



Łukasz Przeniosło

nr albumu: 21789
kierunek studiów: Elektrotechnika
specjalność: urządzenia i instalacje elektryczne
forma studiów: studia niestacjonarne

UNIWERSALNY INTELIGENTNY STEROWNIK SILNIKÓW DO ZASTOSOWAŃ PRZEMYSŁOWYCH

**UNIVERSAL INTELLIGENT MOTOR DRIVER
FOR INDUSTRIAL APPLICATIONS**

Praca dyplomowa magisterska
napisana pod kierunkiem:

dr. inż. Michała Bonisławskiego

Katedra Elektroenergetyki i Napędów Elektrycznych

Data wydania tematu pracy: 01.08.2015 r.

Data złożenia tematu pracy: 01.07.2016 r.

Szczecin 2016

OŚWIADCZENIE AUTORA PRACY DYPLOMOWEJ

Oświadczam, że praca dyplomowa inżynierska/magisterska pt.
„**UNIWERSALNY INTELIGENTNY STEROWNIK SILNIKÓW
DO ZASTOSOWAŃ PRZEMYSŁOWYCH**”
napisana pod kierunkiem: dr. inż. Michała Bonisławskiego
jest w całości moim samodzielnym autorskim opracowaniem sporządzonym przy
wykorzystaniu wykazanej w pracy literatury przedmiotu i materiałów źródłowych.
Złożona w dziekanacie Wydziału Elektrycznego treść mojej pracy dyplomowej w for-
mie elektronicznej jest zgodna z treścią w formie pisemnej.

Oświadczam ponadto, że złożona w dziekanacie praca dyplomowa ani jej frag-
menty nie były wcześniej przedmiotem procedur procesu dyplomowania związa-
nych z uzyskaniem tytułu zawodowego w uczelniach wyższych.

.....
podpis dyplomanta

Szczecin, dn. 01.07.2016 r.

Streszczenie pracy

Praca przedstawia projekt systemu sterowania do różnych silników elektrycznych, na który składają się między innymi: zbudowane w oparciu o 32 bitowy mikrokontroler urządzenie do którego podłączany jest silnik i wykorzystującego periferię oraz aplikacja (interfejs użytkownika) służąca do nastaw niezbędnych do poprawnego działania układu parametrów oraz badania pracy zasilanego przez sterownik silnika. Rozwiązanie ma zostać zastosowane do sterowania elementami ruchomymi w maszynach typu *Pick and Place*, drukarkach szablonowych, drukarkach 3D oraz we wszelkiego rodzaju konwojach (mechanizmy służące do transportowania obiektów). W części opisowej dokumentu, została zaprezentowana niezbędną teorią na temat silników elektrycznych oraz sterowania. Ponadto, zostały porównane komercyjne rozwiązania sterowników różnej klasy, które są dostępne na rynku.

Słowa kluczowe

Sterowanie, silnik, elektronika mocy, oprogramowanie

Abstract

The paper presents a project of a control system, that can be used with a various types of electric motors. Among others the main components of the system are: built upon a 32 bit microcontroller device, to which the motors and peripherals are connected (the driver) and a PC application used for defining critical operating parameters and for gathering connected motor work diagnostics . This application is to be used with Pick and Place machines, stencil printers, 3D printers and all sort of conveyors (machines used for transportation of objects). In the generic part of the text, the essential theory of electric motors and control has been described. Also, commercially available on the market motor drivers of various class has been compared.

Key words

Control, motor, power electronics, software

Spis treści

| | |
|--|-----------|
| Wprowadzenie | 7 |
| Cel pracy | 7 |
| Zakres pracy | 7 |
| 1. Podstawowe informacje o silnikach | 9 |
| 1.1. Podział silników elektrycznych | 9 |
| 1.2. Parametry silników | 11 |
| 1.3. Silnik liniowy (VCM) | 13 |
| 1.4. Silnik krokowy | 15 |
| 1.4.1. Rodzaje silników krokowych | 15 |
| 1.4.2. Sterowanie | 17 |
| 1.4.3. Zalety i wady silników krokowych | 24 |
| 1.5. Silnik komutatorowy (DC) | 25 |
| 1.6. Silniki synchroniczne z magnesami trwałymi | 26 |
| 2. Wybrane metody sterowania i zasilania silników elektrycznych | 29 |
| 2.1. Tryby pracy | 30 |
| 2.2. Sterowanie mostka „H” | 31 |
| 2.2.1. Modulacja bipolarna | 31 |
| 2.2.2. Sterowanie unipolarne | 33 |
| 3. Przegląd wybranych rozwiązań sterowników silników | 35 |
| 3.1. Sterowniki jedno-układowe | 35 |
| 3.1.1. AD3950 - Allegro MicroSystems | 35 |
| 3.1.2. DRV8825 - Texas Instruments | 36 |
| 3.1.3. TMCC160 - Trinamic | 37 |
| 3.2. Sterowniki pełnowymiarowe | 38 |
| 3.2.1. Simplexmotion 100A - Simplex Motion AB | 38 |
| 3.2.2. Mechaduino - Tropical Labs | 39 |
| 3.2.3. Sterowniki firmy Technosoft | 40 |
| 4. Koncepcja projektowanego sterownika | 43 |
| 4.1. Uniwersalność | 44 |
| 4.2. Zgodność protokołu komunikacyjnego | 46 |
| 4.3. Wirtualizacja osi | 47 |
| 5. Projekt uniwersalnego sterownika | 48 |
| 5.1. Płytką zasilającą | 49 |
| 5.2. Płytką enkodera magnetycznego | 50 |
| 5.3. Płytką enkodera optycznego | 53 |
| 5.4. Płytką główną sterownika | 56 |
| 5.4.1. Zasilanie | 59 |
| 5.4.2. Jednostka centralna | 62 |
| 5.4.3. Komunikacja | 64 |
| 5.4.4. Czujnik podciśnienia | 66 |
| 5.4.5. Obwód sterowania silnikami | 67 |
| 6. Oprogramowanie | 78 |
| 6.1. Komunikacja | 80 |
| 6.1.1. Magistrala CAN | 80 |

| | |
|---|------------|
| 6.1.2. Magistrala RS232 | 81 |
| 6.2. Realizacja zadań | 82 |
| 6.3. Algorytmy enkodera magnetycznego | 83 |
| 6.3.1. Uśrednianie | 84 |
| 6.3.2. Filtr Kalmana | 84 |
| 6.3.3. Filtr dolnoprzepustowy | 87 |
| 6.4. Procedury sterowania wysokopoziomowego | 88 |
| 6.4.1. Regulatory pozycji i prędkości | 91 |
| 6.4.2. Generator trajektorii | 92 |
| 6.4.3. Sterowanie prędkością kątową silnika krokowego | 94 |
| 6.4.4. Regulator prądu silnika VCM | 96 |
| 6.4.5. Regulator prądu silnika krokowego | 99 |
| 6.4.6. Sprawdzenie systemu sterowania | 100 |
| 6.5. Procedury sterowania niskopoziomowego | 101 |
| 6.5.1. Modulacja szerokości impulsów | 101 |
| 6.5.2. Pomiar prądu w uzwojeniu | 103 |
| 6.6. Aplikacja testująca | 105 |
| Podsumowanie | 107 |
| A. Pełny schemat ideowy sterownika | 108 |
| Literatura | 124 |
| Spis tabel | 126 |
| Spis rysunków | 127 |
| Spis kodów źródłowych | 131 |

Wprowadzenie

Chcąc zaprojektować system napędowy dla określonej aplikacji należy wziąć pod uwagę bardzo wiele czynników. Najważniejsze z nich to koszt i dostępność rozwiązań gotowych na rynku. Ceny rozwiązań przemysłowych bardzo rzadko mają odzwierciedlenie w zastosowanym w ich strukturze *hardwarze*, jest to raczej cena za *know-how* producenta, wiedzę ekspercką, wsparcie techniczne oraz dziesiątki roboczogodzin programistów i elektroników tworzących produkt. W przypadku posiadania dużych środków takie rozwiązanie wydaje się być najbardziej odpowiednie. Ma ono jednak jedną znaczącą wadę (pomijając cenę). Pomimo tego, że producenci sterowników do silników prześcigają się w ilości posiadanych cech, czy technologii, to korzystając z takiego urządzenia jesteśmy zmuszeni do korzystania z zamkniętego, ograniczonego systemu. System taki może lepiej lub gorzej wpasowywać się w ramy naszej aplikacji. Rzadko kiedy wpasowuje się idealnie, a najczęściej brakuje mu jakichś właściwości lub jest przewymiarowany.

Drugą opcją jest zakup taniego, zintegrowanego układu scalonego, który w zależności od wyłożonych środków (w o wiele mniejszej skali) zawiera w sobie mniej lub więcej logiki i czasami końcówki mocy. Taki układ można także zbudować z elementów dyskretnych, co nie obniża znacząco kosztów, lecz daje możliwość większej kontroli nad parametrami układu. Wady takiego rozwiązania są takie, że zintegrowane sterowniki są projektowane pod zastosowanie z konkretnym rodzajem silnika (np tylko do silników krokowych) i służą do sterowania silnikami małej mocy (jeśli mają zintegrowane końcówki). W przypadku stosowania układów tego typu oraz dyskretnych, projektant jest zmuszony zaprojektować obwód drukowany oraz stworzyć oprogramowanie zawierające wiele skomplikowanych procedur, które muszą działać w czasie rzeczywistym.

W przypadku, kiedy ilość produkowanych maszyn współpracujących ze sterownikami do silników jest duża, posiadając odpowiednią wiedzę i umiejętności, bardziej opłaca się zaprojektować rozwiązanie własne „od zera”. Ponadto, proces tworzenia tego typu systemu dostarcza ogromną ilość wiedzy praktycznej.

Cel pracy

Zbudować sterownik do sterowania różnego typu silników (krokowych, *BLDC*, liniowych i *DC*), z możliwością dostosowywania parametrów, używając aplikacji PC. Aplikacja ma pozwalać na strojenie działających w sterowniku regulatorów, wybrać tryby pracy i badać działanie systemu w czasie rzeczywistym. Urządzenie ma zostać użyte w maszynach przemysłowych służących do produkcji urządzeń elektronicznych.

Zakres pracy

Przegląd i analiza rozwiązań technologicznych, układów elektronicznych i oprogramowania stosowanych w sterownikach silników oferowanych przez wybranych producentów. Projekt uniwersalnego, kompaktowego sterownika do silników *DC* (prądu stałego), *BLDC* (*Brushless DC Motor*), krokowych oraz liniowych *VCM* (*Voice Coil Motor*, inaczej *VCA - Voice Coil Actuator*) wraz z opisem sposobów regulacji

i algorytmów zawierających elementy *AI*, które zostaną zaimplementowane w nowoczesnym 32 bitowym mikrokontrolerze z rodziny *ARM*. Opracowanie aplikacji na PC umożliwiającej dostrajanie parametrów dla konkretnego silnika oraz analizowanie jego pracy. Wykonanie, uruchomienie i badania laboratoryjne prototypu urządzenia. Prezentacja przykładowego zastosowania sterownika działającego w rzeczywistym układzie.

1. Podstawowe informacje o silnikach

Rozdział ten przedstawia podział maszyn elektrycznych ze względu na ich różne parametry, oraz opis tych silników, które są (i mogą być) wspierane przez skonstruowany sterownik. Rozdział został opracowany głównie na podstawie źródeł [34] i [26].

1.1. Podział silników elektrycznych

Istnieje bardzo wiele rodzajów silników, a ponadto cały czas powstają nowe konstrukcje, które trudno jest sklasyfikować przy pomocy klasycznych definicji. Podział na silniki AC i DC nie wystarczy – w przypadku zastosowania elektronicznego modułu sterującego rodzaj zasilania silnika nie ma tak dużego znaczenia. Nie można także jednoznacznie stwierdzić, czy maszyna z wirującymi magnesami to silnik prądu stałego, czy przemennego. Biorąc to pod uwagę, silniki elektryczne można podzielić na trzy podstawowe grupy.

Silniki prądu stałego – DC

Zaliczane są tu silniki komutatorowe:

- z magnesem trwałym (PMDC),
- szeregowe,
- bocznikowe,
- szeregowo-bocznikowe.

Silniki prądu przemennego – AC

Do tej grupy zaliczane są silniki zasilane bezpośrednio z sieci prądu przemennego. W tym przypadku podział jest bardziej skomplikowany, ze względu na inne kryteria:

- Silniki asynchroniczne – charakteryzuje je tzw. „poślizg”. Rotor takiego silnika wiruje z niższą prędkością od prędkości wirowania pola magnetycznego. Najpopularniejszymi silnikami z tej grupy są silniki klatkowe.
- Silniki synchroniczne – rotor wiruje z prędkością równą prędkości wirowania pola magnetycznego. Mogą być jednofazowe lub trójfazowe. Ze względu na konstrukcję, silniki te dzielą się na reluktancyjne lub z wirującym magnesem.
- Silniki komutatorowe uniwersalne – komutatorowe silniki szeregowe przystosowane konstrukcyjnie do zasilania z jednej fazy prądu przemennego.

Silniki z komutacją elektroniczną

Do tej grupy zalicza się silniki, których działanie nie było by możliwe bez elektronicznego układu sterującego. Najpopularniejsze z nich to:

- Silniki krokowe – ze względu na konstrukcję wyróżnia się silniki:
 - o zmiennej reluktancji,
 - z magnesem trwałym,
 - hybrydowe.

Ze względu na sposób sterowania uzwojeniami:

- silniki unipolarne,
- silniki bipolarne.

Daje to sześć różnych kombinacji konstrukcyjnych silników krokowych.



Rysunek 1.1: Przykładowy silnik krokowy hybrydowy ST4118 w standardzie Nema17, produkowany przez firmę Nanotec

Źródło: Na podstawie [11]

Nema (*National Electrical Manufacturers Association*) to stowarzyszenie producentów urządzeń elektrycznych, którego siedziba mieści się w USA. Standard Nema określa między innymi wymiary silników krokowych. Numer 17 oznacza szerokość i wysokość silnika 1.7 x 1.7 cala (43.2 x 43.2 mm).

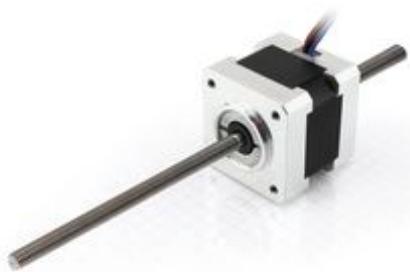


Rysunek 1.2: Przykładowy silnik BLDC (model DB42), produkowany przez firmę Nanotec

Źródło: Na podstawie [11]

- Silniki bezszczotkowe z wirującym magnesem (BLDC) – w literaturze angielskiej silniki te są opisywane jako prądu stałego (*BrushLess DC*). W przypadku silników dwubiegunkowych z czujnikami Halla (np. tego typu, które są używane w małych wentylatorach do chłodzenia radiatorów) taki opis może być uzasadniony. Nie można jednak jednoznacznie zaliczyć do grupy DC silników wielobiegunkowych trójfazowych z wirującym magnesem [34]. Mylne może być także to, że w przypadku silników z wbudowanym sterownikiem nie różnią się one z punktu użytkownika od silnika komutatorowego z magnesem trwałym. Maszyny BLDC charakteryzują się bardzo wysoką prędkością obrotową, w szczególności w porównaniu do silników krokowych.

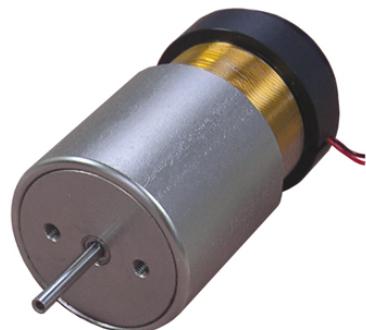
- Silniki liniowe – wynikiem działania takiej maszyny jest przemieszczenie liniowe, a nie ruch obrotowy. Konstrukcja takiego silnika w dużym uproszczeniu polega na „rozwinięciu” wirnika i stojana na płaszczyźnie. Nazwa silnik liniowy lub *Linear Actuator* bardzo często dotyczy zwykłego silnika krokowego z wbudowaną przekładnią śrubową zamieniającą ruch obrotowy na posuwisto-zwrotny, co bardzo często wprowadza niejasności i wątpliwości podczas projektowania systemów napędowych.



Rysunek 1.3: Przykładowy silnik krokowy hybrydowy L35 z przekładnią śrubową, produkowany przez firmę Nanotec

Źródło: Na podstawie [11]

- Silniki liniowe VCM – zasada działania silnika VCM (*Voice Coil Motor*) lub VCA (*Voice Coil Actuator*) jest bardzo zbliżona do zasady działania cewki głośnikowej w szczelinie magnesu. Przesunięcie liniowe tych silników nie przekracza najczęściej 50 mm.



Rysunek 1.4: Przykładowy silnik liniowy VCM (model GVCM-051-051-01), produkowany przez firmę Moticont

Źródło: Na podstawie [10]

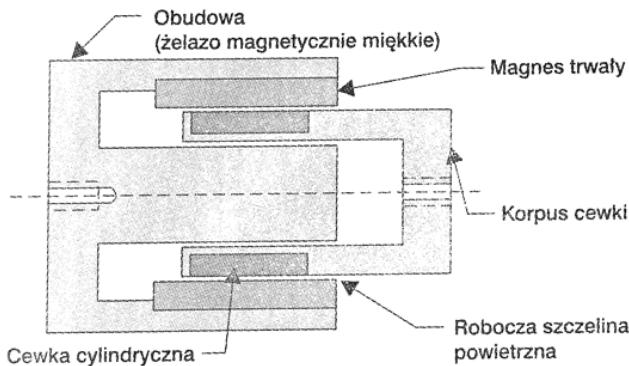
1.2. Parametry silników

Podstawowe parametry silnika są czasami podawane na jego tablicy znamionowej. Pozostałe należy odszukać w jego karcie katalogowej lub oszacować według właściwości poszczególnych rodzajów silników.

- Moc znamionowa: podawana jest moc elektryczna w watach pobierana przez silnik przy pracy z normalną prędkością obrotową. Moc mechaniczna jest mniejsza i zależy od sprawności silnika (typowo od 40 do 80 %).
- Napięcie zasilania: znamionowa wartość napięcia zasilającego (stałego lub zmiennego), przy której określone są pozostałe parametry. Najczęściej silniki mogą być zasilane napięciem niższym. Nie należy stosować napięć dużo wyższych od znamionowego (maks. +15 %).
- Moment obrotowy: parametr ten jest podawany w niutonometrach (Nm). Typowe wartości dla małych silników: od 0.01 do 50 Nm. Dla większości rodzajów silników występuje zależność momentu obrotowego i natężenia prądu pobieranego przez maszynę.
- Moment rozruchowy: ważny parametr informujący o tym czy silnik jest w stanie wystartować pod obciążeniem. Podawany w Nm lub w procentach momentu obrotowego. W zależności od rodzaju silnika moment może być mały (do 150 %), średni (150 - 250 %) lub duży (> 250 %).
- Obroty znamionowe: wyrażane w obrotach na minutę (*RPM – Revolutions per minute*). Zawsze podawane są przy znamionowym obciążeniu i napięciu zasilania lub podana jest charakterystyka obrotów w funkcji obciążenia. Obroty silników zawierają się w granicach od 100 do 100000 RPM, ale najczęściej spotykane wartości to od 1000 do 4000 RPM. Dla maszyn prądu przemysłowego podawane są obroty przy określonej częstotliwości napięcia sieciowego (50 lub 60 Hz). Dla silników krokowych obroty znamionowe nie są podawane. Zamiast tego występują obroty maksymalne lub dopuszczalna częstotliwość impulsów.
- Prąd znamionowy i rozruchowy: pobór prądu w normalnych warunkach pracy i w czasie rozpoczęcia pracy silnika. Ten drugi może być nawet 2 - 8 krotnie większy od znamionowego (największe prądy rozruchowe mają silniki indukcyjne). Prąd rozruchowy nie występuje w silnikach sterowanych impulsowo.
- Kierunek obrotów: parametr dotyczący przede wszystkim silników komutatorowych szeregowych i niektórych silników indukcyjnych jednofazowych. Dla tych silników kierunek obrotów jest wymuszony przez konstrukcję silnika. W pozostałych maszynach kierunek zależy od polaryzacji przyłożonego do ich zacisków napięcia lub w przypadku maszyn ze sterowaniem impulsowym – od kolejności podawanie impulsów na poszczególne uzwojenia.
- Konstrukcja mechaniczna: wymiary, ciężar, średnica wału, mocowania, rodzaj łożysk wirnika itp.
- Konstrukcja obudowy, chłodzenie: spotykane są silniki o konstrukcji otwartej, w której uzwojenia są dostępne z zewnątrz lub zamkniętej w której silnik jest całkowicie obudowany. W silnikach otwartych często wymagane jest chłodzenie strumieniem powietrza – np. silniki odkurzaczy, czy wiertarek.

1.3. Silnik liniowy (VCM)

Pierwszym z silników obsługiwanych przez projektowany sterownik jest silnik liniowy VCM. Jego konstrukcja jest bardzo prosta, aczkolwiek sam proces technologiczny wymagany do jego wytworzenia już nie. Z tego właśnie powodu VCM nawet bardzo małej mocy są drogie (Cena od \$100 za pojedyncze sztuki). Ciężko jest tak naprawdę stwierdzić czy jest to silnik czy siłownik, bo zakres jego ruchu nie przekracza zwykle kilku centymetrów.



Rysunek 1.5: Budowa cylindrycznego silnika VCM

Źródło: Na podstawie [34]

Nazwa silnika pochodzi od wykonywanego ruchu (cewka poruszająca się w szczerbinie magnesu). Zgodnie z regułą Lorentza:

Definicja 1.1. Jeżeli przez cewkę znajdującą się w polu magnetycznym przepływa prąd, to na cewkę działa siła proporcjonalna do natężenia prądu, a napięcie samo-indukcji jest proporcjonalne do szybkości poruszania się cewki [34].

Podstawowe części składowe silnika VCM to:

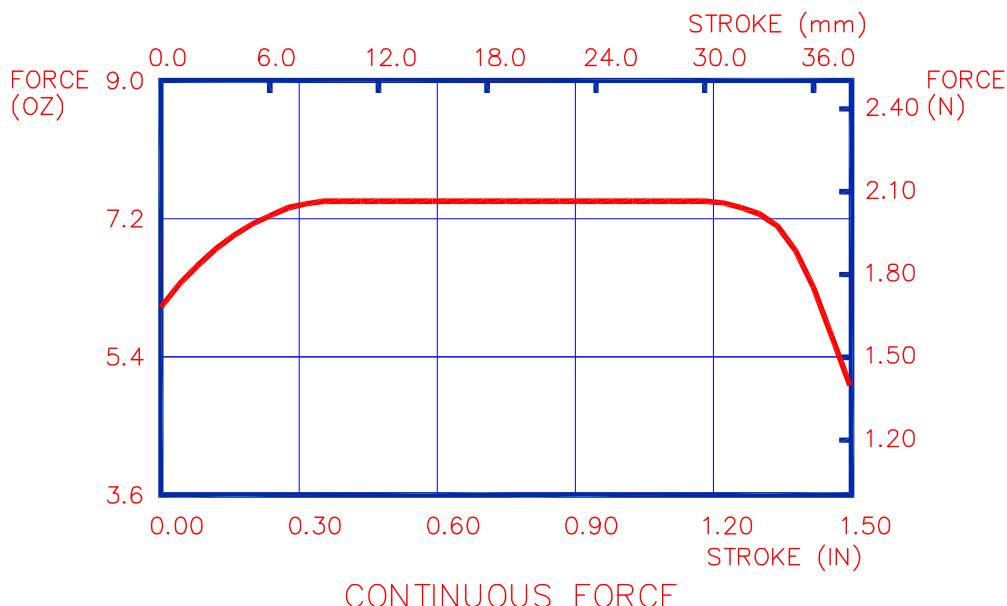
- dwa magnesy trwałe ułożone tak, aby były skierowane tym samym biegunem w stronę cewki,
- cewka poruszająca się w szczerbinie między magnesami,
- obudowa z miękkiego żelaza zamykająca obwód magnetyczny.

Zmieniając natężenie i kierunek przepływającego prądu, możliwe jest bardzo precyzyjne sterowanie położeniem cewki. Z tego powodu układ elektroniczny sterujący silnikiem nie wymaga wielu komponentów, a algorytm sterowania nie jest skomplikowany. VCM ma jednak jeszcze jedną zaletę, która sprawia że jest on wybrany do niektórych aplikacji zamiast np. tańszego silnika krokowego z przekładnią śrubową – pozwala wykonywać bardzo dynamiczne ruchy. „Liniowy” silnik krokowy o podobnej mocy, nawet z zastosowaniem śruby o dużym skoku nie będzie w stanie osiągnąć podobnych prędkości przy zachowaniu korzystnego momentu. Brak konieczności zamiany ruchu obrotowego na posuwisto-zwrotny eliminuje błędy pozycji (które mogą wynikać np. z luzów w przekładni).

Silniki VCM stosowane są wszędzie tam, gdzie potrzebne jest dokładne i szybkie ustalenie pozycji, np. w precyzyjnych urządzeniach mechanicznych i optycznych.

Aby efektywnie i bezpiecznie korzystać z silników VCM w danej aplikacji należy mieć na uwadze kilka ważnych cech tych urządzeń:

- Korpus cewki nie jest w żaden sposób przymocowany do obudowy silnika. Oznacza to że to projektant musi zadbać o to, aby korpus wsuwał i wysuwał się z cylindra równolegle, stosując odpowiednie mocowanie mechaniczne obu części. Nie zapewniwszy tego użwojenie silnika jest narażone na starcie się lakieru izolacyjnego (poprzez ocieranie się o obudowę).
- W zależności od aktualnego położenia (wysunięcia) korpusu, zależność między siłą a przepływanym przez użwojenie prądem jest zmienna.



CONTINUOUS FORCE

Rysunek 1.6: Przykładowy przebieg siły $\left[\frac{N}{A}\right]$ pchającej/ ciągnącej w funkcji wysunięcia korpusu dla silnika liniowego VCM 019-048-02, firmy Moticont.

Źródło: Na podstawie [10]

Graf 1.6 prezentuje zależność między siłą silnika a położeniem jego cewki. Jak widać największa i stała siła ($2\frac{N}{A}$) występuje dopiero kiedy korpus jest trochę wysunięty i zaczyna spadać przy końcowym wysuwie. Jeżeli układ mechaniczny, którego częścią jest silnik, nie zostanie wyposażony w odpowiednią blokadę, korpus może wypaść z cylindra ponieważ silnik nie będzie już w stanie go utrzymać.

- Producent w karcie katalogowej silnika podaje parametr maksymalnej ciągłej mocy (*Max continuous power*) i odpowiada ona zazwyczaj prądowi, który płynie przez cewkę w momencie niepełnego wysunięcia (należy oczywiście wziąć pod uwagę fizyczne obciążenie części ruchomej – masę obudowy lub korpusu, w zależności od tego która z części jest przymocowana, a która jest w ruchu). Przy dużym wysunięciu zależność siły do płynącego przez cewkę prądu drastycznie spada i utrzymując korpus długo w takiej pozycji można w bardzo łatwy sposób spalić użwojenie silnika.

Przy zastosowaniu odpowiedniej konstrukcji układu napędowego (zachowana równoległość części mechanicznych) i sterowania zapewniającego ochronę przed zbyt dużym prądem płynącym przez cewkę, silnik VCM z czasem nie ulega właści-

wie żadnej degradacji i długość jego działania jest ograniczona żywotnością części mechanicznych układu.

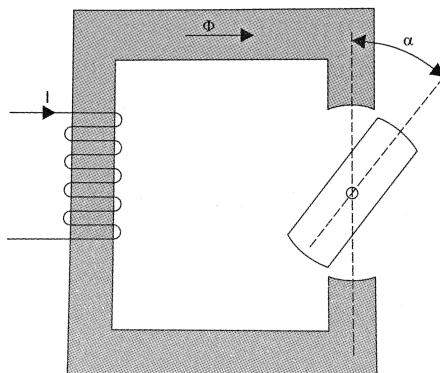
1.4. Silnik krokowy

Silniki krokowe są maszynami elektrycznymi bez-komutatorowymi. Zazwyczaj wszystkie uzwojenia silnika są częścią stojana, podczas gdy rotor jest magnesem trwałym lub w przypadku silników VRM (*Variable Reluctance Motor*) zębatym blokiem z materiału magnetycznie miękkiego. Komutacja (czyli zmiana kierunku/uzwojenia w którym następuje przepływ prądu elektrycznego) musi być wymuszona poprzez zewnętrzne urządzenie sterujące maszyną. Takie sterowniki są zazwyczaj projektowane w taki sposób, aby zapewnić możliwość utrzymania wirnika w niemal dowolnej pozycji kątowej oraz obracać nim w obu kierunkach.

1.4.1. Rodzaje silników krokowych

Silniki krokowe ze względu na konstrukcję można podzielić na trzy główne grupy:

- silniki o zmiennej reluktancji,
- z magnesem trwałym,
- hybrydowe.

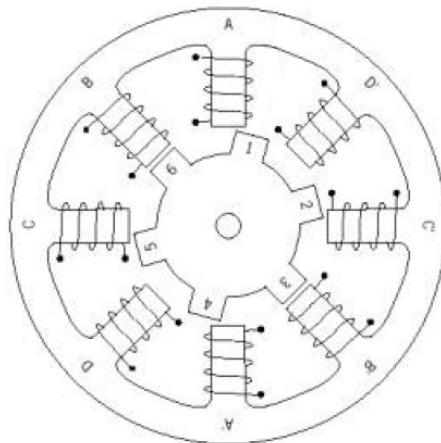


Rysunek 1.7: Powstawanie reluktancyjnego momentu obrotowego.

Źródło: Na podstawie [34]

Obrócenie ruchomego elementu o kąt α spowoduje że będzie on próbował powrócić do położenia $\alpha = 0$, w którym reluktancja obwodu jest najmniejsza. W silnikach VRM przepływ prądu stałego przez uzwojenia powoduje, że zęby wirnika ustawiają się naprzeciw zasilanego uzwojenia.

W odróżnieniu od VRM, wirnik silnika krokowego PM (*Permanent Magnet-Magnes trwały*) nie posiada zębów. Jest wykonany w postaci walca naprzemiennie namagnesowanego biegunami N i S. Specyficzny rodzaj zębów ma rdzeń stojana. W zależności od kierunku przepływu prądu w uzwojeniu przyciągane są odpowiednie bieguny wirnika (rysunek 1.9). Główną zaletą silnika krokowego PM jest niska cena. Wadą tego typu maszyny jest relatywnie niski moment obrotowy i brak możliwości rozwijania dużych prędkości obrotowych.

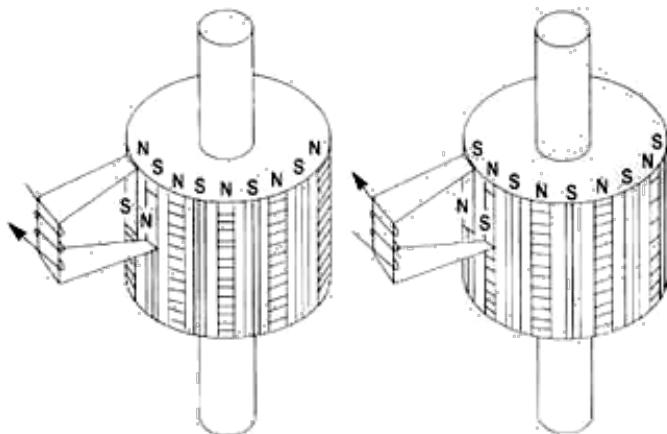


Rysunek 1.8: Silnik o zmiennej reluktancji (VRM)- Przekrój.

Źródło: Na podstawie [34]

Silnik hybrydowy łączy w sobie cechy obu rozwiązań, dzięki czemu zostały poprawione takie parametry jak:

- moment obrotowy,
- maksymalna prędkość obrotowa,
- rozdzielcość kroku.

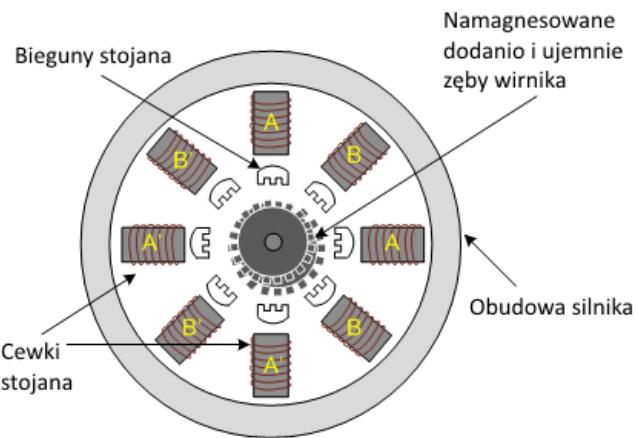


Rysunek 1.9: Zasada działania silnika PM.

Źródło: Na podstawie [12]

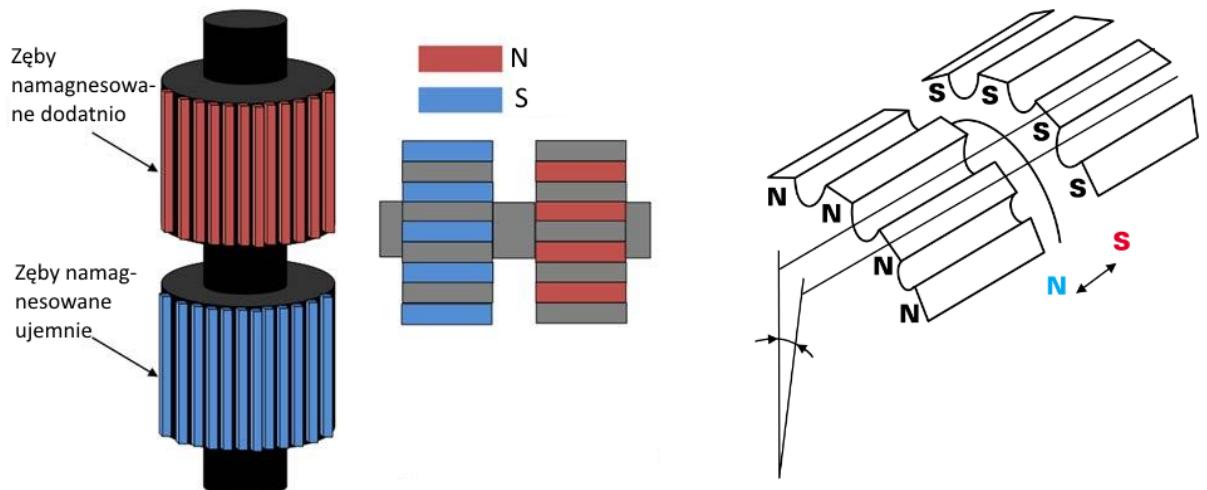
Niestety silniki hybrydowe są około 2-3 razy droższe od silników PM. Wirnik silnika HB (*Hybrid Motor*) jest zbudowany z uzębionych nabiegunków i magnesu trwałego, powodującego naprzemienne magnesowanie zębów biegunami N i S. Uzębiony stojan konstrukcją przypomina ten z silnika VRM.

W silniku HB wirujące pole stojana obraca wirnik z jednego położenia do drugiego na zasadzie jak w silniku VRM. Jest to możliwe dzięki przesunięciu „północnej” i „południowej” części wirnika o pół ząbka. Silniki HB dzięki poprawionym parametrom są obecnie najbardziej popularnymi silnikami krokowymi, pomimo niskiej ceny silników PM.



Rysunek 1.10: Konstrukcja dwufazowego silnika krokowego hybrydowego.

Źródło: Na podstawie [2]



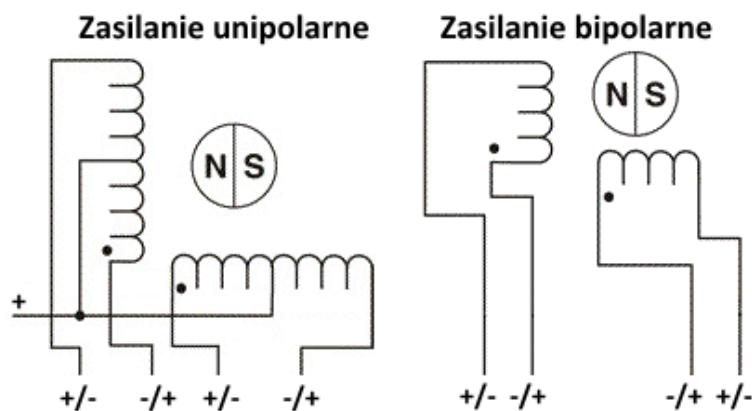
Rysunek 1.11: Budowa wirnika silnika krokowego hybrydowego dwufazowego.

Źródło: Na podstawie [30]

1.4.2. Sterowanie

W pierwszej kolejności należy rozróżnić sposób zasilania silnika krokowego, który może być unipolarny lub bipolarny (rys. 1.12).

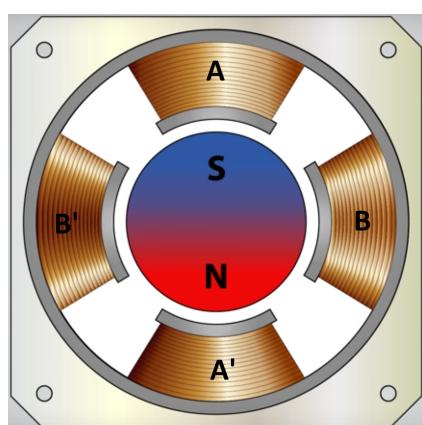
Przy zasilaniu unipolarnym, dla pojedynczego uzwojenia na zewnątrz silnika wyprowadzone są trzy przewody – oba końce cewki i odczep w połowie jej długości. Podłączając do odczepu „+” zasilania można sterować połówkami uzwojeń zwierając w odpowiedniej kolejności i na odpowiedni czas końcówki do masy. W zasilaniu bipolarnym, prąd płynie zawsze przez całe uzwojenie (z silnika wyprowadzone są tylko dwa przewody na cewkę). Zaletą wariantu unipolarnego jest możliwość znacznego uproszczenia układu sterowania. Poważną wadą natomiast jest to, że prąd podawany jest zawsze tylko na połowę danego uzwojenia, co ma negatywne przełożenie na moment obrotowy.



Rysunek 1.12: Sposób sterowania (zasilania) silnika krokowego dwufazowego- z lewej unipolarne, z prawej bipolarne.

Źródło: Na podstawie [4]

Oddzielną sprawą jest sposób/ sekwencja podawania impulsów prądowych do uzwojeń silnika. Sterowanie pod tym względem dzieli się na:



Rysunek 1.13: Uproszczony model silnika krokkowego dwufazowego o kroku 90°.

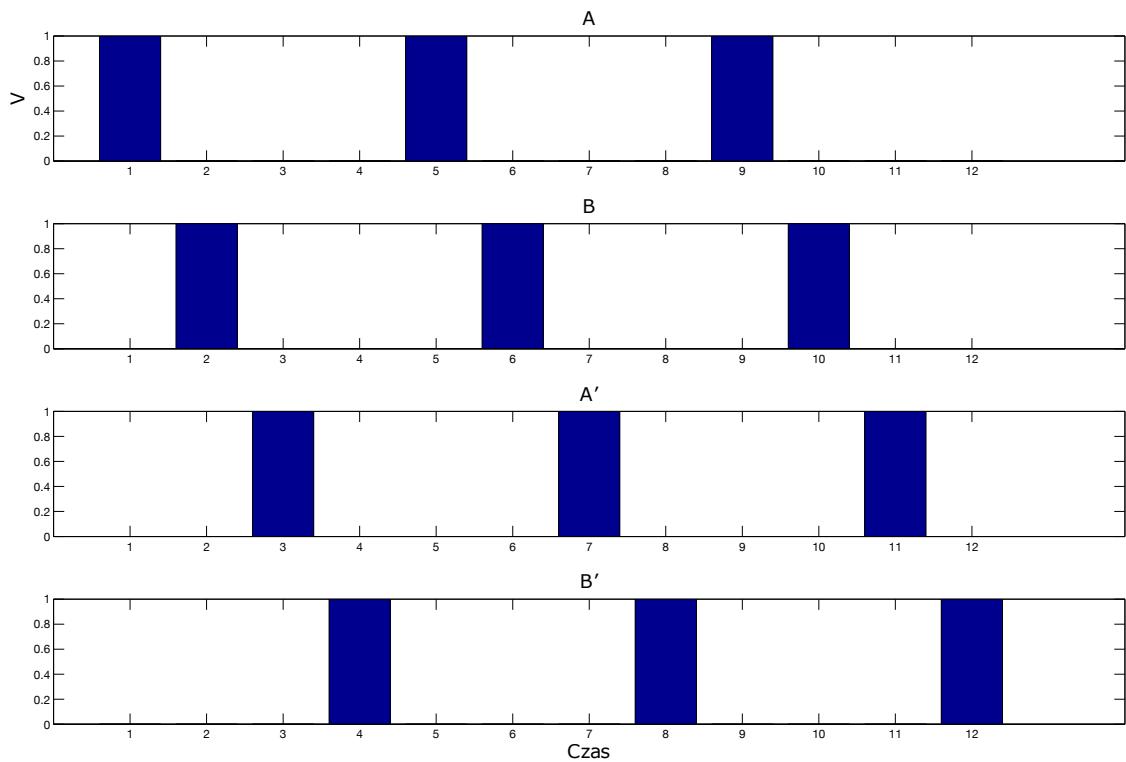
Źródło: Na podstawie [5]

- falowe,
- pełnokrokowe,
- półkrokowe,
- mikrokrokowe.

Do pomocy w wyjaśnieniu poszczególnych typów sterowania posłuży rysunek 1.13, prezentujący uproszczony silnik krokowy. Maszyna składa się z wirnika zbudowanego jak na rys. 1.11 tyle że z ilością zębów (na wirniku i stożek) pozwalającą osiągnąć pełen obrót w czterech pełnych komutacjach. Silnik posiada 2 uzwojenia A i B nawinięte w taki sposób, że pary A/A' i B/B' są ze sobą połączone (uzwojenie podzielone na dwie sekcje). Przykładając napięcie do cewki A, w zależności od polaryzacji tego napięcia, prąd będzie wpływał przez uzwojenie A i wypływał z uzwojenia A' lub na odwrót. Tak samo ma się sytuacja w przypadku cewki B.

Sterowanie falowe

W przypadku braku zasilania, wirnik silnika ustawi się w jednej z czterech spoczynkowych pozycji (0° , 90° , 180° lub 270°). Dzieje się tak dlatego, że namagnesowany wirnik usiłuje zająć taką pozycję, aby reluctancja obwodu magnetycznego była jak najmniejsza, tzn. aby namagnesowane zęby wirnika były ustawione w jednej linii z dowolną parą zębów stojana. Znomy moment obrotowy powodujący takie pozycjonowanie to inaczej moment bezprądowy (spoczynkowy).



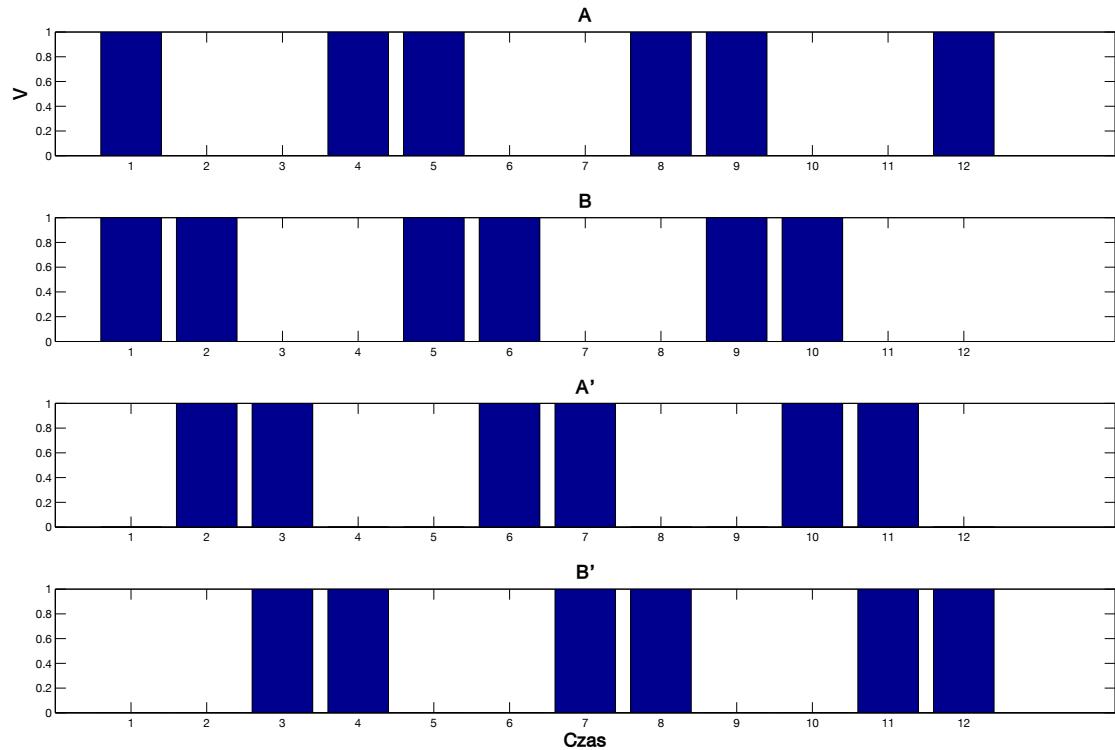
Rysunek 1.14: Sekwencja zasilania uzwojeń przy sterowaniu falowym (obrót wirnika z rys. 1.13 według wskazówek zegara).

Źródło: Na podstawie [33]

Rysunek 1.14 prezentuje sekwencję dzięki której wirnik obróci się o trzy pełne obroty w prawo przy zastosowaniu sterowania falowego. Na osi odciętych znajdują się kolejne chwile w czasie, a na osi rzędnych wartość potencjału przyłożonego do danego zacisku (wartości 1 i 0 prezentujące stan wysoki lub niski potencjału). Aby wirnik obracał się w przeciwnym kierunku, sekwencję należy odtworzyć w odwrotnej kolejności. Jak widać na przebiegach, w danym momencie w czasie prąd znajduje się zawsze tylko w jednym z uzwojeń, co nie umożliwia uzyskania maksymalnego dla danej maszyny momentu obrotowego. Pomimo że sterowanie falowe nie zapewnia uzyskania optymalnego momentu obrotowego, to przy zastosowaniu w niektórych aplikacjach jest wystarczające, a ponadto jest najprostsze do zaimplementowania.

Sterowanie pełnokrokowe

W trybie sterowania pełnokrokowego prąd płynie przez oba uzuojenia jednocześnie. Nie ma to wpływu na prędkość obrotową wirnika, ale ma znaczący wpływ na moment obrotowy, który jest dwa razy większy niż w trybie falowym. Pełny obrót wirnika dalej jest osiągany w czterech "ruchach", ale absolutna pozycja kątowa w każdym z nich jest względem trybu falowego przesunięta o 45° .



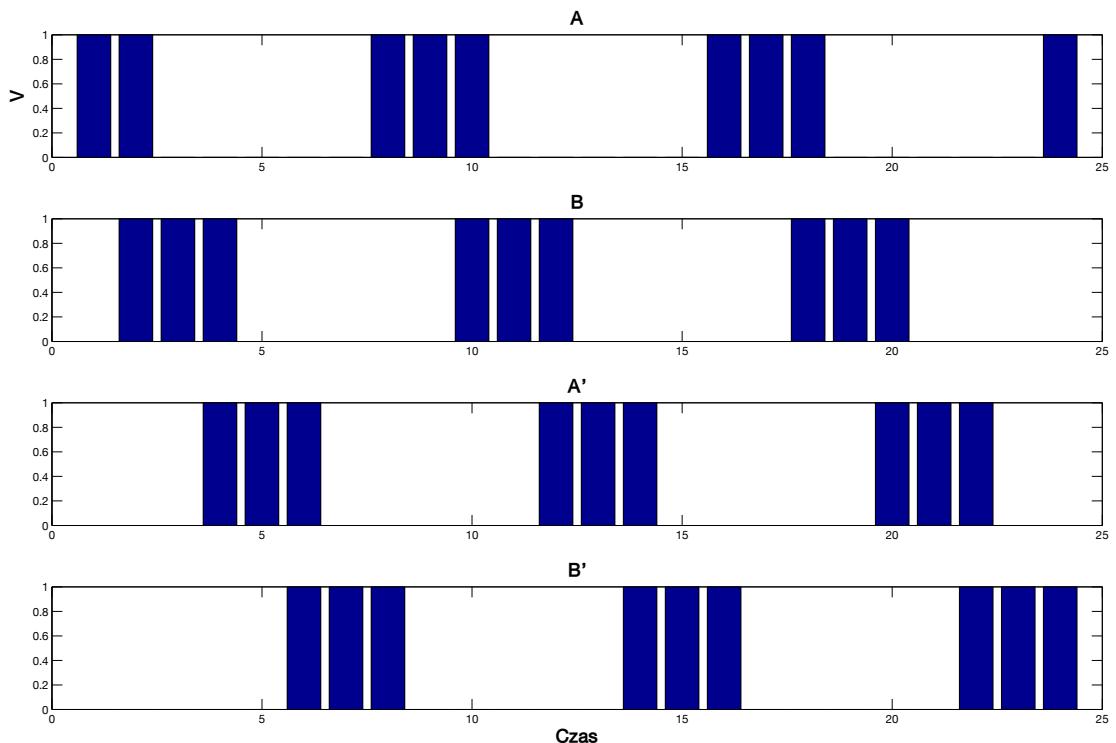
Rysunek 1.15: Sekwencja zasilania uzuojen przy sterowaniu pełnokrokowym (obrót wirnika z rys. 1.13 według wskazówek zegara).

Źródło: Na podstawie [33]

Rysunek 1.15 przedstawia przebiegi dla 3 obrotów, przy sterowaniu pełnokrokowym. Kolejne pozycje wirnika są następujące: 45° w pierwszym interwale czasowym, 135° w drugim, 225° w kolejnym i 315° w ostatnim (dla danego obrotu). Potem cykl się powtarza. Tryb pełnokrokowy pozwala osiągnąć optymalny dla danego silnika moment obrotowy, ale niestety rozdzielczość obrotu jest ograniczona do ilości kroków na obrót danego modelu maszyny. Typowe wartości kroków na obrót w silnikach krokowych hybrydowych to 100 (krok o rozdzielczości 3.6°), 200 (1.8°) i 400 (0.9°). Aby uzyskać większą rozdzielczość kroku, należy zastosować sterowanie półkrokowe lub mikrokrokowe.

Sterowanie półkrokowe

Tryb półkrokowy jest połączeniem sterowania falowego i pełnokrokowego. Rozdzielcość obrotowa względem poprzednich trybów jest dwa razy większa, dlatego że uzwojenia naprzemiennie zasilane są pojedynczo/oba jednocześnie.

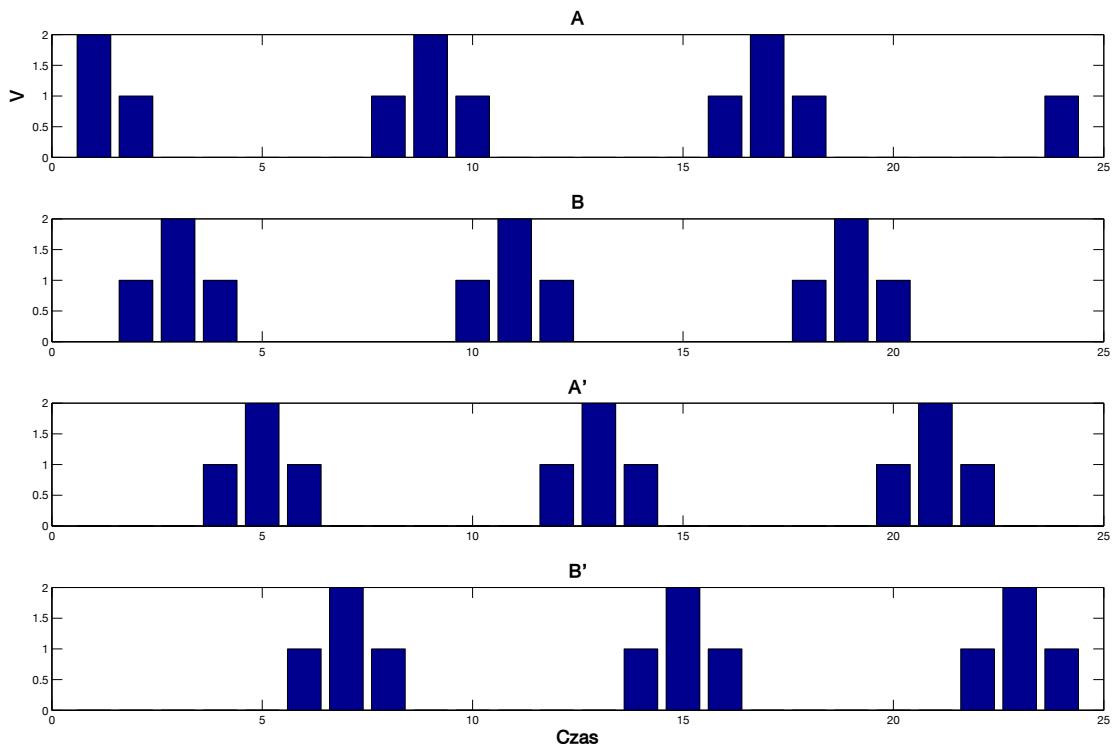


Rysunek 1.16: Sekwencja zasilania uzwojeń przy sterowaniu półkrokowym (obrót wirnika z rys. 1.13 według wskazówek zegara).

Źródło: Na podstawie [33]

Wadą takiego rozwiązania jest to, że w co drugim kroku moment obrotowy będzie mniejszy. Może to spowodować szarpanie wirnika przy dużych prędkościach obrotowych. Przy niskich prędkościach i obciążeniach efekt ten rzadko występuje i można uzyskać polepszoną płynność ruchu, w porównaniu do poprzednich trybów. Aby uniknąć efektu zmiennego momentu, można zastosować sterowanie półkrokowe ze zmiennym prądem (rys. 1.17). Na rysunku widać że w momentach kiedy różnica potencjałów występuje tylko na jednym z uzwojeń, jest ona dwa razy większa niż w przypadku kiedy zasilane są obie cewki. Prąd płynący w uzwojeniach jest proporcjonalny do przyłożonego do nich napięcia, dlatego moment się wyrównuje i silnik nie jest narażony na szarpanie. Rozwiązanie te ma jednak wady– zakładając, że dla momentu w czasie, w którym przez oba uzwojenia płynie prąd wartość tego prądu jest maksymalna (nominalna) dla danego modelu silnika w każdym z uzwojeń, oznacza to, że w chwilach kiedy napięcie jest podawane na tylko jedno uzwojenie, prąd w nim płynący jest większy od nominalnego (tak jak na rys. 1.17). To naraża silnik na nadmierne grzanie, a nawet na uszkodzenie. Z drugiej strony można założyć, że nominalny prąd ma płynąć w momencie zasilania jednego uzwojenia,

a mniejszy przy zasilaniu dwóch. Ten tryb jest bezpieczny, ale optymalny moment obrotowy nie jest uzyskiwany.



Rysunek 1.17: Sekwencja zasilania uzwojeń przy sterowaniu półkrokowym ze zmiennym prądem (obrót wirnika z rys. 1.13 według wskazówek zegara).

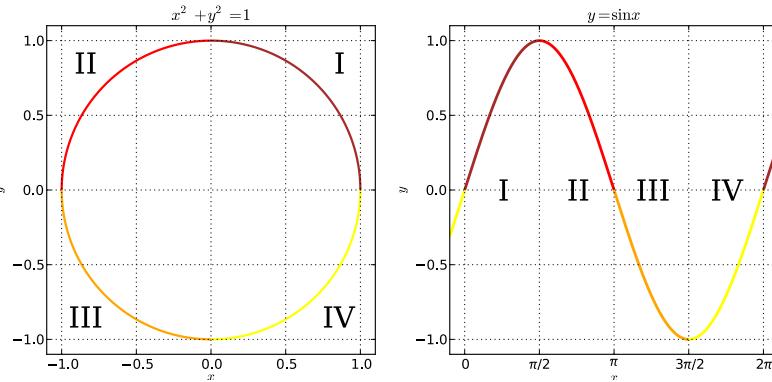
Źródło: Na podstawie [33]

Sterowanie mikrokrokowe

Największą rozdzielcość ruchu (teoretycznie nieograniczoną, a praktycznie ograniczoną możliwościami układu sterującego) można uzyskać przy zastosowaniu trybu mikrokrokowego. Aby obrócić wirnik z rys. 1.13 w najpłynniejszy możliwy sposób, kształt prądu podawany na uzwojenia powinien być naturalny dla danej konstrukcji silnika, czy też dla ruchu, który wykonuje element ruchomy – w tym wypadku jest to ruch obrotowy.

$$\sin^2 \theta + \cos^2 \theta = 1 \quad (1.1)$$

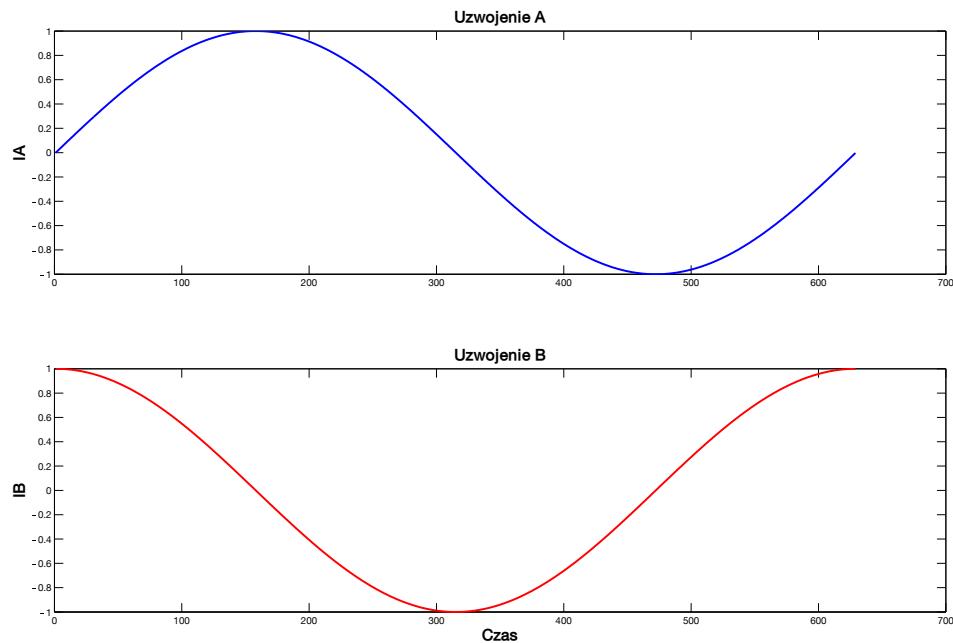
Równanie 1.1 prezentuje wzór na jedynkę trygonometryczną. Można go użyć, aby uzyskać dowolną pozycję wirnika pomiędzy kolejnymi naturalnymi ułożeniami wirnika. W tym momencie należy rozróżnić dwa rodzaje pozycji kątowych wirnika silnika krokowego – magnetyczną oraz fizyczną. W przypadku silnika z rys. 1.13 obie pozycje będą się zawsze pokrywać ze względu na to, że potrzeba jedynie czterech pełnych kroków, aby uzyskać pełen obrót – kąt „magnetyczny” 45° odpowiada obrótowi wirnika o 45° . Dla rzeczywistej sytuacji, w której silnik posiada np. 200 kroków na obrót, jeden stopień magnetyczny odpowiadałby już jedynie $\frac{1}{50}$ stopnia w obrębie. Innymi słowy potrzeba 50 obrotów magnetycznych na 1 obrót fizyczny wirnika.



Rysunek 1.18: Ćwiartki okręgu i funkcji $\sin x$ w kartezjańskim układzie współrzędnych

Źródło: Na podstawie [19]

Tak więc, podając na uzwojenia silnika prąd o kształcie funkcji *sinus* (na pierwsze uzwojenie) i *cosinus* (na drugie), suma ich kwadratów powinna wynosić jeden. Pozwala to na uzyskanie płynnego ruchu wirnika.

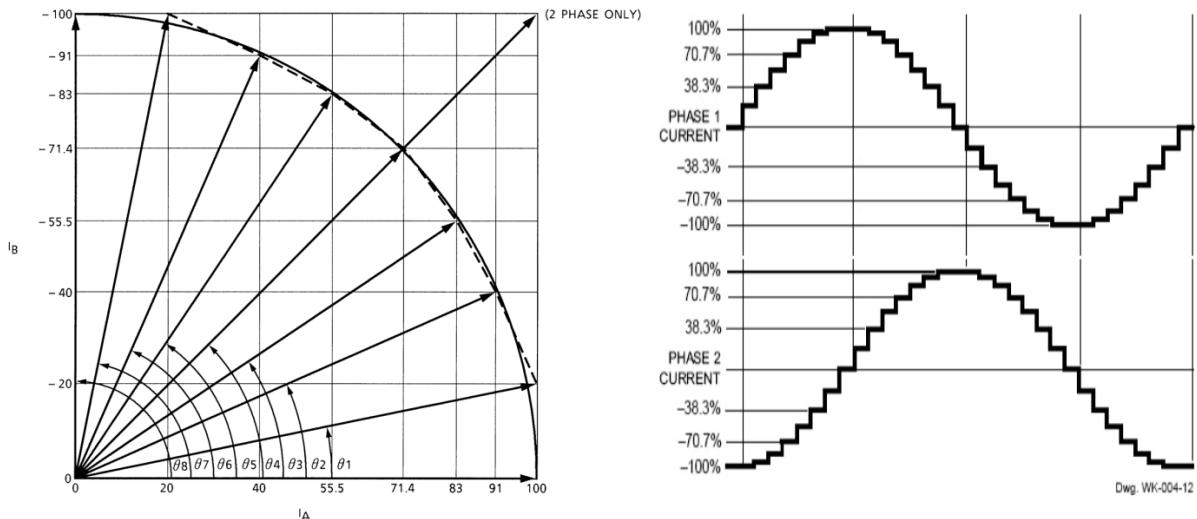


Rysunek 1.19: Kształt prądów podawanych na uzwojenia A i B w celu uzyskania optymalnej płynności obrotu w silniku krokowym hybrydowym dwufazowym.

Źródło: Na podstawie [33]

Prezentowane przebiegi z rys. 1.19 obróćą wał silnika o 360° magnetycznych. Maksymalna amplituda po znalezieniu reprezentuje wartość natężenia nominalnego prądu dla danego modelu silnika. Zazwyczaj silniki krokowe są sterowane z przekształtników, które same są zasilane napięciem stałym. Rozdzielczości używane w typowych sterownikach wynoszą od 8 do 32 mikrokroków na pełen krok.

Tabela 1.1 prezentuje podsumowanie omówionych trybów sterowania silnikami krokowymi dla zasilania bipolarnego i unipolarnego.



Rysunek 1.20: Kształt prądów podawanych na uzwojenia A i B przy sterowaniu mikrokrokom z rozdzielcością $\frac{1}{8}$ kroku.

Źródło: Na podstawie [25]

Tabela 1.1: Cechy poszczególnych trybów sterowania silnikiem krokom

Źródło: Na podstawie [34]

| Rodzaj sterowania | Falowe | | Pełnokrokowe | | Półkrokowe | | Mikrokrokom | |
|------------------------|---------|------|--------------|------|------------|---------|-------------|------|
| | Bi | Uni | Bi | Uni | Bi | Uni | Bi | Uni |
| Moment obrotowy | 50% | 25% | 100% | 50% | 50-100 % | 25-50 % | 50% | 25% |
| Komplikacja sterownika | średnia | mała | średnia | mała | duża | duża | duża | duża |
| Plynność ruchu | mała | mała | mała | mała | duża | duża | duża | duża |
| Rozdzielcość kroku | mała | mała | mała | mała | średnia | średnia | duża | duża |

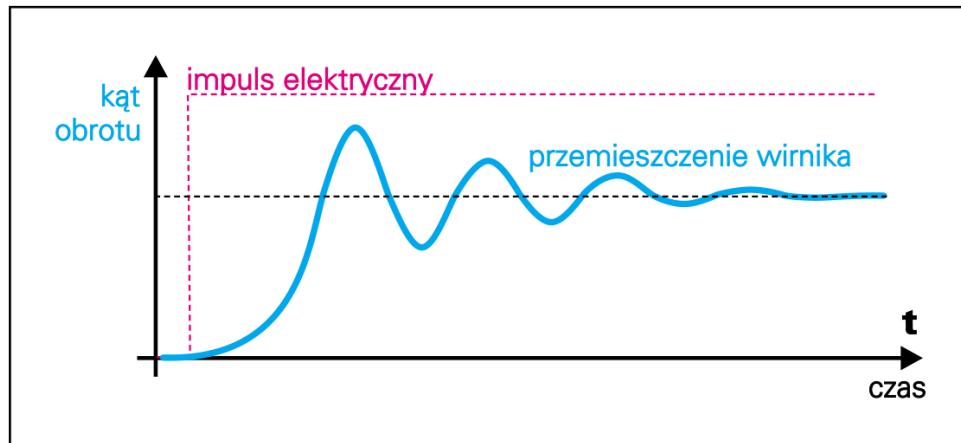
1.4.3. Zalety i wady silników krokom

Możliwość sterowania silnikiem krokom za pomocą impulsów jest dużą zaletą. Przy odpowiednio dobranym układzie napędowym, do pracy w ograniczonym obszarze dokładności silnik krokom nie potrzebuje żadnego sprzężenia zwrotnego (praca w otwartej pętli sprzężenia zwrotnego). Teoretycznie pozycja wirnika jest znana po każdej komutacji wykonanej przez układ sterujący. Niestety, wirnik i jego obciążenie posiadają pewną bezwładność, co sprawia, że przed osiągnięciem ustalonej pozycji występują drgania.

W przypadkach kiedy wymagana jest duża precyzja sterowania, tak duże drgania są bardzo niekorzystne. Wybrane sposoby eliminacji tych problemów to:

- tłumienie mechaniczne poprzez zwiększenie momentu tarcia;
- sprzęgła elastyczne i przekładnie paskowe zastosowane w układzie napędowym, w celu redukcji przenoszonych drgań do obciążenia;

- wybór pracy półkrokowej lub mikrokrokowej: im bardziej przebieg prądu jest zbliżony do sinusoidalnego, tym ciszej i stabilniej pracuje silnik krokowy;
- zastosowanie silnika krokowego o mniejszym skoku.



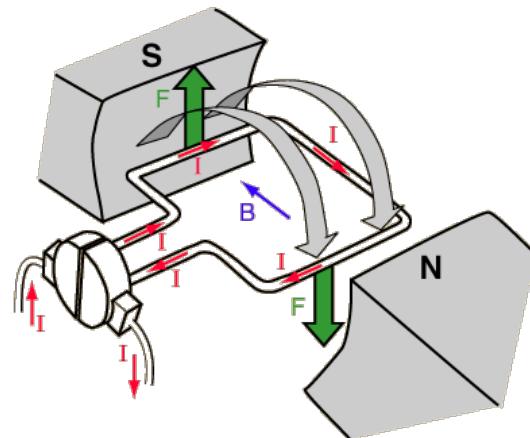
Rysunek 1.21: Wykres czasowy ilustrujący położenie wirnika podczas wykonywania skoku

Źródło: Na podstawie [30]

Należy pamiętać, że o wielkości i częstotliwości oscylacji decyduje głównie masa wirnika oraz wielkość i rodzaj obciążenia.

1.5. Silnik komutatorowy (DC)

Silniki komutatorowe pomimo swoich wad są powszechnie stosowane w produktach konsumenckich (jak i w przemyśle), głównie ze względu na prostotę regulacji prędkości i momentu obrotowego.



Rysunek 1.22: Zasada działania silnika komutatorowego z magnesem trwałym

Źródło: Na podstawie [6]

Komutator jest szeregiem miedzianych płyt umieszczonych na bocznej powierzchni wirnika, po którym ślizgają się (zazwyczaj) węglowe szczotki do których doprowadzone jest napięcie (rys. 1.22). Zadaniem komutatora jest przełączanie

kierunku przepływu prądu w uzwojeniach tak, aby oddziaływanie z polem magnetycznym stojana wprawiło wirnik w ruch obrotowy. Inaczej niż w przypadku silników VR, układ sterujący maszyną jest odciążony od zapewniania odpowiedniej komutacji, co sprawia, że może on być o wiele prostszy w konstrukcji.

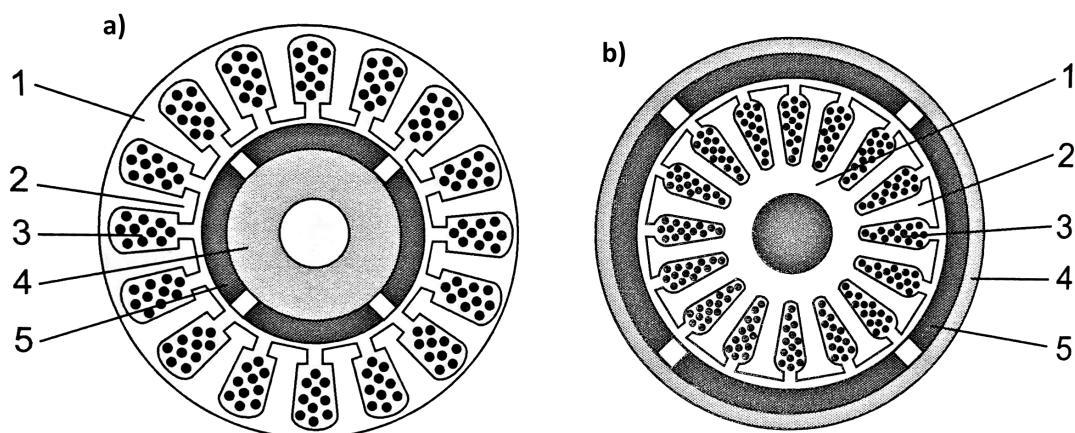
Rysunek 1.22 prezentuje zasadę działania silnika komutatorowego, w którym do wytworzenia pola magnetycznego stojana został użyty magnes trwałego. Tak skonstruowany silnik to PMDC (*Permanent Magnet DC*). Kiedy prąd elektryczny I przepływa przez cewkę znajdująca się w polu magnetycznym B , to siła magnetyczna F (Działająca prostopadle do ułożonych uzwojeń cewki i działającej indukcji) wywołuje moment obrotowy wprawiający wirnik w ruch. W silniku tego rodzaju obroty zależą liniowo od napięcia zasilania, a moment obrotowy od natężenia prądu. Przy wzroście obciążenia spadają obroty, a rośnie moment obrotowy i pobór prądu. Zmiana kierunku obrotów jest uzyskiwana poprzez zmianę bieguności zasilania. Prędkość obrotową można łatwo regulować w granicy od 5% do 110% prędkości znamionowej, z zachowaniem dużego momentu obrotowego. Dobrą stabilizację momentu obrotowego można uzyskać poprzez zasilanie silnika ze źródła prądowego.

Do wytworzenia pola magnetycznego stojana można użyć wzbudzenia elektromagnetycznego zamiast magnesu trwałego. Taki silnik posiada dwa uzwojenia: wirnika i stojana. W zależności od układu połączeń można otrzymać silnik:

- szeregowy,
- bocznikowy,
- szeregowo-bocznikowy.

1.6. Silniki synchroniczne z magnesami trwałymi

Konstrukcja silnika synchronicznego z magnesami trwałymi jest następująca: uzwojenia znajdują się w stojanie, a wirnik jest wykonany z odpowiednio ukształtowanego magnesu. Maszyny te są czasami zaliczane do grupy silników prądu stałego, ale w praktyce pradem stałym zasilany jest jedynie falownik sterujący silnikiem.



Rysunek 1.23: Przekrój typowego silnika trójfazowego z magnesami trwałymi: a) z wirnikiem wewnętrznym, b) z wirnikiem zewnętrznym, 1- jarzmo stojana (twornika), 2- zęby stojana, 3- żłobek z uzwojeniami, 4- wirnik, 5- magnes trwały

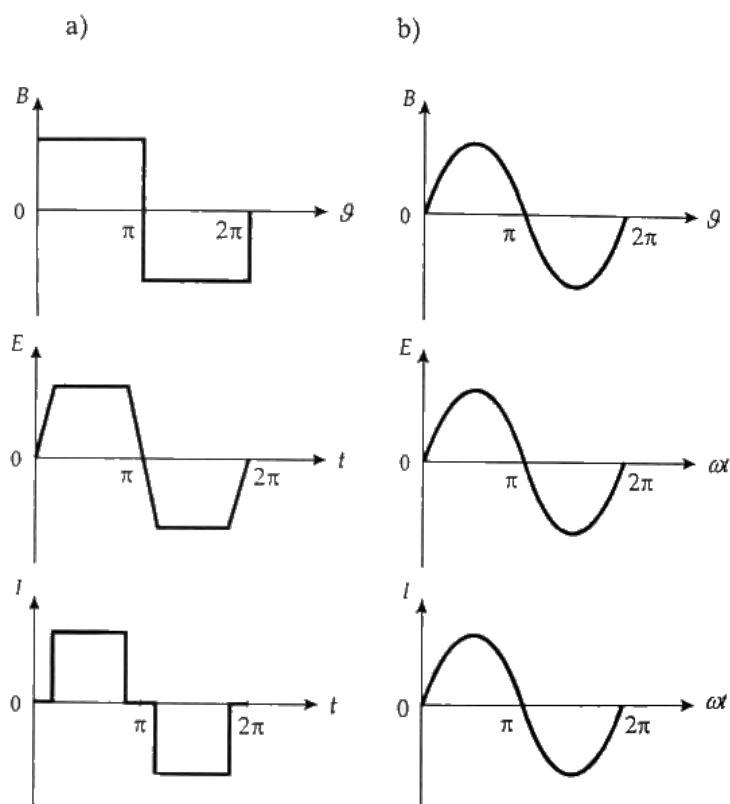
Źródło: Na podstawie [28]

Silniki synchroniczne z magnesami trwałymi są produkowane w dwóch odmianach konstrukcyjnych, różniących się rozkładem indukcji w szczelinie powietrznej oraz przebiegiem prądu i SEM (rys. 1.24):

- PMBLDC (*Permanent Magnet Brushless DC Motor*) – silnik z trapezoidalnym przebiegiem SEM, zasilany prądem o przebiegu w przybliżeniu prostokątnym,
 - PMSM (*Permanent Magnet Synchronous Motor*) – silnik z sinusoidalnym przebiegiem SEM, zasilany prądem o przebiegu sinusoidalnym.

Silniki BLDC są konstrukcyjnie bardzo podobne do silników krokowych – można przyjąć, że silniki BLDC są pewną podgrupą silników krokowych o kroku 180° (silniki 2-fazowe) lub 120° (3-fazowe). Kryterium odróżniające silniki krokowe od BLDC może być takie, że w większości aplikacji dla poprawnej pracy silnika BLDC jego sterownik musi dysponować sygnałem zwrotnym o aktualnym położeniu kątowym wirnika z czujników Hall'a. Silniki krokowe natomiast mogą pracować w otwartej pętli sprzężenia zwrotnego.

PMSM wymagają ciągłego pomiaru położenia wirnika np. przy pomocy enkodera, który im ma większą rozdzielcość, tym dokładniejsze sterowanie umożliwia.



Rysunek 1.24: Przebieg indukcji, SEM i prądu jednej fazy silnika o magnesach trwałych: a) BLDC, b) PMSM

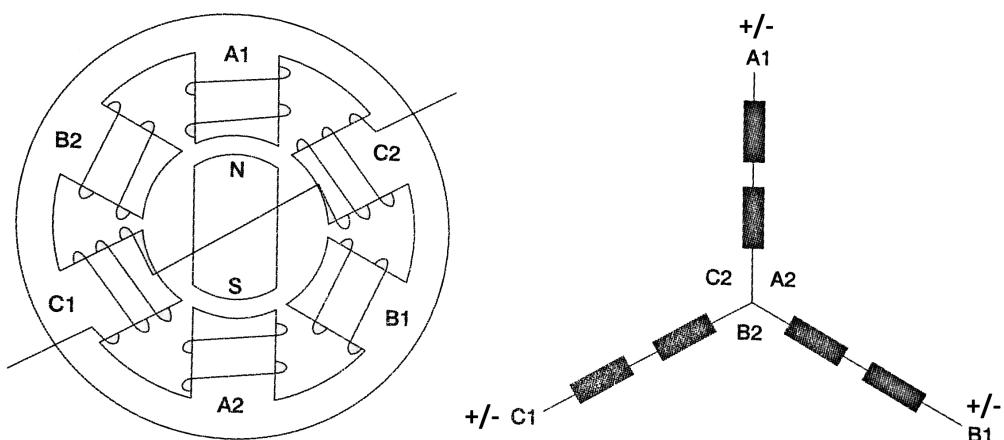
Źródło: Na podstawie [39]

Z tego powodu najczęściej spotyka się rozwiązanie, w którym czujnik położenia wirnika jest zintegrowany z silnikiem we wspólnej obudowie. Wymagana siła elek-

tromotoryczną uzyskuje się przez odpowiednie rozmieszczenie uzwojeń w żłobkach stojana i ukształtowanie rozkładu pola magnesów w szczelinie powietrznej. Obie wersje konstrukcyjne w teorii dają stałą moc, a więc stały moment wzdłuż szczeliny powietrznej. W rzeczywistości jednak moment silnika zawiera pulsacje powodujące nierównomierny ruch wirnika, zwłaszcza w przedziale bardzo małych prędkości [39].

Uzwojenia silnika synchronicznego z magnesami trwałymi dla maszyny 3-fazowej są układane w gwiazdę (rys. 1.25). Prąd z układu sterującego jest doprowadzany do wyprowadzeń A1, B1 i C1 podczas gdy pozostałe końce uzwojeń (A2, B2, C2) są ze sobą zwarte. Oznacza to, że w tym rodzaju maszyny prąd przepływa zawsze przez co najmniej dwa uzwojenia, w jednym z dwóch możliwych kierunków.

Tak jak zaprezentowano na rys. 1.23 silniki bezszczotkowe mogą mieć różną konstrukcję wirnika. W przypadku budowy klasycznej wirnik w kształcie walca jest umieszczony wewnętrznie stojana z uzwojeniami. W silniku z wirnikiem zewnętrznym magnetycznym wirnik ma kształt kubka i obraca się wokół nieruchomego stojana.



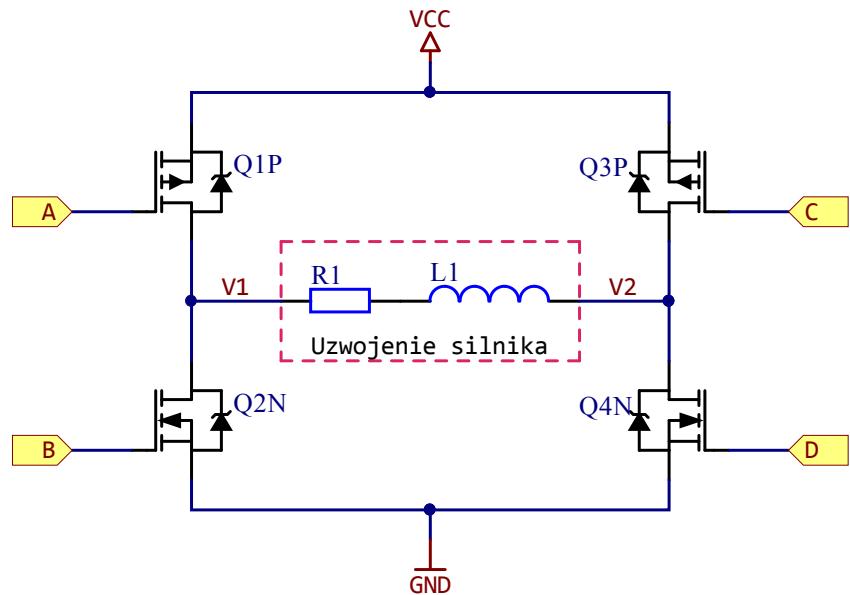
Rysunek 1.25: Konfiguracja uzwojeń trójfazowego silnika synchronicznego w gwiazdę

Źródło: Na podstawie [34]

Do głównych zalet silników BLDC i PMSM należy zaliczyć to że zjawisko rezonansu mechanicznego zredukowane, oraz mogą one uzyskiwać znaczne prędkości obrotowe (powyżej 10000 RPM). Ponadto przy tej samej wartości momentu obrotowego silniki te mogą być mniejsze i lżejsze od krokowego. Z racji tego że nie posiadają komutatora, a co za tym idzie szczotek, ich żywotność jest bardzo długa i jest praktycznie ograniczona żywotnością mechaniki układu napędowego. Największą niedogodnością jest natomiast to, że z silnikiem BLDC i PMSM wymagają bardziej złożonych układów sterowania.

2. Wybrane metody sterowania i zasilania silników elektrycznych

Dzięki indukcyjności w uzwojeniach silników załączając impulsowo napięcie zasilania (w określonych interwałach czasowych) możliwe jest wymuszenie przepływu prądu w uzwojeniu o wartości bliskiej z góry założonej w sterowniku, jednocześnie zasilając maszynę ze źródła napięciowego.



Rysunek 2.1: Typowy mostek typu „H” zbudowany z czterech tranzystorów polowych z izolowaną bramką (Mosfet)

Źródło: Na podstawie [33]

Rys. 2.1 prezentuje konfigurację mostka typu H, zbudowanego z czterech tranzystorów polowych. Tak skonstruowany układ może posłużyć do wymuszenia przepływu prądu w uzwojeniu silnika (które symbolizuje gałąź z rezystorem R1 i cewką L1) w dowolnym kierunku, w zależności od tego które z tranzystorów są aktualnie otwarte/ zamknięte. Jedna połowa mostka (półmostek) tworzy układ *Push-Pull*, który może w punkcie połączenia dwóch tranzystorów (V1 i V2) zapewnić potencjał o wartości przybliżonej do VCC lub GND.

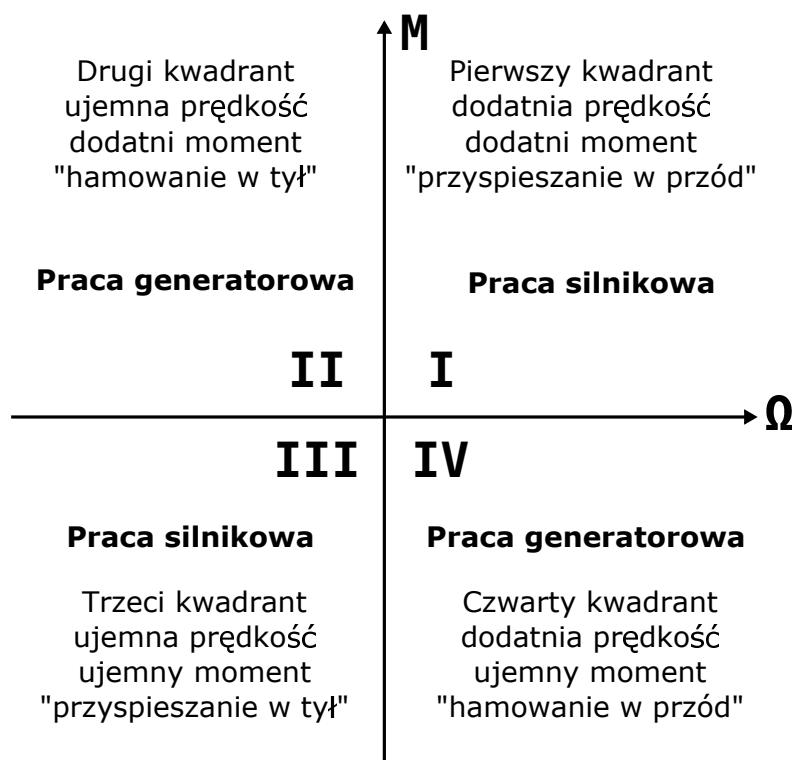
W zależności od aplikacji w mostku są wykorzystywane różne półprzewodniki takie, jak tranzystory bipolarne, polowe czy tranzystory IGBT (*Insulated-gate bipolar transistor*). W komercyjnych sterownikach małej mocy najczęściej są to tranzystory polowe- typu N ze względu na ich lepsze względem typu P właściwości (Mobilność elektronów jest większa niż dziur, a to te pierwsze są nośnikami ładunków w tranzystorze typu N [37]). W takim wypadku jednak do wysterowania górnych tranzystorów potrzebne są dodatkowe układy elektroniczne lub dodatkowe źródło zasilania ze względu na to, że napięcie na bramce musi być wyższe od napięcia na źródle tranzystora, w celu jego pełnego wysterowania.

Diody połączone równolegle do tranzystorów są niezbędne ze względu na indukcyjny charakter uzwojenia. W przypadku ich braku energia zgromadzona na indukcyjności może rozładować się na dwa sposoby:

- nastąpi przebicie tranzystora lub izolacji uzwojenia,
- pojemność pasożytnicza uzwojenia stworzy obwód rezonansowy LC, energia zostanie wypromieniowana w postaci fali elektromagnetycznej i częściowo zamieniona w ciepło poprzez prądy wirowe zaindukowane w rdzeniu [34].

2.1. Tryby pracy

Obserwując kierunek przepływu energii w maszynie elektrycznej, można wyróżnić jej dwa tryby pracy: silnikową oraz generatorową. W stanach ustalonych układu napędowego zjawiska te można zobrazować za pomocą charakterystyk mechanicznych (rys. 2.2), pokazujących zależność momentu M od prędkości Ω . Kształt charakterystyki zależy przede wszystkim od struktury i parametrów układów regulacyjnych i pomiarowych, a w pewnym zakresie również od charakterystyk przekształtnika oraz od typu i parametrów silnika [40].



Rysunek 2.2: Charakterystyki statyczne silnika: Ω - prędkość obrotowa, M - moment obrotowy

Źródło: Na podstawie [33]

Moc na wale maszyny jest wyrażana wzorem:

$$P = \Omega M \quad (2.1)$$

Kiedy jest dodatnia (kwadrant I i III) oznacza to, że maszyna pobiera ją ze źródła napięcia i pracuje jako silnik napędzając element będący obciążeniem. W przypadku kiedy moc jest ujemna (kwadrant II i IV) silnik hamuje obciążenie pracując jako prądnica (praca generatorowa), która poprzez przekształtnik oddaje energię

do źródła [38].

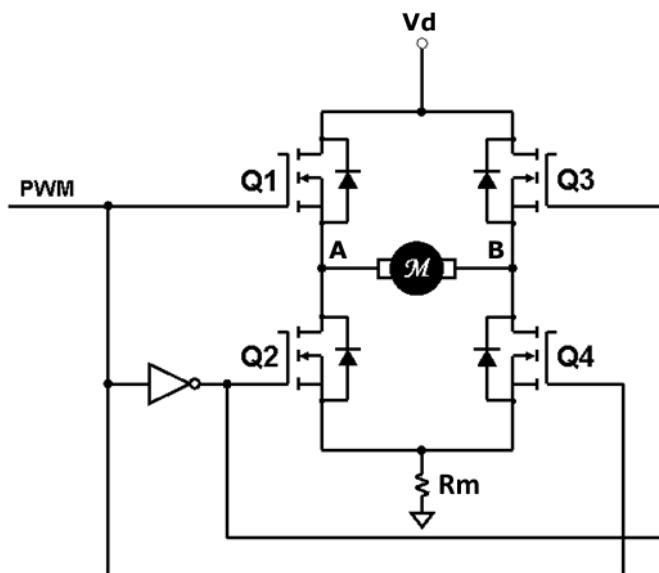
W zależności od topologii układu przekształtnikowego i zastosowanych algorytmów sterowania maszyna elektryczna może mieć możliwość pracy tylko w części kwadrantów lub we wszystkich .

2.2. Sterowanie mostka „H”

Omawiany wcześniej mostek „H” (rys. 2.1), który służy do dostarczania prądu do uzwojeń może być sterowany na dwa sposoby – unipolarny i bipolarny. Oba sposoby mają swoje wady oraz zalety i w związku z tym sterownik, który nie potrafi obsługiwać obu tych trybów może być stosowany tylko w ograniczonych aplikacjach. Ponadto, trybów tych nie należy mylić z trybami zasilania silnika krokowego, które zostały omówione w podrozdziale 1.4.2.

2.2.1. Modulacja bipolararna

Sterowaniem bipolarnym nazywa się takie sterowanie, w którym dla kolejnych okresów sygnałów sterujących PWM, wartość napięcia na uzwojeniu zmienia się w zakresie od V_d (napięcia źródła napięciowego zasilającego silnik) do $-V_d$. Ten sposób wysterowania mostka przedstawia schemat z rysunku 2.3. Jako obciążenie został przyjęty silnik DC.

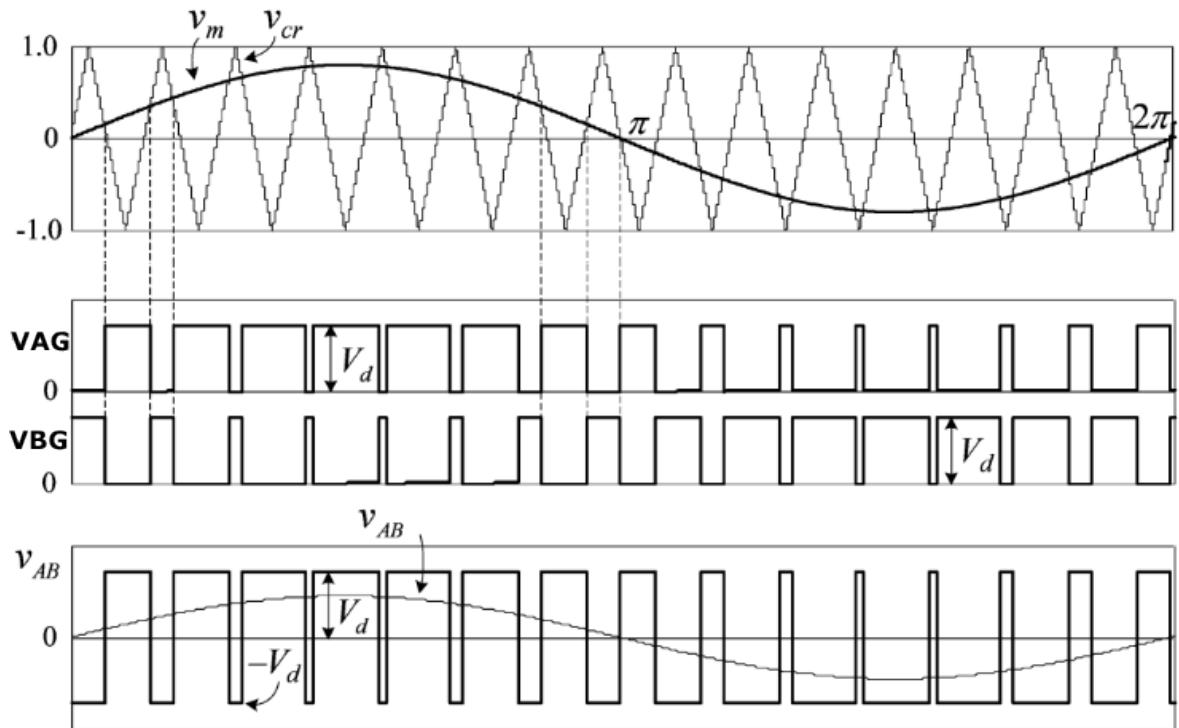


Rysunek 2.3: Silnik DC sterowany bipolarnie

Źródło: Na podstawie [17]

Widoczne na schemacie połączenia i zastosowana bramka NOT jest tylko obrazowym odzwierciedleniem sposobu sterowania implementowanego programowo w sterowniku. Zakładając sytuację bez obciążenia wirnika, wypełnienie impulsu powyżej 50% wymusi ruch „do przodu”, a poniżej 50% – „do tyłu”. Taka maszyna jest elektrycznym odpowiednikiem mechanicznej przekładni bezstopniowej CVT (*Continuously Variable Transmission*) [17]. W tego typu maszynie pozycję spoczynkową otrzymuje się poprzez zadawanie sygnału o wartości środkowej.

Przy sterowaniu bipolarnym maszyna pracuje częściowo w czwartym kwadrancie charakterystyki (rys. 2.2). Tak długo, jak przyłożone do silnika średnie napięcie jest tej samej polaryzacji co SEM generowana w uzwojeniu, i jego amplituda jest wyższa od amplitudy tej SEM, wtedy maszyna pracuje w trybie silnikowym. W przypadku kiedy, napięcie zasilania i SEM mają tą samą polaryzację, lecz SEM ma wyższą amplitudę, to maszyna pracuje w trybie generatorowym.



Rysunek 2.4: Przebiegi napięć w modulacji bipolarnej

Źródło: Na podstawie [29]

Rys. 2.4 prezentuje przebiegu towarzyszące modulacji bipolarnej, zastosowanej w do przykładu mostka z rys. 2.3. Zadany jest sygnał sinusoidalny V_m , a PWM jest taktowany przebiegiem V_{cr} . V_{AG} i V_{BG} to napięcia odpowiednio między potencjałami A i B a masą, natomiast V_{AB} to spadek na silniku.

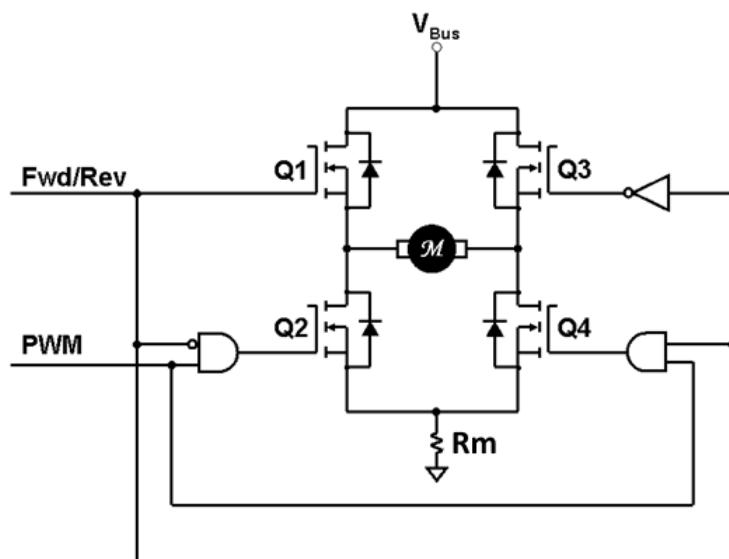
Zaletą sterowania bipolarnego jest to, że ze sprzętowego punktu widzenia wymaga bardzo niewiele zasobów. W przypadku najprostszego sterowania wymagany jest tylko jeden sygnał PWM. Oznacza to, że nawet najtańszy mikroprocesor może posłużyć do zbudowania sterownika do silnika DC. Ważniejszą zaletą jednak jest to, że używając tylko jednego rezystora pomiarowego (R_m na rys. 2.3) istnieje możliwość pomiaru prądu płynącego w uzwojeniu w dowolnym momencie. Daje to możliwość swobodnego doboru sposobu próbkowania prądu i upraszcza układ sterujący.

Najpoważniejszą wadą sterowania bipolarnego jest większa pulsacja prądu w stosunku do metod opisanych w dalszej części pracy.

2.2.2. Sterowanie unipolarne

Zasadę działania sterowania unipolarnego można zaprezentować bazując na rysunku 2.5. Przy jego pomocy możliwe jest wysterowanie silnika w taki sposób, aby jego wirnik mógł obracać się w obu kierunkach (kwadranty I i III). Aby pracować w pierwszym kwadrancie, tranzystor Q1 jest włączony (otwarty, wprowadzony w stan nasycenia), a na bramkę Q4 podawany jest sygnał PWM. Kiedy Q4 jest otwarty, prąd płynie od V_d przez Q1, dalej przez uzwojenie i przez Q4 do masy poprzez rezistor R_m . Kiedy Q4 się zamyka w uzwojeniu dalej jest prąd ze względu na jego indukcyjność. Zwiększający się potencjał w punkcie połączenia drenu Q4 i źródła Q3 sprawia że dioda tranzystora Q3 zaczyna przewodzić. Z racji tego, że Q1 jest cały czas otwarty, obwód prądowy zamknie się właśnie przez ten tranzystor, a nie przez obwód zasilania.

Aby obracać wirnikiem w drugą stronę należy otworzyć Q3 i kluczować (zasilać impulsowo) Q2. W takim wypadku silnik pracuje w trzecim kwadrancie charakterystyki statycznej (negatywna prędkość i moment. Niezależnie od kierunku przepływu prądu w silniku, napięcie na niego podawane zawsze zmienia się w graniczy od V_d do GND, a nie jak w schemacie obrazującym sterowanie bipolarne (rys. 2.3) od V_d do $-V_d$. Sprawia to, że w takim układzie silnik nie może oddawać energii do zasilania, bo prąd jest „zamknięty” w oczku utworzonym z uzwojenia, Q1 i Q3.

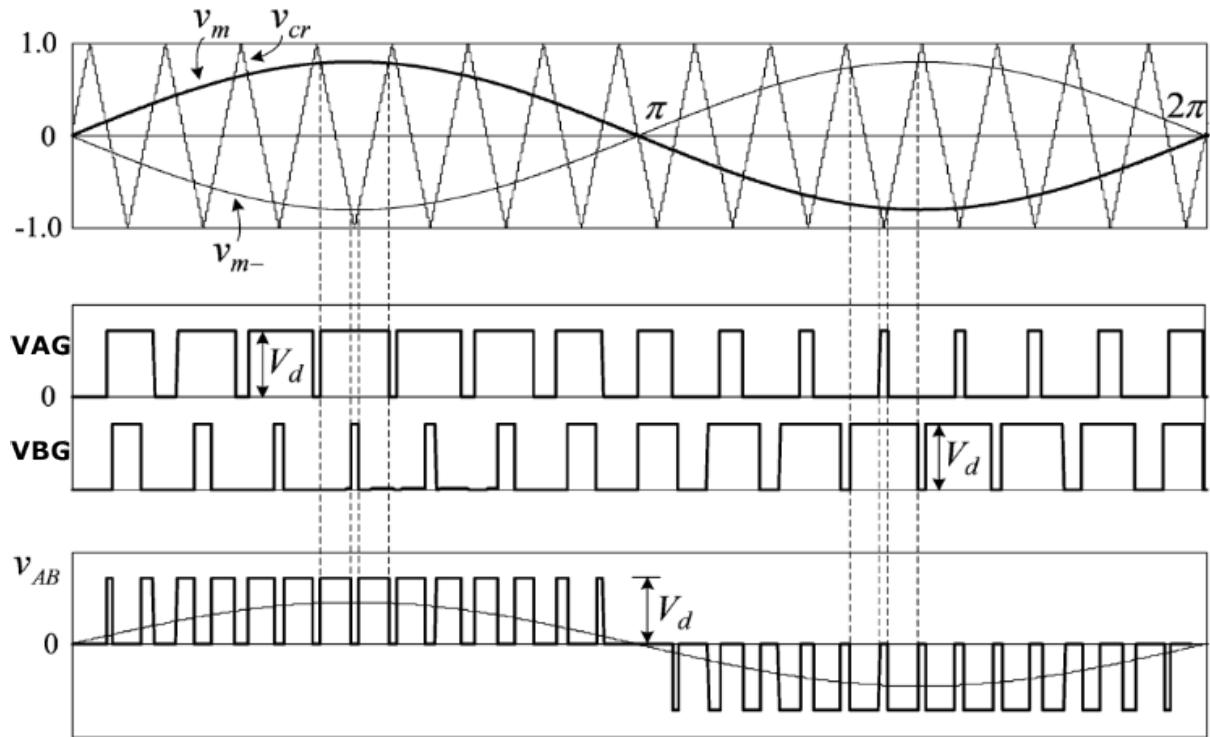


Rysunek 2.5: Silnik DC sterowany unipolarnie

Źródło: Na podstawie [17]

Zaletą prezentowanej topologii jest to, że wymagany jest tylko jeden sygnał PWM do sterowania pojedynczym mostkiem (oraz linia wejściowa ustalająca kierunek obrotów). Ponadto, w danym momencie kluczowany jest tylko jeden tranzystor, a więc co za tym idzie, straty spowodowane przełączaniem są mniejsze niż w sterowaniu bipolarnym.

Modulacja unipolarna wywołuje mniejszą amplitudę tętnień prądu niż modulacja bipolarna przy tej samej częstotliwości modulacji [40]. Z tych przyczyn sterownik korzysta właśnie z tego typu sterowania.



Rysunek 2.6: Przebiegi napięć w modulacji unipolarnie

Źródło: Na podstawie [29]

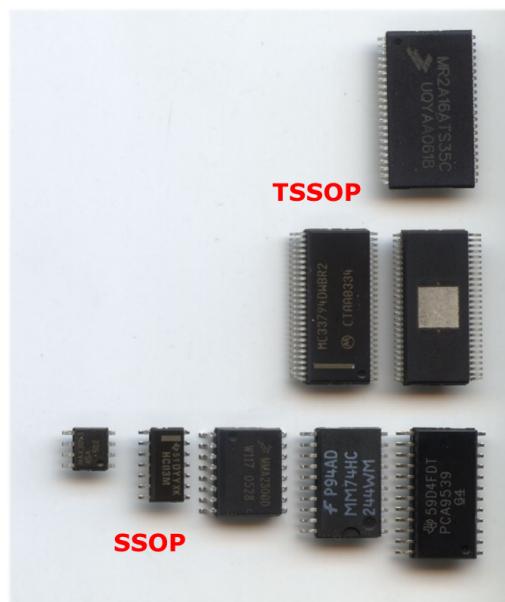
Bez względu na zastosowaną technikę modulacji, do wszystkich silników zasilanych napięciem stałym można użyć mostka typu „H” w celu realizacji tej modulacji.

3. Przegląd wybranych rozwiązań sterowników silników

Na rynku dostępnych jest bardzo wiele rozwiązań pozwalających w szybki sposób zacząć sterować konkretnym rodzajem silnika. Ceny poszczególnych sterowników różnią się od siebie znaczco, w taki sam sposób jak różni się ich funkcjonalność. Są też dostępne projekty bezpłatne (*Open Source i Open Hardware*) w rozumieniu konkretnych rozwiązań technologicznych, czy też „*Know How*”.

3.1. Sterowniki jedno-układowe

Pierwszą grupą sterowników są urządzenia, które dostarczają użytkownikowi jedynie część funkcjonalności związanej ze sterowaniem silnikiem- najczęściej tą która jest najtrudniejsza do zaimplementowania programowo lub sprzętowo na płytce PCB (*Printer Circuit Board*) urządzenia. Ich cena w porównaniu do układów zapewniających pełne sterowanie jest wielokrotnie niższa, a ponadto niejednokrotnie są one częścią tych układów.



Rysunek 3.1: Przykładowe typy obudów w jakich są produkowane układy scalone do sterowania silnikami. Na górze obudowy TSSOP (*Thin Shrink Small Outline Package*) i SSOP (*Shrink small outline package*) na dole

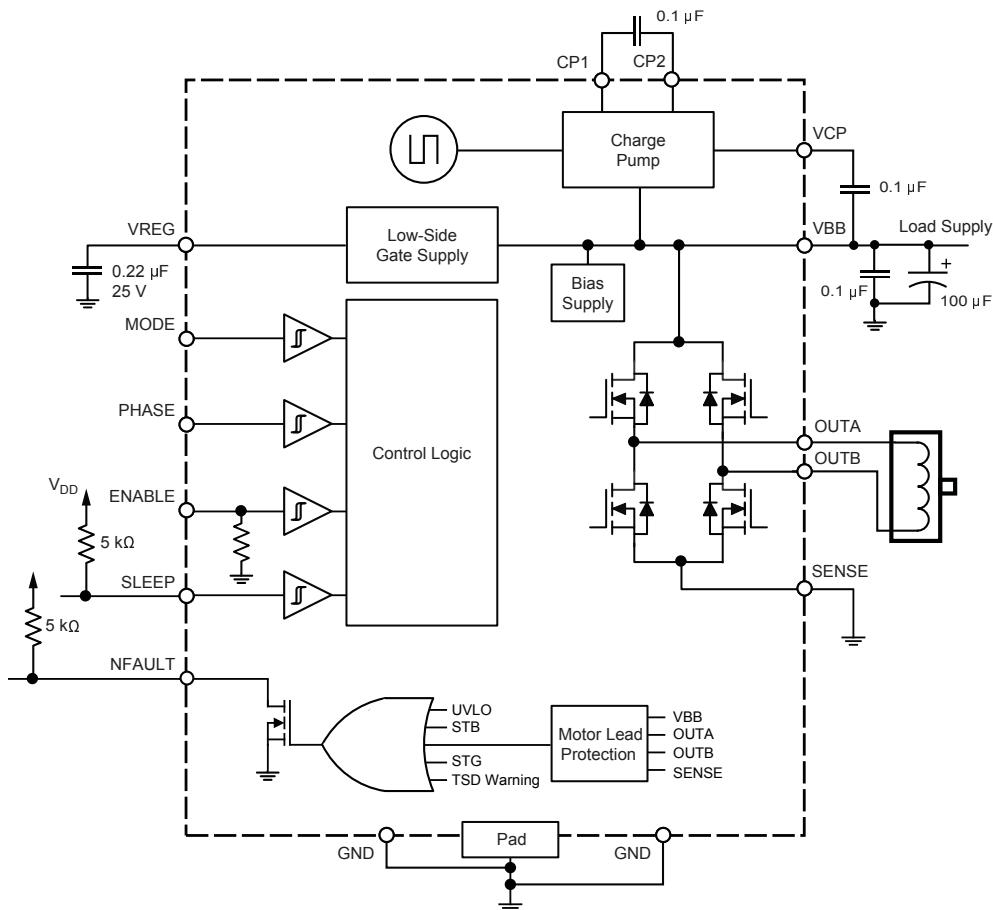
Źródło: Na podstawie [33]

Funkcjonalność danego układu mocno zależy od jego ceny. Wraz z jej wzrostem algorytmy zaimplementowane w sterowniku są coraz lepsze, a dodatkowych potrzebnych komponentów elektronicznych do ich poprawnego działania potrzeba coraz mniej (np. końcówki mocy takie jak mostki H są zintegrowane w układzie). Ponadto sterowniki do obsługi silników, które wymagają bardziej skomplikowanego sterowania (np. krokowe lub BLDC) są droższe od tych, które służą do zasilania prostych maszyn (np. silniki komutatorowe DC).

3.1.1. AD3950 - Allegro MicroSystems

Układ AD3950 produkowany przez firmę Allegro MicroSystems jest urządzeniem służącym do sterowania silnikami komutatorowymi DC, aczkolwiek nadaje się też do

silników VCM. Najwyższe napięcie przy jakim może pracować to 36 V, a chwilowy prąd szczytowy, który może dostarczyć do uzwojenia to $\pm 2.8A$.



Rysunek 3.2: Funkcjonalny schemat blokowy układu A3950

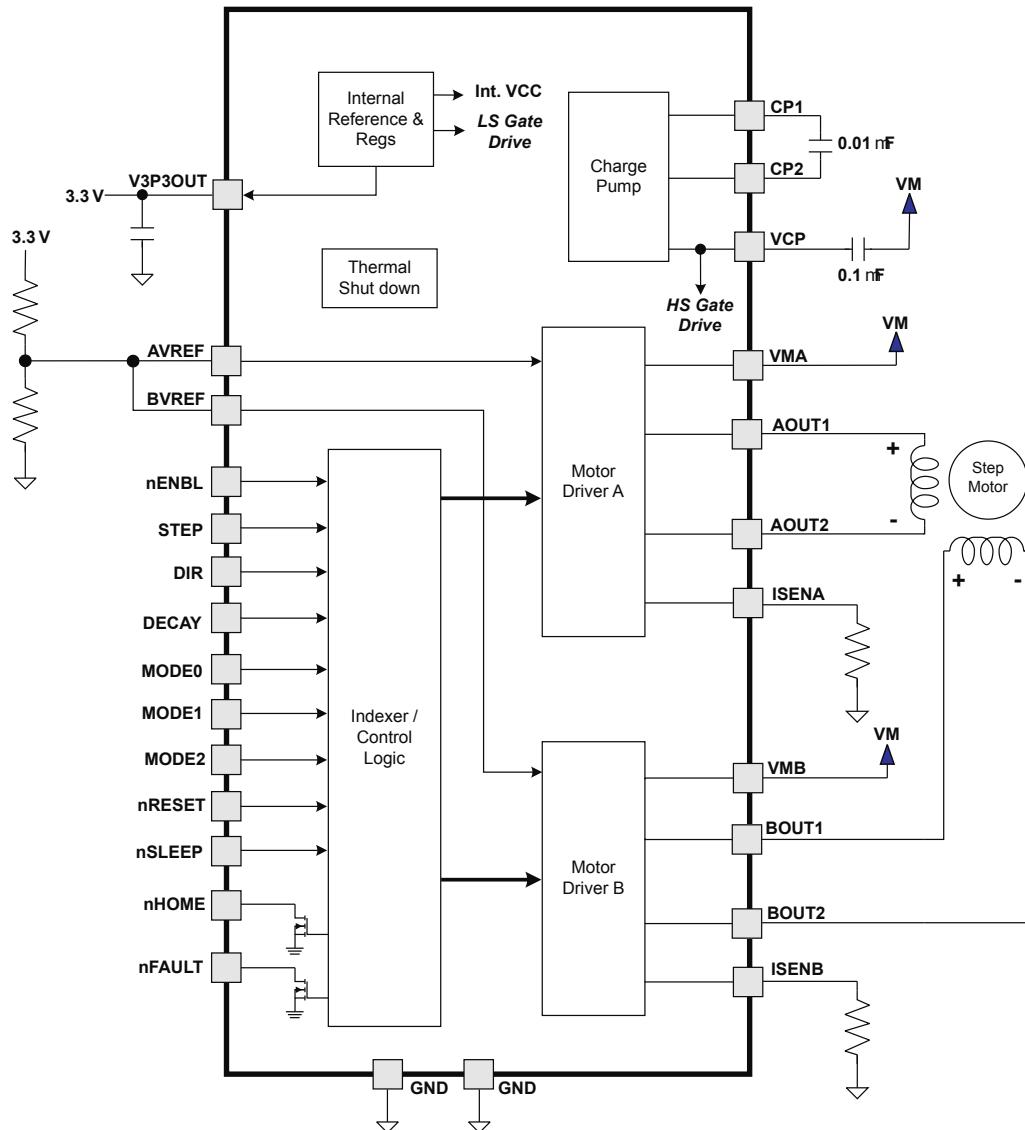
Źródło: Na podstawie [1]

Układ posiada w swojej obudowie mostek H i niezbędne podzespoły pozwalające na sterowanie nim używając pojedynczego sygnału PWM w celu ustalenia prędkości obrotowej/ momentu i kilku linii typu I/O (*Input/ Output*), które służą między innymi do zmiany kierunku obrotów (poprzez zmianę kwadrantów pomiędzy pierwszym, a trzecim) i trybu hamowania (*Slow Decay* jak w sterowaniu unipolarnym lub *Fast Decay* w celu szybkiego hamowania, tak jak przy sterowaniu bipolarnym, tyle że sterownik dodatkowo nie pozwala na wejście silnika w tryb generatorowy zamkając odpowiednie klucze).

3.1.2. DRV8825 - Texas Instruments

Kontroler DRV8825 firmy Texas Instruments jest bardzo popularnym układem zarówno wśród hobbystów jak i profesjonalnych użytkowników. Urządzenie służy do zasilania pojedynczego silnika krokowego bipolarnego lub dwóch silników DC. Sterowanie układem polega na podawaniu na wejście STEP impulsów napięciowych, gdzie w zależności od dokonanej za pomocą linii I/O, każdy taki impuls sprawia że silnik robi pełen krok lub mikrokrok (możliwość ustawienia ziarnistości od $\frac{1}{2}$ do $\frac{1}{32}$ kroku). Zmieniając częstotliwość podawania impulsów i stany linii MODE0 - MODE2 możliwe jest w miarę płynne sterowanie prędkością obrotową silnika krokowe-

go. W urządzeniach takich jak skanery czy drukarki, w których prędkość obrotowa jest zazwyczaj stała, a wymagana dokładność pozycji wysoka, kontroler DRV8825 sprawdza się bardzo dobrze.



Rysunek 3.3: Funkcjonalny schemat blokowy układu DRV8825

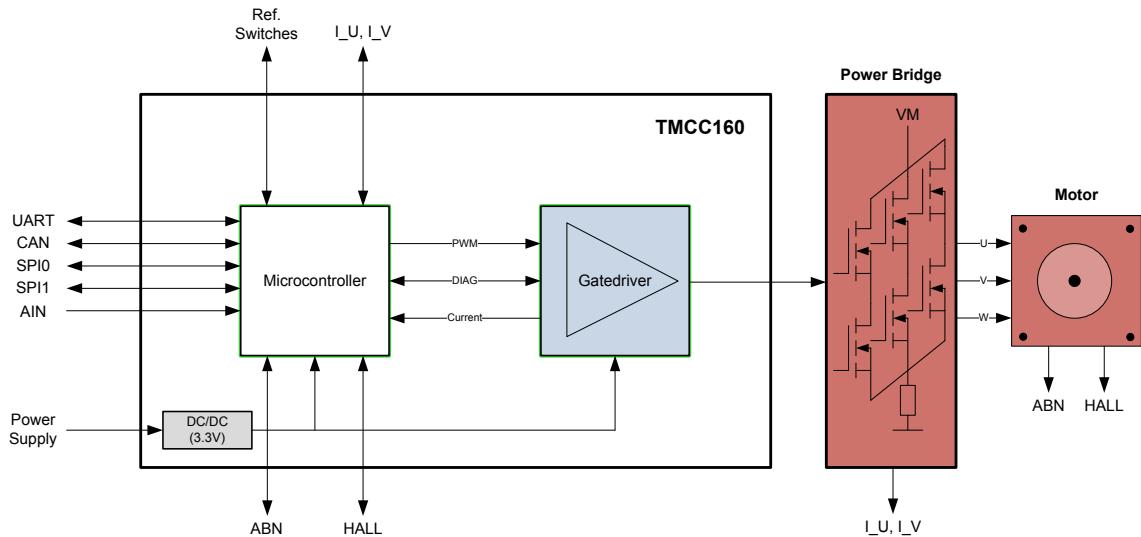
Źródło: Na podstawie [16]

Największą zaletą urządzenia jest taka, że odciąża on sterujący nim układ nadzędny od rygorystycznego, wykonywanego w czasie rzeczywistym sterowania silnikiem krokowym. Znacząca wada jest natomiast taka, że pomimo że możliwe jest zadanie maksymalnego prądu jakim mają być zasilane uzwojenia, to nie ma możliwości jego bezpośredniego pomiaru przez układ nadzędny.

3.1.3. TMCC160 - Trinamic

Trzecim prezentowanym układem jest sterownik do zasilania silników BLDC i PMSM firmy Trinamic – TMCC160. Poziom integracji tego urządzenia jest o wiele wyższy niż w przypadku dwóch pozostałych chipów. W jego strukturze znajduje się wydajny mikrokontroler typu Cortex® M4 z rodziny ARM, dzięki czemu komuni-

kacja z może odbywać się poprzez jeden z dostępnych protokołów szeregowych. W układzie jest też zintegrowana przetwornica DC-DC przez co uproszczona zostaje sekcja zasilania układu.



Rysunek 3.4: Ogólny schemat blokowy układu TMCC160

Źródło: Na podstawie [18]

Urządzenie nie posiada zintegrowanych kluczy tranzystorowych co może być zaletą, gdyż dzięki temu można go dostosować do aplikacji niskiej i dużej mocy. TMCC160 potrafi nawet współpracować z czujnikami Hall'a i enkoderami inkrementalnymi oraz ma zaimplementowane algorytmy sterowania w zamkniętej pętli sprzężenia zwrotnego.

Po dodaniu kilku elementów pasywnych i złącz do płytki PCB z TMCC160, użytkownik otrzymuje funkcjonalny moduł sterownika, który może być sterowany zarówno z mikrokontrolera o bardzo niskich zasobach (wysoki poziom integracji) jak i z komputera, dzięki mnogości dostępnych interfejsów szeregowych (np. RS232 lub CAN).

3.2. Sterowniki pełnowymiarowe

Druga grupa sterowników to urządzenia samodzielne, które do pracy potrzebują jedynie podłączenia do źródła zasilania, sygnału sterującego z układu wyższego poziomu (najczęściej interfejs szeregowy) jeśli komunikacja nie jest bezprzewodowa i samego silnika. Jeśli układ pracuje jako serwomechanizm to wymagane jest jeszcze podłączenie urządzenia dostarczającego sygnał zwrotny, np. enkodera. Istnieje wiele rodzajów kontrolerów tego typu w zakresie mocy od kilku watów do kilkunastu kilowatów.

3.2.1. Simplexmotion 100A - Simplex Motion AB

Urządzenie Simplexmotion 100A jest sterownikiem o wysokim poziomie integracji, ze względu na to, że nawet sterowany silnik znajduje się w obudowie układu. Maszyna znajdująca się w urządzeniu to bezszczotkowy silnik BLDC prądu stałego o mocy 100 W (wartość szczytowa 400 W) i maksymalnej rozwijanej prędkości obro-

towej 6000 rpm. Umieszczając w jednej obudowie silnik i układ sterowania, producent odciążył użytkownika z konieczności dokonywania wielu niezbędnych połączeń potrzebnych do poprawnej pracy samego silnika. Wyprowadzone na zewnątrz zostały jedynie zaciski:

- interfejs usb do zmiany nastaw kluczowych parametrów,
- interfejs RS485 działający w standardzie Modbus,
- wejście na kwadraturowy enkoder inkrementalny,
- wyprowadzenia ogólnego przeznaczenia (I/O, PWM, wejścia analogowe),
- zaciski zasilania (12 - 24 V).



Rysunek 3.5: Inteligentny, wysoce zintegrowany serwo mechanizm Simplexmotion 100A szwedzkiej firmy Simplex Motion. Wymiary: 72 x 64 x 74 mm, waga: 660 gram. Cena: \$353

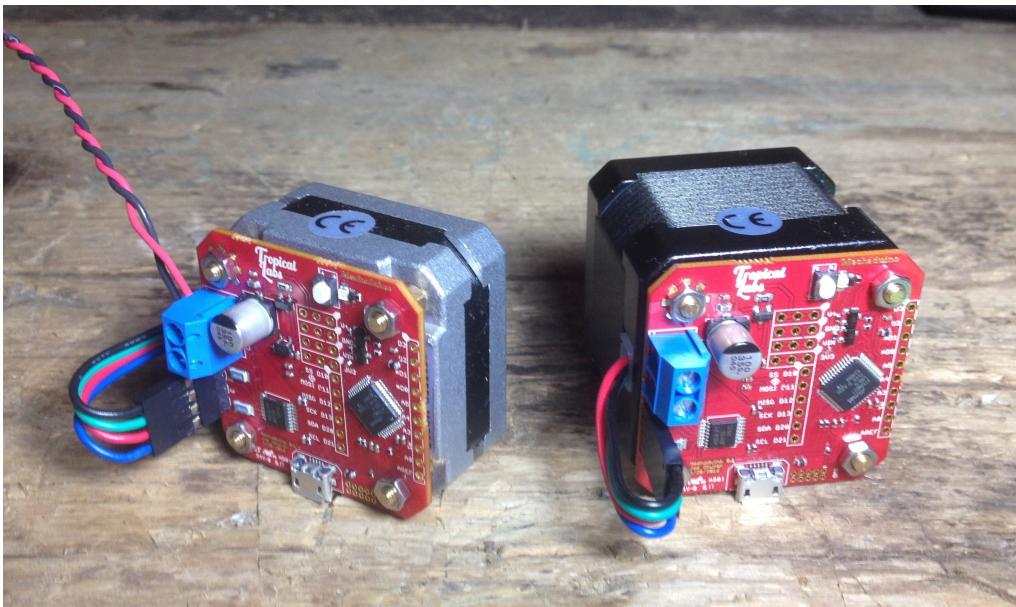
Źródło: Na podstawie [13]

Ponadto producent zapewnia API (.Net, Linux i LabView), dzięki któremu użytkownicy mogą w prosty sposób zaprojektować własną aplikację sterującą tym serwomechanizmem z komputera. Wysoki poziom integracji Simplemotion 100A jest dużą zaletą w sytuacji kiedy zawarty w nim silnik odpowiada naszej aplikacji i czas na wdrożenie rozwiązania jest niewielki. Dodatkowo, pomimo że wymiary urządzenia są kompaktowe to, nie zmieści się ona wszędzie tam, gdzie zmieściłby się sterownik oddzielony od sterowanej maszyny.

3.2.2. Mechaduino - Tropical Labs

Kolejny sterownik jest przykładem kompaktowego projektu udostępnionego na zasadach Open Source/ Hardware, służącego do sterowania silnikami krokowymi. Projekt bazuje na sprzęcie z platformy Arduino Zero, co oznacza że firmware pracujący na sterującym mikrokontrolerze ATSAMD21G18 (Rdzeń ARM Cortex® M0+)

może być napisany w środowisku programistycznym Arduino. Cechą charakterystyczną przedstawionego rozwiązania jest montaż sterownika z tyłu silnika standardu NEMA 17. W dolnej warstwie sterownika znajduje się enkoder magnetyczny, który współpracuje z magnesem przyklejonym na końcu wału silnika. W zamkniętej pętli dokładność pozycjonowania wału silnika wynosi 0.1° po dokonaniu procedury kalibracji enkodera.



Rysunek 3.6: Sterownik do silników krokowych Mechaduino firmy Tropical Labs, kompatybilny sprzętowo i programowo z platformą Arduino Zero

Źródło: Na podstawie [8]

Kompaktowa końcówka mocy znajdująca się na PCB to układ A4954 firmy Allegro Microsystems. Są to zintegrowane w układzie scalonym dwa pełne mostki H, które mogą dostarczyć do silnika prąd o natężeniu szczytowym 2 A. Komunikacja z urządzeniem odbywa się poprzez port szeregowy USB lub I^2C (*Inter-Integrated Circuit*).

Zaletą urządzenia jest otwartość kodu, dzięki czemu można dostosować sterownik do potrzeb swoich aplikacji. Mechaduino jest jednak projektem hobbystycznym, którego oprogramowanie i hardware jeszcze nie są w pełni dopracowane, przez co nie nadaje się jeszcze do profesjonalnego zastosowania.

3.2.3. Sterowniki firmy Technosoft

Technosoft jest wiodącą firmą w zakresie opracowywania specjalizowanych rozwiązań i produkcji systemów napędowych. Ilość sterowników dostępnych w ofercie firmy jest bardzo duża, dlatego w tej sekcji zostanie opisana tylko jedna rodzina, która była inspiracją do stworzenia prezentowanego dalej w pracy sterownika do silników. Płytki *Plug-in* są urządzeniami o wysokim poziomie integracji, ale inaczej niż poprzednio przedstawione sterowniki nie posiadają zintegrowanego silnika. Jednostką centralną jest szybki mikrokontroler zmiennoprzecinkowy, który odpowiada za całe sterowanie.



Rysunek 3.7: Przykładowy sterownik z rodziny *Plug-in Intelligent Servo Drives*, PIM2403. Firma Technosoft

Źródło: Na podstawie [15]

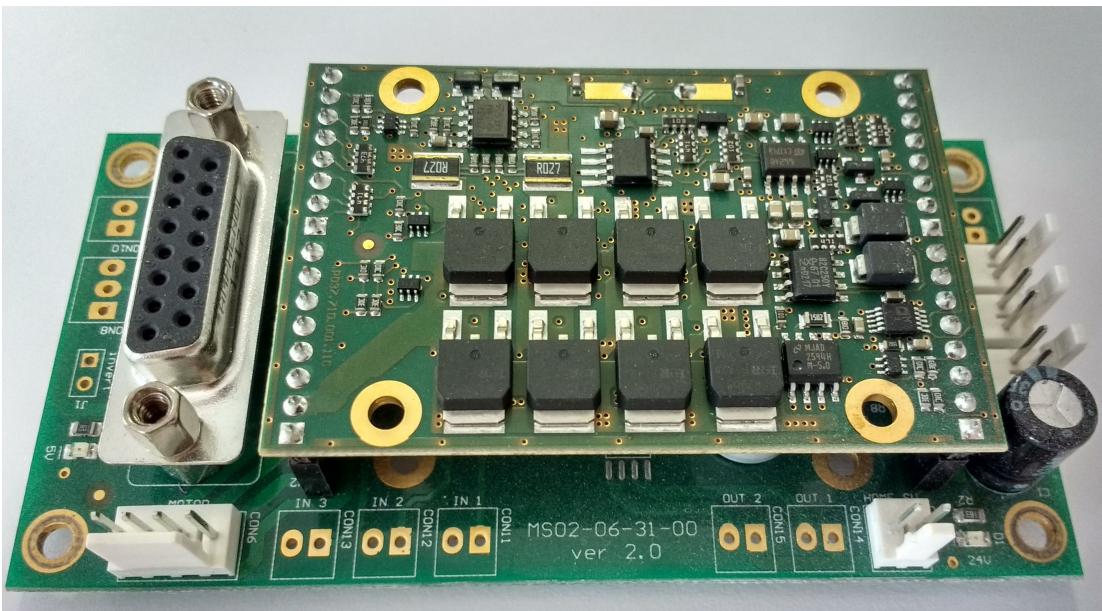
Na PCB nie ma żadnych sterowników pośredniczących, wszystkie algorytmy związane ze sterowaniem silnikiem są wykonywane w mikrokontrolerze. Rodzaje silników obsługiwanych przez kontroler to:

- krokowy,
- BLDC,
- PMSM,
- DC,
- VCM.

Sama płytka sterownika nie wystarczy jednak do rozpoczęcia pracy, bo znajdują się na niej tylko pośrednie peryferia i brakuje odpowiednich wyprowadzeń. Rysunek 3.8 prezentuje sterownik *Plug-in* zamontowany na przykładowej płytce bazowej, która umożliwia komunikację oraz podłączenie do sterownika jako całości wielu końcowych peryferiów potrzebnych do działania. Poprzez złącza można podłączyć:

- zasilanie,
- silnik,
- enkoder inkrementalny,
- czujnik Hall'a działający jako krańcówka,
- interfejsy CAN i RS232.

Jest to przykładowa konfiguracja, która została przystosowana do konkretnej aplikacji. Firma Technosoft udostępnia aplikację *Easy Motion Studio*, dzięki której możliwe jest nie tylko ustawianie parametrów regulatorów itd., ale nawet zaprogramowanie sterownika w niemal dowolny sposób, używając opracowanego przez firmę języka skryptowego *Motion Programming Language*.



Rysunek 3.8: Kontroler PIM2403 zamontowany na płytce bazowej MS02-06-31 firmy Mechatronic Systems

Źródło: Na podstawie [33]

Rozwiązania oferowane przez tego producenta cechuje możliwość szybkiego wdrożenia do projektu. Cena takiego rozwiązania jest jednak wysoka.

Podsumowanie

Na rynku jest dostępnych bardzo wiele rozwiązań pozwalających na sterowanie wszelkiego rodzaju silnikami elektrycznymi. Przedstawione przykładowe kontrolery stanowią tylko małą część dostępnych na rynku produktów, ale ukazują formy w jakich buduje się sterowniki. Najtańszymi układami są jedno-chipowe urządzenia, które do poprawnej pracy potrzebują bardzo wielu elementów dodatkowych. Wraz z poziomem integracji cena rośnie, a dodatkowo trzeba dopłacić za możliwość szybkiego wdrożenia produktu do swojej aplikacji i uniwersalność (pod względem wykonywanych programów i obsługi silników).

Pomimo wszystkich zalet jakie posiadają zaprezentowane w rozdziale 3 urządzenia, żadne z nich nie jest idealne- najmniejsze układy mają za małe możliwości, a wysoko zintegrowane sterowniki są za drogie (i jak na swoją cenę, także mają małe możliwości). Z tych właśnie powodów powstaje uniwersalny sterownik do silników, którego dokładny opis znajduje się w dalszych rozdziałach pracy.

4. Koncepcja projektowanego sterownika

Istnieje wiele powodów które mogły by warunkować zaprojektowanie kompletnego sterownika do silników, zamiast skorzystania z gotowych rozwiązań (np. prezentowanych w rozdziale 3).



Rysunek 4.1: Przykładowy produkt firmy Mechatronic Systems: Automat P10- maszyna typu *Pick and Place* (układająca komponenty elektroniczne na płytach PCB) z wbudowanym dyspenserem pasty lutowniczej/ kleju.

Źródło: Na podstawie [9]

Firma Mechatronic Systems zajmująca się dostarczaniem produktów i rozwiązań z dziedziny technologii SMT (*Surface Mount Technology*), dla której konstruowany jest sterownik stanowiący temat niniejszej pracy używa w swoich konstrukcjach sterowników Plug-in firmy Technosoft, prezentowanych w sekcji 3.2.3. Postanowiono zastąpić je własnym rozwiązaniami, dlatego że:

- po dokonaniu rachunku kosztów okazało się, że długoterminowo opracowanie własnego kontrolera będzie bardziej opłacalne niż regularny zakup gotowych sterowników;

- technologia dostarczana przez Technosoft przestała spełniać wymagania dynamicznie rozwijających się aplikacji w firmie Mechatronic Systems;
- tworząc własny sterownik uzyskuje się bardzo duże doświadczenie z dziedziny napędów i sterowania;

Zostaną dalej przedstawione najważniejsze wymagania i warunki brzegowe, które musi spełnić konstruowane urządzenie w celu zastąpienia nim sterownika Technosoftu.

4.1. Uniwersalność

Mechatronic Systems dostarcza kilka rodzajów produktów, a każdy z nich pracuje z innymi silnikami (Tabela 4.1).

Tabela 4.1: Typ maszyny produkowanej przez Mechatronic Systems i używane przez nią silniki wraz ze sterownikami

Źródło: Na podstawie [33]

| Maszyna | Stosowane silniki | Stosowane sterowniki |
|--------------------------------------|-------------------|--------------------------------|
| Automaty pick and place i dyspensery | krokowe, VCM, DC | Technosoft |
| Drukarki szablonowe | BLDC | inne pełnowymiarowe dedykowane |
| Podajniki automatyczne | DC, krokowe | własne dedykowane |
| Konwejery | DC, BLCD | własne dedykowane |

Na obecnym etapie w części urządzeń stosowane są uniwersalne sterowniki Technosoft'u, które są odpowiednio zaprogramowane do pracy z określonym rodzajem silnika lub własne proste sterowniki, dedykowane pod konkretną maszynę elektryczną (np. silnik komutatorowy DC).

Rysunek 4.2 prezentuje automat P30 z oznaczonymi osiami. Poprzez pojedynczą oś należy rozumieć jeden układ napędowy w którym zastosowany jest jeden silnik i jeden kontroler sterujący. W tego rodzaju maszyny jest aż sześć układów napędowych, a co za tym idzie sześć zastosowanych sterowników typu Plug-in Technosoft'u.

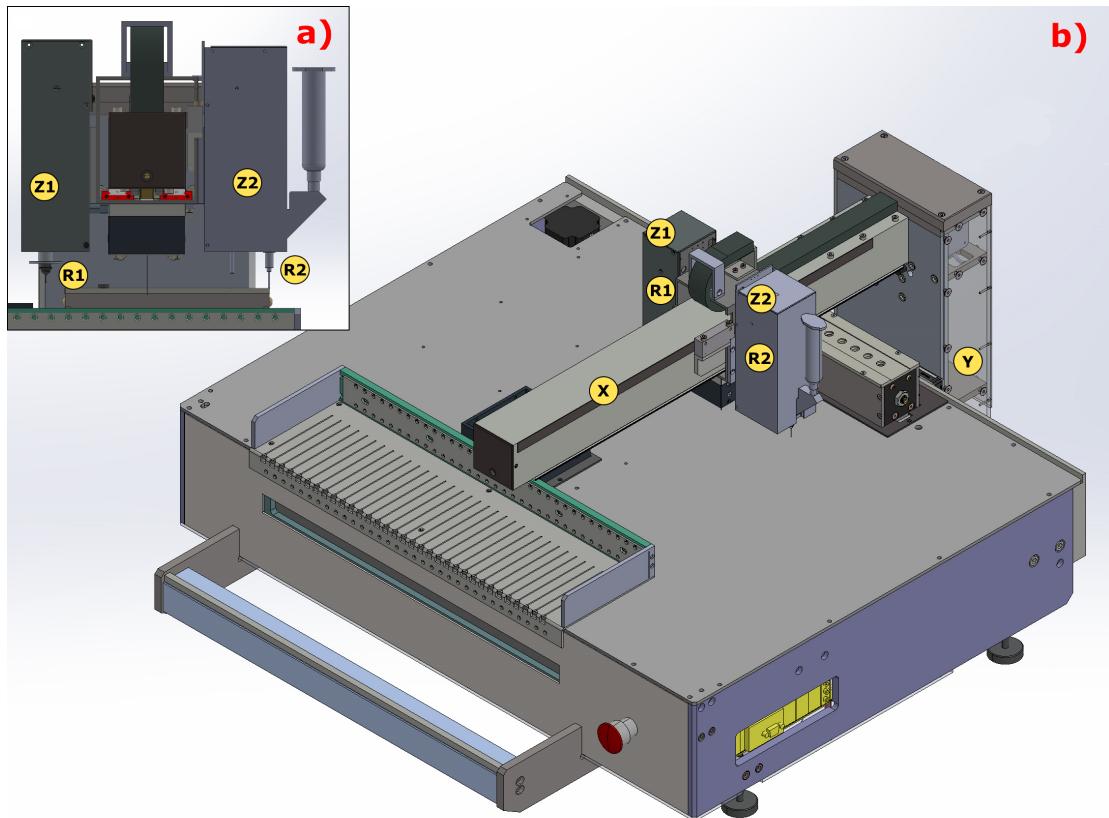
Tabela 4.2: Opis osi w automacie P30

Źródło: Na podstawie [33]

| Oś | Funkcja |
|----|--|
| Y | Ruch ramienia automatu „Prawo- Lewo” |
| X | Ruch ramienia automatu „w przód- w tył” |
| Z1 | Ruch głowicy układającej „Góra- dół” |
| R1 | Obrót głowicy układającej CW i CCW (Clockwise/ Counter clockwise) |
| Z2 | Ruch głowicy dyspensującej „Góra- dół” |
| R2 | Dyspensja medium z głowicy dyspensującej |

Fakt zastosowania tak dużej ilości sterowników znacząco podwyższa koszt jednostkowy wyprodukowania maszyny. Chcąc użyć w napędach poszczególnych osi

własnego kontrolera, musi mieć on możliwość sterowania ostatecznie wszystkimi tymi silnikami elektrycznymi, co obecnie sterowniki Technosoft'u.



Rysunek 4.2: Model 3D maszyny P30 typu Pick and Place z wbudowanym dysploterem z oznaczonymi osiami, a) widok lufy i głowic od przodu maszyny, b) Cała maszyna w rzucie ortogonalnym

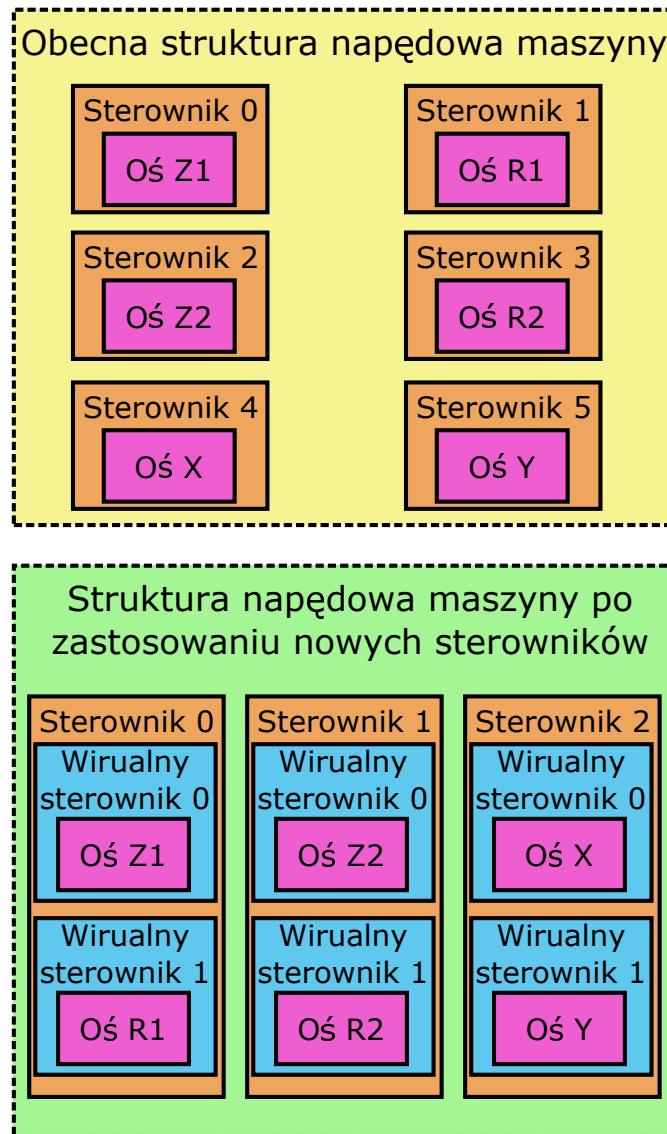
Źródło: Na podstawie [35]

Dodatkowo, sterownikowi zostało postawione jeszcze jedno wymaganie, którego nie spełnia żaden z prezentowanych kontrolerów dostępnych na rynku: praca z dwoma dowolnymi silnikami równolegle (patrz sekcja 4.3). Ta cecha sprawia, że gotowe rozwiązania pozostają daleko w tyle pod względem funkcjonalnym. Koszt jednostkowy maszyny obniży się nie tylko przez to, że każdy ze sterowników zostanie zastąpiony własnym, ale dodatkowo dlatego, że będzie trzeba zastosować dwa razy mniej kontrolerów.

Należy zaznaczyć że krótkoterminowo i w przypadku niewielu wyprodukowanych maszyn, projektowanie własnego kontrolera nie jest opłacalne, gdyż koszt projektu będzie przekraczać koszt zastosowania gotowego rozwiązania. Z perspektywy czasu jednak sytuacja się odwraca. Cena za komponenty elektroniczne i wyprodukowanie PCB w zewnętrznej firmie jest niewielka przy zamówieniach w dużych ilościach. Głównym kosztem jest praca wykonana przez projektanta PCB i programistę w celu opracowania rozwiązania. Zalet tego typu podejścia jest jeszcze więcej i zostaną one w większych szczegółach przytoczone w dalszych rozdziałach i sekcjach.

4.2. Zgodność protokołu komunikacyjnego

Budowany sterownik musi zastąpić kontroler Technosoft' zachowując przy tym możliwie wysoki poziom zgodności. W przypadku nie spełnienia tego wymagania, pozostałe moduły (urządzenia) będące częścią całego automatu obarczone byłyby potrzebą aktualizacji ich oprogramowania lub nawet sprzętu, w celu możliwości komunikacji z nowymi sterownikami.



Rysunek 4.3: Struktura zastosowanych sterowników w maszynie P30 obecnie i w przyszłości po pełnej implementacji opracowywanego rozwiązania

Źródło: Na podstawie [33]

To wprowadziłoby do przedsięwzięcia bardzo wysokie koszty związane z dodatkowym czasem pracy inżynierów i potrzebą zamówienia nowych przerobionych modułów. Dlatego też warstwa odpowiedzialna za komunikację musi odpowiadać tej zastosowanej przez firmę Technosoft. Specyfikacja do tego protokołu (*TML CAN-Technosoft Motion Language*) jest udostępniana przez producenta aby umożliwić użytkownikom ich produktów komunikację z nimi. W tym wypadku zostanie ona za-

stosowana w celu ograniczenia czasu potrzebnego na wdrożenie nowego, własnego rozwiązania.

4.3. Wirtualizacja osi

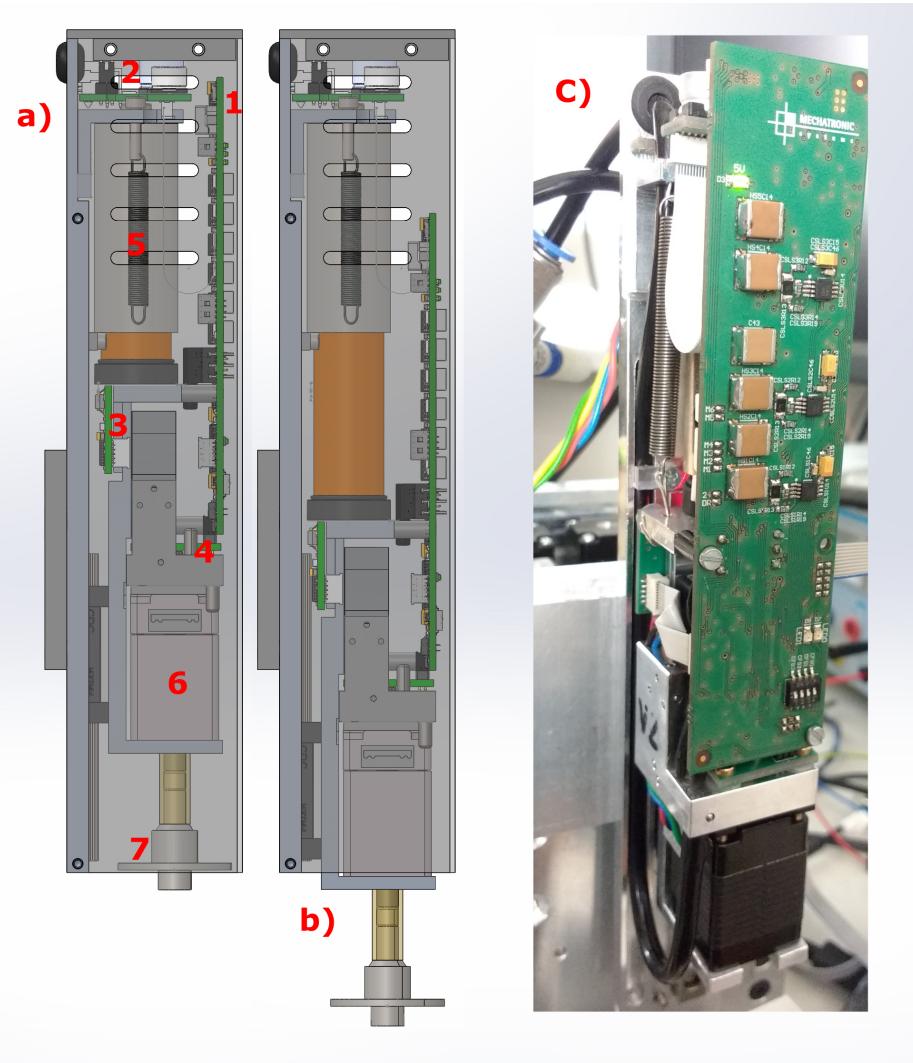
Chcąc zastosować kontroler, który ma możliwość sterowania dwoma silnikami w miejsce takiego, który mógł pracować tylko z jednym, trzeba pójść na pewne ustępstwa. Mianowicie, pozostałe urządzenia muszą dalej postrzegać sterownik jako dwa osobne urządzenia. W tym celu kontroler musi przedstawiać się na magistrali komunikacyjnej jako dwa osobne urządzenia, pomimo tego, że posiada tylko jedno oprogramowanie i są do niego podłączone dwa silniki. Taki koncept wydaje się być rozwiązaniem typu *work-around* (na około), ale ma także swoje plusy z punktu widzenia architektury oprogramowania kontrolera (patrz rozdział 6).

Na chwilę obecną, nowy sterownik jest ciągle w fazie projektowania. W jego pierwotnej formie jest wdrażany do zoptymalizowanej mechanicznie głowicy układającej w automatach Pick and Place. W dalszych rozdziałach i sekcjach, kontroler będzie opisywany w odniesieniu właśnie do tej implementacji. Pomimo tego, że układ może przyjmować nieco zmienioną sprzętową formę w zależności zastosowania w różnych aplikacjach, to jego oprogramowanie i on sam dalej pozostaje uniwersalny. W celu optymalizacji zajmowanego przez układ miejsca, hardware może być przystosowany np. tylko do silników krokowych lub tylko DC. Nie oznacza to jednak, że potrzebne są jakiekolwiek zmiany w oprogramowaniu, lub duży wysiłek w przeprojektowaniu płytki PCB.

Dalsze rozdziały przedstawiają rozwiązania techniczne sprzętowe i programowe w formie szczegółowej.

5. Projekt uniwersalnego sterownika

Pierwszym zastosowaniem projektowanego sterownika jest nowa, zoptymalizowana pod względem mechanicznym głowica układająca do automatów typu Pick and Place w firmie Mechatronic Systems. Na najbliższych stronach znajduje się opis poszczególnych modułów tej głowicy pozwalający zrozumieć jej zasadę działania, ze szczególnym uwzględnieniem części elektrycznej, gdyż to ona stanowi wkład autora w projekt tej głowicy.



Rysunek 5.1: Prototyp głowicy układającej z zastosowaniem projektowanego sterownika: a) Widok modelu 3D głowicy od boku z wsuniętą pinolą, b) ten sam model z pinolą wysuniętą, c) rzeczywiste zdjęcie głowicy. Wymiary przy wsuniętej pinoli 34 x 38 x 184 mm

Źródło: Na podstawie [35]

Rysunek 5.1 prezentuje sterownik zamontowany w miejscu pracy. Poszczególne elementy urządzenia to:

1. Płytkę sterownika główną.
2. Płytkę sterownika pośrednią zasilającą.
3. Płytkę sterownika z układem enkodera dla silnika w osi Z.

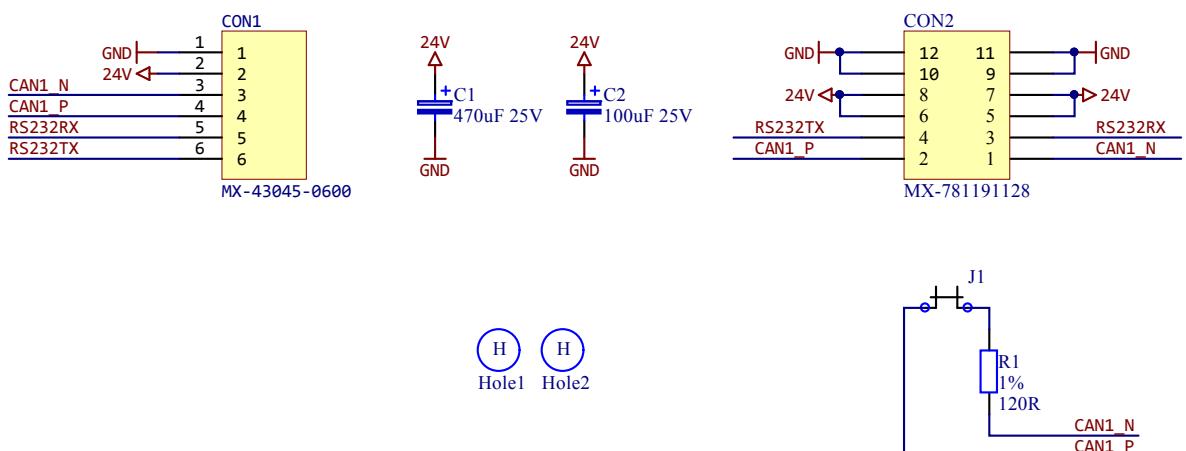
4. Płytką sterownika z układem enkodera dla silnika w osi R.
5. Silnik liniowy VCM odpowiedzialny za ruch posuwisto-zwrotny w osi Z.
6. Silnik hybrydowy krokowy odpowiedzialny za obrót w osi R.
7. Pinola głowicy służąca do podnoszenia i odkładania komponentów SMD.

Ze względu na konstrukcję mechaniczną głowicy, w celu wykorzystania jak największej przestrzeni na PCB, kontroler składa się z czterech płyt. Jak widać na rysunku 5.1 b), elementy 1, 3, 4, 6 i 7 znajdują się w części ruchomej głowicy, a 2 i 5 w części nieruchomej (korpus silnika VCM jest ruchomy, a jego cylinder statyczny – dokładna budowa tego silnika przedstawiona jest na rysunku 1.5). Część ruchoma porusza się po prowadnicy, która zapewnia równoległy względem cylindra ruch korpusu. Zasilanie i sygnały magistrali komunikacyjnej do części ruchomej doprowadzane są elastyczną tasiemką typu *Flex* widoczną na zdjęciu głowicy bez obudowy (5.1 c)). Silnik krokowy służy do obracania przyssanych do pinoli (podnoszenie przez podciśnięcie) komponentów.

Kolejne sekcje przedstawiają opis poszczególnych modułów znajdujących się na każdej z płyt PCB kontrolera. Wszystkie schematy ideowe (także ich części) i płytki PCB prezentowane dalej wykonane przez autora pracy w programie *Altium Designer* dla firmy Mechatronic Systems, chyba że zaznaczono inaczej. Niektóre części schematów zostały przeorganizowane w taki sposób, aby zajmowały jak najmniej miejsca w celu czytelnego wyświetlenia w tekście. Kompletny oryginalny schemat ideowy całego urządzenia znajduje się w załącznikach pracy.

5.1. Płytką zasilającą

Pierwsza z opisywanych płyt (widoczna na rysunku 5.1) jest oznaczona numerem 2. Pełni ona funkcję pośredniczącą między płytą główną sterownika a przewodami doprowadzającymi sygnały sterujące i zasilanie.



Rysunek 5.2: Schemat ideowy płytka zasilającej w głowicy układającej

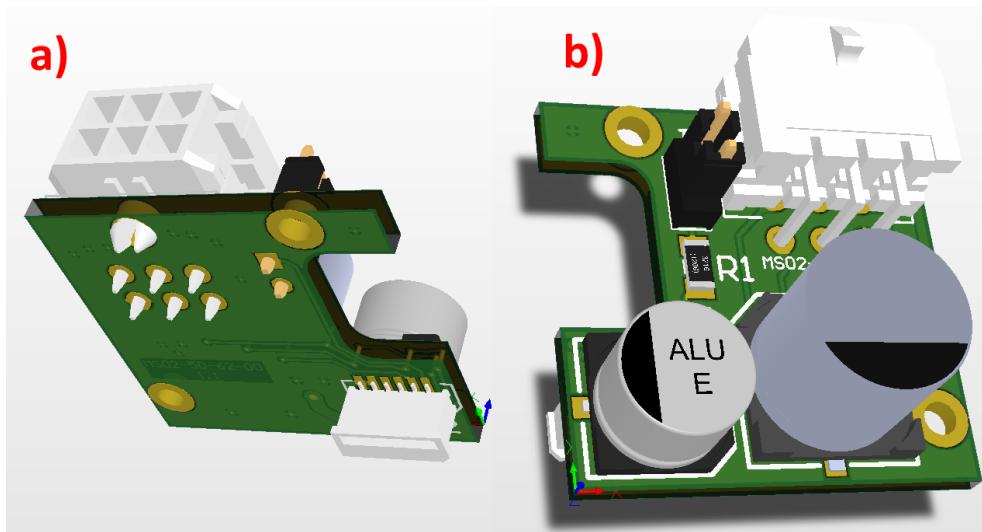
Źródło: Na podstawie [32]

Poprzez przewód do złącza *CON1* do głowicy doprowadzane są:

- zasilanie ze źródła napięcia o amplitudzie 24 V prądu stałego,
- para różnicowa magistrali CAN,

- sygnał odbiorczy i nadawczy magistrali RS232.

Wszystkie powyższe sygnały są następnie przekazywane na złącze *CON2*, które poprzez tasiemkę typu *Premo-Flex* o rastrze 0.5 mm (firma Molex) przewodzi je dalej do płytki głównej. Wytrzymałość prądowa pojedynczej żyły takiego złącza to tylko 0.5 A, dlatego sygnały zasilające używają ich aż 8 (po 4 dla zasilania i dla masy).



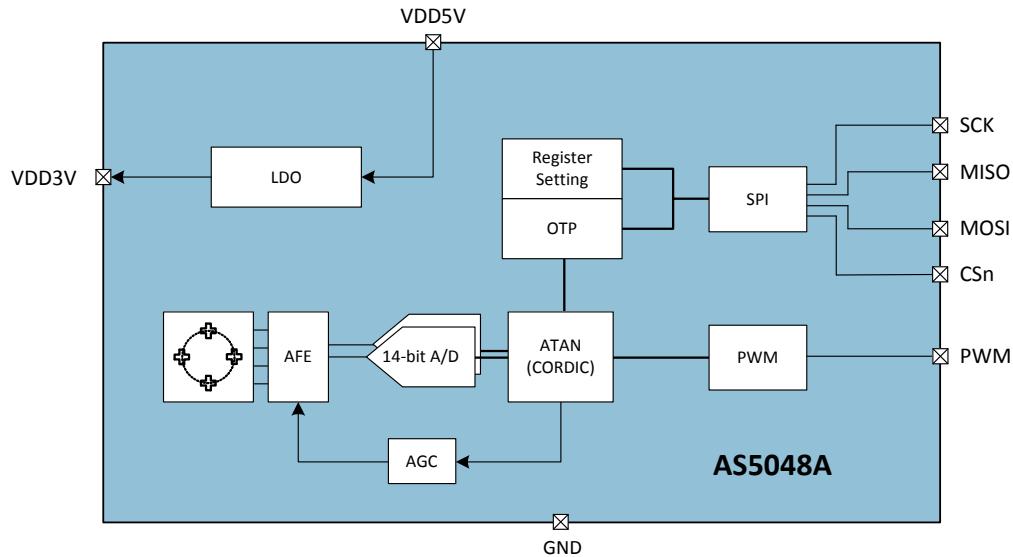
Rysunek 5.3: Widok ortogonalny modelu 3D płytka zasilającej, a) z widokiem warstwy dolnej, b) z widokiem warstwy górnej

Źródło: Na podstawie [32]

Kolejnymi elementami PCB są dwa kondensatory elektrolityczne o dużej pojemności. Ostatni komponent to zworka zwierająca rezistor terminujący na magistrali CAN. W zależności od topologii magistrali można ją załączyć lub zostawić rozwartą. Terminator został umieszczony dosyć daleko od układu sterownika CAN. Odległość ta jest jednak wystarczająco niska dla poprawnego działania układu.

5.2. Płytki enkoderów magnetycznych

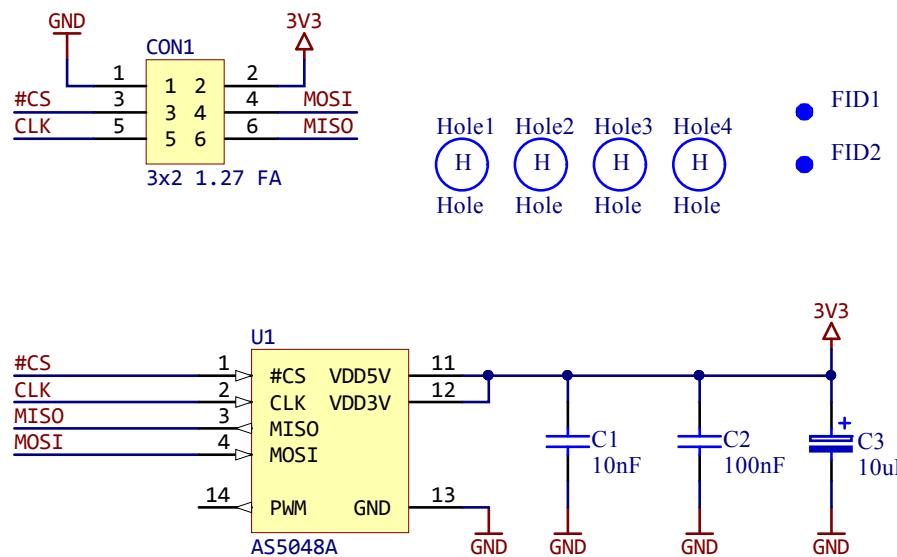
Płytki z enkoderem dla osi R jest zamontowana za silnikiem krokowym (numer 4 na rysunku 5.1). Jako enkoder magnetyczny został zastosowany układ scalony AS5048A firmy AMS. Rozwiążanie to przypomina te wykorzystane przez Tropical Labs w swoim sterowniku Mechaduino (3.2.2). Należy jednak wspomnieć, że koncepcja została zastosowana przed opublikowaniem wpisu o ich sterowniku na stronie hackaday.io i nie jest na nim wzorowany, ani nim inspirowany. AS5048A jest enkoderem absolutnym, który dostarcza informację o aktualnej pozycji kątowej między innymi poprzez interfejs SPI (*Serial Peripheral Interface*). Używając go można uzyskać pozycję z dokładnością do 0.05° po zastosowaniu algorytmów uśredniających [24]. Więcej informacji o zastosowanych w sterowniku algorytmach uśredniających dla tego enkodera jest zawarte w sekcji 6.3. Obracając magnesem w sąsiedztwie układu scalonego, zmienne pole magnetyczne wpływa na matrycę czujników Hall'a.



Rysunek 5.4: Schemat blokowy układu scalonego AS5048A

Źródło: Na podstawie [24]

Przetwornik analogowo cyfrowy zawarty w urządzeniu przetwarza analogową informację w formie zmiennego natężenia pola magnetycznego na informację cyfrową o aktualnej pozycji kątowej.



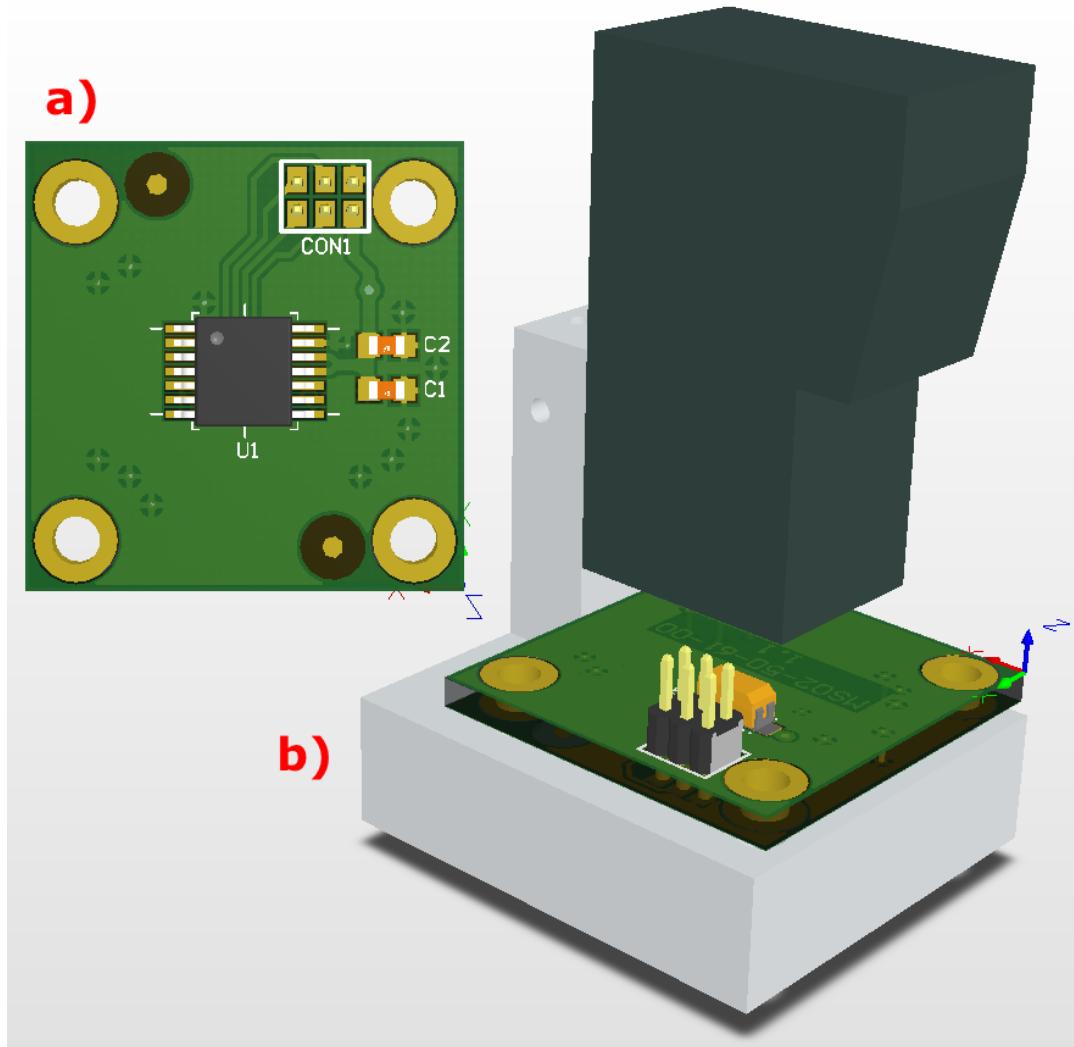
Rysunek 5.5: Schemat ideowy płytka z enkoderem magnetycznym dla osi R w głowicy układającej

Źródło: Na podstawie [32]

Sygnały magistrali SPI są wyprowadzone na pionowe złącze CON1 o rastrze 1.27 mm, przez które dalej idą na płytę główną kontrolera. Elementy oznaczone jako FID1 i FID2 to punkty referencyjne dla górnej kamery automatu Pick and Place (tzw. *Fiducial'e*). Są one widoczne na płytce PCB z rys. 5.6 a), w lewym górnym i prawym dolnym rogu. Ten sam rysunek (obiekt b) przedstawia płytę enkodera zamocowaną na elemencie aluminiowym frezowanym, który z kolei jest przykręcony

do silnika krokowego. Duży czarny element to mikro zawór doprowadzający podciśnienie, sterowany z głównej płytki kontrolera (opis w sekcji 5.4).

Oś poprzeczna układu scalonego musi się idealnie pokrywać z osią obracającego się magnesu w celu jak najlepszego odczytu natężenia pola przez enkoder.



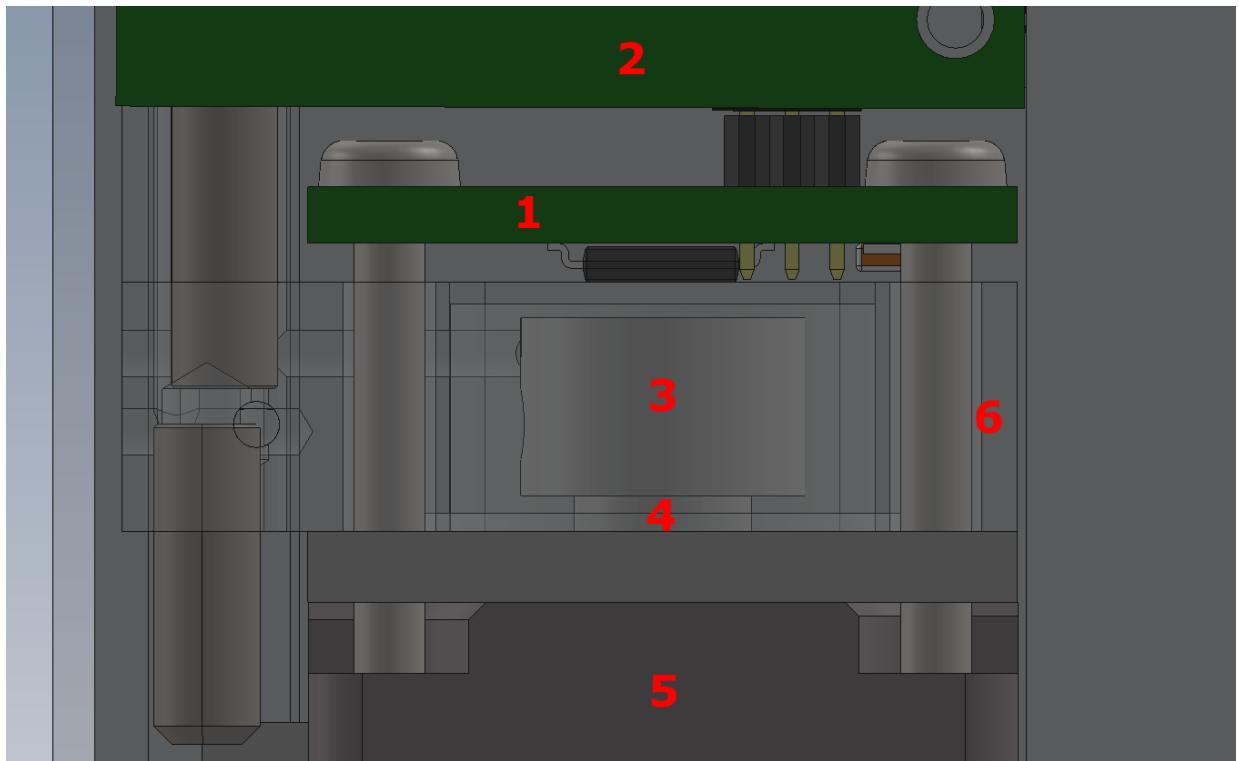
Rysunek 5.6: Widok płytka enkodera magnetycznego- a) dolna warstwa PCB, na której znajduje się AS5048A wraz z kondensatorami blokującymi, b) rzut ortogonalny płytka zamocowanej na elemencie wspólnym dla niej i silnika

Źródło: Na podstawie [32]

Na rysunku 5.7 widać dokładnie co znajduje się w aluminiowej kostce do której zamocowana jest płytka:

1. Płytkę z enkoderem magnetycznym AS5048A.
2. Płytkę sterownika główną, połączoną przez CON1 (5.5).
3. Magnes trwały namagnesowany poprzecznie z otworem w środku, zamocowany na końcu wału silnika.
4. Przelotowy wał silnika krokowego.
5. Silnik krokowy hybrydowy.

6. Element aluminiowy (na modelu przezroczysty) łączący wymienione komponenty.



Rysunek 5.7: Płytkę z enkoderem magnetycznym zamocowana nad obracającym się, przelotowym magnesem trwałym, który z kolei jest zamocowany na przelotowym wale silnika krokowego

Źródło: Na podstawie [35]

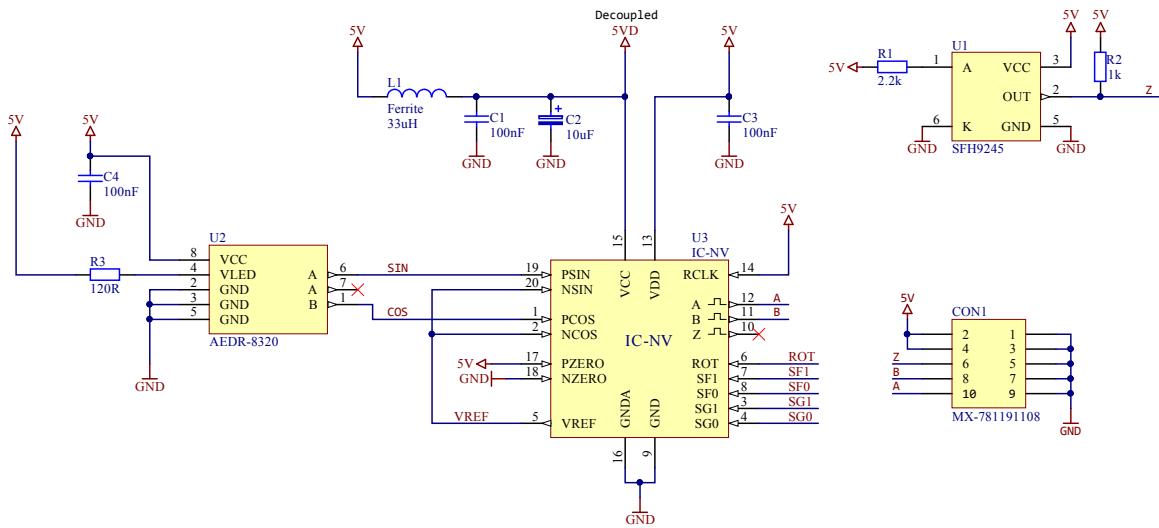
Do elementu łączącego jest także doprowadzone podciśnienie poprzez giętki przewód pneumatyczny – w ten sposób jest możliwe podnoszenie komponentów elektronicznych przez głowicę (przysysając je).

5.3. Płytkę enkodera optycznego

Podobnie jak silnik krokowy współpracuje z enkoderem magnetycznym w osi R, tak samo silnik VCM działa z enkoderem optycznym w osi Z i także posiada osobną płytka PCB dla obwodu tego enkodera. W odróżnieniu do poprzedniej PCB jednak, ta wymaga o wiele więcej komponentów do poprawnej pracy. Pozycja nie jest już podawana absolutnie poprzez magistralę szeregową tylko musi być odczytana jako zmieniający się sygnał analogowy, który następnie poddawany jest interpolacji. Potem trafia do głównej płytki kontrolera w postaci 2 zmieniających się sygnałów zero-jedynkowych (sygnał kwadraturowy). W zależności od tego które zbocze jest wiodące, procesor może określić kierunek przesuwania się enkodera względem liniału.

Rysunek 5.8 prezentuje większość schematu ideowego płytki enkodera. Elemenem odczytyującym pozycję jest AEDR-8320 (obiekt U2 na schemacie). Jest to en-

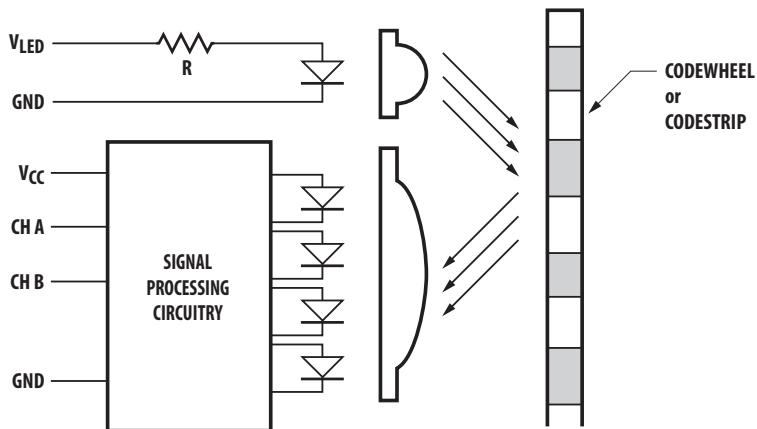
koder optyczny w małej obudowie SMD, który generuje impulsy świetlne, które odbijając się od przeszkód są przez niego wykrywane przy użyciu matrycy fotodiod.



Rysunek 5.8: Część schematu ideowego płytka enkodera liniowego osi Z

Źródło: Na podstawie [32]

W celu uzyskania najlepszego efektu i rozdzielczości, należy użyć specjalnego paska z mikro wpustkami jako powierzchni odbijającej światło (liniału). Zasada działania jest zobrazowana na rysunku 5.9.

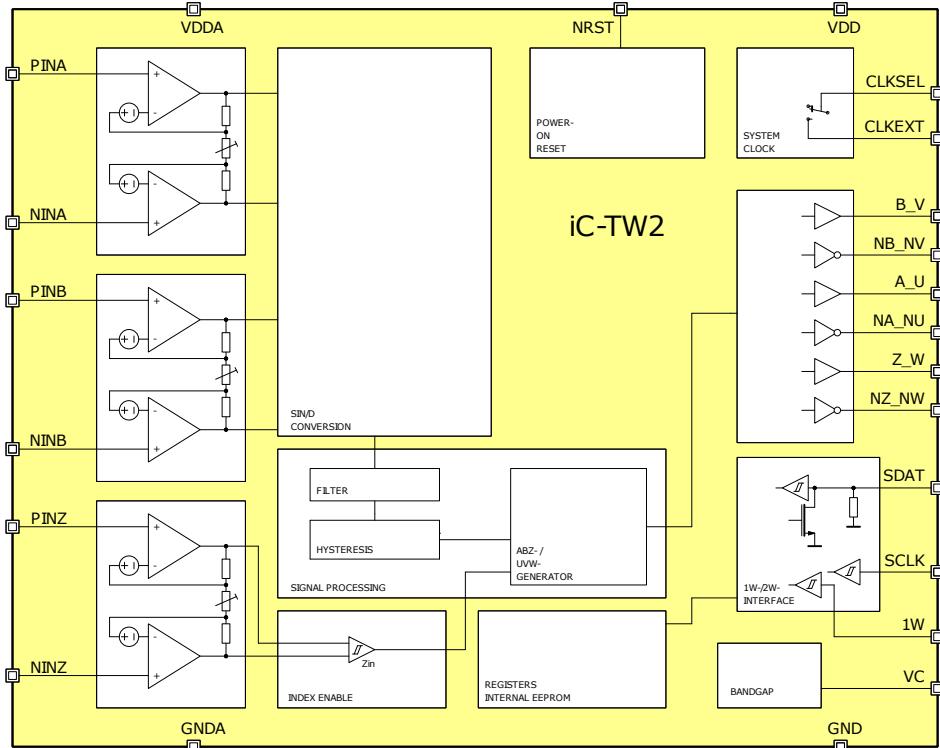


Rysunek 5.9: Schemat blokowy i zasada działania układu scalonego AEDR-8320 firmy Avago Technologies

Źródło: Na podstawie [22]

Sygnały analogowe o kształcie sinusoidy (sygnał wyjściowy A) i cosinusoidy (sygnał wyjściowy B) są podłączone do wejść interpolatora IC-TW (rys 5.10). Jego zadaniem jest zamiana tych sygnałów na postać cyfrową, która może być odczytana przez mikrokontroler sterujący na głównej płytce urządzenia (połączenie przez złącze CON1). Poziom interpolacji i wzmacnianie ustala się zwierając odpowiednia wyprowadzenia chip'a do masy lub potencjału zasilającego. Na PCB enkodera

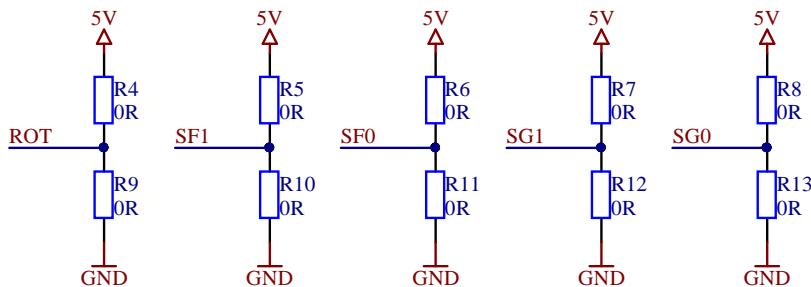
komponenty zostały umieszczone w taki sposób, aby konfigurację można było zmodyfikować przy niewielkim nakładzie pracy - za pomocą zworek, które widoczne są na drugiej części schematu ideowego 5.8 (rezystorów $0\ \Omega$).



Rysunek 5.10: Schemat blokowy 8-bitowego interpolatora IC-TW2 firmy IC-HAUS

Źródło: Na podstawie [23]

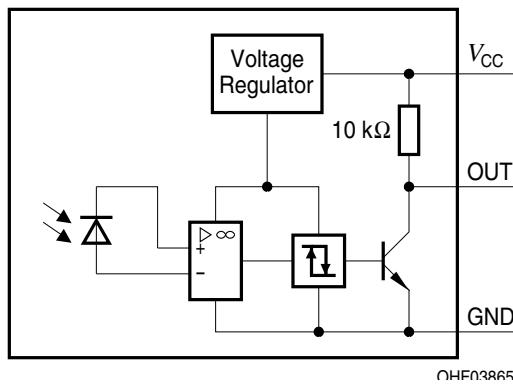
W danym momencie przylutowany jest tylko jeden z rezystorów dla danej linii konfiguracyjnej (zwierający do GND lub 5V).



Rysunek 5.11: Część schematu ideowego płytka enkodera liniowego osi Z, prezentująca zwinki konfiguracyjne

Źródło: Na podstawie [32]

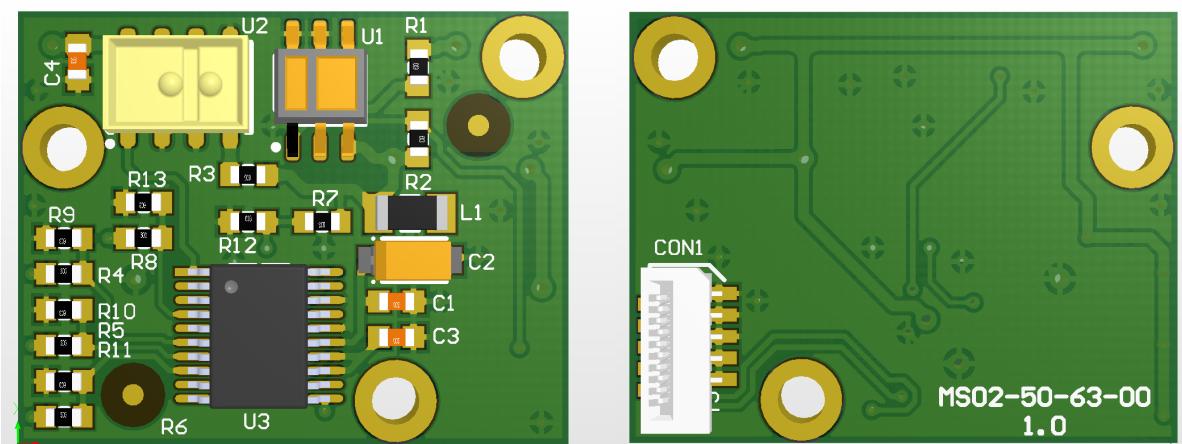
Ostatnim elementem pozostałym do omówienia na schemacie jest U1. SFH9245 to układ elektroniczny działający w podobny sposób do AEDR-8313 (5.9), który jest jednak o wiele prostszy i tańszy. Składa się on z diody emitującej światło podczerwone i z fototranzystora z przerzutnikiem Shmitt'a (5.12). Światło zamiast odbijać się od liniału, odbija się od wydrukowanego na drukarce laserowej paska, przyklejonego koło liniału. Pasek składa się z dwóch „stref” – czarnej i białej.



Rysunek 5.12: Schemat blokowy układu scalonego SFH9245 firmy Osram

Źródło: Na podstawie [21]

Kiedy układ znajduje się nad czarną strefą, światło nie odbija się dobrze i na wyjściu OUT układu jest stan wysoki (wyjście typu *Open-Collector*). W strefie białej światło odbija się bardzo dobrze i stan OUT zmienia się na niski. Mechanizm ten jest używany w głowicy do odnajdowania przez silnik liniowy VCM pozycji odniesienia (*Homing position*). Jest to niezbędne, dlatego że skonstruowany enkoder jest liniowy, a nie absolutny.



Rysunek 5.13: Model 3D płytki drukowanej enkodera liniowego silnika VCM. Po lewej warstwa góra, po prawej dolna

Źródło: Na podstawie [32]

Możliwa do uzyskania rozdzielcość w prezentowanym układzie zależy głównie od dwóch czynników: gęstości umiejscowienia rowków na liniale i możliwości interpolatora. W tym przypadku rozdzielcość ogranicza układ IC-TW2, którego maksymalny poziom interpolacji to x64. Przy zastosowaniu liniału 180 LPI (*Lines per inch*) daje to rozdzielcość dokładnie 2.20486 mikronów.

5.4. Płytki główna sterownika

Czwarta opisywana płytka drukowana cechuje się znacznie większym stopniem komplikacji i gęstości ułożenia elementów. Najważniejsze dane liczbowe płytki:

- wymiary: 109.1 x 25.6 mm,

- końcowa grubość laminatu: 1.55 mm,
- ilość warstw: 4,
- ilość przelotek: 411,
- liczba komponentów na warstwie górnej: 145,
- liczba komponentów na warstwie dolnej: 34.

Kształt i rozmiar PCB został wybrany tak, aby wykorzystać maksymalnie całą dostępną przestrzeń w obudowie głowicy. Rysunek 5.14 prezentuje poglądowe zdjęcie płytki niezamontowanej w urządzeniu.



Rysunek 5.14: Zdjęcie głównej płytki sterującej głowicy układającej (warstwa górnna)

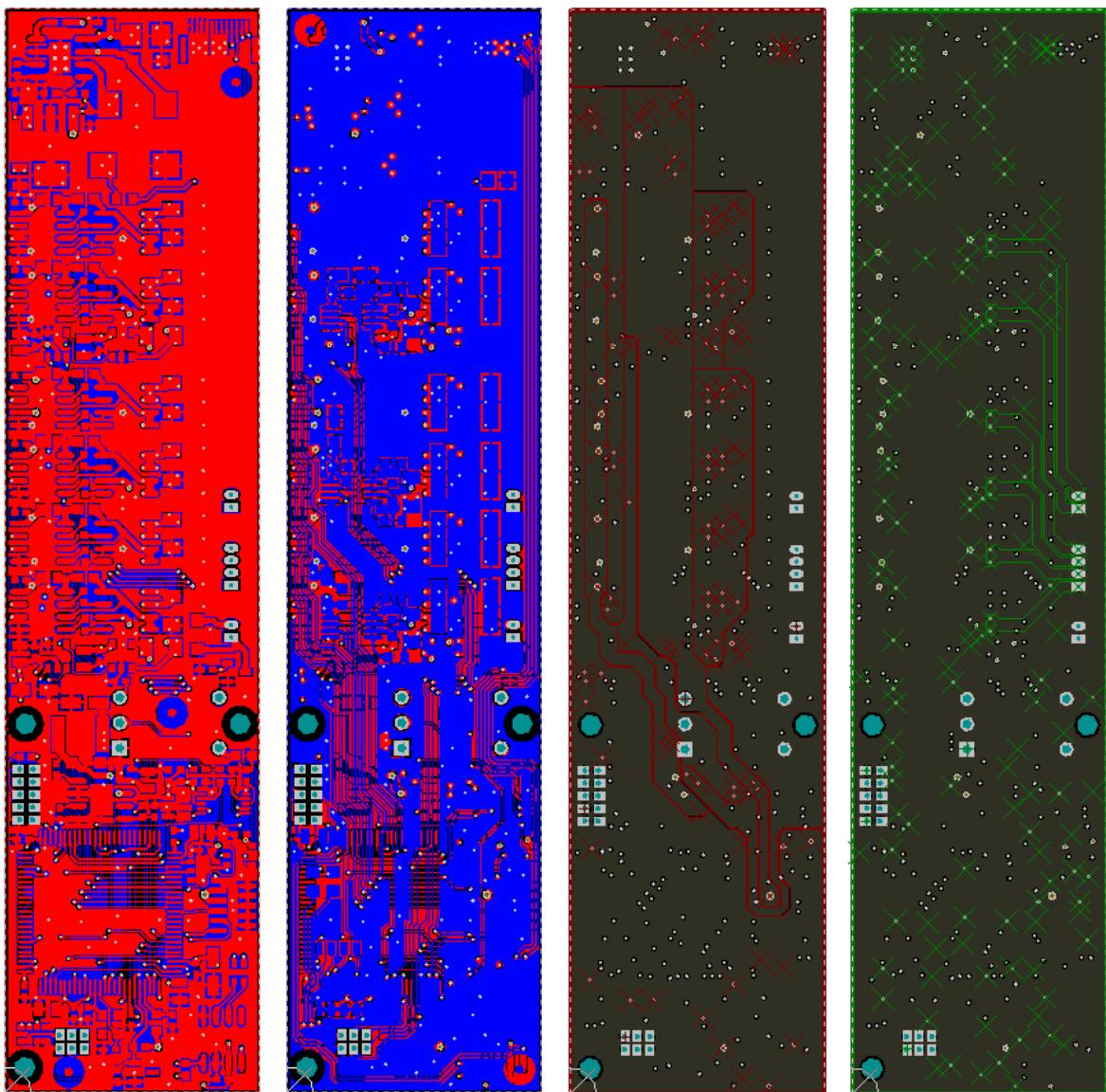
Źródło: Na podstawie [33]

Dwie wewnętrzne warstwy płytki to tzw. *Plane Layers* i służą głównie do doprowadzania zasilania. Na warstwie górnej (*Component layer*) jest ułożonych ok. 80% komponentów. Pozostałe znajdują się na warstwie dolnej (*Solder side*). W momencie kiedy wszystkie komponenty elektroniczne znajdują się tylko po jednej stronie płytki, znacząco upraszcza się jej technologia produkcji. W tym wypadku, aby zamontować płytke przy pomocy automatu typu Pick and Place, należy poza pastą lutowniczą użyć też kleju (dozowanego np. z dyspensera), aby przykleić komponenty po jednej stronie płytki. Dzięki temu, kiedy ta trafi do pieca rozpływowego w celu zlutowania, elementy po dolnej stronie nie spadają. Taka forma projektowania ma jeszcze jedną zaletę- można użyć jednej z warstw tylko do prowadzenia ścieżek, które inaczej byłoby trudno doprowadzić w niektóre miejsca. W tym wypadku jednak trzeba było pójść na kompromis; ścieżki (prowadzi lokalnie „na krzyż”) i komponenty musiały się znaleźć po obu stronach. Ścieżki sygnałowe nie są prowadzone na wewnętrznych warstwach, aby uniknąć pojemności, które mogły by się pojawiać pomiędzy nimi.

Zastosowanie elementów THD (przewlekane– *Through Hole Device*) zostało ograniczone do minimum w celu optymalizacji czasu montażu. Są to:

- złącze łączące płytke główną z płytą enkodera magnetycznego (sekcja 5.2),
- złącze programatora (rys. 5.22),
- czujnik podciśnienia (sekcja 5.4.4).

Na górnej warstwie pomiędzy złączami znajduje się dużo wolnej przestrzeni, na której nie ma umiejscowionych żadnych komponentów. Jest to obszar rozbiegowy dla elastycznej tasiemki łączącej PCB z płytą zasilania (sekcja 5.1 i rysunek 5.1 c).



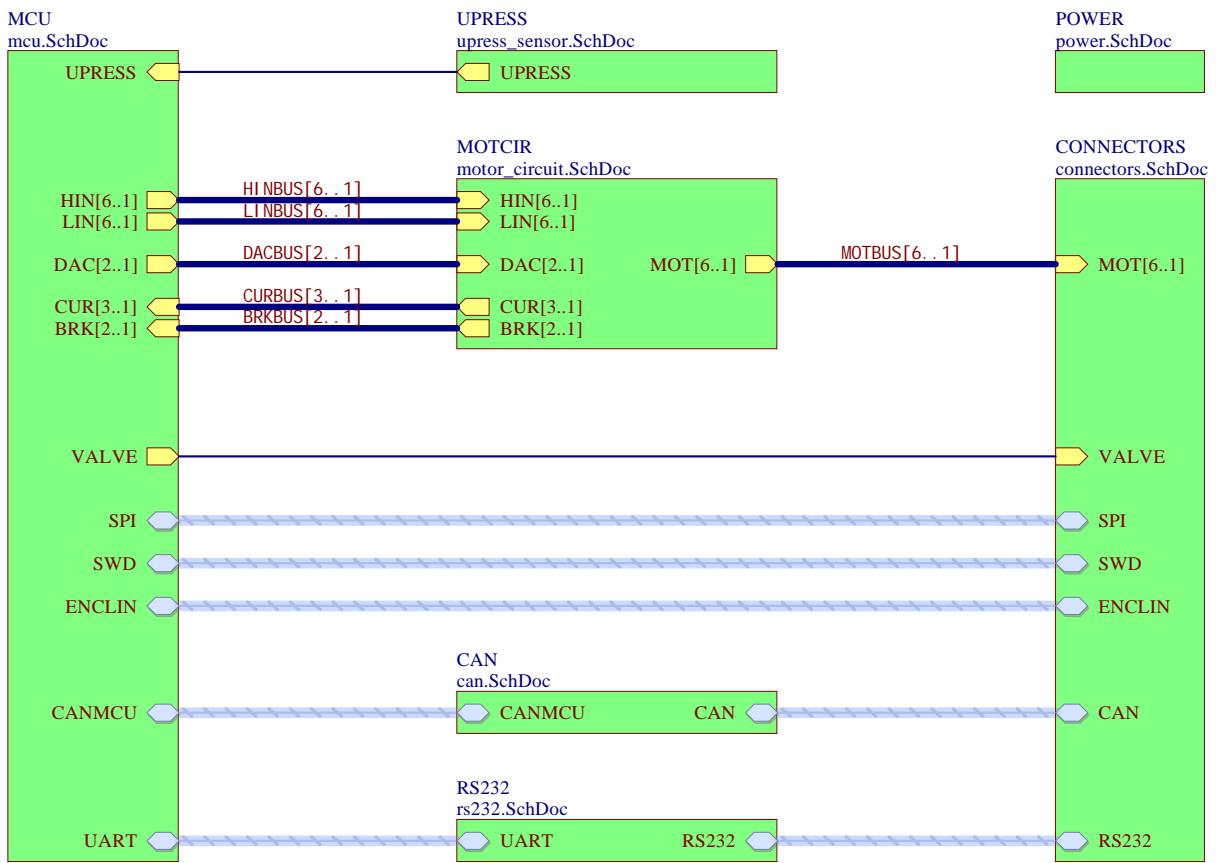
Rysunek 5.15: Poszczególne warstwy płytki głównej sterownika, od lewej góra, dolna, warstwa zasilania (*power plane*), warstwa masy (*ground plane*)

Źródło: Na podstawie [32]

W następnych sekcjach tego rozdziału zostaną szczegółowo opisane poszczególne moduły elektroniczne płytka głównej. Zostaną zaprezentowane tylko części schematów niezbędne do opisu poszczególnych funkcjonalności. Kompletny schemat płytki znajduje się w załączniku A.

Struktura schematu ideowego ma charakter obiektowy- podobnie jak w programowaniu wysokopoziomowym, zdeklarowane są klasy i tworzone są ich egzemplarze. Pozwala to na separację modułów niskiego poziomu i powielanie ich w prosty

sposób tylko na PCB, co przyspiesza projektowanie i zwiększa czytelność schematów.



Rysunek 5.16: Schemat ideowy najwyższego poziomu łączący ze sobą poszczególne moduły w projekcie płytki, w programie Altium Designer

Źródło: Na podstawie [32]

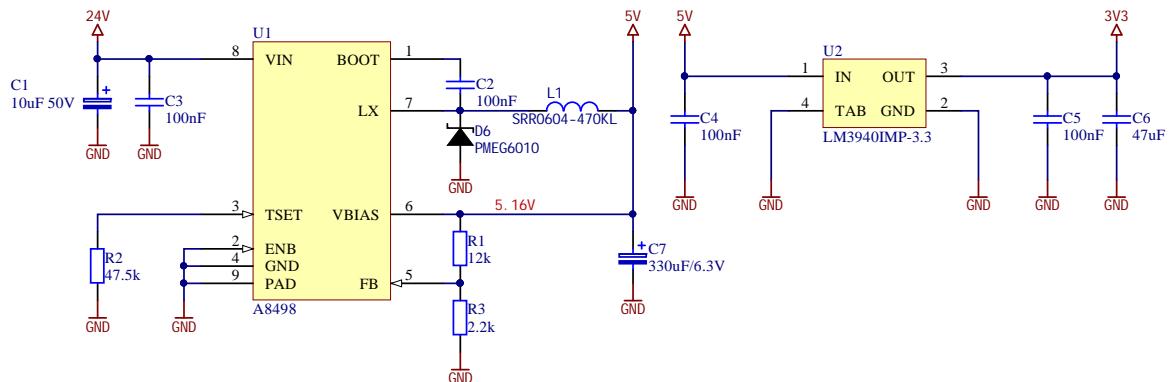
5.4.1. Zasilanie

W układzie sterownika potrzebne jest wiele poziomów napięć ze względu na różnorodność zastosowanych układów scalonych. Aby je wszystkie uzyskać zostały zastosowane różne rozwiązania, zarówno bazujące na regulatorach impulsowych jak i liniowych. Wszystkie tory napięciowe są poprowadzone na jednej z warstw wewnętrznych PCB.

Do głowicy jest doprowadzone tylko jedno napięcie, z którego uzyskiwane są pozostałe, niższe poziomy. To napięcie 24 V jest także używane do zasilania uzwojeń obu silników. „Logika” urządzenia jest na poziomie 3.3 V i z tego względu większość elementów na płytce zasilana jest właśnie tym napięciem. Część układów natomiast wymaga do działania napięcia 5 V. W celu efektywnego uzyskania obu napięć został użyty obwód z rys. 5.17.

W pierwszej kolejności napięcie główne trafia na przetwornicę A8498 (Allegro Microsystems) typu *Buck-converter*, która redukuje je do 5V na wyjściu. Układ jest w stanie dostarczyć prąd o wartości 3A, ale ze względu na niewielką przestrzeń

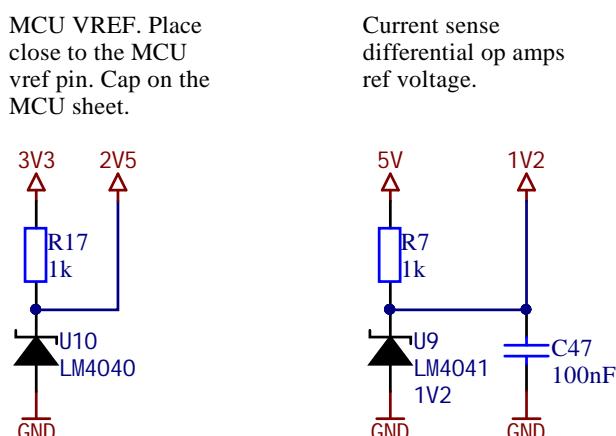
dostępna na płytce elementy mocy przetwornicy (Cewka L1 i dioda D6) zostały dobrane tak, aby być w stanie dostarczyć tylko 400 mA prądu.



Rysunek 5.17: Część schematu zasilania prezentująca sposób uzyskania napięć 3.3 i 5 V na płytce sterownika

Źródło: Na podstawie [32]

Natężenie to jest w zupełności wystarczające do zasilania układów logicznych i analogowych sterownika, także tych pracujących przy napięciu 3.3 V – do wyjścia przetwornicy podłączony jest liniowy stabilizator napięcia LM3940 (Texas Instruments).



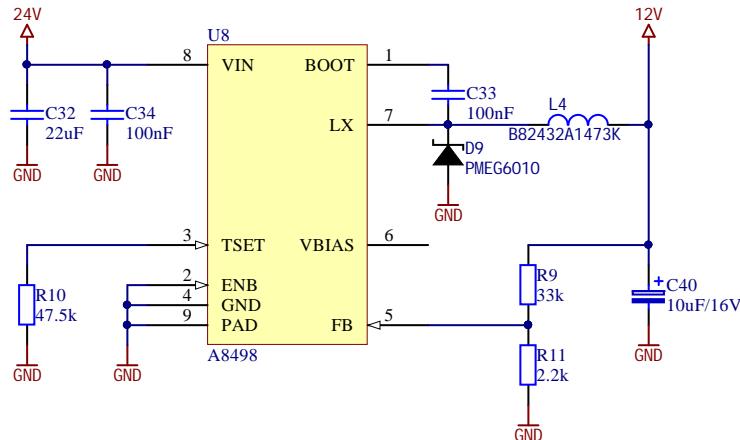
Rysunek 5.18: Źródła napięcia odniesienia dla obwodu analogowego sterownika

Źródło: Na podstawie [32]

Urządzenia LM4040 i LM4041 (rys. 5.18) to diody zenera pracujące jako precyzyjne układy napięcia odniesienia, których napięcie wyjściowe jest odporne na wpływ zmian temperatury otoczenia. Są używane do zasilania układów analogowych stosowanych w obwodzie sterowania silnikami. Więcej na ten temat jest zawarte w sekcji 5.4.5. Ostatnie napięcie to 12V, które jest także uzyskiwane z przetwornicy A8498, w nieco zmienionej konfiguracji (rys. 5.19).

Zmianie względem poprzedniego obwodu z tą przetwornicą uległy wartości rezystorów tworzących dzielik napięcia między wyjściem przetwornicy a jej wejściem FB (*Feedback*), oraz połączenie do wejścia VBIAS, którego na tym schemacie już

nie ma. Rezystory R9/ R11 na schemacie 5.19 i R1/ R3 na schemacie 5.17 ustalają wartość napięcia wyjściowego głowicy. Zwierając VBIAS do wyjścia można zasilić układ logiczny w środku A8498. Wtedy wewnętrzny układ zasilający logikę może być wyłączony i chip wydziela mniej ciepła. Niestety można tak zrobić tylko do napięcia 5 V. Napięcie wyjściowe przetwornicy jest potrzebne do zasilania bramek górnych tranzystorów tworzących mostki H w obwodach silników (sekcja 5.4.5). Wartość tego napięcia może wynosić od 10 do 20 V, gdyż jest to wartość V_{GS} (napięcie między bramką a źródłem) zastosowanych tranzystorów mocy. Przyłożenie powyżej 20 V do bramki tranzystora może spowodować jego uszkodzenie.



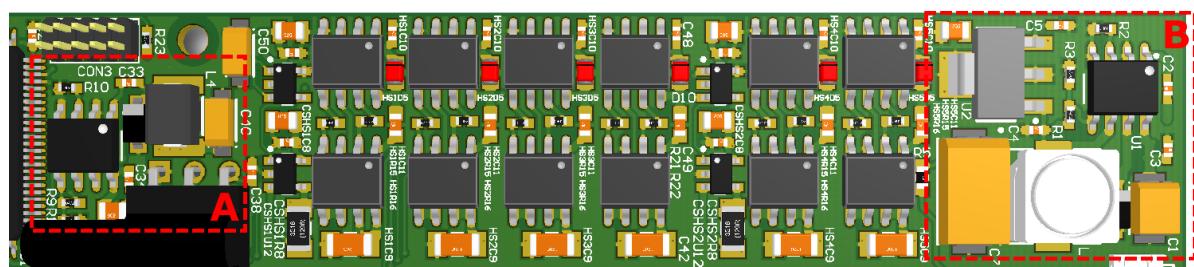
Rysunek 5.19: Przetwornica typu Buck-converter dająca na wyjściu 12V, zasilająca sterowniki typu *High-Side*

Źródło: Na podstawie [32]

Użycie liniowego stabilizatora napięcia zamiast obwodu z rys. 5.19, dającego na wyjściu poniżej 20 V spowodowało by zaoszczędzenie dużej ilości miejsca na płytce. W zamian jednak wydzielała by się na nim duża moc, przez co generowałaby więcej ciepła:

$$P = (V_{we} - V_{out})I \quad (5.1)$$

Rys. 5.20 prezentuje model 3D części płytki, na której widać oba omówione obwody zasilania. Aby maksymalnie wykorzystać dostępną przestrzeń na PCB, moduły zostały między sobą przedzielone obwodem sterowania silnikami.



5.4.2. Jednostka centralna

Głównym elementem sterującym peryferiami w urządzeniu jest 32 bitowy mikrokontroler STM32F407VGT6 z rodziny Cortex® M4 (rodzina ARM) produkowany przez firmę ST Microelectronics. Kluczowymi parametrami decydującymi o doborze układu sterującego były:

- częstotliwość taktowania,
- posiadanie jednostki zmiennoprzecinkowej (*FPU*- Floating Point Unit),
- peryferia wewnętrzne pozwalające na sterowanie dwoma silnikami krokowymi (najwięcej końcówek mocy),
- cena.

Używając wspomnianego mikrokontrolera udało się zaimplementować wszystkie dotychczasowo działające funkcjonalności i na razie nie są znane żadne przeciwwskazania, które mogłyby się pojawić w przyszłości. Nie było to jednak prostym zadaniem- urządzenie jest typowym procesorem „ogólnego użytku” i nie posiada zaawansowanych peryferiów tak bardzo przydatnych przy sterowaniu silnikami jak procesory ku temu dedykowane.

Tabela 5.1: Porównanie mikrokontrolerów STM32F407VGT6 (ST Microelectronics) i TMS320F28377S (Texas Instruments)

Źródło: Na podstawie [33]

| Mikrokontroler | STM32F407VGT6 | TMS320F28377S |
|-------------------------------------|-------------------|----------------|
| Rodzina | ARM | C28x/ CLA |
| Architektura | RISC ¹ | RISC |
| FPU | Single point | Single point |
| Maksymalna częstotliwość taktowania | 168 Mhz | 200 Mhz |
| MIPS ² | 210 | 400 |
| Pamięć flash | 1 MB | 1 MB |
| Pamięć RAM | 194 KB | 164 KB |
| Ilość rdzeni | 1 | 2 ³ |
| Cena ⁴ | \$ 9.34 | \$ 22.15 |

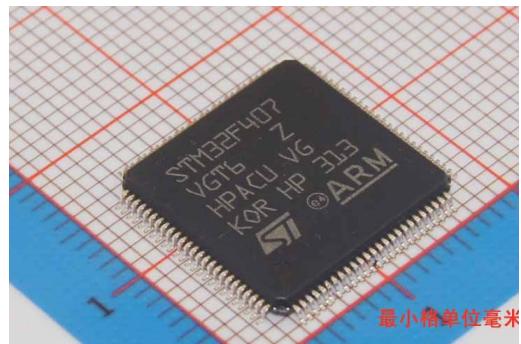
W dalszej części tego rozdziału zostały opisane w szczegółach sposoby w jaki poradzono sobie z ograniczeniami mikrokontrolera. W tabeli 5.1 przedstawiono porównanie zastosowanego układu z mikrokontrolerem dedykowanym pod sterowanie silnikami. Mimo gorszych parametrów, STM ma jedną ważną zaletę, przez którą zdecydowano o jego wyborze- jego współczynnik ceny do możliwości jakie posiada jest bardzo wysoki. Ponadto jest łatwo dostępny. Jest to spowodowane tym, że na

¹ Reduced instruction set computing

² Million Instructions Per Second

³ 1 główny + „control co-processor” do obsługi procedur sterowania

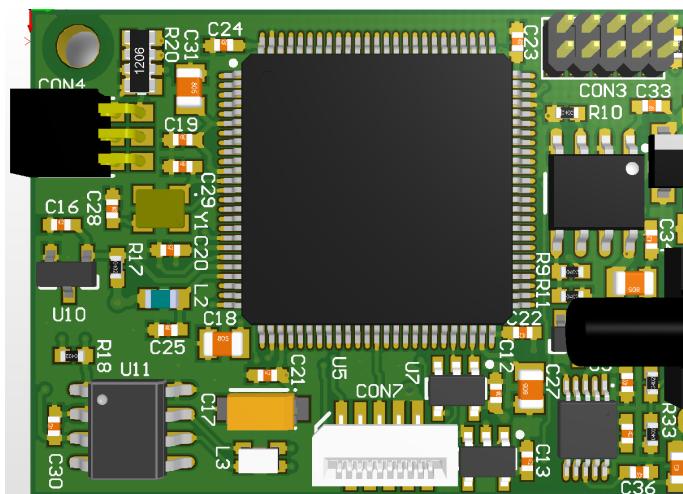
⁴ Przy zamówieniu 100 sztuk w hurtowni Digi-key



Rysunek 5.21: Mikrokontroler STM32F407VGT6 w obudowie TQFP100 o wymiarach 14 x 14 mm

Źródło: Na podstawie [20]

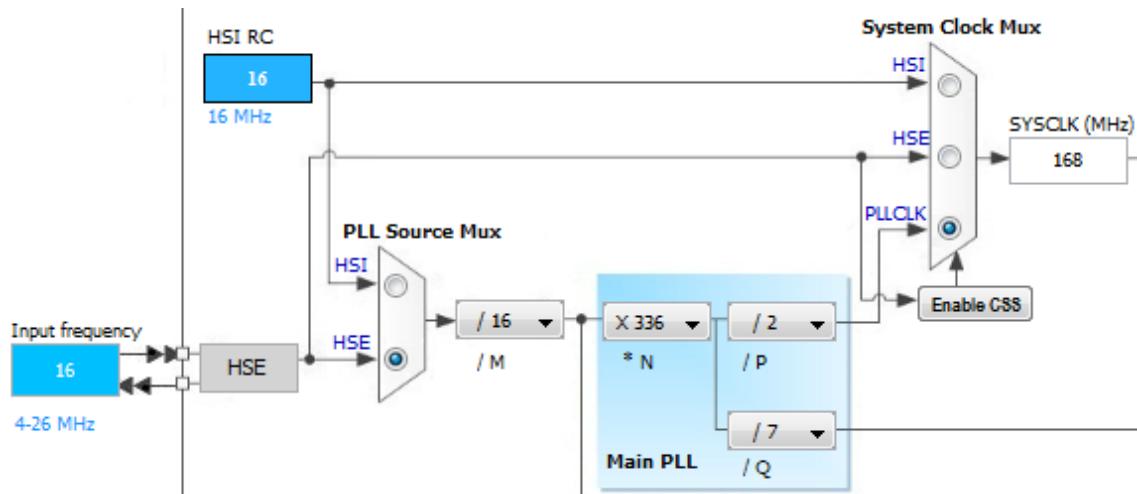
jego bazie powstała płytka ewaluacyjna *STM32-DISCOVERY*, która cieszy się bardzo wysoką popularnością wśród hobbyistów i profesjonalistów.



Rysunek 5.22: Widok 3D mikrokontrolera sterującego i otaczających go komponentów

Źródło: Na podstawie [32]

Układy STM32 są bardzo proste w programowaniu i debugowaniu dzięki dostępnym interfejsom JTAG i SWD (złącze CON3 na rys. 5.22). JTAG jest interfejsem szybszym i bardziej stabilnym od SWD, ale wymaga większej ilości połączeń. W tym projekcie nie mógł zostać zastosowany, gdyż część z pinów, które są potrzebne do jego pracy została użyta przy innych modułach. Źródłem zegarowym dla układu jest zewnętrzny mikro oscylator (Y1) o częstotliwości 16 Mhz. Używając wewnętrznych pętli PLL (*Phase Locked Loop*) uzyskiwana jest główna częstotliwość taktowania procesora 168 Mhz. Rysunek 5.23 przedstawia konfigurację wewnętrznych zegarów w mikrokontrolerze. Szybkie działanie mikrokontrolera jest parametrem kluczowym dla konstruowanego sterownika ze względu na to, że silniki są maszynami wymagającymi obsługi w czasie rzeczywistym. W celu zapewnienia filtracji zakłóceń na liniach zasilania powodowanych głównie pracą silników, każde z wejść zasilających mikrokontrolera jest odprzegnięte poprzez kondensator 100 nF (w obudowach 0402 tak jak większość komponentów pasywnych na płytce).



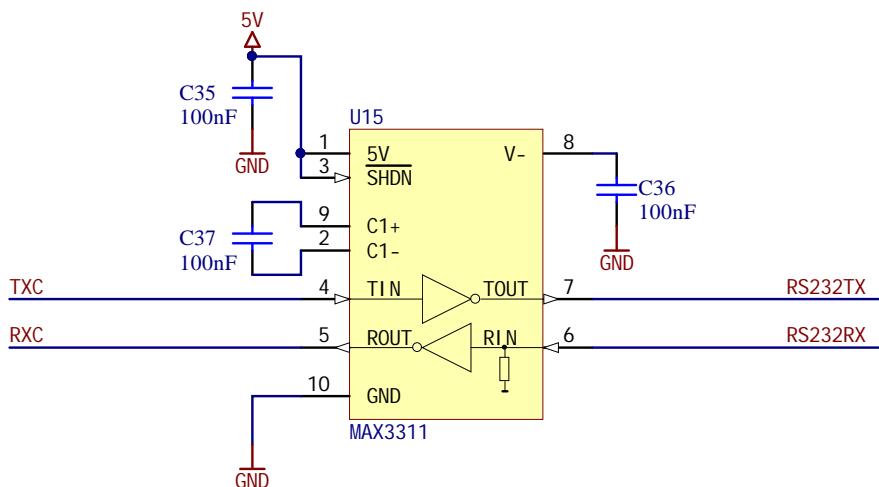
Rysunek 5.23: Konfiguracja pętli PLL dla częstotliwości maksymalnej 168 Mhz

Źródło: Na podstawie [33]

Dodatkowo na rysunku 5.22 widać złącza CON4 (z lewej) i CON7 (na dole), które łączą płytę główną kolejno z enkoderem magnetycznym (5.2) i optycznym (5.3). Połączenie z tym drugim jest zrealizowane tasiemką tego samego typu, co doprowadzane jest zasilanie do głównej PCB.

5.4.3. Komunikacja

Do urządzenia poprzez gniazdo w płytce zasilającej podłączanych jest razem sześć przewodów (CON6 na schemacie 5.2). Cztery z nich służą do komunikacji ze sterownikiem poprzez dwie magistrale szeregowe: CAN (*Controller Area Network*) i RS232.

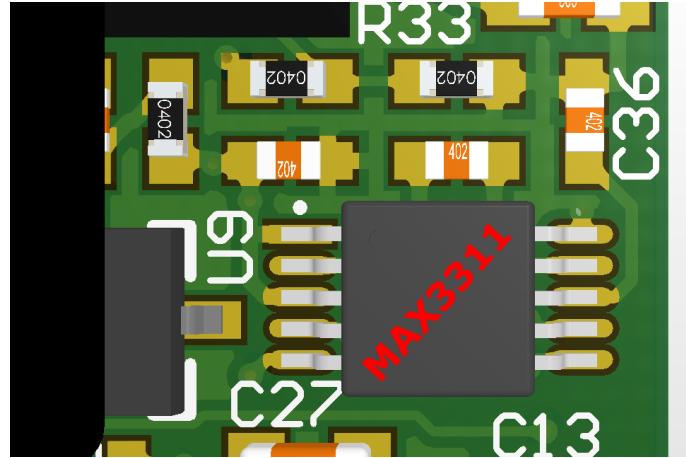


Rysunek 5.24: Zastosowany w kontrolerze układ MAX331 firmy Maxim, służący do translacji poziomów logicznych RS232 na TTL/ CMOS

Źródło: Na podstawie [32]

W większości aplikacji w celu uzyskania poziomów logicznych odpowiednich dla RS232 (+12/-12 V) stosowane są tanie układy typu MAX3232 lub MAX232 (oraz

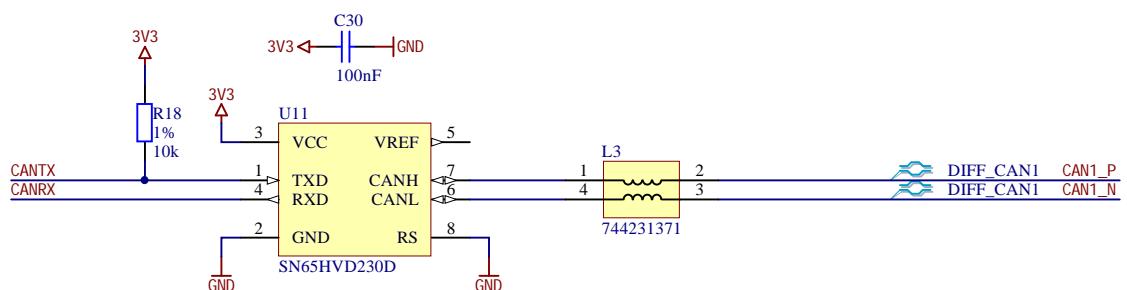
ich odpowiedniki). Najmniejsza obudowa w jakiej są produkowane te układy to TSSOP16 (typ. 5 x 4.4 mm). Dla płytki głównej sterownika jest to za dużo, dlatego został zastosowany mniejszy MAX3311 (rys. 5.24) w obudowie UMAX10 o wymiarach 3 x 3 mm. Urządzenie jest trochę droższe od wspomianych układów i mniej dostępny u popularnych dystrybutorów. Ponadto, musi być zasilany z napięcia 5V, a nie z 3.3 V. Wejścia zastosowanego mikrokontrolera jednak są zabezpieczone przed napięciami na tym poziomie (tzw. *5V compatible*). Wyjściowe 3.3 V z mikrokontrolera wystarczy zmiany stanu w układzie (TTL).



Rysunek 5.25: Komponenty używane przez moduł RS232 zajmują jedynie ok. $20mm^2$

Źródło: Na podstawie [32]

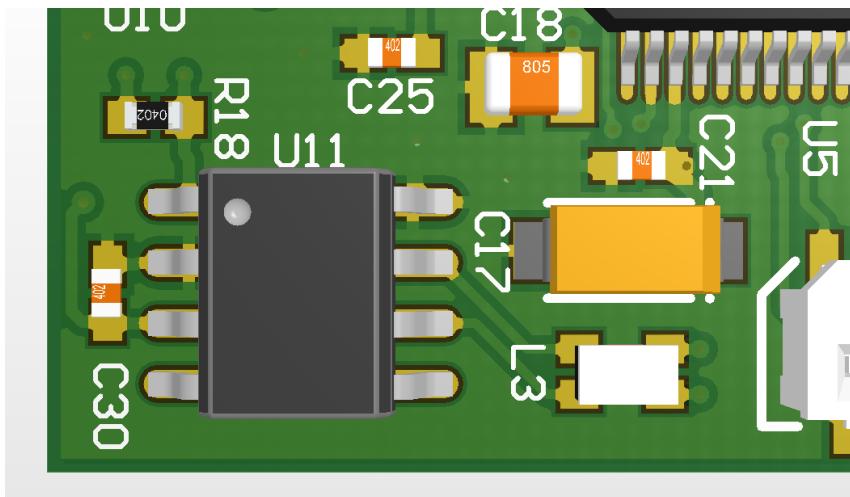
Obwód magistrali RS232 pełni w sterowniku rolę drugorzędną. Główną drogą dla komend sterujących jest CAN, którego zaletą jest to, że bazując na LVDS (*Low-voltage differential signaling*) jest odporny na zakłócenia. W celu zwiększenia impedancji składowej wspólnej od strony sieci został zastosowany dławik (L3 na schemacie 5.26). Ponadto, korzystając z tej magistrali sterownik można podpiąć pod istniejącą sieć, bez potrzeby odłączania pracujących na niej urządzeń.



Rysunek 5.26: Moduł CAN korzystający z układu SN65HVD230D (Texas Instruments) w celu translacji sygnałów z magistrali

Źródło: Na podstawie [32]

Etykietą *DIFF_{CAN1}* (rys. 5.26) oznaczone są linie będące parą różnicową. Daje to możliwość projektując płytę PCB na prowadzenie ścieżek danej pary w taki sposób, aby miały tą samą lub bardzo podobną długość.

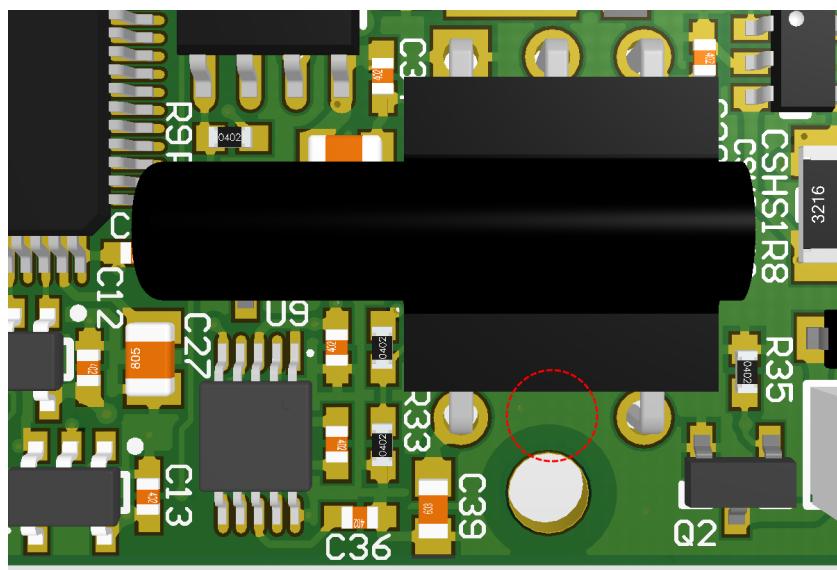


Rysunek 5.27: Widok 3D na moduł CAN

Źródło: Na podstawie [32]

5.4.4. Czujnik podciśnienia

Główica, której częścią jest sterownik, do podnoszenia komponentów elektro-nicznych używa ssawki podciśnieniowej. Możliwość pomiaru wartości aktualnego podciśnienia zapewnia czujnik z serii XFHM firmy Fujikura.



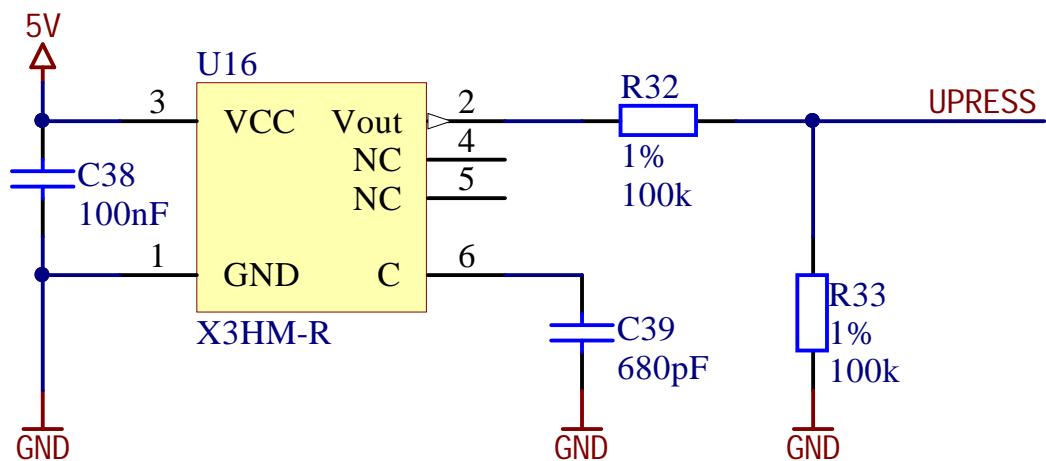
Rysunek 5.28: Czujnik podciśnienia XFHM-100KPGWRH znajdujący się na płytce głównej

Źródło: Na podstawie [32]

Jest to element, którego gabaryty są największe na opisywanej płytce. Ponadto, w wersji z mocowaniem przewodu pneumatycznego poziomo, występuje tylko w obudowie przewlekanej. Czujnik jest pneumatycznie sprzążony z aluminiową kostką (rys. 5.6 i 5.7), w której panuje ciśnienie niższe niż atmosferyczne. Jego wartość spada tym bardziej, im szczerelniej zatkany jest wylot pinoli przez podnoszony komponent. Na wyjściu układu XFHM-100KPGWRH pojawia się potencjał którego

amplituda odzwierciedla wartość mierzonego podciśnienia. Dla zerowej różnicy ciśnień amplituda wynosi 2.5 V (połowa maksymalnego zakresu, którym jest zapięcie zasilania) i spada do 0 V dla -100 kPa lub rośnie do 5V dla 100 kPa. Dodatnia wartość różnicy ciśnienia w normalnej aplikacji nie występuje, kiedy element jest odkładany, zawór zamyka dopływ powietrza i komponent przestaje być zasysany. W niektórych aplikacjach specjalizowanych jednak podnoszone elementy muszą być wydmuchiwane i wtedy mierzona jest dodatkowo wartość dodatnia.

Aby umieścić czujnik w dogodnym miejscu na PCB należało go pozbawić jednego z wyprowadzeń (nr 5) zaznaczonych na rysunku 5.28. Wewnętrznie ten pin nie jest do niczego podłączony, a z racji tego że jest środkowym pinem w danym rzędzie, to pozbicie się go nie zmniejsza stabilności układu.



Rysunek 5.29: Schemat ideowy obwodu czujnika podciśnienia

Źródło: Na podstawie [32]

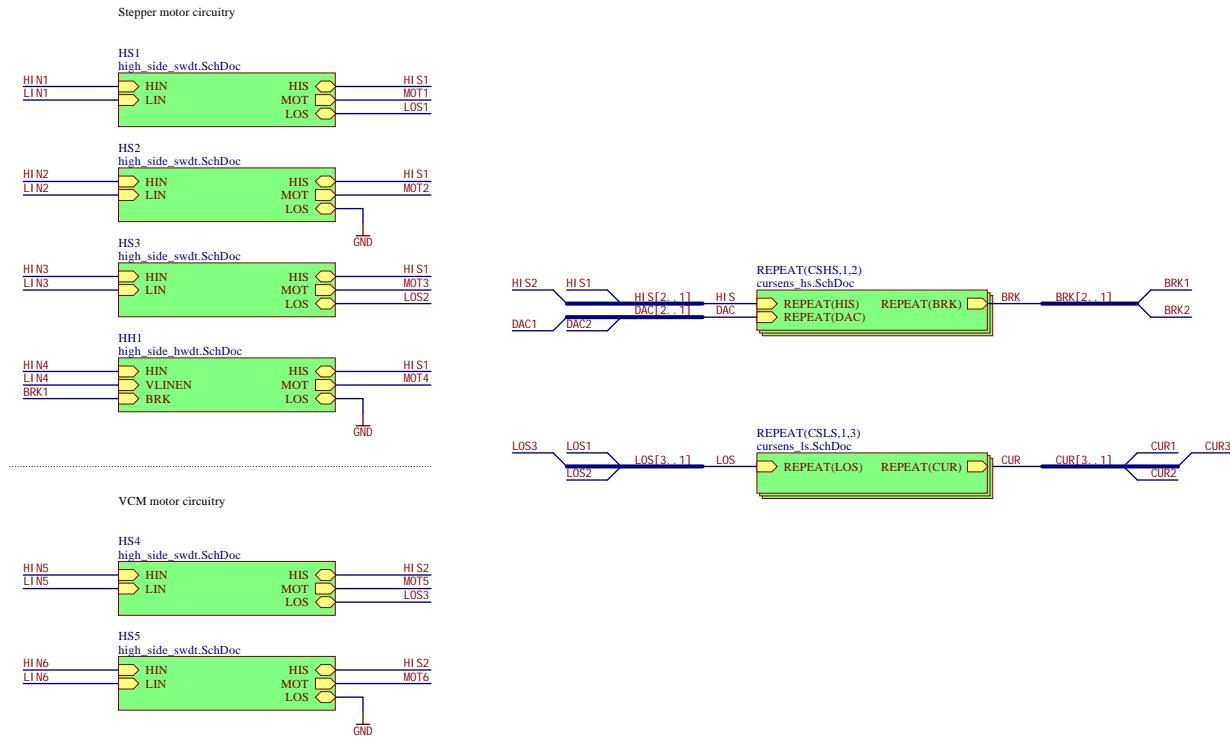
Napięcie odniesienia dla przetwornika analogowo-cyfrowego mikrokontrolera sterującego to 2.5 V (rys. 5.18), dlatego na wyjściu z układu U16 na schemacie 5.29 jest dzielnik napięcia zmieniający zakres napięć z 0 - 5 V do 0 - 2.5 V, aby nie wykraczać poza zakres napięć ADC (*Analog to Digital Converter*).

5.4.5. Obwód sterowania silnikami

W skład modułu sterowania silnikami wchodzą obwody:

- z końcówkami mocy do zasilania danego silnika,
- do cyklicznego pomiaru prądu płynącego w uzwojeniach,
- do odcięcia prądu od uzwojeń w przypadku awarii.

Schemat 5.30 prezentuje opisywany moduł używając zdefiniowanych podmodułów HS, HH, CSHS i CSLS. Funkcja REPEAT w programie Altium służy do mnożenia obiektów, np. w tym wypadku obwody CSHS są dwa, a CSLS trzy.



Rysunek 5.30: Schemat ideowy modułu MOTCIR ze schematu nadzawanego 5.16

Źródło: Na podstawie [32]

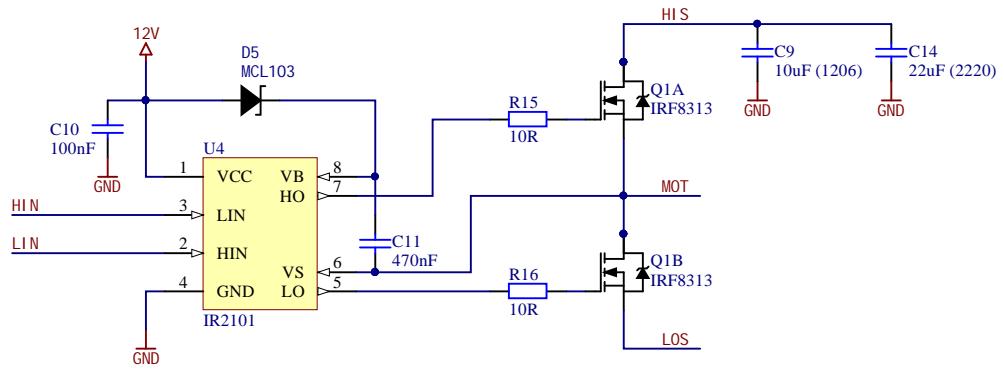
HS - Standardowy sterownik półmostka

W urządzeniu jest zastosowanych 5 obwodów HS (**H**igh **S**ide with **SDt**ime). Służą do tego, aby dostarczać prąd do uzwojeń silników. Pojedynczy moduł HS jest widoczny na schemacie 5.31. Na wejścia układu IR2101 (firma International Rectifier) wchodzą komplementarne sygnały PWM z mikrokontrolera sterującego.

IR2101 jest układem scalonym służącym do sterowania tranzystorami w półmostku po stronie niskiej i wysokiej. Aby otworzyć górnego tranzystora typu N, używane jest napięcie zasilania 12 V z przetwornicy U8 (schemat 5.19). Podczas kiedy dolny tranzystor Q1B jest otwarty, Q1A jest zamknięty, a kondensator C11 ładowany przez diodę Shotkiego D5. Kiedy następuje zmiana wymuszenia kierunku przepływającego w uzwojeniu prądu Q1B się zamyka, a następnie po odczekaniu ustalonego czasu martwego Q1A może zostać otwarty dzięki dodatkowemu napięciu na kondensatorze C11. Tym sposobem potencjał na bramce górnego tranzystora jest wyższy o 12 V od potencjału na jego źródle (minus spadek na diodzie D5 i R15)

IR2101 nie ma zabezpieczenia przed otwarciem dwóch tranzystorów jednocześnie. Czas martwy, który temu zapobiega jest ustawiany w module PWM mikrokontrolera. W celu zapewnienia odpowiedniego kluczowania tranzystorów, w tej aplikacji czas martwy dla wszystkich końcówek mocy został ustawiony na 100 ns.

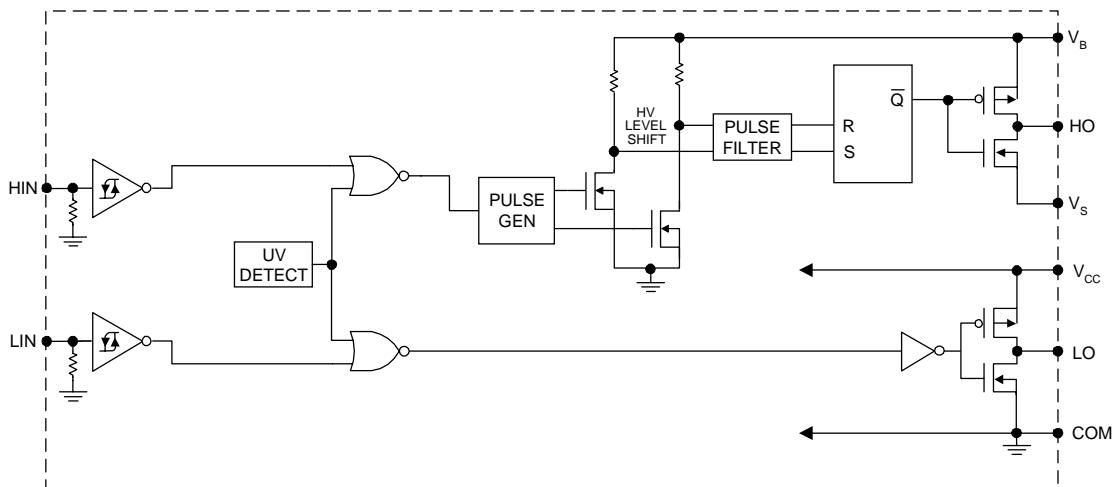
Kondensatory ceramiczne C9 (obudowa 1206) i C14 (2220) zapewniają pojem-



Rysunek 5.31: Schemat ideowy modułu HS (*High Side driver* z programowym czasem martwym)

Źródło: Na podstawie [32]

ność, która pomaga utrzymać stabilny poziom napięcia na uzwojeniu przy kolejnych komutacjach.



Rysunek 5.32: Schemat funkcjonalny *High/ Low side driver'a* IR2101 firmy International Rectifier (obecnie Infineon)

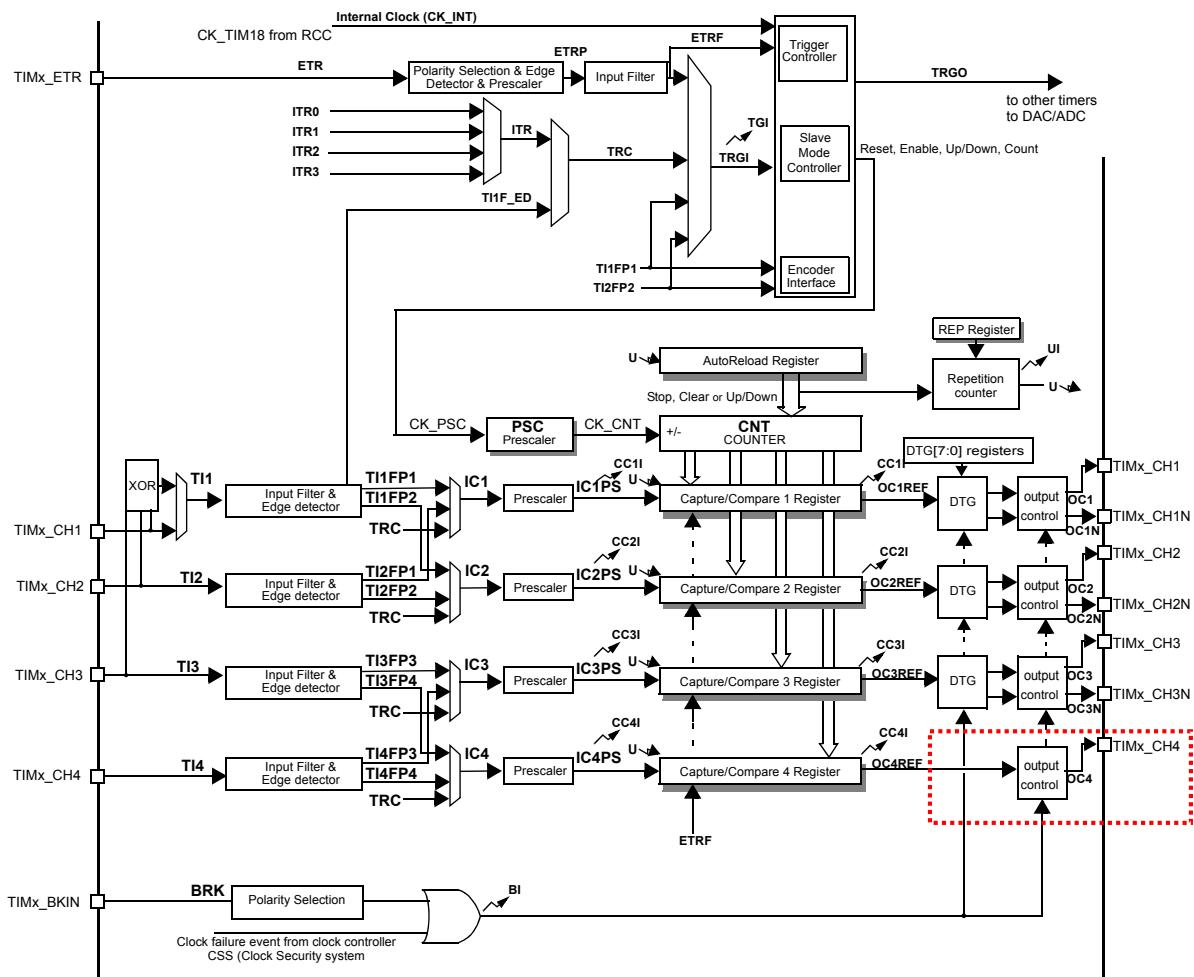
Źródło: Na podstawie [7]

Rezystory w szeregu z bramkami tranzystorów (R15 i R16) służą do ograniczenia prądów bramek. Ponadto zapewniają ochronę układu scalonego IR2101 w przypadku uszkodzenia jednego z tranzystorów, które może doprowadzić do przebicia izolacji między bramką, a drenem i źródłem.

Etykieta MOT oznacza miejsce podłączenia jednego z końców uzwojenia, HIS i LOS to wyprowadzenia z obwodów pomiaru prądu po stronie górnej i dolnej. W obwodzie silnika VCM znajdują się dwa moduły HS, natomiast w module krokowego trzy moduły HS i jeden HH.

HH - modyfikowany sterownik półmostka

Przyczyną dla której w urządzeniu został zastosowany jeden moduł HH (*High Side with Hardware Deadtime*) i pięć HS, zamiast sześciu HS jest to że sterujący mikrokontroler STM32F407VTG6 nie posiada wystarczającej ilości komplementarnych modułów PWM.

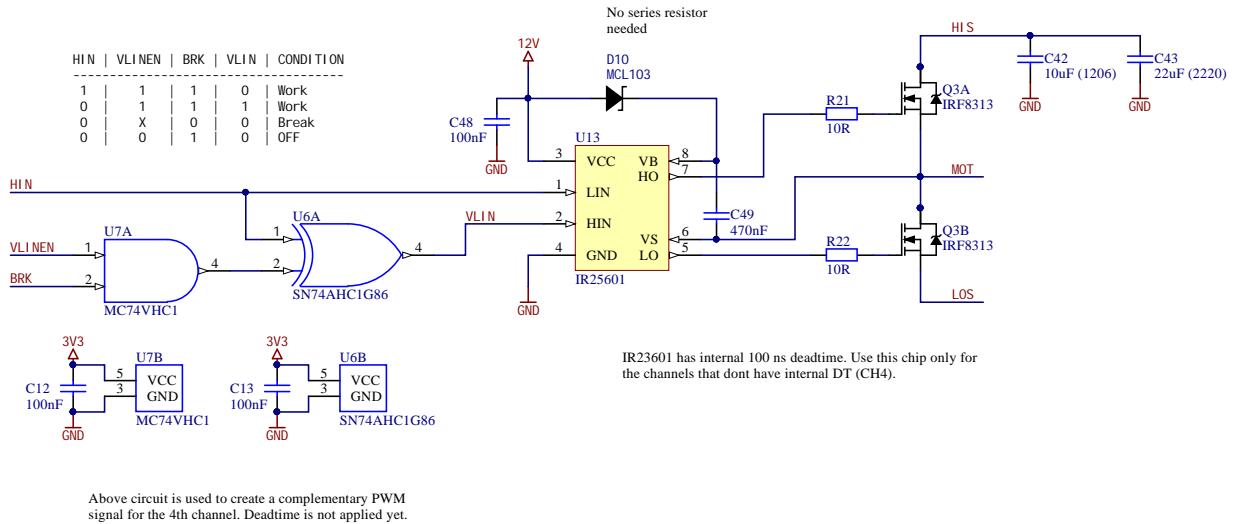


Rysunek 5.33: Schemat blokowy zaawansowanego Timer'a 1 i 8 w mikrokontrolerach STM32F4 firmy ST Microelectronic

Źródło: Na podstawie [14]

Schemat blokowy 5.33 prezentuje architekturę pojedynczego zaawansowanego timer'a w STM32F4. Używany mikroprocesor posiada dwa takie peryferia o numerach 1 i 8. Timer 1 służy do sterowania silnikiem pierwszym (krokowym), a timer 8 drugim (VCM). Jak widać na schemacie, czwarty z kanałów PWM jest uboższy od pozostałych trzech. Nie posiada wyprowadzenia komplementarnego, a co za tym idzie modułu czasu martwego, przez co urządzenie wymaga zastosowania dodatkowego obwodu, który rozszerza możliwości czwartego kanału. Pomimo tego, że timer silnika VCM posiada jeden wolny kanał „bogaty”, to nie mógł on zostać zastosowany w obwodzie silnika krokowego. Burzyłoby to koncepcje uniwersalnego sterownika pod względem programowym, która wymaga aby moduły sterujące silnikami były całkowicie oddzielne. W takim łączonym rozwiązaniu nie

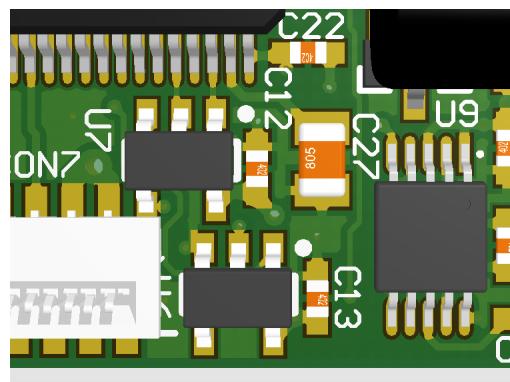
byłoby możliwości między np. zaprogramowania różnej częstotliwości dla każdego silnika. Ponadto, chcąc w przyszłości zastosować obecny kod kontrolera dla nieco odmienionej płytki sterownika, będzie trzeba dokonywać dodatkowych zmian w oprogramowaniu.



Rysunek 5.34: Schemat ideowy modułu HH (*High Side driver*) ze sprzętowym czasem martwym i kanałem komplementarnym

Źródło: Na podstawie [32]

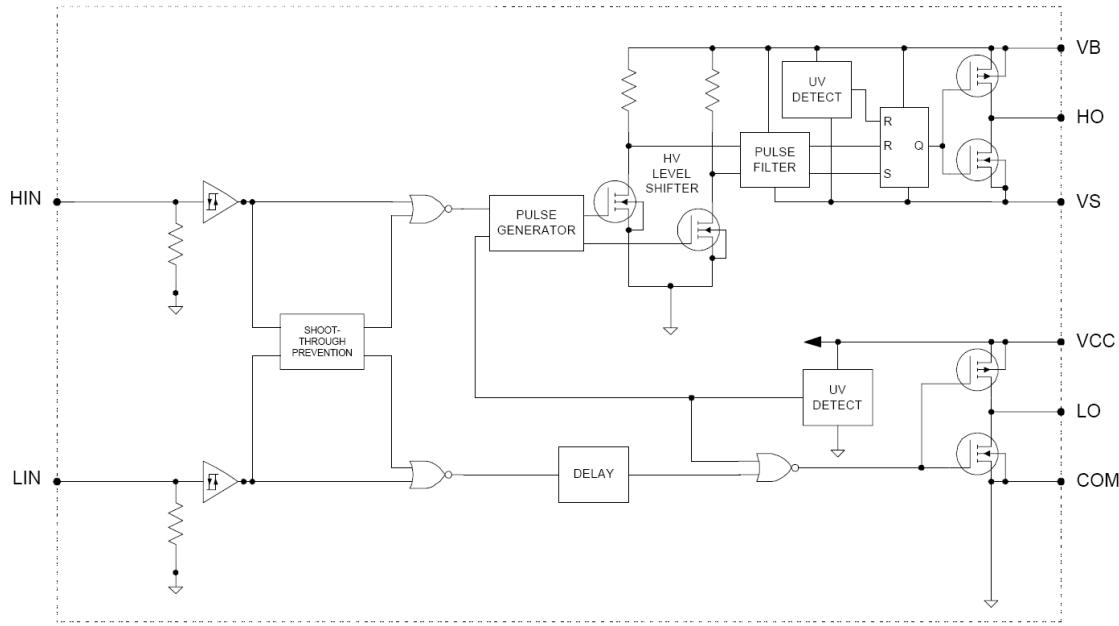
Aby zapewnić czas martwy i dodatkowy kanał komplementarny na kanale czwartym został zastosowany obwód ze schematu 5.34. Sygnał z etykietą HIN jest opisywanym czwartym kanałem. Aby utworzyć z niego dodatkowy sygnał komplementarny została zastosowana bramka XOR (U6), na której jedno z wejść wchodzi HIN. Na drugie wejście U6 jest doprowadzane wyjście z bramki AND (U7). Jej wejścia to z kolei sygnał odblokowujący moduł (etykietą $VLIN_{EN}$) i BREAK, który zmienia stan na niski w przypadku wystąpienia przeciążenia prądowego (rys. 5.37).



Rysunek 5.35: Część PCB w której znajdują się dyskretne bramki U7 i U6 tworzące kanał komplementarny

Źródło: Na podstawie [32]

Bramka AND zapewnia możliwość całkowitego wyłączenia danego półmostka, a czas martwy jest generowany w samym sterowniku, który w obwodzie HH został zamieniony z IR2101 na IR25601 tego samego producenta.



Rysunek 5.36: Schemat funkcjonalny High/ Low side driver'a IR25601 firmy International Rectifier (obecnie Infineon)

Źródło: Na podstawie [7]

Dlatego pomimo tego, że na wejścia układu są podawane komplementarne sygnały bez żadnych opóźnień, to same bramki tranzystorów są przełączane z uwzględnieniem 100 ns czasu martwego (wewnętrzne opóźnienie w układzie).

Tabela 5.2: Tabela prawdy prezentująca działanie zaprojektowanego sprzętowego modułu komplementarnego

Źródło: Na podstawie [33]

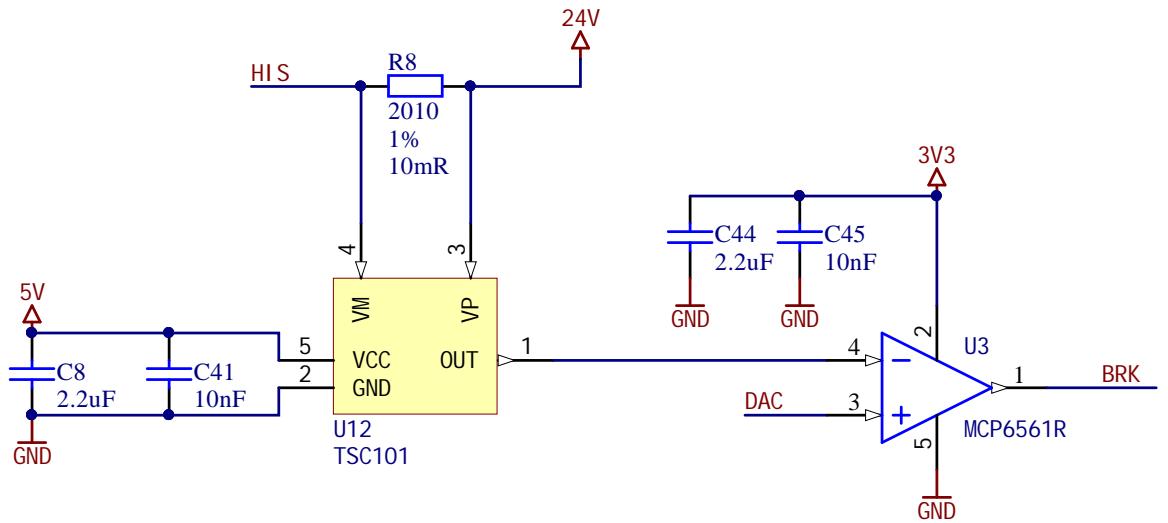
| Sygnały i ich stany ¹ | | | | Tryb pracy |
|----------------------------------|-------------|-----|--------|---|
| HIN | $VLIN_{EN}$ | BRK | $VLIN$ | |
| 1 | 1 | 1 | 0 | normalna praca, górnny tranzystor otwarty |
| 0 | 1 | 1 | 1 | normalna praca, dolny tranzystor otwarty |
| 0 | X | 0 | 0 | przeciążenie prądowe, tranzystory zamknięte |
| 0 | 0 | 1 | 0 | pół-mostek wyłączony |

Obwód w takiej konfiguracji działa bez zarzutu dlatego, że nie wszystkie możliwe stany linii wejściowych są dozwolone. Tabela 5.2 prezentuje cztery możliwe tryby w których może znajdować się obwód. Kiedy sygnał BRK zmienia swój stan na niski, wszystkie wyjścia PWM danego modułu w mikrokontrolerze zmieniają swój stan także na niski (zwykle i komplementarne). To oznacza zamknięcie wszystkich tranzystorów. Dlatego, kiedy BRK jest nisko, HIN też musi i stan na linii $VLIN_{EN}$ nie ma wtedy znaczenia bo $VLIN$ i tak będzie w stanie niskim.

¹ Etykiety sygnałów ze schematu 5.34. Stany: 1 - stan wysoki, 0 - stan niski, X - bez znaczenia przy danym trybie pracy

CSHS - Bezpiecznik prądu krytycznego

Moduły zaawansowanych timerów w mikrokontrolerach STM32F4 posiadają konfigurowalne wejścia zabezpieczające „BREAK” (odpowiednio $TIM1_{BKIN}$ i $TIM8_{BKIN}$, patrz schemat 5.33). Dla zastosowanych w sterowniku ustawień podanie na pin Break stanu niskiego spowoduje wyłączenie końcówek mocy skojarzonego silnika. Urządzenie korzysta z tej funkcjonalności w taki sposób, że kiedy prąd w uzwojeniu przekroczy zadany poziom, silniki przestają działać niezależnie od tego, jakie procedury są akurat wykonywane w oprogramowaniu. Jest to zabezpieczenie ostateczne zapobiegające uszkodzeniu uzwojeń silników.



Rysunek 5.37: Schemat ideowy modułu CSHS (*Current Sense at High Side*) zabezpieczającego uzwojenia przed przeciążeniem prądowym

Źródło: Na podstawie [32]

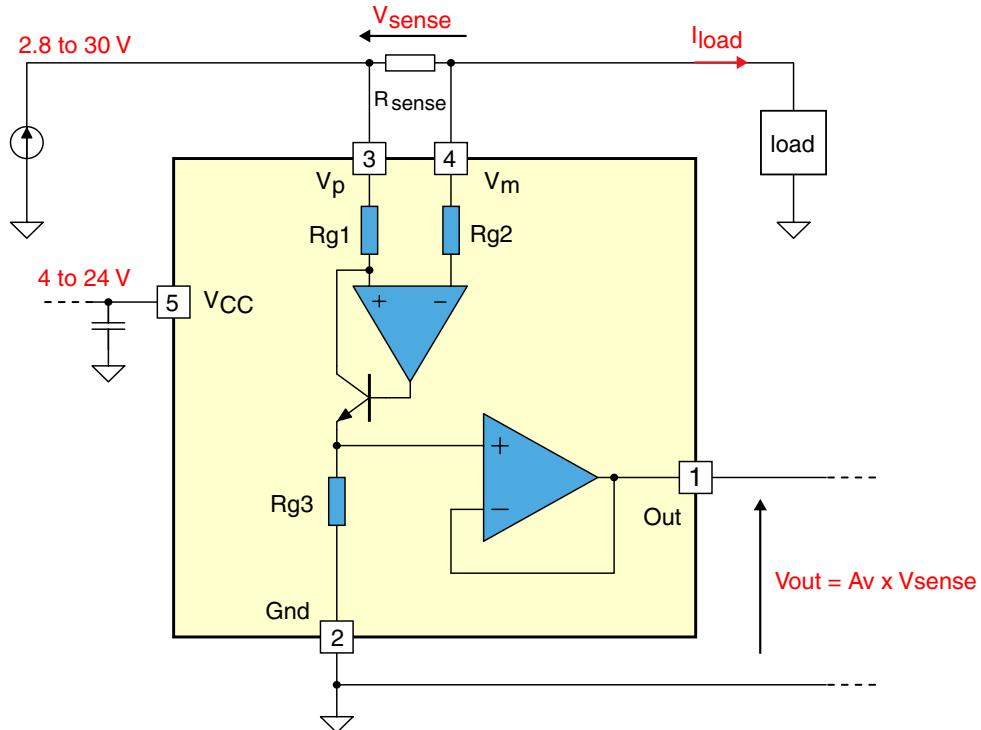
W celu uzyskania opisywanej funkcjonalności został zastosowany obwód z modułu HSCS (rys. 5.37). Etykieta DAC to sygnał z przetwornika cyfrowo-analogowego mikrokontrolera sterującego. W taki sposób ustawiany jest zakres na wejściu nieodwracającym komparatora napięcia MCP6561R. Na jego wejście odwracające doprowadzane jest napięcie z wzmacniacza pomiarowego TSC101 (5.38), które odzwierciedla poprzez napięcie wartość płynącego prądu w uzwojeniach danego silnika. Zasada działania układu scalonego TSC101 bazuje na zwierciadle prądowym. Może on być zasilany z napięcia niższego niż te, które trafiają na jego wejścia. Napięcie na wyprowadzeniu OUT układu można wyliczyć z wzoru:

$$V_{OUT} = R8 \cdot I_{R8} \cdot G \quad (5.2)$$

gdzie G to wzmacnienie typowe dla danego modelu TSC101, układ stosowany w sterowniku ma końówkę „A”, która oznacza $G = 20$.

Kiedy prąd płynący w uzwojeniach danego silnika (i jednocześnie przez rezystor R8) jest wystarczająco wysoki, aby napięcie na wyjściu U12 przekroczyło te zadane

przez mikrokontroler poprzez DAC, to wyjście komparatora U3 przechodzi w stan niski uruchamiając zabezpieczenie prądowe. Poprzez wyjście DAC istnieje możliwość ustawienia prądu odłączenia z bardzo wysoką dokładnością (rozdzielcość przetwornika to 12 bitów). Nie ma jednak możliwości pomiaru prądu. Tą funkcjonalność zapewnia moduł CSLS.



Rysunek 5.38: Schemat funkcjonalny wzmacniacza pomiarowego TSC101 firmy ST Microelectronic

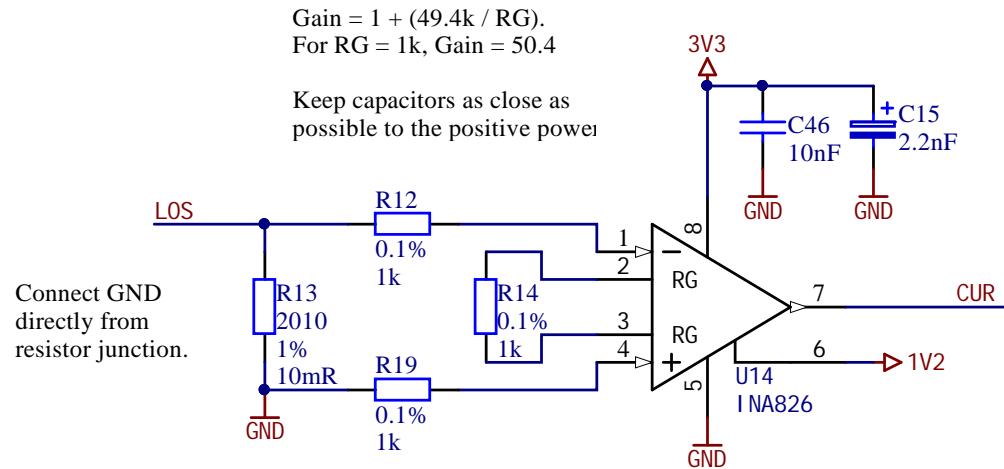
Źródło: Na podstawie [14]

CSHS - Obwód pomiaru prądu

Każdy z mostków H posiada jeden moduł CSLS (dwa w obwodzie silnika krokowego i jeden w VCM). Moduł służy do cyklicznego pomiaru spadku napięcia na rezystorze pomiarowym, znajdującym się w dolnej części mostka. Po przeliczeniu daje to dokładną informację o natężeniu prądu znajdującej się w uzwojeniu, która jest używana w celu programowego zabezpieczenia przed przeciążeniem (zanim zadziała zabezpieczenie krytyczne) i w procedurze regulacji prądu w sterowaniu momentem obrotowym.

Rysunek 5.39 przedstawia schemat ideowy obwodu CSLS. Do pomiaru spadku na rezystorze pomiarowym został zastosowany specjalizowany wzmacniacz pomiarowy INA826 firmy Texas Instruments. Do regulacji wzmacniania używany jest rezystor $1k\Omega$ o dokładności 0.1 % podłączony do zacisków RG. W prezentowanej aplikacji wzmacnienie wynosi $G = 50.4$. Napięcie odniesienia 1.2 V (schemat 5.18) jest używane do ustawienia punktu pracy wzmacniacza (*Bias voltage*). W tej konfiguracji napięcie spoczynkowe na wyjściu wzmacniacza wynosi 1.2 V i obniża się

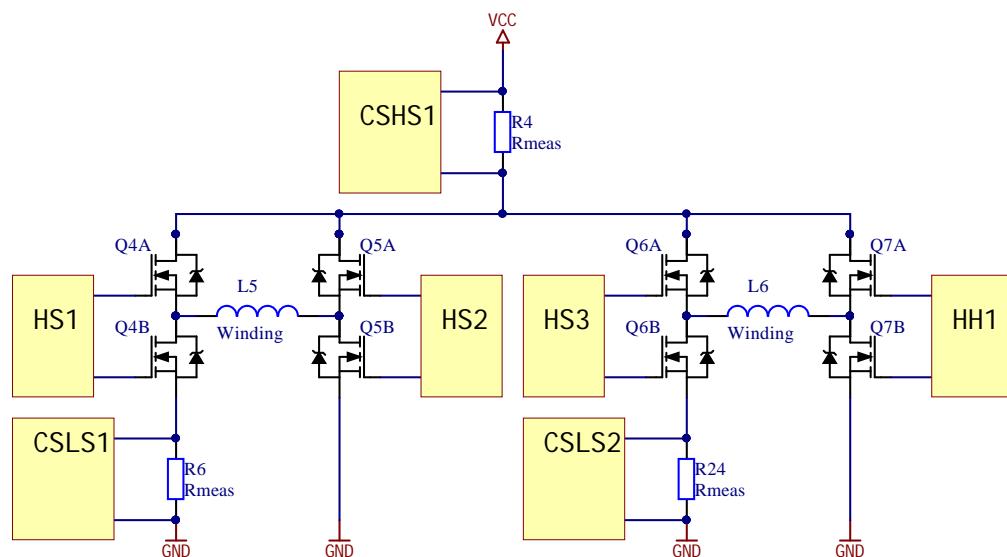
(min. 0 V) lub wzrasta (max 3.3 V) w zależności od kierunku przepływu prądu przez rezystor pomiarowy R13. Użyteczny zakres na przetwornika ADC mikrokontrolera to od 0 do 2.5 V (napięcie odniesienia ADC).



Rysunek 5.39: Schemat ideowy modułu CSLS (*Current Sense at Low Side*) służącego do pomiaru prądu w uzwojeniach

Źródło: Na podstawie [32]

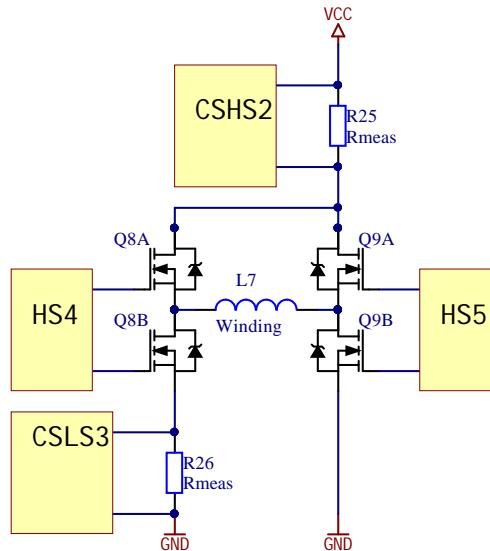
Schematy blokowe 5.40 i 5.41 obrazują konfigurację opisanych modułów w odniesieniu do obwodów obu silników.



Rysunek 5.40: Schemat blokowy obwodu silnika krokowego

Źródło: Na podstawie [33]

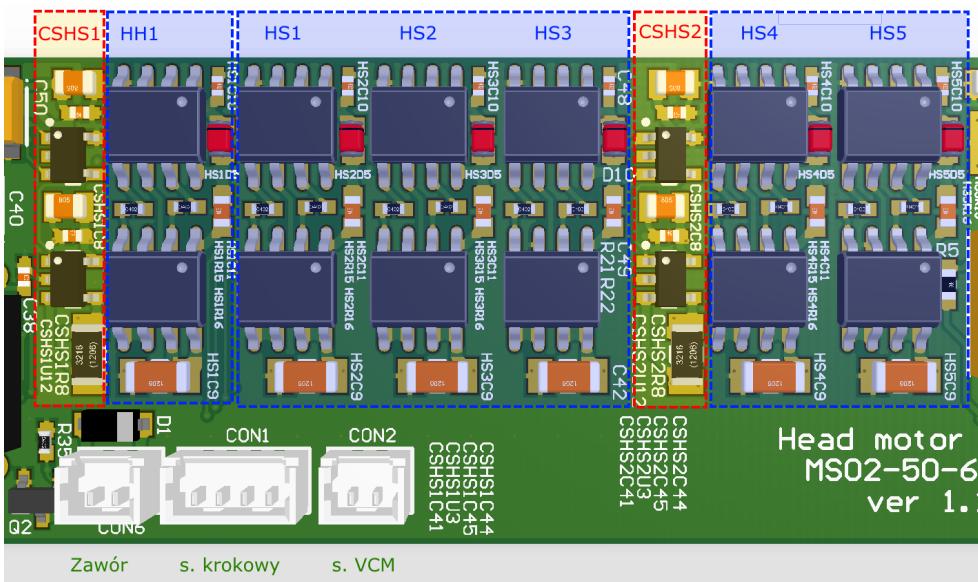
Silnik krokowy wymaga pomiaru natężenia prądu w dwóch miejscach ze względu na to, że posiada dwa uzwojenia. Bezpiecznik prądu jest wymagany tylko jeden, gdyż liczy się dla niego tylko prąd całkowity, bez względu na to w którym uzwojeniu nastąpiło przeciążenie.



Rysunek 5.41: Schemat blokowy obwodu silnika VCM

Źródło: Na podstawie [33]

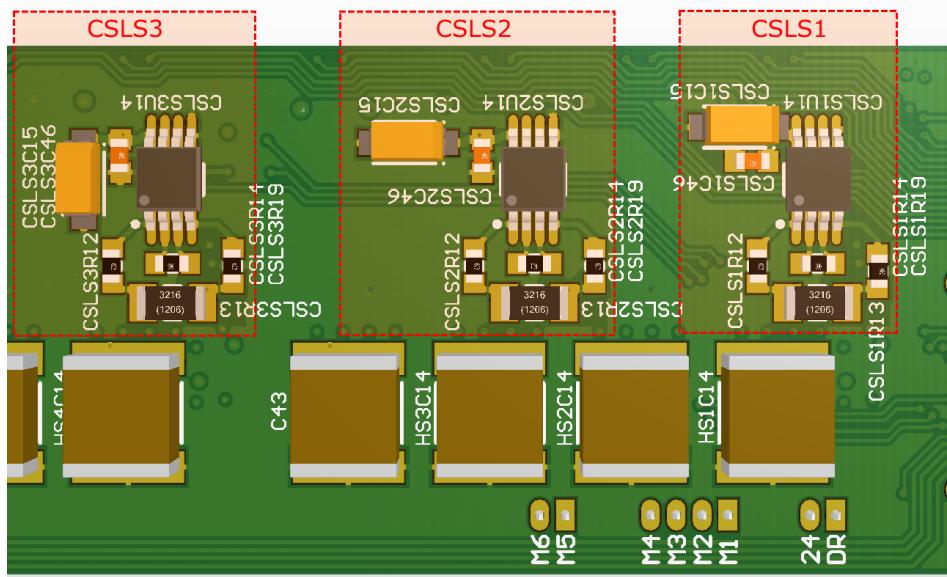
Na rysunku 5.42 widać ułożenie tych z opisywanych modułów, które znajdują się po górnej stronie płytki głównej. Złączami CON1, CON2 i CON5 połączane są kolejno silnik krokowy, silnik VCM i zawór podciśnienia.



Rysunek 5.42: Wyszczególnione moduły obwodów silników po stronie górnej PCB

Źródło: Na podstawie [32]

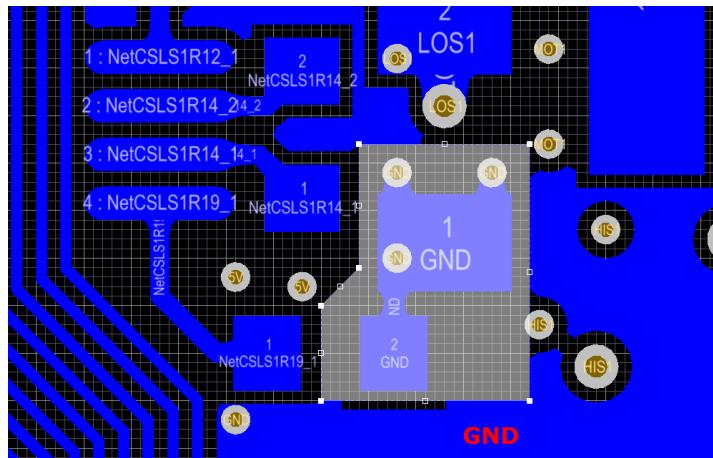
Obiekty CSHS nie zmieściły się na górze i zostały przeniesione na dolną warstwę (rys. 5.43). W celu uzyskania jak najlepszych, bezzakłócenioowych odczytów spadków napięć na rezystorze pomiarowym, potencjał pomiędzy R13 a R19 (schemat 5.39) jest odłączony od „wylanej” na dolnej warstwie masy i jest połączony jedynie przelotkami z warstwą środkową (patrz rys. 5.44). Dokładne procedury pomiaru prądu w uzwojeniach i sterowania silnikiem zostały opisane w rozdziale 6.



Rysunek 5.43: Wyszczególnione moduły obwodu silników po stronie dolnej PCB

Źródło: Na podstawie [32]

Podsumowując, udało się zbudować kompletny sterownik do dwóch silników działających w pętli ze sprzężeniem zwrotnym używając przy tym bardzo ograniczonej przestrzeni. Nie udało się jednak umieścić wszystkich modułów na jednym PCB i dlatego poszczególne obwody są rozproszone na cztery płytki. Mimo to, złożenie prototypu nie stanowi problemu i jest to czynność powtarzalna.



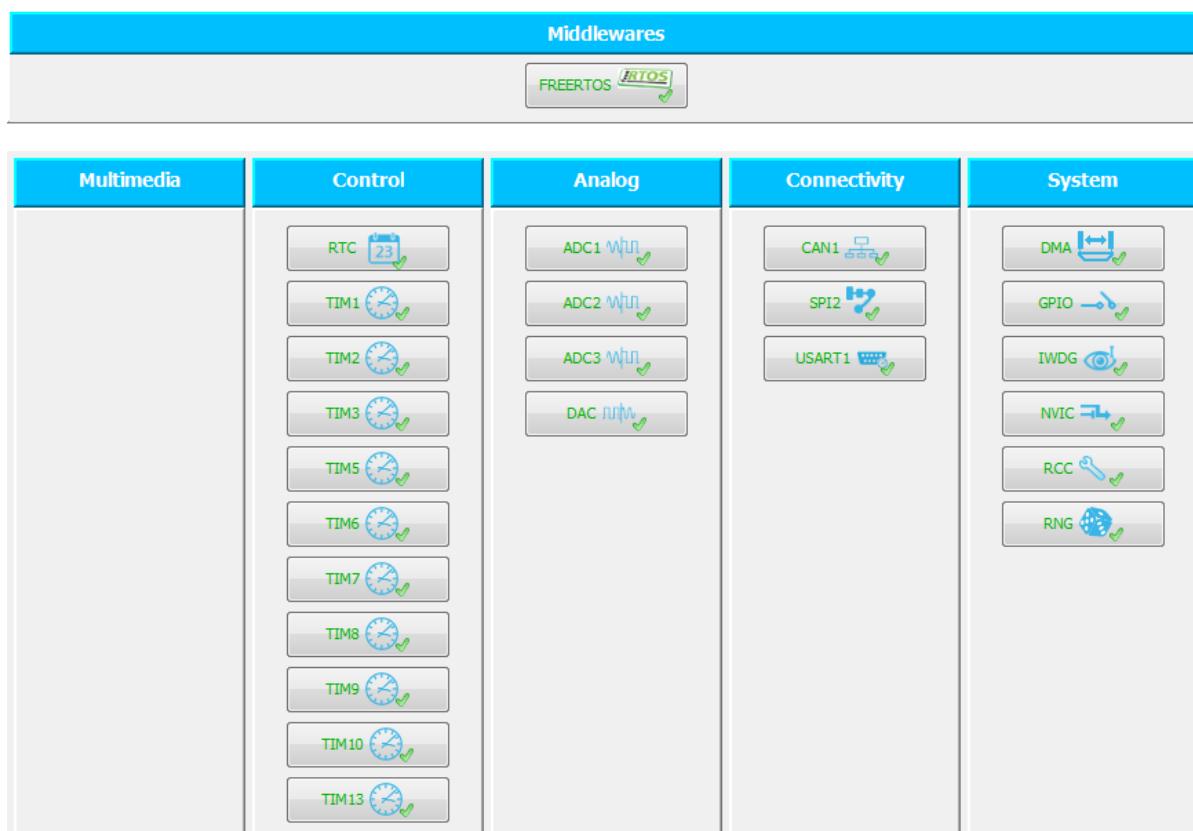
Rysunek 5.44: Zaznaczony niedostępny obszar dla propagacji masy po dolnej stronie PCB

Źródło: Na podstawie [32]

6. Oprogramowanie

Program wykonywany w mikroprocesorze sterującym został napisany w języku C i w nielicznych częściach w assemblerze. Jądrem całej architektury jest system operacyjny czasu rzeczywistego FreeRTOS™. Wszystkie procedury wysokiego poziomu wykonywane w kodzie są umieszczone w wątkach o odpowiednich priorytetach, które porozumiewają się między sobą za pomocą kolejek, semaforów oraz obszarów pamięci współdzielonej. Zadania najniższego poziomu są wykonywane w procedurach obsługi przerwań, np. komutacja lub kolejkowanie ramek CAN.

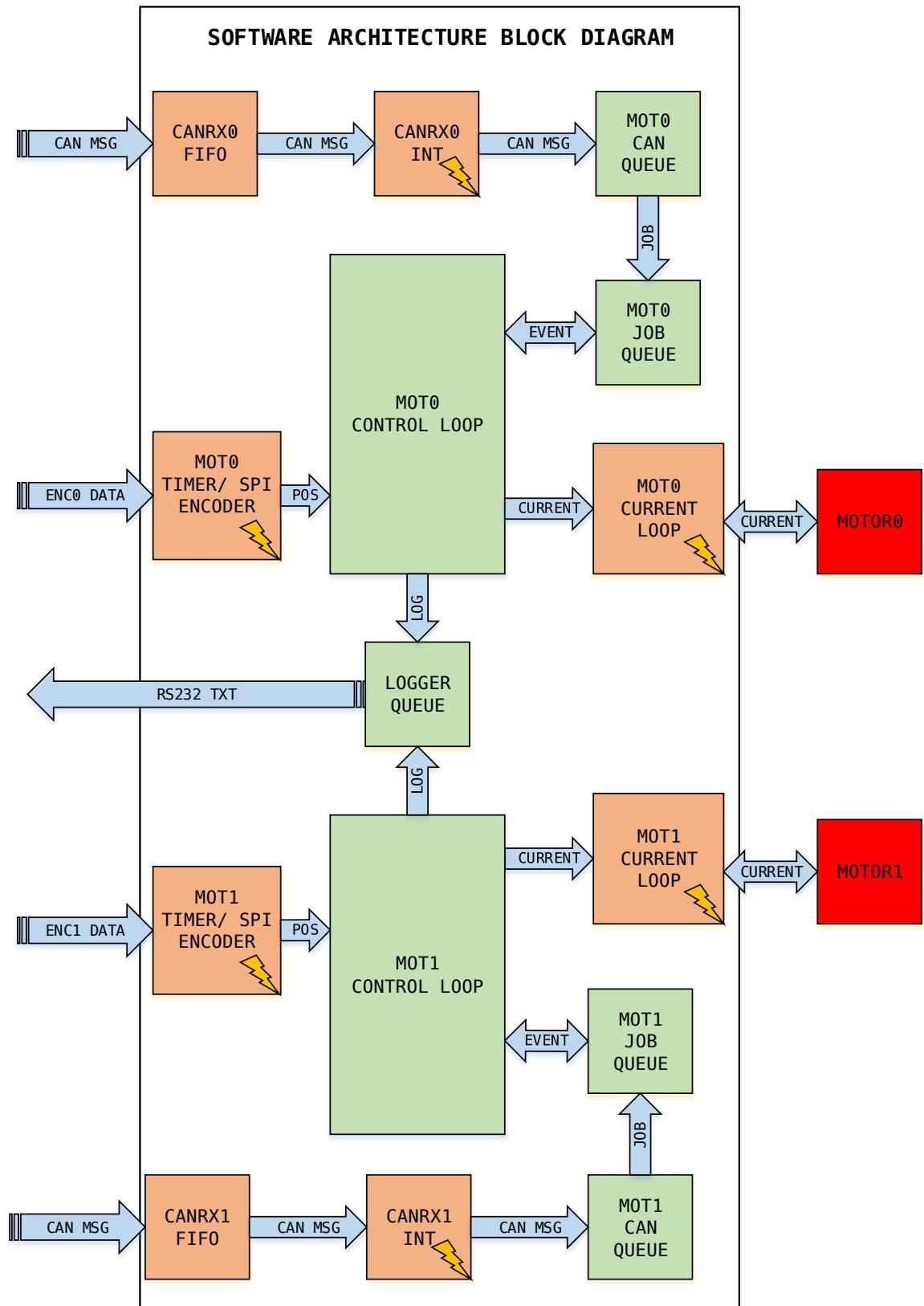
Rysunek 6.2 przedstawia poglądowy schemat blokowy architektury systemu. Pomarańczowymi blokami zostały oznaczone procedury wykonywane na poziomie sprzętowym lub programowym, na warstwie znajdującej się niżej od wątków systemu FreeRTOS™. Kolor zielony oznacza wątki/ procesy w systemie FreeRTOS™. Strzałki z paskami symbolizują komunikaty przychodzące do kontrolera ze świata zewnętrznego, a bez nich komunikaty wewnętrzne systemowe. Dokładniejszy opis poszczególnych bloków znajduje się w korespondujących z nimi sekcjach.



Rysunek 6.1: Widok wstępnie skonfigurowanych peryferiów w programie mxCube firmy ST Microelectronics

Źródło: Na podstawie [33]

Do wygenerowania kodu zawierającego wstępную konfigurację peryferiów służy program mxCube firmy ST Microelectronics.



Rysunek 6.2: Schemat blokowy architektury oprogramowania sterownika

Źródło: Na podstawie [33]

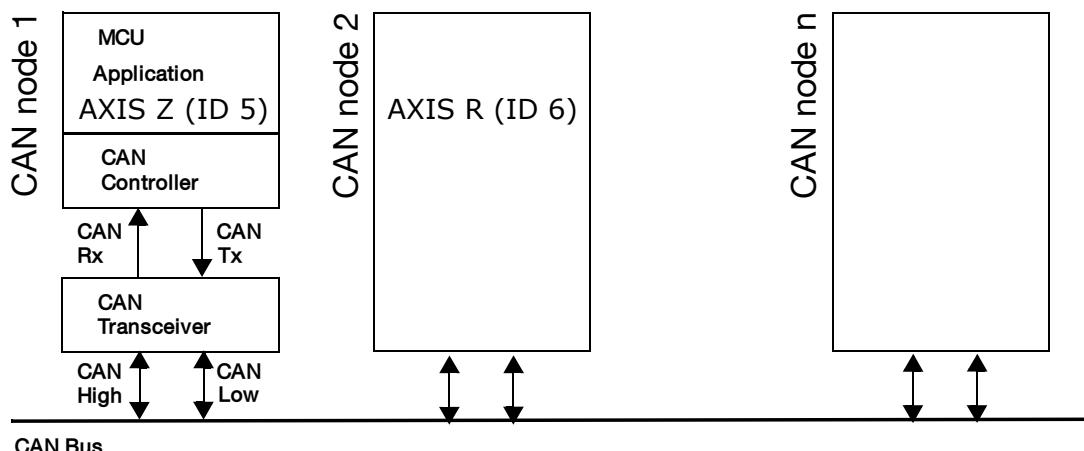
Jego możliwości konfiguracyjne są bardzo podstawowe, ale pozwala on w graficzny sposób zobrazować używane moduły i wyprowadzenia (rys. 6.1). Weryfikację symulacyjną i analizę graficzną działania złożonych algorytmów przeprowadzono w środowisku MATLAB/ SIMULINK.

6.1. Komunikacja

Gotowe do obsługi programowej w urządzeniu są dwie magistrale szeregowe (patrz 5.4.3). W tej sekcji znajduje się opis zastosowania każdej z nich na obecnym poziomie rozwoju.

6.1.1. Magistrala CAN

Magistrala pracuje w protokole CAN 2.0B (rozszerzona ramka: 29 bitowy identyfikator i 64 bitowe pole danych) z prędkością 1 Mbit/s. Używany protokół to TML CAN, który jest używany przed dotychczas używane w maszynach sterowniki firmy Technosoft. Jak zostało wcześniej wspomniane, projektowane urządzenie ma za zadanie zastąpić w maszynie dwa dotychczasowe, dlatego z punktu widzenia sieci są to dwa kontrolery (rys. 6.3).

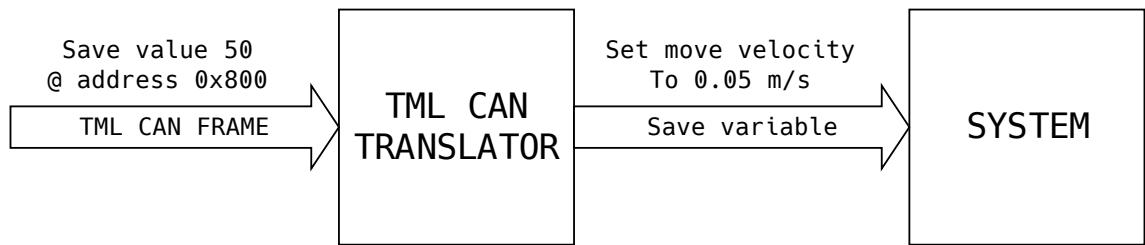


Rysunek 6.3: Topologia sieci CAN. Sterownik przedstawia się jako dwa osobne urządzenia

Źródło: Na podstawie [14]

W rzeczywistości kontroler jest tylko jeden. Mikrokontroler STM32F07VTG6 posiada dwie sprzętowe kolejki typu FIFO (*First In First Out*) do odbierania ramek CAN, które są współdzielone przez moduły CAN1 i CAN2. Urządzenie korzysta tylko z jednego peryferia, więc obie kolejki mogą być zastosowane przez nie. Filtry ramek przychodzących są skonfigurowane w taki sposób, aby przepuszczać tylko te ramki, które posiadają w swoim identyfikatorze numer osi Z lub R (odpowiednio 5 i 6). Po odebraniu wiadomości dla osi Z, rama trafia do pierwszej kolejki FIFO, jeśli wiadomość jest dla osi R, to do drugiej. Taka architektura zapewnia bardzo dobre odseparowanie wirtualnych modułów odbiorczych sterowników już na poziomie sprzętowym. Bufory FIFO mają jednak ograniczoną pojemność – mieszczą tylko 3 ramki. Z tego względu wiadomości do nich wpadające muszą być jak najszybciej odczytane i ulokowane w programowej kolejce o rozmiarze 32 ramek, w której mogą oczekiwać na interpretację przez dłuższy okres.

Dostęp do zmiennych (obszarów pamięci) w sterownikach Technosoft'u odbywa się poprzez zapis do i odczyt z rejestrów. Architektura konstruowanego kontrolera nie wspiera w natywny sposób takiego programowania, dlatego w celu poprawnej pracy potrzebna jest dodatkowa warstwa (rys. 6.4).



Rysunek 6.4: Sposób tłumaczenia ramek TML CAN w kontrolerze. Przykładowy zapis prędkości ruchu w osi Z

Źródło: Na podstawie [33]

W warstwie tłumaczenia ramek TML dokonywanych jest wiele różnych operacji. W zależności od komunikatu może to być zapis do zmiennej lub np. wywołanie funkcji. W przykładzie zaprezentowanym na rysunku 6.4 zewnętrzne urządzenia wysyła do kontrolera wiadomość TML, która po przeskalowaniu ustawia prędkość ruchu w osi Z.

6.1.2. Magistrala RS232

Docelowo interfejs RS232 ma umożliwiać alternatywną metodę komunikacji z urządzeniem w celu programowania i sterowania. Obecnie jednak jest to kanał typowo serwisowy. Korzysta z niego moduł logera (rys. 6.2) aby wysyłać do aplikacji komputerowej informacje o zdarzeniach w HRF (*Human Readable Format*). Typowa ramka tego typu zawiera informacje o czasie wystąpienia zdarzenia, poziomie logowania (np. informacja o priorytecie normalnym lub krytycznym) oraz tekst dowolny (listing 6.1).

Listing 6.1: Przykładowe logi informujące o wywołaniu funkcji poprzez CAN

```

1 | [00:01:18.788] [Info] canRxID:5 opCode:0x402 dcl:0 data7-4:0xFFFF0F00 data3-0:0x0
2 | [00:01:18.788] [Debug] CAN AxisReset
  
```

Źródło: Na podstawie [33]

Innym trybem logowania jest format CSV (*Comma Separated Values*), który pozwala na zbieranie danych w czasie rzeczywistym w celu ich dalszej analizy w programach MATLAB lub Excel. Obsługa modułu logera nie pochłania dużej ilości czasu procesora ze względu na to, że wątek ściągający kolejne logi z kolejki oraz kanał DMA, przez które są one wysyłane mają najwyższe priorytety – procedury wykonują się tylko wtedy, kiedy pozostałe procesy są uśpione.

6.2. Realizacja zadań

Wysunięcie się cylindra silnika VCM o odległość x , zwrócenie wartości podciśnienia w głowicy czy obrót komponentu o 90° – wszystko to jest przykładem wywołania funkcji, czy też realizacją zadania. Każda funkcja ma na wykonanie się określona ilość czasu i jest zbudowana na zasadzie skryptu, który czeka na zajście konkretnych wydarzeń (*Events*), interpretuje je i wykonuje kolejne procedury w zależności od ich wyniku. W przypadku nie wykonania się którejkolwiek części skryptu w danym czasie, skrypt jest anulowany i do urządzenia wydającego komendę zostaje zwrócony kod błędu. System interpretuje obecnie 9 zdarzeń (listing 6.2).

Listing 6.2: Obiekt typu *enum* prezentujący obsługiwane w sterowniku wydarzenia

```
1  /*
2   * @brief      enum representing possible triggers occurring in the driver
3   */
4  typedef enum
5  {
6      e_sysTrig_None          = 0,
7      e_sysTrig_EXTI          = 1,
8      e_sysTrig_PosError      = 2,
9      e_sysTrig_DistReached   = 3,
10     e_sysTrig_Go             = 4,
11     e_sysTrig_Delay_ms       = 5,
12     e_sysTrig_SetPwm1        = 6,
13     e_sysTrig_SetPwm2        = 7,
14     e_sysTrig_CompAdc1       = 8,
15     e_sysTrig_CompAdc2       = 9,
16 } sysTrigg_t;
```

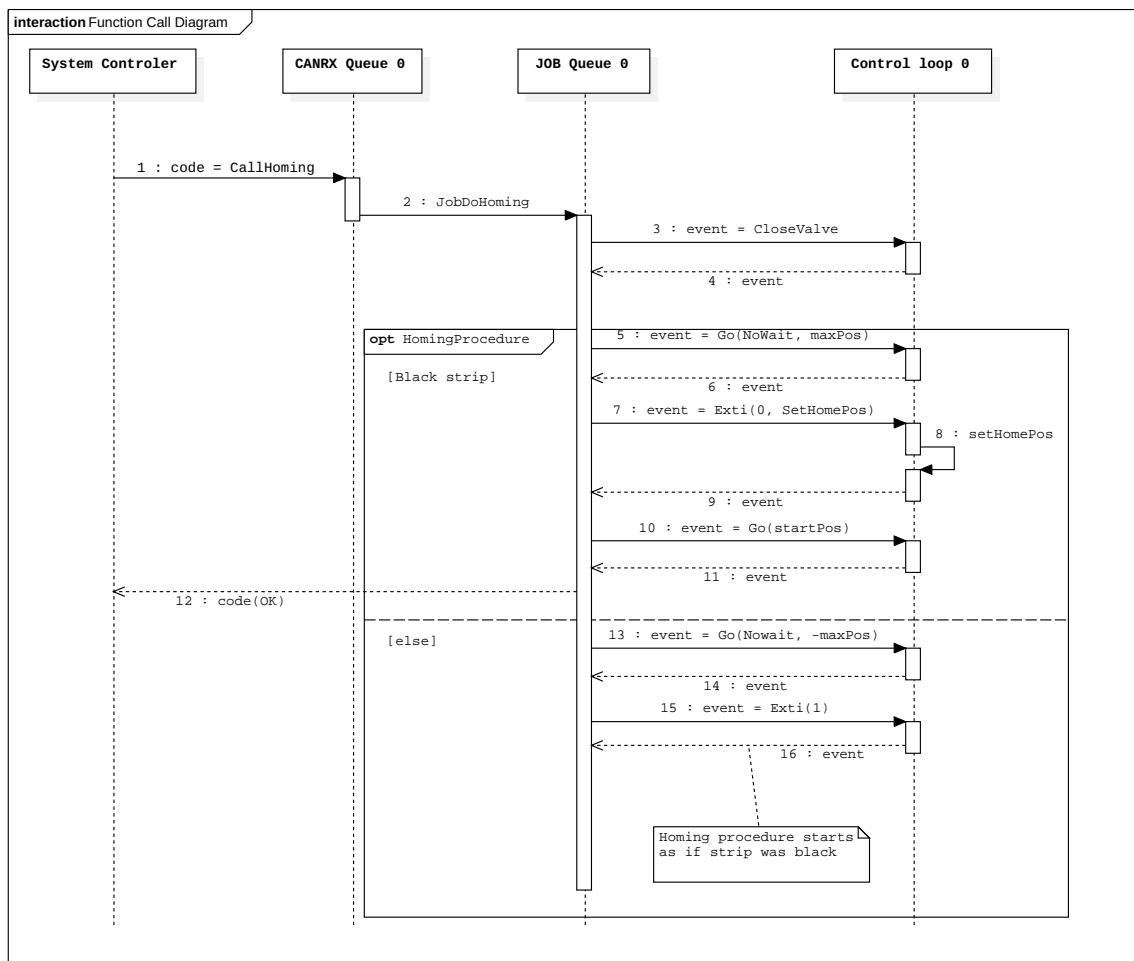
Źródło: Na podstawie [31]

Zdarzenia mogą zachodzić sekwencyjnie lub równolegle. Skrypt może oczekiwany na zajście kilku skryptów (operatorzy *AND* i *OR*), gdzie każdy z nich także posiada ustalony czas na wykonanie się.

Rysunek 6.5 przedstawia przykładową procedurę (homing osi Z) w języku UML (*Unified Modeling Language*) wykonywaną przez system. W pierwszej kolejności zewnętrzne urządzenie (Kontroler systemowy automatu Pick and Place) wysyła poprzez magistralę CAN wywołanie funkcji Homing'u. Ramka trafia do kolejki odbiorczej CAN. Po interpretacji zostaje wybrany odpowiedni skrypt (*JobDoHoming*) i rozpoczyna się jego wykonywanie.

Pierwszym etapem działania skryptu jest zamknięcie zaworu podciśnienia. Następnie sprawdza na którym polu paska referencyjnego znajduje się część ruchoma głowicy (sekcja 5.3). Jeśli jest to pole czarne (cylinder silnika VCM wsunięty), następuje wejście w pierwsze rozwidlenie w skrypcie (Black strip). Procedura homing'u służy do kalibracji pozycji. Zanim zostanie wykonana, absolutna pozycja cylindra nie jest znana. Po włączeniu zasilania przyjmuje się, że pozycja w której aktualnie znajduje się cylinder to zero. Wiedząc, że cylinder jest w polu czarnym paska, następuje wysuwanie cylindra według określonej trajektorii (sekcja 6.4.2). Od razu po tym, funkcja oczekuje na zajście z czarnej strefy paska na białą (wiadomość *EXTI* na diagramie). Kiedy to nastąpi, zostaje ustalona absolutna pozycja cylindra. Od tej

chwili można swobodnie sterować wysuwem głowicy. Na końcu cylinder wraca do ustalonej pozycji startowej.



Rysunek 6.5: Diagram interakcji prezentujący wykonanie przez system procedury bazowania (*homing*) w osi Z

Źródło: Na podstawie [33]

Jeśli z jakiegoś powodu podczas uruchomienia procedury homing'u cylinder nie jest wsunięty (np. rozcięgnięta lub uszkodzona sprężyna) i głowica znajduje się nad polem białym, najpierw następuje jej wsunięcie, aby homing zawsze następował w ten sam sposób (zejście z pola czarnego na białe). Potwierdzenie wykonania zadania jest wysyłane do urządzenia wywołującego funkcję, w tym wypadku do kontrolera systemu.

Wszystkie zadania obecnie obsługiwane przez sterownik są wykonywane na przedstawionej na rys. 6.5 zasadzie (np. podniesienie komponentu, położenie go lub obrócenie).

6.3. Algorytmy enkodera magnetycznego

W odróżnieniu do enkodera liniowego, dane odczytywane ze skonstruowanego modułu enkodera magnetycznego bazującego na układzie AS5048A (rys. 5.4) są obarczone błędami związanymi z nieliniowością układu i typem enkodera. Bez

zastosowania żadnych algorytmów uśredniających odczytaną pozycję kątową, dokładność systemu była by marna, a sam układ napędowy działał by niestabilnie.

Aby umożliwić pracę enkodera z rozdzielczością do 0.05° w sterowniku zostały zaimplementowane następujące algorytmy:

- uśrednianie bazujące na średniej arytmetycznej,
- filtr Kalmana (opracowany na podstawie [27] i [3]),
- filtr dolnoprzepustowy jednobiegunkowy, rekursywny (opracowany na podstawie [36]).

6.3.1. Uśrednianie

Pierwszy z wymienionych algorytmów wykonywany jest zawsze. Pozostałe dwa natomiast są uruchamiane naprzemiennie, w zależności od konfiguracji użytkownika. Uśrednianie działa na prostej zasadzie. Wykonywanych jest kilka odczytów pozycji z enkodera (ilość odczytów zależy od konfiguracji, najczęściej cztery), z których następnie obliczana jest średnia arytmetyczna (wzór 6.1). W kolejnym kroku odczyt jest normalizowany (wzór 6.2) – mechanizm jest wymagany, kiedy pozycja wahę się między 359-tym, a zerowym stopniem.

$$pos = \frac{a_0 + a_1 + a_2 + \dots + a_n}{n} \quad (6.1)$$

$$pos = \begin{cases} pos + 360 & : pos < 0 \\ pos - 360 & : pos > 360 \end{cases} \quad (6.2)$$

- pos : absolutna pozycja enkodera [deg],
- n : ilość próbek do uśrednienia.

Zbyt duża ilość odczytów nie jest opłacalna, gdyż błąd odczytu przestaje znaczco maleć (wzór 6.3).

$$x = \frac{1}{\sqrt{n}} \quad (6.3)$$

- x : szum enkodera (max. 0.06°),
- n : ilość próbek do uśrednienia.

6.3.2. Filtr Kalmana

Zastosowany algorytm jest metodą filtracji dynamicznej. Posługując się tym narzędziem, można wyznaczyć pomiarowo niedostępne zmienne jedynie na podstawie bieżących wartości wielkości pomiarowo dostępnych oraz znajomości modelu matematycznego łączącego ze sobą obydwie te grupy pomiarów.

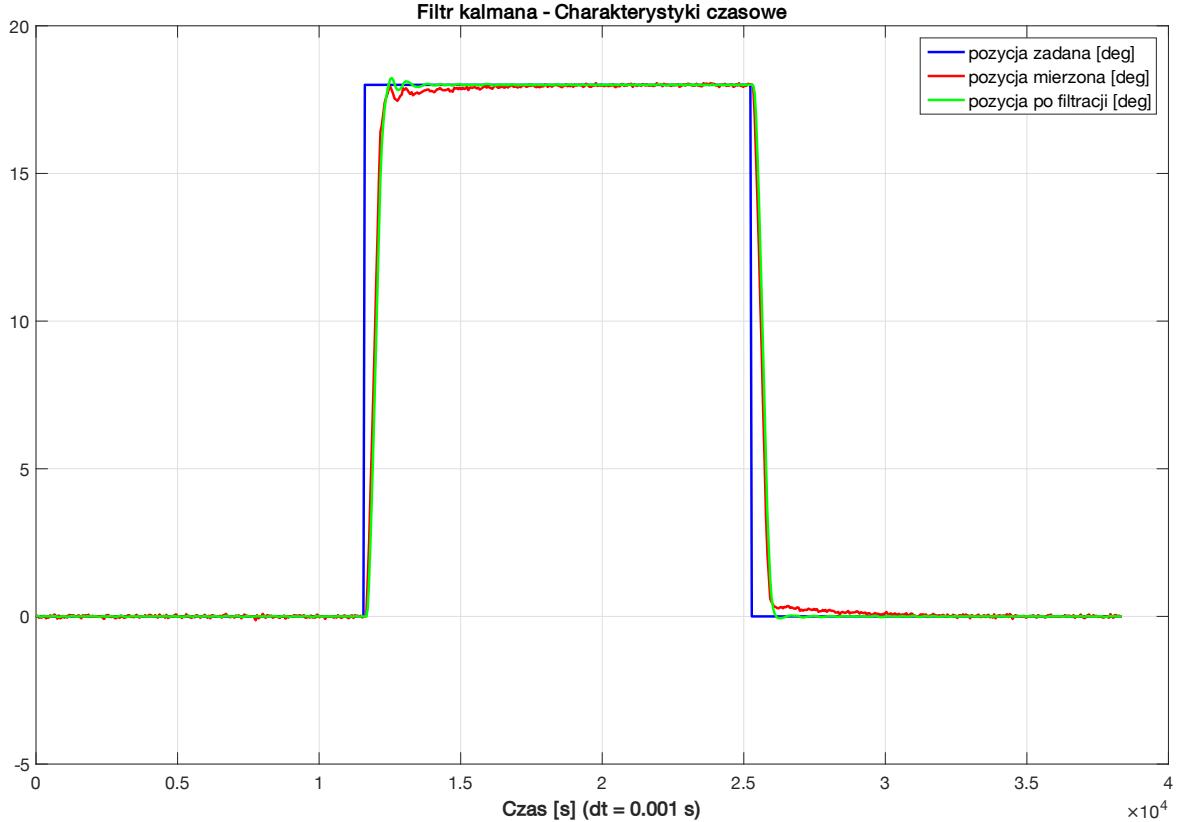
Proces można przedstawić za pomocą dyskretnego modelu w przestrzeni stanu:

$$x(t+1) = Ax(t) + Bu(t) + v(t) \quad (6.4)$$

$$y(t) = Cx(t) + w(t) \quad (6.5)$$

gdzie:

- $x(t)$: stan w chwili czasu $t = 0, 1, \dots$,
- $y(t)$: wyjście układu,
- A : macierz stanu,
- B : macierz wejścia,
- C : macierz wyjścia,
- $v(t)$: szum procesowy,
- $w(t)$: szum pomiarowy.



Rysunek 6.6: Przebieg czasowy prezentujący pozycję wału silnika kroakowego: zadaną, mierzoną i mierzoną przefiltrowaną przez KF

Źródło: Na podstawie [33]

Zastosowany filtr jest dwuwymiarowy. Oznacza to że estymacji są poddawane dwa parametry: pozycja i prędkość. Model stanowy wygląda następująco:

$$x = \begin{bmatrix} pos \\ vel \end{bmatrix} \quad (6.6)$$

$$A = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \quad (6.7)$$

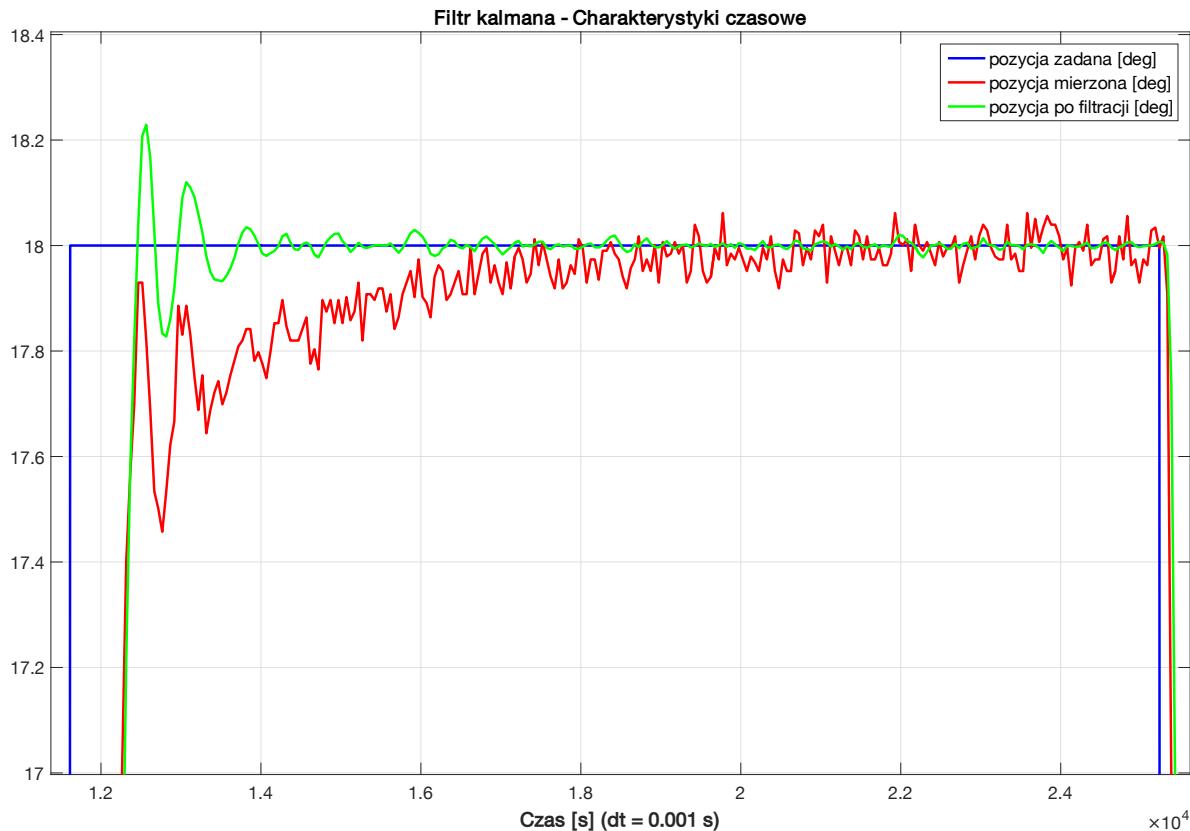
$$B = \begin{bmatrix} dt \\ 0 \end{bmatrix} \quad (6.8)$$

$$C = [1 \ 0] \quad (6.9)$$

Implementacja dwuwymiarowa zapewnia dokładniejszą estymatę pozycji niż w przypadku, kiedy pod uwagę brane byłoby tylko położenie. Jeszcze lepsze wyniki

możnaby otrzymać stosując estymatę trójwymiarową (dodatkowo dodać parametr przyspieszenia). Niestety operacje na macierzach 3×3 pochłaniały za dużo zasobów procesora i z tego rozwiązania w sterowniku zrezygnowano.

Rys. 6.6 prezentuje różne pozycje kątowe w funkcji czasu odczytane ze sterownika w trybie CSV (patrz sekcja 6.1.2). Wstępnie widać, że algorytm działa poprawnie, lecz jego zalety i wady można dokładniej zaobserwować na przebiegu z rys. 6.7.



Rysunek 6.7: Przebieg czasowy (powiększenie przy impulsie skokowym) prezentujący pozycję wału silnika krokowego: zadaną, mierzoną i mierzoną przefiltrowaną przez KF

Źródło: Na podstawie [33]

Po doświadczalnym dostosowaniu parametrów v i w algorytm tłumii zakłócenia bardzo dobrze. Charakterystyczną cechą filtra Kalmana jest jednak to, że wraz ze wzrostem poziomu filtracji rośnie także opóźnienie sygnału. Parametr ten diametralnie wpływa na działanie regulatorów pozycji i prędkości (sekcja 6.4.1). Pomimo tego, że estymowana pozycja jest bardzo blisko zadanej, to czas potrzebny na jej osiągnięcie (w końcowym odcinku) jest zbyt długi.

Z tego powodu został zaimplementowany dodatkowy, szybszy algorytm stosowany w czasowo krytycznych sytuacjach.

6.3.3. Filtr dolnoprzepustowy

Drugim algorytmem stosowanym w celu minimalizacji wpływu szumów towarzyszących odczytywanej pozycji kątowej z enkodera magnetycznego jest rekursywny (lub też *NOI* – o nieskończonej odpowiedzi impulsowej), jednobiegunowy filtr dolnoprzepustowy. Zastosowane zostały jedynie dwa współczynniki rekursywne a_0 i b_1 :

$$a_0 = 1 - x \quad (6.10)$$

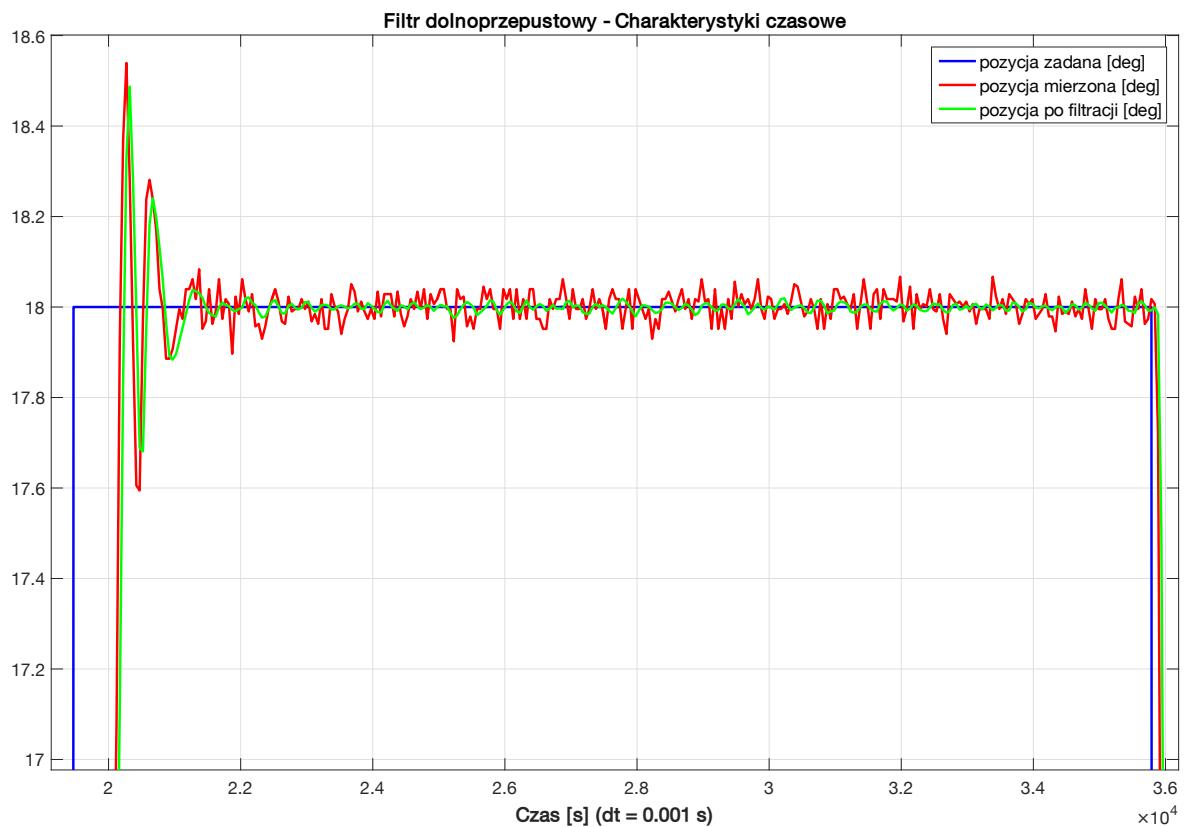
$$b_1 = x \quad (6.11)$$

Współczynnik x wskazuje jakie jest opadanie przebiegu mierzonego między kolejnymi próbками. Wartość ta jest obliczana na podstawie stałej czasowej filtra:

$$x = \exp^{-2\pi f_{CR}} \quad (6.12)$$

f_{CR} jest częstotliwością odcięcia dobieraną w sterowniku doświadczalnie (obecnie 20 Hz). Kolejne próbki na wyjściu filtra obliczane są według wzoru 6.13

$$\text{output}(t) = \text{input}(t)a_0 + \text{input}(t-1)b_1 \quad (6.13)$$



Rysunek 6.8: Przebieg czasowy (powiększenie przy impulsie skokowym) prezentujący pozycję wału silnika krokowego: zadaną, mierzoną nieprzefiltrowaną i mierzoną przefiltrowaną przez filtr dolnoprzepustowy trzeciego rzędu

Źródło: Na podstawie [33]

Rząd filtru jest ustalany przez użytkownika. Rys. 6.8 prezentuje odczyt pozycji wału silnika krokowego. Algorytm ten działa szybciej od zaimplementowanego filtru Kalmana – obrót o 18° w ok. 1 s przy zastosowaniu KF i ok. 170 ms przy filtrze dolnoprzepustowym trzeciego rzędu. Ponadto algorytm jest bardzo prosty i kod potrzebny do jego wykonania nie wymaga dużej ilości zasobów mikrokontrolera (listing 6.3).

Listing 6.3: Kod filtra dolnoprzepustowego wykonywany w pętli sterującej mikrokontrolera w interwale $dt = 0.001s$

```

1  /*
2   * @brief      This function should be called each \ref dt.
3   * @param      input: Filter input.
4   *
5   * return      Filter output.
6   */
7 float lowpass_Tick(lowpass_t* filter, float input)
8 {
9     assert_param(filter);
10    assert_param(filter->order);
11
12    if (filter->order > MAX_ORDER_NR)
13        filter->order = MAX_ORDER_NR;
14    else if (filter->order < 1)
15        filter->order = 1;
16
17    uint32_t i;
18    for (i = 0; i < filter->order; i++)
19    {
20        input = input * filter->a0 +
21                filter->prevOutput[i] * filter->b1;
22        filter->prevOutput[i] = input;
23    }
24
25    return input;
26 }
```

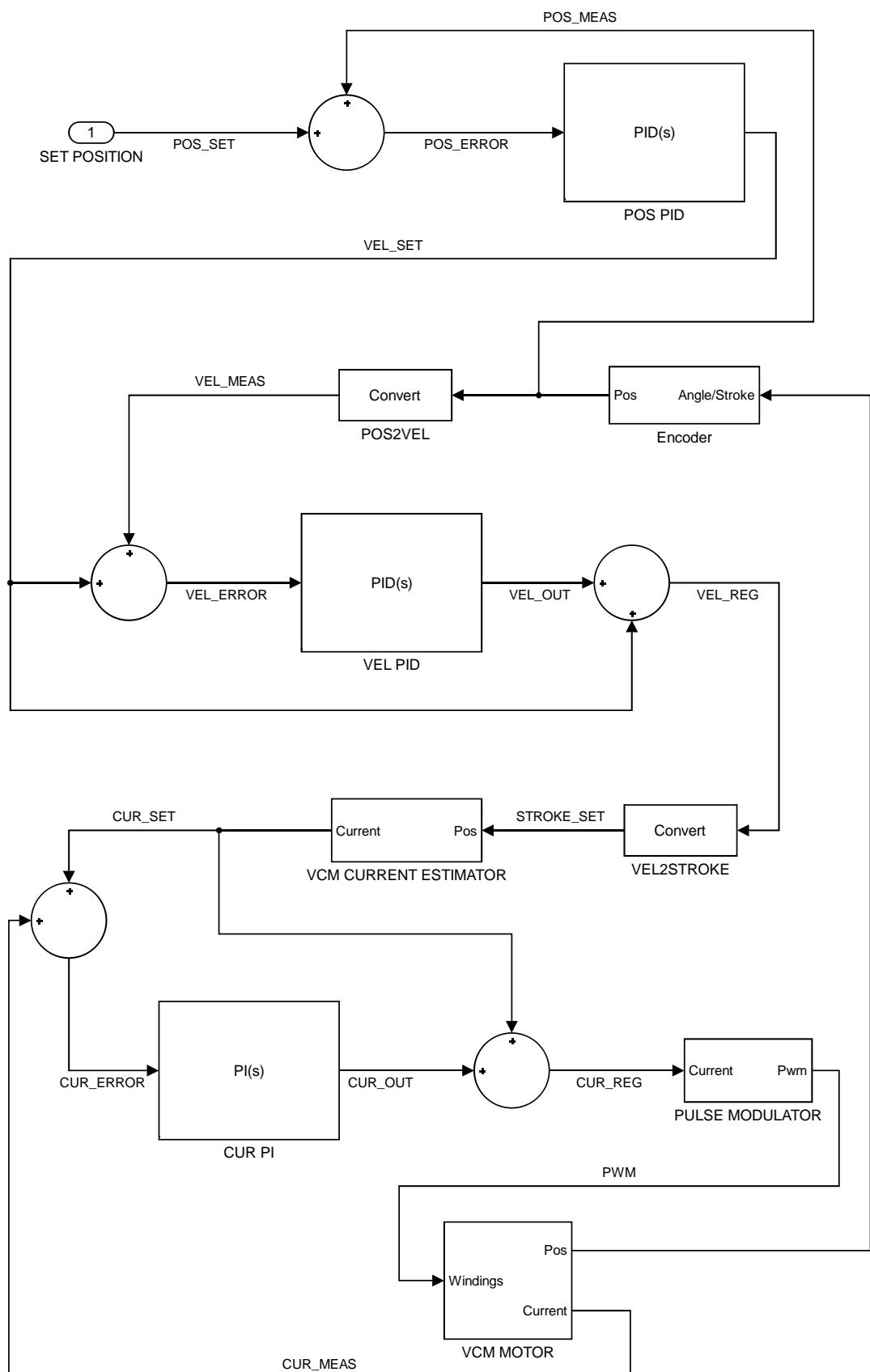
Źródło: Na podstawie [31]

Na chwilę obecną uśrednianie wraz z filtrem dolnoprzepustowym są algorytmami dającymi najlepsze wyniki działania enkodera.

6.4. Procedury sterowania wysokopoziomowego

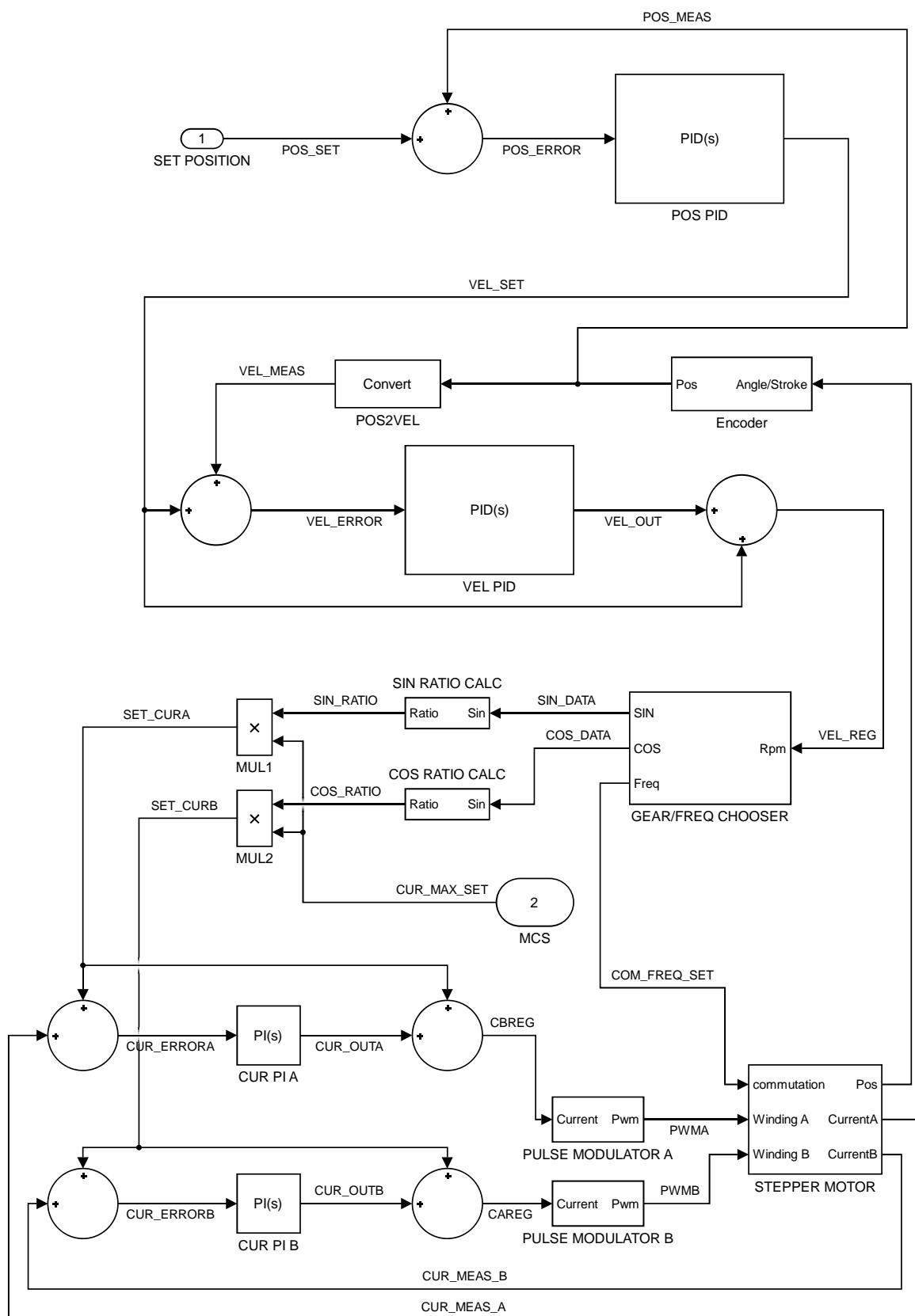
Sterowanie wyższego poziomu w kontrolerze polega na manipulacji, czy też operacjach na danych, które nie mają jeszcze bezpośredniego wpływu prądem w uzwojeniach czy pozycję kątową wału silnika. Odzwierciedlają one jedynie różne wielkości, które następnie są przekazywane do niższej warstwy oprogramowania, która steruje silnikami (6.5).

Procedury sterowania wysokopoziomowego dla obu rodzajów silników (krokowy i VCM) w większości są identyczne. Odmienne wyglądają moduły sterowania prądem w uzwojeniach, ze względu na odmienną konstrukcję silników. Rys. 6.9 prezentuje poglądowy diagram sterowania silnika VCM, a rys. 6.10 silnika krokowego.



Rysunek 6.9: Schemat blokowy sterowania wysokopoziomowego silnikiem VCM, za-implementowanego w kodzie sterownika

Źródło: Na podstawie [33]



Rysunek 6.10: Schemat blokowy sterowania wysokopoziomowym silnikiem kroko-wym, zaimplementowanego w kodzie sterownika

Źródło: Na podstawie [33]

6.4.1. Regulator pozycji i prędkości

W celu zapewnienia zadanej pozycji (kątowej w przypadku silnika krokkowego i liniowej przy VCM) w sterowniku został zastosowany regulator pozycji typu PID–proporcjonalno-całkującą–różniczkującą (*proportional–integral–derivative*). Jego zasadę działania wyraża wzór 6.14.

$$s(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (6.14)$$

Gdzie:

- $s(t)$: pozycja w danej chwili t [m],
- $e(t)$: błąd pozycji $e(t) = v(t) - v(t - 1)$ [m],
- K_p : współczynnik proporcjonalny [m],
- K_i : współczynnik całkujący [$m \cdot s$],
- K_d : współczynnik różniczkujący [$\frac{m}{s}$].

Implementację algorytmu w kodzie mikrokontrolera przedstawia listing 6.4.

Listing 6.4: Kod regulatora pozycji zaimplementowany w mikrokontrolerze, wywoływany w interwale $dt = 0.001s$

```

1  /*
2   * @brief      Regulates the output position,
3   *             taking under consideration current pos and desired pos.
4   *             This function is universal for all type of motors.
5   *
6   * @param      motor: pointer to a motor structure.
7   * @param      currentPos: Current position expressed as angle or position.
8   * @param      desiredPos: Desired position expressed as angle or position.
9   */
10  float pid_PosController(motor_t* motor, float currentPos, float desiredPos)
11 {
12     assert_param(motor);
13
14     float error = desiredPos - currentPos;
15     motor->posPid.integral += (error * motor->posPid.dt);
16
17     if (motor->posPid.integral >= motor->posPid.antiWindUp)
18         motor->posPid.integral = motor->posPid.antiWindUp;
19     else if (motor->posPid.integral < (motor->posPid.antiWindUp * -1))
20         motor->posPid.integral = motor->posPid.antiWindUp * -1;
21
22     float derivative = (error - motor->posPid.prevErr) / motor->posPid.dt;
23     float output = motor->posPid.kp * error +
24                 motor->posPid.ki * motor->posPid.integral +
25                 motor->posPid.kd * derivative;
26
27     motor->posPid.prevErr = error;
28     return output;
29 }
```

Źródło: Na podstawie [31]

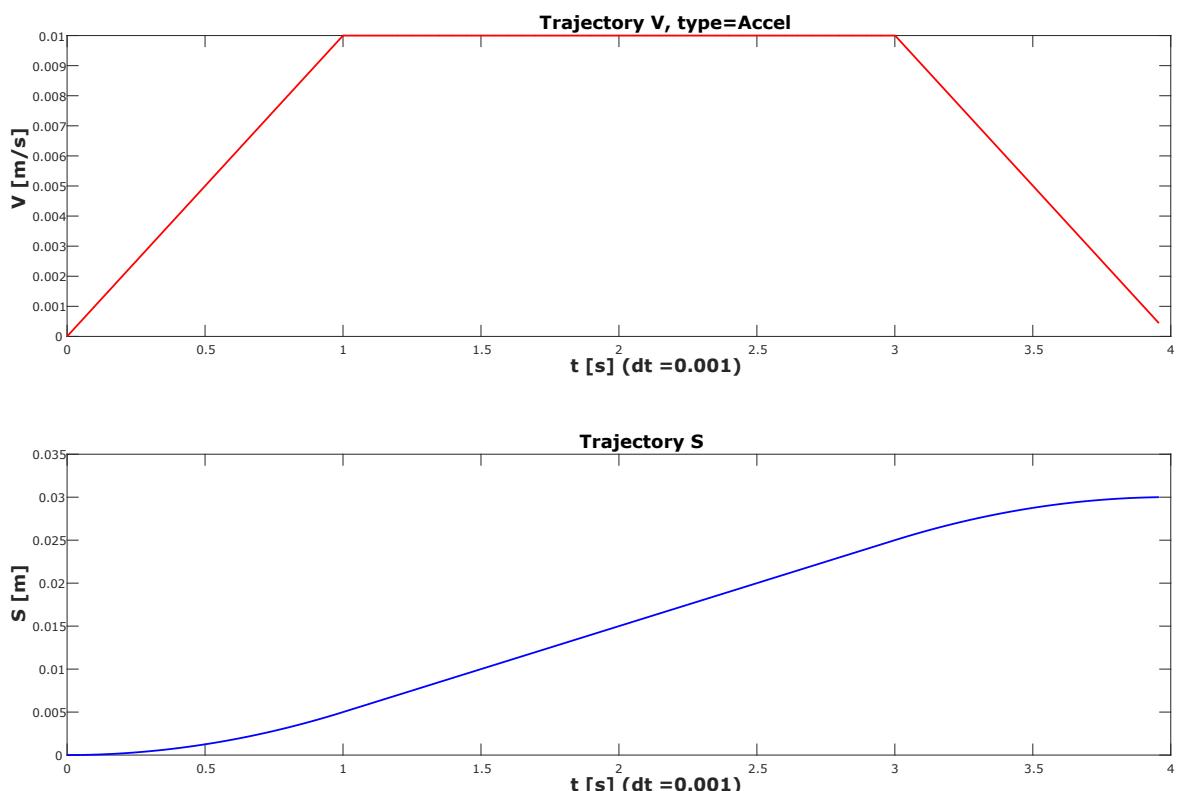
Funkcja zawiera dodatkowo mechanizm zabezpieczający regulator przed zamagazynowaniem zbyt dużego błędu pozycji, co może spowodować przeregulowanie

układu („Anti-Windup”). Jako parametry wejściowe podawane są aktualna pozycja (odczytana z enkodera) i pozycja zadana. Wyjście regulatora pozycji jest następnie podawane na wejście regulatora prędkości (6.9).

Regulator prędkości działa w ten sam sposób co regulator pozycji, z tą różnicą że sygnał wyjściowy z regulatora jest wielkością tego samego typu co sygnał wejściowy (w uproszczeniu, bo tylko regulator typu P daje na wyjściu sygnał tej samej wielkości co na wejściu). Suma (lub różnica w zależności od znaku zmiennej zwracanej z funkcji) tych dwóch sygnałów tworzy regułowaną prędkość zadaną. Zastosowanie oddzielnego regulatora dla pozycji i prędkości pozwala w bardzo precyzyjny sposób konfigurować pracę danego silnika przy pomocy współczynników K_p , K_i i K_d (*tunning*). Kolejne bloki różnią się w zależności od zastosowanego silnika.

6.4.2. Generator trajektorii

W celu większej kontroli nad zadanym położeniem i prędkością, oraz aby usprawnić ogólną pracę silników w procedurze sterowania został dodany generator trajektorii. Moduł ten ma za zadanie doprowadzić element czynny maszyny do zadanej pozycji w optymalny sposób.



Rysunek 6.11: Wygenerowana trajektoria w kształcie trapezu

Źródło: Na podstawie [33]

Sposób optymalny w rozumieniu algorytmu jest to próba uzyskania przebiegu prędkości w zadanym odcinku w kształcie trapezu, co pozwala na płynny start i hamowanie (rys. 6.11). W celu oszczędzenia pamięci operacyjnej mikrokontrolera

kolejne punkty trajektorii są liczone na bieżąco podczas „podążania” za trajektorią. Do jej wyznaczenia potrzebne są następujące dane:

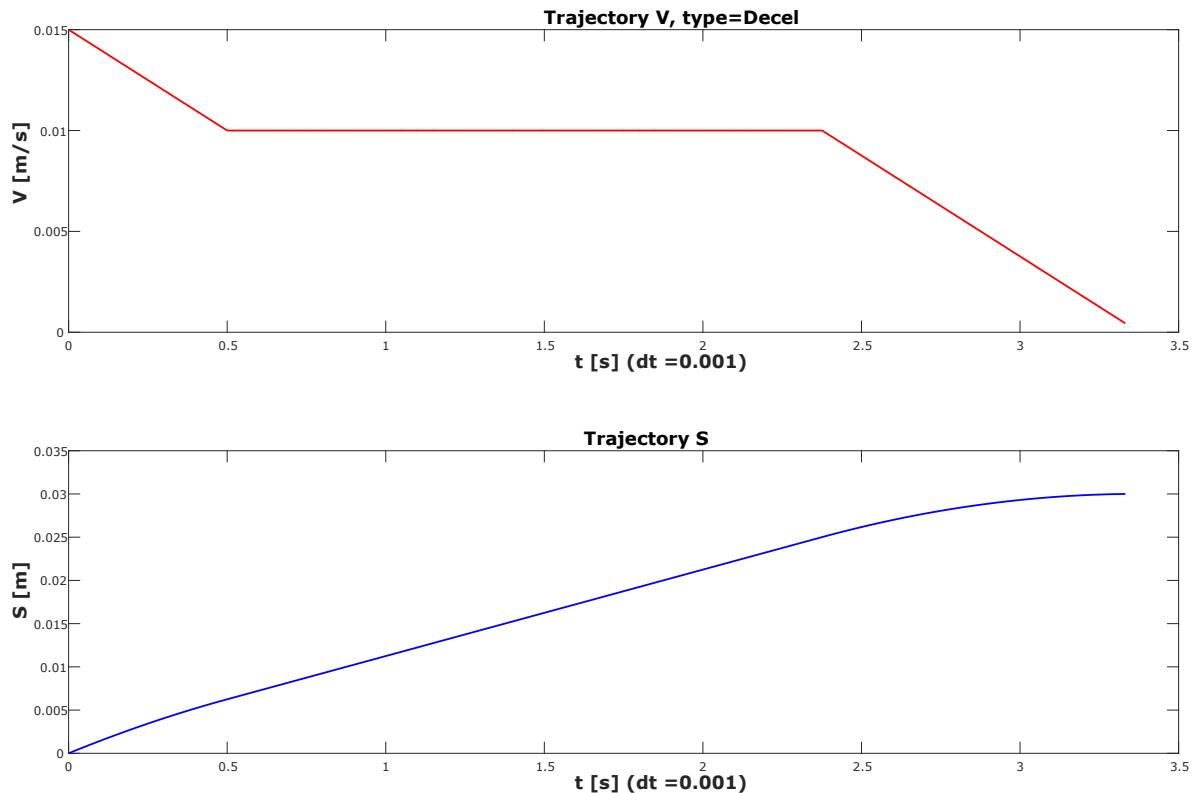
- v_{max} : Maksymalna prędkość do jakiej element czynny może być rozpędzony. Jednostka: $[\frac{m}{s}]$ lub $[\frac{deg}{s}]$,
- a_{max} : Maksymalne przyspieszenie/ opóźnienie. Jednostka: $[\frac{m}{s^2}]$ lub $[\frac{deg}{s^2}]$,
- s : Zadana pozycja. Jednostka: $[m]$ lub $[deg]$.

Każdy obrót głównej pętli sterującej dostarcza regulatorom kolejnych danych wejściowych. W pierwszej kolejności obliczana jest prędkość. W zależności od tego czy aktualne układ przyspiesza, porusza się ze stałą prędkością lub zwalnia, stosowany jest odpowiedni wzór 6.15.

$$v_{set} = \begin{cases} v_0 - \int_{t_0}^t a_{max} dt & : type = decel \\ v_{max} & : type = const \\ v_0 + \int_{t_0}^t a_{max} dt & : type = accel \end{cases} \quad (6.15)$$

Gdzie t oznacza aktualny okres pętli. Korzystając z uzyskanej chwilowej prędkości (liniowej lub kątowej w zależności od silnika) obliczana jest pozycja (6.16).

$$s_{set} = \begin{cases} \int_{t_0}^t v_0 dt - \int_{t_0}^t (v_0 - v_{set}(t)) dt & : type = decel \\ s_{set}(t-1) + v_{set}(t) dt & : type = const \\ \int_{t_0}^t v_0 dt + \int_{t_0}^t (v_{set}(t) - v_0) dt & : type = accel \end{cases} \quad (6.16)$$

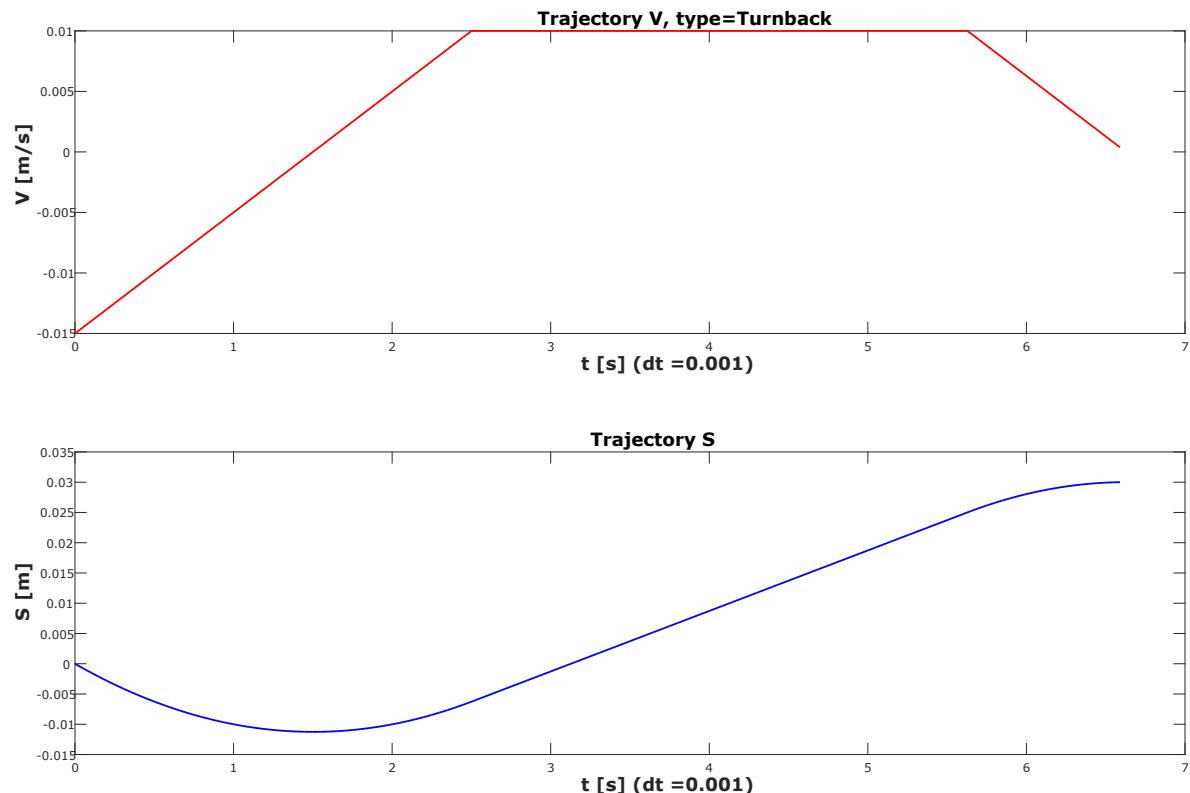


Rysunek 6.12: Wygenerowana trajektoria dla obiektu poruszającego się ($v_0 > v_{max}$)

Źródło: Na podstawie [33]

W przypadku kiedy kontroler odbiera informację o zmianie pozycji, a aktualnie jest w ruchu ($v_{set} \neq 0$), następuje aktualizacja trajektorii. Przykład dla którego v_{max} nowej trajektorii jest mniejsza od aktualnej prezentuje rys. 6.12.

Algorytm radzi sobie także z sytuacją, w której aktualizowana pozycja wymaga zmiany kierunku jazdy (rys. 6.13), lub maksymalna prędkość nie może być osiągnięta ze względu na zbyt małą drogę zapewnioną na rozped (lub zbyt małe maksymalne przyspieszenie). Wtedy charakterystyka prędkości przyjmuje kształt trójkąta zamiast trapezu.



Rysunek 6.13: Wygenerowana trajektoria dla obiektu poruszającego się w przeciwnym kierunku

Źródło: Na podstawie [33]

Po osiągnięciu zadanej pozycji regulatory działają już bez udziału generatora trajektorii, próbując zachować odpowiednio docelową pozycję i prędkość dążącą do zera.

6.4.3. Sterowanie prędkością kątową silnika krokowego

W modelu sterowania, który został zaimplementowany w oprogramowaniu kontrolera, parametrami decydującymi o aktualnej prędkości kątowej silnika krokowego są:

- częstotliwość komutacji,
- tryb pracy („bieg”).

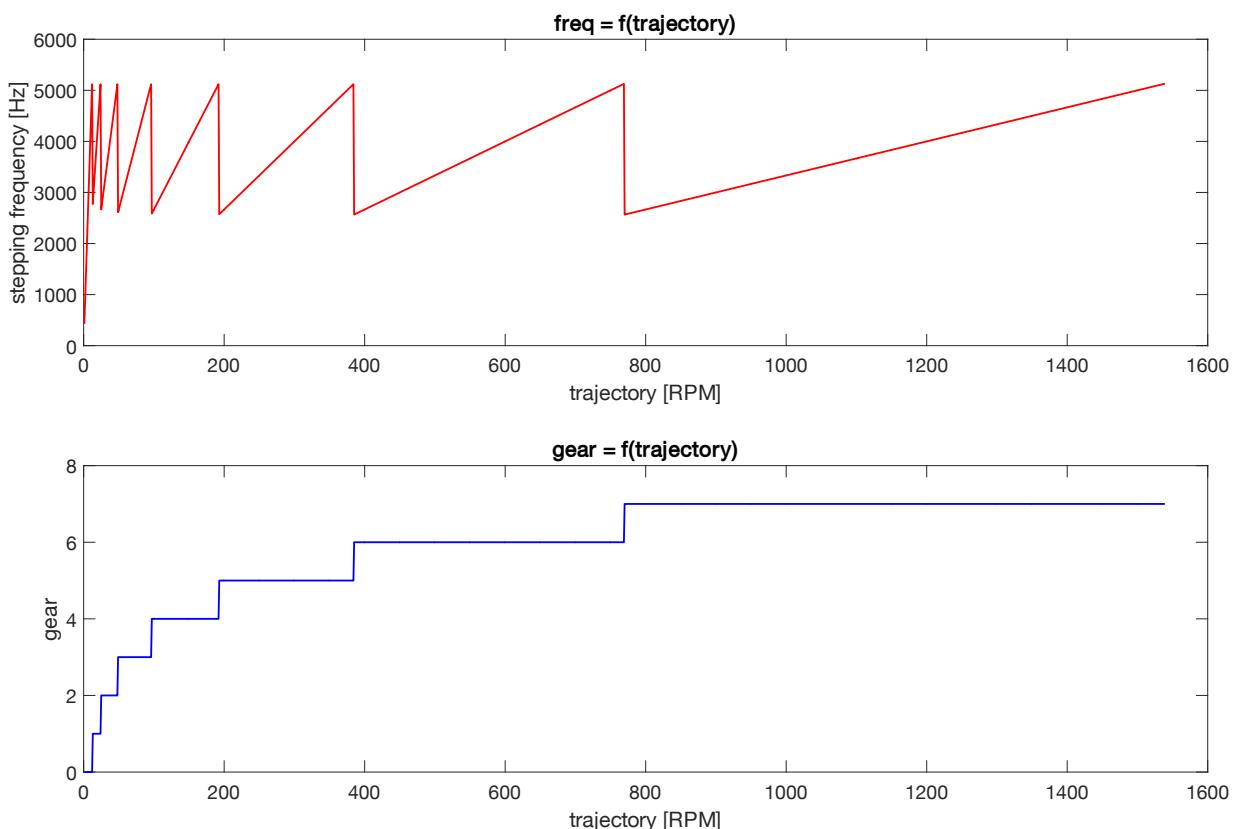
Wartości wymienionych parametrów są automatycznie obliczane w pętli sterowania dla zadanej prędkości obrotowej. W celu inicjalizacji algorytmu należy podać następujące parametry:

- ilość pełnych kroków na obrót dla danego silnika,
- ilość biegów (maksymalna dopuszczalna rozdzielczość pracy mikrokrokowej),
- maksymalna częstotliwość komutacji [Hz],
- maksymalna użyteczna prędkość kątowa dla danego silnika [RPM].

Relacja między ilością biegów i rozdzielczością przy pracy mikrokrokowej jest następująca:

$$gearsNr = \log_2(resolution) + 1 \quad (6.17)$$

W pierwszej kolejności algorytm oblicza wartości progowe prędkości obrotowych dla których następuje „automatyczna zmiana biegu”. Wartości te są nieco inne dla przypadku w którym silnik jest rozpędzany i hamowany, dzięki czemu bieg nie jest zmieniany zbyt często przy wartościach skrajnych (działanie bazujące na histerezie, podobnie jak w przerzutniku *Shmitta*).



Rysunek 6.14: Wizualizacja algorytmu obliczającego częstotliwość komutacji i bieg dla zadanej prędkości obrotowej wału

Źródło: Na podstawie [33]

Dla przypadku rozpędzania wirnika z bezruchu, wirtualna przekładnia rozpoczęta pracę przy największej rozdzielczości mikrokroków. Wraz ze wzrostem prędkości obrotowej częstotliwość komutacji jest zwiększana. Im większa częstotliwość komutacji, tym częściej system musi przerywać pozostałe prace, dlatego maksymalną

częstotliwość należy dobrać w taki sposób, aby nie zagłodzić pozostałych procesów w systemie.

Po osiągnięciu maksymalnej częstotliwości następuje zmiana biegu, czyli dwukrotne zmniejszenie rozdzielczości mikrokroków. Częstotliwość komutacji także dwukrotnie maleje. Działanie to jest powtarzane aż do osiągnięcia maksymalnej ustalonej prędkości obrotowej (i jednocześnie maksymalnego biegu i częstotliwości komutacji). Algorytm pozwala zoptymalizować pracę silnika poprzez korzystanie z największej rozdzielczości tylko wtedy, kiedy jest potrzebna (rozpedzanie wirnika i precyzyjne hamowanie). Rys. 6.14 prezentuje w graficzny sposób działanie układu.

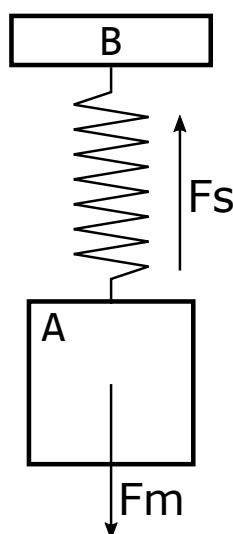
Można zauważyć, że praca na najwyższym biegu (1 mikrokrok na 1 pełny krok) to tak naprawdę tryb falowy (sekcja 1.4.2, rys. 1.14). Układ przechodzi do niego w naturalny sposób z trybu mikrokrokowego. W aplikacji takiej jak głowica układająca duży moment obrotowy nie jest wymagany, dlatego nie ma potrzeby implementacji trybu pełnokrokowego na ostatnim biegu. Docelowo jednak, funkcjonalność ta będzie musiała być dodana (np. sterowanie ruchu portalu w osi X i Y).

6.4.4. Regulator prądu silnika VCM

Wartość prądu zadanego dla uzwojenia maszyny VCM uzyskuje się poprzez jego estymatę, biorąc pod uwagę właściwości fizyczne układu napędowego (blok *VCM CURRENT ESTIMATOR* na schemacie 6.9). Na wejście estymatora trafia wartość absolutnego (względem obudowy), aktualnego wysuwu cylindra silnika. Zadana wartość wysuwu jest uzyskiwana poprzez pomnożenie stałej czasowej układu przez scałkowaną po czasie prędkość z jaką porusza się cylinder (6.18).

$$stroke(t) = \int_0^t v(t)dt \quad (6.18)$$

Do obliczenia zadanego prądu dla silnika VCM należy się posłużyć modelem fizycznym z rys. 6.15.



Rysunek 6.15: Teoretyczny model układu napędowego modułu z silnikiem VCM

Źródło: Na podstawie [33]

Obiekt A reprezentuje część ruchomą głowicy, czyli wszystkie komponenty znajdujące się po stronie cylindra (rys. 5.1). B to część nieruchoma, po stronie której znajduje się obudowa silnika. W pierwszej kolejności należy obliczyć działającą na ciało A siłę sprężyny F_s (6.19):

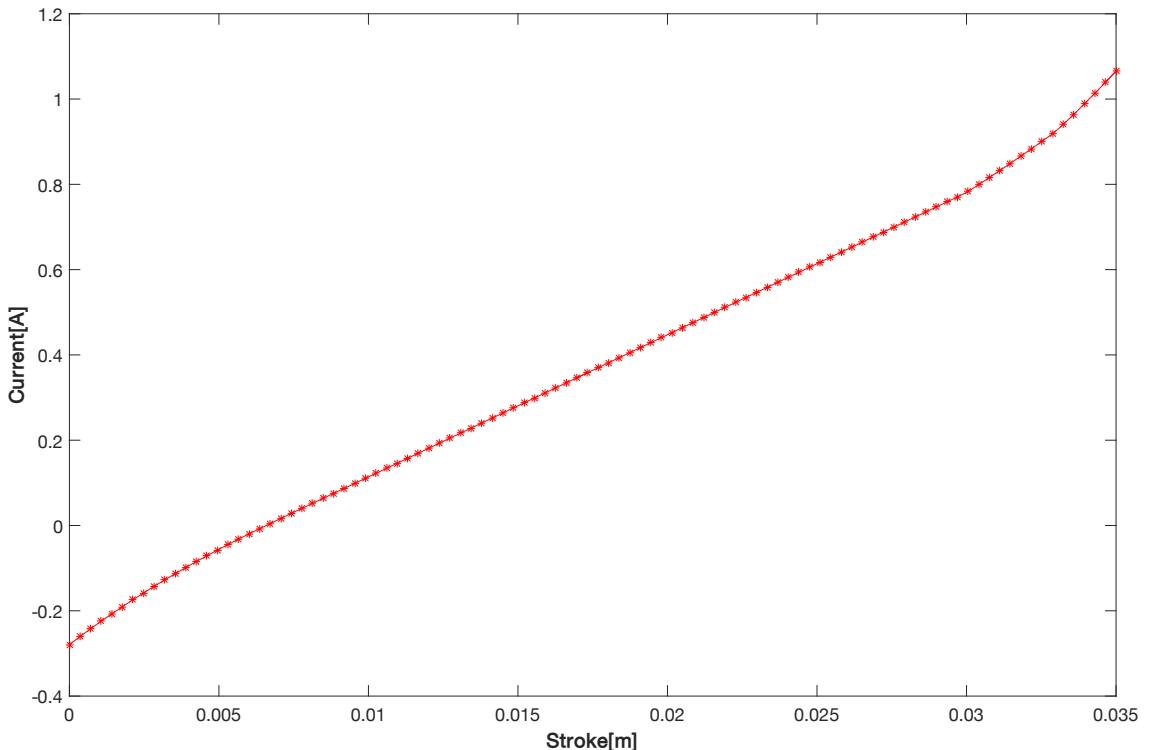
$$F_s = F_{s0} + R_s \cdot S \quad [N = N + \frac{N}{m} \cdot m] \quad (6.19)$$

- F_{s0} : siła początkowa sprężyny [N],
- R_s : stała sprężyny [$\frac{N}{m}$],
- S : wysuwanie cylindra [m].

Następnie należy obliczyć siłę F_m z jaką oddziałuje na ciało grawitacja (6.20):

$$F_m = m_A \cdot g \cdot \sin \alpha \quad [N = kg \cdot \frac{m}{s^2} \cdot 1] \quad (6.20)$$

- m_A : sumaryczna masa części ruchomej głowicy [kg],
- g : przyspieszenie ziemskie [$\frac{m}{s^2}$],
- α : kąt ustawienia głowicy względem płaszczyzny (normalnie 90°) [deg].



Rysunek 6.16: Przebieg teoretycznego prądu (osi Y) wymaganego w uzwojeniu do wysuwanego cylindra na zadaną odległość (osi X)

Źródło: Na podstawie [33]

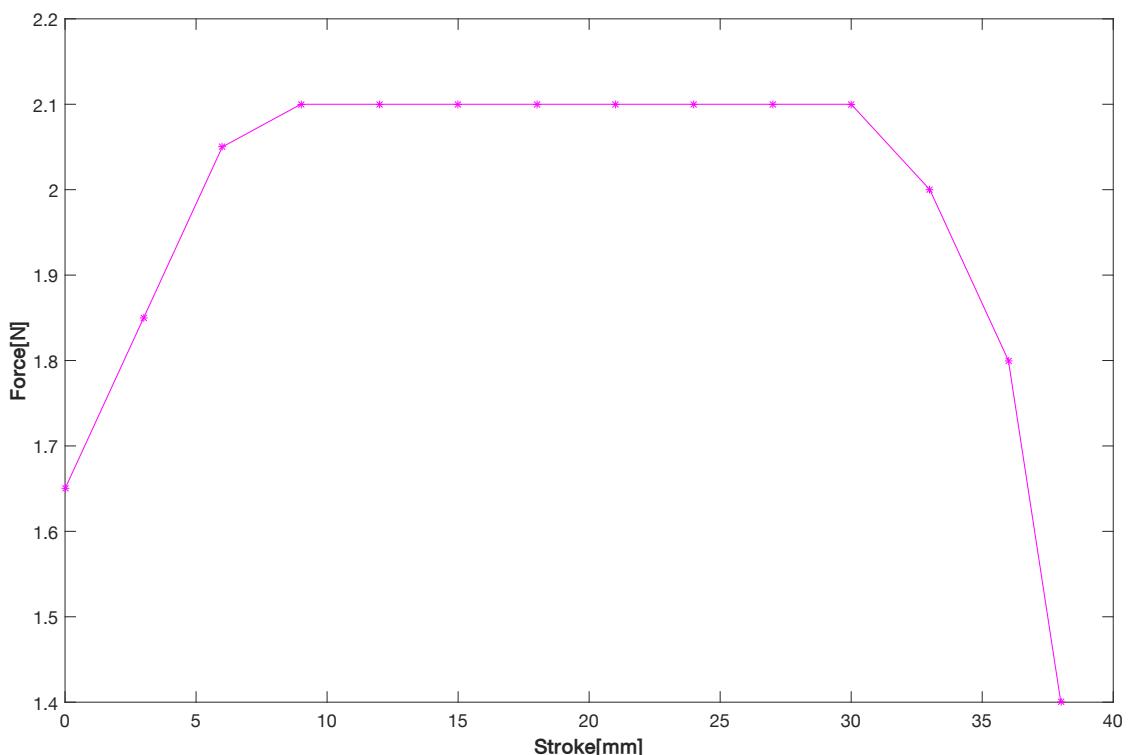
Potem liczona jest wypadkowa siła F (6.21) działająca na ciało A . Im cięższa jest część ruchoma głowicy, tym bardziej jest wysunięty cylinder w stanie spoczynku (bezprądowym).

$$F = F_s - F_m \quad [N = N - N] \quad (6.21)$$

Ostatnim krokiem jest odczytanie stałej silnika VCM (F_c – *force constant*) z charakterystyki dostarczonej przez producenta (np. 1.6) i obliczenie prądu w uzwojeniu dla wymaganego wysuwu cylindra (6.21):

$$I = \frac{F}{F_c} \quad [A = \frac{N}{\frac{N}{A}}] \quad (6.22)$$

Na rys. 6.16 widać wynik symulacji w postaci prądu w funkcji wysułu cylindra. Obliczenia zostały przeprowadzone przy rzeczywistych wartościach parametrów układu napędowego. Wykres wskazuje na to, że w stanie spoczynku część ruchoma głowicy jest wysunięta na odległość ok. 6.7 mm. Charakterystyka ma charakter liniowy w swojej środkowej części, ze względu na kształt charakterystyki prądowej silnika wprowadzonej do algorytmu jako parametr (6.17).



Rysunek 6.17: Charakterystyka prądowa zastosowanego silnika VCM odtworzona w programie używając charakterystyki z karty katalogowej

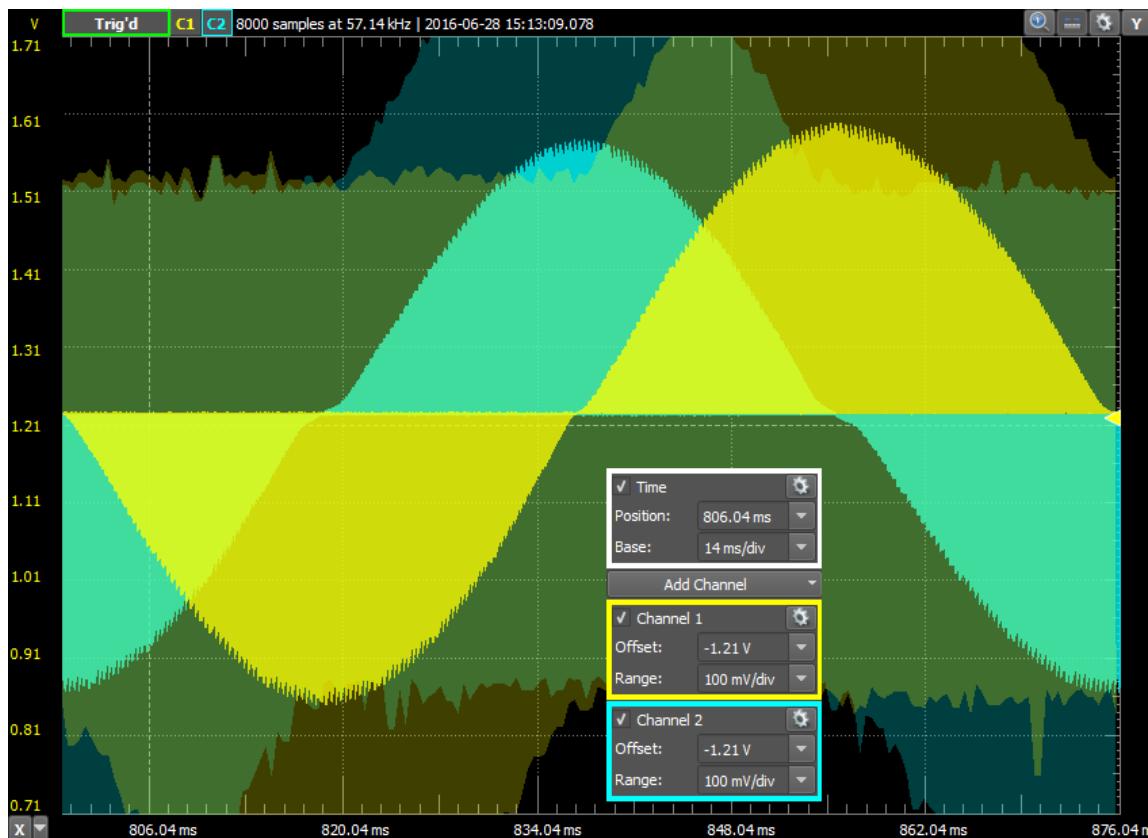
Źródło: Na podstawie [33]

Wyznaczony prąd zadany trafia do właściwego regulatora prądu typu PI. jego realizacja różni się od poprzednich regulatorów tym, że nie ma już członu różnicującego, który w przypadku regulacji prądu nie dał by wielu korzyści. Ponadto stała czasowa pętli prądowej wynosi $100\mu s$.

Wartość prądu jest następnie zamieniana na odpowiednie wypełnienie impulsów w uzwojeniu– więcej informacji na ten temat jest zawarte w sekcji 6.5.

6.4.5. Regulator prądu silnika krokowego

Regulatory prądu (dla uzwojeń A i B) w przypadku silnika krokowego próbują przez cały okres pracy maszyny utrzymywać w uzwojeniach taki prąd, aby spełnione było równanie 1.1.



Rysunek 6.18: Oścylogramy mierzonych spadków napięć na rezystorach pomiarowych w mostkach H silnika krokowego poruszającego się ze stałą zadaną prędkością kątową

Źródło: Na podstawie [33]

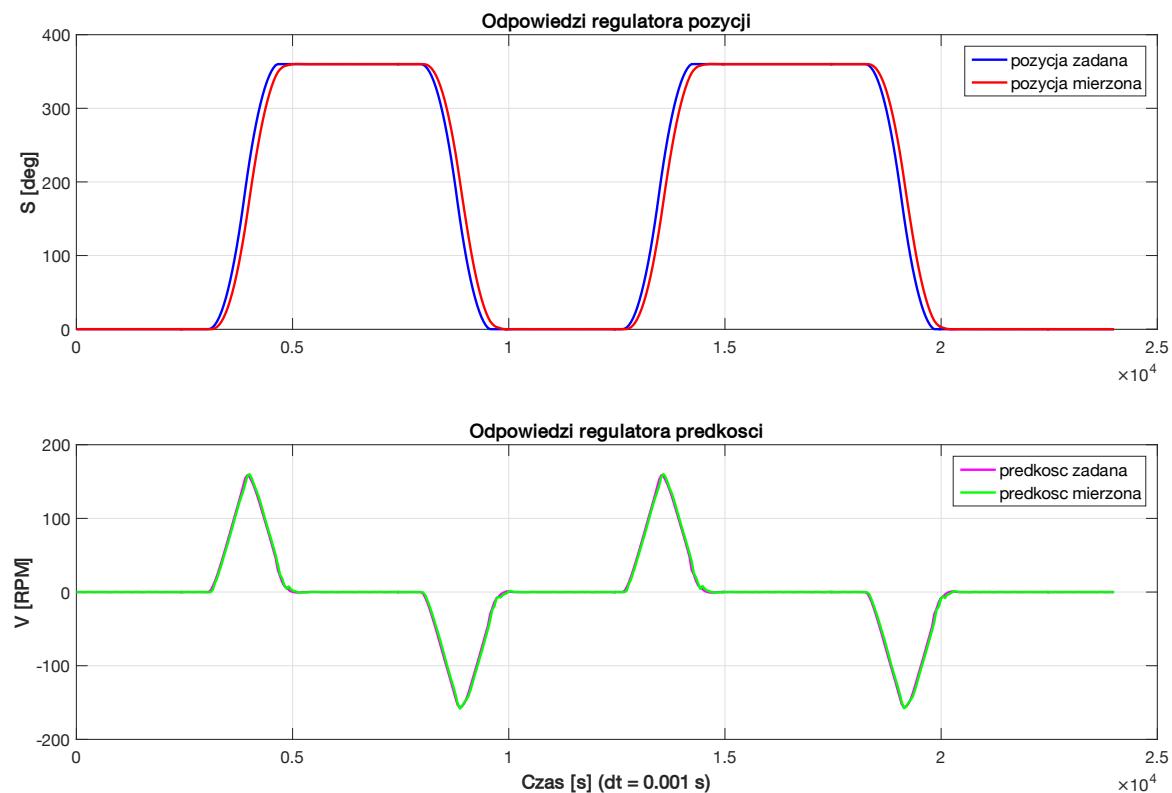
Prąd maksymalny ($\theta = 90^\circ$) jest przemnażany przez odpowiednie współczynniki (zmienne dla danej pozycji) dla obu uzwojeń (rys. 6.10). Iloczynami są zadane prądy uzwojeń, które następnie trafiają do wspomnianych regulatorów. Poprawność działania układu potwierdza oścylogram 6.18.

6.4.6. Sprawdzenie systemu sterowania

W celu zaprezentowania poprawności działania systemu sterowania wykonano kilka przejazdów (dla obu silników) i zarejestrowano zmienne operacyjne w czasie rzeczywistym używając modułu logger'a (sekcja 6.1.2).

Pierwszy przebieg (rys. 6.19) prezentuje odpowiedzi regulatorów pozycji oraz prędkości na czterokrotną zmianę pozycji kątowej wału silnika krokowego o 180° . Do uzyskania zadanych pozycji posłużył generator trajektorii, który w każdym okresie głównej pętli sterowania oblicza kolejne wartości docelowe dla regulatorów.

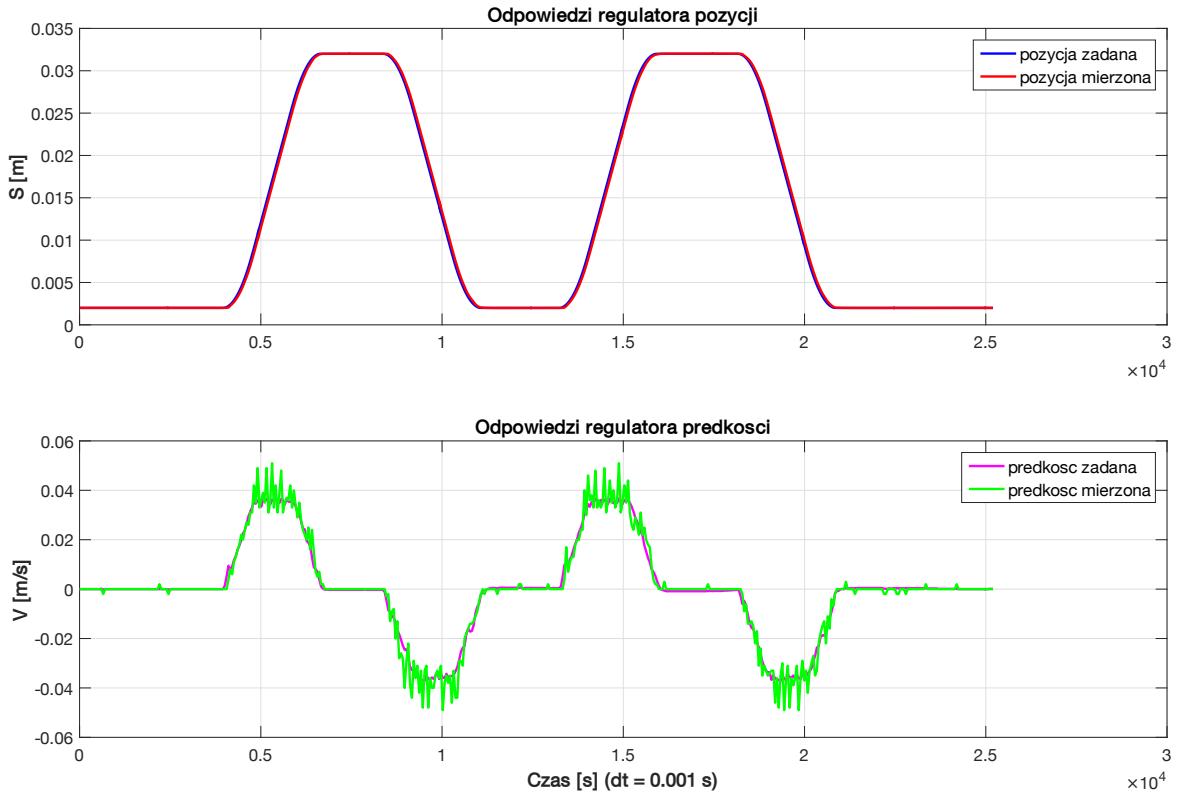
Na wykresie pozycji w funkcji czasu z rys. 6.19 można zaobserwować niewielkie opóźnienie sygnału mierzonego względem zadanego. Wynika ono z faktu zastosowania algorytmów filtrujących w module enkodera (sekcja 6.3). Wykres prędkości jest przykładem trajektorii typu trapezoidalnego, w której nie udało się osiągnąć maksymalnej możliwej prędkości, stąd kształt jest trójkątny.



Rysunek 6.19: Przykładowe odpowiedzi regulatorów pozycji i prędkości silnika krokowego, z zastosowaniem generatora trajektorii

Źródło: Na podstawie [33]

Na rysunku 6.20 widać podobne przebiegi dla silnika VCM. Opóźnienie pozycji w funkcji czasu już nie występuje, gdyż zastosowany jest enkoder liniowy (sekcja 5.3). Zmiany odpowiedzi prędkości są dla tego silnika duże, ze względu na wysoki współczynnik K_p ustawiony w regulatorze prędkości.



Rysunek 6.20: Przykładowe odpowiedzi regulatorów pozycji i prędkości silnika liniowego VCM, z zastosowaniem generatora trajektorii

Źródło: Na podstawie [33]

Stała czasowa regulatora jest jednak na tyle niska, że w praktyce cylinder porusza się bardzo płynnie, bez żadnych zacięć. Podsumowując, cały mechanizm działa dobrze. Testy dowodzą, że wszystkie niedogodności związane z pracą danej maszyny są możliwe do wyeliminowania przy pomocy odpowiednich nastaw regulatorów.

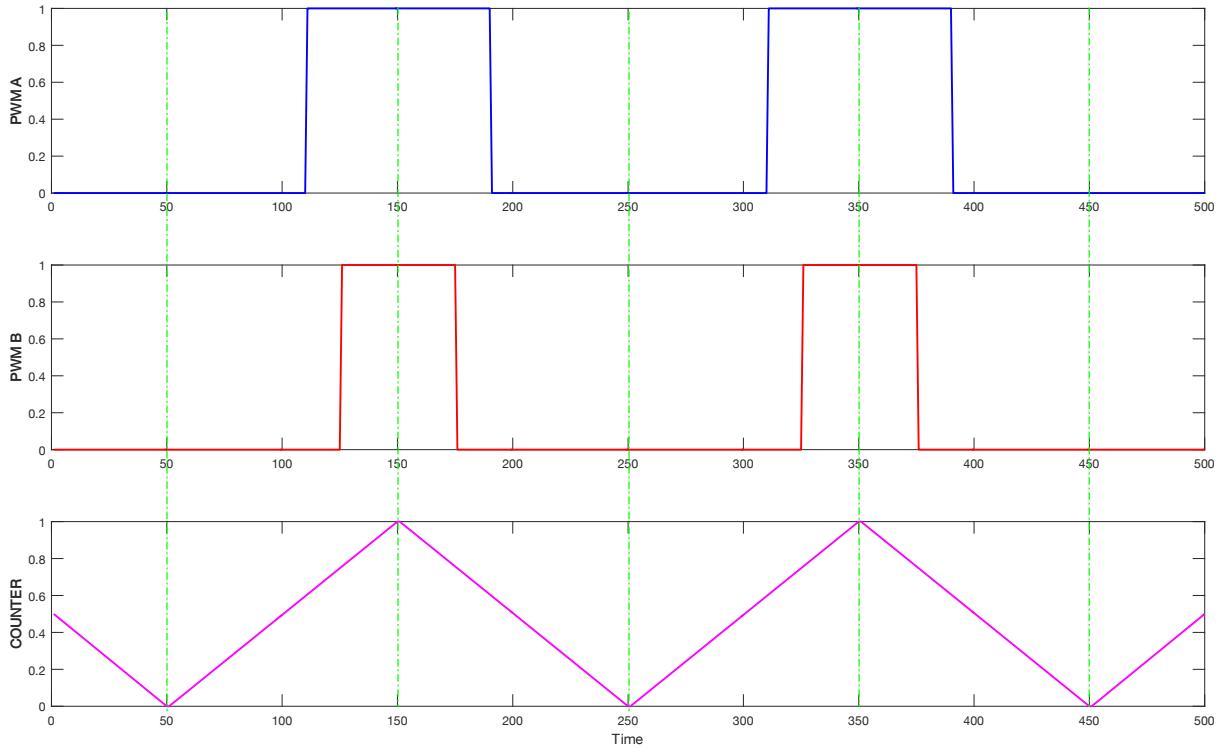
6.5. Procedury sterowania niskopoziomowego

Cechą wspólną dla wszystkich rodzajów silników obsługiwanych przez sterownik jest zasilanie uzwojeń korzystając z modulacji unipolarnej (sekcja 2.2.2) i pomiar prądu tą samą metodą.

6.5.1. Modulacja szerokości impulsów

Moduły PWM sterujące silnikami są skonfigurowane w trybie symetrycznym (*Center aligned mode*). Oznacza to, że poszczególne kanały danego timer'a są względem siebie wyrównane do środka każdego impulsu (rys. 6.21). Licznik w module inkrementuje i dekrementuje się co drugi okres. Fakt ten sprawia, że maksymalna częstotliwość możliwa do uzyskania w tym trybie PWM jest dwa razy mniejsza od tej w trybie niesymetrycznym (*Edge aligned mode*).

Tryb symetryczny pozwala jednak na implementację modulacji unipolarnej w bardzo prosty sposób. Na rys. (rys. 6.21) widać przebiegi napięcia po obu stronach uzwojenia (A i B). W prezentowanym przykładzie wypełnienie impulsu przebiegów nie jest sobie równe, co oznacza, że element wykonawczy porusza się.



Rysunek 6.21: Potencjały na zaciskach uzwojenia wraz z licznikiem danego timer'a

Źródło: Na podstawie [33]

W momentach, w których potencjał A ma inną wartość niż B, prąd płynie przez uzwojenie poprzez naprzeciwne otwarte tranzystory mostka H. Kiedy potencjały mają tą samą wartość, otwarte są te same tranzystory po obu stronach mostka (dwa gÓrne lub dolne). Wtedy prąd przepływa przez uzwojenie, ale rozładowuje siĘ. Aby przeliczyć zadany przez regulator prąd na wartość wypełnienia impulsu, w kontrolerze używane są następujące równania:

$$K_{LR} = \frac{L_w F_{pwm}}{2R_w} \quad (6.23)$$

Równanie 6.23 prezentuje wzór na stałą prądową uzwojenia. Poszczególne zmienne to:

- L_w : indukcyjnośc uzwojenia,
- R_w : rezystancja uzwojenia,
- F_{pwm} : częstotliwość przełączania kluczy,

Dwójka we wzorze jest uproszczeniem, według którego przyjmuje siĘ że w jednej stałej czasowej τ obwodu RL prąd uzwojenia osiąga 50% swojej maksymalnej wartości (6.24), wynikającej z prawa Ohm'a dla obwodu w stanie ustalonym.

$$I_{max} = \frac{V_{CC}}{R_L + R_{meas}} \quad (6.24)$$

Spadki napięć na tranzystorach sĄ w równaniu 6.24 pomijane. Stała K_{LR} jest następnie używana do obliczenia współczynnika PWM dla danego uzwojenia 6.25.

$$r_{pwm} = I_{reg} \cdot K_{LR} \cdot \frac{R_L + R_{meas}}{V_{CC}} \cdot \frac{T_{pwm}}{2} \quad (6.25)$$

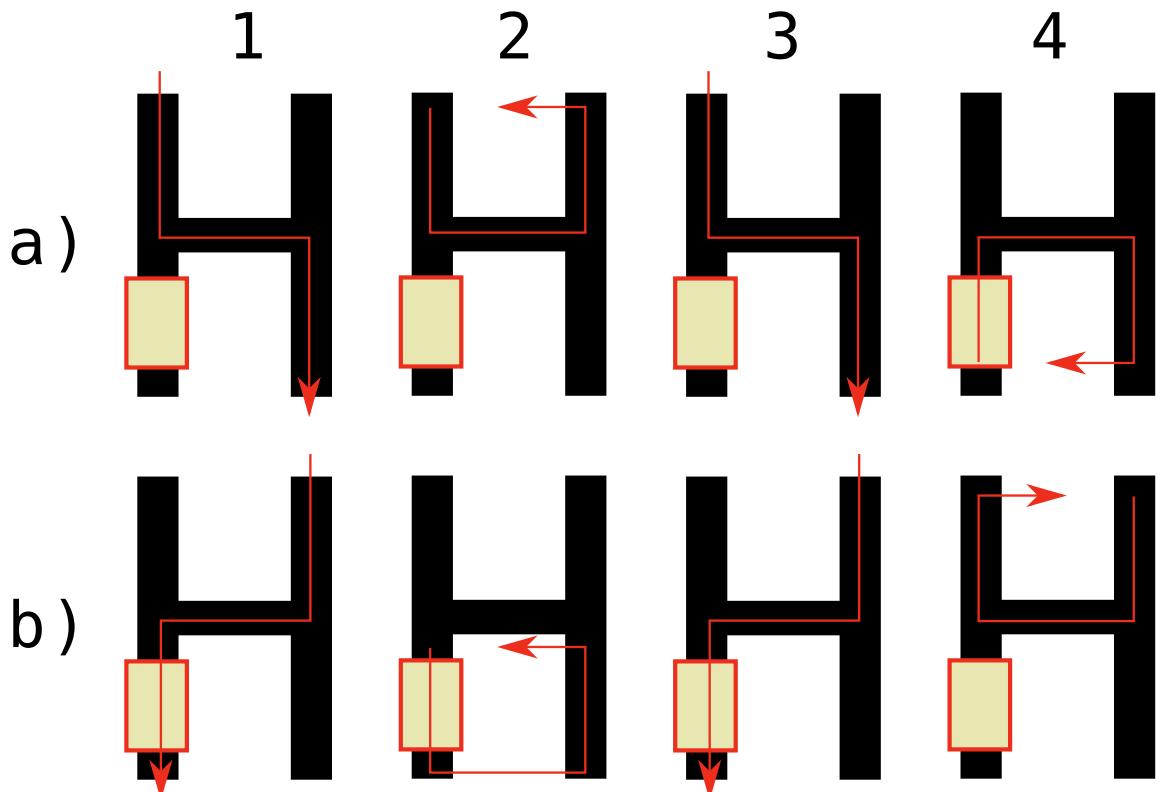
V_{CC} jest napięciem zasilania mostka, a T_{pwm} okresem PWM dla danego silnika. Na końcu obliczane są wartości wypełnień impulsów dla potencjałów po obu stronach uzwojenia 6.26.

$$PWM_{V1} = \frac{T_{pwm}}{2} + r_{pwm} \quad PWM_{V2} = \frac{T_{pwm}}{2} - r_{pwm} \quad (6.26)$$

Dla silnika krokowego obliczane są odpowiednio cztery wypełnienia impulsów zamiast dwóch.

6.5.2. Pomiar prądu w uzwojeniu

Rysunek 6.22 prezentuje uproszczony mostek H ze schematu 5.41. Cyframi zostały oznaczone kolejne sekwencje komutacyjne, w których układ znajduje się przy sterowaniu unipolarnym (sekcja 6.5.1). Litery oznaczają sekwencję dla ruchu (np. cylindra w silniku VCM) w przeciwnych kierunkach. Strzałki wyznaczają drogę przepływu prądu i jego kierunek, natomiast blok w dolnej lewej gałęzi mostka symbolizuje rezistor pomiarowy (schemat 5.39).



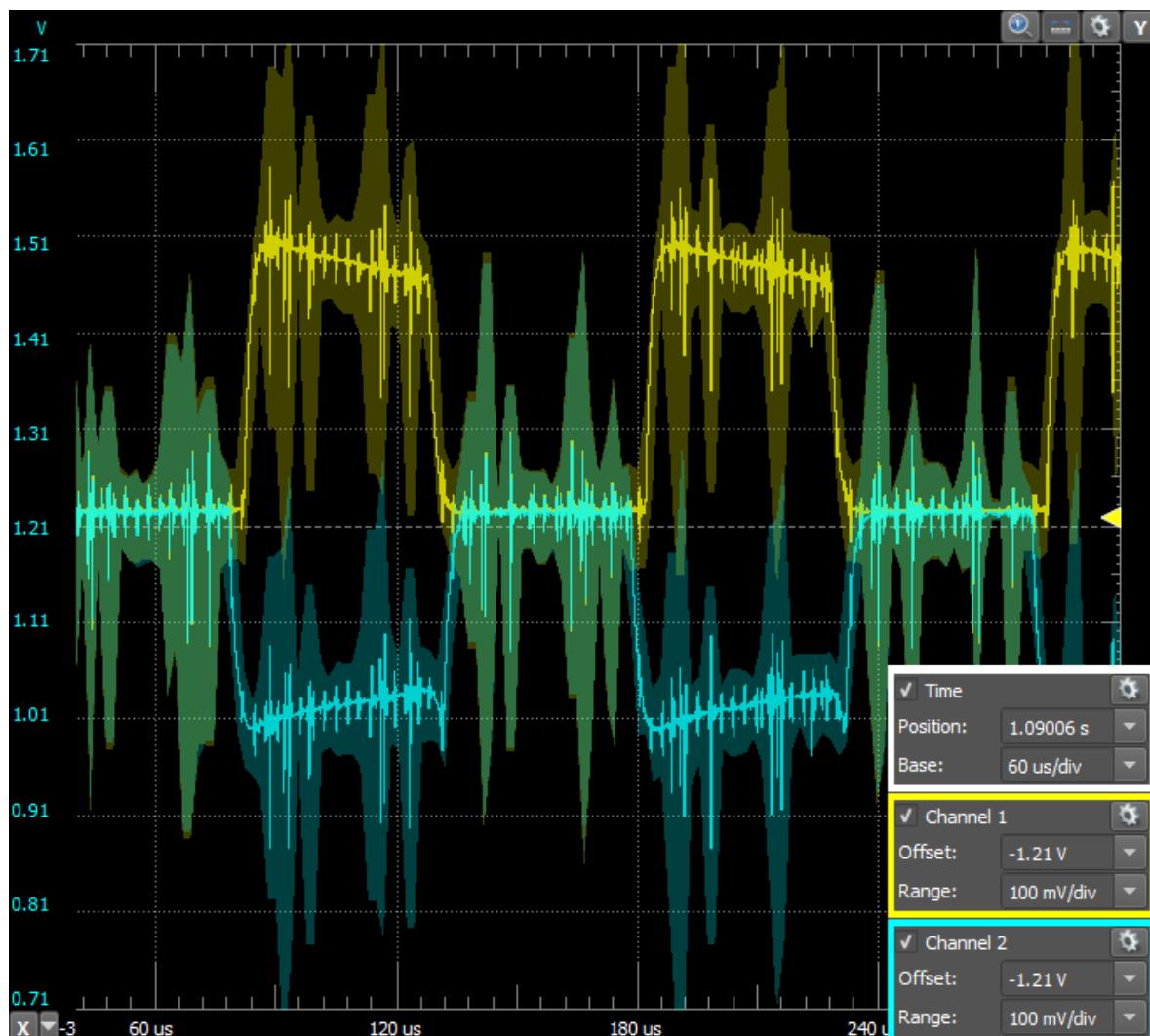
Rysunek 6.22: Przepływ prądu w mostku H dla kolejnych komutacji (1, 2, 3, 4) przy ruchu elementu wykonawczego. a) Ruch w elementu wykonawczego w prawo, b) ruch w lewo

Źródło: Na podstawie [33]

Jak wynika z diagramu, prąd przepływa przez rezistor pomiarowy tylko w połowie możliwych konfiguracji. Fakt ten sprawia, że prąd nie może być mierzony

w dowolnym momencie, a wymaga synchronizacji z modułem timer'a, którego częściami są dane kanały PWM. Aby to zrobić, w kontrolerze posłużono się przerwaniami generowanymi w momencie, kiedy licznik danego timer'a zaczyna liczyć w dół lub w górę. Na rys. 6.21 są to momenty, w których wypełnienie impulsów jest w 50% swojej całkowitej szerokości. Te przerwania następnie uruchamiają przetwornik analogowo-cyfrowy (moduł skonfigurowany sprzętowo), który odczytuje wartość spadku napięcia na rezystorze. W przypadku silnika krokowego, jest to robione dwa razy (kolejno z obu rezystorów) ze względu na to że uzwojenia są dwa.

Rys. 6.23 prezentuje oscylogram napięć odkładających się na rezystorach pomiarowych w obwodzie silnika krokoowego.



Rysunek 6.23: Oscylogramy mierzonych spadków napięć na rezystorach pomiarowych w mostku H silnika krokoowego z rys. 6.18 w przybliżeniu

Źródło: Na podstawie [33]

Można na nim zaobserwować poprawną pracę obwodu CSLS (schemat 5.39). W co drugim okresie timer'a, spadki napięć na rezystorach pomiarowych wynoszą ok. 1.2 V co sugerowałoby, że w tym czasie przez uzwojenie nie płynie prąd (napięcie referencyjne). Powodem tego zjawiska jest jednak to, że tak jak pokazano na

rys. 6.22 w połowie okresów prąd jest dla przetwornika ADC „niewidoczny” – punkt pomiarowy jest dla niego w stanie wysokiej impedancji (nie ma różnicy napięć).

Mimo tego w oprogramowaniu mierzone są napięcia we wszystkich okresach, nawet tych niewidocznych. Implementacja mechanizmu synchronizującego przetwornik z okresami, zarówno jak jego działanie było by bardziej czasochłonne niż zastosowany obecnie algorytm. Mianowicie, w programie sprawdzane jest w którym okresie dla danej pary (wysoki–niski) amplituda jest większa i tylko ją stosuje się do obliczeń. Okres z napięciem referencyjnym jest odrzucany.

Prąd płynący w uzwojeniu jest liczony ze wzoru 6.27.

$$I_{meas} = \frac{V_{ref} \cdot ADC}{G \cdot RES_{ADC} \cdot R_{meas}} \quad (6.27)$$

Gdzie:

- V_{ref} : napięcie źródła odniesienia,
- ADC : wartość z przetwornika ADC,
- RES_{ADC} : Rozdzielcość przetwornika (12 bitów),

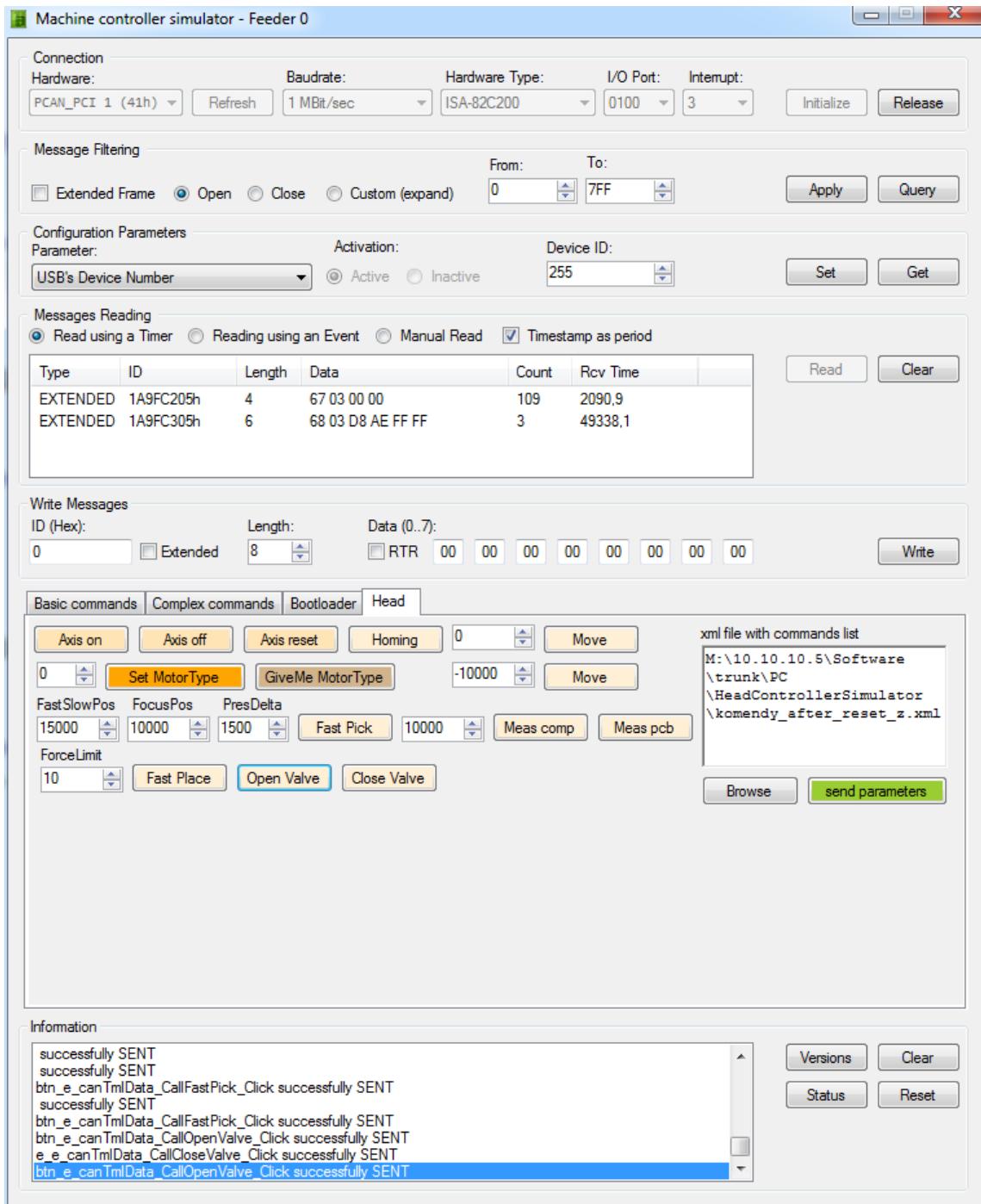
6.6. Aplikacja testująca

Aby umożliwić testowanie sterownika bez konieczności używania kontrolera systemowego maszyny, symulator kontrolera (aplikacja graficzna działająca pod systemem Windows) został wyposażony w dodatkowe komendy obsługujące konstruowaną głowicę układającą.

Program pozwala na ustawianie poszczególnych rejestrów (sekcja 6.1.1, rys. 6.4) pojedynczo używając przycisków lub zbiorczo korzystając z komend zapisanych w pliku XML (*Extensible Markup Language*). Funkcje–skrypty obsługiwane przez system są typowe dla głowicy układającej:

- bazowanie osi : sstawia osie R i Z w pozycjach początkowych,
- ruch : poruszanie się ruchem posuwisto–zwrotnym w osi Z i obrotowym w osi R,
- operacje na zaworze : Otwieranie/ zamknięcie zaworu podciśnienia,
- podnoszenie : Procedura podnoszenia komponentów elektronicznych,
- układanie : Procedura układania komponentów elektronicznych,
- pomiar wysokości: Skrypt służący do pomiaru wysokości komponentu elektronicznego,

Dodatkowo korzystając z zakładki *Bootloader* (rys. 6.24) użytkownik ma możliwość zaktualizowania oprogramowania sterownika, bez korzystania z zewnętrznego programatora. Mikrokontroler przechodzi w stan Bootloader'a (osobny program) z poziomu którego możliwe jest nadpisanie pamięci flash.



Rysunek 6.24: Aplikacja (interfejs użytkownika) pozwalająca na zadawanie/ odczytywanie parametrów i wywoływanie funkcji w sterowniku, korzystając z interfejsu CAN

Źródło: Na podstawie [31]

Podsumowanie

Będący tematem pracy uniwersalny inteligentny sterownik silników do zastosowań przemysłowych nie został jeszcze ukończony, ale jest już funkcjonalny. Ze sprzętowego punktu widzenia kontroler jest gotowy dla głowicy układającej do automatów typu Pick and Place. W przypadku potrzeby zastosowania go w innej aplikacji, należy zaprojektować nową płytę bazując na podzespołach zastosowanych w tym projekcie.

Pod względem oprogramowania wszystkie bazowe moduły sterowanie są napisane. Następnym krokiem będzie dodanie procedur dla maszyn BLDC/ PMSM i komutatorowych. Dużej zmiany w oprogramowaniu wymaga moduł wyższego poziomu, wykonujący poszczególne funkcje (np. podnoszenie komponentu). Są to funkcjonalności charakterystyczne dla danej aplikacji i nie powinny być częścią kodu bazowego uniwersalnego sterownika. Docelowo należy zaprojektować dedykowany język skryptowy, pozwalający użytkownikowi na pisanie kodu wysokiego poziomu i zapisywanie go w pamięci kontrolera. Sterownik powinien potrafić interpretować skrypt i wykonywać sekwencyjnie zapisane w nim instrukcje.

W celu napisania skryptu potrzebne jest specjalne środowisko: dedykowana aplikacja napisana na platformę PC. Program musi także umożliwiać konfigurację parametrów danego sterownika, takich jak np. rodzaj podłączonych silników czy enkoderów. Dodatkowo musi pozwalać na dostrajanie regulatorów i filtrów oraz odczytywanie parametrów chwilowych (np. prądy w uzwojeniach) w czasie rzeczywistym. Moduł logger'a (sekcja 6.1.2) nie jest w stanie zapewnić takiej funkcjonalności, gdyż jego priorytet jest bardzo niski i nie można go zwiększyć nie zaburzając pracy sterownika. Z tego powodu potrzebny jest dodatkowy moduł pozwalający na zapis danych do pamięci RAM mikrokontrolera, a następnie wysłanie ich po zakończeniu skryptu.

Ostatnim elementem (po zakończeniu zadań o wyższym priorytecie) potrzebnym w projekcie jest wyposażenie aplikacji konfiguracyjnej i samego sterownika w elementy sztucznej inteligencji. Zastosowanie np. sieci neuronowej mogło by pozwolić na automatyczne dostosowanie optymalnych współczynników dla regulatorów.

A. Pełny schemat ideowy sterownika

- płytka zasilająca (MS02-50-62-00),
- enkoder magnetyczny (MS02-50-61-00),
- enkoder optyczny (MS02-50-63-00),
- płytka główna sterownika (MS02-50-60-00).

1

2

3

4

A

A

B

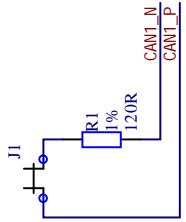
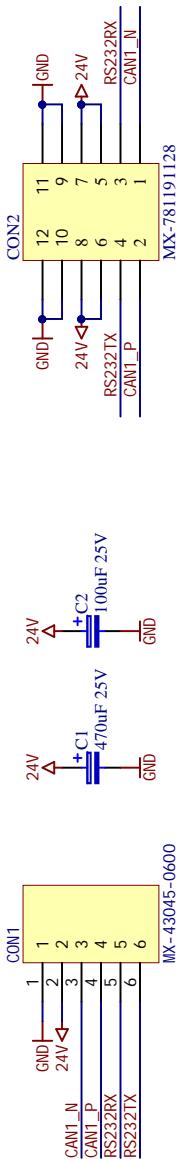
B

C

C

D

D



| | |
|--|---|
| Title: MS02-50-62-00_PowerBoard.SchDoc | MECHATRONIC |
| Project: MS02-50-62-00_PowerBoard.PrjPcb | s y s t e m s |
| Drawn by: Łukasz Przeniosło | |
| Date: 2016-07-01 | Number: MS02-50-42-00 |
| Time: 17:23:08 | File: M:\10.10.5\Hardware\trunk\Head2\MS02-50-62-00_PowerBoard.SchDoc |

4

3

2

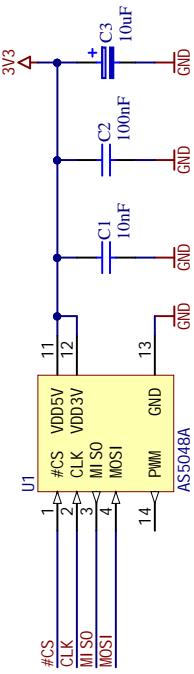
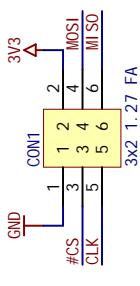
1

A

B

C

D



Title: MS02-50-61-00_magenc.SchDoc

Project: MS02-50-61-00_magenc.PjPcb

Drawn by: Lukasz Przenioslo

Date: 2016-07-01 Time: 17:17:05 Number: MS02-50-42-00

File: M:\10_10_10_5\Hardware\trunk\Head2\MS02-50-61-00_magenc.SchDoc



MECHATRONIC
systems

4

Hole1 Hole2 Hole3 Hole4 FID1
H H H H
Hole Hole Hole Hole FID2

1

2

3

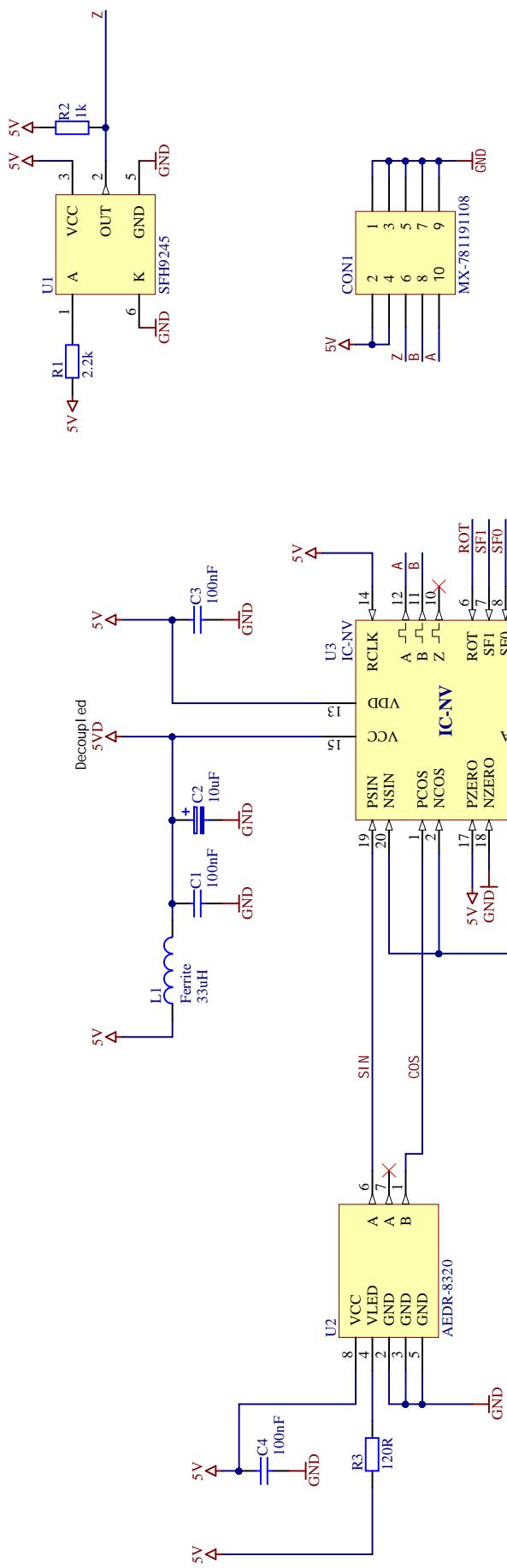
4

4

3

2

1

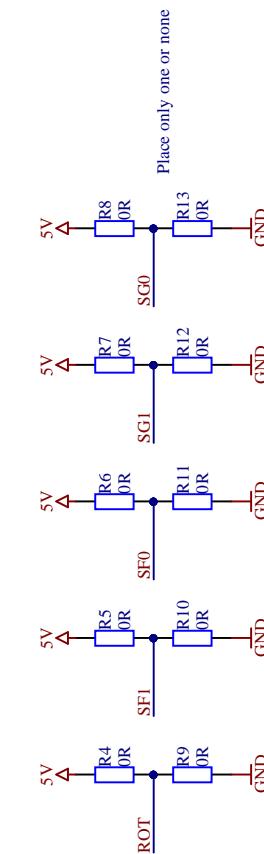


C

C

D

D



Title: MS02-50-63-00-linenec.SchDoc

Project: MS02-50-63-00-linenec.PrtPcb

Drawn by: Łukasz Przeniosło

Date: 2016-07-01 Time: 17:23:48

File: M:\10.10.10.5\Hardware\trunk\Head2\MS02-50-63-00-linenec.SchDoc

Sheet 1 of 1

Number: MS02-50-42-00

3

2

1

1

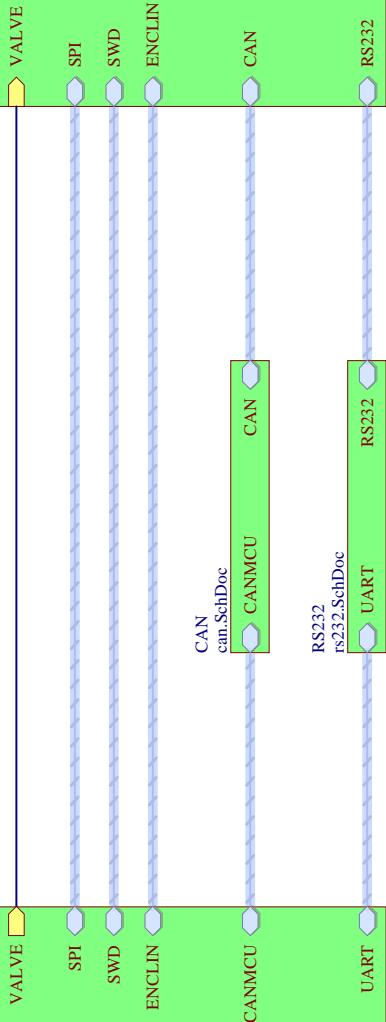
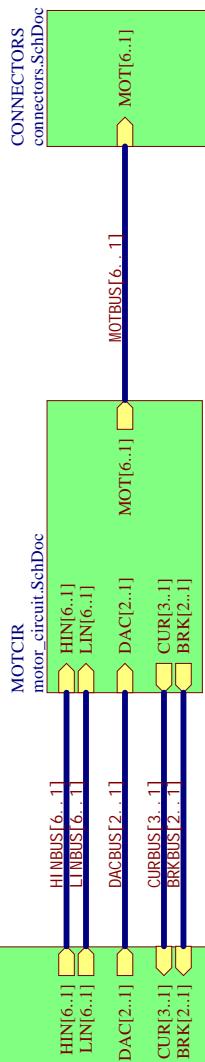


4

3

2

1



Title: top_level.SchDoc

Project: MS02-50-60-00_DualMotHead.PrtPcb

Drawn by: Lukasz Przenioslo

Date: 2016-06-30 Time: 00:16:54

File: M:\10.10.10.5\Hardware\Trunk\Head2\MS02-50-60-00_DualMotHead\top_level.SchDoc

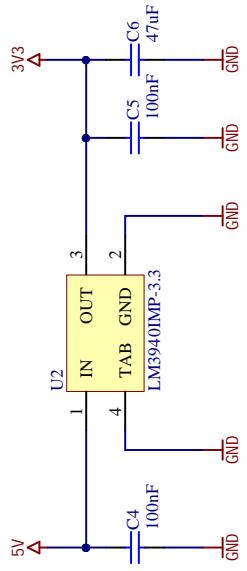
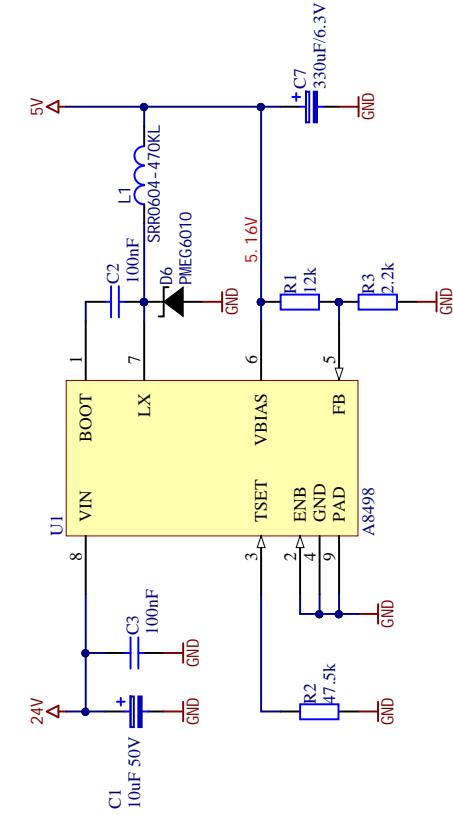


4

3

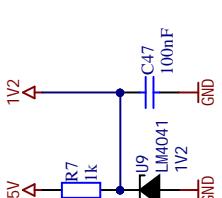
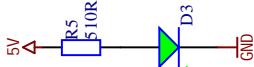
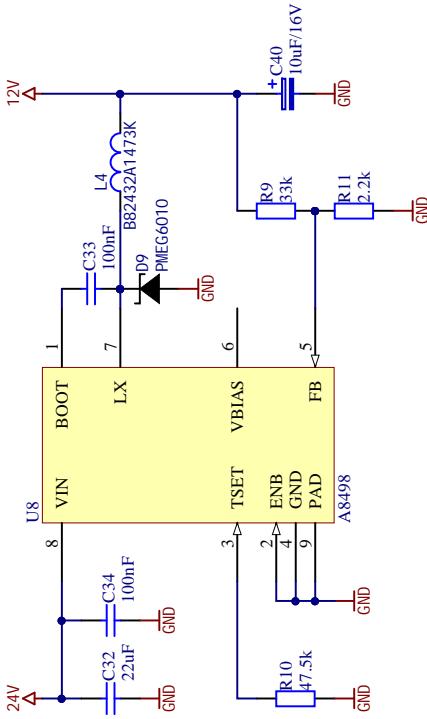
2

1



MCU VREF. Place close to the MCU vref pin. Cap on the MCU sheet.

Current sense differential op amps ref voltage.



Title: power.SchDoc

Project: MS02-50-60-00_DualMotHead.PrjPcb

Drawn by Łukasz Przeniosło

Date: 2016-06-30 Time: 00:11:54

File: M:\10.10.5\Hardware\trunk\Head2\MS02-50-60-00_DualMotHead\power.SchDoc

Sheet 2 of 12

Number: MS02-50-42-00

MECHATRONIC

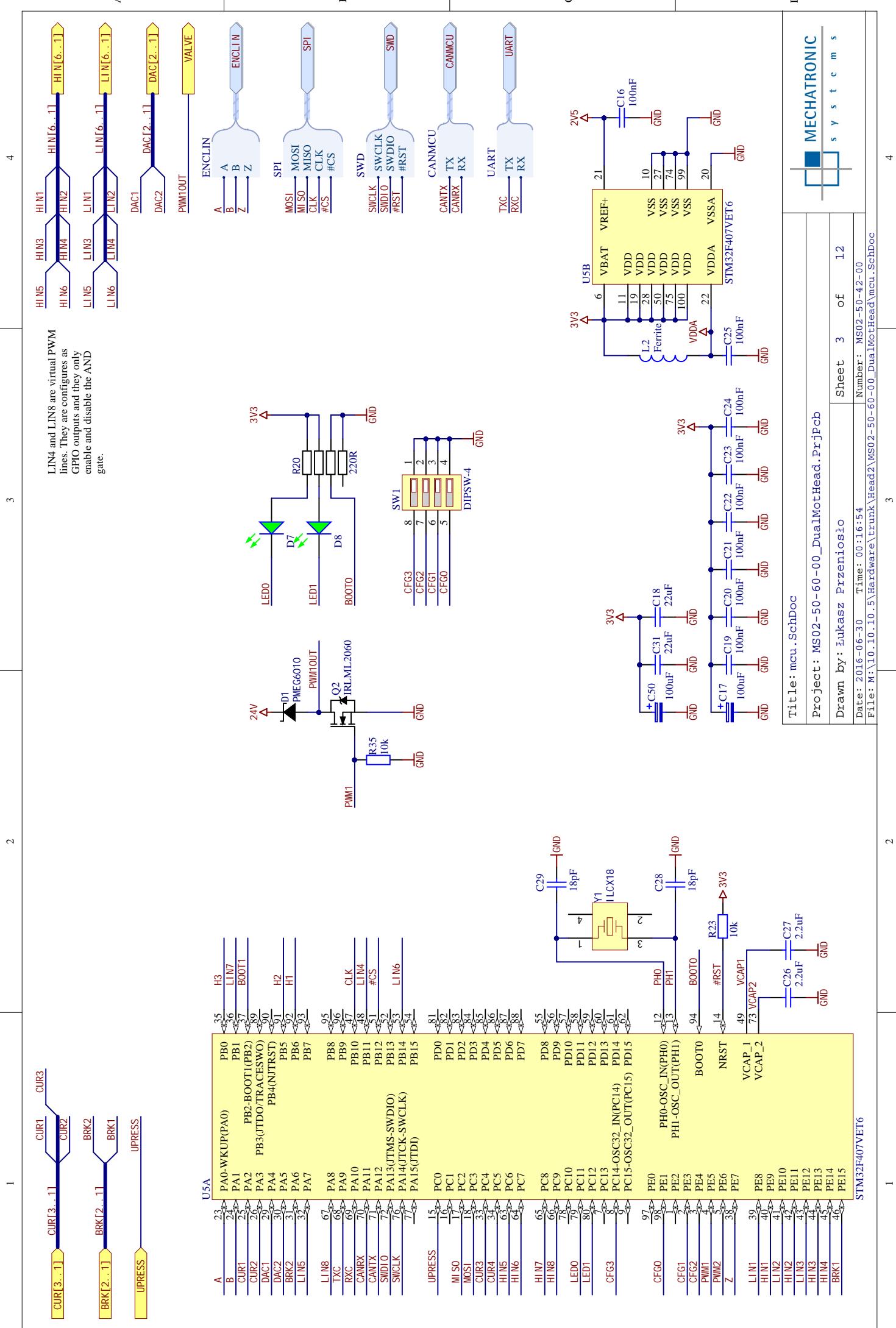
systems

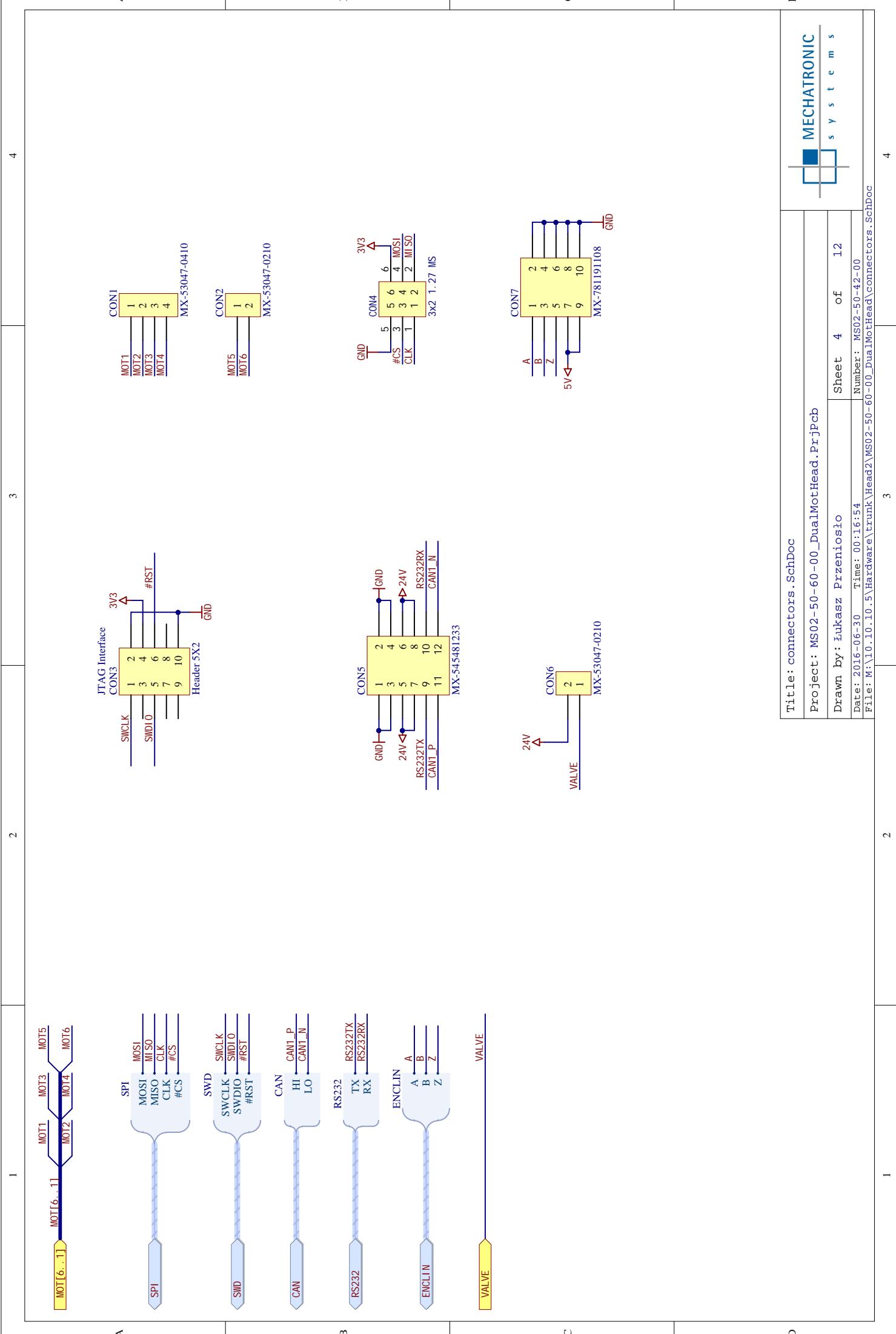
4

3

2

1





4

3

2

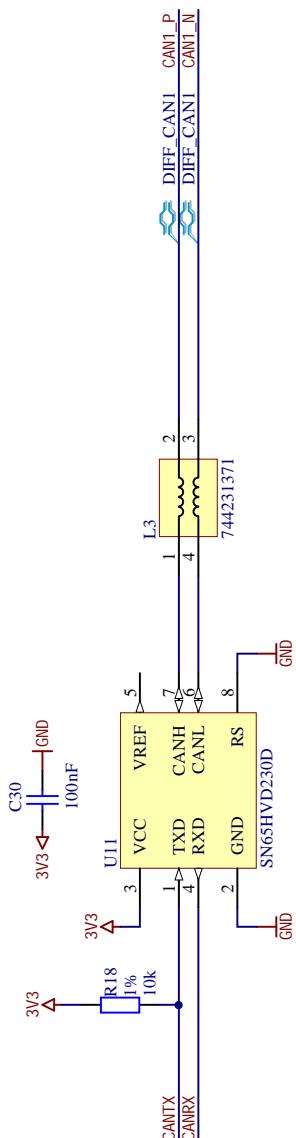
1

A

B

C

D



Title: can.SchDoc

Project: MS02-50-60-00_DualMotHead.PjPcb

Drawn by: Lukasz Przenioslo

Date: 2016-06-30 Time: 00:16:54 Number: MS02-50-42-00

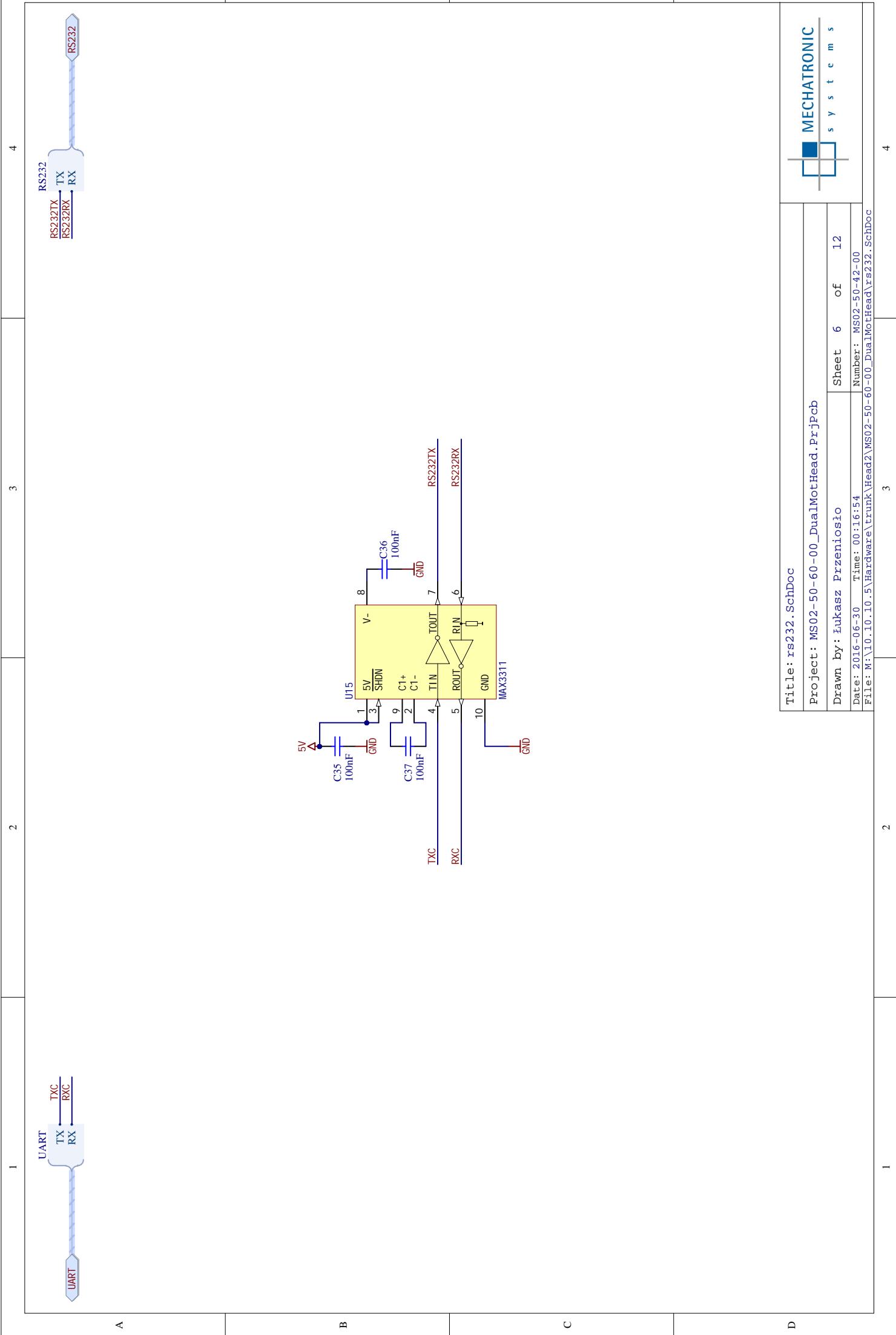
File: M:\10.10.10.5\Hardware\Trunk\Head2\MS02-50-60-00_DualMotHead\can.SchDoc

4

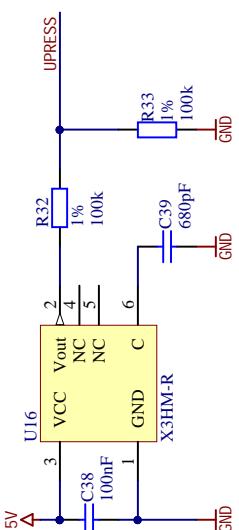
MECHATRONIC
systems

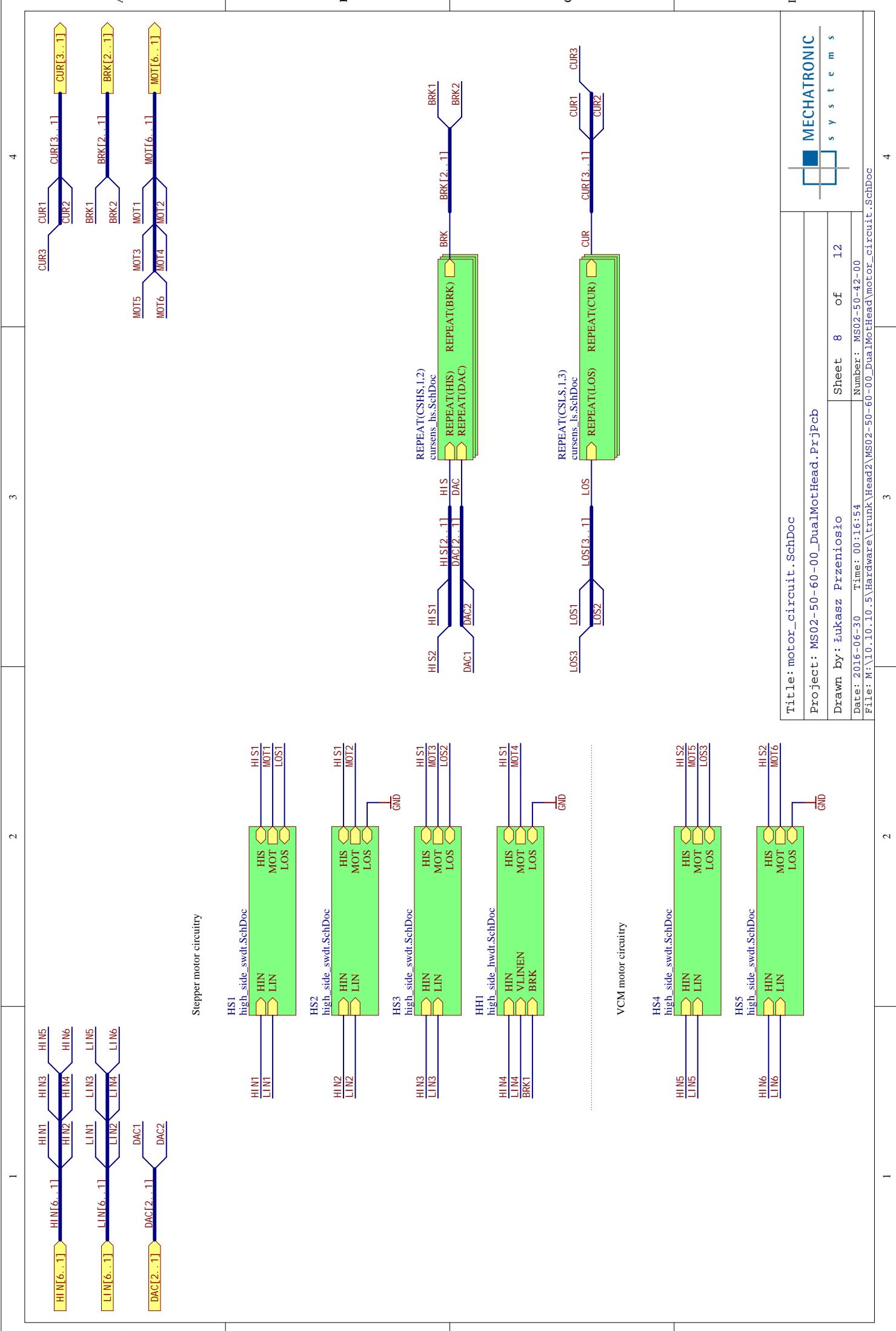
D

4



| | | | |
|---------|---------|---|---|
| A | B | C | D |
| 1 | 2 | 3 | 4 |
| UPPRESS | UPPRESS | | MECHATRONIC systems |
| | | | Number: MS02-50-42-00 |
| | | | Date: 2016-06-30 Time: 00:16:54 |
| | | | File: M:\10_10_10_5\Hardware\trunk\Head2\MS02-50-60-00_DualMotHead\upress_sensor.SchDoc |
| | | | Sheet 7 of 12 |
| | | | Title: upress_sensor.SchDoc |
| | | | Project: MS02-50-60-00_DualMotHead.PjPcb |
| | | | Drawn by: Lukasz Przenioslo |
| | | | |



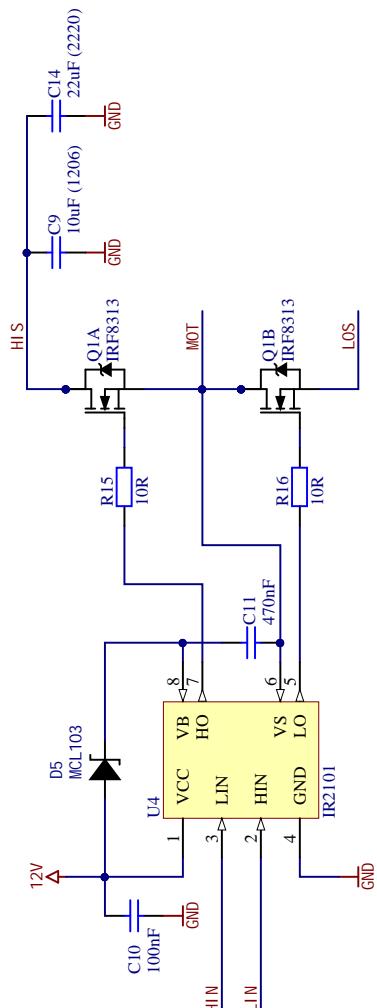


HIN **HI S**
LIN **MOT**
LOS

A

A

B



High side without internal Dead time. MCU has to provide it. No overshoot protection here.

C

C

D

D

Title: high_side_swdt.SchDoc

Project: MS02-50-60-00_DualMotHead.ProyPcb

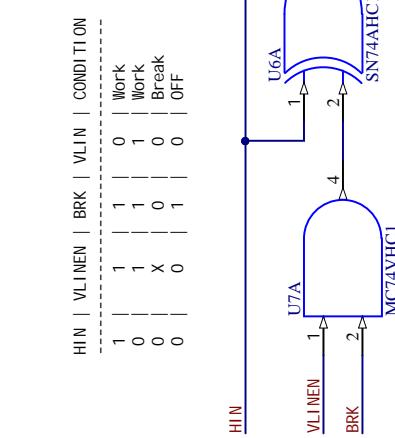
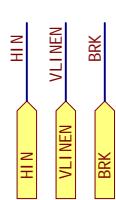
Drawn by: Lukasz Przenioslo

Date: 2016-06-30 Time: 00:16:54 Number: MS02-50-42-00

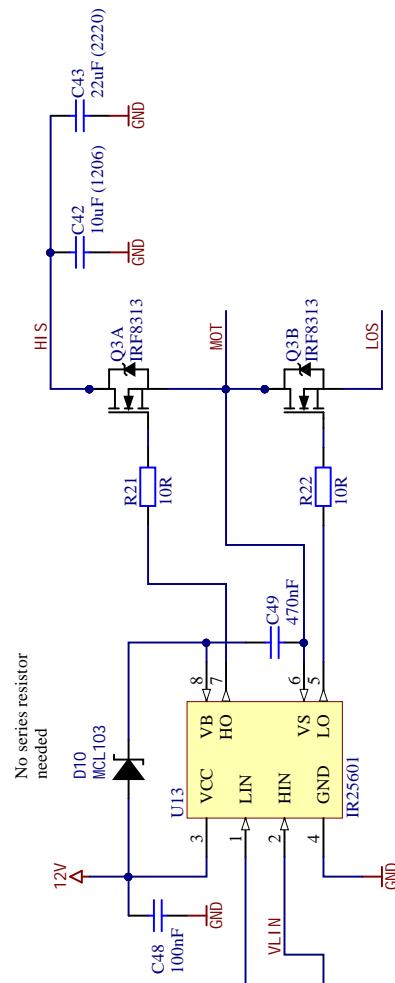
File: M:\10.10.10.5\Hardware\trunk\Head2\MS02-50-60-00_DualMotHead\high_side_swdt.SchDoc



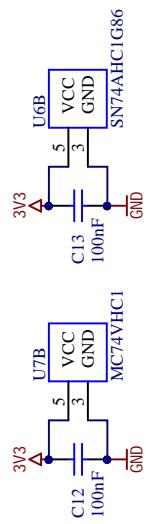
1 2 3 4



No series resistor
needed



IR23601 has internal 100 ns deadtime. Use this chip only for
the channels that dont have internal DT (CH4).



Above circuit is used to create a complementary PWM
signal for the 4th channel. Deadtime is not applied yet.

Title: high_side_hwdt.SchDoc

Project: MS02-50-60-00_DualMotHead.PrjPcb

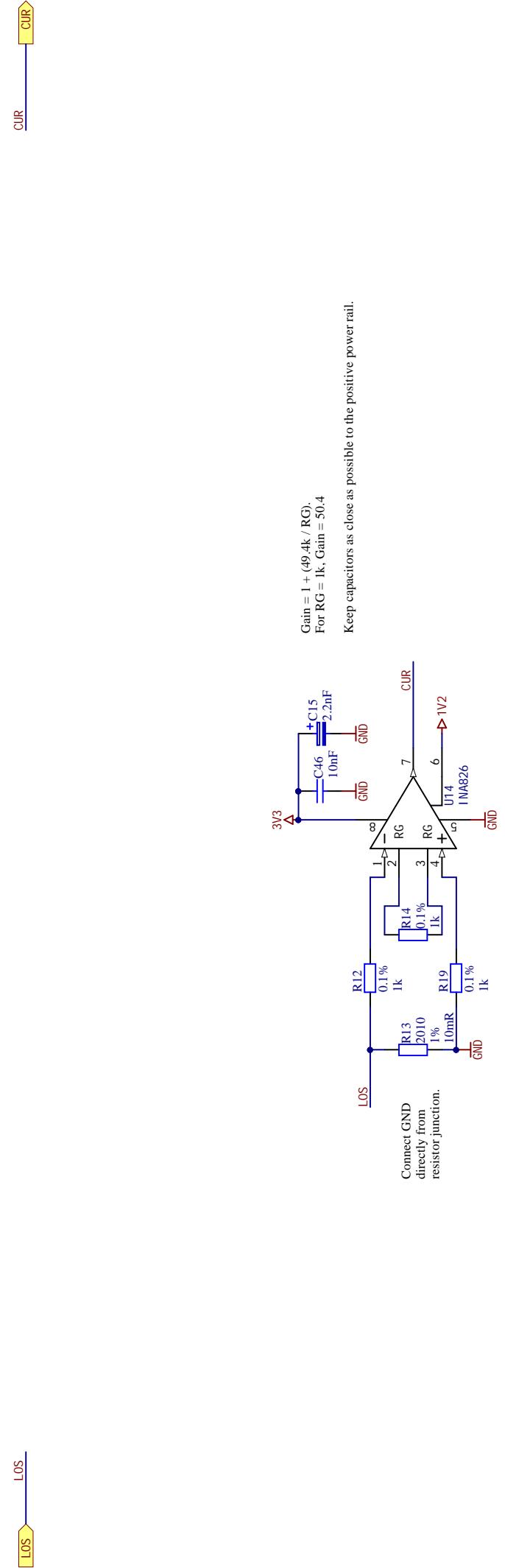
Sheet 10 of 12

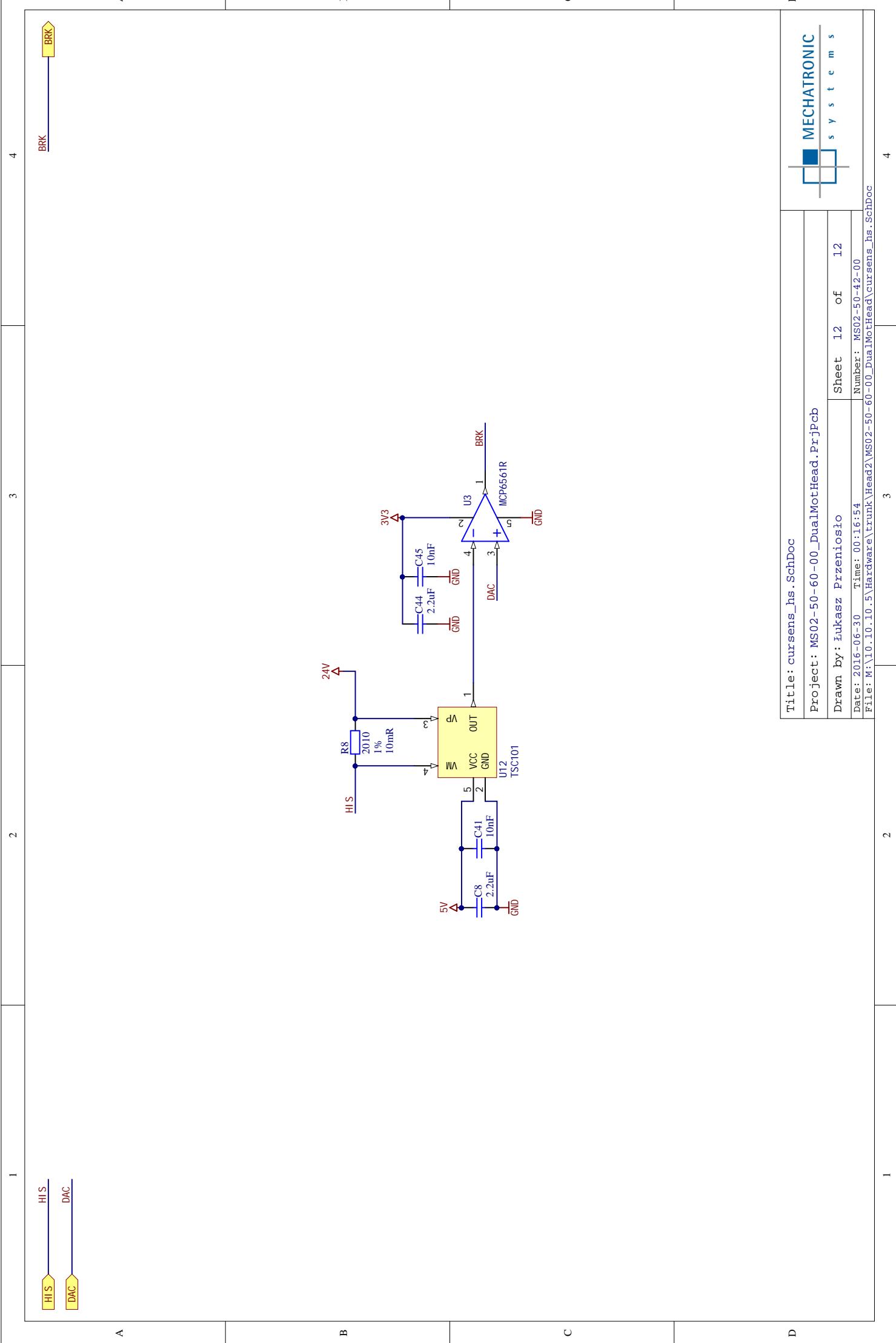
Number: MS02-50-42-00

Date: 2016-06-30 Time: 00:11:54

File: M:\10.10.5\Hardware\trunk\Head2\MS02-50-60-00_DualMotHead\high_side_hwdt.SchDoc







Literatura

- [1] Allegro MicroSystems, LLC. <http://www.allegromicro.com>.
- [2] Engineers Garage. <http://www.engineersgarage.com>.
- [3] Forbot. <http://forbot.pl>.
- [4] Hackaday. <https://hackaday.com>.
- [5] How To Mechatronics. <http://howtomechatronics.com>.
- [6] HyperPhysics. <http://hyperphysics.phy-astr.gsu.edu>.
- [7] International Rectifier. <http://www.infineon.com/>.
- [8] Mechaduino. <http://tropical-labs.com/index.php/mechaduino>.
- [9] Mechatronic Systems. <http://mechatronic-systems.com>.
- [10] MotiCont. <https://http://http://moticont.com>.
- [11] Nanotec Electronic GmbH and Co. KG. <https://http://en.nanotec.com>.
- [12] National Instruments. <http://www.ni.com>.
- [13] Simplex motion AB. <http://simplexmotion.com>.
- [14] ST Microelectronics. <http://www.st.com>.
- [15] Technosoft S.A. <http://technosoftmotion.com>.
- [16] Texas Instruments. <https://ti.com>.
- [17] Texas Instruments- Motor Drive and Control. <https://ti.com/motorblog>.
- [18] Trinamic Motion Control. <http://www.trinamic.com>.
- [19] Wikimedia commons. <https://commons.wikimedia.org/wiki/File:Manoderecha.svg>.
- [20] XIAWU. <http://www.xiawu.com>.
- [21] Dokumentacja układu SFH9245, Wydanie 1.1. OSRAM GmbH, 05-11-2013.
- [22] Dokumentacja układu AEDR-8320. Avago Technologies, 2013.
- [23] Dokumentacja układu IC-TW2. IC-HAUS Integrated circuits, 2013.
- [24] Dokumentacja układu AS5048A/ AS5048B. AMS Sensor Solutions Germany GmbH, 5 czerwca 2014.
- [25] Hołub M., Bonisławski M. *Laboratorium Automatyki Napędu Elektrycznego– Właściwości i układy sterowania silników krokowych*. Zachodniopomorski Uniwersytet Technologiczny w Szczecinie. Katedra Elektroenergetyki i Napędów Elektrycznych, 2015.
- [26] Jones D. W. The University of Iowa, Department of Computer Science.
- [27] Kędzierski J. Filtr Kalmana– zastosowania w prostych układach sensorycznych. *Konferencja Naukowe Robotyków KoNAR (9 października 2007)*. <http://www.konar.pwr.wroc.pl/>.
- [28] Krykowski K. *Silniki PM BLDC- właściwości, sterowanie, aplikacje*, Wydanie 1. Wydawnictwo BTC, Legionowo 2015.
- [29] Namboodiri A., Wani H. S. Unipolar and Bipolar PWM Inverter. *IJIRST- International Journal for Innovative Research in Science and Technology*, Tom 1, wydanie 7 (Grudzień 2014).
- [30] Potocki L. Silniki krokowe od podstaw. *Elektronika dla Wszystkich* (sierpień 2002).
- [31] Przeniosło Ł. Część wewnętrznej dokumentacji lub kodu dla urządzeń wbudowanych lub aplikacji PC. Mechatronic Systems.
- [32] Przeniosło Ł. Część wewnętrznej dokumentacji projektowej CAD/ EDA lub render z projektu CAD/ EDA 3D. Mechatronic Systems.
- [33] Przeniosło Ł. Opracowanie własne.
- [34] Przepiórkowski J. *Silniki elektryczne w praktyce elektronika*, Wydanie 2. Wydawnictwo BTC, Warszawa 2012.
- [35] Siwienki K., Januszkievicz F., Łubieński L. Część wewnętrznej dokumentacji projektowej CAD lub render z projektu CAD 3D. Mechatronic Systems.
- [36] Smith S. W. *Cyfrowe przetwarzanie sygnałów- Praktyczny poradnik dla inżynierów i naukowców*, Wydanie 1. Wydawnictwo BTC, Legionowo 2007.

- [37] Tietze U., Schenk C. *Układy półprzewodnikowe*. Wydawnictwa naukowo-Techniczne, Warszawa 1996.
- [38] Wilson D. So, which PWM technique is best? (Part 1). *Electronic Components News* (luty 2012).
- [39] Zawirski K. Sterowanie silnikiem synchronicznym o magnesach trwałych. *Postępy napędu elektrycznego i energoelektroniki*, Zeszyt nr 51 (Poznań 2005).
- [40] Zawirski K., Deskur J., Kaczmarek T. *Automatyka napędu elektrycznego*, Wydanie 1. Wydawnictwo Politechniki Poznańskiej, Poznań 2012.

Spis tabel

| | |
|--|----|
| 1.1 Cechy poszczególnych trybów sterowania silnikiem krokowym | 24 |
| 4.1 Typ maszyny produkowanej przez Mechatronic Systems i używane przez nią silniki wraz ze sterownikami | 44 |
| 4.2 Opis osi w automacie P30 | 44 |
| 5.1 Porównanie mikrokontrolerów STM32F407VGT6 (ST Microelectronics) i TMS320F28377S (Texas Instruments) | 62 |
| 5.2 Tabela prawdy prezentująca działanie zaprojektowanego sprzętowego modułu komplementarnego | 72 |

Spis rysunków

| | | |
|------|--|----|
| 1.1 | Przykładowy silnik krokowy hybrydowy ST4118 w standardzie Nema17, produkowany przez firmę Nanotec | 10 |
| 1.2 | Przykładowy silnik BLDC (model DB42), produkowany przez firmę Nanotec | 10 |
| 1.3 | Przykładowy silnik krokowy hybrydowy L35 z przekładnią śrubową, produkowany przez firmę Nanotec | 11 |
| 1.4 | Przykładowy silnik liniowy VCM (model GVCM-051-051-01), produkowany przez firmę Moticont | 11 |
| 1.5 | Budowa cylindrycznego silnika VCM | 13 |
| 1.6 | Przykładowy przebieg siły $[\frac{N}{A}]$ pchającej/ ciągnącej w funkcji wysunięcia korpusu dla silnika liniowego VCM 019-048-02, firmy Moticont. | 14 |
| 1.7 | Powstawanie reluktancyjnego momentu obrotowego. | 15 |
| 1.8 | Silnik o zmiennej reluktancji (VRM)- Przekrój. | 16 |
| 1.9 | Zasada działania silnika PM. | 16 |
| 1.10 | Konstrukcja dwufazowego silnika krokowego hybrydowego. | 17 |
| 1.11 | Budowa wirnika silnika krokowego hybrydowego dwufazowego. | 17 |
| 1.12 | Sposób sterowania (zasilania) silnika krokowego dwufazowego- z lewej unipolarnie, z prawej bipolarne. | 18 |
| 1.13 | Uproszczony model silnika krokowego dwufazowego o kroku 90° | 18 |
| 1.14 | Sekwencja zasilania uzwojeń przy sterowaniu falowym (obrót wirnika z rys. 1.13 według wskazówek zegara). | 19 |
| 1.15 | Sekwencja zasilania uzwojeń przy sterowaniu pełnokrokowym (obrót wirnika z rys. 1.13 według wskazówek zegara). | 20 |
| 1.16 | Sekwencja zasilania uzwojeń przy sterowaniu półkrokowym (obrót wirnika z rys. 1.13 według wskazówek zegara). | 21 |
| 1.17 | Sekwencja zasilania uzwojeń przy sterowaniu półkrokowym ze zmiennym prądem (obrót wirnika z rys. 1.13 według wskazówek zegara). | 22 |
| 1.18 | Ćwiartki okręgu i funkcji $\sin x$ w kartezjańskim układzie współrzędnych | 23 |
| 1.19 | Kształt prądów podawanych na uzwojenia A i B w celu uzyskania optymalnej płynności obrotu w silniku krokowym hybrydowym dwufazowym. | 23 |
| 1.20 | Kształt prądów podawanych na uzwojenia A i B przy sterowaniu mikrokrokowym z rozdzielcością $\frac{1}{8}$ kroku. | 24 |
| 1.21 | Wykres czasowy ilustrujący położenie wirnika podczas wykonywania skoku | 25 |
| 1.22 | Zasada działania silnika komutatorowego z magnesem trwałym | 25 |
| 1.23 | Przekrój typowego silnika trójfazowego z magnesami trwałymi: a) z wirnikiem wewnętrznym, b) z wirnikiem zewnętrznym, 1- jarzmo stojana (twornika), 2- zęby stojana, 3- żłobek z uzwojeniami, 4- wirnik, 5- magnes trwały | 26 |
| 1.24 | Przebieg indukcji, SEM i prądu jednej fazy silnika o magnesach trwałych: a) BLDC, b) PMSM | 27 |
| 1.25 | Konfiguracja uzwojeń trójfazowego silnika synchronicznego w gwiazdę | 28 |
| 2.1 | Typowy mostek typu „H” zbudowany z czterech tranzystorów polowych z izolowaną bramką (<i>Mosfet</i>) | 29 |
| 2.2 | Charakterystyki statyczne silnika: Ω - prędkość obrotowa, M - moment obrotowy | 30 |
| 2.3 | Silnik DC sterowany bipolarne | 31 |
| 2.4 | Przebiegi napięć w modulacji bipolarnej | 32 |
| 2.5 | Silnik DC sterowany unipolarne | 33 |
| 2.6 | Przebiegi napięć w modulacji unipolarnej | 34 |

| | | |
|------|---|----|
| 3.1 | Przykładowe typy obudów w jakich są produkowane układy scalone do sterowania silnikami. Na górze obudowy TSSOP (<i>Thin Shrink Small Outline Package</i>) i SSOP (<i>Shrink small outline package</i>) na dole | 35 |
| 3.2 | Funkcjonalny schemat blokowy układu A3950 | 36 |
| 3.3 | Funkcjonalny schemat blokowy układu DRV8825 | 37 |
| 3.4 | Ogólny schemat blokowy układu TMCC160 | 38 |
| 3.5 | Inteligentny, wysoce zintegrowany serwo mechanizm Simplexmotion 100A szwedzkiej firmy Simplex Motion. Wymiary: 72 x 64 x 74 mm, waga: 660 gram. Cena: \$353 | 39 |
| 3.6 | Sterownik do silników krokowych Mechaduino firmy Tropical Labs, kompatybilny sprzętowo i programowo z platformą Arduino Zero | 40 |
| 3.7 | Przykładowy sterownik z rodziny <i>Plug-in Intelligent Servo Drives</i> , PIM2403. Firma Technosoft | 41 |
| 3.8 | Kontroler PIM2403 zamontowany na płytce bazowej MS02-06-31 firmy Mechatronic Systems | 42 |
| 4.1 | Przykładowy produkt firmy Mechatronic Systems: Automat P10- maszyna typu <i>Pick and Place</i> (układająca komponenty elektroniczne na płytach PCB) z wbudowanym dyspenserem pasty lutowniczej/ kleju. | 43 |
| 4.2 | Model 3D maszyny P30 typu Pick and Place z wbudowanym dyspenserem z oznaczonymi osiami, a) widok lufy i głowic od przodu maszyny, b) Cała maszyna w rzucie ortogonalnym | 45 |
| 4.3 | Struktura zastosowanych sterowników w maszynie P30 obecnie i w przyszłości po pełnej implementacji opracowywanego rozwiązania | 46 |
| 5.1 | Prototyp głowicy układającej z zastosowaniem projektowanego sterownika: a) Widok modelu 3D głowicy od boku z wsuniętą pinolą, b) ten sam model z pinolą wysuniętą, c) rzeczywiste zdjęcie głowicy. Wymiary przy wsuniętej pinoli 34 x 38 x 184 mm | 48 |
| 5.2 | Schemat ideowy płytki zasilającej w głowicy układającej | 49 |
| 5.3 | Widok ortogonalny modelu 3D płytki zasilającej, a) z widokiem warstwy dolnej, b) z widokiem warstwy górnej | 50 |
| 5.4 | Schemat blokowy układu scalonego AS5048A | 51 |
| 5.5 | Schemat ideowy płytki z enkoderem magnetycznym dla osi R w głowicy układającej | 51 |
| 5.6 | Widok płytka enkodera magnetycznego- a) dolna warstwa PCB, na której znajduje się AS5048A wraz z kondensatorami blokującymi, b) rzut ortogonalny płytki zamocowanej na elemencie wspólnym dla niej i silnika . | 52 |
| 5.7 | Płytki z enkoderem magnetycznym zamocowana nad obracającym się, przelotowym magnesem trwałym, który z kolei jest zamocowany na przelotowym wale silnika krokowego | 53 |
| 5.8 | Część schematu ideowego płytki enkodera liniowego osi Z | 54 |
| 5.9 | Schemat blokowy i zasada działania układu scalonego AEDR-8320 firmy Avago Technologies | 54 |
| 5.10 | Schemat blokowy 8-bitowego interpolatora IC-TW2 firmy IC-HAUS | 55 |
| 5.11 | Część schematu ideowego płytki enkodera liniowego osi Z, prezentująca zwojki konfiguracyjne | 55 |
| 5.12 | Schemat blokowy układu scalonego SFH9245 firmy Osram | 56 |
| 5.13 | Model 3D płytki drukowanej enkodera liniowego silnika VCM. Po lewej warstwa góra, po prawej dolna | 56 |
| 5.14 | Zdjęcie głównej płytki sterującej głowicy układającej (warstwa góra) | 57 |
| 5.15 | Poszczególne warstwy płytki głównej sterownika, od lewej góra, dolna, warstwa zasilania (<i>power plane</i>), warstwa masy (<i>ground plane</i>) | 58 |

| | | |
|------|---|----|
| 5.16 | Schemat ideowy najwyższego poziomu łączący ze sobą poszczególne moduły w projekcie płytki, w programie Altium Designer | 59 |
| 5.17 | Część schematu zasilania prezentująca sposób uzyskania napięć 3.3 i 5 V na płytce sterownika | 60 |
| 5.18 | Źródła napięcia odniesienia dla obwodu analogowego sterownika | 60 |
| 5.19 | Przetwornica typu Buck-converter dająca na wyjściu 12V, zasilającą sterowniki typu <i>High-Side</i> | 61 |
| 5.20 | Rozmieszczenie modułów zasilania na płytce głównej, A: obwód 5.19, B: obwód 5.17 | 61 |
| 5.21 | Mikrokontroler STM32F407VGT6 w obudowie TQFP100 o wymiarach 14 x 14 mm | 63 |
| 5.22 | Widok 3D mikrokontrolera sterującego i otaczających go komponentów . . | 63 |
| 5.23 | Konfiguracja pętli PLL dla częstotliwości maksymalnej 168 Mhz | 64 |
| 5.24 | Zastosowany w kontrolerze układ MAX331 firmy Maxim, służący do translacji poziomów logicznych RS232 na TTL/ CMOS | 64 |
| 5.25 | Komponenty używane przez moduł RS232 zajmują jedynie ok. 20mm ² . . . | 65 |
| 5.26 | Moduł CAN korzystający z układu SN65HVD230D (Texas Instruments) w celu translacji sygnałów z magistrali | 65 |
| 5.27 | Widok 3D na moduł CAN | 66 |
| 5.28 | Czujnik podciśnienia XFHM-100KPGWRH znajdujący się na płytce głównej . | 66 |
| 5.29 | Schemat ideowy obwodu czujnika podciśnienia | 67 |
| 5.30 | Schemat ideowy modułu MOTCIR ze schematu nadzawanego 5.16 | 68 |
| 5.31 | Schemat ideowy modułu HS (<i>High Side driver</i> z programowym czasem martwym) | 69 |
| 5.32 | Schemat funkcjonalny <i>High/ Low side driver'a</i> IR2101 firmy International Rectifier (obecnie Infineon) | 69 |
| 5.33 | Schemat blokowy zaawansowanego Timer'a 1 i 8 w mikrokontrolerach STM32F4 firmy ST Microelectronic | 70 |
| 5.34 | Schemat ideowy modułu HH (<i>High Side driver</i>) ze sprzętowym czasem martwym i kanałem komplementarnym | 71 |
| 5.35 | Część PCB w której znajdują się dyskretne bramki U7 i U6 tworzące kanał komplementarny | 71 |
| 5.36 | Schemat funkcjonalny <i>High/ Low side driver'a</i> IR25601 firmy International Rectifier (obecnie Infineon) | 72 |
| 5.37 | Schemat ideowy modułu CSHS (<i>Current Sense at High Side</i>) zabezpieczającego uzwojenia przed przeciążeniem prądowym | 73 |
| 5.38 | Schemat funkcjonalny wzmacniacza pomiarowego TSC101 firmy ST Microelectronic | 74 |
| 5.39 | Schemat ideowy modułu CSLS (<i>Current Sense at Low Side</i>) służącego do pomiaru prądu w uzwojeniach | 75 |
| 5.40 | Schemat blokowy obwodu silnika krokowego | 75 |
| 5.41 | Schemat blokowy obwodu silnika VCM | 76 |
| 5.42 | Wyszczególnione moduły obwodu silników po stronie górnej PCB | 76 |
| 5.43 | Wyszczególnione moduły obwodu silników po stronie dolnej PCB | 77 |
| 5.44 | Zaznaczony niedostępny obszar dla propagacji masy po dolnej stronie PCB | 77 |
| 6.1 | Widok wstępnie skonfigurowanych peryferiów w programie mxCube firmy ST Microelectronics | 78 |
| 6.2 | Schemat blokowy architektury oprogramowania sterownika | 79 |
| 6.3 | Topologia sieci CAN. Sterownik przedstawia się jako dwa osobne urządzenia | 80 |
| 6.4 | Sposób tłumaczenia ramek TML CAN w kontrolerze. Przykładowy zapis prędkości ruchu w osi Z | 81 |

| | | |
|------|---|-----|
| 6.5 | Diagram interakcji prezentujący wykonanie przez system procedury bazowania (<i>homing</i>) w osi Z | 83 |
| 6.6 | Przebieg czasowy prezentujący pozycję wału silnika krokowego: zadaną, mierzoną i mierzoną przefiltrowaną przez KF | 85 |
| 6.7 | Przebieg czasowy (powiększenie przy impulsie skokowym) prezentujący pozycję wału silnika krokowego: zadaną, mierzoną i mierzoną przefiltrowaną przez KF | 86 |
| 6.8 | Przebieg czasowy (powiększenie przy impulsie skokowym) prezentujący pozycję wału silnika krokowego: zadaną, mierzoną nieprzefiltrowaną i mierzoną przefiltrowaną przez filtr dolnoprzepustowy trzeciego rzędu . . . | 87 |
| 6.9 | Schemat blokowy sterowania wysokopoziomowego silnikiem VCM, zaimplementowanego w kodzie sterownika | 89 |
| 6.10 | Schemat blokowy sterowania wysokopoziomowego silnikiem krokowym, zaimplementowanego w kodzie sterownika | 90 |
| 6.11 | Wygenerowana trajektoria w kształcie trapezu | 92 |
| 6.12 | Wygenerowana trajektoria dla obiektu poruszającego się ($v_0 > v_{max}$) . . . | 93 |
| 6.13 | Wygenerowana trajektoria dla obiektu poruszającego się w przeciwnym kierunku | 94 |
| 6.14 | Wizualizacja algorytmu obliczającego częstotliwość komutacji i bieg dla zadanej prędkości obrotowej wału | 95 |
| 6.15 | Teoretyczny model układu napędowego modułu z silnikiem VCM | 96 |
| 6.16 | Przebieg teoretycznego prądu (oś Y) wymaganego w uzwojeniu do wysuwu cylindra na zadaną odległość (oś X) | 97 |
| 6.17 | Charakterystyka prądowa zastosowanego silnika VCM odtworzona w programie używając charakterystyki z karty katalogowej | 98 |
| 6.18 | Oscylogramy mierzonych spadków napięć na rezystorach pomiarowych w mostkach H silnika krokowego poruszającego się ze stałą zadaną prędkością kątową | 99 |
| 6.19 | Przykładowe odpowiedzi regulatorów pozycji i prędkości silnika krokowego, z zastosowaniem generatora trajektorii | 100 |
| 6.20 | Przykładowe odpowiedzi regulatorów pozycji i prędkości silnika liniowego VCM, z zastosowaniem generatora trajektorii | 101 |
| 6.21 | Potencjały na zaciskach uzwojenia wraz z licznikiem danego timer'a . . . | 102 |
| 6.22 | Przepływ prądu w mostku H dla kolejnych komutacji (1, 2, 3, 4) przy ruchu elementu wykonawczego. a) Ruch w elemencie wykonawczego w prawo, b) ruch w lewo | 103 |
| 6.23 | Oscylogramy mierzonych spadków napięć na rezystorach pomiarowych w mostku H silnika krokowego z rys. 6.18 w przybliżeniu | 104 |
| 6.24 | Aplikacja (interfejs użytkownika) pozwalająca na zadawanie/odczytywanie parametrów i wywoływanie funkcji w sterowniku, korzystając z interfejsu CAN | 106 |

Spis kodów źródłowych

| | | |
|-----|---|----|
| 6.1 | Przykładowe logi informujące o wywołaniu funkcji poprzez CAN . . . | 81 |
| 6.2 | Obiekt typu <i>enum</i> prezentujący obsługiwane w sterowniku wydarzenia | 82 |
| 6.3 | Kod filtra dolnoprzepustowego wykonywany w pętli sterującej mikrokontrolera w interwale $dt = 0.001s$ | 88 |
| 6.4 | Kod regulatora pozycji zaimplementowany w mikrokontrolerze, wywoływany w interwale $dt = 0.001s$ | 91 |