Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Институт «Электронных и информационных систем»

Кафедра «Информационных технология и систем»

Лабораторная работа №4

по дисциплине:

« ЭВМ и периферийные устройства»

**Отчёт**

Принял преподаватель:

\_\_\_\_\_\_\_\_\_\_ / Шаклеин В.Г

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Выполнил студент группы 9091:

\_\_\_\_\_\_\_\_\_\_ / Чалый С. М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

**Великий Новгород**

**2022**

Цель работы:

Получить практические навыки работы с прерываниями.

Ход работы:

На основе лабораторной работы UART сделать прошивку микроконтроллера , выводящую строку , копирующую входные строки в выходной UART поток, работая с помощью механизма прерываний.

**#define** MY\_UART USART3//именнованная константа для уменьшения переделок

**#define** TX\_PORT GPIOC

**#define** TX\_PIN 10//ноги TX

**#define** RX\_PORT GPIOC

**#define** RX\_PIN 11 //ноги RX

**char** bufToUART[128];

**char** bufFromUART[128];

**typedef** **struct**{

uint32\_t wrPtr;

uint32\_t rdPtr;

uint32\_t size;

**char**\* ptr;

}RingBufStruct;

RingBufStruct ringToUart={0,0,**sizeof**(bufToUART),bufToUART};

RingBufStruct ringFromUart={0,0,**sizeof**(bufFromUART),bufFromUART};

**int** rxCnt=0;

**int** **RingGetLen**(RingBufStruct\* ring)//размер очереди

{

**return** ring->wrPtr- ring->rdPtr;

}

**void** **RingInsert**(RingBufStruct\* ring, **char** ch)//добавление символа

{

**if** (RingGetLen(ring)>=ring->size)

{

**return**;

}

ring->ptr[(ring->wrPtr++)%ring->size]=ch;//берем указатель и закольцовываем остатком от деления

}

**char** **RingGet**(RingBufStruct\* ring)//получение элемента

{

**if**(RingGetLen(ring)==0)//если очередь пустая, возвращаем ноль

{

**return** 0;

}

**return** ring->ptr[(ring->rdPtr++)%ring->size];//сдвинули

}

/\* USER CODE END PV \*/

/\* Private function prototypes -----------------------------------------------\*/

**void** **SystemClock\_Config**(**void**);

/\* USER CODE BEGIN PFP \*/

/\* USER CODE END PFP \*/

/\* Private user code ---------------------------------------------------------\*/

/\* USER CODE BEGIN 0 \*/

**void** **USART3\_IRQHandler**(**void**)

{

**if**(MY\_UART->ISR & USART\_ISR\_RXNE\_RXFNE)

{

**char** ch=MY\_UART->RDR;

RingInsert(&ringFromUart,ch);

**if**(ch== '\r')

{

rxCnt += 1;

}

}

**if**(MY\_UART->ISR & USART\_ISR\_TXE\_TXFNF)//если есть куда отправить, но нету чего отправить

{

**if**(RingGetLen(&ringToUart)==0)

{

BIT\_BAND\_PER(MY\_UART->CR1,USART\_CR1\_TXEIE\_TXFNFIE) = 0;//то отключаем прерывание

}**else**

{

MY\_UART->TDR=RingGet(&ringToUart);//или положили данные и тоже вышли из прерывания

}

}

}

**void** **UartSendString**(**const** **char**\* txt)

{

**while**(\*txt != 0)

{

RingInsert(&ringToUart, \*txt);

}

BIT\_BAND\_PER(MY\_UART->CR1,USART\_CR1\_TXEIE\_TXFNFIE) = 1;

}

RCC->AHB2ENR |= RCC\_AHB2ENR\_GPIOCEN;//тактирование GPIO

RCC->APB1ENR1 |= RCC\_APB1ENR1\_USART3EN;//тактирование USART

TX\_PORT->MODER &= ~(3<<(2\*TX\_PIN));//альтернативный режим для tx

TX\_PORT->MODER |= (2<<(2\*TX\_PIN));

TX\_PORT->OSPEEDR &= ~(3<<(2\*TX\_PIN));//включаем побыстрее TX

TX\_PORT->OSPEEDR |= (3<<(2\*TX\_PIN));

//включаем 7ю альтернативу для ноги

**if**(TX\_PIN>=8){//если ножка больше 8, то записываем в старший[1] AFR регистр в нулевой бит

TX\_PORT->AFR[1] &= ~(0x0f<<(4\*(TX\_PIN-8)));//0х0f - 4бита

TX\_PORT->AFR[1] |= (0x07<<(4\*(TX\_PIN-8)));

}**else**{

TX\_PORT->AFR[0] &= ~(0x0f<<(4\*(TX\_PIN)));

TX\_PORT->AFR[0] |= (0x07<<(4\*(TX\_PIN)));

}

RX\_PORT->MODER &= ~(3<<(2\*RX\_PIN));//альтернативный режим для rx

RX\_PORT->MODER |= (2<<(2\*RX\_PIN));

**if**(RX\_PIN>=8){

RX\_PORT->AFR[1] &= ~(0x0f<<(4\*(RX\_PIN-8)));

RX\_PORT->AFR[1] |= (0x07<<(4\*(RX\_PIN-8)));

}**else**{

RX\_PORT->AFR[0] &= ~(0x0f<<(4\*(RX\_PIN)));

RX\_PORT->AFR[0] |= (0x07<<(4\*(RX\_PIN)));

}

MY\_UART->CR1 &= USART\_CR1\_UE;

MY\_UART->CR1 = USART\_CR1\_FIFOEN | USART\_CR1\_TE | USART\_CR1\_RE | USART\_CR1\_TXEIE\_TXFNFIE | USART\_CR1\_RXNEIE\_RXFNEIE;

MY\_UART->CR2 = 0;

MY\_UART->CR3=USART\_CR3\_OVRDIS

MY\_UART->BRR=184;//USART3 в режим 170/921000

MY\_UART->CR1 |= USART\_CR1\_UE;

NVIC\_EnableIRQ (*USART3\_IRQn*);

/\* USER CODE END SysInit \*/

/\* Initialize all configured peripherals \*/

/\* USER CODE BEGIN 2 \*/

UartSendString("Work is not a wolf, work is a worker\r\n");//выдает строку при старте