

Начало работы с Pico

Для начала работы нужно выбрать язык программирования. Raspberry Pi Pico поддерживает два языка: C и MicroPython. C - низкоуровневый язык, который дает вам полный контроль над аппаратными ресурсами и производительностью. MicroPython - высокоуровневый язык, который облегчает написание кода и имеет множество библиотек.

Для того, чтобы запустить MicroPython на Raspberry Pi Pico, нужно выполнить следующие шаги:

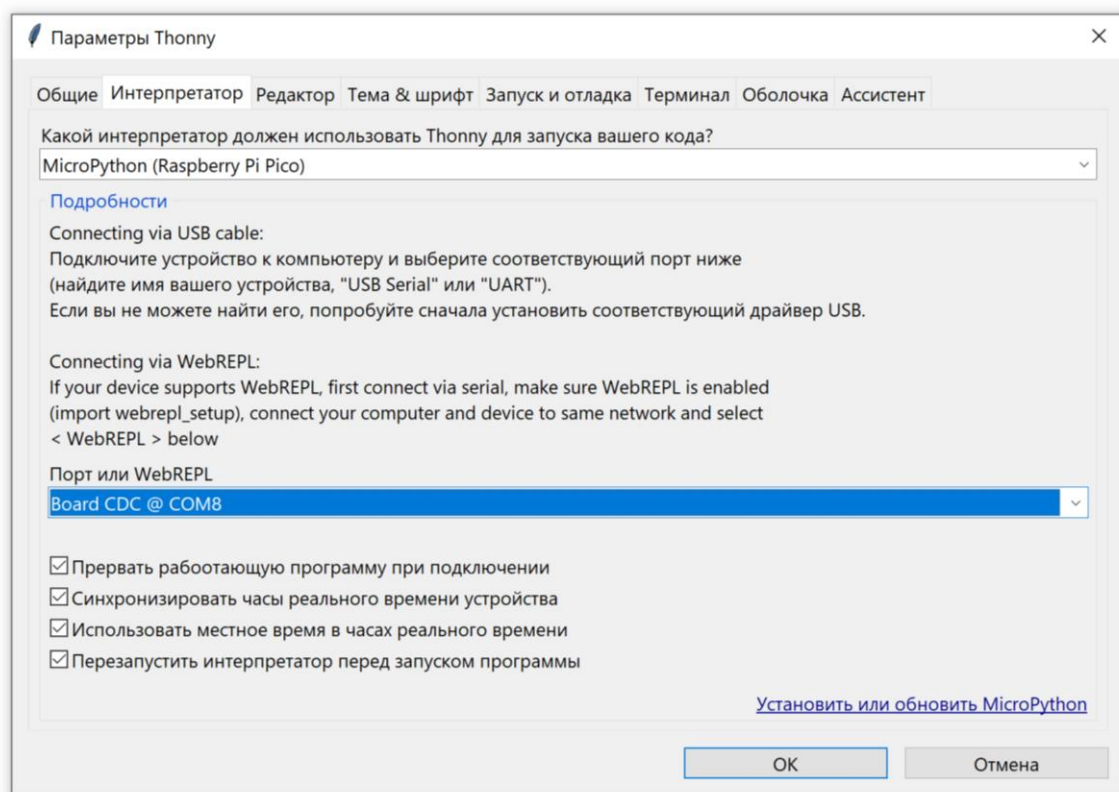
1. Скачайте прошивку-загрузчик MicroPython для Raspberry pi Pico по [ссылке](#). Это файл с расширением .uf2, который содержит все необходимое для запуска MicroPython.
2. Подключите Raspberry Pi Pico к компьютеру с помощью USB-кабеля.
3. Нажмите и удерживайте кнопку **BOOTSEL** на плате, пока не появится новый диск с названием RPI-RP2. Это специальный режим загрузки, который позволяет обновлять прошивку платы.
4. Перетащите файл .uf2, который вы скачали, на диск RPI-RP2. После этого диск исчезнет, а плата перезагрузится с новой прошивкой.
5. Теперь вы можете подключиться к Raspberry pi Pico с помощью программы терминала, например PuTTY или Thonny.

Настройка связи между Pico и Thonny IDE

Thonny - это простая и удобная среда разработки для Python и MicroPython, которая имеет встроенную поддержку Raspberry pi Pico. С помощью Thonny вы можете писать, запускать и отлаживать код, а также просматривать и редактировать файлы.

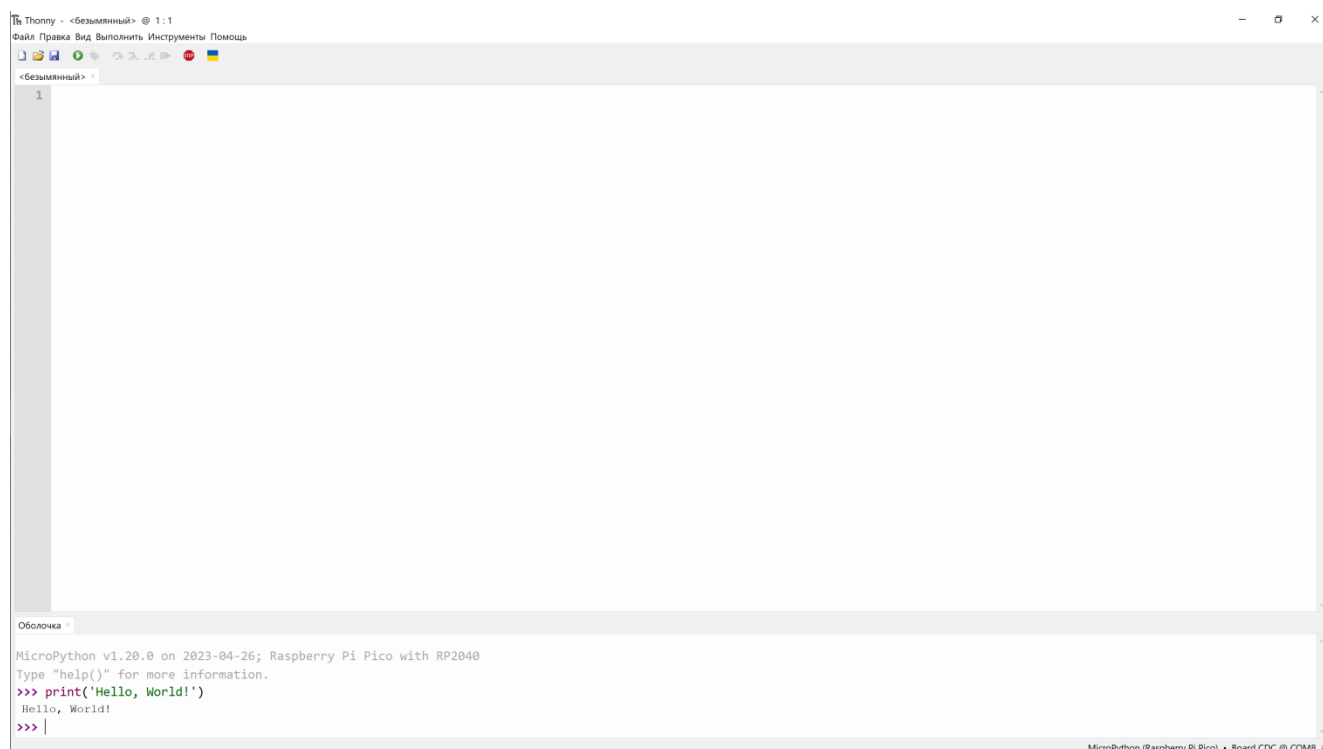
Для того, чтобы установить связь между Raspberry pi Pico и Thonny, вам нужно выполнить следующие шаги:

1. Убедитесь, что вы установили MicroPython, как описано в предыдущем разделе.
2. Скачайте и установите Thonny по [ссылке](#). Если вы работаете на Raspberry Pi 3 или 4, то Thonny уже должен быть установлен.
3. Подключите Raspberry pi Pico к компьютеру. Запустите Thonny и перейдите в меню Инструменты > Параметры. В открывшемся окне выберите вкладку Интерпретатор и в выпадающем списке выберите MicroPython (Raspberry Pi Pico). Выберите порт с вашей платой, она будет называться **Board CDC** или же **CircuitPython**, в зависимости от прошивки платы.



Нажмите ОК, чтобы закрыть окно.

В нижней части окна вы должны увидеть оболочку Python (также называемую REPL - Read-Eval-Print Loop), которая показывает, что Pico подключен и работает. Вы можете вводить команды MicroPython в оболочке и видеть результаты их выполнения.



Снизу в углу видно, что интерпретатор стоит именно Pico.

Теперь вы можете создавать, редактировать и запускать скрипты для Pico в Thonny. Для этого перейдите в меню Файл > Новый или нажмите Ctrl+N, чтобы создать новый файл. Для примера возьмем код мигания светодиодом:

```
# Импортируем модули machine и time

import machine

import time


# Создаем объект Pin для управления светодиодом

led = machine.Pin(25, machine.Pin.OUT)


# Бесконечный цикл

while True:

    # Включаем светодиод

    led.value(1)

    # Ждем полсекунды

    time.sleep(0.5)

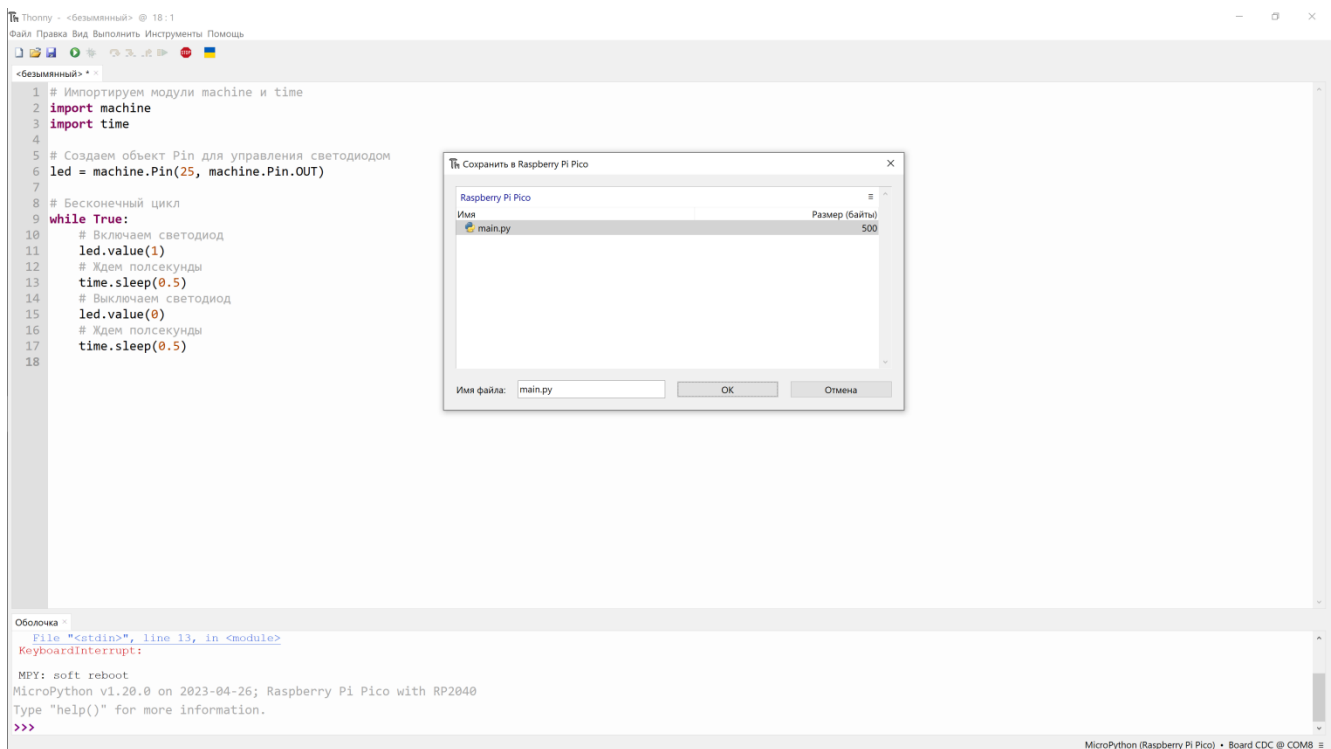
    # Выключаем светодиод

    led.value(0)

    # Ждем полсекунды

    time.sleep(0.5)
```

Введите код в редакторе и сохраните файл на диске RPI-RP2, нажав CTRL+S. Для запуска скрипта нажмите F5.



После запуска кода вы должны увидеть, как мигает светодиод на плате.

Также можно просматривать и редактировать файлы, которые уже находятся на диске RPI-RP2. Для этого перейдите в меню Файл > Открыть или нажмите Ctrl+O, чтобы открыть файловый менеджер. Выберите диск Raspberry Pi Pico дважды щелкните по файлу, который хотите открыть. Вы можете изменить файл и сохранить его.