

# 24

## 24. Jazyk PLC836 - referenční slovník

<b>ABS</b>	syntax: ABS [DWRD]	stránka: 3 - 21
Bezoperandová instrukce. Instrukce ABS provádí absolutní hodnotu DR registru.		
<b>AD</b>	syntax: AD [TYPE.]adr	stránka: 3 - 18
Aritmetická instrukce s operandem. Instrukce AD sečte obsah DR registru s obsahem paměti, na kterou ukazuje operand, nebo s konstantou.		
<b>ANALOG</b>	syntax: ANALOG hodn[,osa]	stránka: 12 - 3
Vyslání hodnoty z DR registru nebo operandu na aktivní analogový výstup.		
<b>ANALOG_PORT</b>	syntax: ANALOG_PORT port	stránka: 12 - 3
Přiřazení aktivního portu ke konkrétnímu fyzickému portu.		
<b>ANDB</b>	syntax: ANDB bun	stránka: 3 - 20
Logický AND po bitech se slovem BYTE, WORD nebo DWORD		
<b>APPEND_T_MACRO</b>	syntax: APPEND_T_MACRO ....	stránka: 22 – 6
Instrukce slouží pro připojení názvu k rezervovaným názvům překladače.		
<b>AX_SPI_x</b>	syntax: AX_SPI_x	stránka: 12 - 6
Instrukce AX_SPI_x změni polohovou vazbu na rychlostní vazbu příslušné osy.		
<b>BCD</b>	syntax: BCD	stránka: 3 - 20
Bezoperandová instrukce. Instrukce BCD převede číslo v binárním tvaru v DR registru na BCD číslo.		
<b>BCDSTR</b>	syntax: BCDSTR text	stránka: 3 - 29
Převod BCD čísla na řetězec.		
<b>BEX</b>	syntax: BEX	stránka: 4 - 4
Bezoperandová instrukce pro logické sekvenční celky. Instrukce BEX definuje začátek logické podmínky před instrukcemi EX0, EX1, TEX0 a TEX1.		
<b>BIN</b>	syntax: BIN	stránka: 3 - 20
Bezoperandová instrukce. Instrukce BIN převede číslo ve tvaru BCD v DR registru na binární číslo.		
<b>BINSTR</b>	syntax: BINSTR text	stránka: 3 - 29
Převod binárního čísla na řetězec.		
<b>CA</b>	syntax: CA	stránka: 3 - 6
Logická bezoperandová instrukce. Instrukce CA provádí negaci obsahu registru RLO.		
<b>CD</b>	syntax: CD citac	stránka: 3 - 17
Čítač dolů závislý na DR, RLO a DFTM.		

<b>CHECK</b> Pomocný příkaz. Kontroluje vyrovnanost zásobníku.	syntax: CHECK	stránka: 3 - 26
<b>CLEAR</b> Instrukce pro nulování paměti	syntax: CLEAR zac, konec	stránka: 3 - 25
<b>CONDR</b> Bezoperandová instrukce pro konverzi dat. Instrukce provede konverzi DR do RLO.	syntax: CONDR	stránka: 3 - 24
<b>CONTROL_T_MACRO</b> Instrukce slouží pro řízení vykonávání všech uživatelských instrukcí.	syntax: CONTROL_T_MACRO par	stránka: 22 - 7
<b>CPU04</b> Instrukce vykoná všechny změny nutné pro přechod na procesor CPU04, ale nemodifikuje instrukce jazyka PLC836 pro rozvoj do "assembleru 386".	syntax: CPU04	stránka: 3 - 26
<b>CU</b> Čítač nahoru závislý na DR, RLO a DFTM.	syntax: CU citac	stránka: 3 - 17
<b>CUBCD</b> BCD čítač nahoru závislý na DR, RLO a DFTM.	syntax: CUBCD citac	stránka: 3 - 17
<b>DATA</b> Pomocný příkaz pro začátek definice dat PLC programu.	syntax: DATA	stránka: 5 - 1
<b>DCR</b> Bezoperandová instrukce DCR dekrementuje DR register.	syntax: DCR [DWRD]	stránka: 3 - 20
<b>DEBUG</b> Instrukce DEBUG vymezuje oblast programu, ve které je povoleno trasování.	syntax: DEBUG [off]	stránka: 7 - 1
<b>DEB_XCHG</b> Instrukce DEB_XCHG umožní změny paměti PLC podle zadání z panelu systému.	syntax: DEB_XCHG	stránka: 7 - 1
<b>DEF_T_MACRO</b> Instrukce DEF_T_MACRO označuje začátek definice uživatelského makra, nebo uživatelské instrukce.	syntax: DEF_T_MACRO nazev [par,..]	stránka: 22 – 2
<b>DFCOND</b> Instrukce pro podmínkové bloky. Definice klíčových bitů.	syntax: DFCOND <bit1,bit2..>	stránka: 3 - 25
<b>DFM</b> Definice bitů v paměti.	syntax: DFM bit1,bit2,,,,,,	stránka: 3 - 3
<b>DFTMxy</b> Časově závislé programové bloky. Úsek program aktivován po 0,1; 1; 10 a 100 vteřinách.	syntax: DFTMxy konec	stránka: 3 - 15
<b>DIVB</b> Aritmetická instrukce. Instrukce MULB vynásobí registr DR s obsahem paměti.	syntax: DIVB [TYPE.]adr	stránka: 3 - 18
<b>DS</b> Definice datových proměnných a inicializace paměti.	syntax: DS n	stránka: 3 - 4
<b>EQ</b> Logická instrukce. Instrukce EQ provádí porovnání DR registru s obsahem paměti nebo s konstantou. Je-li DR roven obsahu paměti, nastaví RLO do log.1	syntax: EQ [TYPE.]adr	stránka: 3 - 22

<b>EQ1</b>	syntax: EQ1 [TYPE.]adr	stránka: 3 - 23
Logická instrukce. Instrukce EQ provádí podmíněné porovnání DR registru s obsahem paměti nebo s konstantou, jenom když RLO=1.		
<b>EQUI</b>	syntax: EQUI konst,hodn	stránka: 3 - 4
Instrukce EQUI definuje konstanty a symboly.		
<b>END_T_MACRO</b>	syntax: END_T_MACRO [ priznaky ]	stránka: 22 - 3
Instrukce END_T_MACRO označuje konec definice uživatelského makra, nebo uživatelské instrukce.		
<b>ESET</b>	syntax: ESET error	stránka: 14 - 2
Instrukce pro nastavení PLC chyby		
<b>ESET1</b>	syntax: ESET1 error	stránka: 14 - 2
Podmíněné nastavení PLC chyby		
<b>EX</b>	syntax: EX	stránka: 4 - 4
Instrukce pro logické sekvenční celky. Instrukce EX způsobí nepodmíněné zastavení sekvence na dobu jednoho cyklu PLC programu.		
<b>EX0</b>	syntax: EX0	stránka: 4 - 4
Instrukce pro logické sekvenční celky. Instrukce EX0 způsobí podmíněné zastavení sekvence po dobu, pokud RLO=0.		
<b>EX1</b>	syntax: EX1	stránka: 4 - 4
Instrukce pro logické sekvenční celky. Instrukce EX1 způsobí podmíněné zastavení sekvence po dobu, pokud RLO=1.		
<b>FL</b>	syntax: FL 0/1,bit	stránka: 3 - 8
Instrukce pro zápis bitu do paměti. Instrukce FL naplní log.0 nebo log.1 do bitu v paměti.		
<b>FL1</b>	syntax: FL1 0/1,bit	stránka: 3 - 9
Instrukce pro podmíněný zápis bitu do paměti. Instrukce FL naplní log.0 nebo log.1 do bitu v paměti, jenom když je RLO=1.		
<b>GE</b>	syntax: GE [TYPE.]adr	stránka: 3 - 22
Logická instrukce. Instrukce GE provádí porovnání DR registru s obsahem paměti nebo s konstantou. Je-li DR větší nebo roven než obsah paměti, nastaví RLO do log.1		
<b>GT</b>	syntax: GT [TYPE.]adr	stránka: 3 - 22
Logická instrukce. Instrukce GT provádí porovnání DR registru s obsahem paměti nebo s konstantou. Je-li DR větší než obsah paměti, nastaví RLO do log.1		
<b>INP</b>	syntax: INP port,bun	stránka: 9 - 1
Instrukce pro snímání binárních vstupů. Instrukce INP sejme jeden osmi bitový port z fyzické adresy.		
<b>IN_MUX</b>	syntax: IN_MUX port,bun	stránka: 9 - 2
Instrukce pro snímání binárních vstupů. Instrukce IN_MUX sejme 64 bitů z osmi portů do paměti.		
<b>IN_FAST</b>	syntax: IN_FAST bun	stránka: 9 - 2
Instrukce pro snímání rychlých binárních vstupů. Instrukce sejme 8 rychlých vstupů		
<b>INR</b>	syntax: INR [DWRD]	stránka: 3 - 20
Bezoperandová instrukce. Instrukce INR zvětší obsah DR registru o jedničku.		
<b>INRBCD</b>	syntax: INRBCD [DWRD]	stránka: 3 - 20
Bezoperandová instrukce. Instrukce INRBCD zvětší obsah DR registru o jedničku v BCD kódu.		
<b>INTERSTACK</b>	syntax: INTERSTACK	stránka: 3 - 26
Pomocná instrukce. Instrukce INTERSTACK způsobí používání vlastního zásobníku procesoru.		

<b>INV</b>	syntax: INV [DWRD]	stránka: 3 - 21
Bezoperandová instrukce. Instrukce INV provede negaci obsahu DR registru.		
<b>JL0</b>	syntax: JL0 adr	stránka: 3 - 10
Instrukce pro větvení programu. Instrukce JL0 zajistí skok na zadanou adresu pouze v případě že RLO=0.		
<b>JL1</b>	syntax: JL1 adr	stránka: 3 - 10
Instrukce pro větvení programu. Instrukce JL1 zajistí skok na zadanou adresu pouze v případě že RLO=1.		
<b>JUM</b>	syntax: JUM adr	stránka: 3 - 10
Instrukce pro větvení programu. Instrukce JUM zajistí nepodmíněný skok na zadanou adresu.		
<b>LA</b>	syntax: LA [[-]bit]	stránka: 3 - 6
Instrukce pro logické operace. Instrukce LA provede logický součin registru RLO s bitem v paměti nebo se zásobníkem.		
<b>LDR</b>	syntax: LDR [-]bit	stránka: 3 - 5
Instrukce pro logické operace. Instrukce LDR provede plnění registru RLO s bitem v paměti.		
<b>LE</b>	syntax: LE [TYPE.]adr	stránka: 3 - 22
Logická instrukce. Instrukce LE provádí porovnání DR registru s obsahem paměti nebo s konstantou. Je-li DR menší nebo roven než obsah paměti, nastaví RLO do log.1.		
<b>LO</b>	syntax: LO [[-]bit]	stránka: 3 - 6
Instrukce pro logické operace. Instrukce LO provede logický součet registru RLO s bitem v paměti nebo se zásobníkem.		
<b>LOD</b>	syntax: LOD [-][TYPE.]adr	stránka: 3 - 13
Instrukce LOD zajistí načtení obsahu paměti nebo konstanty do DR registru.		
<b>LT</b>	syntax: LT [TYPE.]adr	stránka: 3 - 22
Logická instrukce. Instrukce LT provádí porovnání DR registru s obsahem paměti nebo s konstantou. Je-li DR menší než obsah paměti, nastaví RLO do log.1.		
<b>LX</b>	syntax: LX [[-]bit]	stránka: 3 - 6
Instrukce pro logické operace. Instrukce LX provede nonekvivalenci registru RLO s bitem v paměti nebo se zásobníkem.		
<b>MECH_BEGIN</b>	syntax: MECH_BEGIN mech	stránka: 4 - 3
Instrukce pro definici začátku logického sekvenčního celku.		
<b>MECH_END</b>	syntax: MECH_END mech	stránka: 4 - 3
Instrukce pro definici konce logického sekvenčního celku.		
<b>MECH_INIT</b>	syntax: MECH_INIT mech	stránka: 4 - 3
Instrukce pro inicializaci logického sekvenčního celku.		
<b>MSET</b>	syntax: MSET msg	stránka: 14 - 7
Instrukce pro nastavení hlášení.		
<b>MSET1</b>	syntax: MSET1 msg	stránka: 14 - 7
Instrukce pro nastavení hlášení.		
<b>MULB</b>	syntax: MULB [TYPE.]adr	stránka: 3 - 18
Aritmetická instrukce s operandem. Instrukce MULB vynásobí obsah DR registru obsahem paměti, na kterou ukazuje operand, nebo konstantou.		
<b>MV</b>	syntax: MV zdroj,cil,pocet	stránka: 3 - 25
Instrukce pro přesun oblasti paměti.		

<b>ORB</b>	syntax: ORB bun	stránka: 3 - 20
Logický OR po bitech se slovem BYTE, WORD nebo DWORD		
<b>OUTP</b>	syntax: OUTP port,bun	stránka: 9 - 3
Instrukce pro vysílání binárních výstupů. Instrukce OUTP vyšle jeden osmi bitový port na fyzické adresy.		
<b>P386</b>	syntax: P386	stránka: 3 - 26
Instrukce modifikuje instrukce jazyka PLC836 pro rozvoj do "assembleru 386" pro procesor CPU04.		
<b>PIS_CLEAR</b>	syntax: PIS_CLEAR	stránka: 5 - 5
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_CLEAR definuje začátek modulu pro nulování PLC programu.		
<b>PIS_CLEAR_END</b>	syntax: PIS_CLEAR_END	stránka: 5 - 5
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_CLEAR_END definuje konec modulu pro nulování PLC programu.		
<b>PIS_CONT</b>	syntax: PIS_CONT	stránka: 5 - 6
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_CONT definuje začátek nepřerušitelného modulu PLC programu.		
<b>PIS_CONT_END</b>	syntax: PIS_CONT_END	stránka: 5 - 6
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_CLEAR_END definuje konec nepřerušitelného modulu PLC programu.		
<b>PIS_FAST</b>	syntax: PIS_FAST	stránka: 5 - 6
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_FAST definuje začátek modulu pro obsluhu rychlých procesů.		
<b>PIS_FAST_END</b>	syntax: PIS_FAST_END	stránka: 5 - 6
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_FAST_END definuje konec modulu pro obsluhu rychlých procesů.		
<b>PIS_HALT</b>	syntax: PIS_HALT	stránka: 5 - 6
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_HALT definuje začátek modulu pro obsluhu závažné chyby.		
<b>PIS_HALT_END</b>	syntax: PIS_HALT_END	stránka: 5 - 6
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_HALT_END definuje konec modulu pro obsluhu závažné chyby.		
<b>PIS_INIT</b>	syntax: PIS_INIT	stránka: 5 - 5
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_INIT definuje začátek modulu pro inicializaci proměnných a mechanismů PLC programu.		
<b>PIS_INIT_END</b>	syntax: PIS_INIT_END	stránka: 5 - 5
Instrukce pro tvorbu struktury PLC programu. Instrukce PIS_INIT_END definuje konec modulu pro inicializaci proměnných a mechanismů PLC programu.		
<b>POS_CONTROL_x</b>	syntax: POS_CONTROL_x stv,rzch,MP	stránka: 19 - 4
Instrukce pro řízení pohybu polohovací jednotky.		
<b>POS_INIT_x</b>	syntax: POS_INIT_x	stránka: 19 - 3
Instrukce pro inicializaci polohovací jednotky.		
<b>POS_MODE_x</b>	syntax: POS_MODE_x con,pos,zrych,doj	stránka: 19 - 4
Instrukce pro naprogramování polohovací jednotky.		

<b>POS_RESET_x</b> Instrukce pro reset polohovací jednotky.	syntax: POS_RESET_x	stránka: 19 - 4
<b>PRIPRAVNE_FUNKCE</b> Instrukce pro tvorbu struktury PLC programu. Instrukce PRIPRAVNE_FUNKCE definuje začátek modulu pro přípravné funkce PLC programu.	syntax: PRIPRAVNE_FUNKCE	stránka: 5 - 4
<b>PRIPRAVNE_FUNKCE_END</b> Instrukce pro tvorbu struktury PLC programu. Instrukce PRIPRAVNE_FUNKCE_END definuje konec modulu pro přípravné funkce PLC programu.	syntax: PRIPRAVNE_FUNKCE_END	stránka: 5 - 4
<b>PROC_BEGIN</b> Definice začátku procedury	syntax: PROC_BEGIN proc	stránka: 3 - 25
<b>PROC_CALL</b> Definice začátku procedury	syntax: PROC_CALL proc	stránka: 3 - 25
<b>PROC_END</b> Definice konce procedury	syntax: PROC_END proc	stránka: 3 - 25
<b>PROVOZ_VYSTUP</b> Instrukce pro tvorbu struktury PLC programu. Instrukce PROVOZ_VYSTUP definuje začátek modulu pro průběžné funkce PLC programu.	syntax: PROVOZ_VYSTUP	stránka: 5 - 5
<b>PROVOZ_VYSTUP_END</b> Instrukce pro tvorbu struktury PLC programu. Instrukce PROVOZ_VYSTUP_END definuje konec modulu pro průběžné funkce PLC programu.	syntax: PROVOZ_VYSTUP_END	stránka: 5 - 5
<b>RAMP</b> Instrukce pro zadávání analogového napětí. Instrukce způsobí postupné zvětšování nebo zmenšování hodnoty podle zadané strmosti.	syntax: RAMP vysl.strm	stránka: 12 - 8
<b>REGUL_x</b> Instrukce pro nastavení sady parametrů regulátorů v servosmyčce.	syntax: REGUL_x sada	stránka: 13 - 1
<b>RL</b> Bezoperandová instrukce. Instrukce RL provede logický posuv DR registru o "n" bitů vlevo.	syntax: RL [TYPE.]n	stránka: 3 - 21
<b>RR</b> Bezoperandová instrukce. Instrukce RR provede logický posuv DR registru o "n" bitů vpravo.	syntax: RR [TYPE.]n	stránka: 3 - 21
<b>SEKV</b> Instrukce pro vysílání řídicí sekvence do CNC systému. Instrukce SEKV vysílá kódy tlačítek, režimů, formátů a menu struktur.	syntax: SEKV a1[,a2,a3,a4]	stránka: 11 - 1
<b>SEKV_END</b> Instrukce pro vysílání řídicí sekvence do CNC systému. Instrukce SEKV_END ukončí zápis sekvence a způsobí vyslání do CNC systému.	syntax: SEKV_END	stránka: 11 - 1
<b>SPI_AX_x</b> Instrukce SPI_AX_x změni rychlostní vazbu vřetena na polohovou vazbu příslušné osy.	syntax: SPI_AX_x [dojiz]	stránka: 12 - 5
<b>SPI_FILTER</b> Řízení adaptabilního filtru vřetene	syntax: SPI_FILTER bit,prum,adapt	stránka: 12 - 13
<b>START</b> Pomocný příkaz pro začátek programového kódu PLC programu.	syntax: START	stránka: 5 - 1
<b>STO</b> Instrukce STO zajistí zápis obsahu DR registru do paměti.	syntax: STO [TYPE.]adr	stránka: 3 - 14

<b>STO1</b> Instrukce STO zajistí podmíněný zápis obsahu DR registru do paměti, jenom když RLO = log.1.	syntax: STO1 [TYPE.]adr	stránka: 3 - 14
<b>STOP</b> Pomocný příkaz pro konec programového kódu PLC programu.	syntax: STOP	stránka: 5 - 1
<b>STR</b> Definice textového řetězce	syntax: STR n[,text]	stránka: 3 - 27
<b>STRADD</b> Spojení textových řetězců	syntax: STRADD text1,text2	stránka: 3 - 28
<b>STRCPY</b> Kopírování textových řetězců	syntax: STRCPY text1,text2	stránka: 3 - 28
<b>SU</b> Aritmetická instrukce s operandem. Instrukce SU odečte od obsahu DR registru obsah paměti, na kterou ukazuje operand, nebo konstantu.	syntax: SU [TYPE.]adr	stránka: 3 - 18
<b>SYMBOLTAB</b> Pomocná instrukce. Instrukce SYMBOLTAB způsobí zkrácení tabulky symbolů při překladu. (jen pro překladač INTEL).	syntax: SYMBOLTAB	stránka: 3 - 27
<b>T_INCLUDE</b> Připojení definičního souboru maker a instrukcí ke zdrojovému textu.	syntax: T_INCLUDE soubor	stránka: 22 - 1
<b>T_LOCAL</b> Instrukce T_LOCAL se používá pro specifikování lokálních symbolů v rámci makra.	syntax: T_LOCAL sym1, [ sym2, sym3, ]	stránka: 22 - 4
<b>TEX0</b> Instrukce pro logické sekvenční celky. Instrukce TEX0 způsobí podmíněné zastavení sekvence po dobu, pokud RLO=0, ale maximálně po předem stanovenou dobu.	syntax: TEX0 [TYPE.]citac,doba,err,ch	stránka: 4 - 5
<b>TEX1</b> Instrukce pro logické sekvenční celky. Instrukce TEX1 způsobí podmíněné zastavení sekvence po dobu, pokud RLO=1, ale maximálně po předem stanovenou dobu.	syntax: TEX1 [TYPE.]citac,doba,err,ch	stránka: 4 - 5
<b>TIM</b> Instrukce pro logické sekvenční celky. Instrukce TIM způsobí zastavení provádění sekvence po stanovenou dobu.	syntax: TIM [TYPE.]citac,doba	stránka: 4 - 6
<b>TM</b> Časovač závislý na DR, RLO a bloku DFTM.	syntax: TM [TYPE.]citac	stránka: 3 - 16
<b>TSET</b> Zaslání textu do záznamu událostí	syntax: TSET text	stránka: 14 - 9
<b>TSET1</b> Podmíněné zaslání textu do záznamu událostí	syntax: TSET1 text	stránka: 14 - 9
<b>TSETM</b> Zaslání textu do záznamu událostí v mechanismech	syntax: TSETM text	stránka: 14 - 9
<b>TSETM1</b> Podmíněné zaslání textu do záznamu událostí v mechanismech	syntax: TSETM1 text	stránka: 14 - 9
<b>VERINSTRU</b> Pomocná instrukce pro modifikaci instrukcí PLC programu podle zadané verze instrukce.	syntax: VERINSTRU in1_V1, in2_V1	stránka: 3 - 27

**VSTUP**

syntax: VSTUP

stránka: 5 - 4

Instrukce pro tvorbu struktury PLC programu. Instrukce VSTUP definuje začátek modulu pro vstupní operace PLC programu.

**VSTUP\_END**

syntax: VSTUP\_END

stránka: 5 - 4

Instrukce pro tvorbu struktury PLC programu. Instrukce VSTUP\_END definuje konec modulu pro vstupní operace PLC programu.

**WR**

syntax: WR bit

stránka: 3 - 8

Instrukce pro zápis registru RLO do paměti.

**ZAVERECNE\_FUNKCE**

syntax: ZAVERECNE\_FUNKCE

stránka: 5 - 4

Instrukce pro tvorbu struktury PLC programu. Instrukce VSTUP definuje začátek modulu pro závěrečné funkce v PLC programu.

**XORB**

syntax: XORB bun

stránka: 3 - 20

Logický XOR po bitech se slovem BYTE, WORD nebo DWORD