

Лабораторная работа №3

Цель работы:

Получение навыков работы с USART, обмен данными с компьютером.

Программное обеспечение:

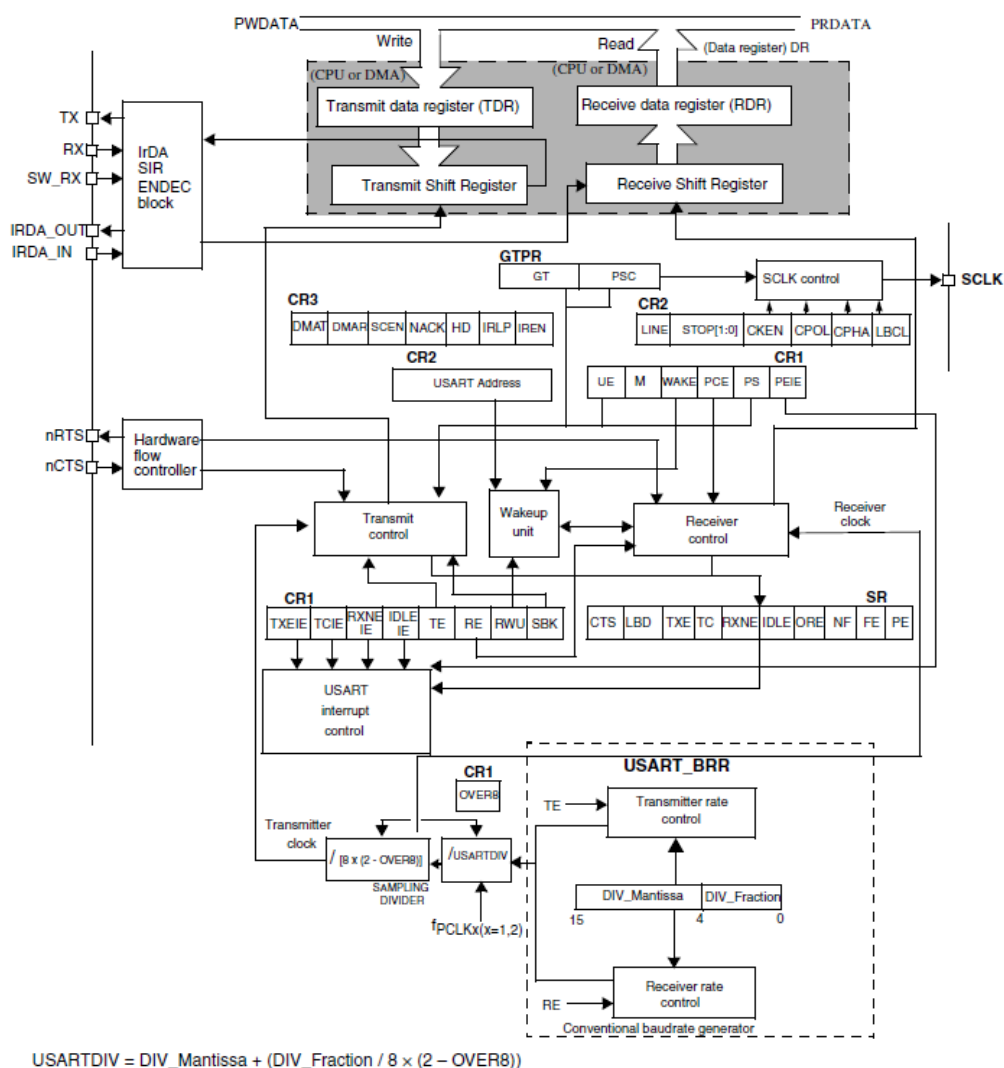
STM32CubeIDE, Terminal.

Общие сведения:

Универсальный 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 одной физической цифровой линии другому аналогичному устройству. Метод преобразования хорошо стандартизован и широко применяется в компьютерной технике (особенно во встраиваемых устройствах и системах на кристалле (SoC)).

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

UART может представлять собой отдельную микросхему (например, Intel I8251, I8250) или являться частью большой интегральной схемы (например, микроконтроллера). Используется для передачи данных через последовательный порт компьютера.



Данные между устройствами передают обычно пакетами определенной фиксированной длины. Длина пакета приема и передачи может быть разная. Например, пакет данных, передаваемых датчиком влажности и температуры:

BYTE COUNT	HEX	DECIMAL	CHARACTER DISPLAYED	Details
1	0x0A	10	'\n' = LF	New Line Character
2	0x48	72	'H'	Fixed Character H
3	0x3A	58	':'	Fixed Character :
4	0x30	48	'0'	Humidity Character-Hundreds
5	0x35	53	'5'	Humidity Character-Tens
6	0x38	56	'8'	Humidity Character-Ones
7	0x20	32	' '	Space
8	0x54	84	'T'	Fixed Character T
9	0x3A	58	':'	Fixed Character :
10	0x30	48	'0'	Temperature Character-Hundreds
11	0x32	50	'2'	Temperature Character-Tens
12	0x34	52	'4'	Temperature Character-Ones
13	0x0D	13	'\r' = CR	New Line Character

В приведенном примере 1 и 13 байты являются начало и концом сообщения.

Алгоритм передачи данных со стороны контроллера с использованием прерываний:

- 1) Формируется буфер для отправки и задается длина отправляемого массива.
- 2) Включается прерывание по пустому регистру отправки данных, обнуляется счетчик отправленных данных.
- 3) В прерывании, по событию написанному в предыдущем пункте, в регистр данных USART записывается байт данных из массива, значение счетчика отправленных данных увеличивается.
- 4) После отправки всех требуемых байт из буфера, отключается прерывание по пустому регистру отправки данных.

Алгоритм приема данных со стороны контроллера с использованием прерываний похож на алгоритм приема передачи данных с небольшими изменениями:

- Прерывание по не пустому регистру принятых данных включено всегда.
- Принимается всегда фиксированное число байт.
- Счетчик принятых данных обнуляется при получении символа конца пакета.

Функции HAL для работы с UART

Передача данных с использованием прерывания:

*HAL_UART_Transmit_IT (UART_HandleTypeDef * huart, uint8_t * pData, uint16_t Size)*

- *huart* – указатель на структуру конфигурации типа UART_HandleTypeDef.
- *pData* – указатель на буфер передаваемых данных.
- *Size* – количество данных, которые надо передать.

Прием данных с использованием прерывания:

*HAL_UART_Receive_IT (UART_HandleTypeDef * huart, uint8_t * pData, uint16_t Size)*

- *huart* – указатель на структуру конфигурации типа UART_HandleTypeDef.
- *pData* – указатель на буфер для принятых данных.
- *Size* – количество данных, которые необходимо принять.

Callback-функция приема данных:

*void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)*

- *huart* – указатель на структуру конфигурации типа UART_HandleTypeDef.

Порядок выполнения работы:

Часть I. Разработка программы при помощи кодгенератора.

1. Запустите STM32CubeIDE, в открывшемся окне выберете путь к вашей рабочей папке. В пути к рабочей папке и названии проекта не должно быть русских букв. В этой папке должны храниться все лабораторные работы.
2. На основе схемы стенда определить, какой UART используется для передачи данных на компьютер.
3. В окне графической инициализации контроллера (name.IOC), задать для UART, определенного в предыдущем пункте, асинхронный режим работы. Установить скорость передачи данных в соответствии с вариантом. Включить глобальные прерывания от UART.
4. В main файле добавить callback-функцию:

```
/* USER CODE BEGIN 4 */  
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart) {  
  
}  
/* USER CODE END 4 */
```

В main функции в соответствующей области запустить прием данных по UART при помощи прерывания.

5. Разработать протокол обмена данными с компьютером. Пакет данных от компьютера должен содержать байты идентификаторы начала и конца посылки, 2 байта числа. Пример: N25E – пакет 4 байта; N,E – байты идентификаторы начала и конца посылки соответственно; 25 – десятичное число записанное в 2х байтах, «2» в одном байте «5» в другом. Пакет данных компьютеру должен быть сформирован аналогичным образом.

Для приема и передачи данных на компьютере необходимо использовать программу Terminal, либо аналогичную. Обратите внимание, что данная программа отправляет данные в виде ASCII кодов:



Пакет «N25E» в строке отправки будет преобразован в массив [78;50;53;69].

6. Реализовать следующий алгоритм работы программы:

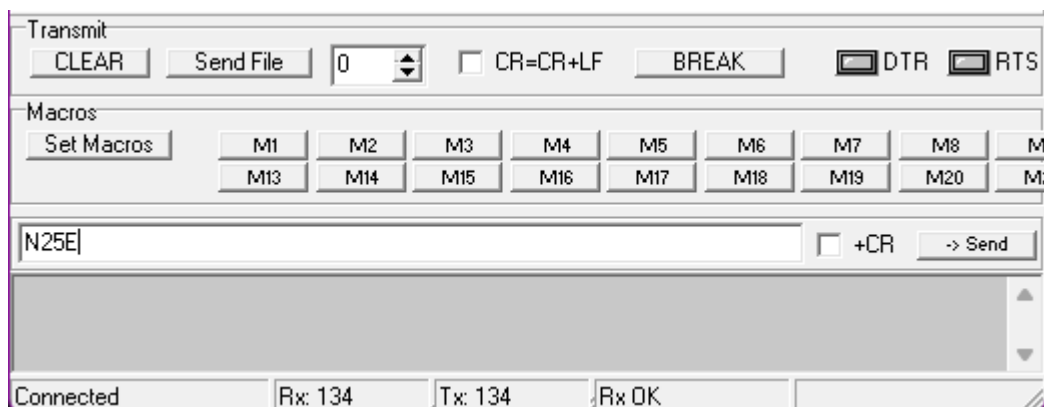
- Принять от компьютера пакет данных;
- Извлечь из посылки число;
- Произвести математическую операцию над этим числом;
- Результат вывести на 7-ми сегментный индикатор;
- Сформировать пакет для отправки результата вычислений на компьютер;
- Отправить пакет данных на компьютер.

Для простоты формирования пакета данных для отправки можно использовать функцию `sprintf`.

Часть II. Разработка программы при помощи регистров.

1. Запустите STM32CubeIDE, в открывшемся окне выберите путь к вашей рабочей папке. В пути к рабочей папке и названии проекта не должно быть русских букв. В этой папке должны храниться все лабораторные работы.
2. На основе схемы стенда определить, какой UART используется для передачи данных на компьютер.
3. Проинициализировать UART для приема и передачи данных, включить прерывание по приему данных. При настройке регистра BRR, принять частоту работы шины, к которой подключён UART, равной 16МГц. Включить обработку прерываний от UART в контроллере прерываний. Создать функцию обработки прерываний от UART.
4. В обработчике прерывания реализовать алгоритмы приема и передачи данных описанные в «Общих сведениях» к лабораторной работе.
5. Разработать протокол обмена данными с компьютером. Пакет данных от компьютера должен содержать байты идентификаторы начала и конца посылки, 2 байта числа. Пример: N25E – пакет 4 байта; N,E – байты идентификаторы начала и конца посылки соответственно; 25 – десятичное число записанное в 2х байтах, «2» в одном байте «5» в другом. Пакет данных компьютеру должен быть сформирован аналогичным образом.

Для приема и передачи данных на компьютере необходимо использовать программу Terminal, либо аналогичную. Обратите внимание, что данная программа отправляет данные в виде ASCII кодов:



Пакет «N25E» в строке отправки будет преобразован в массив [78;50;53;69].

6. Реализовать следующий алгоритм работы программы:

- Принять от компьютера пакет данных;
- Извлечь из посылки число;
- Произвести математическую операцию над этим числом;
- Результат вывести на 7-ми сегментный индикатор;
- Сформировать пакет для отправки результата вычислений на компьютер;
- Отправить пакет данных на компьютер.

Для простоты формирования пакета данных для отправки можно использовать функцию `sprintf`.

Задания

1. Выполните все действия из части I.
2. По возможности выполните все действия из части II.

Все пункты заданий по очереди продемонстрируйте преподавателю.

Варианты:

№ варианта	Baudrate	Функция
1	9600	e^x
2	19200	$\sin(x)$
3	38400	$\cos(x)$
4	57600	$\ln(x)$
5	115200	e^x
6	9600	$\sin(x)$
7	19200	$\cos(x)$
8	38400	$\ln(x)$
9	57600	e^x
10	115200	$\sin(x)$
11	9600	$\cos(x)$
12	19200	$\ln(x)$