# О документе

Автор Прищепенко Борис prishchepenko@inbox.ru

Текущая ревизия 2017-01-09

Протокол связи применим к аппаратным версиям тренажера:

|  |  |
| --- | --- |
| Версия 1 | Приводящие/отводящие мышцы |
| Версия 2 | Квадрицепс, первая рама |
| Версия 2.1 | Квадрицепс, вторая рама |

# Термины и обозначения

HMI - human-machine interface - монитор

RTCU - real-time control unit - блок управления

SERVO - привод электродвигателя

STOP - кнопка аварийного отключения

RFID - бесконтактная идентификация

AUX – вспомогательный механизм (актуатор)

ENCODER – абсолютный энкодер – датчик положения вала

EXTERNAL – внешний компьютер, подключаемый к диагностическому порту блока-управления

# Описание аппаратной части

Рис.1 Логическая структура тренажера



# Описание протокола

Битовая скорость 115200бит/сек

Проверка четности отсутствует

Стоповых бит 1

Обмен сообщениями между HMI и RTCU происходит квазинезависимо:

\* сообщения от RTCU высылаются в зависимости от собственного состояния блока и происходящих с ним событий

\* RTCU высылает копии тех же самых сообщений на диагностический порт

\* сообщения не требуют ответа

\* доставку сообщения от HMI можно проверить косвенным методом (услышав соответствующее периодическое сообщение от RTCU)

Сообщение состоит из следующих полей:

|  |  |  |
| --- | --- | --- |
| TAG (команда) | 1 байт | Тип сообщения |
| ID (или порядковый номер транзакции) | 4 байта little-endian |  |
| Длина поля данных | 2 байта little-endian | Значения от 0 до 65535 |
| Данные | От 0 до 65535 байт |  |
| Контрольная сумма | 4 байта little-endian | CRC32 |

Нумерация транзакции происходит независимо для RTCU и для HMI. Каждое последующее сообщение имеет номер транзакции на единицу больший чем предыдущее сообщение. Это поле может использоваться для контроля надежности связи.



Для идентификации начала сообщения используется байт стаффинг.

|  |  |  |
| --- | --- | --- |
| Байт | Символическое имя | Описание |
| 0xC0 | SLIP\_END | Признак начала пакета, пакет всегда начинается сразу после такого байта |
| 0xDB | SLIP\_ESC | Признак начала замещения байта SLIP\_END или SLIP\_ESC |
| 0xDC | SLIP\_ESC\_END | Пара SLIP\_ESC>SLIP\_ESC\_END замещает SLIP\_END |
| 0xDD | SLIP\_ESC\_ESC | Пара SLIP\_ESC>SLIP\_ESC\_ ESC замещает SLIP\_ ESC |

Пример сообщения:

|  |  |
| --- | --- |
| **C0 C0 C0 C0** 00 **59 02 00 00** 17 00 02 0F 0A OE 00 3B 21 9D 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 89 04 67 81 | Полное сообщение |
| **C0 C0 C0 C0** | **Признак начала пакета** |
| 00 | Тип сообщения |
| **59 02 00 00** | **ID** |
| 17 00 | Длина поля данных (23) |
| *02 0F 0A OE 00 3B 21 9D 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00* | *Поле данных (23 байта)* |
| **89 04 67 81** | **Контрольная сумма** |

# Контрольная сумма (CRC32)

|  |
| --- |
| #ifndef CRC32\_H  #define CRC32\_H  #include <stdint.h>  extern void crc32(uint32\_t \*pdwCrc32,uint8\_t bNewByte);  const uint32\_t CRC32\_SEED = 0x00000000;  #endif // CRC32\_H |
| #include "crc32.h"  uint32\_t crc32Table[256];  bool crc32TableCreated = false;  void make\_crc\_table(void)  {  uint32\_t c;  int32\_t n, k;  for (n = 0; n < 256; n++)  {  c = (uint32\_t) n;  for (k = 0; k < 8; k++)  {  if (c & 1)  {  c = 0xedb88320L ^ (c >> 1);  }  else  {  c = c >> 1;  }  }  crc32Table[n] = c;  }  crc32TableCreated = true;  }  void crc32(uint32\_t \*pdwCrc32,uint8\_t bNewByte)  {  uint32\_t c = (\*pdwCrc32) ^ 0xffffffffL;  if (crc32TableCreated==false)  {  make\_crc\_table();  }  c = crc32Table[(c ^ (uint32\_t)bNewByte) & 0xff] ^ (c >> 8);  (\*pdwCrc32) = (c ^ 0xffffffffL);  } |

# Математическая модель

## Положение

**Точка A** - точка начала эксцентрического движения / точка окончания концентрического движения

**Точка B** - точка начала концентрического движения / точка окончания эксцентрического движения

Эксцентрическое движение – из А в В

Концентрическое движение – из В в А

Положение механизмов может быть абсолютным и относительным.

**Абсолютное** положение механизма измеряется датчиком в шагах и используется для задания

* точек A и B основного механизма (персональные настройки = свойство пользователя)
* точек положения вспомогательных механизмов (персональные настройки = свойство пользователя)
* максимального и минимального допустимого положения основного и вспомогательных механизмов (настройки тренажера = свойство тренажера)

|  |  |  |
| --- | --- | --- |
| **Тип привода** | **Датчик** | **Величина шага абсолютного положения** |
| Festo  (основной механизм) | Абсолютный многооборотный энкодер, встроенный в двигатель | (Версия 1) 0.01мм |
| Nord  (основной механизм) | Абсолютный однооборотный энкодер, связанный с рабочим валом | (Версия 2) 0.088градуса  (1 деление энкодера из 4096 делений на полный оборот)  (Версия 2.1) 0.011градуса  (1 деление энкодера из 32768 делений на полный оборот) |
| Linak  (вспомогательный механизм) | Потенциометр, встроенный в актуатор | 1/4096 полной амплитуды штока |
|  |  |  |
|  |  |  |

**Относительное** положение основного механизма (в процентах):

0% - точка A

100% - точка B

## Скорость

**Скорость** **относительная** задаётся как время перемещения основного механизма между A и B или B и A. **Скорость абсолютная** задаётся как время перемещения основного или вспомогательного механизма между минимальным и максимальным положениями.

Скорость всегда положительна и задаётся в миллисекундах.

## Время

**Паузы** всегда задаются в миллисекундах

## Усилие

Усилие измеряется в условных граммах.

Датчик усилия

forceSensorRawValue – первичное значение

forceSensorValue – значение датчика усилия, используемое в программе и приложении

forceSensorValue = (forceSensorRawValue\* forceSensorGain) + forceSensorOffset

## Энкодер

encoderRawValue – первичное значение

encoderValue – значение датчика положения, используемое в программе и приложении

encoderBitMask – битовая маска на основе количества разрядов энкодера (=0x7FFF для 15бит)

encoderValue = (encoderRawValue\*encoderDirection +encoderOffset) & encoderBitMask

# Тренировка

## Термины

**Тренировка** (excercise) делится на **подходы** (set), состоящие из **повторений** (reps).

Повторение состоит из двух **движений** (moves)

Типы тренировок:

|  |  |
| --- | --- |
| Изокинетическая  (isokinetic excercise) | Движения всегда между точками A и B |
| Обобщенная изокинетическая  (generic isokinetic excercise) | Движения между произвольными точками в пределах точек A и B |

## Изокинетическая тренировка

В изокинетической тренировке **подход** описывается структурой **IsokineticSetSettings**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символичеcкое имя | Описание | Диапазон значений |
| 0 | 4 | int32\_t | pauseTime | Пауза перед подходом в миллисекундах | 1000 ... (2147483647) |
| 4 | 4 | int32\_t | firstInterruptionTime | Пауза перед первым движением | 0 ... (2147483647) |
| 8 | 4 | int32\_t | secondInterruptionTime | Пауза перед вторым движением | 0 ... (2147483647) |
| 12 | 4 | int32\_t | startPoint | Начальное положение подхода  0 - положение A  1 - положение B | 0 или 1 |
| 16 | 4 | int32\_t | speedAB | Относительная скорость движения AB | От 1% до 100% от эквивалента speedAbsMainMax |
| 20 | 4 | int32\_t | speedBA | Относительная скорость движения BA | От 1% до 100% от эквивалента speedAbsMainMax |
| 24 | 4 | int32\_t | repsCount | Количество повторений | 1 ... (2147483647) |
|  | Σ=28 |  |  |  |  |

## Обобщенная изокинетическая тренировка

В обобщенной изокинетической тренировке **движение** описывается структурой **GenericMoveSettings**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символичеcкое имя | Описание | Диапазон значений |
| 0 | 4 | int32\_t | destinationPositionRel | Относительное положение окончания движения | От 0% до 100% |
| 4 | 4 | int32\_t | speed | Скорость движения | В миллисекундах |
|  | Σ=8 |  |  |  |  |

**Подход** описывается структурой **GenericSetSettings**, состоящей из

* пауз перед подходом (2 компоненты)
* начального положения подхода
* количество повторений в подходе
* массив структур GenericMoveSettings (по две на каждое повторение)

Размер этой структуры – переменный, зависящий от количества повторений. Например, для 10 повторений структура подхода будет состоять из 44 значений (4+2\*2\*10), для 20 повторений из 84 значений (4+2\*2\*20).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символичеcкое имя | Описание | Диапазон значений |
| 0 | 4 | int32\_t | pause1 | Пауза перед переходом к стартовому положению | мс |
| 4 | 4 | int32\_t | pause2 | Пауза после перехода к стартовому положению | мс |
| 8 | 4 | int32\_t | startPositionRel | Стартовое положение | (0%...100%)\*100 |
| 12 | 4 | int32\_t | moveCount | Количество движений  (=количество повторений\*2) | (Версия 2.1)  1 ... 50  (для последующих версий железа будет увеличено) |
| 16 | 8\*moveCount | GenericMoveSettings | move[moveCount] | Массив движений |  |
|  | Σ=16+8\* moveCount |  |  |  |  |



# Режимы RTCU

В любой момент времени RTCU может находиться в одном из 7 режимов

|  |  |  |
| --- | --- | --- |
| 0 | Инициализация | INITIALIZING |
| 1 | Выключен | IDLE |
| 2 | Ожидание | WAITING |
| 3 | Парковка | PARKING |
| 4 | Персональные настройки | PERSONAL |
| 5 | Силовой тест - концентрика | TEST\_CONCENTRIC |
| 6 | Силовой тест - эксцентрика | TEST\_ECCENTRIC |
| 7 | Изокинетическая тренировка | EXERCISE\_ISOKINETIC |
| 8 | Авария | ERROR |
| 9 | Силовой тест - статика | TEST\_STATIC |
| 10 | Обобщенная изокинетическая тренировка | GENERIC\_EXERCISE\_ISOKINETIC |

## INITIALIZING [Инициализация]

В этом состоянии RTCU оказывается сразу после включения (подачи питания, сброса)

1) Происходит инициализация и проверка собственных внутренних блоков и интерфейсов RTCU. При обнаружении неисправности RTCU переходит в режим ERROR.

Возможные неисправности:

\* не подключен датчик усилия

\* батарея часов разряжена или часы не настроены

2) Производится установка соединения с SERVO

Если пункты 1 2 выполнены, RTCU переходит в режим IDLE

Все сообщения от HMI игнорируются

## IDLE [Выключен]

(Возможно в будущей версии - RTCU ожидает нажатия на кнопку STOP. Это нажатие служит в том числе для проверки её функционирования.)

SERVO выключен (разблокирован)

RTCU ожидает от HMI команды включения SERVO (**TAG\_EnableServo**) и при ее получении переходит в режим WAITING

## WAITING [Ожидание]

SERVO включен (заблокирован)

В этом режиме производится запись параметров и переход к другим рабочим состояниям

Параметры делятся на

\* персональные параметры - собщение TAG\_LoadPersonalSettings

\* параметры тренировки - сообщение TAG\_LoadIsokineticExcerciseSettings, TAG\_LoadGenericIsokineticExcerciseSettings

\* параметры тренажера - сообщение TAG\_LoadMachineSettings

При получении сообщения TAG\_Parking RTCU переходит в режим PARKING. Это происходит при условии, что в RTCU записаны персональные параметры и параметры тренажера, иначе сообщение игнорируется.

При получении сообщения TAG\_Personal RTCU переходит в режим PERSONAL. Это происходит при условии, что в RTCU записаны параметры тренажера, иначе сообщение игнорируется.

При получении сообщения TAG\_TestConcentric RTCU переходит в режим TEST\_CONCENTRIC. Это происходит при условии, что в RTCU записаны персональные параметры и параметры тренажера, иначе сообщение игнорируется.

При получении сообщения TAG\_TestEccentric RTCU переходит в режим TEST\_ECCENTRIC. Это происходит при условии, что в RTCU записаны персональные параметры и параметры тренажера, иначе сообщение игнорируется.

При получении сообщения TAG\_ExcerciseIsokinetic RTCU переходит в режим EXERCISE\_ISOKINETIC. Это происходит при условии, что в RTCU записаны персональные параметры, параметры тренажера и параметры изокинетической тренировки, иначе сообщение игнорируется.

## PARKING [Парковка]

SERVO включен и двигается в положение парковки (удобное для посадки на тренажер)

Вспомогательные механизмы двигаются в персональное положение

При достижении положения парковки RTCU переходит в режим WAITING

Все сообщения от HMI игнорируются

Остановить движение можно с помощью кнопки STOP

## PERSONAL [Персональные настройки]

SERVO включен (заблокирован)

С помощью сообщений TAG\_PersonalButtonPressed / TAG\_PersonalButtonHold / TAG\_PersonalButtonReleased производится перемещение основного и вспомогательных механизмов.

Перемещение начинается сообщением TAG\_PersonalButtonPressed.

Перемещение останавливается сообщением TAG\_PersonalButtonReleased.

Продолжение перемещения подтверждается сообщением TAG\_PersonalButtonHold.

Если в течение 200мс после сообщения TAG\_PersonalButtonPressed не пришло сообщение TAG\_PersonalButtonHold или TAG\_PersonalButtonReleased, перемещение прекращается.

Если в течение 200мс после сообщения TAG\_PersonalButtonHold не пришло сообщение TAG\_PersonalButtonHold или TAG\_PersonalButtonReleased, перемещение прекращается.

При получении сообщения TAG\_PersonalExit RTCU переходит в режим WAITING

Остановить движение можно с помощью кнопки STOP (аварийно)

## TEST\_CONCENTRIC [Силовой тест - концентрика]

Выйти из режима можно с помощью кнопки STOP (аварийно) или сообщением TAG\_Cancel (планово)

|  |  |  |
| --- | --- | --- |
| Фаза 1 | PHASE\_HOMING=0 | переход в точку B (фаза будет пропущена если основной механизм уже находится в точке B) |
| Фаза 2 | PHASE\_PAUSE=1 | пауза перед тестом |
| Фаза 3 | PHASE\_TEST\_CONCENTRIC=2 | тест |

## TEST\_ECCENTRIC [Силовой тест - эксцентрика]

Выйти из режима можно с помощью кнопки STOP (аварийно) или сообщением TAG\_Cancel (планово)

|  |  |  |
| --- | --- | --- |
| Фаза 1 | PHASE\_HOMING=0 | переход в точку A (фаза будет пропущена если основной механизм уже находится в точке A) |
| Фаза 2 | PHASE\_PAUSE=1 | пауза перед тестом |
| Фаза 3 | PHASE\_TEST\_ECCENTRIC=3 | тест |

## EXCERCISE\_ISOKINETIC [Изокинетическая тренировка]

Выйти из режима можно с помощью кнопки STOP (аварийно) или сообщением TAG\_Cancel (планово)

Подход:

|  |  |  |
| --- | --- | --- |
| Фаза 1 | PHASE\_HOMING=0 | переход в начальное положение подхода (см. IsokineticSetSettings.startPoint) |
| Фаза 2 | PHASE\_PAUSE=1 | пауза перед подходом (см. IsokineticSetSettings.pauseTime) |
| Фаза 3 | PHASE\_ISOKINETIC\_FIRSTMOVE=4 | Движение 1 |
| Фаза 4 | PHASE\_ISOKINETIC\_FIRSTINTERRUPTION=5 | Пауза после первого движения (см. IsokineticSetSettings.firstInterruptionTime) |
| Фаза 5 | PHASE\_ISOKINETIC\_SECONDMOVE=6 | Движение 2 |
| Фаза 6 | PHASE\_ISOKINETIC\_SECONDINTERRUPTION=7 | Пауза после второго движения (см. IsokineticSetSettings.secondInterruptionTime) |



## GENERIC\_EXCERCISE\_ISOKINETIC [Обобщенная\_изокинетическая тренировка]

Выйти из режима можно с помощью кнопки STOP (аварийно) или сообщением TAG\_Cancel (планово)

Подход:

|  |  |  |
| --- | --- | --- |
| Фаза 1 | PHASE\_PAUSE1=9 | пауза перед переходом в начальное положение (см. GenericSetSettings.pause1) |
| Фаза 2 | PHASE\_HOMING=0 | переход в начальное положение подхода (см. GenericSetSettings.startPositionRel) |
| Фаза 3 | PHASE\_PAUSE2=10 | пауза после перехода в начальное положение (см. GenericSetSettings.pause2) |
| Фаза 4 | PHASE\_GENERIC\_MOVE=11 | Движение |

## FAULT [Авария]

RTCU попадает в этот режим если произошла ошибка, после которой невозможно продолжать работу

Тип ошибки

|  |  |  |
| --- | --- | --- |
| ERROR\_ServoInternal | 0x00 | SERVO сигнализировал о собственной внутренней ошибке |
| ERROR\_ServoConnectionLost | 0x01 | Потеря связи с SERVO |
| ERROR\_ServoVoltageLost | 0x02 | Отключилось силовое напряжение |
| ERROR\_ServoControlLost | 0x03 | Кто-то перехватил управление SERVO |
| ERROR\_RtcuLogicError | 0x04 | В программе RTCU зафиксировано недопустимое состояние |
| ERROR\_AuxMoved | 0x05 | Несанкционированный сдвиг вспомогательного механизма |
| ERROR\_HmiParametersError | 0x06 | Недопустимые параметры в сообщениях от HMI |
| ERROR\_StrainGaugeLost | 0x07 | Нет сигнала с датчика усилия |
| ERROR\_StoppedManually | 0x08 | Нажата кнопка STOP |
| ERROR\_PositionMainSensorLost | 0x09 | Нет сигнала с датчика положения основного механизма |
| ERROR\_RfidReaderLost | 0x0A | Нет сигнала с RFID-считывателя |
|  |  |  |
|  |  |  |
| ERROR\_Unknown | 0xFF | Неизвестная ошибка |

Выход из режима

\* по сообщению TAG\_ResetError

\* по длительному нажатию кнопки STOP

## TEST\_STATIC [Силовой тест - статика]

Выйти из режима можно с помощью кнопки STOP (аварийно) или сообщением TAG\_Cancel (планово)

|  |  |  |
| --- | --- | --- |
| Фаза 1 | PHASE\_HOMING=0 | переход в точку теста (фаза будет пропущена если основной механизм уже находится в точке теста) |
| Фаза 2 | PHASE\_PAUSE=1 | пауза перед тестом |
| Фаза 3 | PHASE\_TEST\_STATIC=8 | тест |

# Сообщения

## Таблица всех типов сообщений

|  |  |  |  |
| --- | --- | --- | --- |
| **Значение** | **Символическое имя** | **Отправитель** | **Получатель** |
| 0x00 | TAG\_ReportCurrentMode | RTCU | HMI,EXTERNAL |
| 0x01 | TAG\_ReportPersonalSettings | RTCU | HMI,EXTERNAL |
| 0x02 | TAG\_ReportExcerciseSettings | RTCU | HMI,EXTERNAL |
| 0x03 | TAG\_ReportMachineSettings | RTCU | HMI,EXTERNAL |
| 0x04 | TAG\_RfidProximity | RTCU | HMI,EXTERNAL |
| 0x05 | TAG\_RtcuDebugMessage | RTCU | EXTERNAL |
| 0x06 | TAG\_ReportGenericExcerciseSettings | RTCU | HMI,EXTERNAL |
| 0x07 | TAG\_ReportMachineSettingsExtended | RTCU | HMI,EXTERNAL |
| 0x08 | TAG\_ReportServoMode | RTCU | EXTERNAL |
|  |  |  |  |
| 0x40 | TAG\_EnableServo | HMI | RTCU |
| 0x41 | TAG\_LoadPersonalSettings | HMI | RTCU |
| 0x42 | TAG\_LoadExcerciseSettings | HMI | RTCU |
| 0x43 | TAG\_LoadMachineSettings | EXTERNAL | RTCU |
| 0x44 | TAG\_Parking | HMI | RTCU |
| 0x45 | TAG\_Personal | HMI | RTCU |
| 0x46 | TAG\_PersonalExit | HMI | RTCU |
| 0x47 | TAG\_PersonalButtonPressed | HMI | RTCU |
| 0x48 | TAG\_PersonalButtonHold | HMI | RTCU |
| 0x49 | TAG\_PersonalButtonReleased | HMI | RTCU |
| 0x4A | TAG\_TestConcentric | HMI | RTCU |
| 0x4B | TAG\_TestEccentric | HMI | RTCU |
| 0x4C | TAG\_ExcerciseIsokinetic | HMI | RTCU |
| 0x4D | TAG\_ResetError | HMI | RTCU |
| 0x4E | TAG\_Cancel | HMI | RTCU |
| 0x4F | TAG\_TestStatic | HMI | RTCU |
| 0x50 | TAG\_LoadGenericExcerciseSettings | HMI | RTCU |
| 0x51 | TAG\_GenericExcerciseIsokinetic | HMI | RTCU |
| 0x52 | TAG\_LoadMachineSettingsExtended | EXTERNAL | RTCU |
|  |  |  |  |
| 0x80 | TAG\_CheckRfidProximity | RTCU,RFID | RFID,RTCU |
|  |  |  |  |
|  |  |  |  |

## TAG\_ReportCurrentMode

Передаётся каждые 100мс. В зависимости от режима имеет различную длину поля данных

### Общий формат

Поля *mode, currentDateTime, odometerTime, odometerWay* присутствуют в сообщении всегда, независимо от режима.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 1 | uint8\_t | mode | Текущий режим RTCU |
| 1 | 6 | struct | currentDateTime | Показания встроенных часов RTCU  struct TCurrentDateTime  {  uint8\_t year;//00-99  uint8\_t month;//01-12  uint8\_t date;//01-31  uint8\_t hours;//00-23  uint8\_t minutes;//00–59  uint8\_t seconds;//00–59  }; |
| 7 | 4 | int32\_t | odometerTime | Суммарное время работы тренажера  [в минутах] |
| 11 | 4 | int32\_t | odometerWay | Суммарный путь основного механизма тренажера  [в метрах или оборотах] |

### Режим INITIALIZING

Дополнительных данных нет

Суммарная длина поля данных равна 15 байт

### Режим IDLE / WAITING / PARKING / PERSONAL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |

Суммарная длина поля данных равна 31 байт

### Режим ERROR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 1 | uint8\_t | errorCode | Код ошибки |

Суммарная длина поля данных равна 16 байт

### Режим TEST\_CONCENTRIC / TEST\_ECCENTRIC

Длина поля данных будет разной в зависимости от фазы теста

#### Фаза 1 (переход в начальное положение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_HOMING |

Суммарная длина поля данных равна 35 байт

#### Фаза 2 (пауза перед тестом)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_PAUSE |
| 34 | 4 | int32\_t | timeToTest | Оставшееся до начала теста время (в миллисекундах) |

Суммарная длина поля данных равна 39 байт

#### Фаза 3 (движение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_TEST\_CONCENTRIC или PHASE\_TEST\_ECCENTRIC |
| 34 | 4 | int32\_t | positionRel | Относительное положение основного механизма (в процентах \* 100) |
| 39 | 4 | int32\_t | force | Усилие |

Суммарная длина поля данных равна 43 байт

### Режим EXСERCISE\_ISOKINETIC

Длина поля данных будет разной в зависимости от фазы подхода

#### Фаза 1 (переход в начальное положение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_HOMING |
| 35 | 4 | int32\_t | setIndex | Номер подхода (0…) |
|  |  |  |  |  |

Суммарная длина поля данных равна 39 байт

#### Фаза 2 (пауза перед подходом)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_PAUSE |
| 35 | 4 | int32\_t | setIndex | Номер подхода (0…) |
| 39 | 4 | int32\_t | timeToSet | Оставшееся до начала теста время (в миллисекундах) |
|  |  |  |  |  |

Суммарная длина поля данных равна 43 байт

#### Фаза 3/5 (движение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_ISOKINETIC\_FIRSTMOVE или PHASE\_ISOKINETIC\_SECONDMOVE |
| 35 | 4 | int32\_t | setIndex | Номер подхода (0…) |
| 39 | 4 | int32\_t | repIndex | Номер повторения (0…) |
| 43 | 4 | int32\_t | repDirection | Направление движения  0 – эксцентрическое  1 – концентрическое |
| 47 | 4 | int32\_t | positionRel | Относительное положение основного механизма (в процентах \* 100) |
| 51 | 4 | int32\_t | force | Усилие |
|  |  |  |  |  |

Суммарная длина поля данных равна 55 байт

#### Фаза 4/6 (пауза между движениями)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_ISOKINETIC\_FIRSTINTERRUPTION  PHASE\_ISOKINETIC\_SECONDINTERRUPTION |
| 35 | 4 | int32\_t | setIndex | Номер подхода (0…) |
| 39 | 4 | int32\_t | timeToSecondMove | Оставшееся до начала второго движения время (в миллисекундах) |
|  |  |  |  |  |

Суммарная длина поля данных равна 43 байт

### Режим TEST\_STATIC

Длина поля данных будет разной в зависимости от фазы теста

#### Фаза 1 (переход в положение теста)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_HOMING |
| 35 | 4 | int32\_t | positionRel | Относительное положение основного механизма (в процентах \* 100) |
|  |  |  |  |  |

Суммарная длина поля данных равна 39 байт

#### Фаза 2 (пауза перед тестом)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_PAUSE |
| 35 | 4 | int32\_t | positionRel | Относительное положение основного механизма (в процентах \* 100) |
| 39 | 4 | int32\_t | timeToTest | Оставшееся до начала теста время (в миллисекундах) |

Суммарная длина поля данных равна 43 байт

#### Фаза 3 (тест)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_TEST\_STATIC |
| 35 | 4 | int32\_t | positionRel | Относительное положение основного механизма (в процентах \* 100) |
| 39 | 4 | int32\_t | force | Усилие |
| 43 | 4 | int32\_t | timeToEnd | Оставшееся до окончания теста время (в миллисекундах) |

Суммарная длина поля данных равна 47 байт

### Режим GENERIC\_EXСERCISE\_ISOKINETIC

Длина поля данных будет разной в зависимости от фазы подхода

#### Фаза 1 (пауза перед переходом в начальное положение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_PAUSE1 |
| 35 | 4 | int32\_t | setIndex | Номер подхода (0…) |
| 39 | 4 | int32\_t | timeRemaining | Оставшееся время (в миллисекундах) |
|  |  |  |  |  |

Суммарная длина поля данных равна 43 байт

#### Фаза 2 (переход в начальное положение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_HOMING |
| 35 | 4 | int32\_t | setIndex | Номер подхода (0…) |
|  |  |  |  |  |

Суммарная длина поля данных равна 39 байт

#### Фаза 3 (пауза после перехода в начальное положение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_PAUSE1 |
| 35 | 4 | int32\_t | setIndex | Номер подхода (0…) |
| 39 | 4 | int32\_t | timeRemaining | Оставшееся время (в миллисекундах) |
|  |  |  |  |  |

Суммарная длина поля данных равна 43 байт

#### Фаза 4 (движение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 15 | 4 | int32\_t | positionMain | Текущее положение основного механизма |
| 19 | 4 | int32\_t | positionAux1 | Текущее положение вспомогательного механизма №1 |
| 23 | 4 | int32\_t | positionAux2 | Текущее положение вспомогательного механизма №2 |
| 27 | 4 | int32\_t | positionAux3 | Текущее положение вспомогательного механизма №3 |
| 31 | 4 | int32\_t | phase | Фаза = PHASE\_GENERIC\_MOVE |
| 35 | 4 | int32\_t | setIndex | Номер подхода (0…) |
| 39 | 4 | int32\_t | moveIndex | Номер движения в этом подходе (0…) |
| 43 | 4 | int32\_t | positionRel | Относительное положение основного механизма (в процентах \* 100) |
| 47 | 4 | int32\_t | force | Усилие |
|  |  |  |  |  |

Суммарная длина поля данных равна 51 байт

## TAG\_ReportPersonalSettings

Сообщение высылается каждые 1000мс когда RTCU находится в режиме WAITING.

Если персональные параметры еще не записаны в RTCU, то длина поля данных равна 0 байт.

Если персональные параметры записаны в RTCU, то длина поля данных равна 204 байт.

Персональные параметры находятся в энергозависимой памяти, поэтому стираются при выключении, перезагрузке, инициализации RTCU.

Таб.Х Персональные параметры (поле данных сообщений TAG\_LoadPersonalSettings и TAG\_ReportPersonalSettings)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 4 | int32\_t | positionMainA | Положение точки A основного механизма  (начало эксцентрического движения)  [в абсолютных единицах]  Не может быть выше positionMainMax.  Не может быть ниже positionMainMin. |
| 4 | 4 | int32\_t | positionMainB | Положение точки B основного механизма  (окончание эксцентрического движения)  [в абсолютных единицах]  Не может быть выше positionMainMax.  Не может быть ниже positionMainMin. |
| 8 | 4 | int32\_t | positionMainParking | Положение парковки основного механизма  (удобное для посадки и высадки)  [в абсолютных единицах] |
| 12 | 4 | int32\_t | positionAux1 | Положение вспомогательного механизма #1 |
| 16 | 4 | int32\_t | positionAux2 | Положение вспомогательного механизма #2 |
| 20 | 4 | int32\_t | positionAux3 | Положение вспомогательного механизма #3 |
| 24 | 4 | int32\_t | speedAbsMainParking | Абсолютная скорость основного механизма в режиме PARKING и в фазах PHASE\_HOMING  Абсолютная скорость не может быть выше 100% и ниже 1% от speedAbsMainMax |
| 28 | 4 | int32\_t | speedRelMainTestEccentric | Относительная скорость основного механизма во время Силового теста Эксцентрика  Относительная скорость не может быть выше 100% и ниже 1% от эквивалента speedAbsMainMax  Пример расчета:  speedAbsMainMax = 469ms  positionMainMax = 20000steps  positionMainMin = 100steps  positionMainA = 20000steps  positionMainB = 10000steps  максимальная относительная скорость (эквивалентная speedAbsMainMax ) =  469ms \* |20000-10000| / (20000-100) = 235ms |
| 32 | 4 | int32\_t | speedRelMainTestConcentric | Относительная скорость основного механизма во время Силового теста Концентрика  Относительная скорость не может быть выше 100% и ниже 1% от эквивалента speedAbsMainMax |
| 36 | 21\*4 | int32\_t[21] | forceCurveAB[21] | Параметры изокинетической кривой  для эксцентрического движения  в точках -0% 5% 10% ... 95% 100% диапазона AB |
| 120 | 21\*4 | int32\_t[21] | forceCurveBA[21] | Параметры изокинетической кривой  для концентрического движения  в точках -0% 5% 10% ... 95% 100% диапазона AB |
|  | Всего 204 |  |  |  |

## TAG\_ReportIsokineticExcerciseSettings

Сообщение высылается каждые 1000мс когда RTCU находится в режиме WAITING.

Если параметры тренировки еще не записаны в RTCU, то длина поля данных равна 0 байт.

Если параметры тренировки записаны в RTCU, то длина поля данных равна ххх байт.

Параметры тренировки находятся в энергозависимой памяти, поэтому стираются при выключении, перезагрузке, инициализации RTCU.

Таб.Х Параметры изокинетической тренировки (поле данных сообщений TAG\_LoadIsokineticExcerciseSettings и TAG\_ReportIsokineticExcerciseSettings)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 28\*N | IsokineticSetSettings | sets[N] | Массив структур, описывающих подход. Количество структур в массиве указывает количество подходов.  N = 1 ... 50 |
|  |  |  |  |  |

## TAG\_ReportGenericIsokineticExcerciseSettings

Сообщение высылается каждые 1000мс когда RTCU находится в режиме WAITING.

Если параметры тренировки еще не записаны в RTCU, то длина поля данных равна 0 байт.

Если параметры тренировки записаны в RTCU, то длина поля данных равна ххх байт.

Параметры тренировки находятся в энергозависимой памяти, поэтому стираются при выключении, перезагрузке, инициализации RTCU.

Таб.Х Параметры обобщенной изокинетической тренировки (поле данных сообщений TAG\_LoadGenericIsokineticExcerciseSettings и TAG\_ReportGenericIsokineticExcerciseSettings)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | ? | GenericSetSettings | sets[N] | Массив структур, описывающих подход. Количество структур в массиве указывает количество подходов.  N = 1 ... 10 |
|  |  |  |  |  |

## TAG\_ReportMachineSettings

Сообщение высылается каждые 1000мс когда RTCU находится в режиме WAITING.

Если параметры тренажера еще не записаны в RTCU, то длина поля данных равна 0 байт.

Если параметры тренажера записаны в RTCU, то длина поля данных равна 40 байт.

Параметры тренажера находятся в энергонезависимой памяти, поэтому не стираются при выключении RTCU.

Таб.Х Параметры тренажера (поле данных сообщений TAG\_LoadMachineSettings и TAG\_ReportMachineSettings)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 4 | int32\_t | positionMainMax | Максимально допустимое абсолютное положение основного механизма |
| 4 | 4 | int32\_t | positionMainMin | Минимально допустимое абсолютное положение основного механизма |
| 8 | 4 | int32\_t | positionAux1Max | Максимально допустимое абсолютное положение вспомогательного механизма #1 |
| 12 | 4 | int32\_t | positionAux1Min | Минимально допустимое абсолютное положение вспомогательного механизма #1 |
| 16 | 4 | int32\_t | positionAux2Max | … |
| 20 | 4 | int32\_t | positionAux2Min | … |
| 24 | 4 | int32\_t | positionAux3Max | … |
| 28 | 4 | int32\_t | positionAux3Min | … |
| 32 | 4 | int32\_t | speedAbsMainMax | Максимальная абсолютная скорость основного механизма  \* она записана в электроприводе  \* она эквивалентна 100% относительной скорости  Пример расчета:  в настройках электропривода 424.00mm/s  перевод в абсолютные единицы 42400steps/s  перевод единиц времени 42.400steps/ms  positionMainMax = 20000steps  positionMainMin = 100steps  speedAbsMainMax =  = (20000steps-100steps)/42.4steps/ms =  = 469ms |
| 36 | 4 | int32\_t | speedAbsMainPersonal | Абсолютная скорость основного механизма в режиме PERSONAL  По умолчанию равна 5% от speedAbsMainMax  Абсолютная скорость не может быть выше 100% и ниже 1% от speedAbsMainMax  Пример:  Если speedAbsMainMax = 469 то  speedAbsMainPersonal может быть в диапазоне 469 … 46900 |
|  | Всего 40 |  |  |  |

## TAG\_ReportMachineSettingsExtended

Сообщение высылается каждые 1000мс когда RTCU находится в режиме WAITING.

Если параметры тренажера еще не записаны в RTCU, то длина поля данных равна 0 байт.

Если параметры тренажера записаны в RTCU, то длина поля данных равна ?? байт.

Параметры тренажера находятся в энергонезависимой памяти, поэтому не стираются при выключении RTCU.

Таб.Х Параметры тренажера (поле данных сообщений TAG\_LoadMachineSettings*Extended* и TAG\_ReportMachineSettings*Extended*)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 4 | int32\_t | positionMainMax | См. TAG\_ReportMachineSettings |
| 4 | 4 | int32\_t | positionMainMin | См. TAG\_ReportMachineSettings |
| 8 | 4 | int32\_t | positionAux1Max | См. TAG\_ReportMachineSettings |
| 12 | 4 | int32\_t | positionAux1Min | См. TAG\_ReportMachineSettings |
| 16 | 4 | int32\_t | positionAux2Max | См. TAG\_ReportMachineSettings |
| 20 | 4 | int32\_t | positionAux2Min | См. TAG\_ReportMachineSettings |
| 24 | 4 | int32\_t | positionAux3Max | См. TAG\_ReportMachineSettings |
| 28 | 4 | int32\_t | positionAux3Min | См. TAG\_ReportMachineSettings |
| 32 | 4 | int32\_t | speedAbsMainMax | См. TAG\_ReportMachineSettings |
| 36 | 4 | int32\_t | speedAbsMainPersonal | См. TAG\_ReportMachineSettings |
| 40 | 4 | int32\_t | encoderBitCount | Битность энкодера  (версия2 = 12, версия2.1 = 15) |
| 44 | 4 | int32\_t | encoderDirection | Направление возрастания значений энкодера  Значения 1 или -1 |
| 48 | 4 | int32\_t | encoderOffset | Значение, добавляемое к значению энкодера |
| 52 | 4 | int32\_t | forceSensorOffset | Сдвиг датчика усилия |
| 56 | 4 | float | forceSensorGain | Коэффициент пересчета первичных значений датчика усилия в граммы |
|  |  |  |  |  |
|  |  |  |  |  |
| Всего 60 |  |  |  |  |

## TAG\_RfidProximity

Сообщение высылается каждые 100мс если бесконтактная метка находится в зоне видимости считывателя

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 1 | int32\_t | uidLength | Длина последующего UID |
| 4 | >=4 | uint8[] | uid | Байты UID |

## TAG\_RtcuDebugMessage

Сообщение переменной длины, высылается если в RTCU есть отладочные данные. Сообщение текстовое, кодировка ASCII, предназначено для вывода на экране компьютера, подключенного по диагностическому порту, или для сохранения в файл в HMI.

## TAG\_EnableServo

Переводит RTCU из режима IDLE в режим WAITING

Поле данных отсутствует

## TAG\_LoadPersonalSettings

RTCU принимает это сообщение когда находится в режиме WAITING.

Формат поля данных описан в сообщении TAG\_ReportPersonalSettings

## TAG\_LoadIsokineticExcerciseSettings

RTCU принимает это сообщение когда находится в режиме WAITING.

Формат поля данных описан в сообщении TAG\_ReportIsokineticExcerciseSettings

Если предварительно в RTCU не записаны персональные настройки (TAG\_LoadPersonalSettings) то сообщение TAG\_LoadIsokineticExcerciseSettings игнорируется

## TAG\_LoadGenericIsokineticExcerciseSettings

RTCU принимает это сообщение когда находится в режиме WAITING.

Формат поля данных описан в сообщении TAG\_Report*Generic*IsokineticExcerciseSettings

Если предварительно в RTCU не записаны персональные настройки (TAG\_LoadPersonalSettings) то сообщение TAG\_Load*Generic*IsokineticExcerciseSettings игнорируется

## TAG\_LoadMachineSettings

RTCU принимает это сообщение когда находится в режиме WAITING.

Формат поля данных описан в сообщении TAG\_ReportMachineSettings

Сообщение принимается только на диагностическом порту RTCU. Сообщение от HMI игнорируется.

## TAG\_LoadMachineSettingsExtended

RTCU принимает это сообщение когда находится в режиме WAITING.

Формат поля данных описан в сообщении TAG\_ReportMachineSettingsExtended

Сообщение принимается только на диагностическом порту RTCU. Сообщение от HMI игнорируется.

## TAG\_Parking

Переводит RTCU из режима WAITING в режим PARKING

Поле данных отсутствует

## TAG\_Personal

Переводит RTCU из режима WAITING в режим PERSONAL

Поле данных отсутствует

## TAG\_PersonalExit

Переводит RTCU из режима PERSONAL в режим WAITING

Поле данных отсутствует

## TAG\_PersonalButtonPressed

Уведомляет RTCU что кнопка персональных настроек была нажата

Таб.Х Формат поля данных для сообщений TAG\_PersonalButtonPressed / TAG\_PersonalButtonHold / TAG\_PersonalButtonReleased

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символичекое  имя | Описание |
| 0 | 1 | int8\_t | buttonID | Идентификатор кнопки, которая совершила действие  0 – основной механизм - инкремент  1 – основной механизм - декремент  2 – вспомогательный механизм №1 - инкремент  3 – вспомогательный механизм №1 - декремент  4 – вспомогательный механизм №2 - инкремент  5 – вспомогательный механизм №2 - декремент  6 – вспомогательный механизм №3 - инкремент  7 – вспомогательный механизм №3 - декремент |

**Вспомогательные механизмы:**

1 – регулировка спинки

2 – регулировка голени

## TAG\_PersonalButtonHold

Уведомляет RTCU что кнопка персональных настроек всё еще нажата

Формат поля данных описан в сообщении TAG\_PersonalButtonPressed

## TAG\_PersonalButtonReleased

Уведомляет RTCU что кнопка персональных настроек была отпущена

Формат поля данных описан в сообщении TAG\_PersonalButtonPressed

## TAG\_TestConcentric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 4 | int32\_t | pauseBeforeTest | Пауза (в миллисекундах) перед тестом |

## TAG\_TestEccentric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 4 | int32\_t | pauseBeforeTest | Пауза (в миллисекундах) перед тестом |

## TAG\_ExcerciseIsokinetic

Переводит RTCU из режима WAITING в режим EXERCISE\_ISOKINETIC

Поле данных отсутствует

## TAG\_GenericExcerciseIsokinetic

Переводит RTCU из режима WAITING в режим GENERIC\_EXERCISE\_ISOKINETIC

Поле данных отсутствует

## TAG\_ResetError

Сбрасывает ошибку и переводит RTCU в режим WAITING

Поле данных отсутствует

## TAG\_Cancel

Прекращает и переводит RTCU в режим WAITING

Поле данных отсутствует

## TAG\_TestStatic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 4 | int32\_t | pauseBeforeTest | Пауза (в миллисекундах) перед тестом |
| 4 | 4 | int32\_t | testDuration | Длительность теста (в миллисекундах) |
| 8 | 4 | int32\_t | positionRel | Относительное положение основного механизма во время теста (в процентах \* 100) |

## TAG\_ReportServoMode

Диагностическое сообщение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Смещение  [байт] | Размер  [байт] | Тип | Символическое имя | Описание |
| 0 | 4 | uint32\_t | mainTickID | Номер тика основного обработчика событий электропривода |
| 4 | 6 | struct | currentDateTime | Показания встроенных часов RTCU  struct TCurrentDateTime{  uint8\_t year;//00-99  uint8\_t month;//01-12  uint8\_t date;//01-31  uint8\_t hours;//00-23  uint8\_t minutes;//00–59  uint8\_t seconds;//00–59  }; |
| 10 | 1 | uint8\_t | modeBefore |  |
| 11 | 1 | uint8\_t | submodeBefore |  |
| 12 | 1 | uint8\_t | modeAfter |  |
| 13 | 1 | uint8\_t | submodeAfter |  |
|  |  |  |  |  |
|  |  |  |  |  |