

# Описание протокола A1DSP v1.0

Андрей Нечаев

2020

# Contents

<b>1</b>	<b>Вступление</b>	<b>2</b>
1.1	Возможности . . . . .	3
1.2	Ограничения . . . . .	3
<b>2</b>	<b>Описание</b>	<b>4</b>
2.1	Общие положения . . . . .	4
2.2	Совместимость . . . . .	5
2.3	Сигнатуры . . . . .	6
2.3.1	Служебные коды . . . . .	6
2.3.2	Служебные слова . . . . .	6
2.4	Диаграммы диалогов . . . . .	7
<b>3</b>	<b>Примеры</b>	<b>8</b>
3.1	Структура пакета . . . . .	8
3.2	Пример простого пакета без интеграции дополнительных данных . . . . .	8
3.3	Пример основной части пакета запроса автонастройки от хоста . . . . .	8
3.4	Пример основной части пакета ответа автонастройки от устройства . . . . .	8
3.5	Пример основной части пакета подтверждения автонастройки	8
3.6	Пример полного пакета . . . . .	9
3.7	Пример основной части пакета 2 . . . . .	9

# 1 Вступление

В процессе построения своего "умного дома" автор столкнулся с проблемой отсутствия простейшего протокола, канального уровня, который бы позволял производить не просто обмен данными, но и реализовывал обнаружение и настройку устройств. В виду этого, а также ввиду того, что некоторые найденные протоколы и технологии оказались либо слишком сложны в реализации, либо слишком плохо документированы, было принято решение написать простейший протокол для этих целей.

Всё это послужило предпосылкой для создания этого протокола канального уровня.

Перед тем как создавать этот протокол, были сделаны попытки реализации данной системы на основе существующих протоколов. Первоначально были рассмотрены и использованы такие протоколы и технологии как: "RS485, UART, 1-Wire, Ethernet, Wi-Fi, XML, Json, HTTP, MQTT". Однако отсутствие в них некоторого функционала приводило к попыткам нарастить его за счёт ухода во всё более высокие уровни абстракции и внедрению параллельно работающих протоколов частично зачастую дублирующих друг друга.

Сам по себе протокол строился не с чистого листа, но на базе протокола межблочного обмена внутри устройства между 2-мя микроконтроллерами по UART. Это было сделано как из желания сэкономить время, так и из понимания того, что на данный момент большинство устройств в умном доме всё-равно будут управляться микроконтроллерами, не способными на обработку больших объёмов информации и проведения сложных вычислений.

## 1.1 Возможности

Данный протокол создан с целью использования на максимальном спектре устройств, а также на максимально низком уровне абстракции (начиная с канального и выше). Он позволяет реализовывать как межблочное взаимодействие с заранее обозначенными параметрами, так и взаимодействие по принципу множества с одним. Возможное использование протокола при сильно стеснённых ресурсах достигается за счёт наличия режимов совместимости, ограничивающих максимальные размеры пакетов и их гибкой подстройки. Отсутствие зависимости в реализации протокола от канала передачи данных, и как следствие, универсальный парсер для любых каналов связи. Для совместимости с высокоуровневыми каналами связи предусмотрено дополнительное поле для записи туда адреса.

## 1.2 Ограничения

На канальном уровне поддерживаются следующие топологии:

- точка-точка
- шина
- звезда

На более высоких уровнях абстракции возможно использование данного протокола поверх всех протоколов, которые поддерживают или позволяют эмулировать данные топологии.

Вне зависимости от уровня абстракции при существовании центрального узла предполагается наличие у него аппаратных возможностей для полной поддержки основных алгоритмов согласования данного протокола.

Исходя из требований совместимости при первоначальной настройке предполагается аутентификация только со стороны хоста, устройство не проверяет хост на валидность, но предполагает, что хост не будет меняться после первоначальной настройки.

## 2 Описание

### 2.1 Общие положения

A1DSP (Где DSP означает Data in String Protocol) начинается с комбинации спецсимволов, состоит из основной части, оканчивающийся знаком ; с комбинацией спецсимволов, и дополнительных частей данных, также отделённых и оканчивающихся спецсимволами.

Основная часть состоит из строки с парами (название:данные), где разделителями между парами являются пробелы. Сама пара состоит из названия (используются строго заглавные латинские буквы в кодировке ASCII) не превышающего размер в 10 знаков, и данных в виде числа (целого, или с плавающей точкой), либо строки ограниченной одинарными кавычками. В основной части допускается экранирование кавычек в строке при помощи знака '\ ' (только часть данных).

Включение дополнительных пакетов с данными допускается при помощи их объявления в основной части спецсловами, и отправки сразу после окончания основной части. Каждому пакету данных присваивается номер, а также дополнительные поля для указания типа данных и наименования пакета данных.

Допускается отправка как зашифрованных, так и сырых данных. При отправке вне зависимости от типа данные подвергаются следующим операциям: снятие хеш суммы поступивших данных и кодирование посредством BASE64.

Главной задачей таких операций является обеспечение совместимости и контроль целостности передаваемых данных.

Сам по себе протокол, хоть и подразумевает возможности отправки зашифрованных данных, их шифрование не реализовывает. Вся задача по шифровке/дешифровке должна быть реализована сторонними алгоритмами.

При парсинге пакета на выходе получаются либо парные массивы (`mas_name[i] mas_data[i]`, где `i` порядковый номер), либо ассоциативный массив `mas[NAME1]` (по типу `map` в `c++`).

## 2.2 Совместимость

Размер численных переменных в простой версии протокола определяется за счёт их разрядности 8bit. Разрядность может быть пересогласована при помощи спецслова **A1BL:16** в запросе и ответе автонастройки, где 16 - новая разрядность.

**float** - для целочисленных

**int** - для чисел с плавающей точкой

**char** - для текстовых переменных не более 256 символов

Для большей совместимости предусмотрены несколько уровней.

### Уровни совместимости:

**0** - Макс. размер пакета 100, допданные отсутствуют, char[20]

**1** - Макс. размер пакета 100, допданные присутствуют, char[20]

**2** - Макс. размер пакета 1024, допданные отсутствуют

**3** - Макс. размер пакета 1024, допданные присутствуют

**4** - Макс. размер пакета не ограничен, допданные отсутствуют

**5** - Макс. размер пакета не ограничен, допданные присутствуют

**6-254** - Зарезервировано

Уровень совместимости согласовывается при первоначальной настройке устройства и задаётся служебным словом **DSPL\_0**, где **0** цифра уровня совместимости в запросе автонастройки. В случае отсутствия такового уровень совместимости устанавливается на **0**.

При отсутствии ограничений на размер пакета, установленных уровнем совместимости, они могут быть заданы при помощи спецслов.

**LEN\_MAX:'0'**

Установка макс. размера основной части, где 0 размер в байтах

**D\_MAX:'0'**

Установка макс. размера части данных, где 0 размер в байтах

## 2.3 Сигнатуры

### 2.3.1 Служебные коды

[0x3E][0x95] – Заголовок пакета

[0x3F][0x96] – Начало сегмента данных и разделитель сегмента данных

[0x3D][0x99] – Конец пакета

[0x3E][0x96] – Заголовок пакета запроса автонастройки

[0x3E][0x99] – Заголовок пакета ответа автонастройки

[0x3F][0x96] – Заголовок пакета подтверждения автонастройки

### 2.3.2 Служебные слова

A1BL:16 – Переопределение разрядности переменных, где 16 - новая разрядность

LEN\_MAX:0 – Макс. размер основной части, где 0 - размер в байтах

D\_MAX:0 – Макс. размер данных, где 0 - размер в байтах

DSPL\_0 – Уровень совместимости, где 0 - число в интервале 0-255

D0\_LABEL:'type' – Объявление наличия данных, где 0 номер пакета, LABEL - имя пакета, type - тип пакета

D0\_LEN:1 – Длина данных, где 0 - номер пакета, 1 - размер в байтах

D0\_CRC:'010' – Контрольная сумма данных, где 0 - номер пакета, 010 - контрольная сумма

HOST\_GUID: '\*' – ID хоста, где \* это ID в UUID формате

HOST\_DSTY: '\*' - Типы поддерживаемых хостом данных разделённые \_

HOST\_SADDR: '\*' – IP хоста, где \* это IP адрес

HOST\_KEY: '\*' – Часть ключа, передаваемая хостом

HOST\_PINRQSS:1 – Поддержка хостом подтверждения устройства по пину при автосогласовании

DEV\_GUID: '\*' – ID устройства, где \* это ID в UUID формате

**DEV\_DSTY:'\*\*'** - Типы поддерживаемых устройством данных разделённые \_

**DEV\_SADDR:'\*\*'** - IP устройства, где \* это IP адрес

**DEV\_KEY:'\*\*'** - где \* - ключ (смесь ID устройства и пина, если он поддерживается с обеих сторон)

**DEV\_PINRQS:1** - Требование устройством пина при автосогласовании, 1-да 0-нет

**DEV\_AUTH:1** - Успешность добавления устройства, 1-да, 0-нет

**CRC:'010'** - 010 - контрольная сумма

## 2.4 Диаграммы диалогов



Диаграмма автонастройки



## 3 Примеры

### 3.1 Структура пакета

[Код начала][Пробел][Основная часть];[Код окончания]  
[Код начала][Пробел][Основная часть];[Код отделения][Данные][Код  
окончания]

### 3.2 Пример простого пакета без интеграции дополнительных данных

[0x3E][0x95][SPACE]NAME1:123.02 NAME2:321.08 NAME3:'abc';[0x3D][0x99]

### 3.3 Пример основной части пакета запроса автонастройки от хоста

HOST\_GUID:'-' HOST\_KEY:'12345' HOST\_PINRQSS:'true' CRC:0101010

### 3.4 Пример основной части пакета ответа автонастройки от устройства

HOST\_GUID:'-' DEV\_GUID:'-' DEV\_KEY:'123' D1\_ACONF:'base64'  
DEV\_PINRQS:1 D2\_EACONF:'base64\_aes256' CRC:0101010

### 3.5 Пример основной части пакета подтверждения автонастройки

HOST\_GUID:'-' DEV\_GUID:'-' DEV\_AUTH:1 D1\_ACONF:'base64'  
D2\_EACONF:'base64\_aes256' CRC:0101010

### 3.6 Пример полного пакета

[0x3E][0x95]' '[Основная часть]';'[0x3F][0x96][DATA1][0x3F][0x96][DATA2][0x3D][0x

[MAIN STRING] - DEV\_GUID:" D1\_[LABEL]:'type' D2\_[LABEL]:'type'  
D1\_LEN:12415 D2\_LEN:2548 D1\_CRC:010101 D2\_CRC:010101  
CRC:0101010

DEV\_GUID:" D1\_[LABEL]:'base64' D2\_[LABEL]:'base64\_aes256'  
D1\_LEN:12415 D2\_LEN:2548 D1\_CRC:010101 D2\_CRC:010101  
CRC:0101010

Где:

D1, D2 - пакеты данных

D1 - публичная часть

D2 - закрытая часть

### 3.7 Пример основной части пакета 2

MQV:1.42 VMQ1:0.00 VMQ2:0.07 VIN:11.77 MCVCC:5.00 MCTMP:13.62  
LUX:279.00 DHUM:68.43 DTMP:11.23 VPRE:774.10 VTMP:22.02 RDY=1;  
MQV:1.42 //Напряжение датчиков MQ  
VMQ1:0.00 //Выходное напряжение датчика 1  
VMQ2:0.07 //Выходное напряжение датчика 2  
VIN:11.77 //Входное напряжение  
MCVCC:5.00 //Напряжение на шине 5В (atmega328)  
MCTMP:13.62 //Температура (atmega328)  
LUX:279.00 //Освещённость  
DHUM:68.43 //Влажность датчика (DHT21)  
DTMP:11.23 //Температура датчика (DHT21)  
VPRE:774.10 //Давление (BMP180)  
VTMP:22.02 //Температура датчика (BMP180)  
RDY:1 //Готовность усреднённых данных где 1-готовы, 0-нет