

Установка среды разработки Eclipse для ОС Linux и отладка ПО для МК «Амур»

АО «Микрон»
г. Зеленоград
2025

Содержание

1 Установка системных утилит и библиотек	4
1.1 Предустановка необходимых библиотек и утилит	4
1.2 Сборка OpenOCD	5
1.3 Подключение аппаратуры	7
1.4 Конфигурирование аппаратуры	8
1.5 Конфигурирование путей к утилитам	14
2 Загрузка и настройка среды программирования	18
2.1 Загрузка Eclipse	18
2.2 Импорт демо-проекта	20
2.3 Подготовка к загрузке проекта в контроллер (Program)	23
2.4 Отладка проекта (Debug)	26

Введение

В данном документе представлены основные этапы установки среды проектирования Eclipse под ОС Linux, запуск демо-проекта на отладочной плате MIK32 с использованием программатора на основе FT2232.

Полезные ссылки:

https://gitflic.ru/company/mikron-mik32	Официальный репозиторий поддержки MIK32
https://openocd.org/	OpenOCD — пакет открытых утилит для отладки по JTAG
https://www.eclipse.org/downloads/	IDE Eclipse — загрузить среду разработки
https://xpack.github.io/	Набор утилит xpack — компиляторы и стандартные библиотеки

1 Установка системных утилит и библиотек

В данном разделе рассматривается установка основных утилит

1.1 Предустановка необходимых библиотек и утилит

1. Обновляем пакеты, при необходимости — перезагрузить

```
sudo apt-get update  
sudo apt-get upgrade
```

2. Установка Python

```
sudo apt-get install python3 python3-pip
```

3. Устанавливаем вспомогательные утилиты — для веб-загрузки и Git, а также для сборки пакета и консольный файловый менеджер-редактор

```
sudo apt-get install curl git  
sudo apt-get install checkinstall  
sudo apt-get install mc
```

4. Установка вспомогательных библиотек, необходимых для xpm

```
sudo apt-get install libncurses5-dev
```

5. Установка xpm (см. сайт <https://xpack.github.io/docs/getting-started/prerequisites/>)

```
wget -q -O - https://raw.githubusercontent.com/xpack/assets/master/scripts/install-nvm-node-npm-xpm.sh | ${SHELL}  
export NVM_DIR="$HOME/.nvm"  
source "$NVM_DIR/nvm.sh"
```

```
-> Appending bash_completion source string to /home/mytabpc2/.bashrc  
=> Close and reopen your terminal to start using nvm or run the following to use it now:  
  
export NVM_DIR="$HOME/.nvm"  
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm  
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion  
Downloading and installing node v22.21.0...  
Downloading https://nodejs.org/dist/v22.21.0/node-v22.21.0-linux-x64.tar.xz...  
##### 60.4%
```

```
npm install --location=global xpm@latest
```

```
mylabpc2@mylabpc2-System-Product-Name:~$ npm install --location=global xpm@latest
changed 25 packages in 3s
54 packages are looking for funding
run `npm fund` for details
```

6. Установка Xpack для RiscV

```
xpm install @xpack-dev-tools/riscv-none-elf-gcc@latest --verbose --global
```

```
mylabpc2@mylabpc2-System-Product-Name:~$ xpm install @xpack-dev-tools/riscv-none-elf-gcc@latest --verbose --global
xPack project manager - install package(s)
Processing @xpack-dev-tools/riscv-none-elf-gcc@15.2.0-1.1...
Installing globally in '/home/mylabpc2/.local/xPacks/@xpack-dev-tools/riscv-none-elf-gcc/15.2.0-1.1'...
Downloading https://github.com/xpack-dev-tools/riscv-none-elf-gcc-xpack/releases/download/v15.2.0-1/xpack-riscv-none-elf-gcc-15.2.0-1-linux-x64.tar.gz...
Extracting 'xpack-riscv-none-elf-gcc-15.2.0-1-linux-x64.tar.gz'...
6757 files extracted in '15.2.0-1.1/.content'
Changing permissions to read-only...
```

Обратить внимание на **номер версии**, в данном случае 15.2.0-1.1, которую желательно скопировать для дальнейшего использования в указании пути к утилитам, см. п. 34

7. Установка вспомогательных библиотек для GDB

```
sudo apt-get install gdb-multiarch
```

8. Установка JRE для Eclipse

```
sudo apt-get install default-jre
```

1.2 Сборка OpenOCD

Производится сборка OpenOCD <https://openocd.org/>

9. Установка базовых пакетов

```
sudo apt-get install make
sudo apt-get install libtool
sudo apt-get install pkg-config
sudo apt-get install autoconf
sudo apt-get install automake
sudo apt-get install texinfo
sudo apt-get install libusb-1.0-0-dev
sudo apt-get install libhidapi-dev
sudo apt-get install libftdi-dev
sudo apt-get install libjaylink0
sudo apt-get install jimsh libjim-dev
sudo apt-get install libjaylink-dev
```

Для навигации по файловой системе устанавливается Midnight Commander. В качестве альтернативы можно использовать FAR Manager for Linux .

10. Загрузка OpenOCD

```
git clone https://github.com/openocd-org/openocd.git
cd openocd
```

11. Загрузка подмодулей

```
git submodule init
git submodule update
```

12. Настройка — выполнить команду

```
./bootstrap
```

13. Конфигурирование

```
./configure --enable-ftdi --enable-jlink --enable-ulink --enable-stlink --enable-usb-blaster --enable-dummy --enable-cmsis-dap
```



```
OpenOCD configuration summary
-----
MPSSE mode of FTDI based devices          yes
CH347 based devices                       yes (auto)
ST-Link Programmer                        yes
TI ICDI JTAG Programmer                   yes (auto)
Keil ULINK JTAG Programmer                 yes
ANGIE Adapter                             yes (auto)
Altera USB-Blaster II Compatible          yes (auto)
Bitbang mode of FT232R based devices      yes (auto)
Versaloon-Link JTAG Programmer            yes (auto)
TI XDS110 Debug Probe                     yes (auto)
```

После конфигурирования не должно быть ошибок не установленных пакетов. Далее необходимо выполнить компиляцию.

14. Компиляция

```
make -j 8
```

Следующие шаги являются опциональными, показано создание пакет для удобства работы.

15. Создание пакета для установки

```
sudo checkinstall -D
```

```
The package documentation directory ./doc-pak does not exist.  
Should I create a default set of package docs? [y]: █
```

После чего будет создан deb-пакет в корневой папке openocd, который можно установить командой

16. Установка пакета

```
sudo dpkg -i __имя пакета__.deb
```

Если не удаётся создать пакет, при выполнении команды возникает ошибка, можно прибегнуть к установке с использованием make.

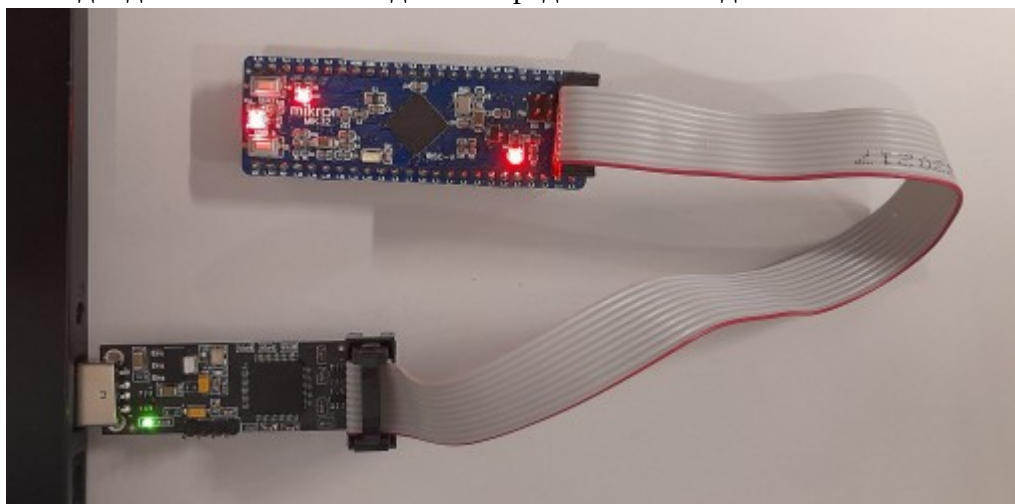
17. Установка с использованием make

```
sudo make install
```

```
done  
make[4]: выход из каталога «/home/mylabpc2/openocd»  
make[3]: выход из каталога «/home/mylabpc2/openocd»  
make[2]: выход из каталога «/home/mylabpc2/openocd»  
make[1]: выход из каталога «/home/mylabpc2/openocd»  
mylabpc2@mylabpc2-System-Product-Name: ~/openocd$ █
```

1.3 Подключение аппаратуры

Можно подключить отладочный модуль в USB. После подключения убедиться, что горят светодиоды питания на отладочном средстве и на подключённой плате.



После подключения отладочного средства необходимо проверить его наличие.

18. Проверка наличия отладочного средства

```
lsusb
Bus 003 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 004 Device 003: ID 0403:6010 Future Technology Devices International, Ltd FT2232C/D/H Dual UART/FIFO IC
mylabpc2@mylabpc2-System-Product-Name:~$
```

Отладочный комплект имеет надпись FT2232C, если она отсутствует — имеется проблема с аппаратурой, необходимо проверить целостность разъёмов, надёжность подключения, попытаться подключить к другому разъёму USB.

1.4 Конфигурирование аппаратуры

Далее используется Midnight Commander для навигации по файловой системе и редактированию необходимых файлов, производится ознакомление с данной утилитой.

19. Запуск консольного файлового менеджера Midnight Commander

```
sudo mc
```

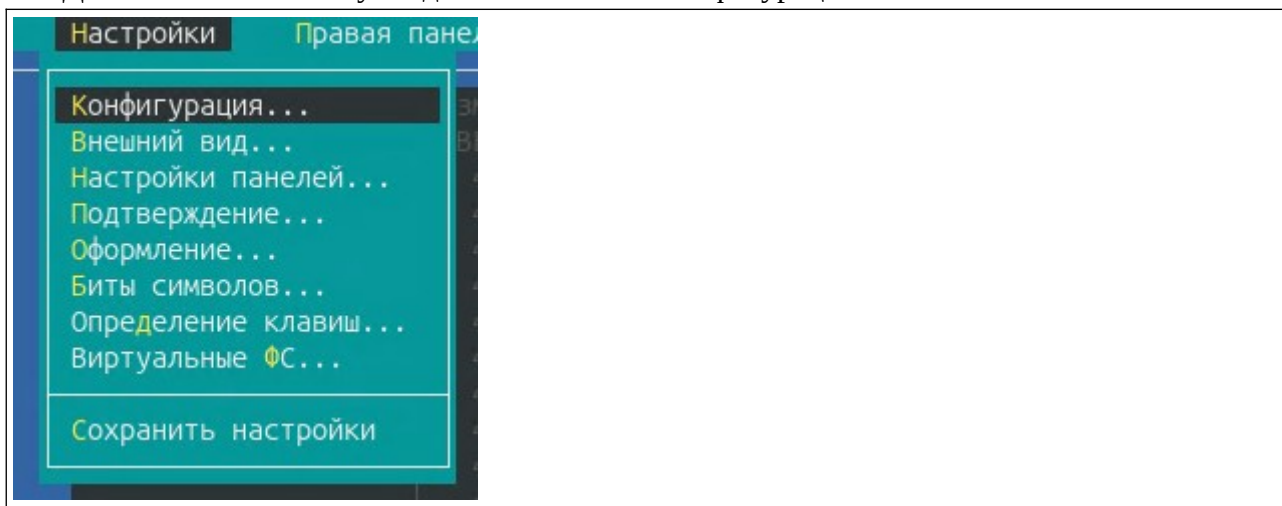
ВНИМАНИЕ!!! Файловый менеджер запущен из-под суперпользователя, любые действия с файлами должны производиться предельно аккуратно во избежание перманентного удаления системы

Справочно.

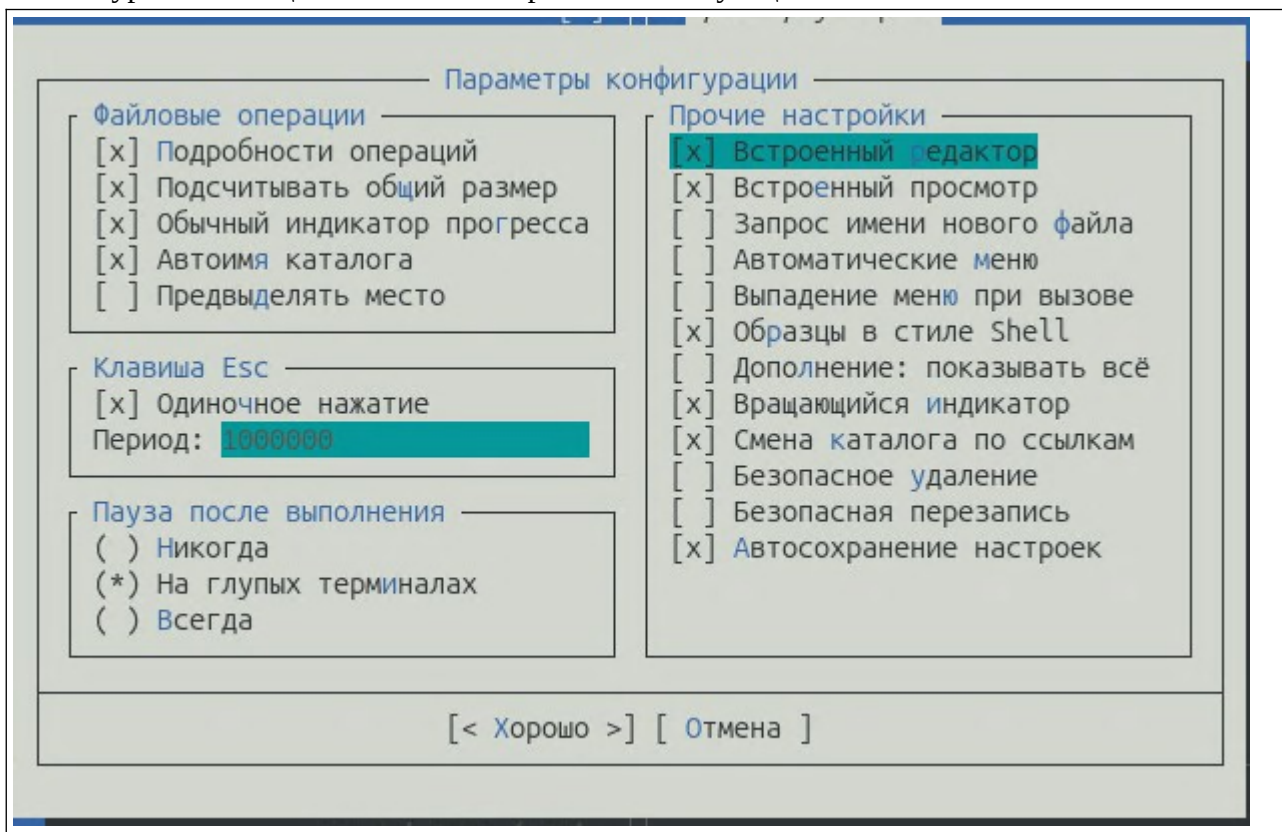
Команда **F8** — удалить файл, **F7** — переместить, **F6** — переименовать) файлы и папки равнозначны.

Суперпользователь обозначен приглашением «#» в командной строке. Обычный пользователь имеет приглашение «\$».

20. Далее нажать клавишу F9 для вызова меню конфигурации



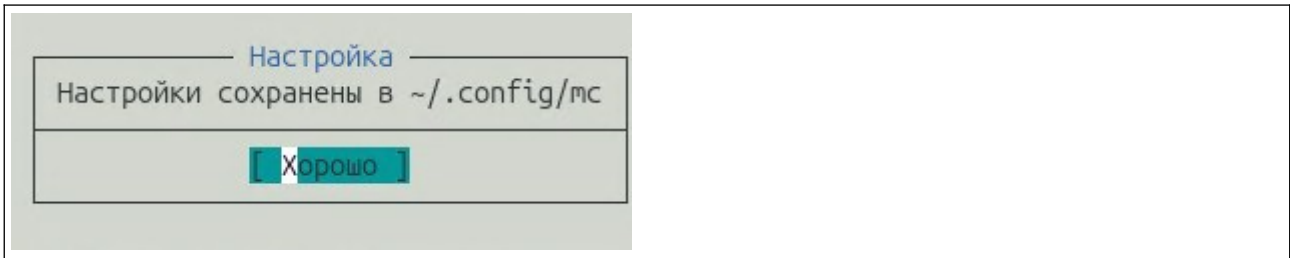
21. Произвести назначение встроенного редактора, поставив крестик мышью или пробелом клавиатуры. Навигация по меню — стрелки или табуляция.



22. Вновь нажать F9, вызвать меню сохранения настроек



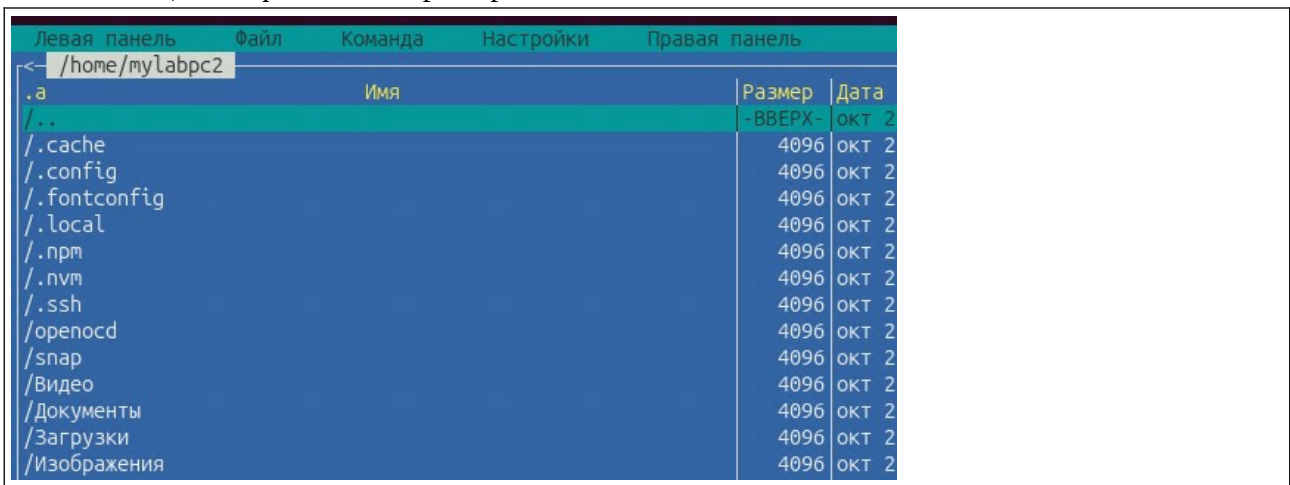
23. Нажать Ввод, настройки успешно сохранены, после чего снова нажать Ввод (Хорошо)



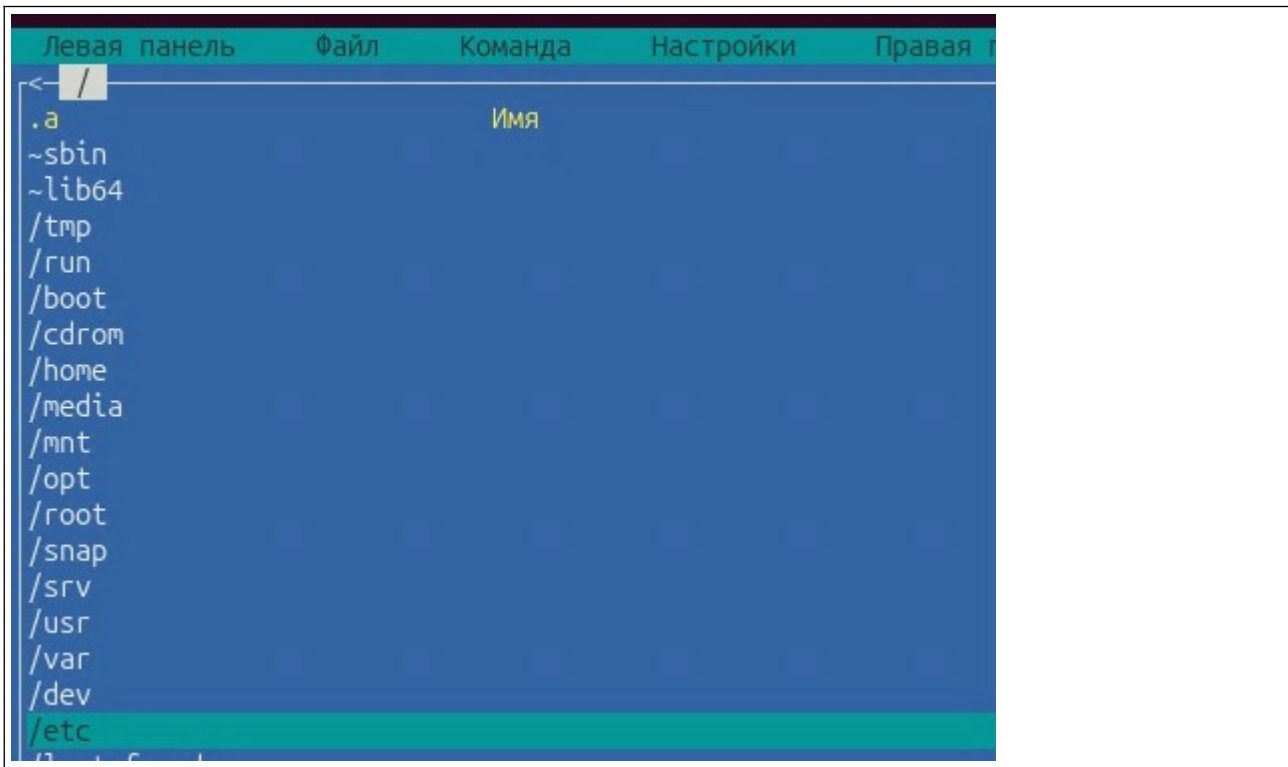
Клавишей табуляции можно переключаться между панелями. Выбрать для удобства любую.

Курсором или нажатием клавиши *Home* выбрать две точки «..» - навигация на папку выше, нажать ввод. Навигация по панели — клавиши вверх-вниз. Клавиша *End* — переход в самый низ списка файлов.

24. Навигация по файлам — пример

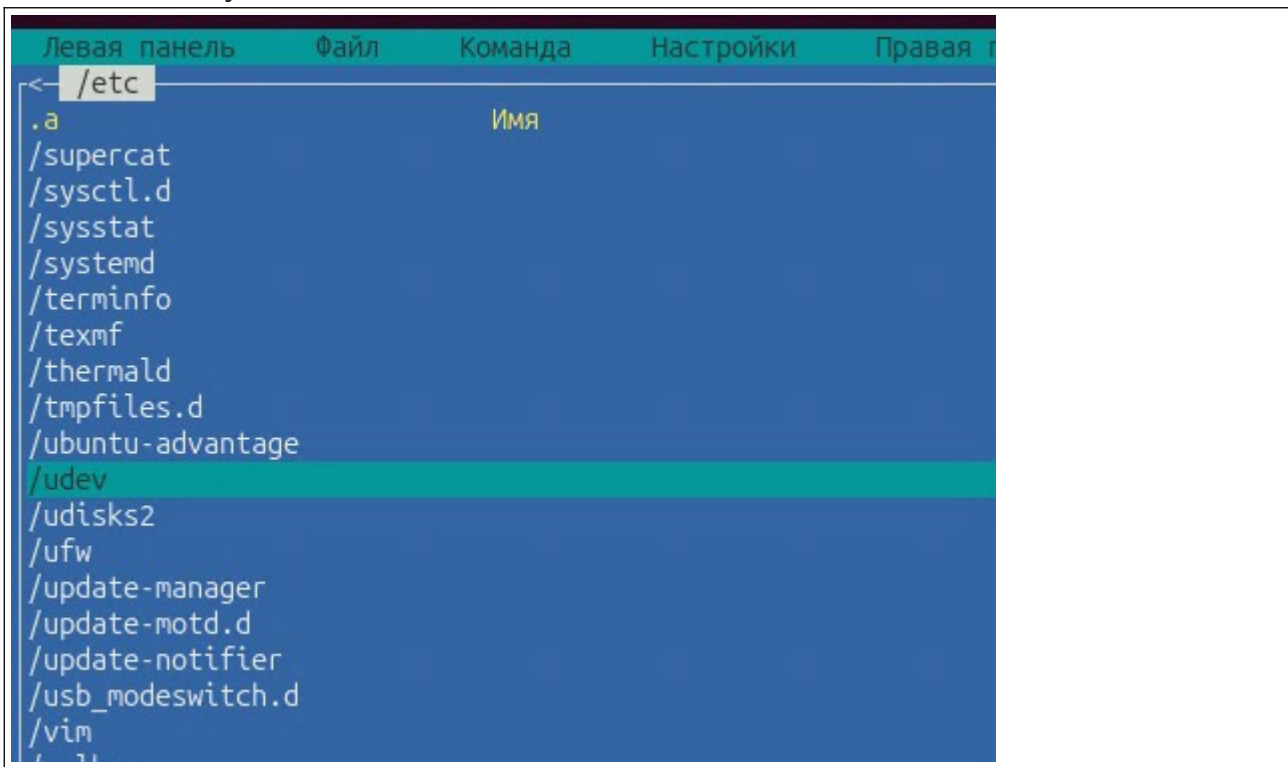


25. Выйти в корневой каталог / и зайти в папку **etc**

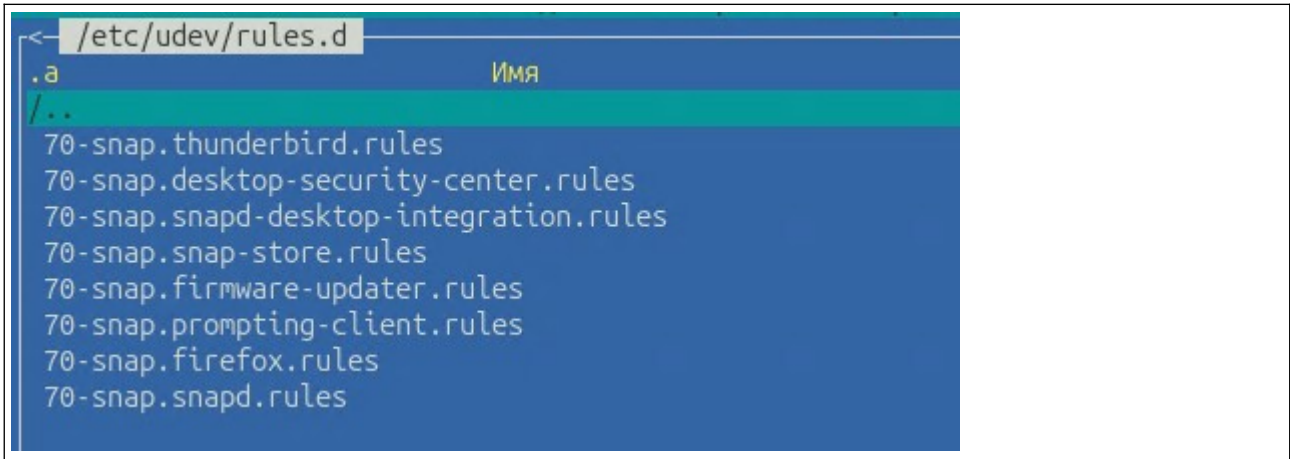


Для упрощения навигации использовать клавиши *Page Up* или *Page Down* — ускорение прокрутки при большом списке

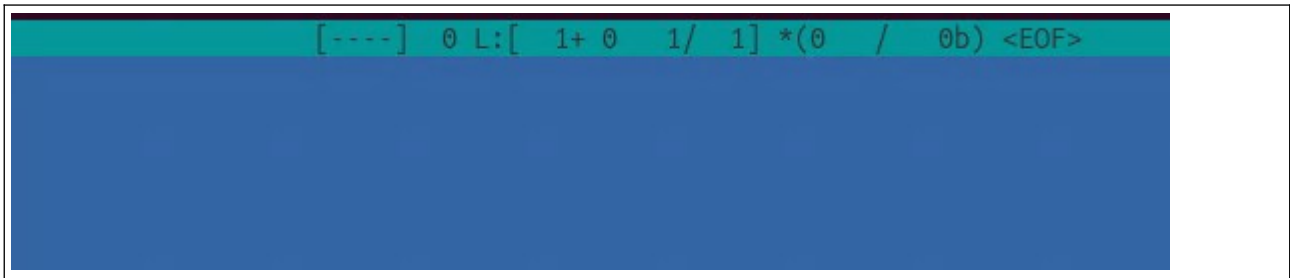
26. Найти папку **udev** и войти в неё



27. Далее проследовать до папки **rules.d** и войти в неё

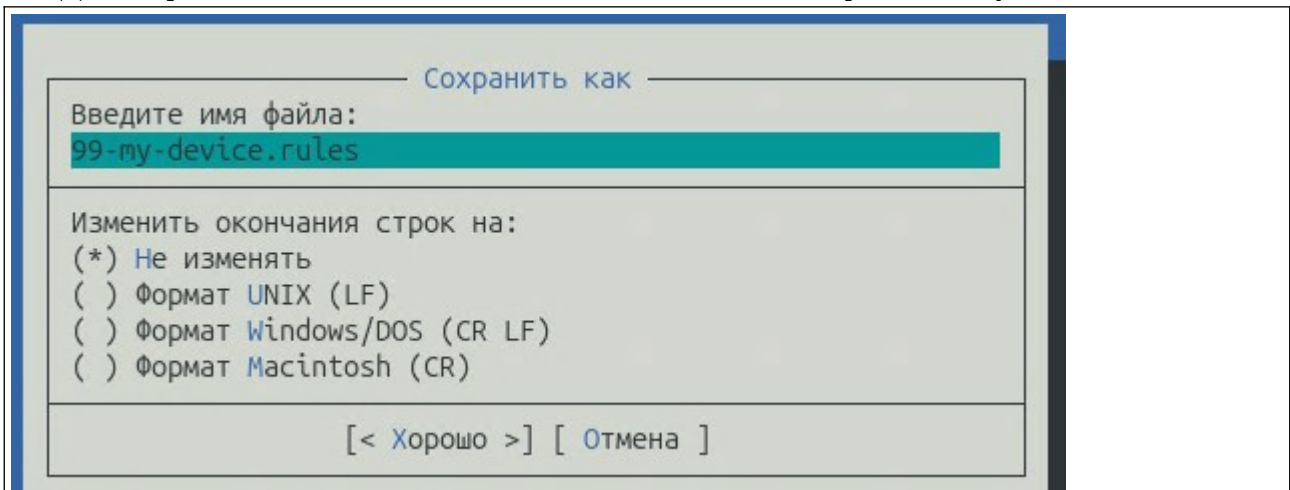


28. Создать новый файл конфигурации правил для аппаратуры необходимо комбинацией клавиш **Shift+F4**:

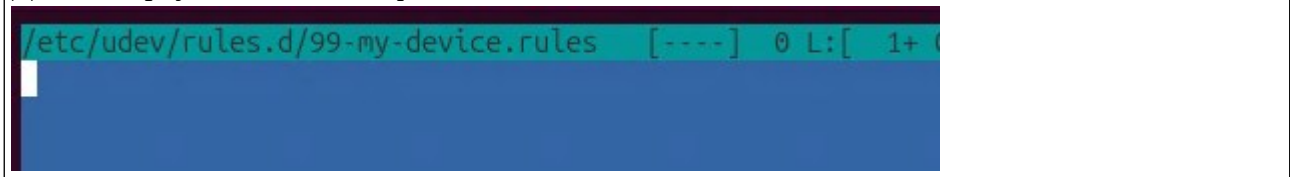


Появится пустой файл.

29. Для сохранения и задания имени нажать **F2**. Ввести имя файла `99-my-device.rules`



Далее вверху появится имя файла



Справочно:

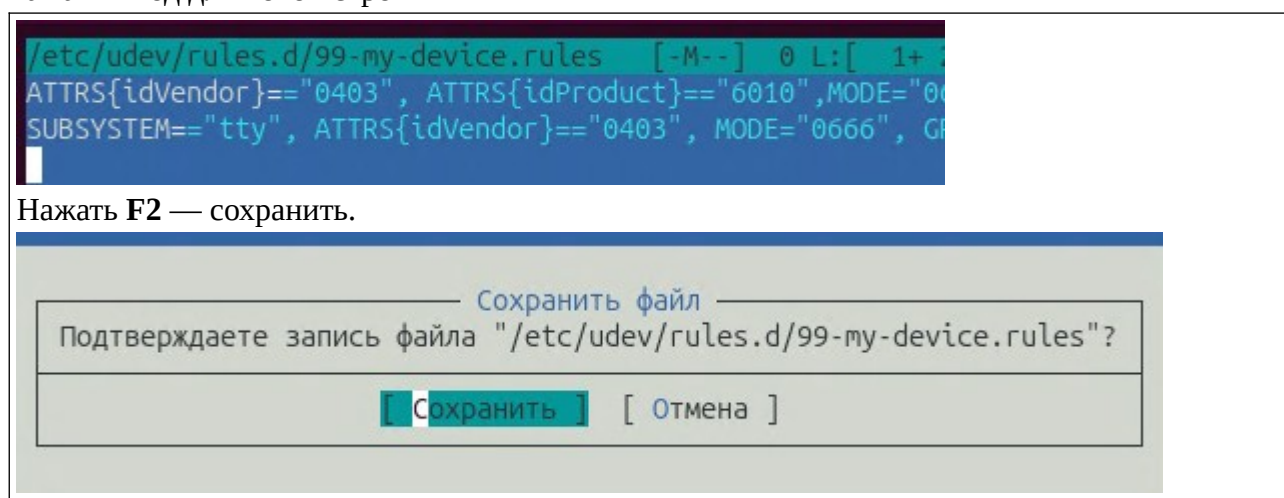
Для вставки текста в консоль, в частности, имени файла и его содержимого использовать **Ctrl+Shift+V**. Для копирования в консоли **Ctrl+Shift+C**.

Для выделения мышью с экрана консоли — использовать зажатый **Shift** и левую клавишу мыши. Не отпуская **Shift** щёлкнуть правой — появится меню копирования содержания экрана терминала.

30. Скопировать следующие строки

```
ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6010",MODE="0666"  
SUBSYSTEM=="tty", ATTRS{idVendor}=="0403", MODE="0666", GROUP="dialout"
```

31. Вставить в терминал и редактор Midnight Commander **Ctrl+Shift+V**, в конце желательно нажать Ввод для новой строки



Нажать **F2** — сохранить.

Нажать два раза **Esc** для выхода из редактора.

Справочно.

Если необходимо вновь отредактировать файл, нажать на нём **F4**, если необходимо просмотреть, нажать **F3**.

Файл создан и имеет не нулевую длину (134 байт)

Имя	Размер	Дата правки
./	-ВВЕРХ-	окт 7 05:33
99-my-device.rules	134	окт 23 19:38
70-snap-thunderbird.rules	1638	окт 7 05:40

Справочно. Для просмотра консоли и возвращения обратно в режим файлового менеджера использовать комбинацию **Ctrl+O**.

Выйти из Midnight Commander введя команду `exit` в командной строке.

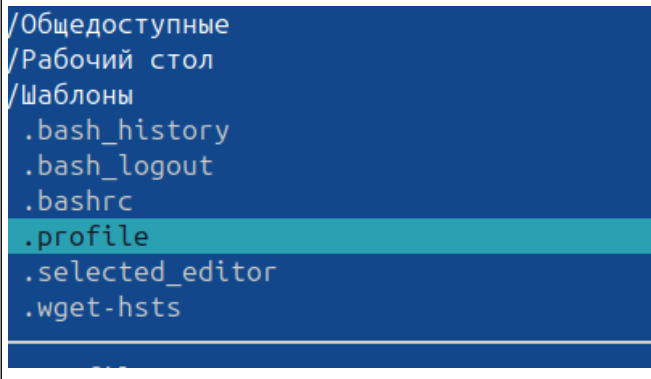
32. Перезапустить правила командой

```
sudo udevadm control --reload-rules
sudo udevadm trigger
```

1.5 Конфигурирование путей к утилитам

Запустить Midnight Commander командой `mc` для обычного пользователя. В приглашении командной строки вместо «#» будет «\$».


33. Прописать путь к созданному xPack. Для этого Midnight Commander-ом выбрать и открыть файл `.profile` в корневом каталоге «~» клавишей **F4**



```
/Общедоступные
/Рабочий стол
/Шаблоны
.bash_history
.bash_logout
.bashrc
.profile
.selected_editor
.wget-hsts
```

34. Появится редактор, вставить в конец строку

```
export PATH=/home/mylabpc2/.local/xPacks/@xpack-dev-tools/riscv-none-elf-gcc/Номер
версии из п. 6/.content/bin:$PATH
```

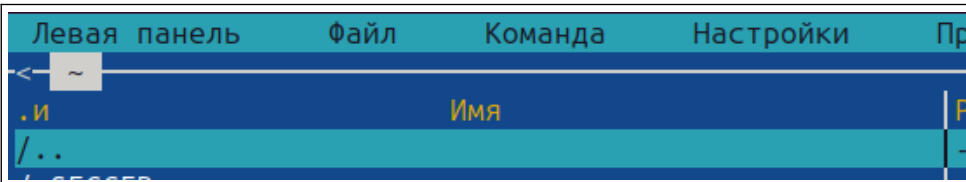


```
PATH=$HOME/bin:$PATH
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
fi
export PATH=~/.local/xPacks/@xpack-dev-tools/riscv-none-elf-gcc/14.2.0-3.1/.content/bin/:$PATH
```

Внимание! Подставить свой номер версии! После чего нажать **F2** для сохранения файла и дважды нажать **Esc** для выхода в основное меню. Далее желательно выйти и зайти снова в систему чтобы обновился путь (перезагружать не обязательно).

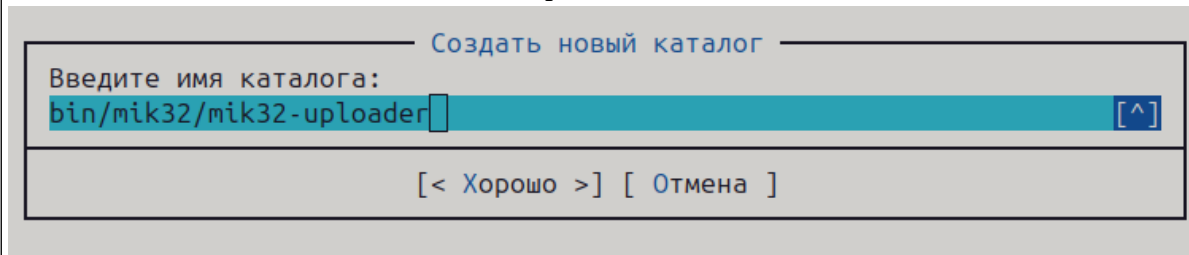
35. Далее необходимо создать папку `~/bin/mik32` в домашнем каталоге, в которой будут размещаться все необходимые утилиты для `mik32`. Проект содержит ссылку на эту папку. Для этого необходимо перейти в домашнюю папку до значка «тильда»



```
Левая панель  Файл  Команда  Настройки  Пр
< ~
.и  Имя  Р
/..  -
```

36. Создать папку и одновременно подпапку клавишей F7

Использовать в поле bin/mik32/mik32-uploader



После чего папка появится в корневом каталоге



внутри неё будет mik32 и ещё внутри mik32-uploader

Справочно. Чтобы в Midnight-Commander-е папки были одинаковые, необходимо нажать **Alt+i**.

37. Зайти внутрь папки mik32 и скачать с использованием утилиты curl или браузера файл загрузчика в EEPROM

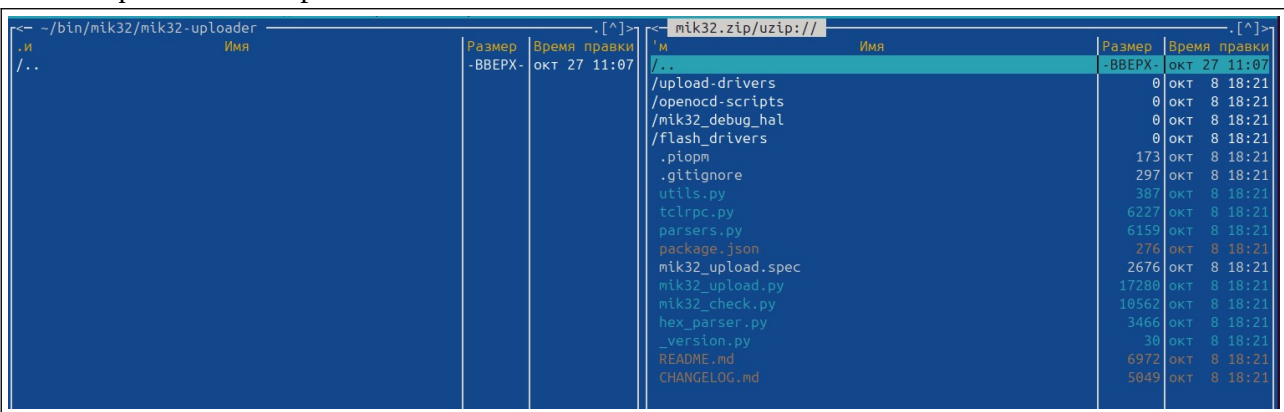
```
curl "https://gitflic.ru/project/mikron-mik32/mik32-uploader/file/downloadAll?branch=master&format=zip" --output mik32.zip
```



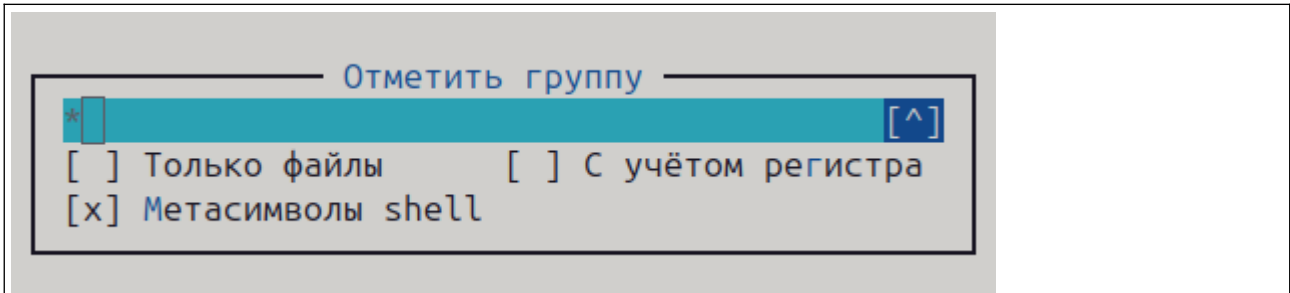
или сделать с применением git по адресу

<https://gitflic.ru/project/mikron-mik32/mik32-uploader.git>, целиком скачав готовую папку

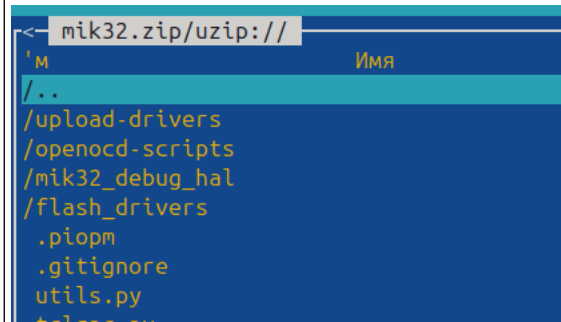
38. В соседней панели (переключение табуляция) выбрать mik32-uploader а в текущей можно войти в файл mik32.zip



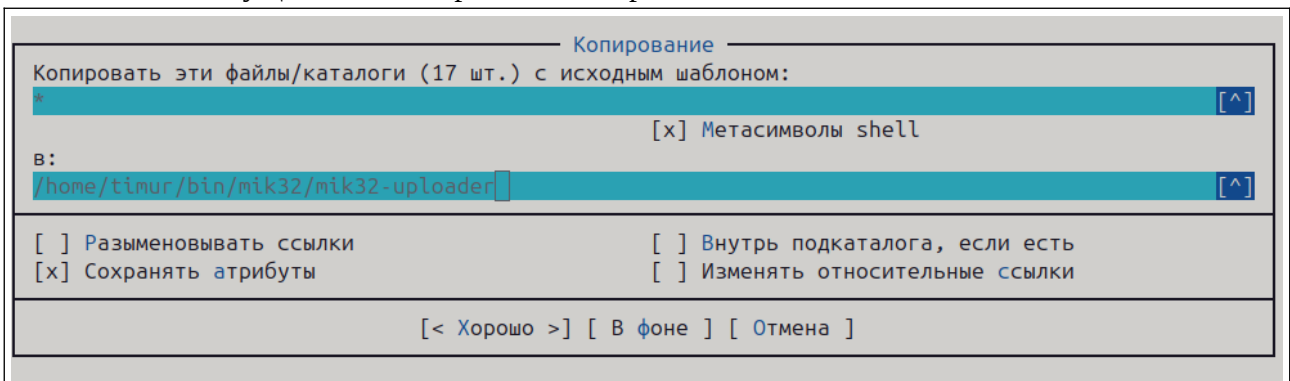
39. Чтобы выбрать все файлы необходимо нажать «+»



Список файлов изменит цвет, который показывает выделение группы:

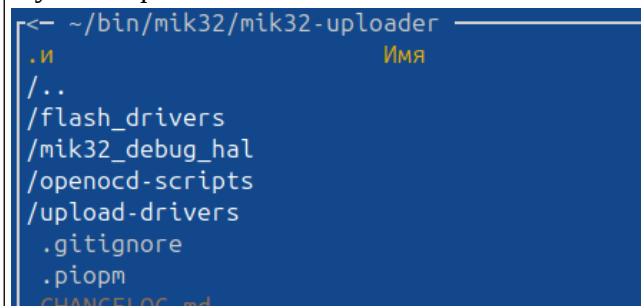


40. Распаковка осуществляется простым копированием клавишей F5

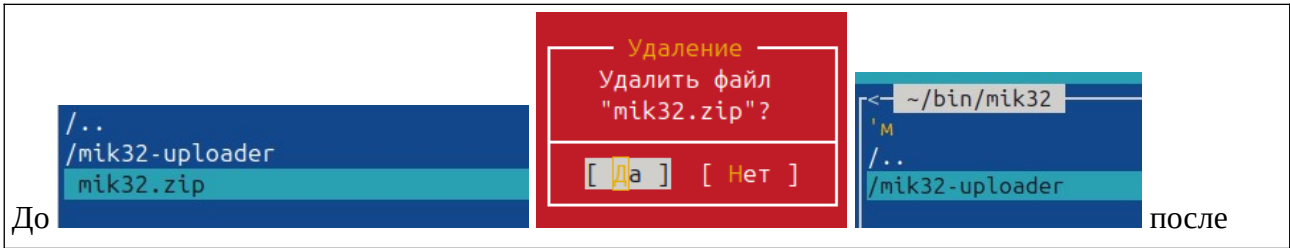


Далее нажать Enter и дождаться распаковки (когда много файлов распаковка МС идёт медленно)

Нужные файлы появятся в каталоге

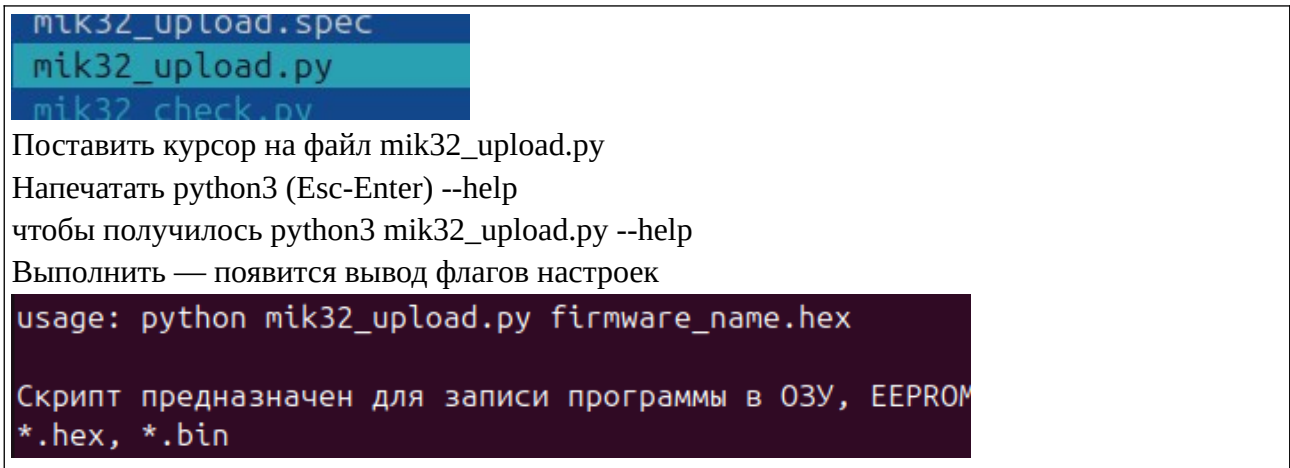


Опционально можно удалить архив поставив на него курсор и нажав F8

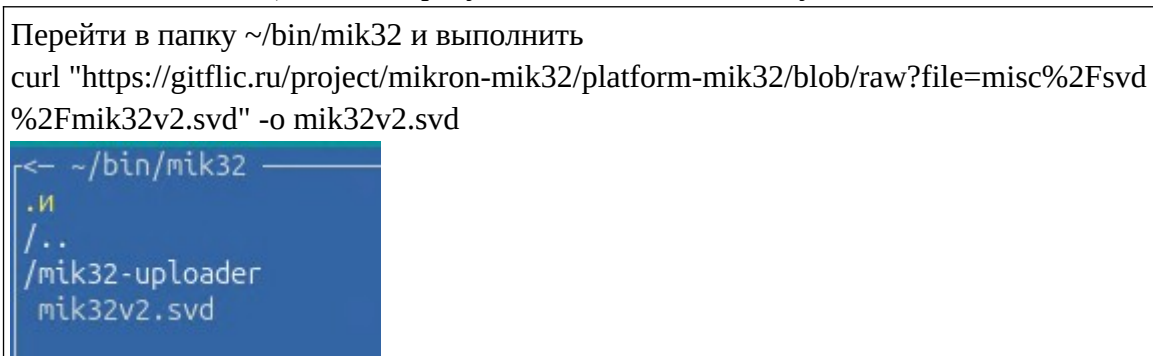


Для проверки основной работоспособности можно выполнить скрипт с использованием интерпретатора Python.

41. Тестовый запуск скрипта — получение подсказки



42. Загрузить и скопировать в папку ~/bin/mik32 файл описания периферийных регистров mik32v2.svd находящийся по адресу или с использованием утилиты curl



2 Загрузка и настройка среды программирования

2.1 Загрузка Eclipse

43. Перейти на сайт, загрузить среду для необходимой ОС и архитектуры компьютера

<https://www.eclipse.org/downloads/>

Выбрать архитектуру целевого компьютера:



В папку «Загрузки» или указанную иную другую будет загружен файл

```
eclipse-inst-jre-linux64.tar.gz 115782K
```

Справочно: чтобы скопировать имя файла в командную строку Midnight Commander (MC) необходимо подвести курсор под имя файла и произвести Esc-последовательность: сначала нажать и отпустить **Esc**, затем нажать **Enter** (не одновременно). Esc-последовательности также являются управляющими командами. Например, чтобы найти файл можно воспользоваться **Esc-?** (нажать **Esc** отпустить, далее быстро **Shift c ?** одновременно)

Далее его можно распаковать с использованием tar: (достаточно набрать tar -zxf и нажать Esc-последовательность в MC), затем нажать Enter и запустить распаковку

44. Распаковка утилитой tar

```
tar -zxf eclipse-inst-jre-linux64.tar.gz
~/Загрузки$ tar -zxf eclipse-inst-jre-linux64.tar.gz
3Просмотр 4Правка 5Копия 6Переименовать
```

Для упрощения задачи:

написать tar -zxf (пробел) далее нажать Esc (отпустить) и быстро нажать Enter, имя файла копируется в консоль

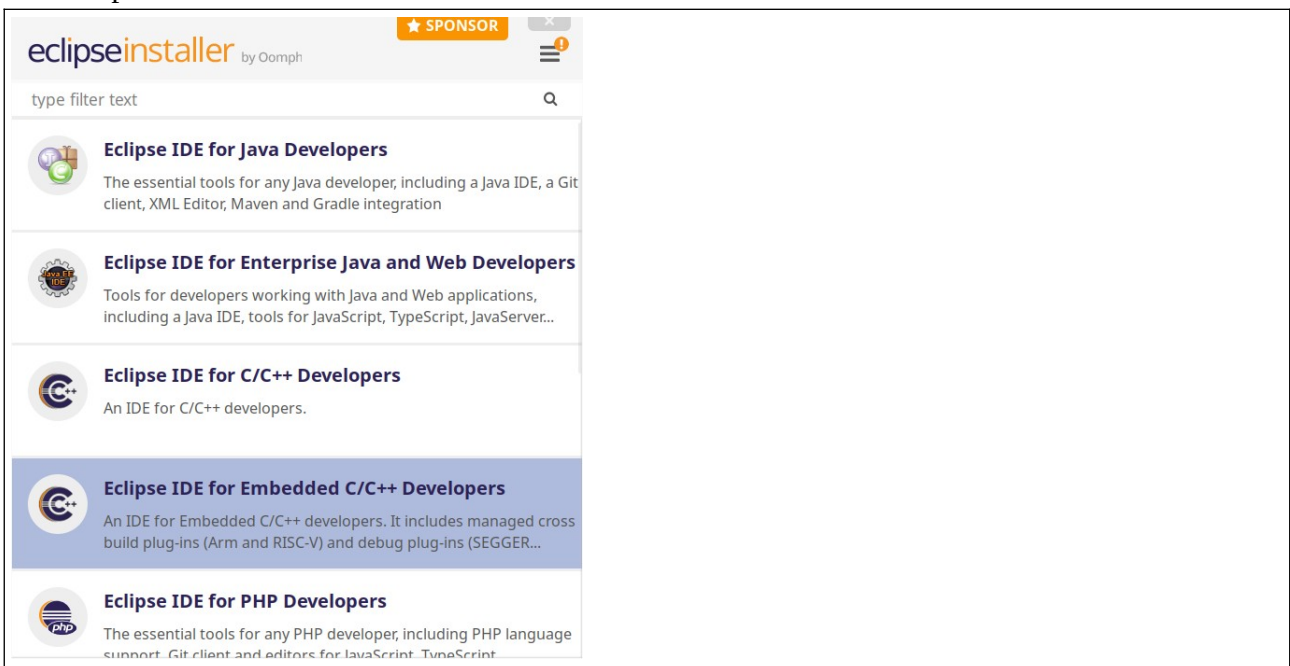
Будет создана папка eclipse-installer

45. Навести курсор на файл eclipse-inst и запустить его, нажав **Enter**

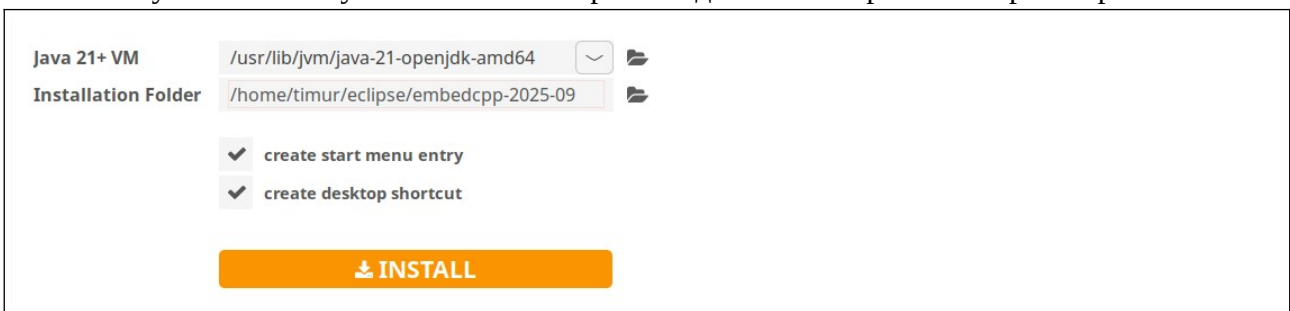


Справочно. Звёздочка спереди означает что файл исполняемый. Исполняемые скрипты можно сделать с использованием команды `chmod +x имя_файла`

46. Выбрать ...Embedded C/C++...



47. Папка установки по умолчанию — eclipse с подпапкой Eclipse и номером версии



Нажать Install

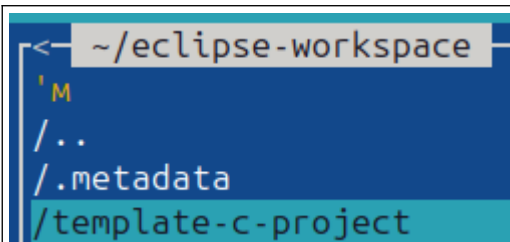
После чего выбрать папку для рабочего окружения, по умолчанию eclipse-workspace в домашней папке.

2.2 Импорт демо-проекта

48. Загрузить демо-проект и распаковать в произвольной папке или воспользоваться загрузкой git

```
git clone https://gitflic.ru/project/mikron-mik32/mik32-examples.git
```

49. Скопировать папку демо-проекта template-c-project в рабочую область



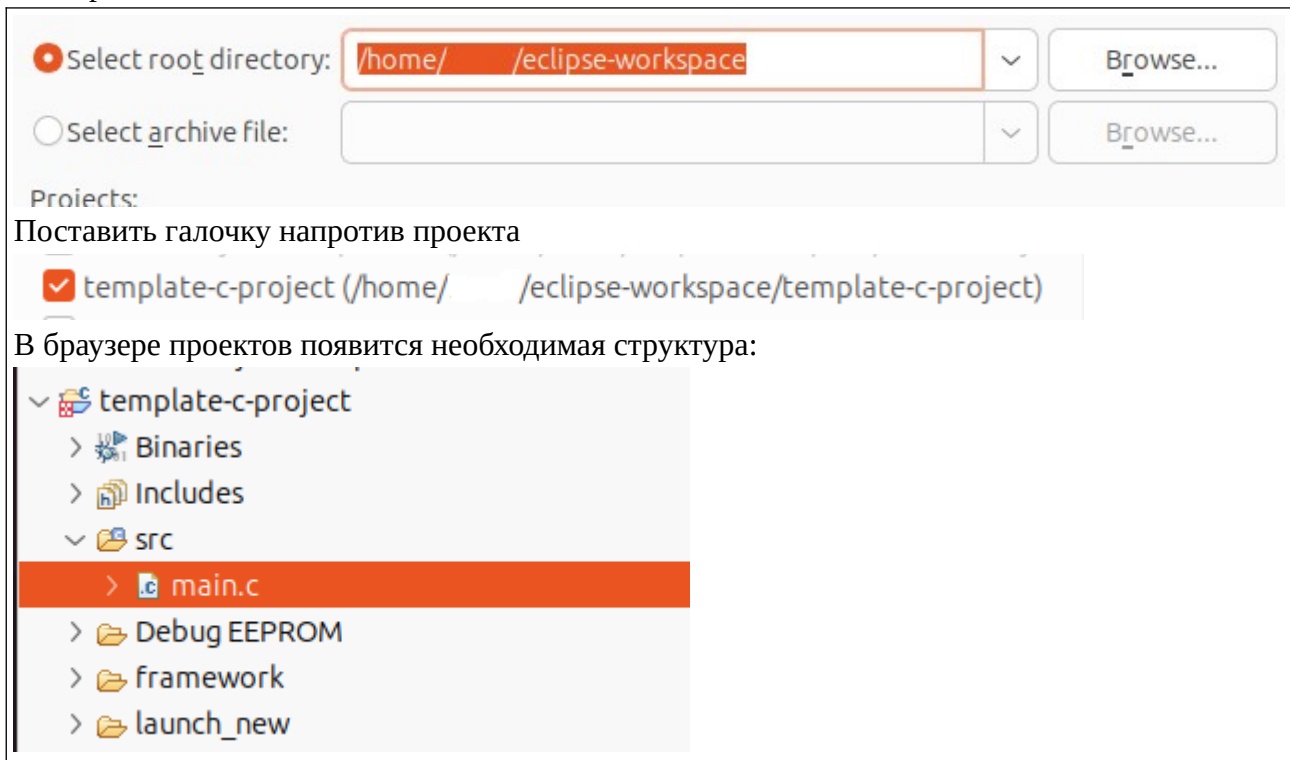
Справочно: Midnight позволяет заходить внутрь архива (см. п. 38)

Справочно: папка .metadata в рабочей области содержит её настройки. При возникновении каких-либо ошибок, проблем, при переустановке Eclipse, можно её удалить (при этом сбросятся все опции!), если эта папка остаётся и устанавливается новая версия, появляется диалоговое окно о преобразовании старой в новую.

Структура папок проекта template-c-project

/Debug EEPROM	исполняемые файлы (каталог появляется после сборки)
/launch_new	файлы конфигурации запуска и отладки
/.settings	настройки Eclipse
/framework	библиотеки Микрон
/src	исходный код пользователя

50. Далее зайти в меню File и выбрать Import. Выбрать General далее Existing Projects Into Workspace



Select root directory: Browse...

Select archive file: Browse...

Projects:

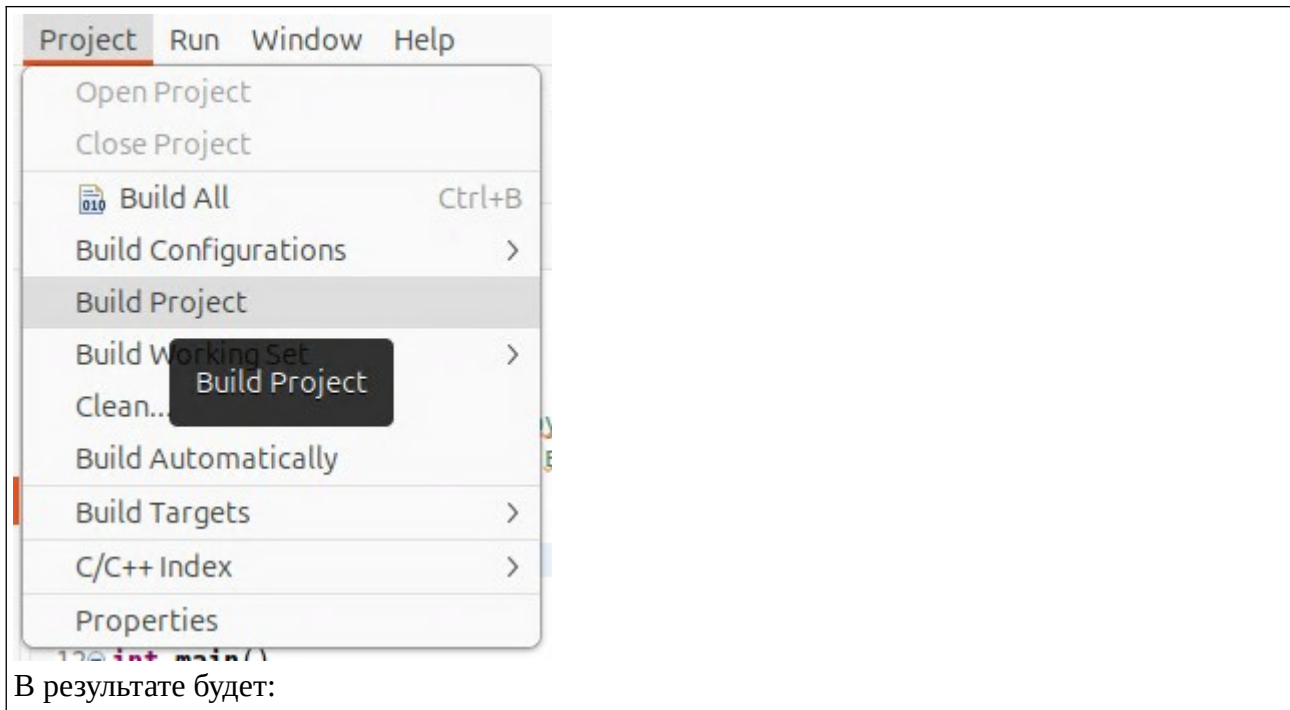
Поставить галочку напротив проекта

template-c-project (/home/.../eclipse-workspace/template-c-project)

В браузере проектов появится необходимая структура:

- template-c-project
 - Binaries
 - Includes
 - src
 - main.c
 - Debug EEPROM
 - framework
 - launch_new

51. После чего попробовать нажать компиляцию или комбинацию клавиш Ctrl+B



Project Run Window Help

- Open Project
- Close Project
- Build All Ctrl+B
- Build Configurations >
- Build Project
- Build Working Set >
Build Project
- Clean...
- Build Automatically
- Build Targets >
- C/C++ Index >
- Properties

В результате будет:

CDT Build Console [template-c-project]

22:44:00 **** Incremental Build of configuration Debug EEPROM for project template-c-project ****

make all

Invoking: GNU RISC-V Cross Print Size

riscv-none-elf-size --format=berkeley "template-c-project.elf"

text	data	bss	dec	hex	filename
------	------	-----	-----	-----	----------

1792	0	1024	2816	b00	template-c-project.elf
------	---	------	------	-----	------------------------

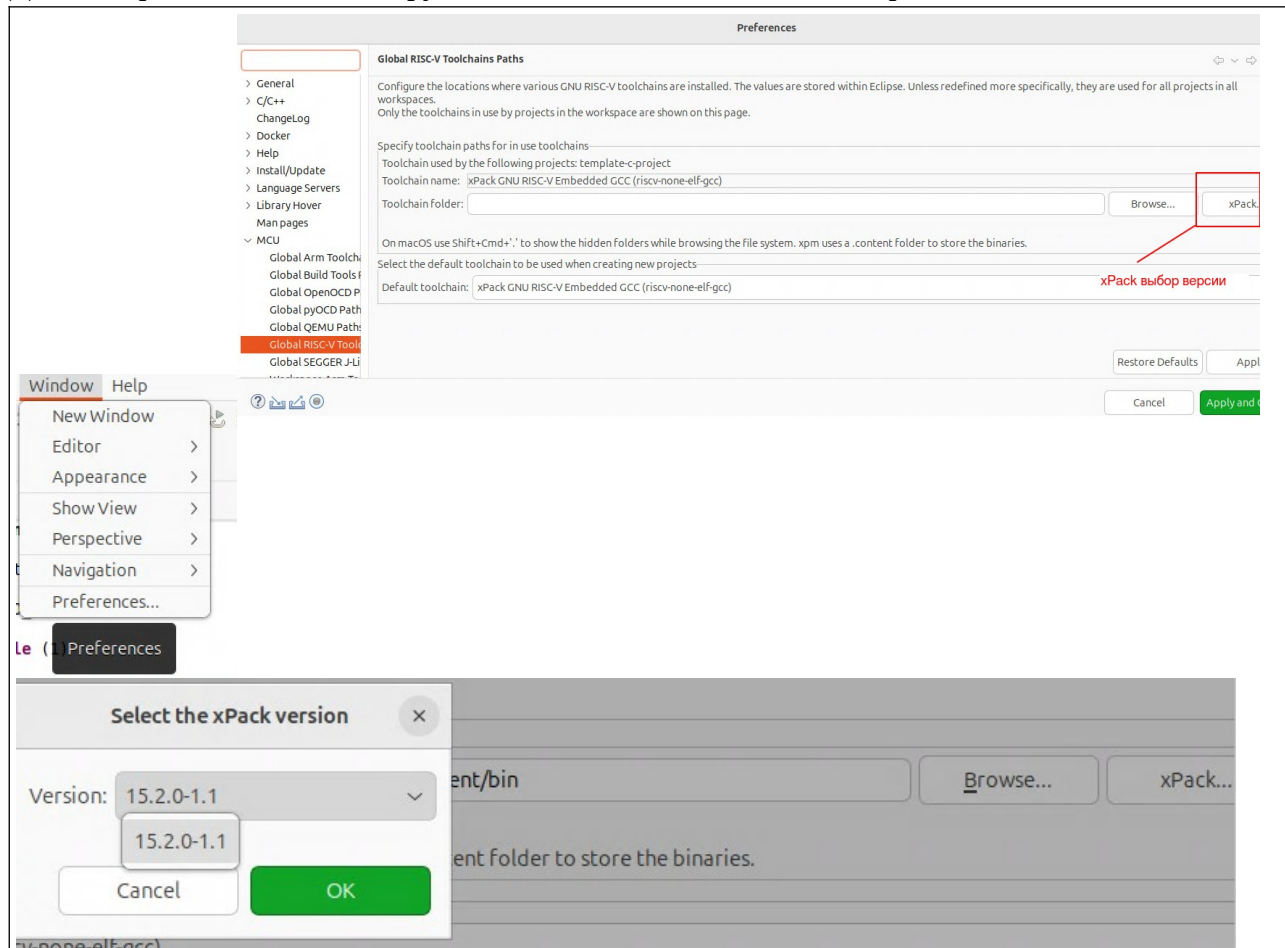
Finished building: template-c-project.siz

22:44:00 Build Finished. 0 errors, 0 warnings. (took 126ms)

2.3 Подготовка к загрузке проекта в контроллер (Program)

52. Подготовка проекта к запуску

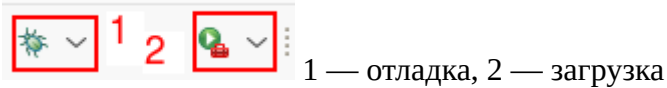
Для выбора глобальных инструментов можно использовать настройки из меню Preferences.



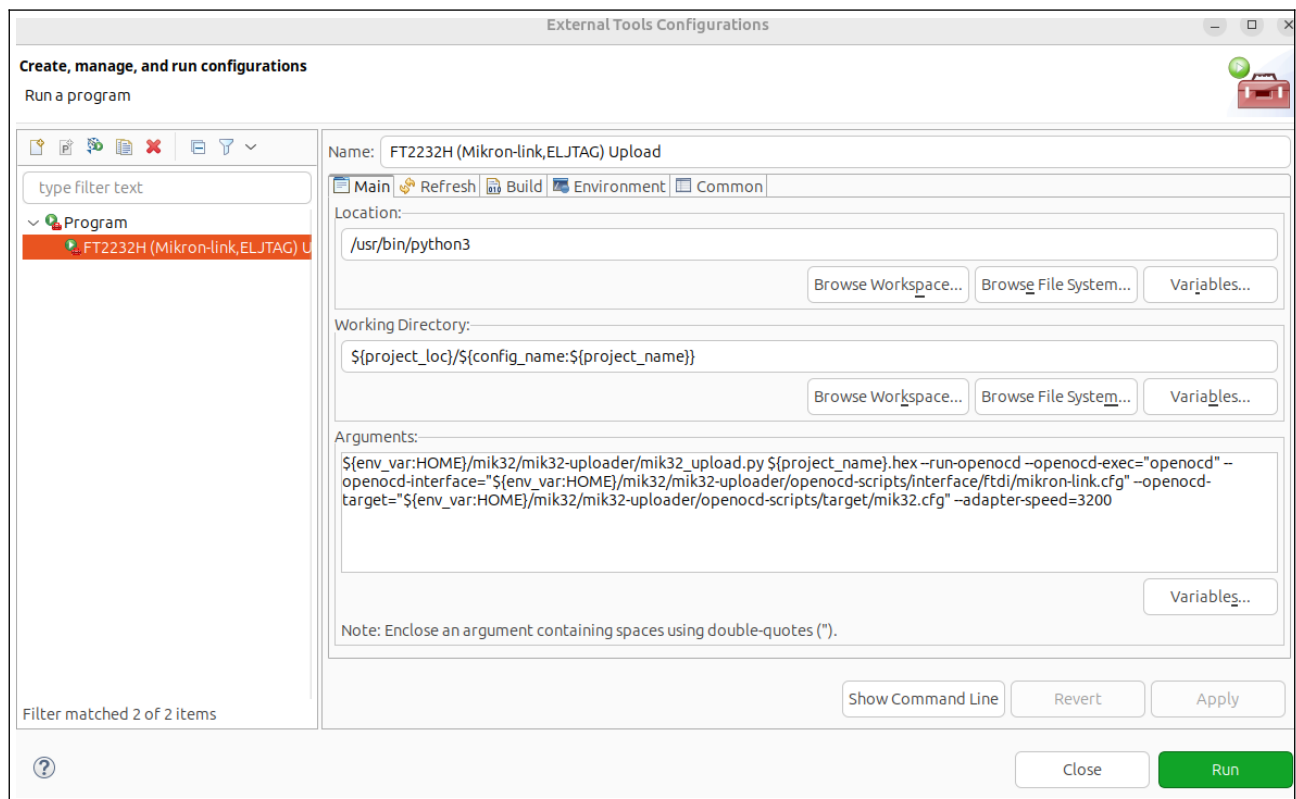
Следует отметить, что помимо этого необходимо использовать как было указано выше переменную среды Path в п. 34.

53. Конфигурация проекта внешних инструментов загрузки с использованием скрипта Python.

Далее необходимо убедиться что имеются правильные настройки загрузки EEPROM без перехода в режим отладки. На панели инструментов имеются соответствующие кнопки:



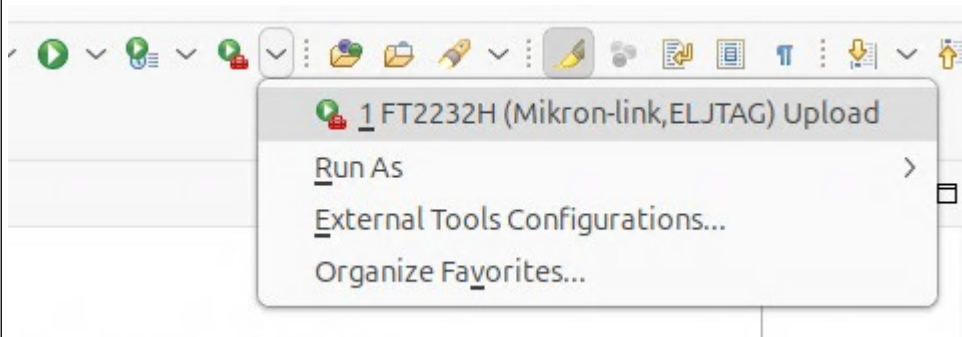
Для просмотра опций и параметров нажать кнопку 2 с раскрывающимся меню и External Tool Configuration



После чего нажать Close, убедившись в правильности путей и настроек.

54. Загрузка проекта в EEPROM с использованием вспомогательного скрипта Python.

Для этого необходимо нажать кнопку в выпадающем меню:





В результате получается следующий вывод консоли:

```
<terminated> FT2232H (Mikron-link,ELJTAG) Upload [Program] /usr/bin/python3 [pid: 46309] (26 окт. 2025 г., 22:45:06 – 22:45:07)
Open On-Chip Debugger 0.12.0+dev-02244-g88b9bd396 (2025-10-23-16:05)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
Info : set servers polling period to 200ms
Info : clock speed 500 kHz
Info : JTAG tap: riscv.cpu tap/device found: 0xde11001 (mfg: 0x000 (<invalid>), part: 0xeb11, ver: 0xd)
Info : JTAG tap: riscv.sys tap/device found: 0xffffffff (mfg: 0x7ff (<invalid>), part: 0xffff, ver: 0xf)
Info : datacount=2 progbufsize=6
Info : Examined RISC-V core; found 1 harts
Info : hart 0: XLEN=32, misa=0x40001104
Info : [riscv.cpu] Examination succeed
Info : [riscv.cpu] starting gdb server on 3333
Info : Listening on port 3333 for gdb connections
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Info : accepting 'tcl' connection on tcp/6666
mik32-uploader-v0.3.3
Using MIK32V2
Clock init... OK!
MCU clock init...
Uploading driver... OK!
Uploading data... OK!
Run driver...
EEPROM writing successfully completed!
[22:45:07] Wrote 1792 bytes in 0.20 seconds (effective 8.7 kbyte/s)
```


Светодиоды на плате должны замигать.

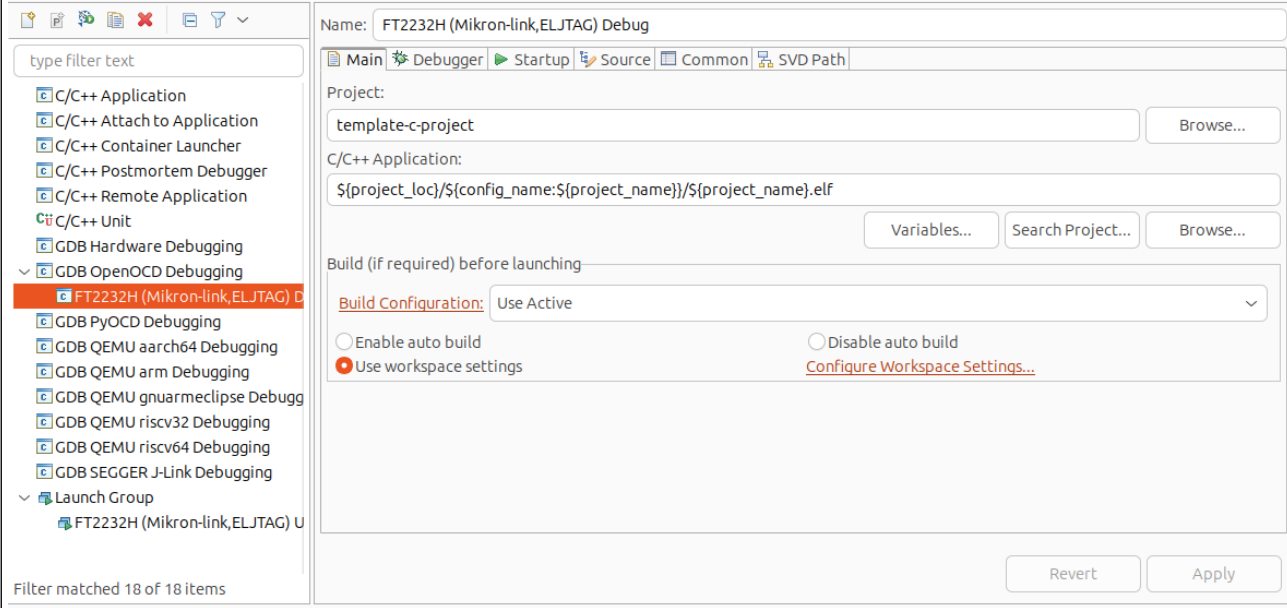
2.4 Отладка проекта (Debug)

55. Настройка отладки.

Для проверки настроек отладки нажать значок  раскрывающегося списка основной кнопки , где выбрать Debug Configuration.

Основные пункты — GDB OpenOCD Debugging слева а также Launch Group.

Create, manage, and run configurations 



Name: FT2232H (Mikron-link,ELJTAG) Debug

Project: template-c-project

C/C++ Application: \${project_loc}/\${config_name:\${project_name}}/\${project_name}.elf

Build (if required) before launching

Build Configuration: Use Active



Enable auto build Disable auto build

Use workspace settings [Configure Workspace Settings...](#)

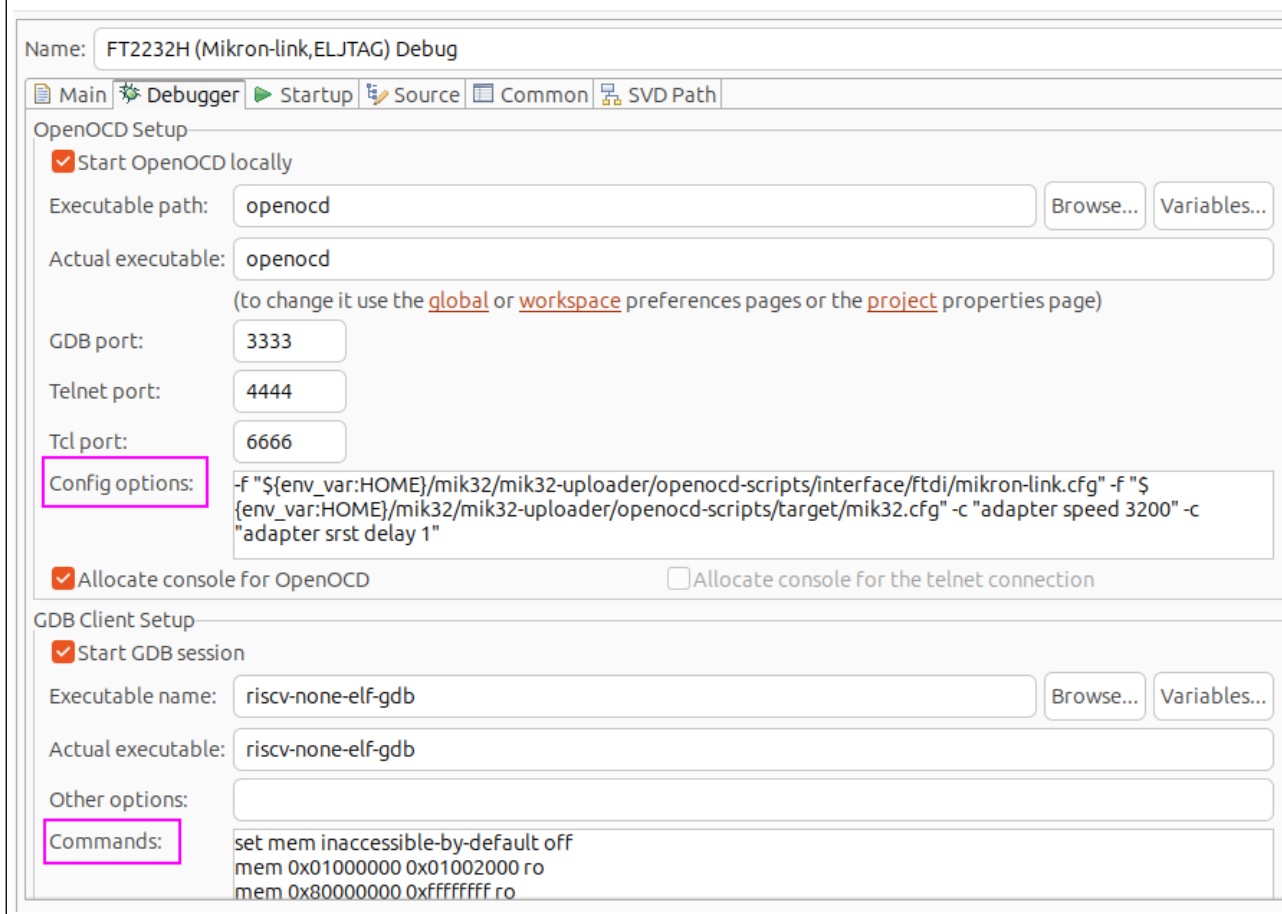
Revert Apply

Close Debug

Исполняемая группа (Launch Group) содержит в себе 2 этапа — загрузка EEPROM и последующую отладку, являющаяся одним из процессов в этой группе.

name	Mode	Action
<input checked="" type="checkbox"/>  Program::FT2232H (Mikron-link,ELJT Run		Wait until terminated
<input checked="" type="checkbox"/>  GDB OpenOCD Debugging::FT2232H Debug		

На вкладке Debugger можно увидеть следующие команды для утилиты OpenOCD

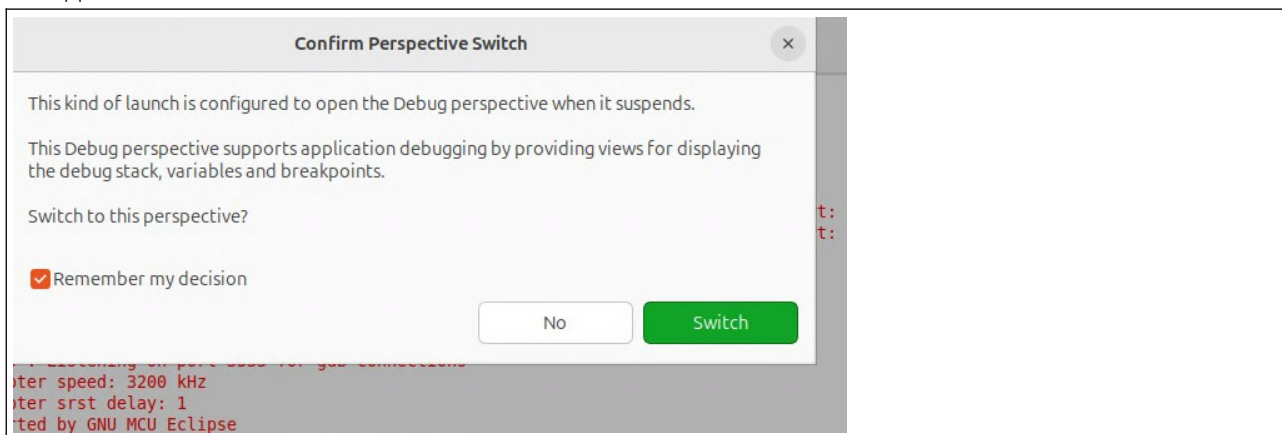


Отладка проекта осуществляется в 2 этапа — сначала запускается скрипт Python для загрузки в EEPROM и далее автоматически следует переход в режим отладки

56. Переход в режим отладки

После проверок опций можно нажать клавишу загрузки и отладки

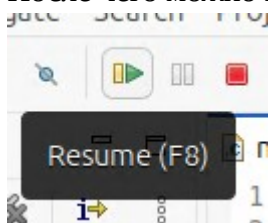
Для отладки выбираем Да — смена рабочей области работы с кодом на рабочую область отладки



Если возникает ошибка «Error: JTAG scan chain interrogation failed: all ones» в консоли, это означает что необходимо увеличить время и выставить параметр adapter srst delay 10 (цифру можно подобрать) в диалоговом окне п. 55.

57. Запуск микроконтроллера

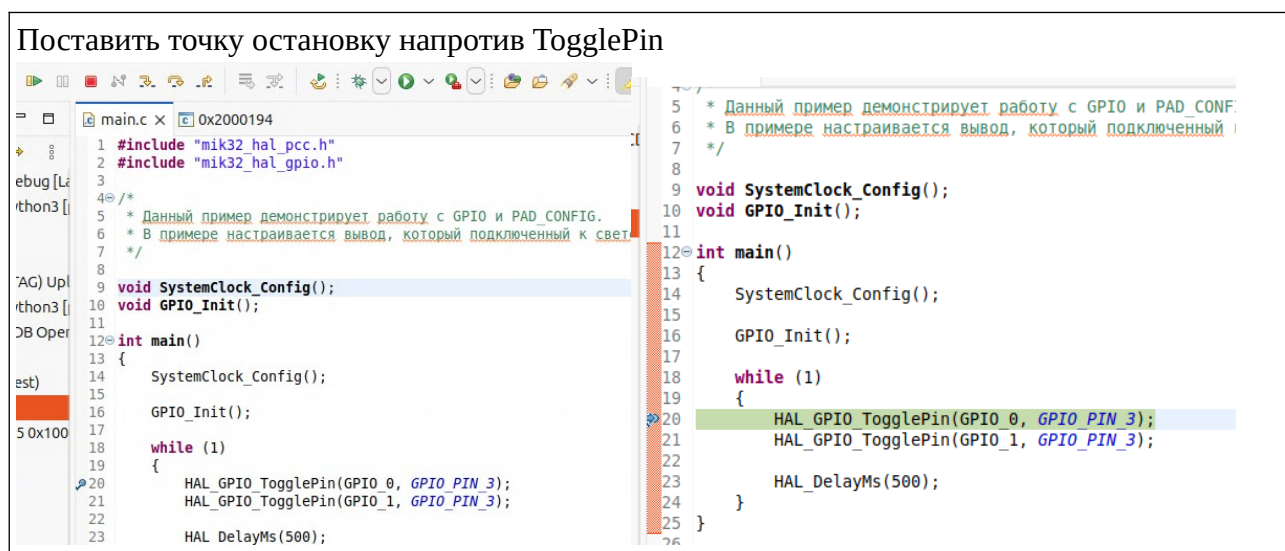
После чего можно нажать «запустить» Светодиоды должны замигать.



58. Отладка


Перейти во вкладку main.c

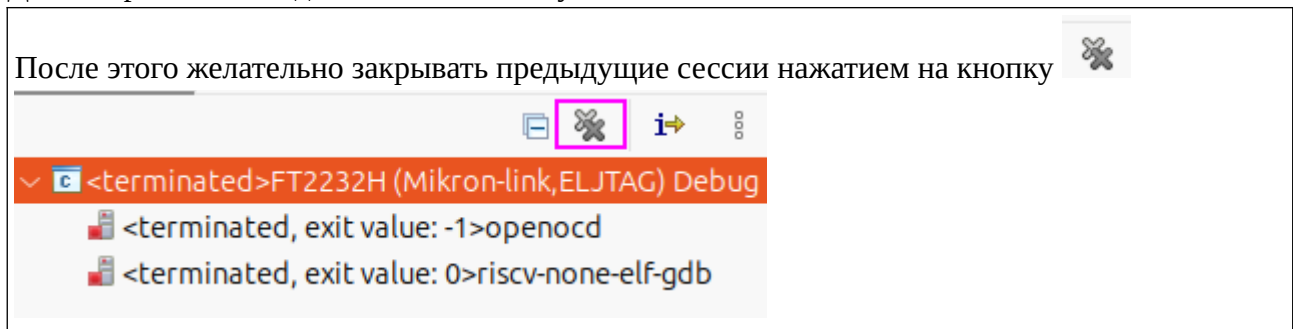
Поставить точку останова напротив TogglePin

A screenshot of an IDE window showing a C source file named 'main.c'. The code includes headers for 'mik32_hal_pcc.h' and 'mik32_hal_gpio.h'. It defines two functions: 'SystemClock_Config()' and 'GPIO_Init()'. The 'main()' function calls these, then enters a 'while (1)' loop. Inside the loop, there are two calls to 'HAL_GPIO_TogglePin' (lines 20 and 21) and a 'HAL_DelayMs(500)' call (line 22). A blue arrow (breakpoint) is positioned to the left of line 20. The IDE interface includes a toolbar at the top and a file explorer on the left.

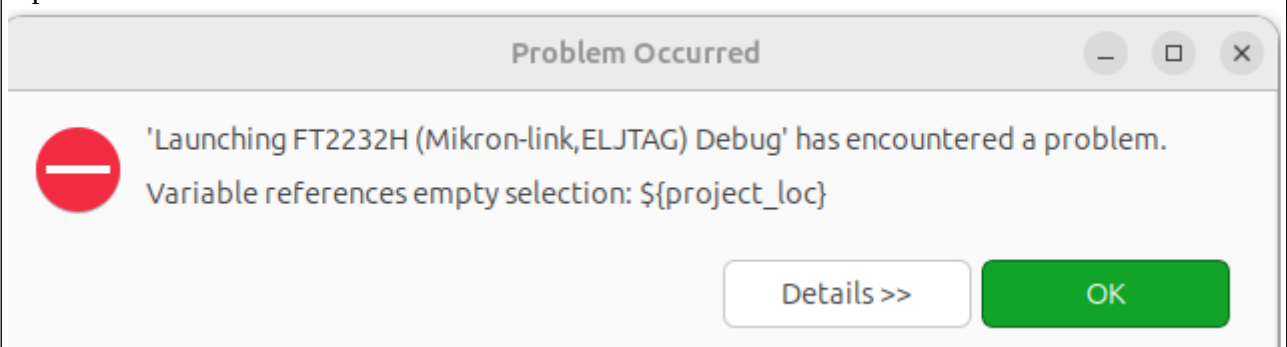
Произойдёт останов. Клавишей F6 можно следовать по шагам соответствующим строкам кода. Нажимая эту клавишу можно управлять миганием.

59. Завершение отладки

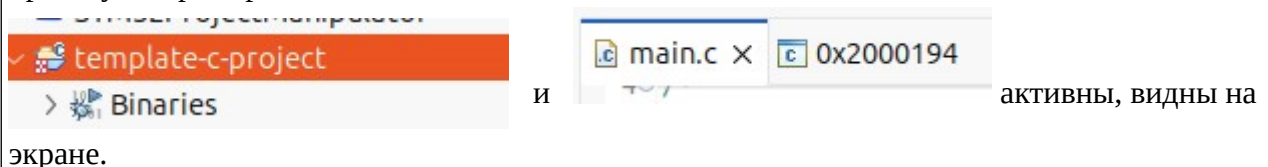
Для завершения отладки нажать клавишу «Стоп» .





При возникновении ошибки



Необходимо чтобы был выделен проект (был в фокусе) в менеджере проектов и был на активном экране, иногда, требуется чтобы был открыт файл, относящийся к активному проекту, например, main.c



ВАЖНО! При повторной компиляции необходимо использовать  выбор

 FT2232H (Mikron-link,ELJTAG) Upload Debug . Если нажать Debug без Upload, отладка будет произведена для предыдущей прошивки и не соответствовать актуально загруженному коду.

60. Отладка с просмотром периферийных регистров. Вновь запустить отладку путём выбора Upload-Debug, поставить точку останова и приостановить работу программы

```
18     while (1)
19     {
20         HAL_GPIO_TogglePin(GPIO_0, GPIO_PIN_3);
21         HAL_GPIO_TogglePin(GPIO_1, GPIO_PIN_3);
22     }
```

61. В режиме отладки можно нажать вкладку Peripherals чтобы посмотреть список регистров периферии.

Вкладка Peripherals — появится окошко загрузки svd файла. Если оно отсутствует, проверить путь к этому файлу, указанный во вкладке SVD диалогового окна настроек п. 55.

The Peripherals view displays the following table:

Peripheral	Address	Description
<input type="checkbox"/> DMA	0x00040000	Прямой доступ к памяти
<input type="checkbox"/> EEPROM_REC	0x00070400	Контроллер EEPROM (8Кб)
<input type="checkbox"/> EPIC	0x00050400	Программируемый контроллер прерываний
<input checked="" type="checkbox"/> GPIO0	0x00084000	Модуль ввода/вывода
<input type="checkbox"/> GPIO1	0x00084400	Модуль ввода/вывода
<input type="checkbox"/> GPIO2	0x00084800	Модуль ввода/вывода

Регистры периферии

Eclipse Embedded CDT CMSIS Packs console

```

2025-10-28 11:42:08
Parsing SVD file "/home/mylabpc2/bin/mik32/mik32v2.svd"...
Completed in 131ms.
    
```

Выбрать интересующий регистр из списка — появится окно просмотра адреса и значения

The Register view displays the following table:

Register	Address	Value
<input checked="" type="checkbox"/> GPIO0	0x00084000	
<input type="checkbox"/> SET	0x00084000	0x00000108
<input type="checkbox"/> STATE	0x00084000	0x00000108
<input type="checkbox"/> CLEAR	0x00084004	0x00000000
<input type="checkbox"/> DIRECTION_OUT	0x00084008	0x0000FFF7
<input type="checkbox"/> DIRECTION_IN	0x0008400C	0x0000FFF7

Регистры периферийного устройства

Расположение файла можно изменить, сделать его единым для всех проектов. Изменить путь к svd-файлу можно в настройках сессии отладки с вкладкой SVD Path