**8. óra: Marouder térkép**

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*:  
      A Harry Potter féle varázslóvilágban a Weasley családnak volt egy érdekes mágikus órája. Küllemre egy egyszerű, régi típusú kakukkos órára hasonlított, csak épp nem az időt mutatta. Több mutatóval rendelkezett. Minden mutató egy családtagot képviselt. A mutatók mögötti hátlapon helyszínek nevei álltak össze betűkből, jelezve, hogy a családtagok épp merre találhatók.

Ebben a projektben egy hasonló berendezést készítunk el, csak épp egyszerűbb kivitelben és nem használunk hozzá varázslatot (tudom, így már nem is olyan izgi). Egy olyan órát készítünk, ami egy mutatóval rendelkezik és azt jelzi, hogy egyedül vagyunk-e otthon vagy sem, arra alapozva, hogy a routerünkre felcsatlakozott-e valakinek a mobilja vagy sem. Természetesen attól, még, hogy valaki mobilja nem ment fel a router wifijére, nem azt jelenti, hogy az illető nincs a házban, de a mai világban nehéz elképzelni, hogy valaki ne netezzen a telefonján keresztül.

* 1. *Előkészületek*:  
     Az áramkor nem áll sok elemből, így könnyen el tudják a diákok készíteni. Érdemes előre letárgyalni, hogyan állapítjuk meg melyik a közös drót az 5-ből. Ha van idő és nem ellenkeznek, akkor keressük ki a mobiltelefonokból a mac és az IP címeket, amit beírhatunk a *data* változóba a kódban.   
     A program esetében is érdemes számukra egy fél kész kódot adni, amit csak be kell fejezniük (erre egy példát ennek a dokumentumnak a végén látunk). A *pass* sort kell helyettesíteni a megfelelő kóddal, illetve a kommentek jelzik, hogy hol mit kell csinálni.  
       
     *Részletekre bontás:*- Először közösen vegyük át hogyan működik és mit csinál a *ping* parancs. Próbáljuk ki a következő parancsokat és magyarázzuk el a kapott eredményt:  
     ping www.google.com

ping 192.168.0.1  
Ugyanezt pythonban is lefuttathatjuk:  
output = subprocess.check\_output(f'ping www.google.com', shell=True)

print(output)  
print(output.decode())  
Ezeket a parancsokat kipróbálhatjuk a mobilok IP címével is.   
  
- Győződjünk meg, hogy az *arp* parancs fel van installálva a gépre. Ha igen teszteljük és magyarázzuk el mi az eredmény.   
arp –a  
output = subprocess.check\_output('arp -a', shell=True)

print(output.decode())

- Teszteljük a motorok működését és állítsuk be a megfelelő pozíciókat, hogy a mutató elforgatása a megfelelő képekre mutasson.

* 1. *Elektronika:*  
     - Kössük össze a motort a vezérlővel. Kössük a vezérlőre az elemeket és a Raspberry pineket.
  2. *Kód:*  
     - Importáljuk a megfelelő csomagokat.  
     - Inicializáljuk a léptető motort.  
     - Definiáljuk a *data* változót, amibe beírjuk a megfelelő információkat.  
     - Elindítunk egy végtelen ciklust  
     - Definiáljuk a *people\_at\_home* változót, ami számolja, hány felhasználó van otthon  
     - Lefuttatjuk az *arp* parancsot, és eltároljuk az infóját.  
     - Végiglépkedünk a *data*-ban megadott IP címeken és megpingeljük őket. Ez frissíti a tárolt IP címeket, így ha épp csak most csatlakozott valaki, az is a rendszerben lesz.  
     - Végiglépkedünk a megadott *mac* címeken és megnézzük benne vannak-e az *arp* parancs eredményében. Ha igen, megnöveljük a *people\_at\_home* változó értékét. Evvel jelezzük, hogy valaki otthon van, hiszen rácsatlakozott a netre.   
     - for ciklus után megvizsgáljuk, hogy hányan vannak otthon, ha nincs senki, akkor a motort a nincs senki állásba forgatjuk (vagy ott hadjuk ha előtte is ott állt), vagy a nem vagy egyedül állásba ha valaki van otthon.
  3. *Tesztelés:*  
     - teszteljük a programunkat és javítsuk a hibákat ha vannak
  4. *Fejlesztés:*  
     - LED-ekkel szimbolizálni egy-egy személyt és egy gomb segítségével kiválasztani valakit, akiről a szerkezet megmondja otthon van-e vagy sem.

- Egy óra hátlapja fölött forgatni a léptető motort és a hátlapon lévő számok jeleznék, hány eszköz csatlakozott a hálózathoz.

- Kiírni a képernyőre, hogy ki van otthon.

Kód minta:

import subprocess

import time

from stepper\_motor import StepperMotor

m = # inicializalni stepper motort

data = [{'name':'Phone1', 'mac':"4a-c0-56-a9-1a-46", 'ip':'192.168.0.166'},

        {'name':'', 'mac':"", 'ip':''}     # toltsd ki az eszkoz nevet, mac cimet, ip cimet string formatumban

]

while True:

    people\_at\_home = 0

    print("starting loop")

    output = subprocess.check\_output(......., shell=True)   # a pontok helyere ird be az arp paranccsal valo vegrehajtast

    print ("starting scan")

    for item in data:

        outp = subprocess.check\_output(f'ping -W 2 -c 3 {......}', shell=True)   # a pontok helyere ird hivd meg az item valtozo ip cimet

        if item['mac'] in output.decode() or item['mac'].replace('-',':') in output.decode():  #

            people\_at\_home = people\_at\_home + 1

    if people\_at\_home == 0:

        if .....:    # ellenorizd le, hogy a motor szoge 180 foknal van-e

            pass     # ha igen, forgasd a motort hatrafele 180 fokkal es a sebessege legyen 1

    else:

        if .......  :  # ellenorizd le, hogy a motor szoge 0 fok e

            pass     # ha igen, forgasd a motort elore 180 fokkal es a sebessege legyen 1

            time.sleep(2)

    time.sleep(5)