1. Список API функций и макросов библиотеки FIL

Раздел содержит список макросов, предоставляемых библиотекой, Раздел содержит полный перечень параметров и функций, какую роль они выполняют,

1.1 Стандартизация названий

По мере пополнения функционала библиотеки для облегченного восприятия команд и быстрого доступа к ним было принято решение - обеспечить стандартизацию названий и разграничения команд по их направленности.

Группы параметров легко выделяются из программного кода, они имеют приставку *__config*, которая сразу дает понять пользователю о принадлежности макроса к группе настроек. Далее идут ключевые слова или подгруппа, основные и критические параметры, в большинстве содержат название *USE*. Их использование сопровождается добавлением чего-либо, поэтому они и называются критическими. Ключевым словом *CALC* обозначаются параметры, включение которых приведет к расширению и добавлению одной и более функций отладки, расчета. Их использование может значительно облегчить отладку разработанного кода.

Некоторые параметры (опциональные) предназначены лишь для детальной настройки конфигурации. Не получив их определения, библиотека продолжит свою работу. Тем не менее, их можно использовать для настройки под специальные условия без низкоуровневого вмешательства через регистровые переменные.

Функции библиотеки FIL выполняют роль не только обертки низкоуровневой части, но и могут выполнять до 15 действий за один вызов. Простые функции можно использовать для установки/очистки ключевых регистров микроконтроллера, инициализацию интерфейсов и прочей периферии. Библиотека лишь предоставляет набор инструментов для отладки

и инициализации, всю основную работу по написанию логики работы программного обеспечения выполняет программист.

1.2 Перечень макросов конфигурации периферии

__configUSE_RCC

```
#define __configUSE_RCC                (0)
```

Описание: Критически необходимый параметр. Используется линкером для предоставления пользователю доступа к сектору тактирования библиотеки, файл RCC.h

Значение:

0 – доступ для пользователя заблокирован, инструменты нельзя использовать в пользовательском пространстве;

1 – предоставление доступа для пользователя. Можно использовать базовый комплект функций библиотеки для контроллера RCC.

__configUSE_GPIO

```
#define __configUSE_GPIO               (0)
```

Описание: Критически необходимый параметр. Используется линкером для предоставления пользователю доступа к сектору настройки портов библиотеки, файл GPIO.h

Значение:

0 – доступ для пользователя заблокирован, инструменты нельзя использовать в пользовательском пространстве;

1 – предоставление доступа для пользователя. Можно использовать базовый комплект функций библиотеки для контроллера GPIO.

Примечания: параметр необходим для создания карты портов, потребует включения для вариантов конфигурации, изложенных в разделе 2.

__configUSE_TIM

```
#define __configUSE_TIM                 (0)
```

Описание: Критически необходимый параметр. Используется линкером для предоставления пользователю доступа к сектору настройки таймеров, файл TIM.h

Значение:

0 – доступ для пользователя заблокирован, инструменты нельзя использовать в пользовательском пространстве;

1 – предоставление доступа для пользователя. Можно использовать базовый комплект функций библиотеки для настройки и подключения таймеров.

__configUSE_USART

```
#define __configUSE_USART               (0)
```

Описание: Критически необходимый параметр. Используется линкером для предоставления пользователю доступа к сектору настройки интерфейса USART, файл USART.h

Значение:

0 – доступ для пользователя заблокирован, инструменты нельзя использовать в пользовательском пространстве;

1 – предоставление доступа для пользователя. Можно использовать базовый комплект функций библиотеки для настройки и подключения USART интерфейса для передачи и приема данных.

__configUSE_DMA

```
#define __configUSE_DMA                 (0)
```

Описание: Критически необходимый параметр. Используется линкером для предоставления пользователю доступа к сектору настройки контроллера прямого доступа к памяти, файл DMA.h

Значение:

0 – доступ для пользователя заблокирован, инструменты нельзя использовать в пользовательском пространстве;

1 – предоставление доступа для пользователя. Можно использовать базовый комплект функций библиотеки для настройки и контроллера прямого доступа DMA.

__configUSE_I2C

```
#define __configUSE_I2C                 (0)
```

Описание: Критически необходимый параметр. Используется линкером для предоставления пользователю доступа к сектору настройки интерфейса I2C, файл I2C.h

Значение:

0 – доступ для пользователя заблокирован, инструменты нельзя использовать в пользовательском пространстве;

1 – предоставление доступа для пользователя. Можно использовать базовый комплект функций библиотеки для настройки и подключения I2C для передачи данных.

__configUSE_ADC

```
#define __configUSE_ADC                 (0)
```

Описание: Критически необходимый параметр. Используется линкером для предоставления пользователю доступа к сектору настройки АЦП, файл ADC.h

Значение:

0 – доступ для пользователя заблокирован, инструменты нельзя использовать в пользовательском пространстве;

1 – предоставление доступа для пользователя. Можно использовать базовый комплект функций библиотеки для настройки и аналого-цифрового преобразователя.

__configUSE_EXTI

```
#define __configUSE_EXTI                (0)
```

Описание: Критически необходимый параметр. Используется линкером для предоставления пользователю доступа к сектору настройки внешний прерываний, файл EXTI.h

Значение:

0 – доступ для пользователя заблокирован, инструменты нельзя использовать в пользовательском пространстве;

1 – предоставление доступа для пользователя. Можно использовать базовый комплект функций библиотеки для настройки и внешних прерываний.

__configUSE_RTC

```
#define __configUSE_RTC                (0)
```

Описание: Критически необходимый параметр. Используется линкером для предоставления пользователю доступа к сектору настройки таймера реального времени, файл RTC.h

Значение:

0 – доступ для пользователя заблокирован, инструменты нельзя использовать в пользовательском пространстве;

1 – предоставление доступа для пользователя. Можно использовать базовый комплект функций библиотеки для настройки и использования таймера реального времени RTC.

__configCALC_RCC

```
#define __configCALC_RCC               (0)
```

Описание: Опциональный параметр. Используется библиотекой FIL для предоставления доступа к расширенному функционалу контроллера RCC, файл RCC.h

Значение:

0 – доступ для пользователя заблокирован, расширенный набор функций не будет добавлен;

1 – предоставление доступа для пользователя. Можно использовать специальный комплект функций библиотеки для расчетов и отладки модуля RCC.

Примечания: параметр может быть задан в случае активации макроса __configUSE_RCC. В противном случае игнорируется и ничего не добавит.

__configCALC_TIM

```
#define __configCALC_TIM                (0)
```

Описание: Опциональный параметр. Используется библиотекой FIL для предоставления доступа к расширенному функционалу таймеров, файл TIM.h

Значение:

0 – доступ для пользователя заблокирован, расширенный набор функций не будет добавлен;

1 – предоставление доступа для пользователя. Можно использовать специальный комплект функций библиотеки для расчетов и отладки модуля TIM.

Примечания: параметр может быть задан в случае активации макроса __configUSE_TIM. В противном случае игнорируется и ничего не добавит.

__configCALC_USART

```
#define __configCALC_USART              (0)
```

Описание: Опциональный параметр. Используется библиотекой FIL для предоставления доступа к расширенному функционалу интерфейса USART, файл USART.h

Значение:

0 – доступ для пользователя заблокирован, расширенный набор функций не будет добавлен;

1 – предоставление доступа для пользователя. Можно использовать специальный комплект функций библиотеки для расчетов и отладки модуля USART.

Примечания: параметр может быть задан в случае активации макроса __configUSE_USART. В противном случае игнорируется и ничего не добавит.

```
#define __configCALC_I2C_SCANNING     (1)
```

Описание: Опциональный параметр. Добавляет расширенный набор функций отладки интерфейса I2C.

Значение:

0 – Доступ к расширенному набору не будет предоставлен.

1 – Предоставление доступа к расширенному набору функций I2C.

__configCONVERT_Volts

```
#define __configCONVERT_Volts          (0)
```

Описание: Опциональный параметр. Изменяет выходной параметр данных, преобразуя данные с АЦП в напряжение.

Значение:

0 – выходные данные будут в дискретном формате

1 – выходные данные будут преобразованы в вольты

Примечание: Не рекомендуется проводить прием данных в прерывании по АЦП, уменьшает время реакции системы. Смотри описание функций AnalogReadInjected и AnalogReadRegular.

__configUSE_Battery_Charging

```
#define __configUSE_Battery_Charging     (0)
```

Описание: Опциональный параметр. Используется для подключения измерения напряжения на батарейке платы.

Значение:

0 – Измерение батарейки не будет добавлено в очередь обработки АЦП.

1 - Измерение батарейки будет добавлено в очередь обработки АЦП.

__configUSE_Temperature_Sensor

```
#define __configUSE_Temperature_Sensor  (0)
```

Описание: Опциональный параметр. Используется для подключения измерения температуры через встроенный термодатчик.

Значение:

0 – Измерение термодатчика не будет добавлено в очередь обработки АЦП.

1 - Измерение термодатчика будет добавлено в очередь обработки АЦП.

Примечание: Включение параметра __configUSE_Battery_Charging имеет высший приоритет, при его включении параметр термодатчика будет игнорирован.

__configADC_InterruptRequest

```
#define __configADC_InterruptRequest    (1)
```

Описание: Опциональный параметр. Включает прерывание АЦП.

Значение:

0 – После инициализации прерывание АЦП не будет запущено.

1 – Запуск прерывания после инициализации АЦП.

__configADC_DMARequest

```
#define __configADC_DMARequest          (1)
```

Описание: Опциональный параметр. Выполняет подготовительные действия для работы с контроллером DMA.

Значение:

0 – Инициализация АЦП в обычном режиме.

1 – Запуск совместной работы с контроллером DMA.

__configADC_Divider

```
#define __configADC_Divider             (3)
```

Описание: Опциональный параметр. Задает величину делителя шины АЦП.

Значение:

0 – Частоты шины АЦП делится на 2 от шины APB2.

1 – Частоты шины АЦП делится на 4 от шины APB2.

2 – Частоты шины АЦП делится на 6 от шины APB2.

3 – Частоты шины АЦП делится на 8 от шины APB2.

__configADC_RESOLUTION

```
#define __configADC_RESOLUTION          (12)
```

Описание: Опциональный параметр. Задает величину разрешения АЦП.

Значение:

6 – Разрядности АЦП будет настроена на 6 бит.

8 – Разрядности АЦП будет настроена на 8 бит.

10 – Разрядности АЦП будет настроена на 10 бит.

12 – Разрядности АЦП будет настроена на 12 бит.

__configADC_CYCLES

```
#define __configADC_CYCLES              (ADC_480_CYCLES)
```

Описание: Опциональный параметр. Указывается величина количества циклов обработки каждого канала АЦП.

Значение:

ADC_3_CYCLES – 3 цикла обработки.

ADC_15_CYCLES – 15 циклов обработки.

ADC_28_CYCLES – 28 циклов обработки.

ADC_56_CYCLES – 56 циклов обработки.

ADC_84_CYCLES – 84 цикла обработки.

ADC_112_CYCLES – 112 циклов обработки.

ADC_144_CYCLES – 144 цикла обработки.

ADC_480_CYCLES – 480 циклов обработки.

__configI2C_FindListSize

```
#define __configI2C_FindListSize     (5)
```

Описание: Опциональный параметр. Указывается максимальное количество устройств, адреса которых можно запоминать в процессу выполнения функции I2C_FindDevices().

Значение: Целое число устройств. Записывается в структуру I2CStatus.

__configI2C_TIMEOUT

```
#define __configI2C_TIMEOUT          (20000)
```

Описание: Опциональный параметр. Задает величину таймаута – времени ожидания соединения и ключевых флагов шины I2C.

Значение: Значение количества итераций ожидания, указывать значение больше от 5000.

Board_Config

Описание: Главная функция применения настроек. Стандартизированное макроопределение применения настроек. Используется готовый вариант из конфигураций либо написанный пользователем лично.

InitPeriph

Описание: Функция настройки режима работы портов микроконтроллера. Стандартизированное макроопределение применения настроек. Используется готовый вариант из конфигураций либо написанный пользователем лично.