

SNP –Tech Wednesday - Class #4 – 2018May30

Controlling a LEGO mindstorm using python



Markus van Kempen

E: mvk@ca.ibm.com

T: @markusvankempen

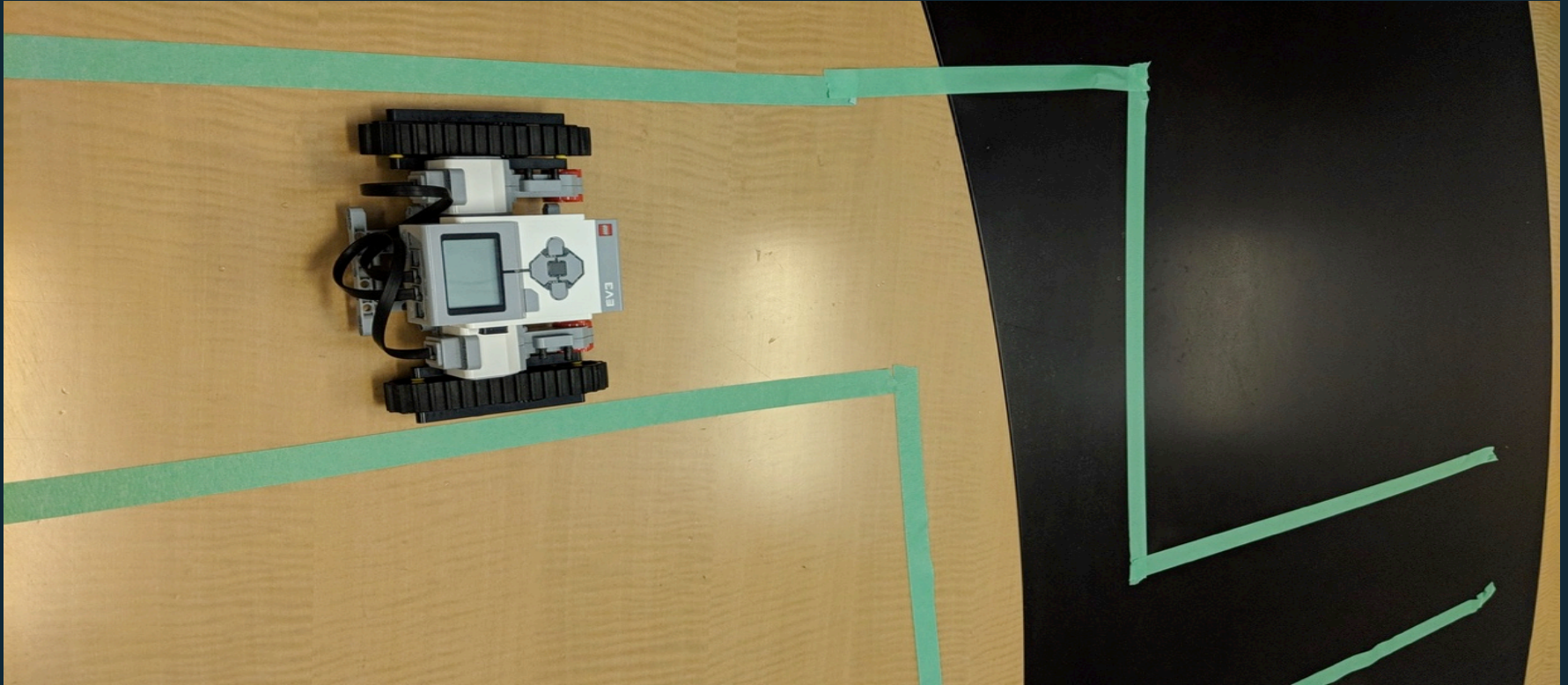
IBM SPEED

Think. Create. Win.

https://en.wikipedia.org/wiki/Six_Nations_Polytechnic



Goal – Control a Lego Robot/Rover via python code



Step 1

- Build a simple Mindstorm Rover
- Keep is simple
- Look at the example one
- Keep to 20-30 Min



Teachers Task

- Connect Pis
- Update Pis
- Write down PI IP-address
- Pair with Mindstorms once the rovers a build
- Write down Mindstorm MAC address

A little bit of Hacking



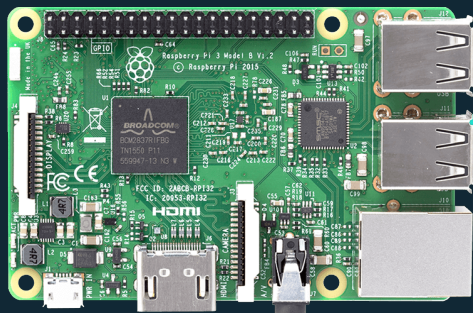
Overview of the Setup

```
pi@192.168.43.114 - Secure Shell Extension 0.8.43
Welcome to Secure Shell Extension version 0.8.43.
Answers to Frequently Asked Questions: https://goo.gl/muppJj (ctrl+click on links to open)
Changelog/release notes: https://goo.gl/YnmX0s
Major changes since 0.8.39:
  * OpenSSH upgraded to 7.6p1 (some older features dropped).
  * The default terminal encoding has changed to UTF-8.
  * Support for hyperlinking text via OSC-8.
  * Support for iTerm2 OSC-1337 inline image display: https://goo.gl/MnSysj
  * Support for extended SGR+4 underlining

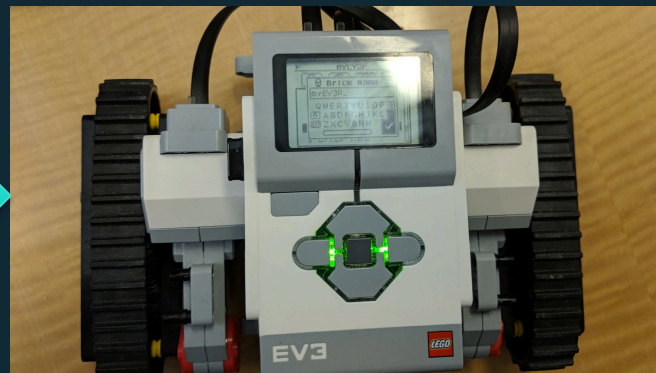
Random Pro Tip #3: Connect from the omnibox by typing 'ssh <profile name>': https://goo.gl/V7o8ki

Connecting to pi@192.168.43.114...
Loading NaCl plugin... done.
```

SSH – shell to remote control the PI

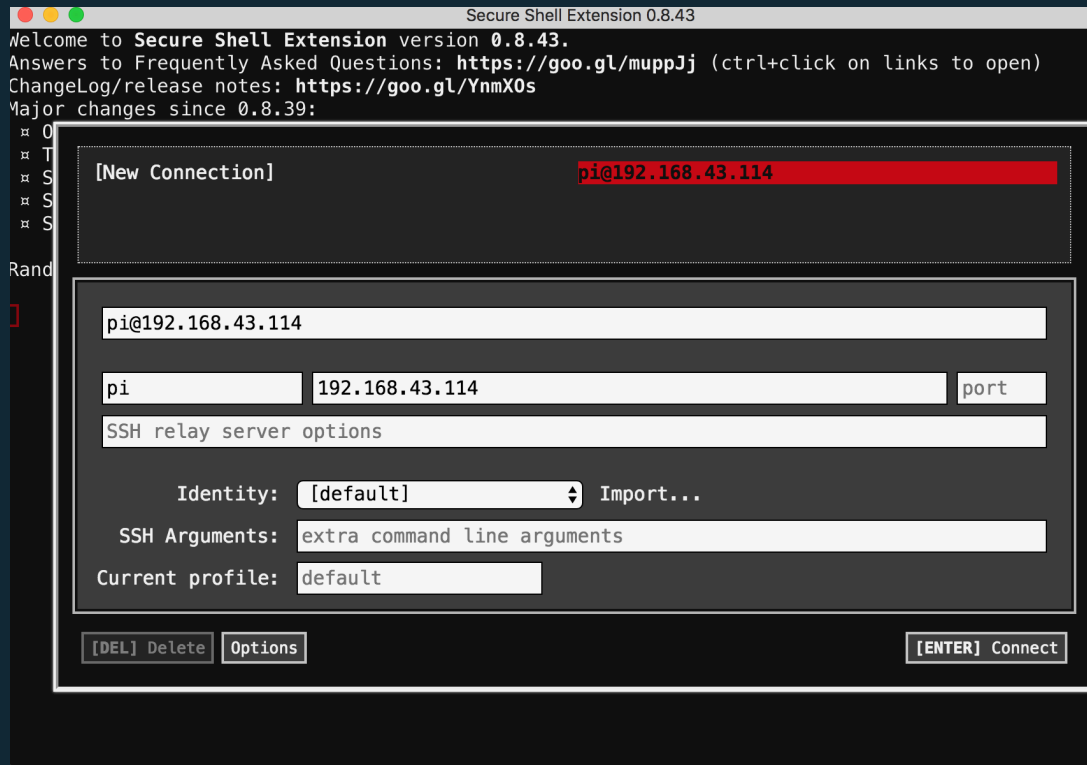


Bluetooth



Install the SSH – secure shell on your desktop / chrome extension

Launch and connect to the PI with the IP address of the Raspberry PI



Use “pi” as the username
And “raspberry”
as the password

Some important Unix commands

Note: Everything is case sensitive

ls - list the directory content

cd - change the directory

cp - copy a program like cp from to

pwd - show where you are in the directory

nano - code editor

python3 - is the programming language and also the command to execute our code

Our code is in the directory EV3 – use cd EV3 to get to the directory

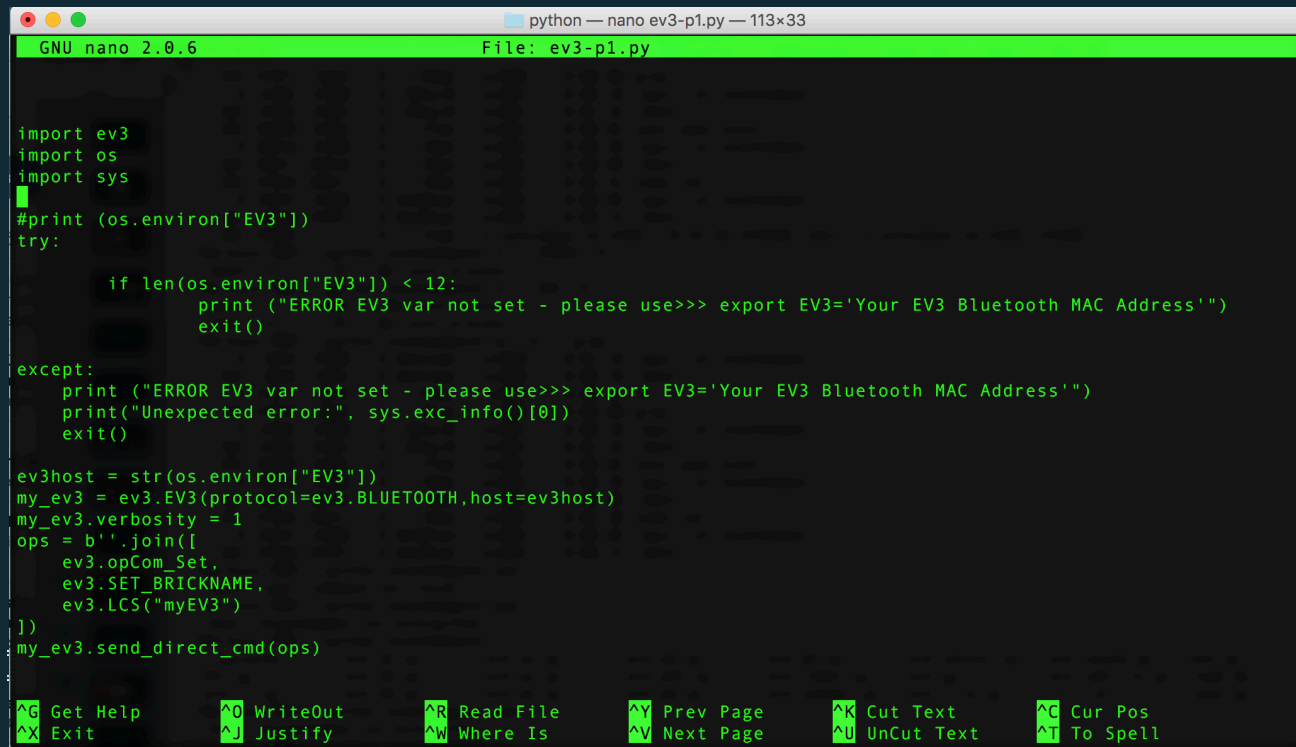
The code ends with *.py

Unix editor - nano

nano is a small editor to edit the programs

like: nano ev3-1.py

Note. Control + X will let use exit and save the program after you changed something



```
python — nano ev3-p1.py — 113x33
GNU nano 2.0.6 File: ev3-p1.py

import ev3
import os
import sys

#print (os.environ["EV3"])
try:

    if len(os.environ["EV3"]) < 12:
        print ("ERROR EV3 var not set - please use>>> export EV3='Your EV3 Bluetooth MAC Address'")
        exit()

except:
    print ("ERROR EV3 var not set - please use>>> export EV3='Your EV3 Bluetooth MAC Address'")
    print("Unexpected error:", sys.exc_info()[0])
    exit()

ev3host = str(os.environ["EV3"])
my_ev3 = ev3.EV3(protocol=ev3.BLUETOOTH,host=ev3host)
my_ev3.verbosity = 1
ops = b''.join([
    ev3.opCom_Set,
    ev3.SET_BRICKNAME,
    ev3.LCS("myEV3")
])
my_ev3.send_direct_cmd(ops)
```

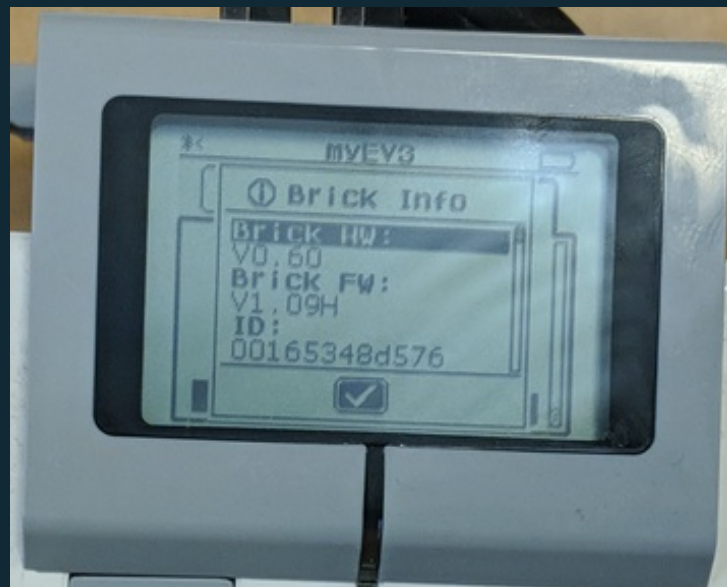
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

Set the MAC address of the EV3

We need to set the MAC address of the EV3 in the shell

The MAC Address you can find in the EV3 under
the system menu -> Brick info -> ID

Note: you have to add “:” for the address
Address /ID translates into
00:16:53:48:d5:76



Set the MAC address in the shell

```
export EV3='00:16:53:48:d5:76'
```

```
ss-MBP-3:python mvankempen$ export EV3='00:16:53:48:d5:76'
ss-MBP-3:python mvankempen$ export EV3='00:16:53:48:d5:76'
ss-MBP-3:python mvankempen$ █
```

Execute code

Run you 1st python3 program

Change to the EV3 directory – `cd EV3`

Execute the program

```
python3 ev3-1.py
```

Look at the code via nano and change it change the EV3 name to myEV(YourGroupNumber) like myEV-32

ev3-1.py = Change the name of your rover

ev3-2.py = Play a sound and switch on the LED

ev3-3.py = Driving program

Executing the 1st python program

```
python — pi@playbulb32: ~/EV3 — ssh pi@192.168.1.41 — 113x33
[pi@playbulb32:~ $ cd EV3
[pi@playbulb32:~/EV3 $ ls
ev3-1.py  ev3-2.py  ev3-3.py  ev3.py  lego.py  __pycache__
[pi@playbulb32:~/EV3 $ python3 ev3-1.py
Rename my rover and Play a Sound
Rename my rover to myEV3-##
Make some noise
program done
[pi@playbulb32:~/EV3 $
[pi@playbulb32:~/EV3 $ python3 ev3-2.py
Play a sound and Flash LED
Make some noise
Set LED to RED FLASH
Switch LED to GREEN Flash
program done
pi@playbulb32:~/EV3 $ █
```


Execute - driving program

The driving program will give you the basic commands

Examples:

```
print ("moving forward")
lego.move(20,0)    # slow forward
lego.time.sleep(5) #Program sleeps 5 seconds
print ("turn and move right")
lego.move(20,-100)
```

Note:

mode (speed, angle)

- speed of the rover +number forward – numbers backward – max speed is 50
- angle turns the rover +number left – numbers right -/+ 100 | -/+ 200 turn is circles
- sleep waits but keep the rover going
- print prints text

Execution on the driving program

Start coding a path for the rover

- Take a look at the `ev3-3.py` program
- Familiarize yourself with the `speed/move` command
- We will create a path for the rover which you have to follow using the example code

Reference

Base install

<https://github.com/markusvankempen/snp-pi-installation>

Presentation

Code:

<https://github.com/markusvankempen/snp-2018/tree/master/2018-04-LegoMindstorm1>

Video:



Markus van Kempen

Executive Architect & Venture Capitalist
IBM Corporate Strategy
Innovating with People and Technology

email: mvk@ca.ibm.com
Twitter: @markusvankempen
Hashtag☺: #MVK

