Research Laboratory of Design Automation
НИЛ автоматизации проектирования
Сделано в России
Тел.: +7 (495) 26-66-700 info@reallab.ru • https://www.reallab.ru

Среда разработки Arduino IDE

Работа с ПЛК и модулями RealLab! в среде разработки Arduino IDE: установка, настройка и запуск проектов

Программирование ПЛК RealLab линейки MC-ST в среде разработки Arduino IDE

Руководство пользователя

Версия от 7 ноября 2025 г.

Одной проблемой стало меньше!

Уважаемый покупатель!

Научно-исследовательская лаборатория автоматизации проектирования (НИЛ АП) благодарит Вас за использование нашей продукции и просит сообщать нам свои пожелания по улучшению программного обеспечения. Ваши пожелания можно направлять по почтовому или электронному адресу, а также сообщать по телефону:

НИЛ АП, пер. Биржевой спуск, 8, Таганрог, 347900,

Тел. (495) 26-66-700,

e-mail: info@reallab.ru, https://www.reallab.ru.

Вы можете также получить консультации по применению нашей продукции, воспользовавшись указанными выше координатами.

Оглавление

1. Вводная часть 4

1.1. Начало работы 4

1.2. Программируемые контроллеры Reallab! с Arduino IDE 5

2. Установка и настройка Arduino IDE 5

2.1. Системные требования 5

2.2. Установка Arduino IDE и подключение библиотек 5

3. Программирование ПЛК линейки MC-ST 8

3.1. Компиляция и запись программы 8

3.2. Работа с кодом 9

3.3. Установка библиотек 10

4. Примеры работы со скетчами 11

4.1. Inputs – Дискретные входы 12

4.2. Outputs – Дискретные выходы 12

4.3. Rtc – Часы реального времени 12

4.4. Temp_ext – Встроенный в плату датчик температуры 12

4.5. Temp_int – Внутренний датчик температуры микроконтроллера 13

4.6. 1wire – Внешний датчик температуры DS18B20 13

4.7. Lcd_and_buttons – Дисплей и кнопки 13

4.8. Analog_amperage – Аналоговые входы (сила тока) 13

4.9. Analog_voltage – Аналоговые входы (напряжение) 14

4.10. Rs485 – Обработка команд по RS-485 14

Лист регистрации изменений 15

1. Вводная часть

Добро пожаловать в руководство по программированию ПЛК Reallab! линейки MC-ST. Данное руководство предназначено как для начинающих, так и для опытных пользователей.

Для работы с ПЛК необходимы базовые навыки программирования микроконтроллеров на базе архитектуры ARM-Cortex-M.

Программирование ПЛК линейки MC-ST в руководстве осуществляется в среде Arduino IDE, с использованием специализированного проекта STMduino. В ПЛК используются микроконтроллеры STM32 серии F4.

Настоящее руководство демонстрирует процесс установки среды программирования Arduino IDE, подключение ПЛК линейки MC-ST, добавление поддержки работы с STM32, примеры скетчей с их описанием.

1.1. Начало работы

Особенность программирования ПЛК линейки MC-ST в данной среде заключается в использовании ядра STM32duino, которое позволяет программировать платы на базе STM32, как если бы они были платами Arduino. Это один из методов работы с STM32, о нюансах такого подхода к программированию можно ознакомиться в открытых источниках.

Написанная в среде Arduino IDE программа называется «скетч». Сама программа пишется в текстовом редакторе (рис. 1.1). В области сообщений выводятся ошибки и пояснения.

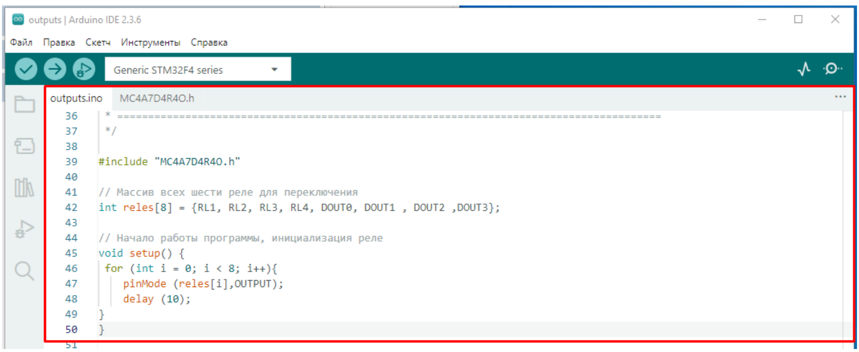


Рис. 1.1. Текстовый редактор

В консоли отображается полный отчёт о компиляции программы, различные ошибки, подсказки и другая полезная для разработчика информация.

Панель инструментов (рис. 1.2) представляет собой набор полезных функций при разработке программы. На панели с часто используемыми командами расположены основные кнопки при разработке программы.

Рис. 1.2 Панель инструментов

1.2. Программируемые контроллеры Reallab! с Arduino IDE

Настоящее руководство посвящено программированию в среде Arduino IDE следующих ПЛК Reallab!: MC-4A7D4R4O-ST, MC-8D2R-ST, MC-8D2S-ST, MC-12D4R4O-ST, MC-12D6R-ST, MC-12D8O-ST.

2. Установка и настройка Arduino IDE

2.1. Системные требования

Перед установкой и использованием Arduino IDE рекомендуется ознакомиться с системными требованиями, указанными на официальном веб-сайте производителя по адресу: https://www.arduino.cc.

Установка программного обеспечения должна выполняться на устройстве, соответствующем указанным минимальным или рекомендованным техническим характеристикам.

2.2. Установка Arduino IDE и подключение библиотек

В качестве примера используется Arduino IDE версии 2.3.6.

Ниже приведена пошаговая инструкция по установке среды и по её первоначальной настройке для работы с примерами.

1. Установите Arduino IDE с официального сайта проекта Arduino (https://www.arduino.cc/en/software).

2. После установки запустите приложение.

3. Откройте меню «Файл → Параметры» (рис. 2.1), далее нажмите на иконку «Дополнительные ссылки для Менеджера плат» (рис. 2.2).

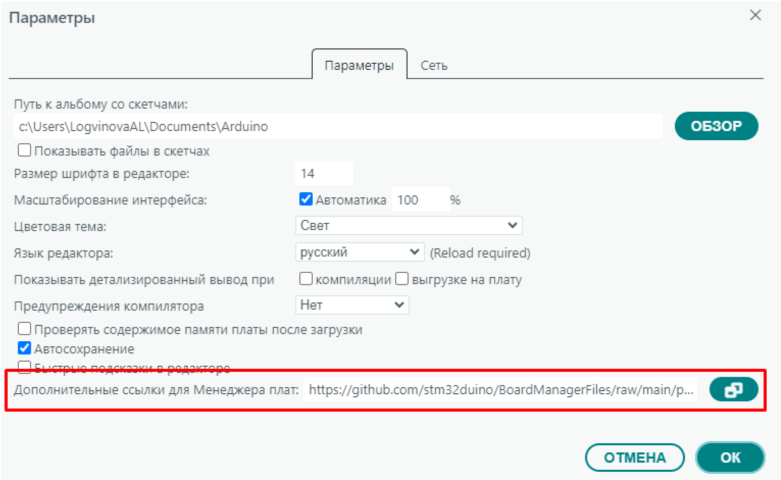


Рис. 2.1 Меню «Файл → Параметры»

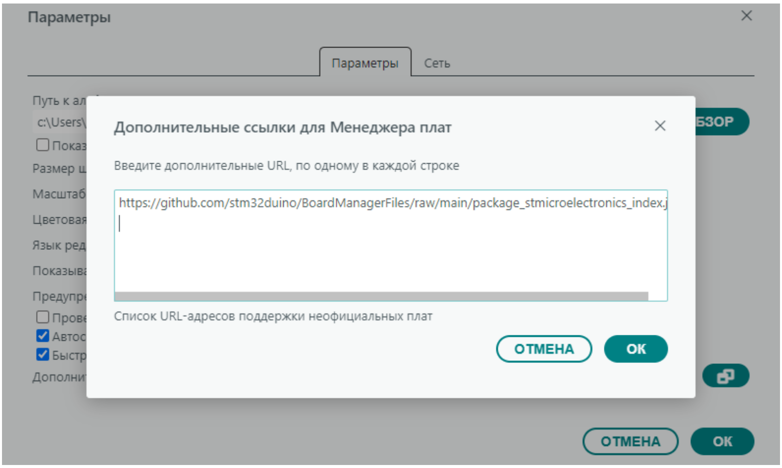


Рис. 2.2 Иконка «Дополнительные ссылки для Менеджера плат»

4. Добавьте в поле ссылку, приведённую ниже, и нажмите «ОК» для сохранения настроек.

https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json

5. Перезагрузите Arduino IDE.

6. Перейдите в меню «Инструменты → Плата → STM32 MCU based boards → Generic STM32F4 series». Если в подменю не появились платы «STM32 MCU based boards» (рис. 2.3), тогда следует открыть «Менеджер плат» и в строке поиска ввести «STM32» (рис. 2.4).

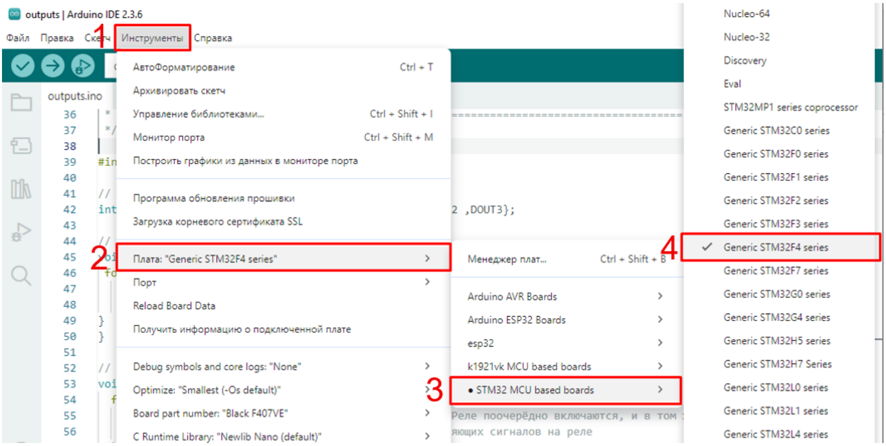


Рис. 2.3 Выбор платы

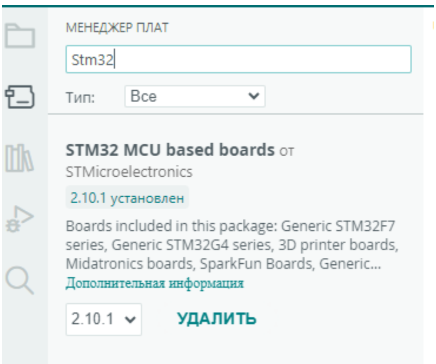


Рис. 2.4 Строка поиска менеджера плат

7. Далее установите или обновите скрипты для плат STM32 (подгружаются автоматически при выборе/установке).

8. Для управления работой пинов (выводами устройства) в коде, рекомендуется использовать заголовочные файлы имеющихся модулей, которые находятся в папках с проектами примеров (например, MC8D2S.h).

3. Программирование ПЛК линейки MC-ST

3.1. Компиляция и запись программы

Перед компиляцией следует задать несколько параметров в меню «Инструменты».

1. Убедитесь, что на вкладке «Плата» выбрана серия «Generic STM32F4 series» (рис. 2.3).

2. В «Board part number» следует выбрать Generic F427ZGTx (рис. 3.1).

3. Укажите доступный порт, с которым собираетесь работать. Метод загрузки (Upload method) следует выбрать – «STM32CubeProgrammer Serial».

4. Для компиляции нажмите кнопку «Проверить»  (или комбинацию клавиш Ctrl+R). После успешной компиляции можно перейти к следующему шагу.

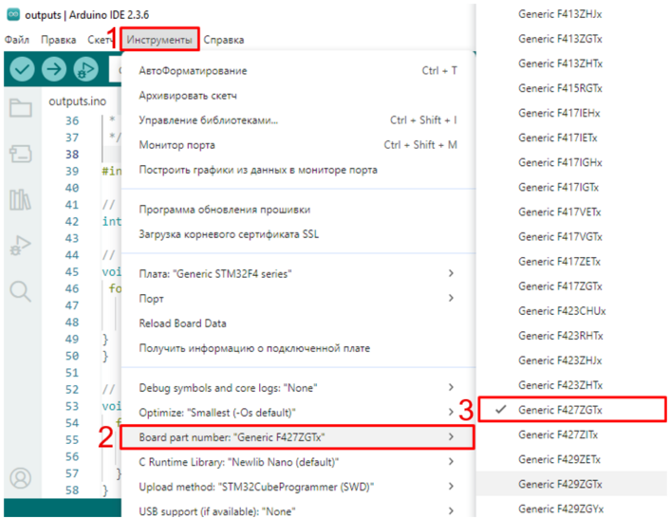


Рис. 3.1 Выбор Board part number

5. Чтобы записать прошивку в модуль необходимо использовать преобразователь интерфейса RS-485 ⇄ USB (например, NLS-485-USB).

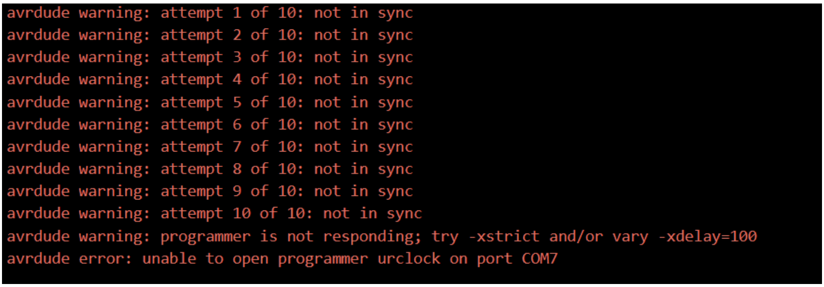
5.1. Для записи программы в ПЛК необходимо перевести его в режим загрузчика, соединив (перед этим отключив питание!), вывод «Init» с «GND», а преобразователь RS-485 ⇄ USB с клеммами «Data1+» и «Data1-».

5.2. Включите питание ПЛК. Если режим загрузчика активирован, засветится красный светодиод. Далее нажмите кнопку «Загрузить на плату»  (или сочетание клавиш Crtl+U).

5.3. При успешной загрузке вы увидите надпись «Done uploading» (рис. 3.2). После этого выключите ПЛК.

Рис. 3.2 Успешная загрузка

Если среда выдаёт ошибку «unable to open programmer urclock on port COM» (рис. 3.3), проверьте соединение между клеммами «Init» и «GND», убедитесь, что выбраны корректный порт и программатор.



```
avrdude warning: attempt 1 of 10: not in sync
avrdude warning: attempt 2 of 10: not in sync
avrdude warning: attempt 3 of 10: not in sync
avrdude warning: attempt 4 of 10: not in sync
avrdude warning: attempt 5 of 10: not in sync
avrdude warning: attempt 6 of 10: not in sync
avrdude warning: attempt 7 of 10: not in sync
avrdude warning: attempt 8 of 10: not in sync
avrdude warning: attempt 9 of 10: not in sync
avrdude warning: attempt 10 of 10: not in sync
avrdude warning: programmer is not responding; try -xstrict and/or vary -xdelay=100
avrdude error: unable to open programmer urclock on port COM7
```

Рис. 3.3 Ошибка

5.4. Отключите клемму «Init» от «GND», включите ПЛК.

Устройство готово к эксплуатации.

3.2. Работа с кодом

При программировании контроллера необходимо подключить заголовочный файл (с расширением .h), который содержит константы – номера пинов, определенные с помощью директив #define. Заголовочный файл включается директивой include следующим образом:

#include "MC12D6R.h".

При этом файл должен находиться в той же папке, что и файл скетча с расширением .ino.

Сборка скетча может требовать наличия установленных библиотек. Если они не были установлены, компилятор сообщит об ошибке и выведет названия недостающих пакетов. Их можно установить через встроенный в среду Arduino IDE менеджер библиотек (рис. 3.4).

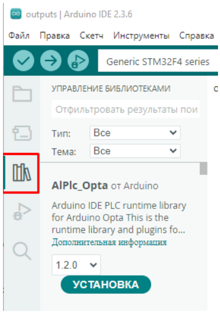


Рис. 3.4 Arduino IDE менеджер

Точные названия библиотек можно узнать из директив include в начале каждого скетча (рис. 3.5).

```
#include "MC12D4R40.h"
#include "HardwareSerial.h"
#include <Arduino.h>
#include <string.h>
#include <stdio.h>
#include <DS18B20.h> // Библиотека для работы с внешним датчиком температуры
```

Рис. 3.5 Названия библиотек в коде

3.3. Установка библиотек

Чтобы установить библиотеку, введите ее название в поиске (рис. 3.6) и нажмите кнопку «Установка». По умолчанию пакеты сохраняются в папку %username%\Documents\Arduino\libraries.

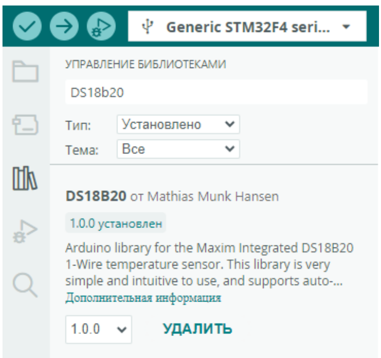


Рис. 3.6 Установка библиотеки

После этого вы сможете собрать и запустить программу.

Список библиотек, установка которых необходима для корректной работы примеров представлен ниже:

• LiquidCrystal от Arduino, Adafruit v.1.0.7 - для работы дисплея.

• DS18B20 от Mathias Munk Hansen v.1.0.0 – для работы внешнего датчика температуры.

• OneWire от Jim Studt, Tom Pollard v2.3.8 – для библиотеки DS18B20.

• STM32duino RTC от STMicroelectronics v.1.8.0 для примера с RTC (перед установкой следует сперва установить ядро STM32duino п.2.2.).

4. Примеры работы со скетчами

Для удобства разработан набор примеров скетчей (с расширением .ino) для ПЛК. Каждому компоненту периферии модуля соответствует один из скетчей-примеров, унифицированных под модификации конкретного ПЛК. В комментариях (шапке) каждого скетча приведено краткое описание примера, его особенности, необходимая информация по работе с ним. Перечень примеров представлен в табл. 4.1.

Табл. 4.1 Перечень разработанных примеров для ПЛК

Название	Описание
inputs	Опрос состояний дискретных входов
outputs	Управление дискретными выходами
rtc	Работа с часами реального времени
temp_ext	Чтение данных со встроенного в плату датчика температуры DS18B20Z
1wire	Подключение внешнего датчика температуры DS18B20 по трёхпроводной шине 1wire
temp_int	Чтение данных с интегрированного в микроконтроллер датчика температуры
lcd_and_buttons	Управление ЖК-дисплеем и кнопками
analog_amperage	Измерение силы тока (MC-4A7D4R4O-ST)
analog_voltage	Измерение напряжения (MC-4A7D4R4O-ST)
rs485	Обработка команд по RS-485

4.1. Inputs – Дискретные входы

inputs.ino

Скетч считывает состояния всех дискретных входов (например, Din0 – Din11, в зависимости от модуля) и выводит их в виде строки, состоящей из «0» и «1», через RS-485. В setup() пины настраиваются как входы, в loop() их состояние считывается с помощью функции digitalRead() и формируется строка для вывода в монитор порта.

Примечание: в модулях MC-8D2S-ST и MC-8D2R-ST имеются 4 дискретных входа-выхода IN_OUT. В примере они настроены и используются как обычные цифровые входы. Работа с ними выполняется в соответствии с руководством по эксплуатации.

4.2. Outputs – Дискретные выходы

outputs.ino

Пример реализует поочерёдное переключение всех имеющихся дискретных выходов (типа «Открытый сток» и/или релейных) модуля в бесконечном цикле. В setup() соответствующие пины настраиваются как выходы. В loop() каждое реле изменяет своё состояние (замыкается, размыкается) с задержкой 3 секунды между переключениями.

Примечание: в модулях MC-8D2S-ST и MC-8D2R-ST имеются 4 дискретных входа-выхода IN_OUT. В примере они настроены и используются как обычные цифровые выходы. Работа с ними выполняется в соответствии с руководством по эксплуатации.

4.3. Rtc – Часы реального времени

rtc.ino

Пример демонстрирует работу со встроенным в микроконтроллер модулем часов реального времени. В setup() инициализируется RTC, а также устанавливается время с помощью функций settime и setdate. В основном цикле программа выводит на последовательный порт дату и время в формате: «DD-MM-YY, hh:mm:ss:ms».

4.4. Temp_ext – Встроенный в плату датчик температуры

temp_ext.ino

В скетче реализован вывод значения температуры. За измерение температуры отвечает встроенный в плату датчик DS18B20Z. При инициализации указывается пин датчика, определенный в заголовочном файле ПЛК константой TERMO. Циклически программа выводит на последовательный порт актуальное значение температуры.

4.5. Temp_int – Внутренний датчик температуры микроконтроллера

temp_int.ino

В скетче реализовано измерение температуры с использованием внутреннего датчика микроконтроллера. За преобразование аналогового сигнала в значение температуры отвечает расчётная формула, учитывающая опорное напряжение и параметры АЦП. В цикле программа отправляет на последовательный порт актуальные значения температуры.

4.6. 1wire – Внешний датчик температуры DS18B20

1wire.ino

Пример предназначен для работы с модулями MC-8D2S-ST и MC-8D2R-ST. В скетче реализована проверка трёхпроводной шины 1-Wire с использованием внешнего датчика температуры DS18B20 (схему подключения см. в руководстве по эксплуатации). Модуль циклически опрашивает датчик по шине 1-Wire и передаёт по RS-485 строку с температурой и названием модели датчика. Пример сообщения: Te= 30,21 C».

4.7. Lcd_and_buttons – Дисплей и кнопки

lcd_and_buttons.ino

Программа взаимодействует с ЖК-дисплеем и кнопками SA4-SA7 контроллера. При запуске на дисплее посимвольно отображается приветственное слово - «Reallab!». При нажатии на одну из кнопок происходит мигание заданным сообщением, соответствующим каждой из них. Реализована простейшая антидребезговая обработка сигналов с кнопок.

4.8. Analog_amperage – Аналоговые входы (сила тока)

analog_amperage.ino

Пример для модуля MC-4A7D4R4O-ST. Программа считывает сигналы силы тока с 4-х аналоговых входов контроллера, настроенных в режим приёмника токового сигнала. Полученные значения преобразуются в физическое значение тока в амперах и передаются по RS-485. На экран последовательного монитора выводится номер входа и рассчитанное значение тока с тремя знаками после запятой. Обновление данных происходит каждые 3 секунды.

Перед запуском примера убедитесь, что движковые переключатели находятся в положении «ON» - подробнее в руководстве по эксплуатации (п.5.2 Органы управления ПЛК).

4.9. Analog_voltage – Аналоговые входы (напряжение)

analog_voltage.ino

Пример для модуля MC-4A7D4R4O-ST. Программа считывает аналоговые сигналы с 4-х входов контроллера и передаёт результаты измерений напряжения по RS-485. Значения отображаются в виде текстового сообщения, включающего номер входа и соответствующее напряжение в вольтах. Данные отправляются каждые 3 секунды. Пример демонстрирует базовую работу с аналоговыми входами и последовательной коммуникацией.

Перед запуском примера убедитесь, что движковые переключатели находятся в положении «OFF» - подробнее в руководстве по эксплуатации (п.5.2 Органы управления ПЛК).

4.10. Rs485 – Обработка команд по RS-485

rs485.ino

Пример демонстрирует приём и обработку команд по интерфейсу RS-485 для управления компонентами модуля (выходами). Команды в виде двухзначных чисел (например, 02) отправляются пользователем на последовательный порт и интерпретируются программой, как запрос на переключение состояния соответствующего канала (релейного/дискретного выхода).

Лист регистрации изменений

Дата изменения	Описание изменения	Примечание