ssJ1587
User's Manual

Version 1.0
Revised February 20th, 2008
Created by the J1587 Experts

ssJ1587 Protocol Stack License

READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT CAREFULLY BEFORE OPENING THE PACKAGE CONTAINING THE PROGRAM DISTRIBUTION MEDIA (DISKETTES, CD, ELECTRONIC MAIL), THE COMPUTER SOFTWARE THEREIN, AND THE ACCOMPANYING USER DOCUMENTATION. THIS SOURCE CODE IS COPYRIGHTED AND LICENSED (NOT SOLD). BY OPENING THE PACKAGE CONTAINING THE SOURCE CODE, YOU ARE ACCEPTING AND AGREEING TO THE TERMS OF THIS LICENSE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT, YOU SHOULD PROMPTLY RETURN THE PACKAGE IN UNOPENED FORM, AND YOU WILL RECEIVE A REFUND OF YOUR MONEY. THIS LICENSE AGREEMENT REPRESENTS THE ENTIRE AGREEMENT CONCERNING THE J1587 PROTOCOL STACK BETWEEN YOU AND SIMMA SOFTWARE, INC. (REFERRED TO AS "LICENSOR"), AND IT SUPERSEDES ANY PRIOR PROPOSAL, REPRESENTATION, OR UNDERSTANDING BETWEEN THE PARTIES.

1. Corporate License Grant. Simma Software hereby grants to the purchaser (herein referred to as the “Client”), a non-exclusive license to use the J1587 protocol stack source code software (collectively referred to as the "Software”) as part of Client’s proprietary software applications. This license is for unlimited client applications that can be distributed to an unlimited number of computers/devices/products. Except as provided above, Client agrees to not assign, sublicense, transfer, pledge, lease, rent, or share the Software Code under this License Agreement.

2. Simma Software's Rights. Client acknowledges and agrees that the Software and the documentation are proprietary products of Simma Software and are protected under U.S. copyright law. Client further acknowledges and agrees that all right, title, and interest in and to the Software, including associated intellectual property rights, are and shall remain with Simma Software. This License Agreement does not convey to Client an interest in or to the Software, but only a limited right of use revocable in accordance with the terms of this License Agreement.

3. License Fees. The Client in consideration of the licenses granted under this License Agreement will pay a one-time license fee.

4. Term. This License Agreement shall continue until terminated by either party. Client may terminate this License Agreement at any time. Simma Software may terminate this License Agreement only in the event of a material breach by Client of any term hereof, provided that such shall take effect 60 days after receipt of a written notice from Simma Software of such termination and further provided that such written notice allows 60 days for Client to cure such breach and thereby avoid termination. Upon termination of this License Agreement, all rights granted to Client will terminate and revert to Simma Software. Promptly upon termination of this Agreement for any reason or upon discontinuance or abandonment of Client’s possession or use of the Software, Client must return or destroy, as requested by Simma Software, all copies of the Software in Client’s possession, and all other materials pertaining to the Software (including all copies thereof). Client agrees to certify compliance with such restriction upon Simma Software’s request.

5. Limited Warranty. Simma Software warrants, for Client’s benefit alone, for a period of one year (called the “Warranty Period”) from the date of delivery of the software, that during this period the Software shall operate substantially in accordance with the functionality described in the User's Manual. If during the Warranty Period, a defect in the Software appears, Simma Software will make all reasonable efforts to cure the defect, at no cost to the Client. Client agrees that the foregoing constitutes Client’s sole and exclusive remedy for breach by Simma Software of any warranties made under this Agreement. Simma Software is not responsible for obsolescence of the Software that may result from changes in Client’s requirements. The foregoing warranty shall apply only to the most current version of the Software issued from time to time by Simma Software. Simma Software assumes no responsibility for the use of superseded, outdated, or uncorrected versions of the licensed software. EXCEPT FOR THE WARRANTIES SET FORTH ABOVE, THE SOFTWARE, AND THE SOFTWARE CONTAINED THEREIN, ARE LICENSED "AS IS," AND SIMMA SOFTWARE DISCLAIMS ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

6. Limitation of Liability. Simma Software's cumulative liability to Client or any other party for any loss or damages resulting from any claims, demands, or actions arising out of or relating to this License Agreement shall not exceed the license fee paid to Simma Software for the use of the Software. In no event shall Simma Software be liable for any indirect, incidental, consequential, special, or exemplary damages or lost profits, even if Simma Software has been advised of the possibility of such damages. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO CLIENT.

7. Governing Law. This License Agreement shall be construed and governed in accordance with the laws of the State of Indiana.

8. Severability. Should any court of competent jurisdiction declare any term of this License Agreement void or unenforceable, such declaration shall have no effect on the remaining terms hereof.

9. No Waiver. The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

TABLE OF CONTENTS

Chapter 1

Introduction

ssJ1587 is high performance SAE J1587 protocol stack written in ANSI C. ssJ1587 adheres to both the SAE J1587 specification and to the software development best practices guidelines described in MISRA C. The purpose of this protocol stack is assist developers with accessing J1587 parameters which are embedded in J1708 messages.

ssJ1587 is a modularized design with an emphasis on software readability and performance. ssJ1587 is easy to understand and can be used on any CPU or DSP with or without an RTOS.

1.1 Features

- Performs parameter length integrity validation on J1587 parameters.
- Converts J1708 messages into J1587 parameters and presents to application layer.
- Decodes multi-parameter J1587 messages and presents parameters to application layer.
- Handles reception of multi-section J1587 messages and presents to application layer.

1.2 Source Files

Filenames	File Description
j1587.c	Core source file for ssJ1587.
j1587.h	Core header file for ssJ1587.
j1587app.c	Application layer file.
j1587app.h	Application header file.

Table 1-1: ssJ1587 files

1.3 Code and RAM usage

Code size varies based on compiler and compiler settings, but should be approximately 500 bytes. RAM usage varies with the size of the multi-section message buffer.

1.4 Requirements

The processor should have the following:

- A J1708 device driver which conforms to the API outlined in chapter 4 of this manual.

Chapter 2

Integration of ssJ1587

This chapter describes how to integrate ssJ1587 into your application. After this is complete, you will be able to receive and transmit J1587 messages. For implementation details, please see the chapters covering the API.

2.1 Integration Steps:

1. Implement the ssJ1708 API which is described in chapter 4.

2. Before using any of the ssJ587 module, make sure the module has been initialized by calling j1587_init. Typically j1587_init() is called shortly after power-on reset and before the application is started. j1587_init() should be called after the J1708 module has been initialized.

3. Call j1587_update_10ms() ever 10 milliseconds. This function will read messages from the J1708 receive buffer and convert them into J1587 parameters which are passed to j1587app_post(). This function also keeps the time base for multi-section message timeout.

4. Place your application specific firmware in the body of j1587app_post(). This function is called with every new j1587 parameter received from the j1708 network. For example, if your application is interested in the road speed parameter (PID 84), then you should handle the processing of PID 84 in this function. See example code in this document.

Chapter 3

ssJ1587 Application Example

The below example demonstrates how the application code is suppose to decode J1587 parameters. The j1587app_post function receives a pointer to a J1587 parameter (see j1587.h for definition). A J1587 parameter structure has four members of interest: mid, pid, buf, buf_len.

The 'mid' member represents the J1708 MID (Message ID). In J1587 the MID represents the source address of the transmitting ECU.

The 'pid' represents the J1587 PID (Parameter ID). This represents the ID of the parameter being transmitted.

The 'buf' parameter points to the data which corresponds to the PID. 'buf_len' indicates how much data is available for this particular parameter. Note, many parameters have a fixed length according to the J1587 specification.

```
void
j1587app_post ( j1587_pmtr_t *pmtr ) {

  switch( pmtr->pid ) {

    case PID_GLOBAL_REQUEST:
      /* process the global request (PID 0) */
      break;

    case PID_ROAD_SPEED:
      /* get new road speed */
      j1587app_road_spd = pmtr->buf[0];
      break;
  }

  return;
}
```

Chapter 4

J1708 Application Program Interface

The ssJ1587 uses a defined J1708 application program interface (API) which is described below. These functions receive and transmit J1708 messages.

Function Prototype	Function Description
uint8_t j1708_rx (uint8_t *buf, uint8_t *size)	Reads a J1708 from receive buffer
uint8_t j1708_tx (uint8_t *buf, uint8_t size, uint8_t pri)	Places a J1708 message in the transmit buffer.

Table 4-1: API functions

j1708_rx

Function Prototype:

uint8_t j1708_rx (uint8_t *msg, uint8_t *size);

Description:

If a valid J1708 message is available it will be returned by this function. Only messages with a valid checksum and correct inter-character spacing should be returned by this function.

Parameters:

msg: Points to memory where the J1708 message will be copied. The first location contains the J1708 MID.

size: Size of J1708 message pointed to by msg.

Return Value

Non-zero: Message was not read from receive buffer.

0: Message was read from receive buffer.

j1708_tx

Function Prototype:

uint_8 j1708_tx (uint8_t *buf, uint8_t size, uint8_t pri);

Description:

This function should copy the J1708 message pointed to by buf and store it into the J1708 transmit buffer. This function is responsible for calculating the checksum and appending it to the end of the message.

Parameters:

msg: Points to the J1708 message. First location is the J1708 MID.
size: Size of J1708 message pointed to by msg.
pri: Priority of J1708 message

Return Value:

Non-zero: Message was not written to the transmit buffer.
0: Message was written to the transmit buffer.

Chapter 5

Configuration

This chapter describes all configurable items of the ssJ1587 module. These items are stored as defines at the top of j1587.c.

J1587 Multi-Section Buffer Length

This defines the largest multi-section message which can be received.

```
#define J1587_MULTISEC_BUF_LEN                    50
```

J1587 Multi-Section Message Max Count

This defines the number of J1587 multi-section messages which can be received simultaneously.

```
#define J1587_MULTISEC_CNT                        2
```

J1587 Multi-Section Message Timeout

This defines the timeout for a multi-section message. If no new data is received within this time frame, then the message will be discarded. This defines has units of 10 milliseconds.

```
#define J1587_MULTISEC_TIMEOUT                    500
```