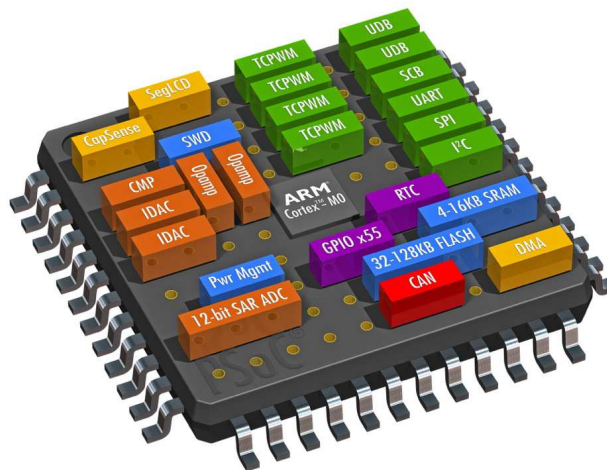


**8 BITS MICROCONTROLLERS**

HCS08

MACRO'S MANUAL

FUNZIONI UTILIZZO MODULI INTERNI

**Document Revision : 3.00 20.Giugno.2018**Author : Rech Marzio – **IW3FBA**marzio.rech@negrelliforcellini.gov.it : **IZ3JCN**

What's New

V 2.91 - V2.92 : 29.Aprile.2017 :

- Aggiunto Modulo **FLASH Data Storage** per il Back Up dei parametri in Ram
- Aggiunto Modulo **ICS Internal Clock Source** con le procedure di calibrazione oscillatore

V2.93 : 30.Maggio.2017

- Aggiunto nel Modulo **ICS** l' **MCU serie AC**

V2.94 : 24.Dicembre.2017

- Corretto descrizione esempio e note nella Funzione SetPin di DIO.

V2.95 : 11.Febbraio.2018

- Inserita opzione nel modulo SCOM funzione PrintSCOMvalue con decimal point negativi

V2.96 : 17.Maggio.2018

- Modificata funzione SCOM : PrintSCOMbool aggiungendo il carattere NewLine opzionale

V3.00: 20.Giugno.2018

- Inserita Funzione CaptureADC in Modulo ADC

Questo manuale elenca le Macro Funzioni utilizzate per la gestione dei Moduli Interni del Microcontrollori HCS08.

Le Funzioni sono catalogate per MODULO e per ogni Funzione vengono riportate le Convenzioni di Chiamata (Calling Conventions) con riportati gli argomenti di Ingresso/Uscita.

Per ogni Funzione vengono riportati degli esempi di chiamata.

E' possibile che un Modulo dipenda da un'altro Modulo. Ad esempio il Modulo PWM si appoggia al Modulo TPM del Timer. In questo caso nella Funzione viene specificata la dipendenza e i Moduli da caricare.

*Prima di illustrare nel dettaglio le Funzioni, viene riportata una tabella di elenco dei Moduli, con una breve descrizione e gli esempi correlati disponibili nella cartella **ESEMPI di S08Projects**.*

Una tabella successiva riporta l'elenco di tutte le funzioni di ogni modulo con una breve descrizione di utilizzo.

LISTA MODULI INTERNI E FILES DI ESEMPIO		
MODULO	DESCRIZIONE	CARTELLA ESEMPI DI UTILIZZO FUNZIONI
DIO <i>DIGITAL I/O PIN</i> <i>Linee di I/O Digitale</i>	Controllo delle Linee di Input Output Digitale del Microcontrollore	LED_FLASH SIGNALS_GENERATION SIGNALS_MEASURE
ADC <i>Convertitore Analogico Digitale</i>	Acquisizione segnali analogici dai Pin di Ingresso dell' ADC	AD_CONVERTER
CTS <i>Chip Temperature Sensor</i>	Sensore di Temperatura del Microcontrollore	
TPM <i>Timer Base Time</i>	Impostazione del Modulo Timer	TIMER_OCMP TIMER_PWM TIMER_ICAP SIGNALS_GENERATION SIGNALS_MEASURE
PWM <i>Timer</i> <i>Pulse Wide Modulator</i>	Generazione di Segnali PWM	TIMER_PWM
OCMP <i>TIMER</i> <i>Output Compare</i>	Generazione di precisione di segnali Impulsivi correlati a Triggers	TIMER_OCMP
ICAP <i>TIMER</i> <i>Input Capture</i>	Misura di precisione di segnali senza impegno del Microcontrollore	
SGEN <i>TIMER SIGNAL</i> <i>Generazione di segnali</i>	Generazione di Precisione di Segnali Ad onda quadra	SIGNALS_GENERATION
TCAP <i>TIMER CAPTURE</i> <i>Cattura segnali</i>	Misura di precisione di segnali con impegno del Microcontrollore	
IIC <i>IIC Bus</i> <i>Inter Integrate Communication Bus</i>	Configurazione del Bus per la comunicazione con una periferica IIC	
RTCM <i>Real Time Counter Module</i> <i>(QE – SH version)</i>	Creazione di un Orologio/Calendario	
SCOM <i>Serial Communication Module</i> SCI	Utilizzo della porta seriale come Terminale di comunicazione	
ICS <i>Internal Clock Source</i>	Definizione Frequenza di BUS e Calibrazione Oscillatore interno	
FLASH <i>Data Storage</i>	Memorizzazione Parametri di sistema Nella FLASH del MCU	FLASH_MEMORY_BACKUP

LISTA FUNZIONI MODULI INTERNI		
MODULO	NOME MACRO	FUNZIONE
DIO DIGITAL I/O PIN Linee di I/O Digitale	DefinePin <i>arg1,arg2,arg3,arg4</i>	Definisce il comportamento Logico ed elettrico di un Pin del Microcontrollore
	InitPin <i>arg1,arg2,arg3,arg4</i>	Configura un Pin del Micro per Ingresso o Uscita digitale
	SetPin <i>arg1,arg2</i>	Setta lo stato logico di un Pin del micro in uscita
	TogglePin <i>arg1</i>	Inverte lo stato logico di un Pin in uscita
	BranchIfPin <i>arg1,arg2,arg3</i>	Dirama secondo lo stato di un Pin in Ingresso
	WaitPinEvent <i>arg1,arg2,arg3</i>	Attende il cambiamento di stato di un Pin di ingresso digitale per un determinato tempo
	PulseOut <i>arg1,arg2,arg3,arg4</i>	Genera un impulso su un Pin di uscita digitale
	PulseIn <i>arg1,arg2,arg3,arg4</i>	Misura la durata di un Impulso su un Pin di Ingresso digitale
	ClearPinIrq <i>arg1</i>	Resetta il Pin IRQ specificato e lo riarma per il successivo evento.
ADC Convertitore Analogico Digitale	ConfigADC <i>arg1,arg2</i>	Configura il Modulo ADC
	InitADC	Inizializza il Modulo A/D Converter della MCU
	GetADC <i>arg1,arg2,arg3</i>	Acquisisce un valore analogico in un determinato canale con filtro media campioni
	CaptureADC <i>arg1,arg2</i>	Cattura un valore analogico con singola conversione.
	GetChipVdd <i>arg1,arg2</i>	Acquisisce il valore della tensione di alimentazione del Chip MCU
CTS Chip Temperature Sensor	GetChipTemp <i>arg1</i>	Acquisisce la Temperatura del sensore interno alla MCU
TPM Timer Base Time	InitTPM <i>arg1</i>	Inizializza il Timer Pwm Module
	ConfigTPM <i>arg1,arg2,arg3,arg4,arg5</i>	Configura il Timer Pwm Module per operare con una modalità prestabilita.
	ClearTimerIRQ <i>arg1</i>	Resetta il Flag di Overflow Interrupt

LISTA FUNZIONI MODULI INTERNI *continuazione ..*

MODULO	NOME MACRO	FUNZIONE
PWM <i>Timer Pulse Wide Modulator</i>	ConfigPWM <i>arg1,arg2,arg3,arg4</i>	Configura il PWM per operare con una modalità prestabilita.
	InitPWM <i>arg1,arg2,arg3</i>	Inizializza il PWM impostando un Duty Cycle di default
	PWMduty <i>arg1,arg2,arg3</i>	Cambia il Duty Cycle del PWM
OCMP <i>TIMER Output Compare</i>	ConfigTPMchannel <i>arg1,arg2,arg3,arg4</i>	Configura il Canale del TPM
	TriggTimerDelay <i>arg1,arg2,arg3,arg4,arg5</i>	Attiva un Timer per la produzione di impulsi o per gestire eventi di tempo tramite Interrupt del Canale del Timer
	ClearTimerIRQ <i>arg1,arg2</i>	Resetta l'Interrupt di canale
ICAP <i>TIMER Input Capture</i>	ConfigTPMchannel <i>arg1,arg2,arg3,arg4</i>	Configura il Canale del TPM
	CaptureTimerEvent <i>arg1,arg2,arg3,arg4</i>	Cattura la durata fra transizioni di stato in un pin del canale del Timer usando gli Interrupt del Canale del Timer
	ClearTimerIRQ <i>arg1,arg2</i>	Resetta l'Interrupt di canale
FLASH <i>Data Storage</i>	FlashDataBackUp <i>arg1,arg2</i>	Effettua il BackUp di Dati in RAM nella FLASH
	FlashDataRestore <i>arg1,arg2</i>	Recupera i Dati dei parametri dalla FLASH E li deposita nella RAM
	CreateParametersData	Crea l'area parametri da salvare in FLASH
	DefineParam <i>arg1,arg2,arg3</i>	Definisce i Parametri da salvare
	CloseParametersData	Chiude l'area parametri
ICS <i>Internal Clock Source</i>	Define <i>TRIMICS,\$XX</i> DefinePin <i>PTxy,FCPIN</i>	Procedura di Calibrazione dell'oscillatore interno

LISTA FUNZIONI MODULI INTERNI <i>continuazione ..</i>		
MODULO	NOME MACRO	FUNZIONE
SGEN <i>TIMER SIGNAL</i> <i>Generazione di segnali</i>	InitSGEN <i>arg1,arg2</i>	Configura il Modulo Segnali
	OutSignal <i>arg1,arg2,arg3</i>	Genera un onda quadra sul Canale indicato alla frequenza desiderata.
	OutBurst <i>arg1,arg2,arg3,arg4, arg5,arg6</i>	Genera N segnali di Burst di una determinata frequenza
	OutSweep <i>arg1,arg2,arg3,arg4, arg5,arg6</i>	Genera un segnale a frequenza variabile
TCAP <i>TIMER CAPTURE</i> <i>Cattura segnali</i>	InitTCAP <i>arg1</i>	Inizializza la modalità per cattura segnali
	ConfigTCAP <i>arg1</i>	Configura il Timer per la cattura di segnali
	PulseIn <i>arg1,arg2,arg3,arg4</i>	Misura la durata di un impulso (Vedi <i>DIGITAL I/O</i>)
IIC <i>IIC Bus</i> <i>Inter Integrate Communication Bus</i>	InitIIC	Inizializza il Modulo IIC Bus
RTCM <i>Real Time Counter Module</i> <i>(QE – SH version)</i>	InitRTCM <i>arg1</i>	Inizializza il Modulo RTCM
	ConfigRTCM	Configura il Modulo RTCM
	ClearRTCMirq	Resetta l' RTCM IRQ e lo riarma per il successivo evento di tempo.

Differenze fra l'utilizzo del modulo ICAP e TCAP

Il Modulo ICAP (Input Capture) e TCAP (Timer Capture) consentono entrambi di intercettare eventi di tempo come transizioni di stato sui PIN. Le differenze sono:

ICAP : Intercetta transizioni di stato logico solo sui PIN che accedono al Timer : TPMxCHy

La durata di tempo fra le transizioni di stato non richiede l'attesa degli eventi di transizione.

TCAP : Intercetta transizioni di stato logico su qualsiasi PIN definito come Ingresso Logico.

La durata di tempo fra le transizioni di stato logico richiede l'attesa degli eventi di transizione.

TCAP viene usato con la funzione **PulseIn** per la misura di Impulsi in µs.

LISTA FUNZIONI MODULI INTERNI <i>continuazione ..</i>		
MODULO	NOME MACRO	FUNZIONE
SCOM <i>Serial Communication</i> <i>Module</i> SCI	ScomSCIport <i>arg1</i>	Definisce la porta seriale di comunicazione
	InitSerialCom <i>arg1</i>	Inizializza la porta seriale con un dato baudrate
	ServiceSCOM <i>arg1</i>	Gestisce i dati ricevuti dalla porta seriale
	PrintSCOMtext <i>arg1,arg2,arg3,arg4,arg5</i>	Invia alla seriale dei caratteri del testo o caratteri
	PrintSCOMttext <i>arg1,arg2,arg3</i>	Invia alla seriale del testo prelevato da tabella
	PrintSCOMvalue <i>arg1,arg2,arg3,arg4,arg5</i>	Invia alla seriale dei valori numerici
	PrintSCOMbool <i>arg1,arg2,arg3,arg4,arg5</i>	Invia alla seriale un testo alternativo a seconda dello stato di un Bit
	GetSCOMvalue <i>arg1,arg2,arg3,arg4</i>	Estrae un valore di qualsiasi datatype dal Buffer di comunicazione
	SCOM <i>Variabili accessibili</i>	Lista variabili riservate accessibili

DIO		DIGITAL I/O PIN : Linee di Input Output digitale		
		DefinePin <i>arg1,arg2,arg3,arg4</i>		
Definisce il comportamento logico ed elettrico di un PIN del Microcontrollore				
ProtoType		DefinePin PinType,PinName,ISR		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	PinType	Label	Nome del pin come definito nel datasheet del MCU PTxy .. per Linee di I/O digitale KBIPy, KBxPy .. per Linee con Interrupt Keyboard Plxy , .. per linee con interrupt generiche ADP ... per linee analogiche
		PinName	Label	Se PinType = PTxy o KBIPy o KBxPy o Plxy Nome che si vuole attribuire al PIN Se PinType = ADP Numero del canale analogico del ADC : 0,1,2.....,Cmax
		ISR	Label	Se PinType = PTxy : ISR è vuoto (non inserito) Se PinType = KBxPy o Plxy Indirizzo o etichetta della ISR (Interrupt Service Routine) da servire nel caso di Interrupt Se PinType = ADP : Nome da attribuire al canale analogico
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Va inserita nel Modulo 'SETTINGS' <i>Attenzione: Pin dello stesso Modulo KBIx / Plx condividono la stessa ISR</i>		
ESEMPI		<div>DefinePin PTB5,ECHO ← Definisce il Pin PTB5 come linea digitale chiamata ECHO</div> <div>DefinePin KBIP2,ECHO ← Definisce il Pin KBIP2 come linea digitale in ingresso con attivazione di Interrupt.</div> <div>DefinePin KBI1P2,ECHO,service ← Definisce il Pin KBI1P5 come linea digitale in ingresso con attivazione di Interrupt e dichiara la routine di nome 'service' come ISR .</div> <div>DefinePin PIA2,ECHO,alarm ← Definisce il Pin PIA2 come linea digitale in ingresso con attivazione di Interrupt e dichiara la routine di nome 'alarm' come ISR .</div> <div>DefinePin ADP,5,Voltage ← Definisce il Pin ADP canale 5 come linea analogica e la chiama 'Voltage'</div>		

DIO		DIGITAL I/O PIN : Linee di Input Output digitale		
		InitPin <i>arg1,arg2,arg3,arg4</i>		
Configura un pin del microcontrollore per Ingresso o Uscita Digitale				
ProtoType		InitPin PinName,Direction,Level,Edge		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	PinName	Label	Nome del pin come da definizione nel modulo 'Settings'
		Direction	Label	Direzione logica o Modo : IN ← Il pin viene impostato come INGRESSO
				OUT ← Il pin viene impostato come USCITA
				IRQ ← Il pin viene impostato come INGRESSO con generazione di Interrupt.
	Level	Label	Se Direction è IN ← Level indica se deve essere inserito Il resistore di PullUp o meno : PULLUP : Inserisce il Pull Up FLOAT : L'ingresso e floating	
			Se Direction è OUT← Level indica il livello di default della linea : LOW : Stato logico basso HIGH : Stato logico alto	
	Edge	Label	Se Direction è IRQ ← Level indica se deve essere inserito Il resistore di PullUp o meno : PULLPIN : Inserisce il Resistore di PullUp o PullDown FLOAT : L'ingresso e floating	
Usato solo se Direction è IRQ. RISING ← Il Pin genera l'interrupt solo sul fronte di salita FALLING ← Il Pin genera l'interrupt solo sul fronte di discesa LOW ← Il Pin genera l'interrupt sul livello basso HIGH ← Il Pin genera l'interrupt sul livello alto				
RITORNA	-	-	-	
	A,H,X	I8	Invariati	
NOTE		Occorre che nel Modulo 'SETTINGS' sia definito il Pin : Es. DefinePin PTA0,TRIGGER Define PTC5,PUSH1... Vedi : DefinePin Va inserito solo nel Modulo 'BOOT'		
ESEMPI		InitPin TRIGGER,OUT,LOW ← Configura il Pin TRIGGER in uscita con livello basso InitPin PUSH1,IN,PULLUP ← Configura il Pin PUSH1 in ingresso con Res. Di Pull Up InitPin ECHO,IRQ,PULLPIN,FALLING ← Configura il Pin ECHO predisposto per generare interrupt sui fronti di discesa ed inserisce il resistore di PullUp InitPin ECHO,IRQ,PULLPIN,RISING ← Configura il Pin ECHO predisposto per generare interrupt sui fronti di salita ed inserisce il resistore di PullDown InitPin ECHO,IRQ,FLOAT,HIGH ← Configura il Pin ECHO predisposto per generare interrupt sul livello alto e non inserisce il resistore di Pull		

DIO		DIGITAL I/O PIN : Linee di Input Output digitale		
		ClearPinIrq <i>arg1</i>		
Resetta il Pin IRQ specificato e lo riarma per il successivo evento				
ProtoType		ClearPinIrq PinName		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	PinName	Label	Nome del pin come da definizione nel modulo 'Settings'
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Il Pin deve essere Definito e Inizializzato : Vedi: DefinePin e InitPin		
		Questa Istruzione va inserita solo alla fine della ISR (Interrupt Service Routine) prima dell'istruzione RTI.		
		ATTENZIONE : Nel caso più PIN condividano la stessa ISR va inserito il nome di uno qualsiasi dei PIN che generano l'IRQ.		
ESEMPI		ClearPinIrq ECHO ← Resetta l'IRQ generato dal Pin ECHO		

DIO		DIGITAL I/O PIN : Linee di Input Output digitale		
		SetPin arg1,arg2		
Setta lo stato logico di un Pin in uscita del Microcontrollore				
ProtoType		SetPin PinName,Status		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	PinName	Label	Nome del pin come da definizione nel modulo 'Settings'
		Status	Label	Livello logico da impostare : HIGH / LOW
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Occorre che nel Modulo 'SETTINGS' sia definito il Pin : Es. DefinePin PTA0,TRIGGER Nel Modulo 'BOOT' il Pin sia inizializzato : Es. InitPin TRIGGER,OUT,LOW		
ESEMPI		SetPin TRIGGER,HIGH ←Setta il Pin TRIGGER a stato logico Alto		

DIO		DIGITAL I/O PIN : Linee di Input Output digitale		
		TogglePin <i>arg1</i>		
Inverte lo stato logico di un Pin in uscita del Microcontrollore				
ProtoType		TogglePin PinName		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	PinName	Label	Nome del pin come da definizione nel modulo 'Settings' Nel quale invertire lo stato Se lo stato di uscita era 1 diventa 0 e viceversa.
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Occorre che nel Modulo 'SETTINGS' sia definito il Pin : <i>Es. DefinePin PTA0,TRIGGER</i>		
ESEMPI		TogglePin TRIGGER,LOW ← Inverte lo stato del Pin TRIGGER		

DIO		DIGITAL I/O PIN : Linee di Input Output digitale		
		BranchIfPin <i>arg1,arg2,arg3</i>		
Dirama ad una etichetta a seconda dello stato logico di un Pin in Ingresso				
ProtoType		BranchIfPin PinName,Level,Address		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	PinName	Label	Nome del pin come da definizione nel modulo 'Settings'
		Level	Label	Livello logico da verificare per la diramazione: HIGH / LOW
		Address	Label	Etichetta/Indirizzo dove diramare se verificato lo stato
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Occorre che nel Modulo 'SETTINGS' sia definito il Pin : <i>Es. DefinePin PTA0,PUSH</i>		
ESEMPI		BranchIfPin PUSH,LOW,star57 ← Dirama a Star57 se il Pin PUSH è basso		

DIO		DIGITAL I/O PIN : Linee di Input Output digitale		
		WaitPinEvent <i>arg1,arg2,arg3</i>		
Attende per un determinato tempo il cambiamento di stato di un Pin di ingresso digitale				
ProtoType		WaitPinEvent PinName,Level,Timeout		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	PinName	Label	Nome del pin come da definizione nel modulo 'Settings'
		Level	Label	Livello logico da verificare per la diramazione: HIGH / LOW
		Timeout	U16	Tempo di attesa in ms che si verifichi l'evento
	RITORNA	C (CCR)	Flag	0 ← Cambiamento di stato avvenuto regolarmente 1 ← Cambiamento di stato non avvenuto nel timeout dichiarato
		A,H,X	I8	Invariati
NOTE		Occorre che nel Modulo 'SETTINGS' sia definito il Pin : <i>Es. DefinePin PTxx,ECHO ... dove xx è la lettera ed il numero del pin della porta</i>		
ESEMPI		WaitPinEvent ECHO,HIGH,50 ← Attende per 50 ms che il Pin ECHO vada alto		

DIO		DIGITAL I/O : Linee di Input Output digitale		
		PulseOut <i>arg1,arg2,arg3,arg4</i>		
Genera un impulso su un Pin di uscita digitale				
ProtoType		PulseOut PinName, Level,Width,Tbase		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	PinName	Label	Nome del pin come da definizione nel modulo 'Settings'
		Level	Label	Livello dell'Impulso : HIGH / LOW
		Width	U16	Indica la durata dell' impulso : 0..50000 µs/ms
		Tbase	Label	MICROSEC for microseconds base time MILLISEC for milliseconds base time
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		<p><i>Occorre che nel Modulo 'SETTINGS' sia definito il Pin :</i></p> <p><i>Es. DefinePin PTA0,TRIGGER DefinePin PTC5,STROBE...</i></p> <p><i>Occorre che nel Modulo 'BOOT' sia definita la modalità del Pin :</i></p> <p>Es. InitPin TRIGGER,OUT,LOW ← Dichiaro il Pin TRIGGER in uscita con livello basso</p> <p>Es. InitPin STROBE,OUT,HIGH ← Dichiaro il Pin STROBE in uscita con livello alto</p>		
ESEMPI		<p>PulseOut TRIGGER,HIGH,100,MICROSEC ← Genero un impulso alto di 100 µs sul Pin TRIGGER</p> <p>PulseOut STROBE,LOW,10,MILLISEC ← Genero un impulso basso di 10 ms sul Pin STROBE</p>		

DIO		DIGITAL I/O : Linee di Input Output digitale		
		PulseIn arg1,arg2,arg3,arg4		
Misura la durata di un impulso su un Pin di ingresso digitale				
ProtoType		PulseIn PinName, Level,Timeout,Width		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	PinName	Label	Nome del pin come da definizione nel modulo 'Settings'
		Level	Label	Livello dell'Impulso : HIGH / LOW
		Timeout	U16	Timeout in ms in attesa dell'impulso
		Width	U32	Variabile dove depositare il valore della durata dell' impulso in μs : 0... 2,2E6 us
	RITORNA	opera..+3	U32	Stesso valore di 'Width'
		Carry (CCR)	Flag	0: Impulso presente e 'catturato' 1: Timeout
		A,H,X	I8	Invariati
NOTE		Necessita dei moduli di Libreria:		
		LoadLibModule TPM ; Modulo Timer		
		LoadLibModule TCAP ; Timer Capture Module		
		Occorre che nel Modulo 'SETTINGS' sia definito il Pin :		
		Es. Define PTA0,ECHO		
		Occorre che nel Modulo 'BOOT' sia definita la modalità del Pin :		
		Es. InitPin ECHO,IN,FLOAT ← Dichiaro il Pin ECHO in ingresso Floating		
ESEMPI		PulseIn ECHO,HIGH,10,wpulse ← Misura la durata di un impulso alto sul pin ECHO		
		PulseIn ECHO,LOW,10,wpulse ← Misura la durata di un impulso basso sul pin ECHO		
		.. L'impulso viene aspettato per 10 ms in entrambi i casi		

ADC		Convertitore Analogico Digitale		
		ConfigADC <i>arg1,arg2</i>		
Configura il Modulo ADC				
ProtoType		ConfigADC NBits,Filter		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	NBits	Label	8BIT : Imposta il Modulo con 8 Bits di risoluzione 10BIT : Imposta il Modulo con 10 Bits di risoluzione 12BIT : Solo su determinate MCU
		Filter	U08	Valore compreso fra 0 e 7 : Effettua $2^{(Filter)}$ Conversioni e media i valori
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita del modulo di Libreria: Vedi InitADC		
ESEMPI		ConfigADC 10BIT,5 ← Configura il Convertitore A/D a10 Bit mediando 32 valori		
		ConfigADC 8BIT,0 ← Configura il Convertitore A/D a 8 Bit acquisendo un valore		

ADC		Convertitore Analogico Digitale		
		InitADC		
Inizializza il Modulo ADC				
ProtoType		InitADC		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	-		Nessun Argomento
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		<p>Necessita del modulo di Libreria ... e della Configurazione</p> <p style="text-align: center;"> LoadLibModule ADC ; Modulo A/D converter ConfigADC NBits,Filter ; Configurazione </p> <p>Nel Modulo Settings devono essere definite I Pin utilizzati come ingressi analogici: Es.</p> <p style="text-align: center;"> DefinePin ADP,2,SONDA ; Imposta Definisce il Pin ADP2 come Ingresso analogico e lo chiama SONDA </p>		
ESEMPI		InitADC ← Inizializza il Convertitore A/D		

ADC		Convertitore Analogico Digitale		
		GetADC <i>arg1,arg2,arg3</i>		
Ottiene il valore analogico dal Modulo ADC				
ProtoType		GetADC Canale,ADvalue,Mode		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Canale	U08	Canale analogico da acquisire : 0..ACH_MAX Vedere Data sheet Micro per i canali disponibili
		ADvalue	U16	Variabile in cui sarà depositato il valore analogico
		Mode	Label	Se manca attiva per default il Modo 'N' N : Normal mode no compensazione variazioni Vdd VDD : Compensazione delle variazioni di VDD
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita del modulo di Libreria: Vedi InitADC		
ESEMPI		<div>GetADC 3,speed ← Acquisisce valore da Canale 3 e lo deposita in speed</div> <div>GetADC 2,Traw,VDD ← Acquisisce canale2 a 8 Bit comp Vdd in Traw</div>		

ADC		Convertitore Analogico Digitale		
		CaptureADC <i>arg1,arg2</i>		
Cattura un valore analogico dal Modulo ADC				
ProtoType		CaptureADC Canale,ADvalue		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Canale	U08	Canale analogico da acquisire : 0..ACH_MAX Vedere Data sheet Micro per i canali disponibili
		ADvalue	U16	Se presente: Variabile in cui sarà depositato il valore analogico
	RITORNA	adcr..+1	U16	Registro MCU del valore analogico
		A,H,X	I8	Invariati
NOTE		Necessita del modulo di Libreria: Vedi InitADC		
ESEMPI		CaptureADC 3,speed ← Acquisisce valore da Canale 3 e lo deposita in speed GetADC 2 ← Acquisisce canale2 e lascia valore in adcr:adcr+1		

ADC		Convertitore Analogico Digitale		
		GetChipVdd arg1		
Acquisisce il valore della tensione di alimentazione del Chip MCU				
ProtoType		GetChipVdd Vdd		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Vdd	U16	Variabile in cui depositare il valore della tensione di alimentazione. Il valore ha 2 decimali
	RITORNA	opera..+3	U32	Valore della tensione del MCU
		A,H,X	I8	Invariati
NOTE		Necessita dei moduli di Libreria: LoadLibModule ADC ; Modulo A/D converter		
		<i>Nel Boot deve essere inserita la linea: InitADC NBits,Inputs,Filter Anche se il Modulo ADC non viene usato esternamente. In questo caso : Nbits deve essere 10 Input : 0 Filter : 7</i>		
ESEMPI		GetChipVdd Vchip ← Acquisisce la Tensione di alimentazione e al deposita in Vchip		

CTS		Chip Temperature Sensor		
		GetChipTemp arg1		
Acquisisce il valore di temperatura del sensore interno alla MCU				
ProtoType		GetChipTemp Var		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Var	U08	Variabile in cui depositare il valore di temperatura. La risoluzione è al grado : -10..+40 A ← Deposita il valore solo nell'accumulatore
		A	U08	Valore Temperatura
	RITORNA	opera..+3	U32	Valore Temperatura
		A,H,X	I8	Invariati
NOTE		Necessita dei moduli di Libreria: LoadLibModule ADC ; Modulo A/D converter LoadLibModule CTS ; Modulo Chip Temperature Sensor		
		Nel Boot deve essere inserita la linea: InitADC NBits,Inputs,Filter Anche se il Modulo ADC non viene usato esternamente. In questo caso : Nbits può essere 8 o 10 Input : 0 Filter : 7		
ESEMPI		GetChipTemp Ctemp	←	Acquisisce la temperatura del CTS e la deposita in Ctemp
		GetChipTemp A	←	Acquisisce la temperatura del CTS e lascia il valore solo in A

TPM		Timer Pwm Module		
		ConfigTPM <i>arg1,arg2,arg3,arg4,arg5</i>		
Configura il Modulo Timer per operare con una modalità prestabilita				
ProtoType		ConfigTPM Timer,ClkSrc, Mode,Frequenza, Prescaler		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		ClkSrc	Label	INT ← Clock Timer Interno EXT ← Clock Timer esterno
		Mode	Label	DEF : Default Mode. Usato quando si desidera attivare sia il PWM che il Timer Capture. In questo modo il PWM è del tipo Edge Aligned con ha frequenza fissa di 115Hz e la risoluzione del Timer è di 1 us PWME : Configura Timer per PWM Edge aligned In questa modalità il Timer gestisce solo il PWM PWMC : Configura Timer per PWM Center Aligned (Controllo Motori) In questa modalità il Timer gestisce solo il PWM TCOC : Configura il Timer con una modalità custom TWIN : Configura Timer per Finestre di Tempo
		Frequenza / Modulo	U16	Mode: DEF : Non usato, omettere il valore Mode: PWME/C : Frequenza del segnale PWM Mode: TCOC : Modulo del Timer : 0...65535 Mode: TWIN : Tempo finestra in µsec : 1...2000
		Prescaler	U16	Mode: DEF : Non usato, omettere il valore Mode: PWME/C : Non usato, omettere il valore Mode: TCOC : Prescaler del Timer. 0...7 Mode:TWIN : Non usato, omettere il valore
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		<i>Dopo il caricamento del Modulo TPM, inserire il, ConfigTMP:</i> LoadLibModule TPM ; Modulo Timer Base Time ConfigTPM parametri <i>In modalità PWME/C, se più canali sono configurati per generare il PWM, questi avranno la stessa frequenza.</i>		
ESEMPI		ConfigTPM 1,INT,DEF ← Configura il TIMER 1 per l'uso del PWM e Cattura Impulsi ConfigTPM 2,INT,PWME,1000, ← Configura il TIMER 2 PWM per l'uso come PWM edge Aligned con frequenza di 1000 Hz ConfigTPM 2,TWIN, 1000, ← Configura il TIMER 2 in modalità free run con finestre Di 1000 µsec		

TPM		Timer Pwm Module		
		ConfigTPMchannel <i>arg1,arg2,arg3,arg4</i>		
Configura il Canale del TPM per operare con una modalità prestabilita				
ProtoType		ConfigTPMchannel Timer,Canale,Param,Value		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
		Param	Label	Parametro del Canale del TPM da impostare. Parametri disponibili: ISR : Definisce la Interrupt Service Routine
		Value	Label	Nome della Interrupt Service Routine da servire
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: <i>LoadLibModule TPM ; Modulo Timer Base Time</i>		
ESEMPI		ConfigTPMchannel 1,0,ISR,Timer.isr ← Configura il Canale 0 del Timer 1 per la generazione di Interrupt serviti alla ISR Timer.isr		

TPM		Timer Pwm Module		
		InitTPM <i>arg1</i>		
Inizializza il Modulo				
ProtoType		InitTPM Timer		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: Vedi ConfigTPM		
ESEMPI		InitTPM 1 ← Inizializza il Timer 1		

TPM		Timer Pwm Module		
		ConfigTPMchannel arg1,arg2,arg3,arg4		
Configura il canale TPM				
ProtoType		ConfigTPMchannel Timer,Canale,Mode,Value		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
		Mode	Label	Parametro del Canale da impostare. Parametri disponibili: ISR : Definisce il nome della ISR
		Value	Label	Indirizzo/Etichetta della I.S.R.
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: ConfigTPM		
ESEMPI		ConfigTPMchannel 1,0,ISR,service.isr ← Configura il PWM Timer 1 Chn 0 a generare un Interrupt servito dalla ISR : service.isr		

TPM		Timer Pwm Module		
		ClearTimerIRQ <i>arg1,arg2</i>		
Resetta i Flags di Interrupt del Timer, riarmandoli per il successivo IRQ				
ProtoType		ClearTimerIRQ Timer,Canale		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Se Manca : Resetta il Flag di IRQ del Timer Se Presente Resetta il Flag di IRQ del Canale del Timer
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Il Timer o il Canale del Timer deve gestire l'Interrupt		
ESEMPI		ClearTimerIRQ 2,3 ← Resetta il Flags di IRQ del Timer 2, Canale3 ClearTimerIRQ 2 ← Resetta il Flags di IRQ del Timer 2 (Overflow)		

PWM		TimerPWM		
		ConfigPWM <i>arg1,arg2,arg3,arg4</i>		
Configura il PWM per operare con una modalità prestabilita				
ProtoType		ConfigPWM Timer,Canale,Param,Value		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
		Param	Label	Parametro del PWM da impostare. Parametri disponibili: SCALE : Imposta la scala del PWM
		Value	U16	Valore della Scala del PWM: 10...10000
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: Vedi InitPWM		
		Dopo il caricamento del Modulo PWM inserire il, ConfigPWM: LoadLibModule PWM ; Modulo PWM Base Time ConfigPWM parametri Se ConfigPWM non è inserito, Il PWM adotta una scala di default pari a 10000		
ESEMPI		ConfigPWM 1,0,SCALE,100 ← Configura il PWM Timer 1 Chn 0 con una scala 0..100 ConfigPWM 1,2,SCALE,2500 ← Configura il PWM con una scala 0..2500		

PWM		TimerPWM		
		InitPWM arg1,arg2,arg3		
Inizializza il Modulo Timer con Duty Cycle di Default				
ProtoType		InitPWM Timer,Canale,Duty		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
		Duty	U16	Valore di Default del DUTY : 0.. 1000 (parti per mille)
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: LoadLibModule TPM ; Modulo Timer Base Time LoadLibModule PWM ; Modulo PWM Occorre inizializzare nel Boot anche il modulo TPM : Vedi TPM module !!!		
ESEMPI		InitPWM 1,0,500 ← Attiva PWM Timer 1 Canale con 50.0% Duty Cycle InitPWM 1,2,50 ← Attiva PWM Timer 1 Canale 2 con 5.0% Duty Cycle		

PWM		Timer PWM		
		PWMduty arg1,arg2,arg3		
Regola il Duty Cycle del segnale PWM				
ProtoType		PWMduty Timer,Canale,Duty,Scale		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
		Duty	U08/U16 Label	Duty può essere : - Costante : Valore numerico a 16 Bits fisso - Variabile : Valore numerico a 8 o 16 bits in una variabile - A : Valore numerico a 8 Bit nell'Accumulatore - H:X : Valore Numerico a 16 Bits nei Registri H:X Il massimo valore di Duty è per default di 1000. Se ConfigPWM SCALE è stato definito, allora il Duty può variare fra : 0...SCALE. Se SCALE < 255 può essere usato sia A o una variabile ad 8 Bits oppure H:X
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: Vedi InitPWM		
ESEMPI		PWMduty 1,0,300 ← Varia il Duty al 30.0% nel Canale 0 del Timer 1 PWMduty 1,1,pduty ← Varia il Duty al valore in pduty nel Canale 1 del Timer 1 PWMduty 1,1,H:X ← Imposta il Duty con il valore contenuto nel registro H:X PWMduty 1,1,A ← Imposta il Duty con il valore contenuto nel registro A		

OCMP		Timer Output Compare		
		TriggTimerDelay <i>arg1,arg2,arg3,arg4,arg5</i>		
Attiva un Timer Delay				
ProtoType		TriggTimerDelay Timer,Canale,Delay,Mode,Option		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
		Delay	U16	Se Delay è una costante, allora esprime il ritardo in µsec Compreso far 10...65535 µs Se Delay è una variabile esprime il ritardo in unità di Base Tempi normalmente 0.868 µs La Base Tempi del Timer è espressa nella costante TPM<T>_BASE dove T è il numero del Timer usato Questa costante dipende dai parametri Modulo e Prescaler della funzione ConfigTPM se Modulo= 0 (65536) e Prescaler=8 vale 868 nsec
		Mode	Label	Comportamento del Pin TMPxCHy : CLEAR : Dopo il Delay il Pin va Basso SET : Dopo il Delay il Pin va alto TOGGLE : Dopo il Delay il Pin si inverte NONE : Nessuna azione sul Pin, solo generazione di Interrupt se attivato.
		Option	Label	Mancante : L'ISR dichiarata nel ConfigTPMchannel viene servita. NOISR : La ISR non viene servita e quindi non viene generato l'Interrupt. Questa opzione è utile qualora il Timer debba agire solo sullo stato del Pin.
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: TPM E la configurazione ConfigTPM e ConfigTPMchannel		
ESEMPI		<p>TriggTimerDelay 1,0,500,TOGGLE ← Triggera un ritardo di 500µs sul Canale 0 del Timer 1, alla fine del quale il Pin TPM1CH0 invertirà lo stato e sarà servita la ISR dichiarata in ConfigTPMchannel</p> <p>TriggTimerDelay 1,0,1200,SET,NOISR ← Triggera un ritardo di 1200µs sul Canale 0 del Timer 1, alla fine del quale il Pin TPM1CH0 andrà alto e NON sarà servita alcuna ISR.</p> <p>TriggTimerDelay 1,0,width,CLEAR, ← Triggera il ritardo contenuto nella variabile 'width' alla fine del quale il Pin TPM1CH0 andrà basso e sarà servita la ISR. Se la variabile 'width' contiene 500 il ritardo sarà di 1000x0,868 =868 µs se TPM1_BASE = 868ns</p>		

TCAP		Timer Capture		
		InitTCAP <i>arg1</i>		
Inizializza il Timer per la Cattura di segnali impulsivi				
ProtoType		InitTCAP Timer		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria:		
		<div>LoadLibModule TPM ; Modulo Timer Base Time</div> <div>ConfigTPM Timer ; Configura il Timer</div> <div>LoadLibModule TCAP ; Modulo Timer Capture</div> <div>ConfigTCAP Timer ; Configura il Timer Capture</div>		
		Es. Il Modulo Config può essere configurato: ConfigTPM 1,INT,DEF ← Configura il TIMER 1 per l'uso del PWM e Cattura Impulsi		
ESEMPI		InitTCAP 1 ← Inizializza il Timer 1 per la misura di impulsi		

TCAP		Timer Capture		
		ConfigTCAP <i>Timer</i>		
Configura la Modalità per Cattura segnali impulsivi				
ProtoType		ConfigTCAP Timer		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: Vedi InitTCAP		
ESEMPI		ConfigTCAP 1 ← Configura il Timer 1 per la cattura di impulsi di breve durata		

SGEN		Timer Signal		
		InitSGEN arg1,arg2		
Inizializza il Modulo Timer come Generatore di Impulsi				
ProtoType		InitSGEN Timer,Canale		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: LoadLibModule SGEN ; Modulo Signal generator ATTENZIONE: Il Modulo SGEN usa in esclusiva il Timer Indicato.Gli altri canali del Timer non possono quindi essere usati per generare altri segnali o per il PWM.		
ESEMPI		InitSGEN 1,0 ← Predispone il Canale 0 del Timer 1 a generare impulsi.		

SGEN		Timer Signal		
		OutSignal arg1,arg2,arg3		
Genera un segnale impulsivo ad una determinata frequenza				
ProtoType		OutSignal Timer,Canale,Frequenza		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
		Frequenza	U32	Valore della frequenza del segnale generato F = 2...1.000.000 Hz
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: Vedi InitSGEN		
ESEMPI		OutSignal 1,0,10000 ← Genera 10 KHz nel canale 0 del Timer 1 OutSignal 1,0,fsignal ← Genera la frequenza in fsignal nel canale 0 del Timer 1		

SGEN		Timer Signal		
		OutBurst arg1,arg2,arg3,arg4,arg5,arg6		
Genera un segnale impulsivo ad una determinata frequenza				
ProtoType		OutBurst Timer,Canale,Frequenza,NB,TimeON,TimeOFF		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
		Frequenza	U32	Valore della frequenza del Burst F = 2...1.000.000 Hz
		NB	U08	Numero di Burst da generare 1..255
		TimeON	U16	Durata del Burst in ms : 0...65535 ms
		TimeOFF	U16	Durata del silenzio del Burst in ms : 0...65535 ms
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: Vedi InitSGEN		
ESEMPI		OutBurst 1,0,10000,10,20,50 ← Genera 10 Burst da 10 KHz nel canale 0 del Timer 1 di durata 20ms e con un silenzio di 50 ms		
		OutBurst 1,0,fsignal,nburst,ton,toff ← Genera I burst in base al contenuto delle variabili		

SGEN		Timer Signal		
		OutSweep arg1,arg2,arg3,arg4,arg5,arg6		
Genera un segnale impulsivo ad una determinata frequenza				
ProtoType		OutSweep Timer,Canale,Fstart,FStep,Nstep,StepdDur		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Timer	U08	Modulo Timer usato : 1... (Vedi MCU)
		Canale	U08	Canale del Timer : 0... (Vedi MCU)
		Fstart	U32	Valore della frequenza di start dello sweep f = 2...1.000.000 Hz
		Fstep	S32	Valore della frequenza di incremento. Può essere anche negativo
		Nstep	U16	Numero di step di incremento : 1...65535
		StepDur	U16	Durata di ogni step : 0...65535 ms
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita dei seguenti moduli di Libreria: Vedi InitSGEN		
ESEMPI		OutSweep 1,0,1000,10,500,5 ← Genera uno sweep di frequenza a partire da 1000 Hz nel canale 0 del Timer 1, La frequenza incrementa di 10 Hz ogni 5 ms per 500 step di incremento. Lo Sweep parte quindi da 1000 Hz ed arriva a 6000Hz In un tempo di 2.5 sec.		
		OutSweep 1,0,5000,-10,400,5 ← Genera uno sweep di frequenza a partire da 5000 Hz nel canale 0 del Timer 1, La frequenza decrementa di 10 Hz ogni 5 ms per 400 step di incremento. Lo Sweep parte quindi da 5000 Hz ed arriva a 1000Hz In un tempo di 2 sec.		
		OutSweep 1,0,fsignal,fstep,nstep,dur ← Genera lo sweep in base al contenuto delle variabili		

IIC		IIC Bus		
		InitIIC		
Inizializza il Modulo IIC Bus				
ProtoType		InitIIC		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	-	-	Nessun argomento in ingresso
		-	-	-
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita del seguente modulo di Libreria: <div>LoadLibModule IIC ; Modulo IIC Bus</div> <div>Se la MCU lo consente e si desidera spostare il BusSDA,SCK negli altri Pins opzionali, Inserire nel modulo Settings la linea:</div> <div>Define IICPS , %1.SHL.1 ← Imposta IIC bus su PB6-PB7</div>		
ESEMPI		InitIIC ← Attiva il Modulo IIC Bus sui Pin previsti		

RTCM		Real Time Counter Module		
		InitRTCM <i>arg1</i>		
Inizializza il Modulo RTCM				
ProtoType		InitRTCM <i>Mode</i>		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Mode	Label	Se assente RTCM è operativo senza generare interrupt L'evento va verificato con la tecnica del Polling IRQ : RTCM genererà un interrupt servito dalla ISR dichiarata nella sezione Module da ConfigRTCM
		-	-	-
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		<p>Necessita del seguente modulo di Libreria:</p> <p>LoadLibModule <i>RTCM</i> ; <i>Modulo RTCM</i> ... e la configurazione ConfigRTCM ClkSource,Prescaler,ClkFrequency,ISR</p>		
ESEMPI		<p>InitRTCM IRQ : RTCM Genera Interrupt servito da ISR</p> <p>InitRTCM : RTCM non genera Interrupt. L'evento è segnalato dal Flag RTCSC_RTIF del Registro 'rtcsc' Usare la funzione BranchIfBit SET, RTCSC_RTIF,rtcsc,LabelToBranch per testare il Flag</p>		

RTCM		Real Time Counter Module		
		ConfigRTCM <i>arg1,arg2,arg3,arg4</i>		
Configura il Modulo RTCM				
ProtoType		ConfigRTCM <i>ClkSource,Prescaler,ClkFrequency,ISR</i>		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	ClkSource	Label	Sorgente del Clock LPO : 1 KHz Low Power Oscillator ERCLK : External Reference Clock source IRCLK0 : Internal Reference Clock Source PSC option 0 IRCLK1 : Internal Reference Clock Source PSC option 1
		Prescaler	U08	Valore indice del Prescaler : 0..15 (Vedi DataSheet MCU per il Fattore di divisione correlato: PSCDIV)
		ClkFrequency	U32	Frequenza del Clock Sorgente in Hz Se ClkSource = LPO allora ClkFrequency = 1000
		ISR	Label	Indirizzo della Interrupt Service Routine
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita del seguente modulo di Libreria: <i>Vedi InitRTCM</i> <i>L'RTCM IRQ viene generato ogni : $IRQfreq= ClkFrequency / PSCDIV / (M+1)$ Dove M è il valore del Modulo del RTCM counter. M = ($ClkFrequency/PSCDIV$) -1</i>		
ESEMPI		ConfigRTCM ERCLK,1,32768,RTCM.isr : Configura il Modulo RTCM con <ul style="list-style-type: none">- Sorgente di Clock Esterna- Indice del Prescaler : 1 <i>equivalente ad una divisione per 1024 (DataSheet)</i>- Frequenza del Clock : 32768 Hz- Quando viene generato l'IRQ viene servita l'ISR: RTCM.svc In questo caso il valore del Modulo counter di RTCM è : $M = (32768/1024)-1 = 31$ L' IRQ viene generato ogni : $IRQfreq = 32768/1024/(31+1) = 1 \text{ sec}$ ConfigRTCM LPO,1,1000,RTCM : Configura il Modulo RTCM senza generare IRQ prelevando il clock dall'oscillatore interno di 1 KHz		

RTCM		Real Time Counter Module		
		ClearRTCMirq		
Resetta l'RTCM IRQ e lo riarma per il successivo evento				
ProtoType		ClearRTCMirq		
		Argomento	Type	Descrizione
ARGOMENT I	ENTRATA	-		
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE				
ESEMPI		Esempio di codice per ISR che gestisce l'RTCM interrupt:		
		RTCM.isr : pshh ; Salva H ; Codice di servizio dell ' IRQ ClearRTCMirq pulh ; Recupera H rti ; Ritorna al Main		

SCOM		Serial Communication module		
		ScomSCIport arg1		
Seleziona e configura la Porta Seriale SCI usata per la comunicazione				
ProtoType		ScomSCIport SCI		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	SCI	U08	Definisce la Serial Communication Interface usata. Porta SCI usata : 1 - 2 Questa linea va inserita subito dopo aver caricato il Modulo di libreria SCOM.
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita del seguente modulo di Libreria: SCOM : LoadLibModule SCOM		
ESEMPI		LoadLibModule SCOM ; Carica il Modulo di libreria SCOM ScomSCIport 2 ; Definisce la Porta SCI 2 come porta di comunicazione del Modulo		

SCOM		Serial Communication module		
		InitSerialCom <i>arg1</i>		
Inizializza la porta di comunicazione del modulo SCOM				
ProtoType		InitSerialCom <i>BaudRate</i>		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	<i>BaudRate</i>	U32	Velocità della comunicazione in BAUD : da 150 baud a 115200 baud
		-	-	-
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Necessita del seguente modulo di Libreria: <i>SCOM : LoadLibModule SCOM</i> ... e la configurazione della porta SCI <i>ScomSCIport</i> (port)		
ESEMPI		InitSerialCom 57600 ; Inizializza la porta SCI alla velocità di 57600 baud		

SCOM		Serial Communication module		
		ServiceSCOM arg1		
Gestisce i Dati Ricevuti dalla porta di comunicazione del modulo SCOM				
ProtoType		ServiceSCOM Address		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Address	Label	Etichetta di indirizzo dove decodificare il frame di dati ricevuto.
		-	-	-
	RITORNA	-	-	-
		A,H,X	I8	Unknown
NOTE		<p>Necessita del seguente modulo di Libreria: ...vedi InitSerialCom</p> <p>ServiceSCOM può essere inserita nel MAIN o inclusa in un modulo/file caricato nella sezione Modules</p> <p>I dati ricevuti sono contenuti nel buffer di comunicazione all'etichetta: SCOM_buffer</p> <p>La lunghezza del Buffer può essere definita con : SCOM_RXBSIZE .equal (SIZE) Nella sezione: myData Se non definita, il valore di default è 40 bytes</p>		
ESEMPI		<p>ServiceSCOM RXdecoder ; Va alla subroutine RXdecoder se frame validi ricevuti</p> <p>RXdecoder:</p> <div><div>OnSamePatternExec "#W10",SCOM_buffer,rel1.off ; Switch Off ?</div><div>OnSamePatternExec "#W11",SCOM_buffer,rel1.on ; Switch On ?</div><div>OnSamePatternExec "#W20",SCOM_buffer,rel2.off ; Switch Off ?</div><div>OnSamePatternExec "#W21",SCOM_buffer,rel2.on ; Switch On ?</div><div>rts </div></div>		

SCOM		Serial Communication module		
		PrintSCOMtext <i>arg1,arg2,arg3,arg4,arg5</i>		
Invia alla porta di comunicazione SCOM dei caratteri/bytes				
ProtoType		PrintSCOMtext <i>Item</i>		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	<i>Item</i>	-	Possono essere stampati fino a 5 Item con vario ordine - Testo compreso fra apici : Es. "HELLO - A : Singolo Byte contenuto nell'accumulatore - Variabile : Singolo Byte contenuto in una variabile U08 - NL : se presente accoda alla fine del testo i codici ASCII CR+LF corrispondenti a NewLine
	RITORNA	-	-	-
		A,H,X	I8	Unknown
NOTE		Necessita del seguente modulo di Libreria: ...vedi InitSerialCom		
ESEMPI		PrintSCOMtext "Hello World" ; Manda alla seriale il testo 'Hello World' PrintSCOMtext "Hello World",NL ; Manda alla seriale il testo e accoda New Line PrintSCOMtext A ; Manda alla seriale il byte contenuto nell'accumulatore. PrintSCOMtext "Numero: ",A, NL ; Manda alla seriale il testo : Numero:. affianca il numero ASCII in Accumul. E manda a Capo PrintSCOMtext " Numero.",A," di 10" ; Se A contiene \$35 ... Invia il messaggio : Numero:5 di 10		

SCOM		Serial Communication module		
		PrintSCOMttext <i>arg1,arg2, arg3</i>		
Invia alla porta di comunicazione SCOM delle stringhe contenute in una tabella				
ProtoType		PrintSCOMttext <i>Element, TableName , NewLine</i>		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	<i>Element</i>	U08	- Numero della stringa dati da inviare : 0...EleMax - A : Numero elemento contenuto nell'accumulatore
		<i>TableName</i>	Label	Indirizzo o etichetta dove inizia la tabella delle stringhe dati. Ogni stringa di dati deve avere come terminatore il byte \$00
		<i>NewLine</i>	Label	NL : se presente accoda alla fine del testo i codici ASCII CR+LF corrispondenti a NewLine
	RITORNA	-	-	-
		A,H,X	I8	Unknown
NOTE		Necessita del seguente modulo di Libreria: ...vedi InitSerialCom		
ESEMPI		DayWeek: Gennaio',\$00		

SCOM		Serial Communication module		
		PrintSCOMvalue <i>arg1,arg2,arg3,arg4,arg5</i>		
Invia alla porta di comunicazione SCOM un valore numerico formattato				
ProtoType		PrintSCOMvalue Type,Value,Digits,Decimals,NewLine		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Type	Label	Data Type del valore : U08 S08 U16 S16 U32 S32 FLOAT ANA
		Value	-	Variabile o costante contenente il valore numerico
		Digits	B08	I Bits B3...B0 indicano il numero delle cifre desiderate Compresi l'eventuale segno del valore ed il punto decimale se esiste. B4 : 1 ← Non recupera decimali nel caso di FLOAT B5 : 1 ← abilita la stampa del punto decimale B6 : 1 ← Toglie gli zeri piu' significativi: B7 : 1 ← Elimina lo Zero prima del punto decimale.
		Decimals	U08	Numero di decimali desiderati : 0...3 Se Decimals manca, viene inserito il punto in base al valore corrente della variabile 'point'
		NewLine	Label	NL : se presente accoda alla fine del testo i codici ASCII CR+LF corrispondenti a NewLine
	RITORNA	-	-	-
		A,H,X	I8	Unknown
NOTE		Necessita del seguente modulo di Libreria: ...vedi InitSerialCom		
ESEMPI		<p>PrintSCOMvalue S16,temp,%01100101,1 Stampa il valore di temp a 5 cifre con un decimale e segno sopprimendo gli zeri più sign</p> <p>PrintSCOMvalue S16,temp,%0110.SHL.4^5 ,1 ; <i>altro modo per la stessa istruzione !</i> (I bits di formattazione sono shiftati di 4 posti a sinistra con .SHL.4, mentre il numero 5 delle cifre è aggiunto con l'operatore OR ^</p> <p>PrintSCOMvalue U32,count,6,0 Stampa il valore di count a 6 cifre senza decimali lasciando gli zeri significativi.</p> <p>PrintSCOMvalue FLOAT,frequenza,%01110000^8,3 ; <i>Formattazione senza lo shift</i> Stampa il valore contenuto nella variabile frequenza di 8 cifre senza recuperare alcun decimale del valore float. Viene però piazzato il punto decimale alla terza cifra partendo da destra. <i>? Questa formattazione è utile nel caso dell'esempio se la frequenza è in Hz ma si vuole esprimerla in KHz.</i></p> <p>PrintSCOMvalue FLOAT,pressione,%01100000^6,2 Stampa il valore contenuto nella variabile Float pressione di 6 cifre, recuperando 2 decimali e piazzando quindi il punto sulla seconda cifra a partire da destra</p> <p>PrintSCOMvalue FLOAT,pressione,%01100000^4,-2 Stampa il val. FLOAT scalandolo di 10^2 senza DP</p>		

SCOM		Serial Communication module		
		PrintSCOMbool arg1,arg2,arg3,arg4,arg5		
Invia un messaggio di testo a seconda dello stato di un Bit				
ProtoType		PrintSCOMbool Bit,Location,Tfalse,Ttrue,NL		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Bit	U08	Bit da testare : 0..7 . Puo' essere una costante o variabile
		Location	U16	Locazione/registro contenente il bit
		Tfalse	Char	Testo fra doppi apici da stampare nel caso il Bit sia falso
		Ttrue	Char	Testo fra doppi apici da stampare nel caso il Bit sia vero
		NL	Labels	If Label "NL" present add a New line char
	RITORNA	-	-	-
		A,H,X	I8	Unknown
NOTE		Necessita del seguente modulo di Libreria: ...vedi InitSerialCom		
ESEMPI		<p>PrintSCOMbool SW1,SW1_port,"#L40","#L41" ; Invia il testo #L40 se il Bit SW1 di SW1_port è falso se vero invia il testo #L41.</p> <p>PrintSCOMbool 5,ptad,"CIAO","HELLO" ,NL Invia il testo CIAO o HELLO a seconda del Bit 5 di ptad e termina con NewLine</p> <p>PrintSCOMbool A,ptad,"CIAO","HELLO" ; Bit puntato dall' Accumulatore</p> <p>PrintSCOMbool star,ptad,"CIAO","HELLO" ; Bit puntato dalla variabile star</p>		

SCOM		Serial Communication module		
		GetSCOMvalue <i>arg1,arg2,arg3,arg4</i>		
Estrae un valore di qualsiasi datatype dal Buffer di comunicazione				
ProtoType		GetSCOMvalue Offset,Lenght,Type,Value		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	Offset	U08	Offset rispetto all'inizio del Buffer di Comunicazione: SCOM_buffer dove inizia la stringa numerica ASCII.
		Lenght	-	Lunghezza del valore numerico in caratteri : Es. se numero a 4 cifre 1276 > Lenght = 4 Es se numero a 5 cifre con segno e decimale : Es -999.99 Lenght = 7
		Type	B08	Data Type del valore da convertire e della variabile Value : U08 S08 U16 S16 U32 S32 FLOAT ANA
		Value	MCAST	Variabile definita da Type dove depositare il valore
	RITORNA	-	-	-
		A,H,X	I8	Unknown
NOTE		Necessita del seguente modulo di Libreria: ...vedi InitSerialCom		
ESEMPI		GetSCOMvalue 4,5,U08,valpwm Converte nella variabile valpwm del tipo U08 un testo numerico lungo al massimo 5 cifre Iniziando la scansione con un offset di 4 byte nel Buffer di comunicazione GetSCOMvalue 4,A,FLOAT,valpwm Converte nella variabile valpwm del tipo FLOAT il cui numero di cifre è contenuto nel Registro Accumulatore, iniziando la scansione con un offset di 4 byte .		

SCOM	Serial Communication module		
	Variabili riservate accessibili dal programmatore		
NOME VARIABILE	Type	Valore di default	Descrizione
SCOM_buffer	Address	-	Inizio del Buffer di Comunicazione
SCOM_RXBSIZE	Costante	40	Lunghezza del Buffer di Comunicazione in Bytes Può essere ridefinita: Es <code>SCOM_RXBSIZE .equal 80</code>
SCOM_msglen	U08	-	Lunghezza del messaggio ricevuto e presente nel Buffer di comunicazione
RXT_CHAR	Costante	LF (\$0A)	Terminatore del Messaggio ricevuto. Può essere ridefinito: Es <code>RXT_CHAR .equal CR</code> <code>RXT_CHAR .equal \$0d</code>
scomflg	U08 (FLAGS)	-	Flags di Stato Comunicazione
			FLAGS USO
			SCMR (b7) 1: Messaggio ricevuto nel buffer
			RXBF (b6) 1: Superamento caratteri nel Buffer

ICS	Internal Clock Source (Calibrazione)
	Procedure di Calibrazione dell'oscillatore interno
<p>Tutti i microcontrollori Freescall HCS08 possono funzionare senza quarzo esterno utilizzando l'oscillatore interno del modulo ICS. La frequenza di questo oscillatore è sufficientemente precisa per le normali operazioni. La frequenza di Bus del MCU è la metà di questa frequenza e può essere settata con la seguente definizione:</p> <p>Define <code>fbus,<frequenza></code> dove <code>frequenza</code> è un valore in Hz con Range che dipende dal Microcontrollore impiegato:</p> <p>HCS08QD4 : <code>fbus max = 8000000 + 10%</code> HCS08QG8 : <code>fbus max = 10000000 + 10%</code> HCS08AC60 : <code>fbus max = 20000000 + 10%</code> HCS08SH8 : <code>fbus max = 20000000 + 10%</code> HCS08QE32 : <code>fbus max = 25000000 + 10%</code></p> <p>Il valore inferiore deve essere ricavato dal Data Sheet ma in genere può arrivare anche alla metà del valore Max.</p> <p>Affinchè il Modulo TimeEvent delle procedure ad evento di tempo, possa generare eventi precisi, occorre che la frequenza di Bus dichiarata sia un multiplo di 1000 ovvero deve terminare con 000. Questo garantisce la precisione al msec degli eventi di tempo.</p> <p>Inoltre, se si utilizza la porta di comunicazione asincrona SCI, occorre che la frequenza di BUS sia anche divisibile per $/(16 \times \text{MaxBaudrate})$, deve cioè produrre un numero intero senza resto. Ad esempio impostando una frequenza di Bus di 9216000 Hz, si garantisce un Baudrate massimo di 115200 Baud : $9216000/(16 \times 115200) = 5$ oltre ad una precisione al ms del modulo TimeEvent.</p> <p>Per avere un valore preciso della frequenza di bus per applicazioni a tempo critico come comunicazioni seriali asincrone, ritardi di precisione ecc. deve essere calibrato anche l'oscillatore interno con la seguente procedura:</p> <p>Definire nel modulo SETTINGS un Pin libero o provvisoriamente libero per poter collegare la sonda di un oscilloscopio e chiamarlo FCPIN. Ad esempio: DefinePin PTB0,FCPIN</p> <p>Nel BOOT devono esserci le seguenti linee (<i>normalmente sempre presenti per default</i>):</p> <pre> ; ----- INIT FREQUENCY CAL PIN IF DEFINED ----- ; ; .ifdef FCPIN ; InitPin FCPIN,OUT,LOW ; Dispone Pin FCPIN in uscita default low ; .endif </pre> <p>Collegare un oscilloscopio tramite una sonda al Pin dichiarato ed impostarlo per visualizzare un onda quadra di circa 500 Hz.</p> <p>Tramite il Debugger NoICE, impostare un WATCH alla locazione ICSTRM o ICGTRM (AC serie) :</p> <p>ICSTRM : \$3A per famiglie QD QE QG \$4A per famiglia SH \$4E famiglia AC</p> <p>Modificare in più o meno di qualche unità o più, la locazione ICSTRIM a seconda che si voglia aumentare o diminuire la frequenza fino ad ottenere un periodo esatto di 2 ms corrispondente a 500Hz.</p> <p>Ricordare il valore esadecimale impostato nella locazione ed inserirlo nel modulo SETTING con la linea:</p> <p>Define TRIMICS,<icstrim> ; <i>Definisce la correzione per ICS</i></p> <p>Al Boot di sistema, la correzione del ICS verrà sempre settata a quel valore, garantendo la precisione della frequenza impostata. In alternativa all'Oscilloscopio può essere collegato anche un frequenzimetro con risoluzione al centesimo di Hz o più. Al termine dell' operazione, rimuovere la linea DefinePin PTB0,FCPIN o metterla come commento in modo da liberarla per altri scopi.</p>	

FLASH		MCU FLASH DATA STORAGE		
		FlashDataRestore <i>arg1,arg2</i>		
Recupera i Dati dalla FLASH e li salva nella RAM				
ProtoType		FlashDataRestore RAMadrs,Size		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	RAMadrs	U16	Indirizzo di partenza della RAM del Microcontrollore dove depositare i dati prelevati dalla FLASH
		Size	U16	Volume di Dati in bytes da recuperare
	RITORNA	-	-	-
		A,H,X	I8	Unknown
NOTE		Necessita del seguente modulo di Libreria da inserire in MODULES: LoadLibModule FLASH * Vedi FLASH CreateParametersData per uso con parametri predefiniti		
ESEMPI		FlashDataRestore \$0130,50 ; Recupera 50 Bytes dalla FLASH e li deposita in RAM a partire dalla locazione \$130 FlashDataRestore PARAM,10 ; Recupera10 Bytes dalla FLASH e li deposita in RAM a partire dalla locazione PARAM PARAM deve essere definita prima della chiamata.		

FLASH		MCU FLASH DATA STORAGE		
		FlashDataBackUp arg1,arg2		
Effettua il BackUp di Dati in RAM e li salva nella FLASH				
ProtoType		FlashDataBackUp RAMadrs,Size		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	RAMadrs	U16	Indirizzo di partenza della RAM del Microcontrollore dove salvare i dati
		Size	U16	Volume di Dati in bytes da salvare
	RITORNA	-	-	-
		A,H,X	I8	Unknown
NOTE		Necessita del seguente modulo di Libreria da inserire in MODULES : LoadLibModule FLASH * Vedi FLASH CreateParametersData per uso con parametri predefiniti		
ESEMPI		FlashDataBackUp \$0130,50 ; Salva 50 Bytes della RAM a partire dalla locazione \$130 FlashDataBackUp PARAM,10 ; Salva 10 Bytes della RAM a partire dalla loc. PARAM PARAM deve essere definita prima della chiamata.		

FLASH		MCU FLASH DATA STORAGE		
		CreateParametersData		
		DefineParam arg1,arg2,arg3		
		CloseParametersData		
Creazione e definizione blocco DATI da salvare nella FLASH				
Il BackUp su FLASH e Recupero Dati da FLASH nella RAM può essere fatto specificando prima nel Modulo STRUCTURES il Tipo e la quantità di Dati da salvare tramite la definizione di un blocco Parametri				
L'apertura e la chiusura del blocco Dati va fatta usando rispettivamente :				
CreateParametersData				
CloseParametersData				
che sono entrambe senza argomenti.				
ProtoType		DefineParam DataType,Name,Default		
		Argomento	Type	Descrizione
ARGOMENTI	ENTRATA	DataType	LABEL	I08 - I16 - I32 - FLOAT
		Name	LABEL	Nome del parametro / Variabile
		Default	Value	Valore di Default del Parametro salvato nella FLASH Può essere indicato in decimale,Esadecimale,Binario
	RITORNA	-	-	-
		A,H,X	I8	Invariati
NOTE		Vedi FlashDataBackUp per Moduli necessari. E' consentita una sola struttura di Parametri.		
ESEMPI		Nel Modulo STRUCTURES viene definita la struttura dei seguenti 6 parametri con diversi DataType CreateParametersData DefineParam U08,option,%00001010 DefineParam U08,mode,%11001100 DefineParam U16,speed,4000 DefineParam U16,setpoint,450 DefineParam U32,factor,2**24 DefineParam FLOAT,speedlight,300000 CloseParametersData Nel Modulo BOOT vengono recuperati i parametri dalla FLASH e inseriti nella RAM. In questo caso non serve alcun argomento alla funzione. FlashDataRestore Nel Modulo MAIN o altri files programma inclusi, vengono salvati i parametri nella FLASH Prelevandoli dalla RAM. Anche in questo caso non serve alcun argomento alla funzione. FlashDataBackUp ATTENZIONE: FlashDataBackUp deve essere chiamata solo quando i parametri sono stati modificati e si desidera mantenere le modifiche !! Scritture nella FLASH in modo eccessivamente ripetitivo degradano la memoria.		

FLASH	MCU FLASH DATA STORAGE		
	Variabili riservate accessibili dal programmatore		
NOME VARIABILE	Type	Valore di default	Descrizione
_MyParamData	U16	-	Indirizzo nella RAM alta dove iniziano i Parametri
BACKUP_RAM_SIZE	U16	-	Volume di Dati in RAM per il BackUp
_DSFRANK	U16	\$FA00	Indirizzo nella FLASH dove inizia il BackUp. Può essere ridefinito.