

3.2 Последовательный порт

1

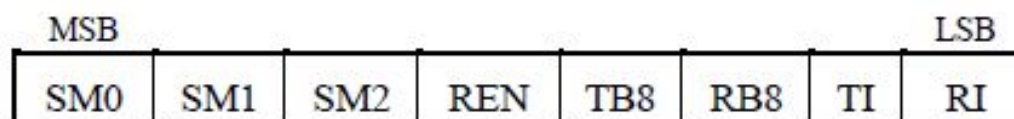
Последовательный порт является дуплексным, что означает возможность одновременной передачи и приема. Программный доступ к приемнику и передатчику – это соответственно считывание из РСФ SBUF и запись в SBUF, т.е. с идентификатором SBUF физически ассоциированы два буферных регистра: один буферный регистр приемника, другой – передатчика. С каждым буферным регистром, в свою очередь, ассоциируется сдвиговый регистр (программно недоступный), преобразующий последовательный код в параллельный и наоборот. Прием и передача осуществляется “младшими разрядами вперед”, т.е. первым передается младший бит байта.

Последовательный порт может функционировать в 4-х режимах. В режиме 0 последовательный порт работает как восьмибитовый сдвиговый регистр. В режимах 1, 2 и 3 – как UART. Последние режимы разнятся форматом передаваемых/принимаемых и способом задания частоты приема/передачи.

ИНИЦИАЛИЗАЦИЯ (УСТАНОВКА ПАРАМЕТРОВ) ПОСЛЕДОВАТЕЛЬНОГО ПОРТА

Параметры последовательного порта определяются, главным образом, бит-адресуемым РСФ SCON и частично РСФ PCON. Описание назначения битов этих регистров представлено на соответствующих рисунках.

Регистр SCON – это регистр с побитовой адресацией. Поэтому можно использовать имена его битов (см. рис. 1) в командах битовых операций. Регистр PCON не имеет побитовую адресацию. Поэтому при работе с его отдельными битами следует использовать логические команды, где вторым операндом является 8-битная константа.



SM0, SM1

Определяет режим последовательного порта

SM0	SM1	Режим	Описание	Скорость
0	0	0	Сдвиговой регистр	$F_{BQ}/12$
0	1	1	8-битный UART	Переменная
1	0	2	9-битный UART	$F_{BQ}/64$ или $F_{BQ}/32$
1	1	3	9-битный UART	Переменная

SM2

Разрешает мультипроцессорную связь в режимах 2 и 3. В этих режимах при SM2=1 бит RI не будет устанавливаться, если полученный 9-й бит (RB8) равен 0. В режиме 1 при SM2=1 бит RI не будет устанавливаться, если не получен стоп-бит. В режиме 0 бит SM2 должен быть 0.

REN

Если установлен, разрешает последовательный прием. Сбрасывается/устанавливается ПО.

TB8

Девятый бит данных, который может быть передан в режимах 2 и 3. Устанавливается и сбрасывается ПО.

RB8

В режимах 2 и 3 – девятый принимаемый бит. В режиме 1, если SM2=0, RB8 – принимаемый стоп-бит. В режиме 0 не используется.

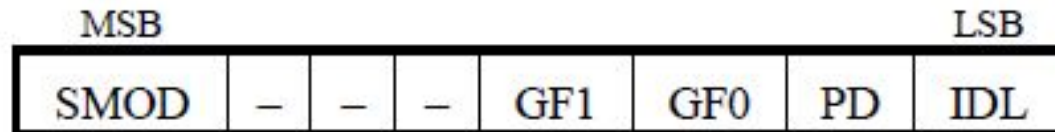
TI

Флаг прерывания передатчика. Устанавливается аппаратно в конце выдачи 8-го бита в режиме 0 или в начале стоп-бита в других режимах. Может быть сброшен ПО.

RI

Флаг прерывания приемника. Устанавливается аппаратно в конце приема 8-го бита в режиме 0 или в середине стоп-бита в других режимах при SM2=0 (см. описание SM2 для случая SM2=1). Может быть сброшен ПО.

Рис. 1. Назначение битов регистра SCON

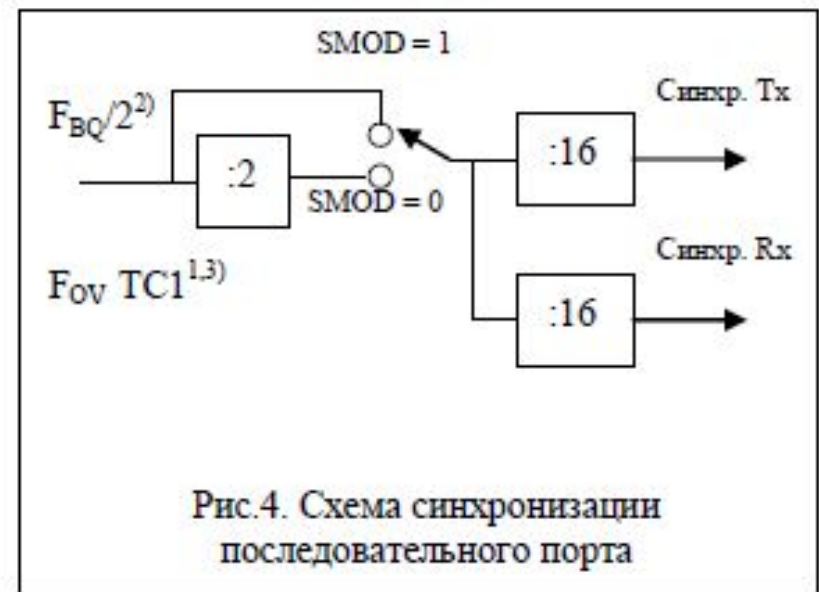
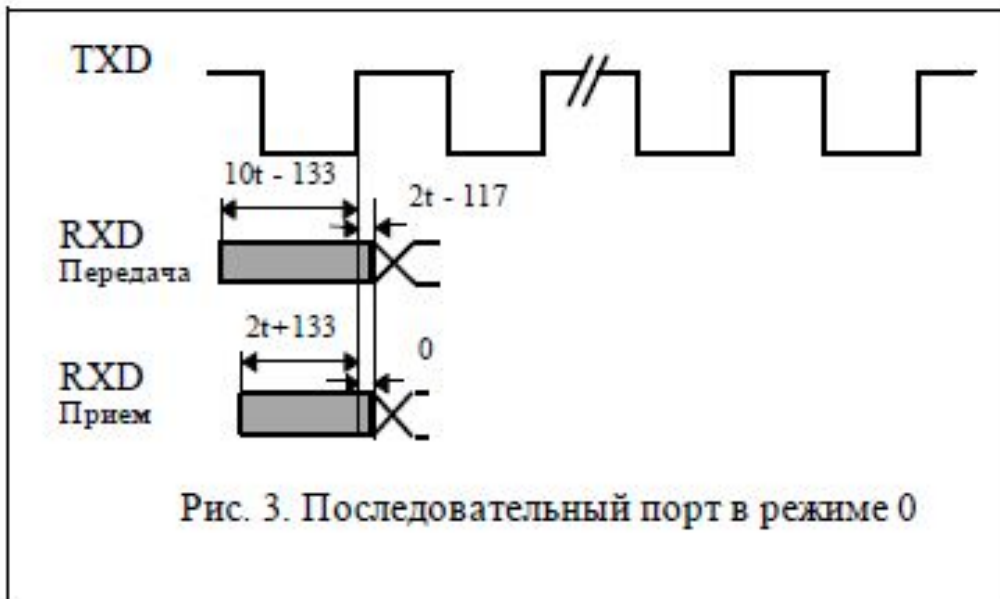


- SMOD** Бит удвоения скорости передачи. Когда TC1 используется для задания скорости передачи/приема последовательного порта и SMOD=1, то скорость приема/передачи удваивается.
- GF1** Флаг общего назначения.
- GF0** Флаг общего назначения.
- PD** Установка флаг переводит МК в режим микропотребления (50 мкА). В этом режиме блокируется работа задающего генератора МК, прекращая тем самым работу всех узлов МК, но сохраняется содержимое ячеек внутреннего ОЗУ. Единственным выходом из этого состояния является аппаратный сброс длительностью не менее 10 мс.
- IDL** Установка бита блокирует узлы центрального процессора МК, что уменьшает энергопотребление (5 мА). При этом сохраняется содержимое внутреннего ОЗУ и РСФ. Активизация любого разрешенного прерывания приводит к сбросу бита IDL и, следовательно, к окончанию холостого хода. После выполнения команды RETI управление будет передано команде, следующей за командой, переведшей МК в режим холостого хода. Другим способом окончания холостого хода является аппаратный сброс длительностью не менее 2 машинных циклов.

Рис.2. Назначение битов регистра PCON

ОПИСАНИЕ РЕЖИМОВ РАБОТЫ ПОСЛЕДОВАТЕЛЬНОГО ПОРТА

Режим 0. В этом режиме последовательный порт работает как сдвиговый регистр. Последовательные данные принимаются или передаются через вывод RXD (P3.0), а на выводе TXD (P3.1) микроконтроллер формирует синхросигнал. Скорость приема/передачи постоянна и составляет 1/12 частоты кварцевого резонатора МК: $BR_0 = F_{BQ}/12$.



На рис. 3 приняты обозначения: t – длительность машинного цикла; числовые константы даны в нс. Эти значения характерны для классического 8051.

Режимы UART (universal asynchronous receiver/transmitter - универсальный асинхронный приемопередатчик)

Все остальные режимы работы последовательного порта соответствуют режимам работы UART, работа которого поясняется следующим рисунком.

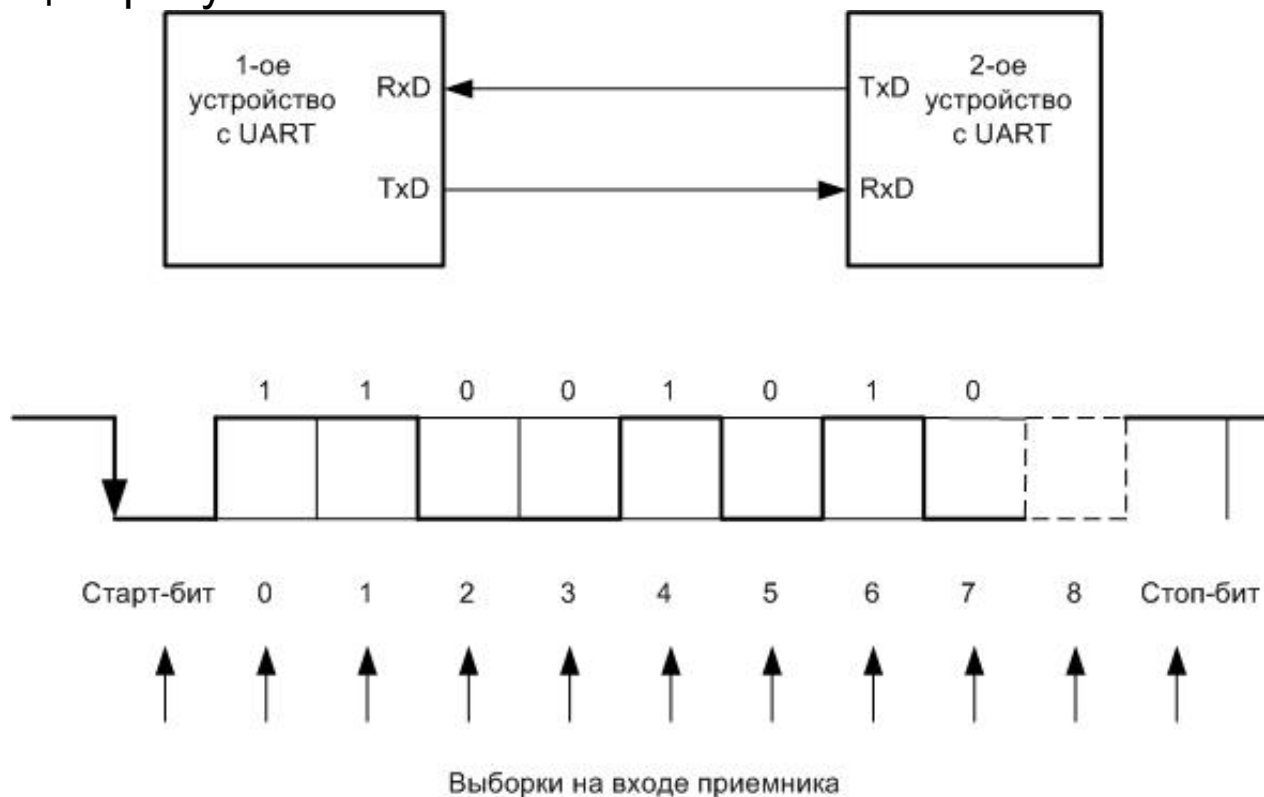


Рис. 5. Временная диаграмма, поясняющая работу приемника

Для правильной работы в режиме UART приемник и передатчик должны иметь с одинаковые скорости передачи бит и формат передачи байтов.

Скорость передачи/приема (BR – Baud Rate) имеет размерность Бод или бит/с (для двоичного канала $BR [\text{Бод}] = BR [\text{бит/с}]$). Формат передачи байта определяет структуру и количество передаваемых бит, включая служебные. Часто формат обозначается краткой записью, например, 1-8-1 (1 старт-бит; 8 бит данных; 1 стоп-бит).

При отсутствии передачи на входе приемника лог. «1». Приемник активизируется, когда на его входе произойдет отрицательный перепад. Чтобы надежнее отличить старт-бит от возможной короткой импульсной помехи, приемник выполняет выборку сигнала в середине старт-бита (лог. «0»). Если уровень лог. «0» не подтверждается, то приемник переходит в режим ожидания старт-бита. Если подтверждается лог. «0», то приемник производит выборки сигнала через интервал, равный длительности передачи одного бита, т.е. выборки опять будут производиться в середине интервала передачи бита.

Выборки сигнала фиксируются в сдвиговом регистре приемника. Последним принимается стоп-бит (лог. «1»), который гарантирует минимальный временной зазор между последовательными персылками байтов. Принятый байт из сдвигового регистра переписывается в регистр SBUF.

Если буфер передатчика пуст после очередной передачи байта, то бит **TI** = 1; если принят очередной байт, то бит **RI** = 1 (подробности см. на рис. 1).

Режим 1. В этом режиме последовательный порт работает в формате 8-битного UART. Через TXD передаются, а через RXD принимаются 10 бит: старт-бит (0), 8 бит данных и стоп-бит. При приеме стоп-бит поступает в RB8 регистра SCON. Скорость приема/передачи определяется частотой переполнения TC1 в режиме 2 и значением бита SMOD регистра PCON (см. рис. 4):

$$BR_1 = \frac{2^{SMOD}}{32} * \frac{F_{BQ}}{12 * [256 - TH_1]}, \quad (1)$$

где BR_1 – скорость передачи, Бод (бит/с); F_{BQ} – частота тактового генератора МК, Гц; TH_1 – константа, которая записывается в регистр TH1, когда TC1 работает в режиме 2.

$$TH_1 = 256 - \frac{2^{SMOD}}{32} * \frac{F_{BQ}}{12 * BR_1}. \quad (2)$$

Максимальная скорость при $F_{BQ} = 11059,2$ кГц составляет 57,6 кбит/с.

Режим 2. В этом режиме последовательный порт работает в формате 9-битного UART с постоянной скоростью. Через TXD передаются, а через RXD принимаются 11 бит: старт-бит (0), 8 бит данных, программируемый 9-й бит данных и стоп-бит. Девятый бит данных при передаче определяется битом TB8 (например для установки паритета). При приеме 9-й бит поступает в RB8. Скорость приема/передачи программируется как $1/32$ или $1/64 F_{BQ}$:

$$BR_2 = \frac{2^{SMOD}}{64} * F_{BQ}. \quad (3)$$

Режим 3. Этот режим аналогичен предыдущему за исключением скорости передачи. Скорость приема/передачи определяется частотой переполнения TC1 и значением бита SMOD регистра PCON: $BR_3 = BR_1$.

Как может использоваться 9-битный режим UART?

1. Для организации 2-х стоп-бит (требуется, например, в интерфейсе MODBUS). В этом случае бит TB8 до передачи устанавливается в «1». При приеме 9-й бит помещается в бит RB8. В результате 9-й бит плюс штатный стоп-бит образуют во времени 2 стоп-бита.

2. Для помехоустойчивого кодирования с контролем на четность или нечетность. В этом случае 9-й бит используется как контрольный бит, значение которого при контроле на четность определяется из уравнения:

$$TB_8 \oplus Y = 0, \quad (4)$$

где TB_8 – значение бита TB8; \oplus – сложение по модулю 2; Y – результат сложения по модулю 2 всех разрядов передаваемого байта. Из уравнения (4) следует, что $TB_8 = Y$. Если используется контроль на нечетность, то правая часть уравнения (4) будет равна «1». Тогда TB_8 равно инверсному значению Y.

Как программно найти Y? Для этого передаваемый байт надо поместить в регистр ACC. После этого бит P (паритет) в регистре PSW примет значение «1» или «0», если в ACC нечетное или четное число единиц, т.е. в этом случае бит P определяет значение Y.

Практически тактовые частоты UART на приемной и передающей стороне могут не совпадать (см. рис. 5).

На сколько могут различаться частоты на приемной и передающей стороне?

Из рис. 5 видно, что в этом случае выборки сигнала будут смещаться относительно середины бита. Это смещение накапливается по мере приема битов. Поэтому худший случай будет иметь место в конце при приеме стоп-бита. Он может ложно определиться, если смещение составит 0,5 от длительности передачи бита. Т.к. обычно длительность всей передачи содержит 10 бит, то относительная погрешность расхождения частот на обеих сторонах в «%» составит $(0,5/10) * 100 \% = 5 \%$. Поэтому на каждой стороне с запасом 0,8 частоты могут отличаться от номинальной не более, чем на 2 %.

Часто последовательный порт тактируется от TC1. В этом случае рассчитывается **целочисленное значение** TN_1 по формуле (2), где частота F_{BQ} делится на 12, BR_1 и 32.

Однако в результате этого деления может получиться нецелое число, что потребует округления, которое может привести к недопустимой погрешности BR_1 (более 2 %).

Эта проблема решается на следующем слайде.

Стандартные интерфейсы, использующие UART, обычно работают на стандартизованных скоростях передачи: 1200, 4800, 9600, 19200 бит/с и т.д.

Очевидное решение рассматриваемой проблемы – устранить округление, т.е. чтобы значение F_{BQ} делилось без остатка на 12, 32 (2 в степени) и стандартизованную скорость BR_1 . Поскольку эта проблема давняя, связанная с последовательной передачей данных, то давно существуют кварцевые резонаторы с такими частотами. Например, в микропроцессорных система часто используют кварцевый резонатор с частотой 11,0592 МГц вместо 12 МГц. В следующей таблице представлены значения TH_1 и соответствующие относительные погрешности BR_1 для двух значений F_{BQ} .

BR_1 , бит/с	$F_{BQ} = 12 \text{ МГц}$		$F_{BQ} = 11,0592 \text{ МГц}$	
	TH_1	δ_{BR1} , %	TH_1	δ_{BR1} , %
1200	204	0,16	208	0
2400	243	0,16	232	0
4800	243	0,16	244	0
9600	249	-7,0	250	0
19200	253	8,5	253	0

Пример инициализации последовательного порта

```
;-----  
; Инициализация последовательного порта с параметрами:  
; BR1 = 4800 бит/с; Fbq = 24 МГц; формат 1-8-2.  
;-----  
TH_1      EQU      230      ; вычисляется с округлением по  
; формуле (2), SMOD = 1  
  
; Инициализация таймера-счетчика TC1  
  
InitTC1:   MOV     TMOD,     #00100000B ; режим 2  
           MOV     TL1,     #TH_1  
           MOV     TH1,     #TH_1  
           SETB   TR1             ; разрешить счет  
           RET  
  
; Инициализация UART  
  
InitUART:  ORL     PCON,     #80H ; SMOD = 1  
           MOV     SCON,     11011000B  
           RET  
  
; Инициализация последовательного порта  
; Эта подпрограмма вносится в список PUBLIC файла инициализации  
  
Init:      LCALL    InitTC1  
           LCALL    InitUART  
           RET
```

МУЛЬТИПРОЦЕССОРНАЯ СВЯЗЬ

Режимы 2 и 3 имеют специальное обеспечение для мультипроцессорной связи. В этих режимах принимается и передается 9 битов данных. При приеме девятый бит поступает в RB8. Затем приходит стоп-бит. Порт может быть запрограммирован так, что когда получен стоп-бит, то прерывание последовательного порта будет активизироваться только при RB8 = «1». Эта особенность разрешается установкой бита SM2 в регистре SCON. Использование этой особенности в МП-системах заключается в следующем.

Когда ведущий процессор хочет передать блок данных к одному из ведомых процессоров, то он, во-первых, посылает адресный байт, который идентифицирует ведомого адресата. Адресный байт отличается от байта данных тем, что 9-й бит = «1» в байте адреса и «0» при байте данных. При $SM2 = «1»$ ни один ведомый не будет прерываться байтом данных. Адресный байт, однако, прервет все ведомые, так что каждый из них может проверить полученный байт и увидеть: является ли он адресуемым. Адресованный подчиненный сбросит свой $SM2$ -бит и приготовится к приему байтов данных. Не адресованные подчиненные устройства оставляют в том же состоянии биты $SM2$ и переходят к выполнению своих задач, игнорируя поступающие байты данных.

Бит $SM2$ не оказывает никакого влияния в режиме 0, а в режиме 1 может быть использован для проверки получения стоп-бита. При приеме в режиме 1, если $SM2 = «1»$, прерывание приемника не будет активизировано пока не получен действительный стоп-бит.