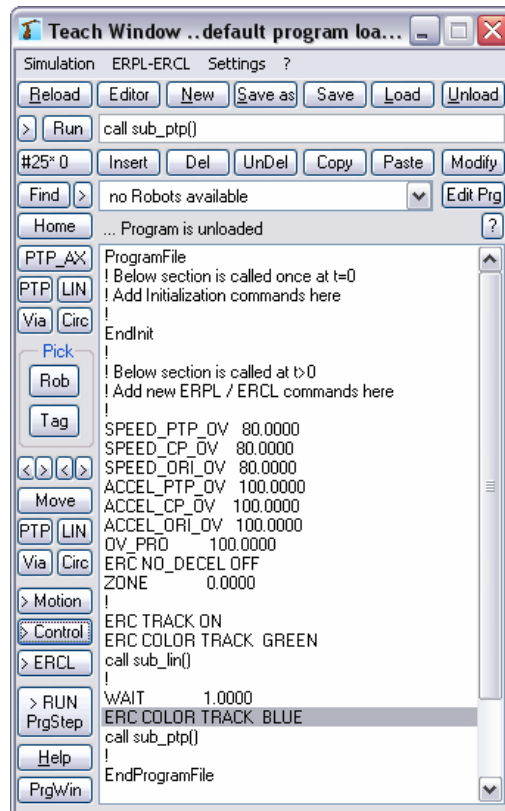


ERPL- / ERCL- Program Language

EASY-ROB™ V5.3



MAY 2010

Version 2.3

EASY-ROB™

Table of contents

Table of contents.....	2
ERPL - EASY-ROB Program Language	3
General Program Structure	3
Robot Motion Commands	4
IF, While and Goto Statements	9
Commands for I/O Signals	10
Math Parser Functions	11
Triangle Parser Functions	12
Robot-related Parser Functions	13
ERCL - EASY-ROB™ Command Language.....	14
ERCL - ON / OFF Commands	14
ERCL - Render Commands	17
ERCL - Camera Commands	17
ERCL - Color Commands	18
ERCL - Transparency- Commands.....	19
ERCL - Reset and Save Commands	19
ERCL - Load Commands	19
ERCL - Move Commands	20
ERCL - Grab and Release Commands.....	21
ERCL – Robot / Device Commands	21
ERCL - TAG Commands.....	22
ERCL - View Commands	22
ERCL - TCP Trace Commands	23
ERCL - Collision Commands	24
ERCL - List Commands (obsolete)	24
ERCL - Attach Commands.....	25
ERCL - Unit Commands.....	25
CALC - Math Commands	25
ERCL - PARAMETER Commands	26
ERCL - KUD Commands	26
ERCL - Additional Commands	27

EASY-ROB™

ERPL - EASY-ROB Program Language

The below table gives an overview about the EASY-ROB program structure and available robot motion commands.

Principle syntax definition:

- Units (length) are meter [m], degree [deg] or percentage [%]
- Speed units are in length unit per seconds e.g. [m/s]
- A cartesian position consists of a position with X, Y and Z values and an orientation with A, B and C angles.
The orientation definition for ABC angles is:
 $\text{Rot}(A,B,C) = \text{Rot}(X,A) * \text{Rot}(Y,B) * \text{Rot}(Z,C)$
- A Tag name is for example 'T_1'. To use a tag name with a motion command, the Tag must exist in the work cell.

General Program Structure

Command and Syntax	Description
PROGRAMFILE	Begin of Program Executing this command as single step command will reset some status data, such as BASE, BASE_PRG, etc.
ENDPROGRAMFILE or END	End of Program Program executing stops
CALL fct_name()	Internal Function Call fct_name() - name of function Note: The function must exist in the current program file
FCT fct_name()	Begin function definition fct_name() - name of function
ENDFCT	end of function
! some text and comments	The '!' sign will comment out the complete command line
EndInit	End of the area for the initialization (for initialization of the „not-time-consuming“ commands)

Robot Motion Commands

Command and Syntax	Description
! some text and comments	The '!' sign will comment out the complete command line
OV_PRO x [%]	Programmable Override x - percentage value
SPEED_CP dx dx_e [m/s]	Speed for continuous path motion dx - cartesian speed [dx_e] – cartesian speed at target
SPEED_ORI dx_ori dx_e_ori [deg/s]	Orientation Speed for continuous path motion dx_ori - orientation speed [dx_e_ori] - orientation speed at target
SPEED_PTP v ve [m/s, deg/s]	Speed for PTP motion v - joint speed [ve] - joint speed at target <i>Remark: command obsolete – use SPEED_PTP_AX instead</i>
ACCEL_CP ax ax_e [m/s ²]	Acceleration for continuous path motion ax - cartesian acceleration [ax_e] - cartesian acceleration at target
ACCEL_ORI ax_ori ax_e_ori [deg/s ²]	Orientation Acceleration for continuous path motion ax_ori - orientation acceleration ax_e_ori - orientation acceleration at target
ACCEL_PTP aq aq_end [m/s ² , deg/s ²]	Acceleration for PTP motion aq - joint acceleration [aq_e] - joint acceleration at target <i>Remark: command obsolete – use ACCEL_PTP_AX instead</i>
SPEED_PTP_AX v1 .. vn [m/s, deg/s]	Speed for PTP motion for every axis v1 - Joint(1) - joint speed [vn] – Joint(n) - joint speed
ACCEL_PTP_AX a1 .. an [m/s ² , deg/s ²]	Acceleration for PTP motion for every axis a1 - Joint(1) acceleration [an] - Joint(n) acceleration
SPEED_PTP_OV x [%]	Percentage speed (for PTP motion) of the maximum speed values (for every axis) x – value in percent
ACCEL_PTP_OV x [%]	Percentage acceleration (for PTP motion) of the maximum acceleration values (for every axis) x - value in percent
SPEED_CP_OV x [%]	Percentage speed (for continuous path motion) of the maximum speed values (for every axis) x - value in percent
ACCEL_CP_OV x [%]	Percentage acceleration (for continuous path motion) of the maximum acceleration values (for every axis) x - value in percent
SPEED_ORI_CP_OV x [%]	Percentage orientation-speed (for continuous path motion) of the maximum speed values (for every axis) x - value in percent
ACCEL_ORI_CP_OV x [%]	Percentage orientation-acceleration (for continuous path motion) of the maximum acceleration values (for every axis) x - value in percent

Command and Syntax	Description
CONFIG n	Robot Configuration n - configuration number
TOOL X Y Z A B C [m,deg]	Tool data from tip to TCP XYZ - Position ABC - Orientation
TOOL tagname	Tool data from tip to TCP tagname - name of Tag
TOOL DEVICE robname	Sets the TCP for the current Robot to the TCP data of the robot 'robname'. This command is useful after another robot is grabbed.
TOOL toolname	Sets the tool / the TCP to the tool with name ,toolname'
TOOL DEVICE tool_dev_name toolname	Sets the TCP for the current Robot to the tool data of the tool ,toolname' of the selected devices 'tool_dev_name'. This command is useful after another robot is grabbed
EXT_TCP X Y Z A B C [m,deg]	Extern TCP XYZ - Position ABC - Orientation
EXT_TCP tagname	Extern TCP tagname - name of Tag
BASE X Y Z A B C [m,deg]	Shift Targets by BASE frame The goal of all BASE commands is to shift program commands. The BASE command is always with respect to the robots base. (see also ERC BASE ...) i.e. All following motion commands are transformed by the current base frame. XYZ - Position ABC - Orientation Note: see also ERC BASE BODY bodyname ERC BASE TCP
BASE tagname	Program BASE tagname - name of Tag
BASE_REL dX dY dZ dA dB dC [m,deg]	Relative Program BASE Shift the current base frame by the relative location dXdYdZ - delta Position, dAdBdC - delta Orientation
BASE_PRG X Y Z A B C [m,deg]	The BASE_PRG command operates with respect to the current BASE frame. The final reference for all motion and position commands with respect to the robots base is calculated as: $T_base_final = T_base_prg * T_base$ (T - homogeneous 4x4 matrix) XYZ – Position, ABC - Orientation
BASE_PRG_REL dX dY dZ dA dB dC [m,deg]	Relative Program BASE_PRG dXdYdZ - delta Position, dAdBdC - delta Orientation

LEADING_POSITION x	Leading Position for continuous path motion ON - Position has priority, keeps speed on a path OFF - Orientation has priority VAR - slowest profile has priority
LEADING_ORIENTATION x	Leading Orientation for continuous path motion (Opposite form LEADING_POSITION) ON - Orientation has priority OFF - Position has priority, keeps speed on a path VAR - slowest profile has priority
HOME n	Home position n - number of home position
HOME homepositionname	Home position homepositionname - name of home position
HOME \$NAME_N	Homeposition \$NAME_N Name of the home position defined by the command ERC SET_PARAMETER \$NAME_1 'name'
PTP X Y Z A B C [m,deg] [Extax1 Extax2 ...]	Synchro PTP XYZ - Position ABC - Orientation Extax – external axis values
PTP_REL dX dY dZ dA dB dC [m,deg]	Relative Synchro PTP dXdYdZ - delta Position dAdBdC - delta Orientation
PTP tagname	Synchro PTP tagname - name of Tag
PTP_AX q1 .. qn [m,deg]	Joint specific Synchro PTP q1..qn - target Joint/Axis
PTP_AX_REL dq1..dqn [m,deg]	Relative joint specific Synchro PTP dq1..dqn - delta Joint/Axis
LIN X Y Z A B C [m,deg]] [Extax1 Extax2 ...]	Linear CP motion XYZ – Position, ABC - Orientation Extax - external axis values
LIN_REL dX dY dZ dA dB dC [m,deg]	Relative Linear CP motion dXdYdZ - delta Position, dAdBdC - delta Orientation
LIN TagName	Linear CP motion Tag name - name of Tag
LIN_ORI ori_type	Orientation Interpolation type for Linear CP motion ori_type - VARIABLE, FIX, TANGENTIAL, AUX, VARIABLE2, QUATERNION

MOVE TagNames[] MOVE TagIdx[]	Moves to one or more Tagpoints in tag motion type. Example: Path with 4 tag points { T1, T2, T3, T4}, T1 has PTP motype T2 has LIN motype T3 has CIRC motype T4 has VIA motype MOVE T1 T4 T3 T2 moves to T1 in ptp, to T3 in circ via T4 and to T2 in lin motion type. Using Tagindex, MOVE 1 4 3 2 causes the same motion.
Along path TagNameStrt TagNameEnd	Move along a path path : Path name to move along TagNameStrt - first Tag to move to TagNameEnd - last Tag to move to TagNameStrt, TagNameEnd are identified by names "T_1" or by Idx Examples: along path01 T_1 T_5, moves from Tag "T_1" to "T_5" along path01 2 6, moves from 2 nd Tag to 6 th Tag in Path "path01" Note: The motion type, speeds, etc. depending on Tag attributes when "ERC USE_TAG_ATTRIBUTES ON" is set.
CIRC X Y Z A B C [X2 Y2 Z2] [m,deg]	Circular CP motion XYZ – Position, ABC - Orientation [X2 Y2 Z2] – Via Point Pose
CIRC_REL dX dY dZ dA dB dC [dX2 dY2 dZ2] [m,deg]	Relative Circular CP motion dXdYdZ - delta Position, dAdBdC - delta Orientation [dX2 dY2 dZ2] - delta Via Point position
CIRC tagname [TagName2]	Circular CP motion tagname - name of target Tag [tagname2] - name of via point tag position
CIRC_ORI ori_type	Orientation Interpolation type for Circular CP motion ori_type - VARIABLE, FIX, TANGENTIAL, AUX, VARIABLE2, QUATERNION
VIA_POS X Y Z A B C [m,deg]	Via Position for Circular CP motion XYZ – Position, ABC - Orientation
VIA_POS_REL dX dY dZ dA dB dC [m,deg]	Relative Via Position for Circular CP motion dXdYdZ - delta Position, dAdBdC - delta Orientation
VIA_POS tagname	Via Position for Circular CP motion tagname - name of via point Tag pose
JUMP_TO X Y Z A B C [m,deg]	Jump to target location XYZ – Position, ABC - Orientation
JUMP_TO tagname	Jump to target location tagname - name of target Tag
JUMP_TO_AX q1 .. qn [m,deg]	Jump to target joint axis q1..qn - target Joint/Axis
WAIT x [sec]	Wait Statement x - time in seconds

ZONE	Rounding parameter. The command „Zone = 0“ forces the robot to move to the exact position
AUTO_ACCEL ON [OFF/AX/POS/ORI]	Automatic calculation of acceleration dependent on programmed speed. AX – Calculation for PTP motions POS – Calculation for CP motions for Position ORI – Calculation for CP motions for Orientation ON – Calculation for PTP, POS and ORI OFF – Calculation deactivated
ACCSET Acc Ramp	Lagging of accelerations. The arguments Acc and Ramp are given in percentage values in the range 20% to 100%. Acc – Acceleration and Deceleration as percentage value of normal values Ramp – Change of Acceleration and Deceleration as percentage value of normal values
PTP_CALC_MODE SHORTEST_ANGLE [TURN, MATH, IN_TRAVEL_RANGE]	Calculation specification for PTP motion. SHORTEST_ANGLE – shortest angle TURN – use TURN parameter MATH – mathematical within [-180°;180°] IN_TRAVEL_RANGE – within valid travel ranges if possible
TURN Turn_Ax1 ... Turn_Axn	TURN-Values for each axis Ax1...Axn, for the following PTP Target

IF, While and Goto Statements

Command and Syntax	Description
IF <i>condition</i> <i>! enter ERPL,ERCL here</i> ELSEIF <i>condition2</i> <i>! enter ERPL,ERCL here</i> ELSEIF <i>condition3</i> <i>! enter ERPL,ERCL here</i> ELSE <i>! enter ERPL,ERCL here</i> ENDIF	<p>IF Statement (checking condition for true and false)</p> <p>The condition is a mathematical expression and will be checked for "True" and "False". The condition is true when greater zero.</p> <p>Example: $xx > yy$, thus the robot will move to Tag 'T_1'</p> <pre>xx=3; yy=1 IF gt(xx,yy) LIN T_1 ELSEIF lt(xx,yy) LIN T_2 ELSEIF eq(xx,yy) LIN T_3 ELSE <i>! enter ERPL,ERCL here</i> ENDIF</pre>
WHILE <i>condition</i> <i>! enter ERPL,ERCL here</i> ENDWHILE	<p>While-loop (checking condition for true and false)</p> <p>The condition is a mathematical expression and will be checked for "True" and "False". The condition is true when greater zero.</p> <p>Example: The robot will move along path 'path01' three times</p> <pre>xx=4; yy=1 WHILE gt(xx,yy) xx = xx-1 ALONG PATH01 T_1 T_5 ENDWHILE</pre>
GOTO LABEL <i>label_name</i> LABEL <i>label_name</i>	<p>GOTO Statement (to jump to another program line)</p> <p>Example:</p> <pre>GOTO LABEL my_label ... <i>! skip all these program lines</i> ... LABEL my_label</pre> <p>Tips:</p> <ul style="list-style-type: none"> - use labels only if necessary - a label can be placed everywhere, but only once in a program file - never use labels to exit an if-else-endif or a while-endwhile statement

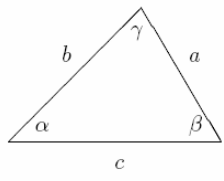
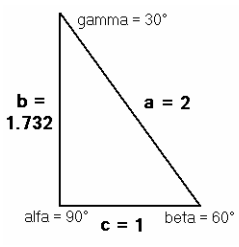
Commands for I/O Signals

Kommando und Syntax	Beschreibung
<p><code>WAIT_UNTIL_SIGNAL_SET</code> <i>my_signal</i></p>	<p>Will wait until the signal is set.</p> <p>The command is checking a condition on „True“ and „False“. The condition is a mathematical expression, which is true if greater than 0.5.</p> <pre> WAIT_UNTIL_SIGNAL_SET my_signal ! continues when the signal „my_signal“ is set ! will wait as long as the signal „my_signal“ is not set </pre>
<p><code>WAIT_UNTIL_SIGNAL_UNSET</code> <i>T my_signal</i></p>	<p>Will wait until the signal is not set anymore.</p> <p>Das Kommando prüft eine Bedingung auf „True“ und „False“. Dabei ist die Bedingung ein mathematischer Ausdruck und ist wahr, wenn größer 0.5</p> <pre> WAIT_UNTIL_SIGNAL_UNSET my_signal ! continues when the signal „my_signal“ is not set anymore ! will wait as long as the signal „my_signal“ is set </pre>
<p><code>WAIT_FOR_CONDITION</code> <i>condition</i></p>	<p>Will check a given condition.</p> <pre> WAIT_FOR_CONDITION gt(my_signal ,0) ! continues if condition is true ! will wait if condition is false </pre>

Math Parser Functions

Command and Syntax	Description	Example
PI	Circle number	3.1415926
RAD	Convert degree to radiant	$RAD = \text{PI} / 180^\circ = 0.017453$
DEG	Convert radiant to degree	$DEG = 180^\circ / \text{PI} = 57.295778$
m2mm	Convert m to mm	$m2mm = 1000$
mm2m	Convert mm to m	$m2mm = 0.001$
abs (x)	Absolute value	$\text{abs}(-6) = 6.000000$
asin (x)*DEG	Inverse Sine	$\text{asin}(0.5)*DEG = 30$
acos (x)*DEG	Inverse Cosine	$\text{acos}(0.5)*DEG = 60$
atan (x)*DEG	Inverse Tangent	$\text{atan}(1)*DEG = 45$
atan2 (x,y)*DEG	Inverse Tangent2	$\text{atan2}(-1,-1)*DEG = -135$
cos (x*RAD)	Cosine	$\text{cos}(45*RAD) = 0.707107$
cosh (x)	Cosine-hyperbola	$\text{cosh}(0) = 1.000000$
ceil (x)	Ceiling	$\text{ceil}(2.83) = 3.000000$
exp (x)	Exponential	$\text{exp}(0) = 1.000000$
eq (x,y)	Equal to	$\text{eq}(5,5) = 1.000000$
fact (x)	Factorial	$\text{fact}(6) = 720.000000$
floor (x)	Floor	$\text{floor}(2.83) = 2.000000$
gt (x,y)	Greater than	$\text{gt}(5,3) = 1.000000$
ge (x,y)	Greater equal	$\text{ge}(5,5) = 1.000000$
int (x)	Integer	$\text{int}(2.83) = 2.000000$
ln (x)	Natural logarithm	$\text{ln}(2) = 0.693147$
Log (x)	Logarithm	$\text{log}(2) = 0.301030$
lt (x,y)	Less than	$\text{lt}(3,5) = 1.000000$
le (x,y)	Less equal	$\text{le}(5,5) = 1.000000$
ne (x,y)	Not equal	$\text{ne}(3,5) = 1.000000$
pow (x,y)	Power	$\text{pow}(2,3) = 8.000000$
rnd (x)	Random	$\text{Rnd}(10)-5 = 3.772546$
Sqrt (x)	Square root	$\text{sqrt}(2) = 1.414214$
sin (x*RAD)	Sine	$\text{sin}(45*RAD) = 0.707107$
sinh (x)	Sine-hyperbola	$\text{sinh}(1) = 1.175201$
sign (x)	Sign	$\text{sign}(-2.83) = -1.000000$
tan (x*RAD)	Tangent	$\text{tan}(45*RAD) = 1.000000$
tanh (x)	Tangent-hyperbola	$\text{tanh}(1) = 0.761594$
trunk (x)	Truncate	$\text{Trunk}(-2.83) = -2.000000$

Triangle Parser Functions

Command and Syntax	Description	Example
$\beta = \text{tr_assa}(\alpha, c, a)$ angle side side > angle	Triangle calculation:  In: α, c, a Out: β	 $\text{tr_assa}(90^\circ\text{RAD}, 1, 2) * \text{DEG} = 60^\circ = \beta$
$\gamma = \text{tr_assa2}(\alpha, c, a)$ angle side side > angle2	In: α, c, a Out: γ	$\text{tr_assa2}(90^\circ\text{RAD}, 1, 2) * \text{DEG} = 30^\circ = \gamma$
$b = \text{tr_asss}(\alpha, c, a)$ angle side side > side	In: α, c, a Out: b	$\text{tr_asss}(90^\circ\text{RAD}, 1, 2) = 1.732051 = b$ $= \sqrt{2^2 - 1^2}$
$b = \text{tr_sass}(c, \beta, a)$ side angle side > side	In: c, β, a Out: b	$\text{tr_sass}(1, 60^\circ\text{RAD}, 2) = 1.732051 = b$ $= \sqrt{2^2 - 1^2}$
$\gamma = \text{tr_sasa}(c, \beta, a)$ side angle side > angle	In: c, β, a Out: γ	$\text{tr_sasa}(1, 60^\circ\text{RAD}, 2) * \text{DEG} = 30^\circ = \gamma$
$\alpha = \text{tr_sssa}(a, b, c)$ side side side > angle	In: a, b, c Out: α	$\text{tr_sssa}(2, \sqrt{3}, 1) * \text{DEG} = 90^\circ = \alpha$
$\gamma = \text{tr_sasssa}(d, \beta, a, b, c)$ side angle side side side > angle	4 - angle calculation In: d, β, a, b, c Out: γ β is between $\langle d, a \rangle$ γ is between $\langle a, b \rangle$	$\text{tr_sasssa}(1, 90^\circ\text{RAD}, 2, 1, 3) * \text{DEG} = 158.6955^\circ = \gamma$
$\delta = \text{tr_sasssa2}(d, \beta, a, b, c)$ side angle side side side > angle	In: d, β, a, b, c Out: δ β is between $\langle d, a \rangle$ δ is between $\langle b, c \rangle$	$\text{tr_sasssa2}(1, 90^\circ\text{RAD}, 2, 1, 3) * \text{DEG} = 33.55731^\circ = \delta$

Robot-related Parser Functions

Command and Syntax	Description	Example
dof(jnt_number)	Joint value of robot joint with 'jnt_number' in radiant or meter	dof(1) * DEG = -0.132455
dofoffsign(jnt_number)	Consider joint offset and sign = dof(jn) * jntsign(jn) - jntoff(jn)	dofoffsign(1) = -0.132455
jntoff(jnt_number)	Joint offset for robot	jntoff(2) = 0.174533
jntsign(jnt_number)	Sign of joint	jntsign(1) = 1.000000
min(1,2)	Minimum of two arguments	min(1,2) = 1.000000
max(1,2)	Maximum of two arguments	max(1,2) = 2.000000
mtounit()	Meter-to-Unit	mtounit() = 1000.000000
robbx()*m2mm	Robot base related to world in x in mm	robbx()*m2mm = 0.000000
robby()*m2mm	Robot base related to world in y in mm	robby()*m2mm = 0.000000
robbz()*m2mm	Robot base related to world in z in mm	robbz()*m2mm = 0.000000
swen(1)	Negative travel range limit [rad,m]	swen(1) = -3.141593
swep(1)	Positive travel range limit [rad,m]	swep(1) = 3.141593
tcpx()*m2mm	TCP (x) related to robot base in mm	tcpx()*m2mm = 1950.001121
tcpy()*m2mm	TCP (y) related to robot base in mm	tcpy()*m2mm = -259.808034
tcpz()*m2mm	TCP (z) related to robot base in mm	tcpz()*m2mm = 800.000131
tcpix()*m2mm	TCP (x) related to world system in mm	tcpix()*m2mm = 1950.001121
tcpiy()*m2mm	TCP (y) related to world system in mm	tcpiy()*m2mm = -259.808034
tcpiz()*m2mm	TCP (z) related to world system in mm	tcpiz()*m2mm = 800.000131
unittom()	Unit-to-Meter	unittom() = 0.001000
collision()	Return 1 if collision detected, else 0	coll = collision()
sim_time()	global simulation time in sec	sim_time = sim_time()

EASY-ROB™

ERCL - EASY-ROB™ Command Language

ERCL is an extension of ERPL, to automate nearly all user interactions inside a robot program. Examples are: Switch ON/OFF the TCP trace, collision, TCP coorsys, load another view, render bodies to flat, wire or invisible, set a new color, change the simulation step size or the motion planner step size, move bodies, define and move body lists, move the robots base, etc.. This feature is useful to create more advanced and effective simulations

Command and Syntax	Description
ERC SET_DEFAULTS	Set default values: Enables the robot, tool and environment bodies. Disables the robot joint coorsys.
ERC SIM_STEP x [sec]	Set Simulation step size x - simulation step size Note: This command switches "Realtime OFF" automatically
ERC CNTRL_STEP x [sec]	Set Controller sample rate x - controller sample rate
ERC SYSTEM_STEP x [sec]	Set robot modell sample rate x - model sample rate
ERC IPO_STEP x [sec]	Set Interpolation sample rate x - ipo sample rate
ERC IPO_LEAD_TIME x [sec]	Set IPO Lead time, the motion will start after this time. x - ipo lead time
ERC IPO_LAG_TIME x [sec]	Set IPO Lag time, at the end of motion, the robot will rest in that pose for this time before moving to the next target pose. x - ipo lag time





ERCL - ON / OFF Commands

Command and Syntax	Description
ERC TRACK ON,OFF	Enables / Disables the TCP trace
ERC DYNAMICS ON,OFF	Enables / Disables Dynamics
ERC STOP_SWE ON,OFF	Enables / Disables monitoring of software end switches
ERC STOP_SPEED ON,OFF	Enables / Disables monitoring of joint/axis speed
ERC STOP_ACCEL ON,OFF	Enables / Disables monitoring of joint/axis acceleration
ERC COLLISION ON,OFF	Enables / Disables monitoring of collision
ERC STOP_COLLISION ON,OFF	Enables / Disables the STOP motion on collision
ERC RED_BLUE_COLLISION ON,OFF	Enables / Disables red blue collision

ERC ROBOTJOINTS ON,OFF or ERC ROBOTPOSITIONS ON,OFF	Enables / Disables the Online Robot IO Output
ERC FLOOR ON,OFF	Enables / Disables the Floor
ERC FLOOR_RENDER ON,OFF	Enables / Disables the flat shaded floor (flat or wire)
ERC EXT_TCP ON,OFF	Enables / Disables external TCP mode
ERC ORTHOGRAFIC ON,OFF	Enables / Disables Orthographic view
ERC DISPLAY_ROBOT ON,OFF	Enables / Disables the visualization of Robot
ERC DISPLAY_ROBOT_COORSY S ON,OFF	Enables / Disables the visualization of yellow/green colored Robot Joint/Axis Coorsys
ERC DISPLAY_TOOL ON,OFF	Enables / Disables the visualization of Tool
ERC DISPLAY_BODYS ON,OFF	Enables / Disables the visualization of Bodies / Environment
ERC TCP_COORSYS ON,OFF	Enables / Disables the visualization of blue colored Tool/TCP Coorsys
ERC IPO_COORSYS ON,OFF	Enables / Disables the visualization of red colored IPO Coorsys
ERC BASE_COORSYS ON/OFF	Enables / Disables the visualization of green colored Base Coorsys
ERC CREATE_TARGET_TAGS ON/OFF	Enables / Disables creating Tag at target location
ERC RESET_ALL_POSITIONS_JOI NTS ON/OFF	Enables / Disables resets all positions and robot joints
ERC NO_DECEL ON/OFF	Enables / Disables the Speed deceleration at target pose NO_DECEL=ON : will set ZONE=0.1 if ZONE<> 0 NO_DECEL=OFF : will set ZONE=0
ERC GRAFIC_UPDATE ON/OFF	Enables / Disables the visualization of the Render Scene during program Execution. This command is useful to hide background command
ERC DISPLAY_TAGS ON/OFF	Enables / Disables the visualization of Tag poses
ERC VIEW_CHOREOGRAPHY ON,OFF	activates / deactivates View Choreography, thus “ERC LOAD View ...” commands are skipped
ERC DISPLAY_ROBOT_COORSY S ON, OFF	activates / deactivates visualization of the robots joint coorsys for all active and passive joints
ERC DISPLAY_ROBOT_NAME ON, OFF	activates / deactivates visualization of the robots name

ERC STATUS_OUTPUT ON/OFF [1-at simstep,2-at target pose] [flnname] [fct# 0- 12]	Enables / Disables the Status Output during program execution. Using this feature allows you to save the complete simulation status in an own defined format. Typical values are joint/axis data and also cartesian TCP location. Parameter: 1 st : 1 - save status data every simstep 2 - save status data at target location 2 nd : filename, e.g. "out.dat" 3 rd : Function number -1 - default output inside EASY-ROB™ for matlab visualization 0 - default output inside EASY-ROB™ 1..12 - user defined output, see example API-DYN status_output_user_1() defined in file <i>dyn_user.cpp</i> Example: in folder ./proj/proj/ Status_Output.cel and Status_Output.prg
ERC INTERPOLATION ON/OFF	Disables continuous interpolation, robot jumps to target location
ERC REALTIME_SIM ON/OFF	Real time Simulation causes program execution in real time. This mode will calculate the simulation step size permanently.
ERC BACKFACES ON/OFF	Enables / Disables visualization of back faces
ERC CAMERA ON/OFF	Enables / Disables the Camera.
ERC USE_TAG_ATTRIBUTES ON/OFF	Enables / Disables the use of tag attributes, such as motion type, speed, acceleration, etc.
ERC MSG_WIN ON/OFF	Enables / Disables the message window
ERC PRG_WIN ON/OFF	Enables / Disables the program window
ERC DISPLAY_CROBOT ON,OFF	Enables / Disables the visualization of the current robot
ERC DISPLAY_CTOOL ON,OFF	Enables / Disables the visualization of the current tool
ERC COLLISION_CROBOT ON/OFF	Enables / Disables the collision check of the current robot
ERC COLLISION_CTOOL ON/OFF	Enables / Disables the collision check of the current tool
ERC COLLISION_CROBOT_REF ON/OFF	Enables / Disables the reference collision check of the current robot
ERC WORLD_COORSYS ON/OFF	Enables / Disables the world coordinate system
ERC ALL_COORSYS ON/OFF	Enables / Disables all coordinate system in the workcell
ERC HISTORY_DEVICE ON/OFF	Enables / Disables the recording of data (for one device) for the History- Diagram
ERC HISTORY_DEVICE ALL_ON / ALL_OFF	Enables / Disables the recording of data (for all devices) for the History- Diagram
ERC HISTORY_OUTPUT ON / ALL_ON [flnname]	Start of History-Diagram recording flnname – file name to store the data
ERC HISTORY_OUTPUT OFF	Stops the History-Diagram recording
ERC CELL_INFO_SHOW ON/OFF	Enables / Disables the Cell Information line

ERCL - Render Commands


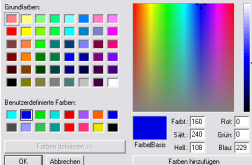
Command and Syntax	Description
ERC RENDER SMOOTH ERC RENDER FLAT	Sets the complete Render Scene FLAT or SMOOTH shaded, same as icon 
ERC RENDER WIRE	Sets the complete Render Scene in WIRE frame, same as icon 
ERC RENDER POINT	Sets the complete Render Scene as POINTS, same as icon 
ERC RENDER BBOX ON,OFF	Sets the complete Render Scene as Bbox (bounded boxes), same as icon 
ERC RENDER group bodyname render	Sets/modifies the render of an unique body group - BODY, ROBOT, TOOL name - name of body render WIRE, FLAT, BBOXWIRE, BBOXFLAT, INVISIBLE, POINT Example: erc render TOOL 'tool' WIRE erc render TOOL 'tool' FLAT
ERC RENDER group render	Sets/modifies the render of all parts in the group group BODY_GRP, ROBOT_GRP, TOOL_GRP render WIRE, FLAT, BBOXWIRE, BBOXFLAT, INVISIBLE, POINT Example: erc render TOOL_GRP WIRE erc render TOOL_GRP FLAT

ERCL - Camera Commands

Command and Syntax	Description
ERC CAMERA ON/OFF	Enables / Disables the Camera.
ERC CAMERA FOCUS focus [mm]	Set Camera focus focus – focus length Range [12.25 mm to 240 mm], default. 50 mm (middle Mouse)
ERC CAMERA Z_OFFSET z_offset [mm]	Set Camera Offset in Z direction z_offset – Z - Offset Range [-200 mm to +500 mm], default. 0 mm (left Mouse) - value zoomes out + value zoomes in
ERC CAMERA BODY [ROBOT,TOOL] bodyname	Attaches the Camera to the origin of CAD geometry
ERC CAMERA POSITION XYZ ABC [m,deg]	Offset position w.r.t. the origin of the CAD geometry where the camera is attached to Note: per default, the camera looks in –Z direction (see also OpenGL™) A 180° rotation about y axis turns the camera in many cases into the right direction.

ERCL - Color Commands

Predefined COLOR values

	1	BLUE
	2	GREEN
	3	CYAN
	4	RED
	5	MAGENTA
	6	BROWN
	7	LIGHTGRAY
	8	DARKGRAY
	9	LIGHTBLUE
	10	LIGHTGREEN
	11	LIGHTCYAN
	12	LIGHTRED
	13	LIGHTMAGENTA
	14	YELLOW
	15	WHITE
	32 Bit RGB Color	

Command and Syntax	Description
ERC COLOR group name color	Sets/modifies the color for an unique body group - BODY, ROBOT, TOOL name - name of body color predefined color Example: erc color TOOL 'tool' RED
ERC COLOR group color	Sets/modifies the color of all parts in the group group BODY_GRP, ROBOT_GRP, TOOL_GRP color predefined color Example: erc color TOOL_GRP RED
ERC COLOR track color	Sets/modifies the color the robots TCP trace track TRACK, TRACK_DYN color predefined color Example: erc color TRACK RED erc color TRACK -1, for alternating color
ERC COLOR tag color	Sets the predefined TAG color color predefined color Note: When creating a new Tag, the tag will have this color

ERCL - Transparency- Commands

Kommando und Syntax	Beschreibung
ERC TRANSPARENCY group name alpha	Set transparency for each single part or body group - BODY, ROBOT, TOOL name - name of body alpha alpha blend value [0,1] Example: erc transparency TOOL 'tool' 0.5
ERC TRANSPARENCY group alpha	Set transparency for each for all parts in a group group BODY_GRP, ROBOT_GRP, TOOL_GRP alpha alpha blend wert [0,1] Example: erc transparency TOOL_GRP 0.5

ERCL - Reset and Save Commands

Command and Syntax	Description
ERC RESET JOINTPOSITION	Reset the joint/axis position to start condition
ERC SAVE JOINTPOSITION	Saves the joint/axis position as start condition

ERCL - Load Commands

This feature gives the user the ability to load a file, e.g. a robot tool or a view file during program execution.

Command and Syntax	Description
ERC LOAD TOOL filename	Loads a Tool file (*.tol)
ERC LOAD VIEW filename	Loads a View file (*.vie) Note: The Render Scene interpolates to the new view with the number of 'view steps' ERC VIEW steps n
ERC LOAD ROBOT filename	Loads a Robot file (*.rob)
ERC LOAD BODY filename	Loads a Body file (*.bod)
ERC LOAD TAGS filename	Loads a Tag file (*.tag)
ERC LOAD MIMIC filename	Loads a Mimic file (*.mmc) ! Mimic = Machine Interface file
ERC LOAD CAMERA filename	Loads a Camera file (*.cam)
ERC LOAD ENVIRONMENT [filename]	Loads an Environment file (*.env)

ERCL - Move Commands

Command and Syntax	Description
ERC MOVE BODY bodyname XYZ ABC [m,deg]	Moves a unique part from the BODY group to a new location bodyname - name of body XYZ - Position. ABC - Orientation
ERC MOVE BODY bodyname TagName	Moves a unique part from the BODY group to a new tag point pose. bodyname - name of body TagName - name of Tag
ERC MOVE TOOL bodyname XYZ ABC [m,deg]	Moves a unique part from the TOOL group to a new location. bodyname - name of body XYZ - Position ABC - Orientation
ERC MOVE TOOL bodyname TagName	Moves a unique part from the TOOL group to a new tag point pose. bodyname - name of body TagName - name of Tag
ERC MOVE ROBOT bodyname XYZ ABC [m,deg]	Moves a unique part from the ROBOT group to a new location. bodyname - name of body XYZ - Position ABC - Orientation
ERC MOVE ROBOT bodyname TagName	Moves a unique part from the ROBOT group to a new tag point pose. bodyname - name of body TagName - name of Tag
ERC MOVE_REL BODY bodyname dXdYdZ dAdBdC [m,deg]	Relative movement of a unique part from the BODY group. bodyname - name of body dXdYdZ - delta Position. dAdBdC - delta Orientation
ERC MOVE_REL TOOL bodyname dXdYdZ dAdBdC [m,deg]	Relative movement of a unique part from the TOOL group. bodyname - name of body dXdYdZ - delta Position. dAdBdC - delta Orientation
ERC MOVE_REL ROBOT bodyname dXdYdZ dAdBdC [m,deg]	Relative movement of a unique part from the ROBOT group. bodyname - name of body dXdYdZ - delta Position. dAdBdC - delta Orientation
ERC MOVE_REL BODY_GRP bodyname dXdYdZ dAdBdC [m,deg]	Relative movement of the complete BODY group, with respect to the reference body. bodyname - name of reference body dXdYdZ - delta Position. dAdBdC - delta Orientation
ERC MOVE_REL TOOL_GRP bodyname dXdYdZ dAdBdC [m,deg]	Relative movement of the complete TOOL group, with respect to the reference body. bodyname - name of reference body dXdYdZ - delta Position. dAdBdC - delta Orientation

ERC MOVE_REL ROBOT_GRP bodyname dXdYdZ dAdBdC [m,deg]	Relative movement of the complete ROBOT group, with respect to the reference body. bodyname - name of reference body dXdYdZ - delta Position. dAdBdC - delta Orientation
ERC MOVE_REL LIST listname dXdYdZ dAdBdC [m,deg]	Relative movement of all parts defined in a LIST listname - name of List dXdYdZ - delta Position. dAdBdC - delta Orientation

ERCL - Grab and Release Commands

Command and Syntax	Description
ERC GRAB BODY 'bodyname'	Grab a body Bodyname - name of body
ERC GRAB BODY_GRP	Grab all parts in the BODY_GRP
ERC RELEASE BODY 'bodyname'	Release a body Bodyname - name of body
ERC RELEASE BODY_GRP	Release all parts in the BODY_GRP
ERC GRAB DEVICE robotname	The current robot grabs the robot/kinematics with name 'robotname'
ERC RELEASE DEVICE robotname	The current robot releases the robot/kinematics with name 'robotname'

ERCL – Robot / Device Commands

Command and Syntax	Description
ERC CURRENT_DEVICE SET 'robotname'	Activates the robots with name 'robotname' Note: All following ERPL- und ERCL-commands refer to the current activated robot. This robot is current 'cRobot'
ERC CURRENT_DEVICE UNSET	Disables the current robot and makes the robot before as current. Example: When writing a function to close a spot weld gun do as follows: We suppose that the cRobot is 'MyRobot' erc current_device set SpotWeldGun ptp_ax 0 0 ! move joint 1 and 2 to 0 erc current_device unset ! disables 'SpotWeldGun', enables 'MyRobot'
ERC CURRENT_DEVICE CLEAR	Clears the complete device stack
ERC CURRENT_DEVICE SHOW	Shows the complete device stack in 'message Window'
TOOL DEVICE robotname	Sets the TCP for the current Robot to the TCP data of the robot 'robotname'. This command is useful after another robot is grabbed.

ERC ROBOT_BASE XYZ ABC [m,deg]	Move the cRobot base to a new location. XYZ - Position ABC - Orientation
ERC ROBOT_BASE tagname	Move the cRobot base to a new Tag location. Tagname - name of TAG
ERC ROBOT_BASE_REL XYZ ABC [m,deg]	Relative movement of the cRobot base. dXdYdZ - delta Position. dAdBdC - delta Orientation

ERCL - TAG Commands

Command and Syntax	Description
ERC CREATE_TARGET_TAGS ON/OFF	Enables / Disables creating Tag at target location.
ERC TAGS PREFIX prefixname	Sets the Prefix for Tags Prefixname - name of prefix Note: When creating a new Tag, the tag will have this prefixname
ERC TAGS DELETE tagname	Delete a Tag pose from the render scene Tagname - name of Tag
ERC TAGS DELETE ALL	Delete all tags from the render Scene

ERCL - View Commands

Command and Syntax	Description
ERC VIEW steps n	Sets the number of view steps, important when loading a (*.vie) file
ERC VIEW hither x	Sets value for the hither plane
ERC VIEW yonder x	Sets value for the yonder Plane
ERC VIEW zoom x	Zoom the render scene by the value x
ERC VIEW zoom_in x	Zoom In the render scene by the value x
ERC VIEW zoom_out x	Zoom Out the render scene by the value x
ERC VIEW tcp_rot_tcp ABC	Rotate the render scene about the current robots TCP with respect to the TCP orientation ABC - relative Rotations
ERC VIEW tcp_rot_world ABC	Rotate the render scene about the current robots TCP with respect to the world frame orientation ABC - relative Rotations
ERC VIEW world_rot_base ABC	Rotate the render scene about the world coorsys frame with respect to the robot base frame orientation ABC - relative Rotations

ERC VIEW world_rot_world ABC	Rotate the render scene about the world coorsys frame with respect to the world frame orientation ABC - relative Rotations
ERC LOAD VIEW filename	Loads a View file (*.vie) Note: The Render Scene interpolates to the new view with the number of 'view steps'

ERCL - TCP Trace Commands

Command and Syntax	Description
ERC TRACK_TYPE type [size]	Sets the trace type and style type POINT LINE LINE_Z_DIRECTION, Z_DIRECTION X_DIRECTION Y_DIRECTION [size] length of direction Examples: erc track_type line erc track_type line_z_direction 0.2 erc track_type line_z_direction -0.2
ERC TRACK ON,OFF	Enables / Disables the TCP trace for current robot 'cRobot'
ERC COLOR track color	Sets/modifies the color the robotsTCP trace track TRACK, TRACK_DYN color predef. color Example: erc color TRACK RED erc color TRACK -1, for alternating standard colors [1..15]

ERCL - Collision Commands

Command and Syntax	Description
ERC Collision ON/OFF	Enables / disables the collision detection Note: In case collision is enabled, all geometries belonging to one robot/tool are checked vs. other robots and tools automatically. This is the default collision queue.
ERC COLLISION BODY [ROBOT, TOOL] bodyname OFF, ON=CONCAVE, CONVEX, BBOX	Enables / disables the collision attribute of a Body to OFF, ! Collision for the geometry is disabled ON = CONCAVE ! Collision is checked concave CONVEX ! Collision is checked with convex hiel BBOX ! Collision is checked with bounded box hiel
ERC COLLISION BODY_GRP [ROBOT_GRP,TOOL_GRP] OFF, ON= CONCAVE, CONVEX, BBOX	Enables / disables the collision attribute for a complete Group to off, on=concave, convex or bbox Note: Robot_Grp and Tool_Grp have effect to the geometries belonging to the current selected robot 'cRobot'
ERC COLLISION QUEUE BODY_ROBOT [BODY_TOOL, ROBOT_TOOL, GRABBODY_ROBOT, GRABBODY_BODY, ROBOT_ROBOT BODY_BODY ALL] ON,OFF	Enables / disables a predefined collision queues BODY_ROBOT – checks collision between body vs. robot BODY_TOOL – checks collision between body vs. tool ROBOT_TOOL – checks collision between robot vs. tool GRABBODY_ROBOT – checks collision between grabbed bodies' vs. robot GRABBODY_BODY – checks collision between grabbed bodies' vs. and not grabbed bodies ROBOT_ROBOT – checks collision between all geometries belonging to the same Robot_Grp BODY_BODY – checks collision between all geometries belonging to the same Body_Grp ALL – enable/disable all predefined queues
ERC RED_BLUE_COLLISION ON, OFF	Enables / disables red blue collision In case collision is checked and detected, all colliding geometries are displayed in red color, non colliding geometries displayed in light blue color.
ERC COLLISION DISTANCE x [mm]	Sets the collision threshold 'x' in mm. Collision is detected if the minimal distance between two tested geometries is less then this collision threshold value 'x'. Note: The minimum collision can be only calculated in convex- or bounded box mode. In case two geometries are checked in concave mode, collision is only detected if the geometries are really colliding, e.g. the distance between them is zero.

ERCL - List Commands (obsolete)

ERC LIST NEW listname [bodyname]	Define a new list listname – Name of a list
ERC LIST DELETE listname	Delete an existing list listname – Name of list to delete
ERC LIST ADD listname bodyname	Add a new body to the list listname – Name of a list bodyname - -Name of body to add
ERC LIST REMOVE listname bodyname	Remove an existing body from a list listname – Name of a list bodyname - -Name of body to remove

ERCL - Attach Commands

ERC ATTACH ROBOT 'bodyname' to_ROBOT ['bodyname2']	Reattaches a body (geometry) to another robot joint inside a Robot_Grp, defined by 'bodyname2' Note: the body is attached to the robot base if no 2 nd body is specified
ERC ATTACH ROBOT 'bodyname' to_TOOL	Reattaches a body to the tip of the cRobot

ERCL - Unit Commands

ERC UNIT M	Sets the unit to meter [m] for all following ERPL & ERCL commands Example: speed_cp 0.1 is 100 mm/s
ERC UNIT MM	Sets the unit to millimeter [mm] for all following ERPL & ERCL commands Example: speed_cp 100 is 100 mm/s
ERC UNIT INCH	Sets the unit to inch [inch] for all following ERPL & ERCL commands Example: speed_cp 1 is 25.4 mm/s
ERC UNIT DEG	Sets the rotation unit to degree [°] Example: speed_ori 20 is 20°s
ERC UNIT RAD	Sets the rotation unit to radiant [rad] Example: speed_ori 3.1415 is 180°s
ERC UNIT TRANS_SCAL x	Scales the internal unit [m] by the user defined value x Example: x=10 speed_cp 0.1 is 1 m/s = 1000 mm/s
ERC UNIT ROT_SCAL x	Scales the internal unit [°] by the user defined value x Example: x= 10 speed_ori 2 is 20°s

CALC - Math Commands

CALC 'math expression'	Example: calc a=0.5 calc b = a * sin (45.0*RAD) LIN_REL 0 0 0.1*a 0 0 0
CALC SHOW_VARS	Shows all global variables in the message window
CALC SHOW_USER_VARS	Shows all user defined global variables (without prefix '\$') in the message window

ERCL - PARAMETER Commands

ERC SET_PARAMETER \$NAME_1 'name'	Copy name into the first of 100 possible global string parameter \$NAME_1 = 'name' Usage: ERC SET_PARAMETER \$NAME_1 T_1 ! Tag point ERC SET_PARAMETER \$NAME_2 MYROB ! a device name ... ERC CURRENT DEVICE SET \$NAME_2! set 'MYROB' as current LIN \$NAME_1 ! move to tag 'T_1'
ERC SET_PARAMETER \$NAME_1 text_word [value]	Copy the text word plus a value name into the first of 100 possible global string parameter \$NAME_1 = text_word [value] Usage: ii=1 ERC SET_PARAMETER \$NAME T_ ii ! Tag point "T_1"
ERC SET_PARAMETER \$NAME_RESET	Deletes all global string parameter
ERC SET_PARAMETER \$NAME_INFO	shows all global string parameter in the message window

ERCL - KUD Commands

ERC KUD row column x	Sets value 'x' for the Kinematics User Data for current robot given by row [1..12] column [1..12] Example: ERC KUD 1 1 1 ERC KUD 12 12 144
ERC KUD name x	Sets value 'x' for the Kinematics User Data for current robot given by a name. Valid names are: KUD_1_1 KUD_12_12 Note: The user can change the KUD name. Example: ERC KUD_1_1 1 ERC KUD_12_12 144

ERCL - Additional Commands

Command and Syntax	Description
ERC STOP	Stops program execution
ERC PAUSE	Halts program execution
ERC ESSI ON,OFF [speed scale value] [size scale value]	Enables / Disables the Interpretation of ESSI NC-Code (Option "NC-Simulation" is required) Optional parameter Speed scale value will scale the programmed speed. Size scale value will scale the target location (program scale) Note: This feature requires a licensed NC-Option
ERC EIA ON,OFF [speed scale value] [size scale value]	Enables / Disables the Interpretation of EIA NC-Code, DIN 66025 (Option "NC-Simulation" is required) Optional parameter Speed scale value will scale the programmed speed. Size scale value will scale the target location (program scale) Note: This feature requires a licensed NC-Option
ERC BASE BODY bodyname	Sets the BASE to the bodyname
ERC BASE TCP	Sets the BASE to the current robots TCP
ERC SWE_NEG swe1 ... swen negative software endswitches [m,deg]	Sets the negative travel range values. The number of values should coincide with the number of joints/axis
ERC SWE_POS swe1 ... swen positive software endswitches [m,deg]	Sets the positive travel range values. The number of values should coincide with the number of joints/axis
ERC TURN_INTERVAL Ax1 ...Axn [deg]	TURN -Interval for each axis Ax1...Axn, within [0;36 0]
ERC JOINT_WEIGHT 0 or 1 for number of joints	Sets the joint_weight vector, to influence the result of the numerical solution The number of values should coincide with the number of joints/axis
ERC MASK_VECTOR [0,1] for X Y Z A B C	Sets the mask vector, to influence the result of the numerical solution
ERC CELL_INFO text	String for cell Information line