Revel Robotic Manipulator User Guide

January 30, 2018

Svenzva Robotics

Disclaimer

This manual exists for informational use only and its contents are subject to change. This document is open source and as such not all changes come from Svenzva Robotics or a Svenzva Robotics representative. Svenzva Robotics assumes no liability or responsibility for errors or inaccuracies.

Found an error? Open an issue on our github repository or fork our repository, fix the change and submit a pull request.

Contents

Chapter 1

Robot Overview

This chapter goes over the components and technical specifications of the Revel robotic manipulator. The goal is to familiarize the user enough to know the features and limitations of the system as well as introduce supplementary materials and documentation that would help the user delve deeper into the mechanics of the robot.

1.1 General Description

The Revel robotic manipulation platform is a general purpose 6 Degree of Freedom robot arm designed specifically to reduce the barrier of entry into the 'high feature' class of robot arms. These features include:

- high baud communication with [proprioceptive, velocity, acceleration, torque] feedback
- three native control modes on the actuator level (position, velocity and force)
- long reach (63cm)
- high full reach payload (1.2kg)
- a lightweight footprint (under 3kg overall weight)

1.2 Technical Specifications

1.2.1 Physical

Revel contains 7 motors- 1 for each degree of freedom plus one motor to drive the rack-and-pinion finger system contained in the gripper. Each motor is a Dynamixel MX-64 smart actuator. Joints 1,2,3 and 5 contain an external gear reduction system.

Each motor has rotation limitations, as specified in Figure 1.1.

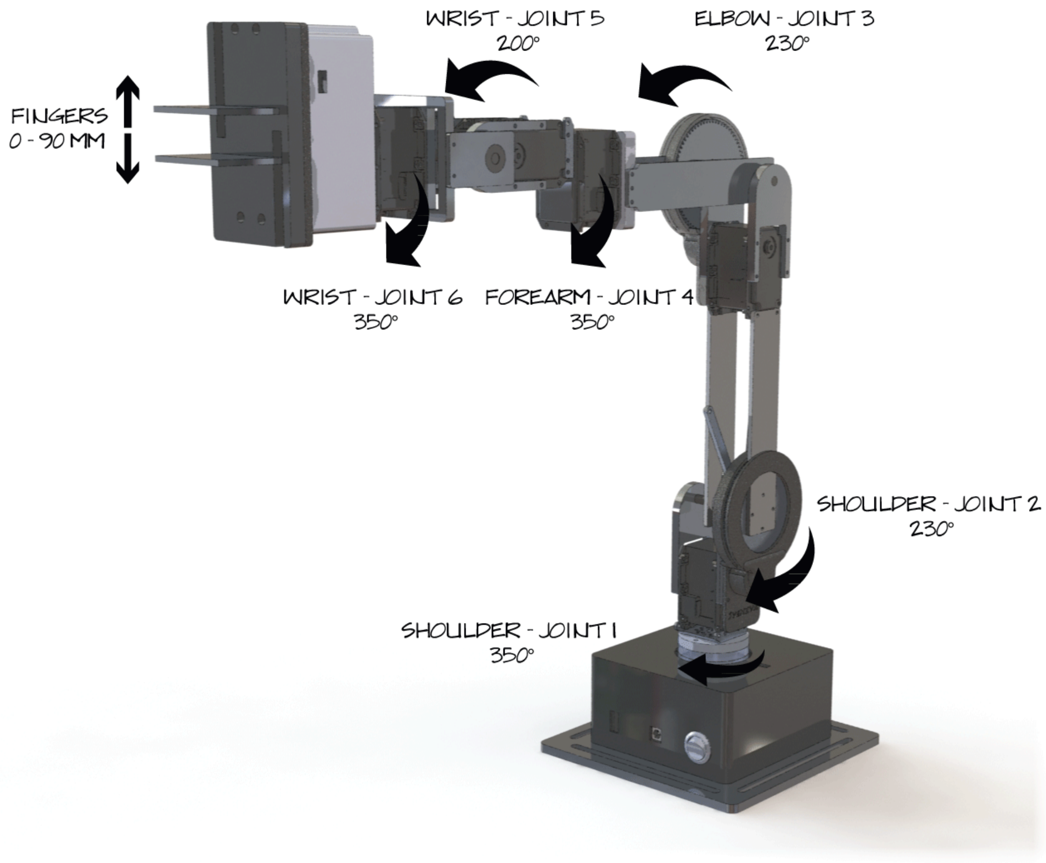

Figure 1.1: Revel robot arm kinematic description

base_link	.0762
link_1	.10542
link_2	.227702
link_3	.1779774
link_4	.07925
link_5	.08
link_6	.060074
link_7	.03785

Figure 1.2: Revel link lengths

Arm Dimensions

Figure 1.2 gives Revel link lengths in meters. `link_7` is given as the nominal height of the standard Revel finger, however, swapping fingers of an alternate style may produce a different value.

For ease of understanding, we denote the link and joint locations in Figure 1.3.

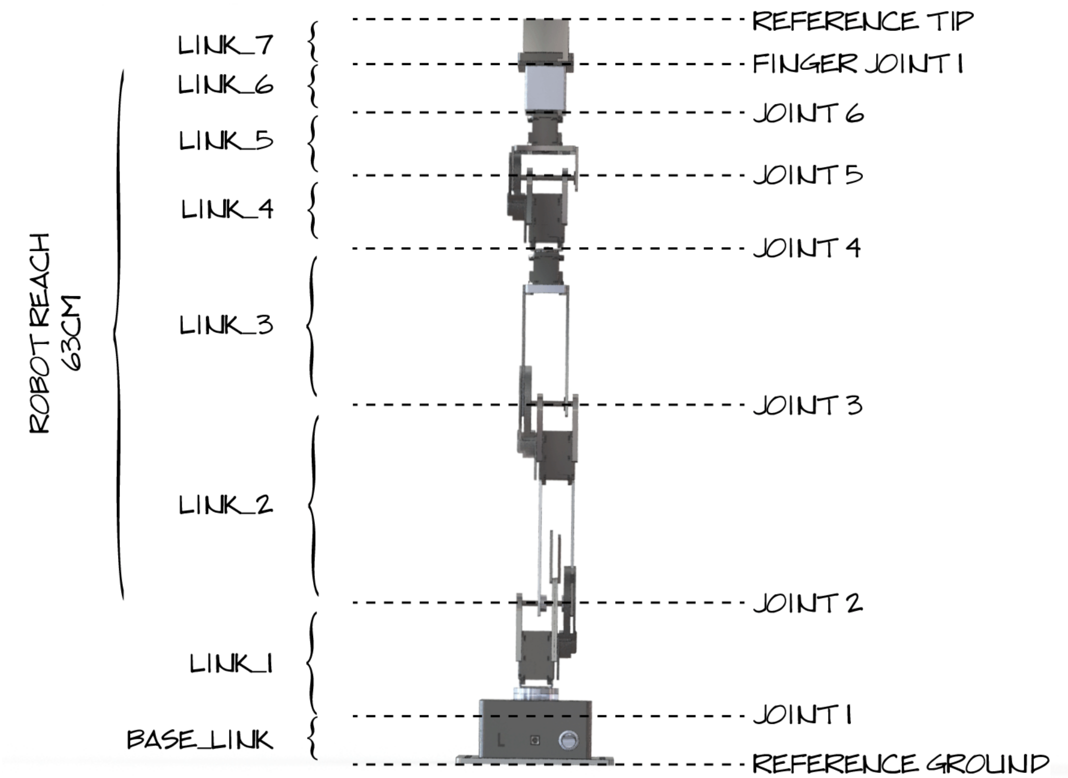

Figure 1.3: Revel robot link and joint locations

Note in Figure 1.3 that the arm reach is measurement from the axis of rotation from `joint_2` to the palm of the gripper. This distance is the maximum reaching distance when `joint_2` is at 90 degrees.

1.2.2 Electrical

Svenzva Robotics officially only supports input voltages of 12V, as is supplied by the power brick included with Revel. It is technically possible to run the arm on 11V or 15V, but this will change the maximum stall torque that each motor can handle. Furthermore, running the motors at 15V can cause the motors to overheat which will damage the motor's processing chips. Svenzva will not cover damaged

Motor type	Maxon
Actuators	7x Dynamixel 64T
Actuator Firmware	Dynamixel MX 2.0 (Version 41 or higher)
Input Voltage	12 V
Power consumption total (min holding)	8.4 watts
Power consumption total (max holding)	66.5 watts
Communication Protocols	TTL (RS-485 also supported)
Communication Frequency	100Hz

Figure 1.4: Revel official electrical data

caused by running the motors at an unsupported voltage.

Communication Protocol and Frequency

Communication frequency is 100Hz with our ROS driver and the motors set to a baud rate of 115200. The motors technically support a baud rate of 4.5M, but we do not guarantee stability at any rate other than the baud rate set from Svenzva HQ.

The actuators we use come in two configurations: TTL or RS-485. By default we use TTL because the motors are slightly lighter. RS-485 motors would be a drop in replacement and would work with our existing USB interface. RS-485 may be a useful option if you plan to add additional sensors or replace the gripper with a custom solution. In both cases, you may be able to use the existing power and data lines.

1.3 System Limitations

1.3.1 Rotational Limits

As a cost saving and dynamics optimizing measure, no joint utilizes a slip ring. Thus, each motor is limited in the amount of rotation that is permitted. See Figure 1.1.

1.3.2 Starting Position

As mentioned previously, several motors utilize an external gear box to increase torque output. This addition greatly increases the positional resolution of the motor + gearbox system. This also has the consequence of requiring that the arm be powered on in approximately the ‘all zero’ position for the motors that have gear boxes.

The arm must be put in the all zero position prior to the robot being powered ‘on’ for the arm’s proprioception feedback to be accurate.

all-zero position

The ‘all zero’ position can be described as follows when looking at the robot’s power and usb connections: the arm is standing straight up, with the `joint_2` gear to the right and orthogonal to the plane that the power and usb connections are mounted; the gripper is turned so that the flat portion of the gripper housing is facing you, parallel with the power and usb connections.

See Figure 1.3 for an example of when the arm is in the zero position.

Details

Each motor has a native position encoder resolution of 4096 steps in 360 degrees, making the resolution 0.088 degrees per step. The encoding system can then count additional rotations outside of this range digitally and add that to the current value of the encoder.

However, for this system to work as expected, a given joint must start in the range of 360 / `GEAR_RATIO` degrees from the true zero. Outside of this range, the motor’s encoder will have overflowed out if its initial 0,4096 step range. Without being powered, the motor is unable to count this overflow digitally.

joint_1	4
joint_2	6
joint_3	6
joint_4	1
joint_5	4
joint_6	1
joint_7	1

Figure 1.5: Revel gear ratios

The arm’s gear ratios for each joint is referenced in Figure 1.5.

1.4 System Safety

Lower level safety measures are in place to protect the motors from damaging themselves. In particular, each motor has a temperature threshold set to approximately 80 degrees C. If a motor's operating temperature exceeds the threshold, the motor will disable itself to prevent damage.

Additionally, but separately, each motor has a proprietary internal model of force exertion which activates when a motor exerts more than 20 percent of its maximal torque rating. This model then permits output torque over a time period proportional to its force exertion above 20 percent. After this time period, the motor will disable itself to prevent damage.

At 20 percent, a motor will continuously exert for about 5 minutes. At 100 percent (the motor's stall torque) a motor will continuously exert for approximately half a second.

Svenzva designed the system dynamics around this fact so that at full payload at full reach, any given motor would not exceed 20 percent output torque. This feature ensures system safety if a user gives an unsafe command that would cause the arm to move through an object, for example, a table. Since the table will not allow the arm to pass through it, the arm will attempt to reach its setpoint past the table with increasing force due to the motor's internal control loop. The motors will then shut off once this high-force stall condition has been detected.

You can learn more about our motor's internal limiting features at here, which include current, voltage, and acceleration limits.
You can learn more about the control loops the motor's employ here.

Chapter 2

Getting Started with Revel

This chapter will teach a new user how to setup the Revel robot arm as-received from Svenzva Robotics, prepare it for use and go over safety precautions. It will also outline the process for how pack the arm for travel.

2.1 Unpacking the Robot

Included in your shipping box should be the following items:

- A locking, hard-shelled Revel travel container
- Revel base
- Revel arm
- 12V Power supply
- USB Female-Male connection cable
- M2.5, M3.0 allen keys
- Associated hardware (M2.5 bolts, washers, base sliding bolts)

The Revel Manipulation Platform will come contained in a hard-shelled safety container containing two parts: the base and the arm. These must be connected to have a functioning robot, which we cover in the following section.

2.2 Assembling the Robot

2.2.1 Connecting base and arm

For ease in transport, the robot sits in its travel case in two parts: the base and the arm. These must be connected before the arm can be used.

Figure 2.1: Included contents of a shipped Revel platform: carrying case containing the robot & cables, and an assorted hardware pack

Figure 2.2: A side view of a bolt with the two washers applied correctly.

Physical

Parts needed:

- 4 x M2.5 bolts
- 4 x M2.5 lock washer
- 4 x M2.5 flat washer
- Revel baseplate
- Revel arm
- M2.5 allen key

The actual assembly of the two pieces is straightforward. There are only two subtleties: organizing the bolt-lock-flatwasher assembly, and orienting the arm on the base correctly.

First, prepare the four bolts. With the bolts sticking up, place a lock washer (bent and offset looking washer) on the bolt shaft first. Then place a flat washer on the bolt shaft second. See Figure 2.2 for an example of a correctly assembled bolt.

Next, place the baseplate in a secure area where it will not slip or tip once the arm is in place. You may even want to secure it using the sliding attachment points or hand clamps as discussed in Section 2.2.3. Place the arm in the bracket facing up, orienting the arm so that the gear is facing the blind for the motor cable. (See

Figure 2.3: Correct placement of the arm into the baseplate motor bracket.

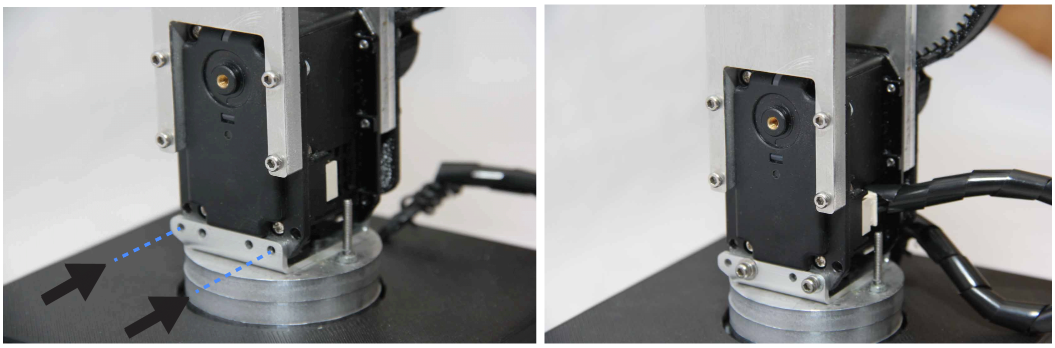

Figure 2.4: Install the prepared bolts in the holes indicated, aligning the holes in the lower arm motor.

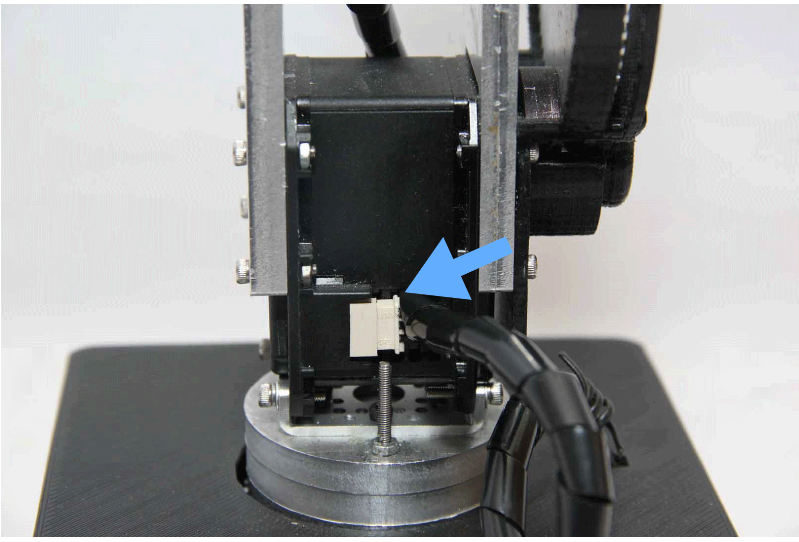

Figure 2.5: Install the cable from the baseplate into the lower arm motor.

Figure 2.3)
Now, align the four holes in the bottom motor of the arm with the holes indicated in 2.4 and screw in the assembled bolts.

Electrical

The final part of assembling the arm is connecting the power/data lines from the base to the arm. Take the cable from the baseplate blind and insert it into the empty communications port on the motor as in Figure 2.5.

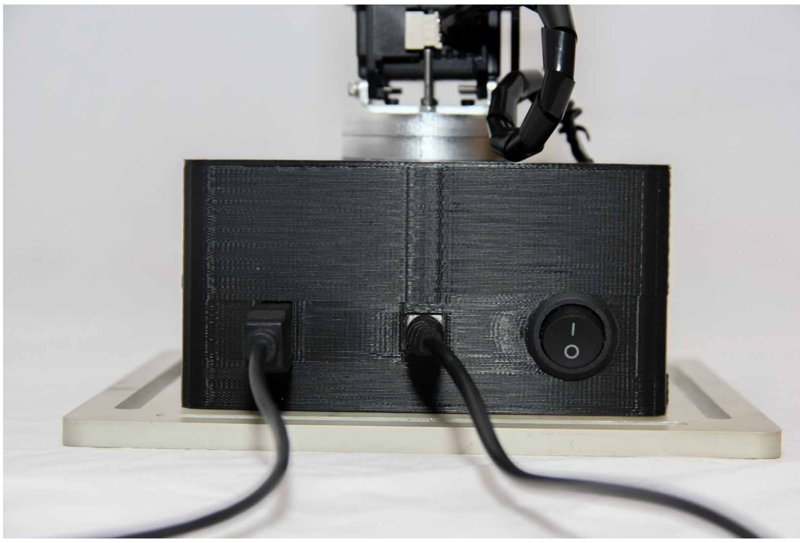

Figure 2.6: Successful attachment of the USB and power connectors.

2.2.2 Connecting cables to power distribution board

You can now attach the USB and power cables. Do so by taking the included Female-Male USB cable and placing the female end into the USB port in the power distribution area on the baseplate. Similarly, take the included power supply barrel connector end and place it into the barrel connecting port on the power distribution area. After this step, the base should look like Figure 2.6.

2.2.3 Mounting baseplate

The baseplate has mounting slots that allow flexibility in secure mounting to existing structures like threaded pegboard or metal plates. The 2 included square-necked bolts, washers and wingnuts allow one to secure the base within these slots. See Figure 2.7 as an example.

Alternatively, some customers opt for using handclamps for temporary installments. This may be an effective choice if you do not have a dedicated structure. But know that the hand-clamp method of attachment does not provide as rigid a mount and would allow the arm to fall should they fail.

2.2.4 Disassembly for Travel

Disassembly of the arm for placement back in the travel case happens in the reverse order of the above guide. That is, specifically:

- Remove power and USB cables from Robot power distribution board

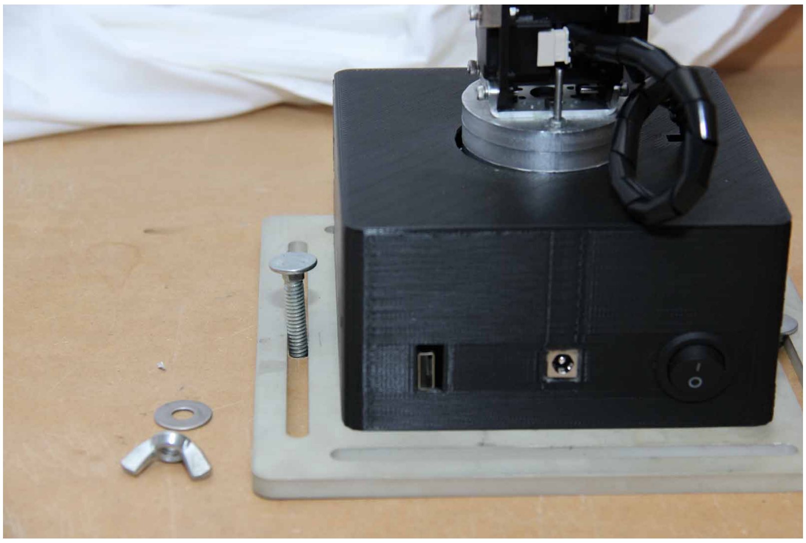

Figure 2.7: A user pushes a mounting bolt through a dedicated mounting board.

- Remove the four M2.5 bolts holding the arm to the base
- Lift the arm off the base and place into travel case
- Remove base mounts
- Place base into travel case

2.3 Turning on the robot

Now that the robot is assembled, the barrel connector end of the power supply should be plugged into the barrel jack on the base of the robot. The female end of the included USB cable should also be plugged into the USB device on the base of the robot.

As mentioned in Section 1.3.2, Revel must be placed in approximately the ‘all zero’ position before being turned on. After you have put the arm in the correct position, flip the power switch to the on (dash) position. When the robot is turned on, all LEDs on the robot’s motors will blink red once. This is positive confirmation that the arm has turned on successfully.

2.4 Using the robot

All code used to power Svenzva Robotics robots is available for download on our github repositories, found at https://github.com/SvenzvaRobotics.

Instructions and details on how to use our codebase and its organizational structure are available on the github wiki's, which evolve alongside code changes.

The `svenzva_ros` wiki can be found at https://github.com/SvenzvaRobotics/svenzva_ros/wiki.

Chapter 3

Data Sheets

Here we provide links to the item's technical data sheets and third party repositories if applicable.

3.1 Data Sheets

http://support.robotis.com/en/product/actuator/dynamixel/mx_series/mx-64(2.0).htm - Full details on the actuators used in the Revel Robotic Manipulator.

3.2 Guides and Wikis

https://github.com/SvenzvaRobotics/svenzva_ros/wiki - `svenzva_ros` wiki, contains quickstart information as well as deeper level code organization.

http://wiki.ros.org/svenzva_ros - The ROS wiki page for `svenzva_ros` package. Will contain tutorials.

3.3 Repositories

https://github.com/SvenzvaRobotics/mx_dynamixel - Lower level MX Dynamixel drivers for firmware at or above 2.0. A ROS-ified python interface to the motors.

https://github.com/SvenzvaRobotics/svenzva_ros - The main code repository for ROS drivers.