

Модульные программируемые контроллеры «К15»: преимущества аппаратного и программного обеспечения



Контроллеры линейки «К15», разработанные российской компанией, построены по модульному принципу, что дает им ряд неоспоримых преимуществ. В статье рассмотрены исполнения ЦПУ и модулей ввода/вывода, входящих в состав ПЛК «К15». Также показаны особенности программирования, в качестве примера приведено решение прикладной задачи.

ООО «Кастом Инжиниринг», г. Уфа

Все последние годы на рынке автоматизации было представлено множество модульных и моноблочных программируемых контроллеров с разной функциональностью и ценой, изготовленных разными производителями. Выбирая ПЛК для своего проекта, заказчики руководствовались не только материальными и практически соображениями, но и привычкой, отдавая предпочтение хорошо знакомым, проверенным брендам, очень часто иностранным. Однако сегодня российскому потребителю приходится искать нужное устройство среди отечественного оборудования, зачастую нового, которое все больше появляется на российском рынке. Новинкой можно назвать и серию модульных программируемых контроллеров «К15».

Модульный принцип построения

Контроллеры линейки «К15» построены по модульному принципу и имеют классическую для ПЛК такого типа конструкцию: центральное процессорное устройство (ЦПУ) плюс большой набор модулей ввода/вывода. Модульные решения, в отличие от моноблочных контроллеров или ПЛК смешанного типа, обладают большой гибкостью. Они позволяют создать

локальную систему управления ровно под те задачи, которые требуется решить в настоящий момент, не переплачивая за лишние функции. При этом такая система обладает хорошей масштабируемостью: ее всегда можно с легкостью расширить и усложнить, просто докупив необходимые модули.

Еще одно преимущество модульных контроллеров — простота и дешевизна эксплуатации. Дорогостоящий ПЛК не придется менять из-за пары каналов ввода/вывода, вышедших из строя. Достаточно поменять неисправный модуль.

У серии «К15» есть и свои особенные преимущества, которыми она отличается от других модульных ПЛК. Во-первых, это эргономичность. Модули отличаются малой шириной и крепятся на DIN-рейку. Все модули, включая ЦПУ, одного форм-фактора, что существенно упрощает компоновку при проектировании шкафного оборудования. Во-вторых, следует отметить удобное расположение интерфейсной шины, соединяющей модули с ЦПУ: она уложена непосредственно в DIN-рейку и фиксируется в ней от выпадения, что позволяет менять модули, не разбирая корзину и даже не

отключая питания. К этой особенности мы вернемся позже, а пока рассмотрим компоненты ПЛК.

Исполнения ЦПУ

Ключевым компонентом ПЛК является центральное процессорное устройство. В линейке «К15» представлены три модели ЦПУ — F1, F4 и H7, которые позволяют выполнить широкий спектр задач автоматизации. Для интеграции с системами верхнего уровня, а также для построения распределенных систем все три модели ЦПУ реализуют протокол обмена Modbus, доступный через все имеющиеся интерфейсы. Наряду с этим предусмотрена возможность реализации нестандартных протоколов обмена.

ЦПУ F1 имеет наименьшую мощность из всех трех моделей и подойдет для небольших систем, где требуется недорогое решение с сохранением всех преимуществ модульной схемы.

F4 (рис. 1а) — более мощное ЦПУ с интерфейсом Ethernet. Наряду с увеличенной тактовой частотой процессора, большим объемом памяти RAM и Flash, имеет встроенный веб-интерфейс, который позволяет следить за состоянием корзины модулей, ее



Рис. 1. ЦПУ модульного контроллера «K15»: а – K15.CPU.F4; б – K15.CPU.H7

составом, осуществлять мониторинг и диагностику в онлайн-режиме. Также веб-интерфейс дает возможность загружать проект в ЦПУ без применения программатора и обновлять ПО подключенных модулей ввода/вывода.

Самым производительным ЦПУ в линейке контроллеров «K15» является H7 (рис. 1б). Достаточно мощный процессор, а также внешний флеш-накопитель позволяют ему выполнять довольно сложные алгоритмы с хорошим быстродействием. Также стоит отметить применение энергонезависимой памяти FRAM вместо EEPROM. Эта версия тоже имеет веб-интерфейс, обеспечивающий удаленную работу.

Все ЦПУ работают под управлением операционной системы реального времени, что повышает надежность и быстродействие системы, позволяет полностью использовать аппаратные возможности процессоров.

Модули ввода/вывода: широкая номенклатура и возможность «горячей замены»

Линейка «K15» может похвастаться обширной номенклатурой модулей ввода/вывода: от самых простых до гальванически развязанных аналоговых модулей вывода, счетных модулей, ШИМ-модулей и других. Комбинируя их, можно создавать самые разные вариации системы: ПЛК совсем без модулей, с одним модулем или сразу с восемью, подключенными к одному ЦПУ, и т. д.

Ключевая особенность всех модулей ввода/вывода линейки «K15» — механизм взаимодействия с ЦПУ, о котором мы уже упоминали. Рассмотрим его подробнее.

Между ЦПУ и модулями ввода/вывода предусмотрена интерфейсная системная шина. Она выполнена на базе CAN-интерфейса и физически представляет собой 5-контактную шину T-Bus, на которую, при монтаже контроллера на DIN-рейку, устанавливаются все его части. Так как шина расположена в задней части корпуса модулей со стороны рейки, появляется возможность снимать и устанавливать модули по отдельности, без демонтажа корзины.

Но самое интересное состоит в том, что модули ввода/вывода можно заменять, даже не отключая питания, то есть выполнять «горячую замену». Это очень важно для объектов, на которых отключение и перезапуск локальной системы управления связаны с серьезными производственными издержками.

Разработчики планируют выпустить специальную серию модулей ввода/вывода «K15», оснащенных интерфейсом RS-485, на котором будет реализован все тот же популярный протокол Modbus. Это позволит произвести буквально помодульное дооснащение имеющихся систем управления, как распределенных, так и локальных, в тех проектах, где уже имеются ЦПУ, но не хватает каналов ввода/вывода.

Стоит отметить, что в «K15» для создания проектов используются не языки МЭК, которые позволяют без глубоких знаний программирования создавать управляющие программы для контроллеров, а языки высокого уровня Си и Си++. Конечно, это не самый популярный метод работы с модульными системами в сфере автоматизации, хотя, например, в контроллерах Siemens, B&R, Allen Bradley, Schneider в качестве скриптовых языков используются именно C, C++, C# и VBA. Какие же преимущества дает создание проектов таким способом?

Для начала не стоит пугаться формулировки «язык высокого уровня»: язык Си очень похож на один из языков МЭК — ST (Structured Text). По сути, это все те же переменные, циклы, условия, переходы. Но если ST позволяет реализовать лишь то, что заложено в Runtime-оболочке контроллера, которая является исполняемой средой для проекта, то язык Си позволяет выйти за рамки этой среды. Разработчику доступен весь синтаксический инструментарий этого мощного орудия программирования. К тому же многие алгоритмические приемы на языках Си реализуются проще и понятнее, а создание структур, приведение типов, callback-функции, перегруженные функции и прочие приемы программирования — приятный бонус использования языка.

Во-вторых, у ПЛК «K15» удобная среда разработки. Далеко не все среды разработки проектов для модульных систем легки и понятны в освоении. Например, описание работы со средой TIA Portal занимает более 1000 страниц руководства программиста. Не менее сложна в освоении и Studio 5000 для контроллеров Allen Bradley серий Control и CompactLogix. А чтобы уверенно работать в CODESYS, придется не один месяц осваивать эту среду программирования.

Для контроллеров «K15» применяется среда разработки CubeIDE. Это официальная свободно распространяемая IDE от компании STMicroelectronics (рис. 2), чьи процессоры сегодня являются главным элементом контроллеров. В комплекте с ПЛК «K15» идет стартовый проект и подробная инструкция с описанием работы. Достаточно просто от-

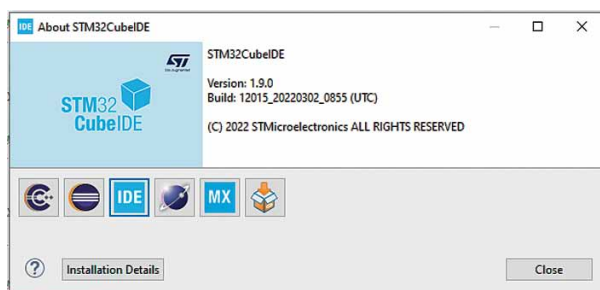


Рис. 2. IDE от компании STMicroelectronics

крыть проект в CubeIDE, скомпилировать его и отправить в контроллер. Система начнет работать, отображать веб-интерфейс, опрашивать модули. Потом, используя описанные структуры и функции, можно реализовать задуманный проект.

Третье преимущество — доступность среды разработки и ценовой аспект. Кто приобретал контроллеры Emerson, Siemens, B&R, Yokogawa, знает, что лицензии для работы с ними стоят весьма дорого. Конечно, такие системы, как CODESYS, Veremiz, OwenLogic, вообще не требуют лицензий, но ведь они поддерживают далеко не все модульные ПЛК. Как правило, производители ПО (DeltaV, TIA Portal, FactoryTalk и пр.) стараются разработать платформозависимый программный продукт либо собственную экосистему. И если до недавнего времени все эти программные продукты можно было хотя бы свободно приобрести, то сегодня их разработчики ушли с российского рынка, поэтому купить их будет затруднительно. Среда CubeIDE, которую использует «K15», доступна и не требует лишних затрат на приобретение лицензий.

Четвертое преимущество — процесс отладки проекта. Ключевая осо-

бенность языков МЭК — механизм онлайн-трассировки и отладки проекта. «K15» также может похвастаться этой функциональностью. Но отладка здесь, как и подобает любому языку высокого уровня, гораздо более многосторонняя и глубокая. Тут и классический вывод текущих переменных, и точки останова, и принудительная запись значений, и пошаговое исполнение кода. Также есть возможность вернуться на шаг назад либо выполнить остановку исполнения кода по условию. Для удобства разработчика реализована контекстная подсветка значений переменных при наведении курсора.

Одним словом, вся мощь языков Си к вашим услугам. Это делает разработку быстрее и качественнее, а дальнейшее сопровождение и рефакторинг проекта — дешевле для конечного потребителя. Убеденным сторонникам МЭК переход на языки Си/Си++ может показаться непростым, но усилия, потраченные на их изучение, с лихвой окупятся в дальнейшем.

Пример решения прикладной задачи

Рассмотрим на конкретном примере, как решаются прикладные задачи средствами разработки «K15». До-

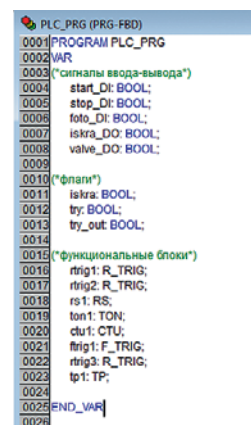


Рис. 3. CODESYS: создание переменных и функциональных блоков

пустим, имеется факельная установка с электроискровым розжигом дежурной горелки и фотодатчиком наличия пламени. Задача разработчика — реализовать режим автоматического розжига дежурной горелки.

Клапан топливного газа должен открываться при подаче сигнала пуска, а розжиг должен происходить циклично до тех пор, пока не загорится пламя. Допускается три попытки розжига. Также алгоритм должен обеспечивать автоматический перерозжиг горелки в случае погасания пламени.

Сначала попытаемся реализовать задуманное средствами CODESYS. Пусть это будет популярный язык МЭК FBD. Создадим необходимые переменные и функциональные блоки (рис. 3).

- ▶ start_DI — сигнал пуска системы (например, кнопка);
- ▶ stop_DI — сигнал остановки системы;
- ▶ foto_DI — фотодатчик наличия пламени;

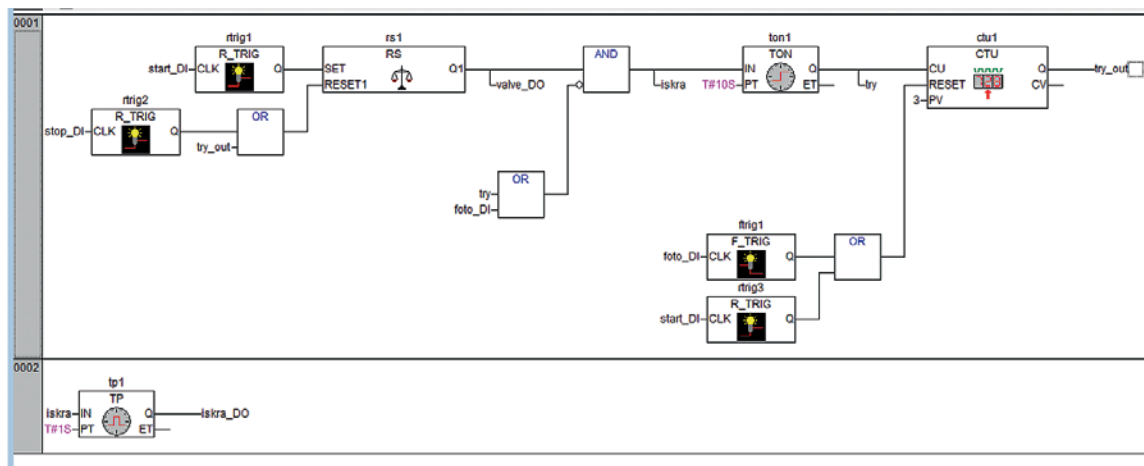


Рис. 4. CODESYS: функциональная схема, реализующая алгоритм

- iskra_DO — управление подачей искры;
- valve_DO — управление клапаном.

Затем создаем функциональную схему, реализующую алгоритм (рис. 4). Рассмотрим ее по порядку.

При подаче сигнала пуска через детектор переднего фронта срабатывает триггер rs1, принимая логическое состояние 1. Открывается клапан, далее сигнал следует на блок AND. Так как флаг совершения попытки розжига try и сигнал с фотодатчика имеют логический 0, то инверсивный сигнал на входе блока имеет 1, и происходит подача искры через таймер импульсного сигнала tp1 во второй строке схемы. Одновременно с этим запускается таймер ожидания пламени ton1. По завершении его счета, спустя 10 секунд, выставляется флаг совершения попытки. Вместе с тем счетчик ctu1 производит инкрементирование числа попыток на 1, сравнивая полученное число с уставкой, равной трем. При достижении трех попыток взводится флаг try_out, который вызывает сброс триггера rs1. Также его сброс вызывает и подачу сигнала остановки системы розжига.

При установке флага try происходит изменение значения флага iskra сначала на 0, затем вновь на 1. Тем самым запускается новый цикл розжига. Здесь следует иметь в виду, что переменная try должна быть объявлена как глобальная и не сбрасываться при каждом цикле скана используемого ROU.

В случае горения флаг foto_DI принимает логическое состояние 1.

```
bool start_DI;
bool stop_DI;
bool foto_DI;
bool iskra_DO;
bool valve_DO;

uint16_t state, count, try_count;
```

Рис. 5. Решение прикладной задачи в среде разработки CubeIDE: переменные

```
67
68
69
70 switch(state){
71 case 0: //стадия СТОП
72     valve_DO=false;
73     iskra_DO=false;
74     count=0;
75     try_count=0;
76     if(start_DI){state=1;};
77     break;
78 case 1: //стадия РОЗЖИГ
79     valve_DO=true;
80     if(count<1000){iskra_DO=true; //подача искры 1 сек
81         osDelay(1);
82         count++;
83     }else{iskra_DO=false;};
84     if(count==1000){count=0; //ожидание пламени 10 сек
85         try_count++;
86     };
87     if(foto_DI){state=2;};
88     if(stop_DI || (try_count==3)){state=0;}; //3 попытки
89     break;
90 case 2: //стадия ГОРЕНИЕ
91     valve_DO=true;
92     iskra_DO=false;
93     if(!foto_DI){count=0;
94         try_count=0; //сбросить число попыток
95         state=1;
96     };
97     if(stop_DI){state=0;};
98     break;
99 };
100
---
```

Рис. 6. Решение прикладной задачи в среде разработки CubeIDE

Благодаря этому через инверсию блок AND на выходе принимает логическое состояние 0, блокируя цикл розжига. Если пламя гаснет, блок AND снова принимает состояние 1 и цикл розжига начинается снова.

Таким образом, для реализации задуманного потребовалось создать, помимо простых переменных, еще восемь функциональных блоков, что для такой простой задачи немало.

Теперь попробуем решить ее в рамках программной функциональности «K15». Весь код размещаем в файле UCL.c. Здесь также не обойтись без объявления переменных (рис. 5).

Для единообразия сигналы имеют схожее обозначение. Дополнительно добавились переменные state (стадия работы системы), count (аккумулятор счетчика времени) и try_count (аккумулятор числа попыток розжига). Далее обратимся к самому коду (рис. 6).

Алгоритм выполняется в основном цикле файла. Работа системы разбита на стадии: СТОП, РОЗЖИГ и ГОРЕНИЕ. Код достаточно компактный и читаемый. Создание дополнительных функциональных блоков не требуется. Следует обратить внимание, что число стадий может быть легко увеличено, например, можно добавить стадию АВАРИЯ.

Даже на примере этого небольшого проекта с тривиальными задачами читатель может заметить преимущества программирования «K15», простоту сопровождения и масштабирования проекта.

Заключение

Создатели «K15» поддерживают обратную связь с клиентами, разрабатываемыми свои проекты на базе этого ПЛК, и учитывают их критику и пожелания. Программная и аппаратная части контроллера постоянно совершенствуются. Любая неполадка или возникший вопрос обрабатываются в течение 24 часов квалифицированными инженерами, которые проконсультируют клиента и предложат варианты решения проблемы. Если вопрос возник при создании проекта, окажут помощь программисты.

Модульные ПЛК «K15» — это постоянно развивающийся, динамичный продукт, способный составить достойную конкуренцию продукции ушедших вендоров.

Д. А. Гришин, инженер-программист,
ООО «Кастом Инжиниринг», г. Уфа,
тел.: 8 (800) 775-7470,
e-mail: info@custom-eng.ru,
сайт: www.custom-eng.ru



vk.com/journal_isup
ВКонтакте



<https://t.me/isupmagaz>
Телеграм



<https://dzen.ru/isup>
Дзен

Все новости и статьи в свободном доступе