**Формат команд обмена Mega и ESP8266**

?параметр = значение

?команда

|  |  |  |  |
| --- | --- | --- | --- |
| Команды, передаваемые с модуля MEGA на модуль ESP | | | |
| **Полученная команда** | Описание | **Пояснение** | Примечание |
| **?mega=1** | - | Модуль mega посылает данные о своей работоспособности. |  |
| **?sendtemp=[floatX]** | Поступление температуры датчика номер X | Модуль mega посылает данные о температуре датчика Х (X =A, то всех датчиков по порядку) | Ответ модуля mega на команду **?reqesttemp=X** |
| **?sendtemp=[float1];[float2];….;** | Поступление температуры всех датчиков |
| **?sendtempХ=[floatX]** | Поступление температуры датчика номер X |  |  |
|  |  |  |  |
| **?sendTargetTemp=[float]** | Поступление текущей целевой температуры с учетом расписания |  | Ответ модуля mega на команду **?reqestTargetTemp** |
|  |  |  |  |
|  |  |  |  |
| **?reqestrssi** | Запрос уровня сигнала wi-fi | Ответ команда: **?sendrssi=х** |  |
| **?reqestLocalIP** | Запрос локального IP |  | WiFi.localIP() |
| **?test** | Тестовый запрос | Ответ команда: **?esplog=”Put command test”** | Тестовая команда |
| **?sendGTargetTemp=**g\_tRoomSetpoint | MEGA передает текущее значение целевой температуры системы | g\_tRoomSetpoint – глобальная переменная целевой температуры системы |  |
|  |  |  |  |
| **?sendSystemParameters=XYZK** | Mega передает значения текущих параметров системы | X – работы насоса котла;  Y – режим работы насоса системы;  Z – алгоритм регулирования температуры системы;  K – режим работы привода заслонки поддувала |  |
| **?sendrssi=х** | Получение от модуля esp информации об уровне wi-fi сигнала. |  |  |
|  |  |  |  |
| Команды, передаваемые с модуля ESP на модуль MEGA | | | |
| **Полученная команда** | Описание | **Пояснение** | Примечание |
| **?esp=1** | - | Модуль mega посылает данные о своей работоспособности. |  |
| **?esplog=отладочный вывод от ESP** | ESP8266 пересылает свой отладочный вывод |  | Выводим полученные данные в Serial |
| **?****reqesttemp=X** | Запрос на передачу температуры датчика Х (=A – всех датчиков по порядку) | Ответ команда: **?sendtemp=[floatX]** |  |
| **?****reqestTargetTemp** | Запрос текущей целевой температуры с учетом расписания |  |  |
|  |  |  |  |
| **?SetGTargetTemp=newT** | Установка значения глобальной целевой температуры (переменная g\_tRoomSetpoint) | **newT – строка, представляющая собой значение целевой температуры в виде десятичной дроби (запятая уже переведена в точку) в диапазоне 20.0 – 26.0** | В модуле mega используем atof(), без всяких проверок |
| **?setBoilerPumpMode=X** | Установка режима работы насоса ТТК |  | 1 - on, 2 - off, 3 - auto |
| **?setSystemPumpMode** | Установка режима работы насоса системы |  | 1 - on, 2 - off, 3 - auto |
| **?setSysTempControlMode** | Установка режима регулирования температуры в системе |  | 6 – мой алгоритм регулирования,  7 - PID регулятор |
| **?setDoorAirMode** | Установка режима работы привода заслонки поддувала. |  | 4 - open, 5 - close,  3 - auto |
| **?getSystemParameters** | Запрашиваем текущие параметры системы | ?sendSystemParameters=XYZK | X – работы насоса котла;  Y – режим работы насоса системы;  Z – алгоритм регулирования температуры системы;  K – режим работы привода заслонки поддувала |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Формат команд**

#|T|0|2|X – строка ASCII символов.

Формат **запроса.**

Состоит из 5 байт.

Байт 1 – признак сообщения,

0x23(#) – признак запроса ведущего,

Байт 2 – код команды.

0x54(T) – запрос температуры датчика DS18B20.

Байт 3 и 4 – параметры конкретной команды.

Байт 5 – контрольная сумма(CS).

Рассчитывается как исключающее ИЛИ всех байт: Б1^Б2^Б3^Б4^0x5E

Формат **ответа** устройства

Байт 1 – признак ответа,

0x24($) – признак ответа устройства;

Байт 2 – код принятой команды;

Байт 3 и 4 – параметры в принятой команде;

Байт 5 …. n – индивидульные байты ответа в зависимости от типа команды.

Байт n – CS.

1. **Получение значения температуры ( float передаем побайтно, 4 байта)**

Запрос: **[0x23][0x54][0x00][0x00][CS] (#T00CS),**

1 байт – 0x23 (#);

2 байт – 0x54 (T);

3 байт – номер датчика температуры;

4 байт – параметр (не используется).

5 байт – CS.

Ответ: [0x24][0x54][0x03][0x04][0x05][0x06][0x07][0x08] [CS] ($T000CS),

1 байт – 0x24 ($);

2 байт – 0x54 (T);

3 байт – номер датчика температуры;

4 байт – параметр, поступивший в запросе;

5 байт – 1й – 4й байты передают значение температуры в переменной float;

6 байт – 2й байт;

7 байт – 3й байт;

8 байт – 4й байт;

9 байт – CS.

1. **Получение значения температуры в формате float в ASCII кодировке**

Запрос: «#t00CS»

1 байт – 0x23 (#);

2 байт – 0x74 (t);

3 байт – номер датчика температуры;

4 байт – параметр (не используется);

5 байт – CS.

Ответ: «$t0025.67°C»

1 байт – 0x24 ($);

2 байт – 0x74 (t);

3 байт – номер датчика температуры;

4 байт – параметр, поступивший в запросе;

остальные байты значение температура и знак °C (0xBO, 0x43).

1. **Получение списка адресов всех подключенных к arduino датчиков температуры**

Запрос: [0x23][0x44][0x00][0x00][CS] (#D00‘CS’),

1 байт – 0x23 (#);

2 байт – 0x44 (D);

3 байт – параметр (не используется);

4 байт – параметр (не используется);

5 байт – CS.

Ответ: Несколько групп 8 байт.

1. **Получение информации об ошибке**

Запрос: [0x23][0x45][CS] (#ECS),

Ответ: [0x24][0x44][8байт\*n][0x44][CS] ($T000CS),

3 байт – номер цифрового порта к которому подключен датчик DS18B20

4-11 байты – восьмибайтный адрес первого найденного устройства,

Предпоследний байт 0x44 – окончание списка адресов подключенных датчиков температуры. По данному байту осуществляется контроль окончания передачи адресов.

Описание системы.

1. Параметры:
   1. Датчики температуры:

№ 1 – датчик ТТК. Адрес: {0x28, 0xB8, 0x45, 0xA3, 0x09, 0x00, 0x00, 0x2C}

№ 2 – датчик ЭК. Адрес: {0x28, 0x59, 0x25, 0xA3, 0x09, 0x00, 0x00, 0x23}

№ 3 – датчик внутри дома

№ 4 – датчик на улице



Коды ошибок:

1. 0x01 - ошибка CRC в поступившем адресе термометра в функции int get\_temperatur\_ds18b20(const byte \*addr, const int pin);
2. 0x02 - ошибка reset сети 1-Wire в функции int get\_temperatur\_ds18b20(const byte \*addr, const int pin);