Uporaba urejevalnika texta VSCode za programiranje sistema STM32F407 Discovery

Andrej Sušnik

October 3, 2022

1 Predpogoji za namestitev

1.1 Operacijski sistem Windows

1.1.1 STM32CubeIDE

Prenesemo in namestimo integrirano razvojno okolje STM32CubeIDE povezava

1.1.2 GCC for ARM embedded

Prenesemo in namestimo GCC povezava. Pot kamor namestimo gcc dodamo v globalno spremenljivko "Path"

1.1.3 Visual Studio Code

Prenesemo in namestimo povezava

1.1.4 xPack QEMU - arm simulator

Prenesemo xPack povezava

Prenešeni arhiv razpakiramo in ga kopiramo v

```
%USERPROFILE%\AppData\Roaming\xPacks\qemu-arm
```

Če smo kopirali pravilno bi moral ukaz

```
%USERPROFILE%\AppData\Roaming\xPacks\qemu-arm\xpack-
qemu-arm-7.1.0-1\bin\qemu-system-gnuarmeclipse.exe --version
```

izpisati nekaj podobnega.

```
xPack QEMU emulator version 7.0.0 (v7.0.0-xpack)
Copyright (c) 2003-2022 Fabrice Bellard and the QEMU Project developers
Drivers
```

1.2 Operacijski sistem Linux

Za uspešno programiranje in razhroščevanje sistem STM32F407 Discovery moramo najprej namestiti nekaj programov.

1.2.1 STM32CubeIDE

Najprej moramo na računalnik namestiti STM32CubeIde. Prenesemo ga na povezavi. Ko ga prenesemo ga še namestimo na distribucijah, ki so zgrajene na debianu to storimo z ukazom

```
sudo sh st-stm32cubeide_1.10.1_12716_20220707_0928_amd64.deb_bundle.sh
```

Za nalaganje programov namestimo še stlink-tools

```
sudo apt install stlink-tools
```

1.2.2 GCC for ARM embedded

Da lahko prevajamo in razhroščujemo programe za vgrajene sisteme moramo namestiti GCC za arm vgrajene sisteme. To storimo z naslednjimi ukazi.

```
cd ~/Downloads

wget https://developer.arm.com/-/media/Files/downloads/gnu-rm/10.3-2021.10/gcc-arm-none-eabi-10.3-202

sudo mv gcc-arm-none-eabi-10.3-2021.10 /usr/share

sudo ln -s /usr/share/gcc-arm-none-eabi-10.3-2021.10/bin/arm-none-eabi-gdb /usr/bin/arm-none-eabi-gdb
sudo ln -s /usr/share/gcc-arm-none-eabi-10.3-2021.10/bin/arm-none-eabi-ld /usr/bin/arm-none-eabi-ld
sudo ln -s /usr/share/gcc-arm-none-eabi-10.3-2021.10/bin/arm-none-eabi-objcopy /usr/bin/arm-none-eabi
sudo ln -s /usr/share/gcc-arm-none-eabi-10.3-2021.10/bin/arm-none-eabi-gcc /usr/bin/arm-none-eabi-gcc
sudo ln -s /usr/share/gcc-arm-none-eabi-10.3-2021.10/bin/arm-none-eabi-g++ /usr/bin/arm-none-eabi-g++
sudo ln -s /usr/share/gcc-arm-none-eabi-10.3-2021.10/bin/arm-none-eabi-objdump /usr/bin/arm-none-eabi
sudo ln -s /usr/share/gcc-arm-none-eabi-10.3-2021.10/bin/arm-none-eabi-size /usr/bin/arm-none-eabi-si
sudo ln -s /usr/share/gcc-arm-none-eabi-10.3-2021.10/bin/arm-none-eabi-nm /usr/bin/arm-none-eabi-nm
```

1.2.3 Visual Studio Code

Program za urejanje texta VSCode si prenesemo z spodnje povezave in ga namestimo VSCode Download.

Ko prenesemo in namestimo VSCode. Ga odpremo in namestimo še 3 razširitve.

1. Arm Assembly - Barvanje kode za arm-ov asembler
2. Cortex-Debug - Razhroščevanje programov
3. Memory Viewer - Pregled spomina

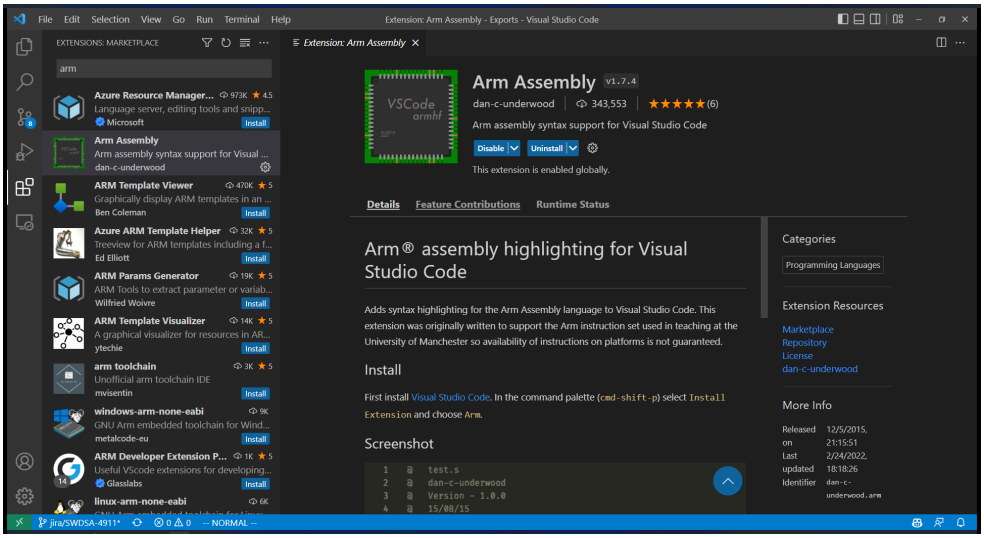

Figure 1: Razširitev Arm Assembly

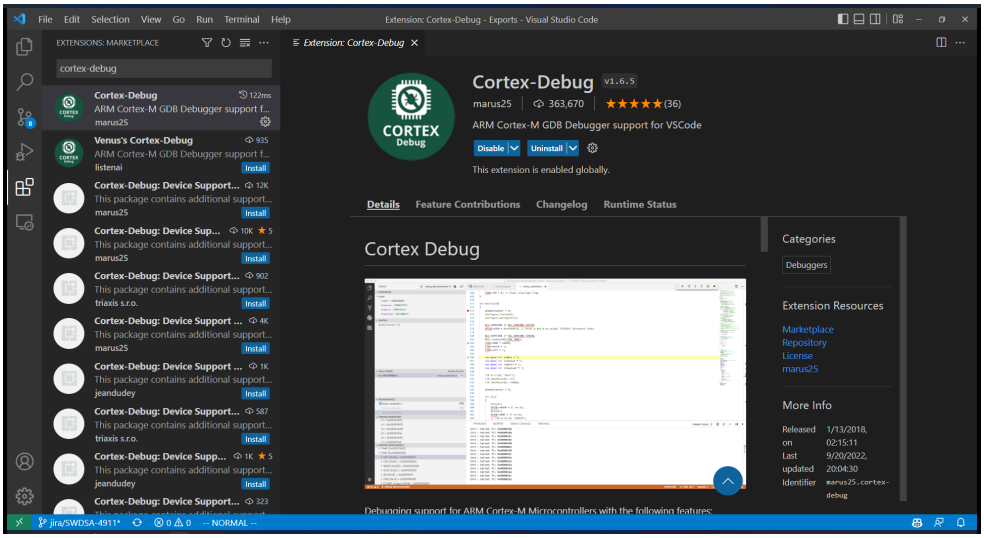

Figure 2: Razširitev Cortex-Debug

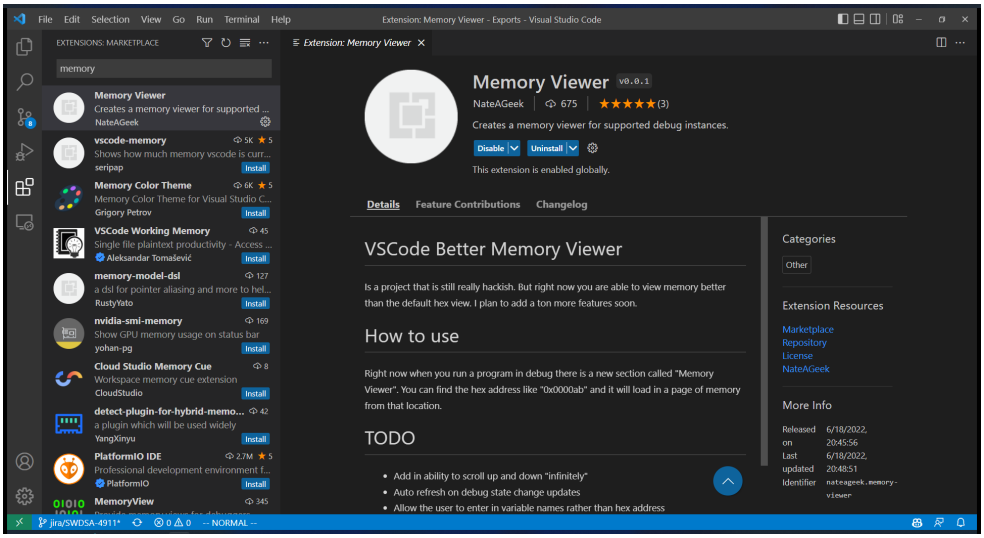

Figure 3: Razširitev Memory Viewer

1.2.4 xPack QEMU - arm simualtor

Prenesemo xPackQEMUArm iz githuba v Downloads direktorij (povezava).

```
mkdir -p ~/.local/xPacks/qemu-arm
cd ~/.local/xPacks/qemu-arm

tar xvf ~/Downloads/xpack-qemu-arm-7.0.0-1-linux-x64.tar.gz
chmod -R -w xpack-qemu-arm-7.0.0-1

ln -s ~/.local/xPacks/qemu-arm/xpack-qemu-arm-7.0.0-1/bin/qemu-system-gnuarmeclipse \
/usr/bin/qemu-system-gnuarmeclipse
```

1.3 Uporaba razvojnega okolja

Z razvojem začnemo tako da prenesemo začetni projekt. To storimo z ukazom

```
git clone https://github.com/AndrejSusnik/STM32AsmTemplate.git
```

V visual studio code projekt odpremo tako da v glavnem meniju kliknemo File in Open folder v podmeniju, odpre se nam okno za izbiro direktorija v katerem izberemo začetni projekt.

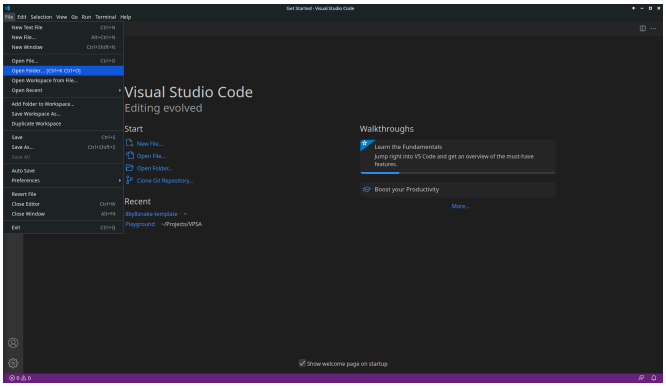

Figure 4: Odpiranje direktorija

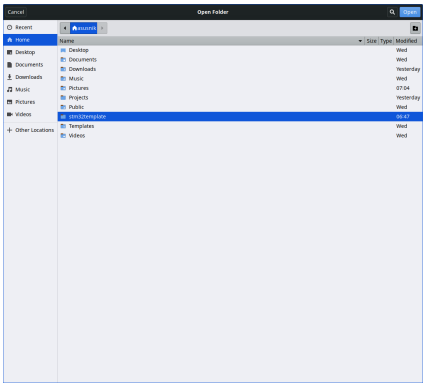

Figure 5: Odpiranje začetnega projekta

Nato odpremo zavihek debug and run

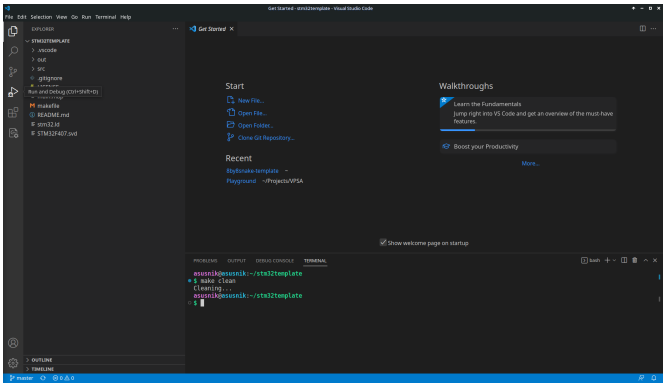

Figure 6: Debug and Run

Kjer v meniju izberemo simulator

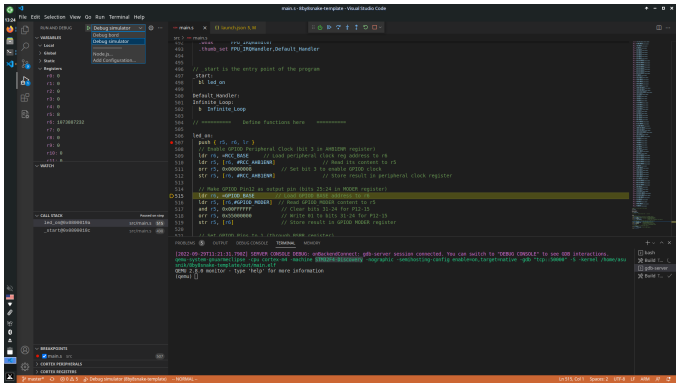

Figure 7: Izbira simulatorja

References

[1] https://code.visualstudio.com/

[2] https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html

[3] https://www.st.com/en/development-tools/stm32cubeide.html

[4] https://developer.arm.com/downloads/-/gnu-rm

[5] https://xpack.github.io/qemu-arm/

[6] https://github.com/zrezke/8by8snake-template