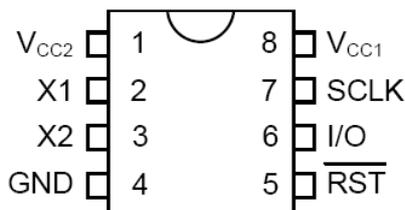


RoHS 认证, 绿色无铅封装  
封装类型: SOIC8/SOP8/DIP8

### 一、产品特点:

- 1、实时钟计秒、分、时、一个月中的日期、月、一周中的每天、到 2100 年的闰年;
- 2、31X8 RAM;
- 3、串行 IO 以节省引脚数目;
- 4、2~5.5V 全工作电压范围(2~5.5V 电压可选, 可完全单电池供电);
- 5、2.5V 下消耗小于 300nA 电流;
- 6、时钟或者 RAM 数据可以单字节或者多字节传输;
- 7、SOIC8、SOP8 或者 DIP8 封装;
- 8、简单的 3 线界面;
- 9、TTL 兼容;
- 10、可选工业级温度范围: -40~85 度。

### 二、引脚分配和说明:



外引脚序号	引脚名称	功能	I/O
1	VCC2	电源电压	P
2	X1	32.768kHz 晶振输入	I
3	X2	32.768kHz 晶振输出	O
4	GND	地	P
5	RST	复位	I
6	I/O	数据传输口	I/O
7	SCLK	数据传输时钟	I
8	VCC1	电源电压	

### 三、参数:

#### 1、极限参数:

# 每一个管脚到地电压: -0.5V~7.0V;

# 工作温度: 0~70 度;

# 存储温度: -55~125 度;

#### 2、推荐直流工作条件: (0~70 度)

# Sola IC

## SLM1302 Trickle Charge Timekeeping Chip

参数		最小	典型	最大	单位
VCC1, VCC2		2		5.5	V
逻辑 1 输入 VIH		2.0		VCC+0.3	V
逻辑 0 输入 VIL	VCC=2V	-0.2		+0.3	V
	VCC=5V	-0.3		+0.8	V

### 3、直流电参数: (0~70度) (VCC=2~5.5V)

参数	符号		最小	典型	最大	单位
输入漏电流	ILI				500	uA
I/O 漏电路	ILO				500	uA
逻辑 1 输出	VOH	VCC=2V	1.6			V
		VCC=5V	2.4			V
逻辑 0 输出	VOL	VCC=2V			0.4	V
		VCC=5V			0.4	V
工作电流	ICC1A	VCC=2V			0.3	mA
		VCC=5V			1.2	mA
时钟保持电流	ICC1T	VCC=2V			0.3	uA
		VCC=5V			1	uA
待机电流	ICC1S	VCC=2V		100		nA
		VCC=5V		100		nA
工作电流	ICC2A	VCC=2V			0.4	mA
		VCC=5V			1.28	mA
时钟保持电流	ICC2T	VCC=2V			25	uA
		VCC=5V			81	uA
待机电流	ICC2S	VCC=2V			25	uA
		VCC=5V			80	uA

### 4、交流参数: (0~70度)

参数	符号		最小	典型	最大	单位
时钟频率	fCLK	VCC=2V			0.5	MHZ
		VCC=5V			2	

#### 四、功能框图和简单功能描述:

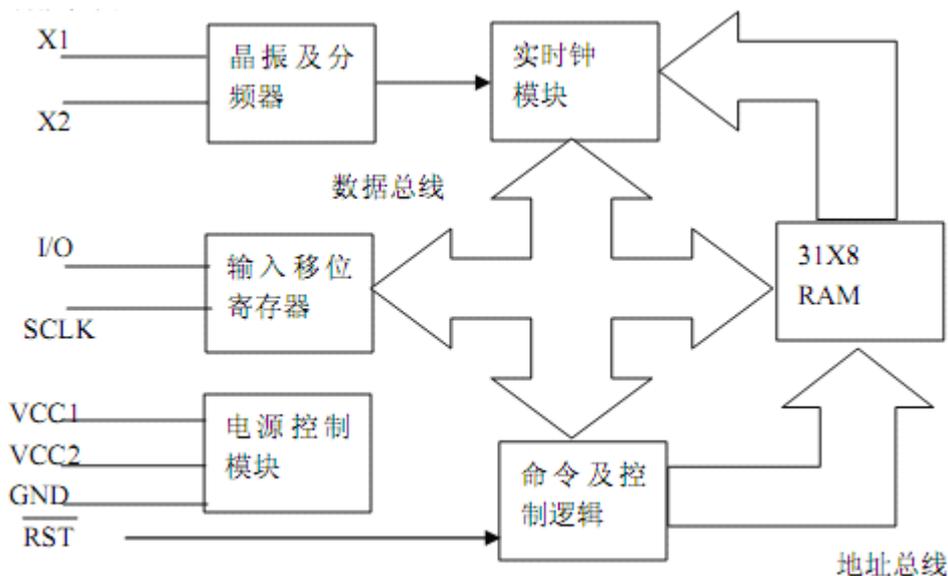
SLM1302 包含了一个时钟/日历实时钟以及 32 字节的静态 RAM, 它通过一个简单的串行界面和微控制器通讯; 时钟/日历实时钟提供了秒、分、时、天、月以及年的信息; 每个月的最后一天根据该月是否有 31 天而自动调节, 包括闰年的校正; 时钟可以是 12 小时或者 24 小时制, 并加以 AM 或者 PM 提示;

SLM1302 与微控制器之间的界面是一个简单的同步串行通讯接口, 只有 CLK、RST、和 I/O 三根线; SLM1302 采用低功耗设计。

# Sola IC

## SLM130Z Trickle Charge Timekeeping Chip

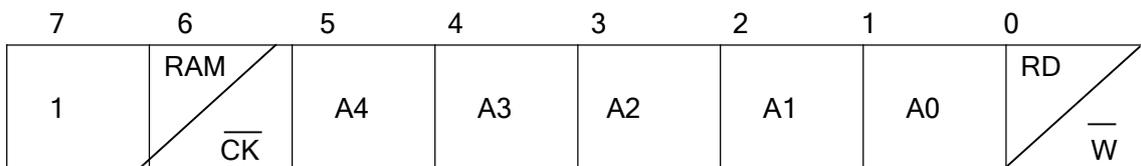
功能框图:



电路包括移位寄存器、控制逻辑模块、晶振、实时钟模块以及 RAM 等几个功能模块；在开始传递数据前， $\overline{\text{RST}}$  变为高，8 bits 数据进入移位寄存器，可以是地址或者命令信息；数据在 SCLK 上升沿串行输入；第一个 8 位数据用来制定哪 40 个字节数据将被操作，不管是读操作或者是写操作；也不管是单字节操作还是多字节操作。命令字节的输入是最低位 Bit0 (LSB) 先输入。

### 五、 主要模块工作原理:

#### 1、 命令字节格式:



每次数据传递都通过命令字节来进行初始化；最高位必须是“1”，如果是“0”的话写进 SLM130Z 将无效；bit6=0 表示时钟/日历数据，=1 表示 RAM 数据；bit1~bit5 指定目标寄存器地址；bit0=0 表示写操作，=1 表示读操作；

#### 2、 复位和时钟控制、数据输入/输出以及多字节模式:

$\overline{\text{RST}}$  引脚是电路的复位引脚，低电平有效。所有数据传输都必须设置  $\overline{\text{RST}}$  为高；一旦  $\overline{\text{RST}}$  是'0' 那么数据立即停止传输。在上电复位过程中， $\overline{\text{RST}}$  必须是'0'，直到  $V_{cc} \geq 2V$ ，同时 SCLK 也必须是'0'。 $\overline{\text{RST}}$  有两个功能：第一：启动控制逻辑；第二：提供一种方法来决定是单字节还是多字节传输；

数据输入

# Sola IC

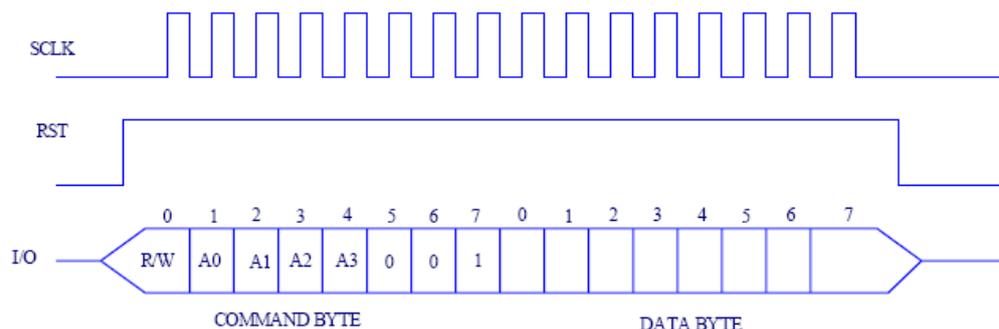
## SLM1302 Trickle Charge Timekeeping Chip

在前面 8 个 SCLK 时钟输入写命令字节后，在下一个 8 个 SCLK 上升沿输入数据。数据输入先从 bit0 开始。

数据输出

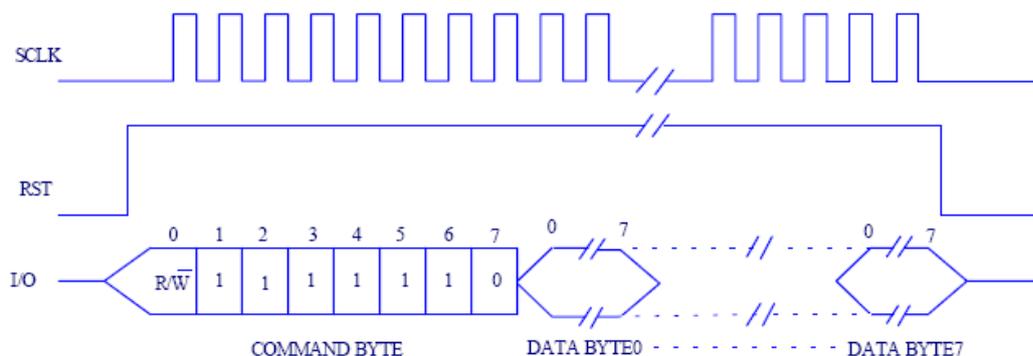
在前面 8 个 SCLK 时钟输入读命令字节后，数据在 SCLK 时钟的下降沿输出。数据输出从 bit0 开始。

数据传输时序图（单字节传输）：



注：上图 I/O 中 bit5 应为 A4，bit6 应为 R/C；

数据传输时序图（多字节传输）：



注：上图 I/O 中 bit6 应为 R/C；bit7 应为 1；

### 3、时钟/日历、时钟停止标志、AM/PM 12-24 小时模式以及写保护位：

时钟/日历信息以 BCD 格式包含在七个读/写寄存器中，见下图：

寄存器地址：

SEC	1	0	0	0	0	0	0	RD — W
-----	---	---	---	---	---	---	---	--------------

寄存器定义：

# Sola IC

## SLM1302 Trickle Charge Timekeeping Chip

00-59

CH	10	SEC	SEC
----	----	-----	-----

寄存器地址:

MIN

1	0	0	0	0	0	1	RD — W
---	---	---	---	---	---	---	--------------

寄存器定义:

00-59

0	10	MIN	MIN
---	----	-----	-----

寄存器地址:

HR

1	0	0	0	0	1	0	RD — W
---	---	---	---	---	---	---	--------------

寄存器定义:

01-12  
00-23

12/24	0	10 A/P	HR	HR
-------	---	-----------	----	----

寄存器地址:

DATE

1	0	0	0	0	1	1	RD — W
---	---	---	---	---	---	---	--------------

寄存器定义:

01-28/29  
01-30  
01-31

0	0	10	DATE	DATE
---	---	----	------	------

寄存器地址:

MONTH

1	0	0	0	1	0	0	RD — W
---	---	---	---	---	---	---	--------------

寄存器定义:

# Sola IC

## SLM1302 Trickle Charge Timekeeping Chip

01-12	0	0	0	10 M	MONTH		
-------	---	---	---	---------	-------	--	--

寄存器地址:

DAY	1	0	0	0	1	0	1	RD W
-----	---	---	---	---	---	---	---	---------

寄存器定义:

01-07	0	0	0	0	0	DAY	
-------	---	---	---	---	---	-----	--

寄存器地址:

YEAR	1	0	0	0	1	1	0	RD W
------	---	---	---	---	---	---	---	---------

寄存器定义:

00-99	10 YEAR				YEAR		
-------	---------	--	--	--	------	--	--

时钟停止标志:

秒寄存器的 bit7 用来定义时钟停止标志, 当这一位设置成 1 时, 时钟晶振定制, SLM1302 进入低功耗待机模式; 当这一位设置成 0 时, 时钟将启动。

AM/PM 12-24 模式:

小时寄存器的 bit7 用来定义 12 或者 24 小时制; 当这一位设置成 1 时, 选择 12 小时制; 在 12 小时制中, bit5 用来设置 AM 或者 PM (当为 1 时, 选择 PM); 在 24 小时制中, bit5 是第二个 10 小时位 (20~23 小时)。

写保护位:

寄存器地址:

1	0	0	0	1	1	1	RD W
---	---	---	---	---	---	---	---------

寄存器定义:

# Sola IC

## SLM1302 Trickle Charge Timekeeping Chip

WP	0	0	0	0	0	0	0
----	---	---	---	---	---	---	---

控制寄存器的 bit7 是写保护位，其它七位强制为 0；在时钟或者 RAM 写操作前，bit7 必须设置为 0，一旦设置为 1，任意寄存器的写操作将被禁止。

### 4、点滴式充电寄存器：

充电寄存器地址：

1	0	0	1	0	0	0	RD — W
---	---	---	---	---	---	---	--------------

充电寄存器定义：

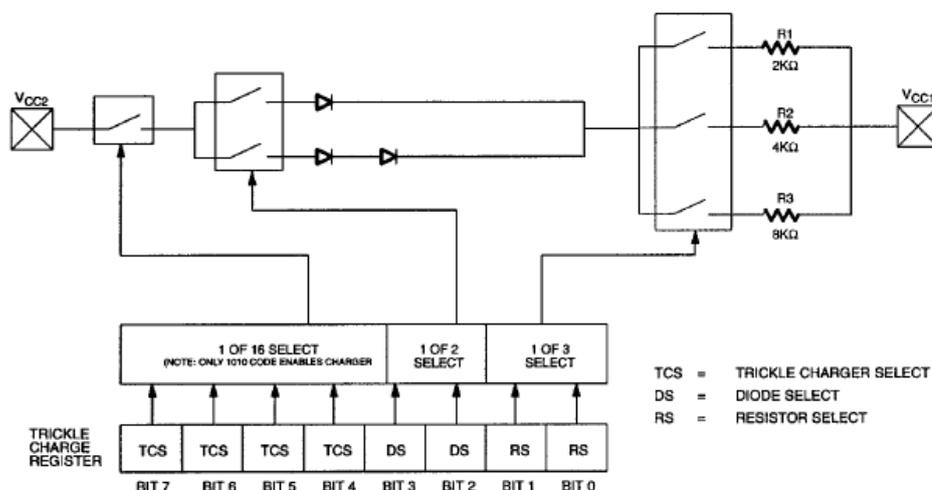
TCS	TCS	TCS	TCS	DS	DS	RS	RS
-----	-----	-----	-----	----	----	----	----

该寄存器控制 SLM1302 的充电特性，由 bit4~7 (TCS) 这四位控制，只有 1010 这种模式才可以使能充电器；

bit2~3 (DS) 用来选择 VCC1 和 VCC2 之间连接一个或者 2 个二极管 (01 表示连接一个二极管、10 表示连接两个二极管，当为 00 或者 11 时，充电器不使能)；

bit1~0 (RS) 用来选择 VCC1 和 VCC2 之间连接的电阻值：00 表示没有电阻，且充电器不使能；01 表示 2K 电阻；10 表示 4K 电阻；11 表示 8K 电阻；

简化的逻辑关系图如下：



# Sola IC

## SLM130Z Trickle Charge Timekeeping Chip

### 5、时钟/日历多字节模式:

CLOCK BURST 寄存器地址:

1	0	1	1	1	1	1	RD — W
---	---	---	---	---	---	---	--------------

### 5、RAM 及 RAM 多字节模式:

寄存器地址:

RAM0	1	0	1	1	1	1	1	RD — W
------	---	---	---	---	---	---	---	--------------

■  
■  
■  
—

RAM30	1	0	1	1	1	1	1	RD — W
-------	---	---	---	---	---	---	---	--------------

寄存器定义:

RAM DATA0								
--------------	--	--	--	--	--	--	--	--

■  
■  
■  
—

RAM DATA30								
---------------	--	--	--	--	--	--	--	--

RAM BUTST 寄存器地址:

1	1	1	1	1	1	1	RD — W
---	---	---	---	---	---	---	--------------

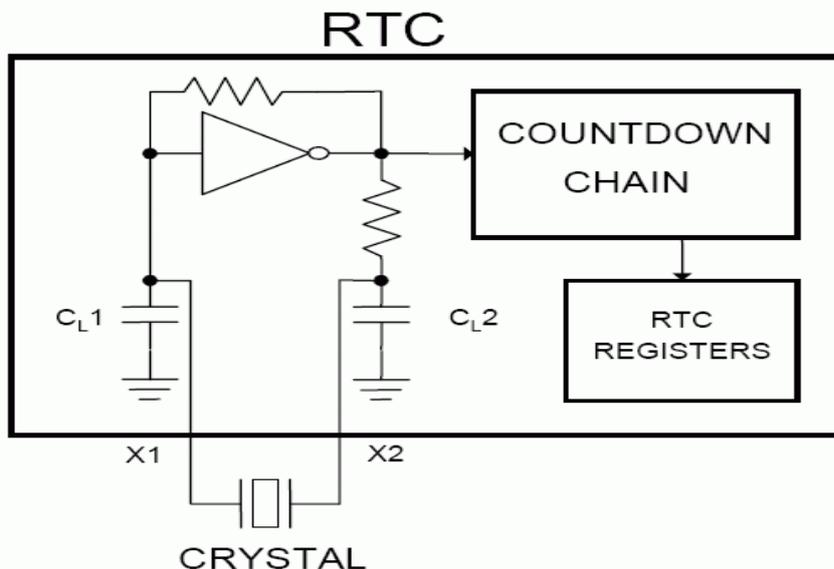
### 6、晶振选择:

# Sola IC

## SLM1302 Trickle Charge Timekeeping Chip

32.768kHz 晶振直接连到 SLM1302 上, 但注意要选择负载为 6pF 的晶振。同时晶振应尽量靠近输入脚, 布线避开可能的干扰。

### 简化图



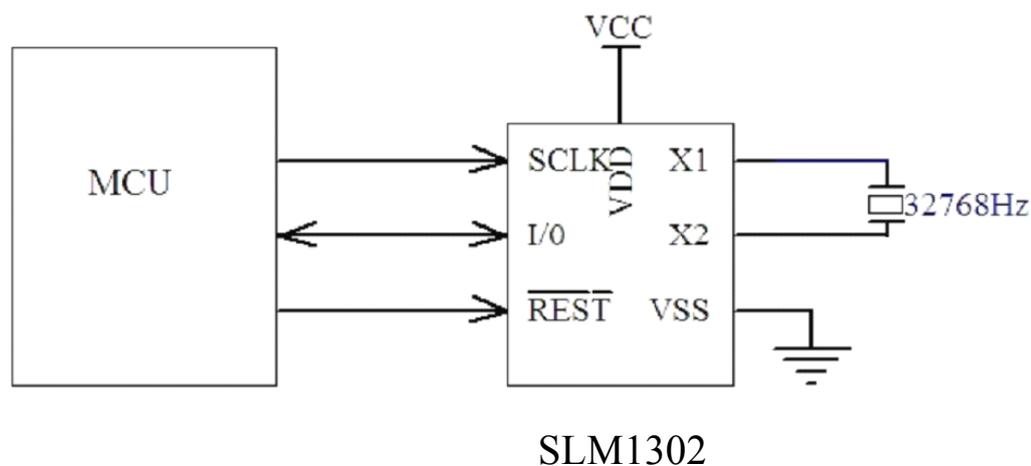
### 7、电源控制:

VCC1 用来提供低功耗模式下的电源;

VCC2 用来提供双供电模式下的电源 (VCC1 也连接作为备用电源);

VCC1 和 VCC2 中大者给 SLM1302 提供电源: 当 VCC2 大于 VCC1+0.2V 时, VCC2 给 SLM1302 供电; 当 VCC2 小于 VCC1 时, VCC1 给 SLM1302 供电。

### 六、典型应用图

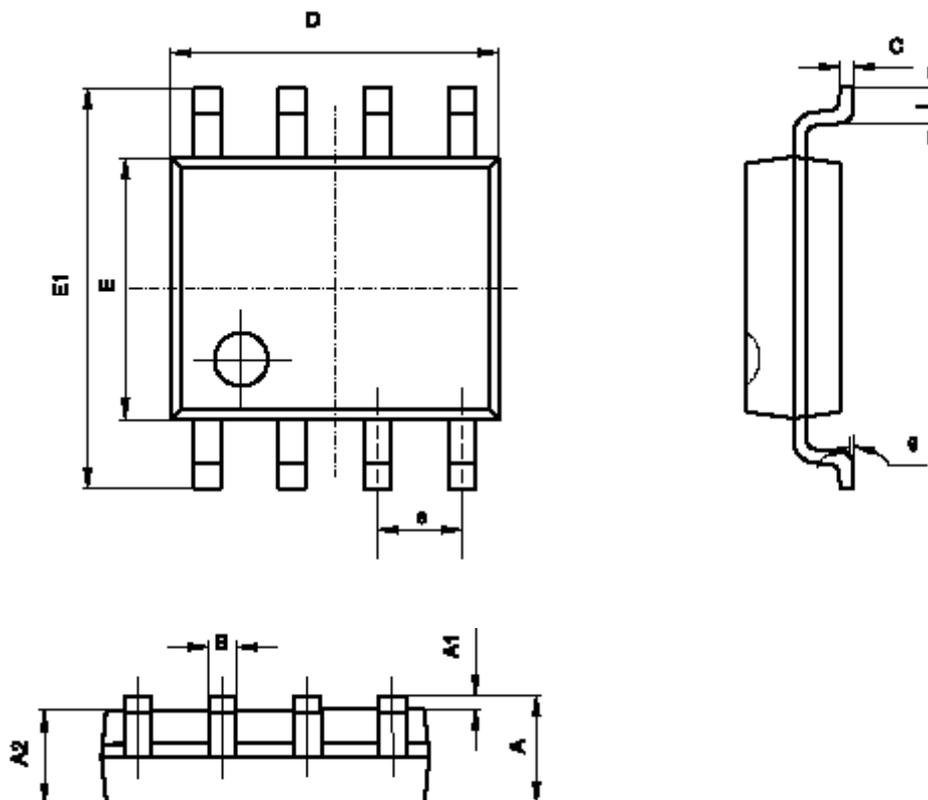


# Sola IC

## SLM1302 Trickle Charge Timekeeping Chip

### 七、外形尺寸图

SOIC8 150MIL



Symbol	Dimensions in Millimeters	
	Min	Max
A	1.350	1.750
A1	0.100	0.250
A2	1.350	1.550
B	0.330	0.510
C	0.180	0.250
D	4.780	6.000
E	3.800	4.000
E1	5.800	6.300
W	1.270(TYP)	
L	0.400	1.270
S	0*	8*