SiLabsin ToolStick-kehitysympäristön aloitusopas

Tässä ohjeessa kuvataan Silicon Laboratoriesin C8051Fxxx-kontrolleriperhettä ja sitä tukevan ToolStick-kehitysympäristön käyttöönottoa erityisesti harrastajan näkökulmasta. Esimerkkinä käytetään C8051F330-kontrolleria, joka soveltuu pienen robotin kontrolleriksi hyvin.

JK 18.8.2009

1 Yleistä SiLabsin kontrollereista

SiLabsilla on valikoimissaan yli 200 erilaista kontrollerityyppiä, kaikki nopeus-, muisti- ja kotelointivaihtoehdot huomioon ottaen. Niiden ydin on 8051-arkkitehtuurin mukainen, mutta I/O-toiminnot ovat kehittyneemmät ja käskyjen suoritusnopeus huomattavasti suurempi kuin alkuperäisissä Intelin 8051-kontrollereissa. Kuitenkin ydin on käsky-kannaltaan täysin yhteensopiva alkuperäisen 8051:n kanssa, joten kaikki kääntäjät ja muut työkalut soveltuvat suoraan käytettäväksi myös näiden kontrollereiden kanssa.

Useat kontrollerimallit on suunnattu ”mixed-signal”-käyttöön, joten niissä on monipuoliset A/D- ja D/A-muuntimet. Muunlaisiakin malleja löytyy, mm. USB-portilla varustettuja ja autokäyttöön suunnattuja.

Jotta PC saadaan liitettyä kontrolleriin ja siirrettyä ohjelmat sinne, tarvitaan sovitin-laitteistoa PC:n USB-portin ja kontrollerin väliin. Tähän on kaksi vaihtoehtoa:
- USB Debug Adapter, joka tulee kaikkien SiLabsin Development Kittien mukana
- ToolStick Debug Adapter, joka voidaan liittää ToolStick:iin

Jälkimmäiseen vaihtoehtoon tutustutaan lähemmin seuraavissa kappaleissa.

Debug Adapterin ja kontrollerin välisessä liitännässä on kaksi vaihtoehtoa:
- vanhemmat kontrollerit, tyyppinumeroltaan C8051F000 ... C8051F299, käyttävät JTAG-liitäntää
- uudemmat, tyyppinumerosta C8051F300 eteenpäin käyttävät SiLabsin omaa C2-liitäntää

JTAG- ja C2-liitäntöjen toiminnallisuudet ovat käyttäjän kannalta likimain samat, mutta C2-liitäntä vie vähemmän I/O-nastoja kontrollerista, joten se on edullisempi kontrollereissa, joiden kotelossa on vähän nastoja (pienimmät mallit ovat QFN11-kotelossa).

Molemmat Debug Adapterit tukevat sekä JTAG- että C2-liitäntää, eli niillä voi liittyä kaikkiin SiLabsin kontrollereihin. Kontrollerin ohjelmoinnin lisäksi liitännät mahdollistavat myös debuggaamisen.

Harrastajan kannalta SiLabsin kontrollereiden huonoja puolia ovat: kotelointi (yhtä mallia lukuunottamatta kaikki ovat pintaliitoskoteloissa), käyttöjännite (kaikissa 3,3 V) ja saatavuus (helpointa ja halvinta tilata Digikeystä).

Yleisluettelo SiLabsin eri kontrollereista löytyy SiLabsin sivuilta:

https://www.silabs.com/Marcom%20Documents/Resources/MCU_Catalog.pdf

2 C8051F330:n ominaisuuksia

C8051F330 on SiLabsin kontrollereista ainoa, jota on saatavissa DIP-kotelossa (DIP20). Lisäksi se on mukana ToolStick Starter Kitissä, joten se on luonnollinen valinta harrastajalle, joka lähtee tutustumaan SiLabsin kontrollereihin. Se on myös tehokas yleiskontrolleri, joten sillä voi tehdä ”oikeitakin” sovelluksia tutustumisen jälkeen.

F330:n ominaisuuksia:
- 8 KB flash-ohjelmamuistia
- 768 tavua RAMia
- 16 digitaalista I/O-pinniä
- 4 ajastinta
- sarjaportti
- I2C- ja SPI-väylät
- 3-kanavainen PCA (Programmable Counter Array)
- 16-kanavainen A/D-muunnin, 10-bitin resoluutio
- 1-kanavainen D/A-muunnin, 10-bitin resoluutio
- ei tarvitse ulkoista kidettä
- käskyjen suoritusnopeus n. 25-kertainen alkuperäiseen 8051:een nähden

Kaikki I/O-vaihtoehdot eivät ole yhtä aikaa käytettävissä, koska kotelossa on vain 20 nastaa. Näistä 16 on käytettävissä I/O:lle, koska 2 tarvitaan käyttösähköille ja 2 menee C2-liitäntään. Periaatteessa toinen C2-liitännän nastoistakin on käytettävissä I/O:hon, mutta tämä vaatii hiukan temppuilua, joten on selkeämpi ajatella, että vapaasti käytettäviä I/O-nastoja on 16.

Ohjelmallisesti voidaan valita, mitä I/O-toimintoja missäkin nastassa on. Jokaisen nastan voi asettaa joko analogiseksi otoksi tai digitaaliseksi-I/O:ksi. Lisäksi sisäisten laitteiden (esim. PCA:n tai I2C-väylän) signaalien paikat fyysisissä nastoissa voidaan valita melko vapaasti. Tähän valintaan käytetään kontrollerin sisäistä laitetta nimeltä crossbar. Se on eräänlainen ristiinkytkijä, joka kytkee sisäisten laitteiden signaalit haluttuihin fyysisiin nastoihin. Tämä helpottaa myös piirilevysuunnittelua merkittävästi.

Piirin datalehti on ladattavissa SiLabsin sivuilta:

https://www.silabs.com/Support%20Documents/TechnicalDocs/C8051F33x.pdf

Datalehti sisältää tarkat kuvaukset I/O:sta, mutta 8051-ytimen käskykanta on kuvattu melko ylimalkaisesti. Tarkempi kuvaus löytyy esim. NXP:n sivuilta:

http://www.nxp.com/acrobat_download/various/80C51_FAM_PROG_GUIDE_1.pdf

3 ToolStick

3.1 ToolStickin osat

ToolStick on väline, jolla PC ja käyttäjän omalla piirilevyllä oleva kontrolleri voidaan kytkeä yhteen siten, että kontrolleriin voidaan siirtää ohjelma ja debugata sitä. Sen avulla voidaan ohjelmaa ajaa kontrollerissa myös jo ennenkuin omaa piirilevyä on edes olemassa.

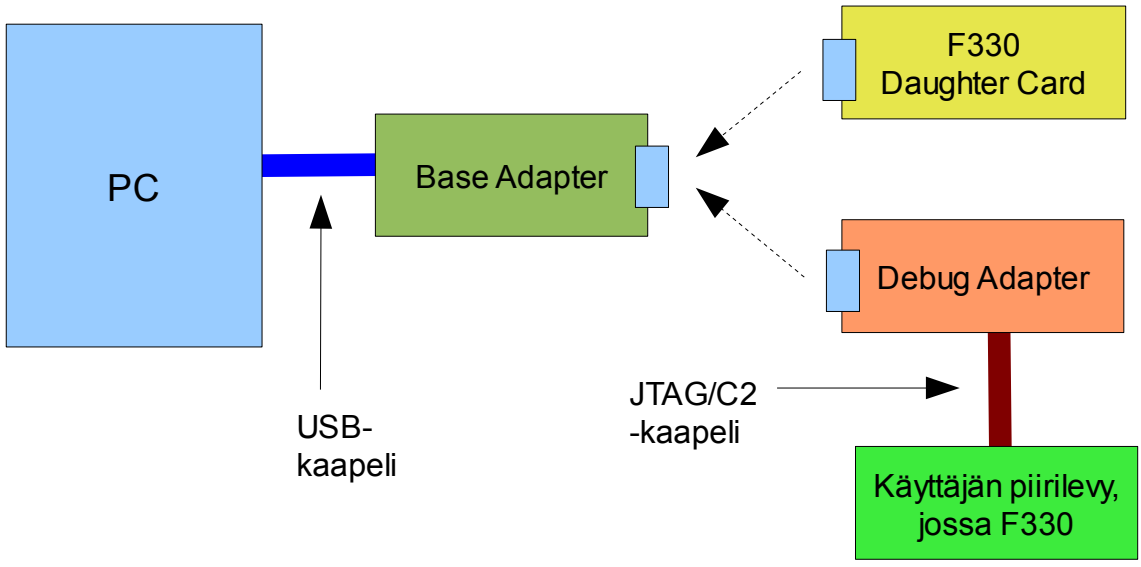
Fyysisesti ToolStick muodostuu joukosta pieniä koteloimattomia piirikortteja, joita voidaan kytkeä yhteen. Tässä ohjeessa tarkastellaan kolmea korttia:

- Base Adapter:
  - liittyy PC:hen USB-kaapelilla
  - toimii ”emolevynä” muille ToolStick-korteille

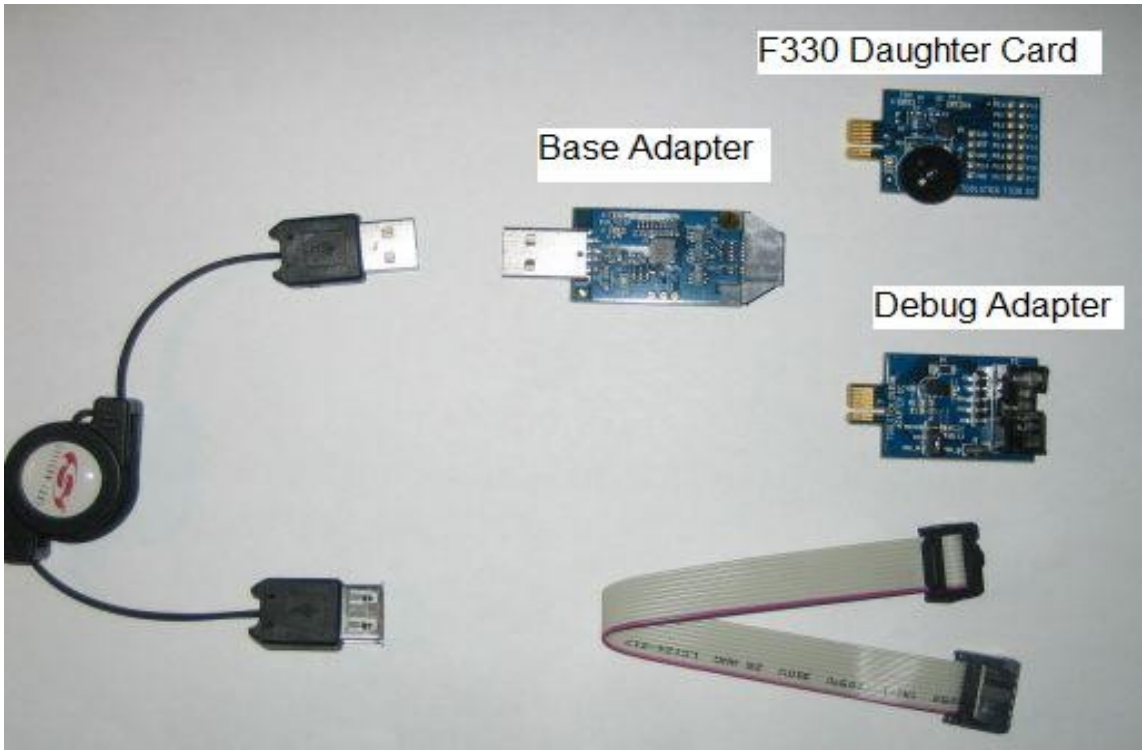
- F330 Daughter Card:
  - liitetään Base Adapteriin
  - kortilla olevaan F330-kontrolleriin voi ladata ohjelman PC:ltä, ajaa ja debugata sitä siellä
  - korvaa (joissakin tapauksissa) oman piirilevyn

- Debug Adapter:
  - liitetään Base Adapteriin
  - liittyy C2-liitännän kautta kehitettävälle piirilevylle (tukee myös JTAGia)
  - tämän kautta voi kehitettävällä piirilevyllä olevaan F330-kontrolleriin ladata ohjelman PC:ltä, ajaa ja debugata sitä siellä

Alla olevassa kuvassa näkyy nämä samat osat. Daughter Card ja Debug Adapter ovat siis vaihtoehtoisia, niistä jompikumpi liitetään Base Adapteriin sen mukaan, mitä halutaan tehdä (testata oman piirilevyn kanssa vai ilman sitä). Korttien liittimissä on koodaushahlo, joten niitä ei voi työntää toisiinsa väärinpäin.



Korttien fyysinen olemus näyttää tältä:



Jokaisesta kortista on myös käyttöohje, joka on ladattavissa SiLabsin sivuilta:

https://www.silabs.com/Support%20Documents/TechnicalDocs/ToolStick_BaseAdapter_UG.pdf

https://www.silabs.com/Support%20Documents/TechnicalDocs/ToolStick_F330_DC_UG.pdf

https://www.silabs.com/Support%20Documents/TechnicalDocs/ToolStick_Debug_Adapter_UG.pdf

3.2 ToolStickin hankkiminen

ToolStickiä ei myydä Suomessa, joten se on helpointa tilata Digikeystä. Yllä kuvatun kokonaisuuden hankkimiseksi pitää tilata kaksi tuotetta:

- ToolStick Starter Kit (Digikeyn koodi 336-1348-ND), sisältää:
  - Base Adapter
  - F330 Daughter Card
  - USB-kaapeli

- ToolStick Debug Adapter (Digikeyn koodi 336-1347-ND), sisältää:
  - Debug Adapter
  - JTAG/C2-kaapeli

Tätä kirjoitettaessa (elokuu 2009) näiden yhteenlaskettu listahinta Digikeyn verkkokaupassa oli 28,16 € + alv. Kimppatilauksen (http://www.vvsoft.fi/dk/index.php) kautta hankittuna hinta veroineen ja toimituskuluineen jää alle 40 €, mikä on varsin vähän toimivasta kehitysympäristöstä. Kaikki tarvittavat ohjelmat ovat ilmaisia.

C8051F330-GP (DIP20-koteloinen malli) maksaa Digikeyssä 2,17 €/kpl + alv.

4 Ohjelmien asennus

4.1 ToolStick-laiteohjain

Ennen ohjelmien asennusta ja testailua kannattaa työntää F330 Daughter Card paikalleen Base Adapterin liittimeen ja kytkeä Base Adapter valmiiksi PC:n USB-porttiin.

ToolStick ei tarvitse mitään erityistä laiteohjainta, se näkyy Windowsille USB HID-laitteena (Human Interface Device). Kun Base Adapter kytketään USB-porttiin, Windows asentaa tälle automaattisesti oletusohjaimen. Kannattaa seurata PC:n näytöltä, että tämä onnistuu.

4.2 Kehitysympäristön ohjelmat

Kehitysympäristöön kuuluu ToolStickin lisäksi joukko PC:hen asennettavia ohjelmia. Kaikki tarvittavat ohjelmat ovat ladattavissa vapaasti SiLabsin web-sivuilta, ei vaadi rekisteröitymistä sivuille.

Ohjelmat ovat saatavilla ainoastaan Windowsiin, vaatimuksena Windows 2000 tai uudempi. Omat kokemukset rajoittuvat Vistaan, jossa on toiminut hyvin.

Nämä ohjelmat tarvitaan:

- SiLabs IDE (Integrated Development Environment):

  Tämä on ohjelmiston ”perusohjelma”, jonka kautta kaikkea muuta käytetään. Se on normaali IDE-tyyppinen käyttöliittymä, jossa on editori ja projektien hallinta, ja siitä voi laukaista kääntäjän ja linkkerin. Sen kautta voi myös ladata ohjelmat kontrolleriin ja sen alaisuudessa suoritetaan debuggaus.

  Ladataan sivulta:
  https://www.silabs.com/products/mcu/Pages/SiliconLaboratoriesIDE.aspx

  Suora linkki:
  https://www.silabs.com/Support%20Documents/Software/mcu_ide.exe

- C- ja assembler-kääntäjä:

  8051:lle on saatavissa useita kaupallisia ja ilmaisia C- ja assembler-kääntäjiä. SiLabsin sivuilta on ladattavissa Keilin kaupallisen kääntäjän evaluointiversio.

Tämä kannattaa ladata siksi, että kaikki SiLabsin esimerkkiohjelmat noudattavat Keilin syntaksia. Ne on helppo konvertoida muillekin kääntäjille, mutta alussa on helpompaa, kun ei tarvitse huolehtia syntaksieroista.

Keil on ammattikäyttöön suunnattu kääntäjä, joten se on hinnaltaan turhan kova harrastajalle. Evaluointiversio on täysin toimiva, mutta siinä on 2 KB koodiraja, joten se kelpaa ensimmäisiin kokeiluihin, mutta ei juuri muuhun.

SDCC on ilmainen (open source) kääntäjä, joka tuottaa varsin laadukasta koodia. Kääntäjässä ei ole koodin kokorajoituksia tai vastaavia.

Muita kaupallisia 8051-kääntäjiä on useita: IAR, Raisonance, Hitech, jne. Lisäksi ilmaisia assembler-kääntäjiä löytyy useita. Myös Basic ja Pascal on saatavissa.

Keil ladataan sivulta:
https://www.silabs.com/products/mcu/Pages/ToolStick.aspx

Suora linkki:
https://www.silabs.com/Support%20Documents/Software/KeilV8Tools_Installer.zip

- ToolStick Development Tools:

  Joukko pieniä esimerkkiohjelmia SiLabsin eri kontrollereille, sekä joitakin apuohjelmia. Esimerkkisovellukset ovat hyödyllisiä, niistä pääsee näkemään, miten kontrollerin eri I/O-laitteita käytetään ohjelmassa.

  Ladataan sivulta:
  https://www.silabs.com/products/mcu/Pages/ToolStick.aspx

  Suora linkki:
  https://www.silabs.com/Support%20Documents/Software/ToolStick_Setup.exe

- Config Wizard:

  Ohjelma generoi valmiit kontrollerin alustusfunktiot (C:nä tai assemblynä) rasti ruutuun -periaatteella tehtyjen valintojen perusteella. Erittäin hyödyllinen ohjelma, kannattaa ottaa alusta saakka käyttöön. SiLabsin kontrollereissa on niin paljon erilaisia konfigurointeja, että niiden kaivaminen datalehdestä on työlästä. Config Wizard helpottaa huomattavasti.

  Ladataan sivulta:
  https://www.silabs.com/products/mcu/Pages/ConfigWizard.aspx

  Suora linkki:
  https://www.silabs.com/Support%20Documents/Software/ConfigAndConfig2Install.exe

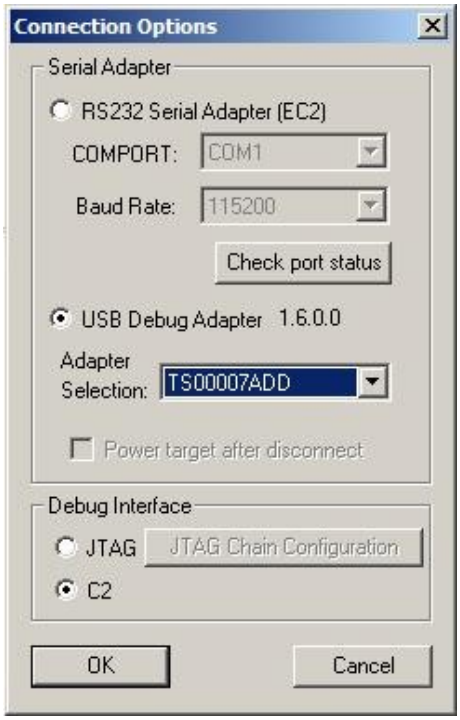
Zip-paketit sisältävät exe:n, jonka ajamalla paketti asentuu. Vastaavasti exet asentuvat ajamalle se.

5 Kehitysympäristön käyttöönotto

Kun ToolStick on kytketty USB-porttiin ja edellä luetellut ohjelmapaketit asennettu, on aika käynnistää IDE. Se käynnistyy projekti tyhjänä.

Aseta ensin käytettävän debug adapterin parametrit:

- Valitse valikosta *Options → Connection Options...* Aukeaa ikkuna:



- Aseta valinnat kuvassa näkyvällä tavalla. Kohdassa *Adapter Selection* näkyvä sarjanumero TSxxxxxx vaihtelee, ja se kertoo, että IDE näkee ToolStickin.

Kokeile seuraavaksi, saako IDE yhteyden Daughter Cardilla olevaan F330-kontrolleriin:

- Valitse valikosta *Debug → Connect* (tälle on myös painike työkalurivillä)
- IDE:n alareunan statusriville tulee kontrollerin tyyppi (*C8051F330*) ja kontrollerin ohjelman tila (*Halted*). Tämä kertoo, että IDE on saanut yhteyden Daughter Cardilla olevaan kontrolleriin.

IDE ja kontrolleri ovat nyt yhteydessä toisiinsa ja kehitysympäristö toiminnassa.

6 Ohjelman kääntäminen ja ajaminen

Ensimmäisenä kannattaa kääntää joku valmiista esimerkkiohjelmista, esim. ledin vilkutusohjelma Blinky.

Avaa projekti IDE:een:

- Valitse valikosta *Project → Open project...*
- Navigoi avautuneessa ikkunassa hakemistoon, johon asensit ToolStick Development Tools -paketin ja tämän alihakemistoon *MCU\Examples\C8051F330_5\Blinky*. Avaa täältä projektitiedosto *F330_Blinky_C.wsp.*
- C-koodin saa näkyviin editoriin kaksoisklikkaamalla vasemmassa reunassa olevassa projekti-ikkunassa tiedostoa F330_Blinky.c. Jos projekti-ikkuna ei ole näkyvillä, sen saa näkyviin valitsemalla *View → Project window*

Käännä ja linkitä projekti:

- Jotta näet kääntäjän tulosteet, avaa output-ikkuna valitsemalla valikosta *View → Output window,* ellei se ole jo auki.
- Valitse valikosta *Project → Rebuild project* (tälle on myös painike työkalupalkissa)
- Seuraa output-ikkunaan tulostuvia kääntäjän ja linkkerin tulosteita. Pitäisi päättyä ilmoitukseen:

  ```
  LINK/LOCATE RUN COMPLETE.  0 WARNING(S),  0 ERROR(S)
  ```

  → Meillä on nyt valmiiksi käännetty objektitiedosto PC:n levyllä.

Lataa ohjelma Daughter Cardin kontrolleriin:

- Tarkista alareunan tilariviltä, että IDE:llä on yhteys kontrolleriin. Jos ei ole, avaa se (*Debug → Connect).*
- Paina *Alt-D* (tai vastaavaa painiketta työkalupalkissa) ja seuraa, että output-ikkunaan tulostuu ”Download successful”.

  → Ohjelma on nyt Daughter Cardilla olevan kontrollerin Flash-muistissa, mutta ei vielä ajossa.

Käynnistä ohjelma kontrollerissa:

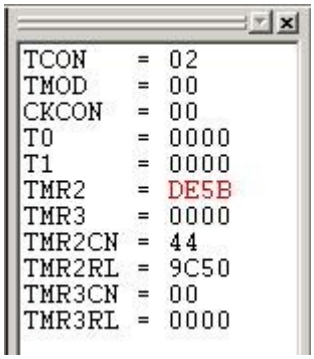
- Valitse valikosta *Debug* → *Go* (tai paina vastaavaa painiketta työkalupalkissa) jolloin ohjelma lähtee käyntiin.

  → Daughter Cardilla olevan vihreän ledin pitäisi nyt vilkkua tiheässä tahdissa.

- Ohjelman voi pysäyttää valitsemalla valikosta *Debug* → *Stop* (tai painamalla vastaavaa painiketta työkalupalkissa).

Esimerkki debuggaustoiminnoista:

- Kun ohjelma on pysäytettynä, valitse valikosta *View* → *Debug windows* → *SFR's* → *Timers*. Tällöin avautuu ikkuna, jossa näkyy kontrollerin ajastimet ja niiden ohjausrekisterit.



- Käynnistä ja pysäytä ohjelmaa muutaman kerran. Koska ohjelma käyttää timer 2:ta, niin aina kun ohjelma pysähtyy, TMR2-rekisterin muuttunut sisältö päivittyy ikkunaan. Muuttuneet arvot näkyvät punaisella.

7 SDCC:n käyttöönotto

Keilin kääntäjä kannattaa vaihtaa SDCC:hen jo aikaisessa vaiheessa. Koska Keilin evaluaatioversion koodin kokorajoitus on 2 KB, raja tulee vastaan nopeasti. ToolStickin pakkauksessa olevan koodin syöttämällä rajan saa nostettua 4 KB:hen, mutta se vain siirtää ongelmaa hiukan. SDCC:ssä ei ole mitään kokorajoitusta.

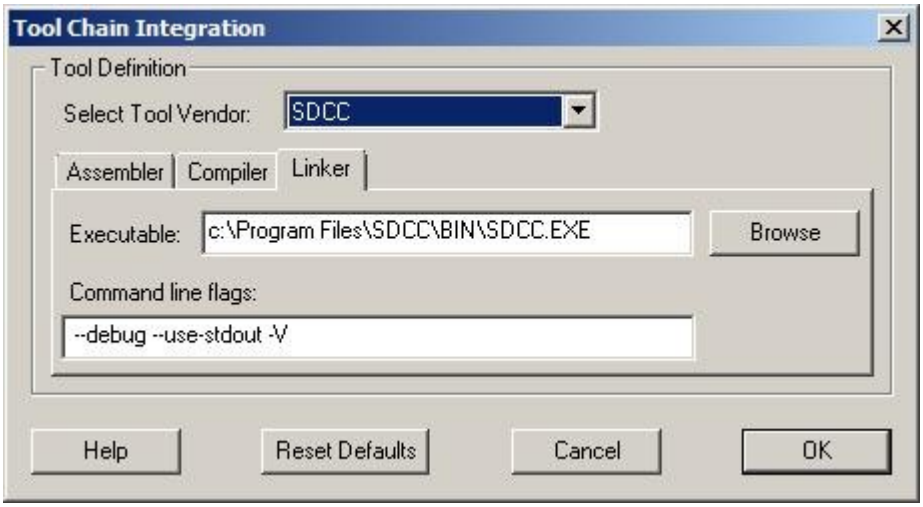
8051:n C-kielessä tarvitaan joitakin standardi-C:n ulkopuolisia laajennuksia, ja nämä ovat kääntäjäkohtaisia, eli asiat ilmaistaan hiukan eri syntaksilla Keilissä ja SDCC:ssä. Siksi on edullisempaa käyttää yhtä syntaksia alusta alkaen. Erot eivät ole isoja, mutta aiheuttavat turhaa editointitarvetta yritettäessä käyttää toiselle kääntäjälle tehtyä koodia.

SDCC:n voi ladata täältä:
http://sourceforge.net/projects/sdcc/files/

Se on saatavissa useisiin eri ympäristöihin, suora linkki Windows-versioon (tätä kirjoitettaessa uusin on 2.9.0):
http://sourceforge.net/projects/sdcc/files/sdcc-win32/2.9.0/sdcc-2.9.0-setup.exe/download

Kääntäjä asentuu ajamalla ladattu exe.

SiLabs IDE:ssä voidaan valita käytettävä kääntäjä. Tämä tieto tallentuu projektin mukana, joten eri projektit voivat haluttaessa käyttää eri kääntäjää. Valinta tehdään valitsemalla *Project → Tool Chain Integration...* Avautuvassa ikkunassa on pudotusvalikko, josta voi valita käytettävän kääntäjän:



Valitse SDCC, jolloin C-kääntäjän, assembler-kääntäjän ja linkkerin asetukset muuttuvat automaattisesti SDCC:lle sopiviksi. Näitä oletusarvoisia asetuksia voi haluttaessa muuttaa vastaavilla välilehdillä kirjoittamalla kohtaan *Command line flags* halutut optiot. Optiot näkee SDCC:n manuaalista (sdccman.pdf). Muutokset tallettuvat projektin mukana.

SDCC:ssä tulee valmiina h-tiedostot useimmille SiLabsin kontrollereille. Tiedostot löytyvät polusta *C:\Program Files\SDCC\include\mcs51*. F330-kontrollerin h-tiedosto on *C8051F330.h*.

Samaa asiaa on selostettu tarkemmin SiLabsin sovellusohjeessa ”AN198: Integrating SDCC 8051 Tools into the Silicon Labs IDE”. Suora linkki:

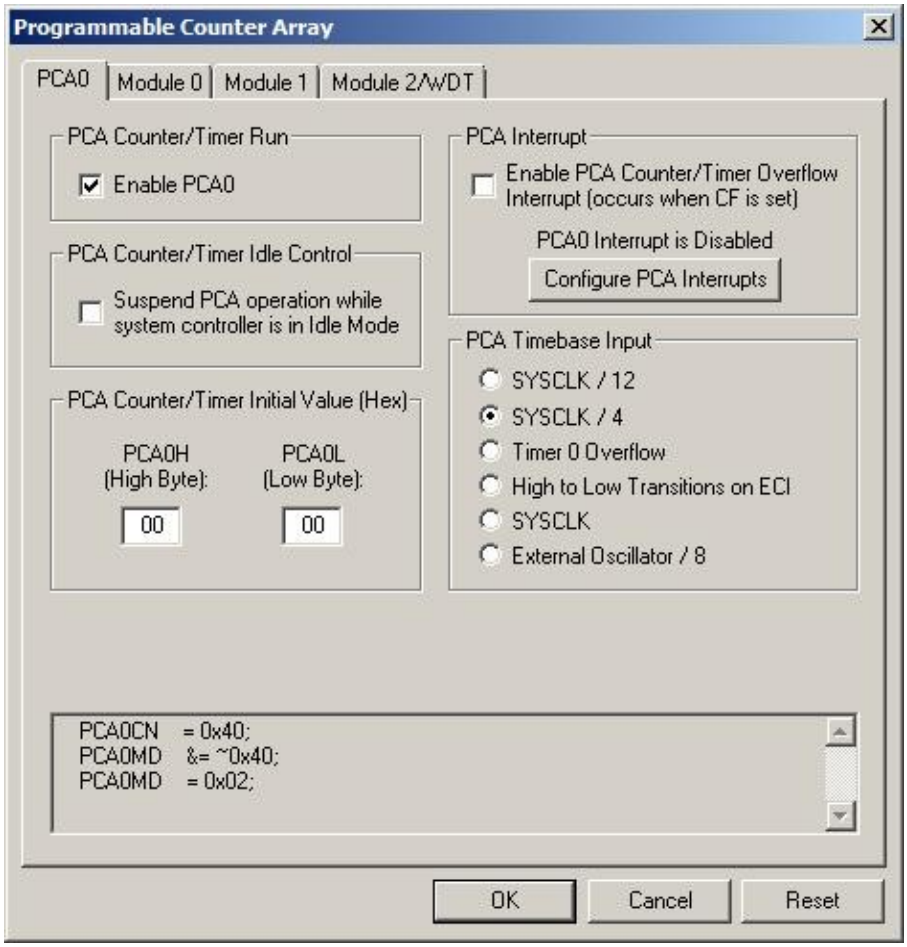
https://www.silabs.com/Support%20Documents/TechnicalDocs/an198.pdf

8 Config Wizardin käyttö

Kaikissa mikrokontrollereissa on joukko rekistereitä, jotka ohjaavat I/O-laitteiden toimintaa. Ohjelman alussa näille pitää asettaa sopivat arvot. SiLabsin kontrollereissa rekistereitä on huomattavasti enemmän kuin perus-8051:ssä, joten kunkin rekisterin alustusarvon muodostaminen vaatii tiivistä datalehden selailua. Config Wizard on ohjelma, jonka tarkoitus on helpottaa tätä vaihetta, ja siinä se onnistuu varsin hyvin. Se tuottaa rasti ruutuun -periaatteella tehtyjen valintojen perusteella valmiit alustusfunktiot, jotka voi ottaa mukaan omaan ohjelmaan.

Config Wizard on erillinen ohjelma, se ei toimi IDE:n alaisuudessa. Kun ohjelma käynnistyy, se esittää *New project* -ikkunan, josta voi valita halutun kontrollerityypin tai mennä avaamaan olemassa olevan projektin. Valitse *Device Family = C8051F330-5* ja *Part Number = C8051F330*.

Ohjelman pääikkunaan ilmestyy tyhjä C-kielinen moduuli. I/O-rekisterit konfiguroidaan valikon *Peripherals* alakohdista. Näistä jokaisesta avautuu ikkuna, jossa voi tehdä ko. I/O-laitteeseen liittyviä valintoja. Esim. PCA:



Kun tehdään valintoja, ikkunan alareunassa näkyvät rekisteriarvot muuttuvat vastaavasti. Painamalla OK tehdyt valinnat siirtyvät pääikkunassa näkyvään C-koodiin. Alussa valitun

kontrollerityypin perusteella ohjelma tietää, millaisia I/O-laitteita kussakin kontrollerityypissä on.

Kun kaikki valinnat on tehty, pääikkunaan on syntynyt joukko alustusfunktioita, joilla on yhteinen pääfunktio InitDevice(). Omassa koodissa, main-funktion alussa, pitää kutsua tätä funktiota.

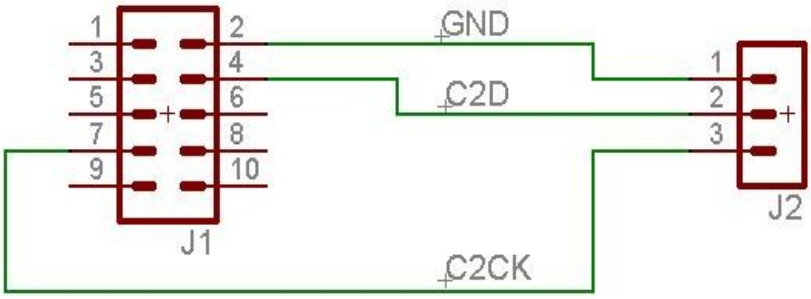
Tehdyt valinnat ja C-koodin voi tallettaa Config Wizardin projektiksi, tiedoston tyyppi .cwg. Yleensä kannattaa tehdä jokaista IDE:n projektia kohti vastaava Config Wizardin projekti.

Syntyvä C-koodi on rakennettu siten, että sen voi tallettaa sellaisenaan tiedostoon (*File → Save Source As...*) ja ottaa IDE:ssä mukaan projektiin omana moduulinaan. Toinen tapa on copy/pasteta koodi Config Wizardista omaan koodiin IDE:n editorissa. Tällöin generoitu #include-lause pitää jättää pois.

9 Käytännön ohjeita

Tähän kappaleeseen on koottu sekalainen joukko omiin kokemuksiin perustuvia käytännön vihjeitä eri tilanteisiin.

- Debug Adapterin mukana tulevan kaapelin molemmissa päissä on 10-napainen nauhakaapeliliitin, jossa on signaalit sekä C2- että JTAG-liitännälle. C2-liitäntä tarvitsee vain 3 signaalia, joten kannattaa tehdä alkuperäisen kaapelin tilalle kuvan mukainen kaapeli. Liitin J2 on 3-napaiseen piikkirimaan menevä pikkuliitin (esim. Harwin 1x3). Tämä vie huomattavasti vähemmän pinta-alaa piirilevyltä.



- IDE:ssä pitää muistaa tallettaa projekti. Työkalupalkin Save-painike tai *File → Save* tallettaa vain editorin tiedoston. Projekti tallentuu vain valinnalla *Project → Save Project*. Projektiin lisätyt tiedostot, auki olevat ikkunat, kääntäjän/linkkerin optiot ym. tallentuvat projektin mukana. Jos IDE kaatuu (minkä se tekee joskus) muutokset katoavat, jos projektia ei ole muistanut tallentaa.

- Useimmissa SiLabsin kontrollereissa, myös F330:ssa, yksi PCA:n moduuleista voi toimia vahtikoirana (WDT, Watchdog timer). Vahtikoiramoodi on aktiivinen resetin jälkeen, joten se pitää muistaa poistaa käytöstä mahdollisimman aikaisin ohjelmassa, ennen kuin se ehtii laueta. Jos ohjelma toimii epämääräisesti tai ei ollenkaan, saattaa olla vaikea huomata, että se on vahtikoira, joka resetoi kontrolleria vähän väliä. Vahtikoira poistetaan toiminnasta nollaamalla PCA0MD-rekisterin bitti 6 (WDTE). Käytännössä main()-funktion alkuun, jo ennen hardwaren alustusta, kannattaa sijoittaa lause

```
PCA0MD &= ~0x40;    // WDTE = 0
```

- Erityisesti sellaiselle, joka on ohjelmoinut muiden valmistajien 8051-kontrollereilla, saattaa olla yllätys, että I/O-portit eivät toimi lähtöinä, jos crossbar ei ole aktiivinen. Tuloina ne kyllä toimivat. Crossbar on resetin jälkeen oletuksena disabloitu. Se enabloidaan XBR1-rekisterin bitillä 6 (XBARE):

```
XBR1 |= 0x40;       // XBARE = 1
```