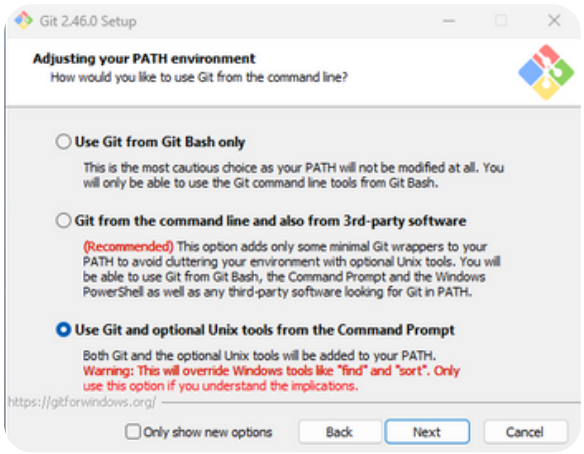
Танг Нано 9К
Синтез и Конфигурация

Установка Git и Клонирование Репозитория

Вы можете скачать Git для Windows с сайтов: https://git-scm.com/download/win или https://gitforwindows.org



После установки Git для Windows и соответствующего инструментария для синтеза FPGA, вы можете клонировать репозиторий через командную строку. Это можно сделать как в обычном терминале, так и в терминале Visual Studio Code (VS Code).

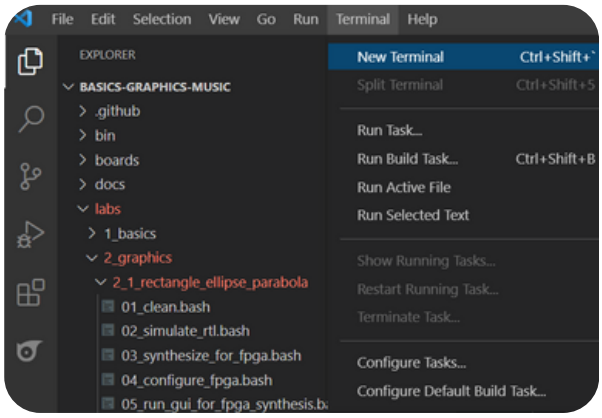
```
git clone https://github.com/yuri-panchul/basics-graphics-music.git
```

Для выполнения скриптов из basics-graphics-music на Linux, необходимо установить продукт с открытым исходным кодом openFPGALoader вдобавок к Gowin EDA. Инструкции по установке доступны по адресу:
https://trabucayre.github.io/openFPGALoader/guide/install.html

Мы рекомендуем выбрать «Использовать Git и опциональные Unix-инструменты из командной строки». Второй вариант также может подойти, однако, если вы выбрали «Использовать Git только из Git Bash», необходимо открыть отдельную консоль Git Bash или использовать Bash-терминал в VS Code.

Создание Терминала в VS Code

- После запуска VS Code откройте директорию, в которую вы клонировали репозиторий basic-graphics-music.
- Далее создайте терминал в VS Code и выполните следующие команды:



Затем вы вводите следующие команды

Для Windows:

```
cd .\labs\2_graphics\2_1_rectangle_ellipse_parabola\
bash 03_synthesize_for_fpga.bash
```

Для Linux:

```
cd labs/2_graphics/2_1_rectangle_ellipse_parabola
./03_synthesize_for_fpga.bash
```

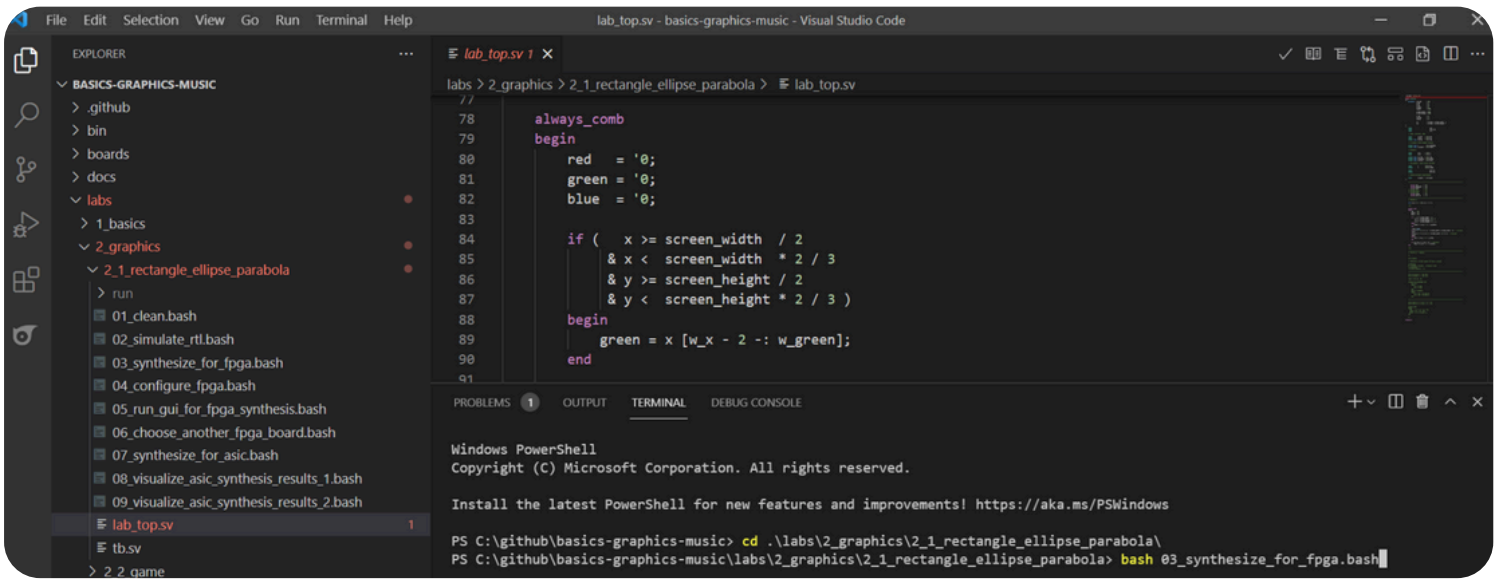
Для платы Tang Nano 9K с 4,3-дюймовым LCD и платой на базе TM1638, введите номер, соответствующий конфигурации *tang_nano_9k_lcd_480_272_tm1638*.

Если у вас нет платы TM1638, используйте *tang_nano_9k_lcd_480_272_no_tm1638*.

Если вы используете открытую инструментальную цепочку на базе Yosys — OSS CAD Suite вместо Gowin EDA, выберите конфигурацию *tang_nano_9k_lcd_480_272_tm1638_yosys*.

Если плата Tang Nano 9K подключена к HDMI-дисплею, используйте конфигурацию *tang_nano_9k_hdmi_tm1638*.

2_1_rectangle_ellipse_parabola Скриншот терминала



Танг Нано 9К
Синтез и Конфигурация

Настройка OSS CAD Suite

Можно экспериментировать с синтезом примеров с использованием открытой инструментальной цепочки OSS CAD Suite вместо Gowin EDA.

Для этого:

1. Скачайте версию инструментальной цепочки для вашей платформы с https://github.com/YosysHQ/oss-cad-suite-build/releases

2. Распакуйте загруженный файл в удобное место, например, *~/oss-cad-suite*.

3. Для Linux:

```
cd ~/oss-cad-suite
source environment
```

4. В примере запустите скрипт *06_choose_another_fpga_board.bash* и выберите номер, соответствующий конфигурации *tang_nano_9k_lcd_480_272_tm1638_yosys*, если используется плата TM1638, или *tang_nano_9k_lcd_480_272_no_tm1638_yosys*, если платы TM1638 нет.

5. Теперь можно запустить скрипт *03_synthesize_for_fpga.bash* для синтеза примера. Обратите внимание, что не все примеры из basics-graphics-music совместимы с OSS CAD Suite на данный момент.

Место установки GOWIN IDE

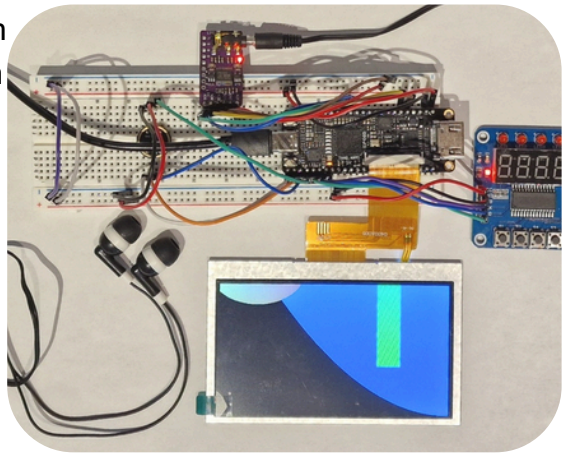
Теперь начнется выполнение скрипта синтеза. Для этой платы, использующей FPGA от Gowin, скрипт ожидает установки Gowin IDE в одном из следующих стандартных мест:

Для Linux:

1. $HOME/Gowin
2. $HOME/gowin
3. /opt/Gowin
4. /opt/gowin
5. /tools/Gowin
6. /tools/gowin

Для Windows:

1. C:\Gowin
2. D:\Gowin
3. E:\Gowin



Пользовательское исходное местоположение установки Gowin можно установить с помощью переменной среды GOWIN_HOME, например:

```
GOWIN_HOME=/home/verilog
```

Вы также можете использовать *GOWIN_VERSION_DIR* для указания местоположения поддерева версии, например:

```
GOWIN_VERSION_DIR=/home/verilog/gowin/0.99
```

Если в процессе синтеза скрипт завершился с ошибкой из-за отсутствия подключения платы, повторный запуск синтеза не требуется. Подключите плату и выполните только конфигурацию:

Проверка Работы Платы

Для Linux:

```
./04_configure_fpga.bash
```

Для Windows или Linux:

```
bash 04_configure_fpga.bash
```

Рекомендуемые лабораторные работы для проверки работоспособности платы:

***1_basics/1_09_hex_counter* – проверяет TM1638.**
***2_graphics/2_1_rectangle_ellipse_parabola* – проверяет графику.**
***3_music/3_1_note_recognizer* – проверяет микрофон.**
***3_music/3_3_note_synthesizer* – проверяет аудио декодер.**
***4_microarchitecture/4_2_fifo/4_2_3_fifo_with_better_debug_1* – лабораторная по микроархитектуре, использующая кнопки для добавления и извлечения значений из очереди FIFO.**
***5_cpu/5_1_schoolriscv* – минималистичный одноконтурный процессор, реализующий подмножество архитектуры RISC-V.**