**Helló Alexa**

**Mit fogsz készíteni?**

Ez a leírás segít egy házilag elkészíthető okosotthon menedzser elkészítésében, mint amilyen az Amazon Alexa vagy a Google Home.

Az általunk írt kód jóval butább lesz mint a kereskedelemben kapható testvérei, de jól szemlélteti, hogy akár néhány sor kóddal képesek vagyunk hangfelismerő, értelmező és utasítást végrehajtó rendszert készíteni.

Ez a projekt egy egyszerűsített okos terminál elkészítésében segít, ami 3 témakörben jártas:

* led diódákat kapcsol fel és le
* zenét játszik
* megmondja az időjárást.

Egyelőre a terminál csak **angol nyelven** tud mondatokat értelmezni.

**Mit tanulsz meg?**

Evvel a projekttel talán az informatika és mérnök szakmák egyik legfontosabb gondolatmenetét tanuljuk meg, hogy ne fedezd fel újra a kereket, azaz próbálj már működő rendszereket újra felhasználni. Megtanuljuk a mások által írt összetett kódok alkalmazását. Emellett a következőket fogod még megtanulni:

* Hogyan kell led diódákat összeszerelni és a Raspberryre kapcsolni.
* Hogyan kommunikálunk szóban vagy írásban a számítógéppel.
* Hogyan építsünk fel egy logikai sorrendet a hang (szöveg) felismerés és vezérlés értelmezésére.

**Milyen angol utasításokkal érdemes próbálkozni:**

**Világítás:**

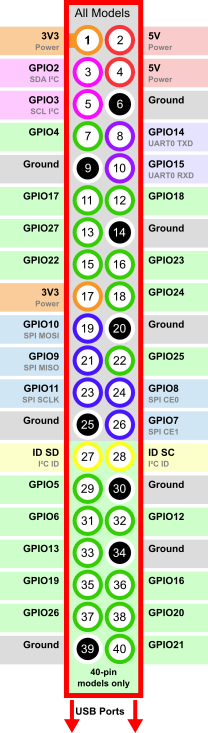
Három különböző színű led dióda áll a rendelkezésedre és ezeket tudod:

* felkapcsolni
* lekapcsolni
* villogtatni (blink)

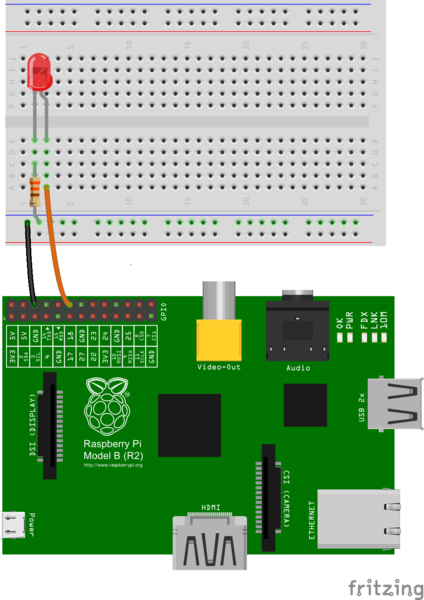
**Zene:**

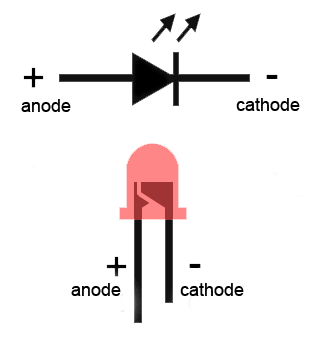
* zenét játszani (a saját adatbázisából)
* zenét leállítani
* előző/következő számot kérni
* megkérdezni az előadó nevét, zeneszám címét

**Időjárás:**

* megkérdezni milyen most az időjárás a következő helyeken: Szeged, Budapest, Debrecen, Pecs, London, Paris, New York, Madrid, Belgrade, Wien, Stockholm, Coppenhagen, Berlin
* megkérdezni milyen lesz ma az időjárás
* ****megkérdezni milyen lesz holnap az időjárás

**LEDek összekötése a Raspberry Pi-al:**

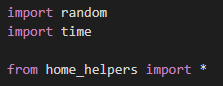
****

1. A LEDek bekötéséhez a fenti ábra szolgál útmutatásként. Szükség van 3 különböző színű LED-re, 3 ellenállásra és 4 db csatlakozó kábelre. A GPIO pinek elosztását a szélső pin rajz mutatja.
2. Az áramkör összeszereléséhez érdemes kikapcsolni a Raspberry Pi-t, hogy még véletlenül se okozhassunk kárt benne rövid zárlatok létrehozásával.
3. Helyezzünk egy kábelt (az ábrán a fekete kábel) a breadbord oldalsó sávjának legszélső lyukjába, a másik végét pedig csatlakoztassuk a GPIO kimenetelek földelésére (GND), pl a 6-os pinre. Ez a csatlakozó úgy fog viselkedni, mint egy elemnek a negatív vége.
4. ****A **KÉK** LED-et és az ellenállást helyezzük el a rajznak megfelelően. Figyelj oda, hogy az ellenállás a LED azon lábára legyen kötve amelyik a vastagabb zászlóhoz csatlakozik a LED-ben. Ez a LED negatív karja.
5. A LED pozitív karjához csatlakoztass egy kábelt, majd a másik végét a 40-es számmal ellátott pinre, ami a GPIO21-es kimenet. A kód írásakor a **21**-es GPIO kimenetre kell majd hivatkozni. Ez a kimenet utasítás hatására képes 0 V-ról 3.3 V-ra változtatni a feszültségét, ami világításra készteti a LED-et.
6. A 3-5-ös pontokat ismételjük meg először egy **PIROS** LED-el, néhány lyukkal a kék mellett elhelyezve, illetve a kábelt a 36-os pinre kötve ami a **16**-os GPIO kimenet.
7. Majd újra ismételjük meg a 3-5-ös pontokat csak most **ZÖLD** LED-el, amit kössünk a 32-es pinre ami a **12**-es GPIO kimenetnek felel meg.
8. A végső kapcsolásnak nagyjából úgy kellene kinézni mint ahogyan a lent látható képen.

Öszzeszerelt kapcsolás képének illesztése ide.

**Írd meg a Hello Alexa programot Pythonban**

1. Nyisd meg a **VALAMELYIK** alkalmazást a *Programming* menüben.
2. Kattints a **Save** ikonra és mentsd el a fájlt **alexa.py** név alatt.
3. Gépeld be a következő kód részletet:

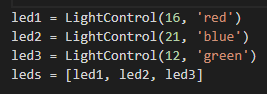


A kódnak ezen része beolvassa azon könyvtárakat, amiket fel fogsz használni a kódod írása során.

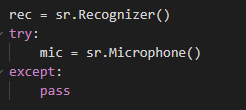
1. A következőkben megadjuk a programnak, hol találja meg a zenéket, **music\_path** amiket lejátszhat, illetve létrehozunk egy objektumot **music\_player**, ami kezeli a zenelejátszót.



1. A LED-ek irányításához létrehozzuk a LED kapcsoló objektumokat **led1, led2, led3**, ahol definiáljuk azt, hogy a LED-ek melyik GPIO-ra lettek rákapcsolva illetve megmondjuk a LED színét:



1. Definiáljuk a számítógépnek a hangfelismerő **rec**, és mikrofon **mic**, felismerő rendszert. Figyelj a sorok tördelésére (bekezdésekre), Pythonban nagy szerepük van.



A try-except páros ellenőrzi, hogy van-e mikrofon kapcsolva a számítógéphez.

1. Létrehozzuk a program leállításáért felelős objektumot **quit\_class**.



1. Most jön a program fő része a **while** ciklus, ami a végtelenségig futtatja a programot, azaz addig amíg ki nem mondjuk, hogy ***shut down*** vagy ***exit***. Ebben a ciklusban kommunikál a számítógép a felhasználóval, pl. megkéri, hogy csak ****az ***Enter*** lenyomása után kezdjen utasításokat adni. Ezután aktiválja a mikrofont és rögzíti és szöveggé alakítja



az elhangzott utasítást. Ha véletlen hiba lépne fel a hang rögzítésében, akkor átkapcsol szöveg begépelési üzemmódba majd kiírja a képernyőre a hallott vagy begépelt szöveget.



1. Ezután következik a szöveg értelmezése. Először megnézi a program, hogy volt-e kikapcsolásra utasítás. Ha igen akkor leállítja a kódot.



Ha nem akkor először megpróbálja meghatározni, hogy milyen témakörben kapott utasítást, világítás, időjárás, zene vagy valami más. Erre a **SpeechMap** nevű objektum hivatott.



Végül pedig végrehajtja a megfelelő témakörhöz tartozó utasításokat.



1. ****A végső kódnak nagyjából így kell kinéznie:

