ОТЛАДОЧНАЯ ПЛАТА СТАРТ

Руководство по эксплуатации

Редакция 1.4

Применимо к изделиям:

START-MIK32-V1

Оглавление

Введение

Отладочная плата «Старт» — программируемое устройство для изучения RISC-V-микроконтроллера MIK32 «Амур» (К1948ВК018) и построения макетов устройств на его основе.

Плата оснащена микроконтроллером с минимальной обвязкой, внешней flash-памятью и программатором.

Основное предназначение устройства — отработка схемотехники изделий при помощи макетных плат, изучение характеристик микроконтроллера, обучение основам программирования и электроники.

Основные сведения

Плата содержит:

- Микроконтроллер MIK32 «Амур» (К1948ВК018);
- Внешняя flash память Winbond W25Q32 (32 Мбит);
- Отладчик, совместимый с OpenOCD, совмещенный с USB-UART преобразователем;
- Разъем для подключения модулей расширения;
- Порты ввода-вывода микроконтроллера MIK32;
- Кварцевый резонатор высокочастотный 32 МГц;
- Кварцевый резонатор низкочастотный 32768 Гц;
- Разъем JTAG для отладки внешних устройств;
- Разъем с дополнительными контактами GND и +3.3V;
- Разъем для опционального подключения внешнего отладчика или соединения MIK32 с встроенным отладчиком;
- Разъем для соединения UART0 MIK32 с USB-UART преобразователем или UART сигналами модулей;
- Разъем для выбора режима загрузки MIK32;
- Разъем для подачи напряжения программирования VPROG;
- Два пользовательских светодиода, а также светодиод-индикатор подачи питания и светодиод режима программатора;
- Кнопку сброса, кнопку выхода из режима пониженного энергопотребления, пользовательскую кнопку.

Режимы работы программатора отладочной платы

Программатор может работать в одном из двух режимов: JTAG-отладчика или USB-UART преобразователя. Выбор производится переключателем режима программатора COM/JTAG (обозначение SW1).

При переключении режима происходит повторная инициализация устройства без необходимости снятия питания.

В положении COM программатор определяется как устройство с последовательным интерфейсом USB.

В положении JTAG программатор определяется как START-MIK32-V1 USB JTAG, для совместимости с OpenOCD, при этом светодиод JTAG/COM светится.

USB-UART преобразователь поддерживает стандартные скорости UART до 57600 бод. Нестандартные скорости должны быть кратны 12*32=384, например, 240000 бод, 768000 бод.

Используется формат передачи 8 бит, 1 стоп бит, без контроля четности (8-N-1).

Подготовка к работе с программатором

Для работы программатора в режиме USB-COM не требуется установка дополнительного драйвера, однако в Linux может потребоваться добавить правила udev для устройства с VID=16c0 и PID=05e1.

Для работы программатора в режиме USB-JTAG в Windows потребуется установка драйвера WinUSB для устройства. Проще всего это сделать с использованием Zadig.

Установка драйвера в Windows

Скачайте программу Zadig с сайта https://zadig.akeo.ie. Ссылка на последнюю версию будет в разделе Download.

Перед установкой драйвера USB-JTAG следует установить переключатель режимов работы отладчика в положение JTAG.

Подключите плату по интерфейсу USB к компьютеру, и запустите Zadig. В меню Options нажмите на List All Devices. В выпадающем списке выберите устройство с названием:

- START-MIK32-V1 USB JTAG.

Затем проверьте что выбран драйвер WinUSB в белом поле ввода (нужный выбирать кнопками со стрелками) и нажмите кнопку Replace Driver или Install Driver (Рисунок 1).

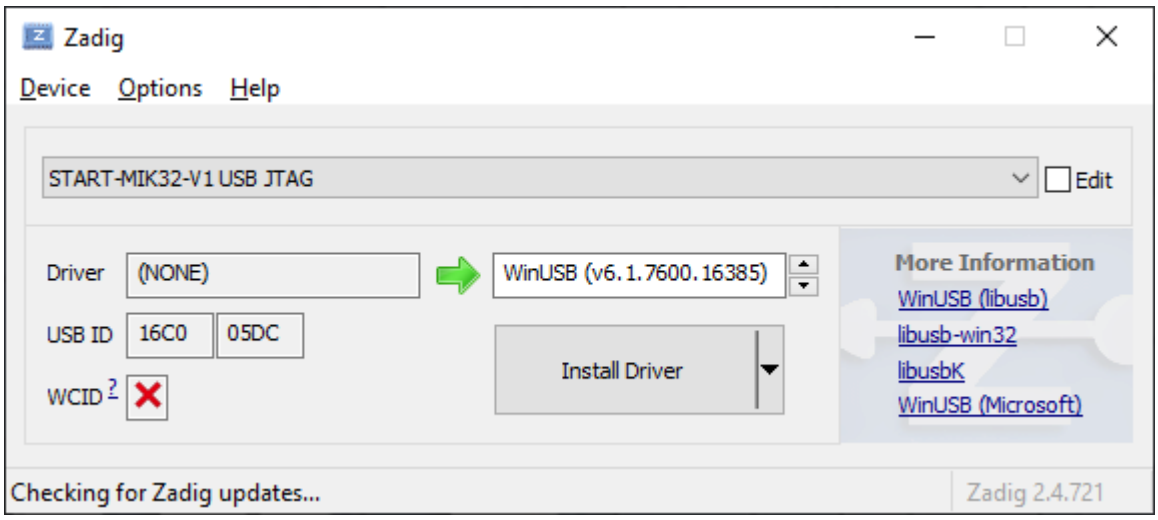


Рисунок 1 – Установка драйвера с использованием Zadig

После установки драйвера отключите устройство и подключите заново для начала работы.

Добавления правил udev в Linux

Приведем пример добавления правил USB-JTAG в Ubuntu Linux.

Необходимо создать файл /etc/udev/rules.d/start-link.rules с содержимым:

SUBSYSTEM=="usb", ATTR{idVendor}=="16c0", ATTR{idProduct}=="05dc", MODE="0666"

После чего потребуется перезагрузить компьютер.

Начало работы с MIK32 IDE v1.2.2

Получить программу можно на сайте АО Микрон на вкладке среды разработки: https://mikron.ru/products/mikrokontrollery/mk32-amur/.

Программа поставляется в виде ZIP-архива. Для начала работы требуется распаковать архив в любую папку. Среда разработки запускается файлом start-ide.cmd. После запуска появится меню выбора рабочих областей, в котором следует нажать кнопку Launch.

При первом запуске потребуется открыть проект-шаблон: перейти в File->Open Projects From File System…. Откроется окно Import Projects from File System or Archive, в котором потребуется выбрать путь к проекту (Рисунок 1). Нажмите кнопку Directory… и в папке {путь к сборке}\mik32-ide-v1-2-2\templates-project выберете один из шаблонных проектов.

Проект template-c-project содержит в себе папку framework, содержащую компоненты для сборки. Библиотеки, стартовый файл, скрипт линковки хранятся внутри проекта.

Проект template-c-project-shared содержит ссылки на директорию framework-shared в корне сборки, библиотеки и другие компоненты хранятся вне проекта и разделены между подобными проектами.

После указания пути, в центральном списке будет отмечен выбранный проект. Для завершения открытия проекта следует нажать кнопку Finish.

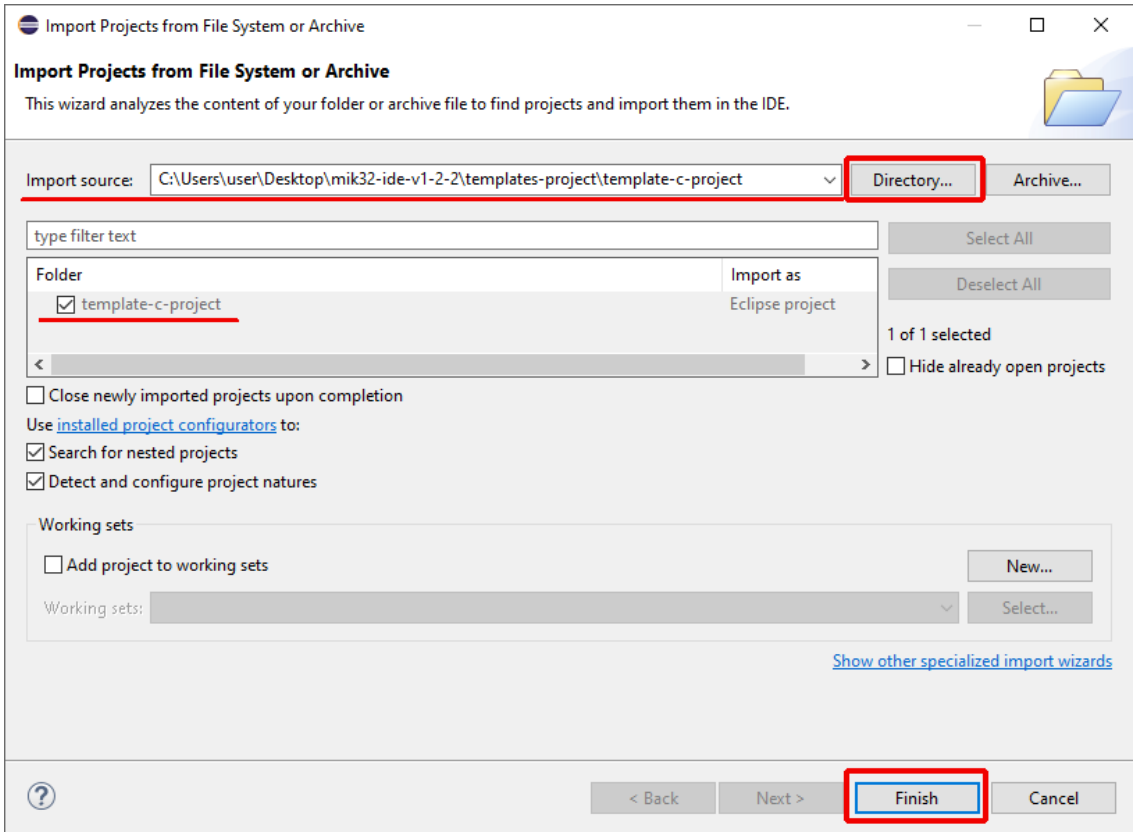


Рисунок 1. Окно открытия проекта

Создание проекта

Создание нового проекта выполняется копированием проекта template-c-project и изменением названия проекта. Для этого в панели Project Explorer следует выбрать проект template-c-project, скопировать и вставить его горячими клавишами Ctrl+C или Ctrl+V или через контекстное меню пунктами Copy и Paste. При выполнении вставки откроется окно Copy Project, в котором следует задать название нового проекта (Рисунок 2).

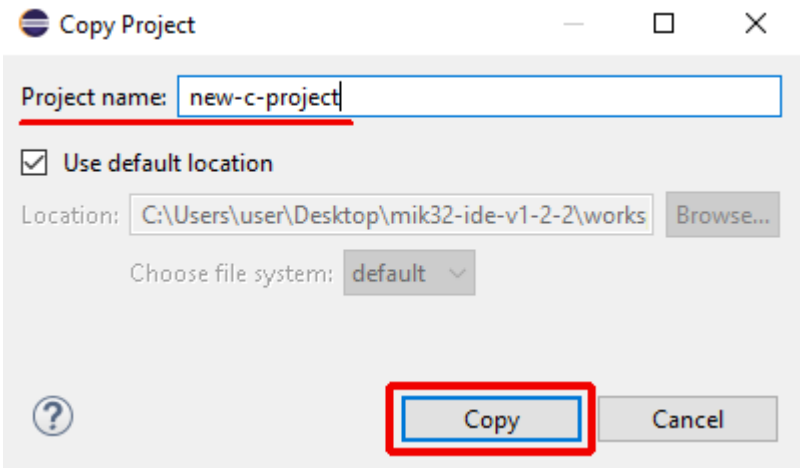


Рисунок 2. Окно копирования проекта

Использование примеров

Чтобы начать работу с примерами, следует скопировать файл исходного кода примера из папки src соответствующего примера в папку src созданного проекта. Примеры находятся в папке examples рабочей области workspace.

Сборка прошивки

При нажатии на кнопку Build  будет собран проект, папка или файл которого были выделены. По умолчанию выбрана конфигурация Debug EEPROM, в которой настроено сохранение программы во внутренней EEPROM памяти. В сборке доступны конфигурации Debug RAM, с сохранением программы в ОЗУ, и Debug Flash, с сохранением программы во внешнюю флеш память.

Загрузка прошивки

Прошивка в микроконтроллер загружается с помощью механизма запуска внешних инструментов. Перед загрузкой, необходимо выбрать папку или файл соответствующего проекта. Для этого необходимо в верхней панели нажать на треугольник рядом с кнопкой External Tools  для открытия выпадающего меню внешних инструментов (Рисунок 3). Нажатие на пункт

соответствующий нужному программатору запустит процесс загрузки прошивки.

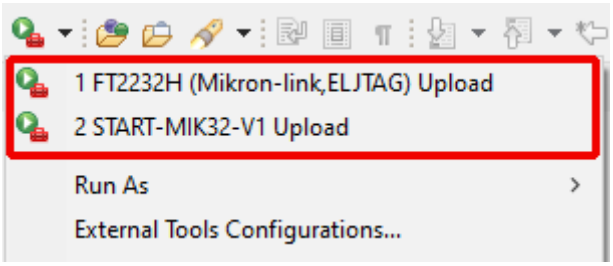


Рисунок 3. Выпадающее меню внешних инструментов

На данный момент есть готовые настройки для программаторов на основе микросхемы FT2232H с JTAG на канале 1 и программатора на плате START-MIK32-V1.

Запуск отладки

Запуск отладки выполняется аналогично загрузке прошивки, запуском подготовленной отладочной конфигурации. Перед запуском отладки, необходимо выбрать папку или файл соответствующего проекта. Для этого необходимо в верхней панели нажать на треугольник рядом с кнопкой Debug Configutaions…  для открытия выпадающего меню конфигураций отладки. Нажатие на пункт соответствующий нужному программатору запустит процесс загрузки прошивки. Конфигурации разделены на отладочные и группы запуска, обозначающиеся Debug и Debug Upload соответственно. Конфигурация, обозначенная Debug выполняет запуск отладки выбранной прошивки, без выполнения загрузки. Группы запуска Debug Upload сначала запускают загрузку, затем запускают отладку.

Расширенная конфигурация MIK32 IDE v1.2.2

Состав сборки

В корневом каталоге сборки можно найти следующие директории:

- build-tools – используется программа make для сборки;
- curl – используется для скачивания svd описания из ветки main репозитория;
- eclipse – среда разработки eclipse ide;
- examples – набор примеров для копирования исходников в проекты;
- framework-shared – компоненты для сборки, подключаемые к проектам;
- git – клиент для обновления компонентов для сборки и примеров из репозитория;
- openocd – для подключения к контроллеру для загрузки прошивки и отладки;
- riscv-gcc – набор инструментов для компиляции;
- templates-debug-upload-configurations – конфигурации загрузки и отладки для импорта в новые рабочие области;
- templates-project – шаблоны проектов с настройками сборки;
- uploader – программа для загрузки прошивок в память микроконтроллера;
- workspace – рабочая область по умолчанию.

Добавление нового отладчика

Для добавления нового отладчика перейдите в раздел Debug Configurations, выберите одну из существующих конфигураций в разделе GDB OpenOCD Debugging в списке слева, например, FT2232H (Mikron-link,ELJTAG) Debug. Затем, в верхней части списка над строкой поиска, нажмите кнопку дублирования  . Выберите созданную конфигурацию. На вкладке Debugger в поле Config options измените путь в

первом аргументе -f на путь, соответствующий скрипту необходимого отладчика (Рисунок 4). Скрипты для многих отладчиков уже есть в составе OpenOCD. Например, для выбора отладчика J-Link, пропишите следующий путь: "${eclipse_home}/../openocd/openocd/scripts/interface/jlink.cfg".

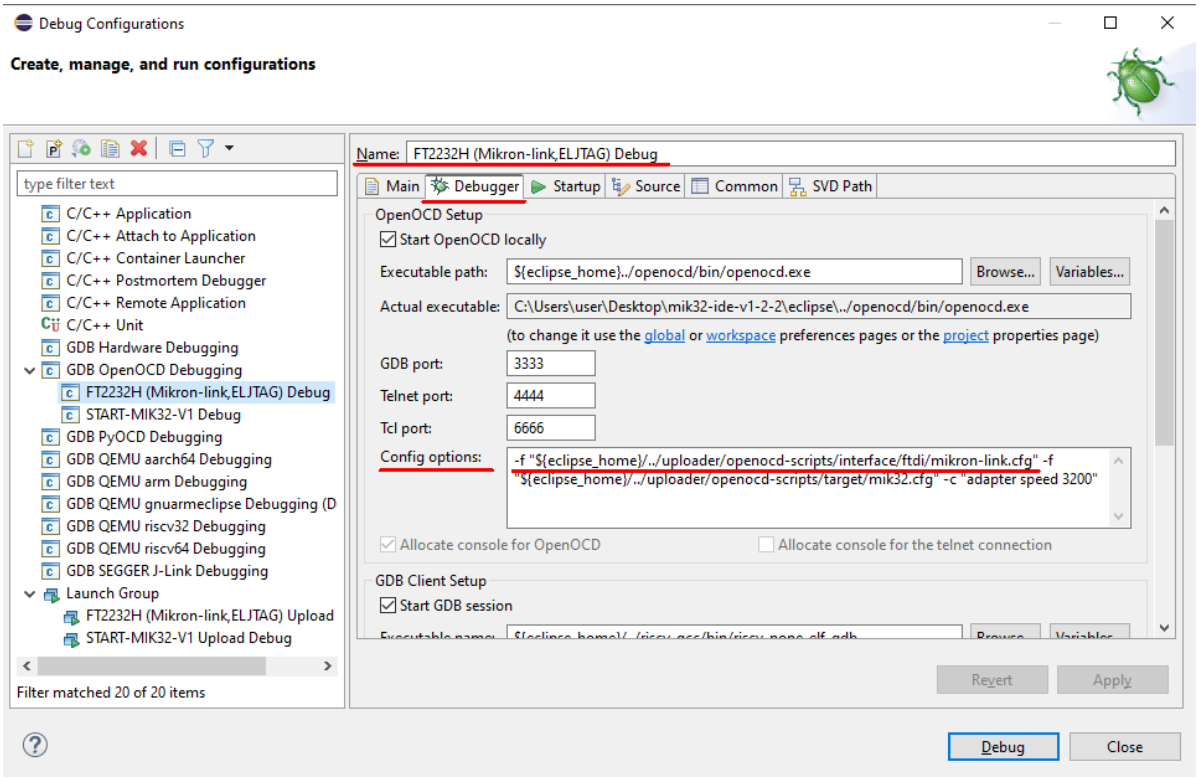


Рисунок 4. Изменение отладчика в конфигурации отладки

Помимо настроек отладки, аналогичным образом нужно дублировать конфигурацию внешнего инструмента для загрузки прошивки (см. раздел Загрузка прошивки) в окне настроек внешних инструментов External Tools, и заменить скрипт для отладчика на вкладке Main в поле Arguments. Затем, в меню отладочных конфигураций Debug Configurations следует продублировать группу запуска Upload Debug и заменить конфигурацию загрузки и конфигурацию отладки. Это можно сделать в меню Edit Launch Configuration, которое открывается по кнопке Edit.

Создание конфигураций сборки

Для создание дополнительной конфигурации сборки выберите любой папку или файл, относящийся к проекту, и выберите в строке меню среды

разработки Project→Build Configurations→Manage…. Откроется окно управления конфигурациями сборки, в котором можно выбрать активную конфигурацию, удалить и создать новую. Для создания новой конфигурации нажмите кнопку New…. В поле Name введите название новой конфигурации, а в группе Copy settings from выберите пункт Existing configuration и конфигурацию, настройки которой будут скопированы в созданную конфигурацию.

Чтобы изменить настройки конфигурации сборки выберите любой папку или файл, относящийся к проекту, и выберите в строке меню среды разработки Project→Properties. В дереве слева выберите пункт C/C++ Build→Settings. В правой части окна откроются настройки конфигурации сборки. В верхней части окна выбирается настраиваемая конфигурация.

Чтобы изменить скрипт линковки, например, для изменения памяти, в которой будет размещаться прошивка, на вкладке Tool Settings настроек конфигурации, в дереве выберите GNU RISC-V Cross Linker→General. В поле Script files можно изменить скрипт линковки. В текущем framework имеются варианты eeprom.ld, spifi.ld, ram.ld.

Обновление компонентов сборки

Компоненты для сборки программ, находятся в папке framework-shared в корневой директории сборки. Для их обновления из GitHub репозитория MikronMIK32/framework-mik32v2-sdk следует запустить скрипт update-framework.bat.

Компоненты для сборки программ в папках проектов находятся в директории framework. Для их обновления из GitHub репозитория MikronMIK32/framework-mik32v2-sdk следует запустить скрипт update-framework.bat в каталоге проекта.

Примеры находятся в папке examples в корневой директории сборки. Для их обновления из GitHub репозитория MikronMIK32/framework-mik32v2-sdk следует запустить скрипт update-framework.bat.

Программа для прошивки памяти контроллера находится в папке uploader в корневой директории сборки. Для его обновления следует перейти по ссылке https://github.com/MikronMIK32/mik32-uploader/releases/latest и скачать архив mik32-uploader-vX.Y.Z-win32-x64.zip. Содержимое папки uploader нужно удалить, затем содержимое директории mik32_upload скачанного архива перенести в каталог uploader.

Файл описания регистров mik32v2.svd находится в рабочей области workspace. Для его обновления следует запустить скрипт update-mik32v2-svd.bat.

Установка PlatformIO IDE в VS Code

Установка Visual Studio Code

Скачайте и установите среду по ссылке:

https://code.visualstudio.com/download.

Установка плагина PlatformIO

Для разработки ПО под микроконтроллеры требуется установить плагин PlatformIO.

Выберите пункт Extensions или нажмите сочетание клавиш Ctrl+Shift+X. Найдите его в списке плагинов и нажмите Install (Рисунок 2).

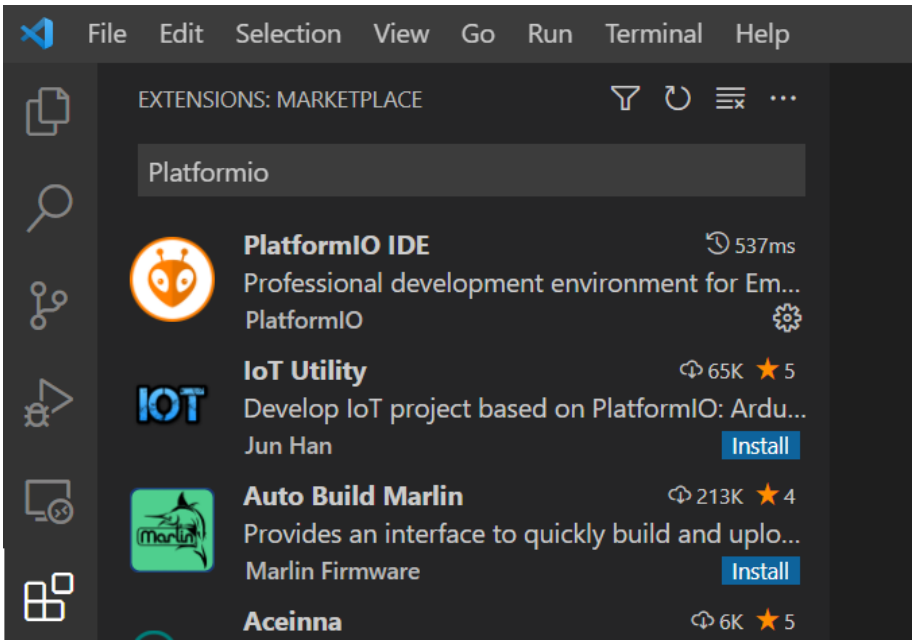


Рисунок 2 – Установка плагина PlatformIO

Дождитесь окончания установки (статус отображается на панели снизу) и перезапустите программу.

Теперь при запуске через некоторое время в панели инструментов (слева) появится логотип Platformio (Рисунок 3).

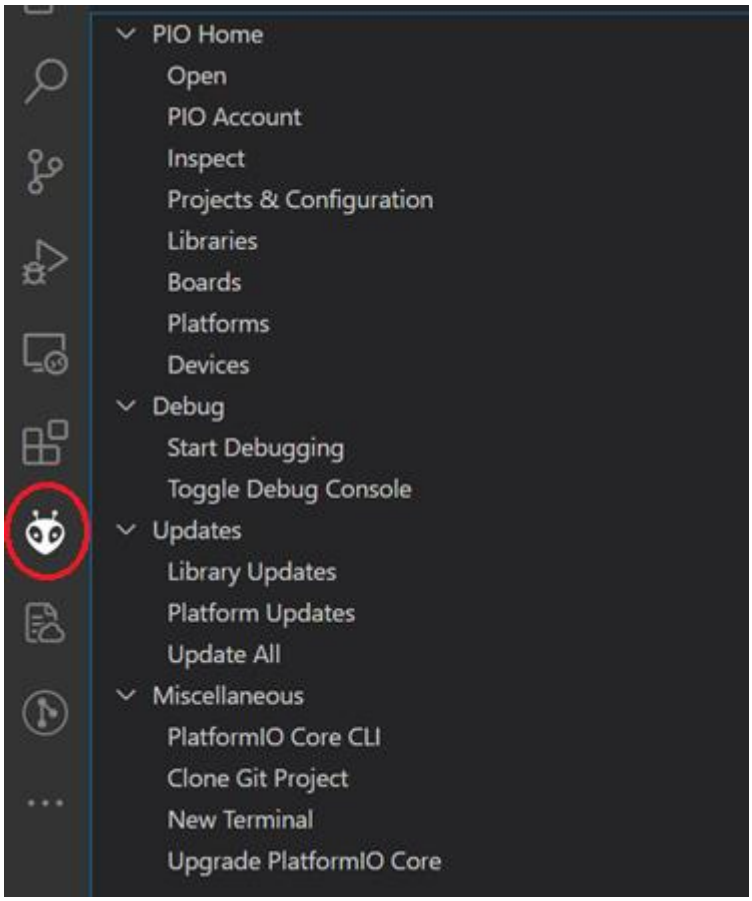


Рисунок 3 – Логотип PlatformIO в боковом меню

Подготовка компонентов для разработки на MIK32

После установки расширения потребуется распаковка архива по пути %USERNAME%\.platformio (для Windows) или ~/.platformio (для Linux). В Windows при установке VS Code на уровне системы (System Installer) директорию можно найти по пути C:\.platformio. В Linux каталоги, название которых начинается с точки, считаются скрытыми, поэтому потребуется включить отображение скрытых файлов.

Необходимо скачать архив из последнего релиза по адресу https://github.com/MikronMIK32/mik32-platformio/releases/. Последние релизы находятся в верхней части экрана, в его карточке необходимо найти группу Assets и в списке выбрать элемент с названием mik32-platformio-{*версия*}.zip. Нажатие по этому элементу запустит скачивание архива. Скачивать файлы, обозначенные как Source Code не следует, поскольку в них будут отсутствовать подмодули пакетов. Затем нужно распаковать содержимое архива по вышеуказанному пути.

После распаковки всех архивов должна получиться следующая структура .platformio:

1. packages
   1.1. framework-mik32v2-sdk
      1.1.1. hal
      1.1.2. shared
   1.2. tool-mik32-uploader
2. platforms
   2.1. mik32

Перезапустите **VS Code.** Теперь у вас подготовлена среда для программирования MIK32.

Работа в среде PlatformIO

Создание проекта

Дождитесь появления значка Platformio в левой части окна и нажмите на него. Откроется панель, в которой следует нажать на Open. Откроется вкладка PIO Home, нажмите на кнопку New Project для создания нового проекта (Рисунок 4).

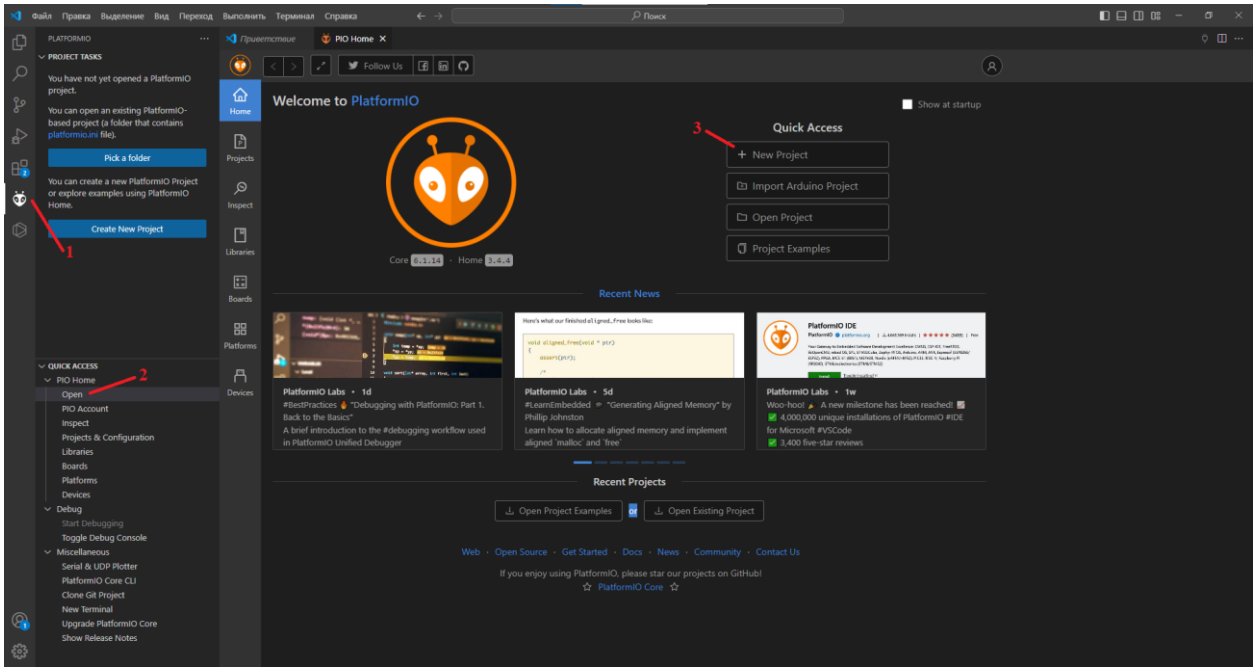


Рисунок 4 – Открытие меню создания проекта

В диалоговом окне Project Wizard в поле Name введите имя проекта, в выпадающем списке Board введите mik32 и выберите START-MIK32-V1 (Mikron), нажмите Finish (Рисунок 5).

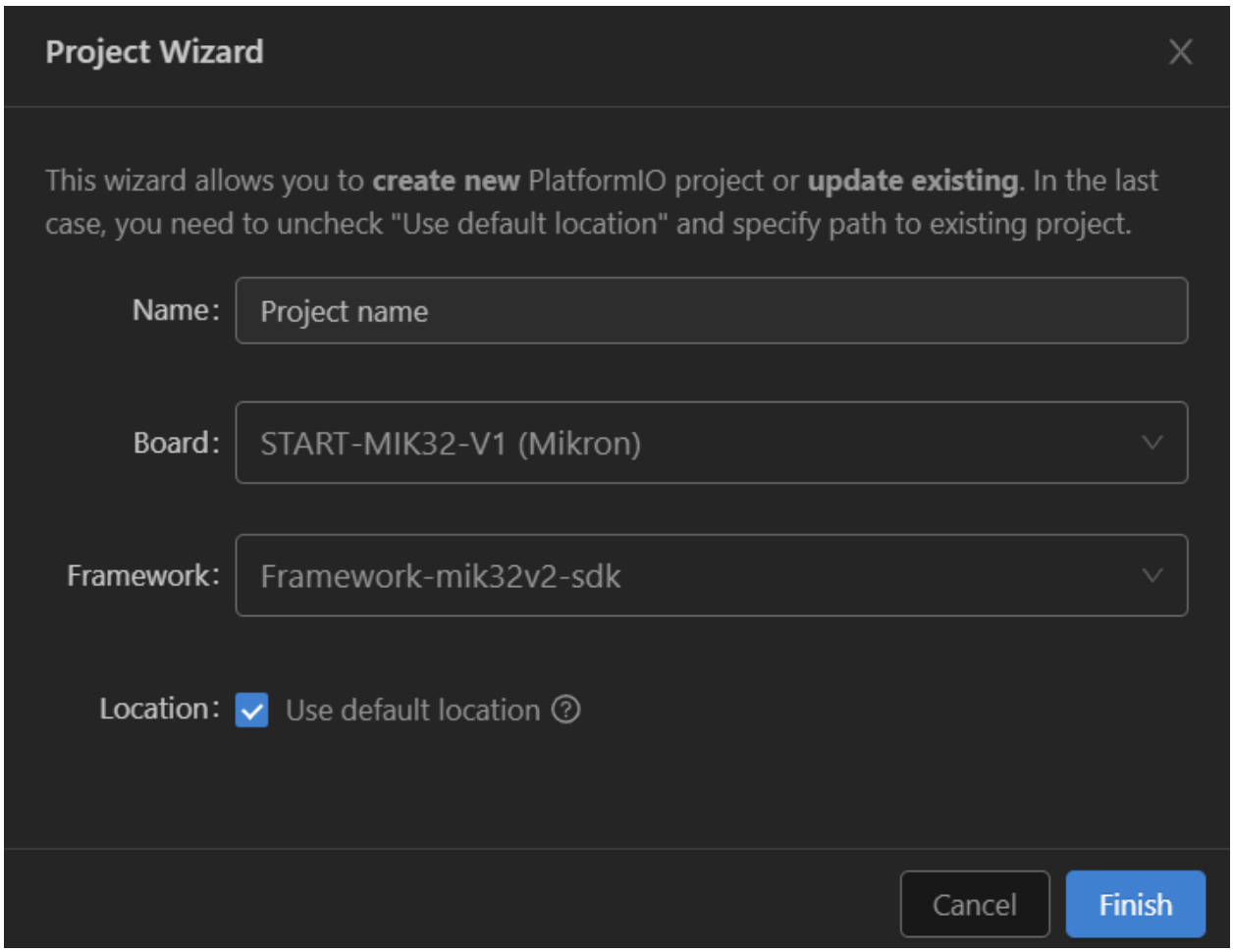


Рисунок 5 – Окно создания проекта

При первом запуске Platformio скачает набор для разработки под RISC-V. Для этого требуется подключение к интернету, будет скачано около 1 ГБ. В это время platformio скачивает такие пакеты как toolchain-riscv, tool-openocd и tool-scons в папку C:\Users\ИМЯ-ПОЛЬЗОВАТЕЛЯ\.platformio\packages. Дождитесь завершения создания проекта.

Для создания файла будущей программы в левой части панели проводника нажмите правой кнопкой мыши на папку src, выберите «Создать файл», введите название файла, например, main.c, и нажмите Enter (Рисунок 6).

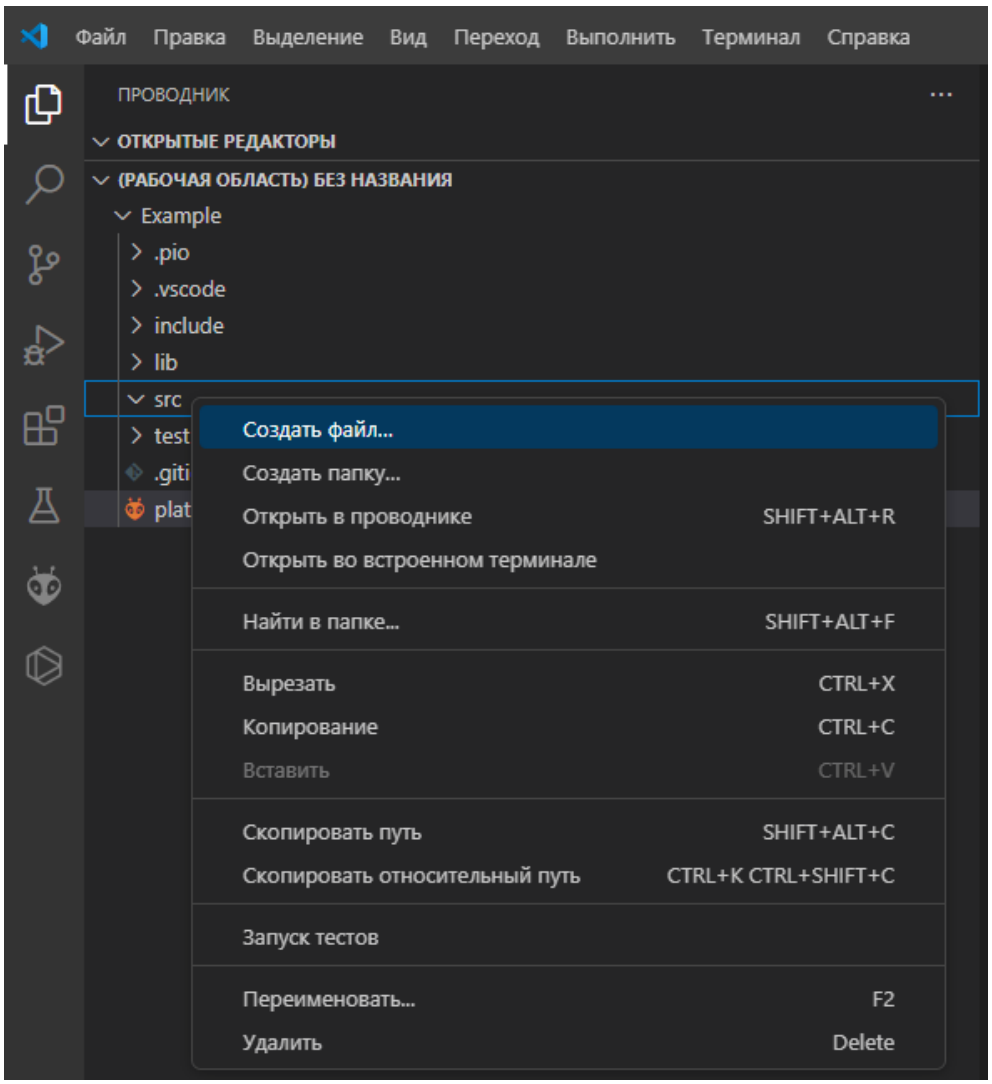


Рисунок 6 – Создание файла в проекте PlatformIO

Примеры программ для MIK32

Ознакомьтесь с примерами и параметрами конфигурационного файла platformio.ini, который находится в каждом проекте platformio.

Ссылка на примеры: https://github.com/MikronMIK32/mik32-examples.

В файле platformio.ini содержатся параметры проекта. Пример настроек сборки программы для внешней flash памяти представлен ниже.

```
; Пример комментария
[env:mik32v2]
platform = MIK32
board = start-mik32-v1
framework = framework-mik32v2-sdk
board_build.ldscript = spifi
```

Настройки platformio.ini

В корне проекта расположен файл platformio.ini который содержит настройки проекта. Основные настройки приведены ниже:

upload_protocol = start-link

board_build.ldscript - выбор ld скрипта;

Значение по умолчанию: eeprom;

Возможные значения:

- eeprom;
- spifi;
- ram.

Пример использования: board_build.ldscript = spifi

upload_speed - частота тактового сигнала интерфейса jtag, кГц.

Значение по умолчанию: 500;

Пример использования: upload_speed = 3200

board_build.f_cpu - переопределение частоты основного генератора тактовой частоты, Гц в формате длинного числа Си.

Значение по умолчанию: 32000000L;

Пример использования: board_build.f_cpu = 4000000L

board_upload.maximum_size - переопределение максимального размера программы, байты.

Размер программы вычисляется Platformio по суммарному размеру секций .text и .data. maximum_size устанавливается по объему памяти, в которой будет храниться программа.

Значение по умолчанию: 8192;

Пример использования: board_upload.maximum_size = 4194304.

Больше настроек можно найти в официальной документации проекта: https://docs.platformio.org/en/latest/projectconf/sections/env/index.html#options

Загрузка программы на плату

Перед загрузкой программы убедитесь, что проект выбран в качестве активного. Для этого в platformio toolbar, расположенном внизу, выберите "switch platformio project environment". В открывшемся выпадающем списке выберите нужный проект. Затем нажмите upload (Рисунок 7).

При загрузке в ОЗУ в случае успешной загрузке в конце будет надпись вида «downloaded xxx bytes in...».

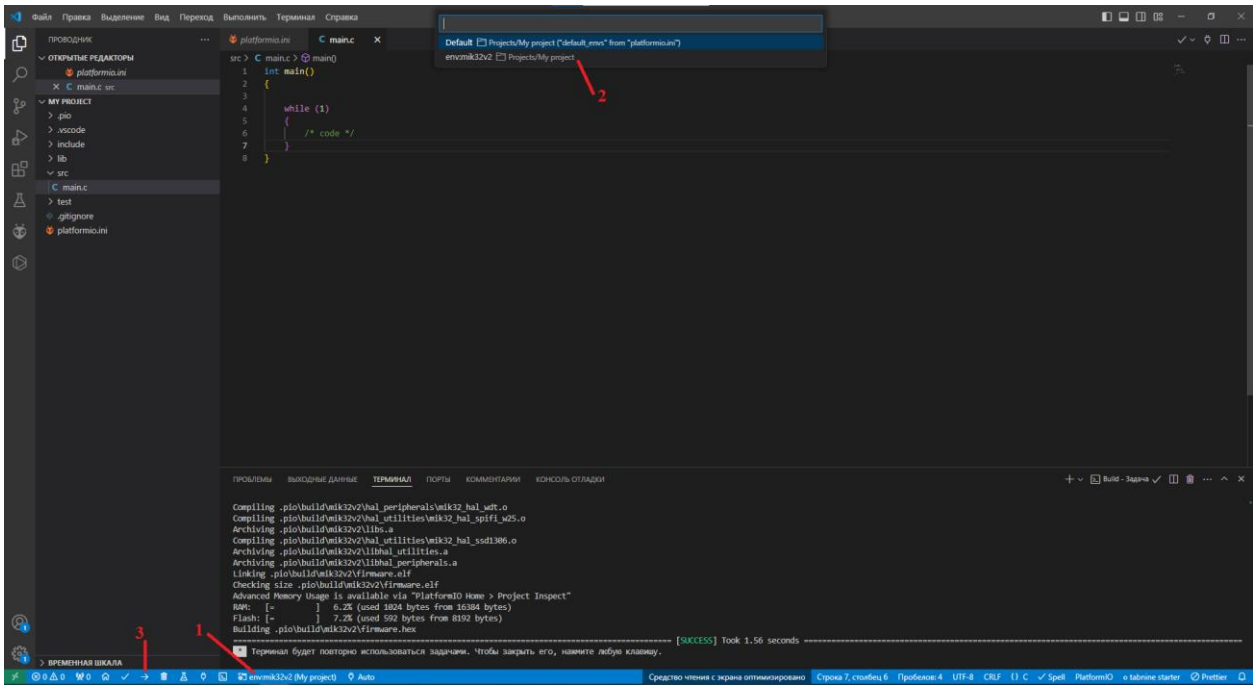


Рисунок 7 – Выбор проекта и загрузка программы в микроконтроллер

Больше информации о MIK32 на сайте https://wiki.mik32.ru

Схемотехнические элементы отладочной платы

Режим загрузки MIK32

На плате выводы BOOT (B0 и B1) определяют режим загрузки MIK32 при включении питания или при внешнем сбросе (RESET) (Рисунок 8).

Состояние B0	Состояние B1	Режим загрузки
0	0	EEPROM
0	1	SPIFI
1	0	RAM
1	1	Зарезервировано

где 0 – низкий логический уровень, 1 – высокий логический уровень.

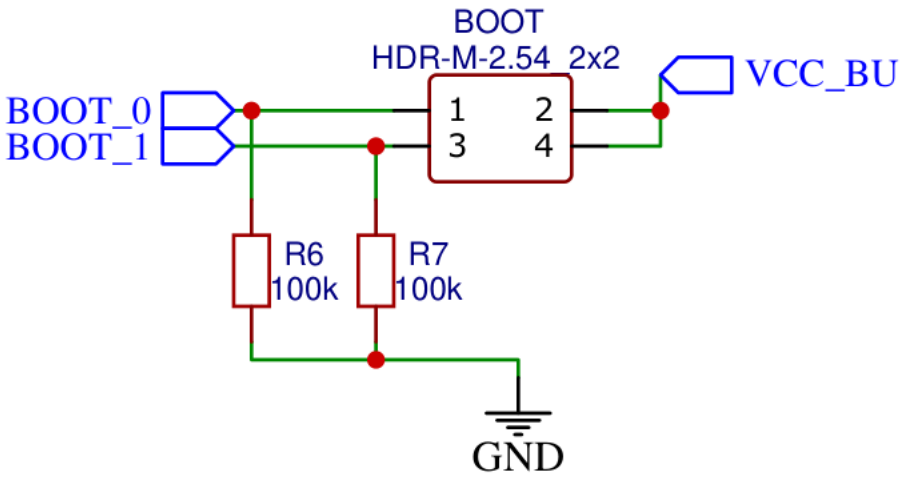


Рисунок 8 – Схема выбора режима загрузки

По умолчанию выводы B0 и B1 подтянуты к земле, что соответствует режиму загрузки из EEPROM. Для изменения режима загрузки следует установить перемычку между одним из выводов BOOT и выходом внутренней линии питания батарейного домена 3.3В - VCC_BU (Рисунок 9).

Рисунок 9 – Выбор режима загрузки перемычкой

Кнопки управления

На плате расположены три кнопки:

- Пользовательская кнопка USER_B с подтяжкой к питанию, которая подключена к выводу MIK32 PORT0_8 и замыкает его на землю;
- Кнопка EXT_WU выхода из режима пониженного энергопотребления;
- Кнопка RESET для внешнего сброса MIK32.

Схема подключения кнопок показана на Рисунок 10

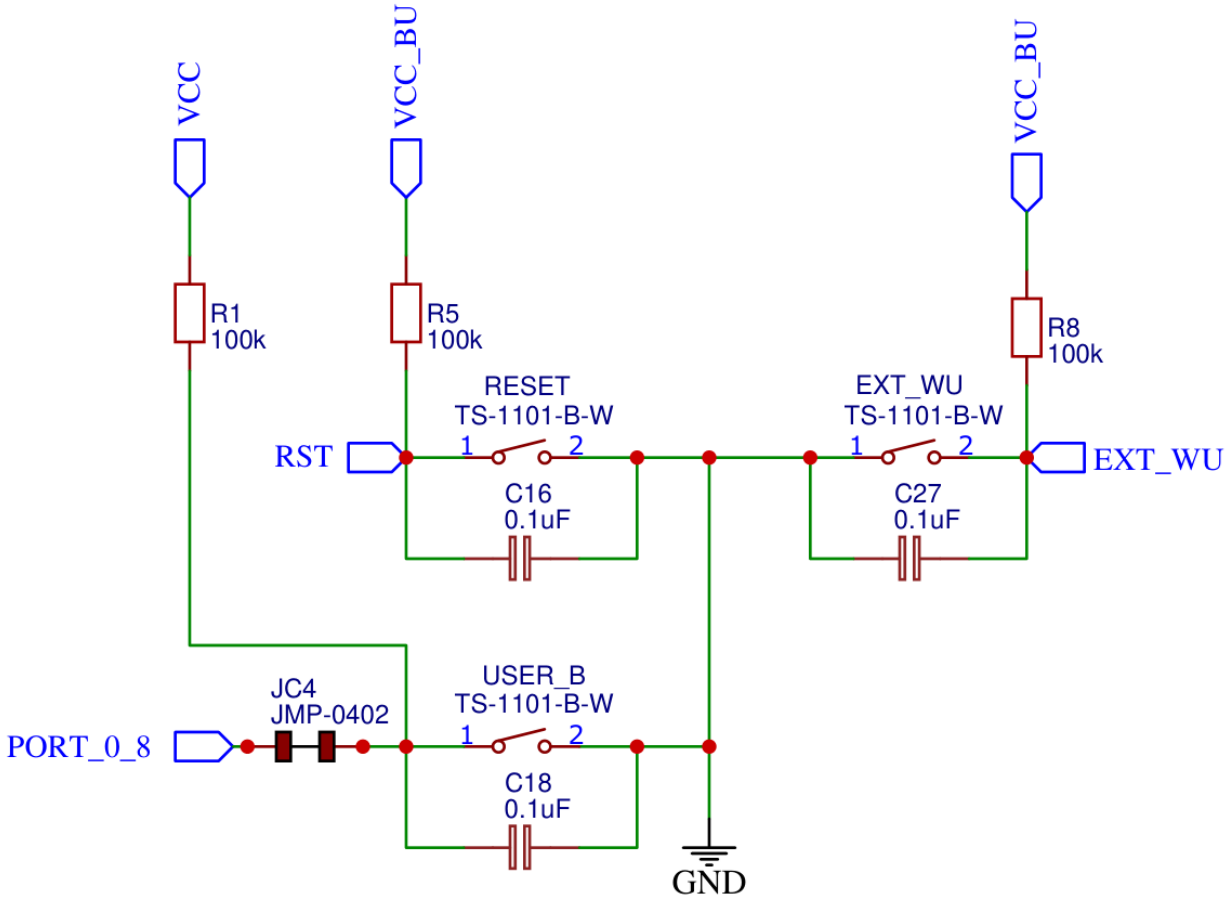


Рисунок 10 – Схема подключения кнопок программирования

Светодиодные индикаторы

Два пользовательских светодиода VD1 и VD2 подключены к выводам MIK32 PORT0_3 и PORT1_3 через перемычку и загораются при подаче на них логической «1» в режиме GPIO. Выводы PORT0_3 и PORT1_3 можно использовать в качестве вывода ШИМ сигнала, который формируется с помощью TIMER32_1 и TIMER32_2 соответственно.

Светодиод POWER (обозначение VD3) сигнализирует о наличии питания платы.

Светодиод JTAG/COM (обозначение VD5) является индикатором текущего режима встроенного программатора.

Схема подключения светодиодных индикаторов показана на Рисунок 11.

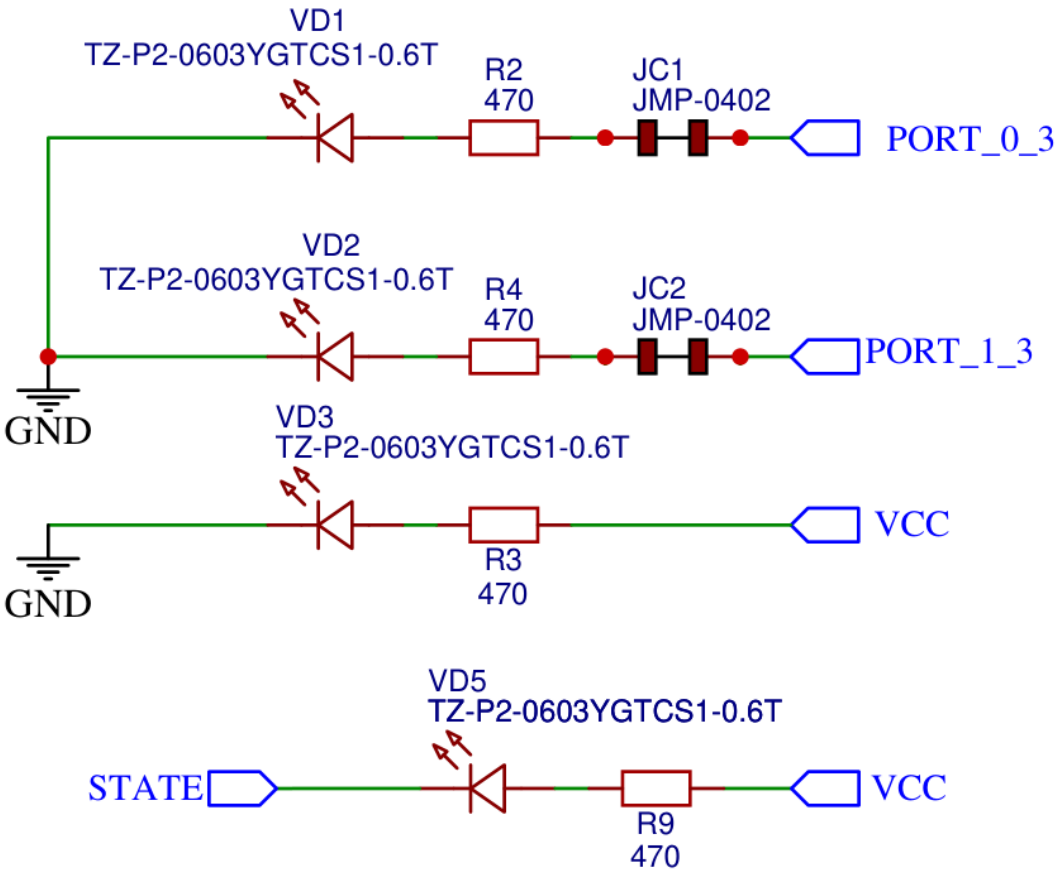


Рисунок 11 – Схема подключения светодиодных индикаторов

Питание и выводы

Напряжение питания MIK32 на выводе VCC составляет 3,3 В, допустимый диапазон (2,97…3,63) В. Вывод AVCC MIK32 - напряжение питания аналоговых блоков по умолчанию соединен через перемычку с выводом VCC. Возможно подать на вывод AVCC напряжение отличное от VCC. Для этого следует разрезать перемычку JC3 (обратная сторона платы) и подать напряжение на вывод платы AVCC величиной 3,3 В, допустимый диапазон (3,15…3,45) В (Рисунок 12).

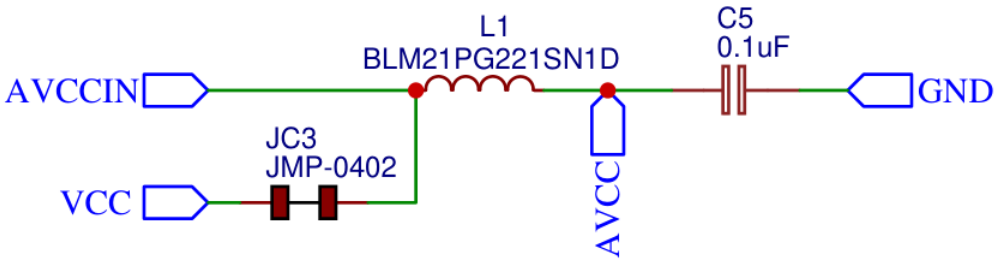


Рисунок 12 – Схема подачи питания аналоговых блоков

Имеется возможность питать плату от напряжения 5 В (вывод 5V), допустимый диапазон (4,75…5,25) В. Питание может поступать от USB разъема, который защищен от перетекания тока диодом D1 (Рисунок 13).

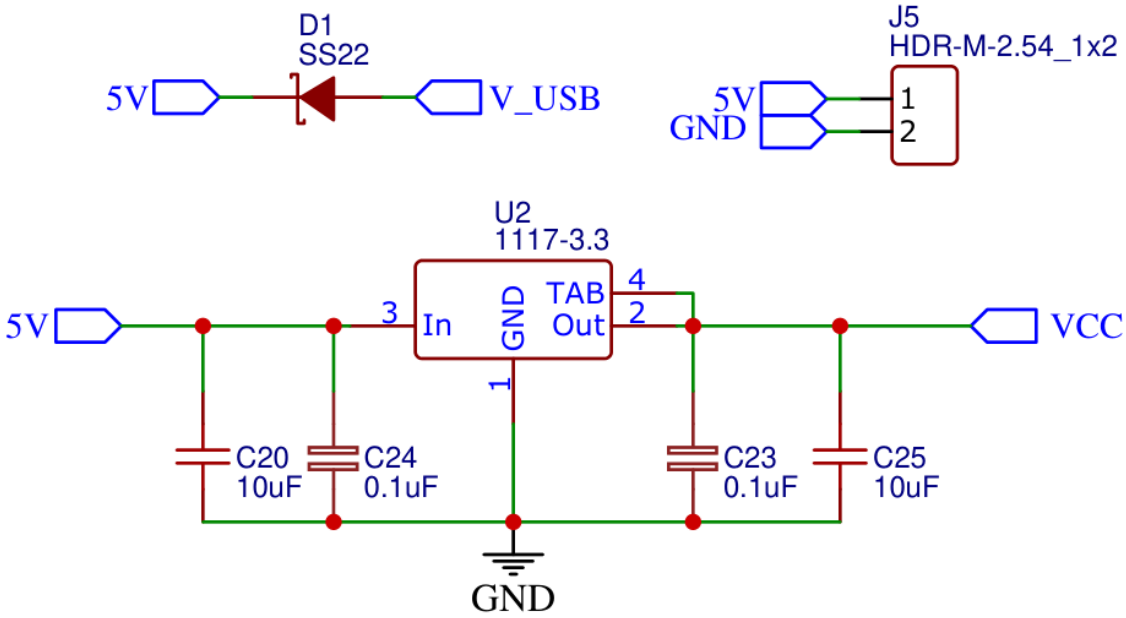


Рисунок 13 – Схема регулятора и защиты линии 5 В

Запасное питание батарейного домена на выводе VBAT предназначено для питания батарейного домена MIK32 величиной 3,3 В, допустимый диапазон (2,5…3,63) В.

Для чтения данных из однократно-программируемой памяти (OTP) MIK32 вывод VPROG на плате соединен с VCC через диодную сборку. Если требуется записать данные в память, на контактную группу J2 (вывод VPROG) следует подать напряжение программирования OTP 8.0В (Рисунок 14).

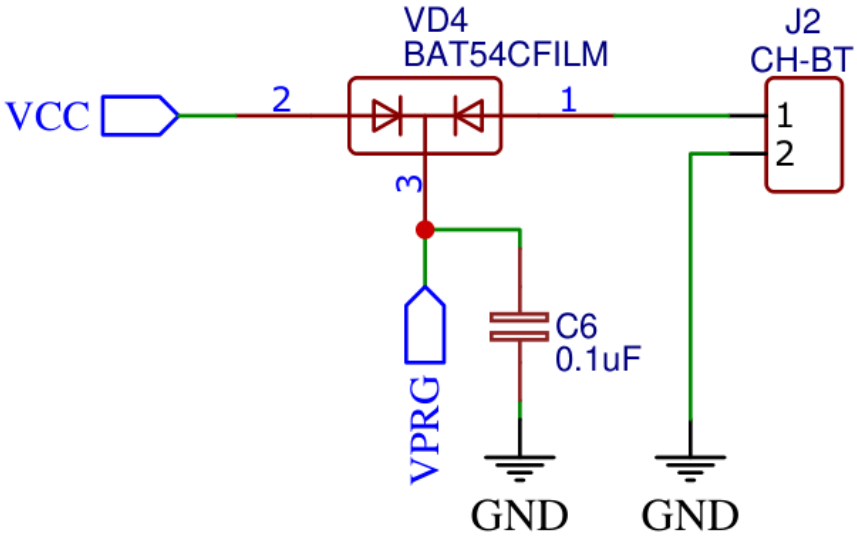
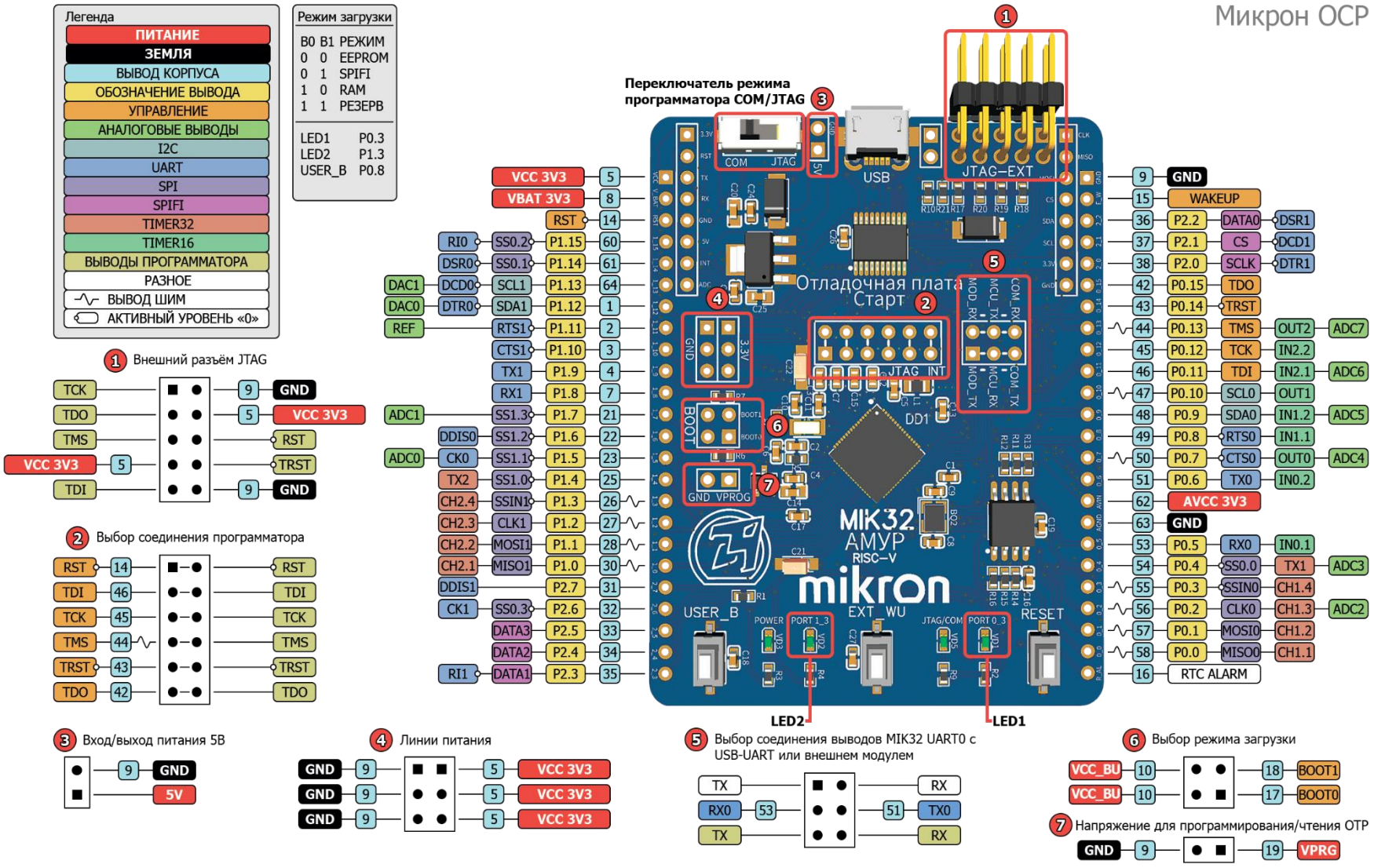


Рисунок 14 – Схема подачи питания однократно программируемой памяти

Кроме этого, на плате выведена линия RTC_ALARM (вывод R_AL), которая сигнализирует о срабатывании будильника RTC, а также линия External Wakeup (вывод E_W, кнопка EXT_WU), которая подтянута к земле для выхода из режимов низкого энергопотребления.

ПРИЛОЖЕНИЕ 1. Назначение выводов



ПРИЛОЖЕНИЕ 2. Лист регистрации изменений

Список изменений

- Ред. 1.0 – первый выпуск;
- Ред. 1.1 – добавлена информация о режиме работы USB-UART преобразователя;
- Ред. 1.2 – исправлена маркировка флеш, опечатки и рисунки;
- Ред. 1.3 – добавлено описание работы с Eclipse IDE (MIK32 IDE v1.2);
- Ред. 1.4 – дополнено описание работы с Eclipse IDE (MIK32 IDE v1.2.2).