SAE INTERNATIONAL

SURFACE VEHICLE STANDARD

J2799™ JUN2024

Issued 2010-03
Revised 2024-06

Superseding J2799 DEC2019

Hydrogen Surface Vehicle to Station Communications Hardware and Software

RATIONALE

This document has a new communication protocol version number 2.00 for high flow fueling protocols, such as SAE J2601/5. This includes a larger TV range and structured formatting rules in order to send additional information in the OD field. The geometry requirements for the IrDA hardware are adjusted for larger receptacle/nozzles.

TABLE OF CONTENTS

1. SCOPE

This standard specifies the communications hardware and software requirements for fueling hydrogen surface vehicles (HSV), such as fuel cell vehicles, but may also be used where appropriate with heavy-duty vehicles (e.g., buses) and industrial trucks (e.g., forklifts) with compressed hydrogen storage. It contains a description of the communications hardware and communications protocol that may be used to refuel the HSV. The intent of this standard is to enable harmonized development and implementation of the hydrogen fueling interfaces.

This standard is intended to be used in conjunction with the hydrogen fueling protocols in SAE J2601 and nozzles and receptacles conforming with SAE J2600.

1.1 Gaseous Hydrogen Fueling System Background

The overall hydrogen fueling system consists of a dispenser, HSV, and hydrogen fueling coupling. The nozzle and receptacle mechanically couples the hydrogen fueling system (dispenser) and the HSV and allows hydrogen gas to flow between them. The interface between the two is defined as the fueling interface. The data is transmitted from the IR transmitter on the HSV to the IR (infrared) receiver on the dispenser. See Figure 1.

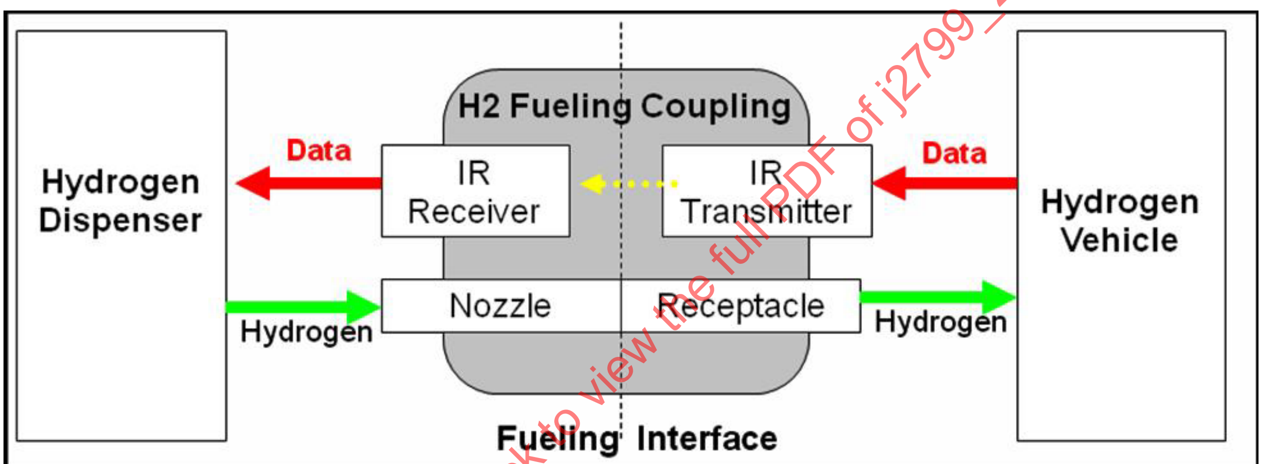

Figure 1 - Overall gaseous hydrogen fueling system

2. REFERENCES

2.1 Applicable Documents

The following publications form a part of this specification to the extent specified herein. Unless otherwise indicated, the latest issue of SAE publications shall apply.

2.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or +1 724-776-4970 (outside USA), www.sae.org.

SAE J2579	Standard for Fuel Systems in Fuel Cell and Other Hydrogen Vehicles
SAE J2600	Compressed Hydrogen Surface Vehicle Fueling Connection Devices
SAE J2601	Fueling Protocols for Light Duty Gaseous Hydrogen Surface Vehicles
SAE J2601-5	High-Flow Prescriptive Fueling Protocols for Gaseous Hydrogen Powered Medium and Heavy-Duty Vehicles

2.1.2 IrDA Publications

Copies of these documents are available online at www.IrDA.org.

IrDA IrPHY 1.4 IrDA Serial Infrared Physical Layer Specification

IrDA IrLAP 1.1 Serial Infrared Link Access Protocol

IrDA Physical Layer Measurement Guidelines V1.1, 8 Sept 2000

2.1.3 ANSI Publications

Copies of these documents are available online at https://webstore.ansi.org/.

CSA HGV 4.3 Test Methods for Hydrogen Fueling Evaluation

2.1.4 IEC Publications

Available from IEC Central Office, 3, rue de Varembe, P.O. Box 131, CH-1211 Geneva 20, Switzerland, Tel: +41 22 919 02 11, www.iec.ch.

IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems

IEC 61511 Functional Safety - Safety Instrumented Systems for the Process Industry Sector

2.1.5 ISO Publications

Copies of these documents are available online at https://webstore.ansi.org/.

ISO 17268 Gaseous hydrogen land vehicle refuelling connection devices

ISO 26262 Road Vehicles - Functional Safety Related Publications

2.2 Related Publications

The following publications are provided for information purposes only and are not a required part of this SAE Technical Report.

2.2.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or +1 724-776-4970 (outside USA), www.sae.org.

AS568 Aerospace Size Standard for O-Rings

SAE J2574 Fuel Cell Vehicle Terminology

SAE J2578 Recommended Practice for General Fuel Cell Vehicle Safety

SAE J2601-2 Fueling Protocol for Gaseous Hydrogen Powered Heavy Duty Vehicles

SAE J2601-3 Fueling Protocol for Gaseous Hydrogen Powered Industrial Trucks

SAE J2601-4 Ambient Temperature Fixed Orifice Fueling

SAE J2760 Pressure Terminology Used in Fuel Cells and Other Hydrogen Vehicle Applications

2.2.2 WE-NET Publications

Lynch, F. (2002). *Communication Between Vehicle Onboard Tanks and a Hydrogen Fueling System*. WE-NET Task 7A-13, HCI.

3. DEFINITIONS

3.1 AMBIENT TEMPERATURE

The temperature of the air measured at the fueling station where the measurement is protected from direct sunlight and other radiative and environmental effects.

3.2 CHSS MEASURED TEMPERATURE

The measured temperature of the gas in the vehicle CHSS.

NOTE 1: If the vehicle contains a temperature measurement device for the purpose of sending a temperature signal to the dispenser during fueling, this temperature is also assumed to be the average temperature of the gas in the vehicle. Due to the accuracy of the sensor, the vehicle manufacturer should consider the tolerances of the temperature measurement and include them as criteria for the Abort and Measured Temperature signals.

NOTE 2: For vehicles with multiple tanks in the CHSS, the measured temperature in each tank can vary. The vehicle manufacturer should consider the best approach for transmitting a single representative temperature as MT. One approach is to transmit the lowest measured temperature from the tanks in the CHSS, since the primary purpose of MT is to determine when to end the fill based on a calculated pressure or density where MT is an input. The highest measured temperature can be utilized in the criteria for sending an Abort signal.

3.3 COMPRESSED HYDROGEN STORAGE SYSTEM (CHSS)

As defined in SAE J2579, the compressed hydrogen storage system consists of the pressurized containment vessel(s), pressure relief devices (PRDs), shut-off devices(s), and all components, fittings, and fuel lines between the containment vessel(s) and these shut-off device(s) that isolate the stored hydrogen from the remainder of the fuel system and the environment.

3.4 COMPRESSED HYDROGEN STORAGE SYSTEM (CHSS) CAPACITY

The total water volume of all the storage vessels in the CHSS, or the total mass of hydrogen stored in all the storage vessels in the CHSS at the nominal working pressure at 15 °C.

NOTE: The total mass of hydrogen stored in the CHSS at the nominal working pressure at 15 °C is equivalent to a 100% state of charge.

3.5 COUPLING OR CONNECTION

A joined assembly of a nozzle and receptacle that permits rapid coupling (or connecting) and decoupling (or disconnecting) of fuel supply hose to the vehicle fuel system, as per SAE J2600.

3.6 DATA COMMUNICATIONS LINK

The data communications portion of the connector consists of an infrared emitter that is mounted in the HSV in close proximity to the receptacle and infrared receivers that are located in the nozzle. These devices allow the HSV to send data to the fuel dispensing system. Figure 2 illustrates the emitter and receiver placement.

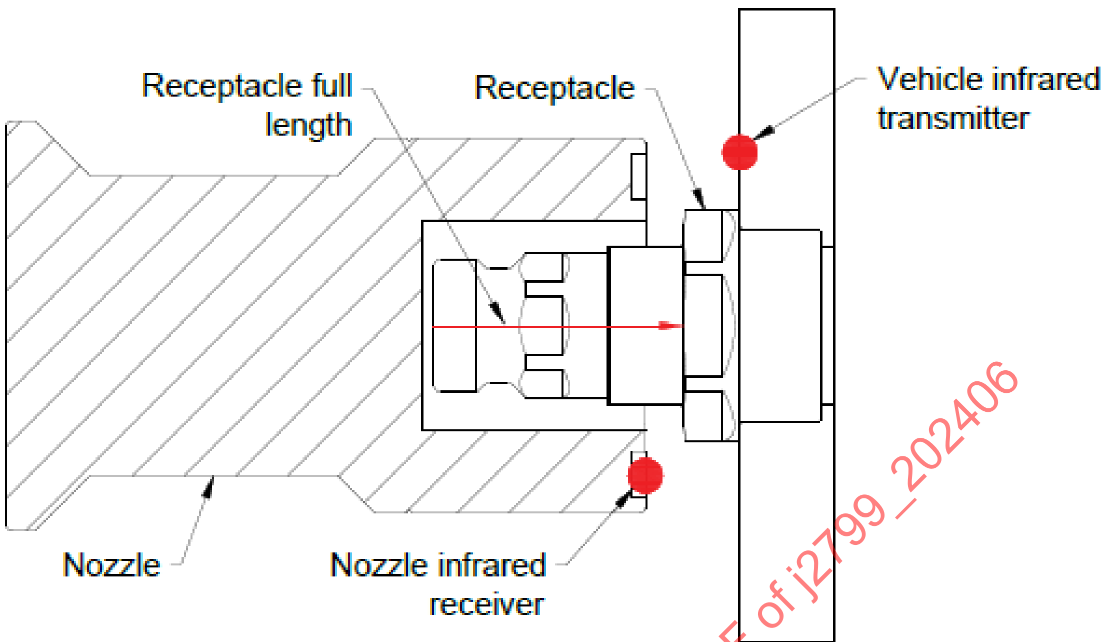

Figure 2 - Connector with infrared communications link

3.7 DESIGN VERFICATION TEST

A method of testing a product to assure that it meets all of its design specifications.

3.8 HYDROGEN DISPENSING EQUIPMENT (DISPENSER)

The equipment required to condition and transfer fuel from the station to vehicle storage for the purpose of fueling the vehicle.

3.9 HYDROGEN SURFACE VEHICLE (HSV)

Any surface vehicle that stores and uses hydrogen as a fuel. An example of an HSV is a Fuel Cell Vehicle.

3.10 NOZZLE

Device connected to a fuel dispensing system that engages the HSV receptacle and permits transfer of fuel. See coupling or connection (see 3.5).

3.11 PRESSURE CLASS

The pressure class will be defined by the protocol's nominal working pressure. The class is denoted by the letter H followed by the nominal working pressure in MPa. For example, H70 is the pressure class for a hydrogen fueling protocol with a NWP of 70 MPa.

3.12 RECEPTACLE

Device connected to a vehicle or storage system that receives the dispenser nozzle and permits transfer of fuel. This may also be referred to as a fueling inlet.

3.13 COMMUNICATIONS DEFINITIONS

3.13.1 COMMUNICATIONS FUELING

Communications fueling means that a valid data connection has been established from vehicle to fueling station dispenser as described in SAE J2799.

3.13.2 NON-COMMUNICATIONS FUELING

Non-communications fueling means that no valid data connection from vehicle to fueling station dispenser as per SAE J2799 exists or that the received data has not been recognized as valid (or wished to be used) by the dispenser.

3.13.3 OPTIONAL DATA (OD Data)

Data contained in the Optional Data Block.

3.13.4 OPTIONAL DATA BLOCK (OD Block)

A set of data in the OD data field that contains a header and data.

3.13.5 OPTIONAL DATA BLOCK HEADER (OD Header)

A signifier at the start of the Optional Data Block (see 3.13.5) used to designate the data contained in the Optional Data Block.

3.13.6 DATA TYPES

3.13.6.1 DYNAMIC DATA

A type of data transmitted from the vehicle to the station that can change during fuelings.

3.13.6.2 PERSISTENT DATA

A type of data transmitted from the vehicle to the station that is fixed during the duration of the fueling.

3.13.6.3 STATIC DATA

A type of data transmitted from the vehicle to the station that is permanently programmed into the vehicle and does not change over the life of the vehicle or component, except during authorized maintenance.

4. ABBREVIATIONS AND SYMBOLS

4.1 Symbols

d_{nmax} distance nozzle maximum

d_{nmin} distance nozzle minimum

d_{vmax} distance vehicle maximum

d_{vmin} distance vehicle minimum

r_n radius nozzle

r_{rec} radius receptacle

r_v radius vehicle

5. COMPRESSED HYDROGEN SURFACE VEHICLE TO STATION COMMUNICATIONS DEVICE

5.1 Non-ASIL/SIL Communications Assumptions

The IrDA communications defined in SAE J2799 have not been Automotive Safety Integrity Level (ASIL) or Safety Integrity Level (SIL) classified to any standard at the time of publishing this standard. This means until ASIL/SIL certification is achieved, the signals are not guaranteed to be accurate and the station is responsible to ensure that process requirements are followed if it uses the communications signals.

The following are some of the guidelines that may be used to qualify components and systems for the station side SIL: IEC 61508/61511 and for vehicle side ASIL: ISO 26262.

5.2 Physical and Functional Requirements

5.2.1 Physical Layer Protocol Specification

The infrared data link uses an optical interface based on the 38400-baud interface specified by IrDA physical layer specification, IrDA IrPHY 1.4. The physical layer implements only a portion of the IrPHY specification, and it is not meant to be compatible with standard IrDA devices.

5.2.1.1 Modulation Scheme

The modulation scheme defined below is described in Section 2.3 of IrPHY 1.4. A return-to-zero-inverted (RZI) modulation scheme is used, where a “0” is represented by a light pulse and a “1” is represented by no light pulse. The optical pulse duration is nominally 3/16 of the bit duration. The maximum pulse duration is 3/16 of the bit duration, plus a tolerance of 0.60 µs.

5.2.1.2 Byte Framing

The signal is organized into IR frames, as shown in Figure 3, that have a direct correlation to UART frames as described in Section B.3 of IrPHY 1.4. Each byte is transmitted asynchronously with a start bit, 8 data bits, and a stop bit. Data bits are transmitted in serial byte order, with the least significant bit (LSB) transmitted first and the most significant bit (MSB) transmitted last. Bits are transmitted at a rate of 38400 bits per second.

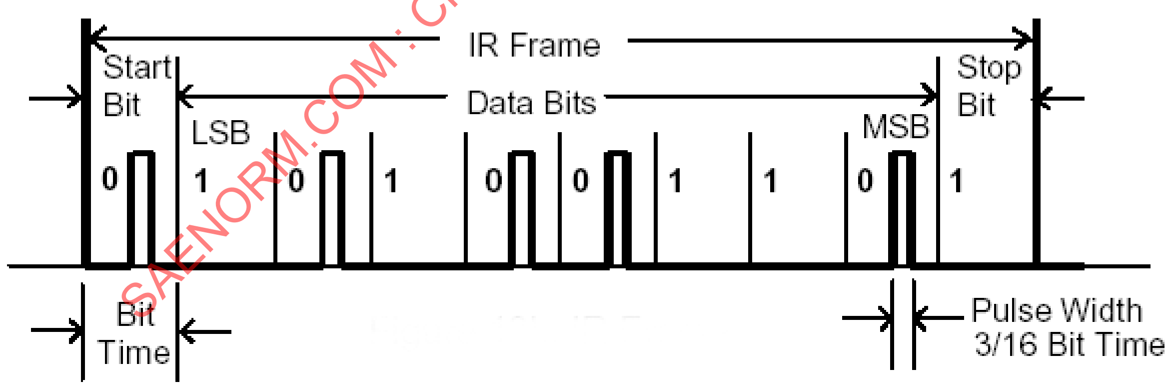

Figure 3 - IR frame

5.2.2 Data Link Layer Protocol Specification

The purpose of the data link layer is to ensure valid application data is transferred from the transmitting side to the receiving side of the data link. The data link defined below is based on the low-speed asynchronous data link specified by IrDA IrLAP 1.1. The data link implements only a portion of the IrDA specification, and it is not meant to be compatible with standard IrDA devices.

5.2.2.1 Data Link Control Characters

The following control characters are used for the data link framing as described in Section 10.1 of IrLAP 1.1.

- (X)BOF: Extra Beginning of Frame character shall be 0xFF hexadecimal.
- BOF: Beginning of Frame character shall be 0xC0 hexadecimal.
- EOF: End of Frame character shall be 0xC1 hexadecimal.
- CE: Control Escape character shall be 0x7D hexadecimal (see A.1.1).

5.2.2.2 Frame Check Sequence (FCS) Field

The frame check sequence field detects errors in the received frame as described in Section 10.1.2 of IrLAP 1.1. The FCS field shall be a 16-bit CRC-CCITT cyclic redundancy check computed on the bytes in the application data. The polynomial for the CRC shall be X16 + X12 + X5 + 1. The FCS is transmitted with the least significant byte first followed by the most significant byte.

An implementation of the CRC-CCITT FCS algorithm for calculating the CRC has been provided in the fast frame check sequence (FCS) implementation section of the appendix.

5.2.2.3 Data Link Frame

The application data packets are transmitted in data link frames as described in Section 10 of IrLAP 1.1. The application data packets shall be transmitted in data link frames. Five (X)BOF characters shall precede the data link frame transmission. A single BOF character shall be transmitted at the start of the data link frame. The application data packet shall be transmitted immediately following the BOF character. A frame check sequence field shall be transmitted immediately following the application data packet. The data link frame shall be terminated with an EOF character as shown in Table 1.

Table 1 - Data link framing

(X)BOFs	BOF	ApplicationData	FCS	EOF

5.2.2.4 Transparency

A transparency character is defined to transform application data bytes (and FCS field bytes), which correspond to data link control characters as described in Section 10.1.3 of IrLAP 1.1. The transmitting side of the data link shall transform any data bytes, or FCS bytes, corresponding to control characters (X)BOF, BOF, EOF, or CE into non-control characters prior to transmitting the data link frame.

The transformation method on the transmitting side shall be:

a. Insert a control escape (CE) byte preceding the data byte.

b. Exclusive OR the data byte with 0x20 hexadecimal.

The receiving side shall reverse the transformation whenever a CE byte is received in the data stream.

See Section B.2 for an IrDA frame example with the maximum transparency characters possible.

The receiving side shall reverse the transformation using the following method:

a. Discard the control escape (CE) byte.

b. Exclusive OR the data byte with 0x20 hexadecimal.

The state machine shown in Figure 4 is an example of what must be implemented by the receiver to properly decode received characters as described in Section 10.1.3 and 10.1.5 of IrLAP 1.1.

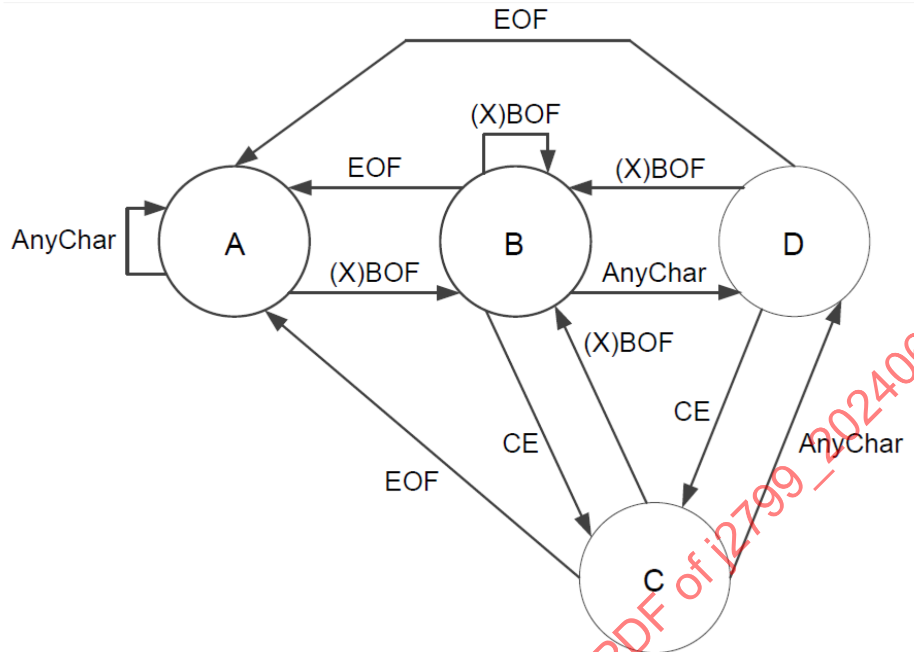

Figure 4 - Receiver transparency state machine

All bytes received are discarded in state A until a BOF (or (X)BOF) is received. All bytes received on the AnyChar events (which cause a transition from state B to state D, or from state C to state D) are inserted into the receive data packet. A complete data packet has been decoded on the EOF event (which causes a transition from state D to state A).

5.2.3 Presentation Layer Protocol Specification

The dispenser shall attempt to support the standard-communications fueling process, as defined in the associated fueling protocol, for which it has received all the required data that meet the required data integrity checks.

5.2.3.1 Data Integrity Checks

5.2.3.1.1 Data Type

All data shall be transmitted in ASCII format.

5.2.3.1.2 Proper Delimiter Check

A single pipe character ("|" ASCII, $7C hexadecimal) shall be used to delimit all pieces of data. All valid data shall be contained between two pipe characters, as shown in Table 2.

Table 2 - Delimiter permutations and interpretations

|MP=010.0|VN=02.99|	Proper format, both valid
MP=010.0|VN=02.99|	MP not valid due to no leading delimiter
|MP=010.0|VN=02.99	VN not valid due to no trailing delimiter
|MP=010.0||VN=02.99|	All data valid, redundant delimiter is not required

5.2.3.1.3 Proper Tag Check

The first two characters after the data delimiter shall contain a defined data tag. All tag comparisons shall be case-sensitive. The third character after the delimiter shall be the "=" character (hexadecimal $3D). Any deviations from this format shall result in all data between the delimiters being discarded.

5.2.3.1.4 Properly Defined Value Check for Required Character Data

All character data for a required data field shall be an exact match for a defined value. All defined data comparisons shall be case-sensitive. Extra spaces or characters shall invalidate the data.

5.2.3.1.5 Properly Defined Value Check for Required Numerical Data

All defined numerical data positions shall be fueled using leading and following zeros as required to fuel the defined format. Extra spaces or characters shall invalidate the data. All defined numerical data shall be greater than or equal to the minimum defined value and less than or equal to the maximum defined value, as shown in Table 3.

Table 3 - Value check permutations and interpretations

|MP=035.0|	Proper format
|MP=35.0|	Not valid due to lack of leading zero
|MP=035|	Not valid due to lack of decimal point and trailing zero
|MP= 035.0|	Not valid due to extra space before numerical data
|MP=3.5E1|	Not valid, scientific notation not supported
|MP=035.0|	Proper range
|MP=100.1|	Not valid, data outside of defined data range (000.0 - 100.0)

5.2.3.2 Data Interval Check

All required data fields for a given fueling process shall be updated with properly formatted message data within five times the nominal transmission interval (5 x 100 ms) to be considered valid, except for the OD data field, where the data interval requirement is to be 10 x 100 ms. If none of the message meets this requirement within 5 x 100 ms, the communications shall be considered lost.

The only other exception to this requirement is that the dispenser shall respond to any properly formatted abort command (FC=Abort) from the vehicle regardless of whether the data has been received within its nominal transmission interval.

5.2.3.2.1 Communications Interval Tolerance

The signal interval tolerance shall be within ±20% of interval time.

5.3 Communications Hardware for Gaseous Fueling

5.3.1 Vehicle Transmitter Requirements

5.3.1.1 The vehicle shall have at least one infrared transmitter as shown in Figure 5.

5.3.1.2 The sum of the infrared transmitter(s) on the vehicle shall have an effective half angle of a=55 degrees. See Figure 7.

5.3.1.3 The center of the transmitter element(s) shall be located on the mounting surface at a radius of $r_v = r_{rec} \times 1.7 \pm 4$ mm from the axis of the receptacle, where r_{rec} is the largest radius of the standardized receptacle defined in connector standards, such as SAE J2600 and ISO 17268. See Figure 5.

5.3.1.4 The external chassis of the vehicle transmitter(s) facing the receiver shall be at least d_{vmin} from reference plane z=0, which is the plane defined by the leading edge of the receptacle. d_{vmin} is defined as the full length of the standardized receptacle (the edge of the receptacle to the stop ring) plus 10 mm. See Figure 7 and Table 4.

5.3.1.5 The external chassis of the vehicle transmitter(s) facing the receiver shall be at most d_{vmax} from reference plane z=0, which is the plane defined by the leading edge of the receptacle. d_{vmax} is defined as the full length of the standardized receptacle (the edge of the receptacle to the stop ring) plus 30 mm. See Figure 7 and Table 4.

5.3.1.6 The minimum intensity of the vehicle infrared emitter shall be 40 mW/sr within the range of the half angle. The maximum intensity of the vehicle infrared emitter shall be 100 mW/sr within the range of the half angle.

5.3.2 Nozzle Receiver Requirements

5.3.2.1 The nozzle shall have at least three infrared receivers. The infrared receivers on the nozzle shall have a minimum half angle of b=55 degrees. See Figure 7.

5.3.2.2 The angle between adjacent nozzle infrared receivers shall be no more than c=120 degrees. See Figure 6.

5.3.2.3 The nozzle receiver elements shall be located $r_n=r_{rec}$ x 1.7 ± 4 mm from the axis of the nozzle, where r_{rec} is the largest radius defined in the standard of the standardized receptacle defined in connector standards, such as SAE J2600 and ISO 17268. See Figure 6.

5.3.2.4 When the nozzle is fully engaged with the receptacle, the external chassis of the receivers on the nozzle facing the transmitter shall be at least d_{nmin} from the reference plane z=0. d_{nmin} is defined as the largest value between 15 mm and the full length of the standardized receptacle (the edge of the receptacle to the stop ring) minus 25 mm. See Figure 7 and Table 4.

5.3.2.5 When the nozzle is fully engaged with the receptacle, the external chassis of the receivers on the nozzle facing the transmitter shall be no more than d_{nmax}=from the reference plane z=0. d_{nmax} is defined as the full length of the standardized receptacle (the edge of the receptacle to the stop ring) minus 5 mm. See Figure 7 and Table 4.

5.3.2.6 For the above specified geometry, the minimum irradiance that the nozzle receiver will be required to detect will be 100 µW/cm^2, and the maximum irradiance that the nozzle receiver will be required to detect will be 50 mW/cm^2.

Table 4 - Definition of minimum and maximum infrared emitter and receiver distances for different connector sizes

	d_{vmin} (mm)	d_{vmax} (mm)	d_{nmin} (mm)	d_{nmax} (mm)
H35	40	60	15	25
H35HF	55	75	20	40
H70	50	70	15	35
H70HF	TBD	TBD	TBD	TBD

NOTE: The H70HF dimensions have not been defined in SAE J2600 or ISO 17268.

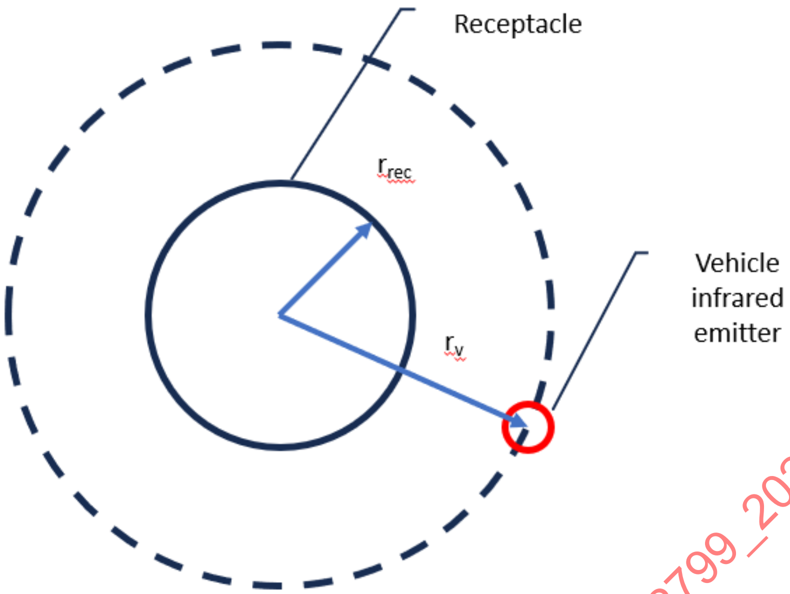

Figure 5 - Vehicle transmitter (front view)

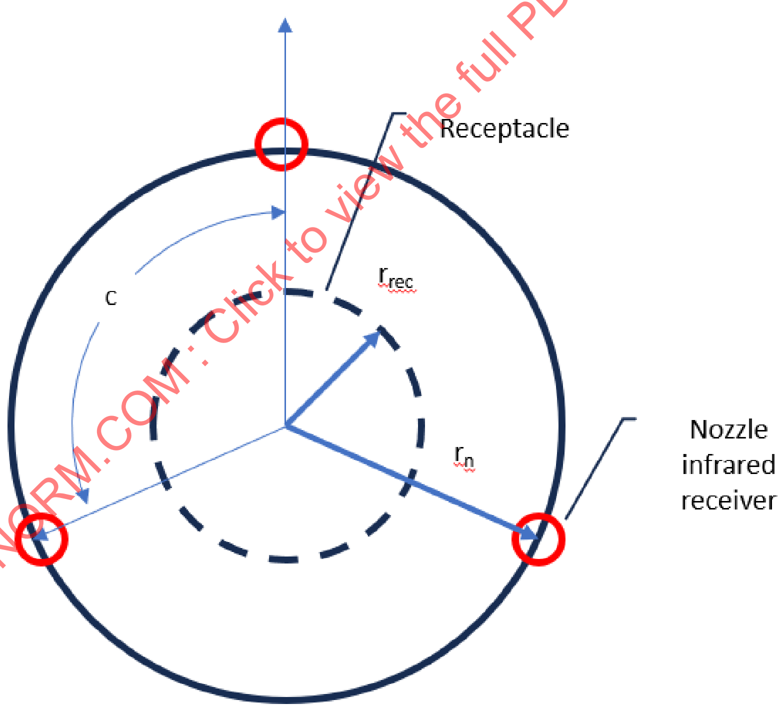

Figure 6 - Nozzle receiver (front view)

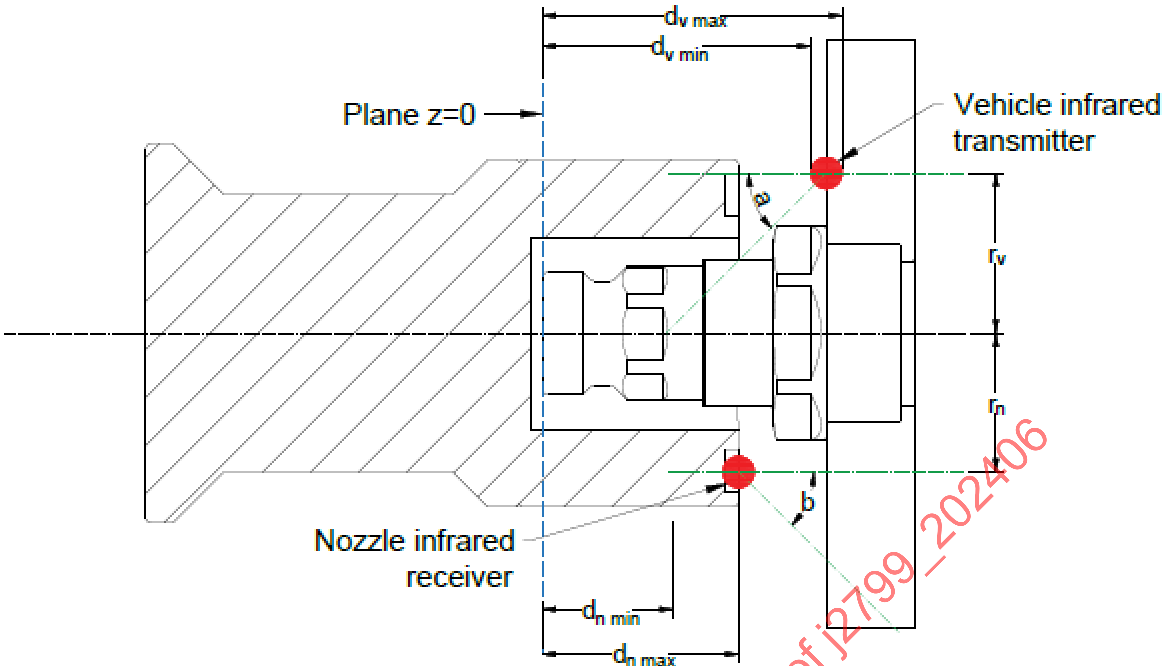

Figure 7 - Vehicle receptacle (side view)

5.4 Communications Definition (Version 01.10)

5.4.1 Gaseous Fueling Communications

5.4.1.1 Data Definitions

5.4.1.1.1 Communication Protocol Identifier

Tag:	ID=
Type:	Static
Units:	Not Applicable
Range:	SAE J2799
Example:	|ID=SAE␣J2799|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the communication protocol identifier to the dispenser to aid in the decoding of the transmitted data. For this revision of this standard, the ID shall be “SAE J2799”. The “␣”symbol is to denote a space and used after SAE in the identifier.

5.4.1.1.2 Data Communications Software Version Number

Tag:	VN=
Type:	Static
Units:	Not Applicable
Format:	##.##
Range:	00.00 - 99.99
Example:	|VN=01.10|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit its version number to the dispenser to aid in the decoding of the transmitted data. The major revision shall be defined as the whole number portion of the version number. The minor revision shall be defined as the decimal portion of the version number. Example: For version number 10.02, the major revision is 10 and the minor revision is 02.

For this version of the communications definition, VN=01.10

5.4.1.1.3 Total Volume

Tag:	TV=
Type:	Static
Units:	Liters
Format:	####.#
Range:	0000.0 - 5000.0
Example:	|TV=0200.0|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the total water volume of the CHSS in liters at the nominal working pressure to the dispenser.

NOTE: This is also known as the CHSS volume.

5.4.1.1.4 Receptacle Type

Tag:	RT=
Type:	Static
Units:	Not Applicable
Range:	H25, H35, H50, H70 (as defined by the SAE J2600 standard)
Example:	|RT=H70|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the pressure class of its CHSS, as defined in 3.11. The dispenser shall not dispense fuel if the RT is less than the pressure class of the nozzle.

5.4.1.1.5 Fueling Command

Tag:	FC=
Type:	Dynamic
Units:	Not Applicable
Range (VN=01.10):	Dyna, Stat, Halt, Abort
Example:	|FC=Dyna|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the fueling command signal to inform the dispenser on how to handle the fuel flow. All stations that are compliant with SAE J2799 shall use the following commands:

FC=Dyna Dispenser shall dispense fuel based upon the procedures defined in the appropriate fueling protocol in SAE J2601.

FC=Stat The dispenser shall dispense fuel based upon the procedures defined in the appropriate fueling protocol in SAE J2601.

FC=Halt The dispenser shall pause the fueling process, using the procedures defined in the appropriate fueling protocol in SAE J2601.

FC=Abort The dispenser shall terminate the fueling process and shall not dispense fuel until the dispenser fueling process is restarted by the operator.

5.4.1.1.6 Measured Pressure

Tag:	MP=
Type:	Dynamic
Units:	MPa
Format:	###.#
Range:	000.0 - 100.0
Example:	|MP=043.7|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the measured CHSS gas pressure in MPa.

5.4.1.1.7 Measured Temperature

Tag:	MT=
Type:	Dynamic
Units:	Kelvin
Format:	###.#
Range:	16.0 - 425.0
Example:	|MT=353.0|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the measured gas temperature of the hydrogen gas in the vehicle's storage tank in Kelvin.

5.4.1.1.8 Optional Data

Tag:	OD=
Type:	Static Persistent, or Dynamic
Units:	None
Range:	Any 0 to 74 Characters not including "|"
Example:	|OD=Parameter(s)|
Interval:	100 ms minimum
Direction:	Vehicle to Dispenser

The vehicle may transmit this optional data set to the dispenser. The data shall contain any character except the delimiter character ("|" ASCII, $7C hexadecimal). The optional data should not be viewed as an open configurable signal without a secondary identification of the vehicle (to avoid misinterpetation of multiple signal definitions).

5.5 Communications Definition (Version 02.00)

The following communication definitions in this section expand the range of total volumes for the TV parameter and provide structured formatting rules for data communicated in the OD field. These changes from version 1.10 to 2.00 are needed to facilitate new high flow fueling protocols such as SAE J2601/5. Communications definitions Version 02.00 may be used for other fueling protocols that need an expanded OD field.

5.5.1 Gaseous Fueling Communications

5.5.1.1 Data Definitions

5.5.1.1.1 Data Communication Protocol Identifier

Tag:	ID=
Type:	Static
Units:	Not Applicable
Range:	SAE J2799
Example:	|ID=SAE␣J2799|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the protocol identifier to the dispenser to aid in the decoding of the transmitted data. For this revision of this standard, the ID shall be “SAE J2799”. The “␣” symbol is to denote a space and used after SAE in the identifier.

5.5.1.1.2 Data Communications Software Version Number

Tag:	VN=
Type:	Static
Units:	Not Applicable
Format:	##.##
Range:	00.00 - 99.99
Example:	|VN=02.00|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit its version number to the dispenser to aid in the decoding of the transmitted data. The major revision shall be defined as the whole number portion of the version number. The minor revision shall be defined as the decimal portion of the version number. Example: For version number 10.02, the major revision is 10 and the minor revision is 02.

For this version of the communications definition, VN=02.00

5.5.1.1.3 Total Volume

Tag:	TV=
Type:	Static
Units:	Liters
Format:	####.#
Range:	0000.0 - 9999.9
Example:	|TV=0200.0|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the total water volume of the CHSS in liters at the nominal working pressure to the dispenser.

NOTE: This is also known as the CHSS volume.

5.5.1.1.4 Receptacle Type

Tag:	RT=
Type:	Static
Units:	Not Applicable
Range:	H25, H35, H50, H70 (as defined by the SAE J2600 standard)
Example:	|RT=H70|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the pressure class of its CHSS, as defined in 3.11. The dispenser shall not dispense fuel if the RT is less than the pressure class of the nozzle.

5.5.1.1.5 Fueling Command

Tag:	FC=
Type:	Dynamic
Units:	Not Applicable
Range (VN=02.00):	Dyna, Stat, Halt, Abort
Example:	|FC=Dyna|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the fueling command signal to inform the dispenser on how to handle the fuel flow. The fueling protocol shall define the actions taken based upon the fueling command.

The fueling commands defined in SAE J2799 are:

- FC=Dyna -
- FC=Stat -.
- FC=Halt -
- FC=Abort -

Fueling protocols may define other fueling commands.

5.5.1.1.6 Measured Pressure

Tag:	MP=
Type:	Dynamic
Units:	MPa
Format:	###.#
Range:	000.0 - 100.0
Example:	|MP=043.7|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the measured CHSS gas pressure in MPa.

5.5.1.1.7 Measured Temperature

Tag:	MT=
Type:	Dynamic
Units:	Kelvin
Format:	###.#
Range:	16.0 - 425.0
Example:	|MT=353.0|
Interval:	100 ms
Direction:	Vehicle to Dispenser

The vehicle shall transmit the measured gas temperature of the hydrogen gas in the vehicle's storage tank in Kelvin.

5.5.1.1.8 Optional Data

Tag:	OD=
Type:	Static, Persistent, or Dynamic
Units:	None
Range:	Any 0 to 240 Characters not including "|"
Example:	|OD=Parameter(s)|
Interval:	100 ms minimum
Direction:	Vehicle to Dispenser

5.5.1.1.8.1 Optional Data Block

The Optional Data (OD=) field, as defined in 5.5.1.1.8, may contain zero or more OD data blocks. The data block(s) shall contain any character except the delimiter character ("|" ASCII, $7C hexadecimal), end of OD data block character ("\" ASCII, 5C hexadecimal), and comma ("," ASCII, $2C hexadecimal). At the end of each OD data block a single backslash character ("\" ASCII, 5C hexadecimal) shall indicate the end of the block. Multiple data blocks are allowed to immediately follow the previous one.

The OD Header shall immediately follow the data delimiter and OD= defined data tag (i.e., |OD=). The OD Header shall be composed of a string of six to 15 characters. A single comma character ("," ASCII, $2C hexadecimal) shall be used to indicate the end of the OD Header. The OD Header should define the type of data that follows. OD Headers that are known to be used in the hydrogen industry are documented in Appendix D.

Zero or more OD Data shall follow the OD Header. Multiple OD Data are allowed to immediately follow the previous one. A single comma character ("," ASCII,$ 2C hexadecimal) should be used to delimit multiple OD Data.

As specified in 5.2.3.1.2, a single pipe character ("|" ASCII, $7C hexadecimal) shall designate the end of the Optional Data field.

Table 5 - Examples of OD Data Blocks, OD Header, and OD Data

Single OD data block Single OD data	|OD=MCFHFG24,TVL=0250\\|
Single OD data block Multiple OD data	|OD=CATDHF24,TVL=0250,FM=090\\|
Multiple OD data blocks Multiple OD data	|OD= MCFHFG24,TVL=0250\CATDHF24, TVL=0250, FM=090\\|

5.6 Verification Tests

5.6.1 Verification Test, Verification of a Gaseous Hydrogen Communications System

5.6.1.1 Verification Test Setup

To qualify an SAE J2799 communications device, whether vehicle or station side, it shall be independently tested using a verification test setup. The verification test setup shall consist of electronics capable of testing the full range of receiver/transmitter locations, powers, and sensitivities as specified in 5.3.1 and 5.3.2. The verification test setup shall have the transmitter and receiver positioned around a simulated receptacle, consisting of a shaft of stainless steel with a diameter of 30 mm. Simulated light sources' intensities and distance are specified by IrDA Physical Layer Measurement Guidelines (see 2.1.2 and 2.2.2). The simulated light sources shall be directed toward and be positioned within line of sight of the receivers. The apparatus shall be located in an environment with no ambient light. The verification test setup shall be capable of including a 38% transmittance filter in the 850 to 900 nm IR range placed 2 mm ± 0.5 mm in front of the transmitter. The setup shall not allow any transmitted or reflected IR light to pass around the filter.

The test setup shown in Figure 8 shall be used to perform the test scenarios listed in Table 6.

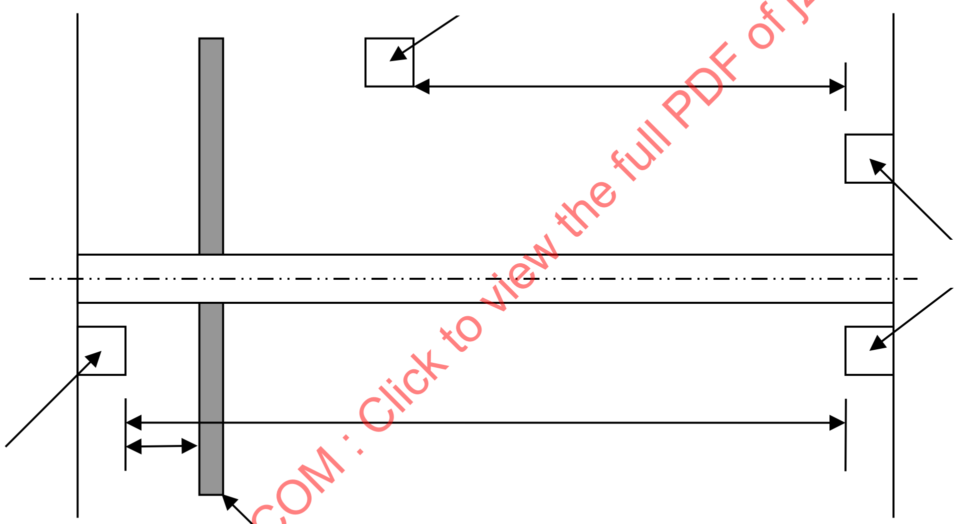

Figure 8 - Verification test setup

5.6.1.2 Test Scenario Definition

The scenarios listed in Table 6 shall be tested:

Table 6 - Test scenarios

Scenario	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
IR Filter	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	Y	Y	Y	Y
Simulated Sunlight	Y	N	N	Y	N	N	Y	N	N	Y	N	N	N	N	N	N
Simulated Fluorescent Light	N	Y	N	N	Y	N	N	Y	N	N	Y	N	N	N	N	N
Simulated Incandescent Light	N	N	Y	N	N	Y	N	N	Y	N	N	Y	N	N	N	N
Transmitter - Receiver Separation (mm)	55	55	55	55	55	55	15	15	15	15	15	15	55	55	15	15
Orientation (degrees)	30	30	30	30	30	30	0	0	0	0	0	0	30	30	30	30
Transmitter or Receiver Radial Dimension (mm)	26	26	26	18	18	18	26	26	26	18	18	18	18	26	18	26

The orientation is determined by setting the 0 degree reference to be where the transmitter directly aligns with any receiver. Scenarios 1 through 6 test the low signal to noise condition, scenarios 7 through 12 test the highest received power scenarios, and scenarios 13 through 16 test a dark ambient light.

5.6.1.3 Transient Lighting Test

In addition to the above scenarios, the transient effects of lighting shall be tested by repeating the poorest performing scenario in which a simulated light source is used. However, during transmission, the simulated light source is turned on for 10 seconds ± 5 seconds and then off for 10 seconds ± 5 seconds. This is repeated throughout the duration of the test.

5.6.1.4 Transmission Definition

For each scenario specified above, the transmitting side shall send 65536 messages, each containing nine unsigned word length numbers (1179648 bytes total). The algorithm to generate the data to be transmitted is included as follows.

```
For Message_Number=0 to 65535 step 1
For Word=0 to 8 step 1
Data(2*Word)=Low_Byte(Message_Number)
Data(2*Word+1)=High_Byte(Message_Number)
Next
Call Transmit_Data
Next
```

Each message shall contain the pre- and post-signal data as required for communications synchronization. The messages shall not contain a data length field, as the messages for this test are of fixed length. The messages shall not contain error-checking fields. The transmit controller shall maintain a message frequency of 10 Hz.

5.6.1.5 Test Criteria

When the transmission is complete, the data will be checked bit-for-bit. The number of incorrect bits will be recorded to determine the bit error ratio (BER) for the test as shown in Table 7.

Table 7 - Bit error ratio

Incorrect Bits	BER
0	<1.060E-07
1	1.060E-07
10	1.060E-06
100	1.060E-05
1000	1.060E-04
N	N/(1179648*8)

The SAE J2799 system tested shall pass each defined test with a BER of less than 10 e -4.

5.7 Verification of a Gaseous Hydrogen Communications System in a Dispenser

CSA HGV 4.3 shall define additional testing to confirm that the station dispenser hardware may receive the signals as per SAE J2799.

5.7.1 Examples of Verification Testing on Dispenser Side for CSA HGV 4.3

5.7.1.1 Demonstration of fueling with communications data from V.1.1 communications version.

5.7.1.2 Response due to an incorrect CRC or incorrect sequence of CRCs.

5.7.1.3 Response to data that is delayed.

5.7.1.4 Response to data that is out of order.

6. NOTES

6.1 Revision Indicator

A change bar (l) located in the left margin is for the convenience of the user in locating areas where technical revisions, not editorial changes, have been made to the previous issue of this document. An (R) symbol to the left of the document title indicates a complete revision of the document, including technical revisions. Change bars and (R) are not used in original publications, nor in documents that contain editorial changes only.

6.2 Patents

One or more patents may apply to one or more aspects of the standards or the entire standard. By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. The patent holder(s) has, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license for the purpose of complying with the standard. Details may be obtained from SAE International at http://www.sae.org/standardsdev/patents.htm.

PREPARED BY SAE FUEL CELL STANDARDS COMMITTEE

APPENDIX A - FAST FRAME CHECK SEQUENCE (FCS)

A.1 FAST FRAME CHECK SEQUENCE IMPLEMENTATION

The following reference implementation of the CRC-CCITT FCS algorithm is an example for the correct calulation method, which also takes care of the transparency character(s) (see 5.2.2.4).

In general, the CRC calculation steps are:

a. Use IrDA-Data-Block for calculation.

b. Invert data bytes before each XOR operation.

c. Reverse CRC result before final XOR.

d. Finally, check transparency.

A.1.1 FCS Computation Method

The following C code provides a CCITT-Tester for calculating the Frame Check Sequence:

Parts of the program may be used for online CCITT calculation at station side with the received IR-Data, as well as a table-based method.

```
// ---------------------------------------------------------------------------

// CCITT-Tester for SAE J2799 VN=01.00

// ---------------------------------------------------------------------------


// includes:

#include "stdafx.h"

#include <string.h>

#include <stdio.h>

#include <conio.h>


// CRC parameters (default values are for CCITT 16 Bit):

const int order=16;

const unsigned long polynom=0x1021;

const int direct=1;

const unsigned long crcinit=0x0000;

const unsigned long crcxor=0x00;
```

```
const int refin=1;

const int refout=1;


// 'order' [1..32] is the CRC polynom order, counted without the leading '1' bit

// 'polynom' is the CRC polynom without leading '1' bit

// 'direct' [0,1] specifies the kind of algorithm: 1=direct, no augmented zero bits

// 'crcinit' is the initial CRC value belonging to that algorithm

// 'crcxor' is the final XOR value

// 'refin' [0,1] specifies if a data byte is reflected before processing (UART) or not

// 'refout' [0,1] specifies if the CRC will be reflected before XOR


// SAE J2799 msg control codes

#define XBOF_sym 0xFF // Extra Begin of Frame

#define BOF_sym 0xC0  // Begin of Frame

#define EOF_sym 0xC1 // End of Frame

#define CE_sym 0x7D   // Control Escape


// Data character string

// no transparency

unsigned char msgdata[]="|ID=SAE␣J2799|VN=01.00|TV=0119.0|RT=H70|FC=Halt|MP=050.0|MT=273.0|";

// with transparency

unsigned char msgdata_2[]="|ID=SAE␣J2799|VN=01.00|TV=0119.0|RT=H70|FC=Dyna|MP=025.1|MT=234.0|";


// internal global values:

unsigned long crcmask;

unsigned long crchighbit;

unsigned long crcinit_direct;

unsigned long crcinit_nondirect;

unsigned long crctab[256];
```

```
// subroutines
unsigned long reflect (unsigned long crc, int bitnum) {
// reflects the lower 'bitnum' bits of 'crc'
unsigned long i, j=1, crcout=0;
for (i=(unsigned long)1<<(bitnum-1); i; i>>=1) {
if (crc & i) crcout|=j;
j<<= 1;
}
return (crcout);
}

void generate_crc_table() {
// make CRC lookup table used by table algorithms
int i, j;
unsigned long bit, crc;
for (i=0; i<256; i++) {
crc=(unsigned long)i;
if (refin) crc=reflect(crc, 8);
crc<<= order-8;
for (j=0; j<8; j++) {
bit=crc & crchighbit;
crc<<= 1;
if (bit) crc^= polynom;
}

if (refin) crc=reflect(crc, order);
crc&= crcmask;
```

```
crctab[i]= crc;

}

}


unsigned long crctablefast (unsigned char* p, unsigned long len) {

// fast lookup table algorithm without augmented zero bytes, e.g., used in pkzip.

// only usable with polynom orders of 8, 16, 24 or 32.

unsigned long crc=crcinit_direct;

if (refin) crc=reflect(crc, order);

if (!refin) while (len--) crc=(crc << 8) ^ crctab[ ((crc >> (order-8)) & 0xff) ^ *p++];

else while (len--) crc=(crc >> 8) ^ crctab[ (crc & 0xff) ^ *p++];

if (refout^refin) crc=reflect(crc, order);

crc^= crcxor;

crc&= crcmask;

return(crc);

}


unsigned long crcbitbybitfast(unsigned char* p, unsigned long len) {

// fast bit by bit algorithm without augmented zero bytes.

// does not use lookup table, suited for polynom orders between 1...32.

unsigned long i, j, c, bit;

unsigned long crc=crcinit_direct;

for (i=0; i<len; i++) {

c=(unsigned long)*p++;

if (refin) c=reflect(c, 8);

for (j=0x80; j; j>>=1) {

bit=crc & crchighbit;

crc<<= 1;

if (c & j) bit^= crchighbit;
```

```
if (bit) crc^= polynom;
}
}

if (refout) crc=reflect(crc, order);
crc^= crcxor;
crc&= crcmask;
return(crc);
}

int msglen(unsigned char * str)
{
int i=0;
while(*(str++))
i++;
   return i;
}

unsigned long chk_transparency(unsigned int crc_in){
// transparency check
unsigned char crc_part1,crc_part2;
union{
unsigned long crc_value;
unsigned char crc_bytes[4];
}crc;
bool Transparency=false;
crc.crc_value=0;

crc_part1= (unsigned char)((crc_in>>8) & 0xff);
```

```
switch (crc_part1)

{

case XBOF_sym:

case BOF_sym:

case EOF_sym:

case CE_sym:

crc.crc_bytes[3]= CE_sym;

crc.crc_bytes[2]= (crc_part1^0x20);

Transparency=true;

break;

default:

crc.crc_bytes[2]= crc_part1;

crc.crc_bytes[3]= 0;


}


crc_part2= (unsigned char)(crc_in & 0xff);

switch (crc_part2)

{

case XBOF_sym:

case BOF_sym:

case EOF_sym:

case CE_sym:

crc.crc_bytes[1]= CE_sym;

crc.crc_bytes[0]= (crc_part2^0x20);

Transparency=true;

break;

default:

crc.crc_bytes[0]= crc_part2;

crc.crc_bytes[1]= 0;
```

```
}

if (Transparency)
return crc.crc_value;
else
return crc.crc_bytes[0] | (crc.crc_bytes[2]<<8);
    }

int _tmain(int argc, _TCHAR* argv[])
{

// test program for checking four different CRC computing types that are:
// crcbitfast() and crctablefast(), see above.
// parameters are at the top of this program.
// Result will be printed on the console.

int i,ii;
unsigned long bit, crc;
unsigned char recchar,*msgdata_ptr;

// at first, compute constant bit masks for whole CRC and CRC high bit

crcmask=((((unsigned long)1<<(order-1))-1)<<1)|1;
crchighbit=(unsigned long)1<<(order-1);

// generate lookup table
generate_crc_table();
```

```
// compute missing initial CRC value

if (!direct) {

crcinit_nondirect=crcinit;
crc=crcinit;
for (i=0; i<order; i++) {

bit=crc & crchighbit;
crc<<= 1;
if (bit) crc^= polynom;
}
crc&= crcmask;
crcinit_direct=crc;
}

else {

crcinit_direct=crcinit;
crc=crcinit;
for (i=0; i<order; i++) {

bit=crc & 1;
if (bit) crc^= polynom;
crc >>= 1;
if (bit) crc|= crchighbit;
}
```

```
crcinit_nondirect=crc;

}


// call CRC algorithms using the CRC parameters above and print result to the console


printf("\n");

printf("CCITT tester for SAE J2799 VN=01.00\nwritten on 20/05/2011 by J. Zaepf (GM APCE / ADAM OPEL AG)\n");

   printf("(based on CRC tester v1.3 written on 4th of February 2003 by Sven Reifegerste)\n");

printf("---------------------------------------------------------------------\n");

printf("\n");

printf("Parameters:\n");

printf("\n");

printf(" polynom            : 0x%x\n", polynom);

printf(" order              : %d\n", order);

printf(" crcinit            : 0x%x direct, 0x%x nondirect\n", crcinit_direct, crcinit_nondirect);

printf(" crcxor             : 0x%x\n", crcxor);

printf(" refin              : %d\n", refin);

printf(" refout             : %d\n", refout);

printf("\n");

   printf(" Default msg data 1:\n |ID=SAE␣J2799|VN=01.00|TV=0119.0|RT=H70|FC=Halt|MP=050.0|MT=273.0|\n\n");

printf(" Default msg data 2 (with transparency):\n
|ID=SAE␣J2799|VN=01.00|TV=0119.0|RT=H70|FC=Dyna|MP=025.1|MT=234.0|\n\n");

printf(" Enter msg data string or press '1' or '2' for default calculation:\n ");

ii=0;


recchar= _getch();

if (recchar==50)

msgdata_ptr= msgdata_2;

else if (recchar==49)
```