EV3 ROBOT – DataLog - Dokumentáció

Projekt leírás:

* Egyenesen halad a robot előre és rögzíti a visszavert fény értékét egy fájlba. Majd elkészít egy kimutatást hogy hány és milyen vastag fekete vonallal találkozott út közben és eltárolja egy Excel táblázatban.

Hivatalos dokumentáció elérése:

* <https://pybricks.com/ev3-micropython/tools.html>

**class DataLog(\*headers, name='log', timestamp=True, extension='csv', append=False)**

*Létrehoz egy fájlt és naplózza az adatokat.*

* + ***headers (col1, col2, …)*** - Oszlopfejlécek. Ezek az adatoszlopok nevei. Például válassza az „idő” és a „szög” lehetőséget.
  + ***name (str)*** – A fájl neve.
  + ***timestamp (bool)*** – Válassza a True beállítást a dátum és az idő hozzáadásához a fájlnévhez. Így a fájl egyedi nevet kap. Az időbélyegző elhagyásához válassza a False lehetőséget.
  + ***extension (str)*** – Fájlkiterjesztés.
  + ***append (bool)*** – Válassza a True beállítást egy meglévő adatnaplófájl újbóli megnyitásához és adatok hozzáfűzéséhez. A meglévő adatok törléséhez válassza a False lehetőséget. Ha a fájl még nem létezik, akkor mindkét esetben üres fájl jön létre.

**log(\*values)**

*Egy vagy több értéket ment a fájl új sorába.*

* + ***values (object, object, …) –*** Egy vagy több objektum vagy érték.

A PROJEKT MODULJAI

1. Main modul – robot parancsok futtatása
2. Feladatok modul – robot parancsok függvénybe írása

Közösen használt metódusok

**FELADAT(self):**

Üzenetet naplóz az adatnaplófájlba, és kinyomtatja az adatnapló objektumot

**VONALAK(self):**

A vonalhoz kapcsolódó feladatok kezelésére szolgál. Jelenleg a vonalHossza() metódust hívja

**VONALHOSSZA(self):**

Elindítja a robotot, naplózza a sorok számát az adatnaplófájlban és megméri a fekete vonalak szélességét

Kódrészlet - main.py

#!/usr/bin/env pybricks-micropython

from pybricks.hubs import EV3Brick

from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor, InfraredSensor, UltrasonicSensor, GyroSensor)

from pybricks.parameters import Port, Stop, Direction, Button, Color

from pybricks.tools import wait, StopWatch, DataLog

from pybricks.robotics import DriveBase

from pybricks.media.ev3dev import SoundFile, ImageFile

import Feladatok

oraimunka = Feladatok.Feladatok()

#oraimunka.feladat()

oraimunka.vonalak()

from py bricks.hubs import EV3Brick

from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor, InfraredSensor, UltrasonicSensor, GyroSensor)

from pybricks.parameters import Port, Stop, Direction, Button, Color

from pybricks.tools import wait, StopWatch, DataLog

from pybricks.robotics import DriveBase

from pybricks.media.ev3dev import SoundFile, ImageFile

import Feladatok

oraimunka = Feladatok.Feladatok()

oraimunka.vonalak()

Kódrészlet - Feladatok.py

#!/usr/bin/env pybricks-micropython

from pybricks.hubs import EV3Brick

from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor, InfraredSensor, UltrasonicSensor, GyroSensor)

from pybricks.parameters import Port, Stop, Direction, Button, Color

from pybricks.tools import wait, StopWatch, DataLog

from pybricks.robotics import DriveBase

from pybricks.media.ev3dev import SoundFile, ImageFile

class Feladatok():

def \_\_init\_\_(self):

# tégla

self.ev3 = EV3Brick()

# motorok

self.jm = Motor(Port.B)

self.bm = Motor(Port.C)

self.km = Motor(Port.A)

# szenzorok

self.cs = ColorSensor(Port.S3)

self.ts = TouchSensor(Port.S1)

self.gs = GyroSensor(Port.S2)

self.us = UltrasonicSensor(Port.S4)

#self.ir = InfraredSensor(Port.S4)

# dupla motorkezelő

self.robot = DriveBase(self.jm, self.bm, 55, 115)

self.ido = StopWatch()

self.data = DataLog(name='adatok', timestamp=False, extension='csv')

#self.data = DataLog('time','angle',name='dataLog', timestamp=False, extension='txt',append=True)

def feladat(self):

self.data.log('Hello','Világ')

print(self.data)

def vonalak(self):

self.vonalHossza()

def vonalHossza(self):

hosszok = []

db = 5

self.robot.drive(100,0)

self.data.log("Hány db vonal:")

self.data.log(db)

for vonalakSzama in range(db):

vege = False

fekete = False

self.robot.drive(100,0)

while not vege:

if self.cs.reflection() < (74+10)/2-20 and not fekete:

fekete = True

self.ido.reset()

if fekete and self.cs.reflection() > (74+10)/2-10:

vege = True

hossz = self.ido.time()

hosszok.append(hossz)

self.robot.stop(Stop.BRAKE)

print(hossz)

print(hosszok)

self.data.log('Hosszak:')

self.data.log(hosszok)

return hosszok

#self.data.log(hosszok)

# print(self.data)

*Word dokumentumot készítette: Nádas Viki, Tar Gergő, Kádár Kristóf*