User manual

# L9305 Evaluation Board and GUI Manual

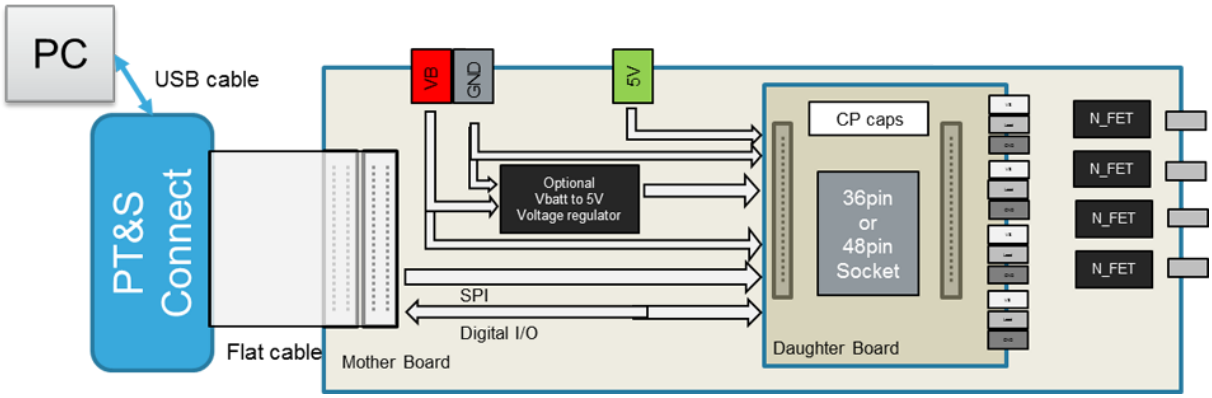
## Introduction

This document provides a description of L9305 Evaluation Board and dedicated Graphic User Interface, as well as guidelines for setting up an effective evaluation environment, both from the hardware and software point of view.

# 1 Evaluation hardware

## 1.1 Overview

Figure 1. **Evaluation hardware overview**




A complete evaluation hardware system for L9305, is composed of a motherboard and daughterboard, available in two different versions, equipped with the proper socket, depending on the L9305 package under evaluation, and a PT&S Connect Automotive microcontroller board.

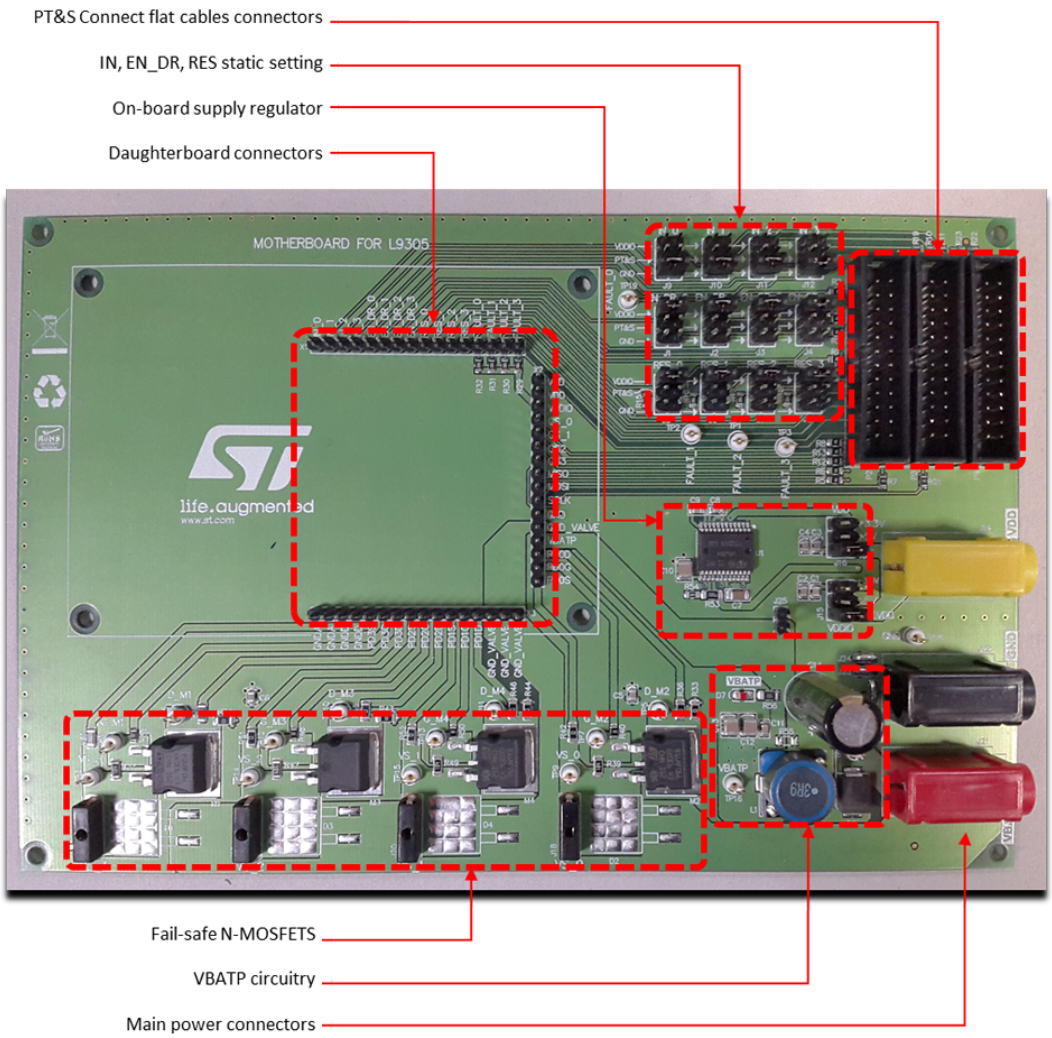
L9305 is mounted in the socket and daughterboard will be plugged on the motherboard, which will connect to the PT&s Connect box through flat cables; finally, PT&S Connect will communicate with the host pc through a USB cable.

For maximum flexibility and expandibility, up to 4 daughterboards can be stacked on motherboard, which will provide supply and PT&S Connect interface signals to all of them.

Description of the items listed above, plus the daughterboards stacking configuration and related external wirings, will be provided in the next paragraphs.

## 1.2 Motherboard

**Figure 2. Motherboard overview**




Motherboard is equipped with the following features (refer to Figure 2):

- 4 mm plugs, to connect VBAT, VDD and ground to an external power supply.
- Protection circuit to obtain VBATP from VBAT.
- on-board regulator, supplied by VBAT, to provide a 5 V and a 3.3 V regulated lines that can be used to feed VDD line.
- 4 N-channel fail-safe MOSFETs, connected to the proper L9305 driving pins.
- 2 mm plugs, to connect fail-safe NMOS sources to daughter boards.
- 3 flat-cable connectors for PT&S Connect interface.
- Jumpers to select static configuration of control signals or to route them to PT&S Connect.

### 1.2.1 Motherboard jumpers configuration

Next table summarizes functionality and configurations for the jumpers available on motherboard.

IN, EN_DR and RES signals are available for each of the allowed stacked daughterboards; proper jumper configuration will route the signals to each daughterboard (details on Section 1.3.4 Daughterboard jumpers configuration).

**Table 1. Motherboard jumpers**

| Jumper | Signal | Position | Function |
|---|---|---|---|
| J1..4 | EN_DR0..3 | VDDIO | Static setting |
| | | PT&S | Drive signal from PT&S Connect |
| | | GND | Static setting |
| J5..8 | RES_0..3 | VDDIO | Static setting |
| | | PT&S | Drive signal from PT&S Connect |
| | | GND | Static setting |
| J9..12 | IN_0..3 | VDDIO | Static setting |
| | | PT&S | Drive signal from PT&S Connect |
| | | GND | Static setting |
| J13 | - | soldered | Connect GND_VALVE to GND |
| J15 | VDDIO | +3.3 V | Connect VDDIO line to 3.3 V from on-board regulator |
| | | +5 V | Connect VDDIO line to 5 V from on-board regulator |
| | | EXT.VDD | Connect VDDIO line to J14 connector |
| J16 | VDD | +3.3 V | Connect VDDIO line to 3.3 V from on-board regulator |
| | | +5 V | Connect VDDIO line to 5 V from on-board regulator |
| | | EXT.VDD | Connect VDDIO line to J14 connector |
| J23 | - | Soldered | Connect GND_A to GND |
| J24 | - | Soldered | Connect GND_D to GND |
| J25 | - | Closed | On-board regulator enable |

## 1.3 Daughterboard

Two versions of daughterboards are available, equipped with different sockets, to accommodate L9305 in PWSSO36 or TQFP48 package. Interface to/from motherboard is the same for both versions; daughterboard_48 allows connection between L9305 four fail-safe drivers and the NMOS equipped on motherboard.

Daughterboard is equipped with the following features (refer to Figure 3 and Figure 4):

- Socket for L9305
- 2 mm plugs to connect up to 4 loads
- Jumpers for static signal configuration and stacked configuration setup
- Connectors to motherboard, replicated on top layer (highlighted yellow on the pictures), to allow stacked configuration

**Figure 3. Daughterboard - PWSSO36 version**

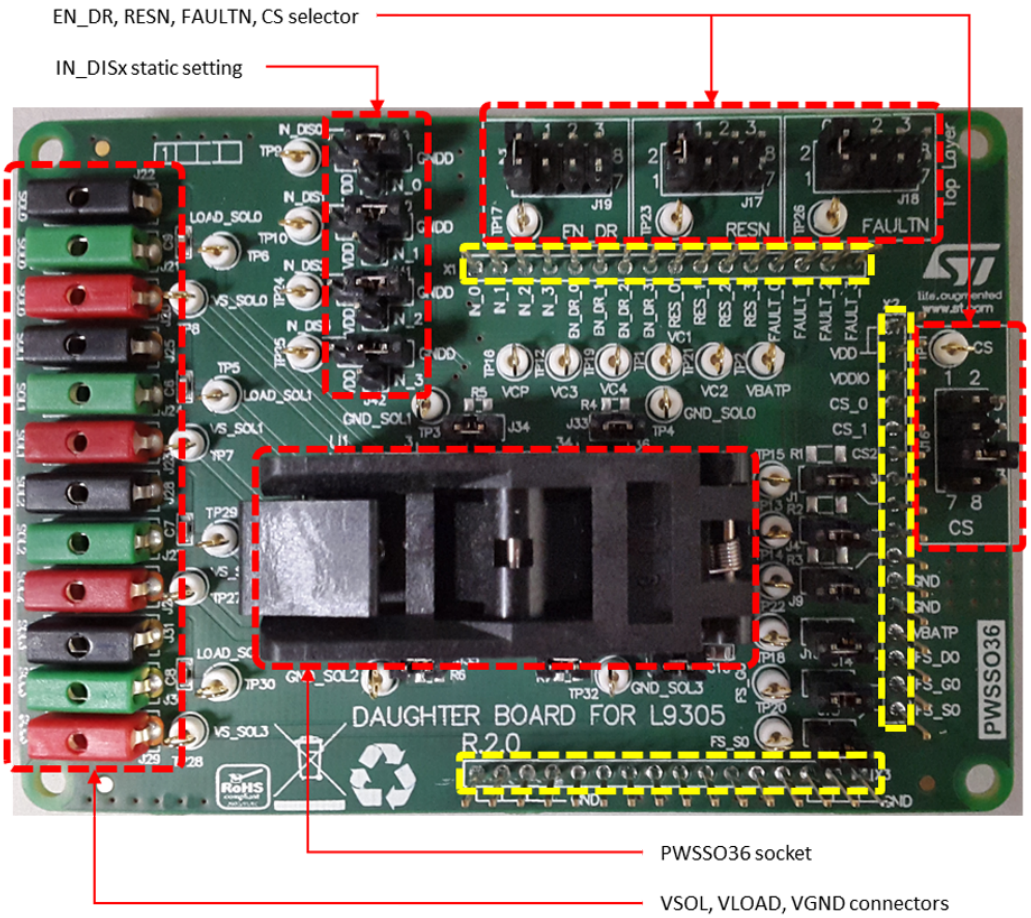
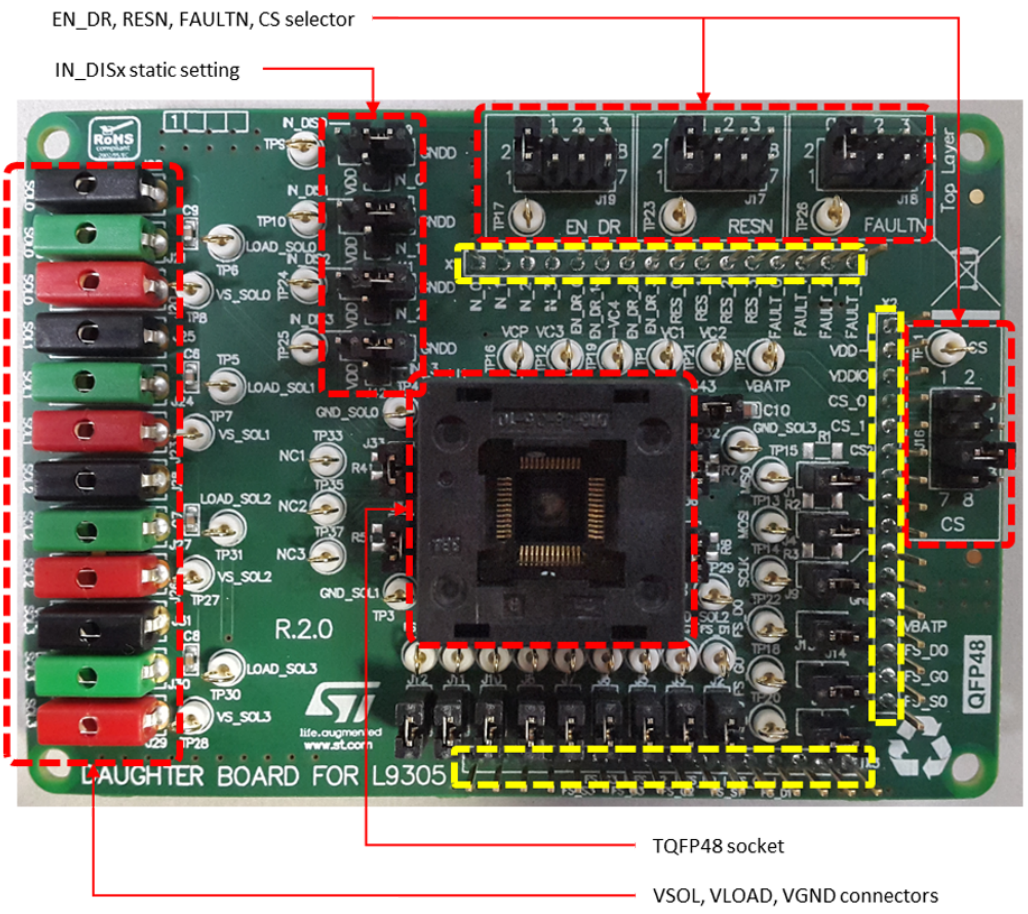


Figure 4. **Daughterboard - TQFP48 version**



### 1.3.1 Single daughterboard configuration

The Figure 5 shows a standard single daughterboard configuration, showing external wires needed for load connection (in this example, 4 loads in HS driver mode. When PSSO36 package type is used, only one fail-safe NMOS (M2) will be driven. When using TQFP48 package, all fail-safe NMOS can be used, as indicated on the Figure 6.

**Figure 5. Typical wiring with PWSSO36 daughterboard**

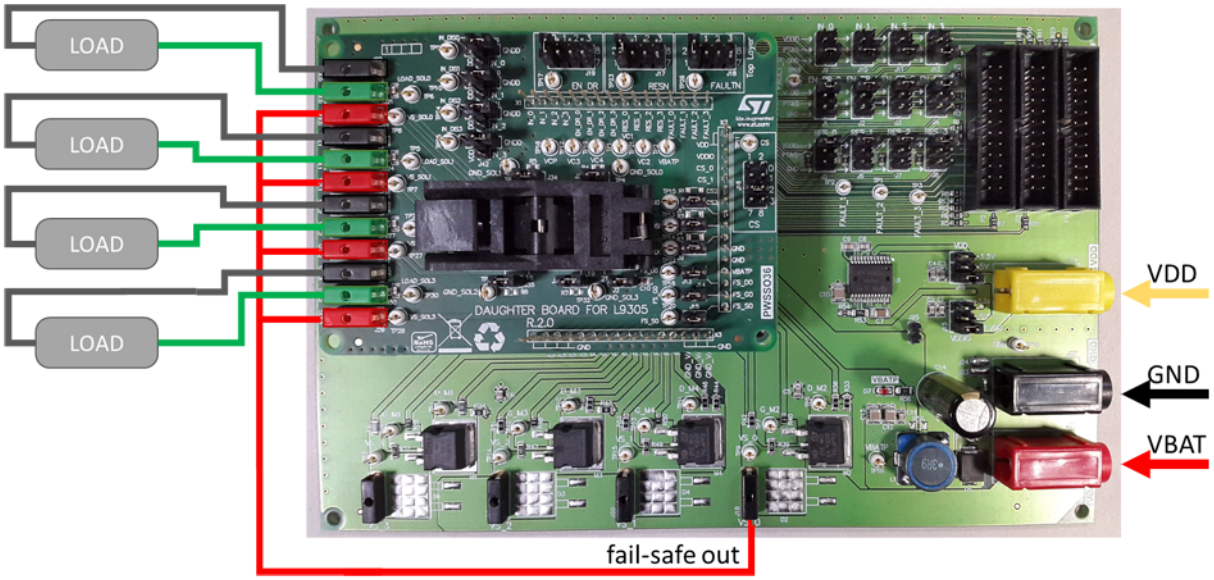
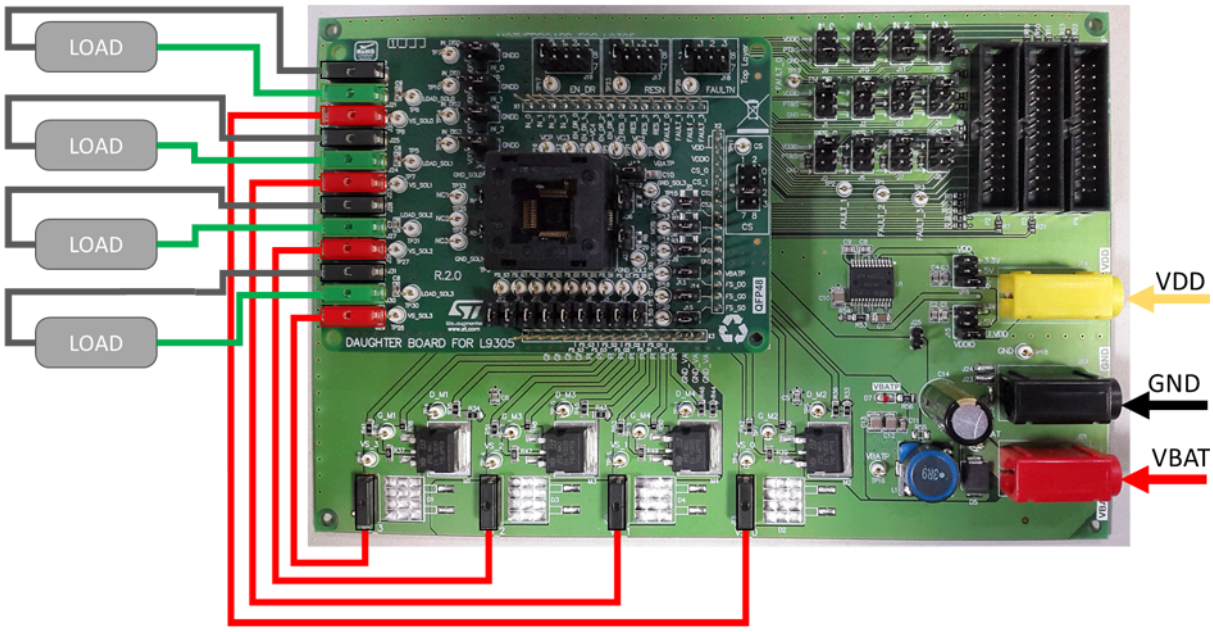



**Figure 6. Typical wiring with TQFP48 daughterboard**



### 1.3.2 Multi daughterboard (stacked) configuration

Figure 7. **Typical wiring with PWSSO36 daughterboard**

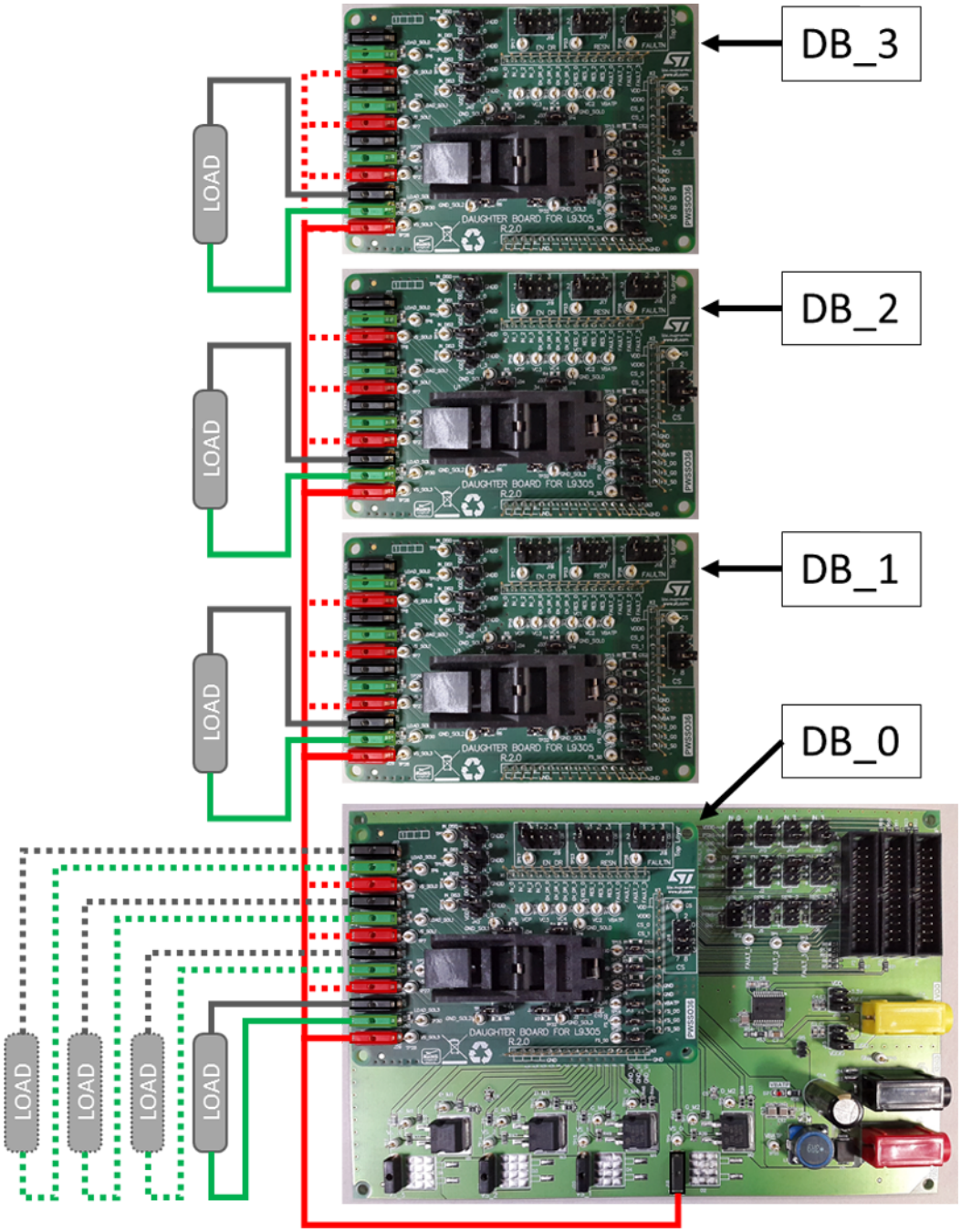
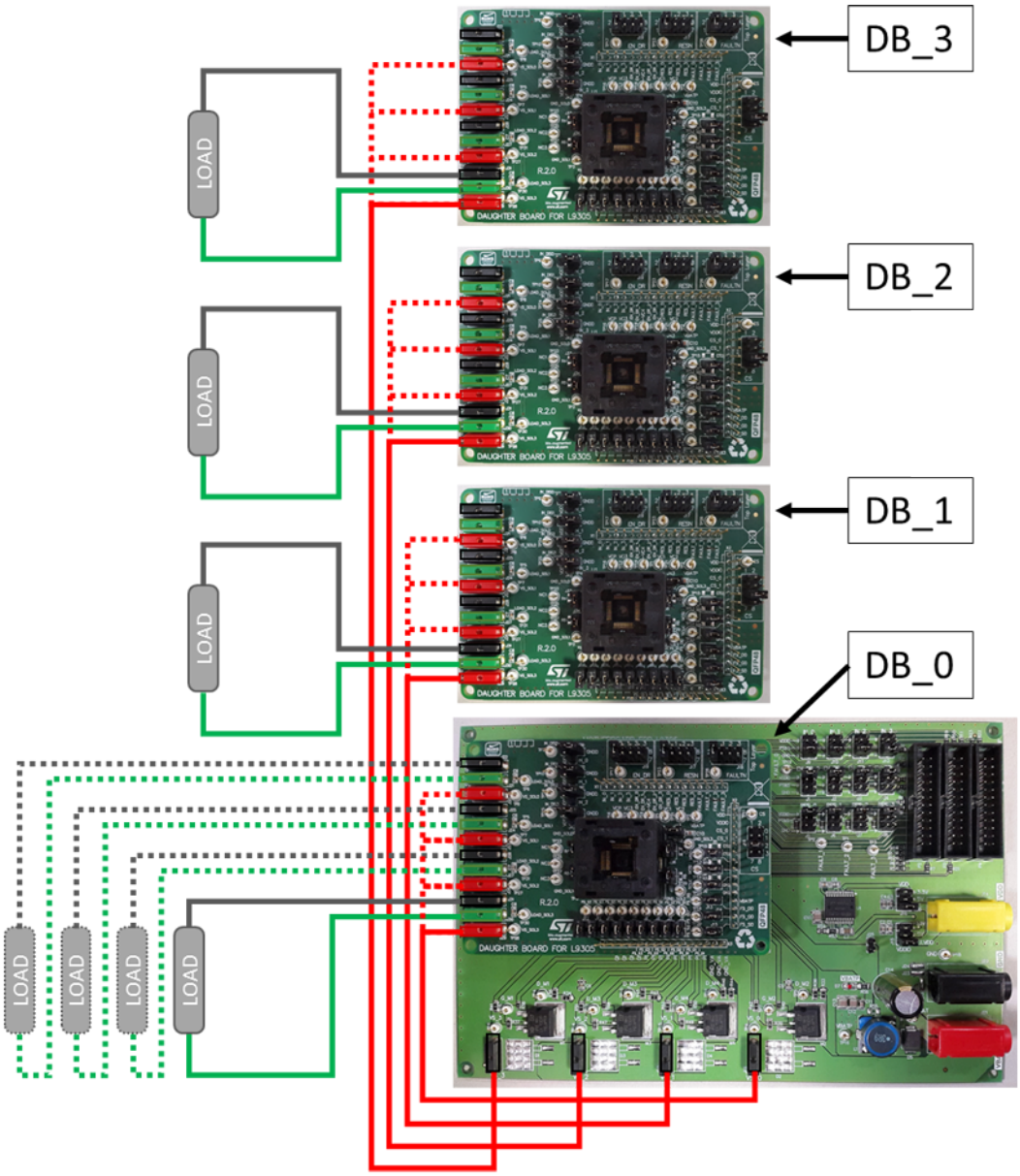


Figure 8. **Typical wiring with TQFP48 daughterboard**




The Figure 7 and Figure 8, show multi-daughterboard (or stacked) configuration, plus the wiring needed for load connection. A maximum of 4 daughterboards can be stacked. To keep picture readability, not all the possible load configurations are indicated. Dotted wires are optional; the same wiring can be applied on any of the stacked daughterboards.

*Note: when stacking daughterboards, a firm connection between the single-in-line headers can be tricky to reach, because of socket's thickness; possible solutions are either using boards with soldered parts or connecting the headers through proper risers (not provided with the boards).*

Stacked configuration, allow to expand load driving capability of a system, by using up to four L9305 controlled by the same pc. Proper jumper settings, ensure the possibility for the control software to directly access to each of the stacked boards, and to configure it in a dedicated way.

Depending on the version of daughterboard used, fail-safe MOS driving capabilities are affected too. When using PWSSO36 daughterboard, no matter how many boards are stacked, only one fail-safe (M2) will be driven by one of the stacked boards, because there's no connection between daughterboard's fail-safe driver lines and motherboard's NMOS other than M2: in this case, a single NMOS would be required to source all the current needed by potentially up to 16 loads.

Only the daughterboard that will drive the fail-safe, must have jumpers FS_D0, FS_G0, FS_S0 closed (see the Table 5 for jumpers details); other daughterboards composing the stack, must have these jumpers opened.

Stacking TQFP48 daughterboards can be less critical in terms of current required by fail-safe NMOS, because the system can be configured to have each daughterboard driving one of the 4 available fail-safe NMOS. Also in this case, the user has to configure each daughterboard's FS_D, FS_G and FS_S jumpers in such a way that each fail-safe NMOS is driven by one and only one daughterboard.

As a clarification, on Section 1.3.4 Daughterboard jumpers configuration some configuration examples are reported, showing how to configure jumpers in different single or multi board scenarios.

Following table summarizes the possible scenarios in terms of loads and fail-safe driving capabilities.

**Table 2. Stacked configuration driving capabilities**

| N. of stacked daughterboards | PWSSO36 daughterboard | | TQFP48 daughterboard | |
|---|---|---|---|---|
| | Pre-drivers | Loads | Pre-drivers | Loads |
| 1 | 1 | 4 | 4 | 4 |
| 2 | | 8 | | 8 |
| 3 | | 12 | | 12 |
| 4 | | 16 | | 16 |

### 1.3.3 Static signals routing

General note on signals routing in stacked configuration: EN_DR and RESn signals provided by motherboard could potentially be connected to any of the correspondant lines on stacked daughterboards by applying the proper jumper setting; the same can be done for FAULTn signal coming from daughterboard "x": it can be routed to any of the FAULTn lines on motherboard. However, to avoid confusion, it is strongly suggested to respect coherence among signals index (i.e: daughterboard 0 will receive EN_DR_0, RESn_0 and will provide FAULTn on FAULTn_0; the same for daughterboard 1, etc.). This criteria will be used in the following configuration instructions.

CS signal will not strictly follow this rule, because of a different mapping of CS values in PT&S Connect firmware (details on Section 2.4 CS GUI mapping and multi daughterboard usage); however, setting for proper functioning of a multi-daughterboard configuration will be proposed.

#### *1.3.3.1 IN_DISx*

IN_DISx jumper allows static enabling or disabling of channel "x" IN control signal when in hardware mode. In software mode, it must be set in "IN_x" position, to allow PT&S Connect to provide PWM signal.

**Figure 9. IN_DISx jumpers**

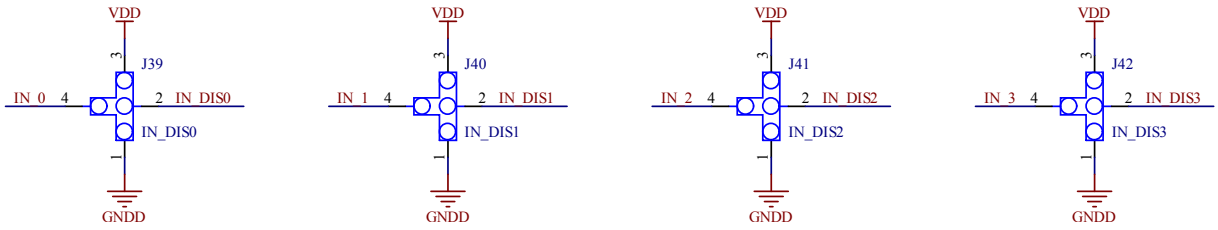



**Table 3. IN_DISx jumper details**

| Jumper | Position | Function |
|---|---|---|
| J39, J40, J41, J42 | GNDD | IN_DISx enabled |
| | IN_x | IN_DISx controlled via PT&S Connect |
| | VDD | IN_DISx disabled |

### 1.3.3.2 *EN_DR, RESn, FAULTn, CS*

Motherboard provides 4 of these signals (EN_DR[0..3], RESn[0..3], FAULTn[0..3], CS[0..3]), to allow the stacked configuration with up to 4 daughterboards. These jumpers allow to select the correct set of signals to/from the daughterboard in use.

**Figure 10. EN_DR, RESn, FAULTn, CS jumpers**

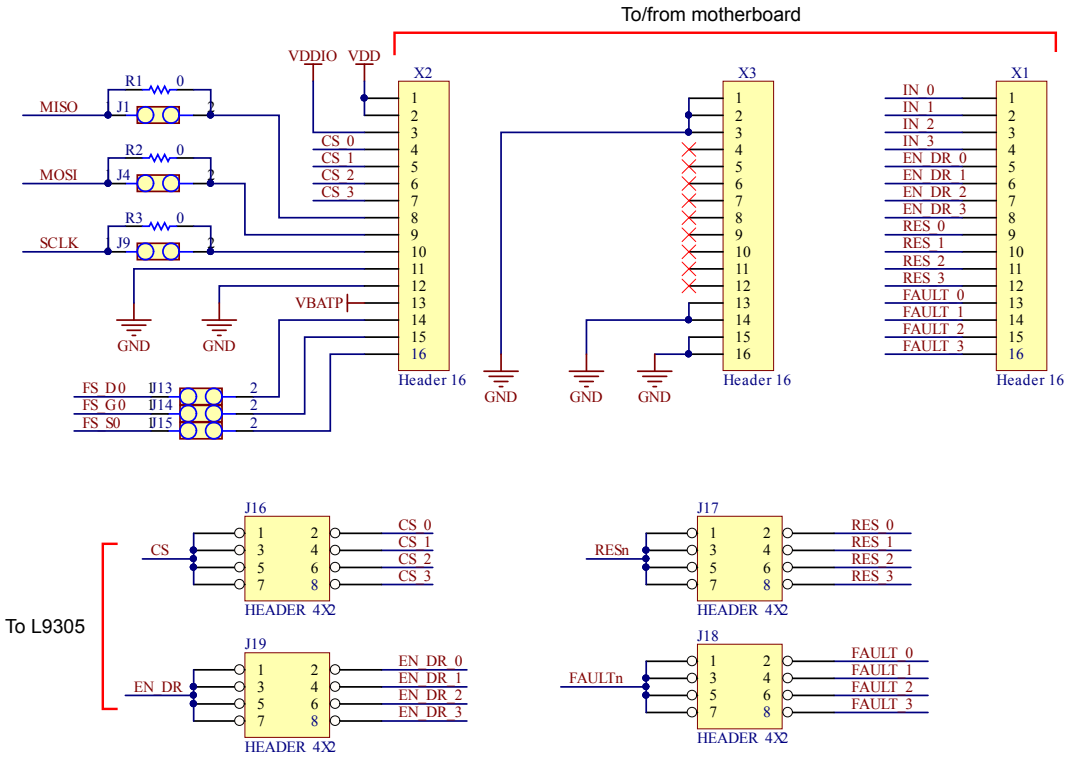



**Table 4. EN_DR, RESn, FAULTn, CS jumpers detail**

| Jumper | Position | Function |
|---|---|---|
| J16 | 0 | Connect CS_0 to daughterboard's CS |
| | 1 | Connect CS_1 to daughterboard's CS |
| | 2 | Connect CS_2 to daughterboard's CS |
| | 3 | Connect CS_3 to daughterboard's CS |
| J17 | 0 | Connect RESn_0 to daughterboard's RES |
| | 1 | Connect RESn_1 to daughterboard's RES |
| | 2 | Connect RESn_2 to daughterboard's RES |
| | 3 | Connect RESn_3 to daughterboard's RES |
| J18 | 0 | Connect FAULT_0 to daughterboard's FAULTn |
| | 1 | Connect FAULT_1 to daughterboard's FAULTn |
| | 2 | Connect FAULT_2 to daughterboard's FAULTn |
| | 3 | Connect FAULT_3 to daughterboard's FAULTn |
| J19 | 0 | Connect EN_DR_0 to daughterboard's EN_DR |
| | 1 | Connect EN_DR_1 to daughterboard's EN_DR |
| | 2 | Connect EN_DR_2 to daughterboard's EN_DR |
| | 3 | Connect EN_DR_3 to daughterboard's EN_DR |

#### 1.3.3.3 *Other jumpers for PWSSO36 daughterboard*

Next table summarizes remaining jumpers for PWSSO36 daughterboard, and their typical configuration for a single daughterboard setup.

**Table 5. Misc jumpers for PWSSO36 daughterboard**

| Jumper | Position | Function |
|---|---|---|
| J1 | Closed | Connect MISO from L9305 to motherboard |
| J2 | Closed | C10 connected to VDD |
| J4 | Closed | Connect MOSI from L9305 to motherboard |
| J9 | Closed | Connect SCLK from L9305 to motherboard |
| J13 | Closed | Connect FS_D0 from L9305 to motherboard |
| J14 | Closed | Connect FS_G0 from L9305 to motherboard |
| J15 | Closed | Connect FS_S0 from L9305 to motherboard |
| J32 | Soldered | Connect GNDA to GND |
| J33 | Closed | Connect GND_SOL0 to GND_VALVE |
| J34 | Closed | Connect GND_SOL1 to GND_VALVE |
| J35 | Closed | Connect GND_SOL2 to GND_VALVE |
| J36 | Closed | Connect GND_SOL3 to GND_VALVE |
| J37 | Soldered | Connect GNDD to GND |
| J38 | Soldered | Connect GND_VALVE to GND |

#### 1.3.3.4 *Other jumpers for TQFP48 daughterboard*

Next table summarizes remaining jumpers for TQFP48 daughterboard, and their typical configuration for a single daughterboard setup.

**Table 6. Misc jumpers for TQFP48 daughterboard**

| Jumper | Position | Function |
|---|---|---|
| J1 | Closed | Connect MISO from L9305 to motherboard |
| J2 | Closed | Connect FS_D1 from L9305 to motherboard |
| J3 | Closed | Connect FS_G1 from L9305 to motherboard |
| J4 | Closed | Connect MOSI from L9305 to motherboard |
| J5 | Closed | Connect FS_S1 from L9305 to motherboard |
| J6 | Closed | Connect FS_D2 from L9305 to motherboard |
| J7 | Closed | Connect FS_G2 from L9305 to motherboard |
| J8 | Closed | Connect FS_S2 from L9305 to motherboard |
| J9 | Closed | Connect SCLK from L9305 to motherboard |
| J10 | Closed | Connect FS_D3 from L9305 to motherboard |
| J11 | Closed | Connect FS_G3 from L9305 to motherboard |
| J12 | Closed | Connect FS_S3 from L9305 to motherboard |
| J13 | Closed | Connect FS_D0 from L9305 to motherboard |
| J14 | Closed | Connect FS_G0 from L9305 to motherboard |
| J15 | Closed | Connect FS_S0 from L9305 to motherboard |
| J32 | Soldered | Connect GNDA to GND |
| J33 | Closed | Connect GND_SOL0 to GND_VALVE |
| J34 | Closed | Connect GND_SOL1 to GND_VALVE |
| J35 | Closed | Connect GND_SOL2 to GND_VALVE |
| J36 | Closed | Connect GND_SOL3 to GND_VALVE |
| J37 | Soldered | Connect GNDD to GND |
| J38 | Soldered | Connect GND_VALVE to GND |
| J43 | Closed | Connect C10 connected to VDD |

### 1.3.4 Daughterboard jumpers configuration

Examples of single-multi daughterboard scenarios, with different fail-safe driving setups, and their corresponding jumpers configuration. For motherboard jumpers, refer to Section 1.2.1 Motherboard jumpers configuration.

#### *1.3.4.1 Daughterboard PWSSO36 configuration*

**Table 7. Multi-daughterboard_36 configuration**

| Jumper | Function | Daughterboard number | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 |
| J1 | MISO to motherboard | Closed | Closed | Closed | Closed |
| J2 | C10 connected to VDD | Closed | Closed | Closed | Closed |
| J4 | MOSI from motherboard | Closed | Closed | Closed | Closed |
| J9 | SCLK from motherboard | Closed | Closed | Closed | Closed |
| J13 | FS_D0 to motherboard | Closed | Opened | Opened | Opened |
| J14 | FS_G0 to motherboard | Closed | Opened | Opened | Opened |
| J15 | FS_S0 to motherboard | Closed | Opened | Opened | Opened |
| J16 | CS | 2 | 3 | 1 | 0 |
| J17 | RESn from motherboard | 0 | 1 | 2 | 3 |
| J18 | FAULTn to motherboard | 0 | 1 | 2 | 3 |
| J19 | EN_DR from motherboard | 0 | 1 | 2 | 3 |
| J32 | GNDA to GND | Soldered | Soldered | Soldered | Soldered |
| J33 | GND_SOL0 to GND_VALVE | Closed | Closed | Closed | Closed |
| J34 | GND_SOL1 to GND_VALVE | Closed | Closed | Closed | Closed |
| J35 | GND_SOL2 to GND_VALVE | Closed | Closed | Closed | Closed |
| J36 | GND_SOL3 to GND_VALVE | Closed | Closed | Closed | Closed |
| J37 | GNDD to GND | Soldered | Soldered | Soldered | Soldered |
| J38 | GND_VALVE to GND | Soldered | Soldered | Soldered | Soldered |
| J39 | IN_DIS0 | GNDD | GNDD | GNDD | GNDD |
| J40 | IN_DIS1 | GNDD | GNDD | GNDD | GNDD |
| J41 | IN_DIS2 | GNDD | GNDD | GNDD | GNDD |
| J42 | IN_DIS3 | GNDD | GNDD | GNDD | GNDD |

For single daughterboard configuration, refer to the Figure 5 for wiring and consider only column “daughterboard number 0” of the Table 7.

For multi daughterboard configuration, refer to the Figure 7 for wiring; FS_D0, FS_G0 and FS_S0 must be closed only on one of the stacked boards.

#### 1.3.4.2 *Daughterboard TQFP48 configuration*

**One fail-safe MOS driven**

**Table 8. Multi-daughterboard_48 configuration - single fail-safe**

| Jumper | Function | Daughterboard number | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 |
| J1 | MISO to motherboard | Closed | Closed | Closed | Closed |
| J2 | FS_D1 to motherboard | Opened | Opened | Opened | Opened |
| J3 | FS_G1 to motherboard | Opened | Opened | Opened | Opened |
| J4 | MOSI from motherboard | Closed | Closed | Closed | Closed |
| J5 | FS_S1 to motherboard | Opened | Opened | Opened | Opened |
| J6 | FS_D2 to motherboard | Opened | Opened | Opened | Opened |
| J7 | FS_G2 to motherboard | Opened | Opened | Opened | Opened |
| J8 | FS_S2 to motherboard | Opened | Opened | Opened | Opened |
| J9 | SCLK from motherboard | Closed | Closed | Closed | Closed |
| J10 | FS_D3 to motherboard | Opened | Opened | Opened | Opened |
| J11 | FS_G3 to motherboard | Opened | Opened | Opened | Opened |
| J12 | FS_S3 to motherboard | Opened | Opened | Opened | Opened |
| J13 | FS_D0 to motherboard | Closed | Opened | Opened | Opened |
| J14 | FS_G0 to motherboard | Closed | Opened | Opened | Opened |
| J15 | FS_S0 to motherboard | Closed | Opened | Opened | Opened |
| J16 | CS | 2 | 3 | 1 | 0 |
| J17 | RESn from motherboard | 0 | 1 | 2 | 3 |
| J18 | FAULTn to motherboard | 0 | 1 | 2 | 3 |
| J19 | EN_DR from motherboard | 0 | 1 | 2 | 3 |
| J32 | GNDA to GND | Soldered | Soldered | Soldered | Soldered |
| J33 | GND_SOL0 to GND_VALVE | Closed | Closed | Closed | Closed |
| J34 | GND_SOL1 to GND_VALVE | Closed | Closed | Closed | Closed |
| J35 | GND_SOL2 to GND_VALVE | Closed | Closed | Closed | Closed |
| J36 | GND_SOL3 to GND_VALVE | Closed | Closed | Closed | Closed |
| J37 | GNDD to GND | Soldered | Soldered | Soldered | Soldered |
| J38 | GND_VALVE to GND | Soldered | Soldered | Soldered | Soldered |
| J39 | IN_DIS0 | GNDD | GNDD | GNDD | GNDD |
| J40 | IN_DIS1 | GNDD | GNDD | GNDD | GNDD |
| J41 | IN_DIS2 | GNDD | GNDD | GNDD | GNDD |
| J42 | IN_DIS3 | GNDD | GNDD | GNDD | GNDD |
| J43 | C10 connected to VDD | Closed | Closed | Closed | Closed |

With settings reported on , user can refer to the same wiring diagram for daughterboard_36 (see the Figure 5 and Figure 7), because they'll make use of a single fail-safe. With the highlighted settings, daughterboard_0 will drive fail-safe NMOS M2 on motherboard. User can decide to drive any different NMOS (by any daughterboard) by closing the correspondant FS_S, FS_D, FS_G lines and opening the same lines on other daughterboards.

### More fail-safe MOS driven

**Table 9. Multi-daughterboard_48 configurations - multiple fail-safe**

| Jumper | Function | Daughterboard number | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 |
| J1 | MISO to motherboard | Closed | Closed | Closed | Closed |
| J2 | FS_D1 to motherboard | Opened | Closed | Opened | Opened |
| J3 | FS_G1 to motherboard | Opened | Closed | Opened | Opened |
| J4 | MOSI from motherboard | Closed | Closed | Closed | Closed |
| J5 | FS_S1 to motherboard | Opened | Closed | Opened | Opened |
| J6 | FS_D2 to motherboard | Opened | Opened | Closed | Opened |
| J7 | FS_G2 to motherboard | Opened | Opened | Closed | Opened |
| J8 | FS_S2 to motherboard | Opened | Opened | Closed | Opened |
| J9 | SCLK from motherboard | Closed | Closed | Closed | Closed |
| J10 | FS_D3 to motherboard | Opened | Opened | Opened | Closed |
| J11 | FS_G3 to motherboard | Opened | Opened | Opened | Closed |
| J12 | FS_S3 to motherboard | Opened | Opened | Opened | Closed |
| J13 | FS_D0 to motherboard | Closed | Opened | Opened | Opened |
| J14 | FS_G0 to motherboard | Closed | Opened | Opened | Opened |
| J15 | FS_S0 to motherboard | Closed | Opened | Opened | Opened |
| J16 | CS | 2 | 3 | 1 | 0 |
| J17 | RESn from motherboard | 0 | 1 | 2 | 3 |
| J18 | FAULTn to motherboard | 0 | 1 | 2 | 3 |
| J19 | EN_DR from motherboard | 0 | 1 | 2 | 3 |
| J32 | GNDA to GND | Soldered | Soldered | Soldered | Soldered |
| J33 | GND_SOL0 to GND_VALVE | Closed | Closed | Closed | Closed |
| J34 | GND_SOL1 to GND_VALVE | Closed | Closed | Closed | Closed |
| J35 | GND_SOL2 to GND_VALVE | Closed | Closed | Closed | Closed |
| J36 | GND_SOL3 to GND_VALVE | Closed | Closed | Closed | Closed |
| J37 | GNDD to GND | Soldered | Soldered | Soldered | Soldered |
| J38 | GND_VALVE to GND | Soldered | Soldered | Soldered | Soldered |
| J39 | IN_DIS0 | GNDD | GNDD | GNDD | GNDD |
| J40 | IN_DIS1 | GNDD | GNDD | GNDD | GNDD |
| J41 | IN_DIS2 | GNDD | GNDD | GNDD | GNDD |
| J42 | IN_DIS3 | GNDD | GNDD | GNDD | GNDD |
| J43 | C10 connected to VDD | Closed | Closed | Closed | Closed |

With settings reported on the , daughterboard_0 will drive fail-safe NMOS M2, daughterboard_1 will drive M4, daughterboard_2 will drive M3 and daughterboard_3 will drive M1. User can refer to the wiring diagram on the Figure 8.

## 1.4 Board layout

### PWSSO36 board layout

Figure 11. **Top layer**

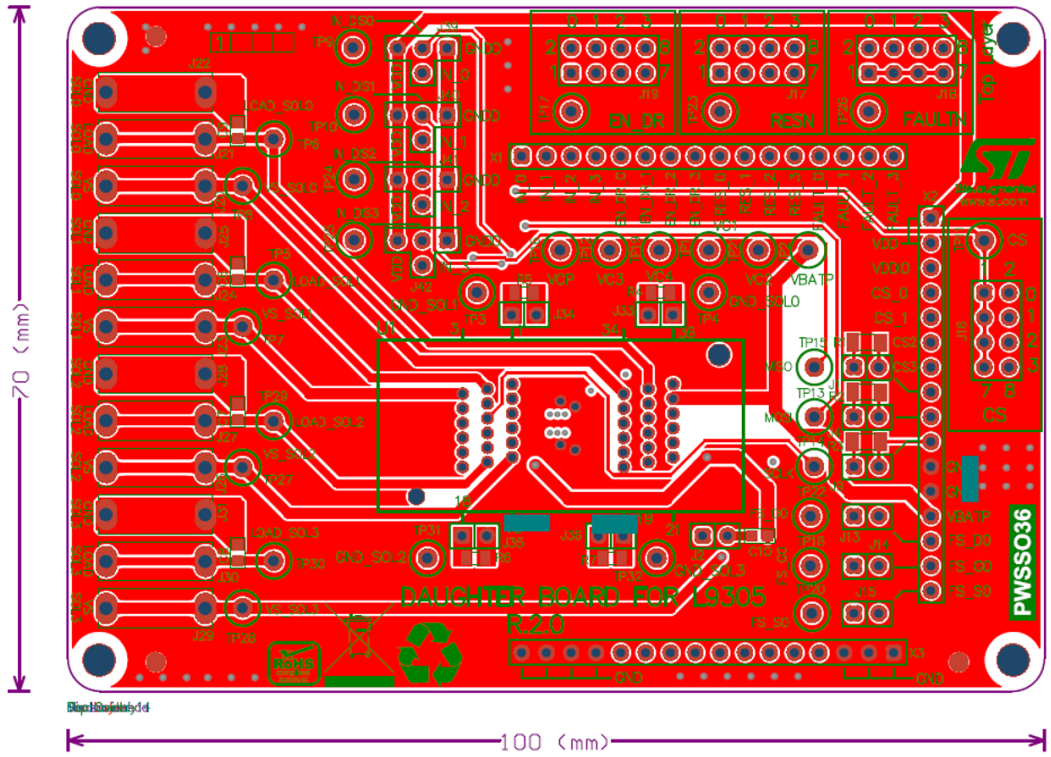
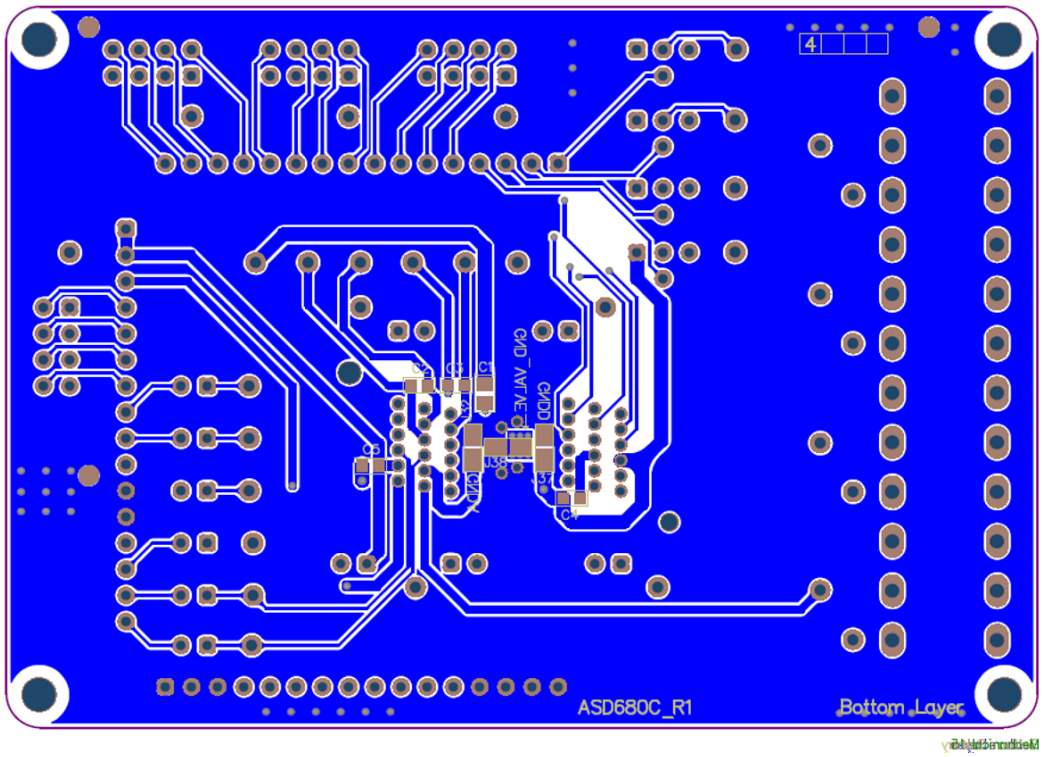



Figure 12. **Bottom layer**



### TQFP48 board layout

Figure 13. **Top layer**

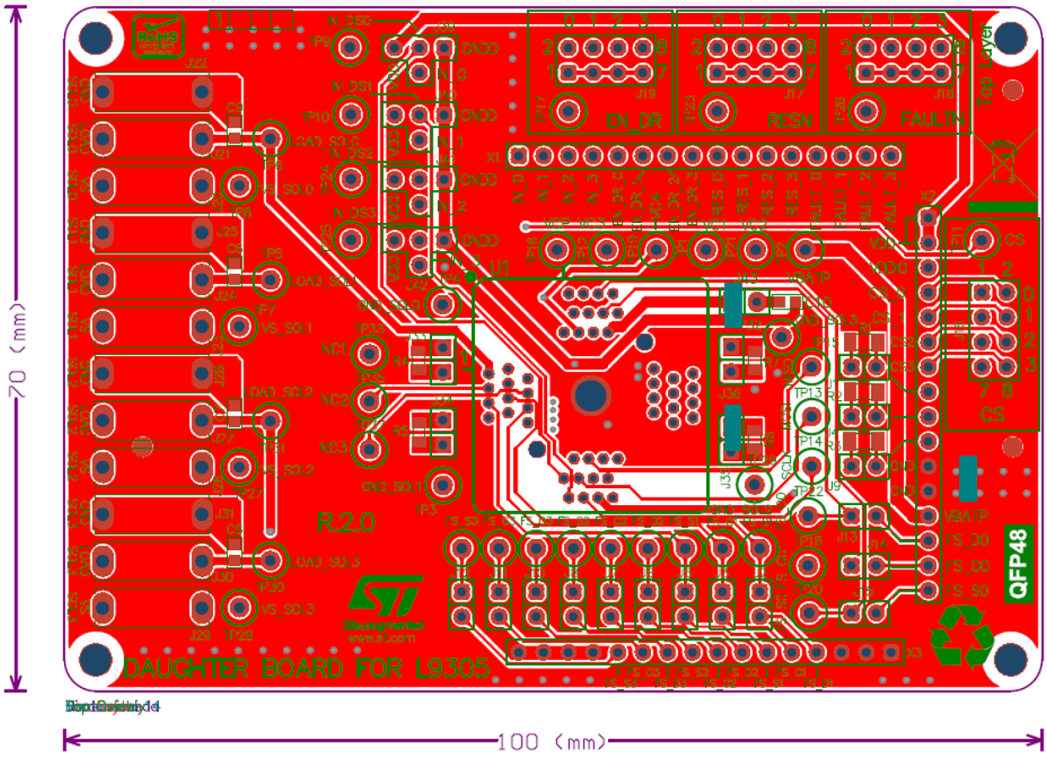
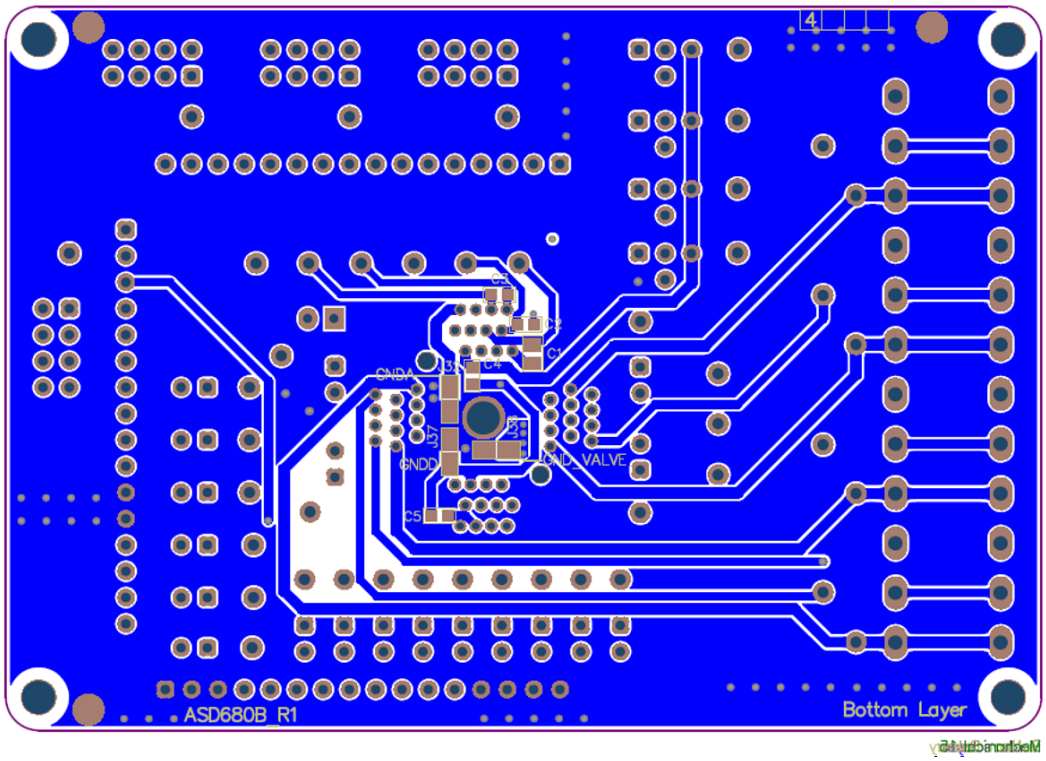



Figure 14. **Bottom layer**



# 2 Evaluation software

Product Validation Software is a software tool that allows user interacting with a ASIC device. This software allows user using multiple interfaces to stimulate a device such as SPI, MSC, PWM, ADC and GPIO. User is able to send a command to the device (through one of the supported communication protocol) and analyze output in an intuitive and friendly Graphical User Interface. All internal device registers, and their related field structures, are shown in a graphical way in order to help user during testing activities. User can also arrange a set of commands together in order to create a custom test sequence for validation process (script).

The software is designed to be used with different APG's product, however, next sections will focus on the usage with L9305.

## 2.1 System overview

System is built around three main components:

**Figure 15. System block diagram**

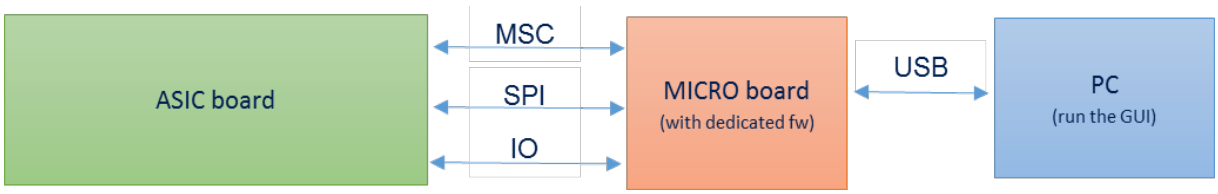



**Figure 16. PT&S Connect Micro board**

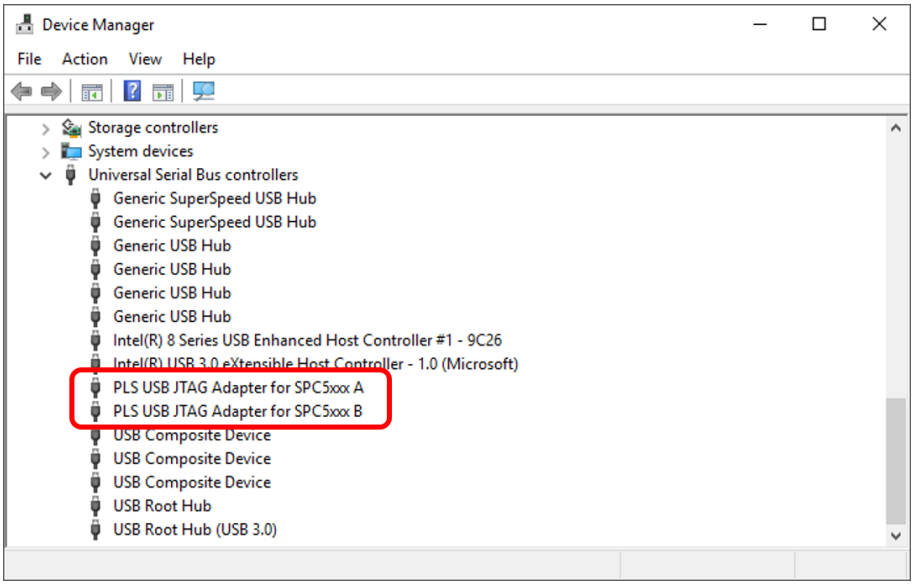

On Micro board (SPC53M64L7 micro on SPC563M-DISP board), a firmware delivered within the SW package is responsible for communication between ASIC interfaces (MSC, SPI, IO, PWM, …) and Graphical User Interface running on user PC.

## 2.2 First PT&S Connect usage

The first time PT&S Connect module is plugged into the PC, after the driver installation, the following procedure is needed to apply the software settings needed to ensure its functionality.

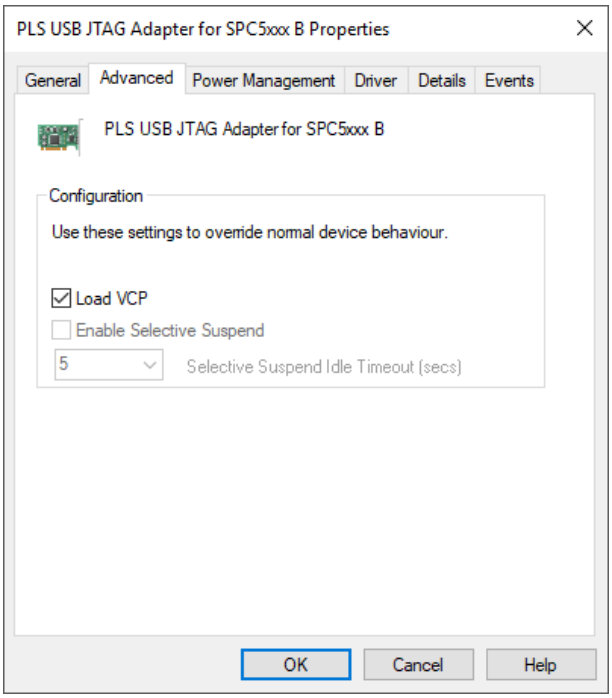
1. Connect PT&S Connect to any USB port;
2. Open Windows Device Manager. PT&S Connect module should be recognized by the PC and listed under Universal Serial Bus controllers, with the items *PLS USB JTAG Adapter for SPC5xxx A/B*, as shown in the following picture;

**Figure 17. Device manager PT&S Connect identification**




3. Right click on *PLS USB JTAG Adapter for SPC5xxx B* and select “Properties”;
4. Select “Advanced” tab, tick “Load VCP” box, as shown in the following picture and click Ok button;

**Figure 18. PT&S Connect driver settings**

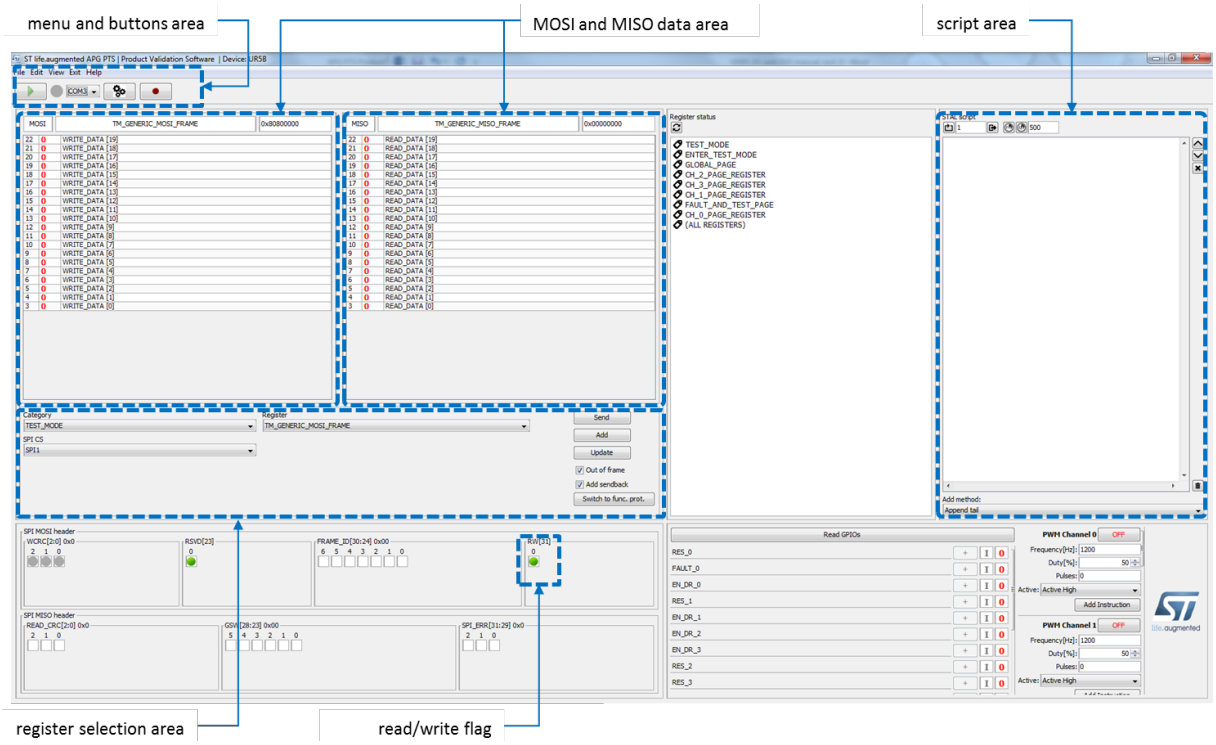


5. As a double-check, right click on *PLS USB JTAG Adapter for SPC5xxx A* and select “Properties”; select “Advanced” tab and make sure “Load VCP” box is not ticked;
6. Disconnect and reconnect PT&S Connect from USB port;
7. PT&S Connect is now ready to communicate with L9305 GUI.

## 2.3 GUI startup and usage

This section instructs on how to use the GUI to control L9305 in a single daughterboard configuration. The Figure 19 shows GUI appearance at startup. Highlighted are the main functional areas available.

**Figure 19. GUI main screen**




Follow next steps to initialize the system:

1. Connect flat cables from Micro board to L9305 motherboard;
2. Connect USB cable to Micro board and PC → 1st LED on MICRO board will turn on;
3. Start GUI on PC and select the right COM port number in menu and buttons area (see Figure 19);
4. Press the “Gear” button → LED in the same area should become green and a “Ready” label should appear;
5. Press Play button in menu and buttons area → 3rd LED on MICRO board will toggle, to verify communication is on-going;
6. Make sure SPI CS in register selection area is set on SPI1 (refer to Section 2.4 CS GUI mapping and multi daughterboard usage for additional informations on CS usage).

A quick sanity check of the system, is reading register GLOBAL_PAGE – CHIPID, and getting hex value “A” as a result (see next section for details on how to send a read instruction).

### 2.3.1 Read operation

1. Select register to be read in the register selection area (the data fields of the chosen register will appear in the MOSI/MISO area);
2. Switch off read/write flag;
3. Since L9305 interface protocol is out-of-frame type, check box “Out of frame”;
4. The various fields of the MOSI header (CRC, cnt, etc.) will be updated automatically;
5. Press “Send” button.

The content of selected register, will be updated in MISO window; the other fields of MISO header will appear in the lower part of the GUI.

To add the read command described to a sequence, uncheck box “Out of frame” and press “Add” button; the instruction will not be sent to L9305, but will appear in the script area, in red characters.

### 2.3.2 Write operation

1. Select register to be written in the register selection area (the data fields of the chosen register will appear in the MOSI/MISO area);
2. Switch on read/write flag;
3. On MOSI area, configure bit by bit the data to be written (as an alternative, select a signal in MOSI area, and an input field will appear in register selection area, to allow keyboard input of data);
4. Since L9305 interface protocol is out-of-frame type, check box “Out of frame”;
5. The various fields of the MOSI header (CRC, cnt, etc.) will be updated automatically;
6. Press “Send” button.

Upon writing, the content of selected register will be updated in MISO window; the other fields of MISO header will appear in the lower part of the GUI.

To add the write command described to a sequence, uncheck box “Out of frame” and press “Add” button; the instruction will not be sent to L9305, but will appear in the script area, in green characters.

### 2.3.3 Script usage

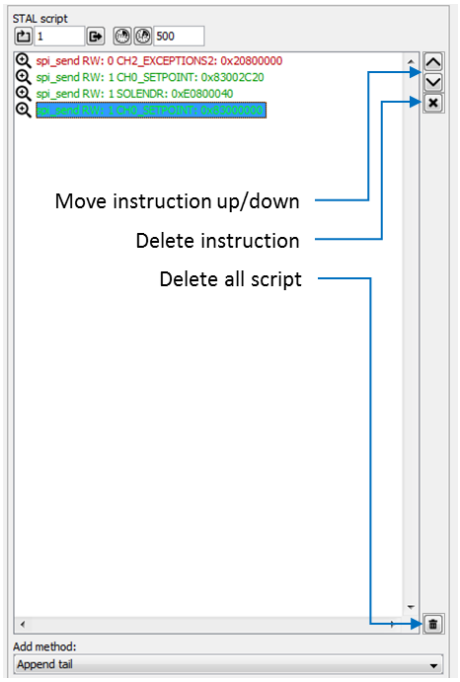
After creating a sequence of read and write instructions, visible in the script area, the user can edit each instruction, change its position in the sequence, save, recall and execute the sequence.

To edit an instruction:

1. select instruction to be modified (MOSI/MISO area will be updated accordingly);
2. apply desired variations on data or R/W flag;
3. make sure “Out of frame” is unchecked;
4. press “Update” button (instruction will not be sent to L9305).

A selected instruction can be moved or deleted using buttons indicated on Figure 20.

**Figure 20. Script editing buttons**




To save a sequence:

1. go into menu File → Save script;
2. choose destination folder, name and press Save;
3. “.stal” extension will be automatically appended to the name.

To load a sequence:

1. go into menu File → Read script;
2. choose folder and file and press Open.

If there’s no syntax error, the sequence will be loaded and appear in the script area.

To run a script, press “Play” button on top-left of the GUI.

## 2.4 CS GUI mapping and multi daughterboard usage

In case of a multi-daughterboard configuration, user must have the possibility to address each of the stacked daughterboards singularly. This is achieved by selecting, through the dedicated jumper, a different CS (Chip Select) for each of the daughterboards; this setting must be coherent with the one visible in the register selection area of the GUI, according with the following table, which shows the firmware mapping between GUI selected SPI and actually used CS line.

**Table 10. GUI SPI mapping**

| GUI SPI selection | CS line |
|---|---|
| SPI1 | 2 |
| SPI2 | 3 |
| SPI3 | 1 |
| SPI4 | 0 |

Applying the correct jumpers setting and taking into account the correspondance described in the table above, the user will be able to address each read/write operation to the desired daughterboard, by selecting its SPI address in the GUI, before sending the command. The same SPI addressing will be applied also if the command is added to a sequence, instead of send it immediately. In this way, it is possible for the user to build an instructions sequence to program each of the daughterboards independently.

It is always possible to set the same CS on all the daughterboards. In this case, every write instruction will have effect on all the daughterboards at the same time; however, read instructions results will be unreliable (there's no way to sort out which daughterboard is actually answering).

# Revision history

Table 11. **Document revision history**

| Date | Version | Changes |
|---|---|---|
| 17-Jun-2021 | 1 | Initial release. |

# Contents

# List of tables

# List of figures