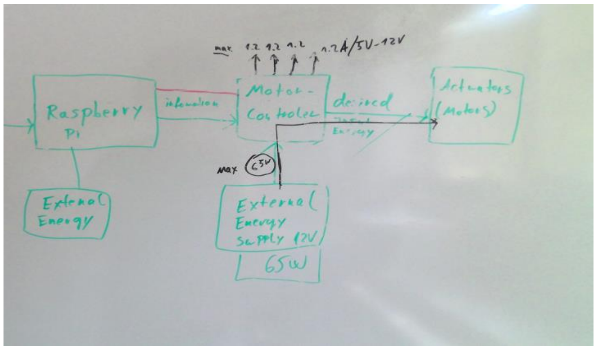
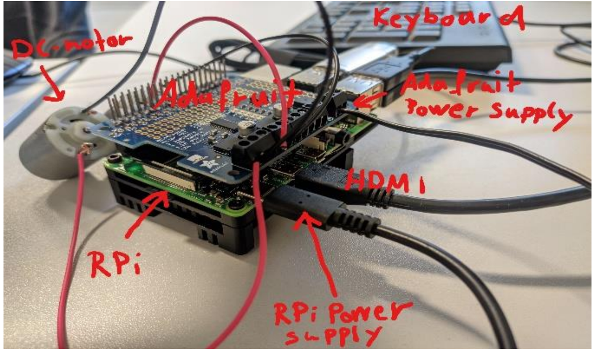
Controlling Motors via Raspberry Pi

With GUI (Graphical Userinterface), DC-motors, Servo-Motors and Adafruit DC+Stepper Motor HAT

This tutorial explains the construction of a motor controller based on the Raspberry Pi (RPi).

The instruction will be divided into the sections: Required Hardware, Hardware construction, Power Supply, Software, Adafruit installation and GUI.



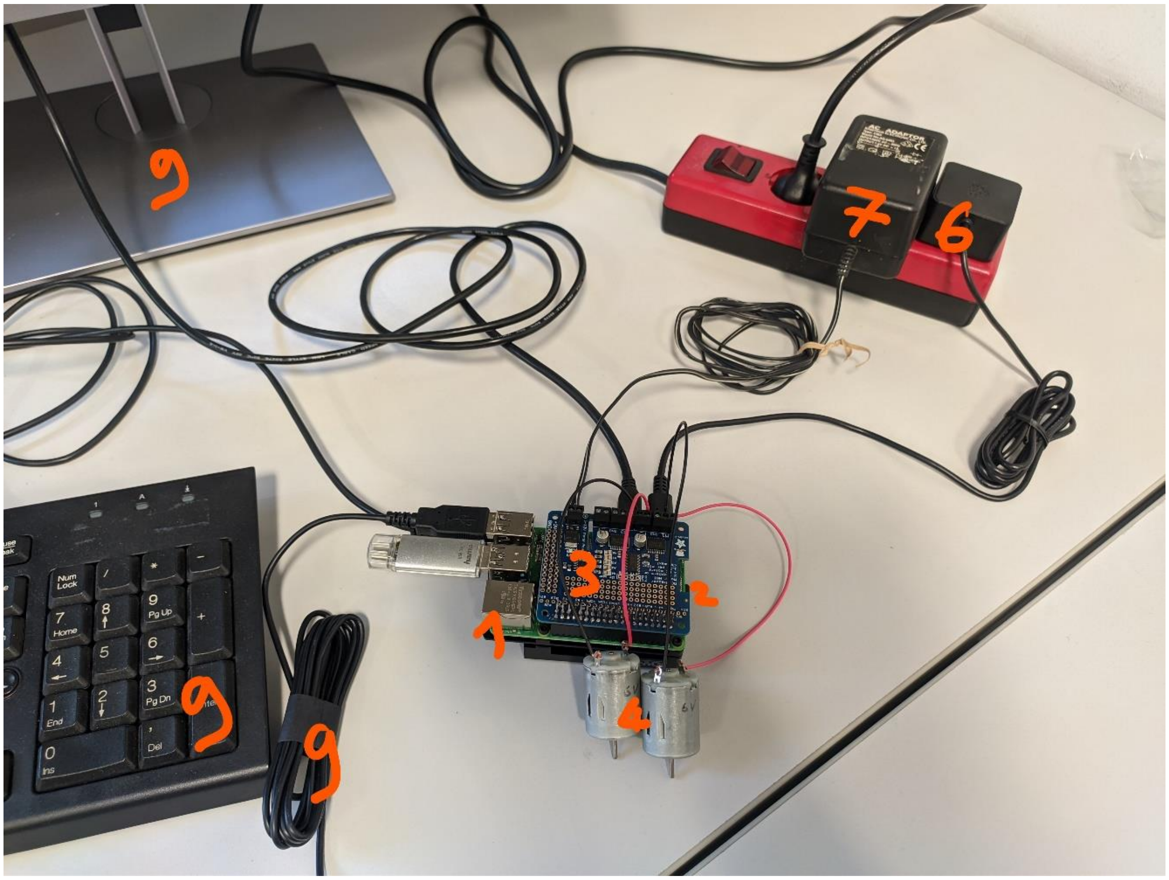


The GPIO PINS of the RPi are encircled

How it works: The RPi is connected to the Motor-controller (Adafruit), which is connected to the Actuators (2x DC-Motors + 1 Servo Motors). The Speed of the DC-motors can be controlled by changing the Voltage at the motors. That's the task of the motor controller, which regulates the external power supply driving the motors. The amount of power "given" to the motors is defined in the RPi and this information is sent to the motor-controller for execution. The motor-controller itself gets its power supply from the RPi. The Servomotor is controlled directly by the RPi via GPIO PINs. The GPIO are the 40 golden connections on the top side of the Pi. (In the bottom left figure are the GPIO PINs marked).

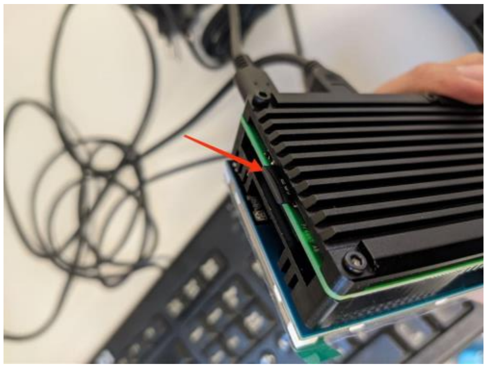
Required Hardware

1. Raspberry Pi (here: Raspberry Pi 4)
2. Micro SD card
3. Motor controller (here: Adafruit DC+Stepper Motor HAT)
4. DC-motors (here: 2x DC Motor: 6V/0.25A)
5. Servo/stepper motor (here: 1x Servomotor 6V, 0,2-0,6A, 120degree)
6. External Powersupply for RPi (here: Output 5V, 3A)
7. External Powersupply for Motors, output voltage and current depends on the motor controller (here: Powerbrick 5V)
8. (External Power supply for the Servo)
9. Keyboard, Mouse, Monitor for RPi
10. Soldering Equipment
11. Wires for RPi and wire cutter Equipment



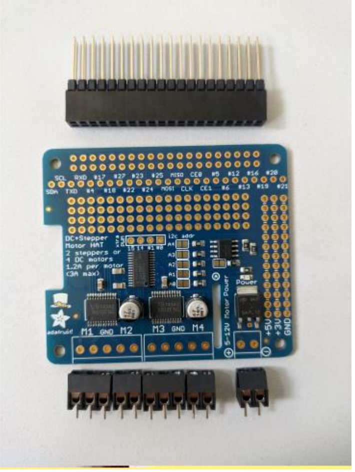
Not all components are shown in this figure. The servo motor (5) and its power supply (8) are not included. As well as the wire cutter (11) and the soldering equipment (10).

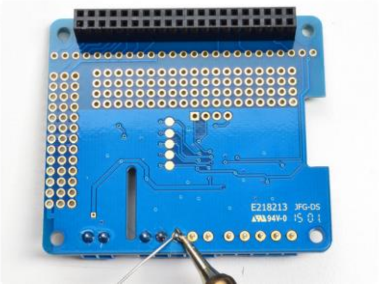
Hardware Construction



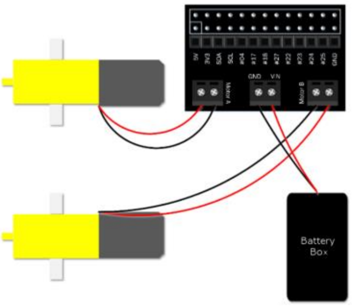
1) Put the micro SD on your PC. Download and install the 'Raspberry Pi Imager' Software, to write the OS to the SD-card (https://www.raspberrypi.org/software). Then put it pack in the RPi. Turn on the RPI and set up a password, wifi and your keyboard Layout.

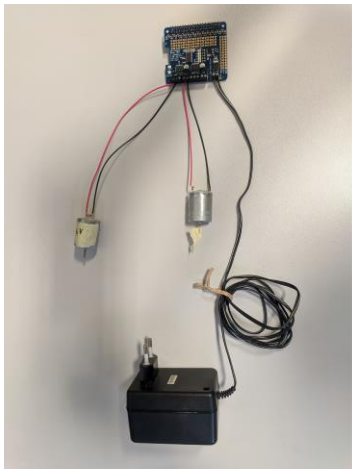
2) Solder the components of the Adafruit in the right positions. Make sure that the stacking component (component in the photo below: above the adafruit) has long metal PINs.

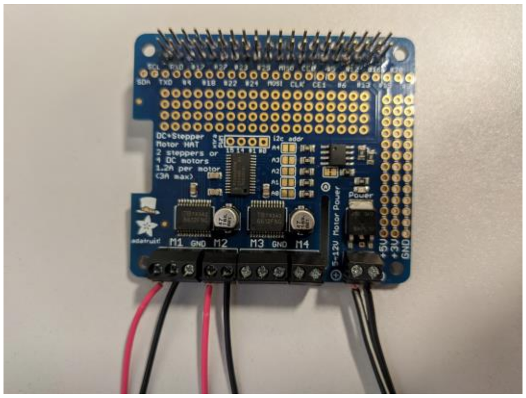




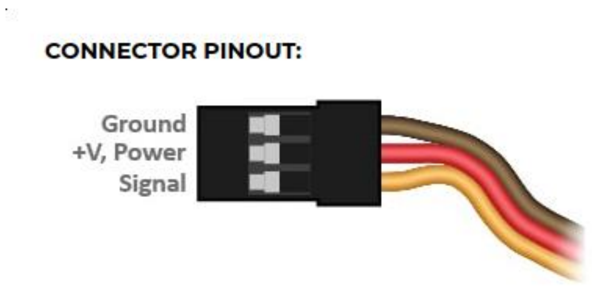
3) Connect the DC Motors to the M1, M2, (M3, M4) connections and the power supply to the "5-12V Motor Power" connection (ATTENTION: CORRECT POLES +-) of the adafruit. The amount of power supply will be covered in the Power Supply section.

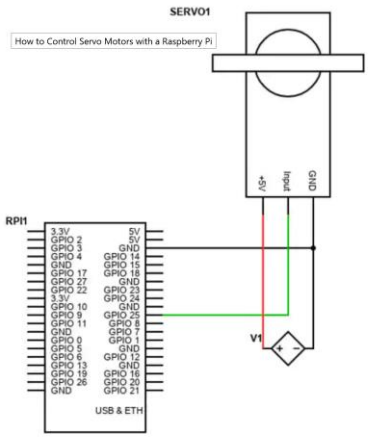




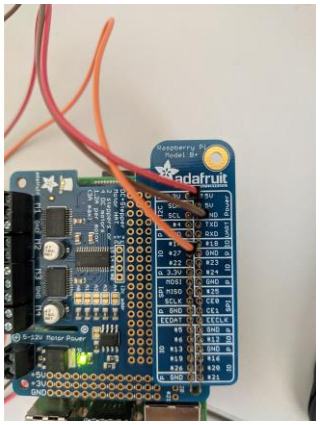


4) Connect the Servomotor to the GPIO PINS and to the power supply. Important is that the servo motor has 3 different wires. The black(brown) wire must be grounded AND connected to the negative power supply. The red wire must be connected to the positive power supply. The yellow(orange) wire must be connected to one GPIO of the RPi, since the servo position is controlled via PWM. It doesn't matter whether it is connected to GPIO 17 or GPIO 25 or another number. It is only important to be connected to one GPIO and to consider the chosen number when you write the code for the controls.





The green wire supposed to be yellow(orange)



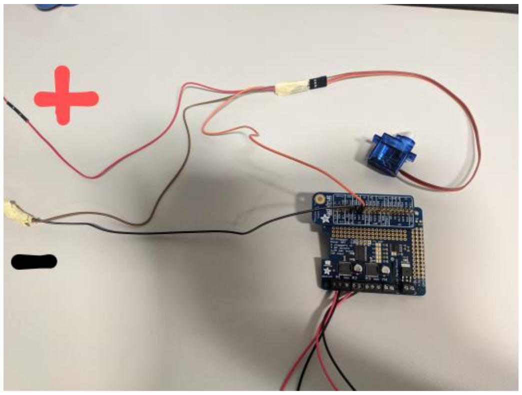
Here are two options to supply the servo with power:

a) Directly from the RPi GPIO PINs (not recommended, due to voltage drops):

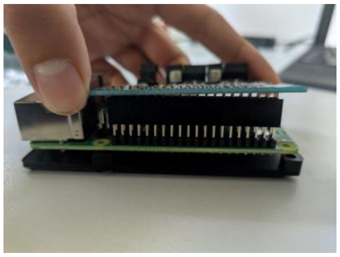
- Red wire to the 5V PIN
- Black wire to GND PIN
- Yellow wire to a GPIO PIN number

b) With external Power supply (recommended):
It can be either a battery or power brick. The max power output of the external power supply must match the requirements of the motor. (If motor needs max 5V, then the power supply should not exceed 5V)

- Red wire to the positive supply voltage
- Black wire to GND PIN of RPi AND negative supply voltage
- Yellow wire to a GPIO PIN number of RPi



5) Attach the Adafruit to the Raspberry Pi via the GPIO Pins



Power supply (Explanation)

Each of the hardware components needs (separate) external power supply (Plug brick or Batteries). These are the RPi, Adafruit, DC-motor and the servo-motor. Be aware that each component can have different requirements/limitations of voltage and current supply.

- The Adafruit motor controller itself is powered by the RPi as an energy source, since its directly connected via GPIO.
- The DC-motors, which are controlled by the Adafruit, needs another external power supply. The Adafruit board has voltage requirements/limits of 5/12V (min 5V/ max 12V). This power supply is connected to the ADAFRUIT board!!! Since each of the 4 motor-output bridges (output: M1, M2, M3, M4) of the Adafruit are limited to 1.2A/12V, EACH of the DC-motors can only supplied by maximum 1.2A/12V. The Adafruit can change the motors speed by varying the voltage between 0V-input voltage.
- Our DC-Motors expects max. 6V power supply. This means the motor power supply connected to the Adafruit should be around 6V. If we use an 12V power supply instead, we would measure ALL our motors showing 12V output voltage, if we run it at maximum speed, which is not good. Be aware that the external power supply must match BOTH voltage/current requirements of the DC-motors AND the Adafruit. For example, an Adafruit (min. 5V) and a small motor (max. 3V) would not work together.
- The Servomotor needs its own external power supply, which is connected directly to the motor. Make sure the external power voltage and current matches the motor requirement.
- If your external power supply source matches the requirements for all components, you can us it to run all/many of the components.

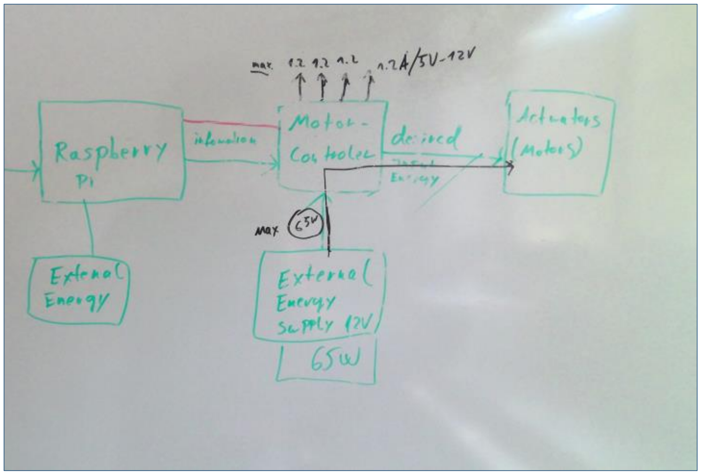


Image: This image shows an overview how it works. The servo motor and its own external power supply is not shown in the overview. Since each 4 bridges of the motor controller (Adafruit) has a max output of 1.2A and 12V each when running 4 motors simultaneously, the external energy supply can be max. 65 W.

Software: Adafruit Installation

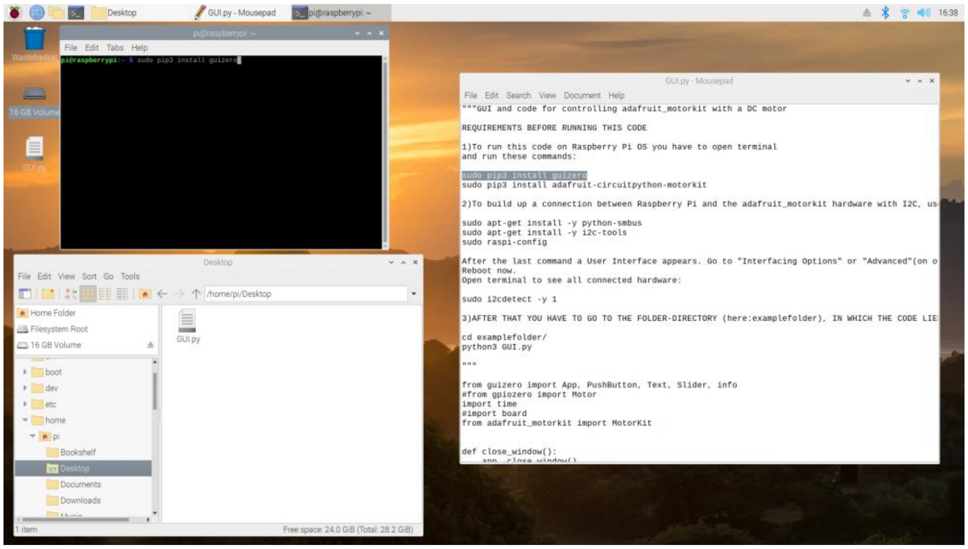
1) Now with the hardware connected to the monitor, mouse and keyboard, you have to turn on the RPi. Now we have to build up a connection between Raspberry Pi and the Adafruit_motorkit hardware with I2C. I2C is an interface connecting both RPi and Adafruit_motorkitmainboards. Open the terminal on your RPi and execute these terminal commands: (tutorial source: https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c).

```
sudo apt-get install -y python-smbus

sudo apt-get install -y i2c-tools

sudo raspi-config
```

2) After the last command a User Interface appears. Go to "Interfacing Options" or "Advanced"(on older versions). Then go to "I2C" and select "yes" to enable ARM I2C Interface.
3) Open terminal to see all connected hardware:

```
sudo i2cdetect -y 1
```

Software: Code and GUI



```
pi@raspberrypi:~ $ sudo pip3 install guizero
```

```
"""GUI and code for controlling adafruit_motorkit with a DC motor

REQUIREMENTS BEFORE RUNNING THIS CODE

1)To run this code on Raspberry Pi OS you have to open terminal
and run these commands:

sudo pip3 install guizero
sudo pip3 install adafruit-circuitpython-motorkit

2)To build up a connection between Raspberry Pi and the adafruit_motorkit hardware with I2C, us

sudo apt-get install -y python-smbus
sudo apt-get install -y i2c-tools
sudo raspi-config

After the last command a User Interface appears. Go to "Interfacing Options" or "Advanced"(on o
Reboot now.
Open terminal to see all connected hardware:

sudo i2cdetect -y 1

3)AFTER THAT YOU HAVE TO GO TO THE FOLDER-DIRECTORY (here:examplefolder), IN WHICH THE CODE LIE

cd examplefolder/
python3 GUI.py

"""

from guizero import App, PushButton, Text, Slider, info
#from gpiozero import Motor
import time
#import board
from adafruit_motorkit import MotorKit


def close_window():
    app.close_window()
```

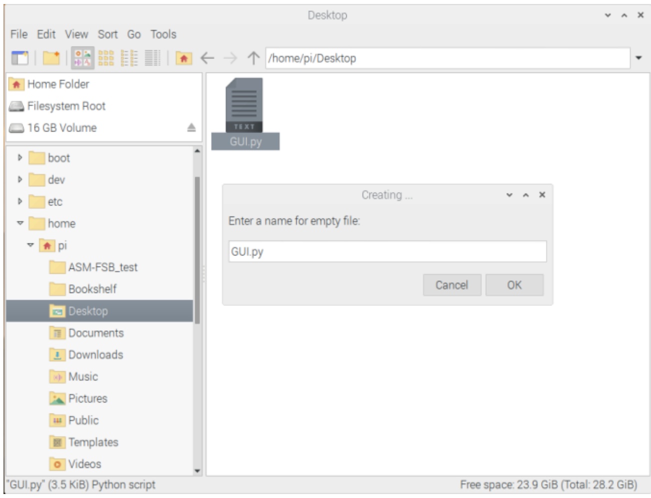
1) For creating a Graphical User Interface (GUI) we are going to code using python and python libraries. Open the Terminal on your RPi and execute this command to find out whether python 3.x is already installed:

```
python3 --version
```

2) Now install the necessary python libraries with these commands:

```
sudo pip3 install guizero

sudo pip3 install adafruit-circuitpython-motorkit

sudo pip3 install RPi.GPIO
```

3) Open or create a folder in your home directory and create a new file named GUI.py:



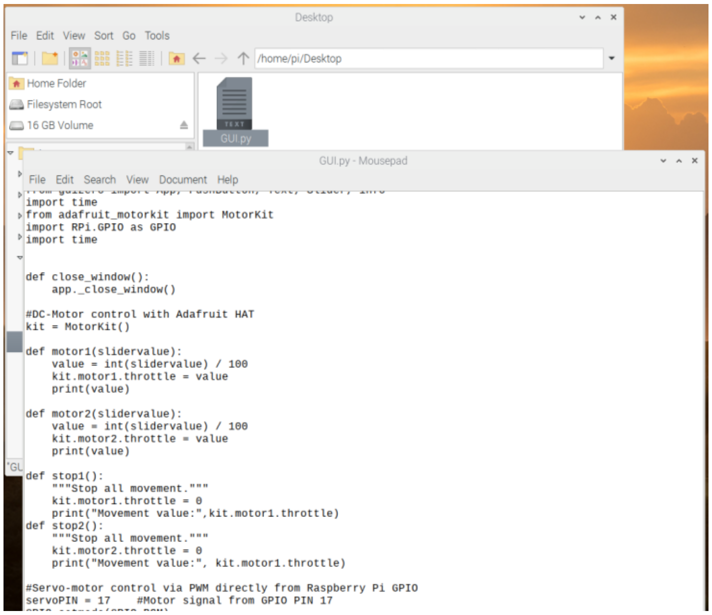
Enter a name for empty file:

GUI.py

As you can see, I put my GUI.py file in the directory:

/home/pi/Desktop

4) Open the GUI.py file and copy&paste the following code inside that file:



```
from guizero import App, PushButton, Text, Slider, Info
import time
from adafruit_motorkit import MotorKit
import RPi.GPIO as GPIO
import time


def close_window():
    app._close_window()

#DC-Motor control with Adafruit HAT
kit = MotorKit()

def motor1(slidervalue):
    value = int(slidervalue) / 100
    kit.motor1.throttle = value
    print(value)

def motor2(slidervalue):
    value = int(slidervalue) / 100
    kit.motor2.throttle = value
    print(value)

def stop1():
    """Stop all movement."""
    kit.motor1.throttle = 0
    print("Movement value:",kit.motor1.throttle)
def stop2():
    """Stop all movement."""
    kit.motor2.throttle = 0
    print("Movement value:", kit.motor1.throttle)

#Servo-motor control via PWM directly from Raspberry Pi GPIO
servoPIN = 17    #Motor signal from GPIO PIN 17
GPIO.setmode(GPIO.BCM)
```

Below is the code:

```
"""GUI and code for controlling adafruit_motorkit with a DC motor
from guizero import App, PushButton, Text, Slider, info
import time
from adafruit_motorkit import MotorKit
import RPi.GPIO as GPIO
import time



def close_window():
app._close_window()

#DC-Motor control with Adafruit HAT
kit = MotorKit()

def motor1(slidervalue):
value = int(slidervalue) / 100
kit.motor1.throttle = value
print(value)

def motor2(slidervalue):
value = int(slidervalue) / 100
kit.motor2.throttle = value
print(value)

def stop1():
"""Stop all movement."""
kit.motor1.throttle = 0
print("Movement value:",kit.motor1.throttle)
def stop2():
```

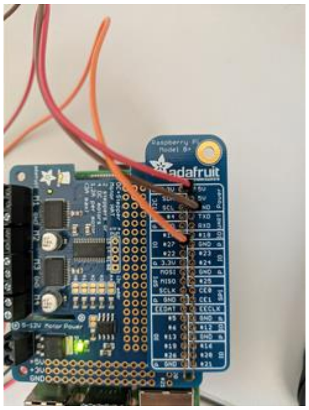
```
"""Stop all movement."""
kit.motor2.throttle = 0
print("Movement value:", kit.motor1.throttle)

#Servo-motor control via PWM directly from Raspberry Pi GPIO
servoPIN = 17    #Motor signal from GPIO PIN 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(servoPIN, GPIO.OUT)

p = GPIO.PWM(servoPIN, 50) # GPIO 17 for PWM with 50Hz
p.start(7.5) # Startposition

"""
With p.ChangeDutyCycle(value) we can change the servo position by entering the DUTY Cycle value
p.ChangeDutyCycle(2.5)   #duty cycle of 2.5% means -90degree angle
p.ChangeDutyCycle(7.5)   #duty cycle of 7.5% means 0degree angle
p.ChangeDutyCycle(12.5)   #duty cycle of 12.5% means +90degree angle
"""
def servo(slidervalue):
value = int(slidervalue)/18 + 7.5
p.ChangeDutyCycle(value)

#GUI window (internal name: app)
app = App(title="Motor Touch Control", width=480, height=320, layout="grid")
infotext = Text(app, text="Control the speed and direction of the boat", grid=[0,0])
infotext.text_size=10
#testbutton = PushButton(app, command=testfunc, text="Testbutton", grid=[0,2])
motor1text = Text(app, text="motor1", grid=[0,3])
```

```
motor2text = Text(app, text="motor2", grid=[0,5])
servotext = Text(app, text="servo(angle)", grid=[0,7])
motor1_slider = Slider(app,height=40 , width=200, command=motor1, start=-50, end=100, grid=[0,4])
motor2_slider = Slider(app,height=40 ,width=200, command=motor2, start=-50, end=100, grid=[0,6])
servo_slider = Slider(app,height=40 ,width=350, command=servo, start=-90, end=90, grid=[0,8])
motor1stop = PushButton(app, command=stop1, text="Stop motor1", grid=[1,4])
motor2stop = PushButton(app, command=stop2, text="Stop motor2", grid=[1,6])
close = PushButton(app, command=close_window, text="EXIT", grid=[1,8])


app.display()
```

5) Be aware of the following code:

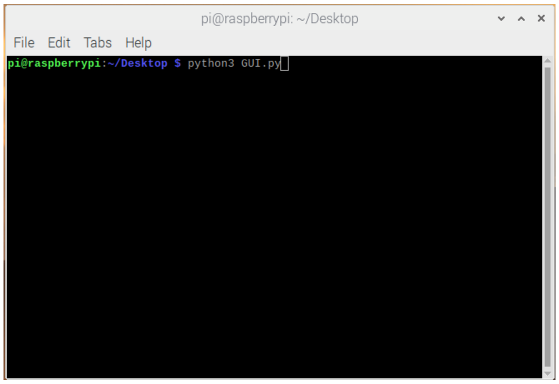
```
servoPIN = 17
```

This is the GPIO PIN number of the RPi, in which the yellow cable of the servo motor is connected. For example if it's connected to the GPIO 25, you must rename it to `servoPIN = 25`.



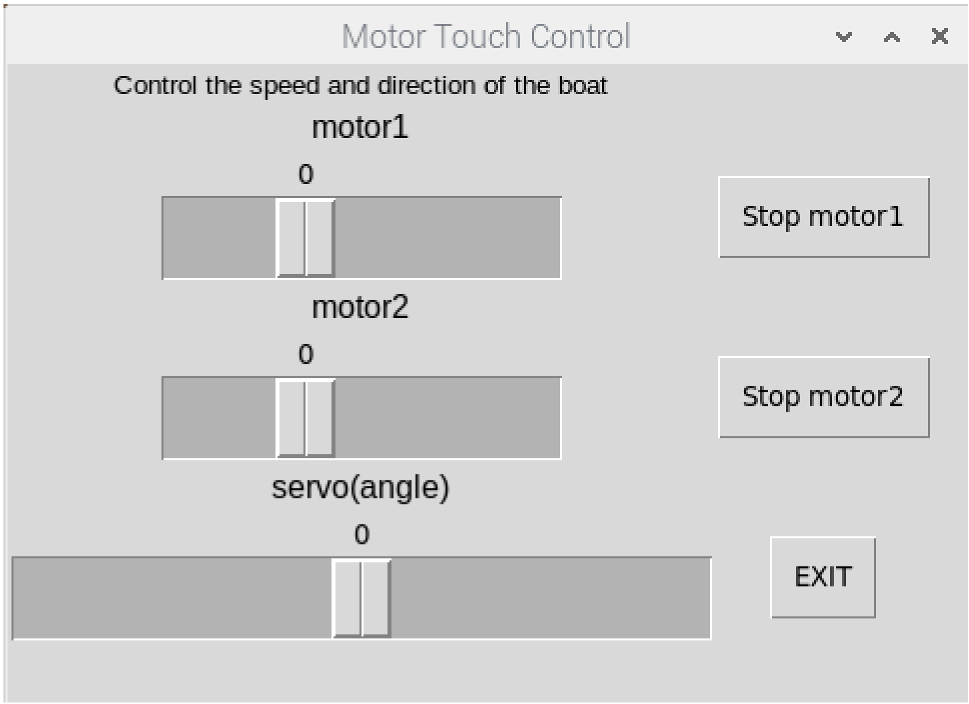
6) Open the RPi terminal and GO TO THE FOLDER-DIRECTORY (here: /Desktop), IN WHICH THE CODE LIES AND RUN IT:

```
cd Desktop

python3 GUI.py
```



```
pi@raspberrypi:~/Desktop $ python3 GUI.py
```

7) Now a GUI opens and three sliders are visible. You can control the speed of the both DC-motors (motor1 & motor 2)



Control the speed and direction of the boat

and the angle of the servo motor.