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.

Бессмертний Р.С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис, дата)

Перевірив:

ст. викладач кафедри АУТС Катін П. Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис, дата)

Київ-2018

**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

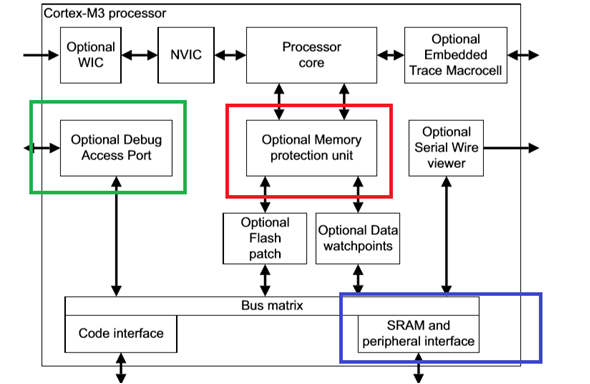


Рисунок 1 – Структура Cortex-M3 [1]

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 РС13 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 leftdrop = intpart/100; //3

intpart = intpart - leftdrop\*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(BRIGHTEST);

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.