TA-Omega + ÖBB X25 HOWTO

Heimo Schön <*heimo.schoen@gmx.at*> Version 0.2 , 22. Februar 2001

Setup a *TDT* PAD *TA-Omega* to work in a ÖBB X25 environment

1 Overview

1.1 About this HOWTO

This document describes how to use the PAD built by *TDT*. We use the *TA-Omega* in a special configuration for the Austrian Federal Railways *ÖBB*. The ÖBB X25 protocol is described in the "ÖBB X25 Pflichtenheft". This protocol specifies the Layer 4 to 7 for vital and none-vital comunication between railway-station equipment (e. q. interlocking, etc.). The ÖBB X25 protocol specification uses the CCITT X.25 / X.21 communication and uses PVC 1 for channel A and PVC 2 for channel B transmission.

The two-channel transmission in channel A and channel B is used for raising the safety of the connection. In channel B the data is transmitted inverted. For more details cunsult the "ÖBB X25 Pflichtenheft".

If you have any suggestions to this document, than feel free to contact me:

- Heimo Schön <*heimo.schoen@gmx.at*>
- *http://fly.to/heimo.schoen*

Have a lot of fun with this easy to use, easy to configure and very small X.25 PAD TA-Omega from TDT.

1.2 Test configuration

For testing the configuration in this document I used the following testequipment:

```
                +-----------------+
              R2|                 |     DAG64
    ch. A -----|                 |R1   +-----+
              R3|    TA-Omega     |-----|Modem|~~~~~~~~~~~~~~~+
    ch. B -----|                 |     +-----+               |
                |                 |                           |
                +-----------------+            2-wire         |
                                               longdistance|
                +-------------+                connection  |
                |             |         DAG64               |
    ch. A -----|             |        +-----+              |
                | Datus-PAD   |---------|Modem|~~~~~~~~~~~~~~~+
    ch. B -----|             |        +-----+
                |             |
                +-------------+

          TA-Omega ..... TA-Omega from TDT
          Modem ........ X.21 Modem with 64 kBit/sec
                         e.q.: Nokia SB 64k
                               or
```

```
                          Siemens DAG 64
Datus-PAD .... 5820 with 2 V.24 connectors and 1 X.21
               connector
```

On channel A and B on both ends of the testconfiguration (Datus-PAD and TA-Omega) i connected the Testrechner TR application, which you also can find on my homepage. You may use any other applications with V.24 connection with 19.200 baud, no parity, 8 Bit, 1 Stopbit.

1.3 Version History

- 0.1 09-Feb-2001 First trial

2 HW connection to TA-Omega

2.1 V.24 Cable to TA-Omega

Build a cable with the following pinout:

```
under construction
```

2.2 X.21 Cable between TA-Omega and Modem

```
under construction
```

Connect the cable to R2 and R3 on the TA-Omega and on the other hand e.g. to ttyS1 (COM2) of your Linux box.

3 SW connection to your TA-Omega

3.1 Terminalemulation minicom

Start minicom on your linuxbox. You can type minicom com2, when you create the following minicom configuration file:

```
# file: /etc/minicrc.com2
#
# this minicom configuration file is used to make
# a connection to a TA-Omega via ttyS1 (COM2)
#
# Machine-generated file - use "minicom -s" to change parameters.
#
pr port             /dev/ttyS1
```

```
pu baudrate        9600
pu rtscts          No
```

4 automatic configuring

4.1 Overview

In the following chapters you find a small C-Program for auto configure the TA-Omega. There is also the Makefile the startscript and the inputscript which is read from the C-program omeag.c.

4.2 Dip-Switch

Behind the frontpanel you find two Dip-switches. On the left side there is a switch numbered from 1 to 12 and on the right side there is a switch numbered from 1 to 8. Here are you find the settings for this two Dip-Switches in Oebb-Mode:

```
Switch 1-12 (left)
1-4   = ON
5,6   = OFF
7,8   = ON
9     = OFF
10-12 = ON
summary: ON  = 1,2,3,4,7,8,10,11,12
         OFF = 5,6,9

Switch 1-8 (right)
1-6 = ON
7   = OFF
8   = ON
summary: ON  = 1,2,3,4,5,6,8
         OFF = 7
```

4.3 omega.c

```
/*
 * omega.c
 *
 * input parser for PAD-configuration scripts
 * ==========================================
 *
 #
 # Copyright (C) 2001 Heimo Schön <heimo.schoen@gmx.at>
 #
 # This program is free software; you can redistribute it and/or modify
 # it under the terms of the GNU General Public License as published by
 # the Free Software Foundation; either version 2, or (at your option)
 # any later version.
 #
 # This program is distributed in the hope that it will be useful,
```

```
 # but WITHOUT ANY WARRANTY; without even the implied warranty of
 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 # GNU General Public License for more details.
 #
 # You should have received a copy of the GNU General Public License
 # along with this program; if not, write to the Free Software
 # Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 #
*/

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <assert.h>
#include <signal.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/ioctl.h>

int inputpid=0;

static int source_fh = 0;       // Filehandle
static int input_fh  = 0;       // Filehandle
static int out_fh    = 0;       // Filehandle
static int log_fh    = 0;       // Filehandle
unsigned char stty_buffer[10000];
unsigned char out_buffer[10000];
unsigned char source_buffer[10000];

void p_usage()
{
  fprintf(stderr, "  \n");
  fprintf(stderr, "  omega <command-file> <device>\n");
  fprintf(stderr, "  \n");
  fprintf(stderr, "       <command-file> filename of a file with\n");
  fprintf(stderr, "                      PAD-parameters; one command\n");
  fprintf(stderr, "                      in each line\n");
  fprintf(stderr, "  \n");
  fprintf(stderr, "       <device>       the serial device where omega can\n");
  fprintf(stderr, "                      find the PAD\n");
  fprintf(stderr, "  \n");
  fprintf(stderr, "  example:  omega test.cmd /dev/ttyW7\n");
  fprintf(stderr, "  \n");
}

void startinput()
{
  unsigned char in_buffer[10000];
  int rece = 0;

  in_buffer[0] = 0;
```

```
  if ( ( inputpid = fork() ) == 0 ) {
    while (1) {
      if ((rece = read(input_fh, in_buffer, 1024)) > 0) {
        in_buffer[rece] = 0;
        if (log_fh > 0)
          write(log_fh, in_buffer, strlen(in_buffer));

        fprintf(stdout, "%s", in_buffer);

        // Flush the output that user can see something ;-)
        fflush(stdout);
        fflush(stderr);

        in_buffer[0] = 0;
      }
    }
  }
  fprintf(stdout, "reader forked with pid %d\n", inputpid);
}



void p_io()
{
  int rece = 0;
  char command[10000];
  int com_p = 0; // command-pointer

  char sobuf[1024];
  char sebuf[1024];
  setvbuf(stdout, sobuf, _IONBF, 0);
  setvbuf(stderr, sebuf, _IONBF, 0);

  while ( (rece = read(source_fh, source_buffer, 1)) == 1) {
    if (source_buffer[0] == '\n') {

      command[com_p] = 0;

      if (command[0] != '#') {

        if ( ! strncmp(command, "sleep", 5)) {
          fprintf(stdout, "sleeping %d seconds ...\n", atoi([5]));
          // Flush the output that user can see something ;-)
          fflush(stdout);
          fflush(stderr);
          sleep ( atoi([5]) );

        } else {

          sprintf(out_buffer, "%s\r",command);
          write(out_fh, out_buffer, strlen(out_buffer));
          sleep(1);

        }
```

```
      } else {

        command[com_p] = '\n';
        command[com_p+1] = 0;
        if (log_fh > 0)
          write(log_fh, command, strlen(command));
        fprintf(stdout, "%s", command);
      }

      // reset Command-pointer
      com_p = 0;

      // Flush the output that user can see something ;-)
      fflush(stdout);
      fflush(stderr);

    } else {
      command[com_p] = source_buffer[0];
      com_p++;
    }
  }
}

// ----------------------------------------------------------------------
int main (int argc, char *argv[])
{
  if (argc >= 3) {
    if (argc == 4)
      log_fh  = open (argv[3], O_WRONLY);
    source_fh = open (argv[1], O_RDONLY);
    out_fh    = open (argv[2], O_WRONLY);
    input_fh  = open (argv[2], O_RDONLY);
    sprintf(stty_buffer,
            "stty 9600 -echo raw < %s  ",
            argv[2]);
    fprintf(stdout,"%s\n",stty_buffer);
    system(stty_buffer);
  } else
    out_fh = 0;

  if (out_fh > 0) {
    startinput();              // fork the reader process
    p_io();                    // the command process
    kill(inputpid, SIGKILL);
  } else {
    p_usage();
  }
}
```

4.4 Makefile

The following listing should be copied into a file named Makefile. Afterwards you can type make clean ; make all ; make strip

```
#
# Makefile
#
#      Makefile for TA-OMEGA Configuration Program
#
# Copyright (C) 2001 Heimo Schön <heimo.schoen@gmx.at>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#

LDOPTIONS = -pedantic -Wall -g -DYY_SKIP_YYWRAP
       CP = cp
    STRIP = strip

# Objects and Sources:
     OBJS = omega.o
     SRCS = omega.c

all:    omega

omega:  $(OBJS) $(DEPLIBS)
        $(RM) $@
        $(CC) -o $@ $(LDENTRY)  $(OBJS) $(LDOPTIONS)  $(LIBS) $(LDLIBS) $(EXTRA_LOAD_FLAGS)

omega.stripped: omega
        $(CP) $< $@
        $(STRIP) $@

strip:: omega.stripped

clean::
        $(RM) omega *.o *~ omega.stripped

depend::
        @if [ -n "$(SRCS)" ] ; then set -x;\
          $(DEPEND) $(DEPENDFLAGS) $(SRCS) ;\
        fi

tags::
        etags *.[ch]
```

4.5 configuration script makeomega

Copy the following lines into a file named makeomega. Dont´t forget to chmod the script: chmod a+x makeomega

```
#! /bin/bash
#
# makeomega - configuring a TA-OMEGA-PAD
#
# Copyright (C) 2001 Heimo Schön <heimo.schoen@gmx.at>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#

if [ $1 ] ; then
  if [ -e /dev/$1 ] ; then
    echo Send OMEGA config to device /dev/$1
    export DATDEV=/dev/$1
    omega omega.cmd $DATDEV | tee omega.log
  else
    echo Could not find device /dev/$1
    echo "    example : $0 ttyS0"
  fi;
fi;
```

4.6 Parameterscript

Here you can find the complete Parameterscript for configuring the TA-Omega. Check if you didn´t delete blank lines in the following file.

```
#
# commandfile for configuring a TA-OMEGA for OeBB Applications
#
# With best regards to Mr. Mirlach at www.tdt.de
#
#
#
# Grundzustand herstellen
b s md:a
# Warmstart
w s md:a
```

```
sleep 5
#
# ?
s p md:8 ltc:3
# ?
s p md:9,10 prof:0
#
# ?
s c md:8 ch:1 md:9 ch:1 ACC:NO
s c md:9 ch:1 md:8 ch:1 ACC:NO
s c md:8 ch:2 md:10 ch:1 ACC:NO
s c md:10 ch:1 md:8 ch:2 ACC:NO
#
# neu vom 09.02.2001
s prof md:9,10 prof:0 1:16
# FDZR needs 19200 baud
s p md:9,10 SP:19200
#
# abschliessend noch einen Warmstart
w s md:a
sleep 5
#
#
# Action done - have a nice day
#
# ========================================
```

5 Disclaimer

5.1 Comments

I would appreciate (but do NOT REQUIRE) that any changes (bugs, new features or spelling errors) be sent to me for inspection, discussion and possible inclusion into the next version.

5.2 Copyright