



ЦИФРОВЫЕ
РЕШЕНИЯ

ALTA IDE (версия 1.0)

ALTA

Быстрый старт

06.2026
версия 1.8

Содержание

1 Быстрый старт	3
1.1 Устанавливаем ALTA IDE	4
1.2 Создаем проект	6
1.3 Пишем программу.....	7
1.4 Создаем задачу	9
1.5 Настраиваем входы и выходы.....	12
1.6 Подключаем и настраиваем ПЛК	15
1.7 Сборка и загрузка проекта	18
1.8 Режим онлайн.....	19

1 Быстрый старт

Цель документа

Данное руководство предназначено для первого знакомства со средой ALTA IDE. Документ **Быстрый старт** создан для:

- **Ускорения адаптации** — позволяет быстро освоить основные функции и возможности, избегая изучения Руководства Пользователя.



ПРИМЕЧАНИЕ

Для углубленного понимания принципов работы ALTA IDE и получения ответов на возникающие вопросы следует обратиться к Руководству пользователя, доступному по нажатию клавиши **F1** внутри среды ALTA IDE.

- **Начальной настройки** — содержит инструкции по установке, настройке и подключению устройств.
- **Получения практического опыта** — пользователь может сразу перейти к практике, написав и протестировав первую программу на основе предоставленных примеров.

Информация в документе представлена в графическом виде с минимальными текстовыми пояснениями. **Быстрый старт** предназначен для пользователей обладающих базовыми навыками работы с ПЛК и минимальным опытом программирования на языке ST, а также имеющих доступ к ПЛК для выполнения практических шагов, приведенных в данном документе.

Для успешного старта следует выполнить действия:

- [Установить ALTA IDE.](#)
- [Создать проект.](#)
- [Написать программу.](#)
- [Создать задачу.](#)
- [Настроить входы и выходы.](#)
- [Подключить и настроить ПЛК.](#)
- [Выполнить сборку и записать проект на ПЛК.](#)
- [Перейти в онлайн и проверить работу программы.](#)

1.1 Устанавливаем ALTA IDE



ВНИМАНИЕ
В данный момент ALTA IDE работает только в ОС Windows

1

Скачайте на ПК и запустите файл *AltaSetup.exe*. Существует вероятность, что система безопасности Windows остановит выполнение приложения:

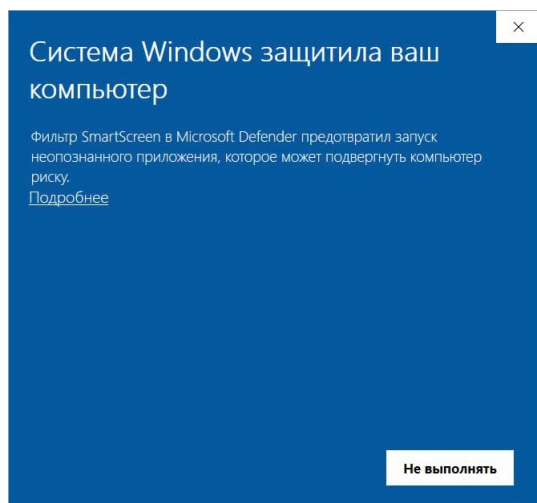


Рисунок 1.1

Нажмите **Подробнее**. Если подобное сообщение не отображается, то переходите к **шагу 3**

2

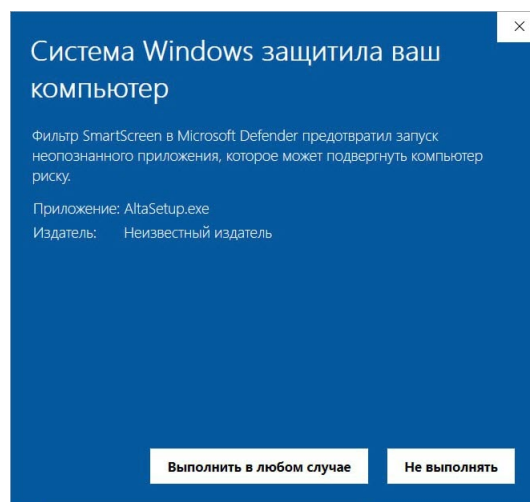


Рисунок 1.2

Выберите **Выполнить в любом случае**. Начнется установка ALTA IDE

3

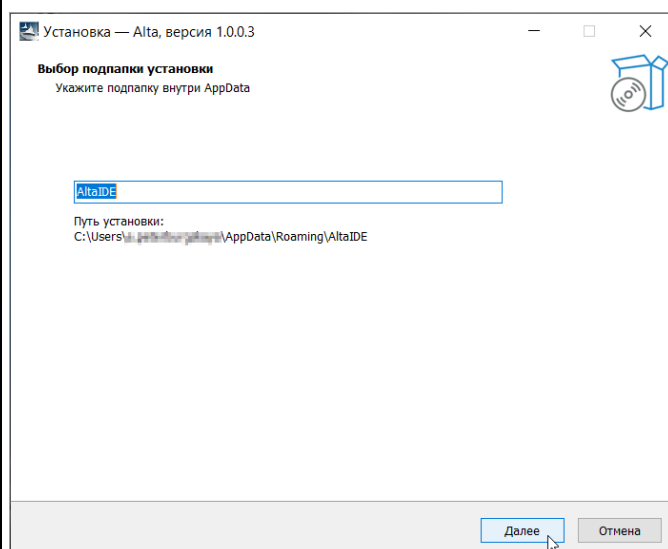


Рисунок 1.3

Выберите папку для установки ALTA IDE. Нажмите кнопку **Далее**

4

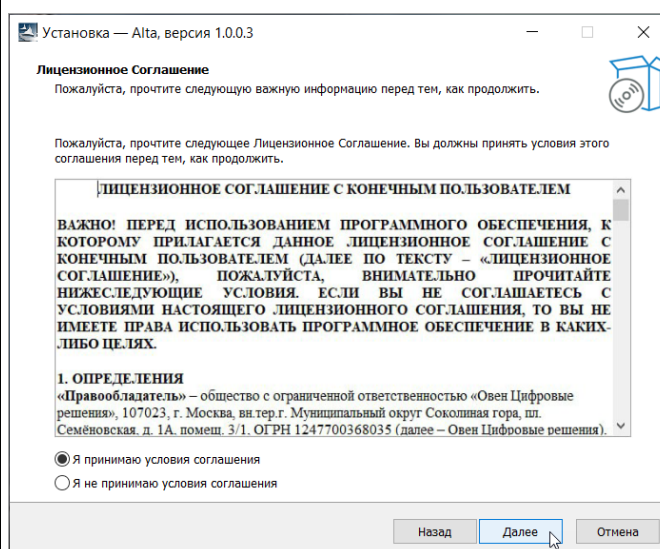


Рисунок 1.4

Ознакомьтесь с лицензионным соглашением, выберите **Я принимаю условия соглашения**. Нажмите кнопку **Далее**

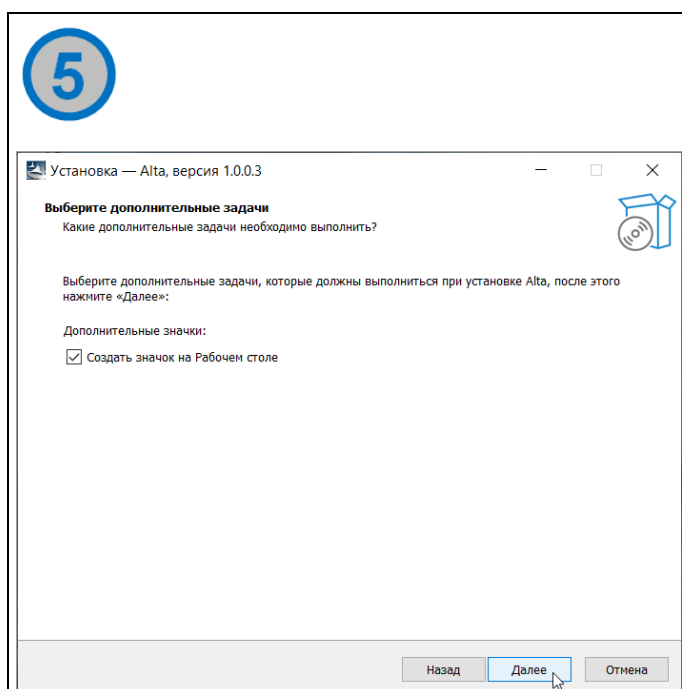


Рисунок 1.5

Установите галочку для создания ярлыка на рабочем столе. Нажмите кнопку **Далее**

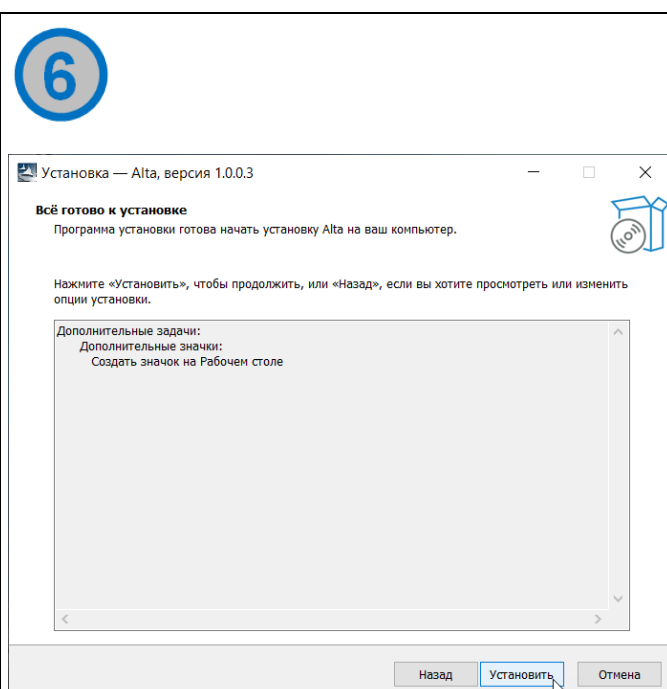


Рисунок 1.6

Ознакомьтесь с информацией об установке и нажмите кнопку **Установить**

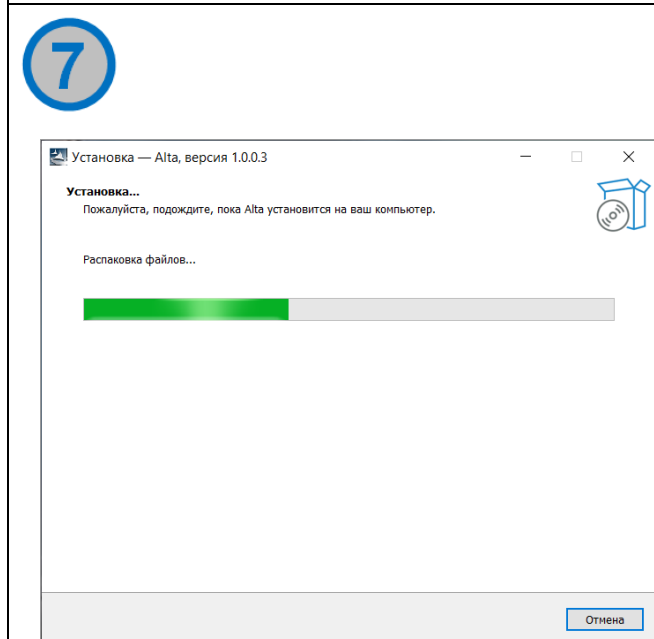


Рисунок 1.7

Дождитесь окончания установки

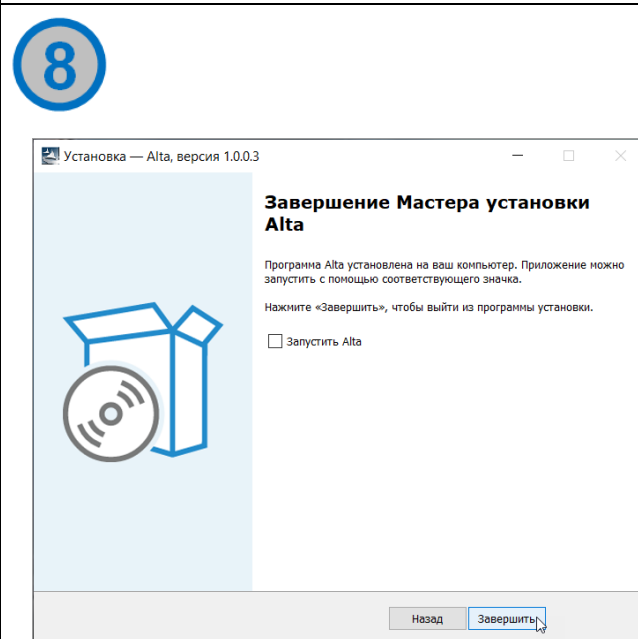


Рисунок 1.8

Установите чекбокс **Запустить Alta**. Нажмите кнопку **Завершить**

4

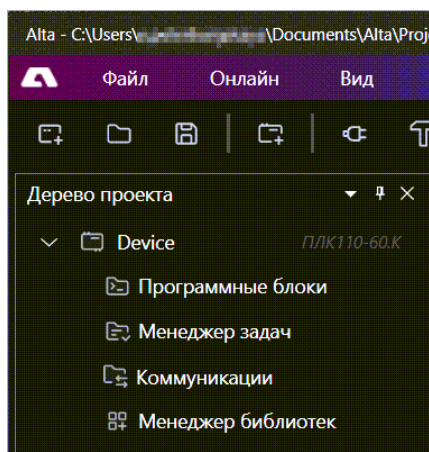


Рисунок 1.13

Проект создан - в главном окне ALTA IDE отобразилось **Дерево проекта**

1.3 Пишем программу

Пришло время создать и написать программу:

1

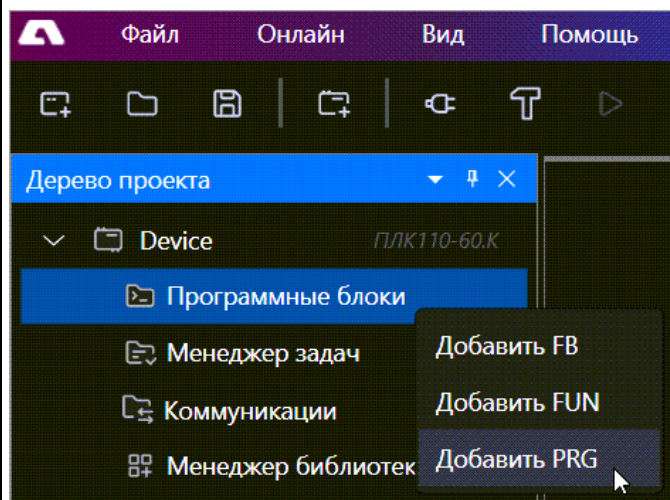


Рисунок 1.14

Нажмите ПКМ (правой кнопкой мыши) на системную папку **Программные блоки** и выберите **Добавить PRG**

2

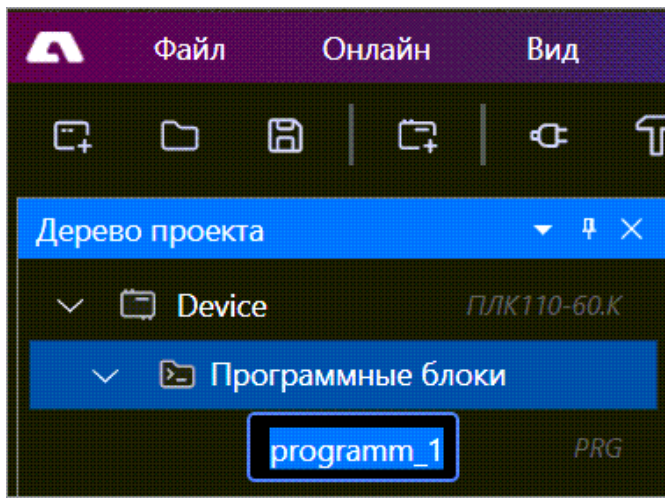


Рисунок 1.15

Вы можете присвоить программе любое имя, в рамках правил именования, указанных в Руководстве Пользователя. Мы будем пользоваться именем по умолчанию. Нажмите **Enter**

3

```

programm_1 X
Программа: programm_1
2  VAR
3  END_VAR
4  END_PROGRAM

```

Рисунок 1.16

Откроется вкладка редактора ST в которой мы будем писать программу. Напишем программу, по условиям которой при подаче сигнала на вход прибора и достижении определенного значения переменной будет замыкаться выход прибора

4

```

programm_1* X
Программа: programm_1
2  (*Объявление переменных programm_1*)
3  VAR
4  dwBitmaskOfInputs: DWORD; (*Битовая маска входов ПЛК*)
5  dwBitmaskOfOutputs: DWORD; (*Битовая маска выходов ПЛК*)
6  rValue: REAL; (*Текущее значение параметра*)
7  xEnable: BOOL; (*Условие запуска алгоритма*)
8  END_VAR
9  (*Начало программы*)
10 xEnable:=dwBitmaskOfInputs.0; (*Условие запуска алгоритма происходит по срабатыванию DI1 ПЛК*)
11 (*Алгоритм - если текущее значение больше или равно 150 - срабатывает DO1 ПЛК*)
12 IF xEnable AND (rValue>=150.0) THEN
13   dwBitmaskOfOutputs.0:=TRUE; (*DO1 ПЛК замыкается при превышении значения*)
14 ELSE
15   dwBitmaskOfOutputs.0:=FALSE; (*DO1 ПЛК выключен*)
16 END_IF
17 END_PROGRAM (*Объявление конца программы*)

```

Рисунок 1.17

Нередактируемая строка *Программа: programm_1* в верхней части редактора ST является объявлением программы. Дополнительно объявлять программу в коде не требуется.

Объявим переменные и напишем алгоритм. Для удобства текст программы размещен ниже.

**ПРИМЕЧАНИЕ**

Подробно синтаксис языка ST описан в Руководстве Пользователя ALTA IDE

5

```

(*Объявление переменных programm_1*)
VAR
  dwBitmaskOfInputs:      DWORD;  (*Битовая маска входов ПЛК*)
  dwBitmaskOfOutputs:    DWORD;  (*Битовая маска выходов ПЛК*)
  rValue:                 REAL;   (*Текущее значение параметра*)
  xEnable:                BOOL;   (*Условие запуска алгоритма*)
END_VAR

(*Начало программы*)
xEnable:=dwBitmaskOfInputs.0;    (*Условие запуска алгоритма происходит
по срабатыванию DI1 ПЛК*)

(*Алгоритм - если текущее значение больше
или равно 150 - срабатывает DO1 ПЛК*)
IF xEnable AND (rValue>=150.0) THEN
  dwBitmaskOfOutputs.0:=TRUE;   (*DO1 ПЛК замыкается при превышении значения*)
ELSE
  dwBitmaskOfOutputs.0:=FALSE;  (*DO1 ПЛК выключен*)
END_IF
END_PROGRAM                    (*Объявление конца программы*)

```

1.4 Создаем задачу

Программа написана, создадим задачу, в которой она будет выполняться:

1

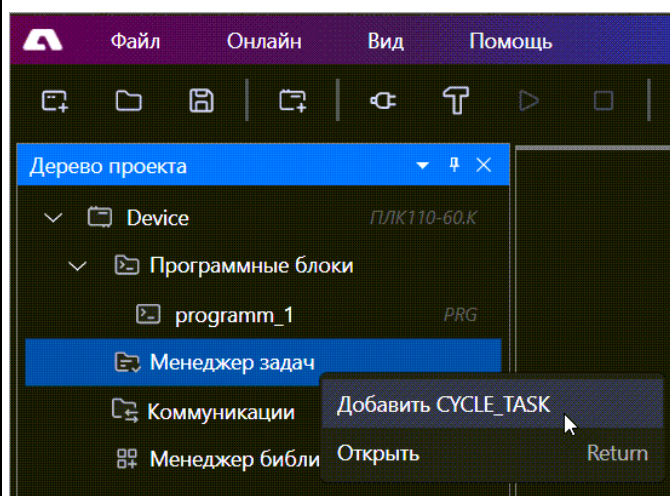


Рисунок 1.18

Нажмите ПКМ на системную папку **Менеджер задач** и выберите **Добавить CYCLE_TASK**

2

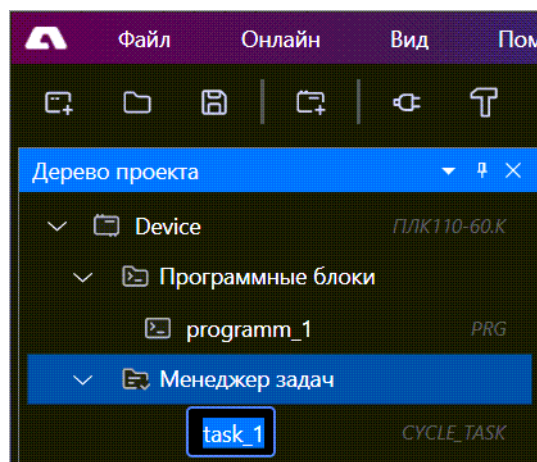


Рисунок 1.19

Вы можете присвоить задаче любое имя, в рамках правил именования, указанных в Руководстве Пользователя. Мы будем пользоваться именем по умолчанию. Нажмите **Enter**

3

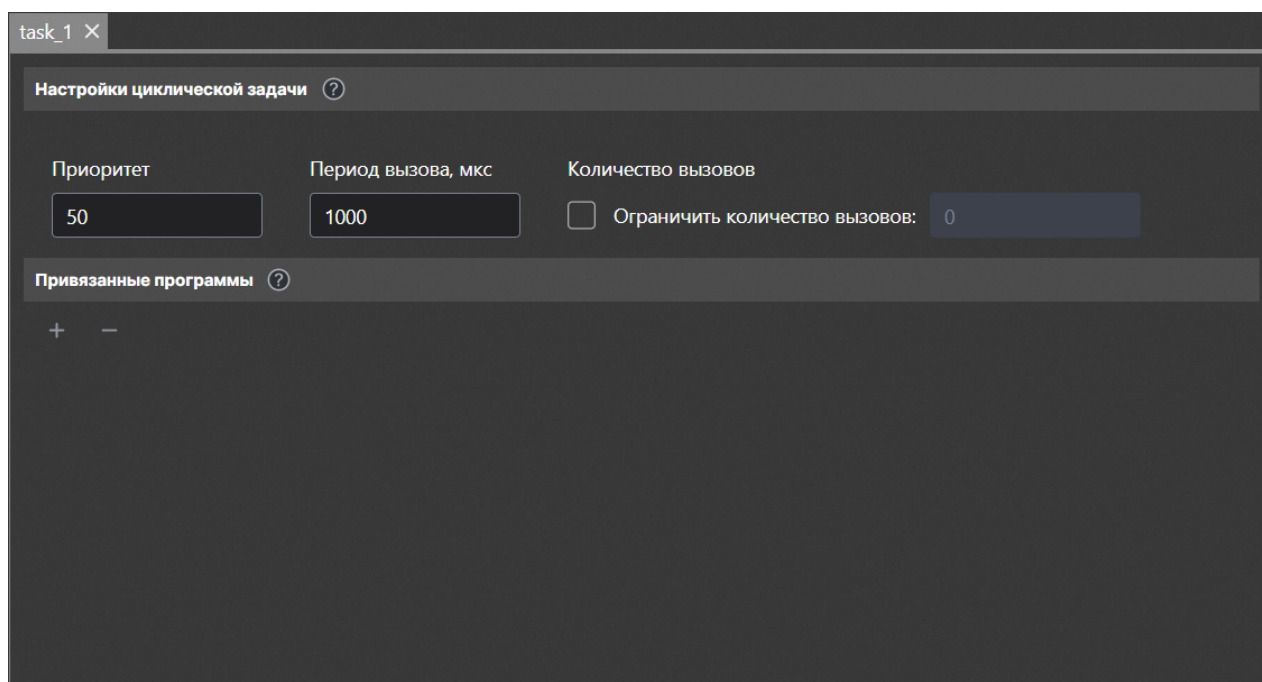


Рисунок 1.20

Откроется окно редактора циклической задачи. Т.к. в нашем проекте будет присутствовать одна задача, приоритет задавать необязательно.

Значения **Период вызова** и **Количество вызовов** мы также оставим по умолчанию. Мы видим, что пока к нашей задаче не привязана ни одна программа

4

Привязать программу к задаче можно одним из двух способов:

1)

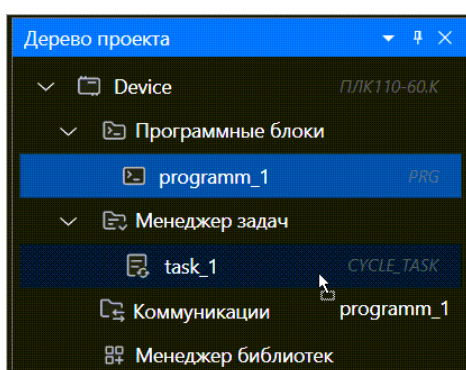


Рисунок 1.21

Нажмите ЛКМ на созданную программу в **Дереве проекта** и с помощью drag&drop переместите программу в задачу

2)

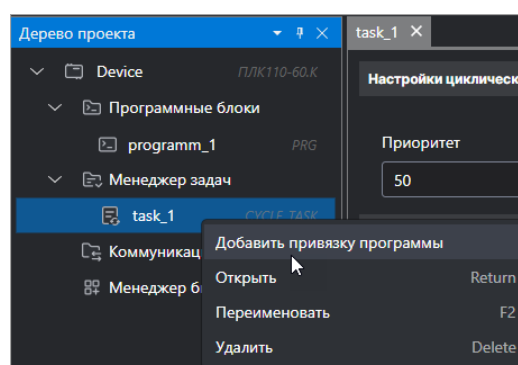


Рисунок 1.22

Нажмите ПКМ на созданную программу в **Дереве проекта**, выберите **Добавить привязку программы**.

В открывшемся окне выберите созданную программу (у нас это *programm_1*). Нажмите **Сохранить**.

5

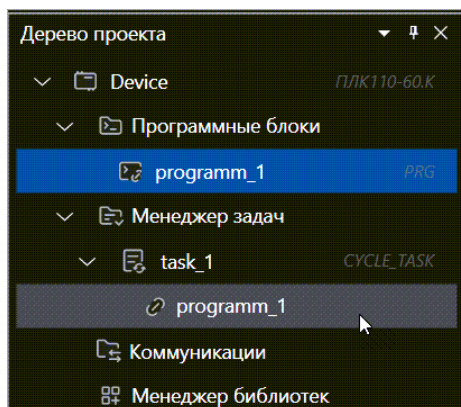


Рисунок 1.23

Программа отобразилась в дереве проекта со значком привязки.

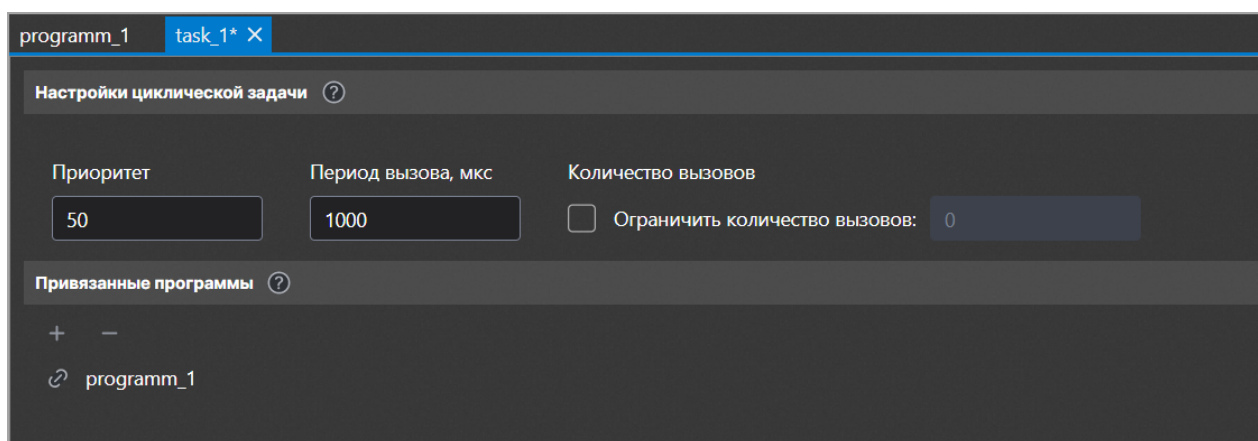


Рисунок 1.24

И появилась в списке привязанных программ во вкладке редактора задачи

1.5 Настраиваем входы и выходы

Привяжем к входам и выходам переменные, которые мы указали в коде программы.

1

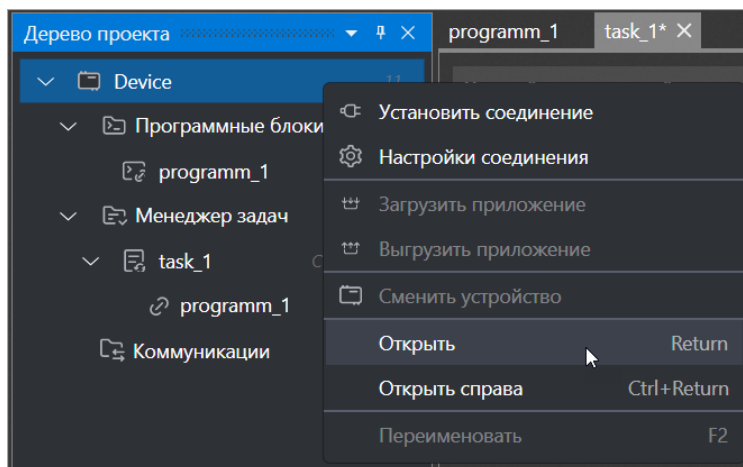


Рисунок 1.25

Нажмите ПКМ на системную папку **Device** в **Дерево проекта** и выберите **Открыть**. Или дважды нажмите ЛКМ по компоненту **Device** в **Дерево проекта**

2

Битовая маска				
Параметр	Значение/Переменная	Тип доступа	Тип данных	
FDI (Четыре счётчика)				
FDO				
DI	Дискретное значение	Чтение	BYTE	
DO				
Периферия				
Счетчик (FDI1)				
Параметр	Значение/Переменная	Тип доступа	Тип данных	
Количество импульсов с...	0	Чтение	UDINT	

Рисунок 1.26

Откроется вкладка **Device**, содержащая настройку входов/выходов. В программе будут задействованы DI1 и DO1 (отсчет идет не считая FDI и FDO). На корпусе ПЛК DI5 и DO5 соответственно, если учитывать в нумерации быстрые входы/выходы.

3

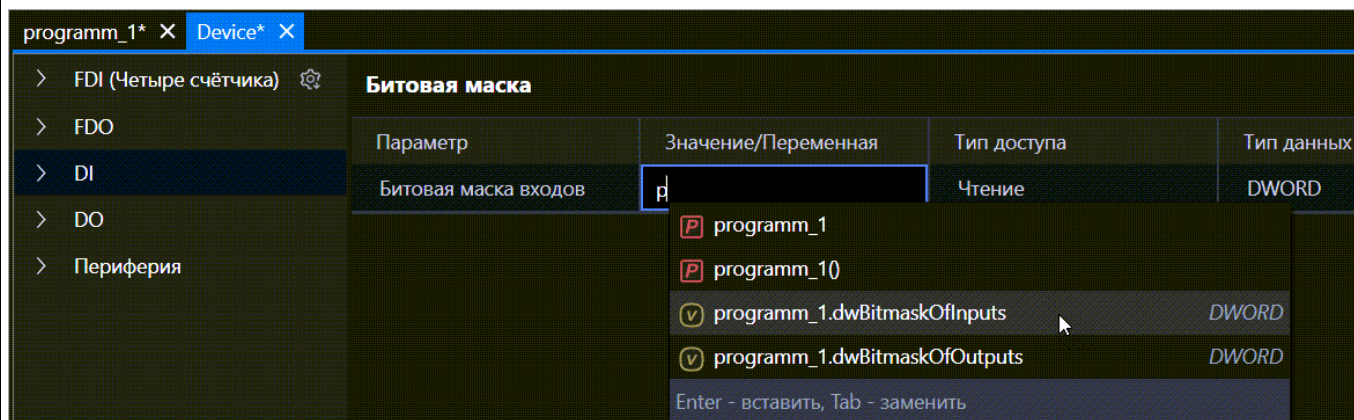


Рисунок 1.27

Выберем в списке входов/выходов, расположенном в левой части окна, входы DI. Привяжем переменную ко входу. Начнем вводить в строке Значение/Переменная имя программы. Автодополнение в ALTA IDE предложит варианты. Выберем *programm_1.dwBitmaskOfInputs*.

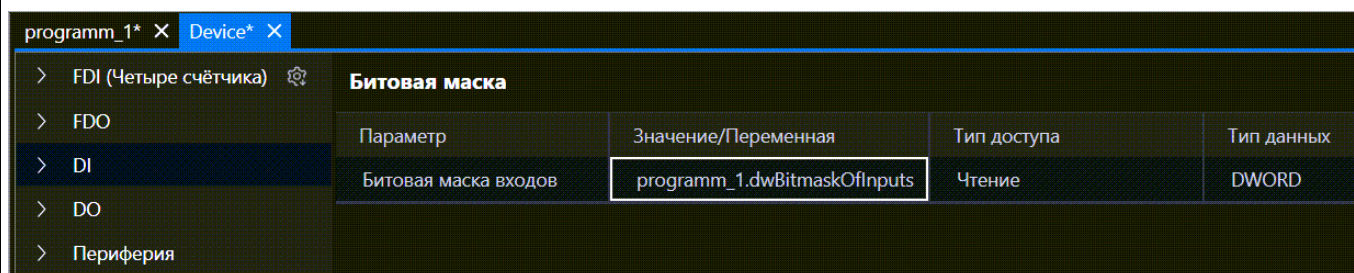


Рисунок 1.28

Поля параметр, тип доступа и тип данных заполнены автоматически и недоступны для редактирования

4

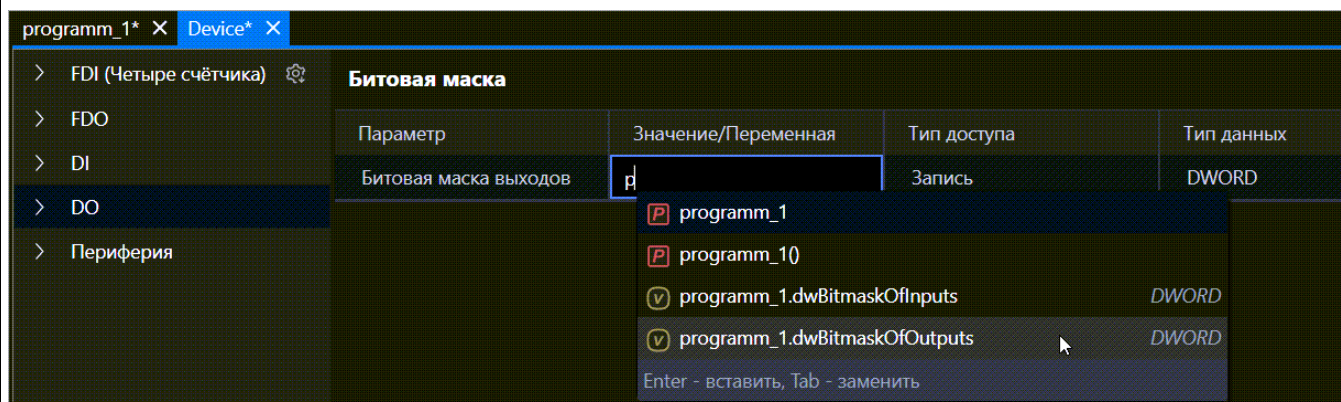


Рисунок 1.29

Для привязки переменной к выходу выполним аналогичные действия. Выберем в списке входов/выходов, расположенном в левой части окна, выходы DO. Начнем вводить в строке Значение/Переменная имя программы. Автодополнение в ALTA IDE предложит варианты. Выберем *programm_1.dwBitmaskOfOutputs*.

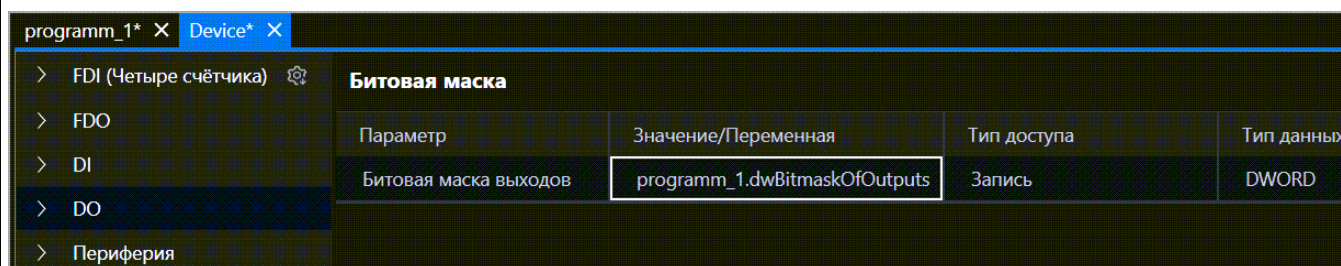


Рисунок 1.30

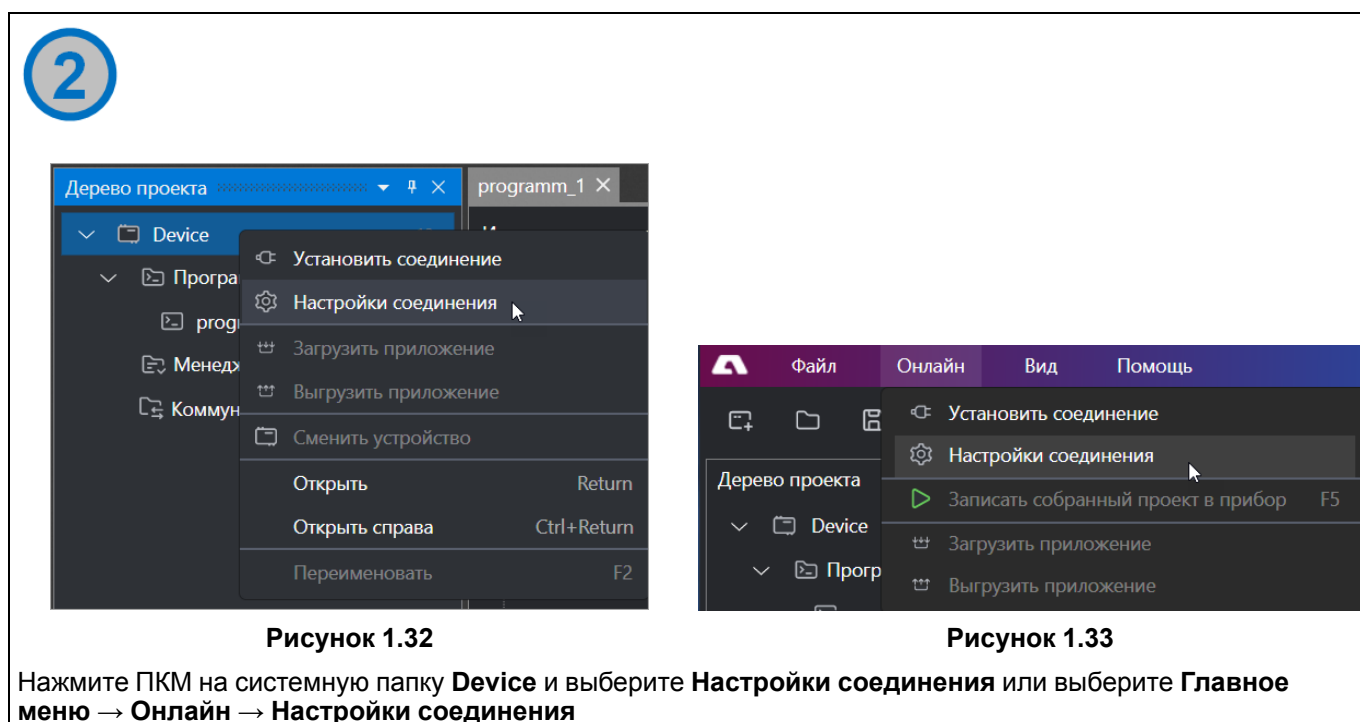
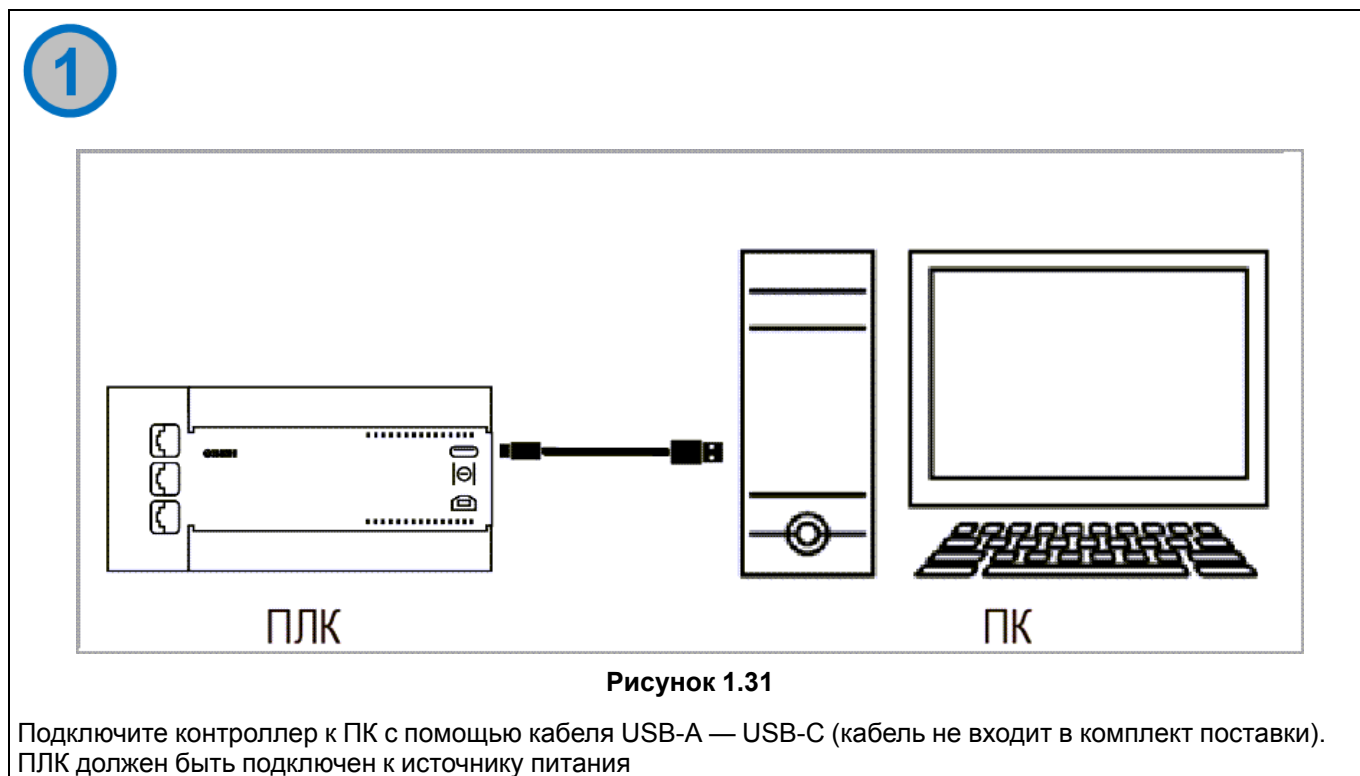
Поля параметр, тип доступа и тип данных заполнены автоматически и недоступны для редактирования

1.6 Подключаем и настраиваем ПЛК

Самое время подключить ПЛК.

Мы будем рассматривать подключение контроллера по интерфейсу USB Device с сетевыми настройками по умолчанию.

При первом подключении ПЛК к ПК с ОС Windows может потребоваться установка драйвера RNDIS. Драйвер доступен в WEB-конфигураторе на странице **Загрузки** или на [сайте](#).



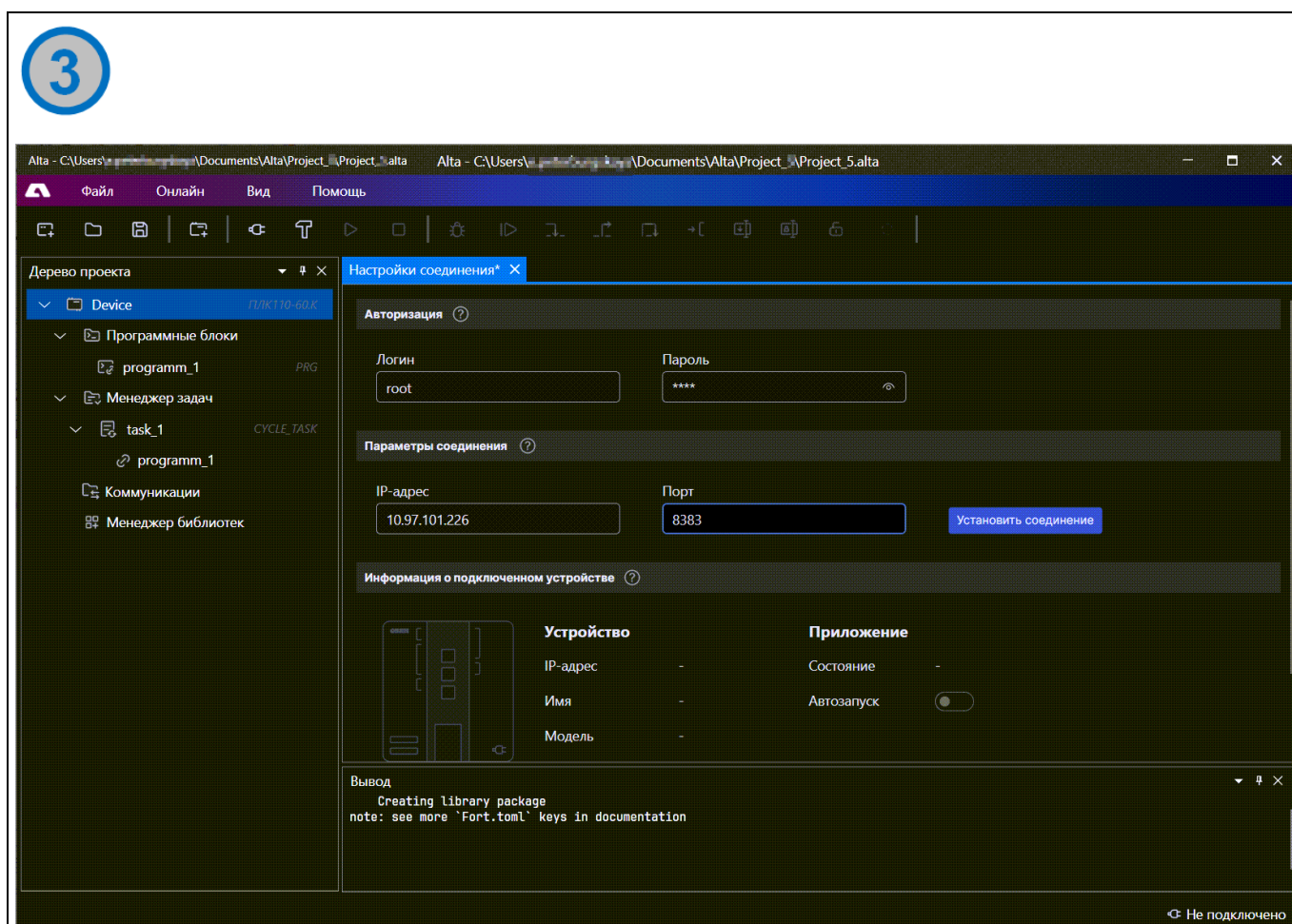


Рисунок 1.34

В открывшейся вкладке **Настройки соединения** авторизуйтесь и введите параметры соединения. Настройки по умолчанию для ПЛК110:

Логин: **root**

Пароль: **owen**

IP-адрес: **10.0.6.10**

Порт: **8383**

Нажмите **Установить соединение**. Иконка статуса подключения на панели инструментов, в строке состояния и на схематическом изображении ПЛК во время установки соединения поменяет цвет на синий

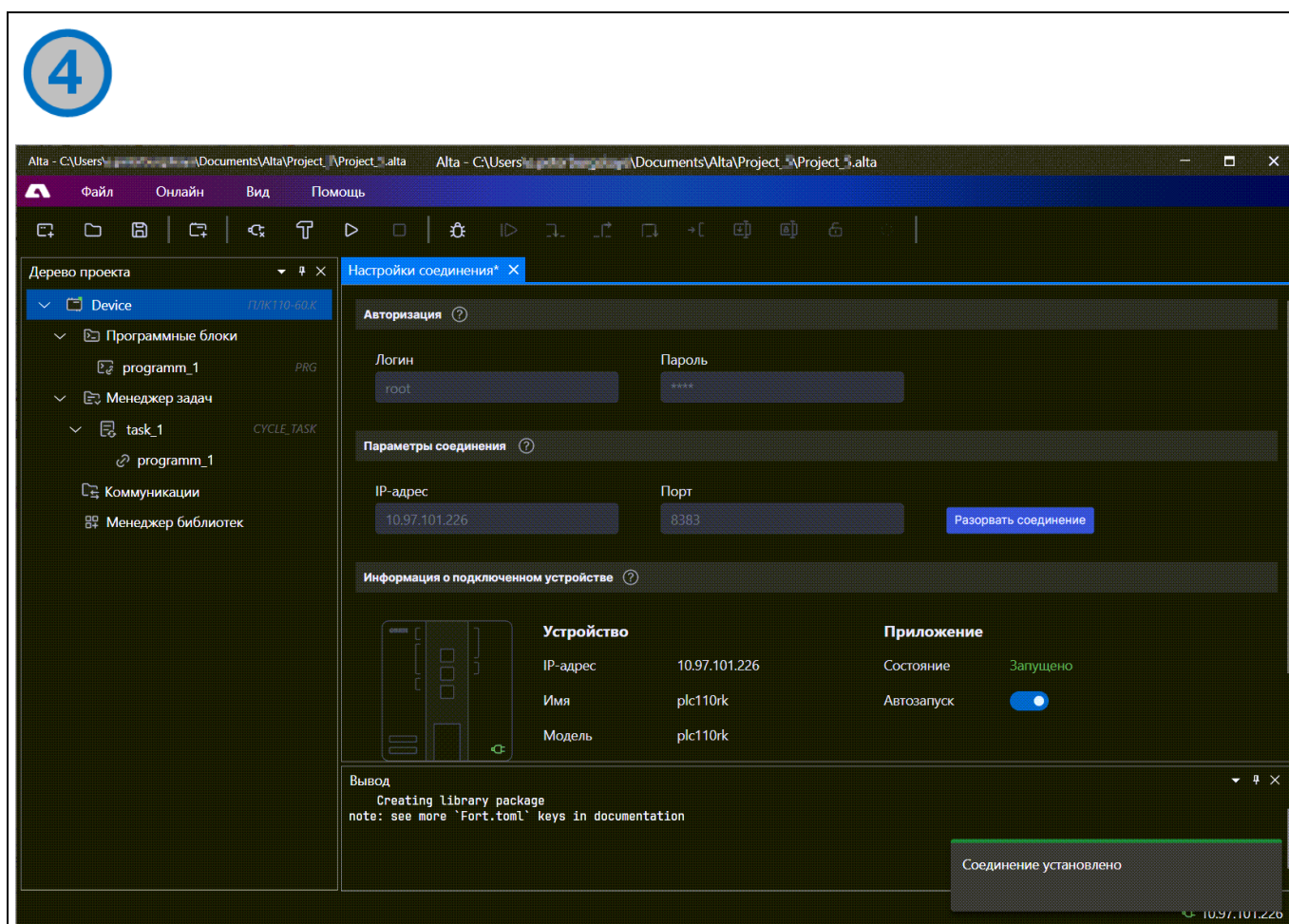
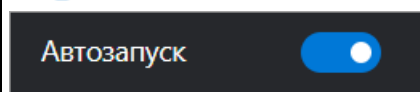


Рисунок 1.35

Соединение установлено. В информации о подключенном устройстве отображается IP-адрес, имя, модель и модификация ПЛК. Иконка статуса подключения на панели инструментов, в строке состояния и на схематическом изображении ПЛК меняет цвет на зеленый

5



Автозапуск приложения включен по умолчанию. Это означает, что исполняемое приложение автоматически запустится на ПЛК (например, после перезагрузки)

1.7 Сборка и загрузка проекта

Проект создан, осталось выполнить сборку и загрузить приложение в ПЛК.

1

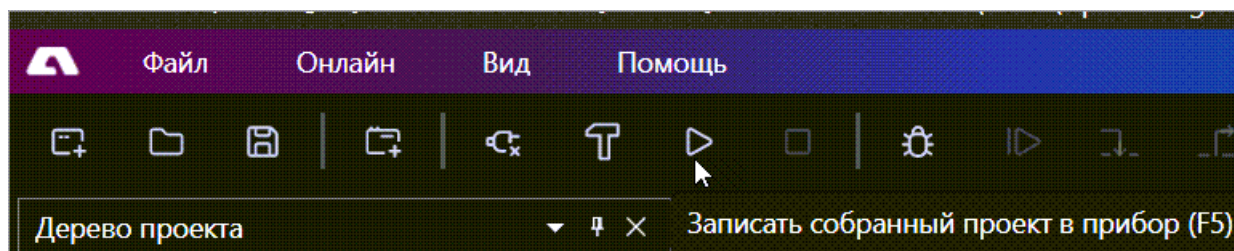


Рисунок 1.36

Нажмите на кнопку **Старт** на панели инструментов. ALTA IDE сохранит все изменения, выполнит сборку проекта и запишет приложение в прибор. Впоследствии с помощью этой кнопки будет осуществляться запуск приложения на приборе

2

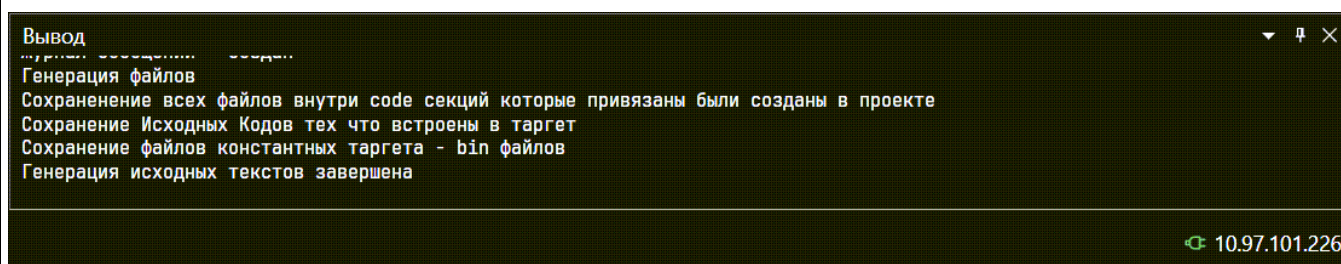


Рисунок 1.37

В окне вывода отображается информация о компиляции проекта. Если в процессе сборки возникнут ошибки они также будут отражены в окне вывода

3

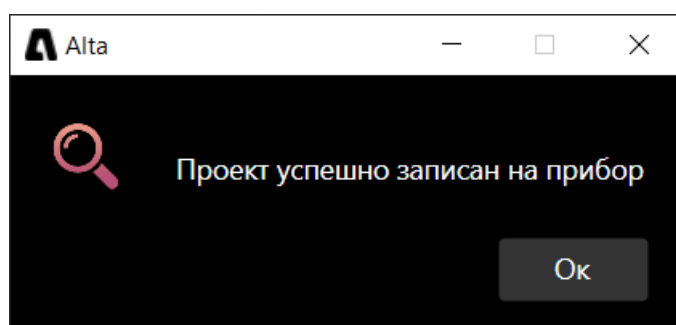


Рисунок 1.38

Успешная загрузка проекта в прибор

1.8 Режим онлайн

Перейдем в режим онлайн и установим нужное нам значение переменной, для того чтобы увидеть работу выхода прибора.



2

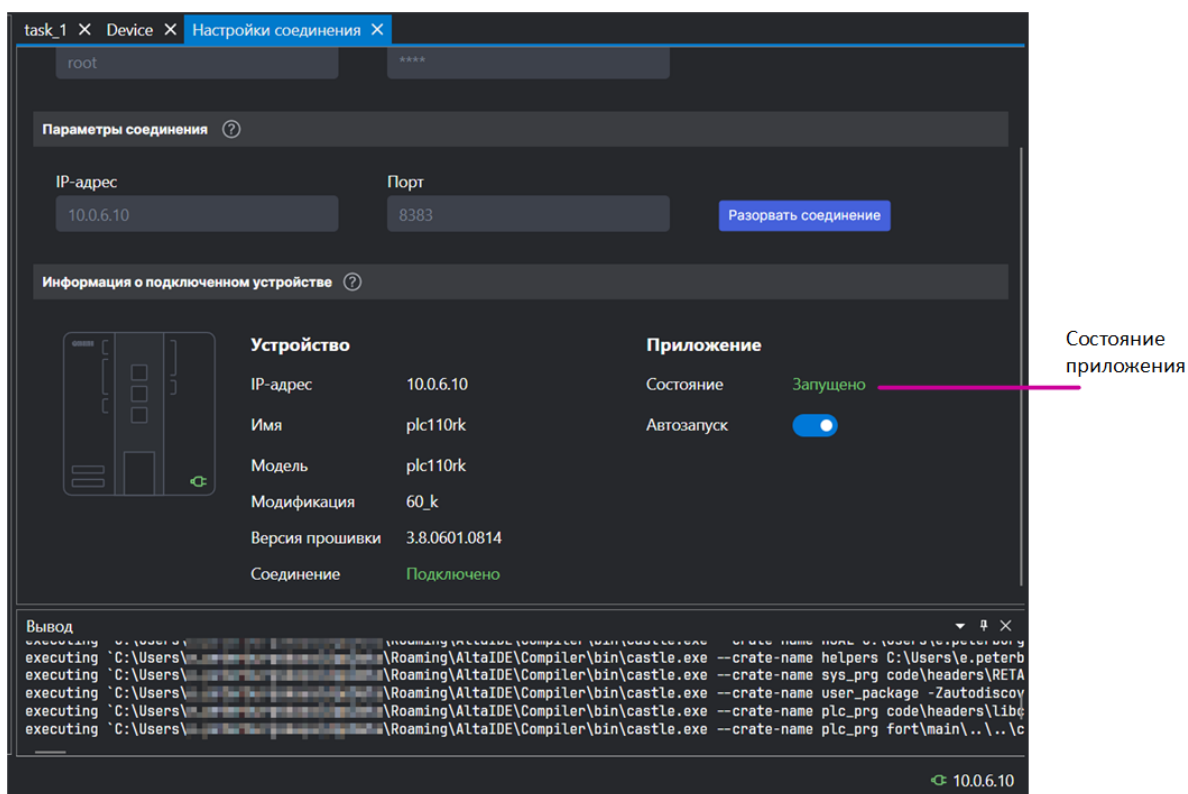


Рисунок 1.40

Перед переходом в режим онлайн убедимся, что приложение запущено. Во вкладке **Настройки соединения** проверим состояние приложения.

Если состояние приложения **Остановлено**, нажмите кнопку **Старт** на панели инструментов

3

Рисунок 1.41

Нажмите ЛКМ на кнопку **Онлайн** на панели инструментов

4

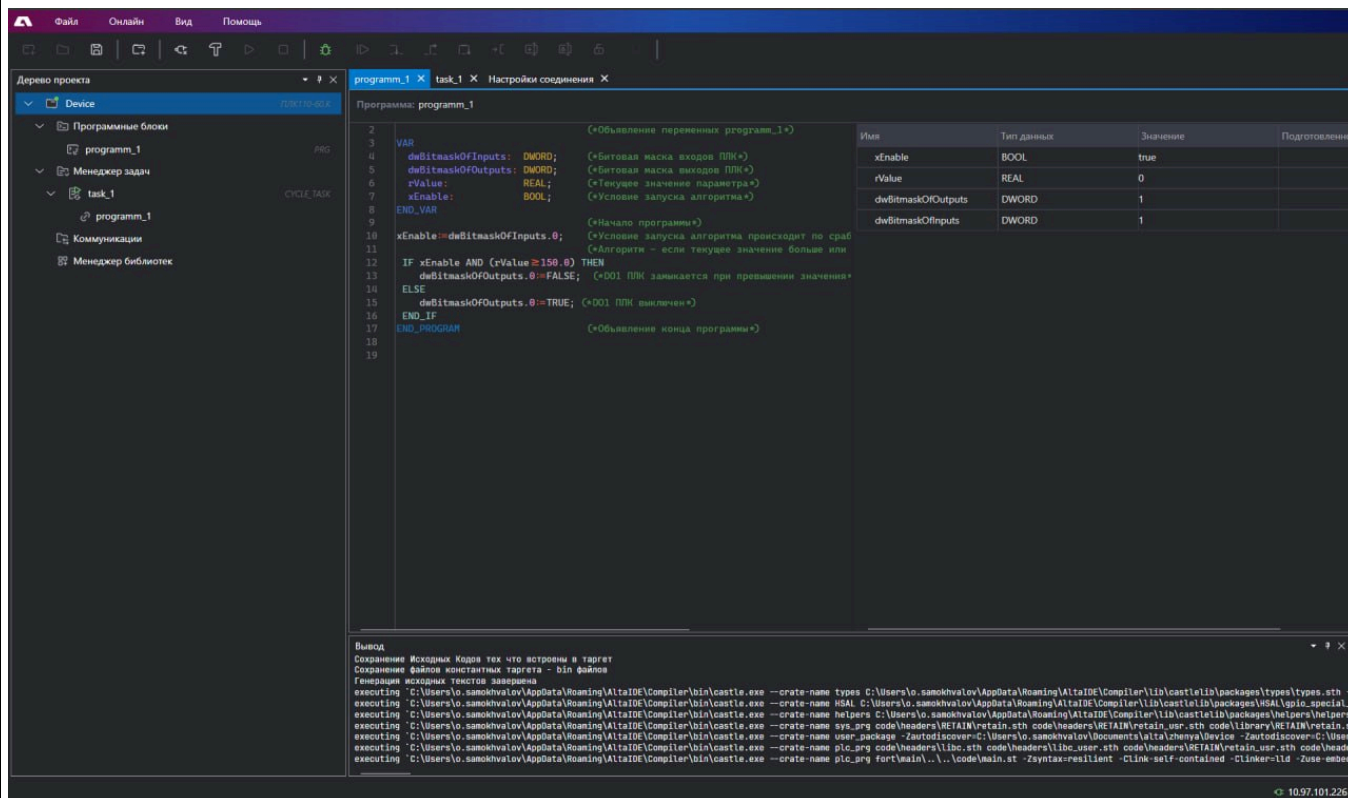


Рисунок 1.42

Кнопка **Онлайн** поменяла цвет на зеленый, на панели инструментов отобразилась панель отладки. Перейдите во вкладку редактора ST, где мы писали программу: справа в рабочей области появилась таблица локальных переменных - это значит, режим онлайн активирован

5

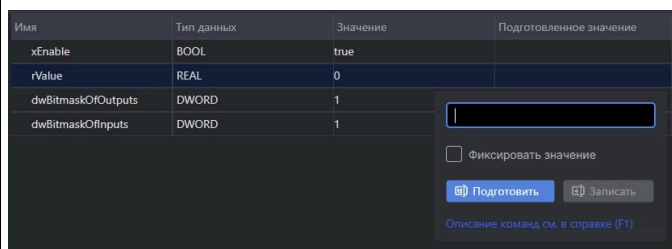


Рисунок 1.43

В таблице локальных переменных найдем строку переменной **rValue**. Дважды нажмите **ЛКМ** в столбце **Подготовленное значение**. Откроется окно ввода значения переменной

6

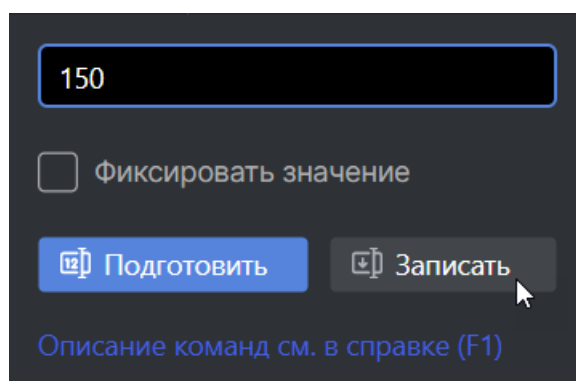


Рисунок 1.44

Введите значение 150, при достижении которого, по условию программы, должен замкнуться выход DO1. Нажмите **Записать**

7

Имя	Тип данных	Значение
xEnable	BOOL	true
rValue	REAL	150
dwBitmaskOfOutputs	DWORD	1
dwBitmaskOfInputs	DWORD	1

Рисунок 1.45

Переменная приняла записанное значение

8

Уровень напряжения,
соответствующий
логической единице

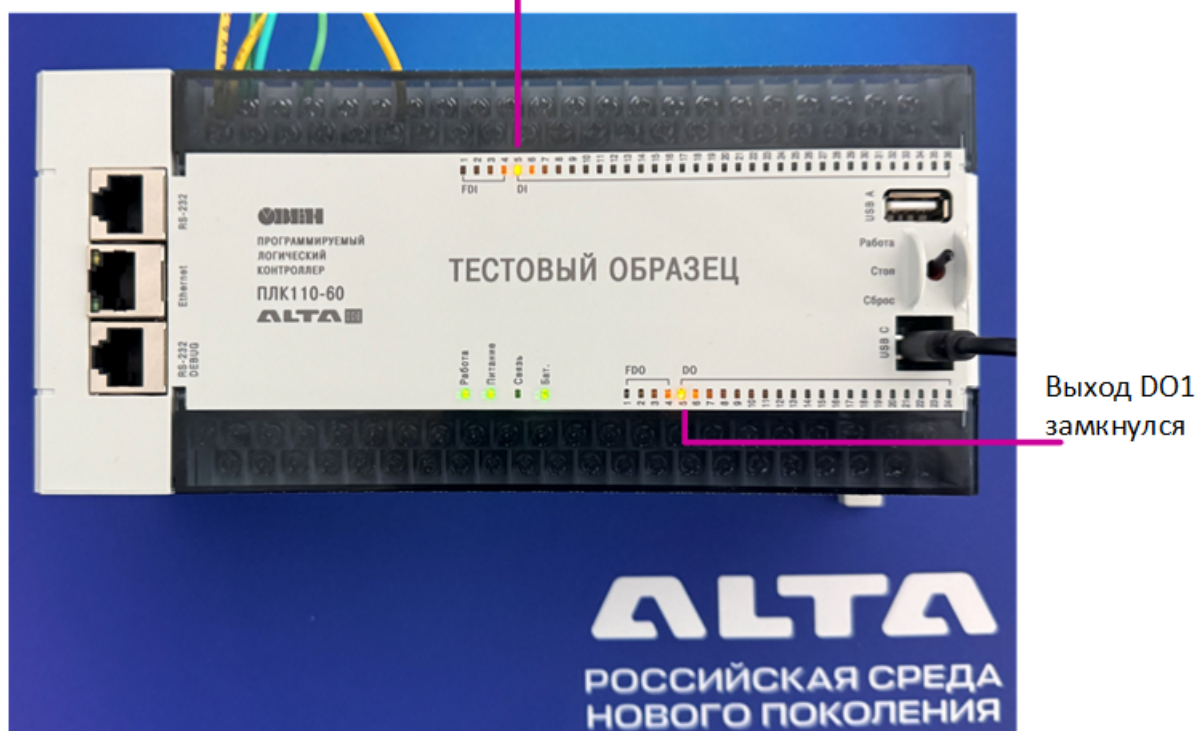


Рисунок 1.46

Выход замкнулся, программа выполнена



ЦИФРОВЫЕ
РЕШЕНИЯ

ООО "Овен Цифровые решения"

Россия, г. Москва, пл. Семёновская, д. 1А, помещ. 3/1

support@owendigital.ru

www.owendigital.ru

рег.:1-RU-154391-1.8