OpenTimer0

Function: Configure and enable timer0.

Include: `timers.h`

Prototype: `void OpenTimer0( unsigned char config );`

Arguments: ***config***

A bitmask that is created by performing either a bitwise AND operation (‘&’) or bitwise OR operation (‘|’) , which is user configurable, with a value from each of the categories listed below. These values are defined in the file `timers.h`.

Enable Timer0 Interrupt:

`TIMER_INT_ON`	Interrupt enabled
`TIMER_INT_OFF`	Interrupt disabled

Timer Width:

`T0_8BIT`	8-bit mode
`T0_16BIT`	16-bit mode

Clock Source:

`T0_SOURCE_EXT`	External clock source (I/O pin)
`T0_SOURCE_INT`	Internal clock source (TOSC)

External Clock Trigger (for `T0_SOURCE_EXT`):

`T0_EDGE_FALL`	External clock on falling edge
`T0_EDGE_RISE`	External clock on rising edge

Prescale Value:

`T0_PS_1_1`	1:1 prescale
`T0_PS_1_2`	1:2 prescale
`T0_PS_1_4`	1:4 prescale
`T0_PS_1_8`	1:8 prescale
`T0_PS_1_16`	1:16 prescale
`T0_PS_1_32`	1:32 prescale
`T0_PS_1_64`	1:64 prescale
`T0_PS_1_128`	1:128 prescale
`T0_PS_1_256`	1:256 prescale

Remarks: This function configures timer0 according to the options specified and then enables it.

File Name: `t0open.c`

Code Example:

With bitwise AND (‘&’) mask:

```
OpenTimer0( TIMER_INT_OFF &
            T0_8BIT       &
            T0_SOURCE_INT &
            T0_PS_1_32 );
```

With bitwise OR (‘|’) mask:

```
OpenTimer0( TIMER_INT_OFF |
            T0_8BIT       |
            T0_SOURCE_INT |
            T0_PS_1_32 );
```