

## Лабораторная работа № 4: Настройка UART интерфейса в микроконтроллере STM32.

**Цель:** научиться настраивать UART интерфейс в микроконтроллере STM32 для передачи данных через COM порт компьютера.

**Оборудование и программное обеспечение:** плата STM32F4 Discovery, среда разработки Coocox ColIDE 1.7.

### Теоретический материал

UART интерфейс является последовательным интерфейсом, который используется для передачи и приема данных в виде последовательных битов между устройствами. В качестве сигналов передачи данных используются две линии: TX (Transmit) и RX (Receive). TX используется для передачи данных с одного устройства на другое, а RX - для приема данных. Обычно для работы с UART микроконтроллер STM32 используются специальные периферийные модули USART, которые позволяют передавать данные с приемопередатчиком.

Универсальный синхронный/асинхронный приёмопередатчик (УАПП, англ. Universal Asynchronous Receiver-Transmitter, UART) — узел вычислительных устройств, предназначенный для организации связи с другими цифровыми устройствами. Преобразует передаваемые данные в последовательный вид так, чтобы было возможно передать их по цифровой линии другому аналогичному устройству. Метод преобразования хорошо стандартизован и широко применялся в компьютерной технике.

Представляет собой логическую схему, с одной стороны подключённую к шине вычислительного устройства, а с другой имеющую два или более выводов для внешнего соединения.

Пример кода для инициализации UART интерфейса в микроконтроллере STM32 на языке программирования C:

```
#include "stm32f4xx.h"

// Функция инициализации UART
void UART_Init(void)
{
    // Инициализация портов для работы с подключенным UART устройством
    GPIO_InitTypeDef GPIO_InitStructure;
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_7;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
```

```

GPIO_Init(GPIOB, &GPIO_InitStruct);

// Настройка альтернативной функции на пинах GPIOB6 и GPIOB7
GPIO_PinAFConfig(GPIOB, GPIO_PinSource6, GPIO_AF_USART1);
GPIO_PinAFConfig(GPIOB, GPIO_PinSource7, GPIO_AF_USART1);

// Инициализация модуля USART1
USART_InitTypeDef USART_InitStruct;
RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);
USART_InitStruct.USART_BaudRate = 9600;
USART_InitStruct.USART_WordLength = USART_WordLength_8b;
USART_InitStruct.USART_StopBits = USART_StopBits_1;
USART_InitStruct.USART_Parity = USART_Parity_No;
USART_InitStruct.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
USART_InitStruct.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
USART_Init(USART1, &USART_InitStruct);

// Включение прерываний при приеме и передаче данных
USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
USART_ITConfig(USART1, USART_IT_TXE, ENABLE);

// Включение модуля USART1
USART_Cmd(USART1, ENABLE);
}

```

Этот код выполнен для микроконтроллера STM32F4, однако в общем случае для других моделей микроконтроллеров STM32 может использоваться подобный подход к настройке UART интерфейса. В данном примере инициализируются порты для работы с подключенным UART устройством (GPIOB6 и GPIOB7), настраивается их альтернативная функция для работы с модулем USART1, задается скорость передачи данных (9600 бит/с), формат передачи данных (8 бит, 1 стоп-бит, без проверки четности) и включаются прерывания для передачи и приема данных. После инициализации модуль USART1 включается для начала работы с UART интерфейсом.

#### **Ход работы:**

1. Подключите микроконтроллер STM32 к компьютеру через COM порт.

2. Напишите программу, которая будет настраивать UART интерфейс на работу с заданной скоростью передачи данных через COM порт.
3. Напишите программу, которая будет отправлять строку данных через UART интерфейс на компьютер.
4. Включите программу и убедитесь, что передаваемая строка данных успешно отображается на экране компьютера.

#### **Индивидуальные задания**

1. Настройте UART интерфейс для работы на скорости 115200 бит/с.
2. Используя прерывания, реализуйте прием и передачу данных через UART интерфейс.
3. Разработайте программу, которая будет передавать и принимать символы через UART интерфейс, и подключите ее к ПК через USB-UART конвертер.
4. Настройте программу для передачи и приема двоичных данных (например, сигналы с датчиков), используя формат данных 8-бит, без проверки четности и 1 стоп-бит.
5. Разработайте программу для считывания данных с UART интерфейса и отправки их на LCD.