World Robot Olympiad
Future Innovators
Senior
2022

Team name:

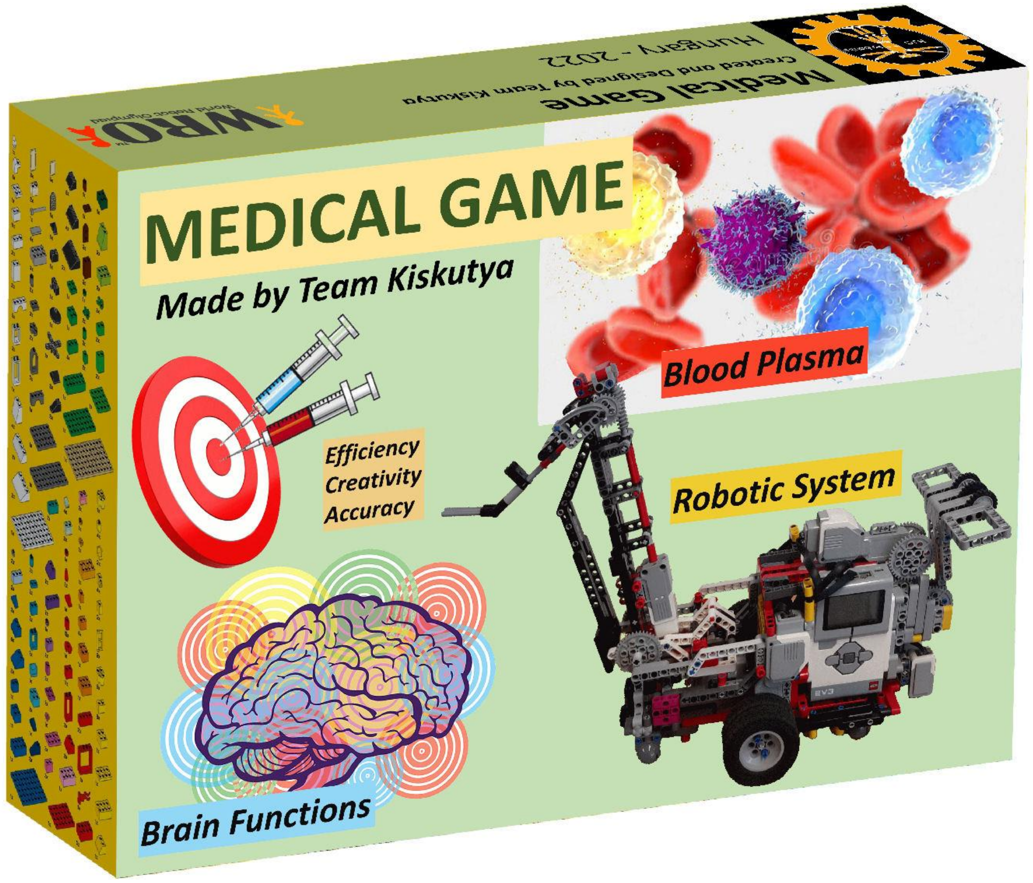
„Kiskutya”

Hédi Zita KOVÁCS-BÁNHALMI

Sára MIHALIK

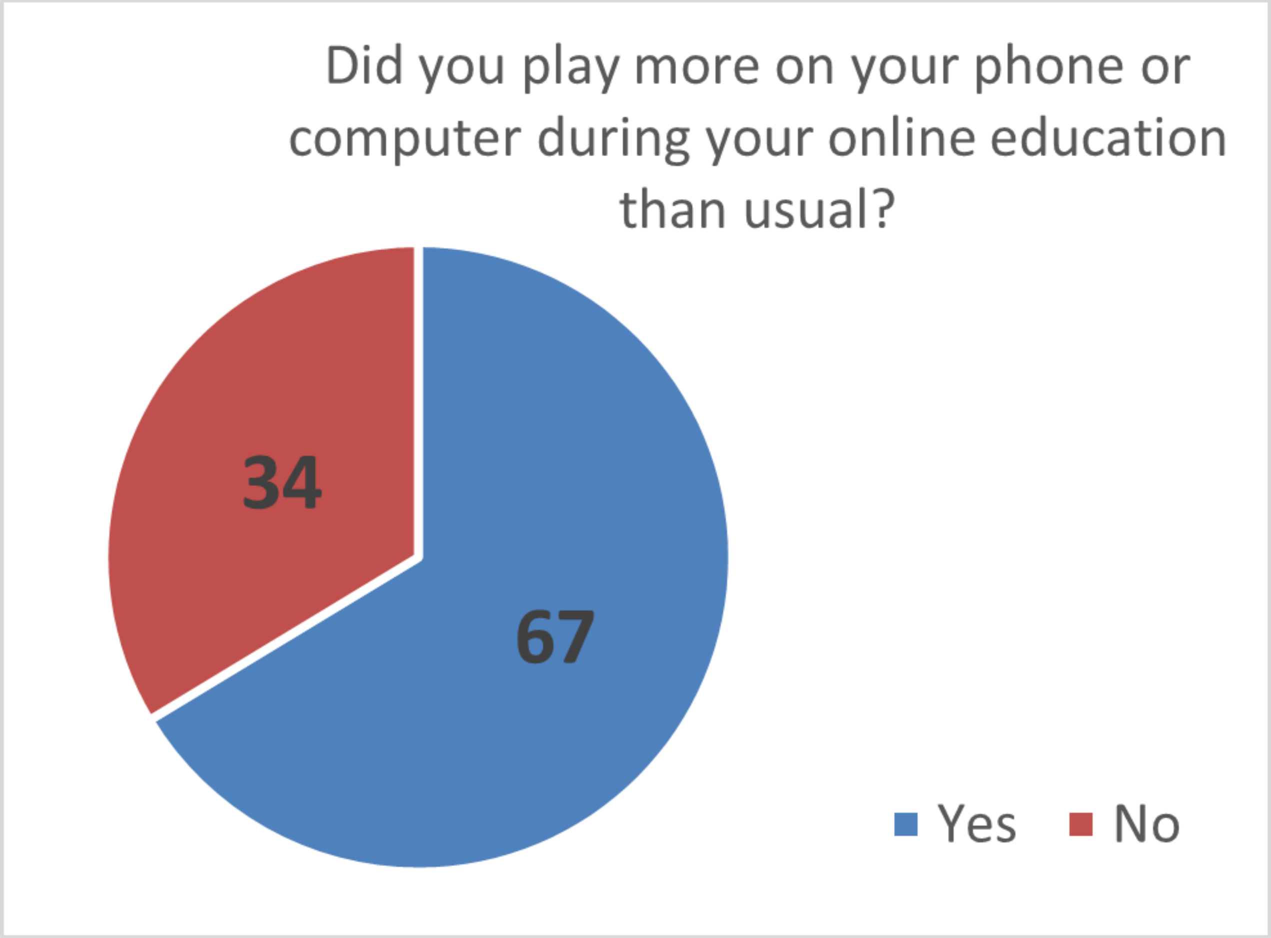
Lilla PETRÁNYI

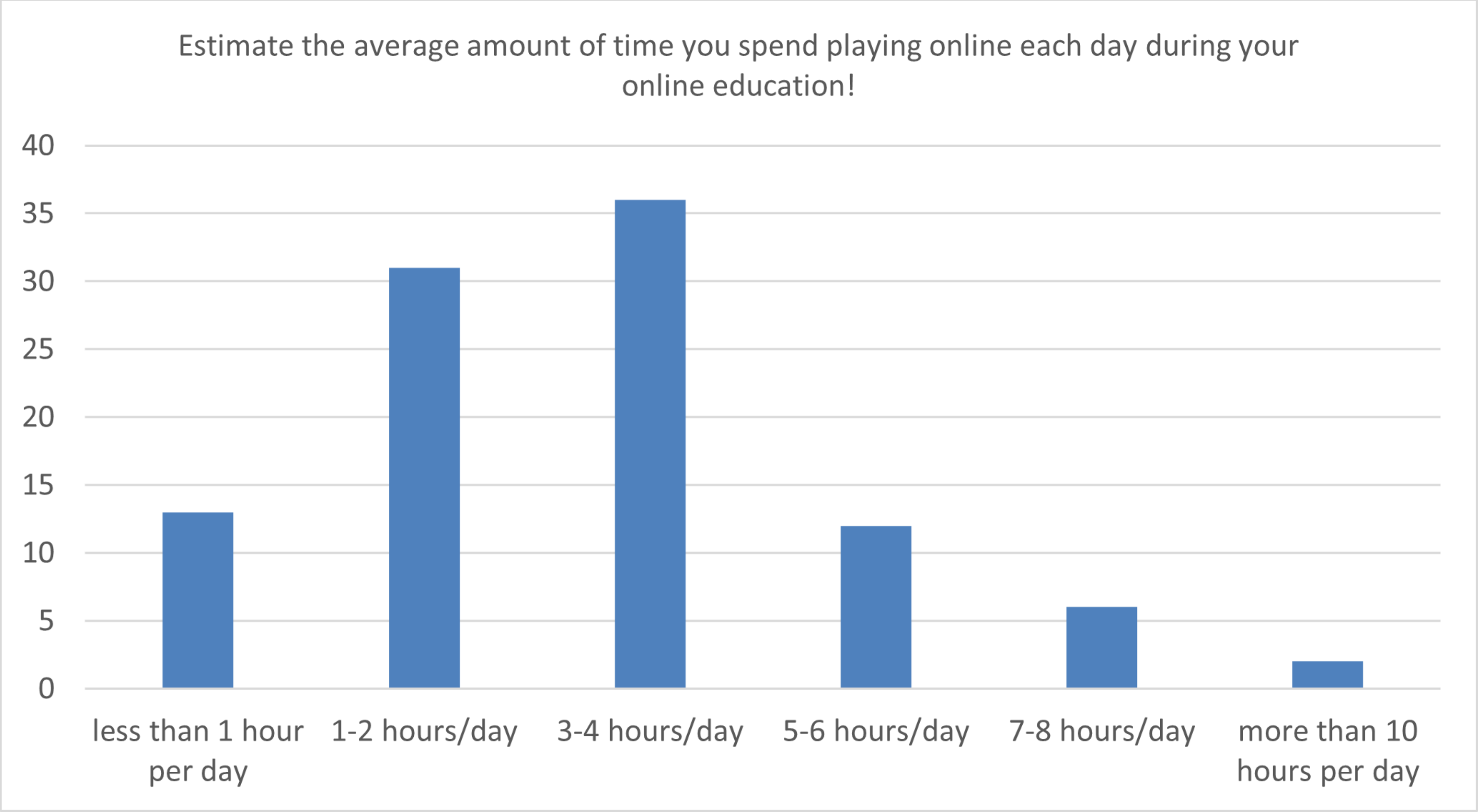
Medical Game

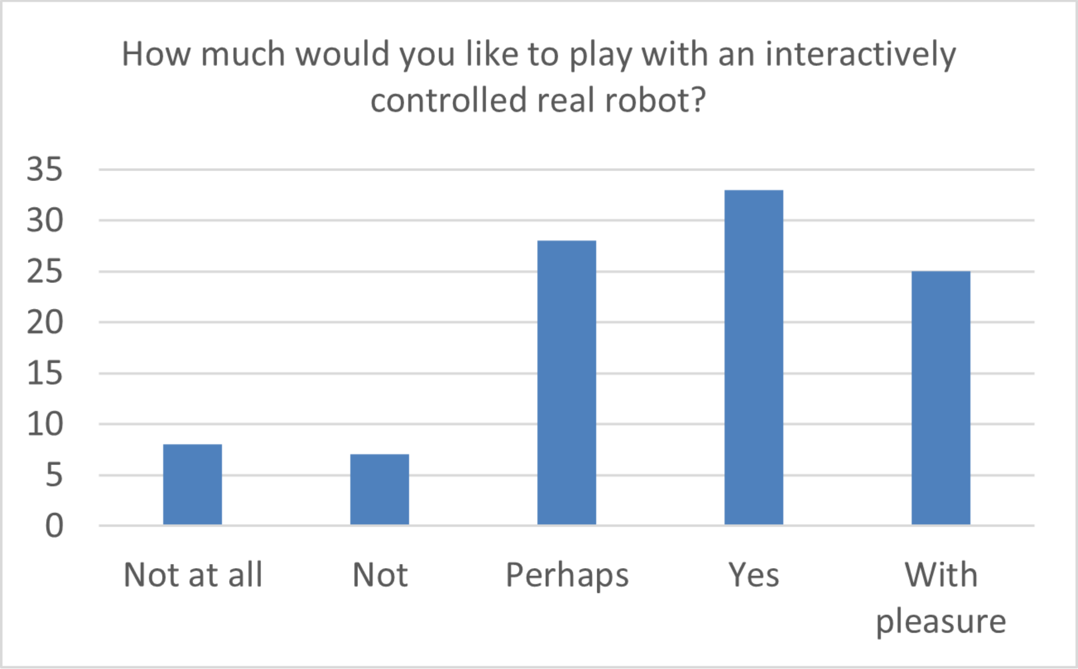


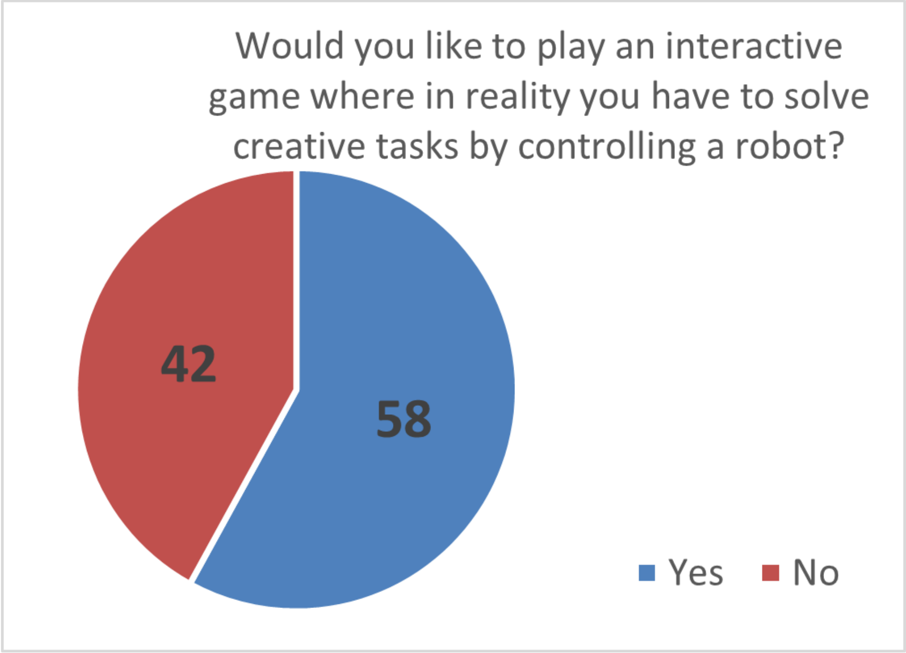
Bányai Júlia Secondary
Grammar School

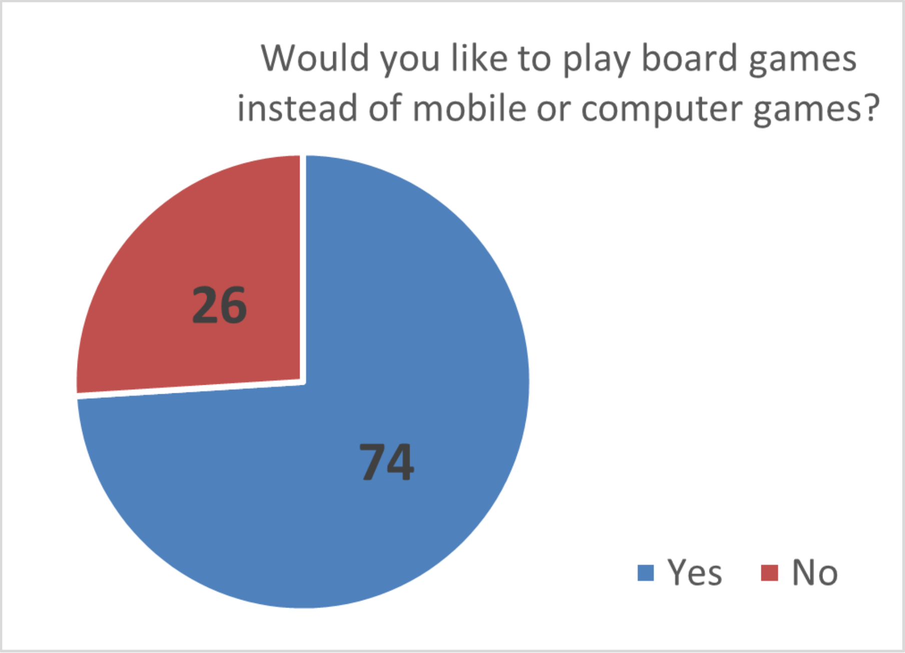
The results of our own questionnaire research on activities during online education.



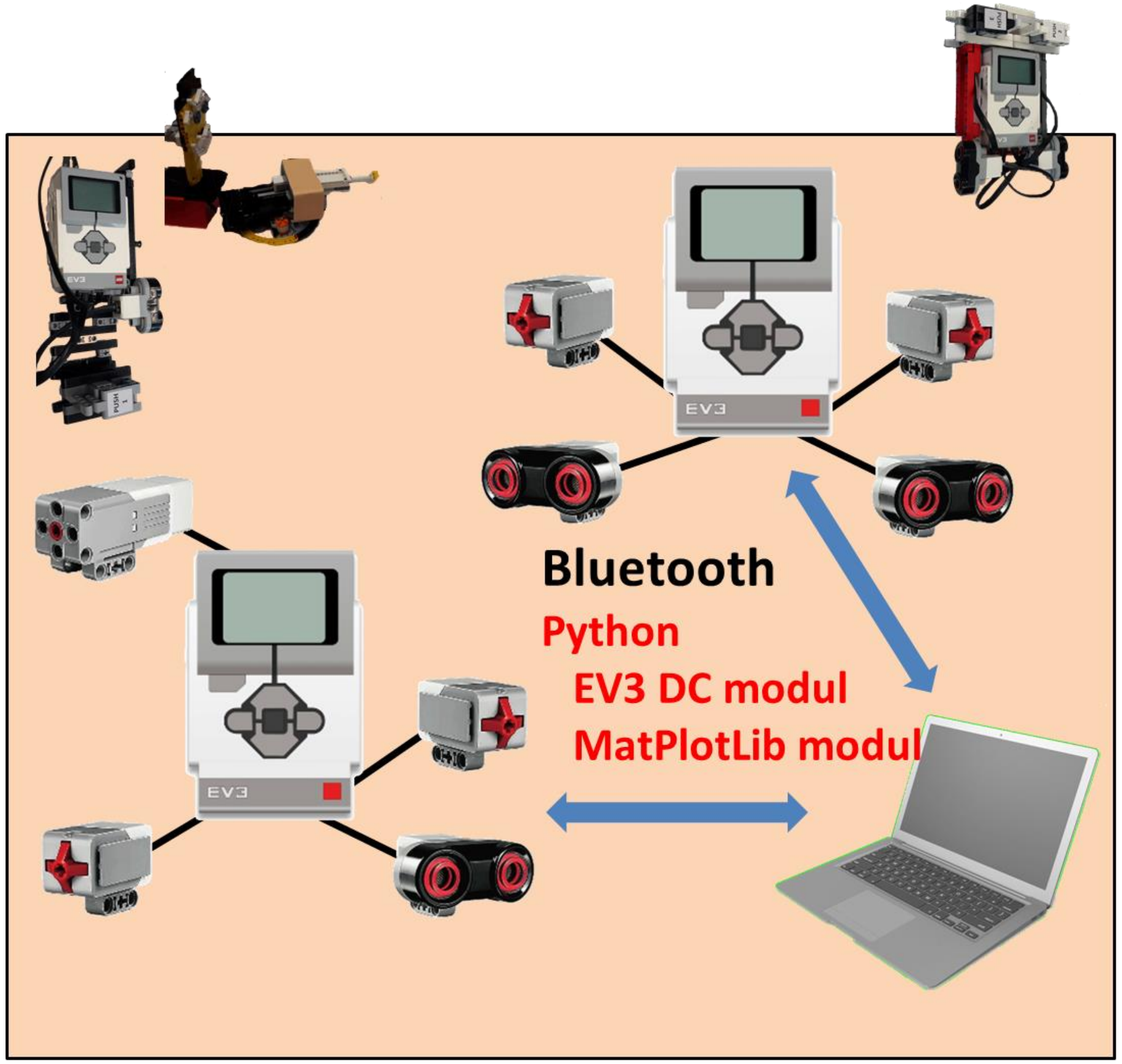








Network structure of the Project

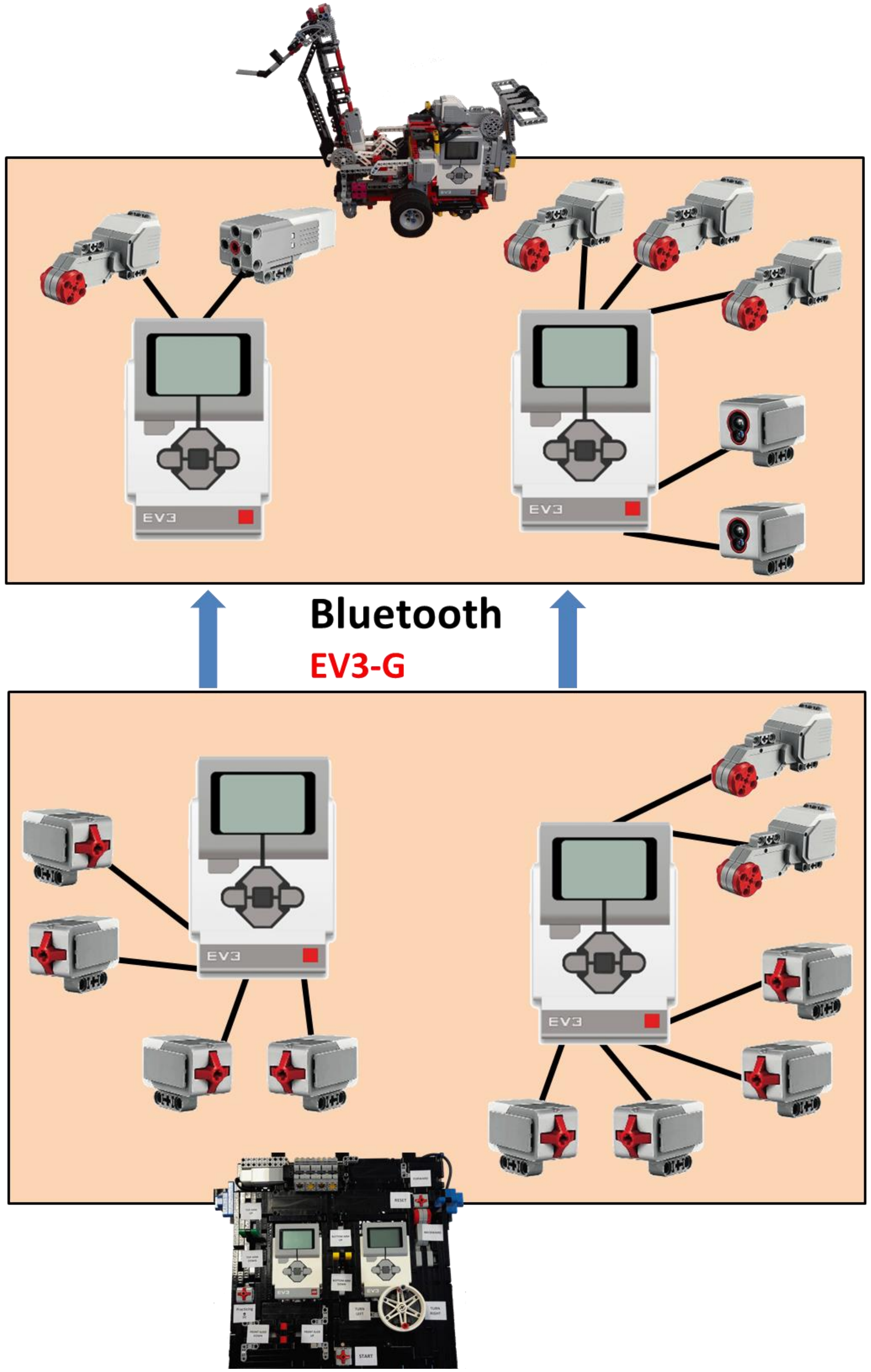


Bluetooth

Python

EV3 DC modul

MatPlotLib modul



Bluetooth

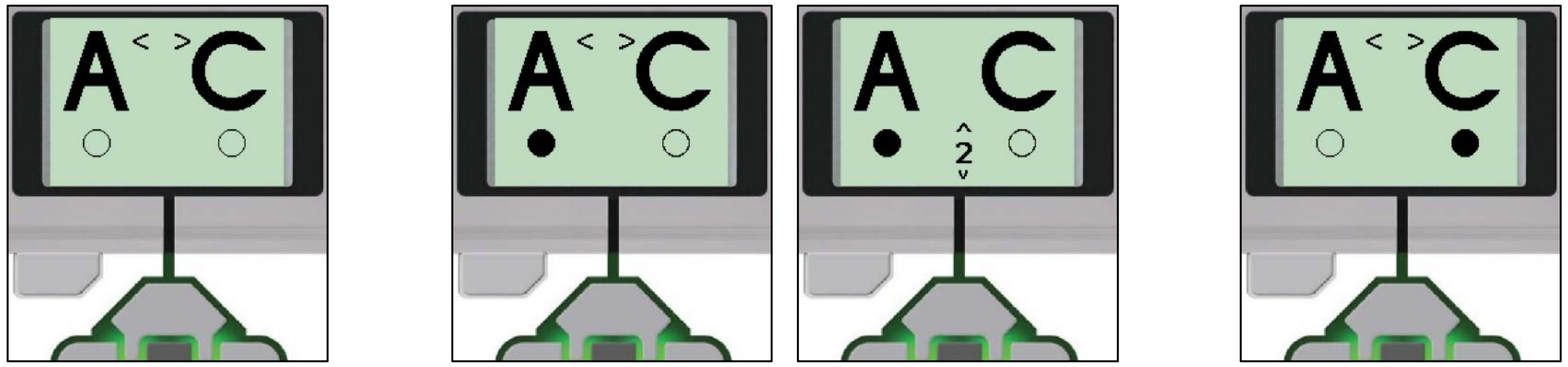
EV3-G

Electronic components

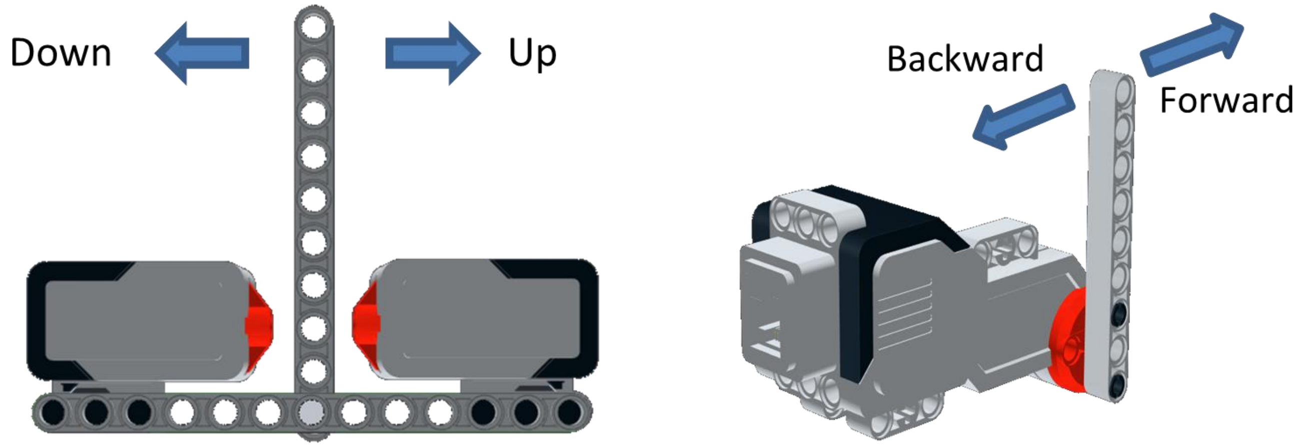
6 EV3 Bricks
6 Large Motors
2 Medium Motors
2 Color Sensors
3 Ultrasonic Sensors
12 Touch Sensor

Hardware Constructions

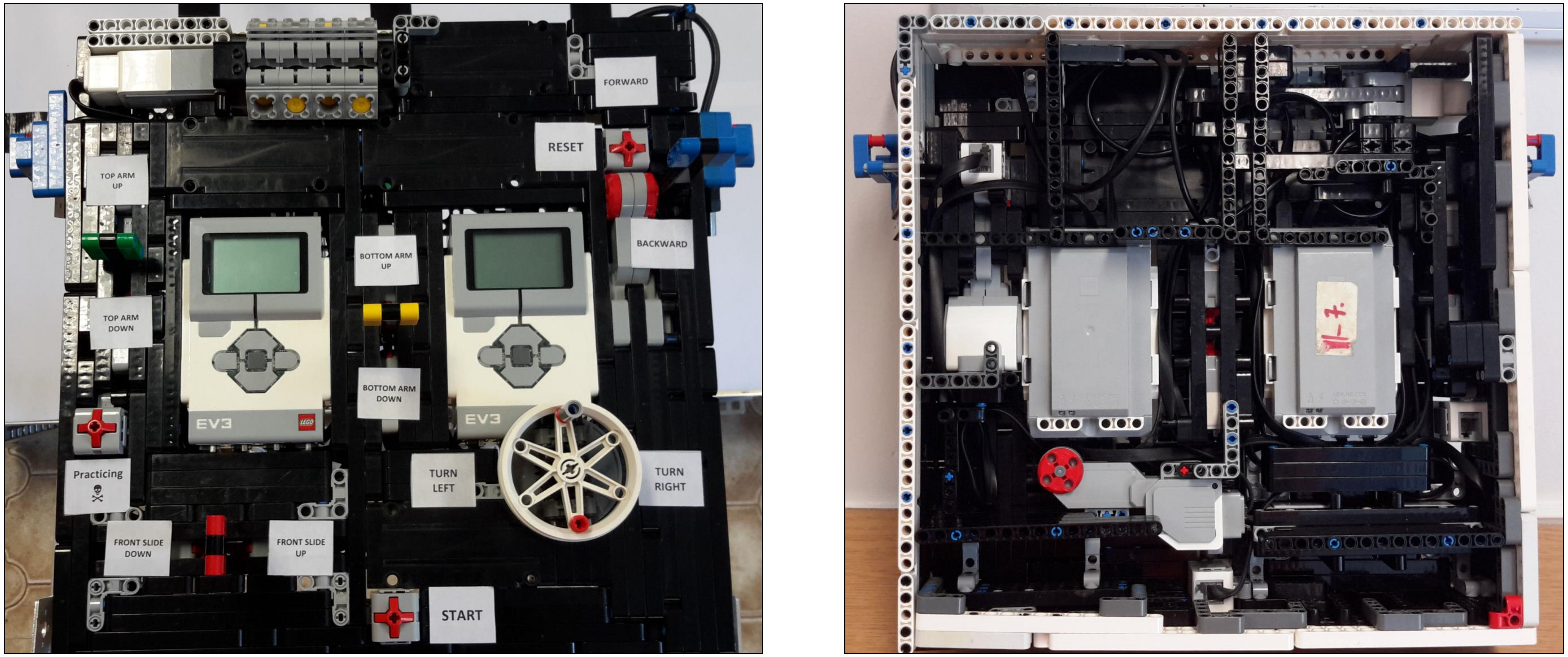
Screen menu



Lifting mechanism controller

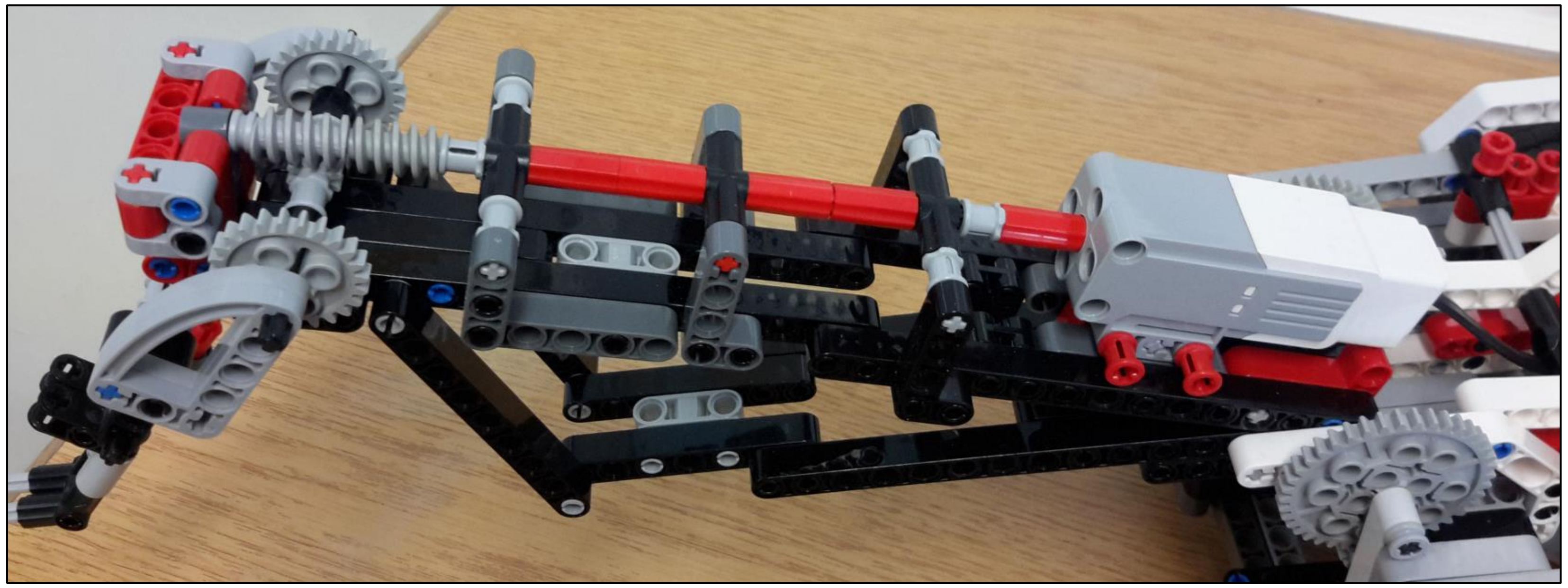


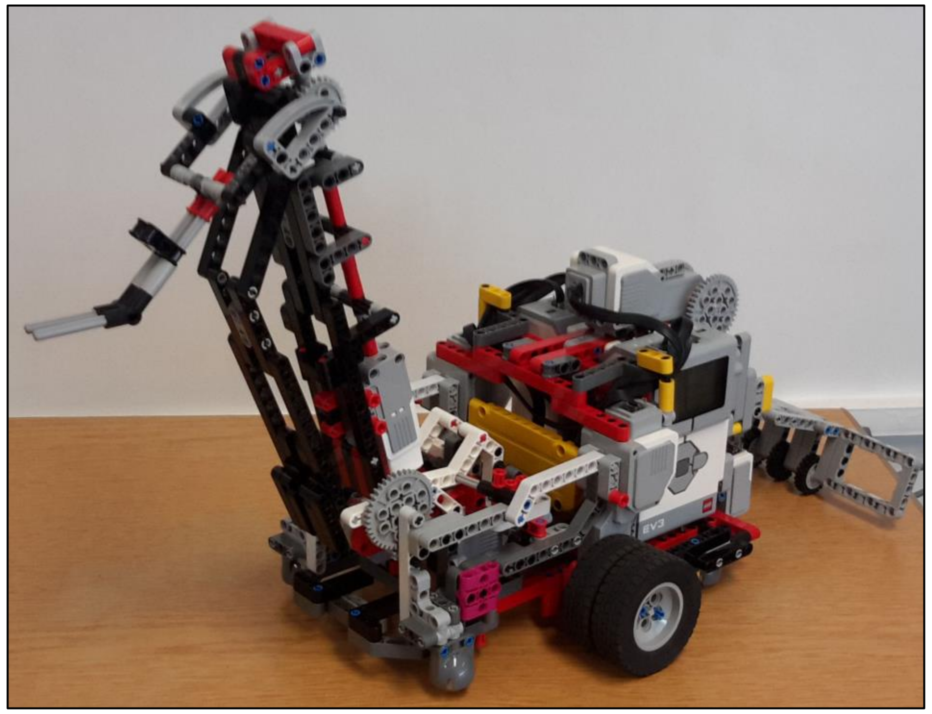
Remote controller

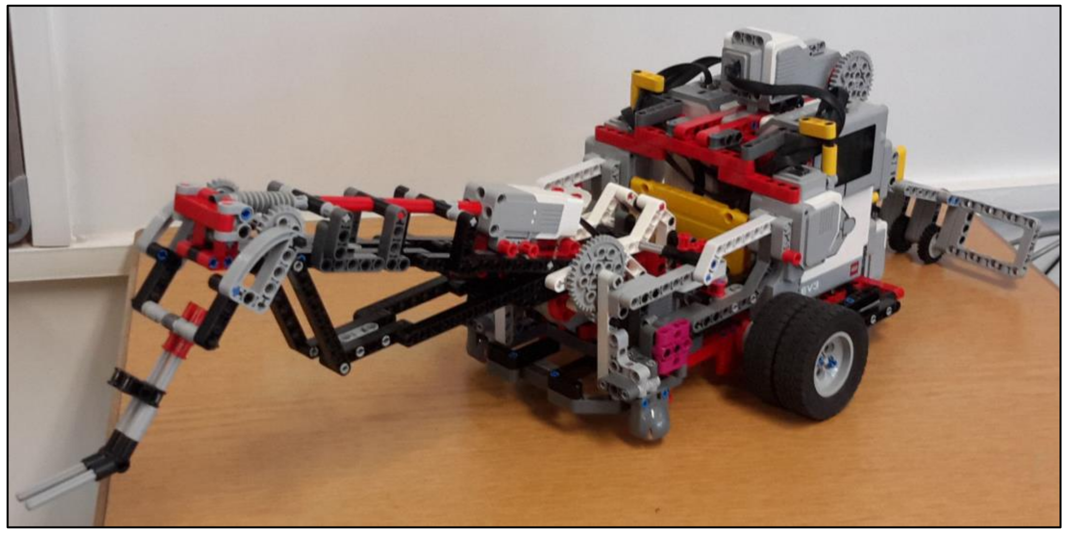


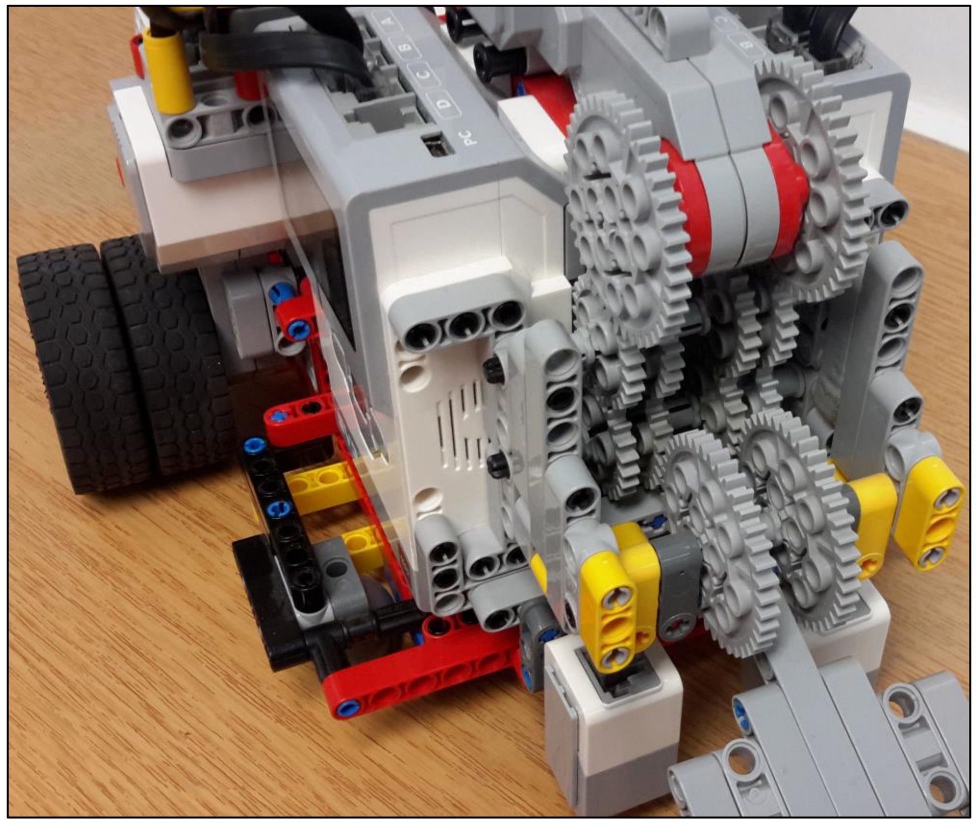
Hardware Constructions

Rover and crane

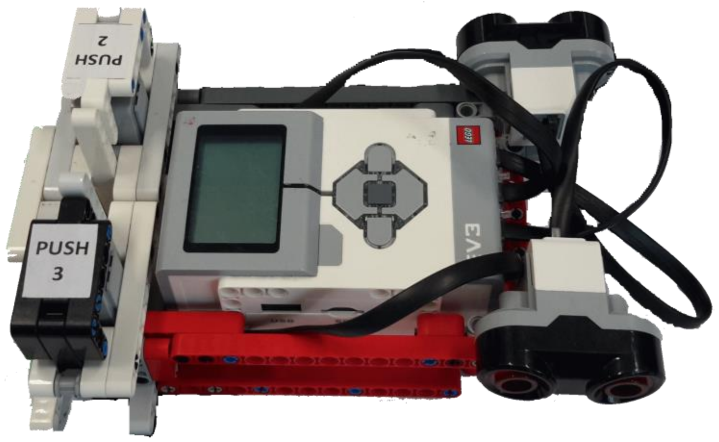


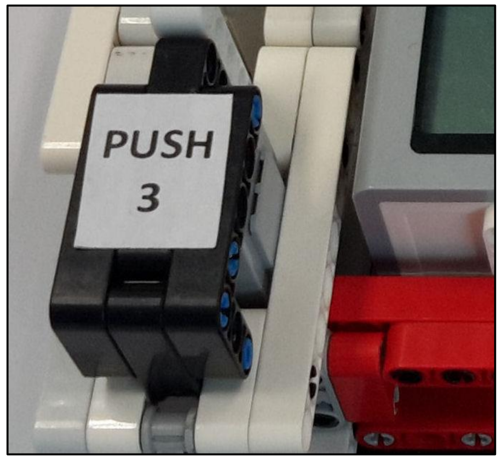


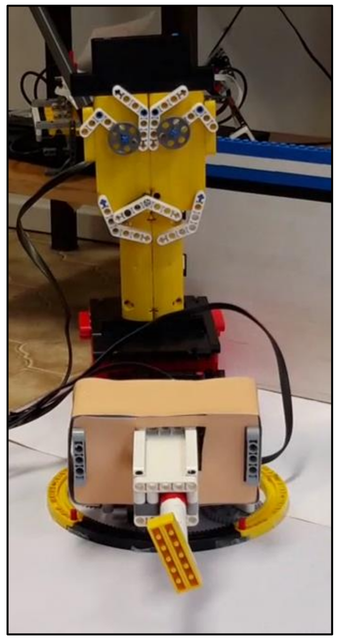


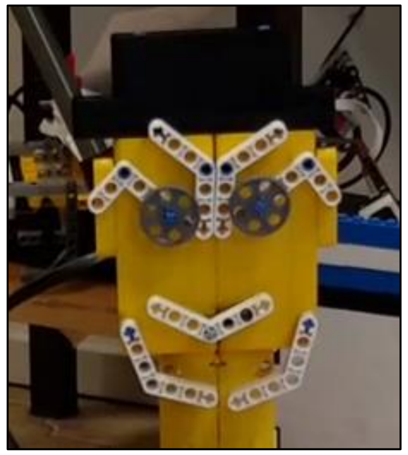


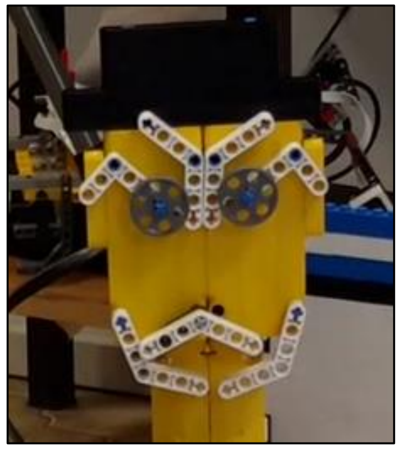
Other programmed constructions



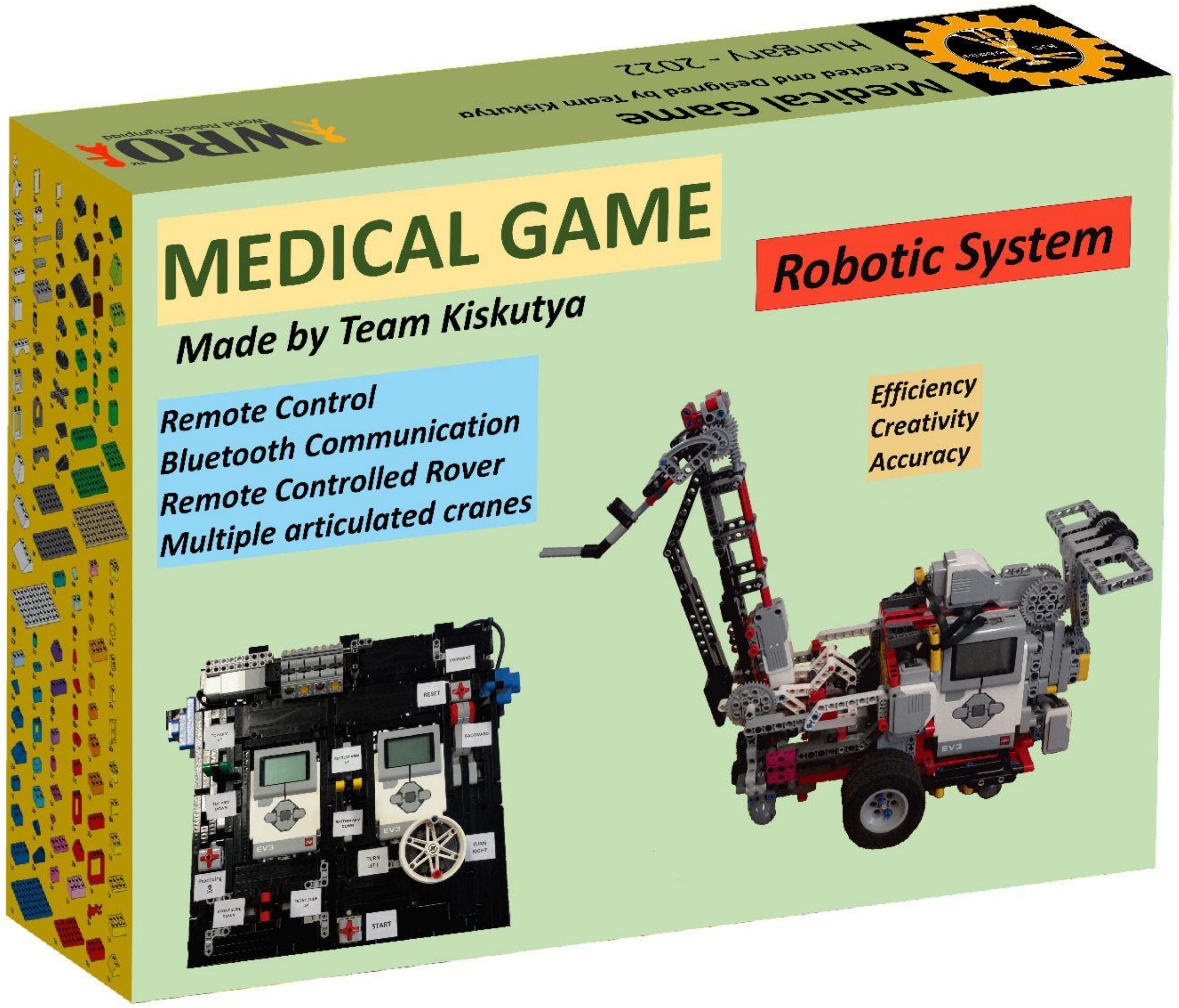








Medical Game



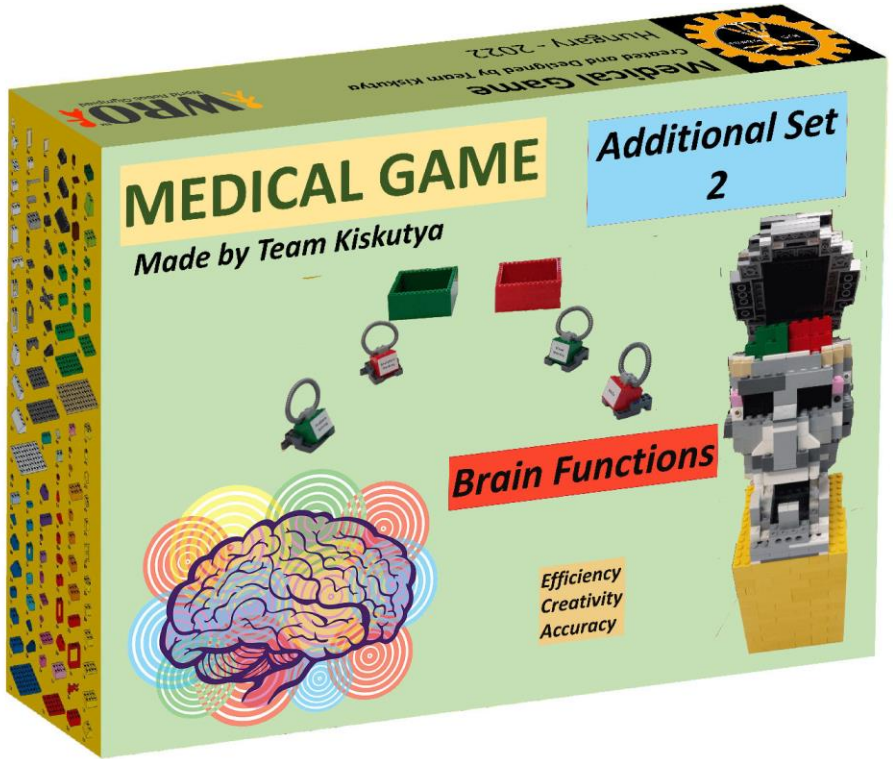
Base Set

- *Remote Control Rover*
- *Bluetooth Communication*

Advanced skills

- *Efficiency*
- *Creativity*
- *Accuracy*
- *Spatial vision*
- *…*

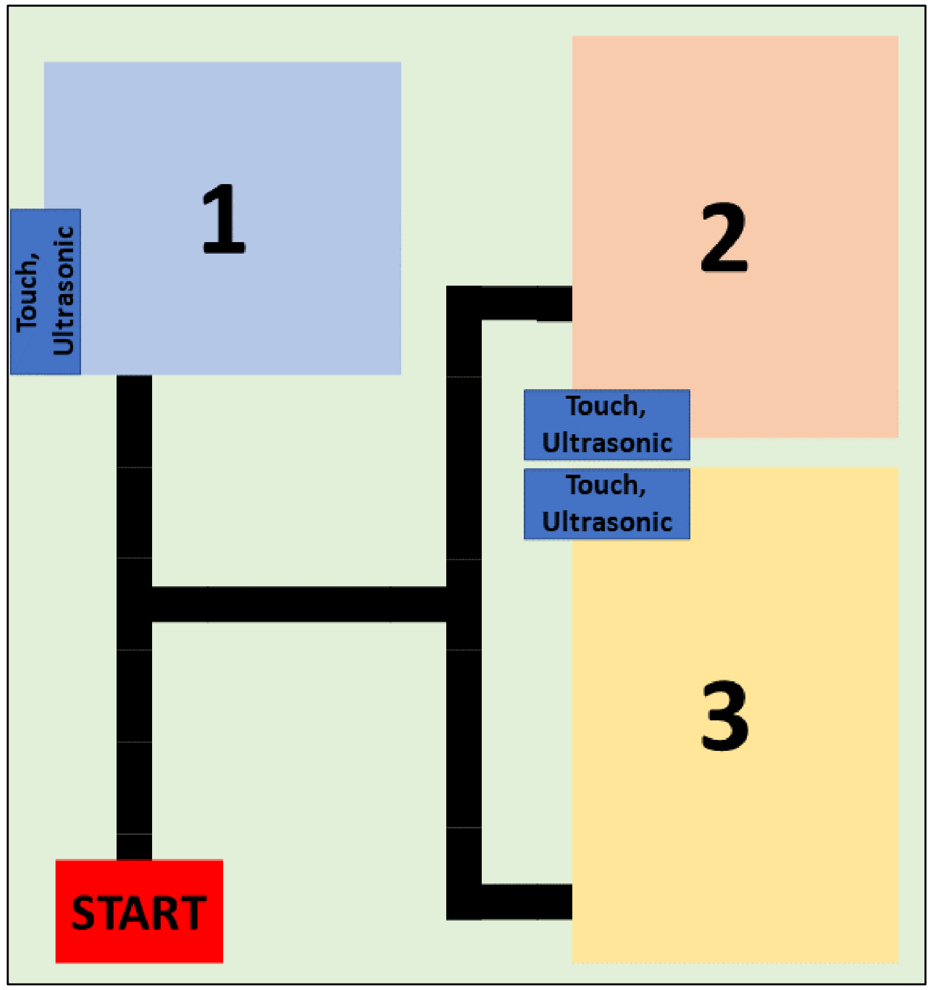
Additional Sets

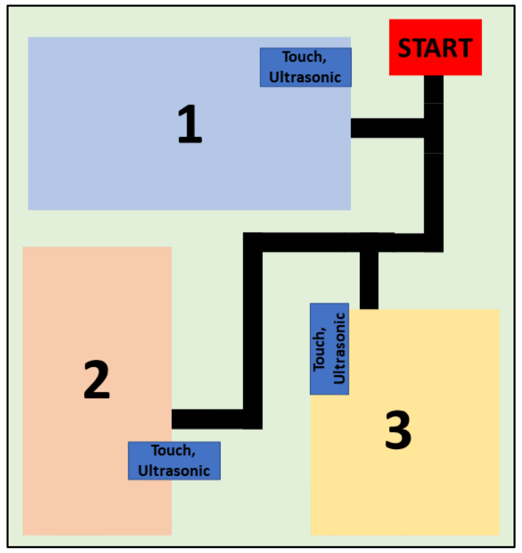


Medical Game

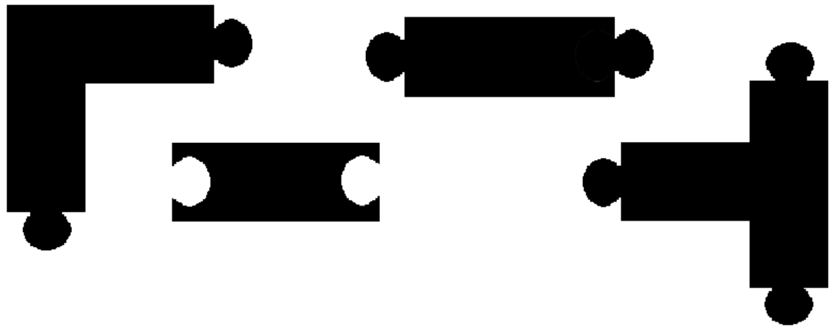
The path between the individual tasks is not fixed, but can be built freely.

For example

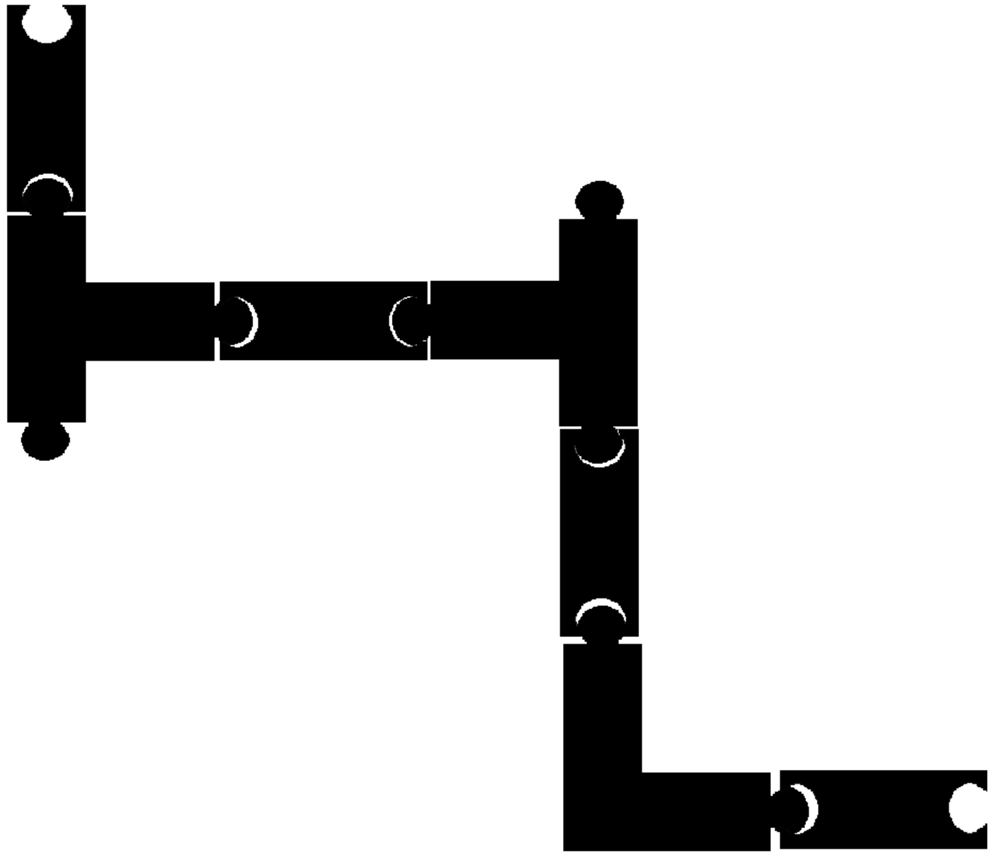




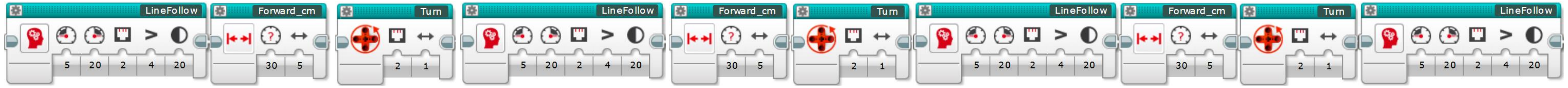
The route connected like a puzzle.



For example



The program code describing the route.



Business Model Canvas

Medical Game

Key Partners

- Developer (hardware)
- Web operator
- Courier service
- Distribution network (e.g. toy stores)
- Professionals (scientific background)
- Research + Development

Key Activities

- Contact
- Problem management
- Writing applications for start-ups
- Web activities (social media)
- Delivery management

Key Resources

- Engineers (electrical engineer, mechanical engineer, product designers)
- Computer scientists
- Web developers
- Brand developers (logo, corporate identity)
- Production

Value Propositions

3D, interactive, skill-based, robotic technology modular game, similar to a video game and software, web service operated by developers

Customer Relationships

- Continuous improvement, additional stocks
- Fan Club on the Internet
- Support site (FAQ)
- Interactive website with scientific background

Channels

Information:

- Television commercials
- Internet
- Social media advertising

Product:

- Online order + courier service
- Toy stores

Customer Segments

- Children (6 years and older) - their parents
- Playful adults
- Educational institutions
- Organizers of children's programs (e.g. camps)
- Children's hospitals
- Interactive playhouses

Cost Structure

- Development (periodic)
- Warehouse rental
- Courier service
- Web operation

Revenue Streams

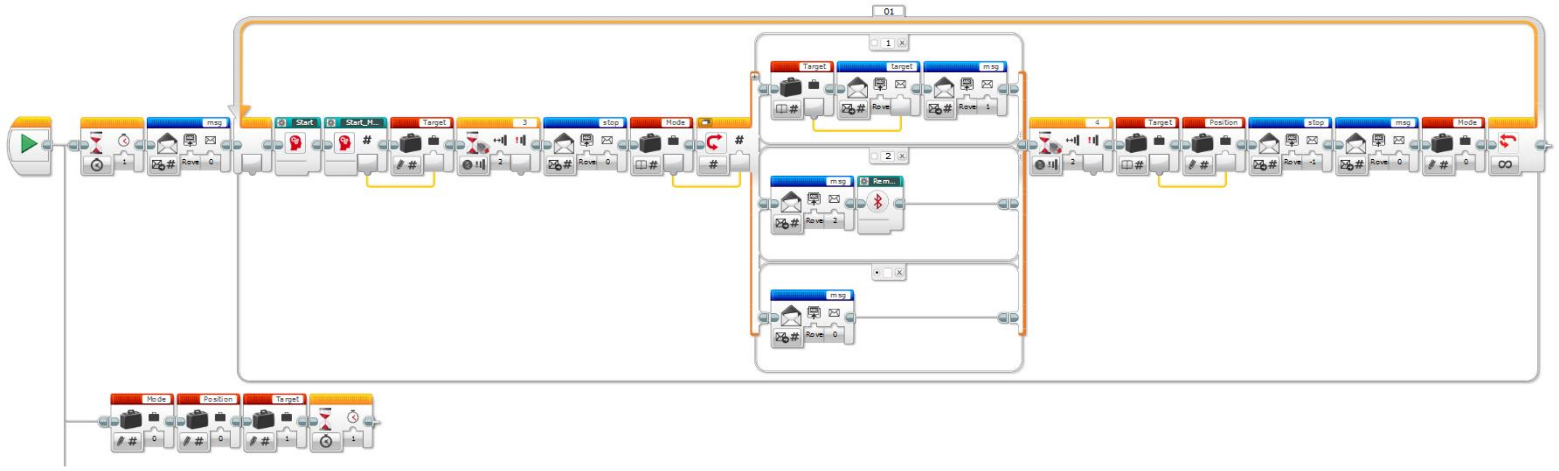
- Product sales (related services are free)

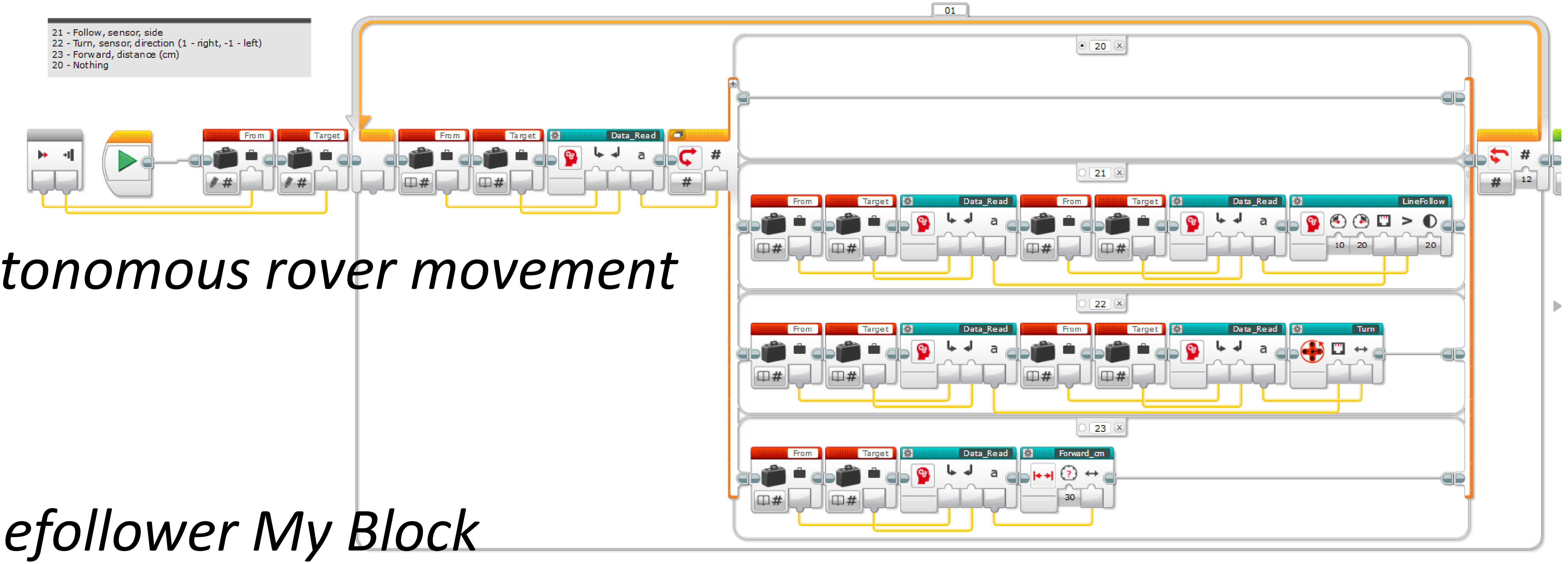
  A rate of about
  400-500 EUR/Base Kit
  200-300 EUR/Additional kits
- Royalties (for accessories produced by other companies)
- Brand sales

Program codes – EV3-G

EV3-G program code content:

248 blocks; 13 My Blocks

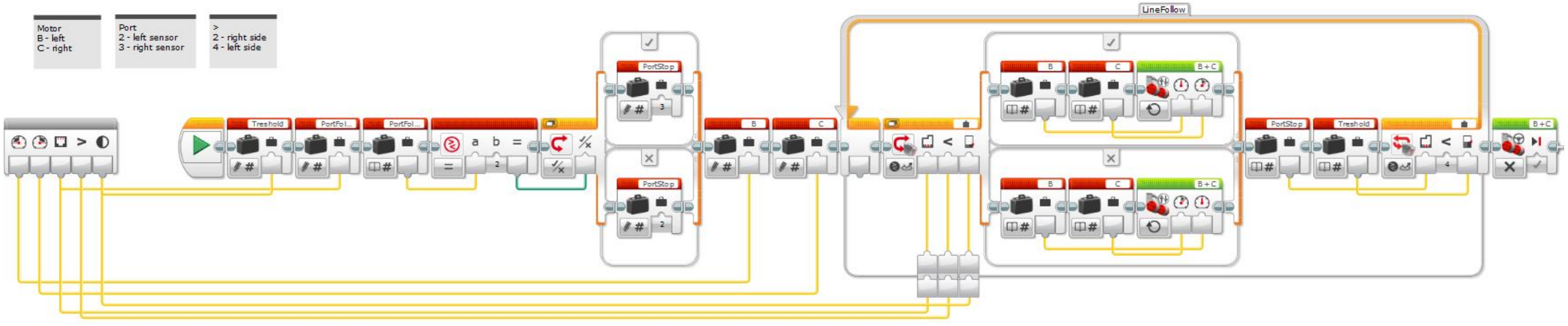




21 - Follow, sensor, side
22 - Turn, sensor, direction (1 - right, -1 - left)
23 - Forward, distance (cm)
20 - Nothing

Autonomous rover movement

Linefollower My Block



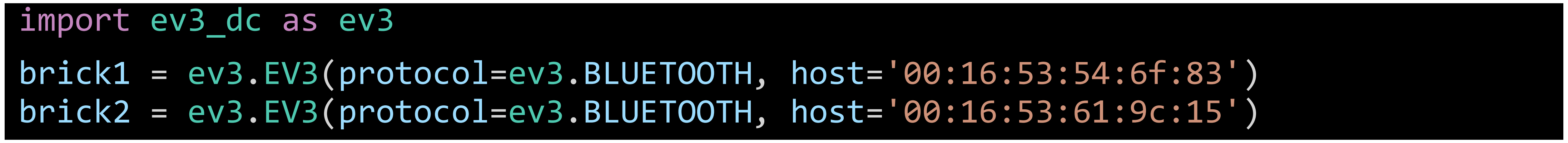
Motor
B - left
C - right

Port
2 - left sensor
3 - right sensor

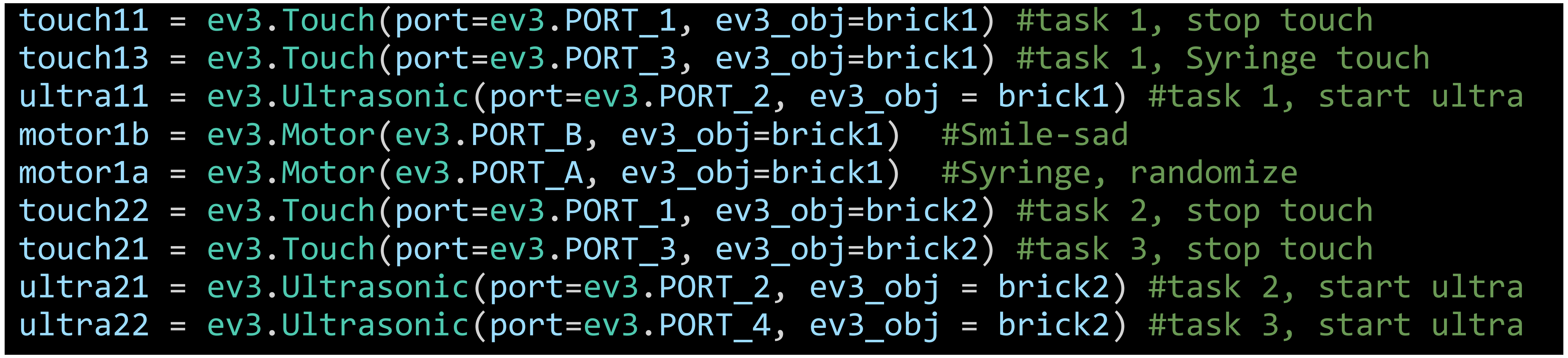
>
2 - right side
4 - left side

Program codes – Python

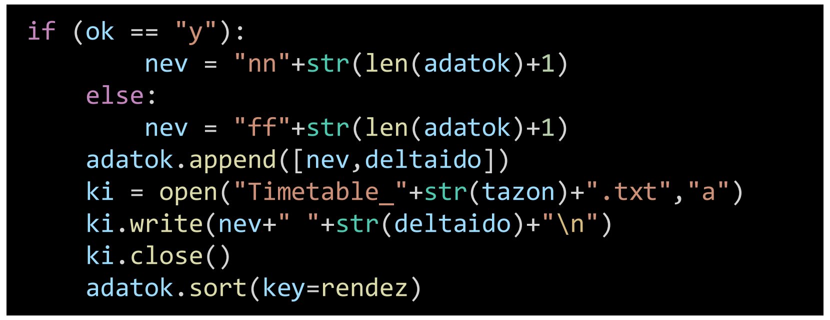
Measuring the completion time of tasks – EV3 DC modul



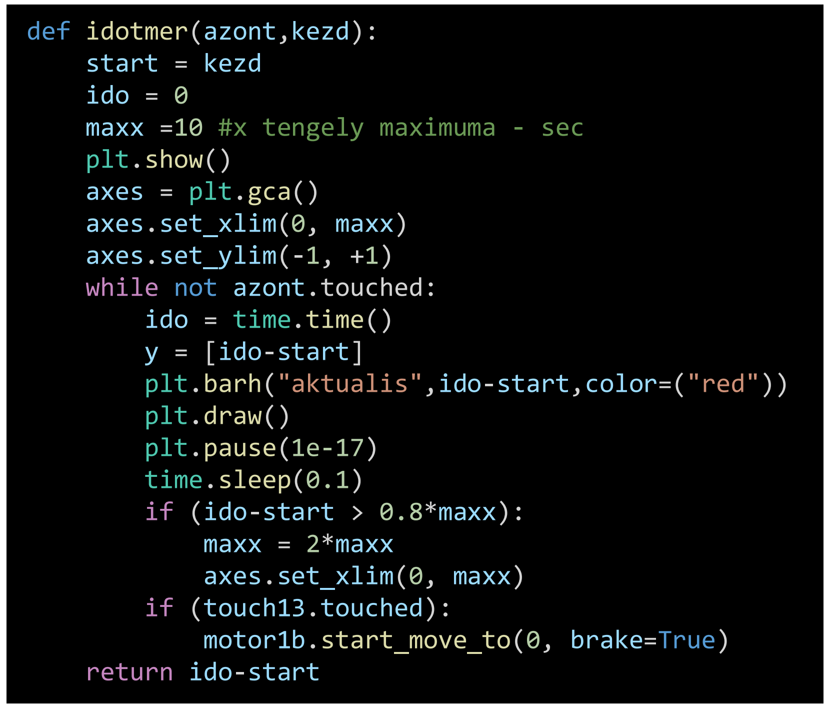
```python
import ev3_dc as ev3

brick1 = ev3.EV3(protocol=ev3.BLUETOOTH, host='00:16:53:54:6f:83')
brick2 = ev3.EV3(protocol=ev3.BLUETOOTH, host='00:16:53:61:9c:15')
```



```python
touch11 = ev3.Touch(port=ev3.PORT_1, ev3_obj=brick1) #task 1, stop touch
touch13 = ev3.Touch(port=ev3.PORT_3, ev3_obj=brick1) #task 1, Syringe touch
ultra11 = ev3.Ultrasonic(port=ev3.PORT_2, ev3_obj = brick1) #task 1, start ultra
motor1b = ev3.Motor(ev3.PORT_B, ev3_obj=brick1)  #Smile-sad
motor1a = ev3.Motor(ev3.PORT_A, ev3_obj=brick1)  #Syringe, randomize
touch22 = ev3.Touch(port=ev3.PORT_1, ev3_obj=brick2) #task 2, stop touch
touch21 = ev3.Touch(port=ev3.PORT_3, ev3_obj=brick2) #task 3, stop touch
ultra21 = ev3.Ultrasonic(port=ev3.PORT_2, ev3_obj = brick2) #task 2, start ultra
ultra22 = ev3.Ultrasonic(port=ev3.PORT_4, ev3_obj = brick2) #task 3, start ultra
```

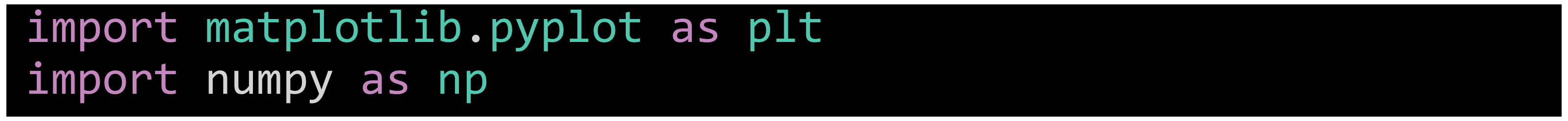


```python
if (ok == "y"):
        nev = "nn"+str(len(adatok)+1)
    else:
        nev = "ff"+str(len(adatok)+1)
    adatok.append([nev,deltaido])
    ki = open("Timetable_"+str(tazon)+".txt","a")
    ki.write(nev+" "+str(deltaido)+"\n")
    ki.close()
    adatok.sort(key=rendez)
```

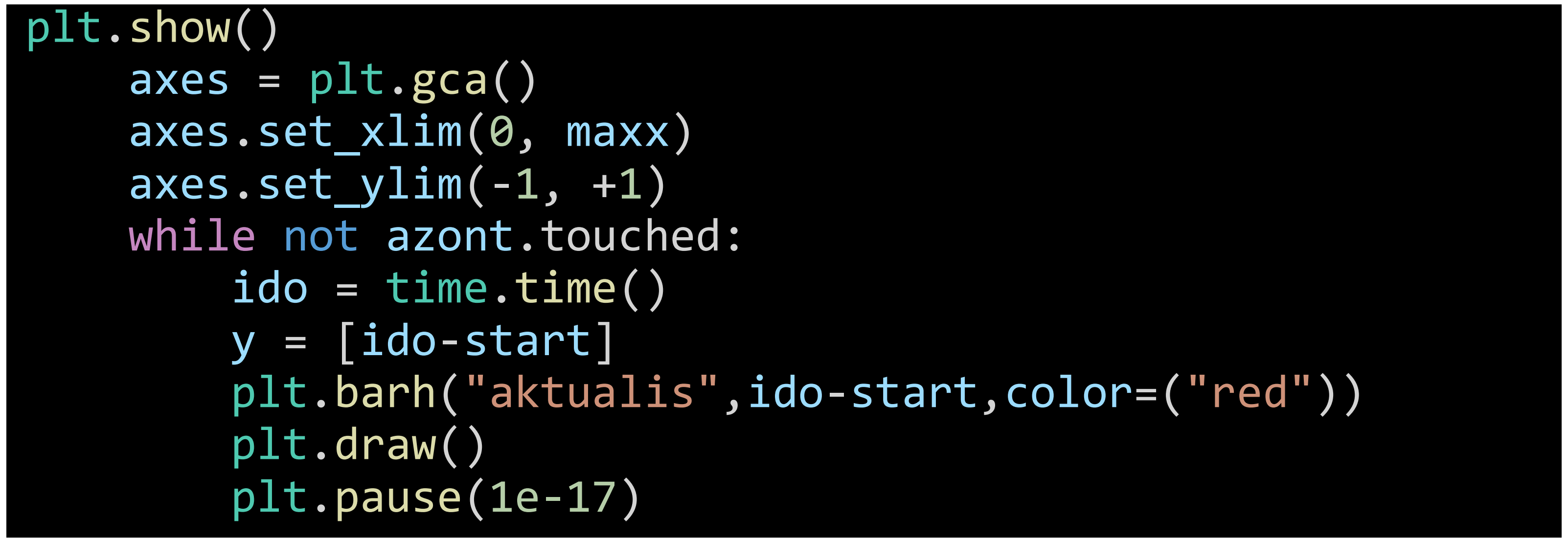


```python
def idotmer(azont,kezd):
    start = kezd
    ido = 0
    maxx =10 #x tengely maximuma - sec
    plt.show()
    axes = plt.gca()
    axes.set_xlim(0, maxx)
    axes.set_ylim(-1, +1)
    while not azont.touched:
        ido = time.time()
        y = [ido-start]
        plt.barh("aktualis",ido-start,color=("red"))
        plt.draw()
        plt.pause(1e-17)
        time.sleep(0.1)
        if (ido-start > 0.8*maxx):
            maxx = 2*maxx
            axes.set_xlim(0, maxx)
        if (touch13.touched):
            motor1b.start_move_to(0, brake=True)
    return ido-start
```

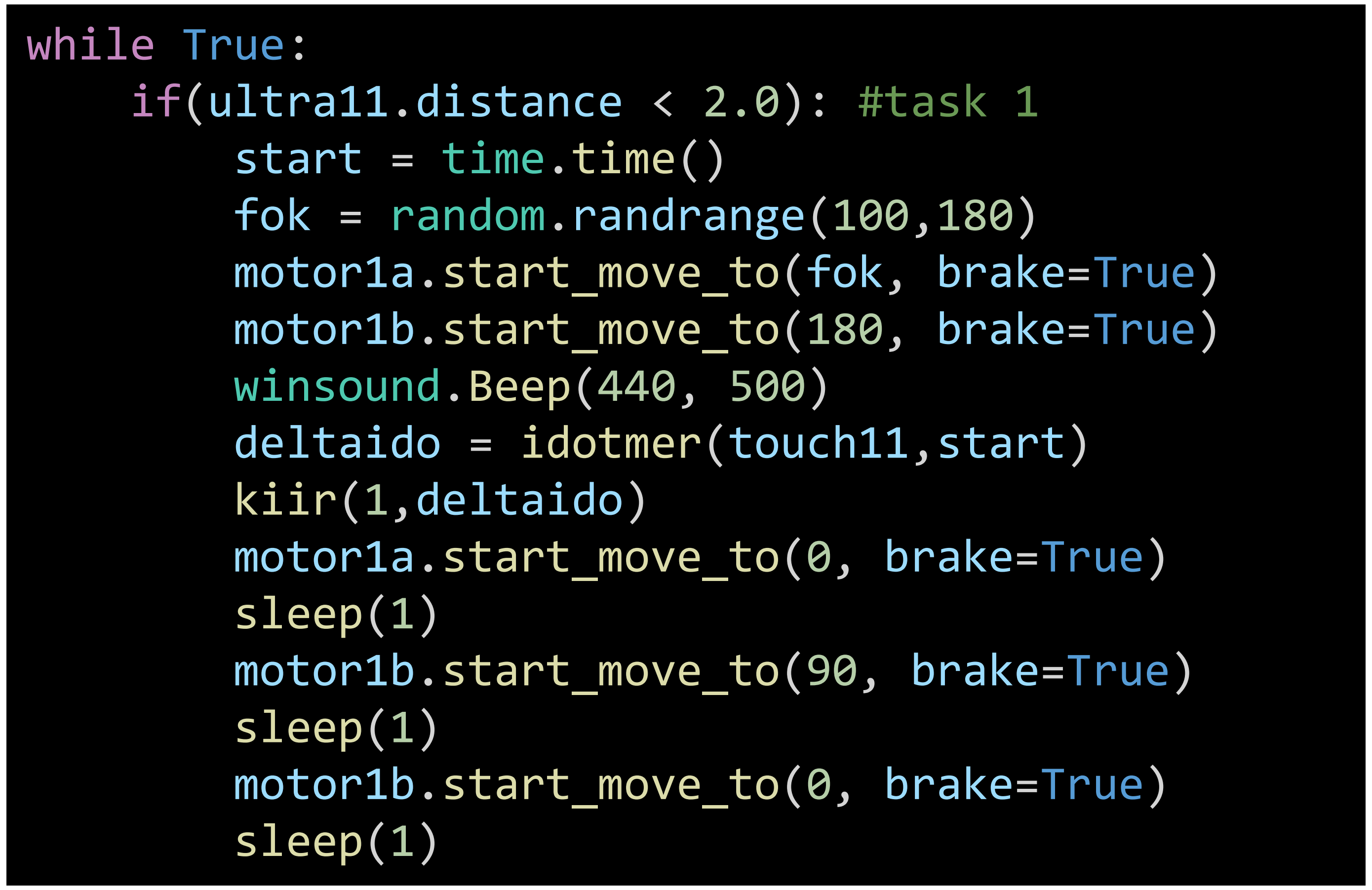
The graphic display – matplotlib modul



```python
import matplotlib.pyplot as plt
import numpy as np
```



```python
plt.show()
    axes = plt.gca()
    axes.set_xlim(0, maxx)
    axes.set_ylim(-1, +1)
    while not azont.touched:
        ido = time.time()
        y = [ido-start]
        plt.barh("aktualis",ido-start,color=("red"))
        plt.draw()
        plt.pause(1e-17)
```



```python
while True:
    if(ultra11.distance < 2.0): #task 1
        start = time.time()
        fok = random.randrange(100,180)
        motor1a.start_move_to(fok, brake=True)
        motor1b.start_move_to(180, brake=True)
        winsound.Beep(440, 500)
        deltaido = idotmer(touch11,start)
        kiir(1,deltaido)
        motor1a.start_move_to(0, brake=True)
        sleep(1)
        motor1b.start_move_to(90, brake=True)
        sleep(1)
        motor1b.start_move_to(0, brake=True)
        sleep(1)
```