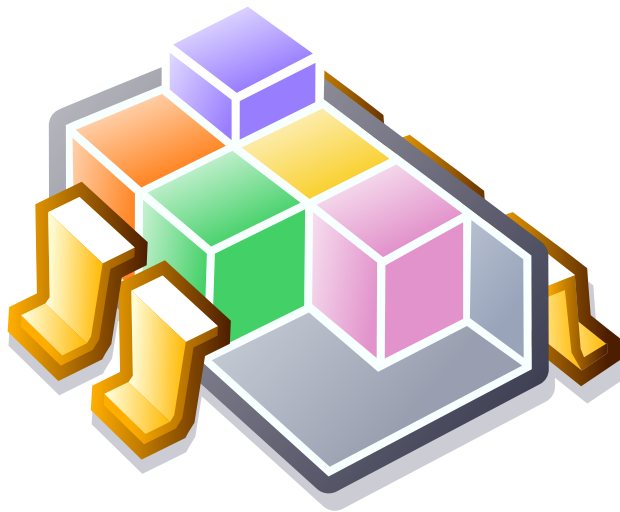


Het handboek van KTechlab

David Saxton
Daniel Clarke



Het handboek van KTechlab

Inhoudsopgave

1 Snelle rondleiding	6
1.1 Inleiding	6
1.2 Documenten	6
1.3 Tekenen	7
2 PIC programma's	8
2.1 Manipulatie	8
2.2 Uploaden	8
3 Electronische schema's	10
3.1 Componenten plaatsen	10
3.2 Componenten verbinden	10
3.3 Attributen van componenten	11
3.4 Simulatie	11
3.5 Oscilloscoop	12
3.6 Subcircuits	12
4 FlowCode	13
4.1 Inleiding	13
4.2 Een programma aanmaken	13
4.3 PIC-instellingen	13
4.4 Nestling FlowCode	14
5 Microbe	15
5.1 Inleiding en algemene syntaxis	15
5.1.1 Naamconventies	15
5.1.2 Conventies voor haakjes	16
5.1.3 Commentaar	16
5.1.4 Programma structuur	16
5.1.5 Subroutines	16
5.2 Referentie Microbe taal	17
5.2.1 if	17
5.2.2 alias	17

Het handboek van KTechlab

5.2.3	repeat	17
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	toetsenbordje	19
5.3	PIC I/O	20
5.3.1	Poort richting	20
5.3.2	Port I/O	20
5.3.3	Pin I/O	20
5.4	Variabelen	21
5.4.1	Bewerkingen met één variabele	21
5.4.2	Rekenen	21
5.4.3	Vergelijken	21
6	Debuggen	23
6.1	De debugger starten	23
6.2	De debugger besturen	23
7	FAQ	25

Samenvatting

KTechlab is een IDE (ontwikkelomgeving) voor microcontrollers en elektronica.

Hoofdstuk 1

Snelle rondleiding

1.1 Inleiding

KTechlab is een IDE voor elektronische schakelingen en microcontrollers. Het kan simulatie uitvoeren van een variëteit of componenten (logische, geïntegreerde, lineaire, niet-lineaire en reactieve), simulatie en debugging van PIC microcontrollers via gpsim en komt met zijn eigen nauw gekoppelde en complementaire hoog niveau talen: FlowCode en Microbe.

Het is ontworpen om gemakkelijk in gebruik te zijn en zo weinig opdringerig als mogelijk; alle componenten en FlowParts hebben contextgevoelige help en simuleren van elektronica is even eenvoudig als componenten naar het werkgebied te slepen en connectors te maken die zichzelf automatisch routeren tussen hun pennen. FlowCode staat toe om gebruikers die nieuw zijn in PIC's om direct hun eigen programma's te maken, terwijl de elektronische simulatie toestaat door een assemblyprogramma van PIC in een circuit te stappen.

1.2 Documenten

Om te beginnen in KTechlab moet u een nieuw document aanmaken, waarvan het type zal afhangen van uw taak:

- FlowCode-document - Construeer een PIC programma via het maken van een flowchart.
- Circuit-document - Simuleer elektronische schakelingen en microcontrollers.
- Microbe-document - Hoog niveau taal voor PIC's, ook gebruikt door FlowCode om assembly te genereren.
- Assembly-document - Begin met schrijven van een PIC assembly-programma.

KTechlab gebruikt een Document-weergave model, waarbij de Document logica compleet separaat is van de geopende weergaven van het document. Hierdoor kunt u meerdere weergaven van hetzelfde bestand hebben.

Bij het creëren van een nieuw document, wordt de weergave in een separate tab aangemaakt. Elke tab kan een groot aantal weergaven hebben, in elk gewenst patroon ingedeeld. Hierdoor kunt u, bijvoorbeeld, een in een circuit opgenomen PIC programma simuleren, terwijl u in hetzelfde tab een assembly-document door het programma heen stapt.

U kunt de inhoud van een tab dupliceren door de tab te verslepen naar een leeg gedeelte van de tab balk. U kunt het ook aan een al bestaande tab toevoegen door het naar die tab te verslepen.

Gedetailleerde instructies over de bovenstaande documenten zijn te vinden in hun eigen overeenkomstige hoofdstukken.

1.3 Tekenen

Bij documenten over elektronische schema's en FlowCode, zijn er verschillende tekenmiddelen beschikbaar, waaronder tekst. Deze komen beschikbaar als u in de werkbalk op het paintbrush pictogram klikt. Om te tekenen, gaat u met de muis naar een geschikt stuk gereedschap om een vorm of een lijn te tekenen.

Als u een tekening heeft geselecteerd, dan kunt u het formaat wijzigen door aan zijn handles te slepen. Door de **Shift** ingedrukt te houden tijdens het slepen, zal de handle aan het onderliggende raster vast klikken. Elk stuk gereedschap heeft standaard bewerkingen (zoals kleuren) die vanaf de werkbalk toegankelijk zijn. Er zijn ook meer geavanceerde mogelijkheden die te vinden zijn in de **Item bewerken**-zijkbalk, zoals lijn- en cap-stijlen.

Hoofdstuk 2

PIC programma's

2.1 Manipulatie

Wanneer u een FlowCode- of tekstdocument maakt, zult u een afrolmenu in de werkbalk bemerken met een pictogram van een raket. Vanaf daar kunt u uw PIC-programma manipuleren; het wijzigen in verschillende vormen.

- **Naar Microbe converteren** - Dit wordt alleen in FlowCode documenten gebruikt. Dit wordt verder uitgelegd in hoofdstuk 4.
- **Naar Assembly converteren** - dit kan gebruikt worden in vier contexten. Wanneer een FlowCode-document open is, zal de uitvoer van de FlowCode in de vorm van assembly instructies zijn. Wanneer een Microbe-document open is, zal het het **microbe** programma gedistribueerd met KTechlab aanroepen om het programma te compileren. Evenzo, als een C-programma open is, zal het proberen om het te compileren via SDCC. Wanneer een tekstdocument met PIC-hex open is, zal het **gpasm** aanroepen om het hex te disassembleren.
- **Naar hex converteren** - dit kan ook gebruikt worden in vier contexten. Zoals met **Naar assembly converteren**, kan dit gebruikt worden met FlowCode-, Microbe- en C-documenten. Het zal ook ingeschakeld worden wanneer een assembly-document open is om het te assembleren via **gpasm**.
- **Naar PIC uploaden** - Dit assembleert het PIC programma dat nu wordt bewerkt en uploadt het met de programmeur die de gebruiker heeft geselecteerd.

Geen van deze acties vereisen dat het huidige document wordt opgeslagen - erg nuttig wanneer een snel programma is vereist. Voor niet-PIC doelen, zal de **Uitvoer**-dialoog aangeroepen door te klikken op een van deze acties kan ofwel uitvoer als resultaat hebben (altijd tekst in de bovenstaande drie gevallen) naar een vers document of naar een bestand. Als de uitvoer is opgeslagen naar een bestand, biedt het ook opties om het bestand na maken te laden en het nieuw gemaakte bestand toe te voegen aan het geopende project (als er een is geopend).

Merk op dat u er voor kunt zorgen dat KTechlab altijd dezelfde weergave voor tonen van de uitgevoerde inhoud gebruikt door de optie Instellingen onder **Algemeen** te selecteren.

2.2 Uploaden

KTechlab gebruikt programmeurs van derden om programma's naar PIC's te uploaden. een variëteit van algemene programmeurs komt voorgedefinieerd. Anderen kunnen toegevoegd worden via de dialoog **Instellingen**.

Het handboek van KTechlab

De lijst met poorten wordt verkregen door te scannen naar seriële en parallelle poorten die zijn te lezen en te schrijven. Naar seriële poorten wordt gezocht in:

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

Naar parallelle poorten wordt gezocht in:

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

Hoofdstuk 3

Electronische schema's

3.1 Componenten plaatsen

Links bevindt zich het tabblad **Componenten**.

Een component uit de zijbalk in het schema slepen zal het onder de muiscursor plaatsen. U kunt ook op een item in de zijbalk **Componenten** dubbelklikken om het herhaaldelijk toe te voegen aan het schema. In deze modus zal een kopie van de geselecteerde component herhaaldelijk bij links klikken worden onder de muis worden geplaatst totdat of **Esc** wordt ingedrukt of een klik rechts met de muis.

Om een component een andere positie te geven, klik er links op en versleep. U zult zien dat het vastklikt op het onderliggende raster. Als u de component buiten de rechter- of onderrand van het werkgebied verplaatst, zal het werkgebied zichzelf vergroten om het te laten passen.

Bij alle componenten kunt u de oriëntatie opgeven; 0, 90, 180 en 270 graden. De types die over de as niet symmetrisch zijn, kunt u ook omkeren. Om een selectie van componenten te roteren, rechtsklik en selecteer in het contextmenu het menu-item **Oriëntatie**, of klik op de roteer-knoppen in de werkbalk. Die laatste kunt u bereiken door op de [and] toetsen (bekend bij Inkscape gebruikers) te drukken. In de **Item**-zijbalk (aan de rechterkant) is een handige methode aanwezig voor het instellen van de oriëntatie vanwege de voorbeelden van de componenten. De componenten omkeren is verder uitsluitend mogelijk via de **Item**-zijbalk.

3.2 Componenten verbinden

Er zijn twee methoden voor maken van verbindingen (draden): automatisch, en handmatig. U kunt een van deze methoden via het **Verbinding opbouw methode** pulldown menu in de werkbalk. Experimenteer met beide - automatisch verbinden is vaak handiger bij kleine schema's, waar bij complexere schema's vaak handmatig verbinden nodig is.

In de automatische mode, creëert u een verbinding door van een pin of van een al bestaande verbinding te slepen en daarna de muis boven de gewenste pin of verbinding los te laten. U zult dan zien dat de rechte lijn oranje kleurt bij het los laten van de muis als het een geldige verbinding is. Als de lijn zwart kleurt, dan is dat omdat er niets beneden de muiscursor is of omdat u twee items probeert te verbinden die al verbonden zijn. Bij flow-charts zijn de criteria voor een geldige verbinding meer complex - maar daar komen we later op terug.

De beste manier om handigheid in het handmatig verbinden te krijgen is door ermee te experimenteren. Klik op de startpunt of op de verbinding, en verleng dan de proto-connector door de muis weg te bewegen van de plek waar u heeft geklikt. Om een hoekpunt te plaatsen, klikt u met

de linkermuisknop. Om het tekenen van verbindingen af te breken, drukt u naar keuze op **Esc**, of met de rechtermuisknop.

KTechlab probeert om de routes voor de door u gemaakte verbindingen te behouden. Maar, als het verslepen van een component erin resulteert dat de eindpunten van een verbinding dichterbij elkaar komen, dan wordt KTechlab gedwongen om de verbinding via auto-routing opnieuw te tekenen. Voor het verplaatsen van een component, kunt u zien welke verbindingen een nieuwe route kunnen krijgen - omdat ze grijs kleuren als u erop klikt.

Om een bestaande verbinding te verwijderen, selecteert u het door een klein selectievierkantje over een gedeelte van de verbinding te tekenen en op **Del** te drukken.

3.3 Attributen van componenten

De meeste componenten hebben bewerkbare attributen, zoals de weerstand bij weerstanden. Standaard kunt u als u een groep van hetzelfde soort componenten heeft geselecteerd, via de werkbalk eenvoudig attributen bewerken. Als in uw selectie meerdere soorten componenten aanwezig zijn (zoals weerstanden en condensatoren), dan krijgt u geen bewerkbare attributen te zien.

Sommige componenten hebben geavanceerde attributen die niet via de werkbalk toegankelijk zijn. U kunt ze vinden in zijbalk **Item** aan de rechterkant. De diode, bijvoorbeeld, heeft een aantal karakteristieken die u hier kunt bewerken.

Er is een type attribuut dat niet is te bewerken door de werkbalk of zijbalk Item - tekst over meerdere regels. Dubbelklikken op het item zal een dialoogvak laten verschijnen waar de tekst ingevoerd kan worden.

3.4 Simulatie

De simulatie zal standaard worden uitgevoerd wanneer u een nieuwe elektronisch schema aanmaakt. De status van de simulatie wordt rechtsonder in de weergave van het schema weergegeven en kan gewijzigd worden via het menu **Hulpmiddelen**. Eerst - enige uitleg over hoe de simulator werkt. Dit zou u in staat stellen er het meeste uit te halen.

Als u een schema heeft gecreëerd of gewijzigd, dan worden de betreffende gebieden verdeeld in groepen van pinnen en verbindingen die als onafhankelijk van elkaar beschouwd kunnen worden. Elke groep wordt als onafhankelijke entiteit gesimuleerd (maar nog steeds via de componenten op elkaar reagerend), waarbij de simulatie afhankelijk is van de complexiteit van de groep. Complexere groepen, zoals die met niet-lineaire componenten zoals LEDs, hebben veel tijd nodig om te simuleren. Groepen die alleen logische pinnen heeft, of waar maar een variabele de waarde van die pinnen controleert, hebben het minste tijd nodig voor de simulatie.

Het resultaat van de simulatie wordt geleverd via verschillende grafische middelen.

De pennen op de componenten zullen zijbalken met voltages tonen. Deze worden oranje gekleurd voor positieve en blauw voor negatieve voltages. Hun lengte hangt af van de hoogte van het voltage en hun breedte van de hoeveelheid stroom door de pin. Dit kan uitgeschakeld worden in de pagina **Algemeen** van de dialoog **Configuratie**.

Als de muis boven een pen of een verbinding zweeft dan krijgt u een kleine tekstballon te zien dat de spanning en de stroom op dat punt in het schema toont. Verschillende componenten kunnen ook grafische feedback geven - bijvoorbeeld, LEDs en spannings- of stroommeters.

Tenslotte is er ook nog de oscilloscoop, dat in de volgende sectie wordt besproken.

3.5 Oscilloscoop

De oscilloscoop kan de data van logica, spanningen en stromen opnemen. De logica-probe is geoptimaliseerd voor de opslag van booleaanse samples, en moet daarom bij het meten van logica worden gebruikt in de plaats van de spanning-probe.

Om data te verzamelen, creëert u eerst een probe, en verbind die met het gewenste punt in het schema. U krijgt onmiddellijk in de oscilloscoop het resultaat te zien. Door meer probes toe te voegen zal de resultaten naast elkaar geplaatst worden - u kunt deze herpositioneren door de pijlen aan de linkerkant van het beeld op de oscilloscoop te verslepen, en u kunt de kleur daarvan wijzigen via de attributen van de probes.

U kunt voor spanning en stroom-probes het meetbereik aanpassen in de **Item Bewerken**-zijkant aan de rechterkant.

U kunt met een schuifknop IN- en UIT-zoomen. De schaal is logaritmisch; Voor elke paar pixels dat de schuifknop beweegt, wordt de zoom factor vermenigvuldigt met een constante. KTechlab simuleert logica tot de maximale precisie van 1 microseconde, en bij een maximale zoom, wordt een microseconde voorgesteld door 8 pixels.

Als u de schuifknop naar het eind heeft verschoven, dan blijft het daar staan als nieuwe data binnenkomt. Als de schuifknop niet aan het eind staat, dan blijft de positie van de schuifknop op het zelfde tijdpunt staan. Het beeld op het scherm van de oscilloscoop kunt u voorwaarts en achterwaarts in de tijd verplaatsen door met de linkermuisknop het beeld te verslepen. Vanwege beperkingen van het gebruikte widget system, verloopt het verschuiven nogal schokkerig als u maximaal ingezoomd bent.

Als u in het beeld van de oscilloscoop met de rechtermuisknop klikt dan opent u een contextmenu waarin u kan instellen hoe vaak het beeld van de oscilloscoop wordt bijgewerkt. Hierdoor kunt u kiezen tussen een vloeiender weergave en minder belasting van de microprocessor.

3.6 Subcircuits

Subcircuits zijn een nette en herbruikbare manier om een schema te gebruiken, als u alleen geïnteresseerd bent reacties van de externe verbindingen van het circuit. Het subcircuit is net als een IC, met de pinnen als interface met het interne circuit.

Eerst moet u een schema construeren dat als sjabloon gaat dienen voor het subcircuit dat u gaat construeren. De interface-punten definieert u met de **Externe verbinding** componenten. Deze moet u plaatsen en daarna verbinden op de locatie waar u de pin in het subcircuit IC wilt hebben.

Selecteer vervolgens de groep componenten en de externe verbindingen die u naar een subcircuit wilt omzetten, en selecteer in het contextmenu **Subcircuit aanmaken**. U kunt een naam opgeven voor het subcircuit. Nadat het subcircuit is gecreëerd, zal in de **Componenten** onder de selectie **Subcircuits** de naam verschijnen. Wat u vervolgens als een normaal component kan hanteren - inclusief de mogelijkheid van het verwijderen door op het item te klikken met de rechtermuisknop en **Verwijderen** in het contextmenu te selecteren.

Hoofdstuk 4

FlowCode

4.1 Inleiding

Met FlowCode kunt u makkelijk en snel een PIC programma construeren. Nadat de gebruiker met behulp van de aanwezige programma-onderdelen een flowchart heeft samengesteld, kan KTechlab daarna de flowchart naar een aantal formats converteren. Voor een uitvoer naar hex, bijvoorbeeld, treden de volgende conversies op:

1. De FlowCode is geconverteerd naar Microbe; een hoog-niveau taal waarvan de compiler meegeleverd is met KTechlab.
2. Het uitvoerbare programma **microbe** compileert daarna het Microbe-bestand naar PIC assembly.
3. Tenslotte neemt **gpasm** het PIC assembly-bestand, en converteert het programma naar hex.

Natuurlijk, als u **gputils** niet heeft geïnstalleerd - dat met **gpasm** mee wordt gedistribueerd - dan kan de laatste stap niet worden uitgevoerd.

4.2 Een programma aanmaken

Elke FlowCode-programma moet een uniek startpunt hebben - dit is de plek vanaf waar het programma gaat werken op een PIC startup. Om dit punt te definiëren, opent u aan de linkerkant de FlowParts zijbalk, en versleept een **Start** part. In KTechlab mag u deze maar een keer gebruiken.

U kunt uw programma construeren door de voorgedefinieerde parts aan de linkerkant te gebruiken of door uw eigen code (in assembly of Microbe-formaat) via het **Ingebedde code** part te gebruiken. De flow van het programma stelt u in via de verbindingen tussen de FlowParts - Section 3.2 geeft meer informatie over het maken van verbindingen.

FlowCode geeft daarnaast beperkingen aan circuits voor wat verbonden kan worden. Bijvoorbeeld, elk FlowPart kan maar één output-verbinding hebben. Andere beperkingen worden beschreven in Section 4.4.

4.3 PIC-instellingen

Wanneer u een nieuw FlowCode document aanmaakt, zult u een plaatje van de PIC opmerken die u gebruikt in de linksboven hoek van het werkgebied. Dit representeert de initiële instelling van de PIC.

Elke pin toont op het plaatje van ed PIC het initiële type van de pin (invoer of uitvoer), en zijn initiële status (hoog of laag). U kunt deze wijzigen door de pin te verslepen om het type in te stellen en erop te klikken om zijn status om te schakelen.

In het dialoog **Instellingen**, dat u opent door op de knop **Instellingen** te klikken, kunt u ook de beginwaarden voor de pin-types en hun statussen instellen - in dit geval, door de binaire waarden die naar de PORT en TRIS registers te bewerken. En de pin bezetting kunt u via een dialoog de beginwaarden van de variabelen in het PIC programma bewerken.

Onderaan is een lijst met voorgedefinieerde pin bezettingen, met daarbij knoppen om ze te bewerken. Pin bezettingen worden gebruikt om te bepalen hoe een zeven segment display of een toetsenbord met een PIC is verbonden. Om een **Zeven Segment** of een **Toetsenbord** in FlowCode te kunnen gebruiken, moet u eerst een pin bezetting definiëren.

4.4 Nestling FlowCode

Veel FlowParts, zoals subroutines en loops, kunnen zelf ook weer code bevatten. Na het creëren van een dergelijke container, kunt u daaraan weer FlowParts toevoegen door ze er simpel naar de container toe te verslepen. De container wordt dan gemarkeerd om aan te geven dat het de nieuwe parent van de FlowPart is geworden.

De container is verantwoordelijk voor de daarin geneste FlowParts. Als de expand knop niet is ingedrukt, zijn alle FlowParts daarin verborgen - en op dezelfde manier, komt de inhoud weer zichtbaar als u op de expand knop drukt. Tussen Flowparts in verschillende containers kunt u geen verbindingen maken en de inhoud van een container wordt bij een verplaatsing van de container mee verplaatst.

Hoofdstuk 5

Microbe

5.1 Inleiding en algemene syntaxis

Microbe compileert programma's geschreven in de speciale taal voor PICs, en is een programma dat bij KTechlab hoort. De syntaxis is ontworpen voor een FlowCode programma. De syntaxis voor het opstarten van **microbe** vanaf de commandoregel is:

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

waarbij de opties zijn:

- `--show-source` - Voegt elke regel van de Microbe broncode als commentaar toe aan de assembly uitvoer voor de assembly instructies zelf in die regel.
- `--no-optimize` - Voorkomt optimalisatie van de uit de broncode gegenereerde instructies. Optimalisatie is normaal gesproken veilig, en daarom wordt deze optie hoofdzakelijk gebruikt bij debugging.

Het `.microbe` invoer-bestand moet de PIC identificeren door de PIC-naam aan het begin van het `.microbe` bestand te plaatsen; bijv. de naam van een PIC16F84 is "P16F84".

Example 5.1 Eenvoudig compleet Microbe programma

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

5.1.1 Naamconventies

De volgende regels zijn van toepassing op namen van variabelen en labels:

- Ze kunnen alleen alfanumerieke tekens [a..z][A..Z][0..9] bevatten en de underscore “_”.
- Ze zijn hoofdlettergevoelig.
- Ze mogen niet met een cijfer beginnen.
- Ze zouden niet mogen beginnen met ‘__’ (dubbele underscore), omdat dit gereserveerd wordt voor gebruik door de compiler.

5.1.2 Conventies voor haakjes

Accolades, {}, geven de start en het eind van een code block aan. U kunt ze overall voor de start en achter het eind van een code block plaatsen. Voorbeelden van acceptabele code blocks:

```
statement1 {
    enige code
}
```

```
statement2 {
    andere code }
```

```
statement3
{
    andere code
}
```

```
statement5 {
    codeblok
} statement6
```

5.1.3 Commentaar

Commentaar plaatsten gaat zoals bij C. // maakt van de rest van de regel commentaar. /* en */ geeft een meerregelige commentaar aan.

```
// dit is een commentaar
x = 2
/* En dit is
meerregelig commentaar */
```

5.1.4 Programma structuur

U moet de PIC id aan het begin van het programma toevoegen. Het eind van het hoofdprogramma geeft u aan met ‘end’. Subroutines moet u achter ‘end’ plaatsen.

5.1.5 Subroutines

U kunt van elke plaats in de code een subroutine aanroepen. Syntax:

```
sub SubName
{
    // Code...
}
```

De subroutine roept u aan met ‘call *SubName*’.

5.2 Referentie Microbe taal

5.2.1 if

Conditionele tweesprong. Syntax:

```
if [expression] then [statement]
```

of

```
if [expression] then  
{  
    [statement block]  
}
```

Vergelijkbaar met else:

```
else [statement]
```

of

```
else  
{  
    [statement block]  
}
```

Example 5.2 if

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

5.2.2 alias

Vervangt een tekenreeks door een ander als alias. Syntax:

```
alias [from] [to]
```

5.2.3 repeat

Herhaalt het uitvoeren van het statement block uit totdat de expressie waar is. De controle of de expressie waar is, wordt na het statement block uitgevoerd, daarom zal de statement block altijd tenminste een keer worden uitgevoerd. Syntax:

```
repeat  
{  
    [statement block]  
}  
until [expression]
```

5.2.4 while

Vergelijkbaar met repeat, dit herhaalt het uitvoeren van het statement block uit. Maar de controle of de expressie waar is wordt voor de de uitvoer gedaan, niet daarna. daarom zal als uitkomst van de controle op de expressie negatief is, de statement nooit worden uitgevoerd. Syntax:

```
while [expression]
{
    [statement block]
}
```

5.2.5 goto

Dit zorgt dat het programma verdergaat met de volgende statement na het opgeven label. Goto syntax:

```
goto [labelname]
```

Label syntax:

```
labelname:
```

Het vermijden van het gebruik van goto wordt vaak als goed programmeren gezien. Het gebruik van control statements en subroutines zal resulteren in een beter leesbaar programma.

Example 5.3 goto

```
goto MyLabel

...

[MyLabel]:
// Code gaat verder op dit punt
```

5.2.6 call

Roept een subroutine aan. Syntax:

```
call [SubName]
```

waarbij *SubName* de naam van de subroutine is die u aanroept.

5.2.7 delay

Dit veroorzaakt dat het uitvoeren van de code voor de opgeven periode stopt. De interval is in millisecondes. Syntax:

```
delay [interval]
```

OPMERKING

Op dit moment, wordt door Microbe aangenomen dat de PIC werkt met een kloksnelheid van 4MHz - bijv. elke instructie heeft 1 microseconde nodig om verwerkt te worden. Als dat niet het geval is, dan moet u de interval proportioneel aanpassen.

5.2.8 sevenseg

Dit wordt gebruikt voor het definiëren van de pinbezetting van een (common cathode) zeven segment display verbonden met de PIC. Syntax:

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

waar [a]...[g] de pinnen van de PIC zijn waarmee de respectievelijke segmenten van de zeven segment display zijn verbonden. De pinnen kunnen zowel als PORTX.N of RXN worden geschreven.

Bij het tonen van een cijfer op de zeven segment, wordt de pin bezetting behandeld als een alleen lezen variabele.

Example 5.4 Een uitvoer naar een zeven segment definiëren

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

5.2.9 toetsenbordje

Dit gebruikt u om de pinnen van een PIC te definiëren waaraan een toetsenbordje is verbonden. Syntax:

```
keypad [name] [rij 1] ... [rij 4] [column 1] ... [column n]
```

waar [rij 1] ... [rij 4] en [column 1] ... [column n] de pinnen van de PIC zijn waaraan de respectievelijke rijen en kolommen van het toetsenbordje zijn verbonden (op dit moment kunt u het aantal rijen niet wijzigen). zie Section 5.2.8 (hierboven) voor meer informatie over pin bezetting.

U moet de kolommen van het toetsenbord met 100k weerstanden omlaag trekken naar aarde. U moet de pinnen van de rijen als uitgang configureren en de pinnen van de kolommen als ingangen. Nadat het toetsenbordje is gedefinieerd, wordt het behandeld als een read only variabele.

Example 5.5 Een toetsenbord definiëren en uitlezen

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

De waarden die u van een toetsenbord inleest zijn standaard:

- De waarde van het getal als het numerieke toets is (1 tot 3 voor de bovenste rij; hexadecimal A tot D langs de vierde kolom omlaag en verder voor elke extra kolom).
- 253 voor de toets in rij 4, kolom 1.
- 254 voor de toets in rij 4, kolom 3.

U kunt deze waarden herdefiniëren door gebruik van het alias commando, waarbij de naam van de toets in rij x, kolom y (rijen en kolommen starten bij 1), is Keypad_x_y. Bijvoorbeeld, om de ster toets op een 4x3 toetsenbord de waarde nul te geven, moet u de volgende alias gebruiken:

Example 5.6 De toets van een toetsenbord vervangen door een waarde

```
alias Keypad_4_1 0
```

5.3 PIC I/O

5.3.1 Poort richting

De poort richting stelt u in door een waarde toe te kennen aan TRIS*, waar * de poortletter is. Als voorbeeld:

Example 5.7 Poort richtingen instellen

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

Dit hierboven stelt de pinnen RB1, RB2 en RB7 op PORTB in als uitgangen, en de andere pinnen op PORTB als ingangen. In dit voorbeeld is b'01111001' een binaire representatie van het output type. De 1 aan de rechterkant stelt een uitgang op RB0 voor, en de 0 aan de linkerkant stelt een ingang op RB7 voor.

5.3.2 Port I/O

U kunt een poort behandelen alsof het een variabele is. Bijvoorbeeld:

Example 5.8 Naar een poort schrijven

```
x = PORTA
```

Dit hierboven koppelt de waarde van PORTA aan de variabele x.

5.3.3 Pin I/O

De pin van een poort kunt u krijgen door het pinnummer vooraf te laten gaan door de naam van de poort; bijv. Pin 2 (startend van Pin 0) van PORTA is bekend als *PORTA.0*. De syntax voor het instellen van de status van een pin is:

```
PORTX.N = STATE
```

waar *STATE* kan zijn *hoog* of *laag*. De syntax voor het testen van de status van een pin is:

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

Door deze voorbeelden te combineren krijgen we:

Example 5.9 Testen en instellen van de pin-status

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

5.4 Variabelen

Alle variabelen zijn 8-bit unsigned integers, in de range van 0 tot 255. Microbe ondersteund de standaard bewerkingen met maar een variabele en binaire bewerkingen (werkend met twee variabelen) die door de PIC kunnen worden uitgevoerd. Daarnaast ondersteund Microbe ook delen en vermenigvuldigen.

5.4.1 Bewerkingen met één variabele

- *rotateleft x* - Verschuift de bits van variabele X naar links, waarbij de bit die links eraf valt weer rechts wordt toegevoegd.
- *rotateright x* - Verschuift de bits van variabele X naar rechts, waarbij de bit die rechts eraf valt weer links wordt toegevoegd.
- *increment x* - Verhoogt de variabele x met 1. Als x de waarde van 255 heeft, dan klappt x om naar 0.
- *decrement x* - Vermindert de variabele x met 1. Als x de waarde van 0 heeft, dan klappt x om naar 255.

5.4.2 Rekenen

Ondersteunde bewerkingen:

- *Optellen*: $x + y$
- *Aftrekken*: $x - y$
- *Vermenigvuldigen*: $x * y$
- *Delen*: x / y
- *Binaire XOR*: $x \text{ XOR } y$
- *Binaire AND*: $x \text{ AND } y$
- *Binaire OR*: $x \text{ OR } y$

5.4.3 Vergelijken

Ondersteunde bewerkingen:

- *Is gelijk aan*: $x == y$
- *Is niet gelijk aan*: $x != y$
- *Is groter dan*: $x > y$
- *Is kleiner dan*: $x < y$
- *Is groter of gelijk aan*: $x >= y$
- *Is kleiner of gelijk aan*: $x <= y$

Bijvoorbeeld:

Example 5.10 Vergelijken

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

Hoofdstuk 6

Debuggen

6.1 De debugger starten

Debuggen is mogelijk bij Assembly, SDCC en Microbe, als ze als tekstdocument zijn geopend. Hierin, kunt u de stappen regelen via het **Debug** menu. Er zijn twee manieren om de debugger te starten.

Als het PIC programma al in een circuit is opgestart, dan zal als u op het PIC component dubbelklikt, het programma openen. Bij assembly PIC programma's, zal de debugger voor dat tekst document worden gelinkt met het PIC component. In dit geval, zal het debug menu het PIC programma niet kunnen stoppen - omdat het nu eigendom is van het PIC component.

Als het assembly-bestand al is geopend, dan kunt u de debugger opstarten via het **Debug** menu. Na de compilatie van het programma, staat de debugger klaar, met het PIC programma gepauzeerd op de eerste instructie. Merk op dat bij het debuggen van high level talen, u het gemarkeerde executie point niet te zien krijgt als er geen regel is dat overeenkomt met de eerste uit te voeren assembly instructie. Klik in dat geval op **Volgende** om het executie point naar de eerste regel van het programma te brengen.

6.2 De debugger besturen

De debugger kan in een van de volgende twee modus zijn: lopend, en stappend. Tijdens het lopen, zal het PIC programma in realtime worden gesimuleerd. Om te kunnen stappen, moet het PIC programma worden gepauzeerd - naar keuze door te klikken op **Onderbreken** in het **Debug** menu, of door te klikken op de pause-knop van het PIC component.

Bij de stap-mode, geeft een groene pijl in de kantlijn van het tekstdocument de volgende uit te voeren regel aan (bekend bij gebruikers van KDevelop). Het kan handig zijn om de kantlijn in te schakelen via het **Weergave** menu (u kunt het permanent inschakelen via het dialoog voor **Bewerken instellen**).

Er zijn drie typen voor stappen:

- **Stap** - Dit voert de geselecteerde instructie uit. De groene pijl gaat daarna naar de volgende uit te voeren instructie.
- **Stap over** - Als de volgende uit te voeren instructie een call is, of dergelijk, dan zal het een "stap over" de call doen, terugkerend naar de stapmode nadat de call is geëindigd. anders zal over een instructie stappen net zoals een normale stap zijn. Technisch uitgedrukt - de start stack level is recorded, en de uitvoer van het programma is gepauzeerd nadat de stack level terug gaat naar zijn start level.

- **Stap uit** - Als de geselecteerde instructie in een call is, of dergelijk, dan zal het wachten totdat de call eindigt. Vergelijkbaar met de stap over, dit is vergelijkbaar met het wachten totdat de stack level een niveau lager is dan het start level, als het start level groter is dan nul.

Met breakpoints kunt u de uitvoering van het PIC-programma pauzeren als het een bepaalde instructie heeft bereikt. Om een breakpoint op de door de cursor gemarkeerde regel in te schakelen, gebruikt u naar keuze het **Debug** menu, of u klikt op de kantlijn van het tekst document.

De **Symbolviewer** zijbalk rechts toont de waarden van de Special Function Registers. Om de waarde van een variabele in de General Purpose Registers uit te vinden, kan u met uw muis boven de in de instructie voorkomende variabelenaam dat dit register bewerkt. Merk op dat de radix instelling bij de **SymbolViewer** ook regelt hoe de waarde wordt getoond als de muis boven een variabele zweeft.

Hoofdstuk 7

FAQ

1. *KTechlab belast de CPU zwaar*

Er kunnen meerdere redenen daarvoor zijn. Simulatie van circuits die zowel reactieve als niet-lineaire componenten (zoals condensatoren en transistors) hebben, hebben veel CPU-tijd nodig voor de simulatie. U kunt de simulatie pauzeren en weer door laten gaan via het **Gereedschap** menu.

Het tekenen van het werkvlak (met name het hertekenen van veel snel veranderingen in de spanningsbalken van pinnen) is ook CPU intensief. U kunt de refresh rate verlagen of de spanningsbalken uitschakelen via het **Instellingen** dialoog. De refresh rate van de **Oscilloscoop** kunt u ook verlagen door rechts te klikken op het scherm ervan.

Vergeet niet dat de volgende major release van KTechlab veel sneller kan zijn bij zowel het tonen van het werkvlak en de simulatie van reactieve en niet lineaire componenten.