
STM32F4 DISCOVERY BOARD

TOOLCHAIN SETUP

Autor: **Dr. Sci. Asmir Gogić**

1 Geany IDE setup

Implementacija komunikacijskih sistema, protokola, modulacija i algoritama biti će izvođena na STM32F4 Discovery (ARM Cortex M4) razvojnoj platformi. Kompajliranje koda za datu platformu izvodit ćemo sa arm-gcc kompajlerom na Ubuntu 14.04 LTS distribuciji Linux OS-a. Procedura za instalaciju neophodnog toolchain-a (compiler, libraries) kao i software-a za spremanje firmware-a na ARM MCU biti će opisana u nastavku. Uz pretpostavku da ste instalirali Ubuntu 14.04 LTS, za programiranje STM32F4 Discovery platforme potrebno je provesti sljedeće korake

1. Instalirati *geany* file editor tako što ćemo otvoriti terminal i izvršiti komandu

```
sudo apt-get update
```

```
sudo apt-get install geany
```

Otvoriti *geany* a zatim iz menija odabrati
Edit→Preferences→Tools. U polju *Terminal* sadržaj

```
x-terminal-emulator -e "/bin/sh %c"
```

zamjeniti sa

```
gnome-terminal -x "/bin/sh" %c
```

Spremiti promjene i zatvoriti geany. Sa komandne linije izvršiti sljedeći kod

```
sudo apt-get install libvte-dev
```

Nakon završetka instalacije, pokrenuti *geany* i uočiti da se u donjem lijevom uglu nalazi tab sa nazivom **Terminal**.

2. Kreirati folder **msut** a zatim se u terminalu postaviti na putanju datog folder (u nastavku sve komande će biti date za Linux korisnika *student*)

```
mkdir /home/student/msut
```

```
cd /home/student/msut
```

3. Download-ovati STM32F407 SDK sa sljedećeg [linka](#) u folder **msut** i ekstrahovati ga
4. Postaviti arm-gcc na sistemsku PATH putanju tako što ćemo prvo otvoriti **.bashrc** file

```
gedit /home/student/.bashrc
```

Na kraju datog file-a u novom redu dodati sljedeću liniju koda

```
export PATH=$PATH:/home/student/msut/STM32F407/gcc-arm-none-eabi/bin
```

Spremiti file i osvježiti putanje sa komandom

```
source /home/student/.bashrc
```

Da bi smo provjerili vidljivost kompajlera u terminlu pozvati komandu

```
arm-none-eabi-gcc -v
```

Ukoliko dobijete ispis na terminalu da se radi o gcc verziji 4.8.4 20140526 možete preći na sljedeći korak u suprotnom pogledajte prethodne korake pažljivo da niste neki od koraka preskočili. Ukoliko mislite da ste sve uradili prema uputama a i dalje imate problem kontaktirajte predmetnog nastavnika ili asistenta putem e-mail-a.

-
5. Otvoriti file main.c sa putanje /home/student/msut/STM32F407/examples/blink u *geany* editoru. Zatim u *geany* terminalu prebaciti se na datu putanju sa komandom

```
cd /home/student/msut/STM32F407/examples/blink
```

Kako bi smo provjerili ispravnost cijelog toolchain-a, u terminalu *geany*-a pozvati sljedeće komande

```
make clean
```

kako bi izbrisali sve asm file-ove (*.s) i sam firmware (*.hex i *.bin) a zatim pozvati komandu

```
make
```

Ukoliko dobijete ispis¹

```
arm-none-eabi-objcopy -O ihex main.elf main.hex
arm-none-eabi-objcopy -O binary main.elf main.bin
ls -l main.bin
-rwxrwxr-x 1 student student 3484 Nov 5 13:23 main.bin
```

ispravno ste podesili vaš toolchain i kompajlirali vaš prvi primjer blink.

6. Instalirati neophodne linux sistemske biblioteke

```
sudo apt-get install build-essential git flex bison libgmp3-dev libmpfr-dev
sudo apt-get install libncurses5-dev libhidapi-hidraw0
```

```
sudo apt-get install libmpc-dev autoconf texinfo libtool libftdi-dev
sudo apt-get install libusb-1.0-0-dev zlib1g zlib1g-dev python-yaml
sudo apt-get install libc6-i386
```

7. Sljedeće što trebamo instalirati je aplikacija za spremanje našeg firmware-a (*.bin) u flash memoriju mikrokontrolera. U tu svrhu ćemo koristiti **stlink** aplikaciju koju možete naći unutar folder msut/STM32F407. Pozicionirati se u terminalu na datu putanju

```
cd /home/student/msut/STM32F407/stlink
```

a zatim izvršiti sljedeće komande

```
make clean
```

```
./autogen.sh
```

```
./configure
```

```
make
```

Kopirati udev rule kako bi mogli pristupiti STM32F4 Discovery ploči

```
sudo cp 49-stlinkv2.rules /etc/udev/rules.d/
```

8. Spojiti STM32F4 Discovery kit sa USB kablom a zatim pozvati komandu

```
/home/student/msut/STM32F407/stlink/st-util
```

Kao rezultat dobit ćete ispis u obliku

¹Zanemariti vrijeme kompajliranja file-ova.

```
2014-11-05T13:43:33 INFO src/stlink-common.c: Loading device
parameters....
2014-11-05T13:43:33 INFO src/stlink-common.c: Device connected is: F4
device, id 0x10016413
2014-11-05T13:43:33 INFO src/stlink-common.c: SRAM size: 0x30000 bytes
(192 KiB), Flash: 0x100000 bytes (1024 KiB) in pages of 16384 bytes
2014-11-05T13:43:34 INFO gdbserver/gdb-server.c: Chip ID is 00000413, Core
ID is 2ba01477.
2014-11-05T13:43:34 INFO gdbserver/gdb-server.c: Target voltage is 2917
mV.
2014-11-05T13:43:34 INFO gdbserver/gdb-server.c: Listening at *:4242...
```

Iskopčajte USB kablo sa razvoje ploče a zatim ponovo spojite (ili izvršite kombinaciju tipki ctrl+c).

9. Sada se možemo vratiti u naš *geany* terminal i postaviti putanju

```
cd /home/student/msut/STM32F407/examples/blink
```

i pozvati komandu za spremanje firmware-a našeg **blink** programa u flash memoriju mikrokontrolera

```
make upload
```

Kao rezultat dobit ćete

```
2014-11-05T13:48:12 INFO src/stlink-common.c: Finished erasing 1 pages of
16384 (0x4000) bytes
2014-11-05T13:48:12 INFO src/stlink-common.c: Starting Flash write for
F2/F4
2014-11-05T13:48:12 INFO src/stlink-common.c: Successfully loaded flash
loader in sram size: 3484
2014-11-05T13:48:12 INFO src/stlink-common.c: Starting verification of
write complete
2014-11-05T13:48:12 INFO src/stlink-common.c: Flash written and verified!
jolly good!
```

Pritisnite reset prekidač na ploči (crni push-button) i blink primjer će biti izvršen (4 LED diode na razvojnoj ploči će se periodično isključivati i uključivati).

10. Dodatno možemo instalirati **picocom** program koji omogućuje štampanje poziva *printUSART2()* funkcije preko USB2UART dongle-a

```
sudo apt-get install picocom
```

Primjer poziva za ttyUSB0 uređaja (provjeriti pod kojim imenom se vidi sa komandom `ls /dev/` nakon spajanja dongle-a) pri bitskoj brzini od 9600 bita

```
picocom /dev/ttyUSB0 -b 9600
```

11. Sada kada imamo **picocom** i USB2UART dongle možemo probati primjer sa USART-om

```
cd /home/student/msut/STM32F407/examples/usart
```

```
make clean
```

```
make
```

```
make upload
```

Resetovati STM32F4 ploču nakon čega će zelena LED periodično uključivati/isključivati. Spojiti USB2UART dongle na PC i povezati Rx pin na dongle sa PA2 pinom na STM32F4 ploči. Ovoriti novi terminal i pozvati komandu

```
picocom /dev/ttyUSB0 -b 9600
```

Na terminalu ćete dobiti ispise kao

```
integer varijabla utmp32: 0
float varijable ftmp: 1.100e0
integer varijabla utmp32: 1
float varijable ftmp: 2.500e0
integer varijabla utmp32: 2
float varijable ftmp: 3.900e0
integer varijabla utmp32: 3
float varijable ftmp: 5.300e0
```

12. Za više detalja preporučuje se sljedeća literatura

- [STM32F4xx Reference manual](#)
- [STM32F4 Discovery User Manual](#)
- [STM32F407xx Datasheet](#)

2 Eclipse IDE setup

Ponoviti kompletnu proceduru u okviru sekcije **Geany IDE setup** (ukoliko ista nije sprovedena) a zatim izvršiti sljedeće korake

1. Download-ovati Eclipse Mars sa sljedećeg [linka](#) i ekstrahovati sadržaj u folder

```
/home/student/msut/eclipse
```

2. Instalirati java-jre koristeći sljedeću komandu

```
sudo apt-get install openjdk-7-jre
```

3. Pokrenuti eclipse sa odgovarajuće putanje

```
/home/student/msut/eclipse
```

4. U okviru menija odabrati

```
Help→Install New Software...
```

a zatim dodati u *Work with* sljedeći tekst

```
GNU ARM Eclipse Plug-ins - http://gnuarmecclipse.sourceforge.net/updates
```

a na listi koja se pojavi ispod odabrati sljedeći software

```
GNU ARM C/C++ Generic Cortex-M Project Template
GNU ARM C/C++ Packs
GNU ARM C/C++ OpenOCD Debugging
```

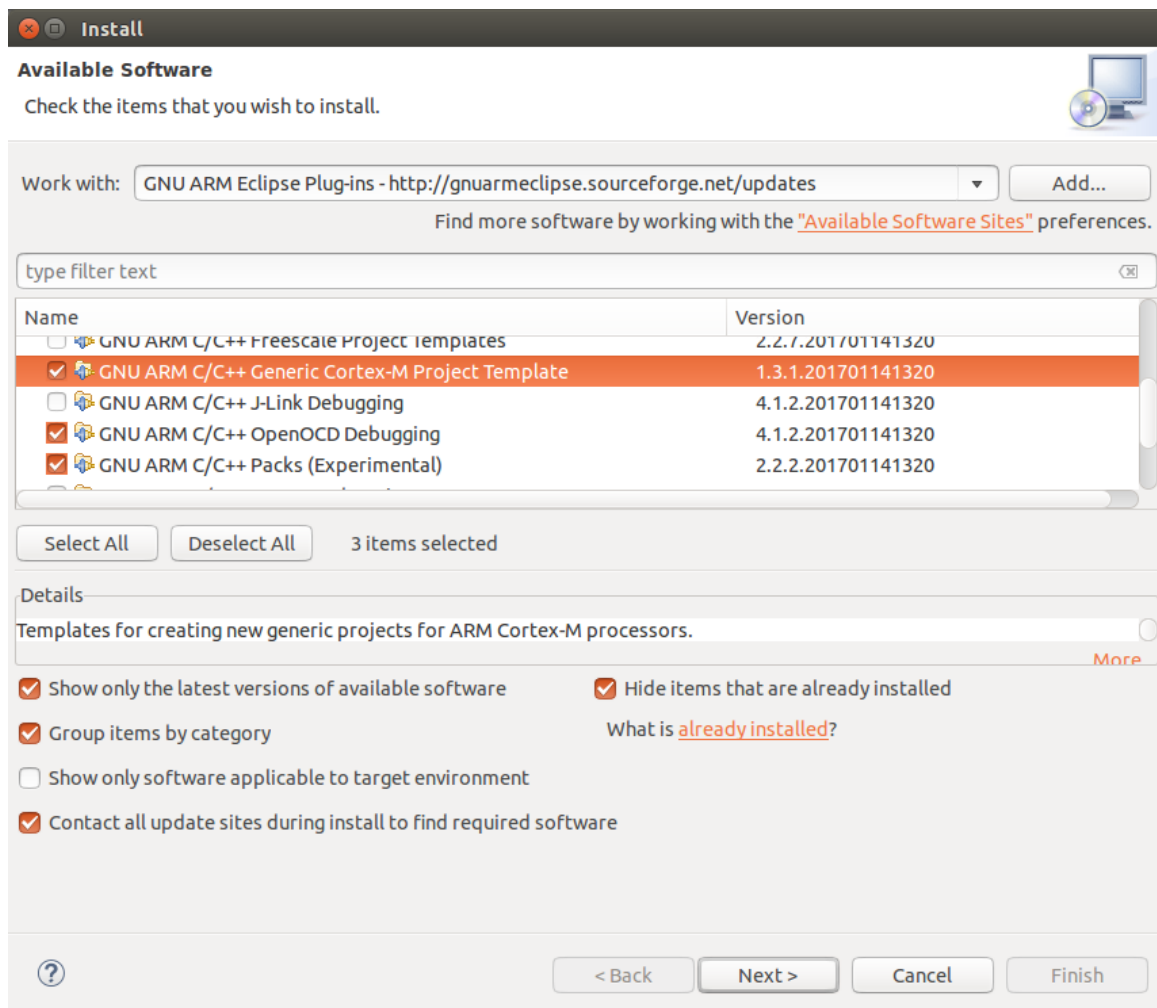


Figure 1: Postavke instalacije Eclipse

U toku instalacije dati potvrđan odgovor na sve upite.

5. U terminalu, postaviti se na putanju

```
cd /home/student/msut/STM32F407/openocd
```

a zatim izvršiti sljedeće komande

```
make clean
./configure
make
sudo make install
```

6. Provjeriti je ispravnost aplikacija openOCD tako da pozovemo komandu `make test-com` unutar folder-a

/home/asmir/msut/STM32F407/examples/usart

```
asmir@asmir:~/msut/STM32F407/examples/usart$ make test-com
../../openocd/src/openocd -f interface/stlink-v2.cfg -c"transport select hla_swd"
-f target/stm32f4x.cfg -c "init"
Open On-Chip Debugger 0.9.0 (2017-02-07-15:24)
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html
hla_swd
```

```

Info : The selected transport took over low-level target control. The
results might differ compared to plain JTAG/SWD
adapter speed: 2000 kHz
adapter_nsrst_delay: 100
none separate
Info : Unable to match requested speed 2000 kHz, using 1800 kHz
Info : Unable to match requested speed 2000 kHz, using 1800 kHz
Info : clock speed 1800 kHz
Info : STLINK v2 JTAG v14 API v2 SWIM v0 VID 0x0483 PID 0x3748
Info : using stlink api v2
Info : Target voltage: 2.879154
Info : stm32f4x.cpu: hardware has 6 breakpoints, 4 watchpoints

```

čime smo verificirali da PC komunicira preko programatora sa STM32F407 MCU. Zaustaviti openOCD sa ctrl+c.

7. Uključiti postojeći projekat `usart` na sljedeći način

File → New → C Project

i odabrati sljedeće postavke

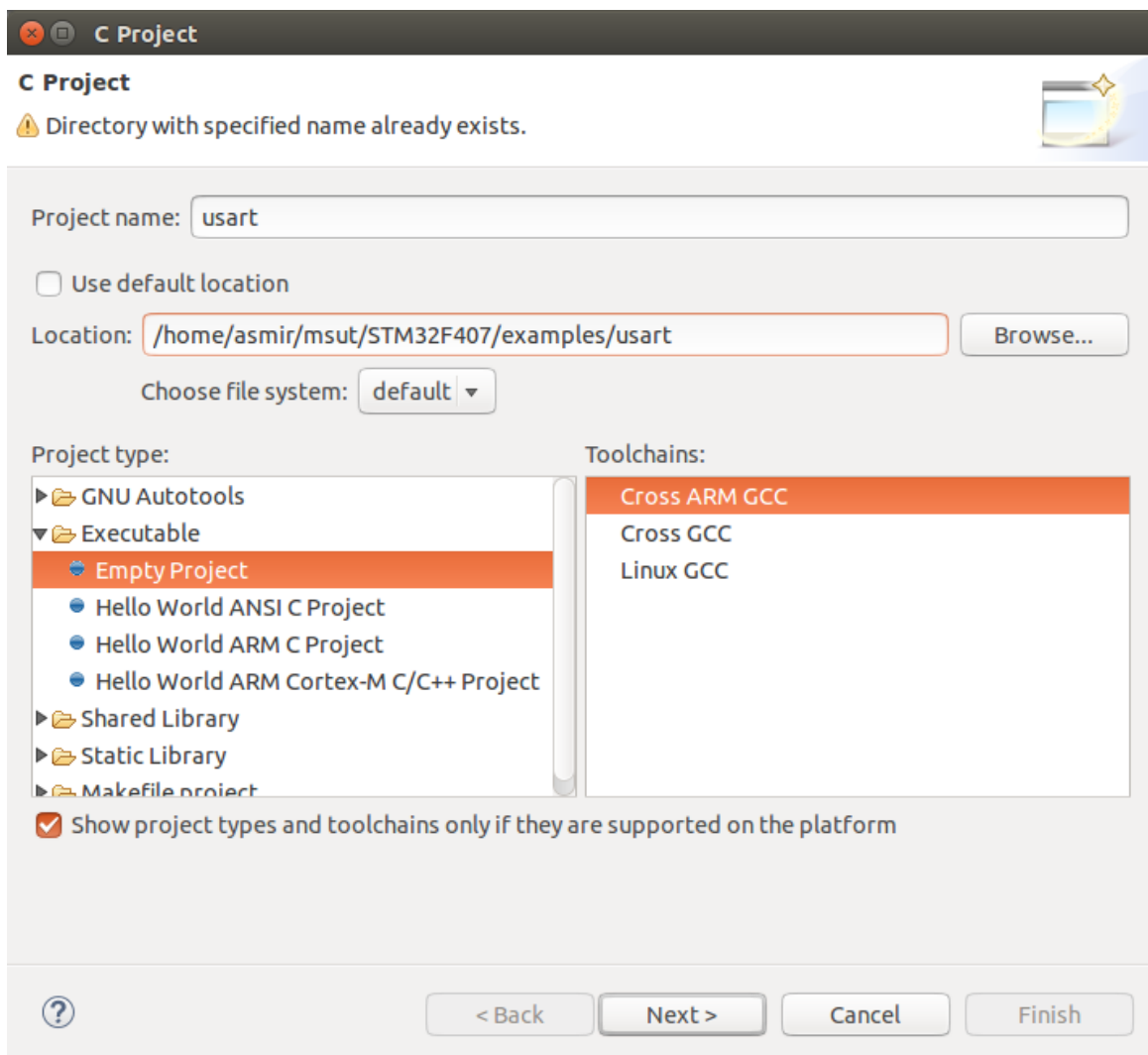


Figure 2: Učitavanje postojećeg projekta u Eclipse - I

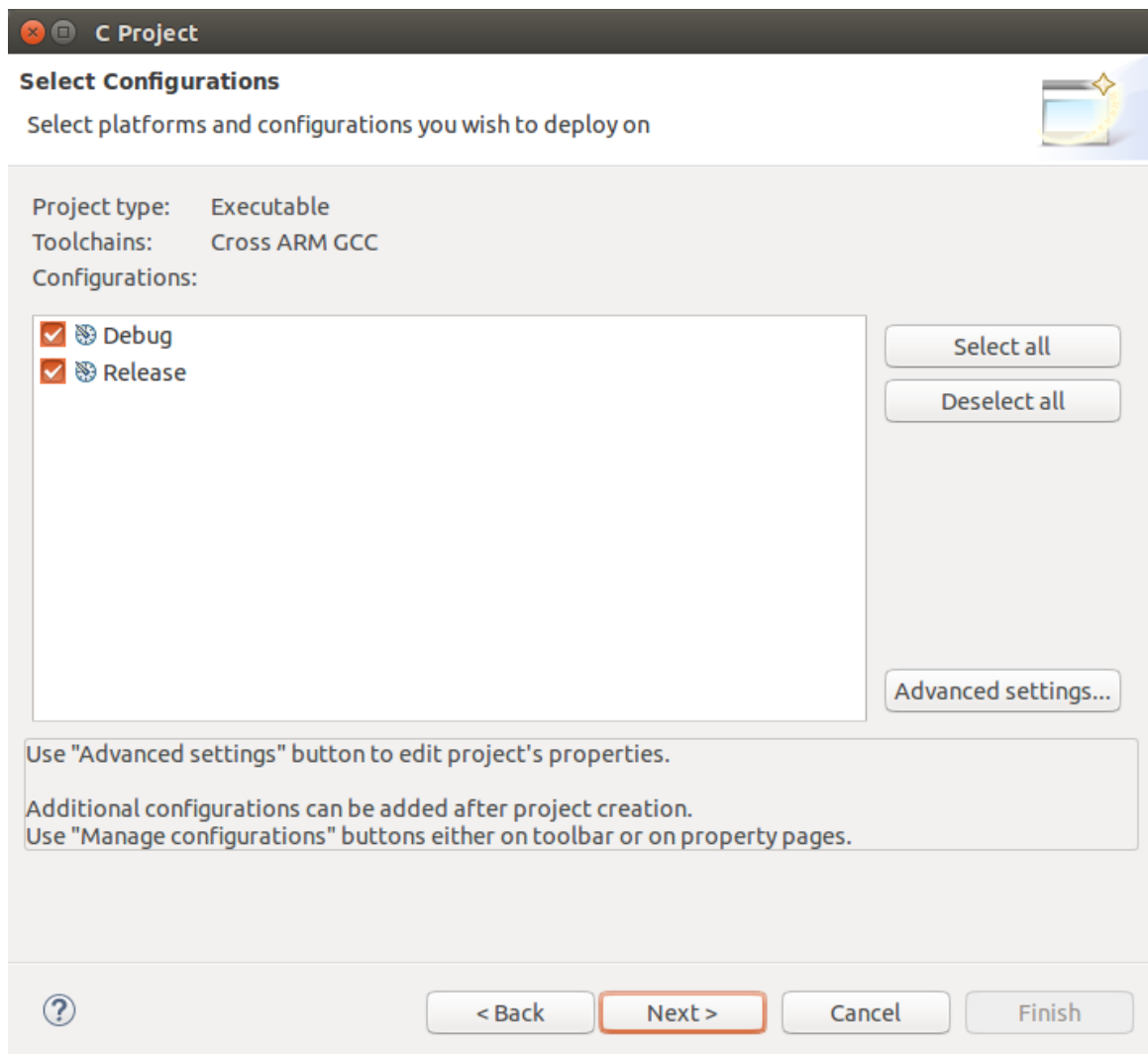


Figure 3: Učitavanje postojećeg projekta u Eclipse - II

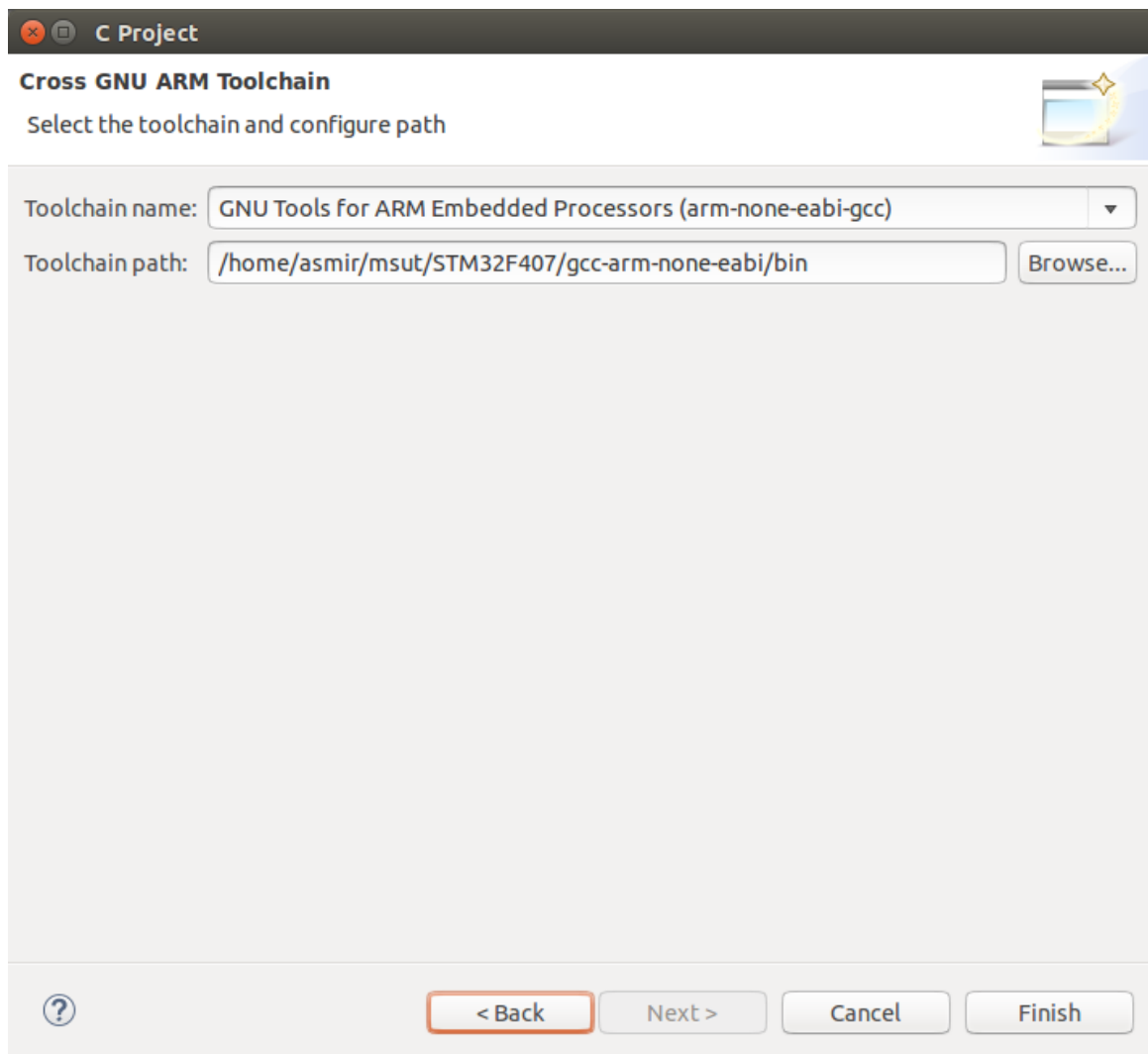


Figure 4: Učitavanje postojećeg projekta u Eclipse - III

8. U okviru **Propertise** trenutnog projekta unijeti sljedeće postavke

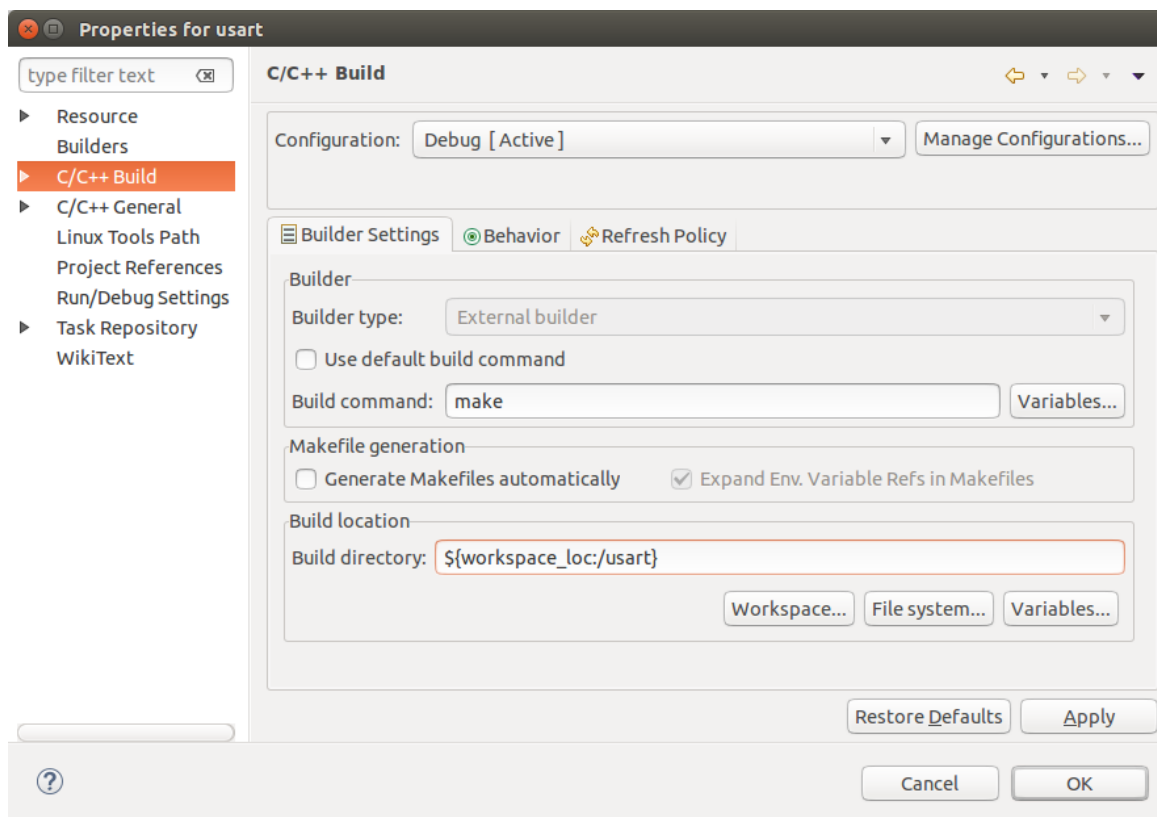


Figure 5: Postavke Eclipse projekta - Propertise

9. Kreirati debug profil kao

Run → Debug profile

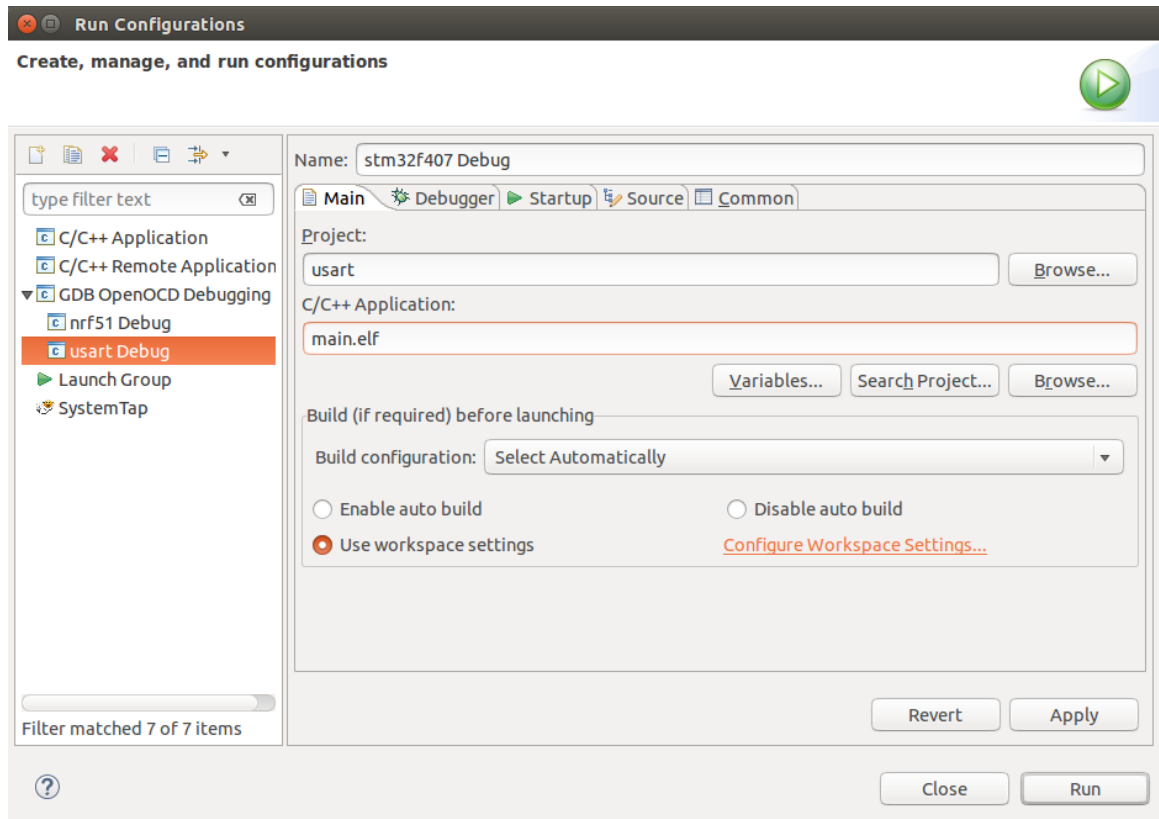


Figure 6: Kreiranje debug profile Eclipse projekta - I

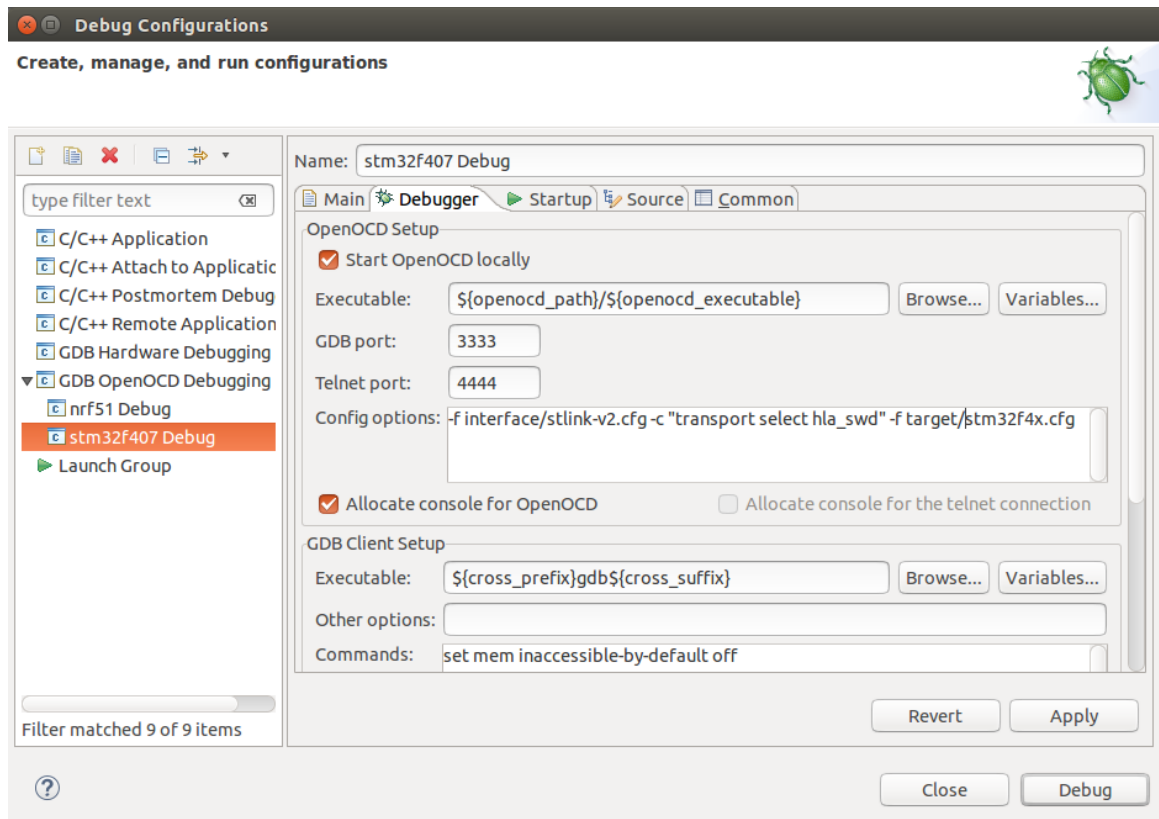


Figure 7: Kreiranje debug profile Eclipse projekta - II

unutar config options neophodno je unijeti sljedeće parametre

```
-f interface/stlink-v2.cfg -c "transport select hla_swd" -f
target/stm32f4x.cfg
```

10. Da bi smo bili u mogućnosti analizirati stanje pojedinih registara neophodno je instalirati embsysregview. U okviru menija odabrati

Help → Install New Software...

a zatim dodati u *Work with* sljedeći tekst

```
http://embsysregview.sourceforge.net/update
```

a na listi koja se pojavi ispod, odabrati embsysregview.

11. embsysregview ćemo aktivirati tokom procesa debug-iranja na sljedeći način

Window → Show View → Other

zatim upisati embsysregview i dodati. U okviru novog tab-a embsysregview desnim klikom aktivirati *propertise* embsysregview i odabrati prema sljedećoj slici

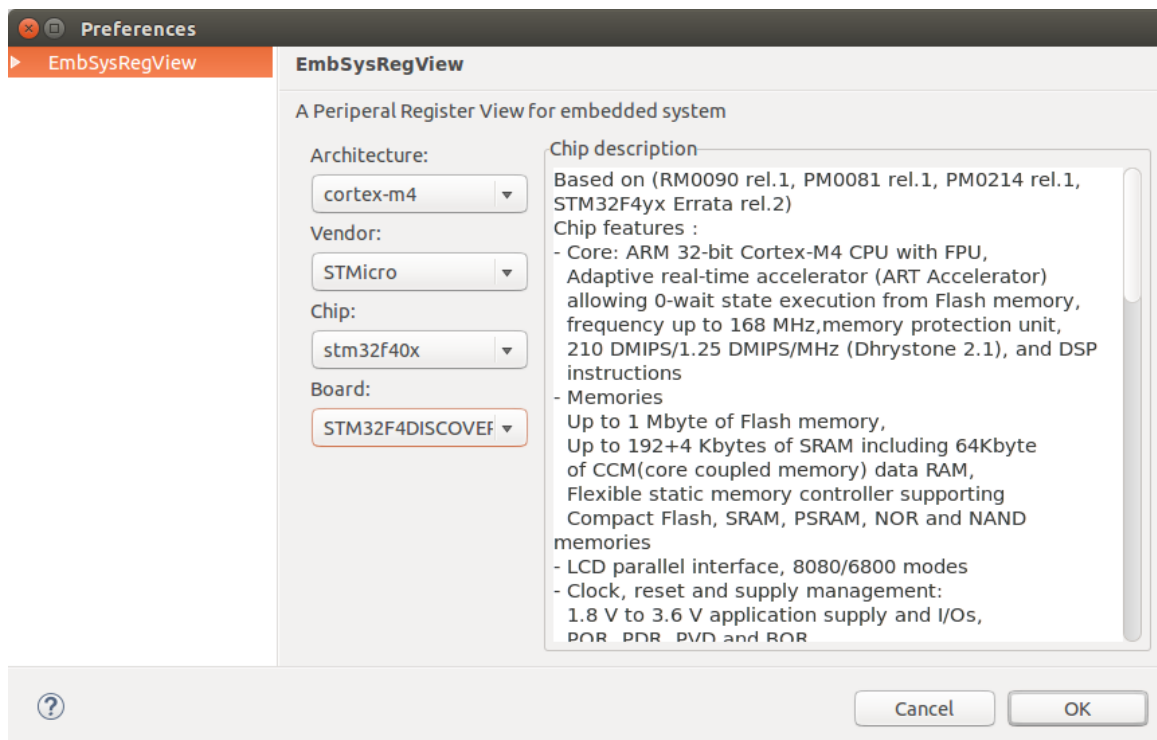


Figure 8: Postavke embsysregview

Sretno Kodiranje!