



BYD Microelectronics Co., Ltd.

BF6910AX/11AX

BF6910AX/11AX 规格书

电容触摸按键控制器

产品概述

日期: 2011 年 6 月 3 日



目录

目录.....	2
BF6910AX/11AX.....	4
1、 规格.....	4
1.1 特性.....	4
1.2 应用.....	4
2、 介绍.....	5
3、 引脚.....	6
3.1 引脚图.....	6
3.2 引脚描述.....	6
4、 电气特性.....	8
4.1 AC 特性.....	8
4.2 DC 特性.....	9
4.3 极限参数.....	9
5、 GPIO 端口	9
5.1 PA 端口.....	10
5.2 PB 端口.....	10
5.3 PC 端口	11
6、 寄存器.....	11
6.1 CTK 寄存器描述.....	12
6.2 特殊功能寄存器总表(SFR).....	17
7、 WDT.....	19
8、 复位.....	20
8.1 上电、掉电复位.....	20
8.2 FLASH 编程复位.....	20
8.3 软件复位.....	21
8.4 看门狗定时器溢出复位.....	21
8.5 PC 指针溢出复位.....	21
9、 空闲模式.....	22
10、 定时器.....	22
10.1 定时器 0 和定时器 1.....	22
10.2 定时器时钟控制.....	26
11、 中断.....	26
11.1 中断源及入口地址.....	26
11.2 中断 SFR	27
11.3 中断响应.....	30
11.4 中断优先级	30



11.5 中断采样	30
11.6 中断等待	31
12、PWM 输出模块	32
13、UART0	33
13.1 UART0 中的 SFR	33
13.2 UART0 工作方式	35
13.3 多机通信	36
14、IIC 通信	37
14.1 主机写从机	37
14.2 主机读从机	38
14.3 IIC 中的 SFR	38
14.4 IIC 时序参数	41
14.5 IIC 说明	42
15、应用电路	46
15.1 BF6911AS22(SOP28) IIC 应用	46
16、封装	47
I——QFN32	47
II——SOP16	48
III——SOP20	49
IV——SOP28	50



BF6910AX/11AX

1、规格

1.1 特性

- 按键数目
 - 10/14/22 个按键(Sensor 可以复用为 IO)
- LP(低功率)模式
 - 低功率模式：空闲模式
- 通信模式
 - IIC(支持标准模式或快速模式)
- 可选择的按键模式
 - 每个按键可以运行于独立的模式
- 供电电压：2.7~5.5V
- ROM
 - 8K FLASH
- RAM
 - 512 字节 SRAM
- 自动适应环境变化
- 按键敏感度
 - 自动配置各按键的灵敏度
- 芯片级自动校准逻辑
 - 可调的自动校准速度
- 封装型号： QFN32/SOP16/SOP20/SOP28

1.2 应用

- 白色家电
- 可移动、便携式设备
- 智能电话
- 控制设备
- 游戏控制器
- 遥控器
- 电脑及外围设备

2、介绍

BF6910AX/11AX 是基于 10/14/22 个电容检测通道的芯片，它可以用来检测近距离感应或者触摸。其内置 MCU，可灵活配置；通过配置可实现按键、滚轮、滑条等多种应用。

10/14/22 个按键都能独立的运行，并且每个按键都能通过对相应的特殊功能寄存器写命令来调节灵敏度。

BF6910AX/11AX 通过 IIC 与主机进行通信。

BF6910AX/11AX 包含一个主频为 24MHZ 的单片机内核和一些其他外围设备；如图 1 所示：

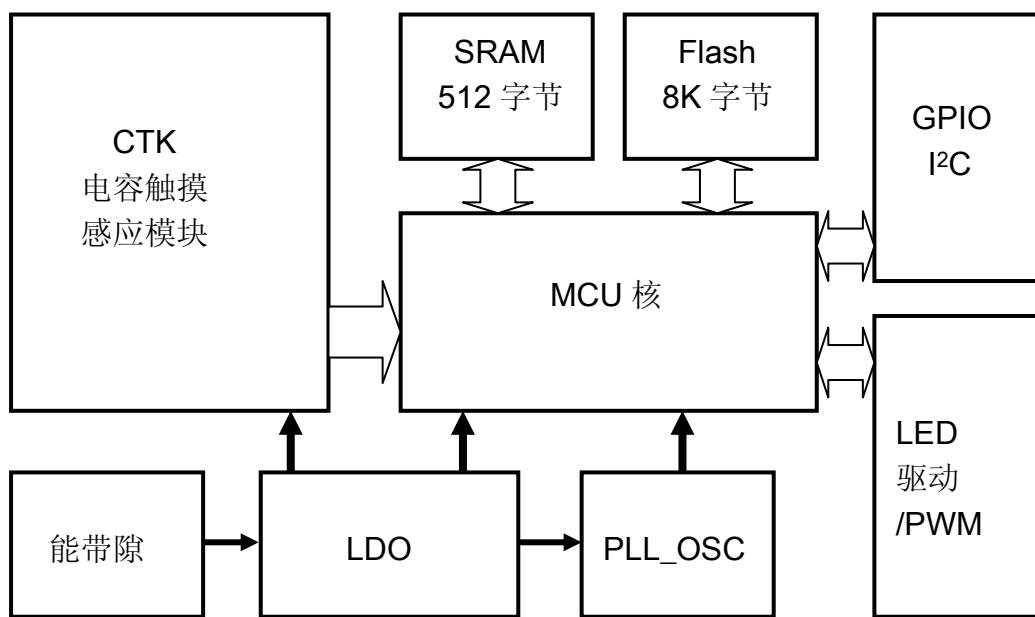
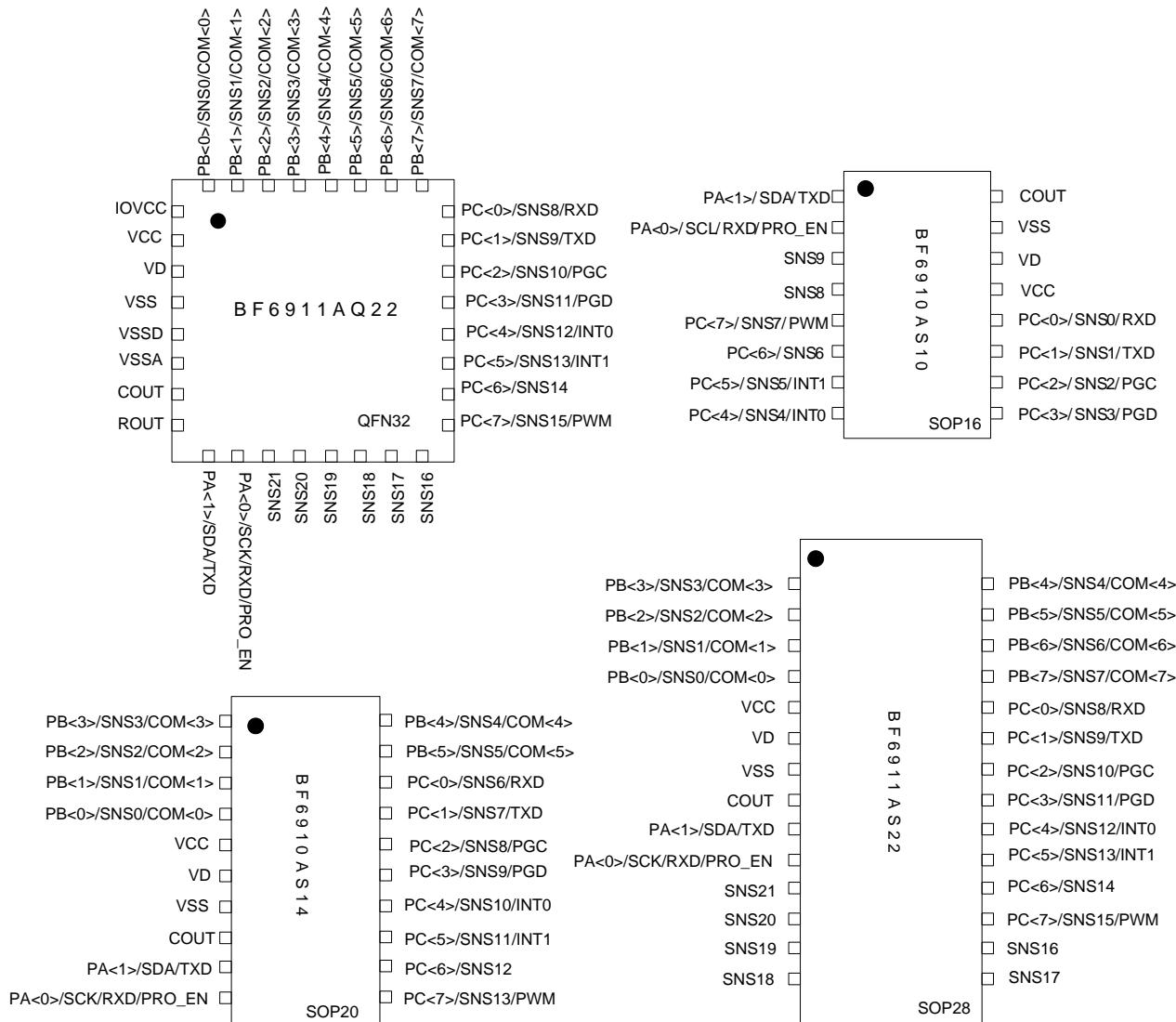


图 1 BF6910AX/11AX 框图

3、引脚

3.1 引脚图



3.2 引脚描述

符号	管脚号				功能描述	未使用
	BF6911 AQ22	BF6910 AS10	BF6910 AS14	BF6911 AS22		
IOVCC	1	-	-	-	供电电压: 2.7~5.5V	-
VCC	2	13	5	5	供电电压: 2.7~5.5V	-



VD	3	14	6	6	内部 LDO 输出。核电压 2.5V	-
VSS	4	-	-	-	地	-
VSSD	5	-	-	-	地	-
VSSA	6	15	7	7	地	-
COUT	7	16	8	8	连接电容	-
ROUT	8	-	-	-	连接电阻	断开
PA<1>/ SDA/ TXD	9	1	9	9	通用 IO 口; IIC 通信数据线; UART 通信发送端口	断开
PA<0>/ SCL/ RXD/ PRO_EN	10	2	10	10	通用 IO 口; IIC 通信时钟线; UART 通信接收端口; 烧录使能端口	断开
PB<0>/ SNS/ COM<0>	32	-	4	4	通用 IO 口; 电容触摸感应通道; LED 驱动时灌入 40mA 电流的 COM 端	断开
PB<1>/ SNS/ COM<1>	31	-	3	3	通用 IO 口; 电容触摸感应通道; LED 驱动时灌入 40mA 电流的 COM 端	断开
PB<2>/ SNS/ COM<2>	30	-	2	2	通用 IO 口; 电容触摸感应通道; LED 驱动时灌入 40mA 电流的 COM 端	断开
PB<3>/ SNS/ COM<3>	29	-	1	1	通用 IO 口; 电容触摸感应通道; LED 驱动时灌入 40mA 电流的 COM 端	断开
PB<4>/ SNS/ COM<4>	28	-	20	28	通用 IO 口; 电容触摸感应通道; LED 驱动时灌入 40mA 电流的 COM 端	断开
PB<5>/ SNS/ COM<5>	27	-	19	27	通用 IO 口; 电容触摸感应通道; LED 驱动时灌入 40mA 电流的 COM 端	断开
PB<6>/ SNS/ COM<6>	26	-	-	26	通用 IO 口; 电容触摸感应通道; LED 驱动时灌入 40mA 电流的 COM 端	断开
PB<7>/ SNS/ COM<7>	25	-	-	25	通用 IO 口; 电容触摸感应通道; LED 驱动时灌入 40mA 电流的 COM 端	断开
PC<0>/ SNS/ RXD	24	12	18	24	通用 IO 口; 电容触摸感应通道; UART 通信接收端口	断开



PC<1>/ SNS/ TXD	23	11	17	23	通用 IO 口; 电容触摸感应通道; UART 通信发送端口	断开
PC<2>/ SNS/ PGC	22	10	16	22	通用 IO 口; 电容触摸感应通道; 烧录时钟端口	断开
PC<3>/ SNS/ PGD	21	9	15	21	通用 IO 口; 电容触摸感应通道; 烧录数据端口	断开
PC<4>/ SNS/ INT0	20	8	14	20	通用 IO 口; 电容触摸感应通道; 外部中断 0	断开
PC<5>/ SNS/ INT1	19	7	13	19	通用 IO 口; 电容触摸感应通道; 外部中断 1	断开
PC<6>/ SNS	18	6	12	18	通用 IO 口; 电容触摸感应通道	断开
PC<7>/ SNS/ PWM	17	5	11	17	通用 IO 口; 电容触摸感应通道; PWM 输出	断开
SNS	16	4	-	11	电容触摸感应通道	断开
SNS	15	3	-	12	电容触摸感应通道	断开
SNS	14	3	-	13	电容触摸感应通道	断开
SNS	13	3	-	14	电容触摸感应通道	断开
SNS	12	3	-	15	电容触摸感应通道	断开
SNS	11	3	-	16	电容触摸感应通道	断开

注：“-”表示工作时必须使用的管脚

4、电气特性

4.1 AC 特性

下表列出了 OSC 时钟特性参数：

参数	符号	条件	OSC时钟	单位
基频	OSC	VCC=2.25~2.75V	1MHz±0.3%	MHz
		正常工作	1MHz±0.3%	
		环境温度-40℃~85℃	1MHz±2.5%	



4.2 DC 特性

下表列出了在电压范围: 2.7V~5.5V, 温度范围: -40°C < TA < 85°C 下各参数的最大值与最小值。经验值为在27°C、3.3V条件下的测量值。

参数	符号	条件	最小值	典型值	最大值	单位
工作电压	VCC		2.7		5.5	V
电流	I _{active}	激活模式, 系统时钟12M		3.5		mA
		激活模式, 系统时钟24M		4.5		mA
	I _{idle}	空闲模式		30	50	μA
输入低电压	V _{IL}	VCC=3.3~5.5V	-	-	0.2*VCC	V
输入高电压	V _{IH}	VCC=3.3~5.5V	0.8*VCC	-	-	V
输出低电压	V _{OL}	I _{OL} =4mA@VCC=3.3V I _{OL} =10mA@VCC=5V	-	-	0.1*VCC	V
输出高电压	V _{OH}	I _{OL} =4mA@VCC=3.3V I _{OL} =10mA@VCC=5V	0.9VCC	-	-	V
输入漏电流	I _{IH}	-	-	1	5	μA
	I _{IL}					

4.3 极限参数

参数	符号	最小值	典型值	最大值	单位
储存温度	T _{stg}	-55	-	125	°C
运行温度	T _{otg}	-40	-	85	°C
输入电压	V _{in}	V _{SS} -0.5	-	V _{SS} +0.5	V
锁存电流	I _U	100	-		mA
静电放电电压	ESD(HBM)	2000	-		V
供电电压	VCC	-0.5	-	5.5	V

5、GPIO 端口

可供使用的 GPIO 口, 与所选用的芯片型号和使用的触摸通道数有关。GPIO 端口的一些引脚和器件外设功能复用。通常情况下, 当使能外设时, 该引脚不能做为通用 IO 引脚使用。PORT 有 3 个寄存器:

TRIS 寄存器 (方向寄存器)

DATA 寄存器 (数据寄存器)

SEL 寄存器 (复用功能选择寄存器)



5.1 PA 端口

PA 端口是 2 位宽的双向端口。PA 口数据寄存器为 DATAA, SFR 地址 0xBC, 不能位操作。相应的数据方向寄存器为 TRISA。TRISA 置 1 可以将对应的 PA 引脚配置为输入引脚, 将 TRISA 位清零可以将对应的 PA 引脚配置为输出引脚。DATAA 是 PA 口对应的数据寄存器。PA 的引脚复用情况:

DATAA 寄存器

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DATAA	-	-	-	-	-	-	PA<1>/SDA	PA<0>/SCL

Bit7-2: 保留

Bit1: PA<1>: 通用 IO 口; SDA: IIC 数据线

Bit0: PA<0>: 通用 IO 口; SCL: IIC 时钟线

在使用相应外设模块功能时根据需要设置正确的端口方向, 端口方向寄存器 TRISA:

TRISA 寄存器

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TRISA	-	-	-	-	-	-	TRISA<1:0>	

5.2 PB 端口

PB 端口是 8 位有效的双向端口。PB 口数据寄存器为 DATAB, SFR 地址 0x90, 可位操作。TRISB 是对应的方向控制寄存器, 可位操作。设置为 1, 对应的端口输入, 设置为 0, 对应端口输出。PB_SEL 为对应的复用功能选择寄存器, 为 0 则为触摸通道, 1 为通用 IO 口。

DATAB 寄存器:

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DATAB	PB<7>	PB<6>	PB<5>	PB<4>	PB<3>	PB<2>	PB<1>	PB<0>

TRISB 寄存器:



SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TRISB	TRISB<7:0>							

PB_SEL 寄存器:

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PB_SEL	PB_SEL<7:0>							

5.3 PC 端口

PC 端口是 8 位有效的双向端口。PC 口数据寄存器为 DATA, SFR 地址 0x80, 可位操作。TRISC 是对应的方向控制寄存器, 可位操作。设置为 1, 对应的端口输入, 设置为 0, 对应端口输出。PC_SEL 为对应的复用功能选择寄存器, 0 选择作为触摸通道, 1 为通用 IO 口。

DATA 寄存器:

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DATA	PC<7>	PC<6>	PC<5>	PC<4>	PC<3>	PC<2>	PC<1>	PC<0>

TRISC 寄存器:

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TRISC	TRISC<7:0>							

PC_SEL 寄存器:

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PC_SEL	PC_SEL<7:0>							

6、寄存器

BF6910AX/11AX通过一系列的寄存器来实现多种功能的应用。主机通过IIC通信对CTK 模块中的寄存器进行读写操作。



6. 1 CTK 寄存器描述

自适应电流源寄存器(CTK_MIR)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CTK_MIR	CTK_MIR<7:0>							

配置通道的自适应电流源。当CTK_MIR为0xFF时，自适应电流源断开。

默认值：0xFF。

自适应电流源步进寄存器(CTK_IREF)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CTK_IREF	CTK_IREF<2:0>							

配置通道的自适应电流源步进值。

默认值：0x07。

CTK_IREF<2:0>与自适应电流源步进关系如下表：

CTK_IREF<2:0>	自适应电流源步进
011	1 uA
100	0.75 uA
101	0.5 uA
110	0.25uA
111	断开电流源
...	保留

放电电阻选择寄存器 (RB_SW)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RB_SW	-							

选择 CTK 模块放电电阻为内置或外接电阻。1 为内置电阻，0 为 ROUT 脚外接电阻 2K-20K。

默认值：0x01。

数据精度寄存器(RESO)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-----	------	------	------	------	------	------	------	------



RESO	-	RESO<2:0>
------	---	-----------

RESO表示按键数据和参考数据的分辨率。当RESO值较大，检测的灵敏度提高，但是采样速度下降。

默认值：0x03

RESO<2:0>	Content
000	7Bits
001	8Bits
010	9Bits
011	10Bits
100	11 Bits
101	12 Bits
110	13 Bits
111	14 Bits

参考电压切换寄存器 (VTH_SW)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
VTH_SW	-								VTH_SW<0>

在电容检测中必须清零。

默认值：0x00。

参考电压寄存器(CTK_RV)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
CTK_RV	-								CTK_RV<2:0>

该寄存器用来设置参考电压值，参考电压值增大，灵敏度降低。

默认值：0x00。

CTK_RV	参考电压
000	0.625V
001	0.781V
010	0.937V
011	1.091V
100	1.25V



检测速度寄存器(CTK_DS)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CTK_DS	-							CTK_DS<2:0>

CTK_DS可以提高按键的响应时间。CTK_DS值越大，按键响应越慢。为了保持采样波形不失真，建议 $f_{CTK_DS} \geq 2f_{PRS_DIV}$ 。

默认值：0x00。

CTK检测速度与CTK_DS的关系如下表：

CTK_DS<2:0>	CTK检测速度
000	12M
100	8M
001	6M
101	4M
010	3M
110	2M
111	1M

通道地址寄存器(CTK_ADDR)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CTK_ADDR	-							CTK_ADDR<4:0>

CTK_ADDR 的低五位表示当前检测 SNS0-SNS21 中的某个通道。如 CTK_ADDR=0x04 时，表示当前检测为 SNS4。

默认值：0x00。

瞬时计数值寄存器(RAWDATA、RAWDATAH)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RAWDATA	RAWDATA<7:0>							

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RAWDATAH	-							RAWDATA<13:8>

该寄存器保存当前检测通道的瞬时计数值。

默认值：0x00



CTK 检测模式寄存器 (DMS)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DMS	-							DMS<1:0>

设置 CTK 检测模式。

默认值: 0x00。

DMS<1:0>	CTK检测模式
11	全同步模式
10	相邻同步模式
01/00	非同步模式

CTK 充放电频率寄存器 (PRS_DIV)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PRS_DIV	-							PRS_DIV<3:0>

设置CTK充放电频率。PRS_DIV可以提高SNR。当处于嘈杂的环境时，PRS_DIV值越大，SNR越大。为了保持采样波形不失真，建议 $f_{CTK_DS} \geq 2f_{PRS_DIV}$ 。

默认值: 0x03。

PRS_DIV<3:0>	CTK充放电频率	时钟类型
0000	1M	固定时钟
0001	2M	
0010	3M	
0011	4M	
0100	6M	
1000	最高频率4M, 最低频率1M, 中心频率1.6M	PRS时钟
1001	最高频率4M, 最低频率1M, 中心频率3.5M	

按键使能寄存器(CTK_CEN1、CTK_CEN2、CTK_CEN3)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CTK_CEN1	SNS7	SNS6	SNS5	SNS4	SNS3	SNS2	SNS1	SNS0
CTK_CEN2	SNS15	SNS14	SNS13	SNS12	SNS11	SNS10	SNS9	SNS8



CTK_CEN3	-	SNS21	SNS20	SNS19	SNS18	SNS17	SNS16
----------	---	-------	-------	-------	-------	-------	-------

用来控制按键的开启与关断。当为1时，允许检测相应的按键；当为0时，禁止检测相应的按键。虽然当按键悬空时不会检测到触摸信息，但是将相应的按键使能位清0能提高采样速率。

默认值：0x00

CTK 模块使能寄存器(CTK_CTRL)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CTK_CTRL	-						CTK_EN	CTK_WR_EN

CTK_CTRL控制CTK模块的使能以及对CTK模块写使能。在CTK检测过程中，CTK_WR_EN必须保持为高。将CTK_EN拉高，则开始扫描配置的通道，在SNS通道扫描时，CTK_EN必须一直持续为高，当前通道扫描完成进入中断后，将CTK_EN拉低。

默认值：0x00

CTK 供电寄存器 (PD_CTK)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PD_CTK	-						PD_CTK<0>	

CTK模块供电使能寄存器。PD_CTK=0，CTK模块正常工作；PD_CTK=1，CTK模块不工作。

默认值：0x01

ADC 供电寄存器 (PD_ADC)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PD_ADC	-	保留					PD_ADC<0>	

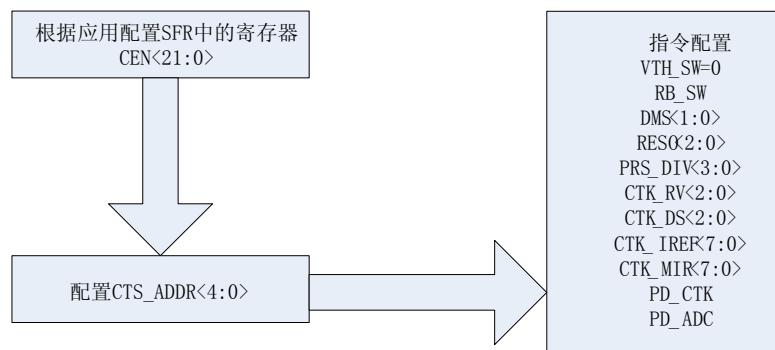
在电容触摸检测时，该寄存器为0x01。

默认值：0x01

由于各芯片所用到的通道数目不同，从而引出的通道也不同，各芯片型号内外部通道对应关系如下表：

外部通道		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
内 部 SNS	BF6911AQ22(QFN32)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	BF6910AS10(SOP16)	8	9	10	11	12	13	14	15	20	21	-	-	-	-	-	-	-	-	-	-	-	
	BF6910AS14(SOP20)	0	1	2	3	4	5	8	9	10	11	12	13	14	15	-	-	-	-	-	-	-	
	BF6911AS22(SOP28)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

下图详细描述了CTK模块参数配置流程：



6.2 特殊功能寄存器总表(SFR)

表1 SFR 总结

名称	默认值	地址	功能描述
DATA0	1111_1111	80h	PC 口数据寄存器
SP	0000_0111	81h	堆栈指针寄存器
DPL0	0000_0000	82h	数据指针寄存器 0 低 8 位
DPH0	0000_0000	83h	数据指针寄存器 0 高 8 位
DPL1	0000_0000	84h	数据指针寄存器 1 低 8 位
DPH1	0000_0000	85h	数据指针寄存器 1 高 8 位
DPS	0000_0000	86h	数据指针选择寄存器
PCON	0011_0000	87h	低功耗模式选择寄存器
TCON	0000_0000	88h	定时器控制寄存器
TMOD	0000_0000	89h	定时器模式寄存器
TL0	0000_0000	8Ah	定时器 0 计数器低 8 位
TH0	0000_0000	8Ch	定时器 0 计数器高 8 位
TL1	0000_0000	8Bh	定时器 1 计数器低 8 位
TH1	0000_0000	8Dh	定时器 1 计数器高 8 位
CKCON	0000_0001	8Eh	时钟控制寄存器
SPC_FNC	0000_0000	8Fh	DM/PM 选择寄存器
DATA1	1111_1111	90h	PB 口数据寄存器
EXIF	0000_1000	91h	外部中断标志寄存器
MPAGE	0000_0000	92h	P2 口锁存寄存器
WDT_CTRL	0000_0111	93h	WDT 时间寄存器
SCON	0000_0000	98h	串口 0 控制寄存器
SBUF	0000_0000	99h	串口 0 数据缓冲寄存器
IICSTAT	0000_0000	A0h	IIC 通信控制寄存器
IICADD	0000_0000	A1h	IIC 地址寄存器
IICBUF	xxxx_xxxx	A2h	IIC 数据收发缓冲寄存器



PWM_DIV	0000_0001	A3h	PWM 周期寄存器
CEN_ADC	0000_0000	A4h	AD/触摸通道选择寄存器
PWM_DUTY	0000_0000	A5h	PWM 脉宽寄存器
ADC_EN	0000_0000	A6h	AD/触摸通道检测指示寄存器
CTK_MIR	1111_1111	A7h	自适应电流源寄存器
IE	0000_0000	A8h	中断使能寄存器
CTK_RV	0000_0000	A9h	参考电压寄存器
CTK_DS	0000_0001	AAh	检测速度寄存器
CTK_ADDR	0000_0000	ABh	检测通道地址寄存器
RAWDATA1	0000_0000	ACh	当前检测通道采样值低 8 位
RAWDATAH	0000_0000	ADh	当前检测通道采样值高 8 位
DMS	0000_0000	AFh	通道检测模式寄存器
TRISB	1111_1111	B0h	PB 口方向寄存器
PB_SEL	1111_1111	B1h	PB 口 IO 口/触摸通道选择寄存器
PC_SEL	1111_1111	B2h	PC 口 IO 口/触摸通道选择寄存器
RST_STAT	xxxx_xxxx	B3h	复位标志状态寄存器
PRS_DIV	xxxx_0011	B4h	PRS 时钟分频寄存器
CTK_CEN1	0000_0000	B5h	通道 7-0 使能寄存器
CTK_CEN2	0000_0000	B6h	通道 15-8 使能寄存器
CTK_CEN3	0000_0000	B7h	通道 21-16 使能寄存器
IP	1000_0000	B8h	中断优先级寄存器
CTK_CTRL	xxxx_xx00	B9h	CTK 使能寄存器
RESO	xxxx_x011	BAh	扫描分辨率寄存器
CTK_IREF	xxxx_x111	BBh	自适应电流源步进寄存器
DATAA	xxxx_xx11	BCh	PA 口数据寄存器
TRISA	xxxx_xx11	BDh	PA 口方向寄存器
IICCON	0000_0000	C0h	IIC 控制寄存器
PD_ADC	0000_0001	C1h	AD 供电使能寄存器
RB_SW	0000_0001	C2h	RB 内置/外接选择寄存器
VTH_SW	0000_0000	C3h	AD 检测参考电压选择寄存器
IO_CON	0000_0000	C4h	外部中断/两路 UART 选择/PWM 输出使能寄存器
PD_CTK	0000_0001	C5h	CTK 供电使能寄存器
SOFT_RST	0000_0000	C6h	软件复位寄存器
PWM_CS	0000_0000	C7h	PWM 供电使能寄存器
TRISC	1111_1111	C8h	PC 口方向寄存器
PSW	0000_0000	D0h	程序状态字寄存器
CFG0	1111_1111	D1h	配置字 0
CFG1	1111_1111	D2h	配置字 1



CFG2	1111_1111	D3h	配置字 2
CFG3	1111_1111	D4h	配置字 3
CFG4	1111_1111	D5h	配置字 4
LED_EN	0000_0000	D6h	LED 灌电流驱动使能寄存器
EICON	0100_0000	D8h	扩展中断控制寄存器
ACC	0000_0000	E0h	累加器
EIE	1110_0000	E8h	外部中断使能寄存器
B	0000_0000	F0h	B 寄存器
EIP	1110_0000	F8h	外部中断优先级寄存器

注： x 表示为不定态。

7、WDT

BF6910AX/11AX 内部有一个使用系统时钟的可编程看门狗定时器（WDT）。当看门狗定时器溢出时，WDT 将强制 CPU 进入复位状态。为了防止复位，必须在溢出发生前由应用软件重新触发 WDT。如果系统出现了软件/硬件错误，使应用软件不能重新触发 WDT，则 WDT 将溢出并产生一个复位，这可以防止系统失控。看门狗的功能由看门狗定时器控制寄存器（WDT_CTRL）控制。

看门狗定时器通过编程可以实现从 18ms 到 2.3s 的定时。定时长度由 SFR(WDT_CTRL) 控制，如下表所示：

WDT_CTRL<2:0>	间隔
000	18ms
001	36ms
010	72ms
011	144ms
100	288ms
101	576ms
110	1152ms
111	2304ms

看门狗定时器在复位结束后一直工作，MCU 处于 idle mode 时看门狗溢出输出中断信号（INT_WDT）唤醒 MCU；其他情况下看门狗溢出输出复位信号（WDT_RST）使系统复位 1ms。

看门狗定时器清零是通过写 WDT_CTRL 寄存器完成的，无论向此寄存器中写入何值都会使看门狗定时器清零。

8、复位

BF6910AX/11AX 中有五种复位模式：上电/掉电复位（POR_BOR_N）、flash 编程复位（PRO_EN）、软件复位（SOFT_RST）、看门狗定时器溢出复位（WDT_RST）、PC 指针溢出复位（ADDR_OVERFLOW）。只要其中任意一种复位发生，系统的全局复位信号就会让整个芯片复位。可由复位标志寄存器来确定芯片进行了何种复位，复位标志位需软件进行清零。

RST_STAT 寄存器

SFR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
RST_STAT	--	--	--	--	ADDROF_F	BO_F	PO_F	WDTRST_F

WDTRST_F：看门狗复位的标志位，在看门狗复位发生时，此位的值是 1。

BO_F：掉电复位的标志位，在掉电复位发生时，此位的值是 1。

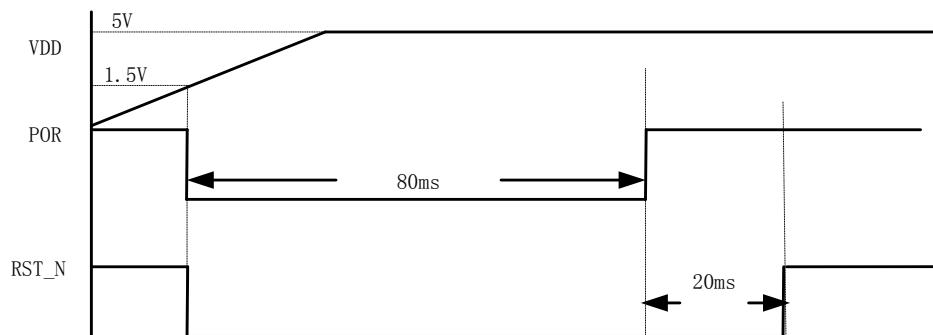
PO_F：上电复位的标志位，在上电复位发生时，此位的值是 1。

ADDROF_F：地址溢出复位标志位，在地址溢出复位发生时，此位的值是 1。

8.1 上电、掉电复位

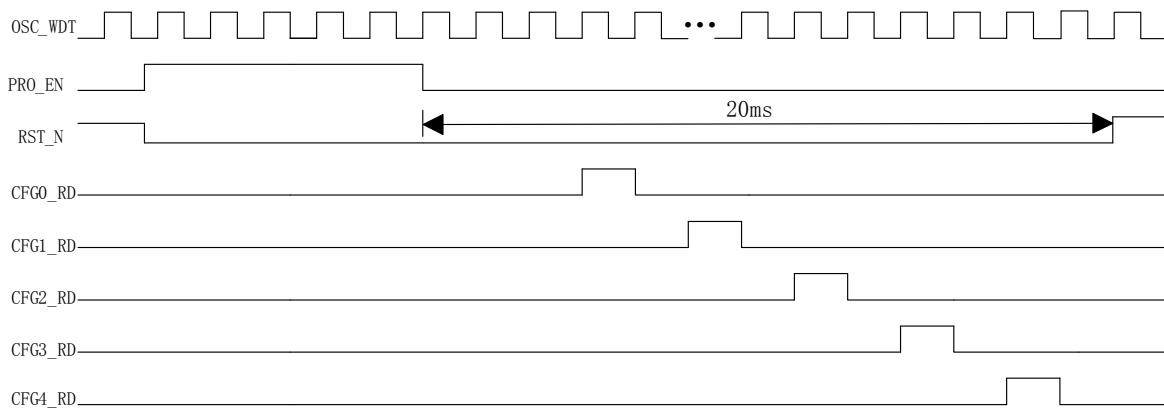
系统发生上电、掉电复位后，系统产生 POR_BOR_N 信号（低电平）并持续 80ms。

POR_BOR_N 为低时整个芯片处于复位状态，变高后全局复位信号继续有效 20ms，这 20ms 内给出 flash 的五个读配置字的使能信号。



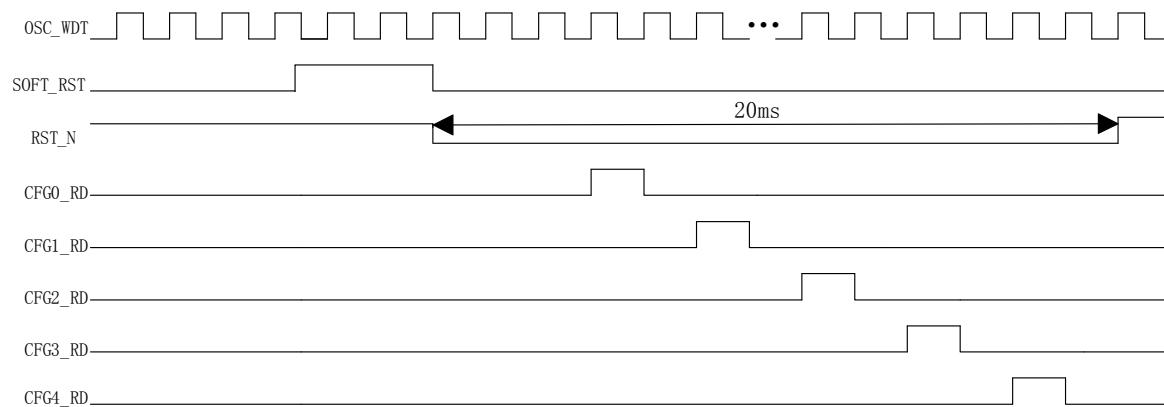
8.2 FLASH 编程复位

PRO_EN 为高时为 flash 的编程模式，此时全局复位信号有效，变低后全局复位信号继续有效 20ms，这 20ms 内给出 flash 的五个读配置字的使能信号。



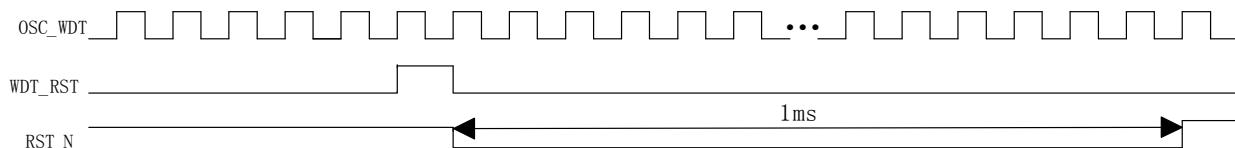
8.3 软件复位

通过向 SOFT_RST SFR 中写入 0x01 使软复位信号 SOFT_RST 有效，使全局复位信号有效 20ms，这 20ms 内给出 flash 的五个读配置字的使能信号。



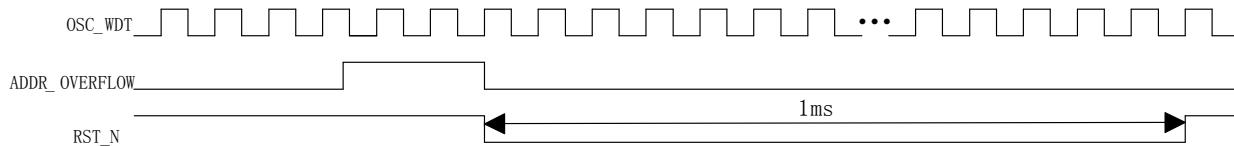
8.4 看门狗定时器溢出复位

看门狗定时器溢出后 **WDT_RST** 信号变高，使全局复位 1ms。



8.5 PC 指针溢出复位

若 MCU 寻址程序存储器时 PC 指针超出了 flash 有效的地址范围，**ADDR_OVERFLOW** 信号变高，使全局复位 1ms。



9、空闲模式

空闲模式：BF6910AX/11AX 向特殊功能寄存器 PCON 的最低位写入 1 使 MCU 进入空闲模式（idle mode），MCU 检测到有中断发生时会硬件清零此标志位，退出空闲模式。系统复位也可将此寄存器清零，退出空闲模式。

在空闲模式下，BF6910AX/11AX 可以由三种中断进行唤醒：INT_WDT（看门狗中断）、INT_EXT（外部电平变化中断）、INT_IIC（IIC 中断）。在进入空闲模式之前，将 OSC 时钟关闭，所以 CTK 中断不能唤醒 MCU。当中断到来时，经过 1ms 的定时，使得时钟和锁相环稳定工作后，再将中断送入内核以唤醒芯片。

由于在外部中断发生后要进行 1ms 定时来稳定时钟和锁相环，所以在进入空闲模式之前建议将定时器以及串口中断关闭，在唤醒后再开启相应的中断。

10、定时器

BF6910AX/11AX 包含两个定时器（定时器 0 和定时器 1）。定时器的时钟为系统时钟的分频时钟。每个定时器包含一个 16 位的寄存器。在被访问时以两个字节的形式出现：一个低字节（TL0 或 TL1）和一个高字节（TH0 或 TH1）。通过设置 IE 寄存器中的 ET0 位使能定时器 0 中断，通过设置 IE 寄存器中的 ET1 位使能定时器 1 中断。

- 定时器 0 – TL0 和 TH0
- 定时器 1 – TL1 和 TH1

10.1 定时器 0 和定时器 1

定时器 0/1 有四种运行模式，由 TMOD SFR 和 TCON SFR 控制。

定时器 0/1 四种模式如下：

- 13 位定时器/计数器（模式 0）
- 16 位定时器/计数器（模式 1）
- 自动重载初值的 8 位计数器（模式 2）
- 两个 8 位计数器（模式 3，只用于定时器/计数器 0）

TMOD 寄存器 – SFR 89h

位	功能
---	----



TMOD.7	GATE-定时器1门控位。当GATE=0,TR1=1时，开启定时器1。
TMOD.6	C/T-选择定时器/计数器。当C/T=0时，定时器/计数器1作为定时器，时钟是clk/4或者clk/12，由T1M(CKCON.4)的状态决定。C/T=1未定义，保留。
TMOD.5-4	M1-定时器1模式选择Bit 1, M0-定时器1模式选择Bit 0, M1M0: 00=模式 0 – 13位定时器/计数器 01=模式 1 – 16位定时器/计数器 10=模式 2 – 自动重载初值的8位计数器 11=模式 3 – 两个8位计数器
TMOD.3	GATE-定时器 0门控位。当GATE=1,INT_EXT=1, TR0(TCON.4)=1时，开启定时器0；当GATE=0,TR0=1时，开启定时器0。
TMOD.2	C/T-选择计数器/定时器。当C/T=0时，定时器 /计数器0是定时器，时钟是clk/4或者clk/12，由T0M(CKCON.3)的状态决定。C/T=1未定义，保留。
TMOD.1-0	M1-定时器0模式选择Bit 1, M0-定时器0模式选择Bit 0 M1M0 00=模式 0 – 13位定时器/计数器 01=模式 1 – 16位定时器/计数器 10=模式 2 – 自动重载初值的8位计数器 11=模式 3 – 两个8位计数器

TCON 寄存器-SFR 88h

位	功能
TCON.7	TF1-定时器1溢出中断标志位。当定时器1计数溢出时该位置1，当CPU进入中断服务程序时，该位由硬件自动清0。
TCON.6	TR1-定时器1使能位。当置1时开启定时器 1。
TCON.5	TF0-定时器 0溢出中断标志位。当定时器 0计数溢出时该位置1，当CPU进入中断服务程序时，该位由硬件自动清0。
TCON.4	TR0- 定时器 0使能位。当置1时开启定时器 0。
TCON.3	保留
TCON.2	保留
TCON.1	IE0- IT0=1时，INT0设定为边缘触发，当INT_EXT检测到下降沿时，IE0被硬件置1，当CPU进入外部中断服务程序时，IE0清0。在边缘触发模式中，IE0也能被软件清0。 IT0=0时，INT0设定为电平触发，当INT_EXT是低电平时，IE0置1，当INT_EXT是高电平时，IE0清0。在电平触发模式中，软件不能写IE0。

TCON.0	IT0-INT0 模式选择。当 IT0=1 时，BF6910AX/11AX 在下降沿检测 INT_EXT。 当 IT0=0 时，BF6910AX/11AX 在低电平处检测 INT_EXT。
--------	---

模式 0 :13 位定时器/计数器

在模式 0 下定时器 0 和定时器 1 的工作过程相同，如图 5 所示。在模式 0 中，定时器为 13 位的计数器，其 0-4 位为 TL0(或者 TL1)，另外 8 位为 TH0(或者 TH1)。TCON 寄存器中的使能位(TR0/TR1)来控制定时器的开启和关闭。C/T 位用来选择定时器/计数器的时钟源。

当 GATE=0 时，定时器对选定的时钟源进行计数；当 GATE=1 时，定时器 0 是否计数取决于 INT_EXT 引脚的信号，当 NT_EXT 由 0 变 1 时，开始计数；当 INT_EXT 由 1 变 0 时，停止计数。

当 13 位计数器计数累积到 1FFFh(全 1 时)，计数器清 0(全 0)，并且 TF0(或者 TF1)置位。

在模式 0 中，TL0(或者 TL1)的高 3 位是不确定的，在读计数值时应屏蔽掉或忽略这 3 位。

模式 1 :16 位定时器/计数器

定时器 0 和定时器 1 的模式 1 是相同的，如图 3 所示。在模式 1 中，定时器为 16 位的计数器。LSB 寄存器(TL0 或者 TL1)的所有 8 位都被使用。当计数器计数累计至 FFFFh 时，计数器清为全 0。除此之外，模式 1 和模式 0 是相同的。

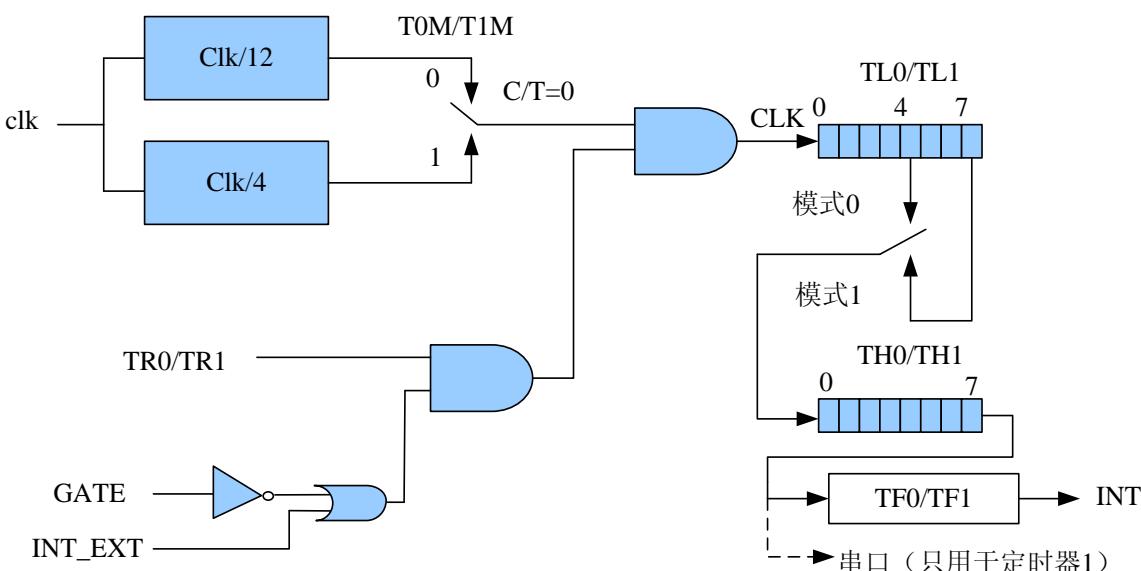


图3 定时器 0/1 -模式0和1

模式 2:自动重载初值的 8 位计数器

定时器 0 和定时器 1 的模式 2 是相同的。在模式 2 中，定时器为一个带有自动重载初值的 8 位计数器。这个计数器就是 LSB 寄存器(TL0 或者 TL1)，需要重载的初值保存在 MSB 寄存器(TH0 或者 TH1)中。

如图 4 所示，模式 2 的计数器控制和模式 0、模式 1 是一样的。但是，在模式 2 中，当 TLn 累计至 FFh，保存在 THn 中的值重载至 TLn。

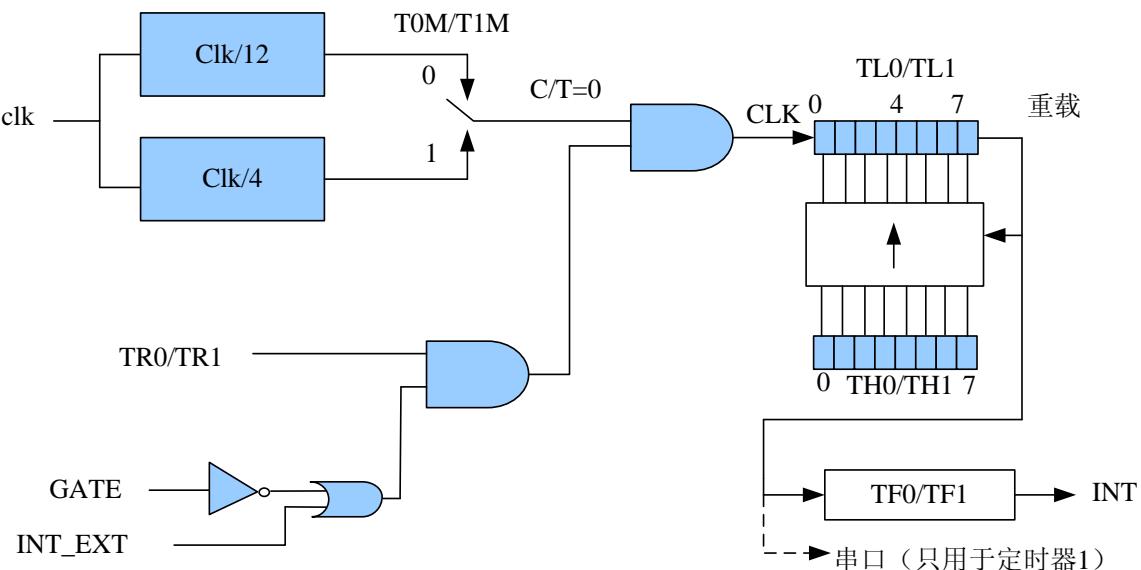


图4 定时器 0/1 -模式2

模式3：两个8位计数器

在模式3中，定时器0为两个8位的计数器，此时定时器1停止计数并且保存它的值。如图5所示，TL0是由定时器0的控制位来控制的8位寄存器。TL0能够用来计算时钟周期(4分频或者12分频)，由C/T位决定。计数器用GATE作为使能端来控制INT_EXT信号接收。

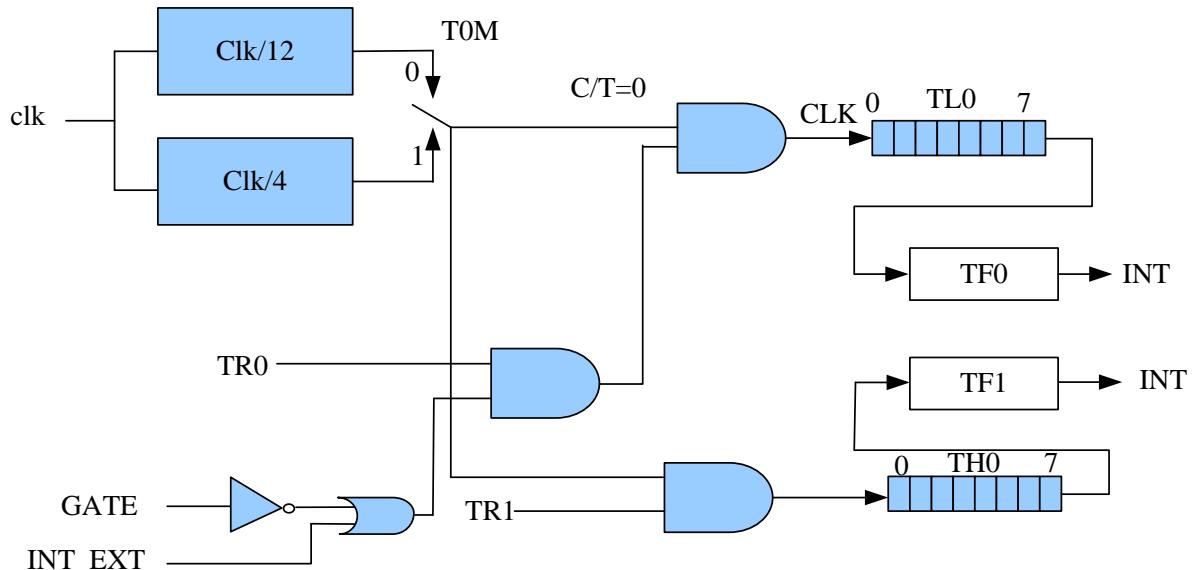


图 5 定时器 0-模式 3

TH0 是一个单独的 8 位计数器。TH0 只能用来计算时钟周期(4 分频或者 12 分频)。定时器 1 的控制位和标志位(TR1 和 TF1)用来作为 TH0 的控制位和标志位。

当定时器 0 工作在模式 3 时，定时器 1 的使用受到限制，因为定时器 0 用到了定时器 1 的控制位(TR1)和中断标志位(TF1)。定时器 1 仍然能用来产生波特率，并且定时器 1 在 TL1 和 TH1 寄存器中的值依然有效。

当定时器 0 工作在模式 3 时，通过定时器 1 的模式位来控制定时器 1。要开启定时器 1，



需要将定时器 1 设置为模式 0、1 或者 2。要关闭定时器 1，将定时器 1 的模式设置为 3。定时器 1 的 C/T 位和 T1M 位依旧对定时器 1 有效。因此，定时器 1 可以作为定时器(时钟为 `clk/12` 或 `clk/4`)，但是由于 TR1 和 TF1 被借用，不能产生溢出中断，所以只能作串口的波特率发生器。当定时器 0 工作在模式 3 时，定时器 1 的 **GATE** 有效。

10.2 定时器时钟控制

在 BF6910AX/11AX 中，默认的定时器时钟是 `clk/12`，默认的指令周期是 `clk/4`。

使用默认的时钟时，能够对现有的应用代码进行实时的完整的处理，例如波特率。但是，当需要快速计数时，应用程序能够通过 **CKCON**(时钟控制寄存器)中的控制位来设置计数器的时钟为 `clk/4`。CKCON 寄存器在 SFR 中的地址为 `8Eh`。

控制定时器时钟比率的 CKCON 寄存器描述如下：

CKCON 寄存器—SFR 8Eh

位	功能
CKCON.7-5	保留
CKCON.4	T1M-定时器 1 时钟选择。当 <code>T1M=0</code> 时，定时器 1 时钟为 <code>clk/12</code> ；当 <code>T1M=1</code> 时，定时器 1 时钟为 <code>clk/4</code> 。
CKCON.3	T0M-定时器 0 时钟选择。当 <code>T0M=0</code> 时，定时器 0 时钟为 <code>clk/12</code> ；当 <code>T0M=1</code> 时，定时器 0 时钟为 <code>clk/4</code> 。
CKCON.2-0	MD2~MD0 控制使用 <code>MOVX</code> 指令访问外部数据存储器的指令周期数

MD2~MD0 对应的周期数如下：

MD2~MD0	MOVX周期	读/写选通脉冲宽度	选通脉冲时间@25MHZ
000	2	2	80ns
001	3(默认)	4	160ns
010	4	8	320ns
011	5	12	480ns
100	6	16	640ns
101	7	20	800ns
110	8	24	960ns
111	9	28	1120ns

11、中断

11.1 中断源及入口地址

BF6910AX/11AX 中使用的中断信息如表 2 所示：

中断源	中断向量	优先级	中断标志	使能位	优先级控制
-----	------	-----	------	-----	-------



INT_EXT(外部中断)	03h	0	TCON.1(IE0)	IE.0	IP.0
TF0(timer0 中断)	0Bh	1	TCON.5(TF0)	IE.1	IP.1
保留	13h	2	-	-	-
TF1(timer1 中断)	1Bh	3	TCON.7(TF1)	IE.3	IP.3
TI/RI (串口 UART0)	23h	4	SCON.0(RI) SCON.1(TI)	IE.4	IP.4
保留	2Bh	-	-	-	-
保留	33h	-	-	-	-
保留	3Bh	-	-	-	-
INT_CTK (CTK 中断)	43h	8	EXIF.4(IE2)	EIE.0	EIP.0
INT_IIC (IIC 通信中断)	4Bh	9	EXIF.5(IE3)	EIE.1	EIP.1
保留	53h	-	-	-	-
保留	5Bh	-	-	-	-
WDTI(看门狗中断)	63h	12	EICON.3	EIE.4	EIP.4

其中INT_EXT、INT_IIC和WDT中断能够在睡眠模式下对IC进行唤醒。其中INT_EXT(外部中断)有两个引脚(int0、int1)可以作为中断触发，但是使用同一中断向量。

11.2 中断 SFR

中断控制中用到的 SFR 有：

- IE---SFR 地址 A8h
- IP---SFR 地址 B8h
- EXIF---SFR 地址 91h
- EICON---SFR 地址 D8h
- EIE---SFR 地址 E8h
- EIP---SFR 地址 F8h
- INT_SEL---SFR 地址 C4h

IE 和 IP 寄存器用来提供标准中断源的使能位和控制中断源的优先级。EXIF、EICON、EIE 和 EIP 寄存器位扩展的中断源提供标志位、允许控制位和优先级控制位。INT_SEL 寄存器用来选择外部中断(INT_EXT)触发引脚为 INT0 或者 INT1。

以下为各寄存器的介绍。

IE 寄存器-SFR A8h

位	功能
IE.7	EA-中断允许位。EA=0 屏蔽所有的中断(EA 优先于中断源各自的中断使能位)。EA=1，中断打开，每个中断源的中断请求是允许还是被禁止，还需由各自的允许位确定。
IE.6-5	保留



IE.4	ES0-串口 0 中断允许位。ES0=0 时，禁止串口 0 申请中断(TI_0 和 RI_0)。ES0=1，允许 TI_0 和 RI_0 标志位申请中断。如果串口 0 没有启用(serial=0)，ES0 存在但不使用。
IE.3	ET1-定时器 1 溢出中断允许位。ET1=0，禁止定时器 1(TF1)申请中断。ET1=1，允许 TF1 标志位申请中断。
IE.2	保留
IE.1	ET0-定时器 0 溢出中断允许位。ET0=0，禁止定时器 0(TF0)申请中断。ET0=1，允许 TF0 标志位申请中断。
IE.0	EX0-INT_EXT 允许位。EX0=0，禁止 INT_EXT 申请中断。EX0=1，允许 INT_EXT 申请中断。

IP 寄存器-SFR B8h

位	功能
IP.7-5	保留
IP.4	PS0-串口 0 中断优先级选择位。PS0=0 时，为低优先级。PS0=1 时，为高优先级。如果串口 0 没有启用(serial=0)，ES0 存在但不使用。
IP.3	PT1-定时器 1 中断优先级选择位。PT1=0 时，为低优先级。PT1=1 时，为高优先级。
IP.2	保留
IP.1	PT0-定时器 0 中断优先级选择位。PT0=0 时，为低优先级。PT0=1 时，为高优先级。
IP.0	PX0-INT_EXT 优先级选择位。PX0=0 时，为低优先级。PX0=1 时，为高优先级。

EXIF 寄存器-SFR 91h

位	功能
EXIF.7-6	保留
EXIF.5	IE3-INT_IIC 标志位。IE3=1 表示在 INT_IIC 检测到下降沿。IE3 由软件清 0。如果中断允许，可以通过软件置位 IE3 来产生中断。
EXIF.4	IE2-INT_CTK 标志位。IE2=1 表示在 INT_CTK 检测到上升沿。IE2 由软件清 0。如果中断允许，可以通过软件置位 IE2 来产生中断。
EXIF.3	保留，默认为 1。
EXIF.2-0	保留，默认为 0。

EICON 寄存器-SFR D8h

位	功能
EICON.7	保留
EICON.6	保留，默认为 1。
EICON.5	保留



EICON.4	保留
EICON.3	WDTI-看门狗定时器中断标志位。WDTI=1 表示在 wdti 脚检测到中断申请。WDTI 位在中断返回之前由中断程序清 0。否则，中断会再一次发生。如果中断允许，可以通过软件置位 WDTI 来产生中断。
EICON.2-0	保留，默认为 0。

EIE 寄存器-SFR E8h

位	功能
EIE.7-5	保留，默认为 1
EIE.4	EWDI-看门狗定时器中断允许位。EWDI=0，禁止 wdti 申请中断。EWDI=1，允许 wdti 脚申请中断。
EIE.3-2	保留
EIE.1	EX3-INT_IIC 允许位。EX3=0，禁止 INT_IIC 申请中断。EX3=1，允许 INT_IIC 申请中断。
EIE.0	EX2-INT_CTK 允许位。EX2=0，禁止 INT_CTK 申请中断。EX2=1，允许 INT_CTK 申请中断。

EIP 寄存器-SFR F8h

位	功能
EIP.7-5	保留，默认为 1
EIP.4	PWDI-看门狗定时器中断优先级选择位。PWDI=0，wdit 中断为低优先级。PWDI=1，wdit 中断为高优先级。
EIP.3	保留
EIP.2	保留
EIP.1	PX3-INT_IIC 优先级选择位。PX3=0，为低优先级。PX3=1，为高优先级。
EIP.0	PX2-INT_CTK 优先级选择位。PX2=0，为低优先级。PX2=1，为高优先级。

IO/复用功能选择寄存器 (IO_CON) -SFR C4h

位	功能
IO_CON.7-5	保留
IO_CON.4-3	UART_IO_SEL: UART/IO 选择位。 00: 不选择 UART 输出 01: 选择 PA<0>/PA<1>作为串口通信端口 10: 选择 PC<0>/PC<1>作为串口通信端口



IO_CON.2-1	INT_SEL: 外部中断选择。 当 Bit2=1 时, 选择引脚(INT1)作为中断触发引脚; Bit2=0 时, 选择引脚(PC<5>)作为普通 IO 口。 当 Bit1=1 时, 选择引脚(INT0)作为中断触发引脚; Bit1=0 时, 选择引脚(PC<4>)作为普通 IO 口。
IO_CON.0	PWM_SEL: PWM 输出使能位。1 选择 PWM 输出; 0 作为 IO 口输出。

11.3 中断响应

当发生中断申请时, CPU根据中断服务程序(ISR)来确定中断的种类, 中断地址和优先级如表2所示。CPU完整的执行ISR, 除非有优先级高的中断源申请中断。每个ISR后有RETI(中断返回)指令。执行RETI指令后, CPU继续执行在中断没有发生之前的程序。

ISR 只能被优先级更高的中断申请中断。也就是, 低优先级的 ISR 能被高优先级的中断申请中断。高优先级的中断 ISR 能被掉电中断(extd_intr=1)中断。

BF6910AX/11AX 执行完当前指令后才响应中断请求。如果正在执行的指令是 RETI 指令, 或者访问 IP、IE、EIP、EIE 寄存器时, 需要执行一条其他指令后才会响应中断请求。

11.4 中断优先级

BF6910AX/11AX 有两个中断优先级: 中断级和默认优先级。中断级(最高级、高级和低级)优先于默认优先级。如果允许, 掉电中断是唯一一个最高级的中断源。其他的中断源可以设置为高优先级或者低优先级。

每个中断源既可以分配优先级(高或者低), 还有默认的优先级。同一级别中的中断源(例如都为高优先级)的优先级由默认的优先级决定。

正在进行的中断服务程序只能被优先级高的中断请求中断。

11.5 中断采样

内部定时器和串口是通过各自的 SFR 中的中断标志位来发生中断请求。当每 1 个指令周期的第 4 个时钟周期(C4)结束时, 在 clk 的上升沿对外部中断进行采样。

INT_EXT 是低电平有效, 并且可以通过 TCON SFR 中的 IT0 位来设置选择边缘触发或者电平触发。例如, 当 IT0=0 时, INT_EXT 为边缘触发, 在采集到 INT_EXT 脚出现由高到低的电平变化时, IE0 置 1。

其他的四个外部中断只能采用边缘触发的方式, int2 和 int4 是上升沿有效, int3_n 和 int5_n 为下降沿有效。

掉电中断(pfi)和看门狗定时器中断(wdti)为高电平有效, 并且每个指令周期被采样一次。掉电中断是电平触发, 看门狗定时器中断是边缘触发。

为了确定边缘触发中断被检测到, 相应的端口要首先保持 4 个时钟的高电平, 然后保持 4 个时钟的低电平。电平触发中断不锁存, 要保持电平状态直到中断响应。



11.6 中断等待

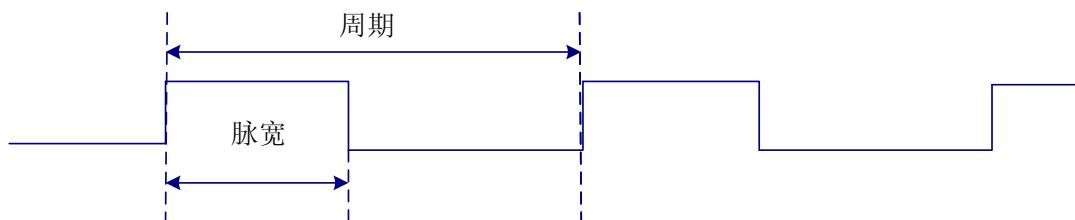
中断的响应时间由 BF6910AX/11AX 当前状态决定。最快的响应时间是 5 个指令周期：1 个周期用来检测中断请求，其他 4 个用来执行长调用(LCALL)至 ISR。

当 BF6910AX/11AX 在执行 RETI 指令，并且后面为 MUL 或者 DIV 指令时，中断等待的时间最长(13 个指令周期)。这 13 个指令周期分别为：1 个周期用来检测中断请求，3 个用来完成 RETI 指令，5 个用来执行 DIV 或者 MUL 指令，4 个用来执行长调用(LCALL)至 ISR。在这种情况下，响应时间为 $13 \times 4 = 52$ 个时钟周期。

12、PWM 输出模块

PWM 脉宽调制模块输出包含一个时基（周期）和一段输出高电平的时间（脉宽）。周期和脉宽都可以通过编程进行配置，周期寄存器 **PWM_DIV** 为 6 位的寄存器，脉宽寄存器 **PWM_DUTY** 为 8 位的寄存器。PWM 周期范围为 1K-20K。

PWM 输出波形如下图所示：



PWM 周期计算公式：

$$\text{PWM 周期} = 512 * \text{PWM_DIV} * T_{\text{MCU}}$$

$$\text{PWM 频率} = 1 / \text{PWM 周期}$$

脉宽的计算公式：

$$\text{占空比} = \text{PWM_DUTY} / 255$$

与 PWM 模块相关寄存器如下：

PWM 周期寄存器 (PWM_DIV)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认
PWM_DIV	-								0x01

PWM 脉宽寄存器 (PWM_DUTY)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认
PWM_DUTY									0x00

PWM 使能寄存器 (PWM_CS)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认
PWM_CS				-					0x00

PWM 使能信号，为 1 有效。

IO/复用功能选择寄存器 (IO_CON)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认
IO_CON		-		UART_IO_SEL	INT_SEL	PWM_SEL			0x00



Bit7-Bit5: 保留

Bit4-Bit3: UART/IO 选择位。

00: 不选择 UART 输出

01: 选择 PA<0>/PA<1>作为串口通信端口

10: 选择 PC<0>/PC<1>作为串口通信端口

Bit2-Bit1: 外部中断使能。当 Bit2=1 时, 选择引脚(INT1)作为中断触发引脚; Bit2=0 时, 选择引脚(PC<5>)作为普通 IO 口。

当 Bit1=1 时, 选择引脚(INT0)作为中断触发引脚; Bit1=0 时, 选择引脚(PC<4>)作为普通 IO 口。

Bit0: PWM 输出使能位。1 选择 PWM 输出; 0 作为 IO 口输出。

PWM 模块配置步骤:

1、设置 IO_CON<0>, 选择 PWM 输出

2、设置 PWM 的周期和占空比, 即 PWM_DIV 寄存器和 PWM_DUTY 寄存器

3、设置 PWM_CS, 对 PWM 模块供电

13、UART0

BF6910AX/11AX 提供了一个串口供用户使用, 有两路引脚可以使用, 占用同一个中断向量。UART0 可工作在同步和异步通信方式下。在同步模式中, BF6910AX/11AX 产生串口时钟, UART0 工作在半双工模式下。在异步模式中, UART0 工作在全双工模式下。在任一模式下, BF6910AX/11AX 将接收到的数据缓存在一个临时寄存器中, 如此一来, 即使程序仍未读取当前 SBUF0 中的数据, UART0 也能够接收数据。

13.1 UART0 中的 SFR

SCON 寄存器

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	0x00

Bit7: SM0 串口 0 方式选择位 0

Bit6: SM1 串口 0 方式选择位 1, 对应方式如下:

SM0 SM1 方式		
0	0	0
0	1	1
1	0	2
1	1	3

Bit5: SM2 多机通信使能控制位。在方式 2 和方式 3 中, SM2 使能控制多机通信的功能。

在方式 2 和方式 3 中, 如果 SM2=1, 当接收到的第 9 位为 0, RI 将不会被激活, 不产生中断; 在方式 1, 只有有效的停止位被接收才触发 RI 置位。在方式 0, SM2



做为波特率选择控制位；当 SM2=0，波特率使 clk/12；当 SM2=1，波特率是 clk/4。

Bit4: REN 接收使能控制位。当 REN=1 时，允许接收数据。

Bit3: TB8 定义工作在方式 2 或方式 3 时第九位数据位的值。

Bit2: RB8 工作在方式 2 和方式 3 时，RB8 保存接收到的第 9 位数据。工作在方式 1，RB8 保存接收停止位的状态，在方式 0 时，RB8 不使用。

Bit1:TI 发送中断标志位。表示数据字节发送完成。在方式 0 中，TI 在第 8 位数据之后被置位。在其他方式，TI 在停止位出现在 TXD 引脚后置位。TI 只能软件清零。

Bit0:RI 接收中断标志位。显示接收到串行数据。在方式 0，RI 在接收第 8 位后置位。在方式 1，RI 将在输入的停止位被采样后置位。在方式 2 和方式 3，RI 在 RB8 被采样后置位。RI 必须软件清零。

SBUF 寄存器

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认
SBUF									0x03

Bit7~Bit0: SBUF [7:0]: UART0缓冲器位7-0 (MSB-LSB) 该SFR实际上是两个寄存器：一个发送缓冲器和一个接收缓冲器。当数据被写入SBUF时，它进入发送移位寄存器等待串行发送。向SBUF写入一个字节即启动发送过程。读SBUF时返回数据来自接收缓冲器。

注：方式 1 和方式 3 的波特率设置需要使用到 Timer1 定时。

IO/复用功能选择寄存器 (IO_CON)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认
IO_CON	保留			UART_IO_SEL	INT_SEL	PWM_SEL			0x00

Bit7-Bit5: 保留

Bit4-Bit3: UART_IO_SEL: UART/IO 选择位。

00: 不选择 UART 输出

01: 选择 PA<0>/PA<1>作为串口通信端口

10: 选择 PC<0>/PC<1>作为串口通信端口

Bit2-Bit1: 外部中断使能。当 Bit2=1 时，选择引脚(INT1)作为中断触发引脚；Bit2=0 时，选择引脚(PC<5>)作为普通 IO 口。

当 Bit1=1 时，选择引脚(INT0)作为中断触发引脚；Bit1=0 时，选择引脚(PC<4>)作为普通 IO 口。

Bit0: PWM 输出使能位。1 选择 PWM 输出；0 作为 IO 口输出。



13.2 UART0 工作方式

UART0 提供四种工作方式（一种同步方式和三种异步方式），通过设置SCON寄存器中的配置位选择。这四种方式提供不同的波特率和通信协议。下面的表概述了这四种方式。

模式	同步/异步	波特率时钟	数据位	开始位/结束位	第九个数据位功能
0	同步	CLK/4或CLK/12	8	无	无
1	异步	Timer1	8	1个开始位, 1个停止位	无
2	异步	CLK/32或CLK/64	8	1个开始位, 1个停止位	0, 1, 奇偶校验
3	异步	Timer1	9	1个开始位, 1个停止位	0, 1, 奇偶校验

方式0：同步方式

方式0 提供同步、半双工通信。在RXD 引脚上发送和接收数据，TXD 引脚提供发送和接收的移位时钟。MCU 必须是主器件，因为它要为两个方向的数据传输产生移位时钟。执行一条写SBUF 寄存器的指令时开始数据发送。发送/接收的数据为8 位，LSB 在先，在第8 个位时间结束后发送中断标志TI (SCON.1) 置位。当接收允许位REN (SCON.4) 被设置为1 并且接收中断标志RI (SCON.0) 被清0 时开始数据接收。在第8位被移入后一个周期RI 标志置位，接收过程停止，直到软件清除RI位。如果中断被允许，在TI或RI置位后将发生一次中断。

方式0 的波特率是clk/12或clk/4，这主要依赖于SM2的值。当SM2=0,波特率为clk/12; 当SM2=1， 波特率为clk/4。

方式1：8 位UART，可变波特率

方式1 提供标准的异步、全双工通信，每个数据字节共使用10位：一个起始位、8个数据位 (LSB在先) 和一个停止位。数据从TXD引脚发送，在RXD引脚接收。在接收时，8个数据位存入SBUF，停止位进入RB8 (SCON.2)。当执行一条向SBUF寄存器写入一个字节的指令时开始数据发送。在发送结束时（停止位开始）发送中断标志TI (SCON.1) 置位。在接收允许位REN (SCON.4) 被设置为1之后任何时间都可以开始数据接收。收到停止位后如果满足下述条件则数据字节将被装入接收寄存器SBUF：RI为逻辑0，并且如果SM2为1，则停止位必须为1。

如果这些条件满足，则8位数据被存入SBUF，停止位被存入RB8，RI标志被置位。如果这些条件不满足，则不装入SBUF和RB8，RI标志也不被置1。如果中断被允许，在TI或RI置位时将产生中断。

方式1 的波特率是定时器溢出时间的函数。UART0可以使用定时器1工作在8位自动重装载方式。每次定时器发生溢出，即从全1 (定时器1为0xFF) 返回到0时向波特率电路发送一个时钟脉冲。

下面给出了方式1 的波特率方程，其中：T1M 为定时器1时钟选择位 (CKCON.4) ，TH1 是定时器1的8位重装载寄存器。

使用定时器1的方式1波特率：

当SMOD0 = 0 时：

方式1波特率 = 1/32 × 定时器1溢出率



当SMOD0 = 1 时：

方式1波特率 = $1/16 \times$ 定时器1溢出率

定时器1溢出率由定时器1 的时钟源 (T1CLK) 和重载值 (TH1) 确定。

定时器1溢出率 = $T1CLK / (256 - TH1)$

其中T1CLK根据定时器1时钟选择位T1M (CKCON.4) 可选择为clk/12(T1M=0)或clk/4(T1M=1)。

波特率的最终计算公式：

$$\text{BaudRate} = (2^{\text{SMOD0}}/32) * (3^{\text{T1M}}/12) * (\text{clk}/(256-\text{TH1}))$$

方式2：9 位UART，固定波特率

方式2 提供异步、全双工通信，每个数据字节共使用11 位：一个起始位、8个数据位 (LSB 在先)、一个可编程的第九位和一个停止位。在发送时，第九数据位由TB8 (SCON.3) 中的值决定。它可以被赋值为PSW中的奇偶标志P。在接收时，第九数据位进入RB8 (SCON.2)，停止位被忽略。

当执行一条向SBUF寄存器写入一个字节的指令时开始数据发送。在发送结束时（停止位开始）发送中断标志TI (SCON.1) 置位。在接收允许位REN (SCON.4) 被设置为1 后的任何时间都可以开始数据接收。收到停止位后如果RI为0并且满足下述条件之一则数据字节将被装入到接收寄存器SBUF：

1. SM2为0；

2. SM2为1，接收的第九位为1，并且接收到的地址与UART0的地址匹配。

如果上述条件满足，则8位数据被存入SBUF，第九位被存入RB8，RI标志被置位。如果这些条件不满足，则不装入SBUF和RB8，RI标志也不被置1。如果中断被允许，在TI或RI置位时将产生中断。方式2波特率为clk/32 或clk/64，由寄存器PCON中的SMOD0位决定。

方式2波特率 = $2^{\text{SMOD0}} * \text{CLK} / 64$

方式3：9 位UART，可变波特率

方式3 使用方式2的传输协议，波特率的确定方式与方式1相同。方式3操作使用11位：一个起始位、8个数据位 (LSB在先)、一个可编程的第九位和一个停止位。用定时器1溢出产生波特率。方式3的发送接收的工作方式与方式2类似，而波特率产生的方法跟方式1相似，因此方式3是方式1的波特率产生方式与方式2协议组合而成的一种工作方式。

13.3 多机通信

当SM2 (SCON.5) 被置位，方式2和方式3通过使用第九数据位和软件识别支持一个主处理器与一个或多个从处理器之间的多机通信。在多机通信模式下，接收到的第九位数据被保存到RB8(SCON.2)，当接收到停止位后，串口中断只有在RB8=1时才被激活。

一个典型的多机通信应用就是当主机想要向一个或多个从机发送数据数据时，它先发送一个用于选择目标从机的地址字节。地址字节与数据字节的区别是：地址字节的第九位为逻辑1；数据字节的第九位总是设置为逻辑0。

当SM2=1时，主机发送数据字节，从机将不产生中断。但是主机发送地址字节，所有的从机都将产生一个中断，从机在这个中断服务程序中判断本机是否被寻址，从机的地址分配

由软件实现。被寻址的从机清除SM2位，等待接收数据。而未被寻址的从机将保持SM2为1以忽略主机发送的数据字节。

14、IIC 通信

本章节介绍电容传感器的标准通信模式。BF6910AX/11AX 支持标准 IIC 通信和快速 IIC 通信。

BF6910AX/11AX 中 IIC 通信具有以下特点：

- IIC 从机支持 7 位的寻址方式，兼容 10 位寻址方式
- 传输速率：100Kbps、400Kbps
- 中断机制，idle 模式可唤醒芯片
- 延长时钟低电平的功能
- 异常通信情况的诊断

在通信过程中，从机需要支持主机以下操作：

- 从 TS 设备读取数据
- 向 TS 设备发送命令

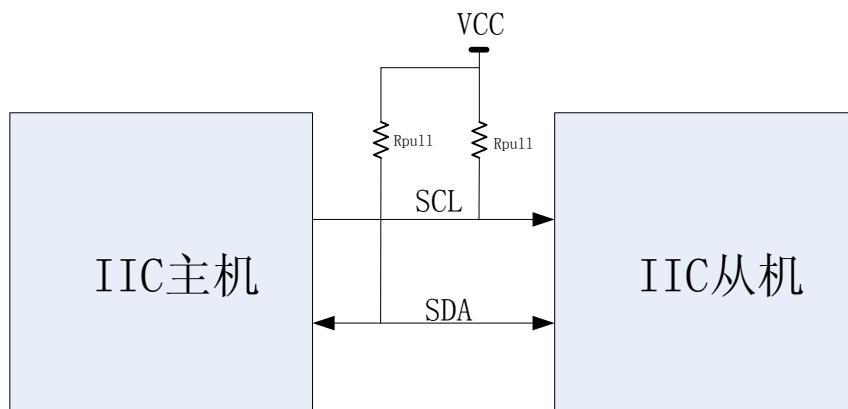


图 6 IIC 应用

如图 6 所示，主机和从机之间由 SCL(串行时钟)线、SDA(串行数据)线连接连接，SCL、SDA 必须接上拉电阻。当 TS 设备有触摸相关的动作，比如触摸、滑动、手指离开等姿势发生时，主机可通过 IIC 通信来读取从机的触摸状态。

14.1 主机写从机

当主机通过 IIC 向从机写数据时，可以通过对寄存器操作的方式来写数据。当从机接收到数据时，将数据保存至特定的寄存器。通信格式如下图所示。在图中，IIC 表头由 IIC 从机地址和 R/W 位组成，在写模式中，R/W=0。



图 7 主机通过 IIC 写单个的寄存器的数据

当主机需要对从机写一系列数据时，可以通过以下的方法来实现。主机首先发送需要写

数据的寄存器的地址，在从机接收到地址后将地址保留。主机接下来开始发送数据，在从机接收到数据后，将数据写入相应的寄存器，并将之前接收到的地址进行自增，从而确保将接收到的下一个字节写入对应的寄存器。该过程的框图如下：

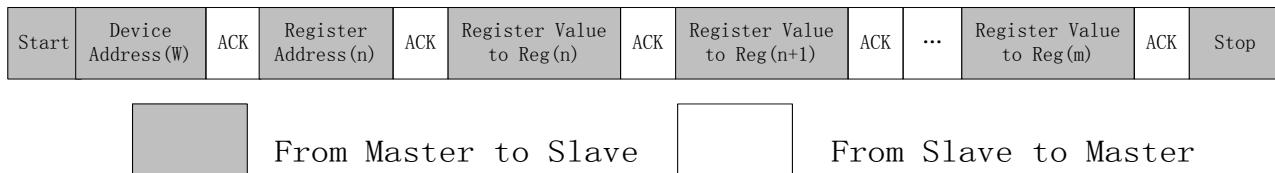


图8 主机通过IIC写连续寄存器的数据

14.2 主机读从机

读取寄存器中的数据与写寄存器的过程相似。主机通过写寄存器的地址，然后从TS设备中相应的寄存器读取数据。一般的寄存器通过IIC总线读数据包格式如图9所示。



图9 主机通过IIC读单个寄存器的数据

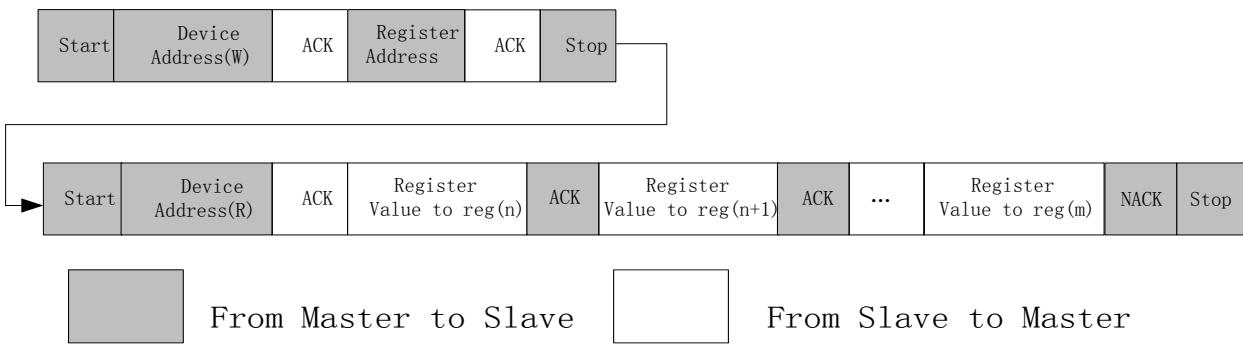


图10 主机通过IIC读连续寄存器的数据

当主机需要读取多个数据时，可以通过连续读取数据的方式来实现。主机发送需要读取的第一个寄存器的地址，从机接收到数据后将数据保存，并将相对应的寄存器中的数据保存至数据发送缓冲器中。接下来主机发送读数据命令，从机将准备好的数据移位传出，并将接收到的地址值自增。当从机接收到主机的有效应答信号时，继续准备数据。在主机读取了所需的数据后，主机发送非应答信号，从机则停止准备数据。由于主机在整个过程中进行了写操作（对从机写所需读取数据的寄存器的地址）和读操作（主机读取数据）的切换，所以在过程中间必须有重启信号（先停止，再开始）。

14.3 IIC 中的 SFR

IIC 模块一共用到 4 组寄存器，它们分别是：

IIC 状态寄存器 (IICSTAT)





IICSTAT	STAT_S	STAT_P	STAT_RW	STAT_DA	STAT_BF	ACKSTAT	WCOL	RECOV
---------	--------	--------	---------	---------	---------	---------	------	-------

IIC 状态寄存器，用于反应通信过程中的状态。其中 Bit0、Bit1 可读可写，其它位只读。

STAT_S: 起始信号状态位。当检测到起始信号之后 **STAT_S** 就会置位，表示总线处于忙的状态。

STAT_P: 停止信号状态位。当检测到停止信号之后 **STAT_P** 就会置位，**STAT_S** 被硬件清零，表示通信结束，总线处于空闲状态。

STAT_RW: 读写标志位。此标志位记录地址匹配后，从地址字节中获取的读写信息位。**STAT_RW=1** 表示主机读取从机的操作，**STAT_RW=0** 表示主机写从机的操作。起始信号、停止信号、非应答信号（NACK）都会清零 **STAT_RW**。此状态位的改变发生在第九个时钟的下降沿。

STAT_DA: 地址数据标志位。它标志当前接收或者发送的字节是地址或者是数据。**STAT_DA=0** 标志接收或者发送的字节是地址；**STAT_DA=1** 标志接收或者发送的字节是数据。开始信号、停止信号、非应答信号对此状态位没有任何影响。此状态位的改变发生在第 8 个时钟的下降沿。

STAT_BF: 缓冲器满标志位。它标志这收发缓冲器 当前是满的或者是空的。**STAT_BF=0** 表示缓冲器没有接收数据，缓冲器为空；**STAT_BF=1** 表示缓冲器接收到数据，缓冲器已满。此状态位只能间接的置位和清零，不能直接的操作。

地址匹配并且 **STAT_RW=0** 的情况下，在第 8 个时钟的下降沿之后 **STAT_BF** 将置位，标志着 IICBUF 接收到数据。在执行中断程序期间应该读取 IICBUF，读取 IICBUF 的操作会间接的清零 **BF** 标志位。如果不读取 IICBUF，主机继续发送数据，则会发生接收溢出，因为 IICBUF 里仍然保存上一个接收的数据，这样从机就会发送一个 NACK，同时 IICBUF 里的数据不会改变，仍为上一个接收到的字节。

地址匹配并且 **STAT_RW=1** 的情况下，从机接收到地址字节后 **STAT_BF** 标志位不会置位；**STAT_RW=1** 表示主机读取从机的操作，从机需要把数据写入 IICBUF，从机写 IICBUF 的操作会置位 **STAT_BF**，然后软件置位 **SCLEN**，释放时钟线；主机就会发送同步时钟，第 8 个时钟过后，IICBUF 里的数据发送出去之后，**STAT_BF** 被硬件清零。

ACKSTAT: 应答状态位。不管主机是读操作还是写操作，从机都会在第 9 个时钟的上升沿采样数据线，记录应答信息。应答位分为有效应答位 **ACK** 和非有效应答位 **NACK**。也就是说第 9 个时钟的上升沿采样到数据为“0”，表示有效应答 **ACK**，同时 **ACKSTAT** 被清零，如果采样到数据“1”则 **ACKSTAT** 被置位，表示非应答。非应答信号之后，主机就会发送停止信号宣告通信结束。重启动信号、停止信号会清零此状态位。

WCOL: 写冲突标志位。IICBUF 只有在 **STAT_RW=1**，并且 **SCLEN=0** 的情况下才可以被写入。其它任何情况下试图写 IICBUF 是被禁止的，如果不满足上面的条件下发生了写 IICBUF 的操作，则数据不会被写入 IICBUF，同时写冲突标志位 **WCOL** 被置位，表示发生了写冲突，此标志位需要软件清零。

RECOV: 接收溢出标志位。在 IICBUF 满的情况下，也就是 IICBUF 里存在数据的情况下，IIC 有接收到新的数据，则会发生接收溢出，**RECOV** 会置位，同时 IICBUF 里的数据不会被更新，新接收的数据会丢失。此状态位也需要软件清零，否则的话会应影响后面的通信过程。



IIC 地址寄存器 (IICADD)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IICADD	IICADD<7:1>							-

IIC 地址寄存器，高 7 位可读可写，最低位保留未用（读取值为 0）。IICADD 寄存器相当于此芯片的 ID。根据 IIC 协议，主机在 STAT_S 信号之后要发送从机的地址码，从机在接收到地址码之后就是与 IICADD 的内容进行比较来进行身份验证的。只有地址匹配的情况下才可以进行通信。从机的 7 位地址通过软件写寄存器 IICADD 来设置。

IIC 发送接收缓冲器 (IICBUF)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IICBUF	IICBUF<7:0>							

IIC 数据收发缓冲器，可读可写的寄存器。

在发送状态下，把数据写入 IICBUF 中后，在主机的同步时钟作用下，数据依次移位发送出去，高位在前。8 个时钟过后，一个字节发送完毕。

在接收状态下，在主机的 8 个时钟过后，数据被写入 IICBUF 中去，第 9 个时钟过后会产生中断，告诉 MCU 读取 IICBUF 中的数据。

把数据写入 IICBUF 此操作是有条件的，只有 STAT_RW=1，并且 SCLEN=0 的情况下才可以把数据写入 IICBUF 中；否则写 IICBUF 的操作是被禁止的。也就是说条件不满足的情况下，写 IICBUF 的操作不能成功的，数据写不进去，IICBUF 的数据不会改变，同时也会造成写冲突。

例如：IICBUF 已经有数据 55h，在写 IICBUF 的条件不满足的情况下，欲把数据 00h 写入 IICBUF。结果是 IICBUF 里的数据仍然是 55h，同时写冲突标志位 WCOL 置位。

IIC 控制寄存器 (IICCON)

SFR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IICCON	保留			SCL_CLR_EN	SCLEN	SR	IICEN	

IIC 控制寄存器，Bit3~Bit0 可读可写。

低 4 位用于 IIC 模块，其中 IICEN 是 IIC 模块的使能信号，只有 IICEN=1 时，电路才工作。SR 是转换速率控制位，SR=1 转换速率控制关闭，端口适应于 100Kbps 的通信，SR=0 转换速率打开，端口适应于 400Kbps 的通信。SCLEN 是时钟使能控制位，虽然从机不能产生通信时钟，但是根据协议从机可以延长时钟的低电平时间。SCLEN=0 时钟线被锁定在低电平，SCLEN=1 释放时钟线。延长时钟低电平的前提是 IICEN=1，否则内部电路不会对 IIC 总线产生任何影响。SCLEN 常用来延长低电平的时间，使主机进入等待状态，这样从机就有足够的时间来处理数据。SCL_CLR_EN 是中断拉低时钟线控制位，SCL_CLR_EN=1 时使能中断拉低时钟线的功能，SCL_CLR_EN=0 时不使能中断拉低时钟线的功能。中断拉低时钟线的前提也是 IICEN=1，否则内部电路不会对 IIC 总线产生任何影响。

在 STAT_RW=1，主机读取从机数据的时候，SCL_CLR_EN 应该配置为 0，SCLEN 在

接收到地址字节和发送完一个字节并且主机发送 ACK 的时候，SCLEN 会被硬件自动拉低，迫使主机进入等待状态。从机要释放 IIC 时钟，需要下面两个操作，顺序不能改变：

把要发送的数据写入 IICBUF 中；

软件置位 SCLEN。

这样能确保拉高 SCL 之前，IICBUF 中已经写入将要发送的数据。

在 STAT_RW=0 的情况下，依据主机的通信速率和处理中断的时间来决定是否拉低时钟线。当在 9 个 IIC 时钟内能处理完中断并退出中断时，SCL_CLR_EN=0 不使能中断拉低时钟线的功能，此时在中断到来时不硬件自动拉低时钟线。当不能在 9 个 IIC 时钟内处理完中断并退出时，SCL_CLR_EN=1 使能中断拉低时钟线的功能，此时在中断到来时硬件自动拉低时钟线，迫使主机进入等待状态，当写入 IICBUF 中的数据被从机读出后，软件置位 SCLEN。从机要释放 IIC 时钟，需要下面两个操作，顺序不能改变：

(1) 把 IICBUF 的数据读出；

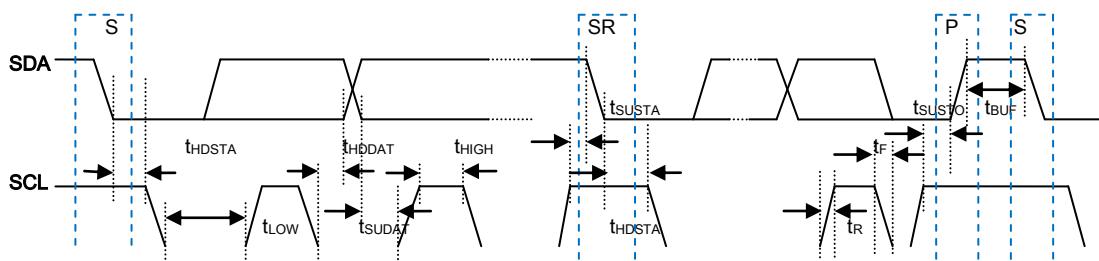
(2) 软件置位 SCLEN。

其中应该注意的问题：当每次通信开始时都要配置 SCL_CLR_EN，在发送和接收最后一个 Byte 数据时，软件应该关闭中断拉低时钟线的功能。

14.4 IIC 时序参数

参数	符号	标准模式		快速模式		单位
		最小值	最大值	最小值	最大值	
SCL 时钟频率	SCL	0	100	0	400	kbps
(重复)起始条件的保持时间	t _{HDSTA}	4.0	-	0.6	-	μs
SCL 时钟的低电平时间	t _{LOW}	4.7	-	1.3	-	μs
SCL 时钟的高电平时间	t _{HIGH}	4.0	-	0.6	-	μs
重复起始条件的建立时间	t _{SUSTA}	4.7	-	0.6	-	μs
数据保持时间	t _{HDDAT}	0	3.45	0	0.9	μs
数据建立时间	t _{SUDAT}	250	-	100	-	ns
停止条件的建立时间	t _{SUSTO}	4.0	-	0.6	-	μs
停止和启动条件之间的总线空闲时间	t _{BUF}	4.7	-	1.3	-	μs
SDA 和 SCL 信号的上升时间	t _R	-	1000	20+0.1C _b	300	ns
SDA 和 SCL 信号的下降时间	t _F	-	300	20+0.1C _b	300	ns
每个连接的器件低电平时的噪声容限	V _{nL}	0.1V _{CCC}	-	0.1V _{CCC}	-	V
每个连接的器件高电平时的噪声容限	V _{nH}	0.2V _{CCC}	-	0.2V _{CCC}	-	V

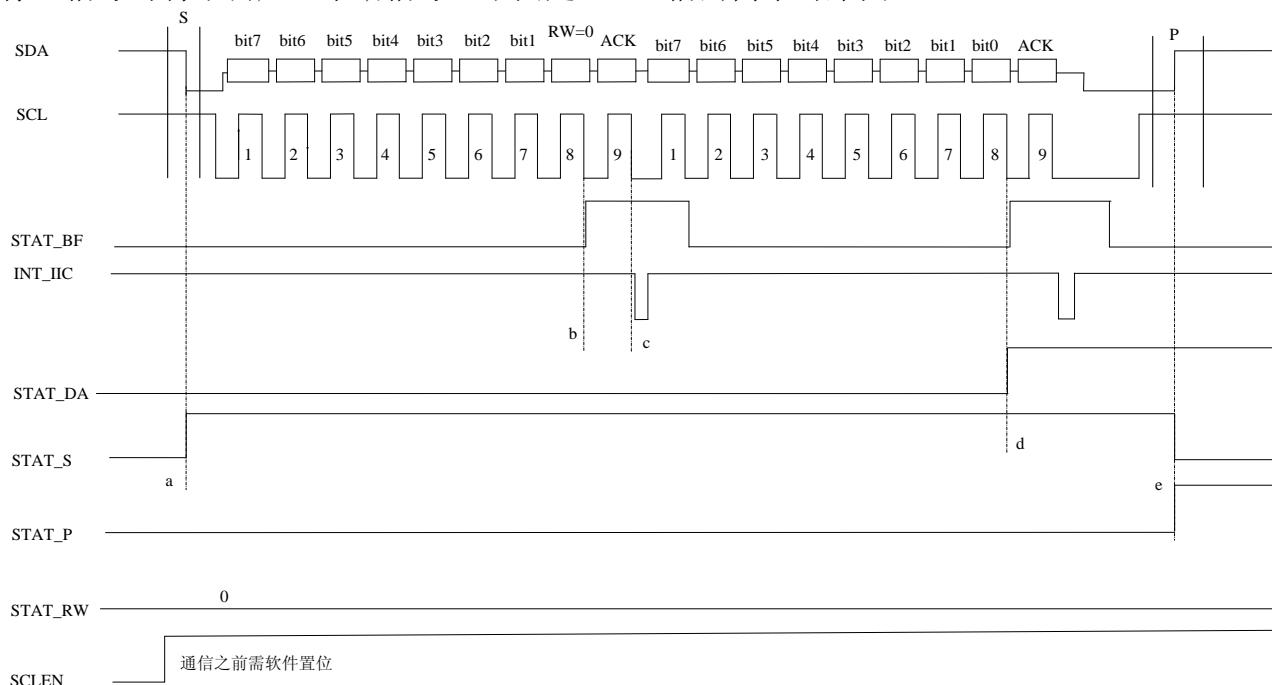
TA=-40°C ~ +85°C , VCC=1.65V ~ 5.5V , VCC=2.7V ~ 5.5V。



IIC时序图

14.5 IIC 说明

BF6910AX/11AX 采用硬件从机。当主机读/写数据时，从机接收到地址后，如果地址匹配，则产生中断，发送有效应答信号。并在主机写数据的第八个时钟后产生中断，主机发送停止信号时将不会产生中断信号。下面是 IIC 通信的简单时序图：



IIC 主机“写”示意图

图 13 主机写数据到从机示意图

图 13 所示是主机写操作时不拉低时钟线的示意图，从中可以看到 IIC 总线的变化情况以及一些内部信号的变化。

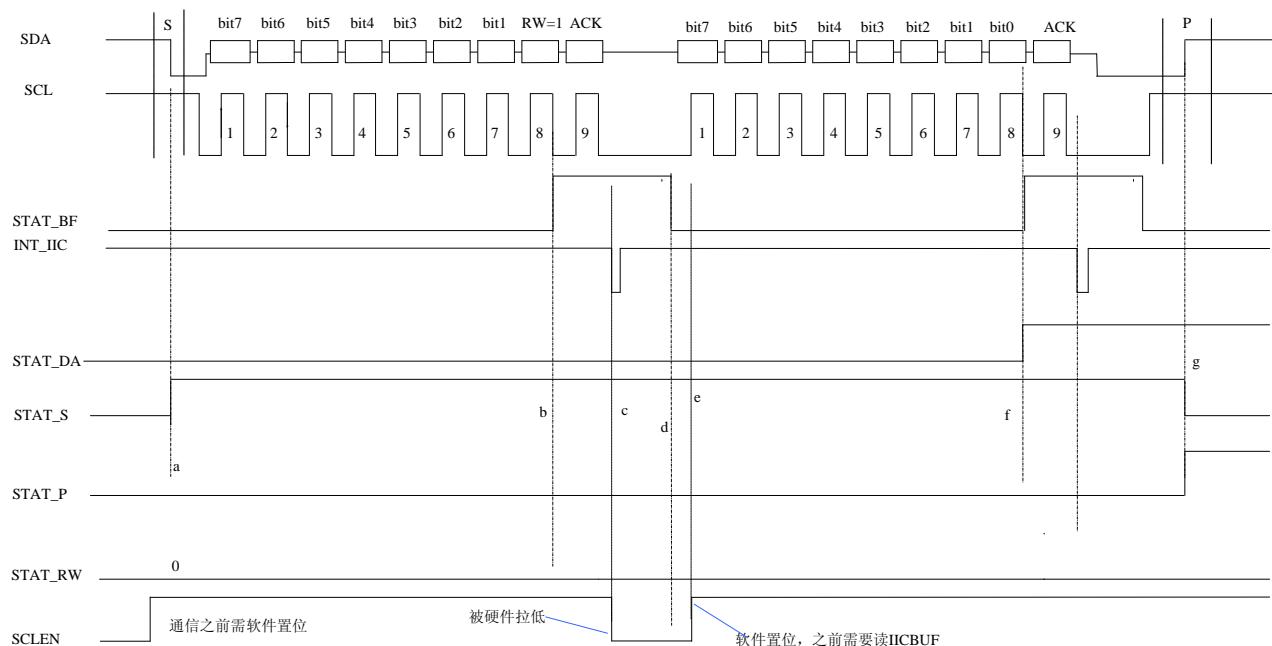
IIC 总线的变化符合 IIC 的协议要求，开始信号以及报文的传输符合协议要求。首先主机发送启动信号 STAT_S，从机检测到 STAT_S 信号之后置位 STAT_S 状态位，如图中虚线 a 所示。

然后主机发送地址字节和读写标志位，从机在接收到地址字节后硬件自动与自身地址比较，如果匹配则在第 8 个时钟的下降沿之后置位 STAT_BF，如虚线 b 所示。

在第 9 个时钟的下降沿之后会产生中断信号 INT_IIC，如虚线 c 所示。MCU 执行中断子程序期间需要读取 IICBUF，即使此数据没有用；读取 IICBUF 的操作会使 STAT_BF 间接的清零。

然后主机继续发送报文，在第 2 个字节的第 8 个时钟的下降沿之后 STAT_BF 同样被置位，同时 STAT_DA 标志位也会置位，标志当前接收到的字节是数据，如虚线 d 所示，停止信号对 STAT_DA 标志位没有影响即检测到停止信号 STAT_P，STAT_DA 标志位不会被清零；第 9 个时钟的下降沿之后同样会产生中断，中断子程序需要作同样的操作。

如果主机要发送多个字节，则可以继续发送，图 13 中仅仅示意主机发送一个数据的情况。最后主机在发送完毕所有的数据之后发送一个停止信号 **STAT_P**，标志着通信的结束，释放 IIC 总线，总线进入空闲状态。停止信号不产生中断。



IIC 主机“写拉低”示意图

图 14 主机写数据到从机示意图

图 14 所示是主机写操作时拉低时钟线的示意图，从中可以看到 IIC 总线的变化情况以及一些内部信号的变化。

IIC 总线的变化符合 IIC 的协议要求，开始信号以及报文的传输符合协议要求。首先主机发送启动信号 **STAT_S**，从机检测到 **STAT_S** 信号之后置位 **STAT_S** 状态位，如图中虚线 **a** 所示。

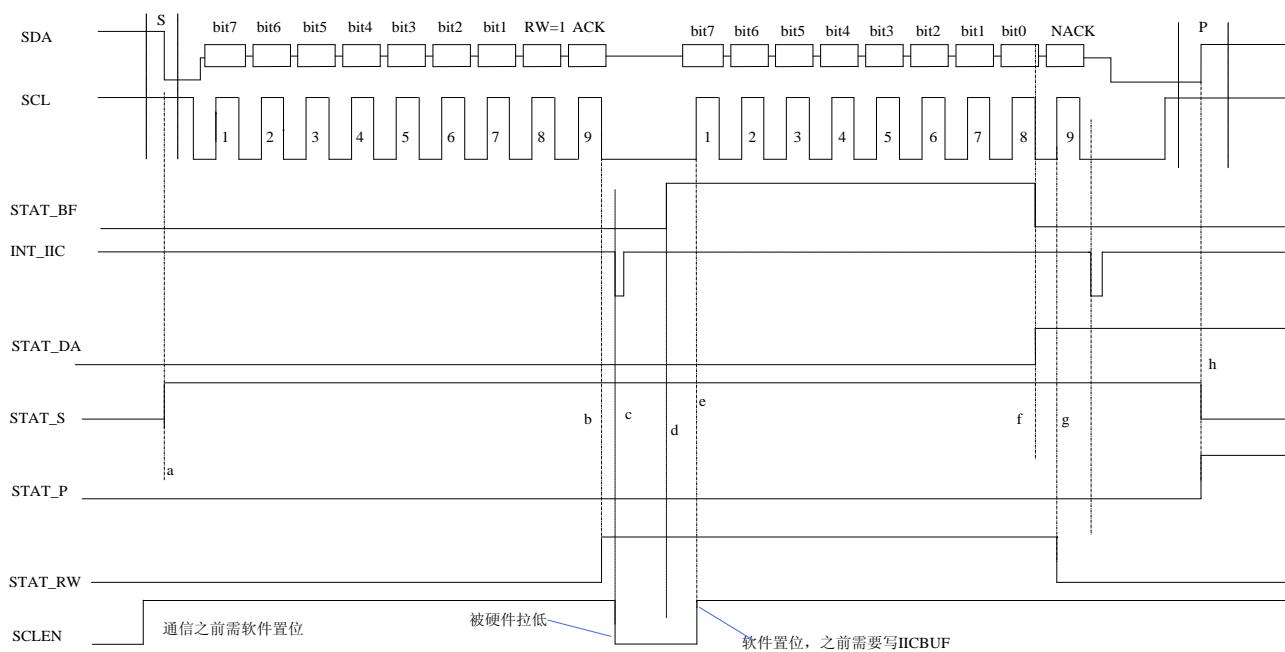
然后主机发送地址字节和读写标志位，从机在接收到地址字节后硬件自动与自身地址比较，如果匹配则在第 8 个时钟的下降沿之后置位 **STAT_BF**，如虚线 **b** 所示。在第 9 个时钟的下降沿之后会产生中断信号 **INT_IIC**，如虚线 **c** 所示。

第 9 个时钟的下降沿之后 **SCLEN** 会被硬件自动清零，此期间用于从机处理或者读取数据，即使此数据没有用，读取 **IICBUF** 的操作会使 **STAT_BF** 间接的清零，如虚线 **d** 所示。再软件置位 **SCLEN**，释放时钟线，如虚线 **e** 所示。

然后主机在检测到从机释放 **SCL** 之后，会继续发送同步时钟，在第 2 个字节的第 8 个时钟的下降沿之后 **STAT_BF** 同样被置位，同时 **STAT_DA** 标志位也会置位，标志当前接收到的字节是数据，如虚线 **f** 所示，停止信号对 **STAT_DA** 标志位没有影响即检测到停止信号 **STAT_P**，**STAT_DA** 标志位不会被清零；第 9 个时钟的下降沿之后同样会产生中断。同时停止信号不产生中断。

如果主机要发送多个字节，则可以继续发送，图 14 中仅仅示意主机发送一个数据的情况。需要注意的情况是在主机发送最后一笔数据时，不能拉低时钟线的功能。最后主机在发送完毕所有的数据之后发送一个停止信号 **STAT_P**，标志着通信的结束，释放 IIC 总线，总线

进入空闲状态。



IIC 主机“读”示意图

图 15 IIC 主机读取数据

如图 15 所示, 是主机读取从机的时序示意图。由图可知道总线的变化情况, 以及一些电路内部信号的变化情况。

主机发送 **STAT_S** 信号, 标志通信的开始, 如虚线 **a** 所示, 内部电路检测到 **STAT_S** 信号时序后置位状态标志位 **STAT_S**。

STAT_S 信号之后主机发送地址字节, 并且 **STAT_RW=1**, 表示主机读取从机。地址匹配的情况下在第九个时钟的下降沿之后, 状态位 **STAT_RW** 置位, 如虚线 **b** 所示, 同时也会产生中断信号, 如虚线 **c** 所示; 如果地址不匹配则 **STAT_RW** 不会置位, 也不会产生中断信号。

第 9 个时钟的下降沿之后 **SCLEN** 也会被硬件自动清零, 此期间用于从机处理或者准备数据, 然后把准备好的数据写入 **IICBUF**, 再软件置位 **SCLEN**, 释放时钟线。如虚线 **d** 所示, 在把数据写入 **IICBUF** 中后, **STAT_BF** 会置位, 标志这 **IICBUF** 是满的。如虚线 **e** 所示, 软件置位 **SCLEN**, 释放时钟线。

然后主机在检测到从机释放 **SCL** 之后, 会继续发送同步时钟, 读取从机的数据, 第 8 个时钟的下降沿之后, 一个字节数据发送完毕, **STAT_BF** 标志位被清零; 同时地址数据标志位也会置位, 表示当前发送的字节数据。如虚线 **f** 所示。

第 9 个时钟的下降沿之后会产生中断, 如果主机需要继续读取从机, 则回复有效应答位 **ACK**, 继续通信; 如果主机需要的数据已经读取完毕, 则回复无效应答 **NACK**, 然后发送停止信号 **STAT_P**, 终止通信。示意图中主机仅仅读取了一个数据后, 回复 **NACK**, 然后发送 **STAT_P** 信号, 终止了通信, 停止信号不产生中断。

在检测到 **NACK** 的时候读写标志位 **STAT_RW** 被硬件清零了, 如虚线 **g** 所示。

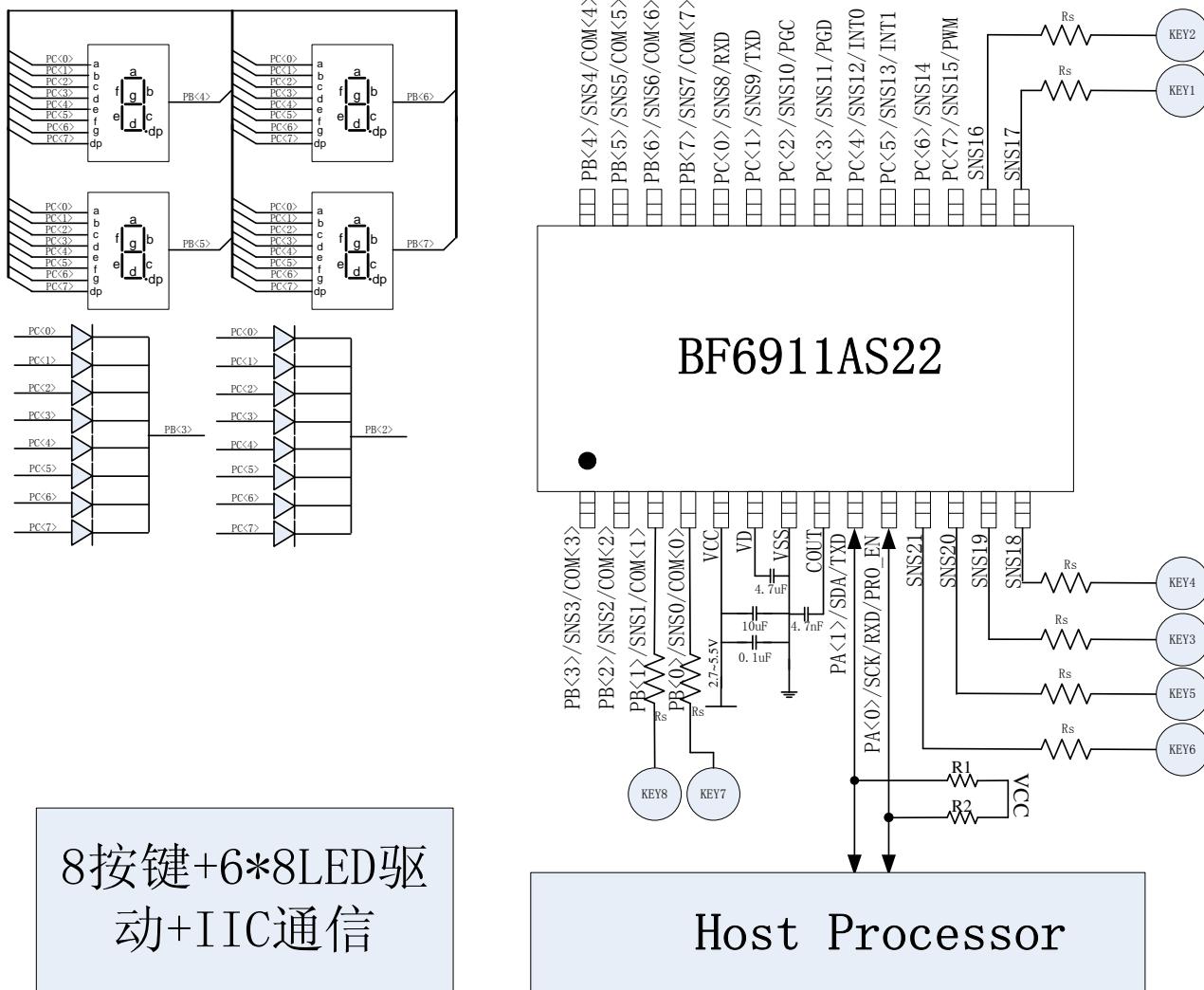
如果主机发送 **NACK** 的时候, 从机 **SCLEN** 是不会被自动拉低的, 这点在应用中要注意。

检测到 **STAT_P** 信号的时候, 状态位 **STAT_P** 置位, **STAT_S** 清零, 如虚线 **h** 所示。



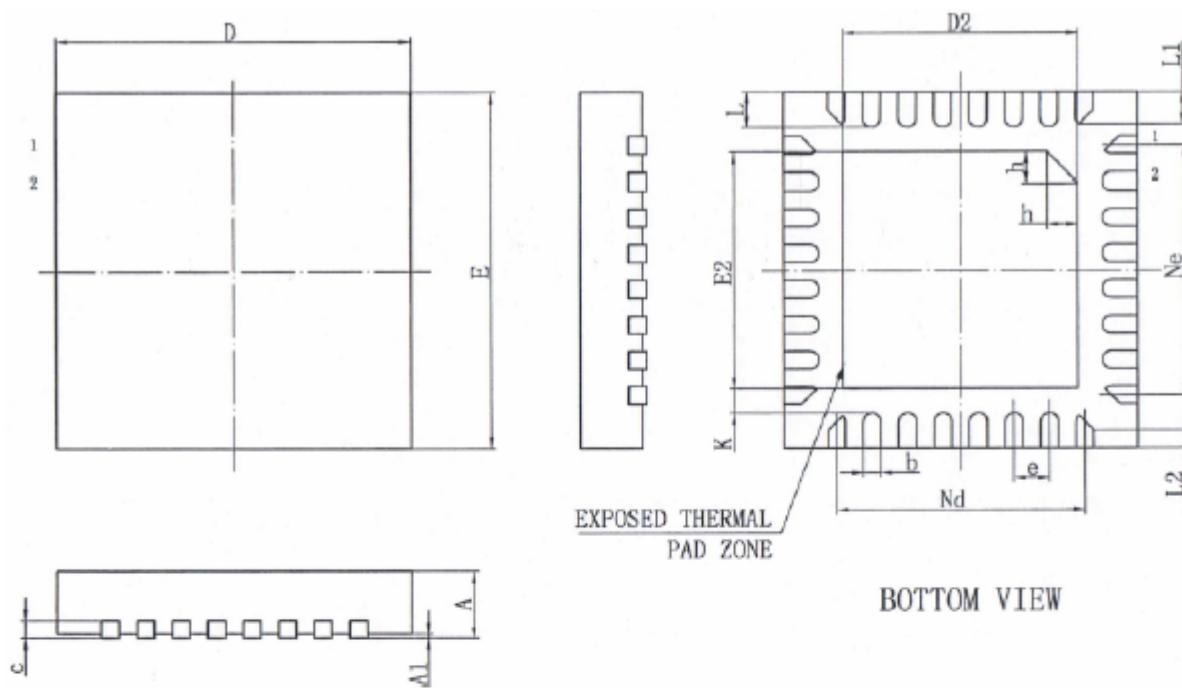
15、应用电路

15.1 BF6911AS22(SOP28) IIC 应用



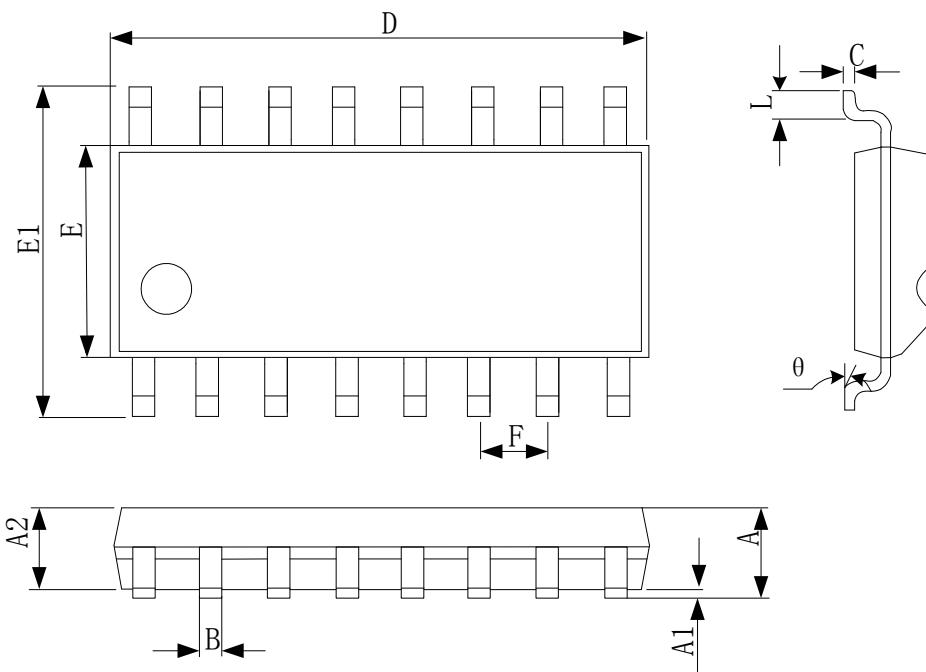
16、封装

QFN32



DIM	MILLIMETERS		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	2.60	2.65	2.70
e	0.40BSC		
Nd	2.80BSC		
E	3.90	4.00	4.10
E2	2.60	2.65	2.70
Ne	2.80BSC		
K	0.20	-	-
L	0.35	0.40	0.45
L1	0.30	0.35	0.40
L2	0.20REF		
H	0.30	0.35	0.40
L/F载体尺寸 (Mil)	112*122		

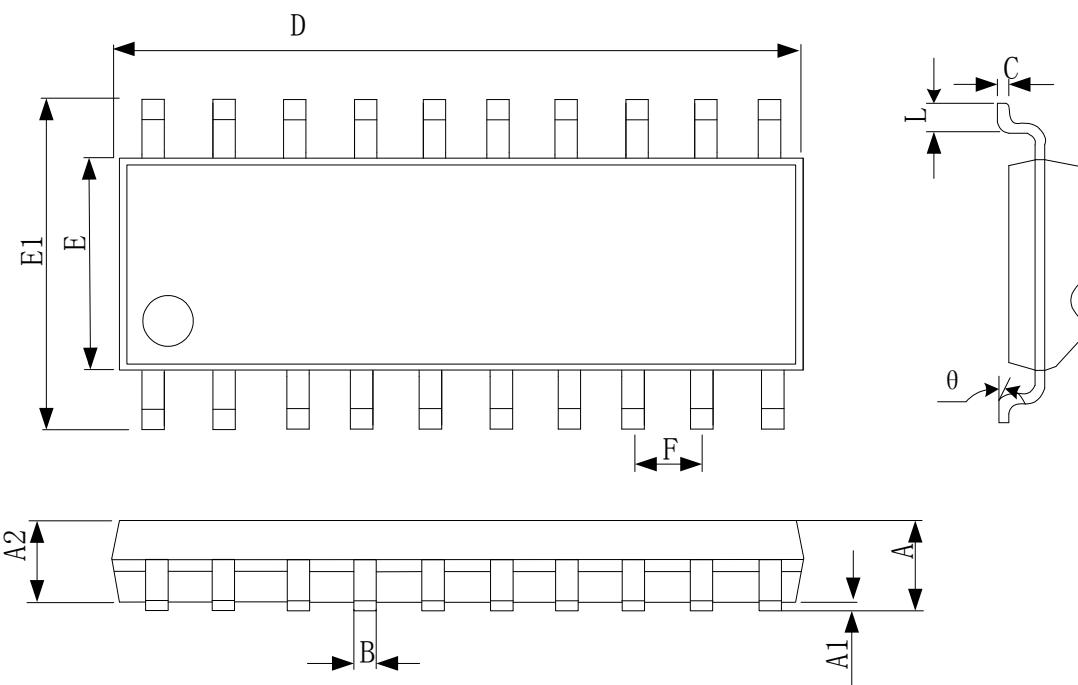
II—SOP16



封装尺寸:

DIM	MILLIMETERS	
	MIN	MAX
A	1.350	1.750
A1	0.100	0.250
A2	1.350	1.550
B	0.330	0.510
C	0.170	0.250
D	9.800	10.200
E	3.800	4.000
F	1.270BSC	
L	0.4000	1.270
θ	0°	8°

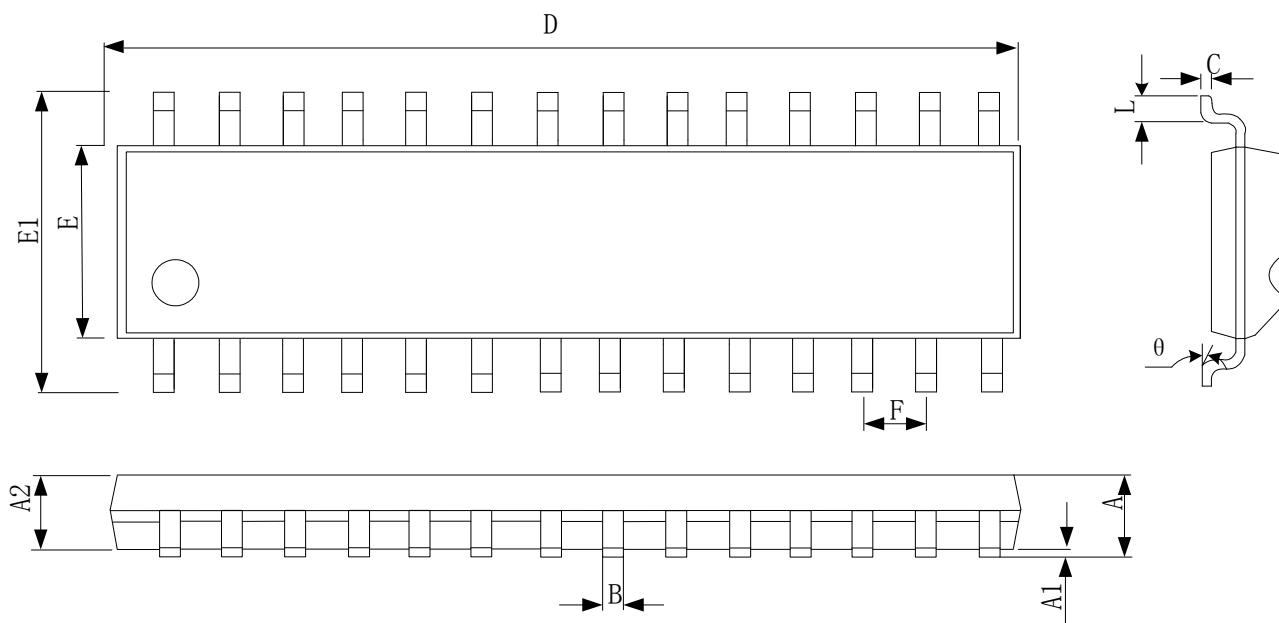
III—SOP20



封装尺寸:

DIM	MILLMETERS		
	MIN	NOM	MAX
A	-	-	2.70
A1	0.10	0.20	0.30
A2	2.10	2.30	2.50
B	0.39	-	0.48
C	0.25	-	0.31
D	12.45	12.65	12.85
E	7.4	7.6	7.8
E1	10.10	10.30	10.50
F	1.27BSC		
L	0.70	0.85	1.00
θ	0°	-	8°

IV—SOP28



封装尺寸:

DIM	MILLIMETERS		
	MIN	NOM	MAX
A	-	-	2.70
A1	0.10	0.20	0.30
A2	2.10	2.30	2.50
B	0.39	-	0.48
C	0.25	-	0.31
D	17.55	17.75	17.95
E	7.4	7.6	7.8
E1	10.10	10.30	10.50
F	1.27BSC		
L	0.70	0.85	1.00
θ	0°	-	8°