КАСТОМ

# МОДУЛЬНЫЕ ПРОГРАММИРУЕМЫЕ КОНТРОЛЛЕРЫ К15 РОССИЙСКОГО ПРОИЗВОДСТВА

**ДМИТРИЙ ГРИШИН**

В статье рассмотрим серию К15 программируемых контроллеров производства ООО «Эй энд Ти Текнолоджис». Это модульные контроллеры, состоящие из центрального процессорного устройства (ЦПУ) и модулей ввода/вывода и подходящие для решения широкого круга задач. Кроме того, одно из их важных преимуществ — разработка и производство в России.

Модульные контроллеры К15 реализуют классическую структуру ПЛК: ЦПУ и несколько модулей ввода/вывода. Модульность, в отличие от моноблочных вариаций или смешанных решений, позволяет создавать гибкие, масштабируемые локальные системы управления именно под те задачи, которые необходимо решить в данный момент. Благодаря этому можно не переплачивать за лишнее «железо» и в то же время иметь возможность практически бесшовно (просто докупив необходимые модули) расширить и усложнить систему.

Еще один несомненный плюс модульного решения — простота и дешевизна эксплуатации. Не нужно менять дорогостоящий ПЛК из-за пары каналов ввода/вывода, вышедших из строя, — только неисправный модуль.

Серия К15 отличается от других подобных решений неплохой эргономичностью за счет классического крепления на DIN-рейку и малой ширины модулей. Все модули, включая ЦПУ, имеют один форм-фактор, что существенно упрощает компоновку при проектировании шкафа. Также следует отметить удобное расположение интерфейсной шины, соединяющей модули с ЦПУ: она уложена непосредственно

**РИС. 1.** ▼
ЦПУ К15.MCU.F1


**ТАБЛИЦА 1. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ ПРОЦЕССОРНОГО МОДУЛЯ К15.MCU.F1**

| | |
|---|---|
| **Основные характеристики** | |
| Центральный процессор | ARM 32-бит Cortex-M3, 72 МГц |
| Часы реального времени | ✓ |
| Возможность подключения дополнительных модулей ввода/вывода | ✓ |
| **Электрические характеристики** | |
| Напряжение питания, В | 24 ± 20% |
| Потребляемая мощность, Вт, не более | 5 |
| Электрическая прочность изоляции цепей, В | 500 |
| Защита входного напряжения | Ограничение тока |
| Количество дискретных выходов, шт. | 4 |
| Количество дискретных входов, шт. | 8 |
| Количество аналоговых входов, шт. | 3 |
| **Коммуникационные характеристики** | |
| Изолированный порт RS-485 | ✓ |
| Индикатор передачи данных по RS-485 | ✓ |
| Индикаторы состояния (Status, Run, Fault) | ✓ |
| Индикаторы дискретных сигналов | ✓ |
| **Механические характеристики** | |
| Размеры, мм | 107×22,5×136 |
| Масса, г | 400 |
| Степень защиты корпуса | IP20 |
| Крепление | DIN-рейка 35 мм |
| **Условия эксплуатации** | |
| Температура, °C | –40...+60 |
| Влажность, % | 10–90 |

в DIN-рейку и зафиксирована, чтобы избежать выпадения. Это дает возможность менять модули, не разбирая «корзину» и даже не отключая питания.

## ХАРАКТЕРИСТИКИ ЦПУ

Ключевой компонент системы K15 — ЦПУ. На данный момент доступны три модели процессоров — F1, F4 и H7, и они призваны покрыть достаточно широкий спектр потребностей в сфере автоматизации.

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. Также есть возможность реализации нестандартных протоколов обмена.

F1 (рис. 1, табл. 1) имеет достаточно скромные характеристики и подойдет для несложных задач.

F4 (рис. 2, табл. 2) — более мощный процессор со встроенным Ethernet-интерфейсом. Помимо увеличенных тактовой частоты, RAM и Flash, это ЦПУ отличает веб-интерфейс. Он позволяет следить за состоянием и составом «корзины», производить ее мониторинг и диагностику в реальном времени. Также веб-интерфейс дает возможность загружать проект в ЦПУ без применения программатора и обновлять программное обеспечение подключенных модулей ввода/вывода.

H7 (рис. 3, табл. 3) это наиболее производительное ЦПУ из представленных. Благодаря неплохим техническим характеристикам, а также возможности применения внешнего Flash-накопителя в виде SD-карты, он способен выполнять достаточно сложные алгоритмы, причем достаточно быстро. Предусмотрено и применение FRAM вместо EEPROM в качестве энергонезависимой памяти. Веб-интерфейс тоже присутствует.

Все ЦПУ работают под управлением операционной системы реального времени, что повышает надежность и быстродействие контроллеров и позволяет использовать аппаратные возможности процессоров на 100%.

## МОДУЛИ ВВОДА/ВЫВОДА

В линейке K15 широкий набор модулей: от простых дискретных модулей ввода/вывода до гальванически развязанных аналоговых модулей вывода, счетных модулей, PWM-модулей и т. д. Комбинируя их в «корзине», можно создавать самые разные системы — совсем без модулей, с одним модулем либо сложные вариации, включающие до 8 модулей на одно ЦПУ.

Ключевая особенность всех этих модулей — механизм взаимодействия с ЦПУ. Как было сказано выше, между ЦПУ и модулями предусмотрена интерфейсная системная шина. Она выполнена на базе CAN-интерфейса и физически представляет собой 5-контактную шину T-Bus, на которую устанавливаются ЦПУ и модули в процессе монтажа на DIN-рейку. Поскольку шина расположена в задней части корпуса модулей со стороны рейки, есть возможность без труда снимать и устанавливать модули без демонтажа всей «корзины». Однако главное — то, что модули ввода/вывода можно

**ТАБЛИЦА 2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ ПРОЦЕССОРНОГО МОДУЛЯ K15.CPU.F4**

| | |
|---|---|
| **Основные характеристики** | |
| Центральный процессор | ARM 32-бит Cortex-M4, 168 МГц |
| Веб-интерфейс | ✓ |
| Часы реального времени | ✓ |
| Возможность подключения дополнительных модулей ввода/вывода | ✓ |
| **Электрические характеристики** | |
| Напряжение питания, В | 24 ±20% |
| Потребляемая мощность, Вт, не более | 5 |
| Электрическая прочность изоляции цепей, В | 500 |
| Защита входного напряжения | Ограничение тока |
| Количество дискретных выходов, шт. | 2 |
| Количество дискретных входов, шт. | 3 |
| **Коммуникационные характеристики** | |
| Порт Ethernet 10/100 Base-T, шт. | 1 |
| Количество изолированных портов RS-485, шт. | 1 |
| Количество неизолированных портов RS-485, шт. | 2 |
| Поддерживаемые протоколы обмена | Modbus RTU/TCP |
| Интерфейс обмена данными с модулями | CAN |
| Индикатор передачи данных по RS-485, шт. | 3 |
| Индикаторы состояния (Status, Run, Fault) | ✓ |
| Индикаторы дискретных сигналов | ✓ |
| **Механические характеристики** | |
| Размеры, мм | 107×22,5×136 |
| Масса, г | 400 |
| Степень защиты корпуса | IP20 |
| Крепление | DIN-рейка 35 мм |
| **Условия эксплуатации** | |
| Температура, °C | –40...+60 |
| Влажность, % | 10–90 |


**РИС. 2. ▼**
ЦПУ K15.CPU.F4


**РИС. 3.** ▲
ЦПУ K15.CPU.H7

**ТАБЛИЦА 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ ПРОЦЕССОРНОГО МОДУЛЯ K15.CPU.H7**

| | |
|---|---|
| **Основные характеристики** | |
| Центральный процессор | ARM 32-бит Cortex-M7, 480 МГц |
| Веб-интерфейс | ✓ |
| Поддержка MicroSD | ✓ |
| Часы реального времени | ✓ |
| Возможность подключения дополнительных модулей ввода/вывода | ✓ |
| **Электрические характеристики** | |
| Напряжение питания, В | 24 ±20% |
| Потребляемая мощность, Вт, не более | 5 |
| Электрическая прочность изоляции цепей, В | 500 |
| Защита входного напряжения | Ограничение тока |
| Количество дискретных выходов, шт. | 2 |
| Количество дискретных входов, шт. | 3 |
| **Коммуникационные характеристики** | |
| Порт Ethernet 10/100 Base-T, шт. | 1 |
| Количество изолированных портов RS-485, шт. | 2 |
| Количество неизолированных портов RS-485, шт. | 1 |
| Поддерживаемые протоколы обмена | Modbus RTU/TCP |
| Интерфейс обмена данными с модулями | CAN |
| Индикатор передачи данных по RS-485, шт. | 3 |
| Индикаторы состояния (Status, Run, Fault) | ✓ |
| Индикаторы дискретных сигналов | ✓ |
| **Механические характеристики** | |
| Размеры, мм | 107×22,5×136 |
| Масса, г | 400 |
| Степень защиты корпуса | IP20 |
| Крепление | DIN-рейка 35 мм |
| **Условия эксплуатации** | |
| Температура, °C | –40...+60 |
| Влажность, % | 10–90 |

заменять даже в работающей системе, не отключая питания (так называемая горячая замена). Это бывает критично важно там, где отключение и перезапуск локальной системы управления связаны с нарушением технологического процесса или серьезными производственными издержками.

В ближайшем будущем планируется выпуск специальной серии модулей ввода/вывода K15, оснащенных интерфейсом RS-485, на котором будет реализован популярный протокол Modbus. Это позволит произвести помодульное дооснащение имеющихся локальных и распределенных систем управления без применения ЦПУ K15 в тех проектах, где уже есть ЦПУ, но не хватает каналов ввода/вывода.

## ОСОБЕННОСТИ ПРОГРАММИРОВАНИЯ

При создании проектов по управлению для подобных систем часто используются классические языки МЭК, которые позволяют обходиться без глубокого знания программирования, однако для K15 предусмотрено программирование на широко распространенных языках высокого уровня C/C++. Это непопулярный метод работы с модульными системами в сфере автоматизации, хотя C, C++, C#, VBA применяются в качестве скриптовых языков в контроллерах Siemens, B&R, Allen Bradley, Schneider и др.

Какие преимущества дает создание проектов таким способом?

C и C++ очень похожи на один из языков МЭК — ST (Structured Text): в них, по сути, те же переменные, циклы, условия, переходы. Однако если ST позволяет реализовать лишь то, что заложено в Runtime-оболочке контроллера, которая является как бы исполняемой средой для проекта, то язык C позволяет выйти за рамки этой среды. Нам доступен весь синтаксический инструментарий этого орудия программирования. Кроме того, многие алгоритмические приемы на C/C++ реализуются проще и понятнее, а создание структур, приведение типов, callback-функции, перегруженные функции и прочие приемы программирования становятся приятным бонусом использования этого языка.

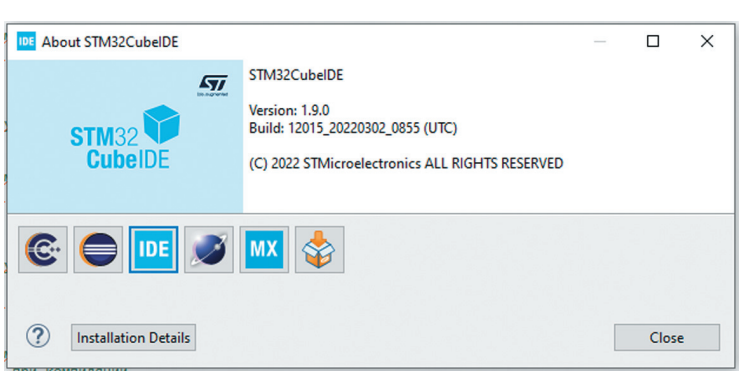


**РИС. 4.** ▶
Краткая информация о среде CubeIDE

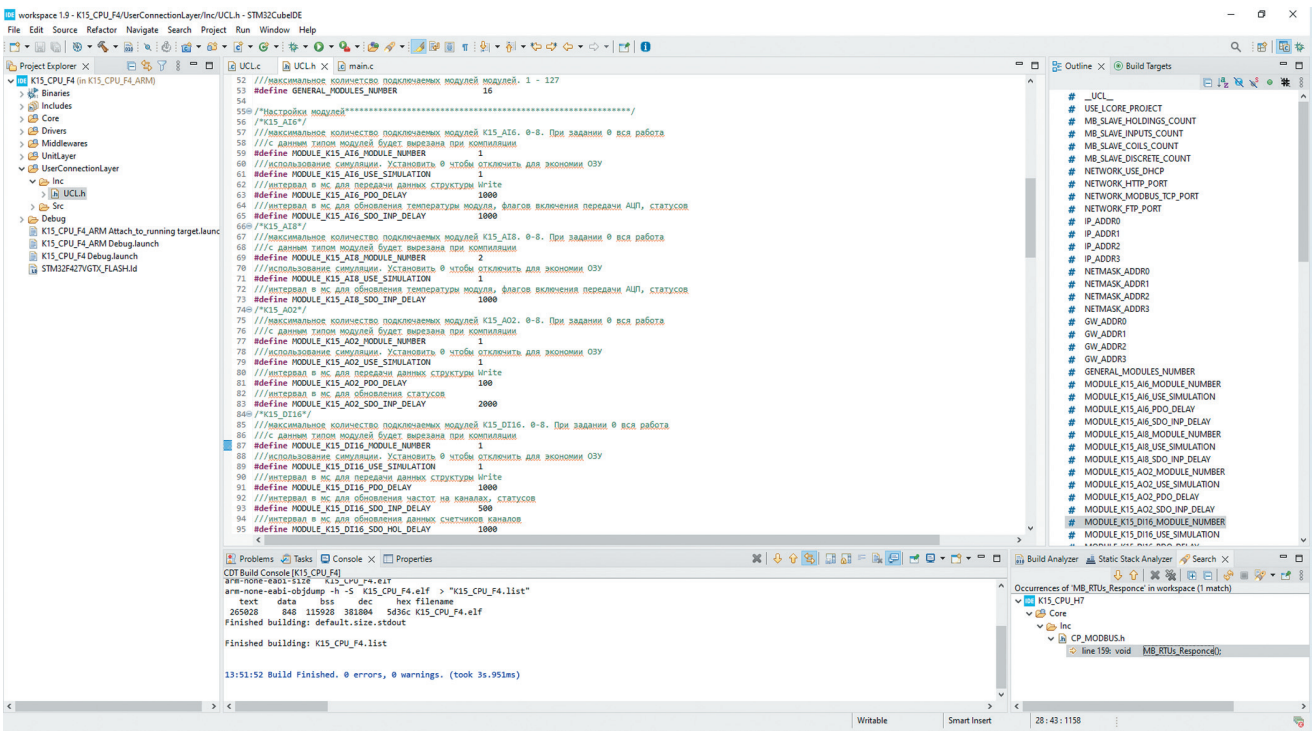

**РИС. 5.** ▲
Интерфейс CubeIDE

Немаловажную роль играет и инструментарий разработки. Далеко не все среды разработки проектов под те или иные модульные системы понятны и легки в освоении. Например, описание работы со средой TIA Portal занимает более 1000 страниц руководства программиста. Не менее сложна и Studio 5000 для контроллеров Allen Bradley серий Control и CompactLogix. А чтобы уверенно работать в CodeSys, нужен не один месяц обучения.

Семейство K15 программируется в среде разработки CubeIDE (рис. 4) — это официальная свободно распространяемая IDE от компании STMicroelectronics, процессоры которой и являются главным элементом контроллеров на данный момент. В комплекте с контроллерами идут стартовый проект и подробная инструкция. Для начала нужно открыть проект в CubeIDE, скомпилировать его и «отправить» в контроллер (рис. 5). Система уже будет работать, отображать веб-интерфейс, опрашивать модули. Затем, используя описанные структуры и функции, можно заняться решением конкретных задач.

Еще одно преимущество среды CubeIDE — доступность. Лицензии часто требуют немалых затрат, а CodeSys, Beremiz, OwenLogic и прочие системы, не требующие приобретения лицензий, поддерживаются далеко не всеми модульными системами. Как правило, каждый производитель старается разработать свой инструментарий как платформозависимый программный продукт либо свою экосистему (примеры — DeltaV, TIA Portal, FactoryTalk). До недавнего времени все эти программные продукты можно было хотя бы свободно приобрести, но, учитывая уход с рынка многих зарубежных игроков, сделать это уже затруднительно. Среда CubeIDE не требует лишних затрат на приобретение лицензий.

Стоит обратить внимание и на процесс отладки проекта, непосредственно влияющий на скорость разработки. Одна из ключевых особенностей языков МЭК — механизм онлайн-трассировки и отладки проекта. K15 также обладает этим функционалом, изначально заложенным в самой IDE. Однако из-за языка высокого уровня отладка проекта в CubeIDE более многосторонняя и глубокая: доступны классический вывод текущих переменных, точки останова, принудительная запись значений, пошаговое исполнение кода. Также есть возможность вернуться на шаг назад либо остановить исполнение кода по условию. Дополнительным удобством для разработчика можно считать контекстную подсветку значений переменных при наведении курсора.

Подводя итог: благодаря использованию языков C/C++ разработка выполняется быстрее и качественнее, а дальнейшее сопровождение и рефакторинг проекта становятся дешевле для конечного потребителя, поэтому, на наш взгляд, стоит потратить силы и время на их изучение.

## ПРИМЕР РЕАЛИЗАЦИИ ЗАДАЧИ

Рассмотрим, как решаются прикладные задачи средствами разработки K15, на конкретном примере. Допустим, у нас есть факельная установка с электроискровым розжигом дежурной горелки и фотодатчиком наличия пламени. Наша 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 будут сделаны три попытки розжига. Также алгоритм должен обеспечивать автоматический перерозжиг горелки в случае погасания пламени.

**РИС. 6.** ▶ Создание переменных и функциональных блоков на языке FBD

```
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
```

Сначала попытаемся реализовать задуманное средствами Codesys, а именно языка FBD. Создадим необходимые переменные и функциональные блоки (рис. 6), при этом учитывая следующие обозначения:

- start_DI — сигнал пуска системы (например, кнопка);
- 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. 7). Рассмотрим ее по порядку действий. При подаче сигнала пуска через детектор переднего фронта срабатывает триггер rs1, принимая логическое состояние 1.

**РИС. 7.** ▼ Создание функциональной схемы, реализующей алгоритм, на языке FBD

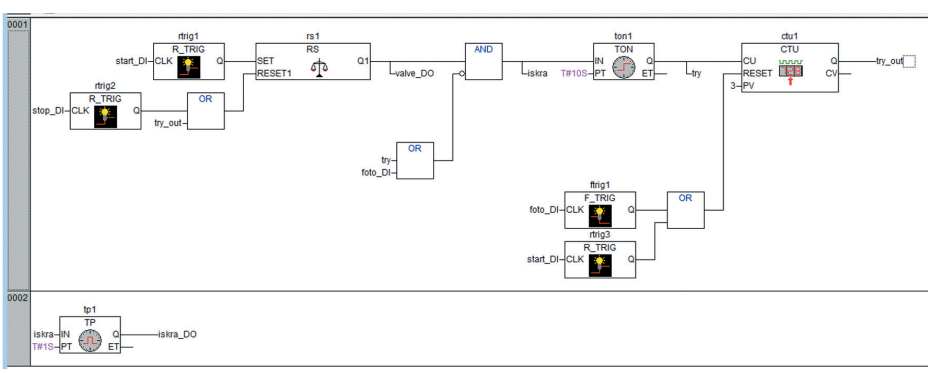


**РИС. 8.** ▶ Объявление переменных в файле UCL.c

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

uint16_t state, count, try_count;
```

**РИС. 9.** ▼ Код в CubeIDE для решения задачи

```
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
```

Открывается клапан, затем сигнал следует на блок AND. Поскольку флаг совершения попытки розжига try и сигнал с фотодатчика имеют логический 0, 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 с, выставляется флаг совершения 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. Следует иметь в виду, 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 ту же задачу в рамках программного функционала K15. Весь код размещаем в файле UCL.c. Здесь также не обойтись без объявления переменных (рис. 8).

Для единообразия сигналы имеют схожее обозначение. Дополнительно добавились переменные state (стадия работы системы), count (аккумулятор счетчика времени) и try_count (аккумулятор числа попыток розжига).

Обратимся к самому коду (рис. 9). 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 можно увеличить — например, добавить стадию АВАРИЯ.

В целом, конечно, проект небольшой, задача тривиальна, но даже в таком сравнении можно заметить хороший потенциал использования K15 в плане программной реализации.

## ЗАКЛЮЧЕНИЕ

Контроллеры K15 предлагают альтернативный подход к реализации модульных систем и предназначены для выполнения множества разных задач. Производитель планирует продолжать совершенствовать этот продукт, подстраиваясь под потребности пользователей. ⬣

*Дополнительную информацию можно получить по e-mail: support@custom-eng.ru*

*По вопросам сотрудничества можно обратиться по тел. 8 (800) 775-74-70 или по e-mail: info@custom-eng.ru*