Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«ТОМСКИЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И

РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

РАБОТА С ПОТОКАМИ В MBED OS

Отчет по лабораторной работе №5

по дисциплине «Системное программирование»

Студенты гр. 718-1

\_\_\_\_\_\_\_\_ Прозорова Е. А.

\_\_\_\_\_\_\_\_ Новокрещенных В. И.

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2022

Принял

М.н.с. ИСИБ

\_\_\_\_\_\_\_\_ Калинин Е. О.

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2022

Томск 2022

1 Введение

Цель работы: Получение навыков работы с потоками в Mbed OS.

2 Ход работы

По заданию лабораторной работы была написана программа, реализующая вывод текста за определенное время через поток с помощью флага (рисунок 2.1).

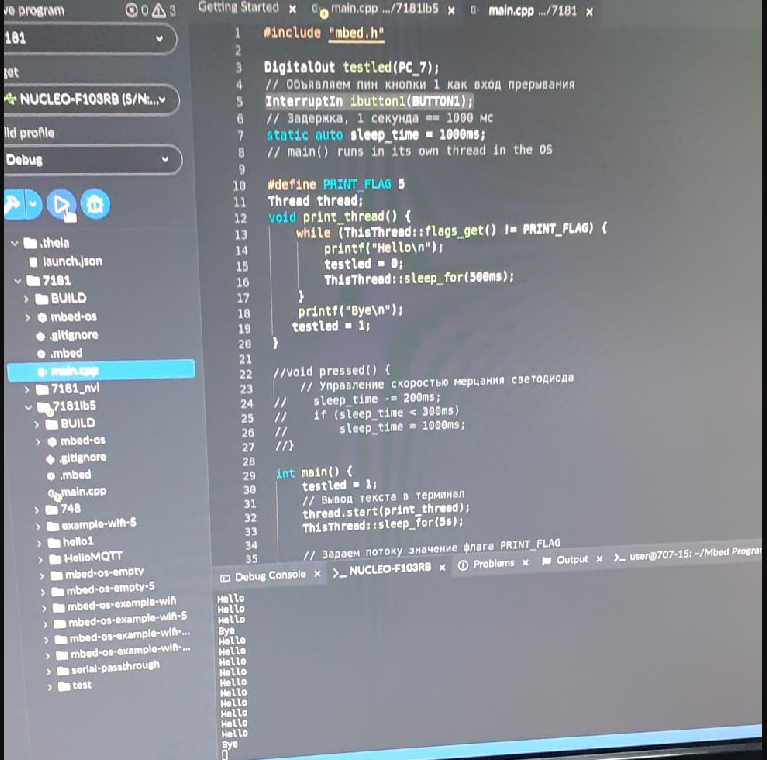


Рисунок 2.1 – Вывод текста за определенное время

Также, была написана программа реализующая постоянный поток, через который было реализовано мигание диода (рисунки 2.2 – 2.3).

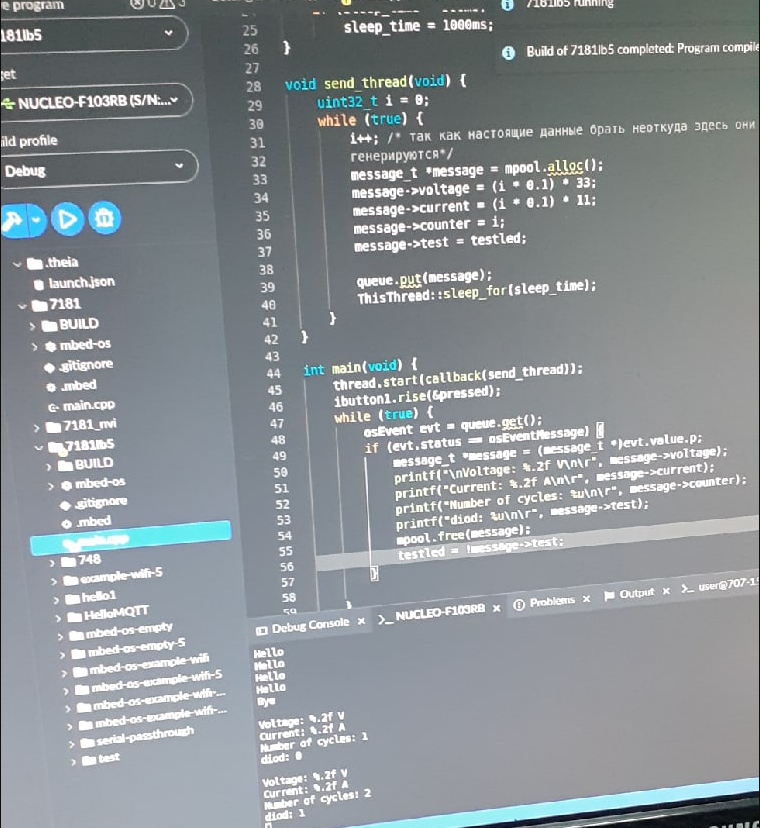


Рисунок 2.2 – Программа постоянного потока

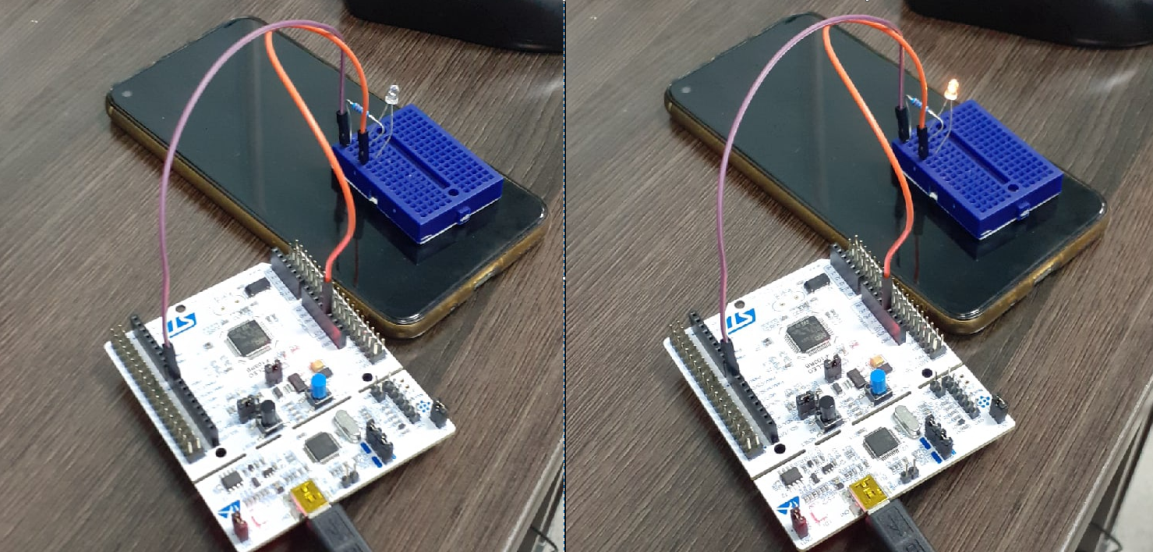


Рисунок 2.3 - Мигание диода

3 Заключение

В рамках выполнения лабораторной работы были получены навыки работы с потоками в Mbed OS. Была написана программа, реализующия мигание светодиода.

Приложение А

(Код программы)

#include "mbed.h"

DigitalOut testled(PC\_7);

// Объявляем пин кнопки 1 как вход прерывания

InterruptIn ibutton1(BUTTON1);

// Задержка, 1 секунда == 1000 мс

static auto sleep\_time = 1000ms;

// main() runs in its own thread in the OS

#define PRINT\_FLAG 5

Thread thread;

void print\_thread() {

while (ThisThread::flags\_get() != PRINT\_FLAG) {

printf("Hello\n");

testled = 0;

ThisThread::sleep\_for(500ms);

}

printf("Bye\n");

testled = 1;

}

//void pressed() {

// Управление скоростью мерцания светодиода

// sleep\_time -= 200ms;

// if (sleep\_time < 300ms)

// sleep\_time = 1000ms;

//}

int main() {

testled = 1;

// Вывод текста в терминал

thread.start(print\_thread);

ThisThread::sleep\_for(5s);

// Задаем потоку значение флага PRINT\_FLAG

thread.flags\_set(PRINT\_FLAG);

// Прикрепляем функцию для обработки нажатия

//ibutton1.rise(&pressed);

//while (true) {

// testled = !testled;

// ThisThread::sleep\_for(sleep\_time);

//}

}

#include "mbed.h"

DigitalOut testled(PC\_7);

InterruptIn ibutton1(BUTTON1);

static auto sleep\_time = 1000ms;

typedef struct {

float voltage; /\* результат измерения напряжения \*/

float current; /\* результат измерения тока\*/

uint32\_t counter;

int test;

} message\_t;

/\*здесь используется объект класса MemoryPool, для определения и

управления пулом памяти фиксированного размера, информацию об этом

классе изучите в документации \*/

MemoryPool<message\_t, 16> mpool;

Queue<message\_t, 16> queue;

Thread thread;

/\* поток отправитель \*/

void pressed() {

// Управление скоростью мерцания светодиода

sleep\_time -= 200ms;

if (sleep\_time < 300ms)

sleep\_time = 1000ms;

}

void send\_thread(void) {

uint32\_t i = 0;

while (true) {

i++; /\* так как настоящие данные брать неоткуда здесь они

генерируются\*/

message\_t \*message = mpool.alloc();

message->voltage = (i \* 0.1) \* 33;

message->current = (i \* 0.1) \* 11;

message->counter = i;

message->test = testled;

queue.put(message);

ThisThread::sleep\_for(sleep\_time);

}

}

int main(void) {

thread.start(callback(send\_thread));

ibutton1.rise(&pressed);

while (true) {

osEvent evt = queue.get();

if (evt.status == osEventMessage) {

message\_t \*message = (message\_t \*)evt.value.p;

printf("\nVoltage: %.2f V\n\r", message->voltage);

printf("Current: %.2f A\n\r", message->current);

printf("Number of cycles: %u\n\r", message->counter);

printf("diod: %u\n\r", message->test);

mpool.free(message);

testled = !message->test;

}

}

}

Приложение Б

(Ссылка на репозиторий)

Ссылки на репозитории gitlab:

https://github.com/SlavaNovok/SystemProgram.git

https://github.com/ledastro/SystemProgramm.git