```
/*****************************************************************************/
/* IRQ.C: IRQ Handler                                                        */
/*****************************************************************************/
/* This file is part of the uVision/ARM development tools.                   */
/* Copyright (c) 2005-2006 Keil Software. All rights reserved.               */
/* This software may only be used under the terms of a valid, current,       */
/* end user licence from KEIL for a compatible version of KEIL software      */
/* development tools. Nothing else gives you the right to use this software. */
/*****************************************************************************/

#include <LPC23xx.H>                    /* LPC23xx definitions               */


short AD_last;                          /* Last converted value              */
unsigned char clock_700us_Puls;                 /* Flag activated each second          *
/
unsigned char clock_700us_Pause;                 /* Flag activated each second
*/
unsigned char clock_1s_Puls;                /* Flag activated each second         */
unsigned char clock_1s_Pause;                /* Flag activated each second         */
int i;
/* Import function for turning LEDs on or off                                */
extern void LED_On (unsigned int num);
extern void LED_Off(unsigned int num);


/* Timer0 IRQ: Executed periodically                                         */
__irq void T0_IRQHandler (void) {
  static int clk_cntr0;
  static int clk_cntr1;
  static int clk_cntr2;
  static int clk_cntr3;

  unsigned int n;
  unsigned int v;

  clk_cntr0++;
  if (clk_cntr0 >= 7) {
       clk_cntr0 = 0;
    clock_700us_Puls = 1;
  }


  clk_cntr1++;
  if (clk_cntr1 >= 10) {
    clk_cntr1 = 3;
    clock_700us_Pause = 1;     }


   clk_cntr2++;
  if (clk_cntr2 >= 5000) {
    clk_cntr2 = 0;
    clock_1s_Puls = 1;                          /* 60ms breiter Puls für serielle Daten
ausgabe mit 120ms Taktrate */
  }
   clk_cntr3++;
  if (clk_cntr3 >= 7500) {
    clk_cntr3 = 2500;
    clock_1s_Pause = 1;                         /* 60ms breite Pause mit 120ms Taktrate
 */
  }


  v = (5*AD_last) >> 9;                 /* Scale the Value                   */
  for (n = 0; n < 8; n++)               /* Show the Value                    */
   if (n < v) LED_On (10);              /* Turn LED On                       */
    else       LED_Off(10);             /* Turn LED Off                       */
```

```
  AD0CR |= 0x01000000;                  /* Start A/D Conversion               */

  T0IR        = 1;                      /* Clear interrupt flag               */
  VICVectAddr = 0;                      /* Acknowledge Interrupt              */
}


/* A/D IRQ: Executed when A/D Conversion is done                               */
__irq void ADC_IRQHandler(void) {

  AD_last = (AD0DR0 >> 6) & 0x3FF;      /* Read Conversion Result             */

  VICVectAddr = 0;                      /* Acknowledge Interrupt              */
}
```