

ЗАДАНИЕ №2

ПРОГРАММИРОВАНИЕ ОБРАБОТКИ ДИСКРЕТНЫХ СИГНАЛОВ И УПРАВЛЕНИЕ ИСПОЛНИТЕЛЬНЫМИ МЕХАНИЗМАМИ

1. Общие сведения

Цель работы: научиться работать с дискретными выходами (Output), вызывать функциональные блоки в главной программе.

Результат: программа управления манипулятором, которая обеспечивает автоматический возврат гриппера в исходное положение по команде оператора.

2. Порядок выполнения работы

2.1. Подготовка проекта

1. Загрузить проект Task2.ap16.
2. Запустить среду TIA Portal v16, выбрать скачанный проект Task2 и перейти в главное окно программы, нажав **Project view**.

2.2. Создание функционального блока управления захватом

3. В дереве проекта необходимо перейти в папку **Program blocks** контроллера **PLC_MIDDLE**.
4. Создать новый функциональный блок на языке SCL. Присвоить ему имя **Gripper_to_startPos** (рис. 2.1). Данный блок будет содержать логику возврата механизма в стартовую позицию.

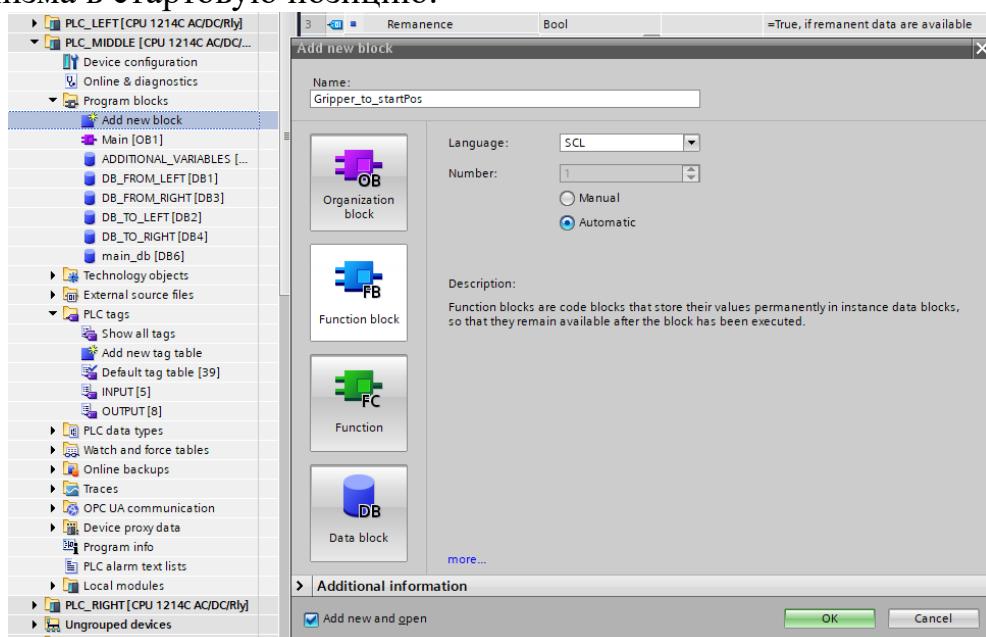


Рис. 2.1 – Создание функционального блока Gripper_to_startPos

2.3. Конфигурирование интерфейса блока

5. В редакторе переменных необходимо обязательно определить входные и выходные параметры, необходимые для управления положением манипулятора и считывания состояний датчиков (рис. 2.2).

	Name	Data type	Default value	Retain	Accessible f...	Writ...	Visi...
	Input						
DI	Gripper_at_start	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
DI	Gripper_at_right	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
DI	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	
	Output						
DO	Gripper_to_right	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
DO	Gripper_to_left	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
DO	Gripper_to_down	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
DO	Gripper_to_open	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Рис. 2.2 – Таблица переменных блока управления захватом

2.4. Интеграция блока в главную программу

6. Откройте организационный блок **Main**.

7. В сети (Network), отвечающей за обработку кнопок «Стоп» и «Старт», нужно добавить вызов блока **Gripper_to_startPos** (рис. 2.3).

```
2      // Кнопка стоп
3  IF "ADDITIONAL_VARIABLES".Stop THEN
4      "Gripper_to_startPos"
5
6 END_IF;
7      // Кнопка старт
8 IF "ADDITIONAL_VARIABLES".Start THEN
9
10 END_IF;
```

Рис. 2.3 – Вызов функционального блока в главной программе

8. При добавлении блока TIA Portal автоматически запросит создание экземпляра блока данных (рис. 2.4). Подтвердить создание, нажав OK.

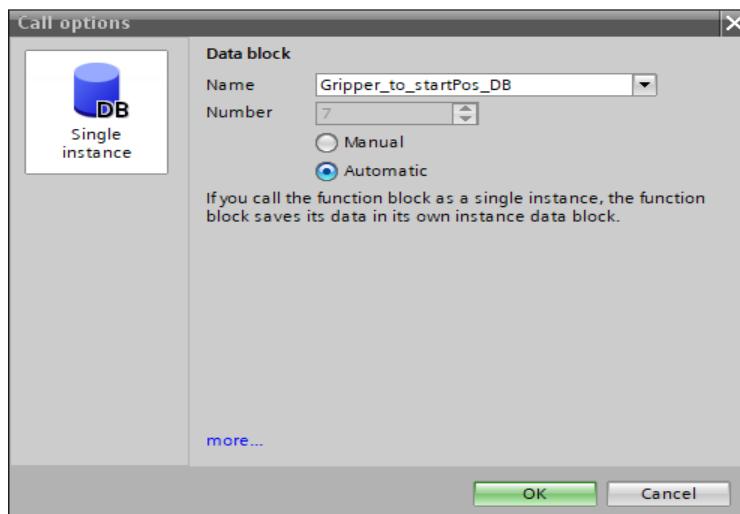


Рис. 2.4 – Создание Data block

2.5. Привязка переменных к физическим адресам

9. Присваиваем нужные переменные к физическим входам/выходам контроллера (рис. 2.5). Физические адреса переменных можно посмотреть в таблице тегов PLC tags → Default tag table соответствующего контроллера.

The screenshot shows a PLC programming interface. On the left, there is a ladder logic diagram with several contacts and coils. A specific coil is highlighted in yellow. On the right, there is a table titled "Default tag table" showing the physical addresses for various variables. The table has columns for variable name, type, and address. One row in the table is also highlighted in yellow, corresponding to the selected coil in the ladder logic.

Variable Name	Type	Address
"handling_input_0_workpiece..."	Bool	%I8.0
"handling_input_1_gripper..."	Bool	%I8.1
"handling_input_2_gripper_at_start"	Bool	%I8.2
"handling_input_3_gripper..."	Bool	%I8.3
"handling_input_4_gripper..."	Bool	%I8.4
"handling_output_0_to_gree..."	Bool	%Q8.0
"handling_output_1_to_yello..."	Bool	%Q8.1
"handling_output_2_to_red"	Bool	%Q8.2

Рис. 2.5 – Таблица тегов контроллера PLC_MIDDLE

В рамках данной работы используются только переменные контроллера PLC_MIDDLE, левый и правый контроллеры не задействованы.

Важно! Будьте внимательны и осторожны при запуске программы. Некорректная логика может привести к механическим повреждениям стенда, например, при движении в горизонтальной плоскости при опущенном гриппере.