

 Potok 4 ноября 2014 в 14:03

Восстанавливаем старые часы «Электроника-7»

DIY или Сделай сам

Доброго времени суток, уважаемые хабражители!

Началась эта история так. Во время работы на объекте, разместившимся в здании бывшего завода (кажется, металлоконструкций) с длинным названием (и, конечно же, имени очередного великого лидера партии) я увидел в куче хлама, предназначавшейся на выброс, одну вещь. Каковая вещь ударила по мне страшным приступом ностальгии, ибо точно такая же висела в холле СКБ (с не менее длинным и многосложным названием, чем вышеупомянутый завод), где работала когда-то моя мама, и где прошло немало времени из моего детства. Встречайте — часы «Электроника 7-06».



Конечно же, я не мог устоять от соблазна восстановить (а может быть, и доработать?) их. Кому интересен процесс, а равно и конечный результат — прошу под кат (осторожно, некоторое количество схем и фото!).

1. Немного теории

Схема часов свободно доступна в интернете. Элементной базой является 176 серия микросхем. Индикаторы — газоразрядные типа ИВ-26. Ниже представлен оригинал схемы.

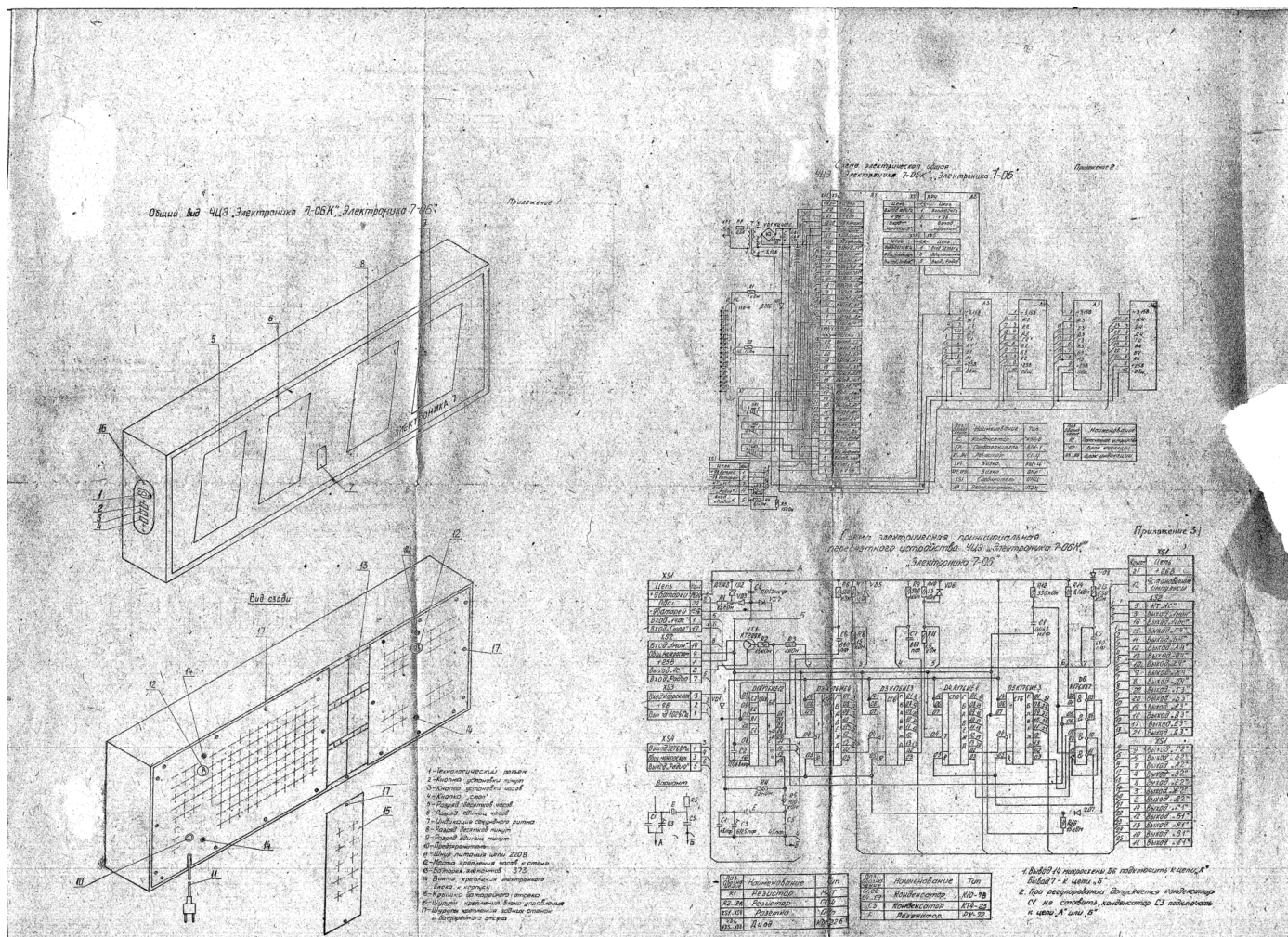


Рис. 1. Оригинальная схема, часть 1

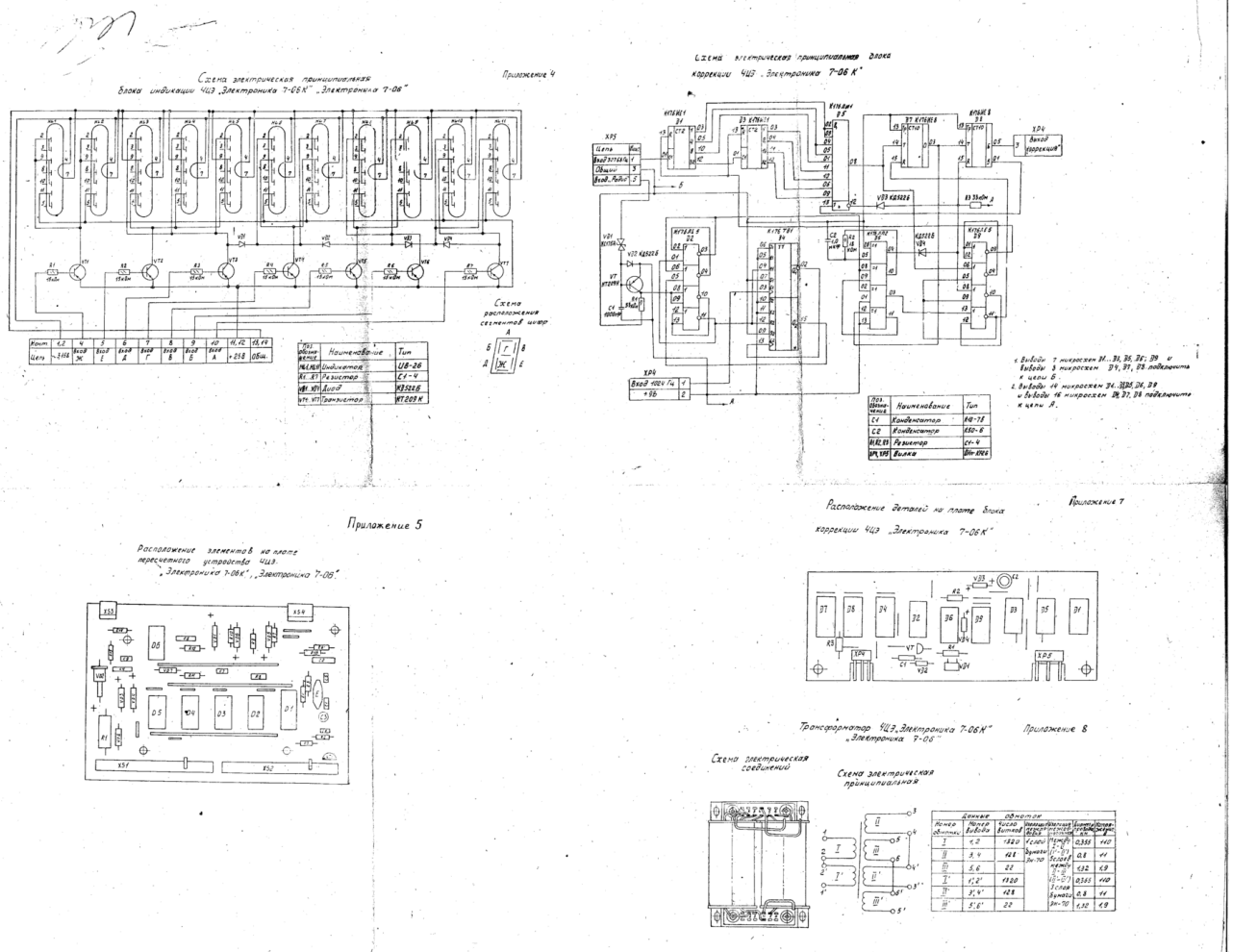
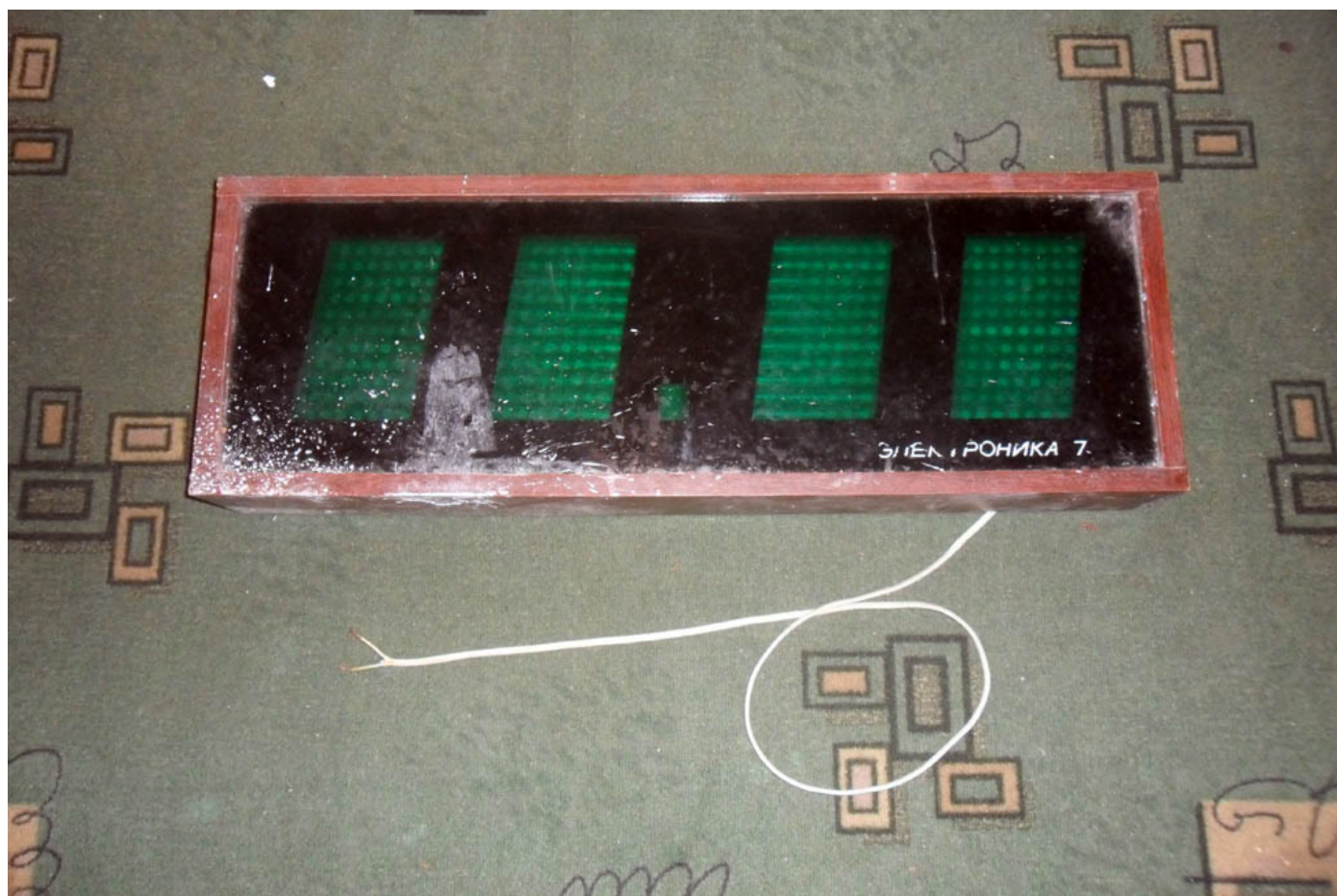


Рис. 2. Оригинальная схема, часть 2

2. Начнем-с

Часы были извлечены из кучи мусора, отвезены домой и препарированы. После очистки от мусора, накопившегося внутри, моим глазам предстало вот что.







Включаем. В принципе, всё работает. Но: индикаторы выгорели. Достать такие же ИВ-26 уже негде. Гугл дает множество ссылок, повествующих нам, как заменить эти ИВ-26 на светодиоды, а то и на готовые семисегментные сборки. Да только вот же незадача — это выглядит уже совсем не так... осовремененно и оттого попсово выглядит, я бы сказал. Поэтому моя задача номер один: восстановить индикаторы на светодиодах, максимально сохранив при этом внешний вид.

3. Табло

При взгляде на расчески из проводов, ведущие к табло, а также на схемы этих табло с сумматорами на диодах, мне становится несколько неуютно. Сложновато в наладке, запросто можно провода перепутать. Да и слабоваты выходы 176-й серии, чтобы напрямую светодиодами управлять. Плюс хотелось бы иметь возможность регулировки яркости индикации, желательно по сценарию тоже — ночью высокая яркость не совсем уместна дома. Стабильность опорного генератора на 25-тилетней давности компонентах мне тоже никто не гарантирует. Прикинув так и этак, я решил видеоизменить схему полностью.

Каждый индикатор — светодиодная матрица 7 x 11, так выходит по количеству точек на оригинальных ИВ-26. Управляет ей хорошо всем известная ATtiny2313. Она же хранит в себе образы символов для отображения, таблицу знакогенератора, говоря иначе. Даже без всяких оптимизаций по 11 байт на символ в нее сотня символов точно влезет — значит, потенциально можно не только цифры писать на табло. И таких матриц у меня будет 4. А что отображать, они пусть получают по UART. Ну а собственно что будет считать время и отправлять по этому интерфейсу данные для табло — это позже. Я подумаю об этом позже (с). Зато к каждой матрице подходит всего 3 провода — GND, +5V и Data. Я посчитал, что однонаправленной линии передачи для данной задачи вполне достаточно.

Индикация динамическая, для выбора строк используется узел на регистрах 74HC595, а для выбора столбца — дешифратор 74HC238. Конструкция AVR + 74HC595 хорошо описана и интереса не представляет. К сожалению, SPI у tiny2313 какой-то порезанный, поэтому загрузка данных в регистры выполняется программно. Плюс к тому при попытке использовать SPI возникли проблемы с разводкой платы, поэтому я отказался от этой идеи. Дешифратор подключен через транзисторную сборку ULN2003 для увеличения мощности.

Изначально я планировал использовать для регулировки яркости светодиодов дополнительный транзистор, управляемый

посредством аппаратного ШИМа на таймере T0, но возникла проблема: ШИМ, накладываясь на динамическую индикацию (их частоты, конечно, не совпадали), порождал неприятное мерцание светодиодов. Поэтому ШИМ программный, а реализован он с помощью дешифратора выбора столбца. Как видно, индикатор-то имеет 7 столбцов, а выходов у дешифратора 8, и последний выход не подключен. Выбирая его, мы гасим всю матрицу.

Ток светодиодов ограничен сопротивлениями. Исходя из документации на примененные LED-5213-PGC-6cd, на них падает 3 — 3,5В при токе 20 мА, примем в среднем 3,2В. Плюс еще 1В падения на ULN2003. Итого $(5 - 3,2 - 1) / 0,02 = 40 \text{ Ом}$. Я взял на 39 Ом.

Переключателями SA1 задается адрес платы. Такой подход позволяет сделать все 4 платы одинаковыми.

К сожалению, металлизацию отверстий в домашних условиях я пока не осилил. Поэтому плата однослойная и количество перемычек на ней может ужаснуть, хоть и сводилось к минимуму всеми усилиями.

Принципиальная схема приведена ниже.

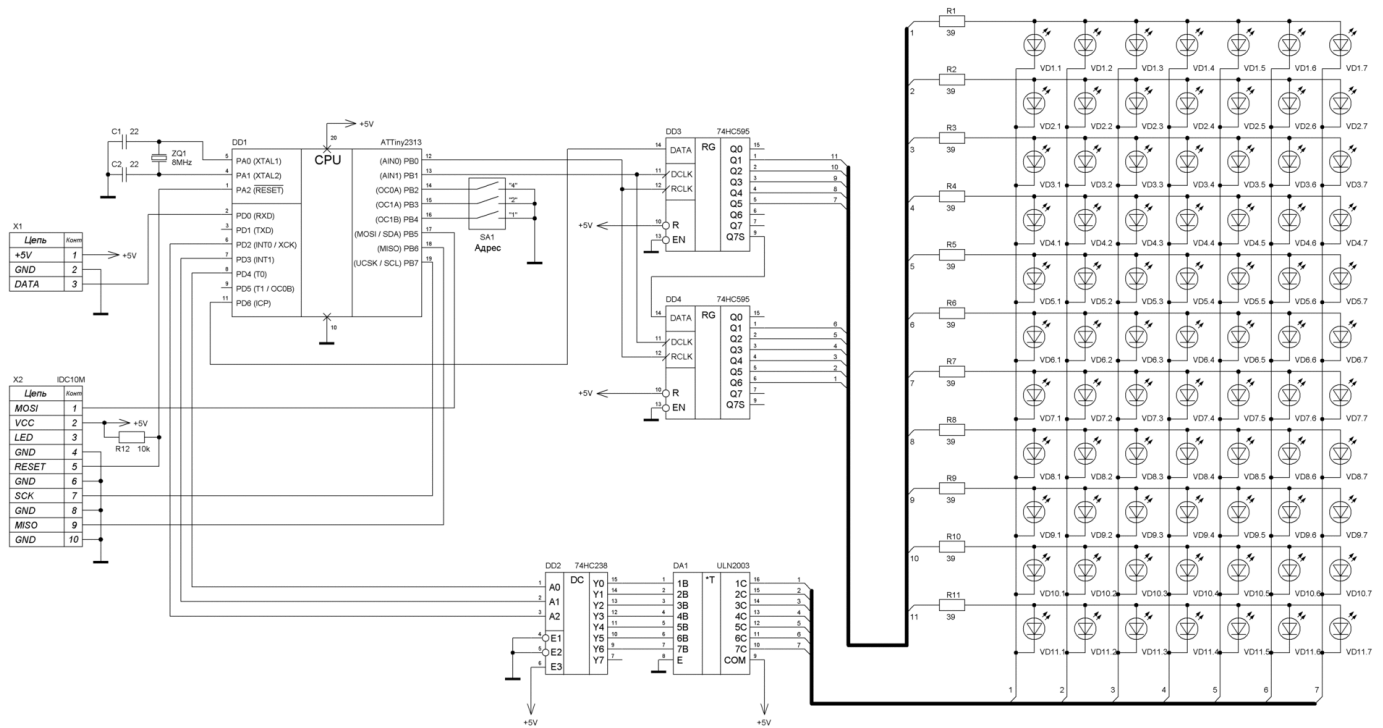
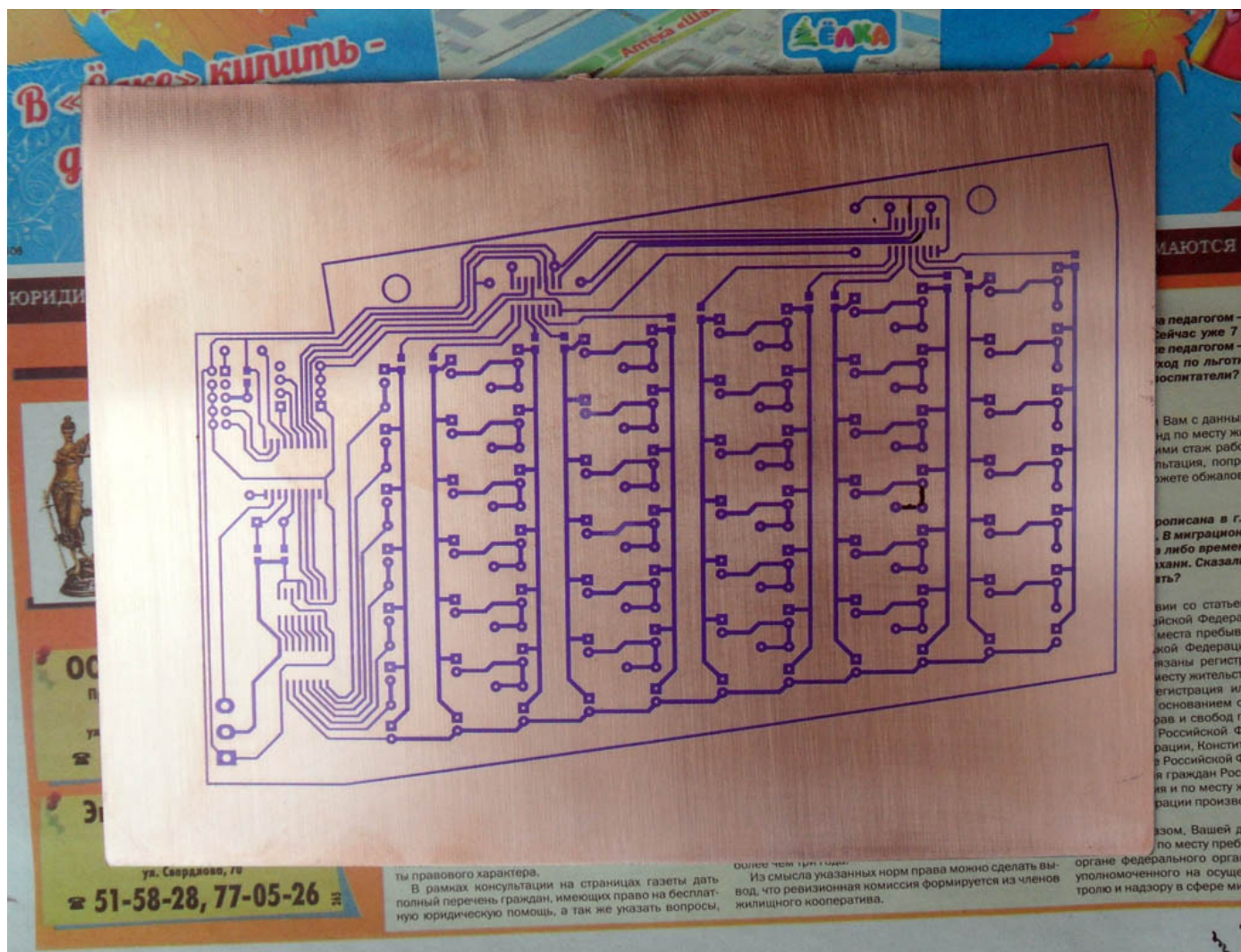


Рис. 3. Принципиальная схема индикатора



А вот фото платы на одном из этапов изготовления (только что нанесен и проявлен фоторезист).

Протокол обмена очень простой:

Первый байт всегда FF, это заголовок пакета.

Второй байт — адрес платы.

Третий байт — данные для отображения, код символа согласно ASCII.

Четвертый — желаемая яркость в диапазоне 00 — FE.

В конце — младшие 8 бит от суммы всех байт пакета, проверка целостности. Если сумма равна FF, заменяем на FE. Пример пакета:

FF 01 32 80 B2 — вывести символ «2» на плате с адресом 1, яркость — половина от максимума.

В процессе написания кода мне также пришла идея сделать так, что плата индикации в момент старта до получения первых данных отображает свой адрес. Оказалось удобно при отладке.

4. Блок питания

Родной блок содержит в себе трансформатор с двумя обмотками: одна выдает 22В, которые применялись для питания анодов индикаторов, и 3,8В для питания их накалов. Конденсаторы, конечно, свою емкость потеряли, к тому же нам потребуются +5В. Значит, схему придется пересмотреть. Кроме того, предусматривается возможность питать логику от 6 батареек по 1,5В, чтобы не сбивалось время при обесточивании. Батарейки — это как-то несерьезно, требуют регулярной замены, так что я переделал этот узел на работу со стандартным аккумулятором 6В, 4.5 А*ч.

Однако ж, $22 * 1,41 = 31В$. Нда, обычной 7805 тут не обойдешься, разве что мы хотим прикрутить сюда еще и функцию комнатного обогревателя. Непродолжительный гуглинг, и на помощь приходит LM2576-5.0 — интегральный импульсный стабилизатор с выходным током до 3А, которая даже нашлась в местном магазине радиодеталей.

Уличную температуру я беру с Рамблера. Выбор обусловлен предпочтениями моей второй половинки.

Итак, поэтапно все действия с «малинкой»:

Вот тут читаем, как подружить ее с Wi-Fi-адаптером TP-Link TL-WN725N.

А вот тут — как установить VNC-сервер, может пригодиться.

Здесь доходчиво расписано, как поднять Samba.

А вот тут, как работать со встроенным UART.

Вот такой скрипт синхронизирует время с мировым с помощью NTP.

[timesync.sh](#)

Этот скрипт читает с Рамблера погоду, складывая принятые данные в файл

[getweather.sh](#)

Основной скрипт, передает данные через UART для отображения:

[send.sh](#)

И — да. Точкой секундной мигаем.

[blink.sh](#)

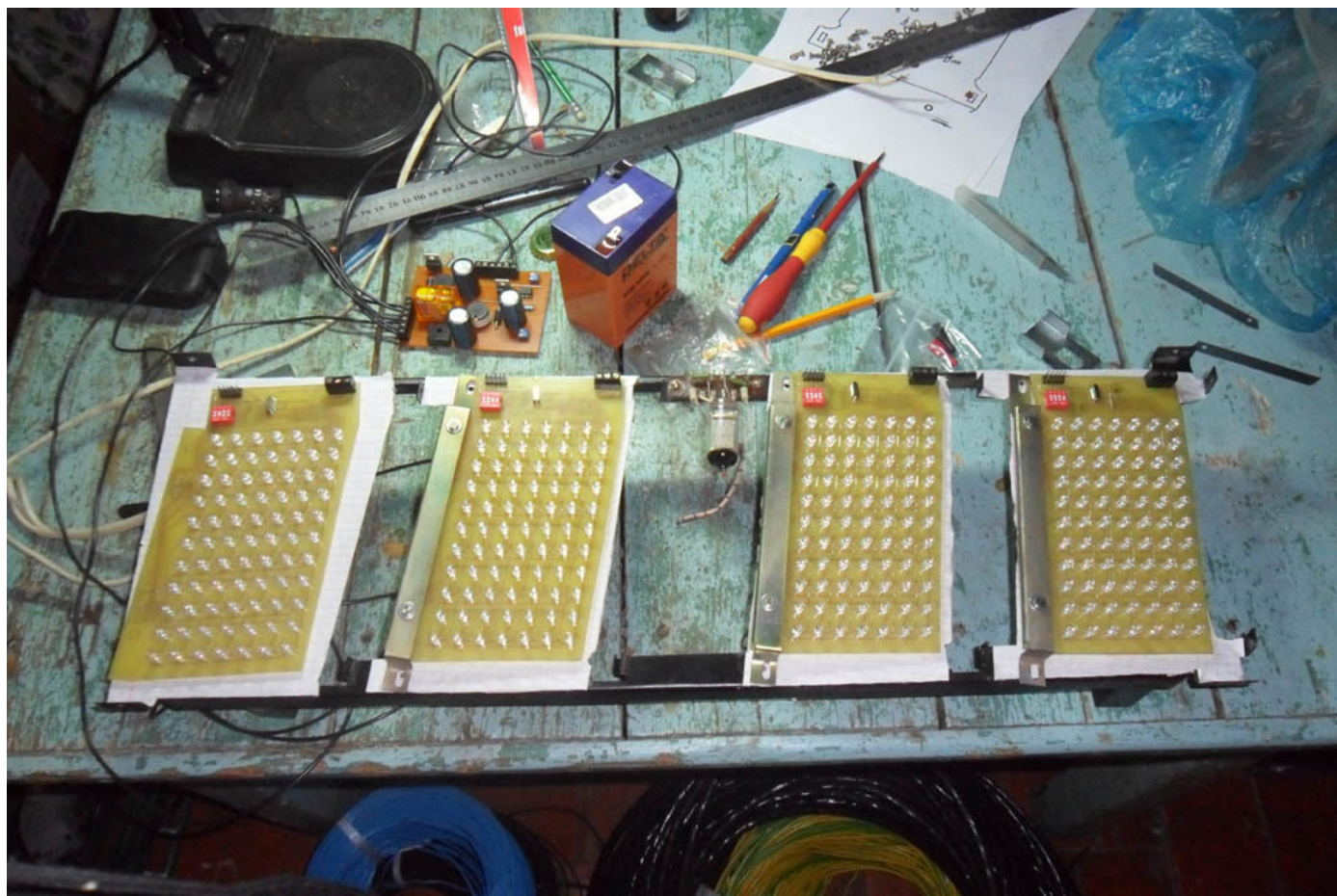
Теперь добавим все это хозяйство в cron:

```
# m h dom mon dow    command
0/15 * * * * /home/pi/clock/timesync.sh
0/15 * * * * /home/pi/clock/getweather.sh
* * * * * sleep 00; /home/pi/clock/send.sh time
* * * * * sleep 10; /home/pi/clock/send.sh weather
* * * * * sleep 15; /home/pi/clock/send.sh time
* * * * * sleep 25; /home/pi/clock/send.sh weather
* * * * * sleep 30; /home/pi/clock/send.sh time
* * * * * sleep 40; /home/pi/clock/send.sh weather
* * * * * sleep 45; /home/pi/clock/send.sh time
* * * * * sleep 55; /home/pi/clock/send.sh weather
```

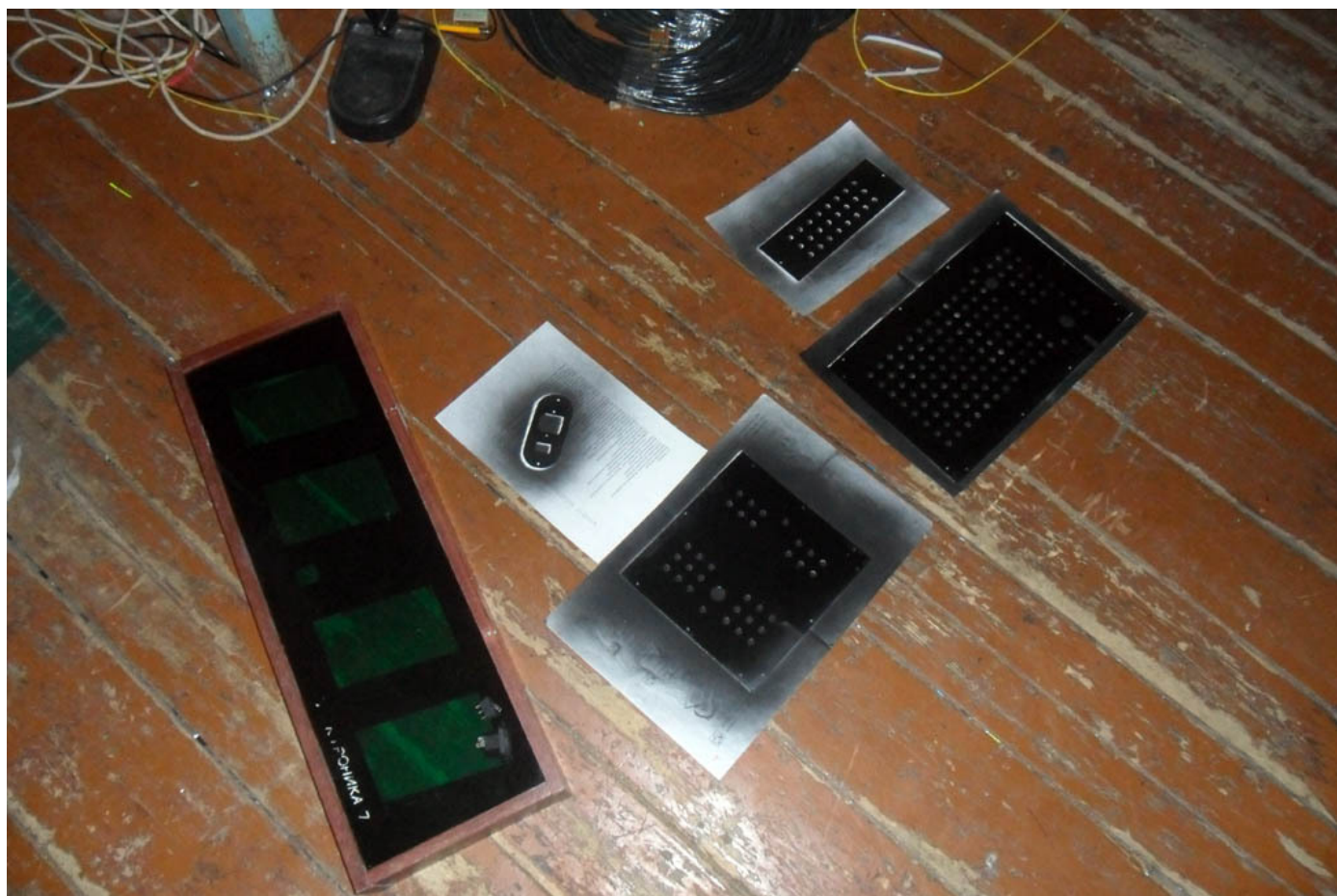
И... и всё. Вешаем на стену, наслаждаемся, ностальгируем. Фото процесса (кликабельно), а также традиционное приветствие жителям Хабра можно увидеть ниже.

Внимание! Автору статьи при рождении вырезали художественное чувство, как будущему инженеру не нужное.

Ценителям незаваленных горизонтов, композиции кадра и всякого прочего баланса белого просьба на этом месте прекратить чтение и перейти сразу к комментариям, во избежание получения серьезных душевных травм.



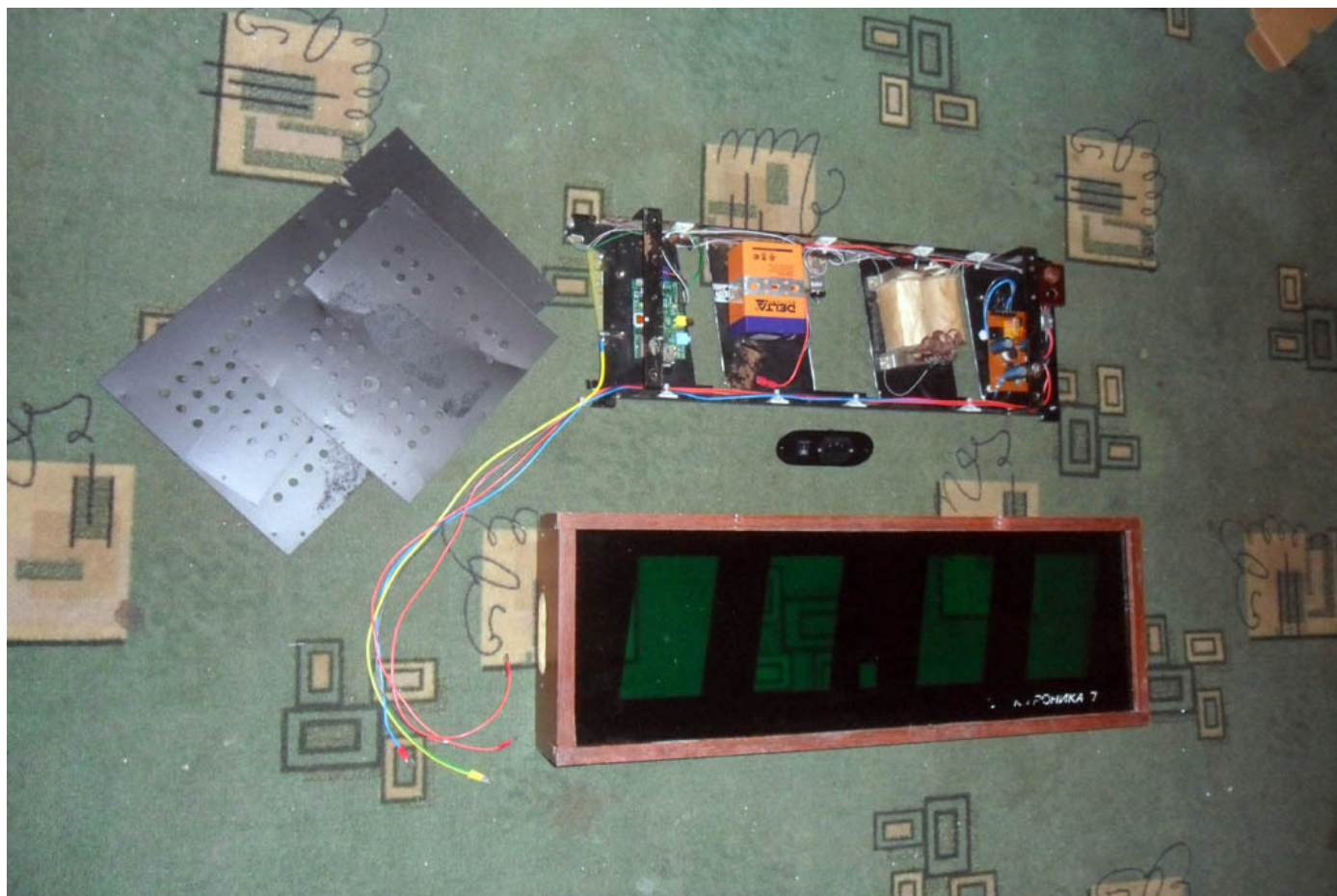
Крепление плат индикации на шасси. Рядом лежит плата блока питания.



Красим проржавевшие задние крышки.



Первое включение в собранном виде. Платы отображают свои адреса.



Все элементы установлены на шасси.



Покрупнее, та же стадия.



Упаковываем в корпус.

И — логическое завершение!



Время.



Температура за бортом.



Все схемы, печатные платы и прошивки можно взять [здесь](#).

Теги: часы для гиков, электроника-7, AVR, raspberry pi, оверинжиниринг, месье знает толк

 **+229**



 **337**

 **122k**

 **85**



117,0

Карма

0,0

Рейтинг

37

Подписчики

0

Подписки

Иванов Георгий Александрович

@Potok

Инженер-программист

ПОХОЖИЕ ПУБЛИКАЦИИ

- 18 августа 2019 в 15:05

Raspberry Pi + CentOS = Wi-Fi Hotspot (или малиновый роутер в красной шляпе)

 **+25**

 **17,1k**

 **145**

 **35**
- 26 июня 2019 в 13:35

Альтернативы Raspberry Pi

 **+47**

 **48,2k**

 **285**

 **158**
- 25 апреля 2019 в 14:29

Создаём датчик контроля качества воздуха на InfluxDB, Grafana, Docker и Raspberry Pi

 **+16**

 **11,6k**

 **135**

 **43**

ВАКАНСИИ		Мой круг
	Backend developer — Vimbox Skyeng · Можно удаленно	от 160 000 ₽
	Front-end разработчик шаблонов для CS-Cart SALEPIX · Можно удаленно	от 50 000 до 100 000 ₽
	Разработчик мобильных приложений для ОС Android IZIBOOK · Москва	от 100 000 до 150 000 ₽
	Senior Golang Engineer Anchous Drive Recruitment · Москва	от 200 000 ₽
	Front-end разработчик для создания симулятора общения с человеком Conventus · Москва · Можно удаленно	от 100 000 до 150 000 ₽
Все вакансии		