

## Themenhandbuch

SAE J1939-STX-API

60880843

We automate your success.

Artikelnummer: 60880843

Version 1.03

09. November 2021 / Printed in Germany

Dieses Dokument hat die Jetter AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt.

Bei Änderungen, Weiterentwicklungen oder Erweiterungen bereits zur Verfügung gestellter Produkte wird ein überarbeitetes Dokument nur beigefügt, sofern dies gesetzlich vorgeschrieben oder von der Jetter AG für sinnvoll erachtet wird. Die Jetter AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.

Die im Dokument aufgeführten Logos, Bezeichnungen und Produktnamen sind geschützte Marken der Jetter AG, der mit ihr verbundenen Unternehmen oder anderer Inhaber und dürfen nicht ohne Einwilligung des jeweiligen Inhabers verwendet werden.

---

# Inhaltsverzeichnis

---

<b>1</b>	<b>SAE J1939</b>	<b>5</b>
	J1939-Layerbeschreibung .....	6
<b>2</b>	<b>SAE J1939-STX-API</b>	<b>9</b>
	STX-Funktion SAEJ1939Init().....	10
	STX-Funktion SAEJ1939Exit().....	11
	STX-Funktion SAEJ1939SetSA() .....	13
	STX-Funktion SAEJ1939GetSA() .....	14
	STX-Funktion SAEJ1939AddRx().....	15
	STX-Funktion SAEJ1939AddTx() .....	18
	STX-Funktion SAEJ1939RequestPGN() .....	21
	STX-Funktion SAEJ1939GetDM1() .....	24
	STX-Funktion SAEJ1939GetDM2() .....	27
	STX-Funktion SAEJ1939SetSPNConversion() .....	30
	STX-Funktion SAEJ1939GetSPNConversion().....	31



---

# 1 SAE J1939

---

**Einleitung**

Dieses Kapitel beschreibt den J1939-Layer und den Aufbau einer J1939-Nachricht.

---

**Inhalt**

<b>Thema</b>	<b>Seite</b>
J1939-Layerbeschreibung .....	6

## J1939-Layerbeschreibung

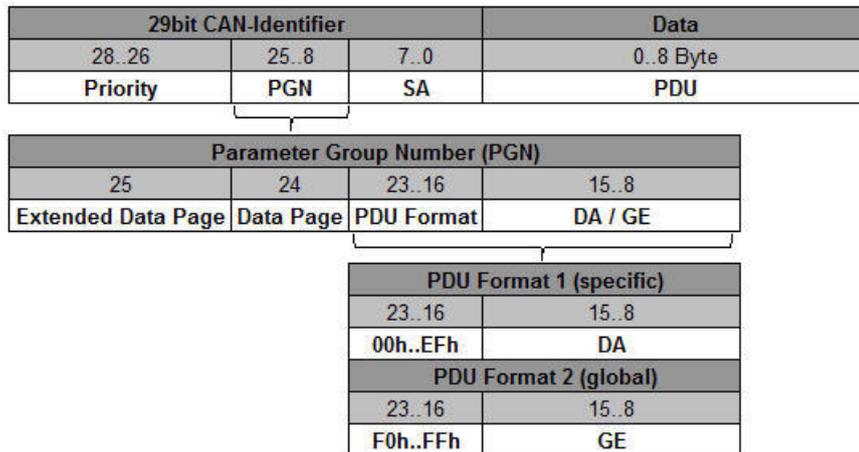
### Physikalischer Layer

Das SAE J1939-Protokoll setzt auf den CAN-Bus auf und verwendet als physikalischen Layer den ISO 11898 CAN-High-Speed mit folgenden Eigenschaften:

- Baudrate 250 kBit
- Max. 30 Knoten
- 2-Draht-Leitung mit einem Abschlusswiderstand von 120 Ω
- Buslänge ohne Stichleitung 40 m
- Länge von Stichleitungen max. 1 m

### Aufbau einer J1939-Nachricht

Die folgende Abbildung zeigt den Aufbau einer J1939-Nachricht:



Abkürzung	Bezeichnung (Bedeutung)
DA	Destination Address (Zieladresse)
GE	Group Extensions (Gruppenerweiterungen)
PDU	Protocol Data Unit (Protokolldateneinheit)
PGN	Parameter Group Number (Parametergruppennummer)
SA	Source Address (Quelladresse)

### Aufbau des Identifiers

Das nachstehende Beispiel zeigt den Aufbau eines Identifiers in hexadezimal: 0x18FEE927

Teil des Identifiers	Beschreibung
27	Source Address
FEE9	Parameter Group Number
18	Priority

### Bedeutung der SPN - Suspect Parameter Number

Die SPN ist eine in der SAE J1939-Norm definierte Nummer, die einen einzelnen Parameter (z. B: Motordrehzahl) als genormte Nachricht beinhaltet. Es folgt ein Beispiel für Parameter einer SPN:

**spn110 - Engine Coolant Temperature** - Temperatur der Motorkühlflüssigkeit im Motorkühlsystem.

Data Length:	1 Byte
Resolution:	1 °C/Bit , -40 °C Offset
Data Range:	-40 bis 210 °C
Type:	Measured (gemessen)
Suspect Parameter Number:	110
Vehicle Application Layer - J1939-71 (J1939-71 Rev. Aug2002)	
Parameter Group Number:	[65262]

### Bedeutung der PGN - Parameter Group Num- ber

Die PGN ist eine in der SAE J1939-Norm definierte Nummer, die mehrere SPNs zu einer sinnvollen Gruppe zusammenfügt. Die PGN ist Teil des CAN-Identifiers. Die 8-Byte-Daten (PDU) beinhalten die Werte der einzelnen SPN.

Es folgt ein Beispiel für die PGN 65262 (0xFEEE):

#### PGN 65262 Engine Temperature 1 - ET1

Teil der PGN	Wert	Bemerkung
Transmission Repetition Rate	1 s	
Data Length	8	
Extended Data Page	0	
Data Page	0	
PDU Format	254	
PDU Specific	238	PGN Supporting Information
Default Priority	6	
Parameter Group Number	65262	in Hex: 0xFEEE

Startposition	Länge	Parametername	SPN
1	1 Byte	Engine Coolant Temperature	110
2	1 Byte	Engine Fuel Temperature 1	174
3 - 4	2 Bytes	Engine Oil Temperature 1	175
5 - 6	2 Bytes	Engine Turbocharger Oil Temperature	176
7	1 Byte	Engine Intercooler Temperature	52
8	1 Byte	Engine Intercooler Thermostat Opening	1134



## 2 SAE J1939-STX-API

<b>Einleitung</b>	Dieses Kapitel beschreibt die STX-Funktionen der SAE J1939-STX-API.																								
<b>Der SAE J1939-Standard</b>	SAE J1939 ist ein offener Standard für die Vernetzung und Kommunikation im Nutzfahrzeugbereich. Schwerpunkt der Anwendung ist die Vernetzung von Antriebsstrang und Chassis. Das Protokoll J1939 stammt von der internationalen Society of Automotive Engineers (SAE) und arbeitet auf dem Physical Layer mit CAN-Highspeed nach ISO 11898.																								
<b>Anwendung</b>	Diese STX-Funktionen werden in der Kommunikation zwischen einer SAE J1939-fähigen Steuerung und anderen Steuergeräten im Fahrzeug angewendet. Es werden in der Regel Motordaten, z. B. Motordrehzahl, Geschwindigkeit oder Kühlwassertemperatur, ausgelesen und auf dem Display angezeigt.																								
<b>Dokumente</b>	Die wichtigsten SAE J1939-Spezifikationen sind: <ul style="list-style-type: none"> <li>▪ J1939-11 - Informationen zum Physical Layer</li> <li>▪ J1939-21 - Informationen zum Data Link Layer</li> <li>▪ J1939-71 - Informationen zum Application Layer Fahrzeuge</li> <li>▪ J1939-73 - Informationen zum Application Layer Bereich Diagnose</li> <li>▪ J1939-81 - Netzwerkmanagement</li> </ul>																								
<b>Geräte</b>	Die folgenden Geräte haben die Funktion der SAE J1939-STX-API: <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="text-align: center;">Art des Geräts</th> <th style="text-align: center;">Bezeichnung</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Steuerung</td> <td>JCM-350-E01/E02, JCM-350-E03, JCM-620 JCM-52x, JCM-511, LMC-x, JCM-63x</td> </tr> <tr> <td style="text-align: center;">Bediengerät</td> <td>BTM07, BTM09(B), BTM010, BTM011(B), BTM012 JVM-104, JVM-407(B), JVM-507(B), JVM-604(B)</td> </tr> </tbody> </table>	Art des Geräts	Bezeichnung	Steuerung	JCM-350-E01/E02, JCM-350-E03, JCM-620 JCM-52x, JCM-511, LMC-x, JCM-63x	Bediengerät	BTM07, BTM09(B), BTM010, BTM011(B), BTM012 JVM-104, JVM-407(B), JVM-507(B), JVM-604(B)																		
Art des Geräts	Bezeichnung																								
Steuerung	JCM-350-E01/E02, JCM-350-E03, JCM-620 JCM-52x, JCM-511, LMC-x, JCM-63x																								
Bediengerät	BTM07, BTM09(B), BTM010, BTM011(B), BTM012 JVM-104, JVM-407(B), JVM-507(B), JVM-604(B)																								
<b>Inhalt</b>	<table border="0" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Thema</th> <th style="text-align: right;">Seite</th> </tr> </thead> <tbody> <tr> <td>STX-Funktion SAEJ1939Init() .....</td> <td style="text-align: right;">10</td> </tr> <tr> <td>STX-Funktion SAEJ1939Exit() .....</td> <td style="text-align: right;">11</td> </tr> <tr> <td>STX-Funktion SAEJ1939SetSA() .....</td> <td style="text-align: right;">13</td> </tr> <tr> <td>STX-Funktion SAEJ1939GetSA() .....</td> <td style="text-align: right;">14</td> </tr> <tr> <td>STX-Funktion SAEJ1939AddRx() .....</td> <td style="text-align: right;">15</td> </tr> <tr> <td>STX-Funktion SAEJ1939AddTx() .....</td> <td style="text-align: right;">18</td> </tr> <tr> <td>STX-Funktion SAEJ1939RequestPGN() .....</td> <td style="text-align: right;">21</td> </tr> <tr> <td>STX-Funktion SAEJ1939GetDM1() .....</td> <td style="text-align: right;">24</td> </tr> <tr> <td>STX-Funktion SAEJ1939GetDM2() .....</td> <td style="text-align: right;">27</td> </tr> <tr> <td>STX-Funktion SAEJ1939SetSPNConversion() .....</td> <td style="text-align: right;">30</td> </tr> <tr> <td>STX-Funktion SAEJ1939GetSPNConversion() .....</td> <td style="text-align: right;">31</td> </tr> </tbody> </table>	Thema	Seite	STX-Funktion SAEJ1939Init() .....	10	STX-Funktion SAEJ1939Exit() .....	11	STX-Funktion SAEJ1939SetSA() .....	13	STX-Funktion SAEJ1939GetSA() .....	14	STX-Funktion SAEJ1939AddRx() .....	15	STX-Funktion SAEJ1939AddTx() .....	18	STX-Funktion SAEJ1939RequestPGN() .....	21	STX-Funktion SAEJ1939GetDM1() .....	24	STX-Funktion SAEJ1939GetDM2() .....	27	STX-Funktion SAEJ1939SetSPNConversion() .....	30	STX-Funktion SAEJ1939GetSPNConversion() .....	31
Thema	Seite																								
STX-Funktion SAEJ1939Init() .....	10																								
STX-Funktion SAEJ1939Exit() .....	11																								
STX-Funktion SAEJ1939SetSA() .....	13																								
STX-Funktion SAEJ1939GetSA() .....	14																								
STX-Funktion SAEJ1939AddRx() .....	15																								
STX-Funktion SAEJ1939AddTx() .....	18																								
STX-Funktion SAEJ1939RequestPGN() .....	21																								
STX-Funktion SAEJ1939GetDM1() .....	24																								
STX-Funktion SAEJ1939GetDM2() .....	27																								
STX-Funktion SAEJ1939SetSPNConversion() .....	30																								
STX-Funktion SAEJ1939GetSPNConversion() .....	31																								

---

## STX-Funktion SAEJ1939Init()

---

**Einleitung**

Mit dem Aufruf der Funktion `SAEJ1939Init()` wird einer der vorhandenen CAN-Busse (nicht CAN 0, wenn keine Ethernet-Schnittstelle vorhanden ist, da dieser für CANopen<sup>®</sup> reserviert) für das J1939-Protokoll initialisiert. Das Gerät hat ab dann die durch den Funktionsparameter `mySA` zugeteilte SA (Source Address). Es handelt sich dabei um die eigene Geräteadresse am Bus.

**Funktionsdeklaration**

```
Function SAEJ1939Init(
    CANNo: Int,
    mySA: Byte,
) : Int;
```

**Funktionsparameter**

Die Funktion `SAEJ1939Init()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNo	CAN-Kanalnummer	0/1 ... CANMAX
mySA	Eigene Source-Adresse	0 ... 253

**Rückgabewert**

Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

**Rückgabewert**

0	ok
-1	Fehler bei der Parameterüberprüfung
-3	Nicht genug Speicher für SAE J1939

**CANNo-Parameter**

Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit `CANNo = 0/1` ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen `CANMAX` ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

**Verwenden der Funktion**

Initialisieren des CAN-Busses 1. Das Gerät hat die SA 20 (0x14). Das Gerät kann jetzt Nachrichten mit der eingestellten SA (und nur diese Nachrichten) senden.

```
Result := SAEJ1939Init(1, 20);
```

**Address Claiming**

*Address Claiming* ist nicht implementiert.

---

## STX-Funktion SAEJ1939Exit()

**Einleitung** Mit dem Aufruf der Funktion `SAEJ1939Exit()` wird die initialisierte CAN-Schnittstelle geschlossen.

**Funktionsdeklaration**

```
Function SAEJ1939Exit (
    CANNNo: Int,
) : Int;
```

**Funktionsparameter** Die Funktion `SAEJ1939Exit()` hat die folgenden Parameter:

Parameter	Beschreibung	Wert
CANNNo	Nummer der CAN-Schnittstelle	0 ... CANMAX

**Rückgabewert** Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm:

Rückgabewert	
0	OK
-1	Fehler bei der Parameterüberprüfung

**CANNNo-Parameter** Dieser Parameter gibt die Nummer der CAN-Schnittstelle an. Die erste Schnittstelle wird mit `CANNNo = 0` ausgewählt. Die Anzahl der CAN-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der CAN-Schnittstellen `CANMAX` ist in den *technischen Daten* und der *Registerübersicht* der jeweiligen Betriebsanleitung erwähnt.

## 2 SAE J1939-STX-API

---

**Verwenden der Funktion** Schließen der CAN-Schnittstelle 0.

```
Result := SAEJ1939Exit(0);
```

---

**Funktionsweise** Die Funktion schließt die CAN-Schnittstelle für die übergebene Schnittstellennummer. Es werden keine PGNs mehr gesendet/empfangen. Soll die Schnittstelle im Anschluss wieder verwendet werden, muss SAEJ1939Init() aufgerufen werden. Im Anschluss müssen auch alle PGNs wieder erneut angemeldet werden (SAEJ1939AddRx und SAEJ1939AddTx).

---

**Verwandte Themen**

- STX-Funktion SAEJ1939Init()
-

## STX-Funktion SAEJ1939SetSA()

**Einleitung** Mit dem Aufruf der Funktion `SAEJ1939SetSA()` ändern Sie die eigene SA (Source Address) zur Laufzeit.

**Funktionsdeklaration**

```
Function SAEJ1939SetSA (
    CANNo: Int,
    mySA: Byte,
) : Int;
```

**Funktionsparameter** Die Funktion `SAEJ1939SetSA()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNo	CAN-Kanalnummer	0/1 ... CANMAX
mySA	Neue SA	0 ... 253

**Rückgabewert** Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

**Rückgabewert**

0	ok
-1	Fehler bei der Parameterüberprüfung

**CANNo-Parameter** Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit `CANNo = 0/1` ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen `CANMAX` ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

**Verwenden der Funktion** Die SA zur Laufzeit ändern.

```
Result := SAEJ1939SetSA(1, 20);
```

**Wichtiger Hinweis** Nachrichten werden sofort mit der neuen SA gesendet oder empfangen.

## STX-Funktion SAEJ1939GetSA()

---

**Einleitung** Mit dem Aufruf der Funktion `SAEJ1939GetSA()` kann die eigene SA (Source Address) festgestellt werden.

---

**Funktionsdeklaration**

```
Function SAEJ1939GetSA(  
    CANNo: Int,  
    ref mySA: Byte,  
) : Int;
```

---

**Funktionsparameter** Die Funktion `SAEJ1939GetSA()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNo	CAN-Kanalnummer	0/1 ... CANMAX
mySA	Aktuell eingestellte SA	0 ... 253

---

**Rückgabewert** Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

**Rückgabewert**

0	ok
-1	Fehler bei der Parameterüberprüfung

---

**CANNo-Parameter** Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit `CANNo = 0/1` ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen `CANMAX` ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

---

**Verwenden der Funktion** Es wird die aktuell eingestellte SA zurückgegeben.

```
Result := SAEJ1939SetSA(1, actual_SA);
```

---

## STX-Funktion SAEJ1939AddrRx()

### Einleitung

Mit dem Aufruf der Funktion `SAEJ1939AddrRx()` wird das Gerät aufgefordert, eine bestimmte Nachricht zu empfangen. Diese Nachricht wird von einem anderen Busteilnehmer gesendet. Die Adresse dieses Busteilnehmers wird als Parameter `bySA` dieser Funktion übergeben. Wenn die Nachricht nicht gesendet wird, bleibt der zuletzt empfangene Wert bestehen. Das zyklische Lesen erfolgt solange, bis die Funktion `SAEJ1939Init()` erneut aufgerufen wird.

### Funktionsdeklaration

```
Function SAEJ1939AddrRx (
    CANNo: Int,
    IPGN: Long,
    bySA: Byte,
    BytePos: Int,
    BitPos: Int,
    DataType: Int,
    DataLength: Int,
    const ref VarAddr,
    ref stJ1939: J1939Rx
    EventTime: Int,
    InhibitTime: Int,
) : Int;
```

### Funktionsparameter

Die Funktion `SAEJ1939AddrRx()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNo	CAN-Kanalnummer	0/1 ... CANMAX
IPGN	PGN Parameter Group Number	0 ... 0x3FFFF
bySA	Source Address vom Sender der Nachricht	0 ... 253
BytePos	Startposition des Bytes der zu empfangenden Daten	1 ... n
BitPos	Startposition des Bits der zu empfangenden Daten	1 ... 8
DataType	Datentyp der zu empfangenden Daten	1 ... 3, 10 ... 16
DataLength	Datengröße der globalen Variablen <code>VarAddr</code>	
VarAddr	Globale Variable, in die der empfangene Wert eingetragen wird	
TJ1939Rx	Kontrollstruktur	
EventTime	Zeitlicher Abstand zw. zwei Telegrammen (> <code>InhibitTime</code> )	Default-Wert: 1.000 ms
InhibitTime	Mindestabstand zwischen zwei empfangenen Telegrammen (< <code>EventTime</code> )	Default-Wert: 100 ms

### Rückgabewert

Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

---

#### Rückgabewert

---

0	ok
-1	Fehler bei der Parameterüberprüfung

---

### CANNo-Parameter

Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit CANNo = 0/1 ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen CANMAX ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

---

### DataType-Parameter

Die folgenden Datentypen sind möglich.

Byte-Typen	Bit-Typen	SAEJ1939
1	-	SAEJ1939_UNSIGNED8 SAEJ1939_BYTE
2	-	SAEJ1939_UNSIGNED16 SAEJ1939_WORD
4	-	SAEJ1939_UNSIGNED32 SAEJ1939_DWORD
n	-	SAEJ1939_STRING
-	1	SAEJ1939_1BIT
-	2	SAEJ1939_2BIT
-	3	SAEJ1939_3BIT
-	4	SAEJ1939_4BIT
-	5	SAEJ1939_5BIT
-	6	SAEJ1939_6BIT
-	7	SAEJ1939_7BIT

---

### Kontrollstruktur TJ1939Rx

```
TJ1939Rx : Struct
// Status der empfangenen Nachricht
    byStatus      : Byte;
// Priorität der empfangenen Nachricht
    byPriority     : Byte;
End_Struct;
```

---

**Verwenden der Funktion**

```
Result := SAEJ1939AddRx (
    1,
    0xFEEE,
    0x00,
    2
    0
    SAEJ1939_BYTE,
    sizeof(var_Fueltemp),
    var_Fueltemp,
    struct_TJ1939Rx_EngineTemperatureTbl,
    1500,
    120);
```

**JetSym-STX-Programm**

Das Gerät mit der eigenen SA von 20 möchte die aktuelle Treibstofftemperatur empfangen und anzeigen. Die Parameter **InhibitTime** und **EventTime** werden beim Funktionsaufruf nicht explizit aufgeführt. In diesem Fall werden die Default-Werte verwendet. Die Steuerung, die die Treibstofftemperatur erfasst, hat die SA von 0. In der Praxis ist die Adresse der Steuerung aus der Dokumentation des Motorherstellers zu erfahren.

Die Treibstofftemperatur hat die SPN 174 und ist Bestandteil (Byte 2) der PGN 65262 Motortemperatur 1.

Var

```
bySAEJ1939Channel    : Byte;
own_Source_Address  : Byte;
```

// PGN 65262 Engine Temperature 1

```
Fueltemp             : Byte;
EngineTemperatureTbl : J1939Rx;
```

End\_Var;

Task main autorun

// Initialisierung CAN 1

```
bySAEJ1939Channel := 1;
own_Source_Address := 20;
SAEJ1939Init (bySAEJ1939Channel, own_Source_Address);
```

// Treibstofftemperatur empfangen

```
SAEJ1939AddRx (bySAEJ1939Channel, 65262, 0x00, 2, 1, SAEJ1939_BYTE,
    sizeof(Fueltemp), Fueltemp, EngineTemperatureTbl);
```

End\_Task;

**Anleitung des Motorherstellers**

Der Kunde erfährt die Daten (Priorität, PGN, SA und den Aufbau der Datenbytes) aus der Anleitung des Motorherstellers.

## STX-Funktion SAEJ1939AddTx()

### Einleitung

Mit dem Aufruf der Funktion `SAEJ1939AddTx()` wird das Gerät aufgefordert, eine bestimmte Nachricht zyklisch über den Bus zu senden.

Das zyklische Senden erfolgt solange, bis die Funktion `SAEJ1939Init()` erneut aufgerufen wird.

Die Daten werden nach Ablauf der Event-Time gesendet. Die Inhibit-Time wird bisher noch nicht berücksichtigt.

### Funktionsdeklaration

```
Function SAEJ1939AddTx (
    CANNNo: Int,
    IPGN: Long,
    BytePos: Int,
    BitPos: Int,
    dataType: Int,
    DataLength: Int,
    const ref VarAddr,
    ref stJ1939: J1939Tx
    EventTime: Int,
    InhibitTime: Int,
) : Int;
```

### Funktionsparameter

Die Funktion `SAEJ1939AddTx()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNNo	CAN-Kanalnummer	0/1 ... CANMAX
IPGN	PGN Parameter Group Number	0 ... 0x3FFFF
BytePos	Startposition des Bytes der zu sendenden Daten	1 ... n
BitPos	Startposition des Bits der zu sendenden Daten	1 ... 8
dataType	Datentyp der zu sendenden Daten	1 ... 3, 10 ... 16
DataLength	Datengröße der globalen Variablen VarAddr	
VarAddr	Globale Variable, in die der sendende Wert eingetragen wird	
J1939Tx	Kontrollstruktur	
EventTime	Zeitlicher Abstand zw. zwei Telegrammen (> InhibitTime)	Default-Wert: 1.000 ms
InhibitTime	Mindestabstand zwischen zwei empfangenen Telegrammen (< EventTime)	Default-Wert: 100 ms

**Rückgabewert**

Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

**Rückgabewert**

0	ok
-1	Fehler bei der Parameterüberprüfung

**CANNo-Parameter**

Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit CANNo = 0/1 ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen CANMAX ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

**DataType-Parameter**

Die folgenden Datentypen sind möglich.

Byte-Typen	Bit-Typen	SAEJ1939
1	-	SAEJ1939_UNSIGNED8 SAEJ1939_BYTE
2	-	SAEJ1939_UNSIGNED16 SAEJ1939_WORD
4	-	SAEJ1939_UNSIGNED32 SAEJ1939_DWORD
n	-	SAEJ1939_STRING
-	1	SAEJ1939_1BIT
-	2	SAEJ1939_2BIT
-	3	SAEJ1939_3BIT
-	4	SAEJ1939_4BIT
-	5	SAEJ1939_5BIT
-	6	SAEJ1939_6BIT
-	7	SAEJ1939_7BIT

**Kontrollstruktur  
TJ1939Tx**

```
J1939Tx: Struct
// Status der gesendeten Nachricht
    byStatus      : Byte;
// Priorität der gesendeten Nachricht
    byPriority     : Byte;
End_Struct;
```

**Verwenden der Funktion**

```
Result := SAEJ1939AddTx (
    1,
    0xFEEE,
    0x00,
    2
    0
    SAEJ1939_BYTE,
```

```
sizeof(var_Fueltemp),
var_Fueltemp,
struct_TJ1939Tx_EngineTemperatureTbl,
1500,
120);
```

---

### JetSym-STX-Programm

Festlegung einer neuen Priorität:

Der Prioritätswert 0 hat die höchste Priorität, der Prioritätswert 7 die niedrigste Priorität. Die Nachricht mit der Priorität 6 kann von einer Nachricht der Priorität 4 verdrängt werden (wenn die Nachrichten gleichzeitig gesendet werden). Die Parameter **InhibitTime** und **EventTime** werden beim Funktionsaufruf nicht explizit aufgeführt. In diesem Fall werden die Default-Werte verwendet.

```
Var
    bySAEJ1939Channel   : Byte;
    own_Source_Address  : Byte;

// PGN 65262 Engine Temperature 1
    Fueltemp           : Byte;
    EngineTemperatureTbl : J1939Rx;
End_Var;

Task main autorun

// Initialisierung CAN 1
bySAEJ1939Channel := 1;
own_Source_Address := 20;
SAEJ1939Init (bySAEJ1939Channel, own_Source_Address);

// PGN 65262 Engine Temperature
// Eine neue Priorität festlegen
EngineTemperatureTbl.byPriority := 6;
SAEJ1939AddTx (bySAEJ1939Channel, 65262, 2, 1, SAEJ1939_BYTE,
sizeof(Fueltemp), Fueltemp, EngineTemperatureTbl);

End_Task;
```

---

### Nachricht TSC1

Eine besondere Behandlung ist bei der Nachricht TSC1 zu beachten (PGN = 0 BytePos = 8). Diese beinhaltet einen Zähler und eine Prüfsumme. Will man diese Nachricht auf den Bus senden, muss die Nachricht angemeldet werden. Der Inhalt des aktuellen Werts auf dem Bus wird nicht in die STX-Variable übertragen. Der Wert steht der Applikation nicht zur Verfügung.

#### Beispiel:

```
stateSAEJ1939Tx.byPriority = 3;
SAEJ1939AddTx(1, 0x0000, 8, 1, SAEJ1939_BYTE, 1, value,
stateSAEJ1939Tx, 10, 10);
```

---

### Anleitung des Motorherstellers

Der Kunde erfährt die Daten (Priorität, PGN, SA und den Aufbau der Datenbytes) aus der Anleitung des Motorherstellers.

---

## STX-Funktion SAEJ1939RequestPGN()

### Einleitung

Mit dem Aufruf der Funktion `SAEJ1939RequestPGN()` wird nach einer PGN eine Anfrage an die DA (Destination Address) gesendet.

Wenn ein gültiger Wert empfangen oder der Timeout von 1.250 ms abgelaufen ist, erst dann wird die Funktion beendet.

Um den Wert der angefragten Nachricht zu erhalten, muss diese mit der Funktion `SAEJ1939AddRx()` für den Empfang angemeldet sein.

Diese Funktion muss zyklisch immer wieder neu aufgerufen werden.

### Funktionsdeklaration

```
Function SAEJ1939RequestPGN(
    CANNo: Int,
    byDA: Byte,
    ulPGN: Long,
    byPriority: Byte,
) : Int;
```

### Funktionsparameter

Die Funktion `SAEJ1939RequestPGN()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNo	CAN-Kanalnummer	0/1 ... CANMAX
byDA	Destination Address Adresse, von der die Nachricht angefordert wird	0 ... 253 Die eigene SA kann nicht verwendet werden.
ulPGN	PGN Parameter Group Number	0 ... 0x3FFFF
byPriority	Priorität	0 ... 7 Default-Wert: 6

### Rückgabewert

Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

#### Rückgabewert

0	Nachricht wurde empfangen
-1	Timeout, weil keine Antwort erhalten

### CANNo-Parameter

Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit `CANNo = 0/1` ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen `CANMAX` ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

### Data Type-Parameter

Die folgenden Datentypen sind möglich.

Byte-Typen	Bit-Typen	SAEJ1939
1	-	SAEJ1939_UNSIGNED8 SAEJ1939_BYTE
2	-	SAEJ1939_UNSIGNED16 SAEJ1939_WORD
4	-	SAEJ1939_UNSIGNED32 SAEJ1939_DWORD
n	-	SAEJ1939_STRING
-	1	SAEJ1939_1BIT
-	2	SAEJ1939_2BIT
-	3	SAEJ1939_3BIT
-	4	SAEJ1939_4BIT
-	5	SAEJ1939_5BIT
-	6	SAEJ1939_6BIT
-	7	SAEJ1939_7BIT

### Verwenden der Funktion

```
Result := SAEJ1939RequestPGN (  
    1,  
    0x00,  
    0xFEE5,  
    5);
```

### JetSym-STX-Programm

Das Gerät mit der eigenen SA von 20 möchte die PGN 65253 *Engine Hours* von einer Motorsteuerung mit der SA 0 anfordern. Aus dieser PGN soll die SPN 247 *Engine Total Hours of Operation* ausgelesen werden. Deshalb ist es notwendig durch Aufruf der Funktion `SAEJ1939AddRx()` den Empfang der SPN 247 anzumelden.

Der Parameter `byPriority` wird beim Funktionsaufruf nicht explizit aufgeführt. In diesem Fall wird der Default-Wert verwendet.

```
Var  
    bySAEJ1939Channel    : Byte;  
    own_Source_Address  : Byte;  
  
    // PGN 65253 Engine Hours, Revolutions  
    EngineTotalHours     : Int;  
    EngineHoursTbl       : J1939Rx;  
End_Var;
```

```
Task main autorun
Var
    Return_value : Int;
End_Var;

// Initialisierung CAN 1
bySAEJ1939Channel := 1;
own_Source_Address := 20;
SAEJ1939Init (bySAEJ1939Channel, own_Source_Address);

// Engine Hours, Revolutions -- on Request
SAEJ1939AddRx (bySAEJ1939Channel, 65253, 0x00, 1, 0,
SAEJ1939_DWORD, sizeof(EngineTotalHours), EngineTotalHours, En-
gineHoursTbl, 5000, 150);

loop
    // Gesamte Betriebsstunden der Maschine anfordern
    Return_value := SAEJ1939RequestPGN (bySAEJ1939Channel, 0x00,
65253);

    If Return_value Then
        Trace ('PGN Request failed');
    End_If;
    Delay(T#5s);
end_loop;
End_Task;
```

---

## STX-Funktion SAEJ1939GetDM1()

### Einleitung

Mit dem Aufruf der Funktion `SAEJ1939GetDM1()` werden die aktuellen Diagnosefehlercodes angefordert (siehe SAE J1939-73 Nr. 5.7.1). Die entsprechende PGN-Nummer ist 65226. Diese Funktion muss zyklisch immer wieder neu aufgerufen werden.

### Funktionsdeklaration

```
Function SAEJ1939GetDM1 (
    CANNNo: Int,
    bySA: Byte,
    ref stJ1939DM1stat: TJ1939DM1STAT
    ref stJ1939DM1msg: TJ1939DM1MSG
) : Int;
```

### Funktionsparameter

Die Funktion `SAEJ1939GetDM1()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNNo	CAN-Kanalnummer	0/1 ... CANMAX
bySA	Source Address vom Sender der Nachricht	0 ... 253 Die eigene SA kann nicht verwendet werden
stJ1939DM1stat	IStatus IMsgCnt  IBuffer	Lamp Status Anzahl der empfangenen Nachrichten Größe von Variable stJ1939DM1msg
stJ1939DM1msg	ISPN byOC byFMI	Fehlercode Fehlerzähler Fehlertyp

### Rückgabewert

Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

#### Rückgabewert

0	ok
-1	Fehler bei der Parameterüberprüfung

### CANNNo-Parameter

Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit `CANNNo = 0/1` ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen `CANMAX` ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

**stJ1939DM1stat.IStatus**      **Default: 0xFF00**

Typ	Byte	Bitgruppe	Beschreibung
Status	1	8 - 7	Malfunction Indicator Lamp Status
		6 - 5	Red Stop Lamp Status
		4 - 3	Amber Warning Lamp Status
		2 - 1	Protect Lamp Status
Flash	2	8 - 7	Flash Malfunction Indicator Lamp
		6 - 5	Flash Red Stop Lamp
		4 - 3	Flash Amber Warning Lamp
		2 - 1	Flash Protect Lamp

Typ	Byte	Bitgruppe Wert	Beschreibung
Status	1	00	Lamps off
		01	Lamps on
Flash	2	00	Slow Flash (1 Hz, 50 % duty cycle)
		01	Fast Flash (2 Hz or faster, 50 % duty cycle)
		10	Reserved
		11	Unavailable / Do not Flash

**stJ1939DM1msg**

**Default-Wert:**

ISPN = 0

byOC = 0

byFMI = 0

Bei älteren Controllern (Grandfathered Setting):

ISPN = 524287 (0x7FFFF)

byOC = 31 (0x1F)

byFMI = 127 (0x7F)

**Verwenden der Funktion**

```
Result := SAEJ1939GetDM1(
    1,
    0x00,
    stdmlstat_pow,
    stdmlmsg_pow,);
```

### JetSym-STX-Programm

Das Gerät fordert mit dem Aufruf der Funktion SAEJ1939GetDM1 () die aktuellen Diagnosefehlercodes an (PGN 65226).

```
Var
    bySAEJ1939Channel    : Byte;
    own_Source_Address  : Byte;
    stdmlstat_pow       : J1939DM1Stat;
    stdmlmsg_pow        : Array[10] of J1939DM1Msg;
    MyTimer              : Timer;
End_Var;

Task main autorun

// Initialisierung CAN 1
bySAEJ1939Channel := 1;
own_Source_Address := 20;
SAEJ1939Init (bySAEJ1939Channel, own_Source_Address);

TimerStart (MyTimer, T#2s);

Loop

When (TimerEnd (MyTimer)) Continue;

// Die Diagnose-Fehlercodes DM1 POW anfordern
stdmlstat_pow.lBuffer := sizeof (stdmlmsg_pow);
SAEJ1939GetDM1 (bySAEJ1939Channel, 0x00, stdmlstat_pow,
stdmlmsg_pow);

TimerStart (MyTimer, T#2s);

End_Loop;

End_Task;
```

---

## STX-Funktion SAEJ1939GetDM2()

### Einleitung

Mit dem Aufruf der Funktion `SAEJ1939GetDM2()` werden die Diagnosefehlercodes, die den Aktuellen vorausgingen, angefordert (siehe SAE J1939-73 Nr. 5.7.2). Die entsprechende PGN-Nummer ist 65227.

### Funktionsdeklaration

```
Function SAEJ1939GetDM2 (
    CANNNo: Int,
    bySA: Byte,
    ref stJ1939DM2stat: TJ1939DM2STAT
    ref stJ1939DM2msg: TJ1939DM2MSG
) : Int;
```

### Funktionsparameter

Die Funktion `SAEJ1939GetDM2()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNNo	CAN-Kanalnummer	0/1 ... CANMAX
bySA	Source Address vom Sender der Nachricht	0 ... 253 Die eigene SA kann nicht verwendet werden
stJ1939DM2stat	IStatus IMsgCnt  IBuffer	Lamp-Status Anzahl der empfangenen Nachrichten Größe von Variable stJ1939DM2msg
stJ1939DM2msg	ISPN byOC byFMI	Fehlercode Fehlerzähler Fehlertyp

### Rückgabewert

Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

#### Rückgabewert

0	ok
-1	Fehler bei der Parameterüberprüfung

### CANNNo-Parameter

Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit `CANNNo = 0/1` ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen `CANMAX` ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

**stJ1939DM2stat.IStatus**      **Default:** 0xFF00

Typ	Byte	Bitgruppe	Beschreibung
Status	1	8 - 7	Malfunction Indicator Lamp Status
		6 - 5	Red Stop Lamp Status
		4 - 3	Amber Warning Lamp Status
		2 - 1	Protect Lamp Status
Flash	2	8 - 7	Flash Malfunction Indicator Lamp
		6 - 5	Flash Red Stop Lamp
		4 - 3	Flash Amber Warning Lamp
		2 - 1	Flash Protect Lamp

Typ	Byte	Bitgruppe Wert	Beschreibung
Status	1	00	Lamps off
		01	Lamps on
Flash	2	00	Slow Flash (1 Hz, 50 % duty cycle)
		01	Fast Flash (2 Hz or faster, 50 % duty cycle)
		10	Reserved
		11	Unavailable / Do not Flash

**stJ1939DM2msg**

**Default-Wert:**

ISPN = 0

byOC = 0

byFMI = 0

Bei älteren Controllern (Grandfathered Setting):

ISPN = 524287 (0x7FFFF)

byOC = 31 (0x1F)

byFMI = 127 (0x7F)

**Verwenden der Funktion**

```
Result := SAEJ1939GetDM2 (
    1,
    0x00,
    stdm2stat_pow,
    stdm2msg_pow,);
```

**JetSym-STX-Programm**

Das Gerät fordert mit dem Aufruf der Funktion SAEJ1939GetDM2 () die aktuellen Diagnosefehlercodes an (PGN 65227).

```
Var
    bySAEJ1939Channel    : Byte;
    own_Source_Address  : Byte;
    stdm2stat_pow       : J1939DM2Stat;
    stdm2msg_pow        : Array[10] of J1939DM2Msg;
End_Var;

Task main autorun

// Initialisierung CAN 1
bySAEJ1939Channel := 1;
own_Source_Address := 20;
SAEJ1939Init (bySAEJ1939Channel, own_Source_Address);

// Wird benötigt für einen zyklischen Task
TaskAllEnableCycle ();
EnableEvents;
End_Task;

Task t_RequestPGN_5000 cycle 5000

Var
    Return_value : Int;
End_Var;

// Die Diagnose-Fehlercodes DM2 POW anfordern
stdm2stat_pow.lBuffer := sizeof (stdm2msg_pow);
Return_value := SAEJ1939GetDM2 (bySAEJ1939Channel, 0x00,
stdm2stat_pow, stdm2msg_pow);

If Return_value Then
    Trace ('DM2 Request failed');
End_If;

End_Task;
```

## STX-Funktion SAEJ1939SetSPNConversion()

### Einleitung

Mit dem Aufruf der Funktion `SAEJ1939SetSPNConversion()` wird die Anordnung der Bytes der Nachricht, die mit der Funktion `SAEJ1939GetDM1()` oder `SAEJ1939GetDM2()` angefordert wird, festgelegt. Anders ausgedrückt, es wird die Konvertierungsmethode festgelegt.

### Funktionsdeklaration

```
Function SAEJ1939SetSPNConversion(
    CANNo: Int,
    bySA: Byte,
    iConversionMethod: Int,
) : Int;
```

### Funktionsparameter

Die Funktion `SAEJ1939SetSPNConversion()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNo	CAN-Kanalnummer	0/1 ... CANMAX
bySA	Source Address vom Sender der Nachricht	0 ... 253
iConversionMethod	Konvertierungsmethode	1 ... 4 4: Automatisch erkannt 2: Default

### Rückgabewert

Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

#### Rückgabewert

0	ok
-1	Fehler bei der Parameterüberprüfung

### CANNo-Parameter

Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit `CANNo = 0/1` ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen `CANMAX` ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

### Verwenden der Funktion

```
Result := SAEJ1939SetSPNConversion(
    1,
    0xAE,
    4);
```

## STX-Funktion SAEJ1939GetSPNConversion()

**Einleitung** Mit dem Aufruf der Funktion `SAEJ1939GetSPNConversion()` wird die aktuell eingestellte Konvertierungsmethode festgestellt.

**Funktionsdeklaration**

```
Function SAEJ1939SetSPNConversion (
    CANNo: Int,
    bySA: Byte,
    iConversionMethod: Int,
) : Int;
```

**Funktionsparameter** Die Funktion `SAEJ1939GetSPNConversion()` hat die folgenden Parameter.

Parameter	Beschreibung	Wert
CANNo	CAN-Kanalnummer	0/1 ... CANMAX
bySA	Source Address vom Sender der Nachricht	0 ... 253
iConversionMethod	Konvertierungsmethode	1 ... 4 4: Automatisch erkannt 2: Default

**Rückgabewert** Die Funktion übergibt die folgenden Rückgabewerte an das übergeordnete Programm.

### Rückgabewert

0	ok
-1	Fehler bei der Parameterüberprüfung

**CANNo-Parameter** Dieser Parameter gibt die Nummer der SAEJ1939-Schnittstelle an. Die erste Schnittstelle wird mit `CANNo = 0/1` ausgewählt. Die Anzahl der SAEJ1939-Schnittstellen ist geräteabhängig. Die mögliche Anzahl der SAEJ1939-Schnittstellen `CANMAX` ist in den *technischen Daten* und der *Kurzreferenz* der jeweiligen Betriebsanleitung erwähnt.

**Verwenden der Funktion**

```
Result := SAEJ1939GetSPNConversion (
    1,
    0xAE,
    actual_conversion_method);
```

Jetter AG  
Gräterstraße 2  
71642 Ludwigsburg | Germany

Tel +49 7141 2550-0  
Fax +49 7141 2550-425  
[info@jetter.de](mailto:info@jetter.de)  
[www.jetter.de](http://www.jetter.de)

We automate your success.