

zlg7289A

串行接口 8 位 LED 数码管及 64 键键盘智能控制芯片

zlg7289A 是一片具有串行接口的，可同时驱动 8 位共阴式数码管（或 64 只独立 LED）的智能显示驱动芯片，该芯片同时还可连接多达 64 键的键盘矩阵，单片即可完成 LED 显示、键盘接口的全部功能。

zlg7289A 内部含有译码器，可直接接受 BCD 码或 16 进制码，并同时具有 2 种译码方式（参看后文），此外，还具有多种控制指令，如消隐、闪烁、左移、右移、段寻址等。

zlg7289A 具有片选信号，可方便地实现多于 8 位的显示或多于 64 键的键盘接口。

典型应用

仪器仪表，工业控制器，条形显示器，控制面板

特点

- 串行接口，无需外围元件可直接驱动 LED
- 各位独立控制译码/不译码及消隐和闪烁属性
- （循环）左移/（循环）右移指令
- 具有段寻址指令，方便控制独立 LED
- 64 键键盘控制器，内含去抖动电路

电特性（ $V_{CC}=5.0V$ ， $F_{osc}=16MHz$ ， $T_A=25^{\circ}C$ ）：

符号	参数	测试条件	最小	典型	最大	单位
V_{CC}	电源电压		4.5	5.0	5.5	V
I_{CC}	工作电流	不接 LED		3	5	mA
I_{CC}	工作电流	LED 全亮， $I_{SEG}=10mA$		60	100	mA
V_{IH}	逻辑输入高电平		2.0		5.5	V
V_{IL}	逻辑输入低电平		0		0.8	V
T_{KEY}	按键响应时间	含去抖动时间	10	18	40	mS
I_{KO}	KEY 引脚输出电流				7	mA
I_{KI}	KEY 引脚吸入电流				10	mA
T1	从 CS 下降沿至 CLK 脉冲时间		25	50	250	uS
T2	传送指令时 CLK 脉冲宽度		5	8	250	uS
T3	字节传送中 CLK 脉冲时间间隔		5	8	250	uS
T4	指令与数据时间间隔		15	25	250	uS
T5	读键盘指令中指令与输出数据时间间隔		15	25	250	uS
T6	输出键盘数据建立时间		5	8	—	uS
T7	读键盘数据时 CLK 脉冲宽度		5	8	250	uS
T8	读键盘数据完成后 DATA 转为输入状态时间				5	uS

引脚说明:

引脚	名称	说明
1, 2	V _{DD}	正电源
3, 5	NC	悬空
4	V _{SS}	接地
6	/CS	片选输入端, 此引脚为低电平时, 可向芯片发送指令及读取键盘数据
7	CLK	同步时钟输入端, 向芯片发送数据及读取键盘数据时, 此引脚电平上升沿表示数据有效
8	DATA	串行数据输入/输出端, 当芯片接收指令时, 此引脚为输入端; 当读取键盘数据时, 此引脚在‘读’指令最后一个时钟的下降沿变为输出端
9	/KEY	按键有效输出端, 平时为高电平, 当检测到有效按键时, 此引脚变为低电平
10-16	SG-SA	段 g—段 a 驱动输出
17	DP	小数点驱动输出
18-25	DIG0-DIG7	数字 0—数字 7 驱动输出
26	OSC2	振荡器输出端
27	OSC1	振荡器输入端
28	/RESET	复位端

控制指令

zlg7289A 的控制指令分为两大类——纯指令和带有数据的指令。

• 纯指令

1. 复位（清除）指令

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	0	1	0	0

当 zlg7289A 收到该指令后, 将所有的显示清除, 所有设置的字符消隐、闪烁等属性也被一起清除。执行该指令后, 芯片所处的状态与系统上电后所处的状态一样。

2. 测试指令

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	1	1	1	1	1

该指令使所有的 LED 全部点亮, 并处于闪烁状态, 主要用于测试。

3. 左移指令

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	0	0	0	1

使所有的显示自右向左（从第 1 位向第 8 位）移动一位（包括处于消隐状态的显示位），但对各位所设置的消隐及闪烁属性不变。移动后, 最右边一位为空（无显示）。例如, 原显示为

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

其中第 2 位 ‘2’ 和第 4 位 ‘4’ 为闪烁显示, 执行了左移指令后, 显示变为

2	3	4	5	6	7	8	
---	---	---	---	---	---	---	--

第二位 ‘3’ 和第四位 ‘5’ 为闪烁显示。

4. 右移指令

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	0	0	0	0

与左移指令类似，但所做移动为自左向右（从第 8 位向第 1 位）移动，移动后，最左边一位为空。

5. 循环左移指令

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	0	0	1	1

与左移指令类似，不同之处在于移动后原最左边一位（第 8 位）的内容显示于最右位（第 1 位）。在上例中，执行完循环左移指令后的显示为

2	3	4	5	6	7	8	1
---	---	---	---	---	---	---	---

第二位‘3’和第四位‘5’为闪烁显示。

6. 循环右移指令

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	0	0	1	0

与循环左移指令类似，但移动方向相反。

• 带有数据的指令

1. 下载数据且按方式 0 译码

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	a ₂	a ₁	a ₀

D7	D6	D5	D4	D3	D2	D1	D0
DP	X	X	X	d ₃	d ₂	d ₁	d ₀

X=无影响

命令由二个字节组成，前半部分为指令，其中 a₂，a₁，a₀ 为位地址，具体分配如下（显示位编号请参阅典型应用电路图）：

a ₂	a ₁	a ₀	显示位
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

d_0 — d_3 为数据，收到此指令时，zlg7289A 按以下规则（译码方式 0）进行译码，如下表：

d_0 — d_3 (十六进制)	d_3	d_2	d_1	d_0	7 段显示
00H	0	0	0	0	0
01H	0	0	0	1	1
02H	0	0	1	0	2
03H	0	0	1	1	3
04H	0	1	0	0	4
05H	0	1	0	1	5
06H	0	1	1	0	6
07H	0	1	1	1	7
08H	1	0	0	0	8
09H	1	0	0	1	9
0AH	1	0	1	0	-
0BH	1	0	1	1	E
0CH	1	1	0	0	H
0DH	1	1	0	1	L
0EH	1	1	1	0	P
0FH	1	1	1	1	空（无显示）

小数点的显示由 DP 位控制，DP=1 时，小数点显示，DP=0 时，小数点不显示。

2. 下载数据且按方式 1 译码

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	1	a_2	a_1	a_0

D7	D6	D5	D4	D3	D2	D1	D0
DP	X	X	X	d_3	d_2	d_1	d_0

X=无影响

此指令与上一条指令基本相同，所不同的是译码方式，该指令的译码按下表进行：

d_0 — d_3 (十六进制)	d_3	d_2	d_1	d_0	7 段显示
00H	0	0	0	0	0
01H	0	0	0	1	1
02H	0	0	1	0	2
03H	0	0	1	1	3
04H	0	1	0	0	4
05H	0	1	0	1	5
06H	0	1	1	0	6
07H	0	1	1	1	7
08H	1	0	0	0	8
09H	1	0	0	1	9
0AH	1	0	1	0	A
0BH	1	0	1	1	B
0CH	1	1	0	0	C
0DH	1	1	0	1	D
0EH	1	1	1	0	E
0FH	1	1	1	1	F

3. 下载数据但不译码

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	a_2	a_1	a_0

D7	D6	D5	D4	D3	D2	D1	D0
DP	A	B	C	D	E	F	G

其中， a_2 ， a_1 ， a_0 为位地址（参见‘下载数据且译码’指令），A-G 和 DP 为显示数据，分别对应 7 段 LED 数码管的各段。数码管各段的定义见下图。当相应的数据位为‘1’时，该段点亮，否则不亮。

4. 闪烁控制

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
d ₈	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁

此命令控制各个数码管的消隐属性。d₁—d₈ 分别对应数码管 1—8，0=闪烁，1=不闪烁。开机后，缺省的状态为各位均不闪烁。

5. 消隐控制

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
d ₈	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁

此命令控制各个数码管的消隐属性。d₁—d₈ 分别对应数码管 1—8，1=显示，0=消隐。当某一位被赋予了消隐属性后，zlg7289A 在扫描时将跳过该位，因此在这种情况下无论对该位写入何值，均不会被显示，但写入的值将被保留，在将该位重新设为显示状态后，最后一次写入的数据将被显示出来。当无需用到全部 8 个数码管显示的时候，将不用的位设为消隐属性，可以提高显示的亮度。

注意：至少应有一位保持显示状态，如果消隐控制指令中 d₁—d₈ 全部为 0，该指令将不被接受，zlg7289A 保持原来的消隐状态不变。

6. 段点亮指令

D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	0	0	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
X	X	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀

此为段寻址指令，作用为点亮数码管中某一指定的段，或 LED 矩阵中某一指定的 LED。指令中，X=无影响；d₀—d₅ 段地址，范围从 00H—3FH，具体分配为：

第 1 个数码管的 G 段地址为 00H，F 段为 01H，... ..A 段为 06H，小数点 DP 为 07H，第 2 个数码管的 G 段为 08H，F 段为 09H，... ..，依此类推直至第 8 个数码管的小数点 DP 地址为 3FH。

7. 段关闭指令

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
X	X	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀

段寻址命令，作用为关闭（熄灭）数码管中的某一段，指令结构与‘段点亮指令’相同，请参阅上文。

8. 读键盘数据指令

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	1	0	1

D7	D6	D5	D4	D3	D2	D1	D0
d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀

该指令从 zlg7289A 读出当前的按键代码。与其它指令不同，此命令的前一个字节 0001010B 为微控制器传送到 zlg7289A 的指令，而后一个字节 d₀—d₇ 则为 zlg7289A 返回的按键代码，其范围是 0—3FH（无键按下时为 0xFF），各键键盘代码的定义，请参阅图 2。

此指令的前半段，zlg7289A 的 DATA 引脚处于高阻输入状态，以接受来自微处理器的指令；在指令的后半段，DATA 引脚从输入状态转为输出状态，输出键盘代码的值。故微处理器连接到 DATA 引脚的 I/O 口应有一从输出态到输入态的转换过程，详情请参阅本文

‘串行接口’一节的内容。

当 zlg7289A 检测到有效的按键时，KEY 引脚从高电平变为低电平，并一直保持到按键结束。在此期间，如果 zlg7289A 接收到‘读键盘数据指令’，则输出当前按键的键盘代码；如果在收到‘读键盘指令’时没有有效按键，zlg7289A 将输出 FFH (11111111B)。

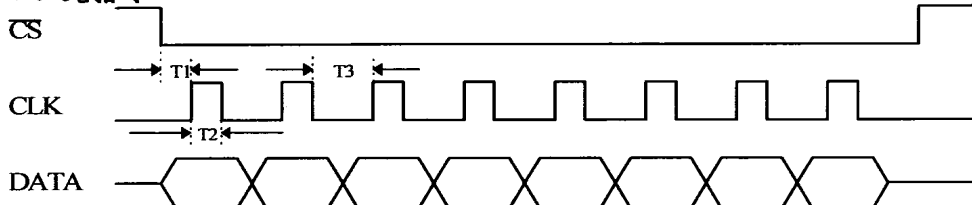
串行接口

zlg7289A 采用串行方式与微处理器通讯，串行数据从 DATA 引脚送入芯片，并由 CLK 端同步。当片选信号变为低电平后，DATA 引脚上的数据在 CLK 引脚的上升沿被写入 zlg7289A 的缓冲寄存器。

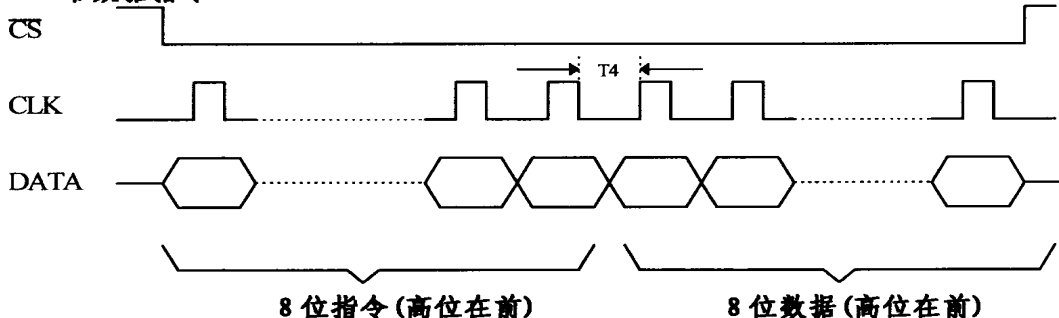
zlg7289A 的指令结构有三种类型：1.不带数据的纯指令，指令的宽度为 8 个 BIT，即微处理器需发送 8 个 CLK 脉冲。2.带有数据的指令，宽度为 16 个 BIT，即微处理器需发送 16 个 CLK 脉冲。3.读取键盘数据指令，宽度为 16 个 BIT，前 8 个为微处理器发送到 zlg7289A 的指令，后 8 个 BIT 为 zlg7289A 返回的键盘代码。执行此指令时，zlg7289A 的 DATA 端在第 9 个 CLK 脉冲的上升沿变为输出状态，并与第 16 个脉冲的下降沿恢复为输入状态，等待接收下一个指令。

串行接口的时序如下图：

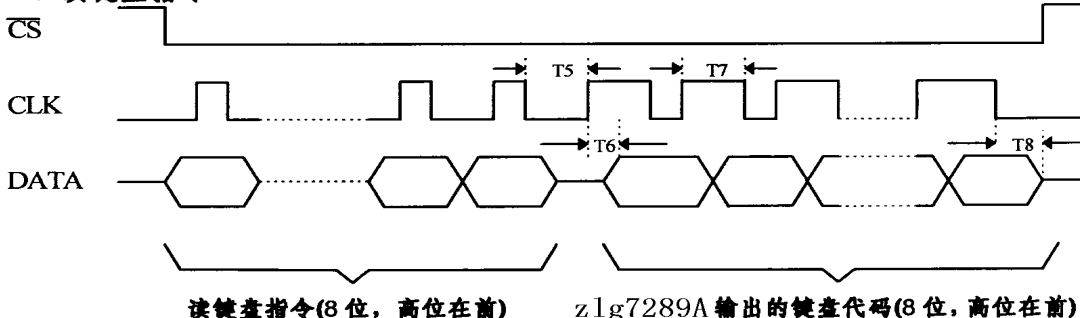
1、纯指令



2、带数据指令



3、读键盘指令



不适用于应用在需要 2 个或 2 个以上键同时按下的场合。

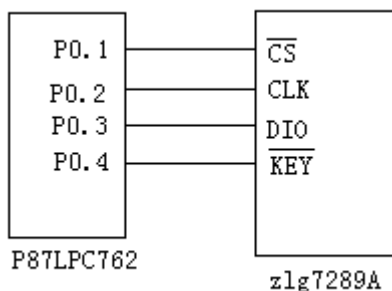
接口程序实例

下面给出 PHILIPS 公司的 P87LPC764 及 MICROCHIP 公司的 PIC16C54 与 zlg7289A 连接的应用实例。2 个程序所完成的功能相同，均为等待键盘输入，然后将所读到的键盘码转换成 10 进制后，送回 zlg7289A 显示，同时将前面的显示内容左移，并使当前按键值闪烁。

1. P87LPC764 接口程序

硬件连接如图，P87LPC764 所用时钟频率为 6MHz，程序编译通过并经过验证。程序中延时时间以 zlg7289A 外接 12MHz 晶体振荡器为准。

BIT_CNT	DATA	30H
DELAY1	DATA	31H
DECIMAL	DATA	32H
REC_BUF	DATA	20H
SEND_BUF	DATA	21H



CS	BIT	P0.1
CLK	BIT	P0.2
DIO	BIT	P0.3
KEY	BIT	P0.4

```

                ORG    00H
                AJMP   RESET
                ORG    80H

RESET:
    MOV        , #40H
    MOV        POM1, #00010000B           ; 设定 P1.4 为输入口, 其他为准双向口
    MOV        POM2, #00000000B
    MOV        WDCON, #00010101B         ; 设定看门狗溢出时间为 1 秒
    SETB      CS
    SETB      KEY
    SETB      DIO
    MOV        DELAY, #25                 ; 延时 25MS
RST_DELAY1:
    MOV        WDRST, #01EH
    
```



```

MOV    WDRST, #0E1H
DJNZ   DELAY1, RST_DELAY1
DJNZ   DELAY, RST_DELAY1
MOV    SEND_BUF, #10100100B      ;初始化命令
CALL   SEND
SETB   CS
MAIN_LP:
MOV    WDRST, #01EH              ;清除看门狗定时器
MOV    WDRST, #0E1H
JB     KEY, MAIN_LP
MOV    SEND_BUF, #00010101B      ;有键按下, 发送读键盘命令
CALL   SEND
CALL   RECEIVE                  ;读键盘
SETB   CS
MOV    B, #10                    ;10 进制转换
MOV    A, REC_BUF
DIV    AB
MOV    DECIMAL, A
MOV    SEND_BUF, #10100001B      ;左移 2 次.
CALL   SEND
SETB   CS
MOV    SEND_BUF, #10100001B
CALL   SEND
SETB   CS
MOV    SEND_BUF, #10000001B      ;下载数据且译码.
CALL   SEND
MOV    SEND_BUF, DECIMAL         ;发送 10 位数到 zlg7289A 显示
CALL   SEND
SETB   CS
MOV    SEND_BUF, #10000000B      ;下载数据且译码.
CALL   SEND
MOV    SEND_BUF, B               ;发送个位数据到 zlg7289A
CALL   SEND
SETB   CS
MOV    SEND_BUF, #10001000B      ;设定刚发送数据显示为闪烁
CALL   SEND
MOV    SEND_BUF, #11111100B
CALL   SEND
SETB   CS
MAIN_LP2:
JNB    KEY, MAIN_LP2            ;等待键松开.
AJMP   MAIN_LP

```

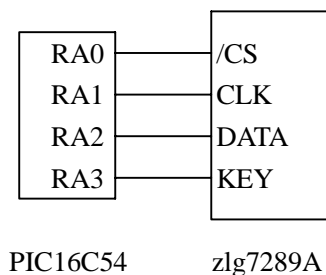
```
;-----  
;发送一字节到 zlg7289A, 高位在前.  
;-----  
SEND:  
    MOV    BIT_CNT, #8                ;设发送位数为 8 位.  
    CLR    CS  
    CALL   LONG_DELAY                ;延时 50 微秒.  
SEND_LP:  
    MOV    A, SEND_BUF  
    RLC    A  
    MOV    SEND_BUF, A  
    MOV    DIO, C  
    NOP  
    NOP  
    SETB   CLK  
    CALL   SHORT_DELAY                ;延时 10 微秒.  
    CLK    CLK  
    CALL   SHORT_DELAY                ;延时 10 微秒.  
    DJNZ   BIT_CNT, SEND_LP  
    CLR    DIO  
    RET  
;-----  
;由 zlg7289A 接收一字节数据, 高位在前.  
;-----  
RECEIVE:  
    MOV    BIT_CNT, #8                ;定义接收 8 位.  
    SETB   DAT  
    CALL   LONG_DELAY                ;延时 50 微秒.  
RECEIVE_LP:  
    SETB   CLK  
    CALL   SHORT_DELAY                ;延时 10 微秒.  
    MOV    C, DIO                    ;接收数据送 REC_BUF 低位.  
    MOV    A, REC_BUF  
    RLC    A  
    MOV    REC_BUF, A  
    CLR    CLK  
    CALL   SHORT_DELAY                ;延时 10 微秒.  
    DJNZ   BIT_CNT, RECEIVE_LP  
    CLR    DIO  
    RET  
LONG_DELAY:  
    MOV    DELAY, #25  
    DJNZ   DELAY, $  
    RET
```

SHORT_DELAY:

```
MOV    DELAY, #4
DJNZ   DELAY, $
RET
```

2. PIC16C54 接口程序

硬件连接如图，PIC16C54 所用时钟频率 4MHz。程序使用 MICROCHIP 公司的 MPASM 编译程序编译通过，并经过验证。程序中延时时间以 zlg7289A 外接 12MHz 晶体振荡器为准，如使用不同的 CPU 时钟频率或 zlg7289A 振荡参数，请注意调整延时时间。



```

TITLE      "zlg7289A TEST"
LIST       P=16C54
INCLUDE    P16C5X.INC

;-----
;寄存器定义
;-----
BIT-COUNT   EQU    0X07
DATA-OUT     EQU    0X08
DATA-IN      EQU    0X09
TEN          EQU    0X0A
TIMER        EQU    0X0B
TIMER1       EQU    0X0C

;-----
;I/O 口定义
;-----
CS           EQU    0           ;CS 连接于 16C54 的 RA0
CLK          EQU    1           ;CLK 连接于 16C65 的 RA1
DAT          EQU    2           ;DAT 连接于 16C54 的 RA2
KEY          EQU    3           ;KEY 连接于 16C54 的 RA3

                ORG     1FFH
                GOTO    RESET
                ORG     00H

;-----
;延时子程序
;-----
LONG_DELAY   MOVLW    D'16'           ;设定延时时间为约 50uS
                MOVWF   TIMER
    
```

```

DELAY_LOOP    DECFSZ    TIMER
               GOTO     DELAY-LOOP
               RETLW     0
SHORT-DELAY    MOVLW     D '3'           ; 设定延时时间为 8uS
               MOVWF     TIMER
SHORT-LP       DECFSZ    TIMER
               GOTO     SHORT-LP
               RETLW     0

; -----
; 发送 1 个字节到 zlg7289A, 高位在前
; -----

SEND           MOVWF     DATA-OUT       ; 待发送数据存入 DATA-OUT
               MOVLW     D '8'
               MOVWF     BIT-COUNT       ; 设定计数器=8
               BCF       PORTA, CS       ; 设 CS 为低电平
               CALL      LONG-DELAY      ; 长延时
SEND-LOOP      BCF       STATUS, C
               RLF       DATA-OUT       ; 输出 1 位
               BCF       PORTA, DAT
               BTFSC     STATUS, C
               BSF       PORTA, DAT
               BSF       PORTA, CLK       ; 设 CLK 为高电平
               CALL      SHORT-DELAY     ; 短延时
               BCF       PORTA, CLK       ; 设 CLK 为低电平
               CALL      SHORT-DELAY
               DECFSZ    BIT-COUNT       ; 检查是否 8 位均发送完毕
               GOTO     SEND-LOOP       ; 未发送完, 发送下一位
               BCF       PORTA, DAT
               RETLW     0               ; 发送完毕, 返回

; -----
; 从 zlg7289A 接收一个字节, 高位在前
; -----

RECEIVE        MOVLW     D '8'
               MOVWF     BIT-COUNT       ; 设定计数器=8
               MOVLW     B '11111100'   ; 设 RA2 (DATA) 口为输入状态
               TRIS      PORTA
               CALL      LONG-DELAY      ; 长延时
RECEIVE-LOOP   BSF       PORTA, CLK       ; 置 CLK 为高电平
               CALL      SHORT-DELAY     ; 短延时
               BSF       STATUS, C
               BTFSS     PORTA, DAT
               BCF       STATUS, C
               RLF       DATA-IN       ; 读取一位数据
               BCF       PORTA, CLK       ; 置 CLK 为低电平

```

```

CALL      SHORT-DELAY
DECFSZ    BIT-COUNT      ; 是否已接收 8 位数据
GOTO      RECEIVE-LOOP
MOVLW     B '11111000'    ; 重新设 RA2 (DATA) 口为输出态
TRIS      PORTA
RETIW     0
; -----
; 初始化
; -----
RESET     MOVLW     B '11111000'    ; I/O 口初始化
          TRIS      PORTA
          MOVLW     B '11111001'
          MOVWF     PORTA
          MOVLW     PORTA
          MOVLW     0X19            ; 延时约 25mS
          MOVWF     TIMER
START-DELAY MOVLW     0XFF
          MOVWF     TIMER1
START-DELAY1 DECFSZ   TIMER1
          GOTO      START-DELAY1
          DECFSZ    TIMER
          GOTO      START-DELAY
          MOVLW     B '10100100'    ; 发复位 (清除) 指令
          CALL      SEND
          BSF       PORTA, CS       ; 恢复 CS 为高电平
; -----
; 主程序
; -----
MAIN      BTFSC     PORTA, KEY      ; 检测是否有键按下
          GOTO      MAIN
          MOVLW     B '00010101'    ; 有键按下, 发送读键盘指令
          CALL      SEND            ; 发送读键盘指令
          CALL      RECEIVE         ; 从 zlg7289A 读键盘代码
          BSF       PORTA, CS       ; 设 CS 为高电平
; -----
; 16 进制—BCD 码转换
; -----
          CLRF      TEN
GET-DEC   MOVLW     D '10'
          SUBWF     DATA-IN, W
          SKPC
          GOTO      ORER
          MOVWF     DATA-IN
          INCF      TEN

```

