

Протокол управления роботизированным лафетным стволом

Материал из wiki

Настоящий протокол управления предназначен для организации взаимодействия между управляющим устройством с одной стороны и программным обеспечением (далее ПО) лафетного ствола с дистанционным управлением или пожарного робота на его основе (далее ЛС).

Описываемый протокол является протоколом прикладного уровня и описывает порядок взаимодействия управляющего устройства с ЛС посредством запросов и ответов. На канальном уровне передача данных возможно посредством различных стандартов связи, в том числе:

- по магистрали управления RS485 с использованием протокола канального уровня;
- по сетям Ethernet с использованием транспортных протоколов стека TCP/IP;
- по радиоканалу с использованием протоколов передачи радиомодемов.

Содержание

- 1 Общие сведения
- 2 Формат заголовка запроса/ответа
- 3 Перечень поддерживаемых запросов
 - 3.1 Move. Перемещение заданного узла в заданную точку
 - 3.2 Stop. Останов работы подсистемы
 - 3.3 GetStatus. Запрос состояния подсистемы
 - 3.3.1 Общие флаги состояния
 - 3.3.2 Флаги состояния General
 - 3.3.3 Флаги состояния Climatics
 - 3.3.4 Флаги состояния Vertical, Horizontal, Nozzle
 - 3.3.5 Флаги состояния Valve1, Valve2
 - 3.3.6 Флаги состояния Control
 - 3.3.7 Флаги состояния Detector
 - 3.3.8 Флаги состояния Deployer
 - 3.3.9 Флаги состояния ExternalConn, Radio
 - 3.3.10 Флаги состояния Buttons, ExtButtons
 - 3.4 GetParam. Запрос значения параметра подсистемы
 - 3.5 SetParam. Установка значения параметра подсистемы
 - 3.6 SetupCorrectionTable. Запись таблицы коррекции угла подъема от давления.
 - 3.7 RetrieveLimits. Запрос пределов перемещения.
 - 3.8 Open. Открытие затвора.
 - 3.9 Close. Закрытие затвора.
 - 3.10 Deploy. Развертывание ствола в рабочее положение
 - 3.11 Wrap. Сворачивание ствола в дежурное положение
 - 3.12 Restart. Перезапуск подсистемы (программы)
 - 3.13 StartJustify. Запуск юстировки системы ствол - детектор очага загорания

- 3.14 StartQuench. Старт тушения
- 3.15 StartTrajectory. Программа тушения по траекториям
- 3.16 StartSeek. Старт поиска очага загорания
- 3.17 SwitchLimits. Включение/выключение контроля пределов перемещений ствола
- 3.18 Lockout. Включить/выключить аварийную блокировку ствола
- 3.19 CleanFlash. Очистить ЭНОЗУ.
- 3.20 GetCrashData. Получить информацию об аппаратном отказе
- 3.21 GetParam Hotbeds. Получение сведений об обнаруженных очагах

Общие сведения

С точки зрения протокола управления, лафетный ствол (далее - ЛС) представляет собой набор независимых подсистем. Подсистемы ЛС имеют общий адрес на магистрали управления (адрес ЛС) и собственный идентификатор внутри ЛС. Числовые значения идентификаторов подсистем вычисляются следующим образом:

$$Id = \text{Conv8}(\text{FNV1A}(\text{«Обозначение подсистемы»})), \text{ где}$$

- "Обозначение подсистемы" - строковое представление имени подсистемы, например "All".
Использование строки для вычисления идентификатора позволяет явным образом связать его с именем подсистемы, что упрощает создание вспомогательного ПО для отправки запросов.
- FNV1A — модифицированная FNV-хэш функция, применяемая к строковому представлению обозначения подсистемы;
- Conv8 — функция свертки 32хбитного значения в байт:

$$\text{Conv8}(w) = (\text{shr}(w, 24) \text{ xor } \text{shr}(w, 16) \text{ xor } \text{shr}(w, 8) \text{ xor } w) \text{ and } 0xFF$$

В таблице перечислены обозначения подсистем ЛС и их идентификаторы.

Обозначение подсистемы	Значение идентификатора	Примечание
All	0x5D	Специальный идентификатор, обозначающий все подсистемы, которые могут обработать данный запрос
Motors	0xAB	Специальный идентификатор, обозначающий все подсистемы-приводы, которые могут обработать данный запрос
General	0xC2	Главная подсистема ЛС, отвечающая за его конфигурирование, обновление ПО и контроль основных параметров: напряжение питания, давление и т.д. Также призвана выдавать ответы на запросы, которые не могут быть обработаны по причине того, что отсутствует их получатель.
Climatics	0x62	Подсистема контроля климатических условий. Отвечает за измерение климатических параметров узлов ЛС (температуры, влажности и т.п.) и за поддержание их на заданном уровне.
Vertical	0x0D	Привод вертикального перемещения.
Horizontal	0x85	Привод горизонтального перемещения.
Nozzle	0x98	Привод насадка
Valve1	0xA6	Затвор 1 (вода)
Valve2	0xC7	Затвор 2 (пена)
Control	0x45	Подсистема согласованного управления, отвечает за работу по различным программам тушения.
Detector	0xC8	Подсистема управления датчиком обнаружения очага загорания. Отвечает за поиск очага загорания.
Deployer	0xA2	Подсистема развертывания/свертывания ствола. Отвечает за перевод ствола из дежурного положения в рабочее и обратно. Физически может представлять собой поворотный узел, блок подъема, может также быть и абстрактной, например выполнять какие-то действия при запуске ЛС.
ExternalConn	0x09	Внешний интерфейс связи, проводной.

Radio	0x8E	Внешний интерфейс связи, радио.
Buttons	0xD4	Кнопочный пост управления.
ExtButtons	0x52	Удаленный кнопочный пост управления.

Каждой подсистеме может быть послан запрос. После обработки запроса подсистема высылает ответ. Запросы также могут посылаются на обработку сразу нескольким подсистемам ЛС одновременно, если в качестве получателя запроса используется идентификатор All. В таком случае, ответ содержит ответы всех подсистем ЛС, которые в состоянии ответить на данный запрос (например: запрос состояния).

Логически подсистемы ЛС выстраиваются в цепочку обработчиков запросов. При этом, если какой-либо запрос одной подсистеме, противоречит текущему состоянию другой подсистемы, находящейся в начале цепочки, эта подсистема может перехватить запрос и проинформировать об ошибке отправителя (например, нельзя управлять стволом, пока он находится в свернутом состоянии). Если какая-либо подсистема отсутствует в составе ствола (не включена в цепочку обработчиков запросов), ответы на запросы, направленные ей, высылает подсистема General, сообщая отправителю о том, что получатель не существует (например, при попытке выполнения команды развернуть ЛС, когда в его составе нет поворотного узла). Внимание, особенностью схемы управления считается то, что некоторые операции по механической работе ствола в свернутом положении являются допустимыми, например StartSeek и StartQuench (в отличие от Open, Move, StarJustify) и не вызывают ошибок поскольку автоматически разворачивают ЛС.

Также General высылает ответы на кадры с не распознанным полем идентификатором запроса. Таким образом, ни один кадр, направленный в адрес ЛС не остается без ответа, если он был принят без ошибок.

При получении нескольких запросов, либо широковещательного запроса (идентификатор получателя = "All"), запросы обрабатываются последовательно в порядке их следования в кадре протокола более низкого уровня, либо в случае широковещательного запроса в порядке перечисления подсистем ЛС в цепочке подсистем, до тех пор, пока в выходном буфере есть место для сохранения ответов. Если в выходном буфере заканчивается место, запросы, которые еще не были выполнены, отклоняются, а подсистема General добавляет ответ с кодом NoRoomForAnswer (*проверка показала что ответ выдает не только General но и тот модуль которому не хватило места [прим. IvanZ]*). При получении данного ответа клиент должен определить, какие запросы не были обработаны, и отправить их повторно.

Включение подсистем в цепочку и их типы определяются конфигурацией ЛС. Порядок, в котором подсистемы в случае их наличия встраиваются в цепочку, соответствует порядку их перечисления в таблице (подсистема General всегда оказывается в конце цепочки).

Формат заголовка запроса/ответа

Ниже представлено описание заголовка запрос/ответа на языке Си

```

-----
struct header {
|   uint8_t request_id; // Идентификатор запроса.
|   uint8_t device_id;  // Идентификатор подсистемы-получателя запроса.
|   uint8_t status;     // Статус обработки запроса (присылке запроса это поле оставляется пустым).
|   uint8_t data_size;  // Размер данных запроса
|   uint8_t data[];    // Данные запроса (DATA_SZ байт)
|};
-----

```

Идентификатор запроса - уникальный числовой идентификатор, значение которого вычисляется из наименования запроса так же, как значение идентификатора подсистемы вычисляется из ее наименования:

Id = Conv8(FNV1A(«Наименование запроса»))

Поле статуса обработки запроса имеет смысл только для ответа на запрос и заполняется ЛС. Оно содержит код статуса исполнения соответствующего запроса. В таблице приведены возможные результаты исполнения запроса.

Обозначение	Значение	Описание
Ok	0	Запрос исполнен полностью.
Accepted	1	Запрос принят к исполнению: этот статус возвращается для операций, требующих для их исполнения некоторого времени, например сохранение параметра или обращение к внутреннему устройству ствола. Данный статус сигнализирует о том, что ПО ЛС приняло запрос и помешать его выполнению может только неисправность или другой форс-мажор (выключение питания, сброс программы и т.п.). В большинстве случаев может быть признан эквивалентным статусу Ok.
NeedConfirm	2	Для исполнения запроса необходимо прислать подтверждение.
Denied	3	Запрос отклонен, т.к. одна из подсистем ствола находится в таком состоянии, что указанный запрос не может быть выполнен, например: открытие затвора в дежурном положении может привести к разрушительным последствиям для ствола, поэтому подсистема свертывания/развертывания запрещает выполнение этого запроса, находясь в этом состоянии.
ModuleNotExist	4	Подсистема, указанная в качестве получателя запроса, отсутствует в составе ствола.
ModuleFault	5	Подсистема, указанная в качестве получателя запроса, неисправна и не может выполнить его.
Busy	6	Подсистема, указанная в качестве получателя запроса, занята, поэтому запрос должен быть повторен через некоторое время.
WrongRequest	7	Неверный идентификатор запроса (не поддерживаемый запрос).
WrongData	8	Неверный объем данных запроса.
NoRoom	9	В выходном буфере не нашлось достаточно места для помещения в него результата выполнения запроса. В реальной работе этот ответ не должен возникать: необходимо составлять запросы таким образом, чтобы ответ на них гарантированно помещался в выходной буфер ствола.
InvalidId	10	Неверный идентификатор (параметра). Возвращается в ответ на запросы чтения/установки параметров, если не найден параметр указанной подсистемы с указанным идентификатором.
InvalidValue	11	Неверное значение параметра.
AccessDenied	12	Доступ к параметру запрещен.
SystemFault	13	Системная ошибка. В программе возникла ошибка во время обработки запроса, в реальной работе этот код не должен возникать.
IncompleteConfig	14	При попытке записи параметра выяснилось, что один или несколько других параметров подсистемы имеют невалидные значения: параметры не будут записаны в ЭНОЗУ ЛС, пока они все не будут иметь валидные значения.
NotCached	15	При попытке прочитать параметр подсистемы, выяснилось, что для этого необходимо обращение к удаленному устройству (например к блоку управления затвором), которое не может быть выполнено в данный момент. Необходимо подождать некоторое время и повторить запрос позже.
Cached	16	При попытке записать параметр подсистемы, выяснилось, что для этого необходимо обращение к удаленному устройству (например к блоку управления затвором), которое не может быть выполнено в данный момент. Необходимо подождать некоторое время и повторить запрос позже.
Unimplemented	17	Указанный запрос не реализован в данной подсистеме. В реальной работе такой код не должен возникать, т.к. он свидетельствует о недоработках программы.
NotSupported	18	Указанный запрос не поддерживается заданной подсистемой, например привод с датчиками конечных положений не поддерживает запрос включения/выключения контроля пределов перемещений.

Перечень поддерживаемых запросов

Ниже приведена сводная таблица поддерживаемых запросов. Далее в этом разделе приведено подробное описание каждого запроса: возможные получатели, структура данных запроса и ответа на него в зависимости от подсистемы-получателя (приводятся на языке Си), а также алгоритм действий той или иной подсистемы при приеме данного типа запроса.

Наименование	Код	Назначение
Move	0x4D	Перемещение заданного исполнительного узла в заданную точку. В качестве адресата могут выступать: Horizontal Vertical, Nozzle и Deployer.
Stop	0x46	Остановка операций, выполняемых заданной подсистемой. В качестве адресата могут выступать: Horizontal Vertical, Nozzle, Deployer, Detector, Control, Valve1, Valve2 и All.
GetStatus	0xD4	Запрос состояния заданной подсистемы. В качестве получателя могут выступать любые подсистемы.
GetParam	0x9D	Запрос параметра заданной подсистемы. В качестве получателя могут выступать любые подсистемы.
SetParam	0xF1	Установка параметра заданной подсистемы. В качестве получателя могут выступать любые подсистемы.
SetupCorrectionTable	0x86	Установка таблицы коррекции угла возвышения. В качестве получателя может выступать Control.
RetrieveLimits	0x80	Запрос пределов перемещения. В качестве получателя могут выступать All, Horizontal Vertical, Nozzle и Deployer.
Open	0x21	Открыть затвор. В качестве получателя могут выступать Valve1, Valve2.
Close	0x24	Закрыть затвор. В качестве получателя могут выступать Valve1, Valve2.

Deploy	0x7E	Развернуть ствол в рабочее положение. В качестве получателя может выступать Deployer.
Wrap	0x8D	Свернуть ствол в дежурное положение. В качестве получателя может выступать Deployer.
Restart	0x0A	Сбросить программу. В качестве получателя могут выступать All, General.
StartJustify	0x55	Запустить юстировку системы ствол - датчик обнаружения очага загорания. В качестве получателя может выступать Detector.
StartQuench	0xC5	Запустить программу тушения очага загорания. В качестве получателя может выступать Control.
StartSeek	0x76	Запустить поиск очага загорания. В качестве получателя может выступать Detector.
SwitchLimits	0xD9	Отключить контроль пределов перемещений ствола. All, Horizontal Vertical, Nozzle и Deployer.
GetHotbed	0x9C	Получить информацию об обнаруженных очагах загорания. Detector.--Demitar (обсуждение) 06:08, 14 октября 2015 (UTC) На данный момент не реализовано, но скоро будет // По состоянию на май 2016 все еще не реализовано, Ошибочный id запроса
Lockout	0xE9	Включить/выключить аварийную блокировку ствола. В качестве получателя может выступать All
CleanFlash	0x16	Очистка ЭНОЗУ. Только для General
GetCrashData	0x06	Заполучить структуру fault_info

Move. Перемещение заданного узла в заданную точку

Идентификатор	0x4D
Адресаты	Horizontal, Vertical, Nozzle, Deployer
Обработчики	Horizontal, Vertical, Nozzle, Deployer, Control, Detector, ExternalConn, Radio
Формат поля данных запроса	<pre>{ "data": { "destination": // значение координаты точки, к которому необходимо начать движение (в угловых минутах) }, "attribute": { "packed": 1 } }</pre>
Формат поля данных ответа	Ответ не содержит данных

Исполнение запроса вызывает начало работы заданного исполнительного узла.

Возможны следующие сценарии исполнения запроса:

1. ЛС выполняет задачу развертывания или свертывания или находится в дежурном положении. В этом случае Deployer возвращает код ошибки Denied. ЛС продолжает выполнение операции свертывания/развертывания или продолжает находиться в дежурном положении.
2. ЛС выполняет задачу поиска очага загорания. В этом случае Detector возвращает код ошибки Denied. ЛС продолжает поиск очага загорания.
3. ЛС осуществляет выполнение программы тушения. В этом случае команда исполняется адресатом, а выполнение программы тушения приостанавливается до того момента, пока не пройдет 30 секунд со времени последней команды управления приводами, после чего программа тушения возобновится относительно той точки, в которую он будет перемещен этими командами.
4. При получении адресатом этого запроса, соответствующий исполнительный узел начинает движение в указанную точку и возвращает статус Ok или Accepted. Если подсистема неисправна, возвращается ModuleFault.
5. Запрос направлен в адрес подсистемы, для которой его исполнение не предусмотрено, тогда последняя возвращает статус WrongRequest.
6. Запрос направлен в адрес подсистемы, которой нет в составе ЛС, тогда General отправляет ответ ModuleNotExist от имени этой подсистемы.

Важно: т.к. нет отдельных запросов для движения ПР/ЛС влево/вправо/вверх/вниз и т.д. эти команды реализуются с помощью указанного запроса. Для этого параметр desination должен иметь либо очень большое положительное значений для движения вверх/влево, либо очень большое по модулю отрицательное значение для движения вниз/вправо (например, 21600 и - 21600).

При получении координаты привод в режиме контроля пределов всегда двигается до предела. В режиме без контроля пределов он будет двигаться к указанной точке.

Stop. Останов работы подсистемы

Идентификатор	0x46
Адресаты	All, Valve1, Valve2, Horizontal, Vertical, Nozzle, Deployer, Detector, Control
Обработчики	Valve1, Valve2, Horizontal, Vertical, Nozzle, Deployer, Detector, Control, ExternalConn, Radio
Формат поля данных запроса	Запрос не содержит данных
Формат поля данных ответа	Ответ не содержит данных

Исполнение запроса вызывает остановку текущей операции заданной подсистемы.

Возможны следующие сценарии исполнения запроса:

1. Если в качестве получателя указан All, все операции, связанные с перемещением ЛС останавливаются, в т.ч. останавливаются затворы, если они закрывались/открывались. Все поддерживающие данный запрос подсистемы высылают ответ Ok или Accepted.
2. ЛС выполняет задачу развертывания или свертывания. В этом случае, если команда направлена не в адрес Deployer, последний возвращает код ошибки Denied. Иначе процесс развертывания/свертывания отаавливается, а соответствующая подсистема высылает ответ Ok или Accepted.
3. ЛС выполняет задачу поиска очага загорания. В этом случае, если команда направлена не в адрес Detector, последний возвращает код ошибки Denied. Иначе процесс поиска очага загорания останавливается, а соответствующая подсистема высылает ответ Ok или Accepted.
4. ЛС осуществляет выполнение программы тушения. В этом случае, если команда направлена не в адрес Control, последний возвращает код ошибки Denied. Иначе процесс тушения останавливается (но затвор не закрывается, если он открыт), а соответствующая подсистема высылает ответ Ok или Accepted.
5. ЛС осуществляет перемещение в заданную точку одним из исполнительных узлов (Horizontal, Vertical, Nozzle, Valve1, Valve2), и запрос направлен в адрес этого исполнительного узла, тогда данный узел останавливается, а соответствующая подсистема высылает ответ Ok или Accepted. Если подсистема неисправна, возвращается ModuleFault.
6. Запрос направлен в адрес подсистемы, для которой его исполнение не предусмотрено, тогда последняя возвращает статус WrongRequest.
7. Запрос направлен в адрес подсистемы, которой нет в составе ЛС, тогда General отправляет ответ ModuleNotExist от имени этой подсистемы.

GetStatus. Запрос состояния подсистемы

Идентификатор	0xD4
Адресаты	любые, All
Обработчики	совпадают с адресатом
Формат поля данных запроса	Запрос не содержит данных
Формат поля данных ответа General	<pre>struct GeneralStatus { uint32_t status; // биты флагов состояния ЛС в целом uint16_t main_voltage; // напряжение питания ЛС в дВ uint16_t pressure; // давление ОТВ в 0.1 атм uint16_t flowrate; // расход ОТВ л/с #if attribute ((packed)) struct ThermostatStatus { uint32_t status; // биты флагов состояния системы контроля климатических параметров uint16_t hor temp; // температура горизонтального привода в 0.1 градус Цельсия uint16_t ver temp; // температура вертикального привода в 0.1 градус Цельсия uint16_t noz temp; // температура привода насадка в 0.1 градус Цельсия uint16_t dep temp; // температура привода системы развертывания/свертывания в 0.1 Цельсия uint16_t box temp; // температура в корпусе ЛС или шкафа управления в 0.1 градус Цельсия uint16_t humidity; // относительная влажность внутри корпус ствола или шкафа управления в % #if attribute ((packed)) struct DriveStatus { uint32_t status; // биты флагов состояния привода uint16_t position; // текущее положение привода в угловых минутах (положение выходного патрубка) или мм (положение насадка) uint8_t temperature; // текущее значение тока привода в дА uint8_t speed; // текущая заданная скорость привода в градус/с или мм/с #if attribute ((packed)) struct ValveStatus { uint32_t status; // биты флагов состояния блок управления затвором #if attribute ((packed)) struct ControlStatus { uint32_t status; // биты флагов состояния системы согласованного управления #if attribute ((packed)) struct DetectorStatus { uint32_t status; // биты флагов состояния датчика обнаружения очага загорания #if attribute ((packed))</pre>
Формат поля данных ответа Vertical, Horizontal, Nozzle	
Формат поля данных ответа Valve1, Valve2	
Формат поля данных ответа Control	
Формат поля данных ответа Detector	

Формат поля данных ответа
Deployer

Формат поля данных ответа
ExternalConn, Radio

Формат поля данных ответа
Buttons, ExtButtons

```
struct DeployerStatus_t {
    uint32_t status; // биты флагов состояния привода;
    uint16_t position; // текущее положение исполнительного узла (угловые минуты, мм);
    uint8_t amperage; // текущее значение тока привода;
    uint8_t speed; // текущая установленная скорость привода в градус/с (мм/с);
    attribute_t packed;
};

struct ConnStatus_t {
    uint32_t status; // биты флагов состояния интерфейса связи;
    attribute_t packed;
};
```

На данный момент не реализовано

Исполнение запроса вызывает отправку сведений о состоянии той или иной подсистемы. Если запрос направлен в адрес All, все подсистемы возвращают сведения о своем состоянии. Биты флагов состояния подсистем разделены две чати: первые 16 бит имеют общее для всех подсистем назначение, вторые 16 бит специфичны для каждой подсистемы.

Общие флаги состояния

Обозначение	Номер бита	Назначение
ConfigFault	0	Ошибка чтения конфигурации из ЭНОЗУ.
ConfigInvalid	1	Некоторые параметры конфигурации имеют недопустимые значения.
ConfigWriteFault	2	Ошибка записи данных в хранилище параметров.
ConnLost	3	Потеря связи с исполнительным устройством (для удаленных устройств).
BadPower	4	Напряжение питания находится вне диапазона допустимых значений.
RemotePaIncomplete	5	Результат обращения к параметрам удаленного устройства: неполные данные.
RemotePaBadId	6	Результат обращения к параметрам удаленного устройства: неверный идентификатор.
RemotePaBadVal	7	Результат обращения к параметрам удаленного устройства: неверное значение.
RemotePaError	8	Результат обращения к параметрам удаленного устройства: внутренняя ошибка.
Restarting	9	Подсистема находится в состоянии ожидания перезапуска.
InternalFault	10	Ошибка управления устройством.
WdtFault	11	Имел место быть нештатный сброс программы по сторожевом таймеру(для удаленных устройств).
ProtocolFault	12	Ошибка протокола связи с устройством.
Disabled	13	Подсистема деактивирована и не влияет на выполнение функций других подсистем.
Terminated	14	Подсистема завершила свою работу.
Reserved	15	Зарезервировано.

Флаги состояния General

Обозначение	Номер бита	Назначение
PressureSensorBreak	16	Обрыв цепи датчика давления
PressureSensorShort	17	Короткое замыкание цепи датчика давления
LockedOut	18	Включена аварийная блокировка ствола (см. запрос Lockout)

Флаги состояния Climatics

Обозначение	Номер бита	Назначение
HorizontalHeat	16	Включен подогрев горизонтального привода
VerticalHeat	17	Включен подогрев вертикального привода
NozzleHeat	18	Включен подогрев привода насадка
DeployerHeat	19	Включен подогрев привода поворотного узла
BoxHeat	20	Включен подогрев внутри корпуса ЛС или шкафа управления
HorizontalFault	21	Неисправность подогрева горизонтального привода
VerticalFault	22	Неисправность подогрева вертикального привода
NozzleFault	23	Неисправность подогрева привода насадка
DeployerFault	24	Неисправность подогрева привода поворотного узла
BoxFault	25	Неисправность подогрева в корпусе ЛС или шкафа управления
HorizontalSensorFault	26	Неисправность датчика температуры горизонтального привода.
VerticalSensorFault	27	Неисправность датчика температуры вертикального привода.
NozzleSensorFault	28	Неисправность датчика температуры насадка
DeployerSensorFault	29	Неисправность датчика температуры поворотного узла
BoxSensorFault	30	Неисправность датчика температуры в корпусе ЛС или шкафа управления
HumiditySensorFault	31	Неисправность датчика влажности

Флаги состояния Vertical, Horizontal, Nozzle

Обозначение	Номер бита	Назначение
Move	16	Привод в движении
PositionWayFault	17	Неверное подключение датчика положения

AmperageOverload	18	Перегрузка по току (этот флаг снимается автоматически через 3 секунды после остановки привода)
PosEncoderFault	19	Неисправность датчика положения
SpeedEncoderFault	20	Неисправность датчика скорости
MotorBroken	21	Неисправность цепи двигателя
SpeedWayFault	22	Неверное подключение датчика скорости
Limitless	23	Включен режим без контроля пределов перемещений
MinLimitReached	24	Достигнут нижний/правый предел (предел шире для насадка)
MaxLimitReached	25	Достигнут верхний/левый предел (предел уже для насадка)
AbsoluteMinLimitReached	26	Достигнуто ограничение, задаваемое датчиком конечных перемещений
AbsoluteMaxLimitReached	27	Достигнуто ограничение, задаваемое датчиком конечных перемещений
SpeedSlowAmp	28	ЛСД: действует режим снижения скорости привода при перегрузке
TRVEngage	29	Минибот: включен режим ТРВ насадка (new! nozzle only)

Флаги состояния Valve1, Valve2

Обозначение	Номер бита	Назначение
Opening	16	Затвор открывается
Closing	17	Затвор закрывается
AmperageOverload	18	Перегрузка по току
Blocked	19	Включена аппаратная блокировка затвора
OpenSensorFault	20	Неисправность датчика положения "Открыто"
CloseSensorFault	21	Неисправность датчика положения "Закрыто"
Open	22	Затвор открыт
Closed	23	Затвор закрыт

Флаги состояния Control

Обозначение	Номер бита	Назначение
Quench	16	Выполняется программа тушения
OutsideLimits	17	Сектор тушения полностью или частично находится вне пределов
ModeFault	18	Программа тушения не может выполняться в текущем состоянии ствола
Paused	19	Программа тушения приостановлена для корректировки оператором
TrajectoryPressureFault	20	Показания датчика давления не подходят для баллистики
TrajectoryRangeFault	21	Слишком далеко для баллистических возможностей ствола
TrajectoryAngleFault	22	Слишком большой угол подъема ствола
TrajectoryCRCFault	23	Структура с данными баллистики повреждена
Trajectory	24	Флаг активности тушения по траекториям

Примечания:

Для режима тушения по траекториям истинны следующие утверждения:

- 1) в случае ошибки приводов появится общий флаг InternalFault, тушение будет прекращено
- 2) в случае выключенной баллистики по тушения по траекториям флаги TrajectoryCRCFault, TrajectoryRangeFault, TrajectoryAngleFault никогда не появятся.
- 3) в случае наезда на пределы на соответствующей итерации появится флаг OutsideLimits, как при обычном тушении.
- 4) флаги (кроме Trajectory, Quench, TrajectoryPressureFault) очищаются только при перезапуске тушения. TrajectoryPressureFault очищается сразу при восстановлении давления, чтобы он не "залипал" на начало тушения пока насосная станция еще не успела поднять давление.
- 5) в случае выключенной баллистики либо при появлении флагов TrajectoryPressureFault, TrajectoryRangeFault, TrajectoryAngleFault - ствол тушит прямой наводкой на самой узкой струе.

Флаги состояния Detector

Обозначение	Номер бита	Назначение
Searching	16	Осуществляется поиск очага
Found	17	Обнаружен один или несколько очагов загорания
DeviceFault	18	Неисправность датчик обнаружения очага загорания
Canceled	19	Поиск очага загорания отменен
BadSector	20	Неверный сектор поиска очага загорания
Justifying	21	Осуществляется юстировка системы ствол-датчик
Justified	22	Юстировка системы ствол-датчик успешно завершена
DirtyLens	23	Загрязненное стекло (для датчиков ИКТВ) new!

Флаги состояния Deployer

Обозначение	Номер бита	Назначение
Move	16	Привод в движении
PositionWayFault	17	Неверное подключение датчика положения
AmperageOverload	18	Перегрузка по току (этот флаг снимается автоматически через 3 секунды после остановки привода)
PosEncoderFault	19	Неисправность датчика положения
SpeedEncoderFault	20	Неисправность датчика скорости
MotorBroken	21	Неисправность цепи двигателя
SpeedWayFault	22	Неверное подключение датчика скорости
Limitless	23	Включен режим без контроля пределов перемещений
MinLimitReached	24	Достигнут нижний/правый предел (предел шире для насадка)
MaxLimitReached	25	Достигнут верхний/левый предел (предел уже для насадка)
AbsoluteMinLimitReached	26	Достигнуто ограничение, задаваемое датчиком конечных перемещений
AbsoluteMaxLimitReached	27	Достигнуто ограничение, задаваемое датчиком конечных перемещений
Deployed	28	ЛС в рабочем положении
Deploying	29	ЛС разворачивается в рабочее положение
Wrapped	30	ЛС в дежурном положении
Wrapping	31	ЛС сворачивается в дежурное положение

Флаги состояния ExternalConn, Radio

Интерфейсы связи не определяют дополнительные флаги состояния. Флаг состояния ConnLost (бит 3) для данных подсистем означает отсутствие сигналов управления по заданным интерфейс связи (никто не пытался управлять ЛС по ним в течение последних 3х секунд).

Флаги состояния Buttons, ExtButtons

Обозначение	Номера битов	Назначение
LeftButtonState	16, 17	Состояние кнопки влево: 0x00 - Не нажата, 0x01 - Нажата, 0x02 - КЗ, 0x03 - обрыв
RightButtonState	18, 19	Состояние кнопки вправо: 0x00 - Не нажата, 0x01 - Нажата, 0x02 - КЗ, 0x03 - обрыв
UpButtonState	20, 21	Состояние кнопки вверх 0x00 - Не нажата, 0x01 - Нажата, 0x02 - КЗ, 0x03 - обрыв
DownButtonState	22, 23	Состояние кнопки вниз: 0x00 - Не нажата, 0x01 - Нажата, 0x02 - КЗ, 0x03 - обрыв
WideButtonState	24, 25	Состояние кнопки шире: 0x00 - Не нажата, 0x01 - Нажата, 0x02 - КЗ, 0x03 - обрыв
NarrowButtonState	26, 27	Состояние кнопки уже: 0x00 - Не нажата, 0x01 - Нажата, 0x02 - КЗ, 0x03 - обрыв
OpenButtonState	28, 29	Состояние кнопки открыть: 0x00 - Не нажата, 0x01 - Нажата, 0x02 - КЗ, 0x03 - обрыв
CloseButtonState	30, 31	Состояние кнопки закрыть: 0x00 - Не нажата, 0x01 - Нажата, 0x02 - КЗ, 0x03 - обрыв

GetParam. Запрос значения параметра подсистемы

Идентификатор	0x9D
Адресаты	все подсистемы
Обработчики	запрос обрабатывает только та подсистема, которой он направлен
Формат поля данных запроса	<pre>struct data { uint16_t id; // Идентификатор параметра. attribute_t((packed)); }; struct answer { uint16_t id; // Идентификатор параметра. uint32_t values[]; // Массив значений параметра. attribute_t((packed)); };</pre>
Формат поля данных ответа	

В ответ на запрос высылается ответ, содержащий текущее значение параметра (параметр может содержать несколько значений, если это параметр - массив), статус ответа Ok. Если параметра с заданным идентификатором нет, высылается ответ InvalidId. Если указанной подсистемы нет в составе ЛС, General высылает ответ ModuleNotExist от имени этой подсистемы.

Со списком параметров подсистем и допустимых значений можно ознакомиться в описании программы АБМИ.00098.

SetParam. Установка значения параметра подсистемы

Идентификатор	0xF1
Адресаты	все подсистемы
Обработчики	запрос обрабатывает только та подсистема, которой он направлен
Формат поля данных запроса	<pre>struct data { uint16_t id; // Идентификатор параметра; int32_t values; // Значение(я) параметра; attribute_t attribute; };</pre>
Формат поля данных ответа	<pre>struct answer { uint16_t id; // идентификатор параметра; attribute_t attribute; };</pre>

В ответ на запрос подсистема проверяет значение параметра и его идентификатор и если все в порядке, то ставит его в очередь на сохранение в ЭНОЗУ. Высылается ответ, содержащий статус ответа (Accepted если параметр принят и поставлен в очередь на запись). Если параметра с заданным идентификатором нет, высылается ответ InvalidId. Если значение некорректно, высылается статус InvalidValue, если количество данных (значений в поле values) не совпадает с размером параметра (больше или меньше данных), высылается статус WrongData. Если указанной подсистемы нет в составе ЛС, General высылает ответ ModuleNotExist от имени этой подсистемы.

Со списком параметров подсистем и допустимых значений можно ознакомиться в описании программы АБМИ.00098.

SetupCorrectionTable. Запись таблицы коррекции угла подъема от давления.

RetrieveLimits. Запрос пределов перемещения.

Идентификатор	0x80
Адресаты	Horizontal, Vertical, Nozzle
Обработчики	Horizontal, Vertical, Nozzle
Формат поля данных запроса	Запрос не содержит данных
Формат поля данных ответа	<pre>struct answer { int32_t min; // минимальная граница; int32_t max; // максимальная граница; attribute_t attribute; };</pre>

Open. Открытие затвора.

Идентификатор	0x21
Адресаты	Valve1, Valve2
Обработчики	Deployer, Detector, ExternalConn, Radio, Valve1, Valve2
Формат поля данных запроса	<pre>struct data { uint16_t key; // ключевое значение для подтверждения команды (всегда 0x4C70); attribute_t attribute; };</pre>
Формат поля данных ответа	Ответ не содержит данных

Исполнение запроса вызывает открытие заданного затвора.

Возможны следующие сценарии исполнения запроса:

1. Если ЛС находится в дежурном положении, разворачивается или сворачивается, то подсистема Deployer выдает ответ Denied, управление затвором не происходит.

2. Если ЛС находится в процессе поиска очага загорания, то подсистема Detector выдает ответ Denied, управление затвором не происходит.
3. Если указанный затвор присутствует в составе ЛС, соответствующая подсистема выдает ответ Accepted и инициирует открытие указанного затвора, если затвор неисправен, она выдает ответ ModuleFault
4. Если запрос направлен в адрес подсистемы, для которой его исполнение не предусмотрено, тогда последняя возвращает статус WrongRequest.
5. Если запрос направлен в адрес подсистемы, которой нет в составе ЛС, тогда General отправляет ответ ModuleNotExist.

Close. Заккрытие затвора.

Идентификатор	0x24
Адресаты	Horizontal, Vertical, Nozzle, Deployer
Обработчики	Horizontal, Vertical, Nozzle, Deployer, Detector, ExternalConn, Radio
Формат поля данных запроса	<pre>struct data { uint16_t key; // ключевое значение для подтверждения команды (всегда 0x7456); __attribute__((packed)); }</pre>
Формат поля данных ответа	Ответ не содержит данных

Исполнение запроса вызывает открытие заданного затвора.

Возможны следующие сценарии исполнения запроса:

1. Если ЛС находится в процессе поиска очага загорания, то подсистема Detector выдает ответ Denied, управление затвором не происходит.
2. Если указанный затвор присутствует в составе ЛС, соответствующая подсистема выдает ответ Accepted и инициирует закрытие указанного затвора, если затвор неисправен, она выдает ответ ModuleFault
3. Если запрос направлен в адрес подсистемы, для которой его исполнение не предусмотрено, тогда последняя возвращает статус WrongRequest.
4. Если запрос направлен в адрес подсистемы, которой нет в составе ЛС, тогда General отправляет ответ ModuleNotExist.

Deploy. Развертывание ствола в рабочее положение

Идентификатор	0x7E
Адресаты	Deployer
Обработчики	Deployer, ExternalConn, Radio
Формат поля данных запроса	Запрос не содержит данных
Формат поля данных ответа	Ответ не содержит данных

Исполнение запроса старт процесса рзвертывания ствола в рабочее положение

Возможны следующие сценарии исполнения запроса:

1. Если ЛС находится в дежурном положении, разворачивается или сворачивается, то подсистема Deployer выдает ответ Accepted и начинает процедуру развертывания.
2. Если ЛС в рабочем положении, то подсистема Deployer выдает ответ Ok и ничего не делает.
3. Если блок управления поворотным узлом неисправен, Deployer выдает ответ ModuleFault
4. Если запрос направлен в адрес подсистемы, для которой его исполнение не предусмотрено, тогда последняя возвращает статус WrongRequest.
5. Если запрос направлен в адрес подсистемы, которой нет в составе ЛС, тогда General отправляет ответ ModuleNotExist.

Wrap. Сворачивание ствола в дежурное положение

Идентификатор	0x8D
Адресаты	Deployer
Обработчики	Detector, ExternalConn, Radio
Формат поля данных запроса	Запрос не содержит данных
Формат поля данных ответа	Ответ не содержит данных

Исполнение запроса старт процесса сворачивания ствола в дежурное положение

Возможны следующие сценарии исполнения запроса:

1. Если ЛС находится в рабочем положении, разворачивается или сворачивается, то подсистема Deployer выдает ответ Accepted и начинает процедуру сворачивания.
2. Если ЛС в дежурном положении, то подсистема Deployer выдает ответ Ok и ничего не делает.
3. Если блок управления поворотным узлом неисправен, Deployer выдает ответ ModuleFault
4. Если запрос направлен в адрес подсистемы, для которой его исполнение не предусмотрено, тогда последняя возвращает статус WrongRequest.
5. Запрос направлен в адрес подсистемы, которой нет в составе ЛС, тогда General отправляет ответ ModuleNotExist от имени этой подсистемы.

Restart. Перезапуск подсистемы (программы)

Запрос принимается только с идентификатором получателя All. Все остальные идентификаторы игнорируются(т.е. перезапустить можно только ПР целиком). В ответ на запрос завершаются все операции ПР, высылаются ответы от всех подсистем, после чего ПО сбрасывается и работа начинается заново. Данный запрос применяется после уставки параметров для вступления их в силу.

На данный момент сбрасывается ПО только системы управления ПР, ПО удаленных приводов, сканеров, затворов и др. оборудования, подключенного к ПР по внутреннему интерфейсу не сбрасывается (соответствующие функции должны быть реализованы в ПО этих устройств, должна быть доработана программа системы управления, чтобы задействовать эти функции).

StartJustify. Запуск юстировки системы ствол - детектор очага загорания

Идентификатор	0x55
Адресаты	Detector
Обработчики	Detector
Формат поля данных запроса	Запрос не содержит данных
Формат поля данных ответа	Ответ не содержит данных

В ответ на запрос ПР и начинает процедуру юстировки системы ПР-сканер

StartQuench. Старт тушения

Идентификатор	0xC5
Адресаты	Control
Обработчики	Control, Deployer, Detector, ExternalConn, Radio

Формат поля данных запроса	<pre> // Полный вариант используется для задания тушения по заданным координатам struct FullData { int16_t xcx; //< Центр очага по горизонтали в минутах int16_t cyu; //< Центр очага по вертикали в минутах int16_t csp; //< Требуемое положение насадка в мм int16_t isx; //< Ширина сектора в минутах int16_t isy; //< Высота сектора в минутах uint8_t ista; //< Шаг сканирования в градусах uint8_t distance; //< Расстояние до очага в метрах, используется для учета баллистики attribute__((packed)); }; // Упрощенный вариант используется для задания тушения по предзаписанной программе struct SimplifiedData { int16_t program; //< Номер программы от 0 до 7 int16_t xcx; //< Центр очага по горизонтали в минутах int16_t cyu; //< Центр очага по вертикали в минутах int16_t csp; //< Требуемое положение насадка в мм attribute__((packed)); }; // Вариант передачи таблицы коррекции вместе с программой тушения struct AdvancedData { int16_t xcx; //< Центр очага по горизонтали в минутах int16_t cyu; //< Центр очага по вертикали в минутах int16_t csp; //< Требуемое положение насадка в мм int16_t isx; //< Ширина сектора в минутах int16_t isy; //< Высота сектора в минутах uint8_t ista; //< Шаг сканирования в градусах int8_t table[256]; //< Таблица коррекции угла возвышения в зависимости от давления (значения в градусах) attribute__((packed)); }; </pre>
	Формат поля данных ответа

Исполнение запроса приводит к старту орошения заданной области (состояние затвора не меняется)

Возможны следующие сценарии исполнения запроса:

StartTrajectory. Программа тушения по траекториям

Возможно два варианта - с указанием точек траекторий непосредственно в пакете, либо с номером траекторий (для работы с пульта)

StartSeek. Старт поиска очага загорания

Идентификатор	0x76
Адресаты	Detector
Обработчики	Deployer, Detector, ExternalConn, Radio
Формат поля данных запроса	<pre> struct Data { int16_t x0; //< Начало сектора поиска по горизонтали в минутах int16_t x1; //< Конец сектора поиска по горизонтали в минутах int16_t y0; //< Начало сектора поиска по вертикали в минутах int16_t y1; //< Конец сектора поиска по вертикали в минутах int16_t analyse; //< Флаг необходимости анализа очагов (0,1) attribute__((packed)); }; </pre>
Формат поля данных ответа	Ответ не содержит данных

Исполнение запроса вызывает старт процесса поиска очага загорания

Возможны следующие сценарии исполнения запроса:

SwitchLimits. Включение/выключение контроля пределов перемещений ствола

Идентификатор	0xD9
Адресаты	All, Motors, Horizontal, Vertical, Nozzle, Deployer,
Обработчики	Horizontal, Vertical, Nozzle, Deployer,
Формат поля данных запроса	<pre> struct Data { uint16_t new_state; attribute__((packed)); }; </pre>
Формат поля данных ответа	Ответ не содержит данных

Поле запроса new_state содержит признак требуемого состояния контроля пределов:

Значение	Обозначение	Назначение
0	Off	Выключить пределы перемещений, при этом для приводов с энкодерами перестют учитываться записанные в ЭНОЗУ значения пределов перемещений, для приводов, подключаемых непосредственно к системе управления появляется возможность бесконечной работы в обе стороны, т.к. фактически, на время движения отключается счет координат. Для удаленных приводов (ВЗГ ствол) счет координат сохраняется, однако пределы все равно не учитываются.
1	On	Включить пределы перемещений, при этом, если в реджмие без пределов были установлены новые пределы они вступают в силу.
2	Reject	Отменить изменение пределов перемещений: для приводов, подключаемых к системе управления непосредственно эта команда отменяет установку новых пределов и оставляет старые пределы (если была изменена нулевая точка, то она все-равно изменится на новую)

Только в режиме без контроля пределов доступны возможности установки нулевой точки и записи текущего положения привода в качестве одного из пределов. В то время как в режиме с пределами эти параметры для каждого привода доступны для записи непосредственных значений.

Для приводов с датчиками конечных положений запрос SwitchLimits не имеет смысла, поэтому такие приводы возвращают статус NotSupported.

Lockout. Включить/выключить аварийную блокировку ствола

Идентификатор	0xE9
Адресаты	All
Обработчики	All
Формат поля данных запроса	<pre>struct Data { uint16_t key; // Ключ операции: 0x2FFA - включить блокировку; 0x7353 - отключить __attribute__((packed)); };</pre>
Формат поля данных ответа	Ответ не содержит данных

При получении данного запроса с кодом 0x2FFA (включить блокировку fnv1a("Enable")) все действия ствола останавливаются, затворы, если они есть закрываются. После выполнения запроса ствол переходит в режим, в котором все команды управления игнорируются (высылается ответ Denied от подсистемы General). Запросы чтения данных, получения статуса и записи параметров по прежнему будут работать.

При получении данного запроса с кодом 0x7353 (выключить блокировку fnv1a("Disable")) ствол переходит в нормальный режим работы.

CleanFlash. Очистить ЭНОЗУ.

Идентификатор	0x16
Адресаты	General
Обработчики	General
Формат поля данных запроса	<pre>struct Data { uint16_t key; // Ключевое значение для подтверждения команды (всегда 0xA4F6) __attribute__((packed)); };</pre>
Формат поля данных ответа	Ответ не содержит данных

Исполнение запроса вызывает полную очистку ЭНОЗУ. Сразу после отправки команды главный модуль сбрасывается.

GetCrashData. Получить информацию об аппаратном отказе

Идентификатор	0x06
Адресаты	General, All
Обработчики	General

Формат поля данных ответа

```
struct __attribute__((packed)) FaultInfo {
    unsigned __attribute__((packed)) signature;
    unsigned __attribute__((packed)) archetype;
    unsigned __attribute__((packed)) sb; //< Указатель стека в исключении отказа;
    unsigned __attribute__((packed)) r0; //< Регистр r0, сохраненный на стеке при возникновении отказа;
    unsigned __attribute__((packed)) r1; //< Регистр r1, сохраненный на стеке при возникновении отказа;
    unsigned __attribute__((packed)) r2; //< Регистр r2, сохраненный на стеке при возникновении отказа;
    unsigned __attribute__((packed)) r3; //< Регистр r3, сохраненный на стеке при возникновении отказа;
    unsigned __attribute__((packed)) r12; //< Регистр r12, сохраненный на стеке при возникновении отказа;
    unsigned __attribute__((packed)) lr; //< Регистр lr, сохраненный на стеке при возникновении отказа;
    unsigned __attribute__((packed)) pc; //< Регистр pc, сохраненный на стеке при возникновении отказа;
    union {
        struct { // Cortex-M3 specific registers
            unsigned __attribute__((packed)) tim2 cnt; //< Счетчик таймера2 (может использоваться для работы с энкодером);
            unsigned __attribute__((packed)) tim3 cnt; //< Счетчик таймера3 (может использоваться для работы с энкодером);
            unsigned __attribute__((packed)) tim4 cnt; //< Счетчик таймера4 (может использоваться для работы с энкодером);
            unsigned __attribute__((packed)) tim5 cnt; //< Счетчик таймера5 (может использоваться для работы с энкодером);
            unsigned __attribute__((packed)) hfsr; //< Регистр статуса тяжелого отказа;
            unsigned __attribute__((packed)) hfar; //< Регистр адреса отказа шины;
            unsigned __attribute__((packed)) mmfar; //< Регистр адреса отказа блока защиты памяти;
            unsigned __attribute__((packed)) cfsr; //< Регистр статус отказов;
            const void __attribute__((packed)) tcb; //< Указатель на tcb потока, исполнявшегося в момент возникновения отказа;
            unsigned __attribute__((packed)) psr; //< Регистр psr, сохраненный на стеке при возникновении отказа;
        };
        struct { // ARM7TDMI specific registers
            unsigned __attribute__((packed)) cpsr; //< current program status register;
            unsigned __attribute__((packed)) spsr; //< saved program status register;
        };
    };
};
```

Формат поля данных запроса

Запрос не содержит данных

- Команда является дублером команды **GetParam General FaultInfo**

GetParam Hotbeds. Получение сведений об обнаруженных очагах

Данная команда относится к команде GetParam, и выведена в отдельный блок Wiki поскольку все часто забывают формат сообщения

Запрос:

```
struct header {
    uint8_t 0x9d; // GetParam
    uint8_t 0xc8; // Detector
    uint8_t 0x00; // request status always == 0
    uint8_t 0x02; // size=2
    uint16_t 0x23FE; // fnv1a( "Hotbeds")
};
```

Ответ состоит из идентификатора параметра fnv1a("Hotbeds") и массива Hotbed[n] где n=0...4

```
struct Hotbed { // Структура одного очага
    int32_t x1; // лев
    int32_t x2; // прав
    int32_t y1;
    int32_t y2;
    int32_t brightness; //Яркость очага
};
```

```

| struct header { // Ответ ПР ЛСД
|   uint8_t 0x9d; // GetParam
|   uint8_t 0xc8; // Detector
|   uint8_t status; // answer status
|   uint8_t size; // sizeof(fnv1a("Hotbeds")) + n*sizeof(Hotbed)
|   uint16_t 0x23FE; // fnv1a( "Hotbeds")
|   Hotbed hotbeds[n]; // массив очагов
| }

```

Источник —

«
http://192.168.1.38/mediawiki/index.php?title=Протокол_управления_роботизированным_лафетным_стволом&oldid=6642»

-
- Последнее изменение этой страницы: 09:36, 26 сентября 2018.

