

Telemetry Szoftver

1. Bevezető

A Quadcopter csapatban felmerült az az igény, hogy a kopterünket valós időben tudjuk követni a térképen. A helymeghatározás mellett az is fontos, hogy a kopter által bejárt útvonalat ki tudjuk rajzolni egy térképre, illetve a könnyebb navigáció miatt célszerű a műhold és egyéb térképajták együttes használata is. A program ezeket a problémákat hivatott megoldani, és még többet is, mert a térkép adatok mellett általános telemetria adatok fogadását és kijelzését is biztosítja, valamint egyéb funkciókkal bővíthető.

2. A szoftverről röviden

A programot C#-ban írtam, Windows Forms segítségével, így egy hagyományos asztali alkalmazásról van szó. Monoval remélhetőleg Linux és OS X platformokon is fut, nem csak Windowson. A program lehetőséget nyújt térkép, élő videó stream és szöveges telemetria adatok GUI-n való megjelenítésére.

3. Kezelőfelület

A kezelőfelület a képen látható módon, panelekből tevődik össze:



A panelek drag'n'drop módszerrel áthelyezhetőek, illetve dokkolhatóak a felületen.

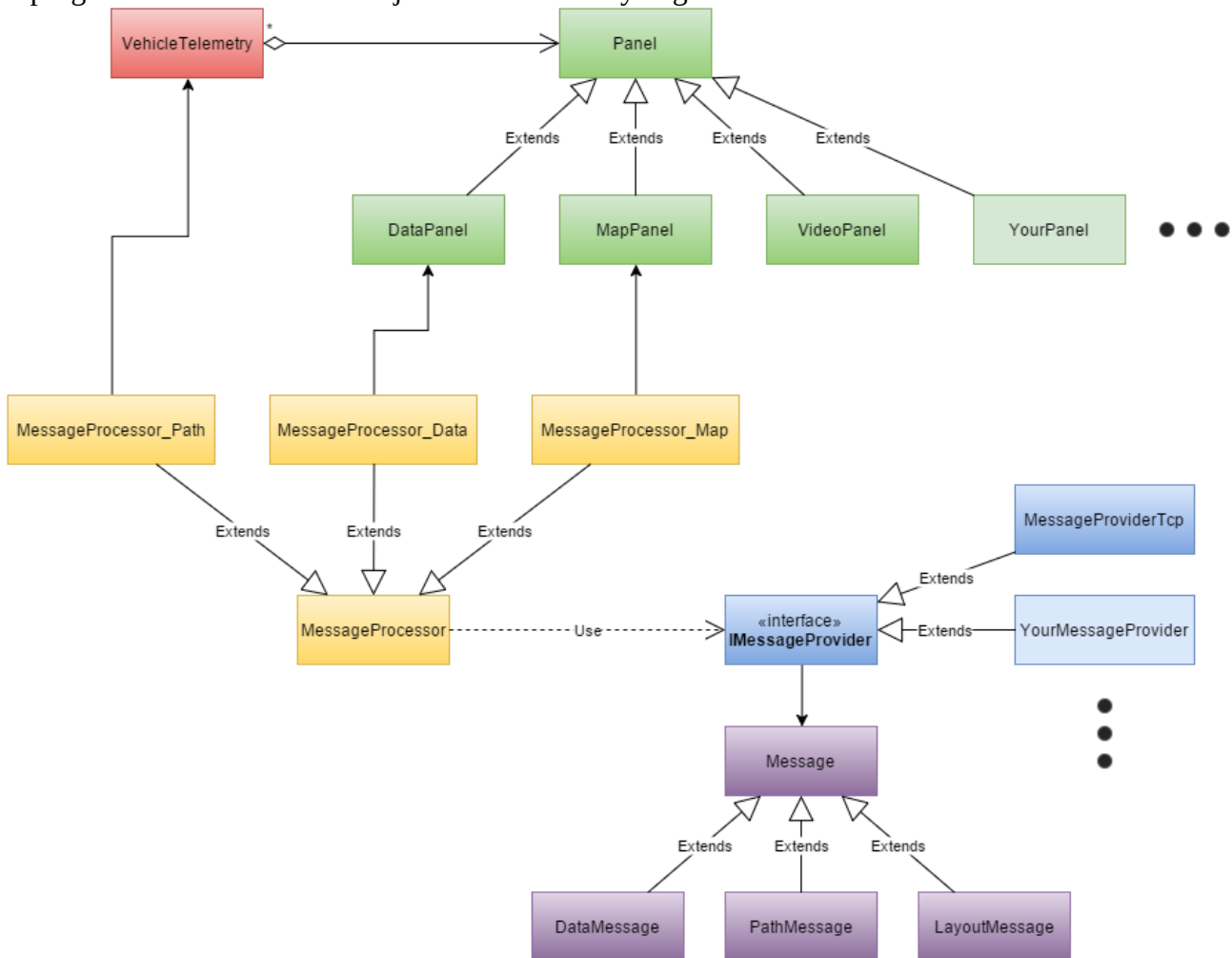
A képen a három alapvető panelt láthatjuk:

- videó (panel 1): a quadcopter kamerájának a felvételét hivatott megjeleníteni
- telemetria (panel 2): valós idejű szöveges információt jelenít meg, például a képen
- térkép (panel 3): követi a koptert, opcionálisan meg tud jeleníteni különböző útvonalakat

A program lehetővé teszi az egyedi panelekkel való bővítést, hogy változatos funkciók megvalósítása is lehetséges legyen.

4. Architektúra

A program vázlatos architektúráját az alábbi osztálydiagram szemlélteti:



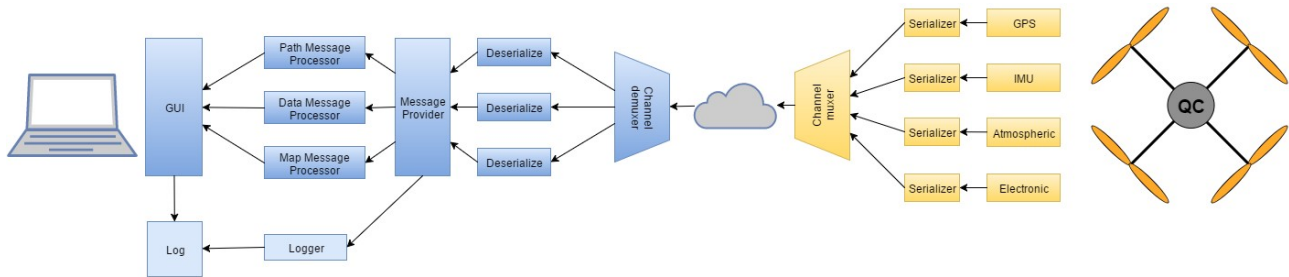
A szoftver jól elkülöníthető részekre bontható:

1. **VehicleTelemetry:** Az egész program magja. A GUI keretrendszert biztosítja, amire az egyes panelek pakolhatók.
2. **Panелеk:** A már ismert három panel mellett implementálhatunk saját, egyedi paneleket.
3. **Üzenetek:** A program a quadcopterrel üzenetek segítségével kommunikál. Az üzenetek telemetria és egyéb meta-adatokat tartalmazhatnak.
4. **Üzenetszolgáltatók:** Az üzeneteket továbbító csatorna létrehozásáért felelősek. Jelenleg egyetlen implementáció van, amely TCP felett üzemel, de bővíthető UDP és rádiós kommunikációra is.
5. **Üzenetfeldolgozók:** Az üzenetszolgáltatókra csatlakozva megkapják, majd feldolgozzák az üzenetet. Az üzenetben lévő adatot lefordítják a kezelőfelület elemeinek megfelelő műveletekre, azaz például a GPS üzeneteknek megfelelően állítják be a térkép pozícióját.

5. Adatáramlás

5.1 Adatforgalom

A következő ábrán nagyító segítségével látható az adatáramlás:



A quadcopter a szenzorok által nyert adatokat feldolgozza, szerializálja, majd elküldi a hálózaton keresztül a telemetria programnak. A program szétválogatja az üzeneteket, majd deszerializálja őket. Az üzenetek az Üzenetszolgáltatón keresztül az Üzenetfeldolgozókhoz kerülnek, amik az adatokat megjelenítik a kezelőfelületen. A rendszer opcionálisan bővíthető egy log rögzítővel is.

5.2 Üzenettípusok

Alapvetően 3 fajta üzenet különböztetünk meg:

- Útvonal üzenet (PathMessage)
- Adat üzenet (DataMessage)
- Elrendezés üzenet (LayoutMessage)

Útvonal üzenet: Az útvonal üzenetek a térképen megjelenítendő útvonalakat írják le. Lehetőség van útvonalak hozzáadására, törlésére, az útvonal pontjainak módosítására.

Az útvonalüzenetek sorosítva így néznek ki:

Type (1 byte)	Action (1 byte)	Path (2 bytes)	Index (4 bytes)	Latitude (4 bytes)	Longitude (4 bytes)	Altitude (4 bytes)
------------------	--------------------	-------------------	--------------------	-----------------------	------------------------	-----------------------

Type, egész: az üzenet típusa, azaz PathMessage

Action, egész: útvonal hozzáadása, útvonal törlése, minden útvonal törlése, pont hozzáfűzése, pont felülírása, útvonal kiürítése

Path, egész: egy szám, ami az útvonalat azonosítja

Index, egész: az útvonalban melyik ponton hajtódik végre az operáció

Lat/Long/Alt, lebegőpontos: a pont GPS koordinátája

Adat üzenet: Az adat üzenet tartalmazzák az átvitt telemetria adatok számértékét.

Sorosítva így néznek ki:

Type (1 byte)	ID (2 bytes)	Dimension (2 bytes)	Data (4 * Dimension bytes)
------------------	-----------------	------------------------	-------------------------------

Type, egész: az üzenet típusa, azaz DataMessage

ID, egész: az adatcsoport azonosítója (lásd: elrendezés üzenet)

Dimension, egész: az adatcsoport értékeinek száma

Data, lebegőpontos: az adatcsoport friss számértékei

Elrendezés üzenet: Amint egy előbbi képen látható, a telemetria adatok olyan kategóriákba szedhetők, mint a GPS vagy motor teljesítmény. Az elrendezés üzenetek ezeket a kategóriákat adják meg. Minden csoporthoz hozzárendelnek egy azonosítót (ID), megadják a csoport és a csoport elemeinek nevét. Természetesen ez leírja a csoport elemeinek számát, azaz dimenzióját is. Az adatüzenetek az ID alapján hivatkoznak egy csoportra, és a csoport elemeinek értékeit tartalmazzák.

Az elrendezés üzenetek sorosított formája elég bonyolult, és gecíálmos vagyok, úgyhogy azt nem írom le.

5.3 Csatornák

A rendszer jelenleg TCP felett küldi az üzeneteket. Amennyiben szükséges, lehet több szolgáltatót implementálni, mint például az UDP, vagy saját rádiós interfész. A TCP sajnos nem hatékony ilyen típusú kommunikációra, ezért elsődlegesen arra használható, hogy egy azonos gépen futó slave program a másik csatornán kapott adatokat TCP-n továbbítsa a telemetria szoftver felé.

6. További munka

Ez így mind szép és jó, azonban a szoftver forráskódján van mit javítani, és több feature nincs még implementálva.

6.1. A kód letisztítása, a framework véglegesítése

A kód egyelőre nagyrészt a „majd később megcsinálom rendesen” fejlesztési elvet tükrözi. Ez azt jelenti, hogy a kivételek és hibák kezelése sok helyen hiányos, az interfészek és implementációk nincsenek megfelelően kialakítva, és sok formai dolog hiányzik. A cél ezeknek a mihamarabbi befoltozása.

6.2 Szükség van egy video stream implementációra

Jelenleg a videó stream csak egy dummy osztály, hogy lehessen a programot egészében tesztelni. A feladat egy modern kodeket, mondjuk H264-et használó megoldás implementálása.

6.3 Hálózati üzenetszolgáltatók

A quadcoptert összekötni a programmal egy alkalmas csatornán, ehhez megírni a szolgáltatót.

6.4 Távirányításhoz szükséges panelek létrehozása

A rendszerhez olyan paneleket is lehet írni, amik lehetőséget adnak a járművek távirányítására. Ezzel az irányítást és a felügyeletet egyetlen program végezné.

7. Egyéb

A források megtalálhatók GitHubon, a következő linken: <https://github.com/petiaccja/QcTracker>. (Mondjuk lehet át kéne nevezni a repot... nevermind). A linken az Executable mappában megtalálhatóak a lefordított fájlok, amiket ki lehet próbálni. Természetesen mindegyikbe tettem legalább három trójait.