

2.4. Лабораторная работа № 4.

Основы организации последовательного порта

Цель работы: научиться использовать последовательный порт МК для различных применений.

2.4.1. Основные сведения о режимах работы последовательного порта

В структуру МК-51 входит дуплексный канал последовательной связи с буферизацией, который может быть запрограммирован для работы в одном из четырех режимов:

- а) режим 0 – синхронный, последовательный ввод-вывод со скоростью $f_{osc}/12$;
- б) режим 1 – асинхронный с 10-битовым кадром и переменной скоростью передачи, зависящей от частоты переполнения таймера / счетчика 1 – T/C1;
- в) режим 2 – асинхронный с 11-битовым кадром и фиксированной скоростью передачи $f_{osc}/32$ или $f_{osc}/64$;
- г) режим 3 – асинхронный с 11-битовым кадром и переменной скоростью передачи, также определяемой частотой переполнения T/C1.

Принятые входные и передаваемые выходные данные в параллельном коде хранятся в буферном регистре SBUF, который располагается в пространстве SFR по адресу 99h. Управление работой приемопередатчиков осуществляется через слово управления и состояния SCON, расположенное в регистре по адресу 98h, имеющем следующую структуру (рис. 2.38):

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

Рис. 2.38. Структура регистра SCON

SCON.0 – RI. Флаг прерывания приемника.

SCON.1 – TI. Флаг прерывания передатчика.

SCON.2 – RB8. Восьмой бит приемника в режимах 2 и 3. В режиме 1, если SM2 = 0, то отображает стоп-бит. В режиме 0 не используется.

SCON.3 – TB8. Восьмой бит передатчика в режимах 2 и 3.

SCON.4 – REN. Разрешение приема.

SCON.5 – SM2. Запрещение приема кадров с нулевым восьмым битом данных. В режиме 0 должен быть сброшен.

SCON.6 – SM1. Младший разряд для кодирования номера режима.

SCON.7 – SM0. Старший разряд для кодирования номера режима.

Режим работы последовательного порта определяется следующим образом:

SM0	SM1	Режим	SM0	SM1	Режим
0	0	0	1	0	2
0	1	1	1	1	3

Биты SCON.0 – SCON.2 устанавливаются аппаратно, а сбрасываются программно, биты SCON.3–SCON.7 устанавливаются и сбрасываются программно.

В режиме 0 работы последовательного порта для синхронизации внешних устройств используется линия TxD (P3.0), по которой передаются синхроимпульсы, а прием и передача информации осуществляется по линии RxD (P3.1).

Скорость приема/передачи в режиме 0 определяется как

$$f_0 = f_{\text{осци}}/12,$$

где $f_{\text{осци}}$ – частота кварцевого резонатора.

За один машинный цикл МК последовательный порт передает один бит информации.

Скорость приема/передачи последовательного порта в режиме 2 определяется как

$$f_2 = (2^{\text{SMOD}}/64)f_{\text{осци}},$$

где бит SMOD является 7 битом регистра PCON (рис. 2.39).

Регистр PCON располагается в пространстве SFR по адресу 87h и полностью реализован в микросхемах КМОП технологии для управления режимом энергопотребления.

SMOD	–	–	–	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

Рис. 2.39. Структура регистра PCON

Назначение бит регистра следующее:

PCON.0 – IDL. Бит холостого кода. При PCON.0 = 1 МК переходит в режим холостого хода.

PCON.1 – PD. Бит пониженной мощности. При PCON.1 = 1 МК переходит в режим пониженного потребления мощности.

PCON.2 – GF0. Флаг, специфицируемый пользователем.

PCON.3 – GF1. Флаг, специфицируемый пользователем.

- PCon.4. Не используется.
 PCon.5. Не используется.
 PCon.6. Не используется.

PCon.7 – SMOD. Удвоенная скорость работы последовательного порта, если SMOD = 1.

Скорость передачи в режимах 1 и 3 определяется не только битом SMOD, но и частотой переполнения таймера-счетчика T/C1. При настройке порта на эти режимы работы необходимо запретить прерывания по переполнению таймера T/C1.

$$f_{1,3} = (2^{\text{SMOD}}/32)f_{\text{OVTI}},$$

где f_{OVTI} – частота переполнения таймера-счетчика T/C1. Наиболее удобно при работе последовательного порта использовать *второй режим работы T/C1 – режим 8-битного суммирующего счетчика с автоперезагрузкой*. При этом частота передачи определяется выражением:

$$f_{1,3} = (2^{\text{SMOD}}/32) \cdot (f_{\text{osc}}/12)/(256 - (\text{TH1})),$$

где (TH1) – содержимое регистра TH1 таймера счетчика T/C1.

Параметры настройки T/C1 для управления частотой работы последовательного порта представлены в табл. 2.12.

Таблица 2.12

Настройки МК для управления частотой последовательного порта

Частота	f_{osc} , МГц	SMOD	C/T1	Режим T/C1	TH1
Режим 0: 1 МГц	12	X*	X	X	X
Режим 2: 375 Кгц	12	1	X	X	X
Режимы 1, 3: 62,5 Кгц	12	1	0	2	0FFh
19,2 Кгц	11,059	1	0	2	0FDh
9,6 Кгц	11,059	0	0	2	0FDh
4,8 Кгц	11,059	0	0	2	0FAh
2,4 Кгц	11,059	0	0	2	0F4h
1,2 Кгц	11,059	0	0	2	0F8h
137,5 Гц	11,059	0	0	2	1Dh
110 Гц	6	0	0	2	72h

*Символ X обозначает безразличие в настройке соответствующего параметра.

Передача данных по последовательному порту инициируется всякий раз, когда новые данные заносятся в регистр SBUF, например

по команде *MOV SBUF, A*. Признаком окончания передачи служит установка флагка прерывания *T1*.

Операция приема данных активизируется только при установленном бите *REN = 1*, когда флагок *RI* сброшен. Установка флагка *RI* свидетельствует о готовности данных для считывания из регистра *SBUF*, тогда может быть использована, например, команда *MOVA, SBUF*.

2.4.2. Порядок выполнения лабораторной работы

Создаем схемный проект Circuit4, устанавливаем на рабочей области микроконтроллер MK-51. К выводу P3.0 подключаем осциллограф (рис. 2.40) для снятия сигнала при передаче информации.

Рассмотрим, например, ассемблерную программу управления последовательным портом в режиме передачи. Допустим, необходимо передать символы А, В, С со скоростью 4800 бит/с в асинхронном режиме 11-битным кадром (3 режим работы). При написании программы будем использовать режим прерываний последовательного порта при передаче символа. Передаваемые данные размещаются в программном коде, начиная с адреса 30h.

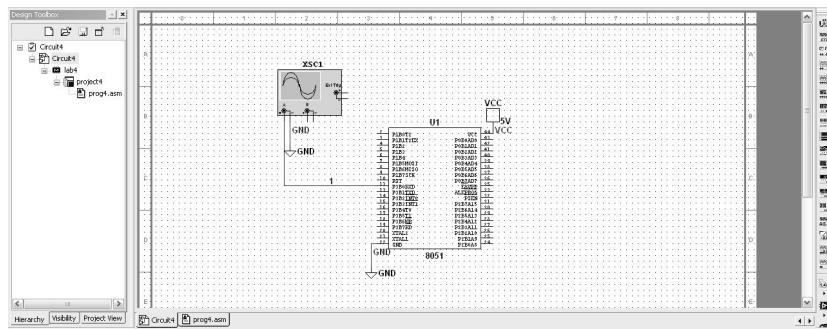


Рис. 2.40. Схема моделирования работы последовательного порта

Активируем вкладку программного файла *prog4.asm* и вставим следующий **ассемблерный код**:

```

$MOD51           ;подключение МК-51
org 00h
jmp start
org 23h          ;адрес начала процедуры прерываний по последовательному порту
clr TI           ;при входе в подпрограмму прерываний обнуляем флаг TI
inc r1            ;инкрементируем r1 – счетчик количества переданных символов
mov a,r1          ;пересылаем значение r1 в аккумулятор
subb a,#3h        ;проверка значения r1, чтобы не превысило трех символов
jnz metka        ;если не превышает, то идем на метку
mov r1,#00h       ;иначе обнуляем r1
metka:
reti             ;возврат из прерывания
org 30h
tabl:
db 61h           ;таблица данных для передачи
db 62h
db 63h
org 40h
start:
clr c            ;обнуляем бит carry
mov scon,#0C0h   ;настраиваемпорт на работу в режиме 3
mov tmod,#20h    ;устанавливаем счетчик T/C1 во второй режим работы
mov IE,#90h      ;разрешаем общее прерывание и прерывания от порта
setb IP.4        ;максимальный приоритет последовательного порта
mov t11,#0FAh    ;заносим в таймер число FAh, чтобы настроить порт
mov th1,#0FAh    ;скорость передачи 4800 бит/с
setb TR1          ;включаем T/C1
mov r1, #0h       ;число байт, подлежащих отправке
mov r0, #03h       ;заносим в dptr начальный адрес таблицы данных
mov dptr, #tabl
met1:

```

```

mov a,r1
movc a, @a + dptr
mov c, PSW.0      ;установить бит четности при передаче кадра
mov TB8, c        ;установить бит TB8 , равным биту четности
mov sbuf,a        ;отправляем символ таблицы на передачу
met:
jnb TI,met        ;ждем прерывание
djnz r0, met1    ;организация цикла передачи
END

```

Рассмотрим *пример проектирования С-программы* для передачи строки «privet iz Ekaterinburga» из последовательного порта при его работе во 2 режиме (11-битовый кадр, фиксированная скорость передачи 375 Кбит/с).

```

#include <8051.h>
//функция отправки символа по последовательному порту
void tput(unsigned char c1)
{
SBUF=c1;           //заносим символ в буфер передачи
while(!TI);        //ожидаем окончания передачи
TI=0;              //сбрасываем флаг окончания передачи
}
void main()
{
char z;
int i;
//объявляем и инициализируем передаваемую строку
unsigned char src[ ]={"privet iz Ekaterinburga"};
//устанавливаем бит SMOD в 1, для того чтобы скорость приема / передачи равнялась 1/32 частоты кварцевого резонатора
PCON=0x80;
for(i=0; i<24; i++)
{
    ACC=src[i];          //заносим текущий символ в аккумулятор
    // число единичных бит в аккумуляторе нечетное?
    if(!(PSW&&0x01))
    {
//включаем асинхронный 9-ти битовый режим работы последовательного порта, бит четности сброшен
z = 0x80;

```

```

}

else
{
//включаем асинхронный 9-ти битовый режим работы последователь-
ного порта, бит четности установлен
z = 0x88;
}
//заносим значение в регистр управления режимом приемопередатчи-
ка
SCON = z;
//передаем текущий символ в функцию отправки
tput (src[i]);
}
while(1){}           //бесконечный цикл
}

```

2.4.3. Задания для лабораторной работы

Допустим, необходимо принять/передать (R/T) N байт информации, настроив последовательный порт на K-режим работы со скоростью обмена S Кбит/с. Требуемые для передачи байты находятся в резидентной памяти данных, начиная с адреса XX.

В режиме приема XX – начальный адрес резидентной памяти данных, где размещаются принятые байты. При моделировании приема будем использовать два микроконтроллера МК-51, один из которых передает информацию в заданном режиме работы, а другой принимает.

Варианты заданий на лабораторную работу представлены в табл. 2.13.

Таблица 2.13

Варианты задания

Параметры	1	2	3	4	5	6	7	8	9	10
R/T	T	T	R	T	R	R	T	T	R	T
K	0	1	2	3	0	1	2	3	1	1
S, Кбит/с	1000	19,2	375	2,4	1000	4,8	187,5	62,2	9,6	1,2
XX	50h	30h	40h	50h	40h	60h	30h	40h	30h	50h
N	10	6	15	8	20	10	20	15	20	10

2.4.4. Содержание отчета

1. Наименование и цель работы.
2. Описание особенностей работы последовательного порта МК в режиме, используемом в задании.
3. Копия схемного файла во время моделирования с указанием позиционных обозначений элементов.
4. Копия программного файла (на ассемблере или на С) с подробными комментариями.
5. Полученные результаты и выводы по работе.

2.4.5. Вопросы для самоконтроля

1. Чем отличается синхронный протокол работы последовательного порта от асинхронного (старт-стопного)?
2. Может ли последовательный порт МК-51 одновременно передавать и принимать данные? Ответ обосновать.
3. Какие программируемые регистры в области памяти SFR имеют последовательный порт? Расскажите об их назначении и адресах.
4. Каково назначение бита контроля, используемого при приеме/передаче информации? В каких режимах работы последовательного порта он используется?
5. Как программно получить бит контроля при передаче информации 11-битным кадром?
6. Как проверить правильность приема символа 11-битным кадром?
7. В чем отличие программно-управляемого режима работы последовательного порта от режима прерывания?
8. Дайте понятие вектора прерывания. Какой вектор прерывания имеет последовательный порт?
9. Почему при приеме и передаче используется один и тот же вектор прерывания?
10. Какие средства имеет МК для разрешения/запрета прерываний от источников и изменения приоритета?