



Udviklingsprogram til Microcontrollere

Udviklingsprogrammet uVision2 fra Keil

Det program, vi anvender til at skrive kilde-teksten, assemblerkoden, og til at oversætte det til maskinkode, er fra Keil. Programmet hedder uVision, og er gratis. Det er en nedgraderet version af et større program, idet der maximalt kan genereres 2 K HEX-kode. Men 2 K Hexkode er ret meget i assembler-sproget !!

Programmet kan i en nyere version downloades fra Keils hjemmeside. Men den "gamle" version kan hentes på min hjemmeside. Den virker udmærket.

Filstruktur:

Det er **meget vigtigt**, at man noterer sig, at hver opgave, dvs. forskellige kildetekster hører til hver deres projekt. Man løber helt sikkert vild i programmet, hvis ikke man overholder dette.

Lav fx en mappe til alle dine projekter, og herunder en undermappe for hver projekt !! !



Opstartsprocedure:

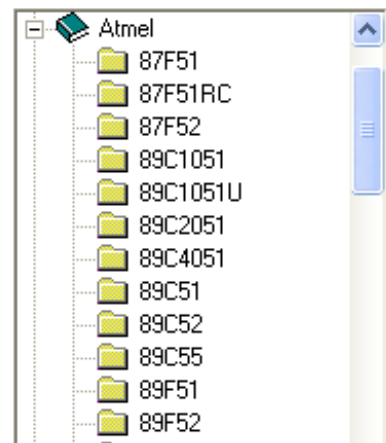
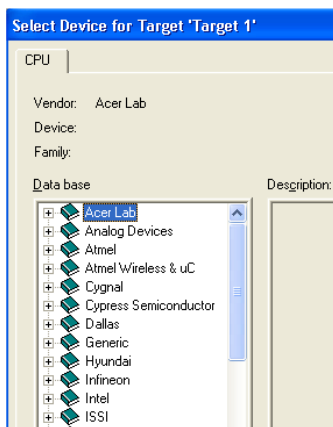
Når **uVision2** startes, åbnes normalt det forrige projekt, man arbejdede på.

Vælg **Project / Close**

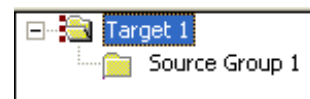
Start nu et nyt project:

Vælg: **Project / NEW Project**, og bestem, hvor projektet skal gemmes. Opret relevante mapper !
Fx Keil\projekter2005\ eget navn\projektnavn. Giv projektet et sigende fornavn. Et projekt får automatisk extension ".uv2"

I det næste vindue, der dukker op " Select Device for Target" vælges **Atmel**, og **Atmel 89C4051**

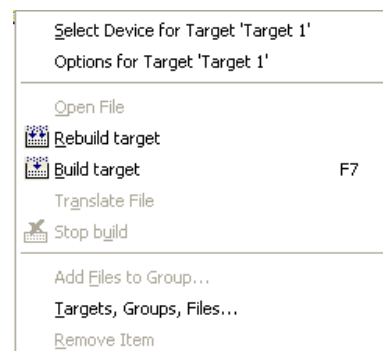


Man har nu i Projektvinduet til venstre **Target 1** og herunder, hvis Target 1 "åbnes" / Source group 1.



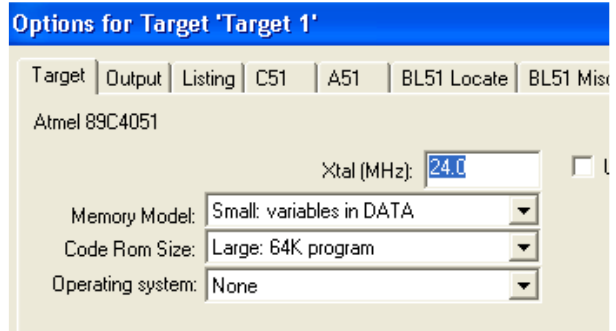
(Evt. skal venstre vindue først vises. Vælg " View / Project Window")

H. Klik på Target 1 og vælg **Options for target "Target 1"**

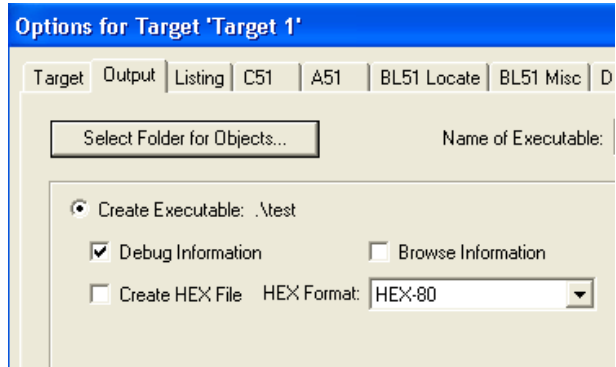




I folderen **target** vælges krystal = 12 MHz. (typisk)



Vælg folderen **Output**, og i "Create Hexfile" sættes hak.



Gå evt her ind i "**Select folder for Objects**" for at tjekke, hvor de filer, der genereres af udviklingsprogrammet, gemmes.

Og evt. i "**Name of executable**" kan bestemmes, hvad output-filen skal hedde, hvis den ikke blot skal have samme navn, som projektnavnet. Output-filen får extension ".HEX"

Vælg **OK**

Indtil videre er det kun projektet, der er oprettet. Der er endnu ingen kildetekstfiler tilknyttet projektet.

Opret nu et dokument til kildeteksten.

Vælg nu **File \ New**.

Herved åbnes et hvidt ark i højre vindue. Det er her, kildeteksten skal skrives.

Start med at gemme det endnu tomme dokument. Vælg **File / Save As**. Filen skal have et navn, og gemmes med extension "**.a51**"

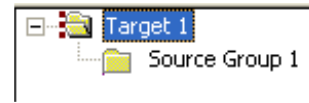
Husk extension .a51. Herved får genkendte ord i editoren forskellige farver, hvilket er en stor hjælp til fejlretning.

Tilknytning af kildeteksten til projektet:



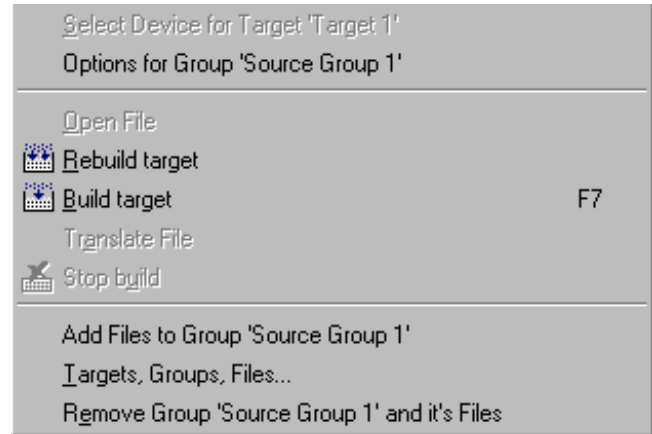
Inden filen kan oversættes, skal den tilknyttes projektet.

Højreklik på **Source group 1** i Project Window til venstre.



Vælg **Add file to source group** og vælg kildetekst-filen, der lige er skabt.

Husk, den skal være med extension ".a51"



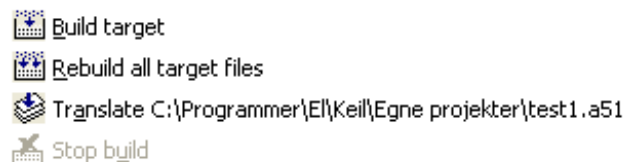
Venstre vindue ser nu således ud, blot er ens filnavn vist under Source Group 1.



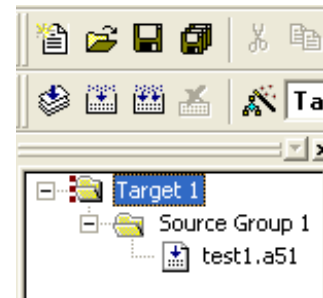
Generering af hex-fil:

Når man er færdig med kildeteksten, gemmes filen.

Vælg øverst **Project / Build Target.**



Til højre ses knapperne til direkte oversættelse af kildeteksten.



Ved oversættelsen, den såkaldte assemblering, skabes en masse filer, bla. en ".HEX" fil, der kan brændes over i controlleren og en ".LST"-fil, der kan printes ud. Den indeholder både kildekoden og de genererede hex-koder mm.

I nederste vindue skulle helst angives noget i retning af:

```
Build target 'Target 1'
assembling atmel.a51...
linking...
creating hex file from "valletst1"...
"valletst1" - 0 Error(s), 0 Warning(s).
```



Indikeres at der er fejl i kildeteksten, dobbeltklikkes på fejltteksten, og linien i kildeteksten med fejlen bliver highlightet.

Se evt. ".A51"-filen i notepad. Her ses hele tekstfilen, der er ens program, – eller rettere kildetekst.

Kildeteksten:

Kildeteksten kan skrives i Notepad, men det er ikke så smart ! Der er mange smarte funktioner i et rigtigt udviklingsværktøj.

Kildeteksten er jo ens program. Har man skrevet flere programmer, finder man tit, at stumper kan genbruges. Dette kan let gøres med Copy & Paste.

Så for det meste er et program en blanding af nyskrevet tekst og genbrug. !

Skærmen er opdelt med 3 vinduer. Til venstre filstrukturen, til højre programvinduet, og nederst meddelevinduet fra oversætteren.

I starten af kildeteksten skal skrives en ordre til oversætteren om at arbejde med "89C2051 sprog".

Det kan se ud som flg:

```
$NOMOD51 ;Se bort fra 8051 registre
$INCLUDE (AT892051.INC) ;Benyt istedet ATMEL89C2051 definitioner
;*****

NAME BLINK8LED ; Navn på program ( Husk i et ord )

;*****
; Def af variable;
    ORG 0000H ;Program start adresse
    Jmp start:

    Org 030h
Start:
    Her kommer programmet !!!!!

    END
```

De nødvendige ting, fx med programhovedet, inkludering af ".inc-fil" osv. kan smart gemmes i en tekst-skabelon-fil. Denne kan så kopieres ind i teksten.



I koden i skabelonfilen er der en linie, hvor der "includeres" en "Includefil". Filen hedder at892051.inc, og denne fil skal ligge i ".ASM" mappen. Den kan findes på min hjemmeside.

Denne include-fil giver oversætteren viden om, hvor i controlerens RAM, de forskellige registre ligger osv.

Include-filen indeholder en masse opsætning, dvs, der er angivet hvor Controlerens RAM, fx reg. A er placeret osv. Filen kan læses med notepad. Filen skal ligge i "\ASM"-mappen, eller i samme mappe som projektet.

I linien NAME gives programmet et navn, fx test. Navnet skal være i et ord !!! Jeg ved egentlig ikke, hvad det skal bruges til !!

Selve program-koden skal skrives i programvinduet mellem "Org" og "End". Eks:

```
ORG 0000h
SJMP Start          ; Der hoppes til programstart
;
ORG 030h            ; Programmet startes i adresse 30h, i ROM'enså der er mulighed
Start:              ;for interrupts.
MOV ACC,#25
osv.
osv.

END
```

Det er muligt at kopiere programstumper fra andre kilde-filer. Åben dem blot, marker de ønskede linier, og træk med Ctrl trykket ned, eller "Copy & Paste".

På denne måde kan fx en subrutine kopieres.

Subrutiner.

Det er normal kutyme i programmering, at skabe et program bestående af mange underprogrammer, kaldet subrutiner. De kan så kaldes fra forskellige steder i programmet, fx mange gange !!

Subrutinehoveder udstyres med beskrivelse af, hvad subrutinen gør. Herved er det lettere senere at gendanne sig de tanker, man gjorde sig, da rutinen blev skrevet.

Eks. på subrutinehoved:

```
:
;
;*****
;
;
```




```
;Navn      :Pause
;
;Funktion   : Genererer en pause, afh. Af
;
;Kaldes med : Call Pause
;Input     : Værdi i reg A = antal sek pause
;Output:    :
;Ødelægger: : Reg R6 og R7          ( beskrivelse af, hvilke registre der får andre værdier )
;
; *****
Pause:
; Her kommer så det egentlige program !!

RET          ; hop retur fra subrutinen.
```

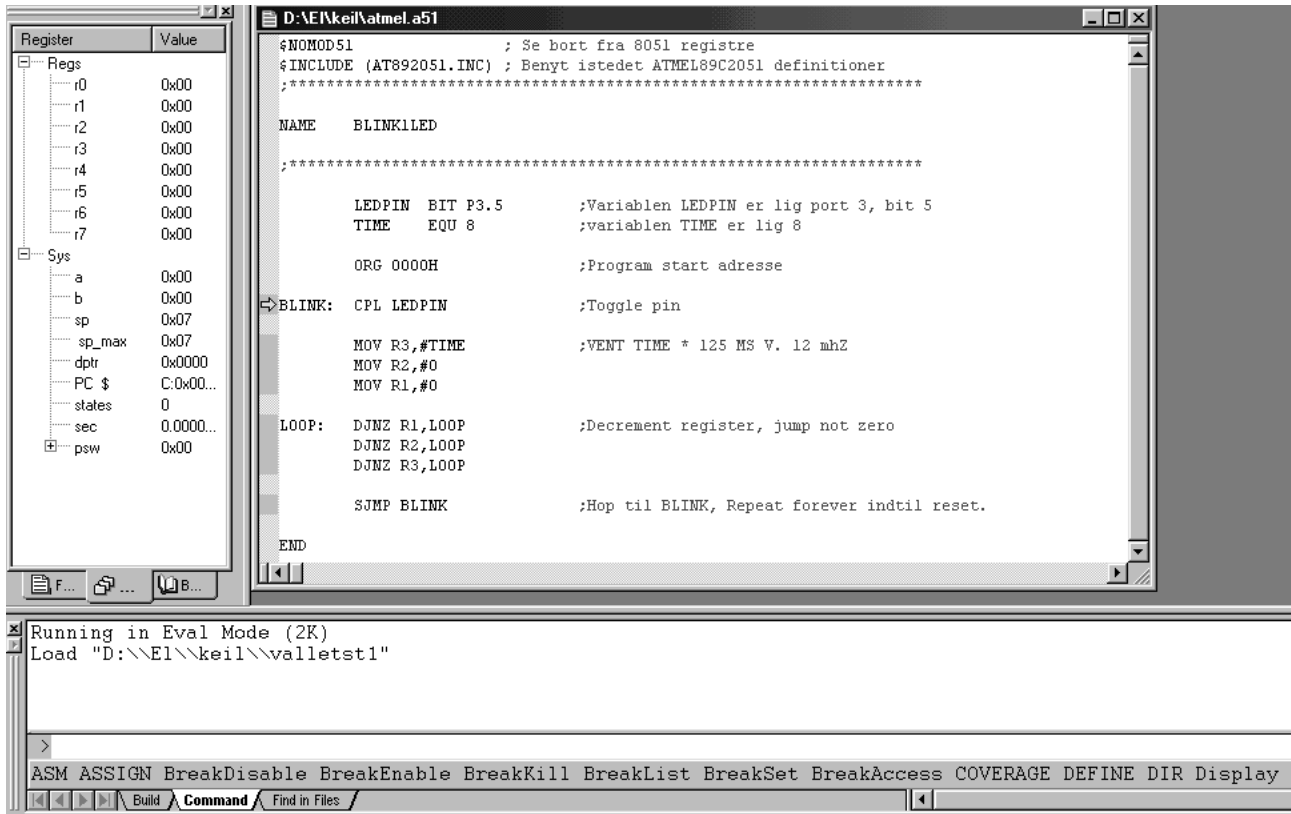
Debug:

Før der brændes over i processoren, **kan programmet debugges**, altså undersøges for fejl. Her gennemgås nogle af mulighederne.

Vælg Project / Debug Eller ikonet for oven til højre med en lup med et  Ctrl+F5

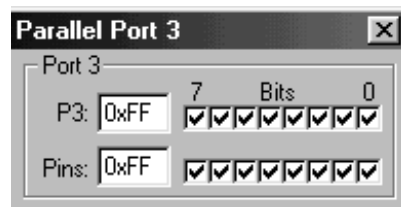
Sig OK til 2K kode, der er max, vores gratis program kan klare.

Skærmen kan nu se ud som flg:



I venstre side kan registrene følges. (Project Window) Til højre ses assembler-programmet.


Vha. Peripherals / I/O port /Port 1 åbnes et nyt vindue, der viser status for portene.



Vær sikker på, at Debug Toolbar vises. (View / Debug Toolbar)



Fra venstre er betydningen som følger: (Kun nogle forklares her !)

Icon	Betydning	Shortcut	Forklaring
	Reset CPU		Programmet hopper til adresse 0, eller der, hvor programmet starter.



UVISION, KEIL

	RUN	F5	Run, execute, udfør program til næste aktive breakpoint
	HALT	ESC	Stop running
	Step into	F11	Udfør en single step ind i en funktion
	Step over	F10	Udfør en single-step forbi en funktion
	Step out	Ctrl+F11	Step ud af nuværende funktion
	Run to cursor line		Udfør indtil nuværende cursor-linie er nået
	Show current statement on program counter		
	Enable / disable trace recording	Ctrl+F7	
	View Trace Records		
	Disassembly window	.	Vis vindue, hvori der er indsat maskinkoder fra disasemblingen
	Watch & Call Stack Window		Se på stakken
	Code Coverage Window		Brug ikke denne !!!
	Serial window # 1		
	Memory window		
	Performance analyser Window		
	Toolbox		Vis "Code update window."

Trykkes på Reset, vil program counteren hoppe til programstart. Dette indikeres med den gule pil i programlinien.

Tryk på får nu processoren til at udføre een ordre. Den gule pil følger med til hvor processoren er kommet til. ??? Og i venstre side kan registrenes værdi følges.

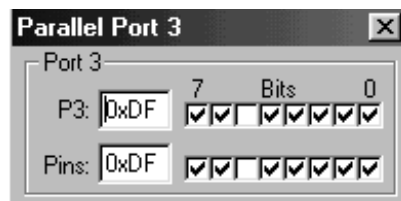
får programmet til at køre, og stopper det. Under programafvikling kan man imidlertid ikke se registrene. De ændrer sig for hurtigt. Men de kan opdateres med den øjeblikkelige værdi, når man

trykker på Windows update. Denne fås frem med

Registrenes værdi, når der er trykket på Update, vises. Det ses, at der arbejdes på r1, r2 og r3. Det ses også, at processoren tilsyneladende har "kørt" 1.924.703.284 cyklusser. Før musen hen over, og det eksakte tal vises.



Register	Value
Regs	
r0	0x00
r1	0x56
r2	0x20
r3	0x06
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0008
states	1924703284
sec	1924.70328400
psw	0x00



Yderligere opdateres portværdien.

Vælges View / Periodic Window Update,



vil portvinduet opdateres periodisk !!!

Det kan være en fordel at tænde vinduet Disassembly window.



Herved kan der følges, hvilke ordre, processoren får.

I Debug er der nogle muligheder for at indsætte breakpoint`s. Disse fungerer således, at programmet kan sættes til at køre – her i debug – mode – indtil det når til et breakpoint. ??? Sæt cursoren i en



linie og dobbeltklik. Eller vælg hånden . Den første er ”Indsæt / remove breakpoint,” den næste ” Kill All Breakpoints” (F7) Dernæst Enable / disable breakpoint, og sidst ”Disable All Breakpoints”

Når programmet er verificeret trykkes igen på



for at afslutte Debug programmet.

















Afslut:

Ved afslutningen husk ”Close project”.




Shortcuts:

F4	Næste fejl
Sh F4	Forrige fejl
F2	Næste bookmark
Sh F2	Forrige bookmark
Ctrl+F	Find
F3	Find igen, forward
Shift+F3	Repeat search backward
Ctrl+F3	Søg efter ord under cursor














Menu Item or Command		Shortcut
	Start/Stop Debug Session Starts or stops a µVision3 Debugging Session.	Ctrl+F5
	Go Runs (executes) the target program until the next active breakpoint.	F5
	Step Executes a single-step into a function.	F11
	Step Over Executes a single-step over a function.	F10
	Step Out of current Function Steps through all instructions in a function and halts after returning to the caller.	Ctrl+F11
	Run to Cursor Line Executes until the current cursor line is reached.	
	Stop Running Stops target program execution.	ESC
	Breakpoints... Opens the Breakpoint Dialog.	
	Insert/Remove Breakpoint Toggles breakpoint on current line.	
	Enable/Disable Breakpoint Enables/disables breakpoint on current line.	Alt+F7
	Disable All Breakpoints Disables all breakpoints in the program.	
	Kill All Breakpoints Removes all breakpoints in the program.	F7
	Show Next Statement Shows next executable statement or instruction.	
	Enable/Disable Trace Recording Enables trace recording.	Ctrl+F7
	View Trace Records Reviews previously executed instructions.	

http://www.keil.com/support/man/docs/uv3/uv3_ui_debug.htm

Menu Item or Command		Shortcut
	Move cursor to beginning of line.	Home
	Move cursor to end of line.	End
	Move cursor to beginning of file.	Ctrl+Home
	Move cursor to end of file.	Ctrl+End
	Move cursor one word left.	Ctrl+Left Arrow
	Move cursor one word right.	Ctrl+Right Arrow
	Select all text in the current file.	Ctrl+A
	Move cursor back to the location before a 'find' or 'go to line' command.	
	Move cursor forward to the location after a 'find' or 'go to line' command.	
	Undo Undo the last edit.	Ctrl+Z



	Redo Redo the last undone edit.	Ctrl+Y
	Cut Cut selected text to clipboard.	Ctrl+X
	Copy Copy selected text to clipboard.	Ctrl+C
	Paste Paste text from clipboard.	Ctrl+V
	Indent Selected Text Indent selected text right one tab stop.	
	Unindent Selected Text Indent selected text left one tab stop.	
	Toggle Bookmark Toggle bookmark at current line.	Ctrl+F2
	Goto Next Bookmark Move cursor to next bookmark.	F2
	Goto Previous Bookmark Move cursor to previous bookmark.	Shift+F2
	Clear All Bookmarks Clear all bookmarks in active file.	Ctrl+Shift+F2
	Find Search for text in the active file.	Ctrl+F
	Repeat text search forward.	F3
	Repeat text search backward.	Shift+F3
	Search for word under cursor.	Ctrl+F3
	Replace Replace specific text.	Ctrl+H
	Find in Files... Search for text in several files.	Shift+Ctrl+F
	Incremental Find Search on a letter-by-letter basis as you type the characters.	Ctrl+I

Kilde 1) ELV journal.

Der kan hentes information på <http://hjemmesider.eucsyd.dk/es4>

og <http://hjemmesider.eucsyd.dk/es4/atmel/indledning.htm>

Tina Hansen, EUCSyd har hjulpet med en gennemgang af udviklingsprogrammet uVision fra <http://www.Keil.com>