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 №3

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 базовые навыки работы с UART в простейшем режиме.

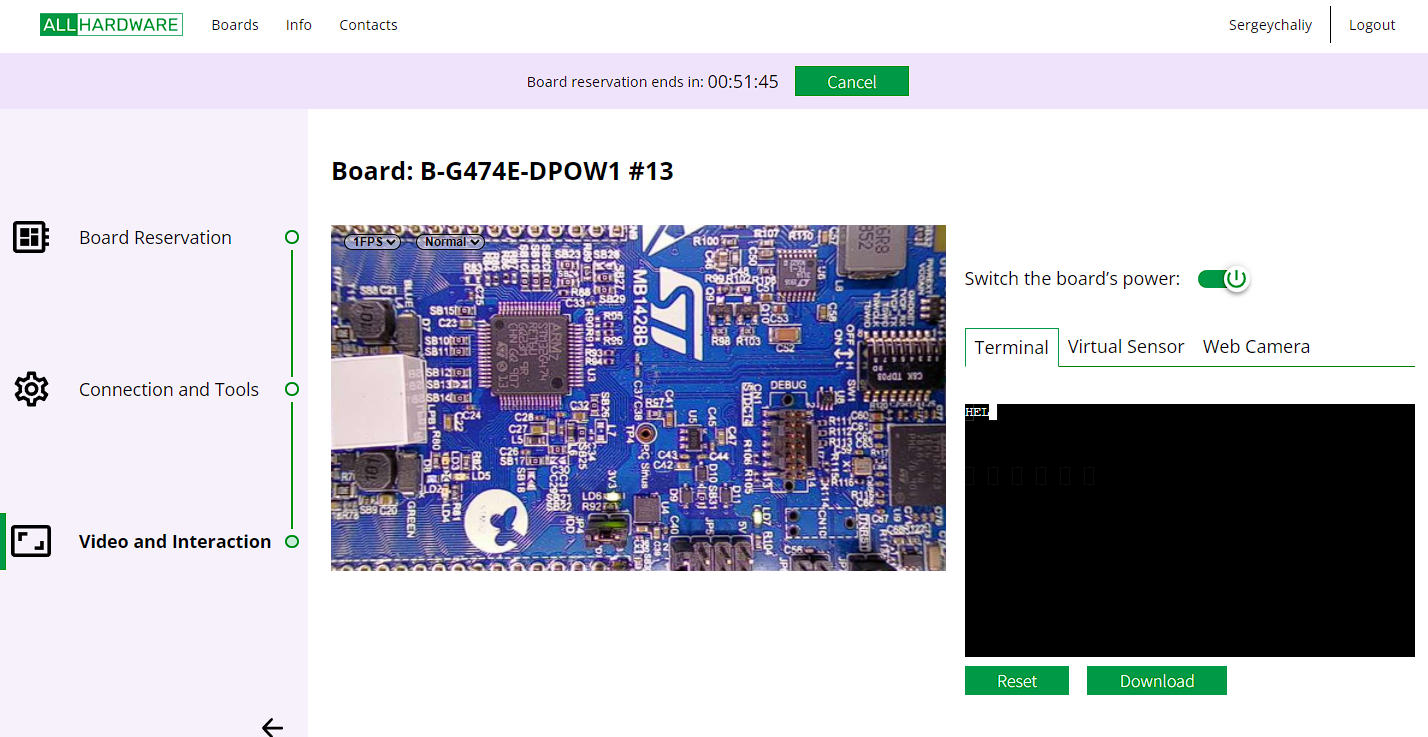
Ход работы:

1. Настроить USART3 в режим YART 8, N, 1 921000
2. Сделать чтобы прошивка при старте выдавала UART приветственную строку “Hello my friend”
3. Добавить функционал, чтобы прошивка принимала строку и посылала её назад в терминал

Задание 1:

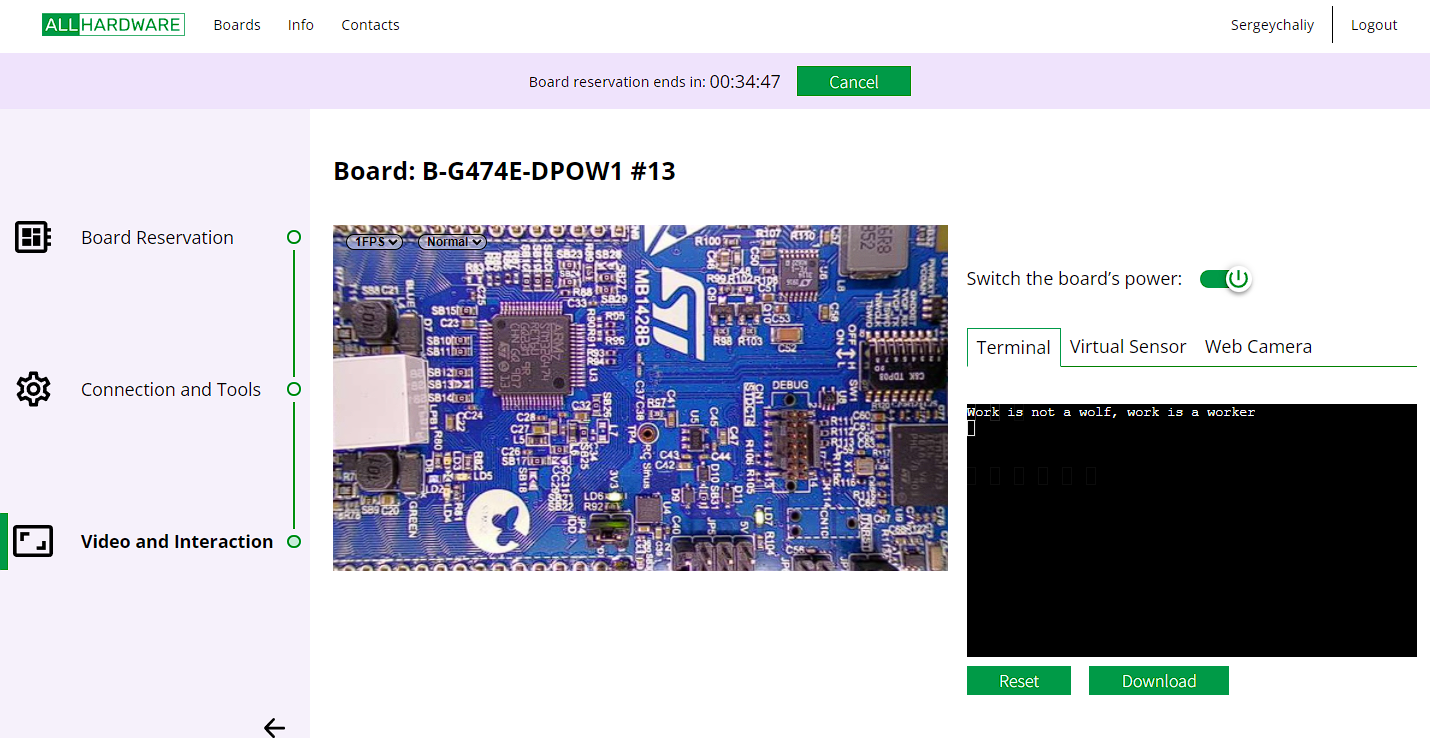
Настроил USART3 в режим YART 8, N, 1 921000.

!Надпись в терминале стиралась, т.к было много отладок для нормального запуска без ошибок.



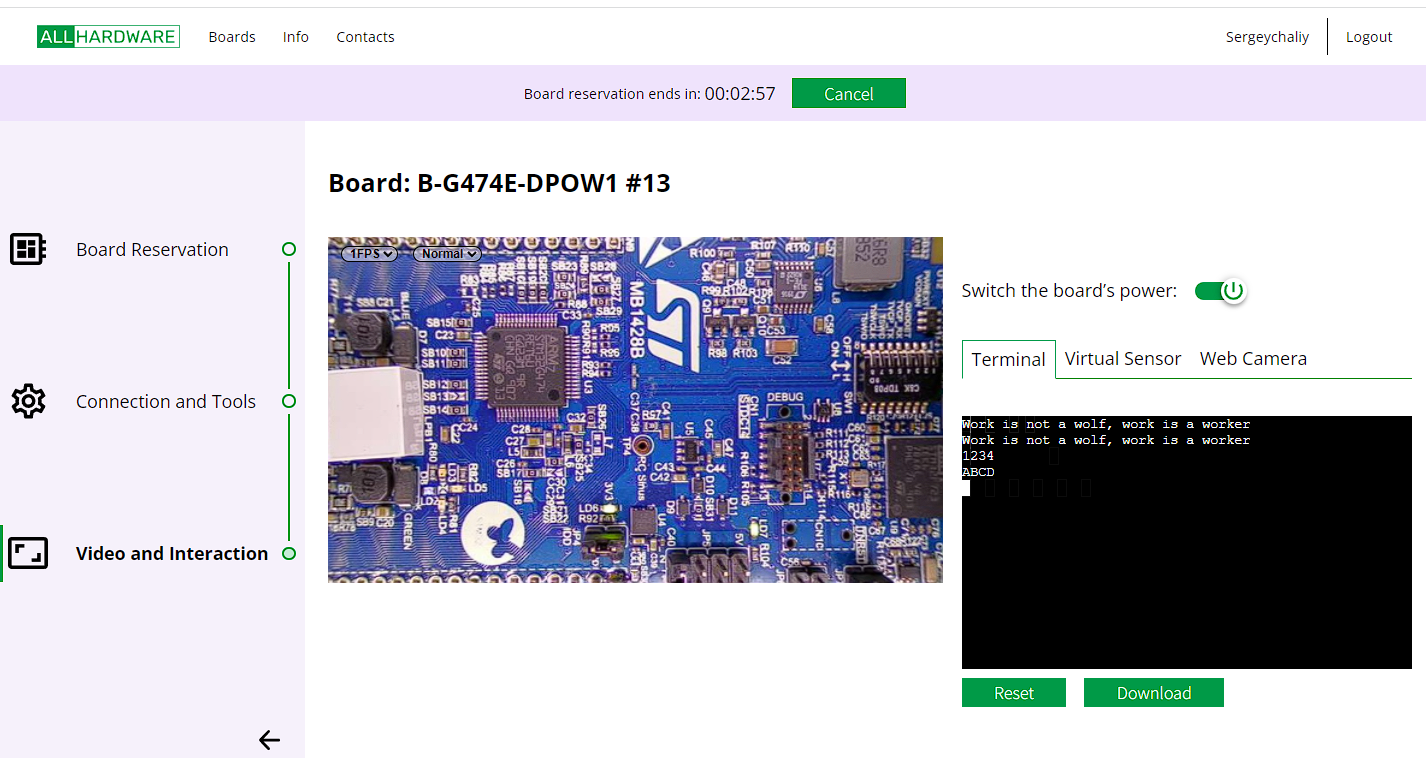
Задание 2:

Добавить приветственную строку при старте.



Задание 3:

Добавил функционал, чтобы прошивка принимала строку и посылала её назад в терминал



/\* USER CODE BEGIN PTD \*/

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

**#define** TX\_PORT GPIOC

**#define** TX\_PIN 10//н оги TX

**#define** RX\_PORT GPIOC

**#define** RX\_PIN 11//ноги RX

/\* USER CODE END PTD \*/

/\* Private define ------------------------------------------------------------\*/

/\* USER CODE BEGIN PD \*/

/\* USER CODE END PD \*/

/\* Private macro -------------------------------------------------------------\*/

/\* USER CODE BEGIN PM \*/

/\* USER CODE END PM \*/

/\* Private variables ---------------------------------------------------------\*/

/\* USER CODE BEGIN PV \*/

/\* 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** **UartSendChar**(**char** ch){

**while**((MY\_UART->ISR & USART\_ISR\_TXE\_TXFNF)==0){

}

MY\_UART->TDR=ch;

}

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

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

UartSendChar(\*txt++);

}

}

**char** **UartGetChar**(**void**){

**while**((MY\_UART->ISR & USART\_ISR\_RXNE\_RXFNE)==0){

}

**return** MY\_UART->RDR;

}

**int** **UartReadString**(**char**\* txt, **int** maxLen){

**int** pos = 0;

**while**(pos<maxLen-1){

txt[pos]=UartGetChar();

**if**(txt[pos]<' '){

txt[pos++]=0;

**return** pos;

}

pos+=1;

}

txt[pos++]=0;

**return** pos;

}

/\* USER CODE END 0 \*/

/\*\*

\* @brief The application entry point.

\* @retval int

\*/

**int** **main**(**void**)

{

/\* USER CODE BEGIN 1 \*/

/\* USER CODE END 1 \*/

/\* MCU Configuration--------------------------------------------------------\*/

/\* Reset of all peripherals, Initializes the Flash interface and the Systick. \*/

HAL\_Init();

/\* USER CODE BEGIN Init \*/

/\* USER CODE END Init \*/

/\* Configure the system clock \*/

SystemClock\_Config();

/\* USER CODE BEGIN SysInit \*/

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;

MY\_UART->CR2 = 0;

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

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

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

/\* USER CODE END SysInit \*/

/\* Initialize all configured peripherals \*/

/\* USER CODE BEGIN 2 \*/

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

**while** (1)

{

/\* USER CODE END WHILE \*/

**char** txt[100];//взводим строку

UartReadString(txt,**sizeof**(txt));//читаем строку

UartSendString(txt);//отправляем

UartSendString("\r\n");

/\* USER CODE BEGIN 3 \*/

}

Вывод:

Таким образом, я приобрел навыки работы с UART и прошивкой в простейшем режиме .