

# HexapodProtocol

Hexapod Bionik

Luty 2024



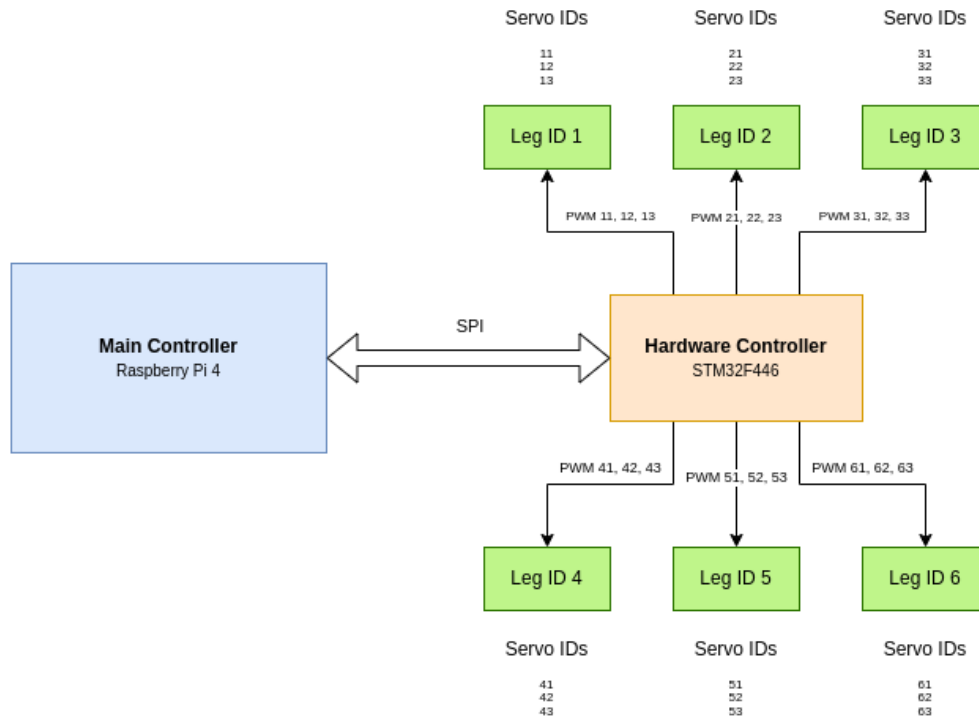
# Spis treści

<b>1</b>	<b>Informacje wstępne</b>	<b>3</b>
1.1	Krótki opis systemu Hexapod . . . . .	3
1.2	Główne założenia HexapodProtocol . . . . .	3
<b>2</b>	<b>HexapodProtocol v1.0.0</b>	<b>4</b>
2.1	Ramka typu ONE SERVO . . . . .	4
2.2	Ramka typu ONE LEG . . . . .	5

# 1 Informacje wstępne

## 1.1 Krótki opis systemu Hexapod

Zanim zostanie opisany sam protokół komunikacji, trzeba opisać komponenty występujące w systemie Hexapod, które odpowiadają za lokomocję robota.



Rysunek 1: Diagram ukazujący połączenia pomiędzy komponentami i odpowiadające im nazwy

W całym systemie nadrzędną rolę pełni główny kontroler, w tym przypadku jest to komputer jednopłytkowy Raspberry Pi 4B. To na nim wykonywane są wszystkie operacje wymagające większej mocy obliczeniowej, takie jak generowanie ścieżki robota czy lokalizacja. Po wyznaczeniu ścieżki do punktu docelowego, wysyła on z wykorzystaniem protokołu Hexapod komendy do sterownika sprzętowego, którego funkcję pełni mikrokontroler STM32F446. Są one następnie interpretowane i w zależności od typu komendy oraz dodatkowych danych pomocniczych wykonywane są określone akcje na określonych odnóżach lub serwomechanizmach.

Każde odnóże ma nadany numer identyfikacyjny od wartości 1 do wartości 6. Powiązane są z nimi numery identyfikacyjne poszczególnych serwomechanizmów wchodzących w skład kończyny. Konwencja jest następująca, pierwsza cyfra w numerze serwomechanizmu oznacza numer nogi, druga natomiast odpowiada numerowi stawu w łańcuchu kinematycznym, który jest przez ten silnik poruszany. Numeracja zaczyna się od wartości 1 licząc od korpusu, kończąc na wartości 3.

## 1.2 Główne założenia HexapodProtocol

Hexapod Protocol wykorzystuje w swoim działaniu protokół SPI. Główny kontroler pełni w nim funkcję mastera, natomiast sterownik sprzętowy pełni rolę slave'a. Połączenie realizowane jest z użyciem 4 sygnałów:

- MISO - Master In Slave Out
- MOSI - Master Out Slave In
- SCK - linia zegarowa
- NCS - zanegowana linia chip select, w trybie nieaktywnym w stanie wysokim, w trybie aktywnym w stanie niskim

Inne istotne informacje:

- Wykorzystywane są ramki typu motorola
- W jednej ramce protokołu SPI przekazywany jest 1 bajt o długości 8 bitów
- Pierwszym bitem w ramce jest bit najbardziej znaczący (MSB first)
- Nie jest generowany kod CRC

Parametry zegara:

- Polaryzacja zegara (Clock polarity) - niska (low) (w stanie nieaktywnym zegar pozostaje w logicznym zerze)
- Faza zegara (Clock phase) - zbocze narastające
- Maksymalna prędkość przesyłu danych obsługiwana przez sterownik sprzętowy - 20.0 MBits/s

Sam protokół Hexapod składa się z zestawu zdefiniowanych ramek, spośród których każda posiada unikatowy kod identyfikacyjny oraz długość.

## 2 HexapodProtocol v1.0.0

W tej wersji protokołu wykorzystywana jest transmisja z prędkością 2.5MBits/s. Spowodowane jest to ograniczeniami sprzętowymi wynikającymi z konieczności wykorzystywania dłuższych połączeń pomiędzy urządzeniem master i urządzeniem slave niż w projekcie docelowym oraz płytek prototypowych. W efekcie w sposób empiryczny oraz z wykorzystaniem odpowiedniej aparatury laboratoryjnej stwierdzono, że dla częstotliwości transmisji  $f = 2.5MHz$  sygnał zegarowy zachowuje swoje właściwości.

Protokół w wersji v1.0.0 daje możliwość sterowania manualnego serwomechanizmami z wykorzystaniem 2 typów ramek:

Typ ramki	Długość ramki	Kod identyfikacyjny
ONE LEG FRAME	14	1
ONE SERVO FRAME	6	2

1. One leg frame - ramka zawierająca informacje dla wszystkich 3 silników znajdujących się w jednym odnózu
2. One servo frame - ramka zaadresowana do konkretnego serwomechanizmu i zawierająca informacje odnoszące się bezpośrednio do niego

### 2.1 Ramka typu ONE SERVO

Podstawową ramką w tej wersji protokołu jest ramka typu one servo. Jej schemat wygląda następująco:

1 Byte	2 Byte	3 Byte	4 Byte	5 Byte	6 Byte
FRAME LEN	FRAME TYPE	SERVO ID	SERVO OP TYPE	INT PART	FLOAT PART

Pola FRAME LEN oraz FRAME TYPE zostały omówione w części ogólnej, więc nie będą one tutaj omawiane, jako że są one wspólne dla wszystkich typów ramek. Przypomnieć należy tylko, że pierwsze pole ma w tym przypadku wartość 6, drugie natomiast wartość 2.

Pole SERVO ID zawiera numer identyfikacyjny serwomechanizmu, zgodny z konwencją zawartą w ogólnych założeniach systemu Hexapod.

Pole SERVO OP TYPE zawiera informacje na temat typu operacji, która ma zostać wykonana przez silnik. Typy operacji, ich kody oraz krótki opis podano w tabeli poniżej.

Typ operacji	Kod operacji	Opis
Start & Set	1	Załącza serwomechanizm, ustawia jego pozycję wg pól INT PART oraz FLOAT PART
Stop	2	Wyłącza serwomechanizm, wartości pól INT PART oraz FLOAT PART nie mają znaczenia
Set	3	Ustawia pozycję serwomechanizmu wg pól INT PART oraz FLOAT PART

Pola INT PART oraz FLOAT PART zawierają odpowiednio całkowitoliczbową część kąta zadanego serwomechanizmowi oraz część zmiennoprzecinkową. Pierwsza powinna należeć do zakresu od 0 do 180 stopni, druga natomiast od 0 do 99 części setnych stopnia.

## 2.2 Ramka typu ONE LEG

Ramka typu ONE LEG jest swego rodzaju połączeniem 3 ramek typu ONE SERVO, a więc jej schemat wygląda następująco:

1 Byte	2 Byte	3 Byte	4 Byte	5 Byte	6 Byte	...
FRAME LEN	FRAME TYPE	SERVO ID 1	SERVO 1 OP TYPE	INT PART 1	FLOAT PART 1	...

11 Byte	12 Byte	13 Byte	14 Byte
SERVO ID 3	SERVO 3 OP TYPE	INT PART 3	FLOAT PART 3

Pola FRAME LEN oraz FRAME TYPE mają odpowiednio wartości 14 oraz 1. Następnie 3 razy sekwencja bajtów dla każdego serwa, zgodna z tą opisaną dla ramki typu ONE SERVO.