

О среде программирования для программируемых логических контроллеров

Н.Б. Лазарева, Д.О. Сычёв

Тихоокеанский государственный университет, Хабаровск

Аннотация: В статье рассмотрены возможности отечественной программной среды разработки CoDeSys на примере программы управления регулировки автомобильного трафика предприятия внутри собственной территории.

Ключевые слова: программа, контроллер, переменная, код, функциональный блок, автоматизированные системы.

Современные промышленные предприятия, в целях повышения качества продукции и снижения расходов на сырье, все активнее автоматизируют те или иные технологические и производственные процессы. Как правило, хорошо себя зарекомендовали те процессы, автоматизация которых была основана на программируемых логических контроллерах (ПЛК). Но для программирования ПЛК пользуются специализированным программным обеспечением (ПО). Такие факторы, как невозможность или сложность покупки или продления лицензии, а также риски, связанные с информационной безопасностью, делают сложным, а порой и невозможным использование того или иного ПО в работе промышленных предприятий. Однако на рынке программных продуктов появились альтернативные решения, разработанные в нашей стране [1].

Принцип работы ПЛК прост. На вход контроллера приходят показания параметров в форме дискретного или аналогового сигнала, измеренные датчиками или сигнализаторами. Эти сигналы поступают в центральный процессор контроллера для дальнейшей обработки. В случае, если процессор обнаружит отклонение регулируемого параметра, то он, основываясь на загруженную в его оперативную память программу, сформирует сигнал рассогласования и через свои выходы отправит этот сигнал на исполнительный механизм. В соответствии с полученным сигналом

рассогласования исполнительный механизм изменит положение регулирующего органа, тем самым выровняет значение регулируемого параметра.

В данной статье рассмотрим среду программирования «CoDeSys» [2] от компании «ОВЕН», как один из примеров доступного и функционального программного обеспечения для промышленного применения.

В CoDeSys входят языки программирования стандарта IEC 61131-3 [3] и язык CFC, который расширяет возможности программирования. Интерфейс CoDeSys основан на знакомых принципах, используемых в популярных инструментах разработки, таких как Visual C++, что делает его доступным для разработчиков с различным опытом. Так же в CoDeSys входят: редактор визуализации для отображения состояния производственных процессов в реальном времени; средства отладки для эффективного поиска и устранения ошибок в коде; конфигураторы протоколов обмена для настройки связи с различными устройствами.

Рассмотрим пример использования CoDeSys для создания программы управления светофором с использованием контроллера ОВЕН ПЛК 110 [M02] [4].

Создаем проект с именем «PLC_PRG» и типом «Программа». Для программирования выберем язык CFC (Continuous Function Chart) [5]. Далее определим входной сигнал, чтобы запускать работу светофора. Создадим команды для каждого цвета светофора (красный, желтый, зеленый) и свяжем их с соответствующими выходами контроллера. Для дальнейшей работы создадим два функциональных блока и одну программу. Чтобы создать соответствие между стадиями работы светофора и цветами его ламп, создадим блок TRAFFICSIGNAL, используя язык Function Block Diagram (FBD). Как таймер будем использовать блок WAIT, язык Список Инструкции

(IL). И программу «SEQUENCE», используя язык Sequential Function Chart (SFC), в которой объединим созданные блоки [6].

Начнём с блока TRAFFICSIGNAL. Объявим входную переменную STATUS типа INT, которая будет определять стадию работы светофора. Определим выходные переменные RED, YELLOW и GREEN типа BOOL.

Раздел объявлений блока TRAFFICSIGNAL:

```
FUNCTION_BLOCK TRAFFICSIGNAL
VAR_INPUT
    STATUS:INT;
END_VAR
VAR_OUTPUT
    GREEN:BOOL;
    YELLOW:BOOL;
    RED:BOOL;
END_VAR
VAR
```

В разделе кода POU (body) составим логические цепи для обработки входных и выходных переменных (рис.1).

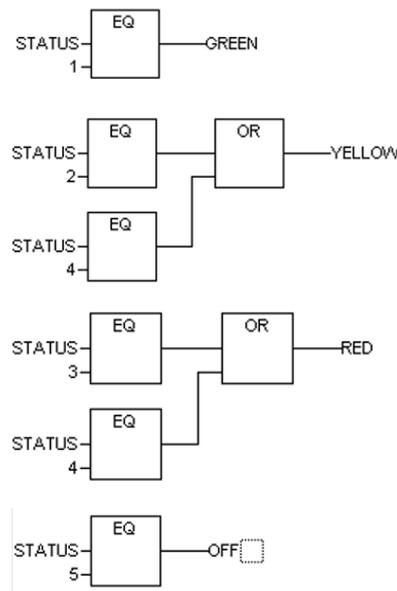


Рис. 1 – Блок TRAFFICSIGNAL

Функциональный блок TRAFFICSIGNAL готов.

Приступим к функциональному блоку WAIT [7]. Объявим входную переменную TIME типа TIME, чтобы управлять временем стадий светофора. Так же в разделе объявления блока создадим выходную переменную OK типа BOOL, которая будет принимать значения TRUE по истечению заданного

времени. В объявлении переменной ОК добавим инициализацию со значением FALSE: ОК: BOOL: = FALSE;

Подготовим переменную ОК для использования в таймере и определим функциональный блок TP [8] для отсчета времени. Таймер TP должен иметь два входа (IN, PT) для запуска и для заданного времени, и два выхода (Q, ET) для сигнала TRUE, сформированного после истечения времени, и для выходного значения оставшегося времени. Таким образом, таймер TP обеспечивает отсчет времени и изменение выходного сигнала Q в зависимости от заданного времени и состояния входного сигнала IN. Объявим в блоке WAIT локальную переменную ZAB типа TP: ZAB: TP;

Раздел объявлений WAIT:

```
FUNCTION_BLOCK WAIT
VAR_INPUT
    TIME_IN: TIME;
END_VAR
VAR_OUTPUT
    OK: BOOL: =FALSE;
END_VAR
VAR
    ZAB: TP;
END_VAR
```

Текст таймера WAIT:

```
LD    ZAB.Q
JMPS  mark

CAL   ZAB (IN: =FALSE)
LD    TIME_IN
ST    ZAB.PT
CAL   ZAB (IN: =TRUE)
JMP   end

mark:
CAL   ZAB

end:
LDN   ZAB.Q
ST    OK

RET
```

С блоком WAIT закончили. Теперь в программе «SEQUENCE» объединим получившиеся блоки. В программе «SEQUENCE» определим входную переменную с именем START типа BOOL, которая будет использоваться для запуска программы. Определим две переменные, назовём их TRAFFICSIGNAL1 и TRAFFICSIGNAL2 типа INT. Они нужны для

управления соответствующими светофорами. Определим переменную DELAY типа WAIT для управления временем задержки между стадиями светофора. Программа «SEQUENCE», раздел объявлений:

```
PROGRAM SEQUENCE
VAR_INPUT
  START: BOOL;
END_VAR
VAR_OUTPUT
  TRAFFICSIGNAL1: INT;
  TRAFFICSIGNAL2: INT;
END_VAR
VAR
  COUNTER: INT;
  DELAY: WAIT;
END_VAR
```

Создадим структуру SFC-диаграммы в программе «SEQUENCE», с помощью которой будем программировать этапы и переходы. Добавим этапы Switch1, Green2, Switch2, Green1 и Init, который не изменит своего имени (рис.2). Переход после Init назовём «START», оставшиеся назовём «DELAY.OK». Последний переход Init назовём «Switch1». «Switch» включает желтый сигнал светофора, в Green1 TRAFFICSIGNAL1 и в Green2 TRAFFICSIGNAL2 – зеленый.

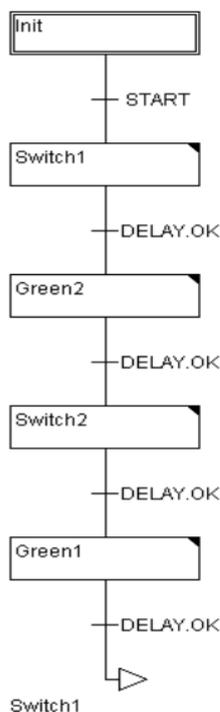


Рис. 2. – Программа SEQUENCE, раздел инструкций

Используя язык (IL), напишем код для управления состоянием светофоров и временем задержки на каждом этапе. Например, этап Green2:

```
LD 3
ST TRAFFICSIGNAL1
LD 1
ST TRAFFICSIGNAL2
CAL DELAY (TIME_IN:=t#5s)
```

Реализация ночного отключения светофоров. Для этого создадим счетчик, который будет отключать светофор после определённого количества циклов TRAFFICSIGNAL. Для этого в разделе объявлений SEQUENCE создадим переменную COUNTER типа INT. Добавляем этапы Count и Off для управления счетчиком и отключения светофоров (рис.3). Переходы назовём EXIT, TRUE и DELAY.OK.

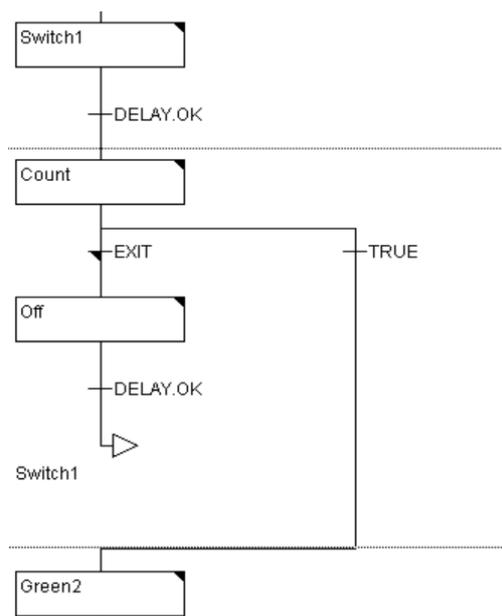


Рис. 3. – Программа SEQUENCE, раздел инструкций

Для каждого этапа в блоке SEQUENCE пишем код:

действие Count:

```
LD COUNTER
ADD 1
ST COUNTER
```

переход EXIT:

```
LD COUNTER
GT 7
```

действие Off:

```
LD 4
ST TRAFFICSIGNAL1
LD 2
ST TRAFFICSIGNAL2
LD 0
ST COUNTER
CAL DELAY (TIME_IN:=t#10s)
```

Наконец распределим входные и выходные переменные в блоке программы PLC_PRG. Но прежде в редакторе объявлений программы PLC_PRG объявим переменные с LIGHT1, LIGHT2 типа TRAFFICSIGNAL:

```
PROGRAM PLC_PRG
VAR
    LIGHT1: TRAFFICSIGNAL;
    LIGHT2: TRAFFICSIGNAL;
END_VAR
```

Объявим глобальные переменные для шести выходов (по 3 для каждого светофора) и одного входа:

```
VAR_GLOBAL
    IN: BOOL;
    L1_GREEN: BOOL;
    L1_YELLOW: BOOL;
    L1_RED: BOOL;
    L2_GREEN: BOOL;
    L2_YELLOW: BOOL;
    L2_RED: BOOL;
END_VAR
```

В редакторе Continuous Function Chart [9] выберем элемент Box и преобразуем его в «SEQUENCE». Добавим элементы TRAFFICSIGNAL и заменим их локальными переменными LIGHT1, LIGHT2 объявленные ранее. Создадим элементы глобальных переменных L1_green, L1_yellow, L1_red, L2_green, L2_yellow, L2_red типа Output и элемент IN типа Input. Соединим их входы и выходы вместе (рис.4).

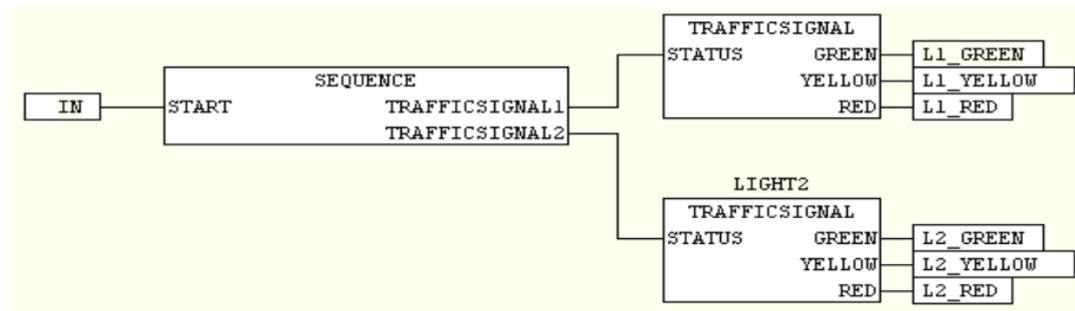


Рис. 4. – Программа PLC_PRG

Получившийся гипотетический светофор имеет специфичный, но при этом достаточно понятный режим работы [10]. Это ограничивает возможность применения данного светофора в городской среде, однако не мешает применению данного светофора на промышленных предприятиях [11]. Так, подобные светофоры используют предприятия, которые нуждаются в регулировки автомобильного трафика внутри собственной территории, так как для их работы часто нужна узкоспециализированная колёсная техника.

Применение контроллеров на современных промышленных предприятиях необходимо и многогранно. В большинстве производственных процессов автоматизированные системы позволяют свести участие человека к минимуму или вовсе исключить. Это позволяет снизить вероятность появления травм у работников, вероятность аварии на производстве и высвободить работников для других работ. С их помощью могут функционировать как достаточно простые системы, например, система вентиляции воздуха, так и более сложные системы. Также ПЛК позволяют реализовать все три уровня автоматизации, к которым прибегают все достаточно крупные предприятия.

Контроллер ОВЕН ПЛК 110[M02] имеет дискретные входы и выходы в своей конструкции, а также поддерживает CoDeSys. На нём основываются множество различных систем автоматизированного контроля и управления на промышленных предприятиях. Это делает его отличной платформой для реализации данной программы светофора и демонстрации возможностей CoDeSys. CoDeSys - это отличная программная среда с широким спектром возможностей, инструментами и функционалом предназначенная для удобной и комфортной работы. Он является простым и понятным инструментом для разработки программ и алгоритмов управления для поддержания сложных технологических процессов и производств.

Литература

1. Лазарева Н. Б. Оптимальный подход к разработке программного обеспечения с использованием современных методологий и технических средств // Инженерный вестник Дона, 2020, №10. URL: ivdon.ru/ru/magazine/archive/n10y2020/6625
2. Pratt G. The Book of CODESYS: The Ultimate Guide to PLC and Industrial Controls Programming with the CODESYS IDE and IEC 61131-3. – ControlSphere LLC, 2021. – P. 492.
3. Гофман П. М., Кузнецов П. А. Инструменты программирования промышленных контроллеров. – Красноярск: СибГУ им. академика М. Ф. Решетнёва, 2019. — 94 с.
4. Гуляев, А. В. Программирование логических контроллеров в среде CoDeSys для автоматизированных систем управления. – Хабаровск: ДВГУПС, 2022. — 64 с.
5. Collins K. PLC Programming for Industrial Automation. – Exposure, 2006. – P. 119.
6. Bryan E.A., Bryan L. A. Programmable Controllers Theory and Implementation. – Amer Technical Pub, 2003. – P. 1035.
7. Jochen Petry. IEC 61131-3 mit CoDeSys V3: ein Praxisbuch für SPS-Programmierer. – 3S-Smart Software-Solutions, 2011. – P. 839.
8. Петров И. В. Програмируемые контроллеры. Стандартные языки и приемы прикладного проектирования. – М., СОЛОН-Пресс, 2004. – 256 с.
9. John R. Hackworth, Frederick D. Hackworth. Programmable Logic Controllers: Programming Methods and Applications. – Pearson College Div, 2003. – P. 303.
10. Bolton W. Programmable Logic Controllers. – Newnes, 2011. – P. 304.
11. Шульженко С. Н., Шульженко Н. А., Сорвина О.В. Системный анализ внедрения инновационных нанопроизводств в структурах

низкорентабельных промышленных предприятий // Инженерный вестник Дона, 2024, №1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8971

References

1. Lazareva N.B. Inzhenernyj vestnik Dona, 2020, №10. URL: ivdon.ru/ru/magazine/archive/n10y2020/6625
2. Pratt G. The Book of CODESYS: The Ultimate Guide to PLC and Industrial Controls Programming with the CODESYS IDE and IEC 61131-3. ControlSphere LLC, 2021. P. 492.
3. Gofman P. M., Kuznecov P. A. Instrumenty programmirovaniya promyshlennykh kontrollerov [Industrial controller programming tools]. Krasnojarsk: SibGU im. akademika M. F. Reshetnjova, 2019. 94 p.
4. Guljaev, A. V. Programmirovaniye logicheskikh kontrollerov v srede CoDeSys dlja avtomatizirovannykh sistem upravleniya [Programming logic controllers in the CoDeSys environment for automated control systems]. Habarovsk: DVGUPS, 2022. 64 p.
5. Collins K. PLC Programming for Industrial Automation. Exposure, 2006. P. 119.
6. Bryan E. A., Bryan L. A. Programmable Controllers Theory and Implementation. Amer Technical Pub, 2003. P. 1035.
7. Jochen Petry. IEC 61131-3 mit CoDeSys V3: ein Praxisbuch für SPS-Programmierer. 3S-Smart Software-Solutions, 2011. P. 839.
8. Petrov I. V. Programmiruemye kontrollery. Standartnye jazyki i priemy prikladnogo proektirovaniya [Standard languages and techniques for applied design]. M., SOLON-Press, 2004. 256 p.
9. John R. Hackworth, Frederick D. Hackworth. Programmable Logic Controllers: Programming Methods and Applications. Pearson College Div, 2003. P. 303.



10. Bolton W. Programmable Logic Controllers. Newnes, 2011. P. 304.

11. Shul'zhenko S. N., Shul'zhenko N. A., Sorvina O.V. Inzhenernyj vestnik Dona, 2024, №1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8971

Дата поступления: 6.05.2024

Дата публикации: 22.06.2024