

Example instructions:

```
setup_timer_0(RTCC_EXT_L_TO_H | RTCC_DIV_1);
```

```
//setup timer0 with external clock source, increment on rising
```

```
//edge of clock signal, prescaler=1 //such that there is no prescaling
```

```
setup_timer_0(RTCC_INTERNAL | RTCC_DIV_1); //setup timer0 with internal clock source, no prescaling
```

```
setup_timer_0(RTCC_INTERNAL | RTCC_DIV_8); //setup timer0 with internal clock source, prescaler=8
```

```
set_timer0(0); //set TMR0 to 0 i.e initialize it ; Note that Timer0 never stops, hence always continues to count!!!
```

```
set_timer0(100); //set TMR0 to 100
```

```
data=get_timer0(); //get TMR0 value, where data must be an 8-bit variable
```

AKA is an abbreviation to "Also Known As". AKA is used frequently in the CCSPIC-C manual II

```
//Connect an LED to PORTB RB7 pin
```

```
//Connect RB0 pin to 0 voltage level than to +5 voltage level to generate interrupt
```

```
#include <16F877A.h>
```

```
#FUSE5 NOWDT,XT,PUT,NOPROTECT,BROWNOUT,NOLVP,NOCPD,NOWRT #use
```

```
delay(clock=4000000)
```

```
#use fast_io(B)
```

```
#int_EXT void EXT_isr()
```

```
{
```

```
delay_ms(20); //software debounce
```

```
output_high(PIN_B7); //RB7 pin at high voltage level
```

```
delay_ms(100); //delay 0.1 seconds
```

```
output_low(PIN_B7); //RB7 pin at low voltage level
```

```
}
```

```
void main()
```

```
{
```

```
set_tris_b(0b01111111); //set PORTB bit 7 as output, other bits as input
```

```
output_bit(PIN_B7,0); //RB7 pin at low voltage level
```

```
ext_int_edge(L_TO_H); //select rising edge of external signal
```

```
enable_interrupts(INT_EXT); //enable external interrupt on RB0 pin
```

```
enable_interrupts(GLOBAL); //enable global interrupts
```

```
while (TRUE); //infinite loop
```

```
} //end of main
```