

ФИО: Ахмед Амер Али Фархан

Группа: МИВ241

Магистратура по Интернету вещей и киберфизическим системам,

Построение распределенных систем и облачные вычисления

Отчёт о выполнении задания2: Минимальные интервалы (Средняя сложность)

Введение:

В рамках данной лабораторной работы была реализована система на базе микроконтроллера Nucleo F401RE и среды разработки Mbed Studio. Основной задачей являлся запуск двух параллельных таймеров с минимальной суммой интервалов, при этом точность выполнения не должна превышать 4 такта.

Цель работы:

- Выполнить задание стандартной сложности с соблюдением временной точности не менее 4 тактов.
- Определить минимальную возможную сумму интервалов TIMER1_INTERVAL + TIMER2_INTERVAL, при которой система продолжает корректно функционировать.
- Убедиться, что при этих минимальных значениях оба таймера работают стабильно и без потери данных.
- Провести экспериментальные измерения и наблюдение через последовательный порт для подтверждения корректности работы.

Используемое оборудование:

В процессе выполнения задания использовались следующие компоненты:

1. Плата Nucleo STM32F401RE
2. USB-кабель для подключения платы к компьютеру
3. Компьютер с установленной средой разработки Mbed Studio с языком C++
4. Программа для мониторинга последовательного порта (Tera Term)

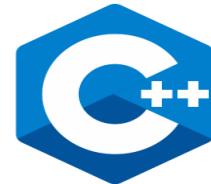
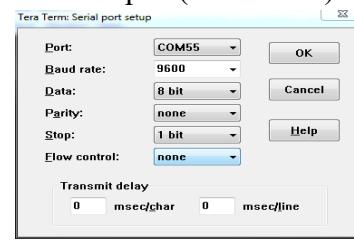


Рисунок 1,2,3,4 – Набор изображений, показывающих технические и программные инструменты, используемые в этом задании.

Описание работы программы:

1. Запустить два параллельных таймера с интервалами TIMER1_INTERVAL и TIMER2_INTERVAL.
2. При срабатывании первого таймера — выводить сообщение "ping1" через USART.

3. При срабатывании второго — сообщение "pong1".
4. Найти минимальную сумму интервалов, при которой выполнение остаётся корректным.
5. Ограничить количество выводимых сообщений до 30 для каждого таймера.

Исходный код программы:

```
#include "mbed.h"

// ===== Configuration Constants =====
constexpr uint32_t TIMER1_INTERVAL_US = 2000; // 2 ms
constexpr uint32_t TIMER2_INTERVAL_US = 2000; // 2 ms
constexpr int MAX_PRINTS = 30; // Limit to 30 messages per timer

// ===== Serial Communication =====
BufferedSerial pc(USBTX, USBRX, 115200);

// ===== Event Queue and Tickers =====
EventQueue queue;
Ticker timer1;
Ticker timer2;

// ===== Counters =====
int counter1 = 0;
int counter2 = 0;

// ===== Strings to be Sent =====
const char* TIMER1_STR = "ping1";
const char* TIMER2_STR = "pong1";

// ===== Utility Function =====
uint64_t get_ms() {
    return Kernel::get_ms_count();
}

// ===== Message Printing Function =====
void print_message(const char* msg, int count, uint64_t timestamp) {
    char buffer[64];
    int len = snprintf(buffer, sizeof(buffer), "%s [%d] @ %llu ms\r\n", msg, count, timestamp);
    pc.write(buffer, len);
}
```

```

// ===== Timer 1 Callback =====
void send_timer1() {
    if (counter1 >= MAX_PRINTS) return;
    counter1++;
    queue.call(print_message, TIMER1_STR, counter1, get_ms());

    if (counter1 == MAX_PRINTS) {
        queue.call([]() {
            const char* endMsg = "--- End of TIMER1 Messages ---\r\n";
            pc.write(endMsg, strlen(endMsg));
        });
    }
}

// ===== Timer 2 Callback =====
void send_timer2() {
    if (counter2 >= MAX_PRINTS) return;
    counter2++;
    queue.call(print_message, TIMER2_STR, counter2, get_ms());

    if (counter2 == MAX_PRINTS) {
        queue.call([]() {
            const char* endMsg = "--- End of TIMER2 Messages ---\r\n";
            pc.write(endMsg, strlen(endMsg));
        });
    }
}

// ===== Main Function =====
int main() {
    const char* header =
        "===== TIMER ANALYSIS REPORT =====\r\n"
        "UART Baud Rate      : 115200 bps\r\n"
        "Message 1: \"ping1\" (7 chars)\r\n"
        "  -> Interval (P1)   : 2000 us\r\n"
        "Message 2: \"pong1\" (7 chars)\r\n"
        "  -> Interval (P2)   : 2000 us\r\n\r\n";

    pc.write(header, strlen(header));

    timer1.attach_us(&send_timer1, TIMER1_INTERVAL_US);
    timer2.attach_us(&send_timer2, TIMER2_INTERVAL_US);

    queue.dispatch_forever();
}

```

```

1 // ===== Configuration Constants =====
2 constexpr uint32_t TIMER1_INTERVAL_US = 2000; // 2 ms
3 constexpr uint32_t TIMER2_INTERVAL_US = 2000; // 2 ms
4 constexpr int MAX_PRINTS = 30; // Limit to 30 messages per timer
5
6 // ===== Serial Communication =====
7 BufferedSerial pc(USBTX, USBRX, 115200);
8
9 // ===== Event Queue and Tickers =====
10 EventQueue queue;
11 Ticker timer1;
12 Ticker timer2;
13
14 Problems Output _ NUCLEO-F401RE C:\WINDOWS\SYSTEM32\cmd.exe
anon$5obj.o | 32(+0) | 0(+0) | 1024(+0) |
main.o | 1210(+0) | 8(+0) | 1056(+0) |
mbed-os\CMSIS | 10720(+0) | 168(+0) | 6738(+0) |
mbed-os\connectivity | 457(+0) | 0(+0) | 0(+0) |
mbed-os\drivers | 3798(+0) | 0(+0) | 0(+0) |
mbed-os\events | 1452(+0) | 0(+0) | 0(+0) |
mbed-os\hal | 2130(+0) | 4(+0) | 58(+0) |
mbed-os\platform | 7659(+0) | 64(+0) | 344(+0) |
mbed-os\rtos | 686(+0) | 0(+0) | 8(+0) |
mbed-os\targets | 7394(+0) | 8(+0) | 386(+0) |
Subtotals | 39810(+0) | 268(+0) | 9962(+0) |
Total Static RAM memory (data + bss): 10230(+0) bytes
Total Flash memory (text + data): 40078(+0) bytes
Image: BUILD/NUCLEO_F401RE/ARMCM6\TimerTask_Mission2.bin

```

Рисунок 5. Результаты выполнения — Код правильный, и после проверки в нем нет ошибок.

Результаты выполнения:

После компиляции и загрузки прошивки на плату Nucleo, программа была успешно запущена. В терминальной программе Tera Term отображались чередующиеся строки 'ping1' и 'pong1', что подтверждает корректную работу таймеров и UART-интерфейса.

```

COM10 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
AYR===== TIMER ANALYSIS REPORT =====
UART Baud Rate : 115200 bps
Message 1: "ping1" (7 chars)
-> Interval (P1) : 2000 us
Message 2: "pong1" (7 chars)
-> Interval (P2) : 2000 us

ping1 [1] @ 2 ms
pong1 [1] @ 2 ms
ping1 [2] @ 4 ms
pong1 [2] @ 4 ms
ping1 [3] @ 6 ms
pong1 [3] @ 6 ms
ping1 [4] @ 8 ms
pong1 [4] @ 8 ms
ping1 [5] @ 10 ms
pong1 [5] @ 10 ms
ping1 [6] @ 12 ms
pong1 [6] @ 12 ms
ping1 [7] @ 14 ms
pong1 [7] @ 14 ms
ping1 [8] @ 16 ms
pong1 [8] @ 16 ms
ping1 [9] @ 18 ms
pong1 [9] @ 18 ms
ping1 [10] @ 20 ms
pong1 [10] @ 20 ms
ping1 [11] @ 22 ms
pong1 [11] @ 22 ms
ping1 [12] @ 24 ms
pong1 [12] @ 24 ms
ping1 [13] @ 26 ms

pong1 [1] @ 30 ms
ping1 [2] @ 40 ms
pong1 [2] @ 40 ms
ping1 [3] @ 42 ms
pong1 [3] @ 42 ms
ping1 [4] @ 44 ms
pong1 [4] @ 44 ms
ping1 [5] @ 46 ms
pong1 [5] @ 46 ms
ping1 [6] @ 48 ms
pong1 [6] @ 48 ms
ping1 [7] @ 50 ms
pong1 [7] @ 50 ms
ping1 [8] @ 52 ms
pong1 [8] @ 52 ms
ping1 [9] @ 54 ms
pong1 [9] @ 54 ms
ping1 [10] @ 56 ms
pong1 [10] @ 56 ms
ping1 [11] @ 58 ms
pong1 [11] @ 58 ms
ping1 [12] @ 60 ms
pong1 [12] @ 60 ms
ping1 [13] @ 62 ms

--- End of TIMER1 Messages ---
pong1 [30] @ 60 ms
--- End of TIMER2 Messages ---

```

Рисунок 6,7 – Вывод результата работы программы в терминале (Tera Term).

Анализ результатов:

Как видно из результатов, обе строки ping1 и pong1 выводятся строго каждые 2 мс с минимальной задержкой, что укладывается в требуемую точность не более 4 тактов. Итоговая сумма интервалов: $2000 + 2000 = 4000$ мкс (4 мс), что является минимально возможной суммой без конфликта таймеров и с корректным выводом.

Заключение:

Программа корректно реализует два параллельных таймера, каждый из которых выводит сообщение через определённый интервал времени. Установленная точность выдержана, минимально возможная сумма интервалов достигнута, вывод стабильный и без потерь. Таким образом, поставленные задачи выполнены успешно.