**3. óra: Villanyrendőr sötétben**

1. 5 perces beszámoló tetszőleges tartalommal és üzenettel valamint tetszőlegesen választott retorikai eszközökkel. Nekünk kell eldönteni, hogy mi lehetett az üzenet. Figyelni kell a beszéd felépítésére (bevezető, fő rész és konklúzió) és arra, hogy a tartalom minden eleme támogassa az üzenetet, illetve, hogy használjunk retorikai eszközöket.
2. 3 perces véleményalkotás  
   - bevezető, fő rész, konklúzió  
   - 2 pozitív és egy konstruktív javaslat  
   - konklúzióban összefoglalni mi lehetett az üzenet, volt-e retorikai eszköz használva és adni még egy pozitív megjegyzést.
3. Villanyrendőr készítés:
   1. *Bevezető*:  
      A város vezetése nagyon meg volt elégedve a múltkori munkáddal, ami a város útkereszteződésénél felálított villanylámpa volt. A város polgármestere egyszer abban a kellemetlen helyzetben találta magát, hogy éjszaka autókázott az útkereszteződésnél, amikor is egy cseppnyi forgalom sem volt, neki mégis ki kellett várni, amíg a piros zöldre vált. A lakosság többi része is tapasztalta már ezt a bosszúságot.

Felkerestek téged és arra kértek, hogy progrmozd át úgy, hogy naplemente után, mikor már sötét van, a körösztöződésben csak a sárga lámpa villogjon. Így a fő és mellékút táblák veszik át az irányítást és megszűnik a sok várakozás.

* 1. *Előkészületek*:  
     - Elkészíteni az áramkört.  
     - Beimportálni a villanyrendőrt működtető függvényt.  
     - Létrehozni a kapcsolatot a webkamerával az opencv csomag használatával.  
     - Inicializálni a LED-ket.  
     - Végtelen ciklusban rögzíteni egy képet és kiszámolni a pixel értékek átlagát.  
     - Az átlag értéke alapján vezérelni a villanyrendőr működését.  
     - A q billentyű lenyomásával jelezni, hogy befejeztük.  
     - Bezárni a kapcsolatot a kamerával.

* 1. *Elektronika:*  
     - Legyen a webcam előkészítve.  
     - Szereljük a 3 LEDet a breadboardra a hozzájuk tartozó ellenállásokkal és kössük őket össze a Raspi GPIO-ival.
  2. *Kód:*  
     - teszteljük le a kamera működését avval, hogy kiolvasunk képet belőle és ábrázoljuk.  
     - Ellenőrízzük le, hogy mindegyik LED vezérelhető-e a Raspberryről.  
     - A raspberry\_functions.py kódban egészítsük ki a *traffic\_light\_sequence* függvényt, hogy az egy villanyrendőr működésének feleljen meg.  
       
     - Mutassuk be a *numpy.mean* függvény működését, hogy a kamerából kiolvasott kép pixeljeinek átlagát ki tudjuk számolni.  
     - Egy *if* szerkezetben vizsgáljuk meg, hogy a kamera pixeljeinek normalizált átlaga kisebb-e egy értéknél (besötétedett-e) és ha igen, villogtassuk a sárga LEDet. Ha nem, működjön a villanyrendőr normálisan.
  3. *Tesztelés:*  
     - teszteljük a kamerák működését.
  4. *Fejlesztés:*  
     - Használjunk a pixelek átlagolása helyett más módszert (pl. pixel értékek összeadása).

- Mi történik ha a kamera képét nem alakítjuk szürke skálára? Keressünk olyan műveleteket, amik arra az esetre is a kívánt működést biztosítják.

Kód minta:

from gpiozero import LED

import cv2, time

import numpy as np

from raspberry\_functions import traffic\_light\_sequence

red    = LED(17)

yellow = LED(27)

green  = LED(22)

cap    = cv2.VideoCapture(0)

while True:

    ret, frame = cap.read()

    frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

    light = np.mean(frame) / 256

    print(light)

    if light < 0.3:

        red.off()

        green.off()

        yellow.on()

        time.sleep(0.5)

        yellow.off()

    else:

        traffic\_light\_sequence(red, yellow, green)

    time.sleep(0.2)

    if cv2.waitKey(1) & 0xFF == ord('q'):

        break

cap.release()

# Bezarunk minden ablakot, amit a program megnyitott

cv2.destroyAllWindows()