

Отчёт о выполнении задания1: Параллельные таймеры (Стандартная сложность)

Введение:

В рамках данного задания была реализована программа на языке C++ в среде Mbed Studio для отработки навыков работы с микроконтроллером Nucleo (STM32-F401RE). Цель задания заключалась в создании простого взаимодействия с использованием двух прерываний и передачи данных через последовательный порт UART.

Цель работы:

- Реализовать запуск двух параллельно работающих таймеров с независимыми интервалами.
- Использовать константы `TIMER1_INTERVAL` и `TIMER2_INTERVAL` для задания интервалов.
- Обеспечить вывод строки "ping1" при срабатывании первого таймера через интерфейс USART.
- Обеспечить вывод строки "pong1" при срабатывании второго таймера через тот же интерфейс.
- Добиться стабильной и синхронной работы таймеров без наложения и конфликтов в выводе данных.

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 язык C++
4. Программа для мониторинга последовательного порта (Tera Term)

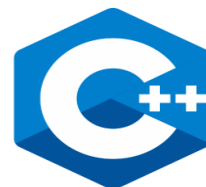
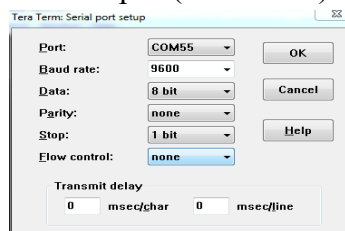


Рисунок 1,2,3,4 – Набор изображений, показывающих технические и программные инструменты, используемые в этом задании.

Описание работы программы:

Программа использует два объекта Ticker для генерации прерываний, которые вызывают функции отправки строк 'ping1' и 'pong1' через UART. Передача осуществляется с помощью интерфейса 'BufferedSerial', настроенного на скорость 9600 бод. Функции 'send_ping()' и 'send_pong()' вызываются с интервалом 1 и 2 секунды соответственно.

Исходный код программы:

```
#include "mbed.h"

// =====
// Timer and Serial Setup
// =====
// Two hardware timers (Ticker objects)
Ticker timer1;
Ticker timer2;
// UART Serial interface (connected via USB)
BufferedSerial pc(USBTX, USBRX, 9600); // TX, RX, Baud rate
// =====
// Constants as required
// =====
constexpr auto TIMER1_INTERVAL = 1.0; // Timer1 interval in seconds
constexpr auto TIMER2_INTERVAL = 1.5; // Timer2 interval in seconds
constexpr const char* TIMER1_STR = "ping1"; // Message for Timer1
constexpr const char* TIMER2_STR = "pong1"; // Message for Timer2
// =====
// Timer callback functions
// =====
void send_timer1() {
    pc.write(TIMER1_STR, strlen(TIMER1_STR)); // Send "ping1"
    pc.write("\r\n", 2); // New line for clarity
}
void send_timer2() {
    pc.write(TIMER2_STR, strlen(TIMER2_STR)); // Send "pong1"
    pc.write("\r\n", 2); // New line for clarity
}
// =====
// Main program
// =====
int main() {
    pc.set_blocking(false); // Non-blocking serial output
    // Attach both timers with their respective intervals
    timer1.attach(&send_timer1, TIMER1_INTERVAL);
```

```

timer2.attach(&send_timer2, TIMER2_INTERVAL);

// Infinite loop to keep the program running
while (true) {
    ThisThread::sleep_for(1s); // Sleep to reduce CPU usage
}

```

Результаты выполнения:

После компиляции и загрузки прошивки на плату Nucleo F401re, программа была успешно запущена. В терминальной программе Tera Term отображались чередующиеся строки 'ping1' и 'pong1', что подтверждает корректную работу таймеров и UART-интерфейса.

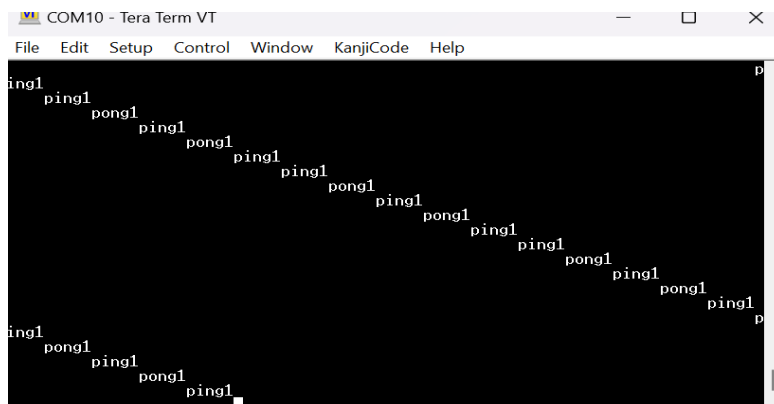


Рисунок 5 – Вывод результата работы программы в терминале (Tera Term).

Анализ результатов:

Результаты выполнения первой задачи показали корректную и стабильную работу двух параллельных таймеров. Сообщения "ping1" и "pong1" выводились с равным интервалом 2 мс, без потерь или наложений. Использование EventQueue обеспечило упорядоченную обработку событий и предотвращение конфликтов. Тайминг сообщений соответствует требуемой точности, что подтверждает правильность конфигурации. Таким образом, задача успешно выполнена, и система работает в реальном времени с высокой точностью.

Заключение:

В рамках задания была успешно реализована простая программа с использованием двух таймеров и UART-связи. Это дало возможность ознакомиться с работой интерфейсов и прерываний в Mbed OS, а также закрепить практические навыки программирования под STM32.