



CY8CKIT-001 PSoC Development Kit- kehitysalusta

Juha Vartiainen

Opinnäytetyö
Toukokuu 2013
Tietotekniikan ko.
Sulautetut järjestelmät ja
elektroniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Sulautetut järjestelmät ja elektroniikka

Juha Vartiainen
CY8CKIT-001 PSoC Development Kit-kehitysalusta

Opinnäytetyö 51 sivua, joista liitteitä 24 sivua
Toukokuu 2013

CY8CKIT-001 PSoC Development Kit on Cypress Semiconductorsin valmistama kehitysalusta, jonka tarkoituksena on ollut toimia prototyypialustana PSoC-arkkitehtuuria käyttävien sovellusten suunnitteluun ja testaukseen. Kehitysalustalla olevia monipuolisia liityntöjä hyödyntämällä voidaan toteuttaa erilaisia sulautettujen järjestelmien harjoitustöitä.

Työn päämääränä oli kehitysalustan ominaisuuksien ja käyttökohteiden kartoittamisen lisäksi asennus- ja käyttöönotto-ohjeistuksen laatiminen. Ohjeiden osalta pyrittiin laatimaan mahdollisimman selkeä ja helppolukuinen käyttöohje, jota voitaisiin käyttää sulautettujen järjestelmien opetuskäytössä laitteeseen tutustumisen apuna. Ohjeiden tekemisen ohessa etsittiin ratkaisuja yleisimpiin sovelluskehityksessä ilmaantuneisiin virhetilanteisiin, joita oli havaittu aikaisemmillä laboratoriotutetuksilla.

Työssä tutustutaan tarkemmin PSoC 3-sarjaan kuuluvaan CY8C38-prosessorikorttiin toteuttamalla sillä Commodore 64-peliohjaimen kytkentä PC-tietokoneeseen USB-väylän kautta. Lisäksi työssä tutustuttiin USB-väylään niiltä osin, mitä työn suorittamisen kannalta oli tarpeen.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Embedded Systems & Electronics

Juha Vartiainen:
CY8CKIT-001 PSoC Development Kit

Bachelor's thesis 51 pages, appendices 24 pages
May 2013

CY8CKIT-001 PSoC Development Kit is a development board designed by Cypress Semiconductors. The board provides a common platform for prototyping and evaluating different solutions for the PSoC 1, PSoC 3 and PSoC 5 family architectures.

The purpose of this study was to describe the essential components and uses of the board and to introduce the main features of the PSoC 3 architecture. Furthermore a quick start guide was written to be used as a helpful tool in laboratory use. This also included solutions for several error situations that had been reported previously.

As a demonstration an example application was designed and programmed using the bundled CY8C38 processor where a Commodore 64 era joystick was connected via the USB bus to a host PC to be used as a HID device.

Key words: PSoC, USB, C64, HID

SISÄLLYS

1	JOHDANTO.....	6
2	SULAUTETUT JÄRJESTELMÄT	7
3	PSOC-JÄRJESTELMÄT	8
3.1	CY8CKIT-001 PSoC Development Kit-kehitysalusta	8
3.2	Tekniset ominaisuudet	10
3.3	CY8C38-prosessorikortti	12
3.3.1	Proessori	12
3.3.2	Muistipiirit	13
3.3.3	Tahdistussignaalit.....	14
3.3.4	I/O-liitännät	15
3.3.5	Alijärjestelmät	15
4	OHJELMOINTIYMPÄRISTÖ	17
5	USB-VÄYLÄ	18
6	PELIOHJAINADAPTERIN TOTEUTUS	19
7	POHDINTA.....	25
	LÄHTEET	26
	LIITTEET	27
	Liite 1. PSoC Development Kit asennus- ja käyttöohje.....	27

LYHENTEET JA TERMIT

ADC	Analogia-digitaalimuunnin, laite joka muuntaa jatkuvan analogisen signaalin arvoja digitaalisiksi lukuarvoiksi
CAN	Controller Area Network, automaatioväylä, jota käytetään mm. koneissa ja teollisuusajoneuvoissa
DAC	Digital to analog converter, laite jolla muutetaan digitaalinen signaali vastaavanlaiseksi analogiseksi signaaliksi
DSI	Digital System Interconnect, digitaalialijärjestelmän reitittäjä
ECC	Error Correcting Code, virhekorjauskoodi
Flash	Puolijohdemuisti, jota voidaan tyhjentää ja uudelleenohjelmoida sähköisesti
GPIO	General Purpose Input/Output, yleiskäyttöinen portti
HDL	Laitteistonkuvauskieli
I ² C	Yksinkertainen kaksisuuntainen ohjaus- ja tiedonsiirtoväylä
LUT	Lookup table, taulukko, josta voidaan lukea arvoja suoraan niiden laskemisen sijaan
rekisteri	CPU:n ytimessä oleva 8-, 16-, tai 32-bittinen datavarasto
SAR ADC	Successive Approximation ADC, muuttaa jatkuvan analogisen aaltomuodon diskreettiin digitaaliseen muotoon
SC/CT	Switched capacitor/continuous time, integroitu piiri, jossa emuloidaan vastuksia kytkemällä nopeasti kapasitanseja kahden eri pisteen välillä
SRAM	Static random access memory, muistiyksikkö joka mahdollistaa datan tallettamisen ja haun suurilla nopeuksilla. Tallettaessa data säilyy muuttumattomana, ellei sitä tarkoituksellisesti muuteta tai kunnes virta katkaistaan
UDB	Universal Digital Block, PSoC-digitaalijärjestelmän keskeisin osa, jolla luodaan järjestelmässä käytetyt digitaalifunktiot

1 JOHDANTO

PSoC (Programmable System-on-Chip) on yleisnimitys Cypress Semiconductorsin valmistamasta sarjasta yhdelle mikropiirille mahtuvia järjestelmiä, joissa on CPU:n lisäksi muokattavat analogi- ja digitaalilohkot eri signaaleille. Työtä varten tutustuttiin PSoC-järjestelmän ohjelmoimiseen CY8CKIT-001 PSoC Development Kit- kehitysalustalla, joka sisältää useita prototyypikomponentteja sekä tietoliikennöintiä. Esimerkkisovellus ohjelmoitiin PSoC 3-sarjaa olevalle CY8C38-prosessorikortille, joka toimitettiin kehitysalustan mukana. Työn aihe ja kehitysalusta saatiin yliopettaja Mauri Inhalla ja työ suoritettiin Tampereen Ammattikorkeakoulun laboratoriotiloissa. Työn tavoitteina oli laatia kehitysalustan asennus- ja käyttöönotto-opas sulautettujen järjestelmien laboratoriotyöt-kurssia varten sekä ratkoa alustan käytössä aiemmin ilmenneitä yleisimpiä ongelma- ja vikatilanteita.

Työssä esitellään kehitysalustan osat, PSoC-järjestelmien tekniset tiedot yleisellä tasolla ja lisäksi tutkitaan käytetyn prosessorin ominaisuuksia. Lisäksi esitellään esimerkkisovelluksen toteutuksessa käytetty USB-väylä lyhyesti sekä kuvaillaan sovelluksen toimintaperiaate.

2 SULAUTETUT JÄRJESTELMÄT

Nykyihminen on tietämättäänkin tekemisissä useiden sulautettujen järjestelmien kanssa päivittäisessä elämässään. Kun eväitä lämmitetään mikroaaltouunissa, astioita sekä vaatteita pestään pesukoneissa ja kodin viihde-elektroniikkaa ohjataan kaukosäätimellä, tuskin kovinkaan moni tulee ajatelleeksi käyttäneensä juuri mikrotietokonetta. Sulautetuksi järjestelmäksi kutsutaankin sellaista laitetta, josta ei ulkoapäin välttämättä huomaa sen olevan tietokone. Sulautettu järjestelmä koostuu mikro-ohjaimesta, muistipiireistä ohjelmoidulle ohjelmalle ja käsiteltävälle tiedolle sekä oheisliitännöistä ulkomaailmaan. Sulautetun järjestelmän päätehtävänä on valvoa ja ohjata sen elektronisen laitteen toimintoja, johon se on integroitu. Esimerkiksi pesukoneessa mikro-ohjaimen tehtävänä voi olla altaan vesimäärän tutkiminen sekä pesurummun moottorin kierrosnopeuden säätäminen. (Vahtera 2003, 13-14.)

Yksinkertaisimmillaan sulautetun järjestelmän ohjelmoimiseen tarvitaan PC-tietokone, johon on asennettu ohjelmien kehitysympäristö tekstieditoreineen ja kääntäjineen. Lisäksi tarvitaan mikro-ohjain, jolle ohjelmat kehitetään sekä kaapelointi PC:n ja mikro-ohjaimen välille. (Vahtera 2003, 15.) Vaivattomin tapa on hankkia jokin valmis kehitysalusta. Jokaisella mikro-ohjainvalmistajalla on omat kehitysalustansa, joten valinnan varaa on sadoista erilaisista malleista. Alustojen hinta riippuu niiden sisältämistä ominaisuuksista ja yksinkertaisimman harrastelijatason kehitysalustan voi saada alle kymmenellä eurolla. (Jämsä, 2011.)

3 PSOC-JÄRJESTELMÄT

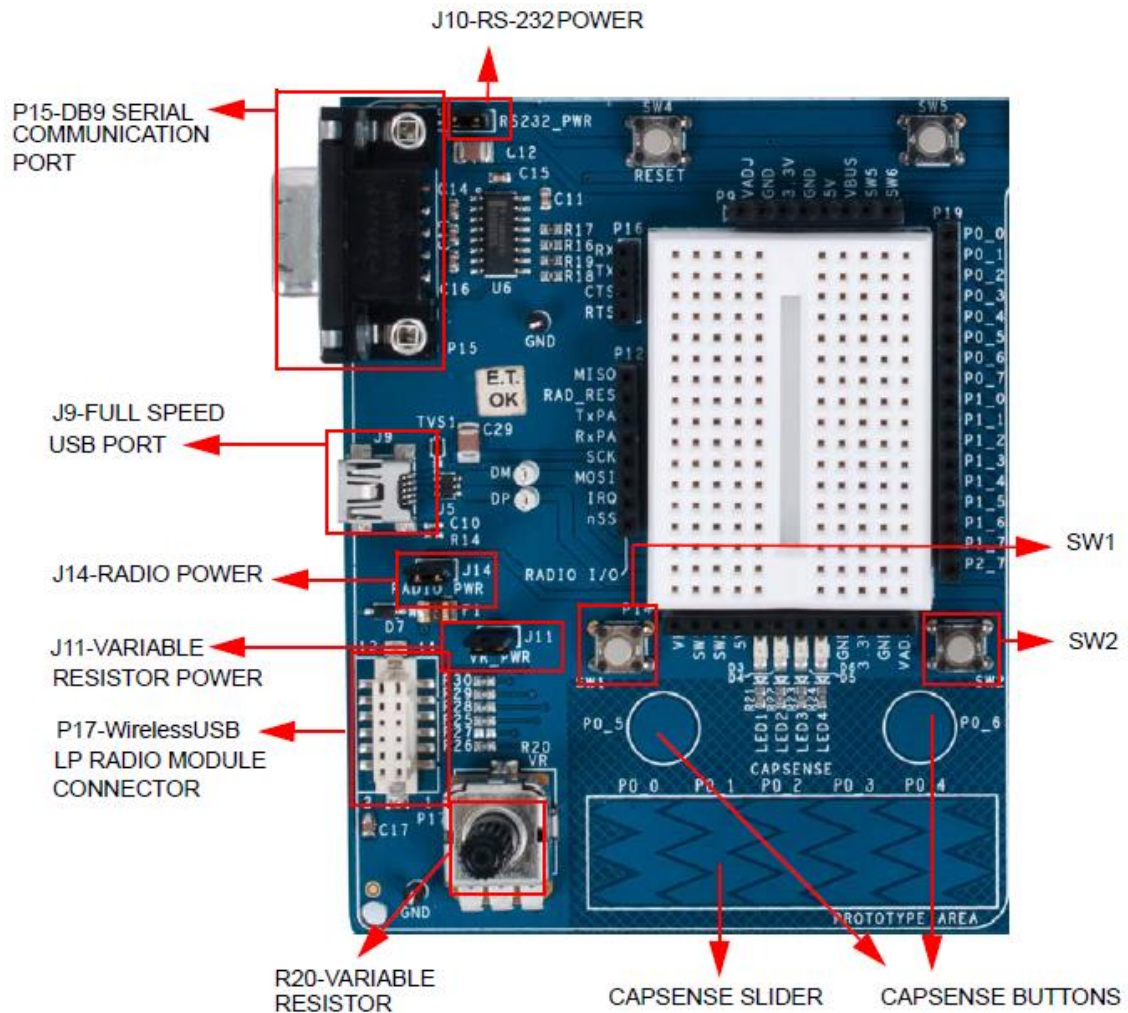
3.1 CY8CKIT-001 PSoC Development Kit-kehitysalusta

CY8CKIT-001 PSoC Development Kit on Cypress Semiconductorsin valmistama kehitysalusta, jolla voidaan tutustua sovellusten kehittämiseen eri PSoC-järjestelmille. Kehitysalusta tukee PSoC 1-, PSoC 3-, sekä PSoC 5-arkkitehtuuria käyttäviä prosessorikortteja. (CY8CKIT-001 PSoC Development Kit Guide 2012, 7.) Kuvassa 1 on esitetty kehitysalusta, siihen asennettu CY8C38-sarjan prosessorikortti, sekä ohjelmointiyksikkö MiniProg3. PSoC itsessään on nimitys yksittäiselle mikropiirille mahtuvasta ohjelmoitavasta järjestelmästä, joka sisältää mikro-ohjaimen, muistin ja muokattavissa olevat digitaaliset sekä analogiset alijärjestelmät. PSoC:n etuna muihin mikro-ohjaimiin on mm. sen alijärjestelmien helppokäyttöisyys. Lisäksi ohjelmointiympäristössä voidaan asettaa useita parametrejä graafisesti, jolloin säästyy aikaa koodin kirjoittamiselta. (PSoC 3 CY8C38 Family Datasheet 2012, 1.)



KUVA 1. CY8CKIT-001 PSoC Development Kit käyttövalmiina

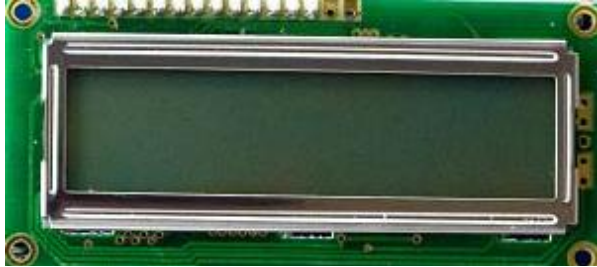
Kehitysalusta sisältää useita kehitystyössä hyödyllisiä komponentteja ja liityntöjä, joista tärkeimmät esitellään tässä luvussa. Kuvassa 2 on esitetty lähikuva prototyypiliitännöistä selityksineen.



KUVA 2. Prototyypiliitännät (CY8CKIT-001 PSoC Development Kit Guide 2012, 177)

Kehitysalustalta löytyy koekytkeäalustan lisäksi mini-B-tyyppinen USB-liitäntä, 9-nastainen D-liitin eli sarjaportti sekä liitin Artaflex AWP24S WirelessUSB-modulille. Painikekytkimet SW1, SW2, SW5 ja SW6, säädettävä 10 k Ω :n vastus R20 sekä neljä pintaliitosledyä voidaan kytkeä hyppylangoilla koekytkeäalustan ympärillä olevista terminaleista kytkeäalustan oikealla puolella oleviin I/O-liitäntöihin. Näitä liitäntöjä on 17 kappaletta (P0_0 - P2_7) ja ne voidaan asettaa käyttäjän haluamasti analogi- tai digitaali-irräntää varten. Lisäksi alustalla on neljä GPIO-laajennusporttia lisäkorttien asennusta varten. Kapasitiivista kosketusta hyödyntäviä CapSense-elementtejä on 3 kappaletta, joiden viisiosaista liikusäädintä sekä kahta painonappia voidaan hyödyntää

useissa kosketusta mittaavissa sovelluksissa. Alustalla on lisäksi kuvan 3 mukainen Hitachi HD44780-standardia käyttävä 2x16 merkin LCD-moduli. (CY8CKIT-001 PSoC Development Kit Guide 2012, 11, 178-180.)



KUVA 3. LCD-moduli (CY8CKIT-001 PSoC Development Kit Guide 2012, 179)

3.2 Tekniset ominaisuudet

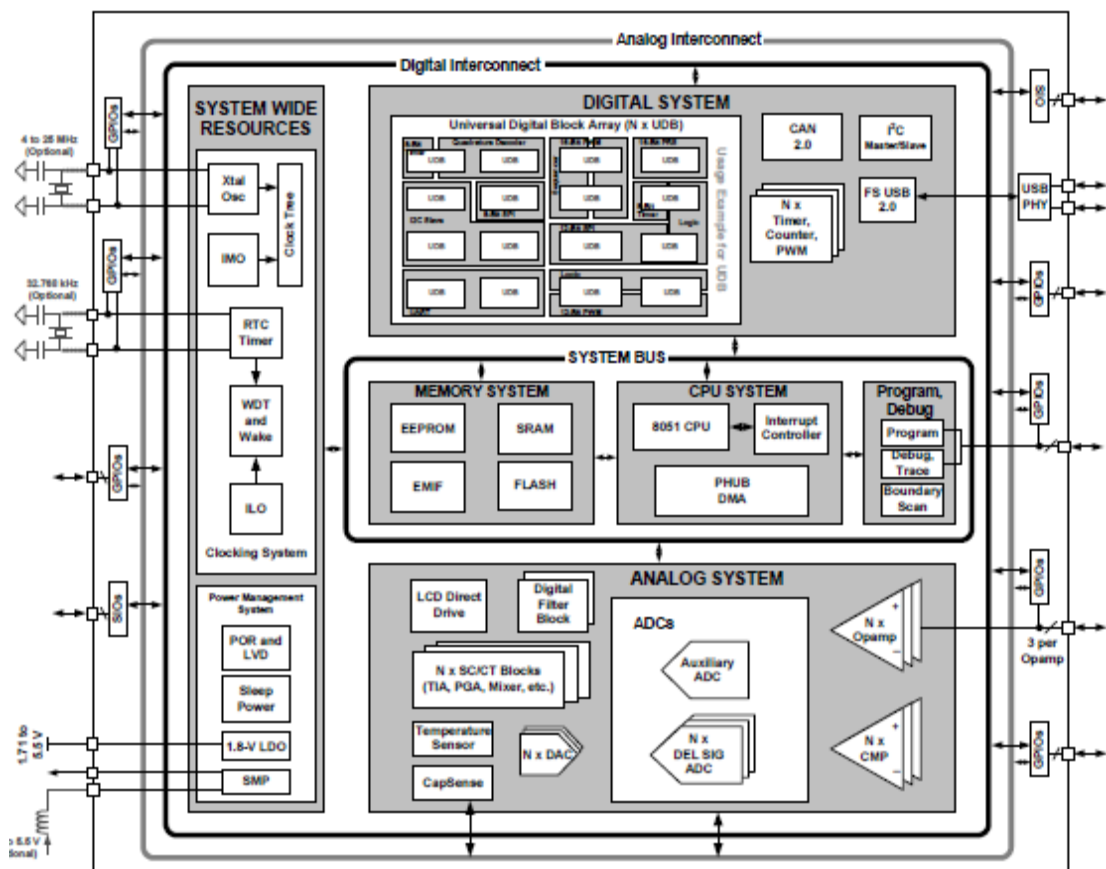
PSoC-piirit voidaan jakaa kolmeen sarjaan: PSoC 1, PSoC 3 ja PSoC 5, joille kaikille voidaan kehittää sovelluksia tässä työssä käytössä olleella kehitysalustalla. Yhteen sarjaan kuuluu useampia eri tuotteita, jotka eroavat toisistaan lähinnä muistin suuruudessa. (CY8CKIT-001 PSoC Development Guide 2012, 7.) Järjestelmien tärkeimmät ominaisuudet ja erot on esitetty taulukossa 1.

TAULUKKO 1. Järjestelmien tekniset ominaisuudet (CY8CKIT-001 PSoC Development Guide 2012, 7)

	PSoC 1	PSoC 3	PSoC 5
Prosessori	8-bittinen M8C	8-bittinen 8051	32-bittinen ARM Cortex M3
Ohjelmamuisti (Flash)	4 kB – 32 kB	8 kB - 64 kB	32 kB - 256 kB
Käyttömuisti (SRAM)	256 B – 2 kB	2 kB - 8 kB	8 kB - 64 kB
ADC	1 Delta-Sigma (6-14 bit)	1 Delta-Sigma ADC (12-20-bit)	1 Delta-Sigma ADC (12-20 bit) 2 SAR ADC:tä (8-12 bit)
DAC	6-8 bit	8-12 bit	8- 12 bit
Digitaalilohkot	16 digitaalista lohkoa	24 UDB:tä	24 UDB:tä
Virrankulutus	Aktiivitila: 2mA Lepotila: 3 μ A	Aktiivitila: 1,2 mA Lepotila: 1 μ A Horrostila: 200 nA	Aktiivitila: 2 mA Lepotila: 2 μ A Horrostila: 300 nA

3.3 CY8C38-prosessorikortti

Tässä luvussa esitellään PSoC3-arkkitehtuuria käyttävän CY8C38-prosessorikortin tärkeimmät osat, joihin kuuluvat prosessori, muistielementit sekä ohjelmoitavat siirräntäportit. Lisäksi tarkastellaan analogi- ja digitaaliosien toimintaa. PSoC 3-piirin yksinkertaistettu lohkokaavio on esitetty kuvassa 4.



KUVA 4: PSoC3-piirin lohkokaavio (PSoC 3 Architecture Technical Reference Manual, 22)

3.3.1 Prosessori

Mikroprosessori on mikrotietokoneen keskusyksikkö, joka suorittaa ohjelmamuistiin tallennettua ohjelmaa. CPU hakee käskyn ohjelmamuistista, tulkitsee sen ja tekee käskyn määrämän toimenpiteen, kuten LCD-näytön ohjauksen. (Vahtera 2003, 16.)

PSoC 3-sarjan ytimenä käytetään 8-bittistä 8051-prosessoria, joka perustuu Intelin vuonna 1981 julkaisemaan mikroprosessoriin. CY8C38-sarjassa käytetty prosessori suorittaa parannusten ansiosta käskyjä 10 kertaa nopeammin kuin alkuperäinen 8051. Parannuksiin kuuluu myös lisätty 256 tavun sisäinen data-RAM. Näiden ohella ydin on täysin yhteensopiva 8051-standardin assembly-käskykannan kanssa. (PSoC 3 Architecture Technical Reference Manual 2012, 35)

Laitteen prosessori ja muut elementit on sidottu toisiinsa PHUB:n avulla, joka koostuu kahdeksasta eri lohkoihin yhdistetystä pinnasta. Jokainen pinna on yhdistetty yhteen tai useampaan lohkoon. PHUB:ssa on myös usean yhtäaikaisen DMA-kanavan tiedonsiirron mahdollistava DMAC-ohjain. Prosessorin ja DMAC-ohjaimen aiheuttamat luku- ja kirjoituspyynnöt hoidetaan PHUB:n kautta, jolloin PHUB:lla käsitellään millä pinnalla kutakin elementtiä käytetään. Mikäli prosessorilla ja DMAC-ohjaimella aloitetaan siirto yhtäaikaan samaa pinnaa käyttäen, PHUB päättää kummalle annetaan prioriteetti. Prioriteetti voidaan asettaa muuttamalla PHUB_CFG-rekisterin `spk_cpu_pri` -bittejä, lukuunottamatta pinnaa 0 joka on varattu ainoastaan DMAC-ohjaimelle SRAM-muistin suoraa käyttöä varten. Myös prosessorilla on suora väylä SRAM-muistiin. (PSoC 3 Architecture Technical Reference Manual 2012, 71)

3.3.2 Muistipiirit

Valmis tietokoneohjelma pitää tallentaa johonkin, jossa se on prosessorin saatavilla. Lisäksi käsiteltävä ja jo käsitelty data täytyy pitää tallessa. Sulautettujen järjestelmien ollessa pienikokoisia sekä fyysisesti että ohjelmallisesti, ei yleensä ole tarpeen käyttää isokokoisia magneettisia levyasemia. Sen sijaan muistipiireinä käytetään puolijohde-muisteja. Puolijohdemuisti on numeroita sähköisesti tallentava elektroninen piiri, ideaalisesti fyysiseltä kooltaan pieni, mutta muistikapasiteetiltaan suuri. (Vahtera 2003, 24-25.)

PSoC:n ohjelmamuistina on käytetty uudelleenohjelmoitavaa Flash-muistia, jota on 64 kilotavun verran. Lisäksi ylimääräiset 8 kilotavua on varattu ECC-virheenkorjausta varten. ECC:llä voidaan korjata yhden bitin virhe ja sillä voidaan lisäksi havaita kahden bitin virhe per 8 bittiä, jolloin virheen tapahtuessa voidaan aiheuttaa esimerkiksi keskeytyskutsu prosessorille.

Flash-muisti koostuu yhdestä lohkokosta, joka on jaettu riveihin. Kussakin rivissä on varattuna 256 tavua ohjelmadataalle ja 32 tavua virheenkorjausta varten. Mikäli ECC:tä ei käytetä, se voidaan hyödyntää datan varastointiin. 64 kilotavun muistissa näitä lohkoja on yhteensä 256 kappaletta. Ohjelmamuisti on myöskin suojattavissa, mikäli sen sisältö halutaan salata. (PSoC 3 Architecture Technical Reference Manual 2012, 106)

CY8C38-järjestelmässä on väliaikaista dataa varten 8 kilotavua staattista RAM-muistia, joka on jaoteltu neljän kilotavun lohkoihin. Muistia voidaan käsitellä suoraan prosessorilta 8-bittisellä väylällä tai PHUB:in kautta 32-bittisellä väylällä. Koska muisti on fyysisesti eri lohkoissa, prosessorilla ja PHUB:illa voidaan käsitellä eri lohkoja yhtäaikaaisesti. SRAM:in sisältämä data säilyy kun laitetta käytetään vähävirta- tai horrostilassa. (PSoC 3 Architecture Technical Reference Manual 2012, 103).

3.3.3 Tahdistussignaalit

Kellosignaalina voidaan käyttää useampaa vaihtoehtoa riippuen käyttötarkoituksesta: sisäinen pääoskillaattori IMO, sisäinen hidas värähtelijä ILO, ulkoinen kideoskillaattori MHzECO sekä 32 kilohertsin ulkoinen kideoskillaattori kHzECO, joiden lisäksi vaihelukittua silmukkaa (PLL) ja 16-bittisiä jakajia käyttäen voidaan generoida käyttäjän haluamia kellotaajuuksia. Oskillaattorien minimi- ja maksimiarvot on esitelty taulukossa 2.

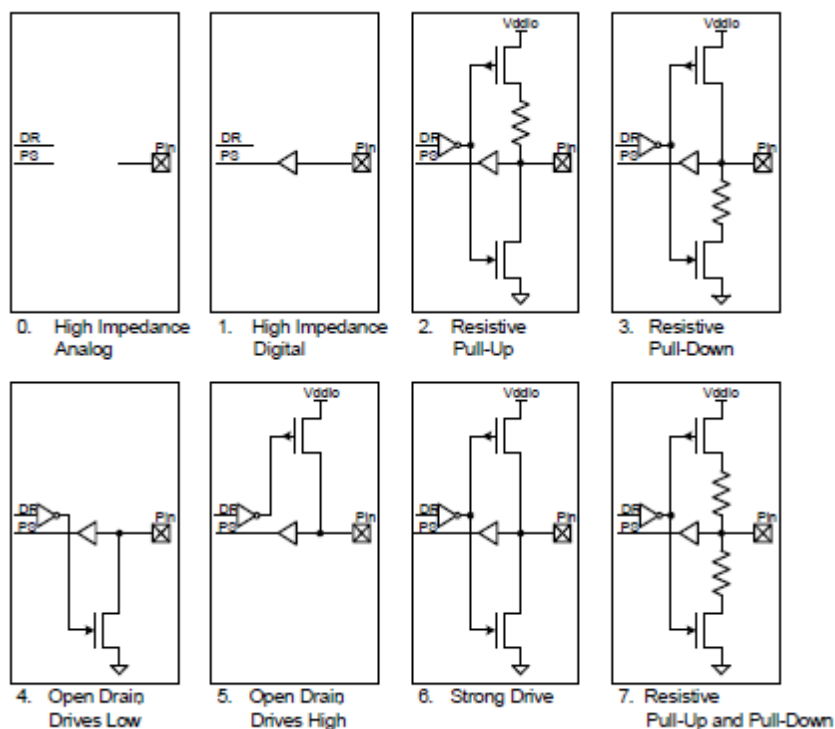
TAULUKKO 2. Oskillaattorien yhteenveto (PSoC Architecture Technical Reference Manual 2012, 125)

Lähde	F _{min}	F _{max}
IMO	3 MHz	62,6 MHz
ILO	1 kHz	100 kHz
MHzECO	4 MHz	25 MHz
kHzECO	32,768 MHz	32,768 MHz
PLL	24 MHz	67 MHz

3.3.4 I/O-liitännät

PSoC-laitteiden kaikki GPIO-siirräntäportit ovat prosessorin ja oheislaiteliitännöiden käytävissä sekä sisään- että ulostuloina. Lisäksi kaikilla nastoilla voidaan aiheuttaa keskeytyskutsu. Jokainen nasta voidaan asettaa yksittäisesti johonkin kuvassa 5 olevasta kahdeksasta tilasta käyttötilanteen mukaan. Tila 0 on valittuna oletuksena, jolloin digitaalinen sisääntulopuskuri on pois päältä. Tilaa käytetään kun halutaan, että nasta on kelluvassa tilassa tai kun käytetään analogista jännitettä. Tilaa 1 käytetään sisääntuleville digitaalisignaaleille. Tiloissa 2 ja 3 on sarjaanvedetty alas tai ylös vetovastus ja tiloja käytetään digitaalisessa I/O-toiminnassa, kuten mekaanisten kytkinten kanssa. Tiloja 4 ja 5 käytetään yleisesti I²C-väylän signaalien ohjaukseen. Tilalla 6 saadaan joko ylhäällä tai alhaalla oleva digitaalinen CMOS-ulostulo. Tila 7 on kuten tilat 2 ja 3, paitsi nasta on aina sarjassa vastuksen kanssa (PSoC CY8C38 Family Datasheet 2012, 39).

KUVA 5. GPIO-tilat (PSoC 3 CY8C38 Family Datasheet 2012, 38)

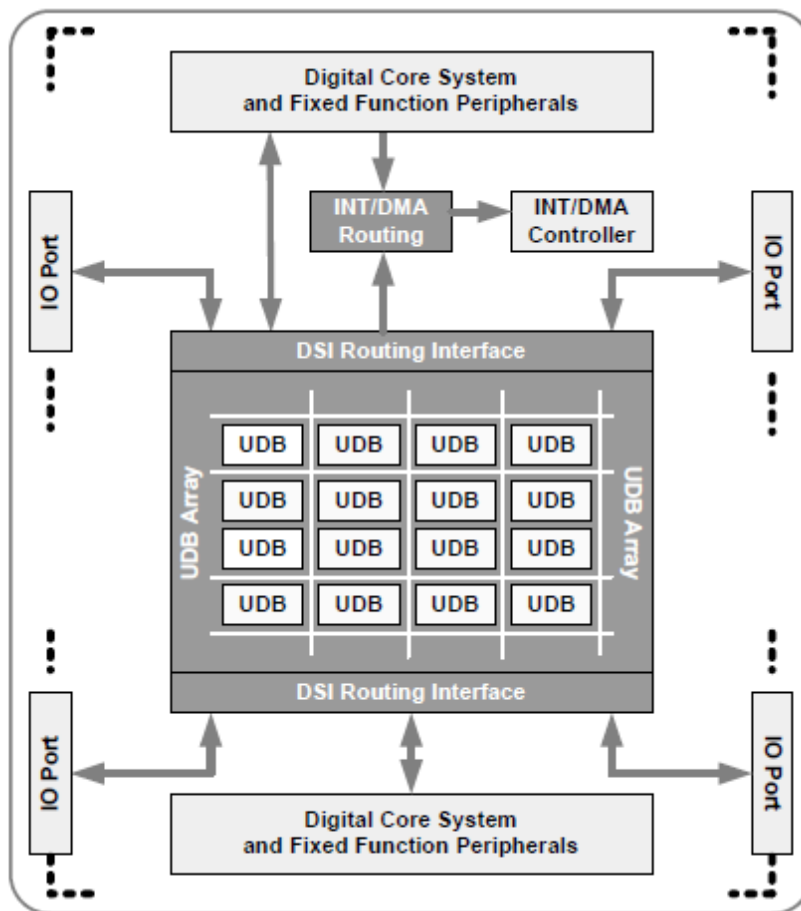


3.3.5 Alijärjestelmät

Digitaalinen alijärjestelmä sisältää useita UDB-lohkoja, joissa jokaisessa on yhdistelmä sekä ohjelmoitavaa että aritmeettista logiikkaa. UDB-lohkoilla voidaan toteuttaa funkti-

oita yksinkertaisista logiikkaporteista ajastin/laskureihin ja tiedonsiirtoon. Ohjelmointiympäristössä on valmiina useita yleisimpiä komponentteja, joita käyttämällä digitaalisen alijärjestelmän rekisteritason toiminnasta ei tarvitse olla perillä käyttäkseen sitä. Alijärjestelmään voi myös kehittää omia toimintoja HDL-kielellä (esim. Verilog). (PSoC 3 Family Datasheet 2012, 42.)

Oheislaitteiden, I/O-nastojen, DMA:n, UDB-lohkojen sekä muut järjestelmän väliset digitaaliset signaalit reititetään DSI:n kautta, kuten kuvassa 6 on esitetty.



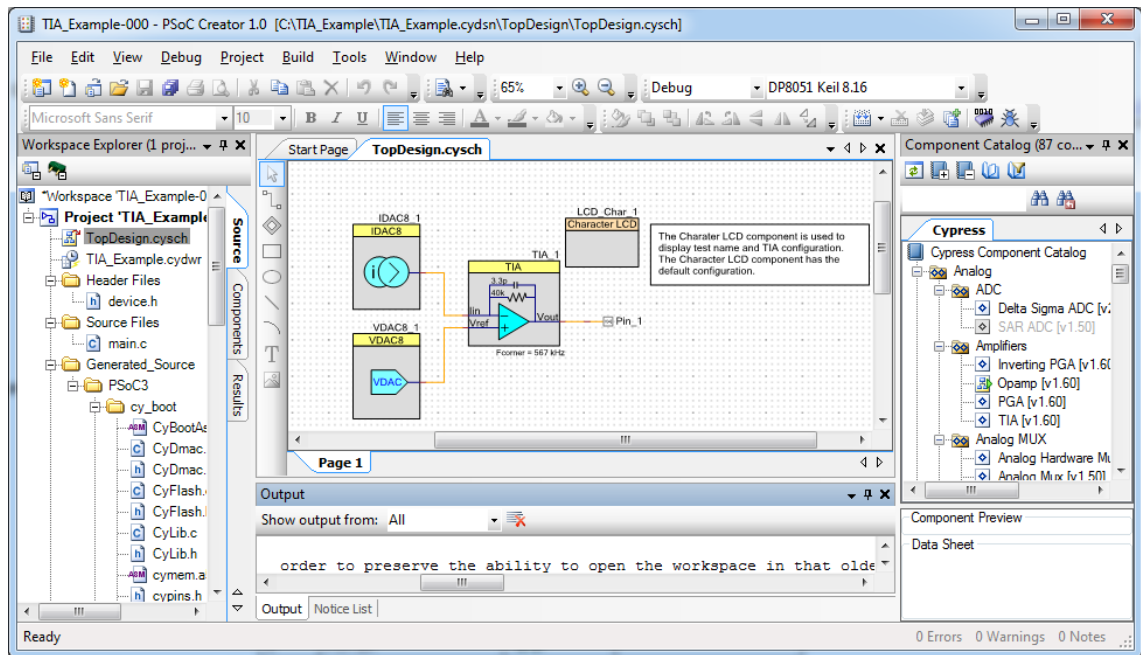
KUVA 6. Digitaalialijärjestelmän arkkitehtuuri (PSoC 3 Architecture Technical Reference Manual 2012, 221)

UDB-lohkojen lisäksi laitteessa on ohjelmoitavat digitaalilohkot erityisfunktioille, kuten USB 2.0, CAN, I²C, sekä 16-bittinen ajastin/laskuri/PWM.

Analoginen alijärjestelmä sisältää korkearesoluutioisen delta-sigma-ADC:n, neljä 8-bittistä DAC:tä, 4 komparaattoria sekä 4 operaatiovahvistinta. Neljää ohjelmoitavaa SC/CT-lohkoa voidaan käyttää mm. jatkuva-aikaisten PGA- ja TIA-vahvistimien sekä näytteenotto- ja pitopiirien toteutukseen.

4 OHJELMOINTIYMPÄRISTÖ

PSoC-järjestelmän suunnittelu ja ohjelmointi tapahtuu PSoC Creator-ohjelmistolla. Ohjelmointikielinä voidaan käyttää C- ja assembly-kieliä. Ohjelmakoodin kääntämiseen käytettiin ohjelmiston mukana tullutta Keil-kääntäjää. Graafisen käyttöliittymän myötä suunnittelu on suoraviivaista ja nopeaa (kuva 7). Tarvittavat komponentit voidaan lisätä kattavasta valikoimasta toimintoja ja asettaa niiden tarvitsemat parametrit suoraan suunnittelunäkymässä. Ohjelmointiympäristö asettaa kellosignaalit automaattisesti, reitittää I/O-toiminnot niitä vastaaviin pinneihin ja luo ohjelmointirajapinnat, joilla komponentteja voidaan ohjata. Komennot on listattu selityksineen komponenttien datalehdissä. (PSoC 3 CY8C38 Family Datasheet 2012, 43.)



KUVA 7. Ohjelmointiympäristö

Komponenttikirjastossa on valittavissa valmiita funktioita logiikkaporteista ja laiterakenteista digitaalisiin ajastimiin ja analogisiin laitteisiin, kuten suotimet ja ADC-/DAC-muuntimet. Kirjastosta löytyy myös I²C-, USB-, ja CAN-väylien protokollat. (PSoC 3 CY8C38 Family Datasheet 2012, 44.) Ohjelmointiympäristön käyttö on kuvattu liitteessä 1 olevassa käyttöohjeessa.

5 USB-VÄYLÄ

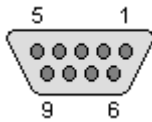
USB eli Universal Serial Bus on väyläteknikka, josta on kehittynyt pikkuhiljaa de facto-standardi oheislaitteiden liittämiseksi tietokoneen keskusyksikköön. Viime vuosina sarja- ja rinnakkaisportit ovat lähes tyystin kadonneet keskusyksiköistä ja PS/2-tyyppiset näppäimistö- ja hiiriliitännätkin ovat jo harvinaisuuksia. (Uski 2006, 50.)

USB onkin nimensä (Universal Serial Bus) veroinen yleisliitäntä, sen avulla voidaan liittää usean tyyppisiä oheislaitteita, kuten digitaalikameroita, tulostimia, skannereita, hiiriä, peliohjaimia, modeemeja ja kaiuttimia tietokoneen keskusyksikköön. USB-laitteet voidaankin jakaa useaan luokkaan käyttötarkoituksensa mukaan. Yleisimpiä laiteluokkia ovat HID-laitteet (Human Interface Device), tallennuslaitteet (Mass Storage Device, MSD) ja kommunikaatiolaitteet (Communication Device Class, CDC). (Murphy 2012, 21.)

USB-väylä on rakenteeltaan yhdistettyjen solmupisteiden verkko, jossa jokainen solmu voi olla oheislaitte, tai erillinen jakosolmu eli hub, johon liitetään muita oheislaitteita (Lähteinen, O. 2004, 259). Kun USB-laite kytketään tietokoneeseen, isäntälaitte määrää kytkettävälle laitteelle osoitteen, minkä avulla laitteen ja isännän välinen dataliikenne hoidetaan. Varsinainen USB-liikenne tapahtuu niin sanottujen piippujen (pipes) avulla. Nämä ovat isännän ja laitteen päätepisteen (endpoint) välisiä polkuja. Päätepiitteet ovat ladattavia puskureita ja voivat sisältää joko laitteesta jo vastaanotettua dataa, tai varsinaista lähetystä odottavaa dataa. USB-laitteella voi olla useampia päätepiitteitä ja jokaiselle niistä on liitetty oma piippu. Näitä piippuja on olemassa kahta tyyppiä: data- ja ohjauspiippuja. USB-standardin mukaisista neljästä siirtotyyppistä ohjauspiippua käytetään ohjaussiirtoja, kuten laitteen asetusten ja komentojen siirtoa varten. Datapiippua pitkin voidaan lähettää joko keskeytyssiirtoja lyhyen puskemaisen datan siirtoon, bulkisiirtoa suurien datamäärien siirtoon tai isokronista siirtoa, kun halutaan varmistaa siirtonopeuden laatu. (Murphy 2012, 2-3.) USB-väylä tarvitsee maa- ja käyttöjännitejohtimien lisäksi vain kaksi muuta johdinta. Nämä kaksi muuta ovat yhdistetty differentiaalinen kello ja data, nimiltään +D ja -D. Differentiaalisuutta tarvitaan väylän nopean dataliikenteen hallintaan ja sillä saadaan myöskin pidettyä sähköiset häiriöt kurissa. Tästä syystä signaalit +D ja -D kulkevat kaapelissa kierrettynä parina. (Lähteinen, O. 2004, 260.)

6 PELIOHJAINADAPTERIN TOTEUTUS

Esimerkkisovelluksena toteutettiin Commodore 64-peliohjaimen liittäminen PC-tietokoneeseen USB-väylän kautta. Peliohjain koostuu viidestä mikrokytkimestä, joista neljällä havaitaan sauvan asentoja ja yhdellä painonapin tilaa. Peliohjaimessa on kuvan 8 kaltainen 9-pinninen naarasliitin.



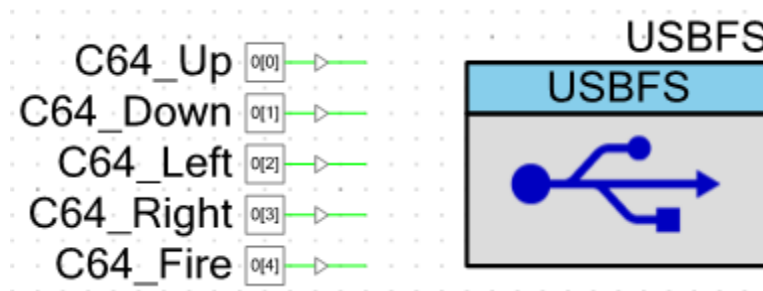
KUVA 8. 9-pin D-SUB-liitin (Handbook of hardware schemes, cables layouts and connector pinouts 2007)

Jotta peliohjain saadaan toimimaan PC:llä, mikrokontrollerilla luetaan kytkinten tiloja ja lähetetään ne USB-väylän kautta työasemalle. Adapteri toteutettiin kytkemällä peliohjaimen liittimen pinnit 1-4 ja 6 kehitysalustan portteihin P0_0 - P0_5 jotka määritettiin digitaaliseksi sisääntuloksi alasetovastuksella. Peliohjaimen ja kehitysalustan maa kytkettiin toisiinsa, jolloin kytkintä painettaessa sitä vastaavan pinnin tila painuu alas. Peliohjaimen pinnien järjestys on esitetty taulukossa 3.

TAULUKKO 3. Peliohjaimen pinnijärjestys (Commodore 64 Programmer's Reference Guide 1982, 395)

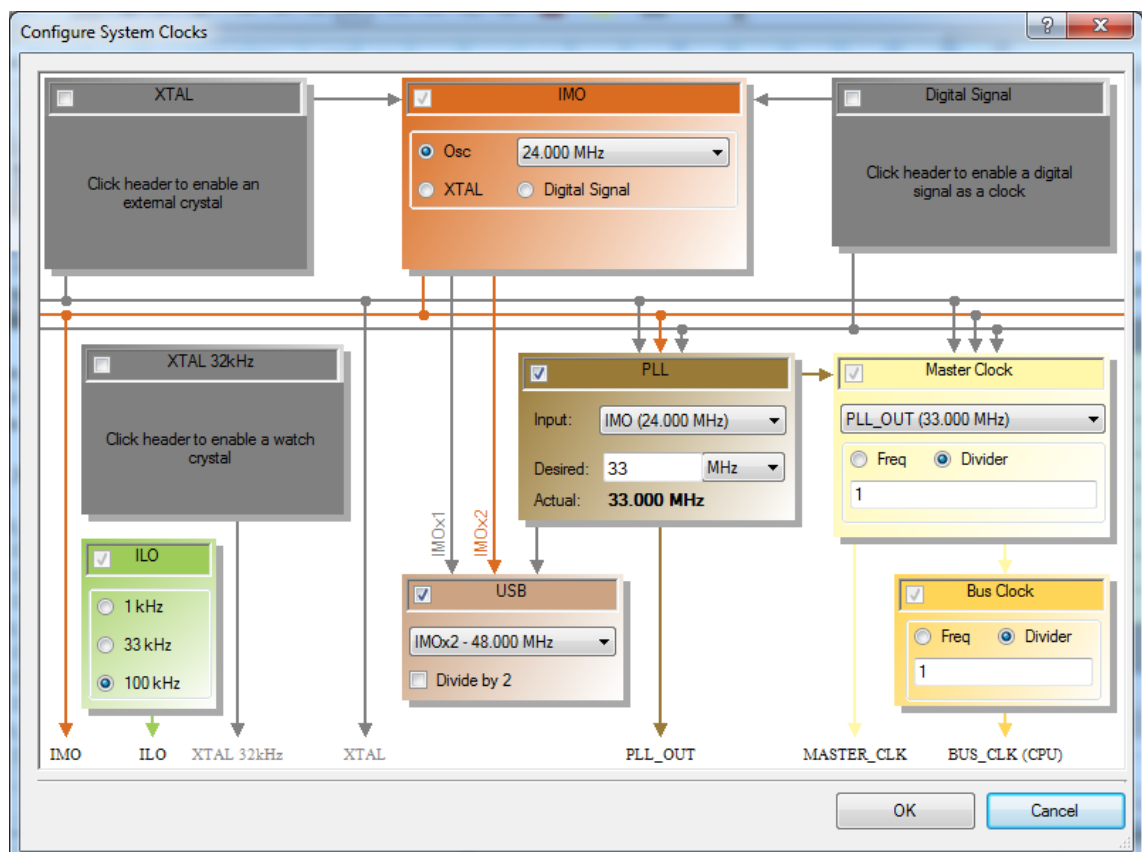
Pinni	Selitys
1	Ylös
2	Alas
3	Vasen
4	Oikea
5	Ei käytössä
6	Nappi 1
7	+5V
8	GND
9	Ei käytössä

USB-väylän määrittelyä varten haettiin komponenttivalikosta USBFS-työkalu ja asetettiin se ohjelmointiympäristön suunnittelunäkymään kuvan 9 mukaisesti.



KUVA 9. Suunnittelunäkymä

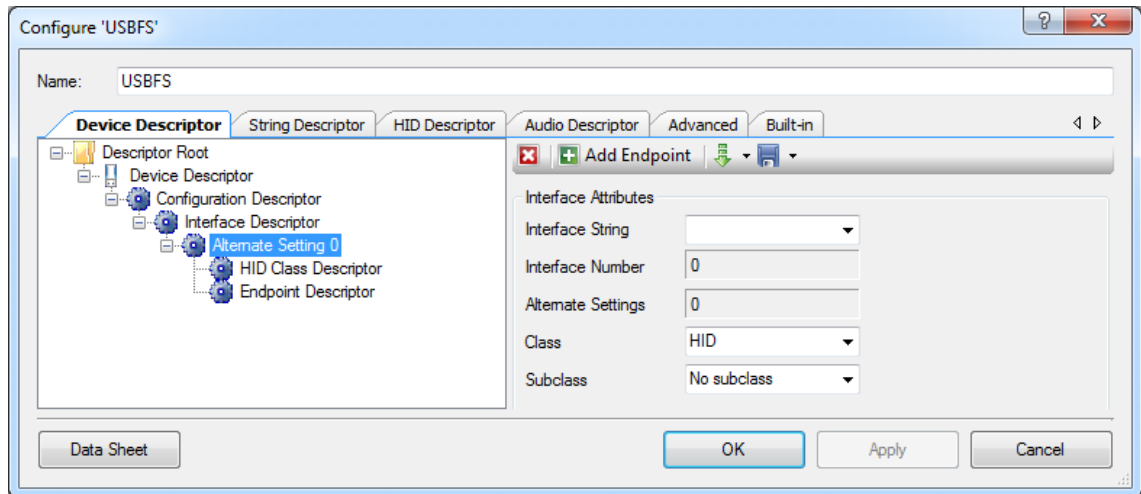
Jotta USB-väylä toimisi, täytyy laitteen kelloasetuksia muuttaa. Käytetty USB-komponentti toimii Fast Speed-nopeudella, joka vaatii 48 MHz:n kellotaajuuden. Tämä tapahtuu valitsemalla projektin .cydwrk-tiedostosta Clocks ja samasta näkymästä Edit Clock, jolloin kelloasetukset voidaan asettaa kuvan 10 mukaisesti.



KUVA 10. Kello-oskillaattoriaisetukset

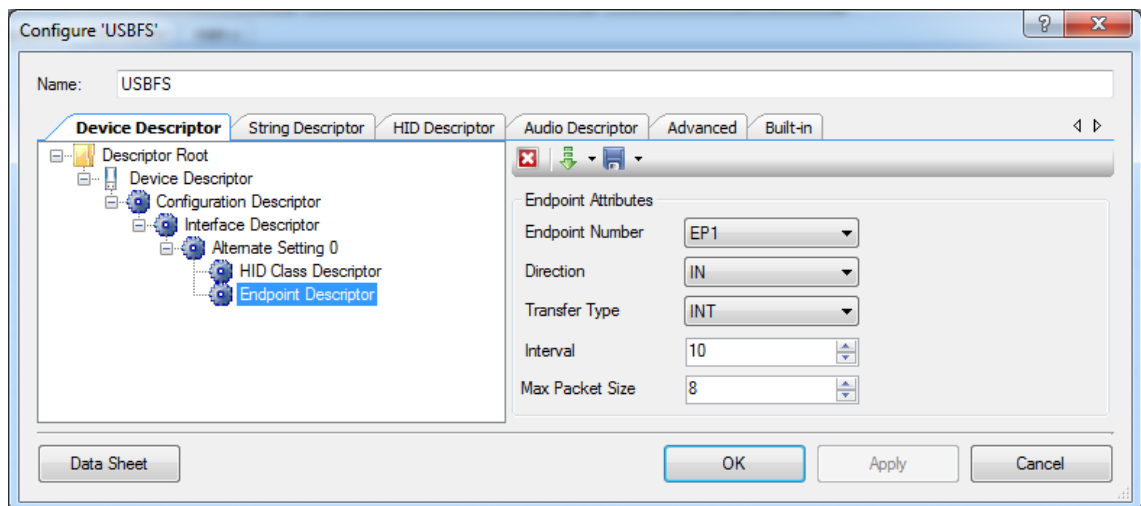
Ohjelmointiympäristössä voidaan säätää varsinaiset USB-määrittelyt helposti muokkaamalla ne USBFS-työkalulla. Peliohjainta käsitellään ohjauslaitteena, joten sen luo-

kaksi valitaan HID. Tällöin lähetettävä data on varsin pientä, jolloin siirtotyyppinä voidaan käyttää keskeytys siirtoa. Asetukset tehdään Device Descriptor-välilehdellä, josta valitaan HID-laite (kuva 11).



KUVA 11. USB-luokka

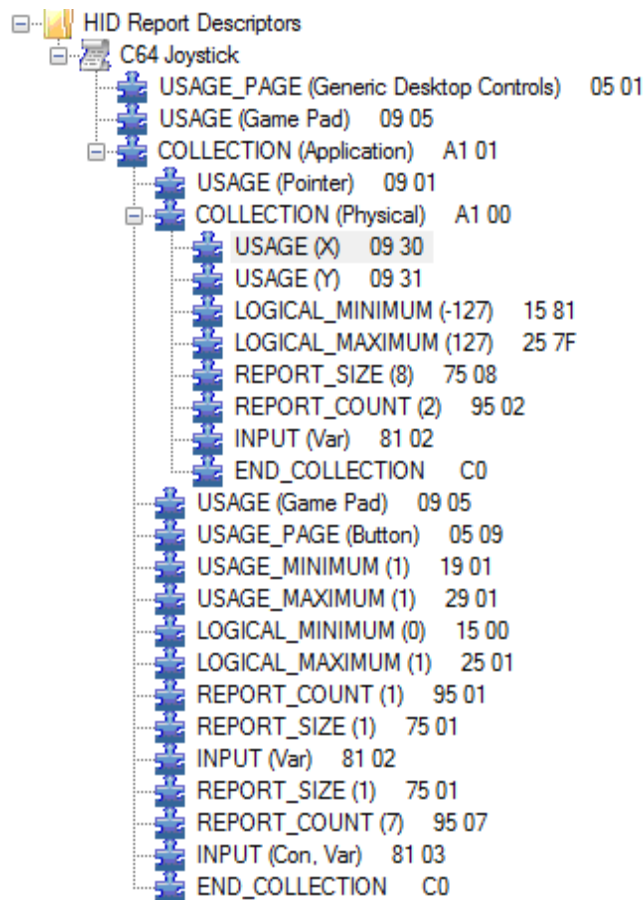
Päätepiste asetetaan Endpoint Descriptor-välilehdellä. Tiedonsiirtosuunta katsotaan isännästä päin, jolloin valitaan sisään (IN) ja tyyppi keskeytys (INT) kuvan 12 mukaisesti. Tämän jälkeen täydennetään laitteen kuvaus täyttämällä HID Descriptor-välilehti.



KUVA 12. Päätepiesteen kuvauksen asetus

Laite määritetään Game Pad-tyyppiseksi ohjauslaitteeksi, joka sisältää X- ja Y-akselit sekä yhden napin. Akseleita varten määritetään arvojen maksimi- ja minimirajoiksi -127 ja 127. Lisäksi määritetään yksi painonappi, joka saa joko arvon 0 tai 1. Painonapin

arvon mahtuessa yhteen bittiin, täytyy tavun loput 7 bittiä merkitä nolliksi, muuten laite ei toimi. Kuvassa 13 on esitetty HID-määrittely arvoineen.



KUVA 13. HID-määrittely

USBFS-komponentin ohjelmointirajapintaa käyttämällä käynnistetään ensin komponentti, odotetaan kunnes laitteen määrittely on valmis jonka jälkeen lähetetään päätepisteen 1 ensimmäinen paketti alla olevan koodin mukaisesti.

```
USBFS_Start(0, USBFS_3V_OPERATION);
while(!USBFS_bGetConfiguration());
USBFS_LoadInEP(1, joy_Data, 3);
```

Alla on esitetty ohjelasilmukan koodi, jossa ensin odotetaan päätepiste 1:n ACK-käyttelysignaalia. Kun signaali on saatu ja varmistuttu dataliikenteen toimivuudesta, luetaan if-lauseella sauvan vastakkaisia suuntia (ylös/alas) ja talletetaan tulos muuttujaan Y_Pos. Mikäli sauvalla ohjataan ylös, asetetaan y-akseli minimiarvoonsa, jolloin PC tulkitsee liikkeen ylöspäin. Samalla tavalla mikäli sauvalla ohjataan alaspäin, asetetaan

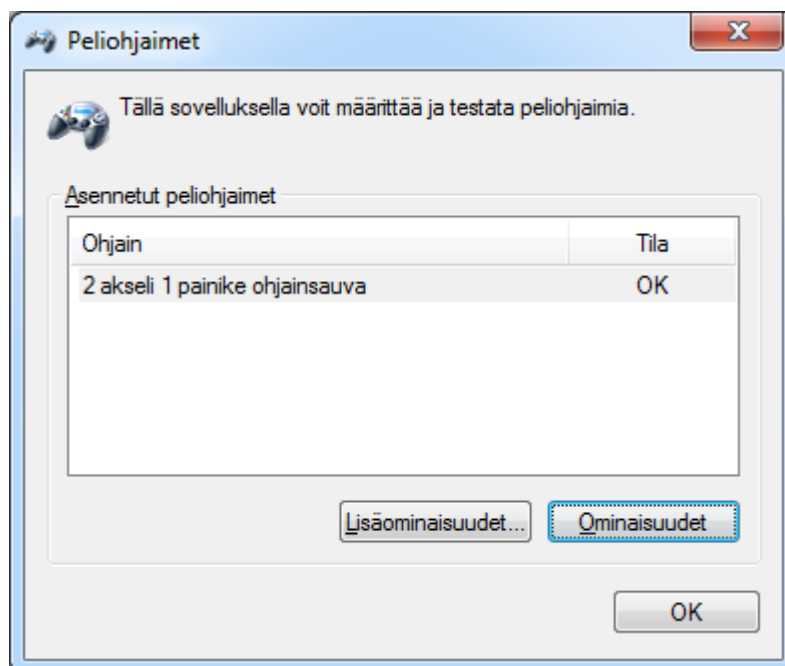
arvoksi maksimi ja liike suuntautuu alaspäin. X-akseli luetaan ja talletetaan samantyyli-
sesti muuttujaan X_Pos.

```
while(!USBFS_bGetEPAckState(1));
if (!C64_Up_Read())
    Y_Pos=-127;
else if (!C64_Down_Read())
    Y_Pos=127;
else
    Y_Pos=0;
```

Painonapin tila talletetaan muuttujaan button, jonka jälkeen arvot talletetaan kolmiaki-
oiseen taulukkoon joy_Data[], järjestyksessä x, y, nappi. Tämän jälkeen data lähetetään
komennolla:

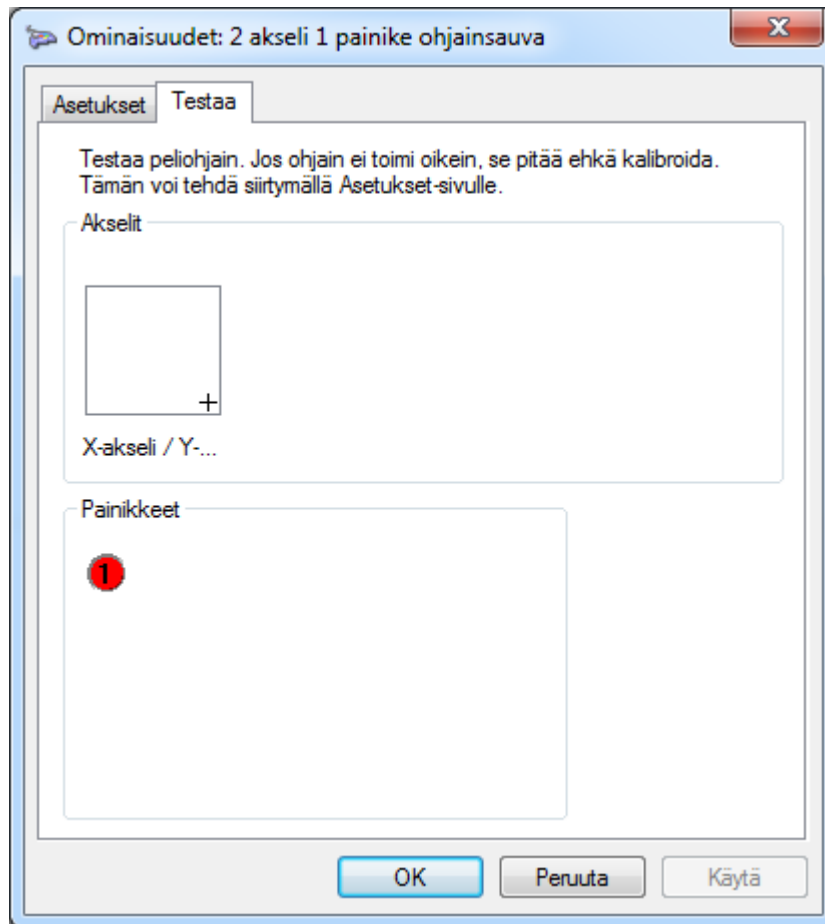
```
USBFS_LoadInEP(1, joy_Data, 3);
```

Tämän jälkeen projekti käännetään ja ohjelmoidaan PSoC-laitteen flash-muistiin. Kehi-
tysalustan mini-B-USB-liitäntää käyttämällä laite saadaan kytkettyä PC-tietokoneeseen.
Laite tunnistuu Windows-ympäristössä automaattisesti peliohjaimeksi kuvan 14 mukai-
sesti eikä erillistä ajuria tarvita.



KUVA 14. Peliohjain kytkettynä

Ominaisuudet-valikossa voidaan testata ohjaimen toimintaa, kuten kuvasta 15 on nähtävissä.



KUVA 15. Peliohjaimen testaus

7 POHDINTA

Opinnäytetyölle asetetut tavoitteet kehitysalustaan ja sen ominaisuuksiin tutustumisesta saavutettiin. Kehitysalustan käytössä ilmenneiden ongelmien syyt löydettiin ja niihin esitettiin ratkaisut liitteenä olevassa käyttöönotto-oppaassa. Esimerkkisovelluksen teko onnistui hyvin ja sen avulla pystyttiin havainnollistamaan suunnitteluprosessin kulkua.

Työn aikaavievin ja haastavin osuus oli mainittujen ongelmakohtien selvittäminen. Ongelmat ilmenivät pääasiassa vasta sovelluskehityksen viimeisessä vaiheessa, jossa projektia yritettiin kääntää ja ohjelmoida piirille. Ohjelmointiympäristössä olleet valmiit esimerkkiohjelmat saatiin kuitenkin siirrettyä piirille ilman ongelmia. Lopulta kääntäjän lisenssiavain jouduttiin muokkaamaan käsin kääntäjän kokoonpanotiedostoon. Piiri saatiin ohjelmoitua, kun pidättäydyttiin päivittämästä PSoC Creator-ohjelmisto uusimpaan 2.0-versioon. Sen sijaan käytettiin aiempaa, 1.0-versiota Service Pack 2-laajennuksella. Tällöin voitiin valita laitteen asetuksista koulun käytössä olevan prosessorikortin aiempi näyteversio, jolloin ohjelmointiympäristöllä voitiin luoda korttiin yhteensopiva tiedosto ohjelmointia varten.

Peliohjainesimerkki toteutettiin ainoastaan havainnollistamaan kehitysalustan käyttöä ja jatkuvaan pelauskäyttöön tulevaan adapteriin tulisikin tehdä hieman jatkokehitystä. Peliohjaimena voitaisiin esimerkiksi käyttää ohjainta, jossa on enemmän nappeja ja esimerkiksi analogisia ohjaussauvoja, jolloin alustan ominaisuuksia saataisiin enemmän käyttöön. Lisäksi koodia voisi parantaa, esimerkissä USB-väylälle lähetetään jatkuvasti dataa, vaikka peliohjainta ei liikuteta mihinkään suuntaan.

LÄHTEET

C64 Control port pinout. 2013. Handbook of hardware schemes, cables layouts and connector pinouts. [www-sivu]. [luettu 28.5.2013].
http://old.pinouts.ru/Inputs/ControlPortC64_pinout.shtml

Commodore 64 Programmer's Reference Guide. 1982. 1. painos. Indianapolis: Howard W. Sams & Co. Inc.

CY8CKIT-001 PSoC Development Kit Guide. 2012. [pdf]. [luettu 26.5.2013].
www.cypress.com/?docID=35977

Jämsä, L. 2011. Mikä olisi paras kehitysalusta? [www-sivu]. [luettu 29.5.2013].
<http://www.ruuvipenkki.fi/2011/04/10/mika-olisi-paras-kehitysalusta>

Lähteinen, O., Pietikäinen, V., Kosonen, H. 2004. Suuri PC-tietokirja. 1. painos. Juva: Sanoma Magazines Finland.

Murphy, R. 2012. USB 101: An introduction to Universal Serial Bus 2.0. [pdf]. [luettu 26.5.2013]. <http://www.cypress.com/?docID=33237>

PSoC 3 Architecture Technical Reference Manual. 2012. [pdf]. [luettu 26.5.2013].
<http://www.cypress.com/?docID=40001>

PSoC 3 CY8C38 Family Datasheet [pdf]. [luettu 26.5.2013].
<http://www.cypress.com/?docID=39699>

Uski J. 2006. USB monena versiona. MikroPC 8/2006, 50-52.

Vahtera, P. 2003. Mikro-ohjaimen ohjelmointi C-kielellä. 1. painos. Porvoo: WS Bookwell Oy.

LIITTEET

Liite 1. PSoC Development Kit asennus- ja käyttöohje

PSoC Development Kit asennus- ja käyttöönotto-ohje

PAKETIN SISÄLTÖ

Cypress PSoC Development kit-paketti sisältää:

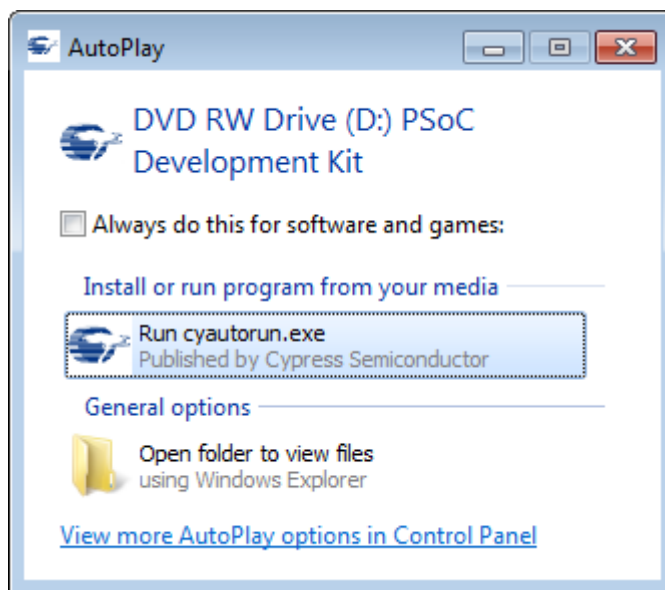
- Cypress CY8CKIT-001 DVK1 kehitysympäristön
- CY8C29 prosessorimoduulin
- CY8C38 prosessorimoduulin
- pikaohjeen kehitysympäristölle
- ohjelehtiset prosessoreille
- MiniProg3 ohjelmointi- ja debuggaustyökalun
- USB- ja ohjelmointikaapelit
- 2 kpl asennus-CD-levyjä
- 12 Vdc-muuntajan
- johdon virtalähteen liittämistä varten
- pussillisen hyppylankoja.

ALKUVALMISTELUT

Ennen käytön aloittamista asennetaan PSoc Creator-, ja PSoC Programmer-ohjelmistot. CY8C38-prosessori käyttää PSoC 3-arkkitehtuuria ja sen tarvitsemat sovellukset asennetaan ”PSoC 3/PSoC5 Software”-nimiseltä CD-levyltä. Myöskään MiniProg-ohjelmointilaitetta ei tule yhdistää koneeseen ennen kuin ohjelmistot on asennettu.

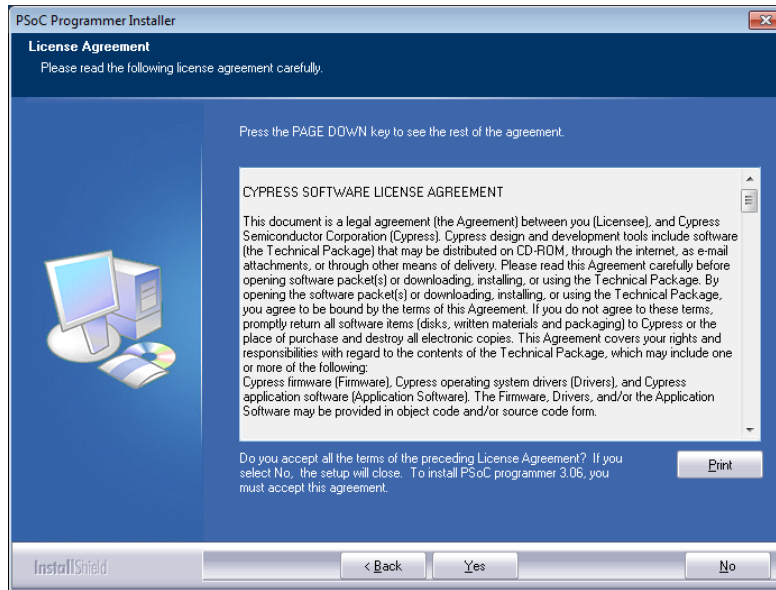
Ohjelmiston asentaminen

Asetetaan ”PSoC 3/PSoC5 Software”-levy optiseen asemaan ja käynnistetään asennus. Mikäli asennus ei käynnisty automaattisesti asennusohjelmaan, valitaan kuvan 1 mukaisesta ruudusta cyautorun.exe-ohjelma.

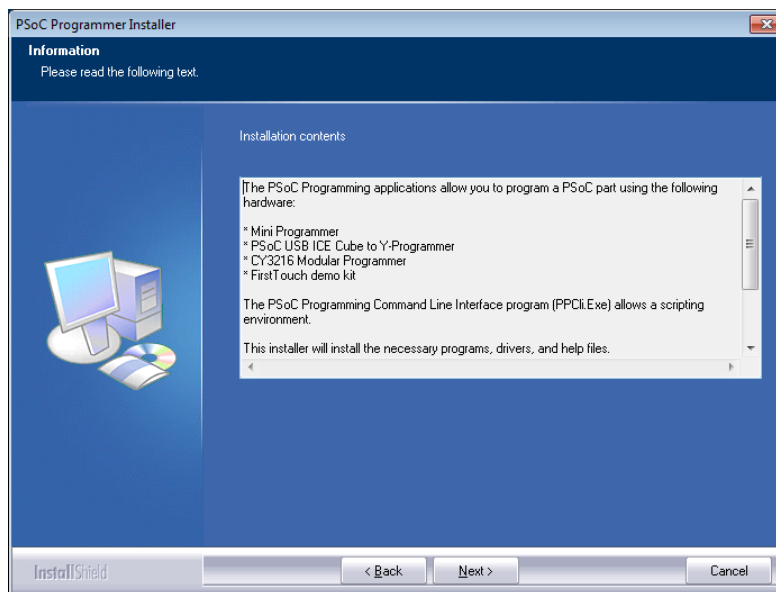


KUVA 1. Aloitus

Asennus aloitetaan valitsemalla Install Software for PSoC 3 / PSoC 5... kuvan 2 mukaisesti.

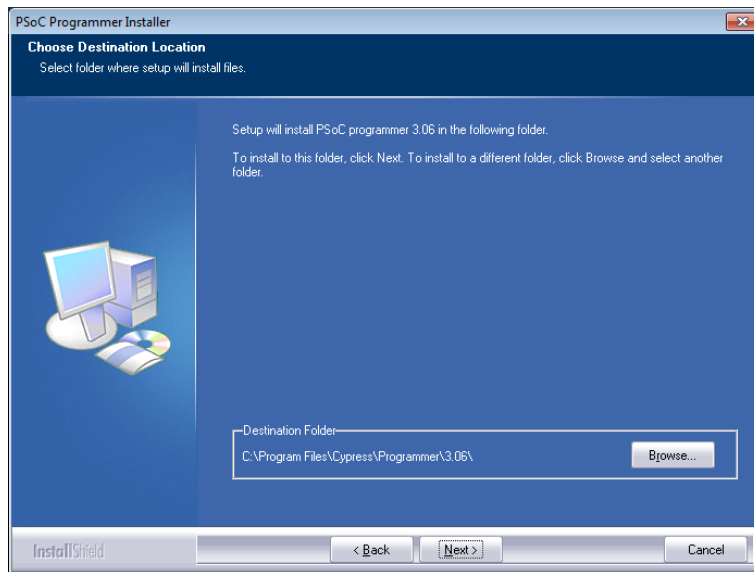


Kuva 4. EULA-ehdot



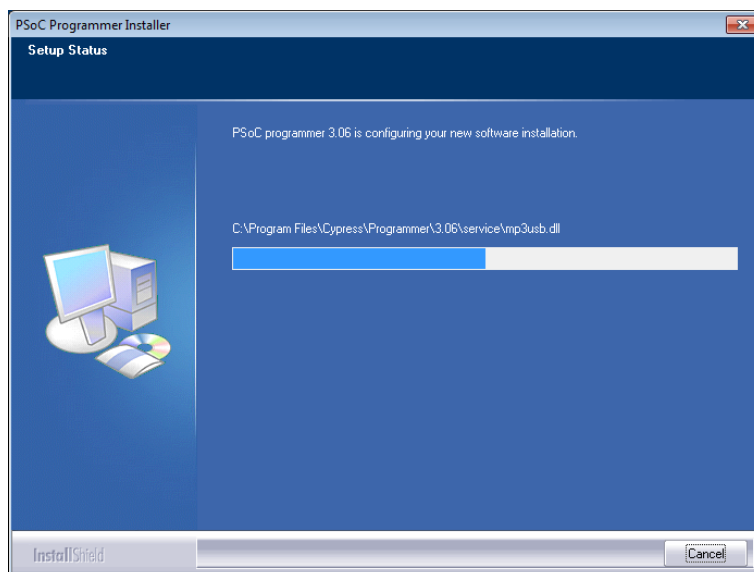
Kuva 5. Yhteenveto

Asennushakemistoksi valitaan oletusvaihtoehto (kuva 6).



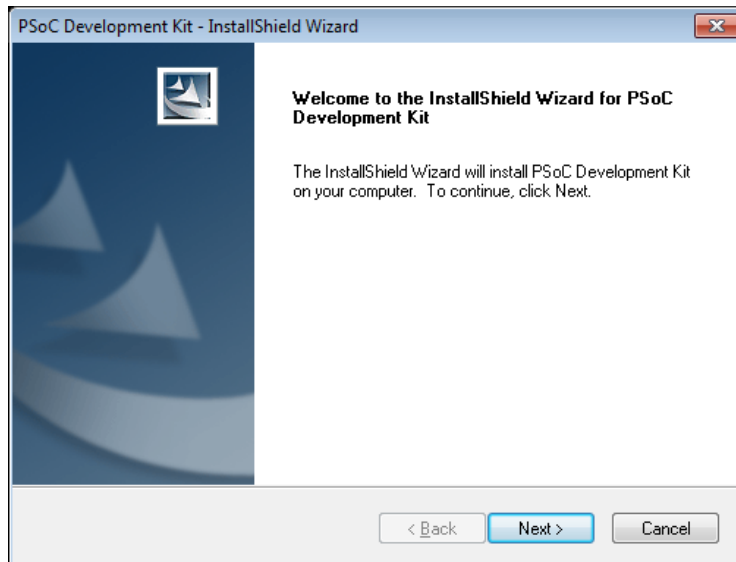
KUVA 6. Asennushakemisto

Ohjelman asennuksen (kuva 7) jälkeen suljetaan asennusohjelma klikkaamalla seuraavan ikkunan *Finish*-nappia.



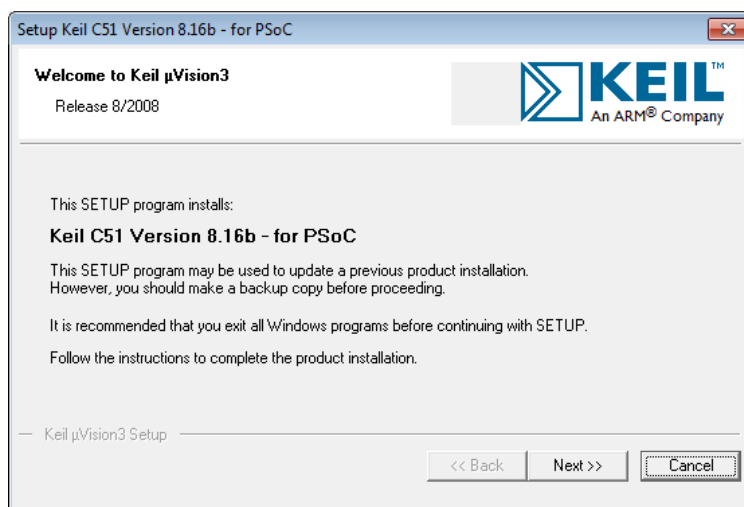
KUVA 7. Asennus

Seuraavaksi asennetaan kehitysympäristön loput osat asennusapuohjelman avulla (kuva 8). Ensimmäiseksi asennetaan PSoC programmerin uudempi versio (3.10). Vaikka kyseinen ohjelma oli juuri lisätty, asennuksen saattamiseksi onnistuneesti loppuun myös tämä tulee asentaa. Ohjelman asennus on samantyyppinen kuin edellisen version, joten sitä ei käydä tässä läpi.



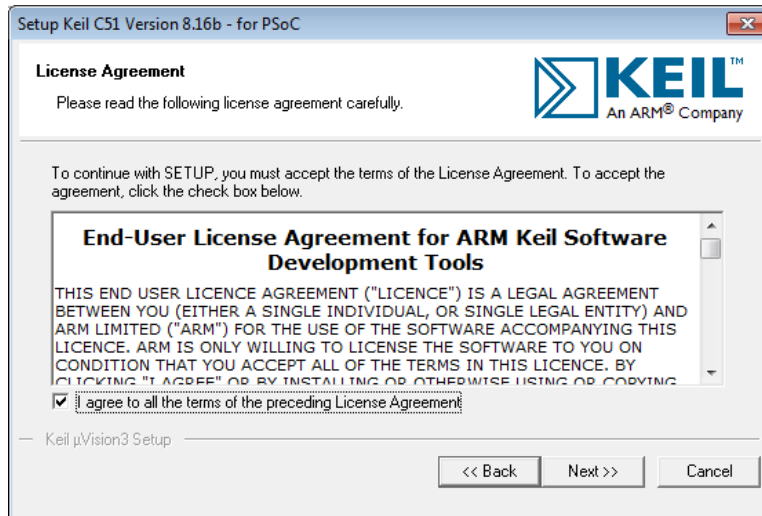
KUVA 8. Asennusapuohjelma

Tämän jälkeen asennetaan Keil-kääntäjä, jolla käännetään C-kielinen lähdekoodi PSoC-laitteelle (kuva 9).

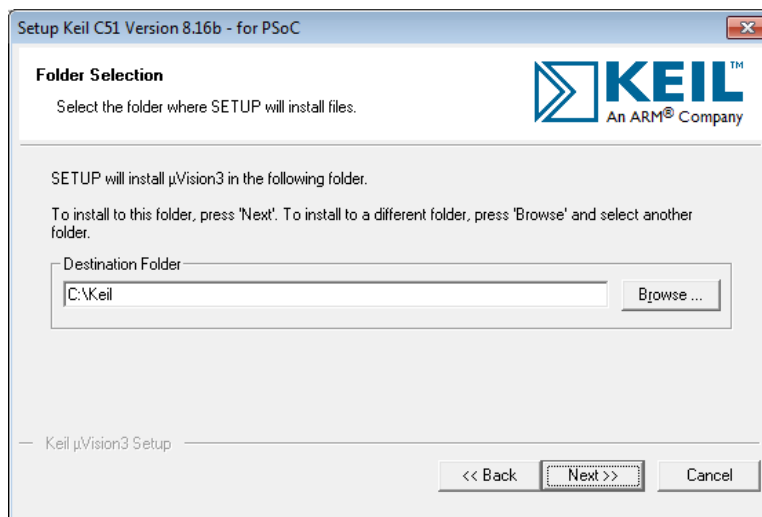


KUVA 9. Keil-kääntäjän asennus

Hyväksytään kuvan 10 käyttöehdot ja käytetään oletustallennushakemistoa (kuva 11).

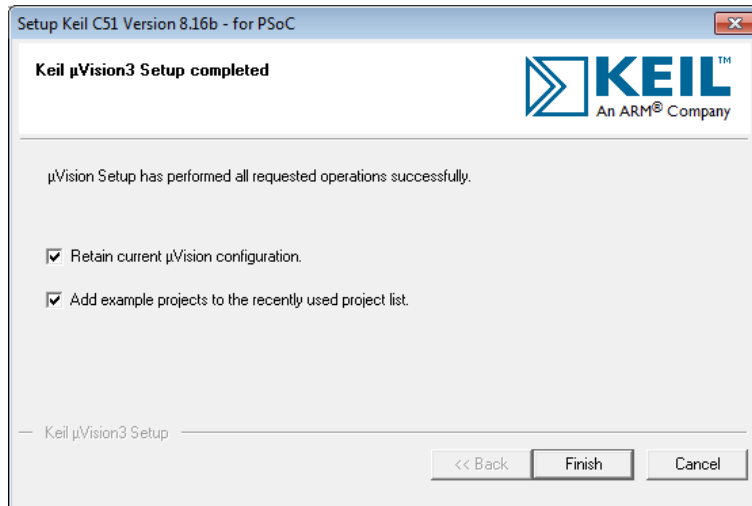


KUVA 10. Keil EULA



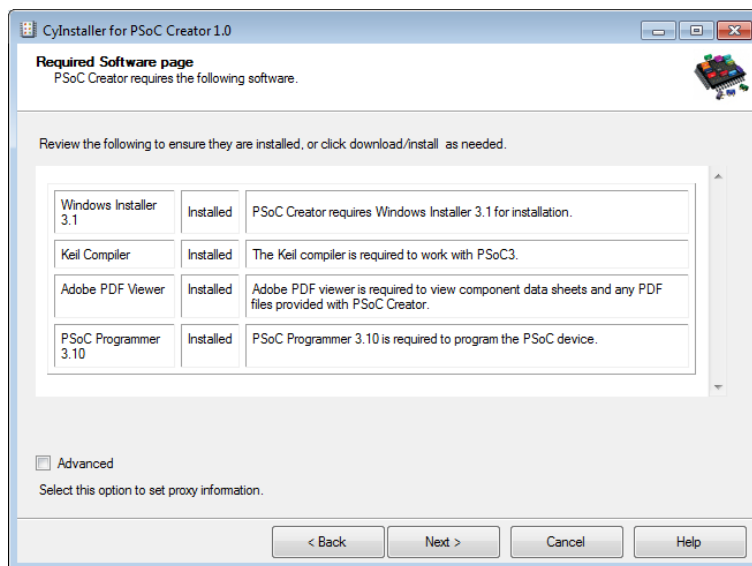
KUVA 11. Asennushakemisto

Asiakastietojen lisäyksen jälkeen lopetetaan asennus klikkaamalla *Finish* (kuva 12).



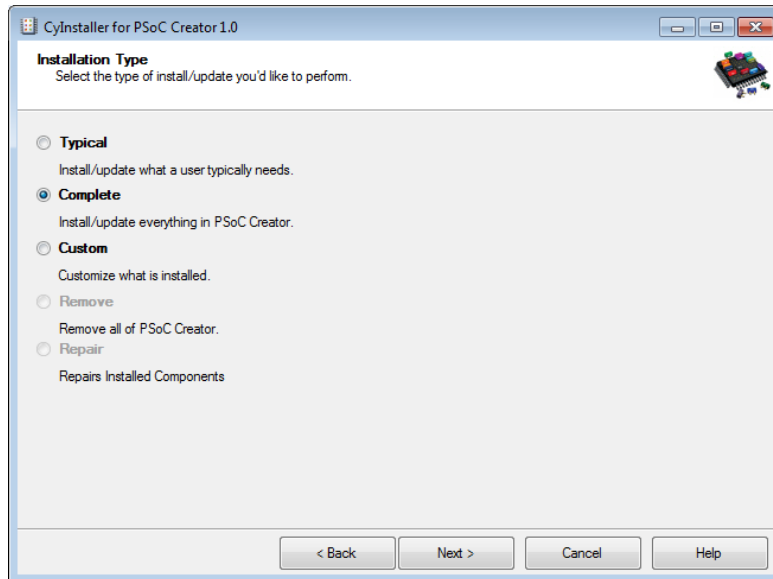
KUVA 12. Keil-asennus on valmis

Sitten asennetaan vielä PSoC Creator-ohjelmisto. Tässä vaiheessa asennuksen tulisi näyttää kuvan 13 mukaiselta.



KUVA 13. Asennuksen tarvitsemat osat

Seuraavasta ruudusta valitaan asennuksen tyyppiksi *Complete* ja jatketaan klikkaamalla *Next* (kuva 14). Asennusohjelma lataa tarvittu osat, jonka jälkeen hyväksytään ohjelmiston ehdot.



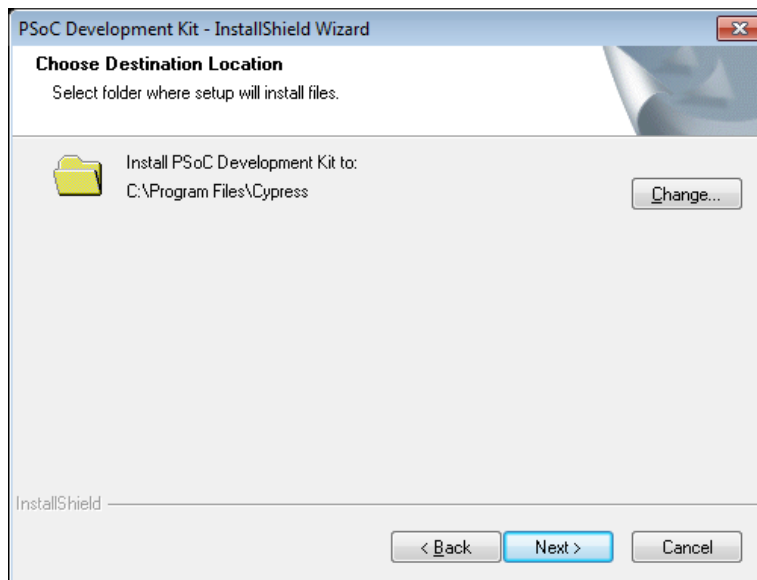
KUVA 14. Asennustyyppi

Asennus ei ole vielä tässä vaiheessa valmis, joten otetaan ruksit pois *Start PSoC Creator*-valinnasta ja jatketaan asennus loppuun klikkaamalla *Finish* (kuva 15).



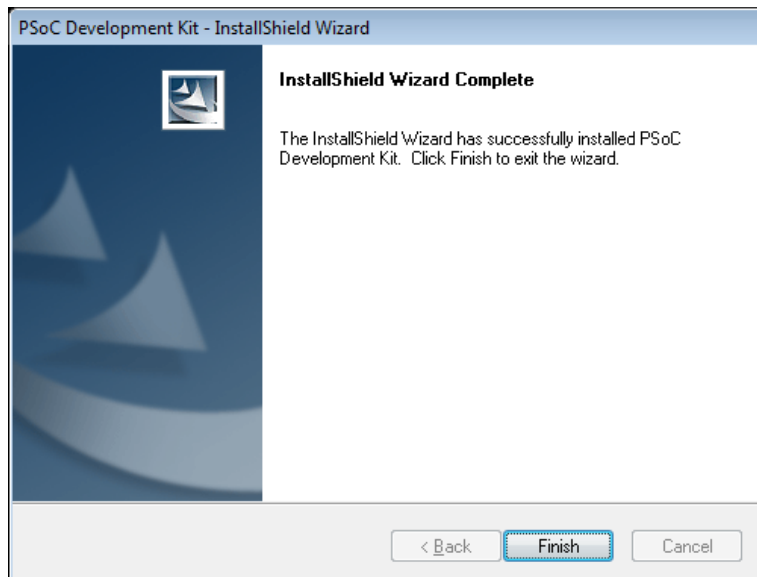
KUVA 15. PSoC Creator-asennus valmis

Asennusapuohjelman lopuksi asennetaan vielä PSoC Development Kit. Käytetään oletushakemistoa kuvan 16 tavalla ja jatketaan klikkaamalla *Next*. Asennustyypin valinnan jälkeen asennusohjelma lataa jälleen tarvittavat tiedostot ja asentaa ne.



KUVA 16. Oletusasennus

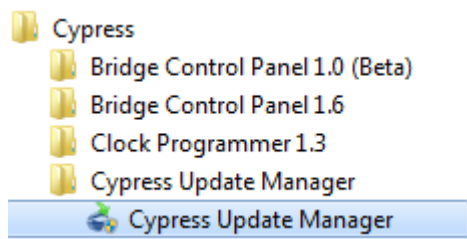
Tämän jälkeen viimeistellään asennus klikkaamalla *Finish* (kuva 17).



KUVA 17. Valmis asennus

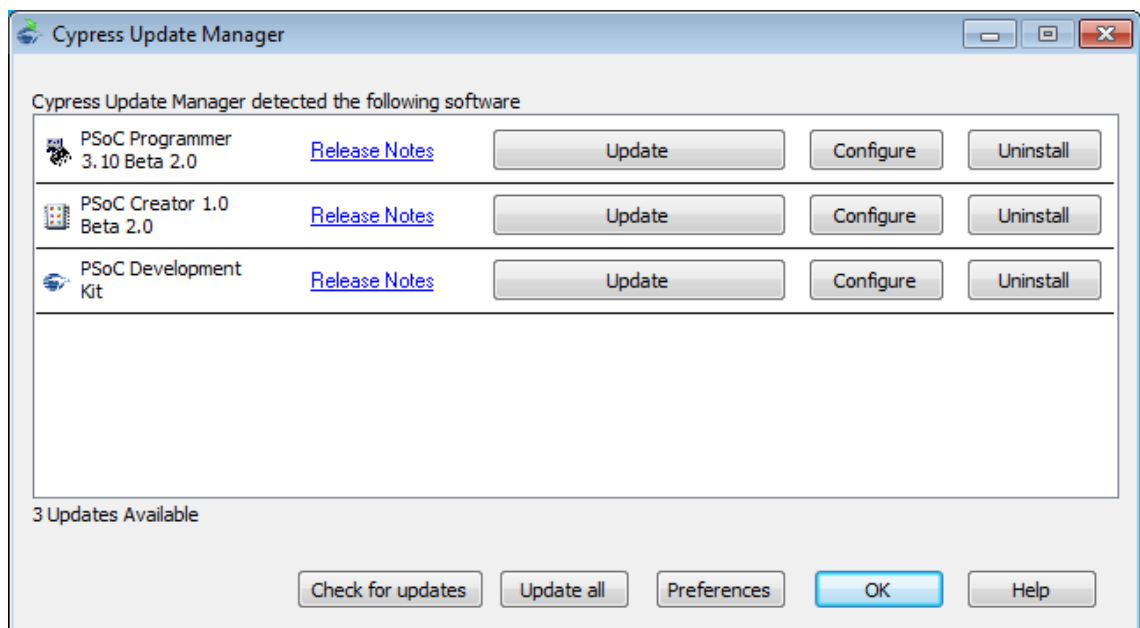
Ohjelmiston päivitys

Ennen käytön aloittamista päivitetään vielä PSoC Creator-ohjelmisto versioon Service Pack 2. Avataan päivitysohjelma käynnistysvalikosta *All programs* -> *Cypress* -> *Cypress Update Manager* (kuva 18).



KUVA 18. Päivitysohjelma

Ennen päivitystä ikkunan tulisi näyttää kuvan 19 kaltaiselta.



KUVA 19. Päivitettävät ohjelmistot

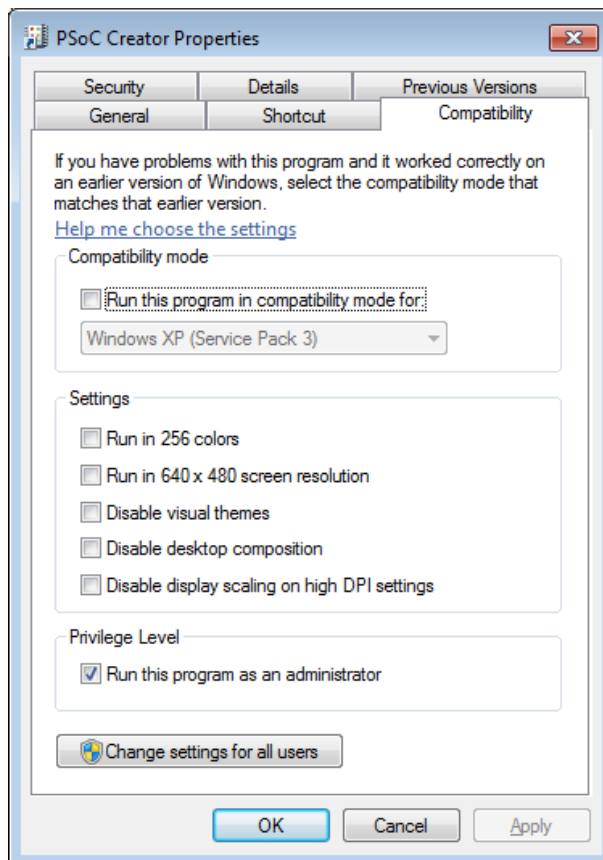
Asennetaan PSoC Creator 1.0 Service Pack 2 klikkaamalla ohjelmiston nimen vieressä olevaa *Update*-nappia. PSoC Programmerin voi halutessaan päivittää uudempaan versioon, mutta tämä ei ole välttämätöntä.

KÄYTTÖÖNOTTO

Kun ohjelmistot on onnistuneesti asennettu työasemalle, voidaan aloittaa PSoC-kehitysympäristöön tutustuminen asentamalla vihreä CY8C38-mallin prosessorikortti kehitysalustalle.

Aloitustoimet

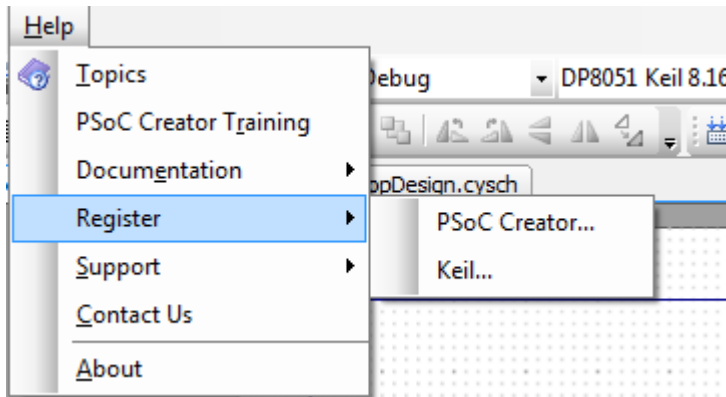
Ennen alustan kytkemistä PC:hen, käynnistetään PSoC Creator 1.0-ohjelma. On suotavaa käyttää ohjelmaa järjestelmänvalvojan oikeuksilla, joten ensimmäiseksi muutetaan pikakuvakkeen asetuksia siten, että *Compatibility*-välilehdeltä ruksitaan kohta *Run this program as an administrator* (kuva 20).



KUVA 20. Järjestelmävalvojan oikeuksien asettaminen

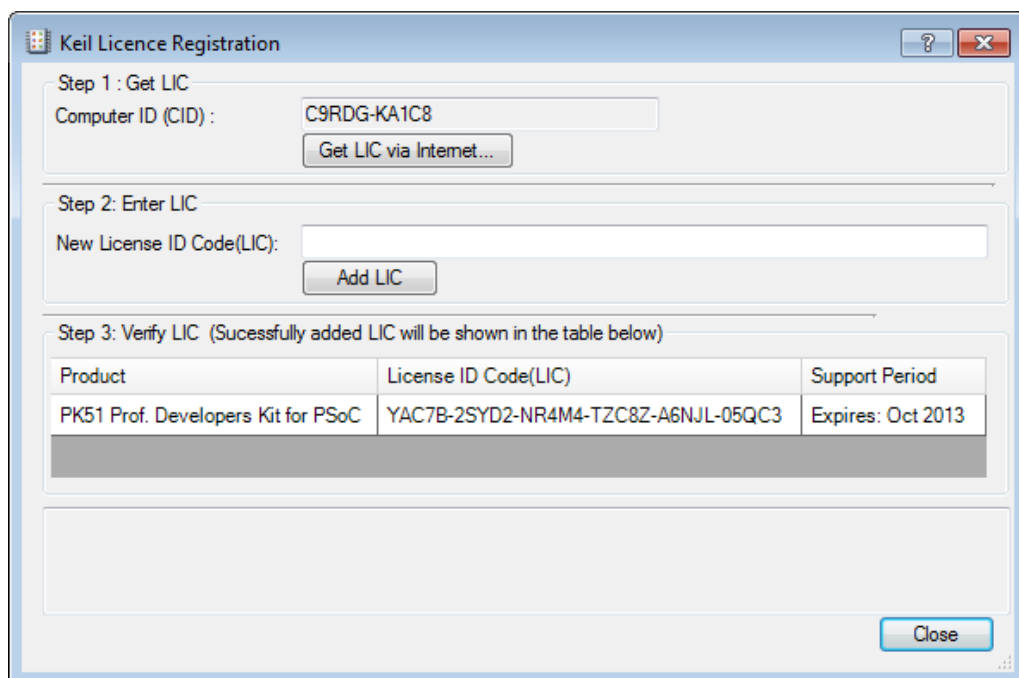
Lisenssiavaimen lisääminen

Mikäli ohjelmisto on asennettu kokonaan uusiksi tai Keil-kääntäjän lisenssi on umpeutunut, lisenssikoodi täytyy syöttää uudelleen. Tämä tapahtuu valitsemalla *Help*-valikosta - > *Register* - > *Keil...* (kuva 21).



KUVA 21. Kääntäjän rekisteröinti

Uuden lisenssiavaimen saa napsauttamalla *Get LIC via Internet*-nappia, jolloin uuden avaimen saa tilattua sähköpostiin. Kuvassa 22 on esitetty rekisteröinti-ikkuna toimivalla lisenssillä.

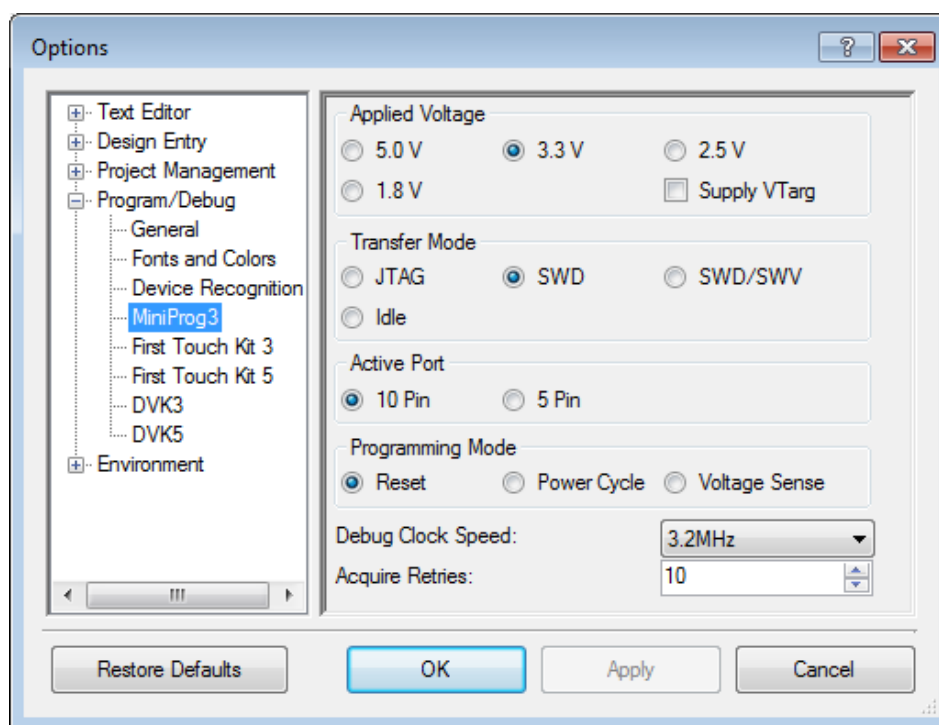


KUVA 22. Lisenssi kunnossa

Jos ohjelma ei jostain syystä hyväksy hankittua lisenssiä, tulee tiedostoon TOOLS.INI lisätä rivi LIC0=, jolle yhtäsuuruusmerkin jälkeen lisätään lisenssiavain. Tiedosto löytyy oletusasennuksessa sijainnista C:\Program Files\Cypress\PSoC Creator\1.0\PSoC Creator\import\keil\pk51\8.16. Tiedoston tallentamista varten tekstinkäsittelyohjelma (esimerkiksi *Notepad*) tulee ensin käynnistää järjestelmävalvojan oikeuksilla.

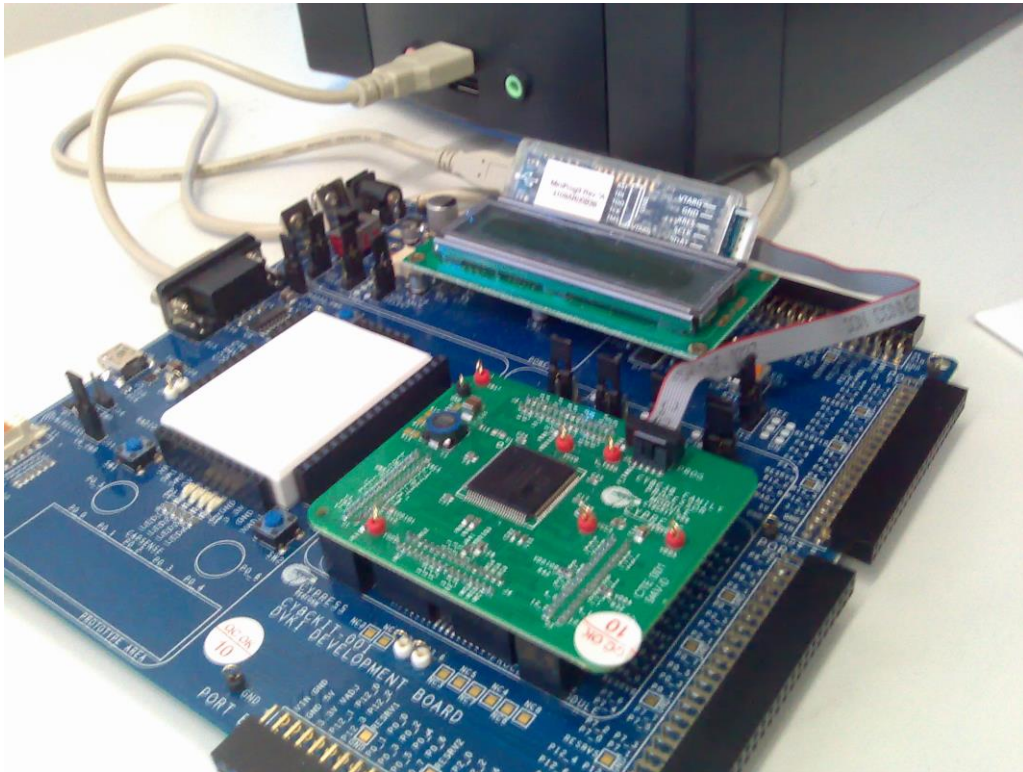
MiniProg3-työkalun asetukset ja kytkentä

Ennen alustan liittämistä PC:hen on hyvä käydä läpi MiniProg3-työkalun asetukset. Nämä löytyvät *Tools*-valikon *Options*-kohdasta. Asetuksiksi valitaan kuvan 23 mukaisesti: jännite 3,3 V, *Transfer mode*: SWD, 10-pinninen liitin, *programming mode*: reset ja kellotaajuudeksi 3,2 MHz.



KUVA 23. MiniProg3-asetukset

Kun asetukset on laitettu kuntoon, voidaan kytkeä MiniProg3-työkalu PC:n USB-porttiin. Tarvittavat ajurit ladataan automaattisesti, ellei näin ole jo aiemmin tehty. Työkalu liitetään prosessorikortin PROG-liitäntään pienellä lattaakaapelilla kuvan 24 mukaisesti.

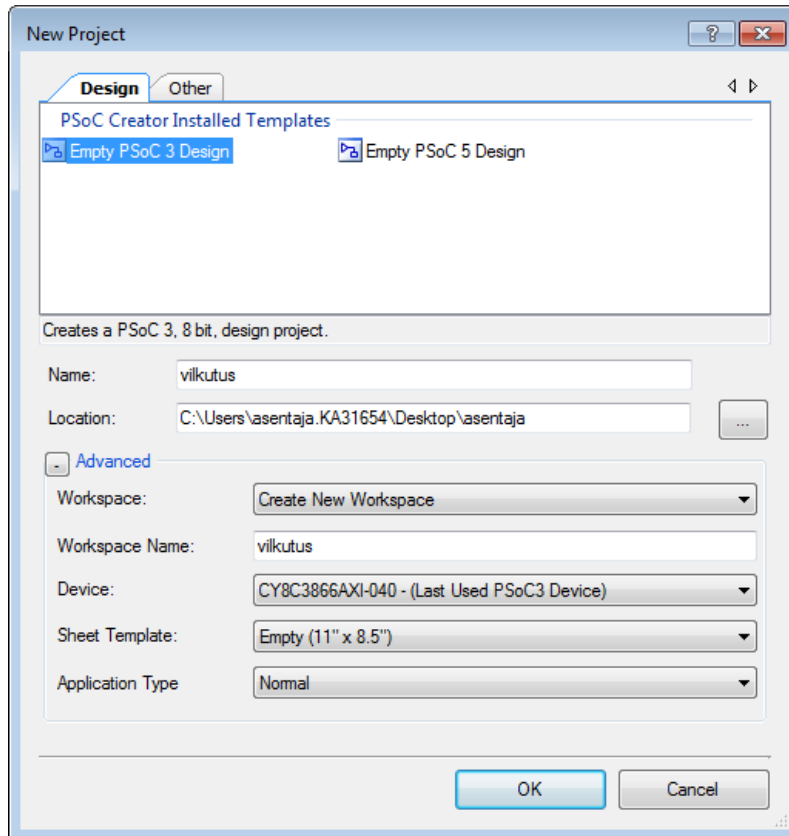


KUVA 24. Laitteen kytkeminen PC-koneeseen

Nyt voidaan aloittaa tilakoneen ohjelmointi.

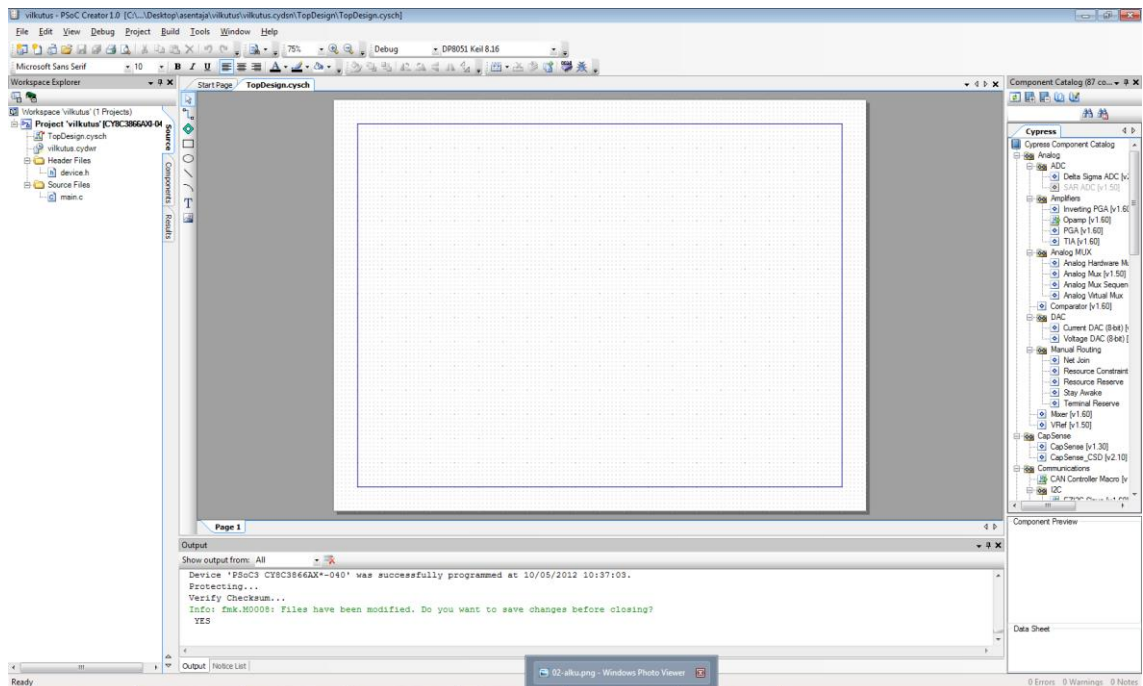
Tilakoneen ohjelmointi

Tutustutaan alustan ohjelmoimiseen toteuttamalla yksinkertainen ledien vilkutusohjelma. Levyllä olevia ledejä vilkutetaan järjestyksessä *LED1*, *LED2*, *LED3*, tyhjä, *LED1* jne. Käynnistetään PSoC Creator 1.0-ohjelma ja luodaan tyhjä PSoC3-projekti valitsemalla *File*-valikosta *New Project*. Valitaan *Empty PSoC 3 Design* ja määritellään projektille nimi ja tallennuspaikka. *Advanced*-valikosta valitaan laitteeksi *CY8C3866AXI-040* ja hyväksytään muutokset napsauttamalla *OK*, kuten kuvassa 25 on esitetty.



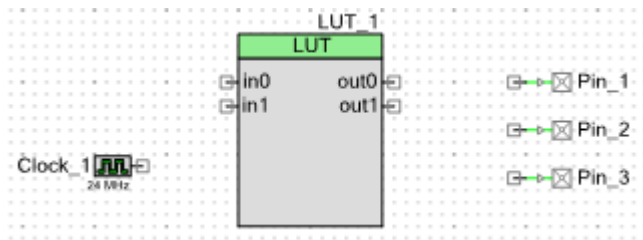
KUVA 25. Uuden projektin luominen

Ruudulle aukeaa kuvan 26 kaltainen näkymä. Oikealla on komponenttikatalogi, josta valitaan projektissa käytettävät osat.



KUVA 26. Suunnittelunäkymä

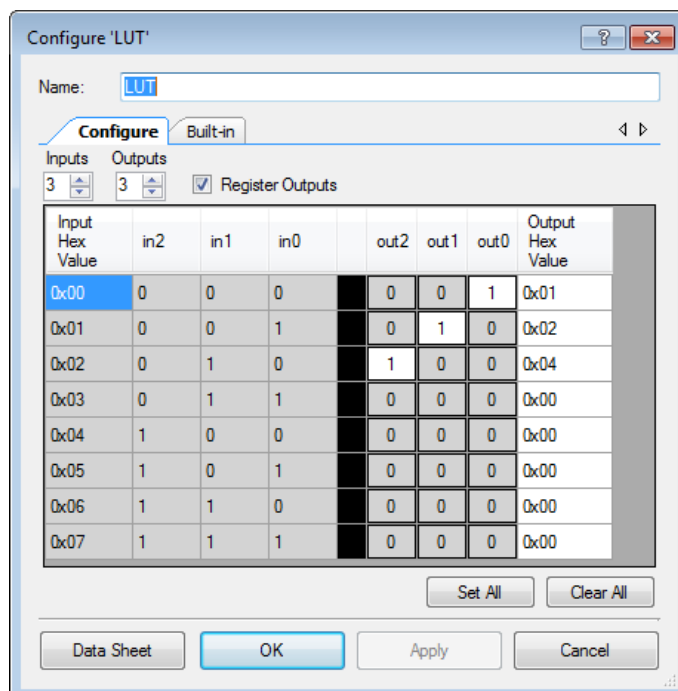
Lisätään komponenttikatalogista komponentteja vetämällä ne TopDesign-näkymään. Valitaan *Ports and Pins*-osasta 3 *Digital out pin*:iä vilkuteltavia ledejä varten, *Logic*-osasta *Lookup Table*, jolla määritetään ledien vilkkumisjärjestys, sekä *System*-valikosta *Clock* vilkutusnopeuden asettamista varten. Tämän jälkeen suunnittelunäkymän tulisi olla jotakuinkin kuvan 27 kaltainen.



KUVA 27. Lisätyt komponentit

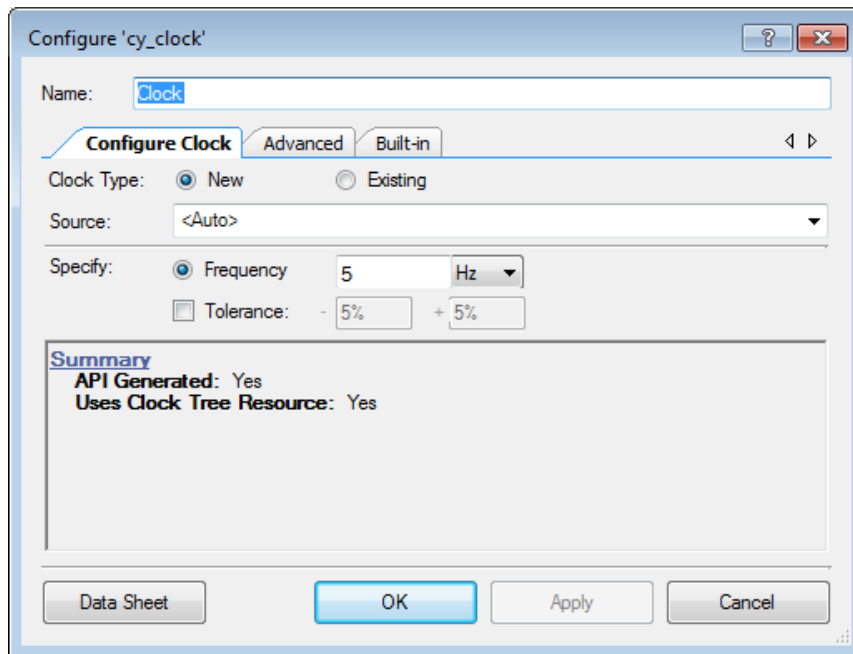
Komponenttien muokkaus

Komponentteja täytyy hieman muuttaa, ennen kuin ne voidaan kytkeä. Aloitetaan kaksoisnapsauttamalla *Lookup Tablea*, jolla määritetään ledien tilat ja vilkutusjärjestys. Vaihdetaan komponentin nimeksi vaikkapa LUT ja lisätään sisään- ja ulostulojen määräksi molempiin 3. Lisäksi ruksitaan *Register Outputs*-kohta, jotta saadaan kellopulssille sisääntulo. Ulostulot määritellään kuvan 28 mukaisesti. Hyväksytään muutokset ja jatketaan kellon asetuksiin.



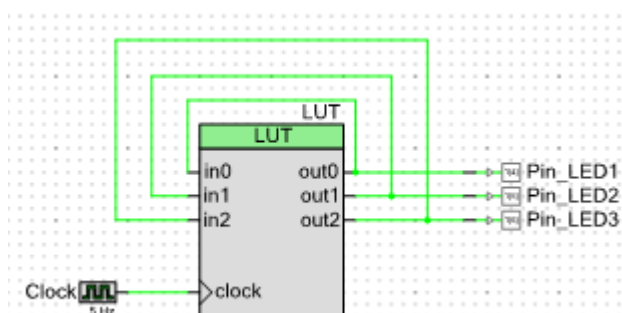
KUVA 27. LUT:n määrittelyt

Kello määritetään kuvan 29 mukaisesti. Taajuudeksi asetetaan vaikkapa 5 hertsiä ja muutetaan nimeksi ”Clock”.



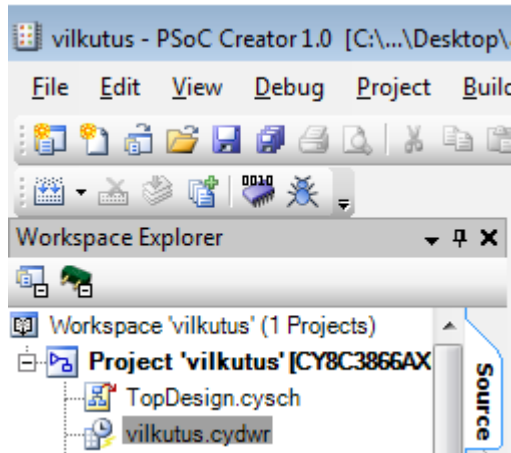
KUVA 29. Kellopulssin asetukset

Vaihdetaan vielä ulostulopinnien nimiksi kuvaavammat Pin_LED1, Pin_LED2 ja Pin_LED3 sekä yhdistetään komponentit kuvan 30 kaltaisesti. Yhdistäminen tapahtuu valitsemalla *TopDesign*-näkymän vasemmalta puolelta *Wire Tool*, tai painamalla näppäimistöltä pikanäppäintä W.



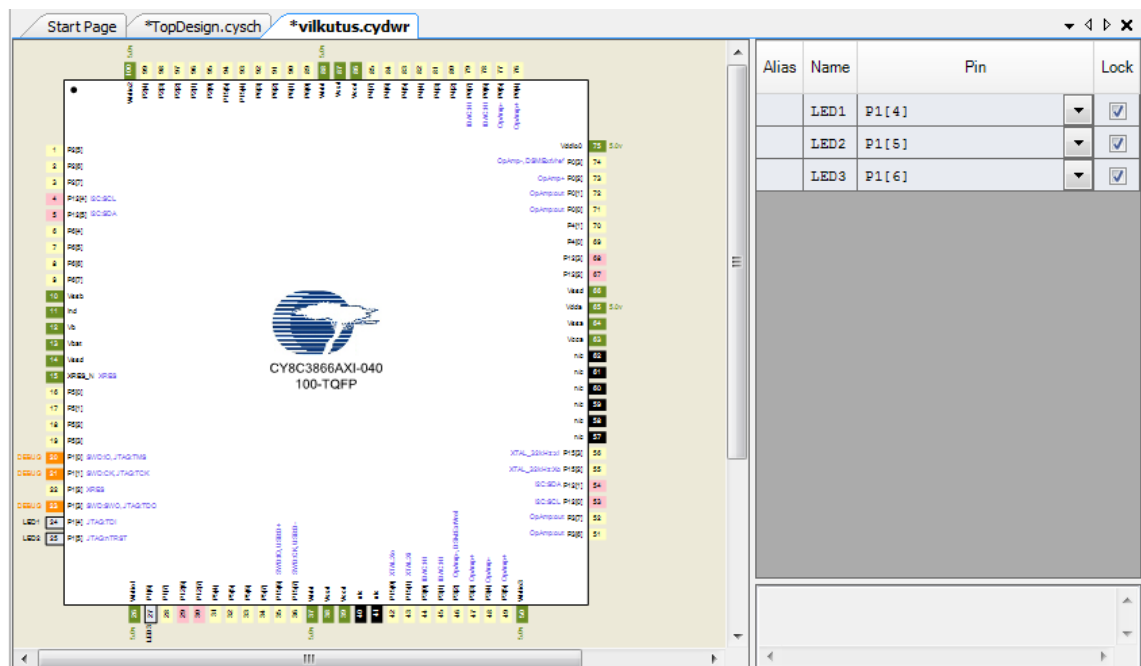
KUVA 30. KytKentä

Ennen kuin ohjelma voidaan kääntää ja ohjelmoida piirille, täytyy määritellä käytettävät nastat, joilla tehdään fyysinen kytkentä. Kaksoisnapsautetaan vasemmalla olevasta *Workspace Explorerista* kohtaa vilkutus.cydwr, kuten kuvassa 31 on esitetty.



KUVA 31. Processorityöympäristön avaaminen

Tehdään avautuneessa uudessa työympäristössä kuvan 32 mukaiset muutokset. Pin_LED1 määritetään jalkaan P1[4], Pin_LED2 jalkaan P1[5] ja Pin_LED3 jalkaan P1[6].

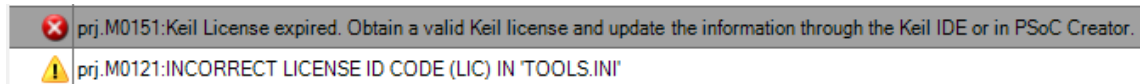


KUVA 32. Processorin jalkojen määrittäminen

Kun asetukset on tehty, voidaan toteuttaa kytkentä koekytkentäalustalle. Yhdistetään hyppylangalla LED1 kohtaan P1[4], kuten äsken oli juuri määritetty. LED2 kytketään kohtaan P1[5] ja LED3 jalkaan P1[6].

Ohjelman kääntäminen

Ohjelma käännetään napsauttamalla *Build*-valikosta ”Build vilkutus”. Mikäli virheitä ei esiinny, hetken päästä heksatiedostoksi käännetty ohjelma on ohjelmoitavissa piirille. Jos tiedostoa ei voida kääntää ja kuvan 33 kaltainen virheilmoitus ilmestyy, lisää lisenssi asennusohjeen kohdan *Lisenssiavaimen lisääminen* mukaisesti.

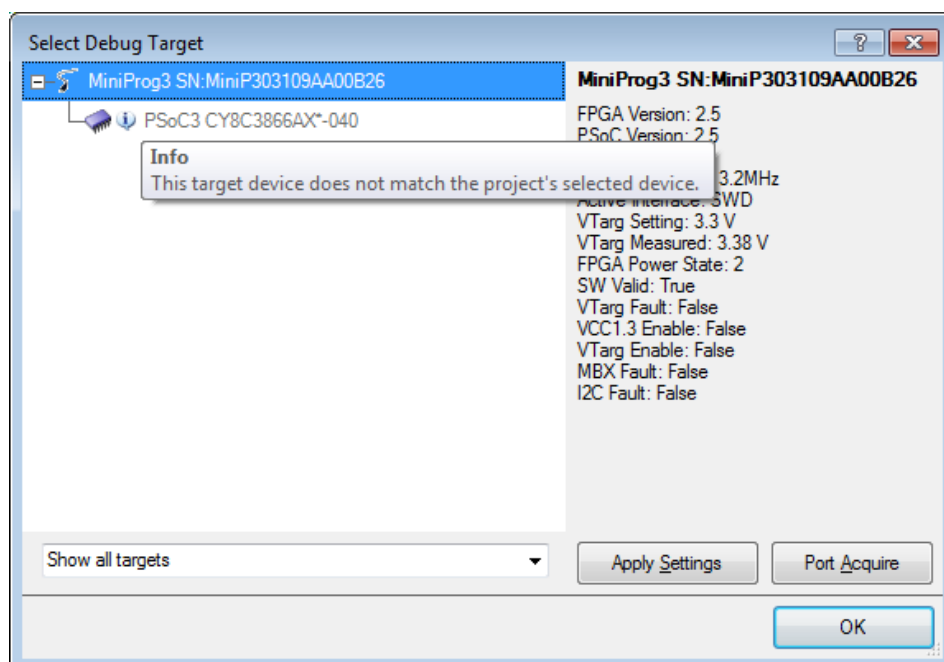


KUVA 33. Lisenssin umpeutumisvirhe

Ohjelmointi piirille

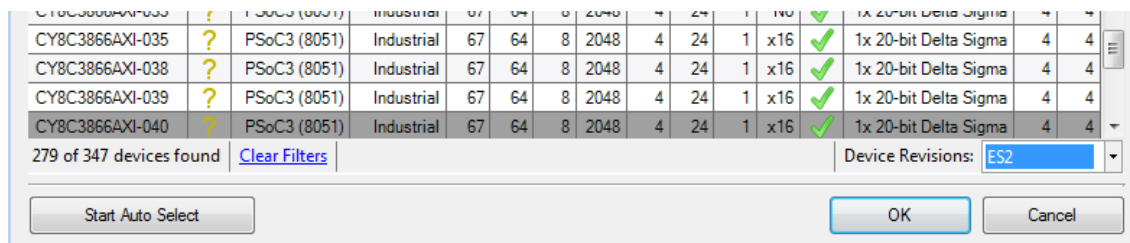
Ennen ohjelmointia piirilevyllä on kytkettävä ulkoinen virtalähde (esimerkiksi 9V paristo). Mikäli projektin ja MiniProg3-työkalun asetukset on määritetty oikein, voidaan piiri ohjelmoida suoraan ohjelmasta valitsemalla työkalurivin *Debug*-valikosta *Program*.

Jos laitteen tunnistus ei kuitenkaan onnistu ja ruudulle ilmestyy kuvan 33 kaltainen virheilmoitus, suljetaan kyseinen ikkuna ja valitaan *Project*-valikosta *Device Selector*-asetukset.



KUVA 33. Tunnistusvirhe

Varmistutaan vielä, että valittuna laitteena on CY8C3866AXI-040. Sen lisäksi kuvan 34 kaltaisen ikkunan oikean alareunan *Device Revisions*-vetovalikosta valitaan **ES2** ja hyväksytään valinnat napsauttamalla *OK*. Nyt piiri voidaan ohjelmoida *Program*-valinnalla tai painamalla pikanäppäimiä *Ctrl + F5*.



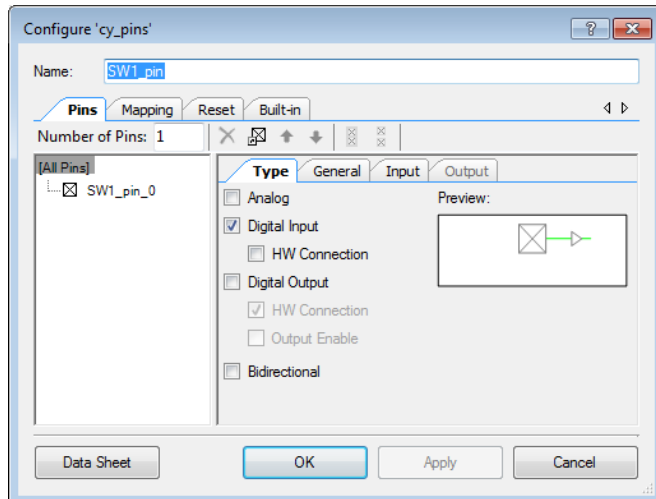
KUVA 34. Laitteistoversion valinta

Vaihtoehtoisesti voidaan käyttää PSoC Programmer-ohjelmaa, jolloin MiniProg3-työkalun asetukset määritetään samoin kuin asennusohjeen kohdassa 3.1.2 on esitetty. Ladattava heksatiedosto löytyy työhakemiston sijainnista `\DP8051_Keil_816\Debug\tiedosto.hex`.

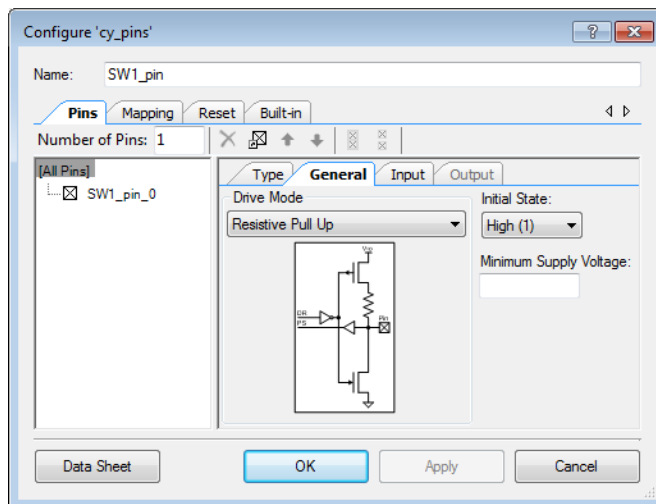
Kun ohjelma on ajettu piirille ja hyppylangat on kytketty oikein, piirilevyllä olevat pinaliitosledit vilkkuvat järjestyksessä LED1, LED2, LED3, kaikki pois päältä, LED1 jne.

Keskeytysohjelman lisäys

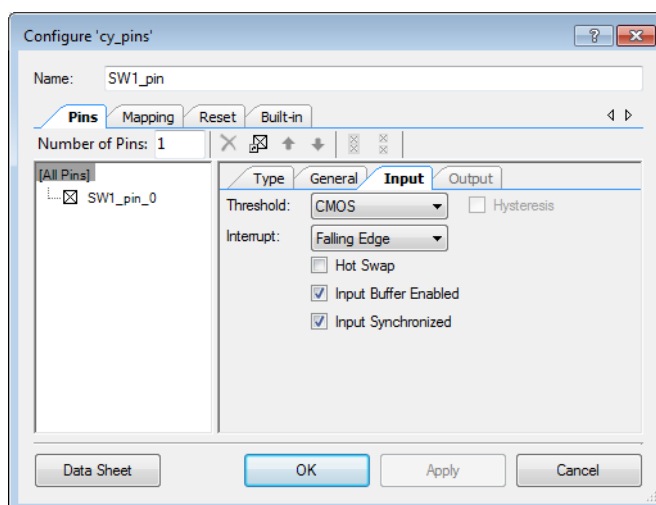
Tehdään vielä ohjelmaan vilkutusnopeuden vaihto. Lisätään TopDesign-ikkunaan komponenttikatalogista *Digital Input Pin* ja *System*-kansioista *Interrupt*. Vaihdetaan *Digital Input Pin*:in ominaisuuksista nimeksi ”SW1_pin” ja otetaan ruksi pois ”HW Connection”-kohdasta (kuva 35). *General*-välilehdeltä vaihdetaan *Drive mode* asetukseen *Resistive Pull Up* (kuva 36) ja *Input*-välilehdeltä vielä keskeytyksen (*Interrupt*) tyyppiä *Falling Edge* (kuva 37).



KUIVA 35. SW1-kytkimen asetukset

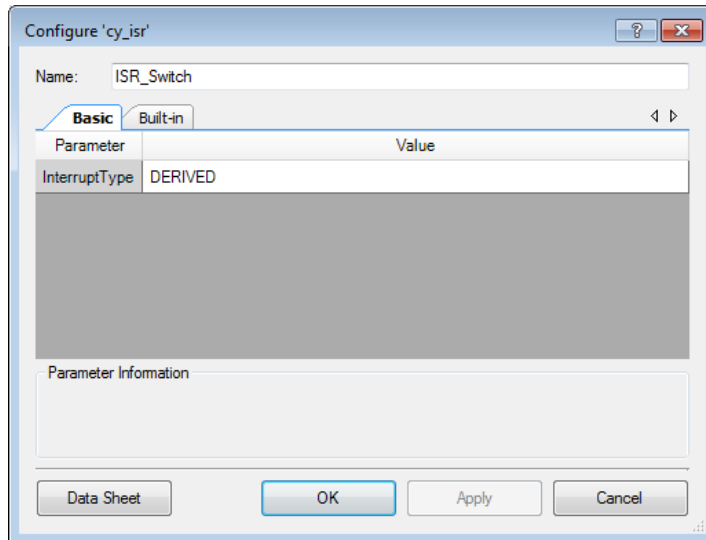


KUIVA 36. SW1-kytkimen drive moden vaihto



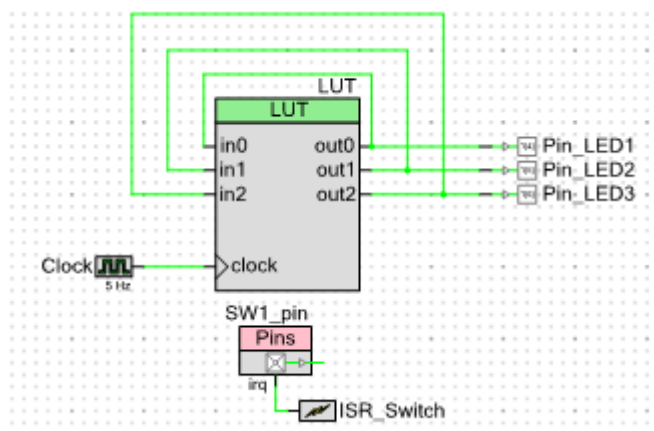
KUIVA 37. SW1-Kytkimen Interrupt-arvon vaihto

Vaihdetaan vielä isr_1-komponentin nimeksi ISR_Switch (kuva 38).



KUVA 38. isr_1-komponentin nimen vaihto

Tämän jälkeen kytketään ISR_Switch ja SW1_pin yhteen kuten kuvassa 39.



KUVA 39. Valmis kytkentä

Lähdekoodin muokkaus

Tähän mennessä ohjelmisto on generoinut kaiken lähdekoodin automaattisesti, mutta keskeytysohjelman toimimiseksi on lisättävä muutama rivi koodia itse. Ensimmäiseksi avataan vasemmalla olevasta valikosta main.c-tiedosto ja lisätään rivit

```
CYGlobalIntEnable; sekä ISR_Switch_Start(); , joilla sallitaan keskeytykset.
```

```

void main()
{
    /* Place your initialization/startup code here (e.g.
    MyInst_Start()) */
    CYGlobalIntEnable;
    ISR_Switch_Start();
    for(;;)

    {
        /* Place your application code here. */
    }
}

```

Seuraavaksi avataan `ISR_Switch.c`-tiedosto, jonka riviä 28 edeltää kommentoituna ”Place your includes, defines and code here”. Kirjoitetaan tähän koodi, jolla kellon ja kytkimen kirjastot saadaan käyttöön:

```

/* `#START ISR_Switch_intc` */
#include "Clock.h"
#include "SW1_pin.h"
/* `#END` */

```

Noin riville 130, `CY_ISR(ISR_Switch_Interrupt)`-aliohjelmaan muokataan seuraavat rivit:

```

CY_ISR(ISR_Switch_Interrupt)
{
    /* Place your Interrupt code here. */
    /* `#START ISR_Switch_Interrupt` */
    static uint16 clkDiv=59;

    Clock_SetDividerValue(clkDiv);

    /* `#END` */
}

```

Lopuksi määritetään `SW1_pin`:lle jalka edelliseen tapaan ja kytketään `SW1`-nastasta hyppylanka siihen määritettyyn jalkaan. Projektin kääntämisen ja piirin uudelleenohjelmoinnin jälkeen kytkintä `SW1` painamalla sekvenssi vaihtaa tilaansa hieman rivakammin. Takaisin alkutilaan päästään painamalla `RESET`-nappia.