

+AD0APQ- +ACo-Documentazione su Great Cow BASIC+ACo-

Il team di sviluppo di Great Cow BASIC al 2019

+AD0APQ- +ACo-Introduzione al Great Cow BASIC+ACo-

Benvenuto nell'Help di Great Cow BASIC. Questo file di aiuto ha lo scopo di fornire approfondimenti e conoscenza per usare Great Cow BASIC. Per informazioni sull'installazione di Great Cow BASIC e diversi altri programmi che potrebbero essere utili, per favore vedi <http://url+AFs-Getting Started with Great Cow BASIC+AF0->.

Se sei nuovo alla programmazione, dovresti provare i programmi dimostrativi di Great Cow BASIC, spiega tutto in modo graduale e non presuppone conoscenze preliminari. Se hai programmato in un'altra lingua, puoi seguire passo passo, i programmi dimostrativi, e il modo migliore per imparare. Se c'è qualcos'altro su cui hai bisogno di aiuto, visita <http://url+AFs-sougeforge.net/forum?group+AF8-id+AD0-169286+AF0->

+ACo-Usare Great Cow BASIC+ACo- -Completa l'installazione utilizzando i valori predefiniti:-seleziona tutti i programmatori ma non quelli in modalita portatile. Il programma di installazione avvia automaticamente l'IDE. Quando viene aperto un file sorgente Great Cow BASIC, controlla il menu Strumenti IDE attraverso questo menu tu puoi accedere ai comandi con un click. Prova con il tasto destro del mouse: questo accedera alle stesse opzioni. I comandi IDE Tools ... (tasti funzione F5 - F8) avviano una utility e Great Cow BASIC chiama il file batch per la compilazione di codice sorgente e programmazione (+ACI-flash+ACI-)(1) del microcontrollore target. Di default questi file batch sono chiamati flashAVR.bat (quando il chipmodel e un AVR) e flashPIC.bat (quando il chipmodel e un PIC) e un file batch per creare e programmare il tuo microcontrollore chiamato makeHEX.bat. Il chipmodel viene rilevato automaticamente dall'IDE.

+ACo- Modificando questi file batch e possibile configurare il software del programmatore per le proprie esigenze. Tutti i programmatori si trovano nella sottocartella GCB+AEA-Syn.

+ACo- Se e presente un file batch +ACI-FlashThis.bat+ACI- nella stessa cartella del file di origine, questo file batch e utilizzato per il flashing. Questo puo essere utile per il flashing specifico del progetto.

Il software per i programmatori incluso ed e:

+ACo- Avrdude per AVR,

+ACo- PICPgm per PIC,

+ACo- PicKit2 e PicKit3,

+ACo- TinyBootLoader,

+ACo- Microchip Xpress Board e molti altri ancora.

(1) E necessario un programmatore adatto e le istruzioni dovrebbero essere incluse con il programmatore su come scaricare i programmi sul microcontrollore.

+ACo-Realease Formale del Compilatore Great Cow Basic v0.98.04+ACo-

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHw-Reference
+AHw-Orario Stampa +AHw-ASCIIDOCs rendered+AHw-2018-12-18 16:58:51 Ora Standard GMT +AHw-
Master ToC information+AHw-2018-11-19 08:38:50 GMT Standard Time
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

+ACo-Cambiamenti+ACo-

+ACo-Realease Formale di Great Cow Basic Compiler v0.98.04+ACo-

+ACo-Gli aggiornamenti HELP sono i seguenti:+ACo-

+ACo- Aggiunta nuova sezione NT7108C.

+ACo- Aggiornata la sezione G77D ST7735 per rimuovere i colori, aggiunti i colori TFT alla sezione panoramica di GLCD.

+ACo- Aggiunto supporto per il canale 8 PMW.

+ACo- Sezione PWM rivista per quanto riguarda i potenziali canali PWMN.

+ACo- Rivisto ST7735 GLCD per supportare OLED.

+ACo-Release 0.98.03+ACo-

+ACo- Aggiornata la sezione Randomize. +ACo- Aggiunto il supporto PCA9685 alla sezione. Libreria.
+ACo- Sezione Libreria aggiornata per includere l'elenco completo dei driver di libreria. +ACo-
Aggiunto HPWM+AF8-CCPTimerN. +ACo- Aggiunto GLCD ILI9326. +ACo- Aggiunta la sezione macOS di
Apple. +ACo- Aggiunta la sezione optionReserveHighProg. +ACo- Aggiunti test condizionali TRUE. +ACo-
Added LCD+AF8-IO+AF8-3. +ACo- Aggiunta nuova sezione HEFM. +ACo- Aggiunta nuova sezione SAFM.

+ACo-Release 0.98.02+ACo-

+ACo- Sezione chip aggiornata.

+ACo- Sezione del convertitore aggiornata per correggere lo stato deletetarget +AD0- y +AHw- n non
purge +AD0- y +AHw- n

+ACo- Aggiunto HSerPrintStringCRLF

+ACo- Aggiornata la sezione HPWMUpdate per chiarire l'utilizzo del tipo di variabile.

+ACo- Aggiornato GLCDCLS per mostrare parametri aggiuntivi per i dispositivi GLCD a colori.

+ACo- Aggiunto il metodo di scala.

+ACo- Aggiunta costante LCD+AF8-WIDTH.

+ACo- Variabili di impostazione aggiornate.

+ACo- Aggiunta sezione di Nextion.

+ACo- Informazioni sul formato, corrette sulla pagina della riga di comando.

+ACo-Release 0.98.01+ACo- +ACo- Aggiornata la sezione ILI9341 per i font OLED.

+ACo- Aggiornato I2C per il modulo I2C: nuove costanti e comandi.

+ACo- Aggiornato per il modello fisso PWM.

+ACo- Aggiornato per PWM a 10 bit con risoluzione e CCP / PWM con.

+ACo- DisableCCPFixedModePWM costante.

+ACo- Aggiunte pagine di contorno PWM a 16 bit.

+ACo- Revisionate le informazioni di ReadTable.

+ACo- Aggiunta sezione GLCD HX8347.

+ACo- Aggiunti nuovi comandi GLCDPrintString, GLCDPrintStringLN e GLCDLocateString.

+ACo-Release 0.98.00+ACo-

+ACo- +ACM-Define aggiornati esempi USART+AF8-TX+AF8-BLOCKING.

+ACo- Pagina delle condizioni, aggiornate con i vincoli noti dell'uso delle funzioni, come condizioni di prova

+ACo- Aggiornata la sezione SPI, aggiungendo il metodo FastHWSPIT Transfer.

+ACo- Sezione GLCD rivista per supportare SSD1306 128 +ACo- 32 pixel.

+ACo- Sezione ADC aggiornata per supportare tutte e tre le opzioni di lettura per ReadAD, ReadAD10 e ReadAD12.

+ACo- Sezione ingresso / uscita aggiornata per chiarire il supporto di AVR.

+ACo- Aggiunta sezione GLCD ILI9341.

+ACo- Aggiunta sezione GLCD ILI9486L.

+ACo- Sezione della tabella di ricerca aggiornata (Updated Lookup Table).

+ACo- Aggiornata la sezione OtherDirectives e aggiunte informazioni migliorate su ChipFamily.

+ACo- Aggiornata la sezione PWM per inserire il supporto per i canali PWM3 e PWM4.

- +ACo- Sezione PWM riveduta per migliorare le informazioni.
- +ACo- Aggiunta sezione HPWMUpdate.
- +ACo- Sezione subroutine aggiornata con utilizzo della memoria per i 16F.
- +ACo- Aggiunta la sezione 47xxx con una nuova sottosezione di memoria.
- +ACo- Aggiunto PWM per la sezione AVR.
- +ACo- Aggiornata la sezione altre direttive con le informazioni sui chip.
- +ACo- Aggiunta sezione timer SMT.
- +ACo- Sezione di manutenzione aggiornata.
- +ACo- Sezione timer aggiornata di nuovo.
- +ACo- Aggiornata la sezione PWM per includere le funzionalita di AVR
- +ACo- Aggiornata la sezione On Interrupt, rimuovendo errori di battitura e aggiungendo altri esempi.
- +ACo- Aggiornati i driver GCLD per aggiungere parametri per i colori.
- +ACo- Aggiunta sezione FVR.
- +ACo- STR aggiornato per includere STR32, STRInteger e un altro esempio.
- +ACo- VAL aggiornato per includere Val32 e un altro esempio.
- +ACo- Aggiornato la sezione GLCD per includere la configurazione di memoria insufficiente SSD1306 e la tabella delle funzionalita GLCD.
- +ACo- Aggiornato la sezione GLCD per includere la configurazione di memoria bassa SSD1306 e SH1106 e aggiunto altri esempi.
- +ACo- Aggiunto Ellipse e FilledEllipse.
- +ACo- Aggiunto Triangolo e Filled Triangolo.
- +ACo- Aggiunto il range di Select Case.
- +ACo- Vincolo di stringa concatenato aggiornato.
- +ACo- Informazioni aggiornate su LCD+AF8-Backlight per le modalita LCD 0, 4, 8 e 404.
- +ACo- Aggiunte informazioni OLED solo alla sezione SSD1306. I font OLED sono supportati solo dal GLCD SSD1306 in questa versione.
- +ACo- Aggiunto +ACM-SameVar e +ACM-SameBit.

+ACo- Aggiunto prefisso +AFs-canskip+AF0- a SettingVariables.

+ACo-Release 0.97.00+ACo-

+ACo- +ACM-Startup revisionata per documentare le ultime modifiche alla priorit  delle subroutine di avvio.

+ACo- Aggiunta sezione di manutenzione.

+ACo- Rivisto inittimer 1,3,5 e 7 per supportare ulteriori sorgenti di clock.

+ACo- Aggiornate tutte le sezioni PPS per supportare +ACM-startup come metodo consigliato.

+ACo- Revisionato Pulsein e aggiunto PulseInInv.

+ACo- Aggiunto PORTCHANGE alla sezione OnInterrupt.

+ACo- Informazioni aggiornate sul compilatore per la cache TRISIO.

+ACo- HSerGetNum aggiornato per supportare variabili Long e aggiunto un esempio.

+ACo- Revisionato LCD+AF8-IO 2+AF8-74xx164 per aggiungere il collegamento allo schema elettrico.

+ACo- Aggiunto l'utilizzo dell'output ad altre direttive.

+ACo- Aggiunto il supporto per SDD1306 per SPI alla sezione GLCD.

+ACo- Aggiunta sezione GetUserID.

+ACo- Aggiunta la sezione Serial Software (ottimizzata).

+ACo-Release 0.96.00+ACo-

+ACo- Uso delle stringhe rivisto ed esempi di stringhe per mostrare l'uso delle virgolette all'interno di una stringa.

+ACo- Revisionato OtherDirectives per includere le costanti mancanti.

+ACo- Aggiunto DisplaySegment e Revised DisplayChar.

+ACo- Aggiornato initimer2 / 4 e 6 per mostrare i prescalers.

+ACo- Ripristinati operazioni bit a bit e SetWith per help.

+ACo-Release 0.95.010+ACo-

+ACo- Aggiunti miglioramenti a SerSend e SerPrint.

+ACo- Sezione 7 segmenti riveduta per fornire chiarezza, migliorare il codice e rimuovere gli errori.

- +ACo- Sezione CCP e PWM riveduta per includere le funzionalita.
- +ACo- Revisione della sezione ADC e aggiunta costante ADReadPreReadCommand.
- +ACo-Release 0.95.009+ACo-
- +ACo- Aggiunto il driver GLCD SH1106.
- +ACo-Release 0.95.008+ACo-
- +ACo- Solo numero di versione aggiornato.
- +ACo-Release 0.95.007a+ACo-
- +ACo- Aggiunto +ACM-option explicit.
- +ACo- Aggiunta spiegazione variabile lifecycle.
- +ACo- Sezione PlayRTTTL rivista e migliorata.
- +ACo- Sezioni Wait e SPIMode riviste.
- +ACo-Release 0.95.007+ACo-
- +ACo- Aggiunte nuove funzionalita per supportare la gamma di microcontrollori 16F1885x.
- +ACo- Nuovi comandi PMW: supporto per hardware PWM.
- +ACo- Nuova opzione Initimer0 per supportare moduli timer a 8 e 16 bit.
- +ACo- Nuove costanti disponibili ChipADC e ChipIO.
- +ACo- Nuovi comandi per supportare PPS. LOCKPPS e UNLOCKPPS.
- +ACo- Aggiunto comando I2CStartoccurred. Mancava dalla Guida.
- +ACo- Ulteriori informazioni per gli script.
- +ACo- Rimozione generale degli errori di battitura nei file della Guida.
- +ACo- Aggiornato la sezione FnLSL rimuovendo l'errore.
- +ACo- La rimozione di SetWith, FnEQUBit e FnNOTBit dovrebbe ora essere sostituita dalle operazioni incorporato nel compilatore. I bit vengono ora gestiti correttamente dall'operatore Not incorporato.
- +ACo- Nuovo comando +ACM-option volatile. I pin sensibili possono essere contrassegnati come volatili e gli operatori di assegnazione nativi si imposteranno senza problemi, sostituendo SetWith.
- +ACo- Aggiunte nuove sezioni sull'ottimizzazione del codice ADC e PWM.

+ACo- Aggiunto il nuovo comando I2C per il software I2C use+AF8-I2C+AF8-timeout.

+ACo-Release 0.95.006+ACo-

+ACo- Riscrittura completata del file di aiuto.

+ACo- Nuovo file di aiuto in PDF.

+ACo- Nuovo file di guida in XML.

+ACo- Nuovo file di aiuto in HTML5.

+ACo- Nuovo file di aiuto in HTML per Web.

+ACo- Sezione Linux riveduta.

+ACo- Sezione SPI Mode rivista.

+ACo-Release 0.95.005a+ACo-

+ACo- Sezione hardware RS232 riveduta per migliorare la leggibilit .

+ACo-Release 0.95.005+ACo-

+ACo- Aggiunto LeftPad.

+ACo-Release 0.95.004+ACo-

+ACo- Aggiornati i comandi HSerial per supportare AVR 1, 2, 3 e 4.

+ACo-Release 0.95+ACo-

+ACo- Aggiunto set di comandi pullup debole.

+ACo- Aggiunto utilizzo della RAM durante la definizione della matrice.

+ACo- Aggiunto un nuovo metodo per usare una costante per definire una matrice.

+ACo- Aggiunte informazioni su come impostare l'indirizzo sui moduli mjkdz I2C LCD 1602.

+ACo- Aggiunte nuove costanti ChipWords e ChipEEPROM.

+ACo- Aggiunto un nuovo metodo di definizione della tabella.

+ACo- Aggiunte nuove funzionalit  alle tabelle di ricerca.

+ACo- Aggiunte nuove funzionalit  a READAD per i microcomputer AVR.

+ACo- Aggiunte istruzioni per compilare Great Cow BASIC sotto Linux.

- +ACo- Aggiunto un nuovo comando per invertire un display GLCD KS0108.
- +ACo- Aggiunto nuovo codice di esempio: FLASH+AF8-LED.
- +ACo- Aggiunto come impostare la velocita del chip a velocita non standard.
- +ACo- Aggiunto un nuovo parametro di comando al set di comandi hardware USART.
- +ACo- Il set di comandi HI2C2.
- +ACo- Aggiunto un nuovo set di comandi per la seconda porta USART.
- +ACo- Aggiunto il nuovo set di comandi del driver GLCD ILI9340.
- +ACo- Aggiunto il set di comandi del driver GLCD SDD1289.
- +ACo- Aggiunto codice di esempio. Misurazione della durata dell'impulso alla risoluzione del sub-microsecondo.
- +ACo- Generazione di impulsi accurati mediante una macro. Come passare un indirizzo di porta alla routine usando una macro.
- +ACo- Aggiunta sezione di supporto .NET.
- +ACo- Parametri del comando SPI Mode rivisti.
- +ACo- Aggiunto +ACM-option NoContextSave.
- +ACo- Aggiunto su Interrupt: il gestore predefinito.
- +ACo- Aggiunte nuove opzioni del compilatore sotto-sezione, spostate le opzioni dalle Direttive del compilatore.
- +ACo- Aggiunto il nuovo supporto per i caratteri per GLCD.
- +ACo- Aggiunto il nuovo supporto per i colori per GLCD.
- +ACo- Aggiunto LCD+AF8-IO+AF8-1 e cambia LCD+AF8-IO+AF8-2 per supportare 74xx164 e 74xx174.
- +ACo- Revisionata la maggior parte della sezione Timer per supportare le informazioni corrette.
- +ACo- Aggiunto il supporto del controller touchscreen ADS7843.
- +ACo- Aggiunto il set di comandi Play e PlayRTTL.
- Aggiunto comando DIFFerence.
- +ACo- Aggiunti tre metodi bit a bit. FnNotBit, FnLSL e FnLSR.
- +ACo- Domande frequenti aggiornate con +ACI-come impostare una porta.

+ACo- Aggiunto il metodo bitwise FnequBit.

+ACo- Aggiunti i dettagli di Timer6, Timer7, Timer8, Timer10 e Timer12 su On Interrupt.

+ACo- Aggiunto un nuovo esempio di on interrupt.

+ACo- Aggiunta la riga di comando / WX.

+ACo- Aggiunti HI2CWaitMSSP e HI2CWaitMSSPTimeout alla sezione I2C hardware.

+ACo- Parametri della riga di comando aggiornati.

+ACo- Aggiunto il comando Fill.

+ACo- Sezione Array rivisto.

+ACo- Aggiunto il comando ReadAD12.

+ACo- Aggiunta la spiegazione di dividere e divisione.

+ACo- Aggiunto Str32 e Val32.

+ACo- Aggiunto supporto per USART2. HSerReceive1 e HserReceive2.

+ACo- Aggiunto HserRecieveFrom.

+ACo- Aggiunto HserGetNum.HSertGetString.

+ACo- Aggiunta modalita di misurazione a canale singolo e modalita di misurazione del canale differenziale a ADC.

+ACo- Panoramica.

+ACo-Release 0.94b+ACo-

+ACo- Aggiunto il supporto HEFM.

+ACo- Aggiunto il supporto del driver GLCD SSD 1306.

+ACo-Release v0.91+ACo-

+ACo- Aggiunto USART+AF8-TX+AF8-BLOCKING.

+ACo- Aggiunto LCD+AF8-SPEED.

+ACo- Sezione LCD migliorata.

+ACo-Release v0.9ho+ACo-

+ACo- Aggiornamento del parametro passato alle routine secondarie.

+ACo- Sezione LCD nuova e rivista per includere LCD+AF8-IO 10 e 12.

+ACo- Rimuovere la LAT dove appropriato.

+ACo- LAT è stato deprecato. Il compilatore reindirizzerà tutte le scritture pin I / O da PORTx a LATx registri su PIC 16F1 / 18F.

+ACo- Utilizzare +ACM-option nolatch se si verificano problemi.

+ACo- ADFormat modificato in deprecato.

+ACo- Aggiungi l'azione predefinita a +ACM-CHIP quando non viene specificata alcuna frequenza.

+ACo-Release v0.9h+ACo-

+ACo- Corretti gli errori nella sezione PWM e esempi migliorati.

+ACo-Release v0.9hn+ACo-

+ACo- Modifiche agli array. Il numero di elementi è ora limitato a 10.000 per dispositivi 12F e 16F o, la RAM disponibile.

+ACo- Tabelle di ricerca aggiornate per supportare i nuovi metodi di compilazione delle tabelle.

1. un singolo valore su ciascuna linea.
2. più elementi su una singola riga separati da virgole.
3. Sono consentite costanti e calcoli all'interno delle voci della tabella di dati a riga singola.

+ACo- Repeat loop modificato per supportare EXIT REPEAT.

+ACo- Nuovo comando Pad. Il metodo Pad viene utilizzato per creare una stringa su una lunghezza specifica che viene estesa con un carattere specifico.

+ACo- Aggiunto il set di comandi DS18B20.

+ACo-v0.9hm+ACo-

+ACo- Aggiornato I2C - software e hardware. Il codice dimostrativo ora utilizza la scheda dimostrativa Chipino. Modificato nella sezione I2C seriale con questi nuovi esempi.

+ACo- Nuovi comandi funzionali:

LCDDisplayOn

LCDDisplayOff

LCDBackLight (On +AHw- Off)

+ACo- I nuovi Define per supportare le funzionalità LCD sono:

-LCD+AF8-SPEED FAST

-LCD+AF8-SPEED MEDIUM

-LCD+AF8-SPEED SLOW

+ACo- Funzionalita Revisionata LCDHex supporta ora la stampa degli zeri iniziali quando il numero HEX e inferiore a 0x10. chiamare LCDHex come segue per assicurarsi che siano presenti zeri iniziali.

+ACo- LCDHex byte+AF8-value, LeadingZeroActive parameter called LeadingZeroActive.

+ACo- Nuovo supporto per i dispositivi GLCD PCD8544.

+ACo- Sezione GLCD modificata dell'aiuto per supportare il nuovo dispositivo.

+ACo-v0.9hl+ACo-

+ACo- HSERPRINTCLRF - Aggiunto parametro per ripetere il numero di CRLF inviato.

+ACo- Aggiunto set di comandi I2C hardware. Questa e una funzionalita riveduta per fornire supporto al modulo MSSP.

+ACo-0.9hk+ACo-

+ACo- File della guida aggiornato per correggere l'immissione di potenza, era nella sezione errata. Spostato nella sezione Matematica e altri errori minori.

+ACo- Correggere le informazioni del timer0. Revisionato per mostrare le costanti e il codice del timer e stato corretto.

+ACo-v0.9hj+ACo-

+ACo- Queste informazioni si riferiscono a Hot Release del 11 maggio.

+ACo-2014.+ACo- Dove la funzionalita non e supportata dalle versioni precedenti di GCB, effettuare l'aggiornamento. Alcune funzioni non funzioneranno nelle versioni precedenti di Great Cow Basic.

+ACo- Nuovi comandi funzionali.

+ADwAPA-a-1,Circle+AD4APg- Disegna un cerchio sullo schermo del GLCD.+ADwAPA-a-2,FilledCircle+AD4APg- Riempie un cerchio sullo schermo del GLCD.+ADwAPA-a-3,Log+AD4APg-function(s) Power funzione. registro funzione(i) Comandi funzionali modificati.

+ADwAPA-a-4,Line+AD4APg- Ora disegna linee tra due punti qualsiasi sul display GLCD. +ACM-define GLCD+AF8-PROTECTOVERRUN . Controlla il disegno di cerchi per impedire il superamento del cerchio a estremi di isualizzazione.

+ACM-define Line OldLine. Aggiungendo questa definizione si tornera alle vecchie routine di disegno a linee. Questo e stato aggiunto per compatibilita con le versioni precedenti.

Aiuto e revisioni dei file

Aggiunto +ADwAPA-a-5,PulseIn+AD4APg-

Aggiunto PulseInAggiunto PulseIn

Aggiunto esempio di intestazione +ADwAPA-a-6,IR+AF8-Remote+AD4APg-

Aggiunto esempio di dimostrazione e rivisto +ADwAPA-a-7,GLCD demonstration+AD4APg-

Aggiunto esempio di controllo +ADwAPA-a-8,RGB LED Control+AD4APg- Aggiunta sezione per mostrare il metodo di documentazione inline,vedere Code Documentation

+ACo-v0.9hk+ACo-

+ACo- Metodo documentato per la documentazione GCGB.

Aggiunto MATHS.H.

Aggiunta la funzione SQRT.

+ACo-v0.9hi+ACo-

+ACo- Supporto per ST7735 documentato. Funzionalita aggiunta a GLCD.h.

+ACo- Supporto per dispositivo grafico ST7920 LCD 128 +ACo- 64.

+ACo- Sezione GLCD modificata per includere il dispositivo nuovo e uno non documentato.

+ACo- Nuovi comandi GLCD per il supporto di GLCD ST7920

ST7920GLCDEnableGraphics

ST7920GLCDDisableGraphics

ST7920GLCDClearGraphics

ST7920Locate

ST7920gTile

ST7920Tile

ST7920cTile

ST7920SetIcon

ST7920GraphicTest

ST7920LineHs

ST7920gLocate

ST7920linev

+ACo- Supporto documentato per GLCD ST7735.

+ACo- Revisionati i comandi GLCD con retro compatibilit :

GLCDCLS

GLCDPrint - supporta le modalit  LCD e GLCD

GLCDDrawString - supporto per la gestione delle stringhe

GLCDDrawChar - Colore opzionale

Box

FilledBox

Line

PSet

GLCDReady

+ACo- InitGLCD, include la correzione per la routine di avvio per i
dispositivi KS0108

Le funzioni Private ST7920 possono essere utilizzate secondo necessita ..

ST7920WriteCommand

ST7920WriteData

ST7920WriteByte

ST7920gReaddata

ST7920GLCDReadByte

GLCDTimeDelay

+ACo- Informazioni test condizionali aggiornate.

+ACo- Informazioni aggiornate sul KeyPad.

+ACo- Informazioni aggiornate sulla tabella di ricerca.

+ACo- Aggiunte informazioni sulla macro.

+ACo- Aggiunta nuova sezione matematica Trig.

+ACo- Aggiunti due nuovi esempi di Circle

+ACo- Aggiunte altre informazioni sulle direttive.

+ACo- Aggiunti programmi di esempio.

+ACo- Cerchi punto medio.

+ACo- Cerchi per trigonometria.

+ACo-v0.9hg+ACo-

+ACo- Pagine di visualizzazione GLCD ad anodi comuni corrette.

+ACo-v0.9hf+ACo-

+ACo- Sezione 7 segmenti riveduta per supportare il catodo comune. Ci sono ora due opzioni disponibili.

+ACo-v0.9he+ACo-

+ACo- Nuovi comandi. Richiesto dopo marzo 2014 LCD.h:
LCDHOME, LCDSPACE, LCDCreateGraph, LCDCursor, LCDCmd

+ACo- Aggiunta della concatenazione.

+ACo- Aggiornato DisplayValue per mostrare il supporto per i valori HEX.
Richiesto dopo marzo 2014 7Segment.h.

+ACo- Codice di esempio GLCD aggiornato per garantire che l'esempio
sia compilato senza file esterni.

+ACo- Aggiunti Trigonometria e l'applicazione di esempio

+ACo- Aggiornamento della panoramica LCD per includere il supporto LATx per una maggiore velocita di clock. Post richiesto Marzo 2014 LCD.h

+ACo-v0.9hd+ACo-

+ACo- Revisionato Ruota per chiarire i tipi di byte supportati.

+ACo-v0.9hc Mar 2014+ACo-

+ACo- Revisione di HSERPRINT per mostrare i numeri interi e i caratteri

lunghi sono supportati e il testo modificato e corretto.

+ACo- Aggiunti HserPrintByteCRLF e HserPrintCRLF

+ACo- Aggiunto esempio alla tabella sine

+ACo- TABELLA Revisionata per mostrare la limitazione rispetto all'utilizzo

di WORDS quando si posizionano TABLES in EEPROM

+ACo-v0.9hb Mar 2014+ACo-

+ACo- Aggiunto PulseOutInv

+ACo- I2CRestart

+ACo- Aggiungi nuove varianti per l'uso dei commenti

+ACo- Aggiunta sezione Assembler

+ACo-14 Gennaio+ACo-

+ACo- Nuovi comandi):

Len, Asc, Chr, Trim, Ltrim, Rtrim, Swap4, Swap, Abs, Average, Trim, Ltrim, Rtrim, Wordtobin, Bytetobin, GLCD, DectoBCD, BCDtoDec.

Uso delle variabili.

Altro su costanti e variabili.

Ringraziamenti.

+ACo- Cambia in:

Str, Hex, Poke, Else, Readtable, Exit (era exitsub)

Parametri della riga di comando Domande frequenti

+ACo- Corretti errori di battitura.

+ACo- Valore di ripetizione massimo REPEAT aggiornato.

+ACo- Aggiornamento della maggior parte delle pagine per il layout.

Risolti i collegamenti alle pagine esterne. Questa volta scaricato come pagine html complete, per POT e LC.

Aggiunto LABEL, Bootloader e revisione Selezione, aggiunto READAD10

Corretto Double SWAP

+ACo- v0.9hg+ACo-

+ACo- Corrected GLCD Common Anode display pages

+ACo- v0.9hf+ACo-

+ACo- Revised 7 Segment section to support Common Cathode. Split 7 Segment entry to show the two options available.

+ACo-v0.9he+ACo-

+ACo- Nuovi comandi. Richiesto dopo marzo 2014 LCD.h:

LCDHOME, LCDSPACE, LCDCreateGraph, LCDCursor, LCDCmd

+ACo- Aggiunta della concatenazione

+ACo- Aggiornato DisplayValue per mostrare il supporto per i valori HEX.

Obbligatorio post marzo 2014 7Segment.h.

+ACo- Codice di esempio GLCD aggiornato per garantire che l'esempio sia compilato senza file esterni.

+ACo- Aggiunto Trigonometria e l'applicazione di esempio.

+ACo- Aggiornamento della panoramica LCD per includere il supporto

LATx per una maggiore velocità di clock. Post richiesto Marzo 2014

LCD.h.

+ACo-v0.9hd+ACo-

+ACo- Revisionato Rotate per chiarire i tipi di byte supportati.

+ACo-v0.9hc Mar 2014+ACo-

+ACo- HSERPRINT rivisto per mostrare i numeri, integer e long sono supportati e il testo modificato e corretto.

+ACo- v0.9hb Mar 2014+ACo-

+ACo- Added PulseOutInv +ACo- I2CRestart +ACo- Add new variants to use of Comments +ACo- Added Assembler Section

+ACo-14 Gennaio+ACo-

+ACo- Nuovi comandi):

Len, sc, Chr, Trim, Ltrim, Rtrim, Swap4, Swap, Abs, Average, Trim,
Ltrim, Rtrim, Wordtobin, Bytetobin, GLCD, DectoBCD, BCDtoDec

Uso delle variabili

Altro su costanti e variabili.

Ringraziamenti.

+ACo- Cambiati in:

Str, Hex, Poke, Else, Readtable, Exit (era exitsub)

Parametri della riga di comando Domande frequenti.

+ACo- Corretti errori di battitura.

+ACo- Valore di ripetizione massimo REPEAT aggiornato.

+ACo- Aggiornamento della maggior parte delle pagine per il layout.

Risolti i collegamenti alle pagine esterne. Questa volta scaricato
come pagine html complete, per POT e LC.

Aggiunto LABEL, Bootloader e revisione Selezione, aggiunto

READAD10.

Corretto Double SWAP.

+ACo- v0.9h+ACo-

+ACo- Pagine di visualizzazione degli anodi comuni GLCD corrette.

+ACo-v0.9h+ACo-

+ACo- Sezione 7 segmenti riveduta per supportare il catodo comune.

Dividi la voce di 7 segmenti per mostrare le due
opzioni disponibili.

+ACo-v0.9he+ACo-

+ACo- Nuovi comandi. Richiesto dopo marzo 2014 LCD.h:

LCDHOME, LCDSPACE, LCDCreateGraph, LCDCursor, LCDCmd.

+ACo- Aggiunta della concatenazione.

+ACo- Aggiornato DisplayValue per mostrare il supporto per i valori HEX.

Richiesto dopo marzo 2014.

7Segment.h

+ACo- Codice di esempio GLCD aggiornato per garantire che l'esempio.

sia compilato senza file esterni.

+ACo- Aggiunti Trigonometria e l'applicazione di esempio.

+ACo- Aggiornamento della panoramica LCD per includere il supporto.

LATx per una maggiore velocità di clock. Post richiesto.

Marzo 2014 LCD.h

+ACo-v0.9hd+ACo-

+ACo- Revisionato Rotate per chiarire i tipi di byte supportati.

+ACo-v0.9hc Mar 2014+ACo-

+ACo- HSERPRINT rivisto per mostrare i numeri interi e i caratteri lunghi sono supportati e ha cambiato il testo corretto.

+ACo- Aggiunti HserPrintByteCRLF e HserPrintCRLF.

+ACo- Aggiunto esempio di tabella sinusoidale.

+ACo- TABELLA Revisionata per mostrare la limitazione rispetto all'utilizzo di WORDS quando si inseriscono TABELLE in EEPROM.

+ACo-v0.9hb Mar 2014+ACo-

+ACo- Aggiunto PulseOutInv.

+ACo- I2CRestart.

+ACo- Aggiungi nuove varianti per l'uso dei commenti.

+ACo- Aggiunta sezione Assembler 14 gennaio.

Nuovi comandi:

Len, sc, Chr, Trim, Ltrim, Rtrim, Swap4, Swap, Abs, Average, Trim,

Ltrim, Rtrim, Wordtobin, Bytetobin,

GLCD, DectoBCD, BCDtoDec

Usare le variabili

Altro su costanti e variabili

Ringraziamenti

+ACo- Cambiano in:

Str, Hex, Poke, Else, Readtable, Exit (era exitsub)

Parametri della riga di comando Domande frequenti

+ACo- Corretti errori di battitura.

+ACo- Valore di ripetizione massimo REPEAT aggiornato.

+ACo- Aggiornamento della maggior parte delle pagine per il layout.

Risolti i collegamenti alle pagine esterne. Questa volta scaricato

come pagine html complete, per POT e LC.

Aggiunto LABEL, Bootloader e revisione Select, aggiunto

READAD10

Corretto Double SWAP

+ACo-parametri della riga di comando+ACo-

+ACo-Informazioni sui parametri della riga di comando+ACo-

GCBASIC +AFs-/O:output.asm+AF0- +AFs-/A:assembler+AF0- +AFs-/P:programmer+AF0- +AFs-/K:+AFwAew-C+AHw-A+AH0AXQ- +AFs-/V+AF0- +AFs-/L+AF0- +AFs-/NP+AF0- nomefile

GCBASIC +AFs-/O:output.asm+AF0- +AFs-/A:assembler+AF0- +AFs-/P:programmer+AF0- +AFs-/K:+AHs-C+AHw-A+AH0AXQ- +AFs-/V+AF0- +AFs-/L+AF0- +AFs-/WX+AF0- +AFs-/NP+AF0- nomefile

GCBASIC +AFs-/O:output.asm+AF0- +AFs-/A:assembler+AF0- +AFs-/P:programmer+AF0- +AFs-/K:+AHs-C+AHw-A+AH0AXQ- +AFs-/V+AF0- +AFs-/L+AF0- +AFs-/WX+AF0- +AFs-/S:Use.ini+AF0- +AFs-/NP+AF0- nomefile

GCBASIC +AFs-/O:output.asm+AF0- +AFs-/A:assembler+AF0- +AFs-/P:programmer+AF0- +AFs-/K:+AHs-C+AHw-A+AH0AXQ- +AFs-/V+AF0- +AFs-/L+AF0- +AFs-/WX+AF0- +AFs-/S:Use.ini+AF0- +AFs-/F+AFs-

O+AF0AXQ- +AFs-/NP+AF0- nomefile

GCBASIC /version

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHwAKg-
switch+ACo- +AHwAKg-descrizione+ACo- +AHwAKg-default+ACo- +AHw-/:filename +AHw-Imposta il
nome del file assembly generato nel nome file. +AHw-Stesso nome del file di input, ma con estensione
.asm.

+AHw-/A:assmber +AHw-File batch utilizzato per chiamare l'assemblatore (1). Se / A: GCASM e dato,
Great Cow BASIC usera il suo assemblatore interno. +AHw-Il programma non verra assemblato

+AHw-/P:programmer +AHw-File batch utilizzato per chiamare il programmatore (1). Questo
parametro viene ignorato se il programma non e assemblato. +AHw-File batch utilizzato per chiamare
l'assemblatore (1). Se / A: GCASM e dato, Great Cow BASIC usera il suo assemblatore interno.

+AHw-/K:+AFs-C A+AF0AfA-Conserva il codice originale nell'output dell'assieme. +AHw-Nessun codice
originale lasciato in uscita.

+AHw-/V+AFs-:+AFs-O F+AF0AWw-1 T+AF0- +AHw-Modalita dettagliata: il compilatore fornisce
informazioni piu dettagliate sulle sue attivita. / Vx sovrascrivera qualsiasi configurazione nel file
utente ini +AHw--

+AHw-/L +AHw-Mostra la licenza e esci. +AHw--

+AHw-/NP +AHw-Non mettere in pausa sugli errori. Utilizzare con IDE. +AHw-Pausa quando si
verifica un errore e attendere che l'utente preme un tasto.

+AHw-/WX +AHw-Forza il compilatore per garantire che tutti i file inclusi siano validi. +AHw- +AHw-
/version +AHw-Data di costruzione mostrata e versione del compilatore. +AHw-

+AHw-/S +AHw-Carica le impostazioni da un file specificato, piuttosto che utilizzare i valori
predefiniti. +AHw-/S:use.ini

+AHw-/F+AFs-:+AFs-0 F+AF0AWw-1T+AF0- +AHw-Utilizzato per ignorare la compilazione quando non
e necessario, il compilatore verifichera che le impostazioni di configurazione nel file gia compilato
corrispondano a quelle richieste per il programmatore. In caso contrario, una ricompilazione sara
forzata. Salta la compilazione se il file hex e aggiornato se ha la configurazione corretta. / F: x (F o 0)
per forzare una nuova compilazione indipendentemente da cio che ini specifica. +AHw-

+AHw-/FO +AHw-Utilizzato per ignorare solo la compilazione e il programma. Il compilatore
verifichera che le impostazioni di configurazione nel file gia compilato corrispondano a quelle
richieste per il programmatore. In caso contrario, una ricompilazione sara forzata. +AHw- +AHw-
nome del file +AHw-Il file da compilare.
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

(1) Per i / A: e / P: switches, ci sono opzioni speciali disponibili. Se +ACU-FILENAME+ACU- e presente,

sara sostituito dal nome del file .asm.

+ACU-FN+AF8-NOEXT+ACU- verra sostituito dal nome del file .asm ma senza un'estensione e +ACU-CHIPMODEL+ACU- verra sostituito con il nome del chip. Il nome del chip sara uguale a quello sul file di dati del chip. Un file batch per caricare l'ASM da Great Cow BASIC in MPASM. La riga di comando dovrebbe essere cosi:

```
C:+AFw-progra+AH4-1+AFw-microc+AH4-1+AFw-mpasms+AH4-1+AFw-MPASMWIN /c- /o- /q+- /l- /x-  
/w1 +ACU-code+ACU-.asm
```

Un file batch da compilare in Great Cow BASIC quindi caricare l'ASM da Great Cow BASIC in GPASM.

La riga di comando dovrebbe essere cosi:

```
gcbasic.exe +ACU-1 /NP /K:A /A:+ACI-..+AFw-gputils+AFw-bin+AFw-gpasm.exe +ACUafg-  
d1+ACUafg-p1+ACUafg-n1.asm+ACI-
```

Per istruire MAKEHEX.BAT per utilizzare GPASM, hai installato

GPUTILS. Il file batch deve essere modificato come segue:

```
REM Create the ASM  
gcbasic.exe /NP /K:A +ACU-1  
REM Use GPASM piping to the GCB error log  
gpasm.exe +ACIAJQB+-d1+ACUafg-p1+ACUafg-n1.asm+ACI- -k -i -w1 +AD4APg- errors.txt
```

Per riassumere, puoi utilizzare uno dei seguenti:

```
gcbasic.exe filetocompile.gcb /A:GCASM /P:+ACI-icprog -L+ACU-FILENAME+ACUAIg- /V  
/O:compiled.asm
```

Great Cow BASIC compilara il file, quindi assemblera il programma

ed eseguirà questo comando:

```
+ACI-icprog -Lcompiled.hex+ACI-
```

Puoi anche creare / modificare il file gcbasic.ini:

```
'Assembler settings
Assembler +AD0- C:+AFw-Program Files+AFw-Microchip+AFw-MPASM Suite+AFw-mpasmwin
AssemblerParams +AD0- /c- /o- /q+- /l+- /x- /w1 +ACIAJQ-FileName+ACUAIg-
'Programmer settings
Programmer +AD0- C:+AFw-Program Files+AFw-WinPic+AFw-Winpic.exe
ProgrammerParams +AD0- /device+AD0-PIC+ACU-ChipModel+ACU- /p +ACIAJQ-FileName+ACUAIg-
```

Questo esempio utilizzerà MPASM per assemblare il programma.

Avvia il programma specificato in assembler +AD0- line, e dargli questi

parametri:

```
+AGA-/c- /o- /q+- /l+- /x- /w1 +ACI-compiled.asm+ACIAYA-
```

Quindi, chiamerà il programmatore e assegnerà questi parametri:

```
+AGA-/device+AD0-PIC16F88 /p +ACI-compiled.hex+ACIAYA-
```

+ACU-ChipModel+ACU- verrà rimpiazzato dal chip che stai usando, quindi questo chip Great Cow BASIC lo passerà a WinPIC.

+ACo-Errorr.txt+ACo-

Il compilatore produce solo il file errors.txt se c'è un errore. La creazione del file errors.txt rende più facile per gli IDE rilevare se il programma è stato compilato correttamente, se il file non è stato prodotto quindi l'IDE non sarà in grado di presentare il messaggio di errore all'utente. Il file error.txt viene sempre prodotto nella stessa cartella del compilatore. Tipicamente: C: +AFw- GCB +AEA- Syn +AFw- GreatCowBasic +AFw- Errors.txt

+AD0APQ- +ACo-Domande frequenti+ACo- +ACo-Perche non succede nulla quando eseguo GCBASIC.exe?+ACo- Great Cow BASIC è un compilatore a linea di comando. Per compilare un file, puoi trascinarlo e rilasciarlo sul Icona GCBASIC.exe. Sono disponibili anche diversi ambienti di sviluppo integrati, o IDE Great Cow BASIC. Questi ti daranno un'area in cui puoi modificare il tuo programma e un pulsante da inviare il programma per il chip. Diversi sono elencati sul sito web di Great Cow BASIC.

+ACo-Quali microcontrollori Microchip PIC e Atmel AVR fanno il supporto di Great Cow BASIC?+ACo- Si spera che tutti i microcontrollori a 8 bit Microchip PIC e Atmel AVR (quelli in PIC10, PIC12, PIC16 e Famiglie PIC18). Se trovi quello che Great Cow BASIC non funziona correttamente, per favore pubblicalo nella sezione Problemi del compilatore del forum di Great Cow BASIC.

+ACo-Case sensitive, Great Cow BASIC +JRwAvw- sensibile ai caratteri maiuscoli o minuscoli?+ACo- No+ACE- Ad esempio, Set, SET, set, SeT, ecc. Sono tutti trattati allo stesso modo da Great Cow BASIC.

+ACo-Posso specificare il bit di una variabile e modificarli usando un'altra variabile?+ACo-
L'impostazione di una variabile con altre variabili e prevista. Great Cow BASIC supporta assegnazioni bit a bit. Come segue:

```
portc.0 +AD0- +ACE-porta.1
```

Puoi anche usare una funzione di spostamento. Come in altri linguaggi, usando la funzione Shift
FnLSL. UN esempio +JRwAvw-:

```
MyVar +AD0- FnLSL( 1, BitNum) e Equivalente a +ACI-MyVar +AD0- 1+ADwAPA-BitNum+ACI-
```

Per impostare un bit di una porta, per evitare glitches durante l'aggiornamento dell'output, utilizzare questo metodo.

```
'aggiungere la seguente opzione per la porta specifica.  
+ACM-option volatile portc.0 ' poi nel tuo codice  
portc.0 +AD0- +ACE-porta.1
```

e possibile utilizzare questo metodo per impostare un bit di una porta. Inserito nel metodo SetWith,
questo elimina anche qualsiasi problema durante l'aggiornamento, utilizzare questo metodo.

```
SetWith(MyPORT, MyPORT OR FnLSL( 1, BitNum))
```

Per cancellare un bit di una porta, utilizzare questo metodo.

```
MyPORT +AD0- MyPORT AND NOT FnLSL( 1, BitNum))
```

Per impostare un bit all'interno di un array, utilizzare questo metodo.

```
video+AF8-buffer+AF8-A1(video+AF8-adress) +AD0- video+AF8-buffer+AF8-A1(video+AF8-adress)  
OR FnLSL( 1, BitNum)
```

Vedere anche +ADwAPA-a-9,Set+AD4APg-, +ADwAPA-a-10,FnLSL+AD4APg-, +ADwAPA-a-
11,FnLSR+AD4APg- e +ADwAPA-a-12,Rotate+AD4APg-

+ACo-TWhy e x la funzione non e implementata?+ACo-

Perche non e stato pensato, o nessuno e stato in grado di implementarlo+ACE- Se ci sono delle
caratteristiche che ti piacerebbe vedere in Great Cow BASIC, per favore pubblicali nella sezione +ACI-
Open Discussion+ACI- del Forum di Great Cow BASIC. Oppure, se puoi, prova ad aggiungere la funzione

da solo+ACE-

+ACo-Quando si utilizza un file include questo usa molta memoria?+ACo- Quando si utilizzano i file include, in questo caso include +ADw-ds3231.h+AD4-, se non si utilizzano tutte le funzioni del file include, GCB sa di non includere le funzioni non utilizzate all'interno del file include durante la compilazione, o include tutto comunque. Ad esempio, se non utilizzo l'hardware I2C, tutto il codice relativo all'hardware I2C viene ancora compilato nel codice?

Great Cow BASIC compila solo funzioni e subroutine se vengono chiamate. Great Cow BASIC inizia a compilare dalla routine principale, quindi qualsiasi cosa chiamata da lì. Ogni volta che trova una nuova subroutine chiamata, la compila e tutto ciò che chiama. Se una subroutine non è necessaria, non viene compilata.

+ACo-Il mio LCD non funzioner+JRwA4Q- come previsto?+ACo-

Aggiungi. +ACM-define LCD+AF8-SPEED SLOW

Questo rallenter+JRwA4Q- la scrittura sul display LCD.

+ACo-L'utilizzo della memoria di Atmel AVR è errata?+ACo-

I valori di memoria di Atmel AVR sono specificati in WORD in Great Cow BASIC. Il compilatore di Great Cow BASIC utilizza le Word, non i byte, per la coerenza tra microcontrollori Microchip PIC e Atmel AVR. Questo mantiene le parti del compilatore più semplici.

+ACo-Non riesco ad aprire il file di aiuto della finestra?+ACo-

Vedere <http://url+AFs-digital.ni.com/public.nsf/allkb/B59D2B24D624B823862575FC0056F3D0+AF0->.

+AD0APQ- +ACo-Risoluzione dei problemi+ACo-

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHw-Problemi
+AHw-Cause Comuni +AHw-Assistenza +AHw-Impossibile compilare il programma +AHw-C'è un errore nel programma. La Great Cow BASIC si lamenta di una particolare linea di codice? +AHw-Forum Great Cow BASIC

+AHw-

+AHw-

C'è un bug in Great Cow BASIC +AHw-Pubblica sul forum Great Cow BASIC. Assicurati di indicare la versione del compilatore e allegare il codice come un ZIP.

+AHw-Un programma viene compilato e scaricato correttamente, ma non verrà eseguito

+AHw-Oscillatore non dichiarato +AHwAPAA8-a-13,Configuration +AD4APg-

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

+AD0APQ- +ACo-Statistiche del Compilatore+ACo-

Questa sezione fornirà alcune informazioni su ciò che fa il compilatore.

Come si comporta il compilatore con i registri di sola lettura nella gamma famiglia PIC12?

All'interno di questo intervallo di chip, il registro delle opzioni e un registro di sola scrittura. La lettura del registro non è consentita.

Great Cow BASIC ha bisogno di aggiornare questo quando l'utente vuole cambiare la configurazione del processo Sleep e un esempio di una modifica dell'utente.

Il compilatore gestisce questo con la creazione della variabile di byte Option-REG. Questo byte è creato dal compilatore per gestire il processo di scrittura richiesto.

La variabile Option-REG è una cache, che il compilatore creerà se sono stati impostati bit di option-REG manualmente.

Se l'utente cambia uno dei bit in un programma, il compilatore troverà qualsiasi utilizzo dell'opzione istruzioni e li inserisce in `OPTION-REG` immediatamente prima dell'istruzione dell'opzione per memorizzare nella cache il valore nel buffer.

Se i bit Option-REG non sono impostati individualmente, quindi option-REG non viene creato, non viene fatto nulla di speciale con l'istruzione opzionale.

Essenzialmente il compilatore mantiene una variabile speciale e gestisce l'intero processo senza che l'utente sia consapevole.

Come fa il compilatore a far fronte al registro TRIS nei PIC10f?

Il compilatore assicura che una cache TRIS corrisponda al registro TRIS attuale. La cache TRIS è una variabile byte chiamata TRISIO. La cache TRISIO è necessaria poiché TRIS è un registro di sola scrittura.

Tutte le porte predefinite per l'input (dove tutti i bit TRIS sono a 1) al reset. Pertanto, si presume che sia il valore 255.

TRISIO viene aggiornato quando richiesto dal codice utente e quindi utilizzato nella scrittura nel registro corretto.

Il codice utente di esempio e l'insieme associato con l'assembly mostra la cache TRISIO in uso. Questo metodo è conforme con la scheda tecnica.

Codice utente

```
'set as input dir gpio.0 in gpio0State +AD0- gpio.0 'impostato come output, sar+JRwA4Q-  
necessario impostare TRIS GPIO utilizzando la cache TRISIO.
```

```
dir gpio.0 out
gpio.0 +AD0- 1
```

ASM

```
+ADs-dir gpio.0 in
  bsf TRISIO,0
  movf TRISIO,W
  tris GPIO
+ADs-gpio0State +AD0- gpio.0
  clrf GPIO0STATE
  btfsc GPIO,0
  incf GPIO0STATE,F
+ADs-dir gpio.0 out
  bcf TRISIO,0
  movf TRISIO,W
  tris GPIO
+ADs-gpio.0 +AD0- 1
  bsf GPIO,0
```

Ovunque sia impostato / azzerato un singolo bit TRIS cambiando la direzione della porta, il bit nella cache è cambiato e poi viene scritto nel registro TRIS. Forcing the ASM to contain comments It may be useful to force comments into the ASM file. The verbose mode of creating the ASM will include ALL the source program as comments but it may be useful to have specific comments in the ASM to aid the understanding of code or to support debugging.

+ACo-Forzare l'assembly per contenere commenti:+ACo-

Potrebbe essere utile forzare i commenti nel file ASM. La modalità di creazione dell'ASM lo farà +AOE- include TUTTO il programma sorgente come commenti, ma potrebbe essere utile avere commenti specifici in ASM per facilitare la comprensione del codice, o per supportare il debug. Per forzare un commento in assembly, utilizzare quanto segue:

```
asm showdebug +AGA-commento+AGA-
```

Dove 'commento' verrà inserito nel file ASM.

Esempio.

Il file sorgente contiene quanto segue, dove si trova il testo del commento OSCCON type is 100

```
asm showdebug OSCCON type is 100
OCCON1 +AD0- 0x60
```

L'assembly generato sarà AOE- il seguente: questo presuppone che la modalità AOE- dettagliata non sia selezionata.

```
INITSYS
+ADs-osccon type is 100
movlw 96
banksel OSCCON1
```

+AD0APQ- +ACo-Panoramica sulle librerie+ACo-

+ACo-Infomazioni sulle librerie+ACo-

Great Cow BASIC (come con la maggior parte degli altri linguaggi di programmazione dei microcontrollori) supporta le librerie.

Puoi creare la tua libreria specifica per il tuo dispositivo, non sei limitato a quelli mostrati di seguito. Se crei una nuova libreria specifica per dispositivo - si prega di inviare per l'inclusione nella prossima versione tramite il Great Cow BASIC Forum.

La manutenzione di queste librerie è completata dal team di sviluppo di Great Cow BASIC. Se vuoi, puoi adattare queste librerie, dovresti creare una copia locale, modificare e salvare all'interno della struttura del tuo file di sviluppo. Il team di sviluppo può aggiornare queste librerie come parte di una release e non vogliamo perdere le modifiche locali.

Per utilizzare una libreria, è sufficiente includere quanto segue nel codice utente:

```
+ACM-include +ADw-3PI.H+AD4- 'questo includera+AOE- le funzionalita+AOE- 3PI all'interno
del programma Per utilizzare una copia locale di una libreria, e sufficiente includere
quanto segue nel codice utente
```

```
+ACM-include +ACI-C: +AFw- mydev +AFw- library +AFw- 3pi.h+ACI-
```

'questo includera+AOE- una copia locale del 3pi nel tuo programma Great Cow BASIC supporta le seguenti librerie di dispositivi.

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHwAKg-
Libreria+ACo- +AHwAKg-Classe+ACo- +AHwAKg-Uso+ACo- +AHw-3PI +AHw-Polulu 3pi robot +AHw-

Una libreria che interfaccia il PWM per pilotare i motori.

+AHw-47XXX+AF8-EERAM.H +AHw-I2C EERAM memory +AHw-Una libreria specifica per dispositivo per le classi di dispositivi EERAM Microchip

+AHw-ALPS-EC11 +AHw-Rotary Encoder +AHw-Una libreria specifica per dispositivo per un encoder rotativo.

+AHw-ADS7843 +AHw-Touch Shield +AHw-Una libreria che si interfaccia con il touch screen ADS7843.

+AHw-BME280 +AHw-Sensore di temperatura, umidita e pressione +AHw-Una libreria che si interfaccia con il sensore BME280 e BMP280.

+AHw-CHIPINO +AHw-Shield +AHw-Una libreria che interfaccia la scheda Chipino con indirizzi di porta come quelli di Arduino.

+AHw-DHT +AHw-Temperatura e umidita +AHw-Una libreria che supporta i sensori di temperatura e umidita DHT22 e DHT11.

+AHw-DS1307 +AHw-Clock +AHw-libreria ch supporta il clock timer e le funzioni NVRAM.

+AHw-DS1672 +AHw-Clock +AHw-Una libreria che supporta l'orologio del timer e le funzioni della NVRAM.

+AHw-DS18B20 +AHw-Temperature +AHw-Una libreria che supporta le funzioni di temperatura.

+AHw-DS18SB0MultiPort +AHw-Temperature +AHw-Una libreria che supporta le funzioni di temperatura con dispositivi collegati a piu porte.

+AHw-DS18S20 +AHw-Temperature +AHw-Una libreria che supporta le funzioni di temperatura.

+AHw-DS2482 +AHw-Clock +AHw-Una libreria che supporta le funzioni OneWire di Dallas I2C.

+AHw-DS3231 +AHw-Clock +AHw-A library that supports the timer clock and NVRAM functions.

+AHw-DUEMILANOVE +AHw-Shield +AHw-Una libreria che interfaccia la scheda Duemilanove con indirizzi di porta come Arduino.

+AHw-EMC1001 +AHw-Temperature +AHw-Una libreria che supporta le funzioni di temperatura e le altre funzionalita del dispositivo.

+AHw-FRAM +AHw-I2C Eeprom +AHw-Una libreria che supporta le funzioni di memoria.

+AHw-GETUSERID +AHw-Microchip read ID +AHw-Una libreria che supporta l'identificazione dei microcontrollori Microchip.

+AHw-GLCD +AHw-Graphical LCD +AHw-Una libreria di base per il supporto grafico LCD.

+AHw-GLCD+AF8-HX8347 +AHw-Graphical LCD +AHw-Una libreria specifica per dispositivo per un

LCD grafico.

+AHw-GLCD+AF8-ILI9340 +AHw-LCD grafico +AHw- Una libreria specifica per dispositivo per un LCD grafico.

+AHw-GLCD+AF8-ILI9341 +AHw-LCD grafico +AHw- Una libreria specifica per dispositivo per un LCD grafico.

+AHw-GLCD+AF8-ILI9481 +AHw-Graphical LCD +AHw-LCD grafico +AHw- Una libreria specifica per dispositivo per un LCD grafico.

+AHw-GLCD+AF8-ILI9486L +AHw-Graphical LCD +AHw-A device specific library for an Graphical LCD.

+AHw-GLCD+AF8-IMAGESANDFONTS+AF8-ADDIN3 +AHw-Graphical LCD +AHw-LCD grafico +AHw- Una libreria specifica per dispositivo per un LCD grafico.

+AHw-GLCD+AF8-KS0108 +AHw-Graphical LCD +AHw-LCD grafico +AHw- Una libreria specifica per dispositivo per un LCD grafico.

+AHw-GLCD+AF8-NEXTION +AHw-Graphical LCD +AHw- LCD grafico +AHw- Una libreria specifica per dispositivo per un LCD grafico.

+AHw-GLCD+AF8-PCD8544 +AHw-Graphical LCD +AHw- LCD grafico +AHw- Una libreria specifica per dispositivo per un LCD grafico.

+AHw-GLCD+AF8-SH1106 +AHw-Graphical LCD +AHw-A device specific library for an Graphical LCD.

+AHw-GLCD+AF8-SSD1289 +AHw-Graphical LCD +AHw-A device specific library for an Graphical LCD.

+AHw-GLCD+AF8-SSD1306 +AHw-Graphical LCD +AHw-A device specific library for an Graphical LCD.

+AHw-GLCD+AF8-SSD1331 +AHw-Graphical LCD +AHw-A device specific library for an Graphical LCD.

+AHw-GLCD+AF8-ST7735 +AHw-Graphical LCD +AHw-A device specific library for an Graphical LCD.

+AHw-GLCD+AF8-ST7920 +AHw-Graphical LCD +AHw-A device specific library for an Graphical LCD.

+AHw-HEFLASH +AHw-HEF Memory Driver +AHw-Una libreria che supporta le funzioni di memoria HEF.

+AHw-HMC5883L +AHw-Triple-axis Magnetometer +AHw- Una libreria che supporta le funzioni del magnetometro.

+AHw-HWI2C+AF8-ISR+AF8-HANDLER +AHw-I2C Slave Driver +AHw-Una libreria che supporta l'uso di un microcontrollore Microchip come slave I2C.

+AHw-HWI2C+AF8-MESSAGEINTERFACE +AHw-I2C Slave +AHw-A support library that supports the use of a Microchip microcontroller as an I2C slave.

+AHw-HWI2C+AF8-ISR+AF8-HANDLERKMODE +AHw-I2C Slave Driver +AHw-A library that supports the use of a Microchip microcontroller as an I2C slave.

+AHw-HWI2C+AF8-MESSAGEINTERFACEKMODE +AHw-I2C Slave +AHw-Una libreria di supporto, che supporta l'uso di un microcontrollore Microchip come I2C slave.

+AHw-I2CEEPROM +AHw-I2C EEPROM memory +AHw-Una libreria che supporta le funzioni di memoria.

+AHw-LCD2SERIALREDIRECT +AHw-LCD to Serial Handler +AHw-Una libreria che supporta l'uso di un terminale seriale e PC come uno psuedo LCD.

+AHw-LEGO-PF +AHw-Lego Mindstorms shield +AHw-Una libreria che supporta l'uso di un terminale seriale e PC come uno psuedo LCD.

+AHw-LEGO +AHw-Lego Mindstorms shield +AHw- Una libreria che supporta il robot Lego Mindstorms

+AHw-MATHS +AHw-Maths routines +AHw-Una libreria che supporta funzioni matematiche come log, power e atan.

+AHw-MAX6675 +AHw-Temperature +AHw-Una libreria che supporta le funzioni di temperatura.

+AHw-MAX7219+AF8-ledmatrix+AF8-driver +AHw-LED 8+ACo-8 Matrix driver +AHw-Una libreria che supporta le matrici LED 8 +ACo- 8 MAX7219

+AHw-MCP23008 +AHw-i2C to serial +AHw-Una libreria che supporta le funzioni I2C in serie.

+AHw-MCP23017 +AHw-i2C to serial +AHw-Una libreria che supporta le funzioni seriali I2C.

+AHw-MCP23017 +AHw-i2C to serial +AHw-Una libreria che supporta le funzioni seriali I2C.

+AHw-MCP4XXXDIGITALPOT +AHw-Digital Pot +AHw-Una libreria che supporta la gamma di potenziometri digitali MCPxxxx.

+AHw-MCP7940N +AHw-Clock +AHw- Una libreria che supporta il clock del timer e le funzioni della NVRAM.

+AHw-NUNCHUCK +AHw-Game controller +AHw-Una libreria che supporta il controller di gioco NunChuck.

+AHw-PCA9685 +AHw-PWM +AHw-Una libreria specifica per dispositivo per il driver PWM a 16 canali. Guarda le dimostrazioni per esempio sull'uso. Supporta fino a quattro dispositivi tramite il bus I2C.

+AHw-PCF8574 +AHw-GLCD +AHw-Una libreria specifica per dispositivo per un LCD grafico.

+AHw-SD +AHw-SD Card +AHw-Una libreria specifica per dispositivo per una scheda SD.

+AHw-SMT+AF8-Timers +AHw-Signal Measurment Timer +AHw-Una libreria per la misurazione del segnale tramite il Timer per microcontrollori Microchip specifici.

+AHw-SOFTSERIAL +AHw-Serial +AHw-Una libreria per seriale software.

+AHw-SOFTSERIALCH1 +AHw-Serial +AHw-Una libreria per seriale1 software.

+AHw-SOFTSERIALCH2 +AHw-Serial +AHw-Una libreria per seriale software2.

+AHw-SOFTSERIALCH3 +AHw-Serial +AHw- Una libreria per seriale software3.

+AHw-SONGLAY +AHw-Music +AHw- Una libreria per ascoltare musica. Supporta il formato QBASIC e RTTTL.

+AHw-SONYREMOTE +AHw-Infrared +AHw-Una libreria che supporta le funzioni di un telecomando Sony.

+AHw-SRF02 +AHw-Distance Sensor +AHw-Una libreria che supporta il sensore ultrasonico SRF02.

+AHw-SRF04 +AHw-Distance Sensor +AHw-Una libreria che supporta il sensore ultrasonico SRF04.

+AHw-TEA5767 +AHw-I2C Radio +AHw-Una libreria che supporta il TEA5767 radioricevitore .

+AHw-TRIG2PLACES +AHw-Maths functions +AHw-Una libreria matematica che supporta la trigonometria in due punti.

+AHw-TRIG3PLACES +AHw-Maths functions +AHw-Una libreria matematica che supporta la trigonometria in tre punti.

+AHw-TRIG4PLACES +AHw-Maths functions +AHw-Una libreria matematica che supporta la trigonometria in quattro punti

+AHw-UNO+AF8-MEGA328P +AHw-Shield +AHw-Una libreria che si interfaccia con lo shield e indirizzi di porta come Arduino.

+AHw-USB +AHw-USB Supoort +AHw-Una libreria che interfaccia l'USB per i microcontrollori 16f e 18f.

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

Great Cow BASIC supporta le seguenti librerie principali. Queste librerie sono automaticamente incluse nel tuo programma utente quindi non e necessario utilizzare '+ACM-include' per accedere alle funzionalita+AOE- delle librerie.

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

+AHwAKg-

Libreria+ACo- +AHwAKg-Classe+ACo- +AHwAKg-Uso+ACo- +AHw-7SEGMENT +AHw-7 Segment LED display +AHw-Una libreria che si interfaccia con i display a 7 segmenti.

+AHw-A-D +AHw-Analog to Digital +AHw-Una libreria che supporta le functionalita ADC.

+AHw-EEPROM +AHw-EEProm +AHw-Una libreria che supporta i dispositivi eeprom I2C.

+AHw-HWI2C +AHw-I2C +AHw-Una libreria che supporta i moduli hardware MSSP e TWI (I2C).

+AHw-HWI2C2 +AHw-I2C +AHw- Una libreria che supporta i moduli hardware MSSP e TWI (I2C) sul canale due.

+AHw-HWSPI +AHw-SPI +AHw- Una libreria che supporta i moduli hardware MSSP e TWI della SPI.

+AHw-I2C +AHw-I2C +AHw- Una libreria che supporta il Bus I2C software .

+AHw-KEYPAD +AHw-KeyPad +AHw-Una libreria che supporta una tastiera.

+AHw-PS2 +AHw-I2C +AHw-Una libreria che supporta la funzionalita della tastiera del PC.

+AHw-LCD +AHw-LCD +AHw- Una libreria che supporta la funzionalita LCD, la libreria supporta molti e diversi metodi di comunicazione.

+AHw-PWM +AHw-Pulse Width Modulation +AHw-Una libreria supporta la funzionalita PWM.

+AHw-RANDOM +AHw-Random Numbers +AHw-Una libreria supporta la funzionalita del numero casuale.

+AHw-REMOTE +AHw-Infrared +AHw-Una libreria che supporta le funzioni di un controllo remoto NEC.

+AHw-RS232 +AHw-Serial +AHw-Una libreria per comunicazioni seriali.

+AHw-SOUND +AHw-Tones +AHw-Una libreria per la generazione di suoni e toni

+AHw-STDBASIC +AHw-Utility Functions +AHw-La libreria che contiene molti dei metodi di utilita.

+AHw-STRING +AHw-String +AHw-La libreria che contiene i metodi di stringa.

+AHw-SYSTEM +AHw-System +AHw-La libreria che contiene i metodi di sistema.

+AHw-TIMER +AHw-Timers +AHw-La libreria che contiene i metodi del timer.

+AHw-USART +AHw-Serial +AHw-La libreria che contiene i metodi seriali hardware che utilizzano il modulo hardware equivilent MSSP o AVR.

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- Great Cow BASIC supporta le seguenti librerie principali. Queste librerie sono automaticamente incluse nel tuo programma utente quindi non e necessario utilizzare '+ACM-include' per accedere alle funzionalita+AOE- delle librerie.

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHwAKg-
 Libreria+ACo- +AHwAKg-Classe+ACo- +AHwAKg-Uso+ACo-

+AHw-7SEGMENT +AHw-display a 7 Segment a LED +AHw-Una libreria che si interfaccia con i display a 7 segmenti.

+AHw-A-D +AHw-Analogico Digitale +AHw-Una libreria che supporta le funzionalita ADC.

+AHw-EEPROM +AHw-EEProm +AHw-Una libreria che supporta i dispositivi eeprom I2C.

+AHw-HWI2C +AHw-I2C +AHw-Una libreria che supporta i moduli hardware MSSP e TWI di I2C.

+AHw-HWI2C2 +AHw-I2C +AHw-Una libreria che supporta i moduli hardware MSSP e TWI (I2C) sul canale due.

+AHw-HWSPI +AHw-SPI +AHw-Una libreria che supporta i moduli hardware MSSP e TWI della SPI.

+AHw-I2C +AHw-I2C +AHw-Una libreria che supporta il bus I2C software.

+AHw-KEYPAD +AHw-KeyPad +AHw-Una libreria che supporta una tastiera.

+AHw-PS2 +AHw-2C +AHw-Una libreria che supporta la funzionalita della tastiera del PC.

+AHw-LCD +AHw-LCD +AHw-Una libreria che supporta la funzionalita LCD, la libreria supporta molti e diversi metodi di comunicazione.

+AHw-PWM +AHw-Modulazione a Larghezza di Impulso +AHw-Una libreria supporta la funzionalita PWM.

+AHw-RANDOM +AHw-Random Numbers +AHw-Una libreria supporta la generazione dei numeri casuali.

+AHw-REMOTE +AHw-Infrared +AHw-Una libreria che supporta le funzioni di un controllo remoto NEC.

+AHw-RS232 +AHw-Serial +AHw-Una libreria per comunicazioni seriali.

+AHw-SOUND +AHw-Tones +AHw-Una libreria per la generazione di suoni e toni.

+AHw-STDBASIC +AHw-Utility Functions +AHw-La libreria che contiene molti dei metodi di utilita.

+AHw-STRING +AHw-String +AHw-La libreria che contiene i metodi di stringa.

+AHw-SYSTEM +AHw-System +AHw-La libreria che contiene i metodi di sistema.

+AHw-TIMER +AHw-Timers +AHw-La libreria che contiene i metodi del timer.

+AHw-USART +AHw-Serial +AHw-La libreria che contiene i metodi seriali hardware che utilizzano il
 modulo hardware equivalent MSSP o AVR.

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

+AD0APQ- +ACo-Ringraziamenti+ACo-

Hugh Considine - Principale sviluppatore di Great Cow BASIC.

Stefano Bonomi - Subroutine LCD a due fili.

Geordie Millar - subroutine Swap e Swap4.

Jacques Nilo - HEFM e file di aiuto per la conversione in asciidoc.

Finn Stokes - Routine moltiplicazione a 8 bit, codice di accesso alla memoria di programma.

Evan Venn - Utilita, routine I2C riviste, questo file di aiuto e in generale tutto il resto+ACE-

+ACo-Contributori della traduzione:+ACo-

Stefano Delfiore - Italiano

Pablo Curvelo - Spagnolo

Murat Inceer - Turco

+ACo-Altri Contributori:+ACo-

Russ Hensel - Great Cow BASIC Note.

Chuck Hellebuyck - La sua documentazione per il GLCD e altri particolari, vedi <http://url+AFs-elproducts.com+AF0->.

Frank Steinberg - GCB+AEA-SYN IDE per Great Cow BASIC, vedere <http://url+AFs-sourceforge.net/projects/gcbasi/files/IDE/GCB+ACU-40Syn/GCB+ACU-40Syn.zip+AF0->

Alexy T. - SynWrite IDE utilizzato per GCB IDE, vedi <http://url+AFs-uvviewsoft.com/synwrite+AF0->

Thomas Henry per gli esempi Select Case e Sine Table.

William Roth per il codice LCD e gli schemi di supporto.

Theo Loermans per le sezioni LCD rivedute e la libreria.

Chris Roper per i metodi bit a bit, inclusi nella libreria FnEquBit, FnNotBit, FnIslBit, FnlsrBit, SetWith e 47xxx.

+ACo-Conversione dei file di documentazione di asciidoctor:+ACo-

Vedere <http://url+AFs-tasciidoctor> Web site+AF0- e il <http://html+AFs-support> forum+AF0-.

+AD0APQ- +ACo-Fondamenti del microcontrollore+ACo-

+ACo-Entrate e Uscite+ACo-

La maggior parte dei pin di uso generico su un microcontrollore può funzionare in una delle due modalità: modalità di ingresso o modalità di uscita.

Quando agisce come ingresso, il pin di ingresso / uscita per uso generico verrà inserito in stato di alta impedenza. Il microcontrollore percepirà quindi il pin di ingresso / uscita per uso generale e il programma può leggere lo stato del pin di ingresso / uscita per uso generale e prendere decisioni basate su di esso.

In modalità di uscita, il microcontrollore connetterà il pin di ingresso / uscita per uso generico a entrambi Vcc (positiva), o Vss (ground), negativa. Il programma può quindi impostare lo stato del pin di ingresso / uscita generico su un livello alto o basso.

Great Cow BASIC tenterà di determinare la direzione di ciascun pin di ingresso / uscita per scopi generici e impostarlo in modo appropriato, quando possibile. Great Cow BASIC cercherà di impostare la direzione per scopo generale pin di ingresso / uscita. Tuttavia, se il pin di ingresso / uscita di uso generale viene letto e scritto nel tuo programma, quindi il pin di ingresso / uscita per uso generico deve essere configurato in ingresso o in uscita xref:a-13+AFs-Dir+AF0- appropriati.

+ACo-Esempio di comandi dir.+ACo-

```
'L'indirizzo della porta è specifico del microcontrollore. Portx.x è un caso comune per i
PIC e AVR
dir portb.0 in
dir portb.1 out
'L'indirizzo della porta è specifico del microcontrollore.
gpio.x è un caso generale per alcuni PIC gpio.x is a general case for some PICs
dir gpio.0 in
dir gpio.1 Out
'Set the whole port as an output
dir portb out
dir gpio out
'Set the whole port as an input
dir portc in
dir gpio in
```

+ACo-Specifiche Microchip per operazioni di lettura / scrittura+ACo-

Per le porte specifiche e pin di ingresso / uscita per uso generico disponibili per uno specifico microcontrollore si prega di fare riferimento alla scheda tecnica.

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHwAKg-Port+ACo-
+AHwAKg-Scopo+ACo- +AHwAKg-Esempi+ACo- +AHw-PORTx mappe per i microcontrollori pin digitali

da 0 a 7. Dove x può essere a,b,c,d,e,f or g +AHw-Read: PORTx il registro dei dati di porta per una lettura operation. +AHw-uservar+AD0-PORTA uservar+AD0-PORTA .1 +AHw-PORTx mappe per microcontrollori pin digitali da 0 a 7. Dove x può essere a,b,c,d,e,f or g +AHw-Write: PORTx il registro dei dati di porta per un'operazione di scrittura e, laddove LATx non è richiesto in quanto Great Cow BASIC implementerà LATx quando necessario. Vedere xref:a-14+AFs-Option NoLach+AF0- per maggiori informazioni sui registri LAT e come per disabilitare questa funzione automatica. +AHw-PORTA+AD0-255 PORTA.1+AD0-1 +AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

Per leggere un pin di ingresso / uscita generico, è necessario assicurarsi che la direzione sia corretta DIR Portx IN is set (default is IN) o al set specifico di bit di porta. Dove uservar +AD0- PORTx.n può essere utilizzato.

Esempio:

```
uservar +AD0- PORTb.0
uservar +AD0- PORTb
```

Per scrivere su un pin di ingresso / uscita generico, è necessario assicurarsi che la direzione sia corretta DIR Portx OUT porta o un insieme specifico di bit di porta. Dove PORTx.n +AD0- uservar può essere utilizzato.

Esempio:

```
PORTb.0 +AD0- uservar
PORTb +AD0- uservar
```

+ACo-ATMEL specifiche per operazioni di lettura / scrittura+ACo-

Using a Mega328p as a general the following provides insights for the AVR devices. For the specific ports and general purpose input/output pins available for a specific microcontroller please refer to the datasheet.

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHwAKg-Porta+ACo-
+AHwAKg-Operazioni di scrittura+ACo- +AHwAKg-Operazioni di lettura+ACo- +AHw-PORTD esegue il mapping ai pin digitali Mega328p (e, i microcontrollori AVR) da 0 a 7 +AHw-PORTD - Port D Data Register: operazione di scrittura (un'operazione di lettura su una porta fornirà lo stato di pull-up) +AHw-PIND - Il registro dei pin di ingresso della porta D - sola lettura

+AHw-PORTB esegue il mapping ai pin digitali da 8 a 13 di Mega328p (e, dei microcontrollori AVR). I due bit alti (6 e 7) mappano ai pin di cristallo e non sono utilizzabili +AHw-PORTB - Registro dati porta B: operazione di scrittura (un'operazione di lettura su una porta fornirà lo stato di pull-up) +AHw-PINB - Il registro dei pin di ingresso della porta B - solo lettura

+AHw-PORTC esegue il mapping ai pin analogici Mega328p (e, i microcontrollori AVR) da 0 a 5. I pin 6 e

7 sono accessibili solo su Mega328p (e, i microcontrollori AVR) Mini+AHw-PORTC - Registro dati porta C: operazione di scrittura (un'operazione di lettura su una porta fornirà lo stato di pull-up)+AHw-PINC - Il registro dei pin di ingresso della porta C - solo lettura +AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

+ACo-Esempi:+ACo-

```
uservar +AD0- PORTb.0  
uservar +AD0- PORTb
```

+ACo-Settare le Porte e i bit delle Porte+ACo-

e possibile impostare una porta come mostrato sopra con una variabile, oppure, è possibile impostare con una costante o qualsiasi combinazione utilizzando gli operatori bit a bit e logici.

```
+ACM-define InitStateofPort 0b11110000 PORTb +AD0- InitStateofPort 'imposterà  
incondizionatamente i bit 4:7
```

```
PORTb +AD0- 0b11110000 'imposterà incondizionatamente bit 4:7
```

```
PORTb +AD0- uservar OR 0b11110000 'uservar o bit 4:7 per controllo che bit 4:7 sono  
visualizzate a 1
```

e anche valido quanto segue: leggere un port.bit e quindi impostare port.bit con una variabile o un valore di porta. Come mostrato sotto.

```
dir portb out  
portb.0 +AD0- NOT portb.0
```

Il codice utente sopra può causare problemi con glitches quando si verificano operazioni di lettura e scrittura. Diamo un'occhiata all'assembly generato.

```
+ADs-portb.0 +AD0- NOT portb.0
banksel SYSTEMP1
clrf SysTemp1
btfsc PORTB,0
incf SysTemp1,F
comf SysTemp1,F
bcf PORTB,0 btfsc
SysTemp1,0
bsf PORTB,0
```

Per risolvere eventuali glitches aggiungere +ACM-option Volatile nel codice.

```
+ACM-option Volatile portb.0
dir portb out
portb.0 +AD0- NOT portb.0
```

Questa opzione fornisce il seguente assembly che risolve il problema tecnico.

```
+ADs-portb.0 +AD0- NOT
portb.0 banksel SYSTEMP1
clrf SysTemp1
btfsc PORTB,0
incf SysTemp1,F
comf SysTemp1,F
btfsc SysTemp1,0
bsf PORTB,0 btfss
SysTemp1,0
bcf PORTB,0
```

Vedere anche xref:a-15+AFs-Dir, +ACM-Option Volatile+AF0-

+ACo-Configurazione+ACo-

(Nota: questa sezione non si applica ai microcontroller Atmel AVR. I microcontrollori AVR hanno impostazioni di configurazione simili, ma sono controllate tramite +ACI-Fusibili di configurazione+ACI-. Con Great Cow BASIC non e possibile impostare questi Fusibili e necessario utilizzare il software del programmatore).

Ogni PIC Microchip ha una Word CONFIG. Questa e un'area della memoria sul chip che memorizza le impostazioni che regolano il funzionamento del chip.

I seguenti aspetti del chip sono regolati dalla CONFIG Word:

+ACo- Selezione dell'oscillatore: il chip verra eseguito da un oscillatore interno o esterno? +ACo-

Ripristino automatico: il chip dovrebbe essere ripristinato se la tensione scende oltre il limite previsto. Se rileva che sta eseguendo lo stesso blocco di codice più e più volte? +ACo- Protezione del codice: quali aree della memoria devono essere nascoste una volta scritte? +ACo- Utilizzo dei pin: quali pin sono disponibili per la programmazione, il reset del chip o l'emissione di segnali PWM?

Le esatte impostazioni di configurazione variano tra i chip. Per trovare un elenco di impostazioni valide, consultare la scheda tecnica per i microcontrollori che desideri utilizzare.

Questo potrebbe creare confusione quindi, Great Cow BASIC imposterà automaticamente alcuni registri di configurazione, salvo diversa indicazione:

+ACo- Programming a bassa tensione (LVP) non abilitata. Ciò consente al pin PGM (di solito B3 o B4) di essere usato come un normale pin I / O. +ACo- Watchdog Timer (WDT) non abilitato. Il WDT ripristina il chip se esegue lo stesso pezzo di codice più e più volte - questo può causare problemi con alcune delle routine di delay più lunghe in Great Cow BASIC. +ACo- Master Clear (MCLR) non abilitato dove possibile. Su molti chip più recenti ciò consente il pin MCLR (spesso PORTA.5 o PORTE.3) di essere utilizzato come porta di input standard. Rimuove anche la necessità di un resistore di pull-up sul pin MCLR. +ACo- Verrà selezionata una modalità oscillatore, basata sulle seguenti regole: +ACo- Se il microcontrollore ha un oscillatore interno e l'oscillatore interno è in grado di farlo generando la velocità specificata nella linea +ACM-chip, verrà utilizzato l'oscillatore interno. +ACo- Se la velocità del clock supera i 4 Mhz, viene selezionato l'oscillatore HS esterno. +ACo- Se la velocità di clock è 4 MHz o inferiore, viene selezionata la modalità oscillatore XT esterno.

Si noti che queste impostazioni possono essere facilmente sovrascritte individualmente quando necessario. Ad esempio, se è necessario il timer watchdog, aggiungendo la linea seguente.

```
+ACM-config WDT +AD0- ON
```

Cio abiliterà il timer del watchdog, senza influire su altre impostazioni di configurazione.

+ACo-Utilizzo della configurazione+ACo-

Una volta stabilite le necessarie opzioni di CONFIG, aggiungerle al programma è facile. Alla stessa riga prima citata o con una nuova riga +ACIAIw-config+ACI- e quindi elencare le opzioni desiderate separate da virgole, come in questa riga:

```
+ACM-config OSC +AD0- RC, BODEN +AD0- OFF
```

Great Cow BASIC supporta anche questo formato sui chip della serie 10/12/16:

```
+ACM-config INTOSC+AF8-OSC+AF8-NOCLKOUT, BODEN+AF8-OFF
```

Tuttavia, per la compatibilità verso l'alto con i chip 18F, è necessario utilizzare le impostazioni di

configurazione dello stile +AD0-.

e possibile avere diverse linee `+ACM-config` in un programma, ad esempio una nel programma principale e uno in ciascuno di diversi file `+ACM-include`. Tuttavia, è necessario prestare attenzione per garantire che le impostazioni in un file non siano in conflitto con quelli in un altro.

Per ulteriore aiuto, vedere [xref:a-16+AFsAIw-config direttive+AF0-](#)

+ACo-Array+ACo-

+ACo-Informazioni sugli Array+ACo-

Un array è un tipo speciale di variabile, che può memorizzare più valori contemporaneamente. È essenzialmente una lista di numeri di byte in cui ognuno può essere indirizzato individualmente attraverso l'uso di un +ACI-indice+ACI-. L'indice è un valore tra parentesi posizionato subito dopo il nome dell'array.

Esempi di nomi di array sono:

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHw-Indice/Array

+AHw-Significato +AHw-Fish(10) +AHw-Definizione di un array con 10 elementi chiamato Fish

+AHW-DataLog(2) +AHW- il secondo array chiamato DataLog

+AHW-ButtonList(Temp)	+AHW-Un elemento nell'array ButtonList che viene selezionato in base al valore	nella	variabile	Temp
-----------------------	--	-------	-----------	------

+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

+ACo-Definire un array+ACo-

Utilizzare il comando DIM per definire un array.

DIM array+AF8-title (number+AF8-of+AF8-element)

Il numero di elementi puo essere il numero o una costante,+ACoAKg-non una variabile+ACoAKg-.

Il valore per gli elementi numerici in un array deve essere un numero o una costante. Il compilatore si occupa di allocare RAM per gli array in fase di compilazione, e quindi non è possibile utilizzare una variabile perché durante la compilazione il valore di una variabile non può essere determinato.

+ACo-Impostazione di un intero array in una sola volta.+ACo-

e possibile impostare diversi elementi di un array con una singola riga di codice. Questo breve esempio mostra come:

```
Dim TestVar(10) TestVar = 1, 2, 3, 4, 5, 6, 7, 8, 9
```

Quando si utilizza il metodo sopra l'elemento 0 dell'array, TestVar verrà impostato sul numero di elementi dell'elenco, che in questo caso è 9. Ogni elemento dell'array verrà quindi caricato con il valore corrispondente nell'elenco: quindi nell'esempio, TestVar(1) sarà impostato su 1, TestVar(2) su 2 e così via.

+A Co-Solo l'elemento 0 è impostato sul numero di elementi nell'array quando si utilizza questo metodo di dati fino a 48 elementi. Array Length L'elemento 0 non deve essere utilizzato per ottenere la lunghezza dell'array. L'elemento 0 sarà solo coerente con la lunghezza della matrice quando l'array è impostato come mostrato sopra.+A Co-

Il metodo corretto consiste nell'utilizzare una costante per impostare la dimensione dell'array e utilizzare la costante all'interno del codice per ottenere la lunghezza dell'array.

```
+A CM-Define ArraySizeConstant 500 Dim TestVar( ArraySizeConstant ) SerPrint  
ArraySizeConstant 'o, altro uso
```

+A Co-Utilizzo di array: (o matrici)+A Co-

Per utilizzare una matrice, viene specificato il suo nome, quindi l'indice. Le matrici possono essere utilizzate ovunque dove normali variabili possono essere usate.

Il limite della dimensione dell'array varia in base al tipo di chip:

1. La serie di chip 12F / 16F il limite dell'array è limitato a pochi byte per la gestione dell'array, la RAM disponibile è limitata.
2. Per l'Atmel AVR o un 18F non c'è limite oltre alla RAM libera.
3. Tuttavia, Great Cow BASIC limita la dimensione di qualsiasi array a 10.000 elementi.

+A Co-Ottenere il massimo dalla memoria disponibile:+A Co-

L'utilizzo della RAM dell'array è determinato dall'architettura del tipo di chip. Ottenere il massimo dalla memoria disponibile è determinata dall'assegnazione dell'array all'interno dei bank di memoria disponibili.

Un esempio è un array di 6 o 7 byte quando ci sono solo 24 byte di RAM e i 24 byte sono divisi attraverso più bank di memoria. Supponiamo in questo esempio che 18 byte siano stati assegnati ad altre variabili e ci sono 29 byte totali disponibili. Un array di 6 byte si inserisce nello spazio libero in un bank, ma un array di 7 no.

Great Cow BASIC attualmente non può dividere un array su diversi bank, quindi se ci sono 6 byte liberi in un bank e 5 in un altro, non puoi avere una matrice di 7 byte. Questo sarebbe molto difficile da fare in modo efficiente nei 12F / 16F, come ci sarebbe una serie di registri di funzioni speciali nel mezzo

della matrice quando si utilizzano i 12F o 16F. Questo vincolo non e il caso di 16F1 / 18F poiche l'indirizzamento lineare lo rende facile da estendere nei bank perche gli SFR non sono un problema (come con 12F / 16F).

+ACo-Usare le tabelle come alternativa.+ACo-

Se ci sono molti elementi nella matrice, potrebbe essere meglio usare una tabella di ricerca per archiviare i dati, e poi copiare alcuni degli elementi di dati in un array piu piccolo, se necessario.

Per ulteriore aiuto vedere xref:a-17+AFs-Declaring arrays e DIM+AF0-

+ACo-Commenti+ACo- +ACo-Informazioni sui Commenti+ACo-

L'aggiunta di commenti al tuo programma Great Cow BASIC viene eseguita con un apostrofo prima della linea di commento. Puoi anche commentare le sezioni di codice se vuoi semplicemente posizionando un apostrofo, un punto e virgola o utilizzare la dichiarazione REM all'inizio di ogni riga. L'IDE SynGCB ha una funzione per farlo automaticamente.

Avvertenza: il BASIC grafico di Great Cow utilizza il punto e virgola per contrassegnare i commenti che ha inserito automaticamente. Non legge questi commenti quando apre un file, quindi qualsiasi commento in un programma Great Cow Basic. Il programma BASIC che inizia con un punto e virgola verra eliminato se il programma viene aperto utilizzando Great Cow BASIC Grafico.

+ACo-Esempio:+ACo-

```
'Il numero di pin che lampeggiano
+ACM-define FlashPins 2 REM e possibile creare un'intestazione utilizzando un apostrofo
prima di ogni riga
REM Questo e un ottimo modo per descrivere il tuo programma
REM Puoi anche usarlo per descrivere le connessioni hardware.
'e possibile inserire commenti sopra il comando o sulla stessa riga Dir PORTB Out
' Inizializza tutta la PORTB come uscita ' Main loop
do PORTB +AD0- 0
' tutti i pin off Wait 1 S
' pausa 1 secondo PORTB +AD0- 0xFF
' tutti i pin on Wait 1 s
' pausa 1 secondo
Loop
```

+ACo-Condizioni+ACo-

+ACo-Informazioni sulle Condizioni+ACo-

In Great Cow BASIC (e nella maggior parte degli altri linguaggi di programmazione) una condizione, e un'affermazione che puo essere o vero o falso. Le condizioni sono usate quando il programma deve prendere una decisione. Una condizione e generalmente data come valore o variabile, un operatore relativo (come +AD0- o +AD4-), e un altro valore o variabile. Diverse condizioni possono essere

combinare per formare una condizione attraverso l'uso della logica operatori come AND e OR.

Great Cow BASIC supporta questi operatori relativi:

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHw-Simbolo
+AHw- Significato +AHw- +AD0- +AHw- Uguale a +AHw- +ADwAPg- +AHw-Diverso da +AHw- +ADw-
+AHw-Minore di +AHw- +AD4- +AHw-Maggiore di +AHw- +ADwAPQ- +AHw-Minore o uguale a +AHw-
+AD4APQ- +AHw-Maggior o uguale a
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-

Inoltre, questi operatori logici possono essere utilizzati per combinare più condizioni in una sola:

+AFs-width+AD0AIg-100+ACUAIg-,options+AD0AIg-header,footer+ACIAXQ-
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9- +AHw-Nome +AHw-
Abbreviazione +AHw- Se la condizione è Vera +AHw-AND +AHwAJg- +AHw-entrambe le condizioni
sono vere

+AHw-OR +AHw- +AHw-almeno una condizione è vera

+AHw-XOR +AHw- +ACM- +AHw-ola condizione è vera

+AHw-NOT +AHw- +ACE- +AHw- la condizione non è vera
+AHwAPQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9AD0APQA9-