

Low Voltage Sub-System Documentation

Diego Lopez

October 21, 2024

Abstract

This document provides details about the Low Voltage Sub-System (LVSS), including its pin configuration, startup sequence, and CAN messaging functionalities.

Contents

1	LVSS Pins	2
2	System Behavior	2
2.1	Vicor Fault Handling	2
3	CANopen	2
3.1	SDO Protocol	3
3.2	Messages	3
4	Finite State Machine (FSM)	3
4.1	FSM Diagram	3
4.2	initState	3
4.3	softStartState	3
4.4	powerUpState	3
5	SDO Class	4
5.1	CANopen Methods	4
5.2	User Defined Methods	4
5.3	To Do:	4

1 LVSS Pins

The LVSS pins are configured as follows:

- **EN_TMS_12V**
 - Bidirectional; PA0
 - GPIO to power switches to enable functionality. Enables TMS line.
- **EN_HB_12V**
 - Bidirectional; PA1
 - GPIO to power switches to enable functionality. Enables Handlebar line.
- **EN_HIB_12V**
 - Bidirectional; PC14
- **USART2_TX**
 - OUT; PA2
- **USART2_RX**
 - IN; PA3
- **VICOR_SENSE_IN / CURRENTSNS**
 - IN; PA4
 - Current going through LVSS that needs to be stepped down.
- **VICOR_FAULT**
 - N; PB4
- **CAN_RX**
 - IN; PA11
- **CAN_TX**
 - OUT; PA12
- **SWDIO**
 - IN; PA13
- **SWCLK**
 - IN; PA14
- **SWITCH_1_SNS_MCU**
 - IN; PC0
- **SWITCH_3_SNS_MCU**
 - IN; PC1
- **SWITCH_2_SNS_MCU**
 - IN; PC2

2 System Behavior

The LVSS system involves a startup sequence and fault handling as follows:

2.1 Vicor Fault Handling

- What to do if the Vicor has a fault?
- Reports the current through the LVSS (using **VICOR_SENSE_IN**).
- The system has a startup sequence for all the switches.

3 CANopen

The LVSS uses CANopen as the main communication protocol. It is a communication protocol stack that makes use of the OSI model.

3.1 SDO Protocol

The Service Data Objects service is a protocol that uses the client/server model. It allows a node to read/edit another nodes values.

3.2 Messages

The following messages are handled over CAN communication:

- Enable/disable boards (input).
- **VICOR_SENSE_IN** reporting (output).
- Power switch errors (output).
- Currents of all power switches (output).
- CAN Message Decryption.
- Determine data format.
- Number of bytes and how to split the message.

4 Finite State Machine (FSM)

The LVSS system uses a finite state machine for control:

4.1 FSM Diagram

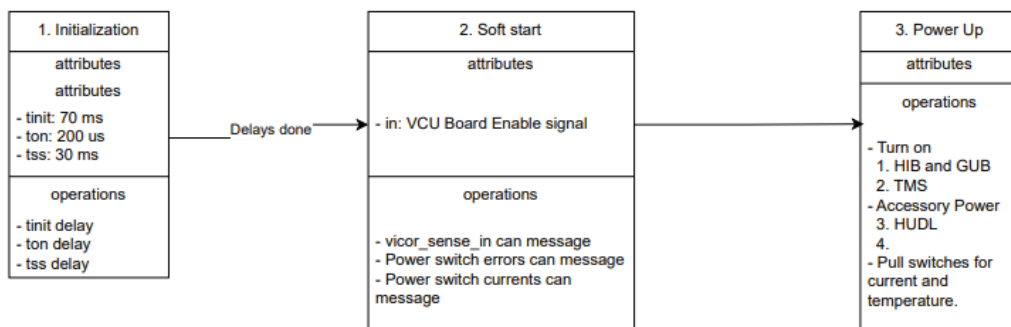


Figure 1: Finite State Machine Diagram

4.2 initState

- Starting state for the FSM.
- Waits a set amount of time based on the Vicor Datasheet FSM.

4.3 softStartState

- Receives CANopen messages from the VCU to determine which boards to turn on.
- If certain boards are not turned on within a set amount of time, the system sends an error message.

4.4 powerUpState

- Sends out signals to communicate which boards to turn on.

5 SDO Class

The SDO class will use CANopen methods to talk to both boards. This will allow the nodes to transmit and edit messages.

5.1 CANopen Methods

The methods used from CANopen.

- **COCSdoFind()**
- **COCSdoRequestDownload()**
 - Writes a value from client to server.
- **COCSdoRequestUpload()**
 - Reads a value from client to server.

5.2 User Defined Methods

The methods used from CANopen

- **SDOTransfer()**
- **SDOReceive()**

5.3 To Do:

- Fix client node
 - Values are not being transferred or received.
- Test Server/Client connection

References

- [1] Vicor Corporation. *Vicor Datasheet Documentation*.
- [2] CANopen Explained - A Simple Intro [2022]. *CSS Electronics*. <https://www.csselectronics.com/pages/canopen-tutorial-simple-intro>
- [3] Embedded Office. *SDO Client*. Canopen, canopen-stack.org/v4.4/api/csdo/. Accessed 26 Oct. 2024. <https://canopen-stack.org/v4.4/api/csdo/>