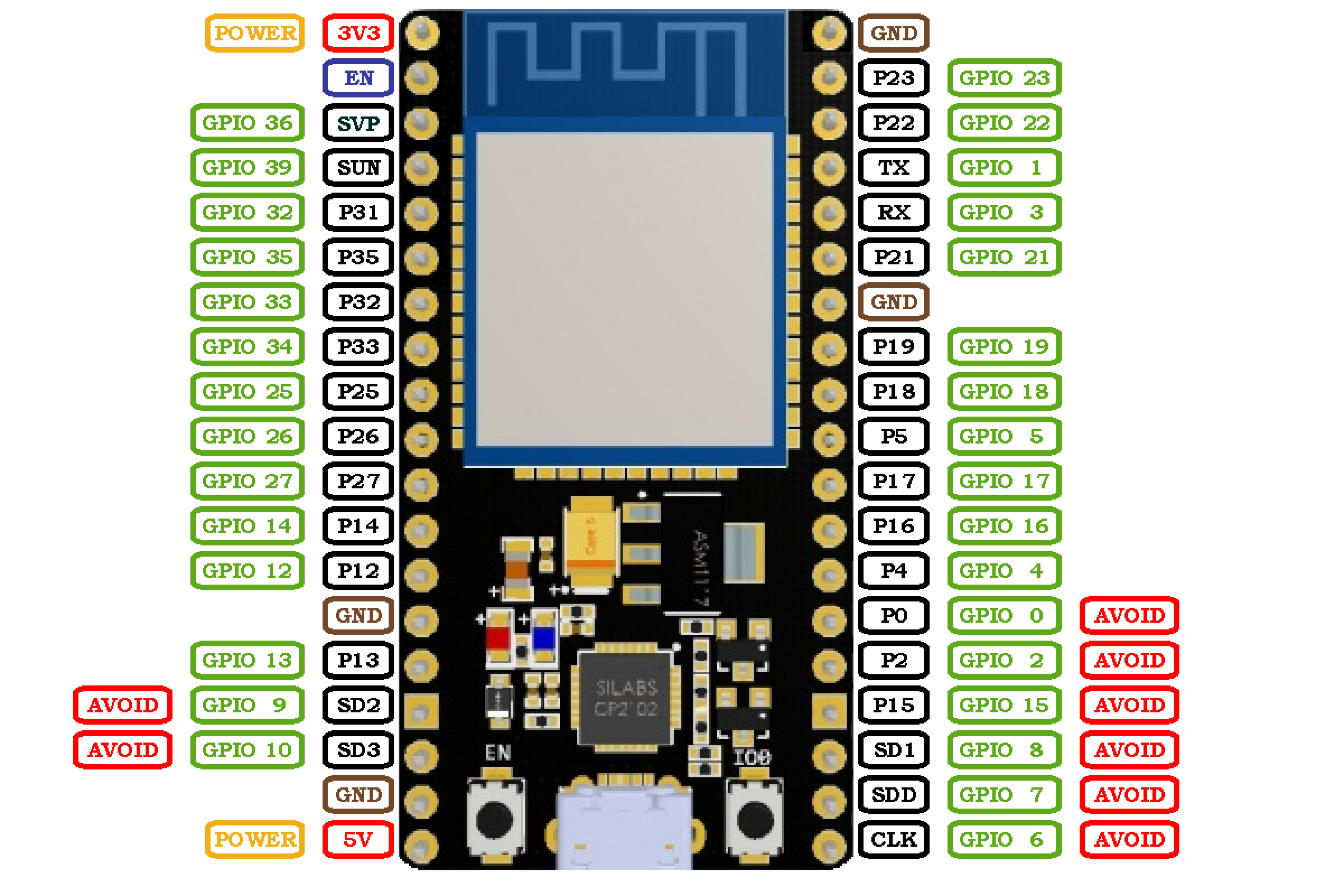
**A cansat programozása**

**1. Hardver:**

- **NodeMCU ESP-32S – mikrokontrollerek (2db)**

Pinout, a csatlakozások (saját készítésű):

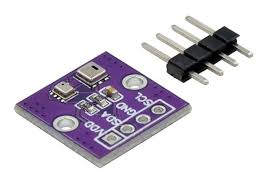


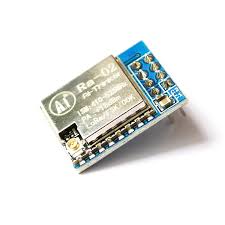
- **AHT20 – szenzor**

- Tudja mérni a hőmérsékletet és a levegő páratartalmát.

- **BMP280 – szenzor**

- Tudja mérni a hőmérsékletet és a légnyomást.

 -Ezt a kettő szenzort egy **egyesített szenzorban** használjuk:

 **-BMP280+AHT20:**

**- LoRa SX1278 – rádió transzmitter (2db)**

- Egy hosszú hatótávolságú, alacsony fogyasztású rádiótechnológia.

**2.Első lépések:**

Először is, fel kell töltenünk a Micropython- t a mikrokontrollerekre, mert a Python-t választottuk programozási nyelvként.

Ehhez először le kell tölteni az esptool nevezetű library-t (*pip install esptool* – a parancssorba).

Ha ez megvan akkor először töröljük a flash memóriát (*esptool.py erase\_flash*), majd feltöltjük a Micropython firmware-t. Ezt a fájlt a „<https://micropython.org/download/ESP32_GENERIC/>„ oldalon találjuk, majd ezzel a paranccsal feltöltjük az ESP32-kre: *esptool --chip esp32 write\_flash -z 0x1000* *<firmvare-hez vezető út (jobbklikk a fájlra, majd copy path)>.*

Most már gyakorlatilag fel vagyunk készülve arra, hogy elkezdhessünk programozni, már csak egy másik python library kell, ami segít a fájl transzferben és futtatásban. Ez a mpremote, amit ezzel a paranccsal lehet letölteni: *pip install mpremote*. Majd leteszteljük azt, hogy a számítógép érzékeli -e a csatlakoztatott ESP32-t, ezzel a paranccsal: *mpremote repl.* Ha ez valami olyasmit ír, hogy „*Connected to MicroPython at COM<egy szám, attól függ, hogy hova van dugva az usb>”*, akkor már tényleg kezdhetjük a programozást.

**3.Tesztelés:**

Most megbizonyosodunk, arról, hogy a szenzorjaink és moduljaink működőképesek.

Kezdjünk a szenzorokkal. Először le kell tölteni és fel kell tölteni a megfelelő szenzor drivereket a jeladó mikrokontrollerre, annak érdekében, hogy az ESP-32 el tudja olvasni a szenzorok által szolgáltatott adatokat. Az AHT20 szenzorhoz van hivatalos driver, amit erről a linkről lehet letölteni: <https://raw.githubusercontent.com/targetblank/micropython_ahtx0/master/ahtx0.py>.

A BMP280-nak nincs hivatalos drivere, de David Stenwall írt egy működő drivert, amit tudunk használni(köszi David Stenwall): <https://github.com/dafvid/micropython-bmp280/blob/master/bmp280.py>.

Most, hogy megvannak a driverjeink tényleg megírhatjuk a kódot, ami teszteli a szenzorokat. Ezt feltöltöttem a Cansat-ra létrehozott github repository-ba: <https://github.com/Grandturkboy/Kaboom_Cansat_2025/blob/main/Scripts/AHT20%26BMP280test.py>.

Jelenleg megvan minden komponensünk és végre beköthetjük a szenzort és feltölthetjük az ESP-32-re a kódot. A kötögetés nem bonyolult, négy csatlakozást kell létrehoznunk:

A képen elektronika, szöveg, Elektromos vezetékek, Elektrontechnika látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.**Szenzor Pin ESP32 Pin**

VDD ==> 3V3

GND ==> GND

SDA ==> P21

SCL ==> P22

Ha minden be van kötve akkor tanuljunk egy kicsit az ESP-32 kódkezeléséről. Ezeket a kódokat nem lehet futtatni csak a gépen, ezért fel kell a drivereket töltenünk a mikrokontroller memóriájába és utána futtatni a kódot a mikrokontrolleren. Ezzel kapcsolatban néhány parancs:

- *mpremote ls*  ESP-32-n lévő fájlok mutatása

- *mpremote cp <fájlhoz vezető út> :* Az ESP-32-re egy fájl másolása (fontos a kettőspont, ne hagyd el)

- *mpremote rm <fájlnév>*  Az ESP-32-ről egy fájl letörlése

-*mpremote run <fájlhoz vezető út>* Egy számítógépen lévő fájl futtatása az ESP-32-n (hasznos a gyors teszteléshez)

Szóval egyszerűen feltöltjük a kettő driver-t és utána a „run” paranccsal futtathatjuk a kódot.

Jelen esetben sikeres a teszt.

A képen szöveg, képernyőkép, hold látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.

Most jöhetnek a LoRa modulok. Nem különböznek sokkal a szenzortól, csak a kódjuk bonyolultabb. Itt is kell majd kötögetni és egy új drivert mindkettő mikrokontrollerre tölteni. Ehhez itt van a link: <https://github.com/sergio303/micropython-sx127x/blob/master/sx127x.py>. Ezt rögtön fel is tudjuk tölteni az ESp-32-k memóriájába: *mpremote cp <fájl út>*. Ha ezzel megvagyunk jöhetnek a csatlakozások. A LoRa moduloknak 8 csatlakozása van:

A képen Elektromos vezetékek, Elektrontechnika, elektronika, villamosenergia-ellátás látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.**LoRa Pin ESP32 Pin**

MISO ==> GPIO19

DIO0 ==> GPIO26

SCK ==> GPIO18

MOSI ==> GPIO23

RST ==> GPIO14

NSS ==> GPIO5

GND ==> GND

VCC ==> 3V3

A képen kábel, Elektromos vezetékek, Elektrontechnika, elektronika látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.Miután ezt megtettük futtassunk le egy kicsi kódot, annak érdekében, hogy megbizonyosodjunk arról, hogy érzékeli az ESP32 a LoRa modult. Ez itt található: <https://github.com/Grandturkboy/Kaboom_Cansat_2025/blob/main/Scripts/LoRaFXversionCheck.py>. Ha minden jól be van kötve ez 18-at vagy egy hasonló egész számot fog visszadobni. Ha nullát, akkor nem érzékeli a modult.

Ha ez működik, akkor összeköthetjük a transmitter mikrokontrollert egy LoRá-val és a szenzorral. Ez nagyrészt problémamentes, de az ESP32-nek, amit használunk csak egy 3V3 pin-je van, ezért ezt egy szerelőlap segítségével egyesítettük a 3V3-akat és a GND-okat. Ha ezt elrontjuk vagy a csatlakozás megszűnik miközben áram alatt van a mikrokontroller, lehetséges, hogy a firmware megsérül és ahogyan az első lépésben, ki kell törölnünk a flash memóriát majd vissza kell töltenünk a meg felelő .bin fájl-t és a drivereket.

Ha ezzel megvagyunk megírhatjuk a transmitter kódját, amit ide raktam: <https://github.com/Grandturkboy/Kaboom_Cansat_2025/blob/main/Scripts/Transmitter.py>. Ez a kód inicializálja a LoRa modult és beállítja úgy, hogy közepesen lassú, de nagy hatótávolságú legyen, majd minden másodpercben lekéri a szenzor adatokat, becsomagolja őket egy string-be és elküldi a LoRá-n keresztül. Ezenkívül használja a CRC-t, ami egy kódot csatol minden csomaghoz. Ha ez a kód nem ugyanaz a csomag érkezésénél akkor a receiver tudni fogja, hogy az adatok sérültek és hogy ignorálni kell őket.

Ha ez működik (folyamatosan írja ki a csomagok tartalmát), akkor megcsinálhatjuk a receivert. Ennek a kötése egyszerű, csak egy LoRa modult kell a ESP-32-be kötni és annak a driverjét a memóriába feltölteni. (Itt nem kellenek a szenzordriverek.)

Ehhez sem túl bonyolult a kód: <https://github.com/Grandturkboy/Kaboom_Cansat_2025/blob/main/Scripts/Reciever.py>. Miután ez is inicializálja és bekalibrálja a LoRa modult, folyamatosan elkezd figyelni a beérkező csomagokra és kiírja őket, ha érzékeli.

Ha ezeket a kódokat feltöltjük a megfelelő mikrokontrollerekre úgy, hogy a másolatokat elnevezzük „main.py” -nak, akkor mindig, amikor az ESP-k áramot kapnak, elkezdik őket futtatni. Ezt így tehetjük: *mpremote cp <fájl út> :main.py* .

A képen képernyőkép, minta, szöveg, szövet látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.A teljes hardver tesztet végre is lehet hajtani, csak áramhoz kell juttatni a mikrokontrollereket és nézni, hogy a receiver kapja el a jelet rádión keresztül.

Sikerest teszt.