Exoskeleton Documentation Contents:

1 Power On/Off Exoskeleton: ... 2

1.1 Automatic relay and motor temperature monitoring: ... 2

2 Electronics setup: ... 3

2.1 Pinout connections: ... 3

3 Four-bar-linkage: ... 3

3.1 Torque sensor calibration data: ... 3

3.2 Output torque calculation: ... 4

3.3 Instantaneous Centre of Rotation (ICR) calculation: ... 4

4 Software required: ... 4

4.1 Installing libraries for python scripts and Arduino codes: ... 4

5 Steps to read sensor data: ... 5

5.1 Arduino Mega and Nano Serial Monitor: ... 5

5.2 PuTTy (logging software): ... 6

5.3 ODrive: ... 6

6 Steps to run the exoskeleton with Joystick only (One side only): ... 6

6.1 Step 1: Perform motor calibration: ... 6

6.2 Step 2: Update the motor limit values: ... 6

7 Troubleshooting: ... 7

1 Power On/Off Exoskeleton:

The complete exoskeleton (Backplate components, left and right actuators) operate between 22.2 and 24 V. Peak current values are between 5 and 7 A.

Make sure that the power switch is OFF and the emergency button is PUSHED DOWN.

To supply power to the exoskeleton:

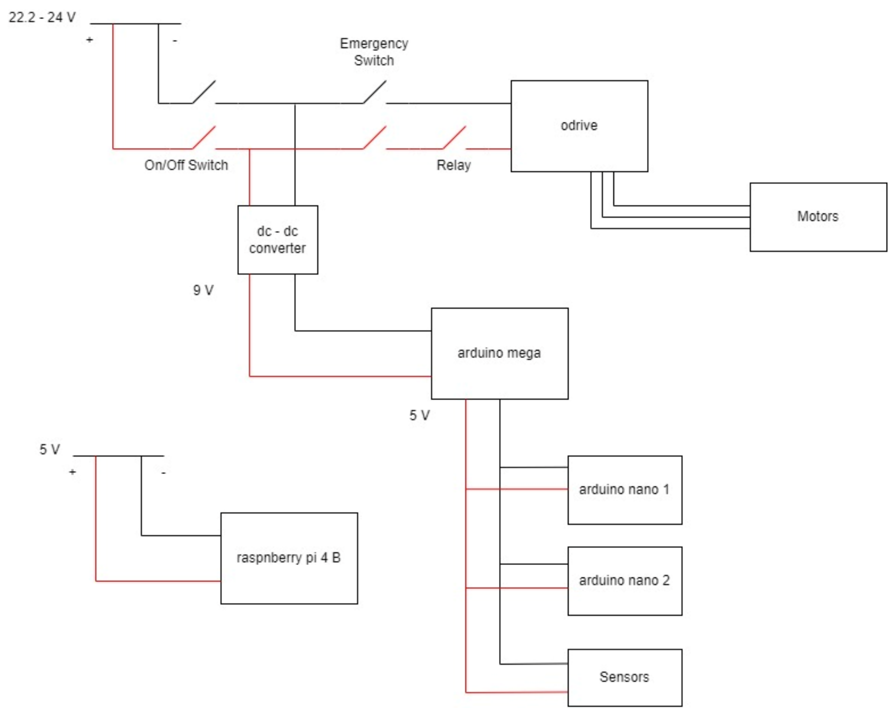
1. Connect the supply terminals to the red (+) and black (-) leads. Connect the leads to the XT-90 connector. Alternatively, the LiPo battery can be connected directly to the XT-90 plug.
2. Turn on the power switch → this powers the Arduino and the actuator sensors only. It can be used to test the sensors, irrespective of the odrive and motor.
3. Afterwards, disengage/twist the emergency switch → this powers the odrive and the motors.

1.1 Automatic relay and motor temperature monitoring:

A third automatic switch is the relay which automatically activates if correct temperature conditions of the motor are met, provided the Arduino Mega is powered on. If the motor is overheating, the relay will stay OFF so that the odrive cannot be powered on. This can be overridden in the Arduino Mega code. The normal ambient range is between 20 to 23 °C. With a reference of 22 °C, heating of the motor raises the ambient temperature to 25 °C. **A temperature above 25 °C indicates that the motor is starting to overheat.**

To power off the complete exoskeleton, the power switch can be turned off.

In case of emergency, the emergency switch will cut power to the odrive and the motors only.

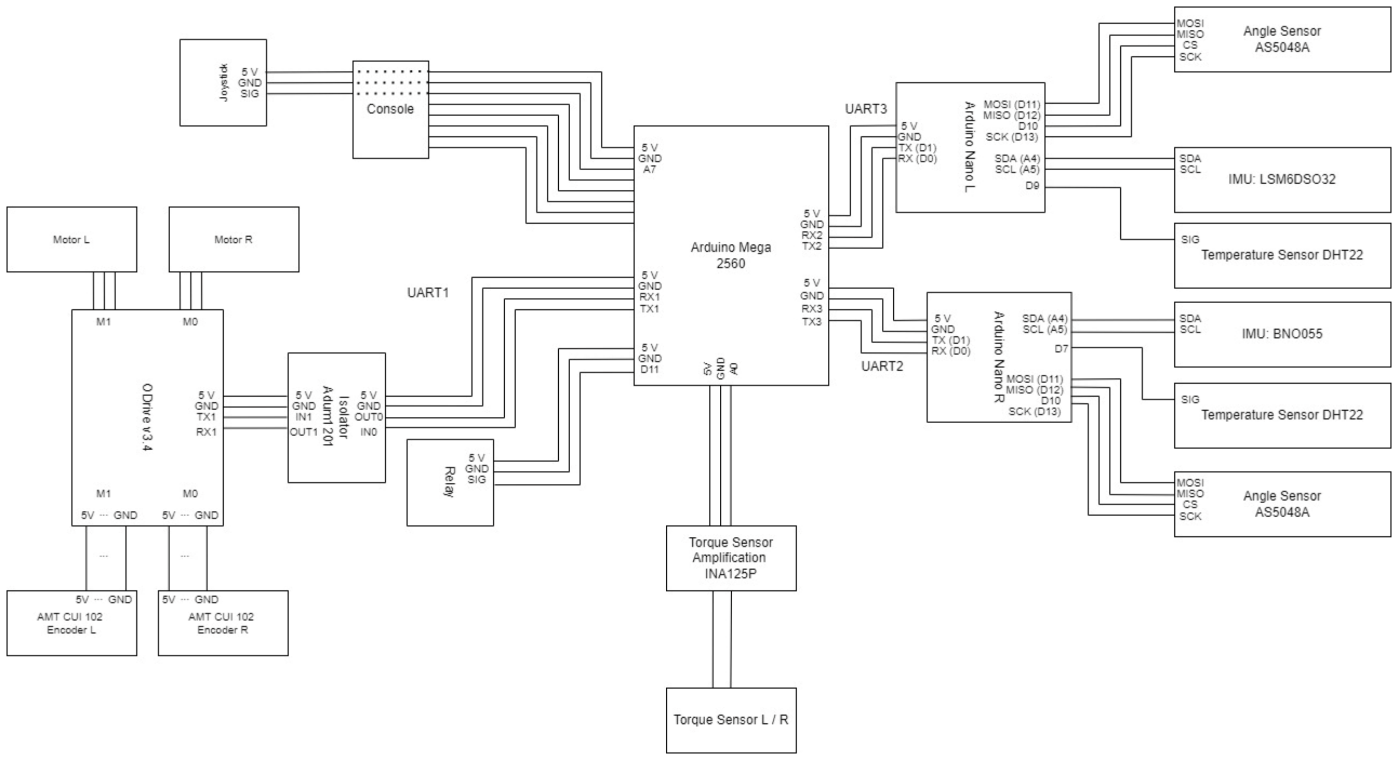


Note: The raspberry pi operates at peak 5V. The Arduino operates at 9V. Thus, the same dc-dc converter must not be used for both. The raspberry pi must be powered by an external adapter rated

5V 2A. The maximum allowable voltage for the raspberry pi is 5.25 V before the microprocessor is potentially damaged.

2 Electronics setup:

2.1 Pinout connections:



The schematic above shows the pinout connections of all the devices within the exoskeleton.

3 Four-bar-linkage:

3.1 Torque sensor calibration data:

While the torque sensor provides a linear relationship between the torque applied and the voltage change, the four-bar-linkage results in a non-linear relation between the torque and the voltage. Moreover, the gain resistance connected to the INA125P (Instrumentation Amplifier for the torque sensor) dictates the slope of the linear relationship.

The peak torque that can be measured within a safety factor of 1.25 is 20 Nm. Effectively, a nominal torque of 10 Nm can be measured with a minimum safety factor of 2.

The linear equation relating the analog value, x_o and the torque applied, T_{crank} is:

$$x_o = 0.867 * T_{crank} + 80$$

$$T_{crank} > 0 \text{ for tension / leg extension}$$

$$T_{crank} < 0 \text{ for compression / leg flexion}$$

Note: Compression torque is negative, while tensile torque is positive. Changing the gain resistance will change this relation. The gain is calculated from

$$G = 4 + (60k\Omega / R_G)$$

3.2 Output torque calculation:

In the directory: Exoskeleton Documentation Set\Python Scripts\Calculations, the Python script "Four Bar Linkage force calculation.py" gives the force at the shank link with respect to the crank force and input angle in the four-bar linkage. The input variables are the crank angle, and the input torque. The output torque at the coupler/shank link is then calculated.

3.3 Instantaneous Centre of Rotation (ICR) calculation:

In the directory: Exoskeleton Documentation Set\Python Scripts\Calculations, the Python script "Angle and Transformation Points of Four-Bar-Mechanism.py" gives the output angle of the shank link, the transformation of the ICR and transformation vectors of the different links in the four-bar linkage.

The input is the crank angle which is denoted as 't2' and the outputs are:

- shank output angle
- p1, p2, p3, p4, p5 and p6
- icr

4 Software required:

The following software/IDEs are required on a computer for the operation of the exoskeleton:

1. Python 3.9 IDLE
2. Arduino 1.8.18 IDE
3. PuTTy (data logging software)

Follow the installation procedures:

For **Python 3.9 IDLE installation**, run the following executable file "python-3.9.9-amd64.exe" in the "Installation Files" folder.

For **Arduino IDE Installation**, run the following executable file "arduino-1.8.18-windows.exe" in the "Installation Files" folder.

For **PuTTy installation**, run the following executable file "putty-64bit-0.77-installer.msi" in the "Installation Files" folder.

4.1 Installing libraries for python scripts and Arduino codes:

Python scripts require the following libraries:

1. Odrive library

To install the libraries, open Windows Command Prompt (with Win Key, then type "cmd", press Enter)

Type in the following commands:

```
pip install odrive
```

To test if the library is properly installed, open Windows Command Prompt and run the command:

```
odrivetool
```

Arduino Codes require the following libraries:

Simply copy the contents of the following folder:

"Installation Files\Arduino libraries" to "My Documents\Arduino\libraries"

It could also have this path:

“Dieser PC\(C:)Lokaler Datenträger\Benutzer\<Benutzer Name>\Documents\Arduino\libraries“

5 Steps to read sensor data:

5.1 Arduino Mega and Nano Serial Monitor:

For this step, the exoskeleton does not need to be powered on. If it is powered on, these steps will not affect its operation.

The Arduino mega is already loaded with a code which displays the different sensor readings on the serial monitor of the Arduino IDE. This can be accessed from the data of a serial port on a computer by connecting the Arduino Mega via USB to the computer.

Similarly, by connecting a USB cable to an individual Arduino Nano, the data that the sensors read are displayed.

Serial Monitor:

1. Make sure the Arduino IDE is opened and the Arduino is connected to the computer
2. Click on Tools and select the appropriate ‘Board’ and ‘Port’
3. Click on the magnifying glass icon on the top right.
4. This will open the Serial Monitor in a new window.
5. Select the Baud: 9600 on the bottom right in the drop down menu.
6. The data from the sensors will be displayed.

The tables below display the order, type, name and description of the sensor data based on the microcontroller:

	Arduino Mega	**Arduino Nano R**	**Arduino Nano L**
Port	22	25	23
Data order	data0, s22, s32, s42	temperature, angle_value, axx, gzz	temperature, values_of_angle, values_of_imu

Data Name	**Data Type**	**Data Description**
data0	integer	Analogue value from torque sensor
s22	unsigned integer	Angle sensor value
s32	integer	IMU sensor value: acceleration in x-axis = axx
s42	integer	IMU sensor value: gyroscope in z-axis = gzz
temperature	integer	Ambient temperature value of right motor
angle_value	integer	Raw angle value of the right absolute encoder
axx	integer	x-acceleration of IMU on right side
gzz	integer	z-gyroscopic value of IMU on right side
temperature	integer	Ambient temperature value of left motor
values_of_angle	char	Raw angle value of the left absolute encoder
values_of_imu	char	z-gyroscopic value of IMU on left side

The port number might differ from computer to computer. The right leg is equipped with the BNO055 which can also output the acceleration value. Arduino stores 16-bit integer.

5.2 PuTTy (logging software):

To use PuTTy to record data from the serial port, once the window opens:

1. Select ‘Session’ on the top left, and then the ‘Serial’ radio button.
2. Enter the correct port in ‘Serial line’.
3. Select ‘Logging’ under ‘Session’, then select ‘All session output’
4. Give a file name in ‘Log file name’ and enter the save location.
5. Finally, click on ‘Open’ at the bottom.

The serial port is opened, and the data displayed. At the same time, the respective data is logged at the desired location in a .log file.

5.3 ODrive:

For odrive, please refer to the odrive 0.5.4 documentation. The section ‘Parameters and Commands’ provides a list of commands that can be used in odrivetool or a python script to obtain data from the odrive directly. Coding in C++ allows the data to be read by the Arduino as well. This is already included in the Arduino codes; however, the sensor data from odrive is not displayed on serial monitor.

6 Steps to run the exoskeleton with Joystick only (One side only):

Ensure ODrive is properly connected to a power source and powered on. Odrive motor controller does not receive power via the USB cable.

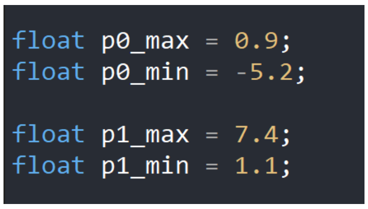
6.1 Step 1: Perform motor calibration:

1. Via PC/RPi: Connect Odrive USB cable to the PC/RPi
2. Run calibration python script “odrive_calibration_python_script.py”
3. The code runs the motor calibration and initializes all parameters (takes about 15 seconds)
4. Errors are displayed. If there are no errors, a 0 is returned.
5. In case of no errors, the position of the motors are displayed in terms of ‘turns’.
6. **DO NOT close or end** the running python script. It is required for the next step.

6.2 Step 2: Update the motor limit values:

The encoder position values must be updated in the “Arduino_mega_code.ino” before it can be operated. This is because the incremental encoder in the actuator will not store the previous position. This is noticeable if the actuator is moved several times while turned off.

1. Move the left lower leg support to the lower position. This is the “p0_min” value
2. Move the left lower leg support to the upper position. This is the “p0_max” value
3. For the right lower leg support, the values are “p1_min” and “p1_max” respectively.



```
float p0_max = 0.9;
float p0_min = -5.2;

float p1_max = 7.4;
float p1_min = 1.1;
```

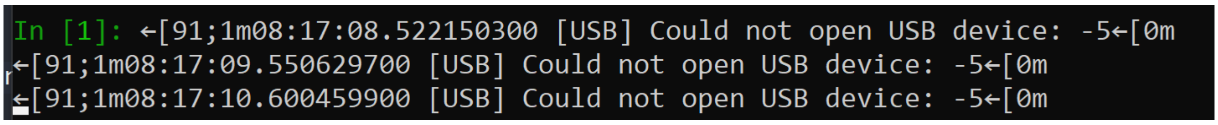
4. Connect the Arduino Mega via the USB cable to the computer USB port
5. Go to Tools:
    a. Select the appropriate **Port** “COM8” and the **Board** “Arduino Mega or Mega 2560”
6. Upload the code to the Arduino Mega by clicking the arrow button
7. The code has been uploaded will be displayed.

8. With the joystick, it is possible to move the actuator based on the position of the joystick.

Note: To choose between the two actuators, replace the `motor_num` variable in the Arduino Code for Mega. The two possibilities are 0 (right) and 1 (left).

7 Troubleshooting:

(1) Could not open USB device when running odrivetool in Command Prompt.



```
In [1]: ←[91;1m08:17:08.522150300 [USB] Could not open USB device: -5←[0m
←[91;1m08:17:09.550629700 [USB] Could not open USB device: -5←[0m
←[91;1m08:17:10.600459900 [USB] Could not open USB device: -5←[0m
```

For this, the software ‘zadig-2.7.exe’ in the Installation Files folder is required.

1. It does not require an installation. Make sure it is run as administrator.
2. When opened, select ‘Options > List All Devices’
3. On the dropdown menu:
4. First select ‘ **Odrive 3.6 CDC Interface** ’
    a. At the option ‘Driver’, on the second box, choose ‘ **USB Serial (CDC)** ‘
    b. Then select click on ‘Replace Driver’ button. The process can take some minutes.
5. Secondly, select ‘ **Odrive 3.6 Native Interface** ’
    a. At the option ‘Driver’, on the second box, choose ‘ **libusb-win32(v1.2.6.0)** ‘
    b. Then select click on ‘Replace Driver’ button. The process can take some minutes.

Afterwards, the problem should not occur again. This is dependent on the USB port being used. If the USB port is changed, the drivers might have to be installed again. In case of the error persisting, unplug the odrive USB cable, restart the odrive (switch off, wait 30 seconds and switch on again) and plug the odrive USB cable back in.