

Szemkövetés virtuális valóság környezetben

1. Fejlesztés során használt eszközök

Arduino IDE:

A létrehozott szemkövetést egy mikroprocesszor segítségével oldottam meg és ezt szükség volt egy fejlesztői környezetet választani. Nyílt forráskódú szoftver, amely megkönnyíti a kód írását és a mikroprocesszor alaplapjára való feltöltést. Támogatja az általam választott ESP32-CAM programozását.

Python:

A szemkövetés során szükség volt a rögzített videót feldolgozni, ehhez én Python magas szintű programozási nyelvet választottam. Sok előre definiált könyvtárat használ, melyek támogatják felgyorsítják és egyszerűsítik a képfeldolgozási algoritmusokat. Megoldásom során az OpenCV könyvtárat alkalmaztam.

2. Megvalósítás

A szemkövetés egy olyan módszer, amely segít abban, hogy a felhasználó vizuális figyelmét elemezni tudjuk. Az adatok elemzése során meg lehet határozni, hogy a felhasználó adott időpontban épp hova néz, mennyi ideig és milyen utat jár be a szeme. Szoftver használhatóságának mérése során egy hasznos aspektus, mely segít megérteni a felhasználói élményt objektív módon. Virtuális valóság rendszereknél a világ a szem elé kerül ezért a szem vizsgálata rengeteg hasznos adattal szolgál.

Követelmények:

- Hardver:
 - EPS32-CAM
 - OV2640 kameramodul
 - Arduino MEGA
 - Vezetékek
- Szoftver
 - Arduino IDE
 - Python
 - OpenCV könyvtár

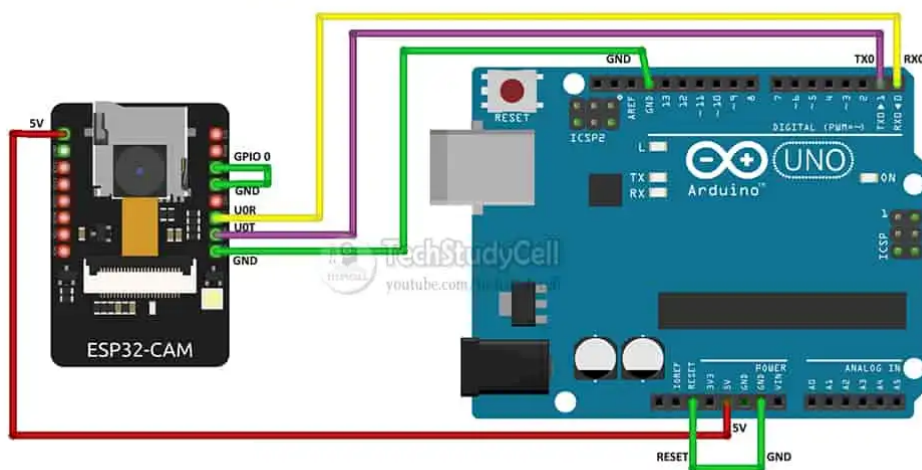
Megvalósítás során videórögzítő eszközként egy ESP32-CAM mikroprocesszor és egy hozzá illeszkedő OV2640 kameramodul szolgált. Mivel a VR szemüveg kis helyet biztosít a szem és a lencsék között ezért esett a választás az ESP32 és a hozzá tartozó kameramodul használatára. Emellett nagy előnyt nyújt, hogy a választott mikroprocesszor rendelkezik beépített WIFI modullal, és egy webszerverre küldi a kamera képét. A mikroprocesszor által létrehozott webszerver Arduino IDE fejlesztőkörnyezetben jött létre. Az ESP32-CAM eszközön nincs USB

port, közvetlenül tehát nem lehet rákötni a számítógépre. Két lehetőség volt a megoldásra: vagy USB-TTL átalakítóval, vagy egy Arduino UNO/MEGA-t. Ebben a megoldásban én egy Arduino MEGA-t használtam a következő bekötéssel, amelyet a 1. táblázat: Bekötési útmutató és 1. ábra: Bekötési rajz mutat.

1. táblázat: Bekötési útmutató

ESP32-CAM	Arduino
GND	GND
5V	5V
U0R	RX
U0T	TX
ESP32-CAM GPIO0 → ESP32-CAM GND	
Arduino RESET → Arduino GND	

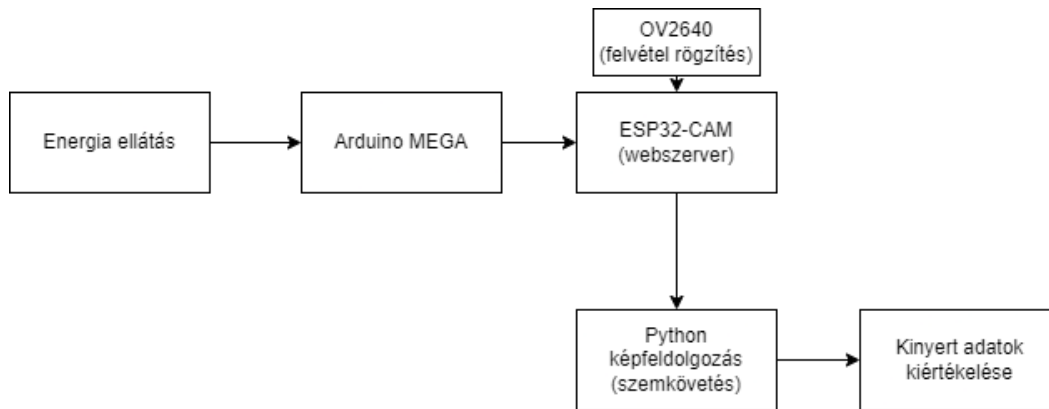
A 1. táblázat: Bekötési útmutató alsó két pontja nélkül nem lehet *flash mode*-ot elérni, ez arra kell, hogy az ESP32-CAM-re rá lehessen tölteni a megírt programot.



1. ábra: Bekötési rajz

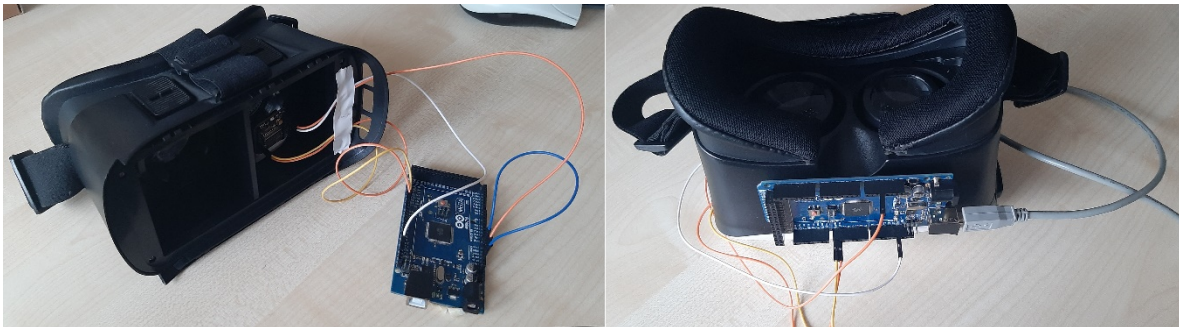
A mikroprocesszor indulásakor elindít egy webszervert az előre konfigurált hálózaton, majd egy IP cím segítségével erre kapcsolódni lehet. A felvétel feldolgozását egy Python program végzi, amely valós időben a webszerverről nyeri a bemenetet 2. ábra: Szemdetektálás.

A 3. ábra: ESP32 rögzítése egy VR szemüvegre látható a mikroprocesszor felszerelése egy virtuális valóság szemüvegre.

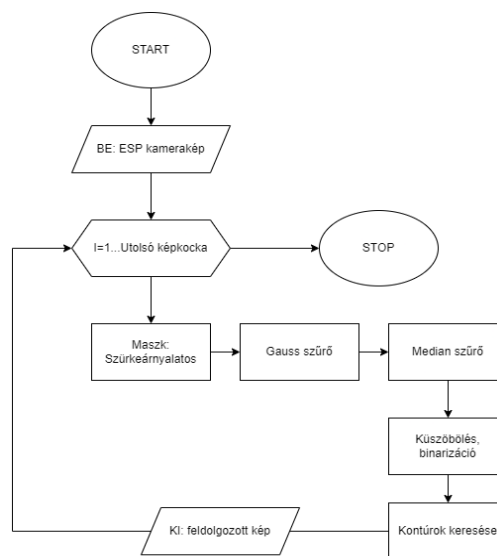


2. ábra: Szemdetektálás

A videót konvertálom szürkeárnyaltossá, majd egy Gauss és medián szűrőt használok a zaj csökkentésére. Az előfeldolgozott képen küszöbölést és binarizálást végzek, és a képen a legsötétebb kontúrokat keresem, mivel ez lesz a pupilla 4. ábra: Python képfeldolgozás.



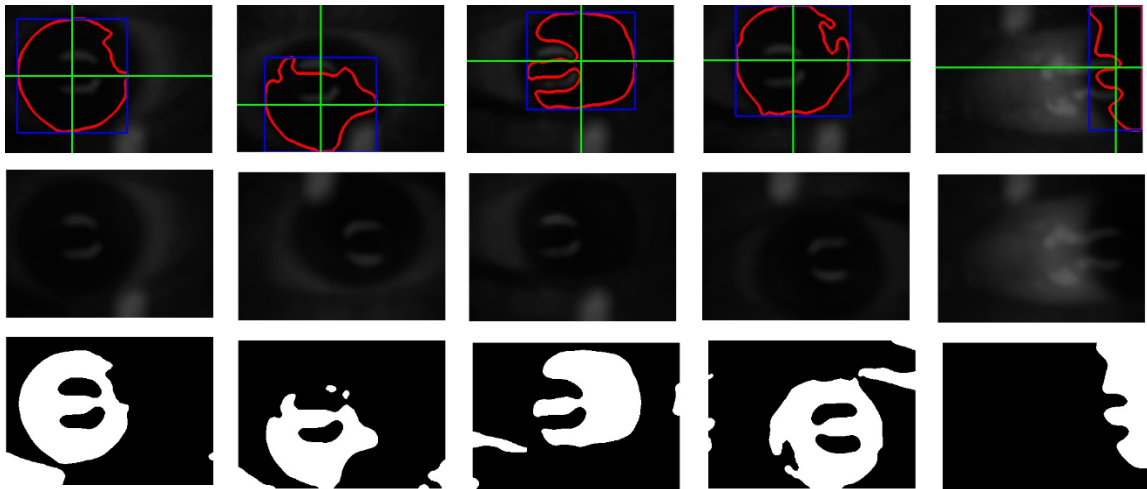
3. ábra: ESP32 rögzítése egy VR szemüvegre



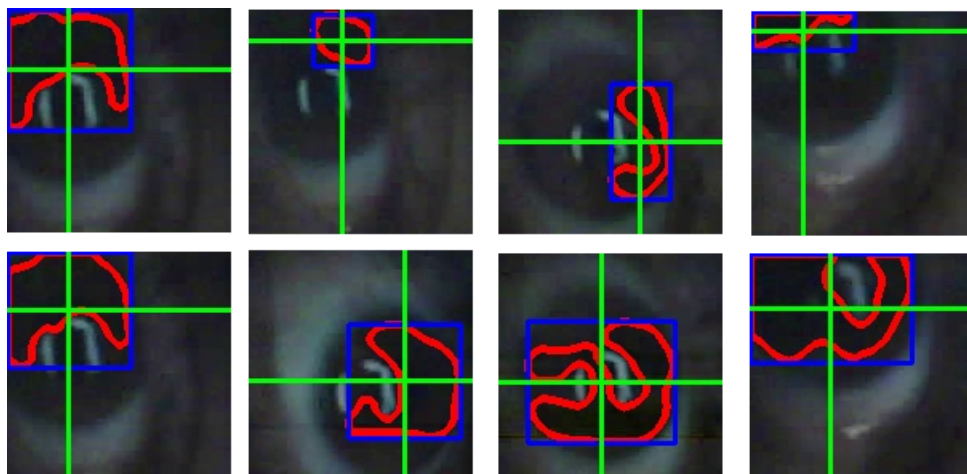
4. ábra: Python képfeldolgozás

3. Szemkövetés tesztelése

Szem detektálása során több nehézségbe is beleütköztem. Első körben a mikroprocesszort és a benne lévő kamerát kellett megfelelő helyre raknom, hogy a kamera pont rálásson a szemre és eközben a berendezés ne takarja el teljesen a látóteret. Szintén nehéz feladat volt kialakítani a képfeldolgozó folyamatot, mivel kevés fény jutott a belső szerkezetbe így a kép sokkal sötétebb. Erre megoldásként arra jutottam, hogy ha jól levágom a kép széleit és a felesleges információkat eltörölöm a képről akkor jobban tudja az algoritmus detektálni a pupillát 5. ábra: Pupilla detektálási fázisok és 6. ábra: Pupilla detektálás másik bementi videón.



5. ábra: Pupilla detektálási fázisok



6. ábra: Pupilla detektálás másik bementi videón