

## Drón - ArduPilot alapú Copter típusú drón kommunikációjának módosítása

- QGroundControl (QGC) szoftver forráskódjának buildelése forrásból QT Creator használatával.
- ArduCopter 4.3.2 forráskódjának letöltése, és buildelése.
- Szimulált drón elindítása.
- Új Mavlink parancs létrehozása, amelyet a QGroundControl küld a drónnak.
- Új Mavlink üzenet létrehozása, amelyet a drón küld a QGroundControlnak, 1 paraméterrel.
- Mavlink headerek generálása C++11-hez.
- QGC használatával a drónra egy tetszőleges mission feltöltése. Mission elindításakor küldje vissza a Mavlink üzenetet a QGC-nek. Az üzenetben a paraméter értéke legyen "12".
- A megérkezett Mavlink üzenetre válaszul küldje a létrehozott Mavlink parancsot a QGC a drónnak.
- A Mavlink parancsra válaszul küldjön egy MAV\_SEVERITY\_ALERT típusú szöveges üzenetet.
- Új paraméter létrehozása „CUSTOM\_PARAM” névvel 0.6-os default értékkel.

A feladatok tetszőleges sorrendben elvégezhetőek, mindegyik külön kerül pontozásra.

Linux operációs rendszer ajánlott a feladat megvalósításához.

Dokumentáció és forráskód, amely alapján reprodukálható a fejlesztési folyamat.

Szükséges kompetencia: C++, Linux

Eredmény: A Mavlink üzenetek átküldésre kerülnek.

## Gépi látás - Kép vagy videó forrásból szegmentálás

Tetszőlegesen kiválasztott képen/videón szegmentálás megvalósítása, Yolo használatával.

A program legyen képes bemeneti fájl alapján meghatározni, hogy képet vagy videót kapott forrásként.

Kiválasztott objektumokra lehessen szegmentálni, a COCO dataset alapján.

A szegmentált objektumok megjelenítése olyan módon, hogy csak a maszkot jeleníti meg a program. Példa: Ember szegmentálás esetén fehér színnel jelölt ember alakot látunk fekete háttér előtt.

A szegmentált objektumok megjelenítése olyan módon, hogy nem csak a maszkot jeleníti meg a program, hanem a kivágott maszkra megjeleníti az objektum valós képét, amelyet a képen/videóban látunk.

Minden esetben, ha nem talál olyan objektumot, amelyre a szegmentálást el tudja végezni, akkor adjon vissza fekete képet.

Minden esetben kimeneti képet (ha kép volt a bemenet) vagy videót (ha videó a bemenet) készítse a program a bemeneti fájl nevével egy „runs” almappa alá.

A program legyen olyan módon paraméterezhető, hogy command line paraméterekként lehessen megadni a futáshoz szükséges paramétereket.

### A szükséges paraméterek listája:

--source: Bemeneti fájl elérési útvonala

--view-img: Futás időben megjeleníti a feldolgozott képet/videót.

--save-output: Mentse el a kimenetet.

--real: Jelenítse meg az objektum valós képét is a maszkon

A feladatok tetszőleges sorrendben elvégezhetőek, mindegyik külön kerül pontozásra.

Dokumentáció és forráskód, amely alapján reprodukálható a fejlesztési folyamat.

Szükséges kompetencia: Python, Linux

Eredmény: A feldolgozott bemenetből kép vagy videó készítése, és bemutatása.

## Webfejlesztés - Real-time GPS nyomkövető dashboard megjelenítő

- Előre meghatározott formátumban, MQTT üzenetekben érkező eszközadatok és GPS koordináták feldolgozása és valós idejű megjelenítése (MQTT brokert biztosítunk).
- A kialakítandó felület egy térkép, amelyen megjelenítésre kerül az adatküldő eszköz aktuális GPS koordináta szerinti helye, tehát automatikusan frissítse a pozíciót a térképen.
- A felületen meg kell jeleníteni az utolsó sikeresen feldolgozott MQTT üzenet időpontját, másodperc pontossággal.
- Az oldalnak jól olvashatónak kell lennie böngészőben PC-n és mobileszközön (telefon és tablet) egyaránt.

### Bónusz

- A felületnek alkalmasnak kell lennie, hogy az utolsó N darab feldolgozott MQTT üzenetből a GPS koordinátákat megjelenítse egyidőben.
- N érték konfigurációból módosítható legyen.
- Az aktuális és a korábbi pozíciók színe legyen eltérő.

Dokumentáció és forráskód, amely alapján reprodukálható a fejlesztési folyamat.

Szükséges kompetencia: Javascript és webfejlesztési ismeretek

Eredmény: A felületen legyen látható az aktuális pozíció.

## AR - Android mobilalkalmazás készítése

Egy telefonos AR applikáció létrehozása, ami négy képernyőből áll.

- Az első scene egy főmenü, ahonnan megnyitható a többi képernyő.
- A második scene legyen egy QR kóddal (vagy akármilyen kép) markeres módszerrel megjelenített 3D objektum (legalább 2 darab).
- A megjelenített 3D objektumok száma min. 2 és max. 9 darab, amely konfigurációból módosítható.
- A harmadik scene-en kerüljön megjelenítésre egy 3D objektum, amely a szokásos gesztusokkal mozgatható, forgatható, valamint mérete változtatható.
- Beállítások menü, amiben egy slider segítségével állítható a 3. scene-ben lévő objektum kezdő mérete. A slider 1-5 skálán legyen mozgatható.

Másik scene-be átlépés és visszalépés után is tartsa meg a slider az értékét.

Az app legyen responsive a különböző kijelzőméretekre.

Dokumentáció és forráskód, amely alapján reprodukálható a fejlesztési folyamat.

Szükséges kompetencia: Unity, Android Studio, C#

Eredmény: Saját telefonon bemutató az AR alkalmazásról.