**ОПИСАНИЕ ПРОТОКОЛА**

**Описание протокола двухплатника на платы с номером от 1 до 3 -х**

**Составил: Климов А.А.**

**Телеграмм канал:** [**https://t.me/kickscootergreenbytes**](https://t.me/kickscootergreenbytes)

**Чат обсуждения:** [**https://t.me/kickscootergreenbyteschat**](https://t.me/kickscootergreenbyteschat)

Параметры

Количество портов связи   
UART\_COUNT 2

Буфер памяти (переполнение)  
UART\_BUFFERSIZE 32

Скорость протокола  
UART\_SPEED 115200

Конфигурация пакета

Заголовок пакета  
DATAP\_HEADER\_PACK 0xF0

Размер пакета  
DATAP\_HEADER\_SIZE 12

Инструкции

Запрос датчиков АЦП  
DATAP\_INSTRUCTION\_ADC 0x01

Запрос обработанных параметров (входных)  
DATAP\_INSTRUCTION\_DEV 0x02

Запрос выходных параметров двигателя  
DATAP\_INSTRUCTION\_BLDC 0x03

Запрос выходных параметров подаваемого сигнала на ШИМ фетов  
DATAP\_INSTRUCTION\_MOSFET 0x04

Служебная инструкция  
DATAP\_INSTRUCTION\_DEBUG 0x05

Считать область памяти флеш (4 байта)  
DATAP\_INSTRUCTION\_READ\_CONFIG 0x06

Записать область памяти флеш (4 байта)  
DATAP\_INSTRUCTION\_WRITE\_CONFIG 0x07

Сбросить параметры устройства  
DATAP\_INSTRUCTION\_ERASE\_FLASH 0x08

Служебная инструкция  
DATAP\_INSTRUCTION\_TURN 0x09

Управление двигателем  
DATAP\_INSTRUCTION\_CONTROL 0x0A

Запрос информации о плате и прошивки  
DATAP\_INSTRUCTION\_VERSION 0x0B

Считать и обновить конфигурацию устройства  
DATAP\_INSTRUCTION\_UPDATE\_CONFIG 0x0C

Служебная инструкция  
DATAP\_INSTRUCTION\_PWM 0x0D

Структура пакета

Пример структуры передачи

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_ADC |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_ADC |
| Байт [1] | 0x00 |
| Байт [2] | 0x01 |
| Байт [3] | 0x00 |
| Байт [4] | 0x02 |
| Байт [5] | 0x00 |
| Байт [6] | 0x03 |
| Байт [7] | 0x00 |
| Байт [8] | 0x04 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Примеры запросов и ответов

Инструкция запроса АЦП  
DATAP\_INSTRUCTION\_ADC

Пример структуры передачи

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_ADC |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_ADC |
| Байт [1] uint16\_t ADC V\_BAT | 0x0001 |
| Байт [3] uint16\_t ADC CURRENT\_DC | 0x0002 |
| Байт [5] uint16\_t ADC THROTTLE | 0x0003 |
| Байт [7] uint16\_t ADC BRAKE | 0x0004 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Запрос обработанных параметров (входных)  
DATAP\_INSTRUCTION\_DEV

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_DEV |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_DEV |
| Байт [1] uint8\_t DEV devHall() | 0x01 |
| Байт [2] uint16\_t DEV devCurrent() | 0x0002 |
| Байт [4] uint16\_t DEV devVoltage() | 0x0003 |
| Байт [6] uint16\_t DEV devThrottle() | 0x0004 |
| Байт [8] uint16\_t DEV devBrake() | 0x0005 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Запрос выходных параметров двигателя  
DATAP\_INSTRUCTION\_BLDC

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_BLDC |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_BLDC |
| Байт [1] uint16\_t BLDC engineTurnover() | 0x0001 |
| Байт [3] uint16\_t BLDC engineDegree() | 0x0002 |
| Байт [5] uint16\_t BLDC engineCurrentPWM() | 0x0003 |
| Байт [7] uint8\_t BLDC engineReverse() | 0x0004 |
| Байт [8] uint8\_t BLDC PIN\_HALL\_A | 0x01 |
| Байт [9] uint8\_t BLDC PIN\_HALL\_B | 0x00 |
| Байт [10] uint8\_t BLDC PIN\_HALL\_C | 0x01 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Запрос выходных параметров подаваемого сигнала на ШИМ фетов  
DATAP\_INSTRUCTION\_MOSFET

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_MOSFET |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_MOSFET |
| Байт [1] uint16\_t MOSFET engineMosfet(A) | 0x0001 |
| Байт [3] uint16\_t MOSFET engineMosfet(B) | 0x0002 |
| Байт [5] uint16\_t MOSFET engineMosfet(C) | 0x0003 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Считать область памяти флеш (4 байта)  
DATAP\_INSTRUCTION\_READ\_CONFIG

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_READ\_CONFIG |
| Байт [1] uint32\_t CONFIG адрес флеш памяти \* | 0x00000000 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_READ\_CONFIG |
| Байт [1] uint32\_t CONFIG адрес флеш памяти \* | 0x00000000 |
| Байт [5] uint32\_t CONFIG считаное значение | 0x00000000 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

\* Адрес необходимо смотреть в даташите, пример и начало структур в Приложении А

Записать область памяти флеш (4 байта)  
DATAP\_INSTRUCTION\_WRITE\_CONFIG

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_WRITE\_CONFIG |
| Байт [1] uint32\_t CONFIG адрес флеш памяти | 0x00000000 |
| Байт [5] uint32\_t CONFIG значение для записи | 0x00000000 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_WRITE\_CONFIG |
| Байт [1] uint32\_t CONFIG адрес флеш памяти \* | 0x00000000 |
| Байт [5] uint32\_t CONFIG значение для записи | 0x00000000 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Сбросить параметры устройства  
DATAP\_INSTRUCTION\_ERASE\_FLASH

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_ERASE\_FLASH |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_ERASE\_FLASH |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Управление двигателем  
DATAP\_INSTRUCTION\_CONTROL

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_CONTROL |
| Байт [1] uint16\_t engineWrite() | 0x0000 |
| Байт [3] uint16\_t engineBrake() | 0x0000 |
| Байт [5] uint8\_t engineEnable() | 0x00 |
| Байт [6] uint8\_t Приоритет управления | 0x00 |
| Байт [7] uint8\_t Обратная связь | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_CONTROL |
| Байт [1] int16\_t devCurrent() | 0x0000 |
| Байт [3] uint16\_t devVoltage() | 0x0000 |
| Байт [5] uint16\_t engineTurnover() | 0x0000 |
| Байт [6] uint8\_t engineReverse() | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Запрос информации о плате и прошивки  
DATAP\_INSTRUCTION\_VERSION

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_VERSION |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_VERSION |
| Байт [1] uint8\_t CONFIG\_BOARD\_TYPE | 0x00 |
| Байт [2] uint32\_t FLASH\_ADRESS | 0x00000000 |
| Байт [6] uint32\_t FLASH\_MAX | 0x00000000 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Считать и обновить конфигурацию устройства  
DATAP\_INSTRUCTION\_UPDATE\_CONFIG

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_UPDATE\_CONFIG |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

Пример ответа

|  |  |
| --- | --- |
| Сигнатура | DATAP\_HEADER\_PACK |
| Байт [0] \ Инструкция | DATAP\_INSTRUCTION\_UPDATE\_CONFIG |
| Байт [1] | 0x00 |
| Байт [2] | 0x00 |
| Байт [3] | 0x00 |
| Байт [4] | 0x00 |
| Байт [5] | 0x00 |
| Байт [6] | 0x00 |
| Байт [7] | 0x00 |
| Байт [8] | 0x00 |
| Байт [9] | 0x00 |
| Байт [10] | 0x00 |
| CRС8 (От 0 до 10 байт) | uint8\_t CRC8(const uint8\_t \*addr, uint8\_t len); |

ПРИЛОЖЕНИЕ А

Структура настроек

typedef struct {

/\* Расположение фаз \*/

uint8\_t phase;

/\* Тип управления (ENGINE\_CONTROL\_PWM) \*/

uint8\_t control;

/\* Тип ШИМ подаваемый на двигатель (ENGINE\_PWM\_SINMOD) \*/

uint8\_t pwm;

/\* Установка градусность холлов (ENGINE\_HALL\_120) \*/

uint8\_t hall;

/\* Инерция двигателя (TRUE) \*/

uint8\_t inertion;

/\* Сторона движения двигателя \*/

uint8\_t reverse;

/\* Кол-во магнитов \*/

uint8\_t magnet;

/\* Ослабление поля (FALSE) \*/

uint8\_t fw;

/\* Максимальный угол опережения \*/

uint8\_t fwcorner;

/\* Максимальный ток ослабления \*/

uint16\_t fwcurrent;

/\* С каких оборотов начнет работать ослабление поля \*/

uint16\_t fwturn;

/\* Максимальный фазный ток в А \*/

uint16\_t phaseCurrent;

/\* Максимальный общий ток в А \*/

uint16\_t current;

/\* Максимальные обороты \*/

uint16\_t turnMax;

/\* Максимальный ток торможения \*/

uint16\_t brakeCurrent;

/\* Диаметр колеса \*/

uint16\_t wheelDiameter;

/\* Минимальная позиция аналогово задания в попугаях \*/

uint16\_t throttleMin;

/\* Максимальная позиция аналогово задания в попугаях \*/

uint16\_t throttleMax;

/\* Минимальная позиция тормоза в попугаях \*/

uint16\_t brakeMin;

/\* Максимальная позиция тормоза в попугаях \*/

uint16\_t brakeMax;

/\* Минимальное напряжение (отключает двигатель) \*/

uint16\_t voltageMin;

/\* Минимальное напряжение для информарования о разряде АКБ \*/

uint16\_t voltageMid;

/\* Максимальное напряжение АКБ, информирование о перезаряде \*/

uint16\_t voltageMax;

/\* Плавность задания аналогово задания \*/

uint8\_t throttlePID;

/\* Плавность срабатывания торможения \*/

uint8\_t breakPID;

/\* Количество попугаев на 1 Вольт \* \*/

uint16\_t bitVoltage;

/\* Количество попугаев на 1 Ампер \*\* \*/

uint16\_t bitCurrent;

uint8\_t bitByte;

uint32\_t bitData;

} config\_t;

ПРИЛОЖЕНИЕ Б

Процедура для формирования CRC8, формирование происходит по таблице от 0 до 10 байта включительно!

uint8\_t dpCRC8(const uint8\_t \*addr, uint8\_t len) {

uint8\_t i;

uint8\_t mix;

uint8\_t crc = 0;

uint8\_t inbyte;

while (len--) {

inbyte = \*addr++;

for (i = 8; i; i--) {

mix = (crc ^ inbyte) & 0x01;

crc >>= 1;

if (mix) crc ^= 0x8C;

inbyte >>= 1;

}

}

return crc;

}