

KIERUNEK: Automatyka i Robotyka (AIR)

PRACA DYPLOMOWA
INŻYNIERSKA

Tytuł pracy:
Sterowanie dronem za pomocą gestów dłoni
z wykorzystaniem przetwarzania obrazów

AUTOR:
Maciej Kaniewski

PROMOTOR:
dr inż. Wojciech Dowski

*Niniejszą pracę dedykuję moim rodzicom Justynie i Michałowi
w podziękowaniach za umożliwienie mi zdobywania wiedzy
i poszerzania horyzontów.*

STRESZCZENIE

Praca przedstawia projekt aplikacji, która umożliwi użytkownikowi sterowanie dronem DJI Ryze Tello za pomocą gestów dłoni. Rozpoznawanie gestów zostało zrealizowane poprzez przetwarzanie obrazu z wbudowanej kamery drona lub kamery komputera. Wykrycie poszczególnych gestów skutkuje wykonaniem przez drona odpowiednich akcji ruchu. Aplikacja udostępnia również możliwość ręcznego sterowania dronem oraz odczyt jego wybranych parametrów.

SUMMARY

The paper presents an application project that allows the user to control the DJI Ryze Tello drone using hand gestures. Gesture recognition was implemented by processing the image from the drone's built-in camera or computer camera. Detection of individual gestures results in the drone performing appropriate movement actions. The application also provides the ability to manually control the drone and read its selected parameters.

Słowa kluczowe: dron, DJI Ryze Tello, robot powietrzny, sterowanie, gesty dłoni, MediaPipe, uczenie maszynowe, przetwarzanie obrazów, interfejs użytkownika, aplikacja

Keywords: drone, DJI Ryze Tello, aerial robot, controls, hand gestures, MediaPipe, machine learning, image processing, user interface, application

Spis treści

1	Wstęp	3
1.1	Teza	5
2	Dron czterowirnikowy	7
2.1	Mechanika lotu	9
2.2	Konfiguracja sprzętowa	12
3	DJI Ryze Tello	17
3.1	Sterowanie	22
3.2	Software Development Kit	24
3.3	Biblioteka DJITelloPy	25
4	Projekt aplikacji	27
4.1	Rozpoznawanie gestów dłoni	29
4.2	Dostępne funkcjonalności	31
4.3	Architektura systemu	36
5	Podsumowanie	39
	Załącznik A	40
	Bibilografia	41

Rozdział 1

Wstęp

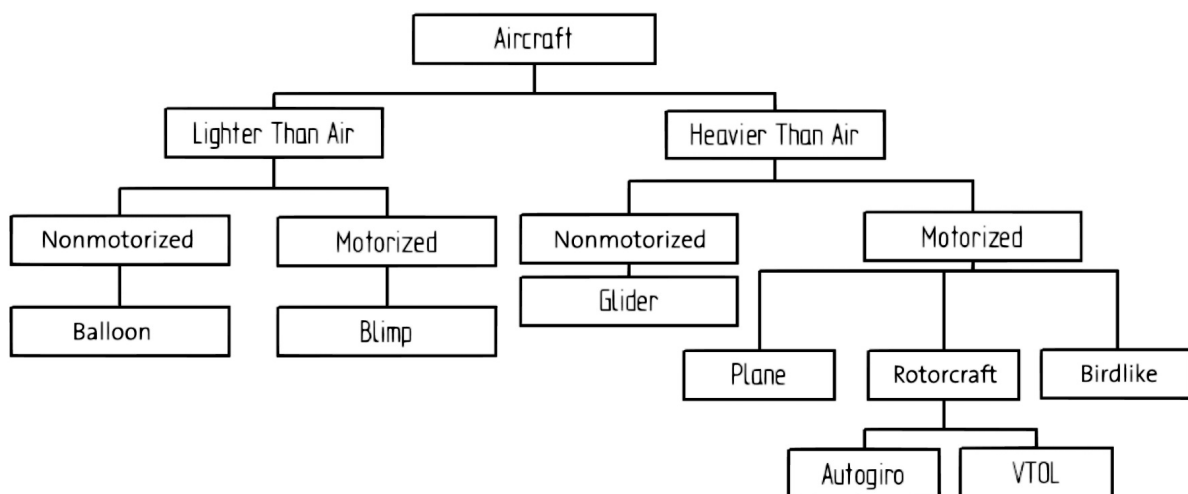
Jeszcze do niedawna wizja technologicznie zaawansowanych urządzeń, które są w stanie automatycznie i z pewną powtarzalnością wykonywać wybrane zadania w sposób bardzo zbliżony do człowieka, a nawet współistnieć z nim czyniąc jego codzienne życie łatwiejszym, była możliwa do zrealizowania jedynie w grach komputerowych, filmach lub książkach wywodzących się z gatunku fantastyki naukowej (z ang. *science fiction*). W dzisiejszych czasach wizja ta stała się już jak najbardziej rzeczywistością, która nawet stopniowo zaczyna przerastać najśmielsze oczekiwania autorów dzieł należących do wspomnianego wyżej gatunku. Za sprawą wielu lat doświadczenia, dostępnych środków oraz prężnie rozwijających się technologii w zakresie mechaniki, elektroniki i oprogramowania człowiek na co dzień może cieszyć się z usług i możliwości, które zapewniają mu te technologicznie zaawansowane urządzenia – roboty.

Ze względu na dużą różnorodność wśród robotów trudno o jednoznaczną terminologię w ramach nazewnictwa poszczególnych rodzajów tych maszyn. Jedną z propozycji jest klasyfikacja robotów na podstawie środowiska, w którym mogą one realizować powierzone im zadania. Tym sposobem można zdefiniować podział na roboty przemysłowe, usługowe i terenowe [53]. Roboty przemysłowe charakteryzują się wykonywaniem zadań we wcześniej zdefiniowanej przestrzeni roboczej, są na stałe usytuowane w obrębie danego stanowiska. Do robotów przemysłowych należą wszelkiego rodzaju manipulatory umożliwiające wykonywanie złożonych procesów technologicznych takich jak spawanie, zgrzewanie oraz montaż i pozycjonowanie elementów o masie od kilku do nawet kilkuset kilogramów. Roboty usługowe poruszają się w przestrzeniach dostosowanych do potrzeb człowieka, ale niedostosowanych do potrzeb robota. Do takich przestrzeni można zakwalifikować budynki mieszkalne, biura, sklepy lub restauracje. Wśród robotów usługowych wyróżnia się roboty osobiste, które wykonują pracę na rzecz człowieka w jego miejscu zamieszkania, czyli na przykład odkurzanie i mycie powierzchni płaskich oraz roboty profesjonalne, które świadczą usługi w miejscach publicznych poprzez dostarczanie zamówień do stolików klientów w lokalach gastronomicznych lub eksponowanie towaru w większych marketach. Roboty terenowe realizują powierzone im zadania w naturalnym środowisku, nieprzystosowanym w żaden sposób do ich budowy, czy też zamierzonych funkcjonalności. Są to na przykład pola, lasy, przestrzeń powietrzna, podmorska [53], a nawet obiekty astronomiczne, czyli na przykład Księżyc lub Mars. Do takich robotów należą wszelkiego rodzaju pojazdy, łaziki oraz zarówno platformy podwodne jak i powietrzne. Roboty usługowe i terenowe można opisać mianem robotów mobilnych.

Roboty mobilne niwelują główną wadę robotów przemysłowych, czyli brak możliwości przemieszczania się. Dzięki zastosowaniu odpowiedniego sposobu lokomocji można zaprojektować robota, który będzie realizował swoje zadania w pełnym zakresie środo-

wiska, do którego zostanie oddelegowany. Do sposobów lokomocji zalicza się chodzenie, skakanie, bieganie, ślizganie, pływanie, latanie jak i również toczenie się [46]. Większość z tych mechanizmów została zaadaptowana na podstawie ich biologicznych odpowiedników. W robotyce można spotkać konstrukcje realizujące poszczególne sposoby przemieszczania się, jednak niektóre z nich cieszą się większą popularnością na tle pozostałych. Są to chodzenie w robotach kroczących, toczenie się w robotach kołowych oraz latanie w robotach powietrznych. Każdy z tych trzech rodzajów robotów mobilnych charakteryzuje się indywidualnym zestawem wad i zalet, który jest czynnikiem decydującym jeśli chodzi o wybór danego rozwiązania. W przypadku robotów kroczących główną ich zaletą jest manewrowość i zdolność adaptacji do poruszania się w trudnym terenie [46]. W takich robotach do realizacji ruchu niezbędny jest jedynie zbiór punktów, w których mogą zostać usytuowane jego kończyny. Dodatkowo roboty te mogą wykorzystać swoje narządy ruchu do manipulowania otaczającymi je przedmiotami. Niestety w celu osiągnięcia nadmienionych funkcjonalności takie urządzenia muszą charakteryzować się bardzo dużą złożonością mechaniczną, co jest nie lada wyzwaniem już na samym etapie projektowym. Roboty kołowe rozwiązują niektóre z wad robotów kroczących mianowicie problem z zachowaniem równowagi oraz efektywnym poruszaniem się po płaskich powierzchniach. Zawdzięczają to zastosowaniu trzech lub więcej liczby kół, które to same w sobie gwarantują zachowanie balansu takiej platformy oraz idealnie nadają się do realizacji ruchu po równym terenie. Jednakże obecność kół przekłada się na mniejszą manewrowość niż w przypadku robotów posiadających nogi. W związku z tym wśród robotów można spotkać się z konstrukcjami wprowadzającymi fuzje dwóch wcześniej opisanych rodzajów platform robotycznych. W takich urządzeniach następuje wówczas połączenie elementów kroczących z kołami. Ostatnim z wyszczególnionych rodzajów robotów mobilnych są roboty powietrzne. Roboty te umożliwiają realizację ruchu w trójwymiarowej przestrzeni, która oferuje więcej możliwości pod względem sterowania niż powierzchnie, po których poruszają się roboty kroczące i kołowe. Niemniej jednak środowisko powietrze stanowi większe wyzwanie od nierównego i zróżnicowanego terenu, co przekłada się na jeszcze bardziej rygorystyczne wymagania pod względem projektowania takich urządzeń.

Roboty powietrzne w znacznym stopniu wywodzą się od swoich większych odpowiedników mianowicie statków powietrznych. W przypadku statków powietrznych można dokonać ich ogólnej klasyfikacji ze względu na sposób realizacji lotu oraz rodzaju zastosowanego napędu. Taki podział został zaprezentowany na rys. 1.1. W pewnym zakresie da się



Rysunek 1.1 Klasyfikacja statków powietrznych [46]

zastosować to uporządkowanie w przypadku robotów powietrznych. Większość z nich można zakwalifikować do kategorii cięższe od powietrza (z ang. *heavier than air*), a kolejno do zmotoryzowane (z ang. *motorized*). Następnie wyróżnia się roboty o konstrukcjach tożsamyh z samolotami (z ang. *plane*), wirnikowcami (z ang. *rotorcraft*) lub ich biologicznymi, w tym przypadku ptasimi, odpowiednikami (z ang. *birdlike*). Roboty powietrzne wzorujące się na samolotach, czyli posiadające powierzchnię nośną, charakteryzują się względnie niewielką manewrowością oraz nie są dobrze przystosowane do wykonywania lotu stacjonarnego, czyli unoszenia się w konkretnym miejscu w przestrzeni [46]. Te problemy rozwiązują wirnikowce. Wśród nich znajdują się urządzenia zrealizowane na wzór helikopterów posiadające pojedynczy wirnik, jednak bardziej popularne są te z czterema wirnikami nazywane kwadrokopterami (z ang. *quadcopter*). Określane są one również mianem platform pionowego startu i lądowania (z ang. *vertical take-off and landing, VTOL*). Urządzenia o takiej konstrukcji wyróżniają się na tle pozostałych. Ich zdolność do realizacji lotu w wielu płaszczyznach, gwałtownej zmiany kierunku ruchu oraz możliwość lotu stacjonarnego sprawiła, że właśnie wirnikowce, a w szczególności kwadrokopetery, znalazły swoje zastosowanie w robotach powietrznych powszechnie nazywanymi dronami.

W kwestii sterowania dronami można wyróżnić poszczególne technologie, które umożliwiają komunikację z takim urządzeniem i tym samym pozwalają użytkownikowi na kontrolę nad ich położeniem. Jedną z nich jest analogowa komunikacja radiowa przenosząca sygnały o częstotliwości 27 MHz i 35 MHz [5]. Obecnie odchodzi się od tego rodzaju wymiany informacji i w zamian tego stosuje się sygnały cyfrowe. Wśród nich jest bezprzewodowy protokół komunikacyjny Wi-Fi zgodny ze standardem IEEE 802.11 [45], który operuje na częstotliwościach 2.4 GHz i 5.8 GHz. Swoje zastosowanie w tej przestrzeni znajdują również sieci komórkowe, które gwarantują niskie opóźnienia i szybki transfer danych [45][52]. Wraz z rozwojem Internetu Rzeczy (z ang. *Internet of Things, IoT*) na popularności zyskał protokół komunikacyjny dalekiego zasięgu LoRa (z ang. *long range*), który ze względu na niski pobór energii bardzo dobrze sprawdza się w robotach powietrznych. Częstotliwości na jakich operuje LoRa zależy od regionu, odpowiednio jest to 433 MHz w Azji, 868 MHz w Europie i 915 MHz w Ameryce Północnej [54]. Roboty mogą również wykorzystywać punkty pośrednie GCP (z ang. *ground control points*) do realizacji powierzonych im zadań w terenie, nad którym operują [43].

Wraz ze wzrostem popularności dronów pojawiały się nowe sposoby kontrolowania tych urządzeń. Wśród nich można znaleźć sterowanie lotem robota za pośrednictwem rękawiczki sensorycznej [37]. Takie rozwiązanie wykorzystuje czujniki zgięcia, których rezystancja zmienia się wraz z ich odkształceniem, co potem odpowiednio przekłada się na wcześniej zdefiniowane akcje ruchu drona. Kolejną metodą sterowania jest analiza postawy ciała człowieka z obrazu otrzymywanego za pośrednictwem kamery. Wykryte pozy przekładają się na realizację ruchu robota w określonych kierunkach [21]. Wzorując się wymienionych rozwiązaniach celem niniejszej pracy inżynierskiej było stworzenie aplikacji, która umożliwiałaby sterowanie dronem za pomocą gestów dłoni otrzymywanych z wbudowanej kamery robota. Aplikacja udostępnia również panel sterowania ręcznego oraz odczytuje i wyświetla istotne informacje o statusie drona.

1.1 Teza

Możliwe jest sterowanie lotem drona za pomocą gestów dłoni z wykorzystaniem przetwarzania obrazów przy zastosowaniu technik uczenia maszynowego.

Rozdział 2

Dron czterowirnikowy

W poniższym rozdziale dokonano charakterystyki wariantu drona, który został wykorzystany do realizacji niniejszej pracy inżynierskiej. Rozdział zawiera informacje dotyczące klasyfikacji statku bezzałogowego, jego sposobu poruszania się w powietrzu, komponentów wchodzących w skład jego budowy oraz obszarów zastosowań.

Drony można zdefiniować jako roboty latające, do których należą bezzałogowe statki powietrzne (z ang. *Unmanned Aerial Vehicle*, UAV), które są w stanie pokonywać dystanse rzędu tysięcy kilometrów jak i również jako ich mniejsze odpowiedniki przeznaczone do operowania w zamkniętych przestrzeniach [25]. Drony są statkami, które nie są przeznaczone do transportowania człowieka kontrolującego ich lot. Ich lot jest kontrolowany za pośrednictwem łączności bezprzewodowej lub w sposób autonomiczny. Wszelkiego rodzaju pociski manewrujące, pociski artyleryjskie, torpedy, miny i satelity nie są uznawane za drony [25]. Obecnie nazwa *dron* jest powszechnie używana w kontekście wszelkiego rodzaju robotów latających. Wynika to ze wzrostu zainteresowania takimi rozwiązaniami, ale przede wszystkim z ich komercjalizacji. Wśród nich znajdują się również opisywane czterowirnikowce, które cieszą się największą popularnością i rozpoznawalnością na tle pozostałych rodzajów platform powietrznych wykorzystywanych w dronach.

Podobnie jak w przypadku ogólnej klasyfikacji robotów, w kwestii robotów powietrznych, a konkretnie dronów, również trudno dokonać ich jednoznacznego podziału. Istnieje klasyfikacja dronów za względu na ich parametry, które bezpośrednio przekładają się na wydajność takich urządzeń. Do parametrów zalicza się wagę, rozpiętość skrzydeł, zasięg, maksymalną wysokość, prędkość, wytrzymałość i koszty produkcji [25]. Biorąc pod uwagę wymienione wcześniej kryteria dron czterowirnikowy wykorzystany w ramach tejże pracy, który został szczegółowo opisany w rozdziale 3, został zakwalifikowany do pojazdów skali mikro (z ang. *Micro Air Vehicle*, MAV).

Drony zaliczające się do tej klasy charakteryzują się wagą w przedziale od 50 g do 2 kg. W przypadku urządzeń posiadających skrzydła ich rozpiętość waha się od 15 cm do 1 m [25]. Drony wyposażone w wirniki, w tym czterowirnikowce, dzięki swoim niewielkim rozmiarom świetnie sprawdzają się w zamkniętych przestrzeniach. Dodatkowo odznaczają się wysoką manewrowością, co pozwala im na wykonywanie zaawansowanych sekwencji ruchu. Ze względu na nadmienione parametry oraz dużą dostępność takich urządzeń na rynku, dron czterowirnikowy idealnie nadaje się do realizacji własnych projektów oraz umożliwia wstępne zapoznanie się z tematem powietrznych robotów mobilnych.

Ze względu na swoje możliwości drony czterowirnikowe znalazły swoje zastosowanie w wielu dziedzinach życia. Te latające roboty wykorzystuje się w fotografii do realizacji materiałów reklamowych dla przedsiębiorstw jak i nagrywania dynamicznych ujęć podczas zawodów sportowych lub produkcji filmowych. Pokaz możliwości technicznych drona

jak i umiejętności pilota można doskonale zaobserwować na materiale nagrany z miejskiego wyścigu kolarskiego organizowanego przez Redbulla [22]. Film został wykonany robotem FPV (z ang. *first-person view*), który umożliwia pilotowi obserwację trasy z perspektywy urządzenia. Firmy kurierskie takie jak Amazon, UPS lub DHL używają dronów do transportu niewielkich paczek zawierających zakupy, leki lub inne przedmioty [14]. Taka usługa jest realizowana na małym dystansie. Na rys. 2.1 przedstawiono drona Mattnet M2 wykorzystywanego przez dostawcę UPS. Czterowirnikowce stosuje się również w agrokulturze, gdzie ich zadaniami są między innymi podlewanie roślin, zbieranie danych o ich stanie zdrowia oraz mapowanie powierzchni. Dzięki możliwości dotarcia w miejsca trudno dostępne dla człowieka drony są używane do przeprowadzania inspekcji różnego rodzaju konstrukcji zarówno podczas ich budowy jak i po jej zakończeniu. Drony wyposażone w kamerę termowizyjną są bardzo istotne podczas klęsk żywiołowych lub innych niebezpiecznych okoliczności, gdzie w trudnym terenie należy odnaleźć ocalałych ludzi. Na rys. 2.2 przedstawiono robota, który jest wykorzystywany przez służby pożarnicze do poszukiwania ludzi w płonących budynkach lub nadzorowania akcji prowadzonej przez zespół strażaków [6]. Dodatkowo można znaleźć informacje o próbach adaptacji pojazdów latających do gaszenia pożarów. Są one wtedy odpowiednio wyposażone w rakiety, dysze gaśnicze lub za pośrednictwem węża pożarniczego są bezpośrednio podłączone do źródła wody [10]. Ze względu na popularność dronów znacznie powiększa się grono indywidualnych użytkowników, którzy wykorzystują te urządzenia do rejestracji wideo, robienia zdjęć lub rozrywki. Niektórzy producenci udostępniają możliwość programowania swoich dronów, co czyni je świetnym narzędziem edukacyjnym zarówno w zakresie oprogramowania jak i fizyki kryjącej się za możliwością lotu takich urządzeń.



Rysunek 2.1 Mattnet M2 [28]

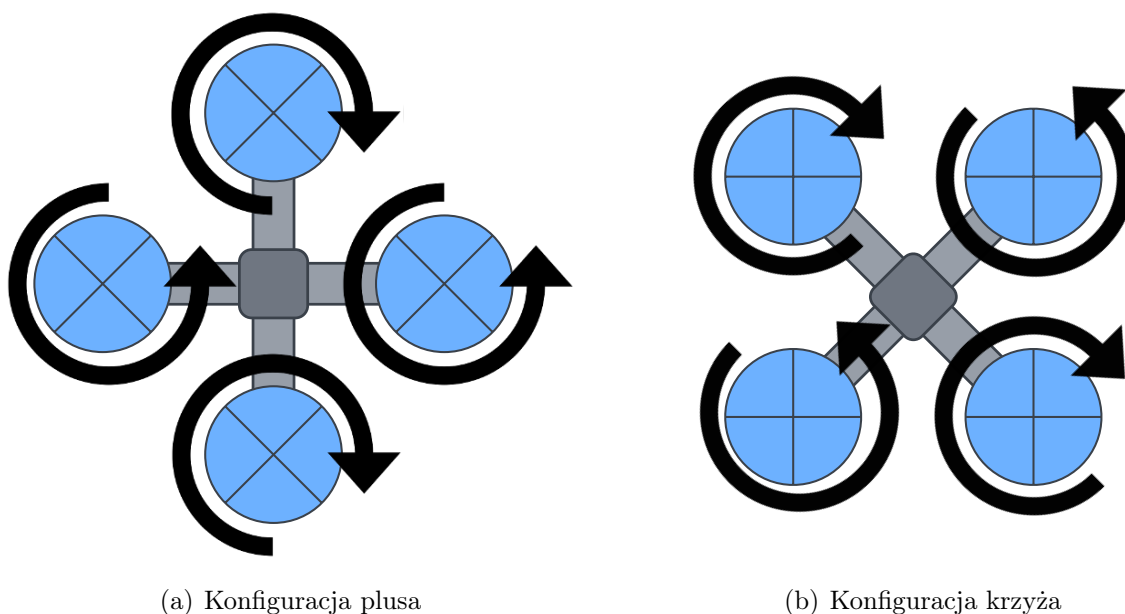


Rysunek 2.2 DJI M300 RTK [47]

2.1 Mechanika lotu

Drony czterowirnikowe są wyposażone w cztery silniki z przymocowanymi śmigłami. W zależności od usytuowania napędów względem korpusu drona można wyróżnić następujące konfiguracje:

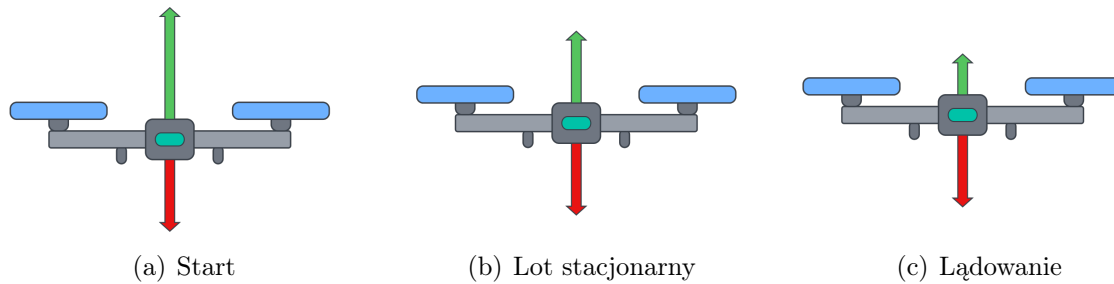
- konfiguracja plusa (z ang. *plus-configuration*),
- konfiguracja krzyża (z ang. *cross-configuration*).



Rysunek 2.3 Konfiguracje czterowirnikowca

Konfiguracja krzyża przedstawiona na rys. 2.3(b) jest częściej stosowana wśród dronów takiego typu, gdyż charakteryzuje się większą stabilnością na niewielkie zmiany prędkości śmigieł w porównaniu do alternatywnej konfiguracji widocznej na rys. 2.3(a) [36]. Na rys. 2.3 zaznaczono kierunki obrotów poszczególnych wirników. W obu konfiguracjach występują dwie pary śmigieł. Jedna z nich realizuje ruch zgodnie z kierunkiem wskazówek zegara (z ang. *clockwise*, CW), a druga ruch przeciwny (z ang. *counterclockwise*, CCW). Śmigła z każdej pary są usytuowane naprzeciwko siebie. Dzięki temu momenty generowane przez silniki niwelują się i urządzenie nie wpada w niekontrolowany ruch obrotowy. Na podobnej zasadzie funkcjonują helikoptery, gdzie śmigło ogonowe przeciwdziała momentowi wytwarzanemu przez główny wirnik pojazdu. Ze względu na dwa różne rodzaje obrotów, które są realizowane przez napędy, śmigła czterowirnikowca muszą być odpowiednio wyprofilowane dla każdego z kierunków. Śmigło posiada dwie krawędzie, jedną wyraźnie grubszą krawędź natarcia, która jest zorientowana w kierunku obrotu śmigła oraz cieńszą krawędź spływu. Taka konstrukcja śmigieł zapewnia prawidłową generację siły umożliwiającej lot robota.

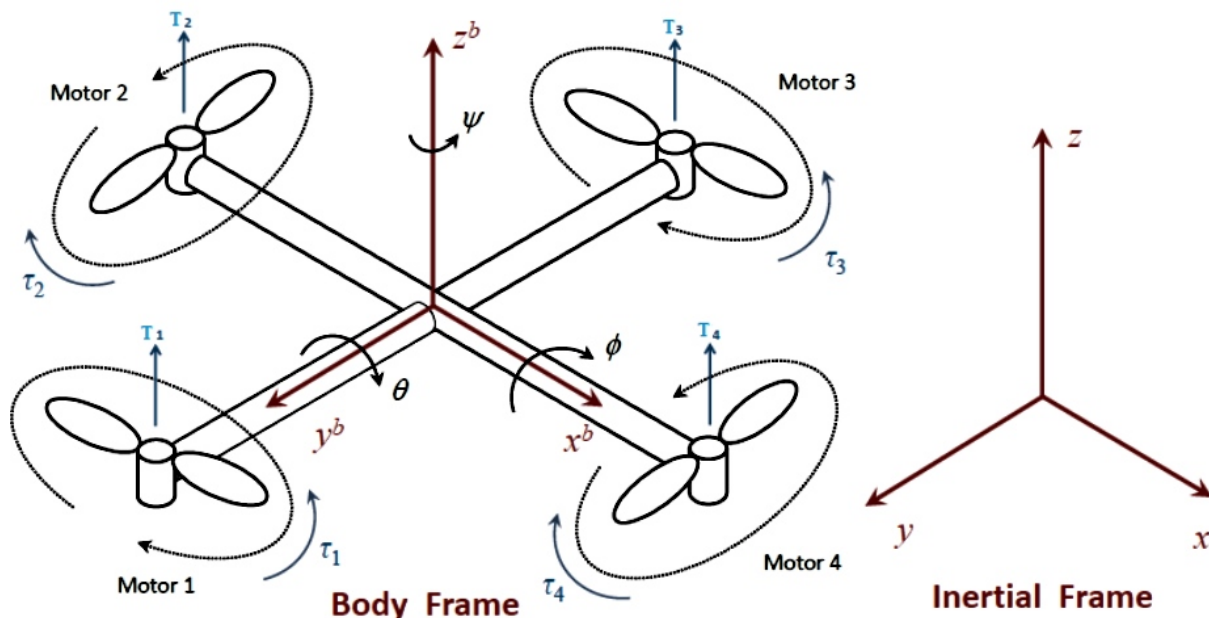
W celu wystartowania drona, należy uruchomić napędy, które wpychają powietrze w dół generując tym samym siłę nośną. Jeżeli siła nośna oznaczona strzałką w kolorze zielonym będzie większa od siły grawitacji oznaczonej kolorem czerwonym, tak jak to przedstawiono na rys. 2.4(a), dron uniesie się w powietrze. W przypadku, gdy te dwie



Rysunek 2.4 Podstawowe akcje ruchu drona

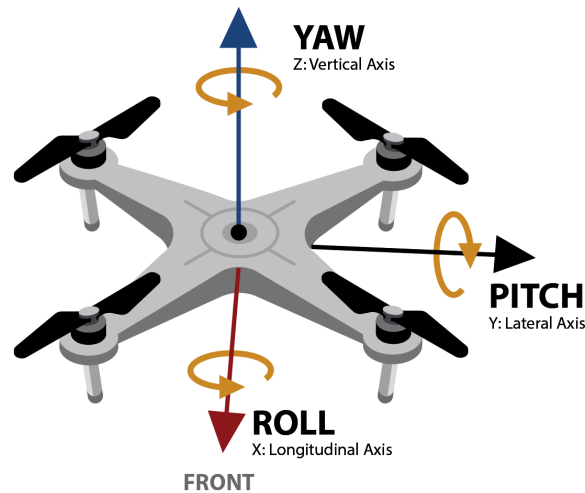
siły będą się równoważyć, co można zaobserwować na rys. 2.4(b) dron będzie utrzymywał stałą wysokość lotu. Aby zrealizować procedurę lądowania należy odpowiednio zmniejszyć prędkości obrotowe napędów, tak aby zredukować siłę nośną. Wtedy siła grawitacji będzie większa od siły powodującej lot urządzenia, co zaprezentowano na rys. 2.4(c). Dron zacznie tracić na wysokości i ostatecznie wylądować.

Czterowirnikowce są najczęściej opisywane w przestrzeni za pośrednictwem dwóch układów współrzędnych odniesienia, które zostały przedstawione na rys. 2.5. Są to odpo-



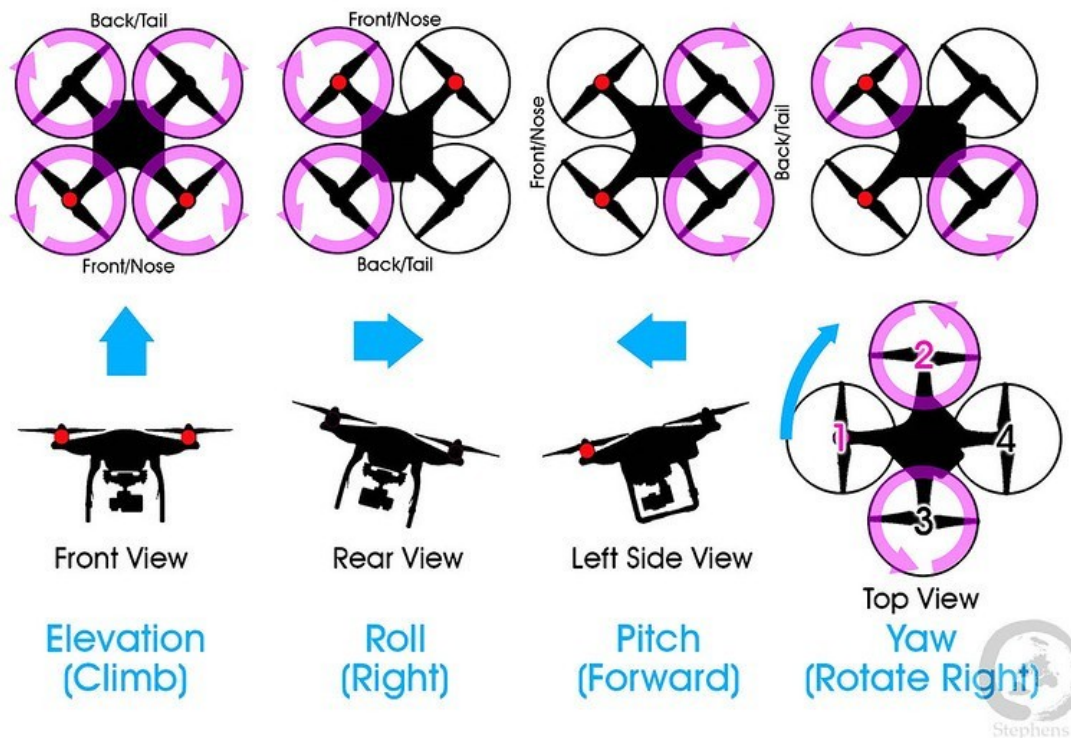
Rysunek 2.5 Układy współrzędnych drona [36]

wiednio układ stowarzyszony z samym robotem (z ang. *body frame*) oraz układ inercyjny (z ang. *inertial frame*). Pierwszy z nich jest określony przez położenie drona, gdzie osie napędów są zorientowane w dodatnim kierunku osi Z , a ramiona robota odpowiednio zgodnie z osiami X i Y [18]. W przypadku układu inercyjnego jest on zdefiniowany na podstawie podłoża. Siła grawitacji w tym układzie jest zorientowana w dół, wzdłuż osi Z . Czterowirnikowce charakteryzują się sześcioma stopniami swobody, gdyż mogą realizować ruch translacyjny w trzech kierunkach oraz ruch obrotowy względem trzech osi. Umożliwia to im przemieszczanie się w przestrzeni na wiele sposobów. Niemniej jednak najczęściej opisuje się fundamentalne ruchy, do których należą odpowiednio *roll*, *pitch* i *yaw*. Każdy z nich wiąże się z rotacją względem poszczególnych osi. *Roll* jest rotacją względem osi X , która jest zorientowana w kierunku frontu urządzenia, *pitch* realizuje obrót względem osi Y , a *yaw* względem osi Z . Orientacja układu współrzędnych dla czterowirnikowca



Rysunek 2.6 Roll, pitch, yaw [11]

w konfiguracji krzyża jak i odpowiednio przypisane nazwy ruchów zostały wyszczególnione na rys. 2.6. Realizacja poszczególnych obrotów jest możliwa poprzez odpowiednie sterowanie prędkościami napędów drona. Zwiększając prędkość wirników znajdujących się po lewej lub prawej stronie drona jest realizowany ruch *roll*, który przekłada się na przemieszczenie urządzenia wzdłuż osi Y . Odpowiednio podwyższając prędkość napędów znajdujących się z przodu lub z tyłu robota uzyskuje się ruch *pitch*, który powoduje lot wzdłuż osi X . Ostatni rodzaj ruchu, czyli *yaw* jest realizowany za pośrednictwem regulacji prędkości naprzeciwległych wirników. Pozwala on wymusić obrót drona w miejscu, dzięki rotacji względem osi Z . Wszystkie opisane powyżej ruchy drona zostały zaprezentowane na rys. 2.7, gdzie różowym kolorem zaznaczono napędy o zwiększonej prędkości obrotowej.

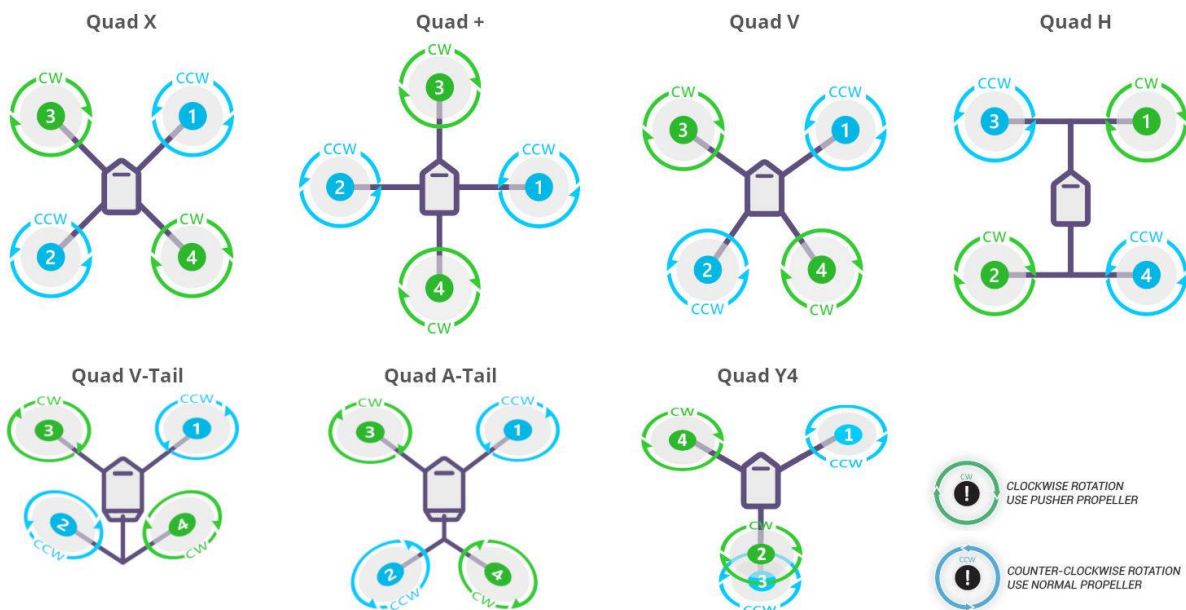


Rysunek 2.7 Ruchy czterwornikowca [50]

2.2 Konfiguracja sprzętowa

Drony czterowirnikowe składają się z poszczególnych podzespołów elektronicznych i elementów mechanicznych. Każdy z nich pełni swoją określoną funkcję i w dużym stopniu jest zależny od pozostałych. Ich wybór jest uwarunkowany od finalnego przeznaczenia urządzenia jak i jego oczekiwanych parametrów. Do podstawowych komponentów, z których można skonstruować drona, należą: rama, silniki, elektroniczny regulator prędkości, śmigła, kontroler lotu, odbiornik i nadajnik umożliwiający komunikację oraz bateria. Dodatkowo można wyposażyć takie roboty w kamerę, która będzie przechwytywała obraz widoczny z jego perspektywy.

Rama urządzenia jest platformą, na której znajdują się wszystkie pozostałe elementy. W związku z tym musi być na tyle wytrzymała i przestronna, aby umożliwić dogodne rozmieszczenie komponentów, ale również możliwie lekka, aby nie zwiększała znacząco całkowitej masy drona. W związku z tym do produkcji ram wykorzystuje się między innymi kompozyty z włókna węglowego (z ang. *carbon fiber reinforced polymer*, CRFP), które charakteryzują się niewielką masą i wysoką wytrzymałością [3]. Oprócz rodzajów konfiguracji przedstawionych na rys. 2.3 można spotkać się z bardziej złożonymi alternatywami, które zostały zaprezentowane na rys. 2.8. Na wybór konfiguracji wpływa przeznaczenie urządzenia. Rozmiar ramy przekłada się bezpośrednio na wielkość śmigieł, co również ma wpływ na gabaryty zastosowanych silników [30]. Następnie od wymiarów silników zależy dobór elektronicznego regulatora prędkości o odpowiedniej wydajności prądowej. Ze względu na korelację pomiędzy wymienionymi elementami selekcja ramy jako podstawy całego robota jest bardzo istotna już na etapie projektowym.



Rysunek 2.8 Ramy w dronach czterowirnikowych [51]

W kwestii wyboru napędów w czterowirnikowcach najczęściej stosuje się silniki bezszczotkowe prądu stałego (z ang. *brushless DC electric motor*, BLDC). Silniki BLDC charakteryzują się lepszymi parametrami od silników szczotkowych prądu stałego. Są w stanie wygenerować większy moment obrotowy, są cichsze oraz mają większą wydajność przy znacznie mniejszej mocy i wadze [2]. Niemniej jednak ze względu na niską cenę i proste sterowanie w niektórych urządzeniach nadal wykorzystuje się zwykłe silniki DC.



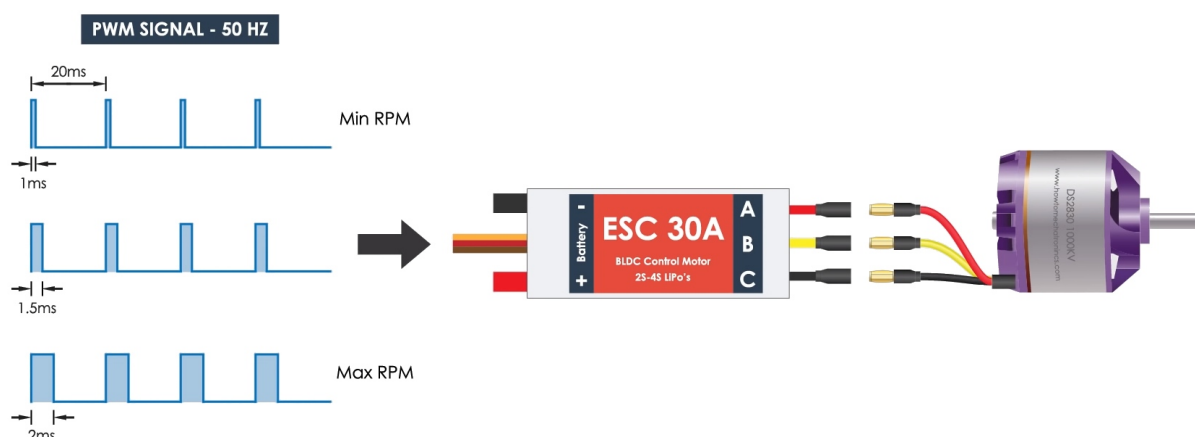
Rysunek 2.9 Silniki BLDC [7]



Rysunek 2.10 ESC [9]

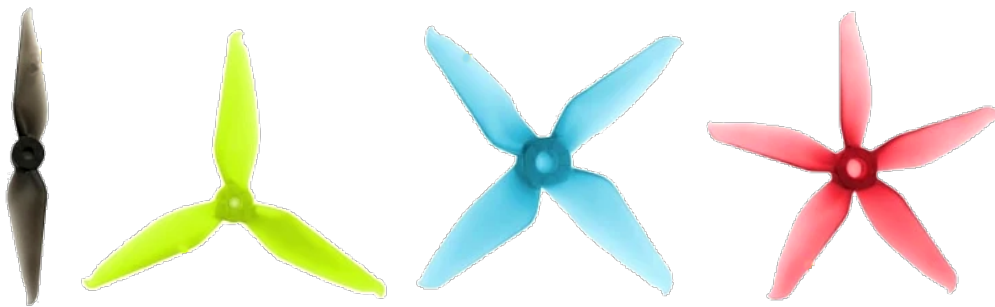
W opisie właściwości silników bezszczotkowych podaje się współczynnik kV . Określa on z jaką prędkością obraca się silnik bez obciążenia po przyłożeniu napięcia o wartości 1 V. Silniki zaprezentowane na rys. 2.9 charakteryzują się współczynnikiem na poziomie 2600 kV, co odpowiada 2600 obrotom na minutę (z ang. *revolutions per minute*, RPM) przy zasilaniu 1 V. Przy wyborze silników należy dodatkowo uwzględnić takie parametry jak napięcie zasilania oraz pobierany prąd.

Następnym elementem jest elektroniczny regulator prędkości (z ang. *electronic speed controller*, ESC). Jego zadaniem jest sterowanie prędkością obrotową silników na podstawie wypełnienia sygnału o modulowanej szerokości impulsu (z ang. *pulse width modulation*, PWM) otrzymywanego z kontrolera robota. Regulator posiada trzy przewody, którymi zasilają się odpowiednio trzy bieguny silnika bezszczotkowego, przewody pozwalające podłączyć źródło zasilania oraz złącze umożliwiające podanie sygnału PWM. Regulator widoczny na rys. 2.10 może być zasilony pakietem baterii Li-Pol (akumulatory litowo-polimerowe) od 2 do 4 ogniw (7,4V do 14,8V). Przykładowe połączenie regulatora prędkości z napędem przedstawiono na rys. 2.11. Dobierając ESC do konkretnego urządzenia należy kierować się zapotrzebowaniem prądowym wykorzystywanych napędów. Szacuje się, że wydajność prądowa regulatora prędkości powinna być od 1.2x do 1.5x większa od maksymalnego prądu silników drona [14]. W przypadku czterowirnikowca każdy z napędów powinien posiadać swój elektroniczny regulator prędkości.

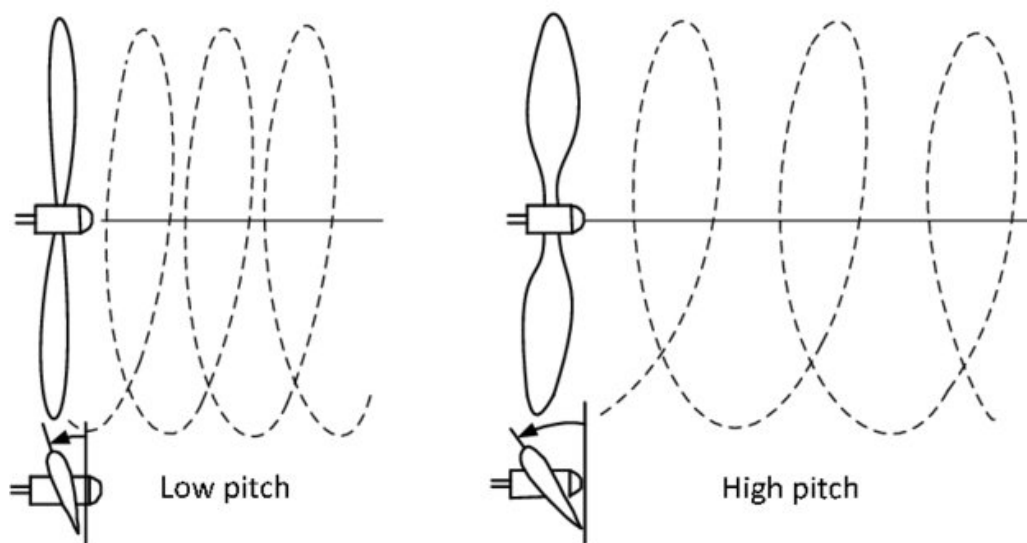


Rysunek 2.11 Schemat połączenia regulatora prędkości z silnikiem [38]

Śmigła są odpowiedzialne za wytworzenie siły ciągu, która umożliwia unoszenie się drona. Produkują je z plastiku lub włókna węglowego. Zaletami plastikowych rozwiązań jest cena oraz wytrzymałość na ewentualne kolizje. W przypadku ich odpowiedników z włókna węglowego uzyskuje się większą siłę nośną, ze względu na ograniczone wibracje podczas obrotów, co przekłada się na bardziej stabilny lot [19]. Niestety są one kosztowne oraz łatwo ulegają zniszczeniu podczas zderzeń. Głównym parametrem śmigła jest ich średnica. Wraz ze wzrostem średnicy śmigła wytwarza większy ciąg, ale jednocześnie potrzebuje więcej energii, aby pokonać opory ruchu. Kolejnym parametrem śmigła jest ich skok (z ang. *pitch*). Jest to dystans jaki przebyłoby śmigło poruszając się przed siebie wykonując jeden obrót, co zostało zobrazowane na rys. 2.13. Skok zależy od kąta nachylenia łopaty. Gdy łopata jest prawie płaska nie będzie napotykała dużych oporów powietrza, jednak wygeneruje mniejszy ciąg. W przypadku śmigła o dużym kącie natarcia występują znaczne opory powietrza, co przekłada się na dużą siłę ciągu. Wśród czterowirnikowców można spotkać śmigła posiadające od 2 do nawet 6 łopat, niektóre z nich przedstawiono na rys. 2.12. Śmigła o większej liczbie łopat generują większą siłę ciągu, jednak jednocześnie wymagają większej mocy, gdyż zwiększają masę samego urządzenia. Biorąc pod uwagę wcześniej wymienione aspekty w dronach najczęściej wykorzystuje się względnie krótkie śmigła o niewielkiej liczbie łopat. Zapewniają one również lepszą manewrowość urządzenia, przy niskim zapotrzebowaniu mocy.

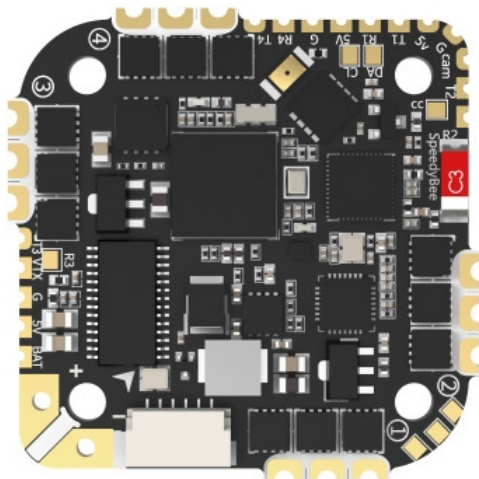


Rysunek 2.12 Śmigła o różnej liczbie łopat [40]



Rysunek 2.13 Skok śmigła [12]

Kontroler lotu (z ang. *flight controller*, FC) jest postrzegany jako element centralny urządzenia latającego [30]. Jest to komponent elektroniczny, który realizuje sterowanie dronem na podstawie odczytów z wbudowanego zestawu sensorów oraz komend nadawanych przez użytkownika. Dodatkowo pełni on również rolę swoistego koncentratora (z ang. *hub*) dla pozostałych elementów peryferyjnych. Posiada dedykowane wyprowadzenia, do których można podłączyć elektroniczne regulatory prędkości, kamerę, interfejsy komunikacyjne oraz moduł GPS (z ang. *global positioning system*). W skład kontrolera lotu wchodzi takie czujniki jak żyroskop, akcelerometr oraz barometr [32]. Wymienione sensory przeważnie są zintegrowane w IMU (z ang. *Inertial Measurement Unit*), który składa się z mikrosystemów elektromechanicznych (z ang. *microelectromechanical system*, MEMS), czujników bezwładnościowych i magnetorezystancyjnych [46]. W projektach DIY (z ang. *do it yourself*) najczęściej wykorzystuje się kontrolery lotu wyposażone w mikroprocesory (z ang. *micro controller unit*, MCU) z rodziny STM32 widoczny na rys. 2.14. Dostępne są również rozwiązania obsługiwane przez procesory Intel, przykładową płytkę przedstawiono na rys. 2.15. Producenci udostępniają również oprogramowanie umożliwiające przystępną konfigurację kontrolera.



Rysunek 2.14 SpeedyBee F745 35A BLS AIO [48]

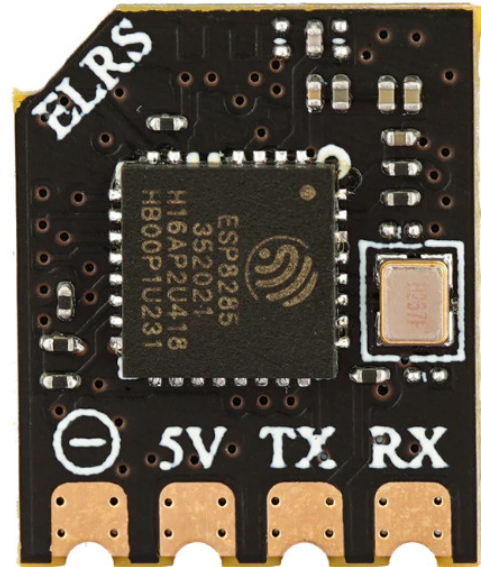


Rysunek 2.15 Intel® Aero Compute Board [27]

Do komunikacji z dronem, którą pośrednią opisano w rozdziale 1 można wykorzystać nadajnik (transmitter radiowy), który umożliwia bezprzewodową kontrolę urządzenia. Nadajniki są realizowane w postaci urządzeń z drążkami sterowniczymi, których wychylenie powoduje adekwatne akcje ruchu robota powietrznego. Przykład takiego urządzenia przedstawiono na rys. 2.16. Transmityery operują na częstotliwości 2.4 GHz lub w przypadku urządzeń dalekiego zasięgu wykorzystujących protokół LoRa (z ang. *long range*) na częstotliwości obsługiwanej przez dany region [33]. W Europie wynosi ona 868 MHz. Aby sygnał był odbierany przez robota, należy wyposażyć go w odbiornik, który będzie współpracował z wybranym nadajnikiem. Odbiornik widoczny na rys. 2.17 pozwala na przesyłanie otrzymanych komend do kontrolera lotu, z którym jest bezpośrednio połączony.



Rysunek 2.16 RadioMaster TX16S Mark II [42]



Rysunek 2.17 RP1 ExpressLRS 2.4 GHz Nano Receiver [41]

Elementy elektroniczne drona potrzebują źródła zasilania. W robotach latających takim źródłem są baterie Li-Pol (akumulatory litowo-polimerowe). Charakteryzują się wysoką gęstością energii i niewielką wagą [31]. Baterie składają się z pojedynczych ogniw. Jedno ogniwo posiada nominalne napięcie o wysokości 3.7 V. W razie zapotrzebowania na wyższe napięcie można połączyć ogniwa szeregowo uzyskując odpowiednio wielokrotności napięcia pojedynczego ogniwa. Takim sposobem otrzymuje się źródła zasilania złożone od 1 do 6 ogniw o napięciu nominalnym od 3.7 V do 22.2 V. Jednym z ważniejszych parametrów baterii jest ich pojemność, która informuje ile prądu można pobrać z baterii zanim nastąpi jej rozładowanie. Jednostką pojemności w bateriach jest *mAh* (z ang. *milliampere hour*). Przy wyborze pojemności należy uwzględnić, że jej wzrost przełoży się na większą masę robota. Baterie posiadają również informację o prądzie rozładowania (z ang. *discharge rate*), który określa maksymalny prąd, który może być pobierany z baterii. Przykładowy pakiet baterii Li-Pol zaprezentowano na rys. 2.18.



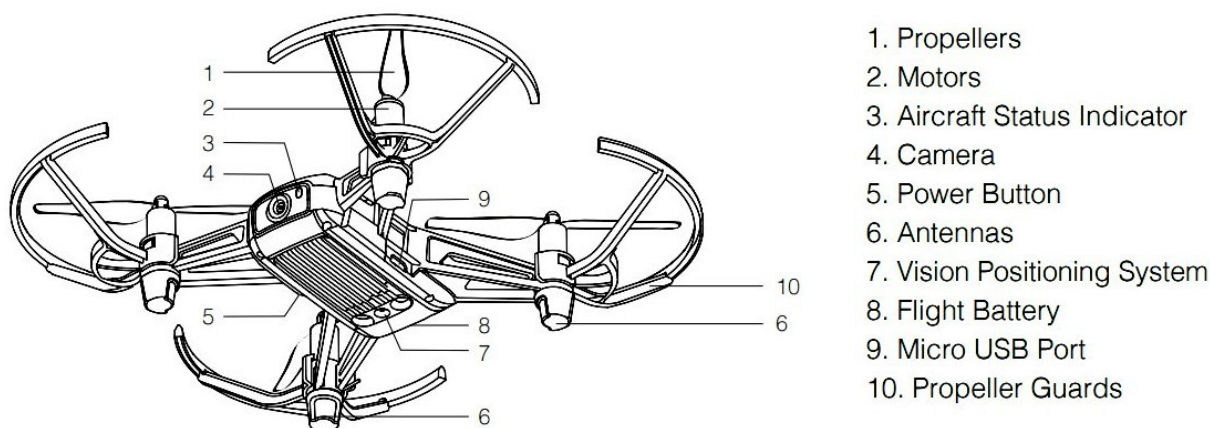
Rysunek 2.18 Pakiet Li-Pol Redox 2200mAh 20C 3S 11.1V [8]

Rozdział 3

DJI Ryze Tello

W poniższym rozdziale scharakteryzowano drona czterowirnikowego DJI Ryze Tello, który został użyty do realizacji projektu w ramach niniejszej pracy inżynierskiej. Dokonano opisu parametrów technicznych tego urządzenia oraz jego sposobów sterowania. Dodatkowo przedstawiono narzędzia programistyczne, które zostały wykorzystane w projekcie aplikacji, umożliwiające interakcję z dronem.

DJI Ryze Tello widoczny na rys. 3.2 jest niewielkim dronem czterowirnikowym, który jest przeznaczony do użytkowania w przestrzeniach zamkniętych i otwartych. Niemniej jednak lot w otwartej przestrzeni może być realizowany tylko przy odpowiednich warunkach pogodowych. Dron składa się z kontrolera lotu, systemu wizyjnego pozycjonowania, systemu nagrywania wideo, układu napędowego i zasilania w postaci baterii [16]. Dodatkowo można go wyposażyć w plastikowe zderzaki, które chronią śmigła podczas ewentualnego bezpośredniego kontaktu z przeszkodą. Poszczególne elementy urządzenia zaznaczono na schemacie przedstawionym na rys. 3.1. Całkowita waga robota powietrznego z zamontowanymi zderzakami wynosi 87 g. Ze względu na niewielką masę wszelkie niedogodności w postaci nawet lekkich podmuchów wiatru znacznie utrudniają sterowanie urządzeniem w plenerze. Wymiary gabarytowe drona to 98 mm długości, 92.5 mm szerokości i 41 mm wysokości, co czyni go niezwykle kompaktowym i łatwym w transporcie.



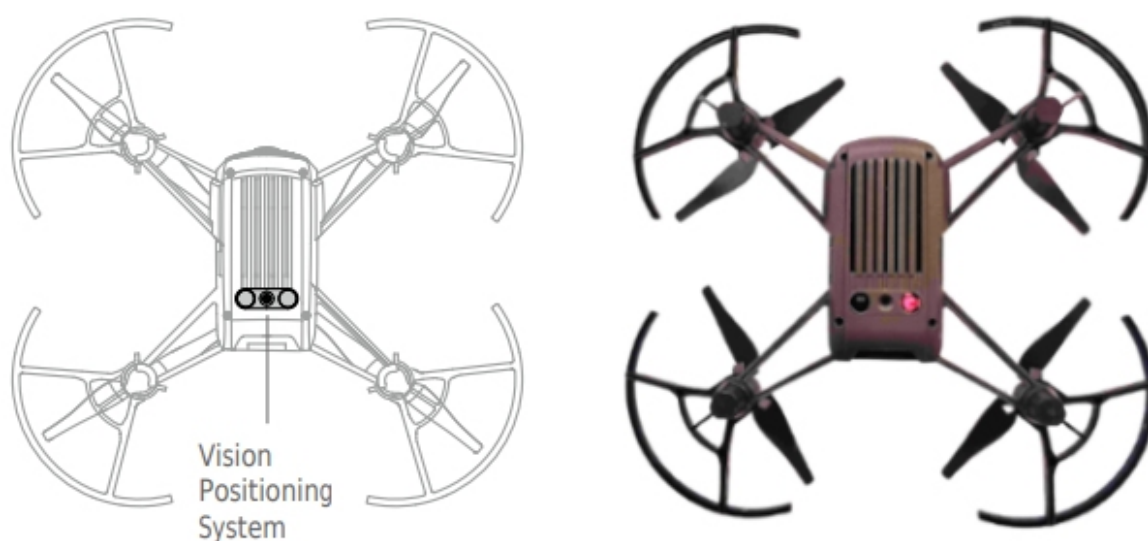
Rysunek 3.1 Schemat drona DJI Ryze Tello [16]



Rysunek 3.2 Dron DJI Ryze Tello [35]

Dron DJI Ryze Tello jest wyposażony w zaawansowany 14-rdzeniowy procesor firmy Intel, który zawiera zintegrowany moduł przetwarzania obrazów Intel[®] Movidius[™] Myriad[™] Vision Processing Unit (VPU) [44]. Moduł ten odciąża główny procesor urządzenia, aby nie musiał on realizować zadań związanych z obliczeniami dotyczącymi przetwarzania obrazów lub danych z sensorów. Jest to przykład akceleracji sprzętowej (z ang. *hardware acceleration*), gdzie konkretne zadanie jest wykonywane na dedykowanym komponencie sprzętowym [26]. Pozwala to na bardziej efektywną realizację poszczególnych poleceń. Równie ważną zaletą VPU jest zachowanie rozsądnej równowagi pomiędzy wydajnością obliczeniową, a konsumpcją energii elektrycznej, co jest bardzo istotnym aspektem w robotach powietrznych. Moduł ten realizuje analizę i przetwarzanie danych z kamery drona oraz czujników znajdujących się na jego pokładzie [44]. Wśród zainstalowanych sensorów znajduje się czujnik optyczny, akcelerometr, żyroskop, barometr oraz czujnik czasu lotu (z ang. *time-of-flight range sensor*. ToF) [39]. Sensor optyczny znajduje się w podwoziu statku i oblicza zmianę współrzędnych położenia robota w osi X i Y . Jest to kamera o niskiej rozdzielczości, która wykonuje zdjęcia podłoża i na podstawie różnicy pomiędzy poszczególnymi klatkami określa współrzędne urządzenia. Czujnik ten pozwala na obliczenie względnej prędkości drona jak i na realizację przez niego lotu stacjonarnego. Sensor ToF (dalmierz) również jest usytuowany w podwoziu czterowirnikowca i umożliwia określenie jego wysokości w stosunku do obiektów znajdujących się pod nim. Barometr dostarcza informację o absolutnej wysokości urządzenia. Żyroskop zapewnia stabilizację pojazdu latającego w powietrzu, a z kolei na podstawie odczytów akcelerometru można określić wartość przyspieszeń liniowych. Wymienione komponenty w połączeniu z oprogramowaniem umożliwiły osiągnięcie wielu funkcjonalności, które czynią drona DJI Ryze Tello konkurencyjnym w swoim przedziale cenowym.

Jednym z systemów występujących w dronie jest system wizyjnego pozycjonowania. Wykorzystuje on wymieniony wcześniej czujnik optyczny oraz dalmierz. Dalmierz jest modułem podczerwieni, który składa się z nadajnika i odbiornika światła podczerwonego (z ang. *infrared light*, IR). Na podstawie pomiaru czasu pomiędzy emisją sygnału w postaci światła, a jego powrotem do odbiornika, sensor określa odległość od obiektów znajdujących się pod dronem. Światło podczerwone nie jest widoczne dla oka człowieka, można zidentyfikować jego źródło poprzez wykonanie zdjęcia aparatem telefonu [20]. Emiter światła wyraźnie wyróżnia się na fotografii, tak jak jest to widoczne na rys. 3.3. Czujnik optyczny w postaci kamery znajduje się pomiędzy nadajnikiem i odbiornikiem światła podczerwonego. System wizyjnego pozycjonowania jest aktywowany automatycznie wraz z uruchomieniem drona i działa poprawnie, gdy robot znajduje się na wysokości w przedziale od 0.3 do 30 m. Niemniej jednak cechuje się największą efektywnością dla wysokości od 0.3 do 6 m [16]. Jeżeli dron znajdzie się poza tym przedziałem system może nie działać prawidłowo. Na wydajność tego systemu oddziałuje kilka czynników. W przypadku prędkości powyżej 18 km/h i lotu na wysokości 1 m lub niżej dron może nie być w stanie pozycjonować się w powietrzu. Kolejnym czynnikiem jest wartość natężenia oświetlenia otoczenia, w którym porusza się dron. Ciemne pomieszczenia i nieoświetlone przestrzenie otwarte, gdzie natężenie światła wynosi poniżej 10 lx są niesprzyjającym środowiskiem dla poprawnego funkcjonowania systemu pozycjonowania. Również podczas lotu w terenie charakteryzującym się natężeniem światła powyżej 100 000 lx, czyli na przykład przy bezpośrednim działaniu promieni słonecznych, mogą pojawić się problemy z utrzymaniem stabilnej pozycji urządzenia [16]. Ostatnim aspektem, który może zaburzyć działanie systemu jest rodzaj powierzchni, nad którą porusza się robot. Dotyczy to powierzchni o jednolitym kolorze, przezroczystych, o właściwościach odbłaskowych lub o wyraźnym braku tekstury i charakterystycznych elementów [16]. Każda z tych rodzajów powierzchni w pewnym stopniu zakłóca działanie czujników pozwalających na pozycjonowanie się urządzenia w przestrzeni. W przypadku, gdy dron znajduje się na wysokości powyżej 6 m, a system wizyjnego pozycjonowania nie będzie działał prawidłowo przez 3 sekundy, nastąpi uruchomienie procedury awaryjnego lądowania. Jeżeli w trakcie tej procedury system wróci do sprawnego funkcjonowania to zostanie ona przerwana i dron zacznie unosić się w miejscu.



Rysunek 3.3 System wizyjnego pozycjonowania [16]



Rysunek 3.4 Front drona DJI Ryze Tello



Rysunek 3.5 Zdjęcie wykonane dronem

Na przednim panelu urządzenia widocznym na rys. 3.4 usytuowana jest wbudowana kamera wideo oraz dioda statusowa. Kamera umożliwia użytkownikowi wykonywanie zdjęć o rozdzielczości 5 megapikseli, co przekłada się na obraz o wymiarach 2592 na 1936 pikseli, gdzie pierwszy z wymiarów dotyczy szerokości, a drugi wysokości zdjęcia [16]. Na rys. 3.5 zaprezentowano zdjęcie wykonane kamerą drona. Poza tym kamera pozwala na nagrywanie filmów w rozdzielczości 720p i do 30 klatek na sekundę w formacie MP4 [16]. Dzięki zastosowaniu elektronicznej stabilizacji cyfrowej dron jest w stanie wykonywać względnie dobre zdjęcia i nagrania wideo podczas swojego lotu. Robot nie jest wyposażony w miejsce na pamięć zewnętrzną na przykład w postaci karty pamięci. W związku z tym wszelkie materiały w postaci zdjęć i nagrań zarejestrowane przez kamerę są bezpośrednio przesyłane do dedykowanej aplikacji na telefonie, która umożliwia sterowanie dronem. Następnie użytkownik może wybrać, które zdjęcia lub filmy chce zapisać na swoim telefonie. Przesyłanie danych i tym samym komunikacja z urządzeniem odbywa się za pośrednictwem sieci Wi-Fi operującej na częstotliwości w zakresie od 2.4 GHz do 2.4835 GHz [16]. Więcej informacji na temat sterowania oraz przesyłu danych znajduje się w rozdziałach 3.1 i 3.2.

Dioda sygnalizująca poszczególne stany pojazdu znajduje się tuż obok kamery. Informacja o poszczególnych statusach jest sygnalizowana za pośrednictwem różnych kolorów diody i jej sposobu świecenia. Wśród dostępnych kolorów znajduje się czerwony, żółty, zielony i niebieski. Dioda może emitować światło w sposób ciągły lub poprzez miganie o różnej częstotliwości. Kombinacja kolorów i wariantów iluminacji diody pozwala na przekazanie informacji o stanie wizyjnego systemu pozycjonowania, ładowania i stanów ostrzegawczych. Wraz z uruchomieniem drona dioda miga zmieniając swoje kolory od czerwonego przez zielony do żółtego. Taki komunikat świadczy o przeprowadzaniu testów auto diagnostycznych przez urządzenie przed przystąpieniem do lotu. Jeżeli system pozycjonowania jest aktywny i działa prawidłowo to dioda co jakiś czas miga dwa razy kolorem zielonym. W przypadku, gdy system nie może określić położenia robota dioda zmienia kolor na żółty i zaczyna powoli mrugać. Podczas ładowania urządzenia wyróżnia się trzy stany sygnalizowane przez niebieski kolor diody. W trakcie tego procesu dioda naprzemiennie zapala się i gaśnie. Po zakończeniu ładowania dioda pali się światłem ciągłym. W przypadku wystąpienia błędu podczas ładowania dioda będzie migała z wysoką częstotliwością. Ostatnimi stanami, o których informuje dioda, są stany ostrzegawcze. Podczas sterowania, gdy dron utraci połączenie z kontrolerem, dioda zacznie szybko migać żółtym światłem. Kolejne krytyczne stany są sygnalizowane przez kolor czerwony. Przy niskim stanie akumulatora pojawia się powolne miganie. Wraz z zmniejszającym się

poziomem naładowania baterii zwiększa się częstotliwość migania. Podczas wystąpienia błędu krytycznego dioda świeci ciągłym, czerwonym światłem. Na rys. 3.6 przedstawiono przykładowe stany drona sygnalizowane przez adekwatne kolory diody.

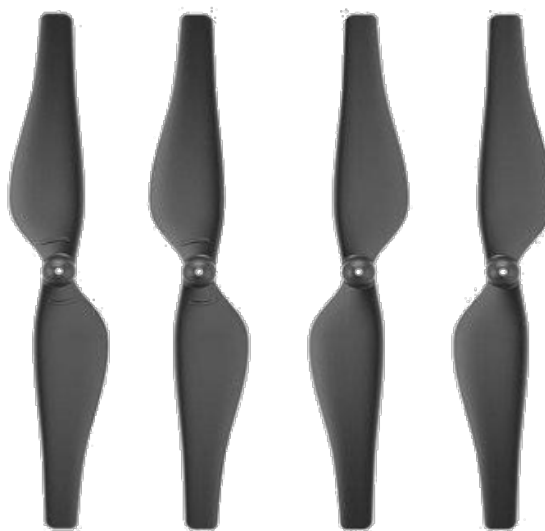


Rysunek 3.6 Kolory diody statusowej

Tak jak wspomniano w rozdziale 2.2 silniki bezszczotkowe pod wieloma względami wypadają lepiej na tle standardowych, szczotkowych silników prądu stałego. Jednakże napędy BLDC charakteryzują się znacznie wyższą ceną i właśnie między innymi z tego względu w dronie DJI Ryze Tello producent zdecydował się na zastosowanie tańszej opcji w postaci silników szczotkowych. Jest to kompromis pomiędzy wydajnością urządzenia, a jego kosztem produkcji, co czyni je dostępnym dla większej liczby potencjalnych użytkowników. W informacjach udostępnionych przez producenta nie widnieje dokładany model zastosowanych napędów. W przypadku uszkodzenia fabrycznego napędu sugerowanym zamiennikiem jest silnik typu 8520 przedstawiony na rys. 3.7. Jest on zasilany napięciem 3.7 V, a jego prędkość pod obciążeniem wynosi 24200 RPM [4]. Na wałkach napędów są zamocowane śmigła 3044P o średnicy 7.5 cm widoczne na rys. 3.8. Są one odpowiednio wyprofilowane w zależności od kierunku obrotów silnika. Tak skonstruowany układ napędowy umożliwia uzyskanie prędkości lotu do 28.8 km/h.



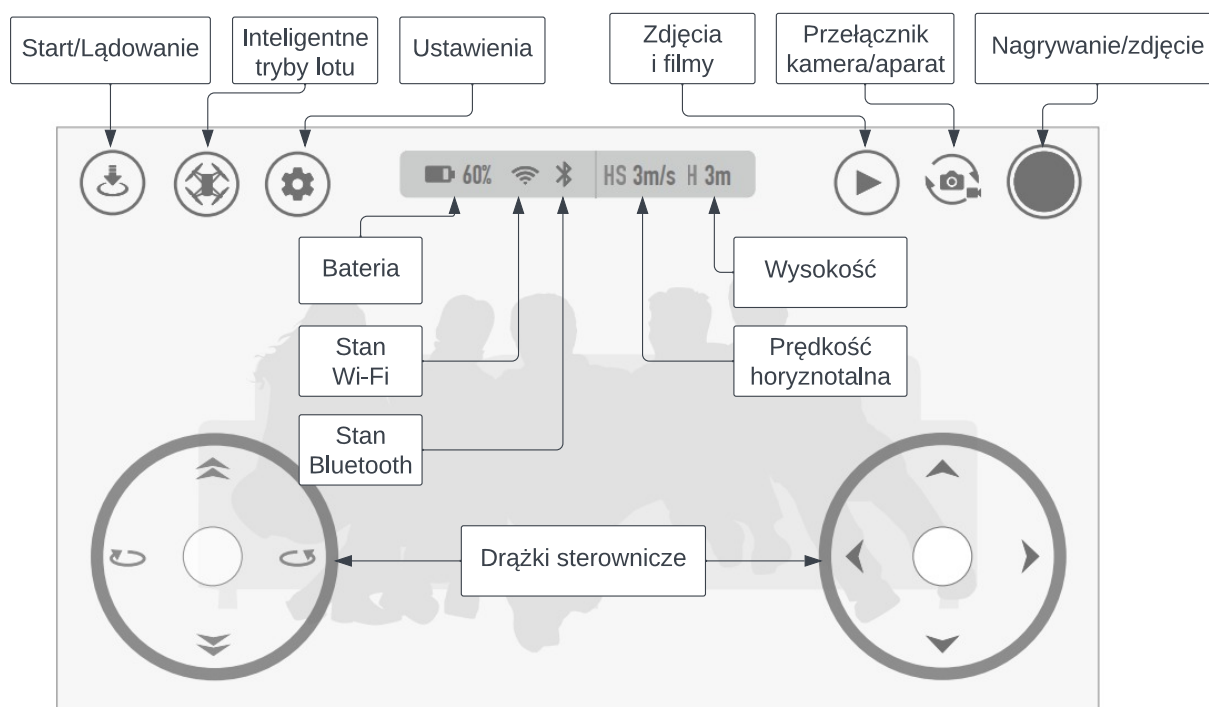
Rysunek 3.7 Silniki RCGEEK 8520 [4]



Rysunek 3.8 Śmigła 3044P [34]

Dron DJI Ryze tello jest zasilany baterią Li-Pol (akumulator litowo-polimerowy) o napięciu znamionowym 3.8 V i pojemności 1100 mAh [16]. Bateria posiada zabezpieczenie nadprądowe (z ang. *overcurrent*) oraz przepięciowe (z ang. *overvoltage*). Bateria przestaje się ładować, gdy zostanie wykryty zbyt duży prąd lub napięcie. Posiada również zabezpieczenie przed nadmiernym rozładowaniem (z ang. *overdischarge*). Jest to istotna kwestia w przypadku tego typu baterii, gdyż przy zbyt mocnym rozładowaniu ogniwa może ono ulec bezpowrotnemu uszkodzeniu. Dodatkowo bateria automatycznie wyłącza się w momencie pojawienia się zwarcia w obwodzie urządzenia. Ładowanie baterii odbywa się poprzez podłączenie drona przez kabel typu micro USB do źródła zasilania. Alternatywną opcją jest ładowanie baterii w dedykowanej stacji, która posiada trzy miejsca na akumulatory. Takie rozwiązanie realizuje ładowanie sekwencyjne, to znaczy baterie są ładowane po kolei, a nie równolegle. Czas ładowania jednej baterii wynosi około 1.5 h [16]. Zastosowana bateria pozwala na lot o długości 13 minut w bezwietrznych warunkach i stałej prędkości 15 km/h [16]. W zależności od intensywności lotu oraz wykonywanych manewrów podczas lotu czas ten może być znacznie krótszy.

3.1 Sterowanie

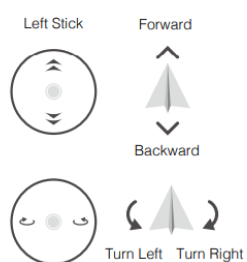


Rysunek 3.9 Widok aplikacji TELLO [16]

Podstawowym sposobem sterowania dronem DJI Ryze Tello jest wykorzystanie dedykowanej aplikacji TELLO dostępnej na platformach z systemem Android i iOS. Rozkład poszczególnych komponentów aplikacji został zaprezentowany na rys. 3.9. Za pośrednictwem aplikacji można obsługiwać kamerę drona, dokonać konfiguracji samego robota i przesyłać materiały zarejestrowane za pomocą kamery do pamięci swojego urządzenia. Żeby sterować dronem należy połączyć swoje urządzenie mobilne z siecią Wi-Fi czterywirtualnego. Nazwa sieci przyjmuje format TELLO-XXXXXX i domyślnie nie wymaga wprowadzenia hasła [16]. Po pomyślnym połączeniu drona w oknie aplikacji pojawi się

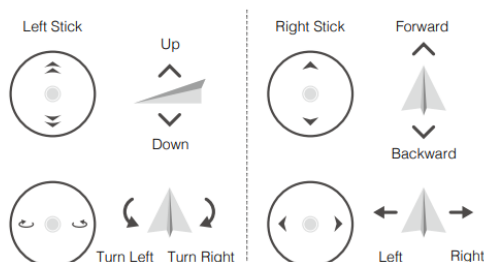
obraz z jego kamery oraz zmieni się status ikonki stanu Wi-Fi. Zaczynając od lewej strony interfejsu widocznego na rys. 3.9 znajduje się tam przycisk umożliwiający realizację procedury startu i lądowania drona. Następny przycisk odpowiada za aktywację inteligentnych trybów lotu. Są to odpowiednio tryb Bounce, Throw & Go, 8D Flips, Up & Away, 360 i Circle [16]. Aby użyć któregoś z trybów bateria robota musi być naładowana w ponad 50%. W trybie Bounce dron automatycznie realizuje lot w górę i w dół w zakresie od 0.5 m do 1.2 m nad powierzchnią, która się pod nim znajduje. W momencie umieszczenia dłoni lub innego obiektu pod pojazdem, zwiększy on swoją wysokość lotu i będzie kontynuował ruch w osi pionowej. Tryb Throw & Go umożliwia start drona poprzez delikatne wyrzucenie go w powietrze. 8D Flips pozwala na wykonywanie przewrotów dronem w jednym z ośmiu kierunków. Pozostałe tryby realizują nagrywanie obrazu podczas wykonywania zdefiniowanych sekwencji ruchów. Tak odpowiednio Up & Away nagrywa krótkie wideo lecąc do góry, a następnie do tyłu. W trybie 360 dron nagrywa otoczenie wykonując pełen obrót wokół własnej osi. Z kolei w trybie Circle realizuje lot po okręgu. Ikonka zębatego koła pozwala przejść do ustawień, gdzie można między innymi dokonać kalibracji czujników drona, zaktualizować jego oprogramowanie, wybrać jakość zdjęć, tryb wolnego lub szybkiego lotu oraz rodzaj sterowania za pośrednictwem drążków. Wolny tryb lotu jest domyślnym ustawieniem drona, gdzie maksymalna prędkość jaką może osiągnąć wynosi 10.8 km/h. Zmieniając tryb na szybki można osiągnąć prędkość nawet do 28.8 km/h. Rodzaj sterowania wiąże się z przypisaniem odpowiednich akcji ruchu do poszczególnych stanów drążków sterowniczych. Dostępne tryby zostały przedstawione na rys 3.11 i 3.10.

Mode 1



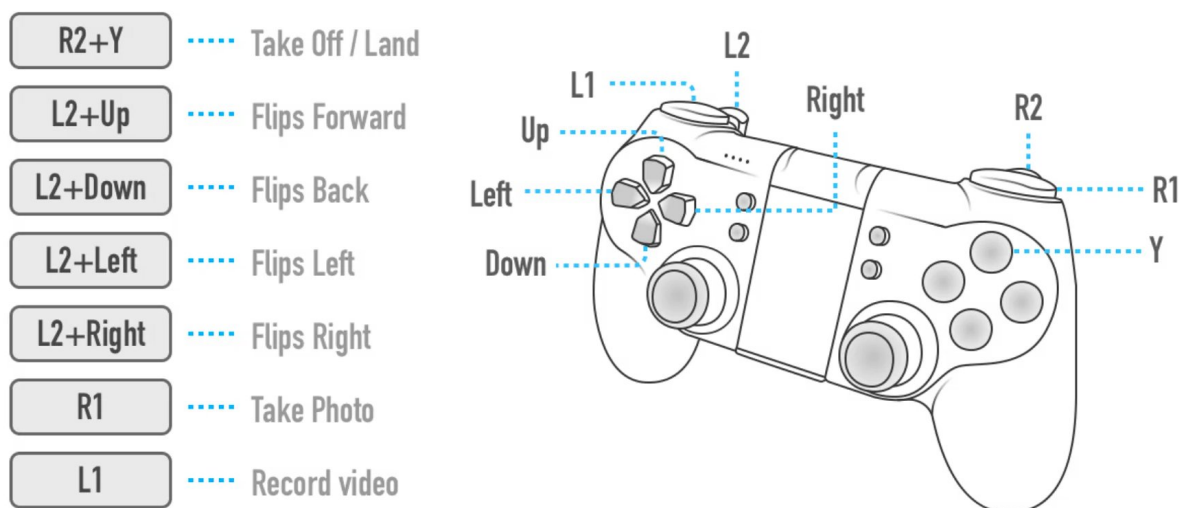
Rysunek 3.10 Tryb 1 sterowania [16]

Mode 2



Rysunek 3.11 Tryb 2 sterowania [16]

Następnym elementem aplikacji jest belka statusowa, która zawiera informacje dotyczące poziomu baterii, stanu sieci Wi-Fi, połączenia Bluetooth z zewnętrznym kontrolerem, prędkości drona jak i wysokości na jakiej się znajduje. W prawym górnym rogu widnieją ikony umożliwiające wykonywanie akcji związanych z wbudowaną kamerą drona. Jest to kolejno dostęp do galerii zawierającej przechwycone do tej pory materiały, gdzie można je następnie zapisać w pamięci telefonu. Kolejnym elementem jest przełącznik pomiędzy trybem nagrywania, a wykonywania zdjęć. Przycisk w prawym górnym rogu w zależności od aktualnego trybu rozpoczyna i zatrzymuje nagrywanie lub robi zdjęcie. W dolnym obszarze aplikacji znajdują się wspomniane już wcześniej drążki sterownicze. Alternatywnym sposobem sterowania jest wykorzystanie dedykowanego kontrolera GameSir T1D, który łączy się z aplikacją za pomocą Bluetooth. Wraz z nawiązaniem połączenia z kontrolerem drążki sterownicze znikają z aplikacji, a sterowanie jest realizowane zgodnie z schematem przedstawionym na rys. 3.12. Kontroler posiada uchwyt, który jest w stanie utrzymać urządzenia mobilne o różnym rozmiarze.



Rysunek 3.12 Kontroler Pad GameSir T1D

3.2 Software Development Kit

Producent drona DJI Tello Ryze udostępnia zestaw narzędzi (z ang. *Software Development Kit*, SDK), który pozwala użytkownikowi na interakcję ze statkiem powietrznym za pośrednictwem komend tekstowych. Połączenie z dronem jest realizowane poprzez port Wi-Fi UDP (z ang. *User Datagram Protocol*, UDP) [15]. UDP jest protokołem, który nie posiada mechanizmów gwarantujących dostarczenia danych [49]. Jego główną zaletą jest szybkość transmisji, która jest kluczowa w sterowaniu robotem powietrznym. Protokół ten nie wymaga potwierdzenia odbioru danych. SDK obsługuje trzy porty UDP na wcześniej zdefiniowanym adresie IP 192.168.10.1 (z ang. *Internet Protocol address*) [15]. Pierwszym z nich jest port 8889, którym przesyła się komendy wymuszające ruch drona lub ustawienie jego parametrów. Na porcie 8890 dron transmituje informacje o swoim statusie do użytkownika. Ostatni port 11111 odpowiada za przesyłanie obrazu z kamery drona.

Wśród komend wymuszających ruch drona znajdują się polecenia powodujące start urządzenia (z ang. *take off*), lądowanie (z ang. *land*), uruchomienie transmisji obrazu z kamery (z ang. *stream on*) jak i jej wyłączenie (z ang. *stream off*). Poza tym dostępne są komendy, które umożliwiają przemieszczenie drona w żądanym kierunku o zadaną odległość wyrażoną w centymetrach. Do komend, które umożliwiają edycję poszczególnych właściwości statku powietrznego, należy ustawianie prędkości w zakresie od 10 do 100 cm/s, modyfikacja parametrów sieci Wi-Fi w tym jej nazwy i hasła lub bezpośrednie wymuszenie przemieszczenia drona za pośrednictwem kontroli czterech kanałów. Składania takiej komendy przyjmuje postać *rc a b c d*, gdzie człon *rc* jest akronimem od zdalnej kontroli (z ang. *remote control*), a wstępujące za nim litery oznaczają odpowiadają kolejno przemieszczeniom wzdłuż osi *Y*, *X*, *Z* oraz obrotowi wokół osi *Z*. Nadawanie opisanych powyżej typów poleceń spowoduje wykonanie przez drona stowarzyszonej z nim akcji i tym samym wysłanie do użytkownika odpowiedzi w formie tekstowej o treści *ok*. W przypadku niepowodzenia wiadomością zwrotną jest *error*. Komendy dotyczące statusu drona umożliwiają sprawdzenie jego aktualnej prędkości, poziomu naładowania baterii lub czasu lotu. Dron posiada zabezpieczenie, które w przypadku braku jakiegokolwiek komendy przez 15 sekund, wymusi jego automatyczne lądowanie.

3.3 Biblioteka DJITelloPy

Biblioteka DJITelloPy została stworzona przez Damià Fuentes Escoté na ramach licencji otwartego oprogramowania MIT i udostępnia interfejs do obsługi drona DJI Ryze Tello w języku Python. Ten typ licencji umożliwia dowolne korzystanie z oprogramowania pod warunkiem umieszczenia jej treści w każdym pliku wykorzystującym jej możliwości. Biblioteka opera się na SDK dostarczonym przez producenta drona, który został opisany w rozdziale 3.2. Biblioteka zapewnia następujące funkcjonalności [17]:

- implementacja wszystkich komend z SDK,
- transmisja obrazu z kamery drona,
- odbieranie i przetwarzanie pakietów danych,
- sterowanie rojem dronów,
- wsparcie dla języka Python w wersji 3.6 i wyżej.

Bibliotekę można zainstalować przez dedykowany program do zarządzania pakietami i modułami dla języka Python – pip. Instalacja odbywa się poprzez wywołanie komendy `pip install djitellopy`. Dzięki temu użytkownik ma możliwość zaimportowania biblioteki do swojego projektu i korzystania z jej funkcjonalności. Paczka ta realizuje enkapsulację narzędzi udostępnianych przez producenta statku powietrznego i udostępnia je w postaci zdefiniowanych metod związanych z instancją drona. Autor paczki dostarcza jej pełną dokumentację oraz odpowiednie przykłady, które pozwalają zapoznać się z jej działaniem. Jeden z podstawowych przykładów został przedstawiony na listingu 3.1. W tym przykładzie następuje inicjalizacja obiektu klasy Tello oraz wywołanie metod pozwalających na nawiązanie komunikacji z dronem i wykonanie podstawowych akcji ruchu takich jak start, ruch w lewo, obrót, ruch do przodu i lądowanie. Przed wykonaniem tego programu konieczne jest wcześniejsze połączenie urządzenia, z którego będzie uruchamiany skrypt, z dronem za pośrednictwem sieci Wi-Fi. Biblioteka na bieżąco wyświetla na standardowym wyjściu wysyłane komendy do drona oraz otrzymane informacje zwrotne. Ta funkcjonalność pozwala na łatwe debugowanie programu jak i jego analizę działania.

```
from djitellopy import Tello

tello = Tello() # Utwórz instancję drona

tello.connect() # Nawiąż połączenie z dronem
tello.takeoff() # Wystartuj

tello.move_left(100) # Przenieść się dronem w lewo o 100 cm
tello.rotate_counter_clockwise(90) # Obróć się dronem o 90° w kierunku CCW
tello.move_forward(100) # Przenieść się dronem do przodu o 100 cm

tello.land() # Wyląduj
```

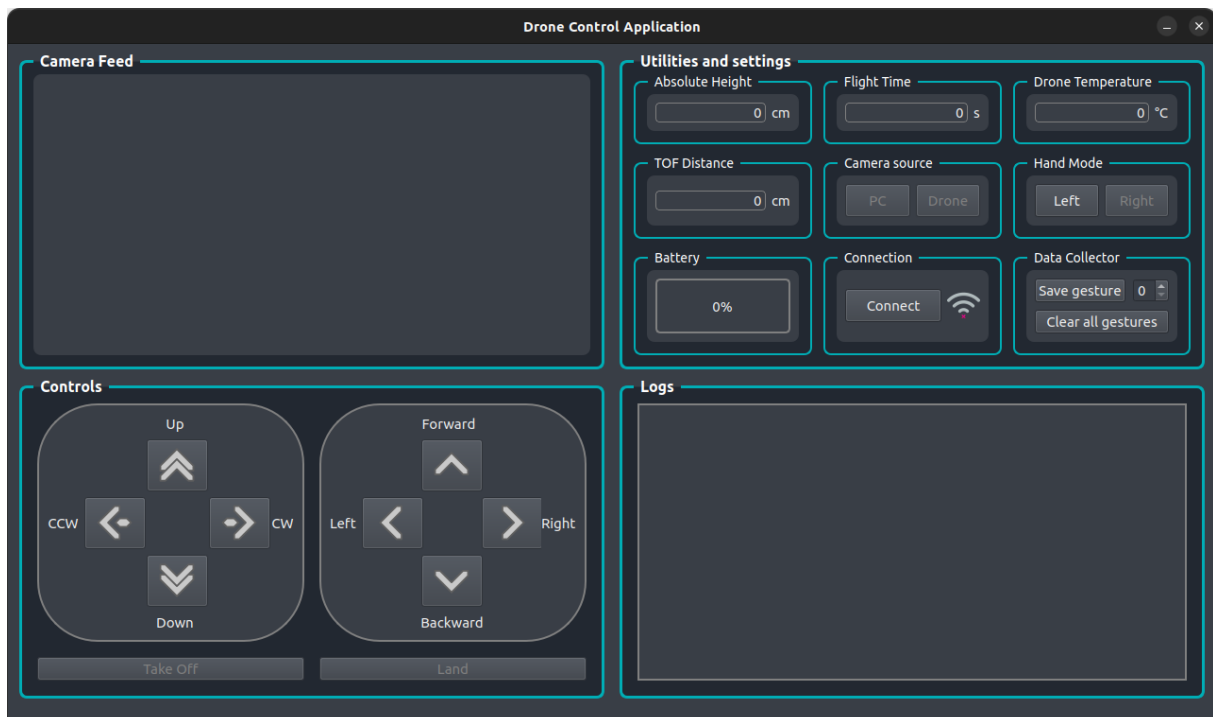
Listing 3.1: Przykładowy kod wykorzystujący bibliotekę DJITelloPy [17]

Rozdział 4

Projekt aplikacji

W poniższym rozdziale przedstawiono aplikację okienkową, którą wykonano w ramach realizacji niniejszej pracy inżynierskiej. Rozdział zawiera informacje o zastosowanych technologiach i rozwiązaniach, które pozwoliły na osiągnięcie założonego celu oraz opis funkcjonalności zaimplementowanych w aplikacji.

Do projektu aplikacji wykorzystano język Python oraz wieloplatformowe oprogramowanie służące do tworzenia graficznych interfejsów użytkownika – Qt. Kod programu był tworzony przy użyciu dedykowanego IDE (z ang. *Integrated Development Environment*) – Qt Creator, które zawiera zintegrowane środowisko przeznaczone do edycji i rozmieszczania graficznych elementów aplikacji. W związku z tym, że to środowisko jest domyślnie stosowane do tworzenia programów w języku C++ konieczne było zastosowanie modułu PySide6, który umożliwia kompilację pliku zawierającego informacje o strukturze interfejsu do pliku w języku Python. Tego samego zabiegu trzeba było dokonać w przypadku pliku, w którym dokonywano definicji wykorzystywanych zasobów w postaci zdjęć. Takim sposobem stworzono aplikację okienkową, która została zaprezentowana na rys. 4.1.



Rysunek 4.1 Interfejs graficzny użytkownika

Do stworzenia poszczególnych elementów aplikacji wykorzystano klasy udostępniane przez środowisko Qt. Następnie dokonano personalizacji wyglądu wybranych elementów za pośrednictwem definicji własnych arkuszy stylów (z ang. *style sheets*), które są tworzone w języku CSS (z ang. *Cascading Style Sheets*), co zostało pokazane na listingu 4.1.

```

QGroupBox#QGroupBox_Camera_Source{
    background-color: #222831; /* Kolor tła */
    border: 2px solid gray; /* Szerokość i kolor obramowania */
    border-color: #00ADB5; /* Kolor obramowania */
    border-radius: 8px; /* Promień zaokrąglenia obramowania */
    margin-top: 1ex; /* Górny margines */
}

```

Listing 4.1: Arkusz stylu dla elementu *QGroupBox_Camera_Source*

Okno programu zostało podzielone na cztery następujące części:

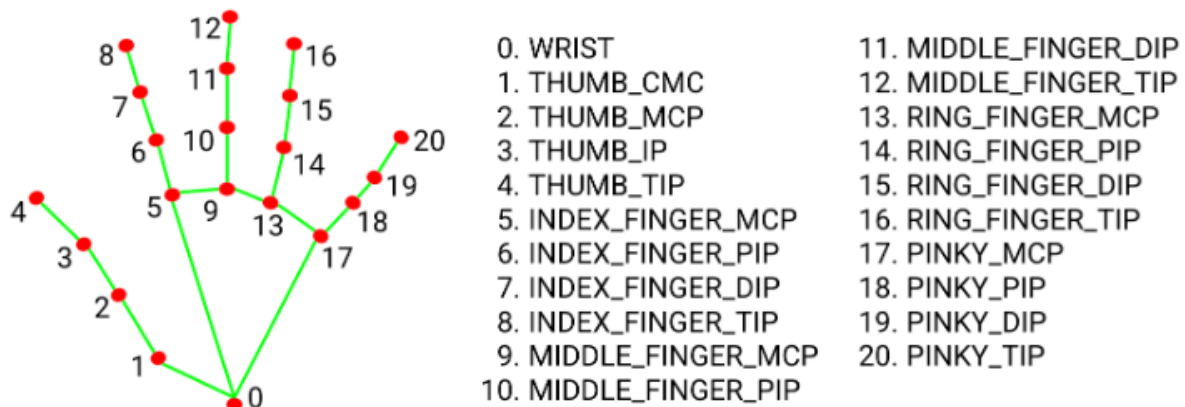
1. Transmisja z kamery (z ang. *camera feed*).
2. Narzędzia i ustawienia (z ang. *utilities and settings*).
3. Sterowanie (z ang. *controls*).
4. Logi (z ang. *logs*).

W pierwszej części jest widoczny obraz z kamery komputera lub po przełączeniu z wbudowanej kamery drona. W następnej sekcji znajdują się odczyty wartości z czujników zainstalowanych na pokładzie drona. Są to odpowiednio wysokość absolutna zwracana przez barometr wyrażona w centymetrach, czas lotu w sekundach, aktualna temperatura statku powietrznego, odległość mierzona przez dalmierz oraz procentowy stan naładowania baterii. Poza tym znajdują się tu przyciski, które umożliwiają użytkownikowi wybór trybu dla osoby praworęcznej lub leworęcznej, połączenie się z dronem jak i zapis współrzędnych gestów, co zostało szerzej omówione w rozdziałach 4.1 i 4.2. W części dotyczącej sterowania są ulokowane przyciski, które umożliwiają realizację procedury startu i lądowania drona jak i wykonanie konkretnych akcji ruchu. Nazwy poszczególnych akcji ruchu zostały umieszczone obok przycisków, które je wywołują. Ostatnią częścią jest okno logów, gdzie wyświetlają się komunikaty o aktualnym statusie danej akcji. Wszelkie funkcjonalności związane z obsługą drona zrealizowano za pośrednictwem biblioteki DJITelloPy opisanej w rozdziale 3.3, a zostały bardziej szczegółowo nakreślone w rozdziale 4.2.

Aplikacja była stworzona i testowana w systemie operacyjnym Ubuntu 22.04, który jest jedną z dostępnych dystrybucji Linuxa, dlatego instrukcja uruchomienia aplikacji zawarta w repozytorium odnosi się do tego właśnie systemu. Repozytorium z oprogramowaniem w przystępny sposób umożliwia użytkownikowi uruchomienie aplikacji na swoim urządzeniu [29]. Jednym z koniecznych wymagań jest instalacja pakietu języku Python w wersji 3.10.6 lub wyższej. Po pobraniu repozytorium zaleca się stworzenie środowiska wirtualnego dla języka Python (z ang. *Python Virtual Environment*), potocznie nazywanym *venv*. Pozwala to odseparować pakiety niezbędne do uruchomienia aplikacji od pozostałej części systemu. Następnie po aktywacji wirtualnego środowiska należy dokonać instalacji niezbędnych paczek przy pomocy programu *pip* oraz dostępnej listy wymaganych komponentów. Po zakończeniu tego procesu można uruchomić aplikację.

4.1 Rozpoznawanie gestów dłoni

Głównym aspektem niniejszej pracy inżynierskiej było rozpoznawanie gestów dłoni na podstawie obrazu otrzymywanego z kamery. Do realizacji tego celu zastosowano framework MediaPipe, który oferuje wieloplatformowe rozwiązania wykorzystujące uczenie maszynowe (z ang. *machine learning*, ML) do przetwarzania danych w postaci obrazu wideo [23]. Framework ten posiada moduł dedykowany do śledzenia dłoni i palców – MediaPipe Hands [24]. Wspomniany moduł za pośrednictwem uczenia maszynowego znajduje 21 trójwymiarowych punktów orientacyjnych dłoni na podstawie analizy zaledwie jednej klatki wideo [24]. Na rys. 4.2 widnieją poszczególne punkty nanoszone na dłoń.



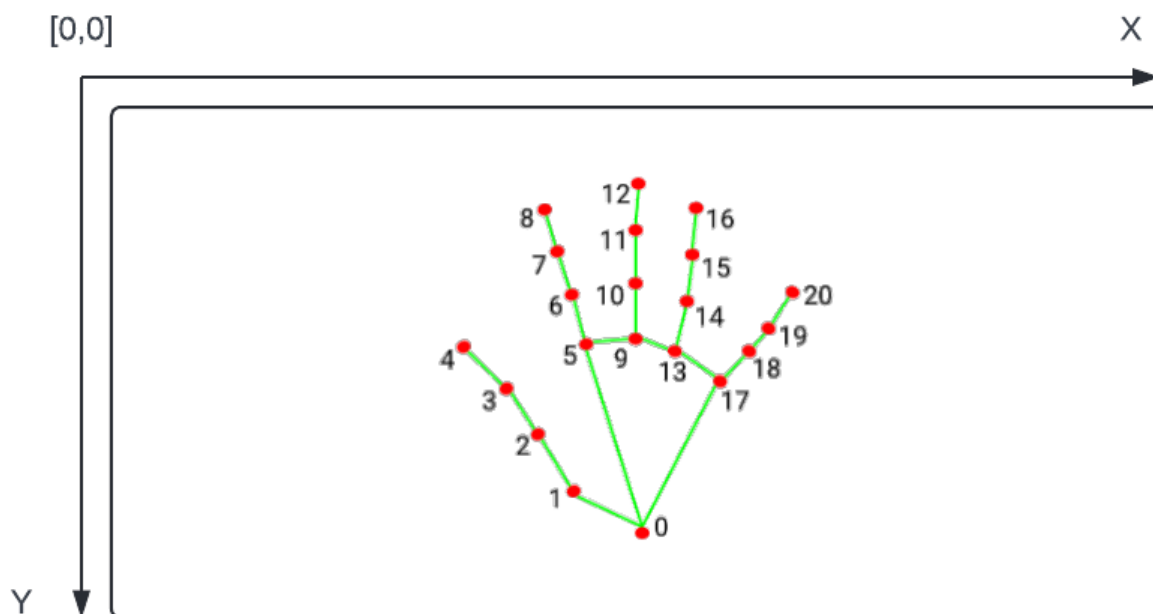
Rysunek 4.2 Punkty orientacyjne dłoni [24]

Pierwszym proponowanym rozwiązaniem w zakresie klasyfikacji gestów dłoni było stworzenie modelu uczenia maszynowego na podstawie dwuwymiarowych współrzędnych poszczególnych punktów orientacyjnych. W tym celu zaimplementowano narzędzie umożliwiające zapisywanie wspomnianych współrzędnych z poziomu aplikacji. Mowa tu o polu *Data Collector* znajdującym się w sekcji Narzędzia i ustawienia. Za punkt odniesienia przyjęto punkt usytuowany w nadgarstku – WRIST, którego współrzędnymi X i Y przypisano zerowe wartości. Pozostałe pozycje punktów orientacyjnych były obliczane względem nadgarstka. Wartości współrzędnych zostały odpowiednio przeskalowane do szerokości i wysokości obrazu. Tym sposobem otrzymano 42 wartości odpowiadające współrzędnym X i Y kolejnych punktów charakterystycznych dłoni. Na końcu takiego pakietu danych była umieszczana etykieta gestu. W celu przygotowania danych do nauki modelu dokonano ich normalizacji za pomocą maksymalnego skalowania bezwzględnego (z ang. *maximum absolute scaling*). Ta operacja spowodowała przeskalowanie każdej ze współrzędnych do zakresu od -1 do 1 zgodnie ze wzorem [1]

$$x_{scaled} = \frac{x}{\max(|x|)}, \quad (4.1)$$

gdzie x to początkowa wartość współrzędnej. Z tak przygotowanymi danymi dokonano próby stworzenia własnego modelu, który rozpoznawałby sygnalizowane gesty na podstawie ich współrzędnych położenia w przestrzeni obrazu z kamery. Budowa modelu była realizowana za pośrednictwem biblioteki TensorFlow, która udostępnia narzędzia do realizacji procesów związanych z uczeniem maszynowym. Niestety otrzymywane rezultaty nie były na tyle satysfakcjonujące, aby wykorzystać je w dalszej części projektu. W przypadku sterowania lotem drona wymagana była wysoka skuteczność rozpoznawania gestów.

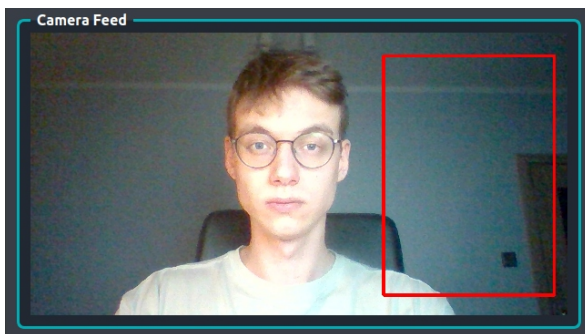
Finalne rozwiązanie rozpoznawania poszczególnych gestów zostało stworzone na podstawie zliczania aktualnie podniesionych palców dłoni użytkownika. Wykorzystując współrzędne punktów orientacyjnych ręki, zapewniane przez framework MediaPipe, zaimplementowano funkcje, która identyfikowała, który z palców jest wysunięty. Funkcja dla wszystkich palców z wyłączeniem kciuka sprawdza współrzędną Y ich czubków i porównuje ją z współrzędną Y punktu znajdującego się o dwie pozycje wcześniej niż czubek. Dla palca wskazującego są to punkty `INDEX_FINGER_TIP` i `INDEX_FINGER_PIP`. Biorąc pod uwagę usytuowanie układu współrzędnych stowarzyszonego z obrazem z kamery, który został przedstawiony na rys. 4.3, warunkiem podniesionego palca wskazującego jest sytuacja, gdy współrzędna Y punktu `INDEX_FINGER_TIP` ma mniejszą wartość od współrzędnej Y punktu `INDEX_FINGER_PIP`. To samo dotyczy pozostałych palców poza kciukiem. Do wykrycia wysuniętego kciuka funkcja porównuje ze sobą współrzędne X punktów `THUMB_TIP` i `THUMB_IP`. W trybie praworęcznym kciuk uznaje się za podniesiony, gdy współrzędna X jego czubka ma mniejszą wartość od współrzędnej X punktu `THUMB_IP`. W przypadku trybu leworęcznego następuje odwrócenie logiki wcześniej opisanego porównania. Wartością zwracaną przez funkcję jest pięcioelementowa lista, która opisuje stany poszczególnych palców począwszy od kciuka, a kończąc na małym palcu. Lista składa się z cyfr 0, które świadczą o opuszczonym palcu, lub cyfr 1, które sygnalizują podniesiony palec. Dla sytuacji, w której byłby podniesiony sam palec wskazujący lista przyjąłaby następującą postać $[0, 1, 0, 0, 0]$. Na takim zestawie danych można zdefiniować 2^5 konfiguracji dłoni. W ramach tego projektu wykorzystano pięć gestów, które następnie powiązano z wykonywaniem poszczególnych akcji ruchu przez drona, co zostało szerzej opisane w rozdziale 4.2. W celu identyfikacji gestów należy dokonać porównania listy zwracanej przez funkcję z zdefiniowaną wcześniej konfiguracją. Aplikacja na bieżąco jest w stanie rozpoznawać poszczególne gesty z kamery komputera lub drona, co umożliwia realizację głównego założenia tego projektu, czyli sterowania lotem drona za pośrednictwem gestów dłoni.



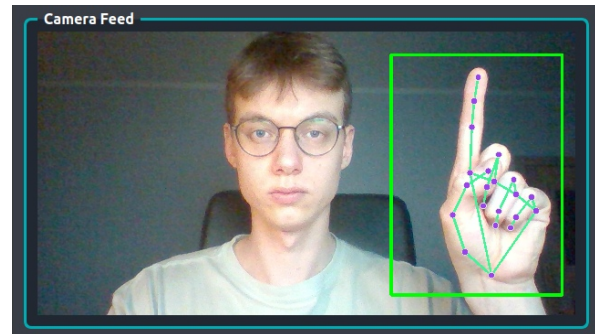
Rysunek 4.3 Układ współrzędnych wyświetlanego obrazu

4.2 Dostępne funkcjonalności

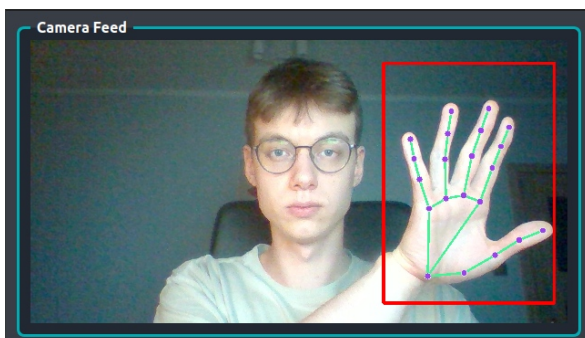
Wraz z uruchomieniem aplikacji w części *Camera Feed* pojawia się obraz z kamery komputera, co przedstawiono na rys. 4.4. W przypadku, gdy taka kamera nie zostanie wykryta pole to pozostanie puste. Rozdzielczość wyświetlanego obrazu wynosi 1280 na 720 pikseli. Prezentowany obraz bez względu na źródło jest poddawany operacji odwrócenia w poziomie. Już od początku działania programu na obrazie jest widoczny czerwony prostokąt, który sygnalizuje obszar, w którym należy umieścić dłoń w celu jej wykrycia. Domyślnie wyświetla się on po prawej stronie ekranu, co jest zaadaptowane dla praworęcznego użytkownika. Jednak aplikacja jest również dostosowana do potrzeb osób leworęcznych, gdyż w każdej chwili pozycja tego obszaru może zostać przeniesiona na lewą stronę obrazu za pośrednictwem przycisku *Left* znajdującego się w polu *Hand Mode* w części *Utilities and settings*. Efekt aktywowania tego przycisku zaprezentowano na rys. 4.7. W przypadku, gdy użytkownik umieści swoją dłoń w przestrzeni wyznaczonej przez prostokąt to podświetli się on na zielono, a na dłoni dzięki frameworkowi MediaPipe opisanemu w rozdziale 4.1 zostaną wyświetlone charakterystyczne punkty połączone linią, co jest widoczne na rys. 4.5. Gdy w trybie dedykowanym dla osoby praworęcznej w wyznaczonym obszarze zostanie umieszczona lewa dłoń to nie nastąpi zmiana koloru prostokąta, a co za tym idzie nie wystąpi dalsza analiza uformowanego gestu. Opisywaną sytuację przedstawiono na rys. 4.6. Program zareaguje w analogiczny sposób, gdy w trybie dla osoby leworęcznej użytkownik umieści w prostokącie prawą dłoń.



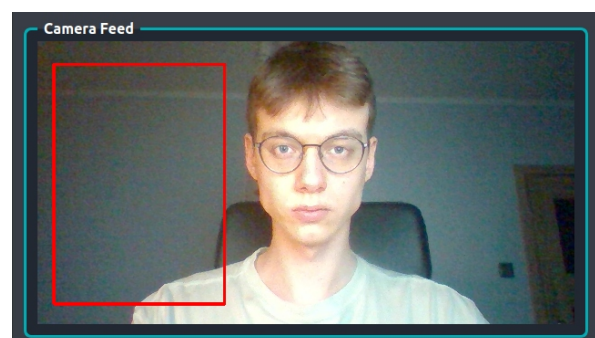
Rysunek 4.4 Brak dłoni



Rysunek 4.5 Wykryta dłoń

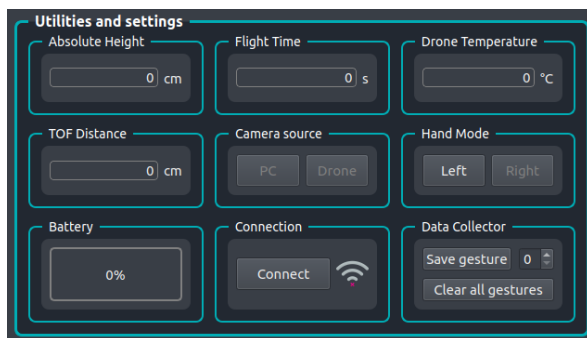


Rysunek 4.6 Nieprawidłowa dłoń

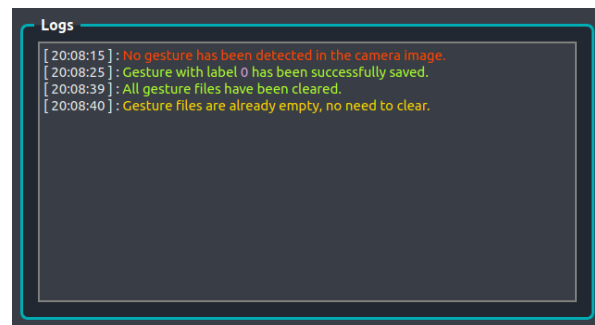


Rysunek 4.7 Tryb leworęczny

W tym momencie użytkownik jest w stanie skorzystać z funkcji pozwalającej na zapis współrzędnych punktów występujących na dłoni wykrytej w wyznaczonym obszarze. Jest ona dostępna jedynie w trybie, gdzie źródłem obrazu jest kamera komputera. Operacje związane z tą opcją można wykonać za pośrednictwem przycisków znajdujących się w polu *Data Collector* w oknie *Utilities and settings*. Zbliżony widok na wspomniane okno przedstawiono na rys. 4.8. Obok przycisku *Save gesture* znajduje się pole umożliwiające wybór identyfikatora zapisywanego gestu w zakresie od 0 do 9. W przypadku wciśnięcia wspomnianego wcześniej przycisku w momencie, gdy dłoń jest poza wyznaczonym obszarem w postaci prostokąta, wyświetli się komunikat informujący użytkownika o niepomyślnym wykryciu dłoni. Jeżeli dłoń znajduje się w prawidłowym obszarze współrzędne gestu zostaną zapisane pomyślnie do pliku o rozszerzeniu CSV, co również zostanie zasygnalizowane w logach aplikacji. Istnieje także możliwość usunięcia wszystkich zapisanych współrzędnych z danego pliku. Można to wykonać za pośrednictwem przycisku *Clear all gestures*. Komunikat o pomyślnym usunięciu danych jest wypisywany w sekcji dotyczącej logów aplikacji. Gdy użytkownik będzie chciał usunąć wpisy z pustego już pliku zostanie o tym poinformowany odpowiednim komunikatem. Wspomniane wcześniej wiadomości informacyjne zostały zaprezentowane na rys. 4.9. W finalnej wersji aplikacji ta funkcjonalność nie posiada szerszego zastosowania, co zostało opisane w rozdziale 4.1.

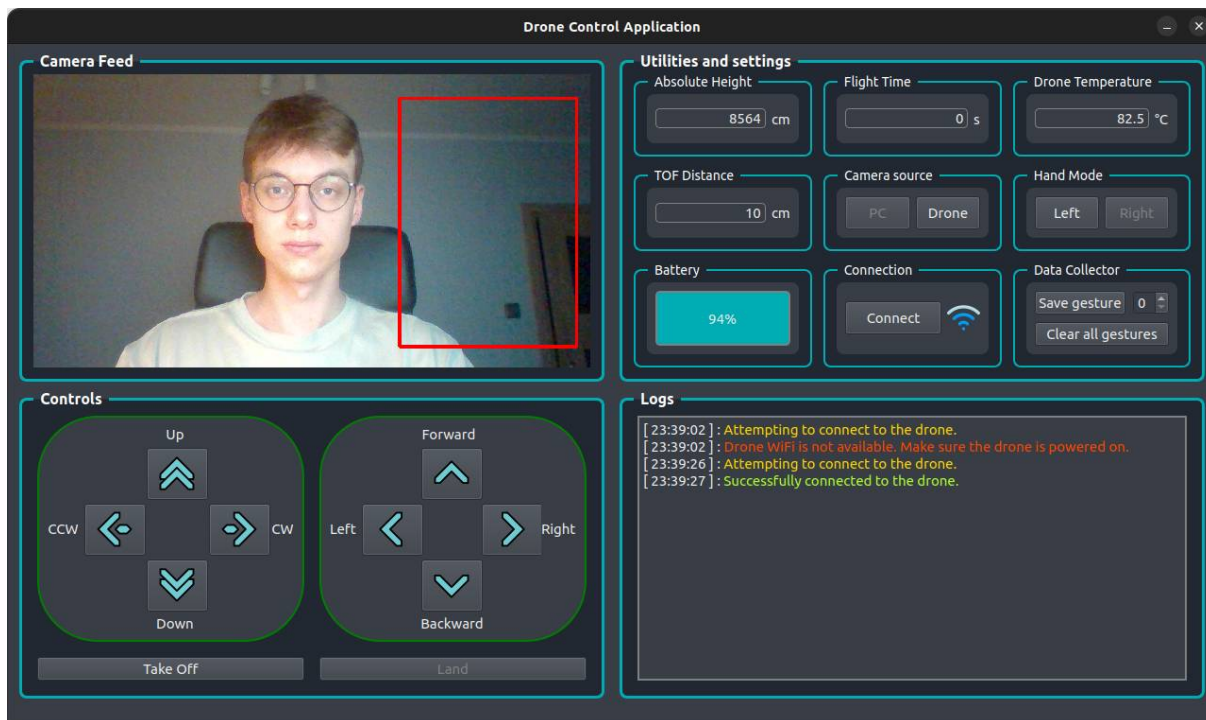


Rysunek 4.8 Narzędzia i ustawienia



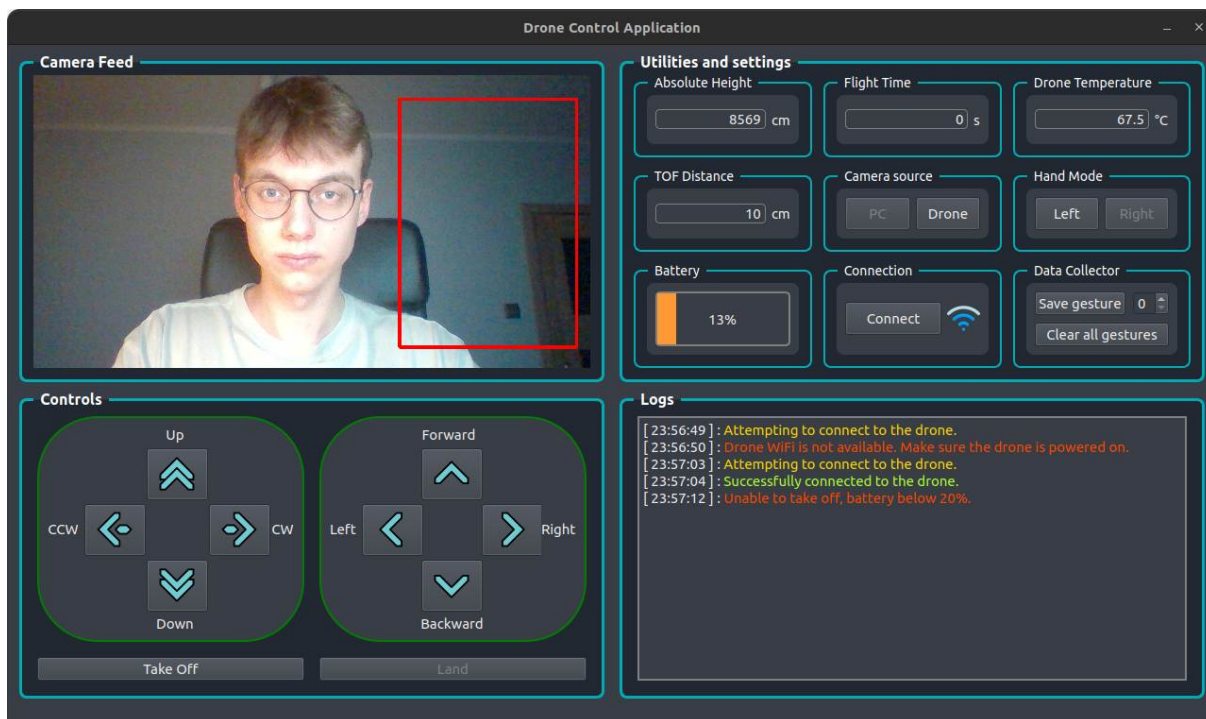
Rysunek 4.9 Logi

Kolejne funkcje są bezpośrednio związane z kontrolą i monitorowaniem stanu drona DJI Ryze Tello. Domyślnie są one niedostępne, dlatego w celu ich odblokowania należy nawiązać komunikację ze statkiem powietrznym. Połączenie z dronem odbywa się za pośrednictwem przycisku *Connect* w polu *Connection*, który jest widoczny na rys. 4.8. Po wciśnięciu przycisku w sekcji z logami pojawia się komunikat o podjęciu próby połączenia z robotem. W sytuacji, gdy program nie znalazł sieci Wi-Fi stowarzyszonej z dronem pojawia się informacja o niepowodzeniu i sugestia w stronę użytkownika, aby upewnił się, czy dron jest na pewno włączony. Po pomyślnym połączeniu z dronem wyświetla się komunikat o udanej próbie. Obok przycisku odpowiadającego za nawiązanie połączenia znajduje się ikonka, która wizualizuje moc sygnału sieci Wi-Fi. Wraz z zmniejszającą się mocą sygnału maleje liczba wypełnionych kreski ikonki. Łączenie z siecią drona jak i monitorowanie jej statusu odbywa się za pomocą narzędzia o nazwie *nmcli*, które służy do kontrolowania menadżera sieci systemu poprzez linię poleceń [13]. Wraz z nawiązaniem połączenia z dronem stają się aktywne przyciski w sekcji *Controls*, a w narzędziach i ustawieniach na bieżąco zaczynają aktualizować się wartości z czujników oraz informacja o procentowym naładowaniu baterii drona. Proces nawiązywania połączenia jak i stan interfejsu bezpośrednio po jego pomyślnym przebiegu zaprezentowano na rys. 4.10.



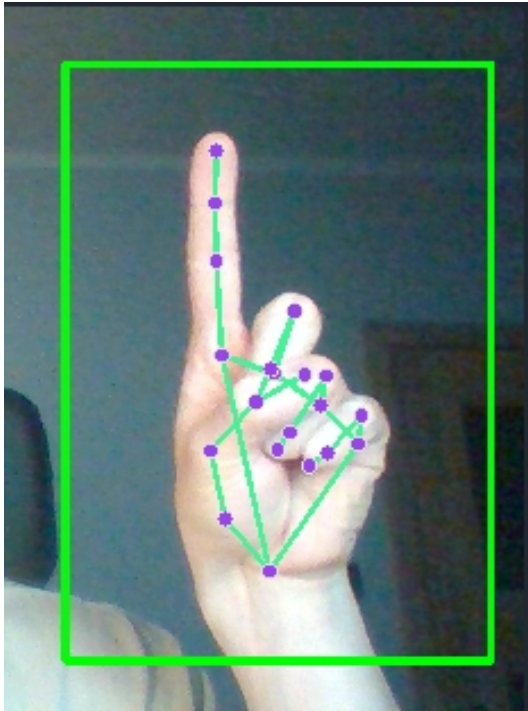
Rysunek 4.10 Stan interfejsu graficznego użytkownika po nawiązaniu połączenia

Na tym etapie użytkownik może przejść do rozpoczęcia lotu drona. W tym celu za pomocą przycisku *Take Off* należy uruchomić procedurę startu robota. Start jest możliwy, gdy poziom baterii urządzenia jest wyższy lub równy 20%. To ograniczenie wynika z kwestii bezpieczeństwa, jest ono również obecne w mobilnej aplikacji Tello dedykowanej do obsługi drona DJI Ryze. Opisana sytuacja została przedstawiona na rys. 4.11.



Rysunek 4.11 Uniemożliwienie startu przy zbyt niskim poziomie baterii

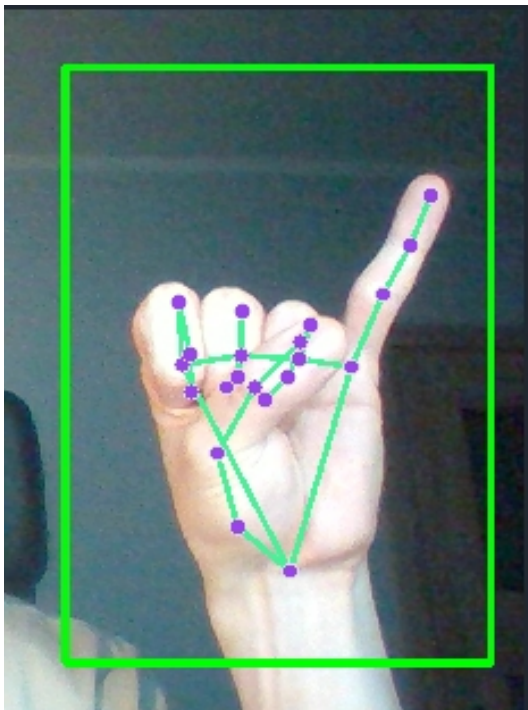
Po pomyślnym starcie robota użytkownik ma możliwość sterowania jego lotem za pośrednictwem przycisków znajdujących się w sekcji *Controls*. Umieszczając dłoń w wyznaczonym obszarze użytkownik może sterować lotem drona przy użyciu odpowiednich gestów dłoni. Obsługiwane gesty przedstawiono kolejno na rys. 4.12 ÷ 4.15.



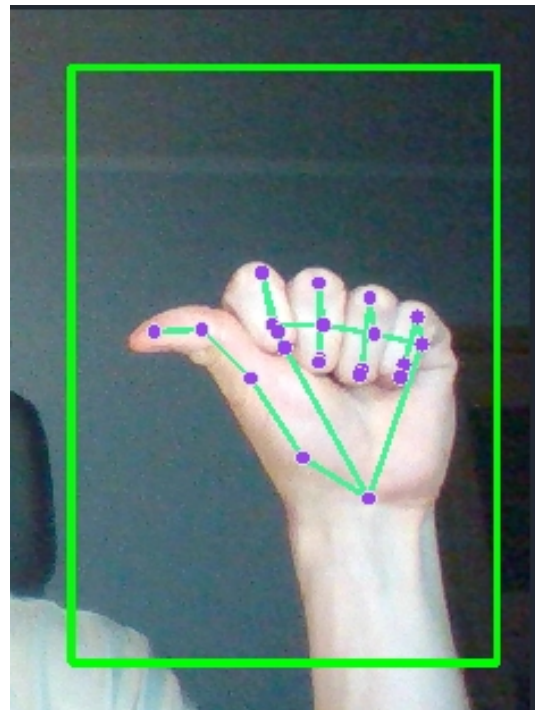
Rysunek 4.12 Ruch w górę



Rysunek 4.13 Ruch w dół

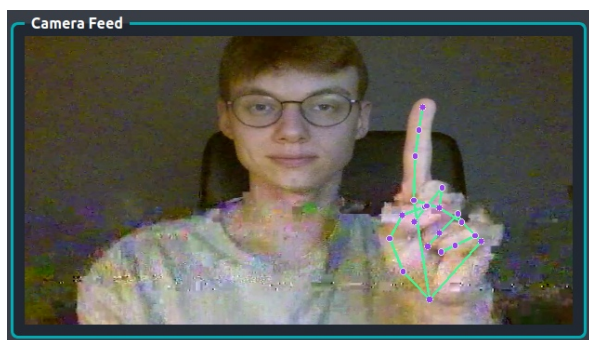


Rysunek 4.14 Ruch w prawo w trybie praworęcznym

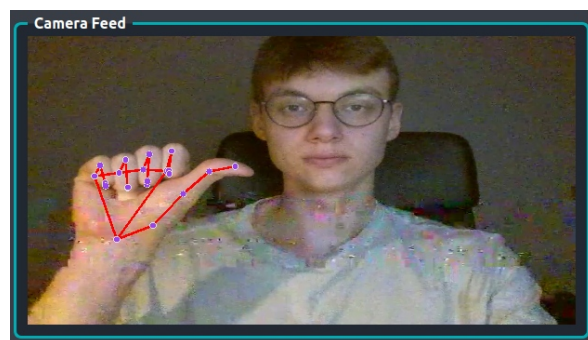


Rysunek 4.15 Ruch w lewo w trybie praworęcznym

Poza gestami wymienionymi powyżej rozpoznawany jest również gest otwartej dłoni, który powoduje zatrzymanie drona. W kwestii trybu leworęcznego wysunięcie kciuka powoduje lot w prawą stronę, a mały palec odpowiada za lot w lewą stronę. Użytkownik może dokonać zmiany źródła obrazu na kamerę drona za pośrednictwem przycisku *Drone* w polu *Camera source*. W tym trybie poszczególne akcje ruchu robota powodują te same gesty, które zostały opisane powyżej. Gdy źródłem obrazu jest wbudowana kamera drona to gesty są rozpoznawane w całym obszarze obrazu, nie występuje wyznaczone pole, gdzie trzeba umieścić dłoń. Wynika to z możliwości wystąpienia potencjalnych trudności z usytuowaniem dłoni w odpowiednim miejscu podczas przemieszczania się statku. W przypadku, gdy obraz jest transmitowany z kamery komputera taki problem nie występuje, gdyż źródło obrazu nie zmienia swojego położenia. Gdy wideo jest odbierane z kamery drona również jest obecna obsługa trybów dla użytkowników praworęcznych i leworęcznych. Zmiana trybu następuje poprzez wciśnięcie jednego z aktywnych przycisków w polu *Hand Mode*. W obu z tych trybów w celu wywołania ruchu drona konieczne jest użycie prawidłowej dłoni to znaczy zgodnej z wybranym trybem. Gdy w obrębie obrazu użytkownik umieści poprawną dłoń linie łączące charakterystyczne punkty przyjmują kolor zielony, co zostało przedstawione na rys. 4.16 i 4.18. W momencie usytuowania dłoni niezgodnej z wybranym trybem linie łączące charakterystyczne punkty przyjmą kolor czerwony, co jest widoczne na rys. 4.17 i 4.19 i nie zostanie wykonana żadna akcja ruchu. W celu zakończenia lotu i wylądowania dronem należy wcisnąć przycisk *Land* z sekcji *Controls*.



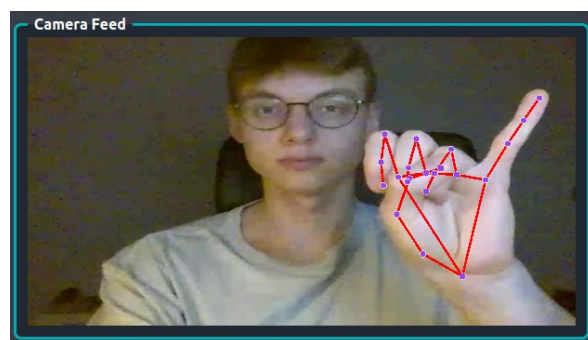
Rysunek 4.16 Prawidłowa dłoń w trybie praworęcznym



Rysunek 4.17 Nieprawidłowa dłoń w trybie praworęcznym

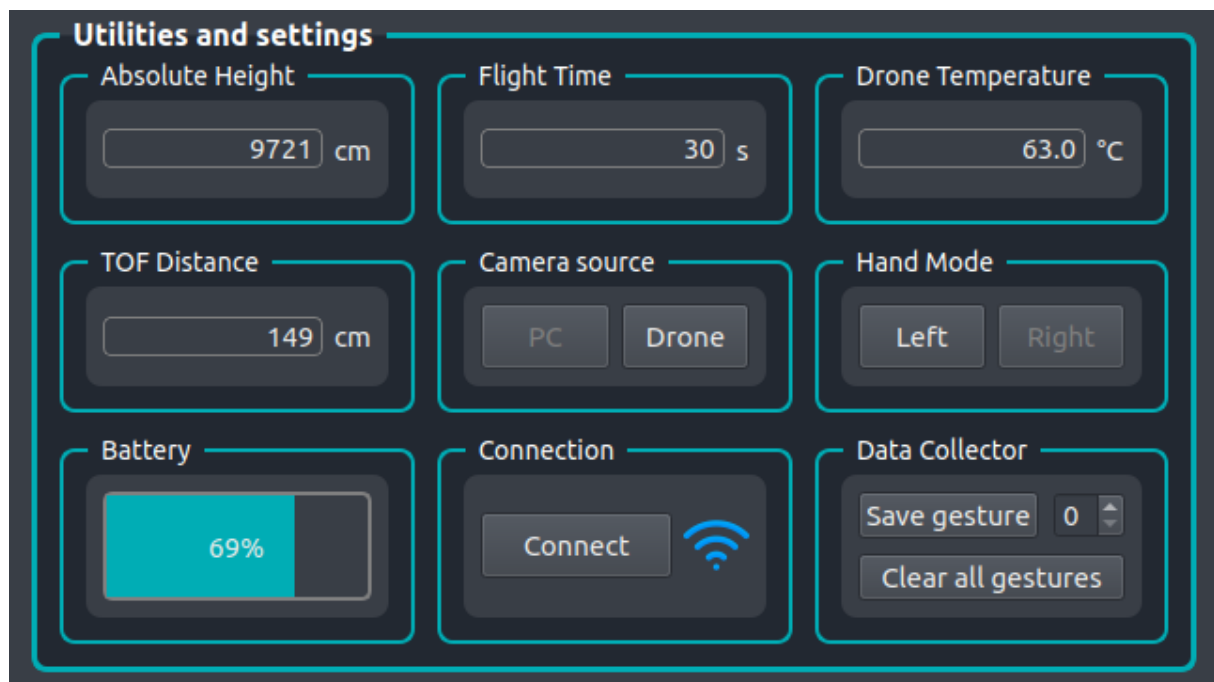


Rysunek 4.18 Prawidłowa dłoń w trybie leworęcznym



Rysunek 4.19 Nieprawidłowa dłoń w trybie praworęcznym

W oknie dotyczącym narzędzi i ustawień poza omówionym już komponentami znajdują się pola wyświetlające wartości z wybranych czujników drona, które zostały pośrednio omówione przy opisie poszczególnych sekcji aplikacji. Wartość w polu *Absolute Height* jest pobierana z barometru zamontowanego na pokładzie drona. Według informacji zawartych w dokumentacji biblioteki DJITelloPy zwracana wartość reprezentuje bezwzględną wysokość wyrażoną w centymetrach, na jakiej znajduje się statek powietrzny [17]. Niestety nie została podana informacja, jaka jest wartość ciśnienie referencyjnego dla zastosowanego barometru, co uniemożliwia weryfikację wskazywanego wyniku. Niemniej jednak barometr prawidłowo reaguje na zmianę wysokości lotu drona. Kolejną wartością jest czas lotu drona, a dokładniej czas przez jaki były aktywne napędy statku. Sekundy naliczane są odpowiednio po starcie urządzenia. Wartość ta nie ulega wyzerowaniu po wylądowaniu dronem, lecz po ponownym nawiązaniu połączenia. Tuż obok znajduje się pole wyświetlające aktualną temperaturę drona. Przed startem waha się ona w okolicach 45 °C, a podczas lotu przekracza nawet 90 °C. Ostatni pomiar widoczny w polu *TOF Distance* jest realizowany przez dalmierz umieszczony w podwoziu statku powietrznego. Zwraca on odległość drona względem powierzchni znajdujących się bezpośrednio pod nim wyrażoną w centymetrach. W przypadku tego sensora udało się zweryfikować poprawność zwracanych przez niego wartości, za pośrednictwem manualnego pomiaru odległości statku od powierzchni i porównaniu go z wyświetlanym w aplikacji.

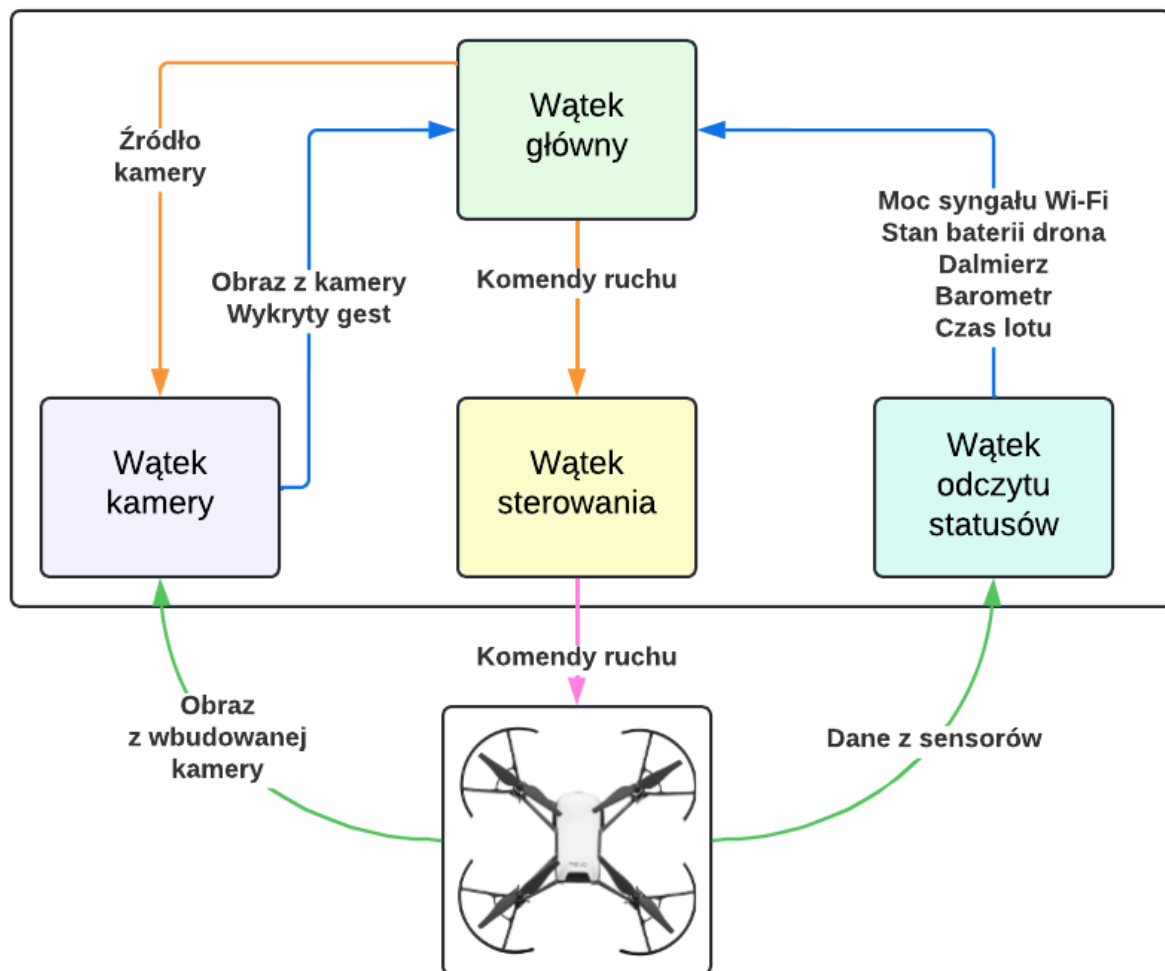


Rysunek 4.20 Odczyty z sensorów drona

4.3 Architektura systemu

Działanie aplikacji opiera się na mechanizmie sygnałów i slotów udostępnianym przez środowisko Qt. Ze względu na konieczność wykonywania się poszczególnych akcji równoległe wykorzystano klasę QThread, która pozwala na implementację wątków w języku Python. Program posiada cztery wątki. Są to odpowiednio wątek główny, wątek kamery, wątek sterowania oraz wątek odczytu statusów. Wątek główny odpowiada za aktualizowa-

nie elementów znajdujących się w obrębie interfejsu i wykonywanie stowarzyszonych z nimi akcji. Do elementów interfejsu zalicza się wszelkiego rodzaju przyciski, pola tekstowe lub paski postępu (z ang. *progress bar*). Wątek główny odbiera dane z wątku obsługującego kamerę oraz odczytu statusów. Następnie wykorzystując otrzymane informacje aktualizuje związane z nimi elementy. Interfejs kamery jest uruchamiany wraz z włączeniem programu i odpowiada za wysyłanie obrazu oraz rozpoznanego gestu do głównego wątku aplikacji. W momencie otrzymania sygnału o zmianie źródła obrazu od głównego wątku następuje zakończenie pracy wątku kamery i jego ponowne uruchomienie z uwzględnieniem wybranego źródła wideo. Wątek sterowania startuje wraz z pomyślnym nawiązaniem połączenia z dronem. Następnie odbiera on komendy ruchu od wątku głównego i wysyła je bezpośrednio do statku powietrznego. Komendy ruchu są wysyłane z wątku głównego w momencie wykrycia gestu dłoni lub wciśnięcia przycisku związanego ze sterowaniem. Ostatnim wątkiem jest wątek odczytu statusów, który przekazuje do wątku głównego zestaw informacji. Wśród nich znajduje się moc sygnału Wi-Fi, procentowy stan naładowania baterii drona, wartości zwracane przez dalmierz i barometr oraz czas trwania lotu. Strukturę całego systemu i opisane powyżej funkcjonalności zobrazowano na rys. 4.21.



Rysunek 4.21 Diagram architektury systemu

Rozdział 5

Podsumowanie

Celem niniejszej pracy inżynierskiej było stworzenie aplikacji, która umożliwiłaby sterowanie dronem za pomocą gestów dłoni. Rozpoznawanie gestów zostało zrealizowane poprzez przetwarzanie obrazu z wbudowanej kamery drona, ale także została zaimplementowana opcja wyboru kamery komputera. Procedura rozpoznawania poszczególnych gestów dłoni została wykonana za pośrednictwem zastosowania frameworku MediaPipe Hands, który udostępnia interfejs uczenia maszynowego. Wykrycie wcześniej zdefiniowanych gestów skutkuje wykonaniem przez drona odpowiednich akcji ruchu. Dodatkowo aplikacja została wyposażona w interfejs umożliwiający sterowanie statkiem powietrznym za pomocą przycisków oraz w elementy wyświetlające wybrane parametry drona.

Podczas realizacji aplikacji zmierzono się z problemami jakie powstały w trakcie implementacji procedury rozpoznawania poszczególnych gestów dłoni. Finalnie pierwsze podejście polegające na stworzeniu własnego modelu uczenia maszynowego okazało mniej efektywne niż przypuszczano. Gesty nie były jednoznacznie identyfikowane, co było niedopuszczalne w aspekcie sterowania robotem powietrznym. W związku z tym zdecydowano się na wykorzystanie zależności pomiędzy charakterystycznymi punktami na dłoni i identyfikowanie poszczególnych gestów na podstawie aktualnie podniesionych placów dłoni. To rozwiązanie pozwoliło na płynne rozpoznawanie gestów, a co za tym idzie umożliwiło realizację głównego celu projektu, czyli sterowanie dronem za pomocą gestów dłoni. Podczas sterowania robotem występowały trudności z jego stabilizacją, lotem stacjonarnym w powietrzu. Trudności te wynikały z charakterystyki jednego z systemów obecnych w dronie DJI Ryze Tello. Mowa tu o systemie wizyjnego pozycjonowania, który jest niezwykle wrażliwy na brak odpowiednich warunków oświetleniowych. Zbyt małe natężenie światła zaburzało działanie tego systemu i powodowało brak stabilizacji w powietrzu. W związku z tym loty testowe musiały być wykonywane w sprzyjających warunkach oświetleniowych.

Pomimo napotkanych trudności udało się osiągnąć zamierzony cel projektu. W ramach rozwoju aplikacji, która powstała podczas realizacji tej pracy, można zaproponować kilka usprawnień. Jednym z nich byłoby zdefiniowanie większej liczby rozpoznawanych gestów, a co za tym idzie przypisanie im pozostałych dostępnych akcji ruchu. Dodatkowo można zwiększyć poziom pewności wykrywanych gestów, a co za tym idzie bezpieczeństwo aplikacji poprzez dodanie specjalnego obszaru usytuowanego obok twarzy użytkownika w trybie, gdzie źródłem obrazu jest kamera drona. Podobna funkcjonalność jest dostępna w trybie kamery z komputera, jednak w przypadku dynamicznie zmieniającego się obrazu z kamery drona, wymagałoby to implementacji rozpoznawania twarzy i wyrysowywania wydzielonego obszaru. Dodatkowo można rozszerzyć interfejs użytkownika o informacje o pozostałych parametrach drona, które udostępnia zestaw narzędzi SDK. Aplikacja mogłaby zostać rozwinięta o możliwość wykonywania zdjęć i nagrywania filmów kamerą drona.

Załącznik A

Do pracy załączono płytę DVD zawierającą w poszczególnych katalogach:

/Praca_inzynierska.pdf — wersja cyfrowa pracy,

/Kod_zrodlowy — kod źródłowy aplikacji.

Literatura

- [1] A. I. Moreno. Data normalization with Pandas and Scikit-Learn. <https://towardsdatascience.com/data-normalization-with-pandas-and-scikit-learn-7c1cc6ed6475>, 2020.
- [2] M. F. Ahmed, M. N. Zafar, J. C. Mohanta. Modeling and Analysis of Quadcopter F450 Frame. *2020 International Conference on Contemporary Computing and Applications (IC3A)*, strony 196–201, 2020.
- [3] M. Altin Karataş, H. Gökkaya. A review on machinability of carbon fiber reinforced polymer (CFRP) and glass fiber reinforced polymer (GFRP) composite materials. *Defence Technology*, 14(4):318–326, 2018.
- [4] Amazon. RCGEEK Drone 8520 Motors Coreless CW CCW Motor Compatible with DJI Ryze Tello 720P Camera FPV RC Quadcopter, 2 Pairs. <https://www.amazon.com/RCGEEK-Motors-Coreless-Compatible-Quadcopter/dp/B07CPT5TSL/>, 2022.
- [5] Autonomian. UAV Data Transmission and Protocols. <https://robo-labor.ee/img/cms/projektid/UAV%20Data%20Transmission%20and%20Communication%20Protocols.pdf>, 2018.
- [6] Baisali. How Firefighters Can Better Manage Emergency Situations Using Drones. <https://www.flytnow.com/blog/drone-fire-fighting>, 2020.
- [7] Banggod. 4X Racerstar Racing Edition 2205 BR2205 2300KV 2-4S Brushless Motor. <https://www.banggood.com/4X-Racerstar-Racing-Edition-2205-BR2205-2300KV-2-4S-Brushless-Motor-For-QAV250-ZMR250-RC-Drone-FPV-Racing-p-1066837.html>, 2022.
- [8] Botland. Pakiet Li-Pol Redox 2200mAh 20C 3S 11,1V. <https://botland.com.pl/akumulatory-li-pol-3s-111v-/8474-pakiet-li-pol-redox-2200mah-20c-3s-111v-5903754000713.html>, 2022.
- [9] Botland. Sterownik silnika bezszczotkowego Skywalker 50A ESC UBEC 2-4S. <https://botland.com.pl/sterowniki-silnikow-bezszczotkowych-bldc/6108-sterownik-silnika-bezszczotkowego-skywalker-50a-esc-ubec-2-4s-6938994402088.html>, 2022.
- [10] S. Captain. Firefighting drones extinguish 10-story blaze in China demonstration. <https://dronedj.com/2020/04/03/firefighting-drones-extinguish-10-story-blaze-china/>, 2020.
- [11] L. Centeno. How Drone Vision Systems Allow Drones to Fly in Low-Light Conditions. <https://www.immervision.com/drone-navigation-system/>, 2022.

- [12] Z. Czyż, P. Karpiński, K. Skiba, M. Wendeker. Wind tunnel performance tests of the propellers with different pitch for the electric propulsion system. *Sensors*, 22(1):2, Dec 2021.
- [13] G. Developer. nmcli. <https://developer-old.gnome.org/NetworkManager/stable/nmcli.html>, 2016.
- [14] Devopedia. Quadcopter. <https://devopedia.org/quadcopter>, 2022.
- [15] DJI. SDK 2.0 User Guide. <https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20SDK%202.0%20User%20Guide.pdf>, 2018.
- [16] DJI. Tello User Manual v1.4. <https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20User%20Manual%20v1.4.pdf>, 2018.
- [17] D. F. Escoté. DJITelloPy. <https://github.com/damiafuentes/DJITelloPy>, 2018.
- [18] A. Farhan, F. Bazzaz, O. Dallal Bashi, M. Al-Dabagh. IoT Based UAV Platform for Far Distance control system. 2020.
- [19] J. Flynt. Types of Drone Propellers: Everything You Need to Know. <https://3dsider.com/types-of-drone-propellers/>, 2020.
- [20] FORBOT. Kurs elektroniki II – #7 – nadawanie i odbieranie podczerwieni. <https://forbot.pl/blog/kurs-elektroniki-ii-scalony-odbiornik-podczerwien-tsop-id9641>, 2022.
- [21] N. Gio, R. Brisco, T. Vuletic. Control of a drone with body gestures. *Proceedings of the Design Society*, 1:761–770, 2021.
- [22] D. D. Gods. REDBULL MOST NOTORIOUS - WORLDS BEST FPV DRONE SHOT? <https://dutchdronegods.com/projects/best-fpv-drone-shot-in-the-world-redbull-valparaiso-mtb-tomas-slavic>, 2022.
- [23] GOOGLE LLC. MediaPipe. <https://google.github.io/mediapipe/>, 2020.
- [24] GOOGLE LLC. MediaPipe Hands. <https://google.github.io/mediapipe/solutions/hands.html>, 2020.
- [25] M. Hassanalain, A. Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91:99–131, 2017.
- [26] HEAVY.AI. Hardware Acceleration. <https://www.heavy.ai/technical-glossary/hardware-acceleration>, 2019.
- [27] Intel. Connecting the Pixhawk Flight Controller to the Intel[®] Aero Compute Board. <https://www.intel.com/content/dam/support/us/en/documents/boardsandkits/aero/apu-170405-pixhawk-flight-guide.pdf>, 2017.
- [28] T. W. S. JOURNAL. UPS Gets FAA Nod for Widespread Drone Deliveries. <https://www.wsj.com/articles/ups-gets-faa-nod-for-widespread-drone-deliveries-11569939601>, 2019.

- [29] M. Kaniewski. Drone-control-with-hand-gestures-using-image-processing. <https://github.com/maciejkaniewski/Drone-control-with-hand-gestures-using-image-processing>, 2022.
- [30] O. Liang. Quadcopter Hardware Overview – Every Component Explained. <https://oscarliang.com/quadcopter-hardware-overview/>, 2018.
- [31] O. Liang. Everything About LiPo Battery for Racing Drones. <https://oscarliang.com/lipo-battery-guide/>, 2019.
- [32] O. Liang. FPV Drone Flight Controller Explained. <https://oscarliang.com/flight-controller-explained/>, 2020.
- [33] O. Liang. The Best Radio Transmitter for FPV Drones. <https://oscarliang.com/radio-transmitter/>, 2022.
- [34] MEGADRON. Śmigła DJI Tello 3044. <https://megadron.pl/pl/products/smigla-dji-tello-3044-8651.html>, 2022.
- [35] Miltonrodney. Tello drone photographed in flight in nature. https://img.freepik.com/premium-photo/araras-sao-paulo-brazil-september-03-2021-tello-drone-photographed-flight-nature-natural-light-small-depth-field-selective-focus_309761-1776.jpg?w=1380, 2021.
- [36] S. Musa. Techniques for Quadcopter Modelling & Design: A Review. *Journal of Unmanned System Technology*, 5:66–75, 2018.
- [37] T. Müezzinoğlu, M. Karaköse. Wearable Glove Based Approach for Human-UAV Interaction. *IEEE International Symposium on Systems Engineering*, strony 1–6, Vienna, Austria, 2020.
- [38] D. Nedelkovski. Arduino Brushless Motor Control Tutorial. <https://howtomechanics.com/tutorials/arduino/arduino-brushless-motor-control-tutorial-esc-bldc/>, 2019.
- [39] S. Pikalov, E. Azaria, S. Sonnenberg, B. Ben-Moshe, A. Azaria. Vision-less sensing for autonomous micro-drones. *Sensors*, 21(16), 2021.
- [40] P. Posea. How to Choose a Propeller for Your Drone. <https://dronesgator.com/how-to-choose-a-drone-propeller/>, 2022.
- [41] RadioMaster. RP1 ExpressLRS 2.4GHz Nano Receiver. <https://www.radiomasterrc.com/products/rp1-expresslrs-2-4ghz-nano-receiver>, 2022.
- [42] RadioMaster. TX16S Mark II Radio Controller (M2). <https://www.radiomasterrc.com/products/tx16s-mark-ii-radio-controller>, 2022.
- [43] J. Rodriguez-Galvis, V. Angulo-Morales, E. Gaona-García, I. Lizarazo-Salcedo. Development of Low-Cost Ground Control System for UAV-Based Mapping. *IEEE International Geoscience and Remote Sensing Symposium*, strony 5111–5114, Waikoloa, HI, USA, 2020.
- [44] Ryze. Cool New Tello Toy Drone Soars into CES 2018. https://www.ryzerobotics.com/news/launch_news_page_en, 2018.

- [45] L. Shi, N. J. H. Marcano, R. H. Jacobsen. A review on communication protocols for autonomous unmanned aerial vehicles for inspection application. *Microprocessors and Microsystems*, 86:104340, 2021.
- [46] R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. *Introduction to Autonomous Mobile Robots, 2nd edition*. The MIT Press, 2011.
- [47] I. Singh. DJI releases new night vision sensor for M300 RTK drone. <https://dronedj.com/2022/03/21/night-vision-camera-dji-drone-zenmuse-h20n/>, 2022.
- [48] SpeedyBee. SpeedyBee F745 35A BLS 25.5x25.5 AIO Flight Controller. <https://www.speedybee.com/speedybee-f745-35a-bls-25-5x25-5-aio-flight-controller/#Manual>, 2022.
- [49] D. Stelmach. Protokół UDP. <https://pasja-informatyki.pl/sieci-komputerowe/protokol-udp/>.
- [50] R. Stephen. Quadcopter Axes and Motions. <https://www.flickr.com/photos/ricstephens/16989545847/>, 2018.
- [51] A. D. Team. Connect ESCs and Motors. <https://ardupilot.org/copter/docs/connect-escs-and-motors.html>, 2021.
- [52] J. Wigard, R. Amorim, I. Z. Kovács. Controlling drones over cellular networks. <https://www.bell-labs.com/institute/white-papers/controlling-drones-over-cellular-networks/#gref>, 2021.
- [53] C. Zieliński. Quo vadis robotyko? *Pomiary Automatyka Robotyka*, 5:5–15, 2010.
- [54] A. Zourmand, A. L. Kun Hing, C. Wai Hung, M. AbdulRehman. Internet of Things (IoT) using LoRa technology. *IEEE International Conference on Automatic Control and Intelligent Systems*, strony 324–330, Selangor, Malaysia, 2019.