

## Блок питания паяльника

Автор: А.В. Жмаев, E-mail: 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. Некоторые скажут - зачем такая сложность, воткнул паяльник в розетку и паяй. Проще, если паяльник 220В, а если нет. Далее:

- развязка от сети;
- при отсутствии датчика температуры, можно регулировать нагрев, опираясь на индикацию мощности;
- естественно, стабилизация напряжения.

В данной конструкции микроконтроллер АТМЕГА328Р работает на частоте 25МГц и тактировка АЦП около 400кГц.

Дополнительная литература:

1. АТmega328\_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. Блок питания лабораторный [https://disk.yandex.ru/i/\\_zjR5FvHzJnZkQ](https://disk.yandex.ru/i/_zjR5FvHzJnZkQ)
8. Аппарат сварочный САБП <https://disk.yandex.ru/i/DSN21ss7YVzVZQ>

## Технические данные и описание



Рис.1. Блок питания паяльника

1. Выходное напряжение: 20 ... 250 В.
2. Ток ограничения, макс.: 1,2 А.
3. Мощность нагрузки, макс.: 100 Вт.
4. Питание блока: 220 В.

Установка напряжения производится вращением энкодера.

Контроль напряжения и мощности производится по индикатору  $U_{\phi}$  и  $P_{\phi}$ .

## Описание конструкции

Общий вид бока показан на Рис.1.

Особой эстетикой не замарачивался. За основу корпуса взят корпус аккумулятора (из экономии), а остальные детали напечатаны на 3D-принтере.

Схема блока классическая, полу-мостовая и показана на Рис.2.

Комплектация: что была под рукой.

Тактовая частота микроконтроллера ATMEGA328P повышена до 25 мГц, что увеличивает динамический диапазон ШИМ и тактовая частота АЦП повышена до 400 кГц. Время преобразования АЦП соответствует – 33,3 мкс.

Формирование двух-полярного ШИМ производится программным методом по прерыванию «Переполнение» Таймера 1.

Трансформатор TV2 намотан на сердечнике ГМ40ВДС К40х25х10:

$\mu = 3000$ ,

$V = 1\text{Тл}$ .

Провод на первичной обмотке: ПЭВ 0,51, на вторичной: ПЭВ 0,43

Дроссель L2, применен от старого монитора «катушечного» типа 620 мкГн, 2 А.

Радиатор на транзисторы VT1 и VT2 применен пластинчатый радиатор 200 см<sup>2</sup>.

Диоды VD5 ... VD7, минусовыми выводами напаяны на медную пластину 0,3 мм площадью 10 см<sup>2</sup>.

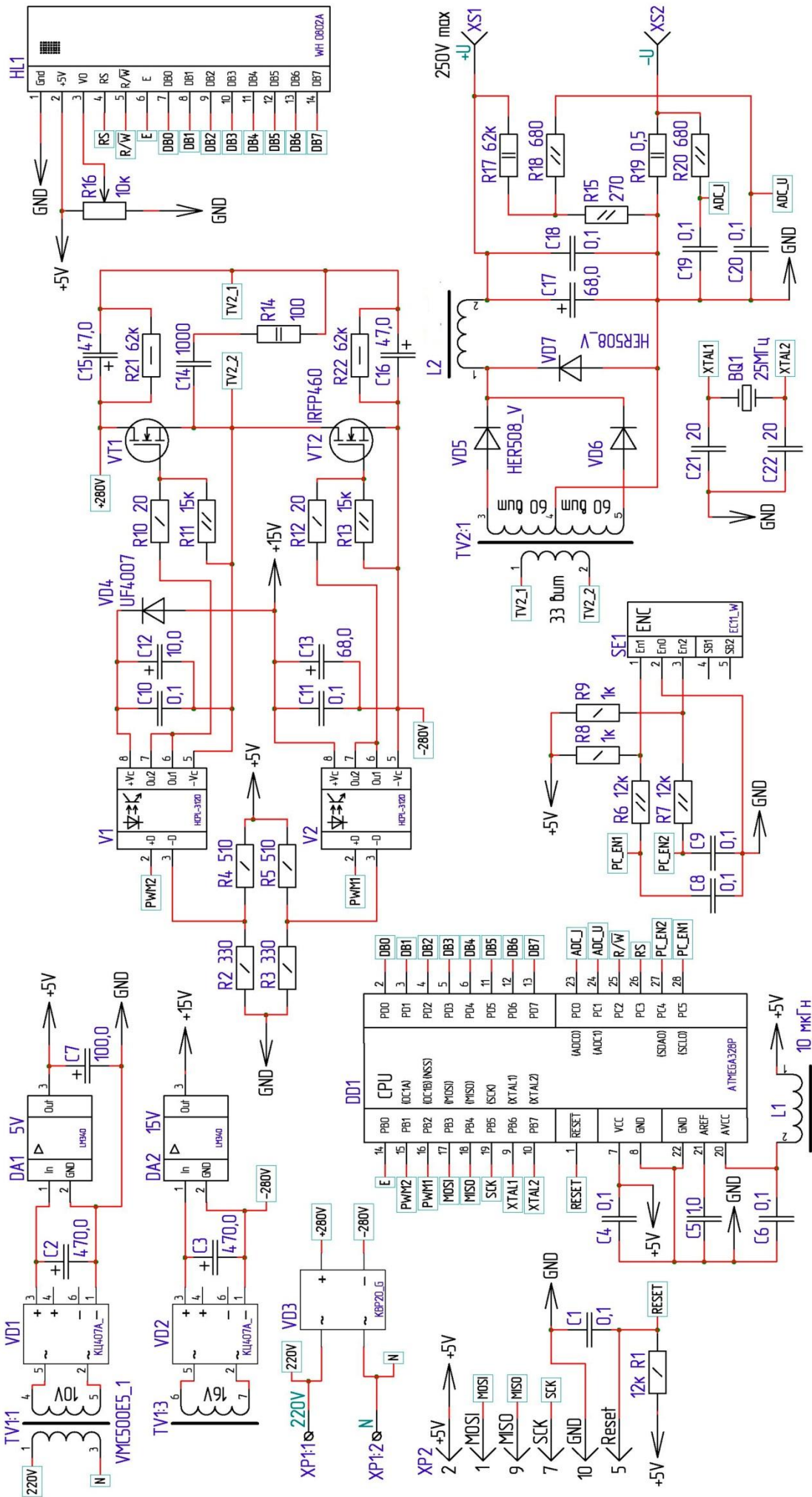


Рис.2. Плата аналоговая Силовая БП вр

## Программное обеспечение блока.

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

```
; Bock_Pitaniy_Pylnika.asm
;
; Created: 12.07.2022 14:30:19
; Author : ADMIN
; cpu:    ATmega328P
; speed:  25 mhz
; voltage: 5 V
; rom:    32767 bytes (0x0000-0x7fff)
; ram:    2048 bytes (0x0000-0x07ff)
; eeprom: 1024 bytes (0x0060-0x045f)
;
;CKDIV8 - 0
;CKOUT - 1
;SUT1 -1, SUT0 - I
;CKSEL3 - 0, CKSEL2 - 1, CKSEL1 - 1, CKSEL0 - 1
;-----
        .cseg
        .org 0
;Начало

Reset:

JMP BEGIN ;Reset
JMP Reset ;IRQ0
JMP Reset ;INTI ;IRQ1
JMP Reset ;PCINT0 ;PCINT0
JMP INT0_PCINT13 ;PCINT1 ;PCINT1

JMP Reset ;PCINT2 ;PCINT2
JMP Reset ;WDT ;Watchdog Timeout
JMP Reset ;TIM2 COMPA ;Timer2 CompareA
JMP Reset ;TIM2 COMPB ;Timer2 CompareB
JMP Reset ;TIM2 OVF ;Timer2 Overflow
JMP Reset ;TIM1 CAPT ;Timer1 Capture
JMP Reset ;TIM1_COMPA ;Timer1 CompareA
JMP Reset ;TIM1 COMPB ;Timer1 CompareB
JMP TIM1_OVF ;Timer1 Overflow
JMP Reset ;TIM0 COMPA ;Timer0 CompareA
JMP Reset ;TIM0 COMPB ;Timer0 CompareB
JMP Reset ;TIM0_OVF ;Timer0 Overflow
JMP Reset ;SPI_STC ;Передача SPI Завершена
JMP Reset ;USART RXC ;USART RX Complete
JMP Reset ;USART UDRE ;USART UDR Empty
JMP Reset ;USART TXC ;USART TX Complete
JMP ADC_END ;ADC Преобразование Завершено
JMP Reset ;EE_RDY ;EEPROM Ready
JMP Reset ;ANA_COMP ;Analog Comparator 0x0030
JMP Reset ;TWI ;2-проводной последовательный
JMP Reset ;SPM_RDY ;SPM Ready
;-----
defR:
        .def MuLL = r0
        .def MuLH = r1

        .def Amount_IL = r2
        .def Amount_IH = r3
        .def Amount_UL = r4
        .def Amount_UH = r5
        .def Ifak = r6
        .def Ufak = r7
```

```

.def D_W = r8
.def Radmux = r9

;.....
.def Sch_c = r10 ;Счетчик циклов для подпрограммы del_16_16
;.....
.def T1_A_B = r11 ;0/1
.def etL = r12 ;Расогласование e(t-1)
.def etH = r13
.def ItL = r14 ;Интегральная составляющая (t-1).
.def ItH = r15
.def AnL = r16 ;Множитель
.def AnH = r17
.def DL = r18 ;Произведение
.def DH = r19
.def CL = r20 ;Произведение
.def CH = r21
.def BL = r22 ;Множимое
.def BH = r23
.def AL = r24 ;Выход ПИД.
.def AH = r25

;.....
defP: ;Привязка к схеме
.equ E = 0 ; PORTB PB0 Запись в индикатор "1"
.equ PWM2 = 1 ; PORTB PB1 Выход ШИМ ТС1
.equ PWM1 = 2 ; PORTB PB2 Выход ШИМ ТС1

.equ R_W = 2 ; PORTC PC2 выбор Чтение"1"/запись"0"
.equ RS = 3 ; PORTC PC3 Выбор данных"1"/инструкции"0"

.equ PC_EN2 = 4 ; PORTC PC4 от энкодера
.equ PC_EN1 = 5 ; PORTC PC5

.equ nIU = 0 ; Radmux[0]: Канал I = 0, Канал U = 1
;-----

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

;-----

Inic_PORTD: ;Инициализация порта D
LDI AL,0xff ;Выход.
OUT DDRD,AL
LDI AL,0
OUT PORTD,AL

;-----

Inic_PORTB: ;Инициализация порта B
LDI AL,(1<<E) ;Выход.
OUT DDRB,AL

;-----

Inic_PORTC: ;Инициализация порта C
LDI AL,(1<<RS)|(1<<R_W) ;Выход.
OUT DDRC,AL
LDI AL,0
OUT PORTC,AL

;-----

Inic_Const:
LDI AL,0

;Обнуление области ОЗУ: 0x400 ... 0x100
LDI XL,LOW(0x300) ;

```

```

        LDI    XH,HIGH(0x300)
        LDI    YL,LOW(0x100)      ;
        LDI    YH,HIGH(0x100)

Ck1_Clr:
        ST     Y,AL
        ADIW   YL,1
        SBIW   XL,1
        BRNE   Ck1_Clr

        LDI    AL,20*4            ; U zad = 20
        STS    dseg_Uzad_min,AL
        STS    dseg_Uzad,AL

        LDI    AL,LOW(250*4)      ; U zad = 250
        LDI    AH,HIGH(250*4)
        STS    dseg_Uzad_max,AL
        STS    dseg_Uzad_max+1,AH

        CLR    XL
        CLR    Amount_IL
        CLR    Amount_IH
        CLR    Amount_UL
        CLR    Amount_UH

        CLR    etL
        CLR    etH

        CLR    ItL
        CLR    ItH

;-----

        CALL   Init_Timer1

        CALL   Inic_INT0_PCINT13

        CALL   Inic_ADC

;-----
Inic_Indic:                                ;Инициализация индикатора
        LDI    ZL,LOW(Indic_Tab_Init*2)    ;Начальный адрес таблицы инициализации
индикатора
        LDI    ZH,HIGH(Indic_Tab_Init*2)
        RCALL  Indic_W_cseg                ;Перезапись из Памяти в Индикатор
;-----

Indic_Begin:
        LDI    ZL,LOW(Indic_Tab_Begin*2) ;Начальный адрес таблицы Начала программы
        LDI    ZH,HIGH(Indic_Tab_Begin*2)
        RCALL  Indic_W_cseg                ;Перезапись из Памяти в Индикатор
;-----

        LDI    AL,(1<<E)|(1<<PWM1)|(1<<PWM2) ;Включение выходов ШИМ.
        OUT    DDRB,AL
;-----

        SEI    ; Разрешение прерываний
;-----

whileP:
        ;Задержка для цикла вывода на индикатор
        LDI    AnL,100

whileP_zad:
        CALL   Zader_1500
        DEC    AnL
        BRNE   whileP_zad

whileP_Pf: ; Вывод на индикатора параметр мощности
        LDI    ZL,LOW(LIN1_Pf*2)          ;

```

```

        LDI    ZH,HIGH(LIN1_Pf*2)
        LDS    DL,dseg_Pfak
        LDS    DH,dseg_Pfak+1
whileP_P_ind:
        LDI    YL,LOW(dseg_Indic)        ;
        LDI    YH,HIGH(dseg_Indic)

whileP_Pf_cd:
        LPM    AnL,Z+
        ST     Y+,AnL
        TST    AnL
        BRNE  whileP_Pf_cd

whileP_P:
        RCALL  Int_2_Z10

whileP_P_2:
        LDS    AL,dseg_Char_cod+4
        CPI    AL,0x30
        BREQ  whileP_P_1

        STS    dseg_Indic+6,AL
        LDS    AL,dseg_Char_cod+3
        STS    dseg_Indic+7,AL
        RJMP  whileP_P_0

whileP_P_1:
        LDS    AL,dseg_Char_cod+3
        LDI    AnL,0x30
        CPSE  AL,AnL
        STS    dseg_Indic+7,AL

whileP_P_0:
        LDS    AL,dseg_Char_cod+2
        STS    dseg_Indic+8,AL

        RCALL  dseg_Ind        ; Перезапись из буферной памяти на индикатор
        RCALL  Zader_1500

whileP_Uf: ; Вывод на индикатор параметра напряжения Uf
        LDI    ZL,LOW(LIN0_Uf*2)        ;
        LDI    ZH,HIGH(LIN0_Uf*2)
        LDS    DL,dseg_Uaver
        LDI    DH,0

whileP_U_ind:
        LDI    YL,LOW(dseg_Indic)        ;
        LDI    YH,HIGH(dseg_Indic)

whileP_U_cd:
        LPM    AnL,Z+
        ST     Y+,AnL
        TST    AnL
        BRNE  whileP_U_cd

whileP_U:
        RCALL  Int_2_Z10

whileP_U_2:
        LDS    AL,dseg_Char_cod+2
        CPI    AL,0x30
        BREQ  whileP_U_1

        STS    dseg_Indic+6,AL
        LDS    AL,dseg_Char_cod+1
        STS    dseg_Indic+7,AL
        RJMP  whileP_U_0

```

```

whileP_U_1:
    LDS    AL,dseg_Char_cod+1
    LDI    AnL,0x30
    CPSE   AL,AnL
    STS    dseg_Indic+7,AL

whileP_U_0:
    LDS    AL,dseg_Char_cod+0
    STS    dseg_Indic+8,AL

    CALL   dseg_Ind      ; Перезапись из буферной памяти на индикатор

    RJMP  whileP

;*****

Indic_W_cseg:      ;Перезапись из Памяти ПЗУ в Индикатор

    LPM    D_W,Z+

    TST    D_W
    BREQ   Indic_W_cseg_cRet

    RCALL  W_CA_DD
    LDI    AL,1
    CPSE   AL,D_W
    RCALL  Zader_1500

    RCALL  Zader_50
    RJMP  Indic_W_cseg

Indic_W_cseg_cRet:

    RET

;*****

dseg_Ind: ; Перезапись из буферной памяти на индикатор

    LDI    ZL,LOW(dseg_Indic)
    LDI    ZH,HIGH(dseg_Indic)
    RCALL  Indic_W_dseg      ;Перезапись из Памяти в Индикатор

    RET

;*****

Indic_W_dseg:      ;Перезапись из Памяти ОЗУ в Индикатор

    LD     D_W,Z+

    TST    D_W
    BREQ   Indic_W_dseg_dRet

    RCALL  W_CA_DD
    RCALL  Zader_50
    RJMP  Indic_W_dseg

Indic_W_dseg_dRet:

    RET

;*****

```

```

W_CA_DD: ;П/п Записи Адреса/Данных в индикатор
;Опознавание кода Адреса
LDI AL,0xCA
CP D_W,AL
BRNE W_CA_DD_d

CBI PORTC,R_W ;выбор Чтение"1"/запись"0"
CLT ;Операция с битом T = 0 в SREG
RET

W_CA_DD_d:
;Опознавание кода Данных
LDI AL,0xDD
CP D_W,AL
BRNE W_CA_DD_ad

SET ;Операция с битом T = 1 в SREG ;SBR RegS,1<<Indic_DD
RET

W_CA_DD_ad:
CBI PORTC,RS ;Выбор данных"1"/инструкции"0"
BRTC W_CA_DD_ad1 ;Операция по биту T в SREG
SBI PORTC,RS ;Выбор данных"1"/инструкции"0"

W_CA_DD_ad1:
NOP
NOP
SBI PORTB,E ;Запись в индикатор - начало
NOP
NOP
OUT PORTD,D_W ;Адрес/Данные в индикатор
NOP
NOP
NOP
NOP
CBI PORTB,E ;Запись в индикатор - конец
NOP
NOP

RET

;*****

Zader_1500: ;Задержка 1550 мкс (при 25 МГц)
LDI AH,100

Zader_1500_ck1:
RCALL Zader_50
DEC AH
BRNE Zader_1500_ck1
RET

;*****

Zader_50: ;Задержка 50 мкс (при 25 МГц)
LDI AL,250

Zader_50_ck1:
NOP
NOP
DEC AL
BRNE Zader_50_ck1
RET

;*****

```



```
SBCI DH,HIGH(100)

LDI AL,HIGH(100) ;100
CPI DL,LOW(100)
CPC DH,AL
BRCC Int_2_Z2_1
```

Int\_2\_Z1:

```
LPM AH,Z
STS dseg_Char_cod+2,AH
```

;

```
CLR ZL ;Частное = 0
LDI AL,HIGH(10) ;10
CPI DL,LOW(10)
CPC DH,AL
BRCS Int_2_Z1_2
```

Int\_2\_Z1\_1:

```
INC ZL ;Частное = Частное + 1
SUBI DL,LOW(10) ;10
SBCI DH,HIGH(10)
```

```
LDI AL,HIGH(10) ;10
CPI DL,LOW(10)
CPC DH,AL
BRCC Int_2_Z1_1
```

Int\_2\_Z1\_2:

```
LPM AH,Z
STS dseg_Char_cod+1,AH
```

;

Int\_2\_Z0:

```
CLR ZL
ADD ZL,DL
LPM AH,Z
STS dseg_Char_cod,AH
```

RET

;

Inic\_INT0\_PCINT13: ; Инициализация прерываия от Энкодера

```
LDI AL,(1<<PCIE1) ; (PCINT [14:8])
STS PCICR,AL
LDI AL,(1<<PCINT13) ; Маска включения смены контактов
STS PCMSK1,AL
```

RET

;

INT0\_PCINT13: ; Прерывание от Энкодера PC\_EN1

```
SEI
PUSH AL
IN AL,SREG
PUSH AL
PUSH AH
PUSH AnL
PUSH AnH

SBIC PINC,PC_EN1
RJMP INT0_PCINT13_exit

LDS AL,dseg_Uzad
```

```

LDS    AH,dseg_Uzad+1
SBIS   PINC,PC_EN2
SBIW   AL,4
SBIC   PINC,PC_EN2
ADIW   AL,4

```

```

;Проверка пределов Uzad

```

```

INT0_PCINT13_max:

```

```

LDS    AnL,dseg_Uzad_max
LDS    AnH,dseg_Uzad_max+1
CP     AL,AnL
CPC    AH,AnH
BRCS   INT0_PCINT13_min

```

```

MOVW   AL,AnL
RJMP   INT0_PCINT13_sts

```

```

INT0_PCINT13_min:

```

```

TST    AH
BRNE   INT0_PCINT13_sts

```

```

LDS    AnL,dseg_Uzad_min
CP     AL,AnL
BRCC   INT0_PCINT13_sts

```

```

MOV    AL,AnL

```

```

INT0_PCINT13_sts:

```

```

STS    dseg_Uzad,AL ; Uzad*4
STS    dseg_Uzad+1,AH

```

```

INT0_PCINT13_exit:

```

```

POP    AnH
POP    AnL
POP    AH
POP    AL
OUT    SREG,AL
POP    AL

```

```

RETI

```

```

;*****

```

```

Init_Timer1:

```

```

;Инициализация Таймер 1

```

```

;Назначение: Формирователь периода ШИМ - получает clk.

```

```

.equ Tper = 832

```

```

.equ tmax = (Tper/2)-20

```

```

LDI AnL,LOW(Tper/2) ;Период прообразования

```

```

LDI AnH,HIGH(Tper/2)

```

```

;ICR1=T_timer1 Переполнение.

```

```

STS ICR1H,AnH

```

```

STS ICR1L,AnL

```

```

LDI AL,LOW(tmax)

```

```

LDI AH,HIGH(tmax)

```

```

SUB AnL,AL

```

```

SBC AnH,AH

```

```

;Вывод ШИМ

```

```

STS OCR1AH,AnH

```

```

STS OCR1AL,AnL

```

```

STS OCR1BH,AnH

```

```

STS OCR1BL,AnL

```

```

;Fast PWM: ICR1 - код 14

```

```

LDI AL,(1<<COM1A1)|(1<<COM1A0)|(1<<COM1B1)|(1<<COM1B0)|(1<<WGM11)

```

```

STS TCCR1A,AL

```

```
LDI AL,(1<<CS10)|(1<<WGM12)|(1<<WGM13)
STS TCCR1B,AL
```

```
CLR T1_A_B
```

```
LDI AL,(1<<TOIE1) ;Разрешить прерывание от переполнения
STS TIMSK1,AL
```

```
RET
```

```
;*****
```

```
TIM1_OVF: ;Прерывание от TIM1_OVF
;Формирование двухполярного ШИМ
```

```
PUSH AL
PUSH AH
PUSH AnL
```

```
LDS AL,dseg_OutPID
LDS AH,dseg_OutPID+1
```

```
SBRC T1_A_B,0
RJMP TIM1_OVF_B
```

```
TIM1_OVF_A: ;Выход A
```

```
STS OCR1AH,AH
STS OCR1AL,AL
LDI AL,LOW(Tper/2) ;Период преобразования
LDI AH,HIGH(Tper/2)
STS OCR1BH,AL
STS OCR1BL,AL
LDI AL,1
MOV T1_A_B,AL
RJMP TIM1_OVF_Txt
```

```
TIM1_OVF_B: ;Выход B
```

```
STS OCR1BH,AH
STS OCR1BL,AL
LDI AL,LOW(Tper/2) ;Период преобразования
LDI AH,HIGH(Tper/2)
STS OCR1AH,AL
STS OCR1AL,AL
LDI AL,0
MOV T1_A_B,AL
```

```
TIM1_OVF_Txt:
```

```
POP AnL
POP AH
POP AL
RETI
```

```
;*****
```

```
Inic_ADC: ;Инициализация АЦП
; Канал: 1.
```

```
LDI AL,(1<<REFS1)|(1<<REFS0) ;Внутреннее значение напряжения 1.1 В с внешним
конденсатором на выводе AREF, Канал I = 0
MOV Radmux,AL
STS ADMUX,Radmux
```

```
LDI AL,(1<<ADC1D)|(1<<ADC0D) ;Запрет цифрового входа
STS DIDR0,AL
```

```
;АЦП включен. Начала преобразования. Автозапуск включен. Прерывание включено.
Предделение:64 (около 400 кГц).
```

```
LDI AL,(1<<ADEN)|(1<<ADSC)|(1<<ADATE)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)
STS ADCSRA,AL
```

```
RET
```

```
;*****
```

```
ADC_END: ;Прерывание от ADC - конец преобразования Такты: Tmaxc = 555
```

```
SEI
PUSH AL
IN AL,SREG
PUSH AL
PUSH AH
PUSH DL
PUSH DH
PUSH AnL
PUSH AnH
```

```
LDS AL,ADCSRA
SBR AL,1<<ADIF
STS ADCSRA,AL
```

```
LDS AnL,ADCL ;Перезапись ADC в память
LDS AnH,ADCH
```

```
;UI = ADC*2
ROL AnL
ROL AnH
```

```
SBRC Radmux,nIU
RJMP ADC_END_I
```

```
;.....31
```

```
ADC_END_U:
```

```
;U - коррекция
```

```
.equ KkorU = 63014/2
```

```
LDI BL,LOW(KkorU)
LDI BH,HIGH(KkorU)
RCALL Mu1s16x16_32 ;Частное в DL
STS dseg_ADC_U,DL
STS dseg_ADC_U+1,DH
```

```
;U = ADC/4
ASR DH
ROR DL
ASR DH
ROR DL
MOV Ufak,DL
```

```
RJMP ADC_END_Aver_I
```

```
;.....36
```

```
ADC_END_I:
```

```
;I - коррекция
```

```
.equ KkorI = 63896/2
```

```
LDI BL,LOW(KkorI)
LDI BH,HIGH(KkorI)
RCALL Mu1s16x16_32 ;Частное в DL
STS dseg_ADC_I,DL
STS dseg_ADC_I+1,DH
MOVW AnL,DL
```

```
;I = ADC/5
LDI BL,LOW(13107)
LDI BH,HIGH(13107)
RCALL Mu1s16x16_32 ;Частное в DL (DH:DL:CH:CL = BH:BL * AnH:AnL)
MOV Ifak,DL
```

```
;.....58
```

```

ADC_END_Aver_I:
    ;Усреднение I по 256 выборкам
    LDI  XH,HIGH(dseg_Aver_I)
    LD   AL,X
    ST   X,Ifak
    ADD  Amount_IL,Ifak
    CLR  AH
    ADC  Amount_IH,AH
    SUB  Amount_IL,AL
    SBC  Amount_IH,AH

    STS  dseg_Iaver,Amount_IH

;.....13
ADC_END_Aver_U:
    ;Усреднение U по 256 выборкам
    LDI  XH,HIGH(dseg_Aver_U)
    LD   AL,X
    ST   X,Ufak
    ADD  Amount_UL,Ufak
    CLR  AH
    ADC  Amount_UH,AH
    SUB  Amount_UL,AL
    SBC  Amount_UH,AH

    STS  dseg_Uaver,Amount_UH

    INC  XL

;.....14
ADC_END_P:
    ;P = I * U - вычисление мощности.
    LDS  AL,dseg_Iaver
    LDS  AH,dseg_Uaver
    MUL  AL,AH ;Частное в DL:
    STS  dseg_Pfak,MuLL
    STS  dseg_Pfak+1,MuLH

;.....12
ADC_END_I_PIDreg:
    ;Управление по току
.equ KpI = 5
.equ KiI = 5
.equ KdI = 0

    LDI  AL,KpI
    STS  dseg_Kp,AL
    LDI  AL,KiI
    STS  dseg_Ki,AL
    LDI  AL,KdI
    STS  dseg_Kd,AL

    LDS  AL,dseg_ADC_I
    LDS  AH,dseg_ADC_I+1

    LDI  BL,LOW(12*50) ;Imax = 1,2 A
    LDI  BH,HIGH(12*50)
    RCALL PIDreg
    STS  dseg_OutPID_I,AL
    STS  dseg_OutPID_I+1,AH

;.....20+123
ADC_END_U_PIDreg:
    ;Управление по напряжению

```

```
.equ KpU = 4
.equ KiU = 3
.equ KdU = 0
```

```
LDI AL,KpU
STS dseg_Kp,AL
LDI AL,KiU
STS dseg_Ki,AL
LDI AL,KdU
STS dseg_Kd,AL
```

```
LDS AL,dseg_ADC_U
LDS AH,dseg_ADC_U+1
```

```
LDS BL,dseg_Uzad
LDS BH,dseg_Uzad+1
```

```
RCALL PIDreg
```

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

```
LDS BL,dseg_OutPID_I
LDS BH,dseg_OutPID_I+1
CP AL,BL
CPC AH,BH
BRCS ADC_END_Exits
MOVW AL,BL
```

```
ADC_END_Exits:
```

```
LDI AnL,LOW(Tper/2) ;Период прообразования
LDI AnH,HIGH(Tper/2)
SUB AnL,AL
SBC AnH,AH
STS dseg_OutPID,AnL ; Вывод кода управления ШИМ
STS dseg_OutPID+1,AnH
```

```
ADC_END_Exit:
```

```
LDI AL,0x01
EOR Radmux,AL
STS ADMUX, Radmux
```

```
POP AnH
POP AnL
POP DH
POP DL
POP AH
POP AL
OUT SREG,AL
POP AL
```

```
RETI
```

```
;*****54+123
```

```
PIDreg:
```

```
/* ОПИСАНИЕ PIDreg
ПИД-Регулятор.
Используются:
Функция: Muls16x16_32
```

```
АН;АL <- АDС
ВН:ВL <- заданный параметр
```

```
АН;АL -> Выход ПИД
ВН:ВL <- рассогласование e(t)
АнН:АнL - Темр
ItL - Интегральная составляющая (t-1).
```

Расчет:

$u(t) = P(t) + I(t) + D(t);$   
 $P(t) = K_p * e(t);$   
 $I(t) = I(t-1) + K_i * e(t);$   
 $D(t) = K_d * \{e(t) - e(t-1)\};$

Такты:  $T_{\text{макс}} = 123.* /$

```

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
;Вычисление e(t)
SUB    BL,AL
SBC    BH,AH
;.....
PID_p: ;Пропорциональная составляющая. P(t) = Kp * e(t);
LDS    AnL,dseg_Kp
LDI    AnH,0
RCALL  Mu1s16x16_32      ;DH:DL:CH:CL = ( BH:BL * AnH:AnL )
MOV    AL,CL             ;P(t)
MOV    AH,CH
;..... 9+25
PID_in: ;Интегральная составляющая. I(t) = I(t-1) + Ki * e(t)
LDS    AnL,dseg_Ki
LDI    AnH,0
RCALL  Mu1s16x16_32      ;DH:DL:CH:CL = ( BH:BL * AnH:AnL )
ADD    ItL,CL
ADC    ItH,CH
MOVW   ItL,CL           ;I(t)
;..... 8+25
PID_dif: ;Дифференциальная составляющая. D(t) = Kd * {e(t) - e(t-1)};
MOVW   DL,BL
LDS    AnL,dseg_Kd
LDI    AnH,0
SUB    BL,etL           ;Вычитание e(t)-e(t-1)
SBC    BH,etH
MOVW   etL,DL           ;Сохранение e(t)
RCALL  Mu1s16x16_32      ;DH:DL:CH:CL = ( BH:BL * AnH:AnL ).
ADD    AL,ItL           ;P(t) + I(t)
ADC    AH,ItH
ADD    AL,CL           ;P(t) + D(t) + D(t)
ADC    AH,CH
/*SUB  AL,CL           ;P(t) + D(t) - D(t)
SBC    AH,CH*/
;.....13+25
;Ограничение для ШИМ по верху и нулю.
BRMI  PID_C1           ;Переход по «отрицательное значение»
LDI    AnL,LOW(tmax)
LDI    AnH,HIGH(tmax)
CP     AL,AnL
CPC    AH,AnH
BRLT  PID_Out         ;Переход по «меньше» (для знаковых данных)
MOVW   AL,AnL
RJMP  PID_Out
PID_C1: CLR    AL
CLR    AH
;.....16
PID_Out: RET
;*****2
```

Muls16x16\_32:

/\* ОПИСАНИЕ Muls16x16\_32

Со знаком умножение двух целых чисел +-32767 на +-32767 с результатом +-1073676289.

Используются:

MULH:MULL,

DH:DL:CH:CL = ( BH:BL \* AnH:AnL ).

Такты: Tмакс = 25.

ПРИМЕЧАНИЕ

Подпрограмма неразрушает операнды.\*/

```
PUSH AL
CLR AL
MULS BH,AnH ; (signed)BH * (signed)AH
MOVW DH:DL,MULH:MULL
MUL BL,AnL ; AL * BL
MOVW CH:CL,MULH:MULL
MULSU BH,AnL ; (signed)BH * AL
SBC DH,AL
ADD CH,MULL
ADC DL,MULH
ADC DH,AL
MULSU AnH,BL ; (signed)AH * BL
SBC DH,AL
ADD CH,MULL
ADC DL,MULH
ADC DH,AL
POP AL
```

ret

\*\*\*\*\*

del\_16\_16:

/\* ОПИСАНИЕ del\_16\_16

\*Без знаковое деление 16 битное число на 16 битное число с 16 битным результатом.

\* Используются:

\*DH:DL = ( DH:DL / BH:BL ); (AH:AL) - остаток.

\*При делении на ноль: DH:DL = 0xffff, AH:AL = 0xffff.

\*Такты: Tмин = 230, Tмакс = 247.

```
; .def Sch_c ;Счетчик циклов
; .def BL ;Делитель
; .def BH
; .def DL ;Делимое/Частное
; .def DH
; .def AL ;остаток
; .def AH
```

```
LDI AL,17
MOV Sch_c,AL ;счетчик цикла init
MOV AL,BL ;Если делитель ноль.
OR AL,BH
BRNE del_16_16_d

SER DL ;Запись 0xff
SER DH
SER AL
SER AH
RJMP del_16_16_Ret
```

del\_16\_16\_d:

```
CLR AL ;Очистить Младший байт остатка
SUB AH,AH ;Очистить Старший байт остатка и перенос
```



