

Лабораторная работа №2.

RS-232. Разработка программы для обмена данными между контроллером и терминалом.

Предварительное задание:

- 1) Изучить раздел UART0 технической документации микроконтроллера C8051F060.
- 2) Изучить функциональные возможности программного продукта Termite.

Режим 1 работы модуля UART логически совместим с протоколом RS-232 компьютера. Кадр состоит из одного стартового бита, восьми бит данных и одного стоп-бита. Бит паритета в этом случае не используется.

Как известно, UART и RS-232 электрически несовместимы из-за разницы электрических характеристик сигналов. Для их согласования отладочный модуль имеет микросхему преобразователя уровней SP3223E, которая преобразует уровни UARTа (0..3,3) В в подходящее для RS-232 напряжение (-13.2..+13.2) В. Микросхема обеспечивает работу на скорости до 120 кбод.

Основной регистр, содержащий информацию о статусе модуля UART0 микроконтроллера C8051F060, называется SCON0:

| | | | | | | | | |
|------|------|------|------|------|------|-----|-----|---------------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Default Value |
| SM00 | SM10 | SM20 | REN0 | TB80 | RB80 | TI0 | RI0 | 00000000 |

SM00:SM10 – выбор режима работы модуля:

| SM00 | SM10 | Режим |
|------|------|--|
| 0 | 0 | Режим 0: синхронный |
| 0 | 1 | Режим 1: асинхр., 8 бит, переменная скорость |
| 1 | 0 | Режим 2: асинхр., 9 бит, постоянная скорость |
| 1 | 1 | Режим 3: асинхр., 9 бит, переменная скорость |

SM20 – разрешение мультипроцессорной передачи.

REN0 – разрешение передатчика. При REN0 = 0 прием данных не производится.

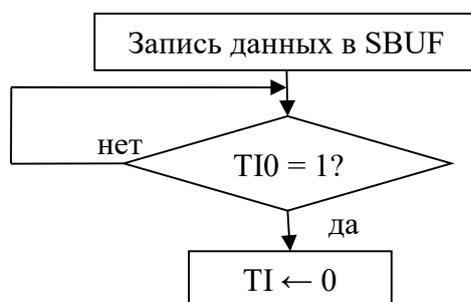
TB80, RB80 – девятый бит передаваемых и принимаемых данных соответственно. Используются для режимов девятибитовой передачи.

TI0 – флаг прерывания передатчика. Устанавливается аппаратно при завершении передачи посылки и инициирует соответствующее прерывание, если оно разрешено. После обслуживания прерывания должен быть сброшен программно.

RI0 – флаг прерывания приемника. Устанавливается аппаратно при завершении приема посылки и инициирует соответствующее прерывание, если оно разрешено. После обслуживания прерывания должен быть сброшен программно.

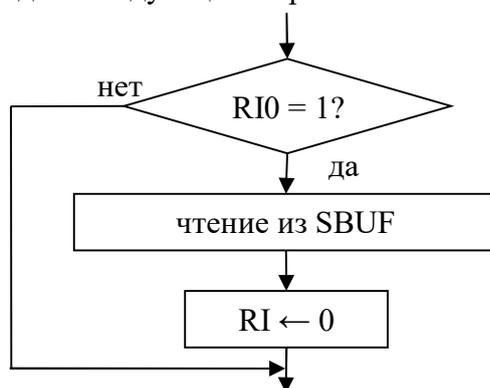
Для хранения передаваемых и принимаемых данных предназначен регистр SBUF0. Физически состоит из двух регистров: буфера приемника и буфера передатчика. При чтении данных из этого регистра производится обращение к принятым данным. При записи в SBUF данные попадают в буферный регистр передатчика.

Таким образом, алгоритм передачи байта данных по последовательному порту (без использования прерываний) может выглядеть следующим образом:



Этот прием иллюстрирует принцип передачи данных, однако он не рационален с точки зрения экономии вычислительных ресурсов. В данном случае программа в пустом цикле дожидается окончания передачи, чтобы сбросить флаг передатчика. Поскольку процессорное ядро тактируется гораздо более высокой частотой, чем частота работы приемопередатчика, за время передачи байта ядро могло бы выполнить большое количество операций. Поэтому такой прием целесообразно использовать на этапе запуска интерфейса и отладки программы, а при разработке целевой программы стоит использовать работу по прерываниям.

Алгоритм приема байта данных по последовательному порту (без использования прерываний) может выглядеть следующим образом:



Поскольку флаги RI и TI являются единственными индикаторами завершения приема и передачи, для корректного обмена данными их нужно своевременно сбрасывать.

Задание

1) С помощью автоконфигуратора задать следующие настройки для микроконтроллера: кроссбар включен, UART0 подключен к выводам P0.0 и P0.1, линии P3.0-P3.7 работают как цифровые двухтактные выходы, UART0 включен, режим асинхронный 8 бит с переменной скоростью, источник тактирования – Таймер 1, скорость обмена 9600 бод, Таймер 1 включен.

2) Написать программу, которая по нажатию кнопки на плате расширения отправляет в терминал произвольный байт, а при приеме байта от терминала отображает его на светодиодах платы расширения.

3) Переписать программу по п. 2 с использованием работы по прерываниям.

4) Увеличивая скорость обмена, определить максимальную скорость устойчивой передачи данных.

Рекомендации к выполнению лабораторной работы

- 1) Перед написанием программного кода всегда составляйте алгоритм.
- 2) Для отладки программы и поиска ошибок используйте выполнение программы по шагам (Step), а также функцию Run to Cursor.
- 3) Для проверки выполнения определенного участка кода, в него можно вставить следующий фрагмент:



Если при выполнении программы светодиод загорелся на 1 с, значит этот фрагмент был выполнен. С помощью этого приема можно также выполнять проверку каких-либо условий, используя оператор `if ()`. Вместо включения светодиода можно выводить на светодиоды платы расширения (порт P3) значащие данные, например, слово состояния периферийного модуля и пр.

- 4) При обращении к регистрам специальных функций (или их битам) не забывайте устанавливать нужную страницу регистров.