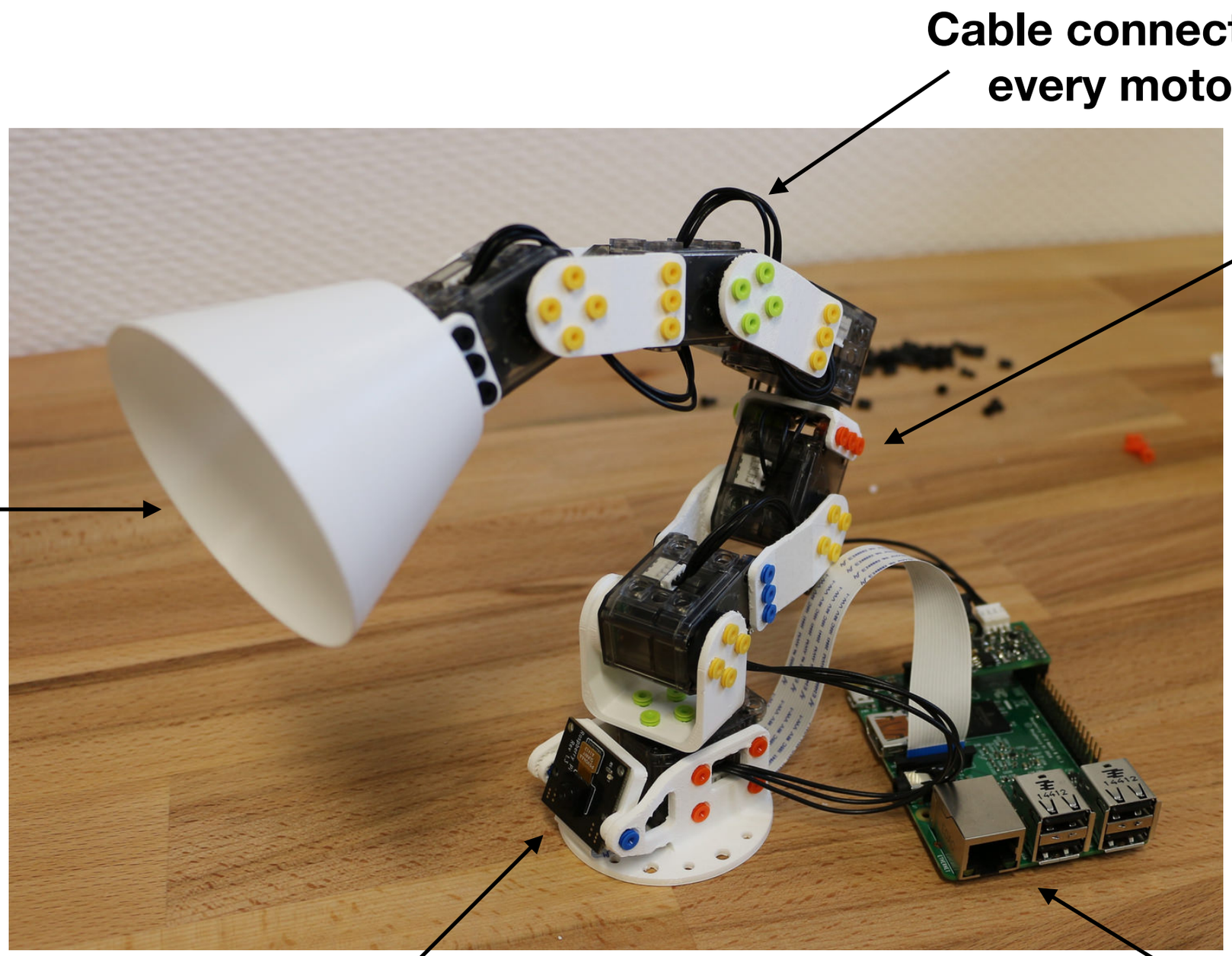
Introduction to fundamentals of Industrial Controls programming with Poppy Ergo Jr.

by Saskia Drechsel and Nahae Kühn

Outline

- What is Poppy Ergo Jr.?
- How do you work with Poppy Ergo Jr.?
- What did we do in the 2 weeks?
- What does it have to do with the CERN?

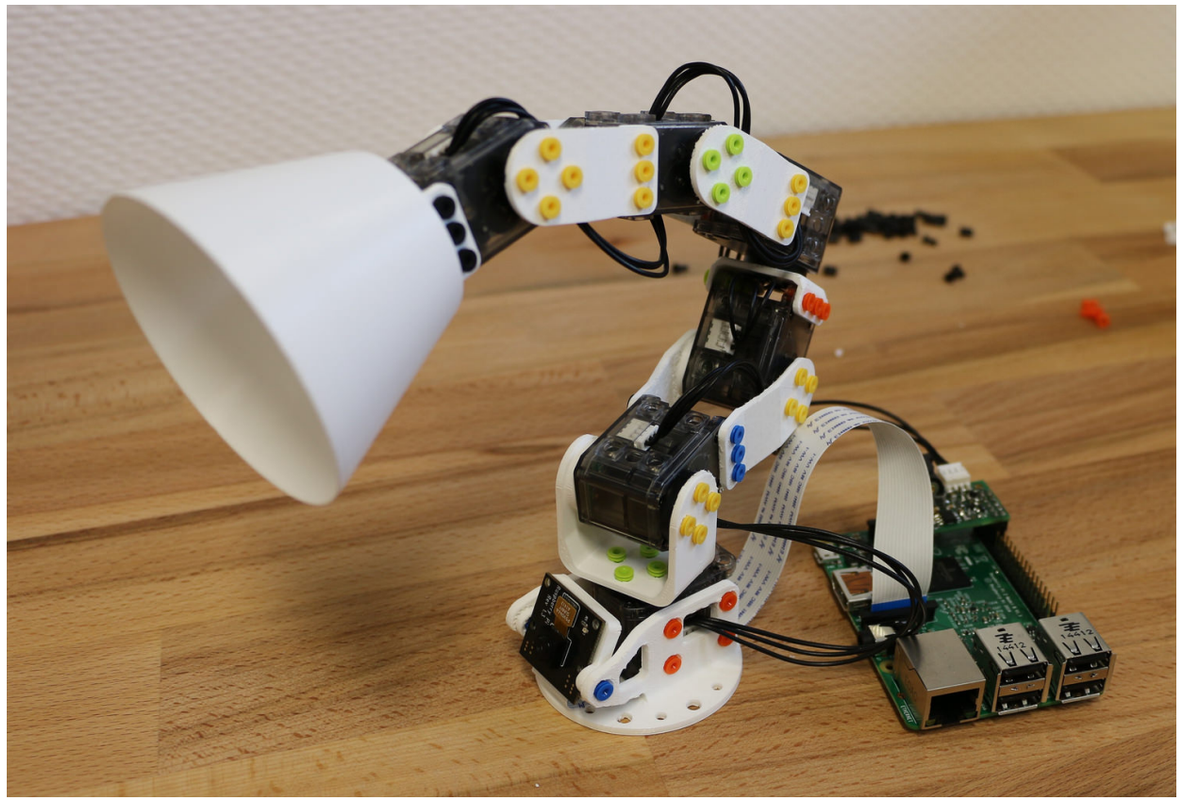
Poppy Ergo Jr.



https://raweb.inria.fr/rapportsactivite/RA2015/flowers/IMG/ergo-jr.jpg

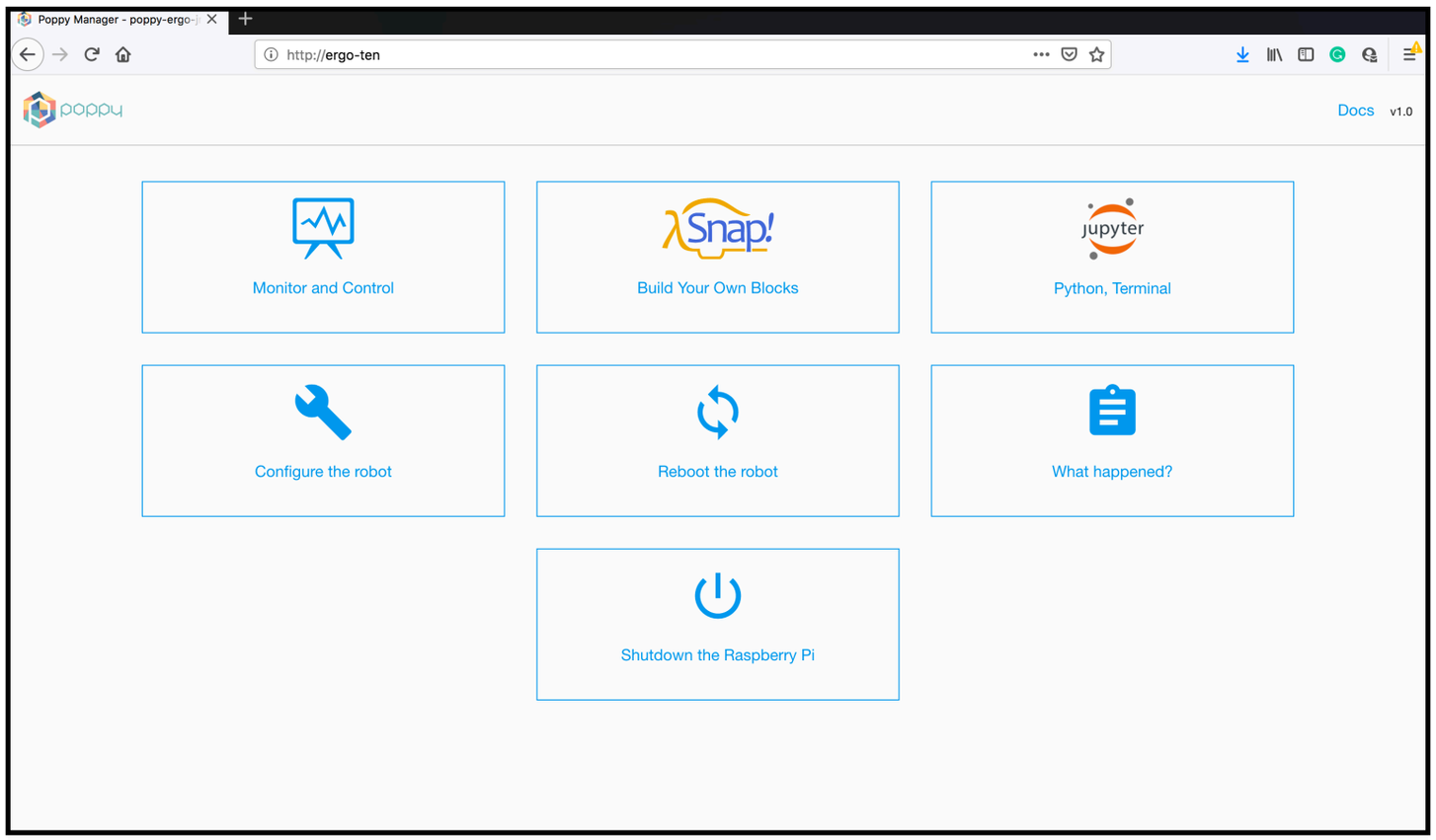
Sensors

- Temperature sensor
- Position sensor
- Moving speed sensor



Working environment

- Work over the CERN network:



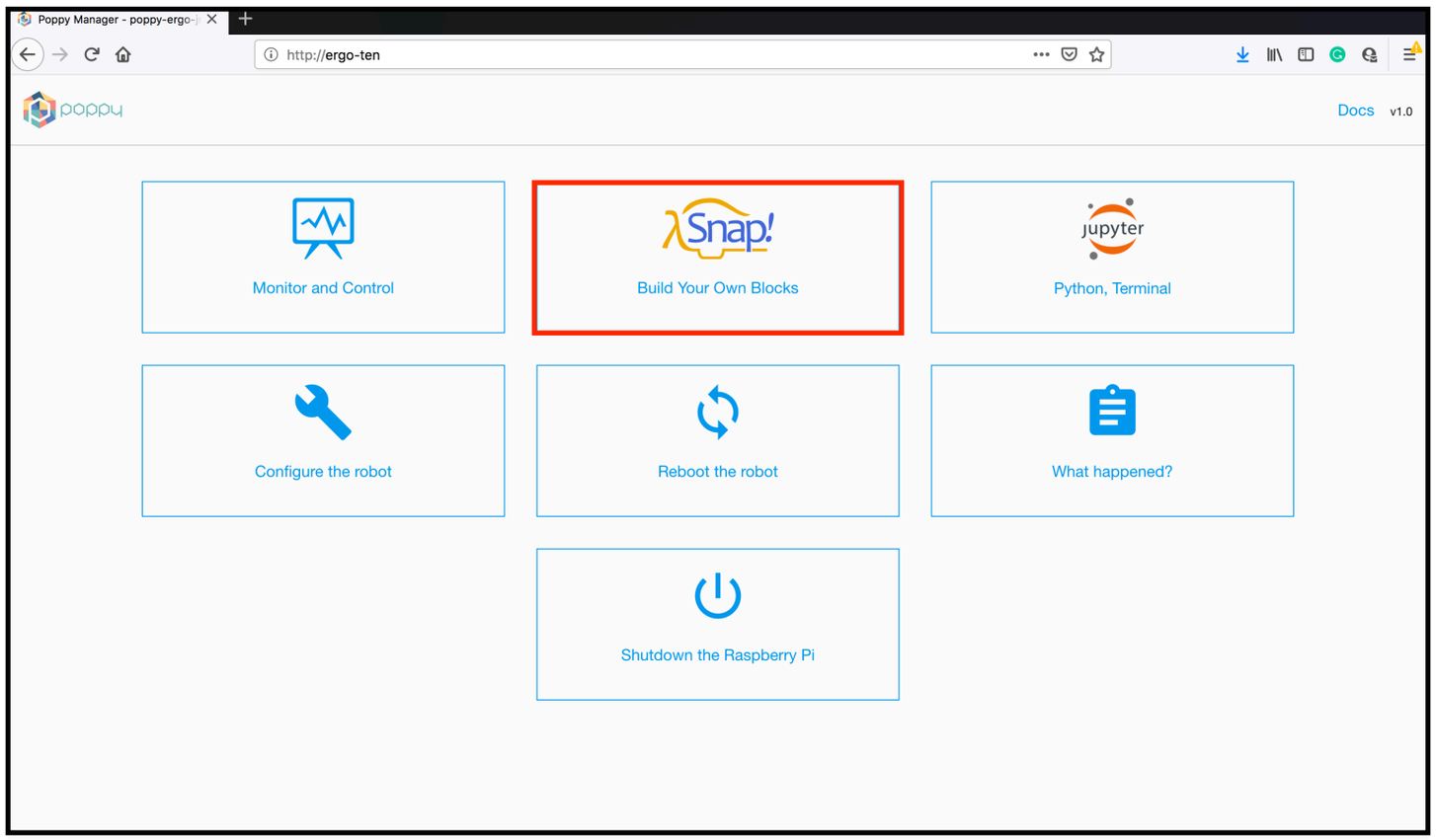
User interface

Working environment

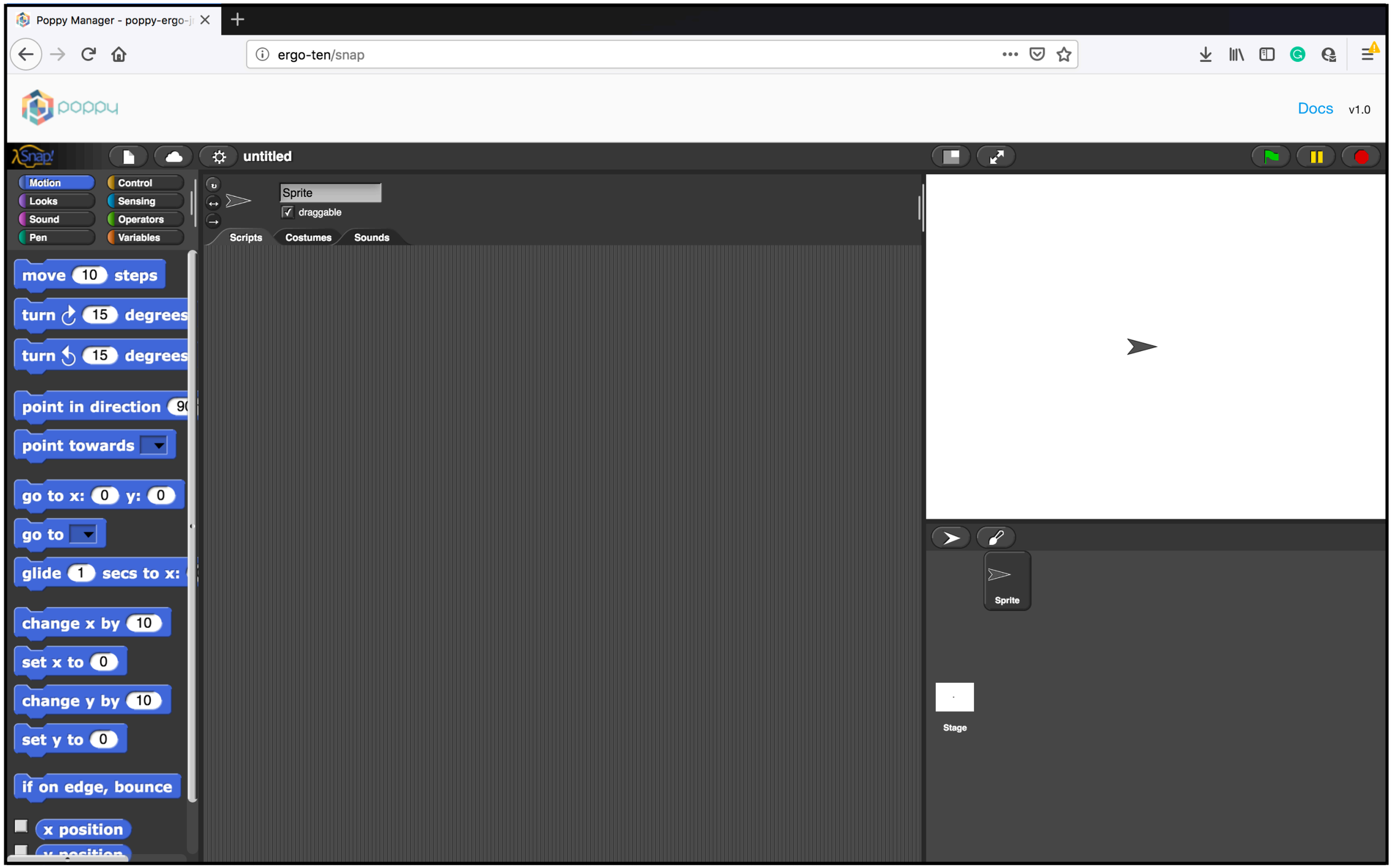
- Visual Programming (similar to Scratch)

Working environment

- Work over the CERN network:



User interface



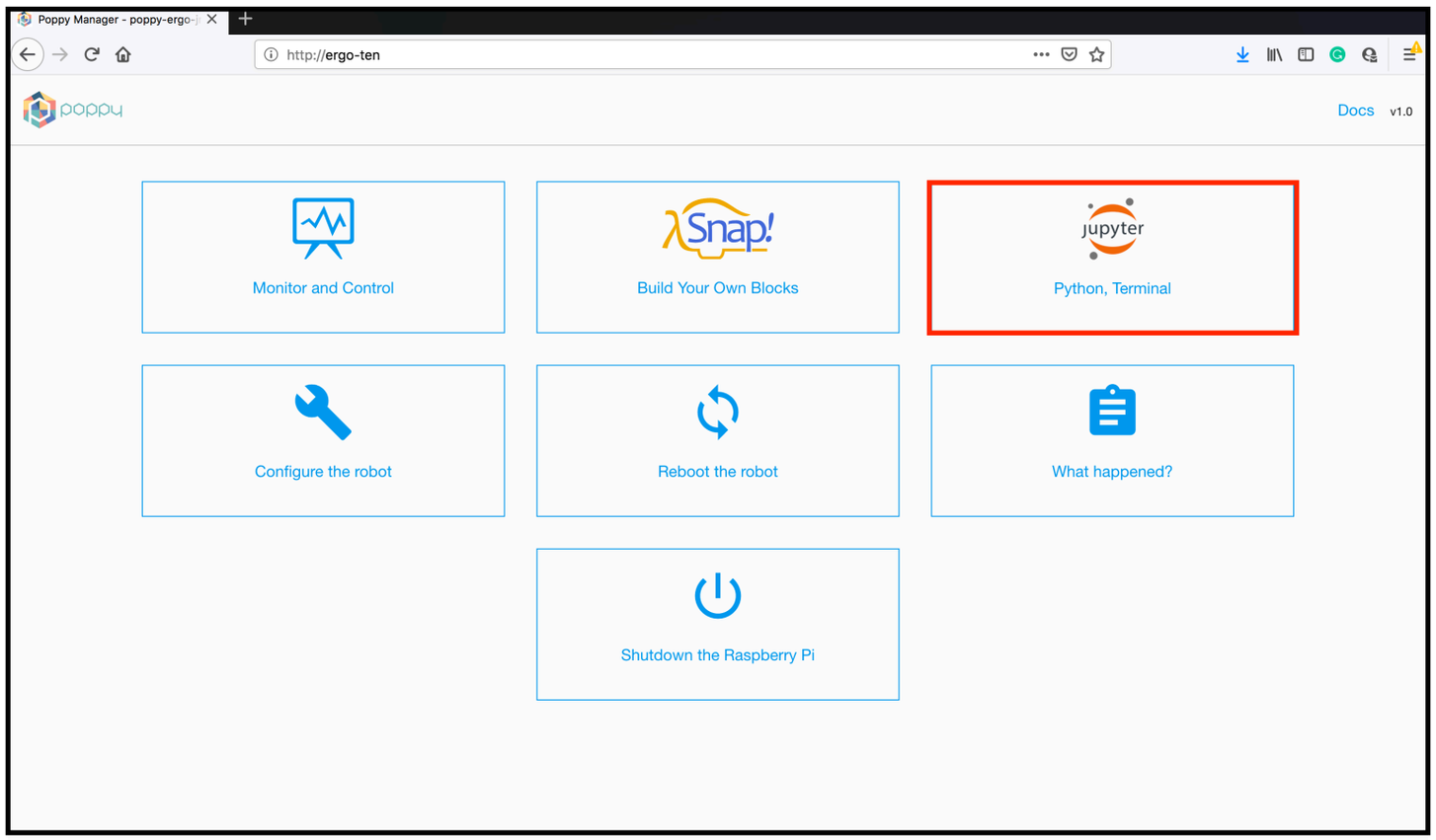
Visual Programming (Snap!)

Working environment

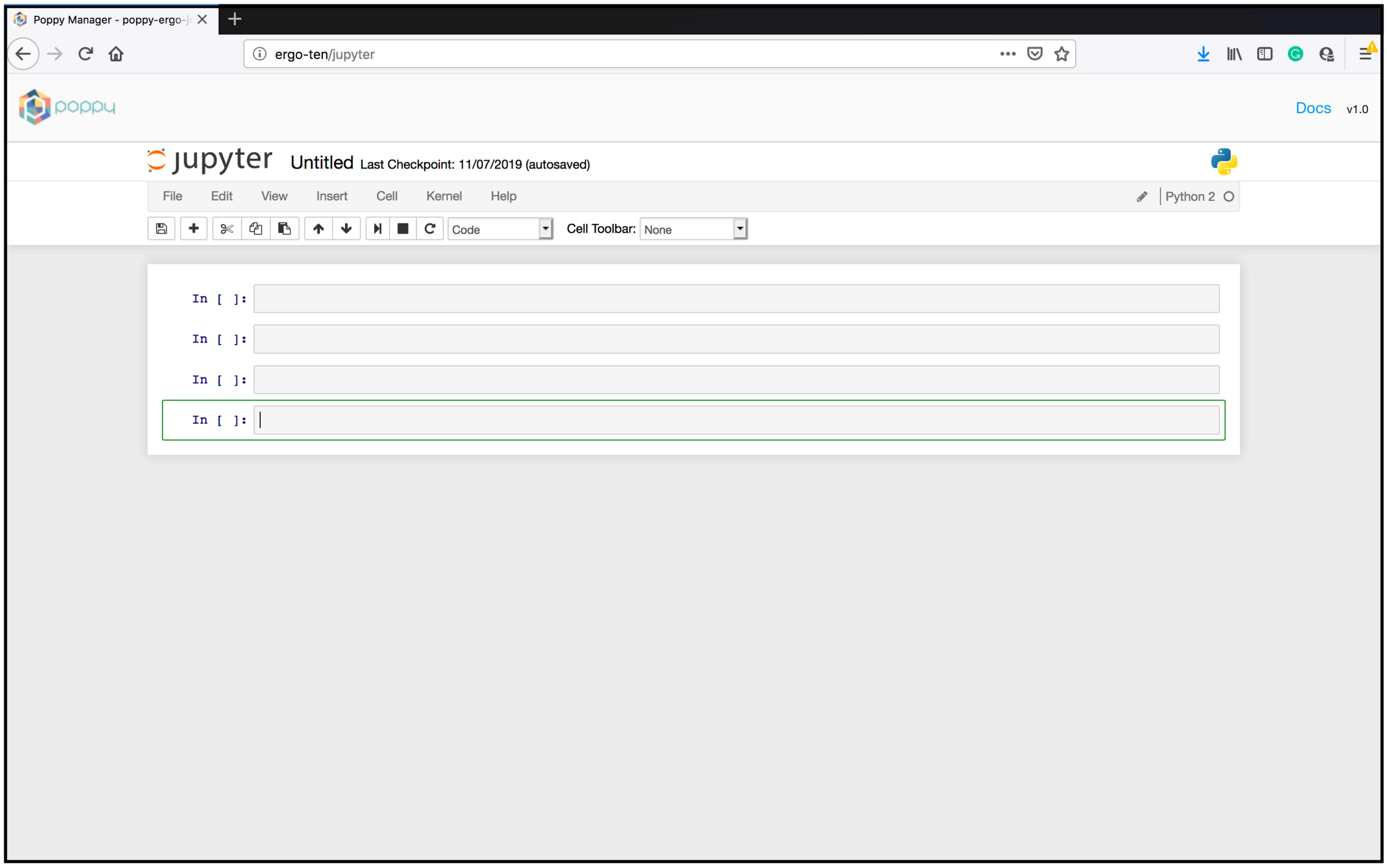
- Visual Programming (similar to Scratch)
- Programming with Python on Jupyter Notebook
- Jupyter Notebook: web-based interactive computational environment for Python programming
- code divided in cells
- you don’t have to compile the whole program at once

Working environment

- Work over the CERN network:



User interface

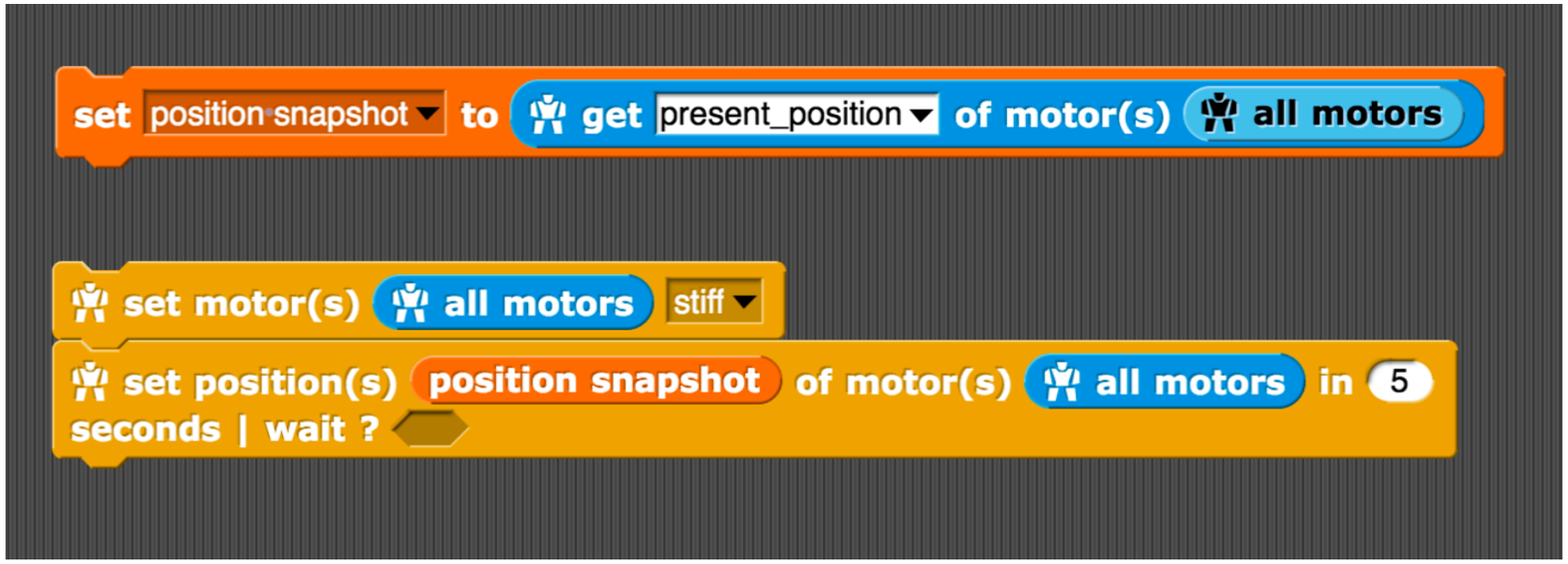


Jupyter Notebook

What we did

- Visual Programming in the beginning: programming Poppy to grab sugar cubes and to put them in a cup

Record and play the movements of Poppy



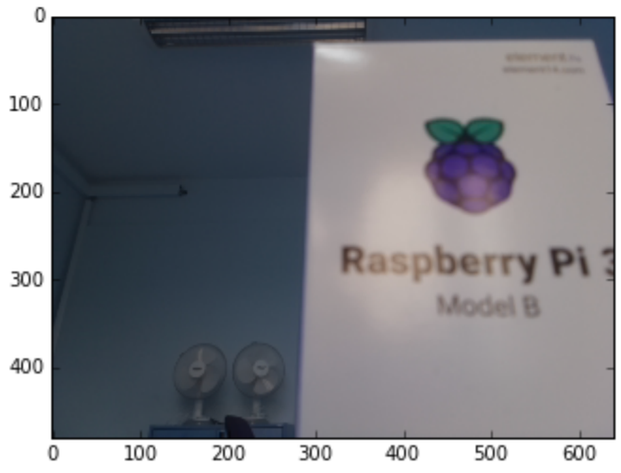
Record each position and Poppy goes through each of them individually

What we did

- Visual Programming in the beginning: programming Poppy to grab sugar cubes and to put them in a cup
- Learned programming with Python and OpenCV
- OpenCV: library of programming functions aimed at computer vision
- Image recognition: programmed Poppy to recognize objects by color and turn to them

In [260]:
```
img= poppy.camera.frame
imshow(img)
```

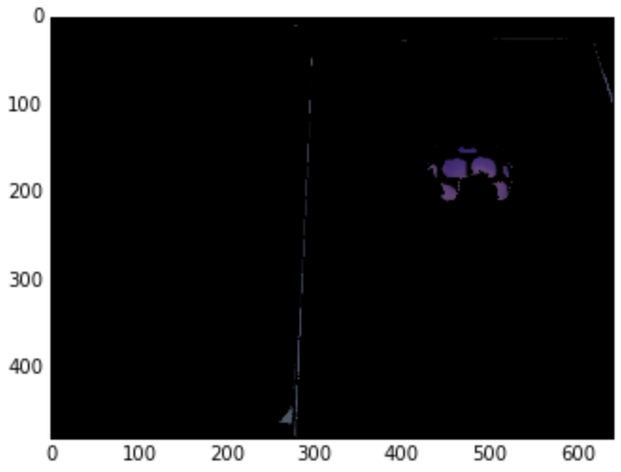
Out[260]: <matplotlib.image.AxesImage at 0x4ed09c70>



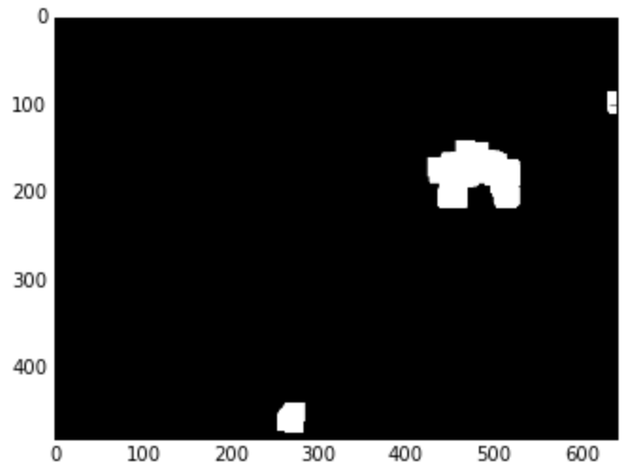
In [261]:
```
mask=cv2.inRange(img,lowerBound,upperBound)
result=cv2.bitwise_and(img, img, mask=mask)

imshow(result)
```

Out[261]: <matplotlib.image.AxesImage at 0x4e526b90>

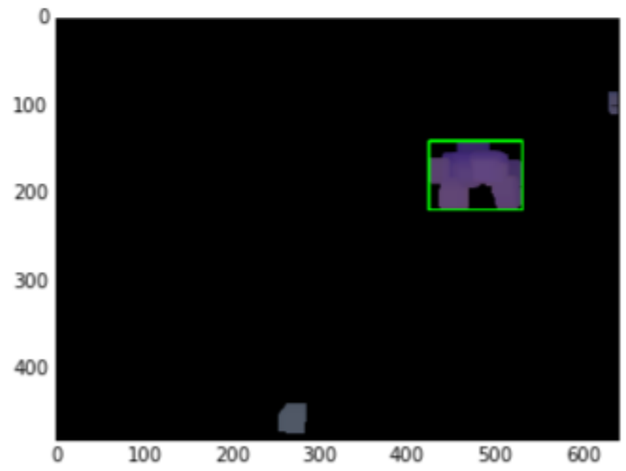


In [262]:
```
x, y, w, h = cv2.boundingRect(mask)
kernel = np.ones((3,3), np.uint8)
result2=cv2.erode(result.copy(), kernel, iterations=1)
rect2=cv2.dilate(result2.copy(), kernel, iterations=10)
ret,thresh = cv2.threshold(rect2,0,255,0)
imshow(thresh)
```



In [271]:
```
for cnt in contours:
    if cv2.contourArea(cnt) >5000:  #raspberry:5630
        x,y,w,h = cv2.boundingRect(cnt)
        rect1=cv2.rectangle(rect2.copy(),(x,y),(x+w,y+h),(0,255,0),2)
imshow(rect1)
```

Out[271]: <matplotlib.image.AxesImage at 0x52333af0>



In [269]:
```
x0, y0 = x+w/2, y+h/2
print(x0,y0)

if x0>(totalwidth/2):
    poppy.m1.goal_position =-1 * (np.arctan((x0-totalwidth/2)/700.000)*(180.000/3.14000))
else: poppy.m1.goal_position =1 * (np.arctan((totalwidth/2-x0)/700.000)*(180.000/3.14000))

print(np.arctan((x0-totalwidth/2)/30.000)*(180.000/3.14000))
```

```
476 180
79.1546009547
```


poppy
Poppy Ergo Jr v1.0
Raspberry Pi 3
Model B
6 40522 71085 0

TIM the Robot

https://kt.cern/sites/knowledgetransfer.web.cern.ch/files/styles/flexslider_full/public/images/success-stories/story-autonomous-monorail-monitoring-underground-water-pipelines.jpg?itok=nvEn8-yl

TIM the Robot

- Train inspection monorail for the LHC → Industrials Controls instance
- Used for real time measurements and inspections along the LHC tunnel
- Equipped with a radioprotection probe for radiation mapping of the LHC
- Monitors the tunnel structure, oxygen, communication bandwidth and temperature
- Provides visual and infrared imaging of the LHC
- Alarms the scientists in an emergency

https://home.cern/news/news/accelerators/meet-tim-lhc-tunnels-robot

Thank you for listening! 👂 😄