

Программа  
*WebSensors*  
версия 1.01

Руководство пользователя

## **Руководство пользователя**

### **1 Введение**

Настоящее руководство предназначено для ознакомления пользователя с функциональными возможностями программы WebSensors (далее – программа) .

#### **1.1 О программе**

Обозначение программы – WebSensors.

#### **1.2 Назначение программы**

Программа предназначена для чтения значений и настройки параметров Датчиков температуры и давления.

Программа является пользовательской и выполняет следующие функции:

- чтение текущих значений
- установка основных параметров;

- запись результата чтения в текстовый файл.

### 1.3 Требования к конфигурации и программному обеспечению

Минимальная конфигурация компьютера для нормального функционирования программы:

- процессор: Pentium II (Celeron) 500;
- ОЗУ: 128 МБ;
- свободное место на жестком диске: более 10 МБ;
- видеосистема с разрешением 800×600, 256 цветов;
- наличие свободного последовательного порта (RS-232);
- операционная система Microsoft Windows 98/Me/NT/XP/2000/Seven.

Для чтения результата расчетов программы из текстового файла (формат \*.wri) необходим текстовый редактор «WordPad» из стандартного комплекта операционной системы или «Microsoft Office Word».

### 1.4 Контактная информация

(C)opyright 2006 ООО «Измерительные технологии»

Адрес: 03055, г. Киев, ул. Ванды Василевской 5, офис 5.

Тел.: (044) 501-2186

E-mail: [office@measure.com.ua](mailto:office@measure.com.ua)

WWW: <http://www.measure.com.ua>

Программа, защищенная авторскими правами, распространяется бесплатно. Вы можете ее использовать и передавать третьим лицам. Вместе с тем Вы не можете программу модифицировать, адаптировать, переводить, переделывать, декомпилировать или дизассемблировать.

## 2 Описание основных характеристик и особенностей программы

### 2.1 Состав программы

В состав программы входят следующие файлы:

**WebSensors.exe** – исполняемый файл;

**WebSensors.pdf** – руководство пользователя;

**license.txt** – лицензионное соглашение в текстовом формате (кодировка Windows-1251);

В ходе работы программы создается файл настроек **websensors.ini**.

### 2.2 Установка программы

Программа не требует инсталляции. Для установки программы необходимо переписать файлы программы на жесткий диск (логический диск должен иметь минимум 10 МБ свободного места).

### 2.3 Подготовка к работе

Перед запуском программы необходимо подключить датчик температуры/давления используя переходник USB-485

### 2.4 Запуск программы

Запустить исполняемый файл. После запуска программы на компьютере появляется меню, приведенное на рис. 1. Необходимо задать номер виртуального COM-порта компьютера, к которому подключен датчик, выбрать параметры соединения по умолчанию 9600, контроль потока fcNone, парити

paNone, тип протокола sensors OpenBus или ModBus и нажать мнемокнопку «**Обновить**»

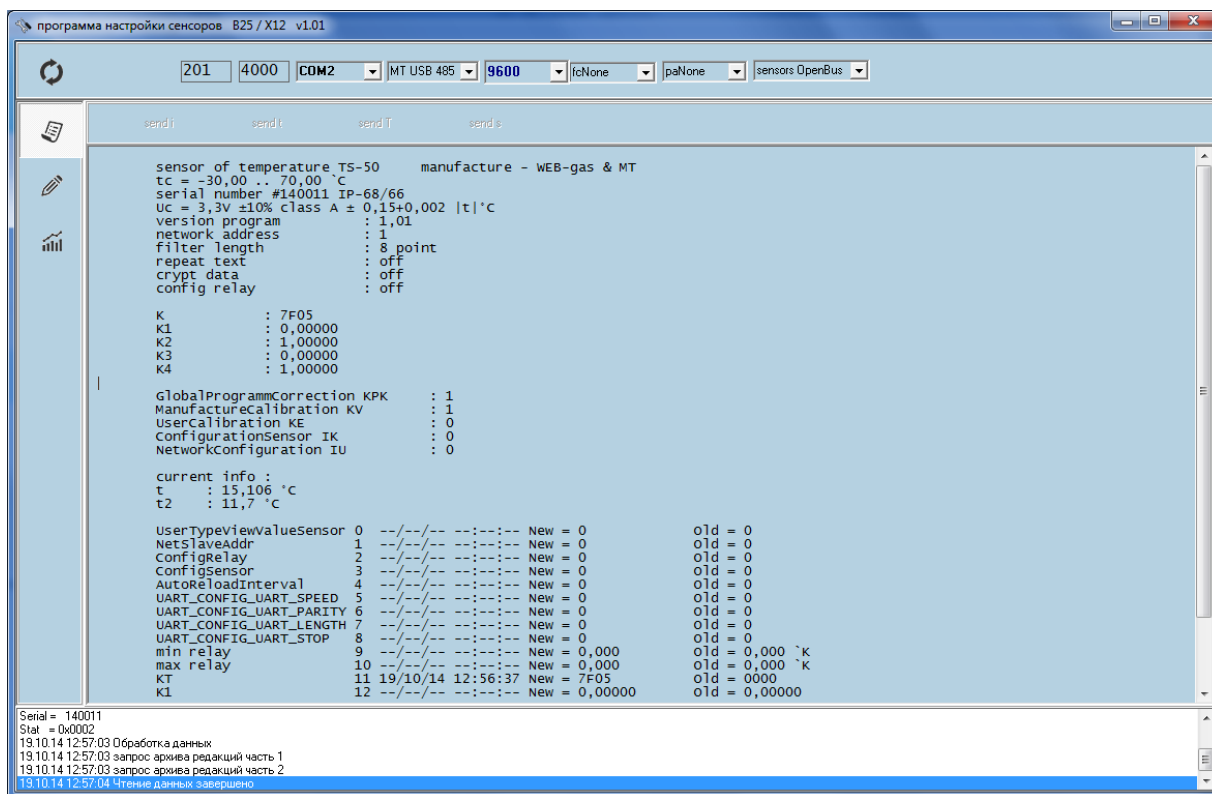


Рисунок 1 Пример запуска программы

В режиме ModBus доступны запросы к текстовому описанию датчика - кнопки t, T и i. Соответственно получить текущее рабочее значение, текущее рабочее значение и температуру окружающей среды и общее описание подключенного датчика, данные запросы запрещено использовать в режиме подключения сеть, только при единичном подключении. Кнопка s - используется для остановки вывода текстовой информации в режиме повтора вывода.

При выполнении чтения в окно вывода информации будет выведена информация о подключенном датчике, его идентификатор и статус выполнения операции чтения.

**Пример:** результат чтения данных:

```
sensor of temperature TS-50 manufacture - WEB-gas & MT
tc = -30,00 .. 70,00 °C
serial number #140011 IP-68/66
Uc = 3,3V ±10% class A ± 0,15+0,002 |t|°C
version program : 1,01
network address : 1
filter length : 8 point
repeat text : off
crypt data : off
```

config relay : off

K : 7F05  
K1 : 0,00000  
K2 : 1,00000  
K3 : 0,00000  
K4 : 1,00000

GlobalProgrammCorrection KPK : 1  
ManufactureCalibration KV : 1  
UserCalibration KE : 0  
ConfigurationSensor IK : 0  
NetworkConfiguration IU : 0

current info :  
t : 15,106 °C  
t2 : 11,7 °C

Old = 0	UserTypeViewValueSensor	0	--/--/--	--:--:--	New = 0
Old = 0	NetSlaveAddr	1	--/--/--	--:--:--	New = 0
Old = 0	ConfigRelay	2	--/--/--	--:--:--	New = 0
Old = 0	ConfigSensor	3	--/--/--	--:--:--	New = 0
Old = 0	AutoReloadInterval	4	--/--/--	--:--:--	New = 0
Old = 0	UART_CONFIG_UART_SPEED	5	--/--/--	--:--:--	New = 0
Old = 0	UART_CONFIG_UART_PARITY	6	--/--/--	--:--:--	New = 0
Old = 0	UART_CONFIG_UART_LENGTH	7	--/--/--	--:--:--	New = 0
Old = 0	UART_CONFIG_UART_STOP	8	--/--/--	--:--:--	New = 0
Old = 0	min relay	9	--/--/--	--:--:--	New = 0,000
Old = 0,000 `K	max relay	10	--/--/--	--:--:--	New = 0,000
Old = 0,000 `K	KT	11	19/10/14	12:56:37	New = 7F05
Old = 0000	K1	12	--/--/--	--:--:--	New = 0,00000
Old = 0,00000	K2	13	--/--/--	--:--:--	New = 0,00000
Old = 0,00000	K3	14	--/--/--	--:--:--	New = 0,00000
Old = 0,00000	K4	15	--/--/--	--:--:--	New = 0,00000
Old = 0	AdminPassword	16	--/--/--	--:--:--	New = 0

```
CryptPassword      17 --/--/-- --:--:-- New = 0
Old = 0
```

## 2.5 Работа с программой

Для редакции установок необходимо ввести пароль доступа, поле пароля закрыто от просмотра маской «####» по умолчанию a123 далее изменить необходимые значения параметров и нажать мнемокнопку «Записать новые настройки» - состояние записи будет отображено в поле вывода сервисной информации. При неверном пароле в доступе будет отказано.

## Редактируемые параметры:

Общая настройка:

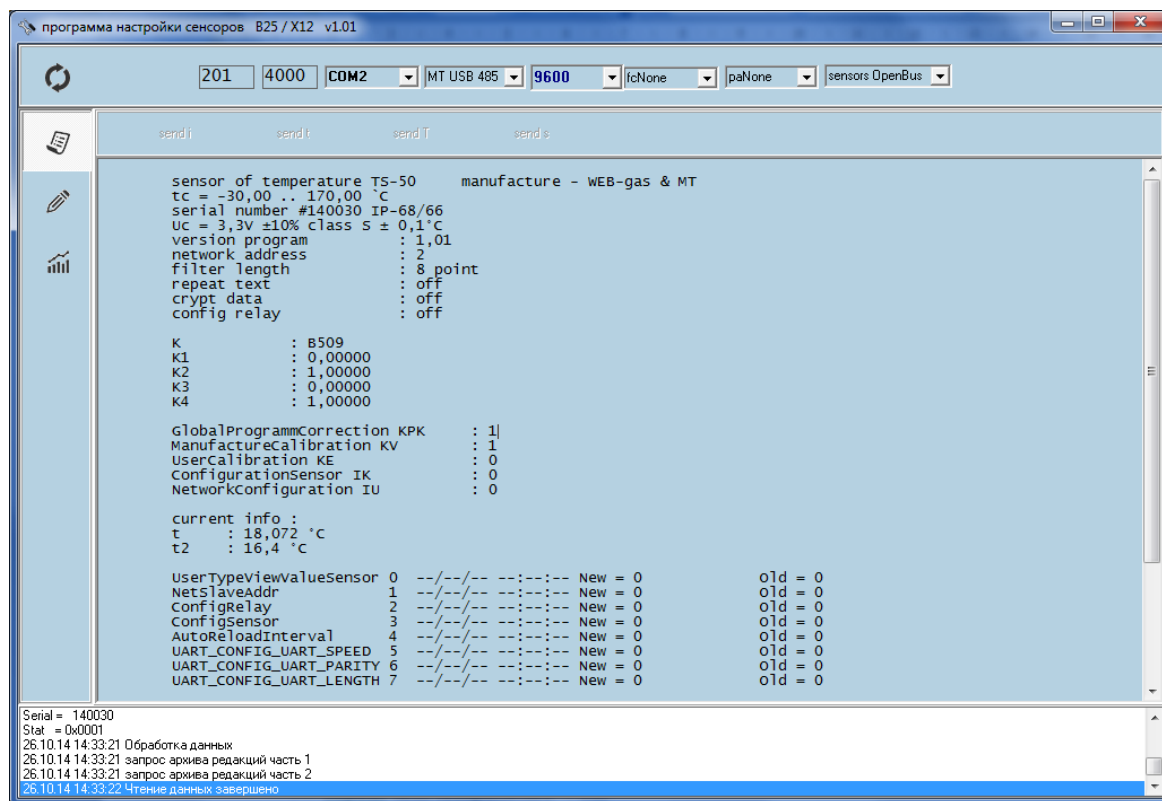


Рис 2 Настройка параметров датчика

Включить фильтр: Включает цифровой фильтр скользящее среднее  
Если включен «8 измерений» то на 8, если выключено то на 4.  
Разрешить текстовый вывод – позволяет получать данные в текстовом виде о текущих значениях

Запрос t ответ 18.145 (31 38 2E 31 34 35 0D 0A )

Запрос T ответ 18.162 16.513 ( 31 38 2E 31 36 32 09 31 36 2E 35 31 33 0D 0A )

где 18.162 рабочая температура, 16.513 температура окружающей среды, данные разделены табуляцией '\t' ( 0x09 ) признак конца строки «\r\n» ( 0D 0A )

Запрос i

```
>> sensor of temperature TS-50 класс S ± 0,1°C  
>> manufacture - WEB-gas & MT  
>> tc = -30..170°C PN-20 atm K = B509  
>> №140030 IP-68/66 ta= -30..+60°C  
>> version program : 1.01  
>> network address : 2  
>> filter : 8 point  
>> repeat text temperature : off  
>> crypt data temperature : off  
>> config relay : off
```

```

>> K1          : 0.00000
>> K2          : 1.00000
>> K3          : 0.00000
>> K4          : 1.00000
>> GlobalProgrammCorrection KPK      :1
>> ManufactureCalibration KV         :1
>> UserCalibration KE                 :0
>> ConfigurationSensor IK            :0
>> NetworkConfiguration IU           :0
>> type view      : Celsius
>> current info :
>> t              : 17.971
>> t2             : 16.361

```

При необходимости последовательного вывода – разрешить «повторять текст» – данные для текущих значений будут повторно выводиться в интерфейс с интервалом в 1 секунду.

Индикация температуры выбор: °K, °C, °F

Индикация давления выбор: МПа или кПа

Новый сетевой адрес – в сети датчиков не должно быть одинаковых сетевых адресов

Режим реле:

Включается если установлен режим. В данном режиме интерфейс работает как коммутатор реле и на запросы не отвечает. При включении питания пауза включения режима 5 сек, при корректном запросе режим реле отключается до следующей перезагрузки по питанию. Данная пауза позволяет выполнить проверку и настройку датчика.

*Цикл измерения* – периодичность опроса датчиков, в мс, если ноль, цикл непрерывный.

Лимиты рабочих диапазонов:

*Tmin* – минимальная температура

*Tmax* – максимальная температура

*Pmin* – минимальное давление

*Pmax* – максимальное давление

*Тип контакта* – нормально замкнут или нормально разомкнут Характеризует срабатывание механизма по фронту или по спаду.

Безопасность:

*Новый пароль* требует подтверждения – пароль для коррекции данных.

Внимание пароль не забывать.

*Код шифрования* требует подтверждения, пароль шифрует данные текущих значений, пароль сбросить нельзя. Только отключить режим шифрования. Повторное включение – ввод нового пароля.

*Параметры порта* :

*Скорость* :

110, 150, 300, 600, 1200, 2400, 4800, **9600**, 14400, 19200, 38400, 57600, 115200



Парити: **paNone**, paMark, paEven, paOdd, paSpace

Данные: 7 бит, **8 бит**

Стоповый бит: **1 стоповый**, 1.5 стоповых, 2 стоповых

Сетевой адрес: 1-200, 201 используется как широковещательный, и нельзя использовать в сети датчиков

Для идентификации датчика рекомендуется записать текстовое описание датчика

Установка коэффициентов датчиков выполняется на предприятии изготовителе.

Запись коэффициентов K1, K2 выполняется при необходимости в линейной коррекции во время межповерочного интервала.

Датчики имеют 2 типа протоколов верхнего уровня – MobBus и Open-SensorBus

Данные имеют одинаковую структуру и отличаются типом заголовка  
Формат запроса MobBus

В протоколе датчики поддерживают только чтение данных из регистров командой 0x03. Регистры 5, 38, 1000-1009 используются только в датчиках температуры. Измерительные функции датчиков температуры реализуются только при чтении регистров с номерами 0 (серийный номер датчика), 3 (температура, К/ давление, МПа), 1000 (температура, °С), 1004 (температура, К), 1006 (температура, °С), 1008 (температура, F).

Регистры 1004-1009 имеют формат информации с 1 знаком после запятой и предназначены для ускоренных режимов обработки без предварительного формирования данных из отдельных регистров.

Регистры датчиков

номер	тип, имя данных	комментарий
0	unsigned long	серийный номер датчика
1	NumberSensor	
2	unsigned short StatusSensor	состояние датчика: бит 0 – выход за верхнюю границу диапазона измерения бит 1 – выход за нижнюю границу диапазона измерения бит 2 – обрыв датчика бит 3 – ошибка АЦП бит 4 – временная блокировка коррекции бит 5 – ошибка данных флэш, датчик подлежит ремонту
3	unsigned long	рабочая температура, К, 3 знака после зпт. или давление, МПа, 6 знаков после зпт.
4	Work Param	
5	unsigned long	окружающая температура, К, 3 знака после зпт
6	Kelvin T2	
7	unsigned long	время изготовления (сек) плюс 01/01/2000 00:00
8	ID	
9	char [8]	текстовая идентификация датчика, например TS-50
10	Type string	

11		
12		
13	unsigned short Version	версия программного обеспечения датчика
14	unsigned short TypeDev	версия конструкции датчика
15	unsigned short TypeSensor	исполнение датчика: 0 – А; 1 – S; 2 – Р или 0 – U ; 1 – S ; 2 – РТ ; 3 – РТ0; 4 – РТ1; 5 – РТ2
16	unsigned short UserTypeView ValueSensor	установка пользователя при выводе температуры на компьютер в текстовом формате в системах мониторинга 0 – К, 1 – °С, 2 – F или 0 – МПа, 1 – кПа, 2 – кГ/см <sup>2</sup> , 3 – psi
17	unsigned short ConfigSensor	значения управляющих битов: 0 – выкл., 1 – вкл. бит 0 – цифровая фильтрация бит 1: 0 – фильтр длины 4 отсчета, 1 – длины 8 отсчетов бит 2 – текстовый вывод бит 3 – автоповтор текста бит 4 – шифрование информации
18	unsigned short NetSlaveAddr	сетевой адрес датчика, от 1 до 200
19	unsigned long UART_CONFIG	байты (0.. 3): UART_SPEED, UART_PARITY, UART_LENGTH, UART_STOP настройки параметров порта: UART_SPEED (0.. 13) – 110, 150, 300, 600 бод, 1,2, 2,4, 4,8, 9,6, 14, 19, 38, 57, 115, 230 кбод UART_PARITY (0.. 3) – no, mark, even, odd UART_LENGTH (0.. 1) – bit7, bit8 UART_STOP (0.. 2) – 1, 1,5, 2 по умолчанию 9,6 kbps, no parity, 1 stop
20		
21	unsigned short ConfigRelay	бит 0 – режим реле бит 1: 1 – нормально замкнуто, 0 – разомкнуто бит 2: 0 – срабатывание по принижению порога, 1 – по превышению
22	unsigned short AutoReload Interval	пауза цикла автообновления в режиме реле, мс
23	unsigned long Lmin	нижний порог срабатывания реле, К, или МПа, 2 знака после зпт.
24		
25	unsigned Long Lmax	верхний порог срабатывания реле, К, или МПа, 2 знака после зпт.
26		
27	unsigned long Lmin_Sensor	нижняя граница диапазона измерения, устанавливается при выпуске, К, или МПа, 2 знака после зпт.
28		
29	unsigned long Lmax_Sensor	верхняя граница диапазона измерения, устанавливается при выпуске, К, или МПа, 2 знака после зпт.
30		
31	unsigned short K	коэффициент преобразования датчика, калибровка при выпуске
32	float K1 или A	калибровка при эксплуатации: сенсор основной температуры или датчик давления (при выпуске значение 0)
33		

34	float K2 или B	калибровка при эксплуатации: сенсор основной температуры или датчик давления (при выпуске значение 1)
35		
36	float K3 или C	калибровка при эксплуатации: сенсор окр. температуры или датчик давления
37		
38	float K4	коэффициент калибровки, сенсор температуры окр. среды, в датчиках температуры
39		
40	unsigned short Type Environment	Тип среды 0 - жидкость, 1 - газ, 2 - пар, 3 - поверхность
41	float Speed_Type Environment	Скорость потока среды
42		
43	char [40] String	пользовательское описание датчика
...		
63		
64	unsigned long	GlobalProgrammCorrection KPK
65		
66	unsigned long	ManufactureCalibration KV
67		
68	unsigned long	UserCalibration KE
69		
70	unsigned long	ConfigurationSensor IK
71		
72	unsigned long	NetworkConfiguration IU
73		
1000	float dataT1C	рабочая температура, °C
1001		
1002	float dataT2CF	окружающая температура, °C
1003		
1004	unsigned short dataT1K	рабочая температура, K, 1 знак после зпт.
1005	unsigned short dataT2K	окружающая температура, K, 1 знак после зпт.
1006	short dataT1C	рабочая температура, °C, 1 знак после зпт.
1007	short dataT2C	окружающая температура, °C, 1 знак после зпт.
1008	short dataT1F	рабочая температура, F, 1 знак после зпт.
1009	short dataT2F	окружающая температура, F, 1 знак после зпт.

#### Формат запроса OpenSensorBus

```

unsigned char SlaveNetAddress; // сетевой адрес датчика
unsigned char len = 4; // запрос всегда равен 4 байтам данных
unsigned char reserv; // не используется
unsigned char TypeCode; // код запрашиваемых данных
unsigned short idp; // идентификатор запроса
unsigned char MasterNetAddress; // сетевой адрес мастера
unsigned short CRC // арифметическая сумма байт запроса

```

где TypeCode:

- 0 - IDDevice
- 1 - CurrentINFO
- 2 - CurrentDATA\_T1
- 3 - CurrentDATA\_T12
- 10 - запрос архива редакций часть 1
- 11 - запрос архива редакций часть 2

Формат ответа      OpenSensorBus

```
typedef struct {
unsigned long   NumberSensor;           // номер датчика
unsigned short  StatusSensor;           // состояние
long   sl_Kelvin_data[2]; // измененное значение
unsigned long   ID;                     // ID
char   Type[8];                          // тип датчика

unsigned short  Version;                 // версия
unsigned short  TypeDev;                 // хардварный тип датчика
unsigned short  TypeSensor;              //

unsigned short  UserTypeViewValueSensor; // тип вывода
union {
TD_ConfigSensor BitConfigSensor;
unsigned short  ConfigSensor; // настройка датчика
};
unsigned short  NetSlaveAddr; // сетевой адрес датчика
union {
TD_UART_CONFIG  UART_CONFIG;
unsigned long    ulUART_CONFIG;
};

unsigned short  ConfigRelay; // настройка реле
unsigned short  AutoReloadInterval; // время автообновления
long   min; // пользовательский минимум
long   max; // пользовательский максимум
long   min_Sensor; // заводской минимум
long   max_Sensor; // заводской максимум
short  KT1; // индикация заводской калибровки

float K1; // коэффициент пользовательской калибровки рабочего значения
float K2; // коэффициент пользовательской калибровки рабочего значения

float K3; // коэффициент калибровки термометра окр. среды
float K4; // коэффициент калибровки термометра окр. среды

unsigned short  Type_Environment; // тип среды
float Speed_Type_Environment; // скорость потока среды использования
unsigned short  CodeTypeHardwareSensor; // тип датчика

char   String[40]; // текстовое описание датчика
```

```

unsigned long GlobalCountCorrection;      //
unsigned long SystemCalibrationCorrection; //
unsigned long UserCalibrationCorrection;  //
unsigned long CountConfigurationSensor;   //
unsigned long CountConfigurationNetwork;  //
} TD_CurrentINFO;

```

Дополнительное описание используемых переменных в структуре данных

```

typedef enum
{
    boud_110,
    boud_150,
    boud_300,
    boud_600,
    boud_1200,
    boud_2400,
    boud_4800,
    boud_9600,
    boud_14400,
    boud_19200,
    boud_38400,
    boud_57600,
    boud_115200,
    boud_err
} TD_UART_SPEED;

```

```

typedef enum
{
    paNone,
    paMark,
    paEven,
    paOdd,
    paSpace
} TD_UART_PARITY;

```

```

typedef enum
{
    lbit7,
    lbit8
} TD_UART_LENGTH;

```

```

typedef enum TUART_STOP
{
    stop1,
    stop15,
    stop2
} TD_UART_STOP;

```

```

typedef struct
{
    TD_UART_SPEED    UART_SPEED;

```

```

        TD_UART_PARITY UART_PARITY;
        TD_UART_LENGTH UART_LENGTH;
        TD_UART_STOP    UART_STOP;
    } TD_UART_CONFIG;

typedef struct
{
    unsigned char  Enable_Filter      :1;  // flag enable filter
    unsigned char  Length_Filter      :1;  // value Filter 4 / 8 point
    unsigned char  Enable_TextOut     :1;  // flag enable text Out
    unsigned char  Repeated_TextOut   :1;  // flag repeat text Out
    unsigned char  Crypt_Temperature :1;  // flag crypt data
} TD_ConfigSensor;

typedef struct {
    union {
        float Data;          // Данные о рабочем значении
        char  CryptData [4]; // либо зашифрованный данные
    };
    unsigned long  NumberSensor; // номер датчика
} TD_CurrentData_T1;

typedef struct { // расширенные значения
    union {
        float Temperature; // Данные о рабочем значении
        char easyCrypt[4];  // либо зашифрованный данные
    };
    union {
        float Temperature_environment; // температура окр. среды
        char easyCrypt2[4];             // температура зашифрованная
    };
    unsigned long  NumberSensor; // номер датчика
} TD_CurrentData_T12;

```

Архиве редакций хранит последнюю коррекцию переменной

```

typedef enum
{
    [ 0] code_UserTypeViewValueSensor,
    [ 1] code_NetSlaveAddr,
    [ 2] code_ConfigRelay,
    [ 3] code_ConfigSensor,
    [ 4] code_AutoReloadInterval,
    [ 5] code_UART_CONFIG_UART_SPEED,
    [ 6] code_UART_CONFIG_UART_PARITY,
    [ 7] code_UART_CONFIG_UART_LENGTH,
    [ 8] code_UART_CONFIG_UART_STOP,
    [ 9] code_Tmin,
    [10] code_Tmax,
    [11] code_KT,
    [12] code_K1,
    [13] code_K2,

```

```
[14] code_K3,  
[15] code_K4,  
[16] code_AdminPassword,  
[17] code_CryptPassword,  
[18] code_UserText,  
[19] code_SystemPassword,  
[20] code_FlashLogAccess,  
[21] code_Type_Environment,  
[22] code_Speed_Type_Environment  
} TECODE_EDIT;
```

Имеет вид

```
typedef struct {  
    unsigned long date; // дата корекции время компьютера инд. параметр  
    unsigned long new_data; // новое значение из формата описания  
    unsigned long old_data; // старое значение  
} TD_bus_pointLog;
```

Архив имеет 2 части по 16 значений. Ячейки архива с 23 по 31 резерв под дополнительную конфигурацию.





UserCalibrationCorrection	= 0	0x00000000
CountConfigurationSensor	= 1	0x00000001
CountConfigurationNetwork	= 0	0x00000000