Университет ИТМО

Лабораторная работа № 1

по дисциплине «Встроенные системы»

Студент:

P3410 Нгу Фыонг Ань

Преподаватель:

Санкт-Петербург

2020 г.

**Цель работы** – освоение структуры SDK 1.1M и программного доступа к его ресурсам.

**ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**

1. Изучите структуру стенда SDK 1.1M;
2. Скачать репозиторий с проектом по ссылке - https://github.com/lmtspbru/SDK\_cLAB;
3. Открыть проект в STM32CubeIDE;
4. Произвести настройку векторов прерываний и компоновщика;
5. Написать программу с помощью фреймворка HAL, которая должна конфигурировать порты микроконтроллера и производить запись значений по варианту в порт. После записи программа должна производить чтение портов и обеспечивать расчет заданной функции. Записанные биты должны быть выведены в трассировочный буфер после чтения с помощью SDK\_TRACE\_Timestamp. Результат рассчитанной функции должны быть записан в трассировочный буфер SDK\_TRACE\_Print. Номера портов в варианте написаны в том порядке, в котором должны быть записаны биты значения (Пример: порты PB0 PB1, значение 1, тогда в PB0 записывается бит 1, в PB1 бит 0)
6. Собрать проект в Debug режиме;
7. Загрузить сгенерированный бинарный файл на ITMO.cLAB для проверки правильности выполнения задания.
8. По итогам работы написать и защитить отчет. Отчет должен содержать описание теоретической и практической частей, а также основной код программы. В процессе защиты работы требуется демонстрация реализованного функционала, поэтому .bin файл для загрузки на ITMO.cLAB должен быть готов к началу сдачи.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ Варианта** | **Задание** | | | | |
| Порты 1 значения | Порты 2 значения | Значение 1 (X) | Значение 2 (Y) | Функция |
| 2 | PB5, PB0, PB2, PB1 | PB6, PB9, PB7, PB8 | 15 | 4 | (X\*Y)|~Y |

1. **Исходный код**

**int** **main**(**void**)

{

/\* USER CODE BEGIN 1 \*/

/\* USER CODE END 1 \*/

/\* MCU Configuration--------------------------------------------------------\*/

/\* Reset of all peripherals, Initializes the Flash interface and the Systick. \*/

HAL\_Init();

/\* USER CODE BEGIN Init \*/

/\* USER CODE END Init \*/

/\* Configure the system clock \*/

SystemClock\_Config();

/\* USER CODE BEGIN SysInit \*/

/\* USER CODE END SysInit \*/

/\* Initialize all configured peripherals \*/

MX\_GPIO\_Init();

MX\_IWDG\_Init();

/\* USER CODE BEGIN 2 \*/

/\* Do not remove this code below \*/

MX\_TRACE\_Init();

SDK\_TRACE\_Start();

/\* Do not remove this code from above \*/

uint8\_t value\_a = 15;

uint8\_t value\_b = 4;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_5, value\_a & 0x01);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_0, (value\_a >> 1) & 0x01);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_2, (value\_a >> 2) & 0x01);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_1, (value\_a >> 3) & 0x01);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, value\_b & 0x01);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_9, (value\_b >> 1) & 0x01);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_7, (value\_b >> 2) & 0x01);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_8, (value\_b >> 3) & 0x01);

SDK\_TRACE\_Timestamp(P0, HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_5));

SDK\_TRACE\_Timestamp(P1, HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_0));

SDK\_TRACE\_Timestamp(P2, HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_2));

SDK\_TRACE\_Timestamp(P3, HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_1));

SDK\_TRACE\_Timestamp(P4, HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_6));

SDK\_TRACE\_Timestamp(P5, HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_9));

SDK\_TRACE\_Timestamp(P6, HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_7));

SDK\_TRACE\_Timestamp(P7, HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_8));

uint8\_t readed\_value\_a = HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_5);

readed\_value\_a = readed\_value\_a | (HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_0) << 1);

readed\_value\_a = readed\_value\_a | (HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_2) << 2);

readed\_value\_a = readed\_value\_a | (HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_1) << 3);

uint8\_t readed\_value\_b = HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_6);

readed\_value\_b = readed\_value\_b | (HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_9) << 1);

readed\_value\_b = readed\_value\_b | (HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_7) << 2);

readed\_value\_b = readed\_value\_b | (HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_8) << 3);

**int** result = (readed\_value\_a \* readed\_value\_b) | (~ readed\_value\_b);

SDK\_TRACE\_Print("Result (a\*b)|~b: %d", result);

/\* Place your code before here \*/

/\* Do not remove this code below \*/

SDK\_TRACE\_Stop();

/\* Do not remove this code from above \*/

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

**while** (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

}

/\* USER CODE END 3 \*/

}

1. **Результат**



1. **Вывод**

Во время работы в лаборатории я ознакомилась с STM32CubeIDE и написала программу с помощью фреймворка HAL.