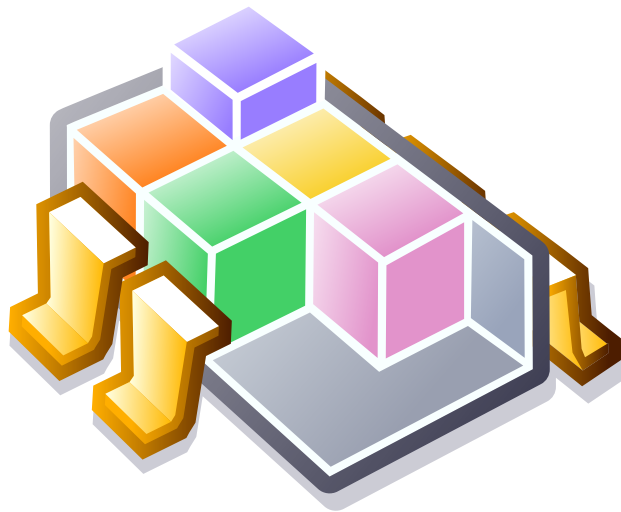


Manuale di KTechlab

David Saxton
Daniel Clarke



Manuale di KTechlab

Indice

1	Beve panoramica	6
1.1	Introduzione	6
1.2	Documenti	6
1.3	Disegno	7
2	I programmi PIC	8
2.1	Manipolazione	8
2.2	Caricamento	8
3	Circuit	10
3.1	Posizionamento dei componenti	10
3.2	Collegare i componenti	10
3.3	Attributi dei componenti	11
3.4	Simulazione	11
3.5	Oscilloscopio	12
3.6	Sotto-circuiti	12
4	FlowCode	13
4.1	Introduzione	13
4.2	Creare un programma	13
4.3	Impostazioni PIC	13
4.4	Annidare il FlowCode	14
5	Microbe	15
5.1	Introduzione e sintassi generale	15
5.1.1	Convenzioni sui nomi	15
5.1.2	Convenzioni sulle parentesi	16
5.1.3	Commentare	16
5.1.4	Struttura del programma	16
5.1.5	Subroutine	16
5.2	Riferimenti al linguaggio Microbe	17
5.2.1	if	17
5.2.2	alias	17
5.2.3	repeat	17

Manuale di KTechlab

5.2.4	while	18
5.2.5	goto	18
5.2.6	call	18
5.2.7	delay	18
5.2.8	sevensseg	19
5.2.9	keypad	19
5.3	Ingresso/uscita del PIC	20
5.3.1	Direzione della porta	20
5.3.2	Porte di ingresso/uscita	20
5.3.3	Piedini di ingresso/uscita	20
5.4	Variabili	21
5.4.1	Operazioni unarie	21
5.4.2	Aritmetica	21
5.4.3	Confronto	21
6	Debug	22
6.1	Avviare il debug	22
6.2	Controllare il debugger	22
7	FAQ	24

Sommario

KTechlab è un IDE per microcontrollori ed elettronica.

Capitolo 1

Beve panoramica

1.1 Introduzione

KTechlab è un IDE per microcontrollori ed elettronica. Può eseguire simulazioni di una gran varietà di componenti (logici, integrati, lineari, non lineari e reattivi), simulare e fare il debug dei microcontrollori PIC con gpsim. Viene anche fornito con i suoi linguaggi di alto livello, strettamente collegati e complementari: FlowCode e Microbe.

È stato progettato per essere semplice da usare e il meno invadente possibile: tutti i componenti (e anche FlowParts) hanno un aiuto contestuale sensibile, e le simulazioni elettroniche sono semplici come trascinare i componenti sull'area di lavoro e creare connettori che si instradano fra i loro piedini. FlowCode permette ai nuovi utenti dei PIC di creare istantaneamente i loro programmi, mentre la simulazione elettronica permette di attraversare il programma assembly del PIC all'interno di un circuito.

1.2 Documenti

Per iniziare con KTechlab hai bisogno di creare un nuovo documento, il cui tipo dipenderà dalla tua attività:

- Documento FlowCode - Costruisci un programma per PIC con i diagrammi di flusso.
- Documento Circuito - Simula i circuiti elettronici ed i microcontrollori.
- Documento Microbe - Linguaggio ad alto livello per i PIC, usato anche da FlowCode per generare il codice assembly.
- Documento Assembly - Inizia a scrivere un programma assembly per i PIC.

KTechlab usa un modello di visualizzazione del documento in cui la logica del documento è completamente separata dalla vista aperta del documento. Questo permette molte viste dello stesso file.

Quando viene creato un documento, la vista viene creata in una scheda separata. Ogni scheda può supportare un qualsiasi numero di viste, ripetute secondo un qualsiasi schema arbitrario. Questo consente, ad esempio, di simulare un programma PIC in un circuito, e di passare al programma in un documento di assembly nella stessa scheda.

Il contenuto delle schede può essere duplicato trascinando la scheda in un'area vuota della barra delle schede; possono inoltre essere inserite in una scheda esistente trascinandole su quella scheda.

Istruzioni dettagliate sul documento qui sopra possono essere trovate nei loro rispettivi capitoli.

1.3 Disegno

Nei documenti Circuito e FlowCode sono disponibili molti strumenti di disegno, incluso il testo, che sono raggiungibili facendo clic sull'icona di un pennello nella barra degli strumenti. Per disegnare trascina il mouse per formare sia una forma, sia una linea appropriata per lo strumento di disegno in uso.

Quando viene selezionato un disegno, questo può essere ridimensionato trascinando le sue maniglie. Tenendo premuto **Shift** durante il trascinamento la maniglia verrà agganciata alla griglia sottostante. Ciascuno strumento ha delle opzioni di base, tipo il colore, che sono accessibili dalla barra degli strumenti. Ci sono anche delle opzioni più avanzate che si trovano nella barra laterale **Editor oggetto**, come ad esempio le linee e gli stili della punta.

Capitolo 2

I programmi PIC

2.1 Manipolazione

Quando crei un documento (FlowCode oppure di testo) noterai un menu a cascata nella barra degli strumenti con l'icona di un razzo. Da qui puoi manipolare il tuo programma PIC, cambiandolo in una forma diversa.

- **Converti in Microbe** - viene usato solo nei documenti FlowCode. Viene spiegato meglio in capitolo 4.
- **Converti in Assembly** - può essere usato in quattro contesti. Se viene aperto un documento FlowCode esso visualizzerà il FlowCode come istruzioni assembly. Se viene aperto un documento Microbe invocherà invece il programma **microbe**, che è distribuito insieme KTechlab, per compilare il programma. In maniera simile, se viene aperto un programma C, tenterà di compilarlo con SDCC. Se infine viene aperto un documento di testo che contiene un PIC esadecimale, invocherà **gpdasm** per disassemblare l'esadecimale.
- **Converti in esadecimale** - può anche essere usato in quattro contesti. Al pari di **Converti in Assembly** può essere usato con FlowCode, Microbe e con i documenti C. Viene anche abilitato quando un documento assembly viene aperto per essere assemblato con **gpasm**.
- **Carica in PIC** - assembla il programma PIC che attualmente viene modificato, e lo carica usando il programmatore selezionato dall'utente.

Nessuna di queste azioni richiede che l'attuale documento venga salvato, il che è molto utile quando è richiesto di fare velocemente un programma. Se la destinazione non è un PIC, la finestra **Output** invocata facendo clic su una di queste azioni può sia mostrare il risultato (sempre come testo in tutti e tre i casi qui sopra) in un nuovo documento, sia in un file. Se il risultato viene salvato in un file è anche possibile caricarlo dopo la sua creazione, inoltre può anche essere aggiunto al progetto aperto (se ce n'è uno).

Nota che puoi fare in modo che KTechlab usi sempre la stessa vista per mostrare il contenuto generato, selezionando le opzioni **Generale** delle impostazioni.

2.2 Caricamento

KTechlab usa un programmatore di terze parti per trasferire il programma nel PIC. Un certo numero di essi è già predefinito, altri possono essere aggiunti dalla finestra **Impostazioni**.

La lista delle porte viene ottenuta cercando le porte seriali e le parallele che sono leggibili e scrivibili. Le porte sono cercate in:

Manuale di KTechlab

- `/dev/ttyS[0..7]`
- `/dev/tts/[0..7]`
- `/dev/ttyUSB[0..7]`
- `/dev/usb/tts/[0..7]`

Le porte parallele sono cercate in:

- `/dev/usb/parport[0..7]`
- `/dev/usb/parports/[0..7]`

Capitolo 3

Circuit

3.1 Posizionamento dei componenti

A sinistra, sarà disponibile la scheda **Componenti**.

Il trascinamento di un componente dalla barra laterale nel circuito, lo posizionerà sotto il cursore del mouse. In alternativa, è possibile fare doppio clic su un elemento nella barra laterale **Componenti** per aggiungerlo ripetutamente al circuito. In questa modalità, una copia del componente selezionato sarà posizionata ripetutamente con il tasto sinistro del mouse fino a quando **Esc** viene premuto o si fa clic con il pulsante destro del mouse.

Per riposizionare un componente, fare clic con il tasto sinistro e trascinare. Lo si troverà agganciato alla griglia sottostante. Se si trascina il componente fuori dai bordi destro o inferiore dell'area di lavoro, l'area di lavoro sarà ridimensionata per adattarsi.

Tutti i componenti hanno una nozione di orientamento; 0, 90, 180 e 270 gradi. Anche quelli che non sono simmetrici rispetto a un asse possono essere capovolti. Per ruotare una selezione di componenti, fare clic con il pulsante destro del mouse e selezionare dal menu **Orientazione** oppure fare clic sui pulsanti di rotazione nella barra degli strumenti. È possibile accedere a quest'ultimo premendo i tasti [e] (familiari agli utenti di Inkscape). La barra laterale **Elemento** (a destra) fornisce un metodo efficace per impostare l'orientazione fornendo anteprime dei componenti. Anche il lancio di componenti è possibile solo tramite la barra laterale **Elemento**.

3.2 Collegare i componenti

Esistono due modalità per la creazione di connessioni (cavi): automatica e manuale. Queste modalità sono selezionate tramite il menu a tendina **Modalità di instradamento della connessione** nella barra degli strumenti. Sperimentare con entrambi: l'instradamento automatico è spesso migliore per i piccoli circuiti, mentre i circuiti più complessi possono richiedere l'instradamento manuale.

In modalità automatica, creare una connessione trascinando da un pin componente o da una connessione esistente e rilasciando il mouse sul pin o sulla connessione desiderata. Si vedrà che la linea retta disegnata diventerà arancione quando sarà creata una connessione valida al rilascio del mouse. Se la linea che si sta disegnando è nera, è perché non c'è nulla sotto il cursore del mouse o si sta tentando di connettere due elementi già collegati. Durante il diagramma di flusso, i criteri per una connessione valida sono più complessi, ma ci arriveremo più avanti.

Il modo migliore per farsi un'idea dell'instradamento manuale delle connessioni è sperimentarlo. Fare clic sul pin o sulla connessione iniziale, quindi estendere il proto-connettore allontanando il mouse dal punto in cui si è fatto clic. Per posizionare un angolo, fare clic con il tasto sinistro. Per

annullare il disegno della connessione, premere **Esc** oppure fare clic con il pulsante destro del mouse.

KTechlab fa del suo meglio per mantenere i percorsi delle connessioni. Tuttavia, se si trascina un componente, i punti finali di una connessione si spostano l'uno rispetto all'altro, KTechlab sarà costretto a ridisegnare la connessione utilizzando l'instradamento automatico. Prima di spostare un componente, è possibile vedere quali connettori dovranno essere reindirizzati, poiché diventeranno grigi al clic.

Per rimuovere una connessione esistente, selezionarla disegnando un piccolo rettangolo di selezione su una parte del connettore e premere **Canc**.

3.3 Attributi dei componenti

La maggior parte dei componenti avrà attributi modificabili, come la resistenza per i resistori. Per impostazione predefinita, è possibile modificare semplici attributi nella barra degli strumenti, quando viene selezionato un gruppo dello stesso tipo di componenti. Se la selezione contiene una combinazione di diversi tipi di componenti (come resistori e condensatori), non saranno visualizzati attributi per la modifica.

Alcuni componenti hanno attributi più avanzati che non sono accessibili tramite la barra degli strumenti. Questi si trovano nella barra laterale **Elemento** sulla destra. Il diodo, ad esempio, ha diverse caratteristiche comportamentali che è possibile modificare qui.

Esiste un tipo di attributo che non può essere modificabile dalla barra degli strumenti o dalla barra laterale **Elemento** - testo multi-linea. Facendo doppio clic sull'elemento, si aprirà una finestra in cui è possibile inserire il testo.

3.4 Simulazione

In modo predefinito, la simulazione sarà eseguita quando si crea un nuovo circuito. Lo stato della simulazione viene visualizzato nella parte inferiore destra della vista di un circuito e può essere modificato tramite il menu **Strumenti**. Innanzitutto, una piccola spiegazione su come funziona il simulatore. Questo dovrebbe consentirvi di trarne il massimo.

Quando viene creato o modificato un circuito, le aree interessate vengono suddivise in gruppi di pin e connessioni che possono essere considerate indipendenti. Ogni gruppo viene quindi simulato come entità separata (sebbene interagisca ancora tramite i componenti), con la simulazione fornita in base alla complessità del gruppo. I gruppi complessi, come quelli che coinvolgono componenti non lineari come i LED, sono lenti da simulare. I gruppi che contengono solo pin logici, di cui solo uno controlla il valore su tali pin, sono i più veloci da simulare.

I risultati della simulazione sono forniti attraverso diversi mezzi grafici.

I pin sui componenti visualizzeranno barre laterali di tensione. Questi sono di colore arancione per tensione positiva e blu per tensione negativa. La loro lunghezza dipende dal livello di tensione e la loro larghezza dalla quantità di corrente che fluisce attraverso il pin. Questi possono essere disattivati nella pagina **Generale** della finestra **Configurazione**.

Passando il puntatore del mouse su un pin o una connessione sarà visualizzata un suggerimento che mostra la tensione e la corrente in quel punto del circuito. Numerosi componenti forniscono anche un riscontro grafico, ad esempio LED e voltmetri o amperometri.

Infine, c'è l'oscilloscopio, discusso nella prossima sezione.

3.5 Oscilloscopio

L'oscilloscopio può registrare dati logici, di tensione e di corrente. La sonda logica è ottimizzata per la memorizzazione di campioni booleani, quindi dovrebbe essere utilizzata al posto della sonda di tensione quando si misura la logica.

Per raccogliere dati, creare un nuovo componente sonda e collegarlo a un punto appropriato nel circuito. Si vedrà il risultato disegnato immediatamente nell'oscilloscopio. L'aggiunta di più sonde schiaccierà le uscite una accanto all'altra: è possibile riposizionarle trascinando le frecce a sinistra della vista dell'oscilloscopio e modificarne i colori tramite gli attributi della sonda.

Per le sonde di tensione e corrente, l'intervallo dei valori di ingresso può essere regolato nella barra laterale **Editor degli elementi** sulla destra.

L'ingrandimento è controllato da un cursore. Il ridimensionamento è logaritmico; per ogni pochi pixel su cui si muove il cursore, il fattore di ingrandimento sarà moltiplicato per una costante. KTechlab simula la logica con una precisione massima di 1 microsecondo e al massimo livello di ingrandimento, un microsecondo è rappresentato da 8 pixel.

Quando la barra di scorrimento viene trascinata fino alla fine, rimarrà lì fino a quando vengono registrati nuovi dati. Altrimenti, la posizione della barra di scorrimento rimane fissa nel tempo. La vista dell'oscilloscopio può anche essere spostata avanti e indietro facendo clic con il tasto sinistro e trascinando la vista. A causa delle limitazioni del sistema di oggetti sottostante, lo scorrimento sarà molto granulare al massimo livello di ingrandimento.

Facendo clic con il tasto destro sulla vista dell'oscilloscopio, si apre un menu in cui è possibile controllare il numero di volte in cui la vista dell'oscilloscopio viene aggiornata. Ciò consente una visualizzazione più fluida, o riduce l'utilizzo della CPU.

3.6 Sotto-circuiti

I sotto-circuiti offrono un modo riutilizzabile e ordinato di utilizzare un circuito, quando si è interessati solo a interagire con connessioni esterne al circuito. Il sotto-circuito viene creato come un IC, con i pin che fungono da interazione con il circuito interno.

Innanzitutto, deve essere costruito il circuito da utilizzare come modello per la creazione di un sotto-circuito. I punti di interazione sono definiti tramite i componenti di **Connessione esterna**. Questi devono essere collegati e posizionati nel punto in cui si desidera posizionare il pin sul sotto-circuito IC.

Quindi, selezionare il gruppo di componenti e connessioni esterne da trasformare in un sotto-circuito, quindi selezionare **Crea sotto-circuito** dal menu di scelta rapida. Vi sarà chiesto di digitare un nome per il sotto-circuito. Una volta creato, il nome apparirà nel selettore **Componenti** nella sezione **Sotto-circuiti**. Questo può essere trattato come qualsiasi componente normale - con l'opzione aggiuntiva di rimuoverlo facendo clic con il tasto destro del mouse sull'elemento e selezionando **Rimuovi**.

Capitolo 4

FlowCode

4.1 Introduzione

FlowCode permette una costruzione molto semplice e veloce di un programma PIC. Dopo che l'utente ha costruito un diagramma di flusso con le parti del programma disponibili, KTechlab può convertire il diagramma di flusso in un certo numero di formati. Per convertire in esadecimale, per esempio, si verifica la seguente catena di conversioni:

1. Il FlowCode viene convertito in Microbe; un linguaggio ad alto livello il cui compilatore è distribuito insieme a KTechlab.
2. L'eseguibile **microbe** compila quindi il file Microbe in assembly PIC.
3. Infine **gpasm** prende il file assembly PIC e genera l'esadecimale per il programma.

Naturalmente se non hai installato gputils, che è distribuito insieme a **gpasm**, non può essere eseguito l'ultimo passo.

4.2 Creare un programma

Ogni programma FlowCode ha bisogno di un punto d'inizio unico (è la posizione da cui il tuo programma sarà eseguito all'avvio del PIC). Per definire questo punto apri la barra Flow Part sulla sinistra e trascina la parte **Inizio**. KTechlab ti permetterà di usarne solamente una.

Puoi quindi costruire il tuo programma usando le parti predefinite sulla sinistra, oppure puoi inserire tu stesso il codice (in formato assembly o Microbe) attraverso la parte **Incorpora**. Il flusso del programma viene controllato dalle connessioni tra le FlowPart; Sezione 3.2 offre maggiori dettagli sulla creazione di queste.

FlowCode impone delle limitazioni in aggiunta a quelle dei circuiti su che cosa può essere connesso; per esempio, ciascuna FlowPart può avere solamente una connessione di uscita. Ulteriori limitazioni sono descritte in Sezione 4.4.

4.3 Impostazioni PIC

Quando crei un nuovo documento FlowCode noterai, nell'angolo in alto a sinistra dell'area di lavoro, un'immagine del PIC che stai usando. Questa rappresenta le impostazioni iniziali del PIC.

Ciascun piedino mostrato sulla figura del PIC mostra il tipo iniziale del piedino (di ingresso o di uscita), ed il suo stato iniziale (alto o basso). Puoi cambiarli entrambi: trascina il piedino per impostare il tipo, e fai clic sui di esso per cambiarne il suo stato.

La finestra **Impostazioni**, invocata facendo clic sul pulsante **Impostazioni**, ti permette anche di modificare il tipo iniziale di piedino e lo stato - in questo caso modificando i valori binari scritti nei registri PORT e TRIS. Oltre alle impostazioni dei piedini, la finestra consente la modifica dei valori iniziali delle variabili nel programma PIC.

In basso c'è un elenco delle mappature di piedini attualmente definite, oltre ai pulsanti per manipolarle. Le mappature di piedini vengono usate per specificare come sono connessi al PIC un sette segmenti o un tastierino. Per usare le parti FlowCode del **Sette Segmenti** o del **Tastierino** avrai prima bisogno di definire la mappatura dei piedini.

4.4 Annidare il FlowCode

Molte FlowPart, come le subroutine ed i cicli, possono contenere del codice. Dopo aver creato tale contenitore, le FlowPart possono essere aggiunte mediante trascinamento nel contenitore; questo sarà evidenziato per indicare che diventerà il nuovo genitore della FlowPart.

Il contenitore si prende la responsabilità delle FlowPart annidate al suo interno. Se non si è fatto clic sul pulsante di espansione tutte le FlowPart saranno nascoste, e allo stesso modo i contenuti saranno mostrati quando si farà nuovamente clic su tale pulsante. Le connessioni non possono essere fatte tra FlowPart che si trovano in contenitori diversi, inoltre il contenuto di un contenitore verrà spostato insieme al contenitore.

Capitolo 5

Microbe

5.1 Introduzione e sintassi generale

Microbe compila i programmi scritti in un linguaggio personalizzato per i PIC, come programma che lavora insieme a KTechlab. La sintassi è stata progettata per adattarsi ai programmi in FlowCode. Per avviare **microbe** da riga di comando la sintassi è:

```
microbe [opzioni] [input.microbe] [output.asm]
```

, mentre le opzioni sono:

- `--show-source` - mette ogni riga di codice sorgente di Microbe come commento nell'output assembly, prima delle istruzioni assembly stesse della riga.
- `--no-optimize` - previene l'ottimizzazione delle istruzioni generate da quel sorgente. L'ottimizzazione è in genere sicura, perciò quest'opzione viene usata principalmente durante il debug.

Il file `.microbe` in ingresso deve identificare il PIC a cui è destinato inserendo il nome di tale PIC all'inizio del file `.microbe`; ad esempio, il nome di un PIC16F84 è «P16F84».

Example 5.1 Un programma Microbe semplice ma completo

```
P16F84
a = 0
repeat
{
    PORTA = a
    a = a + 1
}
until a == 5
end
```

5.1.1 Convenzioni sui nomi

Per i nomi delle variabili e per le etichette si applicano le seguenti regole:

- Possono contenere solo caratteri alfanumerici [a..z][A..Z][0..9] e il trattino basso “_”.
- Distinguono le maiuscole.
- Non possono iniziare con un numero.
- Non dovrebbero iniziare con ‘__’ (doppio trattino basso), in quanto questo è riservato per il compilatore.

5.1.2 Convenzioni sulle parentesi

Le parentesi graffe, {}, indicano l’inizio e la fine di un blocco di codice. Possono comparire in qualsiasi punto prima dell’inizio e dopo la fine del blocco di codice. Esempi accettabili di blocchi di codice:

```
istruzione1 {  
    del codice  
}
```

```
istruzione2 {  
    altro codice }
```

```
istruzione3  
{  
    altro codice  
}
```

```
istruzione5 {  
    blocco di codice  
} istruzione6
```

5.1.3 Commentare

I commenti sono simili a quelli del C, cioè // per commentare il resto della riga, mentre /* e */ denotano un commento multi-riga.

```
// Questo è un commento  
x = 2  
/* Come lo è questo  
commento multi-riga */
```

5.1.4 Struttura del programma

L’identificativo del PIC deve essere inserito all’inizio del programma, mentre la fine del programma principale è denotata con ‘end’. Le subroutine devono essere messe dopo l’end’.

5.1.5 Subroutine

Una subroutine può essere chiamata in qualsiasi parte nel codice. Sintassi:

```
sub NomeSub  
{  
    // Codice...  
}
```

La subroutine viene chiamata con ‘call *NomeSub*’.

5.2 Riferimenti al linguaggio Microbe

5.2.1 if

Diramazione condizionale. Sintassi:

```
if [espressione] then [istruzione]
```

o

```
if [espressione] then
{
    [blocco di istruzioni]
}
```

Lo stesso per else:

```
else [istruzione]
```

oppure

```
else
{
    [blocco di istruzioni]
}
```

Example 5.2 if

```
if porta.0 is high then
{
    delay 200
}
else
{
    delay 300
}
```

5.2.2 alias

Chiama una stringa in un altro modo. Sintassi:

```
alias [da] [a]
```

5.2.3 repeat

Esegue ripetutamente il blocco di istruzioni finché l'espressione viene valutata come vera (true). Questa valutazione viene eseguita prima del blocco di istruzioni, quindi questo sarà sempre eseguito almeno una volta. Sintassi:

```
repeat
{
    [blocco di istruzioni]
}
until [espressione]
```

5.2.4 while

Simile a repeat, esegue ripetutamente il blocco di istruzioni. Tuttavia l'espressione viene valutata prima dell'esecuzione, non dopo. Quindi, se questa viene valutata come falsa (false) al primo passaggio, il blocco di istruzioni non viene proprio eseguito. Sintassi:

```
while [espressione]
{
    [blocco di istruzioni]
}
```

5.2.5 goto

Provoca l'esecuzione del codice dopo l'etichetta specificata e prosegue all'istruzione successiva. Sintassi del goto:

```
goto [nomeetichetta]
```

. Sintassi dell'etichetta:

```
labelname:
```

. Viene spesso considerata una buona pratica quella di evitare l'uso del goto: l'utilizzo di istruzioni di controllo e di subroutine farà infatti sì che il programma risulti molto più leggibile.

Example 5.3 goto

```
goto MiaEtichetta

...

[MiaEtichetta]:
// il codice continuerà in questo punto
```

5.2.6 call

Chiama una subroutine. Sintassi:

```
call [NomeSub]
```

, dove *NomeSub* è il nome della subroutine da chiamare.

5.2.7 delay

Fa sì che l'esecuzione del codice si fermi per un certo periodo di tempo. L'intervallo è in millisecondi. Sintassi:

```
delay [intervallo]
```

NOTA

Al momento Microbe presume che il PIC stia operando alla frequenza di 4MHz, cioè che ciascuna istruzione impieghi 1 microsecondo per essere eseguita. Se non fosse questo il caso è necessario regolare opportunamente l'intervallo.

5.2.8 sevenseg

Viene usato per definire la mappatura dei piedini di un display a sette segmenti (a catodo comune) che è connesso al PIC. Sintassi:

```
sevenseg [nome] [a] [b] [c] [d] [e] [f] [g]
```

, dove [a]...[g] sono i piedini del PIC ai quali sono collegati i rispettivi segmenti del display a sette segmenti. I piedini possono essere scritti anche come PORTX.N o RXN.

Per visualizzare un numero sui sette segmenti la mappatura dei piedini viene considerata come una variabile a sola scrittura.

Example 5.4 Definizione su un sette segmenti e scrittura

```
sevenseg seg1 RB0 RB1 RB2 RB3 RB4 RB5 RB6
seg1 = x + 2
```

5.2.9 keypad

Viene usato per definire la mappatura dei piedini di un tastierino connesso al PIC. Sintassi:

```
keypad [nome] [riga 1] ... [riga 4] [colonna 1] ... [colonna n]
```

, dove [riga 1] ... [riga 4] e [colonna 1] ... [colonna n] sono i piedini del PIC ai quali sono collegate le rispettive righe e colonne del tastierino (al momento il numero delle righe non è modificabile). Vedi Sezione 5.2.8 (sopra) per ulteriori informazioni sulla mappatura dei piedini.

Le colonne del tastierino devono essere collegate a terra tramite una resistenza da 100k. I piedini delle righe devono essere configurati come uscite, mentre quelli delle colonne come ingressi. Una volta che viene definito il tastierino, questo sarà trattato come una variabile a sola lettura.

Example 5.5 Definizione di un tastierino e lettura

```
keypad keypad1 RB0 RB1 RB2 RB3 RB4 RB5 RB6
x = keypad1
```

I valori predefiniti restituiti dal tastierino sono:

- Il valore del numero se è un tasto numerico (da 1 a 3 nella riga in alto; da A a D esadecimali nella quarta colonna, e così via per ogni colonna aggiuntiva).
- 253 per il tasto nella riga 4, colonna 1.
- 254 per il tasto nella riga 4, colonna 3.

Questi valori possono essere ridefiniti usando i comandi di alias, dove il nome del tasto alla riga x, colonna y (le righe e le colonne iniziano da 1) è Keypad_x_y. Per esempio, per dare al tasto con l'asterisco di un tastierino 4x3 il valore zero possono essere usati i seguenti alias:

Example 5.6 Assegnazione di un valore al tasto di un tastierino

```
alias Keypad_4_1 0
```

5.3 Ingresso/uscita del PIC

5.3.1 Direzione della porta

La direzione della porta è impostata assegnando un valore a TRIS*, dove * è la lettera della porta. Per esempio:

Example 5.7 Impostare la direzione della porta

```
TRISB = b'01111001'
```

Questo imposta i piedini RB1, RB2 e RB7 nella PORTB come uscite, e gli altri come ingressi. In questo esempio, b'01111001' è una rappresentazione binaria del tipo di uscita: l'1 alla destra rappresenta un'uscita su RB0, mentre quello alla sinistra un ingresso in RB7.

5.3.2 Porte di ingresso/uscita

La porta può essere trattata come una variabile. Per esempio:

Example 5.8 Scrivere su una porta

```
x = PORTA
```

Qui sopra viene assegnato il valore di PORTA alla variabile x.

5.3.3 Piedini di ingresso/uscita

Ciascun piedino su una porta è ottenuto antepoendo il numero del piedino al nome della porta; ad esempio il Pin 2 (iniziando dal Pin 0) su PORTA è conosciuto come *PORTA.0*. La sintassi per impostare uno stato di un piedino è:

```
PORTX.N = STATE
```

dove *STATE* può essere *alto* o *basso*, mentre la sintassi per testare lo stato di un piedino è:

```
if PORTX.N is STATE then
```

. Combinando questi esempi abbiamo:

Example 5.9 Impostare e testare lo stato del piedino

```
TRISA = 0
TRISB = 255
if PORTA.3 is high then
{
    PORTB.5 = low
}
else
{
    PORTB = PORTA + 15
}
```

5.4 Variabili

Tutte le variabili sono interi senza segno a 8 bit, compresi nell'intervallo che va da 0 a 255. Microbe supporta le tipiche operazioni unarie (cioè quelle che agiscono su una variabile) e quelle binarie (su due) che sono supportate dal PIC, più le divisioni e le moltiplicazioni.

5.4.1 Operazioni unarie

- *rotateleft x* - ruota a sinistra la variabile x attraverso il carry.
- *rotateright x* - ruota a destra la variabile x attraverso il carry.
- *increment x* - incrementa la variabile x. Se x vale 255 allora sarà arrotondata a 0.
- *decrement x* - decrementa la variabile x. Se x vale 0 allora sarà arrotondata a 255.

5.4.2 Aritmetica

Operazioni supportate:

- *Addizione*: $x + y$
- *Sottrazione*: $x - y$
- *Moltiplicazione*: $x * y$
- *Divisione*: x / y
- *XOR binario*: $x \text{ XOR } y$
- *AND binario*: $x \text{ AND } y$
- *OR binario*: $x \text{ OR } y$

5.4.3 Confronto

Operazioni supportate:

- *Uguale*: $x == y$
- *Non uguale*: $x != y$
- *È maggiore di*: $x > y$
- *È minore di*: $x < y$
- *È maggiore o uguale a*: $x >= y$
- *È minore o uguale a*: $x <= y$

Per esempio:

Example 5.10 Confronto

```
if PORTA
>= 5 then
{
    ...
}
```

Capitolo 6

Debug

6.1 Avviare il debug

Il supporto del debug è fornito per Assembly, SDCC e Microbe, quando sono aperti come un documento di testo. Da qui, lo stepping è controllato tramite il menu **Debug**. Esistono due metodi di avviare il debugger.

Se il programma PIC è già in esecuzione in un circuito, facendo doppio clic sul componente PIC si aprirà il programma. Per i programmi PIC assembly, il debugger per quel documento di testo sarà collegato al componente PIC. In questo caso, il menu di debug non può arrestare il programma PIC, poiché appartiene al componente PIC.

Se il file assembly è già aperto, è possibile eseguire il debugger tramite il menu **Debug**. Dopo aver compilato il programma, il debugger sarà pronto, con il programma PIC in pausa alla prima istruzione. Si noti che quando si esegue il debug di linguaggi di alto livello, il punto di esecuzione attuale non sarà mostrato se non vi è alcuna linea corrispondente alla prima istruzione assembly da eseguire. In questo caso, facendo clic su **Avanti** si porterà il punto di esecuzione alla prima riga del programma.

6.2 Controllare il debugger

Il debugger può essere in una delle due modalità: esecuzione e stepping. Durante l'esecuzione, il programma PIC sarà simulato in tempo reale. Per consentire lo stepping, il programma PIC deve essere messo in pausa - facendo clic su **Interrompi** nel menu **Debug** o facendo clic sul pulsante Pausa sul componente PIC.

Nella modalità di stepping, una freccia verde a margine del documento di testo indica la riga successiva da eseguire (familiare agli utenti di KDevelop). Può essere utile attivare il bordo dell'icona tramite il menu **Visualizza** (può essere attivato permanentemente tramite la finestra **Impostazioni dell'editor**).

Esistono tre tipi di stepping:

- **Step** - Questo esegue l'istruzione attuale. La freccia verde è spostata sulla riga successiva da eseguire.
- **Step Over** - Se l'istruzione successiva da eseguire è una chiamata o simile, allora questa «passa sulla» chiamata, tornando alla modalità di stepping dopo che la chiamata è restituita. Altrimenti, lo step over di un'istruzione si comporta in modo identico a step. Per dirlo tecnicamente: il livello iniziale dello stack viene registrato e l'esecuzione del programma viene messa in pausa quando il livello dello stack torna al livello iniziale.

- **Step Out** - Se l'esecuzione attuale si trova all'interno di una chiamata o simile, ciò attenderà fino alla restituzione della chiamata. Analogamente allo step over, questo equivale ad attendere fino a quando il livello dello stack non torna a uno in meno del livello iniziale, se il livello iniziale è maggiore di zero.

I punti di interruzione consentono di mettere in pausa l'esecuzione quando il programma PIC raggiunge una determinata istruzione. Per attivare o disattivare un punto di interruzione sulla riga contenente il cursore, utilizzare il menu **Debug** o fare clic sul bordo dell'icona del documento di testo.

La barra laterale **Visore dei simboli** sulla destra mostra i valori dei registri delle funzioni speciali. Per scoprire il valore di una variabile nei registri di uso generico, è possibile passare il puntatore del mouse sul nome della variabile in un'istruzione che opera su quel registro. Notare che la selezione di base nel **Visore dei simboli** controlla anche come viene visualizzato il valore quando si passa sopra una variabile.

Capitolo 7

FAQ

1. *KTechlab usa molta CPU*

Ci sono molte possibili cause: la simulazione di circuiti contenenti sia componenti reattivi che non lineari (come i condensatori ed i transistor) impiega molto tempo di CPU. Puoi mettere in pausa la simulazione e poi riattivarla dal menu **Strumenti**.

Il disegno dell'area di lavoro (in particolare il ridisegno di molte barre di tensione che vengono aggiornate rapidamente sui piedini) richiede allo stesso modo un elevato uso di CPU. Puoi ridurre la frequenza di aggiornamento o spegnere le barre di tensione dalla finestra **Impostazioni**. La frequenza di aggiornamento dell'**Oscilloscopio** può essere ridotta anche facendo clic con il tasto destro sul display.

Nota che la prossima versione di KTechlab potrebbe essere molto più veloce, sia nel visualizzare l'area di lavoro che nella simulazione di componenti reattivi e non lineari.