Siemens Business Services

Colledia Control

BFE KSC Label Driver

November 2007

Introduction

This Colledia Control driver has been written to send and retrieve UMD labels from the B.F.E KSC UMD hardware. This equipment was first installed in the TV Studios at Pacific Quay in Glasgow.

The driver makes use of an external infodriver and uses a serial interface to communicate with the hardware. The hardware can be configured to use either RS232 or RS422 with a range of speeds (default is 38400).

The KSC UMDs have up to 16 characters in two blocks of 8, treated as left side and right side of the UMD by the driver. In the PQ TV Studios, the left side is used for a fixed name of the source, and the right side for dynamic parts of the label.

The protocol is a simple hex byte based one, with a small number of commands to send and retrieve data to and from the hardware.

Ini File Settings

The driver on start-up will read in required entries from its appropriate device ini file. The number of this dev_ini file must be passed to the driver as part of the start-up command, and is the number of an external infodriver, that as a process must already be running.

The entries from the dev_xxx.ini file are read in from the section headed KscUeZLabel – and denote the comms parameters for the hardware.

```
[KscUeZLabel]
DebugMode=1
LogMode=0
Port=1          -- port number on the PC – will most likely need to be correctly set
Speed=38400     -- default speed etc for BFE hardware
DataBits=8
StopBits=1
Parity=N
```

The driver then reads in as many entries that exist in dev_xxx.db2 and dev_xxx.db3 files.

The dev_ini DB2 file holds the mapping between the router source number (index number) to the QV Address of the UMD,
It is now known that there is a 1:1 mapping of source number and BFE database entry / QV Address – so entries in this db file need to be set 0001=1, 0002=2 etc for the number of sources on the router concerned.

The dev_ini DB3 file holds the settings of who has “control” over the labelling of the UMD.
Settings are : 1 = BNCS Controlled source
2 = BFE Controlled source
0 = no control – source ignored – no data sent to / retrieved from UMD for this source

e.g. 0001=1 -- source is under Colledia control and the driver will send the label data to the BFE hardware.
e.g 0002 =2 -- the driver will NOT send data to BFE hardware, but will request label data from the hardware.

SET ALL entries in this file to 1 –as default – ie source is under BNCS control

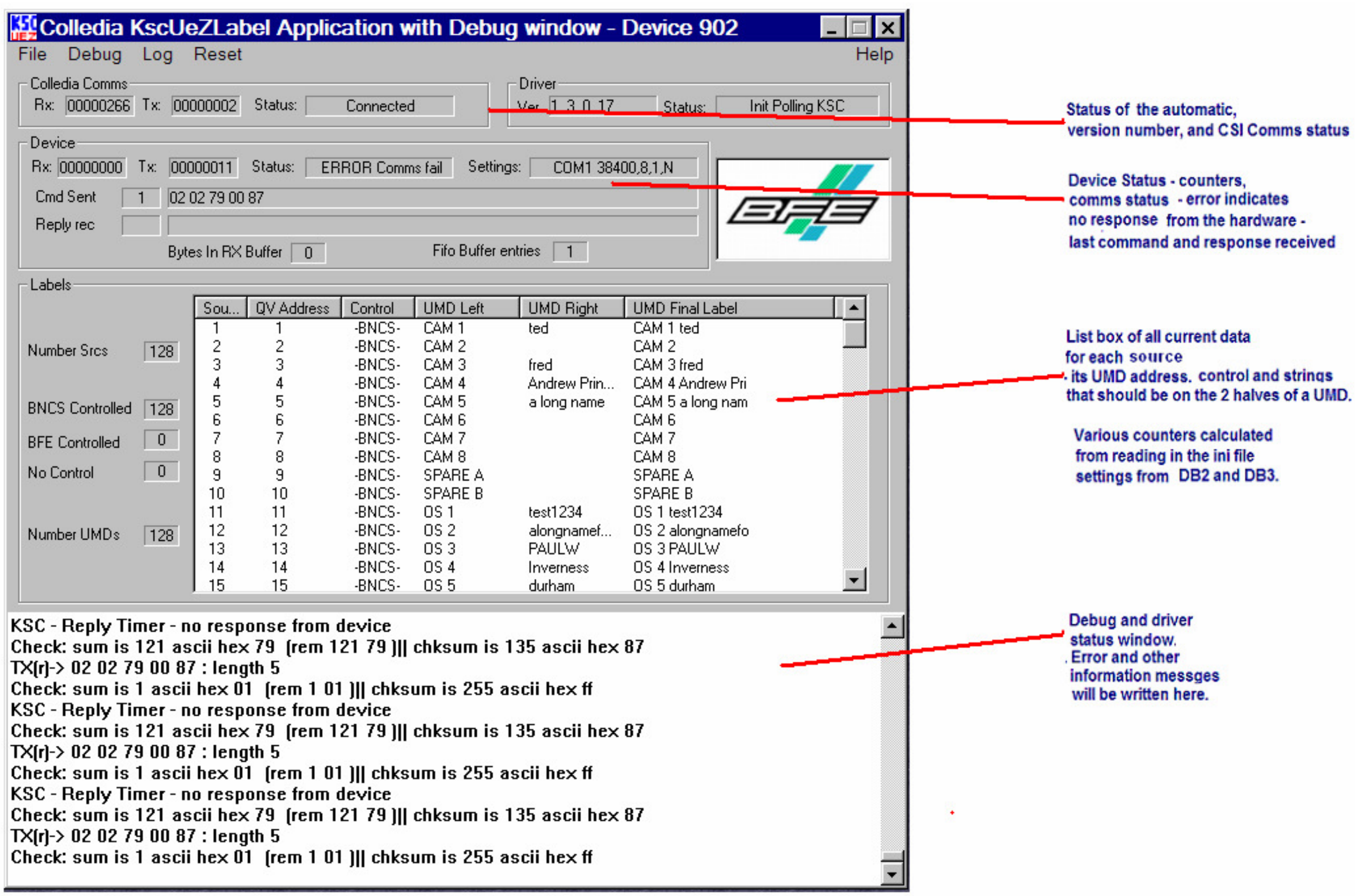
The driver reads all this data in at start-up and displays its findings in the driver application window for verification.

Infodriver Slot Usage

Slots 1 to 4000 are reserved and tied to sources and hold the string data that will be displayed on UMDs, as per the ini file settings already described. The format of the string data is divided into the left and right data for each half of a UMD. A vertical pipe "|" delimits the two halfs, eg " OS_1 | Glasgow" will result in OS_1 on the left side of the UMD and Glasgow on the right side of the UMD. Each half is limited to 8 characters. If no | character is given as part of the string, the driver will make best efforts to place the string on the two parts of the UMD.

Slot 4001 denotes comms OK (1) or comms FAIL (0) to the hardware. Comms failure could be a result of missing, incorrectly wired or disconnected cables, de-powered hardware, or comms port missing, comms port already in use, or comms port failure within the PC.

Driver Application and error messages



Hardware Protocol

Protocol between UeZ and BFE-KSC

State: 01.06.2006
2.0 21.08.2007 16-character-Labels added

Content

1. Physical Layer

KSC9000 as used at PQ

RS422 or RS232, Transferrate 38,4 kBaud maximum, 8,N,1
Transferrate can be modified according to project.
Connector at KSC9000: RJ45, 8-pol

RS422:
Pin Name Function
1 Gnd Ground
2 Gnd Ground
3 TC transwith-Ground
4 TA transwith (-)
5 TB transwith (+)
6 RC receive-ground
7 RA receive (-)
8 RB receive (+)

RS232:
Pin Name Function
1
2
3 Gnd Ground
4
5 TxD transwith data
6
7 RxD receive data
8

2. Logical Layer

1 2 3..n m
STX Data Byte, Count Data Bytes, max 252, Check Sum
with:
STX = 02 hex
Data Byte = Count Number of Databytes
Data Bytes = Databytes (incl.. messageidentifier)
Check Sum 2s Complement of Sum of all Databytes – sum of data bytes + checksum = 00.

Following Answers can occur:
ACK = 06 hex positiv Acknowledge
NAK = 15 hex negative Acknowledge

3. Command and Response Messages

Message 0x79 MC_STARTED
Direction From UeZ to KSC
Coding 0x79 * Status
Description of Parameters
Status 0x00..0xFF

Databytes 2
Notes Is to send after each reboot of UeZ
Answer ACK

Message 0x01 GET_DEVICE_STATUS
Direction From UeZ to KSC
Coding 0x01
Description of Parameters
Databytes 1
Notes Ask for Device-Status. Can be used as a Polling
Answer ACK
DEVICE_STATUS

Message 0x02 DEVICE_STATUS
Direction From KSC to UeZ
Coding 0x02 * Status
Description of Parameters
Status 0x00 Databytes 2
Notes No reall Status-Info, Status-Byte is always 0x00
Answer ACK

Message 0x7B SET_SOURCE_LABEL
Direction From KSC to UeZ and From UeZ to KSC
Coding 0x7B * QV-No * Label
Description of Parameters
QV-No Label
0..9, Number of Crosslink, 4 Bytes
Video-Label of Crosslink, 8 Bytes
Databytes 13
Notes Transmission of a Video-Sourcelabel

Answer ACK

Message 0x77 GET_SIGNAL_LABEL
Direction From UeZ to KSC
Coding 0x77 * Mode * QV-No
Description ofParameters
Mode QV-No
0x00 0..9, Number of Crosslink, 4 Bytes
Databytes 6
Notes Question for a Video-Sourcelabel
Answer ACK
SET_SIGNAL_LABEL

Message 0x78 SET_SIGNAL_LABEL
Direction From UeZ to KSC
Coding 0x78 * Mode * QV-No * <Label>
Description of Parameters
Mode QV-No Label
0x00 0..9, Number of Crosslink, 4 Bytes Video-Label of Crosslink, 8 Bytes
Databytes 14
Notes 1. As a spontaneous Message after Labelchange 2. As Answer to GET_SIGNAL_LABEL
Answer ACK

Message 0x8B SET_SRC_LABEL5
Direction From KSC to UeZ and From UeZ to KSC
Coding 0x8B * QV-No * <Label1><Label2><Label3><Label4><Label5>
Description of Parameters
Qv-No Label1 Label2 Label3 Label4 Label5
0..9, Number of Crosslink, 4 Bytes Video-Label of Crosslink, 8 Bytes
Audio1-Label of Crosslink, 8 Bytes
Audio2-Label of Crosslink, 8 Bytes
Audio3-Label of Crosslink, 8 Bytes
Audio4-Label of Crosslink, 8 Bytes
Databytes 45
Notes Transmission of a Video-Sourcelabel and 4 Audio-SourceLabels
Answer ACK

Message 0x87 GET_SIGNAL_LABEL5
Direction From UeZ to KSC
Coding 0x87 * Mode * QV-No
Description of
Parameters
Mode
QV-No
0x00
0..9, Number of Crosslink, 4 Bytes
Databytes 6
Notes Question for the 5 Labels of Video and Audio
Answer ACK
SIGNAL_LABEL5

Message 0x88 SIGNAL_LABEL5
Direction From KSC to UeZ and From UeZ to KSC
Coding 0x88 * Mode * QV-No * <Label1><Label2><Label3><Label4><Label5>
Description of Parameters
Mode QV-No Label1 Label2 Label3 Label4 Label5
0x00 0..9, Number of Crosslink, 4 Bytes
Video-Label of Crosslink, 8 Bytes
Audio1-Label of Crosslink, 8 Bytes
Audio2-Label of Crosslink, 8 Bytes
Audio3-Label of Crosslink, 8 Bytes
Audio4-Label of Crosslink, 8 Bytes
Databytes 46
Notes Transmission of a Video-Sourcelabel and 4 Audio-SourceLabels with Mode-Byte
Answer ACK

Message 0x9B SET_SOURCE_LABEL16
Direction From KSC to UeZ and From UeZ to KSC
Coding 0x9B * QV-No * Label
Description of Parameters
QV-No Label
0..9, Number of Crosslink, 4 Bytes
Video-Label of Crosslink, 16 Bytes
Databytes 21
Notes Transmission of a Video-Sourcelabel with 16 characters
Answer ACK

Message 0x97 GET_SIGNAL_LABEL16
Direction From UeZ to KSC
Coding 0x97 * Mode * QV-No
Description of Parameters
Mode QV-No
0x0 0..9, Number of Crosslink, 4 Bytes
Databytes 6
Notes Question for a Video-Sourcelabel with 16 characters
Answer ACK

SET_SIGNAL_LABEL16
Message 0x98 SET_SIGNAL_LABEL16
Direction From UeZ to KSC
Coding 0x98 * Mode * QV-No * <Label>
Description of Parameters
Mode QV-No Label
0x00 0..9, Number of Crosslink, 4 Bytes
Video-Label of Crosslink, 16 Bytes
Databytes 22
Notes 1. As a spontaneous Message after Labelchange
2. As Answer to GET_SIGNAL_LABEL16
Answer ACK

4. Miscellaneous

The Idea of this Interfacing is to have some crosslinks from KSC to UeZ and some crosslinks from UeZ to KSC. The name UeZ comes from the german word „Uebertragungs-Zentrum“ which means a Central-Router-System. Between KSC and UEZ the KSC is Slave, UEZ is Master. As long as KSC-Interface doesn’t receive anything from UEZ, it doesn’t send anything on his part.
UeZ starts Communication with der Message MC_STARTED. Afterwards it asks for Crosslink-Labels from KSC to UeZ (if there are any) (GET_SIGNAL_LABEL5), and sends his Sourcelabels according to the crosslink from UeZ to KSC (SET_SIGNAL_LABEL5).
Afterwards all Changes in Labels are sent spontaneously from both sides.
UeZ has to send any message within all 20 seconds, otherwise KSC-Interface would recognize as TIMEOUT. For this purpose one can send DEVICE_STATUS all 10 Seconds DEVICE_STATUS, which will be answered with ACK by the KSC.
The KSC-Interface uses exclusivly command SET_SRC_LABEL5 (that is all 5 Labels, without Mode-Byte) to send Labels spontanesously.