**SEDAP-Express  
Interface Control Document (ICD)**

1. **Scope**

SEDAP-Express is an exceptionally fast path to integrate new applications, sensors, effectors or other similar things into the ecosystem of MESE. That's why it is intentionally kept simple and offers several technical ways of communication. Of course, this results in limitations, but in most cases where quick and easy integration is required, these are negligible. If increased demands arise later on, the “bigger” SEDAP API respective MESE interface can be used if necessary. SEDAP-Express is licensed under the “Simplified BSD License” (BSD-2-Clause). Therefore, there should be no problems using SEDAP-Express in commercial or non-commercial projects or integrating parts of the SEDAP-Express framework.

Everything you need for development and testing can be found on the Internet at http://SEDAP.Express.

1. **Glossary**

MESE = Military Expandable Software Environment

SEDAP = Safety critical Environment for Data exchange And Process scheduling

CSV = Comma-Separated-Values

SEC = SEDAP-Express-Connector

SECMockUp = Simulation of the real SEC

C2MockUp = Simulation of a C2 system with a simple map

SIDC = Symbol identification code (MIL-STD-2525C, STANAG2019)

ASCII = American Standard Code for Information Interchange – in this context the ISO-8859-1 table is meant

Base64 = Binary-to-text encoding scheme, which is using an alphabet of 64 characters

1. **General connection attributes**
2. **Common conventions**

* Basic format is CSV using ; (0x3B), with \n (0x10) terminated
* Elements of lists are separated by # (0x23)
* The messages are human-reable and using the ASCII-table
* (Binary)Data which possibly contains a special character (0x10, 0x23, 0x3B) has to be encoded with Base64
* Unknown/Invalid values must not be transmitted, the respective field will be left empty
* If there are only ; characters left in the message, these could be cut off
* Support for IPv4 or IPv6 (except for serial connection)
* SEC/SECMockUp/Applications can send and receive at any time
* Application shall send heartbeat message not more often than with 1Hz (+-100ms), but can vary if it is required
* SEC/SECMockUp answers heartbeat also with a heartbeat message (see chapter IV.2.8)

1. **TCP Connection**

* Standard port 50000, but customizable
* SEC/SECMockUp = Server (1)
* Application = Client (n)

1. **UDP-Connection**

* Standard port 50000, but customizable
* Support for Uni-, Broad- or Multicast mode
* Standard Multicast-IPv4-Address is 228.2.19.80
* Standard Multicast-IPv6-Address is ff02:8:2:19:80::1
* Multiple messages per UDP-packet possible
* SEC/SECMockUp answers heartbeat message with UDP-Unicast (see chapter IV.2.8)

1. **Serial connection**

* Standard 115200-8-N-1
* Full-Duplex is preferred
* Half-Duplex/Simplex without acknowledge requests are allowed

1. **REST-API**

* Standard HTTP 80 or HTTPS 443, but customizable
* The REST-API is described in an OpenAPI 3.1 schema file, which can be also found in chapter IV.3

1. **Data Exchange between SEC/SECMockUp and client applications**
2. **General**
   1. **TCP-/UDP-/Serial connections**

On principle, messages have a CSV structure with a common header:

<Name>;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>;<Content>

|  |  |
| --- | --- |
| <Name> | Defines the purpose of a message. Sometimes it’s so-called topic. |
| <Number> | This is a hexadecimal string representation of an 8-bit sequential number of a concrete message type that starts again with zero after reaching 0xFF. |
| <Time> | A hexadecimal string representation of a 64-bit Unix time stamp with milliseconds. |
| <Sender> | This field specify the original sender of the message by hexadecimal string representation of a 16-bit unsigned integer. This field won’t be changed, even if a message has been forwarded. This number can be chosen randomly by the participants themselves or permanently assigned by a responsible institution when preparing a specific use/network. |
| <Classification> | Describes the classification or security level of the content. Possible values are P=public, U=unclassified, R=restricted, C=confidential, S=secret, T=top secret |
| <Acknowledgement> | TRUE=request an acknowledgement, FALSE/Nothing=No acknowledgement |
| <Content> | Content of the message, depending on the message purpose. |

* 1. **REST-API connection**

If the REST-API shall be used, it’s preferred to also use the provided OpenAPI 3.1 schema file and the generated code which either comes with the SEDAP-Express SDK or has been generated by yourself. You can find the schema file in chapter IV.3 or on http://sepap.express.

1. **Messages**

This is the list of all so-far available messages and their structures and content including some samples. All units of measurement are generally given in SI-units, but there are deviations where this makes sense due to the usual range of values. In the following the used units will be given within square brackets for all message-descriptions. The altitude is the altitude above sea-level. A value of zero means exactly on ground, if the position is on land. Latitude and longitude are in decimal degrees, while positive values means north and east respective negative values south and west. Relative position values are defined this way, that the x-axis points to the west direction, y-axis points to the north and the z-axis is equal to the height above the unit. Speed and course are meant to be relative to ground. Course and heading have a range from zero to 359,999 and are relative to geographic north or zero degree. In general, all values are mandatory. Optional parameters are marked with (opt).

* 1. **OWNUNIT**

*Description:* Positional, kinematic and identification data of the own (sent by the client) or host (sent by the SEC) unit. If a client is sending this message, it will be converted to a contact and sent into the MESE network.

*Structure:* OWNUNIT;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>(opt);<Latitude>[°];<Longitude>[°];  
<Altitude>[m];Speed[m/s];Course[°];Heading[°];Roll[°];Pitch[°];Name;SIDC (MIL-STD-2525C)

*Sample 1:* OWNUNIT;5E;661D4410;66A3;R;;53.32;8.11;0;5.5;21;22;;;FGS Bayern;sfspfclff------

*Sample 2:* OWNUNIT;5E;661D4410;66A3;R;TRUE;42.32;-123.11;10000;50.23;297;;;33.3;-0.15;sfapmf---------

* 1. **CONTACT**

*Description:* Positional, kinematic and identification data of a contact. For example, this message would be used by a sensor to report a contact it recognized. In return this message would be used to receive the tactical picture from the MESE network.

*Structure:* CONTACT;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>(opt);<ContactID>;<DeleteFlag>;  
<Latitude>[°];<Longitude>[°];<Altitude>[m];<relative X-Distance>[m];<relative Y-Distance>[m];<relative Z-Distance>[m];  
<Speed>[m/s];<Course>[°];<Heading>[°];<Roll>[°];<Pitch>[°];<Name>;<SIDC>

|  |  |  |
| --- | --- | --- |
| ContactID | Number>0 | A positive identification unique number of the contact chosen by the sender of this message |
| DeleteFlag | TRUE | Contact has to be removed |
| FALSE | Contact is current |

*Sample 1:* CONTACT;5E;661D4410;66A3;R;;100;FALSE;53.32;8.11;0;5.5;21;22;;;FGS Bayern;sfspfclff------

*Sample 2:* CONTACT;5F;661D5420;66A3;U;;101;FALSE;36.32;12.11;2000;44;331;;Unknown   
*Sample 3:* CONTACT;60;661B7410;66A3;S;TRUE;102;TRUE

* 1. **EMISSION**

*Description:* Positional, attributes and identification data of an electro-magnetic, optical or acoustic emission.

*Structure:* EMISSON;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>(opt);<EmissionID>;<DeleteFlag>;  
<SensorLatitude>[°];<SensorLongitude>[°];<SensorAltitude>[m];<EmitterLatitude>[°];<EmitterLongitude>[°];<EmitterAltitude>[m];  
<Frequency[Hz]>;<Bandwidth[Hz]>;<Power[db(A)]>;<FreqAgility>;<PRFAgility>;<Function>;<SpotNumber>;<SIDC>

|  |  |  |
| --- | --- | --- |
| EmissionID | Number>0 | A positive identification unique number of the emission chosen by the sender of this message. This number should also be unique in terms of contact numbers. |
| DeleteFlag | TRUE | Emission has to be removed |
| FALSE | Emission is current |
| FreqAgility | 0 | Stable\_Fixed |
|  | 1 | Agile |
|  | 2 | Periodic |
|  | 3 | Hopper |
|  | 4 | Batch hopper |
|  | 5 | Unknown |
| PRFAgility | 0 | Fixed periodic |
|  | 1 | Staggered |
|  | 2 | Jittered |
|  | 3 | Wobbulated |
|  | 4 | Sliding |
|  | 5 | Dwell switch |
|  | 6 | UnknownPRF |
|  | 7 | CW |
| Function | 0 | Unknown |
|  | 1 | Esm\_Beacon/Transponder |
|  | 2 | Esm\_Navigation |
|  | 3 | Esm\_Voice\_Communication |
|  | 4 | Esm\_Data\_Communication |
|  | 5 | Esm\_Radar |
|  | 6 | Esm\_Iff |
|  | 7 | Esm\_Guidance |
|  | 8 | Esm\_Weapon |
|  | 9 | Esm\_Jammer |
|  | 10 | Esm\_Natural |
|  | 11 | Acoustic\_Object |
|  | 12 | Acoustic\_Submarine |
|  | 13 | Acoustic\_Variable\_Depth\_Sonar |
|  | 14 | Acoustic\_Array\_Sonar |
|  | 15 | Acoustic\_Active\_Sonar |
|  | 16 | Acoustic\_Torpedo\_Sonar |
|  | 17 | Acoustic\_Buoys\_Sonar |
|  | 18 | Acoustic\_Decoy\_Signal |
|  | 19 | Acoustic\_Hit\_Noise |
|  | 20 | Acoustic\_Propeller\_Noise |
|  | 21 | Acoustic\_Underwater\_Telephone |
|  | 22 | Acoustic\_Communication |
|  | 23 | Acoustic\_Noise |
|  | 24 | Laser\_Range\_Finder |
|  | 25 | Laser\_Designator |
|  | 26 | Laser\_Beam\_Rider |
|  | 27 | Laser\_Dazzler |
|  | 28 | Laser\_Lidar |

*Sample 1:* EMISSION;5E;661D4410;66A3;R;;100;;53.32;8.11;0;54.51;8.15;0;8725000;20000;3;0;2;6

*Sample 2:* EMISSION;5F;661D5410;66A3;R;;101;;54.86;9.32;0;52.12;9.80;50;25725000;40000;1,5;2;0;6;10233;sngpesr--------

* 1. **METEO**

*Description:* Metrological data of the environment.

*Structure:* METEO;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>(opt);<SpeedThroughWater>[m/s];  
<WaterSpeed>[m/s];<WaterDirection>[°];<WaterTemperature>[°C];<WaterDepth>[m];<AirTemperature>[°C];<DewPoint>[°C];  
<HumidityRel>[%];<Pressure>[hPa];<WindSpeed>[m/s];<WindDirection>[°];<Visibility>[km];<CloudHeight>[m];<CloudCover>[%]

*Sample:* METEO;AC;661D44C0;74BE;U;;15.4;15.5;;;10.2;72;20.3;;55;1005;25;;;2500;33

* 1. **TEXT**

*Description:* Human readable textual data. This could be an alert message, but also a simple text message for chatting.

*Structure:* TEXT;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>(opt);<Type>;<Text>;<Recipient>(opt)

|  |  |  |
| --- | --- | --- |
| Type | 0 | Alert |
|  | 1 | Warning |
|  | 2 | Notice |
|  | 3 | Chat |
| Text | ASCII | Free text of the message |
| Recipient | HexString | This field specifies the recipient for the message by hexadecimal string representation of a 16-bit unsigned integer, as explained in table form chapter IV.1.1 |

*Sample 1:* TEXT;D3;661D44D2;324E;S;TRUE;0;"This is an alert!"

*Sample 2:* TEXT;D4;661D458E;324E;S;TRUE;1;"This is a warning!"

*Sample 3:* TEXT;D5;661D6565;324E;S;;2;"This is a notice!"

*Sample 4:* TEXT;D6;661D7032;324E;S;;3;"This is a chat message!";E4F1

* 1. **COMMAND**

*Description:* Command for one specific or all possible recipients.

*Structure:* COMMAND;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>(opt);<Recipient>;<CmdType>;  
<additional cmd-dependent parameters>\*

|  |  |  |
| --- | --- | --- |
| Recipient | HexString | This field specify the recipient of the command by hexadecimal string representation of a 16-bit unsigned integer. |
| CmdType | 0 | Power off device |
| 1 | Restart device |
|  | 2 | Set device into standby, Optional Unix time stamp for wake up |
|  | 3 | Wake up device |
|  | 4 | Sync time: <IP of a NTP server>(opt) |
|  | 5 | Send status |
|  | 6 | Move: <Latitude>[°];<Longitude>[°];<Altitude>[m](opt) |
|  | 7 | Scan Area: <Latitude1>[°];<Longitude1>[°];<Latitude2>[°];<Longitude2>[°];<angle>(opt) |
|  | 8 | Action, Kind of action (optional, self-defined for specific use case) |
|  | 9 | Take photo: <Number of camera>(opt);<Camera mode>(opt) |
|  | 10 | Switch on video stream: <Number of camera>(opt);<Camera mode>(opt) |
| 11 | Switch off video stream: <Number of camera>(opt) |
|  | 12 | Start engagement: <contactID>(opt) |
|  | 13 | Stop engagement: <contactID>(opt) |

*Sample 1:* COMMAND;27;661D44C0;E4B3;C;TRUE;AB49;2

*Sample 2:* COMMAND;27;661D44C0;E4B3;C;TRUE;AB49;12;1000

* 1. **GRAPHIC**

*Description:* Define graphical plans likes polygons, squares or routes

*Structure:* GRAPHIC;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>(opt);<GraphicType>;<LineWidth>;<LineColor>;<Annotation>;  
<additional GraphicType-dependent parameters>\*

|  |  |  |
| --- | --- | --- |
| GraphicType | 0 | Point: <Latitude>[°];<Longitude>[°];<Altitude>[m](opt) |
| 1 | Path: <Latitude>[°],<Longitude>[°],<altitude>[m](opt) # … |
|  | 3 | Polygon: <Latitude>[°],<Longitude>[°],<Altitude>[m](opt) # … |
|  | 4 | Rectangle: <Rotation\_Angle>[°];<Latitude1>[°],<Longitude1>[°],<Altitude1>[m](opt)#<Latitude2>[°], <Longitude2>[°],<Altitude2>[m](opt) |
|  | 5 | Square: (t.b.d.) |
|  | 6 | Parallelogram: (t.b.d.) |
|  | 7 | Trapezium: (t.b.d.) |
|  | 8 | Circle: <radius>[m];<Latitude>[°];<Longitude>[°];<Altitude>[m](opt) |
|  | 9 | Ellipse: <radius-X>[m];<radius-Y>[m];<CenterLatitude>[°];<CenterLongitude>[°];<CenterAltitude>[m](opt) |
|  | 10 | Block: (t.b.d.) |
|  | 11 | Sphere: (t.b.d.) |
|  | 12 | Cone: (t.b.d.) |
|  | 13 | Pyramid: (t.b.d.) |
|  | 14 | Ellipsoid: <X-Radius>[m];<Y-Radius>[m];<Z-Radius>[m];<Center\_Latitude>[°];<Center\_Longitude>[°]; <Center\_Altitude>[m](opt) |
| LineWidth | => 1 | Width of the line or the point |
| LineColor | RGB | Color of the line or the point in Web notation 800000 for a darker red |
| Annotation | ASCII | Text for an annotation to this graphic |

*Sample 1:* GRAPHIC;77;661D64C0;910E;U;;0;1;FF0000;StartPoint;54.23;12.86

*Sample 2:* GRAPHIC;77;661D64C0;910E;U;;1;1;808080;Transit;54.23,12.86#54.30,12.9#54.55,13.3

*Sample 3:* GRAPHIC;79;661D62C0;910E;U;;8;1;FF8000;Area A;10000;53.43;9.45

* 1. **STATUS**

*Description:* This message offers the possibility to check the connection, which is primarily important, if you are using UDP or serial connection. It should not be sent more often than 1Hz. Nevertheless, if it is needed – one can use a faster repetition. The receiver field is optional and can be one single recipient or a list of recipients.

*Structure:* STATUS;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>(opt);<TecStatus>;<OpsStatus>;  
<FuelLevel>;<BatterieLevel>;<FreeText>

|  |  |  |
| --- | --- | --- |
| TecStatus | 0 | Not operational |
|  | 1 | Initializing |
|  | 2 | Degraded |
|  | 3 | Partly operational |
|  | 4 | Fully operational |
|  | 5 | Fault |
| OpsStatus | 0 | Not operational |
|  | 1 | Initializing |
|  | 2 | Degraded |
|  | 3 | Partly operational |
|  | 4 | Fully operational |
| FuelLevel | % | Relative remaining fuel capacity |
| BatterieLevel | % | Relative remaining batterie capacity |
| FreeText | ASCII | Human readable free text description of the status |

*Sample 1:* STATUS;15;661D44C0;75DA;U;;4;2;;50;Not ready

*Sample 2:* STATUS;16;661D64C0;129E;R;;2;2;0;;Out of fuel!

* 1. **ACKNOWLEDGE**

*Description:* If a client or the SEC requested an acknowledge of a packet one has to use this this message. The acknowledgement flag is fixed set to FALSE. The awaiting client or SEC have to wait maximal 2 seconds before resending the original message with set acknowledgement flag.

*Structure:* ACKNOWLEDGE;<Number>(opt);<Time>(opt);<Sender>;<Classification>;FALSE;<Receiver>;<Name>;<Counter>

|  |  |  |
| --- | --- | --- |
| Recipient | HexString | This field specify the recipient of the acknowledge by hexadecimal string representation of a 16-bit unsigned integer |
| Name | ASCII | The name of the message which should be acknowledged |
| Counter | Number | The number of the message which should be acknowledged |

*Sample:* ACKNOWLEDGE;18;661D64C0;129E;R;FE2A;COMMAND;0D31

* 1. **HEARTBEAT**

*Description:* This message offers the possibility to check the connection, which is primarily important, if you are using UDP or serial connection. It should not be sent more often than 1Hz. Nevertheless, if it is needed – one can use a faster repetition. The receiver field is optional and can be one single recipient or a list of more than one recipient. If no recipient is provided than all possible receivers in the network/serial net are addressed. A heartbeat message has an empty acknowledgement flag, cause you cannot request one for it. Besides this, the acknowledgement flag is fixed set to FALSE.

*Structure:* HEARTBEAT;<Number>(opt);<Time>(opt);<Sender>(opt);<Classification>;FALSE;<Receiver>(opt);

|  |  |  |
| --- | --- | --- |
| Recipient | HexString | This field specify the recipient of the command by hexadecimal string representation of a 16-bit unsigned integer. |

*Sample 1:* HEARTBEAT;42;661D5420;89AD;U;;FE2A

*Sample 2:* HEARTBEAT;43;;1022

*Sample 3:* HEARTBEAT;43;

*Sample 4:* HEARTBEAT

* 1. **GENERIC**

*Description:* This message is an empty container for transporting any kind of data. It has to be defined in the respective case. For example, one can use it to exchange the original MESE/SEDAP messages or other propriety protocol data. In the last case you have to use any other self-defined type.

*Structure:* GENERIC;<Number>;<Time>;<Sender>;<Classification>;<Acknowledgement>(opt);<ContentType>;<EncodingFlag>;<Content>

|  |  |  |
| --- | --- | --- |
| ContentType | SEDAP | Content is an original MESE message |
| \* | Self-defined ASCII string |
| EncodingType | TRUE | Content is Base64 encoded |
| FALSE | Content is NOT encoded |
| Content |  | Any content in printable ASCII or Base64 encoded |

*Sample 1:* GENERIC;5E;661D4410;66A3;R;;SEDAP;FALSE;

*Sample 2:* GENERIC;5E;661D4410;66A3;R;TRUE;SEDAP;TRUE;U2FtcGxlIGJpbmFyeSBwcm90b2NvbCAtIEdyZWV0aW5ncyA7KQ==

*Sample 3:* GENERIC;5E;661D4410;66A3;R;;PROJECT1-RADAR-NMEA;FALSE;$RATTM,11,11.4,13.6,T,7.0,20.0,T,0.0,0.0,N,,Q,,154125.82,A,\*17

1. **SEDAP-Express JSON-Schema**

(t.b.d.)