Moduł zawiera struktury danych i funkcje sterujące sterownikami silników krokowych napędzających osie robota. Każdy driver steruje osią za pomocą zegara sprzętowego. Parametry ruchu osi pochodzą z listy Job, które pobierane są funkcjami w module WORK. Uruchomienie osi następuje jeśli pozycja zadana różni się od pozycji aktualnej.

Parametry pozycji wyrażone są liczbie impulsów, i obliczane za pomocą funkcji ConvertAngleToStep lub ConvertAngleToPwm, ponieważ z telefonu otrzymywane są jako wartość kąta. Podczas inicjalizacji ustawiane są domyślne wartości minimalne i maksymalne oraz pozycja aktualna. Kiedy oś obraca się w prawo pozycja aktualna jest inkrementowana a podczas obrotów w lewo dekrementowana aż do osiągnięcia pozycji zadanej, maksymalnej lub minimalnej. Wstępnie zakres ruchu przewidziany jest na 180 stopni, nie każda oś na początku znajduje się w środku zakresu co należy uwzględnić podczas inicjalizacji.

**Struktura programowego sterownika silnika krokowego (stepper\_driver\_t):**

*uint16\_t* SetpointPosition; /\* pozycja zadana \*/

*uint16\_t* CurrentPosition; /\* aktualna pozycja \*/

*uint16\_t* MaximumPosition; /\* maksymalna wartość pozycji \*/

*uint16\_t* MinimumPosition; /\* minimalna wartość pozycji \*/

*uint8\_t* Direction; /\* kierunek ruchu \*/

TC1\_t \*DriverTimer; /\* timer sterownika \*/

PORT\_t \*DriverPort; /\* port pinów sterownika \*/

*uint8\_t* EnablePin; /\* numer pinu dla sygnału enable \*/

*uint8\_t* DirectionPin; /\* numer pinu dla sygnału dir \*/

*uint8\_t* PulsePin; /\* numer pinu dla sygnału pul \*/

*uint8\_t* IsRunning:1; /\* flaga informująca o pracy/zatrzymaniu sterownika \*/

*uint16\_t* MotorSteps; /\* Liczba kroków silnika na jeden obrót \*/

*uint8\_t* ElectricalRatio; /\* wartość przełożenia elektrycznego (ilość impulsów na krok, ustawienie pinów na sterowniku) \*/

float MechanicalRatio; /\* wartość przełożenia mechanicznego obroty przekładni na obrót silnika \*/

*uint8\_t* Speed; /\* prędkość zadana [obr/min] \*/

*(void\*)* (Start) *(void)* /\* wskaźnik na funkcję start \*/

*(void\*)* (Stop) *(void)* /\* wskaźnik na funkcję stop \*/

*(void\*)* (Convert) *(uint8\_t, void\*)* /\* wskaźnik na funkcję convert \*/

Struktura programowego sterownika servosilnika (servo\_driver\_t):

*int16\_t* SetpointPosition; /\* pozycja zadana \*/

*uint16\_t* MaximumPosition; /\* maksymalna wartość pozycji \*/

*int16\_t* MinimumPosition; /\* minimalna wartość pozycji \*/

TC0\_t \*DriverTimer; /\* timer sterownika \*/

PORT\_t \*DriverPort; /\* port pinów sterownika \*/

*uint8\_t* IsRunning:1; /\* flaga informująca o pracy/zatrzymaniu sterownika \*/

*(void\*)* (Start) *(void)* /\* wskaźnik na funkcję start \*/

*(void\*)* (Stop) *(void)* /\* wskaźnik na funkcję stop \*/

*(void\*)* (Convert) *(uint8\_t, void\*)* /\* wskaźnik na funkcję convert \*/

**Wymagane funkcje:**

1. AxesInit – incjalizacja sterowników wszystkich osi.
2. StepperDriverInit – inicjalizacja programowego sterownika krokówki.
3. ServoDriverInit – inicjalizacja programowego sterownika servosilnika.
4. RunDriver – uruchomienie sterownika.
5. StopDriver – zatrzymanie sterownika.
6. SetParameters – ustawienie parametrów ruchu.
7. ConvertAngleToStep – konwersja kąta na kroki silnika.