

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук

должность, уч. степень, звание

подпись, дата

О.А. Кононов

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №9

Работа с DMA

по курсу: ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4711

подпись, дата

Хасанов Б.Р.

инициалы, фамилия

Санкт-Петербург 2020

Цель работы

Исследовать возможности использования DMA, основные характеристики DMA, взаимодействие с другими устройствами в составе микроконтроллера, с помощью микроконтроллера STM32F407VG.

1 Выполнение работы

1.1 Сведения о DMA

Direct Memory Access (DMA, рос. “Прямой доступ к памяти”) – механизм, используемый в контроллерах ARM для перемещения данных между памятью и периферией без участия процессора. Ключевым моментом является, то что при использовании DMA на перемещение данных не используются ресурсы процессора, что может быть особенно критичным при создании приложений, работающих с большим количеством данных и активно использующих периферию. Работа DMA обеспечивается отдельным контроллером, который выполняет определенные действия по команде процессора.

Рассматриваемый контроллер имеет в своем распоряжении сразу два контроллера DMA, которые обеспечивают в общей сложности 16 потоков (по 8 потоков на каждый контроллер), каждый из которых предназначен для управления запросами к памяти от одного или нескольких устройств. Каждый поток может обеспечивать до 8 каналов (запросов). Каждый контроллер DMA имеет устройство разрешения конфликтов для обработки запросов в соответствии с их приоритетом.

Контроллер DMA позволяет производить запись данных в трех направлениях:

- от периферии в память;
- из памяти к периферии;
- из памяти в память.

Передача ведется либо в режиме непосредственной передачи, либо в режиме очереди. Поддерживается различный размер данных для передачи, причем размер данных для приемника и источника может быть

неодинаковым. В таком случае DMA определяет данную ситуацию и выполняет необходимые действия для оптимизации передачи, однако эта возможность поддерживается только в режиме очереди.

Кроме того, очень полезной может оказаться функция циклической передачи, при использовании которой передача данных начинается с начального адреса снова после передачи последней единицы данных источника.

Программист может задавать приоритеты для запроса каждого конкретного потока или приоритет будет определяться на основе значений по умолчанию.

1.2 Создание проекта

Проект был создан с помощью программы STM32CubeMX, где производилась настройка необходимых портов ввода/вывода, АЦП, DMA, а также тактовой частоты процессора. АЦП находится в режиме постоянного преобразования. Производится считывание данных АЦП по четырём каналам последовательно. Через DMA данные записываются в память, минуя процессор. Пример настройки представлен на рисунках 1 – 4.

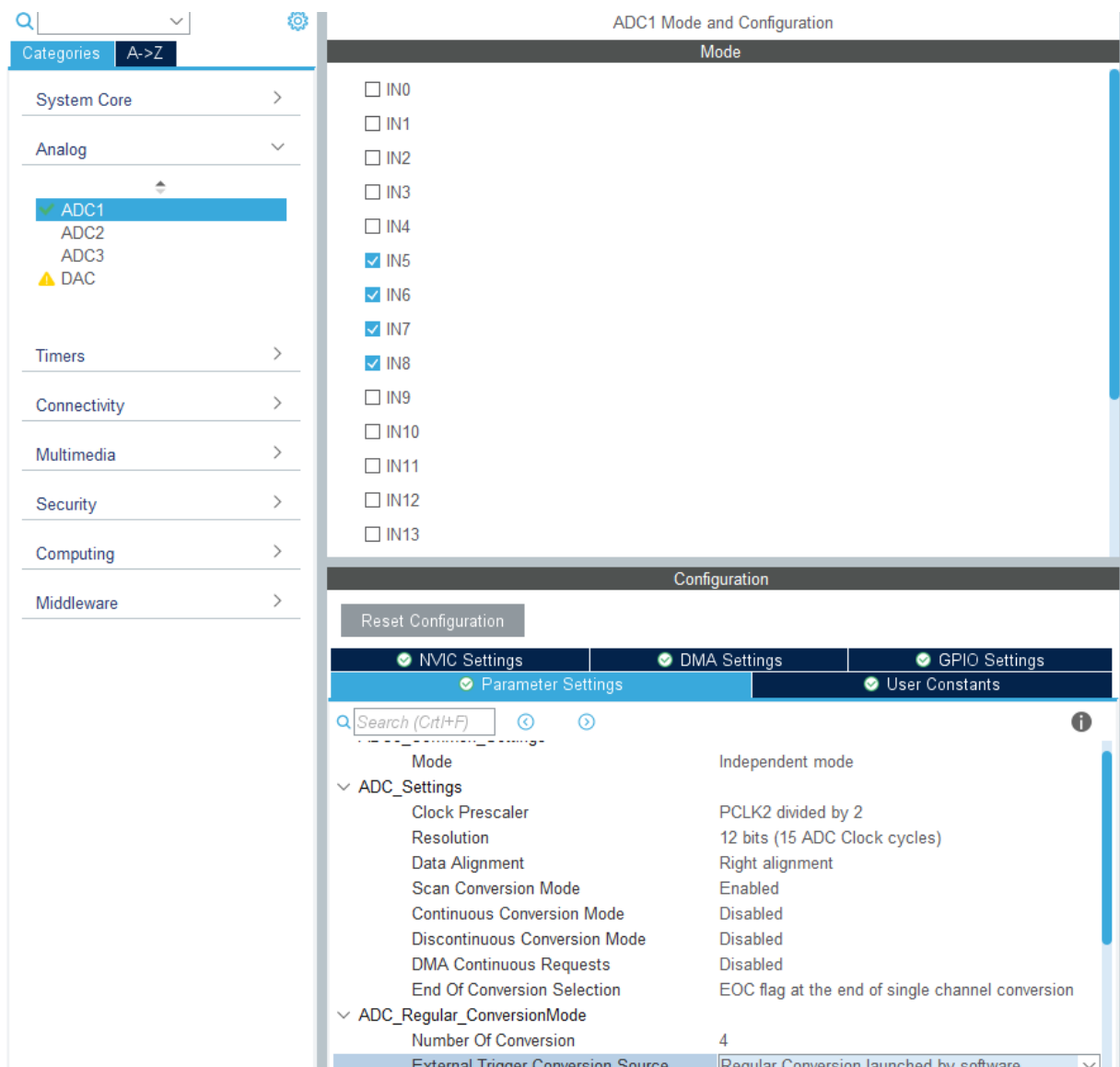


Рисунок 2 – Настройка АЦП в STM32CubeMX

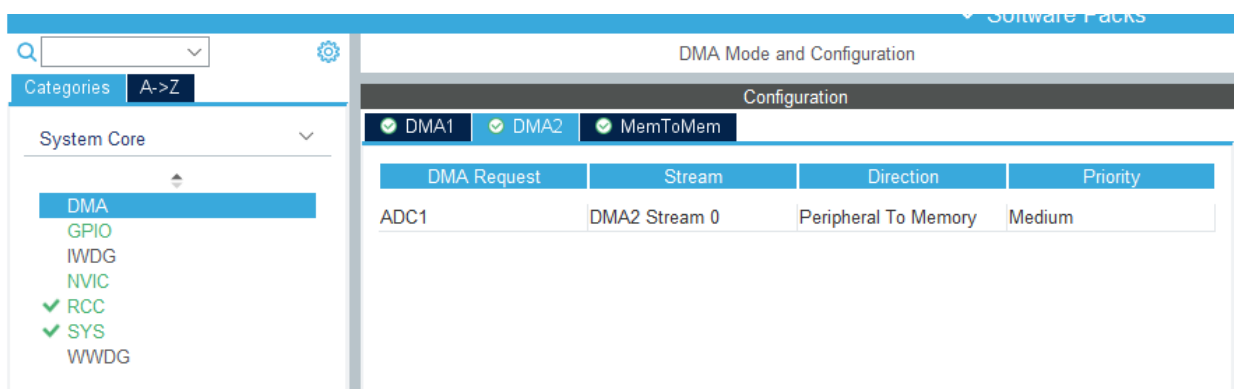


Рисунок 3 – Настройка контроллера DMA в STM32CubeMX

2.3 Код программы на C

Код программы main.c выполнен в среде и CubeIDE for STM32 представлен ниже. На рисунке 4 представлена схема проекта.

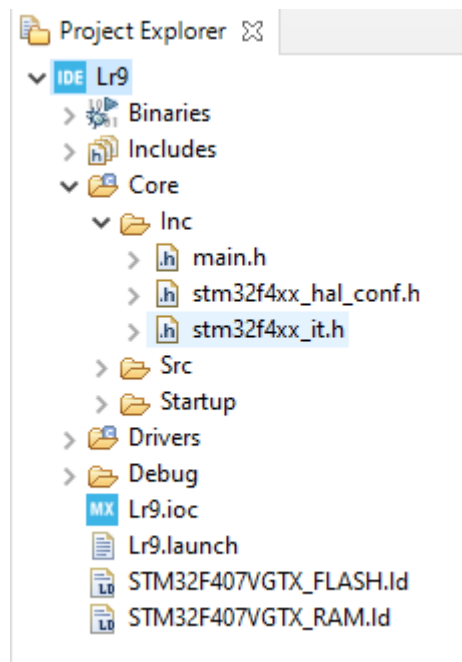


Рисунок 4 – Структурная схема проекта в System Workbench for STM32

Код программы main.c

```
#include "main.h"
```

```
ADC_HandleTypeDef hadc1;  
DMA_HandleTypeDef hdma_adc1;
```

```
void SystemClock_Config(void);  
static void MX_GPIO_Init(void);  
static void MX_DMA_Init(void);  
static void MX_ADC1_Init(void);  
uint16_t ADC_Data[4];  
/* Private user code -----*/  
int main(void)  
{  
    HAL_Init();  
  
    SystemClock_Config();  
  
    MX_DMA_Init();  
  
    MX_ADC1_Init();
```

```
HAL_ADC_Start_DMA(&hadc1,(uint32_t*)&ADC_Data,4); //запись данных с
АЦП по адресу переменной ADC_Data
```

```
while (1)
{
}
}
```

3 Результаты отладки

Expression	Type	Value
ADC_Data	uint16_t [4]	0x200000cc <ADC_Data>
ADC_Data[0]	uint16_t	0
ADC_Data[1]	uint16_t	0
ADC_Data[2]	uint16_t	0
ADC_Data[3]	uint16_t	0
Add new expression		

Рисунок 5 – Окно отладки в момент времени до преобразования АЦП

Expression	Type	Value
ADC_Data	uint16_t [4]	0x200000cc <ADC_Data>
ADC_Data[0]	uint16_t	2044
ADC_Data[1]	uint16_t	3525
ADC_Data[2]	uint16_t	2845
ADC_Data[3]	uint16_t	2340
Add new expression		

Рисунок 6– Окно отладки в момент времени после преобразования АЦП

ВЫВОДЫ

Был реализован прямой доступ к памяти для АЦП. АЦП записывал данные напрямую в память, минуя процессор.

В результате выполнения лабораторной работы были получены навыки:

- работы в программной среде CubeIDE for STM32;
- программирования на языке C.

Была произведена проверка программы на плате STM32F4 Discovery. Исходя из результатов работы платы, можно сделать вывод, что программа работает верно.

Список источников

- 1 Техническая документация по STM32F405xx STM32F407xx/
STMicroelectronics STM32: 2020 – 203 с.
- 2 Reference manual/ STMicroelectronics STM32: 2019 – 1749 с.