

Работа 2

Цель

- Развить навыки настройки и использования периферии USART для последовательной связи.
- Получить практические умения в реализации структурированного протокола связи между МК и ПК.
- Получить опыт использования логического анализатора для отладки и проверки сигналов последовательной связи.

Программное обеспечение

1. Среда разработки: Visual Studio Code.
2. Набор инструментов STM32: STM32CubeCLT.
3. Расширения VS Code:
 - STM32Cube for Visual Studio Code
 - Output Colorizer
4. Терминал: Terminal
5. ПО для логического анализатора: Logic 2

Аппаратное обеспечение

1. Лабораторный стенд
2. Логический анализатор

Задание

Предварительные шаги

- ☐
1. Ознакомьтесь с руководством пользователя платы Nucleo, чтобы определить, какой USART подключён к встроенному программатору ST-LINK (используется как виртуальный COM-порт). Запишите выводы МК, используемые для RX и TX.
- ☐
2. Изучите схему платы, чтобы найти физические точки подключения для выбранных линий RX и TX.
- ☐
3. Используйте код инициализации массива светодиодов из предыдущей лабораторной работы.
- #### Основное
- ☐

1. Добавьте заголовочный файл `stm32f446xx.h` в ваш файл *main.c*.

☐

2. Включите тактирование выбранного периферийного устройства USART в регистрах RCC:

```
RCC->APB1ENR |= RCC_APB1ENR_USART2EN;
```
- ☐
3. В Reference Manual найдите раздел, описывающий периферию USART и её регистры.
- ☐
4. Настройте USART с помощью соответствующих регистров:

☐ а) Включите приёмник и передатчик.

☐ b) Рассчитайте и установите скорость передачи (baud rate) в регистре `BRR` для системной частоты 16 МГц, согласно вашему варианту.

☐ c) Включите генерацию прерываний `RXNE`.

☐ d) Разрешите соответствующий вектор прерываний USART в NVIC.

☐ e) Включите сам USART.

☐

5. Разработайте протокол связи master (ПК)/slave (МК). Используйте структуру пакета из Таблицы 1. Реализуйте команды и ответы из Таблицы 2.

☐

6. Подключите логический анализатор к линиям RX и TX. Захватите и проанализируйте сигналы в процессе обмена, чтобы проверить скорость передачи и целостность данных.

Таблица 1

1-й байт	2-й байт	3-й байт	4-й байт	...	(n - 1)-й байт	n-й байт
Заголовок	Размер пакета в байтах	ID slave	Данные	...	Данные	Терминатор

Таблица 2

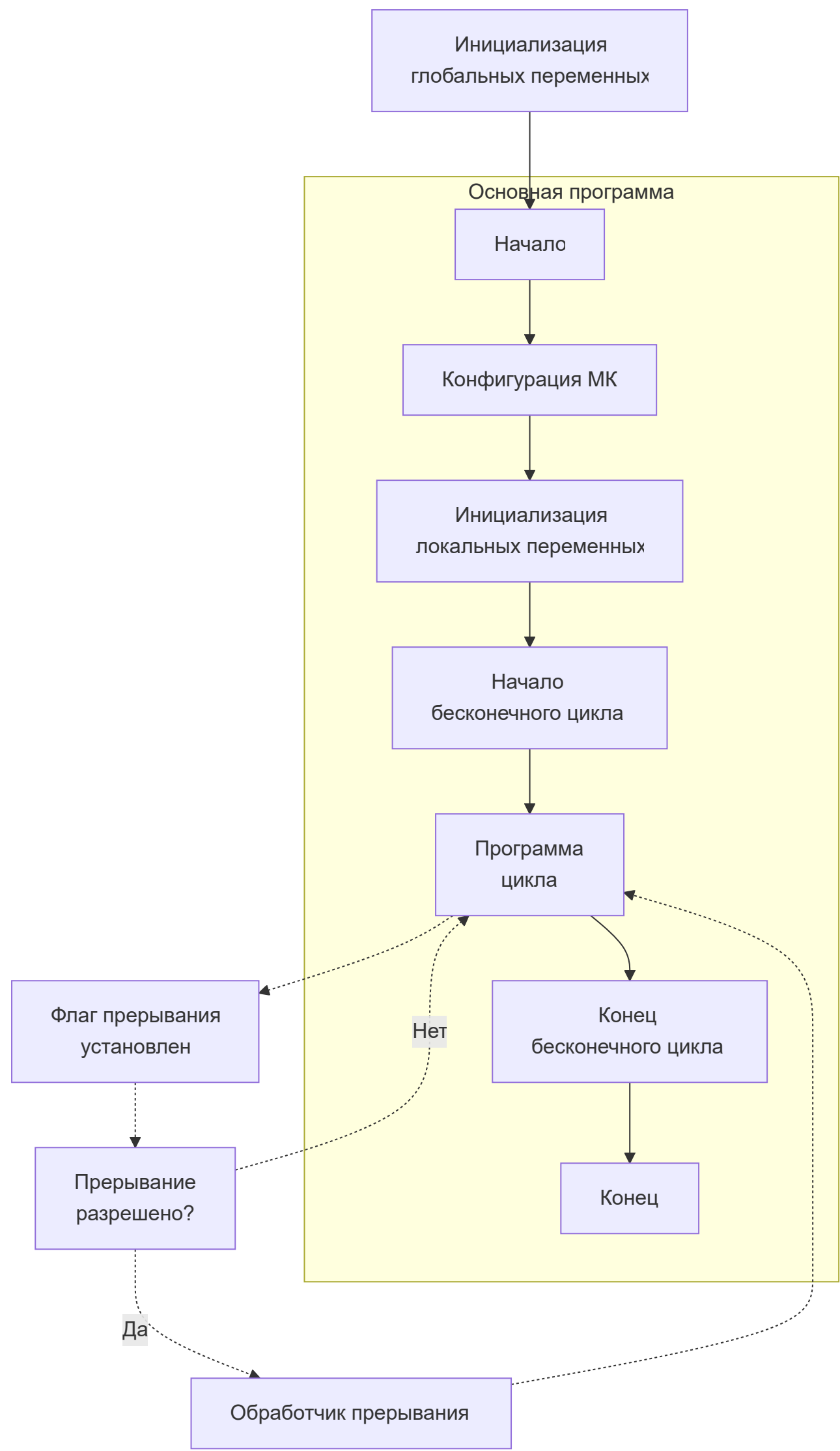
Запрос	Ответ
Echo	Повторяет полученный пакет.
Включить все светодиоды	Повторяет команду и включает все светодиоды.
Выключить все светодиоды	Повторяет команду и выключает все светодиоды.
Установить конкретный паттерн светодиодов (по варианту)	Повторяет команду и включает светодиоды согласно полученным данным.
Запрос состояния светодиодов	Отвечает текущим состоянием светодиодов (байт в двоичном виде, где каждый бит = один светодиод).

Руководство

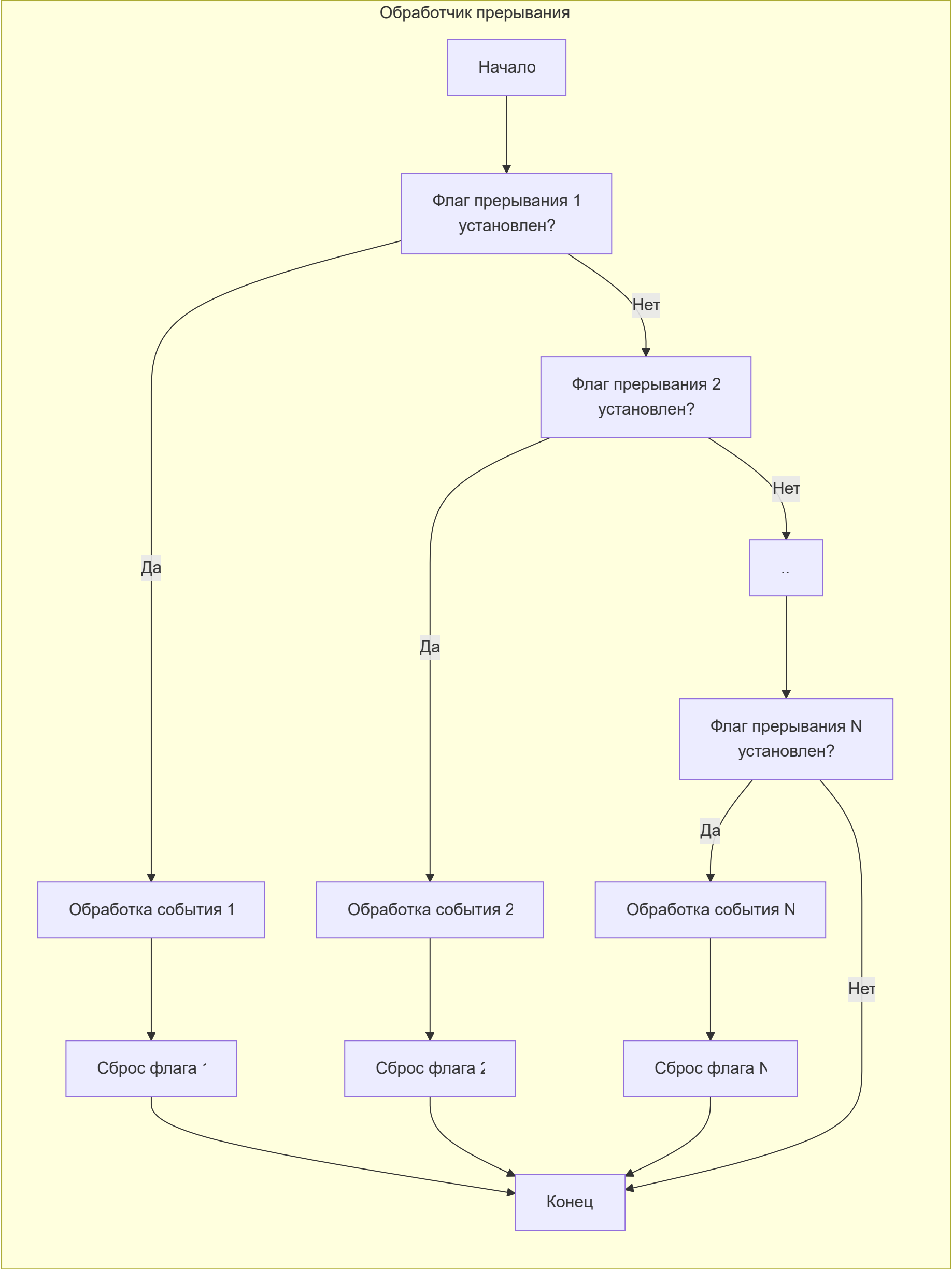
Реализуйте программу постепенно:

- 1. Сначала настройте и проверьте USART. Реализуйте простой echo: МК должен возвращать каждый принятый символ. Проверьте работу через терминал.
- 2. Добавьте возможность обработки простых команд (например, символ '1' включает светодиод, '0' — выключает).
- 3. Реализуйте полноценный протокол обмена пакетами (Табл. 1 и Табл. 2).

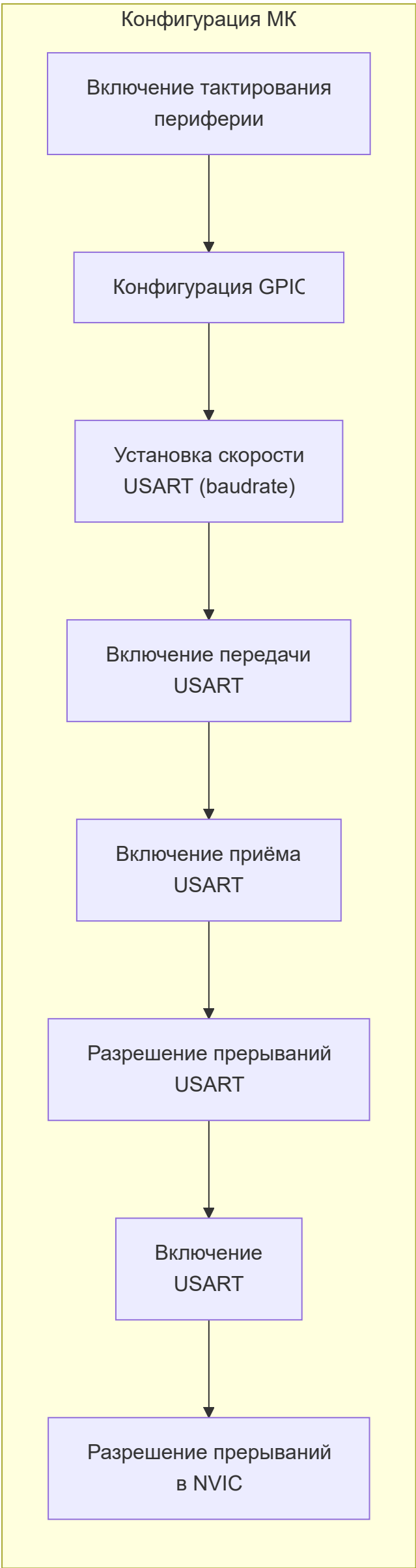
Эта блок-схема иллюстрирует рекомендуемую структуру программы с использованием прерываний:



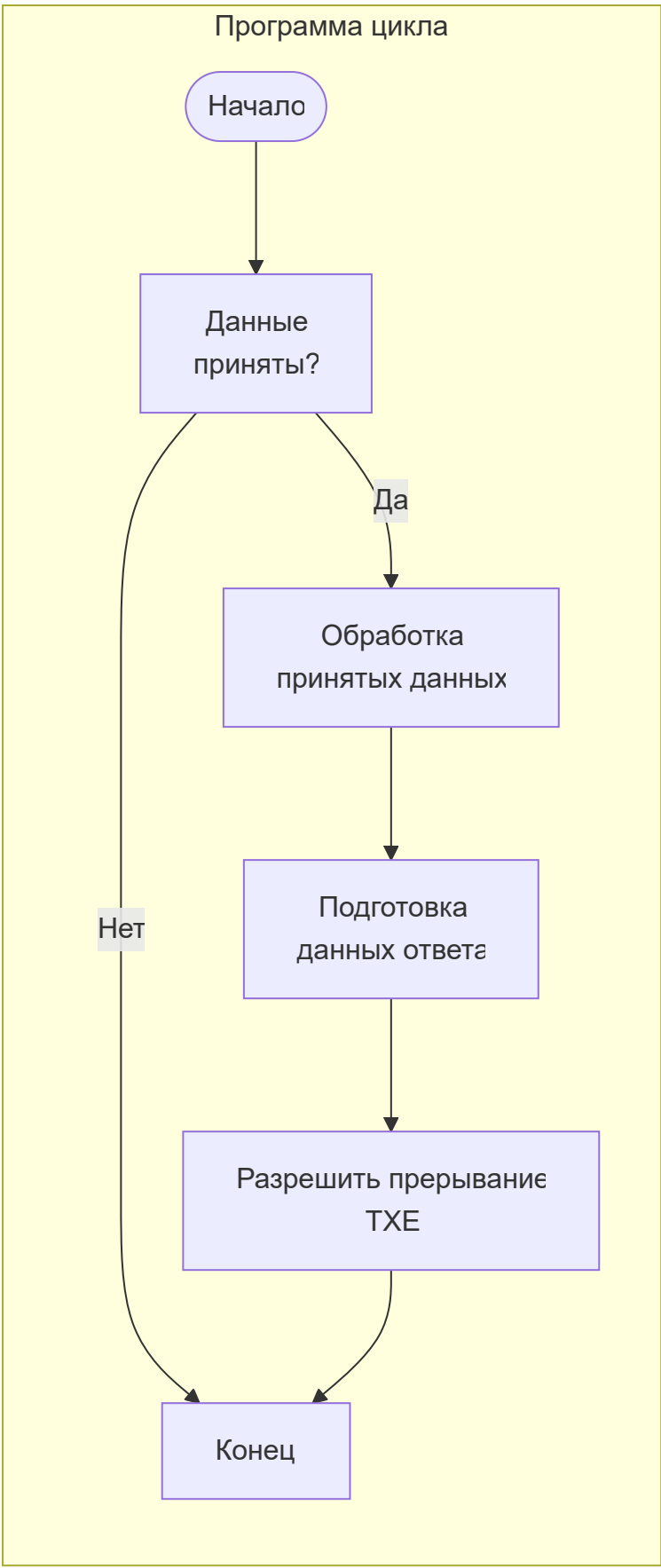
Типовой обработчик прерывания (ISR) проверяет источник и выполняет обработку:



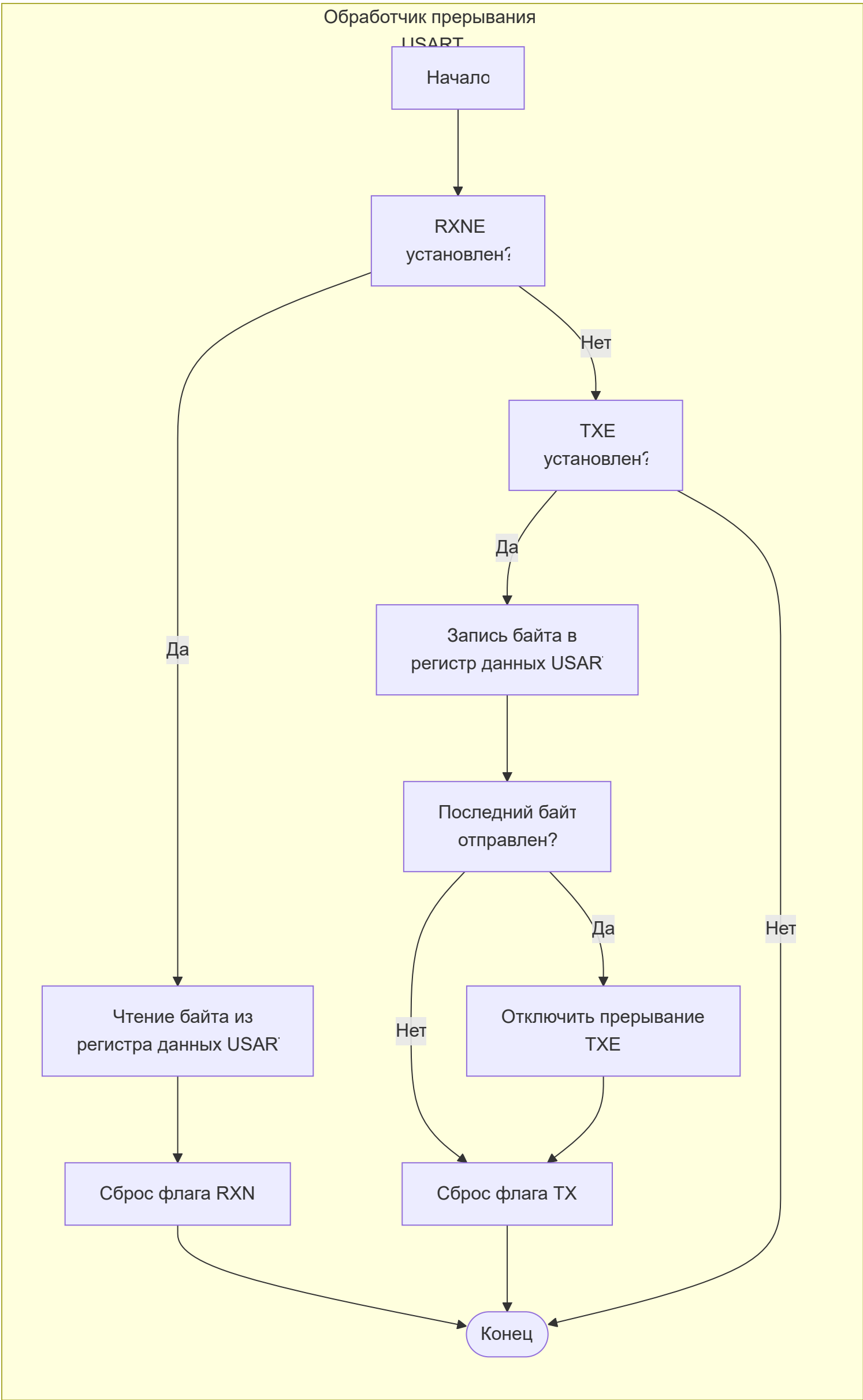
Этап конфигурации USART состоит из последовательности записей в регистры:



Структура программы с USART может быть такой:



Структура обработчика прерывания USART:



Варианты

Таблица 3

Вариант	Baudrate	Формат команды включения светодиодов
1	9600	Отправляется десятичное число (0-8). Включается соответствующее количество светодиодов (напр. 3 → 3 LED).
2	9600	Отправляется десятичное число (0-8). Включается один конкретный светодиод (напр. 3 → только 3-й LED).
3	9600	Отправляются два числа: номер светодиода (0-8) и его состояние (0=выкл, 1=вкл).
4	9600	Отправляется один байт (0b00000000 – 0b11111111). МК отображает его напрямую на массиве светодиодов.
5	9600	Отправляется число (0-255). МК переводит его в двоичный вид и выводит на светодиоды.
6	19200	Отправляется десятичное число (0-8). Включается соответствующее количество светодиодов (напр. 3 → 3 LED).
7	19200	Отправляется десятичное число (0-8). Включается один конкретный светодиод (напр. 3 → только 3-й LED).
8	19200	Отправляются два числа: номер светодиода (0-8) и его состояние (0=выкл, 1=вкл).
9	19200	Отправляется один байт (0b00000000 – 0b11111111). МК отображает его напрямую на массиве светодиодов.
10	19200	Отправляется число (0-255). МК переводит его в двоичный вид и выводит на светодиоды.
11	38400	Отправляется десятичное число (0-8). Включается соответствующее количество светодиодов (напр. 3 → 3 LED).
12	38400	Отправляется десятичное число (0-8). Включается один конкретный светодиод (напр. 3 → только 3-й LED).
13	38400	Отправляются два числа: номер светодиода (0-8) и его состояние (0=выкл, 1=вкл).
14	38400	Отправляется один байт (0b00000000 – 0b11111111). МК отображает его напрямую на массиве светодиодов.
15	38400	Отправляется число (0-255). МК переводит его в двоичный вид и выводит на светодиоды.

Продвинутое задание

Реализуйте приём и передачу данных с использованием DMA. Для определения конца кадра используйте прерывание USART IDLE.

Вопросы

1. Что такое UART и для чего он предназначен?
2. Что значить синхронная и асинхронная передача данных?
3. Каковы основные функциональные блоки периферии USART?
4. Как формируется и интерпретируется кадр данных UART (start bit, data bits, parity bit, stop bit)?
5. Какие основные параметры конфигурации UART необходимо настроить?
6. В чём разница между опросом (polling) и прерываниями в USART? Плюсы и минусы каждого подхода.
7. Какие типы прерываний обычно генерируются периферией UART?
8. Как отличить прерывания в обработчике?
9. Как логический анализатор помогает отлаживать протоколы последовательной связи?
10. Каково назначение протокола на основе пакетов (как в Таблице 1) по сравнению с передачей «сырых» данных?