

```
#include "259macros.h"

/* global variables */
int counter = 0;

/* This ISR will be called every 100ms */
/* Display a counter on the red LEDs */
void counterISR()
{
    /* remember: no waiting in an ISR */
    counter++;
    *pLEDR = counter;

    /* clear source of COUNTER interrupt before returning */
    *pCOUNTER_STATUS = 1;
}

void enableCounterIRQ( int interval_cycles, ptr_to_function newISR )
{
    registerISR( IRQ_COUNTER, newISR ); /* specify which ISR to call when COUNTER interrupts */

    *pCOUNTER          = -interval_cycles; /* initial counter value */
    *pCOUNTER_RELOAD   = -interval_cycles; /* on overflow, start with this value */

    *pCOUNTER_STATUS   = 1; /* allow counter to send interrupts */
    enableInterrupt( IRQ_COUNTER ); /* allow CPU to receive COUNTER interrupts */
}

int main( int argc, char *argv[] )
{
    /* Clears all interrupt enables, then enables interrupts */
    initInterrupts();

    /* Set up COUNTER to interrupt and call the ISR every 100ms. */
    enableCounterIRQ( 100*ONE_MS, counterISR );

    /* main loop is busy doing nothing, forever */
    while( 1 )
        ;

    return 0; /* never gets here */
}
```