Project Name:	Lucid Motors - External communication
Project Key:	J12728
Department:	ENG
Document Type:	Software Detailed Design
Document Acronym:	SDD
Version:	V1.0
Author:	JAR
Quality Level:	Draft
Filename:	SD_ENG_J12728 - Lucid Motors - ExternalCommunication_V01-00.docx
Template:	ADD_ENG_Job-Number_Project-_V01-00.dotx

Phone:	(248) 524-2240
Email:	info@dsasystems.com

DSA Systems Inc. · 1389 Wheaton Dr. Suite 800 · Troy, MI · 48083

Table of Contents

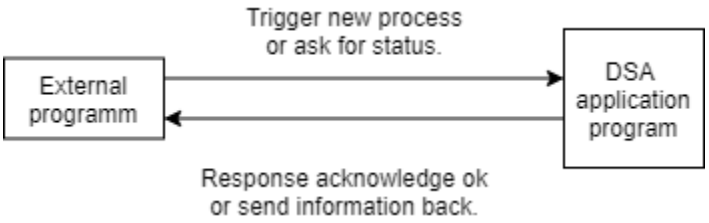
1 Revision History 2
2 Overview 3
3 Communication Description 4
3.1 MPB Template telegrams 5
3.1.1 Request Start data telegram MPB 5
3.1.2 Response Start data telegram MPB 5
3.1.3 Request Status data telegram MPB 5
3.1.4 Response Status data telegram MPB 5
3.2 Driveline Template telegrams 5
3.2.1 Request Start data telegram Driveline 6
3.2.2 Response Start data telegram Driveline 6
3.2.3 Request Status data telegram Driveline 6
3.2.4 Response Status data telegram Driveline 6
3.3 Battery line. 6
3.3.1 Request Start data telegram Battery line 6
3.3.2 Response Start data telegram Battery line 6
3.3.3 Request Status data telegram Battery line 6
3.3.4 Response Status data telegram Battery line 7
3.4 Mockup communication 8
4 Components 9
5 Interfaces 10
6 Product Configuration 11
7 References 12

1 Revision History

Version	Date	Author	Change
V1.0	7/02/2020	Alonso Ramirez	Initial version

2 Overview

The external communication software has been designed in a way that can connect with an external 3rd party software and exchange data using messages in telegram like format.



For this purpose, the DSA application program will act as the server side, waiting for a message to trigger a process, or a message to ask for the current status of the previously trigger process.

3 Communication Description

The program is expected to communicate using telegrams over TCP/IP layer port 6060 using DF1 protocol in a point to point manner (single client) where the DSA tester will expect communication coming in from the 3rd party side.

The telegrams have two main parts, telegram name and body. The telegram name is the one that identifies the type of telegram and can have one of the following string values:

- REQ
- RSP

In the body the telegrams structure or data elements are separated by curly brackets and semi-colon symbols. A structure starts with an identifier followed by a "{" symbol and ends with a "}" symbol. A structure can contain one or more data elements or substructures. A data element is terminated by a semi-colon ";". Data elements can be strings, floating point numeric or integer numeric.

Strings can start and end with quotes ("). Binary (non-printable) data within strings is represented byte-wise by their hexadecimal ASCII value with a leading "\x", e.g. "\x0D\x0A" for a carriage-return line feed. Outside the actual telegram data, spaces, carriage-returns, line feeds and tabs can occur for increased legibility

All telegrams should follow this format:

REQ{ "<cmd>"; 0; <counter>; ["<data>";] }

RSP{ "<cmd>"; <err>; <counter>; ["<data>";] }

The command entry or “cmd” entry can have one of the following values:

1. “START” to trigger the process.
2. “STATUS” to ask for the current status of the already started process.

<err>: This entry in the response telegram is a numeric error identifier. If error is not 0 the data will contain further information about it.

<counter>: The counter is an integer number that should be increase by the client every new message, the counter in the request and response messages for the same request/response pair shall be the same.

[<data>]: The number of data entries and its content will depend on the type of process to execute (location that is running i.e. MPB, Inverter line or Battery line) and the “command type”.

REQ{“START”;0;1;”BCM_flashfile.hex”;}

REQ{“STATUS”;0;1;}

The format of the response telegram (from DSA to external program) should look like these:

RSP{"START";0;1;”ACK_OK”}

RSP{"STATUS";0;2;”50%”}

RSP{"STATUS";0;3;”SN_WRITE_OK”}

RSP{"START_";1;4; "Command not recognized START_ "; }

<err> = [ErrorCodesDescription] (Defined in section 4).

3.1 MPB telegrams

Following is the definition of the different telegrams for the MBP (multipurpose box) location.

3.1.1 Request Start data telegram MPB

The request start telegram for the MBP location needs the following information:

3.1.2 Response Start data telegram MPB

The request start telegram for the MBP location needs the following information:

3.1.3 Request Status data telegram MPB

The request status telegram for the MBP location needs the following information:

3.1.4 Response Status data telegram MPB

The request status telegram for the MBP location needs the following information:

3.2 Driveline telegrams

The first element in all driveline telegrams should be the ECU short name:

1. MCU-F (Motor Control Unit Front)
2. MCU-R (Motor Control Unit Rear)

3.2.1 Request Start data telegram Driveline

The request start telegram for the Driveline location needs the following information:

3.2.2 Response Start data telegram Driveline

The request start telegram for the driveline location needs the following information:

3.2.3 Request Status data telegram Driveline

The request status telegram for the driveline location needs the following information:

3.2.4 Response Status data telegram Driveline

The request status telegram for the driveline location needs the following information:

3.3 Battery line telegrams.

Following is the definition of the different telegrams for the Battery line location.

3.3.1 Request Start data telegram Battery line

The request start telegram for the Battery line location needs the following information:

3.3.2 Response Start data telegram Battery line

The request start telegram for the Battery line location needs the following information:

3.3.3 Request Status data telegram Battery line

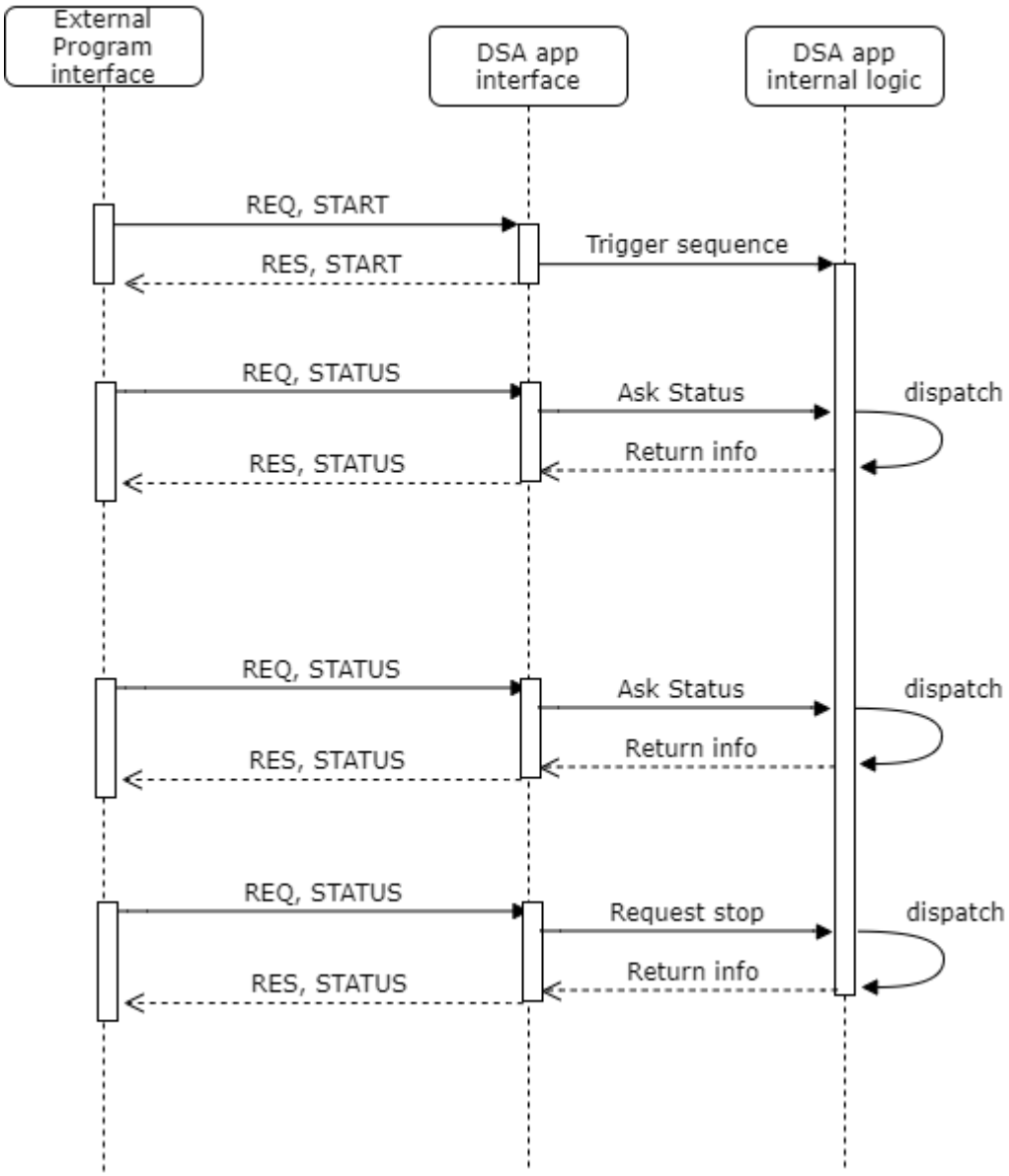
The request status telegram for the Battery line location needs the following information:

3.3.4 Response Status data telegram Battery line

The request status telegram for the Battery line location needs the following information:

3.4 Mockup communication

The following image depicts the flow between the External communication program and the DSA external communication interface.



4 Components

5 Interfaces

6 Product Configuration

This section specifies which items or properties of the product can be configured by customer.

ToDo: Need to determine what items should be configurable by the customer.

7 References

This section lists external references that are applicable in the context of this project.