Владимирский государственный университет

М. С. ДЕНИСОВ И. В. РУМЯНЦЕВ П. А. ЧЕБОТАРЕВ

КОМПЬЮТЕРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

Лабораторный практикум

Владимир 2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

М. С. ДЕНИСОВ И. В. РУМЯНЦЕВ П. А. ЧЕБОТАРЕВ

КОМПЬЮТЕРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

Лабораторный практикум

Электронное издание



ISBN 978-5-9984-1976-8 © ВлГУ, 2024 УДК 004.896 ББК 32.966

Авторы: М. С. Денисов (введение, лабораторные работы № 3 – 5, заключение, рекомендательный библиографический список), И. В. Румянцев (лабораторная работа № 2), П. А. Чеботарев (п. 1 – 3, лабораторная работа № 1)

Рецензенты:

Кандидат технических наук, доцент доцент кафедры технологии машиностроения Владимирского государственного университета имени Александра Григорьевича и Николая Григорьевича Столетовых *А. В. Жданов*

Кандидат технических наук, доцент доцент кафедры автоматизированных технологических систем Брянского государственного технического университета *В. А. Хандожко*

Издается по решению редакционно-издательского совета ВлГУ

Денисов, М. С. Компьютерные системы управления [Электронный ресурс] : лаб. практикум / М. С. Денисов, И. В. Румянцев, П. А. Чеботарев ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2024. – 132 с. – ISBN 978-5-9984-1976-8. – Электрон. дан. (6,01 Мб). – 1 электрон. опт. диск (CD-ROM). – Систем. требования: Intel от 1,3 ГГц ; Windows XP/7/8/10 ; Adobe Reader ; дисковод CD-ROM. – Загл. с титул. экрана.

Посвящен изучению учебного стенда по автоматизации, собранного на основе современных высокопроизводительных комплектующих китайской фирмы Inovance.

Предназначен для студентов направления подготовки 15.03.04 «Автоматизация технологических процессов и производств» очной и заочной форм обучения.

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Ил. 180. Табл. 6. Библиогр.: 8 назв.

ISBN 978-5-9984-1976-8

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ
1. УЧЕБНЫЙ СТЕНД ПО АВТОМАТИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ ПРОПЕССОВ INOVANCE
Контрольные вопросы
2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ КОНТРОЛЛЕРОВ
INOVAINCE
3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПАНЕЛЕЙ INOVANCE
Контрольные вопросы 50
4. ПРАКТИКУМ
Лабораторная работа № 1. ЗАПУСК ПРОЕКТА И ПЕРЕДАЧА ДАННЫХ В ПЛК
Лабораторная работа № 2. РАБОТА С ДИСКРЕТНЫМИ ВХОДАМИ/ВЫХОДАМИ
Лабораторная работа № 3. ЗАПУСК ДВИГАТЕЛЯ ПО СКОРОСТИ
Лабораторная работа № 4. СЕРВОПРИВОДЫ С АБСОЛЮТНЫМ УПРАВЛЕНИЕМ
Лабораторная работа № 5. УПРАВЛЕНИЕ СЕРВОПРИВОДАМИ С ПОТЕНЦИОМЕТРА И ПАНЕЛИ100
ЗАКЛЮЧЕНИЕ130
РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК 13

ВВЕДЕНИЕ

В течение последних десяти лет наблюдается активный переход на компьютерные системы управления на предприятиях и производствах. Промышленные организации в условиях высокой конкурентной борьбы на рынке ищут более эффективные способы для быстрой адаптации к современным производственным тенденциям и пути снижения себестоимости производимой продукции. Использование компьютерных систем управления напрямую способствует достижению поставленных целей.

Компьютерные системы вырабатывают управляющие воздействия, которые поступают на исполнительные механизмы благодаря цифровым и аналоговым преобразователям. Компьютерные системы управления имеют ряд преимуществ относительно других систем управления. Во-первых, в них легко учитывается и компенсируется нелинейность характеристик датчиков, преобразователей, усилителей, а также исполнительных механизмов. Во-вторых, компьютерные системы управления адаптивны и оптимальны, поэтому могут быть реализованы только на основе электронно-вычислительной машины. В-третьих, электронно-вычислительная машина представляет собой универсальную систему, а ее переналадка на другую систему управления заключается только в изменении коэффициентов или загрузке новой программы. При использовании аналоговых систем может возникнуть необходимость в полном перепрограммировании, а следовательно, и в разработке новой конструкторской и технологической документации.

Если рассматривать операционные системы реального времени, то следует помнить о том, что они описываются специальным стандартом. Проблема управления процессом в режиме реального времени заключается в аппаратном аспекте. Для того чтобы создать многозадачную операционную систему реального времени, необходимо использовать электронно-вычислительные машины в масштабах цеха или предприятия. Таким образом, применение компьютерных систем управления для автоматизации технологических процессов – один из самых удобных и выгодных вариантов.

Лабораторный практикум предназначен для изучения принципов компьютерных систем управления, а также обучения программированию алгоритмов технологических процессов и интерфейсов. В теоретической части на примере стенда Inovance описывается возможная аппаратная часть для компьютерных систем управления, а также среды для программирования контроллеров и панелей.

Представленные лабораторные работы нацелены на изучение способов программирования исполнительных устройств и контрольноизмерительных приборов, а также способов программирования интерфейсов и алгоритмов для компьютерных систем управления.

1. УЧЕБНЫЙ СТЕНД ПО АВТОМАТИЗАЦИИ **ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ INOVANCE**

Компания Inovance (Китай) по объёму продаж занимает третье место после Siemens и ABB. Численность сотрудников, работающих в разных уголках мира, – около 18 тыс. человек. Компания производит преобразователи частоты различных серий, панели визуализации, контроллеры и модули ввода/вывода, ЧПУ системы.

На рис. 1.1 представлен стенд Inovance, который включает следующие компоненты:

АМ600 – программируемый логический контроллер (ПЛК)

GR10-EC-6SW – разветвитель сети EtherCAT

GL10-PS2 – блок питания

GL10-1600END/GL10-0016ETP – модули дискретного ввода/

вывода.

Разветвитель сети EtherNET GL10-PS2 – блок питания GL-10-RTU-ECTA – коплер EtherCAT GL10-4AD/GL10-4DA – модули аналогового ввода/вывода Преобразователи частоты SV660N Сервомоторы серии MS1



Рис. 1.1. Стенд Іпочапсе

Назначение стенда: знакомство с продукцией Inovance. Изучение программы для программирования контроллера и панели. Ознакомление с обменом данных: OPC UA, Modbus TCP между панелью и контроллером.

Программируемый логический контроллер

ПЛК – вычислительное устройство, предназначенное для автоматизации технологических процессов, осуществляющее функции автономного сбора, обработки, хранения информации, выработки команд управления (рис. 1.2).



Рис. 1.2. Элементы автоматики Inovance

Стенд включает следующие средства автоматики (см. рис. 1.2): (A4) GR10-EC-6SW – разветвитель сети EtherCAT (UG3) GL10-PS2 – блок питания AM600 – ПЛК (A10-A11)GL10-1600END/GL10-0016ETP – модули дискретного

(A10-A11)GL10-1600END/GL10-0016E1P – модули дискретног ввода/вывода.

АМ600. Контроллер средней производительности, предназначен для связи с объектами управления как через высокоскоростные входы/выходы, расположенные на борту контроллера, так и по цифровой сети EtherCAT. Питание контроллера осуществляется от специализированного блока питания. Характеристики ПЛК:

– микропроцессор ARM-Cortex A8 1 ГГц;

 возможность управления до 32 осей – для двухточечного позиционирования (до 8 – для интерполированного движения);

– работа с математической точностью – IEEE 754 double;

– встроенный порт EtherCAT;

 возможность подключения до 16 дополнительных модулей ввода/вывода;

- языки программирования IEC 61131-3 (CODESYS);

– вход энкодера;

– имитация выходного сигнала энкодера;

– многозадачность.

Поддерживаемые интерфейсы и протоколы передачи данных:

– встроенный Ethernet;

– Modbus TCP/IP;

- OPC UA (сервер);

– интерфейс CANopen/Modbus RTU;

– САМ-функциональность.

(UG3) – блок питания GL10-PS2

Специальный модуль питания, имеет две клеммы. Одна из них является входным портом, другая – выходным.

Напряжение входного порта – 220 В.

Напряжение выходного порта – 1 – 24 В.

Напряжение выходного порта – 2 – 5 В.

Локальные модули, которые подключаются непосредственно на шину контроллера, также запитаны от блока питания. Пример подключения представлен на рис. 1.3.



Рис. 1.3. Подключение блока питания

(A10-A11)GL10-1600END/GL10-0016ETP – модули дискретного ввода/вывода

Взаимодействие ПЛК с исполнительными механизмами, устройствами контроля осуществляется посредством модулей расширения и цифровых интерфейсов, позволяющих определить состояние объектов управления, сформировать внешние управляющие воздействия в соответствии с заложенной логикой.

GL10-1600END – модуль дискретного ввода

Характеристики:

– на борту модуля имеется индикация, соответствующая различным входным сигналам:

индикация ВКЛ: вход активен

индикация ВЫКЛ: вход неактивен;

– 16 входных дискретных каналов;

– режим управления как по плюсу, так и по минусу;

– класс входного напряжения – 24 В постоянного напряжения;

- внутренняя потребляемая мощность (типичная) 5 B, 55 мА;
- входной ток (типичный) 5,3 мА;
- время фильтрации порта и передачи данных в контроллер 10 мс;

– входное сопротивление – 4,3 кОм.

GL10-0016ETP – модуль дискретного вывода

Характеристики:

 на борту модуля имеется индикация, соответствующая различным выходным сигналам:

индикация ВКЛ: вход активен

индикация ВЫКЛ: вход неактивен;

– 16 выходных дискретных каналов;

– режим управления по плюсу;

– источник выходного режима;

– напряжение питания – 24 В постоянного тока;

- внутреннее энергопотребление - 5 B, 65 мА;

– время отклика при включении модуля – менее 0,5 мс (для аппаратного обеспечения).

(A4) GR10-EC-6SW – разветвитель сети EtherCAT

Так как у контроллера имеется только один выход для управления устройствами по EtherCAT, то необходимо использовать разветвитель. Модуль (рис. 1.4) позволяет принимать сигнал с одного устройства и переадресовывать его на один или несколько других портов.

Основные характеристики:

– номинальное рабочее напряжение – 24 В;

– внутренний потребляемый ток – 160 мА;

– протокол связи EtherCAT;

- канал EtherCAT с одним входом и пятью выходами;

– максимальная скорость связи – 100 Мбит/с;

– стандартный сетевой порт с сетевыми кабелями Cat 5e, длина кабеля – не более 100 м;

- рабочая температура - от -10 до +55 °C;

- температура хранения - от -25 до +70 °C;

– влажность – 10 – 95 %, без образования конденсата.

Устройства удаленного доступа подключены к контроллеру по протоколу EtherCAT.



Рис. 1.4. Удаленное подключение модулей ввода/вывода

На рис. 1.4 представлены следующие модули: (A5) – разветвитель сети EtherNET (UG4) GL10-PS2 – блок питания (A20) GL-10-RTU-ECTA – коплер EtherCAT (A21-A22) GL10-4AD/GL10-4DA – модули аналогового ввода/вывода

(A20) GL-10-RTU-ECTA – коплер EtherCAT

Модуль предназначен для удаленного подключения с помощью протокола EtherCAT к контроллеру модулей ввода/вывода.

Основные характеристики:

- источник питания 24 В;
- протокол связи EtherCAT;
- максимальная скорость связи Ethernet 100 Мбит/с;

– сетевой интерфейс: стандартный интерфейс Ethernet (сетевой кабель повышенной категории 5 с длиной кабеля не более 100 м);

– диапазон номеров станций – от 1 до 125, внутренний адрес автоматически упорядочивается в последовательности подключения к сетевой шине;

– может быть подключено до 16 модулей ввода/вывода. Фактическое количество и конфигурация зависят от потребляемой мощности каждого модуля.

(A21-A22)GL10-4AD/GL10-4DA – модули аналогового ввода/вывода

GL10-4AD – модуль аналогового ввода

Основные характеристики:

– четыре входных канала;

– напряжение питания – 24 В;

– потребляемая внутренняя мощность – 5 В, 85 мА (типичное значение);

- входное сопротивление напряжения -> 1 кОм;

- сопротивление выборки тока - 250 Ом;

-диапазон входного тока – от 0 до 20 мА, от 4 до 20 мА, ±20 мА, ±10 В, от 0 до 10 В;

- время выборки – 1 мс;

- пределы тока $-\pm 30$ мА (переходный), ± 24 мА (средний);

- системная программа обновляется через USB-интерфейс;

– защита от короткого замыкания на выходе.

GL10-4DA – модуль аналогового вывода

– четыре выходных канала;

– напряжение питания – 24 B;

– потребляемая внутренняя мощность – 5 В, 85 мА (типичное значение);

[–] пределы напряжения – ±15 В;

- выходное напряжение нагрузки от 1 кОм до 1 МОм;
- диапазон выходного напряжения:
 биполярный: ±5 B, ±10 B;
 однополярный: +5 B, +10 B;
- диапазон выходного тока от 4 до 20 мA, от 0 до 20 мA;
- защита от короткого замыкания на выходе;
- системная программа обновляется через USB-интерфейс.

Преобразователи частоты SV660N

Пропускная способность токовой петли SV660N 3 кГц позволяет двигателю следовать профилю движения с минимальной погрешно-



Рис. 1.5. Преобразователи частоты SV660N

стью.

Высокая скорость передачи данных EtherCAT для устройства составляет 125 мкс.

В устройствах серии SV660N (рис. 1.5) используется высокопроизводительный процессор для высокоскоростной связи, обеспечивающий время цикла 125 мкс для всех режимов работы EtherCAT. Доступны семь режимов работы профиля EtherCAT CiA402 (CoE):

– режим положения профиля (PP);

– режим скорости профиля (PV);

 – режим крутящего момента профиля (PT);

– режим самонаведения (HM);

– циклический синхронный позиционный режим (CSP);

– режим циклической синхронной скорости (CSV);

– режим циклического синхронного крутящего момента (CST).

23-битный последовательный однооборотный/многооборотный абсолютный энкодер с обратной связью высокого разрешения выдает

8 388 608 импульсов за один механический оборот. Информация о нескольких оборотах также может сохраняться при отключении питания, что позволяет избежать необходимости выполнять автонастройку частотного преобразователя при каждом включении питания.

Общие характеристики:

– управление по протоколу EtherCAT;

– режим управления: IGBT-ШИМ-управление, режим возбуждения синусоидальным током 220 В, однофазное;

– 23-разрядный абсолютный энкодер, который может использоваться в качестве инкрементного энкодера при отсутствии батареи;

- пыле-, влагозащищенность - IP20;

– пропускная способность контура скорости – 3 кГц;

– время плавного запуска – до 65 с (ускорение и замедление можно установить отдельно).

Сервомоторы MS1 (рис. 1.6) – серводвигатели последнего поколения, разработанные Inovance. Серводвигатели серии MS1 имеют диапазон мощности от 30 Вт до 7,5 кВт, а размеры фланцев варьируются от 25 до 180 мм. Серводвигатели указанной серии служат для обеспечения быстрого и точного контроля положения, скорости и крутящего момента в оборудовании автоматизации – полупроводниковых приборах, SMT-машинах, машинах для штамповки печатных плат, погрузочно-разгрузочных машинах, механических инструментах и трансмиссионных механизмах.



Рис. 1.6. Сервомотор MS1

Основные характеристики:

- напряжение - 220 B;

– мощность – 400 Вт;

- номинальный ток - 2,8 A;

максимальное количество оборотов в минуту – 3000;

– режим работы – непрерывный;

- уровень вибрации - V15;

– сопротивление изоляции – 500 В постоянного тока, более 10 Мом;

- температура окружающей среды - от 0 до +40 °С;

- температура хранения - от -20 до +60° С (максимальная температура – +80 °С в течение 72 ч);

– режим возбуждения – постоянный магнитный;

способ монтажа – фланец;

- уровень термостойкости уровень F;
- класс защиты корпуса IP67;
- влажность окружающей среды 20 80 % (без конденсата);

- вибрация - ниже 49 м/с².

Панель визуализации ІТ7070

Общие характеристики (рис. 1.7):

– центральный процессор Cortex A8 600 МГц;

- оперативная память - 128 Мб DDR3;

- Flash-память 128 Мб;
- слот для SD карты 1;

- последовательный порт COM1 (RS422/RS485), COM2 (RS232), COM3 (RS485);

- Ethernet $\Pi \text{opt} - 1$;

- Mini USB type B порт -1;

- USB type A порт 1;
- входной ток 250 мА;
- диагональ экрана 7^{••};
- разрешение 800×480;

– яркость – 350;

- цвет дисплея True Color;
- тип отображения LED;
- срок службы подсветки 35 000 ч;
- рабочая температура от -10 до 55 °C;
- рабочая влажность -10 90 %;
- есть возможность написать скрипты на языке JavaScript;
- программное обеспечение бесплатное;

 визуализация может быть адаптирована, за счет стилизации можно накладывать различные стили на все компоненты. Стилистика может применяться сразу ко всему проекту;

– связь с контроллерами происходит по ОРС UA, Modbus TCP и через последовательные интерфейсы.

	INOVANCE	 Initialization is 16:57:20 2023-05 	complete -15	Product	Function	🚖 Plan	E About	<mark>е</mark> м	
	ta Tarana Tarana Tarana		IT7000 se	ries	High-po	erforma	ince	Product	
					Using L	inux, Al	ndroid	Product fam ¹¹ Feature	2
1	te en el	5 The cu	istom styl	IC remote	desktop	<u>}</u> - •		**************************************	
		Vector	ionnat ice	JNScripti		HOALCS	representation de la constante la constante de la constante de la constante de la constante de		

Рис. 1.7. Панель визуализации ІТ7070

Рассмотрим схему подключения всех устройств стенда и принципиальные электрические схемы (рис. 1.8 – 1.16).



Рис. 1.8. Кабельное подключение периферии

S	1		2	3	4	5	6	7	8	9	10
	1 Соде 1. 2. 3. 4. 5. 6. 7. 8.	ржание Содержа Ввод пита Питание Подключи Модули п Подключи Цифрова	2 ние ания, питание АС220 модулей расширения модулей расширения ение аналоговых сигн ение атурвала WSA я сеть	3 В а, выходное питание 2 а налов к модулю А16 еключатели), выходое	4 4В и 5В, подключение в (реле)	5	6	7	8	9	10
Né подл. Подлись и дата Взамен Ине. Ne Ине. Ne подл. Подлись и дата	7. 8.	Подключи Цифрова	ение штурвала WSA я сеть				Изм. Лист N Разраб. Го Проверил Т.хонтр.	2 докум. Подпись Дата 108КИН 66 07 2	СТДА.42 Стенд Inovance №3	1453.22030 (ПЛК АМ600)	ЭЗ я <u>Масса Масштаб</u> 1 Листов 8
Z							Утвердил		1		Формат АЗ

Рис. 1.9. Первая страница электрической принципиальной схемы



Рис. 1.10. Вторая страница электрической принципиальной схемы



Рис. 1.11. Третья страница электрической принципиальной схемы



Рис. 1.12. Четвертая страница электрической принципиальной схемы



Рис. 1.13. Пятая страница электрической принципиальной схемы



Рис. 1.14. Шестая страница электрической принципиальной схемы



Рис. 1.15. Седьмая страница электрической принципиальной схемы



Рис. 1.16. Восьмая страница электрической принципиальной схемы

Как видно из рис. 1.10, питание 220 В в систему поступает на клемму X1. Затем необходимо включить 20 А автомат QF1 для замыкания электромагнитного контактора.

Аварийный выключатель SB находится в закрытом состоянии; если он будет активирован, то цепь будет разомкнута и питание дальше не пойдет.

Автомат QF2 отвечает за подачу питания на розетку собственных нужд.

Автоматы QF3, QF4, QF5 отвечают за включение блоков питания UG1, UG2, UG3 и UG4 соответственно.

После включения автоматов цепь питания идет от блоков питания и от цепи питания в соответствии с указанными стрелочками.

Номера после стрелочек указывают страницу и столбец, куда приходит питание. Например: после стрелочки написано 3.2 – это значит, что нужно смотреть на 3-ю страницу и 2-й столбец.

На рис. 1.11 показано подключение модулей расширения 24 и 5 В.

Автомат QF8 включает панель, модули разветвителей EtherCAT и EtherNET включаются, когда подается напряжение на блоки питания.

Напряжение 24 и 5 В приходит на клеммную колодку X11-X16. От них можно в будущем запитать датчики или модули.

Цифры перед стрелочками в первой колонке указывают, с какой страницы пришла цепь подключения.

На рис. 1.12 автомат QF9 заводит питание на клеммную колодку модулей дискретных входов/выходов и аналоговых входов/выходов.

На рис. 1.13 представлены подключение потенциометра к аналоговому модулю ввода и соединительные перемычки между вводом и выводом.

На рис. 1.14 показано подключение кнопок (SB2-SB4) и тумблеров (SA1-SA3) в дискретные входы и подключение дискретных выходов к реле.

На рис. 1.15 отражено подключение штурвала PG1 к блоку винтовых зажимов A1.1, а затем к высокоскоростным входам/выходам, расположенным на борту контроллера.

На рис. 1.16 показано подключение цифровой сети.

Для обмена данными между панелью A2 и контроллером A1 предусмотрен EtherNET разветвитель A5, между контроллером A1, сервоприводами UZ1/UZ2 и коплером A20 – разветвитель EtherCAT A4.

Контрольные вопросы

1. Что такое контроллер? Назовите функции и характеристики контроллера.

2. Расскажите про модули A21-A22, установленные на стенде. Какие функции они выполняют?

3. Для чего нужен EtherCAT разветвитель?

4. Какую функцию выполняют дискретные вводы/выводы?

5. Для чего предназначен сервомотор MS1? Каковы его основные характеристики?

6. Опишите функции и характеристики преобразователя частоты SV660N.

7. Опишите электрическую принципиальную схему.

2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ КОНТРОЛЛЕРОВ INOVANCE

Действительно функциональным контроллер становится только тогда, когда создают и запускают программу по его использованию. Программу ПЛК Inovance разрабатывают в программном комплексе InoProShop.

Ключевые особенности InoProShop:

– языки программирования IEC 61131-3 (LD, FBD, SFC, ST, IL);

– объектно-ориентированное программирование;

– инструменты отладки;

– стандартные библиотеки МЭК.

Среда разработки основана на стандарте IEC 61131-3, который описывает языки программирования:

а) текстовые:

– Instruction List (IL) – список инструкций;

– Structured Text (ST) – структурированный текст;

б) графические:

– Sequential Function Chart (SFC) – последовательные функциональные схемы;

– Function Block Diagram (FBD) – функциональные блоковые диаграммы;

– Ladder Diagram (LD) – релейно-контактные схемы;

– Continuous Function Chart (CFC) – непрерывные функциональные схемы.

Создание и конфигурация шаблона проекта

Запустите InoProShop, создайте новый проект во вкладке «Файл» (рис. 2.1).

InoProShop(V1.7.3)

Eile	<u>E</u> dit	<u>V</u> iew	<u>Project</u> <u>Build</u>		<u>O</u> nline		<u>D</u> e	<u>D</u> ebug		<u>T</u> ools		<u>W</u> indow		
1	<u>N</u> ew P	roject	. Ctrl+	IN	CI.	Х	Ba	R.	×	- 24	A Z Suir			-

Рис. 2.1. Создание нового проекта

В диалоговом окне выберите шаблон «Standart PLC project», задайте атрибуты проекта («Имя», «Расположение») и нажмите «ОК» (рис. 2.2).



Рис. 2.2. Выбор шаблона

При создании проекта в соответствии с установленной конфигурацией формируется ряд объектов.

Программа работы ПЛК не зависит от конкретной модели контроллера. При программировании ПЛК созданная программа преобразуется в машинный код конкретного процессора. На этапе установления связи с контроллером программа должна взаимодействовать с конфигурацией конкретного ПЛК. Исходная информация о конфигурации ПЛК содержится в предварительных настройках целевой платформы контроллера. Настройки целевой платформы поставляются в виде набора файлов, основным является Target-файл, который содержит информацию о ресурсах конкретного ПЛК, количестве/типе входов и выходов, интерфейсов, памяти, дополнительных устройств и т. д. Целевую платформу (контроллер) выбирают через вкладку «Устройство». В качестве исполнительного устройства может выступать как физический контроллер Inovance, так и виртуальный ПЛК.

Для ознакомления с возможностями среды используется виртуальный ПЛК. Это устройство реализует полнофункциональный ПЛК на персональном компьютере, позволяя работать с внешними устройствами, подключенными к Сот-портам компьютера. В пункте «Устройство» укажите ПЛК АМ600-СРU1608TP/TN.

Вкладка «Language» определяет язык реализации объекта «Программа PLC_PRG». PLC_PRG – это программа ПЛК, которая вызывается один раз за цикл управления.

СFC – графический язык программирования. Программный код представляет собой последовательность цепей, содержащих логическое или арифметическое выражение. Код выполняется по типу определенных шагов. Каждый шаг – это конкретные операции. Как только шаг выполнен – следует действие по передаче управления следующему шагу.

Программирование на языке CFC сводится к выбору библиотечных, готовых функциональных блоков, их позиционированию на экране, установке соединений между их входами и выходами, а также настройке параметров выбранных блоков.

ST (Structured Text) – это текстовый язык высокого уровня общего назначения, по синтаксису схожий с языком Pascal. Удобен для программ, включающих числовой анализ или сложные алгоритмы. Может использоваться в программах, в теле функции или функционального блока, а также для описания действия и перехода внутри элементов SFC.

Задав атрибуты проекта и кликнув по кнопке «ОК», создают проект с указанными настройками.

Интерфейс InoProShop

Интерфейс InoProShop (рис. 2.3) представляет собой набор рабочих областей.

1. Панель управления представлена кнопками, разделами, дублирующими часто используемые команды среды.

2. Окно навигации проекта содержит структуру проекта: POU, типы данных, визуализации, ресурсы, библиотеки.

3. Рабочая область POU отображает редактор программной организационной единицы (POU), каждый программный компонент состоит из раздела объявлений и раздела кода. Для написания кода POU используется один из МЭК языков программирования – IL, ST, FBD, SFC, LD, CFC.

4. Панель инструментов содержит стандартные элементы программирования.

5. Панель сообщений аккумулирует данные компиляции, за-грузки проекта.



Рис. 2.3. Интерфейс программы InoProShop

Состав проекта

Проект представлен структурой (рис. 2.4), включающей в себя программируемое устройство (AM600-CPU1608TP/TN) и приложение (Application).



Рис. 2.4. Выбор структуры

«Application» представляет собой набор объектов, обеспечивающих работу ПЛК.

Объект «Менеджер библиотек» обеспечивает управление библиотеками.

Объект «*PLC_PRG*» – программная организационная единица (рис. 2.5), содержащая код программы. Двойной щелчок ЛКМ по объекту PLC_PRG открывает редактор программы, состоящий из двух областей. В области «Объявления» осуществляется инициализация переменных. В части «Тело программы» на основе инициализированных данных реализуется программный код на выбранном языке.



Рис. 2.5. Организационная программная единица

Объект «Конфигурация задач» служит для организации планирования вычислительного процесса ПЛК. Здесь формулируются задачи и настраиваются их параметры. При создании нового проекта неявно организуется задача, содержащая вызов одной программы PLC_PRG. Данная задача выполняется циклически.

Двойной щелчок по вложенному элементу «MainTask» (Задача) открывает атрибуты задачи.

Параметр «Приоритет» – система исполнения задач ПЛК построена по принципу вытесняющей многозадачности. Задача с высоким приоритетом вытесняет задачу с меньшим приоритетом. Нулевое значение соответствует наивысшему приоритету.

Параметр «*Tun*» определяет способ вызова задачи: через установленный интервал, свободный вызов, событие.

Параметр «*Сторожевой таймер*» обеспечивает защиту от зацикливания задачи. Если задача не возвращает управление дольше заданного времени, то «срабатывает» таймер и происходит вызов специального обработчика системного события.

В нижней части окна «MainTask» (рис. 2.6) располагается редактор, определяющий программы, вызываемые текущей задачей.

Navigator 👻 🖣 🗄	🎦 Менеджер библиотек 🏻 👔 PLC_PRG 🛛 🎉 Конфигурация задач 🖉 🏵 MainTask 🗙	
= 🗿 Test	онфигурация	
Bevice (CODESYS Control Win V3)		
🖹 🎒 Pic Logic	Npwopurter (031): 1 Task Group: IEC-Tasks ~	
🚊 🙆 Application		
👔 Менеджер библиотек	Tun	
	Uurrepsan (Hanp., t#200ms): 20	ms 🗸
🖹 🎯 Конфигурация задач		
🕸 MainTask	Сторожевой таймер	
· 례 PLC_PRG	Включить	
	Prover (upp. +#200mp)-	me 🗸
	upone (nai), rezounts).	11.3
	Восприимчивость: 1	
	A MACHINA Description of the second state in the second state of t	
	Add Call X Remove Call Les Change Call * Move Up * Move Down SUTXparts POU	
	РОО Комментарий	
	PLC_PRG	

Puc. 2.6. MainTask

Объект «Device Diagnose» (рис. 2.7) обеспечивает диагностику ПЛК в онлайн-режиме. В нем отображаются ошибки и предупреждения всех устройств, подключенных к ПЛК, а также ошибки, допущенные при написании кода в программе.

rice Type: All		~	Module Type:		🗸 🤦 Search	📀 Refresh 📃	Clear 🍙 Expo	ort		
DeviceType	ModuleName							Faul	tInfo	
tail TroubleSh	ooting DeepDia	gnosis								
tail TroubleSh	ooting DeepDia	gnosis								
tail TroubleSh ault xtra	ooting DeepDia	gnosis								
tail TroubleSh ault formation 1	ooting DeepDia	gnosis								
tail TroubleSh ault transformation 1 xtra formation 2	ooting DeepDia	gnosis								
tail TroubleSh ault transformation 1 xtra formation 2 xtra formation 3	ooting DeepDia	gnosis								

Puc. 2.7. Device Diagnose

Столбец «Device Type» отображает тип девайса, в котором есть ошибка или предупреждение, столбец «Module Name» – название модуля, столбец «Fault Info» – информацию по текущей ошибке или предупреждению.

Объект «*Network Configuration*» (рис. 2.8) обеспечивает выбор протокола для передачи информации внешним устройствам.



Puc. 2.8. Network Configuration

При нажатии на ПЛК во всплывающем окне отображаются различные протоколы для передачи информации внешним устройствам. По умолчанию в проекте создается связь по протоколу Modbus TCP.

В правой части окна находятся все удаленные устройства Inovance. Двойным щелчком ЛКМ их можно добавить в проект для задания параметров и программирования в проекте.

Объект «Local Config» необходим для выбора устройств, которые будут подключаться непосредственно на шину контроллера. В контроллере AM600-CPU1608TP/TN на локальную шину можно подключать до 16 модулей ввода/вывода.

Все модули ввода/вывода от Inovance представлены в правой части экрана (рис. 2.9). Двойным щелчком ЛКМ можно добавить модуль в проект. После этого добавленный модуль отобразится в дереве проекта.



Puc. 2.9. LocalBus

Объект «*Resources List*» (рис. 2.10) показывает, насколько заполнена память контроллера существующим кодом и переменными, которые используются в нем.

⁄ Q	Device Diagno	osis 🔗	Networ	k Configuratio	m 🕅) Hardwar	e Configu	ration	PLC_PRG	7	HIGH_SPEE	ola	Resour	rces list x
Δ	Click Compile	e to finish refre	esh(Requir	e compiled v	/ithout error	rs)		Compile						
Progra	ım (B)		^	%	M area		9	I area		%Q 8	rea	Program	m Size	
	Total			Display:	🗹 Used	 Search 	n 📃				Display Range	0		Apply < <pre next="" page="">> 1 / 1</pre>
	Used Available			Address	POU	Variable	Туре	Addres	State					
0	Usage rate													
Data (B)													
100	Total													
	Used													
	Available													
0	Usage rate													
Persist	tent Retain(B)													
100	Total													
	Used													
	Available													
0	Usage rate		- 11											
- %M at	rea(B)													
100	Total													
	Used													
	Available													
0	Usage rate		-	1										
%1 are 100	Total		~											

Puc. 2.10. Resources List

Строчка «*Program*» отображает размер программы, строчка «*Data*» – заполнение оперативной памяти, строчка «*Persistent Retain*» – заполнение энергонезависимых переменных, строчка «*M area, I area, Q area*» – размер переменных, используемых для связи с внешними устройствами.

Объект «SoftMotion General Axis pool» позволяет добавить в проект виртуальную ось или энкодер с использованием интерфейса Soft-Motion.

Интерфейс привода SoftMotion – это стандартизированный интерфейс, который используется для подключения, настройки и адресации аппаратного обеспечения привода в рамках программы IEC. Подключая различное оборудование к одному интерфейсу, можно легко обмениваться информацией в ходе технологического процесса и повторно использовать программы IEC. Интерфейс подключает приводы к схеме ввода/вывода и отвечает за обновление и передачу требуемых данных о движении в систему управления приводом.

Интерфейс привода включает следующие компоненты:

1) описание устройств SoftMotion для их представления в дереве устройств;

2) библиотеки, на которые даны ссылки в описании устройства, – они расширяют или перегружают базовые функциональные блоки AXIS_REF_SM3 в соответствии с требованиями конкретных типов приводов;

3) библиотеки, содержащие функциональные блоки для ациклического чтения и записи данных для переноса стандартных функций драйвера полевой шины.
Для добавления оси необходимо нажать ПКМ на объект и выбрать «Add Device».

Во всплывающем окне можно выбрать для добавления в проект (рис. 2.11):

– виртуальный энкодер;

– виртуальную ось для управления по позиции;

– виртуальную ось для векторного управления по скорости.

💀 Add Device			×
Name			
Action Append device Insert device	O PlugDevice O Update de	vice	
	Vendor: <all vendors=""></all>		~
Name	Vendor	Version	Description
🖃 🔗 SoftMotion drives			
🖹 🔗 Free Encoders			
SMC_FreeEncoder	3S - Smart Software Solutions GmbH	3.5.5.0	SoftMotion free Encoder
SM_Drive_PosControl SM_virtual drives	3S - Smart Software Solutions GmbH	4.2.0.0	SoftMotion psoition contr
SM_Drive_Virtual	3S - Smart Software Solutions GmbH	4.0.0.0	SoftMotion virtual drive
 ✓ ✓ Group by category □ Display all ve 	rsions (for experts only) 🗌 Display ou	tdated vers	ions
Please selé	ect a device from the list above.		
(When this window opens, you can s	elect another target node in the navigator)	
		Add De	evice Close

Puc. 2.11. Add Device

Объект «*HIGH_SPEED_IO*» (рис. 2.12) представляет собой конфигуратор высокоскоростных входов/выходов.



Рис. 2.12. Выбор дополнительных элементов

На основном экране необходимо выбрать, как будут использоваться высокоскоростные входы/выходы:

1. «Counter» – добавление счетчика (рис. 2.13). После выставления галочки в окне «Counter parameters» открывается возможность параметризировать счетчик.

Hardware Port Configuration	Counter0 Counter1 Counter2 Counter3 Counter4 Counter5 Counter6 Counter7
Counter Parameters	General Frequency/Rotation Measure
Axis Parameters	Name: HS_Counter0 Type: COUNTER_REF Period: 10 ms ~
Internal I/O Mapping	Counter Format: Linear Ring Pulse Per Rotation:
Status	Externel Trigger (X8)
Information	Input Logic: Positive Negative
	Function: None 🗸

Puc. 2.13. Counter

2. «Axis» – добавление оси (рис. 2.14). После выставления галочки в окне «Axis parameters» открывается возможность задавать параметры оси. При подключении к высокоскоростным входам/выходам будет осуществляться импульсное управление приводом.

Hardware Port Configuration	Axis 0	Axis 1 Axis 2 Axis 3			
Counter Parameters		Axis Name		Home Parameters	
Auia Danamakara		Name: HS_Axis0	Type: HS_AXIS_REF	Home Method:	Method O
Axis Parameters		Positioning Parameters		Home Speed(pulse/s):	1000 🗘
Internal I/O Mapping		Stroke Limit			
Status		Upper(pulse):	2147483647	Creep Speed(pulse/s):	800
Information		Lower(pulse):	-2147483648	Acceleration(pulse/s ²):	1000
		Speed Limit(pulse/s):	200000	Deceleration(pulse/s2);	1000
		Bias Speed(pulse/s):	500	Deceleration(pulse/s-):	
		Rotation Direction:	Positive Negative		Default
		Acc/Dec Method:	● Trapezoid ○ S-curve		
		Help			
		Home Method:Method ()		
			Limit Boa geo aped A here speed bee bee	A:Limit Falling Egdo B:	C:Zero Rising Edd D:Zero Falling Egde

Puc. 2.14. Axis

Максимальное количество осей, подключенных к высокоскоростным входам/выходам, – четыре.

3. «Input/Output» – высокоскоростные дискретные входы/выходы (рис. 2.15). При выставлении галочки во вкладке «Internal I/O Mapping» можно добавлять программные переменные для считывания значений.

Hardware Port Configuration	Find		Filter Show all			•	Add FB	for IO Channel	. Go
Counter Parameters	Variable	Mapping	Channel InputData	Address %IW0	Type UINT	Default Value	Unit	Description	
Axis Parameters	■ • *		OutputData	%QB0	BYTE				
Internal I/O Mapping									
Status									
Information									

Puc. 2.15. Input/Output

Инструменты, расположенные сверху программы, представлены на рис. 2.16:

- Создание нового проекта;
- Открыть существующий проект;
- Сохранить проект;
- Распечатать (программный код, название и т. д.);

– Проверка правильности подключения модулей и удаленных устройств;

– Компиляция проекта. Проверка правильности написания программного кода, подключения устройств и др.;

- Загрузка проекта в ПЛК и запуск программы.



Рис. 2.16. Инструменты основной панели

Для включения режима симуляции (рис. 2.17), чтобы компьютер выступал вместо ПЛК, необходимо нажать вкладку «Online» и выбрать строку «Simulation».

<u>O</u> nli	ine <u>D</u> ebug <u>T</u> o		<u>T</u> ools	<u>W</u> indow	<u>H</u> elp					
СŞ	Lo	gin			Alt+F8					
Сğ	Lo	Logout Ctrl+F8								
	Cr	eate boot	t applicat	tion						
	Do	wnload								
	Or	nline Cha <u>n</u>	ige							
	Do	wnlad <u>P</u> ro	oject Inf	ormation						
	<u>S</u> o	ource dow	nload to	connected	device					
	Mu	ultiple Dov	vnload							
	Re	eset warm	I							
	Re	es <u>e</u> t cold								
	Re	ese <u>t</u> origin	I							
	Sir	nulat <u>i</u> on								
	Se	curity				۲				

Рис. 2.17. Вкладка «Online»

Контрольные вопросы

1. Какие языки программирования представлены в контроллере?

2. Что такое рабочая область POU?

3. Из каких объектов состоит структура проекта?

4. Что такое PLC_PRG? Для чего он используется?

5. Что такое MainTask? Какие действия он выполняет?

6. С помощью какого объекта необходимо выбирать протоколы для связи с удаленными устройствами?

7. Что такое SoftMotion? Для чего он нужен?

8. Какое действие выполняют высокоскоростные входы/выходы? Какие устройства можно подключить к ним?

3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПАНЕЛЕЙ INOVANCE

Программное обеспечение для панелей Inovance – основной инструмент для создания проектов и организации связи в ПЛК.

InoTouchPad – мощный визуальный редактор проектов для операторских панелей Inovance (рис. 3.1).

III InoTouchPad		
Project Edit Compiler View Options Help Tool		
Projects	1	New Project 3
Example	2	C Open Project 4
		Recent Projects : Clear

Рис. 3.1. Приветственное окно InoTouchPad

Двойной щелчок ЛКМ по иконке InoTouchPad открывает основной экран программы, где отображены:

1) проекты, которые в данный момент открыты;

2) примеры проектов от разработчика (как работать с кнопками, как добавлять элементы на экран и др.);

3) вкладка «Создать новый проект»;

4) вкладка «Открыть существующий проект».

После нажатия на иконку «Создать новый проект» на экране появляется всплывающее окно со всеми панелями от фирмы Inovance (рис. 3.2).



Рис. 3.2. Список панелей

Панели делятся на три категории: 7 дюймов – IT7070E 10 дюймов – IT7100E 15 дюймов – IT7150E

На учебном стенде установлена 7-дюймовая панель. Необходимо выбрать из списка панель IT7070E, задать название проекта и путь, куда будет сохранен проект, и нажать кнопку «OK».

После введённой информации откроется проект с установленными по умолчанию различными инструментами (рис. 3.3).

1. Инструменты для работы с объектами, расположенными на экране, а также проверки на правильность, выбора языка, сохранения, открытия проекта.

2. Дерево проекта – на нем отображены экраны, настройки проекта, аварии и предупреждения, рецепты и другие процессы, которыми можно оперировать в проекте.



Рис. 3.3. Общий вид проекта

3. Область проекта – на области отображен экран панели в соотношении 1:1. На осях абсцисс и ординат обозначены пиксели панели.

4. Характеристики добавленных объектов. У каждого объекта есть свои уникальные свойства, которые можно параметризировать.

5. Окно для добавления объектов на экран (рис. 3.4). Для того чтобы добавить объект, необходимо ЛКМ зажать и перенести на экран выбранный объект.



Рис. 3.4. Настройки InoTouchPad

Рассмотрим базовые настройки проекта. Они состоят из следующих элементов:

1. HotKey – бинд клавиши клавиатуры для активации какого-либо действия. Используется, если к панели подключена клавиатура.

2. Scheduler – счетчик. Применяется для активации какого-либо события или скрипта. Скрипт – маленькая программа, написанная на языке JavaScript. С помощью скрипта можно управлять перемещением объектов, изменением цвета и др.

3. Project Version – кнопка необходима для сохранения текущей версии программы. Данная опция позволяет сделать до 15 сохранений проекта, включая все его элементы. Если пользователь захочет вернуть все к какой-либо версии проекта, ему достаточно будет нажать одну кнопку.

4. Printer Settings – настройки принтера. К панели можно подключить принтер для печати различных событий и аварий.

5. Project Settings – основные параметры запуска.

Далее проанализированы расширенные настройки проекта (рис. 3.5):

1. HMI Settings – поле, отображающее тип панели, активного пользователя, стартовое окно, выбранный язык и стиль, примененный к проекту.

2. Screen Saver – поле, отображающее время ухода панели в спящий режим и на стартовый экран.

3. Security Settings – защита паролем. Если установить данную опцию, то при загрузке или выгрузке проекта из панели будет требоваться пароль.

4. Other Settings – настройки резистивного экрана, отображения курсора, звук и др.

5. Alarm Settings – настройки аварий (рис. 3.6). Отображение аварий, звук при появлении предупреждения, всплывающее окно с отображением аварий и др.

6. IP Settings – выставление IP панели.

[]	0001:Screen_1	× 🥓 Projec	t Settings	×			
HN	II Settings						
	Device Type	IT7070E(800x4	80)	-		Start Screen	Screen_1
F F	Project Password			-	S	tart Language	en_US 🔹
	Start Logo	default_logo.pr	g	•		Start Style	<undefined></undefined>
	Default User	admin		•		Author	Mousaka
	Comment				1	1	
- So	reen Saver & Bla	ck Light Setting	IS			- Other Setti	ings
	Screen Sa	wer Wait Time	3 min		÷	🗹 Beep	for clicked
	Screen Saver Act	tivated Screen	<undefined></undefined>		•	🗌 Curso	or Visible
12	Black Li	ght Wait Time	5 min		÷	🗌 Draw	Focus
4						🗌 Displa	ay zero when connection is off
Se	curity Settings —					🗌 🗌 Enab	le preRead
				🗌 disable uploa	ad	Show	tooltips when tag has no limit
	Local pas	sword 111111				Show	DownloadPage when mount SD/UDisk Device
						🗌 Adapi	tive Resolution(only for PC/IPC HMI
	upioad pas	sword				🗌 Enab	le Script var block
	download pas	sword				🗌 Enab	e close button shown in red
	sector pue			2		🗌 Skip	Enter Control Panel
u	pload history pas	sword		3		🗌 Forbi	d Right Button Run on Windows
		L				Pass	word is required for Recovery 4

Рис. 3.5. Настройки проекта (расширенная версия)

 Alarm Settings Beep for unACK alarms continuely Automatically stop the buzzer when a 	ll alarms are cleared	OperationRecord Settings Enable OperationRecord Circular Record(full stop record when Unchecked)				
 Show AlarmWindow SystemAlarm window is Closed Manu 	ally 5	OperationRecord Counts 10000 ‡				
Communication alarm window separat SysAlarm or CommAlarm Duration 2 s	e to display	LoginUserComboBox Enabled Set Login Dialog Set Login Dialog				
Alarm/Data Refresh Rate Settings Alarm/Data Refresh Rate Normal	¥					
Simulator Menu Enable User defined Menu	🗹 Quit	☑ Confirm Before Quit				
☐ Hide Menu when only one Menu item	 Minimized Register Auto Start About Logon Change Screen 	n <undefined></undefined>				
IP Settings	6					
Auto Get IP Address						
Static IP Address						
IP Address: 127 .	0.0	. 1				
Subnet Mask: 255 .	255 . 255	. 0				
GateWay: 127 .	0.0	. 1				

Puc. 3.6. Alarm Settings u IP Settings

Добавление экранов происходит в дереве проекта во вкладке «Screens» (рис. 3.7):

1. Screens – экраны, которые занимают всю область панели.

2. Embed Screens – экраны, которые буду открываться при достижении какого-либо события.

3. Popup Screens – всплывающие экраны. Маленькие окна, которые можно открывать с помощью кнопок, событий и т. д.

В проект можно добавить до 512 экранов.



Puc. 3.7. Screens

Вкладка «Communication» (рис. 3.8) необходима для организации связи. Здесь настраиваются протоколы по передаче данных между панелью и внешними устройствами, а также во вкладке «Tag_Group» создаются теги для дальнейшего использования их в проекте. Практически каждый добавленный на экран элемент требует привязки тега.



Puc. 3.8. Communication

Для добавления тега (рис. 3.9) необходимо нажать на «плюс», расположенный в верхнем левом углу вкладки.

+	Name	- Numbe	r - Connection Id	Data type	Length	Array count	Address -	Acquisition
1	LW 0	1	<internal tag=""></internal>	Int16	2	1	LW 0	100ms
	1	2	3	4		5	6	
on cy	Acquisition m	Data log Id	Logging cycle IdLoggi	ng acqui Uppe	r limit Upper lir	mit alarm Lowe	er limit Lower	limit alarm
	Cyclic on use	<undefined></undefined>	1s Cyclic	c contin <no limi<="" th=""><th>t> Off</th><th><no lim<="" th=""><th>nit> Off</th><th></th></no></th></no>	t> Off	<no lim<="" th=""><th>nit> Off</th><th></th></no>	nit> Off	
		7		8	9	1() 1	1

Рис. 3.9. Конфигурация тега

После добавления тег следует сконфигурировать:

1. «Name» – название тега.

2. «Number» – номер тега.

3. «Connection id» – отображение протокола передачи данных, взаимодействующего с внешними устройствами.

4. «Data type» – тип переменной.

5. «Array count» – какую область пространства будет занимать тег (по умолчанию – 2 байта).

6. «Address» – адрес тега. Адреса бывают:

LW – не энергонезависимая память, сохраняется в оперативной памяти. После выключения панели данные стираются.

RW – энергонезависимая память. После выключения панели данные, которые были в RW памяти, сохраняются.

7. «Data log id» – выбор памяти для сохранения (локальная память, SD карта, USB).

8. «Upper limit» – верхняя граница тега.

9. «Upper limit alarm» – верхняя граница тега для активации аварии.

10. «Lower limit» – нижняя граница тега.

11. «Lower limit alarm» – нижняя граница тега для активации аварии. Вкладка «Data Service» представлена на рис. 3.10.



Puc. 3.10. Data Service

Для передачи данных в облачный сервис используют облачные технологии, через которые можно подключаться к панели через Wi-Fi (рис. 3.11).



Puc. 3.11. Alarm Management

Во вкладке «Alarm Management» создаются аварии и предупреждения, которые будут участвовать в проекте (рис. 3.12).

+.	Text	 Number - 	Class	Trigger tag	Trigger mode	Limit
1	analog_1	1	Errors	<undefined></undefined>	>	<no limit=""></no>

Рис. 3.12. Аналоговые аварии

Основные параметры для конфигурации:

Text – название аварии

Number – номер аварии

Class – выбор типа сообщения (аварии, предупреждения, системные ошибки)

Trigger tag – тег, по которому будет активироваться сообщение

Trigger mode – выбор, по какому из вариантов будет соблюдаться условие (<,>, = и т. д.)

Limit – выбор границы активации сообщения (может быть как тегом, так и числом)

С помощью вкладки «Recipe» в проект добавляют рецепты (рис. 3.13).



Puc. 3.13. Recipe

Рецепт – это набор параметров и их значений, дающий информацию, необходимую для производства продукта или управления процессом. Могут быть созданы различные определения рецептов, например колечек и печенья. Определение рецептов печенья может содержать много различных рецептов (печенье с шоколадной крошкой, сахарное печенье и т. д.).

Благодаря вкладке «Historical Data» создаются области памяти для сохранения переменных, аварий и предупреждений (рис. 3.14).



Puc. 3.14. Historical Data

Строка памяти состоит из названия, номера, количества записей, которые будут сохранены в файл (максимум на локальную память можно записать 100 000 килобит, на SD карту – до 5 000 000 килобит за один файл, максимальное количество файлов для записи – 64), пути, куда будут записываться данные.

Вкладка «Scripts» (рис. 3.15) добавляет в проект скрипт, который можно использовать для активации событий, записи различных текстов в поля ввода/вывода, управления стилями объектов и т. д.

Скрипт пишется на языке программирования JavaScript.



Puc. 3.15. Scripts

Вкладка «User Administration» (рис. 3.16) необходима для составления уровня доступа в проекте.

Уровень доступа – это набор разрешенных операций (например, загрузка файлов, редактирование страниц и другие), привязанный к пользователям и группам пользователей. Уровни доступа применяются в системе управления пользователями, определяются администратором, который может создавать или изменять их.



Puc. 3.16. User Administration

Контрольные вопросы

- 1. Какие вкладки входят в дерево проекта?
- 2. Каковы основные настройки проекта?
- 3. Что такое тег?
- 4. Что такое РорUр и как им управлять?
- 5. Для чего нужны рецепты в проекте?
- 6. Что такое Scheduler?

4. ПРАКТИКУМ

Лабораторная работа № 1

ЗАПУСК ПРОЕКТА И ПЕРЕДАЧА ДАННЫХ В ПЛК

Цель работы: освоить способы подключения панели оператора к контроллеру Inovance.

Задания

1. Создать оригинальный интерфейс для панели оператора.

2. Связать панель оператора с контроллером по OPC UA Server.

3. Связать панель оператора с контроллером по Modbus TCP Protocol.

4. Связать панель оператора с контроллером по Modbus TCP Protocol HMI Slave.

Ход работы

Создание интерфейса панели оператора

Для примера работы проекта создадим два экрана, шаблон и кнопки для переключения (рис. 4.1).



Рис. 4.1. Создание экранов

После добавления шаблона необходимо выбрать, где он будет отображаться (рис. 4.2).

Шаблон(screen)	
General	
± Properties	Settings 🗹 Screen List
	Number 40001 💿 🧭 Экран 1 🗹 Экран 2
	Background color
	Font Arial,16px

Рис. 4.2. Выбор места для отображения

В основных настройках выберите созданные экраны. Это значит, что созданный шаблон с двумя кнопками будет отображаться на всех экранах.

Для добавления кнопок на экран воспользуйтесь инструментами в правой части экрана. Выберите инструмент Button (рис. 4.3) и перенесите его на шаблон.

Tools	a ×
k 2 ■ ■	
Simple Controls	
🦯 Line	
< Polyline	
Polygon	
 Ellipse 	
Rectangle	
🔭 Bezier	
Table	
A Text Field	
Bit Indicator	
He Bit Button	
E Word Indicator	
¹²³ Word Button	
Simple Graphics View	
Graphics View	
01 Number IO Field	
aI String IO Field	
Oate-Time Field	
- Graphic IO Field	
Symbolic IO Field	
Button	
IText Switch	
Graphic Switch	
🖲 Timer	
📑 Gif Display View	

Рис. 4.3. Инструмент Button

Если нажать на элемент на экране «Кнопка», то в нижней части экрана появится поле со списком действий (рис. 4.4).

Button_1(Button) General Animations Events ButtonMode Text Graphic Invisible Click animate Auto Repeat Checkable Hold Delay 0 *1001	Техt ● Text ○ TextList Text OFF Экран 1 □ Text ON Text
---	---

Рис. 4.4. Инструмент Button, вкладка General

Вкладка General у каждого элемента уникальная.

Вкладка Properties (рис. 4.5) служит для изменения цвета элемента, положения, стиля, уровня доступа, прозрачности.

	:
Properties	÷
Appearance	
- Layout	
Text	
 Flashing 	
- StyleSheet	
- Misc	
- Security	
Operation Record	

Рис. 4.5. Вкладка Properties

Вкладка Animation (рис. 4.6) служит для различной анимации, активируемой по тегу, например: если добавить тег и задать, что при целочисленном значении от 0 до 100 кнопка будет красной, а как только значение перейдет в 101, кнопка станет зеленой.

Enable objects – активация или деактивация элемента на восприятие касаний.

Visibility – видимость элемента, активация происходит по тегу.

Button_1(Button)						
General	Enabled					
Animations	Tag	+	Value	Foreground Color	Backgr	Flashing
Enable Object		1	0 - 100	#ff0000	#e	No
Events	Integer	2	101 - 201	#00ff00	#e	No
	Binary					
	○ Bit 0					

Рис. 4.6. Вкладка Animation

Вкладка Events (рис. 4.7) – события, которые будут активироваться при нажатии на кнопку.



Рис. 4.7. Вкладка Events

Необходимо выбрать, на что будет срабатывать событие: на щелчок по кнопке, когда кнопка зажата, по изменению или по отжатию кнопки.

Выберите режим «По щелчку» (рис. 4.8), после чего откроется список с выбором действий:

Calculation – задает целочисленные значения в тег, инвертирует число.

Edit bits – задает биты в тег, инвертирует биты.

Screens – вкладка для управления экранами: переключение, открытие, закрытие, показ предыдущего экрана.

User administration – вкладка уровня доступа. Функция сохраняет название, пароль, группу текущего пользователя в тег, а также открывает окно для ввода логина и пароля.

Hmi Date Time – установка формата и даты на внутренних часах. Settings – настройки панели (спящий режим, смена языка и т. д.).

Communication – выставление протокола, задание IP адреса и т. д. Data Service – облачные технологии.

Print – настройка принтера.

Alarms – управление предупреждениями и авариями.

Logs – выбор, куда будут сохраняться теги и аварии.

Recipes – работа с рецептами, отправка в ПЛК, запоминание в панель и др.

 Recipes Operation for screen objects Report User scripts

Рис. 4.8. Вкладка Events – Click

После того как рассмотрены настройки, необходимо на основном поле задать текст (рис. 4.9) и поменять цвет кнопки.

Button_1(Button)			
General Properties Appearance Layout Text Flashing StyleSheet Misc Security Operation Re Animations	ButtonMode ● Text ● Graphic ● Invisible ✓ Click animate ■ Auto Repeat ■ Checkable Hold Delay 0 *1001 \$	Text Text OFF Text OFF	TextList Экран 1
⊡ Events			

Рис. 4.9. Смена названия

Далее зайдите в Events и выберите Screens. Для переключения понадобится команда Activate Screen (рис. 4.10). Выберите первый экран для первой кнопки. Те же самые действия следует повторить для второго экрана.

+- 盦 t ↓ № 目			
🕀 Calculation	81	ActivateScreen	
🗄 Edit bits			
- Screens		Screen name	Экран 1
- ActivatePreviousScreen			
- ActivateScreen			
ActivateScreenByNumber			

Рис. 4.10. Команда Activate Screen

Создайте две битовые кнопки и измените цвета у поля ввода/вывода. На второй экран добавьте Number io field и Bit button.

Главные настройки для битовой кнопки представлены на рис. 4.11.

Bit Button_1(BitButton)		
General Properties Animations Events	Read Read Tag <undefined></undefined>	Write ✓ Read/Write Tag Same Mode
	press ✓ Click animate ✓ Auto Repeat ✓ Filter Repeat Msg Hold Delay 0 *100ms	

Рис. 4.11. Пример работы с 1-битовыми кнопками

Создайте тег (рис. 4.12) и в событиях кнопки задайте «Установить число в тег».

Bit Button_1(BitButton)				
General	_+_茴↑↓ГЕ目			
 Animations 	E-Calculation	81	SetValue	
Events	- Increase Value		Tag(Out)	LW 0
Press	- InverseLinearScaling		Value	120
	- SetValue			
Deactivate	Random			

Рис. 4.12. Создание тега и установление значений кнопки

Далее выберите экран ввода/вывода, затем основные настройки и поле для изменения цвета.

Создайте первую битовую кнопку со значением 120 для активации красного цвета, затем – вторую кнопку и задайте значение 10 для активации зеленого цвета (рис. 4.13).



Рис. 4.13. Пример визуализации кнопки на экране

Для запуска проекта необходимо проверить правильность заполнения полей. Воспользуйтесь инструментами сверху (рис. 4.14).



Рис. 4.14. Симуляция работы/проверка проекта

На рис. 4.14 цифрами обозначены следующие инструменты:

1) проверка всего проекта;

2) быстрая проверка;

3) запуск проекта в режиме офлайн (на компьютере будет отображаться панель с графикой, без соединения);

4) запуск в режиме онлайн (если панель подключена к контроллеру, то компьютер заменяет ее и все соединения идут с компьютера на ПЛК).

Нажмите на офлайн-запуск.

На мониторе отобразилась панель. Для переключения по экранам следует нажать «Экран 1» или «Экран 2» (рис. 4.15).

٢	*	*			
		Экран 1	Эк	ран 2	
		INO	ANCE		

Рис. 4.15. Панель «Экран 1»

На втором экране, если нажать на кнопку 1, поле ввода/вывода поменяет цвет (рис. 4.16).

_	© ∗	*					
		1	120	2			
		Эі	фан 1			Экран 2	
				INO	ANCE		

Рис. 4.16. Панель «Экран 2»

Для того чтобы закрыть визуализацию, щелкните ПКМ по экрану и нажмите Quit. Данные с ПЛК передаются на панель.

Организация связи контроллера с панелью визуализации IT7000 по OPC UA Server

В InoProShop зайдите во вкладку Device \rightarrow PLC Logic \rightarrow Application \rightarrow Add Object \rightarrow Symbol Configuration (рис. 4.17).

Devices	- n v	
		1 PROGRAM PLC PRG
Device (AM401-CPU1608TP/TN)		2 🖂 VAR
		3 a:REAL;
Ketwork Configuration		4 b:INT;
		5 c:bool;
		END_VAR
= 😳 Application		
📲 Library N 🖺 Copy		1
PLC_PRC 📸 Paste		
🖹 🔛 Task Cor 🐰 Cut		
E S Main		
SoftMation Con	ues	
HIGH SPEED	oject 🕨 🥂	Alarm configuration
	lder	Axis Group
Edit Ob	oject 🔕	Cam table
Edit Ob	oject With 🖄	CNC program
Collaps	e Application	CNC settings
😋 Login	**	DUT
		External File
	0	Global Variable List
		Image Pool
	~0~	Interface
	2	Network Variable List (Receiver)
	2	Network Variable List (Sender)
	T	Persistent Variables
	æ	POU
	æ	POU for implicit checks
		Recipe Manager
		Symbol Configuration
		Text List
	Q	Trace

Puc. 4.17. Bxod в Symbol Configuration

В появившемся окне выберите Support OPC UA features и Optimized Layout (рис. 4.18).

Symbol Configu	ration		
Support OF	PC UA features	1	
Add library	placeholderin D	evice Applicati	on d)
Client Side Dat	a Layout	rgger downloa	-7
🔿 Compatibil	ity Layout		
Optimized	Layout		

Puc. 4.18. Support OPC UA features u Optimized Layout

Далее в окне объявления переменных создайте теги, например переменные a:REAL, b:INT, c:bool, объявленные в PLC PRG.

В Symbol Configuration нажмите Build и выберите созданные переменные (рис. 4.19).



Рис. 4.19. Выбор созданных переменных

Загрузите программу в контроллер и запустите ее.

В InoTouchPad пропишите соединение:

Зайдите во вкладку Project \rightarrow Communication \rightarrow Connections (рис. 4.20).



Puc. 4.20. Project \rightarrow Communication \rightarrow Connections

Создайте новое соединение, нажав на кнопку + . Задайте имя, например "АМ ОРС".

В ячейке Communication protocol выберите OPC \rightarrow OPC UA Client (рис. 4.21).



Puc. 4.21. $OPC \rightarrow OPC$ UA Client

Далее в General в URL:opc.tcp.// (рис. 4.22) пропишите IP контроллера. Для этого нажмите Browser Tags, при этом контроллер должен быть подключен к ПЛК.

URL: opc.tcp:// 19	192.168.1.88 Port: 4840			
Session Name: T7	000_HMI-IT7070E-Demo	-TEST-OPC		
Connection Encryp	tion: None			
Data Encryption:	n: None			
Timeout:	1000 ms			
Interval:	1000 ms	\$		

Рис. 4.22. Работа в разделе General

В открывшемся окне Root \rightarrow Objects \rightarrow DeviceSet \rightarrow Inovance-ARM-Linux \rightarrow Resources \rightarrow Application \rightarrow Programs выберите нужные переменные, например a[Float], b[Int16], c[Boolean], объявленные в PLC_PRG, и нажмите Add Tag (рис. 4.23).



Рис. 4.23. Выбор необходимых переменных

Выберите Create PLC Tags to group, а также нужную группу тегов, в которую требуется записать данные переменные (рис. 4.24).

F		-Import			
Address space:			Display Name	Node ID	Data Type
opc.tcp://192.168.1.88:4840 E Root		1	a@AM401	ns=4;s= var	Float
		2	b@AM401	ns=4;s= var	Int16
		3	c@AM401	ns=4;s= var	Boolean
	Add Tag Delete Tag Remove All				

Рис. 4.24. Запись данных переменных

Организация связи контроллера с панелью визуализации IT7000 no Modbus TCP Protocol

В InoProShop зайдите во вкладку Device \rightarrow Network Configuration и кликните ЛКМ на модуль. В появившемся окне выберите Ethernet \rightarrow ModbusTCP Slave (4.25).



Рис. 4.25. Вход в ModbusTCP Slave

Далее зайдите в Device \rightarrow MODBUS_TCP (ModbusTCP Device) \rightarrow ModbusTCP Slave Configuration \rightarrow Slave Port (рис. 4.26) и установите порт (порт должен быть таким же, как и в панели).

Devices 👻 🕂 🗙				
Untitled1 Device (AM401-CPU1608TP/TN)	Modbus TCP Slave Configuration			
Device Diagnose Network Configuration	Device Diagnose	Slave Port:	502	-
	Status	Time between Frames(ms):	1	÷
GVL_HMI	Information			
Library Manager PLC PRG (PRG) MainTask PLC_PRG PLC_PRG PLC_PRG PCC_PRG PCC_PRG PCC_PRG PCC_PRG PCC_PRG PCC_PRG PCC_PRG PCC_PRG PCC_PRG PCC_PRG PLCC_PCC PLCCC PLCC_PCC PLCCC PLCC PLCC PLCC PLCCC PLCC PLCC PLCCC PLCCC				

Рис. 4.26. Установка порта

Пропишите переменные с их адресами (рис. 4.27).

При определении адреса используются определенные символьные строки для выражения положения и размера памяти (табл. 4.1).

Синтаксис:

%<memory range prefix><size prefix><number|.number|.number....>



Рис. 4.27. Запись переменных

Поддерживаются следующие префиксы диапазона памяти:

I – диапазон входной памяти (физические входы от входных дисков, датчиков).

Q – диапазон выходной памяти (физические выходы для выходных дисков, исполнительных механизмов).

М – диапазон памяти флага.

Поддерживаются следующие префиксы размеров:

Х – (одиночный) бит.

Никакой – (одиночный) бит.

B – BYTE (8 бит).

W – WORD (16 бит).

D - DWORD (32 бита).

Примеры программирования с использованием приведенных выше префиксов приведены в табл. 4.1.

Таблица 4.1

Фрагмент кода	Значение запрограммированной строки			
%QX7.5	Виходной бит 7.5			
%Q7.5	Быходной бит 7,5			
%IW215	Входное слово 215			
%QB7	Выходной байт 7			
94MD48	Двойное слово в позиции памяти 48 в памяти			
/01/10/48	флага			
%IW2 5 7 1	Интерпретация зависит от текущей конфигурации			
/01 W 2.3.7.1	контроллера			
ivar AT %IW0: WORD;	Пример объявления переменной с адресом			

Примеры программирования

В объявлении переменных код присваивает переменную проекта определенному входному адресу, выходному адресу или адресу памяти контроллера, настроенного в дереве устройств.

Синтаксис:

<identifier> AT <address>:<data type>;

Это объявление (АТ) позволяет дать информативное название адресу. Можно внести любые необходимые изменения для входных или выходных сигналов только в одном месте, например в объявлении.

Если назначается переменная адреса, следует обратить внимание на следующее:

1. Нельзя выполнять запись в переменные, которые размещаются на входных данных, поскольку это приведет к ошибкам компиляции.

2. Объявления АТ можно выполнять только для локальных и глобальных переменных, а не для входных и выходных переменных POU.

3. Объявления АТ нельзя использовать в списках постоянных переменных.

4. Если для компонентов структуры или функционального блока используются объявления АТ, то все экземпляры используют одну и ту же память. Это похоже на использование статических переменных в классических языках программирования, таких как С.

5. Компоновка памяти структур также зависит от целевой системы. Пример: D0 содержит B0 – B3, W0 содержит B0 и B1, W1 содержит B1 и B2, а W2 содержит B2 и B3. Для того чтобы избежать перекрытия, не используйте W1 или D1, D2, D3 для адресации.

Если адрес одного бита не определен явно, CODESYS выделяет логические значения в байтах.

Пример: изменение значения влияет на диапазон varbool1 AT %QW0QX0.0QX0.7

В InoTouchPad пропишите соединение:

Зайдите во вкладку Project \rightarrow Communication \rightarrow Connections.

Создайте новое соединение, нажав на кнопку + .

Задайте имя, например "Modbus TCP AM401".

В ячейке Communication protocol выберите Inovance \rightarrow AM600 Series \rightarrow AM600 Modbus TCP Protocol (рис. 4.28).



Рис. 4.28. Установка AM600 Modbus TCP Protocol

B Slave Device установите IP Address контроллера Port, Slave address и Address Interval(words) (рис. 4.29).

IP Address	192 168 1	. 88	Slave address	_1	
Port	502	\$	Address Interval(worc	s) 5	
			Max Read(words)	120	
			Max Write(words)	120	

Рис. 4.29. Указание адреса контроллера

Далее перейдите во вкладку Project \rightarrow Communication \rightarrow Tags \rightarrow \rightarrow Add Tag Group, затем – в созданную группу тегов (рис. 4.30).



Рис. 4.30. Выбор необходимого тега

Создайте переменную, нажав на кнопку + .

Задайте имя переменной (для удобства пользования имя переменной в панели должно совпадать с именем переменной в контроллере).

Выберите в ячейке Connections Id созданное соединение (рис. 4.31).

	Name	 Number 	-	Cor	nneo	tion Id	Data typ	e Length	Array count	A
1	а	113	Мос	dbus '	TCF	2_AM40	1 🝷 Bool	1	1	QX 1
2	b	114				10000		1		22
3	с	115			^	Id	Name	Info		23
				1	٠	0	<internal tag=""></internal>			
				2	٠	19	Modbus TC	Inovance A		
				2	٠	19	Modbus TC	Inovance A		
			+						×	

Рис. 4.31. Выбор созданного соединения

В ячейке Data type выберите тип переменной (должен совпадать с типом переменной в контроллере), в ячейке Address – Area и Address (рис. 4.32) (должен совпадать с адресом в контроллере).

Name -	 Number • 	Connection Id	Data type	Length	Array count	Address -	Acquisition	Acquisition
а	113	Modbus TCP_AM401	Bool		1	QX 1.0 👻	100ms	Cyclic on us
b	114	Modbus TCP_AM401	Int16	2	1	Area	v	
с	115	Modbus TCP_AM401	Int16	2	1	Alea Q	^	
						Address 1		*
						Bit		
								✓ X
	a b c	Name Number a 113 b 114 c 115	Name Number Connection Id a 113 Modbus TCP_AM401 b 114 Modbus TCP_AM401 c 115 Modbus TCP_AM401	Name • Number Connection Id Data type a 113 Modbus TCP_AM401 Bool b 114 Modbus TCP_AM401 Int16 c 115 Modbus TCP_AM401 Int16	Name Number Connection Id Data type Length a 113 Modbus TCP_AM401 Bool Int16 2 b 114 Modbus TCP_AM401 Int16 2 c 115 Modbus TCP_AM401 Int16 2	Name Number Connection Id Data type Length Array count a 113 Modbus TCP_AM401 Bool 1 b 114 Modbus TCP_AM401 Int16 2 1 c 115 Modbus TCP_AM401 Int16 2 1	Name Number Connection Id Data type Length Array count Address a 113 Modbus TCP_AM401 Bool 1 QX 1.0 • b 114 Modbus TCP_AM401 Int16 2 1 Area Q c 115 Modbus TCP_AM401 Int16 2 1 Area Q Address 1 Modbus TCP_AM401 Int16 2 1 Brea Q	Name Number Connection Id Data type Length Array count Address Acquisition a 113 Modbus TCP_AM401 Bool 1 QX 1.0 100ms Area QX Area QX Area Address 1 Area Area Address Area Address 1 Bit 0 Area Area Address Bit 0 Address Addres Address Addres

Рис. 4.32. Выбор Area и Address

При написании программы следует учитывать, что у каждого типа данных – свои варианты написания области и адреса.

Организация связи контроллера с панелью визуализации IT7000 no Modbus TCP Protocol HMI Slave

В InoProShop зайдите во вкладку Device \rightarrow Network Configuration и кликните ЛКМ на модуль. В появившемся окне выберите Ethernet \rightarrow \rightarrow ModbusTCP Master (рис. 4.33).



Рис. 4.33. Выбор ModbusTCP Master

Далее зайдите в Device \rightarrow Network Configuration \rightarrow Network Devices List \rightarrow Ethernet Port \rightarrow MODBUS_TCP (рис. 4.34).



Рис. 4.34. Вход в MODBUS_TCP

Далее зайдите в Device \rightarrow MODBUS_TCP (ModbusTCP Device) \rightarrow \rightarrow modbusTcp (ModbusTCP Slave) \rightarrow ModbusTCP Slave Configuration. Окна для ввода настроек.

Slave IP Address – установите IP адрес панели.

Slave Port – установите порт (порт должен быть таким же, как и в панели).

Unit ID [0..255] – установите адрес в сети Modbus (адрес должен быть таким же, как и в панели).

Slave Enable Variable: SM – задайте переменную включения (рис. 4.35).



Рис. 4.35. Ввод переменной включения

Зайдите в Device \rightarrow MODBUS_TCP (ModbusTCP Device) \rightarrow modbusTcp (ModbusTCP Slave) \rightarrow Modbus TCP Slave Communication Configuration (рис. 4.36) \rightarrow Add (для того чтобы не переводить размер области памяти из десятичной системы счисления в шестнадцатеричную, можно поставить галочку Use decimal offset).



Puc. 4.36. Modbus TCP Slave Communication Configuration

В открывшемся окне задайте название канала, например Channel 01.

Access Туре – выбор типа доступа: чтение или запись переменных (табл. 4.2).

Таблица 4.2

Код функции	Функция	Название	Тип значения	Тип доступа
(Function	Чтение DO	Read Coils	Дискетное	Чтение
Code 01)				
(Function	Чтение DI	Read Discrete	Дискетное	»>
Code 02)		Inputs		
(Function	Чтение АО	Read Holding	16-битное	»
Code 03)		Registers		
(Function	Чтение AI	Read Input	16-битное	»
Code 04)		Registers		
(Function	Запись	Write Single	Дискетное	»>
Code 05)	одного DO	Coil		
(Function	Запись	Write Single	16-битное	»
Code 06)	одного АО	Register		
(Function	Запись	Write Multiple	Дискетное	»
Code 15)	нескольких DO	Coils		
(Function	Запись	Write Multiple	16-битное	»
Code 16)	нескольких АО	Registers		

Функция чтения или записи

Offset – смещение области памяти (табл. 4.3) (необходимо для того, чтобы в одну и ту же область памяти не попали разные переменные).

Таблица 4.3

Переменные в числовом виде	Переменные в буквенном виде
Чтение/запись переменных 0x/1x	Соответствует чтение/запись переменной
(0-12000)*	LB (0 – 11999)
Чтение/запись переменных 3х/4х/5х	Соответствует чтение/запись
(0 - 10000)	переменной LW (0 – 8999)
Чтение/запись переменных 3x/4x/5x	Соответствует чтение/запись
(10000 - 65535)	переменной RW (0 – 55535)

Функции смещения области памяти переменных

* Цифры в скобках – это условный диапазон, в котором может быть сохранена переменная.

Объявите переменную SM в PLC_PRG в соответствии с заданным значением в ModbusTCP Slave Configuration (рис. 4.37, 4.38). С помощью окна Length можно задать количество переменных.
Name	Channel 01	
Access Type	Read Holding Registers(Function Code 03)	~
Trigger	Read Colis(Function Code 01) Read Discrete Inputs(Function Code 02)	
Repeated	Read Holding Registers (Function Code 03) Read Input Registers (Function Code 04) Write Single Coil (Function Code 05) Write Single Register (Function Code 06)	
Comment	Write Multiple Coils(Function Code 15) Write Multiple Registers(Function Code 16)	
Offset	0	
Offset Length(WORD)	0	* *
Offset Length(WORD) Error Handling	0 1 Keep Last Value	÷.
Offset Length(WORD) Error Handling Vrite Register	0 1 Keep Last Value	* *
Offset Length(WORD) Error Handling Vrite Register Offset	0 1 Keep Last Value	×

Рис. 4.37. Обновление переменной SM в PLC_PRG (шаг 1)



Рис. 4.38. Обновление переменной SM в PLC_PRG (шаг 2)

В InoTouchPad пропишите соединение:

Зайдите во вкладку Project \rightarrow Communication \rightarrow Connections.

Создайте новое соединение, нажав на кнопку + .

Задайте имя, например "Modbus TCP Slave_AM401".

В ячейке Communication protocol выберите Inovance \rightarrow IT7000 Series \rightarrow Modbus TCP Protocol Slave (рис. 4.39).

+	Name	🔺 Number 🚽	Communication protocol	Default status
1	Modbus TCP Slave_AM401	1	Modbus TCP Slave -	Online D
Interfa	Ce		 Inovance H1U/H2U/H3U Series H5U Series AM600 Series AC810 Series Transducer Series Transducer Series IT7000 Series Modbus TCP Slave Modbus TCP Slave Modicon Delta Panasonic Free Protocol 	
-HMI	as master device			

Puc. 4.39. Inovance \rightarrow IT7000 Series \rightarrow Modbus TCP Protocol Slave

B Slave Device установите Slave address и Address Interval(words) (рис. 4.40).

Slave	Device-			_
		Slave address	1	*
Port	502 ‡	Address Interval(words)	5	+
		Max Read(words)	120	÷
		Max Write(words)	120	÷

Puc. 4.40. Установка Slave address u Address Interval

Затем перейдите во вкладку Project \rightarrow Communication \rightarrow Tags \rightarrow Add Tag Group и в созданную группу тегов.

Создайте переменную, нажав на кнопку + .

Задайте имя переменной (для удобства пользования имя переменной в панели должно совпадать с именем переменной в контроллере).

В ячейке Data type выберите тип переменной. Она должна совпадать с типом переменной в контроллере.

В ячейке Address выберите Area и Address – они должны совпадать с областью памяти в контроллере (рис. 4.41).

+	🔹 Name 🝷	Number •	Connection Id	Data type	Length	Array count	Address -	Acquisition	Acquisition .
1	LW 2	8	<internal tag=""></internal>	Int16	2	1	LW 2	100ms	Cyclic on us
2	LW 1	7	<internal tag=""></internal>	Int16	2	1	LW 1	100ms	Cyclic on us
3	LW 0	6	<internal tag=""></internal>	Int16	2	1	LW 0 -	100ms	Cyclic on us
							Area LV Address 0	/	* *

Рис. 4.41. Выбор Area и Address

При написании программы следует учитывать, что у каждого типа данных – свои варианты написания области и адреса.

Содержание работы

- 1. Цель работы.
- 2. Задания.
- 3. Ход работы.
- 4. Выводы.
- 5. Контрольные вопросы.

Средства, используемые при выполнении лабораторной работы

1. Методические указания к выполнению лабораторных работ.

2. Данные, предоставленные преподавателем во время занятия.

3. При проведении анализа допускается использование глобальной сети Интернет.

Контрольные вопросы

1. Какие способы связи панели оператора с контроллером вам известны?

2. Как связать панель оператора с контроллером по ОРС UA Server?

3. Как связать панель оператора с контроллером по Modbus TCP Protocol?

4. Как связать панель оператора с контроллером по Modbus TCP Protocol HMI Slave?

5. Какие различия существуют между рассмотренными способами подключения панели оператора и контроллера?

Лабораторная работа № 2

РАБОТА С ДИСКРЕТНЫМИ ВХОДАМИ/ВЫХОДАМИ

Цель работы: освоить способы подключения и работы с модулями при помощи дискретных входов/выходов.

Задания

- 1. Подключиться к предоставленным модулям.
- 2. Написать программу для включения и выключения лампочек.
- 3. Протестировать написанный код на учебном стенде Inovance.

Ход работы

Разработка и создание кода для управления лампами с помощью дискретных входов/выходов

В дереве проекта во вкладке LocalBus Config необходимо добавить локальные модули, которые физически установлены на стенде (рис. 4.42). Первый модуль после контроллера A10 – GL10-1600END, второй A11 – GL10-0016ETP.



Рис. 4.42. Подключение к модулю

Для добавления модулей воспользуйтесь вкладкой справа со всеми модулями (рис. 4.43).



Рис. 4.43. Выбор и добавление модуля

После добавления модулей дерево проекта обновится (рис. 4.44).



Рис. 4.44. Дерево проекта после выбора модуля

После нажатия ЛКМ на модуль дискретных входов на экране появляется конфигурация битов. Так как модуль рассчитан на 16 дискретных входов, то он поделен на два канала по 8 бит. Во вкладке Mapping можно добавлять внутренние программные переменные в модуль для отслеживания состояния дискретных входов. Переменные могут быть добавлены как на каждый бит в канале, так и на весь канал. Переменные должны быть такого же типа, как указано в колонке Туре (рис. 4.45).

CPU DI16 I/O Mapping	Find	Find Filter Show all Add FB for IO Channe								
Status	Variable	Mapping	Channel	Address	Туре	Default Value	Unit	Description		
	F *		IB(I)	%IB2	USINT					
Information	¥>		IO	%IX2.0	BOOL					
			I1	%IX2.1	BOOL					
	🍫		I2	%IX2.2	BOOL					
	¥		I3	%IX2.3	BOOL					
	* ø		I4	%IX2.4	BOOL					
	- * >		15	%IX2.5	BOOL					
	* >		16	%IX2.6	BOOL					
	- * ø		17	%IX2.7	BOOL					
	🖹 🁋		IB(II)	%IB3	USINT					
	¥ø		I0	%IX3.0	BOOL					
			I1	%IX3.1	BOOL					
	¥ø		I2	%IX3.2	BOOL					
			13	%IX3.3	BOOL					
	¥ø		I4	%IX3.4	BOOL					
	ᡟ		15	%IX3.5	BOOL					
	¥ø		16	%IX3.6	BOOL					
			17	%IX3.7	BOOL					

Рис. 4.45. Написание переменных

Создайте переменные в основной программе в поле ввода параметров (рис. 4.46).



Рис. 4.46. Написание переменных в основной программе

По схеме (см. рис. 1.8 – 1.16) можно увидеть, что тумблеры и кнопки запрограммированы в битах от 0 до 2 на первом и втором каналах соответственно. Для добавления переменных необходимо нажать в поле Variable два раза ЛКМ (рис. 4.47).

₩.	2 2	menzio	DOOL
* @	Q4	%QX2.4	BOOL
***	Q5	%QX2.5	BOOL
····· * @	Q6	%QX2.6	BOOL
	Q7	%QX2.7	BOOL

Рис. 4.47. Проверка подключения тумблеров и кнопок

После нажатия появится кнопка с тремя точками. Нажмите на нее. Во всплывающем окне откройте Application и раскройте программу – в ней видны все доступные переменные для выбора (рис. 4.48).

Input Assistant				>
Text Search Categories				
Variables	▲ Name	Туре	Address	Origin
	= 🔘 Application	Application		
	🖻 📄 PLC_PRG	PROGRAM		
	🖤 🔷 Var1	BOOL		
	🖉 🖗 Var2	BOOL		
	🖤 < Var3	BOOL		
	🖤 🖗 Var4	BOOL		
	🖉 🖗 Var5	BOOL		
	🔷 🖗 Var6	BOOL		
	🗉 🎒 IoConfig_Globals	VAR_GLOBAL		
	🗷 🚞 SDElement			
	E {} SM3_Basic	Library		SM3_Basic, 4.10.2.2
	I SM3_CNC	Library		SM3_CNC, 4.2.1.1 (3
	■ {} SM3_Math	Library		SM3_Math, 4.10.0.0
	🖻 🖂 SMElement			
Structured view			Filter	None ~
Documentation		🗹 Insert v	with arguments	Insert with namespace prefix
				OK Cancel

Рис. 4.48. Доступные переменные в программе

Выберите для каждого канала переменную. Все остальные переменные в данной вкладке относятся к системным переменным (рис. 4.49).

PLC_PRG GL10_1	600END ×								
CPU DI16 I/O Mapping	Find	Filter Show	all			✓ Add FB for IO Channel Go to Insta			
Status	Variable	Mapping	Channel	Address	Туре	Default Value	Unit	Description	
	📮 🏘		IB(I) %IB2	%IB2	USINT				
Information	Application.PLC_PRG.Var1	~	10	%IX2.0	BOOL				
	Application.PLC_PRG.Var2	۵۵	I1	%IX2.1	BOOL				
	Application.PLC_PRG.Var3	~⊘	12	%IX2.2	BOOL				
			13	%IX2.3	BOOL				
	*		I4	%IX2.4	BOOL				
	*		15	%IX2.5	BOOL				
	🍫		16	%IX2.6	BOOL				
			17	%IX2.7	BOOL				
	🚊 - 🍫		IB(II)	%IB3	USINT				
	Application.PLC_PRG.Var4	~⊘	10	%IX3.0	BOOL				
	Application.PLC_PRG.Var5	~	I1	%IX3.1	BOOL				
	Application.PLC_PRG.Var6	~	12	%IX3.2	BOOL				
	*		13	%IX3.3	BOOL				
	🍫		I4	%IX3.4	BOOL				
	🍫		15	%IX3.5	BOOL				
	*** **		16	%IX3.6	BOOL				
	*		17	%IX3.7	BOOL				

Рис. 4.49. Созданные и системные переменные

Те же самые операции выполните и для модуля дискретных выходов (рис. 4.50). На каждый канал от 0 до 2 бит физически подключены реле с лампочками.

CPU D016 I/O Mapping	Find	Filter Show a	dl 🛛			Add FB for IC	Chann	el Go to Ins
Status	Variable	Mapping	Channel	Address	Туре	Default Value	Unit	Description
			QB(I)	%QB1	USINT			
Information	Application.PLC_PRG.Var1	۵۵	Q0	%QX1.0	BOOL			
	Application.PLC_PRG.Var2	~	Q1	%QX1.1	BOOL			
	Application.PLC_PRG.Var3	~	Q2	%QX1.2	BOOL			
	[*]		Q3	%QX1.3	BOOL			
	* *		Q4	%QX1.4	BOOL			
			Q5	%QX1.5	BOOL			
			Q6	%QX1.6	BOOL			
	L		Q7	%QX1.7	BOOL			
	🚊 - K ø		QB(II)	%QB2	USINT			
	Application.PLC_PRG.Var4	~	Q0	%QX2.0	BOOL			
	Application.PLC_PRG.Var5	~	Q1	%QX2.1	BOOL			
	Application.PLC_PRG.Var6	~	Q2	%QX2.2	BOOL			
	* \$		Q3	%QX2.3	BOOL			
	*		Q4	%QX2.4	BOOL			
	* ø		Q5	%QX2.5	BOOL			
	🍫		Q6	%QX2.6	BOOL			
			Q7	%QX2.7	BOOL			

Рис. 4.50. Присвоение переменных для модуля дискретных выходов

Загрузите программу в контроллер (стенд) и нажмите Play для того, чтобы программа запустилась (рис. 4.51).

ject <u>B</u> uild	d <u>O</u> nli	ne <u>D</u> ebug	<u>T</u> ools <u>W</u> inde	ow <u>H</u> elp
🚑 🔊	OŞ.	<u>L</u> ogin		Alt+F8
	Oğ.	Log <u>o</u> ut		Ctrl+F8
		Create hoot	application	

Рис. 4.51. Загрузка и запуск программы

После запуска программы включите и выключите тумблеры на стенде, нажмите на кнопки – на реле должны загореться лампочки.

Изучим математические функции CoDeSys.

Выйдите из программы, нажав шестеренки Logout, и зайдите в основную программу.

Запишите новые переменные (рис. 4.52).



Рис. 4.52. Запись новых переменных

Редактируйте значения дискретных входов (рис. 4.53, 4.54).

PU DI16 I/O Mapping	Find	Filter Show	all			 Add FB for I 	0 Chan	nel Go to l
Status	Variable	Mapping	Channel IB(I)	Address %IB2	Type USINT	Default Value	Unit	Description
nformation	🍄 Application.PLC_PRG.Var1	~¢	IO	%IX2.0	BOOL			
	- *		I1	%IX2.1	BOOL			
	¥ø		I2	%IX2.2	BOOL			
	¥		13	%IX2.3	BOOL			
	*		I4	%IX2.4	BOOL			
	*		15	%IX2.5	BOOL			
	*		16	%IX2.6	BOOL			
	L 🧤		17	%IX2.7	BOOL			
	🖮 🍫		IB(II)	%IB3	USINT			
	*		I0	%IX3.0	BOOL			
	*		I1	%IX3.1	BOOL			
	*		I2	%IX3.2	BOOL			
	*		13	%IX3.3	BOOL			
	*		I4	%IX3.4	BOOL			
	* •		15	%IX3.5	BOOL			
	* •		I6	%IX3.6	BOOL			
			17	%IX3.7	BOOL			

Рис. 4.53. Поля ввода значений для дискретных входов

PU D016 I/O Mapping	Find	Filter Show	all			 Add FB for IC 	Chann	el Go to In
tatus	Variable	Mapping	Channel	Address	Туре	Default Value	Unit	Description
	_		QB(I)	%QB1	USINT			
formation	Application.PLC_PRG.Var2output	۵	Q0	%QX1.0	BOOL			
	Application.PLC_PRG.Var3output	~	Q1	%QX1.1	BOOL			
	Application.PLC_PRG.Var4output	۵	Q2	%QX1.2	BOOL			
			Q3	%QX1.3	BOOL			
	* @		Q4	%QX1.4	BOOL			
	*		Q5	%QX1.5	BOOL			
	* @		Q6	%QX1.6	BOOL			
			Q7	%QX1.7	BOOL			
	ii- **		QB(II)	%QB2	USINT			
	Application.PLC_PRG.Var5output	~	Q0	%QX2.0	BOOL			
	Application.PLC_PRG.Var6output	~ >	Q1	%QX2.1	BOOL			
	Application.PLC_PRG.Var7output	۵	Q2	%QX2.2	BOOL			
	🍫		Q3	%QX2.3	BOOL			
	* @		Q4	%QX2.4	BOOL			
	* ø		Q5	%QX2.5	BOOL			
	* @		Q6	%QX2.6	BOOL			
			07	%OX2.7	BOOL			

Рис. 4.54. Установленные значения для дискретных входов

Напишите условие для тумблера SA1. Если он будет включен, то загорятся первые три реле, если выключен, то вторые три реле.

Подсказкой для написания кода служит окно, расположенное справа от программы. В нем отражены все логические операции для написания кода. На данном этапе потребуется логическая операция If, которую можно вынести в тело программы, нажав на нее ЛКМ (рис. 4.55).

ToolBox	ĸ	↓ ₽
Search.		1 10 1
▶ My	Favorite	
🖌 Bas	ic Instructions	
= 눨	ST Sentence	
	♦ IF	Insert "
	FOR	Insert "
	WHILE	Insert "
	 CASE 	Insert "
	REPEAT	Insert "
	CONTINUE	Jump to
	♦ JMP	Jump to
	 EXIT 	Jump o
	RETURN	Return(
± 🚞	LogicOperations	
± 🚞	Timers	
± 🚞	Counters	
🗄 📴	Math Functions	
🗄 📴	Data Process	
± 칠	Data Conversion	
± 칠	Data Shiftment	
± 칠	Selection	
± 🚞	Comparers	

Puc. 4.55. ST Sentence

Загрузите проект в контроллер (рис. 4.56, 4.57) и запустите программу.

Включите/выключите тумблер SA1, должны загореться лампочки на реле.

Разберем логическую операцию NOT (Инверсия сигнала). Добавьте ее в код и запустите программу. Теперь, когда тумблер выключен, горят первые три реле, а когда включен – следующие три реле.



Рис. 4.56. Код с присвоенными значениями

Рис. 4.57. Код с присвоенными значениями с инверсией сигнала

Рассмотрим логическую операцию «Таймер» (TON) (рис. 4.58). Перетащите ее ЛКМ в проект «Таймер» и во всплывающем окне задайте имя таймера (рис. 4.59).

Рис. 4.58. Логическая операция «Таймер»

Auto Declare		×
Scope	Name TON_0	Type TON V >
Object PLC_PRG [Application]	Initialization	Address
Flags CONSTANT RETAIN PERSISTENT	Comment	^ ~
		OK Cancel

Рис. 4.59. Выбор значения таймера

После добавления логической операции TON в проект появится функциональный блок таймера со входами, которые отображаются «:=», и выходами « →». На каждый вход и выход можно добавить свою переменную и отслеживать статус выполнения функционального блока (рис. 4.60).

TON_0(IN:= , PT:= , Q=> , ET=>);

Рис. 4.60. Установка переменных ТОЛ

Для работы таймера принимаем условие: все реле должны включиться через 5 с. Для этого измените ранее написанный код (рис. 4.61).

Рис. 4.61. Финальная версия кода

Значение IN необходимо для включения таймера. В нашем случае таймер будет включатся по тумблеру. Функции, которые можно задать в логической операции TON:

РТ – переменная для установления времени, в течение которого будет работать таймер.

Q – битовая переменная (отображение статуса таймера), которая позволяет после отсчета таймера изменять значения «False» и «True».

ET – переменная, показывающая время на таймере в данный момент.

Содержание работы

1. Цель работы.

- 2. Задания.
- 3. Ход работы.
- 4. Выводы.
- 5. Контрольные вопросы.

Средства, используемые при выполнении лабораторной работы

1. Методические указания к выполнению лабораторных работ.

2. Данные, предоставленные преподавателем во время занятия.

3. При проведении анализа допускается использование глобальной сети Интернет.

Контрольные вопросы

1. Что такое LocalBus config в проекте?

2. Для чего нужен Mapping в модулях?

3. Как загрузить проект в контроллер?

4. Что такое логические операции в CoDeSys?

5. Что делает логическая операция NOT?

6. Для чего нужны входы и выходы функционального блока таймера (TON)?

Лабораторная работа № 3

ЗАПУСК ДВИГАТЕЛЯ ПО СКОРОСТИ

Цель работы: научиться программировать запуск двигателя по скорости.

Задания

- 1. Запрограммировать инициализацию привода.
- 2. Рассчитать коэффициенты редукции.
- 3. Запрограммировать включение/выключение привода по питанию.
- 4. Запрограммировать движение привода по скорости.
- 5. Протестировать написанный код на учебном стенде Inovance.

Ход работы

Написание кода для запуска двигателя по скорости

Для добавления протокола EtherCAT необходимо перейти во вкладку Network configuration (рис. 4.62) и поставить галочку напротив EtherCAT master.

Puc. 4.62. Network configuration

В дереве проекта появятся новая задача EtherCAT и вкладка EtherCAT (рис. 4.63). Задача EtherCAT нужна для опроса всех устройств. Данный протокол является самым приоритетным в проекте.

Puc. 4.63. EtherCAT

Для дальнейшей работы необходимо просканировать все устройства (рис. 4.64). Для этого нажмите ПКМ на вкладку EtherCAT и выберите «Сканировать устройства».

Если при сканировании возникнет ошибка, нужно загрузить проект в контроллер и выйти из него.

Если нет физического доступа к контроллеру для сканирования, можно вручную добавить приводы в проект. Во вкладке Network configuration в правой части необходимо добавить два привода SV660_1Axis.

Devices 👻	a ;	× / 📄 PLC_PRG 🛛 🛞 I	Netw	vork Configuration	n X				[
🗏 📄 Untitled 1		🔹 🗟 😽 🖉 Refresh	e Pa	ste 🔒 Delete 🖞	Undo 🛃 Redo 🛛 🔘	mport EDS File 📫 Import G	iSD File 🕕 Import ECT File 🛛 🔍 Zo	om In 🔍 Zoom Out 🛛 80			
E Movice (AM600-CPU 1608TP/TN)				N ROM 0							
Device Diagnose			1-94		Modbus Master	Modbus Slave	Free Protocol				
Wetwork Configuration		<u>گ</u>		COM 1							
EtherCAT Config		Ų]Modbus Master	🔲 Modbus Slave	Free Protocol				
LocaBus Config		ň.	s	can Devices							×
- I PLC Logic		Ů									
Application				Scanned Devices							
				Devicename	Devicetype						
Task Configuration											
ETHERCAT											
- A) ETHERCAT.EtherCAT Ta	ask										
😑 🥩 MainTask											
PLC_PRG											
- 💷 Resources List											
SoftMotion General Axis Pool											
HIGH_SPEED_IO (High Speed IO Modu	ule)										
MODBUS_TCP (ModbusTCP Device)											
ETHERCAT (EtherCAT Master SoftMotion	a Pa p	Pacto									
		Velete									
· · · · · · · · · · · · · · · · · · ·		Joele									
	R	Refactoring									
G.	à P	roperties									
1	5a 🖌	Add Object									
	5	Add Folder									
	1	Idd Davisa									
-		du Device									
L	5	can For Devices									
		Disable Devices									
		Jpdate Device									_
	ĵE	dit Object							Show	Difference	es
	E	dit Object With						0 10 1 1			
	E	dit IO mapping		Scan Devices	Scan TimeOut 55	S		Copy all to project			R.
		moort mannings from CSV									
		Super Company in the COV									
		export mappings to CSV									
	0	Collapse Application									

Рис. 4.64. Добавление приводов (сканирование)

После сканирования все подключенные устройства появятся во всплывающем окне (рис. 4.65). Нажмите кнопку Copy all to project. Все устройства добавятся в дерево EtherCAT.

anned Devices							
Devicename InoSV660N	Devicetype SV660_1Axis_00915	Alias Address	Write Alias Address	AutoIncAddress 0			
antine Address	1				Shor	v Differer	10

Рис. 4.65. Результат сканирования

Для управления сервоприводами SV660N необходимо разработанную (разрабатываемую) программу поместить в цикл задачи Ether-CAT, как показано на рис. 4.66. Для удобства дальнейшей работы с осями сервомоторов/серводвигателей переименуйте двигатель (рис. 4.66), например Drive.

=	erCAT SV66 Lvis)	Master SoftMotion) 0_1Axis_00915)	1					
	Ē	Paste						
	\times	Delete						
		Refactoring	Ì	Rename 'Axis'	Rename		×	
	Ē.	Properties						
		Add Object			Current name:	Axis		
	6	Add Folder			New name:	Drive		
		Plug Device						
		Disable Devices						
		Update Device					<u>O</u> K <u>C</u> ancel	
	ſ	Edit Object						
		Edit Object With						
		Edit IO mapping						
		Import mappings from CSV						
		Export mappings to CSV						
		Collapse Application						

Рис. 4.66. Присвоение имени двигателю

Как только будет нажато подтверждение на переименование, на экране появится всплывающее окно с подтверждением изменения имени двигателя во всем проекте (рис. 4.67).

Buttlet Optimized Type Nuel Value
Definition Type Wate Value Addres Description • Definition • Output Outpu<

Рис. 4.67. Окно подтверждения имени двигателя

Приступим к изучению привода. С помощью двойного щелчка ЛКМ перейдите на настройку двигателя.

Во вкладке General отображены основные настройки двигателя (рис. 4.68):

1) выбор режима управления (Modulo – круговой/абсолютный; Finite – линейный/векторный);

2) активация лимитов для привода (скорость, дистанция и т. д.);

- 3) рампа разгона (по умолчанию выставлена трапеция) (рис. 4.69);
- 4) лимиты для системы с ЧПУ.

General Setting	Axis type and Limits	5	
Scaling	Virtual mode	Software limits	Velocity ramp type
Homing Setting		Activate Negative 0.0 pulse	Trapozoid
Mapping		Positive 1000.0 pulse	O Quadratic
Commissioning		Over limits reaction	Quadratic(smooth)
SM_Drive_ETC_GenericDSP402: I/O Mapping	• Finite 1	Deceleration 1000 pulse/s ²	Identification
SM_Drive_ETC_GenericDSP402: IEC Objects		Max Distance 10 pulse	ID 0
Status	- CNC Dynamic lin	nits	Position lag supervision
Information	Vel: pulse/s	Acc: pulse/s ² Dec: pulse/s ² Jerk: pulse/s ³ 1000 1000 10000 4	Deactivated \checkmark Lag limit: 1.0 pulse

Рис. 4.68. Основные свойства двигателя

Рис. 4.69. Рампа разгона «трапеция» и «квадратичная»

Вкладка Scaling определяет (рис. 4.70):

1) выбор единиц измерения;

2) выставление разрешения энкодера за оборот;

3) выбор рабочего расстояния за один оборот двигателя в случае, если известен коэффициент редукции исполнительного вала с механизмом;

4) расчет коэффициента редукции (рис. 4.71).

Вкладка Homing Settings используется для выхода в ноль по энкодеру. Существуют различные вариации выхода в ноль по энкодеру: по z-сигналу, концевому выключателю и т. д.

General Setting	Unit in application
Scaling	pulse () mm () um () nm () degree () inch
Homing Setting	Invert Direction
Mapping	Command pulse count per motor rotation 16#100000 pulse/rev 👱
Commissioning	Do not use gearbox
SM_Drive_ETC_GenericDSP402: I/O Mapping	Vork traver distance per motor rotation
SM_Drive_ETC_GenericDSP402: IEC Objects	Command pulse count per motor rotation [DINT]
Status	Number of pulses [pulse] = * Travel distance [Unit in application] 3 Work travel distance per motor rotation [LREAL]
Information	Use gearbox Work travel distance per work rotation (Please refer to the Modulo value in General Setting if the Axis type is Modulo mode)
	Numerator of the gear ratio (the number of teeth (5) in the following picture) 1 Denominator of the gear ratio (the number of teeth (4) in the following picture) 1
	The Axis type is Linear mode
	Reference: Unit Conversion formula Command pulse count per motor rotation [DINT] Numerator of the gear ratio [DINT] Numerator of the gear ratio [DINT]
	Work travel distance per work rotation [LREAL] Denominator of the gear ratio [DINT]
	M: Motor, W: Work

Рис. 4.70. Вкладка Scaling

General Setting	Homing Setting
Scaling	Homing methods Homing Method1 V
Homing Setting	Homing Vel 1.6666667 pulse/s Acceleration 166.6666666 pulse/s ²
Mapping	Homing Crawl Vel 0.166667 pulse/s Time Limit 50000 *10ms
Commissioning	
SM_Drive_ETC_GenericDSP402: I/O Mapping	Mater 7 Circul
SM_Drive_ETC_GenericDSP402: IEC Objects	Negative limit switch
Status	
Information	
	Deceleration point signal is invalid
	Deceleration point

Рис. 4.71. Расчет коэффициента редукции

Для программирования выберите единицу измерения – миллиметры, мм (рис. 4.72).

Unit in application					
🔘 pulse	🔘 mm	◯um	🔾 nm	⊖ degree	◯ inch

Рис. 4.72. Выбор единицы измерения

Следует выставить разрешение энкодера и пройденное расстояние исполняющего механизма за оборот. На всех приводах SV660N стоят 23-битные энкодеры. Для этого в поле нужно ввести 800 000, как показано на рис. 4.73, иначе привод будет работать некорректно.

Travel Distance		
Invert Direction		
Command pulse count per motor rotation	16#800000	pulse/rev
O not use gearbox		
Work travel distance per motor rotation	10	mm/rev

Рис. 4.73. Выбор разрешения энкодера

Вкладка самого привода состоит из вкладок, отраженных на рис. 4.74:

1) автоматическая нумерация и время вызова в программе;

2) PDO – параметры, которые можно изменять во время работы привода. На каждый параметр выделяется уникальный адрес (рис. 4.75);

3) SDO – параметры, которые записываются однократно при включении программы в контроллере (рис. 4.75).

Рис. 4.74. Вкладка привода

Process Data(PDO Setting) Startup parameters(SDO Setting) Online AutoInc Address EtherCAT Address IDI Online CoE Online EoE settings Servo Function Code EtherCAT IJO Mapping EtherCAT IEC Objects Status Information Outser Defined Information In
Online
CoE Online Select DC DC-Synchron EoE settings Imable 4000 Sync Unit Cycle (µs) Servo Function Code Imable Imable Sync 0 Imable EtherCAT I/O Mapping Imable Imable Sync 0 Imable Imable Sync 0 Status Imable Imable Sync 1 Imable Sync 1 Imable Sync 1 Imable Sync 1 Imable Sync 0 Imable Sync 1 Imable Sync 1 </th
EoE settings ✓ enable 4000 Sync Unit Cycle (µs) Servo Function Code ✓ Enable Sync 0 ✓ Enable Sync 0 EtherCAT I/O Mapping ✓ Sync Unit Cycle × 1 4000 ♀ Cycle Time (µs) EtherCAT IEC Objects ✓ User Defined ○ ♀ Shift Time (µs) Status Information ✓ Sync Unit Cycle × 1 4000 ♀ Cycle Time (µs) User Defined ○ ♀ Shift Time (µs) ✓ User Defined ○ ♀ Shift Time (µs) Information ✓ User Defined ○ ♀ Shift Time (µs) ✓ Identification Identification (Alias can be set through the "Overview" interface of the master station) —
Servo Function Code Sync0: EtherCAT I/O Mapping
EtherCAT I/O Mapping Sync Unit Cycle user Defined user Defined shift Time (µs) Status Information Sync Unit Cycle x 1 Q 2 Shift Time (µs) Information Quser Defined Quser Defined Quser Defined Quser Defined Quser Defined Quser Defined Shift Time (µs) Identification (Alias can be set through the "Overview" interface of the master station) Disabled
EtherCAT IEC Objects Status Information User Defined 0
Status Sync1: □ Enable Sync 1 □ Sync Unit Cycle x 1 4000 Cycle Time (µs) □ User Defined 0 Shift Time (µs) Identification (Alias can be set through the "Overview" interface of the master station) -
Information Sync Unit Cycle x 1 4000 Cycle Time (µs) User Defined Identification (Alias can be set through the "Overview" interface of the master station) — Disabled
User Defined 0
Identification (Alias can be set through the "Overview" interface of the master station) — O Disabled
Disabled
Configured Station Alias (ADO 0x0012) 1001 €
Explicit Device Identification (ADO 0x0134)
Data word (2 bytes) ADD (nex) 0

Ниже представлены варианты программирования функционального кода и значения скорости (табл. 4.4).

Таблица 4.4

Порядковый номер	Скорость сервопривода, об./с
1	15
2	25
3	30
4	45
5	55
6	50
7	20
8	35
9	40
10	60

Программирование функционального кода

Программирование будет осуществляться через логическую операцию CASE.

Оператор CASE – это оператор множественного ветвления. В зависимости от некоторого условия может быть выполнено одно из многочисленных продолжений программы.

Для включения привода потребуется:

1) переменная, имеющая тип INT, для переключения шага;

2) функциональный блок MC_Power для включения питания привода;

3) функциональный блок MC_MoveVelocity для управления приводом по скорости;

4) функциональный блок MC_Halt для корректного останова.

Все функциональные блоки находятся во вкладке Motion Control (рис. 4.76, 4.77).

Puc. 4.76. Вкладка Motion Control

PLC_	PRG 🗙 🍳 Device Diagnosis 🔗 Network Configuration 🏻 🗃 InoSV660N 🛛 🕫 Drive
1	PROGRAM PLC_PRG
2 🖓	VAR
3	step:INT; //IIar
4	MC_Power_0: MC_Power; //Блок питания привода
5	MC_MoveVelocity_0: MC_MoveVelocity;//Блок запуска двигателя по скорости
6	MC_Halt_0: MC_Halt;//Блок останова двигателя
7	END_VAR
-	

Рис. 4.77. Функциональные блоки

На нулевом шаге при включении будет инициализироваться привод (рис. 4.78). Без его загрузки дальнейшее движение будет невозможно. После того как привод загрузится на 100 %, перейдите на шаг 1 (step:=1).

Рис. 4.78. Инициализация привода

На первом шаге включается привод по питанию (рис. 4.79). Функциональный блок включает питание, выключает тормоза, включает модуляцию у привода.

```
1: //Включение привода по питанию
MC_Power_0 (Axis:= Drive, Enable:= TRUE, bRegulatorOn:=TRUE, bDriveStart:= TRUE, );
IF MC_Power_0.Status THEN//Если на привод пришло питание,перезодим на следующий шар
step:=2;
END IF
```

Рис. 4.79. Управление приводом по питанию

Выходное значение Status сигнализирует о том, что функциональный блок отработал. Далее происходит переход к шагу 2 (step:=2).

Второй шаг – управление приводом по скорости (рис. 4.80). Функциональный блок MC_MoveVelocity получает входные значения, а именно:

Axis – ось вращения;

Execute – разрешение на включение блока;

Velocity – скорость, с которой привод будет работать;

Acceleration – разгон (в единицах измерения);

Deceleration – торможение (в единицах измерения);

Jerk – толчок (в единицах измерения), влияет на плавность разгона рампы;

Direction – направление вращения вала двигателя.

```
2: //Включение управления привода по скорости
MC_MoveVelocity_0 (Axis:= Drive, Execute:= TRUE, Velocity:= 50, Acceleration:=50, Deceleration:=50, Jerk:= 50, Direction:=positive ,);
3: //Ocтанов привода
MC_Halt_0 (Axis:= Drive,Execute:= TRUE,Deceleration:= 50,Jerk:= 10);
END_CASE
```

Рис. 4.80. Управление приводом по скорости

Третий шаг – останов двигателя. Функциональный блок MC_Halt имеет следующие входные значения:

Axis – ось вращения;

Execute – разрешение на включение блока;

Deceleration – торможение (в единицах измерения);

Jerk – толчок (в единицах измерения), влияет на плавность разгона рампы.

После написания кода загрузите проект в контроллер и запустите программу.

В режиме онлайн-отслеживания можно открыть вкладку двигателя. Во вкладке появится дополнительная колонка с отслеживанием параметров и ошибок (рис. 4.81).

General Setting	-			Quadratic
Scaling	C Sinite	Over limits r	eaction	Quadratic(smooth)
Homing Setting	I Finite		Max Distance	Identification
Mapping			Wax Distance	
Commissioning	CNC Dynamic lin	nits		Position lag supervision
SM_Drive_ETC_GenericDSP402: I/O Mapping	Vel: mm/s 30	Acc: mm/s	² Dec: mm,	In/s² Jerk: mm/s³ 1000 10000 Lag limit: 1.0
SM_Drive_ETC_GenericDSP402: IEC Objects				
Status	Online			
Information	Variable	Set Value	Actual Value	Status: SMC_AXIS_STATE.stopping
	Position	133.22	133.22	2 Communication: Operational (100)
	Velocity	0.00	0.00	0
	Torque	0.00	0.00	Diagnosis Errors
	Torque	0.00	0.00	Error ID:
				SMC_NO_ERROR(0)
				Error Source:
				Error Explain:
	Error Historic	Records		
	Time	Error IL)	Solution:

Рис. 4.81. Вкладка Марріпд

Для того чтобы произвести останов двигателя, необходимо в режиме онлайн в столбце Prepared value выставить шаг 3 (рис. 4.82) и нажать CTRL+F7 для записи значений либо зайти сверху во вкладку Debug и нажать Write values.

<u>Tools Window Help</u> X Ma \$4:1/2 //2 / Ma \$4:5 16: 11 / 16:11 / 1				
PLC_PRG X Q Device Diagnosis 🛞 Network Configuration 🕋 InoSV660N 👦 Drive				
Device.Application.PLC_PRG				
Expression	Туре	Value	Prepared value	Address
Ø step	INT	0	3	
MC_Power_0	MC_Power			
MC_MoveVelocity_0	MC_MoveVelocity			
🛷 setVel	LREAL	0		
ø buffer	LREAL	0		
🛷 Turn	UINT	0		
A pag	LREAL	0		
e pos				

Рис. 4.82. Окно с общим видом запрограммированных действий

Аналогичные действия выполняют со вторым приводом. Для каждого привода нужны новые функциональные блоки управления и, соответственно, новый код.

Содержание работы

- 1. Цель работы.
- 2. Задания.
- 3. Ход работы.
- 4. Выводы.
- 5. Контрольные вопросы.

Средства, используемые при выполнении лабораторной работы

1. Методические указания к выполнению лабораторных работ.

2. Данные, предоставленные преподавателем во время занятия.

3. При проведении анализа допускается использование глобальной сети Интернет.

Контрольные вопросы

1. Для чего нужна задача EtherCAT?

- 2. Как просканировать все устройства?
- 3. Что нужно делать, если появилась ошибка при сканировании?
- 4. Чем отличается режим управления Modulo от Finite?
- 5. Чем различаются рампы разгона «трапеция» и «квадратичная»?
- 6. Какой энкодер стоит на двигателе?
- 7. Чем отличаются PDO параметры от SDO параметров?
- 8. Что такое оператор CASE?

9. Какие функциональные блоки использовались в программе, разработанной в ходе лабораторной работы?

10. Какие вводные параметры используются для функционального блока MC_MoveVelocity? Что они помогают выполнять?

Лабораторная работа № 4

СЕРВОПРИВОДЫ С АБСОЛЮТНЫМ УПРАВЛЕНИЕМ

Цель работы: освоить способ программирования сервоприводов с абсолютным управлением.

Задания

1. Запрограммировать два сервопривода.

2. Протестировать написанную программу на учебном стенде Inovance.

Ход работы

Программирование сервоприводов с абсолютным управлением

Создайте проект с приводами, придерживаясь той же последовательности, что и в лабораторной работе № 3. Выберите привод.

Выставьте управление по позиции (Modulo) и укажите значение Modulo value – 360 градусов (рис. 4.83).

General Setting	Axis type and Limits				
Scaling	Virtual mode	Modulo settings		Velocity r	amp type
Homing Setting		Modulo value:	360.0 degree	Trapo O sin ²	ozoid
lapping	Modulo			O Quad	dratic
Commissioning		Over limits reaction	[]	O Quad	dratic(smooth)
M_Drive_ETC_GenericDSP402: I/O Mapping	○ Finite	Deceleratio	n 1000 de	gree/s [:] Identifica	tion
SM_Drive_ETC_GenericDSP402: EC Objects		Max Distar	ice 10 deg	gree ID 0	
tatus	CNC Dynamic lim	ts		Position I	ag supervision
information	Vel: degree/s	Acc: degree/s ² Dec: 0	degree/s² Jerk: degree	e/s ³ Deactiv 10 Lag limit:	1.0 degree

Рис. 4.83. Установка значения Modulo value

Во вкладке Scaling для программирования выберите единицы измерения – градусы (degree). Необходимо выставить разрешение энкодера и количество градусов за оборот для исполняющего механизма (рис. 4.84). На всех приводах SV660N стоят 23-битные энкодеры. Для этого в поле следует ввести 800 000, как показано на рис. 4.84, иначе привод будет работать некорректно.

Travel Distance		
Invert Direction		
Command pulse count per motor rotation	16#800000	pulse/rev
Do not use gearbox		
Work travel distance per motor rotation	360	degree/rev

Рис. 4.84. Установка значения энкодера

Будем изменять положение сервопривода с помощью штурвала, который расположен на стенде. Для этого во вкладке высокоскоростных входов выберите счетчик «0» и выставьте название счетчика (рис. 4.85).

Hardware Port Configuration		Filt	er(us)	Counter0		Axis0			1
	Counter 0 A Phase — X	(0 2		Mode	A/B Phase Multiple 4 $$	Output Mode	cw/ccw ~	YO	
Counter Parameters				Coincident Output	None \checkmark	Home Method	Marked 0		
Axis Parameters	Counter 0 B Phase — X	(1 2		External Input	X8	Home Method	Method 0 V	¥1	General Output
Internal I/O Manning	Constal Input X	2 200	000	Counters		Axis1			
Internal your apping	General input	200		Mode	A/B Phase Multiple 4 $$	Output Mode	CW/CCW ~	¥2	General Output
Status	General Input — X	(3 200	000	Coincident Output	None \checkmark	Home Method	Method 0 🗸 🗸	V2	Ganaral Output
Information				External Input	X9			15	General Output
	General Input X	(4 200	000	Counter2		AXIS2			C
		_		Mode	A/B Phase Multiple 4 $$	Output Mode	CW/CCW ~	Υ4	General Output
	General Input — X	(5 200	000	Coincident Output	None ~	Home Method	Method 0 🗸 🗸	V5	General Output
	Constal Insuit V	6 200	000	External Input					ocherar output
	General input	200	000	Counter3		AXISS		V6	General Output
	General Input — X	(7 200	000	Mode	A/B Phase Multiple 4 $$	Output Mode	CW/CCW ~	10	- General Output
				Coincident Output	None ~	Home Method	Method 0 \sim	¥7	
	General Input — X	(8 200	000	External Input	XB			l	
				Counter4		- Interrupt Input-			1
	General Input — X	(9 200	000	Mode	A/B Phase Multiple 4 $$	🗆 X0 于 🕹	ft □ X1 f t	A.	
	General Input — X	(A 200	000	Coincident Output	None ~	🗆 X2 🕤 같		R	
	· · · · ·			Counter5				41	
	General Input — X	(B 200	000	Mode	A/B Phase Multiple 4 🗸	L X4 🗾 Ł]t . x5 . t	14	
				Coloridate Outrate		🗆 X6 🥤 🕹	ft □x7 f t	R.	
	General Input — X	(C 200	000	Concident Output	None	- PWM Output			
				Counter6		- win output			
	General Input — X	(D 200	000	Mode	A/B Phase Multiple 4 $$	L YU	L] ¥1		
				Coincident Output	None v	□ Y2	🗌 Y3		
	General Input — X	(E 200	000	Counter7		□ Y4	Y5		
	Constant N	(E 200	000	Mode	A/B Phase Multiple 4 $$	☐ Y6	🗌 Y7		
	General Input — X	200	000	Coincident Output	None	Default			

Рис. 4.85. Выбор счетчика

Во вкладке «Настройки счетчика» (рис. 4.86) выберите формат – линейный, период опроса – 10 мс.

Counter Parameters	General			Frequency/Rotation Mea	sure
Axis Parameters	Name: HS_Co	unter0 Ty	/pe: COUNTER_REF	Period:	10 ms \checkmark
Internal I/O Mapping	Counter Forma	t: 🔘 Linear	○ Ring	Pulse Per Rotation:	1
Status	-Externel Trigger	(X8)			Default
Information	Input Logic:	Positive	Negative		
	Function:	None	\sim		

Рис. 4.86. Настройка счетчика

Ниже представлены варианты программирования функционального кода и значения скорости сервопривода (табл. 4.5). Варианты:

Таблица 4.5

Порядковый номер	Скорость сервопривода, об./с
1	120
2	125
3	130
4	145
5	155
6	150
7	120
8	135
9	140
10	160

Программирование функционального кода

Программирование будет выполняться через логическую операцию CASE (рис. 4.87).

Для включения привода потребуется:

1) переменная, имеющая тип INT, для переключения шага;

2) функциональный блок MC_Power для включения питания привода;

3) функциональный блок MC_MoveAbsolute для управления приводом по положению;

4) функциональный блок MC_Halt для корректного останова;

5) функциональный блок HS_Counter для считывания значений со штурвала.

Рис. 4.87. Программирование включения привода

Далее выполните действия в соответствии с рис. 4.78 – 4.80. После чего задайте настройки в соответствии с рис. 4.88, 4.89.

Рис. 4.88. Добавление

	блока для штурвала	
Auto Declare		×
Scope VAR Object PLC_PRG [Application] Elags CONSTANT RETAIN PERSISTENT	Name HC_Counter_0 Initialization ✓ Comment	<u>Type</u> HC_Counter ✓ > Address
		OK Cancel

Рис. 4.89. Выставление значений в блоке

В блоке необходимо выставить входные и выходные параметры: Counter – название счетчика из высокоскоростных входов;

Enable – включение блока (поставьте условие: когда на привод будет подано питание, тогда будет включаться блок);

Direction – направление вращения (когда направление пустое, может двигаться в любом направлении);

CounterValue – выходное значение счетчика. Значение градуса поворота штурвала, привязанного к сервоприводу. Задайте ранее созданную переменную «setPos» (рис. 4.90).

//блок счетчика HC_Counter_0(Counter:=HS_Counter0,Enable:= MC_Power_0.Status,Direction:= ,CounterValue=> setPos);

Рис. 4.90. Команда «setPos»

Для изменения позиции необходимо выключить функциональный блок и включить его заново, задав новые значения.

Выполните следующие действия:

Создайте переменную-буфер (рис. 4.91) и сравните значения текущей позиции с заданной. Для выключения и включения блока используйте входную переменную блока Execute.

После выполненных действий необходимо приравнять значения буфера к заданным значениям, чтобы программа до следующего изменения положения не заходила в код.

```
//Буффер для сравнения
//
```

Рис. 4.91. Буфер

Далее выполните действия в соответствии с рис. 4.81 – 4.82.

Аналогичные действия выполняют со вторым приводом. Для каждого привода нужны новые функциональные блоки управления и, соответственно, новый код.

Содержание работы

- 1. Цель работы.
- 2. Задания.
- 3. Ход работы.
- 4. Выводы.
- 5. Контрольные вопросы.

Средства, используемые при выполнении лабораторной работы

1. Методические указания к выполнению лабораторных работ.

2. Данные, предоставленные преподавателем во время занятия.

3. При проведении анализа допускается использование глобальной сети Интернет.

Контрольные вопросы

- 1. Что такое Modulo в приводе?
- 2. Какое разрешение имеют энкодеры в приводах SV660N?
- 3. Из чего состоит функциональный блок HS_Counter?
- 4. Для чего служит переменная-буфер?

Лабораторная работа № 5

УПРАВЛЕНИЕ СЕРВОПРИВОДАМИ С ПОТЕНЦИОМЕТРА И ПАНЕЛИ

Цель работы: научиться программировать сервоприводы с потенциометра и панели.

Задания

1. Изучить потенциометр.

2. Создать интерфейс управления сервоприводами.

3. Запрограммировать управление сервоприводом с потенциометра.

4. Выполнить выданное преподавателем итоговое задание по примеру лабораторной работы № 2.

Ход работы

Разработка и создание реального проекта

Создайте проект и добавьте протокол EtherCAT во вкладку Network configuration, выполните действия, до этого указанные на рис. 4.62.

Если нет физического доступа к контроллеру для сканирования, можно вручную добавить приводы в проект. Во вкладке Network configuration в правой части необходимо добавить все устройства, которые установлены на стенде, в соответствии с инструкциями, представленными на рис. 4.64 – 4.65.

Все устройства добавятся в дерево EtherCAT. После сканирования дерево проекта должно выглядеть так, как на рис. 4.92.

Рис. 4.92. Устройства в проекте

Для удобства работы с осями переименуйте двигатели, например: Drive1 и Drive2, в соответствии с рис. 4.66.

Перейдите к настройке привода, а затем двойным щелчком ЛКМ – к настройке двигателя.

Выставьте управление по скорости (Finite) (рис. 4.93).

General Setting	Axis type and Limit	S	
Scaling	Virtual mode	Software limits	Velocity ramp type
		Activate Negative 0.0 mm	🔿 Trapozoid
Homing Setting			€ sin²
Mapping	() Modulo	Positive 1000.0 mm	○ Quadratic
Commissioning		Over limits reaction	O Quadratic(smooth)
SM_Drive_ETC_GenericDSP402: I/O	Finite	Deceleration 1000 mm/s ²	Identification
Mapping		Max Distance 10 mm	
SM_Drive_ETC_GenericDSP402: IEC Objects			
Status	CNC Dynamic lir	nits	Position lag supervision
Information	Vel: mm/s	Acc: mm/s ² Dec: mm/s ² Jerk: mm/s ³	Deactivated \lor
	30	1000 1000 10000	Lag limit: 1.0 mm

Рис. 4.93. Установка управления по скорости

Во вкладке Scaling для программирования выберите единицы измерения – мм.
Необходимо выставить разрешение энкодера в соответствии с рис. 4.73.

Затем перейдите во вкладку Homing setting и выберите метод выхода в ноль энкодера. Так как концевого выключателя нет, единственный метод, который можно выбрать, – это по z-сигналу. Этот метод имеет номера 33 и 34. Выберите метод 34 и выставьте скорость, ускорение и максимальное время нахождения нулевой точки (рис. 4.94).

General Setting	Homing Setting					
Scaling	Homing methods Homing Method34 ~					
Homing Setting	Homing Vel 100 mm/s Acceleration 166.6666666	mm/s²				
Mapping	Homing Crawl Vel 100 mm/s Time Limit 50000 *	*10ms				
Commissioning						
SM_Drive_ETC_GenericDSP402: I/O Mapping						
SM_Drive_ETC_GenericDSP402: IEC Objects	Motor Z signal	_				
Status	incol 2 sgila					
Information						
	Homing Method 34					

Рис. 4.94. Выбор метода

Переместите управляющую программу (рис. 4.95) в задачи EtherCAT.



Рис. 4.95. Перемещение управляющей программы

Программирование функционального кода

Напишите код, по которому приводы будут включаться, управляться по скорости с помощью потенциометра, выходить в ноль, останавливаться, выключаться (рис. 4.96). Все это управление реализуется на панели.

Варианты для выполнения лабораторной работы представлены в табл. 4.6.

Таблица 4.6

Номер варианта	Значение скорости выхода в ноль	Рампа разгона привода
1	120	Трапеция
2	100	Синусоидальная
3	115	Квадратичная
4	90	»
5	105	Синусоидальная
6	110	Трапеция
7	120	Квадратичная
8	125	Синусоидальная
9	105	»
10	100	Трапеция

Варианты заданий

/ 🌆 Ha	rdware Configuration 🖉 PLC_PRG 🗙 🐙 Drive1 🐨 Drive2 🔐 GL10_4AD 🔐 GL10_RTL
1	PROGRAM PLC_PRG
2 📮	VAR
3	//Программные переменные
4	BItStart:BOOL; //Команда на разрешения работы
5	step:INT; //IIar
6	setVel:LREAL;//Задание скорости
7	buffer:LREAL;//Буфер для сравнения
8	InValueVel:INT;//Прием значения с аналогово входа
9	Connectionl:WORD;//Переменная для отслеживания статуса загрузки первого привода
10	Connection2:WORD;//Переменная для отслеживания статуса загрузки второго привода
11	Speedl:LREAL;//Переменная отслеживания скорости 1 привода
12	Speed2:LREAL;//Переменная отслеживания скорости 2 привода
13	Power:BOOL;//Переменная для отслеживания статуса включения питания
14	Invelocity:BOOL;//Переменная для отслеживания статуса выхода на скорость
15	Alarm:BOOL; //Авария приводов
16	//Функциональные блоки
17	MC_Power_0: MC_Power; //Блок питания 1 привода
18	MC_Power_1: MC_Power;//Блок питания 2 привода
19	MC_MoveVelocity_0: MC_MoveVelocity;//Блок запуска 1 двигателя по скорости
20	MC_MoveVelocity_1: MC_MoveVelocity;//Блок запуска 2 двигателя по скорости
21	MC_Home_0: MC_Home;//Блок выхода в ноль 1 привода
22	MC_Home_1: MC_Home;//Блок выхода в ноль 2 привода
23	MC_Reset_0: MC_Reset;//Блок перезагрузки 1 привода
24	MC_Reset_1: MC_Reset;//Блок перезагрузки 2 привода
25	MC_Halt_0: MC_Halt;//Блок останова 1 привода
26	MC_Halt_1: MC_Halt;//Блок останова 2 привода
27	END_VAR

Рис. 4.96. Объявление переменных

В программе используются следующие блоки:

MC Power – блок включения питания;

MC_MoveVelocity – блок управления по скорости;

МС Ноте – блок выхода в ноль;

MC_Reset – блок перезагрузки привода;

MC_Halt – блок корректного останова.

В теле программы создайте условие для активации кейса.

На нулевом шаге при включении будут инициализироваться приводы (рис. 4.97). Без их загрузки дальнейшее движение будет невозможно. После того как приводы загрузились на 100 %, перейдите на шаг 1.

```
1 //Кнопка разрешения работы
2 ☐ IF BItStart THEN//Если кнопка нажата то выполняется кейс
3 4 ☐ CASE step OF
5 0: //Инициализация приводов
6 ☐ IF Drivel.wCommunicationState=100 AND Drive2.wCommunicationState=100 THEN
7 step:=1;
8 END_IF
```

Рис. 4.97. Инициализация приводов

Первый шаг – включение приводов по питанию (рис. 4.98). Функциональный блок включает питание, выключает тормоза, включает модуляцию привода.

```
10 1: //Включение питания
11 MC_Power_0(Axis:= Drivel, Enable:= TRUE, bRegulatorOn:=TRUE, bDriveStart:= TRUE,);
12 MC_Power_1(Axis:= Drive2, Enable:= TRUE, bRegulatorOn:=TRUE, bDriveStart:= TRUE,);
13
```

Рис. 4.98. Включение приводов по питанию

Второй шаг – управление приводом по скорости (рис. 4.99).

```
14
2: //Управление по скорости
MC_MoveVelocity_0(Axis:= Drivel, Execute:= TRUE, Velocity:= setVel, Acceleration:=50, Deceleration:=50, Jerk:= 50, Direction:=positive);
MC_MoveVelocity_1(Axis:= Drive2, Execute:= TRUE, Velocity:= setVel, Acceleration:=50, Deceleration:=50, Jerk:= 50, Direction:=positive);
17
17
```

Рис. 4.99. Включение приводов по скорости

Третий шаг – выход в ноль (рис. 4.100). Перед тем как вызвать блок выхода в ноль, необходимо остановить привод.

```
18
 19 3: //Выход в ноль
 20
        MC_Halt_0(Axis:= Drivel,Execute:= TRUE,Deceleration:=50,Jerk:= 10);
        MC_Halt_1(Axis:= Drive2,Execute:= TRUE,Deceleration:=50,Jerk:= 10);
 21
        IF MC_Halt_0.Done THEN
 22 🖨
         MC_Home_0 (Axis:= Drivel,Execute:= TRUE,Position:= 0);
 23
           MC Home 1(Axis:= Drive2,Execute:= TRUE,Position:= 0);
 24
         IF MC_Home_0.Done AND MC_Home_1.Done THEN
 25 📥
 26
                 MC Home 0 (Axis:= Drivel, Execute:= FALSE);
 27
                MC_Home_1(Axis:= Drive2,Execute:= FALSE);
 28
                step:=l;
 29
            END IF
 30
 32
        END IF
```

Рис. 4.100. Выход в ноль

Блок MC_Ноте имеет следующие вводные параметры: Axis – ось, которая будет осуществлять выход в ноль; Execute – разрешение на работу блока; Position – позиция от заводского нуля. Четвертый шаг – перезагрузка приводов (рис. 4.101).

```
33 4: //Перезагрузка приводов
34
        MC Reset 0 (Axis:= Drivel, Execute:=TRUE);
35
       MC Reset 1 (Axis:= Drive2, Execute:=TRUE);
36 🗄
      IF MC_Reset_0.Done OR MC_Reset_1.Done OR MC_Reset_0.Error OR MC_Reset_1.Error THEN
37
         MC_Reset_0(Axis:= Drivel, Execute:=FALSE);
38
           MC_Reset_1(Axis:= Drive2, Execute:=FALSE);
39
           step:=l;
40
        END IF
41
```

Рис. 4.101. Перезагрузка приводов

Для перезагрузки приводов необходим функциональный блок MC_Reset. Его входные параметры:

Axis – ось, которую блок будет перезагружать; Execute – разрешение работы блока. Пятый шаг – корректный останов приводов (рис. 4.102).

42	5:	//Корректный останов
43		<pre>MC_Halt_0(Axis:= Drivel,Execute:= TRUE,Deceleration:=50,Jerk:= 10);</pre>
44		<pre>MC_Halt_1(Axis:= Drive2,Execute:= TRUE,Deceleration:=50,Jerk:= 10);</pre>
45 E		IF MC_Halt_0.Done AND MC_Halt_1.Done THEN
46		<pre>step:=l;</pre>
47		END_IF
48	_	

Рис. 4.102. Программирование корректного останова

Шестой шаг – выключение приводов	(рис. 4.103).
	(I)

49	6: //Выключение питания			
50	<pre>MC_Power_0(Axis:= Drivel,</pre>	<pre>Enable:= TRUE,</pre>	bRegulatorOn:=FALSE	<pre>bDriveStart:= FALSE)</pre>
51	<pre>MC_Power_1(Axis:= Drive2,</pre>	<pre>Enable:= TRUE,</pre>	bRegulatorOn:=FALSE	<pre>bDriveStart:= FALSE)</pre>
52				
53	END_CASE			
54				
55	END_IF			



Для выставления скорости с потенциометра необходимо добавить переменную в модуль аналогового входа (рис. 4.104).

Startup parameters(SD0 Setting)		
GenericConfiguration	Access - 0	
Device Diagnosis	AD Conversion Mode: -10V~10V(-20000~20000)	Filter Parameter: 8
Status	Offline Sign Overflow Sign	Peak Value Keeping
Information	4 Access - 1	
	Enable access	
	AD Conversion Mode: -10V~10V(-20000~20000) <	Filter Parameter: 8 \lor
	Offline Sign	Peak Value Keeping
	Access - 2	
	Enable access	
	AD Conversion Mode: 4mA~20mA(0~20000) V	Filter Parameter: 8 ~
	Offline Sign Overflow Sign	Peak Value Keeping
	Access - 3	
		Films Development of
		Deals Value Keeping
		Peak value keeping

Рис. 4.104. Установка значений модуля

Зайдите в модуль и переместитесь во вкладку Generic Configuration.

Потенциометр по электрической схеме подключен ко второму каналу и имеет сигнал 4 – 20 мА, поэтому выставьте активацию второго канала и прием сигнала (4 – 20 мА).

Перейдите в Coupler (рис. 4.105), который соединяет все локальные устройства, подключенные к нему, и зайдите во вкладку Mapping для добавления переменной на канал аналогового входа.



Рис. 4.105. Переход в коплер

Выберите второй канал, щелкнув в столбце Variable два раза ЛКМ. Нажмите на три точки для добавления переменной (рис. 4.106).

General	Find		Filter Show all			- Add F	B for IO	Channel Go to Instan
Process Data(PDO Setting)	Variable	Mapping	Channel	Address	Туре	Default Value	Unit	Description
			GL10_4DA DA CH0	%QW2	INT			GL10_4DA DA CH0
Startup parameters(SDO Setting)	😟 - 🍢		GL10_4DA DA CH1	%QW3	INT			GL10_4DA DA CH1
			GL10_4DA DA CH2	%QW4	INT			GL10_4DA DA CH2
Online	🖷 - 🍢		GL10_4DA DA CH3	%QW5	INT			GL10_4DA DA CH3
	😟 ··· 🦄		Device status	%IW2	UINT			Device status
CoE Online	😟 - 🍬		GL10_4AD AD CH0	%IW3	INT			GL10_4AD AD CH0
	😟 ··· 🦄		GL10_4AD AD CH1	%IW4	INT			GL10_4AD AD CH1
Device Diagnosis	😟 - 🍬		GL10_4AD AD CH2	%IW5	INT			GL10_4AD AD CH2
EtherCAT I/O Managina	😟 🧤		GL10_4AD AD CH3	%IW6	INT			GL10_4AD AD CH3
EtherCAT 1/0 Mapping	_							
EtherCAT IEC Objects								
Shahua								
Status								
Information								

Рис. 4.106. Добавление переменной

Во всплывающем окне выберите ранее созданную переменную InValueVel (рис. 4.107, 4.108).

Input Assistant	h de concerco de la concerción		~
Input Assistant			^
Text Search Categories			
Variables	Name	Туре	Address ^
	🗏 🔘 Application	Application	
	PLC_PRG	PROGRAM	
	🖤 🖗 BItStart	BOOL	
	🖤 🕸 buffer	LREAL	
	Connection1	WORD	
	Connection2	WORD	
	🛛 🖗 InValueVel	INT	
	Invelocity	BOOL	
	• WC_Halt_0	MC_Halt	
	MC_Halt_1	MC_Halt	
	# WC_Home_0	MC_Home	
	MC_Home_1	MC_Home	
	MC_MoveVelocity_0	MC_MoveVeloci	
	MC_Plovevelocity_1	MC_Moveveloa	ty
	Pric_Power_0	MC_Power	Y
	<		>
Structured view		Filter	None 🗸
Documentation		Insert with arguments	Insert with namespace prefix
InValueVel: INT(VAR)			~
Прием значения с аналогово вход	a		
			~
			OK Cancel

Рис. 4.107. Выбор созданной переменной

/ariable	Mapping	Channel	Address	Туре	Default Value	Unit	Description
₽*		GL10_4DA DA CH0	%QW2	INT			GL10_4DA DA CH0
± - *		GL10_4DA DA CH1	%QW3	INT			GL10_4DA DA CH1
÷		GL10_4DA DA CH2	%QW4	INT			GL10_4DA DA CH2
🗄 - ^K ø		GL10_4DA DA CH3	%QW5	INT			GL10_4DA DA CH3
🚊 🏘		Device status	%IW2	UINT			Device status
🗄 - 🍫		GL10_4AD AD CH0	%IW3	INT			GL10_4AD AD CH0
🚊 🏘		GL10_4AD AD CH1	%IW4	INT			GL10_4AD AD CH1
🗄 🦄 Application.PLC_PRG.InValueVel	~⊘	GL10_4AD AD CH2	%IW5	INT			GL10_4AD AD CH2
😟 🦄		GL10_4AD AD CH3	%IW6	INT			GL10_4AD AD CH3

Рис. 4.108. Привязка выбранной переменной

Так как диапазон принимающих устройств с аналогового входа варьируется от 0 до 20 000 (рис. 4.109), необходимо преобразовать эти значения в скорость в диапазоне 0 до 100.

```
AD Conversion Mode: 4mA~20mA(0~20000) ~
```

Рис. 4.109. Выбор диапазона

Для этого напишите формулу после программы. По условию включения приводов по питанию будет рассчитываться формула по преобразованию значений с аналогового входа в скорость (рис. 4.110).

```
56 //Преобразование входного диапазона с модуля в диапазон скорости
57 ⊡ IF MC_Power_0.Status AND MC_Power_1.Status THEN
58 setVel:=((INT_TO_LREAL(InValueVel)-0)/(20000-0))*(100-0)+0;
59 END_IF
```

Рис. 4.110. Преобразование входного диапазона с модуля на диапазон скорости

Преобразование INT_TO_LREAL необходимо, потому что в блоках MC_MoveVelocity при задании скорости следует ввести значение типа Lreal, а на вводе аналогового входа типа INT.

Для изменения скорости следует выключать функциональный блок и включать его заново с новыми значениями.

Выполните следующие действия:

Создайте переменную-буфер и сравните значения текущей скорости с заданной.

Для выключения и включения блока используйте входную переменную блока Execute.

После выполненных действий приравняйте значения буфера к заданному значению, чтобы программа до следующего изменения положения не заходила в код (рис. 4.111).

Добавьте условие: если блок останова не работает, то отключить его.

Проконтролируйте систему переменных, которые в дальнейшем будут передаваться в панель.

После выполненных операций программа должна выглядеть как на рис. 4.112.

```
60 //Сравнение буфера текущей с заданной скоростью
61 - IF buffer setVel THEN
62
        MC MoveVelocity 0(Axis:= Drivel, Execute:= FALSE, Velocity:= setVel, Acceleration:=50 , Deceleration:=50 , Jerk:= 100, Direction:=positive);
        MC MoveVelocity 1 (Axis:= Drive2, Execute:= FALSE, Velocity:= setVel, Acceleration:=50 , Deceleration:=50 , Jerk:= 100, Direction:=positive);
63
64
        MC MoveVelocity 0 (Axis:= Drivel, Execute:= TRUE, Velocity:= setVel, Acceleration:=50 , Deceleration:=50 , Jerk:= 100, Direction:= positive);
65
        MC MoveVelocity 1 (Axis:= Drive2, Execute:= TRUE, Velocity:= setVel, Acceleration:=50 , Deceleration:=50 , Jerk:= 100, Direction:= positive);
66
        buffer:=setVel;
67 END IF
68 //Отключения блока останова, если он не используется
69 [] IF step<>5 AND step<>3 THEN
70
        MC_Halt_0(Axis:=Drivel, Execute:=FALSE);
71
        MC Halt 1(Axis:=Drive2, Execute:=FALSE);
72 END IF
74 //Переменные для контроля системы и управления
75 Speedl:=Drivel.fActVelocity;
76 Speed2:=drive2.fActVelocity;
77 Connection1:=Drive1.wCommunicationState;
78 Connection2:=Drive2.wCommunicationState;
79 Power:=MC Power 0.Status AND MC Power 1.Status;
80 Invelocity:=MC MoveVelocity 0.InVelocity AND MC MoveVelocity 1.InVelocity;
81 Alarm:=Drivel.nAxisState=SMC AXIS STATE.errorstop OR Drive2.nAxisState=SMC AXIS STATE.errorstop;
82
```

Рис. 4.111. Программирование буфера сравнения, отключения блока останова и переменных для контроля системы

и управления

```
1 //Кнопка разрешения работы
<sup>2</sup> - IF BItStart THEN//Если кнопка нажата то выполняется кейс
3
4 占
        CASE step OF
5
    0: //Инициализация приводов
 6 占
        IF Drivel.wCommunicationState=100 AND Drive2.wCommunicationState=100 THEN
            step:=1;
8
        END IF
10 1: //Включение питания
11
        MC_Power_0(Axis:= Drivel, Enable:= TRUE, bRegulatorOn:=TRUE ,
                                                                           bDriveStart:= TRUE, );
12
        MC_Power_1(Axis:= Drive2, Enable:= TRUE, bRegulatorOn:=TRUE,
                                                                          bDriveStart:= TRUE, );
13
14 2: //Управление по скорости
15
        MC MoveVelocity 0(Axis:= Drivel, Execute:= TRUE, Velocity:= setVel, Acceleration:=50 , Deceleration:=50 , Jerk:= 50, Direction:=positive);
16
        MC MoveVelocity 1(Axis:= Drive2, Execute:= TRUE, Velocity:= setVel, Acceleration:=50, Deceleration:=50, Jerk:= 50, Direction:=positive);
17
18
19 3: //Выход в ноль
20
        MC_Halt_0(Axis:= Drivel,Execute:= TRUE,Deceleration:=50,Jerk:= 10);
21
        MC Halt 1(Axis:= Drive2,Execute:= TRUE,Deceleration:=50,Jerk:= 10);
22
        IF MC_Halt_0.Done THEN
23
           MC Home 0 (Axis:= Drivel,Execute:= TRUE,Position:= 0);
24
           MC_Home_1(Axis:= Drive2,Execute:= TRUE,Position:= 0);
25 p
           IF MC_Home_0.Done AND MC_Home_1.Done THEN
26
                MC Home 0 (Axis:= Drivel, Execute:= FALSE);
27
               MC_Home_1(Axis:= Drive2,Execute:= FALSE);
28
                step:=l;
29
            END IF
30
31
        END IF
32
33
    4: //Перезагрузка приводов
34
        MC Reset 0 (Axis:= Drivel, Execute:=TRUE);
35
        MC_Reset_1(Axis:= Drive2, Execute:=TRUE);
36 占
        IF MC_Reset_0.Done OR MC_Reset_1.Done OR MC_Reset_0.Error OR MC_Reset_1.Error THEN
37
            MC_Reset_0(Axis:= Drivel, Execute:=FALSE);
38
           MC_Reset_1(Axis:= Drive2, Execute:=FALSE);
39
           step:=1;
40
        END IF
41
42
    5: //Корректный останов
43
        MC_Halt_0(Axis:= Drivel,Execute:= TRUE,Deceleration:=50,Jerk:= 10);
44
        MC_Halt_1(Axis:= Drive2,Execute:= TRUE,Deceleration:=50,Jerk:= 10);
45
        IF MC Halt 0.Done AND MC Halt 1.Done THEN
46
            step:=1;
47
        END IF
48
49
    6: //Выключение питания
50
        MC_Power_0(Axis:= Drivel, Enable:= TRUE, bRegulatorOn:=FALSE , bDriveStart:= FALSE);
51
        MC Power 1 (Axis:= Drive2, Enable:= TRUE, bRegulatorOn:=FALSE, bDriveStart:= FALSE);
52
53
   END CASE
```

Рис. 4.112. Полная программа

```
55 END IF
56 //Преобразование входного диапазона с модуля в диапазон скорости
57 - IF MC Power 0.Status AND MC Power 1.Status THEN
58
        setVel:=((INT TO LREAL(InValueVel)-0)/(20000-0))*(100-0)+0;
59 END IF
60 //Сравнение буфера текущей с заданной скоростью
61 [] IF buffer<>setVel THEN
62
        MC MoveVelocity 0(Axis:= Drivel, Execute:= FALSE, Velocity:= setVel, Acceleration:=50 , Deceleration:=50 , Jerk:= 100, Direction:=positive);
63
        MC MoveVelocity 1 (Axis:= Drive2, Execute:= FALSE, Velocity:= setVel, Acceleration:=50 , Deceleration:=50 , Jerk:= 100, Direction:=positive);
64
        MC MoveVelocity 0(Axis:= Drivel, Execute:= TRUE, Velocity:= setVel, Acceleration:=50, Deceleration:=50, Jerk:= 100, Direction:= positive);
65
        MC MoveVelocity 1 (Axis:= Drive2, Execute:= TRUE, Velocity:= setVel, Acceleration:=50, Deceleration:=50, Jerk:= 100, Direction:= positive);
66
        buffer:=setVel;
67 END IF
68 //Отключения блока останова, если он не используется
69 - IF step<>5 AND step<>3 THEN
70
        MC Halt 0 (Axis:=Drivel, Execute:=FALSE);
71
        MC Halt 1 (Axis:=Drive2, Execute:=FALSE);
72 END IF
73
74 //Переменные для контроля системы и управления
75 Speedl:=Drivel.fActVelocity;
76 Speed2:=drive2.fActVelocity;
77 Connection1:=Drive1.wCommunicationState;
78 Connection2:=Drive2.wCommunicationState;
79 Power:=MC_Power_0.Status AND MC_Power_1.Status;
B0 Invelocity:=MC_MoveVelocity_0.InVelocity AND MC_MoveVelocity_1.InVelocity;
81 Alarm:=Drivel.nAxisState=SMC AXIS STATE.errorstop OR Drive2.nAxisState=SMC AXIS STATE.errorstop;
```

Рис. 4.112. Окончание

Для передачи данных в панель необходимо создать Symbol Configuration во вкладке Application (рис. 4.113).



Puc. 4.113. Application – Symbol Configuration

Во всплывающем окне поставьте галочку «Поддерживать ОРС» (рис. 4.114).

Include o	omments in XML		
Support	OPC UA features		
Add libra (recomm	y placeholder in Dev ended, but may trigg	vice Application ger download)	
Client Side D	ata Layout		
Compati	ility Layout		
Optimize	d Layout		

Рис. 4.114. Выбор ОРС

Далее нажмите кнопку Build (рис. 4.115).

/ 🌆 Har	rdware Configuration	י (<u>ה</u>	PLC_PRG	🛯 🖋 Driv	e1 🐶	Drive2	GL10_4AD		GL10_RTU_ECTA	Symbol Configuration 🗙	
🛛 View 👻 Refresh 😱 Export 🛛 🛱 Settings 👻 Tools 👻											
PExecute "Build" command to be able to select variables (you need an error-free build).											
Changed symbol configuration will be transferred with the next download or online change											
Symbols	Access Rights	Maximal	Attribute	Туре	Members	Comment					

Рис. 4.115. Кнопка Build

ymbols		Access Rights	Maximal	Attribute	Туре	Members	Comment
	Constants						
- 🔳 📄	IoConfig_Globals						
🔳 📄	PLC_PRG						
🔽	🔷 Alarm	*	*		BOOL		Авария приводов
···· 🗸	BItStart	S	St.		BOOL		Команда на разрешения работы
···· 🗸	Connection1	Sto	St.		WORD		Переменная для отслеживания статуса загрузки первого привода
···· 🗸	Connection2	Sto	*		WORD		Переменная для отслеживания статуса загрузки второго привода
	InValueVel		St.		INT		Прием значения с аналогово входа
🗸	Invelocity	S	St.		BOOL		Переменная для отслеживания статуса выхода на скорость
🔳	MC_Halt_0		*		MC_Halt		Блок останова 1 привода
	MC_Halt_1		St.		MC_Halt		Блок останова 2 привода
	MC_Home_0		St.		MC_Home		Блок выхода в ноль 1 привода
	MC_Home_1		*		MC_Home		Блок выхода в ноль 2 привода
	MC_MoveVelocity_0		St.		MC_MoveVelocity		Блок запуска 1 двигателя по скорости
····	MC_MoveVelocity_1		*		MC_MoveVelocity		Блок запуска 2 двигателя по скорости
	MC_Power_0		*		MC_Power		Блок питания 1 привода
	MC_Power_1		St.		MC_Power		Блок питания 2 привода
	MC_Reset_0		*		MC_Reset		Блок перезагрузки 1 привода
	MC_Reset_1		*		MC_Reset		Блок перезагрузки 2 привода
··· 🗸	Power	S	St.		BOOL		Переменная для отслеживания статуса включения питания
	Speed1	Sto	*		LREAL		Переменная отслеживания скорости 1 привода
	Speed2	5	*		LREAL		Переменная отслеживания скорости 2 привода
····	ø buffer		*		LREAL		Буфер для сравнения
	< setVel		*		LREAL		Задание скорости
V	< step	N	N		INT		War

Выберите переменные для передачи (рис. 4.116).

Рис. 4.116. Выбор переменных для передачи

После выбора обязательно нажмите на кнопку Build/Refresh (рис. 4.117) для того, чтобы контроллер запомнил все переменные для передачи.

	Hardware Configuration	PLC_PRG	B Drive1	। स्त्र Driv	re2	GL 10_4A	D O	GL10_RTU_ECTA	Symbol Configuration	×
i 🛛 Vier	🛚 👻 Refresh 🍙 Export	Settings 👻 Tool	3 🕶							
Changed	l symbol configuration will be tr	ansferred with the n	ext downloa	d or online cha	nge					
Symbo	ls Refresh	Access Rights	Maximal	Attribute	Туре		Members	Comment		
	Constants									
🗎 🖻 · 🕅 🚺	IoConfig_Globals									
--	PLC_PRG									

Puc. 4.117. Build/Refresh

Загрузите проект в контроллер и запустите программу.

Откройте среду для панелей InoTouchPad и создайте проект для IT7070E.

Зайдите во вкладку Connection, выберите EtherNET и добавьте новое соединение OPC UA Tags (рис. 4.118).

Project	× 🖾 00001:Screen 1	S Connections x				
Screens(1/512)						
Add Screen		+ Name - Numt	er • Communication protocol	Default State	Display mode Co	mment Enabled
00001:Screen 1	0.000	1 Connection 1 1	OPCUA Tags Access	Online		V
Embed Screens	COM1		, , , , , , , , , , , , , , , , , , ,			
Popup Screens	-	71				
Templates(0/64)						
Communication	COM2					
Connections						
- 🛱 Cycles	1.000					
	COM3	/				
e 🕞 Data Service						
Alarm Management	111101					
Recipes(0/100)	Ethernet					
Historical Data						
Scripts(0/400)						
- 📑 Reports(0/100)						
🗄 📄 Status Lists						
Runtime User Administration	General		Verification		Read mode	
Resource	URL: opc.tcp:// 192.1	168 . 1 . 100 Port: 4840 💈	 Anonymous 		 Initiative 	
🖶 🔅 HMI Settings	Session Name: IT7000	_OPCUA	Use Account		O Monitor	
	Connection Encryption:	None -	User name: User		Monitor time: 100	10 ms 💲

Рис. 4.118. Выбор нового соединения

Введите IP контроллера (рис. 4.119) и нажмите Browser Tags. Важно: при нажатии на Browser Tags программа в контроллере должна работать.

General	
URL: opc.tcp:// 192.168.1.88 Port: 4840 :	÷.
Session Name: IT7000_OPCUA	
Connection Encryption: None	Ŧ
Data Encryption: None	-
Timeout: 1000 ms	* *
Interval: 1000 ms	* *
Browser Tags discover	er

Рис. 4.119. Ввод ІР

Раскройте список и отметьте переменные. Затем нажмите Add tags, поставьте галочку Create PLC tags to group и нажмите OK (рис. 4.120).

IP Browser OPC UA tags ? \times Browse Import Address space: **Display Name** Node ID Data Type opc.tcp://192.168.1.88:4840 - Root Uiews Objects DeviceSet 🚊 🔲 Inovance-ARM-Linux E Resources Application Add Tag Delete Tag Programs Remove All E PLC_PRG Alarm[Boolean] BltStart[Boolean] Connection1[UInt16] Connection2[UInt16] ✓ Invelocity[Boolean] Power[Boolean] Speed1[Double] Speed2[Double] ✓ step[Int16] Zasks GlobalVars 🗌 🥏 Server Create PLC Tags to group: Tag group_2 -ΟК Refresh

Рис. 4.120. Выбор переменных

Переменные добавились в группу тегов, которая находится во вкладке Communication – Tags (рис. 4.121). Название тега можно изменять.

Project	di X	C 🔚	0000	1:Screen_1 ×	S Conne	ctions 🛛 🛪 🖘 Tag	group_2 ×										
Screens(1/512)																	
- 🖅 Add Screen			+.	Name •	 Number 	Connection Id	Data type	Length	Array count	Address •	Acquisition cycle Id	Acquisition m	Data log id	Logging cycle li	d_ogging acqui	Start value	Index Tag
- 🖾 00001:Screen_1			1	Alarm@ Root	1	Connection_1	Bool	1	1	ns=4;s= var Inova	1s	Cyclic on use	<undefined></undefined>	1s	Cyclic contin		<undefined></undefined>
Embed Screens			2	BitStart@ Ro	2	Connection_1	Bool	1	1	ns=4;s= var Inova	1s	Cyclic on use	<undefined></undefined>	1s	Cyclic contin		<undefined></undefined>
E Popup Screens			3	Connection1	3	Connection_1	Uint16	2	1	ns=4;s= var Inova	1s	Cyclic on use	<undefined></undefined>	1s	Cyclic contin		<undefined></undefined>
🖭 🔝 Templates(0/64)			4	Connection2	4	Connection_1	Ulnt16	2	1	ns=4;s= var Inova	1s	Cyclic on use	<undefined></undefined>	1s	Cyclic contin		<undefined></undefined>
Communication			5	Invelocity@IR	5	Connection_1	Bool	1	1	ns=4;s= var Inova	1s	Cyclic on use	<undefined></undefined>	1s	Cyclic contin		<undefined></undefined>
- P Connections			6	Power@Root	6	Connection_1	Bool	1	1	ns=4;s= var Inova	1s	Cyclic on use	<undefined></undefined>	1s	Cyclic contin		<undefined></undefined>
- 📫 Cycles			7	Speed1@ Ro	7	Connection_1	Double	8	1	ns=4;s= var Inova	1s	Cyclic on use	<undefined></undefined>	1s	Cyclic contin		<undefined></undefined>
⊟ Tags(3/128)			8	Speed2@IRo	8	Connection 1	Double	8	1	ns=4:s=lvarilnova	1s	Cyclic on use	<undefined></undefined>	1s	Cyclic contin		<undefined></undefined>
Label lags			9	step@IRoot/	9	Connection 1	Int16	2	1	ns=4:s=lvarlinova	1s	Cyclic on use	<undefined></undefined>	1s	Cyclic contin		<undefined></undefined>
See Show All Tags				10								-4					
Add Tag Group																	
The group 2																	

Рис. 4.121. Группа тегов

Перейдите на главный экран. Его фон можно сделать любого цвета.

Для редактирования управления и контроля приводов (рис. 4.122) понадобятся следующие инструменты:

Text Field – запись текста в поле;

Bit indicator и Bit button – индикатор и кнопка соответственно; Number IO field – поле вывода значений.

Tools	æ,
L> L= ■ ■	
Simple Controls	
Polygon	
 Ellipse 	
Rectangle	
🏹 Bezier	
Table	
A Text Field	
Bit Indicator	ב
时 Bit Button	
E Word Indicator	
¹²³ Word Button	
Simple Graphics View	
Graphics View	
01 Number IO Field	
al String IO Field	
Oate-Time Field	
• Graphic IO Field	
Symbolic IO Field	
Button	
I Text Switch	
Graphic Switch	
Timer	
Gif Display View	

Рис. 4.122. Инструменты редактирования управления и контроля приводов Конечный интерфейс проекта должен соответствовать изображенному на рис. 4.123.

3arp	узка 1 прин	вода	Запуск	Загрузка 2 при	вода
	0000			0000	
Скор	ость 1 при	вода	Шаг программы	Скорость 2 при	вода
	0000	· · · · ·	00	0000	
	Питание		Питание приводов	Питание	
	На скорости	· · · · · ·	Управление по скорости	и	1
			Выход в ноль		
			Перезагрузка приводов		
Авария			Корректный останов		Авария
	· · · · · · · · · · · · · · · · · · ·	· · · · · · ·	Отключение питания		

Рис. 4.123. Визуальное оформление программы

Кнопка «Запуск» имеет следующие настройки:

Выбор тега: Read tag – тег, который был передан с контроллера, выберите его и задайте Mode в положении Set (рис. 4.124).

General ⊕ Properties ⊕ Animations ⊕ Events	Read Read Tag BltStart - Output Reverse	Write ✓ Read/Write Tag Same Mode Set ▼
	press ✓ Click animate ☐ Auto Repeat ✓ Filter Repeat Msg Hold Delay 0 *100ms ≑	

Рис. 4.124. Выбор тега и значения

Выставьте фон и подпись в строке Status: 0 – выключено, 1 – включено (рис. 4.125).

General Proportion											
Status		Status	Graphic	Text	Font	Position	TextColor	BgColor	Flashing	Frequency	Preview
StyleSheet Misc	1	0		Запуск	Droid Sans I	Center	#000000	#fff00	No	1000 ms/time	Запус
Security Operation Record Animations	2	1		Запуск	Droid Sans I	Center	#000000	#00ff00	No	1000 ms/time	Запус

Рис. 4.125. Фон и статус

Далее программируются поля вывода шага программы, скорости привода и его загрузки.

В строке mode выставьте «Отображение» и размер символов – 4. Также измените текст внутри рамки на жирный в Properties – text (рис. 4.126).

 General 	Type		Format		
Animations	Type		Tornac		
	Mode Output	•	Format type	Dec	+
	Icd display		Shift decimal point	0	*
			String field length	4	\$
	Process			Leading zero	
	ReadTag Connect	ion1 🔹			

Рис. 4.126. Оформление отображения

Для каждого поля выставьте свой тег в Read tag:

1) для загрузки – Connection;

2) скорости – Speed;

3) шага – Step.

Далее сконфигурируйте кнопки управления (питание, управление, выход в ноль и т. д.).

В каждой кнопке необходимо выставить Mode – Momentary ON (рис. 4.127).

Bit Button_2(BitButton)		
General ⊕ Properties ⊕ Animations	Read	Write
Events	Read Tag	✓ Read/Write Tag Same
	Output Reverse	Mode MomentaryON -
	press-	
	Click animate	
	Auto Repeat	
	Filter Repeat Msg	
	Hold Delay 0 *100ms	

Рис. 4.127. Установка значения ОЛ

Далее используйте для работы вкладку Animation. Выставьте тег загрузки и примите значение, при котором кнопка будет активна только когда загрузка привода достигнет 100 % (рис. 4.128).

Bit Button_2(BitButton)									
General	Enabled								
Properties	-Tag - Object state								
Animations	Tag Object state								
Enable Object	Connection1 Disabled Enabled								
Visibility									
Events	Туре								
	Interger Range from 100 * to 100 *								
	○ Bit 0								

Рис. 4.128. Выставление тега для анимации

Во вкладке Events выберите Click и SetValue.

Для каждой программируемой кнопки прикрепите тег Step, значения в каждой кнопке будут разные. Данные значения символизируют шаги в программе.

Каждая запрограммированная кнопка имеет определенное значение, которое передается в ПЛК при нажатии на кнопку:

1) питание приводов – «1»;

2) управление по скорости – «2»;

3) выход в ноль – «3»;

4) перезагрузка приводов – «4»;

5) корректный останов – «5»;

6) отключение питания – «6».

Далее программируют индикатор отображения состояния (авария, питание, на скорости) (рис. 4.129).

Bit Indicator_1(BitIndicator)	
General Properties Status Appearance Layout Misc Security H Animations	Read Read Tag Power - Output Reverse

Рис. 4.129. Индикатор отображения состояния

Выставьте теги с ПЛК и в статусе отобразите цвет фона, который может изменяться в зависимости от статуса/состояния (рис. 4.130).

Bit Indicator_1(BitIndicator)											
General											
Properties											
Status		Status	Graphic	Text	Font	Position	TextColor	BgColor	Flashing	Frequency	Preview
Appearance											
Layout					_		_	_			
Misc	1	0		Питание	Droid Sans [Center	#000000	#ff0000	No	1000 ms/time	Питание
Security											
Animations											
	2	1		Питание	Droid Sans I	Center	#000000	#00ff00	No	1000 ms/time	Питание

Рис. 4.130. Отображение питания

При работающем ПЛК загрузите проект в панель через инструменты или через вкладку Compiler.

Задайте IP панели и загрузите (рис. 4.131).

Transfer-Down	nload				?	Х
connect						
Network	•	192	168 .	1	. 100	
password:						
			0%			
✓ sync date ti	me 🗌	boot logo 🗌 clea	ar logs 🗌 clear rw 🗌	retain recipe 🗌 reta	ain user data	
✓ sync time z	one 🗹	retain InstalMent	✓ close dialog when	execute successfully	у	
					Download (Cancel

Рис. 4.131. Присвоение ІР панели

После загрузки соединение с ПЛК будет автоматическим.

Содержание работы

- 1. Цель работы.
- 2. Задания.
- 3. Ход работы.
- 4. Выводы.
- 5. Контрольные вопросы.

Средства, используемые при выполнении лабораторной работы

1. Методические указания к выполнению лабораторных работ.

2. Данные, предоставленные преподавателем во время занятия.

3. При проведении анализа допускается использование глобальной сети Интернет.

Контрольные вопросы

- 1. Что делает функциональный блок MC_Reset?
- 2. Какой сигнал имеет потенциометр на стенде?
- 3. Для чего нужен Symbol Configuration?
- 4. Зачем в проекте была использована команда Enable Object?
- 5. Что делает функция SetValue?

ЗАКЛЮЧЕНИЕ

Компьютерная система управления представляет собой микроэлектронно-вычислительную машину, собирающую информацию от датчиков, которые, в свою очередь, отражают состояние объекта или процесса через аналого-цифровые преобразователи или отражают информацию на пульте управления и сохраняют ее на запоминающем устройстве.

Применение компьютерных систем управления в технических системах – разновидность современных информационных технологий, основанных на использовании средств вычислительной техники и направленных на решение задач управления техническими объектами.

Компьютерные системы управления позволяют добиться повышения производительности технологических процессов на предприятиях, а также повысить качество выпускаемой продукции.

Переход на компьютерные системы управления знаменует начало нового этапа в автоматизации. Наличие данных систем на производстве дает возможность отказаться от дорогостоящего импортного оборудования и одновременно с этим повысить собственную конкурентоспособность на мировом рынке.

130

РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Анашкин, А. С. Техническое и программное обеспечение распределенных систем управления / А. С. Анашкин, Э. Д. Кадыров, В. Г. Харазов ; под ред. В. Г. Харазова. – СПб. : П-2, 2004. – 368 с. – ISBN 5-93893-274-2.

2. *Афонин, В. Л.* Интеллектуальные робототехнические системы / В. Л. Афонин, В. А. Макушкин. – М. : Интернет-университет информационных технологий – ИНТУИТ.ру, 2005. – 208 с. – ISBN 5-9556-0024-8.

3. Денисов, М. С. Системы числового программного управления : лаб. практикум / М. С. Денисов ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2021. – 112 с. – ISBN 978-5-9984-1412-1.

4. *Конюх, В. Л.* Компьютерная автоматизация производства : в 2 ч / В. Л. Конюх. – Кемерово : КузГТУ, 2003. – 118 с. – ISBN 5-89070-271-8 (ч. 1). – ISBN 5-89070-272-6 (ч. 2).

5. *Ловыгин, А. А.* Современный станок с ЧПУ и САD/САМ система / А. А. Ловыгин, А. В. Васильев, С. Ю. Кривцов. – М. : Эльф ИПР, 2006. – 286 с. – ISBN 5-900891-60-7.

6. *Рассказчиков, Н. Г.* Компьютерные системы управления : учеб. пособие / Н. Г. Рассказчиков ; Владим. гос. ун-т. – Владимир : Изд-во Владим. гос. ун-та, 2010. – 155 с. – ISBN 978-5-9984-0017-9.

7. *Селевцов, Л. И.* Автоматизация технологических процессов : учебник / Л. И. Селевцов. – М. : Academia, 2019. – 160 с. – ISBN 978-5-4468-0615-7.

8. Шалыгин, М. Г. Автоматизация измерений, контроля и испытаний : учеб. пособие / М. Г. Шалыгин, Я. А. Вавилин. – СПб. : Лань, 2019. – 172 с. – ISBN 978-5-507-47370-0.

Учебное электронное издание

ДЕНИСОВ Максим Сергеевич РУМЯНЦЕВ Илья Валерьевич ЧЕБОТАРЕВ Петр Александрович

КОМПЬЮТЕРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

Лабораторный практикум

Редактор Е. А. Лебедева Технический редактор Ш. Ш. Амирсейидов Компьютерная верстка Л. В. Макаровой Корректор О. В. Балашова Выпускающий редактор А. А. Амирсейидова

Системные требования: Intel от 1,3 ГГц; Windows XP/7/8/10; Adobe Reader; дисковод CD-ROM.

Тираж 9 экз.

Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых Изд-во ВлГУ rio.vlgu@yandex.ru

Институт машиностроения и автомобильного транспорта кафедра автоматизации, мехатроники и робототехники chebotarev@vlsu.ru