

Releasenumber:

Release: I.A. Elter

Date: 24.01.03

Documentation for level:

from level 3 – programmer

Software revision:

10.0

Control version:

RSV

Date:

18.06.2002

Author:

Michael Feiffer

Title: Language change-over

Operation

REIS GMBH & CO MASCINENFABRIK OBERNBURG

## DOCUMENTATION

1	TABLE OF CONTENTS	2
2	GENERAL DESCRIPTION	3
3	TARGET GROUP	3
4	SAFETY INSTRUCTIONS	4
5	CONDITIONS: HARDWARE AND SOFTWARE	4
6	INSTALLATION/CONFIGURATION	4
7	PROGRAMMING	5
7.1	STRUCTURE OF THE TRANSLATION PRESCRIPTION	5
7.2	CHANGE OF THE LANGUAGE Dependent Entries	7
7.3	TRANSLATION PROCEDURE	10
8	APPENDIX	13

# 1 TABLE OF CONTENTS

From level 3 - programmer

## 3 Target group

The automatic translation function confines itself to translation of the command syntax and of the macro names. All additional texts have to be translated manually.

- Texts for system macros
- Message texts
- Comments in programs resp. macros
- Variable names
- Macro parameters
- Macro names
- Program names
- Programming orders

Texts or character strings can be:

For comfortable programming and control of the RSV, a country specific language setting is essential. From Version 8.1 on, it is possible to configure the command syntax in any language by means of a prescription for translation in the manner of a robot program. The only condition is that the signs resp. their input via keyboard can be represented on the PG (teach pendant).

## 2 General Description

The translation function always turns active if a change of the language with `_LANGUAGE` is done and the program "S:/\$CONFIG/\$CNF/LEX.MPR" with the translation prescription is available.

## Installation/Configuration

### 6

RSV Version 8.1 or higher / RobOffice Version 8.1 or higher. As of software version 10.0 the directory structure is modified for the files depending on language. Handling of language change-over was facilitated.

## Conditions: Hardware and Software

The program sequence can be impaired by input mistakes in the configuration program "LEX". After the translation procedure, it must be ensured that the commands and macros work in the same manner as before.

## 4 Safety instructions

command descriptors  
 <Descriptor> For identification of the section with the  
 robot commands  
 <Command> For identification of the section with the  
 following key words have been used:  
 command for the translation to be made. Until now, the  
 contains specific key words which are interpreted as a  
 name "LEX" in the path "\$:/CONFIG/\$CNF". The program  
 a given structure. It is a main program and is filed under the  
 In principle, the translation prescription is a robot program with

## 7.1 STRUCTURE OF THE TRANSLATION

RobOffice/VRC is necessary.  
 language setting, a restart of the control resp. of  
 "\$:/CONFIG/\$MAC/AUTOEXEC". After the change of the  
 \_LANGUAGE is usually initialized in  
 for "foreign language".

### \_LANGUAGE value 1

for "German" or

### \_LANGUAGE value 0

adjustment of the language:  
 Furthermore you need the following system variable for  
 "\$:/CONFIG/\$CNF" to be taken into consideration.  
 "LEX" and has to be filed under the directory  
 The prescription of the translation is included in the program  
 prescription and change of the system variable \_LANGUAGE.  
 For a change of the language, there is required a translation

# 7 Programming

---

```

I EXKL_ODER;EXCL_OR
I UND;AND
I ODER;OR
I ANA_BING;ANA_INP
I ANA_AUSG;ANA_OUTP
I <Command>
MPR "LEX"
Example:

```

Keywords are defined in the 1 - step. The translation instructions are also contained in an I - step. The description in German and special characters such as "<Blank>" must occur in the description in the foreign language are separated by ", ". No translation instruction.

END

I (Macros in German);(Macros in foreign language)

I <Macro>

I (Constants in German);(Constants in foreign language)

I <Equation>

I (Descriptor in German);(Descriptor in foreign language)

I <Descriptor>

I (Command word in German);(Command word in foreign language)

I <Command>

MPR "LEX"

Fundamental structure:

<Equation> For identification of the section with the system constants as a command parameter	macro names For identification of the section with the command word in foreign language
--	--

---

Example:

The changes have to be exclusively made with the non-German text in the following way:

Change of the language for the robot programming language as "LEX" and through modification of the system variable "LEX" and for the macro names is exclusively done in the program well as for the macro names is exclusively made with the non-German language.

## 7.2 CHANGE OF THE LANGUAGE DEPENDENT ENTRIES

A complete "LEX" program in the German - English version is contained in the appendix.  
The length of the individual terms must not be longer than 20 characters. Ambiguity has to be avoided. It is to be ensured that the allocation of the German terms to the foreign language terms is always clear.  
Terms which are not included in the prescription of the translation will not be noticed. In this case, the original term is preserved.

```

END
...
I NOCHEFINMAKRO;NEXT_MACRO
I TESTMAKRO;MACRO TEST
I <Macro>
...
I #BIN;#BIN
I #SCHLIESSEN;#CLOSE
I #MENT_KONST;#MENT_CONST
I #MENT_AKTU;#MENT_ACTU
I #AKTIV;#ACTIVE
I #PASSIV;#PASSIVE
I <Equate>
...
I Queue;Source
I Zeit;Time
I IPO_Dauer;IPO_Duration
I Wurzel;Root
I Master;Parent
I Number;Number
I <Descriptor>
...

```

After a restart of the controller, the following things occur:

Furthermore, it is to be mentioned that the generation of adaptations with the program "LEX" for RobOffice VRC is to be done. The use of a standard editor and the subsequent compilation with the external compiler (RLLC) can cause problems with the presentation of characters which are located in the enlarged IBM-character set (modified vowels, accents etc.). A description of the character string used in the RSV is given in the documentation of the output and input command.

Anyway, the current command list for the allocation of names has to be considered. The current version is in the appendix.

If the command "OR" is marked with "O" in a language (e.g. for the Spanish language), then it is identical with the PLC-command "O" which has the same spelling in the German language as in the foreign language. This is **not allowed**.

*Example:*

All entered terms such as command words, descriptors, qualifiers and macros have to be clear. Macros must not be labeled like commands. The same applies to commands and macros each being listed one below the other.

*Attention:*

## I WERKZEUG;OUTIL

Insertion of the desired language (e.g. French)

I WERKZEUG;

Delete the non-German text

I WERKZEUG;TOOL

Selection of the command to be changed

As of version 9.0 the file „lang.lib“ for RoboOffice is no longer of importance for language change-over. It is only required with use of the PC-Tool „rfd.exe“ and „rlc.exe“ that are included in the RoboOffice package. The file „lang.lib“ can be generated via the menu selection of the RoboOffice VRC.

The language setting of RoboOffice in the file „lang.lib“ will be changed-over (German <-> English).

As of version 10.0 the MENUCONF - files to be translated must be in the directory S:/\$CONFIG/\$CNF/FOREIGN.

the command words in the files „S:/\$CONFIG/\$CNF/MENUCONF<sup>x</sup>“ (x = 1..4) will be adapted and, if necessary, the macro calls in the programs will be translated and macro definitions (macro programs) will be renamed.

the commands will be initialized internally in the control in the corresponding language.

Macro definitions in SKYCNF resp. MENUCNFx at the time being are also not considered with automatic translation.

Macro parameters will be defined in the definition head of the macro program itself. This definition head will not be considered by the translation program. The macro parameters will be maintained in the original language when called in a robot program.

Especially with the translation of macros, in a greater amount the procedure may be prolonged.

The translation procedure may take several minutes which depends from the number of terms which have to be translated.

**Language conversion finished!**

Please wait!

**Macros will be adapted!**

Please wait!

**Menucnf-files will be adapted!**

Please wait!

**Language conversion active!**

The translation procedure will be accompanied by corresponding messages on the PHG:

### 7.3 TRANSLATION PROCEDURE

.../GERMAN directory.  
As of version 10.0 the other language-dependent programs no longer need be read in anew. The programs are either in the directory S:/\$CONFIG/\$CNF/GERMAN and additionally in the directory S:/\$CONFIG/\$CNF/GERMAN. The files are only used from this directory, if the corresponding file is missing in the .../GERMAN directory.

language dependent programs have to be manually read in. macro programs and macro calls in the programs will also be translated to the corresponding prescription in "LEX". All other German language. The programs MENUCNFx as well as the system internal language will be initialized in

The reverse translation from the foreign language to German will be analog. The value of the system variable \_LANGUAGE will be reset and the system will be restarted afterwards. The will be reset and the system will be restarted afterwards. The German language definition MENUCNFx as well as the system internal language will be initialized in

In case of need, comments, variable and program names have to be manually translated.

they have to be translated by typing.  
As of version 10.0 the files FKEYCNF1 ... FKEYCNFA, MSG\_CNF, SEYCNF1 ... SKYCNF4, UKEYCNF1 ... UKEYCNF4 (if required) and also TXT\_FMT must be stored in the directory S:/\$CONFIG/\$CNF/FOREIGN. Now as before

must be read in to S:/\$CONFIG/\$MAC.

TXT\_FMT

now as before have to be manually translated and to be read in to S:/\$CONFIG/\$CNF, the file

MSG\_CNF,

FKEYCNF1 ... FKEYCNFA,

The files:

It is important that the translation from German to a foreign language and the reversed translation are performed with the same **translation prescription**. Only in this case, a clear language change-over is possible.

Therefore, the program "LEX" has absolutely to be saved (Backup / archival storage).

---

```

I MARK;LABEL
I SPRUNG;BRANCH
I WARTZEIT;WAIT
I KOPIERE;COPY
I MIN_WEG;MIN PATH
I REL_ACHSE;REL AXIS
I TRANS;TRANS
I RELATIV;RELATIVE
I BEREFH_REL;CALC_REL
I BEWEG_ART;INTERPOL
I WERKZEUG;TOOL
I BAHN_BESCHL_PATH_ACCEL
I PTP_BESCHL_PTP_ACCEL
I BAHN_GESCHW;PATH_VELOC
I PTP_GESCHW;PTP_VELOC
I PALETTE;PALETTE
I C;C
I SENSOR;SENSOR
I PROC_CTRL;PROC_CTRL
I PENDELN;OSCILLATE
I SCHWEISSEN;WELDING
I INDEX;INDEX
I ACHSE;AXIS
I STOP;STOP
I VEK_SUB;VEC_SUB
I VEK_ADD;VEC_ADD
I VEK_BETRAG;VEC_VALUE
I MODULO;MODULE
I DIV;DIV
I MUL;MUL
I SUB;SUB
I ADD;ADD
I NEG;NEG
I BETRAG;ABS_VALUE
I LOK_KONST;LOC_CONST
I KONST;CONST
I LOK_VAR;LOC_VAR
I VAR;VAR
I SCHIEBE_L;SHIFT_L
I SCHIEBE_R;SHIFT_R
I INVERT;INVERT
I EXKL_ODER;EXCL_OR
I UND;AND
I ODER;OR
I ANA_AUSG;ANA_OUTP
I ANA_EING;ANA_INP
I <Command>
MPR "LEX"

```

Example of a complete "LEX"-program for translation from German to English and reverse.

## 8 Appendix

---

I PROGRAMM;PROGRAM  
 I U PROG;CALL  
 I POSITION;POSITION  
 I KOP\_OFFSET;COPY\_OFFSET  
 I LIBO\_SENSOR;ARC\_SENSOR  
 I TREECOPY;FREE\_COPY  
 I SEND\_PRG;SEND\_PRG  
 I LOAD\_PRG;LOAD\_PRG  
 I SET\_PRG;SET\_PRG  
 I DEL\_PRG;DEL\_PRG  
 I SEND\_VAR;SEND\_VAR  
 I LOAD\_VAR;LOAD\_VAR  
 I SET\_MODE;SET\_MODE  
 I VERLAENG;EXTEND  
 I TRAFO\_6D;TRAFO\_6D  
 I P\_WINKEL;OSC\_ANGLE  
 I IST\_POS;ACTUAL\_POS  
 I BAHN\_RADIUS\_PATH\_RADIUS  
 I EINGABE\_INPUT  
 I AUSGABE\_OUTPUT\_VAR  
 I TRIG\_FUNC;TRIG\_FUNC  
 I TECH\_MKOR;TECH\_TCOR  
 I VEK\_LAENG;VEC\_LENGTH  
 I CONV\_SYNC;CONV\_SYNC  
 I GESELL\_CD\_REL;VELOC\_CD\_REL  
 I FAHR\_AXS;MOVE\_AXES  
 I TECH\_SSCW;TECH\_SWCD  
 I PRG\_UNBENENN;PRG\_RENAME  
 I BAHN\_DISTANCE;PATH\_DIST  
 I TRAFO\_POS;TRAFO\_Pos  
 I WURZEL;SQRT  
 I HOHE\_SATZ;GET\_STEP  
 I SCHREIBE\_SATZ;WRITE\_STEP  
 I LOESCHE\_SATZ;DELETE\_STEP  
 I ERZUEGE\_FFAID;CREATE\_PATH  
 I TRENNE\_FFAID;SEPARATE\_PATH  
 I KOPIERE\_STR;COPY\_STR  
 I VERGLEICHE\_STR;COMPARE\_STR  
 I LAENGE\_STR;LENGTH\_STR  
 I SUCHE\_STR;SEARCH\_STR  
 I NUM\_STR;NUM\_STR  
 I STR\_NUM;STR\_NUM  
 I BENENNEN\_PROG;RENAME\_PROG  
 I WAELLE\_PROG;SELECT\_PROG  
 I SUCHEN\_MARK;SEARCH\_MARK  
 I MARKIERUNG;MARKER  
 I KOPIERE\_PROG;COPY\_PRG  
 I ERZUEGE\_PROG;CREATE\_PROG  
 I TAV\_CTRL;TAV\_CTRL  
 I SIMU\_TASTE;SIMU\_KEY  
 I BLOCK\_EXPORT;BLOCK\_EXPORT  
 I BLOCK\_IMPORT;BLOCK\_IMPORT  
 I LESE\_TASTE;READ\_KEY  
 I MEMORY;MEMORY  
 I TAV\_CTRL;TAV\_CTRL  
 I SIMU\_TASTE;SIMU\_KEY  
 I BLOCK\_EXPORT;BLOCK\_EXPORT  
 I BLOCK\_IMPORT;BLOCK\_IMPORT

---

```

I LOE_MARK_PRG;DEL_MARK_PRG
I SUCH_EERSETZ_FIND_ERPLACE
I DIRSATZ_DIRSSTEP
I MELDUNG_MESSAGE
I SCHNITTPLKT_INTESECTIO
I MITTELPKT_CENTRE
I MAUS_KALIB_MOUSE_CAL
I MENU_MENUE_MENU
I BAHN_ZEIT_PATH_TIME
I UEBERSCHL_FLVBY
I AX_CALIB_AX_CALIB
I BERECH_HW_CALC_FRAME
I CHECK_CODE_CHECK_CODE
I PWD_ENCODE_PWD_ENCODE
I MS_TRAFO_EIN_MS_TRAFO_ON
I MS_TRAFO_AUS_MS_TRAFO_OFF
I KALIBREREN_CALIBRATION
I SCHR_BIT_WRITE_BIT
I TESTE_BIT_TEST_BIT
I WARTE_BIT_WAIT_BIT
I IMPULS_PULSE
I TESTE_TEST
I I_I
I SUCH_BIN_SEARCH_BIN
I SYS_UPDATE_SYS_UPDATE
I LOAD_TFTP_LOAD_TFTP
I SET_IP_SET_IP
I BAHN_SCHALT_PATH_SWITCH
I WURZEL_ROOT
I ZEIT_TIME
I IPO_Dauer_IPO_Duration
I IPADDRESS_IPAddress
I PROG_Name_Prog_Name
I VORLAUF_Advance
I MAX_ZETT_SS_Max_Time_SS
I VARIABLE_Variable
I BIT_Nr_Bit_No
I BYTE_BYTET
I PEGEL_Level
I PASSWORT_Password
I LOGINNAME_LogInName
I LOGINNAME_LogInName
I VWAER_1_VWAER_1
I VWAER_2_VWAER_2
I VWAER_3_VWAER_3
I RADIUS_RADIUS
I MITTELPKT_Center
I PUNKT1_Point1
I PUNKT2_Point2
I PUNKT3_Point3
I RADII_RADIUS
I VWAER_3_VWAER_3
I VWAER_2_VWAER_2
I VWAER_1_VWAER_1
I ABSSTAND_DISTANCE

```

---

I Schmittypkt;Intersection  
 I Gerade2\_P2;Straight2\_P1  
 I Gerade2\_P2;Straight2\_P2  
 I Gerade2\_P1;Straight2\_P1  
 I Gerade1\_P1;Straight1\_P2  
 I Gerade1\_P2;Straight1\_P1  
 I MeldeNummer;Message\_no.  
 I Einfluegeln;Paste  
 I Offset;Offset  
 I Par;Par  
 I Instr;Instr  
 I Adresse;Address  
 I Nachkommast;Dec\_Places  
 I Optionen;Options  
 I StartIndex;Start\_Index  
 I Muster;Pattern  
 I Ergebnis;Result  
 I Verzeichnisse;Directory  
 I Lauwerk;Device  
 I Frame-Nr;Frame-No  
 I Prog\_Nam2;Prog\_Nam2  
 I [mm];[mm]  
 I Frame;Frame  
 I Prog\_Nam1;Prog\_Nam1  
 I Dauer;Period  
 I Prog\_Nam2;Prog\_Nam2  
 I Index;Index  
 I StrIndex\_1;StrIndex\_1  
 I StrIndex\_2;StrIndex\_2  
 I Ergbnis;Result  
 I StartIndex;Start\_Index  
 I Muster;Pattern  
 I Ergebnis;Result  
 I Lauwerk;Device  
 I Frame-Nr;Frame-No  
 I Prog\_Nam1;Prog\_Nam1  
 I Text;Text  
 I Toleranz [Abs];Tolerance [abs]  
 I Radii [mm];Radius [mm]  
 I Quell\_Prog;Source\_Prog  
 I Zielp\_Prog;Dest\_Prog  
 I Länge;Length  
 I Datei\_Prog;Dest\_Prog  
 I Quell\_Prog;Source\_Prog  
 I Radii [mm];Radius [mm]  
 I Füllerfaktor;Fillerfactor  
 I Abtastzeit [ms];Sample [ms]  
 I Quelle [ms];Source [ms]  
 I Ref\_Ziel;Ref\_Target  
 I Ref\_Ziel;Ref\_Target  
 I Steuerwort;Ctrl\_Word  
 I Alpha;ALPHA  
 I Beta;BETA  
 I Frame;Frame  
 I Marke;Label  
 I Text;Text  
 I Toleranz [Abs];Tolerance [abs]  
 I Quelle;Source  
 I Ref\_Ziel;DestINATION  
 I Zielp\_Label;DestINATION  
 I Quelle;Source  
 I [s];[s]  
 I [mm/s];[mm/s]  
 I Bit\_Nr;Bit\_No  
 I Achse;Axis  
 I Zield\_Vek;Dest\_Vec  
 I Vektor;Vector  
 I Anzahl;Amount

```

I Variable;Variable
I Op_1;Op_1
I Ziel_Var;Dest_Var
I Kanal;Channel
I Spannung;Voltage
I P-Var;P-Var
I Ziel_Prog;Dest_Prog
I Name;Name
I IVar_2;IVar_2
I IVar_1;IVar_1
I RVar_2;RVar_2
I RVar_1;RVar_1
I PVar_2;PVar_2
I PVar_1;PVar_1
I Wert;Value
I Loeesch.[J=1/N=0];del.[Y=1/N=0]
I ausf.[J=1/N=0];exec.[Y=1/N=0]
I edit.[J=1/N=0];edit[Y=1/N=0]
I strukt.[J=1/N=0];struct.[Y=1/N=0]
I umben.[J=1/N=0];rename[Y=1/N=0]
I edit.[J=1/N=0];edit[Y=1/N=0]
I Prog_Name;Prog_Name
I Kennung;Password
I Ziel;Destination
I #OEFFNEN;#OPEN
I #SCHLIESSEN;#CLOSE
I #MOMENT_KONST;#MOMENT_CONST
I #AKTIV;#ACTIVE
I #PASSIV;#PASSIVE
I <Equation>
I Queue;Source
I #INIT;#INIT
I #SOILWERT;#REFERENCE
I #K-FAKTOR;#K-FACTOR
I #POSITION;#POSITION
I #WERKZEUG;#TOOL
I #PTP;#PTP
I #RESET;#RESET
I #STOP;#STOP
I #START;#START
I #HAND;#HAND
I #PSI;#PSI
I #VORHANDEN;#EXISTS
I #FEHLT;#MISSING
I #FREE;#REF
I #BINAER;#BINARY
I #ANALOG;#ANALOG
I #IPD;#IPD
I #EIN;#ON
I #AUS;#OFF
I #COS;#COS
I #SIN;#SIN
I #TAN;#TAN
I #ARCSIN;#ARCSIN
I #ARCCOS;#ARCCOS
I #ARCTAN;#ARCTAN

```

```

END
I NOCHEINMACRO;NEXT_MACRO
I TESTMACRO;MACRO_TEST
I <MACRO>
I #BIN:#BIN
I #HEX:#HEX
I #DEC:#DEC
I #SYSTEM:#SYSTEM
I #BEGIIN:#START
I #ZEIT:#TIME
I #ENDE:#END
I #WEG:#PATH
I #WAIT:#WAIT
I #EINGANG:#INPUT
I #VARIABLE:#VARIABLE
I #MERKER:#MERKER
I #AUSGANG:#OUTPUT
I #SYNCHRON:#SYNCHRONIZE
I #EINRICHTEN:#CALIBRATE
I #SCHREIBEN:#WRITE
I #REAL:#REAL
I #LONG:#LONG
I #WORD:#WORD
I #BYTE:#BYTE
I #LESEN:#READ
I #ANFUEGEN:#APPEND
I #UEBERSCHR:#OVERWRITE
I #FRAME:#FRAME
I #KART:#CART
I #ACHS:#AXIS
I #VORZ:#SIGN
I #SPLINE:#SPLINE
I #ZIRK:#CIRC
I #LINEAR:#LINEAR
I #PTP:#PTP
I #PROG:#PROG
I #FREE:#FREE
I #DRUCKER:#PRINTER
I #AWP:#AWP
I #ZEIT:#TIME
I #HKS_V:#HCS_V
I #HKS_U:#HCS_U
I #PKS:#OCS

```

### Current command list of the RSV

Tab. 0-1: RSV-commands - survey / software version as of 10.0

CONTR	LOG	MATH	MOVE	PERI	PLC	POS	PROG	SPEC	VAR
F1	WAIT_BIT	WRITE_BIT	ADD	INTERPOL	ANA_OUTP	U	#N	PROGRAM	I
F2	TEST_BIT	INVERT	SUB	PTP_VELOC	ANA_INP	UN	#P	CALL	SEARCH_BIN
F3	TEST	SHIFT_R	MUL	PTP_ACCEL		=		SELECT_PROG	COPY_OFFSET
F4	WAIT	SHIFT_L	DIV	PATH_VELOC		O		COPY_PRG	TRAFO_6D
F5	BRANCH	OR	MODULO	PATH_ACCEL		ON		DELETE_PROG	LOC_CONST
F6	LABEL	AND	NEG	FLYBY	SEND_TELE	:		TEST_PRG	C
F1	VARLIST	EXCL_OR	VEC_ADD	TOOL	LOAD_FTP	S		CREATE_PROG	OSCILLATE
F2		VEC_SUB	PATH_RADIUS	SEND_FTP	R		RENAME_PROG	OSC_ANGLE	
F3		VEC_LENGTH	PATH_DIST		L			SENSOR	
F4		VEC_VALUE	PATH_TIME		T			TECHIPO	
F5	OUTPUT_VAR		PATH_SWITCH		LD			ARC_SENSOR	
F6	INPUT		AXIS		TD			PALLET	
F1		ABS_VALUE	RELATIVE		SU			MS_TRAFO_ON	
F2		TRIG_FUNC	CALC_REL		JP			MS_TRAFO_OFF	
F3		SQRT	ACTUAL_POS		M			CALIBRATION	

F4		INTERSECTION	VAR_POS	U(		MOUSE_CAL
F5	CENTER	TRAFO_POS	Oc	)		ROOTLAYER
F6		EXTEND				TOPLAYER
F1		MIN_PATH	+			WELDING
F2		EXT_TOOL	-			PULSE
F3		OPTIMIZE	MU			INDEX
F4		LOAD	D1			BRAKE_TEST
F5			SH			
F6			RO			
F1			=?			COPY_STR
F2			<			COMPARE_STR
F3			>			SEARCH_STR
F4			≤			LENGTH_STR
F5			^			STR_NUM
F6			≤=			NUM_STR
F1			K			SEARCH_MARK
F2			ZR			LABELR
F3			ZV			GET_STEP
F4			TA			WRITE_STEP

F5				HD	BLOCK_EXPORT
F6				DH	BLOCK_IMPORT
F1				XO	MESSAGE
F2				XON	MENU
F3				XOFF	READ_KEY
F4				ST	SIMU_KEY
F5					TAV_CNTRL
F6					MEMORY
F1					SEPARATE_PATH
F2					CREATE_PATH
F3					
F4					
F5					DELETE_STEP
F6					

