How to use the IGW/800 or IGW/900 Watchdog

The DIL/NetPC DNP/528x of your IGW/800 or IGW/900 Linux Device Server offers a hardware watchdog. This watchdog timer generates a reset after a master enable and without a valid trigger sequence within a specific time period. Please see also the document *mHTigw8-9.pdf: How to prepare the IGW/800 or IGW/900 for Watchdog usage*.

- **1. Step**: Add the watchdog trigger code to your Linux application. The following C source code shows a sample:

```
#include <unistd.h>
#include <stdio.h>

#include "ssvhwa.h"

#define MCFBAR  0x40000000

#define WCR     (MCFBAR + 0x00140000)  // 16 bit register
#define WMR     (MCFBAR + 0x00140002)  // 16 bit register
#define WCNTR   (MCFBAR + 0x00140004)  // 16 bit register
#define WSR     (MCFBAR + 0x00140006)  // 16 bit register

int main (void)
{

   // get user identity

   if (geteuid () != 0) {
      fprintf (stderr, "No root access rights !\n");
      exit (1);
   }

   // open ssvhwa driver

   if (ssvhwa_open () < 0) {
      perror ("ssvhwa open");
      exit (-1);
   }

    // internal enable watchdog, set timeout value

    printf ("Switch watchdog on...");
    ssvhwa_write16 (WCR, 0x0001); // enable the watchdog - WCR=1
    ssvhwa_write16 (WMR, 0xFFFF); // max timeout - WMR=0xFFFF
    printf (" done.\n");

    // trigger watchdog every 4 seconds for ever ...
```

```
    for (;;) {

       static int i=0;

       // execute the 2 step watchdog trigger sequence

       ssvhwa_write16 (WSR, 0x5555);   // 1. step - WSR=0x5555
       ssvhwa_write16 (WSR, 0xAAAA);   // 2. step - WSR=0xAAAA

       printf ("\rTrigger watchdog %d times...", ++i);
       fflush (stdout);
       sleep (4);                      // sleep 4 seconds
    }
    ssvhwa_close ();
    return 0;
}
```

- **2. Step**: Make sure, that the watchdog trigger sequence is always executed within your application code. Otherwise the watchdog timer generates a reset.

  That is all.