



ЦИФРОВЫЕ
РЕШЕНИЯ

ALTA IDE



Адаптация проектов Codesys V2.3

09.2026
версия 1.1

Содержание

1 Цель документа	3
2 Правила адаптации	4
2.1 Глобальные переменные	4
2.2 Программные объекты (PRG, FB, FUN).....	5
2.3 Переменные, типы данных	10
2.4 Библиотеки	10
2.5 Задачи	11

1 Цель документа

Настоящее руководство представляет собой сборник рекомендаций по ручной адаптации готового проекта из CODESYS V2.3 в ALTA IDE, с учетом особенностей нашей среды.

В планах развития ALTA IDE инструмент автоматической миграции проектов из CODESYS. Если у вас есть идеи, как улучшить это руководство, или предложения по будущему инструменту автоматической миграции, напишите нам. Если после прочтения у вас останутся вопросы обратитесь в техническую поддержку: support@owendigital.ru

Все документы доступны на сайте компании [ОВЕН Цифровые решения](#).

2 Правила адаптации

Ниже описан пошаговый порядок адаптации проекта из CODESYS V2.3 в ALTA IDE. Рекомендуется соблюдать последовательность переноса программных объектов и настройки проекта.

2.1 Глобальные переменные

Адаптацию проекта рекомендуется начинать с переноса списка глобальных переменных. При последующем переносе программных объектов (PRG, FB, FUN), содержащих обращение к листу глобальных переменных, связь образуется сразу.

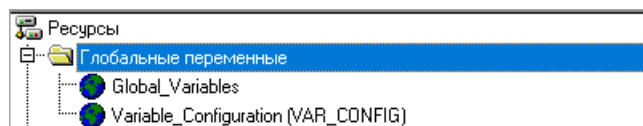


Рисунок 2.1 – Глобальные переменные в CODESYS V2.3

VAR_GLOBAL можно экспортировать аналогично программным объектам (описано [ниже](#)) и вставить после ключевого слова конца программного объекта ALTA IDE.

Создадим программу, в которой будут находиться только глобальные переменные и для удобства назовем ее GVL (Global Variables List).

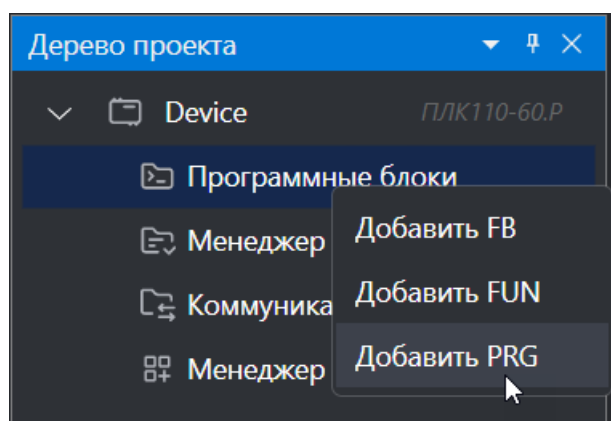


Рисунок 2.2 – Создание программы для объявления GVL

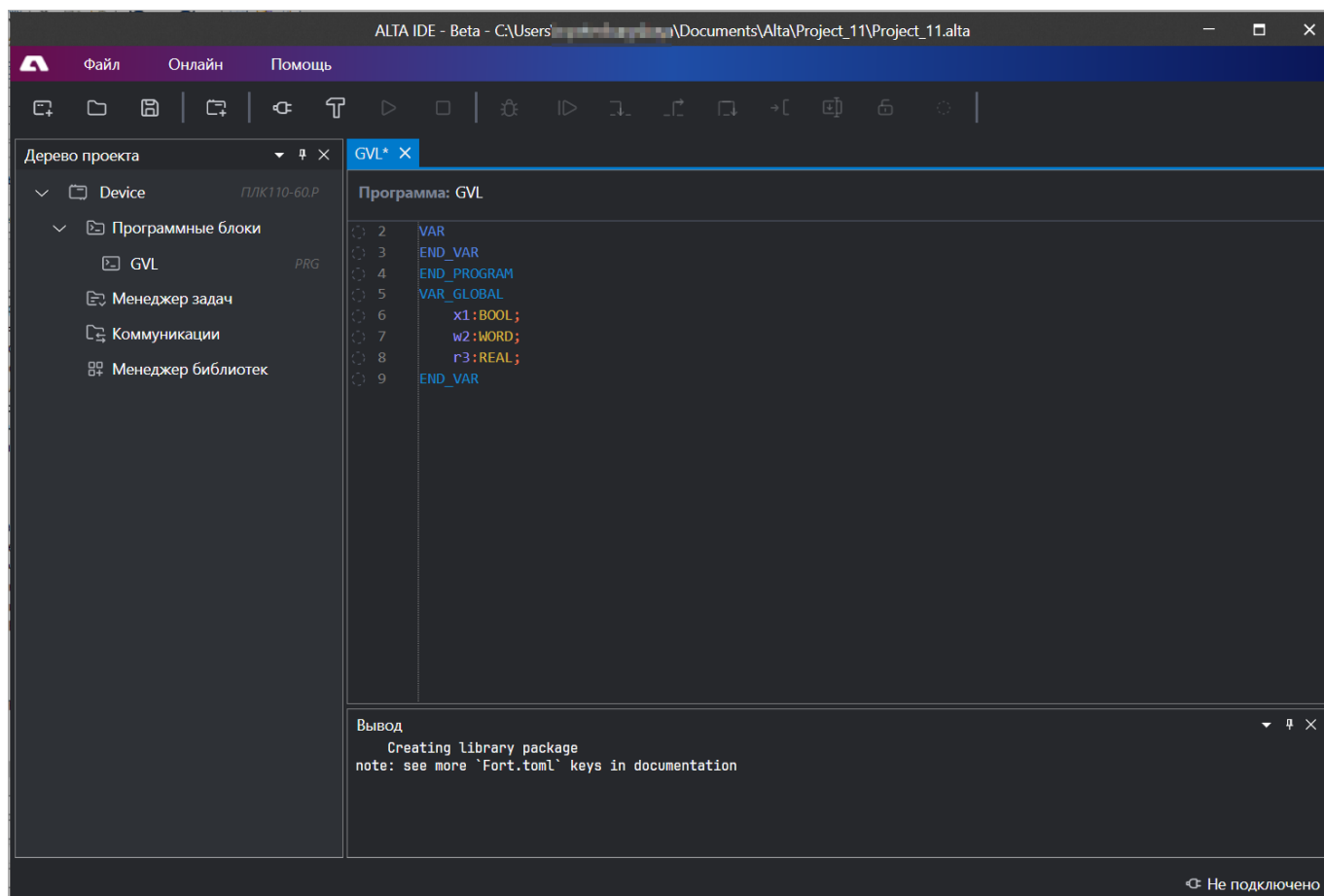


Рисунок 2.3 – Список глобальных переменных ALTA IDE внутри PRG

Перенос **VAR_CONFIG** не поддерживается, т.к. используются для прямой адресации* переменных экземпляров ФБ.

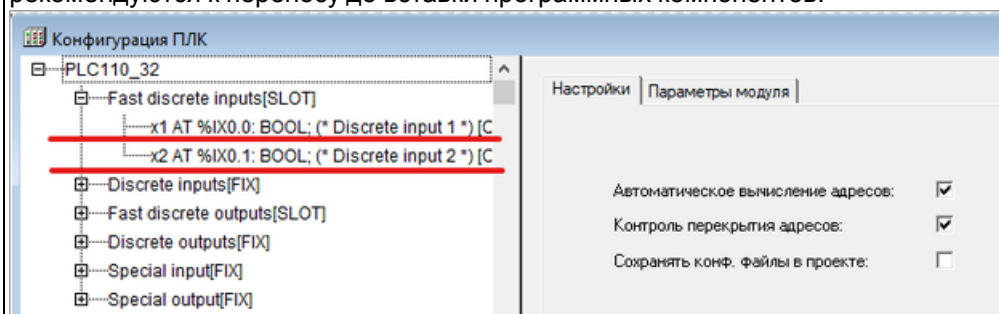
* Прямая адресация к памяти не поддерживается в ALTA IDE.

**ПРИМЕЧАНИЕ**

В будущих версиях ALTA IDE компонент GVL (глобальные переменные) будет вынесен как отдельный объект в дерево проекта.

**ПРИМЕЧАНИЕ**

Переменные, привязанные к целевому файлу в проекте CODESYS V2.3, также являются глобальными и рекомендуются к переносу до вставки программных компонентов.



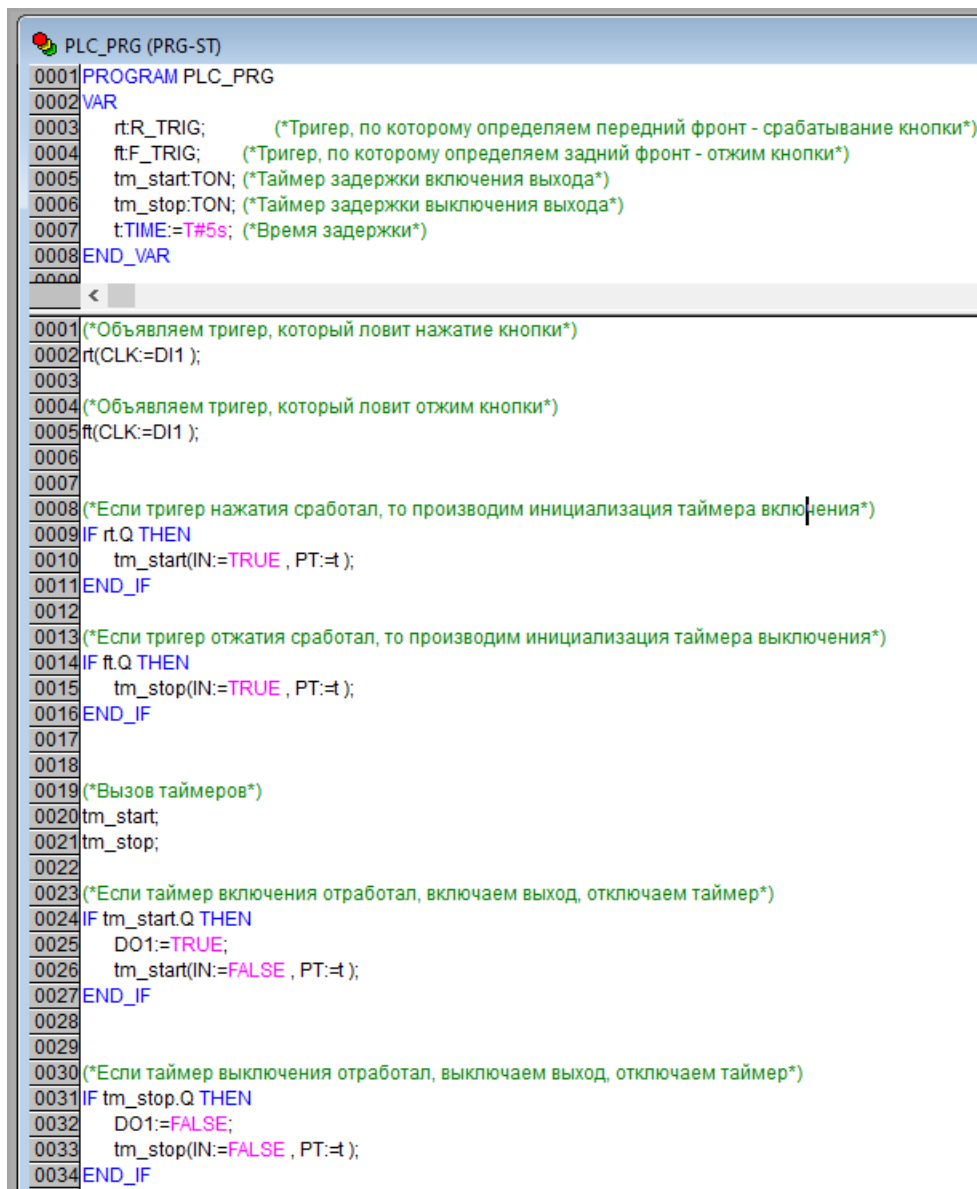
2.2 Программные объекты (PRG, FB, FUN)

**ВНИМАНИЕ**

Таблица 2.1 – Ограничения миграции

Язык проекта	Если проект написан на любом языке, кроме ST, его необходимо полностью переписать на ST (или на CFC, с момента поддержки в ALTA IDE)
Прямая адресация	Если проект использует прямую адресацию AT к ячейкам памяти ПЛК, его необходимо переработать
Псевдонимы (alias), диапазоны	Если проект активно использует псевдонимы типов или ограничение диапазона, необходимо проверить работоспособность после переноса

Исходная программа PLC_PRG Codesys V2.3:



```

0001 PROGRAM PLC_PRG
0002 VAR
0003   rtR_TRIG;      (*Тригер, по которому определяем передний фронт - срабатывание кнопки*)
0004   ftF_TRIG;      (*Тригер, по которому определяем задний фронт - отжим кнопки*)
0005   tm_start:TON;  (*Таймер задержки включения выхода*)
0006   tm_stop:TON;   (*Таймер задержки выключения выхода*)
0007   t:TIME:=T#5s;  (*Время задержки*)
0008 END_VAR
0009
0001 (*Объявляем тригер, который ловит нажатие кнопки*)
0002 rt(CLK:=DI1 );
0003
0004 (*Объявляем тригер, который ловит отжим кнопки*)
0005 ft(CLK:=DI1 );
0006
0007
0008 (*Если тригер нажатия сработал, то производим инициализация таймера включения*)
0009 IF rt.Q THEN
0010   tm_start(IN:=TRUE , PT:=t );
0011 END_IF
0012
0013 (*Если тригер отжатия сработал, то производим инициализация таймера выключения*)
0014 IF ft.Q THEN
0015   tm_stop(IN:=TRUE , PT:=t );
0016 END_IF
0017
0018
0019 (*Вызов таймеров*)
0020 tm_start;
0021 tm_stop;
0022
0023 (*Если таймер включения отработал, включаем выход, отключаем таймер*)
0024 IF tm_start.Q THEN
0025   DO1:=TRUE;
0026   tm_start(IN:=FALSE , PT:=t );
0027 END_IF
0028
0029
0030 (*Если таймер выключения отработал, выключаем выход, отключаем таймер*)
0031 IF tm_stop.Q THEN
0032   DO1:=FALSE;
0033   tm_stop(IN:=FALSE , PT:=t );
0034 END_IF

```

Рисунок 2.4

Существует 2 способа копирования программы:

- Ctrl+C/Ctrl+V — вставка в редактор ST компонента программа в ALTA IDE;
- Экспорт .EXP файла (подробно рассмотрен ниже).

Процесс экспорта PRG/FB/FUN

1

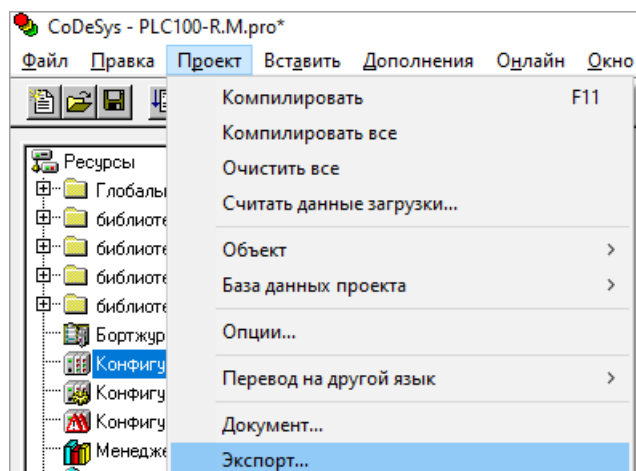


Рисунок 2.5

Выберите Проект -> Экспорт...

2

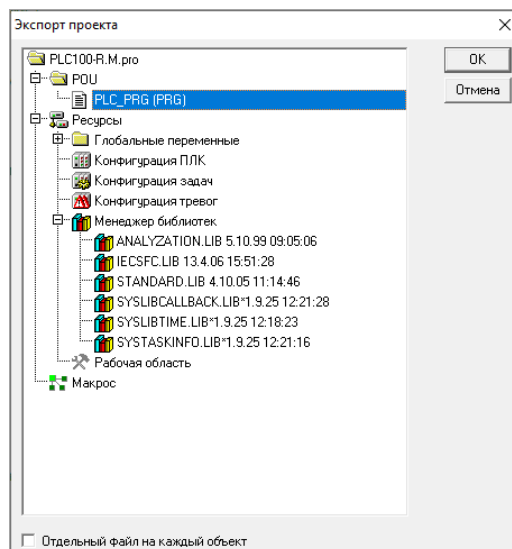


Рисунок 2.6

Выберите необходимый объект (в нашем случае программу)

3

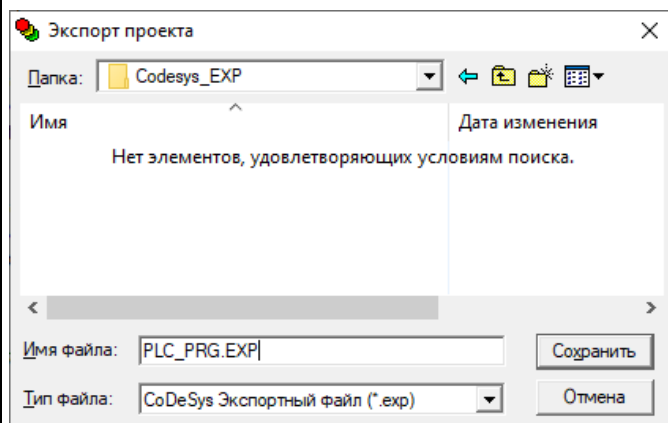
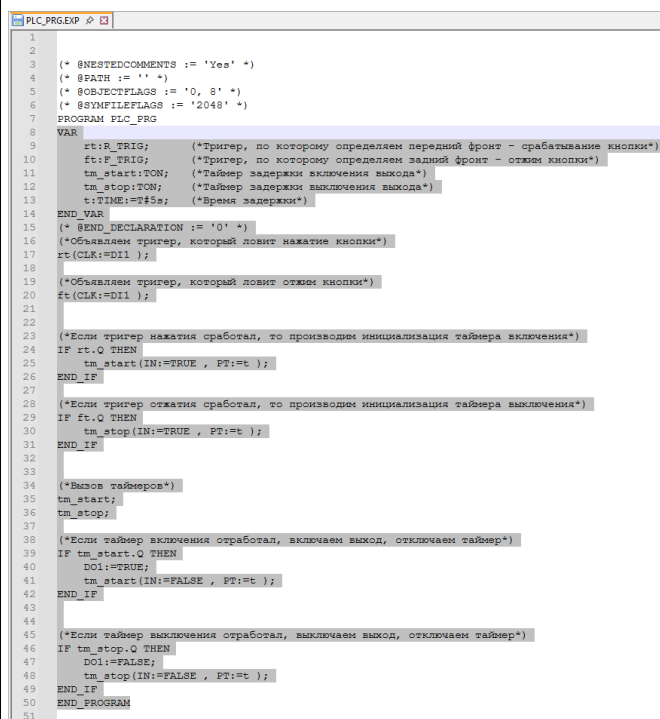


Рисунок 2.7

Введите имя и сохраните файл с расширением .exp

4



Откройте сохраненный файл и скопируйте код

5

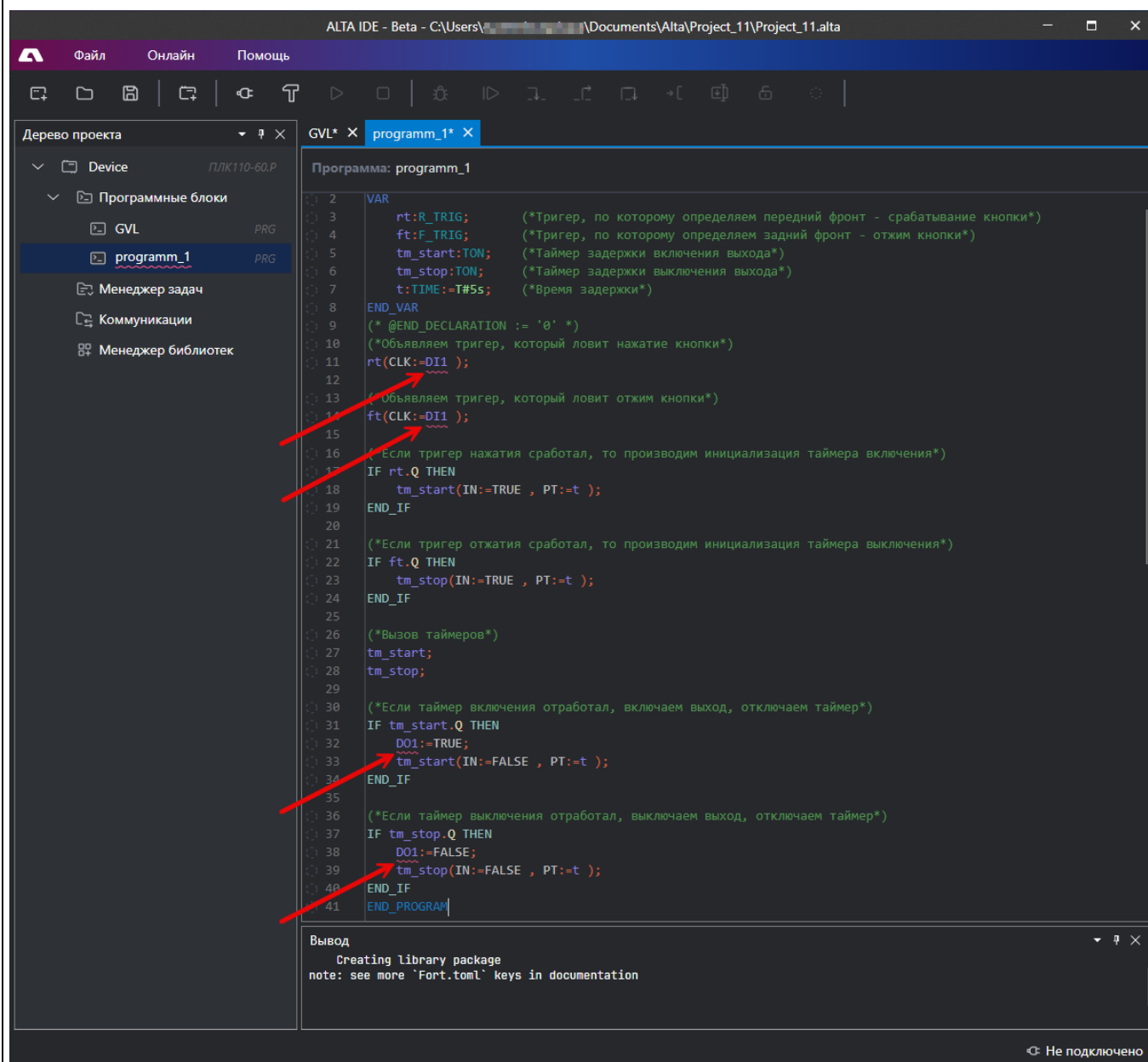


Рисунок 2.9

Создайте в дереве проекта ALTA IDE необходимый программный объект (в нашем случае это программа programm_1).

Вставьте скопированный код в редактор ST, подсветка укажет на места коррекции

6

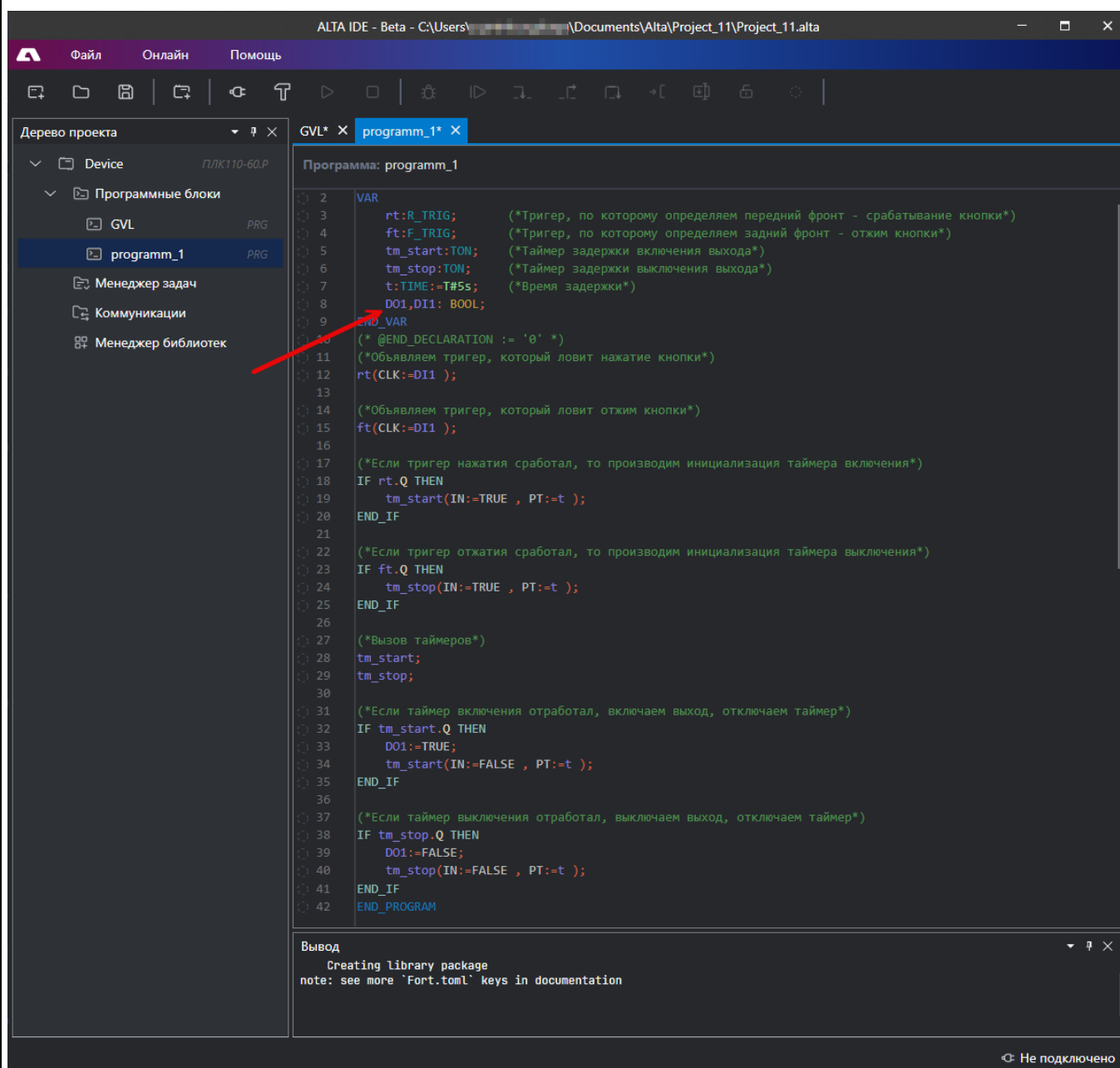


Рисунок 2.10

Скорректируйте код (объявление переменных таргета, проверка синтаксиса)

**ПРИМЕЧАНИЕ**

Рекомендуем пользоваться вариантом через экспорт по следующим причинам:

- При прямом копировании из редактора ST CODESYS V2.3 возможно нарушение кодировки комментариев на русском языке. Чтобы этого избежать, перед копированием переключите раскладку клавиатуры на русский язык.
- В текстовом редакторе объединяются области VAR и кода программы, аналогично редактору ST ALTA IDE.

После вставки кода, необходимо проверить ключевые слова PRG/FB/FUN.

Программа	Функция	Функциональный блок
programm X Программа: programm 1 VAR 2 END_VAR 3 END_PROGRAM	FC_1 X Функция: FC_1 : DINT 1 VAR_INPUT 2 END_VAR 3 VAR 4 END_VAR 5 END_FUNCTION	FB_1 X Функциональный блок: FB_1 1 VAR_INPUT 2 END_VAR 3 VAR_OUTPUT 4 END_VAR 5 VAR 6 END_VAR 7 END_FUNCTION_BLOCK

**ПРИМЕЧАНИЕ**

При переносе программных объектов не торопитесь импортировать все объекты сразу. Вставляйте по одной программе/ФБ/функции, исправляйте, следуя диагностике ошибок (подсветке), и переходите к следующему программному объекту.

2.3 Переменные, типы данных

Переменные, привязанные в таргет-файле в CODESYS V2.3 (Конфигурация ПЛК) следует создать в программе ALTA IDE, соблюдая исходный тип данных. В CODESYS V2.3 при привязке переменной к узлу таргету происходит автоматическое объявление. Дополнительное объявление в программе не требуется. В ALTA IDE требуется сначала объявить переменную с типом данных в программном объекте, а далее привязать в таблицу таргет-файла (мы сделали это в [примере](#) выше)

Основные типы данных в обеих IDE совпадают и соответствуют МЭК61131-3. Но существуют некоторые различия (улучшения) во временных и особых типах данных:

Тип данных / Особенности	ALTA IDE	CODESYS V2.3	Различия
TIME	64 бита, ns, T#213503d23h34m33s709ms551us615-ns	32 бита, ms, максимум T#49d17h2m47s295ms)	+
TIME_OF_DAY (TOD)	32 бита, ms, 4 294 967 295 ms	32 бита, ms, 4 294 967 295 ms	-
DATE	32 бита, ms, битовая маска, D#0000-0-0 - D#9999-12-31	32 бита, D#1970-00-00 - D#2106-02-06	+
DATE_AND_TIME (DT)	64 бита, ms, DT#9999-12-31-23:59:59.999	32 бита, s, DT#1970-00-00-00:00:00 - DT#2106-02-06-06:28:15	+
INSTANT	64 бита, ns, T#213503d23h34m33s709ms551us615-ns	Отсутствует	+
WSTRING, CHAR, WCHAR	Поддерживаются, UCS-2	Отсутствуют	+
STRING	UTF-8	Однобайтовая	+
Указатели	64 бита (ULINT), `REF_TO`/`POINTER TO`	32 бита, `POINTER TO`	+
Структуры (Литералы)	Имена полей обязательны `(x:=1, y:=2)`	Имена полей можно опускать `(1, 2)`	+

2.4 Библиотеки

Основные библиотеки для ALTA IDE аналогичны библиотекам CODESYS V2.3, за исключением минорных исправлений. Более подробно с составом библиотек можно ознакомиться в [документации на конкретную библиотеку](#) ALTA IDE.

Перед вызовом функции или функционального блока из библиотеки в ALTA IDE необходимо указать имя библиотеки через точку (необязательно для библиотеки Standard). Например, вызов функции округления из библиотеки OSCAT_BASIC в коде будет выглядеть так:

```
OSCAT_BASIC.CEIL
```

Важно сохранить регистр именования библиотеки - UTIL.HYSTERESIS вызовет ошибку компиляции, util.HYSTERESIS - нет. Примеры вызова и использования функций и ФБ из библиотек описаны в документации на библиотеки.

Наличие библиотеки	Standard	Util	SysLibCom	SysLibSocket	OSCAT_BASIC	SysLibTime
ALTA IDE	+	+	+	+	+	+
CODESYS V2.3	+	+	+	+	+	+
Вызов функции/ФБ		util.	SysLibCom.	SysLibSocket.	OSCAT_BASIC.	SysLibTime.

2.5 Задачи

На текущий момент в ALTA IDE поддержано только циклическое выполнение задач. Следует учитывать это и при адаптации создавать задачи вручную.

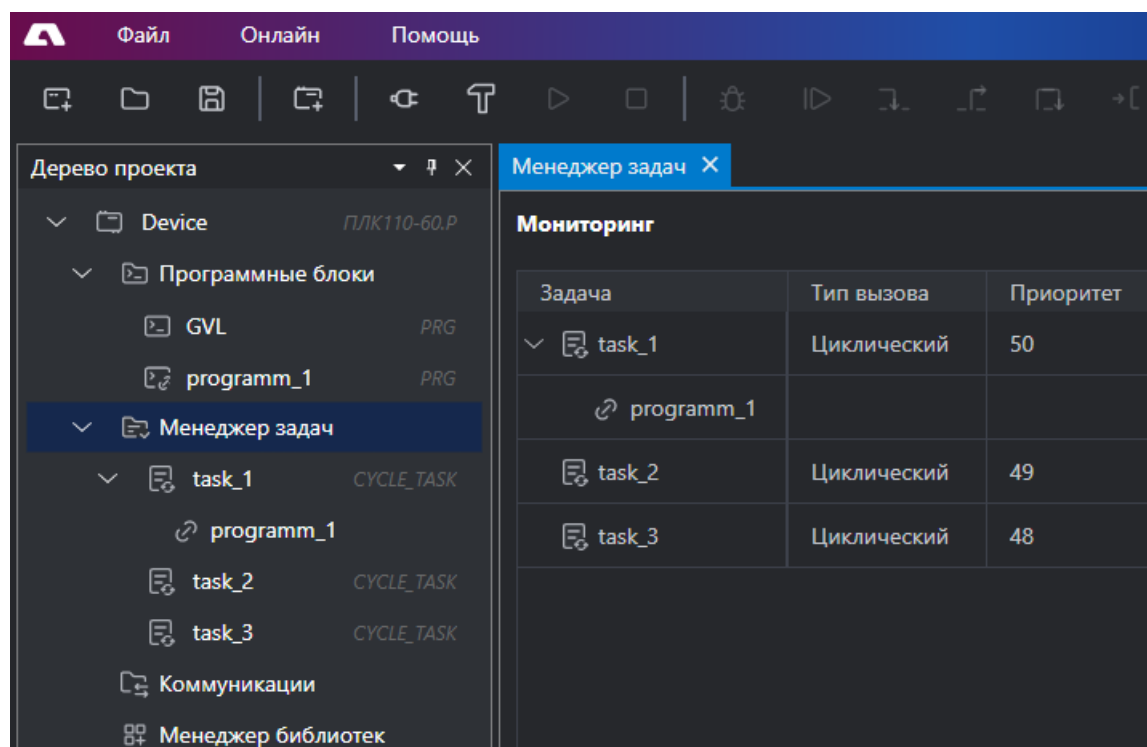


Рисунок 2.11 – Менеджер задач в ALTA IDE



ЦИФРОВЫЕ
РЕШЕНИЯ

ООО "Овен Цифровые решения"

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

support@owendigital.ru

www.owendigital.ru

пер.:1-RU-164160-1.1