

## **Система управления отоплением на базе роутера Keenetic**

Система предназначена для включения/выключения радиаторов отопления по расписанию и показаниям датчиков температуры, работающих по радиоканалу. Также система записывает в базу данных показания температуры, давления и влажности. Управление системой и просмотр журнала показаний осуществляется через веб-интерфейс. Расписание может автоматически переключаться по признаку выходные/будни.

Система представляет собой OpenWRT-пакет, и должна работать также на других роутерах с поддержкой Entware.

Для управления радиаторами используются приводы от “теплого пола”, которые устанавливаются взамен терморегуляторов. Используются нормально закрытые приводы с управляющим напряжением 24 В.

В качестве датчиков используются внешние датчики метеостанций BALDR, работающие на частоте 433 МГц. Датчики передают значения температуры и влажности.

Прием сигнала от датчиков и выдача напряжения на приводы осуществляется устройством на базе платы Arduino Nano.

К плате Arduino подключены стандартные модули приемника 433 МГц, датчика температуры и давления BMP280, и управления нагрузкой.

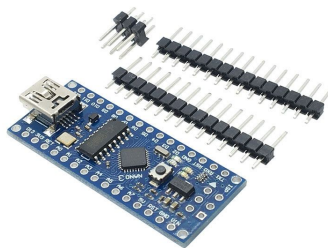
Плата Arduino по USB подключена к роутеру. Программа на роутере запускается по планировщику (cron) каждые 10 минут, считывает по протоколу Modbus значения температуры/давления/влажности и передает их в базу данных. Далее в БД формируется нужное состояние обогрева для комнаты, где расположен датчик. Программа на роутере вычитывает из базы состояние приводов радиаторов, и по Modbus включает/выключает выходы на плате Arduino.

Настройки системы, профили управления отоплением в комнатах хранятся в базе данных MariaDB. Запись журнала показаний в базу - опциональна, при ее отключении база данных будет работать в режиме Read-only, что может быть полезно в целях сохранения ресурса флеш-памяти и места на накопителе.

### **Аппаратная часть**

Прием данных от датчиков температуры в комнатах и управление приводами радиаторов отопления осуществляет плата [Arduino Nano](#), с подключенными дополнительными модулями.

На плате распаян микроконтроллер AVR ATmega328p и преобразователь USB-Serial CH340



Роутер Keenetic распознаёт подключенную к нему по USB плату как модем.

Для измерения температуры в комнатах используются [датчики для метеостанции BALDR](#).



Сигналы от датчиков метеостанций передаются по радиоканалу с использованием широтно-импульсной модуляции. Начало посылки состоит из 4 импульсов по 700 мкс каждый, с такой же по длительности паузой между ними. После этого передаются 40 бит данных. Передача каждого бита занимает 700 мкс, значение бита определяется длительностью импульса. Датчик передает пакеты из нескольких одинаковых посылок данных каждую минуту.

Формат посылки датчика:

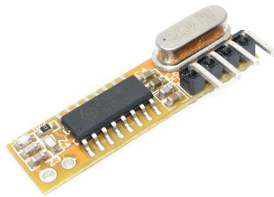
ID	BATT	BUTTON	CHANNEL	TEMPERATURE	HUMIDITY	CRC
8 bit	1 bit	1 bit	2 bit	12 bit	8 bit	8 bit

Температура передается в десятых долях Фаренгейта. Для пересчета в градусы Цельсия используется формула  $T = (Fahrenheit - 32) * 5/9 - 500$ ;

CRC представляет собой младший разряд суммы предыдущих байтов.

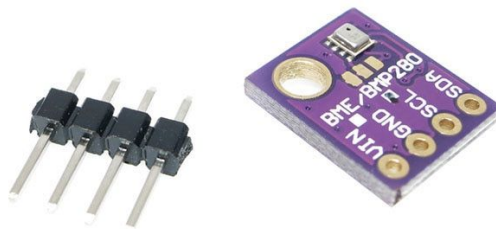
Сигналы от датчиков из разных комнат различаются по номеру канала, который устанавливается переключателем на датчике. Таким образом, в системе возможно до 3 внешних датчиков.

Прием сигнала от датчиков осуществляет стандартный [модуль приемника 433 МГц](#):



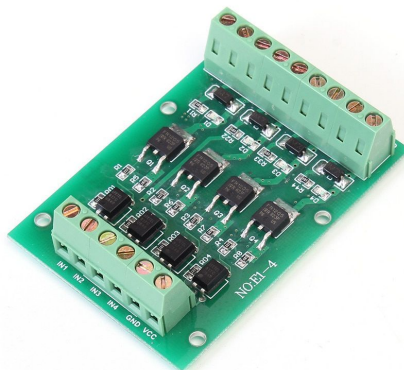
Модуль приемника регистрирует наличие сигнала несущей и формирует импульсы на входе внешнего прерывания AVR. Программная часть проверяет длительность и правильность последовательности поступающих импульсов, осуществляет проверку контрольной суммы и декодирование показаний датчиков.

Чтобы расширить функционал системы и смотреть также атмосферное давление, в состав был также добавлен [модуль датчика атмосферного давления BMP280](#).



Модуль подключается стандартно по интерфейсу I2C. Программа в AVR настраивает модуль на периодическое измерение температуры и атмосферного давления, далее по таймеру считывает показаний.

Управление приводами радиаторов осуществляется через [модуль управления нагрузкой](#):



Для управления нагрузкой используются выходы порта С микроконтроллера. К выходам PORTC0...PORTC3 подключен модуль управления нагрузкой с опторазвязанными полевыми транзисторами. Опторазвязка в модуле - желательна, так как при этом питание приводов радиаторов полностью независимо от питания роутера. Плата Arduino питается от USB-порта роутера, а приводы радиаторов имеют независимое питание от внешнего источника.

Для расчета мощности внешнего источника питания приводов нужно учитывать количество приводов, а также тот факт, что термоэлектрический привод при включении потребляет ток в несколько раз больше номинала. Так, привод номинальной мощностью 3 Вт с номинальным током потребления 120 мА несколько минут после включения потребляет ток 700 мА, затем значение тока приходит к номиналу. Соответственно, для трех приводов по 3 Вт нужная мощность БП должна быть не менее 50 Вт (сейчас используется БП LRS-50-24, 24В, 2.2А, 52Вт).

### Обмен данными с роутером

Передача показаний датчиков в роутер осуществляется по протоколу MODBUS. Роутер является Modbus-мастером и периодически считывает показания датчиков как значения регистров Modbus. Значение записывается в десятых долях градуса Цельсия.

Если данные не приходили более 5 минут, то в регистры будет записано значение температуры -273 С, что является признаком отсутствия достоверных показаний.

Таблица Input-регистров

0	Температура по датчику BMP280
1	Давление по датчику BMP280
2	Температура по датчику канала 1
3	Влажность по датчику канала 1
4	Температура по датчику канала 2
5	Влажность по датчику канала 2
6	Температура по датчику канала 3
7	Влажность по датчику канала 3

Таблица Discrete Inputs

0	Состояние батарейки датчика канала 1
1	Состояние батарейки датчика канала 2
2	Состояние батарейки датчика канала 3

Управление подключенными приводами радиаторов осуществляется через запись значений Modbus Coils

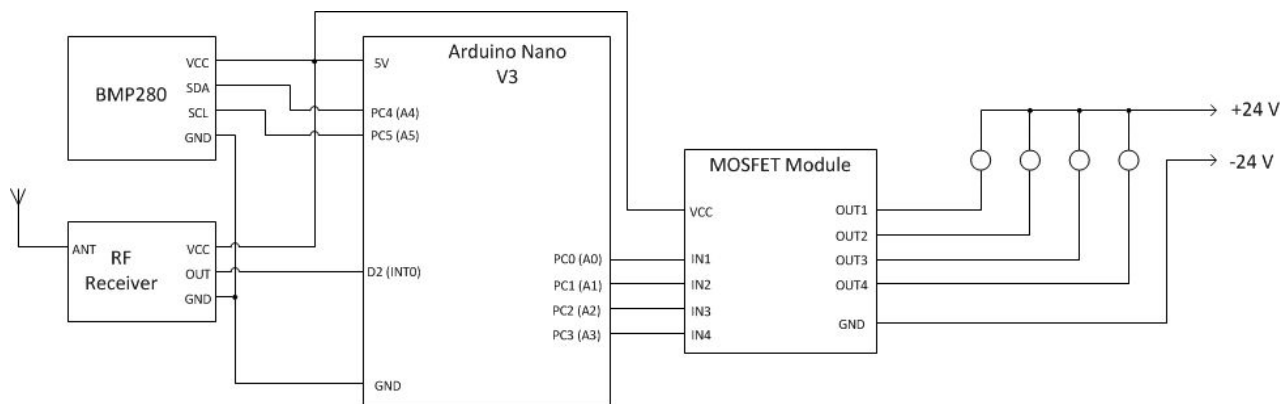
Таблица Output Coils

0	Привод канала 1 (PORTC0)
1	Привод канала 2 (PORTC1)
2	Привод канала 3 (PORTC2)
3	Привод канала 4 (PORTC3)

Оба подключенных модуля - приемник и датчик BMP280 опциональны. Если программа микроконтроллера не обнаружит при старте модуля BMP280, то периодическая вычитка из него не будет производиться, и в соответствующих регистрах Modbus всегда будут некорректные значения.

### Сборка устройства

Принципиальная схема:



На плате Arduino Nano надо обязательно удалить конденсатор, который соединяет вывод 14 CH340 (DTR) с выводом RST микроконтроллера. Если этого не сделать, микроконтроллер будет сбрасываться при каждом открытии порта со стороны роутера.

В микроконтроллер прошивается файл modbus.hex. FUSE-биты микроконтроллера устанавливаются для работы на частоте 16 МГц. Для прошивки hex-файла и установки FUSE-битов нужен программатор.

После сборки и прошивки плату Arduino Nano надо подключить к ПК, и проверить что значения датчиков записываются в соответствующие регистры MODBUS. Проверить можно подключившись MODBUS-клиентом к USB-порту Arduino на скорости 38400. В качестве программы-клиента можно использовать [qModMaster](#). При старте в регистрах температуры будут значения -2730, которые через несколько минут сменятся на показания датчиков.

## Программная часть

Программа для роутера представляет собой OpenWRT-пакет **modsensor.ipk**, который устанавливается в систему Entware. Для работы необходима установленная на роутере СУБД MariaDB и библиотека libmodbus. Также в системе Entware должен быть установлен планировщик cron, и настроен периодический запуск программы **modsensor** (каждые 10 минут).

Для возможности обмена данными с подключенной платой Arduino в роутере должны быть соответствующие драйвера переходников USB-Serial. В роутерах Keenetic это компонент “USB-модемы”

Все настройки системы хранятся в таблице БД. При включении программа считывает имя подключенного по USB устройства (обычно это “/dev/ttyUSB0”) и скорость порта (38400), и пытается подключиться к плате Arduino. Далее открывается таблица в БД, где хранятся номера используемых регистров MODBUS. Затем программа последовательно считывает из Arduino требуемые регистры, и вызывает процедуру БД, передавая в качестве параметра их значения. Для хранения текущих значений регистров в БД используется временная таблица в памяти, которая создается при первом обращении. После того, как таблица будет заполнена текущими значениями, вызывается процедура обновления текущего состояния. При обновлении текущего состояния данные датчиков записываются в БД (если это включено в настройках), а также формируется временная таблица в памяти с нужными значениями дискретных выходов. Программа считывает эти значения из таблицы БД и по Modbus включает/выключает приводы радиаторов.

## База данных

Основная логика работы программы содержится в сохраненных процедурах БД. Используемые номера регистров хранятся в таблице БД с привязкой к датчикам. Температура в регистрах Modbus хранится в десятых долях градуса, при записи в БД значение преобразуется в тип FLOAT. В памяти создаются временные таблицы с текущими значениями показаний датчиков, при включенном в настройках соответствующем параметре также происходит запись в таблицы журнала значений показаний.

В момент вызова процедуры обновления состояния происходит выбор текущего профиля на основе настроек и дня недели. Профили “Будни” и “Выходные” - встроенные, выбираются автоматически при включенном в настройках автовыборе профиля.

В таблице параметров каждого профиля для каждого времени суток задается желаемая температура в комнате, на основе этого рассчитывается, нужно ли включать отопление в комнате. Далее значения состояния обогрева для комнат с датчиками используются для управления приводами радиаторов. Для каждого привода радиатора радиатора можно указать основной датчик и резервный - он будет использоваться, когда нет актуальных показаний от основного датчика (например, разрядилась батарейка).

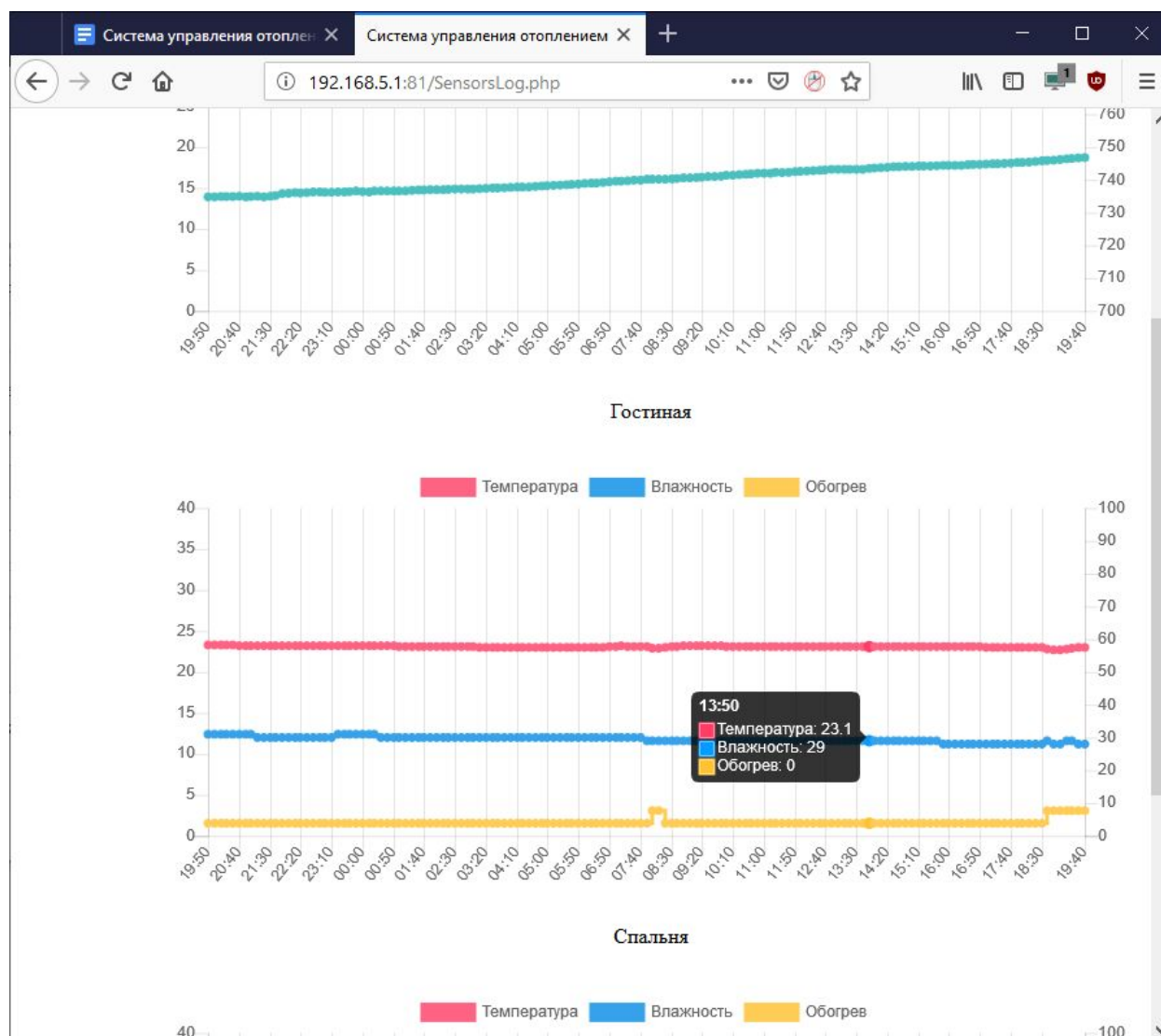
## WEB-интерфейс

Для управления системой, просмотра текущих значений и журнала показаний датчиков служит WEB-интерфейс, реализованный как PHP-приложение. Приложение

отображает данные из базы, реагирует на изменение параметров пользователем и обновляет данные в БД.

Для работы WEB-интерфейса на роутере устанавливается WEB-сервер nginx, и модуль для работы с PHP.

Данные на страницах отображаются с помощью JS-библиотек Chart.js и Tabulator, для графиков и таблиц соответственно.



Состояние

Настройки

Журнал

Профили

Текущее состояние системы

Внутренний датчик

Температура	Давление	Обогрев	Данные обновлены
31.3	746.9	Не задано	2019-04-01 19:50:01

Внешние датчики

Датчик	Температура	Влажность	Батарейка	Обогрев	Данные обновлены
Гостиная	23.1	28	Норма	Выключен	2019-04-01 19:50:01
Спальня	23.2	29	Норма	Выключен	2019-04-01 19:50:01

Автовывбор профиля (Будни/Выходные) включен, текущий профиль: "Будни"

## Установка

В систему Entware нужно установить пакеты MariaDB, nginx из репозитория:

```
~ # opkg update
~ # opkg install nginx php7-fastcgi php7-mod-mbstring
mariadb-server mariadb-client-extra
```

Дальше в каталог **/opt/home** скопировать пакет **modsensor.ipk** и установить:

```
~ # opkg install /opt/home/modsensor.ipk
```

Создать базу данных:

```
~ # mysql_install_db
```

Запустить mariadb-server:

```
~ # /opt/etc/init.d/S70mariadbd start
```



Задать пароль доступа:

```
~ # mysqladmin -u root password 'yourpassword'
```

Войти и создать БД

```
~ # mysql -u root -p
create database modsensor;
GRANT ALL PRIVILEGES ON *.* TO 'modsensor'@'localhost' IDENTIFIED
BY 'modsensor';
FLUSH PRIVILEGES;
\q
```

Перезапустить mariadb-server:

```
~ # /opt/etc/init.d/S70mariadb restart
```

Наполнить базу данных, путем импорта содержимого SQL-файлов. Для этого скопировать каталог SQL в /opt/home на роутере и выполнить:

```
~ # cd /opt/home/sql/modsensor
~ # cat *.* | mysql -u root modsensor
```

Создать скрипт запуска **S79php-fcgi**:

```
#!/bin/sh
export PHP_FCGI_CHILDREN=''
ENABLED=yes
PROCS=php-fcgi
ARGS="-b /opt/var/run/php-fcgi.sock &"
PREARGS=""
DESC=$PROCS
PATH=/opt/bin:/opt/sbin:/opt/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

. /opt/etc/init.d/rc.func
```

Сделать этот скрипт исполняемым:

```
~ # chmod +x /opt/etc/init.d/S79php-fcgi
```

Дальше надо настроить веб-сервер. Так как порт 80 используется веб-интерфейсом роутера, нужно настроить nginx на использование порта 81. Также зададим файл, который будет отображаться по умолчанию - текущее состояние системы. Для этого нужно отредактировать секцию **server** в файле **nginx.conf**

```
server {
    listen      81;
    server_name localhost;

    #charset koi8-r;

    #access_log /opt/var/log/nginx/host.access.log;

    location / {
        root    /opt/share/www;
        index   index.html index.htm Status.php;
    }
    location /mon/ {
        root    /opt/share/nginx;
        index   index.html index.htm;
    }
    location ~ /\.php$ {
        root    /opt/share/www;
        fastcgi_pass unix:/opt/var/run/php-fcgi.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
}
```

Содержимое файла **fastcgi\_params** должно иметь такой вид:

```
fastcgi_param  QUERY_STRING       $query_string;
fastcgi_param  REQUEST_METHOD     $request_method;
fastcgi_param  CONTENT_TYPE       $content_type;
fastcgi_param  CONTENT_LENGTH     $content_length;

fastcgi_param  SCRIPT_FILENAME    $document_root$fastcgi_script_name;
fastcgi_param  PATH_INFO          $fastcgi_path_info;
fastcgi_param  SCRIPT_NAME        $fastcgi_script_name;
fastcgi_param  REQUEST_URI        $request_uri;
fastcgi_param  DOCUMENT_URI       $document_uri;
fastcgi_param  DOCUMENT_ROOT      $document_root;
fastcgi_param  SERVER_PROTOCOL    $server_protocol;
fastcgi_param  REQUEST_SCHEME     $scheme;
fastcgi_param  HTTPS              $https if_not_empty;
```

```

fastcgi_param  GATEWAY_INTERFACE  CGI/1.1;
fastcgi_param  SERVER_SOFTWARE    nginx/$nginx_version;


fastcgi_param  REMOTE_ADDR         $remote_addr;
fastcgi_param  REMOTE_PORT         $remote_port;
fastcgi_param  SERVER_ADDR        $server_addr;
fastcgi_param  SERVER_PORT        $server_port;
fastcgi_param  SERVER_NAME        $server_name;
# PHP only, required if PHP was built with
#--enable-force-cgi-redirect
fastcgi_param  REDIRECT_STATUS    200;

```

Теперь нужно скопировать содержимое каталога **www** исходника каталог **/opt/share/www** роутера.

Дальше нужно запустить сервер

```

~ # /opt/etc/init.d/S80nginx restart
~ # /opt/etc/init.d/S79php-fcgi restart

```

На этом установка веб-интерфейса закончена. Можно проверять работу, попробовав открыть в браузере адрес [http://адрес\\_роутера:81/](http://адрес_роутера:81/)

Должна открыться страница текущего состояния системы (пока с пустыми значениями).

Если все работает - нужно настроить систему, добавив необходимые датчики, настроить профили желаемой температуры для комнат и т.д.

Далее нужно установить планировщик **cron** (если отсутствует) и настроить на запуск программы **modsensor** каждые 10 минут (здесь не привожу, т.к. настройка разных версий cron различается).