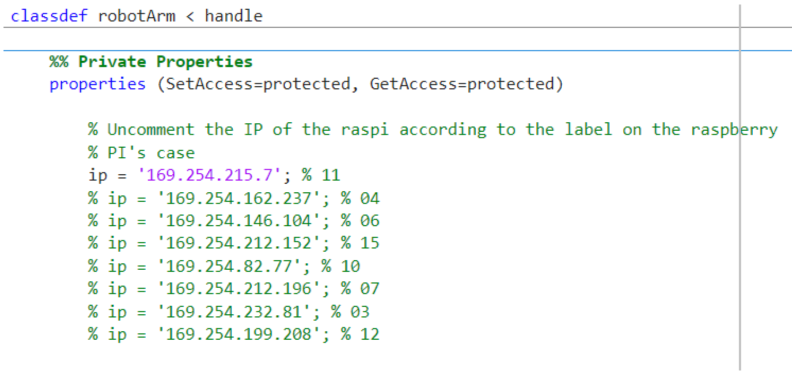
```
>> r = raspi

r =

  raspi with properties:

         DeviceAddress: '169.254.215.7'
                  Port: 18734
             BoardName: 'Raspberry Pi Model B Rev 2'
         AvailableLEDs: {'led0'}
  AvailableDigitalPins: [4,14,15,17,18,22,23,24,25,27,28,29,30,31]
  AvailableSPIChannels: {'CE0','CE1'}
     AvailableI2CBuses: {'i2c-1'}
      AvailableWebcams: {}
           I2CBusSpeed: 100000
AvailableCANInterfaces: {}

  Supported peripherals
```

Figure 7: r = raspi

When you see the aforementioned message, it indicates that you have successfully connected your Raspberry Pi with MATLAB. Remember to take note of the Pi's IP address, as it is now associated with your specific Raspberry Pi and its SD card.

You will need to repeat this process for each Raspberry Pi you are using. Record the IP address of each Raspberry Pi in the 'robotArm.m' file and the 'Workshop.mlx' live script that will be used during the workshop.

For easy identification, label each Raspberry Pi's associated IP address with the corresponding number that's written on the Pi itself. This way, you can skip this process in the future by simply uncommenting the correct IP address when needed.



```
classdef robotArm < handle

    %% Private Properties
    properties (SetAccess=protected, GetAccess=protected)

        % Uncomment the IP of the raspi according to the label on the raspberry
        % PI's case
        ip = '169.254.215.7'; % 11
        % ip = '169.254.162.237'; % 04
        % ip = '169.254.146.104'; % 06
        % ip = '169.254.212.152'; % 15
        % ip = '169.254.82.77'; % 10
        % ip = '169.254.212.196'; % 07
        % ip = '169.254.232.81'; % 03
        % ip = '169.254.199.208'; % 12
```

Figure 8: IP Address Configuration

Note: If you need to rewrite a new MATLAB Raspi image to a new SD Card, be sure to use MATLAB 2021a's Support Package for Raspberry Pi Hardware. Newer versions don't feature the original image:

Matlab 2021a Tutorial for Raspberry PI Setup

Controlling the Arm from MATLAB

After correctly setting up both hardware and software, you can proceed to run the scripts for the workshop. These scripts can be found either on the SharePoint or at this public GitHub repository:

https://github.com/Friedsam2000/MatlabRaspy

The code and workshop materials are user-friendly and well-detailed. As a starting point, we suggest running the 'robotArm.m' file to ensure that everything works as expected.

One crucial point to remember after assembling the Robot Arm from scratch is that the servos' zero position will likely be misaligned. To rectify this, follow these steps:

- Start by running the 'robotArm.m' file. Type 'r = robotArm' into your command line.
- Unfasten the two screws holding the arm segments together and separate them.
- Run the command: 'r.setEndeffektorPosition_Analytic(18,0);'. This action will reposition the servos to their zero state.
- Reassemble the two arm segments, ensuring that the arm is straight. Use the screws to secure the segments.
- Your robot arm should now have the correct zero position.

For additional information or questions, feel free to contact the former Student Ambassador and the author of this guide: **Samuel Zeitler Email: Samuelzeitler2000@gmail.com**