**2. óra: Hőmérséklet kiolvasás**

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. Mindenre válaszoló gép:
   1. *Bevezető*:  
      Első projektünkben megismerkedünk a DS18b20 szenzorral és, hogy hogyan tudunk belőle pythonból információt kinyerni. Ennek első feltétele, hogy a bevezetőben leírt 1-Wire protokolt már aktiváltuk a Raspberry Pi-on.  
      A hőmérséklet kiolvasásának a logikája a következő lesz. Pythonból lefuttatjuk a terminál parancsokat, amik lehetővé teszik a számítógép és a thermisztor közti kommunikációt, majd szoftveresen kiolvassuk a w1\_slave fájlból a hőmérsékletet..
   2. *Előkészületek*:  
      Készítsünk ki próbapaneleket, jumper kábeleket, 4.7kOhm ellenállásokat, DS18B20 thermisztorokat. Emellé szükség lesz mérőműszerre is.   
      - Át kell beszélni a thermisztorok fontosabb tulajdonságait, pl. Mekkora feszültséget (teljesítményt) lehet rá adni.   
      - Ennél a projektnél főként a programozáson van a hangsúly.  
        
      *Részletekre bontás:*Elkészíteni az áramkört.
      * Előkészíteni a kommunikációt a thermisztorral (modprobe parancsok futtatása).
      * Definiálni a hőmérsékletet tartalmazó fájl lokációját.
      * Kiolvasni a fájl tartalmát az open parancs segítségével.
      * Definiálni egy függvényt ami a fájlból kinyeri a hőmérsékletet.
      * Másodpercenként megismételni a hőmérséklet olvasást.
      * változtatjuk úgy követjük a feszültség változását, amiből következtethetünk az ellenállásra.
   3. *Elektronika:*  
      - Készítsük el a feszültségelosztó kapcsolást.
   4. *Kód:*  
      - gyakorlojuk az *os.system() parancsot,* amivel pythonon kívüli parancsokat tudunk lefuttatni pythonból.  
      - *glob.glob()* parancs megismerése, gyakoroljunk vele néhány dolgot megkeresni pl. *Glob.glob(’\*’)*  
      - fájl műveletek gyakorlása, fájl tartalmának kiolvasása, pl. *f = open(filename, ‘r’)  
      -* gyakoroljuk a sorok beolvasását: *f.read()* és ne felejtsük el a kommunikációs csatorna lezárását, *f.close()*  
      - gyakoroljuk a string műveleteket, lásd a példákat a leírásban.  
      - ha bemutattuk a program minden alrészletét, akkor fejezzék be az elkezdett részprogramot.
   5. *Tesztelés:*  
      - teszteljük a thermisztorokat a műszerrel, illetve teszteljük a programot.
   6. *Fejlesztés:*  
      - Módosítsuk a kódot úgy, hogy ne csak a hőmérsékletet írja ki, hanem az hőmérséklet leolvasásának időpontját is.

- Ha a hőmérséklet 30 Celsius fok fölé emelkedne, nyomtassuk ki a képernyőre, hogy Kánikula (és ne írassuk ki a hőmérsékletet) ameddig vissza nem csökken 30 Celsius fok alá.

Kód minta:

import os

import glob

import time

os.system('.....')    # modprobe w1-gpio

#  futtasd le a modprobe w1-therm parancsot is

base\_dir = '/sys/bus/w1/devices/'

device\_folder = glob.glob(......)[0]    # keresd meg van e '28'-al kezdodo fajl a base\_dir mappaban

device\_file = device\_folder + '/w1\_slave'

def read\_temp\_raw(file\_name):

    f = open      # nyisd meg a file\_name nevu fajlt olvasasra

    lines = f.   # olvasd ki a fajlbol az ossz sort egy listaba

    f.           # zard be a megnyitott kommunikaciot

    return lines

def read\_temp(file\_name):

    lines = # olvasd be a file\_name tartalmat a read\_temp\_raw fuggvennyel

    while lines[0].strip()[-3:] != 'YES':

        time.sleep(0.2)

        lines = read\_temp\_raw(file\_name)

    equals\_pos = lines[1].    # talald meg a lines lista masodik elemeben melyik karakter pozicional van a t= string

    if equals\_pos != -1:

        temp\_string = lines[1][equals\_pos+2:]

        temp\_c = #  alakitsd at a megtalalt homersekletet float tipusba es oszd le ezerrel, hogy celsius fokban legyen a homerseklet

        temp\_f = temp\_c \* 9.0 / 5.0 + 32.0

        return temp\_c, temp\_f

while True:

    print(read\_temp(device\_file))

    time.sleep(1)