



ЦИФРОВЫЕ  
РЕШЕНИЯ

# ALTA IDE (версия 1.0)

# ALTA

Быстрый старт

03.2026  
версия 1.5

---

# Содержание

<b>1 Быстрый старт .....</b>	<b>3</b>
1.1 Устанавливаем ALTA IDE .....	4
1.2 Создаем проект .....	6
1.3 Пишем программу.....	8
1.4 Создаем задачу .....	10
1.5 Настраиваем входы и выходы.....	12
1.6 Подключаем и настраиваем ПЛК .....	16
1.7 Сборка и загрузка проекта .....	19
1.8 Режим онлайн.....	20

# 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

Архив поставки содержит следующие каталоги:

1. **01\_Software** — среда ALTA IDE и компоненты среды, включая компилятор.
2. **03\_USB\_Driver** — драйвер RNDIS.
3. **05\_Documentation** — инструкции и сопроводительная документация.



**ВНИМАНИЕ**  
ALTA IDE версии 1.0 работает только в ОС Windows

1

Перейдите в каталог **01\_Software\ALTA\_IDE\Alta\_install** и запустите файл *Alta\_Setup.exe*.  
Существует вероятность, что система безопасности Windows остановит выполнение приложения:

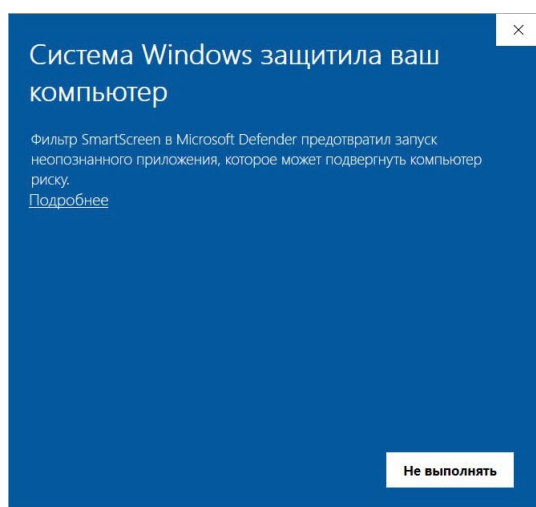


Рисунок 1.1

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

2

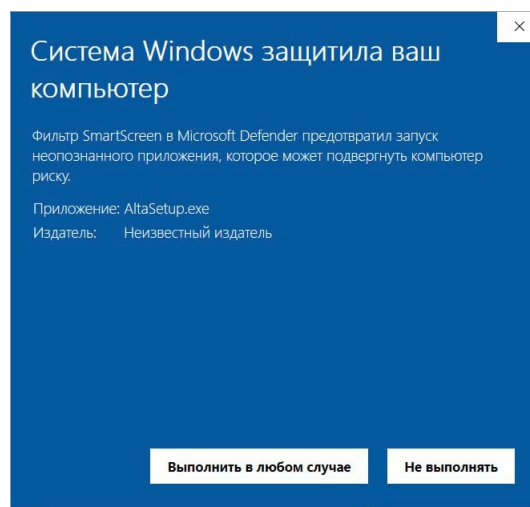


Рисунок 1.2

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

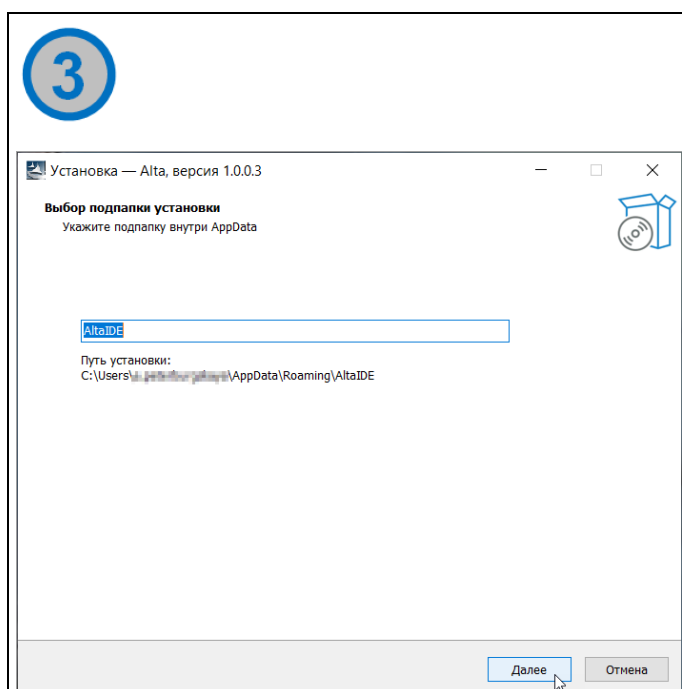


Рисунок 1.3

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

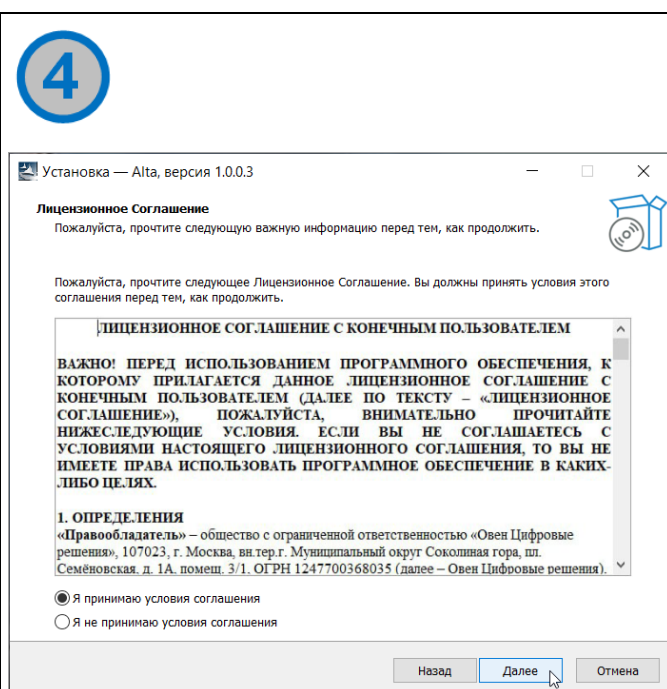


Рисунок 1.4

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

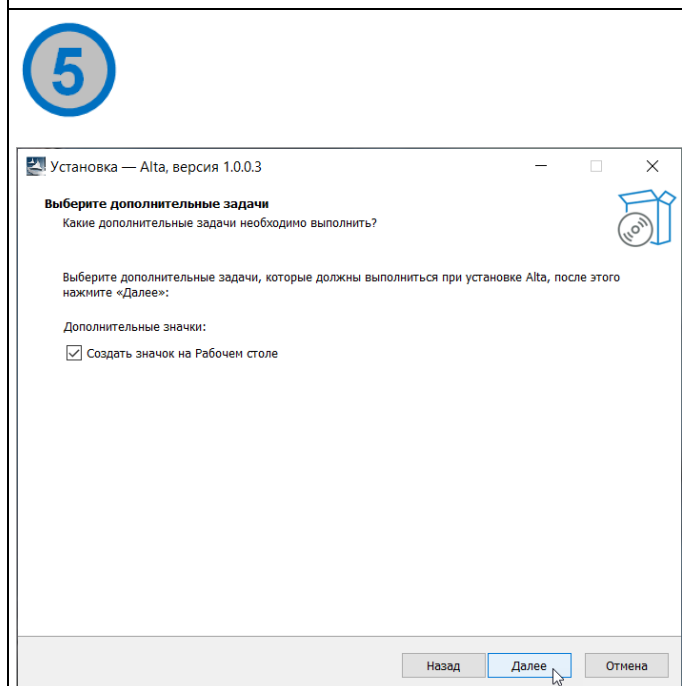


Рисунок 1.5

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

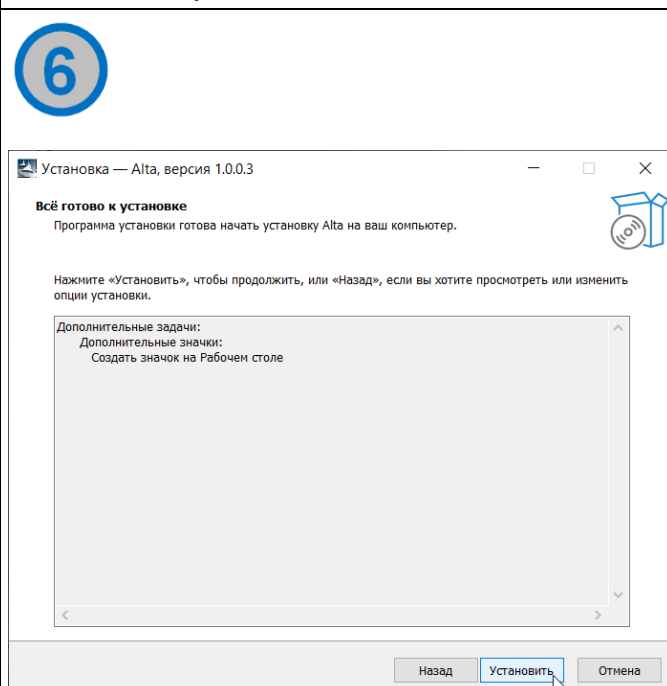
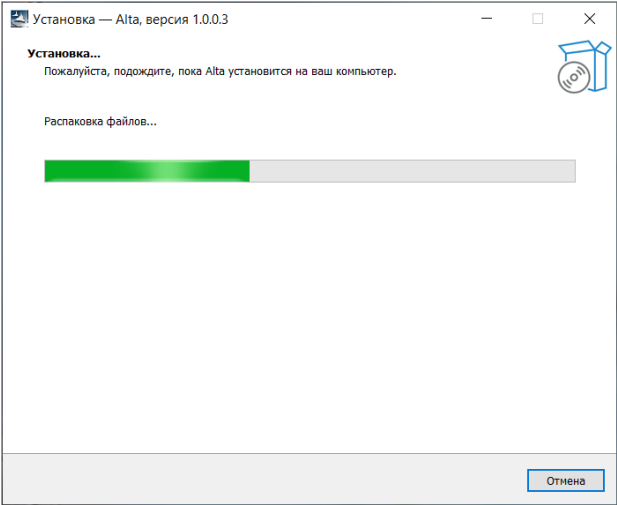


Рисунок 1.6

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

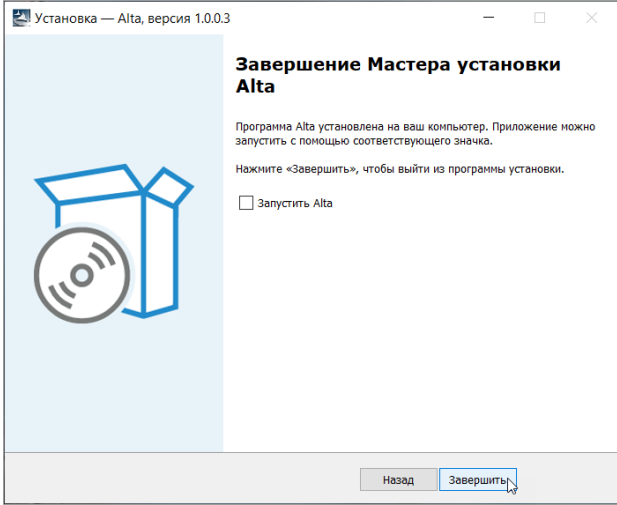
**7**



**Рисунок 1.7**

Дождитесь окончания установки

**8**



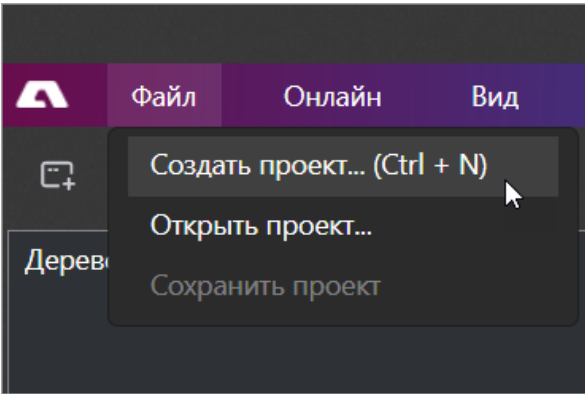
**Рисунок 1.8**

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

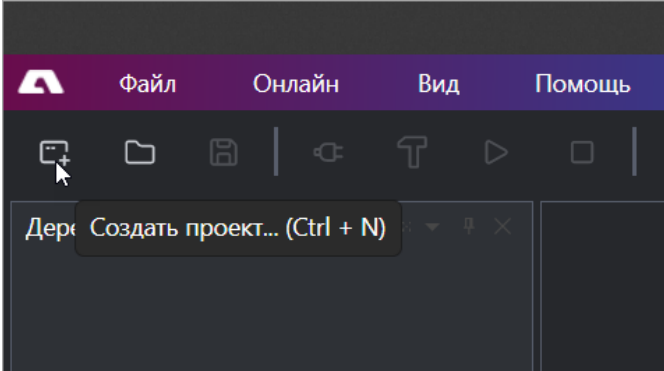
## 1.2 Создаем проект

Создадим наш первый проект:

**1**



**Рисунок 1.9**



**Рисунок 1.10**

**Главное меню** → **Файл** → **Создать проект** или ЛКМ (левой кнопкой мыши) по кнопке **Создать проект** на панели инструментов

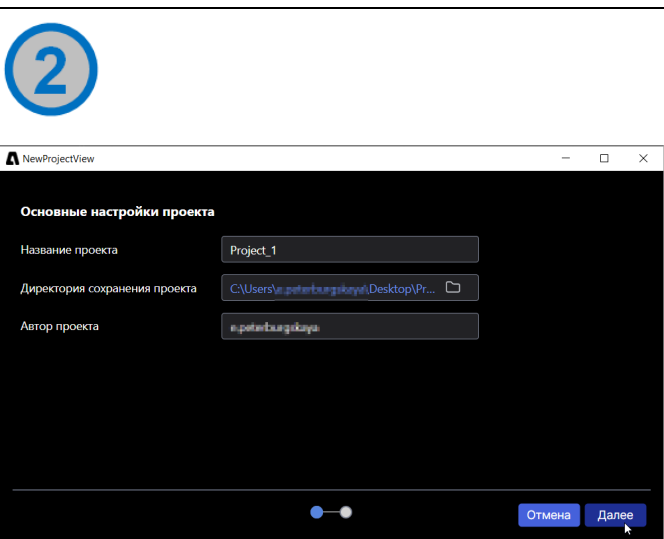


Рисунок 1.11

Вы можете присвоить проекту любое имя, в рамках правил именования, указанных в Руководстве Пользователя. Мы будем пользоваться именем по умолчанию. Выберите директорию для сохранения проекта. Нажмите кнопку **Далее**

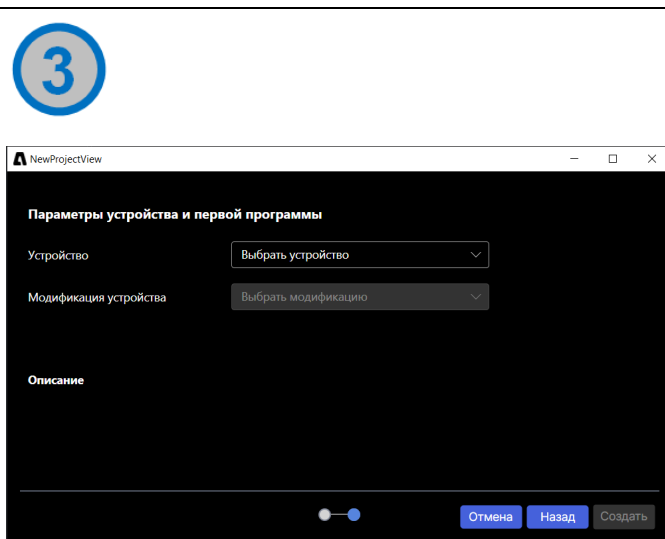


Рисунок 1.12

В качестве устройства у нас ПЛК210 модификация 12. Вы должны выбрать данные того устройства, которое будете использовать в проекте. Нажмите кнопку **Создать**

4

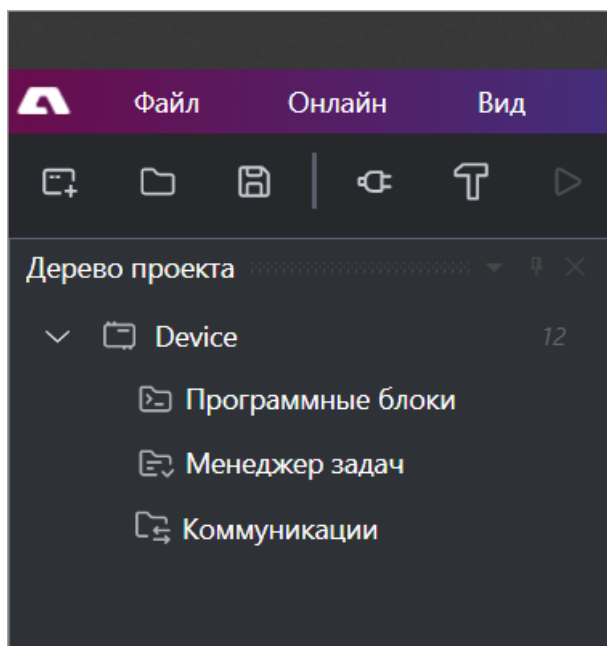


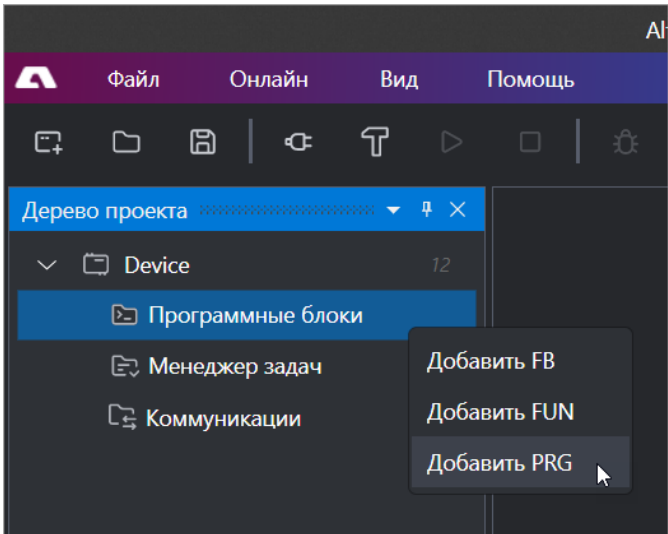
Рисунок 1.13

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

## 1.3 Пишем программу

Пришло время создать и написать программу:

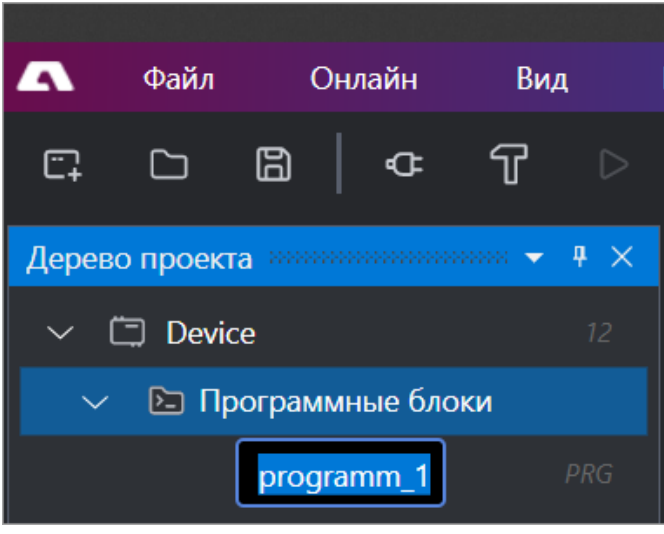
1



**Рисунок 1.14**

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

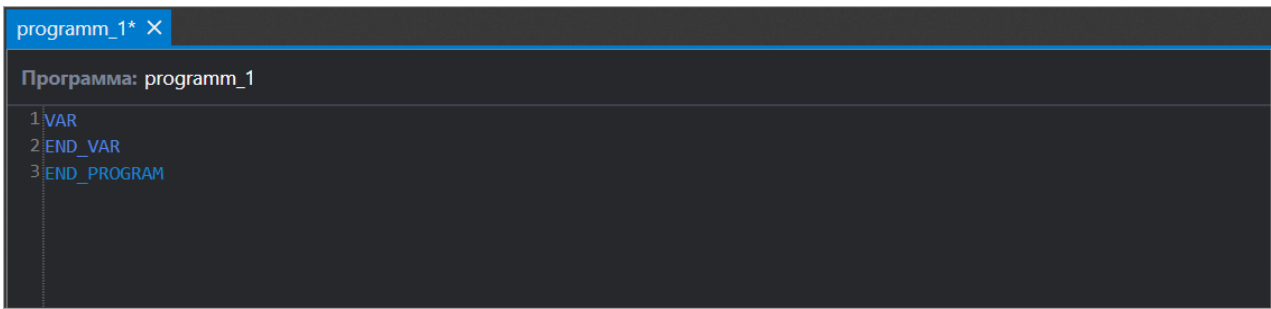
2



**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



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

```

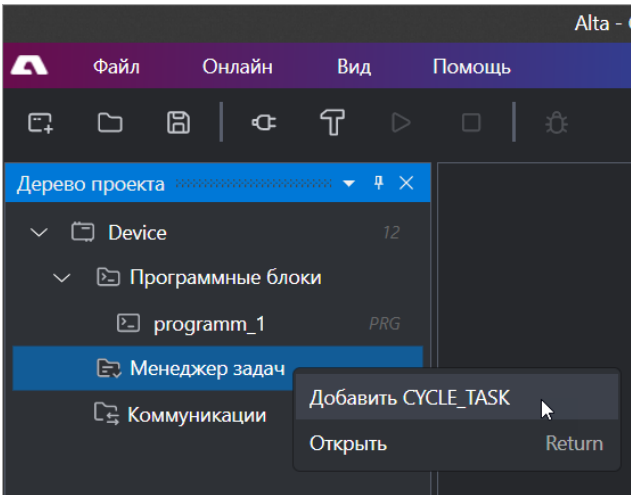
(*Объявление переменных programm_1*)
VAR
  wBitmaskOfInputs:   WORD;      (*Битовая маска входов ПЛК*)
  byBitmaskOfOutputs: BYTE;      (*Битовая маска выходов ПЛК*)
  rValue:             REAL;      (*Текущее значение параметра*)
  xEnable:            BOOL;      (*Условие запуска алгоритма*)
END_VAR
(*Начало программы*)
xEnable:=wBitmaskOfInputs.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
  byBitmaskOfOutputs.0:=TRUE;    (*DO1 ПЛК замыкается при превышении значения*)
ELSE
  byBitmaskOfOutputs.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:

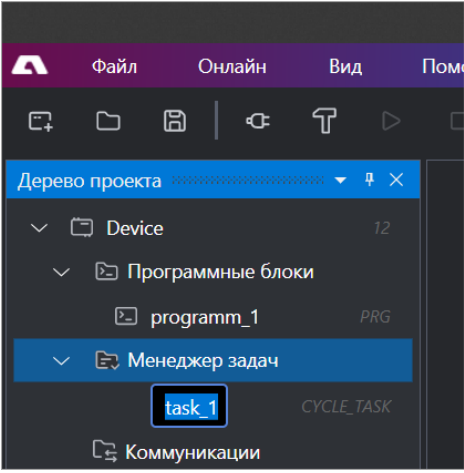
1



**Рисунок 1.18**

Нажмите ПКМ на системную папку **Менеджер задач** и выберите **Добавить CYCLE\_TASK**

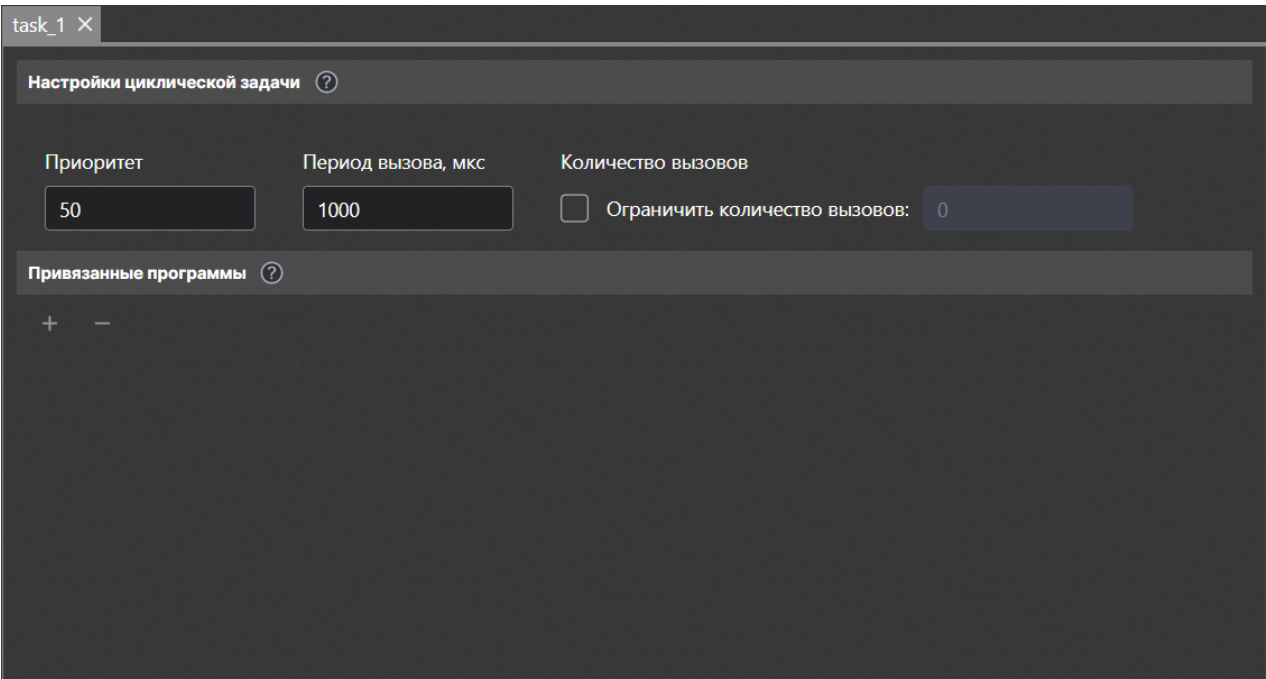
2



**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

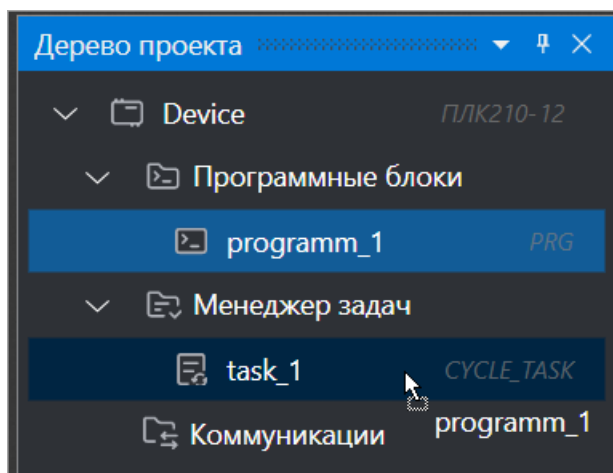


Рисунок 1.21

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

5

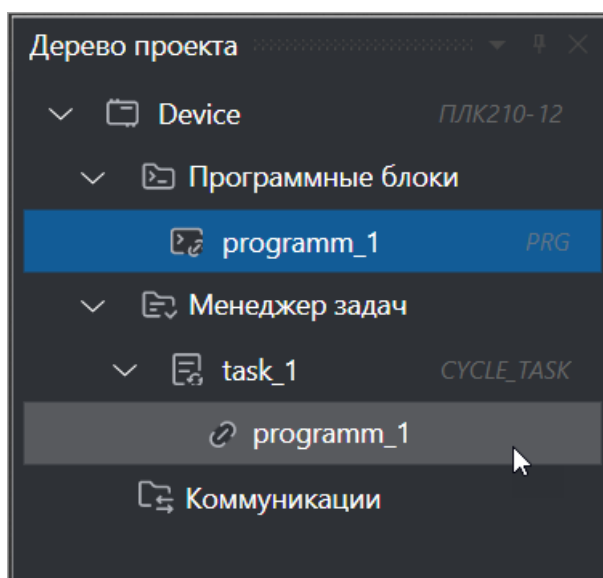


Рисунок 1.22

Программа отобразилась в дереве проекта со значком привязки.

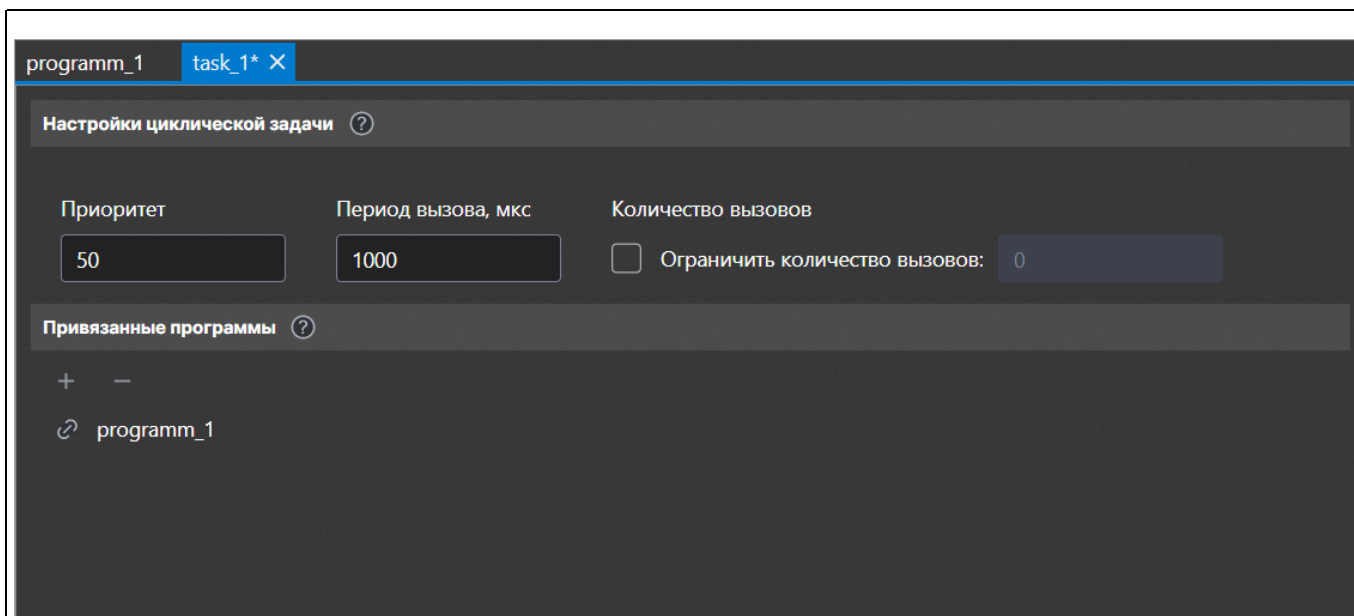


Рисунок 1.23

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

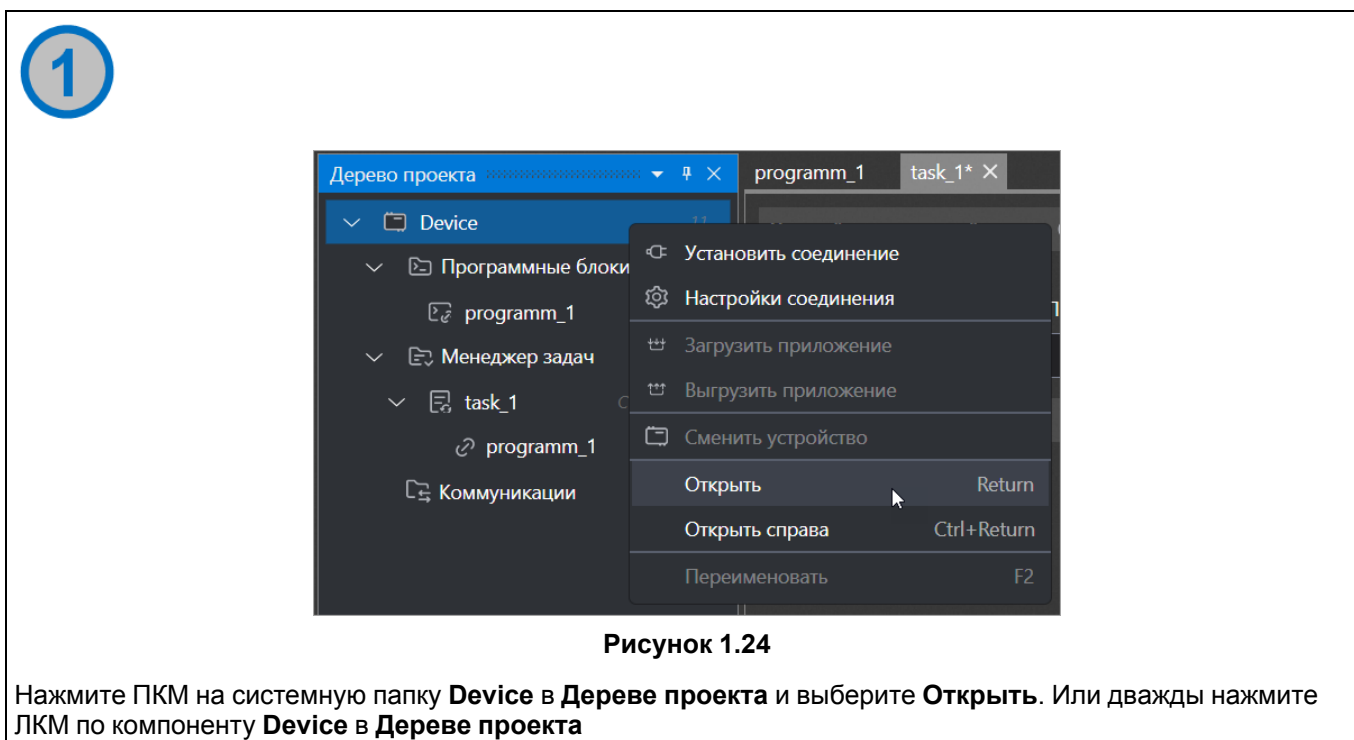


Рисунок 1.24

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

2

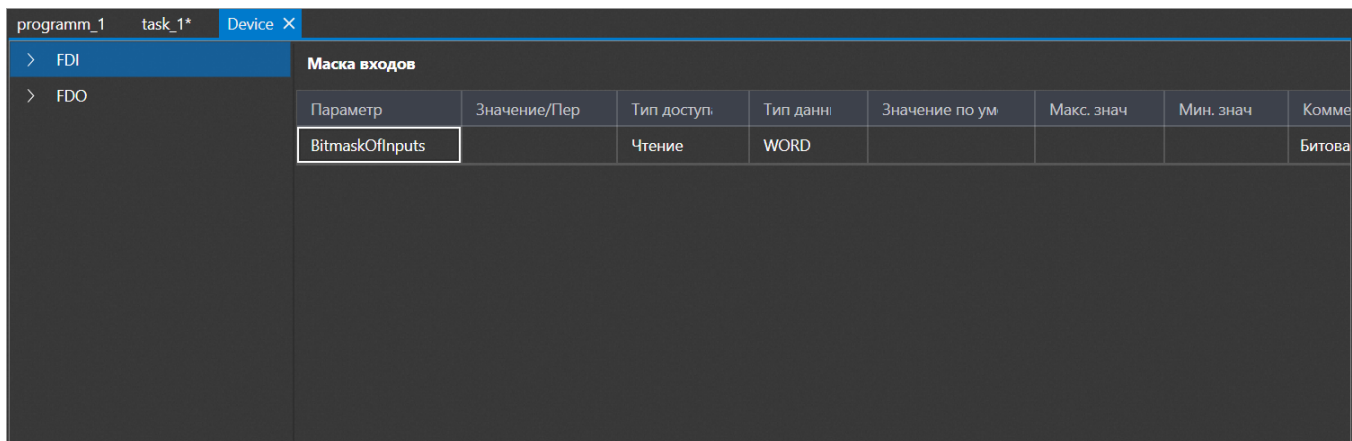


Рисунок 1.25

Откроется вкладка **Device**, содержащая настройку входов/выходов

3

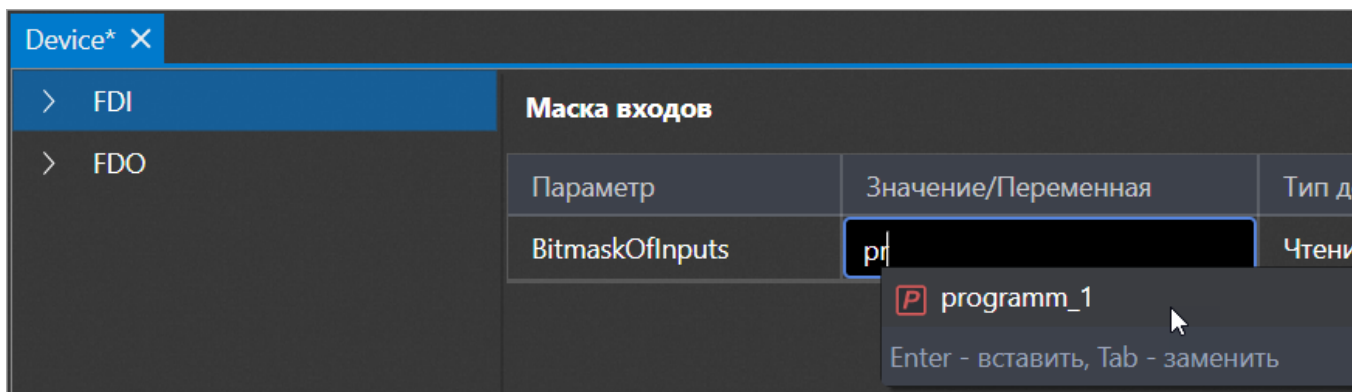


Рисунок 1.26

Привяжем переменную ко входу. Начнем вводить в строке Значение/Переменная имя программы. Автодополнение в ALTA IDE предложит вариант *programm\_1* - выберем его.

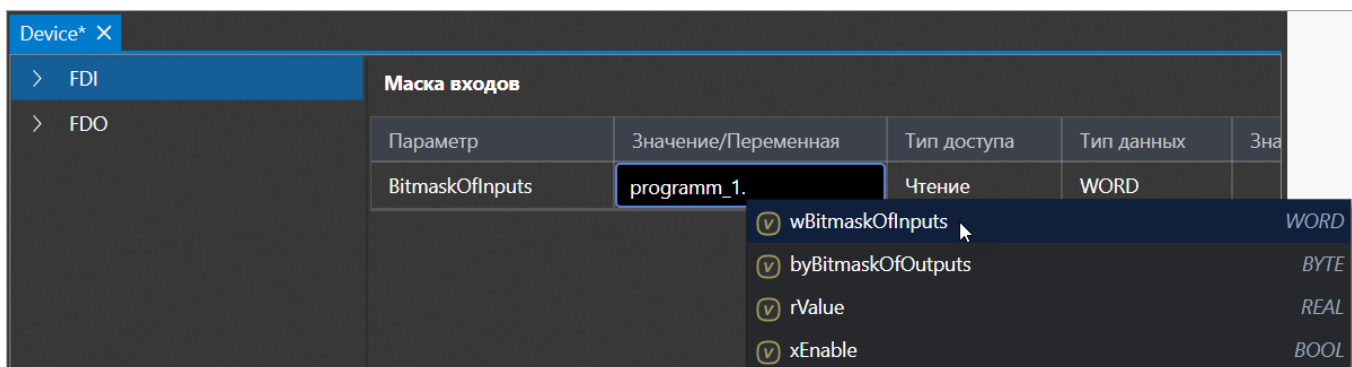


Рисунок 1.27

Ставим точку, автодополнение предложит варианты переменных, которые можно привязать ко входу. Выберем *wBitmaskOfInputs*.

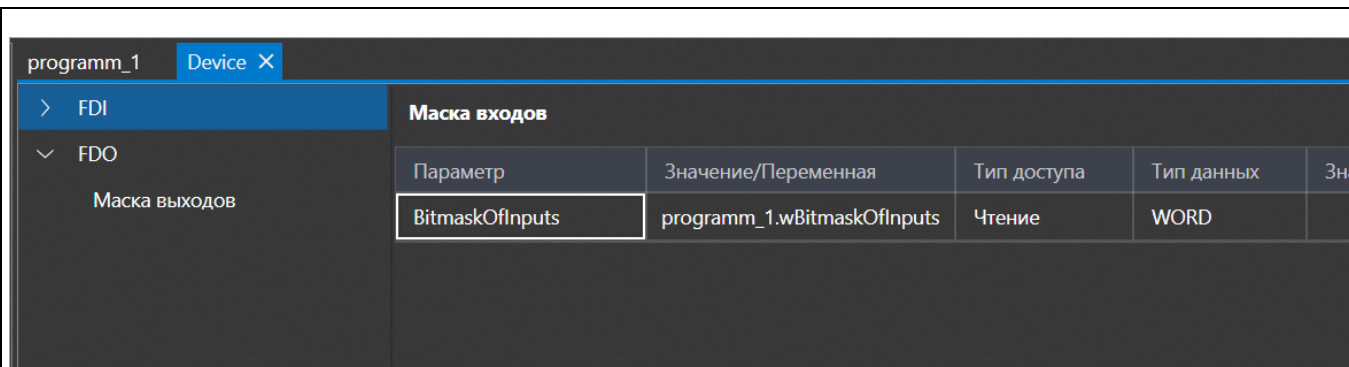


Рисунок 1.28

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

4

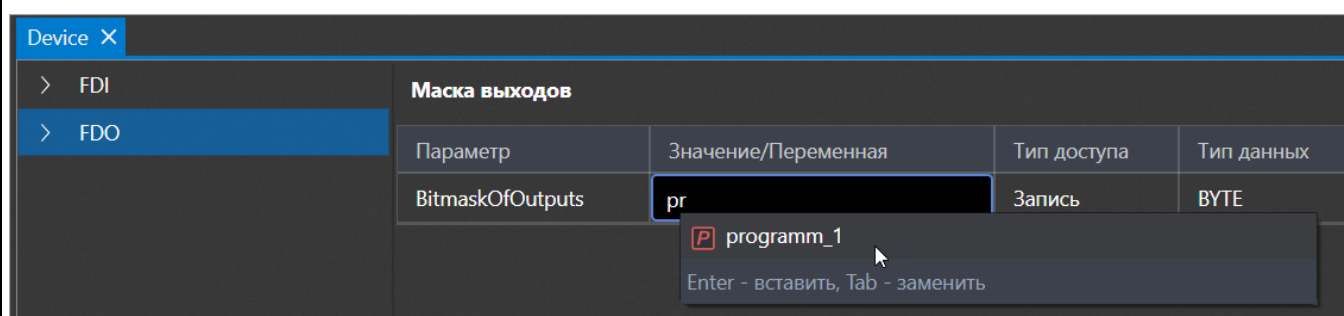


Рисунок 1.29

Для привязки переменной к выходу выполним аналогичные действия. Начнем вводить в строке Значение/Переменная имя программы. Автодополнение в ALTA IDE предложит вариант *programm\_1* - выберем его.

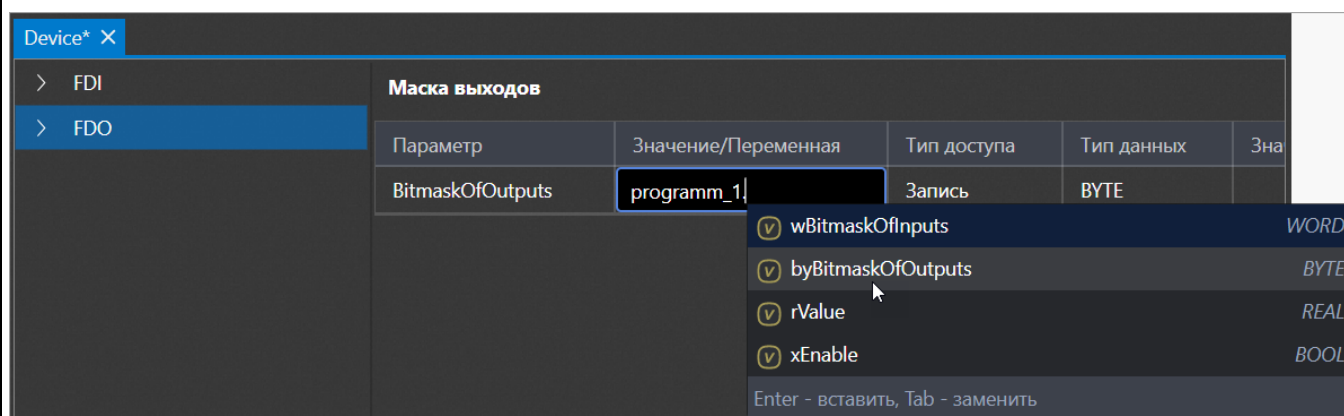


Рисунок 1.30

Ставим точку, автодополнение предложит варианты переменных, которые можно привязать к выходу. Выберем *byBitmaskOfOutputs*.

programm\_1 Device X

> FDI

▼ FDO

Маска выходов

Маска выходов			
Параметр	Значение/Переменная	Тип доступа	Тип данных
BitmaskOfOutputs	programm_1.byBitmaskOfOutputs	Запись	BYTE

**Рисунок 1.31**

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

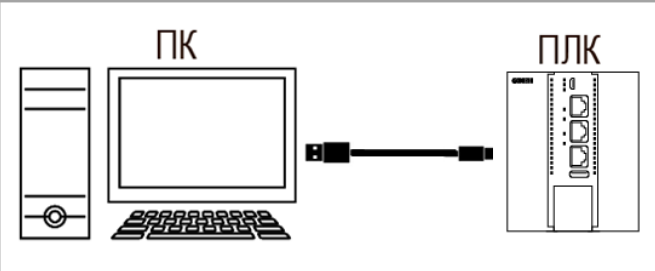
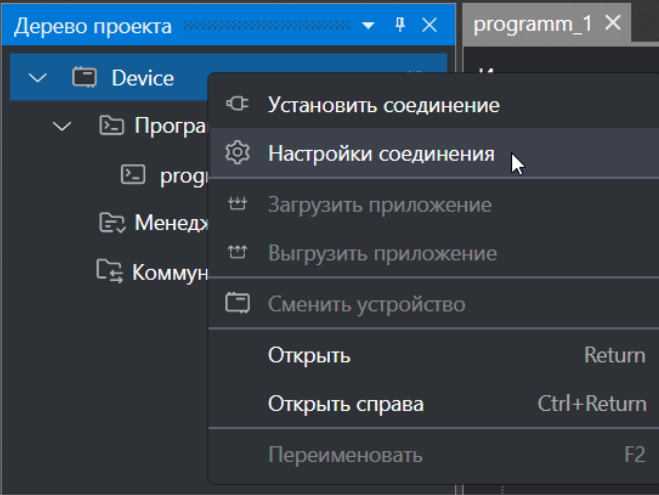
## 1.6 Подключаем и настраиваем ПЛК

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](#).

Также драйвер размещен в архиве с дистрибутивом в каталоге **03\_USB\_Driver**.

<p style="text-align: center;"><b>1</b></p>  <p style="text-align: center;"><b>Рисунок 1.32</b></p> <p>Подключите контроллер к ПК с помощью кабеля (кабель не входит в комплект поставки). ПЛК должен быть подключен к источнику питания</p>	<p style="text-align: center;"><b>2</b></p>  <p style="text-align: center;"><b>Рисунок 1.33</b></p> <p>Нажмите ПКМ на системную папку <b>Device</b> и выберите <b>Настройки соединения</b></p>
---	---

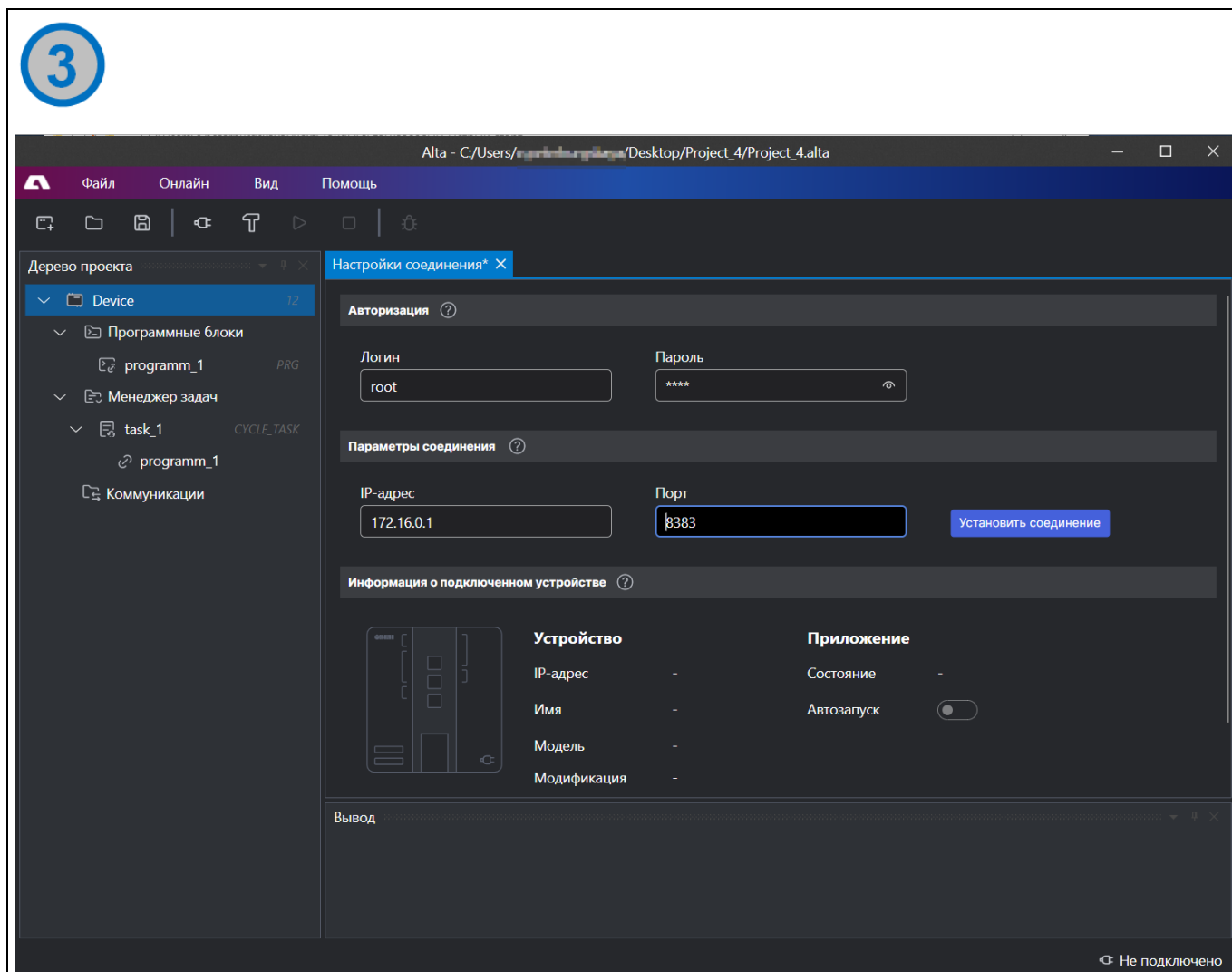


Рисунок 1.34

В открывшейся вкладке **Настройки соединения** авторизуйтесь и введите параметры соединения.

Настройки по умолчанию для ПЛК210:

Логин: **root**

Пароль: **owen**

IP-адрес: **172.16.0.1**

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

4

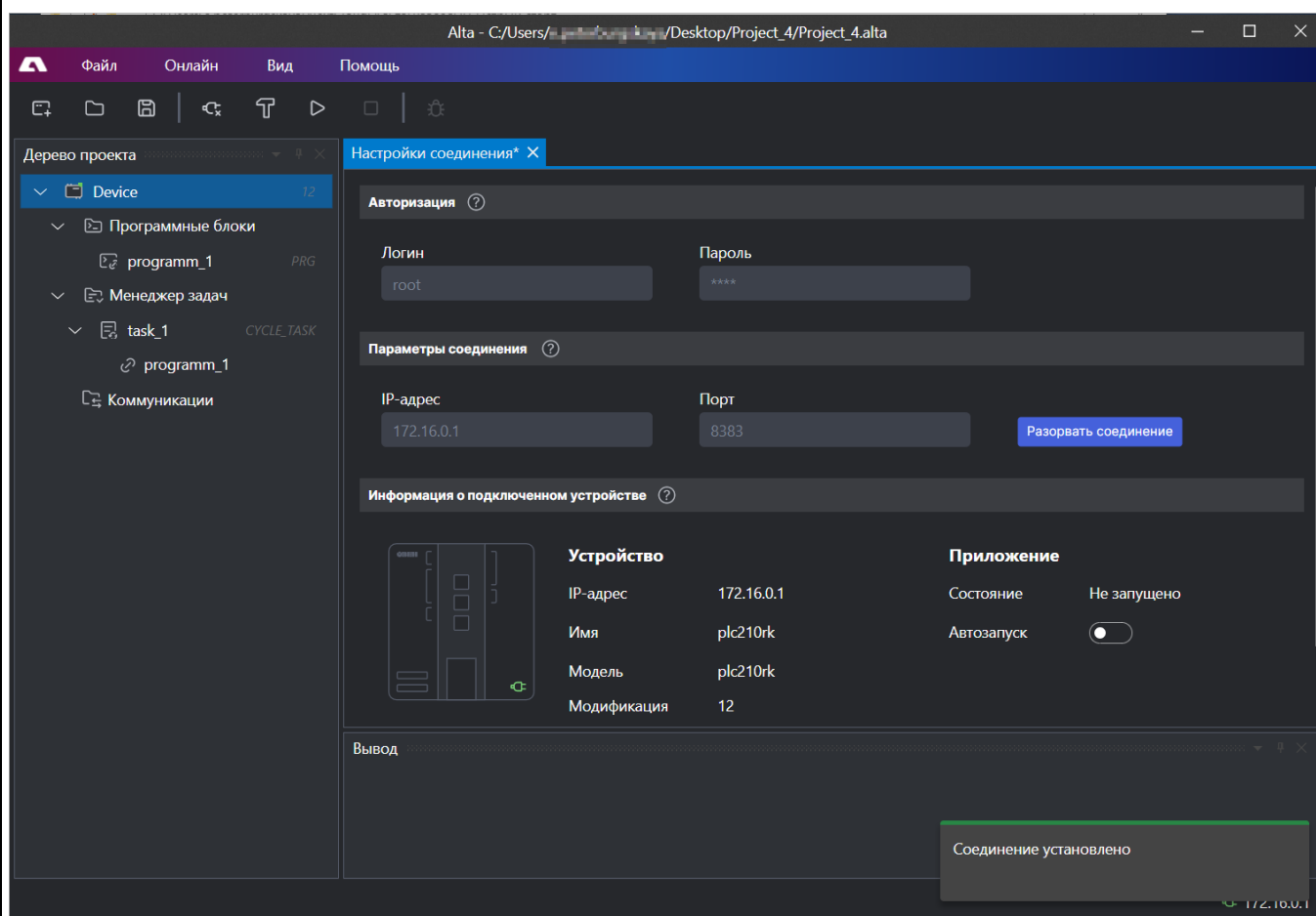
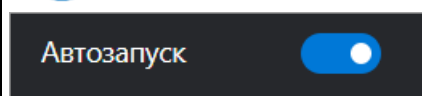


Рисунок 1.35

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



Включите автозапуск приложения на устройстве для автоматического запуска исполняемого приложения (например, после перезагрузки ПЛК)

## 1.7 Сборка и загрузка проекта

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

1

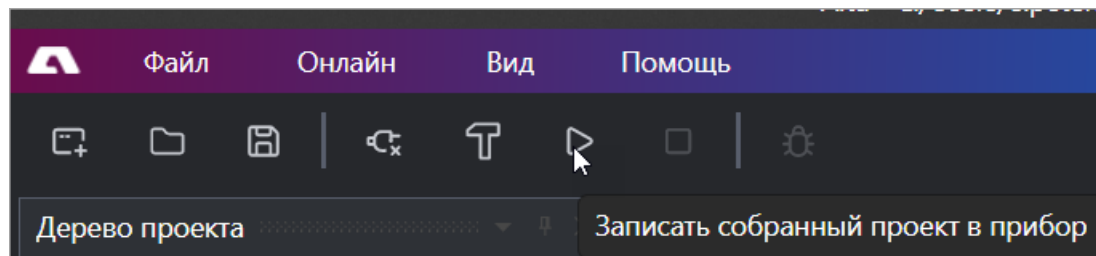


Рисунок 1.36

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

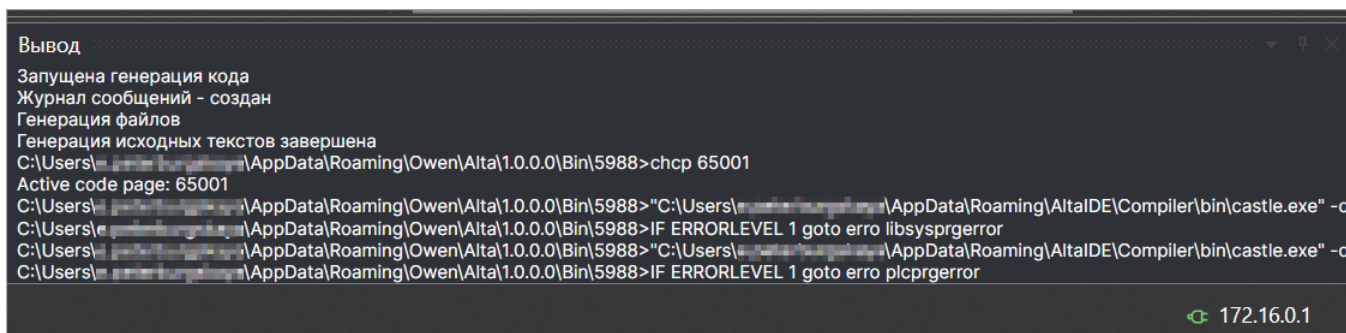


Рисунок 1.37

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

3

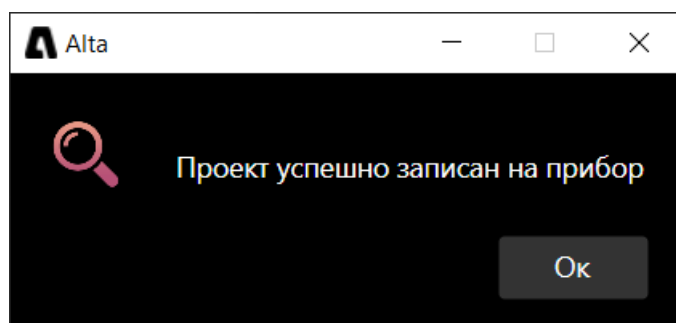
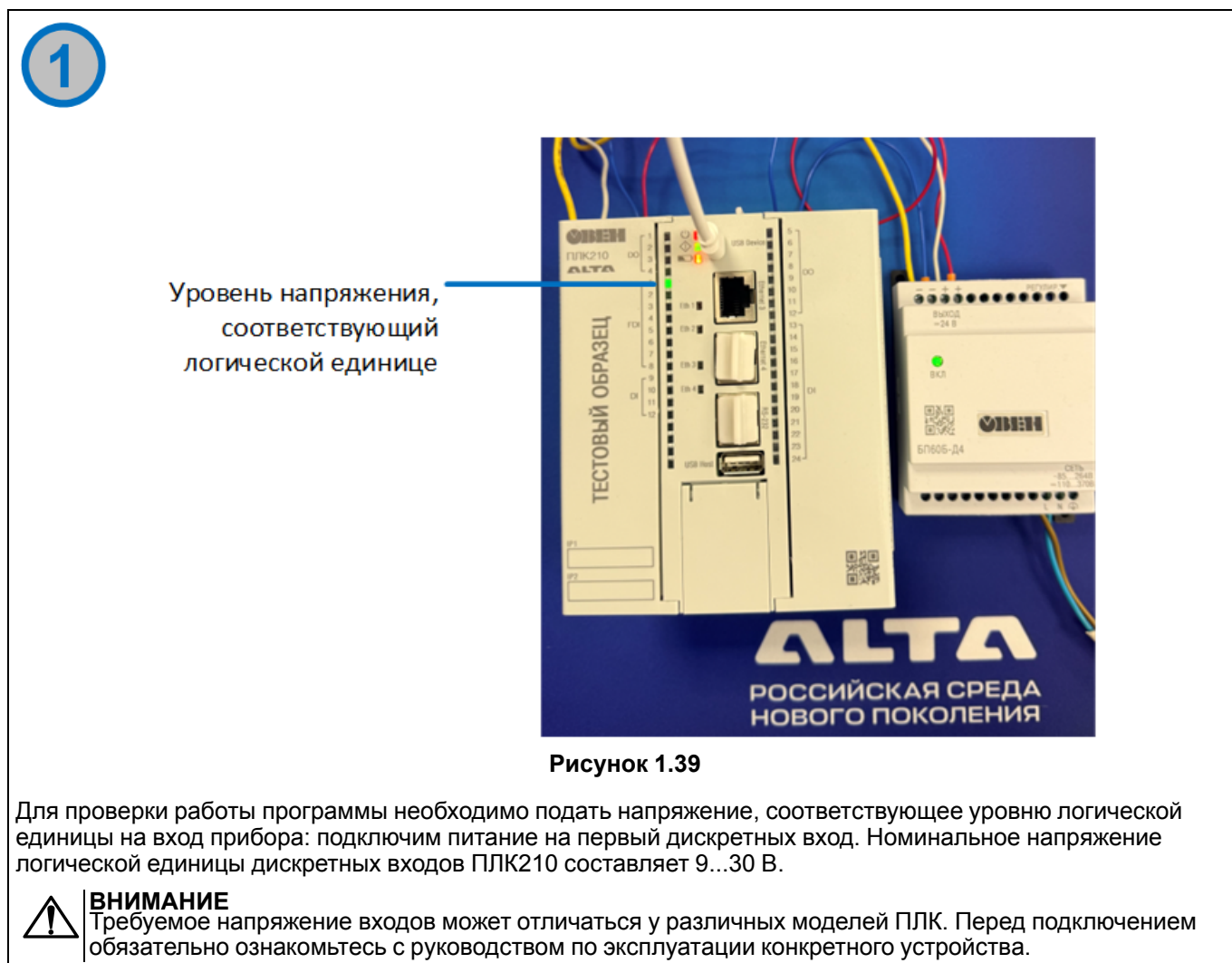


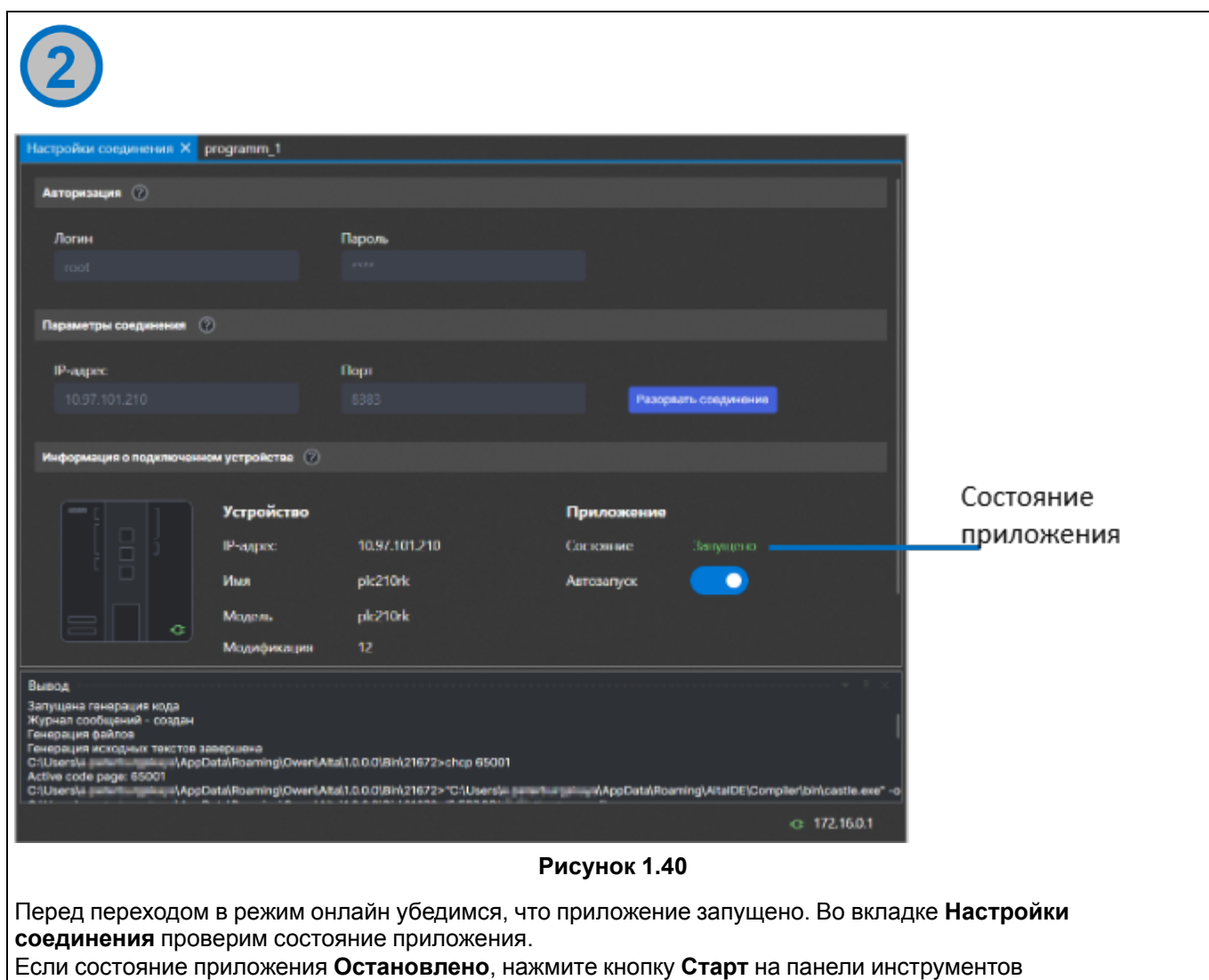
Рисунок 1.38

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.





3

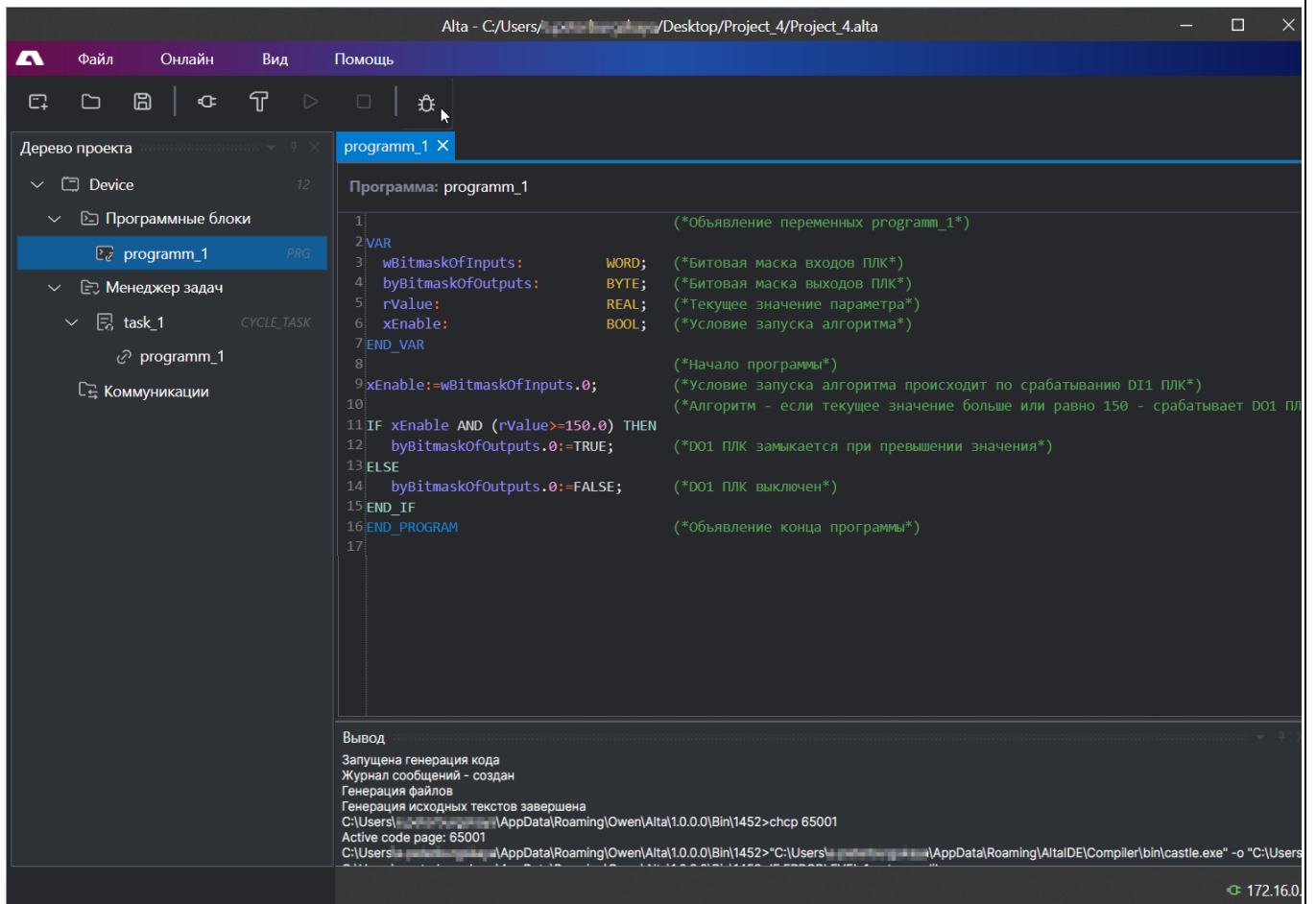


Рисунок 1.41

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

4

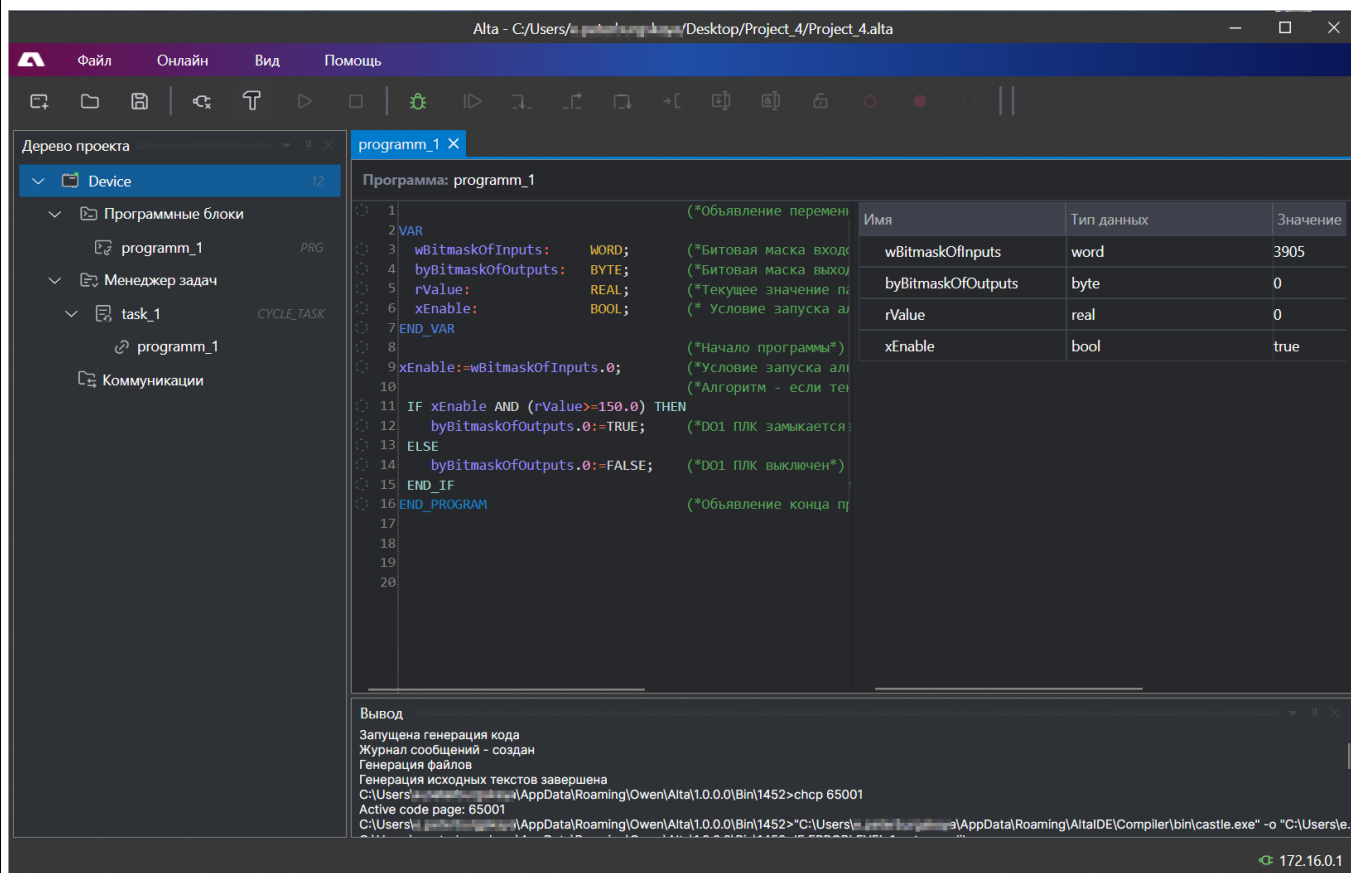


Рисунок 1.42

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

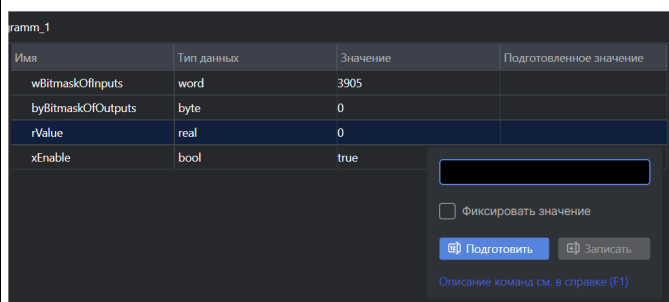


Рисунок 1.43

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

6

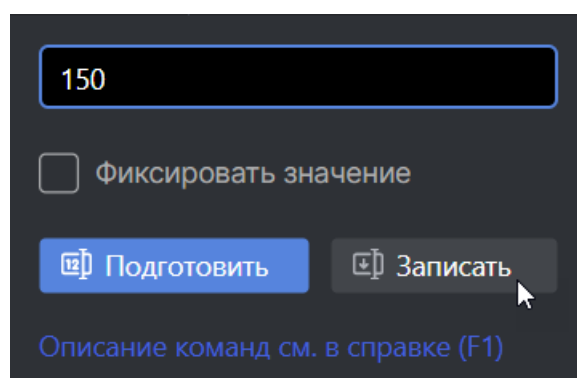


Рисунок 1.44

Введите значение 150, при достижении которого, по условию программы, должен замкнуться выход 1. Нажмите **Записать**

7

Имя	Тип данных	Значение	Подготовл
wBitmaskOfInputs	word	3905	
byBitmaskOfOutputs	byte	1	
rValue	real	150	
xEnable	bool	true	

Рисунок 1.45

Переменная приняла записанное значение

8

Выход 1  
замкнулся  
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](mailto:support@owendigital.ru)

[www.owendigital.ru](http://www.owendigital.ru)

рег.:1-RU-154391-1.5