Přístrojová a řídicí technika

Ovládání LEGO NXT 2

Pomocí

Python 2.7 přes Bluetooth

Vypracoval: Martin Fayad

Obsah

Instalace potřebných modulů

Bluetooth

- Pomocí CMD**: 'py -2 -m pip install pybluez'**

NXT-python

- Stáhnout modul pro ovládání Lego Robota NXT 2 pomocí pythonu
- **nxt-python 2.2.2 (Musí to být verze 2.2.2)**
- https://pypi.python.org/pypi/nxt-python/2.2.2
- Pomocí CMD(e složce se soubory)**: 'py -2 setup.py install'**

Nastavení Bluetooth mezi NXT a PC

Je potřeba spárovat NXT s PC. Zapneme NXT a dále v počítači v nastavení Bluetooth spárujeme NXT s PC. Pod odkliknutí 'Spárovat' je nutno potvrdit na NXT tzv. **Passcode,** který PC bude vyžadovat. Továrně je Passcode: 1234.

Vytvoření spojení mezi PC a NXT

Nalezení ID robota NXT

Jsou 2 způsoby, jak nalézt ID robota pro připojení na Bluetooth.

1) V menu NXT robota -> Settings -> NXT Version -> ID
2) Pomocí python skriptu:

```
import Bluetooth

list_of_devices = {}
names_of_devices = []
x = bluetooth.discover_devices()
for i in x:
    i_name = bluetooth.lookup_name(i)

    list_of_devices[str(i_name)] = i
    names_of_devices.append(i_name)

print(list_of_devices)
```

Připojení přes BlueSock

Předpokládejme, že ID NXT robota je: `BRICK_ID = '00:16:53:13:C6:DE'`

Naimportujeme z modulu NXT BlueSock, pomocí kterého se připojíme na NXT robota.

```
from nxt.bluesock import BlueSock
```

a připojíme se:

```
sock=BlueSock(BRICK_ID)
brick = sock.connect()
```

brick je objekt kostky-NXT robota. Tento objekt slouží jako identita robota.

Ovládání NXT

Motory

Naimportujeme z NXT modul pro ovládání motorů A,B,C.

```
import nxt.motor as motor
```

a vytvoříme si objekty motorů,

```
motorA = motor.Motor(brick, motor.PORT_A)
motorB = motor.Motor(brick, motor.PORT_B)
```

Pro kontinuální běh motoru musíme dát příkaz **run** objektu motorA nebo motorB

Např.

```
motorA.run(100) # motorA se rozjede na 100% svého výkonu.
motorB.run(-100)# motorB se rozjede na 100% svého výkonu v opačném směru.
```

Chceme-li, aby se robot otáčel za jízdy, můžeme mu jednoduše roztočit kola jinou rychlostí.

```
motorA.run(100)
motorB.run(75)
```

Lze též použít funkci **turn,** která otočí motorem určitou rychlostí o určitý úhel.

Senzory

Senzory se ovládají poněkud obtížněji. Neboť každý senzor má jiné funkce.

Jsou 2 typy senzorů.

1) Přímo od výrobců Lega a pro získání objektu těchto senzorů stačí tato jednoduchá syntaxe:
   SOUND_SENSOR = nxt.Sound(**brick**,nxt.Port_X)
   LIGHT_SENSOR = nxt.Light(**brick**,nxt.Port_X)
   Apod.

   Seznam senzorů:

   ```
   Light
   Sound
   Ultrasonic
   Color20
   ```

   Funkce těchto senzorů lze získat pomocí **help** příkazu.
   Např. **help(nxt.Sound),** funkce jsou vyjmenované pod textem **'Methods defined here:'**
   Podrobnější popis lze nalézt ve složce nxt-python 2.2.2/Sensors.

2) HiTechnic sensory

   Seznam senzorů:

   ```
   HTCompass = hitechnic.Compass
   HTAccelerometer = hitechnic.Accelerometer
   HTGyro = hitechnic.Gyro
   HTColorv2 = hitechnic.Colorv2
   HTEOPD = hitechnic.EOPD
   HTIRReceiver = hitechnic.IRReceiver
   HTIRSeekerv2 = hitechnic.IRSeekerv2
   HTPrototype = hitechnic.Prototype
   ```

Postup je obdobný.
Např. pro Gyroskop získám objekt takto:

```
Gyroscope = nxt.sensor.HTGyro(brick,nxt.PORT_4)
```

Funkce tohoto objektu najdu příkazem **help(nxt.HTGyro)**

Příklad funkce:

```
print(Gyroscope.get_rotation_speed())
```

Příklad Ovládání NXT pomocí GUI

Python Skript (Logika):

Jméno skriptu: ***main.py***

```
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from nxt.bluesock import BlueSock
import nxt.motor as m

class MainWindow(BoxLayout): # Kivy trida

    def Connect(self): # pripojeni na kostku a vytvoreni objektu ovladani
        try:
            BRICK_ID = '00:16:53:13:C6:DE' # ID kostky
            self.sock=BlueSock(BRICK_ID)
            self.brick = self.sock.connect()
            self.motorA = m.Motor(self.brick, m.PORT_A)
            self.motorB = m.Motor(self.brick, m.PORT_B)
            print('Connected')
        except:
            print('Not Connected')

    def Stop(self): # Ruzne povely
        self.Movement(0,0)
    def Forward(self):
        self.Movement(100,100)
    def Back(self):
        self.Movement(-100,-100)
    def Left(self):
        self.Movement(0,85)
    def Right(self):
        self.Movement(85,0)
    def Movement(self,A_POWER,B_POWER):
        self.motorA.run(A_POWER)
        self.motorB.run(B_POWER)

class guiApp(App): # Kivy trida
    pass

if __name__ == '__main__':
    guiApp().run()
```

Kivy script

Jméno skriptu: ***gui.kv***
Otevřít jako txt soubor.
Pozor na odsazení.

```
MainWindow:
    orientation: 'vertical'
    Button:
        text: 'Connect'
        on_press: root.Connect()
    Button:
        text: 'Forward'
        on_press: root.Forward()
    BoxLayout:
        Button:
            text: 'Right'
            on_press: root.Right()
        Button:
            text: 'Left'
            on_press: root.Left()
    Button:
        text: 'Stop'
        on_press: root.Stop()
    Button:
        text: 'Back'
        on_press: root.Back()
```

Instalace Kivy

Instalace závislých knihoven.

Příkaz v CMD:

“py -2 -m pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew”

Instalace modulu gstreamer:

Stáhnout gstreamer pro správnou verzi Pythonu

-kivy.deps.gstreamer-0.1.7-cp27-cp27m-win_amd64.whl

nebo

-kivy.deps.gstreamer-0.1.7-cp27-cp27m-win_amd32.whl

Příkaz v CMD:

„py -2 -m pip install kivy.deps.gstreamer-0.1.7-cp34-cp34m-win_amd32.whl“

„py -2 -m pip install kivy“