

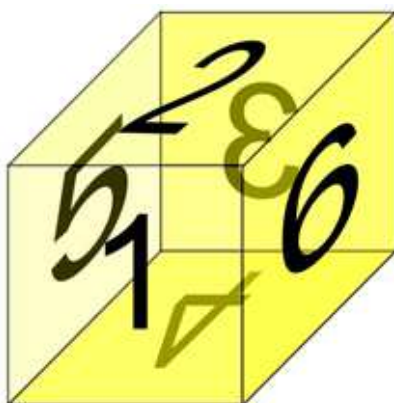


8 BITS MICROCONTROLLERS

HCS08

MACRO'S MANUAL

FUNZIONI DI ARITMETICA INTERA



Document Revision : 1.73 20.Giugno.2018

Author : Rech Marzio – **IW3FBA**

marzio.rech@negrelliforcellini.gov.it : **IZ3JCN**

What's New

V 1.50 : 02.Maggio.2017 :

- Aggiunto Tabella dei DATA TYPE

V 1.60 : 11.Maggio.2017 :

- Aggiunto Funzione : *IntScaleAdj*

- Aggiunto Funzione : *Log40dB*

V 1.70 : 15.Marzo.2018 :

- Aggiunto Funzioni : *U08Divide*, *IntLinearEvaluation*, *ArrayThresholdIndex*, *IntClipValue*

V 1.72 : 08.Maggio.2018 :

- Modifica a Funzione *CastMath*. Inserita come destinazione A e H:X se *DataType* Coerente.

V 1.73 : 20.Giugno.2018 :

- Inserite funzioni : *SquareU08Scale*, *U08ScaleAdjust*

FUNZIONI MACRO

Le Funzioni Macro sono delle procedure costruite dal programmatore che richiamano blocchi di istruzioni in linguaggio Assembly. Lo scopo delle Macro è quello di semplificare la Programmazione. Una Macro può richiedere l'assegnazione di parametri o argomenti che vengono passati contemporaneamente alla sua chiamata.

FUNZIONI MACRO DI ARITMETICA INTERA

Le Funzione di aritmetica Intera espresse in questo manuale, consentono di effettuare le 4 operazioni aritmetiche con numeri interi a 32 Bits oltre ad elevamento a potenza Intera e Radice Quadrata.

Qualora non sia necessario usare le Librerie di Funzioni matematiche di Aritmetica in Virgola Mobile, le funzioni di aritmetica con interi consentono di risparmiare notevolmente lo spazio di memoria FLASH e di essere eseguite piuttosto velocemente.

Questi vantaggi possono essere indispensabili quando la MCU usata abbia una memoria FLASH limitata e Frequenza di BUS non troppo elevata.

TABELLA DATA TYPE

**CONFRONTO E SIMBOLOGIA DEI DATA TYPE NUMERICI
USATI NELL'AMBIENTE DI SVILUPPO IDES08 E ALTRI LINGUAGGI DI
PROGRAMMAZIONE**

DESCRIZIONE		DEFINIZIONE VARIABILE/COSTANTE NEI LINGUAGGI DI PROGRAMMAZIONE O TOOLS				BYTE	RANGE	
		C	LabView	NoICE Debugger	ASSEMBLY	occupati in memoria	DECIMALE	ESADECIMALE
INTERO AD 8 BIT SENZA SEGNO	UNSIGNED 8 BIT	uint8	U8	U08	.byte	1	0...255	\$00...\$FF
INTERO AD 8 BIT CON SEGNO	SIGNED 8BIT	int int8	I8	S08		1	-128...+127	\$80...\$7F
INTERO A 16 BIT SENZA SEGNO	UNSIGNED 16 BIT	uint16	U16	U16	.word	2	0..65535	\$0000...\$FFFF
INTERO A 16 BIT CON SEGNO	SIGNED 16 BIT	Int16	I16	S16		2	-32768...32767	\$8000...\$7FFF
INTERO A 32 BIT SENZA SEGNO	UNSIGNED 32 BIT	uint32	U32	U32	.long	4	0.. 4294967295	\$00000000...\$FFFFFFFF
INTERO A 32 BIT CON SEGNO	SIGNED 32 BIT	Int32	I32	S32		4	-2147483648.. 2147483647	\$80000000...\$7FFFFFFF
VIRGOLA MOBILE PRECISIONE SINGOLA	FLOATING POINT SINGLE PRECISION IEE754	float o float32	SGL	float	.float	4	$\pm \sim 10^{-44.85}$ to $\sim 10^{38.53}$	-
VIRGOLA MOBILE PRECISIONE DOPPIA	FLOATING POINT DOUBLE PRECISION IEE754	float64	DBL	double	.double	8	$\pm \sim 10^{-323.3}$ to $\sim 10^{308.3}$	-

Il presente manuale illustra le procedure per la chiamata di funzioni di aritmetica intera a 32 Bits.

Tutte queste funzioni usano i seguenti operandi a 32 Bit già dichiarati e definiti con il caricamento del modulo di librerie di matematica intera:

OPERANDI

OPERANDI/VARIABILI A 32 BIT USATI NELLE FUNZIONI DI ARITMETICA INTERA			
OPERANDO A (opera) OperA	opera+0 High Byte	32 Bits S32 U32	Primo operando di una operazione matematica e Risultato dell'operazione
	opera+1		
	opera+2		
	opera+3 Low Byte		
OPERANDO B (operb) OperB	operb+0 High Byte	32 Bits S32 U32	Secondo operando
	operb+1		
	operb+2		
	operb+3 Low Byte		

FMATH	Floating Point Math Functions		
	Variabili riservate accessibili dal programmatore		
NOME VARIABILE	Type	Address	Descrizione
math	address	-	Indirizzo di partenza dell'area operandi matematici
opera / OperA	U32/S32 FLOAT	math+0.. ...math+3	Primo operando e risultato di un operazione matematica
operb / OperB	U32/S32 FLOAT	math+4.. math+7	Secondo operando di un operazione matematica
operc	U32/S32 FLOAT	math+8.. math+11	Operando temporaneo di un operazione matematica
operd matwrk	U32/S32 FLOAT	math+12.. math+15	Accumulatore per sommatorie
signA	U08(b7)	-	B7 (Polarità) 0:positivo 1:Negativo operando A Nelle conversioni dei numeri interi ricorda la polarità
signB	U08(b7)	-	B7 (Polarità) 0:positivo 1:Negativo operando B Nelle conversioni dei numeri interi ricorda la polarità
expA	S08	.	Esponente operando A numeri Float
expB	S08	.	Esponente operando B numeri Float
point	U08	-	Punto decimale : 0..3 nella matematica intera

SIGNIFICATO DEI TERMINI USATI

- MACRO** : Istruzione definita dal Programmatore che chiama un blocco di Istruzioni Assembly.
E' definita da un nome che ricorda in qualche modo la funzione svolta dalla Macro
- PARAMETRI** : Dati passati insieme alla macro scritti alla destra del nome della Macro e separati da una virgola.
I Parametri possono essere delle costanti numeriche o delle variabili definite dal programmatore.
- ARGOMENTI** : Vedi PARAMETRI.
- DATA TYPE** : E' il tipo di dato associato ai parametri passati alla macro. Esempio U8, S8, U16, S16, U32, S32..
- U08** : Il dato è un numero Intero senza segno di lunghezza 8 Bit ovvero 1 Byte : 0...255
- S08** : Il dato è un numero Intero con segno di lunghezza 8 Bit ovvero **BYTE** : -128...+127
- U16** : Il dato è un numero Intero senza segno di lunghezza 16 Bit ovvero **WORD** : 0...65535
- S16** : Il dato è un numero Intero con segno di lunghezza 16 Bit ovvero **WORD** : -32768...+32767
- U32** : Il dato è un numero Intero senza segno di lunghezza 32 Bit ovvero **LONG** .
- S32** : Il dato è un numero Intero con segno di lunghezza 32 Bit ovvero **LONG** .
- I08-I16-I32** : Il dato è un numero Intero di cui solo la lunghezza viene specificata a prescindere dal segno.
- FLOAT** : Variabile o costante a 32 Bit del tipo Floating Point (Virgola Mobile) che rappresenta un numero reale composto da Mantissa Esponente e Segno codificata secondo le specifiche IEEE754.
- COST** : Costante: Valore numerico definito e immutabile
- VAR** : Variabile : Valore numerico che può variare all'interno di un determinato intervallo
- LOCAZIONE** : Cella di memoria
- FLAGS** : Bit di una variabile il cui stato segnala una condizione logica piuttosto che un valore numerico
- BYTE** : Variabile o costante a 8 Bit
- WORD** : Variabile o costante a 16 Bit (2 Bytes)
- LONG** : Variabile o costante a 32 Bit (4 Bytes)
- SHORT** : Vedi WORD

'In modo circolare' : Se riferito ad una variabile significa che il suo valore torna al valore minimo dopo un Incremento del suo valore massimo e viceversa.

LISTA DELLE FUNZIONI MACRO DI ARITMETICA INTERA	
NOME MACRO	FUNZIONE
PushMath <i>arg1</i>	Salva l'operando A (opera) o B (operb) nello STACK
PullMath <i>arg1</i>	Recupera l'operando A (opera) o B (operb) dallo STACK
CastInteger <i>arg1,arg2,arg3</i>	Copia una variabile o costante negli operandi A/B effettuando il Type Cast
CastMath <i>arg1,arg2</i>	Muove l' operando A (opera) su una variabile cambiando il DataType
AddS32	Somma gli operandi A e B del tipo S32/I32 (Considerando il segno)
SubS32	Sottrae gli operandi A e B del tipo S32/I32 (Considerando il segno)
MultS32	Moltiplica gli operandi A e B del tipo S32/I32 (Considerando il segno)
DivS32	Divide gli operandi A e B del tipo S32/I32 (Considerando il segno)
U08Divide <i>arg1,arg2,arg3</i>	Semplice divisione con istruzione HCS08 'div' U16 Dividendo \ U08 Divisore = U08 Quoziente
SqrtU32	Effettua la radice quadrata dell'operando A
ScaleBy <i>arg1,arg2</i>	Scala l'operando A (opera) di un fattore definito
GetAbsolute	Cambia il valore dell'operando A (opera) in assoluto salvando il segno
GetRelative	Cambia il valore dell'operando A (opera) in relativo,recuperando il segno.
CopyValue <i>arg1,arg2,arg3</i>	Copia un valore I08/I16/I32 ovunque nella memoria RAM
KonstToMath <i>arg1,arg2</i>	Copia una costante U08/U16/I32 negli operandi A o B
TestU32 <i>arg1</i>	Verifica se una variabile U32 è 0
IntPower <i>arg1,arg2</i>	Eleva a potenza intera un numero U32
IntExp10 <i>arg1,arg2</i>	Moltiplica un numero intero U32 per una potenza di 10

LISTA DELLE FUNZIONI MACRO DI ARITMETICA INTERA cont.	
NOME MACRO	FUNZIONE
IntLinearEvaluation <i>arg1,arg2,arg3,arg4,arg5,arg6,arg7</i>	Calcola il valore dipendente y in base al valore di x in una funzione lineare definita da 2 punti. ; $y=(x-X1)*m+Y1$
IntScaleAdjust <i>arg1,arg2,arg3,arg4,arg5</i>	Modifica e trasla la scala di un valore intero a 8 e 16 Bits
Log40dB <i>arg1,arg2,arg3</i>	Effettua la conversione logaritmica in dB con una dinamica di 40 dB
ArrayThresholdIndex <i>arg1,arg2,arg3,arg4</i>	Trova l'indice dell'array numerico in cui è compresa la soglia
IntClipValue <i>arg1,arg2,arg3,arg4,arg5</i>	Taglia un valore dentro i suoi limiti minimo e massimo
SquareU08Scale <i>arg1,arg2,arg3</i>	Quadra un valore U08 e lo adatta in una scala
U08ScaleAdjust <i>arg1,arg2,arg3,arg4</i>	Converte un valore U08 con una nuova scala

PushMath <i>arg1</i>				
Salva l'operando A o B nello STACK del Microcontrollore				
ProtoType		PushMath Operando		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Operando	U32	Operando da salvare nello stack : opera / operb
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE				
ESEMPI		PushMath opera ← Salva l'operando A (opera) nello stack PushMath operb ← Salva l'operando B (operb) nello stack		

PullMath <i>arg1</i>				
Recupera l'operando A o B dallo STACK del Microcontrollore				
ProtoType		PullMath Operando		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Operando	U32	Operando da recuperare dallo stack : opera / operb
	RITORNA	opera/operb	-	Valore recuperato dallo STACK
		A,H,X	I8	Invariati
NOTE				
ESEMPI		PullMath opera ← Recupera l'operando A (opera) dallo stack PullMath operb ← Recupera l'operando B (operb) dallo stack		

CastInteger <i>arg1,arg2,arg3</i>				
Copia una variabile o costante negli operandi A/B effettuando il Type Cast S32				
ProtoType		CastInteger DataType, Value, Operand		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Type	Etichetta	Tipo di variabile : U08/S08 U16/S16 U32/S32 ANA for analog format
		Value	-	Variabile o costante numerica del type indicato da inserire nell'operando 'opera' o 'operb' A : se il valore è a 8 bit nell'accumulatore H:X : se il valore è a 16 bit nel registro H:X
		Operand	U32	Operando a cui deve essere inviato l'argomento Value opera / operb
	RITORNA	A,H,X	I8	Invariati
NOTE				
ESEMPI		CastInteger S08, -84,opera ← Copia e adatta il Valore -84 nell'operando A CastInteger S16, offset,operb ← Copia e adatta la variabile 'offset' nell'operando B CastInteger U32, FACTOR,opera ← Copia e adatta la costante 'FACTOR' in oper. A CastInteger ANA, H:X,opera ← Copia e adatta il valore di H:X in oper. A CastInteger S08, A,opera ← Copia e adatta il valore di A in oper. A		

CastMath arg1,arg2				
Copia e adatta il valore dell'operando 'OperA' in una variabile				
ProtoType		CastMath DataType, Destination		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Type	Etichetta	Tipo di variabile : U08/S08 U16/S16 U32/S32 /FLOAT ANA for analog format
		Destination	-	Variabile numerica del type indicato dove copiare l'operando OperA. A : Muove un valore U008 in Acc. H:X : Muove un valore I16 in H:X
	RITORNA	A,H,X	I8	Invariati
NOTE		<p>Se il Typecast è FLOAT, vengono considerate le cifre decimali indicate dalla variabile ; point.</p> <p>$OperA = opera..+3$</p> <p>Utile nel caso di un risultato contenuto nell'operando S32 OperA che debba essere trasferito in variabili con data Type a lunghezza inferiore.</p>		
ESEMPI		<p>CastMath S08, speed ← Copia l'operando OperA in speed con datatype S08</p> <p>CastMath FLOAT, speed ← Copia l'operando OperA in speed con datatype FLOAT</p> <p>Posizionando la virgola come indicato da : point</p>		

AddS32				
Somma con segno degli operandi a 32 Bit 'OperA' 'OperB'				
ProtoType		AddS32		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	-	-	
		-	-	
	RITORNA	OperA	I32	Risultato dell'addizione: OperA+OperB
		A,H,X	I8	Invariati
NOTE		Richiede che siano settati gli operandi A e B con il DataType I32		
ESEMPI		AddS32 ← Somma con segno (Oper.A + Oper.B)		

SubS32				
Effettua la sottrazione con segno fra gli operandi a 32 Bit 'OperA' 'OperB'				
ProtoType		SubS32		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	-	-	
		-	-	
	RITORNA	OperA	I32	Risultato della sottrazione: OperA-OperB
		A,H,X	I8	Invariati
NOTE		Richiede che siano settati gli operandi A e B con il DataType I32		
ESEMPI		SubS32 ← Sottrazione con segno (Oper.A – Oper.B)		

MultS32				
Moltiplica con segno gli operandi a 32 Bit 'OperA' 'OperB'				
ProtoType		MultS32		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	-	-	
		-	-	
	RITORNA	OperA	I32	Prodotto OperA x OperB
		A,H,X	I8	Invariati
NOTE		Richiede che siano settati gli operandi A e B con il DataType I32		
ESEMPI		MultS32 ← Moltiplicazione con segno (Oper.A x Oper.B)		

DivS32				
Divide con segno gli operandi a 32 Bit 'OperA' 'OperB'				
ProtoType		DivS32		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	-	-	
		-	-	
	RITORNA	OperA	I32	Quoziente della divisione: OperA / OperB
		A,H,X	I8	Invariati
NOTE		Richiede che siano settati gli operandi A e B con il DataType I32		
ESEMPI		DivS32 ← Divisione con segno (Oper.A / Oper.B)		

U08Divide <i>arg1,arg2,arg3</i>				
Semplice Divisione a 8 bits				
ProtoType		U08Divide <i>Dividend,Divisor,Quotient</i>		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Diviodend	U16	Dividendo: Nome variabile o costante H:X se il dividendo e' nel registro H:X
		Divisor	U08	Divisore: Nome variabile o costante A se il divisore e' nel registro accumulatore
		Quotient	U08	Quoziente
	RITORNA	A	U08	Quoziente arrotondato per eccesso o difetto
		X	U08	Resto della Divisione
NOTE				
ESEMPI		U08Divide total,size,result ← Divide 'total' per 'size' pone il quoziente in 'result' U08Divide H:X,37,result ← Divide 'total' per 37 pone il quoziente in 'result' U08Divide total,45,result ← Divide 'total' per 45 pone il quoziente in 'result' U08Divide H:X,A ← Divide il contenuto di H:X per A. Il quoziente è in 'A'		

SqrtU32				
Effettua la radice quadrata dell'operando OperA				
ProtoType		SqrtU32		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	-	-	
		-	-	
	RITORNA	OperA	U32	√(OperA)
		A,H,X	I8	Invariati
NOTE		L'operando A deve essere intero a 32 bit		
ESEMPI		Sqrt32 ← √opera		

ScaleBy <i>arg1</i>				
Scala l'operando A (opera) di un fattore indicato				
ProtoType		ScaleBy Factor		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Factor	U16	Variabile o Costante per scalare l'operando A
		-	-	
	RITORNA	OperA	I32	OperA / Factor
		A,H,X	I8	Invariati
NOTE				
ESEMPI		ScaleBy 10000 ← Scala l'operando A di 10000 : opera/10000 ScaleBy FACTOR ← Scala l'operando A : OperA/FACTOR ScaleBy Ffactor ← Scala l'operando A del contenuto della variabile factor		

GetAbsolute				
Rende assoluto l'operando A salvando il segno				
ProtoType		GetAbsolute		
		Argomento	Type	Descrizione
ARGOMENTI				Nessun argomento
	RITORNA	± OperA	I32	OperA
		A,H,X	I8	Invariati
NOTE		Analizza l'operando A e se negativo effettua il complemento a 2 e salva i segno		
ESEMPI		GetAbsolute ← +/- OperA ← OperA		

GetRelative				
Cambia il valore dell'operando A (OperA) in relativo recuperando il segno				
ProtoType		GetRelative		
		Argomento	Type	Descrizione
ARGOMENTI				Nessun argomento Il segno S viene recuperato dalla Funzione GetAbsolute fatta precedentemente
	RITORNA	OperA + S	I32	±OperA
		A,H,X	I8	Invariati
NOTE		Questa funzione viene usata dopo una trasformazione in assoluto dell'operando A per recuperare il segno salvato e riconvertire l'operando in relativo.		
ESEMPI		GetRelative ← OperA ← +/- OperA		

CopyValue arg1,arg2,arg3				
Copia una costante o variabile I08/I16/I32/FLOAT in una locazione di memoria RAM				
ProtoType		CopyValue VarType,Sorgente,Destinazione		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Type	Etichetta	Tipo di variabile : I08 / I16 / I32 / FLOAT (** Vedi Nota)
		Sorgente	I08/I16/I32 FLOAT	Variabile o costante da copiare
		Destinazione	I08/I16/I32	Variabile/ locazione di memoria o registro destinazione
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		** In questo caso il Type indica solo la lunghezza in bits della variabile essendo il segno ininfluente. L' indicazione del type con la lettera I anziché U o S sta ad indicare Integer. generico.		
ESEMPI		CopyValue I08,\$75,counter ← Copia il Numero \$75 esadecimale. In counter CopyValue I16,star,counter ← Copia la variabile star in counter CopyValue I32,opera,speed ← Copia l'operando A nella variabile I32 speed CopyValue FLOAT,star,opera ← Copia la variabile star del tipo FLOAT in opera		

KonstToMath <i>arg1,arg2</i>				
Copia una costante o variabile U32/I32 negli operandi matematici a 32 bits 'opera' 'operb'				
ProtoType		KonstToMath Sorgente,Operando		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Sorgente	U32/S32	Costante da copiare definita con ' .equal' o Define
		Operando	U32/S32	opera o operb
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Come CopyValue ma solo per costanti e destinazione solo negli operandi math		
ESEMPI		KonstToMath 3520,opera	← Copia il Numero 3520 in Opera	
		KonstToMath FACTOR,opera	← Copia la costante FACTOR in Opera	

TestU32 arg1				
Verifica se una variabile U32 è 0				
ProtoType		TestU32 Valore		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Valore	U32/S32	Variabile da testare
	RITORNA	Z(CCRI	FLAGS	0: il valore non è nullo 1: Il valore è nullo
		A,H,X	I8	Invariati
NOTE				
ESEMPI		TestU32 speed TestU32 opera	← Testa il valore U32 contenuto nella variabile speed ← Verifica se l'operando A è nullo	

IntPower arg1,arg2				
Eleva a potenza intera un numero U32				
ProtoType		IntPower Base,Exp		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Base	U32	Numero da elevare a potenza : Variabile H:X . Variabile puntata da H:X
		Exp	U08	Esponente : variabile A: l'esponente è contenuto nell' accumulatore
	RITORNA	OperA	U32	(Base) ^{Exp}
		A,H,X	I8	Invariati
NOTE				
ESEMPI		IntPower 3520,25 IntPower speed,A IntPower H:X,4	← Eleva il numero 3520^25 ← Eleva il valore U32 di speed al valore contenuto in A ← Eleva il valore U32 puntato da H:X alla quarta	

IntExp10 arg1,arg2				
Moltiplica un valore U32 per una potenza di 10				
ProtoType		IntExp10 Value,Exp		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Value	U32	Numero da moltiplicare x una potenza di 10: Variabile H:X . Variabile puntata da H:X
		Exp	U08	Esponente del numero 10 : variabile A: l'esponente è contenuto nell' accumulatore
	RITORNA	OperA	U32	Value·10 ^{EXP}
		A,H,X	I8	Invariati
NOTE				
ESEMPI		IntExp10 750,4	← Moltiplica il numero 750x10^4	
		IntExp10 speed,A	← Moltiplica il valore U32 di speed x 10^A	
		IntExp10 H:X,4	← Moltiplica il valore U32 puntato da H:X x 10^4	

IntScaleAdjust arg1,arg2,arg3,arg4,arg5				
Modifica e trasla la scala di un valore intero U16 o U08				
ProtoType		IntScaleAdj Value,Scale,NewScale,Offset,NewVal		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Value	U16	Valore da modificare e traslare in scala
		Scale	U16	Scala corrente del valore
		NewScale	U16	Nuova Scala del Valore
		Offset	S16	Quantità da traslare
		NewVal	S16	Nuovo valore modificato Ammessi : A e H:X se DataType Coerente
	RITORNA	Opera..+3	S32	Nuovo valore modificato
		A,H,X	I8	Invariati
NOTE				
ESEMPI		IntScaleAdjust adcount,1023,330,0,weight ; weight varierà fra 0..330 mentre adcount varia fra 0..1023. IntScaleAdjust adcount,1023,1000,-500,weight ; weight varierà fra -500...+500 mentre adcount varia fra 0..1023. IntScaleAdjust adcount,255,200,100,weight ; weight varierà fra 100..300 mentre adcount varia fra 0..255.		

--	--

IntLinearEvaluation arg1,arg2,arg3,arg4,arg5,arg6,arg7				
Calcola il valore dipendente y in base al valore di x in una funzione lineare definita da 2 punti. Implementa la formula: $y=(x-X1)*m+Y1$ dove: x = Input value y = Output value m = (Y2-Y1)/X2-X1) X1 = Ref1 X point Y1 = Ref1 Y point				
ProtoType		IntLinearEvaluation m,Y1,Y2,X1,X2,x,y		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	m	Konstant	Precisione in numero digit desiderati: 2...6
		Y1	S32	Valore di riferimento Y1 relativo a X1
		Y2	S32	Valore di riferimento Y2 relativo a X2
		X1	S32	Valore di riferimento X1 della funzione lineare
		X2	S32	Valore di riferimento X2 della funzione lineare
		x	S32	Variabile per la quale si vuole conoscere y Se assente, il valore deve essere in OperA inserire la virgola come segnaposto
		y	S32	Variabile di uscita in corrispondenza di x Se assente, il valore sarà depositato in OperA
	RITORNA	Opera..+3	S32	y
		A,H,X	I8	Modificati
NOTE		Utile per tabellare la risposta lineare di un sensore ed effettuare il condizionamento. In questo caso x è il valore fornito dall' ADC, y è il valore acquisito dal sensore ad es la temperatura. X1 e X2 sono i valori dell'ADC in corrispondenza delle grandezze fisiche Y1 e Y2		
ESEMPI		IntLinearEvaluation 4,-300,+700,1035,3520,input,output ; Definisce una funzione lineare con le coordinate 1035,-300 e 3520,700 Calcola il valore in output in base al valore contenuto in input. Fornisce 4 Cifre di precisione. equivalenti a 10000 punti. IntLinearEvaluation 4,-300,+700,1035,3520,adcount,temperatura ; Condiziona un sensore di temperatura in cui : a -30.0 °C l' ADC fornisce 1035 punti, a +70.0°C fornisce 3520 punti temperatura conterrà il valore moltiplicato per 10 in base al valore dell'ADC IntLinearEvaluation 4,-300,+700,1035,3520,, ; Come sopra ma x è inserita in opera..+3 e y sarà ancora in opera...+3		

Log40dB arg1,arg2,arg3				
Effettua la conversione logaritmica in dB con una dinamica di 40 dB				
ProtoType		Log40dB LinVal,LogVal,Scale		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	LinVal	U08	Variabile : Valore lineare da convertire (MAX:100) A : Il valore da convertire è nell'accumulatore
		LogVal	U08	Variabile dove depositare il valore convertito A: Il valore sarà contenuto in A
		Scale	U08	Costante : effettua uno scalamento. Divide per Scale Per ottenere la dinamica desiderata.
	RITORNA	-	-	-
		A,H,X	I8	Alterati
NOTE		Essendo la dinamica massima di 40 dB, il valore massimo di LinVal è uguale a 100		
ESEMPI		Log40dB voltage,barscale	←	barscale = 20log(voltage)
		Log40dB voltage,barscale,4	←	barscale = 20log(voltage)/4
		Log40dB A,A,2	←	A= 20log(A)/2

ArrayThresholdIndex <i>arg1,arg2,arg3,arg4</i>				
Trova l'indice dell'array numerico in cui è compresa la soglia data da un valore				
ProtoType		ArrayThresholdIndex <i>array,Type,Value,Size</i>		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	array	Label	Etichetta Indirizzo dove è tabulato l'array. LA se l'array si trova in un area logica HNP
		Type	Label	Data type dei valori dell'array: U08 o U16
		Value	U08/U16 Label	Valore da ricercare nell'array per trovare l'indice. 'A' se U08 e il valore è contenuto nell' accumulatore 'H:X' se U16 e il valore è contenuto nl reg. H:X
		Size	U08 Label	Numero di valori contenuti nell'array. Per U08 Max 255 Per U16 Max 127 Se l'array si trova in un'area logica, inserire l'alias dell'area logica, oppure 'A' se il numero dell'area logica è nell'accumulatore.
	RITORNA		-	
		A,H,X	I8	Modificati
NOTE		<div>- Utile per: effettuare l'interpolazione lineare su funzioni tabellate in array. Condizionare sensori non lineari inserendo nell'array i punti di interpolazione.</div> <div>- Creare delle scale personalizzate.</div>		
ESEMPI		<div>ArrayThresholdIndex <i>table,U16,adcount,25</i> ; Trova l'indice nell array table di 25 elementi a 16 bits in cui è compreso il valore adcount</div> <div>ArrayThresholdIndex <i>table,U16,H:X,25</i> ; Come sopra ma il valore a 16 bits ora è contenuto in H:X</div> <div>ArrayThresholdIndex <i>table,U08,A,50</i> ; Come sopra ma il valore a 8 bits è contenuto nell' accumulatore</div> <div>ArrayThresholdIndex <i>LA,U16,H:X,frmc</i> ; Trova l'indice nell array contenuto nell'area logica 'frmc' composta da elementi del tipo U16, in cui è compreso il valore contenuto in H:X. compreso il valore</div> <div>ArrayThresholdIndex <i>LA,U16,H:X,A</i> ; Trova l'indice nell array contenuto nell'area logica il cui numero è contenuto in A composta da elementi del tipo U16, in cui è compreso il valore contenuto in H:X.</div>		

IntClipValue arg1,arg2,arg3,arg4,arg5				
Taglia un valore dentro i suoi limiti minimo e massimo				
ProtoType		IntClipValue Type, Value,min,Max,output		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Type	Label	Data type del valore: U08/S08 U16/S16
		Value	I08/I16	Valore da limitare 'A' o 'H.X' se il valore è nei registri
		min	I08/I16	Costante o Variabile del limite minimo
		Max	I08/I16	Costante o Variabile del limite Massimo
		Output	I08/I16	Valore limitato. Se assente, il valore sarà in A o H:X
	RITORNA		-	
		A,H,X	I8	Modificati
NOTE		Utile per limitare i valori di una variabile dentro un determinato range		
ESEMPI		IntClipValue S08,posizione,-30,110,posizione ; Limita il valore posizione nel range -30...110		
		IntClipValue U16,H:X30,1500,8400 ; Limita il valore in H:X nel range 1500..8400		

SquareU08Scale <i>arg1,arg2,arg3</i>				
Quadra un valore U08 e adatta la scala				
ProtoType		SquareU08Scale Value,Scale,Qvalue		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Value	U08	Variabile : Valore lineare da quadrare A : Il valore da convertire è nell'accumulatore
		Scale	U08	Valore massimo della scala X: Il valore sarà contenuto in X
		Qvalue	U08	Variabile di destinazione se presente Altrimenti il valore resta in A
	RITORNA	A	-	Valore quadrato e adattato
		H,X	I8	Alterati
NOTE		Formula : $Qvalue = (Value)^2 / Scale$		
ESEMPI		SquareU08Scale A,X ← Quadra il valore in A e lo adatta alla scala contenuta in X SquareU08Scale speed,100 ← Quadra il valore in speed e lo adatta alla scala 0..100		

	SquareU08Scale speed,scale,newspeed ← Il valore è depositato in newspeed
--	---

U08ScaleAdjust arg1,arg2,arg3,arg4				
Crea una nuova scala per un valore U08				
ProtoType		U08ScaleAdjust Value,NewScale,OldScale,NewValue		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Value	U08	Variabile : Valore da convertire in nuova scala A : Il valore da convertire è nell'accumulatore
		NewScale	U08	Nuovo valore di Fondo Scala X: Il valore sarà contenuto in X
		OldScale	U08	Fondo Scala Scale precedente del valore
		NewValue		Variabile di destinazione se presente altrimenti il valore resta in A
	RITORNA	A	-	Valore modificato alla nuova scala
		H,X	I8	Alterati
NOTE		Formula : $Qvalue = (Value)^2 / Scale$		
ESEMPI		U08ScaleAdjust speed,100,137 ← Modifica scala di speed da 0.137 a 0..100 U08ScaleAdjust A,100,137,star ← Modifica scala di A e deposita valore in star U08ScaleAdjust A,X,255 ← Modifica scala di A avente 255 punti con la nuova scala in X. Il nuovo valore è in A		