



RISC-V SoC and Firmware Development

Lab: 06



MAKEFILE

A **Makefile** is a file that helps you automate the process of compiling. Instead of manually typing commands every time you want to compile your project, a Makefile lists instructions for make, a tool that runs those commands for you.

Syntax

target: dependencies
command

- The targets can be outputs files, action or tasks.
- The dependencies are the input files or different targets on which the current target is dependant.
- The commands are the actual compiling instructions.



MAKEFILE

EXAMPLE

```
all: brom flash

$(FW_FILE): flash

brom:
    $(MAKE) -C fw/fw-brom

flash:
    $(MAKE) -C fw/fw-flash

clean:
    $(MAKE) -C fw/fw-brom clean
    $(MAKE) -C fw/fw-flash clean
```

- The **make all** command is dependant on **brom** and **flash** targets, first it will execute brom and flash targets.



FLASHING

Clone Repository

- \$ `git clone https://github.com/Abdul-muheet-ghani/picotiny.git`

Step01

- Modify the firmware.
- Then navigate to picotiny folder.

Step02

- Navigate to the picotiny folder and run **\$ make all**

```
fayz@Fayz:~/Documents/picotiny$ make all
```

- It will convert the the firmware.c file into hex and into a verilog file.

FLASHING

Step03

- Run the command **\$make program** this will flash the verilog file onto the FPGA through **pico-programmer.py** and press the reset push button.

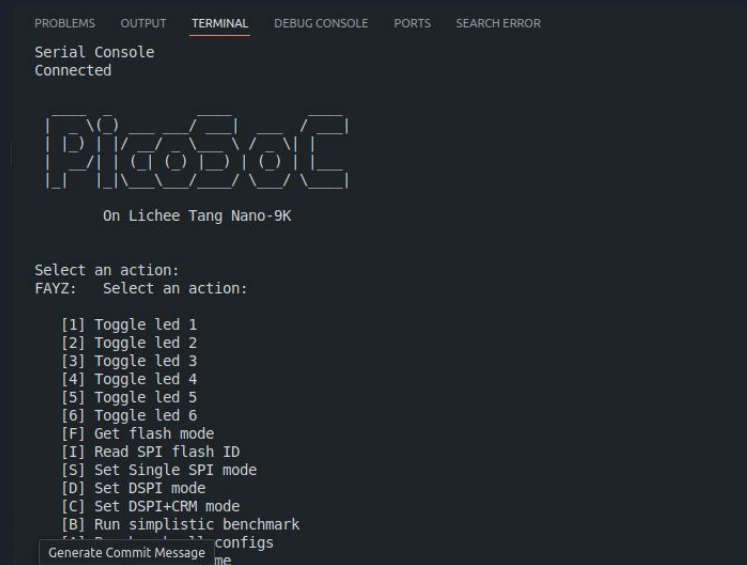
```
fayz@Fayz:~/Documents/picotiny$ make program
make -C fw/fw-flash
make[1]: Entering directory '/home/fayz/.local/share/Trash/files/picotiny.6/fw/fw-flash'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/home/fayz/.local/share/Trash/files/picotiny.6/fw/fw-flash'
python3 sw/pico-programmer.py /home/fayz/Documents/picotiny/fw/fw-flash/build/fw-flash.v /dev/ttyUSB1
Read program with 16945 bytes
- Waiting for reset -
.....
Total sectors 5
Total pages 67
Flashing 1 / 5
Flashing 2 / 5
Flashing 3 / 5
Flashing 4 / 5
Flashing 5 / 5

Flashing completed
```

FLASHING

Step04

- Program the FPGA from vscode.
- Now open the serial terminal from vscode and refresh the push button of FPGA.



The screenshot shows the VS Code interface with the 'TERMINAL' tab selected. The terminal output indicates a 'Serial Console Connected' status. Below this, the 'PICOSOC' logo is displayed in a stylized font, followed by the text 'On Lichee Tang Nano-9K'. A prompt 'Select an action:' is shown, followed by a list of actions: [1] Toggle led 1, [2] Toggle led 2, [3] Toggle led 3, [4] Toggle led 4, [5] Toggle led 5, [6] Toggle led 6, [F] Get flash mode, [I] Read SPI flash ID, [S] Set single SPI mode, [D] Set DSPI mode, [C] Set DSPI+CRM mode, and [B] Run simplistic benchmark. A text input field at the bottom contains the text 'Generate Commit Message configs'.

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS SEARCH ERROR
Serial Console
Connected

PICOSOC

On Lichee Tang Nano-9K

Select an action:
FAYZ: Select an action:

[1] Toggle led 1
[2] Toggle led 2
[3] Toggle led 3
[4] Toggle led 4
[5] Toggle led 5
[6] Toggle led 6
[F] Get flash mode
[I] Read SPI flash ID
[S] Set single SPI mode
[D] Set DSPI mode
[C] Set DSPI+CRM mode
[B] Run simplistic benchmark
Generate Commit Message configs
```



TASK

Customize the **firmware.c** file located in **fw/fw-flash** directory and flash it onto the FPGA.

TESTIMONIAL

Author:

[Abdul Muheet Ghani](#), Research Associate at [MERL-UITU](#).

Under The Supervision Of:

[Dr.Ali Ahmed](#) (Team Lead MERL).

Sponsored By: Edmund from [Symbiotic EDA](#). for sponsoring FPGA.

Thanks: [Lushay Lab](#) (They've provided crucial resources and guidance throughout the project)