**Szegedi Tudományegyetem Informatikai Intézet**

Beltéri hangvezérelt megvilágítás megvalósítása ESP32 mikrovezérlővel

Indoor voice-controlled lighting

using ESP32 microncontroller

Szakdolgozat

|  |  |
| --- | --- |
| Készítette: | Témavezető: |
| **Fodor Teodor** | **Dr. Mingesz Róbert** |
| mérnökinformatikus szakos hallgató | egyetemi adjunktus |

Szeged 2024

# Feladatkiírás:

ESP32 mikrovezérlővel számos IOT-projektet készíthetünk. A mikrovezérlő sokoldalúságának és könnyen proramozhatóságának köszönhetően a lehetőségek tárháza végtelen. Gyakori jelenség, hogy az épületek beltéri megvilágítását különböző elven működő érzékelővel vezérlik. A diplomamunkám célja, hogy a megszokott vezérléseket figyelmen kívül hagyva, egy sokkal kényelmesebb és képzeletet felülmúló, hangvezérelt megoldással álljak elő, amely lehetőleg gyors, könnyen felhasználható és alacsony hibaárányú. A dolgozat írása során természetesen törekszem megtalálni az arany középutat a felhasznált komponensek árában és minőségében, hiszen a legjobb ötlet sem valósítható meg a megfelelő céleszközök nélkül.

A feladat megvalósításának lépései:

1. Pontos specifikáció meghatározása
2. ESP32 fejlesztőkit és mikrofonok kiválasztása
3. Hardverek tesztelése (osszcilloszkóp, egyéb mérések)
4. Program implementálása
5. Program tesztelése
6. Áramkör elkészítése próbapanelen
7. Program javítása és bővítése

# Tartalmi összefoglaló:

## A téma megnevezése:

Beltéri hangvezérelt megvilágítás megvalósítása ESP32 mikrovezérlővel

## A megadott feladat megfogalmazása:

A feladat lényege, hogy egy két zónás led-világítás tudjunk hanggal vezérelni.

## A megoldási mód:

A kiválasztott ESP32 fejlesztőkit és mikrofonok közötti kommunikáció szoftveres megvalósítása.

## Alkalmazott eszközök, módszerek:

A programot a hivatalos, Espressif cég által biztosított ESP-IDF keretrendszert felhasználva készítem el, FreeRTOS-s környezetben. A kapcsolást először breadboard-on rakom össze, majd egy próbapanelen összeforrasztom (a két műveletet megelőzi egy áramkörtervezés).

## Elért eredmények:

Mind a program, mind az áramkör is elkészült, így a szobámban már megtalálható a működő prototípus.

## Kulcsszavak:

ESP-IDF, ESP32, I2S, INMP441, FreeRTOS

# Bevezetés:

A hangvezérlés egy olyan technológia, amely lehetővé teszi a különböző eszközök és alkalmazások vezérlését beszédhanggal. A hangvezérlést biztosító rendszerek használata egyre elterjedtebb a mindennapi életben, például okostelefonok, autók és egyéb elektronikus készülékek esetében, illetve ma már egy korszerű okosotthon szinte teljesen elképzelhetetlen hangvezérlés nélkül. A hangvezérlés egy olyan alternatíva, amely egy érintésmentes és kényelmes szolgáltatást biztosít a hagyományos vezérlési módszerekhez képest.

Felmérések alapján a felhasználók nagy része többnyire a következő 4 platform mellett teszi le a voksát:  Apple Siri, Samsung Bixby, Amazon Alexa vagy Google asszisztens. A felsorolt szolgáltatások mindegyikére jellemző, hogy online, azaz felhő-alapú hangfelismerő algoritmusokat futtatnak nagy teljesítményű szervereken, a platformokat támogató termékek (Amazon echo dot, Amazon echo studio, Apple Homepod, Samsung televízió, stb.) valójában egy mikrofonként szolgálnak, amelyek továbbítják a beszédhangot az előbb említett távoli számítógépeknek. Ezen rendszerek általában drágábbak illetve a használatukhoz egyszerre több terméket is meg kell vásárolnunk. A diplomamunkámmal azt próbálom bemutatni, hogy lehetőség van jutányos áron beszerezni, kiépíteni egy olyan rendszert, amellyel egyszerű eszközök is hangvezéreltté tehetők. Például egy Amazon Alexát támogató villanykörte körülbelül 30 euróba, a hozzá tartozó legegyszerűbb Amazon Echo Dot 50 euróba kerül - az ellátott feladatot természetesen teljesítik az eszközok, azonban az ár egy kissé magas (az Alexás termékek árai messze elmaradnak az Apple termékeihez képest) egyetlen funkcionalitás megvalósításához. Az elkészített projekttel be szeretném mutatni, hogy sokkal kevesebb anyagi ráfordítással is megteremthető a fenti kényelmi funkció.

Fontos megjegyezni, hogy ha az ember ESP32 mikrovezérlővel megvalósított hangfelismerés offline, azaz az algoritmus magán az eszközön fut, ezért egy erősebb chip megvásárlása szükséges, amelyben több memória található, illetve az órajele is sokkal magasabb. Az ilyen chipek ára még mindig messzemenőleg elmarad egy felhő-alapú rendszerhez képest. A legegyszerűbb megoldás az, ha az ember vásárol egy olyan fejlesztőkitet, amiben már megtalálható a mikrofon is, így nem kell foglalkozni a hang eljuttatásával a mikrovezérlőbe. Találtam több kitet is, amelybe már integrálva voltak a mikrofonok, azonban a beszerzésük hosszadalmas és drágább folyamat lett volna. Ezért úgy döntöttem, hogy külön-külön beszerzem a fejlesztőkitet és a két mikrofont(lehetőleg olyan típusút, amelyek a hivatalos kitekbe is integrálva vannak) és elkészítem a közöttük lévő kommunikációhoz szükséges illesztőprogramot. A hangfelismerő algoritmust nem én készítetettem, már része az ESP-IDF keretrendszernek, és sajnos nem nyílt forráskodú a framework ezen könyvtára.

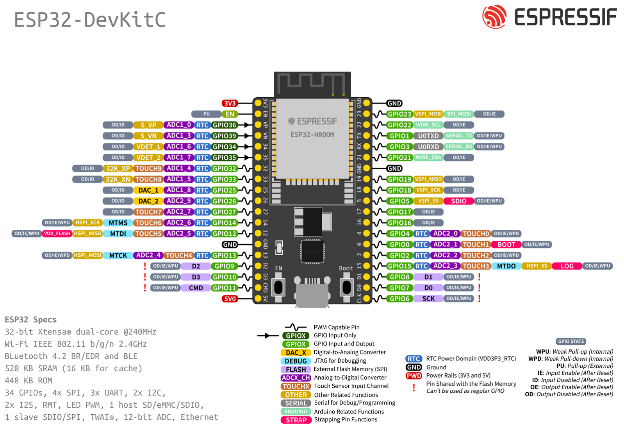
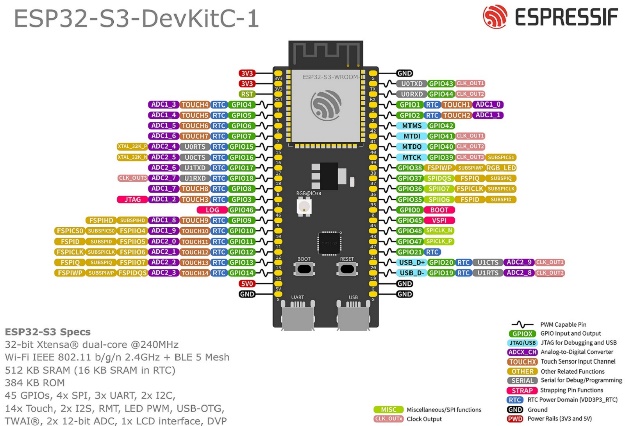
Nem titkolt célja a projektnek, hogy egy pénztárcabarát és könnyen használható, illetve konfigurálható eszközt tervezzek, amely egyszerűbb vezérlési feladatok elvégzésére képes.

# Hardverkörnyezet bemutatása:

A felhasznált hardverek kiválasztásakor alábbi négy szempont szerint jártam el:

Legyen költséghatékony, könnyen elérhető, lehetőleg gyors legyen a kiszállítás és ne legyen gyárilag hibás.

## A felhasznált mikrovezérlő és fejlesztőkit:

Az Espressif ESP termékcsaládja eddig is népszerű volt a hobbi építők és az olcsó IoT eszköz gyártók körében, révén hogy nagy tudása mellett, olcsó és könnyen programozható. Az elterjedt ESP8266 (illetve ESP8265 és változatai) leváltására érkezett az erősebb ESP32. Az ESP32 az elődjéhez képest sokkal több perifériával, nagyobb memóriával és magasabb órajellel rendelkezik. A programozható pinek száma megnőtt, illetve egyre több hálózati kommunikációs protokollt(Thread, Zigbee, Matter) is képesek támogatni a megszokott WiFi-n és Bluetooth-on kívül az újszériás ESP32 chipek. Az elmúlt évek során rengeteg változat jelent meg, mindegyik széria rendelkezik valamilyen specifikus képességgel, így az adott feladathoz megfelelően tudjuk kiválasztani a számunkra megfelelő chipet.

Ábra 1

Ábra 2

A diplomamunkámhoz az ESP32-S3 AI SoC-t választottam, amely a már jól bevált két processzormagos, akár 240 MHz-es órajel elérésére is képes Xtensa LX7 vezérlőegységet tartalmazza. A mikrovezérlőt kifejezetten AIoT alkalmazásokhoz gyártott le az Espessif Systems cég 2020-ban. Az ilyen chipet tartalmazó modulok különlegessége, hogy tartalmaznak egy külső PSRAM (Pseudostatic RAM) memóriaegységet, 2,4,8 és 16 MB-os kapacitással, verziótól függően. A PSRAM nagy méretének köszönhetően, lehetővé teszi a magas számításigényű mesterséges intelligencia algoritmusok (esetünkben hangfelismerő algoritmusok) futását, a nagy mennyiségű digitalizált hangadat pufferelését – kizárólag ezért használható csak az S3-as szériához a keretrendszer hangfelismerő könyvtára. A jelenlegi projekthez én ESP32-S3-DevKitC-1U-N8R8 chip-el ellátott ESP32-S3-DevKitC-1 v1.1 fejlesztőkitet választottam 8MB flash-memóriával és 8 MB PSRAM-al.

Ábra 3

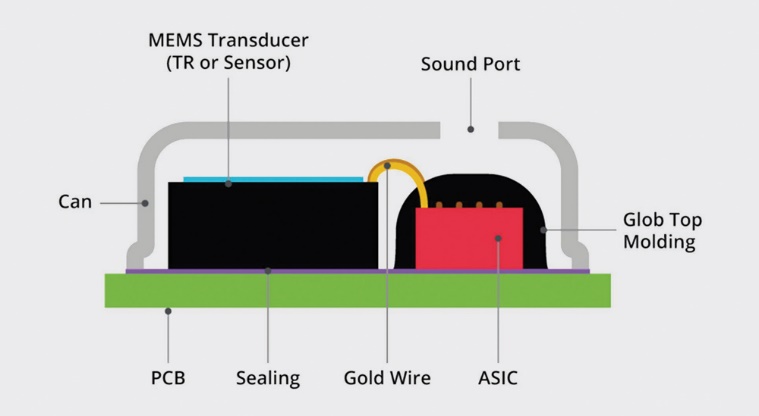
## A felhasznált mikrofonok:

Annak érdekében, hogy a beszédhang digitalizásálával ne kelljen foglalkoznom, mindenképpen egy MEMS-technológiát használó mikrofont választottam, illetve egy olyan kommunikációs protokollt, amellyel könnyen le tudom implementálni a mikrofon és a fejlesztőkit közötti adatátvitelt. Ha körül nézünk a boltok polcain, akkor legtöbb ilyen típusú mikrofon az I²S kommunikációs szabványt támogatja. Az ESP-IDF pedig rendelkezik olyan könyvtárral, amely a fenti kommunikáció lebonyolításáért felelős.

### A MEMS-technológia:

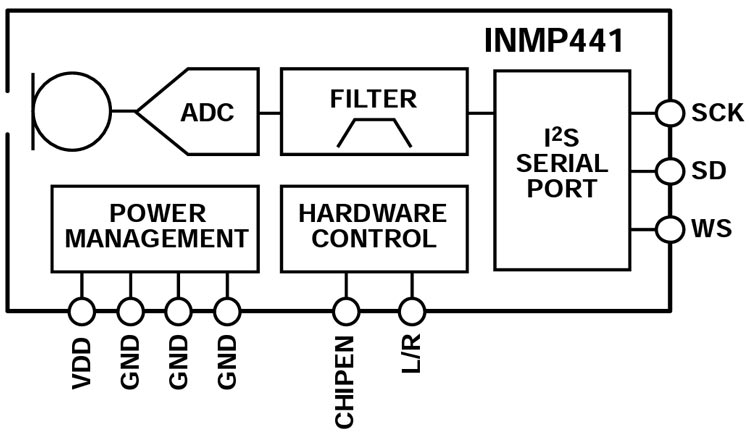
A MEMS mikrofonokat vagy szilíciummikrofonokat már régóta előszeretettel használják okostelefonokban, okostévékben, távirányítókban. Ez nagy részben olyan hangvezérelt személyititkár-alkalmazásoknak köszönhető, amilyen a korábban is más sokszor emlegetett Amazon Alexa, a Google Assistant és az Apple termékek Siri-je. Ezek az alkalmazások olyan hangparancsokra várnak, amelyek ébresztik magát a rendszert, majd a parancs alapján hajtanak végre különböző utasításokat. A tervezők szűrőfunckiókat is integrálnak a mikrofonokba, hogy azok pontosan, jelentős környezeti zajban is messziről érzékeljék a beszédhangot.

A MEMS mikrofonok jellemzően két, egyazon tokban elhelyezett alkatrészből állnak: egy MEMS membránból, amely átalakítja a hanghullámokat elektromos jellé, valamint egy erősítőből, amely impedanciaátalakító-ként működve használható analóg kimenőjelet ad a hangfrekvenciás jelfeldolgozó láncra. A harmadik alkotóelem – ha digitális kimenőjelre van szükség – egy analóg-digitális átalakító, amely szintén elhelyezhető ugyanabban a tokban, számunkra ez a fontos a projekt esetében.

Digitális MEMS mikrofon használata esetén a mikrofon kimenőjele közvetlenül rávezethető digitális áramkörökre, általában egy mikrovezérlőre Az erős villamos zajú környezetekre tervezett hangvezérléses felhasználói felületek jellemzően inkább a digitális mikrofonokat részesítik előnyben, mert a digitális kimenőjeleknek nagyobb a zajtűrésük, mint az analógoknak.

Ábra 4

### INMP441:

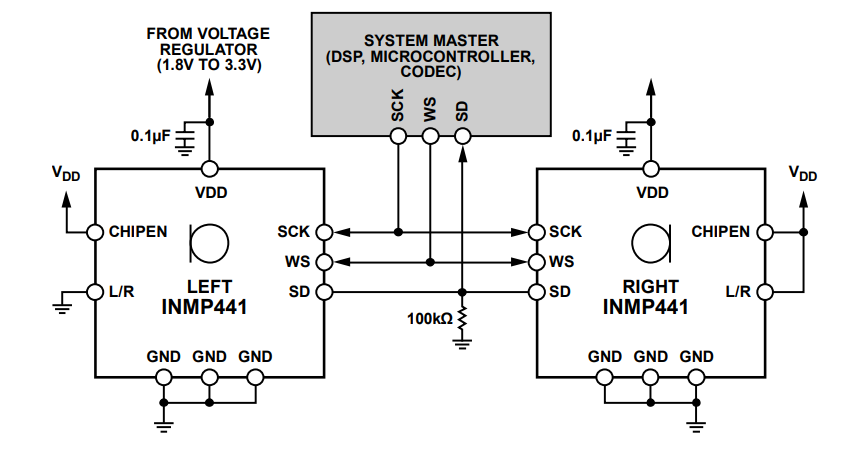
Az INMP441 egy olcsó és könnyen beszerezhető, alacsony fogyasztású MEMS mikrofon modul. Fontos tulajdonsága, hogy I²S kimeneti interfésszel rendelkezik és omnidirekcionális, amely azt jelenti, hogy minden irányból egyformán érzékeli a hangot. A kimenete 24-bites és könnyen alkalmazható sztereó módban, azaz egyszerre 2 azonos típusú mikrofont is könnyen használhatunk, az architektúrájából kifolyólag szinkronizálva van az adatátvitel.

Ábra 5

A fenti ábrán jól látható a „L/R” pin a mikrofon blokkdiagramján – ennek segítségével konfigurálható, hogy az eszköz a „jobb” vagy a „bal” oldali csatornán szolgáltassa a kimeneti adatot. Az alábbi ábra szemlélteti a sztereó-kimenet kapcsolását.

Ábra 6

Ábra 7

Jól látható, hogy az egyik mikrofon csip L/R pinje tápfeszültségre, míg a másik eszköz L/R pinje GND-re van csatlakoztatva. Az ábrán feltüntetett szűrőkondenzátorok és ellenállások már alapból megtalálhatóak a mikrofonos lapocskán, ezért használatuk elhanyagolható. Az ábra utolsó fontos eleme, a „SYSTEM MASTER” blokk esetünkben nem más, mint maga az ESP32 S3-as fejlesztőkit.

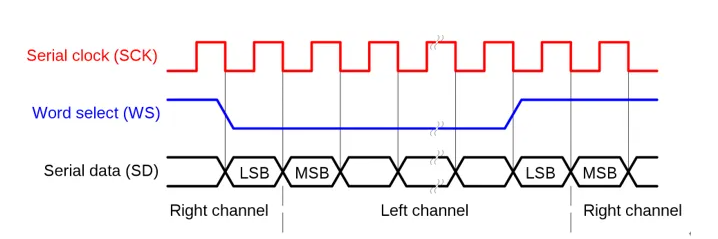
### I²S szabvány:

Az I²S (Inter-IC Sound) egy elektronikus soros busz interfész szabvány, amelyet digitális audioeszközök összekapcsolására használnak. Ezt a szabványt először a Philips Semiconductor vezette be 1986-ban. Digitális audió adatok átvitelére használják az elektronikus eszközök integrált áramkörei között. A kimenet PCM-formátumú.

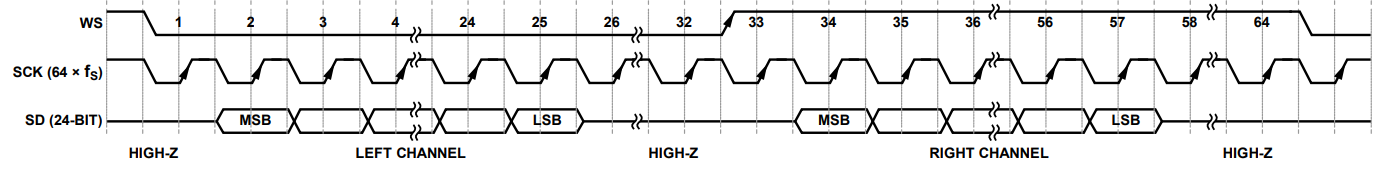
Összesen 3 jelalak jellemzi:

* Bit órajel – az általunk használt mikrofon esetében ez az SCK-pinen jelenik meg. Feladata az átviteli adat bitjeinek az ütemezése
* Szó óravonal – formálisan ez a mikrofon WS-pinjének a jele. Feladata, hogy sztereó kimenet esetén kiválasztja, hogy a jobb vagy bal oldali csatornáról szeretnénk fogadni az adatot. Logikai alacsony szint esetén a bal-, míg logikai magas jelszint esetén a jobb csatornáról történik az adat olvasása. Leegyszerűsítve ez a mintavételi frekvencia egy szóra vonatkozóan, ami esetünkben 24-bit (32 bit).
* Soros adatvonal – ezen az adatvonalon jelennek meg a mikrofon által digitalizált adatok bájtjai. Az INMP441 mikrofon esetén ez az SD-pin.

Az alábbi ábrákon jól látható, hogy mennyire egyforma, de mégis különböző az I2S idődiagramja elméleti szinten és alkalmazott mikrofok esetén.



Ábra 8 – Az I2s idődiagramja



Ábra 9 – Az INMP441 mikrofon I2S idődiagramja sztereó-kimenet esetén

Adatlap alapján a mikrofon kimenetén 24-bites adatok jelennek meg, amelyek a legnagyobb helyiértékű bittel kezdődnek. A dolog érdekessége, hogy mégis 32 impulzus van egy fél-WS-ciklus(1 darab csatorna olvasása ) folyamán. Ez azzal magyarázható, hogy a mikrofon ilyenkor egy magas impedanciás állapotba kerül – ezek a bitek számunkra közömbösek és a szoftveres lekezelésénél is figyelmen kívül hagyjuk őket. Kicsit nehezebben észrevehető, de még mielőtt kiolvasnánk a legnagyobb helyiértékű bitet a WS-ciklus elején szükségünk van egy bit órajel impulzusra, ezt a formátum a „Philips formátum”. Az ESP-IDF I2S könyvtárában ez szerencsére gond nélkül konfigurálható.

Fontos még kiemelni, hogy egy teljes WS-ciklus alatt 64 SCK impulzusnak kell megjelennie.

Például egy 16 kHz-es audió mintavételezési frekvencia esetén (gyakori frekvencia): a konfigurációnak a következőnek kell lennie: