



Formato file ladder diagram (.LAD)

© 2024 Robox SpA

I contenuti di questo documento sono redatti con la massima cura/diligenza, e sottoposti ad un accurato controllo. Robox Spa tuttavia, declina ogni responsabilità, diretta e indiretta, nei confronti degli utenti e in generale di qualsiasi terzo, per eventuali imprecisioni, errori, omissioni, danni (diretti, indiretti, conseguenti, punibili e sanzionabili) derivanti dai suddetti contenuti.

Tabella dei contenuti

Parte I	Indice generale	6
Parte II	Requisiti minimi	6
Parte III	Formato file (versione 1.2)	7
1	Formato file (versione 1.2)	7
2	Intestazione file	7
3	Tabella strutture	10
4	Tabella variabili	11
5	Tabella blocchi	13
6	Tabella circuiti del blocco	14
7	Tabella elementi circuito	15
8	Tabella inizializzatori variabile	26
9	Tabella stringhe	26
10	Tabella stringhe compresse	27
11	Tabella modifiche live	27
12	Tipi di dati predefiniti	28
13	Variabili predefinite	31
Parte IV	Formato file (versione 1.3)	32
1	Formato file (versione 1.3)	32
2	Intestazione file	33
3	Tabella strutture	35
4	Tabella variabili	37
5	Tabella blocchi	40
6	Tabella circuiti del blocco	41
7	Tabella elementi circuito	42
8	Tabella classi OB	53
9	Tabella metodi OB	55
10	Tabella stringhe	57
11	Tabella stringhe compresse	57
12	Tabella modifiche live	58
13	Tipi di dati predefiniti	58
14	Variabili predefinite	62
15	Identificativi sorgente	62
Parte V	Formato file (versione 1.4)	63
1	Formato file (versione 1.4)	63

2	Intestazione file	63
3	Tabella strutture	66
4	Tabella variabili	68
5	Tabella blocchi	71
6	Tabella circuiti del blocco	72
7	Tabella elementi circuito	73
8	Tabella classi OB	85
9	Tabella metodi OB	86
10	Tabella stringhe	88
11	Tabella stringhe compresse	89
12	Tabella modifiche live	89
13	Tipi di dati predefiniti	90
14	Variabili predefinite	93
15	Identificativi sorgente	94

Parte VI Formato file (versione 1.5) **94**

1	Formato file (versione 1.5)	94
2	Intestazione file	95
3	Tabella strutture	98
4	Tabella variabili	100
5	Tabella blocchi	103
6	Tabella circuiti del blocco	104
7	Tabella elementi circuito	105
8	Tabella classi OB	116
9	Tabella metodi OB	118
10	Tabella file inclusione	120
11	Tabella stringhe	121
12	Tabella stringhe compresse	121
13	Tabella modifiche live	122
14	Tipi di dati predefiniti	122
15	Variabili predefinite	125
16	Identificativi sorgente	126

Parte VII Formato modifiche live (versione 1.1) **127**

1	Formato modifiche live (versione 1.1)	127
2	Intestazione modifiche live	127
3	Tabella comandi live	128

Parte VIII Formato modifiche live (versione 1.3) **133**

1	Formato modifiche live (versione 1.3)	133
---	---	-----

2	Intestazione modifiche live	133
3	Tabella comandi live	135
Parte IX Formato modifiche live (versione 1.4)		141
1	Formato modifiche live (versione 1.4)	141
2	Intestazione modifiche live	141
3	Tabella comandi live	142
Parte X Formato modifiche live (versione 1.5)		148
1	Formato modifiche live (versione 1.5)	148
2	Intestazione modifiche live	149
3	Tabella comandi live	150
Indice		157

Indice generale

Formato del file:

- [Formato file \(versione 1.5\)](#)^[94]
- [Intestazione file](#)^[95]
- [Tabella blocchi](#)^[103]
- [Tabella circuiti del blocco](#)^[104]
- [Tabella elementi del circuito](#)^[105]
- [Tabella file esterni](#)^[120]
- [Tabella strutture](#)^[98]
- [Tabella variabili](#)^[100]
- [Tabella classi OB](#)^[116]
- [Tabella modifiche live incorporate](#)^[122]
- [Tabella stringhe](#)^[121]
- [Tabella stringhe compresse](#)^[121]

Formato delle modifiche live:

- [Formato modifiche live \(versione 1.5\)](#)^[148]
- [Intestazione modifiche live](#)^[149]
- [Tabella comandi live](#)^[150]

Altre informazioni utili:

- [Requisiti minimi](#)^[6]
- [Tipi di dati predefiniti](#)^[122]
- [Variabile predefinite](#)^[122]

Formati della versione 1.2:

- [Formato file \(versione 1.2\)](#)^[7]
- [Formato modifiche live \(versione 1.1\)](#)^[127]

Formati della versione 1.3:

- [Formato file \(versione 1.3\)](#)^[32]
- [Formato modifiche live \(versione 1.3\)](#)^[133]

Formati della versione 1.4:

- [Formato file \(versione 1.4\)](#)^[63]
- [Formato modifiche live \(versione 1.4\)](#)^[141]

Requisiti minimi

I requisiti minimi per l'uso del file ladder diagram (.LAD) sono i seguenti, a seconda della versione del file (altrimenti detto livello di compatibilità):

Versione	Requisiti
v1.5	RTE versione 34.19.10 (o superiore) RDE versione 3.51.1 (o superiore)
v1.4	Per destinazione RTE (PowerPC G2): - RTE versione 33.17.0 (o superiore) - RDE versione 3.7.0 (o superiore) Per destinazione RTE (PowerPC P2020): - RTE versione 34.0.0 (o superiore) - RDE versione 3.10.0 (o superiore)
v1.3	RTE versione 33.14.0 (o superiore) RDE3 versione 3.6.0 (o superiore)
v1.2	RTE versione 33.8.0 (o superiore)

Versione	Requisiti
	RDE3 versione 3.5.1 (o superiore)
v1.1	Non utilizzabile (da convertire in v1.2)
v1.0	Non utilizzabile (da convertire in v1.2)

Formato file (versione 1.2)

Formato file (versione 1.2)

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

Un programma in linguaggio di programmazione ladder/diagram è un singolo file (tipo .LAD) che contiene le informazioni per l'editing (sorgente) che le informazioni per l'esecuzione (tipicamente su firmware RTE o compatibili).

Un programma ladder è composto principalmente da una [intestazione](#)^[7], quindi da varie sezioni tra cui le principali sono:

- [tabella strutture](#)^[10], ovvero i tipi dati definiti dall'utente.
- [tabella variabili](#)^[11], ovvero le variabili generali del programma.
- [tabella blocchi](#)^[13], ovvero le sub-routines in altri linguaggi di programmazione;
- [tabella modifiche live incorporate](#)^[27], ovvero eventuali file di modifica live incorporati;
- [tabella stringhe](#)^[26];
- [tabella stringhe compresse](#)^[27].

Nell'elenco dei blocchi, esiste sempre un blocco principale denominato MAIN ed eventualmente ulteriori blocchi (paragonabili a sub-routines di altri linguaggi di programmazione): ogni blocco contiene un [elenco di circuiti](#)^[14] che costituiscono il programma ladder effettivo: ogni circuito a sua volta è un elenco logicamente strutturato di [elementi](#)^[15].

NOTA: tutti i dati numerici memorizzati nel presente file sono da considerarsi in formato little endian (standard x86/x64).

Intestazione file

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

Ad inizio file (offset 0) è presente una intestazione generale (HEADER) che contiene le informazioni di accesso a tutte le restanti componenti e sotto-sezioni del file.

Name	Offset	Tipo	Descrizione
SIGN	+0	U32	Firma del file ('LADF').
VERSION	+4	U32	Versione del file (1.2.0).
GENERATOR	+8	U32	Nome e versione del generatore del file (offset relativo a tabella stringhe).

Name	Offset	Tipo	Descrizione
TYPE	+12	U16	Tipo di esecuzione: 0 = Esecuzione sincrona (SY). 1 = Esecuzione ad alta priorità (HP). 2 = Esecuzione a priorità normale (NP). 3 = Esecuzione a bassa priorità (LP).
FREQUENCY	+14	U16	Frequenza di esecuzione in [hz], da 1hz a 3000hz, default 200hz .
FLAGS	+16	U32	Impostazioni generali: 0x00000001 = il file contiene errori di verifica formale.
BUILDEAR	+20	U16	Anno ultima generazione (4 digit).
BUILDMONTH	+22	U8	Mese ultima generazione (1-12).
BUILDDAY	+23	U8	Giorno ultima generazione (1-31).
BUILDHOUR	+24	U8	Ultima ora generazione (0-23).
BUILDMINUTE	+25	U8	Minuto ultima compilazione (0-59).
BUILDSECOND	+26	U8	Secondo ultima compilazione (0-59).
TARGETID	+27	U8	ID destinazione runtime: 0x00 = Firmware RTE (PowerPC G2) 0x01 = Firmware RTE (PowerPC P2020)
VID	+28	U32	ID versione del file.
PLVID	+32	U32	ID versione del file pre modifiche live (se presenti, altrimenti vale 0).
	+36	U8[92]	(riservati)

Name	Offset	Tipo	Descrizione
BLOCKSOFF	+128	U32	Offset della tabella dei blocchi ^[13] (relativo al file).
BLOCKSCNT	+132	U32	N. di elementi della tabella dei blocchi ^[13] .
STRINGSOFF	+136	U32	Offset della tabella stringhe ^[26] (relativo al file).
STRINGSSIZE	+140	U32	Dimensione della tabella stringhe ^[26] .
CSTRINGSOFF	+144	U32	Offset della tabella stringhe compresse ^[27] (relativo al file).
CSTRINGSSIZE	+148	U32	Dimensione della tabella stringhe compresse ^[27] .
CSTRINGSTYPE	+152	U8	Tipo di compressione usata per la tabella stringhe compresse ^[27] : 0 = Nessuna compressione. 1 = Compressione standard Qt 3.3.x.
EMBLIVESOFF	+153	U32	Offset della tabella modifiche live incorporate (relativo al file) .
EMBLIVESCNT	+157	U32	N. di elementi della tabella modifiche live incorporate.
	+161	U8[3]	(riservati)
STRUCTSOFF	+164	U32	Offset della tabella strutture ^[10] (relativo al file).
STRUCTSCNT	+168	U32	N. di elementi della tabella strutture ^[10] .
VARSOFF	+172	U32	Offset della tabella variabili ^[11] (relativo al file).
VARSCNT	+176	U32	N. di elementi della tabella variabili ^[11] .
VARSINITOFF	+180	U32	Offset della tabella inizziatori

Name	Offset	Tipo	Descrizione
			variabile ^[26] (relativo al file) .
VARINITCNT	+184	U32	N. di elementi della tabella inizializzatori variabile ^[26] .
LASTBLOCKID	+188	U32	Ultimo ID blocco utilizzato.
LASTRUNGID	+192	U32	Ultimo ID circuito utilizzato.
LASTSTRUCTID	+196	U32	Ultimo ID struttura utilizzato.
LASTVARID	+200	U32	Ultimo ID variabile utilizzato.
	+204	U8[308]	(riservati)

Tabella strutture

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

La tabella strutture (STRUCTS) è composta da un numero di definizioni (STRUCT) contigue, ognuna delle quali contiene tutte le informazioni su uno specifica struttura definita dall'utente: in questa tabella non sono presenti i [tipi dati predefiniti](#)^[28].

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[7].STRUCTSOFF e [HEADER](#)^[7].STRUCTSSIZE.

Definizione struttura

Una struttura (STRUCT) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
TYPEID	+0	U32	ID tipo struttura Nota: solo valori 0x00010000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
STATUS	+4	U16	(riservato, uso runtime)
FLAGS	+6	U16	Impostazioni della struttura: 0x0001 Campo TEXT non compresso (NCF0).
NAME	+8	U32	Offset nome della struttura (relativo a

Nome	Offset	Tipo	Descrizione
			stringhe ^[26]).
TEXT	+12	U32	Offset descrizione della struttura (relativo a stringhe compresse ^[27] , o stringhe ^[26] se NCF0).
VARSOFF	+16	U32	Offset della tabella variabili ^[11] (relativo al file) (#1)
VARSCNT	+20	U32	Dimensione tabella variabili ^[11]
SIZE	+24	U32	Dimensione della struttura [byte]
EXTRA	+28	U32	Offset informazioni extra (0=nessuna) (relativo al file)

(#1) La definizione di struttura usa una normale tabella di definizione variabili per definire i propri elementi. In questo caso gli ID delle variabili non hanno senso e dovrebbero essere sempre a 0.

Tabella variabili

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

La tabella variabili (VARS) è composta da un numero di definizioni (VAR) contigue, ognuna delle quali contiene tutte le informazioni su uno specifica variabile definita dall'utente: in questa tabella non sono presenti le [variabili predefinite](#)^[31].

La posizione (relativa al file) e il numero di elementi della tabella delle variabili di programma sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata

- dai campi [HEADER](#)^[7].VARSOFF e [HEADER](#)^[7].VARSSIZE se utilizzati in intestazione file (HEADER);
- dai campi [STRUCT](#)^[10].VARSOFF e [STRUCT](#)^[10].VARSCNT se utilizzati in definizione struttura utente (STRUCT);
- dai campi [LIVECMD](#)^[128].VARSOFF e [LIVECMD](#)^[128].VARSCNT se utilizzati in definizione comando live (LIVECMD).

Definizione variabile

Una variabile (VAR) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
VARID	+0	U32	ID variabile (0 se variabile in definizione struttura utente ^[10])

Nome	Offset	Tipo	Descrizione
			Nota: solo valori 0x00010000-0xFFFFFFFFE; il valore 0xFFFFFFFF è un ID a significato speciale; i valori 0x00000001-0x0000FFFF sono riservati per le variabili predefinite.
TYPEID	+4	U32	ID tipo variabile Nota: può essere un tipo predefinito ^[28] oppure definito dall'utente ^[10] .
STATUS	+8	U16	(riservato, uso runtime)
FLAGS	+10	U16	Impostazioni della variabile: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo BITTEXT non compresso (NCF1)
NAME	+12	U32	Offset nome della variabile (relativo a stringhe ^[26]).
TEXT	+16	U32	Offset descrizione della variabile (relativo a stringhe compresse ^[27] , o stringhe ^[26] se NCF0).
BITTEXT	+20	U32	Offset descrizioni dei bit variabili, ovvero sequenza di stringhe relative ad ogni singolo bit separate dal carattere ascii 0xFF (relativo a stringhe compresse ^[27] , o stringhe ^[26] se NCF0).
ARRAY0	+24	U32	N. di indici 1o livello (0=Nessun array) (#1) o indice 0 per Alias ^[29] .
ARRAY1	+28	U32	N. di indici 2o livello (0=Nessun array) (#1) o indice 1 per Alias ^[29] .
ARRAY2	+32	U32	N. di indici 3o livello (0=Nessun array) (#1) o indice 2 per Alias ^[29] .
ALIASID	+36	U16	ID tipo alias ^[30]

Nome	Offset	Tipo	Descrizione
ALIASBIT	+38	U16	N. di bit per alias (dipende dal tipo di ID alias)
SIZE	+40	U32	Dimensione della variabile. Se Alias ^[29] , la dimensione è 0.
EXTRA	+44	U32	Offset informazioni extra (0=nessuna) (relativo al file)

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW l'indice massimo comunque va considerato solo a 16bit (ovvero massimo indice 0xFFFF).

Tabella blocchi

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

La tabella blocchi (BLOCKS) è composta da un numero di definizioni di blocco (BLOCK) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico blocco e del relativo contenuto su file.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[7].BLOCKSOFF e [HEADER](#)^[7].BLOCKSCNT.

Definizione blocco

Un blocco (BLOCK) è composto dai seguenti dati:

Nome	Offset	Tipo	Descrizione
BLOCKID	+0	U32	ID univoco del blocco. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
STATUS	+4	U16	(riservato, uso runtime)
FLAGS	+6	U16	Impostazioni del blocco: 0x0001 = Blocco principale (ISMAIN). 0x0002 = Campo TEXT non compresso (NCF0). 0x0004 = Campo COMMENT non compresso (NCF1).
NAME	+8	U32	Offset nome del blocco (relativo a stringhe ^[28]). Nota: se flag ISMAIN specificato, il nome viene deve essere "main".

Nome	Offset	Tipo	Descrizione
TEXT	+12	U32	Offset descrizione del blocco (relativo a stringhe compresse ^[27] , o stringhe ^[26] se NCF0).
COMMENT	+16	U32	Offset commenti per il blocco (relativo a stringhe compresse ^[27] , o stringhe ^[26] se NCF1).
RUNGSOFF	+20	U32	Offset della tabella circuiti del blocco ^[14] (relativo a file).
RUNGSCNT	+24	U32	N. di elementi della tabella circuiti del blocco ^[14] .
	+28	B[36]	(riservati)

Tabella circuiti del blocco

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

La tabella circuiti del blocco (RUNGS) è composta da un numero di definizioni di circuito (RUNG) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico circuito e del relativo contenuto.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [BLOCK](#)^[13].RUNGSOFF e [BLOCK](#)^[13].RUNGSCNT.

Definizione circuito del blocco

Un circuito del blocco (RUNG) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
RUNGID	+0	U32	ID univoco del circuito. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
STATUS	+4	U16	(riservato, uso runtime)
FLAGS	+6	U16	Impostazioni del circuito: 0x0001 Campo TEXT non compresso (NCF0).
REFID	+8	U32	ID circuito di riferimento (0=nessuno).
ITEMSOFF	+12	U32	Offset della tabella elementi del circuito ^[15] (relativo al file).

Nome	Offset	Tipo	Descrizione
ITEMSSIZE	+16	U32	Dimensione della tabella elementi del circuito ^[15] .
TEXT	+20	U32	Offset descrizione del circuito (relativo a stringhe compresse ^[27] , o stringhe ^[26] se NCF1).
EXTRA	+24	U32	Offset informazioni extra (0=nessuna) (relativo al file).

Tabella elementi circuito

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

La tabella elementi del circuito (ITEMS) è composta da un numero di definizioni di circuito (ITEM) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico elemento.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata:

- dai campi [RUNG](#)^[14].ITEMSOFF e [RUNG](#)^[14].ITEMSSIZE, se utilizzato in circuito del blocco (RUNG).
- dai campi [LIVECMD](#)^[128].ITEMSOFF e [LIVECMD](#)^[128].ITEMSSIZE, se utilizzato in comandi modifiche live (LIVECMD).

Definizione elemento del circuito

Gli elementi del circuito (ITEM) hanno una parte iniziale comune, quindi un'area dati specifica per ogni tipo di elemento. La parte comune è la seguente:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	Codice tipo elemento.
SIZE	+2	U16	Dimensione completa dell'elemento (tutti i campi).
DATA	+4	...	Dati specifici (opzionali).

Gli elementi previsti sono i seguenti:

Simbolico	ID	Tipo	Dati	Descrizione
BB	0x0001			Inizio ramo.
NL	0x0002			Prossimo livello ramo . Note: deve sempre essere

Simbolico	ID	Tipo	Dati	Descrizione
				compreso tra un BB e un EB.
EB	0x0003			Fine ramo . Note: deve sempre corrispondere ad un BB.
OC	0x0004		U32=Destinazione (#2): solo tipi BOOL ₂₈	Contatto aperto.
CC	0x0005		U32=Destinazione (#2): solo tipi BOOL ₂₈	Contatto chiuso.
RE	0x0006		U32=Destinazione (#2): solo tipi EDGE ₂₉	Fronte di salita.
FE	0x0007		U32=Destinazione (#2): solo tipi EDGE ₂₉	Fronte di discesa.
AF	0x0008			Sempre falso.
OUT	0x0009	F	U32=Destinazione (#2): solo tipi BOOL ₂₈	Output.
OL	0x000A	F	U32=Destinazione (#2): solo tipi BOOL ₂₈	Output latch (set).
OUL	0x000B	F	U32=Destinazione (#2): solo tipi BOOL ₂₈	Output unlatch (reset).
NOUT	0x000C	F	U32=Destinazione (#2): solo tipi BOOL ₂₈	Output negato
CLR	0x000D	F	U32=Destinazione (#2): solo tipi TIMER ₂₉ e COUNTER ₂₉	Azzeramento (clear).
TP	0x000E	F	U32=Destinazione (#2): solo tipi TIMER ₂₉ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer pulse.

Simbolico	ID	Tipo	Dati	Descrizione
TON	0x000F	F	U32=Destinazione (#2): solo tipi TIMER ₂₉ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer on delay.
TOFF	0x0010	F	U32=Destinazione (#2): solo tipi TIMER ₂₉ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer off delay.
TSQ	0x0011	F	U32=Destinazione (#2): solo tipi TIMER ₂₉ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer onda quadra.
TINT	0x0012	F	U32=Destinazione (#2): solo tipi TIMER ₂₉ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer integrale.
CU	0x0013	F	U32=Destinazione (#2): solo tipi COUNTER ₂₉ U32=Valore di preset (#4)	Conteggio up.
CD	0x0014	F	U32=Destinazione (#2): solo tipi COUNTER ₂₉ U32=Valore di preset (#4)	Conteggio down.
CUR	0x0015	F	U32=Destinazione (#2): solo tipi COUNTER ₂₉ U32=Valore di preset (#4)	Conteggio up rotativo.
CDR	0x0016	F	U32=Destinazione (#2): solo tipi	Conteggio down rotativo.

Simbolico	ID	Tipo	Dati	Descrizione
			COUNTER ²⁹ U32=Valore di preset (#4)	
LD	0x0017	F	U32=Destinazione (#2): solo tipi COUNTER ²⁹	Preset su contatore.
MATH	0x0018	F	U32=Espressione (#5) U32=Destinazione (#2) U8[]=Dati espressione matematica ²³	Blocco matematico.
EQ	0x0019		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto uguaglianza (A == B).
NE	0x001A		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto diverso (A != B).
GT	0x001B		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto maggiore (A > B).
GE	0x001C		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto maggiore/uguale (A >= B).
LT	0x001D		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto minore (A < B).
LE	0x001E		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto minore/uguale (A <= B).
COPY	0x001F	F	U32=Sorgente (#2) U32=Destinazione (#2) U32=N. Elementi (#2)	Copia dati da sorgente a destinazione, per il n. di elementi specificato.

Simbolico	ID	Tipo	Dati	Descrizione
CALL	0x0020	F	U32 ID blocco da chiamare	Chiamata di un blocco (sub-routine).
RET	0x0021	F		Ritorno da un blocco all'interno.
ADD	0x0022	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Somma (A + B).
SUB	0x0023	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Sottrazione (A - B).
MUL	0x0024	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Moltiplicazione (A * B).
DIV	0x0025	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Divisione (A / B).
AND	0x0026	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	AND binario (A & B).
OR	0x0027	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	OR binario (A B).
NOT	0x0028	F	U32=Sorgente A (#2)	NOT binario ($\sim A$).

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Destinazione (#2)	
XOR	0x0029	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	XOR binario ($A \wedge B$).
SQRT	0x002A	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Radice quadrata di A.
MOD	0x002B	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Modulo ($A \bmod B$).
POW	0x002C	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Elevamento a potenza ($A \text{ pow } B$).
SIN	0x002D	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Seno di A.
COS	0x002E	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Coseno di A.
TAN	0x002F	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Tangente di A.
ASIN	0x0030	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Arcoseno di A.
ACOS	0x0031	F	U32=Sorgente A (#2)	Arcocoseno di A.

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Destinazione (#2)	
ATAN	0x0032	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Arcotangente di A.
ATAN2	0x0033	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Arcotangente di A, B.
LOG	0x0034	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Logaritmo base 10 di A.
LN	0x0035	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Logaritmo base E di A.
ABS	0x0036	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Valore assoluto di A
SIGN	0x0037	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Segno di A.
ROUND	0x0038	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Arrotondamento all'intero più vicino ad A.
MIN	0x0039	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Minimo tra A e B.
MAX	0x003A	F	U32=Sorgente A (#2) U32=Sorgente B (#2)	Massimo tra A e B.

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Destinazione (#2)	
RANGE	0x003B	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Sorgente C (#2) U32=Destinazione (#2): solo tipi BOOL ₂₈	Verifica se A contenuto nel range B-C.
LIMIT	0x003C	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Sorgente C (#2) U32=Destinazione (#2)	Limita A al range B-C.
ROR	0x003D	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Rotazione bit a destra (A >> B).
ROL	0x003E	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Rotazione bit a sinistra (A << B).
SHR	0x003F	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Shift bit a destra (A >> B).
SHL	0x0040	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#2)	Shift bit a sinistra (A << B).

Simbolico	ID	Tipo	Dati	Descrizione
MOV	0x0041	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Assegna A.
TODEG	0x0042	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Conversione di A da RAD a DEG.
TORAD	0x0043	F	U32=Sorgente A (#2) U32=Destinazione (#2)	Conversione di A da DEG a RAD.
NEG	0x0044			Negazione stato logico.
NOP	0x0045	F		Nessuna operazione.
EVAL	0x0046		U32=Espressione (#5) U8[]= Dati espressione matematica ^[23]	Valutazione espressione (booleana)

(#1) Variabile a bit locale (tipo I32 o U32) con indirizzamento diretto (eventuali indici e numero di bit costanti).

(#2) Per sorgente/destinazione si intende nome di una variabile oppure la rappresentazione ascii di un valore numerico (solo sorgente), ovvero un offset alla [tabella stringhe](#)^[26].

(#4) Per valore (o valore di preset) si intende un valore numerico in formato stringa: stringa vuota (o offset 0) si intende nessun valore assegnato.

(#5) Espressione intesa come stringa con offset alla [tabella stringhe](#)^[26].

Dati espressione matematica

I dati dell'espressione matematica (elemento MATH) sono una serie di operazioni contigue (ordinati in polacca inversa) con la seguente intestazione comune:

Nome	Offset	Tipo	Descrizione
CODE	+0	U8	Codice operazione.
DATA	+1	...	Dati specifici (opzionali).

Le operazioni previste sono le seguenti:

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
PUSH	0x01	DBL=valore	--	DBL	Push valore in stack.

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
PUSHV	0x02	U32=Sorgente (#1)	--	DBL	Push valore variabile in stack.
ADD	0x03		DBL0,DBL1	DBL	Somma.
SUB	0x04		DBL0,DBL1	DBL	Sottrazione (#2).
MUL	0x05		DBL0,DBL1	DBL	Moltiplicazione (#2).
DIV	0x06		DBL0,DBL1	DBL	Divisione (#2).
MOD	0x07		DBL	DBL	Modulo (#3).
ABS	0x08		DBL	DBL	Valore assoluto (#3).
SIN	0x09		DBL	DBL	Seno.
COS	0x0A		DBL	DBL	Coseno.
TAN	0x0B		DBL	DBL	Tangente.
ASIN	0x0C		DBL	DBL	Arcoseno.
ACOS	0x0D		DBL	DBL	Arcocoseno.
ATAN	0x0E		DBL	DBL	Arcotangente.
POW	0x0F		DBL	DBL	Esponente.
LOG10	0x10		DBL	DBL	Logaritmo base 10.
LOG	0x11		DBL	DBL	Logaritmo naturale.
R_AND	0x12		U32	U32	AND binario.
R_OR	0x13		U32	U32	OR binario.
R_NOT	0x14		U32	U32	NOT binario.
R_XOR	0x15		U32	U32	XOR binario.
POPV	0x16	U32=Destinazione (#1)	DBL	--	Pop valore da stack a variabile.
R_ROR	0x17		U32	U32	Rotazione bit a destra.
R_ROL	0x18		U32	U32	Rotazione bit a sinistra.
R_SHR	0x19		U32	U32	Shift bit a destra.

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
R_SHL	0x1A		U32	U32	Shift bit a sinistra.
TOBIN	0x1B		U32	U32	Conversione da BCD a binario.
TOBCD	0x1C		U32	U32	Conversione da binario a BCD.
TODEG	0x1D		DBL	DBL	Conversione da radianti a gradi.
TORAD	0x1E		DBL	DBL	Conversione da gradi a radianti.
ATAN2	0x1F		DBL0,DBL1	DBL	Arcotangente con due parametri.
PUSHVB	0x20	U32=Sorgente (#1)	DBL	--	Push valore variabile a bit in stack.
POPVB	0x21	U32=Destinazione (#1)	DBL	--	Pop valore da stack a variabile bit.
POPL	0x22		BOOL	--	Pop valore logico (0=falso, altrimenti vero).
	0x23				(riservato)
EQ	0x24		DBL0,DBL1	BOOL	Confronto: uguale.
NE	0x25		DBL0,DBL1	BOOL	Confronto: differente.
LT	0x26		DBL0,DBL1	BOOL	Confronto: minore.
LE	0x27		DBL0,DBL1	BOOL	Confronto: minore o uguale.
GT	0x28		DBL0,DBL1	BOOL	Confronto: maggiore.
GE	0x29		DBL0,DBL1	BOOL	Confronto: maggiore o uguale.

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
AND	0x2A		BOOL,BOOL	BOOL	AND logico.
OR	0x2B		BOOL,BOOL	BOOL	OR logico.
NOT	0x2C		BOOL	BOOL	NOT logico.
XOR	0x2D		BOOL,BOOL	BOOL	XOR logico.
CHS	0x2E		DBL	DBL	Cambio segno.
SHR	0x2F		U32	U32	Shift logico a destra.
SHL	0x30		U32	U32	Shift logico a sinistra

(#1) Per sorgente/destinazione si intende nome di una variabile, ovvero un offset alla [tabella stringhe](#)^[26].

Tabella inizializzatori variabile

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

La tabella inizializzatori variabile (VARSINIT) è composta da un numero di definizioni (VARINIT) contigue, ognuna delle quali contiene tutte le informazioni su uno specifica variabile inizializzazione di variabile, detto altresì valore iniziale.

La posizione (relativa al file) e il numero di elementi della tabella delle variabili di programma sono specificate rispettivamente dai campi [HEADER](#)^[7].VARSINITOFF e [HEADER](#)^[7].VARSINITSIZE.

Definizione iniziatore variabile

Un iniziatore variabile (VARINIT) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
NAME	+0	U32	Offset nome della variabile (relativo a stringhe ^[26]).
VALUE	+4	U32	Offset valore della variabile (relativo a stringhe ^[26]).

Tabella stringhe

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

La tabella stringhe (STIRNGS) è composta da un certo numero di stringhe ascii consecutive, ognuna terminata il carattere ascii \0 (zero) e a dimensione variabile. La tabella deve contenere almeno un carattere e deve essere \0, di modo che le stringhe con offset 0 puntino effettivamente ad una stringa nulla.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata:

- dai campi [HEADER](#)^[7].STRINGSOFF e [HEADER](#)^[7].STRINGSSIZE, se usata in file ladder/diagram.
- dai campi [LIVEHEADER](#)^[127].STRINGSOFF e [LIVEHEADER](#)^[127].STRINGSSIZE, se usata in modifiche live.

Tabella stringhe compresse

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

La tabella stringhe compresse (CSTRINGS) è composta da un certo numero di stringhe ascii consecutive, ognuna terminata il carattere ascii \0 (zero) e a dimensione variabile.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente dai campi [HEADER](#)^[7].STRINGSOFF e [HEADER](#)^[7].STRINGSSIZE: lo stato di compressione, che è opzionale, è invece specificato dal campo [HEADER](#)^[7].CSTRINGSTYPE.

Annotazioni:

- in caso di compressione [HEADER](#)^[7].CSTRINGSTYPE = 1, utilizzare le funzioni standard di libreria (QT >= 3.3.4) `qUncompress()` per decomprimere la tabella in modo corretto.
- gli offset che fanno riferimento a questa tabella si intendono sempre riferiti alla tabella non compressa.

Tabella modifiche live

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

La tabella di modifiche live incorporate (EMBLIVES) è composta da un numero di definizioni (EMBLIVE) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico insieme di modifiche live al file ladder stesso: ogni insieme è in realtà un file di [modifiche live](#)^[127] indipendente ma incorporato nel file ladder stesso.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[7].EMBLIVESOFF e [HEADER](#)^[7].EMBLIVESCNT.

In fase di caricamento, i vari file di modifiche live vengono applicati nell'esatto ordine in cui compaiono. L'applicazione effettiva delle modifiche è regolata dall'apposito flag in [EMBLIVE](#)^[27].FLAGS (valore 0x0001): solo l'ultimo file di modifica live può avere eventualmente questo flag a falso, indicando delle modifiche live non ancora applicate.

Definizione modifica live incorporata

Una modifica live incorporata (EMBLIVE) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
FILEOFF	+0	U32	Offset del file di modifiche live ^[127] (offset relativo al file ladder).
FILESIZE	+4	U32	Dimensione del file di modifiche live ^[127] .
FLAGS	+8	U16	Impostazioni del file:

Nome	Offset	Tipo	Descrizione
			0x0001 = Modifiche effettive da applicare.
	+10	B[22]	(riservati)

Tipi di dati predefiniti

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

I tipi di dati vengono identificati in base ad un codice ID univoco, siano essi predefiniti oppure definiti dall'utente: gli ID vengono suddivisi nel seguente modo:

Range	Descrizione
0x00000000-0x0000FFFF	Tipi dati predefiniti
0x00010000-0xFFFFFFFFE	Tipi dati definiti dall'utente
0xFFFFFFFF	Riservato

ATTENZIONE: i tipi di dati predefiniti non vengono mai salvati nel file .LAD, ma si considerano sempre esistenti.

I tipi di dati predefiniti sono i seguenti:

ID	Simbolico	Descrizione	RTE (0x00)
0x00000000	(riservato)	Tipo dato non valido	
0x00000001	I8	Intero 8bit con segno	1
0x00000002	U8	Intero 8bit senza segno	1
0x00000003	I16	Intero 16bit con segno	2
0x00000004	U16	Intero 16bit senza segno	2
0x00000005	I32	Intero 32bit con segno	4
0x00000006	U32	Intero 32bit senza segno	4
0x00000007	I64	Intero 64bit con segno	
0x00000008	U64	Intero 64bit senza segno	
0x00000009	STRING	Stringa (0 terminated)	...
0x0000000A	FLOAT	Floating point 32bit	4
0x0000000B	REAL	Floating point 64bit (double)	8
0x0000000C	BOOL	Booleano	4 (#1)

ID	Simbolico	Descrizione	RTE (0x00)
0x0000000D	TIMER	Struttura predefinita timer ^[29]	32
0x0000000E	COUNTER	Struttura predefinita contatore ^[29]	48
0x0000000F	ALIAS	Alias (o literally) ^[29]	0
0x00000010	EDGE	Storage speciale per elementi fronte di salita (RE) e fronte di discesa (FE)	4
0x00000011	POWERSET	Variabile logica "Power Set"	4 (#2)
0x00000012	AXESGROUP	Variabile logica "Gruppo assi"	4 (#2)
0x00000013	PATH	Variabile logica "Path"	4 (#2)
0x00000014	POINT_L	Variabile logica "Punto di libreria"	4 (#2)

(#1) il tipo BOOL in RTE viene rappresentato, per motivi di performance, come I32, mentre da RDE3 viene visto come U8.

(#2) Le variabili logiche in RTE sono sempre rappresentate come un puntatore in memoria (4bytes), anche qualora fossero incorporate in strutture definite dall'utente. Per ulteriore informazioni sulla composizione (elementi) di tali variabili, consultare la documentazione tecnica di RTE).

La rappresentazione e la dimensione dei tipi di dati dipendono direttamente dall'ID destinazione runtime ([HEADER](#)^[7].TARGETID). Le dimensioni dei tipi dati sono elencati nelle colonne RTE e sono espressi in bytes.

Struttura predefinita timer

La struttura predefinita TIMER è composta dai seguenti campi:

Tipo	Nome	Descrizione	Letture	Scrittura
BOOL	Q	Esito del timer	X	
REAL	ET	Valore attuale [ms]	X	X (#1)
REAL	PT	Valore di preset [ms]	X	X
REAL	PT2	Valore di preset 2 [ms]	X	X
BOOL	EN	Abilitazione timer	X	

(#1) Il campo ET può essere scritto solo quando il timer non è ancora stato assegnato.

Struttura predefinita contatore

La struttura predefinita COUNTER è composta dai seguenti campi:

Tipo	Nome	Descrizione	Letture	Scrittura
BOOL	QU	Esito del conteggio rispetto a preset	X	
BOOL	QD	Esito del conteggio rispetto a 0	X	
I32	CV	Valore attuale di conteggio	X	X (#1)
I32	PV	Valore di preset	X	X
BOOL	CU	Stato del segnale "count up"	X	
BOOL	CD	Stato del segnale "count down"	X	

(#1) Il campo CV può essere scritto solo quando il contatore non è ancora stato assegnato.

Alias (o literally)

L'elenco dei tipi dati utilizzabili come alias sono i seguenti:

ID	Nome	Tipo	Descrizione
0x0001	INP(n)	BOOL	Canale input
0x0002	INP_W(n)	U16	Input word 16bit
0x0003	INP_W(n).b	BOOL	Input word 16bit, accesso a bit
0x0004	INP_DW(n)	U32	Input word 32bit
0x0005	INP_DW(n).b	BOOL	Input word 32bit, accesso a bit
0x0006	OUT	BOOL	Canale output
0x0007	OUT_W(n)	U16	Output word 16bit
0x0008	OUT_W(n).b	BOOL	Output word 16bit, accesso a bit
0x0009	OUT_DW(n)	U32	Output word 32bit
0x000A	OUT_DW(n).b	BOOL	Output word 32bit, accesso a bit
0x000B	R(n)	I32	Registro globale intero (volatile)
0x000C	R(n).b	BOOL	Registro globale intero (volatile), accesso a bit

ID	Nome	Tipo	Descrizione
0x000D	NVR(n)	I32	Registro globale intero (non volatile)
0x000E	NVR(n).b	BOOL	Registro globale intero (non volatile), accesso a bit
0x000F	RR(n)	REAL	Registro globale reale (volatile)
0x0010	NVRR(n)	REAL	Registro globale reale (non volatile)
0x0011	AM(n)	U32	Maschera allarmi
0x0012	AM(n).b	U32	Maschera allarmi, accesso a bit

Annotazioni:

- se la variabile è di tipo array, l'indice di array (n) può essere o una costante o un variabile non indicizzata.
- se la variabile prevede l'accesso a bit, .b può essere una costante o una variabile e deve essere nel range compatibile la grandezza della variabile di riferimento.

Variabili predefinite

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato del file (versione 1.2).

Le variabili vengono identificate in base ad un codice ID univoco, siano esse predefinite o definite dall'utente: gli ID vengono suddivisi nel seguente modo:

Range	Descrizione
0x00000000-0x0000FFFF	Variabile predefinita
0x0001000-0xFFFFFFFFE	Variabile definita dall'utente
0xFFFFFFFF	Riservato

ATTENZIONE: le variabili predefinite non vengono mai salvate nel file .LAD, ma si considerano sempre esistenti.

Le variabili predefinite sono i seguenti:

ID	Nome	Tipo	Descrizione
0x00000001	INP(n)	BOOL	Canale input
0x00000002	INP_W(n)	U16	Input word 16bit
0x00000003	INP_DW(n)	U32	Input word 32bit
0x00000004	OUT	BOOL	Canale output

ID	Nome	Tipo	Descrizione
0x00000005	OUT_W(n)	U16	Output word 16bit
0x00000006	OUT_DW(n)	U32	Output word 32bit
0x00000007	R(n)	I32	Registro globale intero (volatile)
0x00000008	NVR(n)	I32	Registro globale intero (non volatile)
0x00000009	RR(n)	REAL	Registro globale reale (volatile)
0x0000000A	NVRR(n)	REAL	Registro globale reale (non volatile)
0x0000000B	AM(n)	U32	Maschera allarmi

Annotazioni:

- se la variabile è di tipo array, l'indice di array (n) può essere o una costante o un variabile non indicizzata.
- RDE3 crea ulteriori variabili predefinite con assegnazione degli ID in modo del tutto arbitrario, ma maggiore di 0x00000100: RTE riconosce tali variabili in base al nome e non in base a tale ID.

Formato file (versione 1.3)

Formato file (versione 1.3)

Un programma in linguaggio di programmazione ladder/diagram è un singolo file (estensione .LAD) contenente sia le informazioni per l'editing (sorgente) che le informazioni per l'esecuzione (runtime).

Un programma ladder è composto principalmente da una [intestazione](#)^[33], quindi da varie sezioni tra cui le principali sono:

- [tabella strutture](#)^[35], ovvero i tipi dati definiti dall'utente.
- [tabella variabili locali](#)^[37], ovvero le variabili generali del programma.
- [tabella classi OB](#)^[53], ovvero le classi object block a cui il programma fa riferimento.
- [tabella blocchi](#)^[40], ovvero le sub-routines in altri linguaggi di programmazione;
- [tabella modifiche live incorporate](#)^[58], ovvero eventuali file di modifica live incorporati;
- [tabella stringhe](#)^[57];
- [tabella stringhe compresse](#)^[57].

Nell'elenco dei blocchi, esiste sempre un blocco principale detto MAIN ed eventualmente ulteriori blocchi (paragonabili a sub-routines di altri linguaggi di programmazione): ogni blocco contiene un [elenco di circuiti](#)^[41] che costituiscono il programma ladder effettivo: ogni circuito a sua volta è un elenco logicamente strutturato di [elementi](#)^[42].

NOTA: tutti i dati numerici memorizzati nel presente file sono da considerarsi in formato little indian (standard x86).

Intestazione file

Ad inizio file (offset 0) è presente una intestazione generale (HEADER) che contiene le informazioni di accesso a tutte le restanti componenti e sotto-sezioni del file.

Nome	Offset	Tipo	Description
SIGN	+0	U32	Firma del file ('LADF').
VERSION	+4	U32	Versione del file (1.3.1).
GENERATOR	+8	U32	Nome e versione del generatore del file (offset relativo a tabella stringhe).
TYPE	+12	U16	Tipo di esecuzione: 0 = Esecuzione sincrona (SY). 1 = Esecuzione ad alta priorità (HP). 2 = Esecuzione a priorità normale (NP). 3 = Esecuzione a bassa priorità (LP).
FREQUENCY	+14	U16	Frequenza di esecuzione in [hz], da 1hz a 3000hz, default 200hz .
FLAGS	+16	U32	Impostazioni generali: 0x00000001 = Il file contiene errori di verifica formale. 0x00000002 = Il file contiene riferimenti a file .OBB esterni (classi OB)
BUILDEAR	+20	U16	Anno ultima generazione (4 digit).
BUILDMONTH	+22	U8	Mese ultima generazione (1-12).
BUILDDAY	+23	U8	Giorno ultima generazione (1-31).
BUILDHOUR	+24	U8	Ultima ora generazione (0-23).
BUILDMINUTE	+25	U8	Minuto ultima compilazione (0-59).
BUILDSECOND	+26	U8	Secondo ultima compilazione (0-59).

Nome	Offset	Tipo	Description
TARGETID	+27	U8	ID destinazione runtime: 0x00 = Firmware RTE (PowerPC G2) 0x01 = Firmware RTE (PowerPC P2020)
VID	+28	U32	ID versione del file.
PLVID	+32	U32	ID versione del file pre modifiche live (se presenti, altrimenti vale 0).
IARECNT	+36	U32	N. di elementi tipo ARE presenti nel file.
IAFECNT	+40	U32	N. di elementi tipo AFE presenti nel file.
	+44	U8[84]	(riservati)
BLOCKSOFF	+128	U32	Offset della tabella dei blocchi ^[40] (relativo al file).
BLOCKSCNT	+132	U32	N. di elementi della tabella dei blocchi ^[40] .
STRINGSOFF	+136	U32	Offset della tabella stringhe ^[57] (relativo al file).
STRINGSSIZE	+140	U32	Dimensione della tabella stringhe ^[57] .
CSTRINGSOFF	+144	U32	Offset della tabella stringhe compresse ^[57] (relativo al file).
CSTRINGSSIZE	+148	U32	Dimensione della tabella stringhe compresse ^[57] .
CSTRINGSTYPE	+152	U8	Tipo di compressione usata per la tabella stringhe compresse ^[57] ∴ 0 = Nessuna compressione. 1 = Compressione standard Qt 3.3.x.
	+153	U8[3]	(riservati)
EMBLIVESOFF	+156	U32	Offset della tabella modifiche live

Nome	Offset	Tipo	Description
			incorporate ^[58] (relativo al file) .
EMBLIVESCNT	+160	U32	N. di elementi della tabella modifiche live incorporate ^[58] .
STRUCTSOFF	+164	U32	Offset della tabella strutture ^[35] (relativo al file).
STRUCTSCNT	+168	U32	N. di elementi della tabella strutture ^[35] .
VARSOFF	+172	U32	Offset della tabella variabili ^[37] (relativo al file).
VARSCNT	+176	U32	N. di elementi della tabella variabili ^[37] .
OBCLASSESOFF	+180	U32	Offset della tabella classi OB ^[53] (relativo al file)
OBCLASESCNT	+184	U32	N. di elementi della tabella classi OB ^[53] (relativo al file)
LASTBLOCKID	+188	U32	Ultimo ID blocco utilizzato.
LASTRUNGID	+192	U32	Ultimo ID circuito utilizzato.
LASTSTRUCTID	+196	U32	Ultimo ID struttura utilizzato.
LASTVARID	+200	U32	Ultimo ID variabile utilizzato.
LASTOBCLASSID	+204	U32	Ultimo ID classe OB utilizzato
	+208	U8[304]	(riservati)

Tabella strutture

La tabella strutture (STRUCTS) è composta da un numero di definizioni (STRUCT) contigue, ognuna delle quali contiene tutte le informazioni su uno specifica struttura definita dall'utente: in questa tabella non sono presenti i [tipi dati predefiniti](#)^[58].

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[33].STRUCTSOFF e [HEADER](#)^[33].STRUCTSCNT.

Definizione struttura

Una struttura (STRUCT) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID tipo struttura Nota: solo valori 0x00010000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati, uso runtime)
FLAGS	+6	U16	Impostazioni della struttura: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo MODIFIER non compresso (NCF1).
NAME	+8	U32	Offset nome della struttura (relativo a stringhe ^[57]).
TEXT	+12	U32	Offset descrizione della struttura (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF0).
VARSOFF	+16	U32	Offset della tabella variabili ^[37] (relativo al file) (#1)
VARSCNT	+20	U32	Dimensione tabella variabili ^[37]
SIZE	+24	U32	Dimensione della struttura [byte]
SRCID	+28	U32	Identificativo sorgente ^[62] della struttura
OBCID	+32	U32	ID classe OB di riferimento (0=nessuna)(#2,#3)
ATTRIB	+36	U32	Attributi della struttura: 0x00000001 Struttura riferita a classe OB (AREFOBC)(#2) 0x00000002 Struttura principale classe OB (AISOBC)(#2,#3)
MODIFIER	+40	U32	Offset modificatori della struttura (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF1).

Nome	Offset	Tipo	Descrizione
REFID	+44	U32	ID di riferimento tipo struttura (#4)

(#1) La definizione di struttura usa una normale tabella di definizione variabili per definire i propri elementi. In questo caso gli ID delle variabili non hanno senso e dovrebbero essere sempre a 0.

(#2) A pari valore di OBCID non 0, con attributo AREFOBC, può esistere una sola struttura con flag AISOBC, mentre posso essere 0 o più strutture senza attributo AISOBC.

(#3) Una struttura viene definita "orfana" qualora si trova nella condizione di attributi AREFOBC impostato e campo OBCID = 0: questa condizione capita quando aggiornando/rimuovendo una classe OB, vi sono ancora delle variabili (o altro) che vi fanno riferimento.

(#4) In caso di attributo AREFOBC il campo REFID contiene l'ID della struttura come definito del file OBB di origine (0 in case di attributo AISOBC).

Tabella variabili

La tabella variabili (VARS) è composta da un numero di definizioni (VAR) contigue, ognuna delle quali contiene tutte le informazioni su uno specifica variabile definita dall'utente: in questa tabella non sono presenti le [variabili predefinite](#)^[62].

La posizione (relativa al file) e il numero di elementi della tabella delle variabili di programma sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata

- dai campi [HEADER](#)^[33].VARSOFF e [HEADER](#)^[33].VARSCNT se utilizzati in intestazione file;
- dai campi [STRUCT](#)^[35].VARSOFF e [STRUCT](#)^[35].VARSCNT se utilizzati in definizione struttura;
- dai campi [LIVECMD](#)^[135].VARSOFF e [LIVECMD](#)^[135].VARSCNT se utilizzati in definizione comando.

Definizione variabile

Una variabile (VAR) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID variabile (#2) Nota: solo valori 0x00010000-0xFFFFFFFFE; il valore 0xFFFFFFFF è un ID a significato speciale; i valori 0x00000001-0x0000FFFF sono riservati per le variabili predefinite.
TYPEID	+4	U32	ID tipo variabile Nota: può essere un tipo predefinito ^[58] oppure definito dall'utente ^[35] .
	+8	U8[2]	(riservati, uso runtime)

Nome	Offset	Tipo	Descrizione
FLAGS	+10	U16	Impostazioni della variabile: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo BITTEXT non compresso (NCF1) 0x0004 Campo MODIFIER non compresso (NCF2)
NAME	+12	U32	Offset nome della variabile (relativo a stringhe ^[57]).
TEXT	+16	U32	Offset descrizione della variabile (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF0).
BITTEXT	+20	U32	Offset descrizioni dei bit variabili, ovvero sequenza di stringhe relative ad ogni singolo bit separate dal carattere ascii 0xFF (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF0). (#2)
DIM0	+24	U32	1a dimensione array (0=non specificata) (#1) o indice 0 per Alias ^[59] (TYPEID = 0x0000000F).
DIM1	+28	U32	2a dimensione array (0=non specificata) (#1) o indice 1 per Alias ^[59] (TYPEID = 0x0000000F).
DIM2	+32	U32	3a dimensione array (0=non specificata) (#1) o indice 2 per Alias ^[59] (TYPEID = 0x0000000F).
ALIASID	+36	U16	ID tipo alias ^[60] (#2)
ALIASBIT	+38	U16	N. di bit per alias (dipende dal tipo di ID alias)(#2)
SIZE	+40	U32	Dimensione della variabile. (#3)

Nome	Offset	Tipo	Descrizione
SRCID	+44	U32	Identificativo sorgente ^[62] della variabile. (#2, #4)
VALUE	+48	U32	Offset valore iniziale della variabile (relativo a stringhe ^[57] , 0=nessun valore).
ATTRIB	+52	U32	Attributi variabile: 0x00000001 La variabile può essere scritta (WRITE) 0x00000002 La variabile può essere letta (READ) 0x00000004 La variabile può essere forzata (FORCE) 0x00000008 La variabile può essere rilasciata (RELEASE) 0x00000010 La variabile è costante (CONST) 0x00000020 La variabile non ha accesso a bit (NOBITS)
MODIFIER	+56	U32	Offset modificatori della variabile (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF2).
DIM0L	+60	U32	Offset etichetta per 1a dimensione array (relativo a stringhe ^[57]). (#5)
DIM1L	+64	U32	Offset etichetta per 2a dimensione array (relativo a stringhe ^[57]). (#5)
DIM2L	+68	U32	Offset etichetta per 3a dimensione array (relativo a stringhe ^[57]). (#5)

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW la dimensione massima va comunque considerata solo a 16bit (ovvero massimo indice 0xFFFF).

(#2) Campo non utilizzato (valore 0) se utilizzata in definizione struttura

(#3) Campo non utilizzato (valore 0) se utilizzato definendo un Alias (TYPEID = 0x0000000F)

(#4) Campo non utilizzato (valore 0) se utilizzata in definizione comando live.

(#5) Per etichetta si intende nome di una variabile (a valore costante) oppure la rappresentazione ascii di un valore numerico, ovvero un offset alla [tabella stringhe](#)^[57].

Tabella blocchi

La tabella blocchi (BLOCKS) è composta da un numero di definizioni di blocco (BLOCK) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico blocco e del relativo contenuto su file.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[33].BLOCKSOFF e [HEADER](#)^[33].BLOCKSCNT.

Definizione blocco

Un blocco (**BLOCK**) è composto dai seguenti dati:

Nome	Offset	Tipo	Description
ID	+0	U32	ID univoco del blocco. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
STATUS	+4	U16	(riservato, uso runtime)
FLAGS	+6	U16	Impostazioni del blocco: 0x0001 = Blocco principale (ISMAIN). 0x0002 = Campo TEXT non compresso (NCF0). 0x0004 = Campo COMMENT non compresso (NCF1).
NAME	+8	U32	Offset nome del blocco (relativo a stringhe ^[57]). Nota: se flag ISMAIN specificato, il nome viene deve essere "main".
TEXT	+12	U32	Offset descrizione del blocco (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF0).
COMMENT	+16	U32	Offset commenti per il blocco (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF1).
RUNGSOFF	+20	U32	Offset della tabella circuiti del blocco ^[41] (relativo a file).
RUNGSCNT	+24	U32	N. di elementi della tabella circuiti del

Nome	Offset	Tipo	Description
			blocco ^[41] .
	+28	B[36]	(riservati)

Tabella circuiti del blocco

La tabella circuiti del blocco (RUNGS) è composta da un numero di definizioni di circuito (RUNG) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico circuito e del relativo contenuto.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [BLOCK](#)^[40].RUNGSOFF e [BLOCK](#)^[40].RUNGSCNT.

Definizione circuito del blocco

Un circuito del blocco (RUNG) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID univoco del circuito. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati, uso runtime)
FLAGS	+6	U16	Impostazioni del circuito: 0x0001 Campo TEXT non compresso (NCF0).
REFID	+8	U32	ID circuito di riferimento (0=nessuno).
ITEMSOFF	+12	U32	Offset della tabella elementi del circuito ^[42] (relativo al file).
ITEMSSIZE	+16	U32	Dimensione della tabella elementi del circuito ^[42] .
TEXT	+20	U32	Offset descrizione del circuito (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF1).
	+24	U8[4]	(riservati)

Tabella elementi circuito

La tabella elementi del circuito (ITEMS) è composta da un numero di definizioni di circuito (ITEM) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico elemento.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata:

- dai campi [RUNG](#)^[41].ITEMSOFF e [RUNG](#)^[41].ITEMSSIZE, se utilizzato in circuito del blocco (RUNG).
- dai campi [LIVECMD](#)^[135].ITEMSOFF e [LIVECMD](#)^[135].ITEMSSIZE, se utilizzato in comandi modifiche live (LIVECMD).

Definizione elemento del circuito

Gli elementi del circuito (ITEM) hanno una parte iniziale comune, quindi un'area dati specifica per ogni tipo di elemento. La parte comune è la seguente:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	Codice tipo elemento.
SIZE	+2	U16	Dimensione completa dell'elemento (tutti i campi).
DATA	+4	...	Dati specifici (opzionali).

Gli elementi previsti sono i seguenti:

Simbolico	ID	Tipo	Dati	Descrizione
BB	0x0001			Inizio ramo.
NL	0x0002			Prossimo livello ramo . Note: deve sempre essere compreso tra un BB e un EB.
EB	0x0003			Fine ramo . Note: deve sempre corrispondere ad un BB.
OC	0x0004		U32=Destinazione (#3): solo tipi BOOL ^[59]	Contatto aperto.
CC	0x0005		U32=Destinazione (#3): solo tipi BOOL ^[59]	Contatto chiuso.
RE	0x0006		U32=Destinazione (#3): solo tipi EDGE ^[59]	Fronte di salita.

Simbolico	ID	Tipo	Dati	Descrizione
FE	0x0007		U32=Destinazione (#3): solo tipi EDGE ⁵⁹	Fronte di discesa.
AF	0x0008			Sempre falso.
OUT	0x0009	F	U32=Destinazione (#3): solo tipi BOOL ⁵⁹	Output.
OL	0x000A	F	U32=Destinazione (#3): solo tipi BOOL ⁵⁹	Output latch (set).
OUL	0x000B	F	U32=Destinazione (#3): solo tipi BOOL ⁵⁹	Output unlatch (reset).
NOUT	0x000C	F	U32=Destinazione (#3): solo tipi BOOL ⁵⁹	Output negato
CLR	0x000D	F	U32=Destinazione (#3): solo tipi TIMER ⁵⁹ e COUNTER ⁵⁹	Azzeramento (clear).
TP	0x000E	F	U32=Destinazione (#3): solo tipi TIMER ⁵⁹ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer pulse.
TON	0x000F	F	U32=Destinazione (#3): solo tipi TIMER ⁵⁹ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer on delay.
TOFF	0x0010	F	U32=Destinazione (#3): solo tipi TIMER ⁵⁹ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer off delay.
TSQ	0x0011	F	U32=Destinazione (#3): solo tipi TIMER ⁵⁹	Abilita timer onda quadra.

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	
TINT	0x0012	F	U32=Destinazione (#3): solo tipi TIMER ₅₉ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer integrale.
CU	0x0013	F	U32=Destinazione (#3): solo tipi COUNTER ₅₉ U32=Valore di preset (#4)	Conteggio up.
CD	0x0014	F	U32=Destinazione (#3): solo tipi COUNTER ₅₉ U32=Valore di preset (#4)	Conteggio down.
CUR	0x0015	F	U32=Destinazione (#3): solo tipi COUNTER ₅₉ U32=Valore di preset (#4)	Conteggio up rotativo.
CDR	0x0016	F	U32=Destinazione (#3): solo tipi COUNTER ₅₉ U32=Valore di preset (#4)	Conteggio down rotativo.
LD	0x0017	F	U32=Destinazione (#3): solo tipi COUNTER ₅₉	Preset su contatore.
MATH	0x0018	F	U32=Espressione (#5) U32=Destinazione (#3) U8[]=Dati espressione matematica ₅₀	Blocco matematico.
EQ	0x0019		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto uguaglianza (A == B).

Simbolico	ID	Tipo	Dati	Descrizione
NE	0x001A		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto diverso ($A \neq B$).
GT	0x001B		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto maggiore ($A > B$).
GE	0x001C		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto maggiore/uguale ($A \geq B$).
LT	0x001D		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto minore ($A < B$).
LE	0x001E		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto minore/uguale ($A \leq B$).
COPY	0x001F	F	U32=Sorgente (#2) U32=Destinazione (#3) U32=N. Elementi (#2)	Copia dati da sorgente a destinazione, per il n. di elementi specificato.
CALL	0x0020	F	U32 ID blocco da chiamare	Chiamata di un blocco (sub-routine).
RET	0x0021	F		Ritorno da un blocco all'interno.
ADD	0x0022	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Somma ($A + B$).
SUB	0x0023	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Sottrazione ($A - B$).

Simbolico	ID	Tipo	Dati	Descrizione
MUL	0x0024	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Moltiplicazione (A * B).
DIV	0x0025	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Divisione (A / B).
AND	0x0026	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	AND binario (A & B).
OR	0x0027	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	OR binario (A B).
NOT	0x0028	F	U32=Sorgente A (#2) U32=Destinazione (#3)	NOT binario ($\sim A$).
XOR	0x0029	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	XOR binario (A ^ B).
SQRT	0x002A	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Radice quadrata di A.
MOD	0x002B	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Modulo (A mod B).

Simbolico	ID	Tipo	Dati	Descrizione
POW	0x002C	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Elevamento a potenza (A pow B).
SIN	0x002D	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Seno di A.
COS	0x002E	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Coseno di A.
TAN	0x002F	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Tangente di A.
ASIN	0x0030	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arcoseno di A.
ACOS	0x0031	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arcocoseno di A.
ATAN	0x0032	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arcotangente di A.
ATAN2	0x0033	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Arcotangente di A, B.
LOG	0x0034	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Logaritmo base 10 di A.
LN	0x0035	F	U32=Sorgente A (#2)	Logaritmo base E di A.

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Destinazione (#3)	
ABS	0x0036	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Valore assoluto di A
SIGN	0x0037	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Segno di A.
ROUND	0x0038	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arrotondamento all'intero più vicino ad A.
MIN	0x0039	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Minimo tra A e B.
MAX	0x003A	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Massimo tra A e B.
RANGE	0x003B	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Sorgente C (#2) U32=Destinazione (#3): solo tipi BOOL ⁵⁹	Verifica se A contenuto nel range B-C.
LIMIT	0x003C	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Sorgente C (#2) U32=Destinazione (#3)	Limita A al range B-C.

Simbolico	ID	Tipo	Dati	Descrizione
ROR	0x003D	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Rotazione bit a destra (A >> B).
ROL	0x003E	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Rotazione bit a sinistra (A << B).
SHR	0x003F	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Shift bit a destra (A >> B).
SHL	0x0040	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Shift bit a sinistra (A << B).
MOV	0x0041	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Assegna A.
TODEG	0x0042	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Conversione di A da RAD a DEG.
TORAD	0x0043	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Conversione di A da DEG a RAD.
NEG	0x0044			Negazione stato logico.
NOP	0x0045	F		Nessuna operazione.
EVAL	0x0046		U32=Espressione (#5)	Valutazione espressione (booleana).

Simbolico	ID	Tipo	Dati	Descrizione
			U8[]=Dati espressione matematica ^[57]	
OBCALL	0x0047	F	U32=Nome istanza (#6) U32=Nome completo metodo (#9) U32=Elenco nomi parametro (#7) U32=Elenco valori parametro (#7) U32=Risultato (#3) U8[]=Dati espressione matematica ^[57] (#8)	Chiamata a metodo di una istanza OB.
FT	0x0048			Esecuzione iniziale
ARE	0x0049			Fronte di salita (anonimo)
AFE	0x004A			Fronte di discesa (anonimo)

(#1) Variabile a bit locale (tipo I32 o U32) con indirizzamento diretto (eventuali indici e numero di bit costanti).

(#2) Per sorgente/destinazione si intende nome di una variabile oppure la rappresentazione ASCII di un valore numerico (solo sorgente), ovvero un offset alla [tabella stringhe](#)^[57].

(#3) Per destinazione si intende nome di una variabile con offset alla [tabella stringhe](#)^[57].

(#4) Per valore (o valore di preset) si intende un valore numerico in formato stringa: stringa vuota (o offset 0) si intende nessun valore assegnato.

(#5) Espressione intesa come stringa con offset alla [tabella stringhe](#)^[57].

(#6) Per nome istanza si intende nome di una variabile con offset alla [tabella stringhe](#)^[57], il cui TYPEID sia riferito ad una struttura utente con flag FISOBC.

(#7) Per elenco si intende una stringa (con offset alla [tabella stringhe](#)^[57]), composta da un certo numero di sotto-stringhe separate con carattere ASCII 0xFF: i due elenchi (nomi/valori) DEVONO avere sempre lo stesso n. di elementi o essere entrambe vuote (=0).

(#8) La chiamata ad un metodo di una istanza OB viene effettivamente tradotta come una espressione matematica, in cui vengono preparati in stack gli eventuali parametri (valori in ordine naturale), viene effettuata la chiamata al metodo e quindi ritrovato l'eventuale risultato da stack.

(#9) Nome completo del metodo (univoco), inteso come stringa con offset alla [tabella stringhe](#)^[57].

Dati espressione matematica

I dati dell'espressione matematica (usati dagli elementi MATH, COND e OBCALL), sono una serie di operazioni contigue (ordinati in polacca inversa) con la seguente intestazione comune:

Nome	Offset	Tipo	Descrizione
CODE	+0	U8	Codice operazione.
DATA	+1	...	Dati specifici (opzionali).

Le operazioni previste sono le seguenti:

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
PUSH	0x01	DBL=valore	--	DBL	Push valore in stack.
PUSHV	0x02	U32=Sorgente (#1)	--	DBL	Push valore variabile in stack.
ADD	0x03		DBL0,DBL1	DBL	Somma.
SUB	0x04		DBL0,DBL1	DBL	Sottrazione (#2).
MUL	0x05		DBL0,DBL1	DBL	Moltiplicazione (#2).
DIV	0x06		DBL0,DBL1	DBL	Divisione (#2).
MOD	0x07		DBL	DBL	Modulo (#3).
ABS	0x08		DBL	DBL	Valore assoluto (#3).
SIN	0x09		DBL	DBL	Seno.
COS	0x0A		DBL	DBL	Coseno.
TAN	0x0B		DBL	DBL	Tangente.
ASIN	0x0C		DBL	DBL	Arcoseno.
ACOS	0x0D		DBL	DBL	Arcocoseno.
ATAN	0x0E		DBL	DBL	Arcotangente.
POW	0x0F		DBL	DBL	Esponente.
LOG10	0x10		DBL	DBL	Logaritmo base 10.
LOG	0x11		DBL	DBL	Logaritmo naturale.
R_AND	0x12		U32	U32	AND binario.
R_OR	0x13		U32	U32	OR binario.
R_NOT	0x14		U32	U32	NOT binario.

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
R_XOR	0x15		U32	U32	XOR binario.
POPV	0x16	U32=Destinazione (#1)	DBL	--	Pop valore da stack a variabile.
R_ROR	0x17		U32	U32	Rotazione bit a destra.
R_ROL	0x18		U32	U32	Rotazione bit a sinistra.
R_SHR	0x19		U32	U32	Shift bit a destra.
R_SHL	0x1A		U32	U32	Shift bit a sinistra.
TOBIN	0x1B		U32	U32	Conversione da BCD a binario.
TOBCD	0x1C		U32	U32	Conversione da binario a BCD.
TODEG	0x1D		DBL	DBL	Conversione da radianti a gradi.
TORAD	0x1E		DBL	DBL	Conversione da gradi a radianti.
ATAN2	0x1F		DBL0,DBL1	DBL	Arcotangente con due parametri.
PUSHVB	0x20	U32=Sorgente (#1)	DBL	--	Push valore variabile a bit in stack.
POPVB	0x21	U32=Destinazione (#1)	DBL	--	Pop valore da stack a variabile bit.
POPL	0x22		BOOL	--	Pop valore logico (0=falso, altrimenti vero).
	0x23				(riservato)
EQ	0x24		DBL0,DBL1	BOOL	Confronto: uguale.
NE	0x25		DBL0,DBL1	BOOL	Confronto: differente.

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
LT	0x26		DBL0,DBL1	BOOL	Confronto: minore.
LE	0x27		DBL0,DBL1	BOOL	Confronto: minore o uguale.
GT	0x28		DBL0,DBL1	BOOL	Confronto: maggiore.
GE	0x29		DBL0,DBL1	BOOL	Confronto: maggiore o uguale.
AND	0x2A		BOOL,BOOL	BOOL	AND logico.
OR	0x2B		BOOL,BOOL	BOOL	OR logico.
NOT	0x2C		BOOL	BOOL	NOT logico.
XOR	0x2D		BOOL,BOOL	BOOL	XOR logico.
CHS	0x2E		DBL	DBL	Cambio segno.
SHR	0x2F		U32	U32	Shift logico a destra.
SHL	0x30		U32	U32	Shift logico a sinistra
OBCALL	0x31	U32=Metodo (#2)	?	?	Chiamata diretta a metodo per istanza OB
POP	0x32		*	--	Pop valore (skip)
PUSHP	0x33	U32=Sorgente (#1)	U32	U32	Push puntatore a variabile

(#1) Per sorgente/destinazione si intende nome di una variabile, ovvero un offset alla [tabella stringhe](#)^[57].

(#2) Per metodo si intende stringa, ovvero un offset alla [tabella stringhe](#)^[57], con formato "`<nome_istanza>.<hexid_metodo>`" oppure "`<nome_istanza>[<indice>].<hexid_metodo>`" (hexid = numero 32bit esadecimale ascii, senza prefisso 0x. Esempio: #000F00A09).

Tabella classi OB

La tabella classi OB (OBCLASSES) è composta da un numero di definizioni (OBCLASS) contigue, ognuna delle quali contiene tutte le informazioni su una specifica classe object block definita dall'utente.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[33].OBCLASSOFF e [HEADER](#)^[33].OBCLASSCNT.

Definizione classe OB

Una classe OB (OBCLASS) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Description
ID	+0	U32	ID classe OB Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati)
FLAGS	+6	U16	Impostazioni della classe OB: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo MODIFIER non compresso (NCF1).
NAME	+8	U32	Offset nome della classe OB (relativo a stringhe ^[57]).
TEXT	+12	U32	Offset descrizione della classe OB (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF0).
VERSION	+16	U32	Versione classe OB (formato nvMake).
INTFID	+20	U32	Identificativo univoco interfaccia obeit block.
PATHID	+24	U32	Identificativo percorso del file .OBB: 0=Percorso assoluto 1=(riservato) 2=Percorso relativo a workspace 3=Percorso relativo a factory RDE3
FILENAME	+28	U32	Offset nome del file .OBB (relativo a stringhe ^[57])(#1).
SRCID	+32	U32	Identificativo sorgente ^[62] della classe OB.
METHODSOFF	+36	U32	Offset della tabella metodi OB ^[55] (relativo al file)

Nome	Offset	Tipo	Description
METHODSCNT	+40	U32	Dimensione tabella metodi OB ^[55]
ATTRIB	+44	U32	Attributi classe OB.
MODIFIER	+48	U32	Offset modificatori della classe OB (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF1).

(#1) In caso di PATHID diverso da 0, il nome del file .OB non presenta percorsi al suo interno, ma è semplicemente il nome del file con relativa al percorso PATHID: in caso invece di PATHID uguale a 0, il nome del file .OB potrebbe contenere un percorso relativo/assoluto.

Tabella metodi OB

La tabella metodi OB (OBMETHODS) è composta da un numero di definizioni (OBMETHOD) contigue, ognuna delle quali contiene tutte le informazioni su una specifico metodo OB relativo alla classe OB di appartenenza..

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [OBCLASS](#)^[54].METHODSOFF e [OBCLASS](#)^[54].METHODSCNT.

Definizione metodo OB

Una metodo OB (OBMETHOD) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID metodo OB. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati)
FLAGS	+6	U16	Impostazioni del metodo OB: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo MODIFIER non compresso (NCF1).
NAME	+8	U32	Offset nome del metodo OB (relativo a stringhe ^[57]).
KNAME	+12	U32	Offset nome chiave del metodo OB (relativo a stringhe ^[57]).
TEXT	+16	U32	Offset descrizione del metodo OB (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF0).

Nome	Offset	Tipo	Descrizione
CALLID	+20	U32	ID metodo in chiamata (#1)
PARAMSOFF	+24	U32	Offset della tabella parametri per metodo OB ^[56] (relativo al file).
PARAMSCNT	+28	U32	Dimensione della tabella parametri per metodo OB ^[56] .
RETTYPEID	+32	U32	ID tipo variabile di ritorno. Nota: può essere un tipo predefinito ^[58] oppure definito dall'utente ^[35] .
ATTRIB	+36	U32	Attributi metodo.
MODIFIER	+40	U32	Offset modificatori del metodo OB (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF1).

(#1) Il valore del campo CALLID deriva dal valore del campo METHOD.ID, nel file .OBB di origine.

Tabella parametri per metodo OB

La tabella parametri per metodo OB (OBMETHODPS) è composta da un numero di definizioni (OBMETHODP) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico parametro relativo al metodo OB di appartenenza.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [OBMETHOD](#)^[55].PARAMSOFF e [OBMETHOD](#)^[55].PARAMSCNT.

Definizione parametro per metodo OB

Un parametro per metodo OB (OBMETHODP) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
TYPEID	+0	U32	ID tipo variabile Nota: può essere un tipo predefinito ^[58] oppure definito dall'utente ^[35] .
ATTRIB	+4	U16	Attributi parametro: 0x0001 Parametro passato per riferimento (ABYREF) 0x0002 Parametro passato per puntatore (ABYPTR)

Nome	Offset	Tipo	Descrizione
FLAGS	+6	U16	Impostazioni del parametro: 0x0001 Campo TEXT non compresso (NCF0).
NAME	+8	U32	Offset nome del parametro (relativo a stringhe ^[57]).
TEXT	+12	U32	Offset descrizione del parametro (relativo a stringhe compresse ^[57] , o stringhe ^[57] se NCF0).
DIM0	+16	U32	1a dimensione array (0=non specificata) (#1).
DIM1	+20	U32	2a dimensione array (0=non specificata) (#1).
DIM2	+24	U32	3a dimensione array (0=non specificata) (#1).

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW la dimensione massima va comunque considerata solo a 16bit (ovvero massimo indice 0xFFFF).

Tabella stringhe

La tabella stringhe (STRINGS) è composta da un certo numero di stringhe ascii consecutive, ognuna terminata il carattere ascii \0 (zero) e a dimensione variabile. La tabella deve contenere almeno un carattere e deve essere \0, di modo che le stringhe con offset 0 puntino effettivamente ad una stringa nulla.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata:

- dai campi [HEADER](#)^[33].STRINGSOFF e [HEADER](#)^[33].STRINGSSIZE, se usata in file ladder/diagram.
- dai campi [LIVEHEADER](#)^[133].STRINGSOFF e [LIVEHEADER](#)^[133].STRINGSSIZE, se usata in modifiche live.

Tabella stringhe compresse

La tabella stringhe compresse (CSTRINGS) è composta da un certo numero di stringhe ascii consecutive, ognuna terminata il carattere ascii \0 (zero) e a dimensione variabile.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente dai campi [HEADER](#)^[33].STRINGSOFF e [HEADER](#)^[33].STRINGSSIZE: lo stato di compressione, che è opzionale, è invece specificato dal campo [HEADER](#)^[33].CSTRINGSTYPE.

Annotazioni:

- in caso di compressione `HEADER.CSTRINGSTYPE = 1`, utilizzare le funzioni standard di libreria (QT >= 3.3.4) `qUncompress()` per decomprimere la tabella in modo corretto.
- gli offset che fanno riferimento a questa tabella si intendono sempre riferiti alla tabella non compressa.

Tabella modifiche live

La tabella di modifiche live incorporate (EMBLIVES) è composta da un numero di definizioni (EMBLIVE) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico insieme di modifiche live al file ladder stesso: ogni insieme è in realtà un file di [modifiche live](#) indipendente ma incorporato nel file ladder stesso.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi `HEADER.EMBLIVESOFF` e `HEADER.EMBLIVESCNT`.

In fase di caricamento, i vari file di modifiche live vengono applicati nell'esatto ordine in cui compaiono. L'applicazione effettiva delle modifiche è regolata dall'apposito flag in `EMBLIVE.FLAGS` (valore 0x0001): solo l'ultimo file di modifica live può avere eventualmente questo flag a falso, indicando delle modifiche live non ancora applicate.

Definizione modifica live incorporata

Una modifica live incorporata (EMBLIVE) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
FILEOFF	+0	U32	Offset del file di modifiche live (offset relativo al file ladder).
FILESIZE	+4	U32	Dimensione del file di modifiche live .
FLAGS	+8	U16	Impostazioni del file: 0x0001 = Modifiche effettive da applicare.
	+10	B[22]	(riservati)

Tipi di dati predefiniti

I tipi di dati vengono identificati in base ad un codice ID univoco, siano essi predefiniti oppure definiti dall'utente: gli ID vengono suddivisi nel seguente modo:

Range	Descrizione
0x00000000-0x0000FFFF	Tipi dati predefiniti (#1)
0x00010000-0xFFFFFFFF	Tipi dati definiti dall'utente
0xFFFFFFFF	Riservato

(#1) i tipi di dati predefiniti non vengono mai salvati nel file .LAD, ma si considerano sempre esistenti.

I tipi di dati predefiniti sono i seguenti:

ID	Simbolico	Descrizione	RTE (0x00)
0x00000000	(riservato)	Tipo dato non valido	
0x00000001	I8	Intero 8bit con segno	1
0x00000002	U8	Intero 8bit senza segno	1
0x00000003	I16	Intero 16bit con segno	2
0x00000004	U16	Intero 16bit senza segno	2
0x00000005	I32	Intero 32bit con segno	4
0x00000006	U32	Intero 32bit senza segno	4
0x00000007	I64	Intero 64bit con segno	
0x00000008	U64	Intero 64bit senza segno	
0x00000009	STRING	Stringa (0 terminated)	...
0x0000000A	FLOAT	Floating point 32bit	4
0x0000000B	REAL	Floating point 64bit (double)	8
0x0000000C	BOOL	Booleano	4 (#1)
0x0000000D	TIMER	Struttura predefinita timer ⁶⁰	32
0x0000000E	COUNTER	Struttura predefinita contatore ⁶⁰	48
0x0000000F	ALIAS	Alias (o literally) ⁵⁹	0
0x00000010	EDGE	Storage speciale per elementi fronte di salita (RE) e fronte di discesa (FE)	4
0x00000011	POWERSET	Variabile logica "Power Set"	4 (#2)
0x00000012	AXESGROUP	Variabile logica "Gruppo assi"	4 (#2)
0x00000013	PATH	Variabile logica "Path"	4 (#2)
0x00000014	POINT_L	Variabile logica "Punto di libreria"	4 (#2)

(#1) il tipo BOOL in RTE viene rappresentato, per motivi di performance, come I32, mentre da RDE3 viene visto come U8.

(#2) Le variabili logiche in RTE sono sempre rappresentate come un puntatore in memoria (4bytes), anche qualora fossero incorporate in strutture definite dall'utente. Per ulteriore

informazioni sulla composizione (elementi) di tali variabili, consultare la documentazione tecnica di RTE).

La rappresentazione e la dimensione dei tipi di dati dipendono direttamente dall'ID destinazione runtime (`HEADER[33].TARGETID`). Le dimensioni dei tipi dati sono elencati nelle colonne RTE e sono espressi in bytes.

Struttura predefinita timer

La struttura predefinita TIMER è composta dai seguenti campi:

Tipo	Nome	Descrizione	Letture	Scrittura
BOOL	Q	Esito del timer	X	
REAL	ET	Valore attuale [ms]	X	X (#1)
REAL	PT	Valore di preset [ms]	X	X
REAL	PT2	Valore di preset 2 [ms]	X	X
BOOL	EN	Abilitazione timer	X	

(#1) Il campo ET può essere scritto solo quando il timer non è ancora stato assegnato.

Struttura predefinita contatore

La struttura predefinita COUNTER è composta dai seguenti campi:

Tipo	Nome	Descrizione	Letture	Scrittura
BOOL	QU	Esito del conteggio rispetto a preset	X	
BOOL	QD	Esito del conteggio rispetto a 0	X	
I32	CV	Valore attuale di conteggio	X	X (#1)
I32	PV	Valore di preset	X	X
BOOL	CU	Stato del segnale "count up"	X	
BOOL	CD	Stato del segnale "count down"	X	

(#1) Il campo CV può essere scritto solo quando il contatore non è ancora stato assegnato.

Alias (o literally)

L'elenco dei tipi dati utilizzabili come alias sono i seguenti:

ID	Nome	Tipo	Descrizione
0x0001	INP(n)	BOOL	Canale input
0x0002	INP_W(n)	U16	Input word 16bit
0x0003	INP_W(n).b	BOOL	Input word 16bit, accesso a bit
0x0004	INP_DW(n)	U32	Input word 32bit
0x0005	INP_DW(n).b	BOOL	Input word 32bit, accesso a bit
0x0006	OUT	BOOL	Canale output
0x0007	OUT_W(n)	U16	Output word 16bit
0x0008	OUT_W(n).b	BOOL	Output word 16bit, accesso a bit
0x0009	OUT_DW(n)	U32	Output word 32bit
0x000A	OUT_DW(n).b	BOOL	Output word 32bit, accesso a bit
0x000B	R(n)	I32	Registro globale intero (volatile)
0x000C	R(n).b	BOOL	Registro globale intero (volatile), accesso a bit
0x000D	NVR(n)	I32	Registro globale intero (non volatile)
0x000E	NVR(n).b	BOOL	Registro globale intero (non volatile), accesso a bit
0x000F	RR(n)	REAL	Registro globale reale (volatile)
0x0010	NVRR(n)	REAL	Registro globale reale (non volatile)
0x0011	AM(n)	U32	Maschera allarmi
0x0012	AM(n).b	U32	Maschera allarmi, accesso a bit

Annotazioni:

- se la variabile è di tipo array, l'indice di array (n) può essere o una costante o un variabile non indicizzata.
- se la variabile prevede l'accesso a bit, .b può essere una costante o una variabile e deve essere nel range compatibile la grandezza della variabile di riferimento.

Variabili predefinite

Le variabili vengono identificate in base ad un codice ID univoco, siano esse predefinite o definite dall'utente: gli ID vengono suddivisi nel seguente modo:

Range	Descrizione
0x00000000-0x0000FFFF	Variabile predefinita
0x0001000-0xFFFFFFFF	Variabile definita dall'utente
0xFFFFFFFF	Riservato

ATTENZIONE: le variabili predefinite non vengono mai salvate nel file .LAD, ma si considerano sempre esistenti.

Le variabili predefinite sono i seguenti:

ID	Nome	Tipo	Descrizione
0x00000001	INP(n)	BOOL	Canale input
0x00000002	INP_W(n)	U16	Input word 16bit
0x00000003	INP_DW(n)	U32	Input word 32bit
0x00000004	OUT	BOOL	Canale output
0x00000005	OUT_W(n)	U16	Output word 16bit
0x00000006	OUT_DW(n)	U32	Output word 32bit
0x00000007	R(n)	I32	Registro globale intero (volatile)
0x00000008	NVR(n)	I32	Registro globale intero (non volatile)
0x00000009	RR(n)	REAL	Registro globale reale (volatile)
0x0000000A	NVRR(n)	REAL	Registro globale reale (non volatile)
0x0000000B	AM(n)	U32	Maschera allarmi

Annotazioni:

- se la variabile è di tipo array, l'indice di array (n) può essere o una costante o un variabile non indicizzata.
- RDE3 crea ulteriori variabili predefinite con assegnazione degli ID in modo del tutto arbitrario (nel range 0x00000100-0x00010000): RTE riconosce tali variabili in base al nome e non in base a tale ID.

Identificativi sorgente

Gli identificativi sorgente vengono utilizzati per identificare in modo univoco la sorgente di una certa entità (o elemento) all'interno del file ladder, come per esempio strutture, variabili o classi OB. Il valore dell'identificativo U32 viene composto come segue:

- U16 alta, identificativo gruppo;

- U16 bassa, identificativo elemento nel gruppo.

Gli identificativi sorgente previsti sono i seguenti:

Gruppo	Elemento	Descrizione
0x0000	0x0000	Entità definita manuale dall'utente
0x0001	0x0000	Entità importata da file esterno: file generico
0x0002	0x0000	Entità importata (forzata) da progetto RTE: sorgente generica
0x0002	0x0001	Entità importata (forzata) da progetto RTE: sorgente configuratore

Formato file (versione 1.4)

Formato file (versione 1.4)

Un programma in linguaggio di programmazione ladder/diagram è un singolo file (estensione .LAD) contenente sia le informazioni per l'editing (sorgente) che le informazioni per l'esecuzione (runtime).

Un programma ladder è composto principalmente da una [intestazione](#)^[63], quindi da varie sezioni tra cui le principali sono:

- [tabella strutture](#)^[66], ovvero i tipi dati definiti dall'utente.
- [tabella variabili locali](#)^[68], ovvero le variabili generali del programma.
- [tabella classi OB](#)^[85], ovvero le classi object block a cui il programma fa riferimento.
- [tabella blocchi](#)^[71], ovvero le sub-routines in altri linguaggi di programmazione;
- [tabella modifiche live incorporate](#)^[89], ovvero eventuali file di modifica live incorporati;
- [tabella stringhe](#)^[88];
- [tabella stringhe compresse](#)^[89].

Nell'elenco dei blocchi, esiste sempre un blocco principale detto MAIN ed eventualmente ulteriori blocchi (paragonabili a sub-routines di altri linguaggi di programmazione): ogni blocco contiene un [elenco di circuiti](#)^[72] che costituiscono il programma ladder effettivo: ogni circuito a sua volta è un elenco logicamente strutturato di [elementi](#)^[73].

NOTA: tutti i dati numerici memorizzati nel presente file sono da considerarsi in formato little indian (standard x86).

Intestazione file

Ad inizio file (offset 0) è presente una intestazione generale (HEADER) che contiene le informazioni di accesso a tutte le restanti componenti e sotto-sezioni del file.

Nome	Offset	Tipo	Descrizione
SIGN	+0	U32	Firma del file ('LADF').
VERSION	+4	U32	Versione del file (1.4.0).
GENERATOR	+8	U32	Nome e versione del generatore del file (offset relativo a tabella stringhe).
TYPE	+12	U16	Tipo di esecuzione: 0 = Esecuzione sincrona (SY). 1 = Esecuzione ad alta priorità (HP). 2 = Esecuzione a priorità normale (NP). 3 = Esecuzione a bassa priorità (LP).
FREQUENCY	+14	U16	Frequenza di esecuzione in [hz], da 1hz a 3000hz, default 200hz .
FLAGS	+16	U32	Impostazioni generali: 0x00000001 = Il file contiene errori di verifica formale. 0x00000002 = Il file contiene riferimenti a file .OBB esterni (classi OB) 0x00000004 = Il file utilizza codifica UTF-8
BUILDEAR	+20	U16	Anno ultima generazione (4 digit).
BUILDMONTH	+22	U8	Mese ultima generazione (1-12).
BUILDDAY	+23	U8	Giorno ultima generazione (1-31).
BUILDHOUR	+24	U8	Ultima ora generazione (0-23).
BUILDMINUTE	+25	U8	Minuto ultima compilazione (0-59).
BUILDSECOND	+26	U8	Secondo ultima compilazione (0-59).
TARGETID	+27	U8	ID destinazione runtime:

Nome	Offset	Tipo	Descrizione
			0x00 = Firmware RTE (PowerPC G2) 0x01 = Firmware RTE (PowerPC P2020)
VID	+28	U32	ID versione del file.
PLVID	+32	U32	ID versione del file pre modifiche live (se presenti, altrimenti vale 0).
IARECNT	+36	U32	N. di elementi tipo ARE presenti nel file.
IAFECNT	+40	U32	N. di elementi tipo AFE presenti nel file.
	+44	U8[84]	(riservati)
BLOCKSOFF	+128	U32	Offset della tabella dei blocchi (relativo al file).
BLOCKSCNT	+132	U32	N. di elementi della tabella dei blocchi .
STRINGSOFF	+136	U32	Offset della tabella stringhe (relativo al file).
STRINGSSIZE	+140	U32	Dimensione della tabella stringhe .
CSTRINGSOFF	+144	U32	Offset della tabella stringhe compresse (relativo al file).
CSTRINGSSIZE	+148	U32	Dimensione della tabella stringhe compresse .
CSTRINGSTYPE	+152	U8	Tipo di compressione usata per la tabella stringhe compresse : 0 = Nessuna compressione. 1 = Compressione standard Qt 3.3.x.
	+153	U8[3]	(riservati)
EMBLIVESOFF	+156	U32	Offset della tabella modifiche live incorporate (relativo al file).

Nome	Offset	Tipo	Descrizione
EMBLIVESCNT	+160	U32	N. di elementi della tabella modifiche live incorporate ^[89] .
STRUCTSOFF	+164	U32	Offset della tabella strutture ^[66] (relativo al file).
STRUCTSCNT	+168	U32	N. di elementi della tabella strutture ^[66] .
VARSOFF	+172	U32	Offset della tabella variabili ^[68] (relativo al file).
VARSCNT	+176	U32	N. di elementi della tabella variabili ^[68] .
OBCLASSESOFF	+180	U32	Offset della tabella classi OB ^[85] (relativo al file)
OBCLASSESCNT	+184	U32	N. di elementi della tabella classi OB ^[85] (relativo al file)
LASTBLOCKID	+188	U32	Ultimo ID blocco utilizzato.
LASTRUNGID	+192	U32	Ultimo ID circuito utilizzato.
LASTSTRUCTID	+196	U32	Ultimo ID struttura utilizzato.
LASTVARID	+200	U32	Ultimo ID variabile utilizzato.
LASTOBCLASSID	+204	U32	Ultimo ID classe OB utilizzato
	+208	U8[304]	(riservati)

Tabella strutture

La tabella strutture (STRUCTS) è composta da un numero di definizioni (STRUCT) contigue, ognuna delle quali contiene tutte le informazioni su uno specifica struttura definita dall'utente: in questa tabella non sono presenti i [tipi dati predefiniti](#)^[90].

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[63].STRUCTSOFF e [HEADER](#)^[63].STRUCTSCNT.

Definizione struttura

Una struttura (STRUCT) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID tipo struttura Nota: solo valori 0x00010000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati, uso runtime)
FLAGS	+6	U16	Impostazioni della struttura: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo MODIFIER non compresso (NCF1).
NAME	+8	U32	Offset nome della struttura (relativo a stringhe ^[88]).
TEXT	+12	U32	Offset descrizione della struttura (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF0).
VARSOFF	+16	U32	Offset della tabella variabili ^[68] (relativo al file) (#1)
VARSCNT	+20	U32	Dimensione tabella variabili ^[68]
SIZE	+24	U32	Dimensione della struttura [byte]
SRCID	+28	U32	Identificativo sorgente ^[94] della struttura
OBCID	+32	U32	ID classe OB di riferimento (0=nessuna)(#2,#3)
ATTRIB	+36	U32	Attributi della struttura: 0x00000001 Struttura riferita a classe OB (AREFOBC)(#2) 0x00000002 Struttura principale classe OB (AISOBC)(#2,#3) 0x00000004 Struttura (rif.a) definita esternamente (AEXTERNAL)

Nome	Offset	Tipo	Descrizione
MODIFIER	+40	U32	Offset modificatori della struttura (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF1).
REFID	+44	U32	ID di riferimento tipo struttura (#4).
EXTNAME	+48	U32	Offset nome esterno struttura (relativo a stringhe ^[88]).
	+52	U8[16]	(riservati)

(#1) La definizione di struttura usa una normale tabella di definizione variabili per definire i propri elementi. In questo caso gli ID delle variabili non hanno senso e dovrebbero essere sempre a 0.

(#2) A pari valore di OBCID non 0, con attributo AREFOBC, può esistere una sola struttura con flag AISOBC, mentre posso essere 0 o più strutture senza attributo AISOBC.

(#3) Una struttura viene definita "orfana" qualora si trova nella condizione di attributi AREFOBC impostato e campo OBCID = 0: questa condizione capita quando aggiornando/rimuovendo una classe OB, vi sono ancora delle variabili (o altro) che vi fanno riferimento.

(#4) In caso di attributo AREFOBC il campo REFID contiene l'ID della struttura come definito del file OBB di origine (0 in case di attributo AISOBC).

Tabella variabili

La tabella variabili (VARS) è composta da un numero di definizioni (VAR) contigue, ognuna delle quali contiene tutte le informazioni su uno specifica variabile definita dall'utente: in questa tabella non sono presenti le [variabili predefinite](#)^[93].

La posizione (relativa al file) e il numero di elementi della tabella delle variabili di programma sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata

- dai campi [HEADER](#)^[63].VARSOFF e [HEADER](#)^[63].VARSCNT se utilizzati in intestazione file;
- dai campi [STRUCT](#)^[66].VARSOFF e [STRUCT](#)^[66].VARSCNT se utilizzati in definizione struttura;
- dai campi [LIVECMD](#)^[142].VARSOFF e [LIVECMD](#)^[142].VARSCNT se utilizzati in definizione comando.

Definizione variabile

Una variabile (VAR) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID variabile (#2) Nota: solo valori 0x00010000-0xFFFFFFFFE; il valore 0xFFFFFFFF è un ID a significato speciale; i valori 0x00000001-0x0000FFFF sono riservati per le variabili predefinite.
TYPEID	+4	U32	ID tipo variabile

Nome	Offset	Tipo	Descrizione
			Nota: può essere un tipo predefinito ^[90] oppure definito dall'utente ^[66] .
	+8	U8[2]	(riservati, uso runtime)
FLAGS	+10	U16	Impostazioni della variabile: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo BITTEXT non compresso (NCF1) 0x0004 Campo MODIFIER non compresso (NCF2)
NAME	+12	U32	Offset nome della variabile (relativo a stringhe ^[88]).
TEXT	+16	U32	Offset descrizione della variabile (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF0).
BITTEXT	+20	U32	Offset descrizioni dei bit variabili, ovvero sequenza di stringhe relative ad ogni singolo bit separate dal carattere ascii 0xFF (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF0). (#2)
DIM0	+24	U32	1a dimensione array (0=non specificata) (#1) o indice 0 per Alias ^[90] (TYPEID = 0x0000000F).
DIM1	+28	U32	2a dimensione array (0=non specificata) (#1) o indice 1 per Alias ^[90] (TYPEID = 0x0000000F).
DIM2	+32	U32	3a dimensione array (0=non specificata) (#1) o indice 2 per Alias ^[90] (TYPEID = 0x0000000F).
ALIASID	+36	U16	ID tipo alias ^[92] (#2)

Nome	Offset	Tipo	Descrizione
ALIASBIT	+38	U16	N. di bit per alias (dipende dal tipo di ID alias)(#2)
SIZE	+40	U32	Dimensione della variabile. (#3)
SRCID	+44	U32	Identificativo sorgente ^[94] della variabile.(#2,#4)
VALUE	+48	U32	Offset valore iniziale della variabile (relativo a stringhe ^[88] , 0=nessun valore).
ATTRIB	+52	U32	Attributi variabile: 0x00000001 La variabile può essere scritta (WRITE) 0x00000002 La variabile può essere letta (READ) 0x00000004 La variabile può essere forzata (FORCE) 0x00000008 La variabile può essere rilasciata (RELEASE) 0x00000010 La variabile è costante (CONST) 0x00000020 La variabile non ha accesso a bit (NOBITS)
MODIFIER	+56	U32	Offset modificatori della variabile (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF2).
DIM0L	+60	U32	Offset etichetta per 1a dimensione array (relativo a stringhe ^[88]). (#5)
DIM1L	+64	U32	Offset etichetta per 2a dimensione array (relativo a stringhe ^[88]). (#5)
DIM2L	+68	U32	Offset etichetta per 3a dimensione array (relativo a stringhe ^[88]). (#5)

Nome	Offset	Tipo	Descrizione
	+72	U8[16]	(riservati)

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW la dimensione massima va comunque considerata solo a 16bit (ovvero massimo indice 0xFFFF).

(#2) Campo non utilizzato (valore 0) se utilizzata in definizione struttura

(#3) Campo non utilizzato (valore 0) se utilizzato definendo un Alias (TYPEID = 0x0000000F)

(#4) Campo non utilizzato (valore 0) se utilizzata in definizione comando live.

(#5) Per etichetta si intende nome di una variabile (a valore costante) oppure la rappresentazione ascii di un valore numerico, ovvero un offset alla [tabella stringhe](#)^[88].

Tabella blocchi

La tabella blocchi (BLOCKS) è composta da un numero di definizioni di blocco (BLOCK) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico blocco e del relativo contenuto su file.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[63].BLOCKSOFF e [HEADER](#)^[63].BLOCKSCNT.

Definizione blocco

Un blocco (BLOCK) è composto dai seguenti dati:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID univoco del blocco. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
STATUS	+4	U16	(riservato, uso runtime)
FLAGS	+6	U16	Impostazioni del blocco: 0x0001 = Blocco principale (ISMAIN). 0x0002 = Campo TEXT non compresso (NCF0). 0x0004 = Campo COMMENT non compresso (NCF1).
NAME	+8	U32	Offset nome del blocco (relativo a stringhe ^[88]). Nota: se flag ISMAIN specificato, il nome viene deve essere "main".
TEXT	+12	U32	Offset descrizione del blocco (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF0).

Nome	Offset	Tipo	Descrizione
COMMENT	+16	U32	Offset commenti per il blocco (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF1).
RUNGSOFF	+20	U32	Offset della tabella circuiti del blocco ^[72] (relativo a file).
RUNGSCNT	+24	U32	N. di elementi della tabella circuiti del blocco ^[72] .
	+28	B[36]	(riservati)

Tabella circuiti del blocco

La tabella circuiti del blocco (RUNGS) è composta da un numero di definizioni di circuito (RUNG) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico circuito e del relativo contenuto.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [BLOCK](#)^[71].RUNGSOFF e [BLOCK](#)^[71].RUNGSCNT.

Definizione circuito del blocco

Un circuito del blocco (RUNG) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID univoco del circuito. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati, uso runtime)
FLAGS	+6	U16	Impostazioni del circuito: 0x0001 Campo TEXT non compresso (NCF0).
REFID	+8	U32	ID circuito di riferimento (0=nessuno).
ITEMSOFF	+12	U32	Offset della tabella elementi del circuito ^[73] (relativo al file).
ITEMSSIZE	+16	U32	Dimensione della tabella elementi del circuito ^[73] .
TEXT	+20	U32	Offset descrizione del circuito (relativo a

Nome	Offset	Tipo	Descrizione
			stringhe compresse ^[89] , o stringhe ^[88] se NCF1).
	+24	U8[4]	(riservati)

Tabella elementi circuito

La tabella elementi del circuito (ITEMS) è composta da un numero di definizioni di circuito (ITEM) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico elemento.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata:

- dai campi [RUNG](#)^[72].ITEMSOFF e [RUNG](#)^[72].ITEMSSIZE, se utilizzato in circuito del blocco (RUNG).
- dai campi [LIVECMD](#)^[142].ITEMSOFF e [LIVECMD](#)^[142].ITEMSSIZE, se utilizzato in comandi modifiche live (LIVECMD).

Definizione elemento del circuito

Gli elementi del circuito (ITEM) hanno una parte iniziale comune, quindi un'area dati specifica per ogni tipo di elemento. La parte comune è la seguente:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	Codice tipo elemento.
SIZE	+2	U16	Dimensione completa dell'elemento (tutti i campi).
DATA	+4	...	Dati specifici (opzionali).

Gli elementi previsti sono i seguenti:

Simbolico	ID	Tipo	Dati	Descrizione
BB	0x0001			Inizio ramo.
NL	0x0002			Prossimo livello ramo . Note: deve sempre essere compreso tra un BB e un EB.
EB	0x0003			Fine ramo . Note: deve sempre corrispondere ad un BB.
OC	0x0004		U32=Destinazione (#3): solo tipi	Contatto aperto.

Simbolico	ID	Tipo	Dati	Descrizione
			BOOL ^[90]	
CC	0x0005		U32=Destinazione (#3): solo tipi BOOL ^[90]	Contatto chiuso.
RE	0x0006		U32=Destinazione (#3): solo tipi EDGE ^[90]	Fronte di salita.
FE	0x0007		U32=Destinazione (#3): solo tipi EDGE ^[90]	Fronte di discesa.
AF	0x0008			Sempre falso.
OUT	0x0009	F	U32=Destinazione (#3): solo tipi BOOL ^[90]	Output.
OL	0x000A	F	U32=Destinazione (#3): solo tipi BOOL ^[90]	Output latch (set).
OUL	0x000B	F	U32=Destinazione (#3): solo tipi BOOL ^[90]	Output unlatch (reset).
NOUT	0x000C	F	U32=Destinazione (#3): solo tipi BOOL ^[90]	Output negato
CLR	0x000D	F	U32=Destinazione (#3): solo tipi TIMER ^[90] e COUNTER ^[90]	Azzeramento (clear).
TP	0x000E	F	U32=Destinazione (#3): solo tipi TIMER ^[90] U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer pulse.
TON	0x000F	F	U32=Destinazione (#3): solo tipi TIMER ^[90] U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer on delay.
TOFF	0x0010	F	U32=Destinazione (#3): solo tipi TIMER ^[90]	Abilita timer off delay.

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	
TSQ	0x0011	F	U32=Destinazione (#3): solo tipi TIMER ₉₀ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer onda quadra.
TINT	0x0012	F	U32=Destinazione (#3): solo tipi TIMER ₉₀ U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer integrale.
CU	0x0013	F	U32=Destinazione (#3): solo tipi COUNTER ₉₀ U32=Valore di preset (#4)	Conteggio up.
CD	0x0014	F	U32=Destinazione (#3): solo tipi COUNTER ₉₀ U32=Valore di preset (#4)	Conteggio down.
CUR	0x0015	F	U32=Destinazione (#3): solo tipi COUNTER ₉₀ U32=Valore di preset (#4)	Conteggio up rotativo.
CDR	0x0016	F	U32=Destinazione (#3): solo tipi COUNTER ₉₀ U32=Valore di preset (#4)	Conteggio down rotativo.
LD	0x0017	F	U32=Destinazione (#3): solo tipi COUNTER ₉₀	Preset su contatore.
MATH	0x0018	F	U32=Espressione (#5) U32=Destinazione (#3)	Blocco matematico.

Simbolico	ID	Tipo	Dati	Descrizione
			U8[]= Dati espressione matematica ⁸²	
EQ	0x0019		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto uguaglianza (A == B).
NE	0x001A		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto diverso (A != B).
GT	0x001B		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto maggiore (A > B).
GE	0x001C		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto maggiore/uguale (A >= B).
LT	0x001D		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto minore (A < B).
LE	0x001E		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto minore/uguale (A <= B).
COPY	0x001F	F	U32=Sorgente (#2) U32=Destinazione (#3) U32=N. Elementi (#2)	Copia dati da sorgente a destinazione, per il n. di elementi specificato.
CALL	0x0020	F	U32 ID blocco da chiamare	Chiamata di un blocco (sub-routine).
RET	0x0021	F		Ritorno da un blocco all'interno.
ADD	0x0022	F	U32=Sorgente A (#2) U32=Sorgente B (#2)	Somma (A + B).

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Destinazione (#3)	
SUB	0x0023	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Sottrazione (A - B).
MUL	0x0024	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Moltiplicazione (A * B).
DIV	0x0025	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Divisione (A / B).
AND	0x0026	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	AND binario (A & B).
OR	0x0027	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	OR binario (A B).
NOT	0x0028	F	U32=Sorgente A (#2) U32=Destinazione (#3)	NOT binario ($\sim A$).
XOR	0x0029	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	XOR binario (A ^ B).

Simbolico	ID	Tipo	Dati	Descrizione
SQRT	0x002A	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Radice quadrata di A.
MOD	0x002B	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Modulo (A mod B).
POW	0x002C	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Elevamento a potenza (A pow B).
SIN	0x002D	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Seno di A.
COS	0x002E	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Coseno di A.
TAN	0x002F	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Tangente di A.
ASIN	0x0030	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arcoseno di A.
ACOS	0x0031	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arcocoseno di A.
ATAN	0x0032	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arcotangente di A.
ATAN2	0x0033	F	U32=Sorgente A (#2)	Arcotangente di A, B.

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Sorgente B (#2) U32=Destinazione (#3)	
LOG	0x0034	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Logaritmo base 10 di A.
LN	0x0035	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Logaritmo base E di A.
ABS	0x0036	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Valore assoluto di A
SIGN	0x0037	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Segno di A.
ROUND	0x0038	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arrotondamento all'intero più vicino ad A.
MIN	0x0039	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Minimo tra A e B.
MAX	0x003A	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Massimo tra A e B.
RANGE	0x003B	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Sorgente C (#2)	Verifica se A contenuto nel range B-C.

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Destinazione (#3): solo tipi BOOL ₉₀	
LIMIT	0x003C	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Sorgente C (#2) U32=Destinazione (#3)	Limita A al range B-C.
ROR	0x003D	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Rotazione bit a destra (A >> B).
ROL	0x003E	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Rotazione bit a sinistra (A << B).
SHR	0x003F	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Shift bit a destra (A >> B).
SHL	0x0040	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Shift bit a sinistra (A << B).
MOV	0x0041	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Assegna A.
TODEG	0x0042	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Conversione di A da RAD a DEG.

Simbolico	ID	Tipo	Dati	Descrizione
TORAD	0x0043	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Conversione di A da DEG a RAD.
NEG	0x0044			Negazione stato logico.
NOP	0x0045	F		Nessuna operazione.
EVAL	0x0046		U32=Espressione (#5) U8[]=Dati espressione matematica ^[82]	Valutazione espressione (booleana).
OBCALL	0x0047	F	U32=Nome istanza (#6) U32=Nome completo metodo (#9) U32=Elenco nomi parametro (#7) U32=Elenco valori parametro (#7) U32=Risultato (#3) U8[]=Dati espressione matematica ^[82] (#8)	Chiamata a metodo di una istanza OB.
FT	0x0048			Esecuzione iniziale
ARE	0x0049			Fronte di salita (anonimo)
AFE	0x004A			Fronte di discesa (anonimo)

(#1) Variabile a bit locale (tipo I32 o U32) con indirizzamento diretto (eventuali indici e numero di bit costanti).

(#2) Per sorgente/destinazione si intende nome di una variabile oppure la rappresentazione ASCII di un valore numerico (solo sorgente), ovvero un offset alla [tabella stringhe](#)^[88].

(#3) Per destinazione si intende nome di una variabile con offset alla [tabella stringhe](#)^[88].

(#4) Per valore (o valore di preset) si intende un valore numerico in formato stringa: stringa vuota (o offset 0) si intende nessun valore assegnato.

(#5) Espressione intesa come stringa con offset alla [tabella stringhe](#)^[88].

(#6) Per nome istanza si intende nome di una variabile con offset alla [tabella stringhe](#)^[88], il cui TYPEID sia riferito ad una struttura utente con flag FISOBC.

(#7) Per elenco si intende una stringa (con offset alla [tabella stringhe](#)^[88]), composta da un certo numero di sotto-stringhe separate con carattere ascii 0xFF: i due elenchi (nomi/valori) DEVONO avere sempre lo stesso n. di elementi o essere entrambe vuote (=0).

(#8) La chiamata ad un metodo di una istanza OB viene effettivamente tradotta come una espressione matematica, in cui vengono preparati in stack gli eventuali parametri (valori in ordine naturale), viene effettuata la chiamata al metodo e quindi ritrovato l'eventuale risultato da stack.

(#9) Nome completo del metodo (univoco), inteso come stringa con offset alla [tabella stringhe](#)^[88].

Dati espressione matematica

I dati dell'espressione matematica (usati dagli elementi MATH, COND e OBCALL), sono una serie di operazioni contigue (ordinati in polacca inversa) con la seguente intestazione comune:

Nome	Offset	Tipo	Descrizione
CODE	+0	U8	Codice operazione.
DATA	+1	...	Dati specifici (opzionali).

Le operazioni previste sono le seguenti:

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
PUSH	0x01	DBL=valore	--	DBL	Push valore in stack.
PUSHV	0x02	U32=Sorgente (#1)	--	DBL	Push valore variabile in stack.
ADD	0x03		DBL0,DBL1	DBL	Somma.
SUB	0x04		DBL0,DBL1	DBL	Sottrazione (#2).
MUL	0x05		DBL0,DBL1	DBL	Moltiplicazione (#2).
DIV	0x06		DBL0,DBL1	DBL	Divisione (#2).
MOD	0x07		DBL	DBL	Modulo (#3).
ABS	0x08		DBL	DBL	Valore assoluto (#3).
SIN	0x09		DBL	DBL	Seno.
COS	0X0A		DBL	DBL	Coseno.
TAN	0x0B		DBL	DBL	Tangente.
ASIN	0x0C		DBL	DBL	Arcoseno.
ACOS	0x0D		DBL	DBL	Arcocoseno.
ATAN	0x0E		DBL	DBL	Arcotangente.

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
POW	0x0F		DBL	DBL	Esponente.
LOG10	0x10		DBL	DBL	Logaritmo base 10.
LOG	0x11		DBL	DBL	Logaritmo naturale.
R_AND	0x12		U32	U32	AND binario.
R_OR	0x13		U32	U32	OR binario.
R_NOT	0x14		U32	U32	NOT binario.
R_XOR	0x15		U32	U32	XOR binario.
POPV	0x16	U32=Destinazione (#1)	DBL	--	Pop valore da stack a variabile.
R_ROR	0x17		U32	U32	Rotazione bit a destra.
R_ROL	0x18		U32	U32	Rotazione bit a sinistra.
R_SHR	0x19		U32	U32	Shift bit a destra.
R_SHL	0x1A		U32	U32	Shift bit a sinistra.
TOBIN	0x1B		U32	U32	Conversione da BCD a binario.
TOBCD	0x1C		U32	U32	Conversione da binario a BCD.
TODEG	0x1D		DBL	DBL	Conversione da radianti a gradi.
TORAD	0x1E		DBL	DBL	Conversione da gradi a radianti.
ATAN2	0x1F		DBL0,DBL1	DBL	Arcotangente con due parametri.
PUSHVB	0x20	U32=Sorgente (#1)	DBL	--	Push valore variabile a bit in stack.
POPVB	0x21	U32=Destinazione (#1)	DBL	--	Pop valore da stack a variabile bit.

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
POPL	0x22		BOOL	--	Pop valore logico (0=falso, altrimenti vero).
	0x23				(riservato)
EQ	0x24		DBL0,DBL1	BOOL	Confronto: uguale.
NE	0x25		DBL0,DBL1	BOOL	Confronto: differente.
LT	0x26		DBL0,DBL1	BOOL	Confronto: minore.
LE	0x27		DBL0,DBL1	BOOL	Confronto: minore o uguale.
GT	0x28		DBL0,DBL1	BOOL	Confronto: maggiore.
GE	0x29		DBL0,DBL1	BOOL	Confronto: maggiore o uguale.
AND	0x2A		BOOL,BOOL	BOOL	AND logico.
OR	0x2B		BOOL,BOOL	BOOL	OR logico.
NOT	0x2C		BOOL	BOOL	NOT logico.
XOR	0x2D		BOOL,BOOL	BOOL	XOR logico.
CHS	0x2E		DBL	DBL	Cambio segno.
SHR	0x2F		U32	U32	Shift logico a destra.
SHL	0x30		U32	U32	Shift logico a sinistra
OBCALL	0x31	U32=Metodo (#2)	?	?	Chiamata diretta a metodo per istanza OB
POP	0x32		*	--	Pop valore (skip)
PUSHP	0x33	U32=Sorgente (#1)	U32	U32	Push puntatore a variabile

(#1) Per sorgente/destinazione si intende nome di una variabile, ovvero un offset alla [tabella stringhe](#).

(#2) Per metodo si intende stringa, ovvero un offset alla [tabella stringhe](#)^[88], con formato "`<nome_istanza>.<hexid_metodo>`" oppure "`<nome_istanza>[<indice>].<hexid_metodo>`" (hexid = numero 32bit esadecimale ascii, senza prefisso 0x. Esempio metodo n. 0xFA9 dell'istanza OB 'myob' : "myob#0000FA9").

Tabella classi OB

La tabella classi OB (OBCLASSES) è composta da un numero di definizioni (OBCLASS) contigue, ognuna delle quali contiene tutte le informazioni su una specifica classe object block definita dall'utente.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[63].OBCLASSOFF e [HEADER](#)^[63].OBCLASSCNT.

Definizione classe OB

Una classe OB (OBCLASS) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID classe OB Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati)
FLAGS	+6	U16	Impostazioni della classe OB: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo MODIFIER non compresso (NCF1).
NAME	+8	U32	Offset nome della classe OB (relativo a stringhe ^[88]).
TEXT	+12	U32	Offset descrizione della classe OB (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF0).
VERSION	+16	U32	Versione classe OB (formato nvMake).
INTFID	+20	U32	Identificativo univoco interfaccia object block.
PATHID	+24	U32	Identificativo percorso del file .OBB: 0=Percorso assoluto 1=(riservato) 2=Percorso relativo a workspace

Nome	Offset	Tipo	Descrizione
			3=Percorso relativo a factory RDE3
FILENAME	+28	U32	Offset nome del file .OBB (relativo a stringhe ^[88])(#1).
SRCID	+32	U32	Identificativo sorgente ^[94] della classe OB.
METHODSOFF	+36	U32	Offset della tabella metodi OB ^[86] (relativo al file)
METHODSCNT	+40	U32	Dimensione tabella metodi OB ^[86]
ATTRIB	+44	U32	Attributi classe OB.
MODIFIER	+48	U32	Offset modificatori della classe OB (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF1).
	+52	U8[16]	(riservati)

(#1) In caso di PATHID diverso da 0, il nome del file .OBB non presenta percorsi al suo interno, ma è semplicemente il nome del file con relativa al percorso PATHID: in caso invece di PATHID uguale a 0, il nome del file .OBB potrebbe contenere un percorso relativo/assoluto.

Tabella metodi OB

La tabella metodi OB (OBMETHODS) è composta da un numero di definizioni (OBMETHOD) contigue, ognuna delle quali contiene tutte le informazioni su una specifico metodo OB relativo alla classe OB di appartenenza..

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [OBCLASS](#)^[85].METHODSOFF e [OBCLASS](#)^[85].METHODSCNT.

Definizione metodo OB

Una metodo OB (OBMETHOD) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID metodo OB. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati)
FLAGS	+6	U16	Impostazioni del metodo OB:

Nome	Offset	Tipo	Descrizione
			0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo MODIFIER non compresso (NCF1).
NAME	+8	U32	Offset nome del metodo OB (relativo a stringhe ^[88]).
KNAME	+12	U32	Offset nome chiave del metodo OB (relativo a stringhe ^[88]).
TEXT	+16	U32	Offset descrizione del metodo OB (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF0).
CALLID	+20	U32	ID metodo in chiamata (# 1)
PARAMSOFF	+24	U32	Offset della tabella parametri per metodo OB ^[87] (relativo al file).
PARAMSCNT	+28	U32	Dimensione della tabella parametri per metodo OB ^[87] .
RETTYPEID	+32	U32	ID tipo variabile di ritorno. Nota: può essere un tipo predefinito ^[90] oppure definito dall'utente ^[66] .
ATTRIB	+36	U32	Attributi metodo.
MODIFIER	+40	U32	Offset modificatori del metodo OB (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF1).
	+44	U8[16]	(riservati)

(#1) Il valore del campo CALLID deriva dal valore del campo METHOD.ID, nel file .OBB di origine.

Tabella parametri per metodo OB

La tabella parametri per metodo OB (OBMETHODPS) è composta da un numero di definizioni (OBMETHODP) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico parametro relativo al metodo OB di appartenenza.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [OBMETHOD](#)^[86].PARAMSOFF e [OBMETHOD](#)^[86].PARAMSCNT.

Definizione parametro per metodo OB

Un parametro per metodo OB (OBMETHODP) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
TYPEID	+0	U32	ID tipo variabile Nota: può essere un tipo predefinito ^[90] oppure definito dall'utente ^[66] .
ATTRIB	+4	U16	Attributi parametro: 0x0001 Parametro passato per riferimento (ABYREF) 0x0002 Parametro passato per puntatore (ABYPTR)
FLAGS	+6	U16	Impostazioni del parametro: 0x0001 Campo TEXT non compresso (NCF0).
NAME	+8	U32	Offset nome del parametro (relativo a stringhe ^[88]).
TEXT	+12	U32	Offset descrizione del parametro (relativo a stringhe compresse ^[89] , o stringhe ^[88] se NCF0).
DIM0	+16	U32	1a dimensione array (0=non specificata) (#1).
DIM1	+20	U32	2a dimensione array (0=non specificata) (#1).
DIM2	+24	U32	3a dimensione array (0=non specificata) (#1).
	+28	B[16]	(riservati)

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW la dimensione massima va comunque considerata solo a 16bit (ovvero massimo indice 0xFFFF).

Tabella stringhe

La tabella stringhe (STRINGS) è composta da un certo numero di stringhe ascii consecutive, ognuna terminata il carattere ascii \0 (zero) e a dimensione variabile. La tabella deve contenere almeno un carattere e deve essere \0, di modo che le stringhe con offset 0 puntino effettivamente ad una stringa nulla.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata:

- dai campi [HEADER^{\[63\]}.STRINGSOFF](#) e [HEADER^{\[63\]}.STRINGSSIZE](#), se usata in file ladder/diagram.
- dai campi [LIVEHEADER^{\[141\]}.STRINGSOFF](#) e [LIVEHEADER^{\[141\]}.STRINGSSIZE](#), se usata in modifiche live.

Tabella stringhe compresse

La tabella stringhe compresse (CSTRINGS) è composta da un certo numero di stringhe ascii consecutive, ognuna terminata il carattere ascii \0 (zero) e a dimensione variabile.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente dai campi [HEADER^{\[63\]}.STRINGSOFF](#) e [HEADER^{\[63\]}.STRINGSSIZE](#): lo stato di compressione, che è opzionale, è invece specificato dal campo [HEADER^{\[63\]}.CSTRINGSCTYPE](#).

Annotazioni:

- in caso di compressione [HEADER^{\[63\]}.CSTRINGSCTYPE](#) = 1, utilizzare le funzioni standard di libreria (QT >= 3.3.4) `qUncompress()` per decomprimere la tabella in modo corretto.
- gli offset che fanno riferimento a questa tabella si intendono sempre riferiti alla tabella non compressa.

Tabella modifiche live

La tabella di modifiche live incorporate (EMBLIVES) è composta da un numero di definizioni (EMBLIVE) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico insieme di modifiche live al file ladder stesso: ogni insieme è in realtà un file di [modifiche live^{\[133\]}](#) indipendente ma incorporato nel file ladder stesso.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER^{\[63\]}.EMBLIVESOFF](#) e [HEADER^{\[63\]}.EMBLIVESCNT](#).

In fase di caricamento, i vari file di modifiche live vengono applicati nell'esatto ordine in cui compaiono. L'applicazione effettiva delle modifiche è regolata dall'apposito flag in [EMBLIVE^{\[89\]}.FLAGS](#) (valore 0x0001): solo l'ultimo file di modifica live può avere eventualmente questo flag a falso, indicando delle modifiche live non ancora applicate.

Definizione modifica live incorporata

Una modifica live incorporata (EMBLIVE) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
FILEOFF	+0	U32	Offset del file di modifiche live^[141] (offset relativo al file ladder).
FILESIZE	+4	U32	Dimensione del file di modifiche live^[141] .
FLAGS	+8	U16	Impostazioni del file: 0x0001 = Modifiche effettive da applicare.
	+10	U8[22]	(riservati)

Tipi di dati predefiniti

I tipi di dati vengono identificati in base ad un codice ID univoco, siano essi predefiniti oppure definiti dall'utente: gli ID vengono suddivisi nel seguente modo:

Range	Descrizione
0x00000000-0x0000FFFF	Tipi dati predefiniti (#1)
0x00010000-0xFFFFFFFF	Tipi dati definiti dall'utente
0xFFFFFFFF	Riservato

(#1) i tipi di dati predefiniti non vengono mai salvati nel file .LAD, ma si considerano sempre esistenti.

I tipi di dati predefiniti sono i seguenti:

ID	Simbolico	Descrizione	RTE (0x00)
0x00000000	(riservato)	Tipo dato non valido	
0x00000001	I8	Intero 8bit con segno	1
0x00000002	U8	Intero 8bit senza segno	1
0x00000003	I16	Intero 16bit con segno	2
0x00000004	U16	Intero 16bit senza segno	2
0x00000005	I32	Intero 32bit con segno	4
0x00000006	U32	Intero 32bit senza segno	4
0x00000007	I64	Intero 64bit con segno	
0x00000008	U64	Intero 64bit senza segno	
0x00000009	STRING	Stringa (0 terminated)	...
0x0000000A	FLOAT	Floating point 32bit	4
0x0000000B	REAL	Floating point 64bit (double)	8
0x0000000C	BOOL	Booleano	4 (#1)
0x0000000D	TIMER	Struttura predefinita timer ⁹¹	32
0x0000000E	COUNTER	Struttura predefinita contatore ⁹¹	48
0x0000000F	ALIAS	Alias (o literally) ⁹⁰	0
0x00000010	EDGE	Storage speciale per elementi fronte di salita (RE) e fronte di discesa (FE)	4

ID	Simbolico	Descrizione	RTE (0x00)
0x00000011	POWERSET	Variabile logica "Power Set"	4 (#2)
0x00000012	AXESGROUP	Variabile logica "Gruppo assi"	4 (#2)
0x00000013	PATH	Variabile logica "Path"	4 (#2)
0x00000014	POINT_L	Variabile logica "Punto di libreria"	4 (#2)

(#1) il tipo BOOL in RTE viene rappresentato, per motivi di performance, come I32, mentre da RDE3 viene visto come U8.

(#2) Le variabili logiche in RTE sono sempre rappresentate come un puntatore in memoria (4bytes), anche qualora fossero incorporate in strutture definite dall'utente. Per ulteriore informazioni sulla composizione (elementi) di tali variabili, consultare la documentazione tecnica di RTE).

La rappresentazione e la dimensione dei tipi di dati dipendono direttamente dall'ID destinazione runtime ([HEADER](#)₆₃.TARGETID). Le dimensioni dei tipi dati sono elencati nelle colonne RTE e sono espressi in bytes.

Struttura predefinita timer

La struttura predefinita TIMER è composta dai seguenti campi:

Tipo	Nome	Descrizione	Lettura	Scrittura
BOOL	Q	Esito del timer	X	
REAL	ET	Valore attuale [ms]	X	X (#1)
REAL	PT	Valore di preset [ms]	X	X
REAL	PT2	Valore di preset 2 [ms]	X	X
BOOL	EN	Abilitazione timer	X	

(#1) Il campo ET può essere scritto solo quando il timer non è ancora stato assegnato.

Struttura predefinita contatore

La struttura predefinita COUNTER è composta dai seguenti campi:

Tipo	Nome	Descrizione	Lettura	Scrittura
BOOL	QU	Esito del conteggio rispetto a preset	X	
BOOL	QD	Esito del conteggio rispetto a 0	X	

Tipo	Nome	Descrizione	Letture	Scrittura
I32	CV	Valore attuale di conteggio	X	X (#1)
I32	PV	Valore di preset	X	X
BOOL	CU	Stato del segnale "count up"	X	
BOOL	CD	Stato del segnale "count down"	X	

(#1) Il campo CV può essere scritto solo quando il contatore non è ancora stato assegnato.

Alias (o literally)

L'elenco dei tipi dati utilizzabili come alias sono i seguenti:

ID	Nome	Tipo	Descrizione
0x0001	INP(n)	BOOL	Canale input
0x0002	INP_W(n)	U16	Input word 16bit
0x0003	INP_W(n).b	BOOL	Input word 16bit, accesso a bit
0x0004	INP_DW(n)	U32	Input word 32bit
0x0005	INP_DW(n).b	BOOL	Input word 32bit, accesso a bit
0x0006	OUT	BOOL	Canale output
0x0007	OUT_W(n)	U16	Output word 16bit
0x0008	OUT_W(n).b	BOOL	Output word 16bit, accesso a bit
0x0009	OUT_DW(n)	U32	Output word 32bit
0x000A	OUT_DW(n).b	BOOL	Output word 32bit, accesso a bit
0x000B	R(n)	I32	Registro globale intero (volatile)
0x000C	R(n).b	BOOL	Registro globale intero (volatile), accesso a bit
0x000D	NVR(n)	I32	Registro globale intero (non volatile)
0x000E	NVR(n).b	BOOL	Registro globale intero (non volatile), accesso a bit
0x000F	RR(n)	REAL	Registro globale reale (volatile)

ID	Nome	Tipo	Descrizione
0x0010	NVRR(n)	REAL	Registro globale reale (non volatile)
0x0011	AM(n)	U32	Maschera allarmi
0x0012	AM(n).b	U32	Maschera allarmi, accesso a bit

Annotazioni:

- se la variabile è di tipo array, l'indice di array (n) può essere o una costante o un variabile non indicizzata.
- se la variabile prevede l'accesso a bit, .b può essere una costante o una variabile e deve essere nel range compatibile la grandezza della variabile di riferimento.

Variabili predefinite

Le variabili vengono identificate in base ad un codice ID univoco, siano esse predefinite o definite dall'utente: gli ID vengono suddivisi nel seguente modo:

Range	Descrizione
0x00000000-0x0000FFFF	Variabile predefinita
0x0001000-0xFFFFFFFF	Variabile definita dall'utente
0xFFFFFFFF	Riservato

ATTENZIONE: le variabili predefinite non vengono mai salvate nel file .LAD, ma si considerano sempre esistenti.

Le variabili predefinite sono i seguenti:

ID	Nome	Tipo	Descrizione
0x00000001	INP(n)	BOOL	Canale input
0x00000002	INP_W(n)	U16	Input word 16bit
0x00000003	INP_DW(n)	U32	Input word 32bit
0x00000004	OUT	BOOL	Canale output
0x00000005	OUT_W(n)	U16	Output word 16bit
0x00000006	OUT_DW(n)	U32	Output word 32bit
0x00000007	R(n)	I32	Registro globale intero (volatile)
0x00000008	NVR(n)	I32	Registro globale intero (non volatile)
0x00000009	RR(n)	REAL	Registro globale reale (volatile)

ID	Nome	Tipo	Descrizione
0x0000000A	NVRR(n)	REAL	Registro globale reale (non volatile)
0x0000000B	AM(n)	U32	Maschera allarmi

Annotazioni:

- se la variabile è di tipo array, l'indice di array (n) può essere o una costante o un variabile non indicizzata.
- RDE3 crea ulteriori variabili predefinite con assegnazione degli ID in modo del tutto arbitrario (nel range 0x00000100-0x00010000): RTE riconosce tali variabili in base al nome e non in base a tale ID.

Identificativi sorgente

Gli identificativi sorgente vengono utilizzati per identificare in modo univoco la sorgente di una certa entità (o elemento) all'interno del file ladder, come per esempio strutture, variabili o classi OB. Il valore dell'identificativo U32 viene composto come segue:

- U16 alta, identificativo gruppo;
- U16 bassa, identificativo elemento nel gruppo.

Gli identificativi sorgente previsti sono i seguenti:

Gruppo	Elemento	Descrizione
0x0000	0x0000	Entità definita manuale dall'utente
0x0001	0x0000	Entità importata da file esterno: file generico
0x0002	0x0000	Entità importata (forzata) da progetto RTE: sorgente generica
0x0002	0x0001	Entità importata (forzata) da progetto RTE: sorgente configuratore

Formato file (versione 1.5)

Formato file (versione 1.5)

Un programma in linguaggio di programmazione ladder/diagram è un singolo file (estensione .LAD) contenente sia le informazioni per l'editing (sorgente) che le informazioni per l'esecuzione (runtime).

Un programma ladder è composto principalmente da una [intestazione](#)^[95], quindi da varie sezioni tra cui le principali sono:

- [tabella strutture](#)^[98], ovvero i tipi di dati definiti dall'utente.
- [tabella variabili locali](#)^[100], ovvero le variabili generali del programma.
- [tabella classi OB](#)^[116], ovvero le classi object block a cui il programma fa riferimento.

- [tabella blocchi](#)^[103], ovvero le sub-routines in altri linguaggi di programmazione;
- [tabella modifiche live incorporate](#)^[122], ovvero eventuali file di modifica live incorporati;
- [tabella file inclusione](#)^[120], ovvero eventuali file esterni il cui contenuto viene incluso nel programma;
- [tabella stringhe](#)^[121];
- [tabella stringhe compresse](#)^[121].

Nell'elenco dei blocchi, esiste sempre un blocco principale detto MAIN ed eventualmente ulteriori blocchi (paragonabili a sub-routines di altri linguaggi di programmazione): ogni blocco contiene un [elenco di circuiti](#)^[104] che costituiscono il programma ladder effettivo: ogni circuito a sua volta è un elenco logicamente strutturato di [elementi](#)^[105].

NOTA: tutti i dati numerici memorizzati nel presente file sono da considerarsi in formato little indian (standard x86).

Intestazione file

Ad inizio file (offset 0) è presente una intestazione generale (HEADER) che contiene le informazioni di accesso a tutte le restanti componenti e sotto-sezioni del file.

Nome	Offset	Tipo	Descrizione
SIGN	+0	U32	Firma del file ('LADF').
VERSION	+4	U32	Versione del file (1.5.0).
GENERATOR	+8	U32	Nome e versione del generatore del file (offset relativo a tabella stringhe).
TYPE	+12	U16	Tipo di esecuzione: 0 = Esecuzione sincrona (SY). 1 = Esecuzione ad alta priorità (HP). 2 = Esecuzione a priorità normale (NP). 3 = Esecuzione a bassa priorità (LP).
FREQUENCY	+14	U16	Frequenza di esecuzione in [hz], da 1hz a 3000hz, default 200hz .
FLAGS	+16	U32	Impostazioni generali: 0x00000001 = Il file contiene errori di verifica formale. 0x00000002 = Il file contiene riferimenti a file .OBB esterni (classi OB)

Nome	Offset	Tipo	Descrizione
			0x00000004 = Il file utilizza codifica UTF-8
BUILDEAR	+20	U16	Anno ultima generazione (4 digit).
BUILDMONTH	+22	U8	Mese ultima generazione (1-12).
BUILDDAY	+23	U8	Giorno ultima generazione (1-31).
BUILDHOUR	+24	U8	Ultima ora generazione (0-23).
BUILDMINUTE	+25	U8	Minuto ultima compilazione (0-59).
BUILDSECOND	+26	U8	Secondo ultima compilazione (0-59).
TARGETID	+27	U8	ID destinazione runtime: 0x00 = Firmware RTE (PowerPC G2) 0x01 = Firmware RTE (PowerPC P2020) 0x02 = Firmware RTE (ARM Cortex A9)
VID	+28	U32	ID versione del file.
PLVID	+32	U32	ID versione del file pre modifiche live (se presenti, altrimenti vale 0).
IARECNT	+36	U32	N. di elementi tipo ARE presenti nel file.
IAFECNT	+40	U32	N. di elementi tipo AFE presenti nel file.
CCFLAGS	+44	U32	Impostazioni compilatore: 0x00000001 = Ignorare messaggi avviso file di inclusione: importazione variabili 0x00000002 = Ignorare messaggi avviso file di inclusione: importazione classi OB
	+48	U8[80]	(riservati)

Nome	Offset	Tipo	Descrizione
BLOCKSOFF	+128	U32	Offset della tabella dei blocchi ^[103] (relativo al file).
BLOCKSCNT	+132	U32	N. di elementi della tabella dei blocchi ^[103] .
STRINGSOFF	+136	U32	Offset della tabella stringhe ^[121] (relativo al file).
STRINGSSIZE	+140	U32	Dimensione della tabella stringhe ^[121] .
CSTRINGSOFF	+144	U32	Offset della tabella stringhe compresse ^[121] (relativo al file).
CSTRINGSSIZE	+148	U32	Dimensione della tabella stringhe compresse ^[121] .
CSTRINGSTYPE	+152	U8	Tipo di compressione usata per la tabella stringhe compresse ^[121] : 0 = Nessuna compressione. 1 = Compressione standard Qt 3.3.x.
	+153	U8[3]	(riservati)
EMBLIVESOFF	+156	U32	Offset della tabella modifiche live incorporate ^[122] (relativo al file) .
EMBLIVESCNT	+160	U32	N. di elementi della tabella modifiche live incorporate ^[122] .
STRUCTSOFF	+164	U32	Offset della tabella strutture ^[98] (relativo al file).
STRUCTSCNT	+168	U32	N. di elementi della tabella strutture ^[98] .
VARSOFF	+172	U32	Offset della tabella variabili ^[100] (relativo al file).
VARSCNT	+176	U32	N. di elementi della tabella variabili ^[100] .
OBCLASSESOFF	+180	U32	Offset della tabella classi OB ^[116] (relativo al

Nome	Offset	Tipo	Descrizione
			file)
OBCLASSESCNT	+184	U32	N. di elementi della tabella classi OB ^[116] (relativo al file)
LASTBLOCKID	+188	U32	Ultimo ID blocco utilizzato.
LASTRUNGID	+192	U32	Ultimo ID circuito utilizzato.
LASTSTRUCTID	+196	U32	Ultimo ID struttura utilizzato.
LASTVARID	+200	U32	Ultimo ID variabile utilizzato.
LASTOBCLASSID	+204	U32	Ultimo ID classe OB utilizzato
INCFILESOFF	+208	U32	Offset della tabella file inclusione ^[120] (relativo al file)
INCFILESCNT	+212	U32	N. di elementi della tabella file inclusione ^[120] .
LASTINCFILEID	+216	U32	Ultimo ID file inclusione utilizzato
	+220	U8[292]	(riservati)

Tabella strutture

La tabella strutture (STRUCTS) è composta da un numero di definizioni (STRUCT) contigue, ognuna delle quali contiene tutte le informazioni su uno specifica struttura definita dall'utente: in questa tabella non sono presenti i [tipi dati predefiniti](#)^[122].

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[95].STRUCTSOFF e [HEADER](#)^[95].STRUCTSCNT.

Definizione struttura

Una struttura (STRUCT) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID tipo struttura Nota: solo valori 0x00010000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati, uso runtime)

Nome	Offset	Tipo	Descrizione
FLAGS	+6	U16	Impostazioni della struttura: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo MODIFIER non compresso (NCF1).
NAME	+8	U32	Offset nome della struttura (relativo a stringhe ^[121]).
TEXT	+12	U32	Offset descrizione della struttura (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF0).
VARSOFF	+16	U32	Offset della tabella variabili ^[100] (relativo al file) (#1)
VARSCNT	+20	U32	Dimensione tabella variabili ^[100]
SIZE	+24	U32	Dimensione della struttura [byte]
SRCID	+28	U32	Identificativo sorgente ^[126] della struttura
OBCID	+32	U32	ID classe OB di riferimento (0=nessuna)(#2,#3)
ATTRIB	+36	U32	Attributi della struttura: 0x00000001 Struttura riferita a classe OB (AREFOBC)(#2) 0x00000002 Struttura principale classe OB (AISOBC)(#2,#3) 0x00000004 Struttura (rif.a) definita esternamente (AEXTERNAL)
MODIFIER	+40	U32	Offset modificatori della struttura (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF1).
REFID	+44	U32	ID di riferimento tipo struttura (#4).

Nome	Offset	Tipo	Descrizione
EXTNAME	+48	U32	Offset nome esterno struttura (relativo a stringhe ^[127]).
	+52	U8[16]	(riservati)

(#1) La definizione di struttura usa una normale tabella di definizione variabili per definire i propri elementi. In questo caso gli ID delle variabili non hanno senso e dovrebbero essere sempre a 0.

(#2) A pari valore di OBCID non 0, con attributo AREFOBC, può esistere una sola struttura con flag AISOBC, mentre posso essere 0 o più strutture senza attributo AISOBC.

(#3) Una struttura viene definita "orfana" qualora si trova nella condizione di attributi AREFOBC impostato e campo OBCID = 0: questa condizione capita quando aggiornando/rimuovendo una classe OB, vi sono ancora delle variabili (o altro) che vi fanno riferimento.

(#4) In caso di attributo AREFOBC il campo REFID contiene l'ID della struttura come definito del file OBB di origine (0 in case di attributo AISOBC).

Tabella variabili

La tabella variabili (VARS) è composta da un numero di definizioni (VAR) contigue, ognuna delle quali contiene tutte le informazioni su uno specifica variabile definita dall'utente: in questa tabella non sono presenti le [variabili predefinite](#)^[125].

La posizione (relativa al file) e il numero di elementi della tabella delle variabili di programma sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata

- dai campi [HEADER](#)^[95].VARSOFF e [HEADER](#)^[95].VARSCNT se utilizzati in intestazione file;
- dai campi [STRUCT](#)^[98].VARSOFF e [STRUCT](#)^[98].VARSCNT se utilizzati in definizione struttura;
- dai campi [LIVECMD](#)^[142].VARSOFF e [LIVECMD](#)^[142].VARSCNT se utilizzati in definizione comando.

Definizione variabile

Una variabile (VAR) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID variabile (#2) Nota: solo valori 0x00010000-0xFFFFFFFFE; il valore 0xFFFFFFFF è un ID a significato speciale; i valori 0x00000001-0x0000FFFF sono riservati per le variabili predefinite.
TYPEID	+4	U32	ID tipo variabile Nota: può essere un tipo predefinito ^[127] oppure definito dall'utente ^[98] .

Nome	Offset	Tipo	Descrizione
	+8	U8[2]	(riservati, uso runtime)
FLAGS	+10	U16	Impostazioni della variabile: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo BITTEXT non compresso (NCF1) 0x0004 Campo MODIFIER non compresso (NCF2)
NAME	+12	U32	Offset nome della variabile (relativo a stringhe ^[121]).
TEXT	+16	U32	Offset descrizione della variabile (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF0).
BITTEXT	+20	U32	Offset descrizioni dei bit variabili, ovvero sequenza di stringhe relative ad ogni singolo bit separate dal carattere ascii 0xFF (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF0). (#2)
DIM0	+24	U32	1a dimensione array (0=non specificata) (#1) o indice 0 per Alias ^[123] (TYPEID = 0x0000000F).
DIM1	+28	U32	2a dimensione array (0=non specificata) (#1) o indice 1 per Alias ^[123] (TYPEID = 0x0000000F).
DIM2	+32	U32	3a dimensione array (0=non specificata) (#1) o indice 2 per Alias ^[123] (TYPEID = 0x0000000F).
ALIASID	+36	U16	ID tipo alias ^[124] (#2)
ALIASBIT	+38	U16	N. di bit per alias (dipende dal tipo di ID alias)(#2)

Nome	Offset	Tipo	Descrizione
SIZE	+40	U32	Dimensione della variabile. (#3)
SRCID	+44	U32	Identificativo sorgente ^[126] della variabile. (#2, #4)
VALUE	+48	U32	Offset valore iniziale della variabile (relativo a stringhe ^[121] , 0=nessun valore).
ATTRIB	+52	U32	Attributi variabile: 0x00000001 La variabile può essere scritta (WRITE) 0x00000002 La variabile può essere letta (READ) 0x00000004 La variabile può essere forzata (FORCE) 0x00000008 La variabile può essere rilasciata (RELEASE) 0x00000010 La variabile è costante (CONST) 0x00000020 La variabile non ha accesso a bit (NOBITS)
MODIFIER	+56	U32	Offset modificatori della variabile (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF2).
DIM0L	+60	U32	Offset etichetta per 1a dimensione array (relativo a stringhe ^[121]). (#5)
DIM1L	+64	U32	Offset etichetta per 2a dimensione array (relativo a stringhe ^[121]). (#5)
DIM2L	+68	U32	Offset etichetta per 3a dimensione array (relativo a stringhe ^[121]). (#5)
	+72	U8[16]	(riservati)

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW la dimensione massima va comunque considerata solo a 16bit (ovvero massimo indice 0xFFFF).

(#2) Campo non utilizzato (valore 0) se utilizzata in definizione struttura

(#3) Campo non utilizzato (valore 0) se utilizzato definendo un Alias (TYPEID = 0x0000000F)

(#4) Campo non utilizzato (valore 0) se utilizzata in definizione comando live.

(#5) Per etichetta si intende nome di una variabile (a valore costante) oppure la rappresentazione ascii di un valore numerico, ovvero un offset alla [tabella stringhe](#)^[121].

Tabella blocchi

La tabella blocchi (BLOCKS) è composta da un numero di definizioni di blocco (BLOCK) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico blocco e del relativo contenuto su file.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[95].BLOCKSOFF e [HEADER](#)^[95].BLOCKSCNT.

Definizione blocco

Un blocco (BLOCK) è composto dai seguenti dati:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID univoco del blocco. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
STATUS	+4	U16	(riservato, uso runtime)
FLAGS	+6	U16	Impostazioni del blocco: 0x0001 = Blocco principale (ISMAIN). 0x0002 = Campo TEXT non compresso (NCF0). 0x0004 = Campo COMMENT non compresso (NCF1).
NAME	+8	U32	Offset nome del blocco (relativo a stringhe ^[121]). Nota: se flag ISMAIN specificato, il nome viene deve essere "main".
TEXT	+12	U32	Offset descrizione del blocco (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF0).
COMMENT	+16	U32	Offset commenti per il blocco (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF1).

Nome	Offset	Tipo	Descrizione
RUNGSOFF	+20	U32	Offset della tabella circuiti del blocco ^[104] (relativo a file).
RUNGSCNT	+24	U32	N. di elementi della tabella circuiti del blocco ^[104] .
	+28	B[36]	(riservati)

Tabella circuiti del blocco

La tabella circuiti del blocco (RUNGS) è composta da un numero di definizioni di circuito (RUNG) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico circuito e del relativo contenuto.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [BLOCK](#)^[103].RUNGSOFF e [BLOCK](#)^[103].RUNGSCNT.

Definizione circuito del blocco

Un circuito del blocco (RUNG) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID univoco del circuito. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati, uso runtime)
FLAGS	+6	U16	Impostazioni del circuito: 0x0001 Campo TEXT non compresso (NCF0).
REFID	+8	U32	ID circuito di riferimento (0=nessuno).
ITEMSOFF	+12	U32	Offset della tabella elementi del circuito ^[105] (relativo al file).
ITEMSSIZE	+16	U32	Dimensione della tabella elementi del circuito ^[105] .
TEXT	+20	U32	Offset descrizione del circuito (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF1).
	+24	U8[4]	(riservati)

Tabella elementi circuito

La tabella elementi del circuito (ITEMS) è composta da un numero di definizioni di circuito (ITEM) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico elemento.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata:

- dai campi [RUNG](#)^[104].ITEMSOFF e [RUNG](#)^[104].ITEMSSIZE, se utilizzato in circuito del blocco (RUNG).
- dai campi [LIVECMD](#)^[142].ITEMSOFF e [LIVECMD](#)^[142].ITEMSSIZE, se utilizzato in comandi modifiche live (LIVECMD).

Definizione elemento del circuito

Gli elementi del circuito (ITEM) hanno una parte iniziale comune, quindi un'area dati specifica per ogni tipo di elemento. La parte comune è la seguente:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	Codice tipo elemento.
SIZE	+2	U16	Dimensione completa dell'elemento (tutti i campi).
DATA	+4	...	Dati specifici (opzionali).

Gli elementi previsti sono i seguenti:

Simbolico	ID	Tipo	Dati	Descrizione
BB	0x0001			Inizio ramo.
NL	0x0002			Prossimo livello ramo . Note: deve sempre essere compreso tra un BB e un EB.
EB	0x0003			Fine ramo . Note: deve sempre corrispondere ad un BB.
OC	0x0004		U32=Destinazione (#3): solo tipi BOOL ^[123]	Contatto aperto.
CC	0x0005		U32=Destinazione (#3): solo tipi BOOL ^[123]	Contatto chiuso.
RE	0x0006		U32=Destinazione (#3): solo tipi EDGE ^[123]	Fronte di salita.

Simbolico	ID	Tipo	Dati	Descrizione
FE	0x0007		U32=Destinazione (#3): solo tipi EDGE _[123]	Fronte di discesa.
AF	0x0008			Sempre falso.
OUT	0x0009	F	U32=Destinazione (#3): solo tipi BOOL _[123]	Output.
OL	0x000A	F	U32=Destinazione (#3): solo tipi BOOL _[123]	Output latch (set).
OUL	0x000B	F	U32=Destinazione (#3): solo tipi BOOL _[123]	Output unlatch (reset).
NOUT	0x000C	F	U32=Destinazione (#3): solo tipi BOOL _[123]	Output negato
CLR	0x000D	F	U32=Destinazione (#3): solo tipi TIMER _[123] e COUNTER _[123]	Azzeramento (clear).
TP	0x000E	F	U32=Destinazione (#3): solo tipi TIMER _[123] U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer pulse.
TON	0x000F	F	U32=Destinazione (#3): solo tipi TIMER _[123] U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer on delay.
TOFF	0x0010	F	U32=Destinazione (#3): solo tipi TIMER _[123] U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer off delay.
TSQ	0x0011	F	U32=Destinazione (#3): solo tipi TIMER _[123]	Abilita timer onda quadra.

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	
TINT	0x0012	F	U32=Destinazione (#3): solo tipi TIMER _[123] U32=Valore di preset 1 (#4) U32=Valore di preset 2 (#4)	Abilita timer integrale.
CU	0x0013	F	U32=Destinazione (#3): solo tipi COUNTER _[123] U32=Valore di preset (#4)	Conteggio up.
CD	0x0014	F	U32=Destinazione (#3): solo tipi COUNTER _[123] U32=Valore di preset (#4)	Conteggio down.
CUR	0x0015	F	U32=Destinazione (#3): solo tipi COUNTER _[123] U32=Valore di preset (#4)	Conteggio up rotativo.
CDR	0x0016	F	U32=Destinazione (#3): solo tipi COUNTER _[123] U32=Valore di preset (#4)	Conteggio down rotativo.
LD	0x0017	F	U32=Destinazione (#3): solo tipi COUNTER _[123]	Preset su contatore.
MATH	0x0018	F	U32=Espressione (#5) U32=Destinazione (#3) U8[]=Dati espressione matematica _[113]	Blocco matematico.
EQ	0x0019		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto uguaglianza (A == B).

Simbolico	ID	Tipo	Dati	Descrizione
NE	0x001A		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto diverso (A != B).
GT	0x001B		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto maggiore (A > B).
GE	0x001C		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto maggiore/uguale (A >= B).
LT	0x001D		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto minore (A < B).
LE	0x001E		U32=Sorgente A (#2) U32=Sorgente B (#2)	Blocco confronto minore/uguale (A <= B).
COPY	0x001F	F	U32=Sorgente (#2) U32=Destinazione (#3) U32=N. Elementi (#2)	Copia dati da sorgente a destinazione, per il n. di elementi specificato.
CALL	0x0020	F	U32 ID blocco da chiamare	Chiamata di un blocco (sub-routine).
RET	0x0021	F		Ritorno da un blocco all'interno.
ADD	0x0022	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Somma (A + B).
SUB	0x0023	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Sottrazione (A - B).

Simbolico	ID	Tipo	Dati	Descrizione
MUL	0x0024	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Moltiplicazione ($A * B$).
DIV	0x0025	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Divisione (A / B).
AND	0x0026	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	AND binario ($A \& B$).
OR	0x0027	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	OR binario ($A B$).
NOT	0x0028	F	U32=Sorgente A (#2) U32=Destinazione (#3)	NOT binario ($\sim A$).
XOR	0x0029	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	XOR binario ($A \wedge B$).
SQRT	0x002A	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Radice quadrata di A.
MOD	0x002B	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Modulo ($A \bmod B$).

Simbolico	ID	Tipo	Dati	Descrizione
POW	0x002C	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Elevamento a potenza (A pow B).
SIN	0x002D	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Seno di A.
COS	0x002E	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Coseno di A.
TAN	0x002F	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Tangente di A.
ASIN	0x0030	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arcoseno di A.
ACOS	0x0031	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arcocoseno di A.
ATAN	0x0032	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arcotangente di A.
ATAN2	0x0033	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Arcotangente di A, B.
LOG	0x0034	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Logaritmo base 10 di A.
LN	0x0035	F	U32=Sorgente A (#2)	Logaritmo base E di A.

Simbolico	ID	Tipo	Dati	Descrizione
			U32=Destinazione (#3)	
ABS	0x0036	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Valore assoluto di A
SIGN	0x0037	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Segno di A.
ROUND	0x0038	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Arrotondamento all'intero più vicino ad A.
MIN	0x0039	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Minimo tra A e B.
MAX	0x003A	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Massimo tra A e B.
RANGE	0x003B	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Sorgente C (#2) U32=Destinazione (#3): solo tipi BOOL ₁₂₃	Verifica se A contenuto nel range B-C.
LIMIT	0x003C	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Sorgente C (#2) U32=Destinazione (#3)	Limita A al range B-C.

Simbolico	ID	Tipo	Dati	Descrizione
ROR	0x003D	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Rotazione bit a destra (A >> B).
ROL	0x003E	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Rotazione bit a sinistra (A << B).
SHR	0x003F	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Shift bit a destra (A >> B).
SHL	0x0040	F	U32=Sorgente A (#2) U32=Sorgente B (#2) U32=Destinazione (#3)	Shift bit a sinistra (A << B).
MOV	0x0041	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Assegna A.
TODEG	0x0042	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Conversione di A da RAD a DEG.
TORAD	0x0043	F	U32=Sorgente A (#2) U32=Destinazione (#3)	Conversione di A da DEG a RAD.
NEG	0x0044			Negazione stato logico.
NOP	0x0045	F		Nessuna operazione.
EVAL	0x0046		U32=Espressione (#5)	Valutazione espressione (booleana).

Simbolico	ID	Tipo	Dati	Descrizione
			U8[]=Dati espressione matematica ^[113]	
OBCALL	0x0047	F	U32=Nome istanza (#6) U32=Nome completo metodo (#9) U32=Elenco nomi parametro (#7) U32=Elenco valori parametro (#7) U32=Risultato (#3) U8[]=Dati espressione matematica ^[113] (#8)	Chiamata a metodo di una istanza OB.
FT	0x0048			Esecuzione iniziale
ARE	0x0049			Fronte di salita (anonimo)
AFE	0x004A			Fronte di discesa (anonimo)

(#1) Variabile a bit locale (tipo I32 o U32) con indirizzamento diretto (eventuali indici e numero di bit costanti).

(#2) Per sorgente/destinazione si intende nome di una variabile oppure la rappresentazione ASCII di un valore numerico (solo sorgente), ovvero un offset alla [tabella stringhe](#)^[121].

(#3) Per destinazione si intende nome di una variabile con offset alla [tabella stringhe](#)^[121].

(#4) Per valore (o valore di preset) si intende un valore numerico in formato stringa: stringa vuota (o offset 0) si intende nessun valore assegnato.

(#5) Espressione intesa come stringa con offset alla [tabella stringhe](#)^[121].

(#6) Per nome istanza si intende nome di una variabile con offset alla [tabella stringhe](#)^[121], il cui TYPEID sia riferito ad una struttura utente con flag FISOBC.

(#7) Per elenco si intende una stringa (con offset alla [tabella stringhe](#)^[121]), composta da un certo numero di sotto-stringhe separate con carattere ASCII 0xFF: i due elenchi (nomi/valori) DEVONO avere sempre lo stesso n. di elementi o essere entrambe vuote (=0).

(#8) La chiamata ad un metodo di una istanza OB viene effettivamente tradotta come una espressione matematica, in cui vengono preparati in stack gli eventuali parametri (valori in ordine naturale), viene effettuata la chiamata al metodo e quindi ritrovato l'eventuale risultato da stack.

(#9) Nome completo del metodo (univoco), inteso come stringa con offset alla [tabella stringhe](#)^[121].

Dati espressione matematica

I dati dell'espressione matematica (usati dagli elementi MATH, COND e OBCALL), sono una serie di operazioni contigue (ordinati in polacca inversa) con la seguente intestazione comune:

Nome	Offset	Tipo	Descrizione
CODE	+0	U8	Codice operazione.
DATA	+1	...	Dati specifici (opzionali).

Le operazioni previste sono le seguenti:

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
PUSH	0x01	DBL=valore	--	DBL	Push valore in stack.
PUSHV	0x02	U32=Sorgente (#1)	--	DBL	Push valore variabile in stack.
ADD	0x03		DBL0,DBL1	DBL	Somma.
SUB	0x04		DBL0,DBL1	DBL	Sottrazione (#2).
MUL	0x05		DBL0,DBL1	DBL	Moltiplicazione (#2).
DIV	0x06		DBL0,DBL1	DBL	Divisione (#2).
MOD	0x07		DBL	DBL	Modulo (#3).
ABS	0x08		DBL	DBL	Valore assoluto (#3).
SIN	0x09		DBL	DBL	Seno.
COS	0x0A		DBL	DBL	Coseno.
TAN	0x0B		DBL	DBL	Tangente.
ASIN	0x0C		DBL	DBL	Arcoseno.
ACOS	0x0D		DBL	DBL	Arcocoseno.
ATAN	0x0E		DBL	DBL	Arcotangente.
POW	0x0F		DBL	DBL	Esponente.
LOG10	0x10		DBL	DBL	Logaritmo base 10.
LOG	0x11		DBL	DBL	Logaritmo naturale.
R_AND	0x12		U32	U32	AND binario.
R_OR	0x13		U32	U32	OR binario.
R_NOT	0x14		U32	U32	NOT binario.

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
R_XOR	0x15		U32	U32	XOR binario.
POPV	0x16	U32=Destinazione (#1)	DBL	--	Pop valore da stack a variabile.
R_ROR	0x17		U32	U32	Rotazione bit a destra.
R_ROL	0x18		U32	U32	Rotazione bit a sinistra.
R_SHR	0x19		U32	U32	Shift bit a destra.
R_SHL	0x1A		U32	U32	Shift bit a sinistra.
TOBIN	0x1B		U32	U32	Conversione da BCD a binario.
TOBCD	0x1C		U32	U32	Conversione da binario a BCD.
TODEG	0x1D		DBL	DBL	Conversione da radianti a gradi.
TORAD	0x1E		DBL	DBL	Conversione da gradi a radianti.
ATAN2	0x1F		DBL0,DBL1	DBL	Arcotangente con due parametri.
PUSHVB	0x20	U32=Sorgente (#1)	DBL	--	Push valore variabile a bit in stack.
POPVB	0x21	U32=Destinazione (#1)	DBL	--	Pop valore da stack a variabile bit.
POPL	0x22		BOOL	--	Pop valore logico (0=falso, altrimenti vero).
	0x23				(riservato)
EQ	0x24		DBL0,DBL1	BOOL	Confronto: uguale.
NE	0x25		DBL0,DBL1	BOOL	Confronto: differente.

Simbolico	ID	Dati	In (stack)	Out (stack)	Descrizione
LT	0x26		DBL0,DBL1	BOOL	Confronto: minore.
LE	0x27		DBL0,DBL1	BOOL	Confronto: minore o uguale.
GT	0x28		DBL0,DBL1	BOOL	Confronto: maggiore.
GE	0x29		DBL0,DBL1	BOOL	Confronto: maggiore o uguale.
AND	0x2A		BOOL,BOOL	BOOL	AND logico.
OR	0x2B		BOOL,BOOL	BOOL	OR logico.
NOT	0x2C		BOOL	BOOL	NOT logico.
XOR	0x2D		BOOL,BOOL	BOOL	XOR logico.
CHS	0x2E		DBL	DBL	Cambio segno.
SHR	0x2F		U32	U32	Shift logico a destra.
SHL	0x30		U32	U32	Shift logico a sinistra
OBCALL	0x31	U32=Metodo (#2)	?	?	Chiamata diretta a metodo per istanza OB
POP	0x32		*	--	Pop valore (skip)
PUSHP	0x33	U32=Sorgente (#1)	U32	U32	Push puntatore a variabile

(#1) Per sorgente/destinazione si intende nome di una variabile, ovvero un offset alla [tabella stringhe](#)^[121].

(#2) Per metodo si intende stringa, ovvero un offset alla [tabella stringhe](#)^[121], con formato "`<nome_istanza>.<hexid_metodo>`" oppure "`<nome_istanza>[<indice>].<hexid_metodo>`" (hexid = numero 32bit esadecimale ascii, senza prefisso 0x. Esempio metodo n. 0xFA9 dell'istanza OB 'myob' : "myob#00000FA9").

Tabella classi OB

La tabella classi OB (OBCLASSES) è composta da un numero di definizioni (OBCLASS) contigue, ognuna delle quali contiene tutte le informazioni su una specifica classe object block definita dall'utente.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#)^[95].OBCLASSOFF e [HEADER](#)^[95].OBCLASSCNT.

Definizione classe OB

Una classe OB (OBCLASS) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID classe OB Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati)
FLAGS	+6	U16	Impostazioni della classe OB: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo MODIFIER non compresso (NCF1).
NAME	+8	U32	Offset nome della classe OB (relativo a stringhe ^[121]).
TEXT	+12	U32	Offset descrizione della classe OB (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF0).
VERSION	+16	U32	Versione classe OB (formato nvMake).
INTFID	+20	U32	Identificativo univoco interfaccia obeit block.
PATHID	+24	U32	Identificativo percorso del file .OBB: 0=Percorso assoluto 1=(riservato) 2=Percorso relativo a workspace 3=Percorso relativo a factory RDE3
FILENAME	+28	U32	Offset nome del file .OBB (relativo a stringhe ^[121])(#1).
SRCID	+32	U32	Identificativo sorgente ^[126] della classe OB.
METHODSOFF	+36	U32	Offset della tabella metodi OB ^[118] (relativo al file)

Nome	Offset	Tipo	Descrizione
METHODSCNT	+40	U32	Dimensione tabella metodi OB ^[118]
ATTRIB	+44	U32	Attributi classe OB.
MODIFIER	+48	U32	Offset modificatori della classe OB (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF1).
	+52	U8[16]	(riservati)

(#1) In caso di PATHID diverso da 0, il nome del file .OB non presenta percorsi al suo interno, ma è semplicemente il nome del file con relativa al percorso PATHID; in caso invece di PATHID uguale a 0, il nome del file .OB potrebbe contenere un percorso relativo/assoluto.

Tabella metodi OB

La tabella metodi OB (OBMETHODS) è composta da un numero di definizioni (OBMETHOD) contigue, ognuna delle quali contiene tutte le informazioni su una specifico metodo OB relativo alla classe OB di appartenenza..

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [OBCLASS](#)^[117].METHODSOFF e [OBCLASS](#)^[117].METHODSCNT.

Definizione metodo OB

Una metodo OB (OBMETHOD) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID metodo OB. Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
	+4	U8[2]	(riservati)
FLAGS	+6	U16	Impostazioni del metodo OB: 0x0001 Campo TEXT non compresso (NCF0). 0x0002 Campo MODIFIER non compresso (NCF1).
NAME	+8	U32	Offset nome del metodo OB (relativo a stringhe ^[121]).
KNAME	+12	U32	Offset nome chiave del metodo OB (relativo a stringhe ^[121]).
TEXT	+16	U32	Offset descrizione del metodo OB (relativo a

Nome	Offset	Tipo	Descrizione
			stringhe compresse ^[121] , o stringhe ^[121] se NCF0).
CALLID	+20	U32	ID metodo in chiamata (#1)
PARAMSOFF	+24	U32	Offset della tabella parametri per metodo OB ^[119] (relativo al file).
PARAMSCNT	+28	U32	Dimensione della tabella parametri per metodo OB ^[119] .
RETTYEID	+32	U32	ID tipo variabile di ritorno. Nota: può essere un tipo predefinito ^[122] oppure definito dall'utente ^[98] .
ATTRIB	+36	U32	Attributi metodo.
MODIFIER	+40	U32	Offset modificatori del metodo OB (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF1).
	+44	U8[16]	(riservati)

(#1) Il valore del campo CALLID deriva dal valore del campo METHOD.ID, nel file .OBB di origine.

Tabella parametri per metodo OB

La tabella parametri per metodo OB (OBMETHODPS) è composta da un numero di definizioni (OBMETHODP) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico parametro relativo al metodo OB di appartenenza.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [OBMETHOD](#)^[118].PARAMSOFF e [OBMETHOD](#)^[118].PARAMSCNT.

Definizione parametro per metodo OB

Un parametro per metodo OB (OBMETHODP) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
TYPEID	+0	U32	ID tipo variabile Nota: può essere un tipo predefinito ^[122] oppure definito dall'utente ^[98] .
ATTRIB	+4	U16	Attributi parametro: 0x0001 Parametro passato per riferimento (ABYREF)

Nome	Offset	Tipo	Descrizione
			0x0002 Parametro passato per puntatore (ABYPTR)
FLAGS	+6	U16	Impostazioni del parametro: 0x0001 Campo TEXT non compresso (NCF0).
NAME	+8	U32	Offset nome del parametro (relativo a stringhe ^[121]).
TEXT	+12	U32	Offset descrizione del parametro (relativo a stringhe compresse ^[121] , o stringhe ^[121] se NCF0).
DIM0	+16	U32	1a dimensione array (0=non specificata) (#1).
DIM1	+20	U32	2a dimensione array (0=non specificata) (#1).
DIM2	+24	U32	3a dimensione array (0=non specificata) (#1).
	+28	B[16]	(riservati)

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW la dimensione massima va comunque considerata solo a 16bit (ovvero massimo indice 0xFFFF).

Tabella file inclusione

La tabella file inclusione (INCFILES) è composta da un numero di definizioni (INCFILE) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico file di inclusione definito dall'utente.

La posizione (relativa al file) e il numero di elementi della tabella file inclusione sono specificate rispettivamente dai campi [HEADER](#)^[95].INCFILESOFF e [HEADER](#)^[95].INCFILESCNT.

Definizione file inclusione

Un file inclusione (INCFILE) è composto dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
ID	+0	U32	ID file Nota: i valori 0 e 0xFFFFFFFF sono ID a significato speciale.

Nome	Offset	Tipo	Descrizione
FLAGS	+4	U32	Impostazioni file: (nessuna)
PATHID	+8	U16	ID percorso sorgente: 0 = (riservato percorso assoluto; non implementato) 1 = Percorso relativo a file .LAD contenitore
	+10	U8[2]	(riservati)
FILENAME	+12	U32	Offset nome del file (relativo a stringhe ^[12]).
	+16	U8[16]	(riservati)

Tabella stringhe

La tabella stringhe (STRINGS) è composta da un certo numero di stringhe ascii consecutive, ognuna terminata il carattere ascii \0 (zero) e a dimensione variabile. La tabella deve contenere almeno un carattere e deve essere \0, di modo che le stringhe con offset 0 puntino effettivamente ad una stringa nulla.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente nei seguenti campi, a seconda di dove utilizzata:

- dai campi [HEADER](#)^[95].STRINGSOFF e [HEADER](#)^[95].STRINGSSIZE, se usata in file ladder/diagram.
- dai campi [LIVEHEADER](#)^[141].STRINGSOFF e [LIVEHEADER](#)^[141].STRINGSSIZE, se usata in modifiche live.

Tabella stringhe compresse

La tabella stringhe compresse (CSTRINGS) è composta da un certo numero di stringhe ascii consecutive, ognuna terminata il carattere ascii \0 (zero) e a dimensione variabile.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente dai campi [HEADER](#)^[95].STRINGSOFF e [HEADER](#)^[95].STRINGSSIZE: lo stato di compressione, che è opzionale, è invece specificato dal campo [HEADER](#)^[95].CSTRINGSTYPE.

Annotazioni:

- in caso di compressione [HEADER](#)^[95].CSTRINGSTYPE = 1, utilizzare le funzioni standard di libreria (QT >= 3.3.4) `qUncompress()` per decomprimere la tabella in modo corretto.
- gli offset che fanno riferimento a questa tabella si intendono sempre riferiti alla tabella non compressa.

Tabella modifiche live

La tabella di modifiche live incorporate (EMBLIVES) è composta da un numero di definizioni (EMBLIVE) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico insieme di modifiche live al file ladder stesso: ogni insieme è in realtà un file di [modifiche live](#) ^[133] indipendente ma incorporato nel file ladder stesso.

La posizione (relativa al file) e il numero di elementi della tabella sono specificate rispettivamente dai campi [HEADER](#) ^[95].EMBLIVESOFF e [HEADER](#) ^[95].EMBLIVESCNT.

In fase di caricamento, i vari file di modifiche live vengono applicati nell'esatto ordine in cui compaiono. L'applicazione effettiva delle modifiche è regolata dall'apposito flag in [EMBLIVE](#) ^[122].FLAGS (valore 0x0001): solo l'ultimo file di modifica live può avere eventualmente questo flag a falso, indicando delle modifiche live non ancora applicate.

Definizione modifica live incorporata

Una modifica live incorporata (EMBLIVE) è composta dalle seguenti informazioni:

Nome	Offset	Tipo	Descrizione
FILEOFF	+0	U32	Offset del file di modifiche live ^[148] (offset relativo al file ladder).
FILESIZE	+4	U32	Dimensione del file di modifiche live ^[148] .
FLAGS	+8	U16	Impostazioni del file: 0x0001 = Modifiche effettive da applicare.
	+10	U8[22]	(riservati)

Tipi di dati predefiniti

I tipi di dati vengono identificati in base ad un codice ID univoco, siano essi predefiniti oppure definiti dall'utente: gli ID vengono suddivisi nel seguente modo:

Range	Descrizione
0x00000000-0x0000FFFF	Tipi dati predefiniti (#1)
0x00010000-0xFFFFFFFF	Tipi dati definiti dall'utente
0xFFFFFFFF	Riservato

(#1) i tipi di dati predefiniti non vengono mai salvati nel file .LAD, ma si considerano sempre esistenti.

I tipi di dati predefiniti sono i seguenti:

ID	Simbolico	Descrizione	RTE (0x00)
0x00000000	(riservato)	Tipo dato non valido	
0x00000001	I8	Intero 8bit con segno	1
0x00000002	U8	Intero 8bit senza segno	1

ID	Simbolico	Descrizione	RTE (0x00)
0x00000003	I16	Intero 16bit con segno	2
0x00000004	U16	Intero 16bit senza segno	2
0x00000005	I32	Intero 32bit con segno	4
0x00000006	U32	Intero 32bit senza segno	4
0x00000007	I64	Intero 64bit con segno	
0x00000008	U64	Intero 64bit senza segno	
0x00000009	STRING	Stringa (0 terminated)	...
0x0000000A	FLOAT	Floating point 32bit	4
0x0000000B	REAL	Floating point 64bit (double)	8
0x0000000C	BOOL	Booleano	4 (#1)
0x0000000D	TIMER	Struttura predefinita timer ^[124]	32
0x0000000E	COUNTER	Struttura predefinita contatore ^[124]	48
0x0000000F	ALIAS	Alias (o literally) ^[123]	0
0x00000010	EDGE	Storage speciale per elementi fronte di salita (RE) e fronte di discesa (FE)	4
0x00000011	POWERSET	Variabile logica "Power Set"	4 (#2)
0x00000012	AXESGROUP	Variabile logica "Gruppo assi"	4 (#2)
0x00000013	PATH	Variabile logica "Path"	4 (#2)
0x00000014	POINT_L	Variabile logica "Punto di libreria"	4 (#2)

(#1) il tipo BOOL in RTE viene rappresentato, per motivi di performance, come I32, mentre da RDE3 viene visto come U8.

(#2) Le variabili logiche in RTE sono sempre rappresentate come un puntatore in memoria (4bytes), anche qualora fossero incorporate in strutture definite dall'utente. Per ulteriori informazioni sulla composizione (elementi) di tali variabili, consultare la documentazione tecnica di RTE).

La rappresentazione e la dimensione dei tipi di dati dipendono direttamente dall'ID destinazione runtime ([HEADER](#)^[95].TARGETID). Le dimensioni dei tipi dati sono elencati nelle colonne RTE e sono espressi in bytes.

Struttura predefinita timer

La struttura predefinita TIMER è composta dai seguenti campi:

Tipo	Nome	Descrizione	Letture	Scrittura
BOOL	Q	Esito del timer	X	
REAL	ET	Valore attuale [ms]	X	X (#1)
REAL	PT	Valore di preset [ms]	X	X
REAL	PT2	Valore di preset 2 [ms]	X	X
BOOL	EN	Abilitazione timer	X	

(#1) Il campo ET può essere scritto solo quando il timer non è ancora stato assegnato.

Struttura predefinita contatore

La struttura predefinita COUNTER è composta dai seguenti campi:

Tipo	Nome	Descrizione	Letture	Scrittura
BOOL	QU	Esito del conteggio rispetto a preset	X	
BOOL	QD	Esito del conteggio rispetto a 0	X	
I32	CV	Valore attuale di conteggio	X	X (#1)
I32	PV	Valore di preset	X	X
BOOL	CU	Stato del segnale "count up"	X	
BOOL	CD	Stato del segnale "count down"	X	

(#1) Il campo CV può essere scritto solo quando il contatore non è ancora stato assegnato.

Alias (o literally)

L'elenco dei tipi dati utilizzabili come alias sono i seguenti:

ID	Nome	Tipo	Descrizione
0x0001	INP(n)	BOOL	Canale input
0x0002	INP_W(n)	U16	Input word 16bit
0x0003	INP_W(n).b	BOOL	Input word 16bit, accesso a bit

ID	Nome	Tipo	Descrizione
0x0004	INP_DW(n)	U32	Input word 32bit
0x0005	INP_DW(n).b	BOOL	Input word 32bit, accesso a bit
0x0006	OUT	BOOL	Canale output
0x0007	OUT_W(n)	U16	Output word 16bit
0x0008	OUT_W(n).b	BOOL	Output word 16bit, accesso a bit
0x0009	OUT_DW(n)	U32	Output word 32bit
0x000A	OUT_DW(n).b	BOOL	Output word 32bit, accesso a bit
0x000B	R(n)	I32	Registro globale intero (volatile)
0x000C	R(n).b	BOOL	Registro globale intero (volatile), accesso a bit
0x000D	NVR(n)	I32	Registro globale intero (non volatile)
0x000E	NVR(n).b	BOOL	Registro globale intero (non volatile), accesso a bit
0x000F	RR(n)	REAL	Registro globale reale (volatile)
0x0010	NVRR(n)	REAL	Registro globale reale (non volatile)
0x0011	AM(n)	U32	Maschera allarmi
0x0012	AM(n).b	U32	Maschera allarmi, accesso a bit

Annotazioni:

- se la variabile è di tipo array, l'indice di array (n) può essere o una costante o un variabile non indicizzata.
- se la variabile prevede l'accesso a bit, .b può essere una costante o una variabile e deve essere nel range compatibile la grandezza della variabile di riferimento.

Variabili predefinite

Le variabili vengono identificate in base ad un codice ID univoco, siano esse predefinite o definite dall'utente: gli ID vengono suddivisi nel seguente modo:

Range	Descrizione
0x00000000-0x0000FFFF	Variabile predefinita

Range	Descrizione
0x0001000-0xFFFFFFFFE	Variabile definita dall'utente
0xFFFFFFFFF	Riservato

ATTENZIONE: le variabili predefinite non vengono mai salvate nel file .LAD, ma si considerano sempre esistenti.

Le variabili predefinite sono i seguenti:

ID	Nome	Tipo	Descrizione
0x00000001	INP(n)	BOOL	Canale input
0x00000002	INP_W(n)	U16	Input word 16bit
0x00000003	INP_DW(n)	U32	Input word 32bit
0x00000004	OUT	BOOL	Canale output
0x00000005	OUT_W(n)	U16	Output word 16bit
0x00000006	OUT_DW(n)	U32	Output word 32bit
0x00000007	R(n)	I32	Registro globale intero (volatile)
0x00000008	NVR(n)	I32	Registro globale intero (non volatile)
0x00000009	RR(n)	REAL	Registro globale reale (volatile)
0x0000000A	NVRR(n)	REAL	Registro globale reale (non volatile)
0x0000000B	AM(n)	U32	Maschera allarmi

Annotazioni:

- se la variabile è di tipo array, l'indice di array (n) può essere o una costante o un variabile non indicizzata.
- RDE3 crea ulteriori variabili predefinite con assegnazione degli ID in modo del tutto arbitrario (nel range 0x00000100-0x00010000): RTE riconosce tali variabili in base al nome e non in base a tale ID.

Identificativi sorgente

Gli identificativi sorgente vengono utilizzati per identificare in modo univoco la sorgente di una certa entità (o elemento) all'interno del file ladder, come per esempio strutture, variabili o classi OB. Il valore dell'identificativo U32 viene composto come segue:

- U16 alta, identificativo gruppo;
- U16 bassa, identificativo elemento nel gruppo.

Gli identificativi sorgente previsti sono i seguenti:

Gruppo	Elemento	Descrizione
0x0000	0x0000	Entità definita manuale dall'utente
0x0001	0x0000	Entità importata da file esterno: file generico
0x0002	0x0000	Entità importata (forzata) da progetto RTE: sorgente generica
0x0002	0x0001	Entità importata (forzata) da progetto RTE: sorgente configuratore
0x0003	0x0000	Entità importata da file include: file generico

Formato modifiche live (versione 1.1)

Formato modifiche live (versione 1.1)

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato delle modifiche live (versione 1.1).

Le modifiche live sono sostanzialmente un file contenente un elenco di comandi per modificare, in una certa sequenza, un programma ladder/diagram: tali modifiche normalmente sono incorporate nel file ladder principale oppure sono trasmesse da RDE3 al controllo numerico collegato in fase di caricamento/collaudo delle modifiche live stesse.

Le modifiche live sono composte principalmente da una [intestazione](#)^[127] quindi da varie sezioni tra cui le principali sono:

- [tabella comandi live](#)^[128];
- [tabella stringhe](#)^[26].

NOTA: tutti i dati numerici memorizzati nel presente file sono da considerarsi in formato little indian (standard x86).

Intestazione modifiche live

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato delle modifiche live (versione 1.1).

Ad inizio file (offset 0) è presente una intestazione generale (LIVEHEADER) che contiene le informazioni di accesso a tutte le restanti componenti e sotto-sezioni del file.

Nome	Offset	Tipo	Descrizione
VERSION	+0	U32	Versione del file (1.1.0).
VID	+4	U32	ID versione del file
PLVID	+8	U32	ID versione del file prima delle modifiche live. Se le modifiche sono effettive da

Nome	Offset	Tipo	Descrizione
			applicare, questo campo deve valere 0.
	+12	U32	(riservato)
FLAGS	+16	U32	Impostazioni file: 0x00000001 = Il file contiene errori di verifica
LIVEFLAGS	+20	U32	Impostazioni modifiche live: (nessuna)
STRINGSOFF	+24	U32	Offset della tabella stringhe ^[26] (relativo al file).
STRINGSSIZE	+28	U32	Dimensione della tabella stringhe ^[26] .
LIVECMDSOFF	+32	U32	Offset della tabella comandi live ^[128] (relativo al file).
LIVECMDSSIZE	+36	U32	Dimensione della tabella comandi live ^[128] .
LASTBLOCKID	+40	U32	Ultimo ID blocco utilizzato.
LASTRUNGID	+44	U32	Ultimo ID circuito utilizzato.
LASTSTRUCTID	+48	U32	Ultimo ID struttura utilizzato.
LASTVARID	+52	U32	Ultimo ID variabile utilizzato.
	+56	U8[20]	(riservati)

Tabella comandi live

ATTENZIONE: le presenti informazioni fanno riferimento ad una precedente versione del formato delle modifiche live (versione 1.1).

La tabella comandi live (LIVECMDS) è composta da un numero di definizioni (LIVECMD) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico comando live.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente dai campi [LIVEHEADER](#)^[127].ITEMSOFF e [LIVEHEADER](#)^[127].ITEMSSIZE.

Definizione comando live

I comandi live (LIVECMD) hanno una parte iniziale comune, quindi un'area dati specifica per ogni tipo di elemento. La parte comune è la seguente:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	Codice del comando
SIZE	+2	U32	Dimensione completa del comando (tutti i campi).
DATA	+6	...	Dati specifici del comando (opzionali).

I comandi attualmente previsti sono:

- [Comando 0x0001 - Inserimento circuito](#) ¹²⁹
- [Comando 0x0002 - Modifica circuito](#) ¹³⁰
- [Comando 0x0003 - Rimozione circuito](#) ¹³¹
- [Comando 0x0004 - Inserimento struttura](#) ¹³¹
- [Comando 0x0005 - Inserimento variabile](#) ¹³²

Comando 0x0001 - Inserimento circuito

Questo comando permette di inserire un circuito in un dato blocco, in una certa posizione. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0001
SIZE	+2	U32	32 + ITEMSSIZE
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito interessato (và in stato Inserting)
REFID	+14	U32	ID del circuito di riferimento (solo per locazioni inserimento #1, altrimenti 0)
IFLAGS	+18	U16	Impostazioni di inserimento: 0x000F = Locazione di inserimento: 0x0 = Prima del circuito con ID REFID (#1) 0x1 = Dopo il circuito con ID REFID (#1) 0x2 = Come primo circuito del blocco 0x3 = Come ultimo circuito del blocco

Nome	Offset	Tipo	Descrizione
FLAGS	+20	U16	Impostazioni del circuito: 0x0001 (ignorato in modifiche live)
ITEMSOFF	+22	U32	Offset della tabella elementi circuito ^[15] (relativo al file)
ITEMSSIZE	+26	U32	Dimensione della tabella elementi circuito ^[15]
TEXT	+30	U32	Offset descrizione del circuito ((relativo a stringhe) ^[26])

ATTENZIONE: la tabella elementi circuito deve comunque essere accodata al comando live stesso.

Comando 0x0002 - Modifica circuito

Questo comando permette di inserire un circuito modificato, rimuovendo quello vecchio, in un dato blocco. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0002
SIZE	+2	U32	32 + ITEMSSIZE
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito interessato (va in stato Inserting)
REFID	+14	U32	ID del circuito precedente da rimuovere (va in stato Dropping)
	+18	U16	(riservato)
FLAGS	+20	U16	Impostazioni del circuito: 0x0001 (ignorato in modifiche live)
ITEMSOFF	+22	U32	Offset della tabella elementi circuito ^[15] (relativo al file)
ITEMSSIZE	+26	U32	Dimensione della tabella elementi circuito ^[15]

Nome	Offset	Tipo	Descrizione
TEXT	+30	U32	Offset descrizione del circuito (relativo a stringhe _{26^h})

Comando 0x0003 - Rimozione circuito

Questo comando permette di rimuovere un circuito esistente. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0003
SIZE	+2	U32	14
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito da rimuovere (va in stato Deleting)

Comando 0x0004 - Inserimento struttura

Questo comando permette di creare una nuova struttura. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0004
SIZE	+2	U32	32 + (VARSCNT * sizeof (VAR _{11^h}))
TYPEID	+6	U32	ID tipo struttura utente Nota: solo valori 0x0001000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
FLAGS	+10	U16	Impostazioni struttura: 0x0001 (ignorato in modifiche live)
NAME	+12	U32	Offset nome della struttura (relativo a stringhe _{26^h})
TEXT	+16	U32	Offset descrizione della struttura (relativo a stringhe _{26^h})
VARSOFF	+20	U32	Offset della tabella variabili _{11^h} (relativa al file) (#1)

Nome	Offset	Tipo	Descrizione
VARSCNT	+24	U32	Dimensione della tabella variabili ^[11]
SIZE	+28	U32	Dimensione della struttura [byte]

(#1) La definizione di struttura usa una normale tabella di definizione variabili per definire i propri elementi. In questo caso gli ID delle variabili non hanno senso e dovrebbero essere sempre a 0.

Comando 0x0005 - Inserimento variabile

Questo comando permette di creare una nuova variabile. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0005
SIZE	+2	U32	48
VARID		U32	ID univoco della variabile utente Nota: solo valori 0x0001000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
TYPEID		U32	ID tipo dato variabile Nota: può essere un tipo predefinito ^[28] oppure definito dall'utente ^[10] .
FLAGS		U16	Impostazioni della variabile: 0x0001 (ignorato in modifiche live) 0x0002 (ignorato in modifiche live)
NAME		U32	Offset nome della variabile (relativo a stringhe ^[26])
TEXT		U32	Offset descrizione della variabile (relativo a stringhe ^[26])
BITTEXT		U32	Offset descrizioni dei bit variabili, ovvero sequenza di stringhe relative ad ogni singolo bit separate dal carattere ascii 0xFF (relativo a stringhe ^[26]).

Nome	Offset	Tipo	Descrizione
ARRAY0		U32	N. di indici 1o livello (0=Nessun array) (#1) o indice 0 per Alias ^[29] .
ARRAY1		U32	N. di indici 2o livello (0=Nessun array) (#1) o indice 1 per Alias ^[29] .
ARRAY2		U32	N. di indici 3o livello (0=Nessun array) (#1) o indice 2 per Alias ^[29] .
ALIASID		U16	ID tipo alias ^[30]
ALIASBIT		U16	N. di bit per alias (dipende dal tipo di ID alias)
SIZE		U32	Dimensione della variabile. Se Alias ^[29] , la dimensione è 0.

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW l'indice massimo comunque va considerato solo a 16bit (ovvero massimo indice 0xFFFF).

Formato modifiche live (versione 1.3)

Formato modifiche live (versione 1.3)

Le modifiche live sono sostanzialmente un file contenente un elenco di comandi per modificare, in una certa sequenza, un programma ladder/diagram: tali modifiche normalmente sono incorporate nel file ladder principale oppure sono trasmesse da RDE3 al controllo numerico collegato in fase di caricamento/collaudo delle modifiche live stesse.

Le modifiche live sono composte principalmente da una [intestazione](#)^[133] quindi da varie sezioni tra cui le principali sono:

- [tabella comandi live](#)^[133];
- [tabella stringhe](#)^[57].

NOTA: tutti i dati numerici memorizzati nel presente file sono da considerarsi in formato little indian (standard x86).

Intestazione modifiche live

Ad inizio file (offset 0) è presente una intestazione generale (LIVEHEADER) che contiene le informazioni di accesso a tutte le restanti componenti e sotto-sezioni del file.

Nome	Offset	Tipo	Descrizione
VERSION	+0	U32	Versione del file (1.3.0).
VID	+4	U32	ID versione del file
PLVID	+8	U32	ID versione del file prima delle modifiche

Nome	Offset	Tipo	Descrizione
			live. Se le modifiche sono effettive da applicare, questo campo deve valere 0.
	+12	U32	(riservato)
FLAGS	+16	U32	Impostazioni file: 0x00000001 = Il file contiene errori di verifica
LIVEFLAGS	+20	U32	Impostazioni modifiche live: (nessuna)
STRINGSOFF	+24	U32	Offset della tabella stringhe ^[57] (relativo al file).
STRINGSSIZE	+28	U32	Dimensione della tabella stringhe ^[57] .
LIVECMDSOFF	+32	U32	Offset della tabella comandi live ^[135] (relativo al file).
LIVECMDSSIZE	+36	U32	Dimensione della tabella comandi live ^[135] .
LASTBLOCKID	+40	U32	Ultimo ID blocco utilizzato.
LASTRUNGID	+44	U32	Ultimo ID circuito utilizzato.
LASTSTRUCTID	+48	U32	Ultimo ID struttura utilizzato.
LASTVARID	+52	U32	Ultimo ID variabile utilizzato.
LASTOBCLASSID	+56	U32	Ultimo ID classe OB utilizzato.
IARECNT	+60	U32	N. di elementi tipo ARE presenti nel file.
IAFECNT	+64	U32	N. di elementi tipo AFE presenti nel file.
	+68	U8[8]	(riservati)

Tabella comandi live

La tabella comandi live (LIVECMDS) è composta da un numero di definizioni (LIVECMD) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico comando live.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente dai campi [LIVEHEADER](#)^[133].ITEMSOFF e [LIVEHEADER](#)^[133].ITEMSSIZE.

Definizione comando live

I comandi live (LIVECMD) hanno una parte iniziale comune, quindi un'area dati specifica per ogni tipo di elemento. La parte comune è la seguente:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	Codice del comando
SIZE	+2	U32	Dimensione completa del comando (tutti i campi).
DATA	+6	...	Dati specifici del comando (opzionali).

I comandi attualmente previsti sono:

- [Comando 0x0001 - Inserimento circuito](#)^[135]
- [Comando 0x0002 - Modifica circuito](#)^[136]
- [Comando 0x0003 - Rimozione circuito](#)^[137]
- [Comando 0x0006 - Inserimento struttura](#)^[137]
- [Comando 0x0007 - Inserimento variabile](#)^[138]

Comando 0x0001 - Inserimento circuito

Questo comando permette di inserire un circuito in un dato blocco, in una certa posizione. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0001
SIZE	+2	U32	32 + ITEMSSIZE
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito interessato (và in stato Inserting)
REFID	+14	U32	ID del circuito di riferimento (solo per locazioni inserimento #1, altrimenti 0)
IFLAGS	+18	U16	Impostazioni di inserimento: 0x000F = Locazione di inserimento:

Nome	Offset	Tipo	Descrizione
			0x0 = Prima del circuito con ID REFID (#1) 0x1 = Dopo il circuito con ID REFID (#1) 0x2 = Come primo circuito del blocco 0x3 = Come ultimo circuito del blocco
FLAGS	+20	U16	Impostazioni del circuito: 0x0001 (ignorato in modifiche live)
ITEMSOFF	+22	U32	Offset della tabella elementi circuito ^[42] (relativo al file)
ITEMSSIZE	+26	U32	Dimensione della tabella elementi circuito ^[42]
TEXT	+30	U32	Offset descrizione del circuito (relativo a stringhe ^[57])

ATTENZIONE: la tabella elementi circuito deve comunque essere accodata al comando live stesso.

Comando 0x0002 - Modifica circuito

Questo comando permette di inserire un circuito modificato, rimuovendo quello vecchio, in un dato blocco. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0002
SIZE	+2	U32	32 + ITEMSSIZE
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito interessato (va in stato Inserting)
REFID	+14	U32	ID del circuito precedente da rimuovere (va in stato Dropping)
	+18	U16	(riservato)
FLAGS	+20	U16	Impostazioni del circuito:

Nome	Offset	Tipo	Descrizione
			0x0001 (ignorato in modifiche live)
ITEMSOFF	+22	U32	Offset della tabella elementi circuito ^[42] (relativo al file)
ITEMSSIZE	+26	U32	Dimensione della tabella elementi circuito ^[42]
TEXT	+30	U32	Offset descrizione del circuito (relativo a stringhe ^[57])

Comando 0x0003 - Rimozione circuito

Questo comando permette di rimuovere un circuito esistente. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0003
SIZE	+2	U32	14
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito da rimuovere (va in stato Deleting)

Comando 0x0006 - Inserimento struttura

Questo comando permette di creare una nuova struttura. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0006
SIZE	+2	U32	44 + (VARSCNT * sizeof (VAR ^[37]))
TYPEID	+6	U32	ID tipo struttura utente Nota: solo valori 0x0001000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
FLAGS	+10	U16	Impostazioni struttura: 0x0001 (ignorato in modifiche live) 0x0002 (ignorato in modifiche live)

Nome	Offset	Tipo	Descrizione
NAME	+12	U32	Offset nome della struttura (relativo a stringhe ^[57h]).
TEXT	+16	U32	Offset descrizione della struttura (relativo a stringhe ^[57h]).
VARSOFF	+20	U32	Offset della tabella variabili ^[37h] (relativa al file) (#1)
VARSCNT	+24	U32	Dimensione della tabella variabili ^[37h]
SIZE	+28	U32	Dimensione della struttura [byte]
OBCID	+32	U32	ID classe OB di riferimento (0=nessuna)(#2,#3)
ATTRIB	+36	U32	Attributi della struttura: 0x00000001 Struttura riferita a classe OB (AREFOBC)(#2) 0x00000002 Struttura principale classe OB (AISOBC)(#2,#3)
MODIFIER	+40	U32	Offset modificatori della struttura (relativo a stringhe ^[57h]).

(#1) La definizione di struttura usa una normale tabella di definizione variabili per definire i propri elementi. In questo caso gli ID delle variabili non hanno senso e dovrebbero essere sempre a 0.

(#2) A pari valore di OBCID non 0, con attributo AREFOBC, può esistere una sola struttura con flag AISOBC, mentre posso essere 0 o più strutture senza attributo AISOBC.

(#3) Una struttura viene definita "orfana" qualora si trova nella condizione di attributi AREFOBC impostato e campo OBCID = 0: questa condizione capita quando aggiornando/rimuovendo una classe OB, vi sono ancora delle variabili (o altro) che vi fanno riferimento.

Comando 0x0007 - Inserimento variabile

Questo comando permette di creare una nuova variabile dotata di valore di inizializzazione. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0007
SIZE	+2	U32	60
VARID	+6	U32	ID univoco della variabile utente

Nome	Offset	Tipo	Descrizione
			Nota: solo valori 0x0001000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
TYPEID	+10	U32	ID tipo dato variabile Nota: può essere un tipo predefinito ^[58] oppure definito dall'utente ^[35] .
FLAGS	+14	U16	Impostazioni della variabile: 0x0001 (ignorato in modifiche live) 0x0002 (ignorato in modifiche live) 0x0004 (ignorato in modifiche live)
NAME	+16	U32	Offset nome della variabile (relativo a stringhe ^[57])
TEXT	+20	U32	Offset descrizione della variabile (relativo a stringhe ^[57])
BITTEXT	+24	U32	Offset descrizioni dei bit variabili, ovvero sequenza di stringhe relative ad ogni singolo bit separate dal carattere ascii 0xFF (relativo a stringhe ^[57]).
DIM0	+28	U32	1a dimensione array (0=non specificata)(#1) o indice 0 per Alias ^[59] .
DIM1	+32	U32	2a dimensione array (0=non specificata)(#1) o indice 1 per Alias ^[59] .
DIM2	+36	U32	3a dimensione array (0=non specificata)(#1) o indice 2 per Alias ^[59] .
ALIASID	+40	U16	ID tipo alias ^[60]
ALIASBIT	+42	U16	N. di bit per alias (dipende dal tipo di ID alias)
SIZE	+44	U32	Dimensione della variabile. Se Alias ^[59] , la

Nome	Offset	Tipo	Descrizione
			dimensione è 0.
VALUE	+48	U32	Offset valore iniziale della variabile (relativo a stringhe^[57] , 0=nessun valore).
ATTRIB	+52	U32	Attributi variabile: 0x00000001 La variabile può essere scritta (WRITE) 0x00000002 La variabile può essere letta (READ) 0x00000004 La variabile può essere forzata (FORCE) 0x00000008 La variabile può essere rilasciata (RELEASE) 0x00000010 La variabile è costante (CONST) 0x00000020 La variabile non ha accesso a bit (NOBITS)
MODIFIER	+56	U32	Offset modificatori della variabile (relativo a stringhe^[57]).
DIM0L	+60	U32	Offset etichetta 1a dimensione array (relativo a stringhe^[57]). (#2)
DIM1L	+64	U32	Offset etichetta 2a dimensione array (relativo a stringhe^[57]). (#2)
DIM2L	+68	U32	Offset etichetta 3a dimensione array (relativo a stringhe^[57]). (#2)

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW la dimensione massima va comunque considerata solo a 16bit (ovvero massimo indice 0xFFFF).

(#2) Per etichetta si intende nome di una variabile (a valore costante) oppure la rappresentazione ascii di un valore numerico, ovvero un offset alla [tabella stringhe^{\[57\]}](#).

Formato modifiche live (versione 1.4)

Formato modifiche live (versione 1.4)

Le modifiche live sono sostanzialmente un file contenente un elenco di comandi per modificare, in una certa sequenza, un programma ladder/diagram: tali modifiche normalmente sono incorporate nel file ladder principale oppure sono trasmesse da RDE3 al controllo numerico collegato in fase di caricamento/collaudo delle modifiche live stesse.

Le modifiche live sono composte principalmente da una [intestazione](#)^[142] quindi da varie sezioni tra cui le principali sono:

- [tabella comandi live](#)^[142];
- [tabella stringhe](#)^[88].

NOTA: tutti i dati numerici memorizzati nel presente file sono da considerarsi in formato little indian (standard x86).

Intestazione modifiche live

Ad inizio file (offset 0) è presente una intestazione generale (LIVEHEADER) che contiene le informazioni di accesso a tutte le restanti componenti e sotto-sezioni del file.

Nome	Offset	Tipo	Descrizione
VERSION	+0	U32	Versione del file (1.4.0).
VID	+4	U32	ID versione del file
PLVID	+8	U32	ID versione del file prima delle modifiche live. Se le modifiche sono effettive da applicare, questo campo deve valere 0.
	+12	U32	(riservato)
FLAGS	+16	U32	Impostazioni file: 0x00000001 = Il file contiene errori di verifica 0x00000004 = Il file utilizza codifica UTF-8
LIVEFLAGS	+20	U32	Impostazioni modifiche live: (nessuna)
STRINGSOFF	+24	U32	Offset della tabella stringhe (relativo al file).
STRINGSSIZE	+28	U32	Dimensione della tabella stringhe.
LIVECMDSOFF	+32	U32	Offset della tabella comandi live ^[142] (relativo al file).

Nome	Offset	Tipo	Descrizione
LIVECMDSSIZE	+36	U32	Dimensione della tabella comandi live ^[142] .
LASTBLOCKID	+40	U32	Ultimo ID blocco utilizzato.
LASTRUNGID	+44	U32	Ultimo ID circuito utilizzato.
LASTSTRUCTID	+48	U32	Ultimo ID struttura utilizzato.
LASTVARID	+52	U32	Ultimo ID variabile utilizzato.
LASTOBCLASSID	+56	U32	Ultimo ID classe OB utilizzato.
IARECNT	+60	U32	N. di elementi tipo ARE presenti nel file.
IAFECNT	+64	U32	N. di elementi tipo AFE presenti nel file.
	+68	U8[60]	(riservati)

Tabella comandi live

La tabella comandi live (LIVECMD) è composta da un numero di definizioni (LIVECMD) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico comando live.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente dai campi [LIVEHEADER](#)^[141].ITEMSOFF e [LIVEHEADER](#)^[141].ITEMSSIZE.

Definizione comando live

I comandi live (LIVECMD) hanno una parte iniziale comune, quindi un'area dati specifica per ogni tipo di elemento. La parte comune è la seguente:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	Codice del comando
SIZE	+2	U32	Dimensione completa del comando (tutti i campi).
DATA	+6	...	Dati specifici del comando (opzionali).

I comandi attualmente previsti sono:

- [Comando 0x0001 - Inserimento circuito](#)^[143]
- [Comando 0x0002 - Modifica circuito](#)^[144]
- [Comando 0x0003 - Rimozione circuito](#)^[144]

- [Comando 0x0008 - Inserimento struttura](#)^[144]
- [Comando 0x0009 - Inserimento variabile](#)^[146]

Comando 0x0001 - Inserimento circuito

Questo comando permette di inserire un circuito in un dato blocco, in una certa posizione. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0001
SIZE	+2	U32	32 + ITEMSSIZE
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito interessato (và in stato Inserting)
REFID	+14	U32	ID del circuito di riferimento (solo per locazioni inserimento #1, altrimenti 0)
IFLAGS	+18	U16	Impostazioni di inserimento: 0x000F = Locazione di inserimento: 0x0 = Prima del circuito con ID REFID (#1) 0x1 = Dopo il circuito con ID REFID (#1) 0x2 = Come primo circuito del blocco 0x3 = Come ultimo circuito del blocco
FLAGS	+20	U16	Impostazioni del circuito: 0x0001 (ignorato in modifiche live)
ITEMSOFF	+22	U32	Offset della tabella elementi circuito ^[73] (relativo al file)
ITEMSSIZE	+26	U32	Dimensione della tabella elementi circuito ^[73]
TEXT	+30	U32	Offset descrizione del circuito (relativo a stringhe ^[88])

ATTENZIONE: la tabella elementi circuito deve comunque essere accodata al comando live stesso.

Comando 0x0002 - Modifica circuito

Questo comando permette di inserire un circuito modificato, rimuovendo quello vecchio, in un dato blocco. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0002
SIZE	+2	U32	32 + ITEMSSIZE
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito interessato (va in stato Inserting)
REFID	+14	U32	ID del circuito precedente da rimuovere (va in stato Dropping)
	+18	U16	(riservato)
FLAGS	+20	U16	Impostazioni del circuito: 0x0001 (ignorato in modifiche live)
ITEMSOFF	+22	U32	Offset della tabella elementi circuito ^[73] (relativo al file)
ITEMSSIZE	+26	U32	Dimensione della tabella elementi circuito ^[73]
TEXT	+30	U32	Offset descrizione del circuito (relativo a stringhe ^[88])

Comando 0x0003 - Rimozione circuito

Questo comando permette di rimuovere un circuito esistente. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0003
SIZE	+2	U32	14
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito da rimuovere (va in stato Deleting)

Comando 0x0008 - Inserimento struttura

Questo comando permette di creare una nuova struttura. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0008
SIZE	+2	U32	$68 + (\text{VARSCNT} * \text{sizeof}(\text{VAR}_{68}))$
TYPEID	+6	U32	ID tipo struttura utente Nota: solo valori 0x0001000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
FLAGS	+10	U16	Impostazioni struttura: 0x0001 (ignorato in modifiche live) 0x0002 (ignorato in modifiche live)
NAME	+12	U32	Offset nome della struttura (relativo a stringhe₆₈).
TEXT	+16	U32	Offset descrizione della struttura (relativo a stringhe₆₈).
VARSOFF	+20	U32	Offset della tabella variabili₆₈ (relativa al file) (#1)
VARSCNT	+24	U32	Dimensione della tabella variabili₆₈
SIZE	+28	U32	Dimensione della struttura [byte].
OBCID	+32	U32	ID classe OB di riferimento (0=nessuna).(#2,#3)
ATTRIB	+36	U32	Attributi della struttura: 0x00000001 Struttura riferita a classe OB (AREFOBC)(#2) 0x00000002 Struttura principale classe OB (AISOBC)(#2,#3) 0x00000004 Struttura con nome esterno (AEXTNAME)
MODIFIER	+40	U32	Offset modificatori della struttura (relativo a

Nome	Offset	Tipo	Descrizione
			stringhe ^[88]).
REFID	+44	U32	ID di riferimento tipo struttura (#4).
EXTNAME	+48	U32	Offset nome esterno struttura (relativo a stringhe ^[88]).
	+52	U8[16]	(riservati)

(#1) La definizione di struttura usa una normale tabella di definizione variabili per definire i propri elementi. In questo caso gli ID delle variabili non hanno senso e dovrebbero essere sempre a 0.

(#2) A pari valore di OBCID non 0, con attributo AREFOBC, può esistere una sola struttura con flag AISOBC, mentre posso essere 0 o più strutture senza attributo AISOBC.

(#3) Una struttura viene definita "orfana" qualora si trova nella condizione di attributi AREFOBC impostato e campo OBCID = 0: questa condizione capita quando aggiornando/rimuovendo una classe OB, vi sono ancora delle variabili (o altro) che vi fanno riferimento.

(#4) In caso di attributo AREFOBC il campo REFID contiene l'ID della struttura come definito del file OBB di origine (0 in case di attributo AISOBC).

Comando 0x0009 - Inserimento variabile

Questo comando permette di creare una nuova variabile dotata di valore di inizializzazione. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0009
SIZE	+2	U32	76
VARID	+6	U32	ID univoco della variabile utente Nota: solo valori 0x0001000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
TYPEID	+10	U32	ID tipo dato variabile Nota: può essere un tipo predefinito ^[90] oppure definito dall'utente ^[66] .
FLAGS	+14	U16	Impostazioni della variabile: 0x0001 (ignorato in modifiche live) 0x0002 (ignorato in modifiche live) 0x0004 (ignorato in modifiche live)

Nome	Offset	Tipo	Descrizione
NAME	+16	U32	Offset nome della variabile (relativo a stringhe ^[88]).
TEXT	+20	U32	Offset descrizione della variabile (relativo a stringhe ^[88]).
BITTEXT	+24	U32	Offset descrizioni dei bit variabili, ovvero sequenza di stringhe relative ad ogni singolo bit separate dal carattere ascii 0xFF (relativo a stringhe ^[57]).
DIM0	+28	U32	1a dimensione array (0=non specificata)(#1) o indice 0 per Alias ^[90] .
DIM1	+32	U32	2a dimensione array (0=non specificata)(#1) o indice 1 per Alias ^[90] .
DIM2	+36	U32	3a dimensione array (0=non specificata)(#1) o indice 2 per Alias ^[90] .
ALIASID	+40	U16	ID tipo alias ^[92]
ALIASBIT	+42	U16	N. di bit per alias (dipende dal tipo di ID alias)
SIZE	+44	U32	Dimensione della variabile. Se Alias ^[90] , la dimensione è 0.
VALUE	+48	U32	Offset valore iniziale della variabile (relativo a stringhe ^[88] , 0=nessun valore).
ATTRIB	+52	U32	Attributi variabile: 0x00000001 La variabile può essere scritta (WRITE) 0x00000002 La variabile può essere letta (READ) 0x00000004 La variabile può essere forzata (FORCE) 0x00000008 La variabile può essere rilasciata (RELEASE)

Nome	Offset	Tipo	Descrizione
			0x00000010 La variabile è costante (CONST) 0x00000020 La variabile non ha accesso a bit (NOBITS)
MODIFIER	+56	U32	Offset modificatori della variabile (relativo a stringhe ^[88]).
DIM0L	+60	U32	Offset etichetta 1a dimensione array (relativo a stringhe ^[88]). (#2)
DIM1L	+64	U32	Offset etichetta 2a dimensione array (relativo a stringhe ^[88]). (#2)
DIM2L	+68	U32	Offset etichetta 3a dimensione array (relativo a stringhe ^[88]). (#2)
	+72	U8[16]	(riservati)

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW la dimensione massima va comunque considerata solo a 16bit (ovvero massimo indice 0xFFFF).

(#2) Per etichetta si intende nome di una variabile (a valore costante) oppure la rappresentazione ascii di un valore numerico, ovvero un offset alla tabella stringhe .

Formato modifiche live (versione 1.5)

Formato modifiche live (versione 1.5)

Le modifiche live sono sostanzialmente un file contenente un elenco di comandi per modificare, in una certa sequenza, un programma ladder/diagram: tali modifiche normalmente sono incorporate nel file ladder principale oppure sono trasmesse da RDE al controllo numerico collegato in fase di caricamento/collaudo delle modifiche live stesse.

Le modifiche live sono composte principalmente da una [intestazione](#)^[149] quindi da varie sezioni tra cui le principali sono:

- [tabella comandi live](#)^[150];
- [tabella stringhe](#)^[88].

NOTA: tutti i dati numerici memorizzati nel presente file sono da considerarsi in formato little indian (standard x86).

Intestazione modifiche live

Ad inizio file (offset 0) è presente una intestazione generale (LIVEHEADER) che contiene le informazioni di accesso a tutte le restanti componenti e sotto-sezioni del file.

Nome	Offset	Tipo	Descrizione
VERSION	+0	U32	Versione del file (1.5.0).
VID	+4	U32	ID versione del file
PLVID	+8	U32	ID versione del file prima delle modifiche live. Se le modifiche sono effettive da applicare, questo campo deve valere 0.
	+12	U32	(riservato)
FLAGS	+16	U32	Impostazioni file: 0x00000001 = Il file contiene errori di verifica 0x00000004 = Il file utilizza codifica UTF-8
LIVEFLAGS	+20	U32	Impostazioni modifiche live: (nessuna)
STRINGSOFF	+24	U32	Offset della tabella stringhe (relativo al file).
STRINGSSIZE	+28	U32	Dimensione della tabella stringhe.
LIVECMDSOFF	+32	U32	Offset della tabella comandi live ^[150] (relativo al file).
LIVECMDSSIZE	+36	U32	Dimensione della tabella comandi live ^[150] .
LASTBLOCKID	+40	U32	Ultimo ID blocco utilizzato.
LASTRUNGID	+44	U32	Ultimo ID circuito utilizzato.
LASTSTRUCTID	+48	U32	Ultimo ID struttura utilizzato.
LASTVARID	+52	U32	Ultimo ID variabile utilizzato.
LASTOBCCLASSID	+56	U32	Ultimo ID classe OB utilizzato.

Nome	Offset	Tipo	Descrizione
IARECNT	+60	U32	N. di elementi tipo ARE presenti nel file.
IAFECNT	+64	U32	N. di elementi tipo AFE presenti nel file.
LASTINCFILEID	+68	U32	Ultimo ID file include utilizzato
	+72	U8[56]	(riservati)

Tabella comandi live

La tabella comandi live (LIVECMDS) è composta da un numero di definizioni (LIVECMD) contigue, ognuna delle quali contiene tutte le informazioni su uno specifico comando live.

La posizione (relativa al file) e la dimensione della tabella sono specificate rispettivamente dai campi [LIVEHEADER](#)^[149].ITEMSOFF e [LIVEHEADER](#)^[149].ITEMSSIZE.

Definizione comando live

I comandi live (LIVECMD) hanno una parte iniziale comune, quindi un'area dati specifica per ogni tipo di elemento. La parte comune è la seguente:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	Codice del comando
SIZE	+2	U32	Dimensione completa del comando (tutti i campi).
DATA	+6	...	Dati specifici del comando (opzionali).

I comandi attualmente previsti sono:

- [Comando 0x0001 - Inserimento circuito](#)^[150]
- [Comando 0x0002 - Modifica circuito](#)^[151]
- [Comando 0x0003 - Rimozione circuito](#)^[152]
- [Comando 0x0008 - Inserimento struttura](#)^[152]
- [Comando 0x0009 - Inserimento variabile](#)^[154]

Comando 0x0001 - Inserimento circuito

Questo comando permette di inserire un circuito in un dato blocco, in una certa posizione. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0001
SIZE	+2	U32	32 + ITEMSSIZE

Nome	Offset	Tipo	Descrizione
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito interessato (và in stato Inserting)
REFID	+14	U32	ID del circuito di riferimento (solo per locazioni inserimento #1, altrimenti 0)
IFLAGS	+18	U16	Impostazioni di inserimento: 0x000F = Locazione di inserimento: 0x0 = Prima del circuito con ID REFID (#1) 0x1 = Dopo il circuito con ID REFID (#1) 0x2 = Come primo circuito del blocco 0x3 = Come ultimo circuito del blocco
FLAGS	+20	U16	Impostazioni del circuito: 0x0001 (ignorato in modifiche live)
ITEMSOFF	+22	U32	Offset della tabella elementi circuito ^[73h] (relativo al file)
ITEMSSIZE	+26	U32	Dimensione della tabella elementi circuito ^[73h]
TEXT	+30	U32	Offset descrizione del circuito (relativo a stringhe ^[88h])

ATTENZIONE: la tabella elementi circuito deve comunque essere accodata al comando live stesso.

Comando 0x0002 - Modifica circuito

Questo comando permette di inserire un circuito modificato, rimuovendo quello vecchio, in un dato blocco. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0002
SIZE	+2	U32	32 + ITEMSSIZE

Nome	Offset	Tipo	Descrizione
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito interessato (va in stato Inserting)
REFID	+14	U32	ID del circuito precedente da rimuovere (va in stato Dropping)
	+18	U16	(riservato)
FLAGS	+20	U16	Impostazioni del circuito: 0x0001 (ignorato in modifiche live)
ITEMSOFF	+22	U32	Offset della tabella elementi circuito ^[73] (relativo al file)
ITEMSSIZE	+26	U32	Dimensione della tabella elementi circuito ^[73]
TEXT	+30	U32	Offset descrizione del circuito (relativo a stringhe ^[88])

Comando 0x0003 - Rimozione circuito

Questo comando permette di rimuovere un circuito esistente. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0003
SIZE	+2	U32	14
BLOCKID	+6	U32	ID blocco contenitore
RUNGID	+10	U32	ID del circuito da rimuovere (va in stato Deleting)

Comando 0x0008 - Inserimento struttura

Questo comando permette di creare una nuova struttura. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0008

Nome	Offset	Tipo	Descrizione
SIZE	+2	U32	68 + (VARSCNT * sizeof (VAR ₆₈))
TYPEID	+6	U32	ID tipo struttura utente Nota: solo valori 0x0001000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
FLAGS	+10	U16	Impostazioni struttura: 0x0001 (ignorato in modifiche live) 0x0002 (ignorato in modifiche live)
NAME	+12	U32	Offset nome della struttura (relativo a stringhe₆₈).
TEXT	+16	U32	Offset descrizione della struttura (relativo a stringhe₆₈).
VARSOFF	+20	U32	Offset della tabella variabili₆₈ (relativa al file) (#1)
VARSCNT	+24	U32	Dimensione della tabella variabili₆₈
SIZE	+28	U32	Dimensione della struttura [byte].
OBCID	+32	U32	ID classe OB di riferimento (0=nessuna).(#2,#3)
ATTRIB	+36	U32	Attributi della struttura: 0x00000001 Struttura riferita a classe OB (AREFOBC)(#2) 0x00000002 Struttura principale classe OB (AISOBC)(#2,#3) 0x00000004 Struttura con nome esterno (AEXTNAME)
MODIFIER	+40	U32	Offset modificatori della struttura (relativo a stringhe₆₈).
REFID	+44	U32	ID di riferimento tipo struttura (#4).

Nome	Offset	Tipo	Descrizione
EXTNAME	+48	U32	Offset nome esterno struttura (relativo a stringhe ^[88]).
	+52	U8[16]	(riservati)

(#1) La definizione di struttura usa una normale tabella di definizione variabili per definire i propri elementi. In questo caso gli ID delle variabili non hanno senso e dovrebbero essere sempre a 0.

(#2) A pari valore di OBCID non 0, con attributo AREFOBC, può esistere una sola struttura con flag AISOBC, mentre posso essere 0 o più strutture senza attributo AISOBC.

(#3) Una struttura viene definita "orfana" qualora si trova nella condizione di attributi AREFOBC impostato e campo OBCID = 0: questa condizione capita quando aggiornando/rimuovendo una classe OB, vi sono ancora delle variabili (o altro) che vi fanno riferimento.

(#4) In caso di attributo AREFOBC il campo REFID contiene l'ID della struttura come definito del file OBB di origine (0 in case di attributo AISOBC).

Comando 0x0009 - Inserimento variabile

Questo comando permette di creare una nuova variabile dotata di valore di inizializzazione. I dati del comando, sono i seguenti:

Nome	Offset	Tipo	Descrizione
CODE	+0	U16	0x0009
SIZE	+2	U32	76
VARID	+6	U32	ID univoco della variabile utente Nota: solo valori 0x0001000-0xFFFFFFFFE; i valori 0 e 0xFFFFFFFF sono ID a significato speciale.
TYPEID	+10	U32	ID tipo dato variabile Nota: può essere un tipo predefinito ^[90] oppure definito dall'utente ^[66] .
FLAGS	+14	U16	Impostazioni della variabile: 0x0001 (ignorato in modifiche live) 0x0002 (ignorato in modifiche live) 0x0004 (ignorato in modifiche live)
NAME	+16	U32	Offset nome della variabile (relativo a stringhe ^[88]).

Nome	Offset	Tipo	Descrizione
TEXT	+20	U32	Offset descrizione della variabile (relativo a stringhe ^[88^h]).
BITTEXT	+24	U32	Offset descrizioni dei bit variabili, ovvero sequenza di stringhe relative ad ogni singolo bit separate dal carattere ascii 0xFF (relativo a stringhe ^[57^h]).
DIM0	+28	U32	1a dimensione array (0=non specificata)(#1) o indice 0 per Alias ^[90^h] .
DIM1	+32	U32	2a dimensione array (0=non specificata)(#1) o indice 1 per Alias ^[90^h] .
DIM2	+36	U32	3a dimensione array (0=non specificata)(#1) o indice 2 per Alias ^[90^h] .
ALIASID	+40	U16	ID tipo alias ^[92^h]
ALIASBIT	+42	U16	N. di bit per alias (dipende dal tipo di ID alias)
SIZE	+44	U32	Dimensione della variabile. Se Alias ^[90^h] , la dimensione è 0.
VALUE	+48	U32	Offset valore iniziale della variabile (relativo a stringhe ^[88^h] , 0=nessun valore).
ATTRIB	+52	U32	Attributi variabile: 0x00000001 La variabile può essere scritta (WRITE) 0x00000002 La variabile può essere letta (READ) 0x00000004 La variabile può essere forzata (FORCE) 0x00000008 La variabile può essere rilasciata (RELEASE) 0x00000010 La variabile è costante (CONST)

Nome	Offset	Tipo	Descrizione
			0x00000020 La variabile non ha accesso a bit (NOBITS)
MODIFIER	+56	U32	Offset modificatori della variabile (relativo a stringhe^[88]).
DIM0L	+60	U32	Offset etichetta 1a dimensione array (relativo a stringhe^[88]). (#2)
DIM1L	+64	U32	Offset etichetta 2a dimensione array (relativo a stringhe^[88]). (#2)
DIM2L	+68	U32	Offset etichetta 3a dimensione array (relativo a stringhe^[88]). (#2)
	+72	U8[16]	(riservati)

(#1) Sebbene i campi ARRAYn siano a 32bit, nella versione attuale di SW la dimensione massima va comunque considerata solo a 16bit (ovvero massimo indice 0xFFFF).

(#2) Per etichetta si intende nome di una variabile (a valore costante) oppure la rappresentazione ascii di un valore numerico, ovvero un offset alla tabella stringhe .

Indice

- B -

BLOCK 40
BLOCKS 40

- C -

CSTRINGS 57

- E -

EMBLIVE 58
EMBLIVES 58

- F -

Formato file (versione 1.2) 7
Formato file (versione 1.3) 32
Formato file (versione 1.4) 63
Formato file (versione 1.5) 94
Formato modifiche live (versione 1.1) 127
Formato modifiche live (versione 1.3) 133
Formato modifiche live (versione 1.4) 141
Formato modifiche live (versione 1.5) 148

- H -

HEADER 33

- I -

ITEM 42
ITEMS 42

- L -

LIVECMD 128
LIVECMDS 128
LIVEHEADER 127

- R -

Requisiti minimini 6
RUNG 41
RUNGS 41

- S -

STRINGS 57
STRUCT 35
STRUCTS 35

- V -

VAR 37
VARS 37