Kiváló projektet raktál össze az ESP32-alapú alvásmonitor rendszerrel. A „Szenzorok” laboratóriumi bevezetője alapján az előadásnak strukturáltnak, mérnöki szempontból átgondoltnak kell lennie, és teljesítenie kell az alábbi fő pontokat:

**📊 ELŐADÁS VÁZLAT**

**1. Bevezetés – Célkitűzés (1 dia)**

* *Miért készült a rendszer?*
* Az alvásminőség monitorozásának jelentősége.
* Röviden a fő funkciókról: multiszenzoros mérés, WiFi-kommunikáció, eco mód, zajesemény detekció.

📌 *Illusztráció*: rendszer fotója vagy ikonikus ábra (ESP32, WiFi, szenzorok)

**2. Szenzortípusok kiválasztása (1–2 dia)**

* Minden szenzorra külön-külön:
  + Mit mér? Miért ezt választottad?
  + Ár/érték, interfész (pl. I2C), pontosság, programozhatóság.

| **Szenzor** | **Mért jellemző** | **Típus** | **Interfész** | **Ok a választásra** |
| --- | --- | --- | --- | --- |
| TCS34725 | Színhőmérséklet | RGB szenzor | I2C | Színalapú fényanalízis |
| AHTX0 | Hőmérséklet, páratartalom | Kombinált | I2C | Kicsi, megbízható |
| ENS160 | eCO₂, TVOC | Gázszenzor | I2C | Alvásminőségre utaló VOC-k |
| BH1750 | Fényerő | Luxmérő | I2C | Alvásbarát környezethez szükséges |
| INMP441 | Zaj | Mikrofon (I2S) | I2S | Zajesemény detekció |

📌 *Illusztráció*: táblázat + szenzoradatlap részlet (például ENS160)

**3. Rendszerterv és költségterv (1 dia)**

* Rendszerösszetevők listázása (ESP32, szenzorok, tápellátás).
* Költségek prototípusra bontva (durván: ESP32 ~2500 Ft, szenzorok összesen ~6-8000 Ft).
* Kitekintés tömeggyártás esetére (pl. egy PCB-re integrálva).

📌 *Illusztráció*: alkatrészlista táblázat, egyszerű költségdiagram.

**4. Hardveres és szoftveres design (2–3 dia)**

* Blokkvázlat: szenzorok → ESP32 → WiFi → szerver
* Power management: eco mód, CPU frekvencia csökkentés, szenzorok alvó módja
* Nem-blocking szenzor-kezelés: SensorTask szerkezet

📌 *Illusztrációk*:

* Egyszerű blokkdiagram
* Kódrészlet: eco mód váltás és WiFi kezelése (pl. maybeDisableWiFi(), applyEcoIntervals())

**5. Mérési elv, technika (1 dia)**

* Fizikai alapok (röviden): zaj → decibel, RGB → színhőmérséklet (CCT), VOC → levegőminőség.
* Kalibrációs kihívások, például zajküszöb, hiszterézis.

📌 *Illusztráció*: ábra RGB→CCT számításról, decibel RMS alapjai

**6. Adatküldés és szerver (1–2 dia)**

* Üzenetformátum: MsgPack
* REST API endpointok
* Dashboard funkciók: realtime kijelzés, logging, zip letöltés

📌 *Illusztráció*:

* Példa JSON / MsgPack payload
* Flask dashboard UI képernyőkép

**7. IoT aspektus (1 dia)**

* WiFi-kommunikáció (állandó vs. reconnect)
* Távkonfigurációs lehetőségek (/config, /set\_mode)
* Dashboardból vezérelhető funkciók: logging ON/OFF, eco mód

📌 *Illusztráció*: hálózati séma – ESP32 → Router → Laptop (szerverrel)

**8. Eredmények bemutatása (1 dia)**

* Naplózott adatok alapján készült grafikonok (temp, zaj, lux, VOC).
* Zajesemények kimutatása.
* Alvásbarát/nem barát státusz értékelés.

📌 *Illusztráció*: rendszer által generált session\_plots.zip néhány grafikonja

**9. Összegzés, jövőbeli tervek (1 dia)**

* Mi működik jól?
* Mit lehetne bővíteni? (pl. mobil app, gépi tanulásos értékelés)
* Hardveres miniatürizálás, hosszabb akkus működés.

📌 *Illusztráció*: bulletpoint összegzés

**📎 MELLÉKLET (ha engedélyezett)**

* Kódstruktúra rövid bemutatása (main.cpp, app.py, dashboard.html)
* Link vagy QR-kód: demóvideó / GitHub repo

Ha szeretnéd, segíthetek konkrét diák megírásában, ábrák rajzolásában (pl. blokkdiagram), vagy egy előadás-PDF összeállításában is. Szólj, mivel folytassuk!