

06, июнь 2016

УДК 621.865

Разработка системы управления грузовым лифтом на основе ПЛК

*Чесноков И.К., студент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы автоматизации производства»*

*Научный руководитель: Ненашев А.О.
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы автоматизации производства»
bauman@bmstu.ru*

Введение

В настоящее время существует множество различных систем управления грузовыми лифтами. В основном данные системы управления основаны на релейной аппаратной части. Данный подход, помимо преимуществ, имеет ряд недостатков. Основным недостатком является отсутствие возможности изменения управляющей программы лифта без вмешательства в аппаратную часть. Это означает, что при возникновении необходимости изменения управляющей программы потребуется пересмотреть электрическую схему. Это сопряжено с финансовыми затратами на покупку новых элементов, а также с длительным временем перенастройки системы управления. В связи с этим, для управления лифтом предлагается использовать ПЛК (программируемый логический контроллер). Один ПЛК может управлять несколькими лифтами, что важно в случае расширения системы и внесения в нее новых устройств.

1. Описание системы

Грузовой лифт состоит из следующих элементов:

- Основной двигатель с возможностью работы в реверсивном режиме. Для него предусмотрены следующие сигналы:
 - Управление пускателями нормального и реверсивного режимов
 - ОС от автомата защиты

- ОС от пускателей
- Команда на запуск в нормальном режиме
- Команда на запуск в реверсивном режиме
- Двигатель верхнего удерживающего устройства с возможностью работы в реверсивном режиме. Для него предусмотрены сигналы, аналогичные сигналам основного двигателя.
- Двигатель нижнего удерживающего устройства с возможностью работы в реверсивном режиме. Для него предусмотрены сигналы, аналогичные сигналам основного двигателя.
- 10 дискретных датчиков: Датчики положения лифта, положения дверей, положения удерживающих устройств, а также датчики перебега.
- Замок дверей. Для него предусмотрены следующие сигналы:
 - Управление замком (открытие замка по условию)
 - Сигнал от датчиков состояния дверей
- Для управления системой предусмотрены кнопки «Вверх», «Вниз», «Открыть двери», «СТОП» (авария), «RESET» (сброс аварии).

2. Выбор ПЛК

Для выполнения поставленной задачи требуется подобрать подходящий программируемый логический контроллер. Так как в настоящее время существует множество различных производителей и типов ПЛК, необходимо определиться с требованиями к используемому устройству. Помимо технических параметров так же надо учесть то, что контроллер должен быть достаточно популярным. Это позволит быстро находить необходимую информацию и документацию. В связи с этим был выдвинут следующий список требований:

- Возможность подключения достаточного для выполнения поставленной задачи количества модулей входов/выходов
 - Достаточная производительность
 - Наличие удобной среды программирования
 - Популярность модели
 - Наличие программных средств для осуществления визуализации
 - Возможность симуляции

Исходя из поставленных требований, было принято решение использовать ПЛК компании Siemens семейства SIMATIC S7-300.

Из данного семейства был выбран контроллер S7-300 CPU313C-2DP (см. рис. 1), так как он обладает встроенным блоком DI16/DO16 (16 дискретных входов и выходов), а также блоком для подключения к сети PROFIBUS. Также необходимо использовать дополнительный блок на 16 дискретных входов SM321 DI16xDC24V.



Рис. 1. Контроллер S7-300 CPU313C-2DP

3. Среда программирования

Для программирования предлагается использовать среду SIMATIC Step7. Данная среда в базовом варианте позволяет программировать на языках стандарта МЭК 6-1131/3: LAD (язык релейно-контактной логики), STL (язык списка инструкций), FBD (язык функциональных блочных диаграмм).

Особенностью программирования данного контроллера является то, что текст программы размещается в сегментах (Network). Каждый из них содержит маленький кусочек кода, решающего определенную задачу. При выполнении программы все сегменты исполняются параллельно.

В выбранной среде управляющая программа подразделяется на различные блоки. “Главным” блоком является ОВ (организационный блок). Он обрабатывается в CPU циклически. CPU последовательно исполняет указанные в нем команды программы. Когда

CPU возвращается к первой строке программы, он завершает ровно один цикл. Все остальные блоки, для задействования их в программе должны быть вызваны из блока ОВ.

4. Создание ФВ для управления двигателями

Управление двигателями лифта является одной из основных поставленных задач. Для этой цели можно создать функцию (FC) или использовать функциональный блок (FB).

В отличие от функционального блока, функция не может работать с блоками данных (DB). При этом для функции не допускаются статические локальные данные.

Для вызова функционального блока в организационном блоке (OB) необходимо создать экземплярный блок данных. При этом один организационный блок может работать с несколькими собственными блоками данных. Это означает, что один организационный блок может управлять работой всех используемых в лифте двигателей. Централизованное программирование функционального блока позволит существенно сократить объем программы, сделать ее более читаемой, а также упростить отладку.

На основании изложенного выше, было принято решение создать один общий функциональный блок для управления двигателями лифта. Схематичное изображение выбранной реализации представлено на рис. 2.

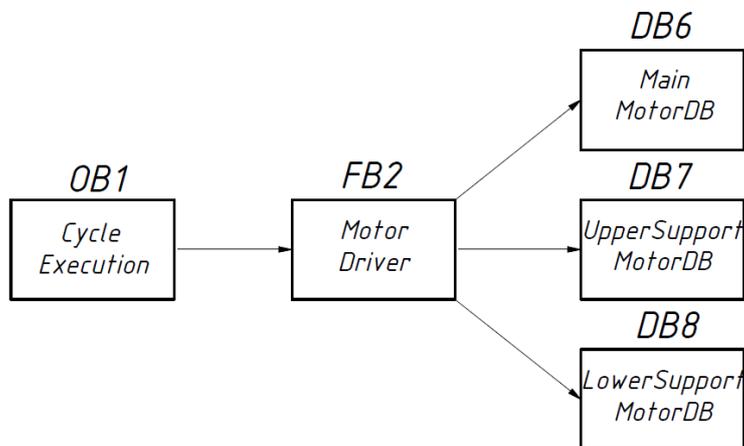


Рис. 2. Выбранная схема управления двигателями

Определившись с общей схемой управления двигателями можно приступить к формированию алгоритма работы УП.

При создании функционального блока управления двигателями в первую очередь

требуется определить все параметры, передающиеся функциональному блоку из других блоков как входные и выходные параметры. Значения этих параметров задаются при вызове функционального блока из организационного отдельно для каждого блока данных. Пример созданного при введении параметров блока данных представлен на рис. 3.

	Address	Declaration	Name	Type	Initial value	Actual value	Comment
1	0.0	in	CMD_Forward	BOOL	FALSE	FALSE	Команда на запуск двигателя в нормальном режиме
2	0.1	in	MotorDef	BOOL	FALSE	FALSE	Автомат защиты двигателя
3	0.2	in	MagnetStarter1	BOOL	FALSE	FALSE	Магнитный пускатель нормального режима двигателя
4	0.3	in	CMD_Reverse	BOOL	FALSE	FALSE	Команда на запуск двигателя в реверсивном режиме
5	0.4	in	MagnetStarter2	BOOL	FALSE	FALSE	Магнитный пускатель реверсивного режима двигателя
6	0.5	in	Reset	BOOL	FALSE	FALSE	Сброс аварий
7	2.0	in	Timer1	TIMER	T 0	T 0	
8	4.0	in	Timer2	TIMER	T 0	T 0	
9	6.0	in	Timer3	TIMER	T 0	T 0	
10	8.0	in	Timer4	TIMER	T 0	T 0	
11	10.0	in	EM_Stop	BOOL	FALSE	FALSE	Авария с кнопки
12	12.0	out	MotorNorm	BOOL	FALSE	FALSE	Включение двигателя в норм режиме
13	12.1	out	MotorRev	BOOL	FALSE	FALSE	Включение двигателя в реверсивном режиме
14	14.0	in_out	Alarm	BOOL	FALSE	FALSE	Авария
15	14.1	in_out	KM_Timer	BOOL	FALSE	FALSE	Перерыв между повторным запуском двигателя
16	14.2	in_out	KM1_Timer	BOOL	FALSE	FALSE	Таймер на получение ОС от пускателя 1
17	14.3	in_out	KM2_Timer	BOOL	FALSE	FALSE	Таймер на получение ОС от пускателя 2
18	16.0	in_out	KM1_Status	INT	0	0	Статус KM1
19	18.0	in_out	KM2_Status	INT	0	0	Статус KM2

Рис. 3. Пример созданного блока данных для FB2

Так как разрабатываемый функциональный блок отвечает исключительно за работу двигателей, то предполагается, что поступление сигнала на запуск двигателя будет приходить из другого блока. Соответственно, двигатели не могут запускаться без наличия данного сигнала. Также, при разработке УП двигателя необходимо учесть все особенности его работы. Для этого необходимо создать таймер, который будет запускаться при отключении пускателя (запускается при условии, что оба пускателя выключены). Пока данный таймер активен, запуск двигателя невозможен.

Второй мерой безопасности является наличие автомата защиты двигателя. При возникновении серьезных неполадок в электрической сети автомат защиты предотвращает поломку двигателя, разрывая цепь питания. Запуск двигателя может осуществляться только при наличии ОС от автомата защиты. В случае отсутствия ОС необходимо формирование сигнала об аварии двигателя.

Сигнал об аварии должен также формироваться в случае отсутствия сигнала ОС от соответствующего пускателя в течение 3 секунд после запуска двигателя. Для выполнения этого требования необходимо создать два таймера. Первый должен срабатывать при

поступлении сигнала на запуск на пускатель 1, второй при поступлении аналогичного сигнала на пускатель 2.

Последним условием формирования сигнала об аварии является нажатие кнопки «Авария».

При поступлении сигнала об аварии все двигатели блокируются, причем запуск становится возможным только после нажатия кнопки «Reset». Это необходимо для осуществления мер безопасности при нештатной работе системы.

Помимо перечисленных выше условий, необходимо программно исключить возможность срабатывания пускателей 1 и 2 одновременно. Это означает, что в условия запуска одного пускателя требуется добавить отсутствие сигнала и команды на запуск другого.

Двигатель должен прекращать работу при отсутствии любого из требуемых условий.

5. Создание FC главной управляющей программы

В описанном ранее функциональном блоке реализована управляющая программа двигателей, которая срабатывает только при поступлении внешнего сигнала на запуск и выполнении необходимых условий работы двигателя. Однако сам сигнал на запуск приходит из функции, ответственной за управление и координацию работы всей системы в целом.

Ниже будут рассмотрено формирование сигналов для всех трех двигателей, а также управление замком лифта.

Прежде всего, требуется создать четыре триггера, необходимых для того, чтобы наложить дополнительные условия на срабатывание различных частей (Network) программы. Их наличие позволяет варьировать работу лифта при схожих входных сигналах, а также дает возможность управлять системой в случае срабатывания датчиков перебега.

1) Триггер кнопки «Вверх» («Активна кнопка вверх»)

- Включение:

(«Кнопка вверх»=1) и («Нижний датчик выхода из рабочей зоны»=0)

- Выключение:

(«Верхний датчик положения лифта»=1)

2) Триггер кнопки «Вниз» («Активна кнопка вниз»)

- Включение:

(«Кнопка вниз»=1) и («Верхний датчик выхода из рабочей зоны»=0)

- Выключение:

(«Нижний датчик положения лифта»=1)

3) Триггер перебега вверх

- Включение:

(«Верхний датчик выхода из рабочей зоны»=1)

- Выключение:

(«Верхний датчик положения лифта»=1)

4) Триггер перебега вниз

- Включение:

(«Нижний датчик выхода из рабочей зоны»=1)

- Выключение:

(«Нижний датчик положения лифта»=1)

Описанные выше триггеры, а также некоторые другие элементы программы (см. приложения 2, 3) работают с меркерной памятью. Меркеры хранят информацию, доступную из любой части программы.

Меркеры могут рассматриваться как дополнительные «пусковые реле» контроллера. Меркеры используются в первую очередь для хранения состояния сигналов. Они могут трактоваться как виртуальные выходы. Меркеры располагаются в области системной памяти CPU, и, следовательно, они всегда доступны. Наличное число меркеров зависит от типа выбранного CPU.

Меркеры используются для хранения промежуточных результатов, которые действительны за пределами блока, и могут обрабатываться более чем в одном блоке.

Как было указано ранее, данные триггеры являются самостоятельными частями программы, написаны в отдельных Network-ах и будут использоваться в других частях УП.

6. Формирование сигналов управления двигателями

Для формирования сигналов управления основным двигателем предлагается использовать SR-триггеры. Это необходимо для того, чтобы запускать двигатель при наличии определенных условий (SET) и останавливать его только тогда, когда сработает набор условий остановки (RESET). При этом двигатель будет продолжать работу даже при изменении состояния системы после пуска.

1) Формирование «Команда на запуск основного двигателя в нормальном режиме»=1:

Для формирования команды на запуск двигателя в нормальном режиме необходимо, в первую очередь, чтобы поступил сигнал от кнопки «вверх». Далее, необходимо проверить положение лифта. Запуск двигателя в нормальном режиме возможен только в том случае, если активен нижний датчик положения лифта или нижний датчик выхода из рабочей зоны. При этом необходимо удостовериться в том, что все удерживающие устройства находятся вне шахты лифта и их двигатели выключены. В противном случае может произойти авария в связи со столкновением удерживающих устройств и лифта. Также необходимо проверить, закрыты ли двери лифта, запуск двигателя лифта с открытыми дверями категорически запрещен в связи с мерами безопасности.

2) Формирование «Команда на запуск основного двигателя в нормальном режиме»=0:

Остановка двигателя должна осуществляться при условии достижения лифтом своего верхнего датчика положения. В случае сбоя верхнего датчика положения, либо других обстоятельств, и достижения лифтом верхнего датчика перебега двигатель также должен быть отключен.

В том случае, если активен триггер перебега вниз (двигатель был запущен в нормальном режиме после перебега вниз), двигатель должен останавливаться при достижении своего нижнего датчика положения.

3) Формирование «Команда на запуск основного двигателя в реверсивном режиме»=1:

Как и в случае с запуском двигателя в нормальном режиме, необходимо, чтобы поступил сигнал от кнопки, в данном случае «вниз». Далее, необходимо проверить положение лифта. Запуск двигателя в реверсивном режиме возможен только в том случае, если активен верхний датчик положения лифта или верхний датчик выхода из рабочей зоны. При этом необходимо удостовериться в том, что все удерживающие устройства находятся вне шахты лифта и их двигатели выключены. В противном случае может произойти авария в связи со столкновением удерживающих устройств и лифта. Также необходимо проверить, закрыты ли двери лифта, запуск двигателя лифта с открытыми дверями категорически запрещен в связи с мерами безопасности.

4) Формирование «Команда на запуск основного двигателя в реверсивном режиме»=0:

Остановка двигателя должна осуществляться при условии достижения лифтом своего нижнего датчика положения. В случае сбоя нижнего датчика положения, либо других обстоятельств, и достижения лифтом нижнего датчика перебега двигатель также должен быть отключен.

В том случае, если активен триггер перебега вверх (двигатель был запущен в реверсивном режиме после перебега вверх), двигатель должен останавливаться при достижении своего верхнего датчика положения.

Удерживающие устройства данного лифта предназначены для того, чтобы в момент отключения двигателя поддерживать лифт в принятом положении. Однако, наличие устройств в шахте во время работы двигателя лифта недопустимо. Это означает, что перед запуском двигателя лифта удерживающее устройство должно уйти из шахты. В связи с этим, при нажатии кнопок «вверх» и «вниз» в первую очередь необходимо убрать удерживающие устройства из шахты, а только после этого запускать основной двигатель лифта.

7. Отладка программы

Для отладки созданной программы необходимо запустить существующий проект в режиме Online. Это можно сделать либо с помощью подключения ПЛК к компьютеру со средой разработки или с помощью использования симулятора ПЛК. В рамках данного проекта решено было использовать симулятор S7-PLCSIM (см. рис.4).

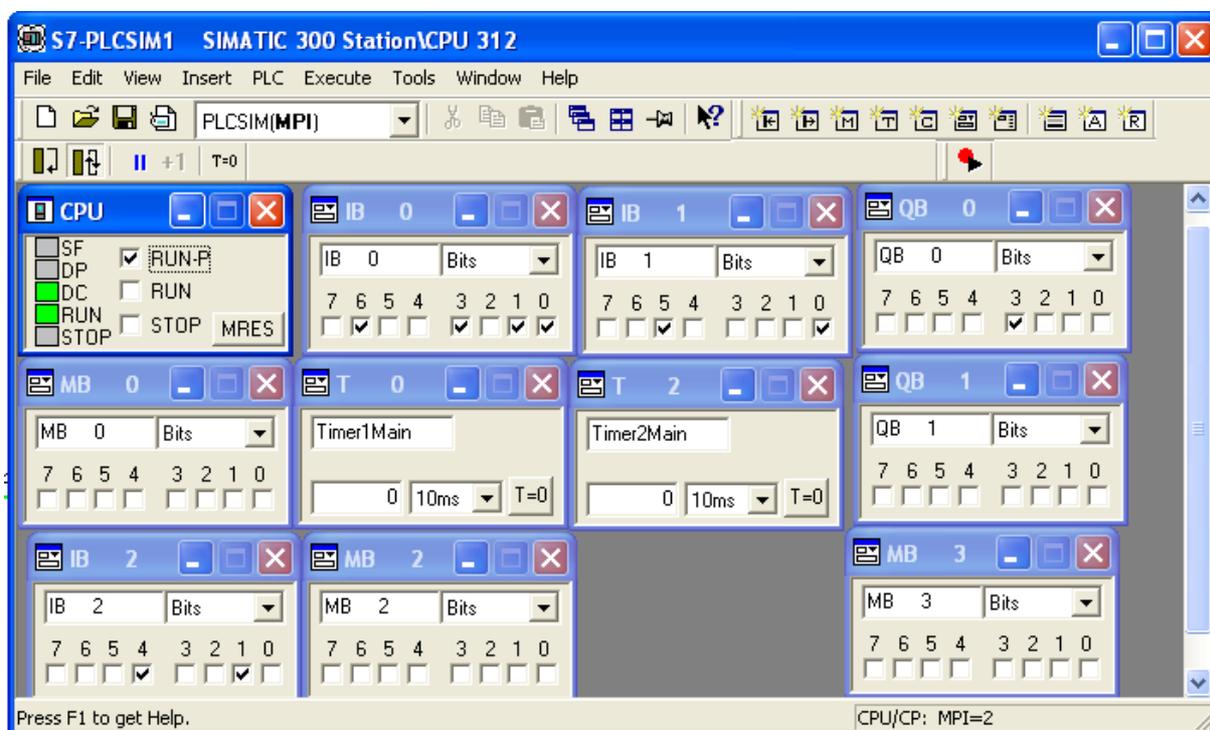


Рис. 4. Окно симулятора S7-PLCSIM

Данный симулятор позволяет задавать значения входов (блоки IV), меркерной памяти (блоки MB) и ряда других параметров, а также отслеживать значения выходов, таймеров и т.д.

При запуске проекта в режиме Online отладки (см. рис. 5) отображаются значения элементов LAD-диаграмм. Это позволяет проверить работоспособность созданной программы, устранить ошибки и т.д.

В данном случае на рис. 4 и рис. 5 демонстрируется ситуация, в которой лифт находится на втором этаже, причем двери лифта закрыты.

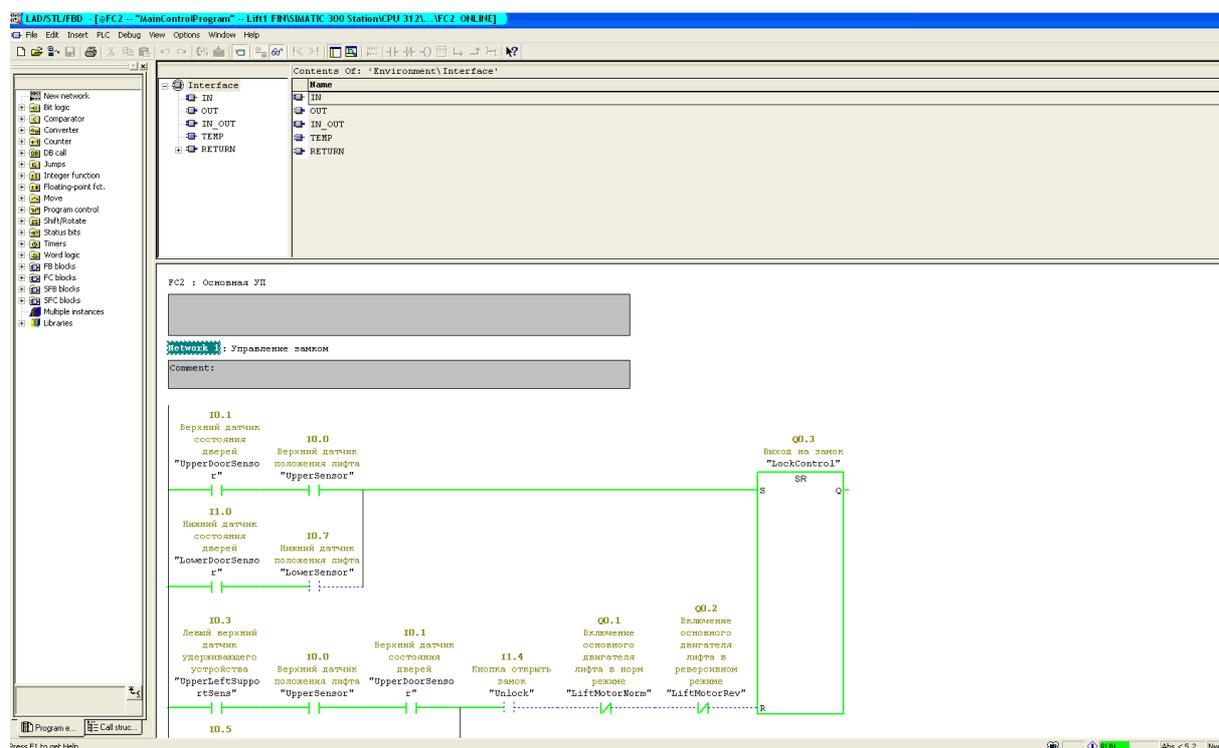


Рис. 5. Режим отладки

8. Разработка визуализации

В ходе работы была поставлена задача не только разработать систему управления лифтом, но и провести наглядную симуляцию его работы. Описанный выше симулятор позволяет имитировать работу ПЛК, однако при этом отсутствует какая-либо наглядность, так как каждый номер входа и выхода симулятора приходилось бы соотносить с объектом с помощью таблицы символов.

Для проведения наглядной симуляции было принято решение создать визуализацию

работы лифта в программе SIMATIC WinCC flexible.

В данном случае каждый из объектов визуализации был создан при помощи использования простых объектов типа «кнопка», «круг», «изображение» и т.д. Каждый используемый объект связывается (рис. 6) с символами из таблицы символов соответствующего проекта, что позволяет задавать и отслеживать значения параметров прямо в визуализационной модели путем взаимодействия с простыми объектами и наблюдения за их состоянием.

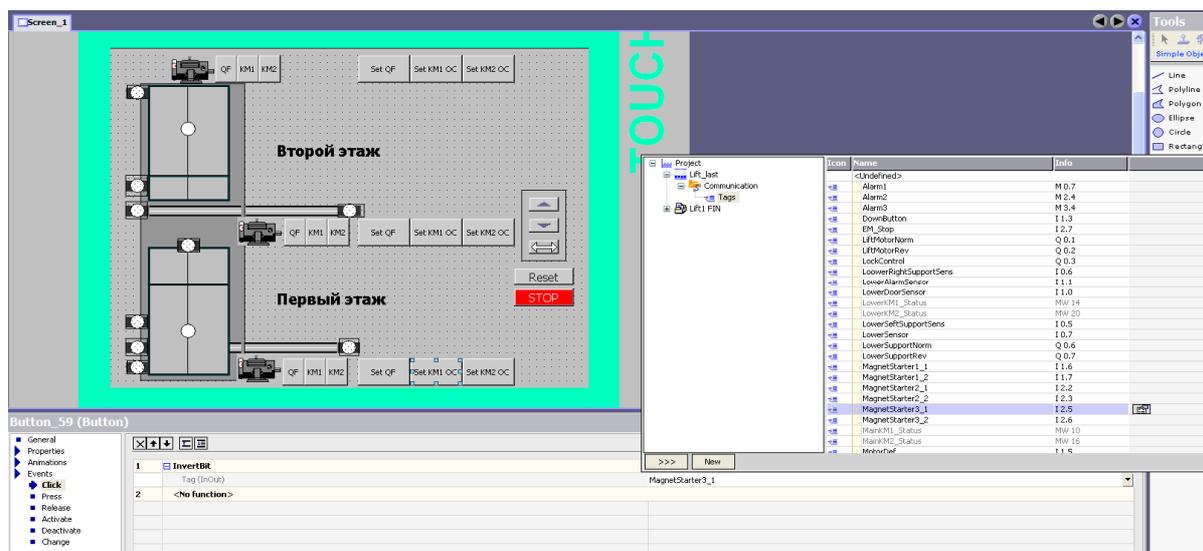


Рис. 6. Привязка объекта к символу

Заключение

Описанная система управления грузовым лифтом является актуальной разработкой, позволяющей сократить затраты на создание систем управления лифтами, в том случае, если один контроллер управляет работой нескольких лифтов.

Для выбранного ПЛК была разработана управляющая программа на языке LAD (Релейно-контактных схем), предусматривающая программную защиту используемых двигателей. В алгоритм управляющей программы были заложены требования по обеспечению безопасности груза и оператора.

Для разработанной системы управления была создана симуляционная модель лифта, с помощью которой была проверена управляющая программа.

Список литературы

- [1]. Berger.H. Automating with STEP7. Erlangen, Publicis Publishing, 2012.
- [2]. SIMATIC S7-300 Tutorial for First Time Users. Режим доступа: <https://support.industry.siemens.com/cs/document/27006123/simatic-s7-300-tutorial-for-first-time-users?dti=0&pnid=13752&lc=en-WW> (дата обращения 14.03.2016).
- [3]. SIMATIC S7-300 Module data. Режим доступа: <https://support.industry.siemens.com/cs/document/8859629/simatic-s7-300-s7-300-module-data?dti=0&lc=en-WW> (дата обращения 14.03.2016).
- [4]. Митин Г.Л., Хазанова О.В. Системы автоматизации с использованием программируемых логических контроллеров: учебное пособие. М.: ИЦ МГТУ «Станкин», 2005. 136 с.