

## Краткое описание уроков видеокурса «Разработка АСУ ТП в Codesys»

<http://elektrik.info/codesys.html>

- 1) Подготовительная работа. Устанавливаем Codesys и целевую платформу контроллера.
- 2) Создаем проект и реализуем в нём первую программу на языке LD для управления приготовлением концентрата кислоты.
- 3-4) Допишем программу из второго урока. Создаем в проекте визуализацию.
- 5) Знакомство с языком FBD. Повторяем на нём реализацию кода управления приготовлением концентрата кислоты.
- 6) Знакомство с языком SFC. Повторяем на нём реализацию кода управления приготовлением концентрата кислоты.
- 7) Знакомство с языком ST. Повторяем на нём реализацию кода управления приготовлением концентрата кислоты.
- 8) Рассматриваем типы данных представленных в Codesys. Работаем с арифметическими операторами.
- 9) Знакомство с временным типом Time. На примерах рассматриваем работу таймеров TP, TON и TOF.
- 10) Рассматриваем способы оптимизации кода с применением реализаций в проекте подпрограмм, функциональных блоков и функций. Подключаем к проекту готовые библиотеки и создаем собственные.
- 11) Знакомимся со следующим техническим заданием. Создаем функциональный блок Reaktor на языке SFC и одновременно с этим разбираем основные элементы этого языка.
- 12) Продолжаем реализовывать функциональный блок Reaktor. По мере реализации этого кода проверяем его работу на примере вызова его экземпляров в главной программе.
- 13) Продолжаем реализовывать функциональный блок Reaktor. В нём задействуем таймер TP для управления закачкой концентрата кислоты в реактор.
- 14) Продолжаем работать над функциональным блоком Reaktor в частности над реализацией кода управления закачкой концентрата кислоты.
- 15) Завершаем в реализации функционального блока Reaktor код управления закачкой концентрата кислоты. Тестируем его работу на экземплярах этого функционального блока. 16) Реализуем в проекте функцию для расчёта времени дозирования содержимого реакторов концентратом кислоты.
- 17) Реализуем функцию, позволяющую нам определять в режиме реального времени объём концентрата кислоты, поступившего в реактор.
- 18) Завершаем реализацию кода функционального блока Reaktor.

- 19) Работаем с кодом основной программы. Связываем в ней экземпляры функционального блока Reaktor с вызовом программы управления приготовлением концентрата кислоты. Для входов и выходов каждого вызову отводим переменную в главной программе. Тестируем полученный результат.
- 20) Работаем над визуализацией. Отражаем на ней ход выполнения работы технологического процесса, реализованного нами в проекте, а также добавляем в неё управленческие элементы.
- 21) Создаем программу управляющую процессом нейтрализации продукта реакторов. Реализовываем её на языке стандартного SFC.
- 22) Продолжаем работать над программой Neitralization. Разбираем отличия МЭК-шага от упрощенного шага SFC.
- 23-24) Завершаем реализацию программы Neitralization. Тестируем её работу.
- 25-27) Рассматриваем работу блока PID из библиотеки util. Реализуем на его основе терморегулирование содержимого емкости нейтрализации.
- 28) Работаем с вызовом программы Neitralization в теле главной программы проекта.
- 29-30) Работаем над визуализацией проекта, добавляем в неё новый функционал.
- 31) Работа с элементом «Тренд». Настраиваем на визуализации графическое отображение хода изменение температуры содержимого емкости нейтрализации.
- 32) Знакомимся с конфигурационной структурой ПЛК. Рассматриваем виды программных модулей аппаратной периферии контроллера.
- 33-34) Связываем входы-выходы ПЛК с переменными проекта.
- 35) Добавляем в проект каналы связи с контроллером. Загружаем программу в ПЛК.
- 36) Рассматриваем способ расширения аппаратной периферии ПЛК за счет подведения к нему модулей ввода/вывода. Знакомимся с модулями МВА8 и МВУ8. Разбираем принцип работы протокола Modbus.
- 37) Занимаемся конфигурированием модуля МВА8. Настраиваем его на подведение к нему датчика термосопротивления и задаем модулю сетевые настройки для возможности считывания значений его входов по протоколу Modbus.
- 38) Конфигурируем модуль МВУ8. Настраиваем его на управление его выходами по протоколу Modbus.
- 39) Связываем ПЛК с модулем МВА8 и передаем в программу проекта измеренную температуру датчиком термосопротивления.
- 40) Настраиваем ПЛК на управление выходами модуля МВУ8.
- 41) Разбираем предназначения HMI-интерфейса (графической панели оператора). Обзор характеристик панели СП270.

42) Создаем проект для панели СП270 в конфигураторе СП200. Задаем сетевые параметры панели СП270 для подключения ее к шине Modbus.

43-44) Продолжаем дорабатывать проект для панели СП270. Добавляем в этот проект визуализационные элементы для отображения на экране панели хода работы технологического процесса, которым управляет программа ПЛК.

45) Настраиваем проект контроллера на взаимодействие с панелью СП270. 46) Проверяем работу взаимодействия проекта панели СП270 с проектом ПЛК.

47) Выводим на экране панели измеренную температуру датчиком, который подведён к модулю МВА8.

48) Передаём из программы ПЛК в проект панели числа с плавающей точкой.

49) Добавляем в HMI-проект новый функционал.

50) Убираем недочёты в проекте панели оператора.

51) Добавляем в проект панели оператора поля для ввода и отображения временного значения.

52) Производим одновременную запись сразу всех полей временного интервала из панели оператора в программу ПЛК.

53) Добавляем в проекты HMI и ПЛК недостающие визуализационные элементы.

54) Дорабатываем проект контроллера и проект панели до логического конца, удовлетворяющего всем требованиям технического задания нашего АСУ.

55) Завершающий этап. Загружаем проект в память панели С270 и производим загрузку последней версии нашей программы в ПЛК.

56) Заключение. Разбираем схемы подключения (щита управления, датчиков, исполнительных механизмов).

Ссылка на курс:

<http://electrik.info/codesys.html>