

## ОПИСАНИЕ ПРОГРАММНОЙ ЧАСТИ

- 1 Конфигурация (Виктор Уфа). Старт стоп Компрессор скролл EVI с шаговым ЭРВ и РТО . Доп опция теплоаккумулятор (пока реализация только на уровне данных).
- 2 Конфигурация (sheeny) Воздушный Старт стоп с шаговым ЭРВ, РТО и датчиком давления испарителя
- 3 Конфигурация (dimex)
- 4 Конфигурация (dobrinia) инвертор+ЭРВ + с РТО и датчиком давления испарителя
5. Конфигурация rav2000 инвертор
6. Конфигурация vad7 Частотник Vacon, 3 фазы, ЭРВ, РТО, 2 датчика давления

### Датчики

### Конфигурации

№	Имя	Описание	PIN	Диапазон	Датчик /протокол	1	2	3	4
1	TOUT	Температура улицы	D23	-40 +40	DS18B20 OneWire	+	+		+
2	TIN	Температура в доме	D23	-30 +40	DS18B20 OneWire	+	+	+	+
3	TEVAIN	Температура на входе испарителя (по фреону)	D23	-50 +12	DS18B20 OneWire	+	+	+	+
4	TEVAOUT	Температура на выходе испарителя (по фреону)	D23	-30 +12	DS18B20 OneWire	+	+	+	+
5	TCONIN	Температура на входе конденсатора (по фреону)	D23	-5+120	DS18B20 OneWire	+	+	+	+
6	TCONOUT	Температура на выходе конденсатора (по фреону)	D23	-5+60	DS18B20 OneWire	+	+	+	+
7	TBOILER	Температура в бойлере ГВС	D23	0 +60	DS18B20 OneWire	+	+	+	+
8	TACCUM	Температура на выходе теплоаккумулятора	D23	-15 +50	DS18B20 OneWire	+	+		-
9	TRTOOUT	Температура на выходе RTO (по фреону)	D23	-30 +12	DS18B20 OneWire		+		+
10	TCOMP	Температура нагнетания компрессора	D23		DS18B20 OneWire	+			+
11	TEVAING	Температура на входе испарителя (по гликолю)	D23		DS18B20 OneWire	+			
12	TEVAOUTG	Температура на выходе испарителя (по гликолю)	D23		DS18B20 OneWire	+			
13	TCONING	Температура на входе конденсатора (по гликолю)	D23		DS18B20 OneWire	+			
14	TCONOUTG	Температура на выходе конденсатора (по гликолю)	D23		DS18B20 OneWire	+			
15	PEVA	Датчик давления испарителя.	AD11	0 – 15 бар	<a href="#">Датчик давления</a>	+	+	+	+
16	PCON	Датчик давления нагнетания	AD10	0 – 15 бар		-	-	-	+
16	SEVA	Датчик протока по	D28		Сухой	+			-

		испарителю			контакт ???				
17	SLOWP	Датчик низкого давления	D29		Сухой контакт ???	+			-
18	SHIGHP	Датчик высокого давления	D30		Сухой контакт ???	+			-
19	SFROZEN	Датчик заморозки (это не аварийный датчик)? Необходим для воздушника	D31		Сухой контакт ???		+		
19	rs485	Чтение показаний счетчика SDM120	uart	modbus				+	+

## Исполнительные устройства

## Конфигурация

№	Имя	Описание	Присоединение	PIN	Характеристики	1	2	3	4
1	RCOMP	Реле включения компрессора (через пускатель)	1 цифровой выход	D46	Реле 220 В 10А	+	+		-
2	RPUMPI	Реле включения насоса входного контура (геоконтур)	1 цифровой выход	D47	Реле 220 В 10А	+		+	+
3	RPUMPO	Реле включения насоса выходного контура (отопление и ГВС)	1 цифровой выход	D48	Реле 220 В 10А	+	+	+	+
4	RBOILER	Включение ТЭНа бойлера.	1 цифровой выход	D49	Реле 220 В 10А	+	+	+	+
5	RTRV	4 ходовой клапан	1 цифровой выход	D50	Реле 220 В 10А	+	+		+
6	RFAN1	Реле включения вентилятора испарителя №1	1 цифровой выход	D44	Реле 220 В 10А		+		-
7	RFAN2	Реле включения вентилятора испарителя №2	1 цифровой выход	D45	Реле 220 В 10А		+		-
8	R3WAY	Переключение системы СО — ГВС (что сейчас греть)	1 цифровой выход	D52	Реле 220 В 10А	+			-
9	REVI	Соленойд для EVI. (испаритель ниже +3гр и конденсатор выше +40гр)	1 цифровой выход	D51	Реле 220 В 10А	+			-
10	RHEAT	Включение ТЭНа СО (электрокотел), может использоваться как догрев, резерв и т.д.	1 цифровой выход	D43	Реле 220 В 10А	-			+
11	RPUMPB	Насос циркуляции горячей воды.	1 цифровой выход	D53	Реле 220 В 10А	+			+

12	RPUMPFL	реле насоса Теплого Пола	D11						
12	EEV	Электронный регулирующий вентиль 480 шагов.	4 цифровых выхода 0-12В 0,5 А	D24 D25 D26 D27	Дополнительная плата управления ШГ	+	+		+
13	INVOUT	ЦАП	1 аналоговый выход, ограниченная поддержка	DAC0	0...32 кГц			+	+
14	Modbus	Обмен с инвертором OMRON MX2 и счетчиком SDM120/630	485	Serial 2		+			

12 — INVERT - Частота управляющих импульсов: 0,5 ...32 кГц ( можно задать уровень 100% оборотов в частотнике в Гц-ах); Скважность: 30 ... 70 %; Уровень «1»: 3,5 ... 13,2 В; Уровень «0»: 0,0 ... 0,8 В;

### Аjax запросы (список по мере продвижения проекта будет пополняться)

Таблица ниже показывает назначение и синтаксис запросов от веб морды к контроллеру. Столбец ОК - «+» уже реализовано в прошивке и можно пользоваться. Все строки заканчиваются 0 символом, указана максимальная длина строк. Для длинных строк желательно предусмотреть перенос.

Каждый запрос начинается со "спец" символа – «&», который служит идентификатором запроса Ajax. Конец запроса обозначается двойными символами «&&». Все что идет после «&&» отбрасывается. Без этого работать не будет. Заголовок «GET /&далее\_тело\_запроса&&\_не\_обрабатывается\_хвост»

Запросы можно объединять в одну посылку например «&get\_Temp(TIN)&get\_Temp(TOUT)&&»

В ответ на запрос контроллер возвращает «Запрос=Значение». При объединенном запросе приходит один объединенный ответ. Для вышеуказанного запроса ответ будет «&get\_Temp(TIN)=23.6&get\_Temp(TOUT)=-12.5&&»

Если в запросе допущена ошибка (синтаксис и логика) вместо значения возвращается код ошибки. Сейчас определены следующие ошибки:

E01 — Нет такого запроса (имя запроса), запрос либо не правильно написан или не поддерживается.

E02 — Нет такого параметра (имя параметра)

E03 — Не соответствие параметра и запроса (например - get\_testPress (TIN)).

E04 — Значения устанавливаемого параметра не валидно - синтаксис (например - set\_testPress (TIN=a.1)).

E05 — Значения устанавливаемого параметра выходит за диапазон (например - set\_testPress (TIN=-30)).

E06 — использование имя устанавливаемого параметра «здесь» запрещено, пример — set\_targetTemp(TOUT=23) – установка целевой температуры улицы не предусмотрена.

E07 - строка ответа на запрос очень длинная, запрос обрезан.

E08 - устанавливаемый номер выходит за диапазон номеров при сканировании OneWire. На пример при сканировании найдено 6 датчиков (разрешенные значения 1-6), а пытаетесь установить 7 номер — будет получена ошибка E08. 0- значение разрешено и означает сброс датчика.

E09 — в функции set\_BOILERMODE не правильно указан индекс режима (допустимые значения 0-4).

E10 — ошибка имени параметра в запросе ЭРВ

E11 — выход за диапазон значений для функции set\_paramEEV.

E12 — выход за диапазон значений для функций set\_EEV и set\_FC

E13 — ошибка имени параметра в запросе FC

E14 — выход за диапазон значений для функции set\_paramFC.

E15 — ошибка преобразования строки в сетевой параметр set\_Network

E16 – ошибка установки параметра в функции set\_paramCoolHP set\_paramHeatHP  
 E17 – ошибка установки параметра в функции set\_optionHP, возможно выход за диапазон  
 E18 - ошибка установки параметра в функции set\_datetime, возможно выход за диапазон  
 E19 - ошибка установки параметра в функции set\_Boiler, возможно выход за диапазон  
 E20 - ошибка установки параметра в функции set\_Message, возможно выход за диапазон  
 E21 - ошибка удаленного датчика - не верный формат строки  
 E22 - ошибка удаленного датчика - ошибка преобразования строки в число  
 E23 - ошибка удаленного датчика - номер датчика выходит за разрешенный диапазон 1-IPNUMBER. Нумерация начинается с 1  
 E24 - ошибка удаленного датчика - значение параметра датчика выходит за разрешенный диапазон, например напряжение питания -2 вольта  
 E25 - ошибка удаленного датчика - текущая прошивка не поддерживает удаленные датчики  
 E26 - ошибка удаленного датчика — не верный параметр get\_sensorIP  
 E27 – ошибка установки параметра в функции set\_paramFC  
 E28– ошибка установки параметра в функции set\_Profile  
 E29 – не верный номер профиля (выход за диапазон)  
 E30 — попытка стереть текущий профиль  
 E31 - ошибка установки параметра в функции set\_SDM  
 E32 - ошибка установки параметра в функции set\_MQTT  
 E33 - не верный номер расписания,  
 E34 - не хватает места для календаря  
 E35 — не верное значение теплоемкости теплоносителя для датчика протока (допустимо 0...5000)  
 Количество объединенных запросов ограничивается объемом выходного буфера (ответа — запрос обычно короче) который установлен 2048 байт (аппаратное ограничение чипа w5200), но программно буфер расширен до 3\*2048 байт. Это позволяет получить за один запрос 50-120 параметров. При превышении размера ответа, объединенный запрос обрезается и в последнем запросе ставится ошибка E07.

Запрещено использование в ответах (это могут быть длинные строки с описанием) символов «&» «=».

Лишние символы в запросах запрещены, в том числе пробелы и будут возвращаться ошибки.  
 Есть запросы на получение информации — обычно начинаются с префикса «get», они обычно имеют один входной параметр.

Есть запросы на установку значений - обычно начинаются с префикса «set» Для установки значения используется следующая конструкция Запрос - «set\_testTemp(TIN=25.6)» ответ «set\_testTemp(TIN)=25.6» если значение не валидно то в ответ передается ошибка E04 - E06 и запрос игнорируется. Дробная часть отделяется точкой «.», при использовании запятой будет возникать ошибка E04 . Возвращается установленное (для ответа используется функция чтения параметра) значение или код ошибки.

Имя запроса	Параметр	Описание	Пример	ОК
<b>Определение конфигурации (списки датчиков и исполнительных устройств)</b>				
get_listTemp	нет	Получить список датчиков температуры разделитель «;» Выводятся только представленные датчики	get_listTemp возврат TOUT;TIN;TEVAIN;TEVAOUT;TBOILER;TACCUM;TRTOOUT;TCOMP;TEVAING;TEVAOUTG;TCONING;TCONOUTG;	+
get_listPress	нет	Получить список аналоговых датчиков разделитель «;» Выводятся только представленные датчики	get_listPress возврат PEVA;	+
get_listInput	нет	Получить список цифровых датчиков разделитель «;» Выводятся только	get_listInput возврат SLOWP;SHIGHP;SERRMX	+

		представленные датчики	2;	
get_listRelay	нет	Получить список реле разделитель «;» Выводятся только представленные реле	get_listRelay возврат RCOMP;RPUMPI;RPUMPO ;RBOILER;RTRV;R3WAY;R HEAT;RPUMPB;RRESET;	+
get_listFlow	нет	Получить список частотных датчиков «;» Выводятся только представленные датчики	get_listFlow возврат FLOWCON; FLOWEVE;	+
<b>Датчики</b>				
get_Temp	Имя датчика	Получение температуры с датчика по его имени. <b>Если есть удаленные датчики, включается их обработка!!!. Без удаленного датчика значение будет совпадать с get_rawTemp</b> Возвращает строку (маx 5+1 символов) температура до десятых градуса В режиме DEMO выдаются случайные значения. Это основная функция.	get_Temp(TIN) возврат 23.4	+
get_rawTemp	Имя датчика	Получение температуры с проводного датчика по его имени. <b>Это данные ТОЛЬКО с проводного датчика</b> Возвращает строку (маx 5+1 символов) температура до десятых градуса В режиме DEMO выдаются случайные значения Используется для вывода результатов сканирования шины.	get_rawTemp(TIN) возврат 23.4	+
get_fullTemp	Имя датчика	Получение температуры проводного датчика + в скобках [] температура с учетом удаленного датчика. В скобках — используемая температура контроллером (get_Temp). Возвращает строку (маx 12+1 символов) температура до десятых градуса В режиме DEMO выдаются случайные значения Если привязки удаленного датчика нет то вывод аналогичен get_Temp (т. е. Без скобок) Функция эквивалентна двум вызовам: get_rawTemp_[get_Temp] Используется на странице датчики.	get_fullTemp(TIN) возврат 23.4 [23.1]	+
get_minTemp	Имя датчика	Получение минимально допустимой температуры с датчика по его имени. Нижняя граница диапазона. Возвращает строку (маx 5+1 символов) температура до десятых градуса	get_minTemp(TIN) возврат -30.0	+
get_maxTemp	Имя датчика	Получение максимально допустимой температуры с датчика по его имени. Верхняя граница диапазона. Возвращает строку (маx 5+1 символов) температура до десятых градуса	get_maxTemp(TIN) возврат 40.0	+
get_adressTemp	Имя датчика	Получение адреса датчика на шине oneWire. Адрес занимает 8 байт. Возвращает строку из	get_adressTemp(TIN) возврат	+

		16+1 символов (шестнадцатеричный вид 6 байт)	AABBCC1122334455	
set_adressTemp	Имя датчика = Новое значение	Установка адреса конкретного датчика температуры по номеру из списка полученного по запросу scan_OneWire. Возвращает установленный адрес (строка 16 символов) или ошибку. Длина 8+1 байт. Если номер из списка не валидный то возвращает ошибку. Номера начинаются с 1, 0 — означает сброс адреса (ставится флаг что адрес не установлен и значение самого адреса устанавливается все 0) E08 и ничего не делает.	set_adressTemp(TIN=2) возврат AABBCC1122334455	+
get_OneWirePin	нет	Получение номера ножки куда присоединены датчики температуры DS18B20. В начале добавляется символ «D» Нога едина для ВСЕХ датчиков температуры. Длина 3+1.	get_OneWirePin возврат D45	+
scan_OneWire	нет	Сканирование шины OneWire. Предназначена для определение адресов температурных датчиков ds18b20. Возвращает строку вида «Номер1:Тип1:Температура1:Адрес1;Номер2:Тип2:Температура2:Адрес2;» и так далее по всем найденным датчикам. Определены четыре типа датчиков DS18S20,DS18B20,DS1822, unknown <b>Посылка этого запроса приводит к остановке работы ТН.</b>	scan_OneWire возврат 1:18b20:23.5:AABBCC112233; 2:18s20:-6.0:AABBCC000100; 3:1822:56.1:AABBCC002200; 4:18b20:-12:AABBCC220011;	+
get_testTemp	Имя датчика	Получение температуры с датчика в режиме теста. Возвращает строку (мах 5+1 символов) температура до десятых градуса	get_testTemp(TIN) возврат 23.4	+
set_testTemp	Имя датчика = Новое значение	<u>Установка</u> температуры с датчика в режиме теста. Возвращает строку (мах 5+1 символов) температура до десятых градуса	set_testTemp(TIN=21.3) возврат 21.3	+
get_presentTemp	Имя датчика	Наличие датчика в конфигурации. Присутствует - «0», отсутствует «1». Строка два байта.	get_presentTemp(TIN) возврат 1	+
get_noteTemp	Имя датчика	Возвращает описание датчика. Строка до 80+1 символов	get_noteTemp(TIN) возврат Температура в доме	+
get_errcodeTemp	Имя датчика	Возвращает код ошибки. Строка до 3+1 символов. 0 — окей все ошибки значения меньше 0	get_errcodeTemp(TIN) возврат 0	+
get_errTemp	Имя датчика	Возвращает значение систематической ошибки датчика в градусах с точностью десятые. Она ДОБАВЛЯЕТСЯ к значениям датчика. Возвращает строку (мах 5+1 символов)	get_errTemp(TIN) возврат -0.3	+
set_errTemp	Имя датчика = Новое	<u>Устанавливает</u> значение систематической ошибки датчика в градусах с точностью десятые. Она ДОБАВЛЯЕТСЯ к значениям датчика. Возвращает строку (мах 5+1	set_errTemp(TIN=0.3) возврат -0.3	+



	значение	символов)		
get_Press	Имя датчика	Получение давления с датчика по его имени. Возвращает строку (максимум 3+1 символ) Давление в барах	get_Press (PEVA) возврат 3.45	+
get_adcPress	Имя датчика	Получение последнего считанного значения АЦП в отсчетах. Возвращает строку (максимум 3+1 символ)	get_adcPress(PEVA) возврат 1234	
get_minPress	Имя датчика	Получение минимально допустимого давления с датчика по его имени. Нижняя граница диапазона. Возвращает строку (максимум 3+1 символ) Давление в барах	get_minPress(PEVA) возврат 0.30	+
get_maxPress	Имя датчика	Получение минимально допустимого давления с датчика по его имени. Нижняя граница диапазона. Возвращает строку (максимум 3+1 символ) Давление в барах	get_maxPress(PEVA) возврат 5.00	+
get_pinPress	Имя датчика	Получение номера пина куда прикреплен датчик. Возвращает строку из 3+1 символа (первая позиция буква А)	get_pinPress(PEVA) возврат A2	+
get_testPress	Имя датчика	Получение давления с датчика в режиме теста. Возвращает строку (максимум 3+1 символ) Давление в барах	get_testPress(PEVA) возврат 3.00	+
set_testPress	Имя датчика = Новое значение	<u>Установка</u> давления с датчика в режиме теста. Возвращает строку (максимум 3+1 символ) Давление в барах	set_testPress (PEVA=3.30) возврат 3.30	+
get_presentPress	Имя датчика	Наличие датчика в конфигурации. Присутствует - «1», отсутствует «0». Строка два байта (1=1).	get_presentPress (PEVA) возврат 1	+
get_notePress	Имя датчика	Возвращает описание датчика. Строка до 80+1 символа	get_notePress (PEVA) возврат Датчик давления испарителя	+
get_errcodePress	Имя датчика	Возвращает код ошибки. Строка до 3+1 символа. 0 — окей все ошибки значения меньше 0	get_errcodePress (PEVA) возврат 0	+
get_zeroPress	Имя датчика	Возвращает значение “0” датчика в отсчетах АЦП. Возвращает строку (максимум 5+1 символ)	get_zeroPress (PEVA) возврат 345	+
set_zeroPress	Имя датчика = Новое значение	<u>Устанавливает</u> значение “0” датчика в отсчетах АЦП. Возвращает строку (максимум 5+1 символ)	set_zeroPress (PEVA=45) возврат 45	+
get_transPress	Имя датчика	Возвращает значение коэффициент преобразования (значение делится на него). Возвращает строку (максимум 4+1 символ) . Может преобразовать в float	get_transPress (PEVA) возврат 34	+

set_transPress	Имя датчика = Новое значение	<u>Устанавливает</u> значение коэффициент преобразования (значение делится на него). Возвращает строку (максимум 4+1 символ) . Может переделано в float	set_transPress (PEVA=31) возврат 31	+
get_Input	Имя датчика	Возвращает состояние цифрового датчика, если на входе 1 то возвращает 1, 0 в противном случае	get_Input(SEVA) возврат 0	+
get_presentInput	Имя датчика	Наличие датчика в конфигурации. Присутствует - «1», отсутствует «0». Строка два байта (1+1). Все остальное будет ошибкой	get_presentInput(SEVA) возврат 0	+
get_noteInput	Имя датчика	Возвращает описание датчика. Строка до 80+1 символов	get_noteInput(SEVA) возврат Датчик низкого давления	+
get_testInput	Имя датчика	Получение состояния датчика в режиме теста	get_testInput(SEVA) возврат 0	+
set_testInput	Имя датчика = Новое значение	<u>Установка</u> состояния датчика в режиме теста. Разрешены ТОЛЬКО два значения 0 и 1 – все остальное будет ошибкой.	set_testInput(SEVA=0) возврат 0	+
get_alarmInput	Имя датчика	Получение состояния датчика в режиме аварии, если значение датчика становится таким то высказывает ошибка	get_alarmInput(SEVA) возврат 1	+
set_alarmInput	Имя датчика = Новое значение	<u>Установка</u> состояния датчика в режиме аварии, если значение датчика становится таким то высказывает ошибка	set_alarmInput(SEVA=0) возврат 0	+
get_errcodeInput	Имя датчика	Возвращает код ошибки. Строка до 3+1 символов. 0 — окей все ошибки значения меньше 0	get_errcodeInput(SEVA) возврат 0	+
get_typeInput	Имя датчика	Получение типа датчика возвращает: «Alarm» - аварийный «Work» - рабочий «Pulse» - импульсный «none» - отсутствует Возвращает строку из 8+1 символов при не верном типе возвращает «err_type»	get_typePress(SEVA) возврат Alarm	+
get_pinInput	Имя датчика	Получение номера пина куда прицеплен датчик. Возвращает строку из 3+1 символов (первая позиция буква D)	get_pinPress(SEVA) возврат D2	+
<b>Исполнительные устройства</b>				
get_Relay	Имя реле	Возвращает состояние реле, для удобства вывода возвращается “1” “0” строки. Длина 3+1	get_Relay(RCOMP) возврат 1	+
set_Relay	Имя реле	Устанавливает состояние реле, 1 - включено 0	set_Relay(RCOMP=0)	+



	= Новое значение	— выключено. Возвращает новое состояние в кодировке 1/0	возврат 0	
get_presentRelay	Имя реле	Наличие реле в конфигурации. Присутствует - «1», отсутствует «0». Строка два байта (1+1).	get_presentRelay(RCOMP) возврат 0	+
get_noteRelay	Имя реле	Возвращает описание реле. Строка до 80+1 символов	get_noteRelay(RCOMP) возврат Реле включения компрессора (через пускатель)	+
get_pinRelay	Имя реле	Получение номера пина куда прицеплено реле. Возвращает строку из 3+1 символов (первая позиция буква D)	get_pinRelay(RCOMP) возврат D37	+
get_EEV	нет	Получение абсолютной позиции шагового двигателя EEV в шагах. Строка 3+1 Если идет движение шаговика значение берется в кавычки <<xxx>>	get_EEV возврат <<345>>	+
set_EEV	нет	<u>Установка</u> абсолютной позиции шагового двигателя EEV в шагах. Строка 4+1. Врзват – значение которые устанавливается если оно валидно.	set_EEV(200) возврат 200	+
set_zeroEEV	нет	<u>Установка</u> абсолютной позиции шагового двигателя EEV в «0». Строка 3+1	set_zeroEEV возврат 0	+
get_noteEEV	нет	Получение описания EEV. Строка 80+1	get_noteEEV возврат Электронно регулируемый вентиль	+
get_nameEEV	нет	Получение имени EEV. Строка 10+1. запрос сделан для единообразия.	get_nameEEV возврат EEV	+
get_minEEV	нет	Возвращает минимальное число шагов для открытия ЭРВ. Длина 3+1	get_minEEV возврат 66	+
get_maxEEV	нет	Возвращает максимальное число шагов ЭРВ. Длина 3+1	get_maxEEV возврат 480	+
get_presentEEV	нет	Наличие ЭРВ в конфигурации. Присутствует - «1», отсутствует «0». Строка два байта (1+1).	get_presentEEV возврат 1	+
get_errcodeEEV	нет	Возвращает код ошибки. Строка до 3+1 символов. 0 — окей все ошибки значения меньше 0	get_errcodeEEV возврат 0	+
get_pinEEV	нет	Получение пинов куда прицеплен контроллер шагового двигателя. Возвращает строку в котором указаны через пробел ноги	get_pinERV возврат D12 D13 D14 D15	+
get_overheatEEV	нет	Получить текущий перегрев, с сотыми градуса. Алгоритм вычисления определяется в зависимости от установок set_ruleEEV().	get_overheatEEV возврат 1.46	+

		При использовании «T[PEVA]-TEVAIN» перегрев определяется с использованием датчика давления, во ВСЕХ остальных случаях как разность температур испарителя.		
get_paramEEV	Имя параметра	Получить параметр ЭРВ. Разрешены следующие имена: T_OWERHEAT — Целевой перегрев TIME_IN — Постоянная интегрирования времени в секундах ПИД ЭРВ K_PRO - Пропорциональная составляющая ПИД ЭРВ K_IN — Интегральная составляющая ПИД ЭРВ K_DIF - Дифференциальная составляющая ПИД ЭРВ M_STEP – Число шагов открытия ЭРВ для правила работы ЭРВ «Manual» CORRECTION – поправка в градусах для правила работы ЭРВ «TEVAOUT-TEVAIN» и T[PEVA]-TEVAOUT.	get_paramEEV(T_OWERHEAT) возврат 2.52	+
set_paramEEV	Имя параметра = значение	<u>Установить</u> параметр ЭРВ. Разрешены следующие имена: T_OWERHEAT — Целевой перегрев TIME_IN — Постоянная интегрирования времени в секундах ПИД ЭРВ K_PRO - Пропорциональная составляющая ПИД ЭРВ K_IN — Интегральная составляющая ПИД ЭРВ K_DIF - Дифференциальная составляющая ПИД ЭРВ M_STEP – Число шагов открытия ЭРВ для правила работы ЭРВ «Manual» CORRECTION – поправка в градусах для правила работы ЭРВ «TEVAOUT-TEVAIN» и T[PEVA]-TEVAOUT	set_paramEEV(T_OWERHEAT=3.30) возврат 3.30	+
get_freonEEV	нет	Получить тип фреона которая используется в ТН. Возвращает СТРОКУ. Разрешенные значения: 0 R22=0, 1 R410A, 2 R600A, 3 R134A, 4 R407C, 5 R12, 6 R290, 7 R404A, 8 R717, Возвращает СТРОКУ. вида R22:0;R410A:0;R600:0;R600A:1;R134A:0;R407C:0;R12:0;R290:0;R404A:0;	get_freonEEV возврат R22:0;R410A:1;R600:0; R600A:0;R134A:0;R407C:0;R12:0;R290:0;R404A:0;	+
set_freonEEV	Новое значение	<u>Установить</u> тип фреона которая используется в ТН. На входе ЧИСЛО (от 0 до 11 — тип фреона) Возвращает СТРОКУ от 0 до 11. Разрешенные значения: 0 R22=0, 1 R410A, 2 R600A, 3 R134A, 4 R407C, 5 R12,	set_freonEEV(1) возврат R22:0;R410A:1;R600:0; R600A:0;R134A:0;R407C:0;R12:0;R290:0;R404A:0;	+

		6 R290, 7 R404A, 8 R717 Возвращает СТРОКУ. вида R22:0;R410A:0;R600:0;R600A:1;R134A:0;R407 C:0;R12:0;R290:0;R404A:0;		
get_ruleEEV	нет	Получить правило работы ЭРВ. Разрешенные значения: 0 - «TEVAOUT-TEVAIN» 1 - «TRTOOUT-TEVAIN» 2 - «TEVAOUT -T[PEVA]» 3 - «TRTOOUT T[PEVA]» 4 - «Table[EVA CON]» 5 – “Manual” Возвращает СТРОКУ. вида TEVAOUT- TEVAIN:1:1;TRTOOUT-TEVAIN:0:0;TEVAOUT- T[PEVA]:0:1;TRTOOUT-T[PEVA]:0:0;Table[EVA CON]:0:1;Manual:0:1; где ИМЯ:ВЫБОР:РАЗРЕШЕННОСТЬ_ВЫБОРА	get_ruleEEV возврат TEVAOUT- TEVAIN:1:1;TRTOOUT- TEVAIN:0:0;TEVAOUT- T[PEVA]:0:1;TRTOOUT- T[PEVA]:0:0;Table[EVA CON]:0:1;Manual:0:1;&&	+
set_ruleEEV	Новое значение	<u>Установить</u> правило работы ЭРВ. Параметр ЧИСЛО от 0 до 3. Разрешенные значения: 0 - «TEVAOUT-TEVAIN» 1 - «T[PEVA]-TEVAOUT» 2 -. «Table[EVA CON]» 3 – “Manual” Возвращает СТРОКУ. вида TEVAOUT- TEVAIN:0;T[PEVA]-TEVAIN:0;Table[EVA CON]:0;Manual:1;	set_ruleEEV(3) возврат TEVAOUT- TEVAIN:0;T[PEVA]- TEVAOUT:0;Table[EV A CON]:0;Manual:1;	+
get_remarkEEV	нет	Получить дополнительную информацию по выбранному алгоритму определения перегрева ЭРВ. Строка 80+1	get_remarkEEV возврат Перегрев не вычисляется. Ручной режим. ЭРВ открывается на заданное число шагов.	+
get_tableEEV	параметр	Получить таблицу ЭРВ. Параметр задает тип таблицы 1. HEAT - отопление 2. COOL — охлаждение Возвращается 25 чисел, разделенных ;		
set_tableEEV				
<b>Инвертор</b>				
get_infoFC	нет	Информацию об инверторе		
get_FC	нет	Получение <b>целевой</b> частоты (герцы) Строка 3+1. ЦЕЛОЕ число	get_FC возврат 78	+
get_dacFC	нет	Получение текущего значения ЦАП Строка 4+1	get_dacFC возврат 1024	+

set_FC	Новое значение	<u>Установка целевой</u> частоты (герцы) Строка 3+1. Возвращает новое значение частоты. ЦЕЛОЕ число	get_FC(45) возврат 45	+
get_noteFC	нет	Получение описания частотного преобразователя. Строка 80+1	get_noteFC возврат Частотный преобразователь	+
get_nameFC	нет	Получение имени частотного преобразователя. Строка 10+1. запрос сделан для единообразия.	get_nameFC возврат FC	+
get_paramFC	Имя параметра	Получить параметр FC. Разрешены следующие имена: 1. ON_OFF Флаг включения выключения 2. MIN_FC Только чтение минимальная частота работы 3. MAX_FC Только чтение максимальная частота работы 4. START_MX — стартовая частота 5. MAX_POWER Только чтение максимальная мощность 6. STATE Только чтение Состояние ПЧ 7. FC <b>Текущая</b> частота ПЧ 8. POWER Текущая мощность ПЧ 9. AUTO_FC Флаг автоматической подбора частоты 10. ANALOG Флаг аналогового управления 11. LEVEL0 Уровень частоты 0 в отсчетах ЦАП 12. LEVEL100 Уровень частоты 100% в отсчетах ЦАП 13. LEVELOFF Уровень частоты в % при отключается инвертор 14. STOP_FC флаг глобальная ошибка инвертора - работа инвертора запрещена блокировку можно сбросить установив в 0 15. ERROR_FC Код ошибки	get_paramFC(LEVEL0) возврат 46	+
set_paramFC	Имя параметра = значение	<u>Установить</u> параметр ЭРВ. Разрешены следующие имена: 1. ON_OFF Флаг включения выключения 2. MIN_FC Только чтение минимальная частота работы 3. MAX_FC Только чтение максимальная частота работы 4. START_MX — стартовая частота 5. MAX_POWER Только чтение максимальная мощность 6. STATE Только чтение Состояние ПЧ 7. FC Текущая частота ПЧ 8. POWER Текущая мощность ПЧ 9. AUTO_FC Флаг автоматической подбора частоты 10. ANALOG Флаг аналогового управления 11. LEVEL0 Уровень частоты 0 в отсчетах ЦАП 12. LEVEL100 Уровень частоты 100% в	set_paramEEV(LEVEL100=2048) возврат 2048	+

		отсчетах ЦАП 13. LEVELOFF Уровень частоты в % при отключается инвертор 14. STOP_FC флаг глобальная ошибка инвертора - работа инвертора запрещена блокировку можно сбросить установив в 0 15. ERROR_FC Код ошибки		
get_pinFC	нет	Получение номера пина куда прицеплен FC. Возвращает строку из 3+1 символов (первая позиция буква A	get_pinFC возврат A23	+
get_presentFC	нет	Наличие FC в конфигурации. Присутствует - «1», отсутствует «0». Строка два байта (1+1).	get_presentFC возврат 1	+
reset_MX2	нет	Сброс инвертора по модбасу	reset_MX2 возврат Инвертор сброшен . . .	+
reset_errorMX2	нет	Сброс ошибок в инверторе	reset_errorMX2 возврат Ошибки инвертора сброшены . . .	+
<b>Общие запросы по ТН</b>				
get_version	нет	Получение значения текущей версии прошивки. Строка 6+1	get_version возврат 0.12a	+
get_config	нет	Получение имени текущей конфигурации. Строка 10+1	get_config возврат pav2000	+
get_configNote	нет	Получение описания текущей конфигурации. Строка 80+1	get_configNote возврат Старт стоп Компрессор скрол EVI с шаговым ЭРВ и РТО	+
get_datetime	параметр	Получение значения параметра времени и даты. Разрешены следующие имена 1. TIME — текущее время 12:45 без секунд 2. DATE — текущая дата типа 12/04/2016 3. NTP — адрес NTP сервера строка до 60 4. UPDATE — время синхронизации с NTP сервером. Разрешены следующие значения 4.1 0 — никогда 4.2 1 — раз в сутки 5. UPDATE_I2C — обновление внутренних от i2c часов (актуально если для внутренних часов используется внутренняя RC цепочка) 5.1 0 — никогда 5.2 1 — раз в час	get_datetime(TIME) возврат 12:34	+
set_datetime	параметр	<u>Устанавливает</u> значения параметра времени и даты. Разрешены следующие имена 1. TIME — текущее время 12:45 без секунд 2. DATE — текущая дата типа 12/04/2016 3. NTP — адрес NTP сервера строка до 60 символов. 4. UPDATE — время синхронизации с NTP	set_datetime(TIME=12:34) возврат 12:34	+

		сервером. Разрешены следующие значениям 4.1 0 — никогда 4.2 1 — раз в сутки 5. UPDATE_I2C — обновление внутренних от i2c часов (актуально если для внутренних часов используется внутренняя RC цепочка) 5.1 0 — никогда 5.2 1 — раз в час		
update_NTP	нет	Обновить время по NTP возвращает или «Update Ok» если удачно или «Update error»	update_NTP возврат Update Ok	+
get_startDT	нет	Получить дату и время последней перезагрузки контроллера	get_startDT возврат 14:23 04/07/2016	+
get_resetCause	нет	Вывести причину последней перезагрузки контроллера	get_resetCause возврат Watchdog	+
get_Network	параметр	Получить сетевой параметр. Определены следующие параметры: 0 — ADDRESS — адрес контроллера 1 — DNS — ДНС сервер 2 — GATEWAY — шлюз 3 - SUBNET — маска подсети 4 — DHCP — использование DHCP 5 — MAC — мас адрес 6. - RES_SOCKET – время сброса зависших сокетов. Разрешены следующие значениям 6.1. none - никогда 6.2. 30 sec – 30 секунд 6.3. 300 sec – 300 секунд Выдает список список вида “none:0;30 sec:1;300 sec:0” 7. - RES_W5200 – время сброса чипа w5200. Разрешены следующие значениям 7.1. none - никогда 7.2. 6 hour – 6 часов 7.3. 24 hour – 24 часа Выдает список список вида “none:0;6 hour:1;24 hour:0” 8. PASSWORD — использование паролей для входа 9. PASSUSER — пароль пользователя (до 10 символов) 10. PASSADMIN — пароль администратора (до 10 символов) 11. SIZE_PACKET — длина пакета w5200 max 2048 байт 12. INIT_W5200 — проверка раз в минуту и при необходимости инициализация чипа w5200. Поддержание работы SPI 13. PORT — адрес порта веб сервера/ 14. NO_ACK — флаг ожидания ответа ack	get_network(ADDRESS) возврат 192.168.1.2	+

		<p>15. DELAY_ACK — задержка на посылку следующего пакета (если включен NO_ACK)</p> <p>16. PING_ADR - адрес для пинга</p> <p>17. PING_TIME - время пинга в секундах, 0 секунд — пинг не нужен, отдает список вида "never:1;1 min:0;5 min:0;20 min:0;60 min:0;"</p> <p>Выходные значения строки адрес IP вида «255.345.456.1» мас вида «34:ad:ff:34:67:1f» использование DHCP «0» или «1»</p>		
set_Network	параметр=значение	<p>Установить сетевой параметр Определены следующие параметры:</p> <p>0 — ADDRESS — адрес контроллера</p> <p>1 — DNS — ДНС сервер</p> <p>2 — GATEWAY — шлюз</p> <p>3 - SUBNET — маска подсети</p> <p>4 — DHCP — использование DHCP</p> <p>5 — MAC — мас адрес</p> <p>6. - RES_SOCKET – время сброса зависших сокетов. Разрешены следующие значениям (это числа от 0 до 3)</p> <p>6.1. 0 - никогда</p> <p>6.2. 1 – 30 секунд</p> <p>6.3. 2 – 300 секунд</p> <p>Возвращает список список вида “none:0;30 sec:1;300 sec:0”</p> <p>7. - RES_W5200 – время сброса чипа w5200. Разрешены следующие значениям (это числа от 0 до 3)</p> <p>7.1. 0 - никогда</p> <p>7.2. 1 – 6 часов</p> <p>7.3. 2 – 24 часа</p> <p>Возвращает список список вида “none:0;6 hour:1;24 hour:0”</p> <p>8. PASSWORD — использование паролей для входа</p> <p>9. PASSUSER — пароль пользователя (до 10 символов)</p> <p>10. PASSADMIN — пароль администратора (до 10 символов)</p> <p>11. SIZE_PACKET — длина пакета w5200 max 2048</p> <p>12. INIT_W5200 — проверка раз в минуту и при необходимости инициализация чипа w5200. Поддержание работы SPI</p> <p>13. PORT — адрес порта веб сервера</p> <p>14. NO_ACK — флаг ожидания ответа ack</p> <p>15. DELAY_ACK — задержка на посылку следующего пакета (если включен NO_ACK)</p> <p>16. PING_ADR - адрес для пинга</p> <p>17. PING_TIME - время пинга в секундах, 0 секунд — пинг не нужен, отдает список вида "never:1;1 min:0;5 min:0;20 min:0;60 min:0;"</p> <p>Выходные значения строки адрес IP вида «255.345.456.1»</p>	<p>get_network(ADDRESS=192.168.1.2) возврат 192.168.1.2</p>	+



		мас вида «34:ad:ff:34:67:1f» использование DHCP «0» или «1» при ошибке преобразования строки или при выходе за диапазон возвращает ошибку «E15»		
set_updateNet	нет	Применить сетевые настройки, перезагрузить сетевую карту	set_updateNet возврат Сброс контроллера, подождите 10 секунд . . .	
get_socketInfo	нет	Получить информацию о сокетах Длинная строка, содержащая 8 подстрок, разделитель «;». Формат подстроки Номер Статус Mac IP Port;	get_socketInfo возврат длинная строка 8 раз Номер Статус Mac IP Port;	+
get_socketRes	нет	Получить число сброшенных сокетов	get_socketRes возврат 34	+
TEST	нет	Заглушка для тестирования веб морды. Возвращает при каждом обращении случайное целое число от -50 до +50 в виде строки 3+1 байт. В окончательной версии веб морды быть не должно.	TEST Возврат 35	+
TASK_LIST	нет	Получить список задач FreeRTOS. Работает если включена опция в либе FreeRTOS (файл FreeRTOSConfig.h) STAT_FREE_RTOS. Запрос http://192.168.0.177/&TASK_LIST&& Использовать только для отладки, сильно ест ресурсы.	TASK_LIST возврат &TASK_LIST=6:Web0:Ready:1:44:1731:<1% 8:Web2:Ready:1:54:2481:<1% 7:Web1:Ready:1:54:4009:<1% 13:IDLE:Ready:0:126:1220184:96% 9:Nextion:Blocked:1:123:35163:2% 4:updateHP:Blocked:2:150:0:<1% 5:updateEEV:Blocked:2:38:0:<1% 10:upStat:Blocked:0:62:0:<1% 1:rSensor:Blocked:4:210:2:<1% 14:Tmr_Svc:Blocked:2:68:0:<1% 3:Command:Suspended:3:92:39:<1% 2:upStepper:Suspended:4:70:0:<1% 11:upPump:Suspended:0:190:0:<1% 12:delayStart:Suspended:3:190:0:<1%&&	+
RESET_JOURNAL	нет	Сбросить системный журнал	RESET_JOURNAL возврат Reset journal . . .	+
RESET	нет	Сброс контроллера Ответ: Сброс контроллера, подождите 10 секунд . . .	RESET возврат Сброс контроллера, подождите 10 секунд . . .	+
RESET_NET	нет	Сброс сетевого чипа УДАЛЕНО Используйте запрос set_updateNet	RESET_NET ответа нет	-
RESET_COUNT	нет	Сброс счетчиков мото часов за СЕЗОН! Есть кнопка на веб морде (страница Система)	RESET_COUNT возврат Сброс BCEX счетчика моточасов	+
RESET_ALL_COUNT	нет	Сброс BCEX счетчиков мото часов. Кнопки нет. Надо в ручную вводить в строке браузера IP/&RESET_ALL_COUNT&&	RESET_ALL_COUNT возврат Сброс BCEX счетчика моточасов	+

RESET_SETTINGS	нет	Сброс настроек ТН	OK	+
CONST	нет	Первая часть (см CONST1) Вывод в длинную строку основных констант. Длина до 6000+1. Формат типа name1 desc1 value1;name2 desc2 value2	CONST Возврат Длинная строка Ла-ла-ла	+
CONST1	нет	Вторая часть констант (см CONST)	CONST+ Возврат Длинная строка Ла-ла-ла	+
get_status	нет	Вывод в длинную строку состояния контроллера. Длина 2000+1. Формат типа ВРЕМЯ;ДАТА;ВЕРСИЯ;ПАМЯТЬ;ЗАГРУЗКА;АПТАЙМ;ПЕРЕГРЕВ;ОБОРОТЫ;СОСТОЯНИЕ ;	get_status возврат 12:10;12/12/17;0.78 beta;4567;12%;1d 34m;2345;Нагрев	+
get_sysInfo	нет	Вывод в длинную строку состояния контроллера. Служебные переменные, перемычки	get_sysInfo возврат Перемычка PIN_WIRE_NET 'Установка сетевых настроек по умолчанию' [активно 0] 1;Перемычка PIN_WIRE_PW 'B	+
get_freeRam	нет	Свободная память контроллера в байтах. Строка 5+1 байт. Free RTOS для DUE использует heap_3 по этому трудно определить размер свободной памяти. <b>Функция может врать!</b> Надо проверить, не понятен размер стека.	get_freeRam возврат 84356	+
get_loadingCPU	нет	Возвращает загрузку CPU в %. Длина 3+1 Величина близкая к 100% показывает полную загрузку контроллера. Интервал усреднения 5 секунд	get_loadingCPU возврат 17	+
set_SAVE	нет	Записать конфигурацию в eeprom i2c. Возвращает число записанных байт. Или код ошибки (если число меньше 0)	set_SAVE возврат 18	+
set_ON	нет	Включить тепловой насос. Сбрасывается последняя ошибка.	set_ON возврат 1	+
set_OFF	нет	Выключить тепловой насос	set_OFF возврат 0	+
get_MODE	нет	Выдает текущий режим работы. Определены следующие режимы (пока!) ТН выключен, Отопление, ГВС	get_MODE возврат ГВС	+
get_WORK	нет	Получить состояние теплового насоса (включен/выключен) Включен — ON выключен - OFF	get_WORK возврат OFF	+
get_errcode	нет	Возвращает код последней ошибки (которая привела к остановке ТН). Строка до 3+1 символов. 0 — окей все ошибки значения	get_errcode возврат -6	+

		меньше 0		
get_error	нет	Возвращает описание последней ошибки (см get_errcode). Формат строки «время имя_источника : описание_ошибки». Строка до 100+1 байт	get_error возврат 12:12:12 TIN:Адрес датчика температуры не установлен	+
get_tempSAM3x	нет	Возвращает температуру чипа SAM3x в градусах Цельсия. Длина 5+1.	get_tempSAM3x возврат 60.12	+
get_uptime	нет	Получить время с последней перезагрузки контроллера	get_uptime возврат 1d 14h 1 m	+
get_VCC	нет	Получить напряжение питания контроллера. Если эта функция поддерживается. Длина 5+1.	get_VCC возврат 12.06	+
get_tempDS3231	нет	Возвращает температуру чипа DS3231 (часы реального времени) в градусах Цельсия. Длина 5+1.	get_tempDS3231 возврат 21.45	+
get_testMode	нет	Возвращает режим работы ТН, с точки зрения тестирования. Определены следующие режимы 0 NORMAL — нормальный режим, работа 1 SAFE_TEST — значения датчиков берутся из полей тест, работа исполнительных устройств имитируется. 2 TEST значения датчиков берутся из полей тест, исполнительные устройства работают кроме компрессора 3 HARD_TEST - значения датчиков берутся из полей тест, исполнительные устройства работают. На выходе строка вида NORMAL:0;SAFE_TEST:1; TEST:0; HARD_TEST:0;	get_testMode возврат NORMAL:0;SAFE_TE ST:1; TEST:0; HARD_TEST:0;	+
set_testMode	нет	<u>Устанавливает</u> режим работы ТН, с точки зрения тестирования. На входе число (индекс в списке). Определены следующие режимы 0 NORMAL — нормальный режим, работа 1 SAFE_TEST — значения датчиков берутся из полей тест, работа исполнительных устройств имитируется. 2 TEST значения датчиков берутся из полей тест, исполнительные устройства работают кроме компрессора 3 HARD_TEST - значения датчиков берутся из полей тест, исполнительные устройства работают (все). На выходе строка вида NORMAL:0;SAFE_TEST:1; TEST:0; HARD_TEST:0;	set_testMode(0) возврат NORMAL:1;SAFE_TE ST:0; TEST:0; HARD_TEST:0;	+
get_remarkTest	нет	Получить дополнительную информацию по выбранному режиму тестирования (testMode). Строка 80+1	get_remarkTest возврат Тестирование отключено. Основной режим работы	+

get_Message	параметр	<p>Получить параметр настройки уведомлений.</p> <p>Разрешены следующие имена параметров:</p> <ol style="list-style-type: none"> <li>1. MAIL - флаг уведомления скидывать на почту</li> <li>2. MAIL_AUTH флаг необходимости авторизации на почтовом сервере</li> <li>3. MAIL_INFO - флаг необходимости добавления в письмо информации о состоянии ТН</li> <li>4. SMS - флаг уведомления скидывать на СМС (пока не реализовано)</li> <li>5. MESS_RESET - флаг уведомления Сброс</li> <li>6. MESS_ERROR - флаг уведомления Ошибка</li> <li>7. MESS_LIFE - флаг уведомления Сигнал жизни</li> <li>8. MESS_TEMP - флаг уведомления Достижение граничной температуры</li> <li>9. MESS_SD - флаг уведомления "Проблемы с sd картой"</li> <li>10. MESS_WARNING - флаг уведомления "Прочие уведомления"</li> <li>11. SMTP_SERVER - Адрес сервера</li> <li>12. SMTP_IP - IP Адрес сервера</li> <li>13. SMTP_PORT - Адрес порта сервера</li> <li>14. SMTP_LOGIN - логин сервера если включена авторизация</li> <li>15. SMTP_PASS - пароль сервера если включена авторизация</li> <li>16. SMTP_MAILTO - адрес отправителя</li> <li>17. SMTP_RCPTTO - адрес получателя</li> <li>18. SMS_SERVICE - сервис отправки смс</li> <li>19. SMS_PHONE - телефон куда отправляется смс</li> <li>20. SMS_P1 -первый параметр для отправки смс</li> <li>21. SMS_P2 - второй параметр для отправки смс</li> <li>22. SMS_NAMEP1 - описание первого параметра для отправки смс</li> <li>23. SMS_NAMEP2 - описание второго параметра для отправки смс</li> <li>24. SMS_IP - IP Адрес сервера для отправки смс</li> <li>25. MESS_TIN - Критическая температура в доме (если меньше то генерится уведомление)</li> <li>26 MESS_TBOILER - Критическая температура бойлера (если меньше то генерится уведомление)</li> <li>27. MESS_TCOMP - Критическая температура компрессора (если больше то генериться уведомление)</li> <li>28. MAIL_RET, // Ответ на тестовую почту</li> <li>29. SMS_RET, // Ответ на тестовую sms</li> </ol>	+
set_Message	параметр	<p><u>Установить</u> параметр настройки уведомлений.</p> <p>Разрешены следующие имена параметров:</p> <ol style="list-style-type: none"> <li>1. MAIL - флаг уведомления скидывать на почту</li> <li>2. MAIL_AUTH флаг необходимости</li> </ol>	+

		<p>авторизации на почтовом сервере</p> <p>3. MAIL_INFO - флаг необходимости добавления в письмо информации о состоянии ТН</p> <p>4. SMS - флаг уведомления скидывать на СМС (пока не реализовано)</p> <p>5. MESS_RESET - флаг уведомления Сброс</p> <p>6. MESS_ERROR - флаг уведомления Ошибка</p> <p>7. MESS_LIFE - флаг уведомления Сигнал жизни</p> <p>8. MESS_TEMP - флаг уведомления Достижение граничной температуры</p> <p>9. MESS_SD - флаг уведомления "Проблемы с sd картой"</p> <p>10. MESS_WARNING - флаг уведомления "Прочие уведомления"</p> <p>11. SMTP_SERVER - Адрес сервера</p> <p>12. SMTP_IP - IP Адрес сервера</p> <p>13. SMTP_PORT - Адрес порта сервера</p> <p>14. SMTP_LOGIN - логин сервера если включена авторизация</p> <p>15. SMTP_PASS - пароль сервера если включена авторизация</p> <p>16. SMTP_MAILTO - адрес отправителя</p> <p>17. SMTP_RCPTTO - адрес получателя</p> <p>18. SMS_SERVICE - сервис отправки смс</p> <p>19. SMS_IP - IP Адрес сервера для отправки смс</p> <p>20. SMS_PHONE - телефон куда отправляется смс</p> <p>21. SMS_P1 -первый параметр для отправки смс</p> <p>22. SMS_P2 - второй параметр для отправки смс</p> <p>23. SMS_NAMEP1 - описание первого параметра для отправки смс</p> <p>24. SMS_NAMEP2 - описание второго параметра для отправки смс</p> <p>25. MESS_TIN - Критическая температура в доме (если меньше то генерится уведомление)</p> <p>26 MESS_TBOILER - Критическая температура бойлера (если меньше то генерится уведомление)</p> <p>27. MESS_TCOMP - Критическая температура компрессора (если больше то генериться уведомление)</p> <p><i>Внимание – некоторые параметры имеют статус только чтение (они зачеркнуты). При попытке их установить, они не меняются и возвращается значение «read only»</i></p>		
test_Mail	нет	Отправить тестовое уведомление по почте, для проверки сервиса.	testMail возврат Send Mail	+
test_SMS	нет	Отправить тестовое уведомление по SMS, для проверки сервиса.	testSMS возврат Send Mail	+
<b>Параметры настройки бойлера и отопления/охлаждения</b>				

get_Boiler	параметр	<p>Получить режим работы бойлера, определены следующие параметры</p> <ol style="list-style-type: none"> <li>1. BOILER_ON - флаг Включения бойлера</li> <li>2. SCHEDULER_ON - флаг Использование расписания</li> <li>3. TURBO_BOILER - флаг использование ТЭН для нагрева ПАРАЛЛЕЛЬНО с ТН (ускоренный нагрев, турбо режим), Режим догрева блокируется, турбо имеет приоритет перед догревом.</li> <li>4. SALLMONELA - флаг Сальмонела раз в неделю греть бойлер до 70 градусов</li> <li>5. CIRCULATION - флаг Управления циркуляционным насосом ГВС</li> <li>6. TEMP_TARGET - Целевая температура бойлера</li> <li>7. DTARGET - гистерезис целевой температуры</li> <li>8. TEMP_MAX - Температура подачи максимальная</li> <li>9. PAUSE1 - Минимальное время простоя компрессора в минутах</li> <li>10. SCHEDULER — Расписание. 7 строк (по дням недели), разделитель между днями «/», далее по часам (0 или 1) разделителя НЕТ (между часами) всего 24*7+7=175 байт. Длина контролируется. Один день — 101000000000000000000000/ это повторяется 7 раз первый день — понедельник, первый час 0</li> <li>11. CIRCUL_WORK — время работы насоса ГВС минуты (работает с установленным флагом CIRCULATION).</li> <li>12. CIRCUL_PAUSE — пауза в работе насоса ГВС минуты (работает с установленным флагом CIRCULATION).</li> <li>13. RESET_HEAT – Флаг сброса лишнего тепла при нагреве бойлера в СО.</li> <li>14. RESET_TIME – время сброса тепла в минутах.</li> <li>15. BOIL_TIME —Постоянная интегрирования времени в секундах ПИД ТН</li> <li>16. BOIL_PRO - Пропорциональная составляющая ПИД ТН</li> <li>17. BOIL_IN — Интегральная составляющая ПИД ТН</li> <li>18. BOIL_DIF - Дифференциальная составляющая ПИД ТН</li> <li>19. BOIL_TEMP — целевая температура ПИД</li> <li>20 ADD_HEATING - флаг использования тена для догрева ГВС. Режим турбо должен быть выключен</li> <li>21 TEMP_RBOILER - температура включения догрева бойлера</li> </ol>	<p>get_paramBoiler(BOILER_O)</p> <p>возврат 1</p>
set_Boiler		<p><u>Установить</u> режим работы бойлера, определены следующие параметры</p>	<p>set_Boiler(BOILER_O=0)</p>

		<p>1. BOILER_ON - флаг Включения бойлера</p> <p>2. SCHEDULER_ON - флаг Использование расписания</p> <p>3. TURBO_BOILER - флаг использование ТЭН для нагрева ПАРАЛЛЕЛЬНО с ТН (ускоренный нагрев, турбо режим), блокировка режима догрева, турбо имеет приоритет перед догревом.</p> <p>4. SALLMONELA - флаг Сальмонела раз в неделю греть бойлер до 70 градусов</p> <p>5. CIRCULATION - флаг Управления циркуляционным насосом ГВС</p> <p>6. TEMP_TARGET - Целевая температура бойлера</p> <p>7. DTARGET - гистерезис целевой температуры</p> <p>8. TEMP_MAX - Температура подачи максимальная</p> <p>9. PAUSE1 - Минимальное время простоя компрессора в минутах</p> <p>10. SCHEDULER — Расписание. 7 строк (по дням недели), разделитель между днями «/», далее по часам (0 или 1) разделителя НЕТ (между часами) всего 24*7+7=175 байт. Длина контролируется. Один день — 101000000000000000000000/ это повторяется 7 раз первый день — понедельник, первый час 0</p> <p>11. CIRCUL_WORK — работа насоса ГВС минуты (работает с установленным флагом CIRCULATION).</p> <p>12. CIRCUL_PAUSE — пауза в работе насоса ГВС минуты (работает с установленным флагом CIRCULATION).</p> <p>13. RESET_HEAT – Флаг сброса лишнего тепла при нагреве бойлера в СО.</p> <p>14. RESET_TIME – время сброса тепла в минутах.</p> <p>15. BOIL_TIME —Постоянная интегрирования времени в секундах ПИД ТН</p> <p>16. BOIL_PRO - Пропорциональная составляющая ПИД ТН</p> <p>17. BOIL_IN — Интегральная составляющая ПИД ТН</p> <p>18. BOIL_DIF - Дифференциальная составляющая ПИД ТН</p> <p>19. BOIL_TEMP — целевая температура ПИД</p> <p>20 ADD_HEATING - флаг использования тена для догрева ГВС. Режим турбо должен быть выключен</p> <p>21 TEMP_RBOILER - температура включения догрева бойлера</p>	<p>возврат 0</p>	
get_modeHP	нет	<p>Получить режим работы ТН возвращает строку вида «Выключено:0;Отопление:1;Охлаждение:0;»</p>	<p>get_modeHP возврат «Выключено:0; Отопление:1;Охлаждение:0;»</p>	+



set_modeHP	параметр	Установить режим работы ТН 0 — выключено 1- отопление 2 - охлаждение	set_modeHP(0) возврат «Выключено:0;Отопление:1;Охлаждение:0;»	+
get_paramCoolHP	параметр	Получить значение параметра охлаждения. Разрешены следующие имена параметров: 1. RULE — алгоритм работы отопления. Список. Разрешены следующие значения: <i>1.1 Hysteresis – гистерезис, интервальный режим</i> <i>1.2 PID – использование ПИД регулятора</i> <i>1.3 Hybrid – смешанный алгоритм, предложил Ljutik</i> Возвращает список !!! 2. TEMP1 — целевая температура в доме 3. TEMP2 — целевая температура обратки 4. TARGET что является целью ПИД - значения 0 (температура в доме), 1 (температура обратки). 5. DTEMP — гистерезис целевой температуры 6. HP_TIME —Постоянная интегрирования времени в секундах ПИД ТН 7. HP_PRO - Пропорциональная составляющая ПИД ТН 8. HP_IN — Интегральная составляющая ПИД ТН 9. HP_DIF - Дифференциальная составляющая ПИД ТН 10. TEMP_IN — температура подачи (минимальная) 11. TEMP_OUT — температура обратки (максимальная) 12. PAUSE — минимальное время простоя компрессора 13. D_TEMP — максимальная разность температур конденсатора. 14 TEMP_PID — целевая температура ПИД 15 WEATHER — использование погодозависимости 0 или 1 16 K_WEATHER — коэффициент погодозависимости 0.0-1.0 (точность до сотой)	get_paramCoolHP(TEMP1 ) возврат 22.03 <u>внимание</u> для параметра TARGET идет возврат списка get_paramCoolHP( TARGET ) возврат Дом:1;Обратка:0; <u>внимание</u> для параметра RULE идет возврат списка get_paramCoolHP( RULE) возврат HYSTERESIS:1;PID:0; HYBRID:0:	+
set_paramCoolHP	параметр р=значение	Установить значение параметра охлаждения. Разрешены следующие имена параметров 1. RULE — алгоритм работы отопления. Список. Разрешены следующие значения: <i>1.1 Hysteresis – гистерезис, интервальный режим</i> <i>1.2 PID – использование ПИД регулятора</i> <i>1.3 Hybrid – смешанный алгоритм, предложил Ljutik</i> Возвращает список !!! 2. TEMP1 — целевая температура в доме 3. TEMP2 — целевая температура обратки 4. TARGET что является целью ПИД - значения	set_paramCoolHP(TEMP1=2 4.11) возврат 24.11 <u>внимание</u> для параметра TARGET идет возврат списка (dfhbfyns 0 и 1) get_paramCoolHP( TARGET =1) возврат Дом:0;Обратка:1; <u>внимание</u> для параметра RULE идет возврат списка get_paramCoolHP( RULE=2 ) возврат	+

		<p>0 (температура в доме), 1 (температура обратки).</p> <p>5. DTEMP — гистерезис целевой температуры</p> <p>6. HP_TIME — Постоянная интегрирования времени в секундах ПИД ТН</p> <p>7. HP_PRO - Пропорциональная составляющая ПИД ТН</p> <p>8. HP_IN — Интегральная составляющая ПИД ТН</p> <p>9. HP_DIF - Дифференциальная составляющая ПИД ТН</p> <p>10. TEMP_IN — температура подачи (минимальная)</p> <p>11. TEMP_OUT — температура обратки (максимальная)</p> <p>12. PAUSE — минимальное время простоя компрессора</p> <p>13. D_TEMP — максимальная разность температур конденсатора.</p> <p>14 TEMP_PID — целевая температура ПИД</p> <p>15 WEATHER — использование погодозависимости 0 или 1</p> <p>16 K_WEATHER — коэффициент погодозависимости 0.0-1.0 (точность до сотой)</p>	<p>HYSTERESIS:0;PID:0; HYBRID:1:</p>	
get_paramHeatHP	параметр	<p>Получить значение параметра Отопления.</p> <p>Разрешены следующие имена параметров:</p> <p>1. RULE — алгоритм работы отопления. Список.</p> <p>Разрешены следующие значения:</p> <p>1.1 <i>HYSTERESIS – гистерезис, интервальный режим</i></p> <p>1.2 <i>PID – использование ПИД регулятора</i></p> <p>1.3 <i>HYBRID – смешанный алгоритм, предложил <a href="#">Ljutik</a></i></p> <p>Возвращает список !!!</p> <p>2. TEMP1 — целевая температура в доме</p> <p>3. TEMP2 — целевая температура обратки</p> <p>4. TARGET что является целью ПИД - значения 0 (температура в доме), 1 (температура обратки).</p> <p>5. DTEMP — гистерезис целевой температуры</p> <p>6. HP_TIME — Постоянная интегрирования времени в секундах ПИД ТН</p> <p>7. HP_PRO - Пропорциональная составляющая ПИД ТН</p> <p>8. HP_IN — Интегральная составляющая ПИД ТН</p> <p>9. HP_DIF - Дифференциальная составляющая ПИД ТН</p> <p>10. TEMP_IN — температура подачи (минимальная)</p> <p>11. TEMP_OUT — температура обратки (максимальная)</p> <p>12. PAUSE — минимальное время простоя компрессора</p> <p>13. D_TEMP — максимальная разность температур конденсатора.</p>	<p>get_paramCoolHP(TEMP1 ) возврат 22.03 <u>внимание</u> для параметра TARGET идет возврат списка get_paramCoolHP( TARGET ) возврат Дом:1;Обратка:0; <u>внимание</u> для параметра RULE идет возврат списка get_paramCoolHP( RULE) возврат HYSTERESIS:1;PID:0; HYBRID:0:</p>	+

		14 TEMP_PID — целевая температура ПИД 15 WEATHER — использование погодозависимости 0 или 1 16 K_WEATHER — коэффициент погодозависимости 0.0-1.0 (точность до сотой)		
set_paramHeatHP	параметр=значение	<p>Установить значение параметра нагрева.</p> <p>Разрешены следующие имена параметров</p> <p>1. RULE — алгоритм работы отопления. Список.</p> <p>Разрешены следующие значения:</p> <p>1.1 <i>HYSTERESIS</i> – гистерезис, интервальный режим</p> <p>1.2 <i>PID</i> – использование ПИД регулятора</p> <p>1.3 <i>HYBRID</i> – смешанный алгоритм, предложил <a href="#">Ljutik</a></p> <p>Возвращает список !!!</p> <p>2. TEMP1 — целевая температура в доме</p> <p>3. TEMP2 — целевая температура обратки</p> <p>4. TARGET что является целью ПИД - значения 0 (температура в доме), 1 (температура обратки).</p> <p>5. DTEMP — гистерезис целевой температуры</p> <p>6. HP_TIME —Постоянная интегрирования времени в секундах ПИД ТН</p> <p>7. HP_PRO - Пропорциональная составляющая ПИД ТН</p> <p>8. HP_IN — Интегральная составляющая ПИД ТН</p> <p>9. HP_DIF - Дифференциальная составляющая ПИД ТН</p> <p>10. TEMP_IN — температура подачи (максимальная)</p> <p>11. TEMP_OUT — температура обратки (минимальная)</p> <p>12. PAUSE — минимальное время простоя компрессора</p> <p>13. D_TEMP — максимальная разность температур конденсатора.</p> <p>14 TEMP_PID — целевая температура ПИД</p> <p>15 WEATHER — использование погодозависимости 0 или 1</p> <p>16 K_WEATHER — коэффициент погодозависимости 0.0-1.0 (точность до сотой)</p>	set_paramCoolHP(TEMP1=24.11) возврат 24.11 <b>внимание</b> для параметра TARGET идет возврат списка (dfhbfyns 0 и 1) get_paramCoolHP( TARGET =1) возврат Дом:0;Обратка:1; <b>внимание</b> для параметра RULE идет возврат списка get_paramCoolHP( RULE=1 ) возврат <i>HYSTERESIS:0;PID:1;HYBRID:0;</i>	+
get_optionHP	параметр	<p>Получить значение отдельной опции ТН.</p> <p>Разрешены следующие параметры</p> <p>1. ADD_HEAT — использование дополнительного нагревателя. Разрешены следующие переменные</p> <p>0 использование ТЭН запрещено</p> <p>1 Резерв, включение ТЭНа при температуре в ДОМЕ меньше чем TEMP_RHEAT</p> <p>2 Бивалент включение ТЭНа при температуре на УЛИЦЕ меньше чем TEMP_RHEAT (актуально для воздушников)</p> <p>На выходе строка вида</p>	get_optionHP(ADD_HEAT) возврат none:0;reserve:1;bivalent:0;	

		<p>«none:0;reserve:1;bivalent:0;»</p> <p>2. TEMP_RHEAT — температура для управления дополнительным теном при нагреве СО (градусы)</p> <p>3. PUMP_WORK — работа насоса конденсатора при выключенном компрессоре секунды</p> <p>4. PUMP_PAUSE — пауза между работой насоса конденсатора при выключенном компрессоре (секунды)</p> <p>5. ATTEMPT — число попыток пуска ТН</p> <p>6 TIME_CHART – период сбора статистики</p> <p>Возвращает строку «10 sec:0;1 min:1;5 min:0;10 min:0;30 min:0;60 min:0»</p> <p>Разрешены следующие переменные</p> <p>0 10 sec</p> <p>1 1 min</p> <p>2 3 min</p> <p>3 10 min</p> <p>4 30 min</p> <p>5 60 min</p> <p>7. BEEP - установить возможность подачи звуковых сигналов</p> <p>8 NEXTION - подключение дисплея Nextion</p> <p>9 EEV_STOP – закрытие ЭРВ при выключении компрессора</p> <p>10 SD_CARD - запись статистики в файл stat.csv</p> <p>11 SAVE_ON – сохранение состояния ТН в ЕЕПРОМ, для восстановления его после сброса.</p> <p>12 EEV_START -Всегда начинать работу ЭРВ со стартовой позиции, если флаг не установлен работа начинается с позиции которая была при выключении</p> <p>13. NEXT_SLEEP – время (минуты) через которое дисплей засыпает если не было прикосновений. Разрешенные значения 0... 30, если введен 0 то дисплее вообще не засыпает.</p> <p>14 NEXT_DIM — яркость дисплея % разрешенные значения 10-100</p> <p>15 EEV_LIGHT_START – приоткрытие ЭРВ при старте компрессора, облегчение пуска компрессора (если больше стартовой) или для уменьшения выхода на режим (если меньше стартовой)</p>	
set_optionHP	параметр	<p><u>Установить</u> значение отдельной опции ТН.</p> <p>Разрешены следующие параметры</p> <p>1. ADD_HEAT — использование дополнительного нагревателя. Разрешены следующие переменные</p> <p>0 использование ТЭН запрещено</p> <p>1 Резерв, включение ТЭНа при температуре в ДОМЕ меньше чем TEMP_RHEAT</p> <p>2 Бивалент включение ТЭНа при температуре на УЛИЦЕ меньше чем TEMP_RHEAT (актуально для воздушников)</p>	<p>set_optionHP(ADD_HEAT=1)</p> <p>возврат</p> <p>none:0;reserve:1;bivalent:0;</p>

		<p>На выходе строка вида «none:0;reserve:1;bivalent:0;»</p> <p>2. TEMP_RHEAT — температура для управления дополнительным ТЭНом при нагреве СО (градусы) см ADD_HEAT</p> <p>3. PUMP_WORK — работа насоса конденсатора при выключенном компрессоре секунды</p> <p>4.PUMP_PAUSE — пауза между работой насоса конденсатора при выключенном компрессоре (секунды)</p> <p>5. ATTEMPT — число попыток пуска ТН</p> <p>6 TIME_CHART – период сбора статистики</p> <p>Разрешены следующие переменные</p> <p>0 10 sec</p> <p>1 1 min</p> <p>2 3 min</p> <p>3 10 min</p> <p>4 30 min</p> <p>5 60 min</p> <p>На выходе строка вида «10 sec:0;1 min:1;5 min:0;10 min:0;30 min:0;60 min:0»</p> <p>7. BEEP - установить возможность подачи звуковых сигналов</p> <p>8 NEXTION - подключение дисплея Nextion</p> <p>9 EEV_STOP – закрытие ЭРВ при выключении компрессора</p> <p>10 SD_CARD - запись статистики в файл stat.csv</p> <p>11 SAVE_ON – сохранение состояния ТН в ЕЕПРОМ, для восстановления его после сброса</p> <p>12 EEV_START -Всегда начинать работу ЭРВ со стартовой позиции, если флаг не установлен работа начинается с позиции которая была при выключении</p> <p>13. NEXT_SLEEP – время (минуты) через которое дисплей засыпает если не было прикосновений. Разрешенные значения 0... 30, если введен 0 то дисплее вообще не засыпает.</p> <p>14 NEXT_DIM — ярость дисплея % разрешенные значения 10-100</p> <p>15 EEV_LIGHT_START – приоткрытие ЭРВ при старте компрессора, облегчение пуска компрессора (если больше стартовой) или для уменьшения выхода на режим (если меньше стартовой)</p>		
<b>Запросы для вывода графиков</b>				
get_listChart	нет	<p>Получить список доступных графиков</p> <p>Определены следующие имена (максимальный список, если датчика нет то он пропускается в списке)</p> <p>0 _NONE – ничего не показываем</p> <p>1 _TOUT</p> <p>2 _TIN</p>	<p>get_listChart возврат _NONE;_TOUT;_TEV AIN;_TEVAOUT;_TCO NIN;_TCONOUT;_TB OILER; и т.д.</p>	+

		3 _TEVAIN 4 _TEVAOUT 5 _TCONIN 6 _TCONOUT 7 _TBOILER 8 _TACCUM 9 _TRTOOUT 10 _TCOMP 11 _TEVAING 12 _TEVAOUTG 13 _TCONING 14 _TCONOUTG 15 _PEVA 16 _PCON 17 _FLOWCON 18 _FLOWEVA 19 _FLOWPCON 20 _posEEV – позиция ЭРВ 21 _freqFC – частота инвертора гц 22 _powerFC – мощность инвертора кВт 23 _RCOMP – включение компрессора 24 _OVERHEAT - перегрев 25 _dCO – дельта CO 26 _dGEO – дельта геоконтур 27 _T[PEVA] – температура рассчитанная из давления 28 _T[PCON] – температура рассчитанная из давления 29 _PowerCO выходная мощность теплового насоса 30 _PowerGEO Мощность геоконтур 31 _COP Коэффициент преобразования холодильной машины 32. _VOLTAGE - Статистика по напряжению 33. _CURRENT - Статистика по току <del>34. _acPOWER – Статистика по активной мощности</del> <del>35. _rePOWER – Статистика по Реактивная мощность</del> 36. _fullPOWER - Статистика по Полная мощность <del>37. _kPOWER – Статистика по Коэффициент мощности</del> 38. _fullCOP - Полный COP разделитель “;”		
get_Chart	параметр	Получить данные одного графика по имени из списка полученного в get_listChart, возвращается строка, где через «;» перечислены значения. Число точек не определено! None — возвращает пустую строку	get_Chart(_TOUT) возврат 10.12;10.24;10.45;	+
<b>Удаленный датчик (wifi).</b> Номер датчика - разрешенный диапазон 1-IPNUMBER. <b>Нумерация удаленных датчиков всегда с 1!!!</b>				
set_sensorIP	строка	Это то что посылает датчик (запроса от контролла не требуются). В скобках идет	set_sensorIP(1:23.01:-85:3.23:123456)	+

		тестовая строка с параметрами датчика формата: <b>номер_датчика:температура:уровень_сигнала:напряжение_питание:счетчик_пакетов</b> номер целое число от 1 до IPNUMBER температура — float от -40 до 120 уровень - float от -100 до -1 питание — мВ целое число от 2000 до 5000 счетчик — целое число >0		
get_infoESP	-	Возвращает следующие параметры через «;» TIN, TOUT, TBOILER, ВЕРСИЯ, ПАМЯТЬ, ЗАГРУЗКА, АПТАЙМ, ПЕРЕГРЕВ, ОБОРОТЫ, СОСТОЯНИЕ.  Используется для удаленных устройств для отображения состояния ТН	get_infoESP возврат 26.2;9.9;9.9;0.771 beta;8220;10;01m ; 3.47;0.00;0ff;	
get_numberIP	-	Получить число удаленных датчиков разрешенных в системе. Если удаленные датчики не поддерживаются в системе возвращается 0	get_numberIP возврат 4	+
get_sensorParamIP	параметр	Получить отдельный параметр удаленного датчика. В качестве параметра строка«номер:параметр» определены параметры: SENSOR_TEMP -Температура SENSOR_NUMBER – Номер датчика -1 данные не валидны RSSI - Уровень сигнала VCC - Напряжение питания SENSOR_USE - Использование SENSOR_RULE -усреднение SENSOR_IP - Адрес SENSOR_COUNT – Счетчик посылок с момента включения. STIME - время с последнего считывания датчика	get_sensorParamIP(2:SENSOR_COUNT) возврат 12345	+
get_sensorIP	Номер датчик	Получить строку с параметрами удаленного датчика вход — номер датчика. Для сокращения трафика лучше использовать эту функцию.	get_sensorIP (1) возврат -1: - 100.00:none:0.0.0.0 :0.0:0.00:0:	+
get_sensorUseIP	Номер датчика	Получить использование удаленного датчика параметр номер датчика=значение 0 или 1	get_sensorUseIP(2) возврат 0	+
set_sensorUseIP	Номер датчика =параметр	Установить использование удаленного датчика параметр номер датчика=значение 0 или 1	set_sensorUseIP(2=0) возврат 0	+
get_sensorRuleIP	Номер датчика	Получить использование усреднения параметр номер датчика=значение 0 или 1	get_sensorRuleIP(2) возврат 1	+
set_sensorRuleIP	Номер датчика =параметр	Установить использование усреднения параметр номер датчика=значение 0 или 1	set_sensorRuleIP(2=1) возврат 1	+



	етр			
set_sensorListIP	Номер датчика =параметр етр	Установить привязку удаленного датчика по индексу из списка (нумерация списка с 0!!)	set_sensorListIP(2=5) ответ 5	+
get_sensorListIP	Номер датчика	Получить список привязки удаленного датчика На входе параметр — номер датчика Reset — сброс привязки привязка не установлена	get_sensorListIP(2) ответ Reset:1;TOUT:0;TIN:0;TEVAIN:0;TEVAOUT:0;TCONOUT:0;TBOILER:0;TACCUM:0;TRTOOUT:0;TCOMP:0;TEVAING:0;TEVAOUTG:0;TCONING:0;TCONOUTG:0;	+

### Работа с профилями (максимально 8 профиля)

saveProfile	Номер профиля	Записать ТЕКУЩИЙ профиль в ячейку НОМЕР ПРОФИЛЯ, возвращает число записанных байт. Внимание! Сохранение загруженного профиля как текущего не происходит. Для сохранения нужно вызвать запрос set_SAVE	saveProfile(3) ответ 234	+
loadProfile	Номер профиля	Сделать профиль (номер профиля) ТЕКУЩИМ возвращает число прочитанных байт	loadProfile(5) ответ 345	+
infoProfile	Номер профиля	Получить строку с информацией о профиле, требуется для заполнения таблицы профилей	infoProfile(5) ответ + ГВС1:пример описания профиля	+
eraseProfile	Номер профиля	Стереть информацию в профиле по его номеру. Возвращает число стертых байт.	eraseProfile(5) ответ 235	+
get_Profile	параметр	Получить отдельный параметр профиля. Разрешены следующие имена параметров: NAME_PROFILE имя профиля до 10 русских букв ENABLE_PROFILE разрешение использовать в списке ID_PROFILE номер профиля NOTE_PROFILE описание профиля DATE_PROFILE дата профиля CRC16_PROFILE контрольная сумма профиля NUM_PROFILE — максимальное число профилей		+
set_Profile	параметр =значение	Установить отдельный параметр профиля. Разрешены следующие имена параметров: NAME_PROFILE имя профиля до 12 русских букв. ENABLE_PROFILE разрешение использовать в списке ID_PROFILE номер профиля NOTE_PROFILE описание профиля до 40		+

		<p>русских букв</p> <p>DATE_PROFILE дата профиля</p> <p>CRC16_PROFILE контрольная сумма профиля</p> <p>NUM_PROFILE — максимальное число профилей</p>		
get_listProfile	нет	Получить список профилей с указанием выбранного. Профиль может быть не выбран (не входит в список разрешенных профилей)	get_listProfile; возврат Профиль 2:0;Профиль 3:1;	+
set_listProfile	Номер профиля	Установить профиль из списка. Возвращает новый список профилей. Внимание! Сразу происходит сохранение выбранного профиля как текущего (запрос set_SAVE сразу выполняется)!	set_listProfile(0); возврат Профиль 2:1;Профиль 3:0;	+
<b>Работа с частотными датчиками (датчики потока)</b>				
get_Flow	Имя датчика	Получить значение потока (в кубометрах в час) по имени датчика	get_Flow(FLOWCON) возврат 0.987	+
get_frFlow	Имя датчика	Получить значение частоты датчика по имени датчика	get_frFlow(FLOWCON) возврат 13.345	+
get_minFlow	Имя датчика	Получить значение минимально допустимого потока потока (в кубометрах в час) по имени датчика	get_minFlow(FLOWCON) возврат 0.2	+
get_noteFlow	Имя датчика	Получить описание датчика по имени датчика	get_noteFlow(FLOWCON) возврат Датчик потока по конденсатору	+
get_pinFlow	Имя датчика	Получить ногу куда прицелен датчик по имени датчика	get_pinFlow(FLOWCON) возврат D35	+
get_kfFlow	Имя датчика	Получить коэффициент пересчета частоты в поток по имени датчика	get_kfFlow(FLOWCON) возврат 0.43	+
get_errcodeFlow	Имя датчика	Получить код ошибки по имени датчика	get_errcodeFlow(FLOWCON) возврат 0	+
get_testFlow	Имя датчика	Получить значение потока (в кубометрах в час)при тестировании по имени датчика	get_testFlow(FLOWCON) возврат 0.7	+
set_kfFlow	Имя датчика= новое значение	<u>Установить</u> коэффициент пересчета частоты в поток по имени датчика	set_kfFlow(FLOWCON=0.5) возврат 0.5	+
set_testFlow	Имя датчика= новое значение	<u>Установить</u> значение потока (в кубометрах в час)при тестировании по имени датчика	set_testFlow(FLOWCON=0.998) возврат 0.998	+
get_capacityFlow	Имя датчика	Получить значение теплоемкости в контуре где стоит датчик Дж/(кг·град)	get_capacityFlow(FLOWCON) возврат 4123	+
set_capacityFlow	Имя датчика=	Установить значение теплоемкости в контуре где	set_capacityFlow(FLOWCON=3800)	+

	новое значение	стоять датчик Дж/(кг·град)	возврат 3800	
<b>Работа с электросчетчиком SDM120/630</b>				
get_SDM	параметр	<p>Получить значение параметра счетчика.</p> <p>Определены следующие параметры:</p> <ol style="list-style-type: none"> <li>1. NAME_SDM - Имя счетчика</li> <li>2. NOTE_SDM - Описание счетчика</li> <li>3. MAX_VOLTAGE_SDM - Контроль напряжения max</li> <li>4. MIN_VOLTAGE_SDM - Контроль напряжения min</li> <li>5. MAX_POWER_SDM - Контроль мощности</li> <li>6. VOLTAGE_SDM - Напряжение</li> <li>7. CURRENT_SDM - Ток</li> <li>8. REPOWER_SDM - Реактивная мощность</li> <li>9. ACPOWER_SDM - Активная мощность</li> <li>10. POWER_SDM- Полная мощность</li> <li>11. POW_FACTOR_SDM - Коэффициент мощности</li> <li>12. PHASE_SDM- Угол фазы (градусы)</li> <li>13. IACENERGY_SDM - Потребленная активная энергия</li> <li>14. EACENERGY_SDM - Переданная активная энергия</li> <li>15. IREENERGY_SDM - Потребленная реактивная энергия</li> <li>16. IREENERGY_SDM - Переданная реактивная энергия</li> <li>17. ACENERGY_SDM - Суммарная активная энергия</li> <li>18. REENERGY_SDM- Суммарная реактивная энергия</li> <li>19. ENERGY_SDM — Суммарная энергия</li> <li>20. LINK_SDM — состояние связи со счетчиком</li> </ol>		+
set_SDM		<p>Установить значение параметра счетчика.</p> <p>Имеет смысл для 3, 4 , 5 пункта, эти параметры можно установить, остальные параметры только чтение и по этому запрос игнорируются, ошибка при этом не возникает.</p> <p>Определены следующие параметры:</p> <ol style="list-style-type: none"> <li>1. NAME_SDM - Имя счетчика</li> <li>2. NOTE_SDM - Описание счетчика</li> <li>3. MAX_VOLTAGE_SDM - Контроль напряжения max</li> <li>4. MIN_VOLTAGE_SDM - Контроль напряжения min</li> <li>5. MAX_POWER_SDM - Контроль мощности</li> <li>6. VOLTAGE_SDM - Напряжение</li> <li>7. CURRENT_SDM - Ток</li> <li>8. REPOWER_SDM - Реактивная мощность</li> <li>9. ACPOWER_SDM - Активная мощность</li> <li>10. POWER_SDM- Полная мощность</li> <li>11. POW_FACTOR_SDM - Коэффициент мощности</li> </ol>		+

		12. PHASE_SDM- Угол фазы (градусы) 13. IACENERGY_SDM - Потребленная активная энергия 14. EACENERGY_SDM - Переданная активная энергия 15. IREENERGY_SDM - Потребленная реактивная энергия 16. IREENERGY_SDM - Переданная реактивная энергия 17. ACENERGY_SDM - Суммарная активная энергия 18. REENERGY_SDM- Суммарная реактивная энергия 19. ENERGY_SDM — Суммарная энергия 20. LINK_SDM — состояние связи со счетчиком		
uplinkSDM	нет	Попытаться возобновить связь со счетчиком при ее потере		+
settingSDM	нет	Запрограммировать параметры связи с заводских на требуемые (заводские скорость 2400, адрес 1), скорость 9600, адрес 2. Счетчик при этом должен находиться в режиме ПРОГРАММИРОВАНИЯ (переводится долгим нажатием на кнопку до появления на экране счетчика надписи «-Set-») 		+

## Работа с MQTT сервером, настройка клиента

set_MQTT	параметр	Установить параметр клиента из строки. Определены следующие параметры: USE_MQTT - флаг разрешения отправки MQTT USE_MQTT — Флаг отправки на сервис ThingSpeak BIG_MQTT - флаг отправки ДОПОЛНИТЕЛЬНЫХ данных на MQTT SDM_MQTT - флаг отправки данных электросчетчика на MQTT FC_MQTT - флаг отправки данных инвертора на MQTT COP_MQTT - флаг отправки данных COP на MQTT TIME_MQTT период отправки на сервер минуты. 0...1000 ADR_MQTT - Адрес сервера IP_MQTT - IP Адрес сервера только чтение (рассчитывается от ADR_MQTT) PORT_MQTT- Порт сервера LOGIN_MQTT- логин сервера PASSWORD_MQTT- пароль сервера ID_MQTT - Идентификатор клиента на MQTT сервере  USE_NARMON - флаг отправки данных на народный мониторинг BIG_NARMON- флаг отправки данных на		+
----------	----------	--	--	---

		<p>народный мониторинг ,расширенная версия ADR_NARMON - Адрес сервера народный мониторинг IP_NARMON - IP Адрес сервера только чтение (рассчитывается от ADR_NARMON) народный мониторинг PORT_NARMON- Порт сервера народный мониторинг LOGIN_NARMON- логин сервера народный мониторинг PASSWORD_NARMON- пароль сервера народный мониторинг</p>		
get_MQTT	параметр	<p>Установить параметр клиента из строки. Определены следующие параметры: USE_MQTT - флаг разрешения отправки MQTT USE_MQTT — Флаг отправки на сервис ThingSpeak BIG_MQTT - флаг отправки ДОПОЛНИТЕЛЬНЫХ данных на MQTT SDM_MQTT - флаг отправки данных электросчетчика на MQTT FC_MQTT - флаг отправки данных инвертора на MQTT COP_MQTT - флаг отправки данных COP на MQTT TIME_MQTT период отправки на сервер минуты. 0...1000 ADR_MQTT - Адрес сервера IP_MQTT - IP Адрес сервера только чтение (рассчитывается от ADR_MQTT) PORT_MQTT- Порт сервера LOGIN_MQTT- логин сервера PASSWORD_MQTT- пароль сервера ID_MQTT - Идентификатор клиента на MQTT сервере</p> <p>USE_NARMON - флаг отправки данных на народный мониторинг BIG_NARMON- флаг отправки данных на народный мониторинг ,расширенная версия ADR_NARMON - Адрес сервера народный мониторинг IP_NARMON - IP Адрес сервера только чтение (рассчитывается от ADR_NARMON) народный мониторинг PORT_NARMON- Порт сервера народный мониторинг LOGIN_NARMON- логин сервера народный мониторинг PASSWORD_NARMON- пароль сервера народный мониторинг</p>		+
<b>Работа с статистикой работы ТН (максимум 364 дня) только для eeprom i2c 64 кб</b>				
get_listStat	нет	<p>Получить список доступных «статистик» Определены следующие имена (максимальный</p>	get_listStat возврат	+

		<p>список, если датчика нет то он пропускается в списке)</p> <p>0. none — нет графика</p> <p>1. Tin -средняя температура дома</p> <p>2. Tout - средняя температура улицы</p> <p>3. Tboiler - средняя температура бойлера</p> <p>4. Hour - число накопленных часов должно быть 24</p> <p>5. Hmoto - моточасы компрессора за сутки</p> <p>6. EnergyCO - выработанная энергия</p> <p>7. Energy220 -потраченная энергия</p> <p>8. -COP- - КОП</p> <p>9. PowerCO - средняя мощность СО</p> <p>10. Power220 - средняя потребляемая мощность</p> <p>разделитель “;”</p>	<p>none:1;Tin:0;Tout:0 ;Tboiler:0;Hour:0;H moto:0;EnergyCO:0;E nergy220:0;- COP-:0;PowerCO:0;Po wer220:0;</p>	
get_Stat	параметр	<p>Получить данные одного графика по имени из списка полученного в get_listStat, возвращается строка, где через «;» перечислены точки. Число точек не определено!</p> <p>Формат одной точки включает дату и значение ДДММГГ:ЗНАЧЕНИЕ;</p> <p>день и месяц всегда имеют две цифры (есть 0 слева если нужно), год только ПОСЛЕДНИЕ две цифры (валидны года с 1950 по 2050).</p> <p>разделитель между датой и значением «:»</p> <p>разделитель между отдельными точками «;»</p> <p>None — возвращает пустую строку</p>	<p>get_Stat(Tout) возврат 010118:10.12;020118:1 0.24;030118:10.45;</p>	+
get_infoStat	нет	Получить информацию об накопленной статистике.	<p>get_infoStat возврат Максимальный объем накапливаемой статистики (дни)  364;Объем накопленной статистики (дни) 364;Начальная дата статистики  12/1/2017;Конечная дата статистики 10/1/2018;Позиция для записи 0;</p>	+
set_testStat	нет	<p>Сгенерировать тестовую статистику. Все данные стираются и записываются случайные данные на весь размер данных (364 дня).</p> <p>Работает только на выключенном ТН</p>	<p>set_testStat возврат</p>	+
RESET_STAT	нет	<p>Форматировать (очистить) статистику. Работает только на выключенном ТН</p>	<p>RESET_STAT возврат Форматирование I2C EEPROM статистики, ожидайте 10 сек . . .</p>	+

## Запросы работы с расписанием

set_SCHDLR	Параметр =значение	<p>Установить значение параметра</p> <p>Определены следующие параметры</p> <p>1. On — включить выключить текущее расписание</p> <p>2 Active — активное расписание</p> <p>3. IstNames — список имен расписаний</p> <p>4. NameX - имя</p> <p>5. CalendarX - информация о расписании (сами данные)</p> <p>X – номер расписания от 0, если пусто, то текущее</p>	<p>set_SCHDLR(On=1) возврат 1</p>	+
get_SCHDLR	параметр	<p>Получить значение параметра</p> <p>Определены следующие параметры</p>	<p>get_SCHDLR(On) возврат</p>	+

		1. On — включить/выключить текущее расписание 2 Active — активное расписание 3. lstNames — список имен расписаний 4. NameX - имя 5. CalendarX - информация о расписании (сами данные) X – номер расписания от 0, если пусто, то текущее	1	
set_SAVE_SCHDLR	нет	Сохранить ТОЛЬКО расписание. Возвращает код ошибки (меньше 0) или 0 в случае успеха	set_SAVE_SCHDLR возврат 0	+

### Заранее определенные файлы. Загрузка/выгрузка файлов в контроллер.

При стандартном запросе **get определенных заранее** файлов, контроллер создает их на лету и выдает их для загрузки. Такой механизм обеспечивает выгрузку/загрузку файлов в контроллер. Определены следующие имена файлов, которые создает контроллер а не грузит с sd карты.

Имя	Тип	Описание	Ок
state.txt	текст	Получить «мгновенное» состояние ТН	+
settings.txt	текст	Получить текущие настройки ТН	+
settings.bin	бинарный	Получить копию настроек из EEPROM	+
journal.txt	текст	Получить копию системного лога	+
test.dat	бинарный	Файл размером 2 мБ для тестирования скорости работы	+
chart.csv	текст	Файл содержащий данные по ВСЕМ графикам из RAM	+
chart_sd.csv	текст	Файл содержащий данные по ВСЕМ графикам с карточки памяти (если включена соответствующая опция)	+
statistic.csv	текст	Файл всей накопленной дневной статистики (максимум 364 дня)	+

### Модификация имен файлов в зависимости от конфигурации.

Контроллер поддерживает модификацию имен запрашиваемых файлов, т.е. заменяет аргумент (он выделяется [] скобками) значением установленным в прошивке. Например:

Веб морда запрашивает файл **plan[HPscheme].png** контроллер видит аргумент **Hpscheme** в скобках, аргумент определен (есть в списке разрешенных) и это приводит к замене его на дефайн HP\_SHEME (равный допустим 2), в итоге получается файл **plan2.png**, который и ищется на диске.

Аргумент	Дефайн в коде	Описание	Ок
HPscheme	HP_SHEME	Номер схемы для веб морды которая отображается на экране	+

## Дисплей Nixtion

### 1. Почитать

[https://www.itead.cc/wiki/Nextion\\_Instruction\\_Set#click: Activate component.27s\\_press.2Frelease event](https://www.itead.cc/wiki/Nextion_Instruction_Set#click: Activate component.27s_press.2Frelease event)  
<http://support.iteadstudio.com/support/discussions/1000058038>  
<http://forum.amperka.ru/threads/nextion-esp8266-%D0%90%D0%B7%D1%8B-arduino.9204/>  
<http://robotclass.ru/tutorials/arduino-nextion-tjc/>  
<https://geektimes.ru/company/flprog/blog/273868/>  
<http://mysku.ru/blog/china-stores/39509.html>

### 2. Основная идея.

На дисплее будет несколько страниц.

Навигация между страницами осуществляется средствами дисплея (контролл про это ничего не знает и не управляет). При смене страницы дисплей выдает в последовательный порт текущий номер страницы (скорее всего это команда sendme).



Взаимодействие между дисплеем и контроллером происходит двумя способами:

### 2.1. Вывод значений на экран дисплея (обновление). Контролл->>дисплей

Большинство полей это текстовые лейблы. Контроллер зная на какой страницы находится дисплей (см предыдущий абзац), и зная что надо выводить на этой странице посылает команды на заполнение поля.

Например. На текущей странице есть лейбл в котором должна отображаться текущая температура в доме, (имя tin). Контроллер посылает команду tin.text=>21.3». Контроллер сам понимает что и когда обновлять. Очень важно согласовать имена и назначение полей (см.п.3).

### 2.2. Изменение значений с использованием экрана. Контролл<<-дисплей

На первом этапе предлагается сделать простейший ввод в виде двух кнопок «+» и «-», при нажатии на кнопку дисплей шлет событие в контролл click ИМЯ, СОБЫТИЕ. Контролл понимает что произошло, обновляет переменную и посылает команду на обновление переменной в дисплее см п.2.2

Второй вариант Дисплей самостоятельно изменяет переменную и по кнопке примет посылает команду click ИМЯ, СОБЫТИЕ. Контролл понимает что надо обновить переменную и запрашивает ее значение командой get ИМЯ.

## 3. Описание полей

В таблицу заносятся ТОЛЬКО те поля которые изменяются или меняют что либо. Просто статические текстовые поля в таблицу не заносятся.

Страница	Имя поля	Описание	Обновление	Тип	События
<i>Главная страница, загружается по умолчанию (page 0)</i>					
Page 0	t0	Температура в доме (TIN)	+	текст	нет
Page 0	t1	Температура целевая в доме	-	текст	нет
Page 0	t2	Температура на улице (TOUT)	+	текст	нет
Page 0	t3	Температура бойлера (TBOILER)	+	текст	нет
Page 0	t4	Температура из геоконтур (TEVAING)	+	текст	нет
Page 0	t5	Температура обратки (TRET)	+	текст	нет
Page 0	bt0	Кнопка включения выключения ТН. Есть две картинки (вкл и выкл) включить «bt0.val=0» выключить «bt0.val=1» Дисплей ничего не делает, только отдает событие	?	кнопка	отпускание
<i>Строка статуса едина для всех экранов (присутствует на всех эранах)</i>					
Все страницы	tninc	Картинка «ТН включен» Команда скрыть "vis tninc,0" Команда показать "vis tninc,1"	+	Картика	нет
Все страницы	tnoff	Картинка «ТН выключен» Команда скрыть "vis tnoff,0" Команда показать "vis tnoff,1"	+	Картика	нет
Все страницы	options	Картинка «Параметры ОК» Команда скрыть "vis options,0" Команда показать "vis options,1"	+	Картика	нет
Все страницы	fault	Картинка «Ошибка» Команда скрыть "vis fault,0" Команда показать "vis fault,1"	+	Картика	нет

Все страницы	heat	Картинка «Отопление» Команда скрыть "vis heat,0" Команда показать "vis heat,1"	+	Картика	нет
Все страницы	cool	Картинка «Охлаждение» Команда скрыть "vis cool,0" Команда показать "vis cool,1"	+	Картика	нет
Все страницы	gvs	Картинка «ГВС» Команда скрыть "vis gvs,0" Команда показать "vis gvs,1"	+	Картика	нет
Все страницы	onlygvs	Картинка «только ГВС» Команда скрыть "vis onlygvs,0" Команда показать "vis onlygvs,1"	+	Картика	нет
Все страницы	time	Вывод текущего времени	+	текст	нет

Страница меню настроек (page 1)

		Активных элементов нет! Нет взаимодействия с контроллером. Есть кнопки 1. Отопление/охлаждение 2. Горячее водоснабжение 3. Сетевые настройки 4. Пароли 5. Система 6. О контроллере 7. Выход			

## Инвертор OMRON MX2

Требуемые функции для управления частотником

№	Код функции	Регистр	Режим	Описание	Тип и длина
1	нет	0003h	ч	Состояние ПЧ А 0: Начальное состояние 2: Остановка 3: Вращение 4: Остановка с выбегом 5: Толчковый ход 6: Торможение постоянным током 7: Выполнение повторной попытки 8: Аварийное отключение 9: Пониженное напряжение (UV	Регистр 1 слово
2	нет	0001h	Ч/З	Команда «Ход»	бит

				1: Ход, 0: Стоп (действительно при A002 = 03)	
3	F001	0001h 0002h	Ч/З	Источник задания частоты (0,01 [Гц]) 0...40000 (действительно только при A001 = 03)	Регистр 2 слова
4	d001	1001h 1002h	Ч	Контроль выходной частоты (0,01 [Гц]) 0...40000	Регистр 2 слова
5	d002	1003h	Ч	Контроль выходного тока (0,01 [А]) 0...65530	Регистр 1 слово
6	d013	1011h	Ч	Контроль выходного напряжения 0,1 [В] 0 . . . 6000	Регистр 1 слово
7	d014	1012h	Ч	Контроль мощности 0,1 [кВт] 0 . . . 1000	Регистр 1 слово
8	d018	1019h	Ч	Контроль температуры радиатора (0.1 градус) -200...1500	Регистр 1 слово
9	d102	1026h	Ч	Контроль напряжения постоянного тока (P-N) 0,1 [В] 0...10000	Регистр 1 слово
10	d080	0011h	Ч	Счетчик аварийных отключений 0...65530	Регистр 1 слово
11	d081	0012h	Ч	Описалово 1 отключения	Регистр 9 слов
12	d082	001Ch	Ч	Описалово 2 отключения	Регистр 9 слов
13	d083	0026h	Ч	Описалово 3 отключения	Регистр 9 слов
14	d084	0030h	Ч	Описалово 4 отключения	Регистр 9 слов
15	d085	003Ah	Ч	Описалово 5 отключения	Регистр 9 слов
16	d086	0044h	Ч	Описалово 6 отключения	Регистр 9 слов
17	нет	0004h	Ч/З	Сброс аварийного отключения (RS) 1: Сброс	Бит
18	нет	0011h	Ч	Готовность ПЧ 1: Готов, 0: Не готов	Бит
19	нет	0010h	Ч	Направление вращения 1: Обратное вращение, 0: Вращение в прямом направлении (взаимоблокировка с "d003")	бит
20	нет	0002h	Ч/З	Команда направления вращения 1: Обратное вращение, 0: Вращение в прямом направлении (действительно при A002 = 03)	Бит
21					
22					
23					
24					

# ОПИСАНИЕ ЖЕЛЕЗА

## для сборки самостоятельно, без готовых печатных плат

Версия 0.830 от 11.06.2017

Описание «Народного контроллера ТН» <https://www.forumhouse.ru/threads/352693/>

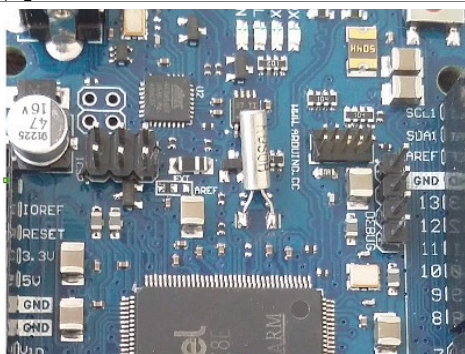
Один из вариантов списка «закупок» <https://www.forumhouse.ru/posts/15801200/>

### 1. Железо.(список закупок)

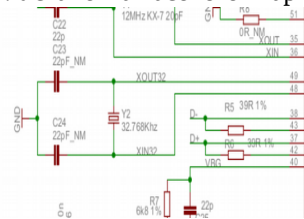
#### 1.1 Ардуино дуе.

Подойдет фактически любая. Желательно что бы был распаян часовой кварц, тогда можно сразу запустить внутренние часы чипа sm3x. Есть обязательные доработки.

Доработка



#### 1. Установка часового кварца



Необходимо припаять часовой кварц на пустые контактные площадки. Крайне желательно уставить 20пф конденсаторы рядом (см площадки). Если генератор не «заводится» то контроллер не стартует.

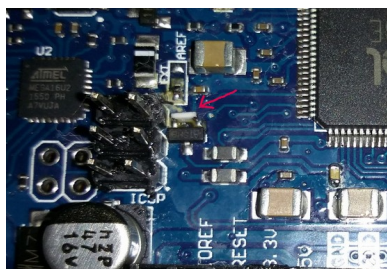
Выбор в программе типа генератора (файл

Control.ino)

`RTC_clock rtc_clock(RC);` // Внутренние часы, используется внутренний RC генератор

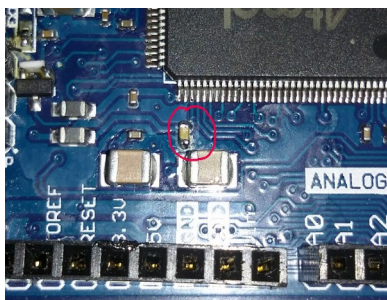
`//RTC_clock rtc_clock(XTAL);` // Внутренние часы, используется часовой кварц

Доработка не обязательная, при использовании внутренней RC цепи необходимо на странице «Дата/Время» установить флаг «Синхронизация раз в час с I2C часами\*»



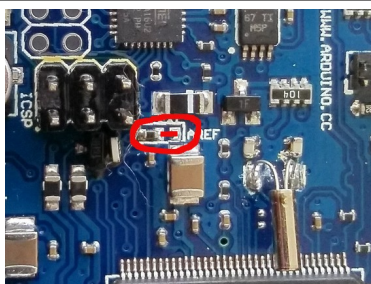
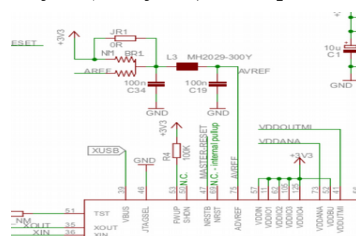
#### 2. Обеспечение «Стабильного» сброса при подаче питания.

Припаять резистор поверхностного монтажа (0603 размер) номиналом 10 кОм между ERASE линией и + 3,3В, Проще всего припаять к двум ногам Т3. Доработка крайне желательна.



#### 3. Улучшение сигнала «Опорного напряжения».

Заменить конденсатор C19 на как можно больше емкости (тантал желателен) 10-50 мкф. Доработка желательна, если нет источника опорного напряжения (см следующий пункт) то доработка крайне желательна.



#### 4. Если планируется использовать внешнюю опору (плата rav2000), необходимо перепаять перемычку как показано на рисунке.

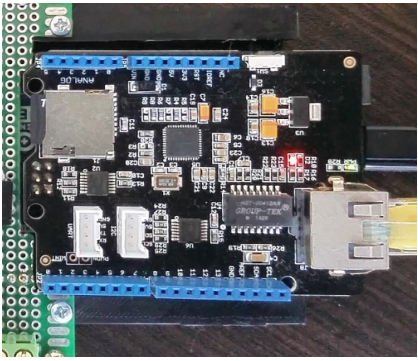
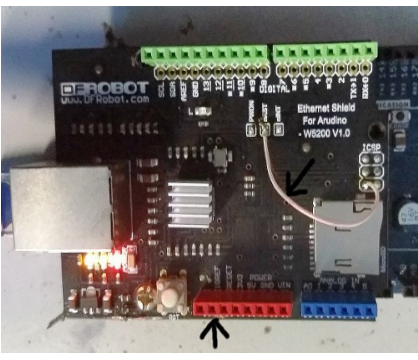
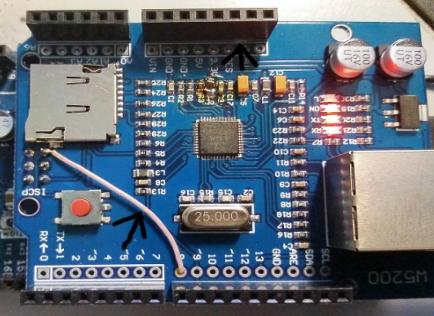
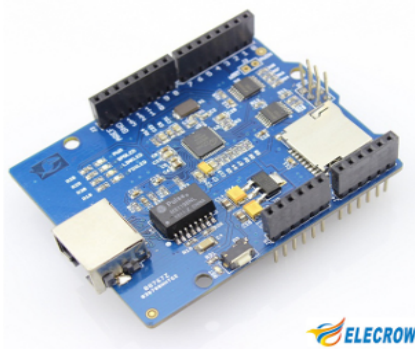
Внешнюю опору, например ADR4530, взяв схему с документации, можно подключить к DUE на ногу AREF.

Нужно изменить define в config.h: EXTERNAL\_AREF и SAM3X\_ADC\_REF.


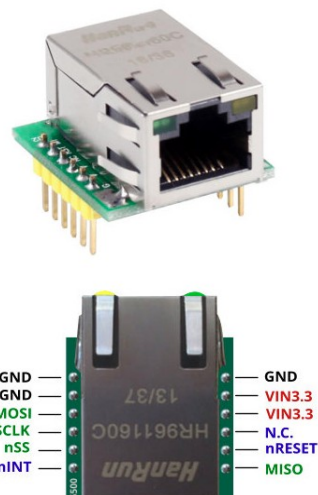
## 1.2 Сетевой шилд.

Можно использовать шилды на основе чипов wiznet w5100 (не проверялось), w5200, w5500. Рекомендую использовать платы с буферами. Есть обязательные доработки (у разных шилдов по разному). По скорости 5100 совсем медленный (там все криво сделано) 5200 и 5500 примерно одинаковы. Основные отличия 5200 от 5500 — тип корпуса и энергопотребление.

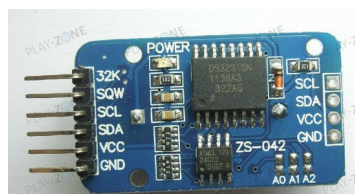
Сеть доработка:

1		Плата №1 работает на 21 мГц. Есть буфера. Заработала сразу, переделок не потребовала, но есть подозрение что нужно ioref резать. Сброс сделан правильно.
2		Плата №2 (сейчас стоит у меня на демо) сразу не заработала, была куплена первой, убил массу времени на нее. Опускаю описание ошибок монтажа и не исправных деталей. Требуется переделки. - IOREF был соединен с 5 вольтами по по этому на w5200 перло 5 вольт. Как не сгорела не понятно. Надо резать дорожку. Исправления показаны стрелками. Проверена в работе на 42, 24, 21 мГц - работает на всех частотах. Буферов нет. -в цепи CS SD стоит делитель, его скорее всего надо убрать. - проверить номинал резистора на вывод чипа bias (нога 12). Должен быть 28.7к 1% (код 45C) <b>СБРОС пробрасывать не надо (011116)!</b>
3		Плата №3 (есть у меня и dobrinia) была куплена последней, на али. Самая интересная плата (скорее всего она будет основной при повторении контроллера). С учетом опыта по плате 2 сразу были сделаны переделки аналогичные плате 2. Исправления показаны стрелками. Проверена в работе на 42, 24, 21 мГц - работает на всех частотах. Буферов нет. <b>СБРОС пробрасывать не надо (011116)!</b> -в цепи CS SD стоит делитель, его скорее всего надо убрать. - проверить номинал резистора на вывод чипа bias (нога 12). Должен быть 28.7к 1% (код 45C)
4		Плата №4 (находится у Sheeny) подробное описание платы <a href="https://feilipu.me/2013/08/03/wiznet-w5200-arduino-shield/">https://feilipu.me/2013/08/03/wiznet-w5200-arduino-shield/</a>  Доработка: удалить резистор R24, закоротить его контакты. Закоротить контакты 1 и 2 микросхемы U6 (работает и без этой доработки, но я сделал) Фото доработки <a href="https://feilipu.files.wordpress.com/2013/08/p1040402crop.jpg">https://feilipu.files.wordpress.com/2013/08/p1040402crop.jpg</a>  схема <a href="http://www.elecrow.com/wiki/images/c/c8/Ethernet_Shield_v1.1.pdf">http://www.elecrow.com/wiki/images/c/c8/Ethernet_Shield_v1.1.pdf</a>



5		<p>Плата №5 Чип <b>w5500</b> (сейчас стоит у меня на ТН)          Требуется конфигурирование библиотеки на этот чип          Заработала без переделок. Есть буфера, скорость до 21 МГц. На 28 МГц не работает UDP  <b>ВАЖНО!</b> Настройка библиотеки Ethernet на этот чип. По умолчанию я выкладываю на w5200</p>
6		<p>Плата №6 <b>W5500 Lite</b> (стоит на обеих платах)</p>

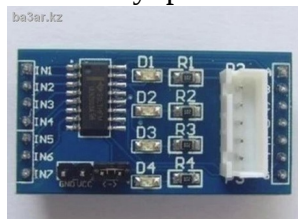
### 1.3 Плата часов реального времени и чип еепром.



Используется плата с микросхемой DS3231 и еепром на 4 кб. Другие чипы часов не поддерживаются. Доработок нет, связь с контроллером по i2c шине. У меня стоит вот такая — см рисунок.

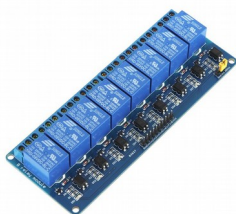
В специальных платах НК (Добрыни и rav2000) память расширена до 64 кб (размер определяется дефайном). В дополнительное место идет запись журнала и статистики. Vad711 адаптировал библиотеку для использования fram памяти 64 кб.

### 1.4 Плата управления ключами для шагового двигателя ЭРВ.



Используется микросхема ULN2003. Есть много вариантов плат. Подойдет любая. Доработок не требует. У меня стоит см фото. Не забываем подать на нее 12 вольт питания для ЭРВ

### 1.5 Плата реле.



Мне хватает 8 реле. Но уже все заняты. Есть желание перейти на твердотелку. Которые не генерируют помех в момент переключения индуктивной нагрузки. Питание реле должно быть 5 вольтовым. Сейчас использую вот такую плату:  
 В дальнейшем точно буду переходить на твердотелку. Что то типа mos3041+ bt136.

### 1.6. Датчики температуры



Общее их число определяется в зависимости от конфигурации ТН. У меня сейчас стоит 13 датчиков (но наверное это избыточно). Можно использовать ds1820 ds18s20 ds18b20. Рекомендуется использовать ds18b20. Желательно использовать датчики из одной партии. Имеют несколько вариантов маркировок по цвету проводов (см инет). Я использую в герметичном исполнении в гильзах (см фото), но можно использовать и «голые» датчики. При подключении используется не «паразитное» питание.

Датчики, для уменьшения количества ошибок и для более надежной работы, нужно подключать через мост I2C – 1-Wire DS2482.

DS2482 установлена на плате rav2000. Есть плата расширения от vad711, с двумя шинами 1-Wire на DS2482, одна из которых имеет паразитное питание. В крайнем случае, можно распаять DS2482 на макетной плате, не забыв конвертер уровней 3.3V – 5V. Схему взять из документации к DS2482.

#### 1.7 Датчик давления. (если есть в конфигурации)

Подойдет с выходом 4-20мА или с выходом напряжения 0-5 вольт. Требуется дополнительное согласование для обоих вариантов. Диапазон работы определяется по характеристикам холодильной части ТН.

1.8 Блок питания 12 вольт 2 ампера. Подойдет любой желательно с большим входным диапазоном напряжений (например 95-240 вольт). Блок питания должен быть качественным.

1.9 Дисплей Nixton (если хотите использовать) (<https://www.itead.cc/wiki/Product>) используется с диагональю 4.3 дюйма вот этот : <https://www.itead.cc/wiki/NX4827T043>

**ОБЯЗАТЕЛЬНО** префикс модели NX, с китайским вариантом будут проблемы с редактором.

Цепляется к первому (Serial1) аппаратному порту uart DUE. (ноги D18 D19) (перекрестное соединение TX-RX). Не смотря на 5 вольтовое питание выходные сигналы с дисплея имеют уровень 3.3 вольта и не требуют согласования уровней.

#### 1. Почитать

[https://www.itead.cc/wiki/Nextion\\_Instruction\\_Set#click: Activate component.27s\\_press.2Frelease\\_event](https://www.itead.cc/wiki/Nextion_Instruction_Set#click: Activate component.27s_press.2Frelease_event)

<http://support.iteadstudio.com/support/discussions/1000058038>

<http://forum.amperka.ru/threads/nextion-esp8266-%D0%90%D0%B7%D1%8B-arduino.9204/>

<http://robotclass.ru/tutorials/arduino-nextion-tjc/>

<https://geektimes.ru/company/flprog/blog/273868/>

<http://mysku.ru/blog/china-stores/39509.html>

#### 2. Основная идея.

На дисплее будет несколько страниц.

Навигация между страницами осуществляется средствами дисплея (контролл про это ничего не знает и не управляет). При смене страницы дисплей выдает в последовательный порт текущий номер страницы (скорее всего это команда sendme).

Взаимодействие между дисплеем и контроллом происходит двумя способами:

2.1. Вывод значений на экран дисплея (обновление). Контролл->>дисплей

Большинство полей это текстовые лейблы. Контроллер зная на какой странице находится дисплей (см предыдущий абзац), и зная что надо выводить на этой странице посылает команды на заполнение поля.

Например. На текущей странице есть лейбл в котором должна отображаться текущая температура в доме, (имя tin). Контроллер посылает команду tin.text=>21.3». Контроллер сам понимает что и когда обновлять. Очень важно согласовать имена и назначение полей (см.п.3).

2.2. Изменение значений с использованием экрана. Контролл<<-дисплей

На первом этапе предлагается сделать простейший ввод в виде двух кнопок «+» и «-», при нажатии на кнопку дисплей шлет событие в контролл click ИМЯ, СОБЫТИЕ. Контролл понимает что произошло, обновляет переменную и посылает команду на обновление переменной в дисплее см п.2.2

Второй вариант Дисплей самостоятельно изменяет переменную и по кнопке примет посылает команду click ИМЯ, СОБЫТИЕ. Контролл понимает что надо обновить переменную и запрашивает ее



значение командой get ИМЯ.

1.10. Стабилизатор напряжения LM1117 для получения напряжения 3.3 вольта из 5. + электролитически конденсаторы 1000 мкф на 16 вольт.

1.11 Прочие детали по необходимости и желанию. Резисторы, конденсаторы, оптроны для развязки входов.

1.12 Конденсаторы блокировочные 600 вольт 0.15 -0.5 мкф для гашения помех на индуктивной нагрузке блока реле. Провод силовой экранированный.

1.13 Подключение периферии к плате контроллера.

Описание (наиболее свежее) куда припаивать периферию находится в файле config.h Пример описания (+ в комментарии это моя конфигурация):

```
// -----
// ЖЕЛЕЗО - привязка к ногам контроллера
// -----
// датчики
#define PIN_ONE_WIRE_BUS 23 // + нога с интерфейсом OneWire BCE температурные датчики
#define ADC_SENSOR_PEVA 11 // + НОМЕР КАНАЛА ацп (внимание - в нумерации sam3x!) нога куда прицеплен
датчик давления PEVA
#define ADC_SENSOR_PCON 10 // - НОМЕР канала ацп (в нумерации sam3x!) нога куда прицеплен датчик давления
PCON
#define PIN_SENSOR_SEVA 28 // + Датчик протока по испарителю
#define PIN_SENSOR_SLOWP 29 // + Датчик низкого давления
#define PIN_SENSOR_SHIGHP 30 // + Датчик высокого давления
// Сервис
#define PIN_WIRE_NET 45 // + Перемычка(вход), при установке в 0 при старте делает настройку сети по умолчанию
192.168.1.177 шлюз 192.168.1.1
#define PIN_WIRE_PW 44 // + Перемычка (вход), при установке в 0 не спрашивает пароль на вход в веб морду
#define PIN_LED 43 // + Выход на светодиод мигает 0.5 герца - ОК с частотой 2 герца ошибка
#define PIN_BEEP 42 // + Выход на пищалку
// устройства
#define PIN_DEVICE_RCOMP 46 // + Реле включения компрессора (через пускатель)
#define PIN_DEVICE_RPUMPI 47 // + Реле включения насоса входного контура (геоконтур)
#define PIN_DEVICE_RPUMPO 48 // + Реле включения насоса выходного контура (отопление и ГВС)
#define PIN_DEVICE_RBOILER 49 // + Включение ТЭНа бойлера
#define PIN_DEVICE_RTRV 50 // + 4-ходовой клапан
#define PIN_DEVICE_RFAN1 44 // - Реле включения вентилятора испарителя №1
#define PIN_DEVICE_RFAN2 45 // - Реле включения вентилятора испарителя №2
#define PIN_DEVICE_R3WAY 52 // + Трех ходовой кран. Переключение системы СО — ГВС (что сейчас греть)
#define PIN_DEVICE_REVI 51 // + Соленоид для EVI. (испаритель ниже +3гр и конденсатор выше +40гр)
#define PIN_DEVICE_RHEAT 43 // - Включение ТЭНа СО (электродкотел), может использоваться как догрев, резерв и т.д.
#define PIN_DEVICE_RPUMPB 53 // + Реле насоса циркуляции бойлера (ГВС)

//ЭРВ - требуется четыре провода Общий синий
#define PIN_DEVICE_EEV_ORANGE 24 // + Фаза А оранжевый провод
#define PIN_DEVICE_EEV_YELLOW 25 // + Фаза ~А желтый
#define PIN_DEVICE_EEV_RED 26 // + Фаза В красный
#define PIN_DEVICE_EEV_BLACK 27 // + Фаза ~В Черный
//Частотный преобразователь
#define PIN_DEVICE_FC DAC0 // + DAC
```

По мере развития проекта ноги могут меняться.!!! надо проверять в свежих исходниках.

## 2. Монтаж

Рекомендации по монтажу контроллера. Борьба с помехами что нужно сделать (мой опыт)

1. Подтягивающий резистор на 1wire 1-2к
2. На все индуктивные нагрузки (RTRV REVI) поставить искрогасящие цепочки 100 ом+0.5 мкф 600 вольт (желательно около нагрузок а не на реле)
3. Подключение (провод) индуктивных нагрузок сделать в заземленном экране
4. Заземление.
5. Желательно разнести и экранировать контролл и блок реле (минимум разнести подальше)

6. На 1wire воткнуть защитные диоды Шотки
7. Длинные провода 1wire либо бросить витой парой либо экранировать.
8. Возможно заменить механические реле на твердотельные (проверить напряжение пробоя).
9. Конденсатор 0.1 мкф на питание 1wire. Желательно такие же конденсаторы на дальние датчики на питание.

Рекомендуется к прочтению <http://mk90.blogspot.ru/2011/04/1-wire.html>

Признаками помех является появление ошибок типа (которые уходят только после передергивая питания):

- Ошибка сброса на OneWire шине (обрыв или замыкание)
- Ошибка записи настроек в eeprom I2C
- Ошибка записи состояния в eeprom I2C
- Ошибка записи счетчиков в eeprom I2C
- Отказ чтения sd карты, и связанные с ней ошибки

### 3. Установка среды компиляции, библиотек, проекта

1. IDE 1.8.2 + DUE 1.6.4 (через board менеджер). Под другие ide не пробовал, если стоит другая IDE и что то не идет, То мой первый совет будет "Поставьте IDE 1.8.2". Наблюдал на DUE 1.6.5 проблемы под arm, разбираться не стал, откатился на DUE 1.6.4 (через board менеджер)

2. Берем ПОСЛЕДНИЙ мой архив. Там исходники, исправления IDE и все библиотеки.

2.1 Библиотеки - ставим в папку libraries в рабочей папке ардуино (путем ПОЛНОЙ замены если они существуют, я не гарантирую работу других проектов с моими либами), при этом IDE не должен быть запущен - при его старте сразу подгружаются библиотеки.

2.2. Исправления IDE - ищем нужные файлы и заменяем (IDE не должен быть запущен). Ищем именно под ARM (в директории куда установилась DUE 1.6.4 ) - под виндами нашел с трудом (скрытая папка), под Linux папка тоже скрытая arduino15 в директории пользователя. Файлы variant в архиве лежат для двух версий IDE (использовать надо 1604), его надо переименовывать перед копированием в просто variant.cpp

2.3 Проект в папку Control, рабочей папки Arduino.

Все должно компилироваться.

По ошибке - библиотеки Ethernet допилина, исправлен косяк с опросом сокетов. Если хотите использовать родную библиотеку Ethernet закомментируйте строку `#define FAST_LIB` - в файле config. h проекта.

Я правлю некоторые библиотеки. Причины - исправление ошибок и портирование на arm

Обратите внимание что часто программный сброс сетевой карты не проходит - надо нажимать кнопку ресет. (руки не доходят посмотреть узел сброса w5200)

### 4. Программа

4.1 Лог загрузки для инвертора демо (инвертор и счетчик отсутствуют)

Init I2C EEPROM journal . . .

I2C journal is ready for use

Scan I2C journal .....

START -----

Found journal I2C EEPROM: total size 36864 bytes, head=0x60da, tail=0x60d9

DEMO - DEMO - DEMO - DEMO - DEMO - DEMO - DEMO

Vesion firmware: 0.901 beta

Chip ID EXID: 677251680

Chip ID SAM3X8E: 51203120-3731374d-32303911-38313035

Last reason for reset SAM3x: Watchdog

Power +5V, +3.3V on board: ON

Last state FreeRTOS task+err\_code: 0x0000

Supply Controller Status Register [SUPC\_SR]: 0x00001080

Supply monitor ON, voltage: 3.2V

Control EEV driver L9333: ON

1. Setting and checking I2C device . . .

I2C bus init on 400 kHz - OK

I2C device found at address 0x18 - OneWire DS2482-100  
I2C device found at address 0x50 - EEPROM AT24CXXX 512 kBit  
I2C device found at address 0x68 - RTC DS3231  
2. Init Heat Pump main class . . .  
Init Modbus RTU via RS485: OK  
Invertor Omron MX2: present config  
\$ERROR source: Omron MX2, code: -39  
SDM120, no connect.  
SDM120, no connect.  
SDM120, no connect.  
SDM120, no connect.  
SDM120, no connect.  
Init I2C EEPROM statictic . . .  
Statistic found: is empty  
3. Read safe Network botton . . .  
Mode safeNetwork OFF  
4. Init and checking SD card . . .  
Initializing SD card...  
SUCCESS - SD card insert in slot.  
Repeat initializing SD card . . .  
SUCCESS - SD card initialized.  
SUCCESS - Found index.html file  
SD card info  
Manufacturer ID: 0x9c  
OEM ID: SO  
Serial number: 0x6c89a1a7  
Volume is FAT32  
blocksPerCluster: 8  
clusterCount: 1924940  
freeSpace: 7863.64 Mb  
5. Load data from EEPROM . . .  
Load counters from eeprom, read: 52 bytes  
Load setting from eeprom OK, read: 931 bytes crc16: 0xf5a0  
Load profile #0 from eeprom OK, read: 259 bytes crc16: 0x321f  
Hash user: dXNlcjphZG1pbG==  
Hash admin: YWRtaW46dXNlcg==  
6. Setting Network . . .  
Disable Ping block  
DEMO mode! Network library setting: W5500, ID chip: 4  
DHCP use: 0  
IP: 192.168.0.177  
Subnet: 255.255.255.0  
DNS: 192.168.0.191  
Gateway: 192.168.0.1  
MAC: de:a1:1e:02:02:02  
Speed Status: 100Mbps  
Duplex Status: full duplex  
SUCCESS: W5500 link ok, PHYSTATUS=0xBF  
7. Setting time and clock . . .  
Init internal RTC sam3x8e  
Set time internal RTC form i2c RTC DS3231: 13/12/2017 16:01:36  
Update time from NTP server: time.nist.gov  
time.nist.gov resolved to 132.163.97.3  
Send packet NTP, wait . . .  
Send packet NTP, wait . . .  
Send packet NTP, wait . . .  
Send packet NTP, wait . . .  
Set time from NTP server: 13/12/2017 16:01:41  
8. Message update IP from DNS . . .  
9. Client MQTT update IP from DNS . . .  
10. Init counter statictic.  
11. Delayed start Heat Pump: NO  
12. Start read ADC sensors  
13. Init Nextion dispaly  
14. Create tasks free RTOS . . .  
Create tasks - OK, size 13828 bytes

15. If you want to send a notification about resetting the controller . . .

16. Information about controller:

Ram used (bytes):

dynamic: 31972

static: 43180

stack: 112

Estimation free Ram: 23040

FREE MEMORY 9220 bytes

Temperature SAM3X8E: 28.06

Temperature DS2331: 31.50

Start Free RTOS scheduler :-))

READY -----

4.2. Для начала надо убедиться что на дуге эзернет работает.

Для этого на любой сборке (можно и 1.6.9) залить стандартный пример Web Server и посмотреть в лог какой адрес она получает. Адрес 0.0.0.0 показывает что обмен по спяю не идет.

Если все ок то двигаемся дальше, если нет то:

Проблемы со спаем, директория библиотеки эзернет файл w5100.cpp сейчас там (у меня)

```
// Set clock to 4Mhz (W5100 should support up to about 14Mhz)
```

```
// SPI.setClockDivider(SPI_CS, 21);
```

```
// SPI.setClockDivider(SPI_CS, 6); // 14 Mhz, ok
```

```
SPI.setClockDivider(SPI_CS, 4); // 21 Mhz, ok
```

```
// SPI.setClockDivider(SPI_CS, 3); // 28 Mhz, ok
```

```
// SPI.setClockDivider(SPI_CS, 2); // 42 Mhz, ok
```

т.е. установлено 21 МГц попробуйте сделать повыше частоту моя плата без буферов работала на 42 или на 28 МГц

Если сетевая карта не работает (та тестовым примере адрес 0.0.0.0) Проверяем «критические» точки:

- Питание на чипе нога 47 должно быть 3.3 вольта

- Сигнал сброса RST 46 нога должен быть "1" при нажатии на кнопку сброса переходит в 0 при отпускании в 1

*RESET (Active LOW)*

*This pin is active Low input to initialize or re-initialize*

*W5200.*

*RESET should be held at least 2us after low assert, and*

*wait for at least 150ms after high de-assert in order for*

*PLL logic to be stable.*

- Сигнал перехода в режим низкого потребления PWDN 45 нога должна быть 0 всегда.

*POWER DOWN (Active HIGH)*

*This pin is used to power down pin.*

*Low : Normal Mode Enable*

*High : Power Down Mode Enable*

4.3. Если кварц не впаян то надо в файле control заменить

```
//RTC_clock rtc_clock(RC); // Внутренние часы, используется внутренний RC генератор
```

```
RTC_clock rtc_clock(XTAL); // Внутренние часы, используется часовой кварц
```

на

```
RTC_clock rtc_clock(RC); // Внутренние часы, используется внутренний RC генератор
```

```
//RTC_clock rtc_clock(XTAL); // Внутренние часы, используется часовой кварц
```

4.4. Подбор частот и режимов работы SPI.

На шине SPI «сидят» два устройства SD карта и сетевой модуль на w5200, по этому очень важно сконфигурировать шину SPI, так что бы не было конфликтов между устройствами и они стабильно работали на максимальной скорости. Все режимы и скорости задаются константами в файлах и оперативно меняться не могут (требуется перекомпиляция). В таблице ниже приведена информация о возможных настройках.

№	Имя	Файл	Описание	По умолчанию	Примечание
1	SD_SPI_CONFIGURATION	SdFatConfig.h см.	Режим SPI SD карты, (0-DMA, 1-standard SPI, 2-software, 3-custom). В режиме 1 не делается	0	Имеет смысл менять 0,1

		библиотеку sdFat	принудительная настройка SPI перед переключением на другое устройство.		возможно 3
2	SD_SPI_SPEED	Constant.h	ЭТО ДЕЛИТЕЛЬ!!! Частота SPI SD = 84/SD_SPI_SPEED т.е. 2-42МГц 3-28МГц 4-21МГц 6-14МГц. Скорость SPI SD карты.	SPI_RATE	Диапазон 2-6
3	SD_REPEAT	Constant.h	Число попыток чтения карты и открытия файлов, при неудаче переход на работу без карты	3	На скорость не влияет
4	W5200_SPI_SPEED	Constant.h	ЭТО ДЕЛИТЕЛЬ!!! Частота SPI W5200 = 84/W5200_SPI_SPEED т.е. 2-42МГц 3-28МГц 4- 21МГц 6-14МГц. В первую очередь влияет на ответ при запросах. При открытии страниц, наиболее медленная операция чтение с карты.	4	Диапазон 2-6
5	SPI_RATE	5100.h Библиотека Ethernet	См. SD_SPI_SPEED	2	Диапазон 2-6

Настройки зависят от используемой карты и шилда w5200 (шилд с буферами держит максимальную скорость spi 28 мГц (W5200\_SPI\_SPEED 3).

Максимальная возможная скорость (теоретическая!) достигается следующими настройками:

SD\_SPI\_CONFIGURATION 0

SD\_SPI\_SPEED 2

W5200\_SPI\_SPEED 2

Минимальная скорость:

SD\_SPI\_CONFIGURATION 1

SD\_SPI\_SPEED 4

W5200\_SPI\_SPEED 4

Настройка карты влияет на первоначальную загрузку страницы, при этом скорость работы сети на это влияет слабо.

Меня настройки необходимо добиться стабильной работы связки карта и w5200, проверяется многократной загрузкой наиболее тяжелой страницы - «Графики» («целевое» время загрузки 0.8-1.5 сек.). Карту возможно требуется подобрать (используем тестовый пример bench).

#### 4.5 Расшифровка кодов работы ТН

В консоль и на страницу «Система» выводится строка состояния ТН когда он включен. Там приводится код состояния ТН — что сейчас делает ТН. Код имеет три поля  
<источник><алгоритм><код\_алгоритма>

Источник: В-бойлер Н-отопление С-охлаждение

Алгоритм: h - гистерезис p - ПИД

Код алгоритма

// 1 - выключение по подаче

// 2 - включение по гистерезису

// 3 - выключение по гистерезису

// 4 - внутри гистерезиса (ПРОДОЛЖЕНИЕ! нагрев или охлаждение)

// 5 - внутри гистерезиса пауза

// 6 - сброс частоты по подаче

// 7 - сброс частоты по мощности

// 8 - сброс частоты по температуре компрессора

// 9 - сброс частоты по давлению

// 10 - разгон, пид не работает

// 11 - время пиды не пришло

// 12 - дошли до ПИДа, регулируем

// 13 - включение по обратной достигнута минимальная температура обратной

// 14 - работа супербойлера ПИД ГВС (заход в бойлер)

// 15 - Бойлер греется от предкондесатора (заход в отопление)

// 16 - сброс частоты по току инвертора

// 17 - блокировка роста частоты ПИДом при подходе к уровням защиты ПОДАЧА

// 18 - блокировка роста частоты ПИДом при подходе к уровням защиты МОЩНОСТЬ  
// 19 - блокировка роста частоты ПИДом при подходе к уровням защиты ТСОМР  
// 20 - блокировка роста частоты ПИДом при подходе к уровням защиты ДАВЛЕНИЮ  
// 21 - блокировка роста частоты ПИДом при подходе к уровням защиты ТОКУ

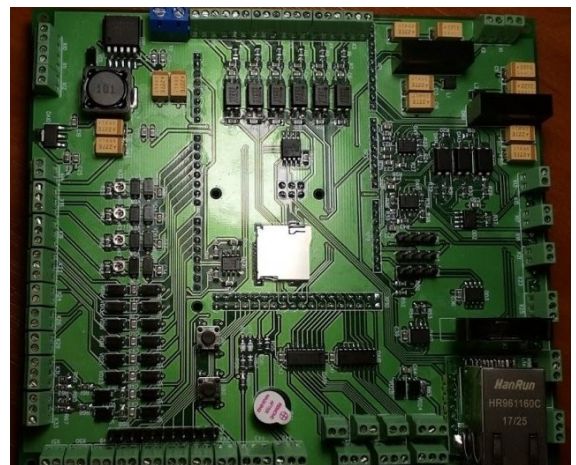
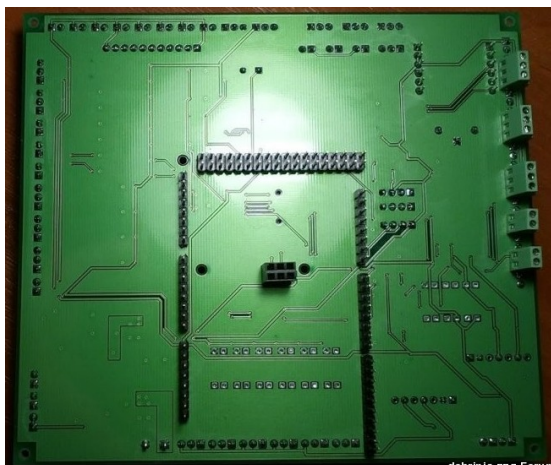
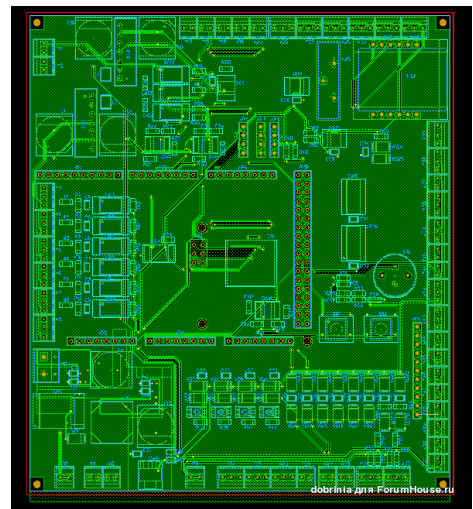
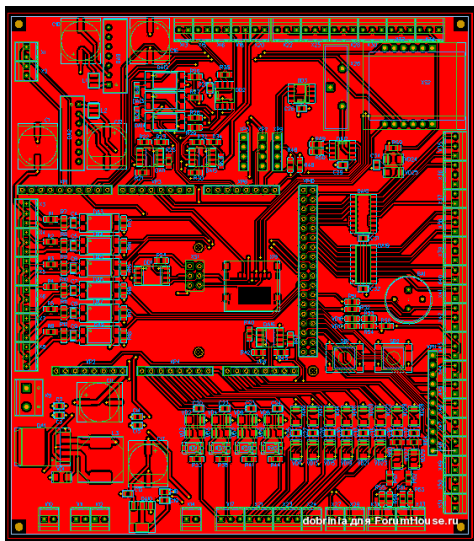
пример: Вр12 это Бойлер - Алгоритм ПИД - изменение частоты ПИДом.

## Плата разработки dobrinia

### Ключевые особенности:

- Питание внешним источником 12в. На плате выведены 12в и 5в наружу, можно запитать какие-либо датчики слаботочные.
- Датчики температуры по 1wire, 6 разъемов под винт для удобства подключения.
- Два налоговых входа для датчиков давления (испарения и конденсации)
- Датчики сухого контакта. Развязка оптронами + DC-DC по питанию. 6 штук
- Аналоговые выходы 0-10 вольт (частотник, насосы, вентиляторы и т. д.) 6 штук
- Аналоговых входы на будущее 10штук
- Выходы на два ЭРВ (под бивалент). Под винты+стандартный разъем для удобства.
- Выходы на внешний экран, и диагностические выводы.
- Разъем для карты памяти (microsd) + встроенная микросхема памяти.
- Сетевуха на w5500 модулем запаивается в плату.
- Развязанный выход 485 для управления частотником и тд.
- DUE вставляется в плату
- Выход на плату реле 8штук простых или твердотелок, универсальный разъем - можно подключить без колхоза любую из двух плат. Фото плат в приложении, цена их на али 9\$.
- плата двухслойная, размер 12.5 x 14.5см.

Наиболее дешевый вариант готовой платы. Требуется внешних силовых элементов.

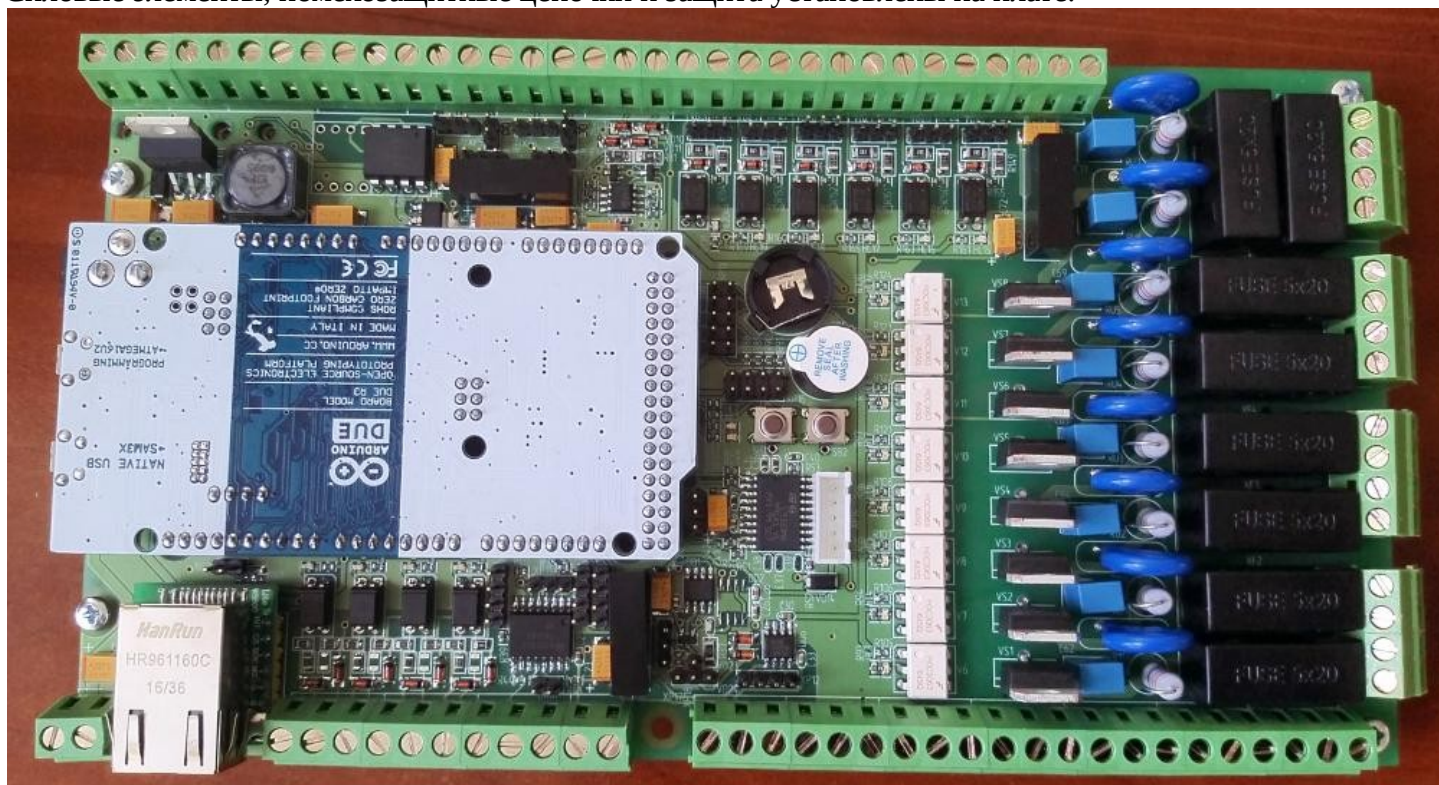




## Плата разработки pav2000

контакты Skype pav2000pav, email firstlast2007@gmail.com

Более дорогой вариант. Многие узлы сделаны по решениям из промышленного оборудования. Силовые элементы, помехозащитные цепочки и защита установлены на плате.



### Описание управления

1. На плате есть кнопка сброса (ближняя к due), работает как штатная кнопка сброса.
2. Дальняя кнопка (KEY1). Если нажата при сбросе, то включается safeNetwork режим, при котором:

- сетевые настройки берутся по умолчанию.
- запрос паролей не производится вне зависимости от настроек безопасности

Это позволяет восстановить контроль над платой при его потери. После старта контроллера эта кнопка включает/выключает ТН.

3. Первый светодиод (распаян красный). После подачи питания на плату он включается, и сигнализирует о подаче питания на плату расширения. Если он начинает мигать, это признак **фатальной** ошибки FREE RTOS. Число вспышек светодиода показывает код ошибки (1:configASSERT fails, 2:malloc fails, 3:stack overflow, 4:hard fault, 5:bus fault, 6:usage fault, 7:crash data). Если включен вачдог, то производится сброс контроллера (сложно подсчитать число вспышек светодиода). Сам факт появления такой ошибки указывает на ошибки в программном коде (аналог «синего экрана смерти» для windows).

4. Второй светодиод (распаян зеленый). Медленно мигает если контроллера работает штатно, быстрое мигание показывает наличие ошибки (код ошибки можно посмотреть на веб морде). Параллельно идет звуковой сигнал об ошибке. Эта ошибка часто не фатальна и не приводит к зависанию или сбросу контроллера.

### Errata для версии платы 1.0

№	Описание проблемы.	Исправления или улучшения
1	Бипер не пищит (он без автогенерации). Стоял пьезоизлучатель HCM1206A JL World	Замена на пьезоизлучатель с автогенерацией (покупал на Али). Марки нет. Можно снять со старой материнки
2	Не работает контроль установки SD карты. Причина: корпус держателя карты не соединен с землей, а является контактом.	Припаять корпус держателя к земле. Там она рядом. <b>Требует доработки разводки.</b>

3	Просадка линии SDA шины I2C. Причина: ИС ADUM1251ARZ конфигурирует SDA как выход на внутреннюю шину. Необходимо ВНЕШНЮЮ линию SDA подтянуть к 5 вольтам. Требуется если установлена ADUM1251ARZ.	На разъеме XP12 (снизу) припаять резистор 10к между SDA и +5 вольт. <b>Требуется доработки разводки.</b>
4	 ЦАП выходной диапазон 0.5-7.8 вольт. Необходимо увеличить коэффициент усиления и добавить смещение.	R29 R30 увеличить до 36 кОм (увеличение коэффициента усиления). Припаять резисторы между инвертирующими входами (DA3 ноги 2 и 6) и +12 вольт. Номинал 240кОм. Это задает смещение на инвертирующем входе ОУ. Эти резисторы вешаются сверху (лучше использовать с проволочными выводами). Требуется переделать это узел схемы. Возможно надо использовать внешней ЦАП. <b>Требуется доработки разводки.</b>
4		Добавить на плату номер версии, и название платы «Народный контроллер ТН» <b>Требуется доработки разводки.</b>
5		Увеличить номиналы делителей для цифровых входов. Нужно что бы можно было коммутировать 24 вольта. Желательно что бы переключки кон фигурировали 24, 12, 5 вольт
6		Для DS2482 предусмотреть токо ограничивающие резисторы
7		Резистор на бипер для уменьшение уровня сигнала и ножку лучше перенести на PWN
8		Развести цепь для контроля входного питания (типа входной делитель 1/6+стабилитрон 2.56 вольта) для защиты
9		SD карта должна выступать за периметр платы, чтобы можно было менять карту не отвинчивая плату.
10		Изменить порядок разъемов I2C (обоих). Правильный порядок ног GND VCC SCL SDA (еще раз проверить!!!!!!)
11		Заменить конденсаторы на более дешевые (не тантал)
12		Желательно стабилизатор на 5 вольт разместить лежа (можно с обратной стороны), а то есть вариант отломать его
13	Подтяжка RS485	Уменьшить номинал резисторов R65 R66 до 1.5-2 ком

#### Замена компонент:

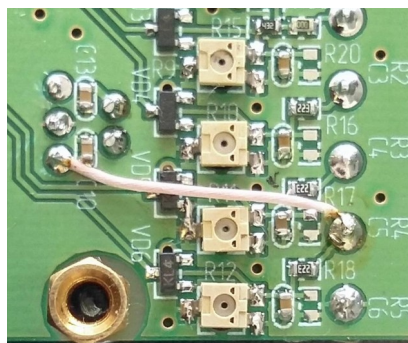
1. Замена ИС памяти AT24C64C (DD3) на AT24C512C, зачем сам не понял (но мысли уже бродят).
2. Замена варистора B72214S2251K101 250B±10% TDK на 275 вольт.
3. Светодиоды использованы разных цветов. (особенно led1 led2)
4. Замена пьезоизлучателя с автогенерацией.
5. Микросхему DD2 ставить не надо, ее поддержка в коде не планируется (уже).
6. Микросхема DD7 (L9333) возможно подойдут L9338 L9339 L93PI (уточнить)

Схема избыточна, распаиваются только требуемые элементы (узлы) которые будут использоваться в конкретном ТН. Входные цепи сделаны под различные типы датчиков, по этому номиналы резисторов могут быть не указаны и определяются самостоятельно пользователем. Возможно некоторые элементы входных цепей паивать не надо (определяется внешней периферией).

Плату Arduino DUE требуется доработать (см выше).

- установка часового кварца
- обеспечение стабильного сброса
- увеличение стабильности опоры, но здесь желательно перейти на внешнюю опору, если она распаяна на плате.

#### *Контроль входного питания 12 вольт*



**Плата может быть сконфигурирована для контроля входного питания 12 вольт (бросить перемычку). Для этого используется канал АЦП А4. Отображение производится на странице «СИСТЕМА».** **Внимание!** Перед началом контроля питания необходимо выставить резистор R12 для допустимого входного напряжения 15 вольт (иначе можно сжечь вход АЦП). Для этого необходимо на вход платы подать напряжение 2 вольта и добиться (R12) что бы на вход АЦП А4 попадало 1/6 от входного напряжения т.е. 0.333 вольта (+5%). Только после этого можно бросать перемычку между VCC и А4. R5 при таком режиме не распаивается.

Также необходимо подобрать коэффициент калибровки K\_VCC\_POWER для показа адекватных показаний.

#### *Подключение датчиков давления с выходом 0-5 вольт (до 2 шт).*

Используются каналы АЦП А9 и А8. Подключение производится на клеммы SX5 XS6 (есть вывод питания +5). При этом конфигурирование производится перемычками (на примере А0) XP1(1-2) XP3(пусто) XP5(2-3).

#### *Подключение датчиков давления с выходом 4-20 мА (до 2 шт).*

Используются каналы АЦП А9 и А8. Подключение производится на клеммы SX5 XS6 (есть вывод питания +12). При этом конфигурирование производится перемычками (на примере А0) XP1(2-3) XP3(1-2) XP5(1-2).

#### *Использование внешнего опорного напряжения*

Необходимо провести доработку Due (п.4) и поставить перемычку XP20. Это надо учитывать при калибровке каналов АЦП.

**Подключение RS485 к инвертору OmronMX2** А(нога RS485A XS8)->SN(OmronMX2) и В(нога RS485B XS8)->SP(OmronMX2), провод земли желателен для выравнивания потенциалов. Перемычка XP10 при необходимости (см описание подключенных к rs485 устройств). Скорее всего не схеме перепутаны названия А и В (согласно общеупотребимым). Для счетчика SDM120 А-В В-А согласно обозначений на схеме.

#### *Использование гальванически развязанного аналогового входа.*

Используются каналы АЦП А0 и А1 (при этом нельзя использовать каналы (А0 А1) на разъеме XS2).

#### **Монтаж частотного преобразователя omron MX2**

В процессе работы частотный преобразователь «излучает» много помех в широком диапазоне частот, по этому необходимо предпринять определенные меры по их подавлению и снижению их влиянию на контроллер. Подробнее это вопрос освящен в руководстве пользователя на преобразователь.

Помехи могут проявляться следующим образом:

- ошибки чтения датчиков температуры



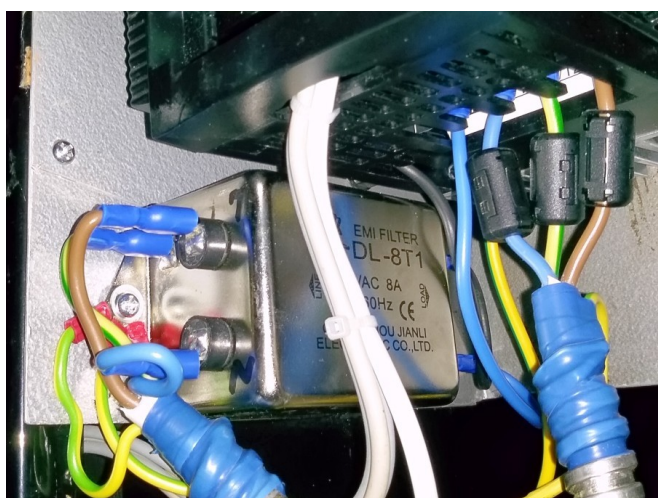
- ошибки чтения/записи в i2c eeprom
- нестабильная работа сети при включенном компрессоре
- ложные срабатывания контактных датчиков
- ошибки протокола modbus
- большая погрешность частотных датчиков

Большинство ошибок контроллер обрабатывает (либо исправление либо останов по ошибке), и сохраняет работоспособность, но для повышения надежности их надо убрать. Мои рекомендации:

1. Наличие хорошего заземления и правильное его использование при монтаже электрики теплового насоса (звезда, отсутствие контуров). Об этом можно почитать в инете.

2. Подводящий кабель к инвертору и кабель от инвертора к компрессору, должен быть экранированным и заземлен с одной стороны (со стороны инвертора) в одной точке. Я для экранировки использовал металлическую гофру, к которой паял заземляющий провод. Желательно использовать толстый провод согласно даташита на инвертор.

3. На входе инвертора по питанию, РЯДОМ с инвертором необходимо установить фильтр гашения помех. Можно использовать штатный фильтр (см руководство пользователя). Я использовал двухзвенный фильтр DL-8T1, 8A, 250V (<https://www.chipdip.ru/product/dl-8t1>). Желательно использовать фильтр рассчитанный на требуемые токи и двухзвенный. Для трехфазного используется трехфазный фильтр.



4. На выходные провода я надел ферритовый фильтр K1NF-30 (<https://www.chipdip.ru/product/k1nf-30>). В особо тяжелых случаях можно намотать по 2-3 витка выходного провода на ОДНО ферритовое кольцо проницаемостью не менее 2000 внешним диаметром 25-35 мм.

5. Необходимо сигнальные провода к инвертору использовать экранированные с точкой заземления у контроллера.

6. Силовые провода по возможности удалить от сигнальных, их пересечение желательно проводить под прямым углом.

7. Правильная настройка инвертора на компрессор. Если параметры установлены верно (параметры H100 и выше) инвертор излучает минимум помех.

### Установка датчика TEVAOUT

Правильная установка датчика температуры фреона на выходе испарителя очень важна, так как его показания участвуют при расчете перегрева и работе ПИД ЭРВ. Его надо крепить как можно ближе к выходу испарителя, желательно на горизонтальном участке сверху трубы. Если в схеме ТН присутствует четырехходовой клапан, то к сожалению датчик надо крепить до его входа, и из этого вытекает необходимость еще одного датчика в режиме охлаждения. Это связано, что на четырехходовом клапане получается разность температур и в итоге не верно рассчитывается перегрев.

Кремление датчика должно обеспечивать максимальный тепловой контакт с трубой. Для уменьшения теплового сопротивления желательно использовать не гильзованные датчики ds18b20.

**Не правильно** — пятно контакта мало

**Правильно**, через специальную накладку с использованием термопасты. Датчик вставляется внутрь верхней трубки. Необходимо хорошо тепло изолировать место крепления датчика



После установки необходимо проверить адекватность работы датчика.

- При включенном ТН (после прогрева и выходе на режим) температура TEVAOUT (Температура на выходе испарителя по фреону) не МОЖЕТ быть больше чем TEVAING (Температура на входе испарителя по гликолю). Надо получить минимальное превышение (если оно отрицательное то гуд) и эту разность с минусом забить в корректировку на странице ЭРВ.

- На холодном ТН (вся масса ТН имеет одну температуру) перегрев должен быть равен 0, если введена корректировка то величине корректировки.