

## Блок питания лабораторный

**Автор: А.В. Жмаев, E-mail: [kuratorj@yandex.ru](mailto:kuratorj@yandex.ru)**

### Введение

Данный материал предназначен для студентов и всех тех, кто увлекается разработкой электронного оборудования. Он может быть, в определенной мере, полезен и разработчикам - профессионалам.

Проще всего осваивать учебный материал, имея под рукой конкретный пример.

Идея сделать лабораторный блок питания (БП) пришла, когда мне надоела нагромождение стандартных БП на рабочем столе, особенно, когда работаешь с небольшими макетами (что часто бывает), разнообразной элементной базой и одновременного включения питания – разной полярности и напряжения.

Для управления применены микроконтроллеры STM32 и ATMEGA.

Параметры БП установил на основе своего опыта.

Дополнительная литература:

1. ATmega328\_P\_rus - <https://disk.yandex.ru/i/FI8J2Xvd5yB02g>
2. AVR система команд - <https://disk.yandex.ru/i/duXKCGks03VAnQ>
3. ARM Assembly Language Fundamentals and Techniques (Rus) <https://disk.yandex.ru/i/FE2giDCHTxX3Vw>
4. 56\_STM32F10X\_RM000\_Rus - <https://disk.yandex.ru/i/RawIXyWDNNfEsg>
5. PM0056 Programming manual\_Rus - <https://disk.yandex.ru/i/8ERjPe09Ps3JOA>
6. Прибор Генератор\_Измеритель - [https://disk.yandex.ru/i/61op\\_Dz25PWYkA](https://disk.yandex.ru/i/61op_Dz25PWYkA)
7. Аппарат сварочный САБП - <https://disk.yandex.ru/i/DSN21ss7YVzVZQ>

## Технические данные и описание

БП (Рис.1) фактически представляет собой два разделенных, электрически, блока питания.

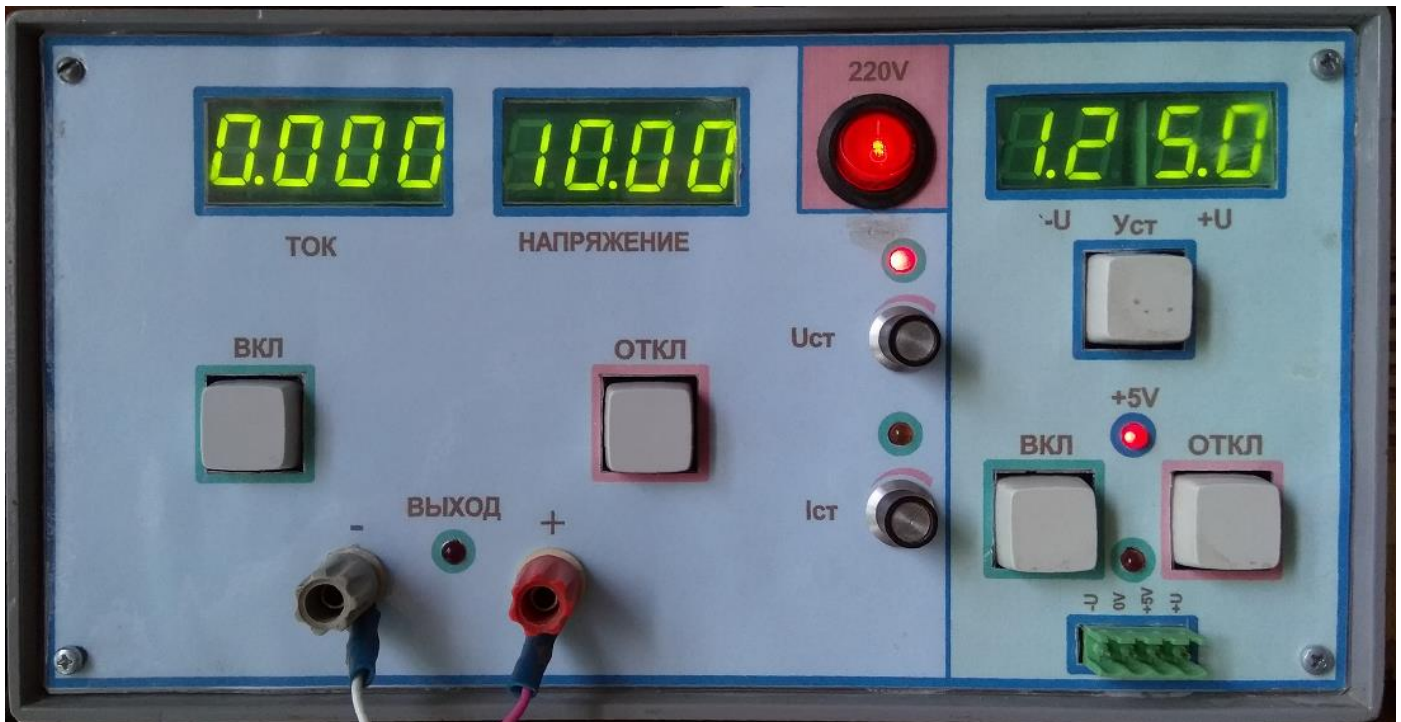


Рис. 1. Блок питания лабораторный

1. Мощный БП со стабилизацией:
  - Напряжение «Uст»: 0,5 ... 50 В;
  - Ток «Iст»: 0,01 ... 4 А.
  - Включение «ВЫХОД» производится путём нажатия и отпускания кнопки «ВКЛ».
  - Отключение «ВЫХОД» производится путём нажатия кнопки «ОТКЛ».
  - Индикация фактического тока и напряжения производится на индикаторах «ТОК» и «НАПРЯЖЕНИЕ».
  - Индикация заданного тока и напряжения производится на индикаторах «ТОК» и «НАПРЯЖЕНИЕ» при нажатой одной из кнопок «ВКЛ» или «ОТКЛ».
  - Установка стабилизации напряжения и тока производится с помощью резисторов «Uст» и «Iст» (контроль заданного тока и напряжения производится при нажатой одной из кнопок «ВКЛ» или «ОТКЛ»).
  - Срабатывание включения «ВЫХОД», стабилизация «Uст» или «Iст» - индицируется светодиодными индикаторами.
2. Маломощный 3-х канальный БП со стабилизацией:
  - каналы: «-U», «+U», «+5V» – Напряжение выбирается путем нажатия кнопки «Uст», путем последовательного перебора в соответствии с таблицей 1.

Красным цветом показаны параметры - По умолчанию, при включении сетевого питания «220V».

Таблица1 Выходные напряжения, В.

| Уст | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7    | 8    | 9    | 10   | 11   |
|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| +U  | 2,5 | 3,3 | 5,0 | 5,0 | 5,0 | 6,0 | 9,0 | 12,0 | 12,0 | 12,0 | 15,0 | 15,0 |
| -U  | 2,5 | 3,3 | 1,2 | 5,2 | 6,0 | 6,0 | 9,0 | 1,5  | 6,0  | 12,0 | 1,5  | 15,0 |
| +5V | 5,0 |     |     |     |     |     |     |      |      |      |      |      |

- При любом нажатии кнопки «Уст» отключаются выходы каналов.
- Защита каналов по току установлена примерно: 0,7 А.
- При срабатывании защиты одного из каналов, высвечиваются прочерки на индикаторах «+U» или «-U», а на канале «+5V» - гаснет индикатор «+5V», соответственно и отключаются все выходы каналов.
- Сброс защиты производится путём нажатия кнопки «ОТКЛ».
- Включение выходов производится путём нажатия и отпускания кнопки «ВКЛ».
- Отключение выходов производится путём нажатия кнопки «ОТКЛ».
- При включении сети «220V» блокируются выходы и для разблокировки их необходимо нажать кнопку «ОТКЛ».

## Описание конструкции

На Рис.1 показана передняя панель БП.

Для удобства работы органы управления и индикации сделаны максимально большими, что позволяет делать меньше ошибок от усталости при длительной работе с БП.

Установка напряжения и тока производятся многооборотными резисторами R4, R8 типа ППМФ-ИМ (Схема на Рис.2). Характеристики резисторов R4, R8 отредактированы резисторами R1, R2, R3 и R5, R6, R7 в виде ломаной кривой, близкой к параболе – соответственно.

Системы стабилизации старался сделать аналоговыми, хотя КПД у них меньше, но и шумов, и импульсных помех практически нет. Для **лабораторного** БП очень важно.

В 3-х канальном стабилизаторе (Схема на Рис.3):

- каналы «-U», «+U» - аналоговые;
- канал «+5V» - импульсный с частотой 850 кГц (Схема на Рис.11), что для питания цифровых схем допустимо. Для перестраховки можно поставить аналоговый стабилизатор, но придется ставить дополнительный радиатор.

**Примечание:** Расчет площади радиаторов (без вентилятора) ведется из параметра не менее 20 см<sup>2</sup>/Вт.

**Управление** 3-х канальным стабилизатором производится микропроцессором (Схема на Рис.4)

Мощный канал стабилизатора построен по 2-х каскадной схеме (Схема на Рис.2):

- первый каскад - импульсный стабилизатор (Схема на Рис.5) со стабилизацией напряжения 4 ... 54 В.
- второй каскад – аналоговый стабилизатор (Схема на Рис.6) со стабилизацией напряжения 0,5 ... 50 В и тока 0,01 ... 4 А.

**Управление** мощным стабилизатором производится микропроцессором (Схемы на Рис.7 ... Рис.10) и с отслеживанием, чтобы напряжение на втором каскаде падало в пределах 3 ... 5 В.

2-х каскадная схема позволяет уменьшить размеры радиаторов и сильно улучшить качество стабильного напряжения.

Индуктивность L1(Схема на Рис.2) необходима для увеличения времени нарастания тока, только в момент короткого замыкания, чтобы система управления успела отработать.

## Схемы

Автор: А.В. Жмаев, E-mail: [kuratorj@yandex.ru](mailto:kuratorj@yandex.ru)

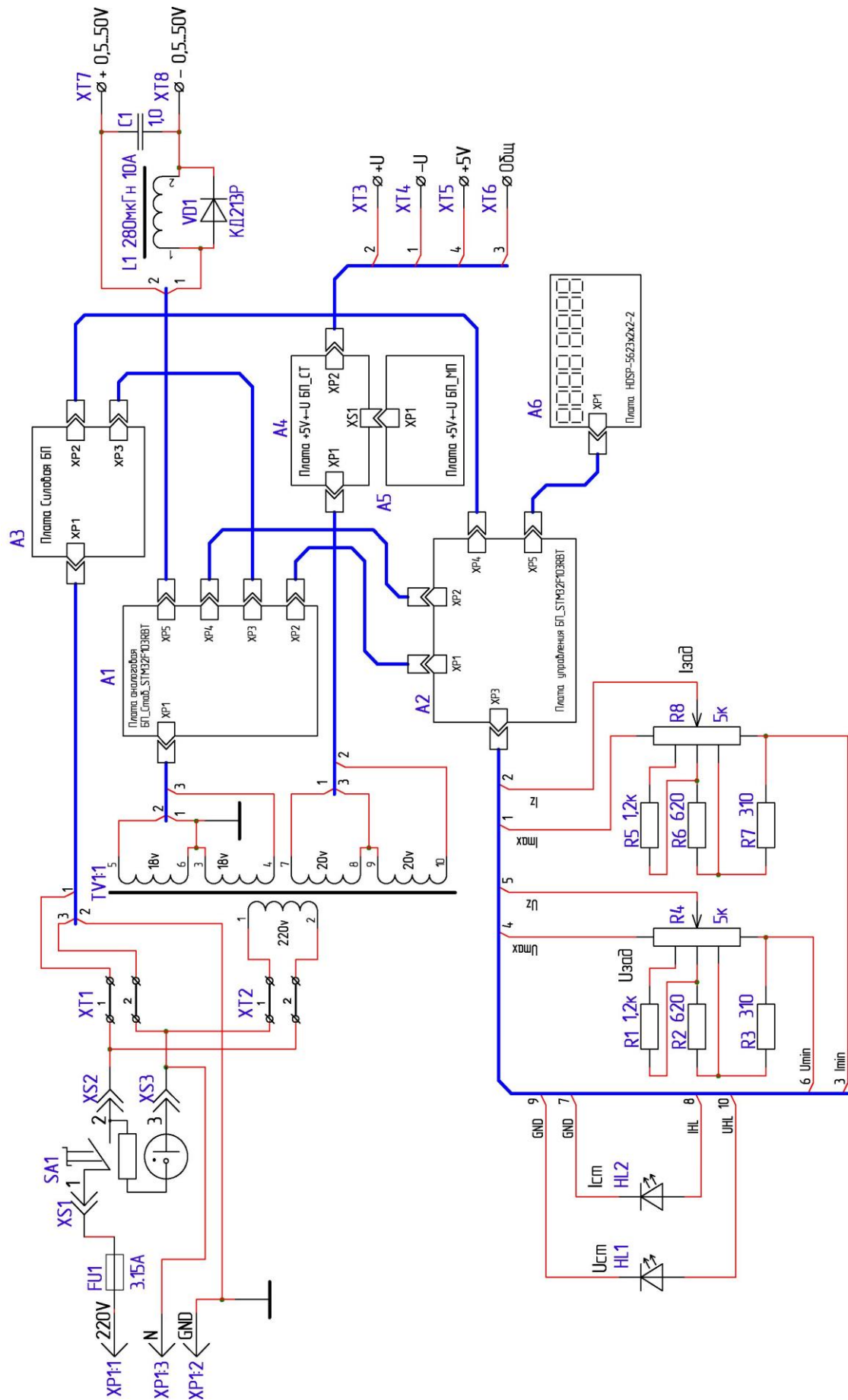


Рис.2. Схема общая БП\_STM32F103RBT

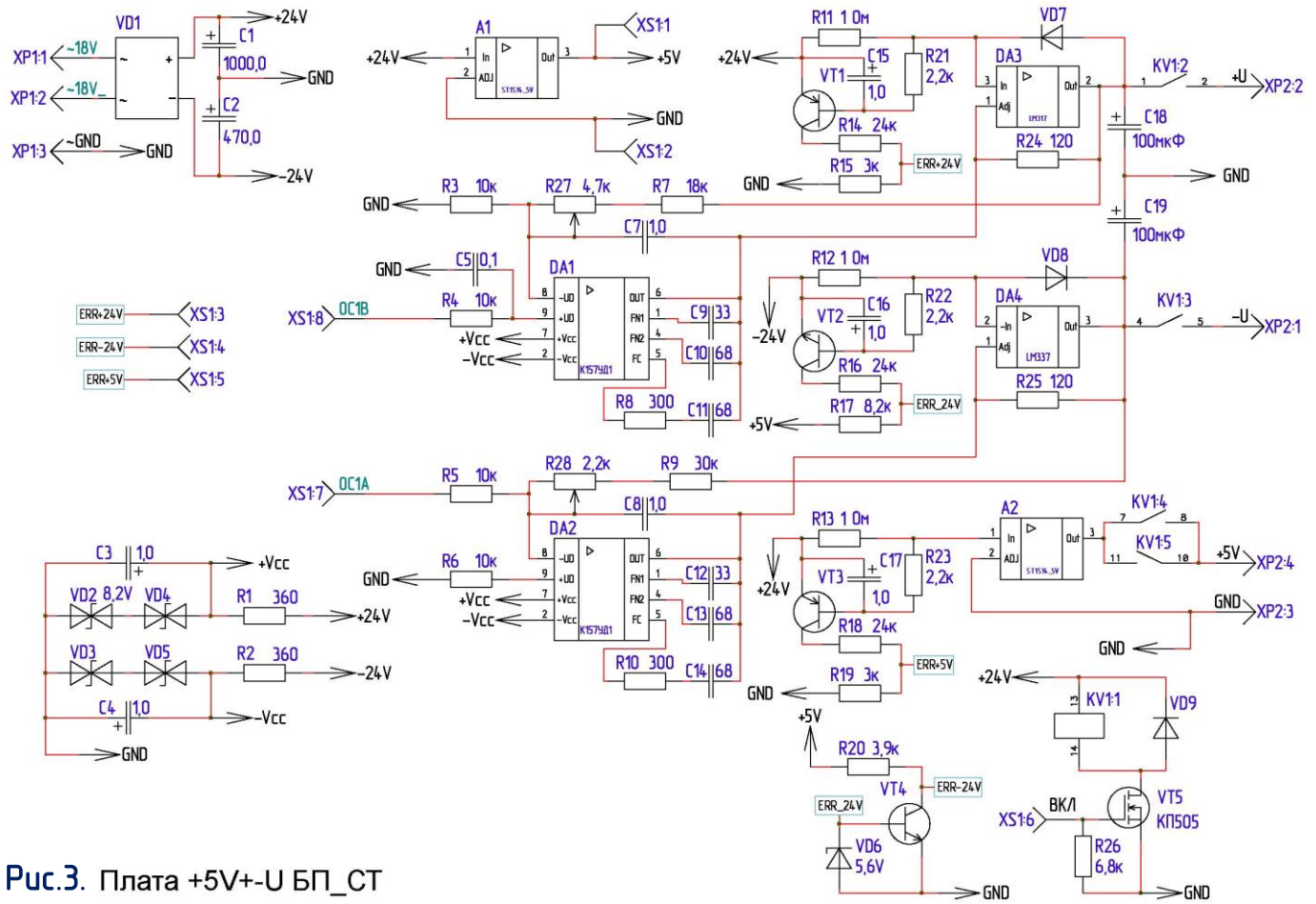


Рис.3. Плата +5V+-U БП\_СТ

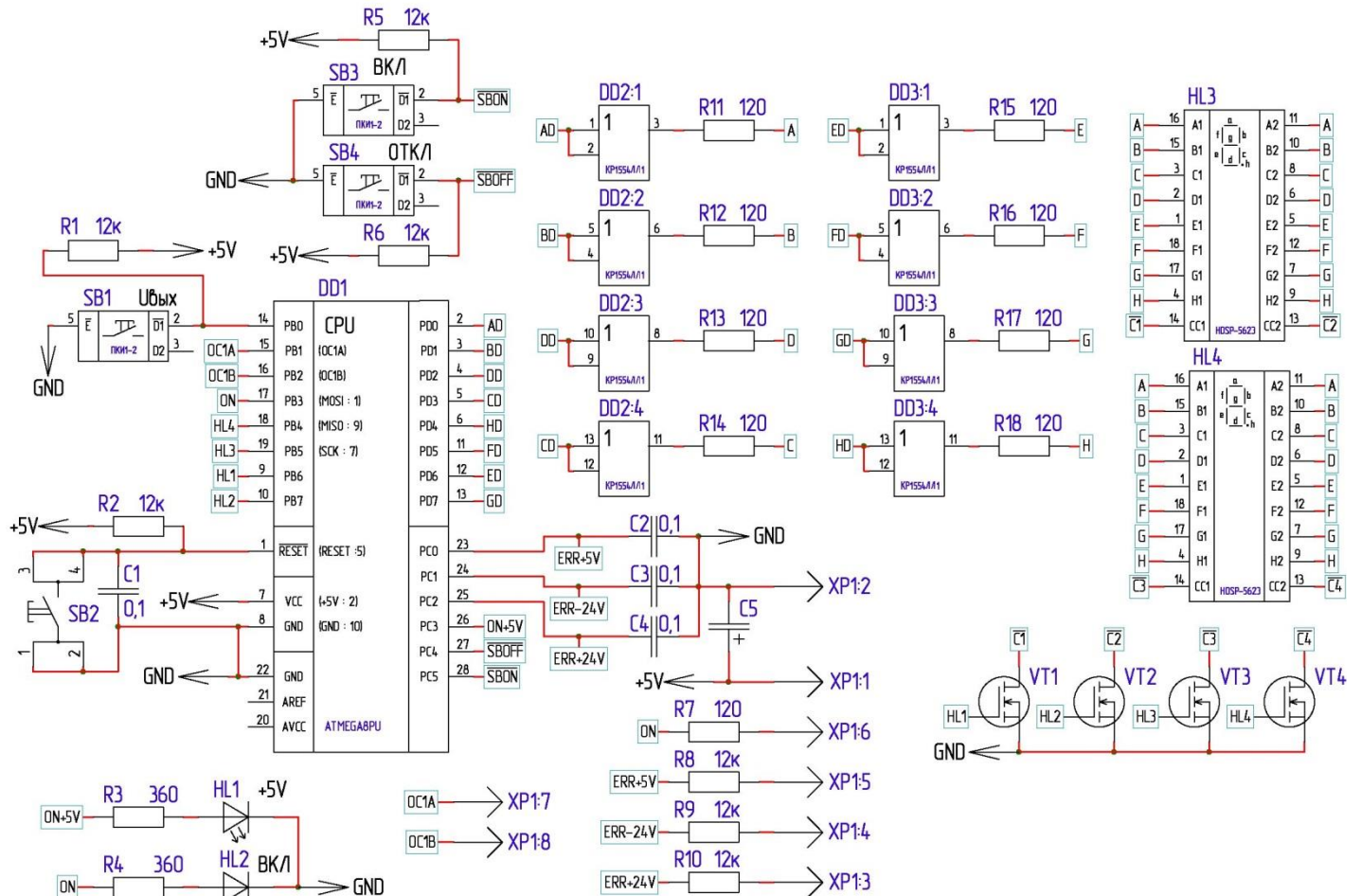


Рис.4. Плата +5V+-U БП\_МП

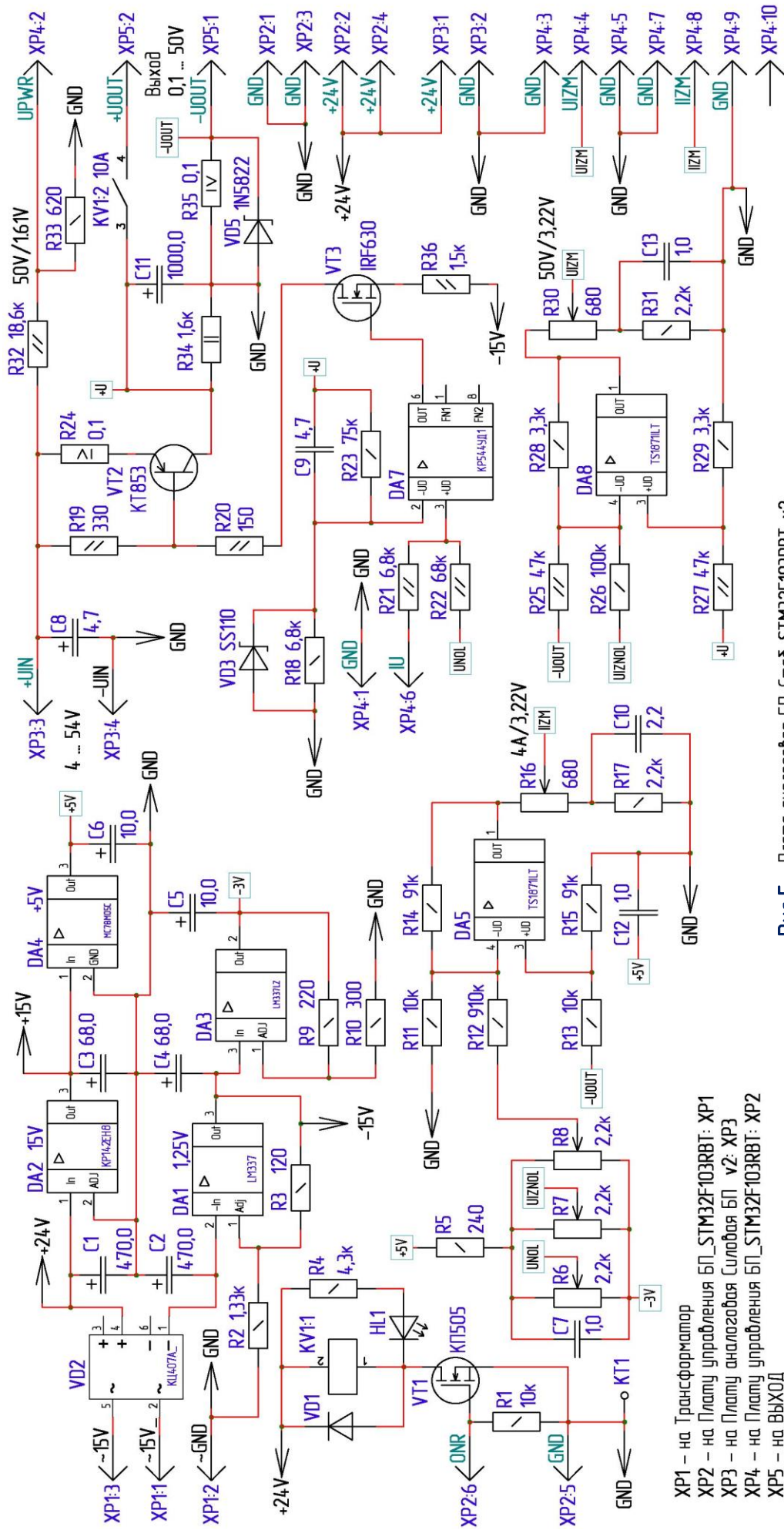


Рис.5. Плата аналоговая БП\_Смад\_STM32F103RBT\_v2



XP1 – на Плату аналоговую БП\_Смад\_STM32F103RBT: XP2  
XP2 – на Плату аналоговую БП\_Смад\_STM32F103RBT: XP4  
XP3 – на резисторы задания U<sub>1</sub>  
XP4 – на Плату аналоговую Силовая  
XP5 – на Плату индикаторов: XP1

**Рис.8.** Плата управления БП STM32F103RBT Лисм1

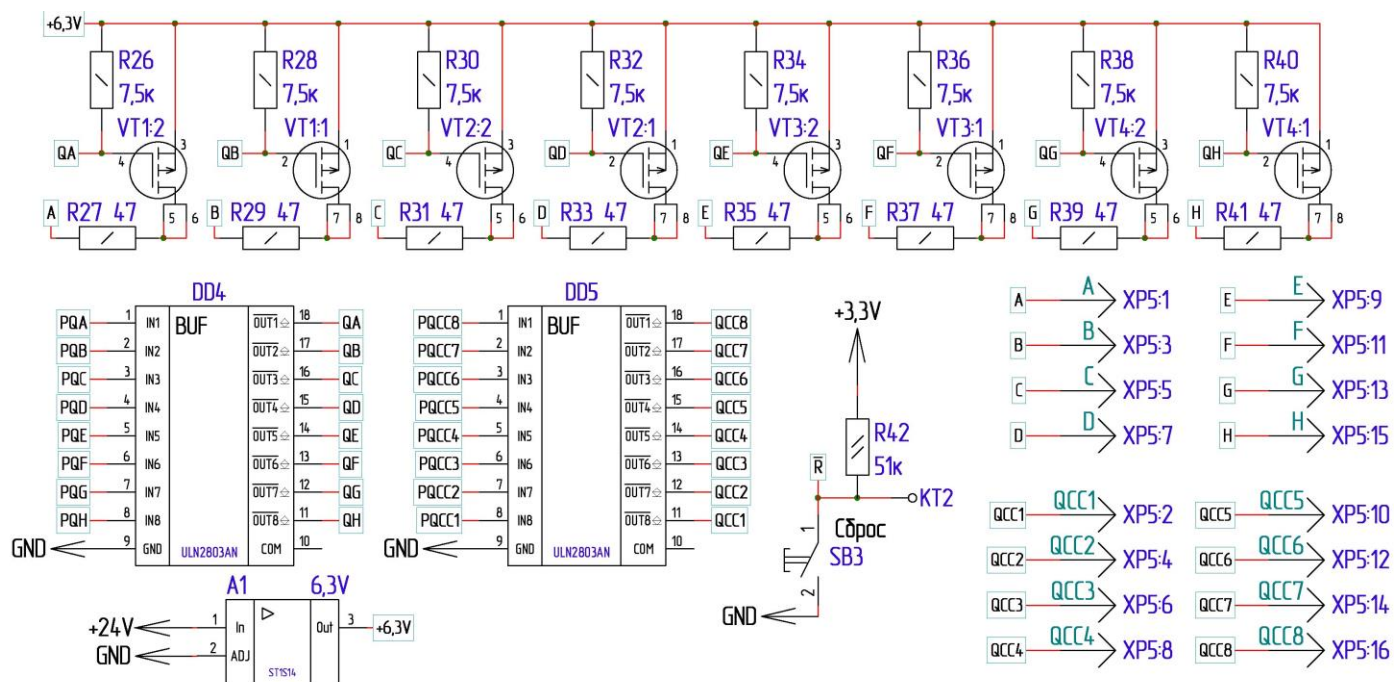


Рис.9. Плата управления БП\_STM32F103RBT Лист2

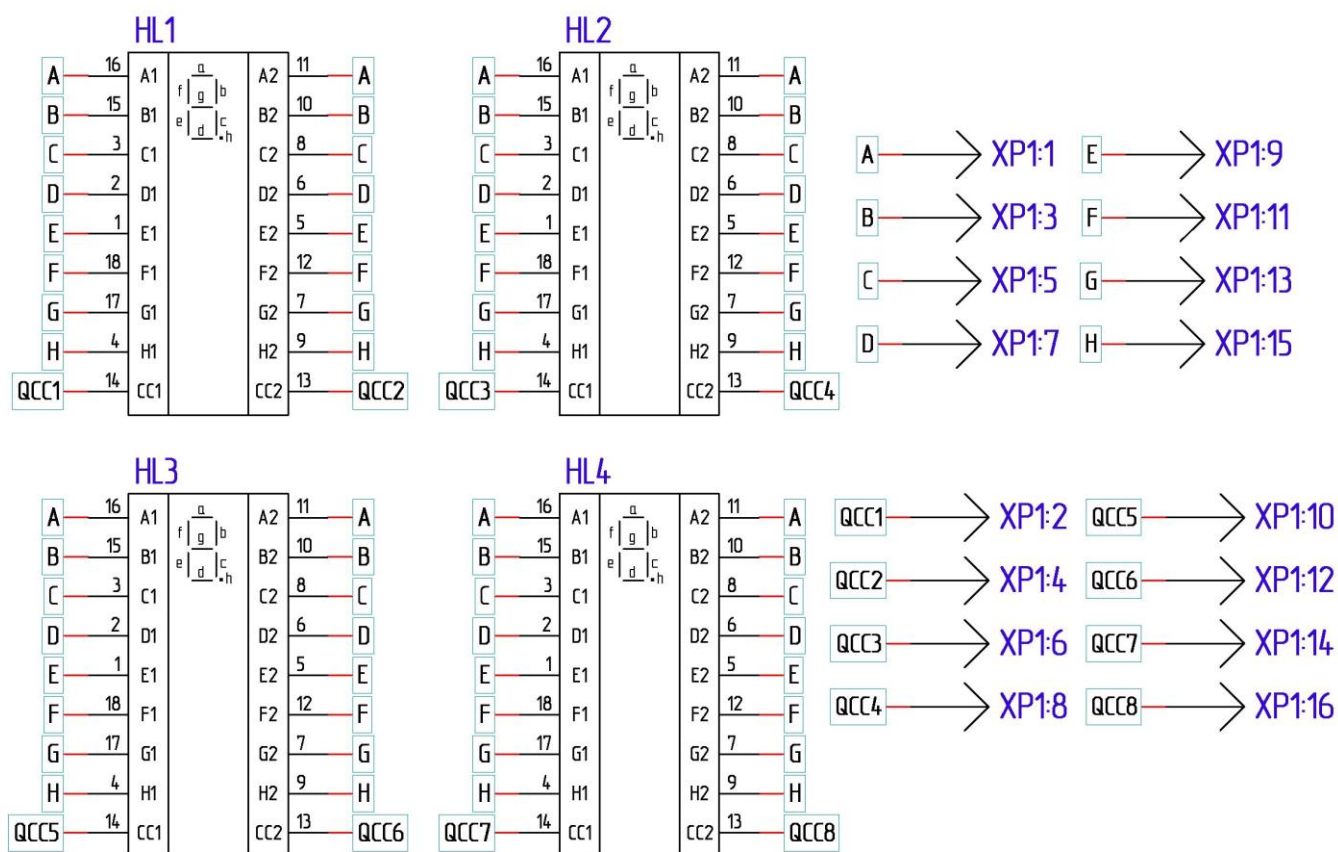


Рис.10. Плата HDSP-5623x2x2-2

### Модуль питания на ST1S14

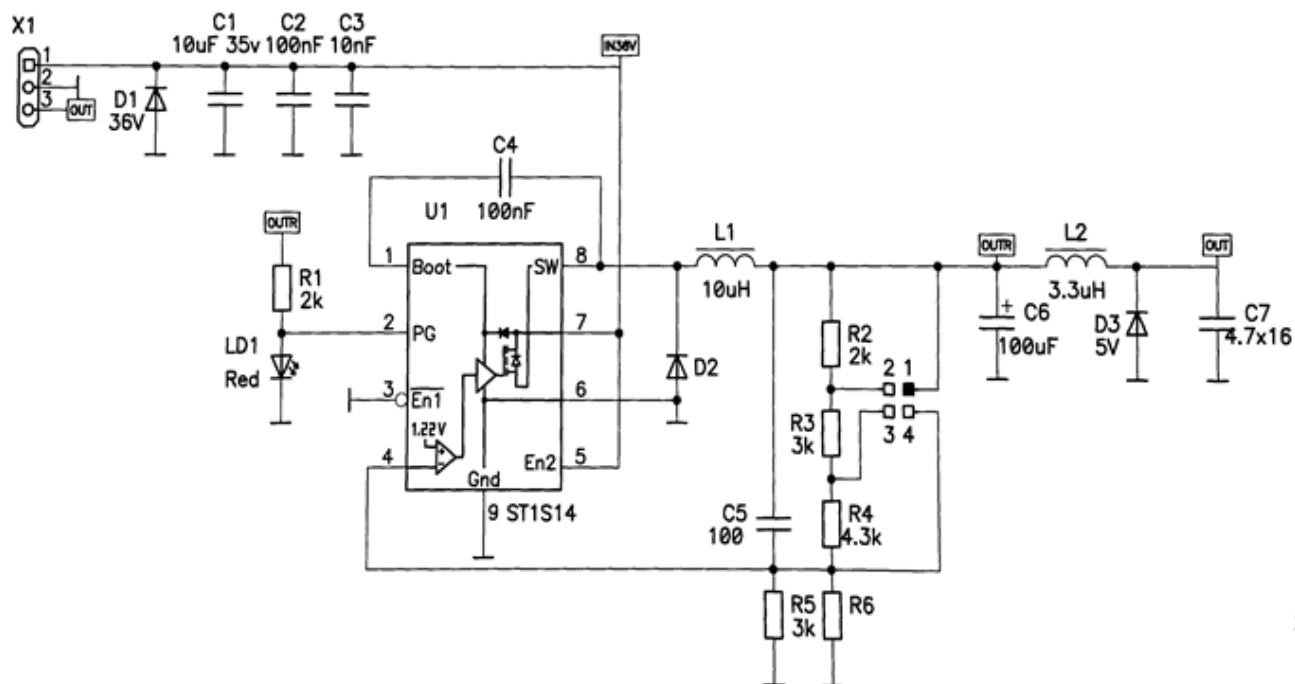
C6, C7 TECAP 68/20VD 10 lowESR (VISHAY)

C13, C14 TECAP 100/10VD 10 lowESR (VISHAY)

L1, L3 CDRH105RNP—100NC (SUMIDA)

L2, L4 CDRH6D38NP—3R3NC (SUMIDA)

D2, D4 SSB44-E3/52T (VISHAY)



| 5V      |       | 12V     |       | 15V |      |
|---------|-------|---------|-------|-----|------|
| 1-2     | 4.19V | 1-2     | 9.96V | R3  | 1,8к |
| 2-3     | 3.78V | 2-3     | 8.95V | R6  | 2к   |
| 3-4     | 3.25V | 3-4     | 6.30V | D3  | 15v  |
| 1-2-3   | 2.97V | 1-2-3   | 6.91V |     |      |
| 2-3-4   | 2.03V | 2-3-4   | 3.25V |     |      |
| 1-2-3-4 | 2.44V | 1-2-3-4 | 4.27V |     |      |
| 1-2-3-4 | 1.22V | 1-2-3-4 | 1.22V |     |      |
|         |       | R4      | 5,6к  |     |      |
|         |       | R6      | 2к    |     |      |
|         |       | D3      | 12v   |     |      |

Рис. 11.

## Программное обеспечение (ПО) блока питания.

Автор: А.В. Жмаев, E-mail: [kuratorj@yandex.ru](mailto:kuratorj@yandex.ru)

### Программа на микроконтроллер ATmega8A

```
; Bloc_Petane2i_ASM_.asm
;
; Created: 24.02.2016 9:27:32
; Author: Жмаев А. В.
; cpu:      ATmega8A
; speed:    8 mhz (max: 8 mhz)
; voltage:  5 V
; ram:      1024 bytes (0x0060-0x045f)
; rom:      4096 bytes (0x0000-0x0fff)
; eeprom:   512 bytes (0x0000-0x01ff)
; CKPOT = 1
; CKSEL3 = 0
; CKSEL2 = 1
; CKSEL1 = 0
; CKSEL0 = 0
;

defM:
    .device ATmega8A
    .include "m8Adef.inc"
    .list

defP:
    .equ SBA_ = 0      ;PORTB
    .equ On = 3        ;PORTB

    .equ ON_5V = 3     ;PORTC

    .equ SBOFF_ = 4    ;PORTC
    .equ SBON_ = 5     ;PORTC

defR:
    .def MulL = r0
    .def MulH = r1

    .def NuL = r2
    .def Sch_In = r4    ;Счетчик индикаторов
    .def RegS = r5      ;Регистр состояния.
        .equ Sb_A = 7      ;1 - признак нажатой кнопки.
        .equ Sb_On = 6     ;1 - признак нажатой кнопки.
        .equ Sb_Off = 5    ;0 - Откл.
        .equ Gotov = 1     ;1 - Готовность
        .equ Err = 0 ;1 - Превышение тока.

    .def N_U = r6        ; Указатель напряжений выхода.
    .def Temp = r7
    .def R_ON5V = r8
    .def R_24V_ = r9
    .def R_24V = r10
    .def UoutH = r11
    .def etL = r12       ;Расогласование e(t-1)
    .def etH = r13
    .def ItL = r14       ;Интегральная составляющая (t-1).
    .def ItH = r15
    .def AL = r16        ;Множитель
    .def AH = r17
```

```

.def BL = r18      ;Множимое
.def BH = r19
.def CL = r20      ;Произведение
.def CH = r21
.def DL = r22      ;Произведение
.def DH = r23
.def AnL = r24
.def AnH = r25

;-----

.cseg
.org 0
;Начало
Reset:RJMP BEGIN
      RJMP Reset
      RJMP Reset
      RJMP TIMER2_COMPA ;Совпадение таймера/счетчика T2
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
      RJMP Reset
;-----

.org 32      ;0x0040
BEGIN:
      LDI AL,LOW(RAMEND) ;Загрузка указателя стека.
      OUT SPL,AL
      LDI AL,HIGH(RAMEND)
      OUT SPH,AL

      RCALL PORT_Init
      RCALL Timer1_Init
      RCALL Timer2_Init
      RCALL ANA_COMP_Init

      LDI AL,1<<WDE      ;Включение сторожевого таймера.
      OUT WDTCR,AL

      LDI AL,4      ;-1,2v _ +5v
      MOV N_U,AL
      CLR Nu1
      MOV RegS,Nu1
      LDI AL,1
      MOV R_ON5V,AL
      MOV R_24V_,AL
      MOV R_24V,AL

      SEI

While:  SBIC PINC,SBON_      ;Нажата - пропуск команды.
      RJMP W_SBON
      LDI AnL,1<<Sb_On      ;Установка Sb_On = 1
      SBRS RegS,Err

```

```

OR RegS,AnL
RJMP W_SBA

W_SBON:    SBRS RegS,Sb_On
           RJMP W_SBA
           SBRC RegS,Gotov           ;Проверка "Готовность"
           SBI PORTB,ON             ;Включение реле выхода.
           LDI AnL,~(1<<Sb_On) ;Установка Sb_On = 0
           AND RegS,AnL

W_SBA:     SBIC PINB,SBA_           ;Нажата - пропуск команды.
           RJMP W_SBA_
           CBI PORTB,ON             ;Отключение реле выхода.
           LDI AnL,1<<Sb_A          ;Установка Sb_A = 1
           OR RegS,AnL
           RJMP W_SBOFF_

W_SBA_:    SBRS RegS,Sb_A
           RJMP W_SBOFF_
           LDI AnL,~(1<<Sb_A) ;Установка Sb_A = 0
           AND RegS,AnL
           ; Переключение Указателя напряжений выхода
           INC N_U
           INC N_U
           MOV AnH,N_U
           CPI AnH,24
           BRLO W_SBOFF_           ;Переход, если меньше.
           CLR N_U

W_SBOFF_:  SBIC PINC,SBOFF_         ;Нажата - пропуск команды.
           RJMP W_UP
           CBI PORTB,ON             ;Отключение реле выхода.
           LDI AnL,1<<Gotov        ;Установка "Готовность" = 1
           MOV RegS,AnL
           SBI PORTC,ON_5V

W_UP:      LDI ZH,HIGH(UP*2)        ;Загрузка указателя UP массива
           LDI ZL,LOW(UP*2)
           ADD ZL,N_U
           ADC ZH,Nu1
           LPM AnL,Z+               ;+Uзад
           LPM AnH,Z

           MOVW BL,AnL
           ASR BH                   ;Деление на 2.
           ROR BL
           OUT OCR1BH,BH            ;Загрузка ШИМ
           OUT OCR1BL,BL

           LDI XH, HIGH(Indec+4)    ;Загрузка указателя Indec массива
           LDI XL, LOW(Indec+4)

           RCALL Int_2_V_Seg10

W_UM:      LDI ZH, HIGH(UM*2)       ;Загрузка указателя UM массива
           LDI ZL, LOW(UM*2)
           ADD ZL,N_U
           ADC ZH,Nu1
           LPM AnL,Z+               ; -Uзад
           LPM AnH,Z

           MOVW BL,AnL
           ASR BH                   ;Деление на 2.
           ROR BL

```

```

OUT OCR1AH,BH          ;Загрузка ШИМ
OUT OCR1AL,BL

LDI XH,HIGH(Indec+8)    ;Загрузка указателя Indec массива
LDI XL,LOW(Indec+8)

RCALL Int_2_V_Seg10

LDI XH,HIGH(Indec+8)    ;Загрузка указателя Indec массива
LDI XL,LOW(Indec+8)
LDI YH,HIGH(IndecUU_+4) ;Загрузка указателя IndecUU_ массива
LDI YL,LOW(IndecUU_+4)

```

```

W_M:    LD AnL,-X
        CPI AnL,0x6f ;Сравнение на ноль.
        BREQ W_M_Z   ;Переход, если равно.
        ST -Y,AnL
        SBRs RegS,Err
        STD Y+8,AnL
        LD AnL,-X
        ORI AnL,0x10 ;Запятая
        ST -Y,AnL
        SBRs RegS,Err
        STD Y+8,AnL
        SBIW XL,2
        RJMP W_P

```

```

W_M_Z:  LD AnL,-X
        ORI AnL,0x10 ;Запятая
        ST -Y,AnL
        SBRs RegS,Err
        STD Y+8,AnL
        LD AnL,-X
        ST -Y,AnL
        SBRs RegS,Err
        STD Y+8,AnL
        SBIW XL,1

```

```

W_P:    LD AnL,-X
        CPI AnL,0x6f ;Сравнение на ноль.
        BREQ W_P_Z   ;Переход, если равно.
        ST -Y,AnL
        SBRs RegS,Err
        STD Y+8,AnL
        LD AnL,-X
        ORI AnL,0x10 ;Запятая
        ST -Y,AnL
        SBRs RegS,Err
        STD Y+8,AnL
        RJMP W_END

```

```

W_P_Z:  LD AnL,-X
        ORI AnL,0x10 ;Запятая
        ST -Y,AnL
        SBRs RegS,Err
        STD Y+8,AnL
        LD AnL,-X
        ST -Y,AnL
        SBRs RegS,Err
        STD Y+8,AnL
        RJMP W_END

```

```

W_End:  RJMP While

```

```

;-----
PORT_Init:
    ;Инициализация порта B
    LDI AL,0b11111110
    OUT DDRB,AL
    CBI PORTB,ON          ;Отключение реле выхода.

    ;Инициализация порта D
    LDI AL,0x00
    OUT PORTD,AL
    ;DDRD=0xff Установка порта D "Вывод"
    LDI AL,0xff
    OUT DDRD,AL

    ;Инициализация порта C
    LDI AL,0b00001000
    OUT DDRC,AL
    SBI PORTC,ON_5V

    RET

;-----
Timer1_Init:
    ;Инициализация Таймер 1
    ;Назначение: Формирователь периода ШИМ - получает clk .
    LDI AL,LOW(750)          ;Частота пробразования 10,666 кГц.
    LDI AH,HIGH(750)
    ;MOVW T_timer1L,AL

    ;ICR1=T_timer1 Переполнение.
    OUT ICR1H,AH
    OUT ICR1L,AL

    LDI AL,0x00
    LDI AH,0x00

    ;OCR1A=0 Совпадение.
    OUT OCR1AH,AH
    OUT OCR1AL,AL

    ;OCR1B=0 Совпадение.
    OUT OCR1BH,AH
    OUT OCR1BL,AL

    LDI AL,(1<<COM1A1)|(1<<COM1B1)|(1<<WGM11)
    OUT TCCR1A,AL

    ;Режим Fast PWM.
    LDI AL,(1<<CS10)|(1<<WGM13)|(1<<WGM12)
    OUT TCCR1B,AL

    RET

;-----
Timer2_Init:
    ;Инициализация Таймер 2
    ;Назначение: Формирователь периода индикации - предделитель clk/64.

    ;TALt=0,001C (1000Гц).
    LDI AL,125
    OUT OCR2,AL

    ;Режим CTC (сброс при совпадении)
    LDI AL,(1<<CS22)|(1<<WGM21)

```

```
OUT TCCR2,AL
```

```
;Прерывание Совпадение Установить.
```

```
LDI AL,(1<<OCIE2)
```

```
OUT TIMSK,AL
```

```
RET
```

```
;-----
```

```
ANA_COMP_Init:
```

```
LDI AL,(1<<ACBG)|(1<<ACIS1)
```

```
OUT ACSR,AL
```

```
LDI AL,(1<<ACME)
```

```
OUT SFIOR,AL
```

```
RET
```

```
;-----
```

```
TIMER2_COMPA:
```

```
PUSH AnL
```

```
IN AnL,SREG
```

```
PUSH AnL
```

```
PUSH YL
```

```
PUSH YH
```

```
PUSH ZL
```

```
PUSH ZH
```

```
PUSH AnH
```

```
WDR ;Сбросить сторожевой таймер
```

```
OUT PORTD,Nu1
```

```
;RJMP T_In
```

```
T_ON5V:
```

```
IN AnL,ADMUX
```

```
CPI AnL,0 ;Проверка номера канала 0.
```

```
BRNE T_24V_
```

```
SBIC ACSR,ACO ; Пропуск команды, если ACO = 0.
```

```
RJMP T_tON5V
```

```
DEC R_ON5V
```

```
BRNE T_24V_
```

```
CBI PORTB,ON ;Отключение реле выхода.
```

```
CBI PORTC,ON_5V
```

```
LDI AnH,1<<Err ;Код сбоя.
```

```
OR RegS,AnH
```

```
LDI YH,HIGH(IndecErr+4) ;Загрузка указателя IndecErr массива
```

```
LDI YL,LOW(IndecErr+4)
```

```
LDI AnL,0x80 ;"-"
```

```
ST -Y,AnL ;Отключение индикации 5V.
```

```
ST -Y,AnL
```

```
LDI AnL,0xad ;"5"
```

```
ST -Y,AnL
```

```
LDI AnL,0x6e ;"U"
```

```
ST -Y,AnL
```

```
RJMP T_24V_
```

```
T_tON5V:
```

```
LDI AnH,1 ;Время задержки срабатывания защиты по току.
```

```
MOV R_ON5V,AnH
```

```
T_24V_:
```

```
CPI AnL,1 ;Проверка номера канала 1.
```

```
BRNE T_24V
```

```
SBIC ACSR,ACO ; Пропуск команды, если ACO = 0.
```

```
RJMP T_t24V_
```

```
DEC R_24V_
```

```
BRNE T_24V
```

```

CBI PORTB,ON          ;Отключение реле выхода.
LDI AnH,1<<Err        ;Код сбоя.
OR RegS,AnH

```

```

LDI YH,HIGH(IndecErr+4) ;Загрузка указателя IndecErr массива
LDI YL,LOW(IndecErr+4)
LDI AnL,0x80           ;"- "
ST -Y,AnL              ;Отключение индикации -24V.
ST -Y,AnL
RJMP T_24V

```

```

T_t24V_: LDI AnH,1          ;Время задержки срабатывания защиты по току.
MOV R_24V_,AnH

```

```

T_24V:   CPI AnL,2          ;Проверка номера канала 2.
BRNE T_AdmC1
SBIC ACSR,ACO          ; Пропуск команды, если ACO = 0.
RJMP T_t24V
DEC R_24V
BRNE T_AdmC1
CBI PORTB,ON          ;Отключение реле выхода.
LDI AnH,1<<Err        ;Код сбоя.
OR RegS,AnH

```

```

LDI YH,HIGH(IndecErr+2) ;Загрузка указателя IndecErr массива
LDI YL,LOW(IndecErr+2)
LDI AnL,0x80           ;"- "
ST -Y,AnL              ;Отключение индикации 24V.
ST -Y,AnL
RJMP T_AdmC1

```

```

T_t24V:   LDI AnH,1          ;Время задержки срабатывания защиты по току.
MOV R_24V,AnH

```

```

T_AdmC1:  INC AnL            ;Проверка номера канала >2.
CPI AnL,3
BRLO T_Adm
CLR AnL

```

```

T_Adm:    OUT ADMUX,AnL

```

```

T_In:     INC Sch_In
SBRC Sch_In,2
CLR Sch_In

```

```

LDI ZH, HIGH(Poz_ZN_Array*2) ;Загрузка указателя Poz_ZN_Array массива
LDI ZL, LOW(Poz_ZN_Array*2)
ADD ZL,Sch_In
LPM Temp,Z
IN AnL,PORTB
ANDI AnL,0x0f
OR AnL,Temp
OUT PORTB,AnL

```

```

LDI YH,HIGH(IndecErr)      ;Загрузка указателя IndecErr массива
LDI YL,LOW(IndecErr)

SBRC RegS,Err
RJMP T_In1

```

```

LDI YH,HIGH(IndecUU_)      ;Загрузка указателя IndecUU_ массива
LDI YL,LOW(IndecUU_)

```

```

T_In1:    ADD YL,Sch_In

```

```

LD AnL,Y
OUT PORTD,AnL

T2_end:    POP    AnH
           POP    ZH
           POP    ZL
           POP    YH
           POP    YL
           POP    AnL
           OUT    SREG,AnL
           POP    AnL
           RETI
;-----

Int_2_V_Seg10:    ;Преобразование числа 2 в 10(Seg).

LDI ZH, HIGH(Form_Zn_Array*2)    ;Загрузка указателя начала массива
LDI ZL, LOW(Form_Zn_Array*2)     ;сегментных кодов чисел.

CLR AH                ;Частное = 0
LDI AL, 0x03          ;1000
CPI AnL, 0xe8
CPC AnH, AL
BRMI K2

K3_1:    INC    AH                ;Частное = Частное + 1
SUBI AnL,0xe8          ;1000
SBCI AnH,0x03

LDI AL, 0x03          ;1000
CPI AnL, 0xe8
CPC AnH, AL
BRCC K3_1

K2:    ANDI ZL,0xf0
ADD ZL,AH
LPM AH,Z
ST -X,AH

;if (Chislo>=100)
CLR AH                ;Частное = 0
LDI AL, 0x00          ;100
CPI AnL, 0x64
CPC AnH, AL
BRMI K1

K2_1:    INC    AH                ;Частное = Частное + 1
SUBI AnL,0x64          ;100
SBCI AnH,0x00

LDI AL, 0x00          ;100
CPI AnL, 0x64
CPC AnH, AL
BRCC K2_1

K1:    ANDI ZL,0xf0
ADD ZL,AH
LPM AH,Z
ST -X,AH

CLR AH                ;Частное = 0
LDI AL, 0x0           ;10
CPI AnL, 0x0a
CPC AnH, AL
BRMI K1_2

```

```

K1_1:      INC     AH           ;Частное = Частное + 1
           SUBI   AnL, 0x0a     ;10
           SBCI   AnH, 0x00

           LDI    AL, 0x0       ;10
           CPI    AnL, 0x0a
           CPC    AnH, AL
           BRCC   K1_1

```

```

K1_2:      ANDI   ZL, 0xf0
           ADD    ZL, AH
           LPM    AH, Z
           ST     -X, AH

           ANDI   ZL, 0xf0
           ADD    ZL, AnL
           LPM    AH, Z
           ST     -X, AH
           RET

```

```

;-----

```

```

           .cseg
           .org 0x180           ;Массив форматов знаков чисел.
Form_Zn_Array: .db 0x6f, 0x0a, 0xc7, 0x8f, 0xaa, 0xad, 0xed, 0x0b, 0xef, 0xaf
               ;(.) = 0x10
           .org 0x188
Poz_ZN_Array: .db 0x10, 0x20, 0x80, 0x40

```

```

           .org 0x190 ; Таблица напряжений +U
UP: .dw 250, 330, 500, 500, 500, 600, 900, 1200, 1200, 1200, 1500, 1500
           .org 0x1b0 ; Таблица напряжений -U
UM: .dw 250, 330, 120, 520, 600, 600, 900, 150, 600, 1200, 150, 1500

```

```

;-----

```

```

           .dseg
           .org 0x80
Indec:     .BYTE 8

           .org 0x90
IndecUU_:  .BYTE 4

           .org 0x98
IndecErr:  .BYTE 4

```

```

;-----

```

## Программа на микроконтроллер STM32F103RB

```
;NAME Svarka 160A_STM32F103C8_Keil
```

```
AREA DV2041sd_103C8_Keil, CODE, READONLY ,align = 3
```

```
GET D:\Documents and Settings\Proekt Keil uVision5\Keil_Bibliotek\stm32f10x_Keil.s
```

```
;GET D:\Documents and Settings\Proekt Keil uVision5\Keil_Bibliotek\mUser.s
```

```
Unit    RN 11    ;"1" - еденица (R11)
```

```
No11    RN 10    ;"0" - ноль (No11)
```

```
Rsos    RN 9     ;Rsos - Регистр СОСТОЯНИЯ ПРОГРАММЫ (R9)
```

```
Rsos_Tind EQU 0      ;Rsos[0...3] - Время выводу на индикатор (Tind*Tper):  
Tper - Период ШИМ Uвх
```

```
Rsos_QCCn EQU 4      ;Rsos[4...7] - Номер разряда Индикатора
```

```
Rsos_Indic_Zad EQU 0x00000100 ; Флаг Rsos: Индикация: 0 - Uout, 1 - Uzaд, Izad
```

```
Rsos_Indic_U_I EQU 0x00000200 ; Флаг Rsos: Индикация стабилизации: 0 - Напряжение, 1 -  
Ток
```

```
Rsos_ROG EQU 0x00000400 ;Включения силового реле зарядки конденсаторов
```

```
;.....
```

```
;Привзка к схеме
```

```
ADCIO EQU 0      ; Канал измерения Iвых*10 (PA0: ADC12_IN0)
```

```
ADCU EQU 1      ; Канал измерения Uвх (PA1: ADC12_IN1)
```

```
ADCUZ EQU 0      ; Канал измерения Uзад (PC0: ADC12_IN10)
```

```
ADCIZ EQU 1      ; Канал измерения Izad (PC1: ADC12_IN11)
```

```
ADCIO EQU 2      ; Канал измерения Iвых (PC2: ADC12_IN12)
```

```
ADCUO EQU 3      ; Канал измерения Uвых (PC3: ADC12_IN13)
```

```
PQA_PA8 EQU 0x0100
```

```
PQB_PA9 EQU 0x0200
```

```
PQC_PA10 EQU 0x0400
```

```
PQD_PA11 EQU 0x0800
```

```
PQE_PA12 EQU 0x1000
```

```
PQF_PC10 EQU 0x0400
```

```
PQG_PC11 EQU 0x0800
```

```
PQH_PC12 EQU 0x1000
```

```
PQCC1_PB12 EQU 0x1000
```

```
PQCC2_PB13 EQU 0x2000
```

```
PQCC3_PB14 EQU 0x4000
```

```
PQCC4_PB15 EQU 0x8000
```

```
PQCC5_PC6 EQU 0x0040
```

```
PQCC6_PC7 EQU 0x0080
```

```
PQCC7_PC8 EQU 0x0100
```

```
PQCC8_PC9 EQU 0x0200
```

```
SB_ON_PA2 EQU 0x00000004 ; PA2 - "0", ВКЛ Выхода
```

```
SB_OFF_PB6 EQU 0x00000040 ; PB6 - "0", ОТКЛ Выхода
```

```
U_PB3 EQU 0x00000008 ; PB3: "1" - Индикация стабилизации напряжения
```

```
I_PB4 EQU 0x00000010 ; PB4: "1" - Индикация стабилизации тока
```

```
ROG_PA5 EQU 0x00000020 ; PA5 - "1": ВКЛ реле зарядки конденсаторов (отключение  
ограничивающего резистора)
```

```
OUT_ONR_PB7 EQU 0x00000080 ; PB7 - "1", ВКЛ реле и индикация Выхода
```

```
Control_PB9 EQU 0x00000200 ;Контрольный вывод
```

```
;.....
```

*;Операивная память*

```

SRAM_Control0 EQU 0x20000000 ; 4 байта
SRAM_Control1 EQU 0x20000004 ; 4 байта
SRAM_Control2 EQU 0x20000008 ; 4 байта
SRAM_Control3 EQU 0x2000000c ; 4 байта

SRAM_Uout_ind EQU 0x20000010
SRAM_Iout_ind EQU 0x20000014

SRAM_Uin EQU 0x20000020
SRAM_Uzad EQU 0x20000024
SRAM_Izad EQU 0x20000028
SRAM_Uout EQU 0x2000002c
SRAM_Iout EQU 0x20000030

SRAM_et_Iout EQU 0x20000038 ; 4 байта
SRAM_It_Iout EQU 0x2000003c ; 4 байта
SRAM_et_Uout EQU 0x20000040 ; 4 байта
SRAM_It_Uout EQU 0x20000044 ; 4 байта
SRAM_et_Uin EQU 0x20000048 ; 4 байта
SRAM_It_Uin EQU 0x2000004c ; 4 байта

SRAM_IU_count EQU 0x20000050 ; 4 байта
SRAM_I_summ EQU 0x20000054 ; 4 байта
SRAM_U_summ EQU 0x20000058 ; 4 байта

SRAM_Iz_summ EQU 0x20000060 ; 4 байта
SRAM_Uz_summ EQU 0x20000064 ; 4 байта

SRAM_Uind EQU 0x20000068 ; 4 байта
SRAM_Iind EQU 0x2000006c ; 4 байта
SRAM_Indic EQU 0x20000070 ; 8 байта
SRAM_Ind_seg EQU 0x20000078 ; 8 байта

SRAM_ADC_UI EQU 0x20000080 ; Адрес DMA - начало массива в SRAM (128 байт)

SRAM_I EQU 0x20000100 ; 512 байт
SRAM_U EQU 0x20000300 ; 512 байт

SRAM_Iz EQU 0x20000500 ; 512 байт
SRAM_Uz EQU 0x20000700 ; 512 байт

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

; Форвардная декларация разделов.

```

```

AREA RESET, DATA, READONLY, align = 2
THUMB

```

```

__vector_table

```

```

;size      acronime      IRQN Описание смещения
DCD 0x20002000 ; - 0x00000000 - адрес начала стека
DCD Reset_IRQHandler ; - 0x00000004 - начало программы
DCD NMI_IRQHandler ; - 0x00000008 - немаскируемое прерывание
DCD HardFault_IRQHandler ; - 0x0000000c - тяжёлый отказ
DCD MemManageFault_IRQHandler ; - 0x00000010 - отказ системы управления памятью
DCD BusFault_IRQHandler ; - 0x00000014 - отказ шины
DCD UsageFault_IRQHandler ; - 0x00000018 - отказ программы
DCD 0 ; - 0x0000001c - зарезервировано
DCD 0 ; - 0x00000020 - зарезервировано
DCD 0 ; - 0x00000024 - зарезервировано
DCD 0 ; - 0x00000028 - зарезервировано
DCD SVCcall_IRQHandler ; - 0x0000002c - вызов супервизора
DCD Debug_Monitor_IRQHandler ; - 0x00000030 - исключение монитора отладки

```

|                                           |                    |                           |            |                                       |                                          |
|-------------------------------------------|--------------------|---------------------------|------------|---------------------------------------|------------------------------------------|
| DCD                                       | 0                  | ; -                       | 0x00000034 | - зарезервировано                     |                                          |
| DCD                                       | PendSV_IRQHandler  | ; -                       | 0x00000038 | - запрос системной службы             |                                          |
| DCD                                       | SysTick_IRQHandler | ; -                       | 0x0000003C | - системный таймер                    |                                          |
| DCD                                       | WWDG_IRQHandler    | ; 0                       | 0x00000040 | - оконный сторожевой таймер           |                                          |
| DCD                                       | PVD_IRQHandler     | ; 1                       | 0x00000044 | - программируемый детектор напряжения |                                          |
| линия 16 модуля EXTI)                     |                    |                           |            |                                       |                                          |
| DCD                                       | TAMPER_IRQHandler  | ; 2                       | 0x00000048 | - прерывание от события TAMPER        |                                          |
| DCD                                       | RTC_IRQHandler     | ; 3                       | 0x0000004C | - глобальное прерывание от часов      |                                          |
| реального времени                         |                    |                           |            |                                       |                                          |
| DCD                                       | FLASH_IRQHandler   | ; 4                       | 0x00000050 | - любое немаскированное прерывание от |                                          |
| модуля FSMC                               |                    |                           |            |                                       |                                          |
| DCD                                       | RCC_IRQHandler     | ; 5                       | 0x00000054 | - любое немаскированное прерывание от |                                          |
| модуля RCC                                |                    |                           |            |                                       |                                          |
| DCD                                       | EXTI0_IRQHandler   | ; 6                       | 0x00000058 | - прерывание от линии 0 модуля EXTI   |                                          |
| (PA0, PB0, PC0, PD0, PE0, PF0, PG0)       | DCD                | EXTI1_IRQHandler          | ; 7        | 0x0000005C                            | - прерывание от линии 1 модуля EXTI      |
| (PA1, PB1, PC1, PD1, PE1, PF1, PG1)       | DCD                | EXTI2_IRQHandler          | ; 8        | 0x00000060                            | - прерывание от линии 2 модуля EXTI      |
| (PA2, PB2, PC2, PD2, PE0, PF2, PG2)       | DCD                | EXTI3_IRQHandler          | ; 9        | 0x00000064                            | - прерывание от линии 3 модуля EXTI      |
| (PA3, PB3, PC3, PD3, PE3, PF3, PG3)       | DCD                | EXTI4_IRQHandler          | ; 10       | 0x00000068                            | - прерывание от линии 4 модуля EXTI      |
| (PA4, PB4, PC4, PD4, PE4, PF4, PG4)       | DCD                | DMA1_Channel1_IRQHandler  | ; 11       | 0x0000006C                            | - любое немаскированное прерывание от    |
| канала 1 модуля DMA1                      | DCD                | DMA1_Channel2_IRQHandler  | ; 12       | 0x00000070                            | - любое немаскированное прерывание от    |
| канала 2 модуля DMA1                      | DCD                | DMA1_Channel3_IRQHandler  | ; 13       | 0x00000074                            | - любое немаскированное прерывание от    |
| канала 3 модуля DMA1                      | DCD                | DMA1_Channel4_IRQHandler  | ; 14       | 0x00000078                            | - любое немаскированное прерывание от    |
| канала 4 модуля DMA1                      | DCD                | DMA1_Channel5_IRQHandler  | ; 15       | 0x0000007C                            | - любое немаскированное прерывание от    |
| канала 5 модуля DMA1                      | DCD                | DMA1_Channel6_IRQHandler  | ; 16       | 0x00000080                            | - любое немаскированное прерывание от    |
| канала 6 модуля DMA1                      | DCD                | DMA1_Channel7_IRQHandler  | ; 17       | 0x00000084                            | - любое немаскированное прерывание от    |
| канала 7 модуля DMA1                      | DCD                | ADC1_2_IRQHandler         | ; 18       | 0x00000088                            | - любое немаскированное прерывание от    |
| модулей ADC1, ADC2                        | DCD                | USB_HP_CAN_TX_IRQHandler  | ; 19       | 0x0000008C                            | - высокоприоритетные прерывания от USB   |
| или от передатчика модуля CAN             | DCD                | USB_HP_CAN_RX0_IRQHandler | ; 20       | 0x00000090                            | - низкоприоритетные прерывания от USB    |
| или от приёмного буфера FIFO_0 модуля CAN | DCD                | CAN_RX1_IRQHandler        | ; 21       | 0x00000094                            | - прерывание от приёмного буфера FIFO_1  |
| модуля CAN                                | DCD                | CAN_SCE_IRQHandler        | ; 22       | 0x00000098                            | - прерывание от события CAN Status       |
| Change Error                              | DCD                | EXTI9_5_IRQHandler        | ; 23       | 0x0000009C                            | - прерывание от линий [9:5] модуля EXTI  |
|                                           | DCD                | TIM1_BRK_IRQHandler       | ; 24       | 0x000000A0                            | - прерывание по событию Break модуля     |
| TIM1                                      | DCD                | TIM1_UP_IRQHandler        | ; 25       | 0x000000A4                            | - прерывание по событию Update модуля    |
| TIM1                                      | DCD                | TIM1_TRG_COM_IRQHandler   | ; 26       | 0x000000A8                            | - прерывание от запуска или события COM  |
| модуля TIM1                               | DCD                | TIM1_CC_IRQHandler        | ; 27       | 0x000000AC                            | - прерывание от события захват/сравнение |
| модуля TIM1                               | DCD                | TIM2_IRQHandler           | ; 28       | 0x000000B0                            | - любое прерывание от модуля TIM2        |
|                                           | DCD                | TIM3_IRQHandler           | ; 29       | 0x000000B4                            | - любое прерывание от модуля TIM3        |
|                                           | DCD                | TIM4_IRQHandler           | ; 30       | 0x000000B8                            | - любое прерывание от модуля TIM4        |
|                                           | DCD                | I2C1_EV_IRQHandler        | ; 31       | 0x000000BC                            | - прерывание от события event модуля     |
| I2C1                                      | DCD                | I2C1_ER_IRQHandler        | ; 32       | 0x000000C0                            | - прерывание от события error модуля     |
| I2C1                                      | DCD                | I2C2_EV_IRQHandler        | ; 33       | 0x000000C4                            | - прерывание от события event модуля I2C |

```

I2C2    DCD    I2C2_ER_IRQHandler      ; 34 0x000000C8 - прерывание от события error модуля
I2C2    DCD    SPI1_IRQHandler         ; 35 0x000000CC - прерывание от модуля SPI1
        DCD    SPI2_IRQHandler         ; 36 0x000000D0 - прерывание от модуля SPI2
        DCD    USART1_IRQHandler      ; 37 0x000000D4 - прерывание от модуля USART1
        DCD    USART2_IRQHandler      ; 38 0x000000D8 - прерывание от модуля USART2
        DCD    USART3_IRQHandler      ; 39 0x000000DC - прерывание от модуля USART3
        DCD    EXTI15_10_IRQHandler   ; 40 0x000000E0 - прерывание от линий [15:10] модуля

EXTI     DCD    RTCAlarm_IRQHandler    ; 41 0x000000E4 - прерывание от события RTC alarm (линия
17 модуля EXTI)
        DCD    USBWakeup_IRQHandler    ; 42 0x000000E8 - прерывание от события USB wakeup from
suspend (линия 18 модуля EXTI)
        DCD    TIM8_BRK_IRQHandler     ; 43 0x000000EC - прерывание по событию Break модуля
TIM8
        DCD    TIM8_UP_IRQHandler      ; 44 0x000000F0 - прерывание по событию Update модуля
TIM8
        DCD    TIM8_TRG_COM_IRQHandler ; 45 0x000000F4 - прерывание от запуска или события COM
модуля TIM8
        DCD    TIM8_CC_IRQHandler      ; 46 0x000000F8 - прерывание от события захват/сравнение
модуля TIM8
        DCD    ADC3_IRQHandler         ; 47 0x000000FC - прерывание от модуля ADC3
        DCD    FSMC_IRQHandler         ; 48 0x00000100 - прерывание от модуля FSMC
        DCD    SDIO_IRQHandler         ; 49 0x00000104 - прерывание от модуля SDIO
        DCD    TIM5_IRQHandler         ; 50 0x00000108 - прерывание от модуля TIM5
        DCD    SPI3_IRQHandler         ; 51 0x0000010C - прерывание от модуля SPI3
        DCD    UART4_IRQHandler        ; 52 0x00000110 - прерывание от модуля UART4
        DCD    UART5_IRQHandler        ; 53 0x00000114 - прерывание от модуля UART5
        DCD    TIM6_IRQHandler         ; 54 0x00000118 - прерывание от модуля TIM6
        DCD    TIM7_IRQHandler         ; 55 0x0000011C - прерывание от модуля TIM7
        DCD    DMA2_Channel1_IRQHandler ; 56 0x00000120 - прерывание от канала Channel1 модуля
DMA2
        DCD    DMA2_Channel2_IRQHandler ; 57 0x00000124 - прерывание от канала Channel2 модуля
DMA2
        DCD    DMA2_Channel3_IRQHandler ; 58 0x00000128 - прерывание от канала Channel3 модуля
DMA2
        DCD    DMA2_Channel4_5_IRQHandler ; 59 0x0000012C - прерывание от каналов Channel4 и
Channel5 модуля DMA2

;;;;;;;;;;;;;

        AREA START, CODE, READONLY, align = 2
        THUMB

        ENTRY

Reset_IRQHandler PROC      ; - начало программы
    B Start
    ENDP

NMI_IRQHandler          ; - немаскируемое прерывание
    BKPT #1

HardFault_IRQHandler    ; - тяжёлый отказ
    TST    LR, #0x4        ; Проверяем 2-й бит значения EXC_RETURN в регистре LR
    ITTEE EQ              ; Если ноль (равно), то
    MRSEQ R0, MSP          ; использовался основной стек, помещаем MSP в R0
    LDREQ R0, [R0, #24]    ; Извлекаем старое значение PC из стека
    MRSNE R0, PSP          ; Иначе, использовался стек процесса, помещаем PSP в R0
    LDRNE R0, [R0, #24]    ; Извлекаем старое значение PC из стека
    BKPT #2

MemManageFault_IRQHandler ; - отказ системы управления памятью
    BKPT #1

BusFault_IRQHandler     ; - отказ шины
    BKPT #1

```

```

UsageFault_IRQHandler      ; - отказ программы
    BKPT #1
SVCall_IRQHandler          ; - вызов супервизора
    BKPT #1
Debug_Monitor_IRQHandler   ; - исключение монитора отладки
    BKPT #1
PendSV_IRQHandler          ; - запрос системной службы
    BKPT #1
SysTick_IRQHandler         ; - системный таймер
    BKPT #1
WWDG_IRQHandler            ; - оконный сторожевой таймер
    BKPT #1
PVD_IRQHandler             ; - программируемый детектор напряжения (линия 16 модуля EXTI)
    BKPT #1
TAMPER_IRQHandler          ; - прерывание от события TAMPER
    BKPT #1
RTC_IRQHandler             ; - глобальное прерывание от часов реального времени
    BKPT #1
FLASH_IRQHandler           ; - любое немаскированное прерывание от модуля FSMC
    BKPT #1
RCC_IRQHandler             ; - любое немаскированное прерывание от модуля RCC
    BKPT #1
EXTI0_IRQHandler           ; - прерывание от линии 0 модуля EXTI (PA0, PB0, PC0, PD0, PE0, PF0, PG0)
    BKPT #1
EXTI1_IRQHandler           ; - прерывание от линии 1 модуля EXTI (PA1, PB1, PC1, PD1, PE1, PF1, PG1)
    BKPT #1
EXTI2_IRQHandler           ; - прерывание от линии 2 модуля EXTI (PA2, PB2, PC2, PD2, PE0, PF2, PG2)
    MOV32 R0, #Inter_sb_On ; PA2
    BX R0
EXTI3_IRQHandler           ; - прерывание от линии 3 модуля EXTI (PA3, PB3, PC3, PD3, PE3, PF3, PG3)
    BKPT #1
EXTI4_IRQHandler           ; - прерывание от линии 4 модуля EXTI (PA4, PB4, PC4, PD4, PE4, PF4, PG4)
    BKPT #1
DMA1_Channel1_IRQHandler   ; - любое немаскированное прерывание от канала 1 модуля DMA1
    BKPT #1
DMA1_Channel2_IRQHandler   ; - любое немаскированное прерывание от канала 2 модуля DMA1
    BKPT #1
DMA1_Channel3_IRQHandler   ; - любое немаскированное прерывание от канала 3 модуля DMA1
    BKPT #1
DMA1_Channel4_IRQHandler   ; - любое немаскированное прерывание от канала 4 модуля DMA1
    BKPT #1
DMA1_Channel5_IRQHandler   ; - любое немаскированное прерывание от канала 5 модуля DMA1
    BKPT #1
DMA1_Channel6_IRQHandler   ; - любое немаскированное прерывание от канала 6 модуля DMA1
    BKPT #1
DMA1_Channel7_IRQHandler   ; - любое немаскированное прерывание от канала 7 модуля DMA1
    BKPT #1
ADC1_2_IRQHandler          ; - любое немаскированное прерывание от модулей ADC1, ADC2
    BKPT #1
USB_HP_CAN_TX_IRQHandler   ; - высокоприоритетные прерывания от USB или от передатчика модуля CAN
    BKPT #1
USB_HP_CAN_RX0_IRQHandler  ; - низкоприоритетные прерывания от USB или от приёмного буфера FIFO_0
    ; модуля CAN
    BKPT #1
CAN_RX1_IRQHandler         ; - прерывание от приёмного буфера FIFO_1 модуля CAN
    BKPT #1
CAN_SCE_IRQHandler         ; - прерывание от события CAN Status Change Error
    BKPT #1
EXTI9_5_IRQHandler         ; - прерывание от линий [9:5] модуля EXTI
    MOV32 R0, #Inter_sb_Off ; PB6
    BX R0
TIM1_BRK_IRQHandler        ; - прерывание по событию Break модуля TIM1
    BKPT #1
TIM1_UP_IRQHandler         ; - прерывание по событию Update модуля TIM1
    BKPT #1

```

```

TIM1_TRG_COM_IRQHandler    ; - прерывание от запуска или события COM модуля TIM1
    BKPT #1
TIM1_CC_IRQHandler         ; - прерывание от события захват/сравнение модуля TIM1
    BKPT #1
TIM2_IRQHandler            ; - любое прерывание от модуля TIM2
    MOV32 R0, #Inter_TIM2
    BX R0
TIM3_IRQHandler            ; - любое прерывание от модуля TIM3
    BKPT #1
TIM4_IRQHandler            ; - любое прерывание от модуля TIM4
    BKPT #1
I2C1_EV_IRQHandler         ; - прерывание от события event модуля I2C1
    BKPT #1
I2C1_ER_IRQHandler         ; - прерывание от события error модуля I2C1
    BKPT #1
I2C2_EV_IRQHandler         ; - прерывание от события event модуля I2C2
    BKPT #1
I2C2_ER_IRQHandler         ; - прерывание от события error модуля I2C2
    BKPT #1
SPI1_IRQHandler            ; - прерывание от модуля SPI1
    BKPT #1
SPI2_IRQHandler            ; - прерывание от модуля SPI2
    BKPT #1
USART1_IRQHandler          ; - прерывание от модуля USART1
    BKPT #1
USART2_IRQHandler          ; - прерывание от модуля USART2
    BKPT #1
USART3_IRQHandler          ; - прерывание от модуля USART3
    BKPT #1
EXTI15_10_IRQHandler       ; - прерывание от линий [15:10] модуля EXTI
    BKPT #1
RTCAlarm_IRQHandler        ; - прерывание от события RTC alarm (линия 17 модуля EXTI)
    BKPT #1
USBWakeup_IRQHandler        ; - прерывание от события USB wakeup from suspend (линия 18 модуля EXTI)
    BKPT #1
TIM8_BRK_IRQHandler        ; - прерывание по событию Break модуля TIM8
    BKPT #1
TIM8_UP_IRQHandler         ; - прерывание по событию Update модуля TIM8
    BKPT #1
TIM8_TRG_COM_IRQHandler    ; - прерывание от запуска или события COM модуля TIM8
    BKPT #1
TIM8_CC_IRQHandler         ; - прерывание от события захват/сравнение модуля TIM8
    BKPT #1
ADC3_IRQHandler            ; - прерывание от модуля ADC3
    BKPT #1
FSMC_IRQHandler            ; - прерывание от модуля FSMC
    BKPT #1
SDIO_IRQHandler            ; - прерывание от модуля SDIO
    BKPT #1
TIM5_IRQHandler            ; - прерывание от модуля TIM5
    BKPT #1
SPI3_IRQHandler            ; - прерывание от модуля SPI3
    BKPT #1
UART4_IRQHandler           ; - прерывание от модуля UART4
    BKPT #1
UART5_IRQHandler           ; - прерывание от модуля UART5
    BKPT #1
TIM6_IRQHandler            ; - прерывание от модуля TIM6
    BKPT #1
TIM7_IRQHandler            ; - прерывание от модуля TIM7
    BKPT #1
DMA2_Channel1_IRQHandler   ; - прерывание от канала Channel1 модуля DMA2
    BKPT #1
DMA2_Channel2_IRQHandler   ; - прерывание от канала Channel2 модуля DMA2
    BKPT #1

```

```

DMA2_Channel13_IRQHandler ; - прерывание от канала Channel3 модуля DMA2
    BKPT #1
DMA2_Channel14_5_IRQHandler ; - прерывание от каналов Channel4 и Channel5 модуля DMA2
    BKPT #1
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    align
; Начало программы
Start    PROC
        NOP
        MOV    Noll, #0
        MOV    Unit, #1

;Инициализация тактирования
        MOV32  R0, #RCC
; Затактировать POWER control
        LDR    R1, [R0,#RCC_APB1ENR]
        ORR    R1, R1, #(RCC_APB1ENR_PWREN+RCC_APB1ENR_BKPEN)
        STR    R1, [R0,#RCC_APB1ENR] ;Включение Такты интерфейса питания и Включение
синхронизации интерфейса резервного копирования

; Установить LATENCY time = 2 машинных цикла
; + задействовать буферизацию FLASH (по умолчанию всегда включена)

        MOV32  R2, #FLASH
        MOV    R1, #(FLASH_ACR_LATENCY_2 + FLASH_ACR_PRFTBE)
        STR    R1, [R2,#FLASH_ACR] ;Два состояния ожидания, если 48 МГц<SYSCLK<72
МГц и включение буфера предварительной выборки

; Включить усилитель кварца HSEON=1
        LDR    R1, [R0,#RCC_CR]
        ORR    R1, R1, #RCC_CR_HSEON ;1:включение тактов HSE
        STR    R1, [R0,#RCC_CR]

; Ждём готовности кварца, контролируя бит RCC_CR_HSERDY
RCC_hserdy
        LDR    R1, [R0,#RCC_CR]
        TST    R1, #RCC_CR_HSERDY ;флаг готовности тактов HSE.
        BEQ    RCC_hserdy
; Если кварц окажется неисправен, этот цикл может стать бесконечным.

PLLMUL EQU    (8-2) << 18 ; Коэффициент умножения основной системы ФАПЧ (PLL). здесь 8 = коэфф.
умножения PLLMUL. SYSCLK = 80 mHz

        MOV32  R1, #(PLLMUL + RCC_CFGR_PLLSRC + RCC_CFGR_PPRE1_DIV2+RCC_CFGR_PPRE2_DIV2)
;PPRE1 = 40 mHz, TIM2...4 - 80 mHz
        STR    R1, [R0,#RCC_CFGR] ; Регистр конфигурации RCC PLL

; Включаем PLL
        LDR    R1, [R0,#RCC_CR]
        ORR    R1, R1, #RCC_CR_PLLON
        STR    R1, [R0,#RCC_CR] ; PLL - Включить

; Ждём готовности PLL
RCC_pllrdy
        LDR    R1, [R0,#RCC_CR]
        TST    R1, #RCC_CR_PLLRDY ; PLL флаг - Такты готовы (0x02000000)
        BEQ    RCC_pllrdy

; Выбираем PLL как источник тактового сигнала для микроконтроллера
        LDR    R1, [R0,#RCC_CFGR]
        MOV32  R2, #(RCC_CFGR_SW_PLL+RCC_CFGR_ADCPRE_DIV8) ; PLL выбор в качестве
системных Тактов, Период Тактов АЦП 0,1 мкс
        ORR    R1, R1, R2
        STR    R1, [R0,#RCC_CFGR] ; Регистр конфигурации RCC PLL

```

```

; Ждём готовности
RCC_swsp11
    LDR    R1, [R0,#RCC_CFGR]
    TST    R1, #RCC_CFGR_SWS_PLL    ; Состояние PLL - использует в качестве системных
Тактов
    BEQ    RCC_swsp11

; Включение внутреннего низкоскоростного генератора LSI
    LDR    R1, [R0,#RCC_CSR]
    MOV32  R2, #RCC_CSR_LSION
    ORR    R1, R1,R2
    STR    R1, [R0,#RCC_CSR]    ; Регистр конфигурации RCC PLL

; Ждём готовности LSI
RCC_lsi
    LDR    R1, [R0,#RCC_CSR]
    TST    R1, #RCC_CSR_LSIRDY; "1" - Встроенный низкоскоростной генератор готов
    BEQ    RCC_lsi

;.....

Init_Clok    ;Инициализация Тактирования альтернатив и портов

    LDR    R1, [R0,#RCC_APB2ENR]
    MOV32  R2, #(RCC_APB2ENR_IOPAEN+RCC_APB2ENR_IOPBEN+RCC_APB2ENR_IOPCEN+
RCC_APB2ENR_AFIOEN)
    ORR    R1, R1,R2
    STR    R1, [R0,#RCC_APB2ENR]

;.....

Init_Afio    ;Инициализация альтернатив

    MOV32  R0, #AFIO
    MOV32  R1, #(AFIO_MAPR_SWJ_CFG_JTAGDISABLE + AFIO_MAPR_TIM2_REMAP_PARTIALREMAP2 +
AFIO_MAPR_TIM1_REMAP_PARTIALREMAP) ;JTAG-DP отключен и SW-DP включен.
    STR    R1, [R0,#AFIO_MAPR]

;Внешнее прерывание
    MOV32  R1, #AFIO_EXTICR1_EXTI2_PA
    STR    R1, [R0,#AFIO_EXTICR1]
    MOV32  R1, #AFIO_EXTICR2_EXTI6_PB
    STR    R1, [R0,#AFIO_EXTICR2]

;.....

Init_Port    ;Инициализация портов

                ;инициализация порт A
    MOV32  R0, #GPIOA
    STR    R0, [R0,#GPIO_ODR]
    MOV32  R1, #((RES_IN<<(ODR7*4))+(RES_IN<<(ODR6*4))+(PP_2<<(ODR5*4))+(RES_IN<<
(ODR4*4))+(RES_IN<<(ODR3*4))+(RES_IN<<(ODR2*4))+(AN_IN<<(ODR1*4))+(RES_IN<<(ODR0*4)))
    STR    R1, [R0,#GPIO_CRL]
    MOV32  R1, #((RES_IN<<(ODR15*4))+(RES_IN<<(ODR14*4))+(RES_IN<<(ODR13*4))+(
PP_2<<(ODR12*4))+(PP_2<<(ODR11*4))+(PP_2<<(ODR10*4))+(PP_2<<(ODR9*4))+(PP_2<<(ODR8*4)))
    STR    R1, [R0,#GPIO_CRH]
    ;Блокировка порта
    MOV32  R1, #0x1ffff
    STR    R1, [R0,#GPIO_LCKR]
    MOV32  R1, #0x0ffff
    STR    R1, [R0,#GPIO_LCKR]
    MOV32  R1, #0x1ffff
    STR    R1, [R0,#GPIO_LCKR]

```

```

LDR    R1, [R0,#GPIO_LCKR]

        ;инициализация порт B
MOV32  R0, #GPIOB

STR     No11, [R0,#GPIO_ODR]

MOV32  R1, #((PP_2<<(ODR7*4))+(RES_IN<<(ODR6*4))+(RES_IN<<(ODR5*4))+(
(PP_2<<(ODR4*4)))+(PP_2<<(ODR3*4)))+(PP_2<<(ODR2*4))+(RES_IN<<(ODR1*4))+(AF_PP_50<<(ODR0*4)))
STR     R1, [R0,#GPIO_CRL]

MOV32  R1, #((PP_2<<(ODR15*4))+(PP_2<<(ODR14*4))+(PP_2<<(ODR13*4))+(
(PP_2<<(ODR12*4)))+(AF_PP_50<<(ODR11*4)))+(AF_PP_50<<(ODR10*4)))+(PP_2<<(ODR9*4))+(
(RES_IN<<(ODR8*4))))
STR     R1, [R0,#GPIO_CRH]
        ;Блокировка порта
MOV32  R1, #0x1ffff
STR     R1, [R0,#GPIO_LCKR]
MOV32  R1, #0x0ffff
STR     R1, [R0,#GPIO_LCKR]
MOV32  R1, #0x1ffff
STR     R1, [R0,#GPIO_LCKR]
LDR     R1, [R0,#GPIO_LCKR]

        ;инициализация порт C
MOV32  R0, #GPIOC

STR     No11, [R0,#GPIO_ODR]

MOV32  R1, #((PP_2<<(ODR7*4))+(PP_2<<(ODR6*4))+(RES_IN<<(ODR5*4))+(
(RES_IN<<(ODR4*4)))+(AN_IN<<(ODR3*4)))+(AN_IN<<(ODR2*4)))+(AN_IN<<(ODR1*4)))+(AN_IN<<(ODR0*4)))
STR     R1, [R0,#GPIO_CRL]

MOV32  R1, #((RES_IN<<(ODR15*4))+(RES_IN<<(ODR14*4))+(RES_IN<<(ODR13*4))+(PP_2<<
(ODR12*4)))+(PP_2<<(ODR11*4)))+(PP_2<<(ODR10*4)))+(PP_2<<(ODR9*4)))+(PP_2<<(ODR8*4)))
STR     R1, [R0,#GPIO_CRH]
        ;Блокировка порта
MOV32  R1, #0x1ffff
STR     R1, [R0,#GPIO_LCKR]
MOV32  R1, #0x0ffff
STR     R1, [R0,#GPIO_LCKR]
MOV32  R1, #0x1ffff
STR     R1, [R0,#GPIO_LCKR]
LDR     R1, [R0,#GPIO_LCKR]

;.....

Init_TIM1    ;Инициализация TIM1 - для ШИМ Uвых

        ; Тактирование TIM1
MOV32  R0, #RCC
LDR     R1, [R0,#RCC_APB2ENR]
ORR     R1, #RCC_APB2ENR_TIM1EN
STR     R1, [R0,#RCC_APB2ENR]

        ;Инициализация TIM1
MOV32  R0, #TIM1

STR     No11, [R0,#TIM_PSC]          ; TIM предделитель
MOV     R1, #16835                   ; Тпер = 204,8 мкс, при f = 80 МГц
STR     R1, [R0,#TIM_ARR]           ; TIM регистр авто-перезагрузки
MOV     R2, #1024
STR     R2, [R0,#TIM_CCR2]
STR     No11, [R0,#TIM_CCR1]
STR     No11, [R0,#TIM_CCR3]

```

```

MOV    R2, #(TIM_CR1_ARPE)
STR    R2, [R0,#TIM_CR1]           ; TIM управляющий регистр 1
STR    R2, [R0,#TIM_CR2]           ; TIM управляющий регистр 2

MOV    R2, #(TIM_CCMR1_OC2M_PWM2+TIM_CCMR1_OC2PE)
STR    R2, [R0,#TIM_CCMR1]
MOV    R2, #(TIM_CCER_CC2E+TIM_CCER_CC2NE)
STR    R2, [R0,#TIM_CCER]

MOV    R2, #TIM_EGR_UG
STR    R2, [R0,#TIM_EGR]

MOV    R2, #TIM_BDTR_MOE
STR    R2, [R0,#TIM_BDTR]

;Включить ШИМ
LDR    R2, [R0,#TIM_CR1]
ORR    R2, #TIM_CR1_CEN           ; Счетчик включить
STR    R2, [R0,#TIM_CR1]

;.....

Init_ADC1      ;Инициализация АЦП1 - обычный режим ADC1

; Тактирование АЦП1
MOV32   R0, #RCC
LDR     R1, [R0,#RCC_APB2ENR]
ORR     R1, #RCC_APB2ENR_ADC1EN
STR     R1, [R0,#RCC_APB2ENR]

;Период Тактов АЦП 0,1 мкс
MOV32   R0, #ADC1
MOV     R1, #ADC_CR2_ADON
STR     R1, [R0,#ADC_CR2]
MOV     R1, #ADC_CR1_SCAN
STR     R1, [R0,#ADC_CR1]
MOV     R1, #((ADC_SMPR_SMP13_5 << (ADC_UZ*3))+(ADC_SMPR_SMP13_5 << (ADC_IZ*3))+
(ADC_SMPR_SMP13_5 << (ADC_U0*3))+(ADC_SMPR_SMP13_5 << (ADC_IO*3))) ;Каналы 10, 11, 12, 13 -
выборка: 13,5 цикла
STR     R1, [R0,#ADC_SMPR1]
MOV     R1, #((ADC_SMPR_SMP13_5 << (ADC_U*3))           ;Каналы 1 - выборка: 13,5 цикла
STR     R1, [R0,#ADC_SMPR2]
MOV32   R1, #((ADC_IN1 << 0)+(ADC_IN10 << 5)+(ADC_IN11 << 10)+(ADC_IN13 <<
15)+(ADC_IN12 << 20))           ;Каналы 1: ADCU, 10: ADCIO, 11: ADCIZ, 12: ADCIO, 13: ADCIO
STR     R1, [R0,#ADC_SQR3]
MOV     R1, #(4<<20)           ;5 - Преобразований
STR     R1, [R0,#ADC_SQR1]
LDR     R1, [R0,#ADC_CR2]
MOV32   R2, #(ADC_CR2_DMA + ADC_CR2_EXTTRIG + ADC_CR2_EXTSEL_SWSTART) ;Запуска от
таймера 2
ORR     R1, R2
STR     R1, [R0,#ADC_CR2]

; Сбросить калибровку
LDR     R1, [R0,#ADC_CR2]
ORR     R1, R1, #ADC_CR2_RSTCAL
STR     R1, [R0,#ADC_CR2]
;Ожидание сброса калибровки
Init_ADC1_RSTCAL
LDR     R1, [R0,#ADC_CR2]
TST     R1, #ADC_CR2_RSTCAL
BNE     Init_ADC1_RSTCAL
; ADC1 Калибровка
LDR     R1, [R0,#ADC_CR2]

```

```

        ORR    R1, R1, #ADC_CR2_CAL
        STR    R1, [R0, #ADC_CR2]
        ;Ожидание конца калибровки
Init_ADC1_CAL
        LDR    R1, [R0, #ADC_CR2]
        TST    R1, #ADC_CR2_CAL
        BNE    Init_ADC1_CAL

        LDR    R1, [R0, #ADC_CR2]
        ORR    R1, R1, #ADC_CR2_ADON        ;1 - ADC1 включить
        STR    R1, [R0, #ADC_CR2]

;.....

Init_DMA1    ;Инициализация DMA1 (Канал 1 по запросу ADC1)

        ; Тактирование DMA1
        MOV32  R0, #RCC
        LDR    R1, [R0, #RCC_AHBENR]
        MOV32  R2, #RCC_AHBENR_DMA1EN
        ORR    R1, R1, R2
        STR    R1, [R0, #RCC_AHBENR]

        ;Инициализация DMA1
        MOV32  R0, #DMA1
        MOV    R1, #5                        ; Количество выборок за период ШИМ
        STR    R1, [R0, #DMA_CNDTR1]
        MOV32  R1, #(ADC1+ADC_DR)           ;Адрес регистра данных ADC1
        STR    R1, [R0, #DMA_CPAR1]
        MOV32  R1, #SRAM_ADC_UI            ;Адрес начало массива в SRAM
        STR    R1, [R0, #DMA_CMAR1]
        MOV    R2, #DMA_IFCR_CHTIF1
        STR    R1, [R0, #DMA_IFCR]
        MOV    R1, #(DMA_CCR1_MSIZE_32+DMA_CCR1_PSIZE_16+DMA_CCR1_MINC+DMA_CCR1_CIRC+
DMA_CCR1_PL_vhigh)
        STR    R1, [R0, #DMA_CCR1]
        LDR    R1, [R0, #DMA_CCR1]
        ORR    R1, R1, #DMA_CCR1_EN ; Включение DMA1
        STR    R1, [R0, #DMA_CCR1]

;.....

Init_SRAM    ;Инициализация SRAM (Обнуление)

        MOV32  R0, #SRAM_Control0
        MOV    R12, #0x400

Init_SRAM_clk    ;Обнуление
        STR    Noll, [R0], #4
        SUBS   R12, #1
        BNE    Init_SRAM_clk

;.....

Init_EXTI    ;Инициализация Внешнее прерывание

        MOV32  R0, #EXTI
        MOV32  R1, #(EXTI_PR_PR6+EXTI_PR_PR2) ;Внешнее прерывание Сброс
        STR    R1, [R0, #EXTI_PR]
        MOV32  R1, #EXTI_FTSR_TR6 ;Внешнее прерывание Падающий фронт SB_OFF
        STR    R1, [R0, #EXTI_FTSR]
        MOV32  R1, #EXTI_RTSR_TR2 ;Внешнее прерывание Нарастающего фронт Кнопки SB_ON
        STR    R1, [R0, #EXTI_RTSR]
        MOV32  R1, #(EXTI_IMR_MR2+EXTI_IMR_MR6) ;Внешнее прерывание Маски Включаем
прерывание
        STR    R1, [R0, #EXTI_IMR]

```

```

;Разрешение глобального прерывания EXTI2_IRQn = 8, EXTI9_5_IRQn = 23
MOV32 R0, #NVIC_ICPR0 ; Сбросить признаки отложенных прерываний 0...32
MOV R1, #-1
STR R1, [R0]
MOV32 R0, #NVIC_ISER0 ;Разрешает внешние прерывания 0...32
MOV32 R1, #((1<<EXTI9_5_IRQn)+(1<<EXTI2_IRQn)) ;
STR R1, [R0]

```

;.....

Init\_Zader\_onRL ; Задержка разрешения включения силового реле зарядки конденсаторов

```

MOV32 R12, #10000000
Init_Zader_onRL_ck1 ;Цикл 5 тактов
NOP
SUBS R12,R12,#1
BNE Init_Zader_onRL_ck1

```

```

MOV32 R0, #GPIOA ; Регистр выходных данных порта GPIO
MOV R1, #ROG_PA5 ;Включения силового реле зарядки конденсаторов
STR R1, [R0,#GPIO_BSRR]

MOV Rsos, Noll
ORR Rsos, #Rsos_ROG

```

;.....

Init\_TIM2 ;Инициализация TIM2: Тпер - Период ШИМ Uвх

; Тактирование TIM2

```

MOV32 R0, #RCC
LDR R1, [R0,#RCC_APB1ENR]
MOV32 R2, #RCC_APB1ENR_TIM2EN
ORR R1, R1,R2
STR R1, [R0,#RCC_APB1ENR]

```

;Инициализация TIM2

```

MOV32 R0, #TIM2
MOV R1, #1199 ; Тпер/2 = 15 мкс, при f = 80 МГц
STR R1, [R0,#TIM_ARR] ; TIM регистр авто-перезагрузки
MOV R2, #0
SUB R1, R1,R2
STR R1, [R0,#TIM_CCR3]
STR R2, [R0,#TIM_CCR4]

```

```

MOV R2, #625
STR R2, [R0,#TIM_CCR2]
STR Noll, [R0,#TIM_CCR1]
STR Noll, [R0,#TIM_PSC] ; TIM предделитель

```

```

MOV R2, #(TIM_CR1_CMS_3+TIM_CR1_ARPE);
STR R2, [R0,#TIM_CR1] ; TIM управляющий регистр 1
MOV R2, #TIM_CR2_MMS_OC2REF
STR R2, [R0,#TIM_CR2] ; TIM управляющий регистр 2

```

```

MOV R2, #(TIM_CCMR2_OC3M_PWM2 + TIM_CCMR2_OC3PE + IM_CCMR2_OC4M_PWM1 +
TIM_CCMR2_OC4PE)
STR R2, [R0,#TIM_CCMR2]
MOV R2, #TIM_CCMR1_OC2M_REF0+TIM_CCMR1_OC2PE
STR R2, [R0,#TIM_CCMR1]
MOV R2, #(TIM_CCER_CC3E + TIM_CCER_CC4E)
STR R2, [R0,#TIM_CCER]

MOV R2, #TIM_DIER_CC1IE
STR R2, [R0,#TIM_DIER]

```

```

MOV    R2, #TIM_EGR_UG;
STR    R2, [R0,#TIM_EGR]

MOV    R2, #TIM_BDTR_MOE;
STR    R2, [R0,#TIM_BDTR]

```

```

;Включить ШИМ
LDR    R2, [R0,#TIM_CR1]
ORR    R2, #TIM_CR1_CEN ; Счетчик включить
STR    R2, [R0,#TIM_CR1]

```

*;Разрешение глобального прерывания TIM2\_IRQ = 28 - любое прерывание от модуля TIM2*

```

MOV32  R0, #NVIC_ICPR0 ; Сбросить признаки отложенных прерываний 0...32
MOV    R1, #-1
STR    R1, [R0]
MOV32  R0, #NVIC_ISER0 ;Разрешает внешние прерывания 0...32
MOV    R1, #(1<<TIM2_IRQn)
STR    R1, [R0]

```

*;.....*

```

Init_IWDG ;Инициализация сторожевого таймера
          ;Время охраны: 0,1 мСек
MOV32    R0, #IWDG
MOV32    R1, #IWDG_KR_Start ;Запуск IWDG
STR      R1, [R0,#IWDG_KR]
MOV32    R1, #IWDG_KR_KEY ;Ключ для допуска к регистрам IWDG_PR и IWDG_RLR
MOV      R2, #2
STR      R1, [R0,#IWDG_KR]
STR      R2, [R0,#IWDG_RLR]
STR      R0, [R0,#IWDG_KR]

```

*;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;*

```

Duty      ; Дежурный цикл Мигание светодиода
MOV32     R12, #3000000

```

```

Duty_HL_ck1
SUBS     R12,#1
BNE      Duty_HL_ck1
; Мигание светодиода
MOV32    R0, #GPIOB ; Регистр выходных данных порта GPIO
LDR      R1, [R0,#GPIO_ODR]
EOR      R1, #GPIO_ODR_ODR2
STR      R1, [R0,#GPIO_ODR]

```

```

Duty_Indic_U_I
MOV32    R0, #GPIOB ; Регистр выходных данных порта GPIO
MOV      R1, #(U_PB3+I_PB4)
STR      R1, [R0,#GPIO_BRR]

TST      Rsos, #Rsos_Indic_U_I ; Флаг Rsos: Индикация стабилизации: 0 -
Напряжение, 1 - Ток
MOVEQ    R1, #U_PB3 ; "1" - Индикация стабилизации напряжения

MOVNE    R1, #I_PB4 ; "1" - Индикация стабилизации тока
STR      R1, [R0,#GPIO_BSRR]

```

```

Duty_Indic_Out_Zad
MOV32    R0, #GPIOA
LDR      R1, [R0,#GPIO_IDR]

TST      R1, #SB_ON_PA2 ; PA2 - "0", ВКЛ Выхода

```

```

MOV32 R0, #GPIOB
LDR R1, [R0,#GPIO_IDR]

TSTNE R1, #SB_OFF_PB6 ; PB6 - "0", ОТКЛ Выхода
MOV32EQ R0, #SRAM_Uzad
MOV32NE R0, #SRAM_Uout

LDRD R2, R3,[R0]
MOV32 R1, #SRAM_Uind
STRD R2, R3,[R1]
BL Bit_Dec_Seg

B Duty

ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Inter_TIM2 align
PROC ;Немаскированное прерывание от канала TIM2

PUSH {LR}
PUSH {R4-R8}

MOV32 R0, #NVIC_ICPR0 ; Сбросить признаки отложенных прерываний 0...32
MOV R1, #(1<<TIM2_IRQn)
STR R1, [R0]

MOV32 R0, #TIM2
STR Noll, [R0,#TIM_SR] ; Очистка состояния TIM2

MOV32 R1, #GPIOB ; Регистр выходных данных порта B
MOV32 R3, #Control_PB9
STR R3, [R1,#GPIO_BSRR]

;Включить ШИМ
MOV32 R0, #TIM2
LDR R2, [R0,#TIM_CR1]
ORR R2, #TIM_CR1_CEN ; Счетчик включить
TST Rsos, #Rsos_ROG
STRNE R2, [R0,#TIM_CR1]

Inter_TIM2_Ind_res
UBFX R5, Rsos,#Rsos_Tind,#4 ;Извлечение Rsos_Tind
ADD R5, #1
ANDS R5, #3
BFI Rsos, R5,#Rsos_Tind,#4
BNE Inter_TIM2_ADC

;Гашение разрядов индикатора
MOV32 R1, #GPIOA ; Регистр выходных данных порта A
MOV32 R3, #(PQE_PA12+PQD_PA11+PQC_PA10+PQB_PA9+PQA_PA8)
STR R3, [R1,#GPIO_BRR]

MOV32 R1, #GPIOB ; Регистр выходных данных порта B
MOV32 R3, #(PQCC4_PB15+PQCC3_PB14+PQCC2_PB13+PQCC1_PB12)
STR R3, [R1,#GPIO_BRR]

MOV32 R1, #GPIOC ; Регистр выходных данных порта C
MOV32 R3, #(PQH_PC12+PQG_PC11+PQF_PC10+PQCC8_PC9+PQCC7_PC8+PQCC6_PC7+PQCC5_PC6)
STR R3, [R1,#GPIO_BRR]

Inter_TIM2_ADC
MOV32 R1, #SRAM_ADC_UI ;Адрес начало массива в SRAM
LDM R1,{R2,R3,R4,R5,R6}

```

```

;R2: ADCU
LSL    R2, R2, #1          ;R2: ADCU <- ADCU*2

```

```

;R3: ADCUZ
LSL    R3, R3, #2          ;ADCUZ*4
;R4: ADCIZ
LSL    R4, R4, #2          ;ADCIZ*4
;R5: ADCUO

```

```

;Коррекция накрояжения UO
MOV    R12, #40
MUL    R8, R5, R12
MOV    R12, #4000
UDIV   R8, R8, R12
SUB    R5, R8

```

```

SUB    R5, #2
USAT   R5, #16, R5

```

```

LSL    R5, R5, #2          ;ADCUO*4

```

```

;R6: ADCIO
;Коррекция тока IO
MOV    R12, #6
MUL    R8, R5, R12
MOV    R12, #4000
UDIV   R8, R8, R12
SUB    R6, R8

```

```

SUB    R6, #0
USAT   R6, #16, R6

```

```

MOV    R12, #100
MUL    R8, R6, R12
MOV    R12, #4000
UDIV   R8, R8, R12
ADD    R6, R8

```

```

LSL    R6, R6, #2          ;ADCIO*4

```

```

;Запуск АЦП
MOV32  R0, #ADC1
LDR    R1, [R0, #ADC_CR2]
ORR    R1, R1, #ADC_CR2_SWSTART
STR    R1, [R0, #ADC_CR2]

```

```

;.....

```

```

;Защита от превышения напряжения

```

```

MOV32  R0, #TIM2
TST    Rsos, #Rsos_ROG
BEQ    Inter_TIM2_Aver

```

```

SUB    R8, R2, R5          ;ADCU - ADCUO > 26V
MOV    R1, #2100*4
CMP    R8, R1
MOVGE  R8, #(TIM_CCMR2_OC3M_REF0 + TIM_CCMR2_OC3PE + TIM_CCMR2_OC4M_REF0 +
TIM_CCMR2_OC4PE)
MOVMI  R8, #(TIM_CCMR2_OC3M_PWM2 + TIM_CCMR2_OC3PE + TIM_CCMR2_OC4M_PWM1 +
TIM_CCMR2_OC4PE)
STR    R8, [R0, #TIM_CCMR2]

```

```

;.....

```

Inter\_TIM2\_Aver ;Усреднение Тока, Напряжения

Nusred EQU 6 ;Количество выборок усреднения

;Счет усреднения

MOV32 R0, #SRAM\_IU\_count ;Счет усреднения

LDR R1, [R0]

ADD R1, #1

CMP R1, #Nusred

MOVEQ R1, No11

STR R1, [R0]

;Усреднение Напряжения Заданное

MOV32 R0, #SRAM\_Uz

LDR R7, [R0, R1, LSL #2]

STR R3, [R0, R1, LSL #2] ;ADCUZ

MOV32 R0, #SRAM\_Uz\_summ

LDR R8, [R0]

ADD R8, R3

SUB R8, R7

STR R8, [R0]

MOV R12, #Nusred

SDIV R3, R8, R12

ASR R7, R3, #2

MOV32 R0, #SRAM\_Uzad

STR R7, [R0] ;Uzad

;Усреднение Тока Заданное

MOV32 R0, #SRAM\_Iz

LDR R7, [R0, R1, LSL #2]

STR R4, [R0, R1, LSL #2] ;ADCIZ

MOV32 R0, #SRAM\_Iz\_summ

LDR R8, [R0]

ADD R8, R4

SUB R8, R7

STR R8, [R0]

MOV R12, #Nusred

SDIV R4, R8, R12

ASR R7, R4, #2

MOV32 R0, #SRAM\_Izad

STR R7, [R0] ;Izad

;Усреднение Напряжения

MOV32 R0, #SRAM\_U

LDR R7, [R0, R1, LSL #2]

STR R5, [R0, R1, LSL #2] ;ADCUO

MOV32 R0, #SRAM\_U\_summ

LDR R8, [R0]

ADD R8, R5

SUB R8, R7

STR R8, [R0]

MOV R12, #Nusred

SDIV R5, R8, R12

ASR R7, R5, #2

MOV32 R0, #SRAM\_Uout

STR R7, [R0] ;Uout

*;Усреднение Тока*

```
MOV32 R0, #SRAM_I
LDR R7, [R0,R1,LSL #2]
STR R6, [R0,R1,LSL #2] ;ADCIO
```

```
MOV32 R0, #SRAM_I_summ
LDR R8, [R0]
ADD R8, R6
SUB R8, R7
STR R8, [R0]
```

```
MOV R12, #Nusred
SDIV R6, R8,R12
```

```
ASR R7, R6,#2
MOV32 R0, #SRAM_Iout
STR R7, [R0] ;Iout_ind
```

;

*;Управление ШИМ*

*;Загрузка PIDreg параметрами*

```
ASR R7, R5,#2
ADD R1, R7,#400 ;ADCUO + 400
```

```
Kp_U EQU 20 ; Пропорциональный коэффициент
Ki_U EQU 3 ; Интегральный коэффициент
Kd_U EQU 0 ; Дифференциальный коэффициент
Del_U EQU 10 ; Делитель
```

```
MOV32 R8, #((Kd_U<<24) + (Ki_U<<16) + (Kp_U<<8) + Del_U)
MOV32 R0, #SRAM_et_Uin
```

BL PIDreg

```
ASRS R2, #6 ; R2 = R2/64
ADDCS R2, Unit
```

Inter\_TIM2\_PWM\_Uin *;Код управления ШИМ - напряжением*

```
USAT R2, #10,R2
MOV R1, #570
CMP R2, R1
MOVHI R2, R1
MOV32 R0, #TIM2
MOV R1, #1199
SUB R1, R1,R2
STR R1, [R0,#TIM_CCR3]
STR R2, [R0,#TIM_CCR4]
```

;

Inter\_TIM2\_Regul\_Ic *;Управление ШИМ по Току*

```
MOV R1, R4 ;ADCIZ
MOV R2, R6 ;ADCIO
```

```
Kp_Iout EQU 100 ; Пропорциональный коэффициент
Ki_Iout EQU 1 ; Интегральный коэффициент
Kd_Iout EQU 0 ; Дифференциальный коэффициент
Del_Iout EQU 20 ; Делитель
```

```
MOV32 R8, #((Kd_Iout<<24) + (Ki_Iout<<16) + (Kp_Iout<<8) + Del_Iout)
MOV32 R0, #SRAM_et_Iout
```

BL PIDreg

PUSH {R2}

;.....

Inter\_TIM2\_Regul\_U ;Управление ШИМ по Напряжению

MOV R1, R3 ;ADCUZ  
MOV R2, R5 ;ADCUO

Kp\_Uout EQU 100 ; Пропорциональный коэффициент  
Ki\_Uout EQU 1 ; Интегральный коэффициент  
Kd\_Uout EQU 0 ; Дифференциальный коэффициент  
Del\_Uout EQU 10 ; Делитель

MOV32 R8, #((Kd\_Uout<<24) + (Ki\_Uout<<16) + (Kp\_Uout<<8) + Del\_Uout)  
MOV32 R0, #SRAM\_et\_Uout

BL PIDreg

;Код управления ШИМ - напряжением  
MOV R5, R2

;Код управления ШИМ - током  
POP {R4}

;.....

CMP R4, R5  
MOVL R2, R4 ; R4 <= R5  
ORRL Rsos, #Rsos\_Indic\_U\_I ; Флаг Rsos: Индикация стабилизации: 1 - Ток  
MOVGT R2, R5 ; R4 > R5  
BICGT Rsos, #Rsos\_Indic\_U\_I ; Флаг Rsos: Индикация стабилизации: 0 -

Напряжение

USAT R4, #14, R2  
MOV32 R0, #TIM1  
STR R2, [R0, #TIM\_CCR2]

;.....

Inter\_TIM2\_Ind

UBFX R4, Rsos, #Rsos\_Tind, #4 ;Извлечение Rsos\_Tind  
TEQ R4, No11  
BNE Inter\_TIM2\_exit

;Включение разрядов индикатора

UBFX R5, Rsos, #Rsos\_QCCn, #4 ;Извлечение Номера разряда Индикатора  
ADD R5, #1  
AND R5, #7  
BFI Rsos, R5, #Rsos\_QCCn, #4

MOV32 R0, #SRAM\_Ind\_seg  
LDRB R4, [R0, R5]

MOV32 R0, #Ind\_QCCn  
LDRB R5, [R0, R5]

MOV32 R0, #GPIOA ; Регистр выходных данных порта A  
MOV R3, No11  
BFI R3, R4, #8, #5 ; PQE\_PA12+PQD\_PA11+PQC\_PA10+PQB\_PA9+PQA\_PA8  
STR R3, [R0, #GPIO\_BSRR]

MOV32 R1, #GPIOB ; Регистр выходных данных порта B  
MOV R3, No11

```

BFI    R3, R5, #12, #4      ; PQCC4_PB15+PQCC3_PB14+PQCC2_PB13+PQCC1_PB12
STR    R3, [R1, #GPIO_BSRR]

MOV32  R2, #GPIOC           ; Регистр выходных данных порта C
UBFX   R3, R5, #4, #4      ; PQCC8_PC9+PQCC7_PC8+PQCC6_PC7+PQCC5_PC6
LSL    R3, R3, #6
UBFX   R4, R4, #5, #3      ; PQH_PC12+PQG_PC11+PQF_PC10
BFI    R3, R4, #10, #3
STR    R3, [R2, #GPIO_BSRR]

```

#### Inter\_TIM2\_exit

```

;Сторожибой таймер
MOV32  R0, #IWDG
MOV32  R1, #IWDG_KR_Restart ;Перезапуск IWDG - запись
STR    R1, [R0, #IWDG_KR]

MOV32  R1, #GPIOB           ; Регистр выходных данных порта B
MOV32  R3, #Control_PB9
STR    R3, [R1, #GPIO_BRR]

POP     {R4-R8}
POP     {LR}
BX      LR
ENDP

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

PIDreg      PROC      align
;ПИД-Регулятор.
; Вход: R1 - Заданные данные, R2 - Фактические данные, R0 - адрес SRAM_et_
; Выход: R2
PUSH       {R3-R7}

SBFX      R7, R8, #0, #8 ;Del

;Вычисление рассогласования: e(t)
SUB       R3, R1, R2      ; e(t)
;.....
PID_p      ;Пропорциональная составляющая.  $P(t) = K_p * e(t) / Del$ ;
SBFX      R4, R8, #8, #8 ;Kp
MUL       R4, R3
SDIV      R4, R4, R7      ; P(t)
SSAT      R4, #17, R4     ;Ограничение R4
;.....
PID_in     ;Интегральная составляющая.  $I(t) = I(t-1) + K_i * e(t) / Del$ 
SBFX      R5, R8, #16, #8 ;Ki
MUL       R5, R3
SDIV      R5, R5, R7
LDR       R2, [R0, #4]    ; SRAM_It
ADD       R5, R2          ; I(t)
SSAT      R5, #17, R5     ;Ограничение R5
STR       R5, [R0, #4]    ; SRAM_It
;.....
PID_dif    ;Дифференциальная составляющая.  $D(t) = K_d * \{e(t) - e(t-1)\} / Del$ ;
LDR       R2, [R0]        ; SRAM_et: e(t-1)
SUB       R6, R3, R2      ; Вычитание e(t)-e(t-1)
STR       R3, [R0]        ; SRAM_et: e(t)
SBFX      R6, R8, #24, #8 ;Kd
MUL       R6, R2          ; D(t)
SDIV      R6, R6, R7
SSAT      R6, #17, R6     ;Ограничение R6
;.....
PID_Dut    ;Выход:  $u(t) = P(t) + I(t) + D(t)$ ;
ADD       R2, R4, R5
SUB       R2, R2, R6      ; u(t)

```

```
SSAT    R2, #17,R2
```

```
POP     {R3-R7}
```

```
BX      LR
ENDP
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
Inter_sb_Off      align
PROC              ;Прерывание от энкодера EXTI6
```

```
PUSH    {LR}
```

```
MOV32    R0, #NVIC_ICPR0      ; Сбросить признаки отложенных прерываний 0...32
MOV32    R1, #(1<<EXTI9_5_IRQn)
STR      R1, [R0]
```

```
MOV32    R0, #EXTI
MOV32    R1, #(EXTI_PR_PR6+EXTI_PR_PR2) ;Внешнее прерывание Сброс
STR      R1, [R0,#EXTI_PR]
```

```
MOV32    R0, #GPIOB
MOV      R1, #OUT_ONR_PB7
STR      R1, [R0,#GPIO_BRR]      ; ОТКЛ реле и индикация Выхода
```

```
Inter_sb_Off_exit
```

```
POP      {LR}
BX       LR
ENDP
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
Inter_sb_On      align
PROC              ; Прерывание EXTI2_IRQHandler от Кнопки Энкондера PA2
```

```
PUSH    {LR}
```

```
MOV32    R0, #NVIC_ICPR0      ; Сбросить признаки отложенных прерываний 0...32
MOV      R1, #(1<<EXTI2_IRQn)
STR      R1, [R0]
```

```
MOV32    R0, #EXTI
MOV32    R1, #(EXTI_PR_PR6+EXTI_PR_PR2) ;Внешнее прерывание Сброс
STR      R1, [R0,#EXTI_PR]
```

```
MOV32    R0, #GPIOB
MOV      R1, #OUT_ONR_PB7
STR      R1, [R0,#GPIO_BSRR]      ; ВКЛ реле и индикация Выхода
```

```
Inter_sb_On_exit
```

```
POP      {LR}
BX       LR
ENDP
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
align
```

```
Bit_Dec_Seg      PROC              ;Преобразование 2 -> 10
                  ;Результат в SRAM_Ind_seg
                  PUSH    {LR}
```

```
MOV32    R0, #SRAM_Indic ; Адрес (младший разряд) области памяти 8-разрядного
десятичного числа
MOV      R12, #8
```

Bit\_Dec\_Seg\_cN

```

; Заполнением нулями
STRB  No11, [R0],#1
SUBS  R12, R12,Unit
BNE  Bit_Dec_Seg_cN

;Напряжение Преобразование 2 -> 10
MOV32 R0, #SRAM_Indic
MOV32 R1, #SRAM_Uind
LDR   R2, [R1]
MOV   R12, #1250 ;Масштабирование
MUL   R2, R12
MOV   R12, #1000
UDIV  R2, R12

```

BL Form\_Dec

```

;Ток Преобразование 2 -> 10
MOV32 R0, #(SRAM_Indic+4)
MOV32 R1, #SRAM_Iind
LDR   R2, [R1]

```

BL Form\_Dec

```

; Преобразование 10 -> Сегментный код
MOV   R12, #8
MOV32 R0, #SRAM_Indic
MOV32 R1, #Dec_Seg
MOV32 R2, #SRAM_Ind_seg

```

Bit\_Dec\_Seg\_clr

```

LDRB  R3, [R0],#1
LDRB  R3, [R1,R3]
STRB  R3, [R2],#1
SUBS  R12, R12,#1
BNE  Bit_Dec_Seg_clr

```

```

;Включение запятой
MOV32 R0, #SRAM_Ind_seg
LDRB  R2, [R0,#2]
ORR   R2, #0x80
STRB  R2, [R0,#2]

LDRB  R2, [R0,#7]
ORR   R2, #0x80
STRB  R2, [R0,#7]
;Гашение разряда, если в нем "0"
LDRB  R2, [R0,#3]
CMP   R2, #0x3f ;Сегм.: "0"
STRBEQ No11, [R0,#3]

```

```

POP   {LR}
BX    LR

```

ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

align

```

Form_Dec      PROC      ;Преобразование 2 -> 10 с форматированием
                  ;R2 - загрузка двоичного числа
                  ;Результат в [R0]
USAT  R2, #14,R2      ;Ограничение 0 =< r2 =< 8191
MOV   R12, #10

```

Bit\_Dec\_cd

```

UDIV  R3, R2,R12      ; R3 = R2/R12.
MLS   R4, R3,R12,R2   ; R4 = R2 - (R3 * R12) = остаток R4

```

```

MOV    R2, R3
STRB   R4, [R0],#1

CMP    R3, #0      ;Проверка на 0
BNE    Bit_Dec_cd

BX     LR

ENDP

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

AREA CONST, DATA, READONLY, align=2

```

```

;*****

```

```

;Массив форматов чисел. Цифры 0...9 -> Сегментный код

```

```

Dec_Seg

```

```

    DCB    0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x00

```

```

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

```

```

Ind_QCCn

```

```

    DCB    0x02,0x01,0x08,0x04,0x20,0x10,0x80,0x40,0x00

```

```

;*****

```

```

    END

```