|  |  |
| --- | --- |
| **Задание**  Разработать и написать драйверы UART для учебно-лабораторного стенда SDK-1.1M, с использованием и без использования прерываний. Написать тестовую программу для разработанных драйверов, которая выполняет определенную вариантом задачу.  **Общие сведения**  Интерфейс UART(USART) широко применяется в вычислительной технике для связи между цифровыми устройствами и фактически является стандартом «де-факто» для подключения периферийных устройств (самый распространённый пример – различные беспроводные модемы).  С помощью UART соединяются отдельные микросхемы, физически располагающиеся близко, как правило – на одной плате. Для передачи сигналов между конструктивно самостоятельными устройствами и на большие расстояния сигнал UART необходимо пропустить через приёмопередатчик, преобразующий передаваемый сигнал в сигналы таких стандартов, как RS-232 (позволяет передавать сигнал на расстояние порядка 15м) или RS-485 (осуществляет передачу сигнала на километры, при соблюдении некоторых условий и подключении ретрансляторов).  Современные микроконтроллеры практически всегда имеют в своём составе отдельную функциональную единицу, контроллер UART(USART), аппаратно реализующую данный интерфейс. Такие контроллеры самостоятельно переключают соответствующие входы/выходы микроконтроллера, формируют и считывают сигналы. Управление контроллера процессором осуществляется через запись/чтение в соответствующие регистры, подключённые к системной шине.  UART – последовательный интерфейс передачи данных. Это предполагает одну сигнальную линию (провод, дорожку на плате) для передачи данных в одном направлении, по которой информационные биты передаются друг за другом, последовательно.  Стандарт UART является чисто асинхронным интерфейсом, но реализующий его контроллер, как правило, может настраиваться в широких пределах и функционировать как в синхронном, так и асинхронном режимах.  Синхронный режим предполагает наличие средств синхронизации передатчика и приемника. Как правило, для синхронизации используют специальную линию для передачи тактовых импульсов. Информация в канале данных считывается приемником только в те моменты, когда на линии синхронизации сигнал активный.  В асинхронном режиме посылке очередного байта информации предшествует специальный старт-бит, сигнализирующий о начале передачи (обычно логический «0»). Затем следуют биты данных (их обычно 8), за которыми может следовать дополнительный бит (его наличие зависит от режима передачи, обычно этот бит выполняет функцию контроля четности). Завершается посылка стоп-битом (логическая «1»), длина которого (длительность единичного состояния линии) может соответствовать длительности передачи 1, 1.5 («полтора стоп-бита») или 2 бит. Стоп-бит гарантирует некоторую выдержку между соседними посылками, при этом пауза между ними может быть сколь угодно долгой (без учета понятия «тайм-аута»). Для асинхронного режима предусмотрен ряд стандартных скоростей обмена: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 и 115200 bps.  Скорость изменения передатчиком состояния сигнальной линии измеряется в бодах (baud) – количестве изменений состояния линии за одну секунду. В простейшем случае в линии имеется всего два состояния сигнала, т.е. одним состоянием кодируется один бит, и тогда скорость изменения состояния в бодах совпадает со скоростью передачи двоичной информации, bps(bit persecond, бит/с). Однако, при использовании других методов модуляции возможны несколько состояний сигнала, что позволяет одним состоянием кодировать сразу несколько передаваемых бит, и здесь скорость передачи данных bps превышает скорость изменения сигнала baud.  Интерфейс UART является дуплексным каналом передачи данных. Вообще, обмен данными бывает:  1. Дуплексным – предполагает прием и передачу данных одновременно (UART, RS232);  2. Полудуплексным – данные передаются в одном направлении с возможностью смены направления (RS-485);  3. Симплексным – данные передаются только в одном направлении  **Передача и прием данных по UART**  Сперва, необходимо произвести настройку Pinout в STM32CubeMX:  UART/USART подключается в разделе Connectivity со следующими параметрами:  • Режим (Mode) – асинхронный (asynchronous);  • Скорость (Baud Rate) – 115200 bps;  • Длина сообщения – 8 бит;  • Четность (Parity) – None;  • Стоп-бит – 1;  • Направление передачи (Data direction) – Receive and Transmit  Для работы с UART на микроконтроллерах STM32 используется библиотека HAL (Hardware Abstract Layer). STM32CubeMX при генерации проекта добавляет в код функцию инициализации UART MX\_UART4\_Init().  static void MX\_UART4\_Init(void)  {  huart4.Instance = UART4;  huart4.Init.BaudRate = 115200;  huart4.Init.WordLength = UART\_WORDLENGTH\_8B;  huart4.Init.StopBits = UART\_STOPBITS\_1;  huart4.Init.Parity = UART\_PARITY\_NONE;  huart4.Init.Mode = UART\_MODE\_TX\_RX;  huart4.Init.HwFlowCtl = UART\_HWCONTROL\_NONE;  huart4.Init.OverSampling = UART\_OVERSAMPLING\_16;  }  if (HAL\_UART\_Init(&huart4) != HAL\_OK)  {  Error\_Handler();  }  В библиотеке HAL существует две стандартных функции, отвечающих за прием и передачу информации:  HAL\_UART\_Transmit(UART\_HandleTypeDef \*huart, uint8\_t \*pData, uint16\_t Size, uint32\_t Timeout);  HAL\_UART\_Receive(UART\_HandleTypeDef \*huart, uint8\_t \*pData, uint16\_t Size, uint32\_t Timeout);  • \*huart – указатель на структуру UART\_HandleTypeDef, которая содержит информацию о конфигурации для указанного модуля UART;  • \*pData – указатель на буфер данных;  • Size – размер данных которые будут переданы или приняты;  • Timeout – длительность времени ожидания. | Nhiệm vụ  Phát triển và viết trình điều khiển UART cho phòng thí nghiệm và đào tạo SDK-1.1M, có và không có gián đoạn. Viết chương trình thử nghiệm cho các trình điều khiển đã phát triển thực hiện một tác vụ cụ thể.  Thông tin chung  Giao diện UART (USART) được sử dụng rộng rãi trong máy tính để **giao tiếp** giữa các **thiết bị kỹ thuật số** và trên thực tế là tiêu chuẩn thực tế để **kết nối các thiết bị ngoại vi** (ví dụ phổ biến nhất là các modem không dây khác nhau).  Sử dụng UART, các vi mạch riêng lẻ được kết nối gần nhau về mặt vật lý, thường là trên cùng một bo mạch. Để truyền tín hiệu giữa các thiết bị độc lập về cấu trúc và trong khoảng cách xa, tín hiệu UART phải được truyền qua một bộ thu phát chuyển tín hiệu đã truyền thành tín hiệu của các tiêu chuẩn như RS-232 (cho phép truyền tín hiệu trong khoảng cách khoảng 15m) hoặc RS-485 (truyền tín hiệu trên km, tùy thuộc vào các điều kiện nhất định và sự kết nối của các bộ lặp).  Các bộ vi điều khiển hiện đại hầu như luôn bao gồm một bộ phận chức năng riêng biệt, bộ điều khiển UART (USART), thực hiện giao diện này trong phần cứng. Các bộ điều khiển như vậy **chuyển đổi độc lập các đầu vào / đầu ra** tương ứng của vi điều khiển, **tạo và đọc tín hiệu**. Bộ điều khiển được bộ xử lý điều khiển bằng cách **ghi / đọc** các **thanh ghi** tương ứng được kết nối với **bus hệ thống**.  **UART** là một **giao diện truyền dữ liệu nối tiếp**. Điều này giả định một đường tín hiệu (dây dẫn, rãnh ghi trên bảng) để **truyền dữ liệu theo một hướng**, theo đó các **bit** thông tin được **truyền tuần tự**.  Tiêu chuẩn UART là một giao diện hoàn toàn không đồng bộ, nhưng **bộ điều khiển** thực hiện nó thường có cấu hình cao và hoạt động ở cả **chế độ đồng bộ và không đồng bộ**.  Chế độ **đồng bộ** giả định rằng **máy phát và máy thu được đồng bộ hóa**. Theo quy định, một đường đặc biệt để truyền xung đồng hồ được sử dụng để đồng bộ hóa. Thông tin trong kênh dữ liệu được máy thu đọc chỉ trong những thời điểm khi tín hiệu hoạt động trên đường đồng bộ.  Trong **chế độ không đồng bộ**, việc gửi byte thông tin tiếp theo được đặt trước   * một **bit** bắt đầu **đặc biệt**, báo hiệu **sự bắt đầu truyền** (thường là "**0" logic**). * Sau đó các **bit dữ liệu** theo sau (thường có **8**), * tiếp theo là một **bit bổ sung** (sự hiện diện của nó phụ thuộc vào chế độ truyền, thông thường bit này thực hiện **chức năng chẵn lẻ**). * Việc gửi kết thúc bằng một **bit dừng** (lôgic **"1"**), độ dài của nó (thời gian của một trạng thái dòng đơn) có thể tương ứng với khoảng thời gian truyền 1, 1,5 ("một bit rưỡi") hoặc 2 bit. Bit dừng đảm bảo một độ trễ nhất định giữa các thông điệp lân cận, trong khi thời gian tạm dừng giữa chúng có thể dài tùy ý (không tính đến khái niệm "thời gian chờ").   Một số tốc độ truyền tiêu chuẩn được cung cấp cho chế độ không đồng bộ: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 và 115200 bps.    Tốc độ thay đổi trạng thái đường tín hiệu của máy phát được đo bằng baud (baud) - số lượng trạng thái đường truyền thay đổi trong một giây. Trong trường hợp đơn giản nhất, chỉ có hai trạng thái tín hiệu trong đường dây, tức là một bit được mã hóa bởi một trạng thái, và sau đó tốc độ thay đổi trạng thái trong baud trùng với tốc độ bit, bps (bit persecond, bit / s). Tuy nhiên, khi sử dụng các phương pháp điều chế khác, có thể có một số trạng thái tín hiệu, cho phép một trạng thái mã hóa một số bit được truyền cùng một lúc và ở đây tốc độ truyền bps vượt quá tốc độ thay đổi tín hiệu baud.  Giao diện UART là một kênh giao tiếp song công đầy đủ. Nói chung, trao đổi dữ liệu là:  1. Song công - giả sử **nhận và truyền dữ liệu đồng thời** (UART, RS232);  2. Bán song công - dữ liệu được **truyền theo một hướng** với **khả năng thay đổi hướng** (RS-485);  3. Simplex - dữ liệu chỉ được **truyền theo một hướng**  Truyền và nhận dữ liệu UART  Đầu tiên, bạn cần định cấu hình Pinout trong STM32CubeMX:  UART / USART được kết nối trong phần Kết nối với các thông số sau:  • Chế độ (Mode) - không đồng bộ (asynchronous);  • Tốc độ Baud - 115200 bps;  • Độ dài bản tin - 8 bit;  • Chẵn lẻ - Không có;  • Dừng bit - 1;  • Hướng dữ liệu - Nhận và Truyền  Để làm việc với UART trên vi điều khiển STM32, thư viện HAL (Lớp trừu tượng phần cứng) được sử dụng. STM32CubeMX, khi tạo một dự án, thêm chức năng khởi tạo UART MX\_UART4\_Init () vào mã.  static void MX\_UART4\_Init (void)  {  huart4.Instance = UART4;  huart4.Init.BaudRate = 115200;  huart4.Init.WordLength = UART\_WORDLENGTH\_8B;  huart4.Init.StopBits = UART\_STOPBITS\_1;  huart4.Init.Parity = UART\_PARITY\_NONE;  huart4.Init.Mode = UART\_MODE\_TX\_RX;  huart4.Init.HwFlowCtl = UART\_HWCONTROL\_NONE;  huart4.Init.OverSampling = UART\_OVERSAMPLING\_16;  if (HAL\_UART\_Init (& huart4)! = HAL\_OK)  {  Error\_Handler ();  }  Có hai chức năng tiêu chuẩn trong thư viện HAL chịu trách nhiệm **nhận và truyền** thông tin:  HAL\_UART\_**Transmit** (UART\_HandleTypeDef \* **huart**, uint8\_t \* **pData**, uint16\_t **Size**, uint32\_t **Timeout**);  HAL\_UART\_**Receive** (UART\_HandleTypeDef \* huart, uint8\_t \* pData, uint16\_t Size, uint32\_t Timeout);  • \* **huart** - một **con trỏ** đến cấu trúc UART\_HandleTypeDef chứa **thông tin cấu hình** cho mô-đun UART được chỉ định;  • \* pData - con trỏ tới **bộ đệm dữ liệu**;  • Kích thước - kích thước của dữ liệu sẽ được truyền hoặc nhận;  • Thời gian chờ - khoảng thời gian chờ đợi. |