

Прибор Генератор – Измеритель

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.

Вот в качестве конкретного примера я решил со «скирдовать» прибор из имеющихся у меня заделов. Я не старался сделать прибор, по конструкции и параметрам оптимальным. Но он вполне пригоден для работ и достаточно точный.

Данный прибор разработан на основе STM32F103C8 и STM32F405RGB, и программное обеспечение (ПО) на платформах KEIL и IAR.

В данной статье приведены схемные решения и ПО прибора, хотя кой в чем они и ограничены, но этого достаточно для повторения.

ПО я стараюсь писать на Ассемблере, так как он ближе к «железу» и лучше понимаешь работу микроконтроллера. Программы получаются более, оптимальными и быстродействующими.

Теперь несколько советов.

Перед тем, как что-то разрабатывать составьте для себя, четко, что вы хотите. Не поленитесь записать все это на бумаге, в виде «Технического задания» (ТЗ) для себя.

А если вы получаете заказ со стороны, тем более будьте внимательны к тому, что предлагают вам, в качестве задания. Часто и густо Заказчик сам толком не знает, что ему нужно. Вот здесь, опять не ленитесь и вникните в ТЗ, и в нормативные документы, если в ТЗ будут на них ссылки.

Обратите внимание на требования в ТЗ о сдаче изделия Заказчику. Особенно это касается ПО там всегда есть к чему придраться. Постарайтесь сделать так, что бы последняя редакция ТЗ была за вами. Вот тогда ставьте подписи.

В общем, взятие, разработчиком разработки ТЗ на себя: много положительного. В процессе задавания «глупых» вопросов Заказчику вы лучше вникните в суть проблемы, и параллельно в вашей голове может сформироваться структура решений по разработке, что сократит вам время в дальнейшей работе.

В процессе разработки изделия, у Заказчика может появиться «аппетит» и он начнет добавлять в ТЗ новые требования. Никогда на это не соглашайтесь, даже если вам кажется, что это возможно. В процессе разработки, в лучшем случае, это увеличения объема работ, и в худшем грозит серьезной переделкой – в зависимости от стадии разработки.

Когда сдадите проект Заказчику, вот тогда пусть он открывает новый проект на доработку. Вот тогда Заказчик будет лучше понимать, что ему нужно, и вы будете лучше разбираться в проблеме.

Никогда не ленитесь рисовать структуру системы разрабатываемого изделия. Правда, в процессе работы квадратики могут, как делиться, так и сливаться.

Запомните, ленивый делает работу дважды, а то и больше, если ленивый не опытный.

Описание, параметры и схемы прибора

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

Описание



Рис. 1.

Общий вид прибора показан на Рис. 1.

В качестве индикатора прибора используется 20-символьный, 4-строчный дисплей.

При включении прибора на индикаторе высвечивается первый кадр Help. Вращая энкодер: «Строка 1» меняем кадры Help и описание режимов.

По каждому параметру, включенного режима, в строке индикатора, идет управление через энкодеры: «Строка 1», «Строка 2», «Строка 3», «Строка 4».

Вращая энкодер, меняем параметр или значения числа параметра, в указанном маркером порядке числа. Изменение положения маркера порядка числа меняется нажатием кнопки энкодера.

Ограничение параметров вложено в ПО.

Символы «<» и «>» показывают увеличение или уменьшение значения числа, или направление сдвига маркера.

Выбор режимов производится нажатием кнопок «Режимы»

Кнопка «ВКЛ/ОТКЛ» предназначена для управления выходами генераторов.

Параметры прибора

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

Режим FTt – измерение



Рис. 2

Вход 1 (FTt) : Счет

Измерение частоты Fпер: 1...30 000 000 Гц;
 Измерение периода Tпер: 0,1...9 999 999,9 мкс;
 Измерение импульса тимп: 0,1...9 999 999,9 мкс;
 Входное напряжение, макс.: ± 30 В;
 Напряжение смещение входа: -2...+2 В;
 Напряжение гистерезиса входа: 0,2 В;
 Сопротивление входа: 280 кОм.

Режим NTn – измерение

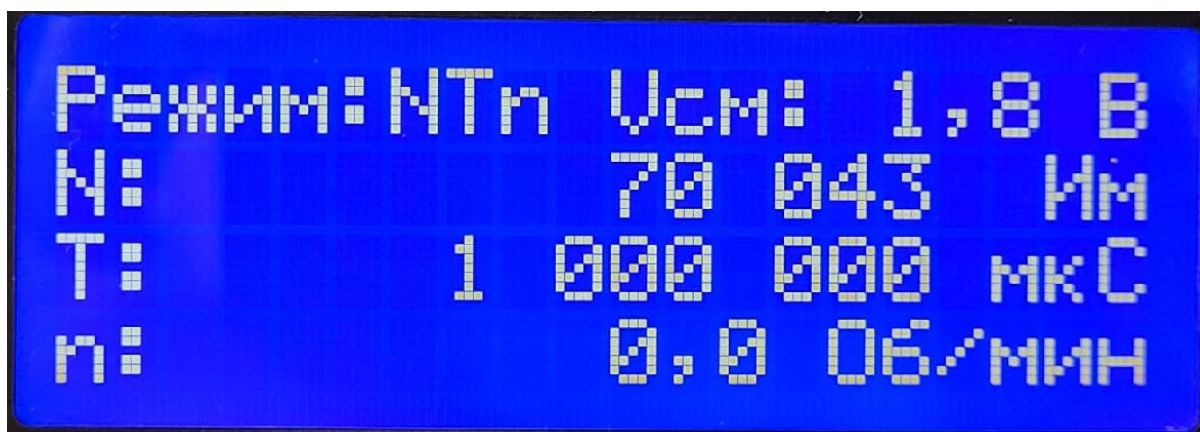


Рис. 3.

Вход 1 (N): Счет

Измерение количества импульсов N: 1...99 999 999 Имп;
 Входное напряжение, макс.: ± 30 В;
 Напряжение смещение входа: -2...+2 В;
 Напряжение гистерезиса входа: 0,2 В;
 Сопротивление входа: 280 кОм.

Вход 1 (Т): Запуск

Измерение времени Т: 1...99 999 999 мкс;
 Входное напряжение, макс.: ± 30 В;
 Напряжение смещение входа: -2...+2 В;
 Напряжение гистерезиса входа: 0,2 В;
 Сопротивление входа: 280 кОм.

Вход 2 (Т): Останов

Измерение времени Т: 1...99 999 999 мкс;
 Входное напряжение, макс.: ± 30 В;
 Напряжение смещение входа: 2,5 В;
 Напряжение гистерезиса входа: 1,2 В;
 Сопротивление входа: 58 кОм.

Вх (n): Счет от датчика

Измерение оборотов n: 6...60 000 Об/мин;
 Входное напряжение: 0...5 В;
 Напряжение смещение входа: 2,5 В;
 Напряжение гистерезиса входа: 1,2 В;
 Сопротивление входа: 5 кОм.

Режим Гsin – синтез синусоидального сигнала

Рис. 4.

Выход (Гsin)

Частота генерации Fзад: 1...2 280 000 Гц;
 Амплитуда выходного напряжения: 0,1...5 В;
 Напряжение смещение выхода: -5...+5 В;
 Дрожание фазы (при 35...2 280 000 Гц) ... ± 25 нс
 Сопротивление выхода: 50 Ом.
 Варианты осциллограмм показаны в приложении.

Режим Гфунк – синтез сигнала заданной функций



Рис. 5.

Выход (Гфунк)

Период генерации Тпер, мин: зависит то функции;

Фронты генерируемого сигнала, макс.: 25 нс;

Амплитуда выходного напряжения: 0,1...5 В;

Напряжение смещение выхода: -5...+5 В;

Сопротивление выхода: 50 Ом;

Количество функций сигнала: определяется ПО.

Варианты осциллограм показаны в приложении.

Режим ШИМ и ЧИМ

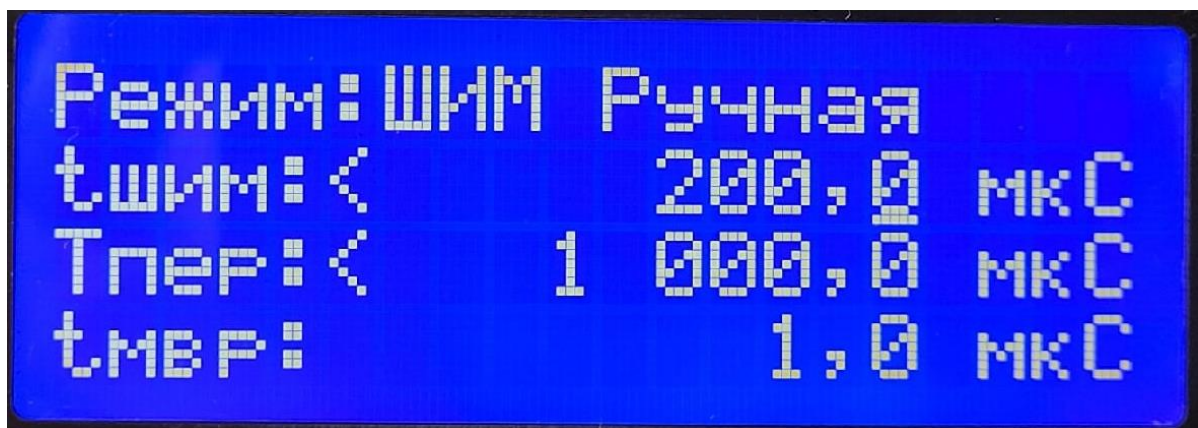


Рис. 6.

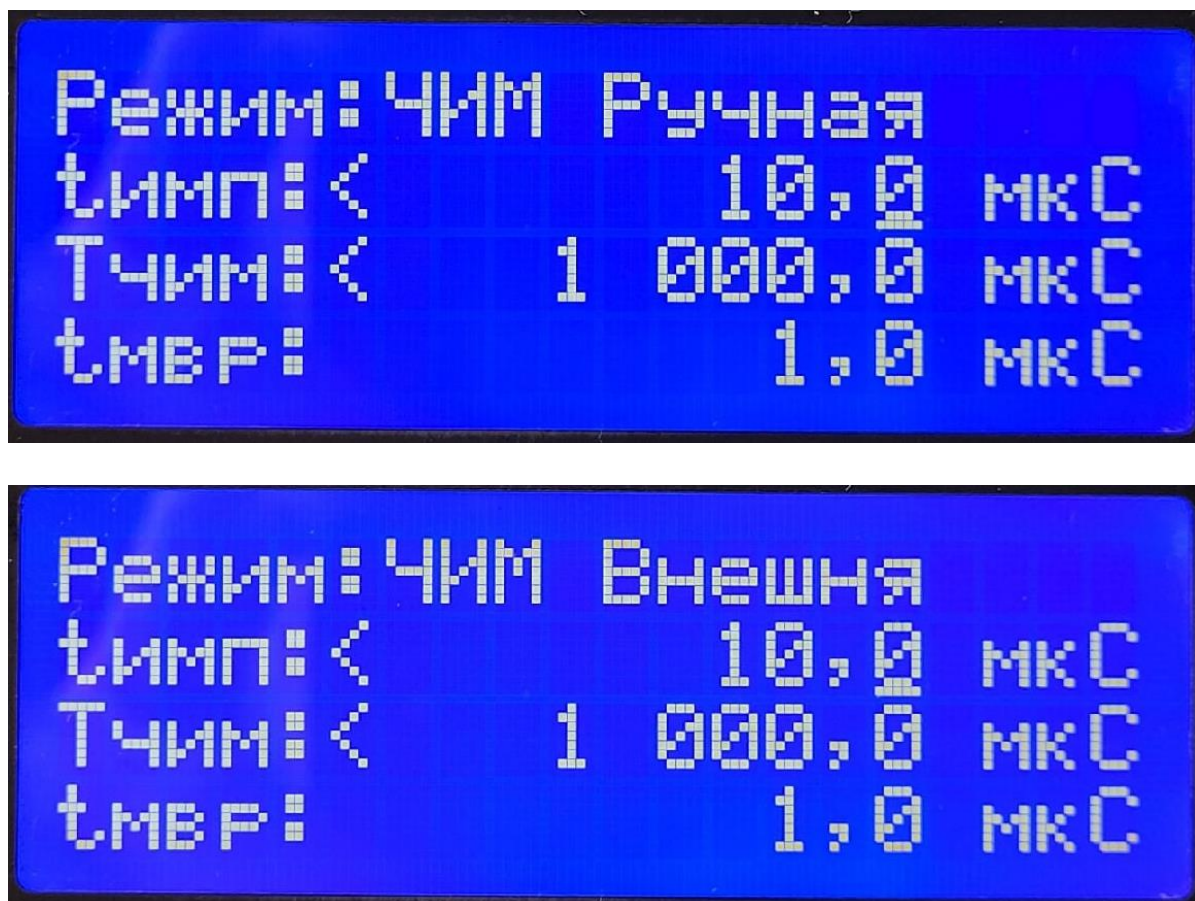


Рис. 7.

Выход (Вых1, Вых2, Вых12)

Выходное напряжение, макс.: 5 В;
 Сопротивление выхода: 50 Ом;
 Разрешение времени ШИМ и ЧИМ 25 нС.

Вход (Uоткл)

Напряжение отключения: 4,2...5 В;
 Сопротивление входа: 6,8 кОм.

Вход (Uупр)

Напряжение управления: 0...5 В;
 Сопротивление входа: 10 кОм.

Модуляция ШИМ и ЧИМ «Ручная», путем вращения энкодеров «Строка 2», «Строка 3», соответственно или «Внешняя» путем подачи напряжения на вход «Uупр».

Вид модуляции выбирается путем вращения энкодера «Строка 1»

Защитное отключение ШИМ и ЧИМ производится путем подачи напряжения на вход «Uоткл».

Схемы прибора

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

Здесь показаны не все схемы, только определяющие.

На Рис. 8 показана схема контроллера связи USART, индикатора и клавиатуры.

Контроллер разработан на базе китайской платы (STM32F103C8)

Схема шифровки кнопок - аналоговая и показана на Рис. 9.

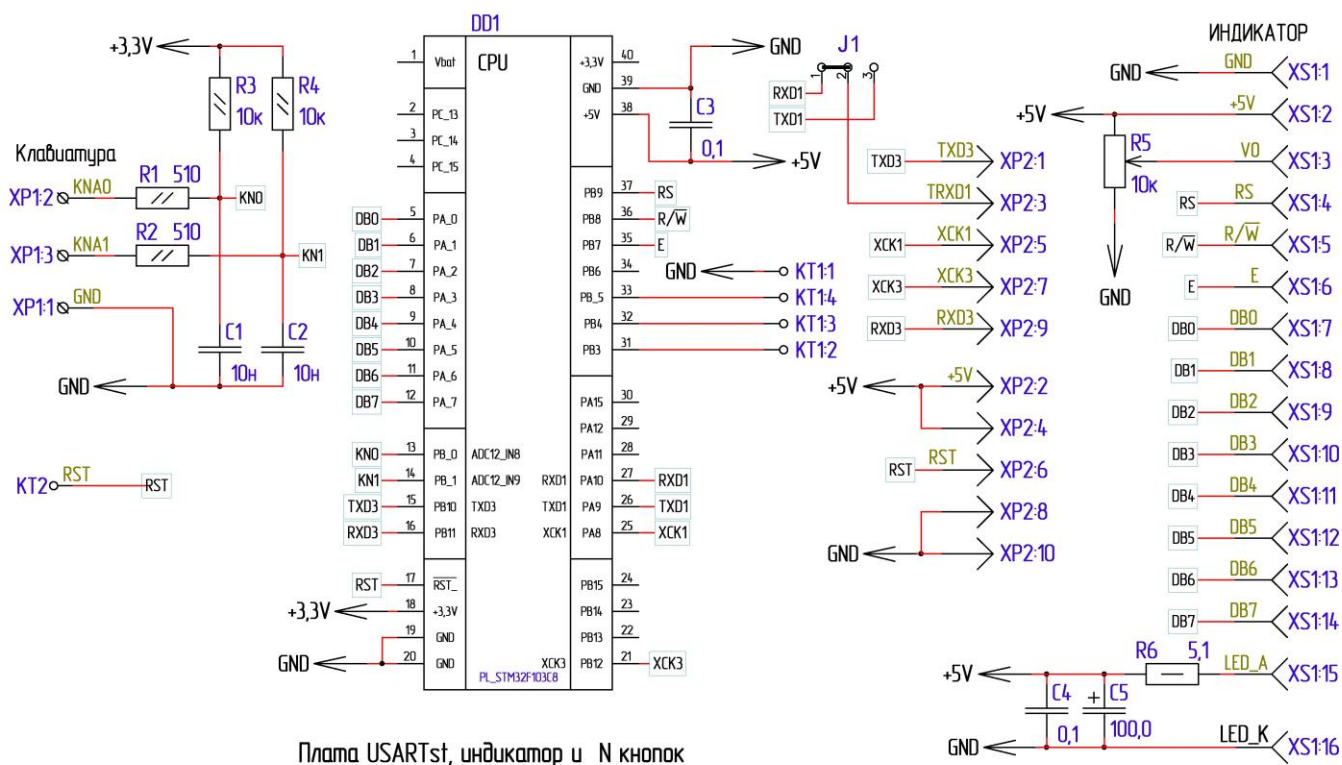


Рис.8

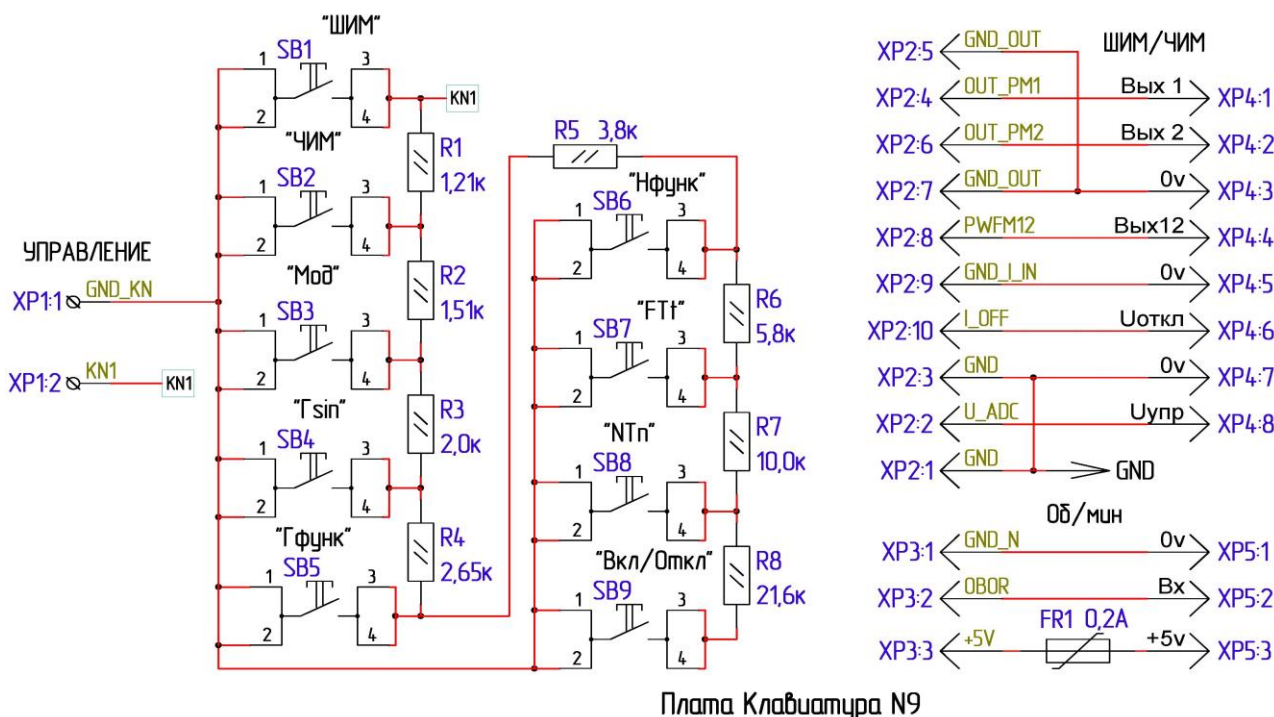


Рис.9

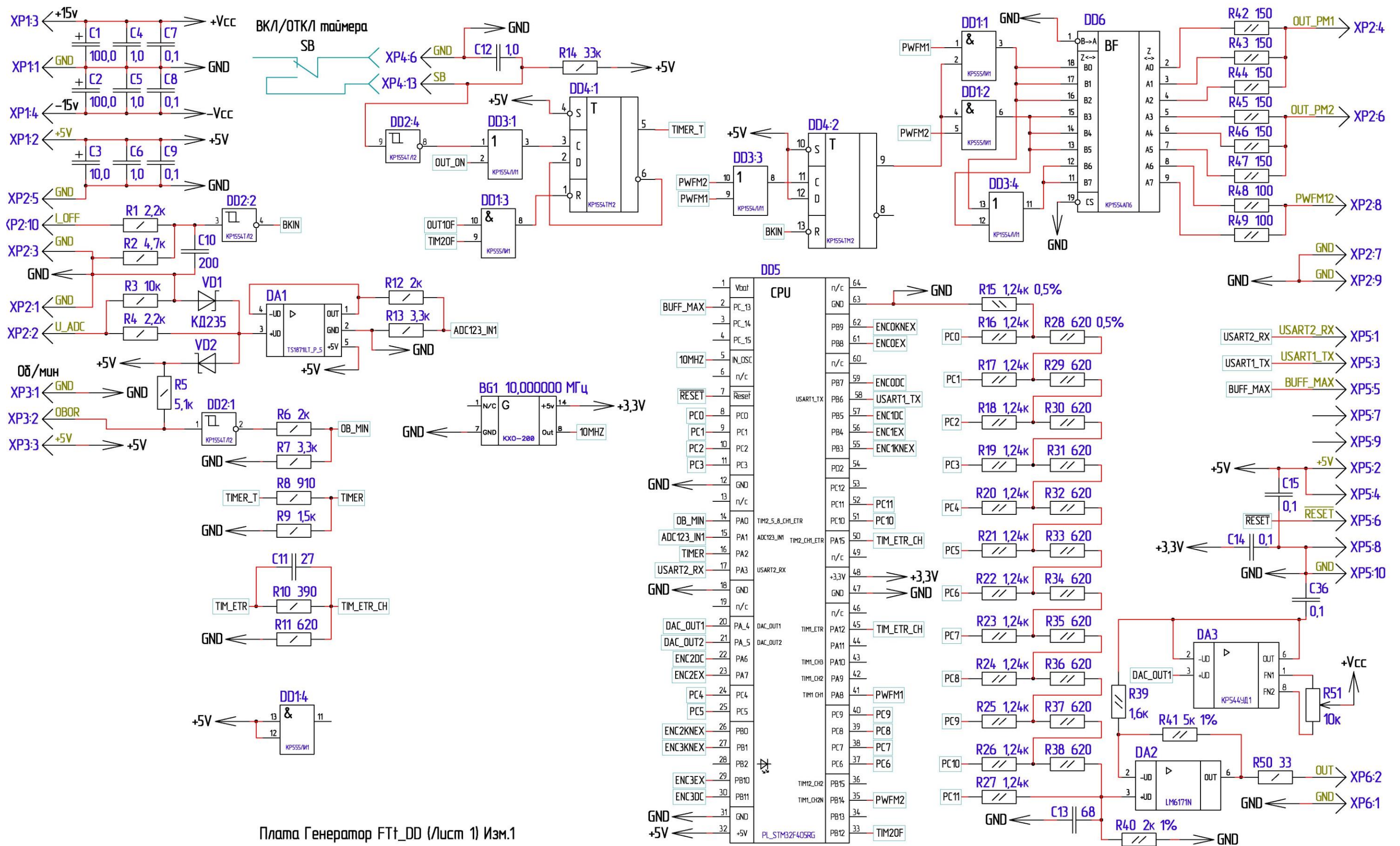
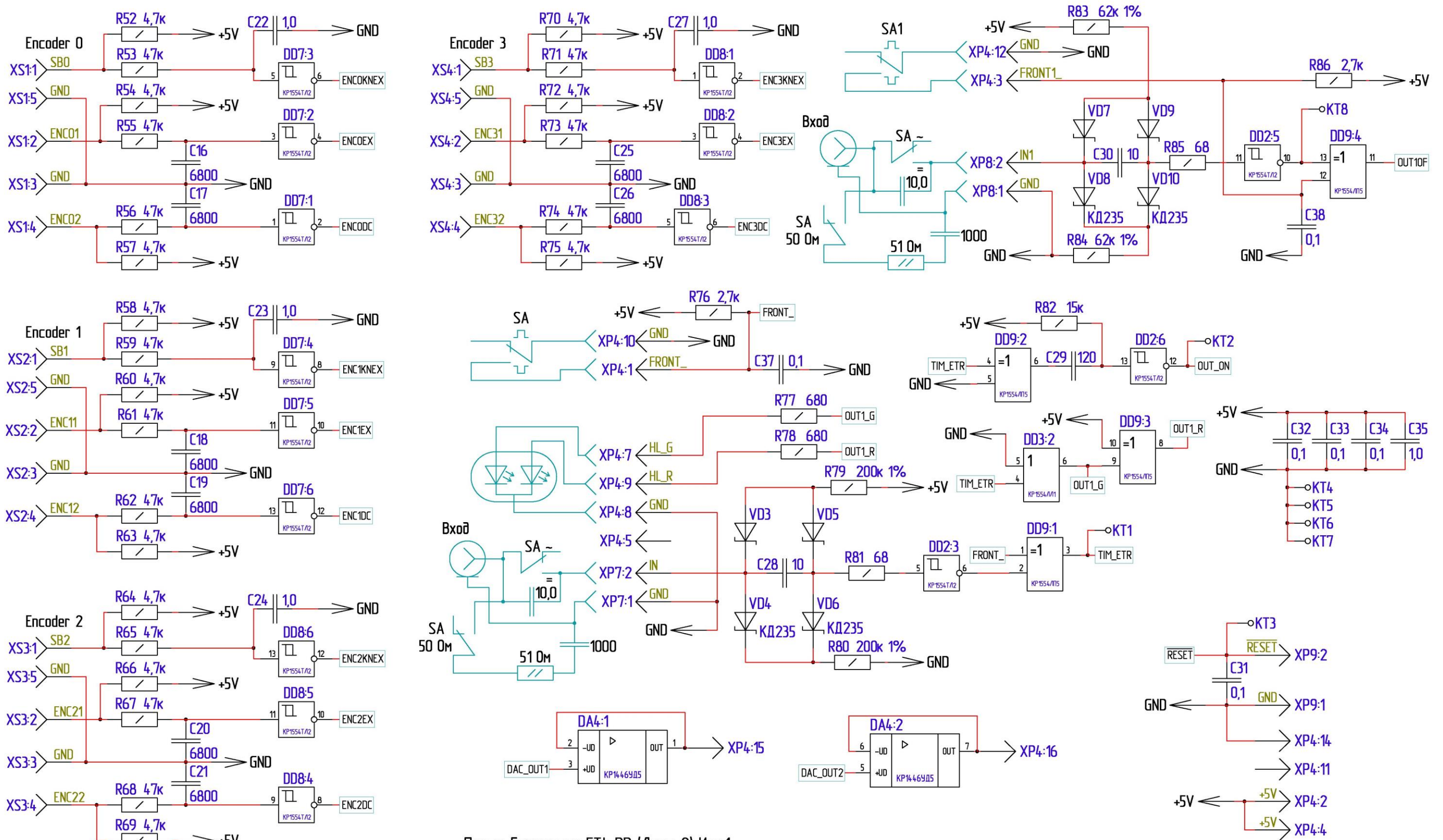


Рис.10



Плата Генератор FTt_DD (Лист 2) Изм.1

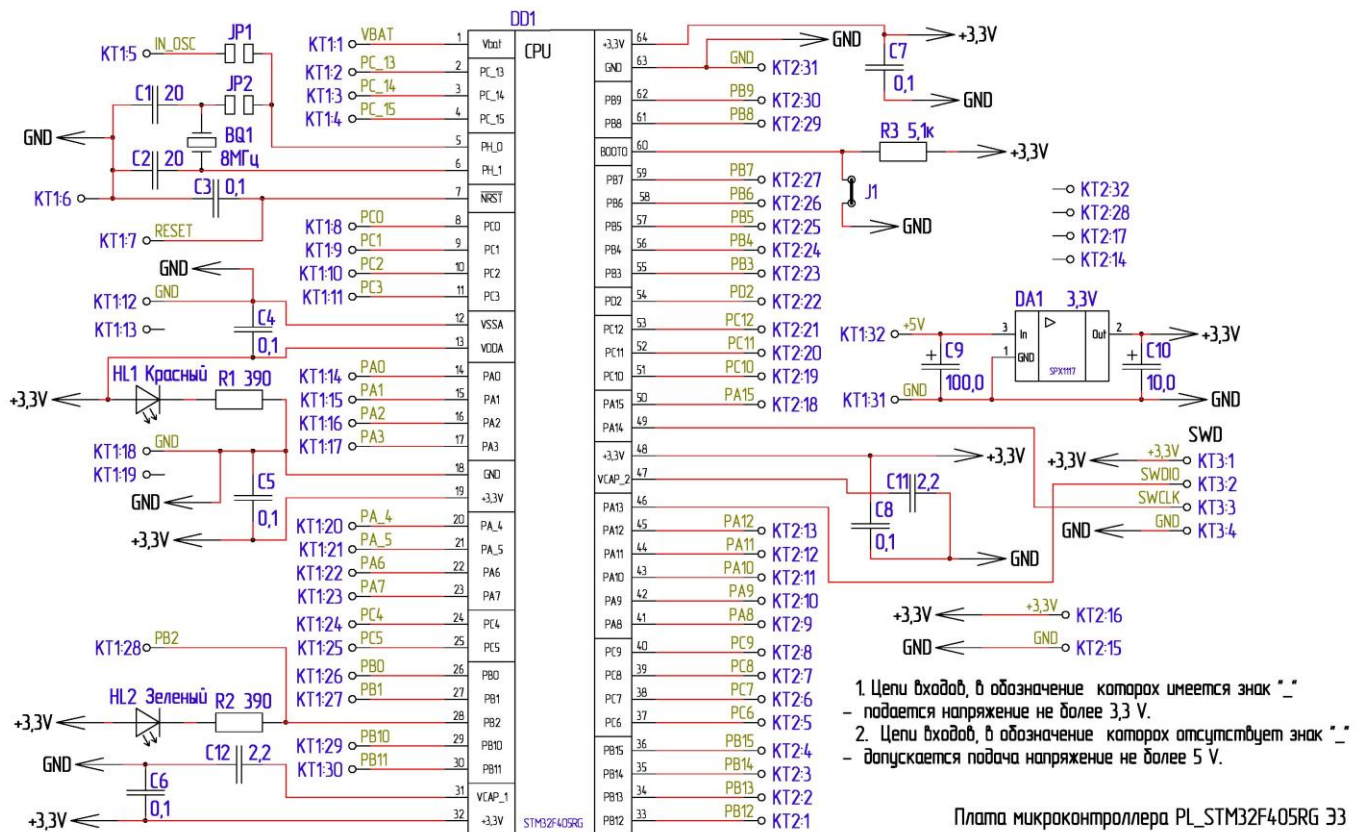


Рис.12

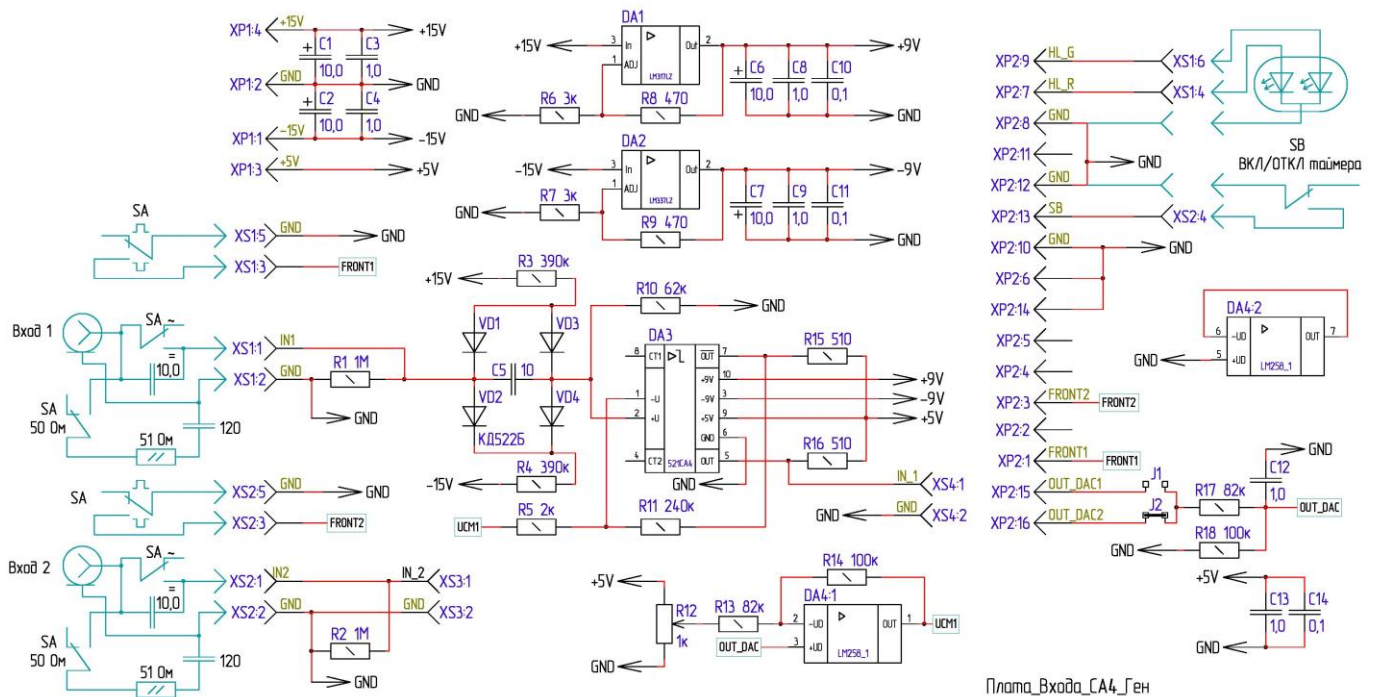


Рис.13

Схема основной платы показана на Рис. 10 и Рис. 11 и разработана с использованием экспериментальной платы PL_STM32F405RGB, схема которой показана на Рис. 12. Плата PL_STM32F405RGB оригинальная и разработана автором.

Для увеличения частотного диапазона вывода аналогового сигнала, использован резистивный R–2R ЦАП (R15...R40).

Максимальный входной частотный диапазон определяется компаратором (DA3) на плате входа. Схема платы входа показана на Рис. 13

Программное обеспечение (ПО) прибора.

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

ПО Плата USARTst, индикатор и N кнопок

```
;NAME DV204Lsd_103C8_Keil

AREA DV204Lsd_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_clk_B    EQU    4        ;Rsos[4...7] - Счетчик принятых байт
Rsos_clk    EQU    0x000000f0 ;Rsos[4...7] - Счетчик принятых байт
Rsos_ADC_Kn    EQU    0x00000100 ; Флаг Rsos: 1 - Кнопка отработана
Rsos_Indic_DD    EQU    0x00000200 ; Флаг Rsos: 1 - Команда: Запись данных в индикатор, 0
- Команда: Запись команд в индикатор.
Rsos_Kod_2x    EQU    0x00000400 ; Флаг Rsos: 1 - Прошел код кнопки 0x2x
Rsos_Kn_On_Off    EQU    0x00000800 ; Флаг Rsos: 1 - Прошел код кнопки Kn_On_Off
Rsos_Buff_Over    EQU    0x00001000 ; Флаг Rsos: 1 - Переполненное Буфера
Rsos_Long_Comm    EQU    0x00002000 ; Флаг Rsos: 1 - Запись длинной команды в индикатор
(1,53 мс)
;.....

;Признаки
Pr00    EQU    0x40        ;Признак 00,0 текста
zpt    EQU    0x80        ;Признак запятой текста
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
;.....

;Привзка к схеме
E    EQU    0x00000080 ; Порт В Установка BS11 - Запись в индикатор: "1"
R_W    EQU    0x00000100 ; Порт В Установка BS12 - выбор Чтение: "1", Запись: "0"
RS    EQU    0x00000200 ; Порт В Установка BS15 - Выбор данных: "1", Команда: "0"
;.....

;Операивная 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_Buff EQU 0x20000048 ; 4 байта

```

```

SRAM_Kn_0 EQU 0x20000050
SRAM_Kn_1 EQU 0x20000054
SRAM_Kn_2 EQU 0x20000058
SRAM_Kn_3 EQU 0x2000005c

```

```

SRAM_D EQU 0x20000060 ; Информация выводимая на индикатор ( Макс. 32 байт)
SRAM_Form_Dec EQU 0x20000080 ; 16 байт
SRAM_Func_Dec EQU 0x20000090 ; Адрес (младший разряд) области памяти 8-разрядного
десятичного числа ( Макс. 16 байт)

```

```

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

```

```

; Форвардная декларация разделов.
AREA RESET, DATA, READONLY, align = 2
THUMB

```

```

__vector_table

```

```

;size  acronym  IRQN  Описание смещения
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, PE0, PF2, PG2) | | | | | |
| DCD | EXTI3_IRQHandler | ; 9 | 0x00000064 | - прерывание от линии 3 модуля EXTI | |
| (PA3, PB3, PC3, PD3, PE3, PF3, PG3) | | | | | |
| DCD | EXTI4_IRQHandler | ; 10 | 0x00000068 | - прерывание от линии 4 модуля EXTI | |
| (PA4, PB4, PC4, PD4, PE4, PF4, PG4) | | | | | |
| DCD | DMA1_Channel1_IRQHandler | ; 11 | 0x0000006C | - любое немаскированное прерывание от канала 1 модуля DMA1 | |
| DCD | DMA1_Channel2_IRQHandler | ; 12 | 0x00000070 | - любое немаскированное прерывание от канала 2 модуля DMA1 | |
| DCD | DMA1_Channel3_IRQHandler | ; 13 | 0x00000074 | - любое немаскированное прерывание от канала 3 модуля DMA1 | |
| DCD | DMA1_Channel4_IRQHandler | ; 14 | 0x00000078 | - любое немаскированное прерывание от канала 4 модуля DMA1 | |
| DCD | DMA1_Channel5_IRQHandler | ; 15 | 0x0000007C | - любое немаскированное прерывание от канала 5 модуля DMA1 | |
| DCD | DMA1_Channel6_IRQHandler | ; 16 | 0x00000080 | - любое немаскированное прерывание от канала 6 модуля DMA1 | |
| DCD | DMA1_Channel7_IRQHandler | ; 17 | 0x00000084 | - любое немаскированное прерывание от канала 7 модуля DMA1 | |
| DCD | ADC1_2_IRQHandler | ; 18 | 0x00000088 | - любое немаскированное прерывание от модулей ADC1, ADC2 | |
| DCD | USB_HP_CAN_TX_IRQHandler | ; 19 | 0x0000008C | - высокоприоритетные прерывания от USB или от передатчика модуля CAN | |
| DCD | USB_HP_CAN_RX0_IRQHandler | ; 20 | 0x00000090 | - низкоприоритетные прерывания от USB или от приёмного буфера FIFO_0 модуля CAN | |
| DCD | CAN_RX1_IRQHandler | ; 21 | 0x00000094 | - прерывание от приёмного буфера FIFO_1 модуля CAN | |
| DCD | CAN_SCE_IRQHandler | ; 22 | 0x00000098 | - прерывание от события CAN Status Change Error | |
| DCD | EXTI9_5_IRQHandler | ; 23 | 0x0000009C | - прерывание от линий [9:5] модуля EXTI | |
| DCD | TIM1_BRK_IRQHandler | ; 24 | 0x000000A0 | - прерывание по событию Break модуля TIM1 | |
| DCD | TIM1_UP_IRQHandler | ; 25 | 0x000000A4 | - прерывание по событию Update модуля TIM1 | |
| DCD | TIM1_TRG_COM_IRQHandler | ; 26 | 0x000000A8 | - прерывание от триггера 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 триггера 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_Channel12_IRQHandler | ; 57 | 0x00000124 | - прерывание от канала Channel2 модуля DMA2 |
| DCD | DMA2_Channel13_IRQHandler | ; 58 | 0x00000128 | - прерывание от канала Channel3 модуля DMA2 |
| DCD | DMA2_Channel14_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
BKPT #1
```

MemManageFault_IRQHandler ; - отказ системы управления памятью
ВКРП #1

```
BusFault_IRQHandler      ; - отказ шины
BKPT #1
```

```
UsageFault_IRQHandler      ; - отказ программы
BKPT #1
```

```
SVCall_IRQHandler      ; - вызов супервизора
BKPT #1
```

Debug_Monitor_IRQHandler ; - исключение монитора отладки
BKPT #1

```
PendSV_IRQHandler      ; - запрос системной службы
BKPT #1
```

SysTick_IRQHandler ; - системный таймер
BKPT #1

WWDG_IRQHandler ; - оконный сторожевой таймер
BKPT #1

PVD_IRQHandler ; - программируемый детектор напряжения (линия 16 модуля EXTI)
BKPT #1

```
TAMPER_IRQHandler      ; - прерывание от события TAMPER
BKPT #1
```

```
RTC_IRQHandler           ; - глобальное прерывание от часов реального времени
BKPT #1
```


| | |
|---------------------------|---|
| FLASH_IRQHandler | <i>;- любое немаскированное прерывание от модуля FSMC</i> |
| BKPT #1 | |
| RCC_IRQHandler | <i>;- любое немаскированное прерывание от модуля RCC</i> |
| BKPT #1 | |
| EXTI0_IRQHandler | <i>;- прерывание от линии 0 модуля EXTI (PA0, PB0, PC0, PD0, PE0, PF0, PG0)</i> |
| BKPT #1 | |
| EXTI1_IRQHandler | <i>;- прерывание от линии 1 модуля EXTI (PA1, PB1, PC1, PD1, PE1, PF1, PG1)</i> |
| BKPT #1 | |
| EXTI2_IRQHandler | <i>;- прерывание от линии 2 модуля EXTI (PA2, PB2, PC2, PD2, PE0, PF2, PG2)</i> |
| BKPT #1 | |
| EXTI3_IRQHandler | <i>;- прерывание от линии 3 модуля EXTI (PA3, PB3, PC3, PD3, PE3, PF3, PG3)</i> |
| BKPT #1 | |
| EXTI4_IRQHandler | <i>;- прерывание от линии 4 модуля EXTI (PA4, PB4, PC4, PD4, PE4, PF4, PG4)</i> |
| BKPT #1 | |
| DMA1_Channel1_IRQHandler | <i>;- любое немаскированное прерывание от канала 1 модуля DMA1</i> |
| BKPT #1 | |
| DMA1_Channel2_IRQHandler | <i>;- любое немаскированное прерывание от канала 2 модуля DMA1</i> |
| BKPT #1 | |
| DMA1_Channel3_IRQHandler | <i>;- любое немаскированное прерывание от канала 3 модуля DMA1</i> |
| BKPT #1 | |
| DMA1_Channel4_IRQHandler | <i>;- любое немаскированное прерывание от канала 4 модуля DMA1</i> |
| BKPT #1 | |
| DMA1_Channel5_IRQHandler | <i>;- любое немаскированное прерывание от канала 5 модуля DMA1</i> |
| BKPT #1 | |
| DMA1_Channel6_IRQHandler | <i>;- любое немаскированное прерывание от канала 6 модуля DMA1</i> |
| BKPT #1 | |
| DMA1_Channel7_IRQHandler | <i>;- любое немаскированное прерывание от канала 7 модуля DMA1</i> |
| BKPT #1 | |
| ADC1_2_IRQHandler | <i>;- любое немаскированное прерывание от модулей ADC1, ADC2</i> |
| BKPT #1 | |
| USB_HP_CAN_TX_IRQHandler | <i>;- высокоприоритетные прерывания от USB или от передатчика модуля CAN</i> |
| BKPT #1 | |
| USB_HP_CAN_RX0_IRQHandler | <i>;- низкоприоритетные прерывания от USB или от приёмного буфера FIFO_0 модуля CAN</i> |
| BKPT #1 | |
| CAN_RX1_IRQHandler | <i>;- прерывание от приёмного буфера FIFO_1 модуля CAN</i> |
| BKPT #1 | |
| CAN_SCE_IRQHandler | <i>;- прерывание от события CAN Status Change Error</i> |
| BKPT #1 | |
| EXTI9_5_IRQHandler | <i>;- прерывание от линий [9:5] модуля EXTI</i> |
| BKPT #1 | |
| TIM1_BRK_IRQHandler | <i>;- прерывание по событию Break модуля TIM1</i> |
| BKPT #1 | |
| TIM1_UP_IRQHandler | <i>;- прерывание по событию Update модуля TIM1</i> |
| BKPT #1 | |
| TIM1_TRG_COM_IRQHandler | <i>;- прерывание от триггера или события COM модуля TIM1</i> |
| BKPT #1 | |
| TIM1_CC_IRQHandler | <i>;- прерывание от события захват/сравнение модуля TIM1</i> |
| BKPT #1 | |
| TIM2_IRQHandler | <i>;- любое прерывание от модуля TIM2</i> |
| BKPT #1 | |
| TIM3_IRQHandler | <i>;- любое прерывание от модуля TIM3</i> |
| BKPT #1 | |
| TIM4_IRQHandler | <i>;- любое прерывание от модуля TIM4</i> |
| BKPT #1 | |
| I2C1_EV_IRQHandler | <i>;- прерывание от события event модуля I2C1</i> |
| 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
    B Inter_USART1
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
RTCArm_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 триггера 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    Noll, #0
MOV    Unit, #1

;Инициализация тактирования
MOV32  R0, #RCC
; Затактировать POWER control
LDR    R1, [R0,#RCC_APB1ENR]
ORR    R1, R1, #RCC_APB1ENR_PWREN ; бит PWREN (номер бита 28) - включить
STR    R1, [R0,#RCC_APB1ENR] ;Включение Такты интерфейса питания

; 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 мГц

MOV32  R1, #(PLLMUL + RCC_CFGR_PLLSRC + RCC_CFGR_PPRE1_DIV2)
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
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

;.....

```


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 ;PA, PB, PC, ADC1, ADC2
STR    R1, [R0,#RCC_APB2ENR]

```

;

Init_Port ;Инициализация портов

;инициализация порт A

```

MOV32  R0, #GPIOA
STR    No11, [R0,#GPIO_ODR]
MOV32  R1, #0x22222222 ;ODR7...0 <- PP_2
STR    R1, [R0,#GPIO_CRL]

MOV32  R1,
#((AF_PP_2<<(ODR15*4))+(RES_IN<<(ODR14*4))+(RES_IN<<(ODR13*4))+(PP_2<<
(ODR12*4))+(PP_2<<(ODR11*4))+(PUD_IN<<(ODR10*4))+(PUD_IN<<(ODR9*4))+(PUD_IN<<(ODR8*4)))
STR    R1, [R0,#GPIO_CRH] ; 0xA4422888

```

;инициализация порт B

```

MOV32  R0, #GPIOB

STR    No11, [R0,#GPIO_ODR]

MOV32  R1,
#((PP_2<<(ODR7*4))+(RES_IN<<(ODR6*4))+(PP_2<<(ODR5*4))+(PP_2<<(ODR4*4))+
(PP_2<<(ODR3*4))+(RES_IN<<(ODR2*4))) ;0x24222400
STR    R1, [R0,#GPIO_CRL]

```

```

MOV32  R1,
#((RES_IN<<(ODR15*4))+(RES_IN<<(ODR14*4))+(RES_IN<<(ODR13*4))+(AF_PP_50<<
(ODR12*4))+(RES_IN<<(ODR11*4))+(AF_PP_50<<(ODR10*4))+(PP_2<<(ODR9*4))+(PP_2<<(ODR8*4)))
STR    R1, [R0,#GPIO_CRH] ; 0x444b4b22

```

;инициализация порт C

```

MOV32  R0, #GPIOC
MOV32  R1, #((RES_IN<<(ODR15*4))+(RES_IN<<(ODR14*4))+(PP_2<<(ODR13*4))+(RES_IN<<
(ODR12*4))+(RES_IN<<(ODR11*4))+(RES_IN<<(ODR10*4))+(RES_IN<<(ODR9*4))+(RES_IN<<(ODR8*4))) ;
STR    R1, [R0,#GPIO_CRH] ; 0x44244444

```

;

Init_Afio ;Инициализация альтернатив

```

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

```

;

Init_ADC2 ;Инициализация АЦП2

; Тактирование

```

MOV32  R0, #RCC
LDR    R1, [R0,#RCC_APB2ENR]
MOV32  R2, #RCC_APB2ENR_ADC2EN
ORR    R1, R1,R2
STR    R1, [R0,#RCC_APB2ENR]

```

```

MOV32  R0, #ADC2

```

```

MOV    R1, #ADC_CR2_ADON
STR    R1, [R0,#ADC_CR2]
MOV    R1, #(ADC_SMPR_SMP239_5 << 27) ;Канал 9, выборка: 239,5 мкс
STR    R1, [R0,#ADC_SMPR2]
MOV    R1, #9 ;Канал 9
STR    R1, [R0,#ADC_SQR3]
; Сбросить калибровку
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]
MOV32  R2, #RCC_APB2ENR_ADC1EN
ORR    R1, R1,R2
STR    R1, [R0,#RCC_APB2ENR]

MOV32  R0, #ADC1
MOV    R1, #ADC_CR2_ADON
STR    R1, [R0,#ADC_CR2]
MOV32  R1, #(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_SMP239_5 << 24) ;Канал 8 - выборка: 239,5 мкс
STR    R1, [R0,#ADC_SMPR2]
MOV    R1, #8 ;Канал 8
STR    R1, [R0,#ADC_SQR3]
LDR    R1, [R0,#ADC_CR2]
ORR    R1, R1,#ADC_CR2_EXTSEL_3TRGO ;Запуск от таймера 3 TRGO
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]
    MOV32  R2, #(ADC_CR2_EXTSEL_3TRGO+ADC_CR2_EXTTRIG+ADC_CR2_DMA) ;Запуска от
таймера 3 TRGO
    ORR    R1, R2
    STR    R1, [R0,#ADC_CR2]
    LDR    R1, [R0,#ADC_CR2]
    ORR    R1, R1,#ADC_CR2_ADON      ;1 - ADC1 включить
    STR    R1, [R0,#ADC_CR2]

;.....

Init_TIM3    ;Инициализация TIM3 - для запуска ADC
    ; Тактирование TIM3
    MOV32  R0, #RCC
    LDR    R1, [R0,#RCC_APB1ENR]
    MOV32  R2, #RCC_APB1ENR_TIM3EN
    ORR    R1, R1,R2
    STR    R1, [R0,#RCC_APB1ENR]

    MOV32  R0, #TIM3
    MOV    R2, #799                ; 10 мкс при f = 80 МГц
    STR    R2, [R0,#TIM_PSC]      ; TIM предделитель
    MOV    R2, #1250              ; 12,5 мс
    STR    R2, [R0,#TIM_ARR]      ; TIM регистр авто-перезагрузки
    STR    R0, [R0,#TIM_CCR1]     ; TIM capture/compare регистр 1
    MOV    R2, #TIM_CR2_MMS_1
    STR    R2, [R0,#TIM_CR2]      ; TIM управляющий регистр 2
    MOV    R2, #TIM_CR1_CEN       ; 1 - Счетчик включить
    STR    R2, [R0,#TIM_CR1]     ; TIM управляющий регистр 1

;.....

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, #4                ; Длина массива
    STR    R1, [R0,#DMA_CNDTR1]
    MOV32  R1, #(ADC1+ADC_DR)    ; Адрес регистра данных ADC1
    STR    R1, [R0,#DMA_CPAR1]
    MOV32  R1, #SRAM_Kn_0        ; Адрес начало массива в SRAM
    STR    R1, [R0,#DMA_CMAR1]
    MOV    R1, #(DMA_CCR1_MSIZE_32+DMA_CCR1_PSIZE_32+DMA_CCR1_MINC+DMA_CCR1_CIRC+
DMA_CCR1_EN)
    STR    R1, [R0,#DMA_CCR1]
    LDR    R1, [R0,#DMA_CCR1]
    ORR    R1, R1,#DMA_CCR1_EN    ; 1 - DMA1 включить

```



```

STR    R1, [R0,#DMA_CCR1]

;.....

Init_USART1 ;Инициализация USART1
            ; Тактирование USART1
MOV32    R0, #RCC
LDR      R1, [R0,#RCC_APB2ENR]
ORR      R1, R1,#RCC_APB2ENR_USART1EN
STR      R1, [R0,#RCC_APB2ENR]
            ;Разрешение глобального прерывания USART1_IRQn = 37
MOV32    R0, #NVIC_ICPR1      ; Сбросить признак отложенного прерывания 32...64
MOV      R1, #(1<<(USART1_IRQn-32))
STR      R1, [R0]
MOV32    R0, #NVIC_ISER1      ;Разрешает внешние прерывания 32...64
MOV      R1, #(1<<(USART1_IRQn-32))
STR      R1, [R0]

DIV_USART1 EQU    0x0010      ; 80/(16*5) fPCLK = 80MHz, fbaud = 5MHz

MOV32    R0, #USART1
MOV      R1, # (USART_CR1_UE)
STR      R1, [R0,#USART_CR1]

MOV      R1, #DIV_USART1
STR      R1,[R0,#USART_BRR]

LDR      R1, [R0,#USART_CR1]
ORR      R1, R1,#(USART_CR1_RXNEIE) ; Разрешения прерывания для чтения
STR      R1, [R0,#USART_CR1]

LDR      R1, [R0,#USART_CR1]
ORR      R1, R1,#(USART_CR1_RE) ; Разрешения чтения
STR      R1, [R0,#USART_CR1]

;.....

Init_USART3 ; Инициализация USART3
            ; Тактирование USART3
MOV32    R0, #RCC
LDR      R1,[R0,#RCC_APB1ENR]
MOV32    R2, #RCC_APB1ENR_USART3EN;
ORR      R1, R1,R2      ;
STR      R1,[R0,#RCC_APB1ENR]

DIV_USART3 EQU    0x0040      ; 40/(16*1,25) fPCLK = 40MHz, fbaud = 1,25MHz

MOV32    R0, #USART3
MOV      R1, #(USART_CR1_UE+USART_CR1_M)
STR      R1, [R0,#USART_CR1]

MOV      R1, #DIV_USART3
STR      R1, [R0,#USART_BRR]

LDR      R1, [R0,#USART_CR1]
ORR      R1, R1,#USART_CR1_TE      ; Разрешения передачи
STR      R1, [R0,#USART_CR1]

;.....

BL Zader_10mC      ; Ожидание выхода на режим индикатора

MOV      Rsos, No11

```

```

;.....

Init_Indic    ;Начальная загрузка индикатора
              MOV32 R0, #Indic_Tab_Init
              BL Indic_W_cdseg
              BL Zader_2mC
              MOV32 R0, #Indic_Znak_HL
              BL Indic_W_cdseg
              BL Zader_10mC
              MOV32 R0, #Indic_Znak_L
              BL Indic_W_cdseg
              BL Zader_10mC
              MOV32 R0, #Indic_Znak_tsh
              BL Indic_W_cdseg
              BL Zader_10mC
              MOV32 R0, #Indic_Znak_PM
              BL Indic_W_cdseg
              BL Zader_10mC

;.....

              BL Init_Buff
              BL Zader_10mC

;.....

              ;Вывод на индикатор Help
              MOV32 R0, #USART3
              MOV    R3, #0x20
              STRB   R3, [R0,#USART_DR]

;.....;

Duty          ; Дежурный цикл    Мигание светодиода
              MOV    R8, #10

Duty_HL_ck1    ; Цикл вывода на индикатор

              MOV32 R12, #400000

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

              MOV32 R0, #SRAM_Buff_0
              LDR    R1, [R0]
              CBNZ   R1, Duty_No11

              BL Knop

Duty_No11
              SUBS   R8, #1
              BNE    Duty_HL_ck1
              ; Мигание светодиода
              MOV32 R0, #GPIOC    ; Регистр выходных данных порта GPIO
              LDR    R1, [R0,#GPIO_ODR]
              MOV    R2, #GPIO_ODR_ODR13

```

```

EOR    R1, R1, R2
STR    R1, [R0,#GPIO_ODR]

```

```

B Duty

```

```

ENDP

```

```

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

```

```

align

```

```

;Прерывание для чтения из USART1

```

```

Inter_USART1 PROC

```

```

    PUSH    {LR}

```

```

    MOV32   R0, #USART1

```

```

    LDRB    R1, [R0,#USART_DR]

```

```

    MOV32   R0, #SRAM_Buff

```

```

    LDR     R2, [R0]

```

```

    ;Выборка номера байта

```

```

    UBFX    R3, Rsos,#Rsos_clk_B,#4    ;R2[3...0] <- Rsos[7...4]

```

```

Inter_USART1_B0    ;Загрузка Байта 0

```

```

    CMP     R3, #0

```

```

    BNE     Inter_USART1_B1

```

```

    MOV     R2, R1

```

```

    STR     R2, [R0]

```

```

    ADD     R9, R9,#0x10

```

```

    B Inter_USART1_exit

```

```

Inter_USART1_B1    ;Загрузка Байта 1

```

```

    CMP     R3, #1

```

```

    BNE     Inter_USART1_B2

```

```

    BFI     R2, R1,#8,#8

```

```

    STR     R2, [R0]

```

```

    ADD     R9, R9,#0x10

```

```

    BEQ     Inter_USART1_exit

```

```

Inter_USART1_B2    ;Загрузка Байта 2

```

```

    CMP     R3, #2

```

```

    BNE     Inter_USART1_B3

```

```

    BFI     R2, R1,#16,#8

```

```

    STR     R2, [R0]

```

```

    ADD     R9, R9,#0x10

```

```

    BEQ     Inter_USART1_exit

```

```

Inter_USART1_B3    ;Загрузка Байта 3

```

```

    CMP     R3, #3

```

```

    BFI     R2, R1,#24,#8

```

```

    BIC     R9, R9,#Rsos_clk

```

```

;Проверка на переполнение Буфера

```

```

    MOV32   R0, #SRAM_Buff_15

```

```

    LDR     R1, [R0]

```

```

    CBZ     R1, Inter_USART1_NoNoll

```

```

    MOV32   R0, #GPIOA

```

```

    MOV     R1, #GPIO_BSRR_BS8

```

```

    STR     R1, [R0,#GPIO_BSRR]

```

```

    B Inter_USART1_exit

```

```

;Установить: Буфер Переполнен

```


Inter_USART1_NoNoll

;Отыскание пустой ячейки буфера

MOV R3, #16
MOV32 R0, #SRAM_Buff_0

Inter_USART1_ck1

LDR R1, [R0],#4
CBZ R1, Inter_USART1_noll
SUB R3, Unit
CBZ R1, Inter_USART1_exit
B Inter_USART1_ck1

Inter_USART1_noll

STR R2, [R0,#-4]

Inter_USART1_exit

POP {LR}
BX LR

ENDP

;;

Init_Buff align
PROC *;Инициализация Буфера*
PUSH {R0,R1}

MOV32 R0, #SRAM_Buff_0
MOV R1, #16

Init_Buff_clk *;Обнуление Буфера*

STR Noll, [R0],#4
SUBS R1, #1
BNE Init_Buff_clk
STR Noll, [R0]

POP {R0,R1}
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

UBFHEQ 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

;;

Align

Indic_W_cdseg 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
BLNE Zader_2mC
```

```
BIC Rsos, #Rsos_Long_Comm
```

```
B Indic_W_cdseg_ck1
```

Indic_W_dRet

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

ENDP

;;

align

W_CA_DD PROC ; П/н Записи Адреса/Данных в индикатор

;Опознавание кода Адреса

```
PUSH {LR}
MOV32 R2, #GPIOB
```

```
CMP R1, #EC
BNE W_CA_DD_d
```

```
MOV R3, #(RS+R_W+E) ; R_W - "0", RS - "0", E - "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
```

; Опознавание кода Данных

```
CMP R1, #DD
BNE W_CA_DD_ad
```

```
MOV R3, #RS ; RS - "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_adi
; 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_adi

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

```
MOV32 R2, #GPIOA
STR R1, [R2, #GPIO_ODR]
BL BF_Zader_1mkC
```

```
MOV R3, #E ;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          ; Задержка на 45мкс при f = 80 МГц
BF_Zader_45mkC PROC
    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 тактов
                  NOP
                  SUBS   R12, R12,#1
                  BNE    Zader_2mC_ck1

                  POP     {R12}
                  BX      LR
                  ENDP

```

,,,,,,,,,,,,,,
,,,,,,,,,,,,,

```

align
Zader_10mC    PROC    ; Задержка на 50 мс при f = 80 МГц
                PUSH    {R12}

                MOV32   R12, #80000

Zader_10mC_ck1                ; Цикл 5 тактов
                NOP
                SUBS     R12, R12, #1
                BNE      Zader_10mC_ck1

                POP      {R12}
                BX       LR
                ENDP

```

))))))

```

align
Кноп PROC ; Обработка нажатия кнопки клавиатуры

    PUSH    {LR}
    PUSH    {R0-R3}

    MOV32   R3, #0x007F007F
    ; Проверка 4-х выборов на равенство

```

```

MOV32 R0, #SRAM_Kn_0
LDR   R1, [R0]
MOV   R1, R1, ASR #5      ; Деление на 32
AND   R1, R1, R3

```

```

LDR   R2, [R0, #(SRAM_Kn_1-SRAM_Kn_0)]
MOV   R2, R2, ASR #5      ; Деление на 32
AND   R2, R2, R3
CMP   R1, R2
BNE   Knop_Exit

```

```

LDR   R2, [R0, #(SRAM_Kn_2-SRAM_Kn_0)]
MOV   R2, R2, ASR #5      ; Деление на 32
AND   R2, R2, R3
CMP   R1, R2
BNE   Knop_Exit

```

```

LDR   R2, [R0, #(SRAM_Kn_3-SRAM_Kn_0)]
MOV   R2, R2, ASR #5      ; Деление на 32
AND   R2, R2, R3
CMP   R1, R2
BNE   Knop_Exit

```

```

Knop_equa      ; Если Все 4-е выборки равны
               ; Разбиение выборок ADC1 и ADC2
UBFX   R2, R1, #16, #7    ; выборка ADC2
UBFX   R1, R1, #0, #7     ; выборка ADC1
MOV    R3, #12
SDIV   R1, R1, R3
SDIV   R2, R2, R3

CMP     R1, R2            ; Кнопки Kn_Out_0 и Kn_Out_1 отжаты и совпадение кодов
выборки
BNE     Knop_Kn0

CMP     R1, #10           ; Кнопки Kn_Out_0 и Kn_Out_1 отжаты
BICEQ   Rsos, Rsos, #Rsos_ADC_Kn

B       Knop_Exit

```

```

Knop_Kn0
MOV32   R0, #Kn_Out_0
LDRB    R3, [R0, R1]      ; Выборка кода числа
CBZ     R3, Knop_Kn1      ; См. Kn_Out_0

TST     Rsos, #Rsos_ADC_Kn
BEQ     Knop_Usart
B       Knop_Exit

```

```

Knop_Kn1
MOV32   R0, #Kn_Out_1
LDRB    R3, [R0, R2]      ; Выборка кода функции
CBZ     R3, Knop_Exit      ; См. Kn_Out_1

```

```

Knop_Usart
TST     Rsos, #Rsos_ADC_Kn
BNE     Knop_Exit

ORR     Rsos, #Rsos_ADC_Kn

AND     R2, R3, #0x30
CMP     R2, #0x20
BNE     Knop_Usart_W

```

```
TST    Rsos, #Rsos_Kn_On_Off
BNE    Knop_Exit
```

```
BFIEQ  Rsos, R3,#0,#4
```

Knop_Usart_w

```
MOV32  R0, #USART3
STRB   R3, [R0,#USART_DR]
```

Knop_Exit

```
POP     {R0-R3}
POP     {LR}
BX      LR
ENDP
```

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

```
align
```

```
Bit_Dec    PROC    ;Преобразование 2 -> 10
              ;R2 - загрузка двоичного числа, результат в SRAM_Func_Dec
```

```
MOV32  R0,#SRAM_Func_Dec ; Адрес (младший разряд) области памяти 8-разрядного
десятичного числа
MOV     R1, #9
MOV     R3, #0x30 ;Код НУЛЯ
```

Bit_Dec_cN ; Заполнением нулями

```
STRB   R3, [R0],#1
SUBS   R1, R1,Unit
BNE    Bit_Dec_cN
```

```
MOV32  R0, #SRAM_Func_Dec
MOV     R1, #10
```

Bit_Dec_cd

```
UDIV   R3, R2,R1    ; Беззнаковое деление, R3 = R2/R1.
MLS    R4, R3,R1,R2  ; R4 = R2 - (R3 * R1) = остаток (2/10;R2 = R3 * R1)
MOV     R2, R3
ORR     R4, R4,#0x30 ; Форматирование числа Dec
STRB   R4, [R0],#1
```

```
CMP     R3, #0      ; Проверка на 0
BNE     Bit_Dec_cd
NOP
```

```
BX      LR
```

```
ENDP
```

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

```
align
```

Form_Dec PROC ; Преобразование 2 -> 10 с форматированием

```
; R0 - Адрес Кода вывода Текста числа индикатора
; R2 - загрузка двоичного числа
```

```
PUSH    {LR}
PUSH    {R0-R7,R12}
PUSH    {R0}
MOV32   R12, #99999999
CMP     R2, R12
```



```

IT HI          ; Если R0>99999999, то R0 <- #99999999
MOVHI R2, R12

BL Bit_Dec     ; Преобразование 2 -> 10

POP {R0}

MOV32 R1, #SRAM_D ; Обнуление - Памяти кода выводимого на индикатор
STR Noll, [R1]
STR Noll, [R1,#4]
STR Noll, [R1,#8]
STR Noll, [R1,#0xC]
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+zpt+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, #zpt ; Проверка на признак запятой
BEQ Form_Dec_No_zpt

LDRB R3, [R0],#1
STRB R3, [R1],#1
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, R2, Unit

```

```

BNE Form_Dec_No11_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]
    BEQ     Form_SRAM_D

NOP

Form_SRAM_D
    MOV32   R0, #SRAM_Form_Dec ; Память форматирования десятичного числа
    MOV32   R1, #SRAM_D        ; Память кода, выводимого на индикатор
    BIC     R5, R5, #(zpt+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
; 0x10 - Kn_Wd : Сдвиг По вертикали 0x20 - Rej_Help: Сброс Сопроцессора
; 0x11 - Kn_Rit : Сдвиг По горизонтали 0x21 - Rej_Fr_Tt: Работа Режим Частотомер,
Режим Импульса и периода
; 0x12 - Kn_On_Off : Пуск/Стор 0x22 - Rej_NTn: Счет импульсов,Счет времени,
Счет оборотов/мин.
; 0x13 - Kn_Res : Сброс 0x23 - Rej_Gsin: Режим Генератора SIN
; 0x24 - Rej_Gfun: Режим Генератора Gfun
; 0x25 - Rej_Pwm: Режим Генератора ШИМ
; 0x26 - Rej_Pfm: Режим Генератора ЧИМ
; 0x27 - Rej_Mod
```

```

;
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
;
;      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
Indic_Tab_Begin
;
;      DCB EC,0x85,DD,0xA5,0xB7,0xBC,0x65,0x70,0x65,0xBD,0xB8,0x65,0x3A,0x20
;
;      DCB EC,0xc2,DD,0x46,0xBE,0x65,0x70,0x2C,0x20,0x54,0xBE,0x65,0x70,0x2C,0x20,0x74,0xB8,
0xBC,0xBE,0x20
;
;      DCB EC,0x94,DD,0xA1,0x65,0xBD,0x65,0x70,0x61,0xE5,0xB8,0xC7,0x3A,0x20,0x53,0x69,0x6E,
0x2C,0x20,0xA5,0xBC,0xBE
;
;      DCB EC,0xd6,DD,0xAC,0xA5,0x4D,0x2C,0x20,0xab,0xA5,0x4D,0x2C,0x20,0x54,0x61,0xB9,0xBC
,0x65,0x70,0x00
;
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
;
;      align
Indic_Znak_HL      ;Код: 0x09
DCB EC,0x48,DD
DCB 2_00011100
DCB 2_00010101
DCB 2_00000010
DCB 2_00000100
DCB 2_00001000
DCB 2_00010101
DCB 2_00000111
DCB 2_00100000,0x00
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
;
;      align
Indic_Znak_L      ;Код: 0x0a
DCB EC,0x50,DD
DCB 2_00100000
DCB 2_00100000
DCB 2_00100000
DCB 2_00100000
DCB 2_00001001
DCB 2_00001001

```

```

DCB 2_00001111
DCB 2_00100000,0x00
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

        align
Indic_Znak_tsh ;Код: 0x0b
DCB EC,0x58,DD
DCB 2_00100000
DCB 2_00100000
DCB 2_00100000
DCB 2_00100000
DCB 2_00100000
DCB 2_00100000
DCB 2_00010101
DCB 2_00100000,0x00
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

        align
Indic_Znak_PM ;Код: 0x0c
DCB EC,0x60,DD
DCB 2_00100000
DCB 2_00100100
DCB 2_00101110
DCB 2_00100100
DCB 2_00100000
DCB 2_00001110
DCB 2_00100000
DCB 2_00100000,0x00
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

        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,0x6f,0xbe
;0xe0
DCB 0x70,0x63,0xbf,0x79,0xe4,0x78,0xe5,0xc0,0xc1,0xe6,0xc2,0xc3,0xc4,0xc5,0xc6,0xc7
;0xf0
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

        align
Tab_Rej_tab ;Работа
DCD Tab_Rej_Help ;Сброс Сопроцессора
DCD Tab_Rej_Fr_Tt ;Работа Режим Частотомер, Режим Импульса и периода
DCD Tab_Rej_NTn ;Режим Таймер
DCD Tab_Rej_Gsin ;Режим Генератора SIN
DCD Tab_Rej_Gfun ;Режим Генератора функциональный
DCD Tab_Rej_Pwm ;Режим Генератора ШИМ
DCD Tab_Rej_Pfm ;Режим Генератора ЧИМ
DCD Tab_Rej_Mod
DCD Tab Rej Cfun

```



```

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

```

```

align
Tab_Rej_Help ;Сброс Конпроцессора (0x20)
DCD 0 ; 0x00000000
DCD 0 ; 0x08000000
DCD 0 ; 0x10000000
DCD 0 ; 0x18000000
DCD 0 ; 0x20000000
DCD 0 ; 0x28000000
DCD 0 ; 0x30000000
DCD 0 ; 0x38000000
DCD Rej_Help ; 0x40000000 Описание общее
DCD Rej_Help_1 ; 0x48000000 Описание общее
DCD Rej_Help_2 ; 0x50000000 Описание FTt
DCD Rej_Help_3 ; 0x58000000 Описание FTt
DCD Rej_Help_FTt ; 0x60000000 Описание NTn
DCD Rej_Help_NTn ; 0x68000000 Описание NTn
DCD Rej_Help_NTn_1 ; 0x70000000 Описание Sin
DCD Rej_Help_Sin ; 0x78000000 Описание Sin
DCD Rej_Help_Imp ; 0x80000000 Описание Imp
DCD Rej_Help_Imp_1 ; 0x88000000 Описание Imp
DCD Rej_Help_PWFM ; 0x90000000 Описание PWM
DCD Rej_Help_PWFM_1 ; 0x98000000 Описание PWM
DCD Rej_Help_PFM ; 0xa0000000 Описание PFM
DCD Rej_Help_PFM_1 ; 0xa8000000 Описание PFM
DCD Rej_Help_Rez1 ; 0xb0000000 Описание
DCD Rej_Help_Rez1_1 ; 0xb8000000 Описание
DCD Rej_Help_Rez2 ; 0xc0000000 Описание
DCD Rej_Help_Rez2_1 ; 0xc8000000 Описание

```

```

GBLS str1 ; установить значение глобальной строковой переменной
GBLS str2

```

```

align
Rej_Help
str1 SETS "HELP: упр. Строка 1.:"
DCB EC,0x01,DD,str1
str1 SETS "Вращен. Строка 1_4:"
DCB EC,0xc0,DD,str1
str1 SETS "-измен. числа в ука-"
DCB EC,0x94,DD,str1
str1 SETS "занн. порядке числа."
DCB EC,0xd4,DD,str1,0x00

```

```

align
Rej_Help_1
str1 SETS "HELP:Кн.Строка 2,3:"
DCB EC,0x01,DD,str1
str1 SETS "-изм. положен. мар-"
DCB EC,0xc0,DD,str1
str1 SETS "кера порядка числа."
DCB EC,0x94,DD,str1
str1 SETS "- Rвых = 500м"
DCB EC,0xd4,DD,str1,0x00

```

```

align
Rej_Help_2
str1 SETS "HELP:-Ubx12max="
str2 SETS "30v"
DCB EC,0x01,DD,str1,0x0c,str2
str1 SETS "-Rbx1 = 280кОм"
DCB EC,0xc0,DD,str1

```

```

str1  SETS  "- Rbx2 = 58кОм"
      DCB EC,0x94,DD,str1
str1  SETS  "- Ubх1гист = 0,2v"
      DCB EC,0xd4,DD,str1,0x00

      align
Rej_Help_3
str1  SETS  "HELP:Ubх2гист = 1,2v"
      DCB EC,0x01,DD,str1
str1  SETS  "- Uсм2гист = 2,5v"
      DCB EC,0xc0,DD,str1
str1  SETS  "- Uвыхmax = "
str2  SETS  "10v"
      DCB EC,0x94,DD,str1,0x0c,str2,0x00

      align
Rej_Help_FTt
str1  SETS  "FTt:измер. Частоты,"
      DCB EC,0x01,DD,str1
str1  SETS  "Периода и Длит. имп."
      DCB EC,0xc0,DD,str1
str1  SETS  "- Fmax = 32МГц"
      DCB EC,0x94,DD,str1,0x00

      align
Rej_Help_NTn
str1  SETS  "NTn: счет Импульсов,"
      DCB EC,0x01,DD,str1
str1  SETS  "Таймер, Обороты/мин."
      DCB EC,0xc0,DD,str1
str1  SETS  "N:Вход1<колич. ИМП."
      DCB EC,0x94,DD,str1
str1  SETS  "T:Вход1,1-2 < ВРЕМЯ "
      DCB EC,0xd4,DD,str1,0x00

      align
Rej_Help_NTn_1
str1  SETS  "NTn; Кн. Время:"
      DCB EC,0x01,DD,str1
str1  SETS  "-Вкл/Откл и Вход2<0v"
      DCB EC,0xc0,DD,str1
str1  SETS  "и кн. "
str2  SETS  " в полож. "
      DCB EC,0x94,DD,str1,0x09,str2
str1  SETS  "n: Об/мин: Bx<3"
str2  SETS  "5v."
      DCB EC,0xd4,DD,str1,0x0b,str2,0x00

      align
Rej_Help_Sin
str1  SETS  "Гsin: генератор SIN"
      DCB EC,0x01,DD,str1
str1  SETS  "- Uвыхmax = 5v_ампл."
      DCB EC,0xc0,DD,str1
str1  SETS  "- dtфазы = "
str2  SETS  "25нс"
      DCB EC,0x94,DD,str1,0x0c,str2,0x00

      align
Rej_Help_Imp
str1  SETS  "Гфунк:гнер. ФУНКЦИЙ"
      DCB EC,0x01,DD,str1
str1  SETS  "- Uвыхmax = 10v."

```

```

        DCB EC,0xc0,DD,str1
str1    SETS  "- Тпер зависит от"
        DCB EC,0x94,DD,str1
str1    SETS  "формы сигнала;"
        DCB EC,0xd4,DD,str1,0x00

        align
Rej_Help_Imp_1
str1    SETS  "Гфунк: tфрон < 30нС."
        DCB EC,0x01,DD,str1,0x00

        align
Rej_Help_PWFM
str1    SETS  "ШЧИМ: генерат. ШИМ, ЧИМ"
        DCB EC,0x01,DD,str1
str1    SETS  "- Уупр = 0"
str2    SETS  "5v;"
        DCB EC,0xc0,DD,str1,0x0b,str2
str1    SETS  "- Уотклмаж = 5v;"
        DCB EC,0x94,DD,str1
str1    SETS  "- Уоткл > 4,2v;"
        DCB EC,0xd4,DD,str1,0x00

        align
Rej_Help_PWFM_1
str1    SETS  "ШЧИМ: - Увых = 5v;"
        DCB EC,0x01,DD,str1
str1    SETS  "- Рвых = 500м;"
        DCB EC,0xc0,DD,str1
str1    SETS  "-Рвх(Уоткл)=6,8кОм;"
        DCB EC,0x94,DD,str1
str1    SETS  "-Рвх(Уупр)=10кОм;"
        DCB EC,0xd4,DD,str1,0x00

        align
Rej_Help_PFM
str1    SETS  "ЧИМ: генератор ЧИМ"
        DCB EC,0x01,DD,str1
str1    SETS  "- Тпер/Тимп=100_max"
        DCB EC,0xc0,DD,str1,0x00

        align
Rej_Help_PFM_1
str1    SETS  "ЧИМ:"
        DCB EC,0x01,DD,str1,0x00

        align
Rej_Help_Rez1
str1    SETS  "Резерв 1"
        DCB EC,0x01,DD,str1,0x00

        align
Rej_Help_Rez1_1
str1    SETS  "Резерв 11"
        DCB EC,0x01,DD,str1,0x00

        align
Rej_Help_Rez2
str1    SETS  "Резерв 2"
        DCB EC,0x01,DD,str1,0x00

        align
Rej_Help_Rez2_1

```

```
str1 SETS "Резерв 21"
DCB EC,0x01,DD,str1,0x00
```

```

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
align

```

```
Tab_Rej_Fr_Tt ;Работа Режим Частотомер, Режим Импульса и периода (0x21)
```

```

DCD 0 ;
DCD Lin0_FTt_Ucm ; 0x08000000
DCD Lin0_FTt_MUcm ; 0x10000000
DCD Lin1_Fr ; 0x18000000
DCD Lin2_Tt ; 0x20000000
DCD Lin3_Tt ; 0x28000000
DCD 0
DCD 0
DCD Rej_Fr_Tt ; 0x40000000

```

```
Rej_Fr_Tt
```

```
str1 SETS "Режим:FTt Vcm: 0,0 В"
```

```
DCB EC,0x01,DD,str1
```

```
str1 SETS "Фпер:"
```

```
str2 SETS "Гц"
```

```
DCB EC,0xc0,DD,str1,EC,0xd2,DD,str2
```

```
str1 SETS "Тпер:"
```

```
str2 SETS "мкс"
```

```
DCB EC,0x94,DD,str1,EC,0xa5,DD,str2
```

```
str1 SETS "тимп:"
```

```
str2 SETS "мкс"
```

```
DCB EC,0xd4,DD,str1,EC,0xe5,DD,str2,EC,0xD0,0x0c,0x00
```

```
align
```

```
Lin0_FTt_Ucm
```

```
DCB EC,0x8e,DD,0x20,0x20,0x20,0x20,0x00,zpt+6 ; Сдвиг: 6
```

```
Lin0_FTt_MUcm ;Число с минусом
```

```
DCB EC,0x8e,DD,0x2d,0x20,0x20,0x20,0x00,zpt+6 ; Сдвиг: 6 0x2d - "-"
```

```
Lin1_Fr
```

```
DCB EC,0xc7,DD,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00,12 ;Сдвиг: 12
```

```
Lin2_Tt
```

```
DCB EC,0x99,DD,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00,zpt+13
```

```
;Сдвиг: 13
```

```
Lin3_Tt
```

```
DCB EC,0xd9,DD,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00,zpt+13
```

```
;Сдвиг: 13
```

```

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

```

```
align
```

```
Tab_Rej_NTn ;Режим Таймер (0x22)
```

```

DCD 0
DCD Lin0_NTn_Ucm ; 0x08000000
DCD Lin0_NTn_MUcm ; 0x10000000
DCD Lin1_N ; 0x18000000
DCD Lin2_T ; 0x20000000
DCD Lin3_n ; 0x28000000
DCD 0
DCD 0
DCD Rej_NTn ; 0x40000000

```

```
Rej_NTn
```

```
str1 SETS "Режим:NTn Vcm: 0,0 В"
```

```
DCB EC,0x01,DD,str1
```

```
str1 SETS "N:"
```

```
str2 SETS "Имп"
```

```
DCB EC,0xc0,DD,str1,EC,0xd2,DD,str2
```



```

str1  SETS  "T:"
str2  SETS  "мкс"
DCB EC,0x94,DD,str1,EC,0xa5,DD,str2
str1  SETS  "n:"
str2  SETS  "06/мин"
DCB EC,0xd4,DD,str1,EC,0xe2,DD,str2,EC,0xd0,0x0c,0x00

    align
Lin0_NTn_Ucm
    DCB EC,0x8e,DD,0x20,0x20,0x20,0x20,0x00,zpt+6      ; Сдвиг: 6
Lin0_NTn_MUcm      ; Число с минусом
    DCB EC,0x8e,DD,0x2d,0x20,0x20,0x20,0x00,zpt+6      ; Сдвиг: 6 0x2d - "-"
Lin1_N
    DCB EC,0xc6,DD,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00,12
Lin2_T
    DCB EC,0x9a,DD,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00,12
Lin3_n
    DCB EC,0xd7,DD,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00,zpt+12

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

    align
Tab_Rej_Gsin      ; Режим Генератора SIN (0x23)
    DCD      0      ;
    DCD      Lin1_Gsin_F      ; 0x08000000
    DCD      0
    DCD      Lin2_Gsin_Uamp      ; 0x18000000
    DCD      Lin3_Gsin_Ucm      ; 0x20000000
    DCD      Lin3_Gsin_MUcm      ; 0x28000000 Число с минусом
    DCD      0
    DCD      0
    DCD      Rej_Gsin      ; 0x40000000
    DCD      0
    DCD      0
    DCD      0
    DCD      Lin0_Gsin_Off      ; 0x60000000
    DCD      Lin0_Gsin_On      ; 0x68000000
    DCD      L1_Gsin_M_L      ; 0x70000000 Маркер: "<" сдвиг в Лево
    DCD      L1_Gsin_M_R      ; 0x78000000 Маркер: ">" сдвиг в Право
    DCD      L1_Gsin_M_0      ; 0x80000000 Маркер Десятичный Порядок числа 0
    DCD      L1_Gsin_M_1      ; 0x88000000
    DCD      L1_Gsin_M_2      ; 0x90000000
    DCD      L1_Gsin_M_3      ; 0x98000000
    DCD      L1_Gsin_M_4      ; 0xa0000000
    DCD      L1_Gsin_M_5      ; 0xa8000000
    DCD      L1_Gsin_M_6      ; 0xb0000000

Rej_Gsin
str1  SETS  "Режим: Гsin"
DCB EC,0x01,DD,str1
str1  SETS  "Fзад:"
str2  SETS  "Гц"
DCB EC,0xc0,DD,str1,EC,0xd2,DD,str2
str1  SETS  "Амплитуда: Va: 0,0 В"
DCB EC,0x94,DD,str1
str1  SETS  "Смещение: Vсм: 0,0 В"
DCB EC,0xd4,DD,str1,EC,0xd0,0x0c,0x00

Lin0_Gsin_Off
str1  SETS  "      "
DCB EC,0x8e,DD,str1,0x00

Lin0_Gsin_On

```

```

str1  SETS  "- Вкл"
      DCB EC,0x8e,DD,str1,0x00

Lin1_Gsin_F
      DCB EC,0xC7,DD,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00,12
Lin2_Gsin_Uamp
      DCB EC,0xa3,DD,0x20,0x20,0x20,EC,0x0c,0x00,zpt+5      ; Сдвиг: 5
Lin3_Gsin_Ucm
      DCB EC,0xe2,DD,0x20,0x20,0x20,0x20,EC,0x0c,0x00,zpt+6      ; Сдвиг: 6
Lin3_Gsin_MUcm      ;Число с минусом
      DCB EC,0xe2,DD,0x2d,0x20,0x20,0x20,EC,0x0c,0x00,zpt+6      ; Сдвиг: 6

L1_Gsin_M_0      DCB EC,0xd0,0x0e,0x00
L1_Gsin_M_1      DCB EC,0xcf,0x0e,0x00
L1_Gsin_M_2      DCB EC,0xce,0x0e,0x00
L1_Gsin_M_3      DCB EC,0xcc,0x0e,0x00
L1_Gsin_M_4      DCB EC,0xcb,0x0e,0x00
L1_Gsin_M_5      DCB EC,0xca,0x0e,0x00
L1_Gsin_M_6      DCB EC,0xc8,0x0e,0x00

L1_PWFM_M_L
L1_Gfun_M_L
L1_Gsin_M_L      DCB EC,0xc5,DD,'<',EC,0x0c,0x00 ; "<"

L1_PWFM_M_R
L1_Gfun_M_R
L1_Gsin_M_R      DCB EC,0xc5,DD,'>',EC,0x0c,0x00 ; ">"

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
align
Tab_Rej_Gfun      ;Режим Генератора Gfun (0x24)
      DCD      0
      DCD      Lin0_Gfun_Gfun      ; 0x08000000
      DCD      Lin1_Gfun_Tper      ; 0x10000000
      DCD      Lin2_Gfun_Uamp      ; 0x18000000
      DCD      Lin3_Gfun_Ucm      ; 0x20000000
      DCD      Lin3_Gfun_MUcm      ; 0x28000000 ;Число с минусом
      DCD      0
      DCD      0
      DCD      Rej_Gfun      ; 0x40000000
      DCD      0
      DCD      0
      DCD      0
      DCD      Lin0_Gfun_Off      ; 0x60000000
      DCD      Lin0_Gfun_On      ; 0x68000000
      DCD      L1_Gfun_M_L      ; 0x70000000 Маркер: "<" сдвиг в Лево
      DCD      L1_Gfun_M_R      ; 0x78000000 Маркер: ">" сдвиг в Право
      DCD      L1_Gfun_M_0      ; 0x80000000 Маркер: Десятичный Порядок числа 0
      DCD      L1_Gfun_M_1      ; 0x88000000
      DCD      L1_Gfun_M_2      ; 0x90000000
      DCD      L1_Gfun_M_3      ; 0x98000000
      DCD      L1_Gfun_M_4      ; 0xa0000000
      DCD      L1_Gfun_M_5      ; 0xa8000000
      DCD      L1_Gfun_M_6      ; 0xb0000000

Rej_Gfun
str1  SETS  "Режим:Гfun: 00"
      DCB EC,0x01,DD,str1
str1  SETS  "Тпер:"
str2  SETS  "мкС"
      DCB EC,0xc0,DD,str1,EC,0xd1,DD,str2
str1  SETS  "Амплитуда: Va: 0,0 В"
      DCB EC,0x94,DD,str1

```

```
str1 SETS "Смещение: Vcm: 0,0 В"
DCB EC,0xd4,DD,str1,EC,0xD0,0x0c,0x00
```

```
Lin0_Gfun_Off
str1      SETS      ""
          DCB EC, 0x8e, DD, str1, 0x00
```

```
Lin0_Gfun_On
str1 SETS "- Вкл"
DCB EC, 0x8e, DD, str1, 0x00
```

```
Lin0_Gfun_Gfun
    DCB EC,0x8b,DD,0x20,0x20,0x20,EC,0x0c,0x00,5 ; Cdbuz: 4
```

```
Lin1_Gfun_Tper
    DCB EC,0xc6,DD,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00,zpt+12
;C00u2: 12
```

```
Lin2_Gfun_Uamp
    DCB EC,0xa3,DD,0x20,0x20,0x20,EC,0x0c,0x00,zpt+5 ; Cδbuz: 5
```

```
Lin3_Gfun_Ucm
DCB EC,0xe2,DD,0x20,0x20,0x20,0x20,EC,0x0c,0x00,zpt+6 ; C0buz: 6
```

```
Lin3_Gfun_MUcm ;Число с минусом
DCB EC,0xe2,DD,0x2d,0x20,0x20,0x20,EC,0x0c,0x00,zpt+6 ; Сдвиг: 6 0x2d - "-"
```

;

| | |
|-------------|--|
| L1_PWFM_M_0 | |
| L1_Gfun_M_0 | DCB EC, 0xc ^f , 0x0 ^e , 0x0 ⁰ |
| L1_PWFM_M_1 | |
| L1_Gfun_M_1 | DCB EC, 0xc ^d , 0x0 ^e , 0x0 ⁰ |
| L1_PWFM_M_2 | |
| L1_Gfun_M_2 | DCB EC, 0xc ^c , 0x0 ^e , 0x0 ⁰ |
| L1_PWFM_M_3 | |
| L1_Gfun_M_3 | DCB EC, 0xc ^b , 0x0 ^e , 0x0 ⁰ |
| L1_PWFM_M_4 | |
| L1_Gfun_M_4 | DCB EC, 0xc ⁹ , 0x0 ^e , 0x0 ⁰ |
| L1_PWFM_M_5 | |
| L1_Gfun_M_5 | DCB EC, 0xc ⁸ , 0x0 ^e , 0x0 ⁰ |
| L1_PWFM_M_6 | |
| L1_Gfun_M_6 | DCB EC, 0xc ⁷ , 0x0 ^e , 0x0 ⁰ |

;

```

align
;Режим Генератора ШИМ (0x25)
Tab_Rej_Pwm
DCD 0
DCD Lin1_PWFM_Tpwm_zpt ; 0x08000000
DCD Lin2_PWFM_Tper_zpt ; 0x10000000
DCD Lin3_PWFM_Tdt_zpt ; 0x18000000
DCD 0
DCD 0
DCD 0
DCD 0
DCD Rej_PWM ; 0x40000000
DCD Lin0_PWFM_Mod_Hand ; 0x48000000
DCD Lin0_PWFM_Mod_Ext ; 0x50000000
DCD Lin0_PWFM_Mod_Gsin ; 0x58000000
DCD Lin0_PWFM_Off ; 0x60000000
DCD Lin0_PWFM_On ; 0x68000000
DCD L1_PWFM_M_L ; 0x70000000 Маркер: "<" сдвиг в Лево
DCD L1_PWFM_M_R ; 0x78000000 Маркер: ">" сдвиг в Право
DCD L1_PWFM_M_0 ; 0x80000000 Маркер Десятичный Порядок
числа 0
DCD L1_PWFM_M_1 ; 0x88000000

```

```

DCD          L1_PWFM_M_2          ; 0x90000000
DCD          L1_PWFM_M_3          ; 0x98000000
DCD          L1_PWFM_M_4          ; 0xa0000000
DCD          L1_PWFM_M_5          ; 0xa8000000
DCD          L1_PWFM_M_6          ; 0xb0000000
DCD          L2_PWFM_M_L          ; 0xb8000000 Маркер: "<" сдвиг в Лево
DCD          L2_PWFM_M_0          ; 0xc0000000 Маркер Десятичный Порядок

```

числа 0

```

DCD          L2_PWFM_M_1          ; 0xc8000000
DCD          L2_PWFM_M_2          ; 0xd0000000
DCD          L2_PWFM_M_3          ; 0xd8000000
DCD          L2_PWFM_M_4          ; 0xe0000000
DCD          L2_PWFM_M_5          ; 0xe8000000
DCD          L2_PWFM_M_6          ; 0xf0000000
DCD          L2_PWFM_M_R          ; 0xf8000000 Маркер: ">" сдвиг в Право

```

Rej_PWM

```

str1 SETS    "Режим:ШИМ"
DCB EC, 0x01, DD, str1
str1 SETS    "тшим:"
str2 SETS    "мкс"
DCB EC, 0xc0, DD, str1, EC, 0xd1, DD, str2
str1 SETS    "Тпер:"
str2 SETS    "мкс"
DCB EC, 0x94, DD, str1, EC, 0xa5, DD, str2
str1 SETS    "тмвр:"
str2 SETS    "мкс"
DCB EC, 0xd4, DD, str1, EC, 0xe5, DD, str2, EC, 0xd0, 0x0c, 0x00

```

Lin0_PWFM_Off

```

str1 SETS    ""
DCB EC, 0x91, DD, str1, 0x00

```

Lin0_PWFM_On

```

str1 SETS    "Вкл"
DCB EC, 0x91, DD, str1, 0x00

```

Lin0_PWFM_Mod_Hand

```

str1 SETS    "Ручная"
DCB EC, 0x8a, DD, str1, 0x00

```

Lin0_PWFM_Mod_Ext

```

str1 SETS    "Внешн"
DCB EC, 0x8a, DD, str1, 0x00

```

Lin0_PWFM_Mod_Gsin

```

str1 SETS    "SIN"
DCB EC, 0x8a, DD, str1, 0x00

```

Lin1_PWFM_Tpwm_zpt

```

DCB EC, 0xc6, DD, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x00, zpt+12 ;
Сдвиг: 12

```

Lin2_PWFM_Tper_zpt

```

DCB EC, 0x9a, DD, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x00, zpt+12 ;
Сдвиг: 12

```

Lin3_PWFM_Tdt_zpt

```

DCB EC, 0xda, DD, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x00, zpt+12 ;
Сдвиг: 12

```

```

L2_PWFM_M_0      DCB EC, 0xa3, 0x0e, 0x00
L2_PWFM_M_1      DCB EC, 0xa1, 0x0e, 0x00
L2_PWFM_M_2      DCB EC, 0xa0, 0x0e, 0x00
L2_PWFM_M_3      DCB EC, 0x9f, 0x0e, 0x00

```

```

L2_PWFM_M_4      DCB EC, 0x9d, 0x0e, 0x00
L2_PWFM_M_5      DCB EC, 0x9c, 0x0e, 0x00
L2_PWFM_M_6      DCB EC, 0x9b, 0x0e, 0x00

```

```

L2_PWFM_M_L      DCB EC, 0x99, DD, '<', EC, 0x0c, 0x00 ; "<"
L2_PWFM_M_R      DCB EC, 0x99, DD, '>', EC, 0x0c, 0x00 ; ">"

```

```

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

```

```

align
Tab_Rej_Pfm      ;Режим Генератора ЧИМ (0x26)
DCD              0
DCD              Lin1_PWFM_Tpwm_zpt ; 0x08000000
DCD              Lin2_PWFM_Tper_zpt ; 0x10000000
DCD              Lin3_PWFM_Tdt_zpt  ; 0x18000000
DCD              0
DCD              0
DCD              0
DCD              0
DCD              Rej_PFM             ; 0x40000000
DCD              Lin0_PWFM_Mod_Hand ; 0x48000000
DCD              Lin0_PWFM_Mod_Ext  ; 0x50000000
DCD              Lin0_PWFM_Mod_Gsin ; 0x58000000
DCD              Lin0_PWFM_Off      ; 0x60000000
DCD              Lin0_PWFM_On       ; 0x68000000
DCD              L1_PWFM_M_L        ; 0x70000000 Маркер: "<" сдвиг в Лево
DCD              L1_PWFM_M_R        ; 0x78000000 Маркер: ">" сдвиг в Право
DCD              L1_PWFM_M_0        ; 0x80000000 Маркер Десятичный Порядок числа 0
DCD              L1_PWFM_M_1        ; 0x88000000
DCD              L1_PWFM_M_2        ; 0x90000000
DCD              L1_PWFM_M_3        ; 0x98000000
DCD              L1_PWFM_M_4        ; 0xa0000000
DCD              L1_PWFM_M_5        ; 0xa8000000
DCD              L1_PWFM_M_6        ; 0xb0000000
DCD              L2_PWFM_M_L        ; 0xb8000000 Маркер: "<" сдвиг в Лево
DCD              L2_PWFM_M_0        ; 0xc0000000 Маркер Десятичный Порядок числа 0
DCD              L2_PWFM_M_1        ; 0xc8000000
DCD              L2_PWFM_M_2        ; 0xd0000000
DCD              L2_PWFM_M_3        ; 0xd8000000
DCD              L2_PWFM_M_4        ; 0xe0000000
DCD              L2_PWFM_M_5        ; 0xe8000000
DCD              L2_PWFM_M_6        ; 0xf0000000
DCD              L2_PWFM_M_R        ; 0xf8000000 Маркер: ">" сдвиг в Право

```

```

align
Rej_PFM
str1  SETS  "Режим:ЧИМ"
      DCB EC, 0x01, DD, str1
str1  SETS  "тимп:"
str2  SETS  "мкс"
      DCB EC, 0xc0, DD, str1, EC, 0xd1, DD, str2
str1  SETS  "Тчим:"
str2  SETS  "мкс"
      DCB EC, 0x94, DD, str1, EC, 0xa5, DD, str2
str1  SETS  "тмвр:"
str2  SETS  "мкс"
      DCB EC, 0xd4, DD, str1, EC, 0xe5, DD, str2, EC, 0xd0, 0x0c, 0x00

```

```

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

```

```

align
Tab_Rej_Mod      ;Режим Rej_Mod (0x27)
DCD              0

```



```

        DCD      0
        DCD      0
        DCD      0
        DCD      0
        DCD      0
        DCD      0
        DCD      0
        DCD      Rej_Mod          ; 0x40000000

Rej_Mod
str1     SETS    "Режим:Мод"
        DCB EC,0x01,DD,str1,EC,0x0c,0x00

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

                align
Tab_Rej_Cfun ;Режим Rej_Cfun (0x28)
        DCD      0
        DCD      0
        DCD      0
        DCD      0
        DCD      0
        DCD      0
        DCD      0
        DCD      0
        DCD      Rej_Cfun        ; 0x40000000

Rej_Cfun
str1     SETS    "Режим:Нфунк"
        DCB EC,0x01,DD,str1,EC,0x0c,0x00

;*****
END
```

ПО Базовая программа FTt_STM32F405RG

```
NAME FTT_STM32F405RG

#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\stm32f4xx_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\
RCC_Init_Ext_Clock_STM32F4_IAR .s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\
Rej_Fr_T_t_STM32F4_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\Rej_NTnt_STM32F4_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\Rej_Sin_STM32F4_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\Rej_Imp_STM32F4_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\Rej_PWFM_STM32F4_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\ENC0_STM32F4_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\ENC1_STM32F4_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\ENC2_STM32F4_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\ENC3_STM32F4_IAR.s"
#include "D:\Documents and Settings\Proekt IAR Systems\IAR_Bibliotek\mUser_IAR.s"

PUBLIC __vector_table
PUBLIC __iar_program_start
PUBLIC main, HardFault_IRQHandler

SECTION .text:CODE:REORDER(1)
; Сохранить таблицу векторов, даже если на нее нет ссылок
REQUIRE vector table
```

```

; Форвардная декларация разделов.
SECTION CSTACK:DATA:NOROOT(3)
SECTION .intvec:CODE:NOROOT(2)
ALIGNRAM 2
DATA

;.....

;R9 - Резултат СОСТОЯНИЯ ПРОГРАММЫ (R9)
;R9:3...0 - РЕЖИМ
Rsos_Rej_Begin EQU 0x00000000
Rsos_Rej_FtT EQU 0x00000001 ; Режим измерения Частотомер, Импульса и периода
Rsos_Rej_NtN EQU 0x00000002 ; Режим: Измерение количества Импульсов(N), Времени(T)
и оборотов(n)
Rsos_Rej_Sin EQU 0x00000003 ; Режим: Генератора SIN
Rsos_Rej_Imp EQU 0x00000004 ; Режим: Генератора Imp
Rsos_Rej_PWM EQU 0x00000005 ; Режим: Генератора ШИМ
Rsos_Rej_PFM EQU 0x00000006 ; Режим: Генератора ЧИМ
Rsos_Rej_Rezerv1 EQU 0x00000007 ;
Rsos_Rej_Rezerv2 EQU 0x00000008 ;

Rsos_OUT_On EQU 0x00000010 ; "1"- Пуск, "0"- Стоп
Rsos_Rej_On EQU 0x00000020 ; Признак выполнения Вкл Режима
Rsos_Update EQU 0x00000040 ; "1"- Обновление
Rsos_T_xSec EQU 0x00000080 ; TIM4

Rsos_Up_CC EQU 0x00000100 ; Признак выполнения Rej_Tt_TIM2_CC
Rsos_Overflow EQU 0x00000200 ; Признак Переполнение параметра

Rsos_TIM1_UP_PWM EQU 0x00001000 ; Признак выполнения Модуляция: Внешняя PWM - по
аналоговому Входу
Rsos_TIM1_UP_PFM EQU 0x00002000 ; Признак выполнения Модуляция: Внешняя PFM - по
аналоговому Входу
Rsos_ENCsEx EQU 0x00008000 ; 1 - Прошло изменение порядка

Rsos_ENC_NumOrd1_0 EQU 0x00000000 ; Десятичный Порядок числа 0x000n0000 n = 0...7
Rsos_ENC_NumOrd1_1 EQU 0x00010000
Rsos_ENC_NumOrd1_2 EQU 0x00020000
Rsos_ENC_NumOrd1_3 EQU 0x00030000
Rsos_ENC_NumOrd1_4 EQU 0x00040000
Rsos_ENC_NumOrd1_5 EQU 0x00050000
Rsos_ENC_NumOrd1_6 EQU 0x00060000
Rsos_ENC_NumOrd1_7 EQU 0x00070000
Rsos_ENC_Kn_dec1 EQU 0x00080000 ; Признак сдвига позиции: 1 - в право, 0 - в лево

Rsos_ENC_NumOrd2_0 EQU 0x00000000 ; Десятичный Порядок числа 0x00n00000 n = 0...7
Rsos_ENC_NumOrd2_1 EQU 0x00100000
Rsos_ENC_NumOrd2_2 EQU 0x00200000
Rsos_ENC_NumOrd2_3 EQU 0x00300000
Rsos_ENC_NumOrd2_4 EQU 0x00400000
Rsos_ENC_NumOrd2_5 EQU 0x00500000
Rsos_ENC_NumOrd2_6 EQU 0x00600000
Rsos_ENC_NumOrd2_7 EQU 0x00700000
Rsos_ENC_Kn_dec2 EQU 0x00800000 ; Признак сдвига позиции: 1 - в право, 0 - в лево

Rsos_ENC_Kn_dec EQU 0x01000000 ; Признак сдвига позиции: 1 - в право, 0 - в лево
Rsos_ENC_NumOrd_L2 EQU 0x02000000 ; Признак LIN2

Rsos_Imp_Crutch EQU 0x10000000 ; Признак "Костыль"
Rsos_Pacet EQU 0x20000000 ; Признак разрешения упаковки
Rsos_Att EQU 0x40000000 ; Признак изменения Задания Режима
Rsos_DMA_MM EQU 0x80000000 ; "1" - Признак режима "Память - Память"
;.....

```

```

;Признаки
Pr00 EQU 0x40 ; Признак 00,0 текста
Zpt EQU 0x80 ; Признак запятой текста
P_end_Txt EQU 0x80000000 ; Признак конца текста

Mod_Hand EQU 0x00 ; Модуляция: Ручая PWM
Mod_Ext EQU 0x01 ; Модуляция: Внешняя PWM - по аналоговому Входу
Mod_Sin EQU 0x02 ; Модуляция: Sin PWM

;.....

;Привязки к схеме
ENC0Dc_PB7 EQU GPIO_IDR_IDR_7
ENC1Dc_PB5 EQU GPIO_IDR_IDR_5
ENC1Kn_PB3 EQU GPIO_IDR_IDR_3
ENC2Dc_PA6 EQU GPIO_IDR_IDR_6
ENC2Kn_PB0 EQU GPIO_IDR_IDR_0
ENC3Dc_PB11 EQU GPIO_IDR_IDR_11

;.....

;Номера Бит вслове
ENC_NumOrd1_nBit EQU 16 ; Rsos_ENC_NumOrd1_n -Десятичный Порядок числа
ENC_NumOrd2_nBit EQU 20 ; Rsos_ENC_NumOrd2_n -Десятичный Порядок числа

;.....

ALIGNRAM 2
;Параметры вывода строк
; Param_ добавляется (ORR) к выводимому числу для опознавания вывода строк на индикатор в
; микропроцессоре

;Работа после Сброса
Param_Help EQU 0x40000000 ; Описание общее
Param_Help_1 EQU 0x48000000 ; Описание общее
Param_Help_FTt EQU 0x50000000 ; Описание FTt
Param_Help_FTt_1 EQU 0x58000000 ; Описание FTt
Param_Help_NTn EQU 0x60000000 ; Описание NTn
Param_Help_NTn_1 EQU 0x68000000 ; Описание NTn
Param_Help_Sin EQU 0x70000000 ; Описание Sin
Param_Help_Sin_1 EQU 0x78000000 ; Описание Sin
Param_Help_Imp EQU 0x80000000 ; Описание Imp
Param_Help_Imp_1 EQU 0x88000000 ; Описание Imp
Param_Help_PWM EQU 0x90000000 ; Описание PWM
Param_Help_PWM_1 EQU 0x98000000 ; Описание PWM
Param_Help_PFM EQU 0xa0000000 ; Описание PFM
Param_Help_PFM_1 EQU 0xa8000000 ; Описание PFM
Param_Help_Rezerv1 EQU 0xb0000000 ; Описание
Param_Help_Rezerv1_1 EQU 0xb8000000 ; Описание
Param_Help_Rezerv2 EQU 0xc0000000 ; Описание
Param_Help_Rezerv2_1 EQU 0xc8000000 ; Описание

;.....

; Работа Режим Частотомер, Режим Импульса и периода (0x21)
Param_Lin0_FTt_Ucm EQU 0x08000000
Param_Lin0_FTt_UcmM EQU 0x10000000
Param_Lin1_Fr EQU 0x18000000
Param_Lin2_Tt EQU 0x20000000
Param_Lin3_Tt EQU 0x28000000
Param_Rej_Fr_Tt EQU 0x40000000

;.....

```

; Режим Счет Импульсов (N), Времени (T), Скорость Оборотов (n). (0x22)

```
Param_Lin0_NTn_Ucm EQU 0x08000000
Param_Lin0_NTn_UcmM EQU 0x10000000
Param_Lin1_N EQU 0x18000000
Param_Lin2_Tim EQU 0x20000000
Param_Lin3_n EQU 0x28000000
Param_Rej_NTn EQU 0x40000000
```

;

; Режим Генератора SIN (0x23)

```
Param_Lin1_Sin_F EQU 0x08000000
Param_Lin2_Sin_Uamp EQU 0x18000000
Param_Lin3_Sin_Ucm EQU 0x20000000
Param_Lin3_Sin_UcmM EQU 0x28000000 ; Число с минусом
Param_Rej_Sin EQU 0x40000000
```

```
Param_Lin0_Sin_Off EQU 0x60000000 ; "ОТКЛ"
Param_Lin0_Sin_On EQU 0x68000000 ; "ВКЛ"
```

```
Param_L1_Sin_M_L EQU 0x70000000 ; Маркер: "<" сдвиг в Лево
Param_L1_Sin_M_R EQU 0x78000000 ; Маркер: ">" сдвиг в Право
```

```
Param_L1_Sin_M_0 EQU 0x80000000 ; Маркер Десятичный Порядок числа 0
Param_L1_Sin_M_1 EQU 0x88000000 ; Маркер Десятичный Порядок числа 1
Param_L1_Sin_M_2 EQU 0x90000000 ; Маркер Десятичный Порядок числа 2
Param_L1_Sin_M_3 EQU 0x98000000 ; Маркер Десятичный Порядок числа 3
Param_L1_Sin_M_4 EQU 0xa0000000 ; Маркер Десятичный Порядок числа 4
Param_L1_Sin_M_5 EQU 0xa8000000 ; Маркер Десятичный Порядок числа 5
Param_L1_Sin_M_6 EQU 0xb0000000 ; Маркер Десятичный Порядок числа 6
Param_L1_Sin_M_7 EQU 0xb8000000 ; Маркер Десятичный Порядок числа 7
```

;

; Режим Генератора Imp (0x24)

```
Param_Lin0_Imp_Imp EQU 0x08000000
Param_Lin1_Imp_Tper EQU 0x10000000
Param_Lin2_Imp_Uamp EQU 0x18000000
Param_Lin3_Imp_Ucm EQU 0x20000000
Param_Lin3_Imp_UcmM EQU 0x28000000 ; Число с минусом
```

```
Param_Rej_Imp EQU 0x40000000
```

```
Param_Lin0_Imp_Off EQU 0x60000000 ; "ОТКЛ"
Param_Lin0_Imp_On EQU 0x68000000 ; "ВКЛ"
```

```
Param_L1_Imp_M_L EQU 0x70000000 ; Маркер: "<" сдвиг в Лево
Param_L1_Imp_M_R EQU 0x78000000 ; Маркер: ">" сдвиг в Право
```

```
Param_L1_Imp_M_0 EQU 0x80000000 ; Маркер Десятичный Порядок числа 0
Param_L1_Imp_M_1 EQU 0x88000000 ; Маркер Десятичный Порядок числа 1
Param_L1_Imp_M_2 EQU 0x90000000 ; Маркер Десятичный Порядок числа 2
Param_L1_Imp_M_3 EQU 0x98000000 ; Маркер Десятичный Порядок числа 3
Param_L1_Imp_M_4 EQU 0xa0000000 ; Маркер Десятичный Порядок числа 4
Param_L1_Imp_M_5 EQU 0xa8000000 ; Маркер Десятичный Порядок числа 5
Param_L1_Imp_M_6 EQU 0xb0000000 ; Маркер Десятичный Порядок числа 6
Param_L1_Imp_M_7 EQU 0xb8000000 ; Маркер Десятичный Порядок числа 7
```

;

; Режим Генератора ШИМ (0x25), ЧИМ (0x26)

```
Param_Lin1_PWFM_Timp_zpt EQU 0x08000000
Param_Lin2_PWFM_Tper_zpt EQU 0x10000000
Param_Lin3_PWFM_Tdt_zpt EQU 0x18000000
```

```
Param_Rej_PWM      EQU    0x40000000
Param_Rej_PFM      EQU    0x40000000
Param_Lin0_PWFM_Mod_Hand EQU 0x48000000
Param_Lin0_PWFM_Mod_Ext EQU 0x50000000
Param_Lin0_PWFM_Mod_Sin EQU 0x58000000
```

```
Param_Lin0_PWFM_Off EQU    0x60000000 ; "ОТКЛ"
Param_Lin0_PWFM_On  EQU    0x68000000 ; "ВКЛ"
```

```
Param_L1_PWFM_M_L   EQU    0x70000000 ; Маркер: "<" сдвиг в Лево
Param_L1_PWFM_M_R   EQU    0x78000000 ; Маркер: ">" сдвиг в Право
```

```
Param_L1_PWFM_M_0   EQU    0x80000000 ; Маркер Десятичный Порядок числа 0
Param_L1_PWFM_M_1   EQU    0x88000000 ; Маркер Десятичный Порядок числа 1
Param_L1_PWFM_M_2   EQU    0x90000000 ; Маркер Десятичный Порядок числа 2
Param_L1_PWFM_M_3   EQU    0x98000000 ; Маркер Десятичный Порядок числа 3
Param_L1_PWFM_M_4   EQU    0xa0000000 ; Маркер Десятичный Порядок числа 4
Param_L1_PWFM_M_5   EQU    0xa8000000 ; Маркер Десятичный Порядок числа 5
Param_L1_PWFM_M_6   EQU    0xb0000000 ; Маркер Десятичный Порядок числа 6
```

```
Param_L2_PWFM_M_L   EQU    0xb8000000 ; Маркер: "<" сдвиг в Лево
```

```
Param_L2_PWFM_M_0   EQU    0xc0000000 ; Маркер Десятичный Порядок числа 0
Param_L2_PWFM_M_1   EQU    0xc8000000 ; Маркер Десятичный Порядок числа 1
Param_L2_PWFM_M_2   EQU    0xd0000000 ; Маркер Десятичный Порядок числа 2
Param_L2_PWFM_M_3   EQU    0xd8000000 ; Маркер Десятичный Порядок числа 3
Param_L2_PWFM_M_4   EQU    0xe0000000 ; Маркер Десятичный Порядок числа 4
Param_L2_PWFM_M_5   EQU    0xe8000000 ; Маркер Десятичный Порядок числа 5
Param_L2_PWFM_M_6   EQU    0xf0000000 ; Маркер Десятичный Порядок числа 6
```

```
Param_L2_PWFM_M_R   EQU    0xf8000000 ; Маркер: ">" сдвиг в Право
```

```
;;
```

```
    ; Режим Генератора ШИМ (0x27)
```

```
Param_Rej_Rezerv1   EQU    0x40000000
```

```
;;
```

```
    ; Режим Генератора ШИМ (0x28)
```

```
Param_Rej_Rezerv2   EQU    0x40000000
```

```
;;
```

```
    ; Оперативная память
```

```
SRAM_Temp           EQU    0x20000000 ; 4 байта Набор числа
SRAM_NTn_N          EQU    0x20000004 ; 4 байта Ячейка Измерение КОЛИЧЕСТВО ИМПУЛЬСОВ
SRAM_NTn_Tim        EQU    0x20000008 ; 4 байта Ячейка Измерение ВРЕМЕНИ L-слово
SRAM_NTn_Clk        EQU    0x2000000c ; 4 байта Ячейка Измерение ВРЕМЕНИ H-слово
SRAM_NTn_n0         EQU    0x20000010 ; 4 байта Ячейка Измерение ЧАСТОТЫ ОБОРОТОВ
SRAM_NumOrd_L1      EQU    0x20000014 ; 4 байта Ячейка Порядка числа (10^Rsos_ENC_NumOrd1)
SRAM_NumOrd_L2      EQU    0x20000018 ; 4 байта Ячейка Порядка числа (10^Rsos_ENC_NumOrd2)
SRAM_PWFM_Rebt      EQU    0x2000001c ; 4 байта Источник в PWM_Reboot_DMA2
```

```
SRAM_Frec           EQU    0x20000020 ; 4 байта Ячейка Измерение ЧАСТОТЫ
SRAM_Sin_F          EQU    0x20000020 ; 4 байта Ячейка Задания ЧАСТОТЫ генератора Sin
SRAM_Imp_Tper       EQU    0x20000020 ; 4 байта Ячейка Задания периода
SRAM_PWFM_Timp      EQU    0x20000020 ; 4 байта Ячейка Задания импульса ШИМ, ЧИМ
```

```
SRAM_Timp           EQU    0x20000024 ; 4 байта Ячейка Измерение импульса
SRAM_Imp            EQU    0x20000024 ; 4 байта Ячейка Задания Режимы ИМП
```



```

SRAM_PWFM_Tper      EQU 0x20000024      ; 4 байта Ячейка Задания периода ШИМ, ЧИМ

SRAM_Tper           EQU 0x20000028      ; 4 байта Ячейка Задания периода
SRAM_PWFM_Mod       EQU 0x20000028      ; 4 байта Ячейка Задания модуляции ШИМ, ЧИМ


SRAM_Imp_n          EQU 0x2000002c      ; 4 байта Ячейка Задания номера слова перелома
SRAM_Sin_Uamp       EQU 0x2000002c      ; 4 байта Ячейка Задание АМПЛИТУДЫ генератора Sin
SRAM_Imp_Uamp       EQU 0x2000002c      ; 4 байта Ячейка Задание АМПЛИТУДЫ генератора Imp

SRAM_FTt_Ucm        EQU 0x20000030      ; 4 байта Ячейка Задания СМЕЩЕНИЯ входного сигнала FTt
SRAM_NTn_Ucm        EQU 0x20000030      ; 4 байта Ячейка Задания СМЕЩЕНИЯ входного сигнала NTn
SRAM_Sin_Ucm        EQU 0x20000030      ; 4 байта Ячейка Задания СМЕЩЕНИЯ амплитуды генератора
Sin
SRAM_Imp_Ucm        EQU 0x20000030      ; 4 байта Ячейка Задания СМЕЩЕНИЯ амплитуды генератора
Imp

SRAM_PWFM_Tdt       EQU 0x20000034      ; 4 байта Ячейка Задания Мертвое время ШИМ, ЧИМ

SRAM_Begin          EQU 0x20000038      ; 4 байта Ячейка Счетчик адреса Описание

SRAM_Frec_Inc       EQU 0x2000003c      ; 4 байта Ячейка Измерение ЧАСТОТЫ

SRAM_Imp_Txt        EQU 0x20000050      ; 256 байт Текст импульса, слово: P_end_Txt[31]:
An[27...16]:Tn[15...0], An_max = 4000, Tn_max = 59999

SRAM_Out            EQU 0x20001000      ;Начало области памяти вывода - 128 кбайт

```

```

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

```

```

; Форвардная декларация разделов.

```

```

SECTION CSTACK:DATA:NOROOT(3)
SECTION .intvec:CODE:NOROOT(2)
ALIGNRAM 2
THUMB
DATA

```

```

__vector_table

```

```

DCD 0x20000e00      ; sfe(CSTACK)
DCD Reset_IRQHandler ; Reset Handler
DCD NMI_IRQHandler   ; NMI Handler
DCD HardFault_IRQHandler ; Hard Fault Handler
DCD MemManage_IRQHandler ; MPU Fault Handler
DCD BusFault_IRQHandler ; Bus Fault Handler
DCD UsageFault_IRQHandler ; Usage Fault Handler
DCD 0
DCD 0
DCD 0
DCD 0
DCD SVC_IRQHandler   ; SVCaLL Handler
DCD DebugMon_IRQHandler ; Debug Monitor Handler
DCD 0
DCD PendSV_IRQHandler ; PendSV Handler
DCD SysTick_IRQHandler ; SysTick Handler

```

```

; External Interrupts

```

```

DCD WWDG_IRQHandler ; Window Watchdog
DCD PVD_IRQHandler ; PVD through EXTI Line detect
DCD TAMPER_IRQHandler ; Tamper
DCD RTC_IRQHandler ; RTC
DCD FLASH_IRQHandler ; Flash
DCD RCC_IRQHandler ; RCC

```

```

DCD  EXTI0_IRQHandler      ; EXTI Line 0
DCD  EXTI1_IRQHandler      ; EXTI Line 1
DCD  EXTI2_IRQHandler      ; EXTI Line 2
DCD  EXTI3_IRQHandler      ; EXTI Line 3
DCD  EXTI4_IRQHandler      ; EXTI Line 4
DCD  DMA1_Stream0_IRQHandler ; DMA1 Stream 0 global Interrupt
DCD  DMA1_Stream1_IRQHandler ; DMA1 Stream 1 global Interrupt
DCD  DMA1_Stream2_IRQHandler ; DMA1 Stream 2 global Interrupt
DCD  DMA1_Stream3_IRQHandler ; DMA1 Stream 3 global Interrupt
DCD  DMA1_Stream4_IRQHandler ; DMA1 Stream 4 global Interrupt
DCD  DMA1_Stream5_IRQHandler ; DMA1 Stream 5 global Interrupt
DCD  DMA1_Stream6_IRQHandler ; DMA1 Stream 6 global Interrupt
DCD  ADC_IRQHandler        ; ADC1, ADC2 and ADC3s
DCD  CAN1_TX_IRQHandler     ; CAN1 TX Interrupt
DCD  CAN1_RX0_IRQHandler    ; CAN1 RX0 Interrupt
DCD  CAN1_RX1_IRQHandler    ; CAN1 RX1
DCD  CAN1_SCE_IRQHandler    ; CAN1 SCE
DCD  EXTI9_5_IRQHandler     ; EXTI Line 9..5
DCD  TIM1_BRK_TIM9_IRQHandler ; Прерывание от Break TIM1 и глобальное прерывание

TIM9
DCD  TIM1_UP_TIM10_IRQHandler ; TIM1 Update
DCD  TIM1_TRG_COM_TIM11_IRQHandler ; TIM1 Trigger and Commutation
DCD  TIM1_CC_IRQHandler      ; TIM1 Capture Compare
DCD  TIM2_IRQHandler         ; TIM2
DCD  TIM3_IRQHandler         ; TIM3
DCD  TIM4_IRQHandler         ; TIM4
DCD  I2C1_EV_IRQHandler      ; I2C1 Event
DCD  I2C1_ER_IRQHandler      ; I2C1 Error
DCD  I2C2_EV_IRQHandler      ; I2C2 Event
DCD  I2C2_ER_IRQHandler      ; I2C2 Error
DCD  SPI1_IRQHandler         ; SPI1
DCD  SPI2_IRQHandler         ; SPI2
DCD  USART1_IRQHandler       ; USART1
DCD  USART2_IRQHandler       ; USART2
DCD  USART3_IRQHandler       ; USART3
DCD  EXTI15_10_IRQHandler    ; EXTI Line 15..10
DCD  RTC_Alarm_IRQHandler     ; RTC Alarm (A and B) through EXTI Line
DCD  OTG_FS_WKUP_IRQHandler   ; USB OTG FS Wakeup through EXTI line
DCD  TIM8_BRK_TIM12_IRQHandler ; TIM8 Break Interrupt and TIM12 global interrupt
DCD  TIM8_UP_TIM13_IRQHandler ; TIM8 Update Interrupt and TIM13 global interrupt
DCD  TIM8_TRG_COM_TIM14_IRQHandler ; TIM8 Trigger and Commutation Interrupt and

TIM14 global interrupt
DCD  TIM8_CC_IRQHandler      ; TIM8 Capture Compare Interrupt
DCD  DMA1_Stream7_IRQHandler ; DMA1 Stream7
DCD  FMC_IRQHandler         ; FMC
DCD  SDIO_IRQHandler        ; SDIO
DCD  TIM5_IRQHandler        ; TIM5
; DCD SPI3_IRQHandler        ; SPI3
; DCD UART4_IRQHandler       ; UART4
; DCD UART5_IRQHandler       ; UART5
; DCD TIM6_DAC_IRQHandler    ; TIM6 and DAC1&2 underrun errors
; DCD TIM7_IRQHandler        ; TIM7
; DCD DMA2_Stream0_IRQHandler ; DMA2 Stream 0
; DCD DMA2_Stream1_IRQHandler ; DMA2 Stream 1
; DCD DMA2_Stream2_IRQHandler ; DMA2 Stream 2
; DCD DMA2_Stream3_IRQHandler ; DMA2 Stream 3
; DCD DMA2_Stream4_IRQHandler ; DMA2 Stream 4
; DCD 0                      ; Reserved
; DCD 0                      ; Reserved
; DCD CAN2_TX_IRQHandler     ; CAN2 TX
; DCD CAN2_RX0_IRQHandler    ; CAN2 RX0
; DCD CAN2_RX1_IRQHandler    ; CAN2 RX1
; DCD CAN2_SCE_IRQHandler    ; CAN2 SCE

```

```

; DCD OTG_FS_IRQHandler ; USB OTG FS
; DCD DMA2_Stream5_IRQHandler ; DMA2 Stream 5
; DCD DMA2_Stream6_IRQHandler ; DMA2 Stream 6
; DCD DMA2_Stream7_IRQHandler ; DMA2 Stream 7
; DCD USART6_IRQHandler ; USART6
; DCD I2C3_EV_IRQHandler ; I2C3 event
; DCD I2C3_ER_IRQHandler ; I2C3 error
; DCD OTG_HS_EP1_OUT_IRQHandler ; USB OTG HS End Point 1 Out
; DCD OTG_HS_EP1_IN_IRQHandler ; USB OTG HS End Point 1 In
; DCD OTG_HS_WKUP_IRQHandler ; USB OTG HS Wakeup through EXTI
; DCD OTG_HS_IRQHandler ; USB OTG HS
; DCD 0 ; Reserved
; DCD 0 ; Reserved
; DCD HASH_RNG_IRQHandler ; Hash and Rng
; DCD FPU_IRQHandler ; FPU

```

```

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

```

```

SECTION .text:CODE:REORDER:NOROOT(1)

```

```

THUMB

```

```

ALIGNRAM 2

```

```

Reset_IRQHandler ; PROC

```

```

    B main

```

```

NMI_IRQHandler ; PROC

```

```

    BKPT #1

```

```

    ; ENDP

```

```

HardFault_IRQHandler ;PROC

```

```

    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

```

```

    ; ENDP

```

```

MemManage_IRQHandler

```

```

    BKPT #3

```

```

BusFault_IRQHandler

```

```

    BKPT #4

```

```

UsageFault_IRQHandler

```

```

    BKPT #5

```

```

SVC_IRQHandler

```

```

    BKPT #6

```

```

DebugMon_IRQHandler

```

```

    BKPT #7

```

```

PendSV_IRQHandler

```

```

    BKPT #8

```

```

SysTick_IRQHandler

```

```

    BKPT #9

```

```

;-----

```

```

WWDG_IRQHandler

```

```

    ; BKPT #10

```

```

PVD_IRQHandler

```

```

    ; BKPT #4

```

```

TAMPER_IRQHandler

```

```

    ; BKPT #4

```

```

RTC_IRQHandler

```

```

    ; BKPT #4

```

```

FLASH_IRQHandler
    ; BKPT #4
RCC_IRQHandler
    ; BKPT #4
EXTI0_IRQHandler ;ENC2sbEx
    B Inter_ENC2sbEx
EXTI1_IRQHandler ;ENC3sbEx
    ; B Inter_ENC3sbEx
EXTI2_IRQHandler
    B Inter_TIMER
EXTI3_IRQHandler ;ENC1sbEx
    B Inter_ENC1sbEx
    ; BKPT #4
EXTI4_IRQHandler ;ENC1Ex
    B Inter_ENC1Ex
    ; BKPT #4
DMA1_Stream0_IRQHandler
    ; BKPT #4
DMA1_Stream1_IRQHandler
    ;BKPT #4
DMA1_Stream2_IRQHandler
    ; BKPT #4
DMA1_Stream3_IRQHandler
    ; BKPT #4
DMA1_Stream4_IRQHandler
    ; BKPT #4
DMA1_Stream5_IRQHandler
    ; BKPT #4
DMA1_Stream6_IRQHandler
    ; BKPT #4
ADC_IRQHandler
    ; BKPT #4
CAN1_TX_IRQHandler
    ; BKPT #4
CAN1_RX0_IRQHandler
    ; BKPT #4
CAN1_RX1_IRQHandler
    ; BKPT #4
CAN1_SCE_IRQHandler
    ; BKPT #4
EXTI9_5_IRQHandler ;ENC2Ex: EXTI7, ENC0Ex: EXTI8, ENC0sbEx: EXTI9
    m_r0_B Inter_EXTI9_5
    ; BKPT #4
TIM1_BRK_TIM9_IRQHandler
    B Inter_TIM9 ; TIM9 прерывания Обновление
    ; BKPT #4
TIM1_UP_TIM10_IRQHandler
    B Inter_TIM1_UP
    ; BKPT #4
TIM1_TRG_COM_TIM11_IRQHandler
    ; BKPT #4
TIM1_CC_IRQHandler
    ; BKPT #4
TIM2_IRQHandler
    B Inter_TIM2 ; TIM2 прерывания Обновление

TIM3_IRQHandler
    ; BKPT #4

TIM4_IRQHandler ; TIM4 Основное прерывание
    B Inter_TIM4

I2C1_EV_IRQHandler

```

```

        ; BKPT #4
I2C1_ER_IRQHandler
        ; BKPT #4
I2C2_EV_IRQHandler
        ; BKPT #4
I2C2_ER_IRQHandler
        ; BKPT #4
SPI1_IRQHandler
        ; BKPT #4
SPI2_IRQHandler
        ; BKPT #4
USART1_IRQHandler
        ; BKPT #4
USART2_IRQHandler
    m_r0_B Inter_USART2      ;Прерывание от USART2

USART3_IRQHandler
    ; BKPT #4

EXTI15_10_IRQHandler      ;ENC3Ex: EXTI10
    m_r0_B Inter_EXTI15_10      ;Прерывание от кнопок PB10
    ; BKPT #4
RTC_Alarm_IRQHandler
    ; BKPT #4
OTG_FS_WKUP_IRQHandler
    ; BKPT #4
TIM8_BRK_TIM12_IRQHandler
    ; BKPT #4
TIM8_UP_TIM13_IRQHandler
    ; BKPT #4
TIM8_TRG_COM_TIM14_IRQHandler
    ; BKPT #4
TIM8_CC_IRQHandler
    ;BKPT #4
DMA1_Stream7_IRQHandler
    ;BKPT #4
FMC_IRQHandler
    ; BKPT #4
SDIO_IRQHandler
    BKPT #4

TIM5_IRQHandler
    B Inter_TIM5      ; TIM5 прерывания Обновление

;SPI3_IRQHandler
;      ;BKPT #4
;UART4_IRQHandler
;      ;BKPT #4
;UART5_IRQHandler
;      ;BKPT #4
;TIM6_DAC_IRQHandler
;      ;BKPT #4
;TIM7_IRQHandler
;      ;BKPT #4
;DMA2_Stream0_IRQHandler
;      ;BKPT #4
;DMA2_Stream1_IRQHandler
;      ;BKPT #4
;DMA2_Stream2_IRQHandler
;      ;BKPT #4
;DMA2_Stream3_IRQHandler
;      ;BKPT #4
;DMA2_Stream4_IRQHandler

```



```

;                ;;BKPT #4
;CAN2_TX_IRQHandler
;                ;;BKPT #4
;CAN2_RX0_IRQHandler
;                ;;BKPT #4
;CAN2_RX1_IRQHandler
;                ;;BKPT #4
;CAN2_SCE_IRQHandler
;                ;;BKPT #4
;OTG_FS_IRQHandler
;                ;;BKPT #4
;DMA2_Stream5_IRQHandler
;                ;;BKPT #4
;DMA2_Stream6_IRQHandler
;                ;;BKPT #4
;DMA2_Stream7_IRQHandler
;                ;;BKPT #4
;USART6_IRQHandler
;                ;;BKPT #4
;I2C3_EV_IRQHandler
;                ;;BKPT #4
;I2C3_ER_IRQHandler
;                ;;BKPT #4
;OTG_HS_EP1_OUT_IRQHandler
;                ;;BKPT #4
;OTG_HS_EP1_IN_IRQHandler
;                ;;BKPT #4
;OTG_HS_WKUP_IRQHandler
;                ;;BKPT #4
;OTG_HS_IRQHandler
;                ;;BKPT #4
;HASH_RNG_IRQHandler
;                ;;BKPT #4
;FPU_IRQHandler
;                BKPT #4

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

                ALIGNRAM 2
                THUMB
__iar_program_start

main            ; PROC
                NOP

;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
; R11 - "1" - единица (R11)
; R10 - "0" - ноль (R10)
                MOV    R10, #0
                MOV    R11, #1
;.....

                BL    RCC_Init_ext_10_160            ; Внешняя Тактировка 10 МГц : 160 МГц

;.....

Peref_Init      ;инициализация периферии

                MOV32  R0, #RCC
                LDR    R1, [R0,#RCC_AHB1ENR]
                ORR    R1, R1, #RCC_AHB1ENR_GPIOAEN    ; включение тактов порта A IO
                STR    R1, [R0,#RCC_AHB1ENR]

                LDR    R1, [R0,#RCC_AHB1ENR]

```

```

    ORR     R1, #RCC_AHB1ENR_GPIOBEN      ; включение тактов порта B IO
STR       R1, [R0,#RCC_AHB1ENR]

    LDR     R1, [R0,#RCC_AHB1ENR]
    ORR     R1, R1, #RCC_AHB1ENR_GPIOCEN   ; включение тактов порта C IO
STR       R1, [R0,#RCC_AHB1ENR]

    LDR     R1, [R0,#RCC_APB2ENR]
    ORR     R1, R1, #RCC_APB2ENR_SYSCFGEN   ; включение тактовой частоты контроллера
конфигурации системы.
STR       R1, [R0,#RCC_APB2ENR]

    LDR     R1, [R0,#RCC_APB2ENR]
    ORR     R1, R1, #RCC_APB2ENR_ADC1EN     ; включение тактов ADC1

    MOV     R1, #64

Peref_Init_Zd
SUBS      R1, R1, R11
BNE Peref_Init_Zd

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

GPIO_Init          ;инициализация портов
MOV32        R1, #GPIOA
LDR          R2, [R1,#GPIO_MODER]
MOV32        R3, #(GPIO_MODER_1_AN+GPIO_MODER_3_AF+GPIO_MODER_4_AN+GPIO_MODER_5_AN)
ORR          R2, R2,R3
STR          R2, [R1,#GPIO_MODER]           ;Регистр режима порта GPIO (GPIOx_MODER)

    LDR      R2, [R1,#GPIO_OSPEEDER]
    MOV32    R3, #(GPIO_OSPEEDER_OSPEEDR0_100+GPIO_OSPEEDER_OSPEEDR2_12+
GPIO_OSPEEDER_OSPEEDR3_100+GPIO_OSPEEDER_OSPEEDR4_12+GPIO_OSPEEDER_OSPEEDR8_100+
GPIO_OSPEEDER_OSPEEDR12_100+GPIO_OSPEEDER_OSPEEDR15_100) ; Высокая скорость < 50 МГц
    ORR      R2, R2,R3
    STR      R2, [R1,#GPIO_OSPEEDER]

    MOV      R2, #(GPIOA_AFR_L_PA3_USART2_RX)
    STR      R2, [R1,#GPIO_AFR_L]           ; GPIO L регистр альтернативных функций

;.....

    MOV32    R1, #GPIOB
    LDR      R2, [R1,#GPIO_MODER]
    MOV32    R3, #(GPIO_MODER_2_OUT+GPIO_MODER_6_AF+GPIO_MODER_12_OUT+
GPIO_MODER_14_AF)
    ORR      R2, R2,R3
    STR      R2, [R1,#GPIO_MODER]

    MOV32    R2, #GPIOB_AFR_L_PB6_USART1_TX
    STR      R2, [R1,#GPIO_AFR_L]           ; GPIO L регистр альтернативных функций

    MOV32    R2, #(GPIOB_AFR_H_PB14_TIM1_CH2N)
    STR      R2, [R1,#GPIO_AFR_H]           ; GPIO H регистр альтернативных функций

    MOV32    R2, #(GPIO_OSPEEDER_OSPEEDR6_100+GPIO_OSPEEDER_OSPEEDR14_50)
    STR      R2, [R1,#GPIO_OSPEEDER]       ; Регистр скорости вывода порта GPIO

    MOV32    R2, #GPIO_BSRR_BS_12
    STR      R2, [R1,#GPIO_BSRR]

;.....

    MOV32    R1, #GPIOC
    MOV32    R2, #(GPIO_MODER_11_OUT+GPIO_MODER_10_OUT+GPIO_MODER_9_OUT+

```

```
GPIO_MODER_8_OUT+GPIO_MODER_7_OUT+GPIO_MODER_6_OUT+GPIO_MODER_5_OUT+GPIO_MODER_4_OUT+
GPIO_MODER_3_OUT+GPIO_MODER_2_OUT+GPIO_MODER_1_OUT+GPIO_MODER_0_OUT) ; 01: Режим вывода
общего назначения
```

STR R2, [R1,#GPIO_MODER]

MOV R2, #0

STR R2, [R1,#GPIO_OTYPER]

```
MOV    R2, #0xffffffff
```

```
STR R2, [R1,#GPIO_OSPEEDER] ; Регистр скорости вывода порта GPIO
```

```
MOV    R2, #GPIO_ODR_ODR_11
```

STR R2, [R1, #GPIO_ODR] ; Регистр выходных данных порта GPIO

;

```
MOV    R1, #32000
```

Peref_Init_Zd1

SUBS R1, R1, R11

BNE Peref_Init_Zd1

;

```
MOV32 R0, #SYSCFG
```

```
MOV R1, #SYSCFG_CMPCR_CMP_PD ; "1" - Включение питания компенсационной
```

ячейки

```
STR    R1, [R0,#SYSCFG_CMPCR]
```

; SYSCFG Регистр управления компенсационной

ячейкой

Peref_Init_Zd2

```
LDR    R1, [R0,#SYSCFG_CMPCR]
```

TST R1, #SYSCFG CMPCR READY ; Флаг готовности ячейки компенсации

BEQ Peref Init Zd2

EXTI_Init

```
MOV R1, #(SYSCFG_EXTICR1_EXTI0_PB+SYSCFG_EXTICR1_EXTI1_PB+
```

SYSCFG_EXTICR1_EXTI3_PB)

```
STR    R1, [R0,#SYSCFG_EXTICR1] ; SYSCFG Регистр конфигурации внешнего
```

прерывания EXTIO, EXTI1, EXTI3

```
MOV    R1, #(SYSCFG_EXTICR2_EXTI7_PA+SYSCFG_EXTICR2_EXTI4_PB)
```

```
STR R1, [R0,#SYSCFG_EXTICR2] ; SYSCFG Регистр конфигурации внешнего
```

прерывания EXTI4, EXTI7

```
MOV R1, #(SYSCFG_EXTICR3_EXTI8_PB+SYSCFG_EXTICR3_EXTI9_PB+
```

SYSCFG_EXTICR3_EXTI10_PB)

```
STR    R1, [R0,#SYSCFG_EXTICR3] ; SYSCFG Регистр конфигурации внешнего
```

прерывания EXTI8, EXTI9, EXTI10

;

```
MOV32 R2, #NVIC_ICER0 ; Отменяет разрешение внешних прерываний 0...32
```

```
MOV     R1, #0xFFFFFFFF
```

STR R1, [R2]

```
STR R1, [R2, #4] ; NVIC ICER1 Отменяет разрешение внешних прерываний
```

32...64

```
MOV32 R0, #EXTI
```

```
MOV     R1, #(EXTI_RTSR_TR0+EXTI_RTSR_TR1+EXTI_RTSR_TR2+EXTI_RTSR_TR3+
```

EXTI_RTSR_TR4+EXTI_RTSR_TR7+EXTI_RTSR_TR8+EXTI_RTSR_TR9+EXTI_RTSR_TR10) ; *Нарастающий бит конфигурации события запуска*

```
STR    R1, [R0,#EXTI_RTSR]
```

; EXTI Регистр выбора запуска нарастания

```
MOV R1, #EXTI_FTSR_TR2
```

```

STR    R1, [R0,#EXTI_FTSR]           ; EXTI Регистр выбора запуска спада

MOV     R1, #(EXTI_PR_PR0+EXTI_PR_PR1+EXTI_PR_PR2+EXTI_PR_PR3+EXTI_PR_PR4+
EXTI_PR_PR7+EXTI_PR_PR8+EXTI_PR_PR9+EXTI_PR_PR10)   ; Отложенный бит. Этот бит очищается
программированием в него «1».
STR     R1, [R0,#EXTI_PR]             ; EXTI Регистр ожидания
MOV     R1, #(EXTI_IMR_MR0+EXTI_IMR_MR1+EXTI_IMR_MR2+EXTI_IMR_MR3+EXTI_IMR_MR4+
EXTI_IMR_MR7+EXTI_IMR_MR8+EXTI_IMR_MR9+EXTI_IMR_MR10) ; Маска прерывания
STR     R1, [R0,#EXTI_IMR]            ; EXTI Регистр маски прерывания

MOV32   R0, #NVIC_ISER0               ; Разрешает внешние прерывания 0...32
LDR      R1, [R0]
MOV32   R2, #((1<<EXTI0_IRQn)+(1<<EXTI2_IRQn)+(1<<EXTI3_IRQn)+(1<<EXTI4_IRQn)+
(1<<EXTI9_5_IRQn))                   ; Разрешение глобального прерывания
ORR      R1, R1,R2
STR      R1, [R0]

MOV32   R0, #NVIC_ISER1               ; Разрешает внешние прерывания 32...64
LDR      R1, [R0]
MOV32   R2, #(1<<(EXTI15_10_IRQn-32)) ; Разрешение глобального прерывания PC13,
NVIC_EXTI15_10 = 40
ORR      R1, R1,R2
STR      R1, [R0]

MOV      R1, #64
Peref_Init_Zd3
SUBS     R1, R1,R11
BNE Peref_Init_Zd3

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

DAC_Init_   ; Инициализация DAC
MOV32   R0, #RCC
LDR      R1, [R0,#RCC_APB1ENR]
ORR      R1, R1, #(RCC_APB1ENR_DACEN) ; Включение тактов DAC
STR      R1, [R0,#RCC_APB1ENR]

MOV32   R0, #DAC
MOV      R1, #(DAC_CR_EN1+DAC_CR_EN2)
STR      R1, [R0,#DAC_CR]
MOV      R1, #2048
STR      R1, [R0,#DAC_DHR12R1]
STR      R1, [R0,#DAC_DHR12R2]

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

Init_USART2 ; Инициализация USART2
; Тактирование USART2
MOV32   R0, #RCC
LDR      R1, [R0,#RCC_APB1ENR]
MOV32   R2, #RCC_APB1ENR_USART2EN;
ORR      R1, R1,R2 ;
STR      R1, [R0,#RCC_APB1ENR]
; Разрешение глобального прерывания USART2_IRQn = 38
MOV32   R0, #NVIC_ICPR1 ; Сбросить признак отложенного прерывания 32...64
MOV      R1, #(1<<(USART2_IRQn-32))
STR      R1, [R0]
MOV32   R0, #NVIC_ISER1 ; Разрешает внешние прерывания 32...64
MOV      R1, #(1<<(USART2_IRQn-32))
STR      R1, [R0]

DIV USART2 EQU 0x0040 ; 40/(16*1,25), fPCLK = 40MHz, fbaud = 1,25MHz

```

```

MOV32 R0, #USART2
MOV    R1, #(USART_CR1_UE)
STR    R1, [R0,#USART_CR1]

MOV    R1, #DIV_USART2
STR    R1, [R0,#USART_BRR]

LDR    R1, [R0,#USART_CR1]
ORR    R1, R1,#(USART_CR1_RXNEIE) ; Разрешения прерывания для чтения
STR    R1, [R0,#USART_CR1]

LDR    R1, [R0,#USART_CR1]
ORR    R1, R1,#(USART_CR1_RE)     ; Разрешения чтения
STR    R1, [R0,#USART_CR1]

```

.....

Init_USART1 ; Инициализация USART1
; Тактирование USART1

```

MOV32 R0, #RCC
LDR    R1, [R0,#RCC_APB2ENR]
MOV32 R2, #RCC_APB2ENR_USART1EN;
ORR    R1, R1,R2
STR    R1, [R0,#RCC_APB2ENR]

```

DIV_USART1 EQU 0x0010 ; 80/(16*5), fPCLK = 80MHz, fbaud = 5MHz

```

MOV32 R0, #USART1
MOV    R1, #(USART_CR1_UE)
STR    R1, [R0,#USART_CR1]

MOV    R1, #DIV_USART1
STR    R1, [R0,#USART_BRR]

LDR    R1, [R0,#USART_CR1]
ORR    R1, R1,#USART_CR1_TE     ; Разрешения передачи
STR    R1, [R0,#USART_CR1]

```

.....

FP_Init ; Инициализация Сопроцессора

```

MOV32 R0, #CPACR ; Регистр управления Доступом
LDR    R1, [R0]
ORR    R1, R1,#(CPACR_CP11_FA+CPACR_CP10_FA) ; Полный доступ
STR    R1, [R0]

```

.....

DSB ; Команда DSB устанавливает специальный барьер синхронизации данных при доступе к памяти.

ISB ; Команда ISB устанавливает барьер синхронизации инструкций.

.....

```

MOV32 R0, #GPIOB ; Регистр установки/сброса битов порта GPIO
MOV    R1, #GPIO_BSRR_BR_2 ; Сброс бита PB2
STR    R1, [R0,#GPIO_BSRRH]

```

```

MOV32 R0, #SRAM_Temp
STR    R10, [R0]

```

```

MOV32 R0, #SRAM_Begin
STR    R10, [R0]

```



```

MOV    R9, #Rsos_Rej_Begin

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
main_Idic
MOV32  R4, #2000000      ;Мигание лампочки

main_C
BL Contr_Rsos
SUBS   R4, R4, R11
BNE   main_C

MOV32  R0, #(GPIOB + GPIO_ODR) ; Регистр выходных данных порта GPIO
LDR    R1, [R0]
MOV    R2, #GPIO_ODR_ODR_2
EOR    R1, R1, R2
STR    R1, [R0]

B main_Idic

; ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
ALIGNRAM 2
Inter_TIM1_UP      ; PROC
PUSH    {LR}
; Флаг прерывания : Сброс
MOV32   R0, #TIM1
LDR     R1, [R0,#TIM_SR] ; TIM Регистр состояния
BIC     R1, R1, #TIM_SR_UIF
STR     R1, [R0,#TIM_SR]

; Усреднение по PwFM_Tper
MOV32   R1, #SRAM_PwFM_Rebt
LDR     R3, [R1] ; Источник в PwFM_Reboot_DMA1

MOV32   R0, #SRAM_Out ; Начало области памяти вывода - 128 кбайт
LDRH    R2, [R0],#2

Inter_TIM1_UP_ck1
SUBS    R3, R3,R11
BEQ     Inter_TIM1_UP_div

LDRH    R4, [R0],#2
ADD     R2, R2,R4
B Inter_TIM1_UP_ck1

Inter_TIM1_UP_div
LDR     R3, [R1] ; Источник в PWM_Reboot_DMA1
UDIV    R2, R2,R3 ; Результат усреднения
USAT    R2, #12,R2 ; Усреднения значения Kadc
;.....
; Проверка на Режим
AND     R5, R9,#0xf

CMP     R5, #Rsos_Rej_PFM
BEQ     Inter_TIM1_UP_PFM
;.....

Inter_TIM1_UP_PWM

```

```
MOV32 R1, #SRAM_PWFM_Tper
LDR R3, [R1]
SUB R2, R2, #1 ; Kadc
USAT R2, #12, R2
LSL R3, R3, #2 ; Tper*8/2

VMOV S2, S3, R2, R3 ; Пересылка числа в сопроцессор с плавающей запятой
VCVT.F32.U32 S2, S2 ; Kadc
VCVT.F32.U32 S3, S3 ; Tper*8/2
```

```
MOV32 R1, #Rej_PWFM_Tmax
VLDR.F32 S6, [R1] ; Tmax = 4095
VDIV.F32 S3, S3, S6 ; Kadc*8/2/Tmax
VMUL.F32 S2, S2, S3 ; (Kadc*8/2/Tmax)* Kadc
```

```
MOV32 R1, #Rej_PWM_Kkor
VLDR.F32 S6, [R1]
VMUL.F32 S2, S2, S6
VCVT.U32.F32 S2, S2
VMOV R4, S2
```

```
ORR    R9, R9, #Rsos_TIM1_UP_PWM
```

B Inter_TIM1_UP_PWFM

;

Inter_TIM1_UP_PFM

```
MOV32 R1, #SRAM_PWFM_Timp
LDR R3, [R1] ; Timp
```

```
VMOV          S2,S3, R2,R3 ; Пересылка числа в сопроцессор с плавающей запятой
VCVT.F32.U32 S2, S2 ; Tadc
VCVT.F32.U32 S3, S3 ; Timp
```

```
MOV32 R0, #Rej_PFM_Kkor
VLDR.F32 S5, [R0] ; Kkor
VMUL.F32 S2, S2, S5 ; Kadc * Kkor
```

```
MOV32 R0, #Rej_PWFM_Tmax
VLDR.F32 S5, [R0] ; Tmax = 4095
VSUB.F32 S2, S5, S2 ; Обратное значение Tadc = 4095 - Tadc
```

```
MOV32 R1, #Rej_PFM_Kpau$
VLDR.F32 S6, [R1] ; Kpau = (Timp/Tper - 1)*8/4095, Timp/Tper = 100
VMUL.F32 S2, S2, S6 ; Kadc * Kkor * Kpau
VMUL.F32 S2, S2, S3 ; Tpaus * Timp
```

```
VCVT.U32.F32 S2, S2
VMOV          R2, S2
USAT         R2, #30, R2
```

ADD R3, R2, R3, LSL #4

```
ORR    R9, R9, #Rsos TIM1 UP PFM
```

;

Inter TIM1 UP PWFM

```
BL Rej_PWFM_TIM1 ; загрузка TIM1
```

BIC R9, R9, #(Rsos TIM1 UP PWM+Rsos TIM1 UP PFM)

```

; Флаг прерывания : Сброс
MOV32 R0, #TIM1
LDR R1, [R0, #TIM_SR] ; TIM Регистр состояния
BIC R1, R1, #TIM_SR_UIF
STR R1, [R0, #TIM_SR]

; Сбрасывает признак отложенного прерывания 0...32
MOV32 R0, #NVIC_ICPR0
MOV R1, #((1<<TIM4_IRQn)+(1<<TIM1_UP_TIM10_IRQn))
STR R1, [R0]

POP {LR}
BX LR

; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Inter_TIM4      ALIGNRAM 2
                 ; PROC ;Прерывание от TIM4: Режим 0: 1Сек, Режим 1: 0,02Сек
PUSH {LR}

MOV32 R0, #TIM4
LDR R1, [R0, #TIM_SR] ; TIM Регистр состояния
BIC R1, R1, #TIM_SR_UIF ; Флаг прерывания : Сброс
STR R1, [R0, #TIM_SR]

MOV32 R0, #TIM1
LDR R2, [R0, #TIM_CNT] ; TIM регистр счетчика
MOV32 R0, #TIM8
LDR R3, [R0, #TIM_CNT] ; TIM регистр счетчика
BFI R2, R3, #16, #16

AND R3, R9, #0x0f
CMP R3, #Rsos_Rej_NTn
BNE Inter_TIM4_FTt

Inter_TIM4_NTn
MOV32 R1, #SRAM_NTn_N
STR R2, [R1]
MOV32 R0, #TIM9
LDR R2, [R0, #TIM_CNT]

MOV32 R1, #SRAM_NTn_Clk
LDR R3, [R1]

BFI R2, R3, #16, #16
MOV32 R1, #SRAM_NTn_Tim
STR R2, [R1]

B Inter_TIM4_exit

Inter_TIM4_FTt
MOV32 R1, #SRAM_Frec
STR R2, [R1]

MOV32 R0, #TIM1
STR R10, [R0, #TIM_CNT]
MOV32 R0, #TIM8
STR R10, [R0, #TIM_CNT]

Inter_TIM4_exit

ORR R9, R9, #Rsos_T_xSec

```

```

MOV32 R0, #TIM4

LDR R1, [R0,#TIM_CR1] ; TIM управляющий регистр 1
ORR R1, R1, #TIM_CR1_CEN ; 1 - Счетчик включить
STR R1, [R0,#TIM_CR1]

POP {LR}
BX LR

; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Inter_TIM2
ALIGNRAM 2
; PROC ; Tper, Timp
PUSH {LR}
;BKPT #6
MOV32 R0, #TIM2
STR R10, [R0,#TIM_DIER] ; TIM DMA /прерывания регистр разрешения
STR R10, [R0,#TIM_SR]
MOV32 R1, #SRAM_Tper
LDR R2, [R0,#TIM_CCR1]
STR R2, [R1] ;SRAM_Tper
LDR R2, [R0,#TIM_CCR2]
STR R2, [R1,#4] ;SRAM_Timp

POP {LR}
BX LR

; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Inter_TIM5
ALIGNRAM 2
; PROC ; Tbp
PUSH {LR}
MOV32 R0, #TIM5
STR R10, [R0,#TIM_DIER] ; TIM DMA /прерывания регистр разрешения

STR R10, [R0,#TIM_SR]

LDR R2, [R0,#TIM_CCR1] ; TIM capture/compare регистр 2
MOV32 R1, #SRAM_NTn_n0
STR R2, [R1]

POP {LR}
BX LR

;ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Inter_TIM9
ALIGNRAM 2
; PROC ; Tbp
PUSH {LR}
MOV32 R0, #TIM9
STR R10, [R0,#TIM_DIER] ; TIM DMA /прерывания регистр разрешения

STR R10, [R0,#TIM_SR]

MOV32 R1, #SRAM_NTn_Clk
LDR R2, [R1]
ADD R2, R2, R11
STR R2, [R1]

MOV32 R0, #TIM9
MOV R1, #TIM_DIER_UIE ; Capture/Compare 1 Включить прерывание

```

```

STR    R1, [R0,#TIM_DIER]  ; TIM DMA /прерывания регистр разрешения

POP    {LR}
BX     LR

      ; ENDP

```

```

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

```

```

      ALIGNRAM 2
Inter_TIMER ; PROC ;Прерывание от PA2: Режим 1:Измерение времени
PUSH    {LR}
MOV32   R0, #EXTI
MOV     R1, #EXTI_PR_PR2
STR     R1, [R0,#EXTI_PR]      ; EXTI Регистр ожидания

MOV32   R1, #GPIOA
LDR     R2, [R1,#GPIO_IDR]     ; Чтение из порта PA2

MOV32   R0, #TIM9

LDR     R3, [R0,#TIM_CR1]
MOV32   R1, #SRAM_NTn_Clk
TST     R2, #GPIO_IDR_IDR_2
ITTE    NE
STRNE   R10, [R0,#TIM_CNT]
ORRNE   R3, R3, #TIM_CR1_CEN    ; 1 - Счетчик включить
STRNE   R10, [R1]
BICEQ   R3, R3, #TIM_CR1_CEN    ; 0 - Счетчик отключить

STR     R3, [R0,#TIM_CR1]      ; TIM управляющий регистр 1

POP     {LR}
BX      LR

      ; ENDP

```

```

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

```

```

      ALIGNRAM 2
ENCsbEx_NumOrd ; PROC Определение и обработка порядка числа
      ; Вход: R2 (Десятичный Порядок числа Rsos_ENC_NumOrd <-- Rsos), [R1]: R1 -
адрес числа Порядка, R12 - Мах число
      ; Выход: R5 -- > [R0], R0 - адрес числа
PUSH    {LR}

TST     R9, #Rsos_ENC_NumOrd_L2
ITE     EQ
MOVEQ   R6, #Rsos_ENC_Kn_dec1
MOVNE   R6, #Rsos_ENC_Kn_dec2

;Проерка на 0
USAT    R2, #3,R2      ; Прядок числа
CMP     R2, #0
IT      EQ
BICEQ   R9, R9,R6

; Определение порядка числа
TST     R9, R6
ITE     EQ
ADDEQ   R2, R2,R11
SUBNE   R2, R2,R11
USAT    R2, #3,R2

```

```

ENCsbEx_NumOrd_Calc
    ; Вычисление Числа Порядка (1, 10, 100, 1000, ... 1000000.)
    MOV    R3, R11
    MOV    R4, #10
    MOVS   R5, R2          ; Прядок числа

ENCsbEx_NumOrd_Ck1
    BEQ    ENCsbEx_NumOrd_NumOrd

    MUL    R3, R3,R4      ; Число порядка
    SUBS   R5, R5,R11

    BNE    ENCsbEx_NumOrd_Ck1

ENCsbEx_NumOrd_NumOrd

    STR    R3, [R1]
    LDR    R5, [R0]

    CMP    R5, R3
    IT    LO
    MOVLO  R5, R3
    STR    R5, [R0]

ENCsbEx_NumOrd_Max
    BIC    R9, R9,#Rsos_Overflow

    ;Проерка на max
    CMP    R5, R12
    ITTTT  HI
    MOVHI  R5, R12
    STRHI  R5, [R0]
    ORRHI  R9, R9,#Rsos_Overflow
    ORRHI  R9, R9,R6

    ITT    HI
    SUBHI  R2, R2,R11
    UDIVHI R3, R3,R4
    STR    R3, [R1]

    POP    {LR}
    BX     LR

    ; ENDP

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

    ALIGNRAM 2
Determ_Ord_Num    ; PROC ;Определение Порядка Числа
    ; Вход: R2 - число
    ; Выход: R3 (Десятичный Порядок числа Rsos <-- Rsos_ENC_NumOrd)

    PUSH    {R2}
    MOV     R3, R10
    MOV     R12, #10

Determ_Ord_Num_ck1
    UDIV    R2, R2,R12
    MOVS    R2, R2
    IT     NE
    ADDNE   R3, R3,R11

```



```

BNE Determ_Ord_Num_ck1

POP    {R2}

BX LR

; ENDP

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

ALIGNRAM 2
M_Wd_W ; PROC ;Вывод маркера

PUSH    {LR}
PUSH    {R3}
AND     R2, R9,#0x0f ; Проверка на допустимый Режим

M_Wd_W_Sin
CMP     R2, #Rsos_Rej_Sin
ITT EQ
MOV32EQ R0, #Param_L1_Sin_M_0
UBFXEQ R3, R9,#ENC_NumOrd1_nBit,#3
BEQ M_Wd_W_Kn_dec

M_Wd_W_Imp
CMP     R2, #Rsos_Rej_Imp
ITT EQ
MOV32EQ R0, #Param_L1_Imp_M_0
UBFXEQ R3, R9,#ENC_NumOrd1_nBit,#3
BEQ M_Wd_W_Kn_dec

M_Wd_W_PWFM
CMP     R2, #Rsos_Rej_PWM
IT NE
CMPNE R2, #Rsos_Rej_PFM
BNE M_Wd_W_exit

TST R9, #Rsos_ENC_NumOrd_L2
ITTEE EQ
MOV32EQ R0, #Param_L1_PWFM_M_0
UBFXEQ R3, R9,#ENC_NumOrd1_nBit,#3

MOV32NE R0, #Param_L2_PWFM_M_0
UBFXNE R3, R9,#ENC_NumOrd2_nBit,#3

M_Wd_W_Kn_dec
MOV     R2, R10
ADD     R0, R0,R3,LSL#27 ; Вычисление n для Param_L1_PWFM_M_n,
Param_L2_PWFM_M_n
BL USART1_W ; Вывод на индикатор

M_Wd_W_exit
POP     {R3}
POP     {LR}
BX     LR

; ENDP

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

ALIGNRAM 2
Dir_Marker ; PROC ;Вывод "<", ">" направления смещение маркера

PUSH {LR}

```

```
AND    R2, R9, #0xf           ; Проверка на допустимый Режим
```

```
CMP     R2, #Rsos_Rej_Sin
BEQ     D_Wd_W_dir1
```

```
CMP     R2, #Rsos_Rej_Imp
BEQ     D_Wd_W_dir1
```

```
CMP     R2, #Rsos_Rej_PWM
BEQ     D_Wd_W_dir2
```

```
CMP     R2, #Rsos_Rej_PFM
BEQ     D_Wd_W_dir2
```

```
BNE     D_Wd_W_exit
```

```
D_Wd_W_dir2
```

```
MOV     R2, R10
TST     R9, #Rsos_ENC_Kn_dec2
ITE     EQ
MOV32EQ R0, #Param_L2_PWFM_M_L
MOV32NE R0, #Param_L2_PWFM_M_R
BL      USART1_W ; Вывод на индикатор
```

```
D_Wd_W_dir1
```

```
MOV     R2, R10
TST     R9, #Rsos_ENC_Kn_dec1
ITE     EQ
MOV32EQ R0, #Param_L1_PWFM_M_L
MOV32NE R0, #Param_L1_PWFM_M_R
BL      USART1_W ; Вывод на индикатор
```

```
D_Wd_W_exit
```

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

```
; ENDP
```

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

```
ALIGNRAM 2
```

```
Inter_EXTI9_5 ; PROC ;Прерывание EXTI9_5_IRQHandler (ENC2Ex: EXTI7, ENC0Ex: EXTI8, ENC0sbEx: EXTI9) от Кнопки Энкондера PA7, PB8, PB9
```

```
PUSH    {LR}
MOV32   R4, #1000
PUSH    {R4,R5}
```

```
MOV32   R0, #EXTI
LDR     R1, [R0, #EXTI_PR] ; EXTI Регистр ожидания
AND     R1, R1, #(EXTI_PR_PR5+EXTI_PR_PR6+EXTI_PR_PR7+EXTI_PR_PR8+EXTI_PR_PR9)
```

```
TST     R1, #EXTI_PR_PR7
BNE     Inter_ENC2Ex
```

```
TST     R1, #EXTI_PR_PR8
BNE     Inter_ENC0Ex
```

```
TST     R1, #EXTI_PR_PR9
ITT     NE
MOV32NE LR, #Inter_ENC0sbEx
```

```

        BXNE    LR

        STR     R1, [R0,#EXTI_PR]      ; EXTI Регистр ожидания

Inter_EXTI9_5_exit

        POP     {R4,R5}
        POP     {LR}
        BX      LR

                                ; ENDP

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

                                ALIGNRAM 2
Inter_EXTI15_10      ; PROC ;Прерывание EXTI15_10_IRQHandler (ENC3Ex: EXTI10) от Энкондера PB10

        PUSH    {LR}
        MOV32   R4, #1000
        PUSH    {R4,R5}

        MOV32   R0, #EXTI
        LDR     R1, [R0,#EXTI_PR]      ; EXTI Регистр ожидания
        AND     R1, R1,
        #(EXTI_PR_PR10+EXTI_PR_PR12+EXTI_PR_PR13+EXTI_PR_PR14+EXTI_PR_PR15)

        TST     R1, #EXTI_PR_PR10
        BNE     Inter_ENC3EX

Inter_EXTI15_10_exit

        POP     {R4,R5}
        POP     {LR}
        BX      LR

                                ; ENDP

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

                                ALIGNRAM 2
Inter_USART2      ; PROC ;Прерывание от USART

        PUSH    {LR}
        MOV32   R4, #1000
        PUSH    {R4,R5}

        MOV32   R0, #NVIC_ICPR1      ; Сбросить признак отложенного прерывания 32...64
        MOV     R1, #(1<<(USART2_IRQn-32))
        STR     R1, [R0]

        MOV32   R0,#USART2
        LDRB    R2,[R0,#USART_DR]

Inter_USART2_In_R

        AND     R3, R2,#0xf0
        CMP     R3, #0x10      ; Проверка на кнопки: 0x10...0x13 - коды Функций
        BEQ     Inter_USART2_In_R_10

        CMP     R3, #0x20      ; Проверка на кнопки: 0x20...0x28 - коды режимов.
        BEQ     Inter_USART2_In_R_20

        B       Tab_End

Inter_USART2_In_R_10 ; Выбор по кнопкам Функций: 0x10...0x13 - коды Функций
        MOV32   R0, #Tab_Kn

```

```

AND    R2, R2, #0x0f
ADD    R0, R0, R2, LSL#2
LDR    R1, [R0]

BX      R1      ; 12 - Tab_Kn_On_Off: Пуск/Стоп
           ; 13 - Tab_Kn_Res : Сброс

Inter_USART2_In_R_20      ; Выбор по кнопкам Режим: 0x20...0x25 - коды режимов.
AND    R2, R2, #0x0f
TST    R9, #Rsos_Rej_On    ; Пропуск переключение режимов, если ВКЛ
BNE    Inter_USART2_In_exit

MOV    R9, #Rsos_Rej_Begin
ORR    R9, R9, #Rsos_Update
ORR    R9, R9, R2

Inter_USART2_In_exit
B      Tab_End

; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Tab_Kn_On_Off      ALIGNRAM 2
; Пуск/Стоп
AND    R1, R9, #0xF
CMP    R1, #Rsos_Rej_Sin
BEQ    Tab_Kn_On_Off_Sin

CMP    R1, #Rsos_Rej_Imp
BEQ    Tab_Kn_On_Off_Impt

CMP    R1, #Rsos_Rej_PWM
BEQ    Tab_Kn_PWM_On_Off

CMP    R1, #Rsos_Rej_PFM
BEQ    Tab_Kn_PFM_On_Off

B      Tab_End

; -----

Tab_Kn_On_Off_Sin    ALIGNRAM 2
; Включение и Отключение выхода генератора Sin

TST    R9, #Rsos_Rej_On
ITTEE  EQ
ORREQ  R9, R9, #Rsos_Rej_On
ORREQ  R9, R9, #Rsos_OUT_On
BLNE   Off_Sin_FU
BICNE  R9, R9, #Rsos_Rej_On

B      Tab_End

; -----

Tab_Kn_On_Off_Impt    ALIGNRAM 2
; Включение и Отключение выхода генератора ИМП

TST    R9, #Rsos_Rej_On
BNE    Tab_Kn_Off_Impt

ORR    R9, R9, #Rsos_Rej_On
ORR    R9, R9, #Rsos_OUT_On
MOV32  R0, #Param_Lin0_Imp_On
BL     USART1_W      ; Вывод на индикатор

```

```

BL M_Wd_W ; Вывод маркера режима
B Tab_End

Tab_Kn_Off_Impt
    BIC R9, R9, #Rsos_Rej_On
BL Off_Impt
    MOV32 R0, #Param_Lin0_Imp_Off
BL USART1_W ; Вывод на индикатор
BL M_Wd_W ; Вывод маркера режима
B Tab_End

; -----

ALIGNRAM 2
Tab_Kn_PWM_On_Off ; Включение и Отключение выхода генератора ШИМ
Tab_Kn_PFM_On_Off ; Включение и Отключение выхода генератора ЧИМ

EOR R9, R9, #(Rsos_OUT_On+Rsos_Rej_On)

B Tab_End

; -----

Tab_Kn_
NOP
B Tab_End

; *****

Tab_End
POP {R4,R5}
POP {LR}
BX LR

; ENDP

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

ALIGNRAM 2
Contr_Rsos
PUSH {LR}
TST R9, #Rsos_Update
BEQ Contr_Rsos_Update
; Отменяет разрешение внешних прерываний 0...32
MOV32 R0, #NVIC_ICER0
MOV R1, #((1<<TIM4_IRQn)+(1<<TIM1_UP_TIM10_IRQn)+(1<<TIM1_BRK_TIM9_IRQn))
STR R1, [R0]
; Сбрасывает признак отложенного прерывания 0...32
MOV32 R0, #NVIC_ICPR0
MOV R1, #((1<<TIM4_IRQn)+(1<<TIM1_UP_TIM10_IRQn))
STR R1, [R0]
; Отменяет разрешение внешних прерываний 32...64
MOV32 R0, #NVIC_ICER1
MOV R2, #(1<<(TIM5_IRQn-32))
; Сбрасывает признак отложенного прерывания 32...64
MOV32 R0, #NVIC_ICPR1
MOV R2, #(1<<(TIM5_IRQn-32))
STR R2, [R0]

; Сброс периферии
MOV32 R0, #RCC
LDR R1, [R0,#RCC_APB2RSTR] ; RCC APB2
MOV32 R2, #(RCC_APB2RSTR_TIM1RST+RCC_APB2RSTR_TIM8RST+RCC_APB2RSTR_ADCRST)

```

```

    ORR    R1, R1, R2
    STR    R1, [R0, #RCC_APB2RSTR]
    LDR    R1, [R0, #RCC_APB2RSTR]
    BIC    R1, R1, R2
    STR    R1, [R0, #RCC_APB2RSTR]

    MOV32  R1, #(0x01ff+RCC_APB1RSTR_DACRST)      ; RCC APB1
    STR    R1, [R0, #RCC_APB1RSTR]
    STR    R10, [R0, #RCC_APB1RSTR]

; Отключение тактов TIM1, TIM8, TIM9, ADC2
    LDR    R1, [R0, #RCC_APB2ENR]
    MOV32  R2, #(RCC_APB2ENR_TIM1EN+RCC_APB2ENR_TIM8EN+RCC_APB2ENR_TIM9EN+
RCC_APB2ENR_ADC2EN)
    BIC    R1, R1, R2
    STR    R1, [R0, #RCC_APB2ENR]

; Отключение тактов TIM2, TIM4, TIM5, DAC
    LDR    R1, [R0, #RCC_APB1ENR]
    MOV32  R2, #(RCC_APB1ENR_TIM2EN+RCC_APB1ENR_TIM4EN+RCC_APB1ENR_TIM5EN+
RCC_APB1ENR_DACEN)
    BIC    R1, R1, R2
    STR    R1, [R0, #RCC_APB1ENR]

; Отключение потока: 4 DMA2
    MOV32  R2, #DMA2
    LDR    R7, [R2, #DMA_S4CR] ; Регистр конфигурации потока: 4 DMA (DMA_SxCR) (x =
0..7)
    BIC    R7, R7, #DMA_SxCR_EN      ; EN: отключение потока
    STR    R7, [R2, #DMA_S4CR]

; Отключение потока: 5 DMA2
    MOV32  R2, #DMA2
    LDR    R7, [R2, #DMA_S5CR] ; Регистр конфигурации потока: 5 DMA (DMA_SxCR) (x =
0..7)
    BIC    R7, R7, #DMA_SxCR_EN      ; EN: отключение потока
    STR    R7, [R2, #DMA_S5CR]

; Отключение тактов DMA1, DMA2
    MOV32  R0, #RCC
    LDR    R1, [R0, #RCC_AHB1ENR]
    BIC    R1, R1, #(RCC_AHB1ENR_DMA1EN+RCC_AHB1ENR_DMA2EN)
    STR    R1, [R0, #RCC_AHB1ENR]

    B Contr_Rsos_Rej
;-----
    ALIGNRAM 2
Contr_Rsos_Update
    TST    R9, #(Rsos_Up_CC+Rsos_OUT_On+Rsos_T_xSec)
    BNE Contr_Rsos_Rej_exe
    POP    {LR}
    BX     LR
;-----
    ALIGNRAM 2
Contr_Rsos_Rej_exe
    AND    R1, R9, #0x0f
    MOV32  R0, #Tab_Rej_exe
    ADD    R0, R0, R1, LSL#2
    LDR    R1, [R0]
    BX     R1
;-----
    ALIGNRAM 2
Contr_Rsos_Rej
    BIC    R9, R9, #(Rsos_Up_CC+Rsos_T_xSec+Rsos_Update)
    AND    R1, R9, #0x0f
    MOV32  R0, #Tab_Rej

```



```

ADD    R0, R0, R1,LSL#2
LDR     R1, [R0]
BX      R1

```

```

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

```

```

POP     {LR}
BX      LR

```

```

; -----

```

```

ALIGNRAM 2

```

```

Tab_Rej_Help

```

```

MOV32   R0, #Param_Help
BL  USART1_W ; Вывод на индикатор
BL  Zad_5mC
POP     {LR}
BX      LR

```

```

; -----

```

```

ALIGNRAM 2

```

```

Tab_Rej_Rezerv1

```

```

MOV32   R0, #Param_Rej_Rezerv1
BL  USART1_W ; Вывод на индикатор
BL  Zad_5mC
POP     {LR}
BX      LR

```

```

; -----

```

```

ALIGNRAM 2

```

```

Tab_Rej_Rezerv2

```

```

MOV32   R0, #Param_Rej_Rezerv2
BL  USART1_W ; Вывод на индикатор
BL  Zad_5mC
POP     {LR}
BX      LR

```

```

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

```

```

ALIGNRAM 2

```

```

Ucm_DAC_Out ; PROC ; Формирование кода числа Ucm для DAC, R2 <-- Ucm

```

```

ADD     R3, R2, #50
VMOV    S13, R3 ; Пересылка числа в сопроцессор с плавающей запятой
VCVT.F32.U32 S3, S13 ; Пробразование в число с плавающей запятой

```

```

MOV32   R0, #Ucm_DAC_4095
VLDR    S15, [R0]
VMUL.F32 S3, S15

```

```

VCVT.U32.F32 S13, S3
VMOV    R3, S13
MOV     R1, #0xffff
EOR     R3, R3, R1
MOV32   R1, #DAC
STR     R3, [R1, #DAC_DHR12R1]

```

```

BX      LR

```

```

; ENDP

```

```

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

```

```

ALIGNRAM 2

```

```

Rej_PWFM_TIM1 ; PROC ; загрузка TIM1
; R3 - загрузка Трег

```

```

; R4 - загрузка Timp
; R5 - загрузка Tdt
PUSH {LR}

TST R9, #Rsos_TIM1_UP_PFM
BNE Rej_PWFM_TIM1_Timp

MOV32 R0, #SRAM_PWFM_Tper
LDR R3, [R0] ; Tper
LSL R3, R3, #3 ; Tper * 8

Rej_PWFM_TIM1_Timp
TST R9, #Rsos_TIM1_UP_PWM
BNE Rej_PWFM_TIM1_Tdt

MOV32 R0, #SRAM_PWFM_Timp
LDR R4, [R0] ; Timp
LSL R4, R4, #3 ; Timp * 8

Rej_PWFM_TIM1_Tdt
MOV32 R0, #SRAM_PWFM_Tdt
LDR R5, [R0]
LSL R5, R5, #3 ; Tdt * 8

MOV R7, R11

Rej_PWFM_TIM1_cr1 ; Настройка в диапазон TIM_ARR
MOV R6, #64000
CMP R3, R6
BLS Rej_PWFM_TIM1_nex

LSL R7, R7, #1
ASR R3, R3, #1 ; Tper/2
ASR R4, R4, #1 ; Timp/2
ASR R5, R5, #1 ; Tdt/2
B Rej_PWFM_TIM1_cr1

Rej_PWFM_TIM1_nex

CMP R4, R3, ASR #1
IT HI ; Timp*8 > Tper*8/2
MOVHI R4, R3, ASR #1

ADD R6, R5, R4 ; Tdt*8 + Timp*8
CMP R3, R6, LSL #1
ITT LO ; Tper*8 < (Tdt*8 + Timp*8)*2
ASRL0 R4, R3, #1
SUBLO R4, R4, R5

SUB R7, R7, R11

MOV32 R0, #TIM1 ; Формирование: ШИМ
STR R7, [R0, #TIM_PSC] ; TIM предделитель
STR R3, [R0, #TIM_ARR] ; TIM регистр авто-перезагрузки

STR R4, [R0, #TIM_CCR2]
SUB R6, R3, R4 ; Tper - Trpwm
STR R6, [R0, #TIM_CCR1]

POP {LR}
BX LR

; ENDP

```

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

```

                ALIGNRAM 2
DAC_Init        ; PROC ;Инициализация DAC

MOV32 R0, #DAC
MOV     R1, #(DAC_CR_EN1+DAC_CR_EN2)
STR     R1, [R0,#DAC_CR]
MOV     R1, #2048
STR     R1, [R0,#DAC_DHR12R1]
STR     R1, [R0,#DAC_DHR12R2]

BX LR

                ; ENDP

```

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

```

                ALIGNRAM 2
Off_Sin_FU      ; PROC ;ОТКЛ выхода режима Sin

ORR     R9, R9, #Rsos_Att

; Отключение (переход на)через стек
MOV32 R1, #Tab_Rej_exe_Sin      ; Адрес возврата
BIC     R1, R1, R11             ; Устанавливается четный адрес
MOV     R0, R13                 ; Указатель стека MRS
STR     R1, [R0,#0x24]          ; Запись со смещением в стек

BX LR

                ; ENDP

```

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

```

                ALIGNRAM 2
USART1_W        ; PROC ;ПОДПРОГРАММА ВЫВОДА НА ИНДИКАТОР
; R2 - Число
; R0 - Параметр Строки или Управления (при Управлении R2 - не устанавливается)
PUSH     {R0,R1,R2,R3}

; Проверка на переполнение Буфера сопроцессора.
MOV32 R1, #GPIOC
LDR     R1, [R1,#GPIO_IDR]
TST     R1, #GPIO_IDR_IDR_13
BNE USART1_W_Exit

BIC     R2, R2, #0xf8000000

MOV32 R12, #99999999
CMP     R2, R12
IT HI
MOVHI R2, R12      ; Если R2>99999999, то R2 <- #99999999

ORR     R2, R2, R0
MOV32 R0, #USART1
; Выбор и передача Байта 0
UBFX R3, R2, #0, #8
STRB R3, [R0,#USART_DR]

USART1_W_0      ; Ожидание передачи Байта 0
LDR     R3, [R0,#USART_SR]
TST     R3, #USART_SR_TXE
BEQ USART1_W_0

```

```

; Выбор и передача Байта 1
UBFX R3, R2, #8, #8
STRB R3, [R0, #USART_DR]
USART1_W_1 ; Ожидание передачи Байта 1
LDR R3, [R0, #USART_SR]
TST R3, #USART_SR_TXE
BEQ USART1_W_1
; Выбор и передача Байта 2
UBFX R3, R2, #16, #8
STRB R3, [R0, #USART_DR]
USART1_W_2 ; Ожидание передачи Байта 2
LDR R3, [R0, #USART_SR]
TST R3, #USART_SR_TXE
BEQ USART1_W_2
; Выбор и передача Байта 3
UBFX R3, R2, #24, #8
STRB R3, [R0, #USART_DR]
USART1_W_3 ; Ожидание передачи Байта 3
LDR R3, [R0, #USART_SR]
TST R3, #USART_SR_TXE
BEQ USART1_W_3
USART1_W_Exit
POP {R0, R1, R2, R3}
BX LR

; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

ALIGNRAM 2
Zad_1mC ; PROC ;Задержка 1 мс
PUSH {R4}

MOV32 R4, #40000
Zad_1mC_ck1
NOP
SUBS R4, R4, R11
BNE Zad_1mC_ck1

POP {R4}
BX LR

; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

ALIGNRAM 2
Zad_5mC ; PROC ;Задержка 5 мс
PUSH {R4}

MOV32 R4, #200000
Zad_5mC_ck1
NOP
SUBS R4, R4, R11
BNE Zad_5mC_ck1

POP {R4}
BX LR

; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

PUBLIC Tab_May_Data

SECTION MYDATA:CODE:NOROOT

```

ALIGNRAM 2
THUMB
DATA

Tab_May_Data

Ucm_DAC_4095 DF32 40.95 ; 4095/100

;;

Rej_PWFM_Tmax DF32 4095.0 ; Tmax = 4095

Rej_PWM_Kkor DF32 1.0417

Rej_PFM_Kpaus DF32 0.1934066 ; (Timp/Tper - 1)*8/4095, Timp/Tper = 100

Rej_PFM_Kkor DF32 1.0354

;;

Tab_Kn

DCD Tab_End ; Сдвиг По вертикали

DCD Tab_End ; Сдвиг По горизонтали

DCD Tab_Kn_On_Off ; Пуск/Стор

;DCD Tab_Kn_Res ; Сброс

;;

Tab_Rej ; Инициализация

DCD Tab_Rej_Help

DCD Tab_Rej_Fr_Tt ; Режим Частотомер, Импульса и периода

DCD Tab_Rej_NTn ; Режим Таймер

DCD Tab_Rej_Sin ; Режим Генератора SIN

DCD Tab_Rej_Imp ; Режим Генератора Imp

DCD Tab_Rej_Pwfm ; Режим Генератора ШИМ

DCD Tab_Rej_Pwfm ; Режим Генератора ЧИМ

DCD Tab_Rej_Rezerv1

DCD Tab_Rej_Rezerv2

Tab_Rej_exe ; Работа

DCD Tab_Rej_Help

DCD Tab_Rej_exe_Fr_Tt ; Работа Режим Частотомер, Импульса и периода

DCD Tab_Rej_exe_NTn ; Режим Таймер

DCD Tab_Rej_exe_Sin ; Режим Генератора SIN

DCD Tab_Rej_exe_Imp ; Режим Генератора Imp

DCD Tab_Rej_exe_Pwfm ; Режим Генератора ШИМ

DCD Tab_Rej_exe_Pwfm ; Режим Генератора ЧИМ

;;

END

;;

ПО ENC0_STM32F4_IAR

; NAME ENC0_STM32F4_IAR

PUBLIC Inter_ENC0Ex, Inter_ENC0Ex_Rej_Imp_ImpT, Inter_ENC0Ex_Rej_FTt_NTn_Ucm

PUBLIC Inter_ENC0sbEx

PUBLIC Ucm_DAC_In_4095

SECTION .text:CODE:REORDER:NOROOT(1)

THUMB

ALIGNRAM 2

Inter_ENC0Ex ; Прерывание EXTI8 от Энкондера PB8

```
MOV R1, #EXTI_PR_PR8
STR R1, [R0,#EXTI_PR] ; EXTI Регистр ожидания

MOV32 R1, #GPIOB
LDR R3, [R1,#GPIO_IDR] ; Чтение из порта (PB7)

AND R2, R9, #0x0f

CMP R2, #Rsos_Rej_Imp
BEQ Inter_ENC0Ex_Rej_Imp_ImpT

CMP R2, #Rsos_Rej_Begin
BEQ Inter_ENC0Ex_Rej_Help

CMP R2, #Rsos_Rej_PWM
BEQ Inter_ENC0Ex_Rej_PWM_Mod

CMP R2, #Rsos_Rej_PFM
BEQ Inter_ENC0Ex_Rej_PWM_Mod

MOV32 R0, #Inter_ENC0Ex_Rej_FTt_NTn_Ucm
CMP R2, #Rsos_Rej_FTt
IT EQ
BXEQ R0

CMP R2, #Rsos_Rej_NTn
IT EQ
BXEQ R0

B Inter_EXTI9_5_exit
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

Inter_ENC0Ex_Rej_Imp_ImpT

MOV32 R0, #SRAM_Imp
LDR R2, [R0]
TST R3, #ENC0Dc_PB7
ITE EQ
ADDEQ R2,R11
UQSUB8NE R2, R2,R11 ; Вычитание и без знаковое насыщение (Проерка на <0)

;Проерка на max
MOV R12, #12
CMP R2, R12
IT HI
MOVHI R2, R12
STR R2, [R0]

MOV32 R0, #Param_Lin0_Imp_Imp
BL USART1_W ; (См. FTt_STM32F405RG)

TST R9, #Rsos_Rej_On
BEQ Tab_End

ORR R9, R9, #Rsos_Imp_Crutch
POP {R4,R5}

B Tab_Rej_exe_Imp
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

Inter_ENC0Ex_Rej_Help

TST R9, #Rsos_Rej_On ; Пропуск переключение режимов, если ВКЛ
```



```

BNE Inter_EXTI9_5_exit

MOV32 R0, #SRAM_Begin
LDR R2, [R0]

TST R3, #ENC0Dc_PB7
ITE EQ
ADDEQ R2, R2,R11
UQSUB8NE R2, R2,R11

; Проерка на max параметр 17
MOV32 R12, #17
CMP R2, R12
IT HI
MOVHI R2, R12
STR R2, [R0]
MOV32 R0, #Param_Help
ASR R0, R0, #27
ADD R0, R0, R2
LSL R0, R0, #27
BL USART1_W ; (См. Ftt_STM32F405RG)
BL Zad_5mC

```

```

B Inter_EXTI9_5_exit

```

```

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

```

```

Inter_ENC0Ex_Rej_PWM_Mod

```

```

TST R9, #Rsos_Rej_On ; Пропуск переключение режимов, если ВКЛ
BNE Inter_EXTI9_5_exit

MOV32 R0, #SRAM_PWFM_Mod
LDR R2, [R0]

TST R3, #ENC0Dc_PB7
ITE EQ
ADDEQ R2, R2,R11
UQSUB8NE R2, R2,R11

; Проерка на maxmax 1
MOV32 R12, #1
CMP R2, R12
IT HI
MOVHI R2, R12
STR R2, [R0]

CMP R2, #Mod_Hand
IT EQ
MOV32EQ R0, #Param_Lin0_PWFM_Mod_Hand

CMP R2, #Mod_Ext
IT EQ
MOV32EQ R0, #Param_Lin0_PWFM_Mod_Ext

CMP R2, #Mod_Sin
IT EQ
MOV32EQ R0, #Param_Lin0_PWFM_Mod_Sin
BL USART1_W ; (См. Ftt_STM32F405RG)

MOV32 R0, #SRAM_PWFM_Timp
LDR R2, [R0]
MOV32 R0, #Param_Lin1_PWFM_Timp_zpt
BL USART1_W ; Вывод на индикатор (См. Ftt_STM32F405RG)

```

BL M_Wd_W ; Вывод направления смещение маркера (См. FTt_STM32F405RG)

B Inter_EXTI9_5_exit

;;

Inter_ENC0Ex_Rej_FTt_NTn_Ucm

MOV32 R0, #Ucm_DAC_In_4095

VLDR S15, [R0]

MOV32 R0, #SRAM_FTt_Ucm

LDR R2, [R0]

TST R3, #ENC0Dc_PB7

ITE EQ

ADDEQ R2, R11

SUBNE R2, R2, R11

; Проерка на тах +-2,0 В

MOV32 R12, #20 ;2,0

CMP R2, R12

IT GT

MOVGT R2, R12

MOV32 R12, #-20 ;-2,0

CMP R2, R12

IT LT

MOVLt R2, R12

STR R2, [R0]

ADD R3, R2, #20 ;2,0

VMOV S13, R3

VCVT.F32.U32 S3, S13 ;Пересылка числа в сопроцессор с плавающей запятой

VMUL.F32 S3, S15

VCVT.U32.F32 S13, S3

VMOV R3, S13

MOV32 R1, #DAC

STR R3, [R1, #DAC_DHR12R2]

TST R2, #0x80000000 ; Проверка числа на минус

ITTE NE

SUBNE R2, R10, R2 ; Смена знака числа

MOV32NE R0, #Param_Lin0_FTt_UcmM

MOV32EQ R0, #Param_Lin0_FTt_Ucm

BL USART1_W ; (См. FTt_STM32F405RG)

B Inter_EXTI9_5_exit

;;

Inter_ENC0sbEx ; Прерывание EXTI9 от Кнопки Энкондера PB9

MOV R1, #EXTI_PR_PR9

STR R1, [R0, #EXTI_PR] ; EXTI Регистр ожидания

B Inter_EXTI9_5_exit

;;

SECTION MYDATA:CODE:NOROOT

ALIGNRAM 2

THUMB
DATA

;;

Ucm_DAC_In_4095 DF32 100.0 ;4095/40

;;

ИО ENC1_STM32F4_IAR

; NAME ENC1_STM32F4_IAR

PUBLIC Tab_Rej_Imp, DMA2_0, DMA2_1

PUBLIC Tab_Rej_exe_Imp, Tab_Rej_exe_Imp_muud, Tab_Rej_exe_Imp_OuD

PUBLIC Tab_Rej_exe_Imp_SRAM_Out, Tab_Rej_exe_Imp_clk, Tab_Rej_exe_Imp_DMA_EN

PUBLIC Off_ImpT

SECTION .text:CODE:REORDER:NOROOT(1)

THUMB

ALIGNRAM 2

Inter_ENC1sbEx ;PROC ; Прерывание EXTI3_IRQHandler от Кнопки Энкондера 1 PB3

PUSH {LR}

MOV32 R4, #1000

PUSH {R4,R5}

MOV32 R0, #EXTI

MOV R1, #EXTI_PR_PR3

STR R1, [R0,#EXTI_PR] ; EXTI Регистр ожидания

BIC R9, R9,#Rsos_ENC_NumOrd_L2

.....

AND R2, R9,#0x0f ;Проверка на допустимый Режим

CMP R2, #Rsos_Rej_Sin

BEQ Inter_ENC1sbEx_Rej_Sin_F

CMP R2, #Rsos_Rej_Imp

BEQ Inter_ENC1sbEx_Rej_Imp_Tper

CMP R2, #Rsos_Rej_PWM

BEQ Inter_ENC1sbEx_Rej_PWM_Tpwm

CMP R2, #Rsos_Rej_PFM

BEQ Inter_ENC1sbEx_Rej_PFM_Timp

B Inter_ENC1Ex_ENC1sbEx

.....

Inter_ENC1sbEx_Rej_PFM_Timp

MOV32 R0, #SRAM_PWFM_Timp

MOV32 R1, #SRAM_PWFM_Tper

LDR R3, [R1]

MOV R12, R3,ASR #1

MOV R3, #50000

CMP R3, R12

```

IT LO
MOVLO R12, R3
B Inter_ENC1sbEx_b1

```

```

;.....

```

```

Inter_ENC1sbEx_Rej_PWM_Tpwm

```

```

MOV32 R1, #SRAM_PWFM_Tper
LDR R3, [R1]
MOV R12, R3, ASR #1
MOV32 R0, #SRAM_PWFM_Timp
B Inter_ENC1sbEx_b1

```

```

;.....

```

```

Inter_ENC1sbEx_Rej_Imp_Tper

```

```

MOV32 R0, #SRAM_Imp_Tper
MOV32 R12, #9999999
B Inter_ENC1sbEx_b1

```

```

;.....

```

```

Inter_ENC1sbEx_Rej_Sin_F

```

```

MOV32 R0, #SRAM_Sin_F
MOV32 R12, #2280000

```

```

;.....

```

```

Inter_ENC1sbEx_b1

```

```

UBFX R2, R9, #ENC_NumOrd1_nBit, #3 ; Rsos_ENC_NumOrd <-- Rsos
MOV32 R1, #SRAM_NumOrd_L1
BL ENCsbEx_NumOrd ; Определение и обработка порядка числа (См. Ftt_STM32F405RG)
; Вход: R2 (Десятичный Порядок числа Rsos_ENC_NumOrd <-- Rsos), [R1]: R1 - адрес
числа Порядка, R12 - Мах число
; Выход: R5 -- > [R0], R0 - адрес числа

```

```

BFI R9, R2, #ENC_NumOrd1_nBit, #3 ; Rsos <-- Rsos_ENC_NumOrd
BL Dir_Marker ; Вывод "<", ">" направления смещение маркера
ORR R9, R9, #Rsos_ENCsbEx ; Прошло изменение порядка

```

```

B Inter_ENC1Ex_ENC1sbEx

```

```

; ENDP

```

```

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

```

```

ALIGNRAM 2

```

```

Inter_ENC1Ex ; PROC ; Прерывание EXTI4_IRQHandler от Энкондера 1 PB4

```

```

PUSH {LR}
MOV32 R4, #1000
PUSH {R4, R5}

MOV32 R0, #EXTI
MOV R1, #EXTI_PR_PR4
STR R1, [R0, #EXTI_PR] ; EXTI Регистр ожидания

```

```

Inter_ENC1Ex_ENC1sbEx

```

```

BIC R9, R9, #Rsos_ENC_NumOrd_L2

```

; Изменение задания значений Sin_F, Imp_Tper, PWM_Tpwm, PFM_Timp

AND R2, R9, #0x0f ; Проверка на допустимый Режим

```
CMP    R2, #Rsos_Rej_Sin
BEQ    Inter_ENC1Ex_Rej_Sin_F
```

```
CMP    R2, #Rsos_Rej_Imp
BEQ    Inter_ENC1Ex_Rej_Imp_Tper
```

```
CMP    R2, #Rsos_Rej_PWM
BEQ    Inter_ENC1Ex_Rej_PWM_Tpwm
```

```
CMP    R2, #Rsos_Rej_PFM
BEQ    Inter_ENC1Ex_Rej_PFM_Timp
```

B Tab_End

;

Inter_ENC1Ex_Rej_PFM_Timp

```
MOV32 R1, #SRAM_PWFM_Tper
LDR R3, [R1]
MOV R12, R3, ASR #1
MOV R3, #50000
CMP R3, R12
IT LO
MOVLO R12, R3
MOV32 R1, #SRAM_PWFM_Timp
MOV32 R0, #Param_Lin1_PWFM_Timp_zpt
```

BL ENC_Cont_F_T_t1 ; Bθoδ: R0-Param_Lin, R1 - Aδpec θ SRAM, R3 - GPIOx, R5 - ENC1Dc_Pxn, R12 - max Rsos_OUT_On

```

TST     R9, #Rsos_Rej_On
BEQ     Tab_End

```

POP {R4,R5}

```
BL Rej_PWFM_TIM1 ; загрузка TIM1
    POP    {LR}
    BX     LR
```

;

Inter_ENC1Ex_Rej_PWM_Tpwm

```
MOV32 R1, #SRAM_PWFM_Timp
MOV32 R0, #SRAM_PWFM_Tper
LDR R4, [R0]
MOV R12, R4, ASR #1
MOV32 R0, #Param_Lin1_PWFM_Timp_zpt
```

ENC1Dc Pxn, R12 - max Rsos OUT On

TST R9, #Rsos_Rej_On
BE0 Tab End

POP {R4,R5}

```
BL Rej_PWFM_TIM1 ; загрузка TIM1
    POP    {LR}
```

BX LR

;;

Inter_ENC1Ex_Rej_Imp_Tper

MOV32 R1, #SRAM_Imp_Tper
MOV32 R0, #Param_Lin1_Imp_Tper
MOV32 R12, #9999999

BL ENC_Cont_F_T_t_1 ; Вход: R0-Param_Lin, R1 - Адрес в SRAM, R3 - GPIOx, R5 - ENC1Dc_Pxn, R12 - max

TST R9, #Rsos_Rej_On
BEQ Tab_End

POP {R4,R5}

B Tab_Rej_exe_Imp

;;

Inter_ENC1Ex_Rej_Sin_F

MOV32 R1, #SRAM_Sin_F
MOV32 R0, #Param_Lin1_Sin_F
MOV32 R12, #2280000

BL ENC_Cont_F_T_t_1 ; Вход: R0-Param_Lin, R1 - Адрес в SRAM, R3 - GPIOx, R5 - ENC1Dc_Pxn, R12 - max

TST R9, #Rsos_Rej_On
IT NE
BLNE Off_Sin_FU

B Tab_End

; ENDP

;;

ALIGNRAM 2

; Контроль Sin_F, Imp_Tper, PWM_Tpwm

ENC_Cont_F_T_t_1 ;PROC
; Вход: R0-Param_Lin, R1 - Адрес Числа, R12 - max

PUSH {LR}

LDR R2, [R1]

TST R9, #Rsos_ENCsbEx ; Проверка на изменение порядка
BNE ENC_Cont_F_T_t_1_Cont

MOV32 R4, #SRAM_NumOrd_L1
LDR R4, [R4]

MOV32 R3, #GPIOB
LDR R3, [R3,#GPIO_IDR] ; Чтение из порта

;BKPT #5

MOV32 R5, #ENC1Dc_PB5
TST R3, R5 ; Проверка на знак изменения Энкодера
ITT EQ
; Увеличение
ADDEQ R2, R2,R4
BICEQ R9, R9,#Rsos_ENC_Kn_dec1

[illegible]

ПО ENC2_STM32F4_IAR

```

; NAME ENC2_STM32F4_IAR

PUBLIC Inter_ENC2Ex, Inter_ENC2Ex_Rej_PWM_Tper, Inter_ENC2Ex_Rej_Imp,
Inter_ENC2Ex_Rej_Sin
PUBLIC ENC2_Cont_Uamp
PUBLIC Inter_ENC2sbEx

SECTION .text:CODE:REORDER:NOROOT(1)
THUMB
ALIGNRAM 2

Inter_ENC2sbEx      ; PROC      ;Прерывание EXTI0_IRQHandler от Кнопки Энкондера PB0

    PUSH    {LR}

    MOV32   R4, #1000
    PUSH    {R4,R5}

    MOV32   R0, #EXTI
    MOV     R1, #EXTI_PR_PR0
    STR     R1, [R0,#EXTI_PR] ; EXTI Регистр ожидания

    ORR     R9, R9, #Rsos_ENC_NumOrd_L2

;.....

    AND     R2, R9, #0x0f      ; Проверка на допустимый Режим

    CMP     R2, #Rsos_Rej_PWM
    BEQ     Inter_ENC2sbEx_Rej_PWM_Tper

    CMP     R2, #Rsos_Rej_PFM
    BEQ     Inter_ENC2sbEx_Rej_PFM_Tpfm

    B       Inter_ENC2Ex_ENC2sbEx

;.....

Inter_ENC2sbEx_Rej_PFM_Tpfm

    MOV32   R0, #SRAM_PWFM_Tper
    MOV32   R12, #250000
    B       Inter_ENC2sbEx_b1

;.....

Inter_ENC2sbEx_Rej_PWM_Tper

    MOV32   R0, #SRAM_PWFM_Tper
    MOV32   R12, #250000

;.....

Inter_ENC2sbEx_b1

    UBFX    R2, R9, #ENC_NumOrd2_nBit, #3 ; Rsos_ENC_NumOrd <-- Rsos
    MOV32   R1, #SRAM_NumOrd_L2
    BL      ENCsbEx_NumOrd ; Определение и обработка порядка числа (См. Ftt_STM32F405RG)
                        ; Вход: R2 (Десятичный Порядок числа Rsos_ENC_NumOrd <-- Rsos), [R1]: R1 -
адрес числа Порядка, R12 - Max число
                        ; Выход: R5 -- > [R0], R0 - адрес числа

```

```

        BFI R9, R2,#ENC_NumOrd2_nBit,#3 ; Rsos <-- Rsos_ENC_NumOrd
BL Dir_Marker ; Вывод "<", ">" направления смещение маркера (См. Ftt_STM32F405RG)
        ORR R9, R9,#Rsos_ENCsbEx ; Прошло изменение порядка

```

```

        B Inter_ENC2Ex_ENC2sbEx

```

```

        ; ENDP

```

```

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

```

```

Inter_ENC2Ex ; Прерывание EXTI7 от Энкондера PA7

```

```

        MOV32 R0, #EXTI
        MOV R1, #EXTI_PR_PR7
        STR R1, [R0,#EXTI_PR] ; EXTI Регистр ожидания

```

```

Inter_ENC2Ex_ENC2sbEx

```

```

        ORR R9, R9,#Rsos_ENC_NumOrd_L2

        AND R2, R9,#0x0f ; Проверка на допустимый Режим

        CMP R2, #Rsos_Rej_Sin
        ITT EQ
        MOV32EQ R0, #Inter_ENC2Ex_Rej_Sin
        BXEQ R0

        CMP R2, #Rsos_Rej_Imp
        BEQ Inter_ENC2Ex_Rej_Imp

        CMP R2, #Rsos_Rej_PWM
        BEQ Inter_ENC2Ex_Rej_PWM_Tper

        CMP R2, #Rsos_Rej_PFM
        BNE Inter_EXTI9_5_exit

```

```

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

```

```

Inter_ENC2Ex_Rej_PFM_Tpfm

```

```

        MOV32 R1, #SRAM_PWFM_Tper
        MOV32 R12, #250000
        MOV32 R0, #Param_Lin2_PWFM_Tper_zpt
        BL ENC_Cont_F_T_t_2 ; Ввод: R0-Param_Lin, R1 - Адрес в SRAM, R12 - max
Rsos_OUT_On

```

```

        MOV32 R0, #SRAM_PWFM_Tper
        LDR R2, [R0]
        MOV32 R1, #SRAM_PWFM_Timp
        LDR R3, [R1]
        CMP R2, R3,LSL #1
        BHS Inter_ENC2Ex_Rej_PFM_Tpfm_T ; Tpfm >= Timp*2

        MOV R2, R3,LSL #1
        STR R2, [R0]
        MOV32 R0, #Param_Lin2_PWFM_Tper_zpt
        BL USART1_W ; Вывод на индикатор (См. Ftt_STM32F405RG)
        BL Dir_Marker ; Вывод направления смещение маркера (См. Ftt_STM32F405RG)
        BL M_Wd_W ; Вывод маркера (См. Ftt_STM32F405RG)

```

```

Inter_ENC2Ex_Rej_PFM_Tpfm_T

```

```

        TST R9, #Rsos_Rej_On

```

```

BEQ Tab_End

POP {R4,R5}

BL Rej_PWFM_TIM1 ; загрузка TIM1 (См. Ftt_STM32F405RG)

POP {LR}
BX LR
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
Inter_ENC2Ex_Rej_PWM_Tper

MOV32 R1, #SRAM_PWFM_Tper
MOV32 R0, #Param_Lin2_PWFM_Tper_zpt
MOV32 R12, #250000
BL ENC_Cont_F_T_t_2 ; Ввод: R0-Param_Lin, R1 - Адрес в SRAM, R12 - max

MOV32 R0, #SRAM_PWFM_Timp
LDR R2, [R0]
MOV32 R1, #SRAM_PWFM_Tper
LDR R3, [R1]
CMP R2, R3, ASR #1
BLO Inter_ENC2Ex_Rej_PWM_Tper_T ; Tpwmm < Tper/2

MOV R2, R3, ASR #1
STR R2, [R0]
MOV32 R0, #Param_Lin1_PWFM_Timp_zpt
BL USART1_W ; Вывод на индикатор (См. Ftt_STM32F405RG)
BL Dir_Marker ; Вывод "<", ">" направления смещение маркера (См. Ftt_STM32F405RG)
BL M_Wd_W ; Вывод маркера (См. Ftt_STM32F405RG)

Inter_ENC2Ex_Rej_PWM_Tper_T
TST R9, #Rsos_Rej_On
BEQ Tab_End

POP {R4,R5}

BL Rej_PWFM_TIM1 ; загрузка TIM1 (См. Ftt_STM32F405RG)

POP {LR}
BX LR
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
Inter_ENC2Ex_Rej_Imp

MOV32 R0, #SRAM_Imp_Uamp
BL ENC2_Cont_Uamp ; Контроль Uamp
MOV32 R0, #Param_Lin2_Imp_Uamp
BL USART1_W ; Вывод на индикатор (См. Ftt_STM32F405RG)
TST R9, #Rsos_Rej_On
BEQ Inter_EXTI9_5_exit

POP {R4,R5}

B Tab_Rej_exe_Imp
;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
Inter_ENC2Ex_Rej_Sin

MOV32 R0, #SRAM_Sin_Uamp
BL ENC2_Cont_Uamp ; Контроль Uamp

MOV32 R0, #Param Lin2 Sin Uamp

```

```

BL USART1_W      ; Вывод на индикатор (См. Ftt_STM32F405RG)
TST  R9, #Rsos_Rej_On
IT NE
BLNE Off_Sin_FU   ; Вкл и Отк выхода генератора Sin (См. Ftt_STM32F405RG)

B Tab_End

```

```

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

```

```

      ; Контроль Uamp
ENC2_Cont_Uamp      ; PROC ; R0 - Адрес в SRAM

MOV32 R1, #GPIOA
LDR   R3, [R1, #GPIO_IDR]      ; Чтение из порта (PA6)
LDR   R2, [R0]

TST   R3, #ENC2Dc_PA6
ITE EQ
ADDEQ R2, R2, R11
SUBNE R2, R2, R11

      ; Проерка на 0,1 В
MOVS  R2, R2
IT EQ
MOVEQ R2, R11

      ; Проерка на тахтах 5,0 В
MOV32 R12, #50
CMP   R2, R12
IT HI
MOVHI R2, R12
STR   R2, [R0]

BX LR

```

```

      ; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

      ; Контроль PWM_Tper, PFM_Tpfm

ENC_Cont_F_T_t_2    ; PROC
      ; Вход: R0-Param_Lin, R1 - Адрес Числа, R12 - тах

PUSH  {LR}

LDR   R2, [R1]

TST   R9, #Rsos_ENCsbEx      ; Проверка на изменение порядка
BNE  ENC_Cont_F_T_t_2_Cont

MOV32 R4, #SRAM_NumOrd_L2
LDR   R4, [R4]

MOV32 R3, #GPIOA
LDR   R3, [R3, #GPIO_IDR]    ; Чтение из порта

MOV32 R5, #ENC2Dc_PA6
TST   R3, R5      ; Проверка на знак изменения Энкодера
ITT EQ
      ; Увеличение
ADDEQ R2, R2, R4
BICEQ R9, R9, #Rsos_ENC_Kn_dec2
BEQ  ENC_Cont_F_T_t_2_Cont

```

[illegible]

ПО ENC3_STM32F4_IAR

```

; NAME ENC3_STM32F4_IAR

PUBLIC Inter_ENC3EX, Inter_ENC3EX_Rej_PWFM_T_Tdt, Inter_ENC3Ex_Rej_Imp,
Inter_ENC3Ex_Rej_Sin
PUBLIC ENC3Ex_Rej_Sin_Imp_Ucm
PUBLIC Inter_ENC3sbEx

SECTION .text:CODE:REORDER:NOROOT(1)
THUMB
ALIGNRAM 2

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

Inter_ENC3EX ; Прерывание EXTI10 от Энкондера PB10

MOV    R1, #EXTI_PR_PR10
STR    R1, [R0,#EXTI_PR] ; EXTI Регистр ожидания

MOV32  R1, #GPIOB
LDR    R3, [R1,#GPIO_IDR] ;Чтение из порта (PB11)

AND    R2, R9,#0x0f ; Проверка на допустимый Режим

CMP    R2, #Rsos_Rej_Sin
BEQ    Inter_ENC3Ex_Rej_Sin

CMP    R2, #Rsos_Rej_Imp
BEQ    Inter_ENC3Ex_Rej_Imp

CMP    R2, #Rsos_Rej_PWM
BEQ    Inter_ENC3EX_Rej_PWFM_T_Tdt

CMP    R2, #Rsos_Rej_PFM
BNE    Inter_EXTI15_10_exit

;

Inter_ENC3EX_Rej_PWFM_T_Tdt

MOV32  R0, #SRAM_PWFM_Tdt
LDR    R2, [R0]

TST    R3, #ENC3Dc_PB11
ITE EQ
ADDEQ  R2, R2,R11
SUBNE  R2, R2,R11

; Проерка на 0
MOVS   R2, R2
IT EQ
MOVEQ  R2, R11

; Проерка на max
MOV32  R12, #1000
CMP    R2, R12
IT HI
MOVHI  R2, R12
STR    R2, [R0]

MOV32  R0, #Param_Lin3_PWFM_Tdt_zpt
BL  USART1_W ; (См. Ftt_STM32F405RG)

```

POP {LR}

```

BX LR
; ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Inter_ENC3sbEx ; PROC ;Прерывание EXTI1_IRQHandler от Кнопки Энкондера PB1

    PUSH    {LR}

    MOV32   R0, #EXTI
    MOV     R1, #EXTI_PR_PR1
    STR     R1, [R0,#EXTI_PR] ; EXTI Регистр ожидания

    POP     {LR}
    BX LR

; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

SECTION MYDATA:CODE:NOROOT
ALIGNRAM 2
THUMB
DATA

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

```

ПО Режим Частотомер F, Измерение Периода T, длительности Импульса t

```

;NAME Rej_Fr_T_t_STM32F4_IAR

PUBLIC Tab_Rej_Fr_Tt, Tab_Rej_exe_Fr_Tt

SECTION .text:CODE:REORDER:NOROOT(1)
THUMB
ALIGNRAM 2

;Инициализация: Режим Частотомер F, Измерение Периода T и длительности Импульса t

Tab_Rej_Fr_Tt

    MOV32   R0, #RCC

    LDR     R1, [R0,#RCC_APB2ENR]
    MOV32   R2, #(RCC_APB2ENR_TIM1EN+RCC_APB2ENR_TIM8EN) ;включение тактов TIM1, TIM8
    ORR     R1, R1, R2
    STR     R1, [R0,#RCC_APB2ENR]

    LDR     R1, [R0,#RCC_APB1ENR]
    MOV32   R2, #(RCC_APB1ENR_TIM2EN+RCC_APB1ENR_TIM4EN+RCC_APB1ENR_DACEN) ;включение
тактов TIM2, TIM4, DAC
    ORR     R1, R1, R2
    STR     R1, [R0,#RCC_APB1ENR]

    MOV32   R1, #GPIOA
    LDR     R2, [R1,#GPIO_MODER]
    BIC     R2, R2,#GPIO_MODER_0_AF
    MOV32   R3, #(GPIO_MODER_12_AF+GPIO_MODER_15_AF)
    ORR     R2, R2, R3
    STR     R2, [R1,#GPIO_MODER] ; Регистр режима порта GPIO (GPIOx_MODER)
    MOV32   R2, #(GPIOA_AFRH_PA12_TIM1_ETR+GPIOA_AFRH_PA15_TIM2_CH1_ETR)
    STR     R2, [R1,#GPIO_AFRH] ; GPIO Н регистр альтернативных функций

```

```

BL DAC_Init
MOV32 R0, #Param_Rej_Fr_Tt
BL USART1_W
BL Zad_5mC

```

```

;.....

```

```

MOV32 R1, #TIM8
STR R10, [R1,#TIM_PSC] ; TIM предделитель
STR R10, [R1,#TIM_CNT] ; TIM регистр счетчика
MOV R2, #0xFFFF
STR R2, [R1,#TIM_ARR] ; TIM регистр авто-перезагрузки
MOV R2, #(TIM_SMCR_TS_ITR0+TIM_SMCR_SMS_STR) ;TIM_SMCR_TS_ITR0[000]-
внутренний запуск 0 (ITR0) TIM1_TRGO, TIM_SMCR_SMS_STRS[101]-Стробируемый режим
STR R2, [R1,#TIM_SMCR] ; TIM регистр управления подчиненным режимом

STR R10, [R1,#TIM_CNT] ; TIM регистр счетчика
MOV R2, #TIM_CR1_CEN ; Счетчик включить
STR R2, [R1,#TIM_CR1] ; TIM управляющий регистр 1

```

```

;.....

```

```

MOV32 R1, #TIM1
STR R10, [R1,#TIM_CNT] ; TIM регистр счетчика
MOV R2, #0xFFFF
STR R2, [R1,#TIM_ARR] ; TIM регистр авто-перезагрузки
MOV R2, #TIM_CR2_MMS_Upd
STR R2, [R1,#TIM_CR2]
MOV R2, #(TIM_SMCR_ECE+TIM_SMCR_TS_ETRF+TIM_SMCR_SMS_STR) ;
TIM_SMCR_ECE[1]-Включение внешних тактов,
;TIM_SMCR_TS_ETRF[111]-Вход внешнего запуска (ETRF), TIM_SMCR_SMS_STRS[101]-Стробируемый
режим
STR R2, [R1,#TIM_SMCR] ; TIM регистр управления подчиненным режимом

STR R10,[R1,#TIM_CNT] ; TIM регистр счетчика
MOV R2, #TIM_CR1_CEN ; Счетчик включить
STR R2, [R1,#TIM_CR1] ; TIM управляющий регистр 1

```

```

;.....

```

```

MOV32 R0, #TIM4 ; Формирование: 1 секунда
MOV R2, #1599
STR R2, [R0,#TIM_PSC] ; TIM предделитель
MOV R2, #50000
STR R2, [R0,#TIM_ARR] ; TIM регистр авто-перезагрузки
STR R10, [R0,#TIM_CCR1] ; TIM capture/compare регистр 1
MOV R2, #TIM_CR1_OPM ;Одно импульсный режим
STR R2, [R0,#TIM_CR1] ; TIM управляющий регистр 1
MOV R2, #TIM_CR2_MMS_Enb
STR R2, [R0,#TIM_CR2] ; TIM управляющий регистр 2

MOV R2, #TIM_DIER_UIE ; Включить прерывание от обновления
STR R2, [R0,#TIM_DIER] ; TIM DMA /прерывания регистр разрешения
LDR R2, [R0,#TIM_CR1]
ORR R2, R2,#TIM_CR1_CEN ; 1 - Счетчик включить
STR R2, [R0,#TIM_CR1] ; TIM управляющий регистр 1

MOV32 R0, #NVIC_ISER0 ; Разрешает внешние прерывания 0...32
MOV R2, #(1<<TIM4_IRQn)
STR R2, [R0]

```

```

;.....

```

```
MOV32 R0, #TIM2 ; Измерение Периода T и длительности Импульса t

MOV R1, #7
STR R1, [R0,#TIM_PSC] ; TIM предделитель
MOV R1, #(TIM_CCMR1_CC2S_TI1+TIM_CCMR1_CC1S_TI1) ; TIM_CCMR1_CC2S[010]-канал
CC2 настроен как вход, IC2 отображается на TI1
;TIM_CCMR1_CC1S[001]-канал CC1 настроен как вход, IC1 отображается на TI1.
STR R1, [R0,#TIM_CCMR1] ; TIM регистр режима захвата/сравнения 1
MOV R1, #(TIM_CCER_CC2P+TIM_CCER_CC2E+TIM_CCER_CC1E) ; Захват/Сравнение 1
вход включить,
; TIM_CCER_CC2P[1],TIM_CCER_CC1P[1]-Захват/Сравнение 2, 1 полярности выхода
STR R1, [R0,#TIM_CCER] ; TIM захват/сравнение включить регистр
MOV R1, #(TIM_SMCR_MSM+TIM_SMCR_TS_TI1FP1+TIM_SMCR_SMS_RST) ;
TIM_SMCR_MSM[1]-Режим ведущий/ведомый,
; TIM_SMCR_TS[101] - Фильтрованный вход таймера 1 (TI1FP1),
; TIM_SMCR_SMS_RST[100]-Режим сброса
STR R1, [R0,#TIM_SMCR] ; TIM регистр управления подчиненным режимом

MOV R1, #TIM_DIER_CC1IE ; Capture/Compare 1 Включить прерывание
STR R1, [R0,#TIM_DIER] ; TIM DMA /прерывания регистр разрешения
MOV R2, #TIM_CR1_CEN ; Счетчик включить
STR R2, [R0,#TIM_CR1] ; TIM управляющий регистр 1

MOV32 R0, #NVIC_ISER0 ; Разрешает внешние прерывания 32...64
MOV R2, #(1<<TIM2_IRQn)
STR R2, [R0]

;.....
MOV32 R1,#SRAM_Tper
STR R10, [R1]
STR R10, [R1,#4]

MOV32 R1, #SRAM_FTt_Ucm
STR R10, [R1]

POP {LR}
BX LR

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

;Работа Режим Частотомер, Режим Импульса и периода

Tab_Rej_exe_Fr_Tt
BIC R9, R9, #Rsos_T_xSec
MOV32 R1, #SRAM_Frec
LDR R2, [R1]
MOV32 R5, #4294881399 ;Корректировочный коэффициент
UMULL R5, R3,R5,R2 ;[R3:R5] <-- R5*R2 Умножение без знака - 64-бит результат
MOV R2, R3

MOV32 R0, #Param_Lin1_Fr
BL USART1_W

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

BIC R9, R9, #Rsos_Up_CC
MOV32 R1,#SRAM_Tper
LDR R2, [R1]
ADD R2, R2, #1 ;Корректировка
MOV32 R0, #Param_Lin2_Tt
BL USART1_W
LDR R2, [R1,#4] ;SRAM_Timp
ADD R2, R2, #1 ;Корректировка
```

```

MOV32 R0, #Param_Lin3_Tt
BL USART1_W
BL Zad_5mC
MOV32 R0, #TIM2
MOV R1, #TIM_DIER_CC1IE ;Capture/Compare 1 Включить прерывание
STR R1, [R0,#TIM_DIER] ; TIM DMA /прерывания регистр разрешения

POP {LR}
BX LR

```

```

;.....

```

ПО Режим Измерение количества Импульсов(N), Времени(T) и оборотов(n)

```

;NAME Rej_NTnt_STM32F4_IAR

```

```

PUBLIC Tab_Rej_NTn, Tab_Rej_exe_NTn

```

```

SECTION .text:CODE:REORDER:NOROOT(1)
THUMB
ALIGNRAM 2

```

```

Tab_Rej_NTn ;Инициализация Режим 1: Измерение количества Импульсов(N), Времени(T) и
оборотов(n)

```

```

MOV32 R0, #RCC
LDR R1, [R0,#RCC_APB2ENR]
MOV32 R2, #(RCC_APB2ENR_TIM1EN+RCC_APB2ENR_TIM8EN+RCC_APB2ENR_TIM9EN) ;
включение тактов TIM1, TIM8
ORR R1, R1, R2
STR R1, [R0,#RCC_APB2ENR]

```

```

LDR R1, [R0,#RCC_APB1ENR]
MOV32 R2, #(RCC_APB1ENR_TIM4EN+RCC_APB1ENR_TIM5EN+RCC_APB1ENR_DACEN) ;
включение тактов TIM2, TIM4, TIM5, DAC
ORR R1, R1, R2
STR R1, [R0,#RCC_APB1ENR]

```

```

MOV32 R1, #GPIOA
LDR R2, [R1,#GPIO_MODER]
BIC R2, R2, #GPIO_MODER_15_AF
MOV32 R3, #(GPIO_MODER_0_AF+GPIO_MODER_12_AF)
ORR R2, R2, R3
STR R2, [R1,#GPIO_MODER]

```

```

LDR R2, [R1,#GPIO_AFRL]
MOV R3, #(GPIOA_AFRL_PA0_TIM5_CH1+GPIOA_AFRL_PA2_TIM9_CH1+
GPIOA_AFRL_PA3_USART2_RX)
ORR R2, R2, R3
STR R2, [R1,#GPIO_AFRL] ; GPIO L регистр альтернативных функций

```

```

MOV32 R2, #GPIOA_AFRH_PA12_TIM1_ETR
STR R2, [R1,#GPIO_AFRH] ; GPIO H регистр альтернативных функций

```

```

BL DAC_Init

```

```

;.....

```

```

MOV32 R0, #Param_Rej_NTn
BL USART1_W
BL Zad_5mC

```

```

;Измерение количества Импульсов(N)

```

```

MOV32 R1, #TIM8
STR R10, [R1,#TIM_PSC] ; TIM предделитель
STR R10, [R1,#TIM_CNT] ; TIM регистр счетчика
MOV R2, #0xFFFF
STR R2, [R1,#TIM_ARR] ; TIM регистр авто-перезагрузки
MOV R2, #(TIM_SMCR_TS_ITR0+TIM_SMCR_SMS_SYN) ;TIM_SMCR_TS_ITR0[000]-
внутренний запуск 0 (ITR3) TIM4_TRGO, TIM_SMCR_SMS_STRS[101]-Стробируемый режим
STR R2, [R1,#TIM_SMCR] ; TIM регистр управления подчиненным режимом
STR R10, [R1,#TIM_CNT] ; TIM регистр счетчика
MOV R2, #TIM_CR1_CEN ;1 - Счетчик включить
STR R2, [R1,#TIM_CR1] ; TIM управляющий регистр 1

```

```

MOV32 R1, #TIM1
STR R10, [R1,#TIM_CNT] ; TIM регистр счетчика
MOV R2, #0xFFFF
STR R2, [R1,#TIM_ARR] ; TIM регистр авто-перезагрузки
MOV R2, #TIM_CR2_MMS_Upd
STR R2, [R1,#TIM_CR2]
STR R10, [R1,#TIM_SMCR]
MOV R2, #(TIM_SMCR_ECE) ;TIM_SMCR_ECE[1]-Включение внешних тактов,
STR R2, [R1,#TIM_SMCR] ; TIM регистр управления подчиненным режимом
MOV R2, #TIM_CR1_CEN ;1 - Счетчик включить
STR R2, [R1,#TIM_CR1] ; TIM управляющий регистр 1

```

.....

; Тактировка вывода

```

MOV32 R0, #TIM4 ; Формирование: 0,02 секунда
MOV R2, #7999
STR R2, [R0,#TIM_PSC] ; TIM предделитель
MOV R2, #2000
STR R2, [R0,#TIM_ARR] ; TIM регистр авто-перезагрузки
STR R10, [R0,#TIM_CCR1] ; TIM capture/compare регистр 1
MOV R2, #TIM_CR2_MMS_Enb
STR R2, [R0,#TIM_CR2] ; TIM управляющий регистр 2
MOV R2, #TIM_DIER_UIE ; Включить прерывание от обновления
STR R2, [R0,#TIM_DIER] ; TIM DMA /прерывания регистр разрешения
LDR R2, [R0,#TIM_CR1]
ORR R2, R2, #TIM_CR1_CEN ; Счетчик включить
STR R2, [R0,#TIM_CR1] ; TIM управляющий регистр 1

```

```

MOV32 R0, #NVIC_ISER0 ; Разрешает внешние прерывания 0...32
MOV R2, #(1<<TIM4_IRQn)
STR R2, [R0]

```

.....

; Измерение Периода оборотов(n)

```

MOV32 R0, #TIM5
MOV R1, #79
STR R1, [R0,#TIM_PSC] ; TIM предделитель
MOV R1, #TIM_CCMR1_CC1S_TI1 ; TIM_CCMR1_CC1S[001]-канал CC1 настроен как
вход, IC1 отображается на TI1.
STR R1, [R0,#TIM_CCMR1]
MOV R1, #(TIM_CCER_CC1E)
STR R1, [R0,#TIM_CCER]
MOV R1, #(TIM_SMCR_TS_TI1FP1+TIM_SMCR_SMS_RST) ; TIM_SMCR_TS[101] - вход
таймера 1 (TI1FP1), TIM_SMCR_SMS_RST[100]-Режим сброса
STR R1, [R0,#TIM_SMCR]
MOV R1, #TIM_DIER_CC1IE ; Capture/Compare 1 Включить прерывание
STR R1, [R0,#TIM_DIER]
MOV R2, #TIM_CR1_CEN ; Счетчик включить
STR R2, [R0,#TIM_CR1]

```



```

MOV32 R0, #NVIC_ISER1          ; Разрешает внешние прерывания 32...64
MOV   R2, #(1<<(TIM5_IRQn-32))
STR   R2, [R0]

MOV32 R1, #SRAM_Tper
STR   R10, [R1]
STR   R10, [R1,#4]

MOV32 R1, #SRAM_FTt_Ucm
STR   R10, [R1]

```

;

;Измерение ВРЕМЕНИ

```

MOV32 R0, #TIM9
MOV   R2, #159
STR   R2, [R0,#TIM_PSC]      ; TIM предделитель
MOV32 R2, #0xFFFF
STR   R2, [R1,#TIM_ARR]      ; TIM регистр авто-перезагрузки
MOV   R1, #TIM_DIER_UIE      ; Capture/Compare 1 Включить прерывание
STR   R1, [R0,#TIM_DIER]     ; TIM DMA /прерывания регистр разрешения

```

```

MOV32 R0, #NVIC_ISER0          ; Разрешает внешние прерывания 0...32
MOV   R2, #(1<<TIM1_BRK_TIM9_IRQn)
STR   R2, [R0]

```

```

POP   {LR}
BX    LR

```

;

;Режим Измерение количества Импульсов(N), Времени(T) и оборотов(n): Работа

Tab_Rej_exe_NTn

```

BIC   R9, R9, #Rsos_T_xSec

```

;Вывод количества Импульсов(N)

```

MOV32 R1, #SRAM_NTn_N
LDR   R2, [R1]
MOV32 R0, #Param_Lin1_N
BL   USART1_W

```

;Вывод Времени(T)

```

MOV32 R1, #SRAM_NTn_Tim
LDR   R2, [R1]
MOV32 R0, #Param_Lin2_Tim
BL   USART1_W

```

;Вывод оборотов(n)

```

MOV32 R1, #SRAM_NTn_n0
LDR   R2, [R1]
MOV   R3, #600
CMP   R2, R3
IT   LO
MOVLO R2, R3
NOP
MOV32 R3, #600000000
UDIV  R2, R3, R2
MOV32 R3, #999999
CMP   R2, R3
IT   HI
MOVHI R2, R3

MOV32 R0, #Param_Lin3_n

```

```
BL USART1_W
```

```
MOV32 R0, #TIM5
MOV R1, #TIM_DIER_CC1IE ;Capture/Compare 1 Включить прерывание
STR R1, [R0,#TIM_DIER]

POP {LR}
BX LR
```

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

ПО Режим Генератора SIN

```
; NAME Rej_Sin_STM32F4_IAR
```

```
PUBLIC Tab_Rej_Sin, Tab_Rej_exe_Sin, Tab_Rej_exe_Sin_exit, Tab_Rej_exe_Sin_On
PUBLIC Tab_Rej_exe_Sin_Off, Tab_Rej_exe_Sin_Sin_On, Tab_Rej_Sin_ck1,
Tab_Rej_exe_Sin_Out
```

```
SECTION .text:CODE:REORDER:NOROOT(1)
THUMB
ALIGNRAM 2
```

```
; Инициализация Режим: Генератора SIN
```

```
Tab_Rej_Sin
```

```
MOV32 R0, #RCC
LDR R1, [R0,#RCC_APB1ENR]
ORR R1, R1, #(RCC_APB1ENR_DACEN) ; Включение тактов DAC
STR R1, [R0,#RCC_APB1ENR]
```

```
BL DAC_Init
```

```
MOV32 R0, #SRAM_NumOrd_L1
STR R11, [R0]
```

```
MOV32 R0, #Param_Rej_Sin
```

```
BL USART1_W
```

```
BL Zad_5mC
```

```
BL Zad_5mC
```

```
MOV32 R0, #(Rsos_Rej_On+Rsos_OUT_On)
BIC R9, R9, R0
```

```
MOV32 R0, #SRAM_Sin_Uamp
MOV R2, #25
STR R2, [R0]
MOV32 R0, #Param_Lin2_Sin_Uamp
```

```
BL USART1_W
```

```
MOV32 R0, #SRAM_Sin_Ucm
MOV R2, #0
STR R2, [R0]
MOV32 R0, #Param_Lin3_Sin_Ucm
```

```
BL USART1_W
```

```
BL Dir_Marker
```

```
BL M_Wd_W ; Вывод маркера режима _Sin_F
```

```
MOV32 R0, #SRAM_Sin_F
MOV R2, #1000
```

```

        STR    R2, [R0]
        MOV32  R0, #Param_Lin1_Sin_F
BL USART1_W

        MOV    R2, #0
BL Ucm_DAC_Out
        ORR    R9, R9, #Rsos_Up_CC

```

```

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

```

```

; Режим Генератора SIN: Работа

```

```

Tab_Rej_exe_Sin

```

```

        MOV32  R1, #GPIOC
        MOV    R2, #2048
        STR    R2, [R1,#GPIO_ODR] ; Регистр выходных данных порта GPIO

        TST    R9, #Rsos_OUT_On
        BNE    Tab_Rej_exe_Sin_On

```

```

Tab_Rej_exe_Sin_exit

```

```

        BIC    R9, R9, #Rsos_OUT_On
        POP    {LR}
        BX     LR

```

```

;.....

```

```

Tab_Rej_exe_Sin_On

```

```

        MOV32  R0, #SRAM_Sin_F
        LDR    R2, [R0]
        MOV32  R0, #Param_Lin1_Sin_F
BL USART1_W
BL M_Wd_W ; Вывод маркера режима _Sin_F

```

```

        TST    R9, #Rsos_Rej_On
        BNE    Tab_Rej_exe_Sin_Sin_On

```

```

        TST    R9, #Rsos_Att ; Признак изменения Задания Режима
        BEQ    Tab_Rej_exe_Sin_Off

```

```

        BIC    R9, R9, #Rsos_Att
        B      Tab_Rej_exe_Sin

```

```

Tab_Rej_exe_Sin_Off

```

```

        MOV32  R0, #Param_Lin0_Sin_Off
BL USART1_W
        B      Tab_Rej_exe_Sin_exit

```

```

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

```

```

Tab_Rej_exe_Sin_Sin_On

```

```

        MOV    R2, #8196 ; Объем памяти для sin(pi/2)
        VMOV   S12, R2 ; Пересылка числа в сопроцессор с плавающей запятой
(8196)
        VCVT.F32.U32 S4, S12 ; Пробразование в число с плавающей запятой (8196)

```

```

        MOV32  R0, #SRAM_Out
        MOV    R1, R2, LSL#2 ; R1 <-- Fk*4
        ADD    R3, R0, R1 ; R3 <-- #SRAM_Out + Fk*4
        MOV    R1, R2, LSL#2 ; R1 <-- Fk*4
        ADD    R5, R0, R1 ; R3 <-- #SRAM_Out + Fk*4
        MOV    R1, R2, LSL#3 ; R1 <-- Fk*8

```

```

ADD    R7, R0, R1          ; R3 <-- #SRAM_Out + Fk*8

PUSH    {R0}
MOV32   R0, #Rej_Sin_13f
VLDLDR  S3,[R0]             ; 1/3!
VLDLDR  S5,[R0,#4]          ; 1/5!
VLDLDR  S7,[R0,#8]          ; 1/7!
VLDLDR  S9,[R0,#12]         ; 1/9!
VLDLDR  S10,[R0,#16]        ; pi/(2*8196)
VLDLDR  S13,[R0,#20]        ; 50.0
VLDLDR  S16,[R0,#24]        ; 2035.0
POP      {R0}

; Определение коэффициента для изменения выходного напряжения
MOV32   R1, #SRAM_Sin_Uamp
LDR     R1,[R1]
VMOV    S14, R1
VCVT.F32.U32 S14, S14
VDIV.F32 S14, S13          ; Kout

MOV     R1, #0             ; Счетчик точек вычислений 0...8196
MOV     R2, #8196          ; Объем памяти для sin(pi/2)
ADD     R2, R2, R1         ; ?????

; sin(x) = (x - (x^3)/3! + (x^5)/5! - (x^7)/7! + (x^9)/9!)*Kout
Tab_Rej_Sin_ck1
ALIGNRAM 2
PUSH    {R2}
VMOV    S12, R1            ; Пересылка числа в сопроцессор с плавающей запятой
VCVT.F32.U32 S4, S12
VMUL.F32 S0, S10, S4
VMUL.F32 S8, S0, S0
VMOV.F32 S1, S0
VMUL.F32 S6, S8, S0
VFMS.F32 S1, S6, S3
VMUL.F32 S6, S6, S8
VFMA.F32 S1, S6, S5        ; накопление = накопление + умножение
VMUL.F32 S6, S6, S8
VFMS.F32 S1, S6, S7        ; накопление = накопление - умножение
VMUL.F32 S6, S6, S8
VMLA.F32 S1, S6, S9
VMUL.F32 S11, S16, S1
VMUL.F32 S11, S11, S14
VCVT.U32.F32 S2, S11
VMOV    R2, S2

ADD     R2, R2, #12        ; Коррекция ступеньки

ADDW    R2, R2, #2047
STRH    R2, [R0], #2
STRH    R2, [R3], #-2
MOVW    R4, #4095
SUB     R4, R4, R2
STRH    R4, [R5], #2
STRH    R4, [R7], #-2
POP     {R2}
ADD     R1, R1, R11
CMP     R1, R2
BNE     Tab_Rej_Sin_ck1

; .....

MOV32   R0, #Param_Lin0_Sin_On ; Вывод на индикатор символа ВКЛ
BL      USART1_w

```

```
BL M_Wd_W           ;Вывод маркера режима _Sin_F  
MOV32 R0, #SRAM_Sin_F  
LDR   R2, [R0]  
MOVS  R5, R2  
IT EQ  
MOVEQ R5, R11      ;Если в R5: 0, записать в R5: 1  
STR   R5, [R1]  
  
PUSH {R5}  
  
MOV   R2, R5  
MOV32 R0, #Param_Lin1_Sin_F  
BL USART1_W  
BL M_Wd_W          ;Вывод мигающего маркера режима  
POP   {R5}  
  
VMOV S5, R5         ; Пересылка числа в сопроцессор с плавающей запятой  
VCVT.F32.U32 S5, S5    ; Преобразование в число с плавающей запятой  
MOV32 R0, #Rej_Sin_K  
VLDR S6, [R0]  
VMUL.F32 S5, S6  
VCVT.U32.F32 S4, S5  
VMOV R4, S4  
  
MOV32 R0, #SRAM_Out  
MOV32 R1, #GPIOC  
MOV R5, R10  
MOV R3, R10  
  
;,,,,,,,,,,,,,,,,,,,,,  
  
ALIGNRAM 2  
  
Tab_Rej_exe_Sin_Out     ; Вывод Sin, Цикл: 7 тактов  
LDRH R2, [R0,R3,LSL#1]  
STR R2, [R1,#GPIO_ODR] ; Регистр выходных данных порта GPIO  
ADD R5, R4  
MOV R3, R5, LSL#17  
B Tab_Rej_exe_Sin_Out  
  
;,,,,,,,,,,,,,,,,,,,,,  
  
SECTION MYDATA:CODE:NOROOT  
ALIGNRAM 2  
THUMB  
DATA  
  
Rej_Sin_13f DF32 0.166666666666666667 ; 1/3!  
Rej_Sin_15f DF32 0.008333333333333333 ; 1/5!  
Rej_Sin_17f DF32 0.0001984126984127 ; 1/7!  
Rej_Sin_19f DF32 0.0000027557319224 ; 1/9!  
Rej_Sin_pi_D DF32 0.00019165401742251 ; pi/(2*8196) (pi/(2*90град.))  
Rej_Sin_50 DF32 50.0  
Rej_Sin_2035 DF32 2035.0  
Rej_Sin_K DF32 187.95 ; (2^32)*7.0/160000000.0  
  
;,,,,,,,,,,,,,,,,,,,,,
```

ПО Режим Генератор Функциональный

```
;NAME Rej_Imp_STM32F4_IAR

PUBLIC Tab_Rej_Imp, DMA2_0, DMA2_1
PUBLIC Tab_Rej_exe_Imp, Tab_Rej_exe_Imp_muud, Tab_Rej_exe_Imp_OuD
PUBLIC Tab_Rej_exe_Imp_SRAM_Out, Tab_Rej_exe_Imp_clk, Tab_Rej_exe_Imp_DMA_EN
PUBLIC Off_ImpI
PUBLIC Rej_Imp_VUnit, Rej_Imp_VNoll, Rej_Imp_Umax, Rej_Imp_Kkor,Rej_Imp_Tmax,
Rej_Imp_Noll
PUBLIC Tab_IMP, Tab_IMP_0, Tab_IMP_1, Tab_IMP_2, Tab_IMP_3, Tab_IMP_4, Tab_IMP_5,
Tab_IMP_6
PUBLIC Tab_IMP_7, Tab_IMP_8, Tab_IMP_9, Tab_IMP_10, Tab_IMP_11, Tab_IMP_12

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

SECTION .text:CODE:REORDER:NOROOT(1)
THUMB
ALIGNRAM 2

;Инициализация Режим: Генератор Функциональный

Tab_Rej_Imp      ; Режим Генератора Imp

MOV32 R0, #RCC
LDR   R1, [R0,#RCC_APB1ENR]
ORR   R1, R1,#RCC_APB1ENR_DACEN           ; Включение тактов DAC
STR   R1, [R0,#RCC_APB1ENR]

LDR   R1, [R0,#RCC_APB2ENR]
MOV32 R2, #RCC_APB2ENR_TIM1EN           ; включение тактов TIM1
ORR   R1, R1, R2
STR   R1, [R0,#RCC_APB2ENR]

MOV32 R0, #RCC
LDR   R1, [R0,#RCC_AHB1ENR]
ORR   R1, R1, #RCC_AHB1ENR_DMA2EN       ; включение тактов DMA2
STR   R1, [R0,#RCC_AHB1ENR]

BL DAC_Init

MOV32 R0, #SRAM_NumOrd_L1
STR   R11, [R0]

MOV32 R0, #Param_Rej_Imp
BL USART1_W
BL Zad_5mC

; Инициализация TIM1
MOV32 R2, #TIM1
STR   R10, [R2,#TIM_PSC] ; TIM регистр предделителя
MOV   R4, #7
STR   R4, [R2,#TIM_ARR] ; TIM регистр авто-перезагрузки
MOV   R4, #(TIM_DIER_UDE); Включить запрос запуск DMA. Обновление - разрешение
запроса
STR   R4, [R2,#TIM_DIER] ; TIM DMA /прерывания регистр разрешения
MOV   R4, #(TIM_CR2_MMS_Upd) ;Обновление - событие обновления выбирается как
выход запуска (TRGO).
STR   R4, [R2,#TIM_CR2] ; TIM управляющий регистр 2

MOV32 R0, #(Rsos_Rej_On+Rsos_OUT_On)
BIC   R9, R9,R0
```

```

MOV32 R0, #SRAM_Imp_Uamp
MOV    R2, #25
STR    R2, [R0]
MOV32  R0, #Param_Lin2_Imp_Uamp
BL     USART1_W
MOV32  R0, #SRAM_Imp_Ucm
MOV    R2, #0
STR    R2, [R0]
MOV32  R0, #Param_Lin3_Imp_Ucm
BL     USART1_W
MOV    R2, #0
BL     Ucm_DAC_Out

```

```

MOV32 R0, #SRAM_Imp_Tper
MOV32  R2, #7500
STR    R2, [R0]
MOV32  R0, #Param_Lin1_Imp_Tper
BL     USART1_W

```

```

MOV32 R0, #SRAM_Imp
MOV    R2, R10
STR    R2, [R0]
MOV32  R0, #Param_Lin0_Imp_Imp
BL     USART1_W
BL     M_Wd_W      ; Вывод маркера режима

```

```

MOV32 R1, #GPIOC
MOV    R2, #GPIO_ODR_ODR_11
STR    R2, [R1,#GPIO_ODR]      ; Регистр выходных данных порта GPIO

BIC    R9, R9,#Rsos_OUT_On

POP    {LR}
BX     LR

```

```

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

```

```

Tab_Rej_exe_Imp      ; Режим Генератора Imp

```

```

BL     Off_Imp      ; Отключение потока

```

```

MOV32 R0, #SRAM_Imp_Tper
LDR    R8, [R0]
MOVS   R8, R8      ; Проверка на 0
IT     EQ
MOVEQ  R8,R11

```

```

MOV    R5, #4
MUL    R8, R8,R5    ; Tper*R5
SUB    R8, R8, R11

```

```

MOV32 R1, #GPIOC
MOV    R2, #GPIO_ODR_ODR_11
STR    R2, [R1,#GPIO_ODR]      ; Регистр выходных данных порта GPIO

```

```

DMA2_1      ; DMA2

```

```

MOV32 R0, #DMA2
MOV32  R2, #0xF7D0F7D
STR    R2, [R0,#DMA_LIFCR]      ; Регистр очистки флага прерывания низкого
уровня DMA

```

```

уровня DMA
STR    R2, [R0,#DMA_HIFCR]      ; Регистр очистки флага прерывания высокого
уровня DMA

```

```

MOV32 R2, #(GPIOC + GPIO_ODR)

```



```

STR    R2, [R0,#DMA_S5M0AR]      ;Запись
MOV32  R2, #SRAM_Out
STR    R2, [R0,#DMA_S5PAR]        ;Чтение
MOV32  R2, #(DMA_SxCR_CHSEL_CH6+DMA_SxCR_PL_Vh+DMA_SxCR_MSIZE_16+
DMA_SxCR_PSIZE_32+DMA_SxCR_DIR_MM +DMA_SxCR_PINC+DMA_SxCR_CIRC)
;CHSEL[110]: выбор канала. PL[11]: уровень приоритета - Очень высокий. PINC: 1
- режим периферийного приращения. CIRC: 1 - круговой режим.
;DIR [1:0]: направление передачи данных.10: Память в память.
STR    R2, [R0,#DMA_S5CR] ; Регистр конфигурации потока: 5 - DMA (DMA_SxCR) (x =
0..7)

```

```

ORR    R9, R9, #Rsos_DMA_MM
MOV    R12, #30000 ; Tmax
MOV    R4, #4

```

```

Tab_Rej_exe_Imp_muud
CMP    R8, R12 ; Проверка на Tmax
BLS    Tab_Rej_exe_Imp_OuD

```

```

BIC    R9, R9, #Rsos_DMA_MM

```

```

DMA2_0 ; DMA2
MOV32  R0, #DMA2
MOV32  R2, #0xF7D0F7D
STR    R2, [R0,#DMA_LIFCR] ; Регистр очистки флага прерывания низкого
уровня DMA
STR    R2, [R0,#DMA_HIFCR] ; Регистр очистки флага прерывания высокого
уровня DMA

```

```

MOV32  R2, #(GPIOC + GPIO_ODR)
STR    R2, [R0,#DMA_S5PAR] ; Запись
MOV32  R2, #SRAM_Out
STR    R2, [R0,#DMA_S5M0AR] ; Чтение
MOV32  R2, #(DMA_SxCR_CHSEL_CH6+DMA_SxCR_PL_Vh+DMA_SxCR_MSIZE_16+
DMA_SxCR_PSIZE_16+DMA_SxCR_DIR_MP+DMA_SxCR_MINC+DMA_SxCR_CIRC)
; CHSEL[110]: выбор канала. PL[11]: уровень приоритета - Очень высокий. MINC: 1
- режим периферийного приращения. CIRC: 1 - круговой режим.
; DIR [1:0]: направление передачи данных.10: Память в периферию.
STR    R2, [R0,#DMA_S5CR] ; Регистр конфигурации потока: 5 - DMA (DMA_SxCR) (x =
0..7)

```

```

ASR    R8, R8, #1 ; R8 = R8*R5/2
LSL    R4, R4, #1 ; R4 = R4 * 2 Вычисление деления TIM (TIM_ARR)
B Tab_Rej_exe_Imp_muud

```

```

Tab_Rej_exe_Imp_OuD

```

```

VMOV    S24, R4 ; На хранение

```

```

MOV32  R0, #Rej_Imp_VUnit
VLDR    S11, [R0] ; 1.0
VLDR    S10, [R0, #4] ; 0.0
VLDR    S13, [R0, #8] ; _Umax
VLDR    S17, [R0, #12] ; _Kkor Коррекция Uamp
VLDR    S12, [R0, #16] ; _Tmax
VLDR    S19, [R0, #20] ; _NoLL

```

```

; Определение коэффициента для изменения выходного напряжения

```

```

MOV32  R1, #SRAM_Imp_Uamp
LDR    R1, [R1]
VMOV    S14, R1
VCVT.F32.U32 S14, S14

```

```

VDIV.F32 S14, S13 ; Kout

```

```
VSUB.F32    S18 ,S11,S14      ; S18 = 1 - Kout
VMUL.F32    S19 ,S18
```

```
MOV32 R0, #SRAM_Imp
LDR    R1, [R0]
MOV32 R0, #Tab_IMP
LSL    R1, R1,#2
LDR    R0, [R0,R1]
```

```
MOV32 R1, #SRAM_Out
MOV    R5, R10      ; Загрузка Счетчика Точек
VMOV          S5,R5
```

```
LDR    R3, [R0],#4
```

```
;Распаковка с пересылкой числа в сопроцессор с плавающей запятой
```

```
UBFX   R7, R3,#0,#16      ; Извлекает битовое поле 16 бит (Tn1) из R3, начиная с
0 бита, расширяет его как число без знака до 32-х бит и сохраняет в регистре R2.
```

```
UBFX   R3, R3,#16,#13     ; Извлекает битовое поле 13 бит (Tn1) из R3, начиная с
16 бита, расширяет его как число без знака до 32-х бит и сохраняет в регистре R3.
```

```
Tab_Rej_exe_Imp_SRAM_Out
```

```
MOV    R6, R7
MOV    R2, R3
```

```
LDR    R3, [R0],#4
```

```
TST    R3, #P_end_Txt      ;Признак конца текста
```

```
BNE Tab_Rej_exe_Imp_DMA_EN
```

```
;Распаковка с пересылкой числа в сопроцессор с плавающей запятой
```

```
UBFX   R7, R3,#0,#16      ;Извлекает битовое поле 16 бит (Tn1) из R3, начиная с
0 бита, расширяет его как число без знака до 32-х бит и сохраняет в регистре R7.
```

```
UBFX   R3, R3,#16,#13     ;Извлекает битовое поле 13 бит (Tn1) из R3, начиная с
16 бита, расширяет его как число без знака до 32-х бит и сохраняет в регистре R3.
```

```
CMP    R7, R6
```

```
IT LS
```

```
ADDLS R7, R6,R11      ;Если Меньше или равно
```

```
SUB    R6, R7,R6      ;dTn = Tn1 - Tn0
```

```
VMOV          S6, R6
```

```
VCVT.F32.U32 S6, S6
```

```
VMOV          S8, R8
```

```
VCVT.F32.U32 S8, S8      ;Tper
```

```
VMUL.F32    S4, S6,S8      ; S4 = (Tn1 - Tn0)*Tper
```

```
VDIV.F32    S6, S4,S12     ; S6 = (Tn1 - Tn0)*Tper/Tmax
```

```
VCVTR.U32.F32S6, S6
```

```
VMOV          R6, S6
```

```
CMP    R6, R11
```

```
IT LT
```

```
MOVL T R6, R11      ; Если Меньше или равно
```

```
VMOV          S6, R6
```

```
VCVT.F32.U32 S6, S6      ; Пересылка числа в сопроцессор с плавающей запятой
```

```
; dTnt
```

```
VMOV          S2,S3, R2,R3 ; Пересылка числа в сопроцессор с плавающей запятой
```

```
VCVT.F32.U32 S2, S2      ; An0
```

```

VCVT.F32.U32 S3, S3 ; An1
VSUB.F32 S5, S3,S2 ; dAn = An1 - An0 - Разница An между точками
перелома
VDIV.F32 S7, S5,S6 ; San = (An1 - An0)*Tmax/((Tn1 - Tn0)*Tper) -
приращение Uamp за один такт или одно обновление TIM1

```

```

VMOV.F32 S15, S10
VMOV.F32 S3, S2

```

Tab_Rej_exe_Imp_clk

```

VCVT.U32.F32 S16,S16
VMOV R2, S16
USAT R2, #12,R2

```

```

STRH R2, [R1],#2 ; Запись в SRAM_Out

```

;Аппроксимация линейная

```

VADD.F32 S15, S15,S11 ; N: S15 = S15 + 1.0

```

```

VMUL.F32 S4, S7,S15

```

```

VADD.F32 S2, S3,S4 ; An = An0 + N*San

```

```

VMUL.F32 S16, S2,S14 ; Uamp*Kout - изменение амплитуды Uamp

```

```

VADD.F32 S16, S16,S17 ;Коррекция Uст

```

```

VADD.F32 S16, S16,S19

```

```

ADD R5, R11 ;Счетчик Точек

```

```

SUBS R6, R6,R11 ; R6 = R6 - 1

```

```

BNE Tab_Rej_exe_Imp_clk

```

```

B Tab_Rej_exe_Imp_SRAM_Out

```

Tab_Rej_exe_Imp_DMA_EN

```

TST R9,#Rsos_DMA_MM
IT NE
ASRNE R5, R5, #1 ;R5 = R5/2

```

```

MOV32 R0, #DMA2

```

```

STR R5, [R0,#DMA_S5NDTR] ; Поток: 5 - DMA регистров количества

```

данных

```

LDR R6, [R0,#DMA_S5CR] ; Регистр конфигурации потока: x DMA (DMA_SxCR) (x =

```

0..7)

```

ORR R6, R6, #DMA_SxCR_EN ;EN: включение потока

```

```

STR R6, [R0,#DMA_S5CR]

```

```

VMOV R4, S24 ;Из хранения

```

```

MOV32 R0, #TIM1

```

```

SUB R4, R4, R11

```

```

STR R4, [R0,#TIM_ARR] ; TIM регистр авто-перезагрузки

```

; Счетчик включить

```

TST R9, #Rsos_DMA_MM

```

```

ITTT EQ

```

```

LDREQ R1, [R0,#TIM_CR1] ; TIM управляющий регистр 1

```

```

ORREQ R1, R1,#TIM_CR1_CEN

```

```

STREQ R1, [R0,#TIM_CR1] ;Выполняется, если режим: Периферия в Память

```

| | |
|-----|------|
| POP | {LR} |
| BX | LR |

,,,,,,,,,,,,,,
,,,,,,,,,,,,,

; Счетчик отключить

: Отключение потока

0..7)

Off_Impt_EN

```
BIC    R9, R9, #Rsos_OUT_On
```

BX LR

;

ALIGNRAM 2

THUMB

DATA

;

```
Tab_IMP      DCD      Tab_IMP_0
              DCD      Tab_IMP_1
              DCD      Tab_IMP_2
              DCD      Tab_IMP_3
              DCD      Tab_IMP_4
              DCD      Tab_IMP_5
              DCD      Tab_IMP_6
              DCD      Tab_IMP_7
```

```

DCD    Tab_IMP_8
DCD    Tab_IMP_9
DCD    Tab_IMP_10
DCD    Tab_IMP_11
DCD    Tab_IMP_12

Tab_IMP_0    DCD    (4000<<16)+0,(4000<<16)+1999,(0<<16)+2000,(0<<16)+3999,P_end_Txt
Tab_IMP_1    DCD    0,(4000<<16)+3999,P_end_Txt
Tab_IMP_2    DCD    (2000<<16)+0,(2000<<16)+999,(4000<<16)+1000,(4000<<16)+1999
DCD    (2000<<16)+2000,(2000<<16)+2999,(0<<16)+3000,(0<<16)+3999,P_end_Txt
Tab_IMP_3    DCD    0,(4000<<16)+1999,(0<<16)+3999,P_end_Txt
Tab_IMP_4    DCD    0,(4000<<16)+1999,(0<<16)+2000,(0<<16)+3999,P_end_Txt
Tab_IMP_5    DCD    0,(0<<16)+1999,(4000<<16)+2000,(0<<16)+3999,P_end_Txt
Tab_IMP_6    DCD    (2000<<16)+0,(2000<<16)+999,(4000<<16)+1000,(2000<<16)+1999
DCD    (2000<<16)+2999,(0<<16)+3000,(2000<<16)+3999,P_end_Txt
Tab_IMP_7    DCD    (2000<<16)+0,(2000<<16)+999,(4000<<16)+1999,(2000<<16)+2000
DCD    (2000<<16)+2999,(0<<16)+3999,P_end_Txt
Tab_IMP_8    DCD    0,(0<<16)+999,(4000<<16)+1999,(4000<<16)+2999,(0<<16)+3999,P_end_Txt
Tab_IMP_9    DCD    (4000<<16)+0,(0<<16)+999,(0<<16)+1999,(4000<<16)+2000
DCD    (4000<<16)+3999,P_end_Txt
Tab_IMP_10   DCD    0,(0<<16)+999,(4000<<16)+1999,(4000<<16)+3999,P_end_Txt
Tab_IMP_11   DCD    0,(626<<16)+100,(1236<<16)+200,(1816<<16)+300,(2351<<16)+400,(2828<<16)+499
DCD    (3236<<16)+599,(3564<<16)+699,(3804<<16)+799,(3951<<16)+899
DCD    (4000<<16)+999,(3951<<16)+1100,(3804<<16)+1200,(3564<<16)+1300
DCD    (3236<<16)+1400,(2828<<16)+1499,(2351<<16)+1599,(1816<<16)+1699
DCD    (1236<<16)+1799,(626<<16)+1899,(0<<16)+1999,(0<<16)+3999,P_end_Txt
Tab_IMP_12   DCD    0,(626<<16)+200,(1236<<16)+400,(1816<<16)+600,(2351<<16)+800
DCD    (2828<<16)+1000,(3236<<16)+1200,(3564<<16)+1400,(3804<<16)+1600
DCD    (3951<<16)+1800,(4000<<16)+1999,(3951<<16)+2199,(3804<<16)+2399
DCD    (3564<<16)+2599,(3236<<16)+2799,(2828<<16)+2999,(2351<<16)+3199
DCD    (1816<<16)+3399,(1236<<16)+3599,(626<<16)+3799,(0<<16)+3999,P_end_Txt

```

ПО Режим Генератора ШИМ,ЧИМ

```

; NAME Rej_PWFM_STM32F4_IAR

PUBLIC Tab_Rej_Pwfm, Tab_Rej_exe_Pwfm, Tab_Rej_exe_Pwfm_Tdt
PUBLIC Tab_Rej_exe_PWFM_exit, Rej_PWFM_Tmax
PUBLIC SRAM_PWFM_Mod

SECTION .text:CODE:REORDER:NOROOT(1)
THUMB
ALIGNRAM 2

;Инициализация Режим Генератора ШИМ, ЧИМ
Tab_Rej_Pwfm

MOV32 R1, #GPIOA
LDR R2, [R1,#GPIO_MODER]
MOV32 R3, #(GPIO_MODER_0_AF+GPIO_MODER_12_AF+GPIO_MODER_15_AF)
BIC R2, R2,R3
ORR R2, R2,#GPIO_MODER_8_AF
ORR R2, R2,#GPIO_MODER_1_AN
STR R2, [R1,#GPIO_MODER] ; Регистр режима порта GPIO (GPIOx_MODER)
MOV32 R2, #GPIOA_AFRH_PA8_TIM1_CH1
STR R2, [R1,#GPIO_AFRH] ; GPIO Н регистр альтернативных функций

MOV32 R0, #SRAM_NumOrd L1

```

```

STR    R11, [R0]
MOV32  R0, #SRAM_NumOrd_L2
STR    R11, [R0]

MOV32  R0, #SRAM_NumOrd_L1
STR    R11, [R0]

MOV32  R0, #SRAM_NumOrd_L2
STR    R11, [R0]

```

```

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

```

```

; Инициализация значений PWM, PFM

```

```

AND    R2, R9, #0x0f           ; Проверка на допустимый Режим

CMP    R2, #Rsos_Rej_PFM
BEQ    Tab_Rej_Pwfm_PFM

```

```

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

```

```

Tab_Rej_Pwfm_PWM      ;Инициализация значений PWM

```

```

MOV32  R0, #Param_Rej_PWM
BL USART1_W           ; Вывод на индикатор (См. FTt_STM32F405RG)
BL Zad_5mC            ; (См. FTt_STM32F405RG)

MOV32  R0, #Param_Lin0_PWFM_Mod_Hand
BL USART1_W           ; Вывод на индикатор (См. FTt_STM32F405RG)

MOV32  R0, #SRAM_PWFM_Tdt
MOV    R2, #10
STR    R2, [R0]

MOV32  R0, #SRAM_PWFM_Tper
MOV    R2, #10000
STR    R2, [R0]

MOV32  R0, #SRAM_PWFM_Timp
MOV    R2, #2000
STR    R2, [R0]

B Tab_Rej_Pwfm_PWFM

```

```

Tab_Rej_Pwfm_PFM      ; Инициализация значений PFM

```

```

MOV32  R0, #Param_Rej_PFM
BL USART1_W           ; Вывод на индикатор (См. FTt_STM32F405RG)
BL Zad_5mC            ; (См. FTt_STM32F405RG)

MOV32  R0, #Param_Lin0_PWFM_Mod_Hand
BL USART1_W           ; Вывод на индикатор (См. FTt_STM32F405RG)

MOV32  R0, #SRAM_PWFM_Tdt
MOV    R2, #10
STR    R2, [R0]

MOV32  R0, #SRAM_PWFM_Tper
MOV    R2, #10000
STR    R2, [R0]

MOV32  R0, #SRAM_PWFM_Timp
MOV    R2, #100

```

```

        STR    R2, [R0]

Tab_Rej_Pwfm_PWFM
    BL  Zad_5mC
        MOV32  R0, #Param_Lin0_PWFM_Mod_Hand
    BL  USART1_W      ; Вывод на индикатор (См. Ftt_STM32F405RG)
        MOV32  R0, #SRAM_PWFM_Tdt
        LDR    R2, [R0]
        MOV32  R0, #Param_Lin3_PWFM_Tdt_zpt
    BL  USART1_W      ; Вывод на индикатор (См. Ftt_STM32F405RG)

        MOV32  R0, #SRAM_PWFM_Tper
        LDR    R2, [R0]
        MOV32  R0, #Param_Lin2_PWFM_Tper_zpt
    BL  USART1_W      ; Вывод на индикатор (См. Ftt_STM32F405RG)

        MOV32  R0, #SRAM_PWFM_Timp
        LDR    R2, [R0]
        MOV32  R0, #Param_Lin1_PWFM_Timp_zpt
    BL  USART1_W      ; Вывод на индикатор (См. Ftt_STM32F405RG)
    BL  Dir_Marker    ; Вывод "<", ">" направления смещение маркера (См. Ftt_STM32F405RG)
    BL  M_Wd_W        ; Вывод маркера (См. Ftt_STM32F405RG)

        MOV32  R0, #SRAM_PWFM_Mod
        STR    R10, [R0]

;включение тактов TIM1, TIM8, ADC1
    MOV32  R0, #RCC
    LDR    R1, [R0,#RCC_APB2ENR]
    MOV32  R2, #(RCC_APB2ENR_TIM1EN+RCC_APB2ENR_TIM8EN+RCC_APB2ENR_ADC1EN)
    ORR    R1, R1, R2
    STR    R1, [R0,#RCC_APB2ENR]

;включение тактов DMA2
    MOV32  R0, #RCC
    LDR    R1, [R0,#RCC_AHB1ENR]
    ORR    R1, R1, #RCC_AHB1ENR_DMA2EN
    STR    R1, [R0,#RCC_AHB1ENR]

;Инициализация      TIM1 - Формирование: ШИМ
    MOV32  R0, #TIM1
    MOV    R2, #(TIM_CR1_CMS_3+TIM_CR1_ARPE);
    STR    R2, [R0,#TIM_CR1]      ; TIM управляющий регистр 1
    STR    R10, [R0,#TIM_CR2]    ; TIM управляющий регистр 2
    MOV    R2, #TIM_SMCR_TS_ETRF
    STR    R2, [R0,#TIM_SMCR]
    MOV    R2, #(TIM_CCMR1_OC1M_PWM2+TIM_CCMR1_OC1PE+TIM_CCMR1_OC1CE+
TIM_CCMR1_OC2M_PWM2+TIM_CCMR1_OC2PE+TIM_CCMR1_OC2CE)
    STR    R2, [R0,#TIM_CCMR1]
    MOV    R2, #(TIM_CCER_CC1E+TIM_CCER_CC1NE + TIM_CCER_CC2E+TIM_CCER_CC2NE)
    STR    R2, [R0,#TIM_CCER]

; Разрешение прерывания TIM1_UP_TIM10_IRQn
    MOV32  R0, #NVIC_ISER0      ; Разрешает прерывания 0...32
    MOV    R2, #(1<<TIM1_UP_TIM10_IRQn)
    STR    R2, [R0]

; Инициализация TIM8 - для запуска ADC
    MOV32  R0, #TIM8
    MOV    R2, #79              ; 1,0 мкс при f = 80 МГц
    STR    R2, [R0,#TIM_ARR]    ; TIM регистр авто-перезагрузки
    MOV    R2, #TIM_CR2_MMS_Upd

```



```

STR    R2, [R0,#TIM_CR2]      ; TIM управляющий регистр 2
MOV    R2, #TIM_CR1_CEN      ; Счетчик включить
STR    R2, [R0,#TIM_CR1]      ; TIM управляющий регистр 1

; Инициализация ADC1
MOV32  R0, #ADC1 ;
LDR    R2, [R0,#ADC_CCR]
ORR    R2, R2,#ADC_CCR_ADCPRE_4 ; ADCPRE[0,1] - ADC предделитель: /4
STR    R2, [R0,#ADC_CCR]
MOV32  R2, #ADC_SMPR2_SMP1_0   ; Выбор времени выборки канала: 15 тактов
STR    R2, [R0,#ADC_SMPR2]
STR    R10, [R0,#ADC_SQR1]     ; ADC_SQR1_L[0000]:1 - длина последовательности
каналов
STR    R10, [R0,#ADC_SQR2]
MOV32  R2, #ADC_SQR3_SQ1_0     ; 1-е преобразование в обычной
последовательности - Канал 1 (PA1).
STR    R2, [R0,#ADC_SQR3]
MOV32  R2, #ADC_CR1_SCAN
STR    R2, [R0,#ADC_CR1]

MOV32  R2, #(ADC_CR2_EXTEN_RT+ADC_CR2_EXTSEL_8TRGO+ADC_CR2_DMA+ADC_CR2_DDS) ;
Обнаружение запуска по восходящему краю, Запуск от таймера 8
STR    R2, [R0,#ADC_CR2]
LDR    R1, [R0,#ADC_CR2]
ORR    R1, R1,#ADC_CR2_ADON      ; ADC1 включить
STR    R1, [R0,#ADC_CR2]
LDR    R1, [R0,#ADC_CR2]
ORR    R1, R1,#ADC_CR2_SWSTART   ; ADC1 включить
STR    R1, [R0,#ADC_CR2]

; Инициализация DMA2
MOV32  R0, #DMA2
MOV32  R1, #0x0F7D0F7D
STR    R1, [R0,#DMA_LIFCR]      ; Регистр очистки флага прерывания низкого
уровня DMA
STR    R1, [R0,#DMA_HIFCR]      ; Регистр очистки флага прерывания высокого
уровня DMA
MOV    R1, #6000
STR    R1, [R0,#DMA_S4NDTR]     ; Поток: 4 - DMA регистр количества данных
MOV32  R1, #(ADC1 + ADC_DR)     ; Адрес Чтение
STR    R1, [R0,#DMA_S4PAR]
MOV32  R1, #SRAM_Out            ;Начало области памяти вывода - 128 кбайт
STR    R1, [R0,#DMA_S4M0AR]
MOV32  R1, #(DMA_SxCR_CHSEL_CH0+DMA_SxCR_PL_Low+DMA_SxCR_MSIZE_16+
DMA_SxCR_PSIZE_16+DMA_SxCR_DIR_PM+DMA_SxCR_MINC+DMA_SxCR_CIRC) ;CHSEL[000]: выбор канала 7.
; PL[11]: уровень приоритета - Очень высокий.
; MINC: 1 - режим приращения памяти. CIRC: 1 - круговой режим.
; DMA_SxCR_DIR_PM[00]: Периферийное устройство в память.
STR    R1, [R0,#DMA_S4CR] ; Регистр конфигурации потока: 4 - DMA (DMA_SxCR) (x =
0..7)
LDR    R1, [R0,#DMA_S4CR] ; Регистр конфигурации потока: 4 DMA (DMA_SxCR) (x =
0..7)
ORR    R1, R1, #DMA_SxCR_EN      ;EN: Включение потока
STR    R1, [R0,#DMA_S4CR]

BL  Zad_5mC ; (См. Ftt_STM32F405RG)

MOV32  R1, #SRAM_Out            ;Начало области памяти вывода - 128 кбайт
STR    R10, [R1]

POP    {LR}
BX     LR

```



```

MOV32 R0, #Param_Lin2_PWFM_Tper_zpt
BL USART1_W ; Вывод на индикатор (См. Ftt_STM32F405RG)

```

```

MOV R2, R4
MOV32 R0, #Param_Lin1_PWFM_Timp_zpt
BL USART1_W ; Вывод на индикатор (См. Ftt_STM32F405RG)
BL M_Wd_W ; Вывод направления смещение маркера (См. Ftt_STM32F405RG)

```

```

;.....

```

```

ALIGNRAM 2

```

```

Tab_Rej_exe_PWFM_OUT_On

```

```

TST R9, #Rsos_Rej_On
BNE Tab_Rej_exe_PWFM_On

MOV32 R0, #TIM1 ; Формирование: ШИМ
LDR R2, [R0,#TIM_CR1]
BIC R2, R2, #TIM_CR1_CEN ; Счетчик отключить
STR R2, [R0,#TIM_CR1] ; TIM управляющий регистр 1

STR R10, [R0,#TIM_DIER] ; TIM отключить прерывание

```

```

Tab_Rej_exe_PWFM_OUT_Off

```

```

MOV R2, #TIM_BDTR_MOE;
STR R2, [R0,#TIM_BDTR]
LDR R2, [R0,#TIM_CCMR1]
MOV R3, #(TIM_CCMR1_OC1M_PWM2+TIM_CCMR1_OC2M_PWM2)
BIC R2, R2, R3
MOV R3, #(TIM_CCMR1_OC1M_REF0+TIM_CCMR1_OC2M_REF1)
ORR R2, R2, R3
STR R2, [R0,#TIM_CCMR1]

```

```

MOV32 R2, #DMA2
LDR R7, [R2,#DMA_S4CR] ; Регистр конфигурации потока: 4 DMA (DMA_SxCR) (x =
0..7)
BIC R7, R7, #DMA_SxCR_EN ; EN: отключение потока
;STR R7, [R2,#DMA_S4CR]

```

```

MOV32 R0, #SRAM_PWFM_Timp
LDR R2, [R0] ; Tper
MOV32 R0, #Param_Lin1_PWFM_Timp_zpt
BL USART1_W ; Вывод на индикатор (См. Ftt_STM32F405RG)

MOV32 R0, #Param_Lin0_PWFM_Off
BL USART1_W ; Вывод на индикатор (См. Ftt_STM32F405RG)
BL M_Wd_W ; Вывод направления смещение маркера (См. Ftt_STM32F405RG)

```

```

B Tab_Rej_exe_PWFM_exit

```

```

;.....

```

```

ALIGNRAM 2

```

```

Tab_Rej_exe_PWFM_On

```

```

MOV32 R0, #Param_Lin0_PWFM_On
BL USART1_W ; Вывод на индикатор (См. Ftt_STM32F405RG)
BL M_Wd_W ; Вывод маркера (См. Ftt_STM32F405RG)

BL Rej_PWFM_TIM1 ; загрузка TIM1 (См. Ftt_STM32F405RG)

; Сбрасывает признак отложенного прерывания 0...32

```

```

MOV32 R0, #NVIC_ICPR0
MOV   R1, #(1<<TIM1_UP_TIM10_IRQn)
STR   R1, [R0]

MOV32 R0, #TIM1      ; Формирование: ШИМ

MOV32 R1, #SRAM_PWM_Mod
LDR   R1, [R1]
CMP   R1, #Mod_Ext
ITTT EQ
BLEQ  PWFm_Reboot_DMA2

MOVEQ R1, #TIM_DIER_UIE      ; Capture/Compare 1 Включить прерывание
STREQ R1, [R0,#TIM_DIER]    ; TIM DMA /прерывания регистр разрешения

LDR   R1, [R0,#TIM_SR]      ; TIM Регистр состояния
BIC   R1, R1, #TIM_SR_UIF   ;Флаг прерывания : Сброс
STR   R1, [R0,#TIM_SR]

LDR   R2, [R0,#TIM_CCMR1]
MOV   R3, #(TIM_CCMR1_OC1M_PWM2+TIM_CCMR1_OC2M_PWM2)
ORR   R2, R2, R3
STR   R2, [R0,#TIM_CCMR1]

LDR   R2, [R0,#TIM_EGR]
ORR   R2, R2, #TIM_EGR_UG;
STR   R2, [R0,#TIM_EGR]

LDR   R2, [R0,#TIM_BDTR]
ORR   R2, R2, #TIM_BDTR_MOE;
STR   R2, [R0,#TIM_BDTR]

;Включить ШИМ
LDR   R2, [R0,#TIM_CR1]
ORR   R2, R2, #TIM_CR1_CEN      ; Счетчик включить
STR   R2, [R0,#TIM_CR1]      ; TIM управляющий регистр 1

```

Tab_Rej_exe_PWFm_exit

```

BIC   R9, R9, #Rsos_OUT_On
POP   {LR}
BX    LR

```

; ENDP

;;

```

ALIGNRAM 2
PWFm_Reboot_DMA2 ;PROC Перезагрузка DMA регистра Количество данных
PUSH {R0-R3}

```

```

MOV32 R0, #SRAM_PWFm_Tper
LDR   R3, [R0]
MOV   R12, #10
UDIV  R3, R3, R12 ;Уменьшаем Тршт до микросекунд

```

PWFm_Reboot_DMA2_cr1 ;Настройка в диапазон < 65500

```

MOV   R12, #64000
CMP   R3, R12
BLS   PWFm_Reboot_DMA2_dma

ASR   R3, R3, #1 ;Tper/2

```

```

B PWFm_Reboot_DMA2_cr1

PWFm_Reboot_DMA2_dma
MOV32 R3, R3
IT EQ
MOVEQ R3, R11

MOV32 R1, #SRAM_PWFm_Rebt
STR R3, [R1] ; SRAM_PWFm_Rebt Используется в Inter_TIM1_UP

MOV32 R1, #DMA2
LDR R7, [R1, #DMA_S4CR] ; Регистр конфигурации потока: 5 DMA (DMA_SxCR) (x =
0..7)

BIC R7, R7, #DMA_SxCR_EN ; EN: отключение потока
STR R7, [R1, #DMA_S4CR]
NOP
NOP
STR R3, [R1, #DMA_S4NDTR] ; Поток: 5 - DMA регистр количества данных

MOV32 R7, #0x0F7D0F7D
STR R7, [R1, #DMA_LIFCR] ; Регистр очистки флага прерывания низкого
уровня DMA
STR R7, [R1, #DMA_HIFCR] ; Регистр очистки флага прерывания высокого
уровня DMA
LDR R7, [R1, #DMA_S4CR] ; Регистр конфигурации потока: 4 DMA (DMA_SxCR) (x =
0..7)

ORR R7, R7, #DMA_SxCR_EN ; EN: Включение потока
STR R7, [R1, #DMA_S4CR]

POP {R0-R3}

BX LR
; ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SECTION MYDATA:CODE:NOROOT
ALIGNRAM 2
THUMB
DATA

```

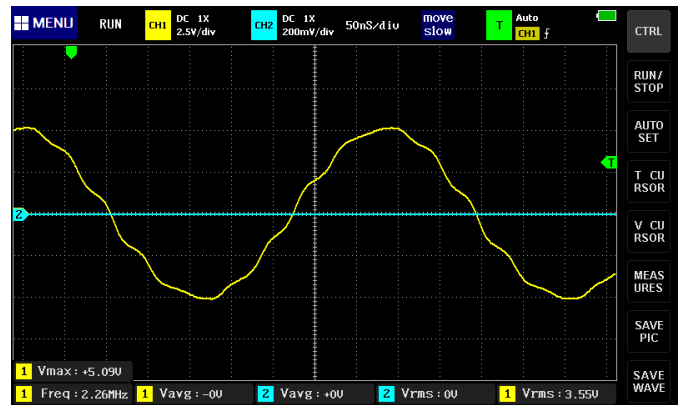
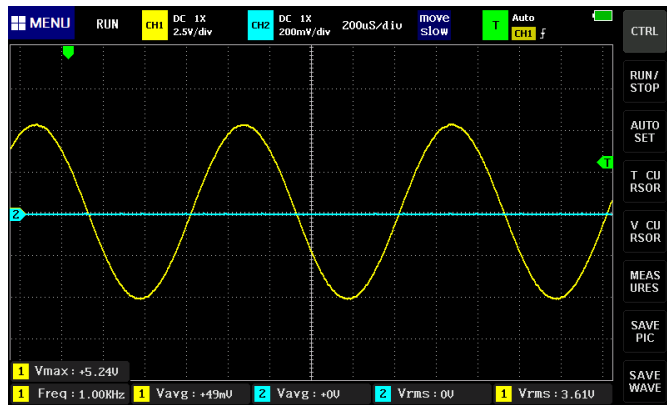
ПО Замечание

Тестирование ПО проведено не полностью. Возможно Вам придется столкнуться с «костылями» или «блохами».

Приложение

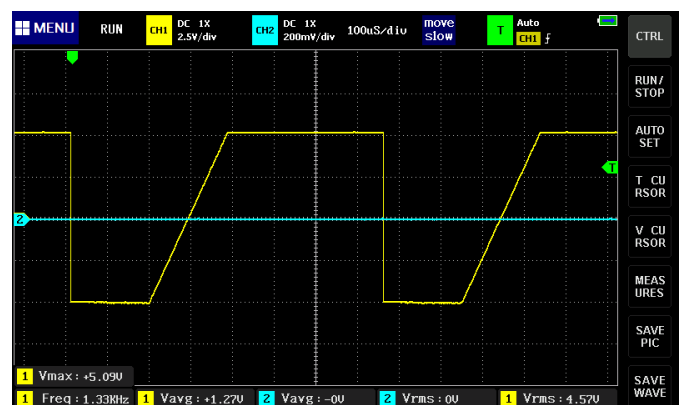
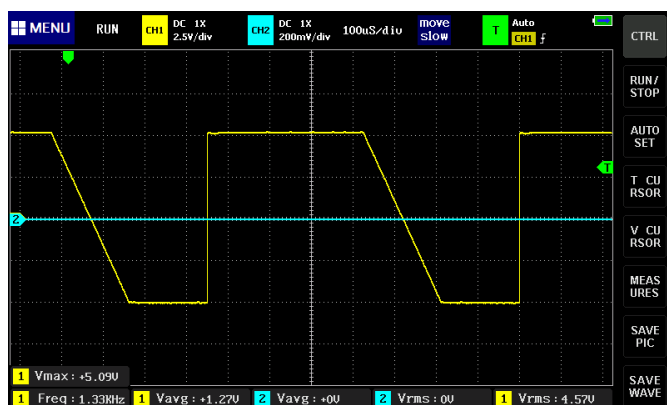
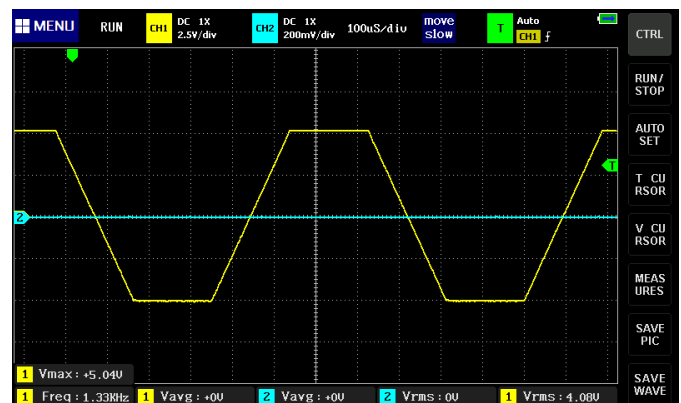
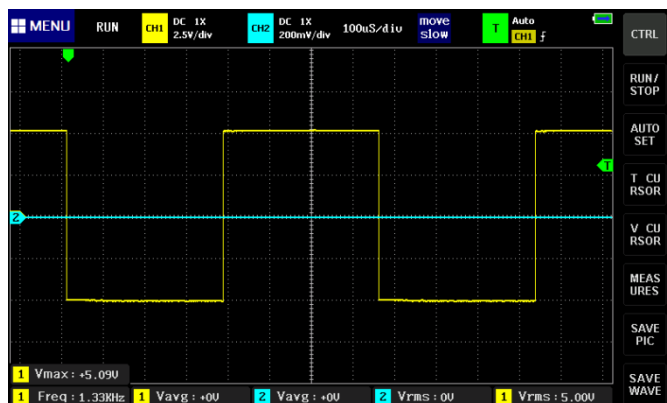
Режим Гsin – синтез синусоидального сигнала

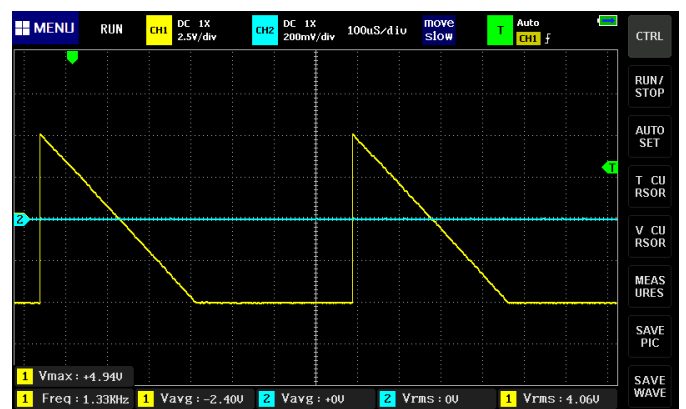
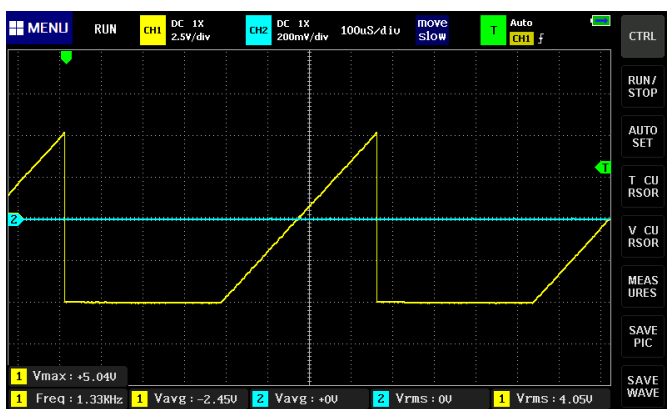
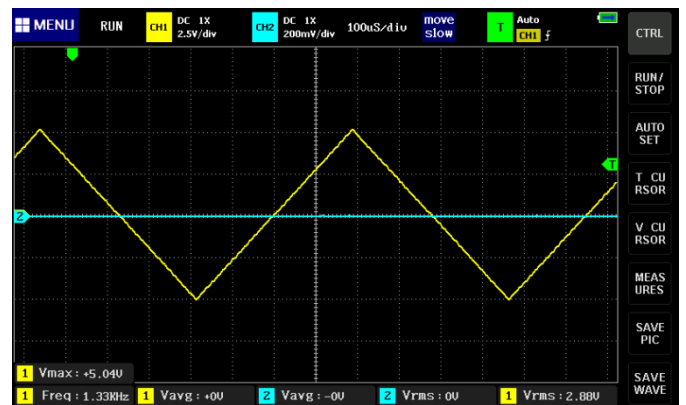
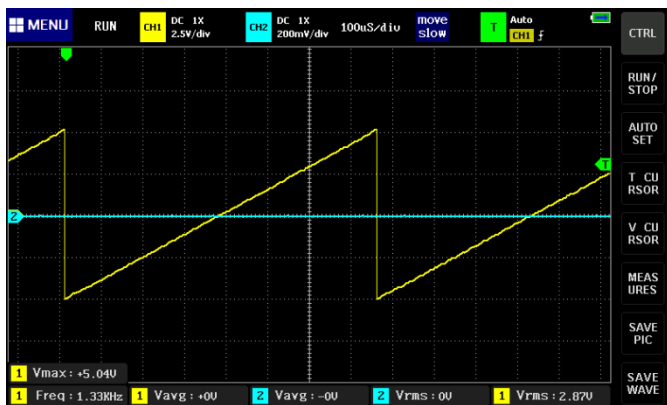
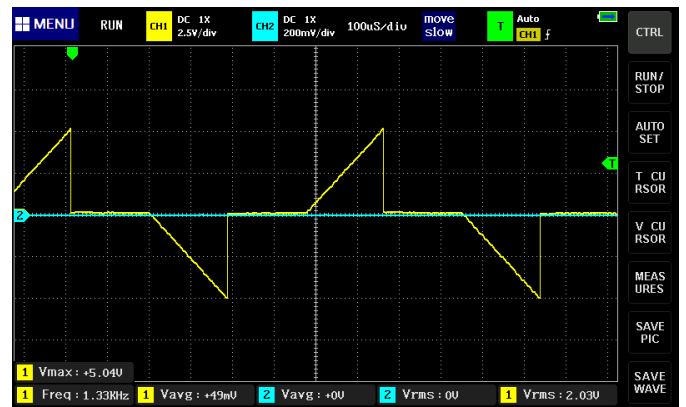
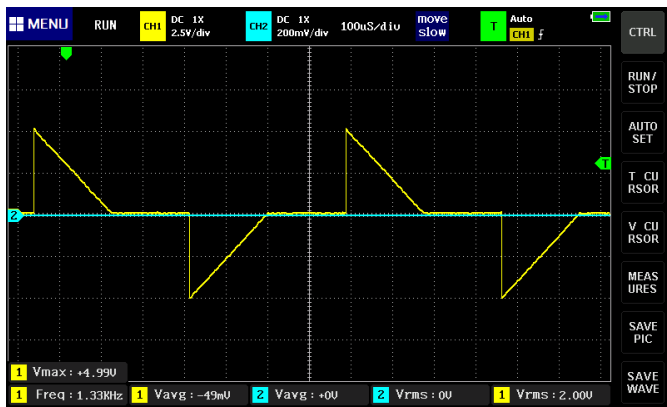
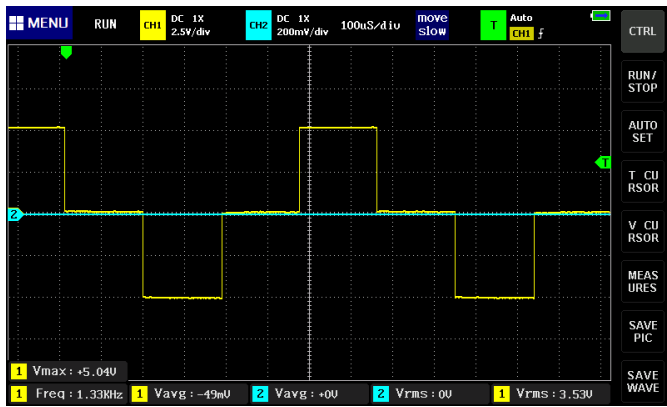
Напряжение смещение $U_{см} = 0V$

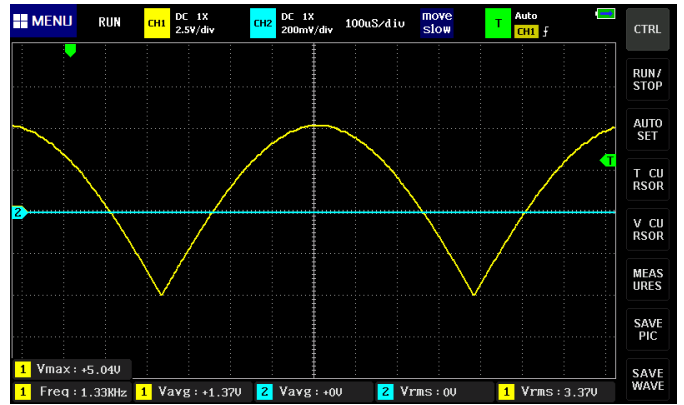
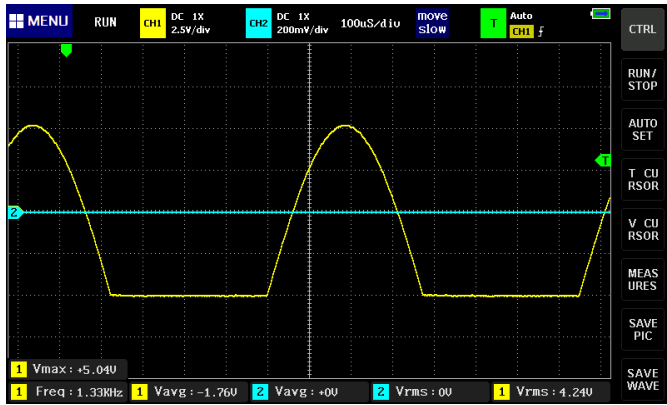


Режим Гфунк – осциллограммы

Напряжение смещение $U_{см} = 0V$







Напряжение смещение $U_{см} = 4,6V$

