

## Модульные программируемые контроллеры «К15» - ЭФФЕКТИВНОЕ ОТЕЧЕСТВЕННОЕ РЕШЕНИЕ, ОСНОВАННОЕ НА ЛУЧШИХ ПРАКТИКАХ

Д.А. Гришин (ООО «Эй энд Ти Текнолоджис»)

*Представлены конструктивные особенности и функциональные возможности модульных отечественных программируемых контроллеров серии «К15». Рассмотрены принципы программирования контроллеров в среде C++. Приведен пример проекта, реализованного с применением контроллера «К15».*

*Ключевые слова: модульность, программируемый контроллер, центральные процессорные устройства, модули ввода/вывода.*

### Введение

Модульные программируемые контроллеры «К15» реализуют классическую концепцию ПЛК: центральное процессорное устройство (ЦПУ) плюс «корзина» модулей ввода/вывода. Такой подход позволяет создать гибкую, масштабируемую локальную систему управления под задачи пользователей. Модульность конструкции позволяет в любой момент расширить и усложнить систему, заменять каналы, вышедшие из строя.

Серия ПЛК «К15» характеризуется высокой эргономичностью: классическое крепление на DIN рейку сочетается с малой шириной модулей. Все модули, включая ЦПУ, имеют один форм-фактор, что существенно упрощает компоновку при проектировании шкафного оборудования. Отметим также удобное расположение интерфейсной шины, соединяющей модули с ЦПУ: она уложена непосредственно в DIN рейку и фиксируется в ней от выпадения. Это дает возможность менять модули, не разбирая «корзину» и даже не отключая питания.

Рассмотрим компоненты ПЛК серии «К15» подробнее.

### Центральные процессорные устройства

Сердцем и ключевым компонентом системы «К15» является ЦПУ. Семейство ЦПУ включает три модели: F1, F4 и H7, способные покрыть достаточно широкий спектр потребностей автоматизации:

- модуль F1 предназначен для небольших задач, где требуется изящное недорогое решение с сохранением всех преимуществ модульной схемы (рис. 1а, таблица);

- модуль F4 – более мощный собрат F1, имеющий на борту Ethernet интерфейс. Помимо увеличенных тактовой частоты процессора, RAM и Flash, данное ЦПУ имеет встроенный Web-интерфейс. Он позволяет следить за состоянием корзины, ее составом, производить ее мониторинг и диагностику в реальном времени. Также Web-интерфейс дает возможность производить загрузку проекта в ЦПУ без применения программатора и обновлять программное обеспечение подключенных модулей ввода/вывода (рис. 1б, таблица);

- модуль H7 – наиболее производительное ЦПУ из представленных. Имея неплохие характеристики процессора, а также возможность применения внешнего Flash-накопителя в виде SD-карты, он способен реализовывать достаточно сложные алгоритмы с хорошим быстродействием. В этом модуле применяется FRAM вместо EEPROM в качестве энергонезависимой памяти. Присутствует Web-интерфейс (рис. 1с, таблица).

Для интеграции с системами верхнего уровня, а также для построения распределенных систем все ЦПУ поддерживают широко распространенный протокол обмена Modbus, доступный через все имеющиеся интерфейсы. Кроме того, возможна реализация нестандартных протоколов обмена.

Все ЦПУ работают под управлением ОС реального времени, что повышает надежность системы и ее быстродействие.

### Модули ввода/вывода

Линейка ПЛК «К15» включает обширную номенклатуру модулей ввода/вывода: от дискретных модулей ввода/вывода, до гальванически развязанных аналоговых модулей вывода, счетных модулей, PWM модулей и т.д. Комбинируя их в «корзине», можно создавать абсолютно произвольные системы. Причем это могут быть как системы совсем без модулей, с одним модулем либо сложные вариации до восьми модулей с одним ЦПУ.

Ключевая особенность всех этих модулей – механизм взаимодействия с ЦПУ. Между ЦПУ и модулями предусмотрена интерфейсная системная шина. Она выполнена на базе CAN интерфейса и физически представляет собой пятиконтактную шину T-Bus, на которую устанавливаются ЦПУ и модули в процессе монтажа на DIN рейку. Так как шина расположена в задней части корпуса модулей со стороны рейки, появляется возможность снимать и устанавливать модули без демонтажа всей «корзины».

Важно, что модули ввода/вывода можно заменять даже на работающей системе, не отключая питания (так называемая «горячая замена»). Это бывает весьма критично там, где отключение и перезапуск локальной системы управления связаны с нарушением технологического процесса или серьезными производственными издержками.

В ближайшей перспективе планируется выпуск специальной серии модулей ввода/вывода «К15», оснащенных интерфейсом RS-485, на котором будет реализован протокол Modbus. Это позволит произвести буквально помодульное дооснащение имеющихся локальных и распределенных систем управления без применения ЦПУ «К15» в тех проектах, где уже имеются ЦПУ, но не хватает каналов ввода/вывода.

### Программирование контроллера

В отличие от классических языков МЭК, которые позволяют без глубоких знаний программирования создавать управляющие проекты для систем автоматизации, «К15»

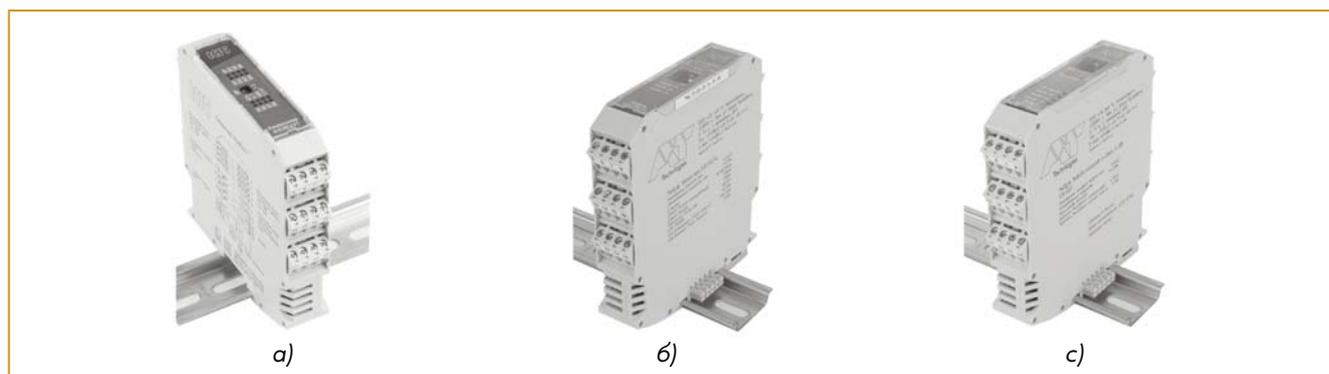


Рис. 1. Центральные процессорные устройства ПЛК «K15»: а) F1, б) F4, с) H7

Таблица. Технические характеристики процессорных модулей

Основные технические характеристики			
Процессорный модуль	K15.MCU.F1	K15.CPU.F4	K15.CPU.H7
Центральный процессор ARM® 32-bit	Cortex®-M3, 72 МГц	Cortex®-M4, 168 МГц	Cortex®-M7, 480 МГц
Web-интерфейс	-	+	+
Часы реального времени	+	+	+
Поддержка MicroSD	-	-	+
Возможность подключения дополнительных модулей ввода/вывода	+	+	+
Электрические характеристики			
Напряжение питания, В	24 ±20%		
Потребляемая мощность, не более, Вт	5		
Электрическая прочность изоляции цепей, В	500		
Защита входного напряжения	ограничение тока		
Число дискретных входов/выходов, ед	8/4	3/2	3/2
Число аналоговых входов, ед.	3	-	-
Коммуникационные характеристики			
Число изолированных портов RS-485, ед.	1	1	2
Число не изолированных портов RS-485, ед.	-	2	1
Число портов Ethernet 10/100 Base-T, ед	-	1	1
Число индикаторов передачи данных по RS-485, ед.	1	3	3
Индикаторы состояния (Status, Run, Fault)	+	+	+
Индикаторы дискретных сигналов	+	+	+
Поддерживаемые протоколы обмена	-	ModBus RTU/TCP	
Интерфейс обмена данными с модулями	-	CAN	
Механические характеристики			
Размеры (Д x Ш x В), мм	107x22,5x136		
Масса, грамм	400		
Степень защиты корпуса	IP20		
Крепление	DIN-рейка 35 мм		
Условия эксплуатации			
Температура, °С	-40...60		
Влажность, %	10...90		

предлагает создание проектов на широко распространенных языках высокого уровня C/C++. Это не совсем популярный метод работы с модульными системами в сфере автоматизации. Хотя в качестве скриптовых языков C, C++, C#, VBA используются в контроллерах Siemens, B&R, Allen Bradley, Schneider и других изделиях.

Перечислим преимущества, открывающиеся перед пользователями благодаря использованию языков высокого уровня.

Во-первых, C и C++ очень похожи 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 как бы исполняемой средой для проекта, то язык Си позволяет заглянуть за рамки этой среды. Пользователю доступен весь синтаксический инструментарий этого мощного орудия программирования. К тому же многие алгоритмические приемы на C/C++ реализуются проще и понятнее, а создание структур, приведение типов, callback функции, перегруженные функции и прочие приемы программирования – приятный бонус использования языка.

Во-вторых, немаловажное значение играет инструментарий разработки. Далеко не все среды разработки проектов под те или иные модульные системы 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 для уверенной работы в CodeSys нужен не один месяц освоения среды.

Семейство «K15» программируется в среде разработки CubeIDE – это официальная свободно распространяемая IDE от компании STMicroelectronics, чьи процессоры и являются главным элементом контроллеров на данный момент. В комплекте с контроллерами идет стартовый проект и подробная инструкция с описанием работы. Просто открываем проект в CubeIDE, компилируем его и «отправляем» в контроллер. Система уже будет работать, отображать Web-интерфейс, опрашивать модули. Затем, используя описанные структуры и функции, можно реализовать требуемые по проекту функции.

В-третьих, доступность среды разработки и ценовой аспект. Стоимость лицензий для работы с контроллерами зарубежных производителей составляет значительную сумму. Как правило, каждый производитель старается разработать свой инструментарий как платформозависимый программный продукт либо как свою экосистему. И если до недавнего времени все эти программные продукты можно было хотя бы свободно приобрести, то, учитывая повальный уход с рынка зарубежных игроков, это уже будет сделать весьма затруднительно. При этом специалисты, конечно, знают, что CodeSys, Veremiz, OwenLogic и подобные системы не требуют приобретения лицензий, однако далеко не все модульные системы ими поддерживаются. В этих условиях повышается привлекательность решения «K15», позволяющее использовать доступную среду CubeIDE, не требу-

ющую лишних затрат на приобретение лицензий.

Далее остановимся на процессе отладки проекта, непосредственно влияющем на скорость разработки. Одной из ключевых особенностей языков МЭК является механизм on-line трассировки и отладки проекта. «K15» также имеет эту функциональность, который изначально заложена в самой IDE. Но отладка здесь, как и подобает любому языку высокого уровня, гораздо более многосторонняя и глубокая.

Тут и классический вывод текущих переменных, и очки останова, и принудительная запись значений, и пошаговое исполнение кода. Также есть возможность вернуться на шаг назад либо выполнить остановку исполнения кода по условию. Дополнительным удобством для разработчика можно считать контекстную подсветку значений переменных при наведении курсора.

Одним словом, клиенты могут использовать всю мощь языков C/C++. Это делает разработку быстрее и качественнее, а дальнейшее сопровождение и рефакторинг проекта – дешевле для конечного потребителя. Конечно, переход на такой способ разработки для убежденных сторонников МЭК нельзя назвать совсем простым. Но некоторые усилия, потраченные на изучение языков Си, по крайней мере, в рамках поставленных задач, с лихвой окупятся в дальнейшем.

**Пример реализации задачи**

Теперь рассмотрим, как решаются прикладные задачи средствами разработки «K15». Возьмем вполне конкретный пример. Имеем факельную установку с электроискровым розжигом дежурной горелки и фотодатчиком наличия пламени. Задача: реализовать режим автоматического розжига дежурной горелки.

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

Сначала реализуем задуманное средствами Codesys на языке МЭК – FBD. Создадим необходимые переменные и функциональные блоки (рис. 2):

- start\_DI – сигнал пуска системы (например, кнопка);

```

PLC_PRG (PRG-FBD)
0001|PROGRAM PLC_PRG
0002|VAR
0003|(*сигналы ввода-вывода*)
0004|start_DI: BOOL;
0005|stop_DI: BOOL;
0006|foto_DI: BOOL;
0007|iskra_DO: BOOL;
0008|valve_DO: BOOL;
0009|
0010|(*флаги*)
0011|iskra: BOOL;
0012|try: BOOL;
0013|try_out: BOOL;
0014|
0015|(*функциональные блоки*)
0016|rtrig1: R_TRIG;
0017|rtrig2: R_TRIG;
0018|rs1: RS;
0019|ton1: TON;
0020|ctu1: CTU;
0021|ftrig1: F_TRIG;
0022|rtrig3: R_TRIG;
0023|tp1: TP;
0024|
0025|END_VAR|
0026|
    
```

Рис. 2. переменные и функциональные блоки в среде Codesys

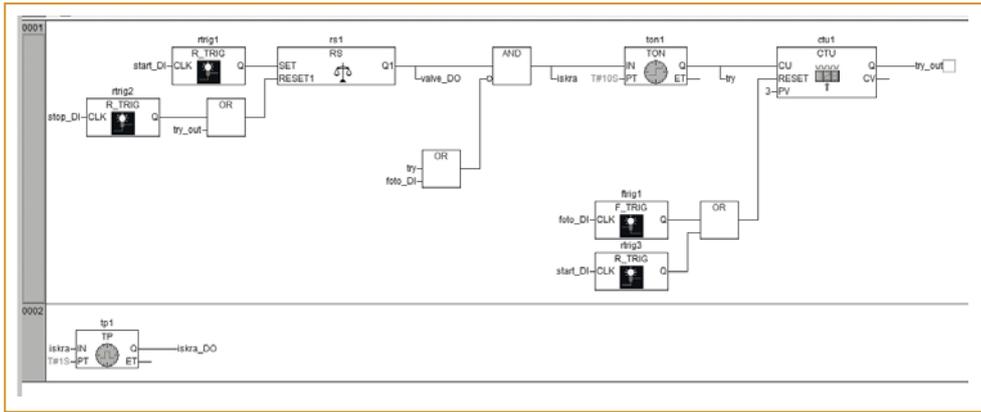


Рис. 3. Функциональная схема проекта

```

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

uint16_t state, count, try_count;
    
```

Рис. 4. Переменные в C++

```

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>=10000){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.5. Код проекта в C++

- stop\_DI – сигнал остановки системы;
- foto\_DI – фотодатчик наличия пламени;
- iskra\_DO – управление подачей искры;
- valve\_DO – управление клапаном.

Затем создаем функциональную схему, реализующую алгоритм (рис. 3). 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 завершения 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 tpr1 во второй строке схемы. Одновременно с этим запускается таймер ожидания пламени ton1.

По завершению его счета, спустя 10 секунд выставляется флаг завершения попытки. Вместе с этим 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 подача 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. Отметим, что переменная try должна быть объявлена как глобальная и не сбрасываться при каждом цикле скана используемого POU.

В случае горения флаг foto\_DI принимает логическое состояние 1. 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 задачу в рамках программной функциональности «K15». Весь код размещаем в файле UCL.c. Здесь также не обойтись без объявления переменных (рис. 4).

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. 5).

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. Отметим, что число стадий может быть легко увеличено, например, можно добавить стадию АВАРИЯ.

Этот небольшой проект демонстрирует хороший потенциал использования «K15» в плане программной реализации, простоту сопровождения и масштабирования проекта.

**Сопровождение**

Любая современная разработка не может обходиться без постоянного совершенствования, а также без исправления неизбежно существующих текущих недостатков. Если продукт «живой», он нуждается в неусыпном надзоре его непосредственных создателей. «K15» полностью соответствует этому принципу. Более того, постоянно собирается обратная связь от клиентов, разрабатывающих

свои проекты на «K15», а также конечных потребителей, эксплуатирующих эти изделия у себя на объектах. Затем, путем коллективного анализа пожеланий наиболее критические из них сразу же передаются разработчикам. Менее важные — перманентно внедряются в последующих релизах. Это касается как программной части, так и аппаратного оснащения. Например, увеличение скорости работы FRAM, асинхронное чтение регистров Modbus, реализация пользовательских протоколов и многое другое — это то, что хотели видеть потребители в контроллерах.

В случае неполадок важна оперативная реакция и своевременное их устранение. Любая неполадка или возникший вопрос обрабатываются в течение 24 ч квалифицированными инженерами, которые оперативно проконсультируют клиента и предложат варианты решения проблемы. Если клиент столкнулся с трудностями создания проектов, программисты вникнут в проблематику и поделятся своим опытом решения.

*Гришин Дмитрий Анатольевич — инженер-разработчик ООО «Эй энд Ти Текнолоджис».*  
*E-mail: grishin@a-t-tech.ru info@custom-eng.ru*  
*Контактный телефон 8 (800) 775-74-70.*

Контроллеры «K15» реализуют не только классическую модель ПЛК, но также альтернативный подход к реализации модульных систем. Перекрывая большинство задач, которые решают обычные модульные ПЛК, данная линейка дает разработчику нечто большее, позволяет вывести проекты на качественно иной уровень.

Линейка «K15» - отечественная разработка в сфере автоматизации как промышленного, так и других сегментов, требующих надежных решений. Это постоянно развивающийся динамичный продукт, способный составить достойную конкуренцию ушедшим вендорам, является экономически эффективным по стоимости владения, эргономичным и надежным решением, отвечающим необходимым стандартам качества. Срок производства серийных моделей «K15» составляет 30 дней. Стандартный гарантийный период 24 мес. с момента поставки. Расширенный гарантийный период - до 48 мес. с момента поставки.

### Новости «Систэм Электрик»

Российская производственная компания «Систэм Электрик» (Systeme Electric, ранее Schneider Electric в России) представляет новинки.

#### *Вакуумные выключатели SystemePact VCB*

SystemePact VCB — это вакуумные выключатели номинальным напряжением 6...35 кВ. Они предназначены для защиты сетей и электроустановок среднего напряжения и могут применяться как для использования в новых распределительных устройствах с воздушной изоляцией, так и для замены отслуживших свой срок выключателей в модернизируемых распределительных устройствах.

Новая серия позволяет реализовать максимально комфортный переход от аппаратов Schneider Electric к аппаратам Systeme Electric на любом этапе жизненного цикла электроустановки — от проектирования до эксплуатации. Линейка Systeme Electric совместима с линейками выключателей HVX, EasyPact EXE, которые ранее поставляла Schneider Electric. SystemePact VCB не уступают по части качества выключателям HVX, а также имеют идентичные размеры и характеристики.

Вакуумные выключатели SystemePact VCB поставляются в готовом виде с уже установленными аксессуарами. Стандартная комплектация включает мотор-привод, а также катушки отключения и включения. Оформить заказ можно через личный кабинет.

#### *Однофазные источники бесперебойного питания (ИБП) для широкого спектра применений*

В новой линейке представлены модели Back-Save BV, Smart-Save SMT, Smart-Save Online SRV и Smart-Save Online SRT, а также аккумуляторные батареи.

Серия Back-Save BV (BVSE) предназначена для офисных и домашних применений. ИБП серии BVSE способен питать маломощные устройства, такие как модем или маршрутизатор, а также устройства высокой мощности — ПК и игровые приставки. ИБП позволяют не прерывать работу в случае перебоев электропитания умеренной продолжительности, а при более длительных — гарантируют автоматическое корректное завершение работы аппаратуры. Серия BVSE оснащена встроенным стабилизатором, который корректирует провалы и скачки напряжения, что особенно важно для регионов с плохим качеством подачи электроэнергии. Компактные размеры и

небольшой вес позволяют разместить ИБП в любых ограниченных пространствах или повесить его на стену.

Серия Smart-Save SMT (SMTSE) создана для защиты серверного, сетевого и телекоммуникационного оборудования в сравнительно несложных конфигурациях. Также подходит для магазинов розничной торговли, малых и средних предприятий. ИБП серии Smart-Save SMT имеет USB-порт и последовательный порт для интеграции ИБП и сервера с помощью программного обеспечения, позволяющего настроить автоматическое выключение сервера при низком остатке заряда батареи. Компактные размеры позволяют разместить ИБП в навесных или напольных шкафах небольшой глубины.

Серия Smart-Save Online SRV (SRVSE) предназначена для серверного оборудования. ИБП серии SRVSE обеспечивает оптимальное питание даже при нестабильной работе электросети — нарушениях подачи электроэнергии, скачках напряжения и тока, колебаниях напряжения в электросети и крупных сбоях энергосистемы. Устройства SRVSE обладают конвертируемым фактором для напольной установки или для установки в серверные стойки.

Серия Smart-Save Online SRT (SRTSE) — это производственные ИБП, разработанные для самых сложных условий электропитания, с возможностью наращивания времени автономной работы за счет подключения дополнительных аккумуляторных батарей и функции «горячей» замены. Они обеспечивают высокоэффективную защиту питания для серверных залов, а также для сетей голосовой связи и передачи данных. ИБП SRTSE подходят для залов с ограниченным пространством: при мощности до 10 кВА универсальный корпус позволяет осуществлять напольную установку или монтаж в стойку 19", занимая всего 2U...5U в базовой комплектации. Модели SH имеют небольшую глубину, что позволяет разместить ИБП в небольших шкафах.

Для серий SRVSE и STRSE доступны внутренние и внешние батарейные блоки SERBC и BPSE. В батарейных блоках используются свинцово-кислотные аккумуляторы, которые имеют высокий уровень защиты от утечек. Они имеют полную совместимость с интеллектуальной системой управления аккумуляторами, а также все необходимые сертификаты безопасности.

<https://systeme.ru>