SP2327/8DP 数据应用手册

1、概述

SP2327/8DP 系采用低功耗 CMOS 工艺设计的 UART 多串口扩展芯片，可实现将一个较高波特率 UART 串口扩展为三个较低波特率 UART 串口。它主要是为解决目前大多数 8 位和 16 位单片机 UART 接口太少（绝大多数都只有一个 UART 口），而特别设计的专用串口扩展芯片。同时，该芯片也很好的解决了许多使用双串口单片机的串口配置问题，明显缩短开发周期，降低开发成本和生产成本。

2、特性

- 宽工作电压：2．4V~5．5V。
- 低工作电流：典型电流为 3．7mA（三个子串口都为：4800bps）。
- 宽工作速率：75bps~4800bps。
- 波特率设置简单：不需软件设置只需更改输入时钟频率即可。
- 四个 UART 串口都为全双工异步工作模式。
- 具有节电模式：进入节电模式后典型静态电流小于 1．2uA。
- 任意一个 RX 端口有数据出现时自动唤醒 SP2327/8DP。
- 每个串口的数据输出波特率误差小：典型误差小于 0.375%。
- 每个串口的数据接收波特率范围宽：小于 2.5%时所有数据能可靠接收。
- 数据传输误码率极低：小于 10^{-9}(接收的数据波特率误差小于 2%时)。

3、应用领域

- 采用电池供电的数据采集或通信设备。
- 集中监视及控制系统或设备。
- 高稳定性低速 MODEM 池（多路 MODEM）应用。
- 取代部分多串口卡应用（降低系统成本，减少串口连接电缆数量提高系统稳定性和可靠性）。
- 其他对成本、功耗敏感的多路数据传输应用。
- 多路数据需要同时进行电气隔离传输的数据传输应用。

4、引脚说明

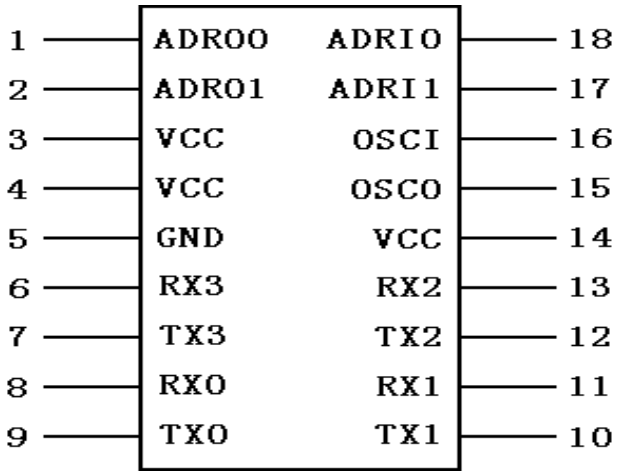


图 1　DIP 和 SOIC 封装

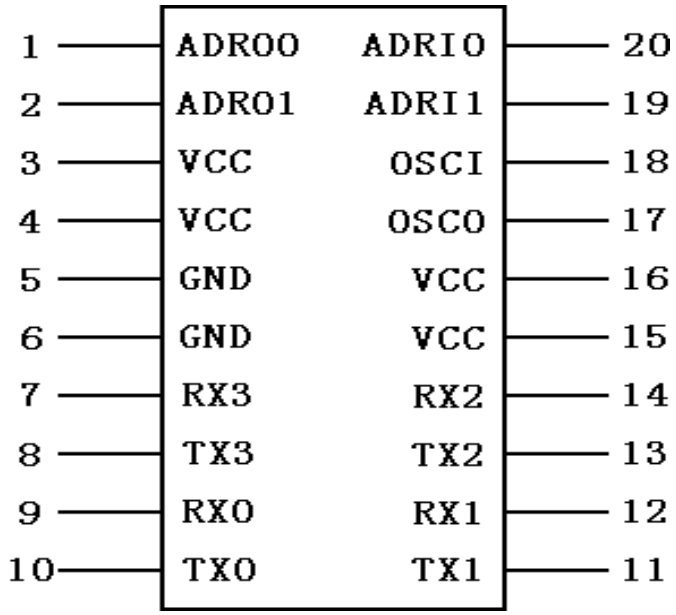


图 2 SSOP 封装

表 1

管脚名称	管脚编号 (DIP、SOIC)	管脚编号 (SSOP)	管脚类型	管脚描述
ADRI0	18	20	I	串口 3 发送数据地址线 0
ADRI1	17	19	I	串口 3 发送数据地址线 1
ADRO0	1	1	O	串口 3 接收数据地址线 0
ADRO1	2	2	O	串口 3 接收数据地址线 1
RX0	8	9	I	串口 0 数据接收
TX0	9	10	O	串口 0 数据发送
RX1	11	12	I	串口 1 数据接收
TX1	10	11	O	串口 1 数据发送
RX2	13	14	I	串口 2 数据接收
TX2	12	13	O	串口 2 数据发送
RX3	6	7	I	串口 3 数据接收
TX3	7	8	O	串口 3 数据发送
OSCI	16	18	I	时钟输入
OSCO	15	17	O	时钟输出
VCC	3、4、14	3、4、15、16	---	电源
GND	5	5、6	---	参考地

5、设计选型

SP232 X XX

A B C

表 2

代码	内容
A	一个 UART 串口扩展为三个 UART 串口芯片系列
B	7：发送、接收的数据位为 7 位数据 8：发送、接收的数据位为 8 位数据
C	DP：双列直插封装 SO：双列宽体贴片封装 SS：双列缩小窄体贴片封装

6、功能框图

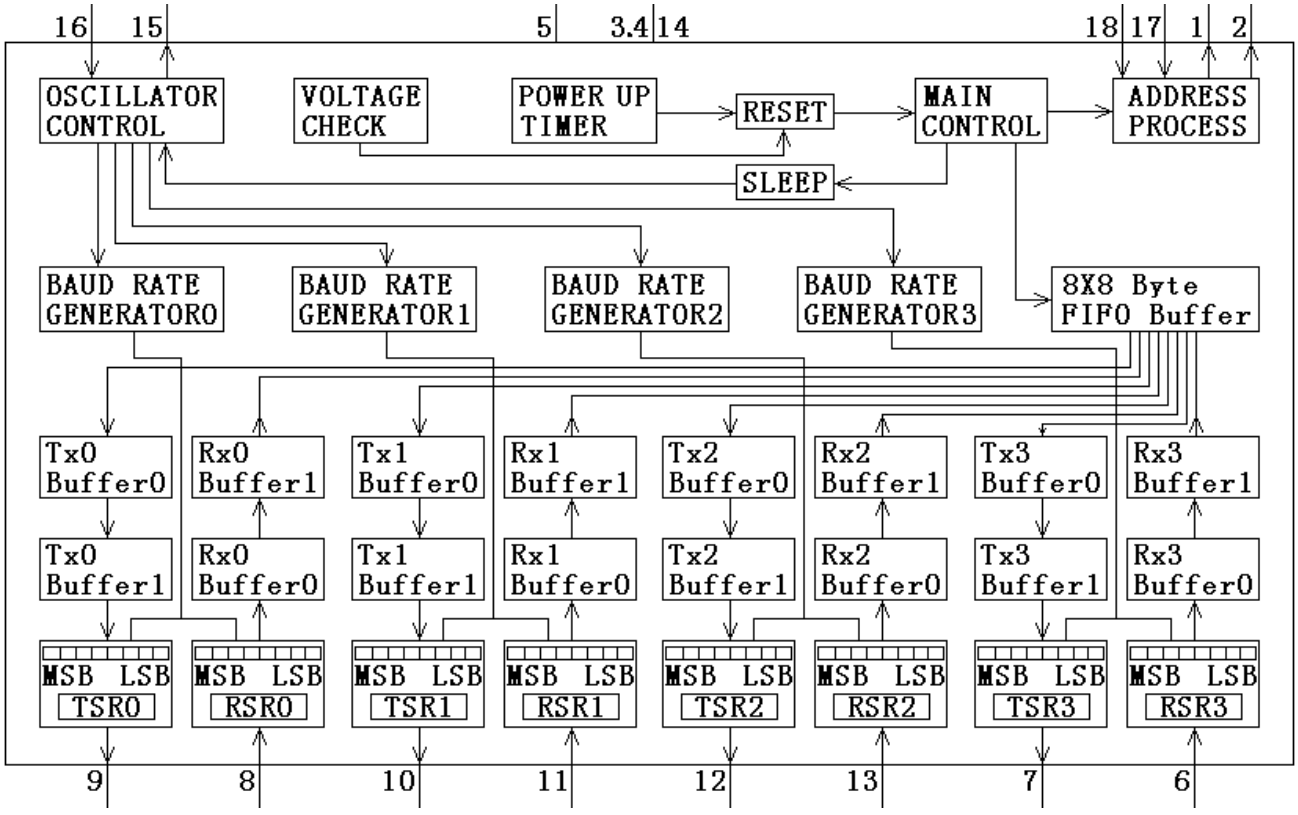


图 3 内部功能框图

7、应用说明

- SP2327DP 适用于 7 位数据位的应用，SP2328DP 适用于 8 位数据位的应用。
- 串口 0~串口 2 为三个较低波特率的子串口。
- 串口 3 为较高波特率的母串口，它的数据传输速率是子串口的 4 倍。例如：如输入的时钟频率为 4.00MHz，则串口 3 的波特率为 4800bps，串口 0~串口 2 的波特率为 4800bps/4=1200bps；如果需要在串口 0~串口 2 上获得波特率 K，则只需按下面的公式：“输入的时钟频率=4.00*K/1200”，改变输入时钟频率即可。
- ADRI1、ADRI0 为下行地址线，“00”、“01”、“10”分别对应三个子串口：串口 0~串口 2；地址线“11”为串口 3 的地址，它也是 SP2327/8DP 等的功能配置地址；
- 具体应用参见例子：如果上位机需要将数据“0X28”由串口 2 发送出去，则需要先将 ADRI1 置为“1”、ADRI0 置为“0”，再将数据“0X28”通过上位机的 UART 口送出即可。向串口 3（地址为“11”）写入数据：“0X35”或“0XB5”将实现芯片软件复位（复位时间为 21.75mS），如果写入的数据为“0X55”或“0XD5”则芯片将进入“Sleep”模式。
- 芯片“Wake up”条件为：向串口 0~串口 3 中的任意一个串口发送数据（由于 SP2327/8XX 的唤醒时间需要 25mS 左右，用于芯片唤醒的数据将不能够被接收。建议芯片唤醒处理流程：先发送一个用于唤醒芯片的数据（建议发送“0X63”），延时至少 25mS 后即可进行有效的数据传输）。

注：由于串口 3 的波特率是串口 0、串口 1、串口 2 的波特率的 4 倍，同时由于没有数据发送完标志，为了快速可靠的传输大量的数据可以采用下面的方法完全有效的解决波特率不匹配的问题：

① 如上位机只有数据需要由串口 1 发送，则可先向串口 2 发送完有效的数据（地址 2 为“01”）后再向地址“11”（串口 3 的地址）连续发送三个字节的无用数据（如：“0XFF”，但不能发送以下的四个数据“0X35”、“0X55”、“0XB5”、“0XD5”），其后再发送下一个有效的数据，再向串口 3 发送三个字节无用数据……，以此方式循环发送有效数据即可；

② 如果上位机有数据需要同时向两个子串口发送，则可以先分别向两个子串口发送有效的数据后，再向串口 3 发送两个字节的无用数据，再循环发送两个子串口的有效数据即可；

③ 如果上位机有数据需要同时向三个子串口发送，则可以在分别发送完三个子串口的有效数据后再向串口 3 发送一个字节的无用数据，再循环发送三个子串口的有效数据即可。

具体应用可以参见后面我公司提供的例子程序（在应用中为了节省单片机有限的资源，提高数据传输速率和可靠性，建议不要采用定时中断方式处理上位机串口数据发送：如果定时中断时间太长则数据的有效传输速率将降低，如果定时中断时间太短，将发生数据丢失问题）。

- ADRO1、ADRO0 为上行数据的串口地址线，“00”、“01”、“10”分别对应串口 0、串口 1、串口 2。
- 串口 3 接收数据处理：当上位机的 UART 接收到由串口 3 送来的数据时，立即读取地址线 ADRO1 和 ADRO0 的状态，根据两个地址线的状态即可判断接收到的数据是由哪个串口上传的。

8、SP2327/8XX 极限参数

- 工作温度：0°C~70°C 或-40°C ~85°C 可选。
- 存储温度： -65°C~125°C。
- 最高工作电压：6.0V。

9、直流电气特性

（测试温度：0°C~70°C，VCC=5.0V±5%，GND=0V）

表 3

特性	最小值	典型值	最大值	单位	测试条件
工作电压	2.4	—	5.5	V	
输入低电平	GND	—	0.8	V	
输入高电平	2.0		VCC	V	
输入漏电流	-1.0	0	+1.0	u A	Input pin at VCC or GND
输出低电平	—	—	0.6	V	I_{OL}=5.0mA
输出高电平	3.7	—	—	V	I_{OH}=4.0 mA
输入时钟 1	DC	—	20.0	MHz	VCC=5.0V
输入时钟 2	DC	—	16.0	MHz	VCC=3.3V
工作电流 1	—	1.2	1.7	mA	F_{osc}=4.0MHz Input pin at VCC Output pin floating
工作电流 2	—	3.7	6.3	mA	F_{osc}=16.0MHz Input pin at VCC Output pin floating
睡眠电流	—	0.5	1.2	u A	Input pin at VCC Output pin floating

10、 典型应用电路

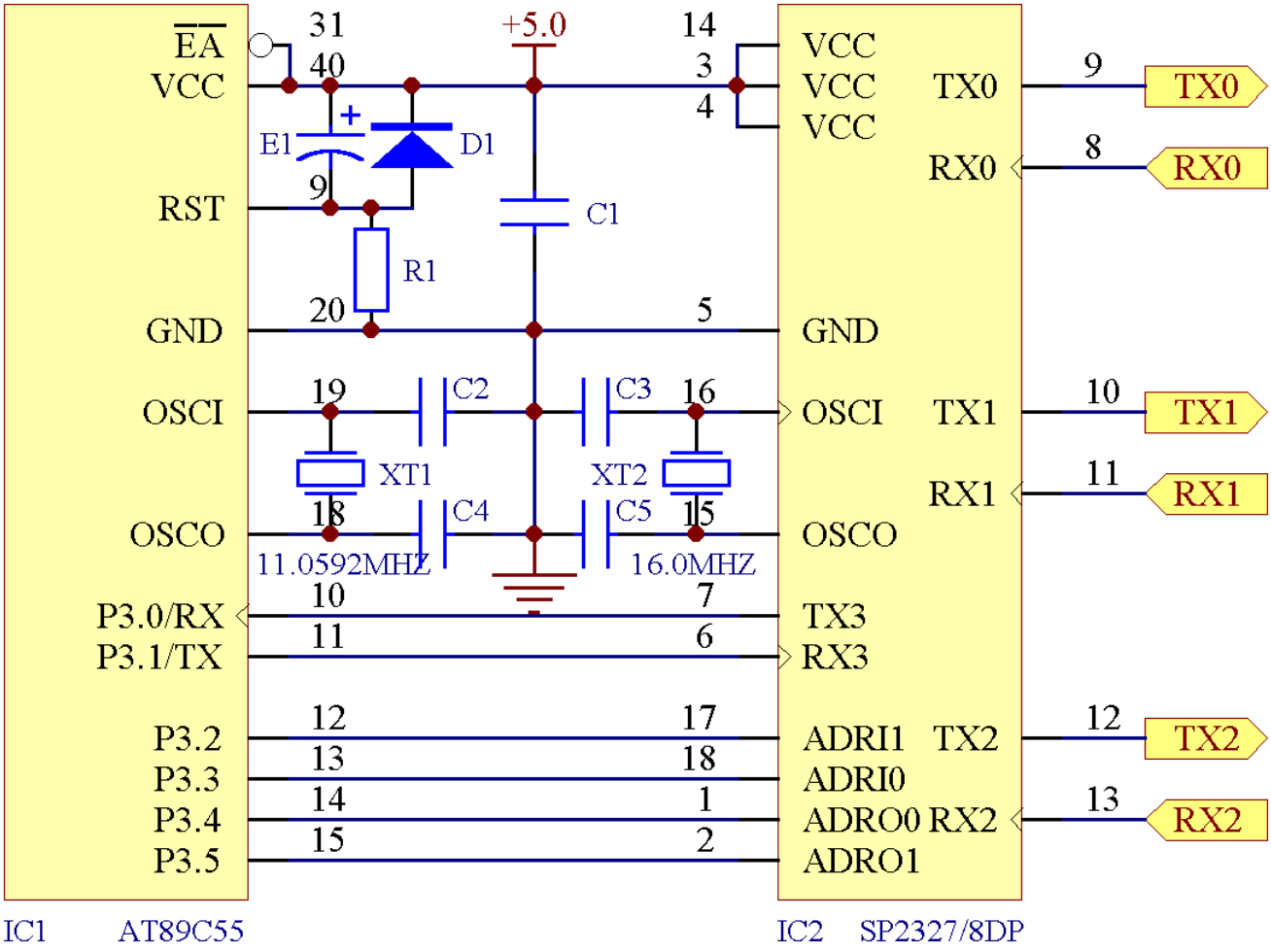
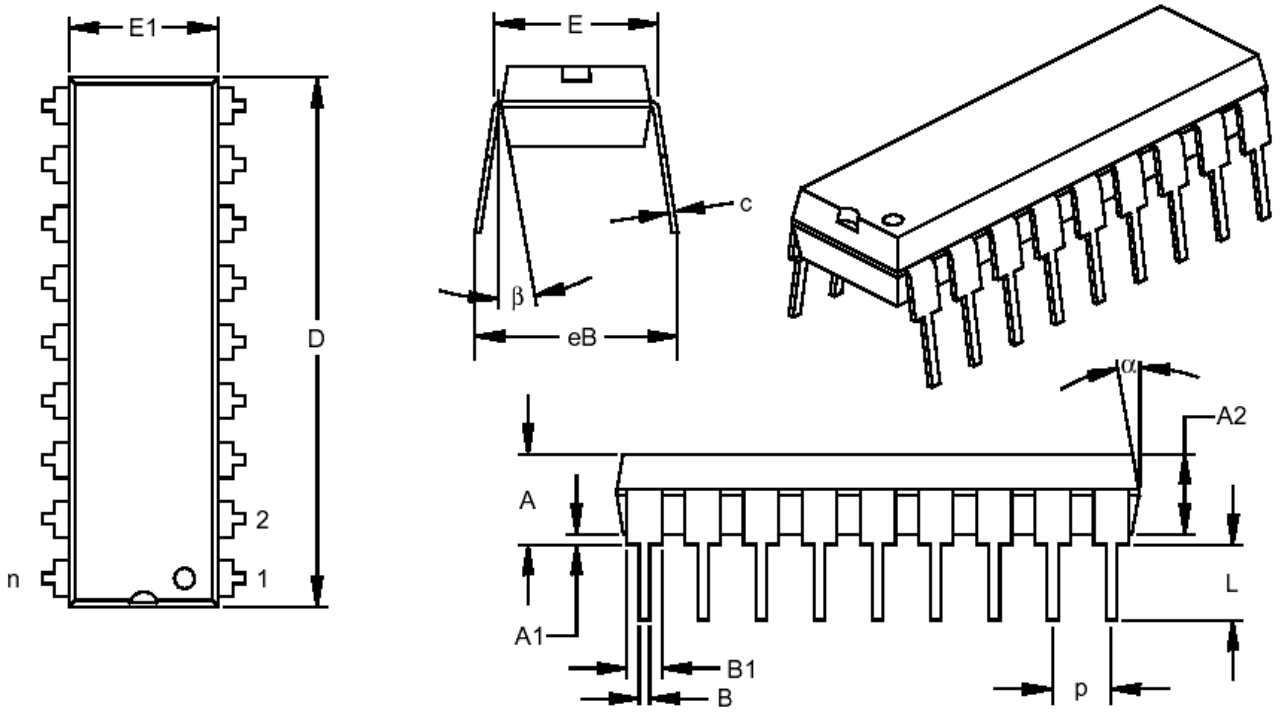

EA
VCC
RST
GND
OSCI
OSCO
P3.0/RX
P3.1/TX
P3.2
P3.3
P3.4
P3.5
31
40
9
20
19
18
10
11
12
13
14
15
+5.0
E1
D1
R1
C1
C2
C3
C4
C5
XT1
XT2
11.0592MHZ
16.0MHZ
IC1 AT89C55
14
3
4
5
16
15
7
6
17
18
1
2
VCC
VCC
VCC
GND
OSCI
OSCO
TX3
RX3
ADRI1
ADRI0
ADRO0
ADRO1
TX0
RX0
TX1
RX1
TX2
RX2
9
8
10
11
12
13
TX0
RX0
TX1
RX1
TX2
RX2
IC2 SP2327/8DP

图 4

11、 封装信息

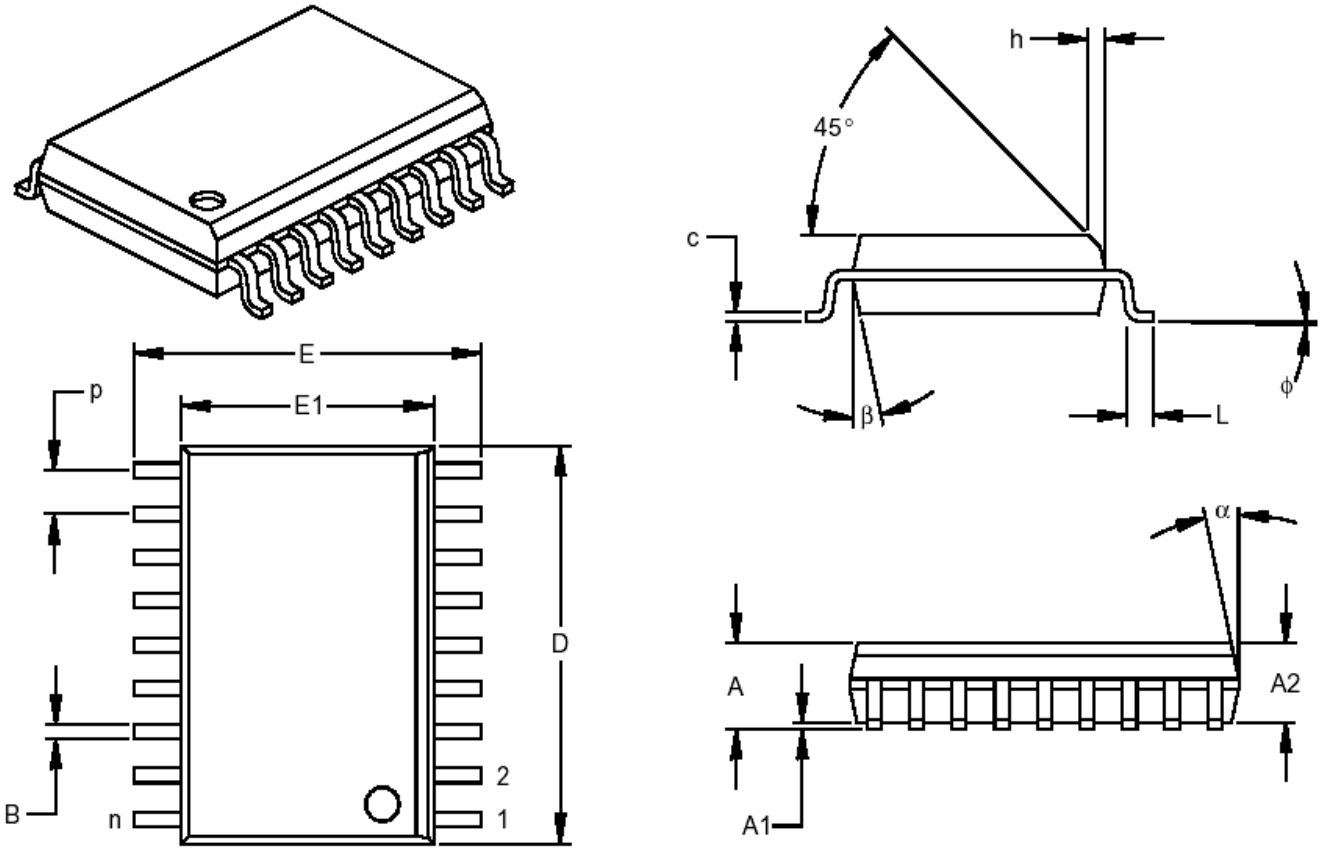
- DIP 封装



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

图 5 DIP 封装数据

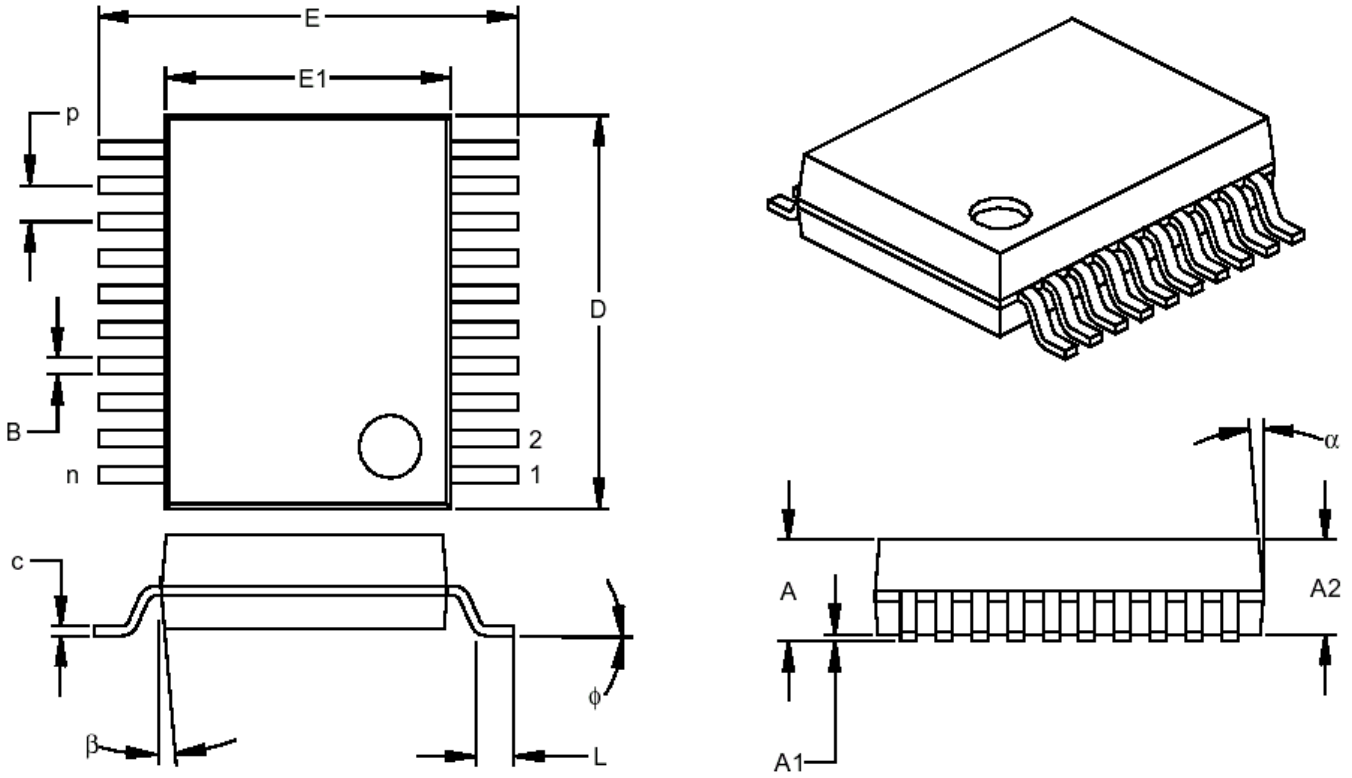
- SIOC 封装



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff §	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.291	.295	.299	7.39	7.49	7.59
Overall Length	D	.446	.454	.462	11.33	11.53	11.73
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	ϕ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.012	0.23	0.27	0.30
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

图 6 SOIC 封装数据

- SSOP 封装



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		20			20	
Pitch	p		.026			0.65	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff §	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.322	7.59	7.85	8.18
Molded Package Width	E1	.201	.207	.212	5.11	5.25	5.38
Overall Length	D	.278	.284	.289	7.06	7.20	7.34
Foot Length	L	.022	.030	.037	0.56	0.75	0.94
Lead Thickness	c	.004	.007	.010	0.10	0.18	0.25
Foot Angle	ϕ	0	4	8	0.00	101.60	203.20
Lead Width	B	.010	.013	.015	0.25	0.32	0.38
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

图 7 SSOP 封装数据

例子程序：

```
/*=============================================================
REG51.H
Header file for generic 80C51 and 80C31 microcontroller.
Copyright (c) 1988-2001 Keil Elektronik GmbH and Keil Software, Inc.
All rights reserved.
=============================================================*/
/*  BYTE Register  */
sfr P0   = 0x80;
sfr P1   = 0x90;
sfr P2   = 0xA0;
sfr P3   = 0xB0;
sfr PSW  = 0xD0;
sfr ACC  = 0xE0;
sfr B    = 0xF0;
sfr SP   = 0x81;
sfr DPL  = 0x82;
sfr DPH  = 0x83;
sfr PCON = 0x87;
sfr TCON = 0x88;
sfr TMOD = 0x89;
sfr TL0  = 0x8A;
sfr TL1  = 0x8B;
sfr TH0  = 0x8C;
sfr TH1  = 0x8D;
sfr IE   = 0xA8;
sfr IP   = 0xB8;
sfr SCON = 0x98;
sfr SBUF = 0x99;

/*  BIT Register  */
/*  PSW   */
sbit CY   = 0xD7;
sbit AC   = 0xD6;
sbit F0   = 0xD5;
sbit RS1  = 0xD4;
sbit RS0  = 0xD3;
sbit OV   = 0xD2;
sbit P    = 0xD0;

/*  TCON  */
sbit TF1  = 0x8F;
sbit TR1  = 0x8E;
sbit TF0  = 0x8D;
sbit TR0  = 0x8C;
sbit IE1  = 0x8B;
sbit IT1  = 0x8A;
sbit IE0  = 0x89;
```

```
sbit IT0  = 0x88;
/*  IE   */
sbit EA   = 0xAF;
sbit ES   = 0xAC;
sbit ET1  = 0xAB;
sbit EX1  = 0xAA;
sbit ET0  = 0xA9;
sbit EX0  = 0xA8;

/*  IP   */
sbit PS   = 0xBC;
sbit PT1  = 0xBB;
sbit PX1  = 0xBA;
sbit PT0  = 0xB9;
sbit PX0  = 0xB8;

/*  P3  */
sbit RD   = 0xB7;
sbit WR   = 0xB6;
sbit T1   = 0xB5;
sbit T0   = 0xB4;
sbit INT1 = 0xB3;
sbit INT0 = 0xB2;
sbit TXD  = 0xB1;
sbit RXD  = 0xB0;

/*  SCON  */
sbit SM0  = 0x9F;
sbit SM1  = 0x9E;
sbit SM2  = 0x9D;
sbit REN  = 0x9C;
sbit TB8  = 0x9B;
sbit RB8  = 0x9A;
sbit TI   = 0x99;
sbit RI   = 0x98;

/*============================================================
SP2328XX.c  head file SP2328XX serial communication example.
Author: Chen Wei, 2002-4-22
============================================================*/

#ifndef __SP2328XX_
#define __SP2328XX_

/*structure of recieved data from SP2328XX*/
struct REC_DATA
{
        unsigned char m_PortIdx; /* index of sub serial port of SP2328XX,range of 0~2 */
        char m_Data;                /* data of corresponding serial port */
};
```

```
/*********************************************************************
void InitSP2328XX(unsigned long nFosc, unsigned long nBRate)

initialize the SP2328XX, mainly set the baud rate of host's serial port.
nFosc  -        frequency of crystal oscillatory.
nBRate-         baud rate
*********************************************************************/

void InitSP2328XX(unsigned long nFosc, unsigned long nBRate);

/*********************************************************************
void SP2328Reset(void)

reset all the serial port of SP2328XX.
*********************************************************************/

void SP2328Reset(void);

/*********************************************************************
void SP2328SetSleep(void)

set the SP2328XX go into sleep mode.
*********************************************************************/

void SP2328SetSleep(void);

/*********************************************************************
struct REC_DATA SP2328ReceiveData(void)

get one byte data from SP2328XX, store the data
into a structure as struct REC_DATA.

member m_PortIdx of the structure is the index of sub serial port of SP2328XX,
and member m_Data of it is the data of the corresponding sub serial port of SP2328XX
*********************************************************************/
struct REC_DATA SP2328ReceiveData(void);

/*********************************************************************
void SP2328SendData(char sdata, unsigned char nPortIdx)

send one byte data to a sub serial port of SP2328XX.
sdata   -       the data that sended.
nPortIdx        -       the index of sub serial port of SP2328XX.
*********************************************************************/

void SP2328SendData(char sdata, unsigned char nPortIdx);

#endif /* __SP2328XX_ */
```

```
/*=================================================================
SP2328XX.c  source file SP2328XX serial communication example.
for MCS51 MCU
Author: Chen Wei, 2002-4-22
=================================================================*/
#include <reg51.h>
#include "SP2328XX.h"

/* 1- compile with main(), 0- compile without main() */
#define _SUB_MODULE_DEBUG_ 0
#define ABS(val) (((val)>=0)?(val):(0-(val)))

sbit ADRO_0  = P3^4          ; /* connect to ping ADRO0 of SP2328XX */
sbit ADRO_1  = P3^5          ; /* connect to ping ADRO1 of SP2328XX */
sbit ADRI_0   = P3^3         ; /* connect to ping ADRI0  of SP2328XX */
sbit ADRI_1   = P3^2         ; /* connect to ping ADRI1  of SP2328XX */

/*****************************************************************
void InitSP2328XX(unsigned long nFosc, unsigned long nBRate)
initialize the SP2328XX, mainly set the baud rate of host's serial port.
nFosc  -        frequency of crystal oscillatory.
nBRate-         baud rate
*****************************************************************/
void InitSP2328XX(unsigned long nFosc, unsigned long nBRate)
{
        unsigned char TH_VAL_SMOD0,TH_VAL_SMOD1;
        float B_RATE_SMOD0,B_RATE_SMOD1;
        float B_RATE_ERR_SMOD0,B_RATE_ERR_SMOD1;
        unsigned char SMOD_VAL,TH_VAL,TL_VAL;

        ADRO_0 = 1;
        ADRO_1 = 1;
        ADRI_0 = 1;
        ADRI_1 = 1;

      TH_VAL_SMOD0 = 256-(unsigned char)(((float)(1*nFosc))/((float)(32*12*nBRate)));
      TH_VAL_SMOD1 = 256-(unsigned char)(((float)(2*nFosc))/((float)(32*12*nBRate)));

        B_RATE_SMOD0 = ((float)(1*nFosc))/((float)((256-TH_VAL_SMOD0)*32*12));
        B_RATE_SMOD1 = ((float)(2*nFosc))/((float)((256-TH_VAL_SMOD1)*32*12));

        B_RATE_ERR_SMOD0 = ABS(B_RATE_SMOD0-nBRate);
        B_RATE_ERR_SMOD1 = ABS(B_RATE_SMOD1-nBRate);

        if( B_RATE_ERR_SMOD0 <= B_RATE_ERR_SMOD1 )
        {
                SMOD_VAL = 0;
                TH_VAL = TH_VAL_SMOD0;
        }
```

```
        else{
                SMOD_VAL = 1;
                TH_VAL = TH_VAL_SMOD1;
            }

        TL_VAL = TH_VAL;

        /* timer 1 */
        TMOD=TMOD & 0x0f;TMOD=TMOD | 0x20;
        TH1=TH_VAL;
        TL1=TL_VAL;
        TR1=1;

        /* serial port */
        PCON=(PCON&0x7f)|(SMOD_VAL<<7);  /* set value of SMOD bit */
        SM0=0;
        SM1=1;                           /* mode 1: 10 bit async */
        SM2=0;
        REN=1;                           /* enable receive */

        /* interrupt */
        ES=0;

        /*EA=0;*/
        SP2328Reset();

}

/*********************************************************************
void SP2328Reset(void)

reset all the serial port of SP2328XX.
*********************************************************************/
void SP2328Reset(void)
{

        TI = 0;
        ADRO_0 = 1;
        ADRO_1 = 1;
        SBUF = 0x35;
        while(!TI);

}
```

```
/*****************************************************************
void SP2328SetSleep(void)

set the SP2328XX go into sleep mode.
*****************************************************************/
void SP2328SetSleep(void)
{

        TI = 0;
        ADRO_0 = 1;
        ADRO_1 = 1;
        SBUF = 0x55;
        while(!TI);

}

/*****************************************************************
struct REC_DATA SP2328ReceiveData(void)

get one byte data from SP2328XX, store the data
into a structure as struct REC_DATA.

member m_PortIdx of the structure is the index of sub serial port of SP2328XX,
and member m_Data of it is the data of the corresponding sub serial port of SP2328XX
*****************************************************************/
struct REC_DATA SP2328ReceiveData(void)
{
        struct REC_DATA rdata;

        while(!RI);
        rdata.m_Data = SBUF;
        rdata.m_PortIdx = (((unsigned char)ADRO_1) << 1) | ((unsigned char)ADRO_0);

        RI = 0;

        return(rdata);
}
```

```
/*****************************************************************
void SP2328SendData(char sdata, unsigned char nPortIdx)

send one byte data to a sub serial port of SP2328XX.
sdata   -       the data that sended.
nPortIdx        -       the index of sub serial port of SP2328XX.
*****************************************************************/
void SP2328SendData(char sdata, unsigned char nPortIdx)
{
        static unsigned char nPortFlag[3] = {0, 0, 0};

        /* delay if the port index is the previous port */
        if(nPortIdx < 3)
        {
                while(nPortFlag[nPortIdx])
                {
                        TI = 0;
                        ADRO_0 = 1;
                        ADRO_1 = 1;
                        SBUF = 0;
                        while(!TI);

                        if(nPortFlag[0]) nPortFlag[0]--;
                        if(nPortFlag[1]) nPortFlag[1]--;
                        if(nPortFlag[2]) nPortFlag[2]--;

                }

                TI = 0;

                ADRO_0 = nPortIdx & 0x01;
                ADRO_1 = (nPortIdx >> 1) & 0x01;
                SBUF = sdata;

                while(!TI);

                if(nPortFlag[0]) nPortFlag[0]--;
                if(nPortFlag[1]) nPortFlag[1]--;
                if(nPortFlag[2]) nPortFlag[2]--;
                nPortFlag[nPortIdx] = 3;
                /* because the port send data just now, delay 3 times of send data. */
        }
        else;

}
```

```
#if _SUB_MODULE_DEBUG_

void main(void)
{
        struct REC_DATA r;

        InitSP2328XX(11059200L, 19200);

        r = SP2328ReceiveData();

        SP2328SendData('0', 0);
        SP2328SendData('1', 1);
        SP2328SendData('2', 2);
        SP2328SendData('0', 0);
        SP2328SendData('1', 1);
        SP2328SendData('2', 2);

        SP2328SendData('2', 2);

        SP2328SendData('1', 1);
        SP2328SendData('1', 1);

        SP2328SendData('0', 0);
        SP2328SendData('0', 0);

        SP2328SetSleep();

        while(1);
}

#endif /* _SUB_MODULE_DEBUG_ */
```

成都思普科技有限公司 SEPER TECHNOLOGY CO.，LTD
地址： 成都市高新区二环路南四段 18 号 3 号楼 112 房间 邮 编： 610041
TEL： 028-88074377 85138086 FAX：028-85138086 Mobile： 13980023800
E-mail: powerysp@sina.com