Diagnostic Communication of Vehicles

Main Functionality of an Automotive Control Unit

- Main task: Fulfill its controlling functionality (e.g. control transmission unit)
- Measure and collect the needed data for its main tasks
- Check and filter the incoming / measured signals
- Detect and log errors
- In case of error — according to the type — surmount error, or restrict functionality
- Configuration interface
- Automatic adaptation to environment
- Protection against stealing / manipulation
- Fulfill international and local standards
- Programmable / Patchable
- Provide information to other control units
- Communicate with an external tester
Goals of the Diagnostic Communication I.

- Main task: Fulfill its controlling functionality
- Measure and collect the needed data for its main task
  - Continuous observation of the control unit
  - Check reactions of the control unit
- Check and filter the incoming / measured signals
- Detect and log errors
  - Read error memory (source of the error, art and status)
  - Check reliability of incoming / measured signals

Goals of the Diagnostic Communication II.

- In case of error: surmount error or restrict own functionality according to the type of error
  - Read out error code
  - Provide information to find the source of the error
- After fixing error clear error memory
- Provide information to other control units
- Configuration interface
  - Read and write configurable parameters (adapting to the other control units or local law)
- Automatic adaptation to environment
  - Be able to clear learning values
Goal of the Diagnostic Communication III.

- Protection against stealing / manipulation
  - Authentication of ignition key and other control units
  - Chip-tuning protection
- Programmable / Patchable
  - The main part of the control unit SW can be reprogrammed
- Should fulfill international and local standards
  - CARB (Californian Air Resource Board) /
  - OBD (OnBoard Diagnostic)
  - EOBD (European OnBoard Diagnostic)
  - KOBD (Korean OnBoard Diagnostic)
  - ...

Vehicle Network Architecture in a High End Car
Communication System in the Vehicle

Position Lever Unit

Engine Control Unit

Brake Control Unit

Transmission Control Unit

Display Unit

Debreceni Egyetem - Informatikai Napok

Applicability and relationship between international standards

<table>
<thead>
<tr>
<th>Applicability</th>
<th>OSI 7 layer</th>
<th>Enhanced diagnostics services (non emission-related)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seven layer</td>
<td>Application (layer 7)</td>
<td>ISO 14229 / ISO 15765-3, ISO 14229 / ISO 14230-3</td>
</tr>
<tr>
<td></td>
<td>Presentation (layer 6)</td>
<td>---</td>
</tr>
<tr>
<td>according to</td>
<td>Session (layer 5)</td>
<td>ISO 15765-3, ISO 14230-3</td>
</tr>
<tr>
<td>ISO/IEC 7498</td>
<td>Transport (layer 4)</td>
<td>ISO 15765-2, ---</td>
</tr>
<tr>
<td>and</td>
<td>Network (layer 3)</td>
<td>ISO 15765-2, ---</td>
</tr>
<tr>
<td>ISO/IEC 10731</td>
<td>Data link (layer 2)</td>
<td>ISO 11898, ISO 14230-2</td>
</tr>
<tr>
<td></td>
<td>Physical (layer 1)</td>
<td>ISO 11898, ISO 14230-1</td>
</tr>
</tbody>
</table>

CAN K-Line
K-Line Layer 1

Logical 0:
- K-Line is under 20% of battery voltage at transmitter side
- K-Line is under 30% of battery voltage at receiver side

Logical 1:
- K-Line is above 80% of battery voltage at transmitter side
- K-Line is above 70% of battery voltage at receiver side
K-Line Layer 2: Overview

- No continuous communication: must be initialized by tester (WUP or 5BaudInit)
- No communication between the control units
- Each host has one or more fix addresses; multicasting is also possible
- Fixed Baud-rate at start of the communication (10.4 kBaud), it can be changed (until 250kBaud)
- Defined but configurable timeout values for communication session, request – response, response – next request separation time
- Initialization phase determines several format parameters of the further communication
- Arbitration: CSMA/CD

K-Line Layer 2: Initialization

Diagram showing the initialization phase with time stamps and sequence of events involving WUP, StartComm, and Communication phases.
## K-Line Layer 2: Message Structure

<table>
<thead>
<tr>
<th>Header</th>
<th>Data bytes</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fmt</td>
<td>Tgt&lt;sup&gt;1&lt;/sup&gt;</td>
<td>Src&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td>max. 4 bytes</td>
<td>max. 255 bytes</td>
<td>1 byte</td>
</tr>
</tbody>
</table>

Checksum calculation:
1) Byte is optional, depending on format byte.
2) Service identification, part of data bytes.

- **Fmt**: Format byte
- **Tgt**: Target identifier
- **Src**: Source identifier
- **Len**: Length of data bytes
- **Std**: Service identification
- **Data**: Data bytes
- **CS**: Checksum

## KWP2000 – K-Line, Layer 5-7: Service IDs

<table>
<thead>
<tr>
<th>Header</th>
<th>Data bytes</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fmt</td>
<td>Tgt&lt;sup&gt;1&lt;/sup&gt;</td>
<td>Src&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td>max. 4 bytes</td>
<td>max. 255 bytes</td>
<td>1 byte</td>
</tr>
</tbody>
</table>

Checksum calculation:
- **Fmt**: Format byte
- **Tgt**: Target identifier
- **Src**: Source identifier
- **Len**: Length of data bytes
- **Std**: Service identification
- **Data**: Data bytes
- **CS**: Checksum

- **0x01 – 0x3E, 0x80 – 0xBF**: Possible Service IDs
- **SID + 0x40**: positive response
- **0x7F**: negative response
KWP2000 – K-Line, Layer 5-7: Service IDs: Example

<table>
<thead>
<tr>
<th>Tester</th>
<th>C</th>
<th>33</th>
<th>F</th>
<th>81</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCU</td>
<td>83</td>
<td>F1</td>
<td>18</td>
<td>C1</td>
<td>XX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tester</th>
<th>C</th>
<th>33</th>
<th>F</th>
<th>21</th>
<th>XX</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCU</td>
<td>8x</td>
<td>F1</td>
<td>18</td>
<td>61</td>
<td>XX</td>
<td>XX</td>
</tr>
</tbody>
</table>

| Motronic | 8x | F1 | 10 | 61 | XX | XX | CS |

| ECUx     | 83 | F1 | XX | 7F | 21 | 31 | CS |

Debreceni Egyetem - Informatikai Napok

K-Line
- Reserved for diagnostic communication
- Longer data packets can be transmitted
- Configurable communication speed
- Arbitration must be implemented by SW (UART)
- Additional wire + HW Component (Layer1)
- Additional SW Driver for Layer 2

CAN
- Diagnostic & continuous communication between ECUs
- A CAN frame is max. 8 bytes: encapsulation of request required
- Fixed speed: because of the continuous bus configuration
- Bus arbitration, CAN-frame structure is handled by HW
- Wire + required HW component already exists
- SW Drivers already exist, only sw of diagnostic communication must be implemented
## CAN Layer 1 and Layer 2

Defined by ISO 11898

### CAN Layer 3 and Layer 4

<table>
<thead>
<tr>
<th>CAN ID</th>
<th>dlc</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>2-8</td>
<td>N_PCI 0xXX 0xXX 0xXX 0xXX 0xXX 0xXX</td>
</tr>
<tr>
<td>FF</td>
<td>8</td>
<td>N_PCI 0xXX 0xXX 0xXX 0xXX 0xXX 0xXX</td>
</tr>
<tr>
<td>FC</td>
<td>3-8</td>
<td>N_PCI 0xXX 0xXX 0xXX 0xXX 0xXX 0xXX</td>
</tr>
<tr>
<td>CF</td>
<td>2-8</td>
<td>N_PCI 0xXX 0xXX 0xXX 0xXX 0xXX 0xXX</td>
</tr>
</tbody>
</table>

- **SF**: start frame
- **FF**: frame format
- **FC**: frame control
- **CF**: continuation frame

- **0x0L**: L = Length of Data (4 bit)
- **0x1L**: L = Length of Data (12 bit)
- **0x3F BB SS**: Flow Control Status
- **F**: F
- **B**: BS: max number of CF without FC
- **S**: STmin follow-up time of CFs
- **0x2N**: N = Sequence Number
### KWP2000 over CAN

<table>
<thead>
<tr>
<th>CAN ID</th>
<th>dlc</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x7DF</td>
<td>8</td>
<td>0x02 0x10 0x81 0xXX 0xXX 0xXX 0xXX</td>
</tr>
<tr>
<td>0x7E9</td>
<td>8</td>
<td>0x02 0x50 0x81 0xXX 0xXX 0xXX 0xXX</td>
</tr>
<tr>
<td>0x7DF</td>
<td>8</td>
<td>0x02 0x21 0x80 0xXX 0xXX 0xXX 0xXX</td>
</tr>
<tr>
<td>0x7E9</td>
<td>8</td>
<td>0x02 0x0A 0x81 0xXX 0xXX 0xXX 0xXX</td>
</tr>
<tr>
<td>0x7E1</td>
<td>8</td>
<td>0x30 0x00 0xXX 0xXX 0xXX 0xXX 0xXX</td>
</tr>
<tr>
<td>0x7E9</td>
<td>8</td>
<td>0x21 0x05 0x06 0x07 0x08 0xXX 0xXX</td>
</tr>
</tbody>
</table>

### Applicability and relationship between international standards

<table>
<thead>
<tr>
<th>Applicability</th>
<th>OSI 7 layer</th>
<th>Enhanced diagnostics services (non emission-related)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seven layer</td>
<td>Application (layer 7)</td>
<td>ISO 14229 / ISO 15765-3</td>
</tr>
<tr>
<td>according to</td>
<td>Presentation (layer 6)</td>
<td>---</td>
</tr>
<tr>
<td>ISO/IEC 7498</td>
<td>Session (layer 5)</td>
<td>ISO 15765-3</td>
</tr>
<tr>
<td>and ISO/IEC 10731</td>
<td>Transport (layer 4)</td>
<td>ISO 15765-2</td>
</tr>
<tr>
<td></td>
<td>Network (layer 3)</td>
<td>ISO 15765-2</td>
</tr>
<tr>
<td></td>
<td>Data link (layer 2)</td>
<td>ISO 11898</td>
</tr>
<tr>
<td></td>
<td>Physical (layer 1)</td>
<td>ISO 11898</td>
</tr>
</tbody>
</table>

**CAN K-Line**
Purpose of the Services

- Diagnostic and Communication Management
- Data Transmission
- Stored Data Transmission
- Input / Output Control
- Remote Activation of Routine
- Upload / Download
- CARB / OBD / EOBD / KOBD

KWP2000

StartDiagnosticSession (0x10)
- Sets diagnostic session: this session determines the availability of other services
- Prepares jump to other SW parts of the ECU
- Over K-Line: change Baud-rate

StopDiagnosticSession (0x20)

UDS

DiagnosticSessionControl (0x10)
- Sets diagnostic session: this session determines the availability of other services
- Prepares jump to other SW parts of the ECU
- May change timing values (not configurable)
- Enables or sets back SecurityAccess
- Can reactivate deactivated CAN communication, deactivated error storing
- Resets responseOnEvent
KWP2000
ECUReset (0x11)
- After checking preconditions restarts the ECU software
- Reset type may be hard or key-on-off

TesterPresent (0x3E)
- Keeps communication alive: avoid communication timeout

UDS
ECUReset (0x11)
- After checking preconditions restarts the ECU software
- Reset type may be hard, key-on-off, soft, enable / disable rapidPowerDown

TesterPresent (0x3E)
- Keeps communication alive: avoid communication timeout
KWP2000

**ReadECUIdentification (0xA)**
- The tester requests only a LocalID
- The ECU provides information about itself, e.g. SW and HW IDs, versions, other logistic data

**UDS**

**CommunicationControl (0x28)**
- Activate / Deactivate sending and receiving CAN Frames for/from other ECUs (vehicle internal communication)
- Increases bandwidth for flash programming; prevents CAN timeout errors
- Does not deactivate diagnostic CAN channel

**SecuredDataTransmission (0x84)**
- This service encapsulates another service
- Security SubLayer of the transmitter encodes the encapsulated service
- Security SubLayer of the receiver decodes the encapsulated service
UDS

ControlDTCSettings (0x85)
- Activate / Deactivate storing of errors into error memory
- Used at flash programming and development

ResponseOnEvent (0x86)
- Configures the ECU to send a response without a request in case of a defined event
- Permanent and temporally configuration

LinkControl (0x87)
- Configures communication speed, if the medium is configurable

Example: Starting of Flash Programming
Purpose of the Services

- Diagnostic and Communication Management
- Data Transmission
- Stored Data Transmission
- Input / Output Control
- Remote Activation of Routine
- Upload / Download
- CARB / OBD / EOBD / KOBD

KWP2000 and UDS

ReadMemoryByAddress (0x23)
- The tester requests a memory address and number of bytes
- The ECU – if authentication level allows – reads out the block
- The format of the request is different at KWP2000 and UDS

WriteMemoryByAddress (0x3D)
- The tester sends a memory address, and number of bytes and a data string (according to the number of bytes)
- The ECU writes the data string into its memory
- The format of the request is different at KWP2000 and UDS
KWP2000

ReadDataByLocalIdentification (0x21)
- The tester requests one or more LocalIDs (1 byte long ID)
- A localID identifies a part of memory (not necessarily in one block)
- The ECU – if authentication level allows – reads out the block
- These IDs identify mainly pre-defined memory parts, but dynamic definition is also allowed
- A localID must not be unique within a vehicle
- LocalID can be read out periodically

WriteDataByLocalIdentification (0x3B)
- The tester sends a localID (1 byte long ID) and a data string
- The ECU writes the data string into its memory

SetDataRate (0x26)
- If any data is requested periodically, the repetition of the response can be fast, medium or slow
- This message defines, what does “fast”, “medium” and “slow” means
KWP2000

**ReadDataByCommonID (0x22)**
- Similar to `readDataByLocalID (0x21)`
- CommonID is 2 bytes long, but unique or means the same for more ECUs

**WriteDataByCommonID (0x2E)**
- Similar to `writeDataByLocalID (0x3B)`
- CommonID is 2 bytes long, but unique or means the same for more ECUs

UDS

**ReadDataByID (0x22)**

**ReadScalingDataID (0x2D)**
- Descendant of KWP2000 0x1A, 0x21 and 0x22
- The same as KWP2000 0x22, but periodical read not supported

**WriteDataByID (0x2E)**
- The same as KWP2000 0x2E

KWP2000

**DynamicallyDefineLocalID (0x2D)**
- The tester requests one or more memory addresses and length, AND/OR LocalIDs, AND/OR CommonIDs
- The ECU orders the requested memory parts to a dynamic LocalID
- By requesting that dynamic ID internal data can be read out with low CPU overload
- DynamicallyDefinedLocalID can be erased

UDS

**DynamicallyDefineDataID (0x2D)**
- The tester requests one or more memory addresses and length, AND/OR dataIDs
- The ECU orders the requested memory parts to a dynamic DataID
- By requesting that dynamic ID internal data can be read out with low CPU overload
- DynamicallyDefinedLocalID can be erased
**UDS**

ReadDataByPeriodicID (0x2D)

- Similar to ReadDataByID
- The response is periodical

---

**Purpose of the Services**

- Diagnostic and Communication Management
- Data Transmission
- Stored Data Transmission
- Input / Output Control
- Remote Activation of Routine
- Upload / Download
- CARB / OBD / EOBD / KOBD
Onboard Diagnostic

- **Input Signal**
  - Signal Reliable?
    - Yes
    - No
    - *Inc. counter* → *Dec. counter*
  - Cnt. threshold reached?
    - Yes
    - No

**Signal is analog or digital**
- Fulfills filter criteria; open/short-circuited, timeout

**Beside the error code the type of the error, the actual status; and actual value of some related signals are also stored**

---

KWP2000

- **ReadDiagnostic Trouble Code (0x13)**
- **ReadDiagnostic Trouble Code By Status (0x18)**
- **ReadStatus Of Diagnostic Code (0x17)**
- **ReadFreezeFrame Data (0x12)**
  - Error memory queries, by different filtering criteria

UDS

- **ReadDTC Information (0x19)**
  - Error memory queries, by different filtering criteria
  - The filtering criteria is sent by the sub-function of the service
KWP2000 and UDS

ClearDiagnosticInformation (0x14)

- Clears one, a group of, or all the stored errors from error memory

Purpose of the Services

- Diagnostic and Communication Management
- Data Transmission
- Stored Data Transmission
- Input / Output Control
- Remote Activation of Routine
- Upload / Download
- CARB / OBD / EOBD / KOBD
KWP2000
InputOutputControlByLocalID (0x30)
InputOutputControlByCommonID (0x2F)
  Can directly control the actuator signals
  Access to actuators is application-specific

UDS
InputOutputControlByIdentifier (0x2F)
  Can directly control the actuator signals
  Access to actuators is application-specific

Purpose of the Services
  ➔ Diagnostic and Communication Management
  ➔ Data Transmission
  ➔ Stored Data Transmission
  ➔ Input / Output Control
  ➔ Remote Activation of Routine
  ➔ Upload / Download
  ➔ CARB / OBD / EOBD / KOBD
KWP2000
StartRoutineByLocalID (0x31)
StartRoutineByAddress (0x38)
StopRoutineByLocalID (0x32)
StopRoutineByAddress (0x39)
RequestRoutineResultByLocalID (0x33)
RequestRoutineResultByAddress (0x3A)

→ Controls routines, as erasing memory, calculate checksum, testing routines, etc

UDS
RoutineControl (0x31)

→ Controls routines, as erasing memory, calculate checksum, testing routines, etc

→ Control option is at the sub-function

Purpose of the Services

→ Diagnostic and Communication Management
→ Data Transmission
→ Stored Data Transmission
→ Input / Output Control
→ Remote Activation of Routine
→ Upload / Download
→ CARB / OBD / EOBD / KOBD
KWP2000 and UDS

RequestDownload (0x34)
- The tester specifies an address and a length (not one byte), but no data
- Specifies compressing methods
- The ECU starts a downloading session

RequestUpload (0x35)
- The same as RequestDownload, but initializes uploading session

TransferData (0x36)
- Transfers the data bytes, with a help of sequence number, until the required amount of data is reached

RequestTransferExit (0x37)
- Terminates downloading / uploading

Questions?