

Драйвер канала RS-485 (modbus rtu)

1 Описание

Описание места драйвера в программном комплексе ZigZag см. в документе «ZigZag: организация взаимодействия по RS-485».

2 Требования

Драйвер должен обеспечивать работу верхних уровней в режимах master, slave и sniffer.

Драйвер должен обеспечивать возможность переключения между режимами на лету.

Драйвер должен поддерживать разные скорости потока.

3 Анализ

3.1 *Режимы работы верхнего уровня*

3.1.1 Master

В данном режиме верхний уровень по очереди опрашивает ведомых, отправляя в шину пакеты-запросы, и ожидая определенное время ответа. Интерфейс должен позволять прерывать режим прослушивания в любой момент для инициации следующего опроса.

Верхний уровень должен быть проинформирован о каждом принятом пакете, в том числе о «битых пакетах» для диагностирования неисправности или конфликтной ситуации на шине.

3.1.2 Sniffer

В данном режиме верхний уровень должен быть проинформирован о каждом пакете, передаваемом по шине.

Так как, в соответствии со спецификацией Modbus RTU, пакеты могут следовать с интервалом 3,5 символа, драйвер должен обеспечить буферизацию принимаемых пакетов для сглаживания нагрузки.

3.1.3 Slave

Как и в предыдущих режимах, верхний уровень должен быть проинформирован о каждом пакете, передаваемом по шине для выделения пакета-запроса, адресованного данному узлу. Когда такой пакет найден, верхний уровень инициирует передачу ответного пакета. Причем передача ответа должна уложиться в окно ожидания ведущего.



Иллюстрация 3.1.: Возможная последовательность событий на узле в режиме SLAVE

Так как в данном режиме длительная обработка требуется только ограниченному количеству пакетов — адресованных данному устройству, а остальные пакеты только свидетельствуют об активности на шине и отбрасываются, будет достаточно буфера на два пакета. Эта ситуация отображена на иллюстрации 3.1: пакет адресованного данному устройству запроса может появиться на шине через 3,5 символа после конца предыдущего пакета. В этот момент только начинается обработка предыдущего пакета, и принимаемый запрос надо помещать в другую область памяти.

3.1.4 Переключение Master-Slave

3.1.5 Переключение Slave-Master

3.2 Сценарии использования

3.2.1 Отправка данных

Драйвер может находиться в пассивном режиме или в режиме прослушивания

1. Верхний уровень запрашивает передачу порции данных.
2. Драйвер отправляет данные в канал, после чего оповещает верхний уровень.
3. Драйвер переходит в режим прослушивания канала.

3.2.2 Прием данных

Драйвер находится в режиме прослушивания

1. Приходит пакет.
2. Драйвер оповещает верхний уровень о приеме пакета.
3. Драйвер переходит в режим прослушивания.

3.2.3 Ошибки отправки

Драйвер сообщает об ошибке, если запрос на отправку данных приходит во время приема или передачи пакета.

4 Предлагаемые интерфейсы

4.1 Микроконтроллер

Функция	Описание
void mb_rx_enable(bool enable)	Включить или выключить прослушивание канала
uint8_t* get_tx_buf()	Получить указатель на буфер отправки
result_t mb_tx(size_t length)	Отправить первые length байт из буфера отправки
void mb_rx_indication(bool correct)	Индикация приема пакета. correct = TRUE, если пакет принят корректно, FALSE в противном случае.
void mb_get_next(uint8_t** pdata, size_t* length, bool* more)	Получить указатель на принятый пакет и его размер. Параметр more = TRUE, если в буфере драйвера есть еще пакеты.
void mb_clear_rx_buf()	Очистит буфер входящих пакетов.

4.2 ПК