

Flight recorder simulator

1 Informacje wstępne

1.1 Wymagania

Do prawidłowego działania programu wymagana jest następująca wersja programu: **Python 3.4.3**. Do uruchomienia konieczne jest również posiadanie pakietu **SQLite3** do obsługi bazy danych. Niezbędne może się okazać również doinstalowanie do naszego Python'a następujących bibliotek: **Matplotlib** - odpowiedzialnej za rysowanie wykresów oraz **Tkinter** - odpowiedzialnej za graficzny interfejs.

1.2 Dane

Niezbędne jest również posiadanie przykładowych danych, które posłużą nam do symulacji lotu. W swoim projekcie wykorzystałem informacje o lotach pochodzące ze strony [FlightAware](#).

Time	Position		Orientation		Groundspeed		Altitude	Reporting Facility	
CET	Latitude	Longitude	Course	Direction	KTS	km/h	feet	Rate	Location/Type
Thu 21:57:00	Left Gate (KRK / EPKK) @ Čtvrtek 21:57:00 CET								Airline
Taxi Time: 1 minutes									
Thu 21:58:01	Departure (KRK / EPKK) @ Čtvrtek 21:58:01 CET								FlightAware ADS-B (OSR / LKMT)
Thu 21:58:01	50.0698	19.7222	259°	West	181	335	2,500	FlightAware ADS-B (OSR / LKMT)	
Thu 21:58:21	50.0685	19.6942	280°	West	203	377	3,000	2,542 FlightAware ADS-B (OSR / LKMT)	
Thu 21:59:00	50.0805	19.6373	289°	West	232	430	5,000	2,836 FlightAware ADS-B (KRK / EPKK)	
Thu 21:59:16	50.0859	19.6109	288°	West	253	468	5,600	2,516 FlightAware ADS-B (KRK / EPKK)	
Thu 21:59:31	50.0913	19.5844	287°	West	264	489	6,300	3,316 FlightAware ADS-B (KRK / EPKK)	
Thu 21:59:54	50.1028	19.5429	297°	West	272	504	7,700	3,158 FlightAware ADS-B (KRK / EPKK)	
Thu 22:00:09	50.1115	19.5156	297°	West	289	536	8,300	2,000 FlightAware ADS-B (KRK / EPKK)	
Thu 22:00:24	50.1202	19.4886	297°	West	309	573	8,700	2,600 FlightAware ADS-B (KRK / EPKK)	
Thu 22:00:39	50.1309	19.4550	297°	West	317	587	9,600	3,830 FlightAware ADS-B (KRK / EPKK)	
Thu 22:01:11	50.1511	19.3903	296°	West	317	587	11,700	3,469 FlightAware ADS-B (KRK / EPKK)	
Thu 22:01:43	50.1710	19.3253	296°	West	326	604	13,300	2,806 FlightAware ADS-B (OSR / LKMT)	

Rysunek 1: Przykład pobranej tabeli zawierającej informacje o locie

Jak widać informacja dotycząca dokładnej daty nie została umieszczona, dlatego ręcznie w pliku zamieniłem dni tygodnia na datę w formacie yyyy-mm-dd co ma kluczowe znaczenie podczas działania projektu. Przykładowe dane umieściłem w plikach z projektem w folderze **Example data**.

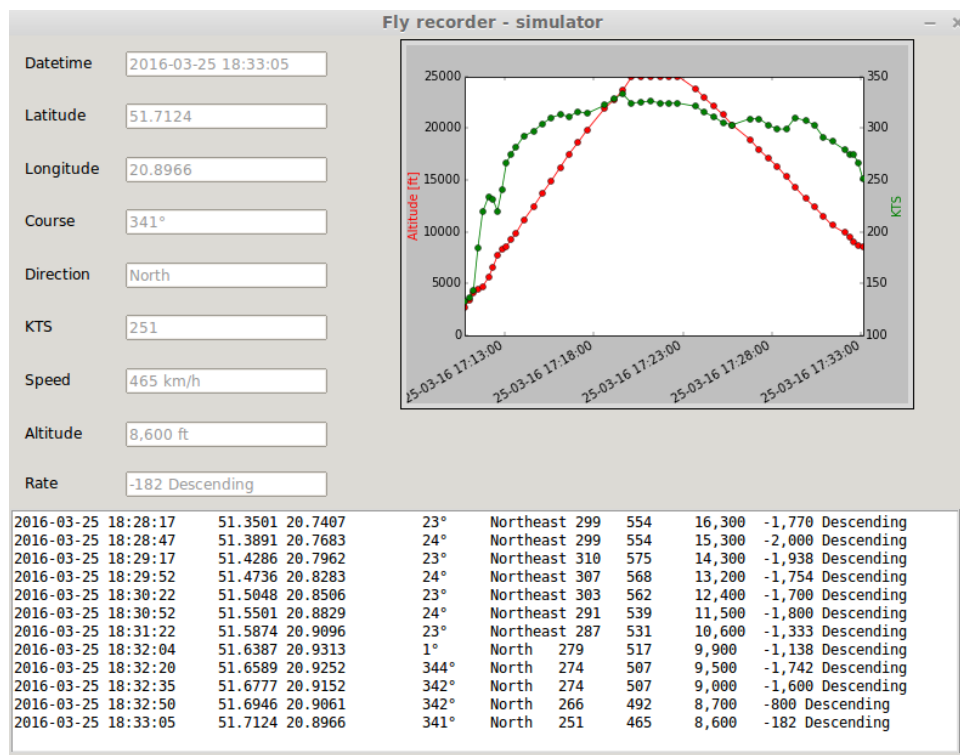
1.3 Uruchomienie

Aby uruchomić program wystarczy skorzystać z komendy **python3 main.py**

2 Opis działania programu

Symulacja lotu

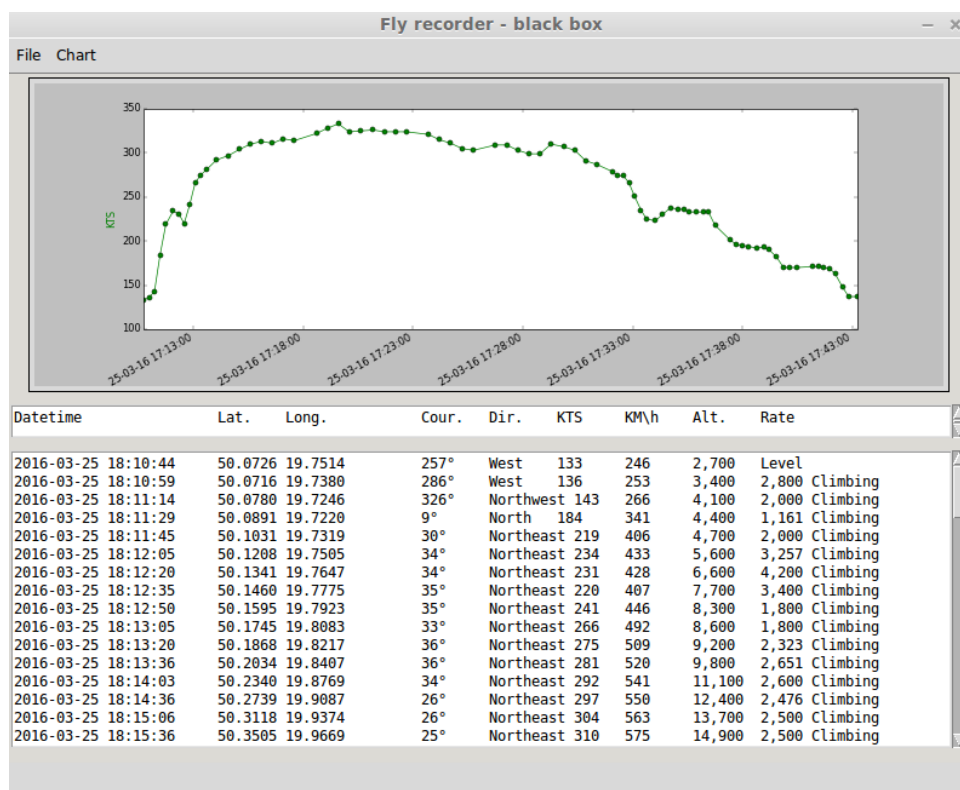
Aby rozpocząć symulację lotu należy wcześniej podać ścieżkę, w której chcemy zapisać naszą bazę [**Save database**] oraz plik z danymi do symulacji [**Import logs**]. Później korzystamy z opcji [**Generate**] i rozpoczyna się symulacja - odtworzenie naszego lotu. Dane zapisywane są w bazie co około 0.5s i na bieżąco generowany jest do nich wykres oraz log.



Rysunek 2: Symulacja lotu

Podglądanie czarnej skrzynki

Aby podglądać naszą **czarną skrzynkę** korzystamy z opcji [Open blackbox]. Następnie wybieramy plik z utworzoną bazą danych [Open database]. Możemy zmienić wyświetlany wykres w [Chart], zapisać go do pliku [Save chart] oraz zapisać nasz log do pliku CSV [Export to CSV].



Rysunek 3: Podglądanie czarnej skrzynki

3 Wykorzystane funkcjonalności

- Moduł **configparser** posłużył mi do utworzenia pliku konfiguracyjnego zawierającego informacje o naszej bazie danych oraz nazwie pliku z naszymi danymi do generowania.
- Moduł **sqlite3** posłużył mi do zaimplementowania funkcjonalności związanych z bazą danych. W tym celu stworzyłem klasę **MyDB** odpowiedzialną za operacje dodawania rekordów. Dla uproszczenia przyjąłem, że wszystkie pola w bazie danych są typu tekstowego i przechowywane są tylko w jednej tabeli. Oczywiście można ulepszyć tą funkcjonalność normalizując naszą bazę danych i stworzyć tylko jedną bazę, która przechowywałaby informacje dotyczące innych pomiarów. Rozwiązanie to wiąże się jednak z pewnym nakładem czasu, który poświęciłem na tworzenie GUI, dlatego na potrzeby projektu przyjąłem, że każdy plik z bazą danych symbolizuje czarną skrzynkę.
- Moduł **matplotlib** do rysowania wykresów oraz **csv** do zapisania naszych danych z logu w pliku.
- Moduł **Tkinter** posłużył mi do stworzenia graficznego interfejsu do naszego programu. Ponieważ to pierwsza moja styczność z tworzeniem GUI wygląd jest dosyć prosty. GUI wykorzystuje również szereg pomocniczych funkcji do interakcji z użytkownikiem (wyświetlania, czyszczenia oraz przetwarzania informacji). Jak już wspomniałem to mój początek w projektowaniu GUI, dlatego nie udało mi się stworzyć odpowiednika klasy zarządzającej interfejsem przez co przeglądając kod wydaje się on na pierwszy rzut oka chaotyczny.

4 Dokumentacja

Opis wszystkich klas, funkcji oraz zmiennych znajduje się w plikach o rozszerzeniu **.py**.