



Все важнейшие
IT-события здесь

5

6

7

8

9

10

11

12

13

14

15

16

17

18



empenoso

1 июн 2020 в 05:23

Умная хрущёвка на максималках

🕒 10 мин 👁 63K

Open source*, Веб-дизайн*, Гаджеты, Настройка Linux*, Умный дом

Моя статья [про проводной умный дом](#) была популярной на Хабре (159 тысяч просмотров) — в ней я описал положительный опыт от идеи до реализации умного дома в новостройке.

Вместе с тем, у меня есть опыт интеграции и со старыми домами — квартира в типовом советском пятиэтажном кирпичном доме семидесятых годов постройки уже 5 лет автоматизирована. Она имеет единую панель управления для разных подсистем и не зависит от облачных сервисов. В отличие от новостройки, эта квартира в пятиэтажке была автоматизирована не сразу, а поэтапно за несколько лет, по мере возникновения интереса к разным элементам управления и контроля, таким как:

1. мониторинг приборов учета;
2. датчики контроля: домофона, температуры, шума, открытия дверей, ...;
3. единая панель управления умным домом;
4. подъездное видеонаблюдение;
5. настенный дисплей для текущего времени и температуры за окном;
6. освещение в квартире, включая кнопку «выключить всё»;
7. удаленное обесточивание квартиры;
8. удаленное перекрытие воды.



3D модель этой двухкомнатной квартиры 47 кв.м, которая была автоматизирована, нарисованная в Sweet Home 3D

Мониторинг счётчиков воды и электричества

Начну рассказ с самого начала — в 2016 году мне потребовался удаленный сбор показаний квартирных счетчиков.

Для этого, чтобы не устанавливать «считывающие накладки», в любом случае, была необходима замена обычных счётчиков воды и электроэнергии на счетчики с соответствующим интерфейсом. И вариантов этого интерфейса было доступно всего два:

1. Цифровой RS485.
2. Импульсным выход.

Если говорить о счётчиках на воду, то разница в цене по этим двум вариантам составляла примерно 10 раз. Обычный счетчик на воду с импульсным выходом стоил рублей на 200 больше обычного, то есть где-то 400...600 рублей, а счетчик воды с RS485 стоил порядка 3...5 тысяч рублей.

Из-за большой ценовой разницы я выбрал модель с импульсным выходом [Valtec VLF-R-IL](#)

(сейчас эта модель уже не производится).

Была еще одна особенность, из-за того что я живу в Перми — возникла проблема просто найти и купить эти счётчики с импульсным выходом — их не было в наличии и фактически я потратил целый месяц на ожидание доставки по России — ведь нельзя просто купить в Китае счётчик и установить — это оборудование, которое должно «поверяться» и обладать соответствующим сертификатом, иначе на учёт его не поставят. В Москве, насколько я знаю, подобных проблем нет и наоборот управляющие компании заставляют устанавливать счётчики на воду с импульсным выходом.



Счётчики на воду с импульсным выходом сразу после установки и опломбирования

Следующим моим шагом стал выбор счётчика электрической энергии — и здесь тоже надо иметь в виду, что он тоже должен обладать всеми необходимыми бумагами и сертификатами для постановки на учёт — ведь я слышал, что иногда ставят даже два счётчика:

1. Первый — «отчётный» — для учёта, перед поставщиком электроэнергии.
2. Второй — без российских документов с популярного китайского сайта, который устанавливают сразу после первого для использования в целях домашней автоматизации — чтобы можно было удаленно считывать показания. Обычно через цифровой интерфейс RS485.



Однофазный односторонний счетчик Меркурий 201.5 сразу после установки и опечатывания

В 2016 году я выбрал односторонний счётчик с импульсным выходом, потому что я не понимал, как можно из командной строки сервера считывать показания счетчика электроэнергии по RS485 и пушить их в единую панель управления умным домом, которую я собирался собрать.

Как оказалось позже — однофазные электросчетчики Меркурий достаточно популярны для домашней автоматизации и при цене счетчика, сопоставимой с аналогичными счетчиками других производителей можно получить устройство, которое полностью совместимо с

современными системами домашней автоматизации. Для этого надо лишь иметь [адаптер USB -> RS485](#), стоимость которого составляет менее 100 рублей (в Китае). Этот адаптер может работать с любым микрокомпьютером у которого есть USB-порт, например, из линейки Raspberry Pi, которые часто используются в качестве сервера для домашней автоматизации, для запуска Home Assistant [или любого другого](#) программного хаба.

Сам скрипт опроса электросчетчика Меркурий написан энтузиастами и выполняется из командной строки, а также может быть интегрирован в любую систему домашней автоматизации. Мой конкретный пример интеграции для RS485 в другой квартире, где установлен Меркурий 200, который я сделал уже позже, можно посмотреть, например, [на форуме](#) или просто поискать варианты [в поиске](#).

```

38 all_time_phase 6'
39
40 # tariffs list
41 ZONES='T1 01
42 T2 02
43 T3 03
44 T4 04'
45
46
47
48 # commands list
49 # alias command size answer parser format description
50 COMMANDS='test `form_com x00${CSN}x00` 4 Тестирование связи
51 #close `form_com x00${CSN}x02` 4 Завершение сеанса
52
53 kwatthour `form_com x00${CSN}x27` 23 "02x${((${zona} * 4 - 3)),02x${((${zona} * 4 - 2)),02x${((${zona} * 4 - 1)),02x${((${zona} * 4)})#Aprintf("\%0.2f",^b0/100)|02x5,02x6,02x7,02x8#Aprintf("\%0.2f",^b0/100)|02x9,02x10
54 kwatthour_all `form_com x00${CSN}x27` 23 "02x1,02x2,02x3,02x4#Aprintf("\%0.2f",^b0/100)|02x5,02x6,02x7,02x8#Aprintf("\%0.2f",^b0/100)|02x9,02x10
55 kwatt_summ `form_com x00${CSN}x27` 23 test
56 kwatt_summ2 `form_com x00${CSN}x27` 23 Опрос накопленной энергии с выводением информации
57 kwatt_month_summ `form_com x00${CSN}x32x0${month}` 23 test
58
59 amper `form_com x00${CSN}x63` 14 "02x3,02x4#Aprintf("\%s=%0.2f",^b0/100)" Сила тока A (A)
60 power `form_com x00${CSN}x63` 14 "02x5,02x6,02x7#Aprintf("\%s=%0.3f",^b0/1000)" Мощность P (кВт)
61 volt `form_com x00${CSN}x63` 14 "02x1,02x2#Aprintf("\%s=%0.1f",^b0/10)" Напряжение U (В)
62 avp `form_com x00${CSN}x63` 14 Сила тока A (A) Напряжение U (В) Мощность P (кВт)
63
64 null `form_com x00${CSN}${mode}` ${MAX_BLOCK_SIZE} "${OutParser}" Произвольная команда
65
66 batvolt `form_com x00${CSN}x29` ${MAX_BLOCK_SIZE} "2x1#Battery\|0=\\0\|t.#0#2#\|t\|0V" Напряжение батареи
67 serialnum `form_com x00${CSN}x2f` ${MAX_BLOCK_SIZE} "1,2,3,4#d#Sn\|0=\\0\|t" Серийный номер счетчика
68 version_date `form_com x00${CSN}x28` ${MAX_BLOCK_SIZE} "02x4#\|t.#0#02x5#\|t.#0#02x6" Дата версии ПО
69 version `form_com x00${CSN}x28` ${MAX_BLOCK_SIZE} "1#version=\\t#0#2#\|t" Версия ПО
70 #datetime `form_com x00${CSN}x21` ${MAX_BLOCK_SIZE} "2#0#3#\|t:#0#4#\|t\|0#0#5#\|t.#0#6#\|t.#0#7" Дата время по счетчику
71 #datemake `form_com x00${CSN}x66` ${MAX_BLOCK_SIZE} "02x1#0#02x2#\|t.#0#02x3" Дата изготовления
72 #last_on `form_com x00${CSN}x2c` ${MAX_BLOCK_SIZE} "2#0#3#\|t:#0#4#\|t\|0#0#5#\|t.#0#6#\|t.#0#7" Время последнего включения
73 #last_off `form_com x00${CSN}x2b` ${MAX_BLOCK_SIZE} "2#0#3#\|t:#0#4#\|t\|0#0#5#\|t.#0#6#\|t.#0#7" Время последнего выключения
74 datetime `form_com x00${CSN}x21` ${MAX_BLOCK_SIZE} "5#datetime=\\t#0#6#\|t.#0#7#\|t\|0#0#2#\|t:#0#3#\|t:#0#4" Дата время по счетчику
75 datemake `form_com x00${CSN}x66` ${MAX_BLOCK_SIZE} "02x1#make=\\t#0#02x2#\|t.#0#02x3" Дата изготовления
76 last_on `form_com x00${CSN}x2c` ${MAX_BLOCK_SIZE} "5#0n=\\t#0#6#\|t.#0#7#\|t\|0#0#2#\|t:#0#3#\|t:#0#4" Время последнего включения
77 last_off `form_com x00${CSN}x2b` ${MAX_BLOCK_SIZE} "5#0ff=\\t#0#6#\|t.#0#7#\|t\|0#0#2#\|t:#0#3#\|t:#0#4" Время последнего выключения

```

Элемент скрипта для получения данных с электросчетчика Меркурий 200

Однако в 2016 году после установки счетчиков без цифрового RS485 мне понадобилось устройство считывания этих импульсов.

Выбор учетного и управляющего контроллера

Поскольку я выбрал и установил три счетчика с импульсными выходами, мне надо было иметь какое-то устройство, которое было бы всегда на связи с ними и круглосуточно

считывало бы эти импульсы, конвертируя их в «человеческий формат». Я стал искать подобный контроллер и нашёл несколько вариантов:

1. [Контроллер с программируемой логикой ОВЕН](#).
2. Модули счётчика импульсов от разных производителей, например [vot](#).
3. Универсальный проводной контроллер [Mera](#) с возможностью интеграции в систему умного дома.

Расскажу обо всем подробнее. Перед тем как покупать что-либо, я решил обратиться с запросом к первому найденному варианту — к ОВЕН и в его техподдержке мне буквально написали, что моя идея влетит в копеечку. Ответ техподдержки:

"В целом система получается достаточно дорогостоящей + реализация ПО под Вашу задачу -
...
Можно и более дешевый, но WEB-визуализации у них нет. Тут потребуется Scada система уже

Но, к любому ПЛК, по любому интерфейсу связи можно сделать визуализацию (это не WEB-виз
Также рекомендую посмотреть каталог проектов <http://www.owen.ru/projects> с фильтрацией

Получается, что проект может обойтись очень дорого, а тратить несколько десятков тысяч рублей на это мне совершенно не хотелось, но удаленный учёт все же хотелось иметь, поэтому я перешёл к следующему варианту.

Вторым вариантом была программно-аппаратное решение по учету импульсов от одного из российских производителей, но эта система была закрыта и позволяла только учитывать импульсы, а мне хотелось иметь и другие функции умного дома — такие как детекция уровня шума в децибелах и учёт температуры стояков. Цена вопроса данного варианта составляла около 5 тысяч рублей, насколько я помню.

Поэтому я продолжил гуглить дальше и в поисках третьего варианта и в итоге наткнулся на Мегу — проект Андрея Буртового из города Тольятти, который вырос из простого хобби в коммерческий проект. В нём подкупало соотношение цены и заявленных возможностей — по цене комплект (контроллер и модуль только с аппаратно-конфигурируемыми входами) выходил примерно на уровне второго варианта — простого счетчика импульсов с веб-интерфейсом. Но при этом появлялась возможность интеграции с существующими [программными хабами](#).



После выбора и монтажа оборудование в 2016 году «сердце» начинающей умной квартиры выглядело так. От обилия блоков питания я потом избавился (фото ниже)

Провода всё таки нужны

Поскольку все три найденных решения были проводными, мне пришлось тянуть витую пару UTP cat.5E до каждого из счетчиков. Ещё я захотел добавить несколько недорогих датчиков и поскольку выбранный контроллер Мега был проводной мне пришлось разработать схему что и куда надо протянуть, чтобы удовлетворить все потребности в проводах. Ещё решил поставить следующие датчики:

- Улавливания наличия газа на кухне (т.к. в доме есть газ).
- Температурный датчик, который отображает температуру, влажность и атмосферное давление и текущий уровень освещенности на улице.
- Датчик на батарею, чтобы знать её температуру.

- Датчик открытия двери — простой геркон.
- Температурные датчики на холодную и горячую воду.
- Датчик шума в гостиной.

Схема протяжки витой пары выглядела так:

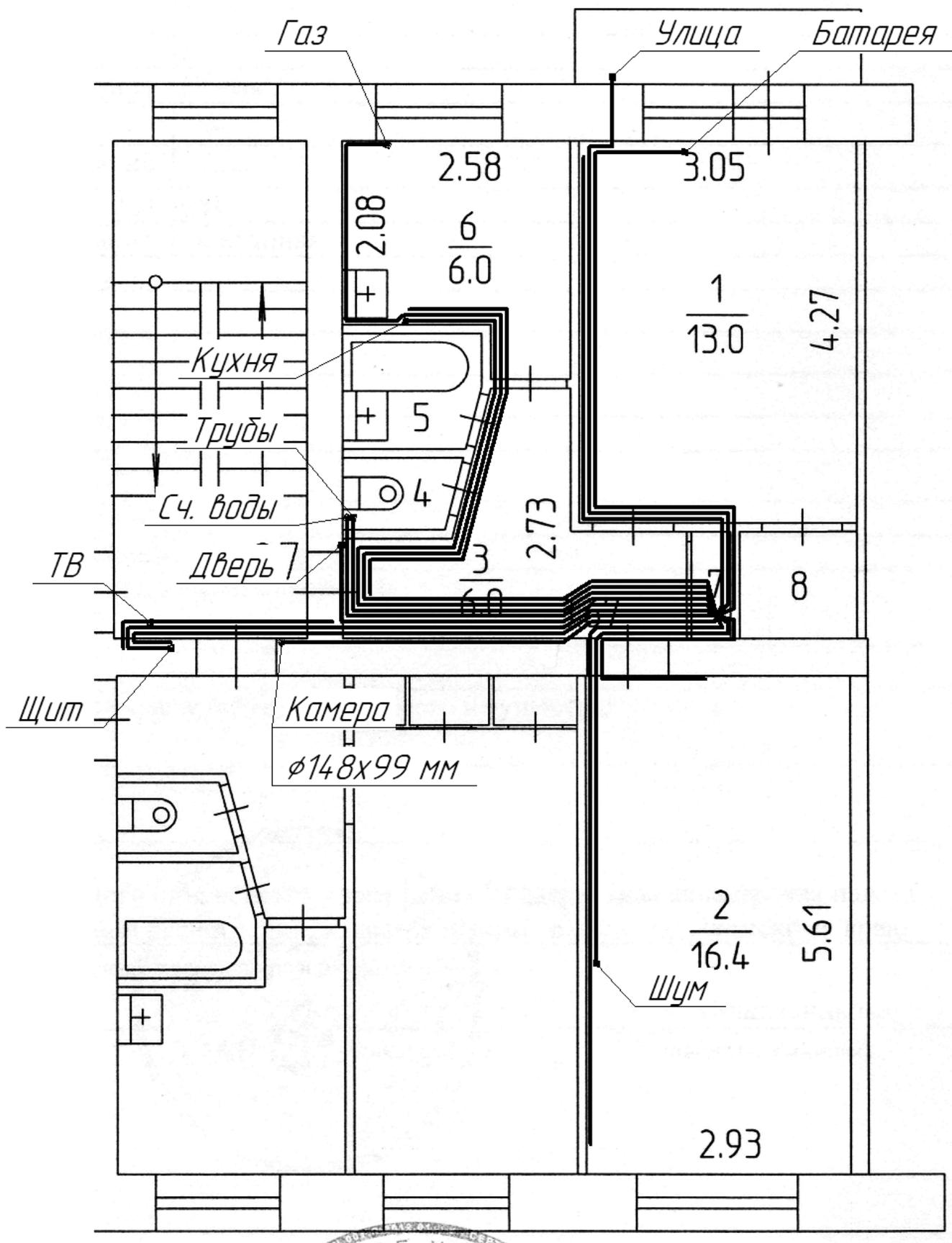


Схема прокладки витой пары в двухкомнатной квартире 47 кв. м. и на лестничной площадке

Штроб в квартире не было сделано — всю витую пару удалось протянуть в пластиковых коробах (размером 15x10 мм), расположенных чуть выше уровня плинтуса.

В этом месте необходимо сказать, что в хороший существующий ремонт вписать всё это будет весьма проблематично, но в этой квартире его не было и терять было нечего.

Также мне пришлось разработать схему подключения датчиков непосредственно к контроллеру. Схема подключения датчиков к Меге в варианте 2016 году ниже:

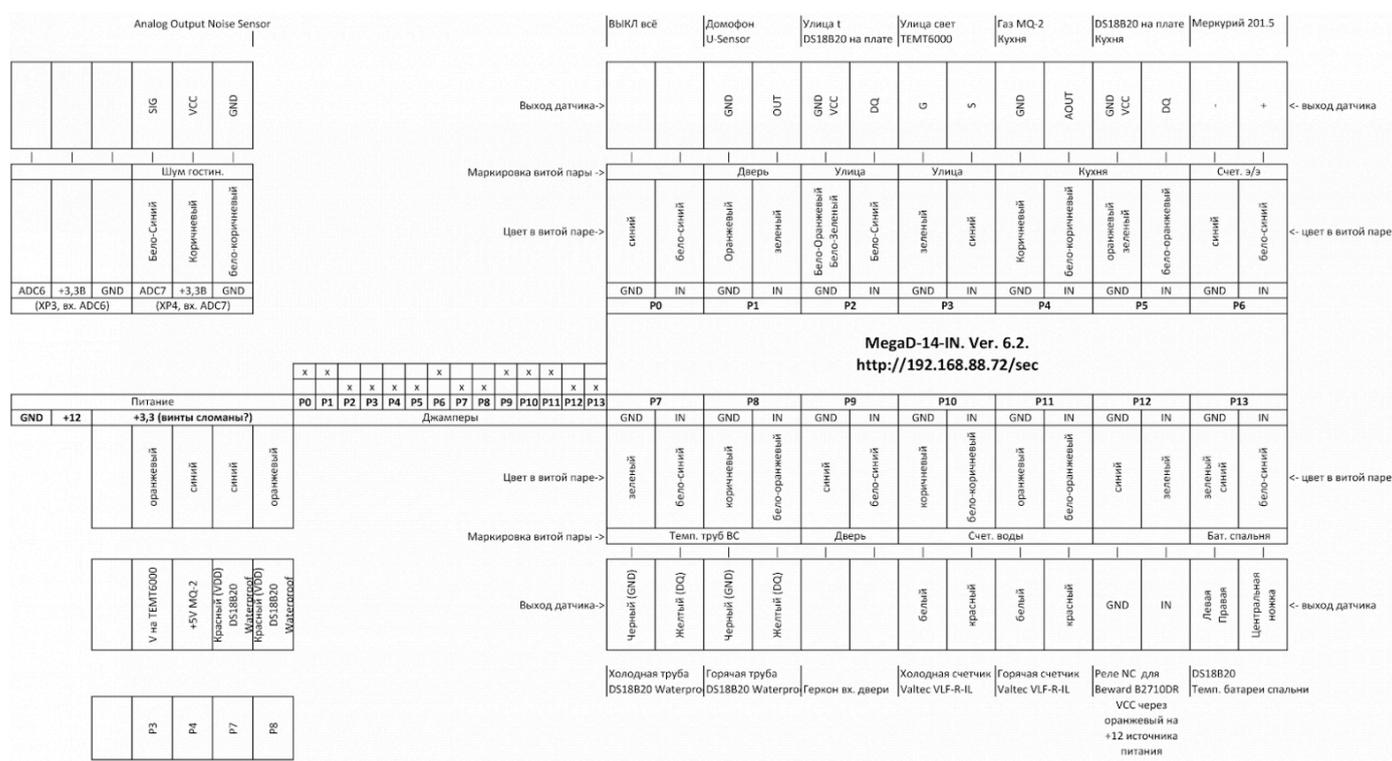


Схема подключений датчиков на Мега-328

Через пару лет в 2018 году я заменил Мега-328 на более новую 2561 и избавился от аналоговых датчиков, которые иногда работали некорректно и зависали, а для исправления требовалась полная перезагрузка контроллера. После замены я поставил больше точных цифровых датчиков с I2C интерфейсом, объединив их на нескольких портах:

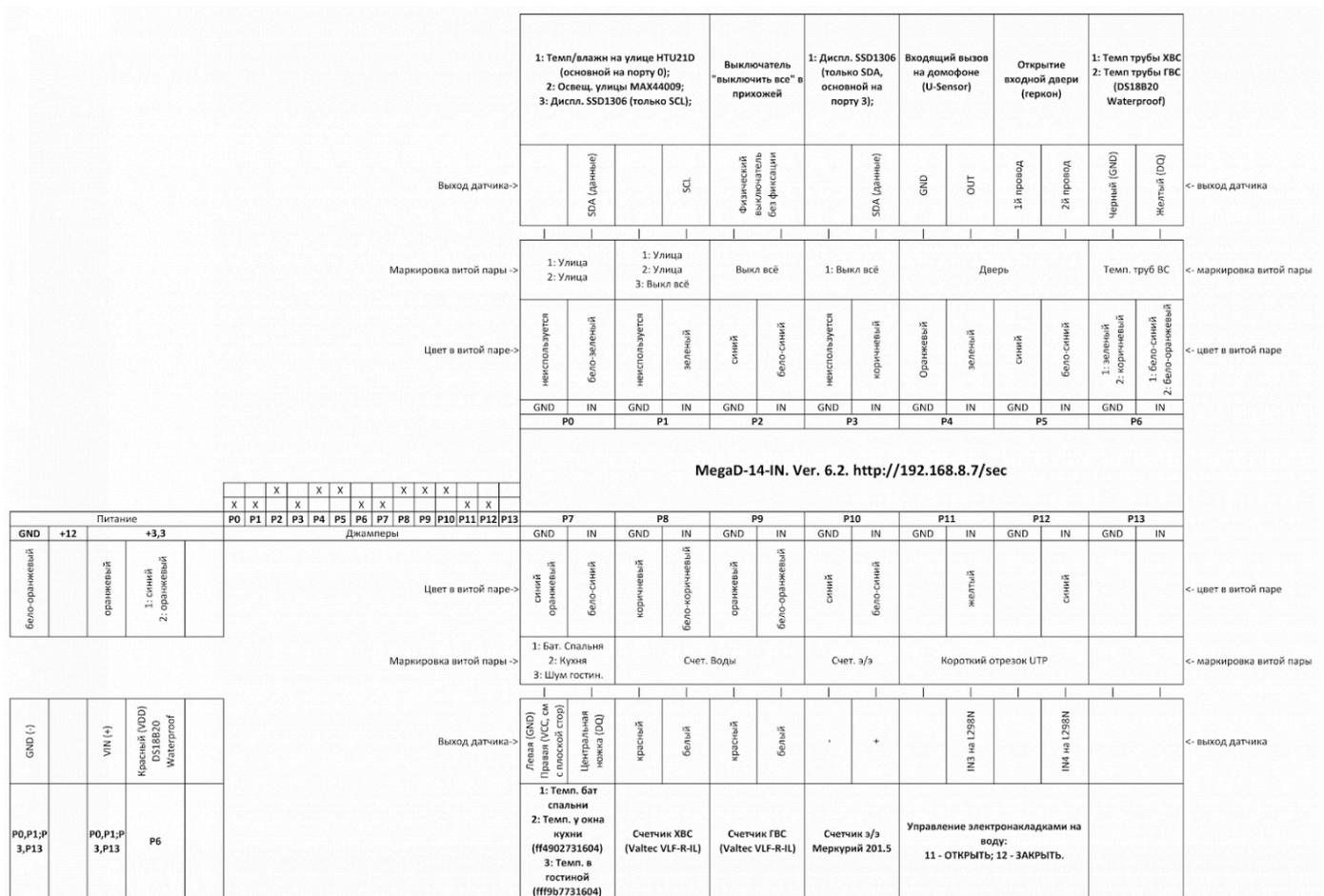


Схема подключений на Мега-2561

Датчики температуры

Для измерения температуры часто используются датчик DS18B20, который представляет из себя цифровой термометр с интерфейсом 1-Wire, который может снимать показания в диапазоне $-55...125^{\circ}\text{C}$. Для конкретно моего контроллера DS18B20 датчики можно подключать шиной — это значит по несколько датчиков на один порт, что очень экономит порты, и из-за этого автоматизация выходит ещё дешевле.



Внешний вид датчика DS18B20 для закрепления установки на батарее отопления

В Home Assistant эти данные с одного порта разбираются очень просто — у каждого из этих температурных датчиков есть [свой уникальный адрес](#), по которому они и распознаются контроллером.

Главный недостаток DS18B20 — они реально очень мелкие и поэтому электрику с ними работать сложно и непривычно.



Датчик DS18B20 в термоусадке, закрепленный на батарее отопления

Интеграция квартирного домофона в общую систему

Квартирный домофон тоже подключен к общей системе, но только в информационном режиме — вызов виден, но открыть дверь нельзя. Подключение осуществляется через датчик [U-Sensor \(5B\)](#), который подключен параллельно домофонной трубке (координатный домофон).

Для справки — я недавно нашел решение, которое позволит не только видеть вызов, но и

открыть дверь, но пока прочитал про это только в [чужой статье](#).

Датчики температуры на холодную и горячую воду

Поскольку я уже не раз сталкивался с тем, что горячая вода была не всегда так уж и горяча, я решил добавить датчики температуры на воду, которые бы непрерывно фиксировали температуру для того чтобы контактировать с управляющей компанией, если будут проблемы с температурой воды. В качестве датчиков температуры воды я решил использовать [цифровой термометр DS18B20](#), который выпускается в разных исполнениях, в том числе и в водозащищенном.



Цифровые термометр DS18B20 с интерфейсом 1-wire, закрепленные на стяжках.

Дисплей у двери

Вообще, я нашел идеальный формат панели управления у двери [в своей статье на Хабре](#), но это случилось только в начале 2020 года. А этот дисплейчик для отображения времени и погоды за окном у входной двери был установлен гораздо раньше, за пару лет до этого момента, и работает до сих пор.



Внешний вид — OLED дисплей 128x64 на SSD1306 с управлением по I2C в коробке заводского изготовления

Дисплеем надо управлять и в данном случае я делаю это через контроллер Мега, посылая управляющие сигналы с управляющего программного хаба. Вот мои варианты правил управления этим дисплеем для систем [openHAB](#) и [Home Assistant](#).

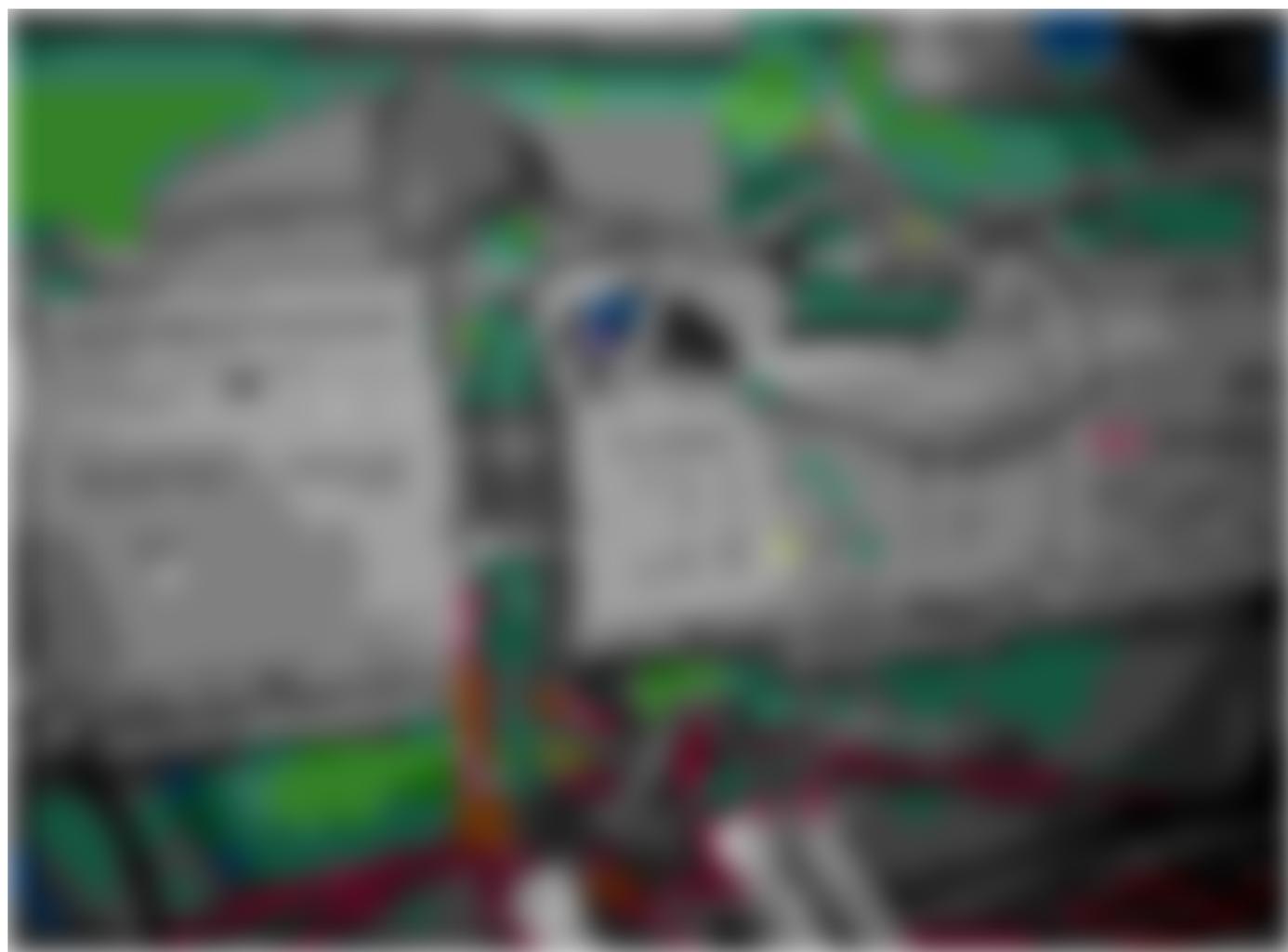
Датчик шума внутри умного дома

Это отдельная и очень длинная история, которая растянулась на несколько лет поисков из-

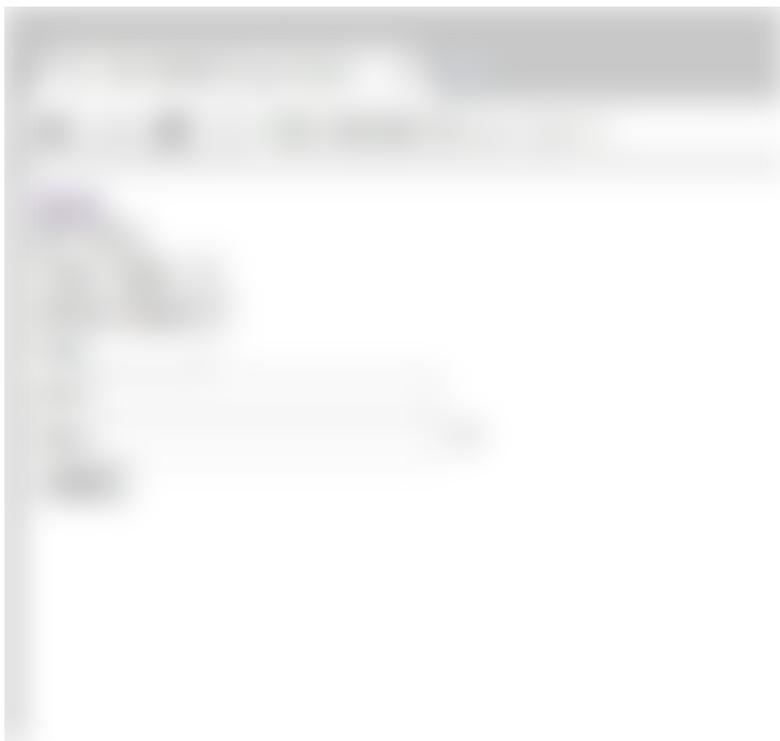
за того, что я хотел использовать дешёвые датчики, но как оказалось, для корректного получения уровня шума в децибелах, это невозможно.

Почему я вообще заинтересовался шумом? Существует допустимый уровень шума, который установлен в законе о санитарно-эпидемиологическом благополучии населения, а также СанПиН 2.1.2.2645-10 и СН 2.1.8.562-96. По этим правилам дневное время — это время с 7:00 до 23:00, ночное — с 23:00 до 7:00. Нормальный уровень фонового шума для жилых комнат и квартир днём — 40 децибел, ночью — 30 децибел. Максимальный уровень непостоянного шума — 55 децибел в дневное время и 45 децибел в ночное. Нарушением считается любой звук, который превышает норму и мешает жильцам в любое время, кроме новогодней ночи.

Сначала я пытался использовать [дешёвые модули, которые заявлены как модули детекции шума](#), подключая их к АЦП контроллера Меги чтобы определить уровень шума, но это было безрезультатно. Одна из попыток на фото ниже:



Один из датчиков, подключенных к аналого-цифровому преобразователю (АЦП) Меги



Так выглядел датчик, приведенный выше в веб-интерфейсе Меги

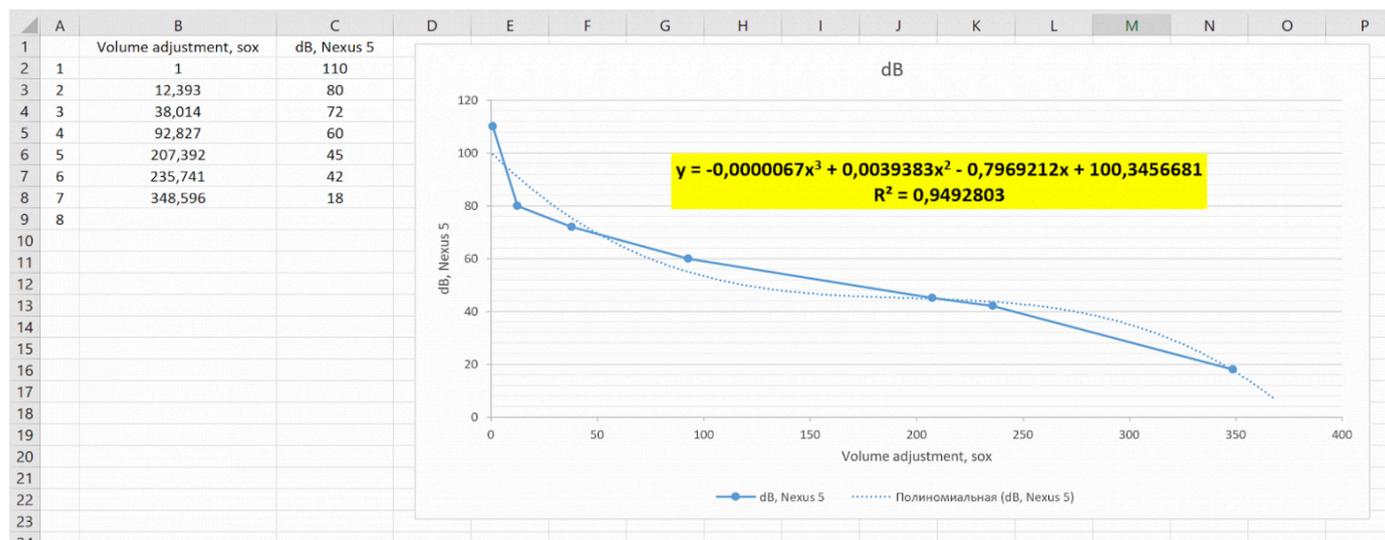
Контроллер видел эти дешевые датчики, получал с них какие-то цифры, но эти значения с реальным уровнем шума никак не коррелировались.

Первые пару лет уровень шума хоть и очень опосредованно удавалось получать с [USB микрофона](#), воткнутого в порт Raspberry Pi через bash-скрипт, выполнение которого вызывалось из openHAB: `#!/bin/bash`

```
/usr/bin/arecord -D plughw:1,0 -d 15 -f S16_LE /home/openhabian/USB_dB.wav 2>/dev/null;
sleep 15;
sox /home/openhabian/USB_dB.wav -n stat 2>&1 | sed -n 's#^Maximum amplitude:[^0-9]*\([0
```

Вот мой скрипт [на GitHub](#). Полученное значение Volume adjustment пересчитывалось в децибелы уже внутри [правил автоматизации OpenHAB](#). Конечно это был «костыль», но хотя бы, опосредовано, он работал.

В 2016 году я даже пытался найти формулу для расчета корректного значения уровня шума в децибелах, основанную на показаниях шумомера в Nexus 5 и значениях, получаемых с USB микрофона и вывел её (на скриншоте ниже $y = -0,0000067x3 + \dots$). Это несколько месяцев работало, но потом уровень получаемых с микрофона цифр неожиданно уплыл и больше не возвращался. После этого я нашел [другую формулу](#): `[dB = 20 x log10(A)]`, которая и показана в файле [с правилами для openHAB](#).



Попытка вывести формулу, основанную на показаниях шумомера в Nexus 5 и значениях, полученных с USB микрофона

Недостатками решения с USB микрофоном были:

1. Возможное нарушение конфиденциальности — хотя звук в привычном понимании и не записывался и анализировался только wav файл, но это все же был USB микрофон.
2. Невозможность выноса — удлинить USB кабель больше чем метра на три не представляется возможным.
3. Неточность — уровень измерений плавал — и в какой-то момент из графика было понятно что шум усилился, но на сколько сильно это произошло в реальности было непонятно.

Эпопея с поиском доступного решения завершилась только в 2019 году, когда мне подсказали решение — плату [Gravity: Analog Sound Level Meter от DFRobot](https://gravity.name/), которая хоть и вышла по цене около 3,5 тысячи рублей, но закрыла все проблемы и с расстоянием и с конфиденциальностью и конечно же с точностью.



Плата измерения уровня шума от американского производителя DFRobot

Для модуля измерения уровня шума даже не потребовалось сложное правило пересчета — внутри Home Assistant это происходит [на уровне описания элемента](#):

```
- platform: mqtt
  name: "Датчик шума"
  device_class: signal_strength
  state_topic: "megad/7/36"
  value_template: "{{ (value_json.value|float/1023*3.3*50) | round(1) }}"
  unit_of_measurement: "дБА"
```

Итог

Поскольку статья про автоматизацию квартиры получалась очень большая, я решил разбить её на две части, где в первой части, которая сейчас перед вами, я рассказываю только о том, как собирал информацию о состоянии квартиры с различных датчиков, а во

второй части (**UPD: уже готова**) буду рассказывать о том, как уже принялся активно управлять всеми системами и выполнил:

- Полную беспроводную автоматизацию всего освещения.
- Кнопку «выключить всё».
- Полное удаленное обесточивание квартиры (включая и сам сервер управления вместе с контроллером) и обратное удаленное включение.
- Электронакладки (устройства поворота) для кранов на воду.

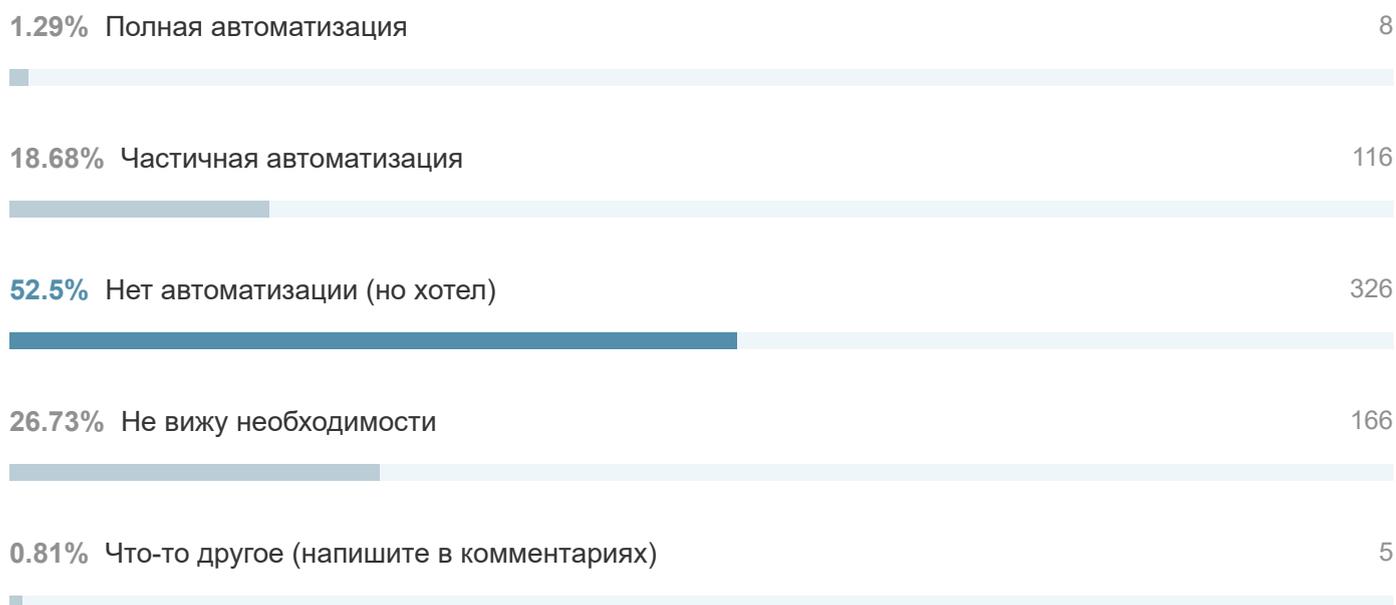
Также во второй части я рассказываю, что бы я сделал по другому, если бы делал это сейчас, спустя несколько лет.

Автор: [Михаил Шардин](#).

1 июня 2020 г.

Только зарегистрированные пользователи могут участвовать в опросе. [Войдите](#), пожалуйста.

Вы живете в квартире/доме с домашней автоматизацией?



Проголосовал 621 пользователь. Воздержались 48 пользователей.

Теги: [esp8266](#), [diy](#), [умный дом](#), [sonoff](#), [electrodragon](#), [openHAB](#), [Home Assistant](#), [ПЛК ОВЕН](#), [MegaD](#)

Хабы: [Open source](#), [Веб-дизайн](#), [Гаджеты](#), [Настройка Linux](#), [Умный дом](#)

Редакторский дайджест



Присылаем лучшие статьи раз в месяц



179

30.4

Карма

Рейтинг

Михаил Шардин [@empenoso](#)

Разработчик

Подписаться



[Сайт](#) [Сайт](#) [Github](#)

Комментарии 106

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



Erwinmal

5 часов назад

Кто поджёт Лос-Анджелес? Свежая конспирология о виноватых НЛО, Пи Дидди, урбанистах и корюшке



Простой



14 мин



3.8K

[Обзор](#) +25 4 34**DimDimDimDimDim**

6 часов назад

Rust 1.84: новый релиз отличного языка программирования. Еще лучше, еще эффективнее, как всегда

 6 мин  2.5K +17 8 4**JBFW**

14 часов назад

Подключаем длинную линию 1-wire к Ардуино

 3 мин  4.3K +17 32 28**DAN_SEA**

1 час назад

«Профессор, конечно, лопух, но аппаратура при нём» — или немного о костной проводимости

 Средний  9 мин  653[Обзор](#) +13 2 3**arturdumchev**

1 час назад

Заговор разработчиков против корпораций

 Средний  15 мин  1.7K[Мнение](#) +11 6 6

**DENEVGSTAR**

5 часов назад

Распознавание образов в мозге с помощью микроплееров

Средний 8 мин 1K

Из песочницы

+11

16

6

**chlorine**

7 часов назад

Кэш. Теория кэширования. Устройство и разновидности кэша

Простой 7 мин 2K

Из песочницы

+11

68

16

**burenkov**

3 часа назад

Стереокамера машинного зрения с поддержкой ИИ на базе FPGA и Arduino Portenta H7

10 мин 814

Из песочницы

+10

14

0

**mikhailmurzak**

21 час назад

Делаем Телеграм-бота в Cursor AI без знания кода

Простой 5 мин 6.6K

Тutorial

+10

76

15

**subatiq**

2 часа назад

Зарплата как отражение ценностей компании

Простой 9 мин 1.2K

Из песочницы

Перевод

+7

5

10

Разбираем подходы к реализации гиперконвергентной среды

Турбо

Показать еще

ИСТОРИИ



Годнота из блогов компаний



Выравнивания планет



Нейрозима 2025



Статьи с новогодним вайбом



Кто выступит на конференции мечты

КУРСЫ

Онлайн курс по Linux

22 января 2025 · Merion Academy

Курс по этичному хакингу. Белый хакер

22 января 2025 · Merion Academy

L101 Администрирование Linux. Базовый курс

3 февраля 2025 · Hi-TECH Academy

L102 Системное администрирование Linux

10 февраля 2025 · Hi-TECH Academy

KL 013.11.4 Kaspersky Endpoint Security для Linux

13 февраля 2025 · Hi-TECH Academy

[Больше курсов на Хабр Карьере](#)

МИНУТОЧКУ ВНИМАНИЯ**Испытание огнем и ловушками:
спасти мир с помощью лассо****Иди со мной, если хочешь на
перекур: будущее ИИ на заводах****Как хабравчане следят за
здоровьем?**

РАБОТА[Веб дизайнер](#)

16 вакансий

[Все вакансии](#)

БЛИЖАЙШИЕ СОБЫТИЯ



30 января

Зимний тест-драйв Хабра для компаний

Москва

Маркетинг

Другое

[Больше событий в календаре](#)

Хабр



 [Настройка языка](#)

[Техническая поддержка](#)

© 2006–2025, Habr