**7. óra: Szellőztető rendszer**

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. DHT11 mérése raspberryvel:
   1. *Bevezető*:  
      Az ember hőmérséklet érzetét nem csak maga a hőmérséklet befolyásolja, hanem többek között a páratartalom is. A páratartalom emellett egy zárt térben még a penészedést is nagyban befolyásolja. Ezért ahol nincs megoldva rendesen a szellőztetés, oda érdemes szellőztető rendszert kialakítani.

A lakhelyeteken a sok mosás, benti ruhaszárítás és hosszas melegvízes fürdő miatt rengeteg pára keletkezik, ami hosszú távon ahhoz vezett, hogy a penész felütötte a fejét. Mivel már kicsit jártas vagy a barkácsolásban, kigondoltad, hogy összeraksz egy egyszerű szellőzőrendszert. Rászerelsz egy motorra egy ventillátor lapátot, összekötöd egy hő és páramérővel, majd beállítod, hogy ha 50% fölé szökik a páratartalom, akkor a ventillátorod kapcsoljon be és szívja el a párát mindaddig, amíg az 50% alá nem csökken.

* 1. *Előkészületek*:  
     Készítsünk ki próbapaneleket, jumper kábeleket, 10 kOhm ellenállásokat a thermisztornak, DHT11 thermisztorokat, DC motorokat és az LD293 vezérlőt. Emellé szükség lesz mérőműszerre is.   
     - Rakjuk össze az áramkört.   
     - Teszteljük le pythonból a motort és a DHT11 műküdését, hogy jó e az áramkör.  
       
     *Részletekre bontás:*Elkészíteni az áramkört.
     + Importálni az adafruit DHT11 objektumát, és a Motor objektumot, illetve a temperature\_functions modulból a megfelelő függvényeket.
     + Inicializálni a motort és a DHT11-t.
     + Definiálni egy függvényt, ami kiolvassa a hőmérsékletet.
     + Definiálunk egy függvényt ami kiolvassa a páratartalmat.
     + Definiálunk egy függvényt, ami a megadott feltételek alapján vezérli a ventillátort.
     + Egy végtelen while ciklusban mérjük és kiírjuk a hőmérsékletet és páratartalmat a képernyőre valamint meghívjuk a ventillátor vezérlőt.
  2. *Elektronika:*  
     - Készítsük el az áramkört. Érdemes kihasználni a közös földelést és tápot, hiszen sok dróttal dolgozunk majd.
  3. *Kód:*  
     - Egeszítsük ki a lenti kódvázlatot.  
     - Vegyük át a motor mozgatásához szükséges metódusokat, pl. Forward, backward, stop.  
     - Vegyük át dht11 objektum metódusait, pl. temperature és humidity.   
     - Vegyünk példát a try except szerkezetre és magyarázzuk el miért fontos:  
       
     a = 3

b = 0

c = a/b  
  
try:

c = a/b

except:

print('Hiba tortent c ertekeben, ezert konstans erteket rendelunk hozza')

c = 2

print(f'c = {c}')

* 1. *Tesztelés:*  
     - teszteljük a programot.
  2. *Fejlesztés:*

-Módosítsuk a ventillátor vezérlést úgy, hogy akkor kapcsoljon ki, ha a hőmérséklet 28 ∘ C alá vagy a páratartalom 40% alá csökken. A ventillátor bekapcsolásának feltétele maradjon változatlan.

-A hőmérséklet és páratartalom kiolvasásánál, ha az sikertelen, oldjuk meg, hogy ne 0 értéket írjon ki, hanem az előző mért értéket. Ehhez a temperature\_functions modulban a get\_hum és get\_temp függvényeket kell módosítani.

Kód minta:

Temperature\_functions\_studentVersion.py:

def get\_temp(dev):

    '''

    dev: DHT11 eszkozt szimbolizalo objektum

    '''

    try:

        t = # olvassuk ki a dht11 objektumbol a homersekletet a temperature attributumot meghivva (nem metodus es nem fuggveny)

    except:

        t = 0

    return t

def get\_hum(dev):

    '''

    dev: DHT11 eszkozt szimbolizalo objektum

    '''

    try:

        hum = # olvassuk ki a dht11 objektumbol a paratartalmat a humidity attributumot meghivva (nem metodus es nem fuggveny)

    except:

        hum = 0

    return hum

def ventillation(m, temp, hum):

    if temp and hum:

        if .......:  # ha homerseklet > 30 celsius fok vagy paratartalom > 50 %

            # forgassuk a motort elore

        else:

            # forgassuk a motort hatra

Prog06….

# importalni az adafruit\_dht csomagot

# importalni a time csomagot

# a gpiozero csomagbol a Motor klasszt importalni

# a temperature\_functions\_studentVersion fajlbol a get\_hum, get\_temp, ventillation fuggvenyeket importalni

dev = # inicializalni a DHT11 eszkozt

motor = # inicializalni a motort. 3 bemeno parameter: elore forgato pin, hatra forgato pin es az enable pin szama

while True:

    t = # olvasd ki a homersekletet a get\_temp fuggvennyel

    h = # olvasd ki a homersekletet a get\_hum fuggvennyel

    # nyomtasd ki a kepernyore a homersekletet es a paratartalmat: pl. megjeleno szoveg legyen: Temp: 23 Celsius degree , Hum: 50 %

    # hivd meg a ventillation fuggvenyt ami iranyitja a motor bekapcsolasat

    time.sleep(0.5)