

Цыгичко Алексей Николаевич

Воронежская область

Город Воронеж

МБОУ гимназия имени А. В. Кольцова

**Устройство многоканального, дистанционного
мониторинга и управления удалёнными объектами
на основе GSM/GPRS каналов связи
с использованием “домашнего” облачного сервера
«Супер-Вера-2»**

Все материалы по проекту 2020 года: <http://185.55.40.127/sirius/2020/>
(Проект 2019 года <http://185.55.40.127/sirius/2019/>)

Целью данного проекта является создание устройства дистанционного управления удалёнными объектами по GPRS с использованием «облачного» сервера, который находится под максимальным контролем пользователя в юрисдикции Российского Рунета.

Цель я разделил на задачи:

- 1) Разработать систему дистанционного управления через чужой облачный сервер.
- 2) Создать прототип своего облачного сервера
- 3) Сделать свое исполнительное устройство, способное эффективно реализовывать удаленный мониторинг и управление по нескольким каналам связи.

Это устройство может быть полезно во всех сферах деятельности, в которых требуется дистанционное управление, особенно там, где заказчику важно защитить свои устройства (и информацию).

Я планирую использовать «Супер-Веру 2» для дистанционного мониторинга и удалённого полива растений на нашем дачном участке вместо «Супер-Веры 1», которая сейчас работает через облачный сервер «Keenetic».

После того, как через устройство будут подключены к серверу видекамеры, я планирую совсем отказаться от использования модема и роутера с целью мониторинга и управления на своей даче.

Удалённое управление объектами по интернету с использованием GPRS, в общем-то, достаточно просто. Однако все способы предусматривают привлечение в качестве своеобразного посредника облачные серверы, которые, как показывает практика, не всегда надежны и, прямо сказать, не вызывают полного доверия.

Википедия:

https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82_%D0%B2%D0%B5%D1%89%D0%B5%D0%B9

«В отчёте Национального разведывательного совета США (англ. *National Intelligence Council*) 2008 года «интернет вещей» фигурирует как одна из шести подрывных технологий, указывается, что повсеместное и незаметное для потребителей превращение в интернет-узлы таких распространённых вещей, как товарная упаковка, мебель, бумажные документы, может заметно повысить риски в сфере национальной информационной безопасности^[8]»

К сожалению, эти опасения подтверждаются.

В качестве примера можно привести систему управления поливом по GPRS, которую я собрал в 2019 году, и она сейчас установлена на нашей даче в СНТ «Горизонт» расположенном под Воронежем. Система получила название «Супер-Вера!». (Вера - это имя нашей соседки по даче, которая в отличии от нас летом почти постоянно живёт на даче и иногда по нашей просьбе поливает наш огород.)

Система состоит из следующих составных частей:

- Направленная GSM антенна;
- Роутер “Keenetic Omni”;
- Исполнительный модуль на основе ESP8266 D1;
- Модуль перезагрузки на основе ESP 8266;
- Группа исполнительных реле и краны для управления поливом.
- Две IP видеокамеры, одна из которых PTZ (установлены в дополнение к системе);

Эта система сейчас функционирует и доступна по облачному доменному имени <https://kran.dacha66.keenetic.pro>

Схема работы устройства представлена на **рис. 1**.

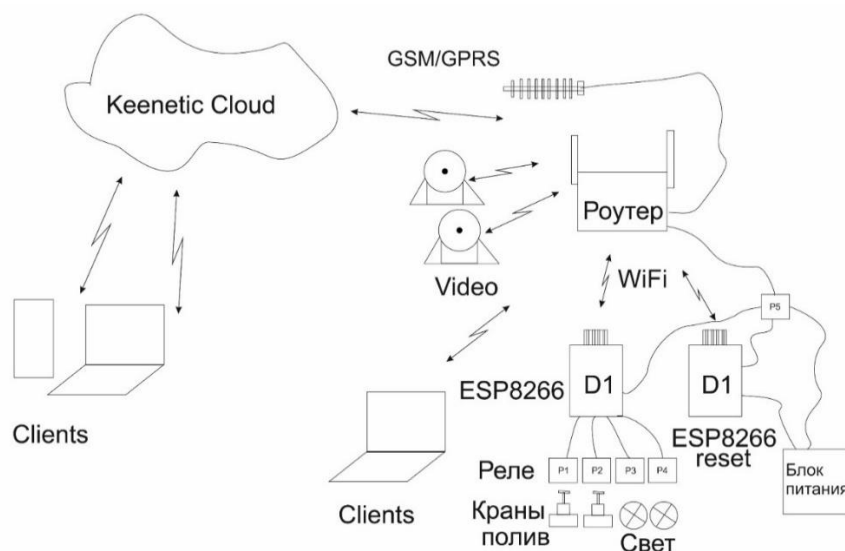


Рис. 1

Исполнительный модуль на основе ESP8266 по WiFi связан с Роутером Keenetic Omni. Роутер имеет облачное имя, которое предоставляет фирма “Keenetic”. Посредством вторичного доменного имени, которое может быть присвоено элементу сети, ESP8266 и видеокамеры доступны извне.

Для организации взаимодействия с исполнительными модулями на них развёрнуты примитивные серверы, которые выдают на поступивший запрос HTML страницы. Эти страницы отображаются браузером клиента. Клиент анализирует страницу и путем клика на соответствующий элемент формирует новый GET запрос. Сервер анализирует входящий запрос от клиента и изменяет состояние соответствующих JPIO в зависимости от содержания запроса.

В процессе эксплуатации системы проявились её недостатки.

- Роутер периодически «зависал». При обращении в сервисную службу по их рекомендации была перепрошита программа роутера для периодической (раз в 12 часов) перезагрузки. Кроме того, был введён в систему второй ESP 8266 для принудительной перезагрузки всей системы путём временного отключения питания.

- Облачный сервис периодически был недоступен и обращения в сервисную службу не давали результатов.

- Кроме этого PTZ видеокамеры, которые, как только мы их подключили к роутеру, и они стали «видны» в «глобальной» сети мгновенно

подключились к облаку XMEye и стали периодически, несанкционированно изменять свое положение!!! Что подтверждает опасения ЦРУ!

Выводы: Облачные серверы не надёжны, и существует очень большая вероятность того, что их используют или могут использовать для НСД (несанкционированных действий и шпионажа). А это нам не нужно.

Новая система получила название «Супер-Вера - 2». Взаимодействие элементов системы представлено на **рис. 2**.

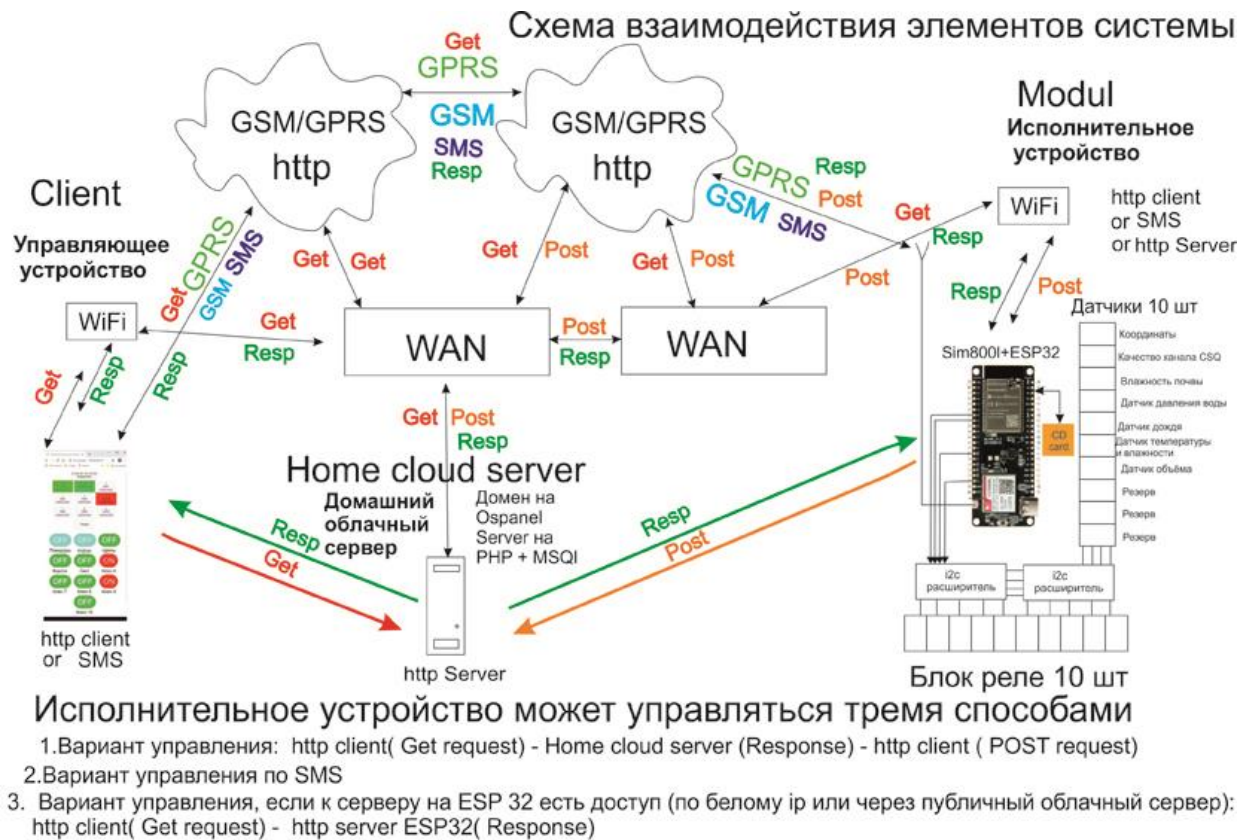


Рис. 2

Исполнительное устройство реализовано на модуле “Т-Call” **рис. 3**.

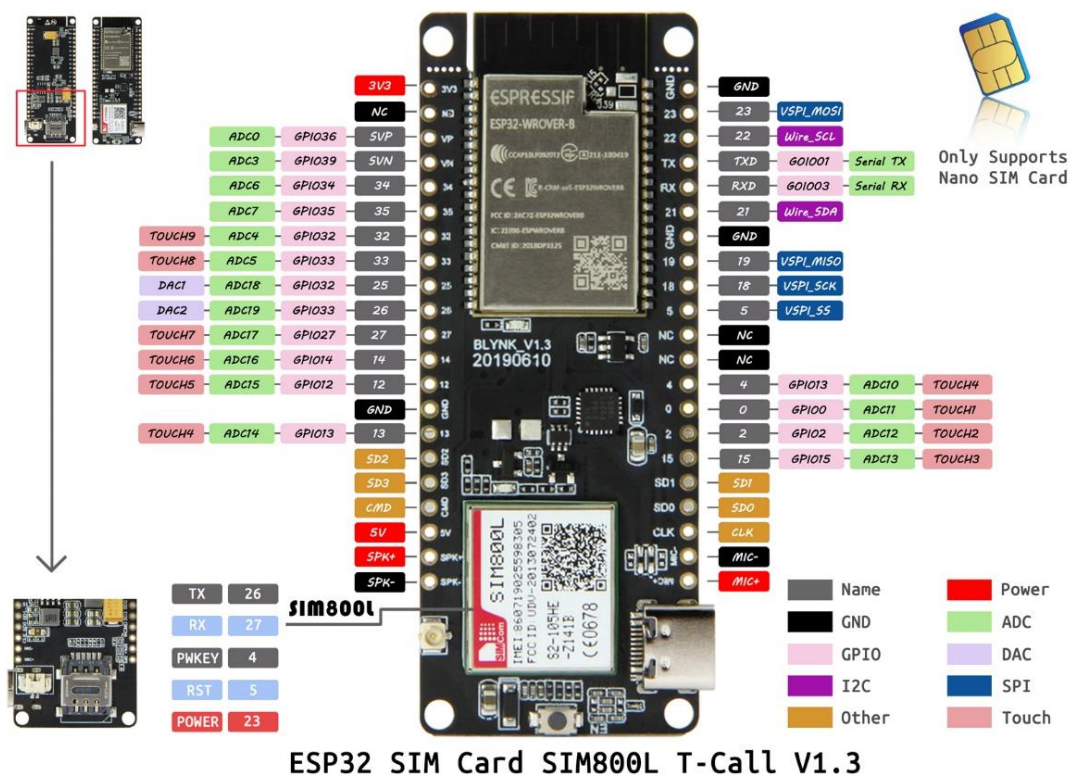


Рис. 3

Оно представляет собой сборку двух модулей GPS: модуля SIM800 и микроконтроллера ESP32. К модулю по шине I2C подключены два расширителя JPIO по 8 выходов/входов, 10 из которых используются для управления исполнительными реле, а оставшиеся 4 используются для получения информации от датчиков с релейным выходом «Да/Нет».

Ещё 4 внешних датчика подсоединяются к 4-м аналоговым входам модуля.

Кроме этого, для анализа используются два встроенных в SIM800 датчика (датчик местоположения, и датчик качества связи). Итого 10 датчиков.

В качестве дополнительной долговременной памяти к модулю по шине spi подключена SD карта, которая хранит ряд файлов необходимых для нормального функционирования системы.

Облачный сервер программно реализован на OSPanel и физически расположен на компьютере, который находится по адресу г.Воронеж ул. Фридриха Энгельса 52. Сервер, по ip адресу 185.55.40.127 порт 80 доступен из

WAN. Следует отметить, что сервер может иметь доступ через любое, в том числе и облачное доменное имя. Что успешно подтверждено в ходе отладки ПО на моём домашнем компьютере с доменным именем <https://alex.nikolay26.keenetic.link/> (может быть не всегда доступен, т.к. сервер может быть выключен), а на 185.55.40.127 сервер доступен круглосуточно. Перевод модуля с одного сервера на другой реализован программно и заключается в изменении всего одной строки в файле config.txt, который записан на SD и доступен дистанционно.

Работа системы.

Модуль может управляться по 3-м каналам.

1. По GSM с помощью СМС
2. По GSM по схеме, описанной выше как в «Супер-Вера 1».
3. С помощью «домашнего» облачного сервера.

При этом возможна совместная работа по трём каналам.

- Управление с помощью СМС заключается в том, что анализируется приходящая смс и, если она соответствует заранее записанной в памяти, то производится выполнение команды. Например СМС «11» включает реле1, СМС «90» выключает реле 9. СМС «Help» запрашивает ответную СМС о состоянии исполнительных реле и датчиков , а СМС «Config» запрашивает ответную СМС с файлом конфигурации.

- Управление по схеме «Супер-Вера 1» практически ничем не отличается от первого варианта, описанного выше был бы доступ к серверу на ESP32.

- Алгоритм работы системы по варианту номер 3:

ESP 32 при включении последовательно считывает с SD файлы в формате txt, в которых хранится набор команд для модуля SIM800 и заранее записанные ответы, которые должен выдать модуль при их выполнении: sim800linit.txt, ConfigDistr.txt, config.txt, gsm.txt, Gettime.txt, readhtml.txt и пр.

Для примера файл gsm.txt:

```
AT+CSQ@OK>
```

```
AT+SAPBR=3,1,"Contype","GPRS"@OK>
```

```
AT+SAPBR=3,1,"APN","CMNET"@ OK >
```

```
AT+SAPBR=1,1@OK>
```

```
AT+SAPBR=2,1@+SAPBR: 1,1>
```

```
AT+HTTPINIT@OK>
```

```
AT+HTTPPARA="CID",1@OK>#
```

Это набор команд для модуля SIM8001, с помощью которых модуль подключается к надстройке GPRS, а потом готовит обмен по HTTP протоколу. Требуемый ответ от модуля обозначен после @.

Терминатор строки - « > », терминатор набора команд – «#».

ESP32 последовательно выдаёт набор команд в SIM800L и ждёт на него ответ.

Программа состоит из главной программы и 14 модулей. См. папку Programs и Arduino.

Общение между клиентом и модулем происходит через сервер.

В оффлайн режиме (когда клиент не подключен к серверу) модуль все-равно посылает запросы на сервер. Это делается для отслеживания статистики датчиков. Например, если во время отсутствия клиента было произведено отключение питания модуля, то сервер через некоторое время запишет, что в это время от модуля не приходило запросов.

Теперь рассмотрим ситуацию онлайн работы. Перед началом работы клиенту нужно войти в свой аккаунт, данные которого записаны в таблице MySQL, или же зарегистрироваться. Также хочу заметить, что все пароли хранятся в базе данных в виде Хэш-Кода. Это делается для сохранности учетных данных. После входа заполненная форма отправляется на проверку. Если введенные данные были неверные, то клиента просят ввести их заново, иначе он попадает на главную страницу с интерфейсом управления модулем.

При нажатии на одну из кнопок в верхнем блоке открывается статистика переменных, поступающих от модуля. Эта статистика записана в отдельных файлах. Например seevm10, seecsq, seecrd и т.д.

51.662366,39.193955
Координаты

23 CSQ	1 Влажность почвы	1 Дождь	0 Температура	0 Влажность воздуха
0 Норма	1 Датчик газа(ц)	0 Давление воды	4095 Датчик газа(а)	Резерв

Пример части страницы статистики:

Время и дата	Значение Переменной
19.02.2020 23:43:17	4095.00
19.02.2020 23:43:03	3663.00
19.02.2020 23:42:04	4095.00
19.02.2020 23:41:50	4083.00
19.02.2020 23:37:33	4095.00
19.02.2020 23:37:19	3913.00
19.02.2020 22:46:58	4095.00
19.02.2020 22:45:09	0.00
19.02.2020 22:40:37	4095.00
19.02.2020 17:15:19	0.00
18.02.2020 17:38:14	4095.00
18.02.2020 17:37:59	4058.00
18.02.2020 17:37:16	4095.00
18.02.2020 17:37:03	3067.00
18.02.2020 17:36:49	4095.00
18.02.2020 17:36:34	3605.00
18.02.2020 17:36:15	4095.00

В самом верху находятся координаты они нужны для того, чтобы отслеживать объект, если он может находиться в движении. Далее слева сверху находится переменная CSQ, эта переменная показывает качество связи у модема. Назначение остальных датчиков подписаны под их значением.

Далее идет блок управления реле.



Их цвет зависит от состояния реле, а также от того, как было изменено их значение. Кнопка будет мигать, если клиент нажал на кнопку и отправил

команду модулю, но модуль пока не ответил (кнопка мигает тем цветом того состояния, в которое её перевел клиент). Зелёным кнопка горит, когда реле было переключено клиентом через сервер в выключенное состояние, красным же во включенное. Голубым и оранжевым они горят, если через СМС они были выключены или включены соответственно.

В самом низу интерфейса есть 2 кнопки: Admin и Exit. Они нужны для того, чтобы открыть систему управления файлами модуля и для выхода из аккаунта соответственно.



Ниже приведен пример страницы системы управления файлами модуля. В которую для изменения загружен файл help.txt

Имя файла:

/help.txt

Файл сейчас

Запросить файл

Очистить

Файл:

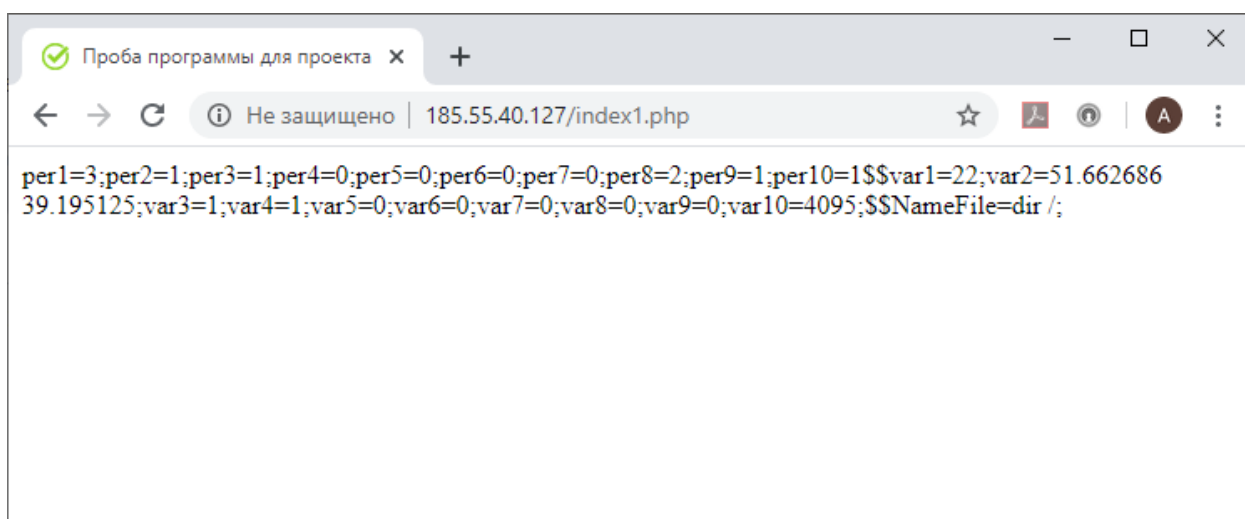
Отправить измененный файл

Очистить

```
1 :off= Poliv ogurcy
2!ON! = Poliv ovoshi
3!ON! = Osveshenie dvor
4!ON! = -----
5 :off= -----
6 :off= -----
7 :off= -----
8!ON! = -----
9!ON! = -----
10!ON! = -----

turn on --- "Num 1"
turn off -- "Num 0"
```

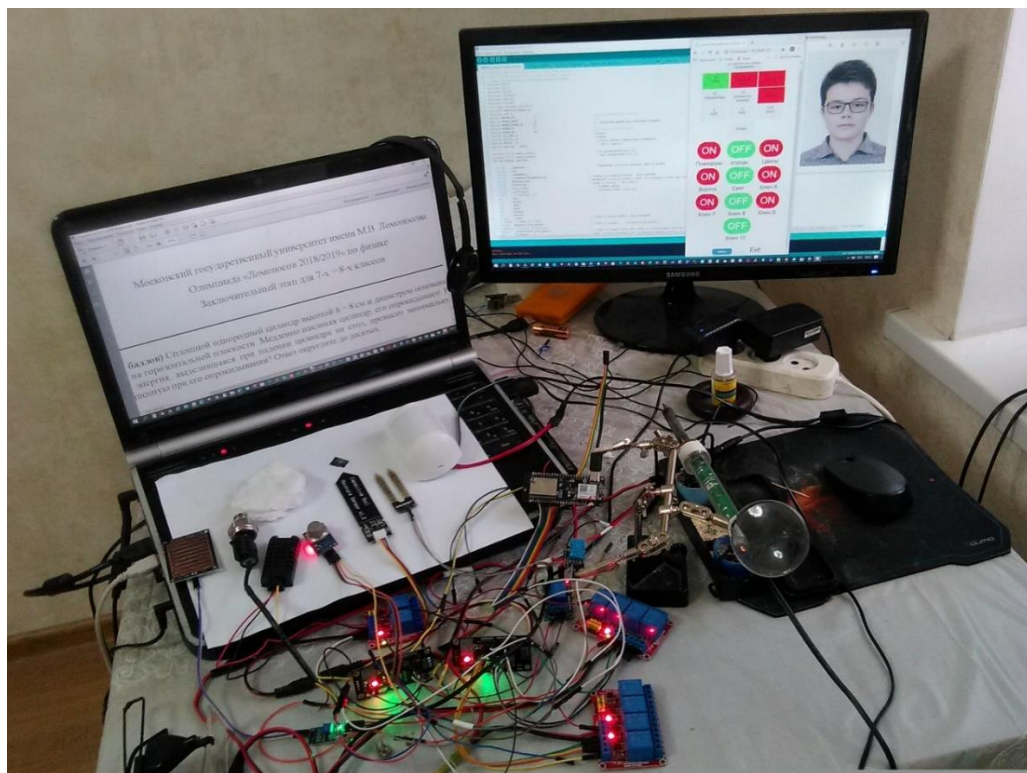
Для общения с модемом на сервере есть специальная страница `index1.php`



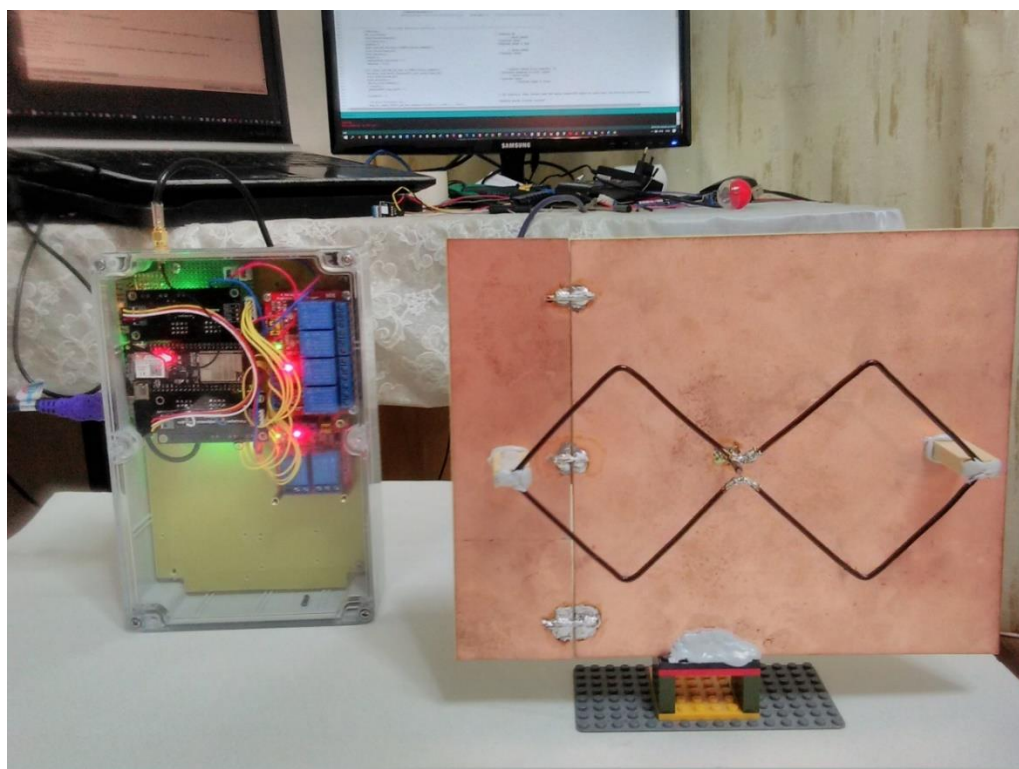
Модуль считывает данные со страницы, сравнивает их со своими значениями и отдает нам новые значения. Например, `per 1-10` – это значения реле, которые отдает сервер модулю. То есть, например, при переключении 1 реле с `on` на `off` или наоборот, модуль считает `per1=0` или `per1=1` соответственно, дальше в `var 1-10` идут переменные с датчиков, которые нам отдает модуль. А в поле `NameFile` мы отдаем и имя файла и то, что мы хотим с ним сделать. Например, если перед названием файла стоит `red`, то мы хотим, чтобы модуль отправил нам этот файл, а если перед ним стоит `write`, то мы хотим, чтобы модуль перезаписал уже измененный нами файл, а если ничего, то ничего модулю делать и не надо. Все происходит в `POST` запросах. Также для защиты модуль передает переменную `PS` (сокращение от `Password`), если `PS` не совпадает с паролем, то переменные не меняют своих значений.

В данный момент проект находится на стадии доработки. Но работа над ним идет в ускоренном режиме. Для примера можно привести 2 фотографии, сделанные с разницей во времени 1 неделя.

До:



После:



У этого проекта есть и аналоги. Главными конкурентами я посчитал комплекты умного дома от Xiaomi и от Ростелекома.

Xiaomi:



Ростелеком:



Основным плюсом моего проекта является многоканальность. Во всех аналогах существует только подключение по Wi-Fi. Данный проект реализует сразу 3 возможных способа управления. Это очень удобно, ведь не у всех людей есть возможность установить Wi-Fi роутер (например, в отдаленных деревнях), или не во всех районах есть достаточно хороший мобильный интернет.

Также плюсом является относительно низкая цена. На просторах интернета минимальную стоимость за комплект Xiaomi была 4200 рублей, а за набор Ростелекома – начальный взнос (за самый маленький комплект) 11500руб.

Себестоимость моего устройства составляет:

700 руб.(sim800) + 850 руб.(реле) + 500 руб.(I2C) + 500 руб.(датчики) + 300 руб.(коробка) + 250 руб. (расходные материалы) \approx **3100** рублей \pm 500руб. (зависит от места покупки, типа датчиков, курса доллара и т.д.) + абонентская плата за sim-карту(зависит от тарифа, оператора и т.д.).

Вывод: в ходе проделанной работы был разработан собственный облачный сервер. Создано устройство, которое эффективно справляется с поставленной задачей дистанционного управления удалёнными объектами по GPRS с использованием «облачного» сервера, который находится под максимальным контролем пользователя.

Также был найден способ совместить управление по Wi-Fi, через Сервер и по СМС (во время работы с аналогами были найдены только устройства, в которых использовались максимум 1-2 из этих способов).