

# Разработка контроллера суточного интервального таймера

Разработал  
Руководитель

Галиуллин А.Д.  
Мельников Ю.Ф.

## Актуальность:

На производстве, месторождениях по добыче нефти, газа применяются электронные приборы, которые управляют технологическими процессами, измеряют те или иные параметры процессов. Появлению этих приборов предшествует разработка их. Поэтому очень важно научиться разработке и изготовлению таких приборов. В качестве примера электронного устройства в данной работе поставлена конструкторская задача – разработать схему электрическую принципиальную и изготовить устройство суточного интервального таймера, используя современную элементную базу – интегральные микросхемы и микроконтроллер. Таймеры находят широкое применение, от управления освещением до управления технологическими процессами в промышленности

## Цели и задачи

- Целью данной работы является разработка и макетирование такого суточного интервального таймера на интегральных микросхемах и микроконтроллере, произвести монтаж, проработать конструкцию изделия и выполнить его. Выполнить наладку и регулировку устройства.
- Познакомиться с соответствующей литературой, справочниками, ресурсами интернет по соответствующему направлению.
- Получить навыки работы по разработке схем, их изготовлению и наладке.
- Изучить работу функциональных узлов, применяемых в устройстве, научиться применять их.
- Выполнить наладку и регулировку устройства.

## Разработка контроллера суточного интервального таймера

Техническое задание: Разработать контроллер суточного интервального таймера, выполняющий (имеющий):

- Индикацию текущего времени в 24 часовом формате – часы и минуты;
- Ручную установку точного времени по сигналу путём сброса минут и секунд;
- Восьмиканальное устройство включения нагрузок по программируемому времени начала и конца работы с индикацией состояния этих каналов;
- 5 режимов работы:
  1. Работа, когда индицируется текущее время и есть возможность установки точного времени нажатием на кнопку. Сравнивается текущее время с уставками каналов, и, при совпадении текущего времени с начальным временем срабатывания канала, включается канал с индикацией работающих каналов. При совпадении текущего времени с конечным временем срабатывания, соответствующий канал

выключается;

2. Установка времени, когда имеется возможность установки отдельно часов и отдельно минут в порядке возрастания путём автоматического увеличения чисел с полусекундным интервалом при нажатии на кнопку;

3. Установка номера канала, когда выбирается номер канала для установки времён срабатывания и этот номер индицируется;

4. Установка начального времени срабатывания выбранного канала с индикацией устанавливаемого времени и номера канала;

5. Установка конечного времени работы выбранного канала с индикацией устанавливаемого времени и номера канала;

- Питание от сети 220В 50 Гц;

- Работа часов (без индикации) и хранение уставок времени срабатывания каналов (без срабатывания самих каналов) должно быть энергонезависимым от наличия напряжения сети;

- При восстановлении напряжения сети после его включения контроллер должен возобновлять работу с включением тех каналов, начальное время срабатывания которых меньше текущего, а время конечного времени ещё не настало;

Анализируя техническое задание, выбираем комплектующие изделия:

В качестве микроконтроллера выбираем **AT89C51** фирмы Atmel, имеющий 4 параллельных байтовых порта ввода-вывода P0 – P4, 128 байт ОЗУ, 4 Кбайт ПЗУ для управляющей программы, система команд состоит из 255 команд.

Часы реального времени – **DS1307** - (RTC) отсчитывают секунды, минуты, часы, дату месяца, месяц;

-56-байтовое энергонезависимое ОЗУ с питанием от батареи для хранения пользовательских данных;

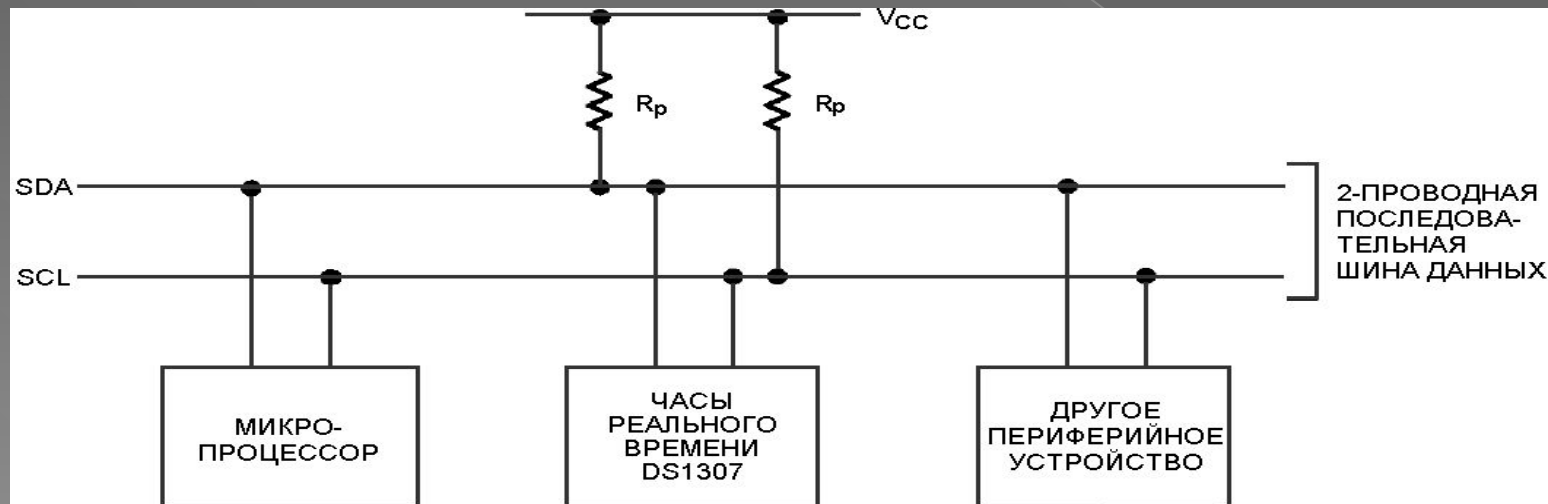
- двухпроводной последовательный интерфейс I2C;

- программируемый выходной сигнал с прямоугольными импульсами (для тактирования внешних устройств);

- автоматическое обнаружение падения напряжения и схема переключения на батарею.

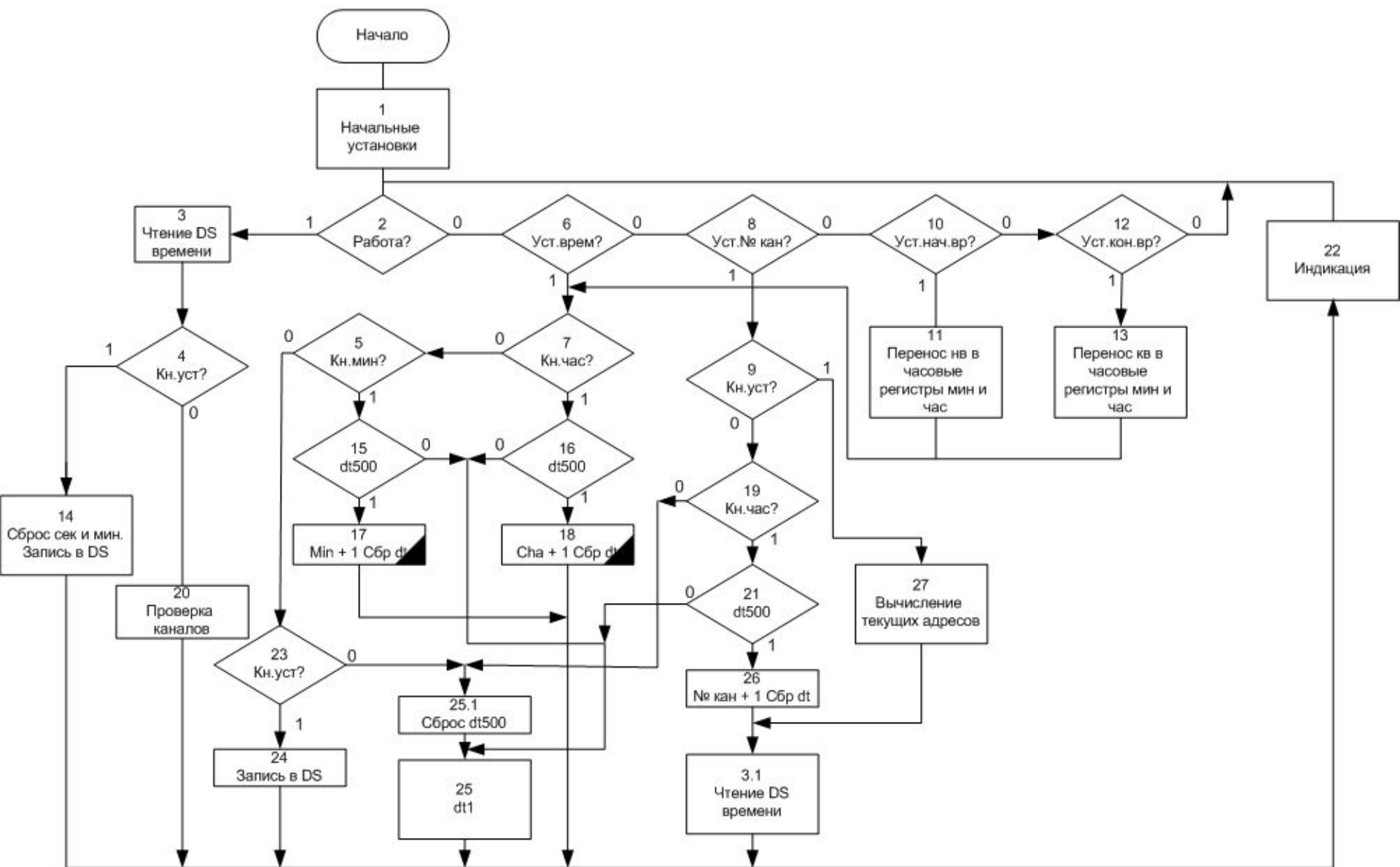
Индикаторы цифровой информации – **BT-N321RD** – трёхразрядные цифровые индикаторы с общим катодом для динамической индикации.

DS1307 поддерживает обмен данными по протоколу I2C по двухпроводной двунаправленной шине. Устройство, которое передаёт данные на шину, является передатчиком, а устройство, принимающее данные, - приёмником. Устройство, управляющее передачей данных, называется ведущим. Устройство, которым управляет ведущий, называется ведомым. Ведущее устройство генерирует синхроимпульсы (serial clock - SCL), управляет доступом к шине и генерирует условия START и STOP. DS1307 работает на шине как ведомое устройство. Типовая конфигурация шины с использованием протокола I2C показана на рисунке.



# Разработка начинается с составления блок-схемы

21.03.12





При включении контроллера сначала выполняются начальные установки, потом проверяется положение переключателя рода работ: если включена работа то происходит считывание информации с RTC, и, если кнопка установки точного времени не нажата, то проверяются каналы на соблюдение условия срабатывания, и, при необходимости включаются в работу. Если кнопка точного времени нажата, то сбрасываются в ноль секунды и минуты, и записываются в RTC.

Если режим установки времени, то проверяются кнопки установки часов и минут, и если какие ни будь из них нажаты, то после полусекундной выдержки изменяются показания, которые после нажатия кнопки установки записываются в RTC.

Аналогично работают и режимы установки начального и конечного времени срабатывания каналов, но при этом дополнительно с временем индицируется ещё и номер канала.

Распределяем ресурсы микроконтроллера:

**P0 – выходы – сегменты индикатора (активный 0):**

P0.0 – a P0.1 – b P0.2 – c P0.3 – d

P0.4 – e P0.5 – f P0.6 – g P0.7 – h

**Биты P2 – выходы катодных ключей (активный 0):**

P2.3 – 1 разряд P2.4 – 2 разряд P2.5 – 3 разряд

P2.6 – 4 разряд P2.7 – 5 разряд

**I2C:** P1.0 – scl для I2C, P1.1 – sda для I2C;

**Режимы работы:**

P1.3 – работа, P1.4 – установка времени, P1.5 – Установка № канала, P1.6 – Установка начального времени, P1.7 – Установка конечного времени.

**DS1307** - P1.2 – выход импульсов 1с.

**Кнопки:** P2.1 – "1" – кнопка установки часов и № канала

P2.2 – "2" – кнопка установки минут, P2.0 – "Д" - Ввод, Enter.

**РПД:**

**Байты:**

41 – минуты (соответствует регистру 01 минут DS1307),

младшая тетрада – биты 3,2,1,0 – единицы минут;  
старшая тетрада – биты 6,5,4 – десятки минут (4 и 5 разряды индикатора).

42 – часы (соответствует регистру 02 часов DS1307),  
младшая тетрада – биты 3,2,1,0 – единицы часов;  
часть старшей тетрады – биты 5,4 – десятки часов при 24 часовом формате (2 и 3 разряды индикатора).

43 – номер канала (1 разряд индикатора).

### Индикатор:

1 разряд	2 разряд	3 разряд	4 разряд	5 разряд
Мл.тетр.43Н	Ст.тетр.42Н	Мл.тетр.42Н	Ст.тетр.41Н	Мл.тетр.41Н

### Сканирование разрядов катодными ключами:

20Н – байт флагов, 27Н – число принимаемых бит (8), 23Н – число принимаемых байт, 21 – байт копия P2 для катодных ключей (с доступом бит). 4А, 4В – текущий адрес конечной и начальной установки времени. 4С – промежуточный байт для сдвига – указатель канала.

4E – байт включения/выключения каналов, 4f – промежуточный для уставок, 70, 71 – текущий адрес уставки в DS начального и конечного времени, 72, 73 – текущий адрес уставки в МК начального и конечного времени,  
76, 77 – счётчики выдержки времени 500 мС.  
7e, 7f – байты для выдержки времени 1 ms.  
50 – 5f, 60 – 6f – байты считанных с DS уставок начального и конечного времени.

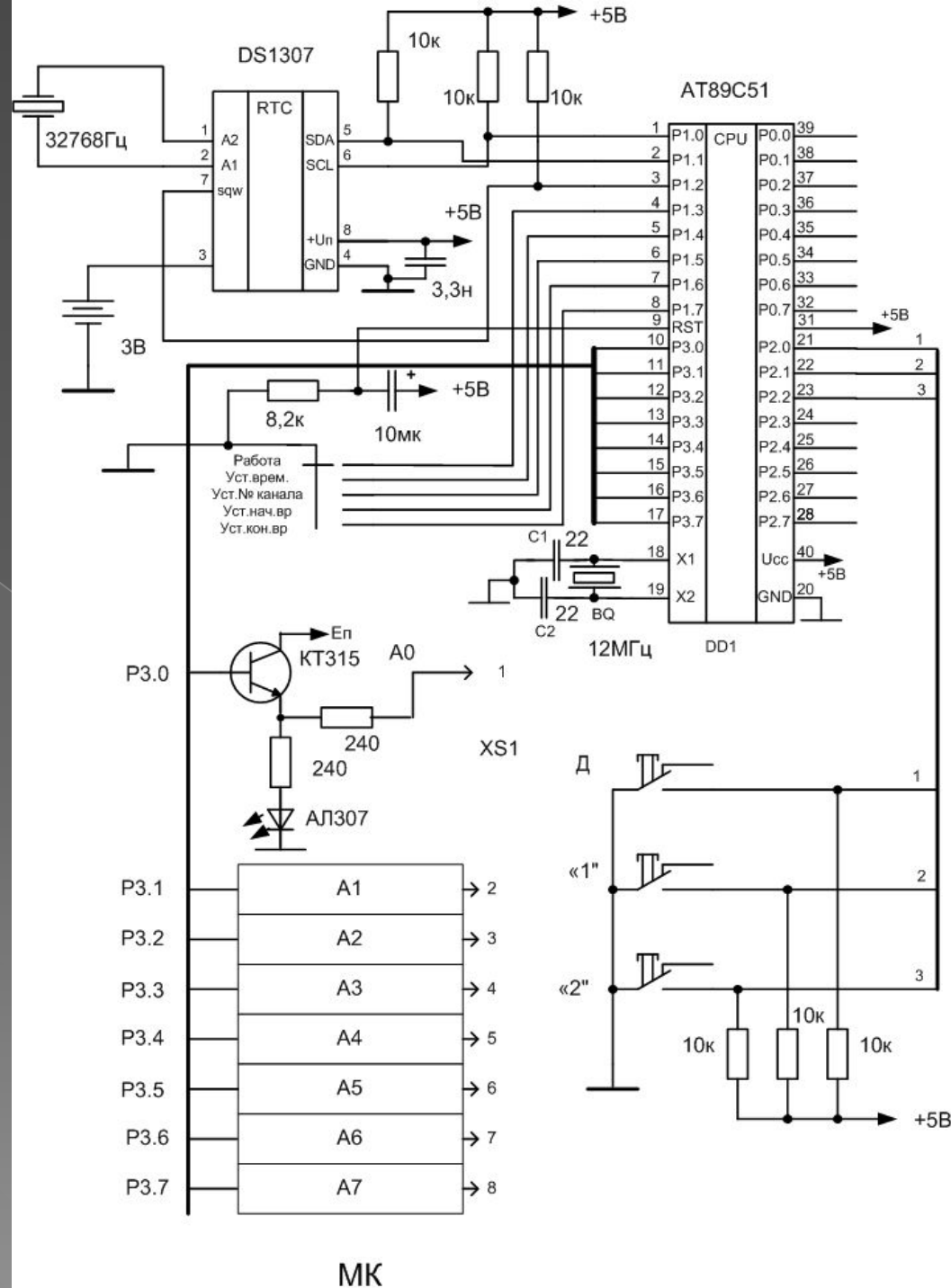
### Биты:

17 – для подпрограммы переключения катодных ключей;

11 – флаг проверки начальной или конечной уставки,  
14 – флаг окончания выдержки 500 мС,  
15 – флаг активности 7 бита аккумулятора при индикации.

Разрабатываем электрическую схему контроллера:

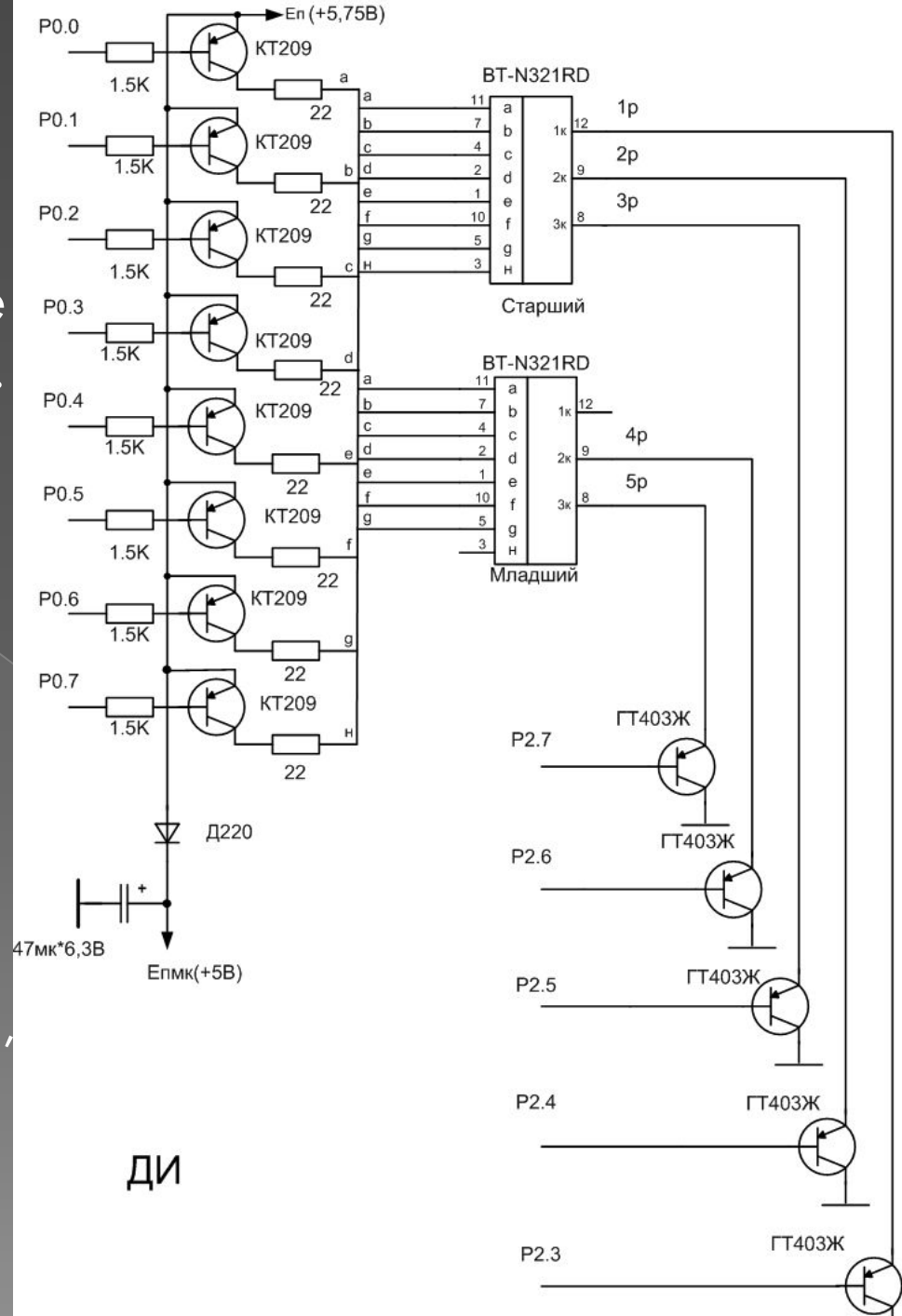
Здесь приведена схема микроконтроллерного узла, состоящая из микроконтроллера, RTC, переключателя рода работ, кнопок, цепи начального сброса микроконтроллера и узлов индикации сработавших каналов на транзисторах КТ315 и светодиодах АЛ307. Шина I2C реализована битами 0 и 1 порта P1. Работа микроконтроллера синхронизируется кварцевым резонатором 12 МГц. К RTC подключена литиевая батарея 3В.



МК

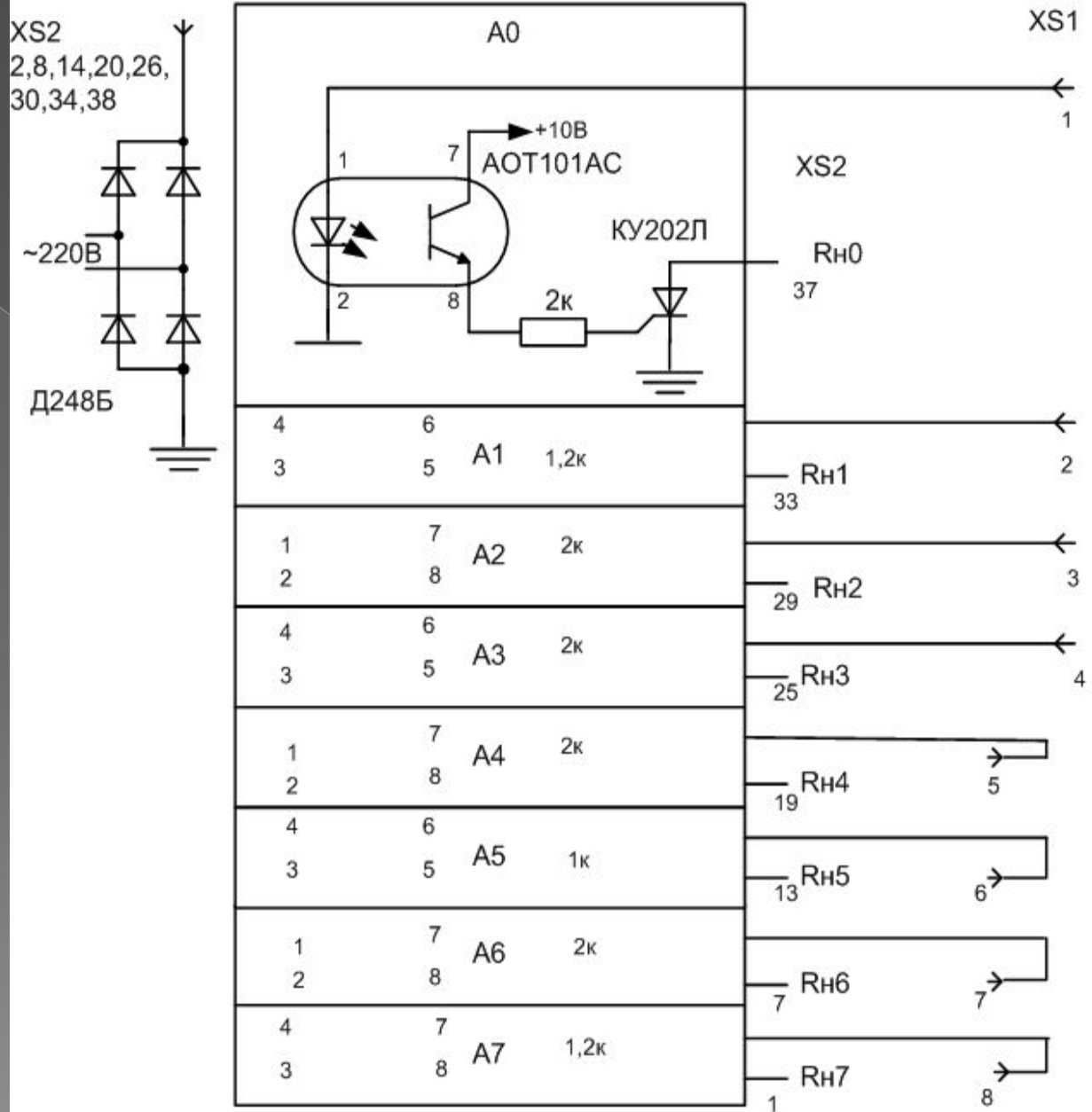
Схема динамической индикации, включает 2 индикатора ВТ-N321RD, сегментные ключи на транзисторах КТ209 и катодные ключи на транзисторах ГТ403Ж.

Это самая энергоёмкая часть контроллера, поэтому она отделена от питания микроконтроллерного узла диодом Д220 и ёмкостным фильтром. На сегментные ключи подаётся код цифры и одновременно включается катодный ключ, и на этом разряде высвечивается цифра, затем то же для других разрядов с периодом 1 мс.



# Схема включения каналов содержит диодный мост на Д248Б,

оптроны АОТ101АС, тиристоры КУ202Л. Так как цепи питания тиристоров не отделены от сети, то они отделены оптронами от микроконтроллерного блока. Сигнал включения канала приходит от схемы индикации включения канала, а оптрон включает соответствующий тиристор, к аноду которого подключена нагрузка.



В соответствии с разработанной блок-схемой, составляем исходный текст управляющей программы на языке ассемблера.

Приводим основную программу: **Определения:**

```
scl    bit    p1.0 ;сигнал scl для i2c
sda    bit    p1.1 ;сигнал sda для i2c
dsw    equ    0D0h ;Адр DS1307 для записи
dsr    equ    0D1h ;Адр DS1307 для чтения
flags  data   20h ;байт флагов
lastread bit  flags.0 ;последний байт читается
ack    bit    flags.5 ;флаг подтверждения приёма
busflt bit    flags.6 ;ошибка линии i2c
_2w_busy bit  flags.7 ;линия i2c занята
bitcnt data   27h ;число принимаемых бит (=8)
bytecnt data  23h ;число принимаемых байт
min    data   41h ;минуты
cha    data   42h ;часы
nk     data   43h ;№ канала
control data  2bh
```



alm\_hour data 2ch

alm\_min data 2dh

fnv bit 16h ;флаг уставки начального времени

fkv bit 10h ;флаг уставки конечного времени

pu data 4fh ;- промежуточная для уставок

vvk data 4eh ;- вкл/выкл каналы

frk bit 11h ;флаг проверки начальной или кон уставки

rvvk data 4ch ;- пром для сдвига

taucn data 4bh ;- тек адр час нач вр

tauck data 4ah; - тек адр час кон вр

sqw bit p1.2 ;импульсы 1Гц DS1307

rab bit p1.3 ;режим работа

uv bit p1.4 ;режим установки времени

unk bit p1.5 ;режим уст № канала

unv bit p1.6 ;режим уст нач времени

ukv bit p1.7 ;режим установки кон вр  
vv bit p2.0 ;кнопка ввод  
vch bit p2.1 ;кнопка уст часов  
vm bit p2.2 ;кнопка уст минут  
fkk bit 17h ;флаг катодных ключей  
kkk data 21h ;копия катодных ключей  
audn data 70h ;текущий адрес уставки в DS нач  
audk data 71h ;текущий адрес уставки в DS кон  
aukn data 72h ;текущий адрес уставки в МК нач  
aukk data 73h ;текущий адрес уставки в МК кон  
dt500\_f data 76h ; счётчик выдержки 500 мС  
dt500\_2 data 77h ; счётчик выдержки 500 мС  
bo500 bit 14h ;флаг окончания выдержки 500 мС

### **;начальные установки**

```
mov kkk,#0fbh ;для катодных ключей  
clr fkk ;очистка флага катодных ключей  
mov dptr,#dgts ; в dptr адрес 7-сегм кодов  
mov nk,#00h ; обнуление № канала
```

```
mov    r0,#1fh ;количество уставок для считывания из DS
mov    r1,#50h ;первичные адреса для считывания из DS
mov    dt500_f,#0ffh ; загрузка счётчика выдержки 500 мС
mov    dt500_2,#02h ; загрузка счётчика выдержки 500 мС
```

```
mov    ru,51h ; начальный адрес уставки часов
mov    rvvk,#01h ; 0 канал
mov    p3,#00h ; выключение всех каналов
```

```
mov    vvk,#00h ;очистка вкл/выкл
mov    taucn,#51h ;адрес часов 0 канала уст нач
mov    tauck,#61h ;адрес часов 0 канала уст кон
```

## **;начало программы**

```
acall set_ds ;нач уст - 1 Гц (1)
acall get_ut ;считывание уставок
```

ss:

```
jb    rab,m6 ;Если не реж раб (2)
```

call sfu ;вызов сброса флагов нач и кон вр уст

acall get\_time ;если реж раб (3)

jb vv,kpk ;Если кн ввод не наж (4)

acall toch\_ust ;если нажата - точ уст врем (14)

sjmp m22 ;переход на индикацию

kpk: acall main ;вызов проверки каналов (20)

jmp m22 ;переход на индикацию

m6:

jb uv,m8 ;если не режим уст вр, то переход (6)

call sfu ;вызов сброса флагов нач и кон вр уст

iz10: jb vch,m5 ;если кн уст час не наж (7)

acall dt500 ;вызов задержки (если реж уст час)(16)

jnb bo500,m25 ;если не окончилась выдержка

acall sbr500 ;если окончилась - сброс

acall ucha ;установка час (18)

m22: acall di ;(22)

jmp ss ;(22)

```
m23:  jb vv,m25sbr ;если кн ввод не наж - переход к выд1 (23)
      call send_start ;(24)
      mov a,#dsw ;(24)
      call send_byte ;(24)
      jb uv,utn ;если не режим устан времени
      mov a,#01h ;(24)
      sjmp pos
utn:   jb unv,utk ;если не режим устан устав нач времени
      mov a,audn ;тек адр уст в DS нач
      sjmp pos
utk:   mov a,audk
pos:   acall send_byte ;(24)
      mov a,min ;(24)
      call send_byte ;(24)
      mov a,cha ;(24)
      call send_byte ;(24)
      call send_stop ;(24)
      jnb fnv,kukt ;если не уст фл уст нач вр
```

```
mov r1,augn ;в r1 тек адр уст в МК нач вр  
sjmp prod
```

```
kukt: jnb fkv,kz ;если не уст фл уст кон вр  
mov r1,aukk ;в r1 тек адр уст в МК кон вр
```

```
prod: mov @r1,min ;мин по адресу  
inc r1 ;+1 - на часы  
mov @r1,cha ;час по адр  
call sfu
```

```
kz: sjmp m22
```

```
m25sbr: acall sbr500 ;25 со сбросом dt500
```

```
m25: acall dt1 ;(25) без сброса  
sjmp m22
```

```
m5: jb vm,m23 ;(5)
```

```
acall dt500 ;(15)
```

```
jnb bo500,m25 ;если вр 500 не законч пер
```

```
acall sbr500 ;сброс dt500
```

```
mov a,min ;(17)
```

```
clr c
inc a      ;(17)
da a      ;(17)
mov min,a  ;(17)
cjne a,#60h,kum ;(17)
mov min,#00 ;(17)
acall ucha ;(17)
kum:  sjmp m22 ;(17)
m8:  jb unk,m10 ; если не уст нач вр
      call sfu ;сбр флагов
      jb vv,rku ;если не наж кн ввода
      mov a,nk ;№ кан в а -вычисл адр по № кан
      clr c ;очистка c
      rlc a ;сдвиг - умн на 2
      add a,#10h ;+10 - след адр уст нач вр в DS(с 10)
      mov audn,a ;получ адр в тек адр уст в DS нач
      add a,#10h ;+10 - след адр уст кон вр в DS(с 20)
      mov audk,a ;получ адр в тек адр уст в DS кон
```

add a,#30h ;+30 - след адр уст нач вр в МК (с 50)

mov aukk,a ;получ адр в тек адр уст в МК нач

add a,#10h ;+10 - след адр уст кон вр в МК (с 60)

mov aukk,a ;получ адр в тек адр уст в МК к он

sjmp ch

pkv: jb vch,ch ;если не наж кн уст час - уст № кан

acall dt500 ;вызов задержки 500

jnb bo500,m25 ;если зад не законч

acall sbr500 ;если законч - сброс зад 500

mov a,nk ;№ кан в a

inc a ;+1

cjne a,#08,kv ;если не 08 (каналы 0-7)

mov a,#00h ;если №=08, то №=0

kv: mov nk,a ;№ в номер канал

ch: call get\_time ;чтение DS

m22o: jmp m22

m10: jb unv,m12 ;если не уст нач вр (10)

clr fkv ;если уст нач вр то сбр фл кон вр



```
jb fnv,dal ;если уст фл нач вр
    setb fnv ;если нет, то устанавливаем
mov r1,aukn ;в r1 адрес тек адр уст нач вр
m10n:mov min,@r1 ;уст кон вр в рег мин
inc r1 ;+1
mov cha,@r1 ;уст кон вр в рег час
dal: jmp iz10
m12:  jb ukv,m22o ;если не режим уст кон вр (12)
    clr fnv ;если уст кон вр то сбр фл нач вр
jb fkv,dal ;если уст фл кон вр
setb fkv ;если уст фл кон вр
mov r1,aukk ;в r1 адрес тек адр уст кон вр
jmp m10n
```

***В приведённой основной части программы указаны подробные комментарии для каждой команды после точки с запятой. В скобках указаны номера блоков по блок-схеме. Для сокращения объёма программы не приведены тексты подпрограмм.***

После составления исходного текста программы она подвергается трансляции. В данном случае был применён MCStudio. В результате получается файл в формате фирмы Intel с расширением hex.

Далее следует отладка программы с применением внутрисхемного эмулятора WICE 8052. Эмуляционная головка эмулятора вставляется вместо микроконтроллера в отлаживаемое устройство. Из ПК в эмулятор загружается управляющая программа.

Этот процесс показан на фото 1 и 2.

При нахождении ошибок, они исправляются в исходнике, затем трансляция и снова загрузка в эмулятор. Этот итерационный процесс заканчивается получением правильно работающей программы, которая загружается в микроконтроллер с помощью программатора. После этого микроконтроллер вставляется в разрабатываемое устройство, и, после проверки, контроллер готов к работе (фото 3 и 4).

# Фото 1

Управляющий ПК

Отлаживаемый контроллер

Источник питания

Внутрисхемный эмулятор



# ФОТО 2

Эмуляционн  
ая головка

Сегментн  
ые ключи

Катодные  
ключи

Цифровые  
индикатор  
ы

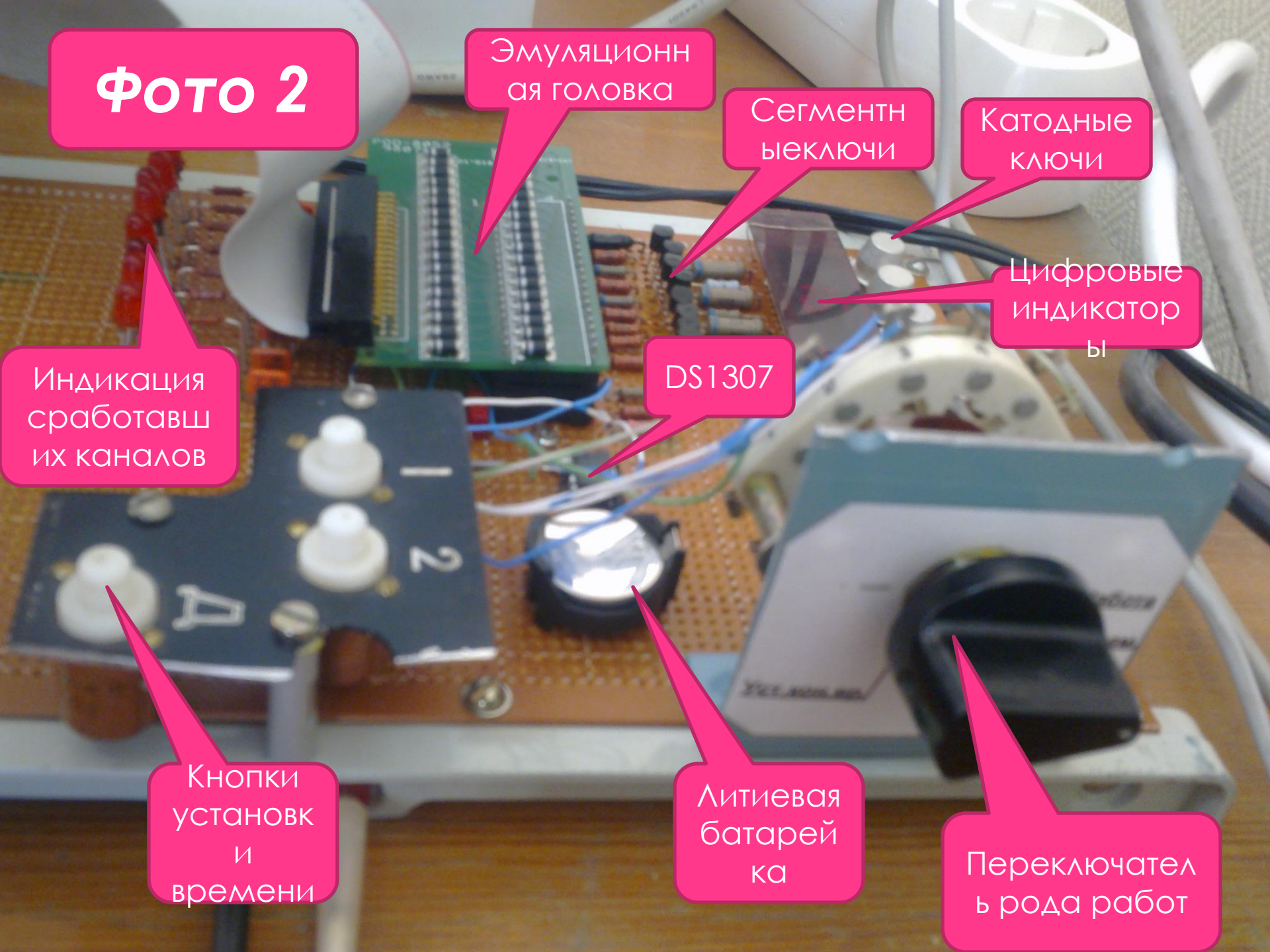
DS1307

Индикация  
сработавш  
их каналов

Кнопки  
установк  
и  
времени

Литиевая  
батарея  
ка

Переключател  
ь рода работ



# ФОТО 3

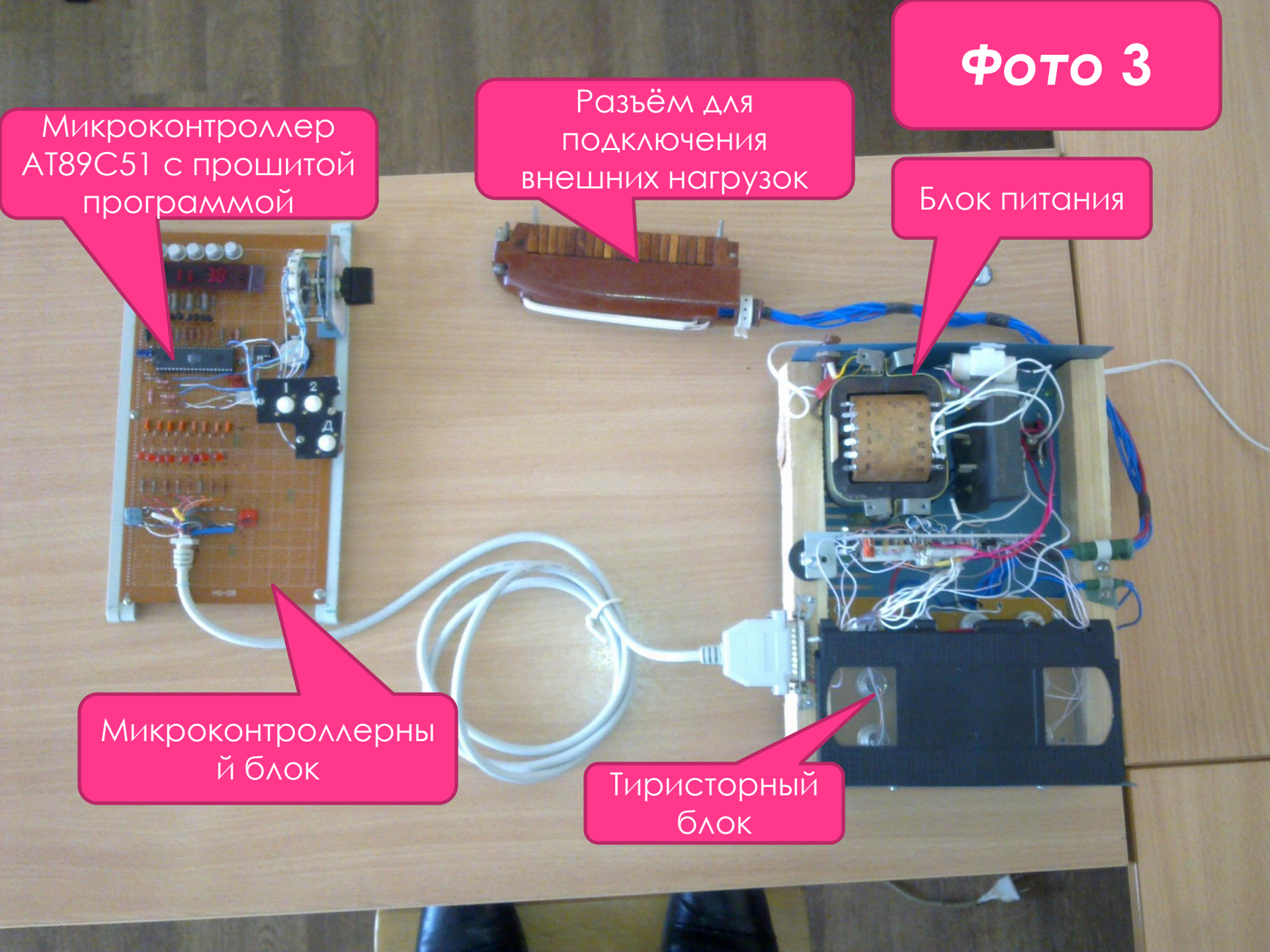
Микроконтроллер  
AT89C51 с прошитой  
программой

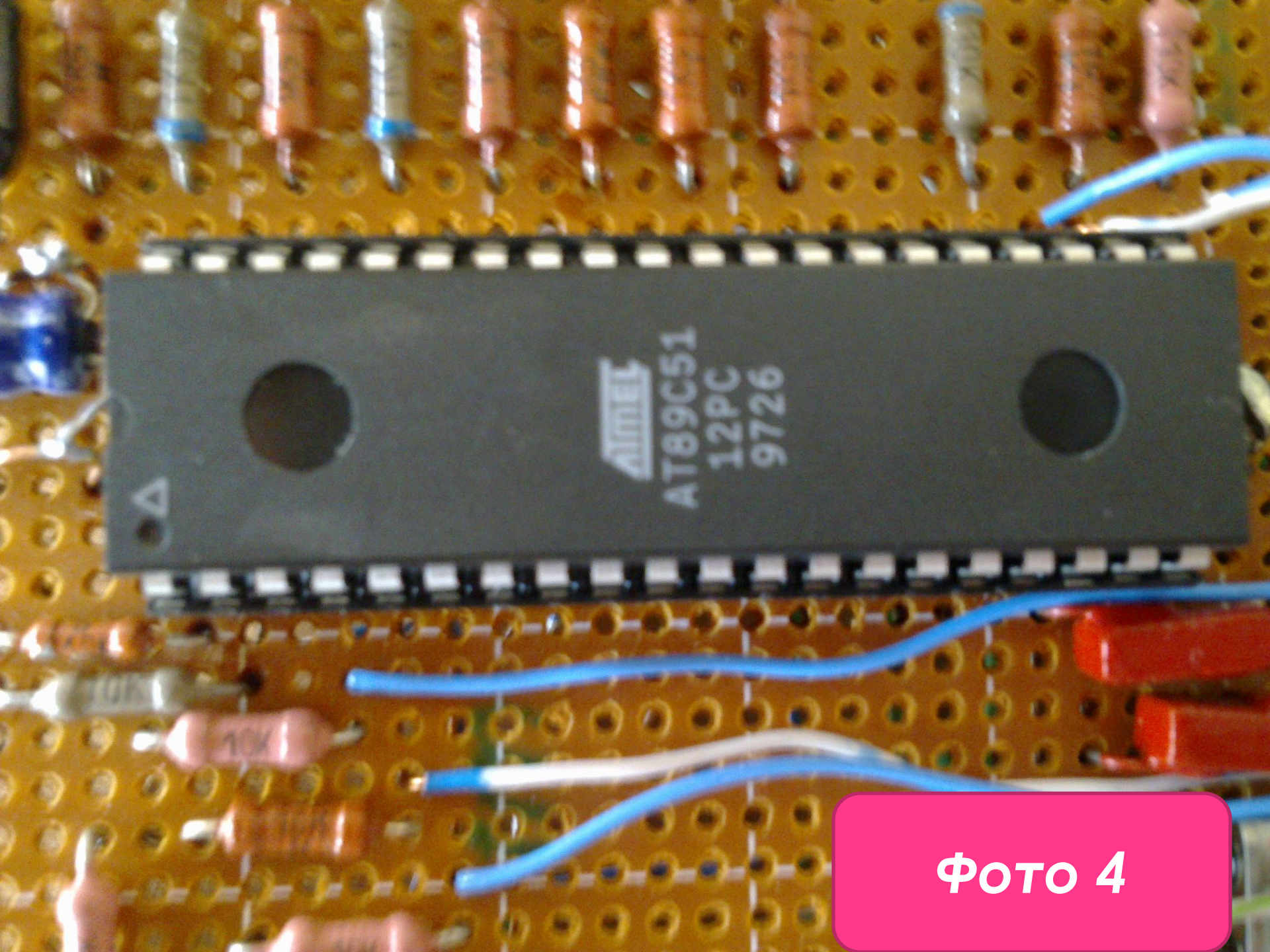
Разъём для  
подключения  
внешних нагрузок

Блок питания

Микроконтроллерны  
й блок

Тиристорный  
блок





ATMEL  
AT89C51  
12PC  
9726

ФОТО 4

## Вывод:

В данной работе разработан и выполнен макет контроллера интервального суточного таймера в соответствии с техническим заданием. Он полностью соответствует заданию, следовательно, поставленная цель достигнута.

В качестве развития данного таймера можно предложить увеличение числа каналов, а также расширения видов питания и номенклатуры подключаемых нагрузок, - например, выход с помощью реле, биполярных транзисторов с изолированным затвором.