

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ
СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ТЕХНІЧНИМИ
СИСТЕМАМИ

Лабораторна робота № 1

Варіант 1

По дисципліні «Програмування мікроконтролерних систем»

Тема: «Основи програмування з використанням бібліотеки Cortex
Microcontroller Software Interface Standard»

Виконали:

студенти групи ІТ-51

Цитовцева А.С.

Бессмертный Р.С.

(підпис, дата)

Перевірив:

ст. викладач кафедри АУТС

Катін П. Ю.

(підпис, дата)

Мета: Провести дослідження програмного включення ПВВЗП на альтернативну функцію для підготовки роботи з більш досконалою периферією.

Хід роботи:

В ході даної лабораторної роботи було узгоджене використання рішення на базі архітектури Cortex-M3. Для програмування мікроконтролерів використовується різні IDE або MDK. Як середу розробки обраний Keil, що є універсальним для різних типів МК і підтримує розробки багатьох виробників. У якості базового МК обрано плату налагодження на базі STM32F103C8

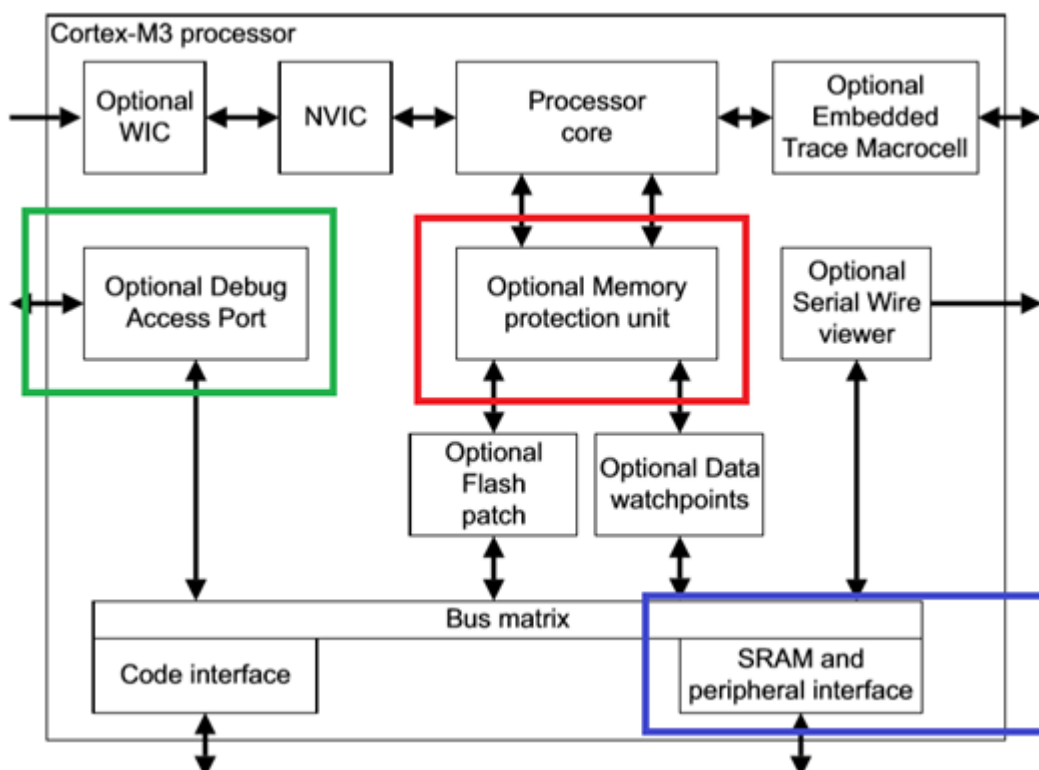


Рисунок 1 – Структура Cortex-M3 [1]

Модель ядра Cortex-M3 включає обчислювальне ядро, систему переривань і матрицю шин для управління загальною системною шиною.

Завдання:

Варіант 2

Розробити і протестувати бібліотеку, що виводить два цілих числа у 4 та 3 сегменті, десятки та одиниці відповідно. Цілі числа розділяють точкою від 2 молодших дрібних розрядів. У цілому розряді виводяться одиниці і десятки, у 2 інших соті частини. Вхідним значенням є тип даних з плаваючою комою, приклад значення що відображається 31.45.

Виконання:

Для налаштування скористуємося кодом і функціями CMSIS

```
GPIO_StructInit (& GPIO_InitStructure);  
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13;  
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
```

```
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz;
GPIO_Init(GPIOC , &GPIO_InitStructure);
```

У даному прикладі відбувається конфігурація апаратного виводу PC13 на вихід, швидкість тактування визначається у 2 МГц. Для функціоналу застосована функція CMSIS запису до відповідного регістру порту.

```
GPIO_WriteBit(GPIOC , GPIO_Pin_13 , (ledval) ? Bit_SET : Bit_RESET);
ledval = 1 - ledval;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
GPIO_Init(GPIOA , &GPIO_InitStructure);
```

```
void usart_init(void)
```

```
{
    /* Enable USART1 and GPIOA clock */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1 |
RCC_APB2Periph_GPIOA, ENABLE);

    /* Configure the GPIOs */
    GPIO_InitTypeDef GPIO_InitStructure;

    /* Configure USART1 Tx (PA.09) as alternate function push-pull */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    /* Configure USART1 Rx (PA.10) as input floating */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    /* Configure the USART1 */
    USART_InitTypeDef USART_InitStructure;

    /* USART1 configuration -----*/
    /* USART1 configured as follow:
        - BaudRate = 115200 baud
        - Word Length = 8 Bits
        - One Stop Bit
        - No parity
        - Hardware flow control disabled (RTS and CTS signals)
        - Receive and transmit enabled
        - USART Clock disabled
        - USART CPOL: Clock is active low
        - USART CPHA: Data is captured on the middle
        - USART LastBit: The clock pulse of the last data bit is not output to
```

```

        the SCLK pin
    */
    USART_InitStructure.USART_BaudRate = 115200;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl =
USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Rx |
USART_Mode_Tx;

    USART_Init(USART1, &USART_InitStructure);

    /* Enable USART1 */
    USART_Cmd(USART1, ENABLE);

    /* Enable the USART1 Receive interrupt: this interrupt is generated when
the
    USART1 receive data register is not empty */
    USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
}

```

Далі для читання значення, що записано до порту використовується функція
 GPIO_ReadInputDataBit(GPIOA , GPIO_Pin_0).

Функція main вихідного коду:

```

int main(void)
{
    float display_number = 31.45;

    int intpart = (int)display_number; //310
    int lefdrop = intpart/100; //3
    intpart = intpart - lefdrop*100; //10

    int decpart = (int)((display_number - intpart)*100); //45

    int num_1 = (int)(intpart/10);
    int num_2 = intpart - num_1*10; //31-30=1
    int num_3 = (int)(decpart/10); //4
    int num_4 = decpart - num_3*10; //45-40=5

    sys_tick_ini();
    TM1637_init();
    TM1637_brightness(BRIGHTTEST);

    while (1) {
        TM1637_display(0,num_1);
        TM1637_display(1,num_2);
        TM1637_display(2,num_3);
        TM1637_display(3,num_4);
    }
}

```

```
}  
}
```

Висновки: Під час виконання лабораторної роботи ми поглибили знання в роботі типового мікроконтролера на базі Cortex-M3, дослідили порт вводу-виводу загального призначення та його включення на альтернативну функцію, а також розібрали можливість відображення даних на LED дисплеях.