Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Программное обеспечение встроенных систем (ПОВС)

ОТЧЕТ

по лабораторной работе № 2

Тема работы: Построение графика плотности вероятности СВ

Выполнили

студенты: гр. 151003 Барановский Р.А.,

Матошко И.В.

Проверил: Деменковец Д.В.

Минск 2024

**Вариант 3**

**lib.h**

#ifndef LIB\_DECLARATION

#define LIB\_DECLARATION

#include "stm32f1xx\_hal.h"

#define CORRECTION\_TIME 500

extern volatile int previousUnPressedTime;

extern unsigned int seed;

void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin);

int random\_number(int min, int max);

#endif // LIB\_DECLARATION

**lib.c**

#include "lib.h"

#include "main.h"

#include "display.h"

volatile int previousUnPressedTime = 0;

unsigned int seed = 12345;

unsigned int lcg() {

seed = (1103515245 \* seed + 12345) % 2147483648;

return seed;

}

int random\_number(int min, int max) {

return (lcg() % (max - min + 1)) + min;

}

uint8\_t data = 0;

Buffer displayBuffer = 0;

void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)

{

if(GPIO\_Pin == GPIO\_PIN\_1 && (HAL\_GetTick() - previousUnPressedTime) > CORRECTION\_TIME)

{

data = (uint8\_t) random\_number(1, 6);

HAL\_UART\_Transmit\_IT(&huart2, &data, sizeof(uint8\_t));

previousUnPressedTime = HAL\_GetTick();

pushNumberIntoBuffer(&displayBuffer, data);

printNumberOnDisplay(&displayBuffer);

}

else if (GPIO\_Pin == GPIO\_PIN\_4 && (HAL\_GetTick() - previousUnPressedTime) > CORRECTION\_TIME)

{

data = 0;

HAL\_UART\_Transmit\_IT(&huart2, &data, sizeof(uint8\_t));

previousUnPressedTime = HAL\_GetTick();

pushNumberIntoBuffer(&displayBuffer, data);

printNumberOnDisplay(&displayBuffer);

}

}

**display.h**

#ifndef \_\_DISPLAY\_H

#define \_\_DISPLAY\_H

#include "stm32f1xx\_hal.h"

#include "main.h"

typedef uint16\_t Buffer;

void pushNumberIntoBuffer(Buffer \*buffer, uint8\_t number);

void printNumberOnDisplay(const Buffer \*buffer);

#endif

**display.c**

#include "display.h"

const uint8\_t digits[] = {

0xC0, //0

0xF9, //1

0xA4, //2

0xB0, //3

0x99, //4

0x92, //5

0x82, //6

0xF8, //7

0x80, //8

0x90 //9

};

const uint8\_t segments[] = {

0xF1,

0xF2,

0xF4,

0xF8

};

uint8\_t getDigitFromBuffer(const Buffer \*buffer, int index)

{

return (\*buffer >> (index \* 4)) & 0x000F;

}

void pushDigitIntoBuffer(Buffer \*buffer, int index, uint8\_t digit)

{

\*buffer |= (digit & 0x000F) << (index \* 4);

}

void shiftRegister(uint8\_t digitOrSegment)

{

for (int i = 0; i < 8; i++)

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, digitOrSegment & 0x80 ? GPIO\_PIN\_SET : GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_8, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_8, GPIO\_PIN\_SET);

digitOrSegment <<= 1;

}

}

void printDigitOnDisplay(uint8\_t segment, uint8\_t digit)

{

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_5, GPIO\_PIN\_RESET);

shiftRegister(digits[digit]);

shiftRegister(segments[segment]);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_5, GPIO\_PIN\_SET);

}

void printNumberOnDisplay(const Buffer \*buffer)

{

for (int i = 0; i <= 3; i++)

{

printDigitOnDisplay(i, getDigitFromBuffer(buffer, i));

}

}

void pushNumberIntoBuffer(Buffer \*buffer, uint8\_t number)

{

\*buffer = 0;

for (int i = 3; i >= 0; i--)

{

pushDigitIntoBuffer(buffer, i, number % 10);

number /= 10;

}

}