



Razvoj ugradbenih sustava s EFM8 mikrokontrolerom – 1. dio

Ivica Čulina¹

Uvod

Sve se češće spominje važnost STEM područja i Interneta Stvari (IoT) u novim promjenama koje slijede u gospodarstvu i društvu. Inženjerska grana Ugradbene elektronike (eng. Embedded engineering) sve je traženije zanimanje kako računala prodiru u sve pore našeg društva. Dok su platforme kao Arduino zgodne za igranje i jednostavan razvoj kao potpora istraživanju u drugim područjima, imaju malo veze s tehnikama i vještinama potrebnim za profesionalni razvoj ugradbenih sustava komercijalne kvalitete. Ukoliko ste zainteresirani za karijeru u ugradbenom inženjerstvu dobra vijest je da danas možete uz vrlo malo financijskog ulaganja (i puno rada) početi sa samostalnim razvojem. Važno je što prije početi programirati na razvojnoj pločici. Kroz iterativni proces eksperimentalnog programiranja (napiši kôd – učitaj na mikrokontroler – koristi debugger) stječemo potrebno iskustvo i sigurnost prilikom rada. Ugrađeni debbuger je vrlo važan za učenje i razvoj – prilikom programiranja mikrokontrolera potrebno je biti vrlo oprezan, dovoljan je jedan krivo upisan registar i kôd ne radi. Debugger nam omogućava “uvid unutra” – možemo izvršavati kôd naredbu po naredbu (eng. single stepping) te iščitati trenutne vrijednosti registara procesora i radne memorije.

Najvažniji prvi korak je krenuti samostalno raditi. Programiranje mikrokontrolera, zbog direktnog rada sa sklopovljem i registrima, može u početku biti dosta frustrirajuće. Napišete kôd, sve izgleda u redu, učitate ga na mikrokontroleru i . . . ništa se ne događa! Svi smo prošli takve situacije i upravo je tu prvu barijeru najbitnije probiti za uspješan daljnji rad. Važno je ne odustati! Tada nakon par sati razmišljanja i listanja tehničke dokumentacije shvatite da ste jedan registar krivo inicijalizirali. Izmijenite kôd – i LED dioda sada svijetli. Kroz (nekoliko) takvih iteracija brzo ćete steći osjećaj za tehniku i sigurnost u radu. Iznenadit ćete se koliko ćete nakon toga brzo doći do naprednijih (i zanimljivijih) projekata. U ovoj seriji radionica programirat ćemo EFM8 mikrokontroler (8051 arhitektura) proizvođača Silicon Labs (Silabs).

Programne ćemo pisati u assembleru (eng. assembly language). Iako imamo dostupan C kompajler za 8051 arhitekturu, nećemo koristiti C jezik iz dva razloga:

- programiranjem u assembleru najbolje usvajamo fundamentalna znanja iz ugradbenog računalstva koja će nam biti od velike pomoći u ozbiljnijim projektima;
- 8051 arhitektura nije pogodna za C jezik.

Ovu radionicu objavit ćemo u tri nastavka. U prvom dijelu obradit ćemo pripremni dio posla, a u sljedećim nastavcima pogledat ćemo detaljnije arhitekturu EFM8 mikrokontrolera i napraviti tri uvodna projekta.

¹ Autor je dipl. ing. elektrotehnik, stručni suradnik na Veleučilištu Velika Gorica; e-pošta: ivica.culina@hotmail.com

Sadržaj radionice

Priprema:

- opis opreme
- nabavka razvojne pločice
- instalacija i namještanje programske podrške
- rad s programskom podrškom i pregled dokumentacije
- osnovni pokazni projekt;

Projekti:

- detaljniji pregled arhitekture EFM8 mikrokontrolera
- služenje tehničkom dokumentacijom
- rad s prekidima (interrupts)
- rad s vanjskim jedinicama.

Oprema

Razvojna pločica

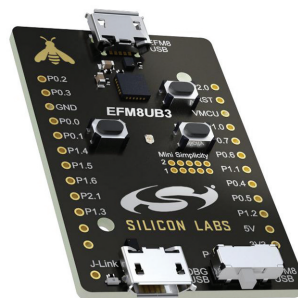
Za početak, dovoljno je imati kvalitetno stolno računalo (kako bi mogli bezbolno raditi u razvojnim programima) i kupiti jednu od pokaznih razvojnih pločica proizvođača (eng. Development board/ kit). Cijene počinju već od 5 €, a optimalna pločica bi se dobila za nekih 15–30 €.

Sa samo tom opremom možete uz dosta rada steći vrlo konkretne vještine razvoja kôda za ugradbene sustave.

Razvojne pločice isporučuje proizvođač mikrokontrolera kako bi inženjerima ubrzao upoznavanje i razvoj s određenim mikrokontrolerom.

Na pločici se nalaze:

1. **Mikrokontroler** čije su izlazne nožice (eng. pins) spojene na spojnik na rubovima pločice (eng. breakout pads) radi lakšeg spajanja s vanjskim uređajima.
2. **Sklopovlje za prilagodbu napajanja** za mikrokontroler (eng. power supply). Mikrokontroler je moguće napajati putem USB kabela preko računala (najčešći odabir prilikom razvoja) ili priključivanjem vanjskog napajanja.
3. **Sklopovlje za debbuger** koje nam znatno olakšava razvoj programa i otklanjanje grešaka. Razvojnu pločicu jednostavno spojimo sa stolnim računalom putem USB kabela. Sklopovlje na pločici i program za razvoj koji isporučuje proizvođač (eng. IDE – Integrated Development Environment) nam omogućavaju da učitamo strojni kôd u Flash memoriju mikrokontrolera te otklanjamo greške u kôdu (izvršavanje kôda naredbu-po-naredbu, čitanje radne memorije, registara, itd.).
4. **Dodatne vanjske ulazno/izlazne jedinice** ovise o pločici koju odaberemo. O broju dodatnih jedinica na pločici ovisi i cijena same pločice. Za početak je bitno nabaviti pločicu koja ima nekoliko tipkala (eng. push-buttons) i LED dioda (eng. LED diodes) kako bismo mogli napisati kôd koji “zbilja nešto radi”. Naprednije pločice mogu dolaziti s ugrađenim LCD ekranom (eng. LCD display), sklopovljem za radiokomunikaciju, itd. Naravno, možemo i sami spajati dodatno sklopovlje na mikrokontroler putem spojnika (breakout pads) na pločici. O tome detaljnije u nastavku.



Slika 1. Primjer osnovne razvojne pločice (cijena oko 5 €).

Za nabavku razvojne pločice preporučujem američkog distributera Mouser (<https://hr.mouser.com>). Kompanija ima skladišta u Europi, vrlo su profesionalni, te imaju kvalitetnu internet stranicu za online narudžbe. Pogledajte trenutnu ponudu *Silicon Labs* pločica na stranici Mouser distributera: <https://hr.mouser.com/c/embedded-solutions/engineering-tools/embedded-processor-development-kits/development-boards-kits-8051/?m=Silicon%20Laboratories>

Slobodno me kontaktirajte putem e-maila prilikom odabira pa možemo zajedno prodiskutirati vaš odabir.

Dodatna oprema za laboratorij

Iako možete krenuti samo s razvojnom pločicom, preporučujem da od dodatne opreme nabavite barem multimeter. S njime možete otkloniti 90 % problema u radu. Dodatna oprema za početak nije potrebna, no svejedno ću je u daljnjem navesti za referencu.

Za nabavku dodatne opreme neću navesti specifične izvore. Imamo veliki izbor online trgovina, te nekoliko domaćih trgovina. Google vam je prijatelj.

1. *Multimeter* (eng. *Multimeter*)

Svakako preporučujem nabavku multimetra. Dovoljan će biti osnovni uređaj koji se danas može vrlo jeftino nabaviti doslovce na svakom koraku. On nam omogućuje mjerenje osnovnih veličina kao što su napon, struja i spojenost vodova. Vrlo je koristan prilikom otklanjanja grešaka u kôdu i elektroničkom sklopovlju.



Slika 2. Osnovni multimeter.



Slika 3. Lemna stanica (Hakko 936 klon).

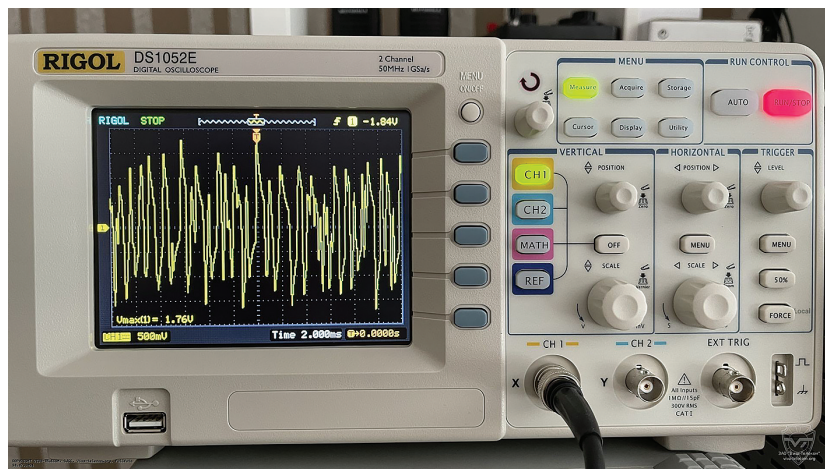
2. *Lemna stanica ili lemilica* (eng. *Soldering station, Soldering iron*)

Kako napredujete u vještini, vrlo brzo ćete poželjeti spojiti dodatno sklopovlje na razvojnu pločicu. Dodatno sklopovlje možete razvijati sami, no u današnje vrijeme preporučujem kupovinu gotovih modula (gotovih tiskanih pločica). Putem online trgovina (npr. <http://www.adafruit.com/> ili www.mikroe.com) možete pronaći cijeli niz gotovih ulazno-izlaznih modula za različite uporabe (pogon elektromotora, releji, senzori, ...). No te module morate spojiti žicama s razvojnom pločicom. Te žice zalemite na izlazne spojnik (breakout pads) na razvojnoj pločici. To možete obaviti običnom lemlicom, no preporučujem lemnu stanicu, jer je ugodnija i sigurnija za rad, te omogućava kontrolu temperature. Vrlo dobre i povoljnije lemne stanice su klonovi "stare" Hakko stanice 936 (ili 937/8). Možete ih pronaći online za nekih 20–30 €.

3. Digitalni osciloskop (eng. Digital oscilloscope, scope)

Ovo je najskuplji dio opreme i svakako nije nužan na početku. Osciloskop omogućava uvid u elektroničke signale. Vrlo je koristan kod otklanjanja greški u komunikacijskim protokolima i provjeri izlaznih signala na pločici. Ako se odlučite ozbiljnije baviti ugradbenim sustavima i elektronikom nabavite osciloskop i naučite se služiti s njime. Neće vam biti žao.

Za razvoj ugradbenih sustava sasvim su dovoljni osnovni modeli, čija se cijena kreće oko 200–300 €.



Slika 4. Osnovni Rigol digitalni osciloskop.

EFM8 mikrokontroler i razvojna pločica

Za početak rada na ugradbenim sustavima odabrao sam EFM8 mikrokontroler američkog proizvođača SiliconLabs (Silabs). Riječ je o malom 8-bitnom mikrokontroleru baziranom na Intel 8051 procesoru. U praksi su 32-bitni mikrokontroleri bazirani na ARM Cortex-M procesorima u zadnjih par godina gotovo izbacili 8-bitne mikrokontrolere zbog sve manje razlike u cijeni, čineći mnoge prije standardne mikrokontrolere gotovo izumrlima (primjerice PIC i AVR). S druge strane, EFM8 mikrokontroleri su zbog svoje vrlo niske cijene i visokih performansi još uvijek nezamjenjivi za tradicionalne 8-bitne aplikacije. Tako imamo priliku raditi s mikrokontrolerom koji je vrlo tražen u sadašnjoj praksi, a opet nam, kako će biti objašnjeno u nastavku, omogućava jednostavniji ulazak u svijet ugradbenog računarstva.

Iako je ARM arhitektura zapravo jednostavnija za programiranje od 8051 arhitekture, za početnika je puno pametnije krenuti s jednostavnijim 8051 mikrokontrolerom (EFM8). Razlog tomu je što su 8-bitni mikrokontroleri puno manji i interno jednostavniji od 32-bitnih ARM mikrokontrolera. Primjera radi, tehnička dokumentacija (eng. reference manual) za EFM8 mikrokontroler se sastoji od nekih 350 stranica, dok ista dokumentacija za ARM-Cortex-M4 mikrokontroler sadrži 2-3 tisuće stranica, što je za početnika često previše informacija. Kada dobro svladate neki jednostavniji mikrokontroler vrlo ćete lako preći na kompleksnije uređaje jer se većina iskustva iznenađujuće lako prenese na druge procesore.

Za ovu radionicu odabrana je razvojna pločica EFM8BB2-SLSTK2021A iz razloga što je trenutno imamo na radnom stolu. Odabrana pločica ima dobar broj vanjskih

jedinica (joystick, tipkala, LED diode, LCD ekran), a cijena je oko 25–30 €. Pokazani kôd i način rada bit će gotovo jednaki za bilo koju razvojnu pločicu baziranu na EFM8 mikrokontroleru (potrebne preinake kôda bit će navedene u tekstu). Konkretni mikrokontroler na mojoj pločici je EFM8-BB2-2F16G. Pogledajmo detaljnije proizvođački naziv ovog konkretnog mikrokontrolera.

EFM8 označava liniju mikrokontrolera s istom jezgrom (procesorom), u ovom slučaju riječ je o 8051 arhitekturi. Oznaka BB2 specificira familiju mikrokontrolera, daljnje oznake specificiraju količinu memorije, itd.

Nije bitno koji ćete specifični mikrokontroler iz EFM8 linije imati na svojoj pločici. Razlike su čisto tehničke prirode (brzina procesora, broj i tip vanjskih jedinica, količina memorije). Pokazani kôd će jednako raditi na svim uređajima uz male preinake koje ćete lako sami uraditi uz konzultaciju s tehničkom dokumentacijom za vaš konkretni mikrokontroler (pokazat ćemo kako).

U daljnim koracima pretpostavit ćemo da je odabrana ista pločica koju imamo (EFM8BB2-SLSTK2021A), no ponovit ćemo iste korake, samo s drugim pločicama.

1. Priprema za rad

Pripremimo računalo za rad s projektima, skinimo potrebnu tehničku dokumentaciju i instalirajmo programsku podršku za razvoj.

Direktorij (folder) s projektima

Stvorimo novi folder gdje će se nalaziti naši projekti (nazovite ga po želji, *mcu_projects* ili slično).

Unutar njega stvorite folder naziva *efm8*. Tu će se nalaziti svi naši projekti (kôd) i dokumentacija za EFM8 mikrokontrolere. Unutar foldera *efm8* stvorite folder *DOCS* za dokumentaciju.

```
/mcu_projects
```

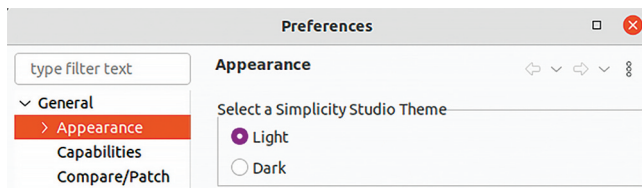
```
  /efm8
```

```
    /DOCS
```

Instalacija programske podrške

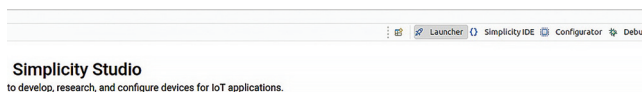
Silicon Labs – Silabs (proizvođač EFM8 mikrokontrolera) nudi svoje *integrirano razvojno okruženje* (eng. *IDE – Integrated Development Environment*). *IDE* nam nudi cijelu programsku podršku (toolchain, builder, text editor, debugger,...) na jednom mjestu. Silabsov *IDE* se zove *SimplicityStudio* i baziran je na open-source programu Eclipse (<http://www.eclipse.org/ide/>). U trenutku pisanja ovog članka, najnovija verzija je v5. Skinimo je sa stranice proizvođača i instalirajmo.

Posjetimo <http://www.silabs.com/developers/simplicity-studio> te skinimo instalacijsku datoteku za naš operacijski sustav (trenutno dostupno za Win10, Ubuntu linux i macOS). Pokrenimo *SimplicityStudio* (moj operacijski sustav je Ubuntu). Slike programa (screenshots) date u članku većinom su učinjenje sa svijetlom temom (*Light Theme*) radi bolje čitljivosti na papiru. U našem radu preporučuje se odabir tamne teme (*Dark Theme*) radi znatno manjeg zamora očiju prilikom dugotrajnog rada. Postavke možete izmijeniti pod *Window* → *Preferences* (*General* > *Appearance*).



Slika 5. Tamna/svijetla tema.

Nakon pokretanja programa dočekat će nas Welcome ekran.



Slika 6. Početni ekran.

Napomena. Ukoliko Studio prilikom pokretanja pita za odabir Workspace direktorija, odaberimo predhodno stvoreni efm8 folder.

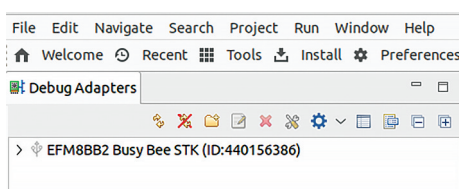
Obratite pozornost na dio ekrana gore desno na slici 6. Imamo tabove (ikonice) Launcher, Simplicity IDE, itd. Pomoću njih u Eclipse okruženju mijenjamo trenutni prikaz ekrana i zovu se *Perspectives*. Uvjerite se da je trenutno odabrana perspektiva *Launcher*.

Spojimo razvojnu pločicu s računalom putem USB kabela isporučenog s pločicom. To će biti glavna veza IDE-a s razvojnom pločicom preko koje ćemo učitavati strojni kôd u FLASH memoriju i pogoniti debugger. Nakon spajanja program će automatski prepoznati pločicu i instalirati potrebne drivere za pogon USB sučelja (potrebno je malo pričekati da se driveri instaliraju). Nakon uspješnog prepoznavanja pločice bit će prikazan naziv pločice pod *Connected Devices* (slika 7).



Slika 7. Prepoznata razvojna pločica.

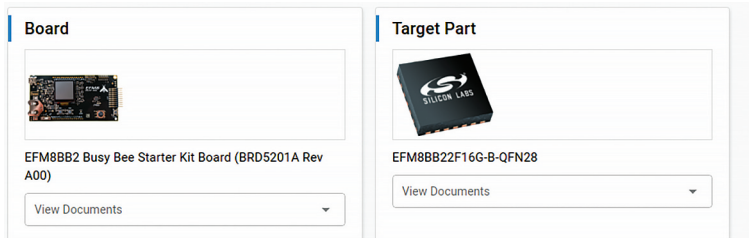
Na lijevoj strani ekrana također će biti prikazana naša pločica kao *Debug Adapter* (slika 8).



Slika 8. Razvojna pločica kao debug adapter.

Pritisnite dugme **Start**.

Prozor će se izmijeniti te će biti prikazani nazivi vaše razvojne pločice i specifičnog mikrokontrolera unutar EFM8 linije koji se nalazi na razvojnoj pločici. Klikom na padajući izbornik *View Documents* ispod **Board** i **Target Part** dobivate popis dostupne dokumentacije za specifičnu pločicu i mikrokontroler (slika 9).



Slika 9. Dokumentacija za razvojnu pločicu i dati mikrokontroler.

Otvorite i spremite u predhodno stvoreni folder *DOCS* sljedeće dokumente:

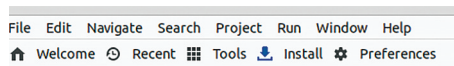
- *EFM8BB2-SLSTK2021A User's Guide* Opis razvojne pločice te prikaz kako je dodatno sklopovlje koje se nalazi na pločici (npr. Tipkala i LED diode) spojeno na nožice mikrokontrolera. Po želji i predhodnom znanju možete također skinuti dodatnu dokumentaciju za razvojnu pločicu (elektroničku shemu, popis materijala, . . .).
- *EFM8BB2-datasheet* Datasheet za familiju mikrokontrolera (EFMB-BB2) pokazuje električne karakteristike mikrokontrolera i detaljniji opis karakteristika specifičnog uređaja u familiji. Za početak rada nije toliko bitan dokument. Jedino za sada možete pogledati karakteristike za vaš specifični uređaj u tablici na početku kako biste vidjeli njegove karakteristike.
- *EFM8BB2-Reference Manual (rm)* Najbitniji dokument s kojim ćete stalno raditi prilikom razvoja kôda. Unutra je detaljno opisana unutrašnjost mikrokontrolera s programskim modelom (eng. programming model).

Napomena. Ukoliko iz nekog razloga ne možete pristupiti nekom dokumentu uvijek ga možete pronaći Google pretraživanjem ili na stranicama proizvođača <http://www.silabs.com/>. Za dodatnu uputu oko korištenja Simplicity Studio programa skinite dokument *Simplicity Studio 5 User Guide* s <https://docs.silabs.com/simplicity-studio-5-users-guide/1.0/>.

Kako je Simplicity Studio zapravo Eclipse IDE (Eclipse CDT) s dodanim pluginovima, za sve nejasnoće oko rada u programu možete putem Googla tražiti sličan problem za Eclipse CDT (puno više sadržaja dostupno je na internetu nego za Simplicity Studio)

Nadalje, potrebno je instalirati kompajler (toolchain) i SDK (Software Development Kit) za odabrani mikrokontroler.

Kliknite na ikonicu **Install** u alatnoj traci (gore lijevo), slika 10.



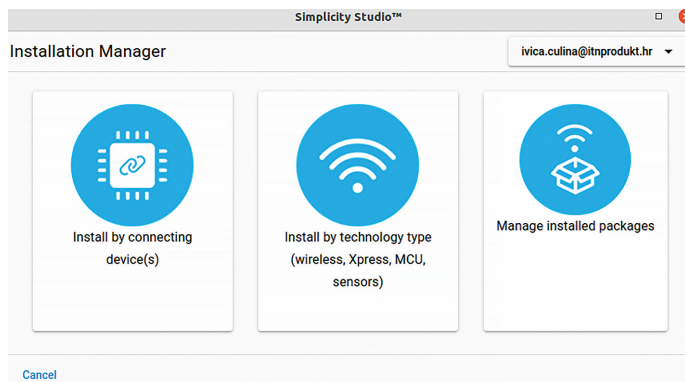
Slika 10. Install ikona na alatnoj traci.

Otvorit će se prozor, slika 11.

Kliknite na prvu ikonu **Install by connecting device(s)**.

Odaberite vašu pločicu (bit će prikazana na ekranu), pritisnite **Next**. Zbog jednostavnosti odaberite **Auto** opciju i pokrenite instalaciju.

Nakon završetka provjerite instalaciju pritiskom na ikonu **Managed installed packages**.



Slika 11. Instalacija programske podrške.

Provjerite da su sljedeći programi ispravno instalirani:

Keil 8051 Toolchain – Ovo su toolchain (kompajler, asembler, linker) i dodatni programi pomoću kojih ćemo stvoriti strojni kôd za učitavanje u mikrokontroler.

Silicon Labs nudi besplatnu licencu za 8051 toolchain kompanije Keil (u stvari standardni industrijski kompajler za 8051 arhitekturu). Kako biste dobili licencu registrirajte se (bez registracije dobivate verziju s ograničenjem na veličinu stvorenog programa koja je dovoljna za našu radionicu). Ukoliko želite dobiti punu verziju pređite na *Help* → *Licensing* → *Keil 8051* te slijedite upute.

Dokumentaciju za Keil toolchain možete pronaći na https://www.keil.com/support/man_c51.htm.

Nama su zanimljivi sljedeći programi unutar toolchaina:

- **Ax51** (assembler)
- **LX51** (linker).

Također primijetite da na stranici s dokumentacijom imate link *8051 Instruction Set Manual* s dobrim opisom instruktorskog skupa za 8051 procesor.

8051 SDK – Ovo je skup izvornih datoteka (source files) koji sadrže već napisan kôd koji isporučuje proizvođač. Kao što ćemo vidjeti kasnije, trebat će nam radi datoteka koje definiraju registre vanjskih jedinica.

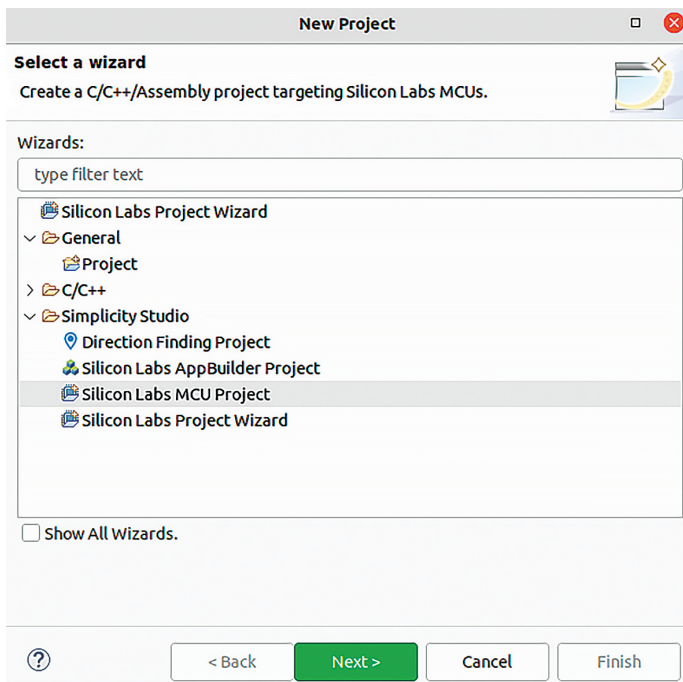
2. Novi projekt

Napomena. Ukoliko već niste odabrali *Workspace*, odaberite *File* → *Switch Workspace...*, te tamo odaberite folder */efm8*.

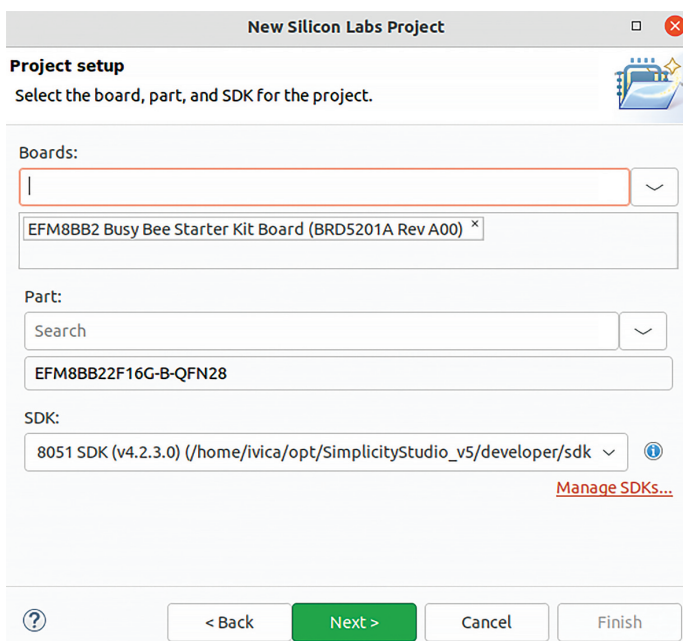
Workspace je pojam iz Eclipse IDE-a koji grupira projekte zajedno. Dobro je imati zaseban workspace za svaki mikrokontroler s kojim radimo jer ćemo imati isti skup postavki IDE-a (preferences) za svaki projekt.

Odaberite *File* → *New* → *Project*.

U izborniku (pod *SimplicityStudio*) odaberite *Silicon Labs MCU Project*.



Slika 12. Stvaranje novog projekta (1).



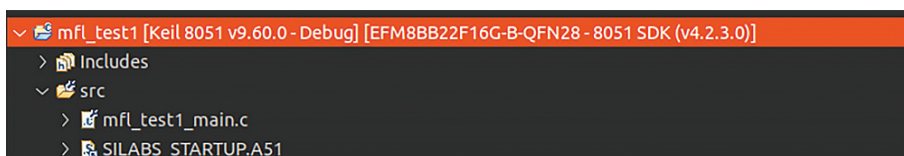
Slika 13. Stvaranje novog projekta (2).

Na sljedećem prozoru sve postavke bi trebale biti automatski ispravno odabrane, provjerite vaše prema slici 13.

Nastavite, te na kraju odaberite opciju *Empty C Project*.

Nazovite projekt *mfl_test1* odaberite opciju *Link sdk and copy project sources*.

Nakon što je stvoren novi projekt biti će prikazan kao folder na desnoj strani ekrana (Project explorer)

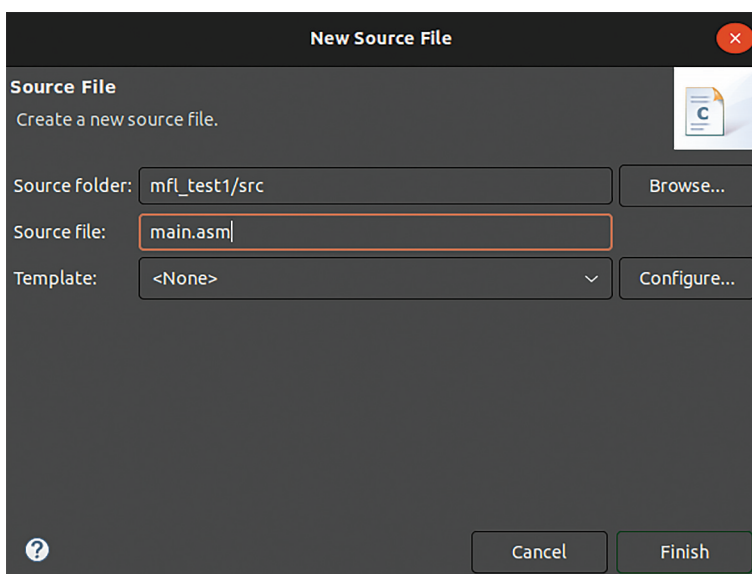


Slika 14. Stvoreni projekt u Project Exploreru.

Izbrišite dvije automatski stvorene datoteke (*mfl_test1_main.c* i *SILABS_STARTUP.A51*).

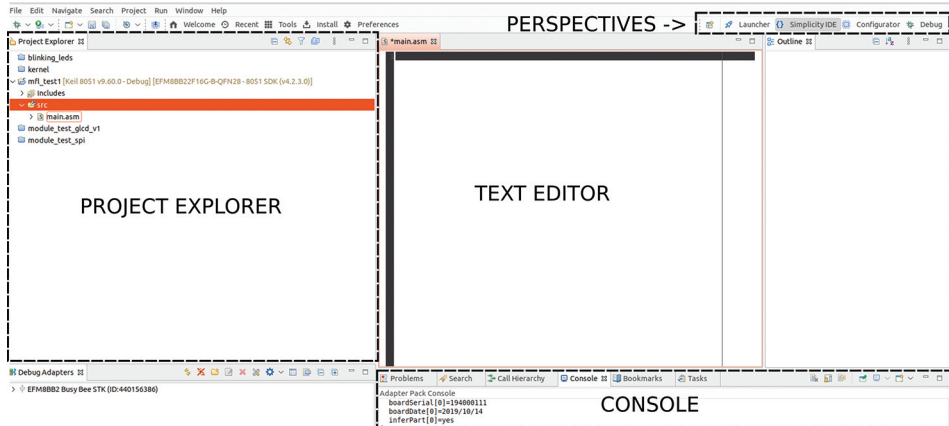
Stvorite novu datoteku (source file) desnim klikom na *src* direktorij odaberite *New* → *Source file*.

File nazovite *main.asm* (ime možete odabrati po želji, bitan je nastavak *.asm* koji govori toolchainu da je riječ o source fileu koji sadrži asemblerski kôd).



Slika 15. Nova izvorna datoteka (source file).

Pogledajmo sada izgled ekrana IDE-a na slici 16 i uočimo glavne dijelove.



Slika 16. Elementi ekrana radnog okruženja u SimplicityIDE perspektivi.

Project explorer – slično kao standardni file manager na našem računalu. Pokazuje projekte u otvorenom workspaceu I izgledom odgovara datotekama na hard-disku. Možemo raditi kao i sa svim datotekama na računalu, kreirati i brisati datoteke, direktorije, kopirati, lijepiti, itd. Trenutno aktivan projekt je posebno označen (projekte možemo otvoriti i zatvarati desnim klikom miša *Open project/Close project*).

Text editor – standardni program za uređivanje teksta. Ovdje pišemo naš program.

Perspectives – niz tabova koji mijenjaju prikaz, odnosno radnu površinu, ovisno o zadacima koje trenutno obavljamo. Pritiskom na odgovarajuću ikonu odabiremo novu perspektivu. Dvije najvažnije perspektive su *SimplicityIDE* i *Debug*. Kada pišemo kôd nalazimo se u **SimplicityIDE** perspektivi (trenutno prikazana na ekranu)

Console – ovdje vidimo izlazne informacije kompajlera. Tu ćemo vidjeti da li je naš kôd uspješno kompajliran ili su nađene greške.

Ovdje ćemo završiti prvi dio radionice. U drugom dijelu nastavit ćemo s pokaznim projektom te napisati prvi kôd za mikrokontroler. Naučit ćemo se služiti s debuggerom te pogledati detaljnije arhitekturu EFM8 mikrokontrolera.