

FT64F0AX

IIC Application note

目录

1. I2C 接口	3
1.1. I2C 接口相关寄存器汇总	4
1.2. I2C 的工作原理	10
1.2.1. 主机发送	10
1.2.2. 主机接收	11
1.2.3. 从机发送	12
1.2.4. 从机接收	13
1.2.5. 广播呼叫 (General Call)	13
2. 应用范例	14
联系信息	23

FT64F0Ax IIC 应用

1. I2C 接口

I2C 为双线接口 (数据线 SDA 和串行时钟线 SCL), 可通过 I2C 协议与外部设备进行通信, 特性如下:

- 主机模式和从机模式
- 多主机支持
- 标准模式 (100kHz) 和快速模式 (400kHz)
- 7 位和 10 位地址模式
- General call 支持
- Clock stretching
- 发送 NACK (从机模式)

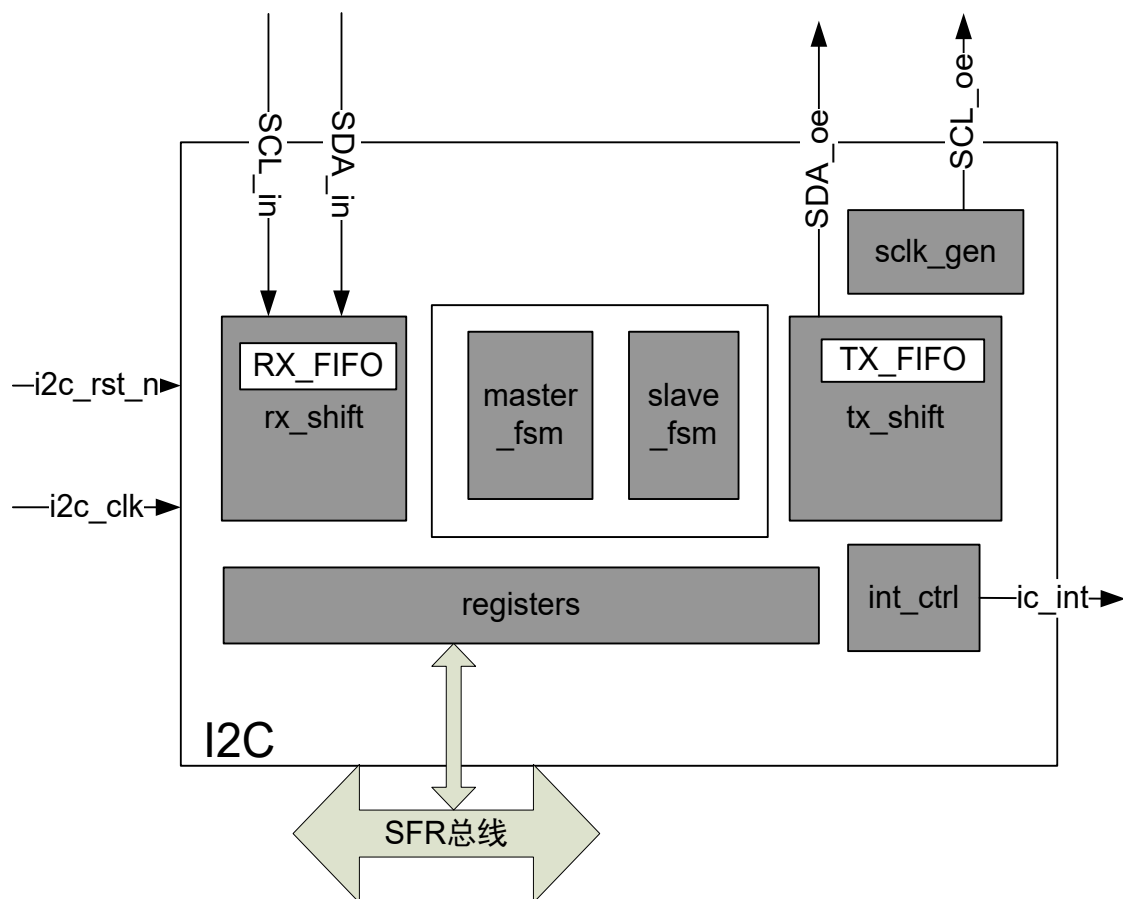


图 1-1 I2C 结构框图

1.1. I2C 接口相关寄存器汇总

名称	状态	寄存器	地址	复位值
MST10B ¹	<u>主机发送地址格式</u> 1 = 10 位 0 = <u>7 位</u>	I2CCR1[4]	0x40C	RW-0
SLV10B ¹	<u>从机响应地址格式</u> 1 = 10 位 0 = <u>7 位</u>	I2CCR1[3]		RW-0
SPEED ¹	<u>I2C 通信速度</u> 1 = 快速模式 (400kHz) 0 = <u>标准模式 (100kHz)</u>	I2CCR1[1]		RW-0
MASTER ¹	<u>工作模式</u> 1 = 主机模式 0 = <u>从机模式</u>	I2CCR1[0]		RW-0
SOFTTRST	<u>软件复位</u> (当 ACTIVE = 1 时可写) 1 = 复位 I2C 模块 0 = <u>无意义</u>	I2CCR2[6]	0x40D	RW-0
AGCALL ¹	<u>广播呼叫 (General call)</u> 主机模式: 1 = 发送 General call 地址 (0x00) 0 = <u>发送正常的从机地址</u> 从机模式: 1 = 响应 General call 0 = <u>不响应 General call</u>	I2CCR2[5]		RW-0
SNACK ¹	<u>接收应答</u> 1 = 发送 NACK 0 = <u>发送 ACK (地址匹配或接收到数据)</u>	I2CCR2[4]		RW-0
RXHLD ¹	<u>时钟拉伸 (当 RX-FIFO 非空时)</u> 1 = 使能 (拉低 SCL) 0 = <u>禁止 (新接收的数据将会丢失)</u>	I2CCR2[1]		RW-0
EVSTRE	<u>时钟拉伸 (当 SBF / ADDF / ADD10F 置位后)</u> 1 = 使能 (拉低 SCL) 0 = <u>禁止</u>	I2CCR3[2]	0x40E	RW-0
ENABLE	<u>I2C 接口</u> 1 = 使能 0 = <u>关闭</u>	I2CCR3[0]		RW-0

¹ 当 ENABLE = 0 时可写。

名称	状态	寄存器	地址	复位值
ADD[7:0] ²	<u>从机地址低有效位 (LSB)</u> 7 位地址: ADD[6:0]有效, ADD[7]忽略; 10 位地址: ADD[7:0] = 低 8 位; 注: 主机模式下为目标从机地址, 从机模式下为本机地址;	I2COARL[7:0]	0x40F	RW-0000 0000
ADD[9:8] ²	<u>从机地址高有效位 (MSB)</u> 7 位地址: ADD[9:8]忽略; 10 位地址: ADD[9:8] = 高 2 位; 注: 主机模式下为目标从机地址, 从机模式下为本机地址;	I2COARH[1:0]	0x410	RW-00
I2CEN	<u>I2C 模块时钟</u> 1 = 打开 0 = 关闭	PCKEN[6]	0x09A	RW-0
SYSON	<u>睡眠模式下, 系统时钟控制</u> 1 = 保持运行 0 = 关闭	CKOCON[7]	0x095	RW-0
FREQ[5:0] ²	<u>I2C 外设时钟频率 Fmaster</u> 000000 = 禁止 000001 = 1MHz 000010 = 2MHz 011000 = 24MHz > 011000 = 禁止 注: Fmaster 必须与 SysClk 相同	I2CFRWQ[5:0]	0x411	RW-0000 00
DUTY ²	<u>快速模式下, 占空比设置</u> 1 = SCLL / SCLH = 16 / 9 0 = SCLL / SCLH = 2 / 1 注: 标准模式下, SCLL / SCLH = 1 / 1	I2CCCRH[6]	0x415	RW-0
CCR[7:0] ²	主机模式下, SCL 时钟周期低 8 位	I2CCCRL[7:0]	0x414	RW-0000 0000
CCR[11:8] ²	主机模式下, SCL 时钟周期高 4 位 SCL 时钟周期公式:			
	模式	周期	SCLL	SCLH
	标准模式	2*CCR*Fmaster	CCR*Fmaster	CCR*Fmaster
	快速模式 (DUTY=0)	3*CCR*Fmaster	2*CCR*Fmaster	CCR*Fmaster
	快速模式 (DUTY=1)	25*CCR*Fmaster	16*CCR*Fmaster	9*CCR*Fmaster
		I2CCCRH[3:0]	0x415	RW-0000

² 当 ENABLE = 0 时可写。

名称	状态	寄存器	地址	复位值
DR[7:0]	<u>数据寄存器</u> 写时：将新数据写入到 TX-FIFO 中 读时：返回 RX-FIFO 中未读的数据 注： TX-FIFO 和 RX-FIFO 的深度均为 1 写数据时，需先写 DR，再写 I2CCMD	I2CDR[7:0]	0x412	RW-0000 0000
RESTART	<u>主机发送 Start / Restart</u> 1 = 发送 0 = <u>不发送</u>	I2CCMD[2]	0x413	WO-0
STOP	<u>主机发送 Stop (字节传输后，或主机拉伸 SCL 时)</u> 1 = 发送 (发送成功后自动清零) 0 = <u>不发送</u>	I2CCMD[1]		WO-0
MSTDIR	<u>主机模式，数据传输方向 (读写位 R/W)</u> 1 = 读取 0 = <u>发送</u>	I2CCMD[0]		WO-0
GCALL	<u>从机模式接收到 General call 标志</u> 1 = Yes (接收且 ACK 后置位) 0 = <u>No</u> 注：检测到 Start/Stop 或 ENABLE = 0 时硬件自动清零；	I2CSR3[5]	0x419	RO-0
RDREQ	<u>从机模式，数据传输方向标志</u> 1 = 发送 (从机接收地址字节的读写位为 1 时置位) 0 = 接收 注：检测到 Start/Stop 或 ENABLE = 0 时硬件自动清零；	I2CSR3[2]		RO-0
ACTIVE	<u>主/从机状态</u> 1 = Busy (繁忙) 0 = <u>IDLE (空闲)</u> 注：从机模式，地址匹配成功后即置位，接收到 Start / Restart / Stop 后清零；	I2CSR3[1]		RO-0
RXHOLD	<u>RX-FIFO 非空保持标志</u> 1 = 非空 (SCL 被拉低，读 DR 后释放) 0 = <u>空 (SCL 未被拉低)</u>	I2CSR3[0]		RO-0

表 1-1 I2C 相关寄存器

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (PEIE, ITBUFEN, ITEVEN, ITERREN 适用) 0 = <u>全局关闭</u> (唤醒不受影响)	INTCON[7]	Bank 首 地 址 +0x0B	RW-0
PEIE	外设总中断	1 = 使能 (ITBUFEN, ITEVEN, ITERREN 适用) 0 = <u>关闭</u> (无唤醒)	INTCON[6]		RW-0
ITBUFEN	FIFO 状态中断	1 = 使能 (当 IICTXE = 1 或 IICRXNE = 1 时 产生中断) 0 = <u>关闭</u> (无唤醒)	I2CITR[2]	0x416	RW-0
IICTXE ³	TX-FIFO 状态	1 = 空 0 = <u>非空</u>	I2CSR1[7]	0x417	RO-0
IICRXNE ³	RX-FIFO 状态	1 = 非空 0 = <u>空</u>	I2CSR1[6]		RO-0
ITEVEN	事件中断	1 = 使能 0 = <u>关闭</u> (无唤醒) <u>事件中断产生条件:</u> SBF = 1 (主机) ADD10F = 1 (主机) ADDRF = 1 (主/从机) STOPF = 1 (从机)	I2CITR[1]	0x416	RW-0
STOPF ⁴	主/从机模式, Stop 标志	1 = Yes 0 = <u>No</u>	I2CSR1[4]	0x417	RO-0
ADD10F ⁴	主机模式, 目标从机 地址高有效位 MSB 匹配标志	1 = <u>匹配</u> (ACK 后置位) 0 = <u>不匹配, 或未发送地址</u>	I2CSR1[3]		RO-0
ADDRF ⁴	主机模式, 目标从机 地址低有效位 LSB 匹配标志	1 = <u>匹配</u> (ACK 后置位) 0 = <u>不匹配, 或未发送地址</u>	I2CSR1[1]		RO-0
	从机模式, 本机地址 匹配标志	1 = <u>匹配或识别到 General Call</u> 0 = <u>不匹配</u>			
SBF ⁴	主机发送 Start 标志	1 = 已发送 0 = <u>未发送</u>	I2CSR1[0]		RO-0

³ 写 DR 或 ENABLE = 0 时硬件自动清零。

⁴ 读 I2CSR1 或 ENABLE = 0 时硬件自动清零。

名称	状态	寄存器	地址	复位值
ITERREN	错误中断 1 = 使能 0 = 关闭 (无唤醒) <u>错误中断产生条件:</u> OVR = 1 AF = 1 ARLO = 1 BERR = 1	I2CITR[0]	0x416	RW-0
TXARBT ⁵	传输终止标志 (发送过程中出错或异常原因导致) 1 = 发生终止 0 = <u>未发生终止</u>	I2CSR2[4]	0x418	RW0-0
OVR ⁵	Overrun 产生标志 1 = Yes 0 = <u>No</u> <u>Overrun 产生条件:</u> TX-over: 当 TX-FIFO 非空时仍写 DR; RX-over: 当 RX-FIFO 非空时仍接收数据; RX-under: 当 RX-FIFO 空时进行读操作;	I2CSR2[3]		RW0-0
AF ⁵	主/从机模式, 接收应答状态 1 = NACK 0 = <u>ACK</u>	I2CSR2[2]		RW0-0
ARLO ⁵	主机模式, 总线仲裁失败标志 1 = 仲裁失败 0 = <u>未发生仲裁失败</u>	I2CSR2[1]		RW0-0
BERR ⁵	总线错误 (检测到错位的 Start / Stop) 标志 1 = 检测到 (字节传输阶段检测到 Start/Stop 时置位) 0 = 未检测到	I2CSR2[0]		RW0-0

表 1-2 I2C 中断使能和状态位

名称	状态	寄存器	地址	复位值
AFP0[0]	<u>I2C SDA 引脚</u> 1 = PB6 0 = <u>PB4</u>	AFP0[0]	0x19E	RW-0
AFP0[5]	<u>I2C SCL 引脚</u> 1 = PA2 0 = <u>PB3</u>	AFP0[5]	0x19E	RW-0
I2COD	<u>I2C SCL, I2C SDA 引脚开漏输出设置</u> 1 = 使能 0 = <u>关闭</u>	ODCON0[1]	0x21F	RW-0

表 1-3 I2C 接口引脚控制

⁵ 写 0 清零, 或 ENABLE = 0 时硬件自动清零。

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
PCKEN	0x9A	UART2EN	I2CEN	UART1EN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
I2CCR1	0x40C	—	—	—	MST10B	SLV10B	—	SPEED	MASTER	---0 0-00
I2CCR2	0x40D	—	SOFTIRST	AGCALL	SNACK	—	—	RXHLD	—	-000 ---0-
I2CCR3	0x40E	—					EVSTRE	—	ENABLE	---- -0-0
I2COARL	0x40F	ADD[7:0]								0000 0000
I2COARH	0x410	—	—	—	—	—	—	ADD[9:8]		---- --00
I2CFREQ	0x411	—	—	FREQ[5:0]						--00 0000
I2CDR	0x412	DR[7:0]								0000 0000
I2CCMD	0x413	—	—	—	—	—	RESTART	STOP	MSTDIR	---- -000
I2CCCRL	0x414	CCR[7:0]								0000 0000
I2CCCRH	0x415	—	DUTY	—	—	CCR[11:8]				-0—0000
I2CITR	0x416	—					ITBUFEN	ITEVEN	ITERREN	---- -000
I2CSR1	0x417	IICTXE	IICRXNE	—	STOPF	ADD10F	—	ADDRF	SBF	00-0 0-00
I2CSR2	0x418	—	—	—	TXABRT	OVR	AF	ARLO	BERR	---0 0000
I2CSR3	0x419	—	—	GCALL	—	—	RDREQ	ACTIVE	RXHOLD	--0- -000

表 1-4 I2C 相关寄存器地址

1.2. I2C 的工作原理

I2C 模块主要有四种工作模式，即主机接收、主机发送、从机发送、从机接收。每种模式下又包含了 7 位地址模式和 10 位地址格式。

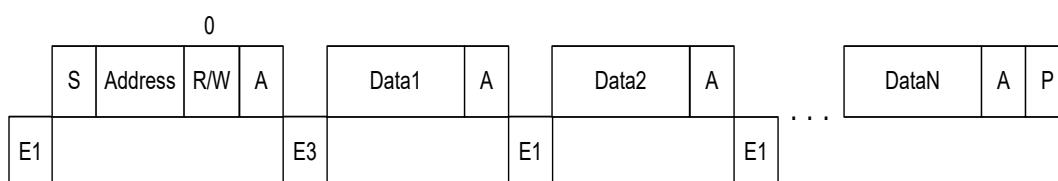
1.2.1. 主机发送

主机发送模式下，输出时钟到 SCL，发送串行数据到 SDA。

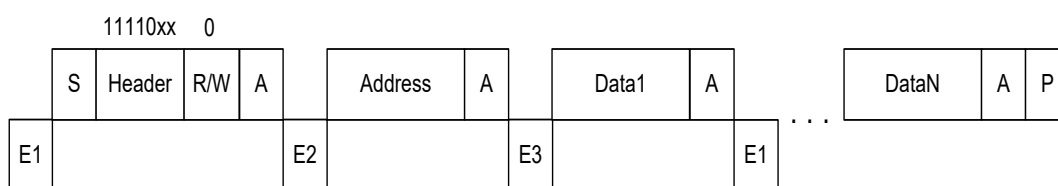
7 位地址(MST10B = 0)，主机发送的第 1 个 byte 包括地址和读写位(0)，然后开始发送 8 位串行数据。

10 位地址(MST10B = 1)，主机发送的第 1 个 byte 包括地址头段序列和读写位(0)，第 2 个 byte 为低有效位地址，然后开始发送 8 位串行数据。

7bit 地址：



10bit 地址：



注：xx为高有效位

图 1-2 主机发送流程

注：

S: Start 信号；

A: ACK 信号；

P: Stop 信号；

E1: IICTXE=1, TX-FIFO 为空, 写 DR 和 I2CCMD 将清零该标志；

E2: ADD10F=1, 读 I2CSR1 将清零该标志；

E3: ADDRDF=1, 读 I2CSR1 将清零该标志；

1.2.2. 主机接收

主机接收模式下，输出时钟到 SCL，从 SDA 线上接收串行数据。

7 位地址(MST10B = 0)，主机发送的第 1 个 byte 包括地址和读写位(1)，然后开始接收 8 位串行数据。

10 位地址(MST10B = 1)，主机发送的第 1 个 byte 包括地址头段序列和读写位(0)，第 2 个 byte 为低有效位地址，然后重新发送 Start 信号和地址头段序列和读写位(1)，开始接收 8 位串行数据。

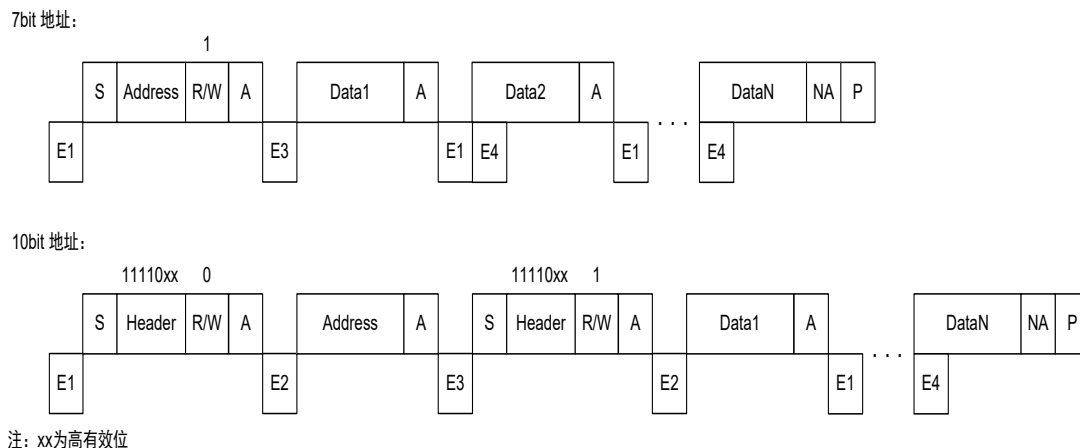


图 1-3 主机接收流程

注:

S: Start 信号;

A: ACK 信号;

P: Stop 信号;

E1: IIC TXE=1, TX-FIFO 为空, 写 DR 和 I2CCMD 将清零该标志;

E2: ADD10F=1, 读 I2CSR1 将清零该标志;

E3: ADDRDF=1, 读 I2CSR1 将清零该标志;

E4: IIC RXNE=1, RX-FIFO 非空, 读 DR 将清零该标志;

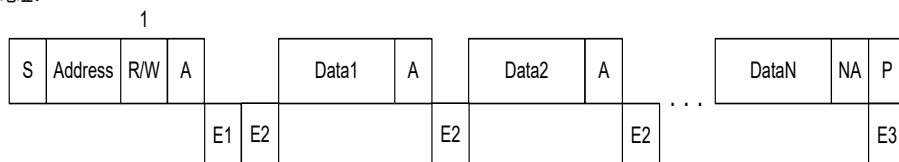
1.2.3. 从机发送

从机发送模式下，发送串行数据到 SDA。

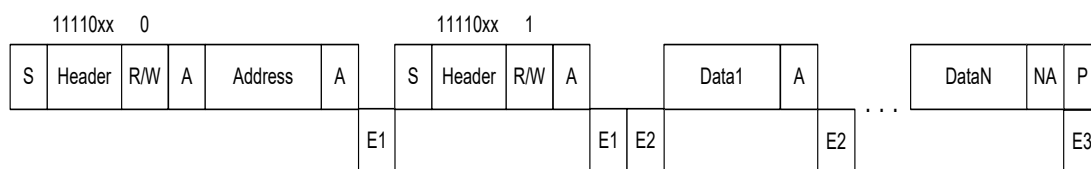
7 位地址(SLV10B = 0)，从机接收的第 1 个 byte 包括地址和读写位(1)，然后开始发送 8 位串行数据。

10 位地址(SLV10B = 1)，从机接收的第 1 个 byte 包括地址头段序列和读写位(0)，第 2 个 byte 为低有效位地址，然后重新检测 Start 信号并接收地址头段序列和读写位(1)，开始发送 8 位串行数据。

7bit 地址:



10bit 地址:



注: xx为高有效位

图 1-4 从机发送流程

注:

S: Start 信号;

A: ACK 信号;

P: Stop 信号;

E1: ADDRFR=1, 拉低 SCL 线, 读 I2CSR1 将清零该标志;

E2: IICTXE=1, TX-FIFO 为空, 拉低 SCL 线, 读 RDREQ 为 1, 写 DR 和 I2CCMD 将清零该标志;

E3: AF=1, 写 0 清零;

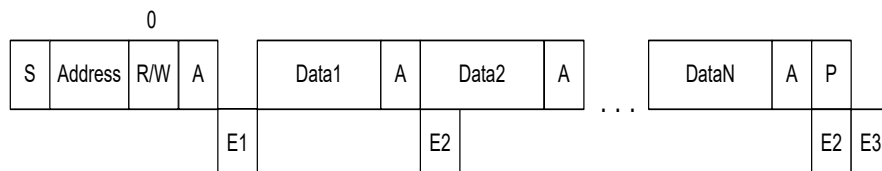
1.2.4. 从机接收

从机接收模式下，从 SDA 线上接收串行数据。

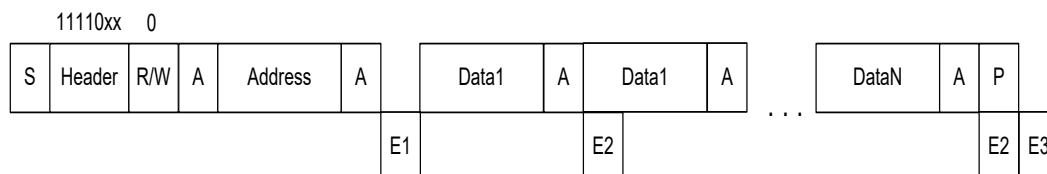
7 位地址(SLV10B = 0)，从机接收的第 1 个 byte 包括地址和读写位(0)，然后开始接收 8 位串行数据。

10 位地址(SLV10B = 1)，从机接收的第 1 个 byte 包括地址头段序列和读写位(0)，第 2 个 byte 为低有效位地址，然后开始接收 8 位串行数据。

7bit 地址:



10bit 地址:



注: xx为高有效位

图 1-5 从机接收流程

注:

S: Start 信号;

A: ACK 信号;

P: Stop 信号;

E1: ADDRFR=1, 读 I2CSR1 将清零该标志;

E2: IICRXNE=1, RX-FIFO 非空, 读 DR 将清零该标志;

E3: STOPF =1, 读 I2CSR1 将清零该标志;

1.2.5. 广播呼叫 (General Call)

General Call 模式在主机置位了 AGCALL 以后, 就会向地址为 0x00 的地址发送数据, 这种模式下主机只允许进行写数据, 不允许读数据; 从机模式下在置位了 AGCALL 以后, 就会响应主机发来的 General Call; 通信的过程跟主机发送, 从机接收模式相同。

2. 应用范例

```

=====
/* 文件名: ASM_64F0Ax_IIC.ASM
* 功能:    FT64F0Ax_IIC 功能演示
* IC:      FT64F0A5    TSSOP20
* 内部:    16M/2T
* 说明:    此演示程序位 64F0Ax_IIC 的演示程序.
*          该程序把 0x55 写入(24C02)0x12 地址,后读 0x12 地址的值, 判断是否写入成功
*
*          FT64F0A5  TSSOP20
*          -----
* NC-----|1(PA5)      (PA4)20|-----NC
* NC-----|2(PA6)      (PA3)19|-----NC
* NC-----|3(PA7)      (PA2)18|-----NC
* NC-----|4(PC0)      (PA1)17|-----NC
* NC-----|5(PC1)      (PA0)16|-----NC
* NC-----|6(PB7)      (PB0)15|-----NC
* GND-----|7(GND)     (PB1)14|-----NC
* NC-----|8(PB6)      (PB2)13|-----NC
* VDD-----|9(VDD)     (PB3)12|----IIC_SCL
* NC-----|10(PB5)     (PB4)11|----IIC_SDA
*
*          -----
*/
=====
#include <FT64F0AX.INC>;
=====
;RAM DEFINE
=====
TEMP            EQU        0X40
TEMP1           EQU        0X41
TEMP2           EQU        0X42
IICDATA         EQU        0X43
IICADDR         EQU        0X44
IICDATA2        EQU        0X45
W_TMP           EQU        0X70
S_TMP           EQU        0X71
=====
;CONSTANT DEFINE
=====
INTCON_DEF      EQU        B'00000000' ;禁止所有中断
OSCCON_DEF      EQU        B'01110001' ;16MHz,1:1

WPUA_DEF        EQU        B'00000000' ;弱上拉的开关, 0-关, 1-开
WPUB_DEF        EQU        B'00011000'

```

```

WPUC_DEF      EQU      B'00000000'

WPDA_DEF      EQU      B'00000000' ;弱下拉的开关, 0-关, 1-开
WPDB_DEF      EQU      B'00000000'
WPDC_DEF      EQU      B'00000000'

TRISA_DEF     EQU      B'00000000' ;输入输出设置, 0-输出, 1-输入
TRISB_DEF     EQU      B'00000000' ;PB3-OUT
TRISC_DEF     EQU      B'00000000'

PSRC0_DEF     EQU      B'11111111' ;源电流设置最大
PSRC1_DEF     EQU      B'11111111'
PSRC2_DEF     EQU      B'00001111'

PSINK0_DEF    EQU      B'11111111' ;灌电流设置最大
PSINK1_DEF    EQU      B'11111111'
PSINK2_DEF    EQU      B'00000011'

ANSELA_DEF    EQU      B'00000000' ;设置对应的 IO 为数字 IO

PCKEN_DEF     EQU      B'01000000' ;使能 I2C 模块时钟

ODCON0_DEF    EQU      B'00000010'
;I2C_SCL,I2C_SDA 的开漏输出设置, 高有效

I2CCR1_DEF    EQU      B'00000001'
;主机模式, 标准模式 (100kHz), 7 位地址格式
I2CCR2_DEF    EQU      B'00000000'
I2CCR3_DEF    EQU      B'00000000'
;禁用 I2C 模块, 才能写目标从机地址, 设置外设时钟频率以及 SCL 时钟周期

I2COARL_DEF   EQU      B'01010000' ;从机地址
I2COARH_DEF   EQU      B'00000000'

I2CFREQ_DEF   EQU      B'00010000' ;外设时钟频率为 16MHz

I2CCCRL_DEF   EQU      B'10000000'
;标准模式下, SCL 时钟周期为 2*CCR*Fmaster (Fmaster 为 2*128*16M)
I2CCCRH_DEF   EQU      B'00000000'

I2CITR_DEF    EQU      B'00000000' ;不使能 IIC 中断
;=====
;USER DEFINE
;=====

```

```
#DEFINE      IIC_SCL      PORTB,3
#DEFINE      IIC_SDA      PORTB,4
;=====
;PROGRAM START
;=====
    ORG        0x0000
    LJUMP      RESTART
    ORG        0x0004
    STR        W_TMP
    SWAPR      STATUS,W
    STR        S_TMP
    LJUMP      INT_PROGRAM
;=====
;SYSTEM  START
;=====
RESTART:
    BANKSEL    PORTA
    LCALL      INITIAL
    LCALL      IIC_INITIAL

    BANKSEL    IICADDR
    LDWI       0X12          ;要写的地址
    STR        IICADDR
    LDWI       0X55          ;要写的地址
    STR        IICDATA
    LCALL      IIC_WRITE

    LCALL      DELAY_32MS

    BANKSEL    IICDATA
    CLRF       IICDATA
    BANKSEL    IICDATA
    LDWI       0X12          ;要写的地址
    STR        IICADDR
    LCALL      IIC_READ

    BANKSEL    IICDATA
    STR        IICDATA          ;读到的数值存放在 ICDATA

MAIN:
    NOP
    LJUMP      MAIN
;=====
;INT_PROGRAM
```



```

;=====
INT_PROGRAM:
    SWAPR      S_TMP,0
    STR        STATUS
    SWAPR      W_TMP,1
    SWAPR      W_TMP,0
    RETI

;=====
;SYSTEM    INITIAL
;=====
INITIAL:
    BANKSEL    OSCCON
    LDWI        OSCCON_DEF
    STR        OSCCON

    BANKSEL    INTCON
    LDWI        INTCON_DEF
    STR        INTCON

    BANKSEL    PORTA
    LDWI        0X00
    STR        PORTA
    STR        PORTB
    STR        PORTC

    BANKSEL    TRISA
    LDWI        TRISA_DEF
    STR        TRISA
    LDWI        TRISB_DEF
    STR        TRISB
    LDWI        TRISC_DEF
    STR        TRISC

    BANKSEL    WPUA
    LDWI        WPUA_DEF
    STR        WPUA
    LDWI        WPUB_DEF
    STR        WPUB
    LDWI        WPUC_DEF
    STR        WPUC

    BANKSEL    WPDA
    LDWI        WPDA_DEF
    STR        WPDA

```

```
LDWI    WPDB_DEF
STR      WPDB
LDWI    WPDC_DEF
STR      WPDC
```

```
BANKSEL  PSRC0
LDWI     PSRC0_DEF
STR      PSRC0
LDWI     PSRC1_DEF
STR      PSRC1
LDWI     PSRC2_DEF
STR      PSRC2
```

```
BANKSEL  PSINK0
LDWI     PSINK0_DEF
STR      PSINK0
LDWI     PSINK1_DEF
STR      PSINK1
LDWI     PSINK2_DEF
STR      PSINK2
```

```
BANKSEL  ANSELA
LDWI     ANSELA_DEF
STR      ANSELA
```

```
,*****Clear SRAM*****
```

```
BANKSEL  PORTA
LDWI     0X00
STR      FSR0H
```

```
CLEAR_RAM_BANK0:
```

```
LDWI     20H
STR      FSR0L
```

```
CLEAR_RAM_BANK0_LOOP:
```

```
CLRR     INDF0
INCR     FSR0L,F
LDWI     80H
XORWR    FSR0L,W
BTSS     STATUS,Z
LJUMP    CLEAR_RAM_BANK0_LOOP
```

```
CLEAR_RAM_BANK1:
```

```
LDWI     0A0H
STR      FSR0L
```

```
CLEAR_RAM_BANK1_LOOP:
```

```
CLRR     INDF0
INCR     FSR0L,F
```

```
LDWI      00H
XORWR     FSR0L,W
BTSS      STATUS,Z
LJUMP     CLEAR_RAM_BANK1_LOOP
INCR      FSR0H,F
```

CLEAR_RAM_LOOP:

```
LDWI      10
SUBWR     FSR0H,W
BTSS      STATUS,0
LJUMP     CLEAR_RAM_BANK0
RET
```

=====

;初始化 I2C

=====

IIC_INITIAL:

```
BANKSEL   PCKEN
LDWI      PCKEN_DEF
STR        PCKEN
```

```
BANKSEL   ODCON0
LDWI      ODCON0_DEF
STR        ODCON0
```

```
BANKSEL   I2CCR1
LDWI      I2CCR1_DEF
STR        I2CCR1
LDWI      I2CCR2_DEF
STR        I2CCR2
LDWI      I2CCR3_DEF
STR        I2CCR3
```

```
BANKSEL   I2COARL
LDWI      I2COARL_DEF
STR        I2COARL
LDWI      I2COARH_DEF
STR        I2COARH
LDWI      I2CFREQ_DEF
STR        I2CFREQ
```

```
BANKSEL   I2CCCRL
LDWI      I2CCCRL_DEF
STR        I2CCCRL
LDWI      I2CCCRH_DEF
STR        I2CCCRH
```

```

LDWI      I2CITR_DEF
STR        I2CITR

BANKSEL    I2CCR3
BSR        I2CCR3,ENABLE    ;使能 I2C 模块

RET

;=====
;IIC 读出指定位置的数据
;=====
IIC_READ:
    BANKSEL    I2CSR1
    BTSC       I2CSR1,IICTXE
    LJUMP      READ_STEP1
    LJUMP      $-2
READ_STEP1:
    BANKSEL    IICADDR
    LDR        IICADDR,0
    BANKSEL    I2CDR
    STR        I2CDR
    BANKSEL    I2CCMD
    LDWI       B'00000110'
    STR        I2CCMD
READ_STEP2:
    BANKSEL    I2CSR1
    BTSC       I2CSR1,IICTXE
    LJUMP      READ_STEP3
    LJUMP      READ_STEP2
READ_STEP3:
    BANKSEL    I2CCMD
    LDWI       B'00000011'
    STR        I2CCMD
READ_STEP4:
    BANKSEL    I2CSR1
    BTSC       I2CSR1,IICRXNE
    LJUMP      READ_STEP5
    LJUMP      READ_STEP4
READ_STEP5:
    BANKSEL    I2CDR
    LDR        I2CDR,0
    RET

;=====
;IIC 把指定数据写入指定位置
;=====

```

IIC_WRITE:

```

    BANKSEL    I2CSR1
    BTSC       I2CSR1,IICTXE
    LJUMP      WRITE_STEP1
    LJUMP      $-2

```

WRITE_STEP1:

```

    BANKSEL    IICADDR
    LDR        IICADDR,0
    BANKSEL    I2CDR
    STR        I2CDR
    BANKSEL    I2CCMD
    LDWI       B'00000000'
    STR        I2CCMD

```

WRITE_STEP2:

```

    BANKSEL    I2CSR1
    BTSC       I2CSR1,IICTXE
    LJUMP      WRITE_STEP3
    LJUMP      WRITE_STEP2

```

WRITE_STEP3:

```

    BANKSEL    IICDATA
    LDR        IICDATA,0
    BANKSEL    I2CDR
    STR        I2CDR
    BANKSEL    I2CCMD
    LDWI       B'00000010'
    STR        I2CCMD

```

WRITE_STEP4:

```

    BANKSEL    I2CSR1
    BTSC       I2CSR1,IICTXE
    LJUMP      WRITE_STEP4
    RET

```

```

;=====

```

```

;DELAY_32MS(16M/2T)

```

```

;=====

```

DELAY_32MS:

```

    LDWI       H'C8'
    STR        TEMP

```

DELAY_32MS_LOOP1:

```

    LDWI       H'50'
    STR        TEMP1

```

DELAY_32MS_LOOP2:

```

    LDWI       H'04'
    STR        TEMP2

```

DELAY_32MS_LOOP3:

```
CLRWDT
DECRSZ      TEMP2,F
LJUMP       DELAY_32MS_LOOP3
DECRSZ      TEMP1,F
LJUMP       DELAY_32MS_LOOP2
DECRSZ      TEMP,F
LJUMP       DELAY_32MS_LOOP1
RET

END
```

联系信息

Fremont Micro Devices (SZ) Corporation

#5-8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811

Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Corporation

#16, 16/F, Block B, Veristrong Industrial Centre,
34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186

Fax: (+852) 2781 1144

<http://www.fremontmicro.com/>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices (SZ) Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices (SZ) Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices (SZ) Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices (SZ) Corporation. The FMD logo is a registered trademark of Fremont Micro Devices (SZ) Corporation. All other names are the property of their respective owners.