Inhaltsverzeichnis

The SKDS Protocol

to be moved onto the OpenOnBoardDiagnostics- Homepage

Actually „protocol“ only means the communication between two SKDS instances through Skype, but as this communication might also been done e.g. via TCP/IP, it was decided to publish the protocol here for possible third party implementations.

The communication is based on the exchange of XML-Strings, meaning that all information is packed into an XML format. The rootnode is always <SKDS>

Initiate a Connection

The remote link is initiated by the remote SKDS (the so called Master) by sending a request to the local SKDS (to which the modules are physically connected, the so called Slave):

```
<SKDS><MASTER></MASTER></SKDS>
```

If the slave accepts the request, it sends back the positive response

```
<SKDS><SLAVE></SLAVE></SKDS>
```

or in the case of any rejection it sends a negative response

```
<SKDS><BUSY>reason of rejection</BUSY></SKDS>
```

where the „reason of rejection“ is displayed at the Master as feedback to the user, as to why the connection was rejected.

Close a Connection

To close a connection, each side can send a <BUSY>- Tag to close:

```
<SKDS><BUSY>reason of rejection</BUSY></SKDS>
```

Telegram Exchange

Sending a Request

To request data from the module, the Master sends a message containing the request. As the handling of old KWP-telegrams or today's UDS-telegrams are different, there are 2 different types used:

```
<SKDS><FDS_REQ>packed KWP data</FDS_REQ></SKDS>
```

or

```
<SKDS><GGDS_REQ>packed UDS data<GGDS_REQ></SKDS>
```

packed KWP data

The packet KWP data consists of a sequence of the following tag and value combinations:

Tag	Content
SA	Sourceaddress of the Tester as numeric value, but only the last byte (0x7FE → 0x7E → 254)
TA	Targetaddress of the module as numeric value, but only the last byte (0x77E → 0x7E → 246)
LEN	Length of the telegram in bytes as numeric value
ERR	error byte as numeric value
DATA	base64 coded telegram content

packed UDS data

The packet UDS data consists of the same sequence as KWP, plus one additional Tag:

Tag	Content
SA	Sourceaddress of the Tester as numeric value, but only the last byte (0x7FE → 0x7E → 254)
TA	Targetaddress of the module as numeric value, but only the last byte (0x77E → 0x7E → 246)
LEN	Length of the telegram in bytes as numeric value
ERR	error byte as numeric value
DATA	base64 coded telegram content
OLDFRAME	0 or 1, reserved for later extensions

Sending the Response

After the slave has forwarded the request to the module and received the answer, it sends the result back to the master by

```
<SKDS><FDS_ANS>packed KWP data</FDS_ANS></SKDS>
```

or

```
<SKDS><GGDS_ANS>packed UDS data<GGDS_ANS></SKDS>
```

where again the packed data contains the details as shown above

Module Changes

When the slave detects an operational state feedback from a module in the communication flow, it reports this operational state back to the master:

```
<SKDS><SCANNER_EVENT_MOD_STATE><ID>id</ID><TEXT>text</SCANNER_EVENT_MOD_STATE></SKDS>
```

with the data

Tag	Content
ID	Address of the Module as numeric value, but only the last byte (0x7FE → 0x7E → 254)
TEXT	Operational State of the module in hexadecimal writing (e.g. $81)

When in contrast the Master „deletes“ a module ID by executing the ChangeID-command or by closing a window, the master sends the following command to the slave, mainly to switch off the tester present message of that module on the slave side:

```
<SKDS><SCANNER_COMMAND_MOD_REMOVE><ID>id</ID></SCANNER_COMMAND_MOD_REMOVE></SKDS>
```

again with the data

Tag	Content
ID	Address of the Module as numeric value, but only the last byte (0x7FE → 0x7E → 254)

Base64 Coding

to be corrected

As shown above, the telegram data itself is transfered as base64 coded binary data. For own implementations it is important to know that SKDS uses a slighty different character set compared to the base64 standard [1)].

The character set used by SKDS is

```
const
   Base64: string =
'23456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz~#%&*+-';
```

[1)]
this might be fixed in a later version