

Informacje ogólne

- Zadania przedstawione w niniejszej instrukcji należy wykonać **samodzielnie** na systemie operacyjnym Ubuntu 18.04.2 (x64). Hasło użytkownika to tello.
- Wykonanie każdego zadania należy **udokumentować zrzutami ekranu obejmującymi cały ekran monitora**. Niezastosowanie się do powyższego będzie równoznaczne z pominięciem zadania przy ocenie.
- Po zakończeniu pracy należy usunąć pliki własne.

Zagadnienia

- środowisko ROS 2 Eloquent Elusor (ang. *Robot Operating System*),
- terminologia i podstawowe komendy,
- instalacja i uruchomienie biblioteki DJITelloPy,
- podstawy komunikacji z dronem DJI Ryze Tello.

Materiały pomocnicze

- <https://docs.python.org/3.6/>
- <https://www.jetbrains.com/pycharm/download/#section=linux>
- <https://github.com/damiafuentes/DJITelloPy>

Wprowadzenie

Sterowanie dronem DJI Ryze Tello z poziomu środowiska ROS charakteryzuje się znacznymi możliwościami interakcji oraz umożliwia symulację jednostki latającej, bez konieczności narażanie rzeczywistego drona na ewentualne uszkodzenia. Niewątpliwą wadą takiego rozwiązania jest przede wszystkim wysoka złożoność kodu, a tym samym znaczny nakład prac, które należy wykonać, aby projekt skonfigurować i uruchomić. Alternatywnym, a w dodatku w pełni multiplatformowym rozwiązaniem, jest biblioteka DJITelloPy, która pozwala na bezpośrednią komunikację z dronem, wprost z kodu napisanego w języku Python.

Implementacja biblioteki została przygotowana w oparciu o oficjalną dokumentację firmy DJI. Pozwala ona na wykorzystanie wysokopoziomowego API do akwizycji danych oraz do sterowania dronem. Możliwe jest pobranie danych z IMU (ang. *Inertial Measurement Unit*), danych z kamery, kontrolowanie poziomu naładowania baterii, czy odczytanie danych z sensorów peryferyjnych, takich jak termometr, czy barometr. Nie zabrakło także zbioru komend umożliwiających start, lot, czy lądowanie. Pełną dokumentację biblioteki można znaleźć na stronie autorów¹.

Instalacja środowiska

Pracę należy rozpocząć od aktualizacji instalatora pakietów pip. W tym celu konieczne jest wywołanie terminala i wpisanie komendy:

```
| pip3 install --upgrade pip
```

Wymagane będzie połączenie z Internetem. Po zakończeniu aktualizacji programu, możliwa stanie się instalacja właściwej biblioteki DJITelloPy. Można tego dokonać za pomocą polecenia:

```
| pip3 install djitellopy
```

¹ <https://djitellopy.readthedocs.io/en/latest/tello/>

Weryfikacja poprawności instalacji polega przede wszystkim na uruchomieniu przykładowego kodu, zamieszczonego w sekcji „Przykład komunikacji z dronem”. W przypadku pojawienia się komunikatu błędu, konieczne będzie sprawdzenie, czy wybrany został właściwy interpreter języka Python, czy dron został włączony oraz czy zostało z nim nawiązane połączenie bezprzewodowe (dron-komputer).

Podstawowe metody biblioteki DJITelloPy

Biblioteka DJITelloPy oferuje rozbudowany zestaw metod, które można wykorzystać do sterowania jednostką latającą. Część z nich wymaga dedykowanego joysticka, który umożliwia m.in. pomiar pokonanej odległości i jednocześnie określenie pozycji drona.

Najważniejsze z metod zostały zebrane poniżej i podzielone na dwie grupy – metod sterujących, pozwalających na kontrolę robota latającego oraz metod pozwalających na akwizycję danych z drona. Kompletną listę wraz ze szczegółowym opisem można znaleźć w dokumentacji.

Tabela 1 – zbiór podstawowych metod sterujących

Metoda	Opis
connect()	Nawiązuje połączenie z dronem.
end()	Zamyka połączenie z dronem i niszczy instancję klasy.
takeoff()	Wysyła komendę startu (wzniesienie się na ustaloną wysokość).
land()	Wysyła komendę lądowania.
emergency()	Wysyła komendę lądowania awaryjnego (odłącza napędy).
streamon()	Włącza możliwość transmisji danych wizualnych z kamery pokładowej.
streamoff()	Wyłącza możliwość transmisji danych wizualnych.
send_rc_control(Y_vel, X_vel, Z_vel, YAW_vel),	Wysyła komendę sterującą. <ul style="list-style-type: none">• Y_vel to prędkość w osi Y: <-100; 100> (lewo; prawo),• X_vel to prędkość w osi X: <-100; 100> (tył; przód),• Z_vel to prędkość w osi Z: <-100; 100> (dół; przód),• YAW_vel to prędkość obrotu wokół kąta Yaw: <-100; 100> (CCW; CW).

Tabela 2 – zbiór podstawowych metod pozwalających na pozyskanie danych

Metoda	Opis
get_current_state()	Pobranie wszystkich danych pokładowych (oprócz obrazu) w formie słownika.
get_state_field(key)	Pobranie konkretnych danych pokładowych (key – nazwa pola do pobrania).
get_acceleration_x()	Odczytanie danych z akcelerometru (oś X).
get_acceleration_y()	Odczytanie danych z akcelerometru (oś Y).
get_acceleration_z()	Odczytanie danych z akcelerometru (oś Z).
get_speed_x()	Pobranie wartości prędkości dla osi X.
get_speed_y()	Pobranie wartości prędkości dla osi Y.
get_speed_z()	Pobranie wartości prędkości dla osi Z.
get_pitch()	Odczytanie wartości kąta obrotu Pitch.
get_roll()	Odczytanie wartości kąta obrotu Roll.
get_yaw()	Odczytanie wartości kąta obrotu Yaw.
get_barometer()	Pobranie danych z barometru.
get_height()	Pobranie danych odnoszących się do wysokości drona.

Metoda	Opis
<code>get_flight_time()</code>	Pobranie informacji o czasie lotu.
<code>get_battery()</code>	Odczytanie poziomu naładowania baterii.
<code>get_temperature()</code>	Odczytanie temperatury.
<code>get_frame_read()</code>	Pobranie uchwytu do obrazu z kamery pokładowej.

Przykład komunikacji z dronem

Po instalacji środowiska konieczne jest sprawdzenie, czy została ona ukończona poprawnie. Do tego celu można skorzystać z prostego kodu zamieszczonego poniżej. Skrypt działa w następujący sposób:

1. tworzy instancję klasy Tello na potrzeby dalszej komunikacji z dronem,
2. podejmuje próbę nawiązania połączenia z dronem,
3. pobiera i wyświetla informację o poziomie naładowania baterii drona,
4. zamyka połączenie i niszczy instancję klasy.

```
from djitellopy import Tello

drone = Tello() # tworzy instancję klasy Tello
drone.connect() # nawiązuje połączenie z dronem
print(drone.get_battery()) # pobiera poziom naładowania baterii drona
drone.end() # zamyka połączenie, niszczy instancję klasy
```

Uwagi

- Każda próba uruchomienia skryptu mogąca skutkować nawiązaniem połączenia z dronem musi zostać uprzednio zgłoszona prowadzącemu i uzyskać jego zgodę.
- Niedopuszczalne jest podjęcie próby startu drona bez wyraźnej zgody prowadzącego.
- Każda osoba przebywająca w laboratorium powinna posiadać okulary ochronne i z nich właściwie korzystać.
- W danym momencie w powietrzu może znajdować się tylko jedna jednostka latająca.
- Niezastosowanie się do powyższego i/lub poleceń prowadzącego będzie skutkować niezaliczeniem ćwiczenia i koniecznością opuszczenia sali laboratoryjnej.

Zadania szczegółowe

1. Zainstaluj bibliotekę DJITelloPy i przetestuj poprawność jej działania. Sprawdź, czy skrypt jest poprawnie interpretowany (czy podejmowane są próby połączenia z dronem).
2. Zapoznaj się z komendami zawartymi w *tabeli 1* oraz *tabeli 2*.
3. Skopiuj przykładowy kod i przeanalizuj jego strukturę. Poproś prowadzącego o udostępnienie drona oraz przetestuj poprawność działania skryptu. Czy wynik jego działania jest zgodny z przewidywaniami? Uzasadnij.
4. Przygotuj kod, który pozwoli na cykliczne odczytywanie danych z akcelerometru jednostki (10 Hz). Uruchom skrypt i przetestuj jego działanie. Zanotuj zakresy wartości, które są zwracane przez dedykowane metody.
5. Rozszerz kod o możliwość cyklicznego odczytywania danych kątowych (roll, pitch, yaw). Przetestuj jego działanie. Zanotuj zakresy wartości, które są zwracane przez dedykowane metody biblioteki DJITelloPy.
6. Zdefiniuj funkcję pozwalającą na wizualizowanie przebiegów czasowych dla poszczególnych kątów z punktu 5 w czasie rzeczywistym. Wykorzystaj do tego celu bibliotekę `matplotlib` oraz przykładowy kod, zamieszczony na platformie *eKursy*. Każda krzywa powinna zostać wyróżniona innym stylem i/lub kolorem.
7. Zdefiniuj funkcję pozwalającą na zapis danych pokładowych do pliku w formacie CSV. Dane te powinny posiadać nagłówki (nazwy pól) oraz składać się z takiego zbioru pól jak:
 - dane z akcelerometru (X, Y, Z),
 - prędkości (X, Y, Z),
 - dane kątowe (roll, pitch, yaw),
 - czas lotu,
 - poziom naładowania baterii.Dane rejestruj z częstotliwością nie niższą niż 10 Hz. Przetestuj działanie całego skryptu i zweryfikuj poprawność zapisanych danych.