**3. óra: Hőmérséklet kimentése fájlba**

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. Hőmérséklet kimentése fájlba:
   1. *Bevezető*:  
      Mint házi ügyeletes szakit, a szüleid megbíztak, hogy készíts egy szoba hőmérsékletét rögzítő berendezést, ugyanis szeretnék látni és kielemezni, hogy különböző időpontokban, hogy változik az otthonotokban a hőmérséklet.

Arra kértek, hogy készíts egy programot, ami az általad megadott ideig méri, az általad megadott időközönként az otthon hőmérsékletét. Az eredményt pedig egy fájlba menti le, ami tartalmazza a mérés időpontját és a hőmérsékletet Celsius és Fahrenheit fokokban.

* 1. *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.
     + Definiálni egy függvényt, ami előkészíti a kimentendő információt (string formában)
     + Definiálni egy függvényt, ami kimenti az információkat egy fájlba.
     + Időmérést elindítani és *while* ciklusban nézni az eltelt időt.
     + Kiolvasni az időpontot, hőmérsékletet és elmenteni fájlba.
  2. *Elektronika:*  
     - Készítsük el a feszültségelosztó kapcsolást.  
     - Ismerkedjünk meg a *datetime* csomag műveleteivel (helyezzünk hangsúlyt az idő string formába alakításra):

*Kód:*  
  
now = dt.datetime.today() *# elmentjuk a mostani pillanatot,*

*# EV-HONAP-NAP ORA:PERC:MS*

print(now)

tomorrow = dt.datetime(2021,2,17) *# letrehozunk egy datetime valtozot*

print(tomorrow)

print(now.minute) *# lekerjuk a valtozo PERC erteket*

print(type(now)) *# kinyomtatjuk a datetime tipusat*

now\_string = now.strftime('**%d**/%B/%Y') *# atalakitjuk a datetime objektumot*

*# stringge NAP/HONAP/EV alakban*

print(now\_string)

print(type(now\_string))

date\_str = '1999-08-11 15:12:33' *# string formaban idopont definialas*

*# string idoforma datetime formava alakitasa*

date\_dt = dt.datetime.strptime(date\_str, '%Y-%m-**%d** %H:%M:%S')

print(date\_dt)

print(date\_dt.month)

print(type(date\_dt))

- fájl műveletek gyakorlása, fájl tartalmának kiolvasása, pl. *f = open(filename, ‘a’)*  
- gyakoroljuk a string műveleteket, lásd a példákat a leírásban.  
- gyakoroljuk az eltelt idő mérését a *time.time()* függvénnyel.  
- ha bemutattuk a program minden alrészletét, akkor fejezzék be az elkezdett részprogramot.

* 1. *Tesztelés:*  
     - teszteljük a thermisztorokat a műszerrel, illetve teszteljük a programot.
  2. *Fejlesztés:*  
     - Változtassuk a mérés hosszát és gyakoriságát.

- Módosítsuk a mért adatok lementését úgy, hogy az első sor a fejléc legyen, és a következőket tartalmazza: Datum,t\_C,t\_F.

Kód minta:

import os

import glob

import time

import datetime as dt

os.system('modprobe w1-gpio')

os.system('modprobe w1-therm')

base\_dir = '/sys/bus/w1/devices/'

device\_folder = glob.glob(base\_dir + '28\*')[0]

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

filename =     # add meg a fajl nevet

duration =     # add meg a mintavetelezes hosszat masodpercben

def read\_temp\_raw(file\_name):

    f = open(file\_name, 'r')

    lines = f.readlines()

    f.close()

    return lines

def read\_temp(file\_name):

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

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

        time.sleep(0.2)

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

    equals\_pos = lines[1].find('t=')

    if equals\_pos != -1:

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

        temp\_c = float(temp\_string) / 1000.0

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

        now =   # mentsd el az aktualis idot es alakitsd at string formaba '%Y-%m-%d %H:%M:%S'

        return temp\_c, temp\_f, now

def write2file(txt,filename='temperature.csv'):

    # nyisd meg a filename nevu fajlt , append formaban

    # ird bele a txt valtozo tartalmat

    # zard be a fajlt

def prepare2save(tc,tf,ts):

    # a txt valtozohoz add hozza az idopontot, homersekletet celsius, homersekletet fahrenheitban (string formaban) vesszovel elvalasztva

    return txt

t0 = # inditsd el az ido szamlalasat (referencia ido)

t1 = # ez a valtozo tarolja majd az aktualis idot, amit minden ciklusban felulirunk

while .......:     # Az aktualisan eltelt idonek kisebbnek kell lenni mint a duration valtozo

    tc, tf, ts = read\_temp(device\_file)

    t1 = time.time()

    print(t1-t0)

    text = # alakitsd at a mert adatokat stringbe a prepare2save fuggvennyel

    # ird ki fajlba a text valtozo tartalmat a write2file fuggvennyel

    time.sleep(1)