**SPOT Quickstart**

(Zdroj: [Boston Dynamics Spot documentation](https://dev.bostondynamics.com/))

# Důležité informace:

ROBOT\_IP: 192.168.80.3

Heslo k Wifi: cly8agxhvytz

Uživatelské jméno: user

Heslo: ziulppoqix0g

Uživatelské jméno správce: admin

Heslo: 6k1ad7psb2a5

Instalace a nastavení softwaru

# Instalace Pythonu

Z oficiálních webových stránek Pythonu (<https://www.python.org/downloads/release/python-370/>) stáhněte a nainstalujte Python 3.6 / 3.7 / 3.8 pomocí příslušného souboru v závislosti na operačním systému počítače.

Ujistěte se, že instalace proběhla úspěšně, a to tak, že se pokusíte spustit python z příkazového řádku:

**Linux:** $ python3.7

**Windows:** py.exe -3.7

# Instalace PIP (Package Installer for Python)

Ujistěte se, že je PIP správně nainstalován tím, že si vyžádáte jeho verzi (PIP je předinstalován s verzí Pythonu vyšší nebo rovnou 3.4 staženou z oficiálních webových stránek):

**Linux:** $ python3.7 -m pip --version

**Windows:** py.exe -3.7 -m pip --version

Pokud program PIP nenajdete, nainstalujte jej z příkazového řádku:

**Linux:** $ python get-pip.py

**Windows:** py.exe get-pip.py

# Instalace Python Image Library (PIL)

**Linux:** $ python3.7 -m pip install Pillow

**Windows:** py.exe -3.7 -m pip install Pillow

# Instalace balíčků Python pro SPOT

Nainstalujte balíčky Spot python (ujistěte se, že nainstalovaná verze je stejná jako verze nalezená v robotu Spot).

**Linux:** $ python3.7 -m pip install bosdyn-client bosdyn-mission bosdyn-choreography-client

**Windows:** py.exe -3.7 -m pip install bosdyn-client bosdyn-mission bosdyn-choreography-client

Pokud dojde k chybě nekompatibility verzí, odinstalujte a znovu nainstalujte balíčky.

**Linux:** $ python3.7 -m pip uninstall bosdyn-client bosdyn-mission bosdyn-api bosdyn-core

$ python3.7 install bosdyn-client bosdyn-mission

**Windows:** py.exe -3.7 -m pip uninstall bosdyn-client bosdyn-mission bosdyn-api bosdyn-core

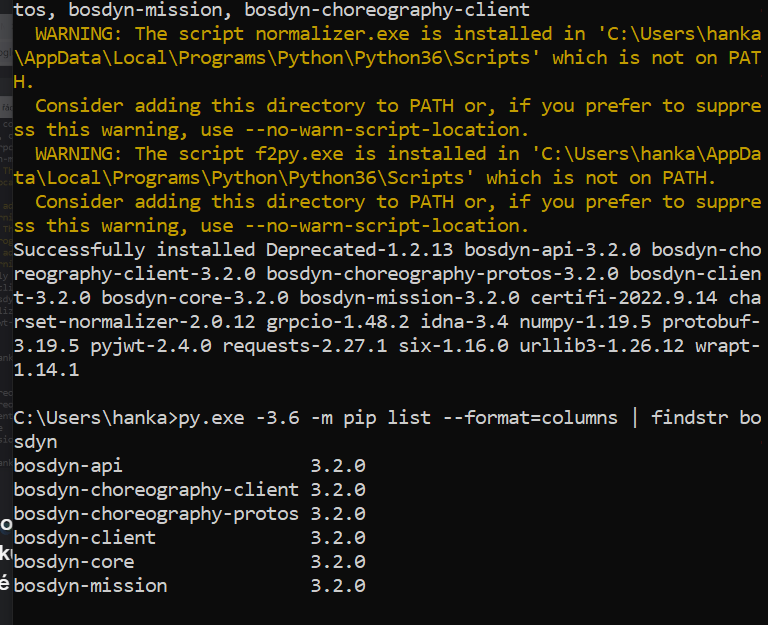
py.exe -3.7 -m pip install bosdyn-client bosdyn-mission

Poté se ujistěte, že jsou všechny balíčky správně nainstalovány. V seznamu by mělo být šest balíčků: bosdyn-api, bosdyn-choreography-client, bosdyn-choreography-protos, bosdyn-client, bosdyn-core, bosdyn-mission.

**Linux:** $ python3.7 -m pip list –format=columns | grep bosdyn

**Windows:** py.exe -3.7 -m pip list –format=columns | findstr bosdyn

Výstup:



# Stažení SDK

Kompletní distribuci SDK si stáhněte z githubu (https://github.com/boston-dynamics/spot-sdk) buď jako soubor ZIP, který je třeba rozbalit do domovského adresáře, nebo pomocí příkazu git:

Linux & Windows: git clone https://github.com/boston-dynamics/spot-sdk.git

V případě, že git není k dispozici, nainstalujte si jej z oficiálních webových stránek: <https://git-scm.com/downloads>

# Spuštění programu

1. Ujistěte se, že je tlačítko Motor Enable na zadním panelu Spota stisknuté.
2. Připojte se k Wifi Spota
3. Nezapomeňte, že musíte nejprve spustit program e-stop:
   1. Nejprve přejděte do správné složky : /spot-sdk/python/examples/estop
   2. Nainstalujte všechny závislosti :

**Linux:** python3.7 -m pip install -r requirements.txt

**Windows:** py.exe -3.7 -m pip install -r requirements.txt

* 1. Zadejte do příkazového řádku username a heslo (viz začátek dokumentu) – Pozn. Heslo se při zadávání z bezpečnostních důvodů nezobrazuje - nejsou vidět žádné znaky, tečky ani nic podobného. Nikdo nemůže vidět, jaké je heslo, nebo dokonce jeho délka! Musíte věřit, že heslo zadáváte správně.
  2. Spuštění e-stopu **s GUI :**

**Linux:** python3.7 estop\_gui.py ROBOT\_IP

**Windows:**  py.exe -3.7 estop\_gui.py ROBOT\_IP

Spuštění e-stopu **bez GUI**:

**Linux:** python3.7 estop\_nogui.py ROBOT\_IP

**Windows:** py.exe -3.7 estop\_nogui.py ROBOT\_IP

1. Ujistěte se, že Spot leží na břiše (baterií k zemi).
2. V případě spouštění příkladu programu z oficiálního githubu Boston Dynamics nejprve použijte soubor requirements.txt, který najdete spolu s programem, a automaticky nainstalujte všechny závislosti (nejprve přejděte do správné složky): /spot-sdk/python/examples/hello\_spot

**Linux:** python3.7 -m pip install -r requirements.txt

**Windows:** py.exe -3.7 -m pip install -r requirements.txt

1. Nyní můžete spustit program Python z příkazového řádku:

**Linux:** python3.7 test\_program.py ROBOT\_IP

**Windows:** py.exe -3.7 test\_program.py ROBOT\_IP

Nastavení SPOTa a příkazy

>>> import bosdyn.client

**Creation of an SDK object allowing access to the robot’s API**

>>> sdk = bosdyn.client.create\_standard\_sdk(‘any\_client\_name’)

**Creation of a robot object**

>>> robot = sdk.create\_robot(‘192.168.80.3’)

**Synchronous access to the robot-id service**

>>> id\_client = robot.ensure\_client(‘robot-id’)

>>> id\_client.get\_id()

**Asynchronous access to the robot-id service**

>>> fut = id\_client.get\_id\_async()

>>> fut.result()

**Authentication, required to access the majority of the robot’s services**

>>> robot.authenticate(‘user’, ‘password’)

**Inspection of the robot state (location, battery charge…)**

>>> state\_client = robot.ensure\_client(‘robot\_service’)

>>> state\_client.get\_robot\_state()

**List of the different available camera source names**

>>> from bosdyn.client.image import ImageClient

>>> image\_client = robot.ensure\_client(ImageClient.default\_service\_name)

>>> sources = image\_client.list\_image\_sources()

>>> [sources.name for source in sources]

**Picture from the left fisheye camera and display**

>>> image\_response = image\_client.get\_image\_form\_sources([“left\_fisheye\_image“])[0]

>>> from PIL import Image

>>> import io

>>> image = Image.open(io.BytesIO(image\_response.shot.image.data))

>>> image.show()

**Creation of a unique E-Spot and configuration to allow motor power on**

>>> estop\_endpoint = bosdyn.client.estop.EstopEndpoint(client=estop\_client, name=’my\_estop’, estop\_timeout=9.0)

>>> estop\_endpoint.force\_simple\_setup()

>>> estop\_client.get\_status()

>>> estop\_keep\_alive = bosdyn.client\_estop.EstopKeepAlive(estop\_endpoint)

>>> estop\_client.get\_status()

**Transfer of the robot’s lease to the client to allow control**

>>> lease\_client = robot.ensure\_client(‘lease’)

>>> lease\_client.list\_lease()

>>> lease = lease\_client.acquire()

>>> lease\_keep\_alive = bosdyn.client.lease.LeaseKeepAlive(lease\_client)

>>> lease\_client.list\_leases()

**Robot power on**

>>> robot.power\_on(timeout\_sec=20)

>>> robot.is\_powered\_on()

**Synchronization of Spot’s internal clock with the client’s clock**

>>> robot.time\_sync.wait\_for\_sync()

**Command to make spot stand**

>>> from bosdyn.client.robot\_command import RobotCommandClient, blocking\_stand

>>> command\_client = robot.ensure\_client(RobotCommandClient.default\_service\_name)

>>> blocking\_stand(command\_client, timeout\_sec=10)

**Command to make spot stand with his body rotated (0.4 rad) and at a certain height (0.1 m)**

>>> from bosdyn.geometry import EulerZXY

>>> footprint\_R\_body = EulerZXY(yaw=0.4, roll=0.0, pitch=0.0)

>>> from bosdyn.client.robot\_command import RobotCommandBuilder

>>> cmd = RobotCommandBuilder.synchro\_stand\_command(foorprint\_R\_body=footprint\_R\_body)

>>> command\_client.robot\_command(cmd)

>>> cmd = RobotCommandBuilder.synchro\_stand\_command(body\_height=0.1)

>>> command\_client.robot\_command(cmd)

**Robot power off (Spot will sit first if cut\_immediately==False)**

>>> robot.power\_off(cut\_immediately=False)

Příklad příkazu pro pohyb SPOTa

>>> def relative\_move(dx, dy, dyaw, frame\_name, robot\_command\_client, robot\_state\_client, stairs=False):

>>> transforms = robot\_state\_client.get\_robot\_state().kinematic\_state.transforms\_snapshot

>>> body\_tform\_goal = math\_helpers.SE2Pose(x=dx, y=dy, angle=dyaw)

>>> out\_tform\_body = get\_se2\_a\_tform\_b(transforms, frame\_name, BODY\_FRAME\_NAME)

>>> out\_tform\_goal = out\_tform\_body \* body\_tform\_goal

>>> robot\_cmd = RobotCommandBuilder.synchro\_se2\_trajectory\_point\_command(

>>> goal\_x=out\_tform\_goal.x, goal\_y=out\_tform\_goal.y, goal\_heading=out\_tform\_goal.angle,

>>> frame\_name=frame\_name, params=RobotsCommandBuilder.mobility\_params(stair\_hint=stairs))

>>> end\_time = 10.0

>>> cmd\_id = robot\_command\_client.robot\_command(lease=None, command=robot\_cmd, end\_time\_secs = time.time() + end\_time)

>>> while True:

>>> feedback = robot\_command\_client.robot\_command\_feedback(cmd\_id)

>>> mobility\_feedback = feedback.feedback.synchronized\_feedback.mobility\_command\_feedback

>>> if mobility\_feedback.status != RobotCommandFleedbackStatus.STATUS\_PROCESSING:

>>> print(“Failed to reach the goal”)

>>> return False

>>> traj\_feedback = mobility\_feedback.STATUS\_AT\_GOAL and traj\_feedback.body\_movement\_status == traj\_feedback.BODY\_STATUS\_SETTLED):

>>> print(“Arrived at the goal.”)

>>> return True

>>> time.sleep(1)

>>> return True