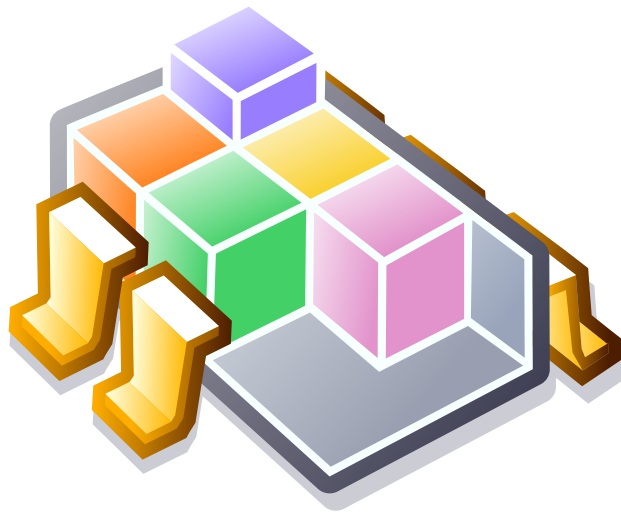


El manual del KTechlab

David Saxton
Daniel Clarke



El manual del KTechlab

Índex

1	Una volta ràpida	6
1.1	Introducció	6
1.2	Documents	6
1.3	Dibuixar	7
2	Programes PIC	8
2.1	Manipular	8
2.2	Transmetre	8
3	Circuits	10
3.1	Posicionar els components	10
3.2	Connectar els components	10
3.3	Atributs del component	11
3.4	Simulació	11
3.5	Oscil·loscopi	11
3.6	Subcircuits	12
4	FlowCode	13
4.1	Introducció	13
4.2	Crear un programa	13
4.3	Configuració del PIC	13
4.4	FlowCode imbricat	14
5	Microbe	15
5.1	Introducció i sintaxi general	15
5.1.1	Convencions de la nomenclatura	15
5.1.2	Convencions de les claus	16
5.1.3	Comentar	16
5.1.4	Estructura del programa	16
5.1.5	Subrutines	16
5.2	Referència del llenguatge Microbe	17
5.2.1	if	17
5.2.2	alias	17
5.2.3	repeat	17

El manual del 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	E/S del PIC	20
5.3.1	Direcció del port	20
5.3.2	E/S del port	20
5.3.3	E/S dels pins	20
5.4	Variables	21
5.4.1	Operacions unàries	21
5.4.2	Aritmètica	21
5.4.3	Comparació	21
6	Depuració	23
6.1	Iniciar el depurador	23
6.2	Controlar el depurador	23
7	PMF	25

Resum

El KTechlab és un IDE per a microcontroladors i electrònica.

Capítol 1

Una volta ràpida

1.1 Introducció

El KTechlab és un IDE per a circuits electrònics i microcontroladors. Pot realitzar la simulació d'una varietat de components (lògics, integrats, lineals, no lineals i reactius) i depuració de microcontroladors PIC mitjançant el «gpsim», i ve amb els seus propis llenguatges d'alt nivell estretament vinculats i complementaris: FlowCode i Microbe.

Ha estat dissenyat per a ser el més fàcil d'emprar i no intrusiu possible. Tots els components i les FlowPart tenen ajuda sensible al context, i la simulació de l'electrònica és tan senzilla com arrossegar els components a l'àrea de treball i crear connectors que s'encaminen automàticament entre els seus pins. El FlowCode permet als usuaris nous en el PIC, que creïn instantàniament els seus propis programes, mentre que la simulació electrònica permet recórrer el programa en assemblador del PIC dins d'un circuit.

1.2 Documents

Per a començar en el KTechlab, haureu de crear un document nou, el tipus dependrà de la vostra tasca:

- Document de FlowCode: construeix un programa PIC mitjançant diagrames de flux.
- Document de Circuit: simula circuits electrònics i microcontroladors.
- Document de Microbe: llenguatge d'alt nivell per als PIC, també utilitzat pel FlowCode per a generar en assemblador.
- Document en assemblador: comença a escriure un programa en assemblador del PIC.

El KTechlab utilitza un model Document-Vista, en el que la lògica del document està completament separada de les vistes obertes del document. Això permet diverses vistes del mateix fitxer.

En crear un document nou, la vista es crearà en una pestanya separada. Cada pestanya pot admetre qualsevol nombre de vistes, en mosaic i en qualsevol patró arbitrari. Això permet, per exemple, simular un programa PIC en circuit, mentre recorre el programa en un document en assemblador en la mateixa pestanya.

El contingut de les pestanyes es pot duplicar arrossegant la pestanya a una àrea buida a la barra de pestanyes. Es poden inserir en una pestanya existent arrossegant a aquesta pestanya.

Les instruccions detallades sobre els documents anteriors es poden trobar en els seus propis capítols respectius.

1.3 Dibuixar

En els documents de Circuit i FlowCode, hi ha disponibles diverses eines de dibuix, inclòs el text. Aquestes estan disponibles fent clic sobre la icona de pinzell que hi ha a la barra d'eines. Per a dibuixar, arrossegueu el ratolí per a formar una forma o una línia apropiada per a l'eina de dibuix en ús.

Quan se selecciona un dibuix, es pot canviar de mida arrossegant les seves nanses. Si manteniu premuda la tecla **Maj** mentre arrossegueu, el controlador s'ajustarà a la quadrícula subjacent. Cada eina té opcions bàsiques accessibles des de la barra d'eines, com els colors. També hi ha opcions més avançades a la barra lateral **Editor d'elements**, com els estils de línia i de topall.

Capítol 2

Programes PIC

2.1 Manipular

Quan creeu un FlowCode o un document de text, notareu un menú desplegable a la barra d'eines amb una icona de coet. Des d'allí, podreu manipular el programa PIC. Canviant-lo a formes diferents.

- **Converteix a Microbe:** s'utilitza només en documents de FlowCode. Això s'explica en més detall en el capítol 4.
- **Converteix en assemblador:** es pot utilitzar en quatre contextos. Quan està obert un document de FlowCode, generarà el FlowCode com instruccions en assemblador. Quan està obert un document de Microbe, invocarà el programa **microbe** distribuït amb el KTechlab per a compilar el programa. De manera similar, si està obert un programa en C, intentarà compilar mitjançant SDCC. Quan està obert un document de text que conté hexadecimal PIC, invocarà el **gpdasm** per a desassemblar l'hexadecimal.
- **Converteix a hexadecimal:** això també es pot utilitzar en quatre contextos. Igual com **Converteix a assemblador**, això es pot utilitzar amb documents de FlowCode, Microbe i C. També s'habilitarà quan s'obri un document en assemblador mitjançant el **gpasm**.
- **Puja al PIC:** això converteix en assemblador el programa PIC que s'està editant i el puja utilitzant el programador que ha seleccionat l'usuari.

Cap d'aquestes accions requereix que es desi el document actual -el qual és molt útil quan es requereix un programa ràpid-. Per als objectius que no són PIC, el diàleg **Sortida** que s'invoca en fer clic sobre una d'aquestes accions pot generar el resultat (sempre text en els tres casos anteriors) en un document nou o en un fitxer. Si la sortida es desa en un fitxer, també proporcionarà opcions per a pujar el fitxer després de crear-lo i afegir el fitxer acabat de crear al projecte obert (si n'hi ha un obert).

Recordeu que podeu fer que el KTechlab utilitzi sempre la mateixa vista per a mostrar el contingut generat seleccionant l'opció sota la configuració **General**.

2.2 Transmetre

El KTechlab utilitza programadors de tercers per a pujar els programes als PIC. Venen predefinitos una varietat de programadors habituals. Se'n poden afegir altres mitjançant el diàleg **Arranjament**.

La llista de ports s'obté d'escanejar els ports en sèrie i paral·lel que siguin de lectura i escriptura. Els ports en sèrie se cerquen a:

El manual del KTechlab

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

Els ports en paral·lel se cerquen a:

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

Capítol 3

Circuits

3.1 Posicionar els components

A l'esquerra, trobareu la pestanya **Components**.

Arrossegant un component des de la barra lateral cap al circuit el col·locarà a sota el cursor del ratolí. De manera alternativa, podeu fer doble clic sobre un element a la barra lateral **Components** per a afegir-lo repetidament al circuit. En aquest mode, es col·locarà repetidament una còpia del component seleccionat sobre el clic esquerre fins que es premi **Esc** o es faci clic dret.

Per a reposicionar un component, feu clic esquerre i arrossegueu. El trobareu ajustat en la quadrícula subjacent. Si arrossegueu el component a fora de les vores dreta o inferior de l'àrea de treball, l'àrea de treball canviarà de mida per a adaptar-s'hi.

Tots els components tenen una noció de l'orientació; 0, 90, 180 i 270 graus. Aquells que no siguin simètrics vora un eix també podran capgirar-se. Per a girar una selecció de components, feu-hi clic dret i seleccioneu des del menú **Orientació**, o feu clic sobre els botons de gir que hi ha a la barra d'eines. També es pot accedir a aquest últim prement les tecles [i] (familiar per als usuaris de l'Inkscape). La barra lateral **Element** (a la dreta) proporciona un poderós mètode per a configurar l'orientació proporcionant vistes prèvies dels components. També és possible girar els components mitjançant la barra lateral **Element**.

3.2 Connectar els components

Hi ha dos modes per a crear les connexions (cables): automàtic i manual. Aquests modes se seleccionen mitjançant el menú desplegable **Mode d'encaminament de les connexions** que hi ha a la barra d'eines. Experimenteu amb ambdós: l'encaminament automàtic sol ser millor per a circuits petits, mentre que els circuits més complexos poden necessitar encaminament manual.

En el mode automàtic, creeu una connexió arrossegant des d'un pin de component o una connexió existent i deixeu anar el ratolí sobre el pin o connexió desitjada. Veureu que la línia recta que es dibuixa es torna taronja quan es crea una connexió vàlida en deixar anar el ratolí. Si la línia que esteu dibuixant és negra, és perquè no hi ha res a sota del cursor del ratolí o perquè esteu intentant connectar dos elements que ja estan connectats. En crear un diagrama de flux, els criteris per a una connexió vàlida són més complexos, però arribarem a això més endavant.

La millor manera de familiaritzar-se amb l'encaminament manual de les connexions és experimentant. Feu clic sobre el pin o connexió d'inici i després esteneu el protoconnector movent el ratolí lluny d'on heu fet clic. Per a posicionar-lo en una cantonada, feu-hi clic esquerre. Per a cancel·lar el dibuix de la connexió, premeu **Esc** o feu-hi clic dret.

El KTechlab fa tot el possible per a mantenir els camins que prenen les connexions. No obstant això, si en arrossegar un component, els punts finals d'una connexió es mouen entre si, el KTechlab es veurà obligat a tornar a dibuixar la connexió mitjançant l'encaminament automàtic. Abans de moure un component, veureu quins connectors s'han de tornar a encaminar, ja que es tornaran grisos en fer-hi clic.

Per a eliminar una connexió existent, seleccioneu-la dibuixant un petit rectangle de selecció sobre part del connector i premeu **Supr.**

3.3 Atributs del component

La majoria dels components tindran atributs editables, com la resistència per a les resistències. De manera predeterminada, podreu editar els atributs senzills a la barra d'eines, quan se selecciona un grup del mateix tipus de components. Si la selecció conté una mescla de diferents tipus de components (com resistències i condensadors), no es mostraran atributs per a editar.

Alguns components tenen atributs més avançats als quals no es pot accedir a través de la barra d'eines. Aquests es troben a la barra lateral **Element** que hi ha a la dreta. El díode, per exemple, té una varietat de característiques de comportament que podreu editar aquí.

Hi ha un tipus d'atribut que no es pot editar ni a la barra d'eines ni a la barra lateral **Element** -text de múltiples línies-. En fer doble clic sobre l'element, apareixerà un diàleg on es podrà introduir el text.

3.4 Simulació

De manera predeterminada, la simulació s'executarà quan creeu un circuit nou. L'estat de la simulació es mostra a la part inferior dreta d'una vista de circuit i es pot canviar mitjançant el menú **Eines**. En primer lloc, una petita explicació sobre com funciona el simulador. Això hauria de permetre aprofitar-lo al màxim.

Quan es crea o modifica un circuit, les àrees afectades es divideixen en grups de pins i connexions que es poden considerar independents. Després, cada grup se simula com una entitat separada (tot i que encara interactua a través dels components), i la simulació proporcionada dependrà de la complexitat del grup. Els grups complexos, com els que involucren components no lineals com els LED, són lents de simular. Els grups que contenen només pins lògics, dels quals només un controla el valor en aquests pins, són els més ràpids de simular.

El resultat de la simulació es proporciona a través de diversos mitjans gràfics.

Els pins en els components, mostraran barres laterals de voltatge. Aquests són de color taronja per al voltatge positiu i blau per al voltatge negatiu. La seva longitud dependrà del nivell de voltatge i la seva amplada de la quantitat de corrent que flueix a través del pin. Aquests es poden desactivar a la pàgina **General** del diàleg **Arranjament**.

En passar el ratolí per sobre d'un pin o connexió, es mostrarà un petit consell d'eina que mostri el voltatge i el corrent en aquest punt del circuit. Diversos components també proporcionen retroalimentació gràfica, per exemple, els LED i voltímetres o amperímetres.

Finalment, hi ha l'oscil·loscopi, el qual s'analitza en la següent secció.

3.5 Oscil·loscopi

L'oscil·loscopi pot enregistrar les dades de lògica, voltatge i corrent. La prova lògica està optimitzada per a emmagatzemar mostres booleanes, per la qual cosa s'ha d'utilitzar en lloc de la prova de voltatge quan es mesura la lògica.

Per a recopilar les dades, creeu un component de prova nou i connecteu-lo amb un punt apropiat en el circuit. Veureu la sortida dibuixada immediatament en l'oscil·loscopi. Afegir més sondes aixafarà les sortides una a una al costat de l'altra. Podeu reposicionar-les arrossegant les fletxes que hi ha a l'esquerra de la vista de l'oscil·loscopi i canviar els seus colors mitjançant els atributs de la prova.

Per a les proves de voltatge i corrent, l'interval de valors d'entrada es pot ajustar a la barra lateral **Editor d'elements** que hi ha a la dreta.

El zoom es controla mitjançant un control lliscant. L'escala és logarítmica. Per a cada pocs píxels que recorri el control lliscant, el factor de zoom es multiplicarà per una constant. El KTechlab simula la lògica amb una precisió màxima d'1 microsegon, i amb el nivell de zoom al màxim, un microsegon està representat per 8 píxels.

Quan la barra de desplaçament s'arrossega fins al final, romandrà allà a mesura que es registren dades noves. En cas contrari, la posició de la barra de desplaçament romandrà fixa en el temps. La vista de l'oscil·loscopi també es pot moure cap endavant i cap enrere fent-hi clic esquerre i arrossegant la vista. A causa de les limitacions del sistema subjacent de ginys, el desplaçament serà molt granular amb el zoom al màxim.

En fer clic dret sobre la vista de l'oscil·loscopi, apareixerà un menú on podreu controlar el nombre de vegades que s'actualitza la vista de l'oscil·loscopi. Això permet una vista més suau o reduir l'ús de la CPU.

3.6 Subcircuits

Els subcircuits ofereixen una forma reutilitzable i ordenada d'utilitzar un circuit, quan només esteu interessat a interactuar amb connexions externes amb el circuit. El subcircuit es crearà com un CI (circuit intern), amb els pins actuant com a interacció amb el CI.

Primer, per a crear un subcircuit, s'ha de construir el circuit que s'utilitzarà com a plantilla. Els punts d'interacció es defineixen mitjançant components de **Connexió externa**. Aquests han d'estar connectats i posicionats on voleu que es posi el pin al CI del subcircuit.

A continuació, seleccioneu el grup de components i connexions externes que es convertiran en un subcircuit, i seleccioneu **Crea un subcircuit** des del menú contextual. Se us oferirà introduir un nom per al subcircuit. Un cop creat, el nom apareixerà al selector de **Components** sota de la selecció **Subcircuits**. Això es pot tractar com qualsevol component normal, amb l'opció addicional d'eliminar fent clic dret sobre l'element i seleccionant **Elimina**.

Capítol 4

FlowCode

4.1 Introducció

El FlowCode permet una construcció molt ràpida i senzilla d'un programa PIC. Una vegada que l'usuari ha construït un diagrama de flux a partir de les parts disponibles del programa, el KTechlab pot convertir el diagrama de flux en diversos formats. Per a generar hexadecimal, per exemple, es porta a terme la següent cadena de conversions:

1. El FlowCode es converteix a Microbe, un llenguatge d'alt nivell en el qual el compilador es distribueix amb el KTechlab.
2. L'executable **microbe** després compila el fitxer de Microbe en ensamblador PIC.
3. Finalment, el **gpasm** pren el fitxer en ensamblador PIC i genera l'hexadecimal per al programa.

Per descomptat, si no teniu instal·lades les gputils -amb les quals es distribueix el **gpasm**- no es podrà realitzar l'últim pas.

4.2 Crear un programa

Cada programa de FlowCode necessita un punt de partida únic: aquest és el lloc des d'on s'executarà el vostre programa en iniciar el PIC. Per a definir aquest punt, obriu la barra lateral FlowParts que hi ha a l'esquerra i arrossegueu-lo a través de la part **Inici**. El KTechlab només us permetrà utilitzar-ne un.

Després podreu construir el vostre programa utilitzant les parts predefinides que hi ha a l'esquerra, o inserir el codi propi (en el format en ensamblador o Microbe) a través de la part **Incrusta**. El flux del programa es controla a través de les connexions entre les FlowPart, la Secció 3.2 ofereix més detalls sobre la creació de les connexions.

El FlowCode imposa limitacions a més de les dels circuits sobre els quals es pot connectar. Per exemple, cada FlowPart només pot tenir una connexió de sortida. Les limitacions addicionals es descriuen en la Secció 4.4.

4.3 Configuració del PIC

Quan creeu un document nou de FlowCode, veureu una imatge del PIC que esteu utilitzant a la cantonada superior esquerra de l'àrea de treball. Això representa la configuració inicial del PIC.

Cada pin que es mostra a la imatge del PIC mostra el tipus inicial del pin (entrada o sortida) i el seu estat inicial (alt o baix). Podeu canviar-los arrossegant el pin per a establir el tipus i fent-hi clic per a canviar el seu estat.

El diàleg **Arranjament**, el qual s'invoca fent clic al botó **Arranjament**, també permet editar els tipus i estats dels pins inicials, en aquest cas, editant els valors binaris escrits en els registres PORT i TRIS. No obstant això, a més de la configuració dels pins, el diàleg permet editar els valors inicials de les variables en el programa PIC.

A la part inferior, hi ha una llista dels mapes de pins que hi ha definits actualment, així com els botons per a manipular-los. Els mapes de pins s'utilitzen per a especificar com es connecten set segments o un teclat numèric amb un PIC. Per a utilitzar les parts de FlowCode **Set segments** o **Teclat numèric**, primer haureu de definir aquí un mapa de pins.

4.4 FlowCode imbricat

Moltes FlowPart, com subrutines i bucles, poden contenir codi propi. Després de crear un contenidor, les FlowPart es poden afegir arrossegant o deixant-les anar al contenidor. El contenidor es ressaltarà per a indicar que es convertirà en el nou pare de la FlowPart.

El contenidor assumeix la responsabilitat de les FlowPart imbricades a l'interior. Si no es fa clic al botó expandeix, totes les FlowPart contingudes s'ocultaran i, de la mateixa manera, el contingut es mostrarà quan es faci clic de nou al botó expandeix. No es poden fer connexions entre les FlowPart en contenidors diferents, i el contingut d'un contenidor es mourà juntament amb el contenidor.

Capítol 5

Microbe

5.1 Introducció i sintaxi general

El **microbe** compila els programes escrits en el llenguatge personalitzat per als PIC, com a programa complementari del KTechlab. La sintaxi ha estat dissenyada per a adaptar-se a un programa de FlowCode. La sintaxi per a executar el **microbe** des de la línia d'ordres és:

- `--show-source` — Col·loca cada línia des del codi font de Microbe com a un comentari a la sortida en assemblador abans de les instruccions en assemblador per a aquesta línia.
- `--no-optimize` — Evita l'optimització de les instruccions generades des de la font. L'optimització sol ser segura, de manera que aquesta opció és útil sobretot per a depurar.

El fitxer d'entrada `.microbe` haurà d'identificar el PIC de destinació inserint el nom del PIC a la part superior del fitxer, p. ex., el nom d'un PIC16F84 és 'P16F84'.

Example 5.1 Programa de Microbe complet i senzill

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

5.1.1 Convencions de la nomenclatura

Les regles següents s'apliquen als noms i etiquetes de les variables:

- Només poden contenir caràcters alfanumèrics `[a..z][A..Z][0..9]` i el guió baix `'_'`.
- Distingeixen entre majúscules i minúscules.
- No poden començar amb un número.
- No han de començar amb `'__'` (doble subratllat), ja que està reservat perquè l'utilitzi el compilador.

5.1.2 Convencions de les claus

Les claus, {}, indiquen l'inici i el final d'un bloc de codi. Poden aparèixer en qualsevol lloc abans de l'inici i després del final del bloc de codi. Exemples de blocs de codi acceptables:

```
sentència1 {  
    algun codi  
}
```

```
sentència2 {  
    un altre codi }
```

```
sentència3  
{  
    un altre codi  
}
```

```
sentència5 {  
    bloc_de_codi  
} sentència6
```

5.1.3 Comentar

Comentar és similar a C. Les «//» comenten la resta de la línia. I les «/*» i «*/» denoten un comentari de múltiples línies.

```
// Aquest és un comentari  
x = 2  
/* Aquest és un comentari  
de múltiples línies */
```

5.1.4 Estructura del programa

L'identificador PIC s'ha d'inserir a la part superior del programa. El final del programa principal s'indica amb 'end'. Les subrutines s'han de col·locar després de 'end'.

5.1.5 Subrutines

Es pot cridar a una subrutina des de qualsevol part del codi. Sintaxi:

```
sub SubNom  
{  
    // Codi...  
}
```

La subrutina es crida amb 'call *SubNom*'.

5.2 Referència del llenguatge Microbe

5.2.1 if

Claus condicionals. Sintaxi:

```
if [expressió] then [sentència]
```

o

```
if [expressió] then  
{  
    [bloc_de_sentències]  
}
```

De manera similar per a la resta:

```
else [sentència]
```

o

```
else  
{  
    [bloc_de_sentències]  
}
```

Example 5.2 if

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

5.2.2 alias

L'àlies d'una cadena a una altra. Sintaxi:

```
alias [des de] [a]
```

5.2.3 repeat

Executa el bloc de sentències repetidament fins que l'expressió s'avalua com a certa. L'avaluació de l'expressió es realitza després del bloc de sentències, de manera que el bloc de sentències sempre s'executarà almenys una vegada. Sintaxi:

```
repeat  
{  
    [bloc_de_sentències]  
}  
until [expressió]
```

5.2.4 while

Similar a «repeat», això executarà repetidament el bloc de sentències. No obstant això, l'expressió s'avaluarà abans de l'execució, no després. Llavors, si l'expressió s'avalua com a falsa en el primer pas, el bloc de sentències no s'executarà. Sintaxi:

```
while [expressió]
{
    [bloc_de_sentències]
}
```

5.2.5 goto

Això fa que l'execució del codi continuï en la següent sentència després de l'etiqueta especificada. Sintaxi de «goto»:

```
goto [nom_etiqueta]
```

Sintaxi de l'etiqueta:

```
nom_etiqueta:
```

Sovint es considera una bona pràctica de programació evitar l'ús de «goto». L'ús de sentències de control i subrutines donarà com a resultat un programa molt més llegible.

Example 5.3 goto

```
goto Etiqueta_meva
...
[Etiqueta_meva]:
// El codi continuarà en aquest punt
```

5.2.6 call

Crida a una subrutina. Sintaxi:

```
call [SubNom]
```

on *SubNom* és el nom de la subrutina a cridar.

5.2.7 delay

Fa que s'aturi l'execució del codi durant el període de temps especificat. L'interval serà en mil·lisegons. Sintaxi:

```
delay [interval]
```

NOTA

En l'actualitat, el Microbe assumeix que el PIC està operant a una freqüència de 4 MHz, és a dir, cada instrucció triga 1 microsegon a executar-se. Si aquest no és el cas, l'interval s'haurà d'ajustar proporcionalment.

5.2.8 sevenseg

Això es fa servir per a definir l'assignació dels pins per a una pantalla de set segments (càtode habitual) connectada al PIC. Sintaxi:

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

on [a]...[g] són els pins del PIC als quals s'adjunten els segments respectius de la pantalla de set segments. Els pins es poden escriure com a PORTX.N o RXN.

Per a mostrar un número en els set segments, l'assignació dels pins es tracta com una variable de només escriptura.

Example 5.4 Definició i sortida a un set segments

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

5.2.9 keypad

Això es fa servir per a definir l'assignació dels pins per a un teclat numèric connectat al PIC. Sintaxi:

```
keypad [nom] [fila 1] ... [fila 4] [columna 1] ... [columna n]
```

on [fila 1] ... [fila 4] i [columna 1] ... [columna n] són els pins del PIC als quals s'adjunten les respectives files i columnes del teclat numèric (de moment, el nombre de files és no modificable). Per a obtenir més informació sobre les assignacions dels pins, vegeu la Secció 5.2.8 (a dalt).

Les columnes del teclat numèric s'han de baixar a terra mitjançant resistències de 100 k. Els pins de la fila s'han de configurar com a sortides i els pins de la columna com a entrades. Una vegada s'ha definit el teclat numèric, es tractarà com una variable de només lectura.

Example 5.5 Definir i llegir des d'un teclat numèric

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

De manera predeterminada, els valors retornats per un teclat numèric són:

- El valor del nombre si és una tecla numèrica (1 a 3 a la fila superior, hexadecimal des de l'A fins a la D a la quarta columna i continuant per a cada columna addicional).
- 253 per a la tecla a la fila 4, columna 1.
- 254 per a la tecla a la fila 4, columna 3.

Aquests valors es poden redefinir utilitzant l'ordre «alias», on el nom de la tecla a la fila «x», columna «y» (files i columnes que comencen amb 1), és «Keypad_x_y». Per exemple, per a donar-li el valor zero a la tecla d'asterisc en un teclat numèric de 4x3, s'utilitzarà el següent àlies:

Example 5.6 Assignar un valor a una tecla del teclat numèric

```
alias Keypad_4_1 0
```

5.3 E/S del PIC

5.3.1 Direcció del port

La direcció del port s'estableix assignant un valor a TRIS*, on «*» és la lletra del port. Per exemple:

Example 5.7 Establint les direccions dels ports

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

L'anterior estableix els pins RB1, RB2 i RB7 a PORTB com a sortides i els altres pins en PORTB com a entrades. En aquest exemple, b'01111001' és una representació binària del tipus de sortida. L'1 de la dreta representa una sortida a RB0 i el 0 de l'esquerra representa una entrada a RB7.

5.3.2 E/S del port

El port pot tractar-se com una variable. Per exemple:

Example 5.8 Escrivint a un port

```
x = PORTA
```

Tot l'anterior assigna el valor de PORTA a la variable «x».

5.3.3 E/S dels pins

Cada pin sobre un port s'obté anteposant el número del pin al nom del port, p. ex., el pin 2 (començant pel pin 0) a PORTA es coneix com a *PORTA.0*. La sintaxi per a establir un estat del pin és:

```
PORTX.N = ESTAT
```

on *ESTAT* pot ser *high* o *low*. La sintaxi per a provar l'estat d'un pin és:

```
if PORTX.N is ESTAT then
```

Combinant aquests exemples, tenim:

Example 5.9 Establint i provant l'estat dels pins

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

5.4 Variables

Totes les variables són enters de 8 bits sense signe, el qual dona un interval des de 0 fins a 255. El Microbe admet les operacions unàries típiques (actuant sobre una variable) i les operacions binàries (actuant sobre dues variables) que són admeses pel PIC. A més, el Microbe també admet la divisió i la multiplicació.

5.4.1 Operacions unàries

- *rotateleft* x — Gira la variable « x » cap a l'esquerra a través del transport.
- *rotateright* x — Gira la variable « x » cap a la dreta a través del transport.
- *increment* x — Augmenta la variable « x ». Si « x » té un valor de 255, llavors « x » s'arrodoneix a 0.
- *decrement* x — Disminueix la variable « x ». Si « x » té un valor de 0, llavors « x » s'arrodoneix a 255.

5.4.2 Aritmètica

Operacions admeses:

- *Suma*: $x + y$
- *Resta*: $x - y$
- *Multiplicació*: $x * y$
- *Divisió*: x / y
- *XOR binari*: $x \text{ XOR } y$
- *AND binari*: $x \text{ AND } y$
- *OR binari*: $x \text{ OR } y$

5.4.3 Comparació

Operacions admeses:

- *Igual*: $x == y$
- *No és igual*: $x != y$
- *És més gran que*: $x > y$
- *És més petit que*: $x < y$
- *És més gran o igual que*: $x >= y$
- *És més petit o igual que*: $x <= y$

Per exemple:

Example 5.10 Comparació

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

Capítol 6

Depuració

6.1 Iniciar el depurador

Es proporciona suport de depuració per a Assemblador, SDCC i Microbe, quan estan oberts com a un document de text. Des d'aquí, els passos es controlen mitjançant el menú **Depura**. Hi ha dos mètodes per a iniciar el depurador.

Si el programa PIC ja s'està executant en un circuit, en fer doble clic sobre el component PIC s'obrirà el programa. Per als programes PIC en assemblador, el depurador per a aquest document de text es vincularà amb el component PIC. En aquest cas, el menú de depuració no pot aturar el programa PIC, ja que és propietat del component PIC.

Si el fitxer en assemblador ja està obert, el depurador es podrà executar mitjançant el menú **Depura**. Després de compilar el programa, el depurador estarà llest, amb el programa PIC en pausa a la primera instrucció. Recordeu que en depurar llenguatges d'alt nivell, el punt d'execució actual no es mostrarà si no hi ha una línia que es correspongui amb la primera instrucció en assemblador que s'executarà. En aquest cas, fer clic a **Següent** portarà al punt d'execució a la primera línia en el programa.

6.2 Controlar el depurador

El depurador pot estar en un de dos modes: en execució i per passos. Mentre s'executa, el programa PIC se simularà en temps real. Per a permetre per passos, el programa PIC haurà d'estar en pausa, ja sigui fent clic a l'element de menú **Depura** → **Interromp** o fent clic al botó de pausa en el component PIC.

En el mode per passos, una fletxa verda en el marge del document de text indicarà la següent línia a executar (familiar per als usuaris del KDevelop). Pot ser útil activar la vora de la icona mitjançant el menú **Visualitza** (es pot activar permanentment mitjançant el diàleg **Configuració de l'editor**).

Hi ha tres tipus de passos:

- **Pas** — Executa la instrucció actual. La fletxa verda es mou a la següent línia per a executar-se.
- **Pas sobre** — Si la següent instrucció que s'executarà és una crida o similar, això 'passarà sobre' la crida i tornarà al mode per passos quan la crida hagi retornat. En cas contrari, passar sobre una instrucció es comporta de forma idèntica a fer un pas. Per a dir-ho tècnicament, el nivell inicial de la pila es registra i l'execució del programa s'aturarà una vegada que el nivell de la pila torni al seu nivell inicial.

- **Pas cap a fora** — Si l'execució actual està dins d'una crida o similar, això esperarà fins que la crida retorni. De manera similar a fer un pas sobre, això equival a esperar fins que el nivell de la pila retorni a un menys que el nivell inicial, si el nivell inicial és més gran que zero.

Els punts d'interrupció permeten que l'execució estigui en pausa quan el programa PIC abasta una instrucció determinada. Per a canviar un punt d'interrupció en la línia que conté el cursor, utilitzeu el menú **Depura** o feu clic sobre la vora de la icona del document de text.

La barra lateral **Visualitzador de símbols** que hi ha a la dreta, mostra els valors del registre de les funcions especials. Per a trobar el valor d'una variable en el registre de propòsit general, podeu passar el ratolí per sobre del nom de la variable en una instrucció que opera en aquest registre. Recordeu que la selecció de la base en el **Visualitzador de símbols** també controla com es mostrarà el valor en passar el cursor sobre una variable.

Capítol 7

PMF

1. *El KTechlab utilitza un munt de CPU*

Hi ha diverses causes possibles. La simulació de circuits que contenen components tant reactius com no lineals (com condensadors i transistors) requereix molt de temps de CPU per a simular. Podeu fer una pausa i reprendre la simulació mitjançant el menú **Eines**.

Dibuixar l'àrea de treball (en particular, tornar a dibuixar moltes barres de voltatge que s'actualitzen ràpidament en els pins) també requereix un ús intensiu de la CPU. Podeu reduir la freqüència d'actualització o desactivar les barres de voltatge mitjançant el diàleg **Configuració**. La freqüència d'actualització de l'**Oscil·loscopi** també es pot reduir fent clic dret sobre la seva pantalla.

Recordeu que la pròxima versió principal del KTechlab podria ser molt més ràpida tant en la visualització de l'àrea de treball com en la simulació dels components reactius i no lineals.