

Ćwiczenie 5

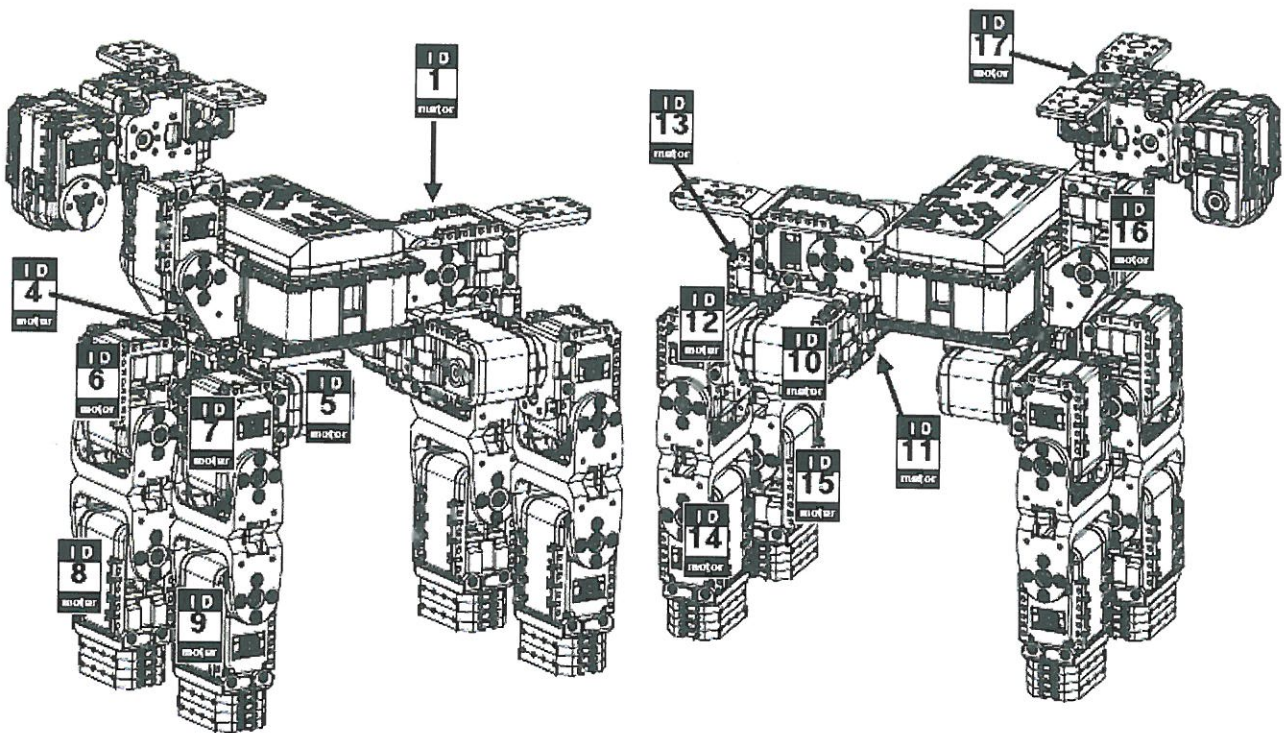
Podstawy programowania robota Bioloid firmy Robotis na przykładzie konstrukcji Puppy

1. Wprowadzenie

Celem ćwiczenia jest zapoznanie się oraz zrozumienie metod programowania robota za pomocą "Behavior control programmer" (programowanie za pomocą odpowiednich komend określających zachowania robota) oraz "Motion Editor" (programowanie poszczególnych pozycji robota).

2. Opis robota

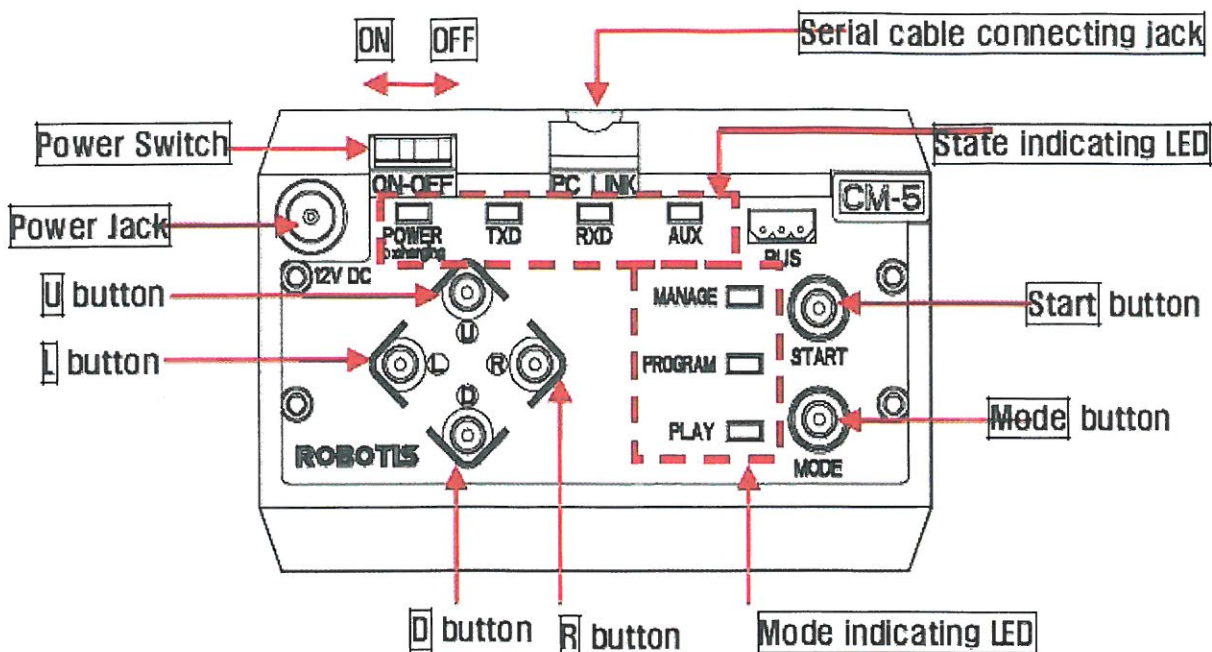
System Bioloid firmy Robotis to zestaw napędów (18 serw AX-12+), czujników (moduł AX-S1 zawierający natężenia światła, czujniki odległości podczerwieni, dźwięku oraz głośnik) oraz elementów konstrukcyjnych (różne kształty szkieletów, kable, śrubki i nakrętki) umożliwiające budowanie różnorodnych robotów. W bieżącym ćwiczeniu zmontowany został czteronożny robot w kształcie psa – Puppy, pokazany na Rys 6.1. Nogi robota mają 3DOF, głowa 2DOF, tułów 1DOF. Mózgiem Bioloid'a jest blok CM-5 zawierający komputer oraz akumulator będący źródłem zasilania. Ponadto w głowie robota zamocowany jest moduł sensorów AX-S1.



Rys 6.1 Robot Bioloid w wersji Puppy, zaznaczono identyfikatory poszczególnych napędów

3. Opis bloku sterującego CM-5:

Blok CM-5 jest kontrolerem bloków AX-12+ i AX-S1. Widok bloku sterującego CM-5 z opisem funkcji poszczególnych przycisków pokazano na Rys. 6.2.



Rys 6.2 Sterownik CM-5 robota Bioloid

Robot komunikuje się z komputerem PC za pomocą kabla RS-232 podłączonego do wejścia "Serial cable connecting jack". Robot posiada trzy tryby pracy zmieniane za pomocą przycisku "Mode": "Manage", "Program" oraz "Play" - są one sygnalizowane za pomocą migających diod, przy nazwach konkretnych trybów pracy. Aby przejść ze stanu czuwania (migająca dioda), należy przycisnąć przycisk "Start" - wtedy dioda zapala się. Tryby pracy:

- "Manage Mode" - do weryfikacji stanu poszczególnych bloków oraz testowania ruchów. Opcja zaawansowana.
- "Program Mode" - do edycji ruchów. Jednostka sterująca CM-5 przekształca zmienne w dany tryb pracy, jeżeli używany jest "Motion editor".
- "Play Mode" - uruchamiany po przesłaniu przez "Behavior control programmer" programu użytkownika do pamięci robota.

!!! W celu zatrzymania pracy robota naciskamy "Mode button" lub wyłączamy robota za pomocą przełącznika "Power Switch" !!!

Znaczenie pozostałych diod:

- "Power" - dioda włączona, gdy robot włączony; mruga, gdy robot ładowany.
- "TXD" - dioda włączona, gdy CM-5 przesyła dane do urządzenia zewnętrznego.
- "RXD" - dioda włączona, gdy CM-5 otrzymuje dane z urządzenia zewnętrznego.
- "AUX" - do wykorzystania przy programowaniu za pomocą "Behavior control programmer".

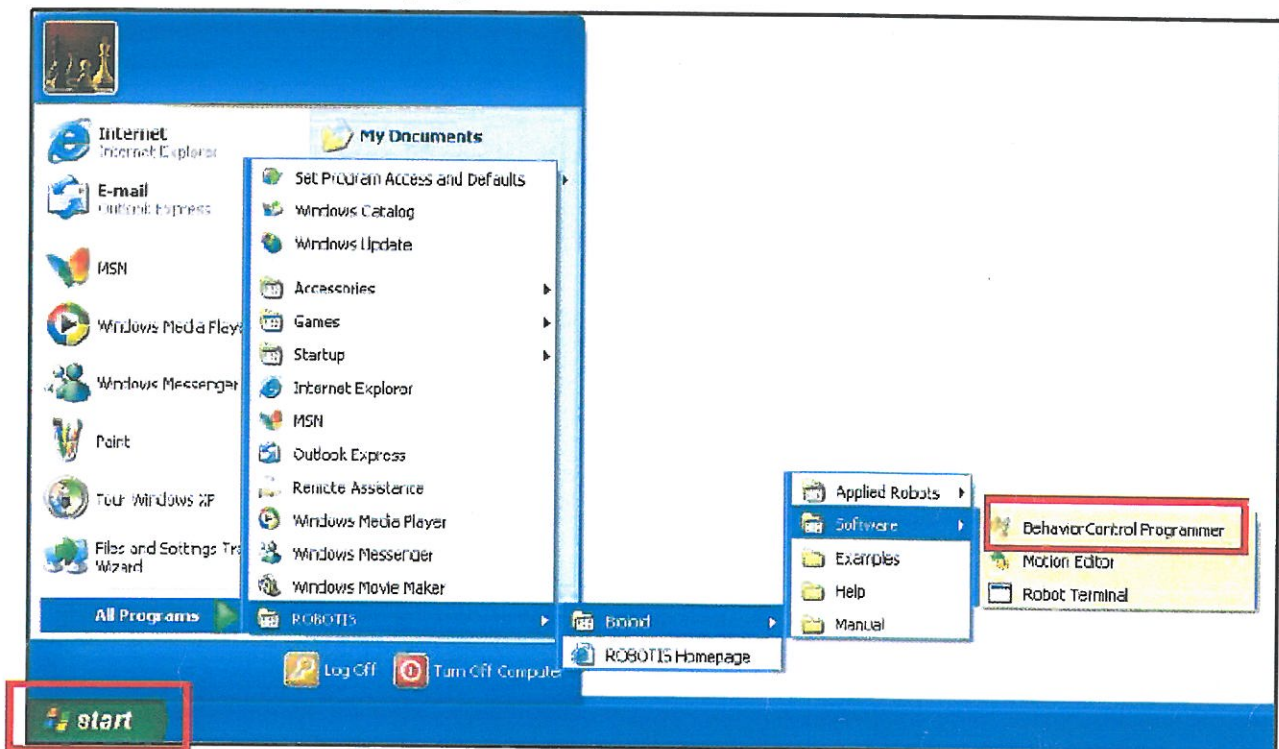
Przyciski U, D, L i R są wykorzystywane przy programowaniu, np. przy zdalnym sterowaniu robotem można zmieniać tryby wykonywanych procedur przez robota.

4. Behavior control programmer:

Jest to język programowania (uproszczony za pomocą obrazków), w którym można zapisać program wykonywany później przez robota. Zapisujemy tam wzorzec, zgodnie z którym powinien działać robot.

1) Podstawy obsługi programu:

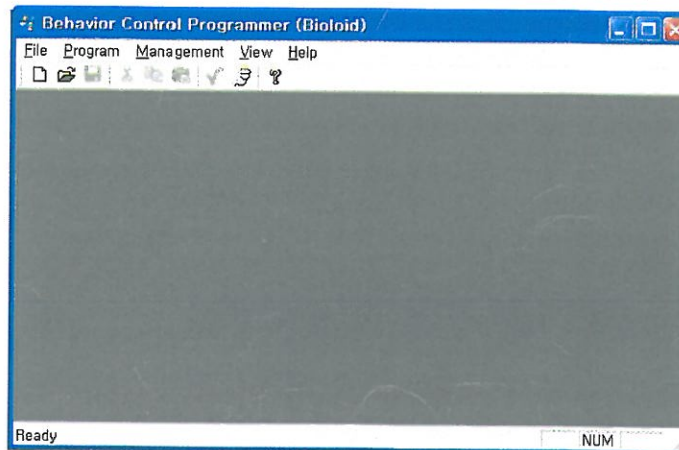
Jak włączyć "Behavior control programmer":



Rys 6.3 Gdzie znaleźć Behavior control programmer

Strona podstawowa "Behavior control programmer":

Initial page of behavior control programmer



Rys 6.4 Behavior control programmer zaraz po uruchomieniu

Jest to program kompatybilny z okienkowym systemem Windows i obsługa podstawowych poleceń (nowy dokument, otwórz, zapisz, itp.) nie różni się od obsługi np. programu OpenOffice.

Warto zwrócić uwagę, że po otwarciu nowego dokumentu pojawiają się numerowane linie programu. Aby wpisać w nie odpowiednie polecenie trzeba dwukrotnie kliknąć w pole "Double click".

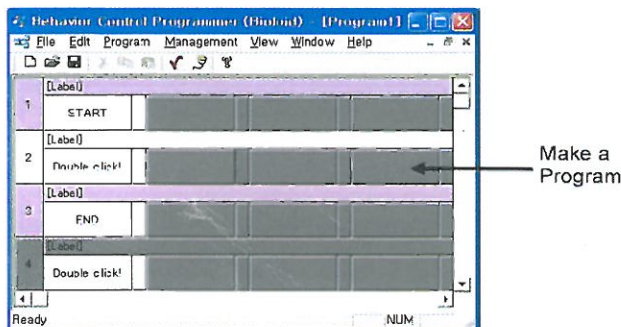
2) Język programowania:

"Behavior control programmer" posiada 11 poleceń podzielonych na 3 typy:

Typ	Komenda	Funkcja
Operacja	START	do wskazywania początku programu
	END	do wskazywania końca programu
	LOAD	do doprowadzenia różnych rodzajów danych
	COMPUTE	do obsługi czterech operacji arytmetycznych oraz operacji logicznych
Stan	IF	do obsługi następujących komend, gdy podany stan jest "prawdą"
	ELSE IF	stawiany za IF, do obsługi następujących komend, gdy podany stan IF jest "fałszem", a stan ELSE IF jest "prawdą"
	ELSE	stawiany za IF lub ELSE IF, do obsługi następujących komend po ELSE, gdy obydwa stany IF i ELSE IF są "fałszywe"
	CONT IF	do kontynuacji polecenia IF (rozwińcie polecenia IF) - wielokrotny warunek IF
Odgałęzienie	JUMP	do przeskoczenia do wyznaczonego punktu w programie
	CALL	do wywołania wyznaczonego polecenia
	RETURN	do powrotu programu do następnego wiersza po poleceniu CALL, po wykonaniu sekwencji poleceń procedury wywołanej przez polecenie CALL

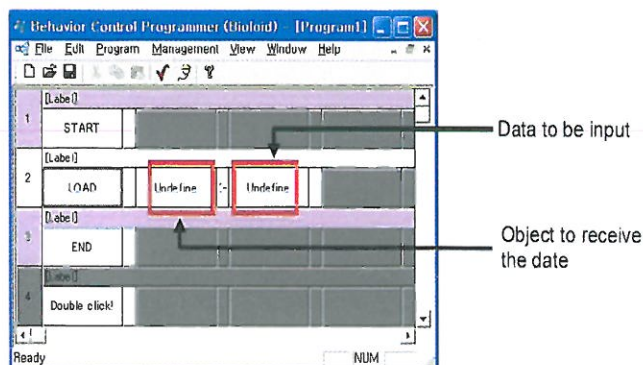
Przykłady:

- START, END - cały program powinien zawierać się między komendami START i END



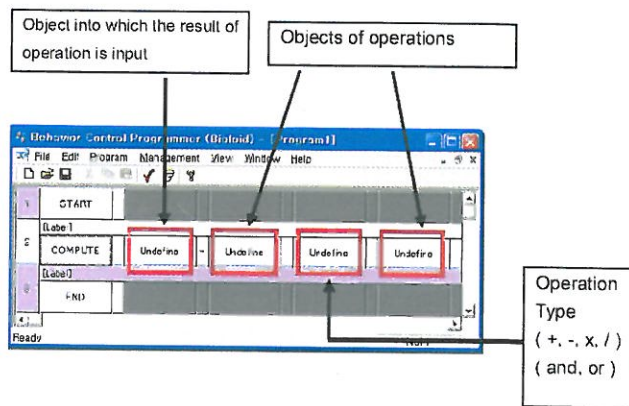
Rys 6.5 Struktura programu użytkownika w Behavior control programmer

- LOAD - po wprowadzeniu tego polecenia pojawiają się dwa bloki (od lewej strony): blok odczytywania danych i blok do wpisywania danych. Po dwukrotnym wciśnięciu pola "Undefine" rozwinięta zostaje lista: CM-5 (podrozdział 4.4), Typing (wpisać z klawiatury) oraz Dynamixel (odnosi się do serw AX-12+ - podrozdział 4.5 oraz bloku AX-S1 - podrozdział 4.6)



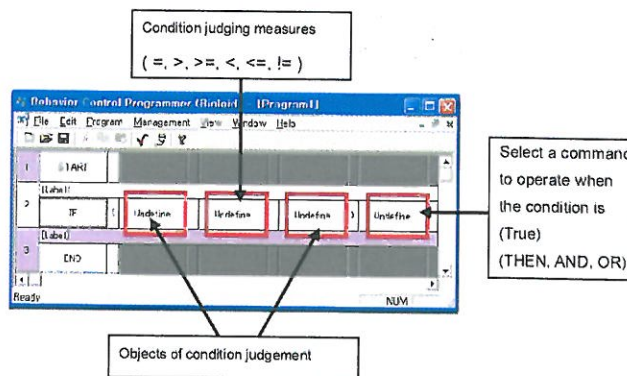
Rys 6.6 Struktura programu użytkownika w Behavior control programmer

- COMPUTE - po wprowadzeniu tego polecenia pojawiają się cztery bloki (od lewej strony): blok, w którym wynik operacji jest wejściem; blok operacji; typ operacji (+, -, x, /)(and, or); blok operacji.



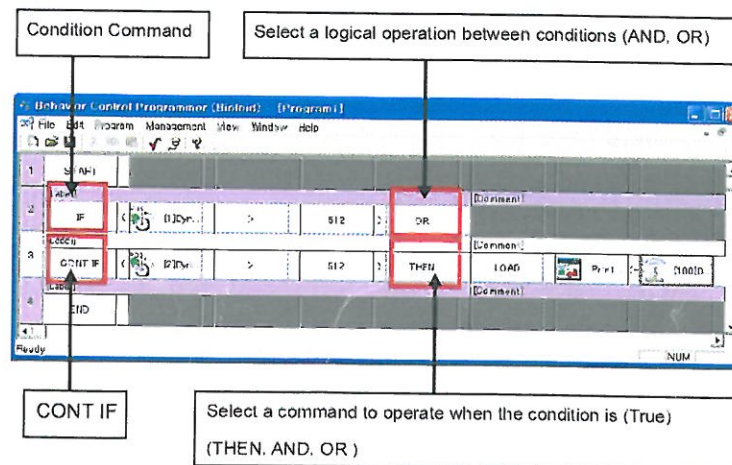
Rys 6.7 Struktura programu użytkownika w Behavior control programmer

- IF - po wprowadzeniu tego polecenia pojawiają się cztery bloki (od lewej strony): pierwszy blok do porównania; znak porównania (=, >, >=, <, <=, !=); drugi blok do porównania; wybrana komenda (THEN, AND, OR) do wykonania, jeżeli wynik to "prawda"



Rys 6.8 Struktura funkcji IF w Behavior control programmer

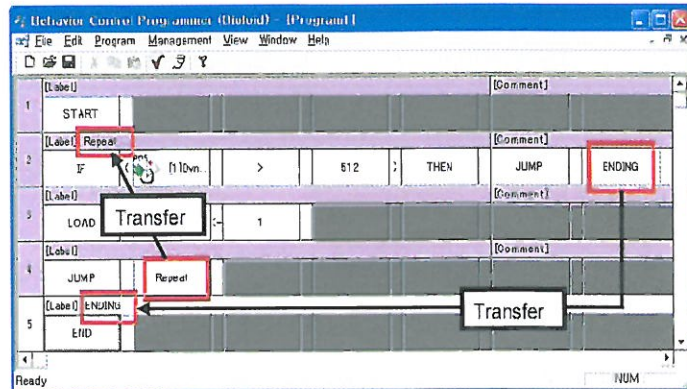
- ELSE IF - składnia taka sama jak przy poleceniu IF
- CONT IF - można użyć polecenia do oceny wielu warunków w tym samym czasie, np:



Rys 6.9 Struktura funkcji IF w Behavior control programmer

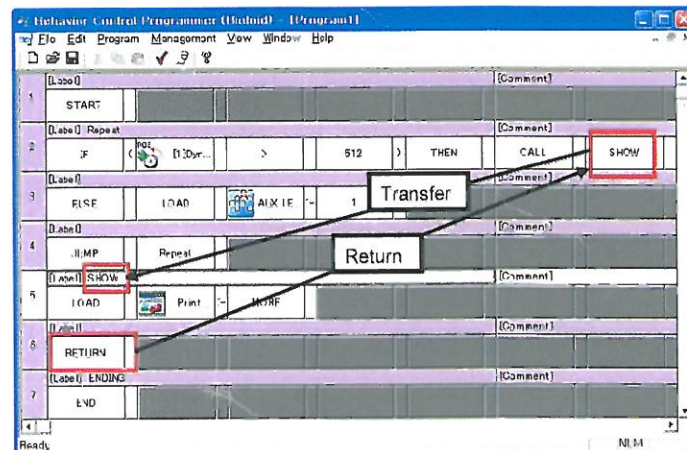
- ELSE - kiedy wszystkie pozostałe warunki są "fałszywe" wykonywane jest polecenie w ELSE (tylko jeden blok)

- JUMP - zmienia kolejność wykonywania poleceń (wierszy). W bloku na prawo od polecenia JUMP wpisujemy nazwę konkretnego wiersza (klikamy dwa razy na [Label] w pożądanym wierszu) - wtedy skok następuje do danego wiersza



Rys 6.10 Struktura funkcji JUMP w Behavior control programmer

- CALL, RETURN - upraszcza program, pozwala na zgrupowanie poleceń odpowiadających za np. ruch nogi, który jest wielokrotnie wykonywany w programie

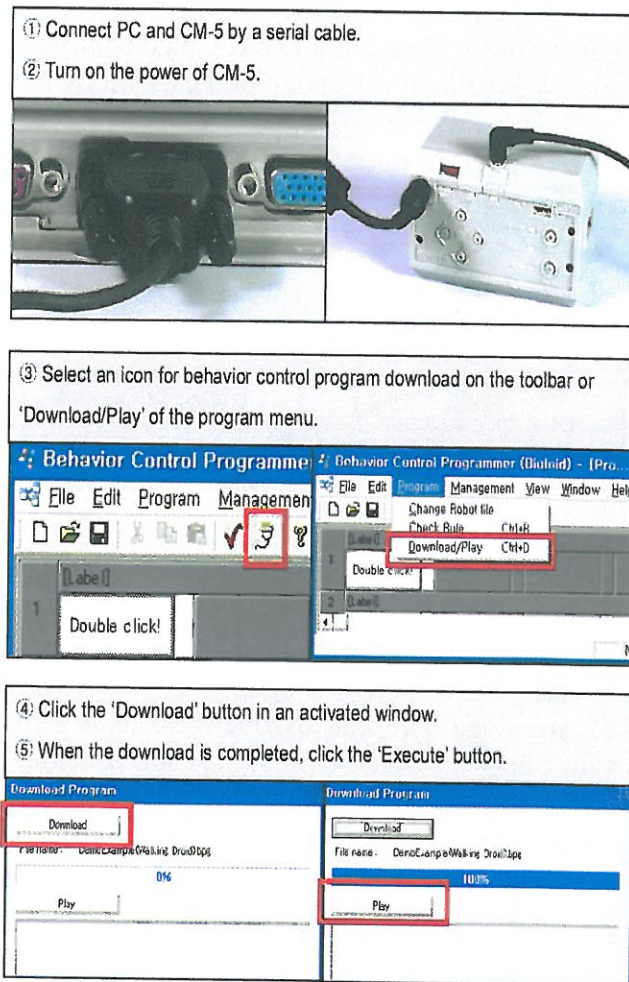


Rys 6.11 Struktura funkcji CALL w Behavior control programmer

1) Sprawdzenie i uruchomienie programu

Po napisaniu programu należy go skompilować, aby sprawdzić czy podana składnia nie zawiera błędów. Wykorzystujemy do tego ikonkę \checkmark , skrót klawiszowy CTRL+R lub z menu Program->Check Rule. Jeżeli wystąpi błąd, blok niezgodny z procedurą zostanie podświetlony na czerwono. Blok modyfikujemy klikając na niego najpierw lewym klawiszem myszki, żeby go podświetlić, a następnie prawym, żeby dokonać właściwej modyfikacji.

Jeżeli program jest dobrze skompilowany, możemy przesłać go do jednostki centralnej CM-5 za pomocą ikonki z kablem RS-232, skrót klawiszowy CTRL+D lub z menu Program->Download/Play jak pokazano na Rys 6.12.



Rys 6.12 Procedura załadowania programu użytkownika do sterownika w Behavior control programmer

Pomoc przy programowaniu:

4) Kontrola CM-5

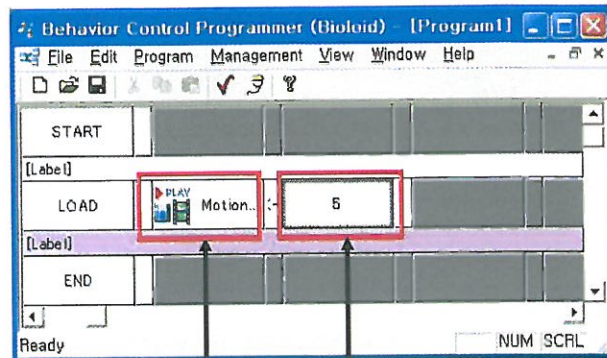
"Behavior control programmer" ma wbudowane następujące ikony oraz funkcje do zarządzania CM-5:

Ikona	Nazwa	Funkcja
	Motion play page	Kiedy wstawiś numer strony ruchu, wystartuje odpowiedni ruch z Motion program
	Motion play status	Ma wartość "1" dopóki wykonywany jest ruch, a "0" gdy nie jest wykonywany
	AUX LED	Włączony gdy ustawiona jest "1", a wyłączony gdy - "0"
	CM-5 Button	Pięć przycisków umieszczonych na jednostce sterującej są wykorzystywane jako urządzenia wejściowe programu
	Print	Wartość wejściowa jest pokazywana na ekranie
	Print with line feed	Po wyświetleniu wartości wejściowej, linia jest zmieniana na jedną niżej (print+enter)
	Timer	Wartość wejściowa jest zmniejszana o 1 co 0,125s

Przykłady:

- Motion play page - ruch stworzony w programie "Motion editor" (rozdział 5), może być wywołany w "Behavior control programmer", za pomocą tego oraz kolejnego polecenia.

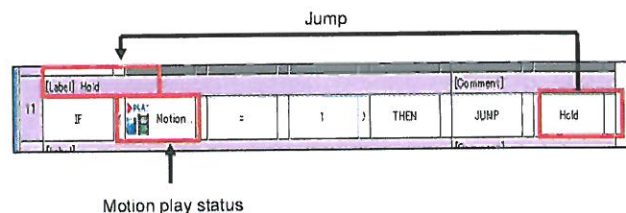
Używając komendy LOAD możemy zobrazować dane polecenie. Klikamy dwukrotnie na pierwszy wolny blok na prawo od komendy LOAD, wybieramy z rozwijanej listy CM-5 i dalej Motion play page. W kolejnym wolnym bloku ustawiamy numer strony, która ma zostać wywołana: dwukrotnie klikamy, wybieramy z rozwijanej listy Typing i wpisujemy z klawiatury numer strony.



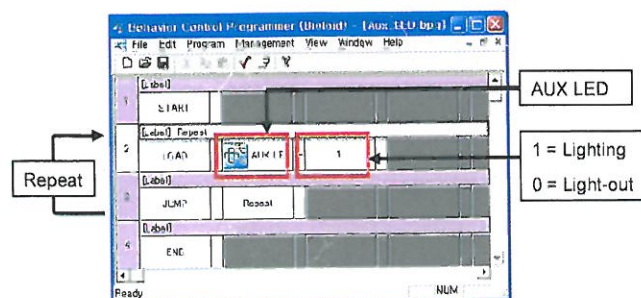
Icon of Motion play page Motion Page Number

Warto pamiętać, że wywoływane strony nie ograniczają się jedynie do 7 pozycji (tyle mieści się na jednej stronie), ale mogą być przenoszone na kolejne strony, co zostanie opisane w rozdziale 5.2. Idąc tym tropem, można w prosty sposób ustawić, aby robot powtarzał daną stronę w kółko (lub konkretny zestaw stron, np. 4-2-3-6-4-2-3-6...).

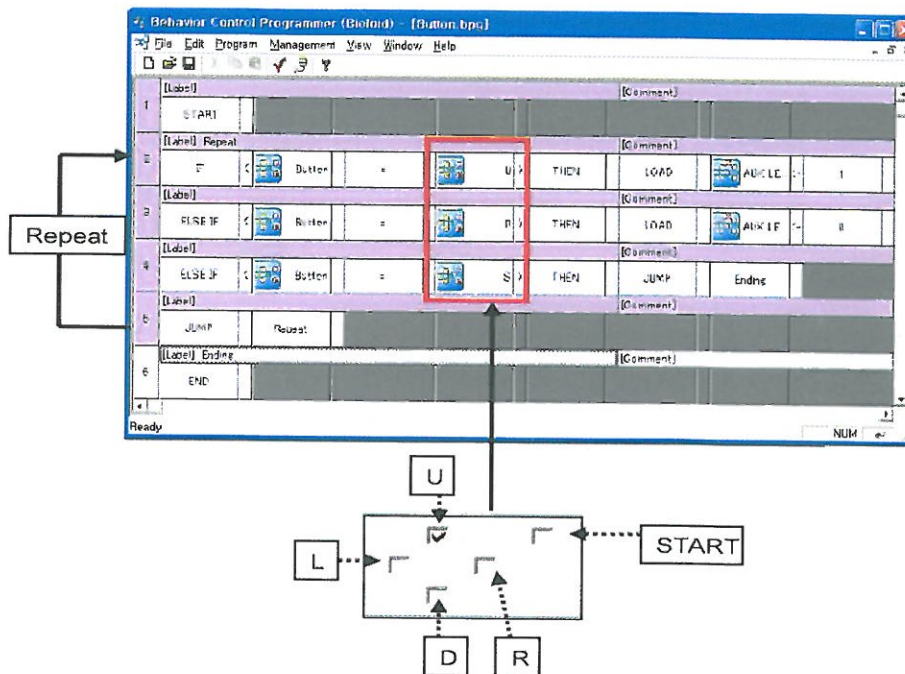
- Motion play status - warto zapamiętać, że potrzeba określonego czasu do wykonania ruchu. "Behavior control programmer" wywołuje polecenie wykonania pewnego ruchu (np. Motion play page strona 1) i od razu przechodzi do polecenia w kolejnym wierszu, nie wiedząc, że ruch z poprzedniego polecenia mógł nie zostać wykonany. Dzięki Motion play status można "zapętlić" "Behavior control programmer" aż do zakończenia ruchu z poprzedniego wiersza:



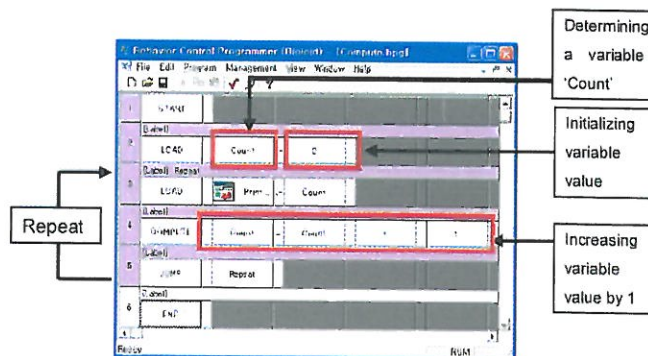
- AUX LED - załączenie diody AUX (od tego momentu, przykłady będą opisywane słownie tylko wtedy, gdy wprowadzane będą nowe funkcje; pozostałe funkcje opisane zostały w powyższych programach).



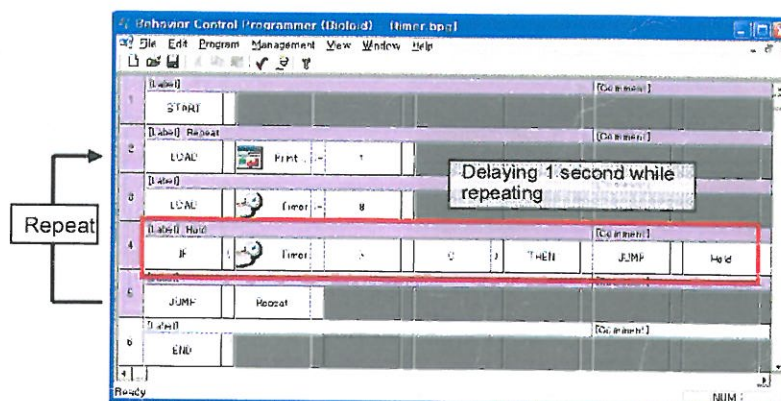
- CM-5 Button - dioda AUX włączana przyciskiem U, wyłączana przyciskiem D, a program kończy zadanie, gdy naciśniemy przycisk "START".



- Print i Print with line feed - wyświetlanie ekranie liczb rosnących od 0 co wartość 1. Należy zadeklarować zmienną "Count" (wprowadzana z klawiatury po wybraniu z rozwijanej listy pozycji Typing) poprzez wykonanie wiersza 2.







- Timer - wyświetlaj jedynkę na ekranie co każdą sekundę. Wartość timera ustawiona na 8,



ponieważ $8 \cdot 0,125s = 1s$ - timer zmienia wartość co 0,125s.

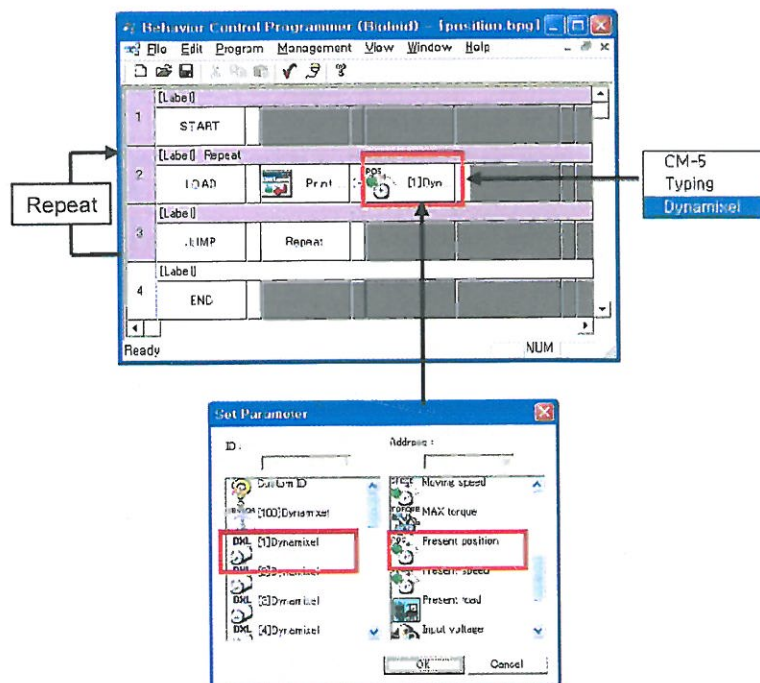
5) Kontrola AX-12+

"Behavior control programmer" ma wbudowane następujące (wybrane) ikony oraz funkcje do zarządzania AX-12+:

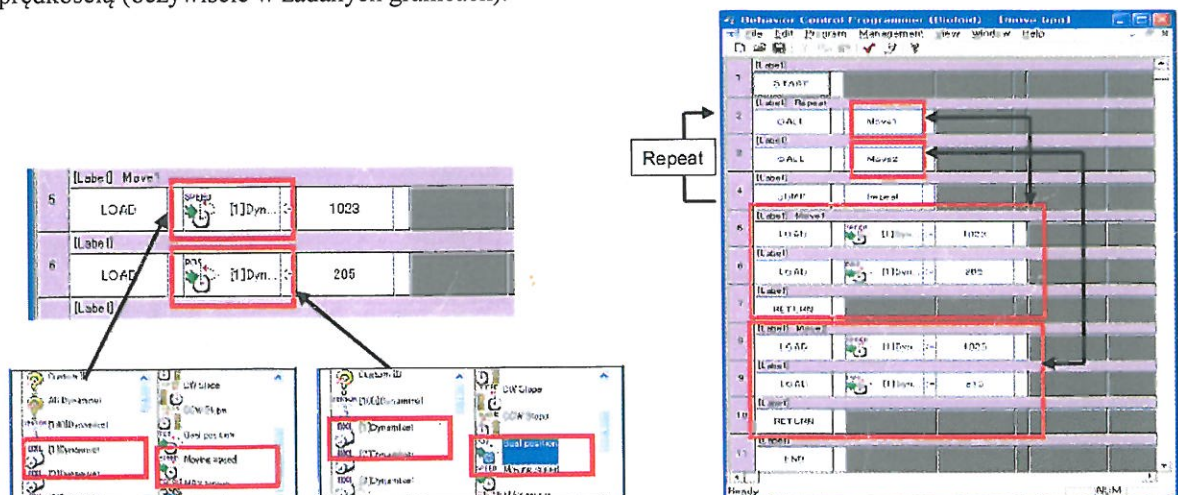
Ikona	Nazwa	Funkcja
	Present position	Wartość aktualnej pozycji serwa (od 0 do 1023)
	Present speed	Aktualna prędkość (od 0 do 1023)
	Goal position	Doprowadź do pozycji od 0 do 300° (od 0 do 1023)
	Moving speed	Wartość prędkości podczas ruchu (od 0 do 1023)

Przykłady:

- Present position - serwa AX-12+ kontrolują pozycję między 0, a 300°, co odpowiada wartości tej funkcji od 0 do 1023. Korzystając z funkcji LOAD, wprowadzamy do wolnego bloku (na prawo od LOAD) poznaną wcześniej funkcję Print with line feed, a w następnym wolnym bloku - funkcję Present position. W tym celu wybieramy z rozwijanej listy pozycję Dynamixel, a następnie z okna Set Parameter z lewej listy numer serwa, a z prawej kolumny Present position.














- Goal position i Moving speed - za pomocą tych funkcji można obrócić serwo do żądanej pozycji z wybraną prędkością (oczywiście w zadanych granicach).



!!! Zamiast słowa "Repeat" w wierszu 4 proszę wprowadzić słowo "Ending", a następnie nazwać wiersz 11 tym samym słowem (dwukrotnie kliknąć prawym klawiszem na [Label]), ze względu na zbyt krótki kabel przy ramieniu robota !!!

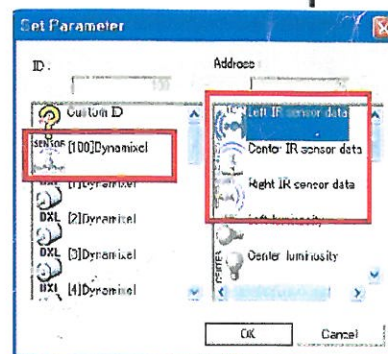
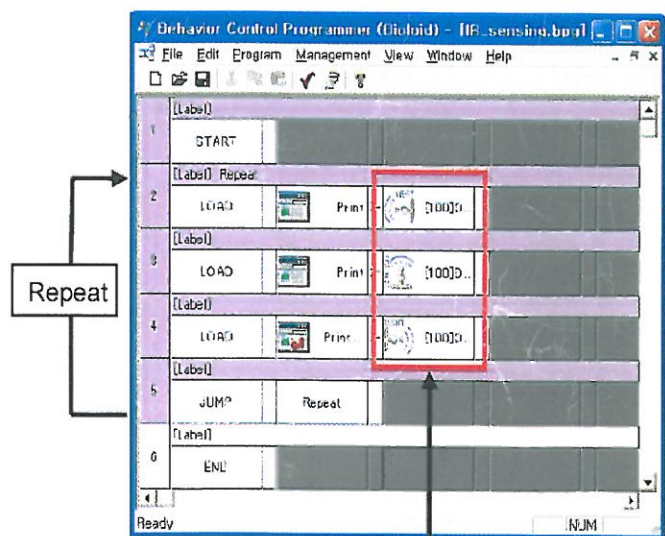
6) Kontrola AX-S1

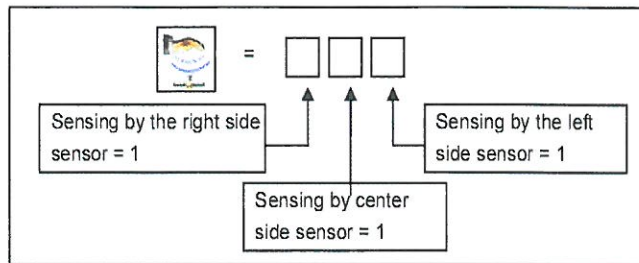
"Behavior control programmer" ma wbudowane następujące (wybrane) ikony oraz funkcje do zarządzania AX-S1:

Ikona	Nazwa	Funkcja
	Left IR sensor data	Odczyt wartości z lewej strony czujnika odległości
	Center IR sensor data	Odczyt wartości z przodu czujnika odległości
	Right IR sensor data	Odczyt wartości z prawej strony czujnika odległości
	Obstacle sensor	Włączony gdy wartość czujnika odległości większa niż standardowa
	Obstacle detected compare	Standard danych powyższej funkcji
	Sound data	Poziom głośności odczytany przez czujnik dźwięku
	Sound data max hold	Najgłośniejszy dźwięk, który został odczytany przez czujnik
	Sound detected count	Liczba dźwięków odczytanych przez czujnik (np. liczba kłaśnieć)
	Sound detected time	Czas trwania dźwięku
	Buzzer index	Wartości od 0 do 52 (wartość 0 odpowiada nucie 'La')
	Buzzer time	Jedna jednostka to 0,1s. Maksymalna wartość 5s. Minimalna wartość 0,3s.

Przykłady:

- IR sensor data - im bliżej przeszkody znajduje się czujnik, tym odczytywana wartość rośnie. Żeby zobrazować działanie czujników odległości należy wykonać poniższy przykład i zbliżając przeszkodę do poszczególnych czujników zaobserwować zmiany wskazań odczytów czujników na ekranie.

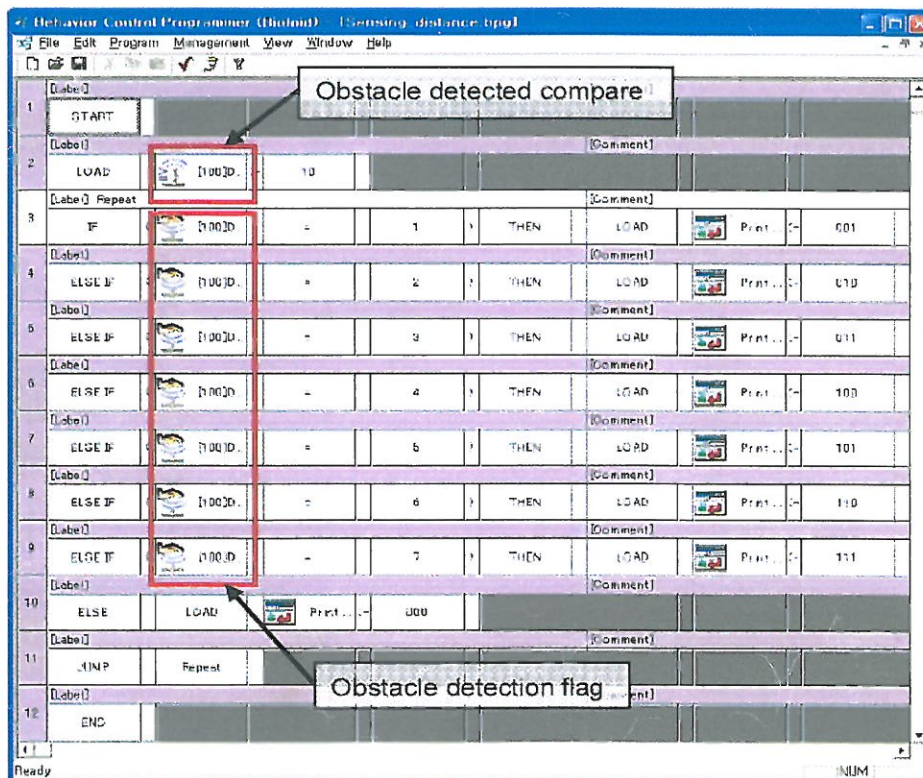




- Obstacle sensor - określa z której strony znajduje się przeszkoda zgodnie z tabelą:

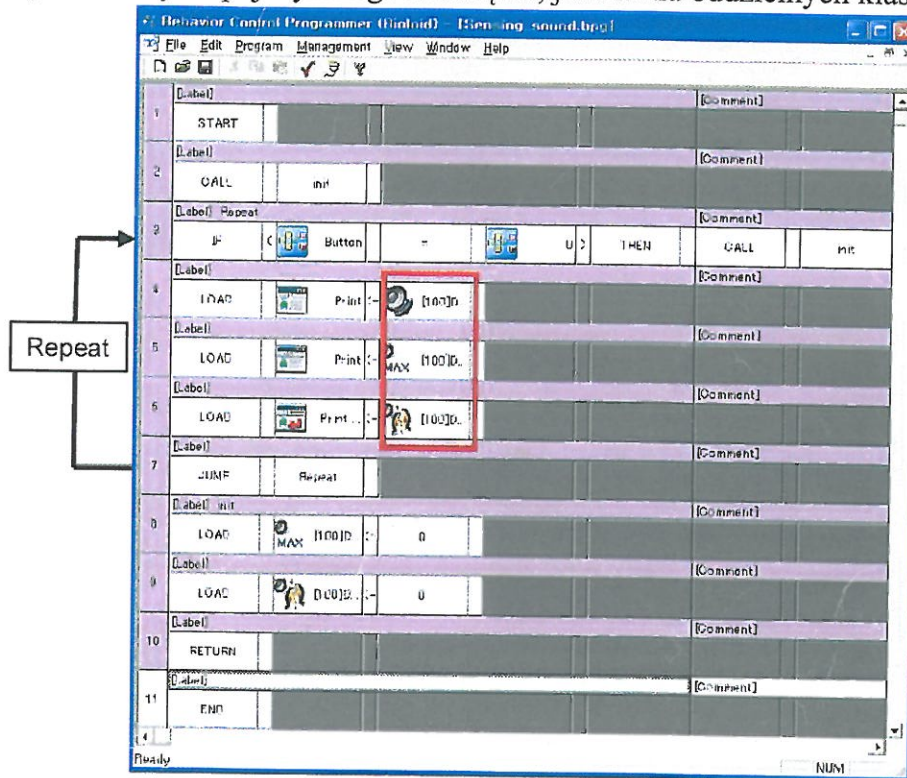
Wartość binarna	Wartość dziesiętna	Znaczenie Obstacle sensor
000	0	Przeszkoda nie wykryta
001	1	Lewy sensor wykrył przeszkodę
010	2	Środkowy sensor wykrył przeszkodę
011	3	Lewy i środkowy sensor wykrył przeszkodę
100	4	Prawy sensor wykrył przeszkodę
101	5	Lewy i prawy sensor wykrył przeszkodę
110	6	Prawy i środkowy sensor wykrył przeszkodę
111	7	Wszystkie sensory wykryły przeszkodę

- Obstacle detected compare - określa dystans przeszkody od robota, poniżej którego Obstacle sensor wykrywa przeszkodę.



- Sound data - jeżeli nie ma odgłosów, wartość utrzymuje się na poziomie około 128. Im głośniejszy dźwięk, tym wartość bardziej zbliża się do wartości 255. Dane dźwiękowe są próbkowane około 3800 razy na sekundę.
- Sound detected count - funkcja zliczająca dźwięki, które mają natężenie większe niż

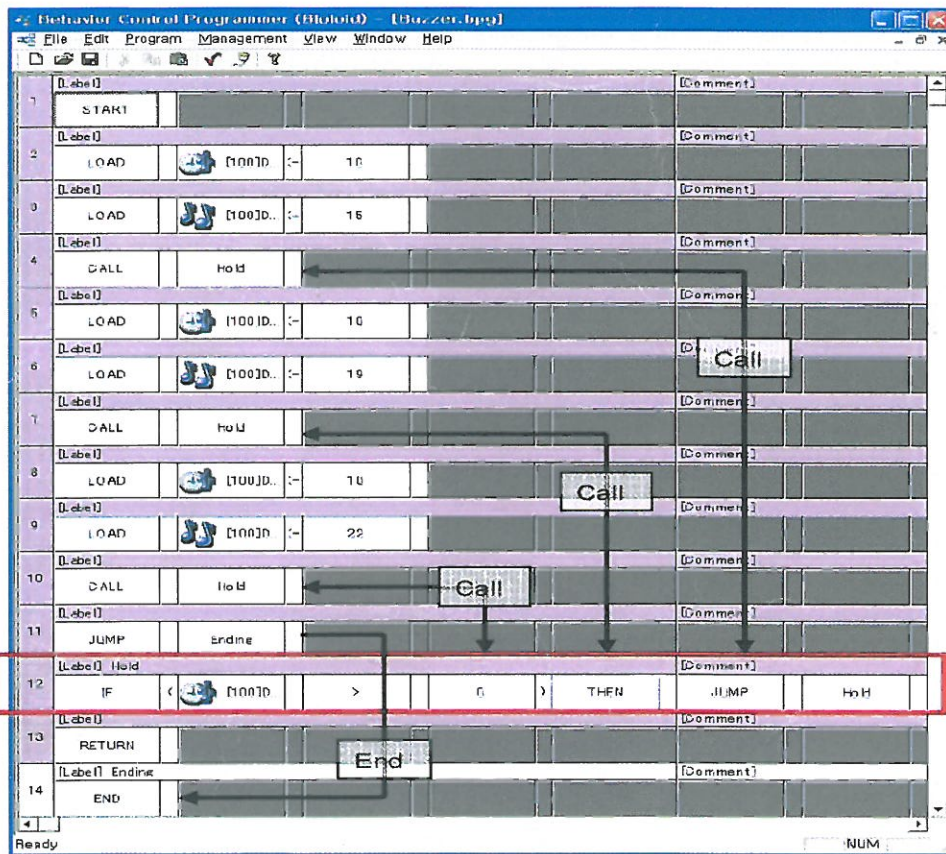
założone. Przy odczytywaniu klaśnięcia, przez 80ms od początku klaśnięcia funkcja nie zlicza, żeby nie odczytać pojedynczego klaśnięcia, jako kilku oddzielnych klaśnięć.



- Buzzer index - tabela dźwięków możliwych do uzyskania:

Wartość buzzera	Nuta	Wartość buzzera	Nuta	Wartość buzzera	Nuta	Wartość buzzera	Nuta
0	La ^A	15	Do	27	Do	39	Do
1	Ra#	16	Do#	28	Do#	40	Do#
2	Si ^B	17	Re	29	Re	41	Re
3	Do ^C	18	Re#	30	Re#	42	Re#
4	Do#	19	Mi	31	Mi	43	Mi
5	Re ^D	20	Fa	32	Fa	44	Fa
6	Re#	21	Fa#	33	Fa#	45	Fa#
7	Mi ^E	22	Sol	34	Sol	46	Sol
8	Fa ^F	23	Sol#	35	Sol#	47	Sol#
9	Fa#	24	La	36	La	48	La
10	Sol ^G	25	La#	37	La#	49	La#
11	Sol#	26	Si	38	Si	50	Si
12	La ^A					51	Do
13	La#						
14	Si ^B						

Przykład: każda z nut "Do", "Mi" i "Sol" grana przez 1s (należy pamiętać, żeby "Behavior control programmer" nie zaczął wykonywać kolejnego polecenia zanim nie skończy wykonywać poprzedniego - do tego wywoływany jest wiersz "Hold", który zatrzymuje wykonywanie kolejnych wierszy programu do momentu, aż dźwięk buzzera nie ucichnie)

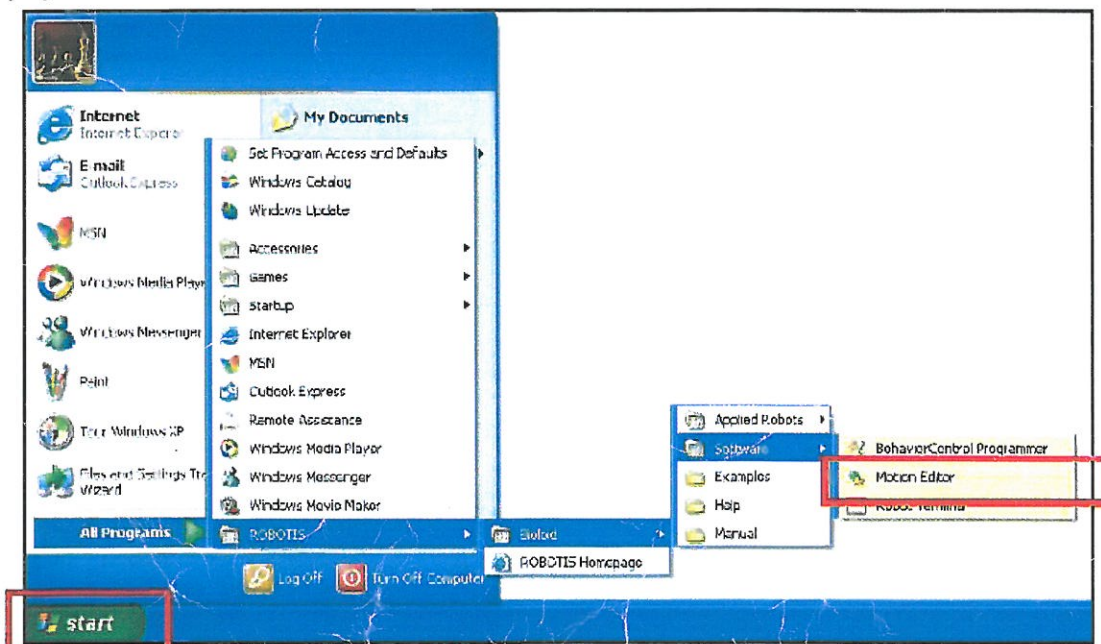


5. "Motion editor"

Jest to program, w którym użytkownik ustawia pojedyncze pozycje robota, co tworzy poszczególne kroki. Wprowadzanie poszczególnych pozycji może być manualne (fizyczne ustawianie poszczególnych złącz) lub automatyczne (wprowadzanie przez użytkownika wartości przy konkretnych złączach przez program "Motion editor").

1) Podstawy obsługi programu:

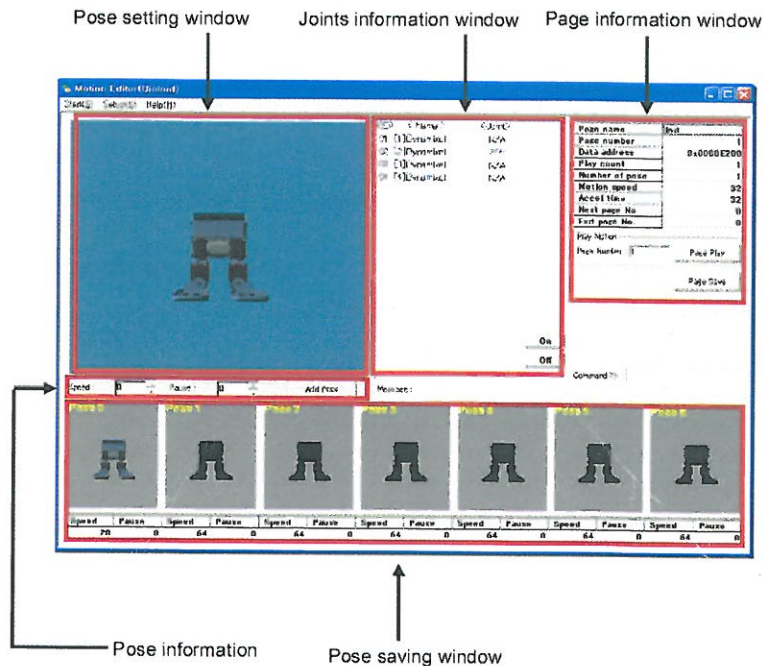
Jak włączyć "Motion editor":



2) Interfejs programu

Interfejs programu składa się z:

- okna ustawienia pozycji robota
- okna informacji o nastawach serw
- okna informacji o stronie
- informacji o pozycji (długość pauzy przed wykonaniem pozycji, prędkość wykonania pozycji)
- okna zachowanych pozycji na stronie



Przykład: "Walking Droid"

3) *Informacje o stronie* - można modyfikować dwukrotnie klikając na żądane pole. Warto zapamiętać, że aby zatwierdzić zmianę należy nacisnąć klawisz enter, a nie od razu klikać na inne pole (wtedy zmiana nie zostanie zachowana).

Page name	Init
Page number	1
Data address	0x0000E200
Play count	1
Number of pose	1
Motion speed	32
Accel time	32
Next page No.	0
Exit page No.	0
Play Motion	
Page Number :	1
Page Play	
Page Save	

Opis poszczególnych pozycji w stronie:

1. Page name - nazwa strony. Początkowa wartość bez nazwy.
2. Page number - numer strony (od 1 do 127). CM-5 może wykonać i zachować 127 stron.
3. Data address - pokazuje lokalizację strony w pamięci. Zaawansowane.
4. Play count - ustawia ile razy ma być wykonywana pozycja na stronie. Początkowa wartość 1.
5. Number of pose - wykonuje ruch od pozycji 0 do ustawionej w tym oknie pozycji. Pozycje poza nim zostają podświetlone na czarno.
6. Motion speed - reguluje wartością prędkości, z którą wykonywana jest dana strona. Wartość początkowa 32. Zalecana wartość przy ćwiczeniach 10.
7. Acceleration Time - gdy ruch jest wykonywany, wszystkie serwa pracują zgodnie z zasadą:

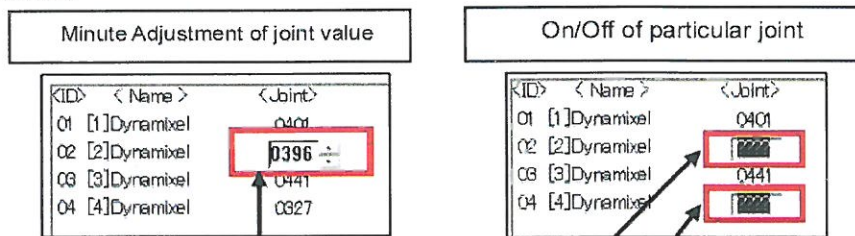
przyspieszenie->utrzymanie prędkości->spowolnienie. Tutaj ustawiamy czas przyspieszenia/spowolnienia. Jeżeli zbyt zmniejszymy dany czas, serwo może ulec uszkodzeniu. Przy zbyt długim czasie - serwo może się nie poruszyć.

8. Next page no. - ustawia, która następna strona ma zostać wykonana. Pomocne, jeżeli wykonywany zestaw ruchów wykracza poza 7 pozycji na jednej stronie. Przy wykonywaniu ruchu "Page play", automatycznie zostanie wykonana aktualna strona oraz strona zaznaczona w tym oknie. Jeżeli ta funkcja nie jest potrzebna ustawić wartość 0.
9. Exit page no. - wyznacza stronę do wykonania, gdy ruch został zatrzymany. Gdy "Behavior control programmer" wywoła polecenie "STOP", wykonana zostaje strona zaznaczona w tym polu i robot zostaje zatrzymany. Jeżeli ta funkcja nie jest potrzebna ustawić wartość 0.
10. Play motion - wykonanie pozycji zachowanych w oknie pozycji z zachowaniem kolejności. Klawisz ten zmienia się w "Stop button", gdy wykonywany jest ruch.
11. Save page - zachowuje ustawione pozycje robota do pamięci robota. Warto zapamiętać: należy zapisywać za każdym razem, gdy chcemy uruchomić robota, gdyż przy niezapisaniu, zostanie wykonana sekwencja, która znajdowała się w robocie przed zmianami. Należy zapisywać także przed przejściem na inne strony, w przeciwnym wypadku wykonane zmiany zostaną utracone.

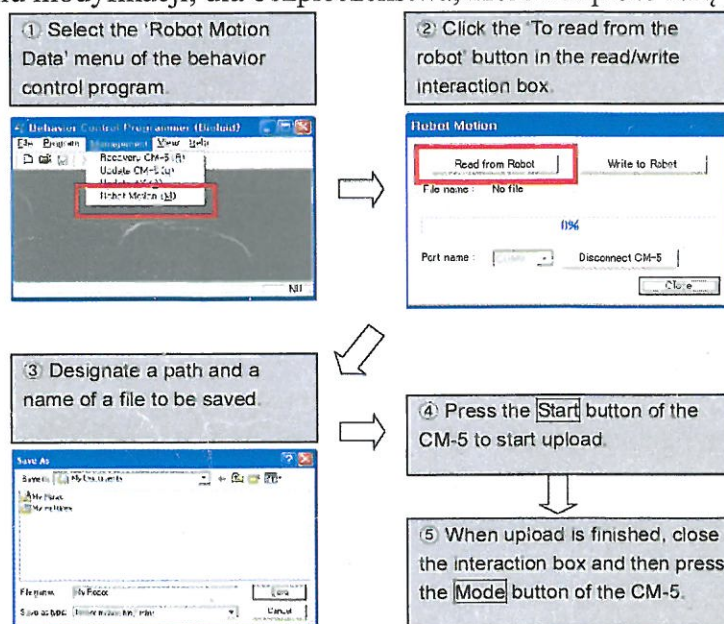
4) Uruchomienie i zarządzanie programem (kolejne pozycje)

Ustawienia serw fizycznie można zmieniać po uprzednim "zwolnieniu" ich - w oknie informacji o serwach przycisk "Off" zwalnia serwa, a przycisk "On" zapamiętuje pozycje ustawione przez użytkownika. Nie trzeba zwalniać wszystkich serw na raz - można zaznaczyć konkretne serwo, naciskając przycisk "Off" i regulować tylko danym serwem.

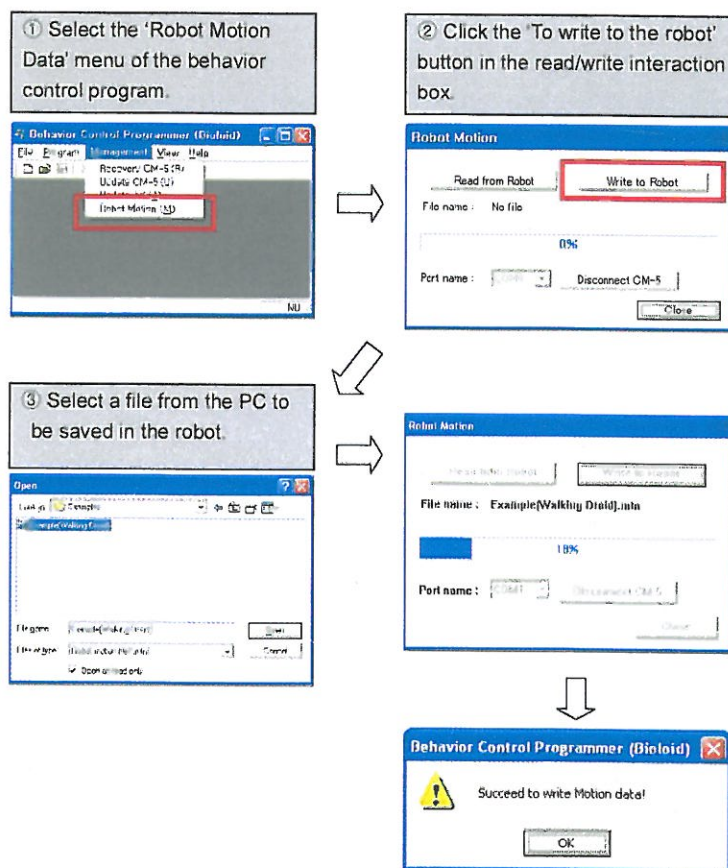
Można zmieniać ustawienia serw poprzez wpisywanie konkretnej wartości w polu <Joint> przy żądanym serwie:



Po zakończeniu modyfikacji, dla bezpieczeństwa, można zapisać całą pamięć robota na PC:



Jednocześnie można zapisać do pamięci robota, dane wcześniej zapisane na PC:



!!! Dzięki zapisowi tych danych na dysku zawsze można wrócić do przerwanej wcześniej pracy, nawet jeżeli ktoś zmienił już pamięć robota !!!

6. Zadania do wykonania.

Zapoznać się z działaniem demo zaproponowanym przez firmę Robotis. Zgodnie z powyższą instrukcją należy za pomocą przycisku "Mode", spowodować, że zacznie migać dioda "Play", a następnie wcisnąć przycisk "Start". Szczegóły działania programu demonstracyjnego pokazano w pliku wideo: DemoExample(Puppy).wmv.

Wykonanie zadania polega na zaprogramowaniu robota, aby wykonał zadany ruch. Do wyboru są następujące zestawy ruchów robota:

a) *prosty ruch* – na ocenę 3:

Korzystając z "Motion editor" na wybranych 3 stronach wykonać:

1. robot podnosi tylną łapę do góry
2. robot staje na głowie
3. robot prezentuje zachowanie dynamiczne np.: przewrót przez plecy i powstanie na 4 łapy

Korzystając z "Behavior control programmer" wywołać sekwencję co najmniej 7 nut o różnych czasach trwania poszczególnych dźwięków

b) *ruch złożony* (po zaliczeniu podpunktu a) – na ocenę 4:

Korzystając z podpunktu a), w "Behavior control programmer" napisać program, który wywołuje pierwsze zachowanie po przyciśnięciu klawisza U, drugie – po przyciśnięciu klawisza L, trzecie – po zbliżeniu przeszkody do dowolnego czujnika odległości (ewentualnie każdego z nich) i czwarte po wykonaniu dwóch klaśnień.

c) *zadanie zaawansowane* (po zaliczeniu podpunktów a i b) – ocena 4,5 i 5:

Aplikacja z punktu b) zostaje uzupełniona: po przyciśnięciu przycisków R i D robot porusza się do przodu według jednego lub dwóch wzorców chodu dla robotów 4-nożnych.

