Low Voltage Sub-System Documentation

Electrical: Andrew Wolters, Ben Fauteux
Firmware: Diego Lopez

October 21, 2024

Abstract

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

Contents

1 Schematic Architecture

1.1 Altium Schematic

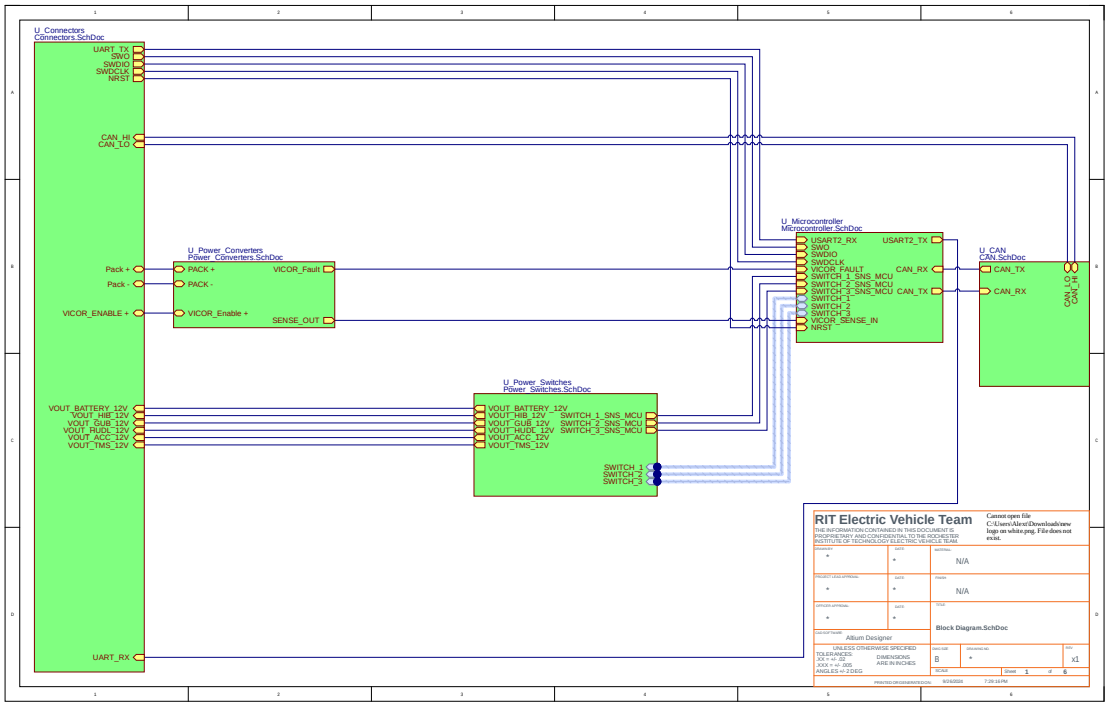

Figure 1: Altium Top Level Schematic

1.2 Pins

1.2.1 EN_TMS_12V

- Bidirectional; PA0
- GPIO to power switches to enable functionality. Enables TMS line.

1.2.2 EN_HB_12V

- Bidirectional; PA1
- GPIO to power switches to enable functionality. Enables Handlebar line.

1.2.3 EN_HIB_12V

- Bidirectional; PC14

1.2.4 USART2_TX

- OUT; PA2

1.2.5 USART2_RX

- IN; PA3

1.2.6 VICOR_SENSE_IN / CURRENTSNS

- IN; PA4
- Current going through LVSS that needs to be stepped down.

1.2.7 VICOR_FAULT

- N; PB4

1.2.8 CAN_RX

- IN; PA11

1.2.9 CAN_TX

- OUT; PA12

1.2.10 SWDIO

- IN; PA13

1.2.11 SWCLK

- IN; PA14

1.2.12 SWITCH_1_SNS_MCU

- IN; PC0

1.2.13 SWITCH_2_SNS_MCU

- IN; PC2

1.2.14 SWITCH_3_SNS_MCU

- IN; PC1

2 System Behavior

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

2.1 Main Processes

Main functions of the LVSS.
GitHub Repo

2.1.1 Finite State Machine

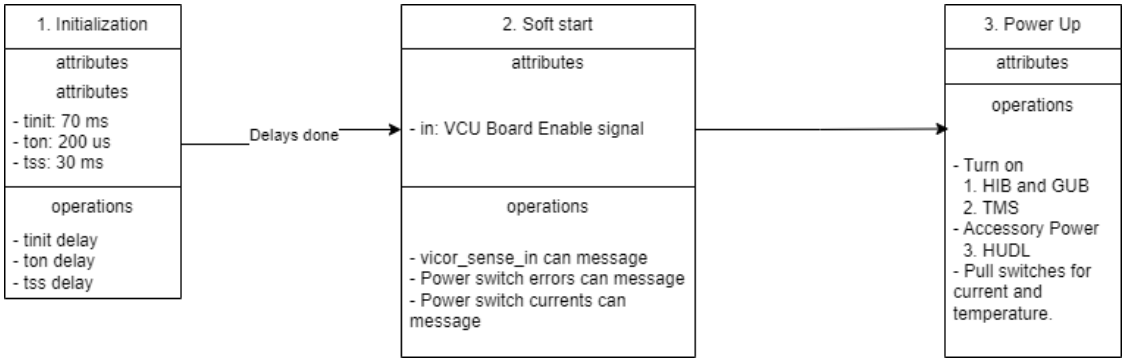

Figure 2: Finite State Machine Diagram

- initState()
  - Starting state for the FSM.
  - Waits a set amount of time based on the Vicor Datasheet FSM.
- 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.
- powerUpState()
  - Sends out signals to communicate which boards to turn on.
- VCU send a signal to LVSS with which boards should be turned on.

2.1.2 Switch Data

- Read current and temperature data.

2.2 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 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.

3.2 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.

3.2.1 CANopen Methods

Methods from CANopen submodule.

- **COCSdoFind()** : Searches for SDO node.
- **COCSdoRequestDownload()** : Client writes a value to server.
- **COCSdoRequestUpload()** : Client reads a value from server.

3.3 SDONode Class

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

3.3.1 User Defined Methods

The methods used from CANopen

- **SDOTransfer()** : Uses the COCSdoRequestDownload() method that requests that the server downloads the data from the client (This is backwards as fuck but ok).
- **SDOReceive()** : Uses the COCSdoRequestUpload() method that requests that the server send the data to the client.

3.4 Useful Link

- CANopen Overview
- SDO Methods Overview
- SDO Errors
- Octave SDO Reference
- CiA SDO Protocol

4 SDONode Class

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

4.1 User Defined Methods

Each method starts with **COCSdoFind()** which searches for SDO node.

4.1.1 SDOTransfer()

- Uses the COCSdoRequestDownload() method that requests that the server downloads the data from the client (This is backwards as fuck but ok).
- **COCSdoRequestDownload()** : Client writes a value to server.

4.1.2 SDOReceive()

- Uses the COCSdoRequestUpload() method that requests that the server send the data to the client.
- **COCSdoRequestUpload()** : Client reads a value from server.

References

[1] Vicor Corporation. *Vicor Datasheet Documentation.*

[2] CANopen Explained - A Simple Intro [2022]. *CSS Electronics.*

[3] Embedded Office. *SDO Client.*