

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Информационная безопасность систем и технологий»

Отчет

по практической работе №6

на тему «Функции управления аналого-цифровым преобразованием в
среде STM32CubeIDE»

Дисциплина: ПМК

Группа: 21ПИ1

Выполнил: Гусев Д. А.

Количество баллов:

Дата сдачи:

Принял: Хворостухин С. П.

1 Цель работы: изучить функции управления аналого-цифровым преобразованием библиотеки HAL в среде разработки STM32CubeIDE.

2 Задание на практическую работу.

2.1 Выполнить настройку АЦП в графическом интерфейсе среды разработки STM32CubeIDE в соответствии с вариантом, приведенным в таблице 1. АЦП сконфигурировать в режиме многоканального многократного преобразования. В работе АЦП использовать регулярную группу каналов. Количество преобразований во дном цикле работы установить равным количеству обрабатываемых каналов. Все каналы должны быть задействованы в преобразовании. Вариант задания представлен на рисунке 1.

8	<i>AIN0, AIN7, VREF INT</i>
---	-----------------------------

Рисунок 1 — Вариант задания

2.2 Сгенерировать программный код, соответствующий заданной конфигурации.

2.3 Проанализировать функции настройки контроллеров АЦП и DMA-файл main.c.

2.4 Проанализировать программный код библиотеки HAL, приведенный в файлах:

- stm32f4xx_hal_dma.c;
- stm32f4xx_it.c;
- stm32f4xx_hal_msp.c;
- stm32f4xx_hal_adc.c.

2.5 Сформировать описание проанализированного программного кода.

3 Выполнение практической работы:

3.1 Была выполнена настройка АЦП в графическом интерфейсе среды разработки STM32CubeIDE.

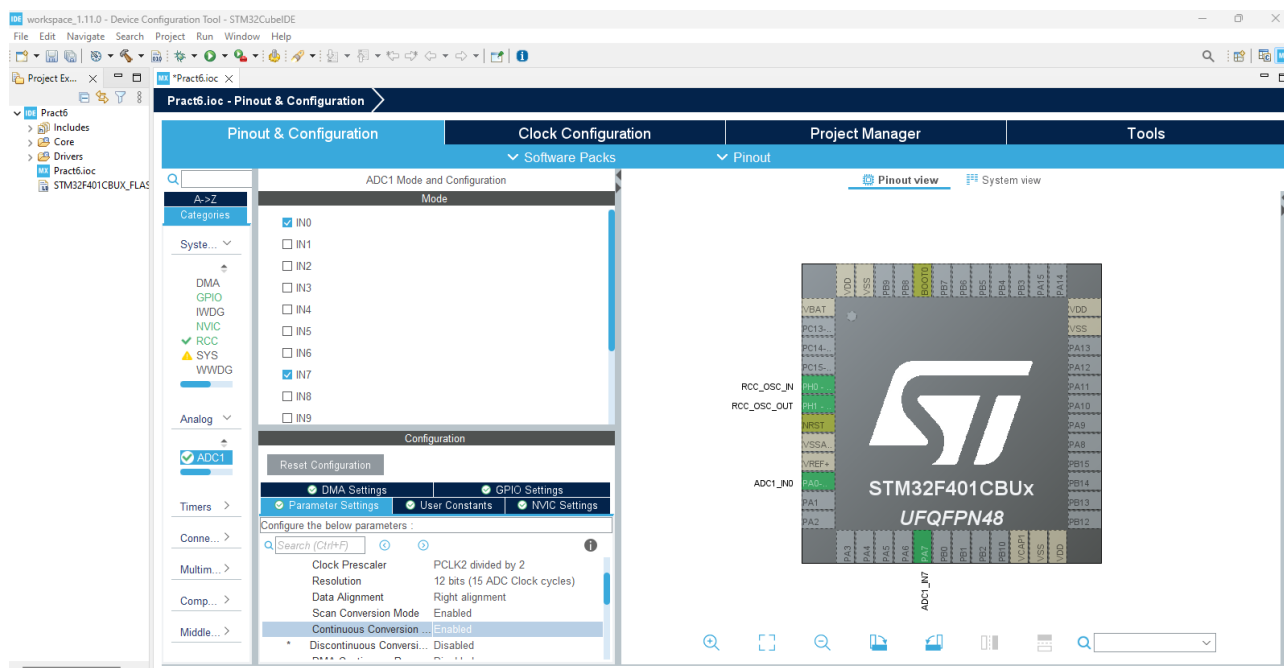


Рисунок 2 — Настройка АЦП

3.2 Был сгенерирован программный код.

3.3 Были проанализированы функции настройки контроллеров АЦП и DMA в файле `main.c`, а также было составлено их описание.

- Функция *MX_ADC1_Init* служит для инициализации модуля аналогово-цифрового преобразования ADC1 и находится в файле *main.c*. Она настраивает параметры и регистры этого модуля для его корректной работы;

Параметры функции — нет параметров;

Возвращаемое значение — нет возвращаемого значения;

Краткий алгоритм работы:

Установка параметров тактовой частоты, разрешения, порядка выполнения преобразований и других характеристик; Инициализация ADC1 с настройками, определенными в предыдущем шаге; Настройка выбранного канала ADC1 (в данном случае CHANNEL_0) с указанием его ранга в последовательности преобразования и времени выборки сигнала.

- Функция *`MX_DMA_Init`* инициализации и синхронизации модуля DMA находится в файле `main.c`.

Параметры функции — нет параметров;

Возвращаемое значение — нет возвращаемого значения;

Краткий алгоритм работы:

Включает тактирование DMA контроллера DMA2; Устанавливается приоритет прерывания DMA2_Stream0_IRQn в 0; Включается прерывание DMA2_Stream0_IRQn.

- Функция инициализации GPIO *MX_GPIO_Init* находится в файле *main.c*.

Параметры функции — нет параметров;

Возвращаемое значение — нет возвращаемого значения;

Краткий алгоритм работы:

3.4 Был проанализирован и описан код в файле *stm32f4xx_hal_dma.c*.

- Функция запуска *DMA HAL_DMA_Start*.

Возвращаемое значение: структура *HAL_StatusTypeDef*;

Параметры функции:

*DMA_HandleTypeDef *hdma*: указатель на структуру, содержащую настройки и состояние DMA канала; *uint32_t SrcAddress*: адрес источника данных; *uint32_t DstAddress*: адрес назначения данных; *uint32_t DataLength*: длина передаваемых данных.

Краткий алгоритм работы:

Проверка параметров функции с помощью макроса *assert_param*; Блокировка ресурса, чтобы избежать конфликтов доступа; Проверка состояния DMA: если DMA готов к работе, то продолжаем, иначе возвращаем ошибку *HAL_BUSY*; Устанавливаем состояние DMA в *HAL_DMA_STATE_BUSY*; Инициализируем код ошибки в *hdma* на *HAL_DMA_ERROR_NONE*; Конфигурируем адреса источника, назначения и длину данных с помощью функции *DMA_SetConfig*; Включаем периферийное устройство DMA; Если DMA не был готов к работе, то разблокируем ресурс и возвращаем ошибку *HAL_BUSY*. В противном случае возвращаем *HAL_OK*.

- Функция инициализации передачи DMA с включенным перебиванием *HAL_DMA_Start_IT* находится в файле *stm32f4xx_hal_dma.c*.

Возвращаемое значение: структура *HAL_StatusTypeDef*;

Параметры функции: *DMA_HandleTypeDef *hdma*: указатель на структуру, содержащую настройки и состояние DMA канала; *uint32_t SrcAddress*: адрес источника данных; *uint32_t DstAddress*: адрес назначения данных; *uint32_t DataLength*: длина передаваемых данных.

Краткий алгоритм работы:

Расчет базового адреса DMA и номера потока DMA; Проверка параметров функции с помощью макроса *assert_param*; Блокировка ресурса, чтобы избежать конфликтов доступа; Проверка состояния DMA: если DMA готов к работе, продолжаем, иначе возвращаем ошибку *HAL_BUSY*; Устанавливаем состояние DMA в *HAL_DMA_STATE_BUSY*; Инициализируем код ошибки в *hdma* на *HAL_DMA_ERROR_NONE*; Конфигурируем адреса источника, назначения и длину данных с помощью функции *DMA_SetConfig*; Очищаем все флаги прерываний по правильному смещению в регистре; Включаем общие прерывания (*Transfer Complete*, *Transfer Error*, *Direct Mode Error*); Если установлена функция обратного вызова для переключения половины передачи, включаем прерывание *Half Transfer*; Включаем периферийное устройство DMA. Если DMA не был готов к работе, разблокируем ресурс и возвращаем ошибку *HAL_BUSY*. В противном случае возвращаем *HAL_OK*.

3.5 Был проанализирован и описан код в файле *stm32f4xx_it.c*;

- Функция обработки системного таймера отсчета времени *SysTick_Handler* находится в файле *stm32f4xx_it.c*.

Параметры функции — нет параметров;

Возвращаемое значение — нет возвращаемого значения;

- Функция обработки глобальных переменных *ADC1 ADC_IRQHandler* находится в файле *stm32f4xx_it.c*.

Параметры функции — нет параметров;

Возвращаемое значение — нет возвращаемого значения;

- Функция обработки переменных DMA2 *DMA2_Stream0_IRQHandler* находится в файле *stm32f4xx_it.c*.

Параметры функции — нет параметров;

Возвращаемое значение — нет возвращаемого значения;

3.6 Был проанализирован и описан код в файле *stm32f4xx_hal_msp.c*;

- Функция инициализации ADC MSP *HAL_ADC_MspInit* находится в файле *stm32f4xx_hal_msp.c*.

Параметры функции — нет параметров;

Возвращаемое значение — нет возвращаемого значения;

- Функция деинициализации ADC MSP *HAL_ADC_MspDeInit* находится в файле *stm32f4xx_hal_msp.c*.

Параметры функции — указатель на структуру *ADC_HandleTypeDef** *hadc*;

Возвращаемое значение — нет возвращаемого значения;

3.7 Был проанализирован и описан код в файле *tm32f4xx_hal_adc.c*.

- Функция инициализации периферийного устройства ADC *HAL_ADC_Init* находится в файле *tm32f4xx_hal_adc.c*.

Параметры функции — указатель на структуру *ADC_HandleTypeDef** *hadc*;

Возвращаемое значение — структура *HAL_StatusTypeDef*;

Краткий алгоритм работы:

Проверяется корректность переданного указателя на структуру *hadc*;
Проверяются параметры структуры *hadc->Init* на соответствие допустимым значениям; Выполняется инициализация обратных вызовов и низкоуровневых настроек, включая установку колбеков и инициализацию обработчика ошибок;
Проверяется состояние модуля АЦП и инициализируется, если эти действия успешно завершены; В конце функции освобождается блокировка и возвращается статус инициализации.

- Функция включения запроса ADC DMA после последней передачи и включения периферийного устройства ADC *HAL_ADC_Start_DMA* находится в файле *tm32f4xx_hal_adc.c*.

Параметры функции:

DMA_HandleTypeDef *hdma: указатель на структуру, содержащую настройки и состояние DMA канала; uint32_t SrcAddress: адрес источника данных; uint32_t DstAddress: адрес назначения данных; uint32_t DataLength: длина передаваемых данных.

Возвращаемое значение — структура HAL_StatusTypeDef;

Краткий алгоритм работы:

Проверяются параметры структуры hadc->Init на соответствие допустимым значениям; Включается периферийное устройство АЦП (если оно не было включено) и ожидается время стабилизации; Проверяется режим DMA и отключается, если уже включен; Обновляется состояние модуля АЦП, устанавливаются колбеки для обработки событий DMA, очищаются флаги и запускается DMA канал для передачи данных; Проверяется наличие внешнего триггера и запуск преобразования, если его нет; В случае ошибки устанавливается внутренняя ошибка модуля АЦП.

- Функция отключения ADC DMA и периферийного устройства АЦП *HAL_ADC_Stop_DMA* находится в файле *tm32f4xx_hal_adc.c*.

Параметры функции — указатель на структуру ADC_HandleTypeDef* hadc;

Возвращаемое значение — структура HAL_StatusTypeDef;

Краткий алгоритм работы:

Проверяется корректность параметра hadc->Instance.

Выполняется блокировка процесса; Прекращаются текущие преобразования на регулярных и внедренных каналах, а затем отключается периферийное устройство АЦП; Проверяется, что АЦП успешно отключено, отключается режим DMA и отключается канал DMA (если он был в режиме занятого);

Выключается прерывание переполнения АЦП; Обновляется состояние модуля АЦП; Процесс разблокируется; Функция возвращает статус операции.

- Функция для настройки параметров канала АЦП *HAL_ADC_ConfigChannel* находится в файле *tm32f4xx_hal_adc.c*.

Параметры функции:

ADC_HandleTypeDef hadc*- Указатель на структуру *ADC_HandleTypeDef*, которая представляет конфигурацию и статус ADC; *ADC_ChannelConfTypeDef* sConfig* - Указатель на структуру *ADC_ChannelConfTypeDef*, которая содержит конфигурацию канала ADC, включая канал, ранг и время сэмплирования.

Возвращаемое значение — структура *HAL_StatusTypeDef*;

Проверяется корректность параметров; Происходит блокировка процесса; Устанавливаются новые значения времени сэмплирования для выбранного канала; Устанавливаются значения ранга выбранного канала; В случае выбора канала ADC1 Channel_18 для VBAT Channel, включается VBATE; Если выбран канал ADC1 Channel_16 или Channel_18 для датчика температуры или Channel_17 для VREFINT, включается TSVREFE; В случае датчика температуры, ожидается стабилизация и потом процесс разблокируется; Функция возвращает статус HAL_OK.

- Функция настройки аналогового тамера *HAL_ADC_AnalogWDGConfig* находится в файле *tm32f4xx_hal_adc.c*.

Параметры функции:

ADC_HandleTypeDef hadc*: Указатель на структуру *ADC_HandleTypeDef*, представляющую конфигурацию и статус ADC (Analog-to-Digital Converter) на микроконтроллерах STM32; *ADC_AnalogWDGConfTypeDef* AnalogWDGConfig*: Указатель на структуру *ADC_AnalogWDGConfTypeDef*, которая содержит параметры для настройки аналоговой защиты ADC, такие как режим защиты, канал, режим прерывания и пороговые значения.

Возвращаемое значение — структура *HAL_StatusTypeDef*;

Кратки алгоритм работы:

Проверяет параметры функции с помощью макросов `assert_param`;
Блокирует доступ к ADC для обеспечения безопасности; В зависимости от режима прерывания, включает или выключает прерывание аналоговой защиты ADC; Сбрасывает биты настроек аналоговой защиты в регистре CR1; Устанавливает режим включения аналоговой защиты, верхний и нижний пороговые значения в регистрах CR1, HTR, и LTR соответственно; Устанавливает канал для аналоговой защиты в регистре CR1; Разблокирует доступ к ADC; Возвращает статус выполнения функции (HAL_OK в случае успешного выполнения).

- Функция инициализации ADC в соответствии с указанными параметрами *ADC_Init* находится в файле *tm32f4xx_hal_adc.c*.

Возвращаемое значение — нет значения;

Параметры функции:

Указатель на структуру `ADC_HandleTypeDef`, содержащую параметры настройки ADC

Краткий алгоритм работы:

Определяет указатель на общий регистр управления ADC (`ADC_Common_TypeDef`) исходя из переданного указателя на конкретный ADC (`hadc`); Устанавливает предделитель тактовой частоты ADC.

Устанавливает режим сканирования ADC; Устанавливает разрешение ADC; Устанавливает выравнивание данных ADC; Если выбрано внешнее срабатывание (триггер) для запуска преобразования, устанавливает внешний триггер и его полярность; Включает или выключает режим непрерывного преобразования ADC; Если выбран режим дискретного преобразования, устанавливает количество дискретных преобразований и включает соответствующий режим; Устанавливает количество преобразуемых каналов ADC; Включает или выключает запросы DMA для непрерывного режима преобразования ADC; Включает или выключает выбор окончания преобразования ADC.

4 Вывод: были изучены функции управления аналого-цифровым преобразованием библиотеки HAL в среде разработки STM32CubeIDE.