

产品名称

MK7A22P

例案标题 MK7A22P的AD转换功能使用说明

简介

MK7A22P 提供了12路AD转换模拟输入端口，分别与I/O口复用。用户可以通过寄存器AD_CTL3的Bit3~0来进行分配，下面是AD_CTL3寄存器功能结构表。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL3	--	--	--	--	PISEL3	PISEL2	PISEL1	PISEL0

用户可以对照下表选择所需要的AD转换模拟输入端口。

Bit3	Bit2	Bit1	Bit0	配置
PISEL3	PISEL2	PISEL1	PISEL0	
0	0	0	0	所有 port 为数字输入
0	0	0	1	AN0
0	0	1	0	AN0~1
0	0	1	1	AN0~2
0	1	0	0	AN0~3
0	1	0	1	AN0~4
0	1	1	0	AN0~5
0	1	1	1	AN0~6
1	0	0	0	AN0~7
1	0	0	1	AN0~8
1	0	1	0	AN0~9
1	0	1	1	AN0~10
1	1	0	0	AN0~11

对MK7A22P来说，ADC功能分为“转换”和“比较”两种模式。这两种模式是由AD_CTL1寄存器的MODE(Bit5)位控制的。当MODE为0时为转换模式，也就是将对应输入端口中的模拟量转换成数据，存入AD_DAT寄存器里面；当MODE为1时为比较模式，也就是将对应输入端口中的模拟量转换成数据，并与事先存放在AD_DAT寄存器里面的数据进行比较，比较之后的结果通过AD_CTL2寄存器的RSUT(Bit7)位的值来体现，下面分别是AD_CTL1和AD_CTL2的功能结构表。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL1	EN	--	MODE	--	CHSEL3	CHSEL2	CHSEL1	CHSEL0

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL2	RSUT	--	--	--	--	--	CKSEL1	CKSEL0

在对多个模拟输入通道进行AD转换操作的时候，用户可以根据下表来选择当前的转换通道。表中的CHSEL0~CHSEL3分别位于AD_CTL1寄存器的Bit0~Bit3。

Bit3	Bit2	Bit1	Bit0	Input channel
CHSEL3	CHSEL2	CHSEL1	CHSEL0	
0	0	0	0	Channel 0, PA0 pin
0	0	0	1	Channel 1, PA1 pin
0	0	1	0	Channel 2, PA2 pin
0	0	1	1	Channel 3, PA3 pin
0	1	0	0	Channel 4, PA4 pin
0	1	0	1	Channel 5, PA5 pin
0	1	1	0	Channel 6, PA6 pin
0	1	1	1	Channel 7, PA7 pin
1	0	0	0	Channel 8, PC7 pin
1	0	0	1	Channel 9, PD1 pin
1	0	1	0	Channel 10, PD2 pin
1	0	1	1	Channel 11, PD3 pin

另外，在进行转换之前，用户应该根据实际需要由下表选择适当的转换时钟。表中的CKSEL1和CKSEL0分别位于AD_CTL2寄存器的Bit1和Bit0。

Bit1	Bit0	转换时钟
CKSEL1	CKSEL0	
0	0	系统时钟 X2
0	1	系统时钟 X8
1	0	系统时钟 X32
1	1	系统时钟 X128

为了让用户更深入的了解MK7A22P的AD转换功能的使用方法，我们提供了如下实例：

DEMO程序

汇编程序文档

```

;-----
#include "MK7A22P.INC" ;编译该文档需包含"MK7A22P.INC"文件

```

```

ORG      00H          ;复位地址
LGOTO    RESET
ORG      04H          ;中断入口地址
IRET

```

RESET:

```

NOP
CLR      IRQM
CLR      IRQF
MOVLA    B'11111111' ;设所有AD采样口为输入状态
MOVAM    PA_DIR
MOVLA    B'10000000'
MOVAM    PC_DIR
MOVLA    B'00001110'
MOVAM    PD_DIR

MAIN:
NOP
NOP
MOVLA    00H
MOVAM    AD_CTL1     ;Bit7:adc使能， Bit5:模式， Bit3-0通道选择
MOVLA    03H
MOVAM    AD_CTL2     ;Bit7:比较结果， Bit1-0: adc时钟倍频
MOVLA    0CH
MOVAM    AD_CTL3     ;Bit3-0: 复用管脚功能选择
NOP
NOP
BTSC     AD_CTL1,7   ;等待EN为0（ad转换结束）， 以便进入下一次采样
LGOTO    $-1
NOP                      ;延时
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
BS       AD_CTL1,7   ;启动AD
BC       IRQF,6      ;清中断标志位
NOP
NOP
NOP
NOP
BTSS     IRQF,6      ;等待转换结束
LGOTO    $-1
BC       IRQF,6      ;转换结束。采样数据低位保存在AD_DATL
NOP                      ;高位保存在AD_DATH
NOP
NOP
NOP
NOP
```

```
NOP
NOP
LGOTO    $
```

```
END
```

```
;-----
```

MK7A22P.INC 文档

```
;-----Special Register-----
```

```
#DEFINE      INDF      00H
#DEFINE      PCL       01H
#DEFINE      PCH       02H
#DEFINE      STATUS    03H
#DEFINE      FSR       04H
#DEFINE      PA_DIR    05H
#DEFINE      PA        06H
#DEFINE      PB_DIR    07H
#DEFINE      PB        08H
#DEFINE      PC_DIR    09H
#DEFINE      PC        0AH
#DEFINE      PD_DIR    0BH
#DEFINE      PD        0CH
```

```
#DEFINE      TM0_CTL   10H    ;TIMER0:16-BIT TIMER
#DEFINE      TM0L_LA   11H
#DEFINE      TM0H_LA   12H
#DEFINE      TM0L_CNT  13H
#DEFINE      TM0H_CNT  14H
```

```
#DEFINE      TM1_CTL1  15H    ;TIMER1:8-BIT PWM(period)&Timer
#DEFINE      TM1_CTL2  16H
#DEFINE      TM1_LA    17H
#DEFINE      TM1_CNT   18H
```

```
#DEFINE      TM2_CTL   19H    ;TIMER2:8-BIT PWM(Duty)&Timer
#DEFINE      TM2_LA    1AH
#DEFINE      TM2_CNT   1BH
```

```
#DEFINE      TM3_CTL1  1CH    ;TIMER3:8-BIT PWM(period)&Timer
#DEFINE      TM3_CTL2  1DH
#DEFINE      TM3_LA    1EH
#DEFINE      TM3_CNT   1FH
```

```
#DEFINE      TM4_CTL   20H    ;TIMER2:8-BIT PWM(Duty)&Timer
#DEFINE      TM4_LA    21H
#DEFINE      TM4_CNT   22H
```

```
#DEFINE      IRQM      25H    ;IRQ
#DEFINE      IRQF      26H
#DEFINE      IRQM_1    27H
#DEFINE      IRQF_1    28H
```

```

#DEFINE      AD_CTL1  29H  ;ADC Control
#DEFINE      AD_CTL2  2AH
#DEFINE      AD_CTL3  2BH
#DEFINE      AD_DATL  2CH
#DEFINE      AD_DATH  2DH

```

```

#DEFINE      CMP_CTL  2FH
#DEFINE      FREQ_CTL 30H
#DEFINE      PA_PLU   31H
#DEFINE      PB_PLU   32H
#DEFINE      PC_PLU   33H
#DEFINE      PD_PLU   34H
#DEFINE      IO_INTE  35H
#DEFINE      PA_INTE  3AH
#DEFINE      WDT_CTL  3DH
#DEFINE      TAB_BNK  3EH
#DEFINE      SYS_CTL  3FH

```

;------BIT DEFINE-----;

```

#DEFINE      C        STATUS,0
#DEFINE      DC       STATUS,1
#DEFINE      Z        STATUS,2
#DEFINE      PDB      STATUS,3
#DEFINE      TOB      STATUS,4
#DEFINE      SA0      STATUS,5
#DEFINE      SA1      STATUS,6

```

```

#DEFINE      TM0M     IRQM,0
#DEFINE      TM1M     IRQM,1
#DEFINE      TM2M     IRQM,2
#DEFINE      TM3M     IRQM,3
#DEFINE      TM4M     IRQM,4
#DEFINE      PAM      IRQM,5
#DEFINE      ADCM     IRQM,6
#DEFINE      INTM     IRQM,7

```

```

#DEFINE      TM0F     IRQF,0
#DEFINE      TM1F     IRQF,1
#DEFINE      TM2F     IRQF,2
#DEFINE      TM3F     IRQF,3
#DEFINE      TM4F     IRQF,4
#DEFINE      PAF      IRQF,5
#DEFINE      ADCF     IRQF,6
#DEFINE      INTF     IRQF,7

```