

UNIwersytet Rzeszowski
Kolegium Nauk Przyrodniczych



Krystian Jamrogiewicz

Nr albumu: 114403

Mechatronika

**Projekt i wykonanie bezzałogowego statku latającego w oparciu o
mikrokontroler ESP32**

Praca inżynierska

Praca wykonana pod kierunkiem

dr Mariusza Bestera

Rzeszów, 2023

SPIIS TREŚCI

WSTĘP	3
CEL I ZAŁOŻENIA PROJEKTU	4
1. CZĘŚĆ TEORETYCZNA	5
1.1. Środowisko sprzętowe	7
1.1.1. Śmigła i silniki	8
1.1.2. Regulatory ESC	16
1.1.3. Kontroler lotu	17
1.1.4. Żyroskop i akcelerometr	20
1.1.5. Zasilanie.....	23
1.2. Środowisko programowe	26
1.2.1. Arduino IDE	27
1.2.2. Aplikacja sterująca	30
1.2.3. Autodesk Eagle.....	32
1.2.4. Autodesk Inventor.....	33
2. UKŁAD STEROWANIA	35
2.1. Połączenie elementów elektronicznych	35
2.2. Wykonanie oprogramowania sterującego.....	40
3. UKŁAD MECHANICZNY	49
3.1. Projektowanie ramy	49
3.2. Wykonanie ramy	56
3.3. Rozmieszczenie elementów na ramie.....	61
3.4. Przeprowadzenie testów	62
PODSUMOWANIE.....	64
BIBLIOGRAFIA	65

WSTĘP

Bezzałogowe statki latające już w latach trzydziestych XX wieku były określane przez armię USA jako drony. Drony są sterowane zdalnie lub za pomocą zaprogramowanej trajektorii. Aktualnie wykorzystywane są zarówno podczas obserwacji jak i wykonywania różnych czynności w miejscach gdzie człowiek byłby narażony na niebezpieczeństwo [7].

Pierwszy bezzałogowy statek latający powstał w 1935 roku i był sterowany za pomocą nadajnika radiowego. Dron szybko zainteresowało armię USA, początkowo konstrukcje tego typu pełniły funkcję celów szkoleniowych dla obrony przeciwlotniczej. Z czasem zaczęły pełnić funkcje samolotów, a ich zadaniem było zbieranie danych i fotografowanie obszarów, do których bezpośredni lot załogowym samolotem lub helikopterem byłby niemożliwy lub zbyt niebezpieczny [5, 7].

W Polsce formalnie drony określa się jako bezzałogowe statki latające (BSL), w przestrzeni międzynarodowej wykorzystuje się częściej skrót UAV (z angielskiego: Unmanned Aerial Vehicle) lub UAS (z angielskiego: Unmanned Aircraft System).

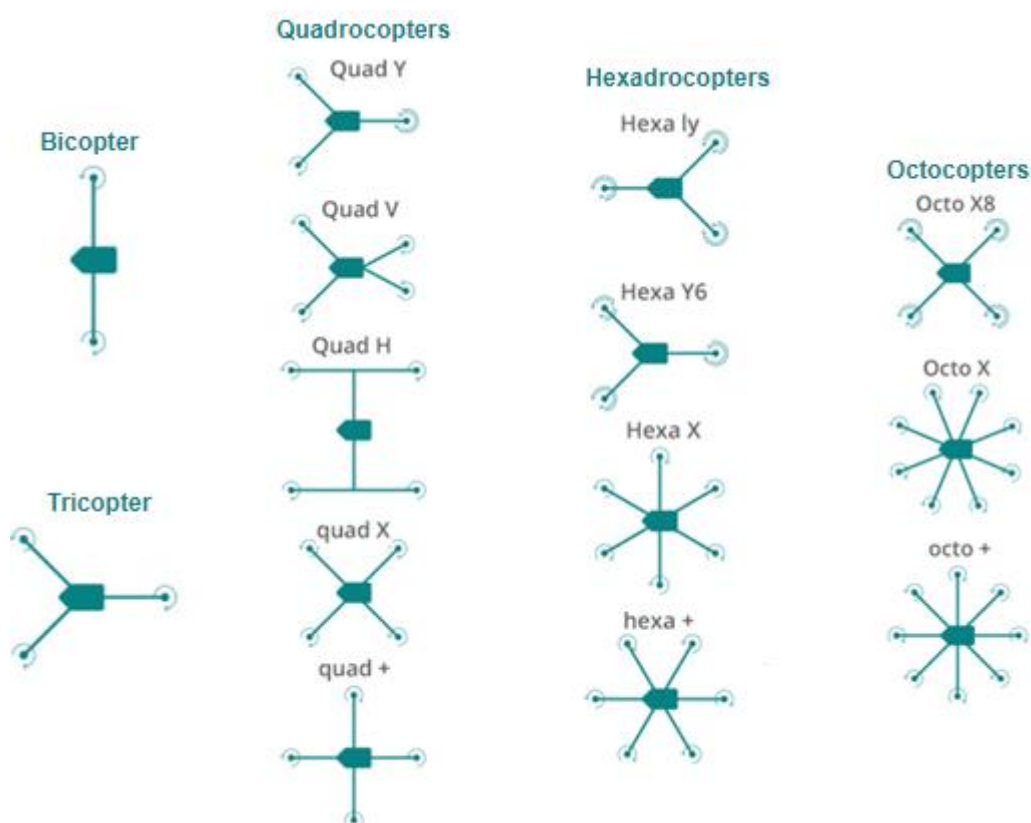
Podobnie jak większość zaawansowanych technologii, tak drony początkowo były bardzo drogie w produkcji. Natomiast od kilku lat, dzięki dynamicznemu rozwojowi elektroniki, zaczęły powstawać drony w formie wielowirnikowców. Wielowirnikowce w celu uzyskania siły nośnej korzystają z wielu wirników (śmigieł). Dzięki możliwości zawieszenia w powietrzu są stosowane głównie w celach fotograficznych, a także niekiedy transportują lżejsze obiekty w trudno dostępne miejsca. Bezzałogowe statki latające w formie wielowirnikowców w literaturze technicznej określa się również jako koptery lub multikoptery, w tej pracy inżynierskiej będą wykorzystywane właśnie takie terminy [5, 10, 15].

CEL I ZAŁOŻENIA PROJEKTU

Celem pracy inżynierskiej jest zaprojektowanie i wykonanie bezzałogowego statku latającego w formie wielowirnikowca bazując na układzie ESP32. Praca będzie opierać się na przedstawieniu teorii działania wielowirnikowca, dobraniu podzespołów potrzebnych do jego wykonania, zaprojektowaniu i wykonaniu układu elektronicznego, a także oprogramowania umożliwiającego zdalne sterowanie układem. Całość będzie dopełniona o zaprojektowanie oraz wykonanie ramy multikoptera, rozmieszczeniu na niej wyżej wymienionych elementów oraz przeprowadzeniu testów całości. Zakłada się wykonanie w pewnym stopniu działającego prototypu bezzałogowego statku latającego, poznanie jego możliwości, oraz przeanalizowanie uzyskanych wyników.

1. CZĘŚĆ TEORETYCZNA

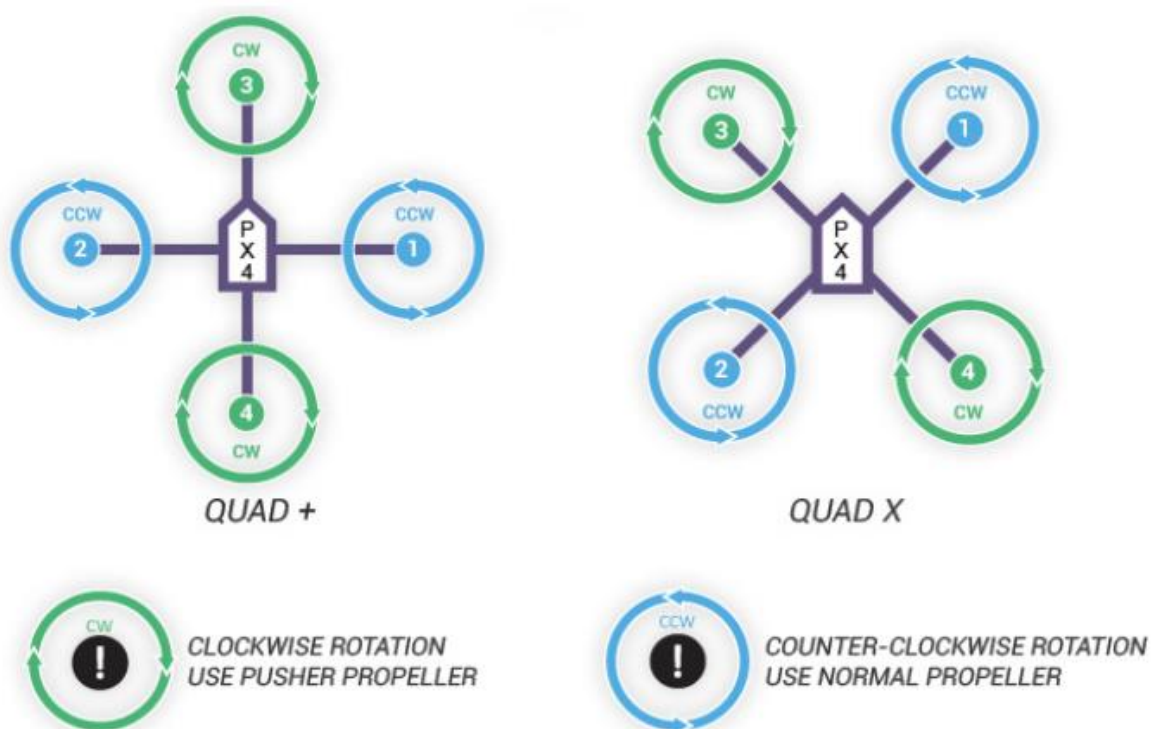
Profesjonalnie produkowane wielowirnikowce są sterowane za pomocą fal radiowych. Standardem zaczyna być technologia FPV (z angielskiego: First Person View) umożliwia ona podgląd na żywo obrazu z kamery zamontowanej na multikopterze przy użyciu np. smartfonu. Transmisja obrazu odbywa się za pośrednictwem radiowego przesyłania danych. Klasyfikacja konstrukcji wielowirnikowców opiera się głównie o liczbę ramion na których zamontowane są zespoły napędowe (silniki wraz ze śmigłami). Najpopularniejsze z nich to: trikoptyery – trzy zespoły napędowe kwadrokoptyery – cztery zespoły napędowe, heksakoptery – sześć zespołów napędowych [15, 16].



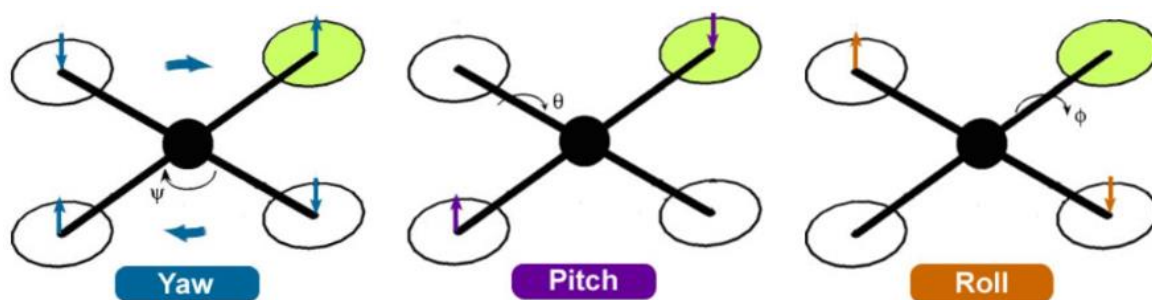
Rys. 1.1. Rodzaje konstrukcji wielowirnikowców [16]

Konstrukcja z większą liczbą ramion pozwala na bardziej stabilny lot, natomiast podwójne zespoły napędowe montowane na mniejszej liczbie ramion (np. Y, X8) zwiększają siłę ciągu, umożliwiając większy udźwig i ubezpieczając równoległy zespół napędowy na wypadek awarii. Najbardziej rozpowszechnione są koptery w układzie X ponieważ umożliwiają montaż kamery w miejscu o najlepszej widoczności [16]. Praca inżynierska zostanie zrealizowana w oparciu o konstrukcje kwadrokoptera typu X

Aby w ogóle wielowirnikowiec był w stanie zawisnąć podczas lotu i nie obracać się w niekontrolowany sposób wokół własnej osi stosuje się pary zespołów napędowych, które obracają się w przeciwnych kierunkach, będzie to omówione szerzej w podrozdziale dotyczącym śmigieł. W celu uzyskania kontrolowanego obrotu „yaw” należy zmniejszyć ciąg powietrza dla jednej pary a zwiększyć dla drugiej, pozwala to na uzyskanie efektu obrotu bez strat na wysokości. Podczas swobodnego zwisu wielowirnikowca konieczne jest takie sterowanie ciągiem zespołów napędowych aby kompensowały one jego przechylenia. Zbyt duże przechylenie doprowadzi do obrócenia się obiektu śmigłami w dół i jego upadku. Ruch do przodu, do tyłu „pitch” oraz na boki „roll” jest realizowany poprzez przechylenie kwadrokoptera w oparciu o zmniejszenie lub zwiększenie ciągu zespołów napędowych. Przykładowo ruch do przodu odbędzie się poprzez zmniejszenie ciągu w przedniej części multikoptera a zwiększenie w tylnej, uzyska się wtedy efekt pochylenia do przodu co zapewni siłę pchającą skierowaną w kierunku pochyłonej niżej strony. Cały proces sterowania poszczególnymi zespołami napędowymi jest realizowany przez kontroler lotu.



Rys. 1.2. Kierunki obrotu śmigieł dla kwadrokopterów [17]



Rys. 1.3. Rodzaje ruchu dla kwadrokoptera typu + [17]

1.1. Środowisko sprzętowe

Ten podrozdział został poświęcony opisowi urządzeń oraz części wykorzystanych podczas wykonania projektu inżynierskiego. Poruszone zostaną zasady wyboru

poszczególnych podzespołów z uwzględnieniem ich parametrów technicznych. Znajdą się w nim informacje na temat: zespołów napędowych, regulatorów prędkości, kontrolera lotu, urządzenia odczytującego położenie, oraz źródła zasilania.

1.1.1. Śmigła i silniki

Śmigła i silniki są głównymi systemami napędowymi multikoptera i odgrywają decydującą rolę. Elementy te poddawane są największym obciążeniom spośród wszystkich komponentów multikoptera. Z wszystkich podzespołów składających się na całość modelu latającego parametry śmigieł oraz silników powinny zostać dobrane w pierwszej kolejności. Aby wielowirnikowiec mógł poruszać się w powietrzu stosunek ciągu wszystkich zespołów napędowych do masy całego obiektu powinien wynosić przynajmniej 2:1. Rozmiar śmigieł decyduje o rozmiarze wielowirnikowca i to właśnie na nich spoczywa ciężar całej konstrukcji [15].

Podstawową rolę śmigła jest wytwarzanie ciągu powietrza, czyli siły nośnej działającej w osi śmigła umożliwiającej unoszenie się modelu. Śmigło o większej średnicy będzie generowało większą siłę nośną niż jego mniejszy odpowiednik przy tej samej prędkości obrotowej. Cechą większych śmigieł jest większa masa i potrzeba zapewnienia większego momentu obrotowego do ich napędzenia co skutkuje doбором mocniejszego silnika. Większe śmigło generuje również większy nacisk na piastę za sprawą większej siły nośnej, należy pamiętać aby wybierać śmigła możliwie dobrej jakości w celu uniknięcia pęknięcia śmigła podczas jego wysokich obrotów. Zbyt małe śmigło powoduje częstą pracę silnika na maksymalnych obrotach, natomiast zbyt duża powierzchnia łopat nadmiernie obciąża silnik co prowadzi do braku możliwości uzyskania maksymalnego możliwego ciągu. Ilość łopat na śmigle wpływa na ciąg w podobny sposób co jego średnica. Im więcej będzie łopat (tym większa ich powierzchnia) tym większą siłę ciągu uda się uzyskać, jednak przełoży się to na masę śmigła i zwiększone zużycie energii przez silnik. Zaletą takiego rozwiązania jest jednak kompaktowość konstrukcji ponieważ śmigło o mniejszej średnicy z większą liczbą łopat może pochwalić się podobnym ciągiem co jego większy odpowiednik z mniejszą ich ilością. Kolejnym ważnym parametrem jest skok śmigła, można go porównać do skoku gwintu na śrubie. Ściśle zależy on od kąta pochylenia łopat śmigła jednak nie jest tym samym. Skok

śmigła to odległość jaką pokonało by śmigło w powietrzu w czasie jednego obrotu gdyby nie dochodziło do jego poślizgu. Analogicznie dla śruby będzie to odległość jaką pokona wierzchołek śruby podczas wkręcania jej w gwint przez jeden pełny obrót. Parametry śmigła podaje się za pomocą jego cechy czyli napisu widocznego na śmigle. Przykładowo oznaczenie 1045 lub 10x4,5 oznacza, że śmigło ma średnicę 10 cali i skok 4,5 cala [6, 12, 15].



Rys. 1.4. Śmigła 1045 CW oraz CCW [18]

Podsumowując, im większe śmigło tym większy uzyskany ciąg, im większy skok śmigła przy tej samej średnicy tym większa będzie prędkość modelu. Dla modelu akrobacyjnego stosuje się śmigło o większym skoku ponieważ będzie on latał z większymi prędkościami, modele przeznaczone do celów fotograficznych latające bardzo wolno, którym zależy na dużej stabilności powinny mieć największą możliwą średnicę śmigła przy możliwie małym skoku. Istnieje podział śmigieł na prawe oraz lewe, jest to spowodowane naturalnym zjawiskiem obrotu całego koptera w kierunku ruchu śmigła. Aby pozbyć się tego efektu

stosowane są pary silników obracające się w przeciwnie strony co skutkuje koniecznością posiadanie dwóch typów śmigieł, które w obydwu przypadkach muszą zapewnić ciąg powietrza w odpowiednim kierunku. Wyróżniamy śmigła ciągnące (prawe) obracające się zgodnie ze wskazówkami zegara oznaczone CW z angielskiego: Clockwise, lub R od regular. Śmigła pchające (lewe) kręcą się w lewą stronę posiadają oznaczenia CCW z angielskiego: Counter Clockwise - przeciwnie do wskazówek zegara, oraz P – pusher [19].

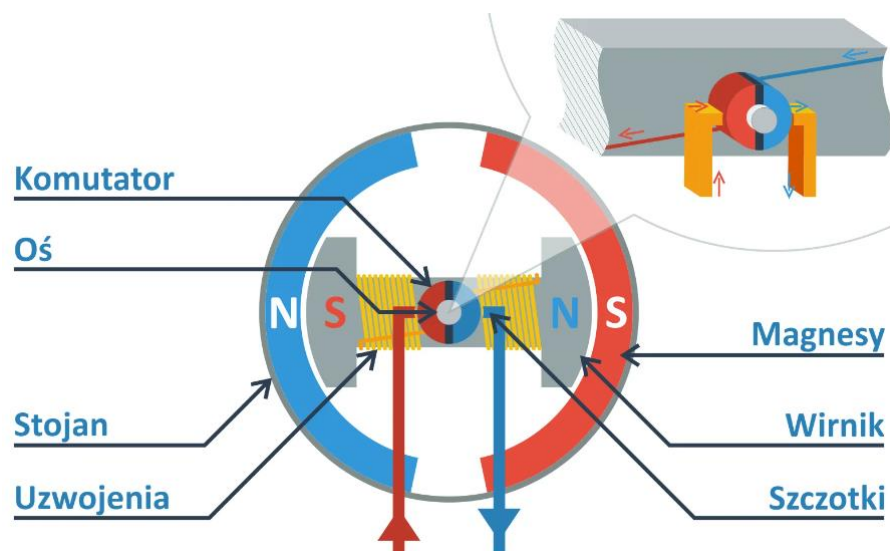
Tab. 1.1. Zależność ciągu od parametrów śmigła [19]

4Inch A Series Blade								
Prop	RPM	Thrust (g)	Prop	RPM	Thrust (g)	Prop	RPM	Thrust (g)
4inch A-3 Blade	9300	122.00	4inch A-4 Blade	9180	152.00	4inch A-6Blade	8340	164.00
	9960	140.00		9660	166.00		8880	186.00
	10440	156.00		9900	170.00		9240	196.00
	11760	206.00		11230	230.00		10500	252.00
	12900	242.00		12300	272.00		11340	302.00
	15900	382.00		13560	320.00		12360	372.00
5Inch A Series Blade								
Prop	RPM	Thrust (g)	Prop	RPM	Thrust (g)	Prop	RPM	Thrust (g)
5inch A-3 Blade	8820	126	5inch A-4 Blade	8700	142	5inch A-6Blade	8100	168
	9360	146		9240	160		8520	182
	9660	156		9480	170		8820	198
	11160	216		10860	224		10020	256
	12060	252		11700	274		10860	290
	13320	312		12840	328		11340	350
6Inch A Series Blade								
Prop	RPM	Thrust (g)	Prop	RPM	Thrust (g)	Prop	RPM	Thrust (g)
6inch A-3 Blade	8460	196	6inch A-4 Blade	8100	218	6inch A-6Blade	7680	236
	8940	226		8580	236		8100	262
	9300	238		8880	258		8340	280
	10440	312		10020	338		9420	372
	11340	374		10800	392		10020	428
	12300	448		11760	476		10680	496
6Inch M Series Blade								
Prop	RPM	Thrust (g)	Prop	RPM	Thrust (g)	Prop	RPM	Thrust (g)
6inch M-3 Blade	8700	138	6inch M-4 Blade	8160	152	6inch M-6Blade	7740	172
	9180	154		8640	172		8100	194
	9540	170		8940	188		8580	216
	10740	224		10200	248		9600	276
	11760	272		10980	288		10260	318
	12840	324		12060	356		10860	372

Do materiałów z których wykonuje się śmigła zaliczamy: poliwęglan, nylon oraz włókno szklane, oraz wszelkie ich mieszanki. W zależności od materiału użytego w produkcji śmigła mogą być miękkie (poliwęglan, nylon) lub twarde (włókno szklane). Zaletą śmigieł miękkich jest duża wytrzymałość na uderzenia, jednak trzeba liczyć się z ich odkształcaniem. Natomiast śmigła twardsze mają dużo lepsze parametry dotyczące stabilności lotu i brak jakichkolwiek ugięć, z kolei ich wadą jest możliwość łatwiejszego uszkodzenia podczas upadku [6, 15, 19].

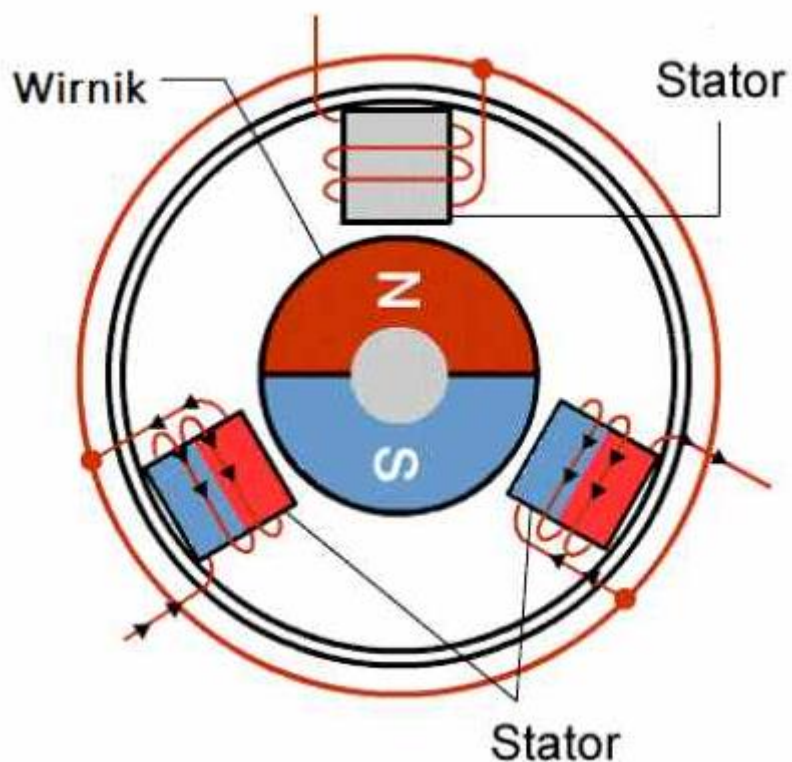
Silnik elektryczny zamienia energię elektryczną w energię mechaniczną, za pomocą pola magnetycznego. W opisywanym projekcie ma za zadanie poruszyć śmigło z odpowiednią prędkością w celu wytworzenia przez nie optymalnego ciągu powietrza. Silniki wykorzystywane w kopterach dzielimy na szczotkowe oraz bezszczotkowe.

Odmiana szczotkowa często nazywana DC charakteryzuje się zasilaniem przez prąd stały, zbudowana jest z: wirnika (rotoru), przez który przechodzi wyprowadzona na zewnątrz oś, uzwojenia nawiniętego na wirnik, stojanu (statoru) zawierającego magnesy stałe, który otacza wirnik, komutatora, który przełącza uzwojenia na wirniku, oraz szczotek doprowadzających zasilanie do komutatora. Szczotki są elementem podlegającym zużyciu na skutek tarcia, ponadto silniki szczotkowe mają mniejszą sprawność niż bezszczotkowe. Sterowanie obrotami odbywa się poprzez zmianę napięcia zasilającego silnik [9, 15].



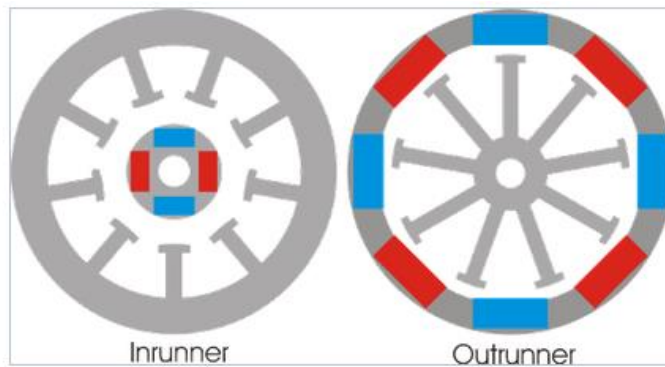
Rys. 1.5. Budowa silnika szczotkowego [9]

Odmiana bez szczotkowa BLDC jest zbudowana z wirnika (rotoru), na którym zamocowane są magnesy oraz stojanu inaczej statoru, na którym znajdują się uzwojenia.



Rys. 1.6. Budowa silnika bez szczotkowego [4]

Na podstawie rozmieszczenia tych dwóch komponentów, silniki bez szczotkowe dzielą się na inrunner „gdzie rotor jest otoczony przez stator” i outrunner „gdzie stator jest otoczony przez rotor”. W wielowirnikowcach częściej stosowany jest typ outrunner, który pozwala na uzyskanie większego momentu obrotowego [15].



Rys. 1.7. Budowa silników Inrunner i outrunner [27]

Dzięki prostej budowie i nie posiadaniu elementów podatnych na zużycie silniki bezszczotkowe górują nad wariantem szczotkowym za sprawą znacznie mniejszej awaryjności, większej sprawności, oraz możliwością płynniejszej regulacji obrotów. W konstrukcjach wielowirnikowych wykorzystywane są silniki zasilane prądem trójfazowym, ich działanie jest niemożliwe bez komutatora którego rolę pełni osobna elektronika zwana regulatorem ESC. Regulacja prędkości odbywa się przez zmianę częstotliwości przełączania faz [15].

Najważniejszymi parametrami silnika są: napięcie zasilające podawane w voltach lub za pomocą parametru S (zostanie omówione w podrozdziale poświęconemu zasilaniu), maksymalny pobór prądu, maksymalna liczba obrotów na minutę, oraz liczba obrotów na minutę uzyskana z jednego wolta napięcia zasilania oznaczana jako KV. Im mniejsza jest

wartość parametru KV tym mniejsze są maksymalne obroty silnika, jednak rośnie wartość momentu obrotowego. Dla większych śmigieł stosuje się silniki z mniejszym współczynnikiem KV [15].

Aby poprawnie dobrać zespół napędowy (śmigło i silnik) należy skorzystać z informacji dotyczących parametrów silnika najczęściej znajdujących się na stronie sklepu. Kluczową informacją jest zakres dopuszczalnych rozmiarów śmigieł przewidziany dla tego silnika.

Tab. 1.2. Parametry doboru śmigieł dla silnika A2212/13T 1000KV [20]

Motor

Grayson-Welgard 2212/13: 1000Kv: 50g: 12A (150W) max 4.11.07, PH 25A ESC

Temperatures (52°C) from Rear mount plate

No-load:

7.0v, 0.60A, 7380 rpm

8.0v, 0.65A, 8460 rpm

10.0v, 0.75A, 10500 rpm

Peak Effic. Calc: Kv: ~1061Kv, Rm: ~ 0.276 ohms

Prop	Volts	Amps	Watts	RPM	Pitch Speed (mph)	Thrust (g)	Thrust (oz)	g/W
8x4 GWS HD								
Motor 27°C: Set on PS: 7v	7.0	3.35	23	6630	25.1	226	7.96	9.83
8v	7.9	4.10	32	7410	28.1	287	10.11	8.97
9v	8.9	4.85	43	8220	31.1	347	12.22	8.07
10v	9.9	5.65	56	8940	33.9	420	14.79	7.50
77mAh, 32°C: 11v	10.9	6.50	71	9660	36.6	495	17.43	6.97
9x5 GWS HD								
Motor 25°C: Set on PS: 7v	6.9	5.50	38	6000	28.4	348	12.25	9.16
8v	7.9	6.70	53	6660	31.5	436	15.35	8.23
9v	8.9	7.85	70	7290	34.5	526	18.52	7.51
10v	9.9	9.25	92	7920	37.5	627	22.08	6.82
130mAh, 38°C: 11v	10.9	10.45	113	8430	39.9	715	25.18	6.33
10x5 APCE								
Motor 31°C: Set on PS: 7v	6.9	7.00	48	5610	26.6	406	14.30	8.46
8v	7.9	8.45	66	6120	29.0	505	17.78	7.65
9v	8.9	9.90	87	6690	31.7	604	21.27	6.94
10v	9.9	11.45	112	7170	33.9	702	24.72	6.27
214mAh, 53°C: 11v	10.9	13.00	141	7650	36.2	802	28.24	5.69
10x6 GWS HD								
Motor 22°C: Set on PS: 7v	6.9	7.20	50	5610	31.9	424	14.93	8.48
8v	7.9	8.70	69	6180	35.1	526	18.52	7.62
9v	8.9	10.10	89	6690	38.0	617	21.73	6.93
10v	9.9	11.70	115	7200	40.9	722	25.42	6.28
156mAh, 43°C: 11v	10.9	13.25	144	7680	43.6	817	28.77	5.67
10x8 GWS HD								
Motor 25°C: Set on PS: 11v	10.8	18.20	195	6390	48.4	733	25.81	3.76
+60mAh, to 44°C:								

Silnikami i dla projektowanego koptera będą: A2212/13T 1000KV. Z dokumentacji technicznej wynika, że dla tego typu silników optymalne będą śmigła o wymiarach w zakresie od 8x4 do 10x8. Silnik powinien być zasilany napięciem 11,1V. Kolumny „Amps” oraz „Thrust” w powyższej tabeli informują o maksymalnym poborze prądu, oraz wytworzonym ciągu dla danej prędkości obrotowej oraz rozmiaru śmigła. Z uwagi na łatwą dostępność dla tego projektu zostaną wybrane śmigła o oznaczeniu 1045.

Wybrany zespół napędowy będzie charakteryzował się: napięciem zasilania 11,1V maksymalnym poborem prądu równym 13A, oraz maksymalnemu ciągowi około 750g. Z uwagi na brak danych na temat śmigła o pożądanych wymiarach (1045) parametry zostały oszacowane na podstawie śmigła (1050) o zbliżonych wymiarach. Dobrze dobrany zespół napędowy powinien dysponować odpowiednim ciągiem i nie pobierać przy tym olbrzymich ilości energii.



Rys. 1.8. Śmigła i silniki zastosowane w projekcie

1.1.2. Regulatory ESC

Elektroniczna kontrola prędkości ESC - z angielskiego: Electronic Speed Control. Zadaniem regulatorów ESC jest przekazanie energii z baterii do silnika. Na podstawie sygnału sterującego PWM z kontrolera lotu regulator ESC przekazuje do silnika napięcie które zapewni mu odpowiednią prędkość obrotową. Głównym podziałem regulatorów jest ten ze względu na rodzaj obsługiwanego silnika: silniki szczotkowe (2 przewody zasilające silnik) i silniki bezszczotkowe (3 przewody zasilające silnik). Niektóre Regulatory ESC posiadają możliwość zasilania urządzeń mniejszym napięciem niż to z baterii, przykładowo gdy napięcie baterii jest większe niż 6V można zasiląć płytkę kontroli lotu napięciem 5V z odpowiedniego wyprowadzenia regulatora ESC. Regulator który to umożliwia posiada obwód eliminatora baterii zwany BEC - Battery Elimination Circuit. Kolejnym ważnym parametrem jest napięcie pracy często wyrażane w liczbie ogniw baterii np.: 2S, 3S, 4S. Ten aspekt zostanie szerzej omówiony w części poświęconej zasilaniu. Do wcześniej wybranych śmigieł i silników należy dobrać regulatory ESC o odpowiednim prądzie pracy. Zbyt słaby regulator może ulec uszkodzeniu, zbyt mocny regulator nie przynosi negatywnego efektu poza większą wagą i ceną. Regulator powinien być tak dobrany aby jego prąd pracy był o około od 20% do 30% większy od maksymalnego prądu pobieranego przez silnik z uwzględnieniem wielkości śmigła. Jeżeli silnik często jest blokowany lub w pełni obciążony (na przykład w samochodach RC) zalecane jest dobranie regulatora o prądzie pracy bliskim prądowi szczytowemu silnika [15, 21, 22].



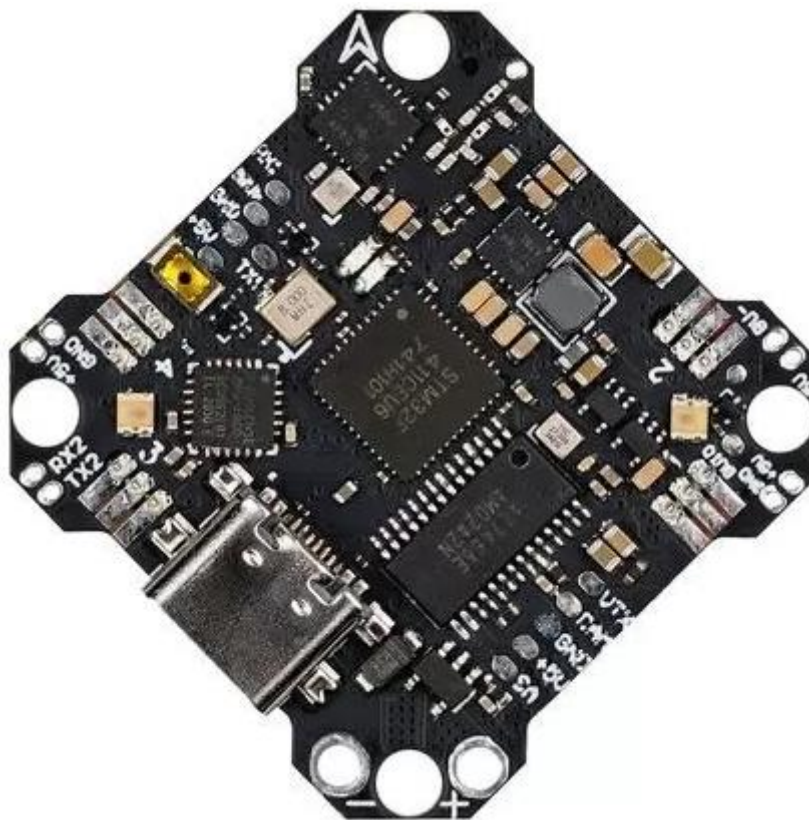
Rys. 1.9. Zastosowany w projekcie regulator ESC [23]

Podczas wykonywania tego projektu wybrano regulatory ESC: Brushless 30A 2-4S. Regulator tego typu może pracować z ciągłym prądem równym 30A i prądem szczytowym równym 40A. Posiada BEC - możliwość zasilania innych podzespołów napięciem 5V z maksymalnym prądem 2A. Może pracować z akumulatorami od 2S do 4S.

1.1.3. Kontroler lotu

Kontroler lotu jest poniekąd „mózgiem” modelu, który odpowiada za jego prawidłowe reakcje na wydawane mu polecenia sterujące, oraz czynniki zewnętrzne takie jak podmuchy wiatru. Dzięki danym wejściowym: pochyleniom i przyspieszeniom (względem osi: x, y, z), zaprogramowany algorytm kontrolera lotu wyznacza orientację multikoptera w przestrzeni. W następnej kolejności ustala odpowiednie prędkości poszczególnych silników. Profesjonalne kontrolery lotu składają się z mikroprocesora, żyroskopu i akcelerometru na jednej kompaktowej płytce. Bardziej zaawansowane konstrukcje posiadają również dodatkowe czujniki i systemy jak: kompas magnetometr, barometr. Mikroprocesor płytki kontroli lotu powinien być wystarczająco szybki aby

podjąć wielu skomplikowanym obliczeniom w możliwie najkrótszym czasie. Wolne procesory mogą być przyczyną nadmiernych korekt podczas lotu [10, 15].



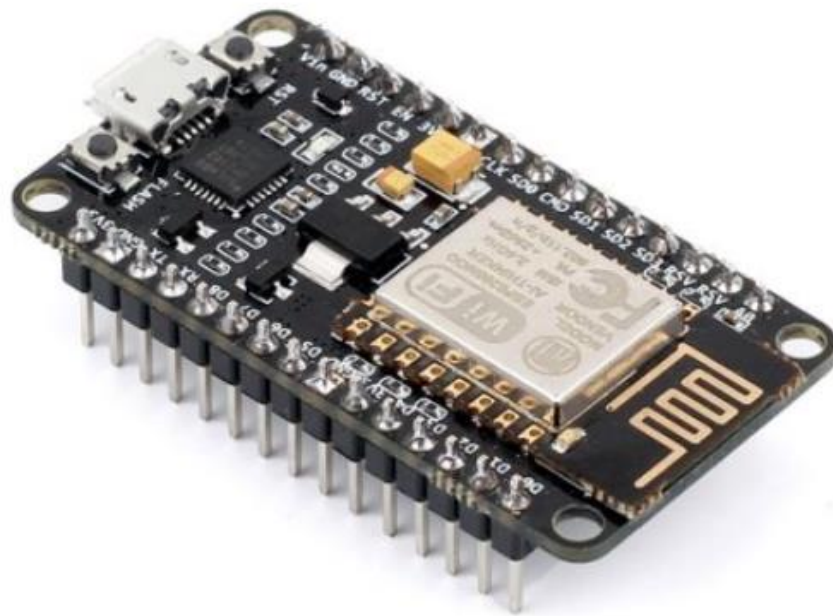
Rys. 1.10. Kontroler lotu F4 1S 12A AIO V2 (ELRS 2.4G) – BETAFPV [20]

Układ ESP8266 osadzony w płytce NodeMCU v2 jest przykładem mikrokontrolera z możliwością sterowania sygnałem WiFi. W omawianym projekcie będzie on spełniał funkcję kontrolera lotu.

ESP8266 jest układem SoC (z angielskiego: System on a Chip) od chińskiego producenta Espressif Systems. Układ zawiera w sobie elementy niezbędne do komunikacji przez WiFi w standardzie 802.11 b/g/n pracujących z częstotliwością 2,4 GHz. Posiada również wbudowany stos TCP/IP i może działać w trybach AP (Access Point), STA (standalone) oraz AP+STA [2, 8].

NodeMCU v2 jest płytką, która wykorzystuje układ ESP8266 w postaci 22 pinowego modułu z wbudowanym 32-bitowym układem RISC taktowanym zegarem 80MHz. Układ wyróżnia się 10 portami GPIO, 2 portami UART oraz jednym 10-bitowym wejściem ADC. Każdy z portów GPIO może być wykorzystany jako PWM, I²c, SPI lub 1-Wire.

Moduł NodeMCU v2 został wyposażony w stabilizator napięcia 3,3V oraz konwerter USB-to-UART CP2120, dzięki czemu jest możliwe zasilanie i programowanie płytki bezpośrednio z portu USB. Programowanie płytki może odbywać się za pomocą języka skryptowego Lua lub przy pomocy Arduino IDE [2, 8].

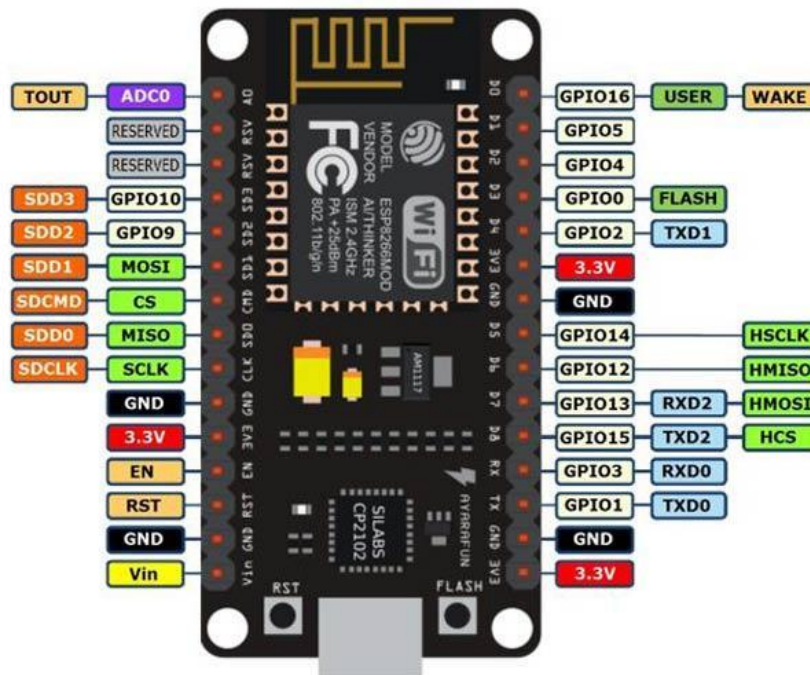


Rys. 1.11. Płytki NodeMCU v2 [8]

Dane techniczne:

- Moduł zbudowany w oparciu o układ ESP8266-12E z anteną PCB
- Łączność Wi-Fi w standardzie 802.11 b/g/n
- Działa w trybach AP (Access Point), STA (Standalone), AP+STA
- Obsługuje TKIP, WEP, CRC, CCMP, WPA/WPA2, WPS
- Zasilanie: 4,5V - 10V (lub 5V przez port USB)
- Maksymalny pobór prądu: 215mA
- Procesor RISC 80MHz
- 10 portów GPIO - PWM / I2C / SPI / 1-Wire
- Maksymalne natężenie na portach I/O: 12mA
- Konwerter USB-UART - CP2102
- Konwerter ADC - 10-bitowy
- 30 portów w rastrze 2,54mm
- Złącze micro USB
- Wymiary: 49 x 25mm

- Dwa przyciski: FLASH, służący do programowania, oraz RST (RESET)
- Niebieska dioda LED podłączona do GPIO16



Rys. 1.12. Opis wyprowadzeń płytki NodeMCU v2 [28]

1.1.4. Żyroskop i akcelerometr

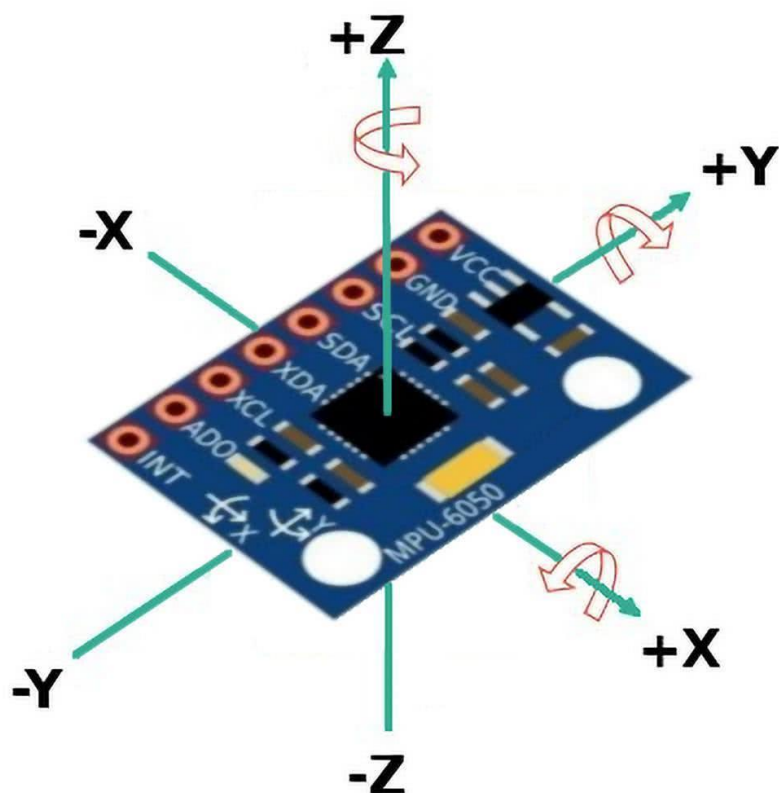
Orientacja multikoptera określa jego pochylenie, przechylenie i odchylenie. Czujniki potrafiące odczytać orientację to akcelerometry i żyroskopy. Są one umieszczane w taki sposób, aby mogły mierzyć ruch we wszystkich trzech płaszczyznach. Akcelerometry odczytują przesunięcia wzdłuż trzech poszczególnych osi (x, y, z, nazywane też translacjami), podczas gdy żyroskopy mierzą obroty wokół osi pochylenia, przechylenia i odchylenia. System sterowania powinien być jak najszybszy i najczulszy, aby w porę mógł zareagować np. na gwałtowny podmuch wiatru. Optymalny efekt można uzyskać stosując zarówno żyroskop jak i akcelerometr. [10, 15].

Istnieją dwa typy akcelerometrów: piezoelektryczne i pojemnościowe. Akcelerometry piezoelektryczne są spotykane najczęściej. Zawierają one mikroskopijne kryształy, które reagują na nacisk, gdy struktura kryształu zostanie odkształcona, generowany jest prąd elektryczny. Mierząc polaryzację i wielkość tego prądu, system

sterowania może ustalić kierunek i prędkość ruchu. W akcelerometrach pojemnościowych mierzona jest pojemność elektryczna między mikrostrukturami znajdującymi się blisko czujnika. W miarę jak mikrostruktury te zbliżają się lub oddalają od czujnika, zmienia się pojemność. Mierząc ją, system sterowania może określić kierunek i prędkość ruchu [15].

Istnieje również kilka rodzajów żyroskopów: kryształowe (których działanie jest zbliżone do akcelerometrów piezoelektrycznych), ceramiczne (w których wykorzystywane są miniaturowe ceramiczne kolumny lub pryzmaty) i silikonowe (których budowa różni się w zależności od producenta) [15].

W projekcie został wykorzystany moduł żyroskopu z akcelerometrem MPU6050. Zamontowany na płytce modułu układ scalony MPU6050 zawiera struktury MEMS (z angielskiego: Micro Electro Mechanical Systems) pełniące funkcję półprzewodnikowych żyroskopów i akcelerometrów. Czujniki układu odgrywające rolę akcelerometrów mierzą wartość przyspieszenia w 3 osiach, a czujniki żyroskopów mierzą rotację względem tych osi. Gdy moduł jest ustawiony poziomo i pozostaje w bezruchu, akcelerometr osi Z powinien zmierzyć wartość około $9,81 \text{ m/s}^2$ co odpowiada średniemu przyspieszeniu ziemskiemu, natomiast akcelerometry osi X i Y powinny podawać wartości bliskie 0 m/s^2 . Odwrócenie płytki „do góry nogami” skutkuje odczytem wartości ujemnej – $9,81 \text{ m/s}^2$ zmierzonej w osi Z. Każde przesunięcie płytki np. w kierunku +X spowoduje pomiar w tej osi chwilowego przyspieszenia o wartości dodatniej, proporcjonalnego do prędkości ruchu płytki. Z kolei w czasie obracania płytki wokół osi żyroskopy mierzą obrót w jednostkach stopni kątowych na sekundę [13, 25].



Rys. 1.13. Opis osi dla modułu MPU6050 [13]

Płytką jest wyposażona w układ stabilizatora LDO pozwalający rozszerzyć dopuszczalne napięcie zasilania do zakresu 3,3V - 5V. Charakteryzuje się niskim poborem prądu do około 350μA. Dodatkowo na płytce zamontowano oporniki podciągające linie magistrali I²C do napięcia zasilającego oraz opornik ustawiający na wyprowadzeniu wyboru adresu ADO poziom niski. Uzyskiwane w czasie pomiarów sygnały analogowe akcelerometrów i żyroskopów są poddawane obróbce w wewnętrznych przetwornikach ADC i jako wartości cyfrowe są dostępne do odczytu z rejestrów układu MPU6050. [13].

Oznaczenia i funkcje wyprowadzeń:

- VCC – napięcie zasilające 3,3...5 V
- GND – masa zasilania
- SCL – linia zegarowa magistrali I²C podciągnięta do VCC
- SDA – linia danych magistrali I²C podciągnięta do VCC
- XDA – linia danych dodatkowej niezależnej magistrali I²C
- XCL – linia zegara dodatkowej niezależnej magistrali I²C
- ADO – wejście wyboru adresu, na który będzie reagował moduł w obrębie głównej magistrali I²C (dla poziomu niskiego będzie to adres 0x68, dla poziomu wysokiego (3,3 V) 0x69. Linia jest podciągana do poziomu niskiego poprzez opornik na płytce modułu)

- INT – wyjście przerwania, może być zaprogramowana do wystawiania stanu niskiego w przypadku wystąpienia zdarzenia np. przekroczenia poziomu przyspieszenia

1.1.5. Zasilanie

Istnieje wiele typów akumulatorów jednak multikoptery najczęściej są zasilane za pomocą wariantów litowo polimerowych (Li-Po). Ten typ akumulatora ma najlepszy stosunek masy do energii jaką magazynuje. Składa się z jednego lub kilku połączonych ogniw (cel). Ogniwo Li-Po jest zbudowane ze stopów litu i polimerów o właściwościach przewodzących, zwykle w postaci stałej lub żelu, jak standardowe ogniwa posiada anodę oraz katodę. Napięcie pracy pojedynczego ogniwa mieści się w zakresie od 2,75V do 4,2V. Do poprawnej eksploatacji akumulatora Li-Po niezbędne będzie używanie specjalnych ładowarek z balanserem, które dostarczają takie samo napięcie do każdego ogniwa, należy pamiętać, by ładowarka nie ładowała akumulatora prądem większym niż jego maksymalny prąd ładowania. Aby cieszyć się długim działaniem akumulatora nie należy dopuszczać do rozładowania się pojedynczych ogniw poniżej napięcia 3V. W celu długotrwałego przechowywania akumulatora należy naładować pojedyncze ogniwa do napięcia od 3,7V do 3,8V [15].



Rys. 1.14. Akumulator zastosowany w projekcie [29]

Napięcie znamionowe jest podstawowym parametrem akumulatora, jego wielkość musi mieścić się w zakresie napięć obsługiwanych przez silniki. Jest podawane w woltach, lub za pomocą parametru S. Liczba stojąca przed S określa z ilu ogniw (cel) składa się akumulator, przykładowo 3S oznacza, że akumulator jest zbudowany z 3 ogniw. Biorąc pod uwagę, że napięcie znamionowe jednego ogniw wynosi 3,7V to akumulator 3S będzie miał napięcie znamionowe równe: $3 \times 3,7V = 11,1V$ [15].

Pojemność akumulatora wyrażona w mAh (miliamperogodzinach) określa jak duży prąd przez określony czas można pobrać z w pełni naładowanego akumulatora aby go w pełni rozładować. Im większa pojemność tym akumulator będzie mógł dłużej zasilać odbiorniki, jednak wraz ze wzrostem pojemności zwiększają się rozmiary i masa akumulatora [15].

Wydajność prądowa, określana parametrem C jest to maksymalne natężenie prądu jakie można pobrać z akumulatora. Oblicza się ją poprzez mnożenie liczby stojącej przed C oraz pojemności akumulatora. Przykładowo akumulator o pojemności 2200mAh i oznaczeniu 30C będzie cechował się następującą wydajnością prądową:

Zamiana na jednostki SI: $2200mAh = 2,2Ah$

Wydajność prądowa: $2,2Ah \times 30 = 66A$

Stosuje się także oznaczenia np.: 30C-40C, pierwsza cyfra (30) to współczynnik dla ciągłej wydajności prądowej, a druga (40) dla chwilowej, co oznacza, że przez krótką chwilę akumulator może osiągnąć większą wydajność prądową bez jego uszkodzenia [15].

Dobrze dobrany akumulator musi być w stanie zaspokoić zapotrzebowanie prądowe dla wszystkich zespołów napędowych oraz innych podzespołów multikoptera. Zbyt mała wydajność prądowa doprowadzi do uszkodzenia akumulatora oraz nie uzyskania maksymalnego możliwego ciągu przez zespół napędowy.

Parametry odbiorników:

Zespół napędowy:

Napięcie zasilania silnika: 11,1V

Maksymalny pobór prądu dla jednego silnika: 13A

Maksymalny pobór prądu dla wszystkich silników: $13A \times 4 = 52A$

Kontroler lotu:

Napięcie zasilania: 4,5V – 10V

Maksymalny pobór prądu = 215mA = 0,215A

Żyroskop / akcelerometr:

Napięcie zasilania: 3,3V – 5V

Maksymalny pobór = 350μA = 0,00035A

Maksymalny pobór prądu całego układu = 52A + 0,215A + 0,00035A = 52,21535A

Napięcie zasilania całego układu będzie wynosić 11,1V ponieważ właśnie takiego wymagają silniki, Kontroler lotu oraz żyroskop / akcelerometr będą zasilane napięciem 5V przekonwertowanym z 11,1V przez regulatory ESC.

Maksymalny pobór prądu całego układu będzie wynosił około 52,21535A. Dla bezpieczeństwa wydajność prądowa akumulatora powinna być przynajmniej 1A większa niż obliczony pobór całego układu. Znając maksymalny pobór prądu można oszacować czas lotu. Teoretycznie dla akumulatora o pojemności 2200mAh obciążenie pobierające prąd 2200mA rozładowuje go w pełni po upływie godziny. Jednak w praktyce akumulatory Li-Po aby nie doznać uszkodzenia nie mogą być rozładowywane poniżej 20% ich pojemności.

Szacowanie czasu pracy dla obciążenia 53A i pojemności 2200mAh:

Pojemność akumulatora = 2200mAh = 2,2Ah

Zmniejszenie pojemności o 20%: $2,2 \times 0,8 = 1.76\text{Ah}$

Czas pracy akumulatora wyraża się wzorem:

$$t = C_p / I_r$$

gdzie:

t – czas pracy w godzinach

C_p – pojemność akumulatora

I_r – prąd pobierany przez odbiorniki

$$C_p = 1,76\text{Ah}$$

$$I_r = 53\text{A}$$

$$t = 1,76\text{Ah} / 53\text{A} = 0.0332075471698113\text{h} \approx 120\text{s} = 2\text{min}$$

Czas pracy podczas poboru 53A wynosi 2 min. Należy pamiętać, że przyjęty pobór prądu jest oszacowany dla silników kręcących się z maksymalną prędkością, a czas przeciętnego lotu powinien potrwać nieco dłużej. Aby dokładniej oszacować czas lotu należy obliczyć uśrednioną wartość poboru prądu.

Wybór odpowiedniego akumulatora dla projektu:

Wymagany będzie akumulator o napięciu znamionowym 11,1V i wydajności prądowej większej niż 53A. Dla danego układu zastosowane będzie źródło zasilania typu Li-Po firmy Redox o oznaczeniach: 3S 2200mAh 30C. Posiada ono napięcie znamionowe 11,1V, pojemność 2200mAh, oraz wydajność prądową równą 66A. Jego masa to 150 g a wymiary: 103 x 34 x 20 mm. Oszacowany czas lotu na jednym ładowaniu będzie nie mniejszy niż 2 minuty. Dla projektu o tych parametrach bardziej zalecany byłby akumulator o większej pojemności zapewniający dłuższy czas lotu, jednak z powodu dostępności oraz testowego przeznaczenia projektu zostało wybrane po wyższe źródło zasilania.

1.2. Środowisko programowe

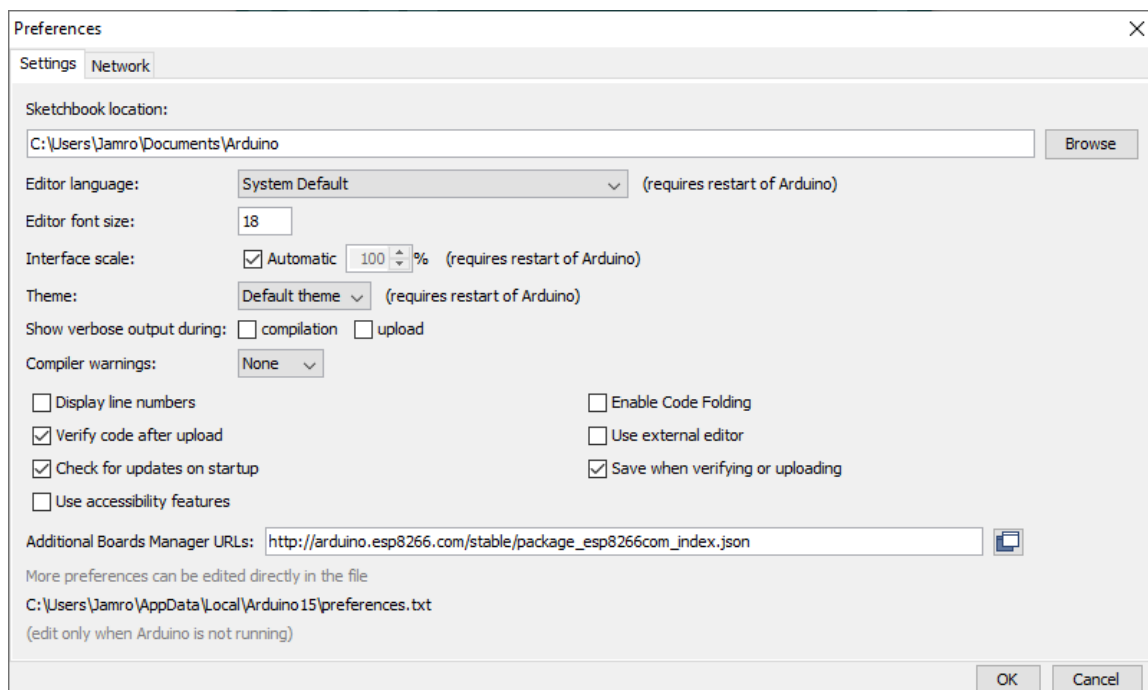
W tym podrozdziale zostały omówione oprogramowania oraz aplikacje wykorzystane podczas wykonania pracy inżynierskiej. Znajdą się w nim informacje na temat oprogramowania służącego do programowania mikrokontrolerów, aplikacji która posłużyła do sterowania urządzeniem, a także programów używanych do projektowania obwodu elektrycznego, oraz modelowania 3D.

1.2.1. Arduino IDE

Arduino IDE (z angielskiego: Integrated Development Environment) jest otwartym środowiskiem programistycznym, które upraszcza proces pisania kodu i pozwala przesać go na płytki z rodziny Arduino. Otwarty kod źródłowy - open source umożliwia ciągłe udoskonalanie środowiska przez osoby które z niego korzystają. Oprogramowanie jest także open hardware – otwarte sprzętowo co oznacza, że można je stosować na dowolnej płytce która została wykonana zgodnie ze standardem Arduino [11]. Oprogramowanie można pobrać ze strony: <https://www.arduino.cc/en/software> [dostęp: 31.01.2023 r.].

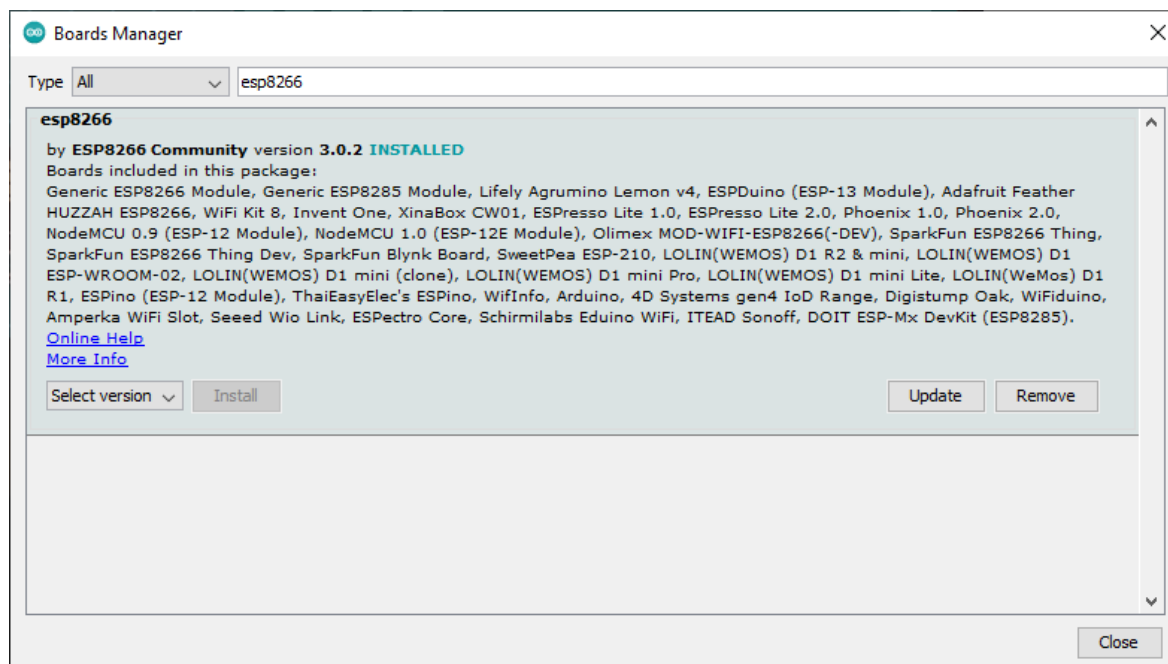
Oprogramowanie to posłużyło do zaprogramowania modułu NodeMCU v2 aby spełniał on funkcję kontrolera lotu. W tym celu należało odpowiednio skonfigurować środowisko Arduino IDE. Podczas wykonywania omawianego projektu korzystano z Arduino IDE 1.8.18.

Pierwszym zabiegiem jest otworzenie zakładki: „File -> Preferences” a następnie w polu: „Additional Boards Manager URLs” wpisać link: http://arduino.esp8266.com/staging/package_esp8266com_index.json [8].



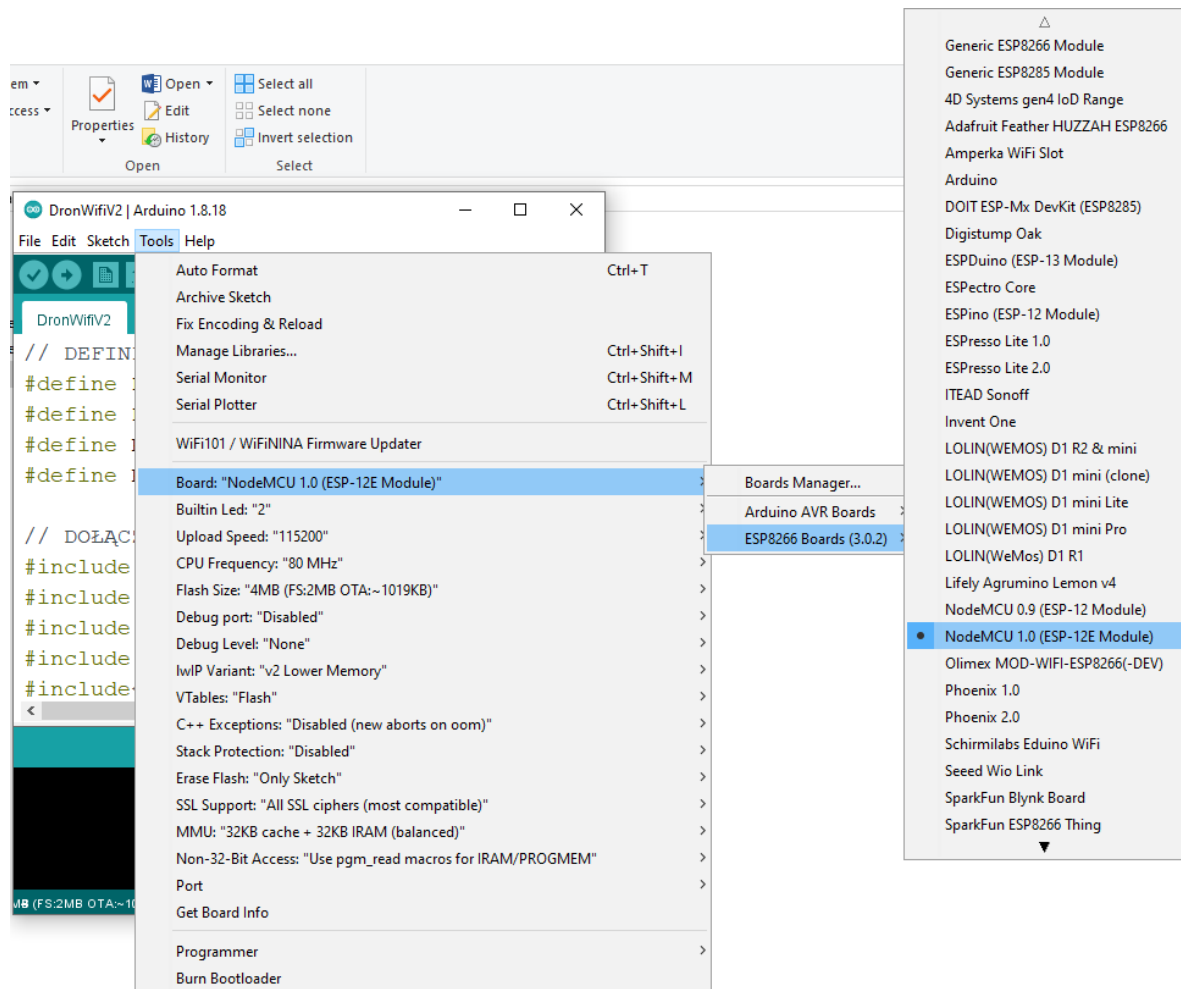
Rys. 1.15. Ustawienia w zakładce „Preferences”

W następnej kolejności należy otworzyć zakładkę „Tools -> Board -> Boards Manager” oraz zainstalować pakiet dla płytek esp8266.



Rys. 1.16. Wybór pakietu dla płytek ESP8266

Ostatnim krokiem jest wybór płytki NodeMCU 1.0 (ESP12-E Module) w zakładce: "Tools -> Board -> ESP8266 Boards (3.0.2)". Po wyborze odpowiedniej płytki należy ustawić resztę parametrów tak jak na poniższym zdjęciu oraz wybrać odpowiedni port usb.



Rys. 1.17. Wybór odpowiedniej płytki wraz z jej ustawieniami.

Do prawidłowego działania systemu konieczne jest zainstalowanie sterownika CH340. Po wykonaniu tych wszystkich czynności płytkę NodeMCU v2 można będzie programować w środowisku Arduino IDE za pomocą składni języka obsługiwanej przez klasyczną płytkę Arduino [13].

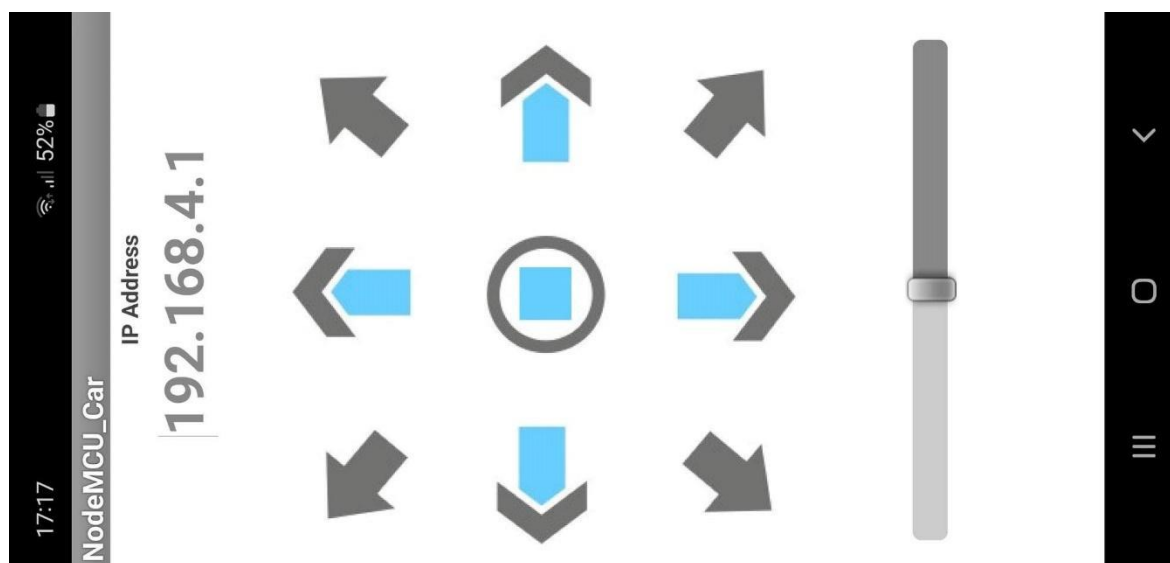
1.2.2. Aplikacja sterująca

Do sterowania zaprojektowaną konstrukcją posłużyła aplikacja o nazwie: „NodeMCU_Car” udostępniona przez twórców kanału na YouTube o nazwie „Pro Know”. Aplikacja jest dostępna na urządzenia z androidem, można ją pobrać ze strony:

https://drive.google.com/drive/folders/1DwreeBjXSEnIHlfhAz_XfE8erKbKwk

[dostęp: 31.01.2023 r.].

Do komunikacji z kontrolerem lotu wykorzystuje sygnał WiFi którego źródłem jest układ ESP8266. Aby połączyć telefon z multikopterem należy wyszukać odpowiednią sieć WiFi i się z nią połączyć. Interfejs aplikacji posiada 9 przycisków monostabilnych oraz suwak o 10 położeniach podtrzymujący sygnał ostatnio ustawionej pozycji.



Rys. 1.18. Interfejs aplikacji

Działanie aplikacji polega na wysyłaniu do modułu ESP8266 komend które są przypisane do odpowiedniego przycisku i pozycji suwaka.

Tab. 1.3. Komendy wysyłane przez interfejs [opracowanie własne]

Symbol	Komenda
Strzałka w górę	F
Strzałka w dół	B
Strzałka w prawo	R
Strzałka w lewo	L
Strzałka w górę i prawo	I
Strzałka w górę i lewo	G
Strzałka w dół i prawo	J
Strzałka w dół i lewo	H
Przycisk na środku	S
Suwak w pozycji 0	0
Suwak w pozycji 1	1
Suwak w pozycji 2	2
Suwak w pozycji 3	3
Suwak w pozycji 4	4
Suwak w pozycji 5	5
Suwak w pozycji 6	6
Suwak w pozycji 7	7
Suwak w pozycji 8	8
Suwak w pozycji 9	9

Podczas projektowania multikoptera zdecydowano, że bardziej ergonomicznym rozwiązaniem będzie używanie interfejsu sterującego w pozycji poziomej tak jak na załączonym powyżej zdjęciu. Odwrócenie przycisków zostało uwzględnione w poniższej tabeli.

Tab. 1.4. Elementy interfejsu zastosowane w projekcie [opracowanie własne]

Funkcja	Symbol	Komenda
Lot do przodu	Strzałka w prawo	R
Lot do tyłu	Strzałka w lewo	L
Lot w prawo	Strzałka w dół	B
Lot w lewo	Strzałka w górę	F
Obrót w prawo	Strzałka w dół i prawo	J
Obrót w lewo	Strzałka w góra i prawo	I

Suwak będzie odpowiadał za sterowanie prędkością obrotową silników. Położenie 0 będzie oznaczało, że silniki są wyłączone, natomiast 9 oznacza maksymalną ich prędkość. Położenia pomiędzy będą w możliwie płynny sposób odpowiadały za prędkości pomiędzy zerem a maksimum.

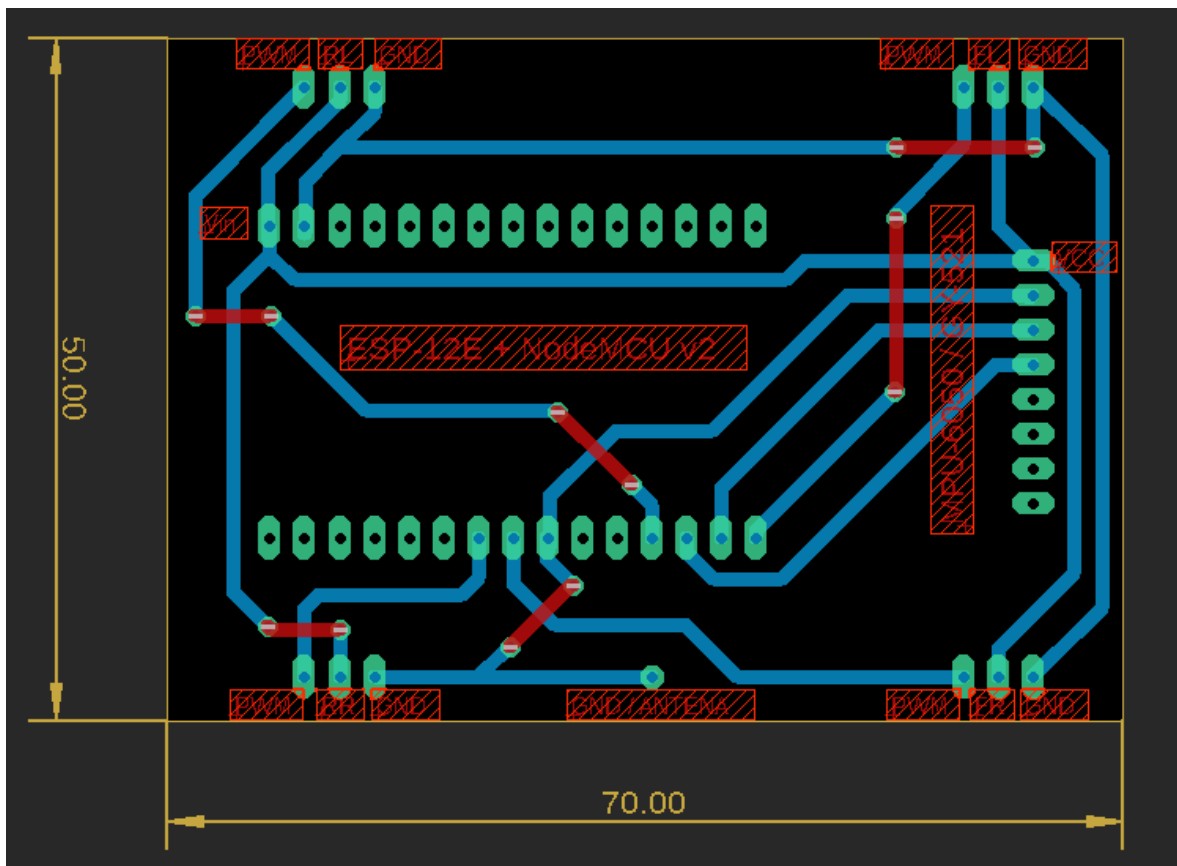
1.2.3. Autodesk Eagle

Autodesk Eagle jest to oprogramowanie komputerowe przeznaczone do wspomagania projektowania obwodów elektronicznych. Zostało stworzone przez firmę CadSoft a obecnie jest rozwijane przez Autodesk. Posiada edytor schematów oraz płytek drukowanych, a także bogatą bibliotekę elementów elektrycznych. Umożliwia modyfikację wielu parametrów, ważniejsze z nich to: szerokość ścieżki oraz kąt jej zagięcia. Przydatna okazuje się siatka wymiarowa pomagająca w dokładnym rozmieszczeniu elementów na płytce, a także poznaniu jej całkowitych wymiarów. Funkcja tworzenia tekstu daje możliwość opisu danych elementów, co pomaga w orientacji w trakcie podglądu, istnieje także możliwość naniesienia tekstu na wykonywaną płytkę. Podstawową wersję tego oprogramowania można pobrać za darmo ze strony Autodesk:

<https://www.autodesk.com/products/eagle/free-download> [dostęp: 31.01.2023 r.].

Podczas wykonywania pracy inżynierskiej oprogramowanie Autodesk Eagle 9.6.2 zostało zastosowane do zaprojektowania płytki PCB dla kontrolera lotu oraz urządzenia odczytującego orientację bezzałogowego statku powietrznego. Płytkę umożliwia proste

połączenie elektryczne podzespołów, ich zasilanie, przekaz sygnału sterującego, a także co ważne daje możliwość ich stabilnego umocowania.



Rys. 1.19. Zaprojektowana płytką PCB

1.2.4. Autodesk Inventor

Autodesk Inventor to oprogramowaniem typu CAD rozpowszechnione przez firmę Autodesk. Jest parametrycznym systemem projektowania w przestrzeni trójwymiarowej (3D) z podsystemem automatycznego generowania dokumentacji dwuwymiarowej (2D), umożliwiającą stosowanie również techniki projektowania adaptacyjnego. Dzięki oprogramowaniu można projektować części 3D, łączyć je w zespoły, przeprowadzać ich analizy wytrzymałościowe, tworzyć dokumentację techniczną a także animacje.

Typowy algorytm parametrycznego adaptacyjnego projektowania 3D wygląda następująco:

- Modelowanie niezależne części lub modelowanie ich od razu z poziomu zespołu.
- Modelowanie zespołu.
- Redagowanie parametrycznie sprzężonej z modelami dokumentacji zespołu pod różnymi postaciami (klasyczne rzuty na płaszczyźnie, widoki i animacje montażu i demontażu itp.).
- Parametryczna adaptacyjna modyfikacja projektu z poziomu modeli lub rysunków.

Autodesk Inventor tworzy przede wszystkim modele hierarchiczne, których elementem jest sekwencyjne drzewo historii operacji znajdujące w przeglądarce obiektów [1].

Podczas wykonywania bezzałogowego statku powietrznego oprogramowanie Autodesk Inventor Professional 2023 posłużyło do zaprojektowania ramy multikoptera. Oprogramowanie to jest płatne jednak firma Autodesk dysponuje darmową licencją dostępną dla osób posiadających status studenta. Można je pobrać ze strony producenta:

<https://www.autodesk.pl/products/inventor/overview?term=1-YEAR&tab=subscription>

[dostęp: 31.01.2023 r.].

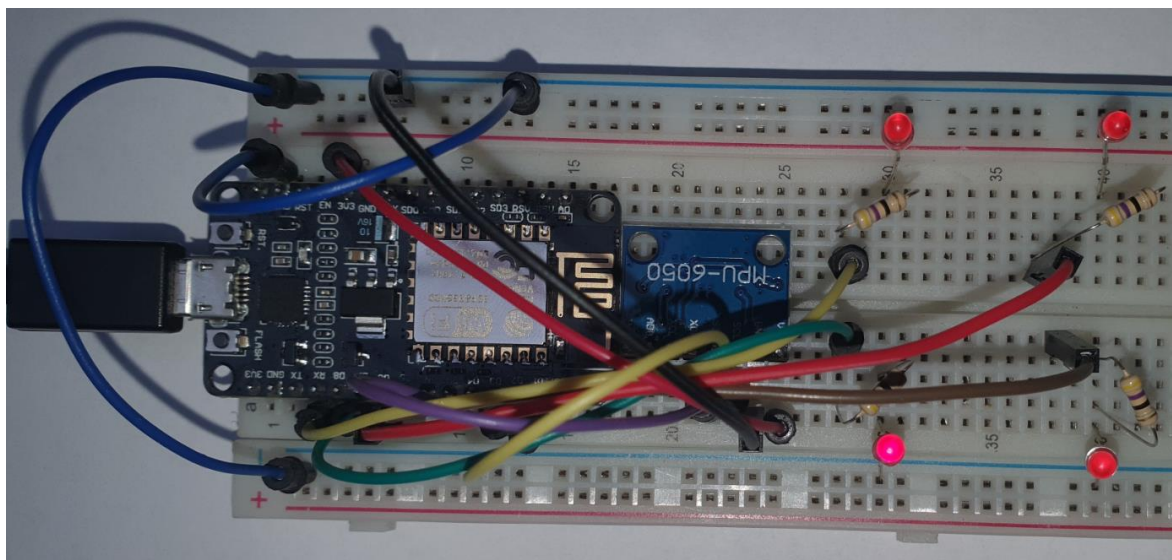
2. UKŁAD STEROWANIA

Układ sterowania odpowiada za odpowiednią reakcję multikoptera na przekazywany mu sygnał z interfejsu sterującego. Składa się zarówno z elementów elektronicznych jak i sterującego nimi algorytmu. W tym rozdziale został omówiony aspekt zaprojektowania oraz wykonania układu elektrycznego a także jego odpowiedniego zaprogramowanie. Aby układ sterowania działał zgodnie z założeniem zostały przeprowadzone symulacje do których posłużyła płytka stykowa.

2.1. Połączenie elementów elektronicznych

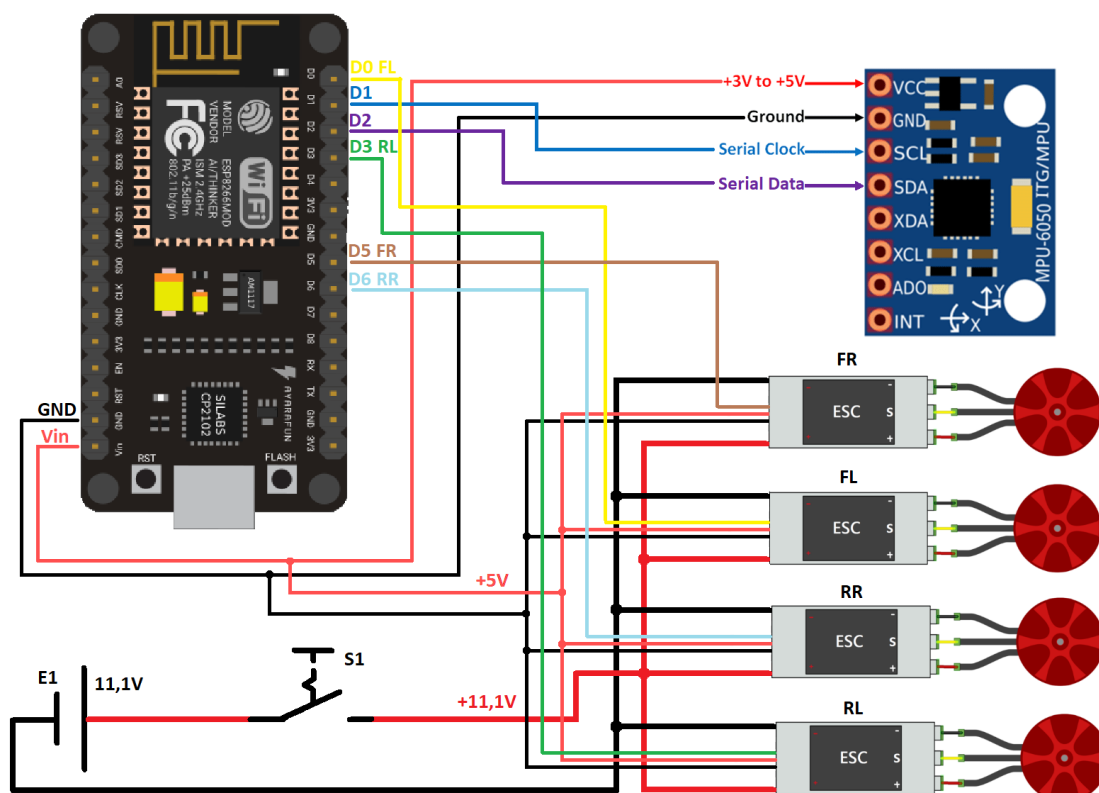
Ten podrozdział został poświęcony sposobowi połączenia wcześniej wybranych elementów elektronicznych według zaprojektowanych schematów. Omówione w nim są poszczególne kroki wykonywane podczas dążenia do ostatecznej wersji układu oraz napotkane problemy.

W początkowej fazie projektu układ elektryczny został zbudowany na płycie stykowej. Do jego wykonania posłużyły: płytka stykowa, NodeMCU v2, MPU6050, diody symulujące działanie zespołów napędowych, oraz rezystory konieczne do działania diod. Cały układ był zasilany z płytki NodeMCU v2 napięciem 3,3V która z kolei była zasilana napięciem 5V z portu USB. Do portów D1 oraz D2 został podłączony moduł MPU6050, do portów D5, D6, D7, D8 zostały podłączone diody. Całość bardzo dobrze sprawdziła się podczas tworzenia algorytmu sterującego. Diody były zasilane sygnałem PWM, który zależał od sygnału z interfejsu sterującego, oraz pochylenia całego układu odczytywanego przez moduł MPU6050. Poziom światła jakie wytwarzały diody symulował prędkość obrotową poszczególnych zespołów napędowych.



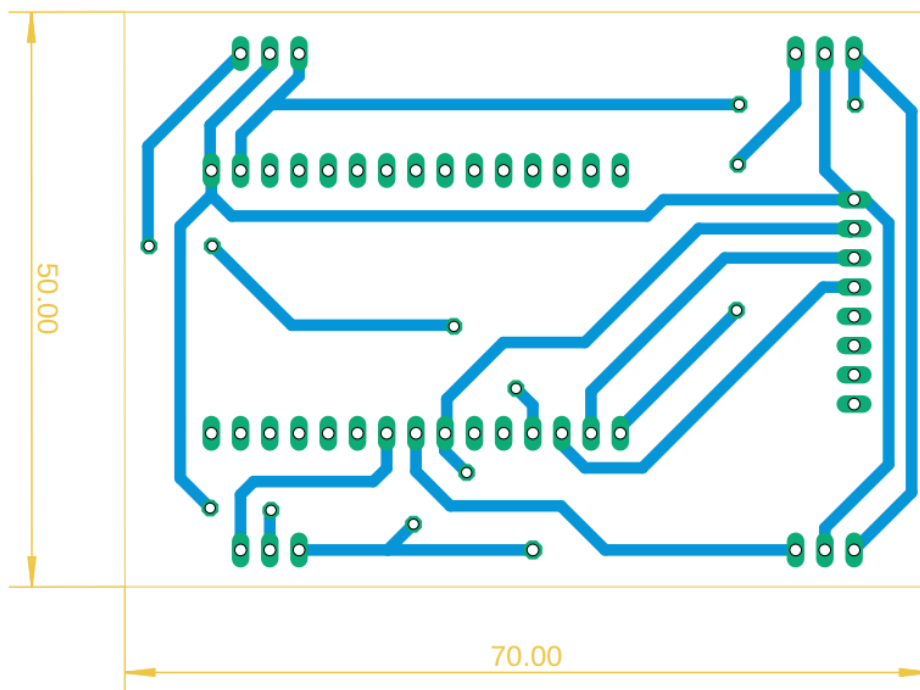
Rys. 2.1. Układ elektryczny wykonany na płytce stykowej

Po uzyskaniu optymalnego efektu na diodach zostały one zastąpione docelowymi regulatorami ESC do których podłączono silniki bez szczotkowe oraz źródło zasilania 11,1V. Wykonując ten zabieg napotkano na problem braku reakcji silników na sygnał sterujący, miał na to wpływ fragment kodu oraz rozdzielczość sygnału PWM co zostało omówione w następnym podrozdziale. Po poprawieniu kodu okazało się, że silniki których sygnał sterujący jest podawany z portów D7 oraz D8 nadal nie działają. Powodem tego zjawiska okazał się być tryb działania tych portów, problem rozwiązano poprzez wyprowadzenie sygnału sterującego silnikami z portów D0 oraz D3. Po wykonaniu tych operacji układ zaczął działać poprawnie. W ten sposób została zaprojektowana ostateczna wersja schematu elektrycznego.

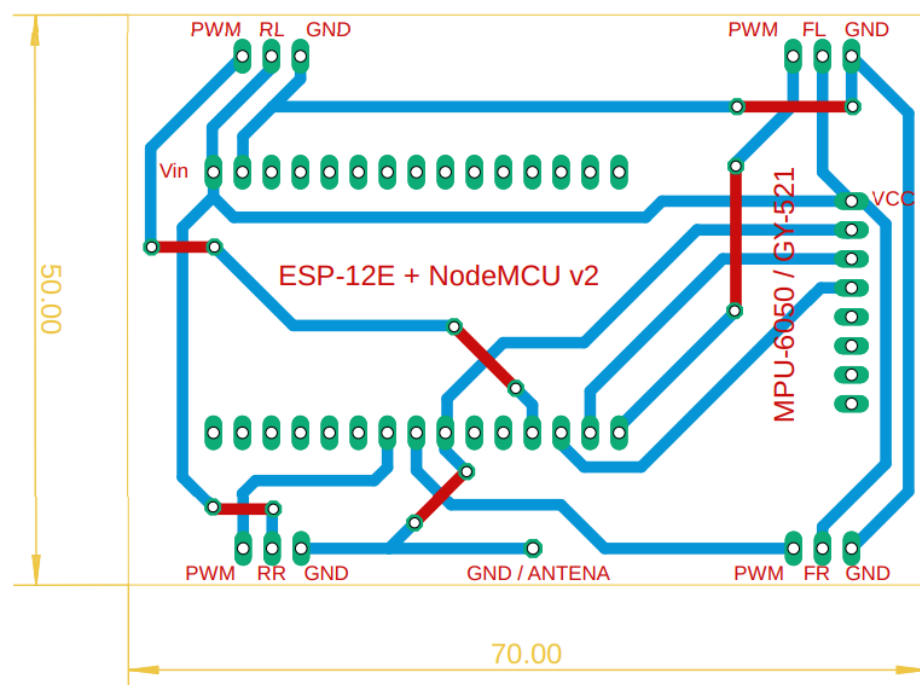


Rys. 2.2. Schemat układu elektrycznego

Aby w ergonomiczny sposób wykorzystać przestrzeń zdecydowano się na zaprojektowanie oraz wykonanie płytki PCB zgodnie z powyższym schematem. Projekt został wykonany w programie Eagle. Zaprojektowana płytka posiada gniazda montażu NodeMCU v2 oraz MPU6050, a także wyprowadzenia umożliwiające przekazywanie sygnału sterującego oraz jej zasilanie. Dodano także wyprowadzenie pozwalające na montaż anteny zewnętrznej.

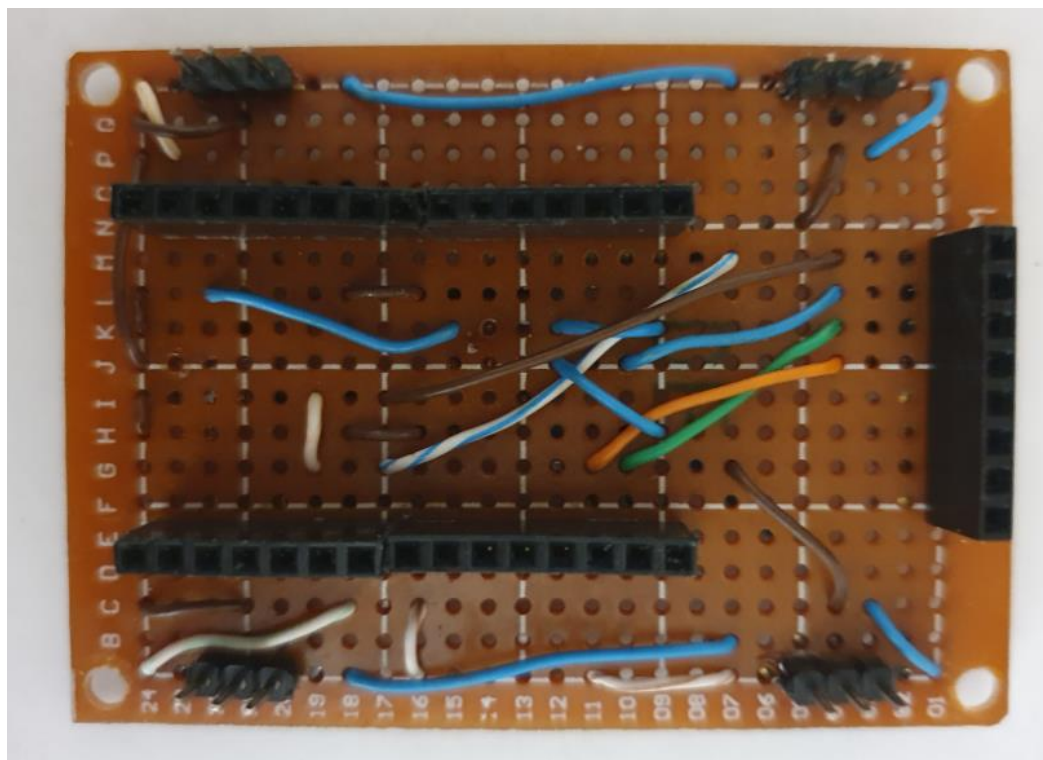


Rys. 2.3. Zaprojektowana płytki PCB, widok dolnej warstwy



Rys. 2.4. Zaprojektowana płytki PCB, widok na górną i dolną warstwę

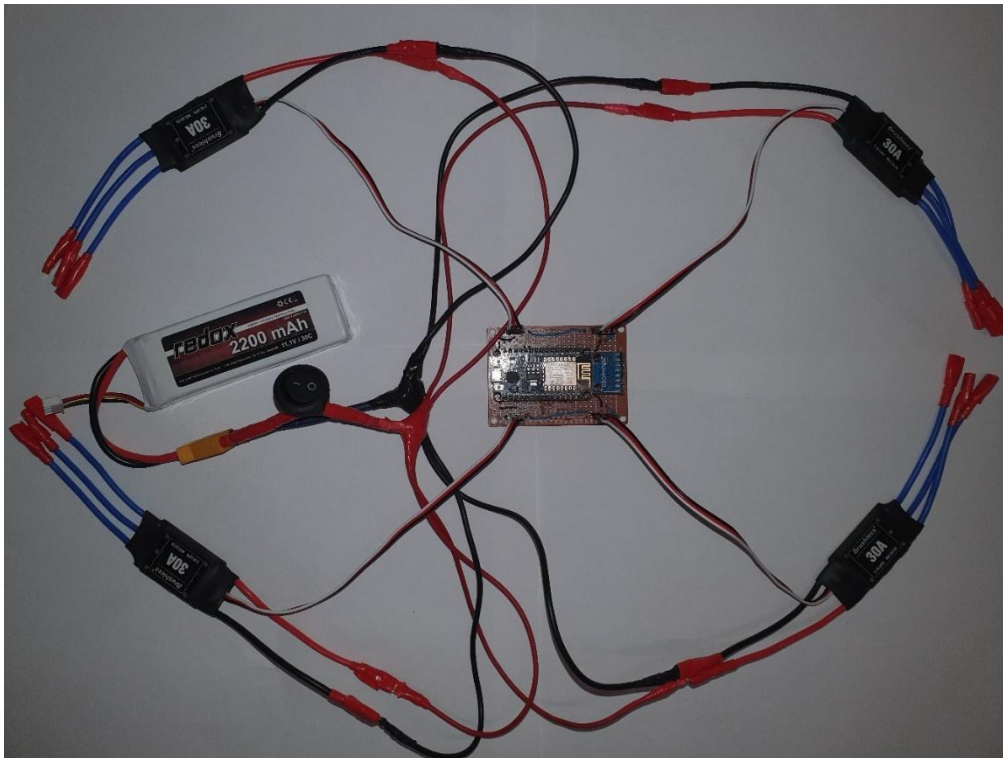
Z powodu łatwej dostępności i czasu realizacji płytka została wykonana na bazie płytki uniwersalnej.



Rys. 2.5. Wykonana płytka PCB

Wykonana płytka PCB umożliwiła stabilne osadzenie układów NodeMCU v2 oraz MPU6050, rozmieszczone na niej otwory umożliwiły zamontowanie jej do ramy za pomocą śrub. Jest zorientowana modułem MPU6050 do przodu multikoptera. Ze względu na zmianę orientacji modułu MPU6050 względem płytki stykowej należało zamienić jego osie w kodzie kontrolera lotu.

Ostatnim krokiem było wykonanie obwodu elektrycznego źródła zasilania. W tym celu wykorzystano przewody o wydajności prądowej do 54,5A dla połączenia od akumulatora do rozgałęzienia na poszczególne regulatory ESC oraz o wydajności do 18A od rozgałęzienia do regulatorów ESC. W celu połączenia układu ze źródłem zasilania zastosowano wtyczkę XT60 oraz przycisk bistabilny, połączenie do przycisku oraz regulatorów ESC zostało doprowadzone za pomocą konektorów.



Rys. 2.6. Połączony układ sterowania

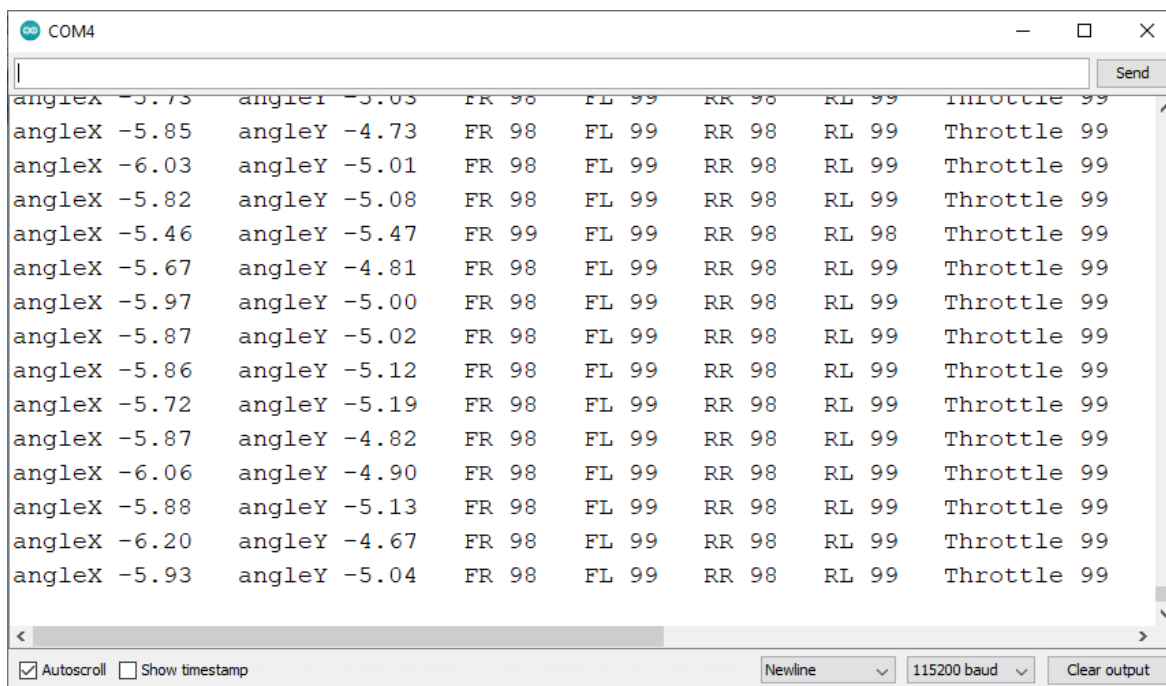
2.2. Wykonanie oprogramowania sterującego

Wykonanie oprogramowania sterującego odbyło się za pomocą środowiska Arduino IDE 1.8.18 skonfigurowanego w sposób umożliwiający mu komunikację z płytką NodeMCU v2. Oprogramowanie sterujące ma za zadanie odczytywać komendy przesyłane przez interfejs sterujący oraz informacje z czujników położenia, a następnie tak zmieniać sygnały sterujące prędkością obrotową poszczególnych zespołów napędowych aby otrzymać pożądaną efekt podczas lotu.

Podczas początkowej fazy tworzenia oprogramowania dużym ułatwieniem było wykonanie kompaktowego układu elektrycznego na płytce stykowej. Do nawiązania komunikacji między aplikacją na telefonie (interfejsem sterującym) a płytką zastosowano biblioteki i funkcje udostępnione przez autora aplikacji. Aby sterowanie było możliwe zostały napisane funkcje: `pitchFront()`, `pitchBack()`, `rollLeft()`, `rollRight()`, `yawRight()`, `yawLeft()` które odpowiadają za przechylenia statku powietrznego względem osi x (prawo,

lewo) i y (przód, tył) a także obrót wokół osi z (śmigieł). Zmienna throttle została stworzona w celu przechowywania stopnia otwarcia przepustnicy sterowanego za pomocą suwaka w interfejsie. W celu połączenia funkcji ruchu z komendami zastosowano instrukcje warunkowe. Fragment kodu odpowiedzialny za odczytywanie przechylenia został napisany w oparciu o artykuł poświęcony modułowi MPU6050 [25]. Wartości przechyleń zostały zsumowana algebraicznie w sposób zależny od lokalizacji diody (zespołu napędowego) z poziomem otwarcia przepustnicy. W ten sposób podczas przechylenia układu dioda znajdująca się poniżej środka ciężkości otrzymywała większą, a ta powyżej mniejszą moc, co ma symulować dostosowywanie prędkości obrotowych silników do przechylenia multikoptera. Aby móc regulować oddziaływanie wartości kąta przechylenia układu na wartość sygnału sterującego diodami wykonano regulatory PD dla osi x oraz y [14]. Na tym etapie możliwe było sterowanie jasnością diod za pomocą aplikacji oraz kąta przechylenia układu za pośrednictwem sygnałów PWM. Wykorzystano także funkcję `Serial.print()` w celu wyświetlenia w monitorze portu szeregowego informacji na temat najważniejszych zmiennych.

Podczas zmiany diod na docelowe regulatory ESC wraz z silnikami układ przestał reagować. Konieczne było zastosowanie biblioteki „Servo.h” oraz dodanie do kodu fragmentu odpowiedzialnego za ustawienie regulatorów ESC czyli: zdefiniowanie portu oraz minimalnej i maksymalnej szerokości impulsu sygnału PWM. Należało także zamienić rozdzielczość sygnału PWM z zakresu 0 - 255 na 0 - 180 do czego wykorzystano funkcję `map()` [3]. Wymagana okazała się być także zamiana portów z D7 i D8 na D0 i D3, ponieważ te pierwsze nie chciały współpracować z regulatorami ESC. W następnym etapie okazało się, że zmienna throttle najpierw musi zostać wykorzystana do obliczeń w swojej pierwotnej postaci (zakres 0 - 255) a dopiero w następnej kolejności może zostać zamieniona na zakres 0 – 180, ponieważ odwrotne działanie prowadziło do „zawieszenia się algorytmu”. W tym celu zmienianie rozdzielczości sygnału PWM z 0 - 255 na 0 - 180 zostało przeprowadzone osobno dla wszystkich sygnałów sterujących silnikami.



Rys. 2.7. Monitor portu szeregowego

Do zmiennych umożliwiających zmianę parametrów lotu należą: XKp, XKd, YKp, YKd są to nastawy regulatora PD odpowiadające za czułość stabilizacji lotu poprzez odczyty z czujników położenia. Kolejne zmienne to: Move1 – współczynnik odpowiadający za zwiększenie ciągu podczas funkcji ruchu, Move2 – współczynnik zmniejszający ciąg podczas funkcji ruchu, są odpowiedzialne za to jak bardzo statek powietrzny ma przechylić się w daną stronę podczas ruchu np. podczas lotu do przodu, decyduje to o jego prędkości i stabilności.

Ostateczna wersja kodu sterującego multikopterem:

```
// DEFINIOWANIE PINÓW STERUJĄCYCH ŚILNIKAMI
#define FR_PIN D5 // D5 Przedni prawy OBRÓT W LEWO
#define FL_PIN D6 // D6 Przedni lewy OBRÓT W PRAWO
#define RR_PIN D0 // D0 Tylny prawy OBRÓT W PRAWO
#define RL_PIN D3 // D3 Tylny lewy OBRÓT W LEWO

// DOŁĄCZENIE WYMAGANYCH BIBLIOTEK
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
```

```

#include <ESP8266WebServer.h>
#include<Servo.h>

// DEFINIOWANIE PINÓW STERUJĄCYCH SILNIKAMI DLA BIBLIOTEKI SERVO
Servo FR, FL, RR, RL;

// KOMUNIKACJA WiFi
String command;
const char* ssid = "DRON_ESP32"; // nazwa sieci
ESP8266WebServer server(80);

// ŻYROSKOP / AKCELEROMETR MPU6050
float angleX; // Kąt pochylenia względem osi X
float angleY; // Kąt pochylenia względem osi Y
const int MPU_addr = 0x68;
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
int minVal = 265;
int maxVal = 402;

// REGULATOR PD
float XKp = 0.04, XKd = 0.5; //Nastawy regulatora osi X
float YKp = 0.04, YKd = 0.5; //Nastawy regulatora osi Y

int setTime = 20; // (dt Czas po jakim pobiera się próbkę) Czas po którym zostanie ponownie przeliczona wartość wyjścia
float XPout, XDout; //Przechowują wyniki obliczeń członów regulatora osi X
float YPout, YDout; //Przechowują wyniki obliczeń członów regulatora osi Y
float Xout; //Wyjście regulatora osi X
float Yout; //Wyjście regulatora osi Y
float now, lastTime, timeChange; //Zmienne konieczne do ustalenia czasu ponownego przeliczenia wyjścia w regulatorze PD
float Xinput; //Sygnał z Żyroskopu / akcelerometru dla osi X (kąt)
float Yinput; //Sygnał z Żyroskopu / akcelerometru dla osi Y (kąt)
float Xerror, XlastError; //Błąd w obecnej i poprzedniej chwili czasu dla osi X
float Yerror, YlastError; //Błąd w obecnej i poprzedniej chwili czasu dla osi Y
float setPoint = 0; //Docelowy kąt dla osi X i Y
float XDerror; //Suma błędów dla części różniczkującej dla osi X
float YDerror; //Suma błędów dla części różniczkującej dla osi Y

// FUNKCJE STEROWANIA
float Move1 = 1.18; // Współczynnik dla zwiększenia ciągu
float Move2 = 1.18; // Współczynnik dla zmniejszenia ciągu
int throttle, throttle180, FRxy, FLxy, RRxy, RLxy; // Zmienne dla obliczania sygnału sterującego silnikami

void pitchFront(){ // Przechylenie do przodu
    FR.write(throttle180 / Move2);
    FL.write(throttle180 / Move2);
    RR.write(throttle180 * Move1);
    RL.write(throttle180 * Move1);
}

```

```

void pitchBack(){ // Przechylenie do tyłu
    FR.write(throttle180 * Move1);
    FL.write(throttle180 * Move1);
    RR.write(throttle180 / Move2);
    RL.write(throttle180 / Move2);
}

void rollLeft(){ // Przechylenie w lewo
    FR.write(throttle180 * Move1);
    FL.write(throttle180 / Move2);
    RR.write(throttle180 * Move1);
    RL.write(throttle180 / Move2);
}

void rollRight(){ // Przechylenie w prawo
    FR.write(throttle180 / Move2);
    FL.write(throttle180 * Move1);
    RR.write(throttle180 / Move2);
    RL.write(throttle180 * Move1);
}

void yawRight(){ // Obrót w prawo
    FR.write(throttle180 / Move2);
    FL.write(throttle180 * Move1);
    RR.write(throttle180 * Move1);
    RL.write(throttle180 / Move2);
}

void yawLeft(){ // Obrót w lewo
    FR.write(throttle180 * Move1);
    FL.write(throttle180 / Move2);
    RR.write(throttle180 / Move2);
    RL.write(throttle180 * Move1);
}

void setup() {
    // NAWIĄZANIE KOMUNIKACJI PORTU SZEREGOWEGO NA PRĘDKOŚCI 115200
    Serial.begin(115200);

    // KOMUNIKACJA WiFi
    WiFi.mode(WIFI_AP);
    WiFi.softAP(ssid);
    IPAddress myIP = WiFi.softAPIP();
    Serial.print("AP IP address: ");
    Serial.println(myIP);
    // Starting WEB-server
    server.on ( "/", HTTP_handleRoot );

```

```

server.onNotFound ( HTTP_handleRoot );
server.begin();

// NAWIĄZANIE KOMUNIKACJI Z ŻYROSKOPEM / AKCELEROMETREM MPU6050
Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x6B);
Wire.write(0);
Wire.endTransmission(true);

// USTAWIENIE REGULATORÓW ESC
FR.attach(FR_PIN, 1000, 2000); // Numer portu, minimalna szerokość impulsu (ms), maksymalna szerokość impulsu (ms)
FL.attach(FL_PIN, 1000, 2000);
RR.attach(RR_PIN, 1000, 2000);
RL.attach(RL_PIN, 1000, 2000);
FR.write(0); // Wyłączenie silników
FL.write(0);
RR.write(0);
RL.write(0);
delay(2000);
}

void loop() {

    //POŁOŻENIE ŻYROSKOP / AKCELEROMETR MPU6050

    // angleX + przechylenie w prawo
    // angleX - przechylenie w lewo
    // angleY + przechylenie do tyłu
    // angleY - przechylenie do przodu

    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr,14,true);
    AcX = Wire.read()<<8|Wire.read();
    AcY = Wire.read()<<8|Wire.read();
    AcZ = Wire.read()<<8|Wire.read();

    int xAng = map(AcX,minVal,maxVal,-90,90);
    int yAng = map(AcY,minVal,maxVal,90,-90);
    int zAng = map(AcZ,minVal,maxVal,-90,90);

    angleX = RAD_TO_DEG * (atan2(-yAng, -zAng)+PI) - 180;
    angleY = RAD_TO_DEG * (atan2(-xAng, -zAng)+PI) - 180;

    // REGULATOR PD
    Xinput = angleX; // Przypisanie do wejścia regulatora wartości kąta przechylenia

```

```

Yinput = angleY;

// OBLICZENIE CZASU DO PONOWNEJ REGULACJI
now = millis(); // Czas jaki upłynął od włączenia urządzenia
timeChange = now - lastTime; // Różnica między aktualnym czasem a zapamiętanym
if (timeChange >= setTime) // Jeśli zmiana czasu >= od ustawionego czasu dla timera to ustawiony czas minął
{
    // OBLICZENIE WYJŚCIA REGULATORA DLA OSI X
    Xerror = setPoint - Xinput; // Część proporcjonalna (wymagany kąt - aktualny kąt)
    XDerror = (Xerror - XlastError) / timeChange; // Część różniczkująca
    XPout = XKp * Xerror; // Przemnożenie części proporcjonalnej przez nastawę
    XDout = XKd * XDerror; // Przemnożenie części różniczkującej przez nastawę
    Xout = XPout + XDout; // Obliczenie wartości wyjścia regulatora
    XlastError = Xerror; // Ustawienie nowej wartości dla poprzedniego błędu

    // OBLICZENIE WYJŚCIA REGULATORA DLA OSI Y
    Yerror = setPoint - Yinput;
    YDerror = (Yerror - YlastError) / timeChange;
    YPout = YKp * Yerror;
    YDout = YKd * YDerror;
    Yout = YPout + YDout;
    YlastError = Yerror;

    lastTime = now; // Ustawienie nowej wartości dla zapamiętanego czasu
}

// KOMUNIKACJA WiFi
server.handleClient();

// KOMENDY STEROWANIA (Telefon pionowo)
//strzałka w górę F
//strzałka w dół B
//strzałka w lewo L
//strzałka w prawo R
//strzałka w góra prawo I
//strzałka w góra lewo G
//strzałka w dół prawo J
//strzałka w dół lewo H
//przycisk na środku S

// OPÓŹNIENIE PRZED ZMIANĄ OTWARCIA PRZEPUSTNICY NIWELUJĄCE ZAKŁÓCENIA
int lastTime2;
int timeChange2 = now - lastTime2;
int delay1 = 50; // Czas trwania opóźnienia

// PRZYPISANIE FUNKCJI RUCHU DO OTRZYMANEJ KOMENDY
command = server.arg("State"); // Funkcja odczytująca komendę
if (command == "R") pitchFront();

```

```

else if (command == "L") pitchBack();
else if (command == "F") rollLeft();
else if (command == "B") rollRight();
else if (command == "I") yawRight();
else if (command == "J") yawLeft();
//else if (command == "0") throttle = 0; // Nie korzystać z położenia 0
else if (command == "1" && timeChange2 >= delay1) throttle = 0;
else if (command == "2" && timeChange2 >= delay1) throttle = 50;
else if (command == "3" && timeChange2 >= delay1) throttle = 70;
else if (command == "4" && timeChange2 >= delay1) throttle = 90;
else if (command == "5" && timeChange2 >= delay1) throttle = 100;
else if (command == "6" && timeChange2 >= delay1) throttle = 120;
else if (command == "7" && timeChange2 >= delay1) throttle = 130;
else if (command == "8" && timeChange2 >= delay1) throttle = 140; // MAX 200
//else if (command == "9") throttle = 200; // Nie korzystać z położenia 9

// ZMIANA ROZDZIELCZOŚCI SYGNAŁU PWM Z 255 NA 180 DLA PRZEPUSTNICY
throttle180 = map(throttle, 0, 255, 0, 180);

// OBLICZENIE SYGNAŁU PWM DLA POSZCZEGÓLNEGO SILNIKA Z UWZGLĘDNIENIEM PRZEPUSTNICY I POCHYLENIA
FRxy = throttle + Yout - Xout;
FLxy = throttle + Yout + Xout;
RRxy = throttle - Yout - Xout;
RLxy = throttle - Yout + Xout;

// ZABEZPIECZENIE PRZED PRZEKROCZENIEM ZAKRESU PWM
if (FRxy > 255) FRxy = 255;
if (FLxy > 255) FLxy = 255;
if (RRxy > 255) RRxy = 255;
if (RLxy > 255) RLxy = 255;
if (FRxy < 0) FRxy = 0;
if (FLxy < 0) FLxy = 0;
if (RRxy < 0) RRxy = 0;
if (RLxy < 0) RLxy = 0;

// ZMIANA ROZDZIELCZOŚCI SYGNAŁU PWM Z 255 NA 180 DLA STEROWANIA SILNIKAMI
FRxy = map(FRxy, 0, 255, 0, 180);
FLxy = map(FLxy, 0, 255, 0, 180);
RLxy = map(RLxy, 0, 255, 0, 180);
RRxy = map(RRxy, 0, 255, 0, 180);

// USTAWIENIE PRĘDKOŚCI SILNIKÓW NA OBLICZONĄ
FR.write(FRxy);
FL.write(FLxy);
RR.write(RRxy);
RL.write(RLxy);

```

```

//WYŚWIETLANIE PARAMETRÓW NA MONITORZE PORTU SZEREGOWEGO
Serial.print("angleX "); // Przechylenie lewo+, prawo- (stopnie)
Serial.print(angleX);

Serial.print(" ");
Serial.print("angleY "); // Przechylenie tył+, przód- (stopnie)
Serial.print(angleY);

Serial.print(" ");
Serial.print("FR "); // Silnik przód prawy, sygnał PWM (180)
Serial.print(FRxy);

Serial.print(" ");
Serial.print("FL "); // Silnik przód lewy, sygnał PWM (180)
Serial.print(FLxy);

Serial.print(" ");
Serial.print("RR "); // Silnik tył prawy, sygnał PWM (180)
Serial.print(RRxy);

Serial.print(" ");
Serial.print("RL "); // Silnik tył lewy, sygnał PWM (180)
Serial.print(RLxy);

Serial.print(" ");
Serial.print("Throttle "); // Przepustnica PWM (180)
Serial.print(throttle180);
Serial.print("\n");

lastTime2 = now; // Zapamiętany czas dla redukcji zakłóceń

}

// KOMUNIKACJA WiFi
void HTTP_handleRoot(void) {

if( server.hasArg("State") ){
    Serial.println(server.arg("State"));
}
server.send ( 200, "text/html", "" );
delay(1);
}

```


3. UKŁAD MECHANICZNY

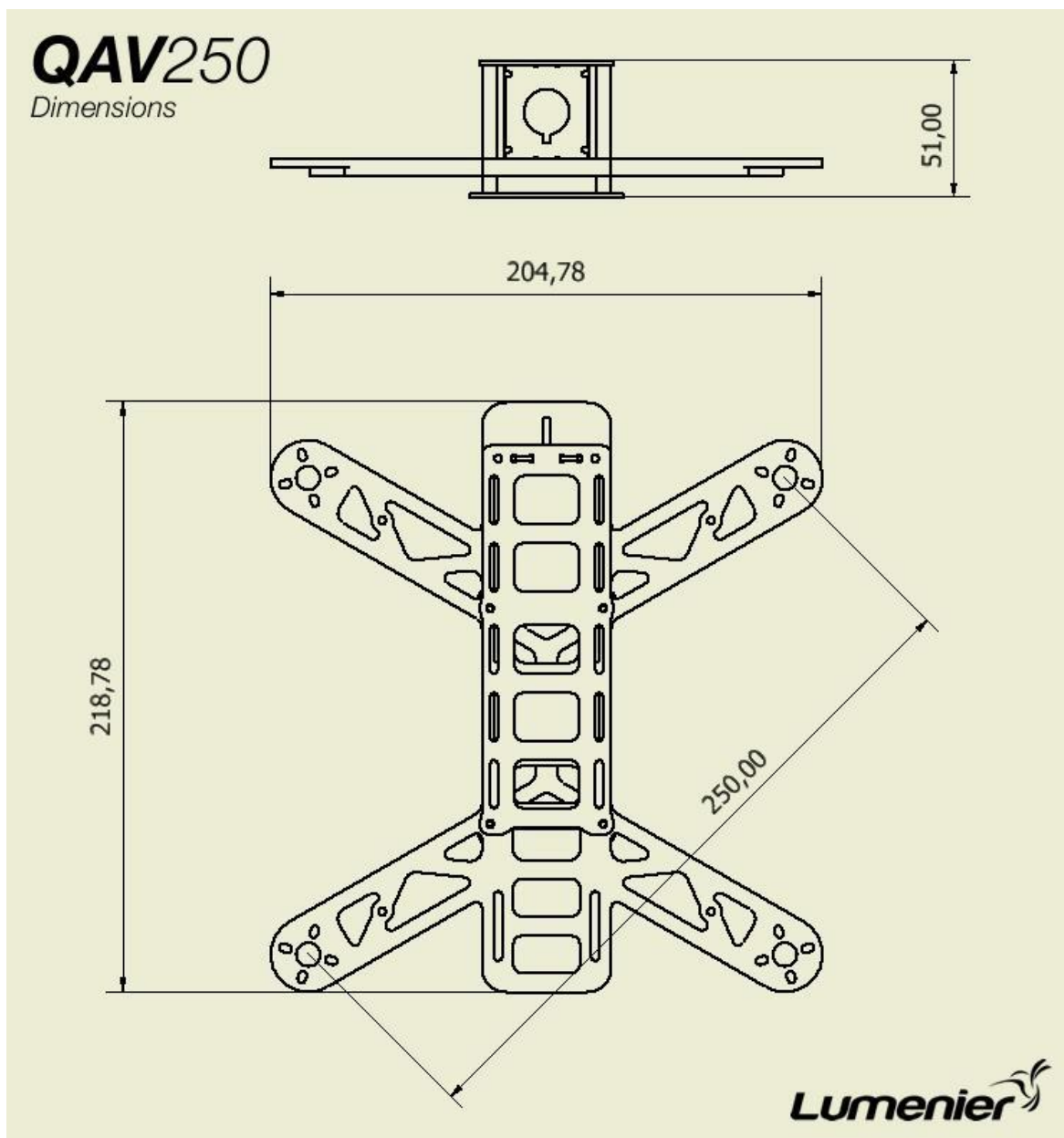
Układ mechaniczny stanowi rama wraz z zespołami napędowymi. Są to elementy na które działają największe siły i od ich właściwości fizycznych zależą możliwości lotu. Parametry ramy muszą być odpowiednio dopasowane do parametrów zespołu napędowego.

3.1. Projektowanie ramy

Rama jest podstawowym elementem każdego wielowirnikowca ma za zadanie zapewnić optymalne rozmieszczenie podzespołów, sztywność podczas lotu i ochronę podzespołów podczas ewentualnych kolizji. Podstawowymi parametrami podczas dobierania ramy jest jej klasa, przewidziana ilość zespołów napędowych, a także rozstaw otworów przeznaczonych dla ich montażu. Klasa stanowi o wielkości ramy, jest to odległość między dwoma przeciwległymi zespołami napędowymi mierzona po przekątnej podawana w milimetrach. Wybór ramy o danej klasie zależy od rozmiaru śmigieł. Ważne, aby odległość między zespołami napędowymi była na tyle duża, by śmigła nie kolidowały ze sobą.

Tab. 3.1. Przewidziana klasa ramy dla danego rozmiaru śmigieł [opracowanie własne]

Klasa ramy	Rozmiar śmigieł
180	Do 4"
250	5"-6"
330	6"-8"
450	10"-12"

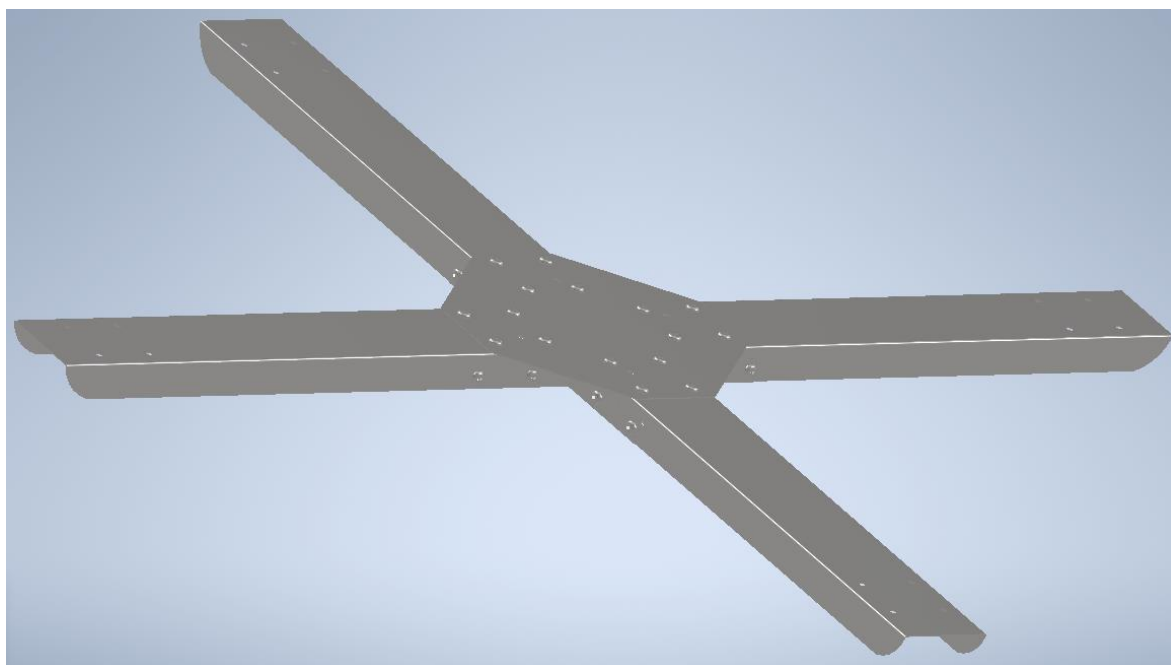


Rys. 3.1. Przykładowa rama klasy 250 [26]

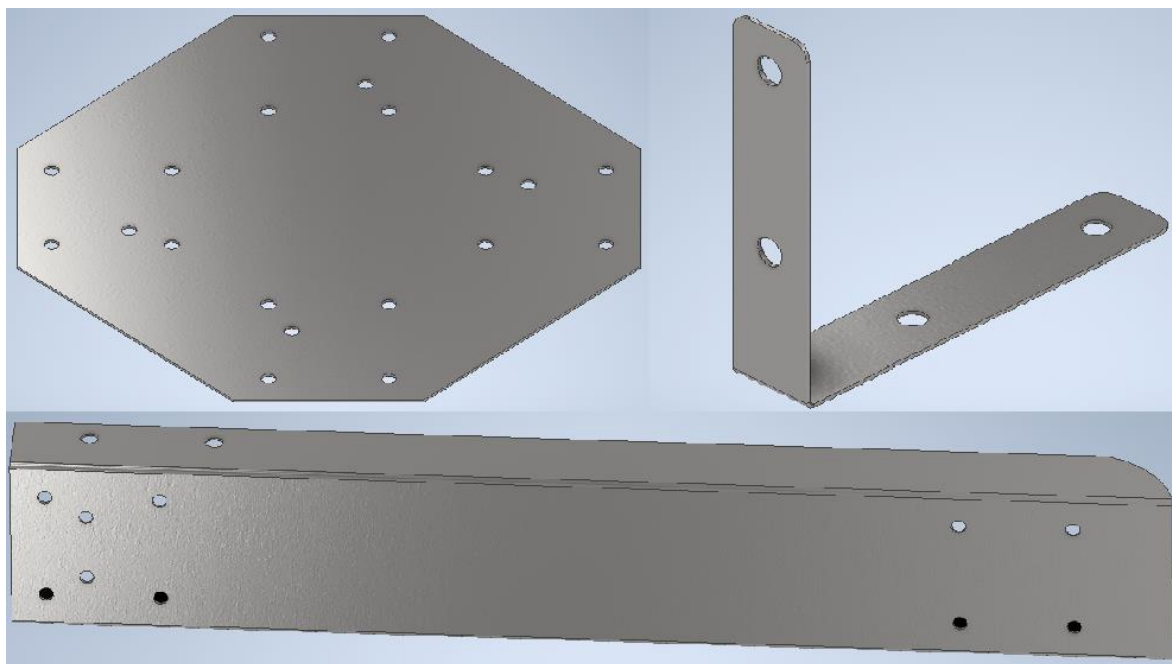
Materiały z jakich wykonuje się ramy dla multikopterów to między innymi: drewno, tworzywa sztuczne, aluminium, włókno węglowe. Włókno węglowe jest wytrzymałe, bardzo lekkie i sztywne. Jego wady to: wysoka cena, oraz podczas uszkodzeń może zacząć się kruszyć. Do naprawy włókna węglowego używa się żywicy epoksydowej lub kleju cyjanoakrylowego, w tym materiale nie można wiercić ponieważ łatwo pęka. Aluminium z kolei posiada zalety włókna węglowego, jest nieco cięższe i trochę mniej sztywne. Ma

przewagę nad włóknem węglowym ze względu na niższą cenę, oraz możliwość obróbki, co upraszcza naprawę uszkodzonej ramy [15].

Do przeprowadzenia pełnych testów napisanego kodu oraz wykonanego układu elektrycznego konieczne było zaprojektowanie oraz wykonanie ramy prototypowej z łatwo dostępnego materiału. Projekt został wykonany w programie Autodesk Inventor 2023. Do jego wykonania zastosowano konstrukcje blachowe które następnie zostały połączone w zespół. Rama została zaprojektowana w standardzie klasy 450, oraz tak aby możliwe było zamontowanie silników o rozstawie otworów 33 x 33mm, a także płytki uniwersalnej 64 x 44mm.

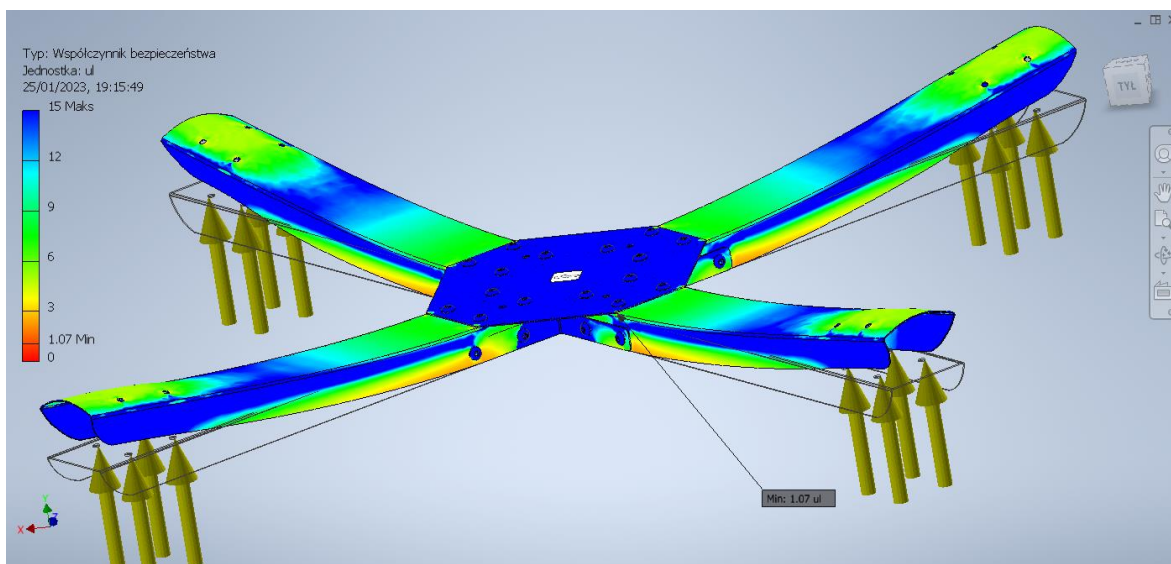


Rys. 3.2. Zaprojektowana rama prototypowa (zespół)



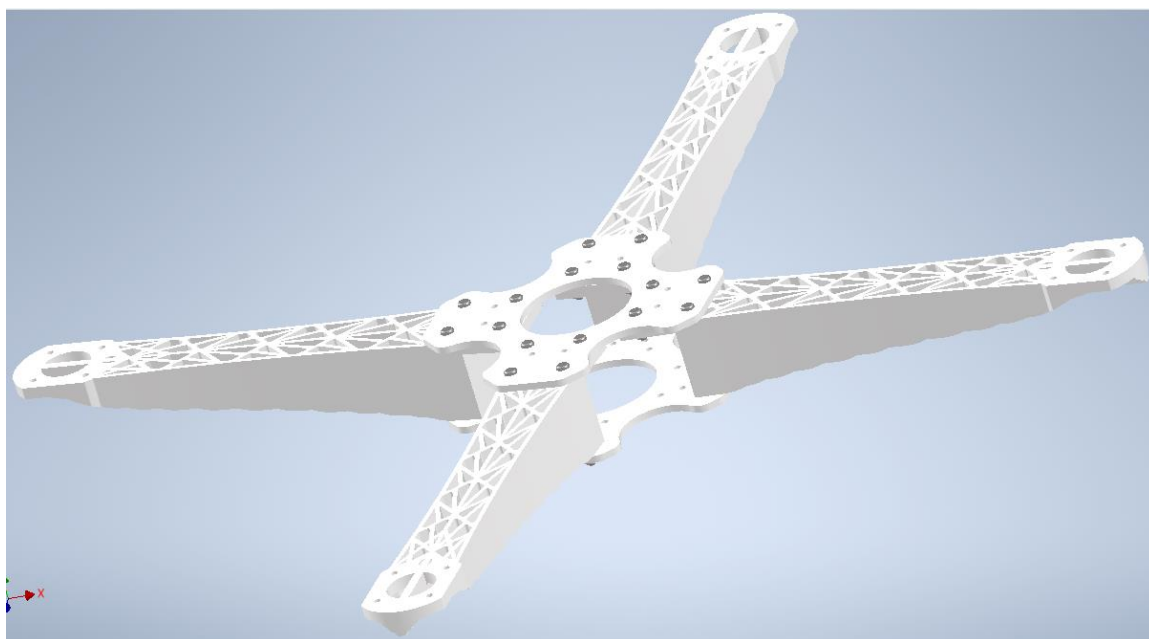
Rys. 3.3. Zaprojektowane części ramy prototypowej

Przeprowadzona analiza wytrzymałościowa ramy prototypowej wykazała, że konstrukcja wytrzymuje maksymalnie siłę 180N podzieloną na 4 ramiona. Wskazuje na to wartość współczynnika wytrzymałości która jest większa od 1. Środek konstrukcji został unieruchomiony, a siła została przyłożona w miejscach montażowych zespoły napędowe. Siła 180N oznacza, że konstrukcja wytrzyma nacisk około 18 kg podzielony na 4 ramiona. Siła ta spowodowała, że końce ramion przemieszczają się maksymalnie o około 0,89mm w kierunku działania siły. Cała konstrukcja ma ważyć około 0,330 kg i zostać wykonana z blachy stalowej ocynkowanej o grubości 0,5mm.

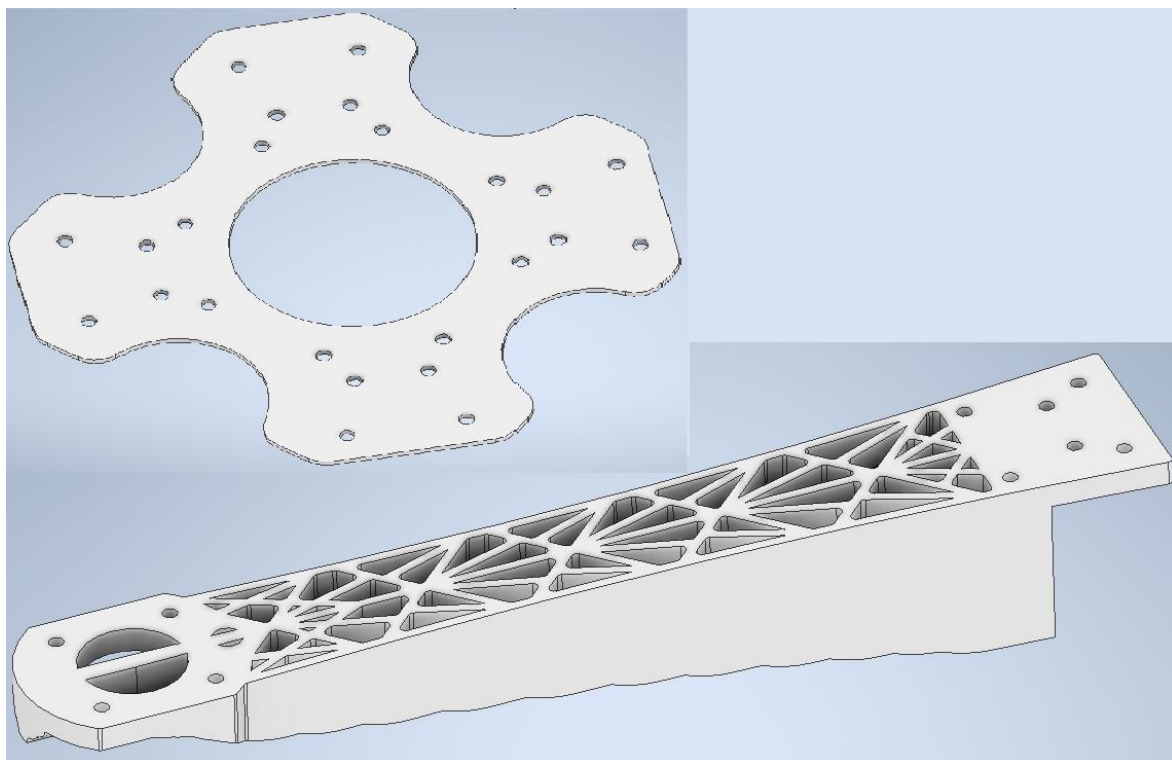


Rys. 3.4. Analiza wytrzymałościowa zespołu ramy prototypowej

Docelowa rama została zaprojektowana w tych samych standardach co prototypowa, ponadto zastosowana geometria pozwala na jej proste wykonanie w technologii druku 3D. Posiada dwie podstawy co umożliwia bardziej ergonomiczne rozmieszczenie podzespołów. Składa się z zaledwie dwóch projektowanych części (ramiona oraz podstawy) do jej złożenia potrzebne są 4 ramiona oraz 2 podstawy.

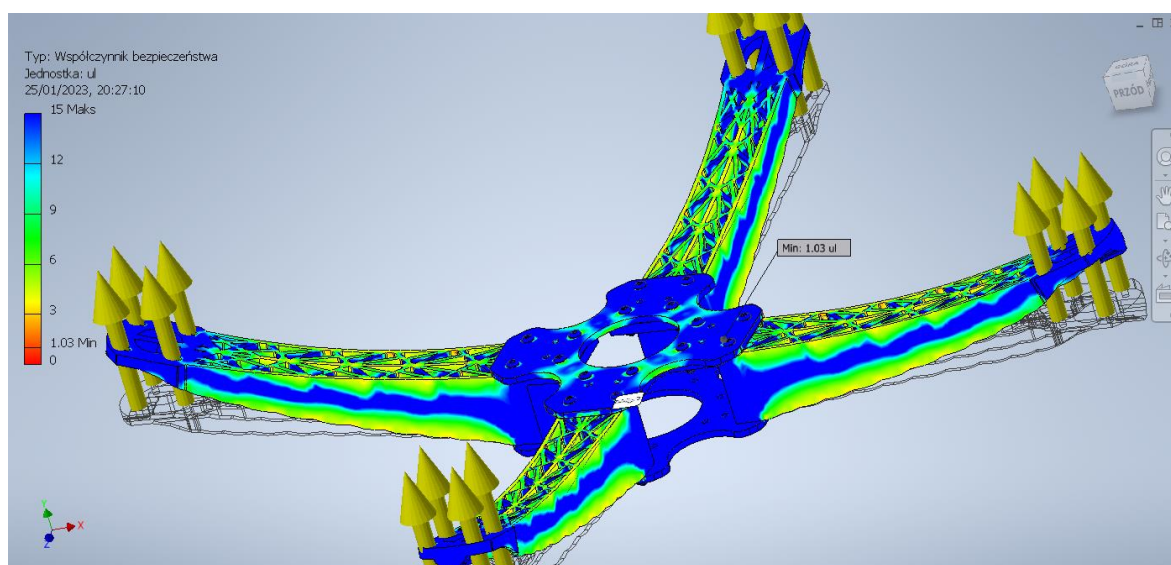


Rys. 3.5. Zaprojektowana rama docelowa (zespół)

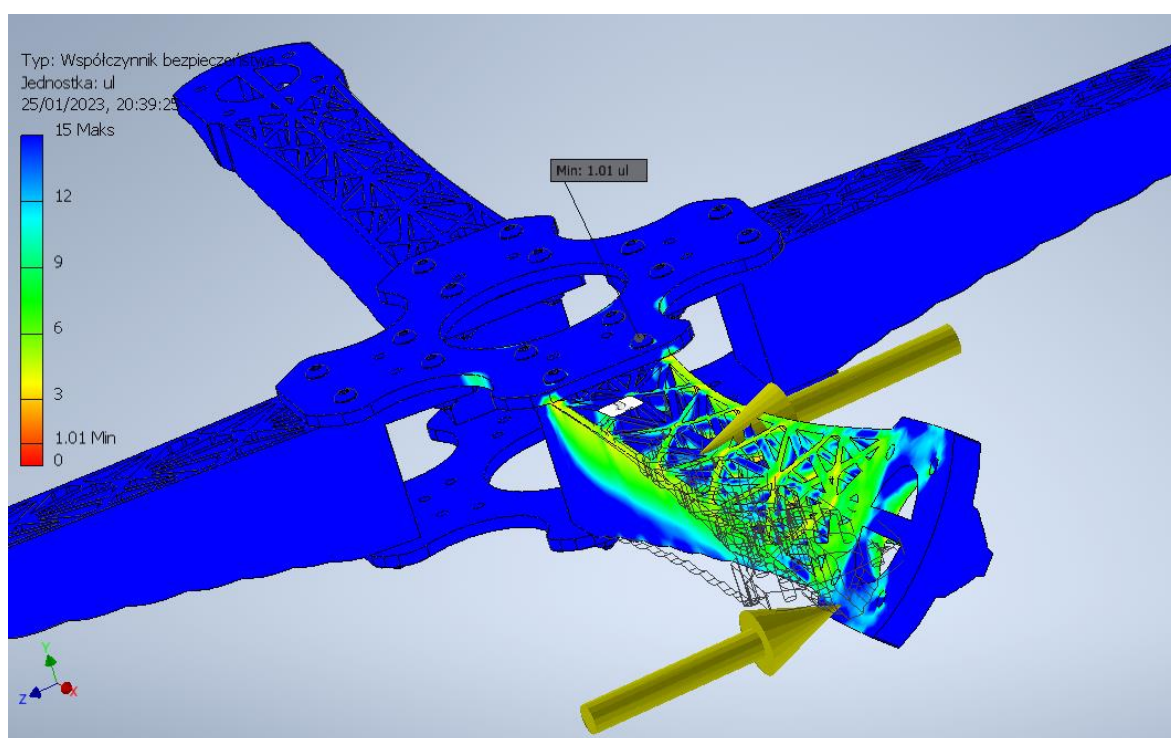


Rys. 3.6. Zaprojektowane części ramy docelowej

Przeprowadzona analiza wytrzymałościowa docelowej ramy wykazała, że konstrukcja ta również wytrzymuje siłę do 180N tak jak wariant z blachy, a przy tym jest od niego lżejsza. Dla tej symulacji maksymalne przemieszczenie końców ramion wynosi 3,9mm. Dodatkowo przeprowadzono symulację dla siły skręcającej jedno ramię. Wypadkowa obydwu sił skręcających wynosiła 60N. Rama była na granicy pęknięcia ponieważ współczynnik wytrzymałości wynosił 1,01. Siła ta spowodowała, że koniec ramiona przemieszcza się maksymalnie o około 3,5mm w kierunku działania sił. Wyniki projektowe są zadowalające, ponieważ wytrzymałość ramy jest dość duża, a masa profesjonalnej ramy tej klasy wynosi 0,282kg. Zaprojektowana konstrukcja ramy ma ważyć około 0,304 kg i zostać wykonana z ABS w technologii druku 3D.



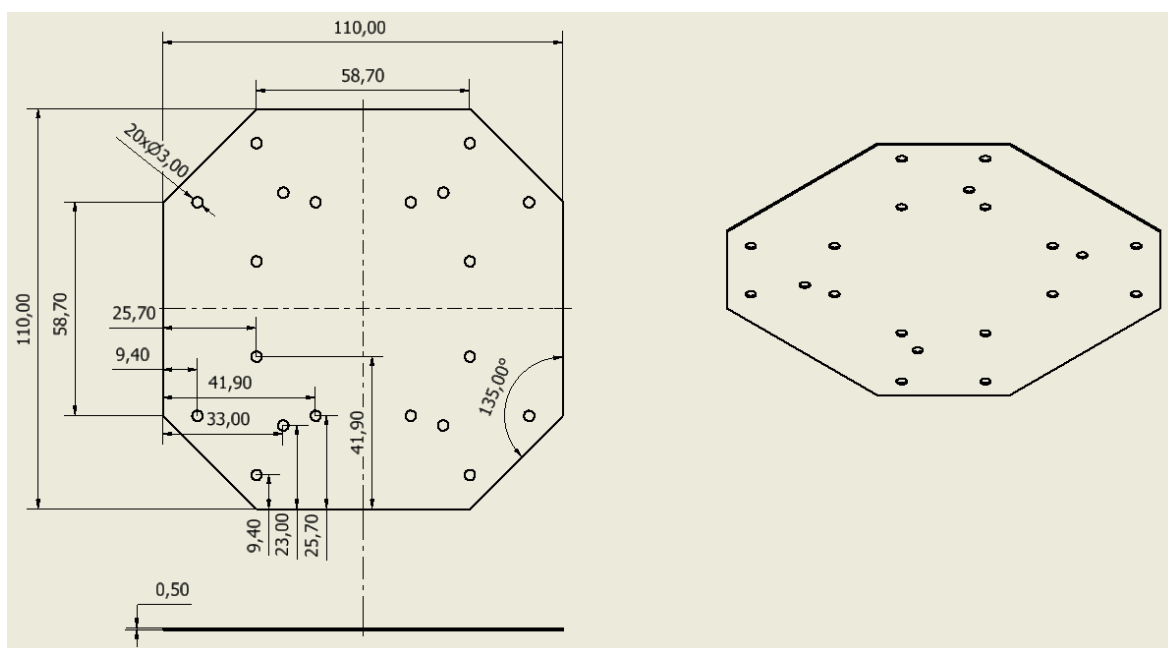
Rys. 3.7. Analiza wytrzymałościowa zespołu ramy docelowej dla siły skierowanej do góry



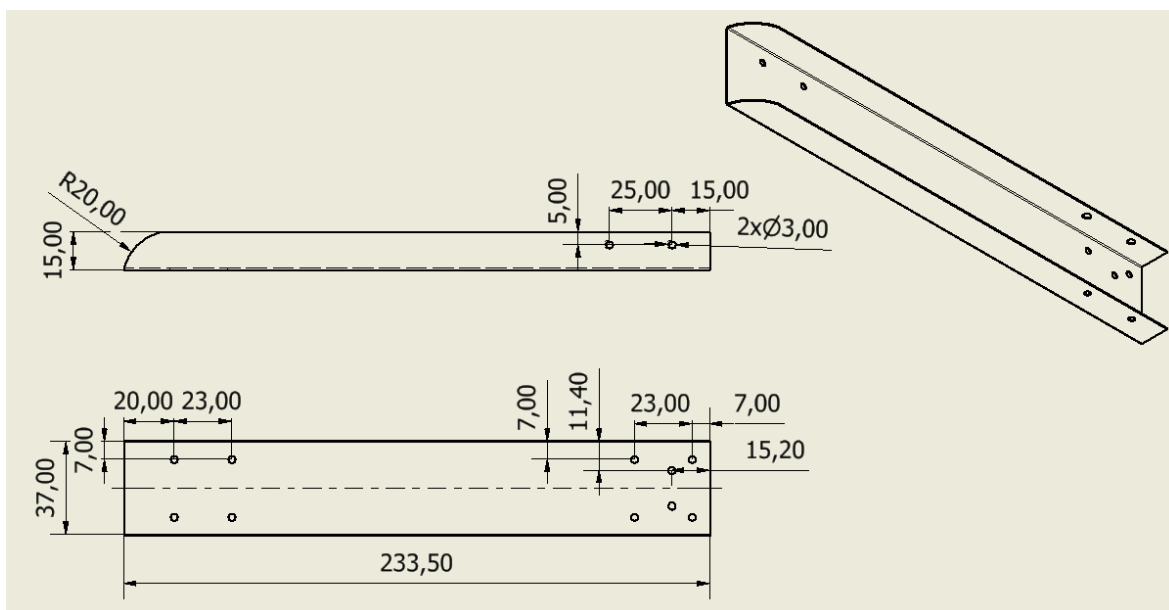
Rys. 3.8. Analiza wytrzymałościowa zespołu ramy docelowej dla siły skręcającej ramie

3.2. Wykonanie ramy

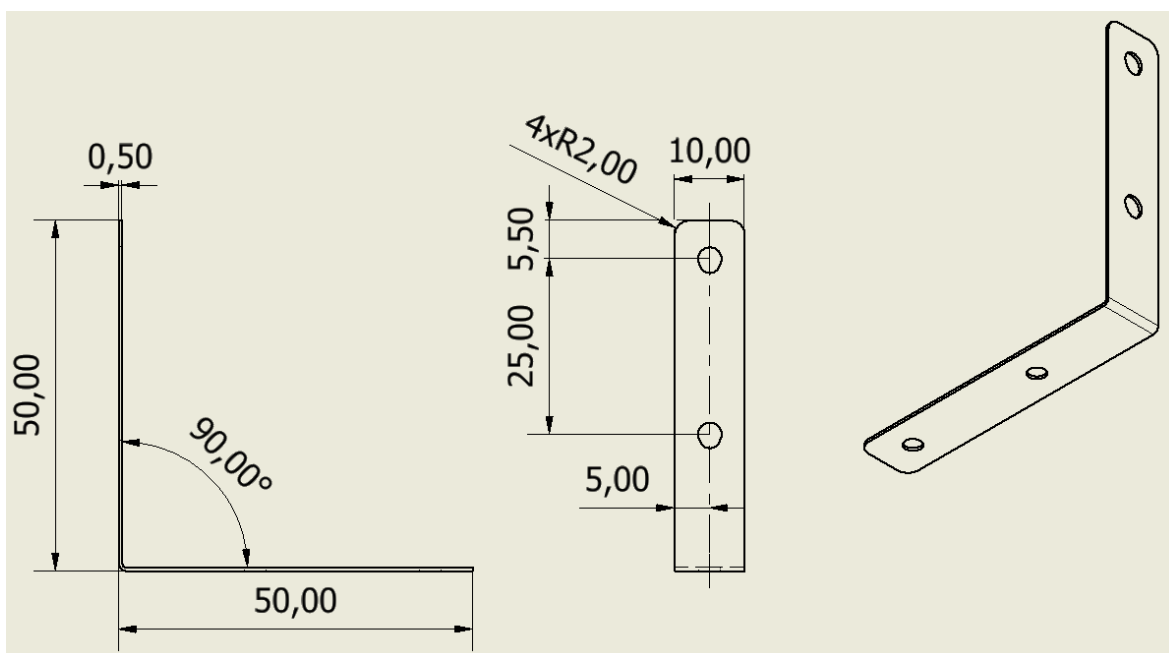
Wykonanie części ramy prototypowej odbyło się poprzez odpowiednie wycięcie i wygięcie stalowej ocynkowanej blachy o grubości 0,5mm. Części zostały połączone za pomocą zestawu śrub M3x6, podkładek M3 oraz nakrętek M3.



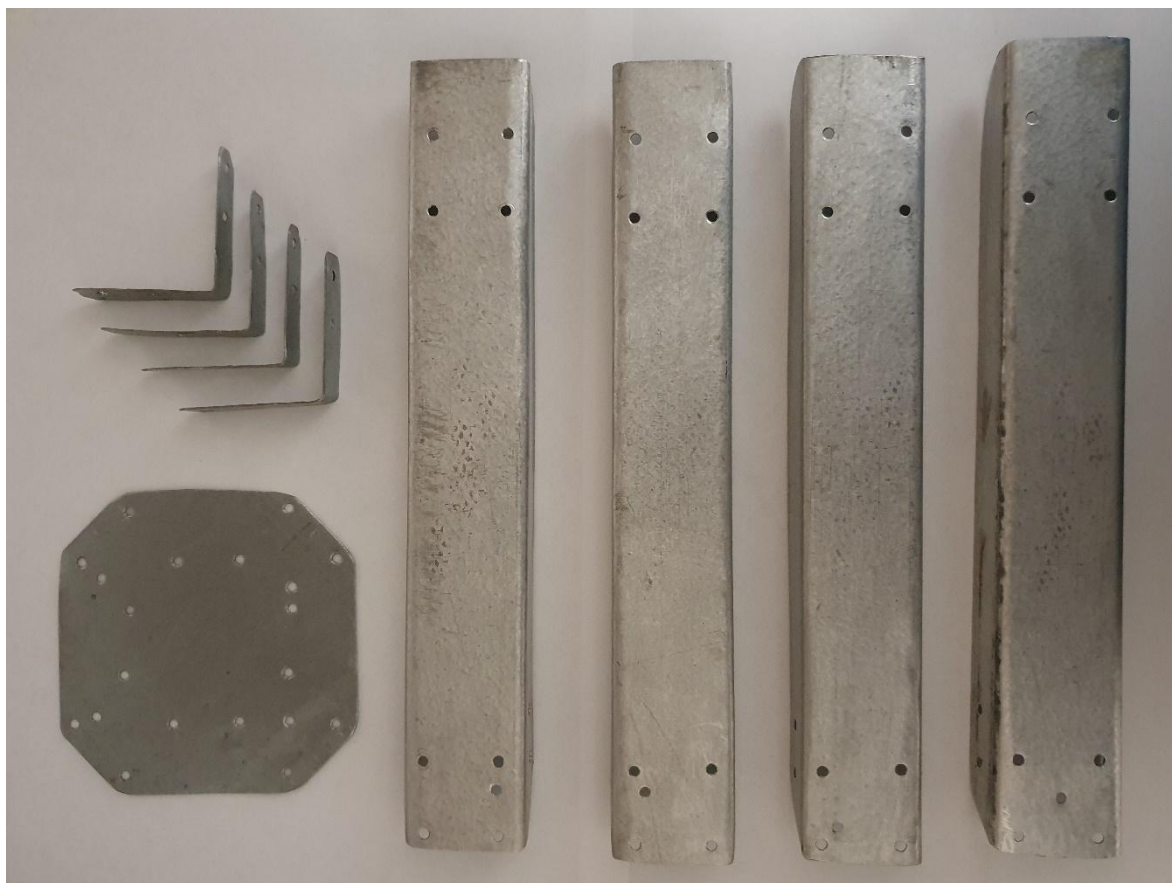
Rys. 3.9. Schemat wykonania części środkowej dla ramy prototypowej



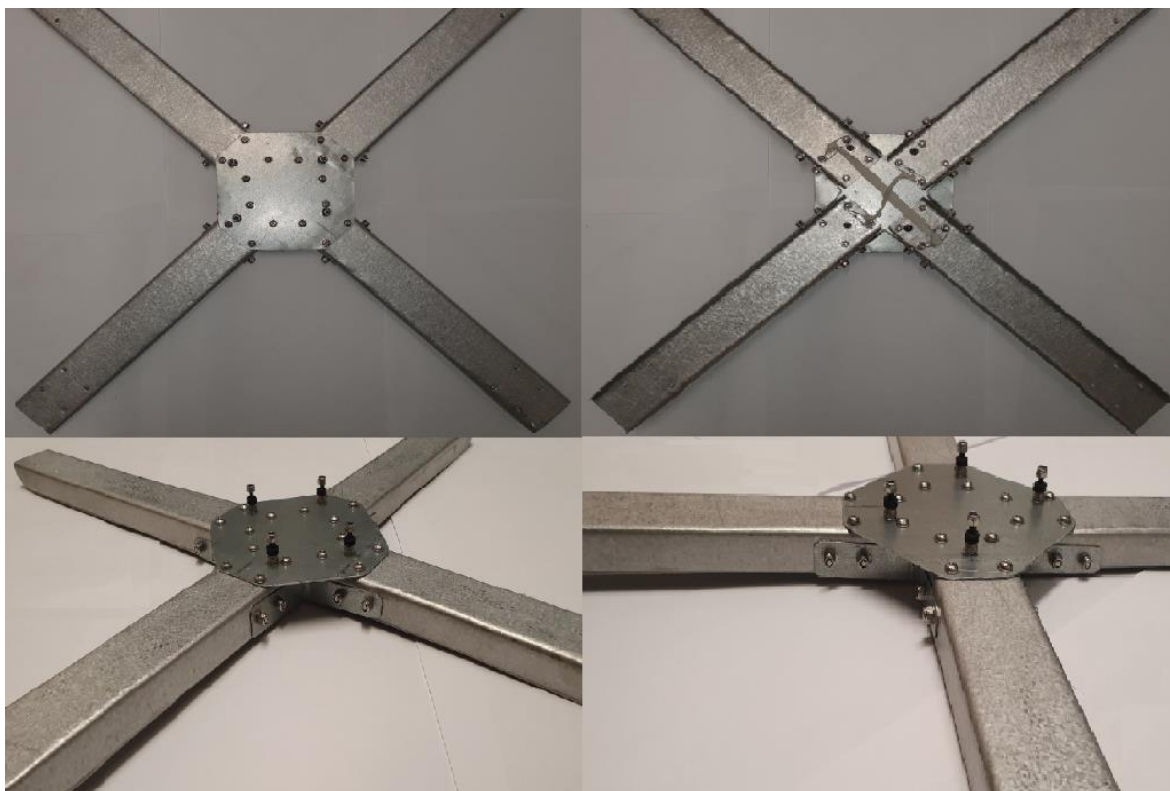
Rys. 3.10. Schemat wykonania ramienia dla ramy prototypowej



Rys. 3.11. Schemat wykonania kątownika dla ramy prototypowej



Rys. 3.12. Wykonane części ramy prototypowej.



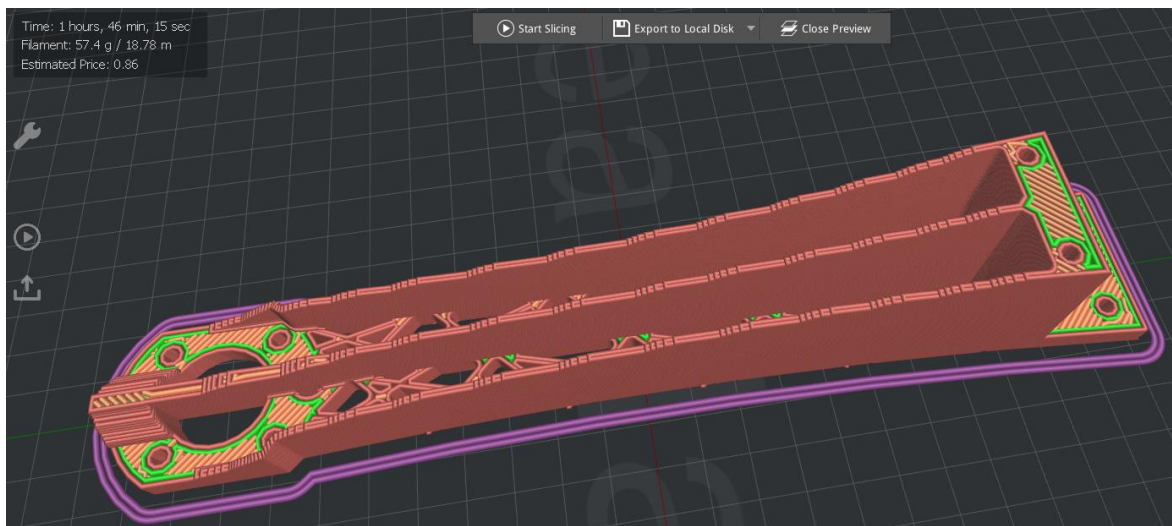
Rys. 3.13. Zmontowana rama prototypowa.

Rama o docelowej konstrukcji miała zostać wykonana z materiału ABS w technologii druku 3D, jednak przez awarię drukarki nie zostało to zrealizowane. Części zaprojektowane w programie Autodesk Inventor 2023 zostały zapisane jako format .stl. Do przekonwertowania plików .stl na g-code zastosowano oprogramowanie ideaMaker, można je pobrać za darmo ze strony producenta: <https://www.raise3d.com/ideamaker/> [dostęp: 31.01.2023 r.].

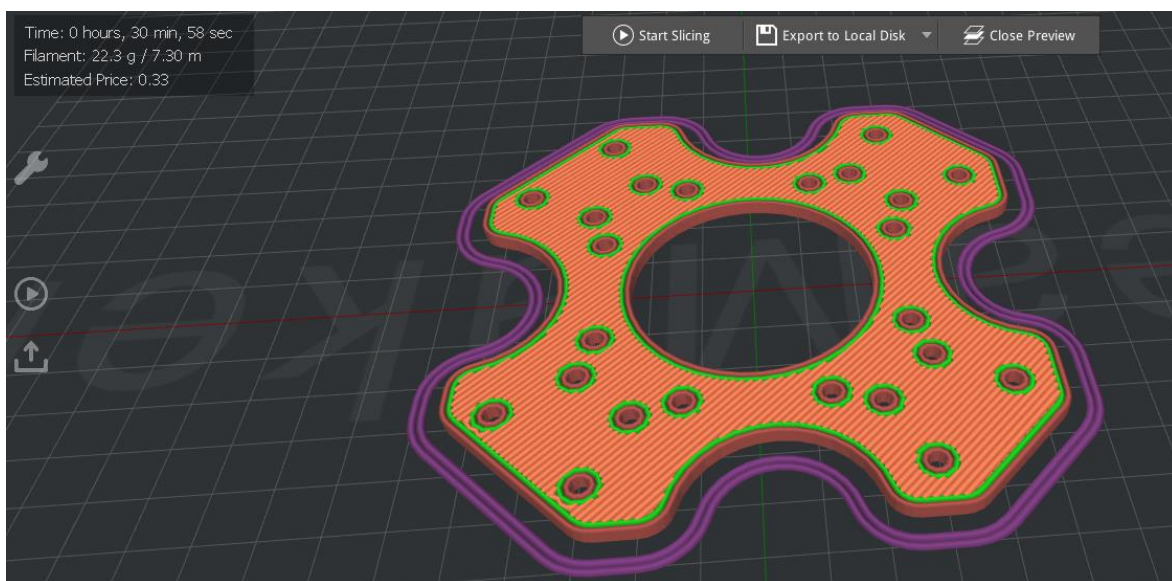
Drukowanie miało charakteryzować się z następującymi parametrami:

- średnica dyszy: 0,6mm
- wysokość warstwy: 0,4mm
- szerokość warstwy: 1,2mm
- gęstość wypełnienia 30%

Czas drukowania jednego ramiona wynosi około 1 godziny i 46 minut, wykorzystane zostanie około 57,4g filamentu. Czas drukowania jednej podstawy wynosi około 31 minut, wykorzystane zostanie około 22,3g filamentu. Reasumując dla całego zespołu ramy wykorzystane zostanie około 274,2g filamentu. Koszt wydruku całego zespołu biorąc pod uwagę jedynie ceny filamentu ABS z dnia 26.01.2023 wyniosłby około 22 złote.



Rys. 3.14. Podgląd wydruku ramiona w programie ideaMaker



Rys. 3.15. Podgląd wydruku podstawy w programie ideaMaker

3.3. Rozmieszczenie elementów na ramie

Rozmieszczenie elementów na ramie prototypowej odbyło się za pomocą zespołów śrub, podkładek, nakrętek w standardzie M3, oraz opasek zaciskowych. Na górnej części ramy zostały rozmieszczone: płytki PCB, regulatory ESC, oraz zespoły napędowe, a także łączące je przewody. Płytki PCB jak i zespoły napędowe zostały zamocowane za pomocą połączenia śrubowego w oparciu o przeznaczone do tego otwory. Regulatory ESC wraz z wyprowadzonymi z nich przewodami znajdują się pomiędzy płytką PCB a zespołami napędowymi i są zamocowane za pomocą opasek zaciskowych. Za pomocą kolejności podłączenia przewodów między regulatorami ESC, a zespołami napędowymi decyduje się o kierunku obrotu silników, dla tego modelu ustalone są następujące kierunki obrotów:

- przedni prawy silnik - obrót w lewo
- przedni lewy silnik - obrót w prawo
- tylny prawy silnik - obrót w prawo
- tylny lewy silnik - obrót w lewo

Gdzie kierunek przedni jest określany przez umiejscowienie modułu MPU6050 na płytce PCB.

Od dolnej strony ramy znajdują się: źródło zasilania z wyprowadzonymi przewodami oraz włącznik. Akumulator został zamocowany za pomocą gumy krawieckiej w środkowej części konstrukcji. Włącznik jak i przewody zostały zamocowane za pomocą opasek zaciskowych. Przewody zostały poprowadzone przez lukę znajdującą się pomiędzy kątownikami a podstawą ramy.



Rys. 3.16. Rozmieszczenie elementów na ramie

3.4. Przeprowadzenie testów

Ponieważ docelowa konstrukcja ramy nie została wykonana testy zostały przeprowadzone dla konstrukcji wykonanej na ramie prototypowej. Pierwszą rzeczą jaką zaobserwowano jeszcze podczas testu na diodach były duże zakłócenia podczas sterowania poziomem otwarcia przepustnicy występujące zwykle na poziomie minimum a rzadziej na maksimum, objawiało się to nagłym zwiększeniem lub zmniejszeniem mocy co było kłopotliwe podczas testów z już zamontowanymi śmigłami. Zasięg na świeżym powietrzu wynosi około 20 metrów, co dla przeprowadzanych testów było wystarczające. Cały model

ważył około 1kg co dawało mu stosunek ciągu do masy na poziomie 3:1. Taka wartość powinna pozwolić na efektywne latanie, w praktyce jednak okazało się, że przy większych obrotach silników całość wpadała w duże wibracje co powodowało turbulencje, a następnie spadanie kwadrokoptera. Sterowanie za pomocą interfejsu powoduje, że model zamierza poruszyć się we wskazanym mu kierunku, jednak efekt opisany powyżej to uniemożliwia. Podczas ręcznego przechylenia statku powietrznego odczuwa się efekt regulacji mocy silników mający na celu jego powrót do pozycji poziomej. Zmiany nastaw regulatora PD jak i parametrów ruchu w dużym stopniu wpływają na zachowanie modelu. Reasumując wykonany kwadrokopter jest w stanie się unieść, jednak po kilku sekundach wpada w turbulencje i spada.



Rys. 3.17. Pomiar masy wykonanego kwadrokoptera

PODSUMOWANIE

Podczas wykonywania pracy inżynierskiej udało się zaprojektować oraz wykonać kwadrokopter. Przeprowadzone na nim testy wykazały zakłócenia w komunikacji między smartfonem a kontrolerem lotu. Przyczyną tego może być błąd w aplikacji sterującej lub zakłócenia sieci WiFi. Niewielki zasięg sygnału może zostać zwiększony przez zastosowanie anteny zewnętrznej podłączonej do modułu ESP32. Poprawę z pewnością przyniosłoby zastosowanie lepszej aplikacji sterującej lub w ostateczności zmiana metody komunikacji na radiową. Przyczyną zaobserwowanych wibracji okazała się być mało sztywna rama prototypowa, zastosowanie docelowej konstrukcji wykonanej w technologii druku 3D z pewnością zniwelowałoby to zjawisko. Działanie systemu sterowania było prawidłowe, nie wliczając zakłóceń kwadrokopter reagował poprawnie na wszystkie komendy bez widocznego opóźnienia. Urządzenie odczytujące orientację modelu działało bardzo szybko i precyzyjnie, największą trudność stanowiło odpowiednie dobranie nastaw regulatora PD odpowiedzialnych za stabilizację koptera. Całkowita masa obiektu wynosiła 1005g, zastosowanie docelowej ramy pozwoliło ją nieco zmniejszyć. Stosunek maksymalnego ciągu do masy na poziomie około 3:1 jest dobrym wynikiem. Po wprowadzeniu powyższych zmian i osiągnięciu poprawnego efektu lotu w przyszłości korzystne będzie zastosowanie akumulatora o większej pojemności oraz kamery.

BIBLIOGRAFIA

Książki i artykuły naukowe:

- [1] Andrzej Jaskulski: Autodesk Inventor Professional 2021 PL / 2021+ / Fusion 360. Metodyka projektowania. Helion, Gliwice, 2020.
- [2] Damian Szymański: ESP8266 - co warto wiedzieć?. FORBOT, Poznań, 2020.
- [3] Dejan: Arduino Brushless Motor Control Tutorial ESC BLDC. How To Mechatronics, San Francisco, 2019.
- [4] EBMiA: Silnik bezszczotkowy BLDC – co to jest, jak działa, zastosowanie, zalety. Wiedza EBIMA, Augustów, 2021.
- [5] F3M: Geneza powstania dronów. F3M, Warszawa, 2019.
- [6] Federal Aviation Administration: Pilot's Handbook of Aeronautical Knowledge. Skyhorse Publishing, Oklahoma City, 2017.
- [7] Herbert Gnaś: Historia dronów – od starożytności do projektów Tesli. Twoja Historia, Poznań, 2022.
- [8] Korneliusz Jarzębski: NodeMCU v2 - idealna płytką dla IoT. Dev Jarzębski, Gliwice, 2015.
- [9] Michał Kurzela: Jak działają szczotkowe silniki DC? Budowa i pomiary. Damian Szymański, Poznań, 2019.
- [10] Opracowanie zbiorowe: Tak To Działa. Drony. Bizbi Media SA, Lipowo, Nr 1/2016.
- [11] Oskar Pacelt: Prosto o Arduino IDE – co to jest?. Botland, Gola, 2021.
- [12] PM: Śmigła do quadcoptera. PORTAL DRONY - STUDIO 5S, Warszawa, 2017.
- [13] Ryszard Szymaniak: Moduł żyroskopu i akcelerometru MPU-6050. Elektronika praktyczna, Warszawa, 2021.
- [14] S.: Regulatory PID na Arduino - wstęp. Starter kit, Warszawa, 2013.
- [15] Ty Audronis: Drony wprowadzenie. Helion, Gliwice, 2015.

Internet:

- [16] <https://aeromind.pl/Budowa-drona-ccms-pol-52.html> [dostęp: 31.01.2023 r.]

- [17] <http://zrobsobiedrona.blogspot.com/2014/06/2-budujemy-quadcopter-jak-to-dziala.html> [dostęp: 31.01.2023 r.]
- [18] https://rc-planeta.pl/4096-thickbox_default/smigla-gemfan-10x45-cwccw-black-para-smigiel-1045.jpg [dostęp: 31.01.2023 r.]
- [19] <https://abc-rc.pl/Podstawy-wiedzy-o-smiglach-modelarskich-blog-pol-1542270254.html> [dostęp: 31.01.2023 r.]
- [20] <https://abc-rc.pl/product-pol-6764-Silnik-ABC-Power-A2212-1000KV-2-3S-135W-ciag-820g.html> [dostęp: 31.01.2023 r.]
- [21] <https://dronemanya.com/pl/post/co-to-jest-regulator-predkosci>
[dostęp: 31.01.2023 r.]
- [22] <https://www.gimmik.net/blog/dobor-regulatora-obrotow-silnika-esc-modeli-rc/>
[dostęp: 31.01.2023 r.]
- [23] <https://a.allegroimg.com/s512/031c56/ece67ad646cbb8821fe7d0520616/REGULATOR-30A-40A-UBEC-5V-2A-2S-4S-7-4V-14-8V-RC-EAN-GTIN-18350100>
[dostęp: 31.01.2023 r.]
- [24] <https://dronavista.pl/pl/akcesoria/8028-kontroler-lotu-f4-1s-12a-aio-elrs-24g-betafpv.html#mz-expanded-view-1086059022504> [dostęp: 31.01.2023 r.]
- [25] <https://how2electronics.com/measure-tilt-angle-mpu6050-arduino/>
[dostęp: 31.01.2023 r.]
- [26] https://farm8.staticflickr.com/7408/10979127516_775c5fe166_o.jpg
[dostęp: 31.01.2023 r.]
- [27] <https://i.dronemanya.com/%D0%91%D0%9A-%D0%BC%D0%BE%D1%82%D0%BE%D1%80-inrunner-outrunner-bcfb0.png>
[dostęp: 31.01.2023 r.]
- [28] <https://steemitimages.com/p/2gsigna1uruvUuS7ndh9YqVwYGPLVszbFLwwpAYXa4cjbfbxrw9ExfSUxdzvniw9ynCNCib37sJiidQoFmknNTXyuV1VgxeqzaaRDJxoESCGRikDU?format=match&mode=fit&width=640> [dostęp: 31.01.2023 r.]
- [29] <https://image.ceneostatic.pl/data/products/93125258/i-redox-pakiet-li-pol-2200mah-30c-3s-11-1v.jpg> [dostęp: 31.01.2023 r.]

Temat pracy w języku polskim: Projekt i wykonanie bezzałogowego statku latającego w oparciu o mikrokontroler ESP32

Temat pracy w języku angielskim: Design and implementation of an unmanned aerial vehicle based on the ESP32 microcontroller

Słowa kluczowe: bezzałogowy statek latający, dron, wielowirnikowiec, multikopter, ESP32, NodeMCU v2, ESP8266

Streszczenie:

Praca inżynierska opiera się na zaprojektowaniu oraz wykonaniu bezzałogowego statku latającego w formie kwadrokoptera, z zastosowaniem mikrokontrolera z rodziny ESP32. Część teoretyczna porusza temat zasady działania wielowirnikowców, a także przedstawia podstawowe podzespoły z jakich są one zbudowane. Zostały tam omówione zasady którymi kierowano się podczas doboru poszczególnych podzespołów. W dalszej części skupiono się na oprogramowaniach komputerowych, które przyczyniły się do zrealizowania projektu. W części poświęconej układowi sterowania zostały opisane aspekty projektowania, a następnie wykonania układu elektrycznego, a także proces wykonywania oprogramowania sterującego układem. Rozdział układ mechaniczny skupia się na zaprojektowaniu oraz wykonaniu ramy, złożeniu wszystkich podzespołów w całość oraz przeprowadzeniu testów wykonanego modelu. Finalne udało się wykonać częściowo działający prototyp bezzałogowego statku latającego. Analiza wyników została omówiona w podsumowaniu, a także części poświęconej testowaniu kwadrokoptera.

Summary:

The engineering thesis is based on the design and construction of an unmanned aerial vehicle in the form of a quadcopter, using a microcontroller from the ESP32 family. The theoretical part concerns the principle of operation of multicopters, and also presents the basic components from which they are built. The principles that guided the selection of individual components were discussed there. Next, the computer software that contributed to the project is described. The part devoted to the control system describes the aspects of designing and then implementing the electrical system, as well as the process

of creating software controlling the system. The mechanical layout chapter focuses on designing and building the frame, putting all the components together, and testing the finished model. Finally, a partially working prototype of the unmanned aerial vehicle was created. The analysis of the results is discussed in the summary, as well as in the quadrocopter test section.