

# Инструкция по первому запуску ELBEAR ACE-UNO

Для работы с платой ELBEAR ACE-UNO необходимо установить и настроить среду программирования, а так же настроить программатор ELJTAG.

## Настройка среды программирования

Для создания и отладки проектов используется редактор Visual Studio Code с установленным плагином PlatformIO. Процесс установки плагина описан здесь

- [https://wiki.mik32.ru/Быстрый\\_старт\\_в\\_Visual\\_Studio\\_Code](https://wiki.mik32.ru/Быстрый_старт_в_Visual_Studio_Code). Процесс установки пакетов, необходимых для работы с контроллером MIK32, подробно описан здесь

- [https://wiki.mik32.ru/Установка\\_библиотек\\_для\\_разработки\\_под\\_MIK32](https://wiki.mik32.ru/Установка_библиотек_для_разработки_под_MIK32).

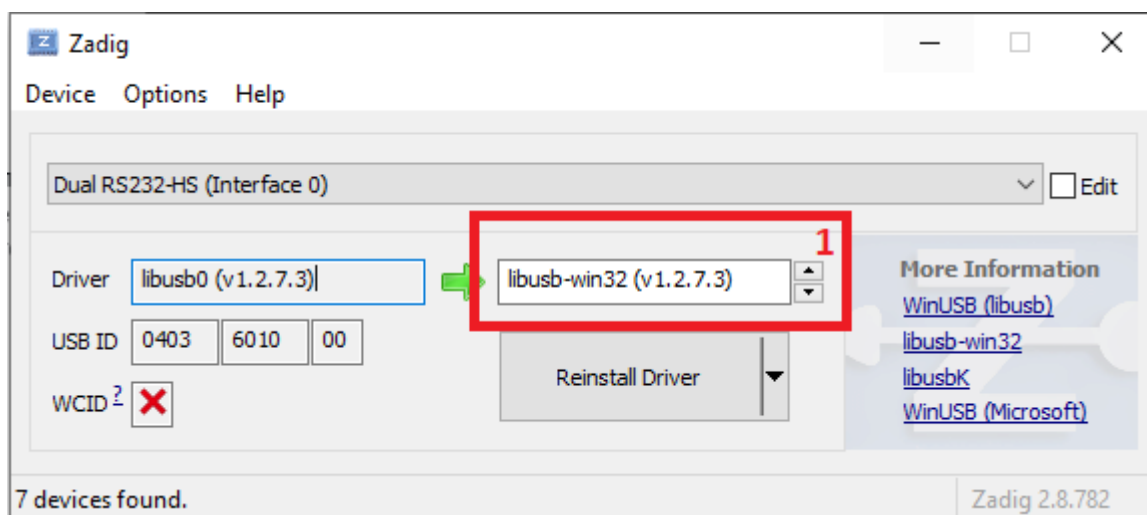
При создании проекта и дальнейшей работе с ним необходимо удостовериться, что в настройках проекта (в файле platformio.ini) и при выборе активного проекта (при создании/открытии каждого нового проекта) везде, где указана версия контроллера mik32v\_, выбрана версия v2 (mik32v2), а не v0.

## Настройка программатора

Для корректной работы программатора ELJTAG необходимо установить подходящие драйвера. Для установки драйверов можно скачать и использовать программу Zadig (<https://zadig.akeo.ie/>).

Последовательность действий:

1. Подключить ELJTAG к ПК.
2. Запустить программу Zadig. Во вкладке Option выбрать опцию List All Devices. После этого в выпадающем списке на главном окне должны определяться некоторые устройства.
3. В выпадающем списке выбрать "Dual RS232-HS (Interface 0)".
4. В строке Driver в окне 1 (см рисунок) выбрать "libusb-win32", нажимая на кнопки со стрелками.



5. Нажать кнопку "Replace Driver" и подождать завершения установки.
6. В выпадающем окне выбрать устройство "Dual RS232-HS (Interface 1)".
7. Повторить пункты 4...5.
8. После установки драйверов необходимо отключить устройство и подключить заново для начала работы.

## Загрузка прошивки в микроконтроллер

Для загрузки созданного в PlatformIO проекта необходимо:

- подключить программатор ELJTAG к ПК,
- подключить плату ELBEAR ACE-UNO к программатору,
- собрать проект в PlatformIO,
- убедиться, что в PlatformIO правильно выбран активный проект,
- нажать кнопку загрузки Upload.

## Создание своего проекта

Для создания нового проекта:

- Открыть vs code, в боковой панели перейти на вкладку PlatformIO
- Выбрать Quick Access -> PIO Home -> Open. Откроется вкладка PIO Home
- На вкладке PIO Home нажать кнопку New Project. В диалоговом окне ввести название проекта, в графе Board выбрать MIK32V2 Generic Board (Mikron). Содержимое графы Framework должно заполниться автоматически после выбора платы. При необходимости можно выбрать расположение для сохранения проекта в графе Location.
- В созданном проекте открыть файл platformio.ini и убедиться, что везде указана версия v2, а не v0.
- В нижнем меню быстрого доступа найти кнопку "Switch PlatformIO Project Environment", нажать ее и выбрать созданный проект в качестве активного.
- Для добавления файлов в боковой панели выбрать "Проводник", раскрыть содержимое проекта, нажать правой кнопкой мыши на папку, в которую необходимо добавить файл, и выбрать "Создать файл". Ввести название файла вместе с расширением, например, "main.c".
- В созданный файл добавить необходимый код. Собрать проект, подключить программатор к ПК и к плате, загрузить прошивку в микроконтроллер.

Пример файла main.c, реализующий моргание пользовательским светодиодом:

```
#include <mcu32_memory_map.h>
#include <pad_config.h>
#include <gpio.h>
#include <power_manager.h>
#include <wakeup.h>

// Вывод, к которому подключен светодиод - PORT_2_7
#define PIN_LED 7

void InitClock()
{
    // Включение тактирования GPIO
    PM->CLK_APB_P_SET |= PM_CLOCK_APB_P_UART_0_M | PM_CLOCK_APB_P_GPIO_0_M |
    PM_CLOCK_APB_P_GPIO_1_M | PM_CLOCK_APB_P_GPIO_2_M;
    // Включение тактирования блока для смены режима выводов
    PM->CLK_APB_M_SET |= PM_CLOCK_APB_M_PAD_CONFIG_M | PM_CLOCK_APB_M_WU_M |
    PM_CLOCK_APB_M_PM_M;
}

void ledBlink()
{
    // Инвертирование вывода
    GPIO_2->OUTPUT ^= 1 << PIN_LED;
    // Задержка
    for (volatile int i = 0; i < 500000; i++)
        ;
}

int main()
{
    // Включение тактирования GPIO
    InitClock();
    // Установка вывода 7 порта 2 в режим GPIO
    PAD_CONFIG->PORT_2_CFG &= ~(0b11 << (2 * PIN_LED));
    // Установка направления вывода 7 порта 2 на выход
    GPIO_2->DIRECTION_OUT = 1 << PIN_LED;

    while (1)
    {
        // Моргание светодиода
        ledBlink();
    }
}
```

Обычный текст

## Использование проектов-примеров

Компания Микрон предоставляет набор примеров на базе PlatformIO, с помощью которых можно протестировать микроконтроллер MIK32. Скачать примеры проектов можно по ссылке - <https://github.com/MikronMIK32/mik32-examples/tree/main>.

copyright ELRON LLC 2024

<https://elron.tech/>

Для работы с примером:

- открыть папку интересующего примера в VS Code
- выбрать открытый проект в качестве активного для версии mik32v2 через кнопку "Switch PlatformIO Project Environment" на нижней панели
- при необходимости изменить номера используемых в программе выводов в соответствии с подключением на вашей плате
- собрать проект и загрузить прошивку в микроконтроллер.