Programowanie manipulatora na przykładzie robota IRB-1400 i języka RAPID

Janusz Jakubiak
Instytut Informatyki, Automatyki i Robotyki
Politechnika Wrocławska

Informacja o prawach autorskich

Prezentacja ta stanowi materiał pomocniczy do zajęć w Laboratorium Robotyki 010.

Zamieszczone ilustracje pochodzą z materiałów firmy ABB.

Poziomy programowania

Produktu: opis końcowego kształtu; planowanie kolejności operacji systemu robotycznego.

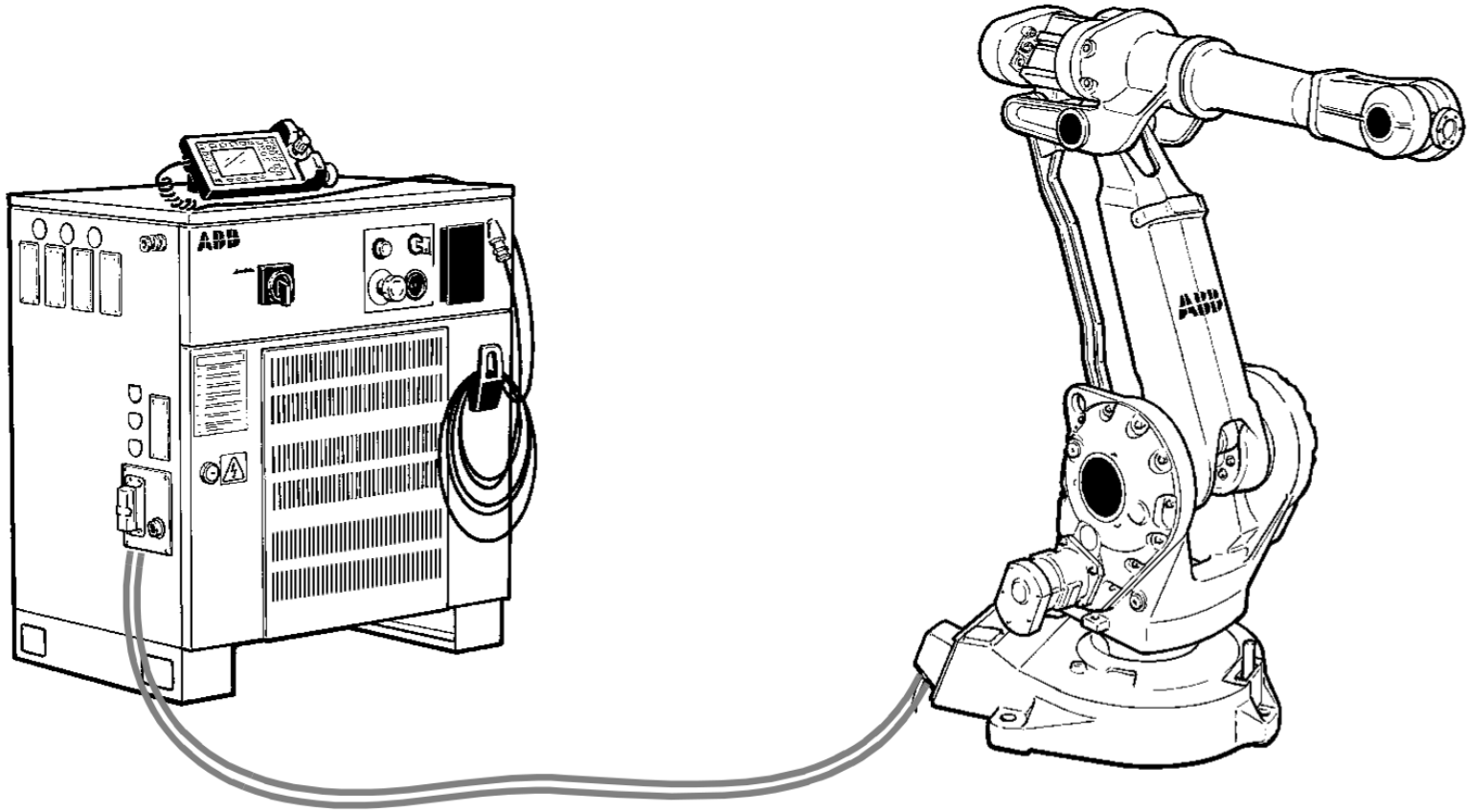
Procesu: dla znanego ciągu operacji wyznaczenie ich parametrów.

Narzędzia: opis ruchu narzędzia dla realizacji konretnej operacji.

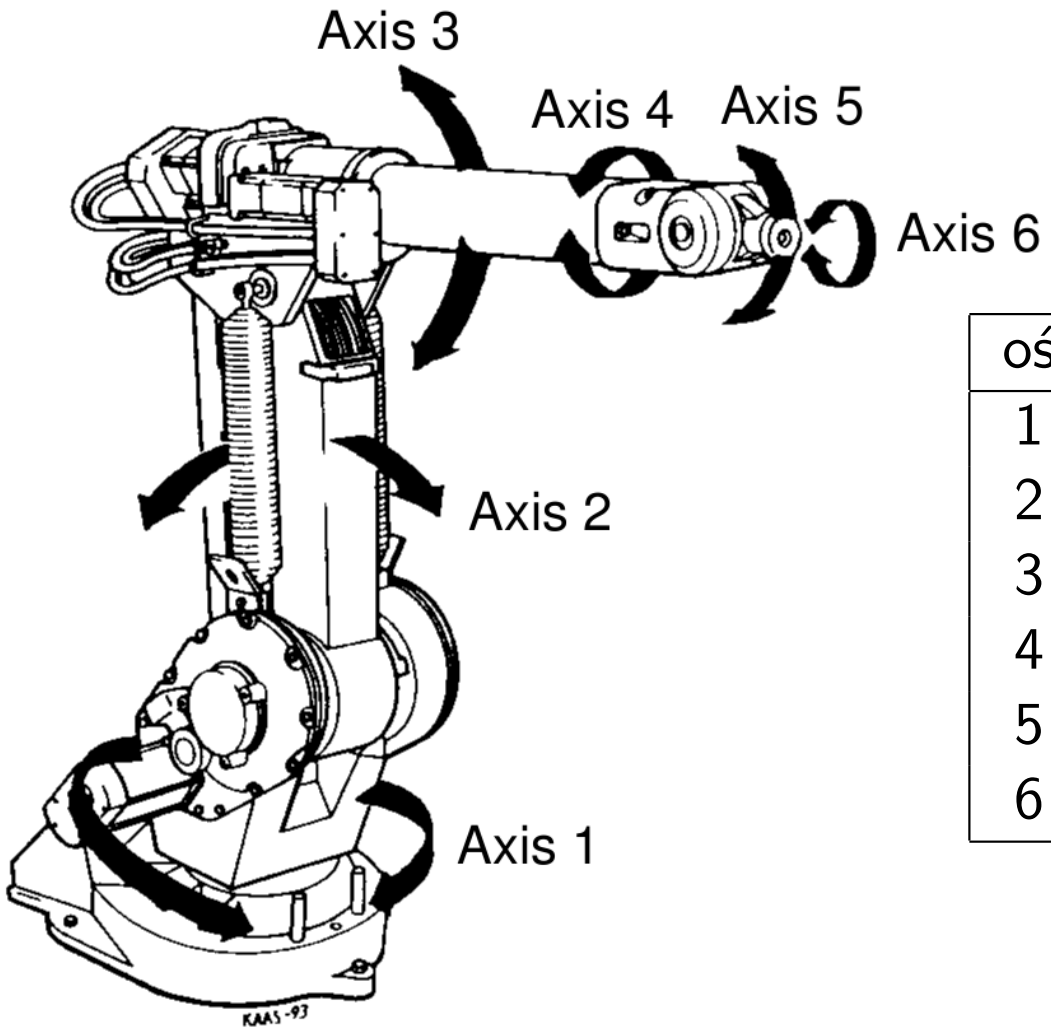
Ramienia: planowanie ruchu ramienia realizującego zadany ruch narzędzia.

Przegubu: planowanie ruchu każdego z przegubów.

Robot IrB 1400

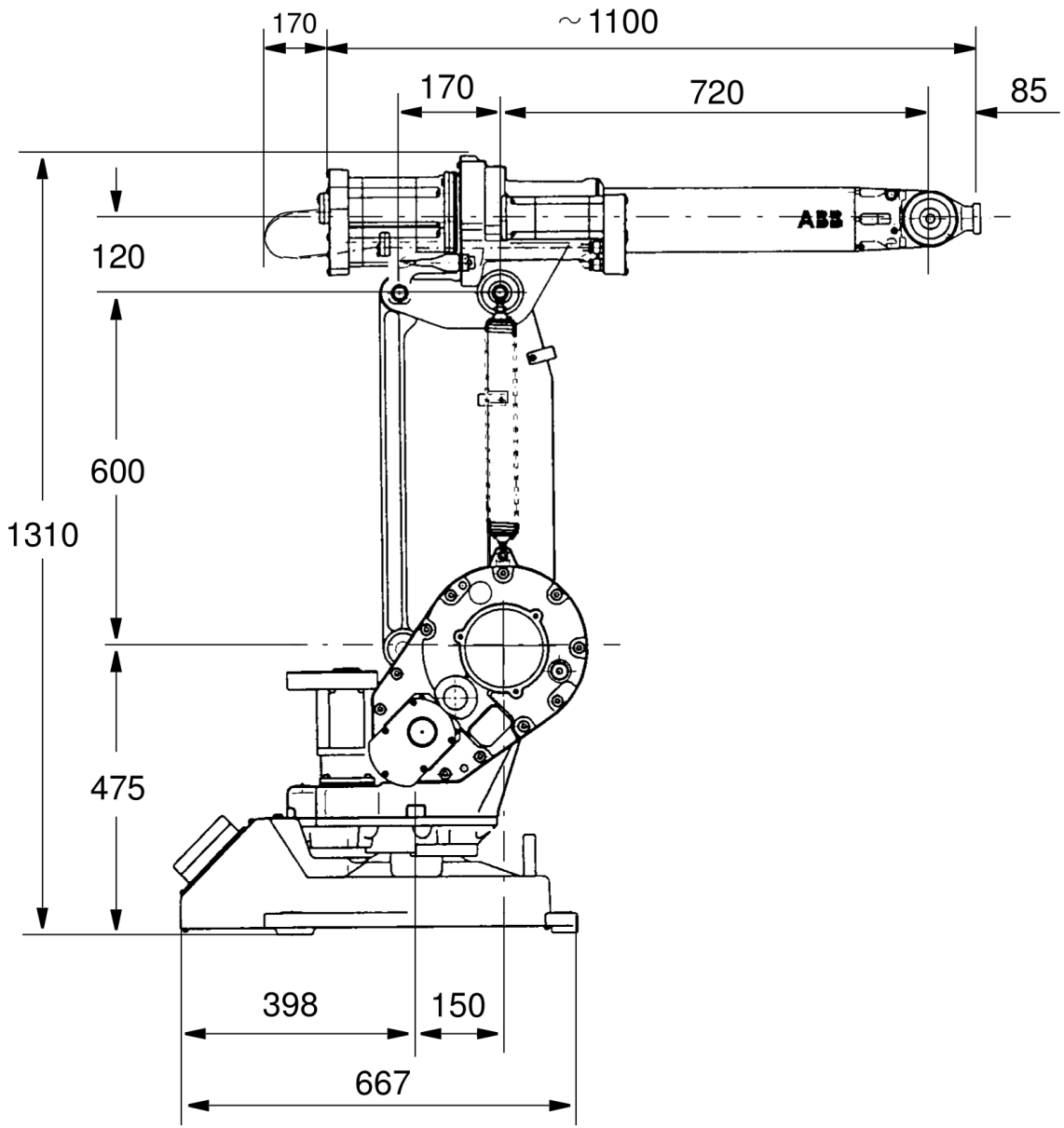


Robot IrB 1400

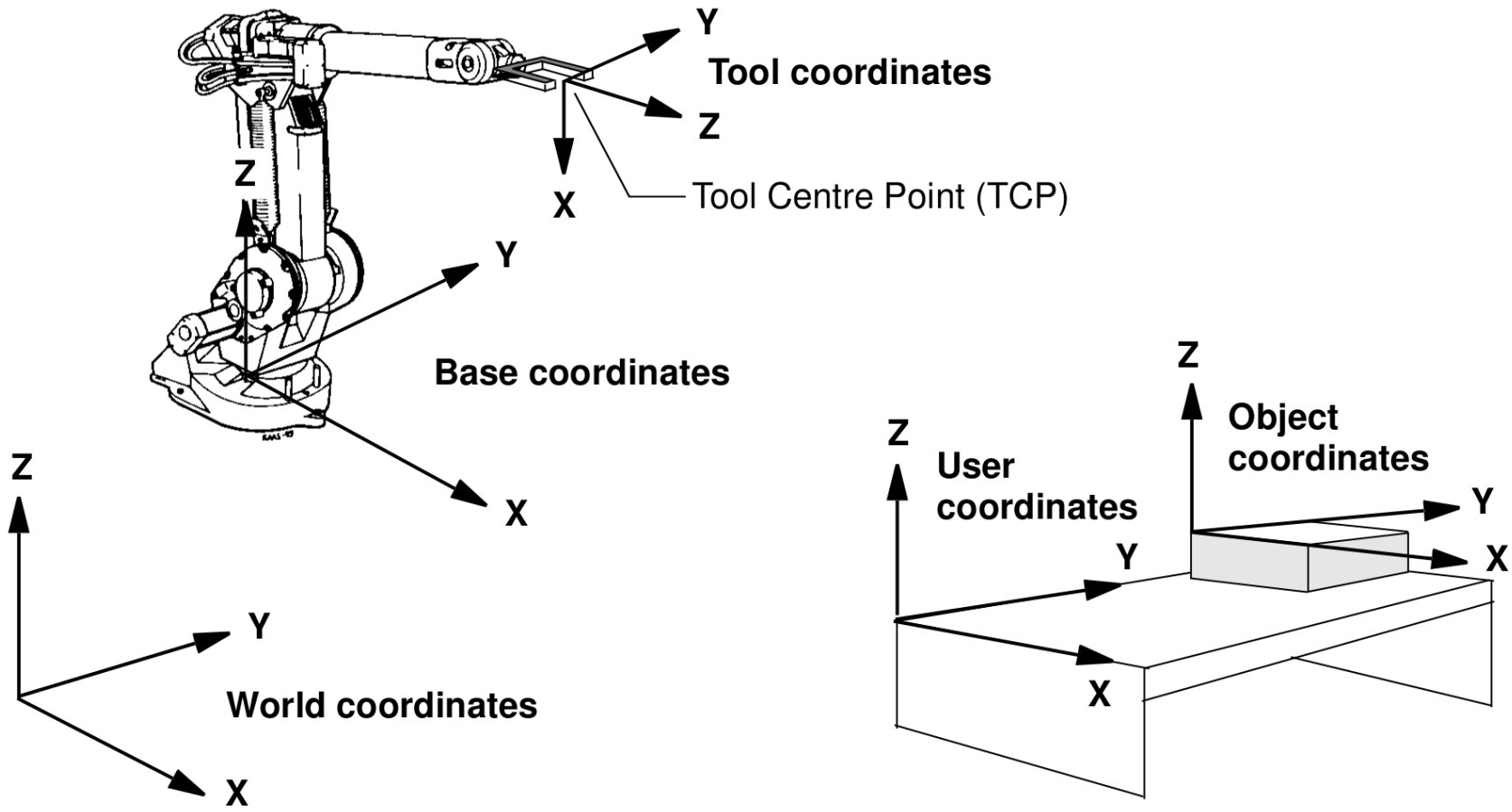


oś	zakres ruchu
1	+170° — -170°
2	+70° — -70°
3	+70° — -65°
4	+150° — -150°
5	+115° — -115°
6	+300° — -300°

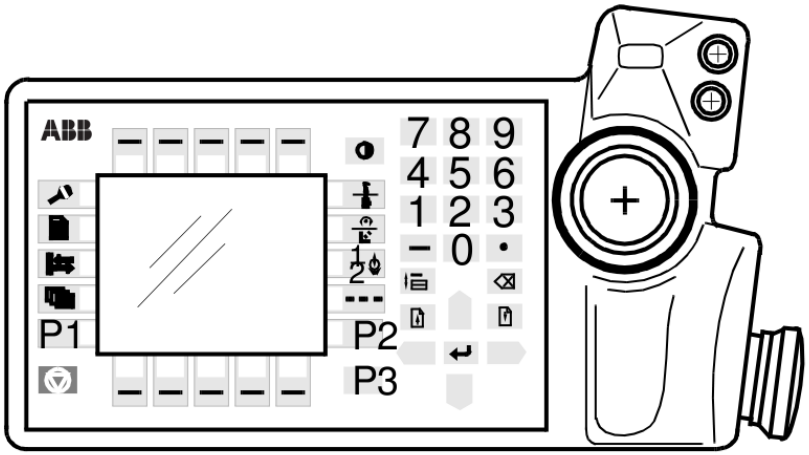
Parametry kinematyki

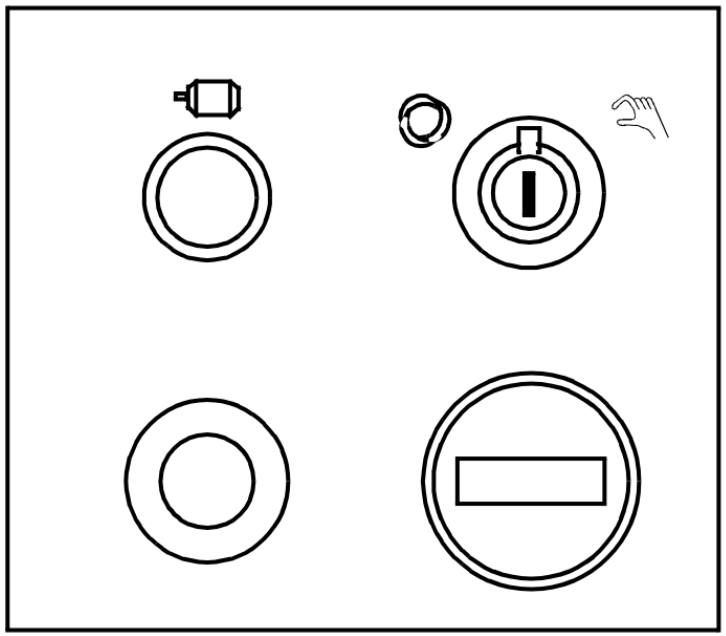


Robot IrB 1400

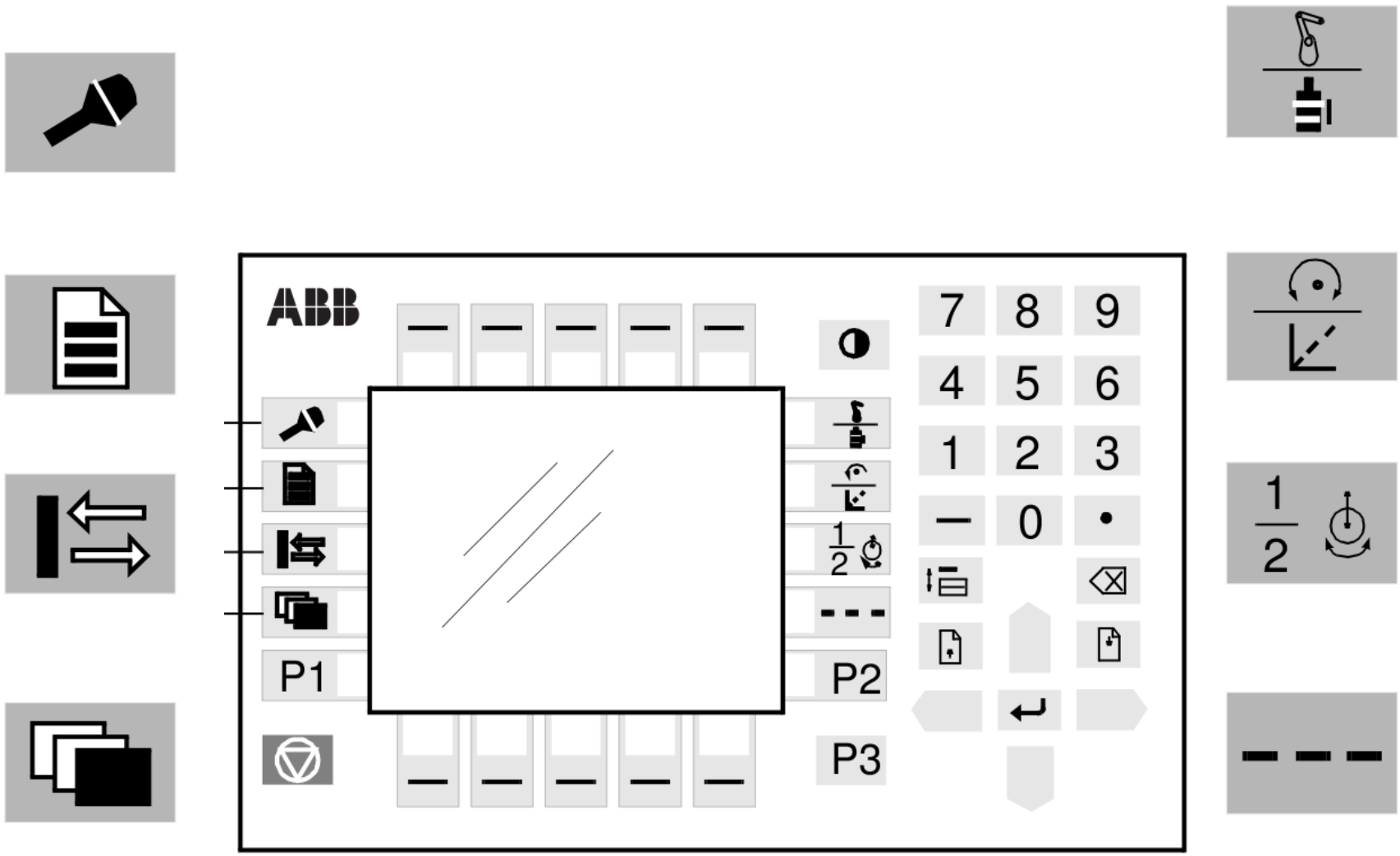


Robot IrB 1400

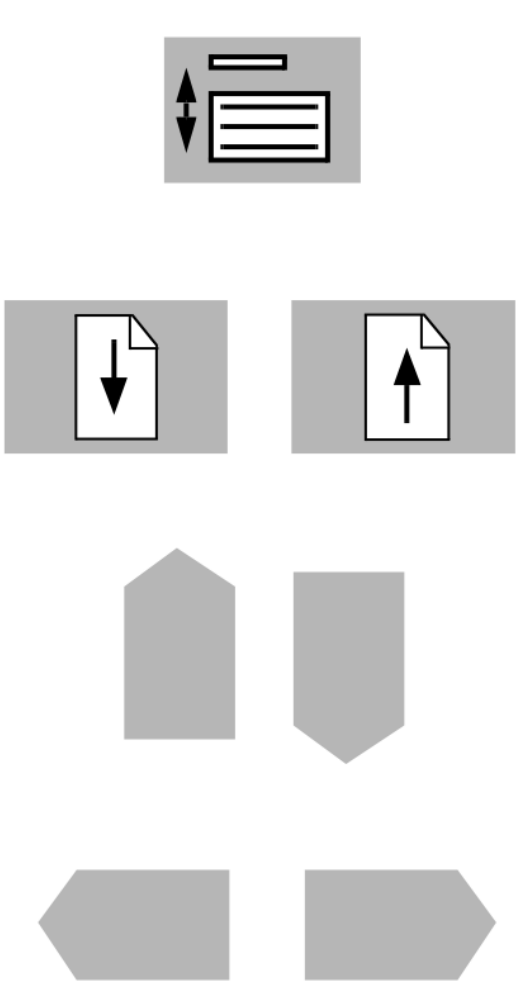




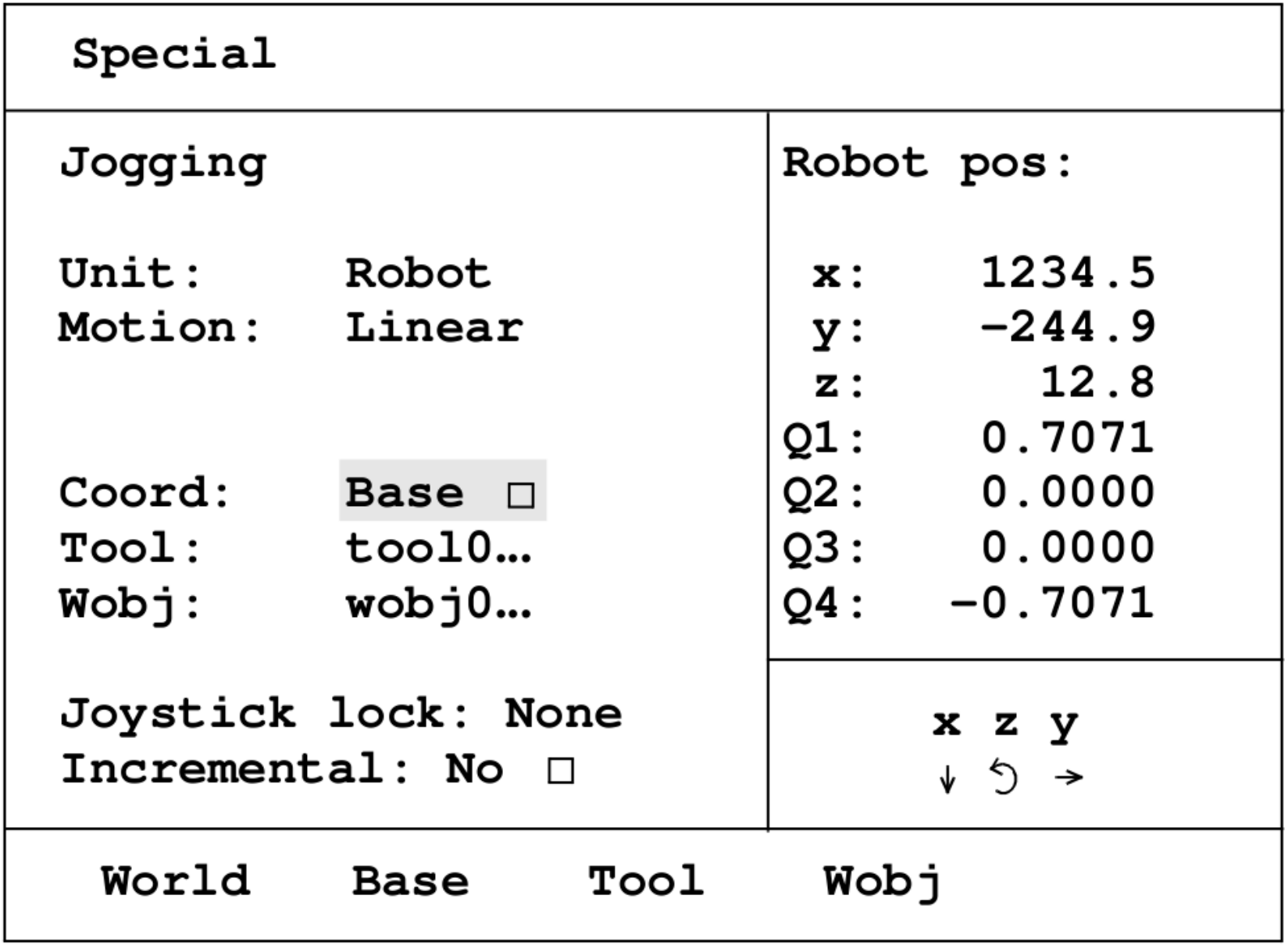
Elementy panelu 1



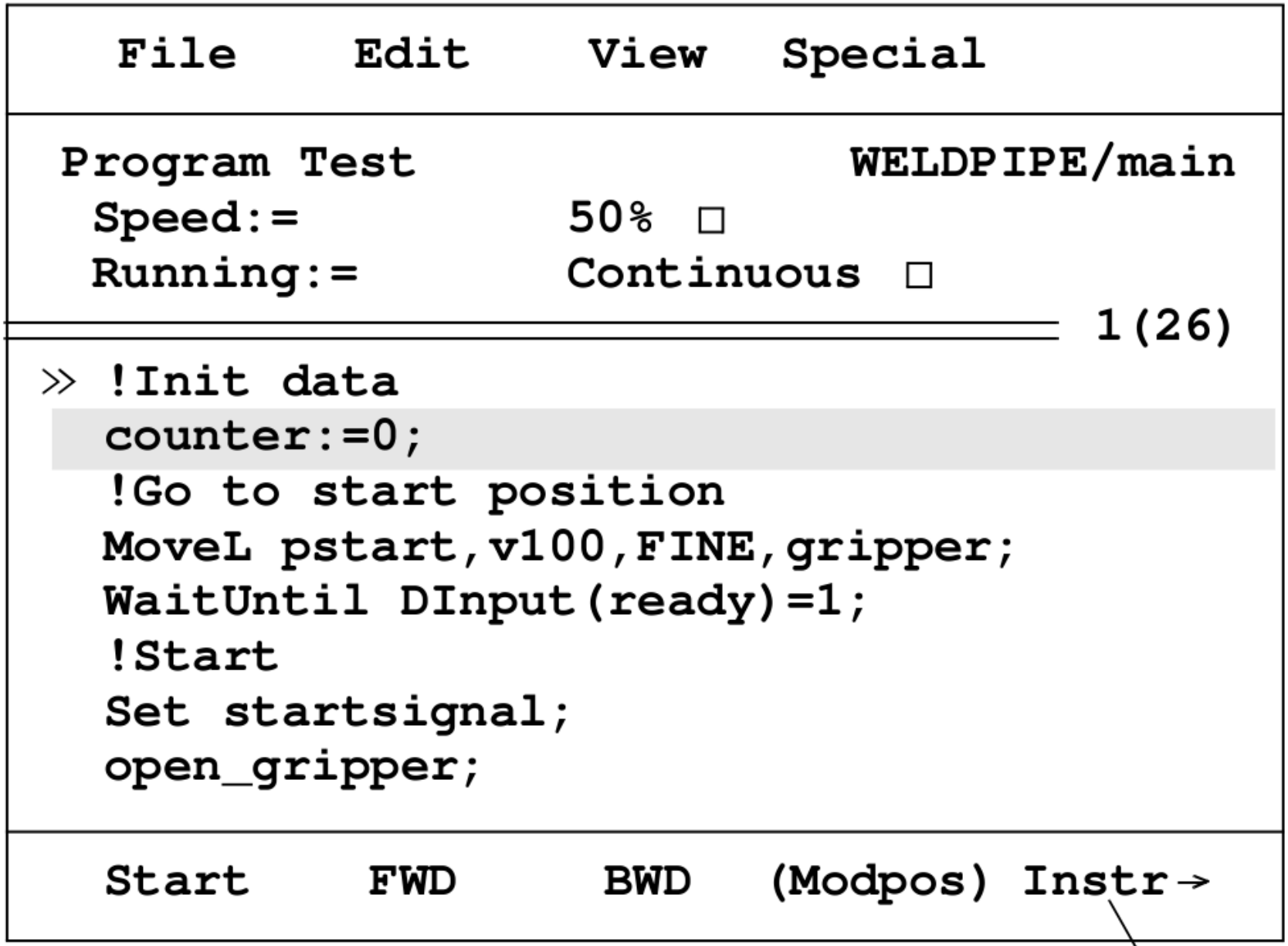
Elementy panelu 2



Ekran panelu w trybie pracy ręcznej



Ekran panelu w trybie wykonywania programu



Elementy języka RAPID

Rodzaje podprogramów (routines)

- procedury (procedure)
- funkcje (function)
- pułapki (trap routine)

Argumenty instrukcji

- numeryczne – całkowite, lub zmiennoprzecinkowe poj. precyzji, np. $1, 2.5, -2.4E-2$;
- logiczne, *TRUE*, *FALSE*;
- zmienne, np. $p10, reg1$;
- wyrażenia, np. $5 + reg1 * 2$;
- wywołania funkcji, np. $Offs(p10, 10, 0, 0)$;
- napisy (do 80 znaków), np. $''xyz''$;

Elementy języka RAPID

Typy danych

- atomowe
  - numeryczne (num)
  - logiczne (bool)
- rekordy
  - konfiguracja robota (robtarget)
  - przemieszczenie (pose)
  - wektor pozycji (pos)
  - orientacja (orient)
  - narzędzie (tooldata)
  - ładunek (loaddata)
  - obiekt (wobjdata)

Różne (Miscellaneous)

:=	Assigns a value
WaitTime	Waits a given amount of time
WaitUntil	Waits until a condition is met
comment	Inserts comments into the program
OpMode	Reads the current operating mode
RunMode	Reads the current program execution mode
Dim	Gets the size of an array
Present	Tests if an optional parameter is used
Load	Loads a program module during execution
UnLoad	Deletes a program module during execution

Struktury sterujące (Program flow)

ProcCall	Calls a new procedure
CallByVar	Calls a procedure by a variable
RETURN	Finishes execution of a routine
FOR	Repeats a given number of times
GOTO	Goes to (jumps to) a new instruction
Compact IF	If a condition is met, then execute one instruction
IF	If a condition is met, then execute a sequence of instructions
label	Line name (used together with GOTO)
TEST	Depending on the value of an expression
WHILE	Repeats as long as
Stop	Stops execution
EXIT	Stops execution when a restart is not allowed
Break	Stops execution temporarily

Ustawienia ruchu (Motion settings) I

AccSet	Reduces the acceleration
ConfJ	Controls the robot configuration during joint movement
ConfL	Monitors the robot configuration during linear movement
VelSet	Changes the programmed velocity
GripLoad	Defines the payload
SingArea	Defines the interpolation method through singular points
PDispOn	Activates program displacement
PDispSet	Activates program displacement by specifying a value
DefFrame	Defines a program displacement automatically
DefDFrame	Defines a displacement frame

Ustawienia ruchu (Motion settings) II

EOffsOn	Activates an offset for an external axis
EOffsSet	Activates an offset for an external axis using a value
ORobT	Removes a program displacement from a position
SoftAct	Activates soft servo for a robot axis
TuneServo	Tunes the servo

Instrukcje ruchu (Motion) I

MoveC	Moves the TCP circularly
MoveJ	Moves the robot by joint movement
MoveL	Moves the TCP linearly
MoveAbsJ	Moves the robot to an absolute joint position
MoveXDO	Moves the robot and set an output in the end position
SearchC	Searches during circular movement
SearchL	Searches during linear movement
ActUnit	Activates an external mechanical unit
DeactUnit	Deactivates an external mechanical unit
Offs	Displaces a position
RelTool	Displaces a position expressed in the tool coordinate system

Instrukcje ruchu (Motion) II

MirPos	Mirrors a position
CRobT	Reads current robot position (the complete robtarget)
CJointT	Reads the current joint angles
CPos	Reads the current position (pos data)
CTool	Reads the current tool data
CWObj	Reads the current work object data
StopMove	Stops robot motion
StartMove	Restarts robot motion

Obsługa wejść/wyjść (Input and output signals) I

InvertDO	Inverts the value of a digital output signal
PulseDO	Generates a pulse on a digital output signal
Reset	Sets a digital output signal to 0
Set	Sets a digital output signal to 1
SetAO	Sets the value of an analog output signal
SetDO	Sets the value of a digital output signal after a defined time
SetGO	Sets the value of a group of digital output signals
WaitDI	Waits until a digital input is set
WaitDO	Waits until a digital output is set
AInput	Reads the value of an analog input signal
DInput	Reads the value of a digital input signal
DOutput	Reads the value of a digital output signal

Obsługa wejść/wyjść (Input and output signals) II

GInput	Reads the value of a group of digital input signals
GOutput	Reads the value of a group of digital output signals
TestDI	Tests if a digital input signal is set
IODisable	Disables an I/O module
IOEnable	Enables an I/O module

Przerwania (Interrupts)

ISignalDI	Orders interrupts from a digital input signal
ISignalDO	Orders interrupts from a digital output signal
ITimer	Orders a timed interrupt
IDelete	Cancels an interrupt
ISleep	Deactivates an interrupt
IWatch	Activates an interrupt
IDisable	Disables interrupts
IEnable	Enables interrupts
CONNECT	Connects an interrupt to a trap routine

Obsługa błędów (Error Recovery)

EXIT	Terminates program execution
RAISE	Calls an error handler
RETRY	Restarts following an error
TRYNEXT	Skips the instruction that has caused the error
RETURN	Returns to the routine that called the current routine

Komunikacja z użytkownikiem (Communication)

TPErase	Erases text printed on the teach pendant
TPWrite	Writes on the teach pendant
TPReadFK	Reads function keys
TPReadNum	Reads a number from the teach pendant
ErrWrite	Stores an error message in the error log

Czas (System& Time)

ClkReset	Resets a clock used for timing
ClkStart	Starts a clock used for timing
ClkStop	Stops a clock used for timing
ClkRead	Reads a clock used for timing
CDate	Reads the current date as a string
CTime	Reads the current time as a string
GetTime	Gets the current time as a numeric value

Matematyczne (Mathematics) I

Add	Adds a numeric value
Clear	Clears the value
Decr	Decrements by 1
Incr	Increments by 1
Abs	Calculates the absolute value
Sqrt	Calculates the square root
Exp	Calculates the exponential value with the base e
Pow	Calculates the exponential value with an arbitrary base
ACos	Calculates the arc cosine value
ASin	Calculates the arc sine value
ATan/ATan2	Calculates the arc tangent value
Cos	Calculates the cosine value

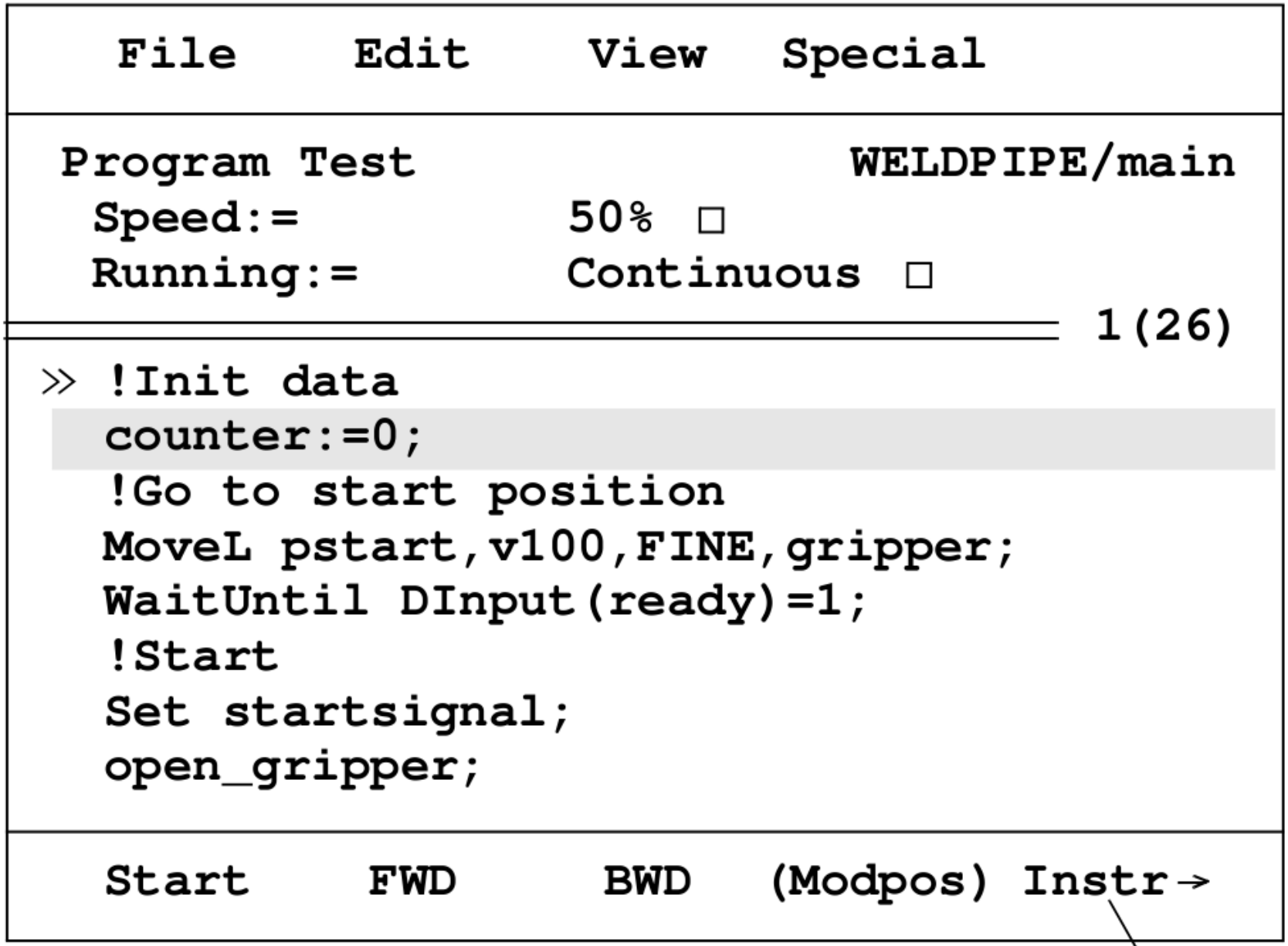
Matematyczne (Mathematics) II

Sin	Calculates the sine value
Tan	Calculates the tangent value
EulerZYX	Calculates Euler angles from an orientation
OrientZYX	Calculates the orientation from Euler angles
PoseInv	Inverts a pose
PoseMult	Multiplies a pose
PoseVect	Multiplies a pose and a vector
Round	Rounds a numeric value
Trunc	Truncates a numeric value

Napisy(Text strings)

NumToStr	Converts numeric value to string
StrFind	Searches for a character in a string
StrLen	Gets the string length
StrMap	Maps a string
StrMatch	Searches for a pattern in a string
StrMemb	Checks if a character is a member of a set
StrOrder	Checks if strings are ordered
StrPart	Gets a part of a string
StrToVal	Converts a string to a numeric value
ValToStr	Converts a value to a string

Ekran panelu w trybie wykonywania programu



Poziomy programowania

Produktu: opis końcowego kształtu; planowanie kolejności operacji systemu robotycznego.

Procesu: dla znanego ciągu operacji wyznaczenie ich parametrów.

Narzędzia: opis ruchu narzędzia dla realizacji konretnej operacji.

Ramienia: planowanie ruchu ramienia realizującego zadany ruch narzędzia.

Przegubu: planowanie ruchu każdego z przegubów.

Przykład programu na poziomie produktu

```
start_task ; // Inicjalizacja procesu
WHILE Dinput(taskstop) = 0 DO //Powtarzaj aż do wciśnięcia przycisku
   fetch_part ; //Pobierz element z podajnika
   insert_part ; //Umieść w maszynie
   process_part ;//Uruchom obróbkę
   remove_part; //Odbierz element
   leave_part ; //Odłóż do podajnika
ENDWHILE
stop_task ; //Zatrzymaj proces
```

Przykład programu na poziomie narzędzia

```
PROC cam_pick()
 MoveJ app_point, v1500, z25, tool0;
    // Wstępne pozycjonowanie przy stole roboczym (z tolerancją 25 mm)
 MoveJ Offs(camera11,x,y,-30), v1500, fine, tool0;
    // Pozycjonowanie 30 mm powyżej pozycji x, y uzyskanej z kamery
 temp1:=CRobT();     // Zapamiętanie aktualnej pozycji
 MoveL Offs(temp1,0,0,-38), v400, fine, tool0;
    // Ruch w kierunku obiektu z uwzględnieniem eleastyczności chwytaka (8 mm)
 Set DO08;                                  // Włączenie chwytaka (pompy próżniowej)
 WaitTime 1.5;                                    // Oczekiwanie 1.5 s
 MoveL Offs(temp1,0,0,10), v400, fine, tool0; // Ruch w górę 10 mm
 MoveJ app_point, v1500, z25, tool0;
    // Ruch do pozycji początkowej z tolerancją 25 mm
 MoveJ place_box, v1500, fine, tool0; // Umieszczenie obiektu w pudełku
 Reset DO08;                                // Wyłączenie chwytaka, zwolnienie obiektu
 Set DO07;                                  // Włączenie dmuchawy
 WaitTime 0.8;                              // Oczekiwanie 0.8 s
 Reset DO07;                                // Wyłączenie dmuchawy
ENDPROC
```