

Аппарат сварочный САБП

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.

Многие скажут, что в разработках сварочных аппаратов ничего нового нет и в продаже их, как «грязи». Не в этом дело. Я использовал тему сварочных аппаратов, как пример использования микроконтроллера STM32F103C8 в силовых устройствах.

Чтобы начинающий разработчик почувствовал связь между сермяжной правдой жизни и тонкостями конструирования, и программирования.

Данный сварочный аппарат дополнен некоторыми функциями и показал себя хорошо в эксплуатации.

Идею и идеи мне подкинул хороший сварщик, с которым я сидел рядом за столом на празднике. Сложностью конструкции старался не замарачиваться, но не все получилось.

Комплектация применялась, что была под рукой.

Внешний вид я не показываю, он не презентабельный из-за эксплуатации.

Дополнительная литература:

1. ARM Assembly Language Fundamentals and Techniques (Rus)
<https://disk.yandex.ru/i/FE2giDCHTxX3Vw>
2. Прибор Генератор_Измеритель - https://disk.yandex.ru/i/61op_Dz25PWYkA

Технические данные и описание

Управление аппаратом производится:

- тумблером ВЫХОД - ВКЛ/ОТКЛ управлением выходом,
- кнопкой ЭНКОДЕР - управления режимами,
- движок ЭНКОДЕР - изменение параметров режима.

Тумблером ВЫХОД включается напряжение на выходе аппарата. При включенном выходе блокируется ЭНКОДЕР. Включении питания сети, при включенном тумблере ВЫХОД, выход блокируется.

Для дальнейшей работы - тумблер ВЫХОД надо выключить.

Нажимая на кнопку ЭНКОДЕР можно последовательно переходить к нужным режимам, индикация которых показана на Рис.1 ... Рис.5.

Вращая ЭНКОДЕР можно изменить нужный параметр. Изменения запоминаются в батарейной памяти.

Параметры ограничиваются программно.



Рис.1. Режим 0 - выбор электрода.

В «Режиме 0» - выбор диаметра электрода или «пользовательский электрод».

Пример: выбран электрод – 2,5мм (далее все параметры показаны для него).

Параметры электрода загружаются нажатием кнопки ЭНКОДЕР и запоминаются в батарейной памяти. При включении питания аппарата, из батарейной памяти, параметры загружаются - как «пользовательский электрод».

В «Режиме 0» включение выхода блокируется.



Рис.2. Режим 1,2 – задание I_z , U_z или индикации фактического I_f , U_f .

В «Режиме 1» задается ток стабилизации I_z .

В «Режиме 2» задается напряжение стабилизации U_z .

В «Режиме 1, 2» индикация фактического тока I_f и напряжения U_f выхода.



Рис.3. Режим 3,4 – задание Тимп, U_z от залипания электрода

В «Режиме 3» задается время перегрева электрода при залипании.

В «Режиме 4» задается напряжение дуги: $U_{мин}$ - меньше которого, включается режим отработки залипания.



Рис.4. Режим 5,6 – задание тока I_{max} перегрева при залипании и напряжение зажигания дуги $U_{заж}$.

В «Режиме 5» задается ток перегрева I_{max} при залипании

В «Режиме 6» задается напряжение зажигания дуги $U_{заж}$ ($U_{max} \cdot 1$ или $U_{max} \cdot 2$).

U_{max} – максимальное выходное напряжение, при текущем сетевом напряжении.



Рис.5. Режим 7 – ВКЛ/ОТКЛ гашения дуги.

В «**Режиме 7**» можно ВКЛ/ОТКЛ гашения дуги. Гашение дуги бывает необходимо в некоторых режимах сварки. **Например:** сварка тонких элементов деталей - прихватками.

Отключение выхода производится при превышении напряжения на дуге более - 30 В, на время - 0,5 Сек.

Если аппарат находится в одном из режимов: «**Режим 3**» ... «**Режим 7**» и при этом включаете ВЫХОД, индикация переходит в Режим 1, 2.

В «**Режиме 8**» (Рисунок индикации не показан) – индицируется «Нагрев: МАХ». Эта надпись выводится, когда температура датчика температуры превысит 60°C и будет индицироваться пока температура датчика не упадет ниже 40°C. В этом режиме работа с аппаратом будет заблокирована.

Аппарат может работать, как **мощный стабилизатор тока** (1 ... 70 А – длительный режим) и **напряжения** (1 ... 50 В). В этом случае желательно на выход ставить дополнительные конденсаторы емкостью – минимум 2200 мкФ.

Аппаратом можно **заряжать аккумуляторы и запускать двигатель автомобиля** при разряженном аккумуляторе.

Описание конструкции

Все описание конструкции свел к описанию силовой части и помехоустойчивости.

Конструкция

Конструкция упирается в основном в длину и сечение проводников силовых цепей. Меньше длина и больше сечение, будет меньше нагрев. Естественно это противоречит с уменьшением размера и веса. Больше всего греются силовые дроссели и трансформатор. Около них крепится термодатчик.

Конструктив силовой платы показан на Рис.6 (размер 230x190 мм).

Силовые дорожки шириной 10 мм усилены (напаяны) одним слоем меди 0,3 мм.

Силовые дорожки шириной более 12 мм усилены двумя слоями меди 0,3 мм.

Красным показаны навесные проводники. Узкие – проводом, на стороне элементов.

Широкие, напаяны мостиком: 10 мм - одним слоем меди 0,3 мм, 12 мм - двумя слоями меди 0,3 мм.

Установка силовых транзисторов VT1 ... VT4 показана на Рис.7 – такая установка улучшает тепловой режим транзистора, хотя это увеличивает затраты.

Силовой дроссель L1 (см. Рис.8) намотан на 3-х сложенных прямоугольных ферритовых сердечников от магнитной антенны приемников - медной шиной ПЭТВ 2x5 – 30 вит. Контрольная индуктивность дросселя 150 ... 180 мкГн.

Силовой трансформатор TV2 (см. Рис.9, Рис.11) для лучшего охлаждения намотан бескаркасной намоткой. Обмотки мотаются на оправках размеры, которых позволяли, чтобы расстояние между обмотками и сердечником было не менее 1,5 мм.

Витки и сами обмотки фиксируются эпоксидным клеем и узкими вставками из стеклотекстолита 1,5 ... 2 x 4 мм.

Первичная обмотка мотается медной шиной ПЭТВ 1,2x2,5 – 15 вит.

Вторичная обмотка мотается медной шиной ПЭТВ 2x5 – 2x3 вит.

Конечно изготовление подобного трансформатора - муторная работа, но параметры у него лучше, чем на кольце – меньше рассеивание магнитного поля, что влияет на уменьшение помех.

Импульсный трансформатор TV1 (см. Рис.11) играет не меньшую роль в надежности аппарата, чем остальные силовые элементы и одновременно является самым простым согласованием силовых транзисторов со схемой управления.

Главное, чтобы трансформатор не звенел сильно на фронтах импульсов. Это обеспечивается минимумом паразитных индуктивностей и емкостей трансформатора, ну естественно и конструктивом аппарата.

Конкретный трансформатор намотан на 4-х кольцах ГМ14ДС-3 K26x12x5 (под рукой были): $\mu = 30000$. Контрольная индуктивность: 15 ... 18 мГн и при замыкании любой другой обмотки падает до 8 ... 2 мкГн.

Можно применить 2 шт. импульсных трансформатора МИТ-4В, но у него больше паразитные емкости, а при импульсах до 300 В это имеет значения для помехоустойчивости.

Помехоустойчивость

Для начала надо осознать: Пути электричества неисповедимы.

Борьба с помехами не бывает “бескровной” и это специфика мощных импульсных систем. Но вернемся к аппарату.

В основу управления взято «железо» - китайская плата на основе (см. Рис.12) микроконтроллера STM32F103C8. Поскольку китайцы слишком экономные, они не поставили достаточную емкость по 3,3 В, что сказалось на помехоустойчивости системы.

Пришлось брать из материнских плат керамические ЧИП – конденсаторы 10 мкФ и вешать на выводы китайской платы, как показано на схеме (см. Рис.12 – конденсаторы цвета морской волны). Ставить рядом в обрамлении, большого эффекта не дало.

Человека соображающего, удивит нагромождение микросхем все это из-за помех. Первый вариант был очень простой, а после борьбы с помехами я получил то, что получил в имеющемся конструктиве. Я могу предположить, что в вашем конструктиве «погода» с помехами может быть лучше, а может и хуже.

В основном помехи перли через энкодер. Пришлось энкодер перевести на 5 В, добавив микросхемы с триггером Шмидта. Но иногда помехи проскальзывали и чтобы не заморачиваться, при включенном выходе программно блокировал ЭНКОДЕР.

Добавлено было еще несколько программных защит, но они пока ни как себя не проявили.

Программная блокировка портов обязательна!

Замечания

Перед установкой силовых элементов, проверяйте их по параметрам и если не жалко вскройте один, особенно транзистор, путем обжига. И если кристалл меньше 4 мм, то это не лучший вариант и быстро сдохнет.

Если вы этого не сделаете, потеряете много времени, пока не разберетесь и долго будете материть китайцев.

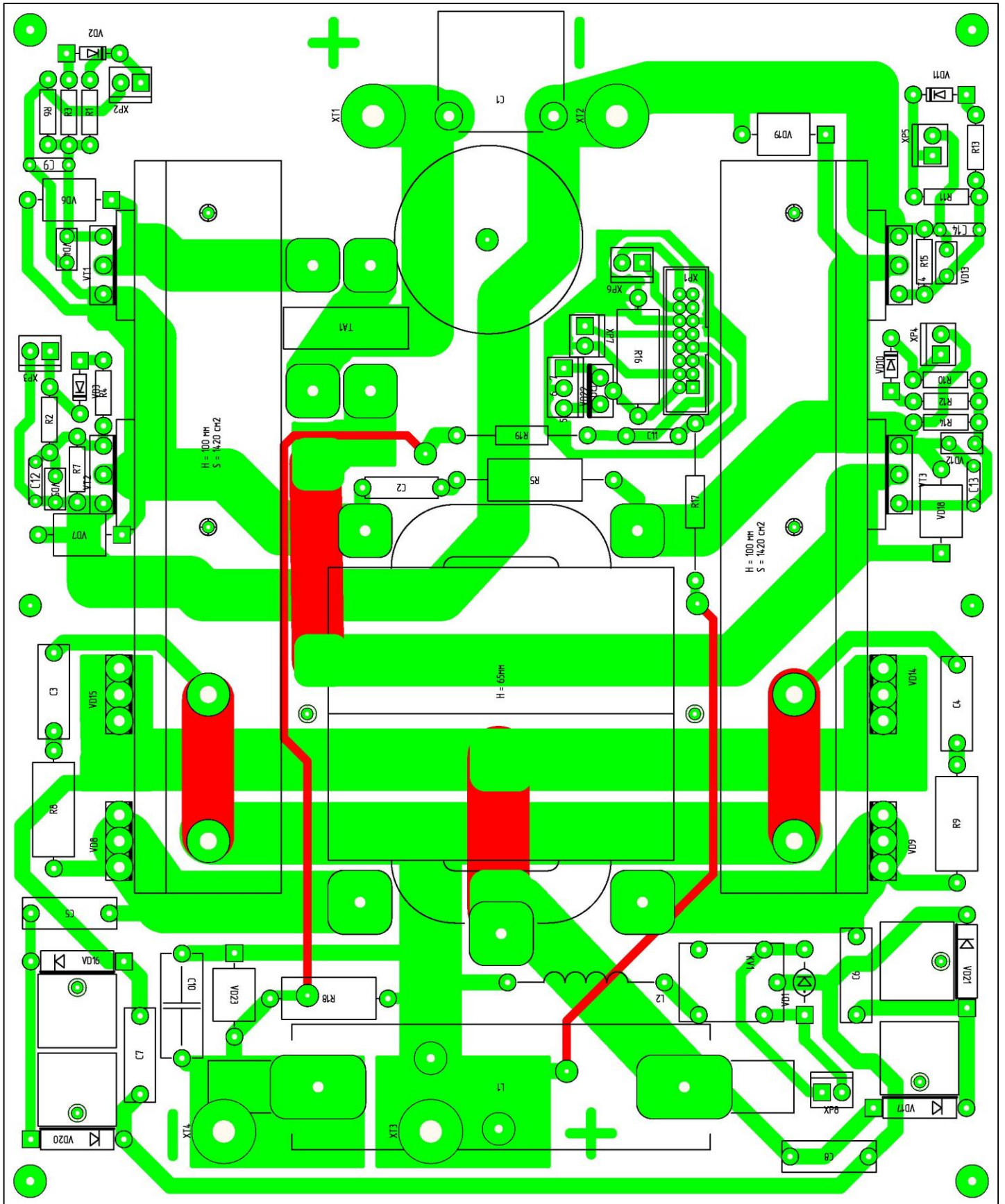


Рис.6. Плата силовая 160А

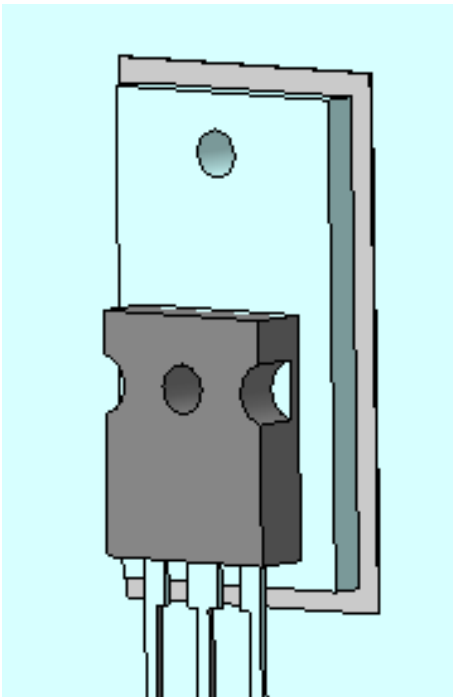


Рис.7 Установка силового транзистора



Рис.8 Силовой дроссель



Рис.9 Силовой трансформатор.

Рис.10 Плата силовая 160А НЧ

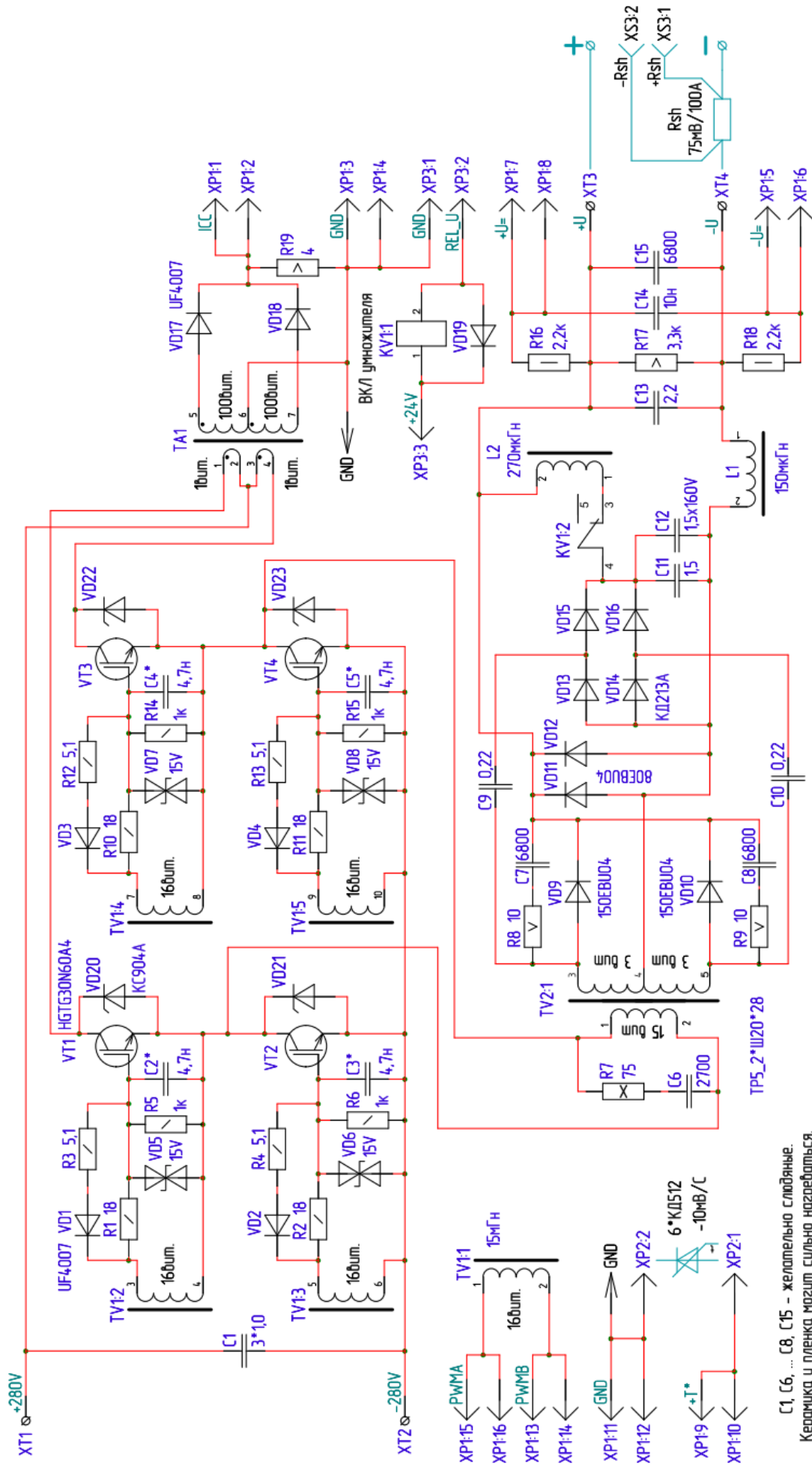


Рис.11 Плата силовая 160А

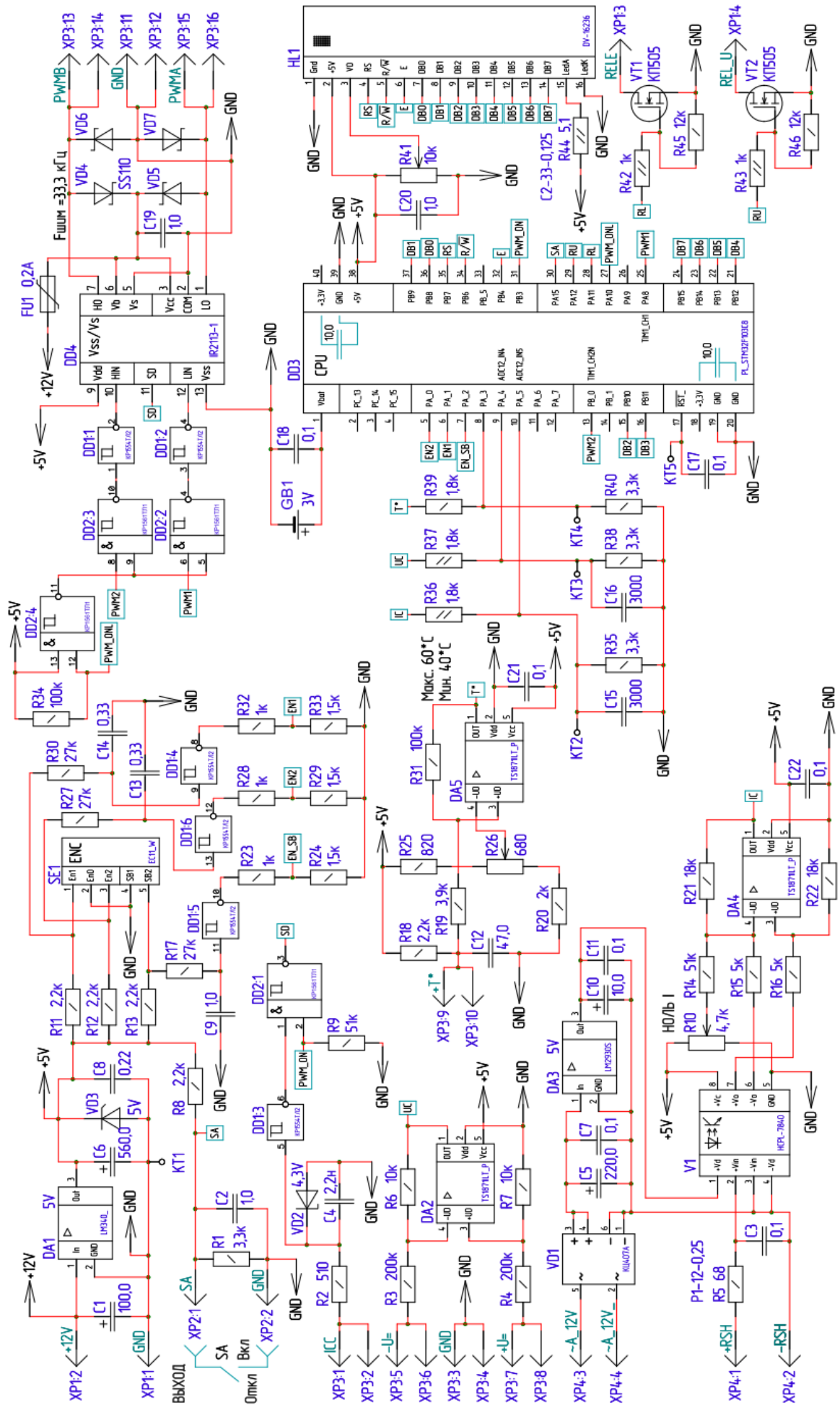


Рис.12 Плата сборки 160A PL_STM32F103C8_v2

Программное обеспечение (ПО) аппарата.

Автор: А.В. Жмаев, E-mail: kuratorj@yandex.ru

```
;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

Unit   RN 11      ;"1" - единица (R11)
Noll   RN 10      ;"0" - ноль (Noll)

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

Rsos_Rej      EQU    0      ;Rsos[0...3] - РЕЖИМ
Rsos_Proh_Pauses EQU    0x00000100 ; Обработки Паузы: 0 - Запрет, 1 - Разрешение
Rsos_Pauses    EQU    0x00000200 ; 1 - Признак Паузы выхода
Rsos_Gash_Dugi EQU    0x00000400 ; Гашения дуги: 0 - ОТКЛ, 1 - ВКЛ
Rsos_Indic_DD  EQU    0x00000800 ; 1 - Запись данных в индикатор, 0 - Запись команд в
индикатор.
Rsos_W_Rezerv EQU    0x00001000 ; 1 - Разрешение записи данных в резервную область
Rsos_Perm_RL   EQU    0x00002000 ; 1 - Включения силового реле зарядки конденсаторов
Rsos_Dial_IU   EQU    0x00004000 ; Выбор набора: 0 - Ток, 1 - Напряжение
Rsos_Zajigan_U EQU    0x00008000 ; 0 - Uout*1, 1 - Uout*2
Rsos_Sticking  EQU    0x00010000 ; Признак Залипания электрода
Rsos_Long_Comm EQU    0x00020000 ; Признак Запись длинной (по времени) команды в
индикатор (1,53 мс)
Rsos_tC_max    EQU    0x00040000 ; 1 - Превышение температуры.
Rsos_SA_Trq    EQU    0x00080000 ; SA: 1 - ОТКЛЮЧЕН, 0 - ВКЛЮЧЕН

Rsos_Rej_Copy  EQU    28     ;Rsos[28...31] - Копия РЕЖИМА
;.....

;Признаки
Pr00          EQU    0x40     ;Признак 00,0 текста числа
zpt1          EQU    0x80     ;Признак запятой текста числа 0,0
zpt2          EQU    0x90     ;Признак запятой текста числа 0,00
zpt3          EQU    0xa0     ;Признак запятой текста числа 0,000
P_end_Txt     EQU    0x80000000 ;Признак конца текста
CMar_Que      EQU    0xc0000000 ;Признак SRAM_Buff_x[27]: "0" - вывод на индикатор числа, "1" -
вывод на индикатор управления маркером
;.....

;Привязка к схеме
E_PB4         EQU    0x00000010 ; Порт В Установка PB4 - Запись в индикатор: "1"
R_W_PB6       EQU    0x00000040 ; Порт В Установка PB6 - выбор Чтение: "1", Запись: "0"
RS_PB7        EQU    0x00000080 ; Порт В Установка PB7 - Выбор данных: "1", Команда: "0"

PWM_ONL_PA10  EQU    0x00000400 ; Порт А Установка PA10 - "0": ВКЛ Выхода ШИМ
PWM_ON_PB3    EQU    0x00000008 ; Порт В Установка PB3 - "1": ВКЛ Выхода ШИМ

tC_max_PA3    EQU    0x00000008 ; Порт А Установка PA3 - "0": Превышение температуры

RL_PA11       EQU    0x00000800 ; Порт В Установка PA11 - "1": ВКЛ силового реле зарядки
конденсаторов
RU_PA12       EQU    0x00001000 ; Порт А Установка PA12 - "1": Подключение умножителя

SA_PA15       EQU    0x00008000 ; Порт А Установка PA15 - "0": ВКЛ, "1": ОТКЛ - Выхода
аппарата
;.....
```

;Оперативная память

; SRAM_Buff_x[30...27] - параметр вывода на индикатор строк или управления маркером
 ; SRAM_Buff_x[31] - CMar_Que: "0" - вывод на индикатор числа, "1" - вывод на индикатор управления маркером
 ; SRAM_Buff_x[26...0] - выводимое на индикатор число

SRAM_Buff_0	EQU	0x20000000
SRAM_Buff_1	EQU	0x20000004
SRAM_Buff_2	EQU	0x20000008
SRAM_Buff_3	EQU	0x2000000c
SRAM_Buff_4	EQU	0x20000010
SRAM_Buff_5	EQU	0x20000014
SRAM_Buff_6	EQU	0x20000018
SRAM_Buff_7	EQU	0x2000001c
SRAM_Buff_8	EQU	0x20000020
SRAM_Buff_9	EQU	0x20000024
SRAM_Buff_10	EQU	0x20000028
SRAM_Buff_11	EQU	0x2000002c
SRAM_Buff_12	EQU	0x20000030
SRAM_Buff_13	EQU	0x20000034
SRAM_Buff_14	EQU	0x20000038
SRAM_Buff_15	EQU	0x2000003c
SRAM_Buff_16	EQU	0x20000040

SRAM_Izad	EQU	0x20000048
SRAM_Uzad	EQU	0x2000004c
SRAM_Timp	EQU	0x20000050
SRAM_Umin	EQU	0x20000054
SRAM_Imax	EQU	0x20000058

SRAM_Iadc	EQU	0x2000005c
SRAM_Uadc	EQU	0x20000060

SRAM_PWM_Tper	EQU	0x20000064
SRAM_PWM_Timp	EQU	0x20000068
SRAM_Nsamp	EQU	0x2000006c

; Количество выборок за период ШИМ

SRAM_et_I	EQU	0x20000070
SRAM_It_I	EQU	0x20000074
SRAM_et_U	EQU	0x20000078
SRAM_It_U	EQU	0x2000007c

SRAM>Ifak	EQU	0x20000080
SRAM_Ufak	EQU	0x20000084

SRAM_Control0	EQU	0x20000088
SRAM_Control1	EQU	0x2000008c

SRAM_Iout	EQU	0x20000090
SRAM_Uout	EQU	0x20000094

SRAM_IU_count	EQU	0x20000098
SRAM_I_summ	EQU	0x2000009c
SRAM_U_summ	EQU	0x200000a0

SRAM_IU_fak	EQU	0x200000b0
-------------	-----	------------

SRAM_D	EQU	0x20000200	; Информация выводимая на индикатор (Макс. 32 байт)
SRAM_Form_Dec	EQU	0x20000220	; 16 байт
SRAM_Func_Dec	EQU	0x20000230	; Адрес (младший разряд) области памяти 8-разрядного десятичного числа (Макс. 16 байт)
SRAM_ADC_IU	EQU	0x20000250	; 256 байт

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

```
;Форвардная декларация разделов.
```

```
AREA RESET, DATA, READONLY, align = 2
```

```
THUMB
```

```
__vector_table
```

```
DCD 0x20001000 ; - 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 SVCall_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, PE2, 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 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
```

```
MOV32 R0, #SCB_AICR
```

```
MOV32 R1, #(SCB_AICR_VECTKEY+SCB_AICR_SYSRESETREQ)
```

```
STR R1, [R0]
```

```
BKPT #1
```

```
MemManageFault_IRQHandler ; - отказ системы управления памятью
```

```
MOV32 R0, #SCB_AICR
```

```
MOV32 R1, #(SCB_AICR_VECTKEY+SCB_AICR_SYSRESETREQ)
```

```
STR R1, [R0]
```

```
BKPT #1
```

```
BusFault_IRQHandler ; - отказ шины
```

```
MOV32 R0, #SCB_AICR
```

```
MOV32 R1, #(SCB_AICR_VECTKEY+SCB_AICR_SYSRESETREQ)
```

```
STR R1, [R0]
```

```
BKPT #1
```

```
UsageFault_IRQHandler ; - отказ программы
```

```
MOV32 R0, #SCB_AICR
```

```
MOV32 R1, #(SCB_AICR_VECTKEY+SCB_AICR_SYSRESETREQ)
```

```
STR R1, [R0]
```

```
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)
    MOV32    R0, #Inter_ENCEX
    BX R0

EXTI2_IRQHandler      ; - прерывание от линии 2 модуля EXTI (PA2, PB2, PC2, PD2, PE0, PF2, PG2)
    MOV32    R0, #Inter_ENCsbEx
    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
    MOV32    R0, #Inter_ADC1
    BX      R0
    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
    BKPT #1
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
    BKPT #1
TIM3_IRQHandler          ; - любое прерывание от модуля TIM3
    BKPT #1
TIM4_IRQHandler          ; - любое прерывание от модуля TIM4
    MOV32    R0, #Inter_TIM4
    BX R0
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_Channel3_IRQHandler ; - прерывание от канала Channel3 модуля DMA2
BKPT #1
DMA2_Channel4_5_IRQHandler ; - прерывание от каналов Channel4 и Channel5 модуля DMA2
BKPT #1

```

.....

```

align
; Начало программы
Start PROC

NOP
MOV     No11, #0
MOV     Unit, #1

;Инициализация тактирования
MOV32   R0, #RCC
; За тактировать 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 (10-2) << 18 ; Коэффициент умножения основной системы ФАПЧ (PLL). здесь 10 =
коэфф. умножения PLLMUL. SYSCLK = 80 mHz

MOV32   R1, #(PLLMUL + RCC_CFGR_PLLSRC + RCC_CFGR_PPRE1_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, 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_TIM1_REMAP_PARTIALREMAP) ;JTAG-DP отключен и SW-DP включен.
    STR        R1, [R0,#AFIO_MAPR]

    MOV32      R1, #(AFIO_EXTICR1_EXTI1_PA+AFIO_EXTICR1_EXTI2_PA) ;Внешнее прерывание
    STR        R1, [R0,#AFIO_EXTICR1]

;.....

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

    MOV        R1, #PWM_ONL_PA10 ;Отключение ШИМ
    STR        R1, [R0,#GPIO_BSRR]

    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)))+(RES_IN<<(ODR9*4)))+(AF_PP_50<<(ODR8*4)))
    STR        R1, [R0,#GPIO_CRH]
;Блокировка порта
    MOV32      R1, #0x1ffff
    STR        R1, [R0,#GPIO_LCKR]

```

```

MOV32    R1, #0xffff
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))+(PP_2<<(ODR6*4))+(RES_IN<<(ODR5*4))+
(P_2<<(ODR4*4)))+(PP_2<<(ODR3*4)))+(RES_IN<<(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)))+(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, #0x19fff
STR      R1, [R0,#GPIO_LCKR]
MOV32    R1, #0x09fff
STR      R1, [R0,#GPIO_LCKR]
MOV32    R1, #0x19fff
STR      R1, [R0,#GPIO_LCKR]
LDR      R1, [R0,#GPIO_LCKR]

;инициализация порт C
MOV32    R0, #GPIOC
LDR      R1, [R0,#GPIO_CRH]
BIC      R1, #RES_IN<<(ODR13*4)
ORR      R1, #PP_2<<(ODR13*4)
STR      R1, [R0,#GPIO_CRH] ; 0x44244444
;Блокировка порта
MOV32    R1, #0x19fff
STR      R1, [R0,#GPIO_LCKR]
MOV32    R1, #0x09fff
STR      R1, [R0,#GPIO_LCKR]
MOV32    R1, #0x19fff
STR      R1, [R0,#GPIO_LCKR]
LDR      R1, [R0,#GPIO_LCKR]

;.....

Init_TIM1 ;Инициализация TIM1 - для ШИМ

; 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

MOV      R1, #2400/2 ; Tper = 30 мкс, при f = 80 МГц
SUB      R2, R1,Unit ;Tper/2 - 1
STR      R2, [R0,#TIM_ARR] ; TIM регистр авто-перезагрузки
MOV      R2, #0
SUB      R1, R1,R2
STR      R1, [R0,#TIM_CCR1]
STR      R2, [R0,#TIM_CCR2]
STR      No11, [R0,#TIM_CCR3]

```

```

STR      No11, [R0,#TIM_PSC]      ; TIM предделитель

MOV      R2, #(TIM_CR1_CMS_3+TIM_CR1_ARPE);
STR      R2, [R0,#TIM_CR1]      ; TIM управляющий регистр 1
STR      No11, [R0,#TIM_CR2]      ; TIM управляющий регистр 2

MOV      R2, #(TIM_CCMR1_OC1M_PWM2+TIM_CCMR1_OC1PE+ TIM_CCMR1_OC2M_PWM2+
TIM_CCMR1_OC2PE)
STR      R2, [R0,#TIM_CCMR1]
MOV      R2, #(TIM_CCER_CC1E+TIM_CCER_CC1NE + TIM_CCER_CC2E+TIM_CCER_CC2NE)
STR      R2, [R0,#TIM_CCER]

MOV      R2, #TIM_DIER_CC3IE      ; Разрешить захват/сравнение 3 прерывания
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]

;.....

Init_SRAM      ;Инициализация SRAM (Обнуление)

MOV32     R0, #SRAM_Buff_0
MOV      R1, #1024

Init_SRAM_clk      ;Обнуление
STR      No11, [R0],#4
SUBS     R1, #1
BNE Init_SRAM_clk

;.....

Init_BKP      ;Переслать Данные из BKP в SRAM
MOV32     R0, #BKP+BKP_DR1
MOV32     R1, #SRAM_Izad
BL Forward_Data      ;Переслать Данные: R0 - адрес источника, R1 - адрес приемника

;.....

Init_TIM4      ;Инициализация TIM4
;Тпауз - время гашение дуги, если Ufak > 30 V

; Тактирование TIM4
MOV32     R0, #RCC
LDR      R1, [R0,#RCC_APB1ENR]
ORR      R1, #RCC_APB1ENR_TIM4EN
STR      R1, [R0,#RCC_APB1ENR]

;Инициализация TIM4

MOV32     R0, #TIM4

MOV      R1, #50000-1      ;Tper = 1,25 мс при f = 80 МГц
STR      No11, [R0,#TIM_PSC]      ; TIM предделитель

```

```

MOV      R1, #400          ;Трауз = 0,5 С при f = 80 МГц
STR      R1, [R0,#TIM_ARR] ; TIM регистр авто-перезагрузки

STR      R0, [R0,#TIM_CR2]      ; TIM управляющий регистр 2

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

;Разрешение глобального прерывания TIM4_IRQ = 30 - любое прерывание от модуля TIM4
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<<TIM4_IRQn)
STR      R1, [R0]

;.....

Init_TIM2      ;Инициализация TIM2
               ;Время повышение тока при залипании электрода и время гашение дуги

; Тактирование TIM2
MOV32    R0, #RCC
LDR      R1, [R0,#RCC_APB1ENR]
ORR      R1, #RCC_APB1ENR_TIM2EN
STR      R1, [R0,#RCC_APB1ENR]

;Инициализация TIM2

MOV32    R0, #TIM2

MOV      R1, #50000-1        ;Tper = 1,25 мС при f = 80 МГц
STR      R1, [R0,#TIM_PSC]   ; TIM предделитель

MOV      R1, #1000-1         ;Tper = 1,25 С при f = 80 МГц
STR      R1, [R0,#TIM_ARR]   ; TIM регистр авто-перезагрузки
;Время повышение тока при залипании электрода
MOV      R1, #8              ;Тимп = 0,01 С при f = 80 МГц
STR      R1, [R0,#TIM_CCR1]
;Тпауз - время гашение дуги, если Uфак > 28 V
MOV      R2, #400           ;Трауз = 0,5 С при f = 80 МГц
STR      R2, [R0,#TIM_CCR2]

STR      R0, [R0,#TIM_CR2]   ; TIM управляющий регистр 2

MOV      R2, #(TIM_DIER_CC2IE+TIM_DIER_CC1IE +TIM_DIER_UIE) ;
STR      R2, [R0,#TIM_DIER]

;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]

;.....

Nsamp_Adc      ;Количество запусков Nsamp_Adc за Tper = 30 мкС при f = 80 МГц
EQU      15

;.....

```

```

Init_ADC2      ;Инициализация АЦП2
               ; Тактирование
MOV32          R0, #RCC
LDR            R1, [R0,#RCC_APB2ENR]
ORR            R1, #RCC_APB2ENR_ADC2EN
STR            R1, [R0,#RCC_APB2ENR]
               ;Перод Тактов АЦП 0,1 мкс
MOV32          R0, #ADC2
MOV            R1, #(ADC_CR2_ADON+ADC_CR2_CONT)
STR            R1, [R0,#ADC_CR2]

MOV            R1, #ADC_CR1_SCAN
STR            R1, [R0,#ADC_CR1]

MOV            R1, #(ADC_SMPR_SMP7_5 << 12)      ;Канал 4, выборка: 7,5 цикла
STR            R1, [R0,#ADC_SMPR2]

MOV32          R1, #((4<<0)+(4<<5)+(4<<10)+(4<<15)+(4<<20)+(4<<25)) ;Канал 4: Ufak
STR            R1, [R0,#ADC_SQR3]
MOV32          R1, #((4<<0)+(4<<5)+(4<<10)+(4<<15)+(4<<20)+(4<<25)) ;Канал 4: Ufak
STR            R1, [R0,#ADC_SQR2]
MOV32          R1, #((4<<0)+(4<<5)+(4<<10)+((Nsamp_Adc-1)<<20)) ;Канал 4: Ufak
STR            R1, [R0,#ADC_SQR1]
               ; Сбросить калибровку
LDR            R1, [R0,#ADC_CR2]
ORR            R1, R1,#ADC_CR2_RSTCAL
STR            R1, [R0,#ADC_CR2]

Init_ADC2_RSTCAL ;Ожидание сброса калибровки
LDR            R1, [R0,#ADC_CR2]
TST            R1, #ADC_CR2_RSTCAL
BNE Init_ADC2_RSTCAL
               ; ADC2 Калибровка
LDR            R1, [R0,#ADC_CR2]
MOV32          R2, #(ADC_CR2_CAL)
ORR            R1, R1,R2
STR            R1, [R0,#ADC_CR2]

Init_ADC2_CAL   ;Ожидание конца калибровки
LDR            R1, [R0,#ADC_CR2]
TST            R1, #ADC_CR2_CAL
BNE Init_ADC2_CAL

LDR            R1, [R0,#ADC_CR2]
ORR            R1, R1,#ADC_CR2_EXTTRIG
STR            R1, [R0,#ADC_CR2]
LDR            R1, [R0,#ADC_CR2]
ORR            R1, R1,#ADC_CR2_ADON      ;1 - ADC2 включить
STR            R1, [R0,#ADC_CR2]

;.....

Init_ADC1      ;Инициализация АЦП1 - обычный одновременный режим ADC1 и ADC2
               ; Тактирование АЦП1
MOV32          R0, #RCC
LDR            R1, [R0,#RCC_APB2ENR]
ORR            R1, #RCC_APB2ENR_ADC1EN
STR            R1, [R0,#RCC_APB2ENR]

               ;Перод Тактов АЦП 0,1 мкс
MOV32          R0, #ADC1
MOV            R1, #(ADC_CR2_ADON+ADC_CR2_CONT)
STR            R1, [R0,#ADC_CR2]

```

```

MOV32      R1, #(ADC_CR1_SCAN+ADC_CR1_DUALMOD_2+ADC_CR1_DUALMOD_1+ADC_CR1_EOCIE)
; 0110: Только обычный одновременный режим ADC1 и ADC2
STR        R1, [R0,#ADC_CR1]

MOV        R1, #(ADC_SMPR_SMP7_5 << 15)      ;Канал 5 - выборка: 7,5 цикла
STR        R1, [R0,#ADC_SMPR2]

MOV32      R1, #((5<<0)+(5<<5)+(5<<10)+(5<<15)+(5<<20)+(5<<25)) ;Канал 5: Ufak
STR        R1, [R0,#ADC_SQR3]
MOV32      R1, #((5<<0)+(5<<5)+(5<<10)+(5<<15)+(5<<20)+(5<<25)) ;Канал 5: Ufak
STR        R1, [R0,#ADC_SQR2]
MOV32      R1, #((5<<0)+(5<<5)+(5<<10)+((Nsamp_Adc-1)<<20)) ;Канал 5: Ufak
STR        R1, [R0,#ADC_SQR1]

LDR        R1, [R0,#ADC_CR2]
MOV32      R2, #ADC_CR2_DMA
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)
            ; Тактирование DMA1
MOV32      R0, #RCC
LDR        R1, [R0,#RCC_AHBENR]
MOV32      R2, #RCC_AHBENR_DMA1EN
ORR        R1, R1,R2      ;
STR        R1, [R0,#RCC_AHBENR]

MOV32      R0, #DMA1
MOV        R1, #Nsamp_Adc      ; 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_IU     ;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_32+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]

;.....

;Разрешение глобального прерывания EXTI1_IRQn = 7, EXTI2_IRQn = 8
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<<EXTI1_IRQn)+(1<<EXTI2_IRQn))
STR            R1, [R0]

;.....
BL Zader_2mC ;Ожидание выхода на режим индикатора

MOV           Rsos, Noll

;.....
Init_Indic    ;Начальная загрузка индикатора
MOV32 R0, #Indic_Tab_Init
BL Indic_W_cdseg
BL Zader_2mC

BL Ind_Rej_Alektr
BL Zader_2mC

MOV32         R1, #SRAM_Ifak
STR            Noll, [R1]
STR            Noll, [R1,#4]

;.....
Init_Zader_onRL ; Задержка разрешения включения силового реле зарядки конденсаторов
MOV32         R12, #16000000
Init_Zader_onRL_ck1 ;Цикл 5 тактов
NOP
SUBS          R12,R12,#1
BNE Init_Zader_onRL_ck1

ORR            Rsos, #Rsos_Perm_RL ;Разрешения включения силового реле зарядки
конденсаторов

;.....

Init_EXTI     ;Инициализация Внешнее прерывание

MOV32         R0, #NVIC_ICPR0          ; Сбросить признаки отложенных прерываний 0...32
MOV           R1, #-1
STR            R1, [R0]

MOV32         R0, #EXTI
MOV32         R1, #(EXTI_PR_PR1+EXTI_PR_PR2) ;Внешнее прерывание Сброс
STR            R1, [R0,#EXTI_PR]
MOV32         R1, #EXTI_FTSR_TR1 ;Внешнее прерывание Падающий фронт Энкодер
STR            R1, [R0,#EXTI_FTSR]
MOV32         R1, #EXTI_RTSR_TR2 ;Внешнее прерывание Нарастающего фронт Кнопки
Энкодера
STR            R1, [R0,#EXTI_RTSR]
MOV32         R1, #(EXTI_IMR_MR1+EXTI_IMR_MR2) ;Внешнее прерывание Маски Включаем
прерывание

```

```

STR            R1, [R0,#EXTI_IMR]

;.....

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]

ADC1           ;Разрешение глобального прерывания ADC1_2_IRQn = 18 - любое прерывание от модуля
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<<ADC1_2_IRQn)
STR            R1, [R0]

;.....

Duty           ; Дежурный цикл  Мигание светодиода
MOV            R8, #5

Duty_HL_ck1    ; Цикл вывода на индикатор

MOV32          R12, #200000

Duty_W_Buff
MOV32          R0, #SRAM_Buff_0
LDR            R1, [R0]
CBZ            R1, Duty_Buff_Ind_No11
BL W_Buff_Ind      ; Перезапись из буферной памяти на индикатор

Duty_Buff_Ind_No11
SUBS           R12, Unit
BNE Duty_W_Buff

Duty_NoNo11
SUBS           R8,#1
BNE Duty_HL_ck1
; Мигание светодиода
MOV32          R0, #GPIOC      ; Регистр выходных данных порта GPIO
LDR            R1, [R0,#GPIO_ODR]
EOR            R1, #GPIO_ODR_ODR13
STR            R1, [R0,#GPIO_ODR]

BL PtC_max

BL OUT_OnOff

Duty_IU
BL Buff_NoNo11      ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

AND            R1, Rsos,#0x0f
CMP            R1, #1
BNE Duty_no_IU

```

```

MOV32    R1, #SRAM_Ifak
LDR      R2, [R1]
MOV32    R1, #0x10000000
ORR      R1, R2
STR      R1, [R0]
MOV32    R1, #SRAM_Ufak
LDR      R2, [R1]
MOV32    R1, #0x20000000
ORR      R1, R2
STR      R1, [R0,#4]

Напряжение
TST      Rsos, #Rsos_Dial_IU      ;Rsos_Dial_IU Выбор набора: 0 - Ток, 1 -
MOV32EQ  R1, #0x48000000
MOV32NE  R1, #0x50000000
STR      R1, [R0,#8]

Duty_no_IU
резервную область
TST      Rsos, #Rsos_W_Rezerv      ; Флаг Rsos: 1 - Разрешение записи данных в
BLNE     W_Rezerv      ; Запись данных в резервную область
BIC      Rsos, #Rsos_W_Rezerv      ; Флаг Rsos: 0 - Разрешение записи данных в
резервную область

B Duty

ENDP

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

align
Inter_ADC1      PROC      ;Немаскированное прерывание от канала TIM1_CC

PUSH      {LR}
PUSH      {R4-R8}

MOV32     R0, #NVIC_ICPR0      ; Сбросить признаки отложенных прерываний 0...32
MOV       R1, #(1<<ADC1_2_IRQn)
STR       R1, [R0]

; Усреднение за период ШИМ
MOV32     R0, #SRAM_ADC_IU      ;Адрес начало массива в SRAM
MOV       R12, #Nsamp_Adc      ; Количество выборок за период ШИМ
MOV       R2, No11      ; Ток Iadc
MOV       R3, No11      ; Напряжение Uadc

;.....

;Усреднение Iadc и Uadc за период ШИМ
Inter_ADC1_ck1
LDR       R1, [R0],#4
UBFX     R4, R1,#0, #12
ADD      R2, R4      ; Ток Iadc
UBFX     R4, R1,#16, #12
ADD      R3, R4      ; Напряжение Uadc
SUBS     R12, Unit
BNE      Inter_ADC1_ck1

MOV       R12, #Nsamp_Adc      ; Количество выборок за период ШИМ
UDIV     R2, R2,R12
UDIV     R3, R3,R12

;Коррекция нуля

```

```

SUB      R2, #10
USAT     R2, #31, R2    ; Ток Iadc
SUB      R3, #0
USAT     R3, #31, R3    ; Напряжение Uadc

```

```

;Коррекция Тока и Напряжения

```

```

MOV      R12, #103
MUL      R2, R12
MOV      R12, #100
UDIV     R2, R12

MOV32    R0, #SRAM_IU_fak
STR      R2, [R0, #(SRAM_Iout-SRAM_IU_fak)]

```

```

MOV      R12, #103
MUL      R3, R12
MOV      R12, #100
UDIV     R3, R12
STR      R3, [R0, #(SRAM_Uout-SRAM_IU_fak)]

```

```

;.....

```

```

Inter_ADC1_Aver

```

```

;Усреднение Тока, Напряжения для индикатора

```

```

;Счёт усреднения

```

```

LDR      R1, [R0, #(SRAM_IU_count-SRAM_IU_fak)] ;
ADD      R1, #1
CMP      R1, #64
MOVEQ    R1, No11
STR      R1, [R0, #(SRAM_IU_count-SRAM_IU_fak)]
ADD      R1, R0, R1, LSL #2

```

```

;Усреднение Тока

```

```

LDRH     R4, [R1]
STRH     R2, [R1]
LDR      R6, [R0, #(SRAM_I_summ-SRAM_IU_fak)]
ADD      R6, R2
SUB      R6, R4
STR      R6, [R0, #(SRAM_I_summ-SRAM_IU_fak)]
MOV      R12, #64*20
UDIV     R4, R6, R12
STR      R4, [R0, #(SRAM_I_fak-SRAM_IU_fak)] ;Ifak

```

```

;Усреднение Напряжения

```

```

LDRH     R5, [R1, #2]
STRH     R3, [R1, #2]
LDR      R7, [R0, #(SRAM_U_summ-SRAM_IU_fak)]
ADD      R7, R3
SUB      R7, R5
STR      R7, [R0, #(SRAM_U_summ-SRAM_IU_fak)]
MOV      R12, #64*40
UDIV     R5, R7, R12
STR      R5, [R0, #(SRAM_U_fak-SRAM_IU_fak)] ;Ufak

```

```

;.....

```

```

Inter_ADC1_Regul_I ;Управление ШИМ по Току

```

```

Inter_ADC1_Pauses

```

```

TST      Rsos, #Rsos_Gash_Dugi    ; Гашения дуги: 0 - ОТКЛ, 1 - ВКЛ

```

```

BEQ      Inter_ADC1_Sticking

```

```

CMP      R6, #64*20*1             ;Ifak <= 64 * 20 * 1A

```

```

BLS      Inter_ADC1_Sticking

```

```

;Проверка на обрыв дуги
CMP     R5, #30
BLS     Inter_ADC1_Sticking      ;Ufak <= 30V

;Ufak > 30V - обрыв дуги
ORR     Rsos, #Rsos_Pauses ;Установить признаки: Пауза
;Включить TIM4 (Время установки Паузы)
MOV32   R0, #TIM4
MOV     R1, #TIM_EGR_UG ;Генерация обновления
STR     R1, [R0,#TIM_EGR]
MOV     R1, #TIM_CR1_CEN ; Счётчик включить
STR     R1, [R0,#TIM_CR1]
B       Inter_ADC1_Regul_Ic

```

Inter_ADC1_Sticking

```

;Проверка на залипание
MOV32   R0, #SRAM_Umin
LDR     R7, [R0] ;Umin
CMP     R5, R7
BHI     Inter_ADC1_noSticking    ;Ufak > Umin

;Ufak <= Umin - Залипание электрода
ORR     Rsos, #Rsos_Sticking ;Установить признак: Залипания электрода
;Включить TIM2 (Время установки Imax)
MOV32   R0, #TIM2
MOV     R1, #TIM_EGR_UG ;Генерация обновления
STR     R1, [R0,#TIM_EGR]
MOV     R1, #TIM_CR1_CEN ; Счётчик включить
STR     R1, [R0,#TIM_CR1]

```

Inter_ADC1_noSticking

```

;Ufak > Umin - нет Залипание электрода
BIC     Rsos, #Rsos_Sticking ;Сбросить признак: Залипания электрода
MOV32   R0, #TIM2
STR     Noll, [R0,#TIM_CR1] ;Стоп Счётчика TIM2
STR     Noll, [R0,#TIM_CNT]

```

Inter_ADC1_Regul_Ic

```

MOV32   R0, #SRAM_IU_fak
LDR     R1, [R0,#(SRAM_Izad-SRAM_IU_fak)]

TST     Rsos, #Rsos_Pauses ;Ufak >= 30V - начало обрыва дуги
MOVNE   R1, #0

TST     Rsos, #Rsos_Sticking ;Ufak <= Umin - Залипание электрода: Есть
LDRNE   R1, [R0,#(SRAM_Imax-SRAM_IU_fak)]

MOV     R12, #20
MUL     R1, R12 ;Izad*20
ADD     R1, #5
LDR     R2, [R0,#(SRAM_Iout-SRAM_IU_fak)]

```

```

Kp_I EQU 20 ; Пропорциональный коэффициент
Ki_I EQU 5 ; Интегральный коэффициент
Kd_I EQU 2 ; Дифференциальный коэффициент
Del_I EQU 10 ; Делитель

```

```

MOV32   R8, #((Kd_I<<24) + (Ki_I<<16) + (Kp_I<<8) + Del_I)
MOV32   R0, #SRAM_et_I
BL     PIDreg
PUSH    {R2}

```

```

;.....

Inter_ADC1_Regul_U ;Управление ШИМ по Напряжению

        MOV32      R0, #SRAM_IU_fak
        LDR        R1, [R0,#(SRAM_Uzad-SRAM_IU_fak)]
        MOV        R12, #40
        MUL        R1, R12
        ADD        R1, #20
        LDR        R2, [R0,#(SRAM_Uout-SRAM_IU_fak)]

Kp_U    EQU    20    ; Пропорциональный коэффициент
Ki_U    EQU    5     ; Интегральный коэффициент
Kd_U    EQU    2     ; Дифференциальный коэффициент
Del_U   EQU    10    ; Делитель

        MOV32      R8, #((Kd_U<<24) + (Ki_U<<16) + (Kp_U<<8) + Del_U)
        MOV32      R0, #SRAM_et_U
BL PIDreg

;.....

        ;Код управления ШИМ - током
        POP        {R4}
        ;Код управления ШИМ - напряжением
        MOV        R5, R2

        CMP        R4, R5
        MOVLS      R2, R4 ; R4 =< R5
        MOVHI      R2, R5 ; R4 > R5

        MOV32      R0, #TIM1
        MOV        R1, #1200
        SUB        R1, R1,R2
        STR        R1, [R0,#TIM_CCR1]
        STR        R2, [R0,#TIM_CCR2]

;.....

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

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

        ENDP

;.....

PIDreg    PROC    ;ПИД-Регулятор.
            ; Вход: R1 - Заданные данные, R2 - Фактические данные, R0 - адрес SRAM_et_
            ; Выход: R2

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

            ;Вычисление рассогласования: e(t)
            SUB        R3, R1,R2 ; e(t)

;.....

```

```

PID_p ;Пропорциональная составляющая.  $P(t) = K_p * e(t)/\Delta t$ ;
    SBFX      R4, R8,#8,#8 ;Kp
    MUL       R4, R3
    SDIV      R4, R4, R7    ; P(t)
    SSAT      R5, #17,R4    ;Ограничение R4
;.....
PID_in ;Интегральная составляющая.  $I(t) = I(t-1) + K_i * e(t)/\Delta t$ 
    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.  $D(t) = K_d * \{e(t) - e(t-1)\}/\Delta t$ ;
    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
    ADD       R2, R2,R6     ; u(t)

    ;Ограничение  $0 < R2 < 512$ 
    ASRS      R2, #7
    ADDCS     R2, Unit
    USAT      R2, #9,R2

    BX        LR

    ENDP

;.....

Inter_TIM2 PROC align ; Конец Времени установки Iмах

    PUSH      {LR}

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

    MOV32     R0, #TIM2

    LDR       R2, [R0,#TIM_SR]
    STR       R2, [R0,#TIM_SR] ; Очистка состояния TIM2

    BIC       R2, #R2s_Sticking ;Сбросить признак: Залипания электрода

    STR       R2, [R0,#TIM_CR1] ;Стоп Счетчика TIM2
    STR       R2, [R0,#TIM_CNT]

    POP       {LR}
    BX        LR

    ENDP

;.....

```

```

Inter_TIM4      align
PROC            ; Конец Времени установки Паузы

PUSH            {LR}

MOV32           R0, #NVIC_ICPR0      ; Сбросить признаки отложенных прерываний 0...32
MOV            R1, #(1<<TIM4_IRQn)
STR            R1, [R0]

MOV32           R0, #TIM4

LDR            R2, [R0,#TIM_SR]
STR            R0, [R0,#TIM_SR] ; Очистка состояния TIM4

BIC            Rsos, #Rsos_Pauses
ORR            Rsos, #Rsos_Proh_Pauses ; 1 - Запрещение отработки Паузы

STR            R0, [R0,#TIM_CR1]      ; Стоп Счетчика TIM4
STR            R0, [R0,#TIM_CNT]

POP            {LR}
BX             LR

ENDP

```

```

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

```

```

OUT_OnOff      align
PROC            ; ВКЛ/ОТКЛ выхода аппарата

PUSH            {LR}

TST            Rsos, #Rsos_tC_max ; Флаг Rsos: 1 - Превышение температуры.
BNE OUT_OnOff_OFF

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

TST            R1, #SA_PA15 ; "0": ВКЛ, "1": ОТКЛ - Выхода аппарата
BICNE         Rsos, #Rsos_SA_Trq
BNE OUT_OnOff_OFF

TST            Rsos, #Rsos_SA_Trq
BNE OUT_OnOff_exit

ORR            Rsos, #Rsos_SA_Trq

ANDS           R2, Rsos, #0x0f
BEQ OUT_OnOff_OFF

```

```

OUT_OnOff_ON    ; ВКЛ выхода аппарата

BFC            Rsos, #Rsos_Rej, #4 ; Очистка Rsos[3...0] - Режим
ORR            Rsos, #1 ; Режим: "1" - Индикация ТОК

UBFX           R2, Rsos, #0, #4
MOV32          R1, #Tab_Rej_Ind
LDR            R1, [R1, R2, LSL #2]
BLX           R1

TST            Rsos, #Rsos_Perm_RL
BEQ OUT_OnOff_OFF

```

```

;ВКЛ силового реле зарядки конденсаторов
MOV32    R0, #GPIOA
MOV      R1, #RL_PA11
STR      R1, [R0,#GPIO_BSRR]
BL Zader_50mC ; Ожидание замыкания контактов реле

```

```

;ВКЛ выхода ШИМ
MOV32    R0, #GPIOB
MOV      R1, #PWM_ON_PB3
STR      R1, [R0,#GPIO_BSRR]
MOV32    R0, #GPIOA
MOV      R1, #PWM_ONL_PA10
STR      R1, [R0,#GPIO_BRR]

```

```

;Разрешение глобального прерывания EXTI1_IRQn = 7
MOV      R1, #(1<<EXTI1_IRQn)
MOV32    R0, #NVIC_ICER0 ;Запрет внешние прерывания 0...32
STR      R1, [R0]
MOV32    R0, #NVIC_ICPR0 ; Сбросить признаки отложенных прерываний 0...32
STR      R1, [R0]

```

```

B OUT_OnOff_exit

```

```

OUT_OnOff_OFF ;ОТКЛ выхода аппарата
;ОТКЛ силового реле зарядки конденсаторов и ШИМ

```

```

MOV32    R0, #GPIOB
MOV      R1, #PWM_ON_PB3
STR      R1, [R0,#GPIO_BRR]
MOV32    R0, #GPIOA
MOV      R1, #RL_PA11
STR      R1, [R0,#GPIO_BRR]
MOV      R1, #PWM_ONL_PA10
STR      R1, [R0,#GPIO_BSRR]

```

```

;Разрешение глобального прерывания EXTI1_IRQn = 7
MOV      R1, #(1<<EXTI1_IRQn)
MOV32    R0, #NVIC_ICPR0 ; Сбросить признаки отложенных прерываний 0...32
STR      R1, [R0]
MOV32    R0, #NVIC_ISER0 ;Разрешение внешние прерывания 0...32
STR      R1, [R0]

```

```

OUT_OnOff_exit

```

```

POP      {LR}
BX       LR

```

```

ENDP

```

```

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

```

```

PtC_max align
PROC ;Превышение температуры

```

```

PUSH     {LR}

```

```

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

```

```

TST      R1, #tC_max_PA3 ; "1": Норма температуры
BNE PtC_max_norma

```

```

PtC_max_max

```

```

TST      Rsos, #Rsos_tC_max ; Флаг Rsos: 1 - Превышение температуры.

```

```

BNE PtC_max_exit

ORREQ      Rsos, #Rsos_tC_max ; Флаг Rsos: 1 - Превышение температуры.
;Установка индикации Нагрев: MAX
BFIEQ      Rsos, Rsos,#Rsos_Rej_Copy,#4 ;Запоминание режима
BFCEQ      Rsos, #Rsos_Rej,#4 ;Очистка Rsos[3...0] - Режим
ORREQ      Rsos, #7 ;Режим: Индикация Превышение температуры
B PtC_max_Ind ;Превышение температуры

```

```
PtC_max_norma
    TST      Rsos, #Rsos_tC_max ; Флаг Rsos: 1 - Превышение температуры.
    BEQ PtC_max_exit
```

```
BICNE      Rsos, #Rsos_tC_max
;Возврат  индикации к Режиму
UBFXNE     R2, Rsos, #Rsos_Rej_Copy, #4
BFCNE     Rsos, #Rsos_Rej, #4
ORRNE     Rsos, R2
```

```
PtC_max_Ind
    UBFX      R2, Rsos,#0,#4
    MOV32     R1, #Tab_Rej_Ind
    LDR       R1, [R1,R2,LSL #2]
    BLX      R1
```

PtC_max_exit	POP	{LR}
	BX	LR

ENDP

))))))

```

Inter_ENCEx      align      PROC      ;Прерывание от энкодера EXTII1

                PUSH          {LR}

MOV32            R0, #NVIC_ICPR0      ; Сбросить признаки отложенных прерываний 0...32
MOV              R1, #(1<<EXTII1_IRQn)
STR              R1, [R0]

MOV32            R0, #EXTI
MOV32            R1, #(EXTI_PR_PR1+EXTI_PR_PR2) ;Внешнее прерывание Сброс
STR              R1, [R0,#EXTI_PR]

TST              Rsos, #Rsos_tC_max   ; Флаг Rsos: 1 - Превышение температуры.
BNE Inter_ENCEx_exit

UBFX             R2, Rsos,#0,#4
MOV32            R1, #Rej_Ind
LDR              R1, [R1,R2,LSL #2]
BLX              R1

ORR              Rsos, #Rsos_W_Rezerv ; Флаг Rsos: 1 - Разрешение записи данных в
резервную область

```

```
Inter_ENCEx_exit
    POP     {LR}
    BX      LR
```

ENDP

;

```

;Перезагрузка Индикатора
Rej_Ind DCD Ind_Alekr ;Режим Выбора Электродов
        DCD Ind_Svarka_IU ;Работа Режим Сварки (Задание Стабилизации Тока и
Напряжения)
        DCD Ind_Forsaj_T ;Режим Настройки параметров: Залипание электрода
        DCD Ind_Forsaj_U ;Режим Настройки параметров: Залипание электрода
        DCD Ind_Zajigan_I ;Режим Настройки параметров: Зажигание электрода
        DCD Ind_Zajigan_U ;Режим Настройки параметров: Зажигание электрода
        DCD Ind_Gash_Dugi ;Режим Настройки параметров: Гашения дуги
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                align
Ind_Alektr PROC ;Режим Выбора Электродов

                PUSH {LR}

                UBFX R2, Rsos,#4,#4

                MOV32 R0, #GPIOA
                LDR R1, [R0,#GPIO_IDR]
                TST R1, #GPIO_IDR IDR0
                ADDNE R2, Unit
                SUBEQ R2, Unit
                ;Ограничение R2: 0...5
                USAT R2,#4,R2
                CMP R2, #5
                MOVHI R2, #5 ; HI: R2 > 5
                BFI Rsos, R2,#4,#4

                CBNZ R2, Ind_Alektr_Alekr

Ind_Alektr_User_Zad ; Пользователь
                MOV32 R0, #BKP+BKP_DR1
                MOV32 R1, #SRAM_Izad
                BL Forward_Data ;Переслать Данные: R0 - адрес источника, R1 - адрес приемника
                B Ind_Alektr_Mark

Ind_Alektr_Alekr
                MOV32 R0, #Alekr_User ;Пользователь
                ADD R0, R2,LSL #2
                LDR R0, [R0]
                MOV32 R1, #SRAM_Izad
                BL Forward_Data ;Переслать Данные: R0 - адрес источника, R1 - адрес приемника

Ind_Alektr_Mark
                MOV32 R1, #Alekr_User_M ;Пользователь
                ADD R1, R2,LSL #2
                LDR R1, [R1]

                BL Buff_NoNull ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

                STR R1, [R0]

                POP {LR}
                BX LR

                ENDP
;

```

```

align
Alektr_User      DCD      Alektr_User_Zad      ; Пользователь
Alektr_40        DCD      Alektr_40_Zad       ; d=4,0 мм
Alektr_30        DCD      Alektr_30_Zad       ; d=3,0 мм
Alektr_25        DCD      Alektr_25_Zad       ; d=2,5 мм
Alektr_20        DCD      Alektr_20_Zad       ; d=2,0 мм
Alektr_16        DCD      Alektr_16_Zad       ; d=1,6 мм

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

Alektr_User_Zad   ; Пользователь
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

Alektr_40_Zad     ; d=4,0 мм
DCD      125      ; Izad
DCD      130      ; Uxad
DCD      10       ; Timp*100
DCD      40       ; Umin*10
DCD      160      ; Imax

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

Alektr_30_Zad     ; d=3,0 мм
DCD      95       ; Izad
DCD      130      ; Uxad
DCD      10       ; Timp*100
DCD      50       ; Umin*10
DCD      160      ; Imax

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

Alektr_25_Zad     ; d=2,5 мм
DCD      70       ; Izad
DCD      130      ; Uxad
DCD      10       ; Timp*100
DCD      60       ; Umin*10
DCD      120      ; Imax

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

Alektr_20_Zad     ; d=2,0 мм
DCD      55       ; Izad
DCD      130      ; Uxad
DCD      10       ; Timp*100
DCD      70       ; Umin*10
DCD      90       ; Imax

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

Alektr_16_Zad     ; d=1,6 мм
DCD      32       ; Izad
DCD      130      ; Uxad
DCD      10       ; Timp*100
DCD      80       ; Umin*10
DCD      60       ; Imax

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

Alektr_User_M     align
Alektr_40_M       DCD      0x48000000      ; Пользователь
Alektr_30_M       DCD      0x50000000      ; d=4,0 мм
Alektr_25_M       DCD      0x58000000      ; d=3,0 мм
Alektr_20_M       DCD      0x60000000      ; d=2,5 мм

```

```

Alektr_20_M      DCD    0x68000000    ; d=2,0 мм
Alektr_16_M      DCD    0x70000000    ; d=1,6 мм

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

W_Rezerv          align
PROC              ; Запись данных в резервную область
PUSH              {LR}

MOV32 R0, #PWR
MOV              R1, #PWR_CR_DBP          ; Отключить: защиту резервного домена от
записи
STR              R1, [R0,#PWR_CR]

MOV32            R0, #SRAM_Izad
MOV32            R1, #BKP+BKP_DR1
BL Forward_Data  ;Переслать Данные: R0 - адрес источника, R1 - адрес приемника

MOV32            R0, #PWR
STR              R0, [R0,#PWR_CR] ; Включить: защиту резервного домена от записи

POP              {LR}
BX               LR

ENDP

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

Forward_Data      align
PROC              ;Переслать Данные
;R0 - адрес источника, R1 - адрес приемника
PUSH              {R2}

LDR              R2, [R0]
STR              R2, [R1]
LDR              R2, [R0,#4]
STR              R2, [R1,#4]
LDR              R2, [R0,#8]
STR              R2, [R1,#8]
LDR              R2, [R0,#12]
STR              R2, [R1,#12]
LDR              R2, [R0,#16]
STR              R2, [R1,#16]

POP              {R2}
BX               LR

ENDP

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

Ind_Svarka_IU     align
PROC              ;Работа Режим Сварки (Задание Стабилизации Тока и Напряжения)

PUSH              {LR}

TST              Rsos, #Rsos_Dial_IU      ;Rsos_Dial_IU Выбор набора: 0 - Ток, 1 -
Напряжение
BNE Ind_Svarka_IU_U

Ind_Svarka_IU_I   ;Работа Режим Сварки (Задание Стабилизации Тока)
MOV32            R1, #SRAM_Izad
LDR              R2, [R1]

```

```

MOV32    R0, #GPIOA
LDR      R3, [R0,#GPIO_IDR]
TST      R3, #GPIO_IDR_IDR0
ADDEQ    R2, Unit
SUBNE    R2, Unit
;Ограничение R2: 1...135 A.
CMP      R2, #135
MOVHI    R2, #135      ; HI: R2 > 135
CMP      R2, #1
MOVLO    R2, #1        ; HI: R2 < 1
STR      R2, [R1]

```

BL Buff_NoNoll ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

```

MOV32    R1, #0x08000000
ORR      R1, R2
STR      R1, [R0]

```

```

MOV32    R1, #0x48000000
STR      R1, [R0,#4]

```

B Ind_Svarka_IU_exit

Ind_Svarka_IU_U ;Работа Режим Сварки (Задание Напряжения)

```

PUSH     {LR}

```

```

MOV32    R1, #SRAM_Uzad
LDR      R2, [R1]

```

```

MOV32    R0, #GPIOA
LDR      R3, [R0,#GPIO_IDR]
TST      R3, #GPIO_IDR_IDR0
ADDEQ    R2, Unit
SUBNE    R2, Unit
;Ограничение R2: 1...130 B.
CMP      R2, #130
MOVHI    R2, #130      ; HI: R2 > 130
CMP      R2, #1
MOVLO    R2, #1        ; HI: R2 < 1
STR      R2, [R1]

```

BL Buff_NoNoll ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

```

MOV32    R1, #0x18000000
ORR      R1, R2
STR      R1, [R0]

```

```

MOV32    R1, #0x50000000
STR      R1, [R0,#4]

```

Ind_Svarka_IU_exit

```

POP      {LR}
BX       LR

```

ENDP

;;

Ind_Forsajj_T **align PROC** ;Режим Настройки параметров: Залипание электрода

```

PUSH     {LR}

```

```
MOV32    R1, #SRAM_Timp
LDR      R2, [R1]
```

```
;Время повышение тока при залипании электрода
```

```
MOV32    R0, #GPIOA
LDR      R3, [R0,#GPIO_IDR]
TST      R3, #GPIO_IDR_IDR0
ADDEQ    R2, Unit
SUBNE    R2, Unit
```

```
;Ограничение R2: 0,1...1,9 С.
```

```
CMP      R2, #99
MOVHI    R2, #99      ; HI: R2 > 99
CMP      R2, #1
MOVLO    R2, #1      ; HI: R2 < 1
STR      R2, [R1]
```

```
MOV32    R0, #TIM2
MOV      R3, #8      ;Тимп = 0,01...0,99 С при f = 80 МГц
MUL      R3, R2
STR      R3, [R0,#TIM_CCR1]
```

```
BL Buff_NoNoll      ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0
```

```
MOV32    R1, #0x08000000
ORR      R1, R2
STR      R1, [R0]
```

```
MOV32    R1, #0x48000000
STR      R1, [R0,#4]
```

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

```
ENDP
```

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

```
Ind_Forsaj_U      align
PROC              ;Режим Настройки параметров: Залипание электрода
```

```
PUSH     {LR}
```

```
MOV32    R1, #SRAM_Umin
LDR      R2, [R1]
```

```
MOV32    R0, #GPIOA
LDR      R3, [R0,#GPIO_IDR]
TST      R3, #GPIO_IDR_IDR0
ADDEQ    R2, Unit
SUBNE    R2, Unit
```

```
;Ограничение R2: 3,0...9,9 С.
```

```
CMP      R2, #99
MOVHI    R2, #99      ; HI: R2 > 99
CMP      R2, #30
MOVLO    R2, #30      ; HI: R2 < 30
STR      R2, [R1]
```

```
BL Buff_NoNoll      ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0
```

```
MOV32    R1, #0x10000000
ORR      R1, R2
STR      R1, [R0]
```

```
MOV32    R1, #0x50000000
STR      R1, [R0,#4]
```

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

```
ENDP
```

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

```
Ind_Zajigan_I      align PROC
```

```
PUSH      {LR}
```

```
MOV32    R1, #SRAM_Imax
LDR      R2, [R1]
```

```
MOV32    R0, #GPIOA
LDR      R3, [R0,#GPIO_IDR]
TST      R3, #GPIO_IDR_IDR0
ADDEQ    R2, Unit
SUBNE    R2, Unit
```

```
;Ограничение R2: Imax...160 A.
```

```
CMP      R2, #160
MOVHI    R2, #160 ; HI: R2 > 160
```

```
;Не допускать Imax < Izad
```

```
MOV32    R3, #SRAM_Izad
LDR      R3, [R3]
```

```
CMP      R2, R3
MOVLO    R2, R3 ; HI: R2 < Izad
STR      R2, [R1]
```

```
BL Buff_NoNoll ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0
```

```
MOV32    R1, #0x08000000
ORR      R1, R2
STR      R1, [R0]
```

```
MOV32    R1, #0x48000000
STR      R1, [R0,#4]
```

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

```
ENDP
```

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

```
Ind_Zajigan_U      align PROC ;ВКЛ умножителя
```

```
PUSH      {LR}
```

```
MOV32    R0, #GPIOA
LDR      R2, [R0,#GPIO_IDR]
MOV      R3, #RU_PA12
```

```
TST      R2, #GPIO_IDR_IDR0
BICEQ    Rsos, #Rsos_Zajigan_U
ORRNE    Rsos, #Rsos_Zajigan_U
```

```

TST      Rsos, #Rsos_Zajigan_U
STREQ    R3, [R0,#GPIO_BRR]
MOV32EQ  R1, #0x58000000
STRNE    R3, [R0,#GPIO_BSRR]
MOV32NE  R1, #0x60000000

```

```
BL Buff_NoNol1      ;Отыскание пустой ячейки буфера
```

```

STR      R1, [R0]

MOV32    R1, #0x50000000
STR      R1, [R0,#4]

POP      {LR}
BX       LR

```

```
ENDP
```

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

```

Ind_Gash_Dugi      align
PROC              ;Режим Настройки параметров: Гашения дуги

```

```

PUSH      {LR}

TST      Rsos, #Rsos_Gash_Dugi
ORREQ    Rsos, #Rsos_Gash_Dugi      ; Обработка гашения дуги: 1
MOV32EQ  R1, #0x58000000
BICNE    Rsos, #Rsos_Gash_Dugi
MOV32NE  R1, #0x60000000

```

```
BL Buff_NoNol1      ; Отыскание пустой ячейки буфера
```

```

STR      R1, [R0]

MOV32    R1, #0x50000000
STR      R1, [R0,#4]

POP      {LR}
BX       LR

```

```
ENDP
```

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

```

Ind_Gradus      align
PROC            ;Режим Перегрев

```

```
PUSH      {LR}
```

```
BL Buff_NoNol1      ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0
```

```

MOV32    R1, #0x40000000
STR      R1, [R0]

POP      {LR}
BX       LR

```

```
ENDP
```

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

```
align
```

```

Inter_ENCsbEx      PROC    ; Прерывание EXTI2_IRQHandler от Кнопки Энкодера 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_PR1+EXTI_PR_PR2) ;Внешнее прерывание Сброс
    STR             R1, [R0,#EXTI_PR]

    TST             Rsos, #Rsos_tC_max   ; Флаг Rsos: 1 - Превышение температуры.
    BNE Inter_ENCsbEx_exit

    ADD             Rsos, Unit
    UBFX            R2, Rsos,#0,#3       ;Ограничение Режимов: 0...7
    CMP             R2, #7
    MOVEQ           R2, No11

    CMP             R2, #2
    BNE Inter_ENCsbEx_C

    TST             Rsos, #Rsos_Dial_IU   ;Rsos_Dial_IU - Выбор набора: 0 - Ток, 1 -
Напряжение
    ORREQ           Rsos, #Rsos_Dial_IU
    MOVEQ           R2, #1
    BICNE           Rsos, #Rsos_Dial_IU

    TST             Rsos, #Rsos_SA_Trg    ;SA: 0 - ОТКЛЮЧЕН, 1 - ВКЛЮЧЕН
    MOVNE           R2, #1

Inter_ENCsbEx_C
    BFI             Rsos, R2,#0,#4
    MOV32           R1, #Tab_Rej_Ind
    LDR             R1, [R1,R2,LSL #2]
    BLX            R1

Inter_ENCsbEx_exit
    POP             {LR}
    BX              LR

    ENDP

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

    align

Tab_Rej_Ind        ;Перезагрузка Режимы Индикатора
    DCD            Ind_Rej_Alektr         ;"0" Режим Выбора Электродов
    DCD            Ind_Rej_Svarka_IU      ;"1" Работа Режим Сварки, Режим Стабилизации Тока и
Стабилизации Напряжения
    DCD            Ind_Rej_Forsaj_T       ;"2" Режим Настройки параметров: Залипание электрода
    DCD            Ind_Rej_Forsaj_U       ;"3" Режим Настройки параметров: Залипание электрода
    DCD            Ind_Rej_Zajigan_I      ;"4" Режим Настройки параметров: Зажигание электрода
    DCD            Ind_Rej_Zajigan_U      ;"5" Режим Настройки параметров: Зажигание электрода
    DCD            Ind_Rej_Gash_Dugi      ;"6" Режим Настройки параметров: Гашения дуги
    DCD            Ind_Rej_Gradus         ;"7" Индикация Превышение температуры
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

    align

Ind_Rej_Alektr     PROC    ;Режим Выбора Электродов

    PUSH            {LR}

```

```

BL Buff_NoNol1      ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

MOV32               R1, #0x40000000
STR                 R1, [R0]

MOV32               R1, #0x48000000      ;Пользователь
STR                 R1, [R0,#4]

POP                 {LR}
BX                  LR

ENDP

```

```

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

```

```

align
Ind_Rej_Svarka_IU   PROC      ;Работа Режим Сварки (Стабилизации Тока), Режим Стабилизации
Напряжения

```

```

PUSH                {LR}

BL Buff_NoNol1      ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

MOV32               R1, #0x40000000
STR                 R1, [R0]

MOV32               R1, #0x08000000
MOV32               R2, #SRAM_Izad
LDR                 R2, [R2]
ORR                 R1, R2
STR                 R1, [R0,#4]

MOV32               R1, #0x10000000
MOV32               R2, #SRAM>Ifak
LDR                 R2, [R2]
ORR                 R1, R2
STR                 R1, [R0,#8]

MOV32               R1, #0x18000000
MOV32               R2, #SRAM_Uzad
LDR                 R2, [R2]
ORR                 R1, R2
STR                 R1, [R0,#12]

MOV32               R1, #0x20000000
MOV32               R2, #SRAM_Ufak
LDR                 R2, [R2]
ORR                 R1, R2
STR                 R1, [R0,#16]

MOV32               R1, #0x48000000
STR                 R1, [R0,#20]

POP                 {LR}
BX                  LR

ENDP

```

```

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

```

```

align
Ind_Rej_Forsaj_T    PROC      ;Режим Настройки параметров: Залипание электрода

```

PUSH {LR}

```

BL Buff_NoNull    ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0
MOV32             R1, #0x40000000
STR               R1, [R0]

MOV32             R1, #SRAM_Timp
LDR               R2, [R1]
MOV32             R1, #0x08000000
ORR               R1, R2
STR               R1, [R0,#4]

MOV32             R1, #SRAM_Umin
LDR               R2, [R1]
MOV32             R1, #0x10000000
ORR               R1, R2
STR               R1, [R0,#8]

MOV32             R1, #0x48000000
STR               R1, [R0,#12]

POP               {LR}
BX                LR

ENDP

```

;;;

align

Ind_Rej_Forsaj_U PROC ;Режим Настройки параметров: Залипание электрода

PUSH {LR}

```

BL Buff_NoNull    ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

MOV32             R1, #SRAM_Umin
LDR               R2, [R1]
MOV32             R1, #0x10000000
ORR               R1, R2
STR               R1, [R0]

MOV32             R1, #0x50000000
STR               R1, [R0,#4]

POP               {LR}
BX                LR

ENDP

```

;;;

align

Ind_Rej_Zajigan_I PROC ;Режим Настройки параметров: Зажигание электрода

PUSH {LR}

```

BL Buff_NoNull    ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

MOV32             R1, #0x40000000
STR               R1, [R0]

MOV32             R1, #SRAM_Imax

```

```

LDR      R2, [R1]
MOV32    R1, #0x08000000
ORR      R1, R2
STR      R1, [R0,#4]

TST      Rsos, #Rsos_Zajigan_U
MOV32EQ  R1, #0x58000000
MOV32NE  R1, #0x60000000
STR      R1, [R0,#8]

MOV32    R1, #0x48000000
STR      R1, [R0,#12]

POP      {LR}
BX       LR

ENDP

```

```

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

```

```

                                align
Ind_Rej_Zajigan_U      PROC      ;Режим Настройки параметров: Зажигание электрода

                                PUSH      {LR}

                                BL Buff_NoNoll      ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

                                TST      Rsos, #Rsos_Zajigan_U
                                MOV32EQ  R1, #0x58000000
                                MOV32NE  R1, #0x60000000
                                STR      R1, [R0]

                                MOV32    R1, #0x50000000
                                STR      R1, [R0,#4]

                                POP      {LR}
                                BX       LR

                                ENDP

```

```

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

```

```

                                align
Ind_Rej_Gash_Dugi      PROC      ;Режим Настройки параметров: Гашения дуги

                                PUSH      {LR}

                                BL Buff_NoNoll      ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

                                MOV32    R1, #0x40000000
                                STR      R1, [R0]

                                TST      Rsos, #Rsos_Gash_Dugi
                                MOV32EQ  R1, #0x60000000
                                MOV32NE  R1, #0x58000000
                                STR      R1, [R0,#4]

                                MOV32    R1, #0x50000000
                                STR      R1, [R0,#8]

                                POP      {LR}
                                BX       LR

```

ENDP

;;;

```

Ind_Rej_Gradus      align
                    PROC          ;Режим Перегрев

                    PUSH          {LR}

                    BL Buff_NoNoll ;Отыскание пустой ячейки буфера, выход: адрес пустой ячейки в R0

                    MOV32         R1, #0x40000000
                    STR           R1, [R0]

                    POP           {LR}
                    BX           LR

                    ENDP

```

;;;

```

Buff_NoNoll         align
                    PROC          ;Отыскание пустой ячейки буфера
                                ; Выход R0 - адрес пустой ячейки буфера

                    PUSH          {R1, R3}

                    MOV           R3, #16
                    MOV32         R0, #SRAM_Buff_0

Buff_NoNoll_ck1
                    LDR           R1, [R0],#4
                    CBZ           R1, Buff_NoNoll_noll

                    SUB           R3, Unit
                    CBZ           R1, Buff_NoNoll_exit
                    B Buff_NoNoll_ck1

Buff_NoNoll_noll

                    SUB           R0, #4

Buff_NoNoll_exit
                    POP           {R1, R3}
                    BX           LR

                    ENDP

```

;;;

```

                    align

W_Buff_Ind          PROC          ; Перезапись по параметру в буферной памяти на индикатор

                    PUSH          {LR}

W_Buff_Ind_st
                    ;Вычисление Адреса Кода вывода Текста
                    MOV32         R0, #Tab_Rej_tab
                    UBFX          R2, Rsos,#0,#4      ;R2[3...0] <- Rsos[3...0]
                    LDR           R0, [R0,R2,LSL #2]
                    UBFX          R2, R1,#27,#5      ;R2[4...0] <- R1[31...27]
                    LDR           R0, [R0,R2,LSL #2] ;R0 - числа индикатора

                    TST           R1, #CMar_Que      ; Выбор записи числа или текста, управления в
индикатор

```

```

UBFXEQ      R2, R1, #0, #27      ;R2[26...0] <- R1[26...0] - загрузка двоичного
числа
BLEQ Form_Dec
BLNE Indic_W_cdseg

W_Buff_Ind_No11      ; Сдвиг сдвиг данных из старшего буфера в младший буфер

MOV32      R0, #SRAM_Buff_0
MOV32      R1, #SRAM_Buff_1
MOV        R3, #15
W_Buff_Ind_Sh
LDR        R2, [R1], #4
STR        R2, [R0], #4
SUBS      R3, Unit
BNE W_Buff_Ind_Sh
; Запись "0" в старший буфер SRAM_Buff_15
STR        No11, [R0]

; Сбросить: Буффер Переполнен
MOV32      R0, #GPIOA
MOV        R1, #GPIO_BSRR_BR8
STR        R1, [R0, #GPIO_BSRR]

;Проверка на"0" младшего буфера
MOV32      R0, #SRAM_Buff_0
LDR        R1, [R0]
CMP        R1, No11
BNE W_Buff_Ind_st

POP        {LR}
BX        LR

ENDP

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

Indic_W_cdseg      align
PROC      ;Перезапись из Памяти в Индикатор

PUSH      {LR}
PUSH      {R3, R4}

Indic_W_cdseg_ck1
LDRB      R1, [R0], #1      ;Адрес
CBZ      R1, Indic_W_dRet

BL W_CA_DD
BL BF_Zader_45mkC

TST      Rsos, #Rsos_Long_Comm ;Проверка на длинную (по времени ) команду
BLNE Zader_2mC

BIC      Rsos, #Rsos_Long_Comm

B Indic_W_cdseg_ck1

Indic_W_dRet
POP      {R3, R4}
POP      {LR}
BX      LR

ENDP

```

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

```

W_CA_DD      align
              PROC    ;П/п Записи Адреса/Данных в индикатор

              ;Опознавание кода Адреса
PUSH          {LR}
MOV32         R2, #GPIOB

CMP           R1, #EC
BNE W_CA_DD_d

MOV           R3, #(RS_PB7+R_W_PB6+E_PB4) ; R_W_PB6 -"0", RS_PB7 -"0", E_PB4 -"0"
STR           R3, [R2,#GPIO_BRR]
BIC           Rsos, #Rsos_Indic_DD
B W_CA_DD_ext

W_CA_DD_d
              ;Команда длинная (1,54 мС): Очистить дисплей
CMP           R1, #0x01
ORREQ        Rsos, #Rsos_Long_Comm      ;Установка признака длинной (по времени)

команды

              ;Опознавание кода Данных
CMP           R1, #DD
BNE W_CA_DD_ad

MOV           R3, #RS_PB7                ; RS_PB7 - "1"
STR           R3, [R2,#GPIO_BSRR]
ORR           Rsos, Rsos,#Rsos_Indic_DD
B W_CA_DD_ext

W_CA_DD_ad
TST           Rsos, #Rsos_Indic_DD
BEQ W_CA_DD_ad
              ; Перекодировки русских символов.
PUSH          {R0}
CMP           R1, #0x80
SUBHS        R1, R1,#0x80
MOV32HS      R0, #Tab_Transcoding
LDRHS        R1, [R0,R1]
POP           {R0}

W_CA_DD_ad
              ;Запись в индикатор
MOV           R3, #E_PB4                  ;Запись в индикатор: начало
STR           R3, [R2,#GPIO_BSRR]
NOP
NOP

MOV32         R2, #GPIOB
LSL           R1, R1,#8
LDR           R3, [R2,#GPIO_ODR]
BFI           R1, R3,#0,#8
STR           R1, [R2,#GPIO_ODR] ; Запись байта данных
BL BF_Zader_1mkC

MOV           R3, #E_PB4                  ;Запись в индикатор: конец
MOV32         R2, #GPIOB
STR           R3, [R2,#GPIO_BRR]
NOP
NOP

W_CA_DD_ext

```

```

POP      {LR}
BX       LR

ENDP

```

```

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

```

```

align
BF_Zader_1mkC PROC ;Задержка на 1мкс при f = 80 МГц
    PUSH      {R12}

    MOV       R12, #20
BF_Zader_1mkC_ck1 ;Цикл 4 такта
    SUBS     R12, R12, #1
    BNE BF_Zader_1mkC_ck1

    POP      {R12}
    BX       LR

ENDP

```

```

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

```

```

align
BF_Zader_45mkC PROC ;Задержка на 45мкс при f = 80 МГц
    PUSH      {R12}
    MOV       R12, #900
BF_Zader_45mkC_ck1 ;Цикл 4 такта
    SUBS     R12, R12, #1
    BNE BF_Zader_45mkC_ck1

    POP      {R12}
    BX       LR

ENDP

```

```

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

```

```

align
Zader_2mC PROC ;Задержка на 10 мс при f = 80 МГц

    PUSH      {R12}
    MOV32     R12, #32000
Zader_2mC_ck1 ;Цикл 5 такта
    NOP
    SUBS     R12, R12, #1
    BNE Zader_2mC_ck1

    POP      {R12}
    BX       LR

ENDP

```

```

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

```

```

align
Zader_50mC PROC ;Задержка на 50 мс при f = 80 МГц

    PUSH      {R12}
    MOV32     R12, #80000
Zader_50mC_ck1 ;Цикл 5 такта
    NOP
    SUBS     R12, R12, #1
    BNE Zader_50mC_ck1

```



```

STR      Noll, [R1,#0x10]
STR      Noll, [R1,#0x14]
STR      Noll, [R1,#0x16]

```

Form_Dec_Ci ; Перезапись Кода вывода Текста числа индикатора в SRAM

```

LDRB     R3, [R0],#1
STRB     R3, [R1],#1
CMP      R3, #0 ;Проверка на 0 (конец Кода вывода числа)
BNE Form_Dec_Ci

```

```

LDRB     R5, [R0] ;Смещение+zpt1..3+Pr00
MOV      R4, #3 ;Количество символов: разбиение по 3
MOV      R2, #10 ;Количество символов: Мах
MOV      R6, #0x20 ; 'Код пробела'
MOV32    R0, #SRAM_Func_Dec ; Адрес (младший разряд) области памяти 8-
разрядного десятичного числа

```

```

MOV32    R1, #SRAM_Form_Dec ; Память форматирования десятичного числа
TST      R5, #zpt1 ;Проверка на признак запятой
BEQ Form_Dec_No_zpt

```

```

LDRB     R3, [R0],#1
STRB     R3, [R1],#1

```

```

TST      R5, #(zpt2-zpt1) ;Проверка на признак запятой
LDRBNE   R3, [R0],#1
STRBNE   R3, [R1],#1
BNE Form_Dec_zpt

```

```

TST      R5, #(zpt3-zpt1) ;Проверка на признак запятой
LDRBNE   R3, [R0],#1
STRBNE   R3, [R1],#1

```

Form_Dec_zpt

```

MOV      R3, #0x2c ;Установка ', '
STRB     R3, [R1],#1
SUB      R2, R2, Unit

```

Form_Dec_No_zpt ;Нет запятой

```

LDRB     R3, [R0],#1 ;Перезапись, если R4 не равно "0"
STRB     R3, [R1],#1
SUBS     R4, R4, Unit
ITT EQ
MOVEQ    R4, #3 ;Количество символов: разбиение по 3
STRBEQ   R6, [R1],#1 ;Запись 'пробела'
SUBS     R2, R2, Unit
BNE Form_Dec_No_zpt

```

```

MOV      R2, #10
MOV32    R1, #(SRAM_Form_Dec+11)
STRB     Noll, [R1],#-1

```

Form_Dec_Noll_zpt ;Замена старших разрядов со значением "0" на "Noll" до разряда не равного "0".

```

LDRB     R3, [R1],#-1
CMP      R3, #0x2c ;Проверка на ', '
BEQ Form_Dec_zpt_Noll

```

```

BICS     R3, R3, #0x30 ; '0'
BNE Form_SRAM_D ;Переход, если код числа не равен 0x30 или 0x20

```

```

STRB     R3, [R1],#1]
SUBS     R2, Unit
BNE Form_Dec_Noll_zpt

```

Form_Dec_zpt_No11

```

MOV      R3, #0x30          ; '0'
STRB     R3, [R1,#2]
STRB     R3, [R1,#3]
TST      R5, #Pr00
STRBEQ   No11, [R1,#3]
STRB     No11, [R1,#4]

```

Form_SRAM_D

```

MOV32    R0, #SRAM_Form_Dec ; Память форматирования десятичного числа
MOV32    R1, #SRAM_D         ; Память кода, выводимого на индикатор
BIC      R5, R5, #(zpt2:OR:zpt3:OR:Pr00)
ADD      R1, R5

```

Form_SRAM_D_c

```

LDRB     R3, [R0],#1
CMP      R3, #0
BEQ      Form_SRAM_exit

STRB     R3, [R1],#-1
B        Form_SRAM_D_c

```

Form_SRAM_exit

```

MOV32    R0, #SRAM_D
BL      Indic_W_cdseg

POP      {R0-R7,R12}
POP      {LR}
BX       LR

ENDP

```

```

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
        AREA CONST, DATA, READONLY, align=2
;*****
;Массив форматов чисел. 0x30...0x39 - цифры 0...9.
Kn_Out_0
    DCB     0x34,0x33,0x32,0x31,0x30,0x35,0x36,0x37,0x38,0x39,0x00

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
                align
; Массив форматов Функций и Режимов. 0x10...0x13 - коды Функций, 0x20...0x27 - коды Режимов.
Kn_Out_1
    DCB     0x25,0x26,0x27,0x28,0x24,0x23,0x21,0x22,0x12,0x00,0x00
;      Коды управления записью в индикатор
;.....

;      0x00 - конец записи текста
;.....

;      ВНИМАНИЕ! Коды DD,EC не должны совпадать с кодами, применяемых адресов и данных.
DD     EQU 0x7d      ;запись Данных
EC     EQU 0x7c      ;запись Команды, Адреса
;.....

; Сдвиг = 3 + Количество символов числа (тах - учитывать пробел и запятую. Первый символ нулевой)
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
                align
Indic_Tab_Init      ;Инициализация индикатора, адрес: 0 (0x80)
;Набор функций, Дисплей ВКЛ/ВЫКЛ управления, Очистить дисплей
    DCB EC,0x30,0x30,0x30,0x38,0x0e,0x06,0x01,0x00;

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
                align
Tab_Transcoding      ; Таблица перекодировки русских символов.

;      0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0a,0x0b,0x0c,0x0d,0x0e,0x0f
    DCB 0x80,0x81,0x82,0x83,0x84,0x85,0x86,0x87,0x88,0x89,0x8a,0x8b,0x8c,0x8d,0x8e,0x8f ;0x80
    DCB 0x90,0x91,0x92,0x93,0x94,0x95,0x96,0x97,0x98,0x99,0x9a,0x9b,0x9c,0x9d,0x9e,0x9f ;0x90
    DCB 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xa2,0x00,0x00,0x00,0x00,0x00,0x00,0x00 ;0xa0
    DCB 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xb5,0x00,0x00,0x00,0x00,0x00,0x00,0x00 ;0xb0
    DCB 0x41,0xa0,0x42,0xa1,0xe0,0x45,0xa3,0xa4,0xa5,0xa6,0x4b,0xa7,0x4d,0x48,0x4f,0xa8 ;0xc0
    DCB 0x50,0x43,0x54,0xa9,0xaa,0x58,0xe1,0xab,0xac,0xe2,0xad,0xae,0x62,0xaf,0xb0,0xb1 ;0xd0
    DCB 0x61,0xb2,0xb3,0xb4,0xe3,0x65,0xb6,0xb7,0xb8,0xb9,0xba,0xbb,0xbc,0xbd,0xbf,0xbe ;0xe0
    DCB 0x70,0x63,0xbf,0x79,0xe4,0x78,0xe5,0xc0,0xc1,0xe6,0xc2,0xc3,0xc4,0xc5,0xc6,0xc7 ;0xf0

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
                align
        GBLS str1      ; установить значение глобальной строковой переменной
        GBLS str2

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
                align
Tab_Rej_tab          ;Работа
        DCD Tab Rej Alektr      ;Режим Выбора Электродов

```

```

DCD    Tab_Rej_Svarka_IU    ;Работа Режим Сварки (Стабилизации Тока , Напряжения)
DCD    Tab_Rej_Forsaj_T     ;Режим Настройки параметров
DCD    Tab_Rej_Forsaj_U     ;Режим Настройки параметров
DCD    Tab_Rej_Zajigan_I    ;Режим Настройки параметров
DCD    Tab_Rej_Zajigan_U    ;Режим Настройки параметров
DCD    Tab_Rej_Gash_Dugi    ;Режим Настройки параметров: Гашения дуги (0x06)
DCD    Tab_Rej_Gradus       ;Режим Перегрев

```

```

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

```

```

                                align
Tab_Rej_Alektr                 ;Режим Выбора Электродов (0x00)
                                0
DCD                            0
DCD                            0
DCD                            0
DCD                            0
DCD                            0
DCD                            0
DCD                            0
DCD                            0
DCD    Rej_Alektr              ; 0x40000000
DCD    L1_Alektr_User          ; 0x48000000 Пользователь
DCD    L1_Alektr_40            ; 0x50000000 d=4,0 мм
DCD    L1_Alektr_30            ; 0x58000000 d=3,0 мм
DCD    L1_Alektr_25            ; 0x60000000 d=2,5 мм
DCD    L0_Alektr_20            ; 0x68000000 d=2,0 мм
DCD    L0_Alektr_16            ; 0x70000000 d=1,6 мм

```

```

Rej_Alektr
str1  SETS "Электр.:1,6; 2,0"
      DCB EC,0x80,DD,str1
str1  SETS "2,5; 3,0; 4,0; П"
      DCB EC,0xc0,DD,str1,0x00

```

```

L0_Alektr_16    DCB EC,0x89,0x0f,0x00    ;d=1,6 мм
L0_Alektr_20    DCB EC,0x8e,0x0f,0x00    ;d=2,0 мм
L1_Alektr_25    DCB EC,0xc1,0x0f,0x00    ;d=2,5 мм
L1_Alektr_30    DCB EC,0xc6,0x0f,0x00    ;d=3,0 мм
L1_Alektr_40    DCB EC,0xcb,0x0f,0x00    ;d=4,0 мм
L1_Alektr_User  DCB EC,0xcf,0x0f,0x00    ;Пользователь

```

```

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

```

```

                                align
Tab_Rej_Svarka_IU             ;Работа Режим Сварки Стабилизации Тока (0x01)

                                0;
DCD    Lin0_Izad               ; 0x08000000
DCD    Lin0>Ifak               ; 0x10000000
DCD    Lin1_Uzad               ; 0x18000000
DCD    Lin1_Ufak               ; 0x20000000
DCD    0
DCD    0
DCD    0
DCD    Rej_IU                  ; 0x40000000
DCD    L0_Mark                  ; 0x48000000
DCD    L1_Mark                  ; 0x50000000

```

```

Rej_IU
str1  SETS "Iз/φ:      /      A"
      DCB EC,0x80,DD,str1
str1  SETS "Uз/φ:      /      B"
      DCB EC,0xc0,DD,str1,0x00

```

```

Lin0_Izad      DCB EC,0x85,DD,0x20,0x20,0x20,0x00,5 ; Cðbuz: 5
Lin0>Ifak      DCB EC,0x8b,DD,0x20,0x20,0x20,0x00,5 ; Cðbuz: 5
Lin1_Uzad      DCB EC,0xc5,DD,0x20,0x20,0x20,0x00,5 ; Cðbuz: 5
Lin1_Ufak      DCB EC,0xcb,DD,0x20,0x20,0x20,0x00,5 ; Cðbuz: 5

L0_Mark        DCB EC,0x87,0x0f,0x00
L1_Mark        DCB EC,0xc7,0x0f,0x00

```

;

```

align
Tab_Rej_Forsaj_T      ;Режим Настройки параметров: Залипание электрода (0x02, 0x03)
Tab_Rej_Forsaj_U
DCD                    0
DCD                    Lin0_Timp      ; 0x08000000
DCD                    Lin1_Umin      ; 0x10000000
DCD                    0
DCD                    0
DCD                    0
DCD                    0
DCD                    0
DCD                    Rej_UT         ; 0x40000000
DCD                    L0_Mark        ; 0x48000000
DCD                    L1_Mark        ; 0x50000000

```

```
Rej_UT
str1  SETS "    Тимп:      C  "
      DCB EC,0x80,DD,str1
str1  SETS "    Умин:    0,0 В  "
      DCB EC,0xc0,DD,str1,0x00
```

```

    align
Lin0_Timp
    DCB EC,0x87,DD,0x20,0x20,0x20,0x20,0x00,zpt2+7    ; Cdbuz: 7
Lin1_Umin
    DCB EC,0xc8,DD,0x20,0x20,0x20,0x00,zpt1+6          ; Cdbuz: 6

```

;

```

align
Tab_Rej_Zajigan_I      ;Режим Настройки параметров: Зажигание (0x04, 0x05)
Tab_Rej_Zajigan_U
DCD 0
DCD Lin0_Imax          ; 0x08000000
DCD 0
DCD 0
DCD 0
DCD 0
DCD 0
DCD 0
DCD Rej_IU_Zaj         ; 0x40000000
DCD L0_Mark            ; 0x48000000
DCD L1_Mark            ; 0x50000000
DCD Lin1_Uzaj_2        ; 0x58000000
DCD Lin1_Uzaj_1        ; 0x60000000

```

[illegible]

```
str1  SETS  "  Haгpeв: MAX  "  
      DCB  EC,0x80,DD,str1  
str1  SETS  "  "  
      DCB  EC,0xc0,DD,str1,0x00  
;*****  
  
      END
```