EVM User's Guide: INA700-701EVM

INA700-701EVM Evaluation Module

Description

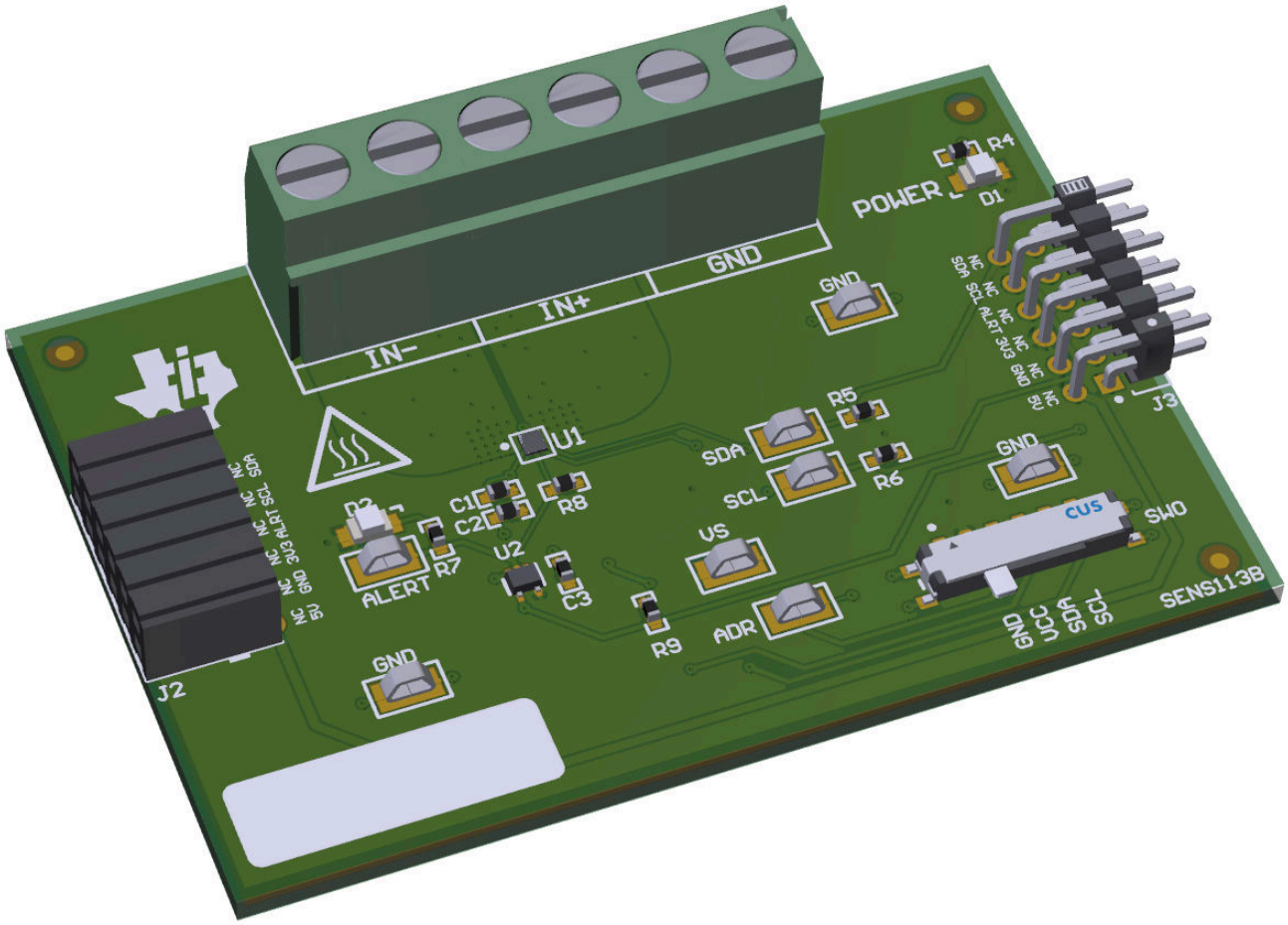
The INA700-701EVM is an easy-to-use platform that can evaluate the main features and performance of the INA700 or INA701 device. The evaluation module (EVM) comes with graphical user interface (GUI) support to read and write to device registers.

Features

- GUI support to read and write device registers as well as view and save results data
- EVM detached from SCB for custom use cases
- Multiple EVM support with single SCB or GUI
- Conveniently powered from a common micro-USB connector through the SCB

Get Started

1. Buy the INA700 or INA701 Evaluation Module.
2. Buy the TI-SCB sensor control board (SCB).
3. Download and install the PAMB Windows USB Drivers.
4. Read this user's guide to set up the hardware.
5. Run the EZShunt-EVM-GUI.



1 Evaluation Module Overview

1.1 Introduction

The EVM is an easy-to-use platform for evaluating the main features and performance of the INA700 or INA701. The EVM includes a graphical user interface (GUI) used to read and write device registers as well as view and save results data. See the individual device data sheets for current limits for the desired use case.

This EVM user's guide describes the characteristics, operation, and use of the INA700 and INA701 evaluation modules (EVMs). These EVMs are designed to evaluate the performance of the INA700 or INA701. Throughout this document, the terms evaluation board, evaluation module, and EVM are synonymous with the INA700-701EVM. This document includes a schematic, reference printed circuit board (PCB) layouts, and a complete bill of materials (BOM).

1.2 Kit Contents

Table 1-1 lists the contents of the EVM kit. Contact Texas Instruments' customer support center if any component is missing.

Table 1-1. Kit Contents

ITEM	QUANTITY
INA700EVM OR INA701EVM	1

Note

This EVM requires the TI Sensor Control Board (SCB), which is sold separately and can be found here.

1.3 Specification

The EVM is intended to provide basic functional evaluation of the device. The layout is not intended to be a model for the target circuit, nor laid out for electromagnetic compatibility (EMC) testing. The EVM consists of one printed-circuit board (PCB). The PCB is referred to as the EVM, and has either the INA700 or INA701 installed.

1.4 Device Information

The INA700 and INA701 are digital power monitors with an integrated shunt resistor along with a 16-bit delta-sigma ADC specifically designed for current sensing applications. The device can measure full-scale currents up to ±15.728 A (INA700) or ±6.225 A (INA701), over the common-mode voltage range of –0.3 V to +40 V.

The INA700 and INA701 reports current, bus voltage, die temperature, power, energy and charge accumulation with a precision ±0.5% integrated oscillator, all while performing the needed calculations in the background. The integrated temperature sensor is ±3°C accurate over the junction temperature range.

The low offset and gain error drift of the INA700 and INA701 allows use in systems that do not undergo temperature calibration during manufacturing.

The device features selectable ADC conversion times from 50µs to 4.12ms as well as sample averaging from 1x to 1024x, which further helps reduce the noise of the measured data and allows for optimization of overcurrent detection windows.

The device is available in a tiny PowerWCSP (DSBGA) package to minimize the solution size and maximize thermal performance.

Table 1-2. Device Summary

PRODUCT	DIGITAL PROTOCOL	ADC RESOLUTION	MAX GAIN ERROR	MAX OFFSET CURRENT
INA700	I^2C	16-bit	±0.5% (at 5A)	±1.5 mA
INA701	I^2C	16-bit	±0.5% (at 0.5A)	±0.5 mA

2 Hardware

2.1 Quick Start Setup

The following instructions describe how to set up and use the EVM.

1. Purchase an SCB if you do not already have one.
   a. To use a PAMB Controller instead, see *PAMB Compatibility*.
2. Download this driver and install **as an administrator**: https://www.ti.com/lit/zip/sbac253.
   a. Follow the download prompts; a myTI account is required.
   b. Note that this driver is labeled as a PAMB driver, but is also used for the SCB.
3. Attach the EVM to the SCB Controller as shown in Figure 2-1.
   a. Refer to Figure 2-2 when connecting multiple EVMs of the same type together.
4. Connect the EVM to the PC using the provided USB cable.
   a. Insert the micro-USB cable into the SCB Controller onboard USB receptacle J2.
   b. Plug the other end of the USB cable into a PC.
5. Access the GUI from this link in either Chrome®, Firefox®, or Safari®: https://dev.ti.com/gallery/info/CurrentSensing/EZShunt-EVM-GUI.
6. Connect the GND reference of the external system to the GND node of the EVM (pins 1 or 2 of J1).
7. Verify that the load current is disabled, then connect the IN+ and IN– nodes to the load line by connecting the signal leads to J1 pin 5 or 6 and J1 pin 3 or 4 on the EVM as explained in *Current Sensing Operation*.

2.2 EVM Operation

To use the EVM with the SCB Controller (sold separately), connect the EVM as shown in Figure 2-1.

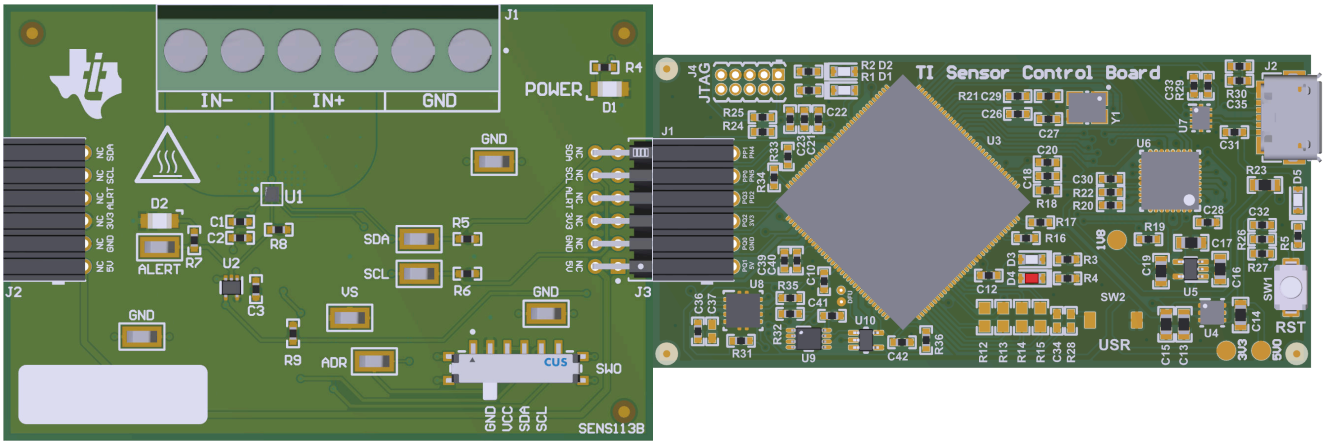


Figure 2-1. EVM (Left) Connected to SCB Controller (Right)

If using multiple EVMs, connect them as shown in Figure 2-2. Make sure to use a different address for each device. The GUI only supports one EVM and device type at a time; up to four EVMs total.

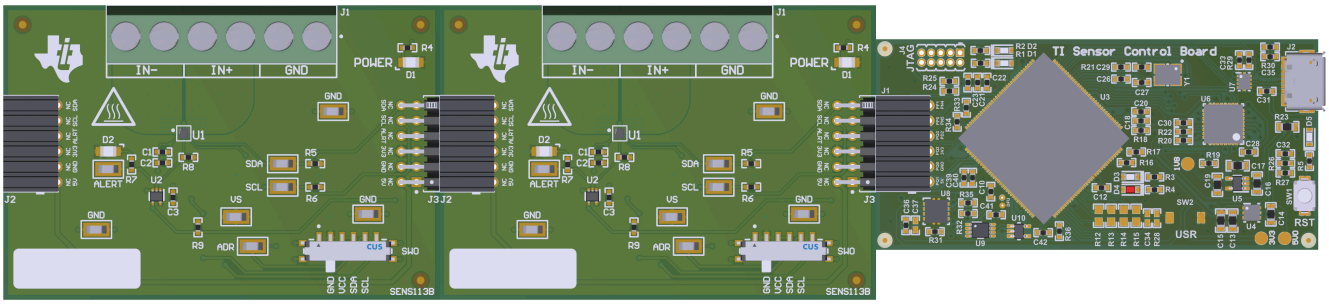


Figure 2-2. Multiple EVMs Connected to SCB Controller

2.2.1 Current Sensing Operation

The INA700 and INA701 each come equipped with an internal shunt resistor and is out-of-the-box ready to measure current (see the individual device data sheets for current limits for the desired use case). There are two terminals each for IN- (J1 pins 5 and 6) and IN+ (J1 pins 3 and 4) for convenience.

2.2.1.1 Detailed Setup

To configure a measurement evaluation, follow these steps:

1. Connect a current-carrying line by connecting a line input to the IN+ terminal and the resulting output line to the IN– terminal in series with the load while powered off.

   WARNING

   When measuring current, first verify that the equipment (shunt resistor, wires, connectors, and so on) can support the amperage and power dissipation. Secondly, verify that the current flowing through J1 does not exceed the current limit for the device being used based on the specific use case. Failure to do so can result in damage to the EVM, or personal injury.

   The EVM can get hot.

2. Connect the system ground to either of the GND terminals on J1 (pins 1 or 2).
3. Power on the system, then observe the device states and outputs through the GUI.

2.3 Circuitry

This section summarizes the EVM subsystems and components.

2.3.1 Current Sensing IC

This section describes the main device and supporting components.

U1 is the main current-sensing device, either the INA700 or INA701. C1 and C2 are bypass capacitors that are placed near the sensor to help mitigate power supply noise and provide current quickly to the device when needed. LED D1 with current limiting resistor R4 are used to indicate when the EVM is powered on.

2.3.2 Input Signal Path

This section describes the circuitry of the input signal path.

J1 is the main connection terminal. Pins 1 and 2 of J1 are used to tie the system ground to the EVM ground. Pins 3 and 4 are tied to IN+, and Pins 5 and 6 are tied to IN–. There are two pins each for IN– and IN+ for convenience, as well as current capabilities. For currents greater than 10A, verify that all pins of the terminal blocks are populated.

2.3.3 Digital Circuitry

This section describes the digital circuitry around the device.

J2 and J3 are the main header pins that connect the digital and power pins to the SCB Controller or other EVMs. J3 connects to the EVM or SCB on the right, while J2 connects to more EVMs on the left. R5 and R6 are used as pullup resistors for the SDA and SCL pins.

SW0 sets the I^2C address of the device. This can be useful when using the EVM with a custom controller (other than the SCB Controller), or when connecting multiple EVMs together. Currently the SCB Controller and GUI are set up to use up to four EVMs at a time.

R8 is used as a pullup resistor for the ALERT pin. LED D2 and current limiting resistor R7 are used to indicate when the ALERT has triggered. U2 is an open-drain buffer that forwards the alert signal to J2 and J3 without allowing the signal to propagate from the ALERT bus to the device. This feature is primarily used when working with multiple EVMs, so that the individual ALERT LEDs can be seen on each EVM while still using the ALERT bus. R9 is a pullup resistor for the ALERT bus. C3 is a bypass capacitor placed near the buffer to mitigate power supply noise and to help provide current quickly to the device when needed.

When connecting multiple EVMs together, the pullup resistors for the I2C Bus and the ALERT Bus of some EVMs can need to be removed to avoid pulling up too strongly.

2.4 PAMB Compatibility

If desired, this EVM and GUI can be used with the PAMB Controller (DC081A) by jumper wiring the pin headers of the PAMB to the EVM. Figure 2-3 shows which pins on the PAMB correspond to the EVM header pins. Note not to add too much resistance in the jumper wire connection setup or the signal can degrade and cause communication errors.

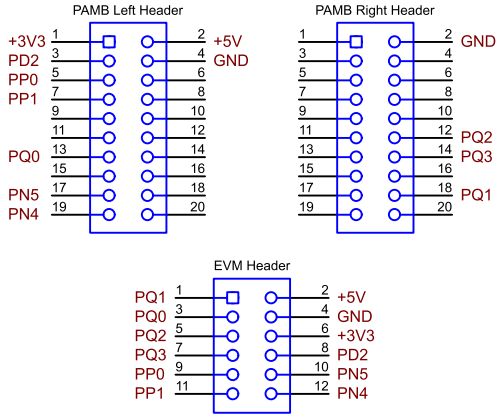


Figure 2-3. EVM to PAMB Connection

3 Software

3.1 Setup

3.1.1 Driver Installation

Download and install this driver: https://www.ti.com/lit/zip/sbac253. This is a one-time step per computer and requires a myTI account. Note that this driver is labeled as a PAMB driver and is also used for the SCB. Unzip the folder and **run the .exe file with administrator privileges**.

3.1.2 Firmware

Firmware updates are pushed through the GUI (requires previous driver to be installed). Downloaded offline GUIs only update the SCB Controller with the latest firmware available at the time of download. To check for the latest GUI or Firmware updates, launch the latest GUI version from the web browser.

3.1.2.1 Firmware Debug

If the firmware must be manually reinstalled for any reason, then follow these steps to reinstall the firmware. Verify that the EVM is connected to the SCB before being powered on.

1. First, see if the GUI can program the firmware manually.
   a. Plug the SCB controller into the PC.
      i. Verify that the EVM is connected to the SCB first.
   b. Launch the GUI and close the READEME.md window.
   c. If the MCU is already in Device Firmware Upgrade (DFU) mode, then a dialog box appears saying "Connection Failure - No SCB controller detected". Close this window.
      i. For a new SCB, the LEDs cycles 4 times and then enters DFU mode automatically.
   d. If the GUI does not update automatically, then go to *File* > *Program Device...*.
      i. If the *Program Device...* button is still grayed out, then click the connect button in the lower left corner and then try again.

2. If step 1 is unsuccessful (or if the *Program Device...* button is still grayed out), then manually configure the MCU on the SCB Controller to be in DFU Mode. This can be done through either of the below methods with the SCB Controller powered on:
   a. Through software:
      - Send the command 'bsl' on the USB Serial (COM) port of the SCB.
   b. Though hardware:
      - For safety, **turn off and disconnect all load sources and external voltages**.
      - While shorting the two test points labeled *DFU* (shown in Figure 3-1) with a pair of tweezers (or wire), press and release the RESET button.

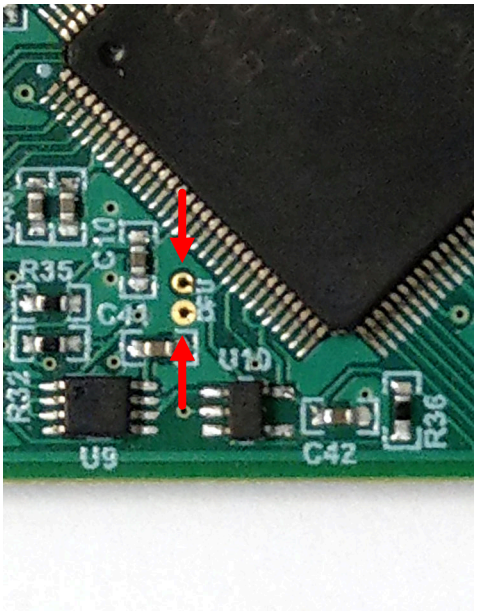


Figure 3-1. Test Points Used to Enter DFU Mode Manually

- If the PAMB board is being used instead, then these test points are located near PK1 and PK2.

With the MCU in DFU mode, upload the firmware through the method outlined in step 1. If the device is in DFU mode, but the firmware update continues to fail, then manually associate the driver ("boot_usb.inf") with the DFU device.

3.1.3 GUI Setup and Connection

The GUI can be accessed from this link in either Chrome®, Firefox®, or Safari®: https://dev.ti.com/gallery/info/CurrentSensing/EZShunt-EVM-GUI.

3.1.3.1 Initial Setup

To set up the GUI the first time:

1. Make sure that the previously mentioned driver is installed successfully so that everything works properly and that the GUI can update the EVM firmware, if necessary.
2. Check to make sure the EVM and SCB Controller unit is plugged into the PC, then go to the previously-provided GUI link.
   a. Verify that the EVM is connected to the SCB before plugging the SCB into a USB port.
3. Click the *GUI Composer* application to launch the GUI from the web browser (see Figure 3-2) .

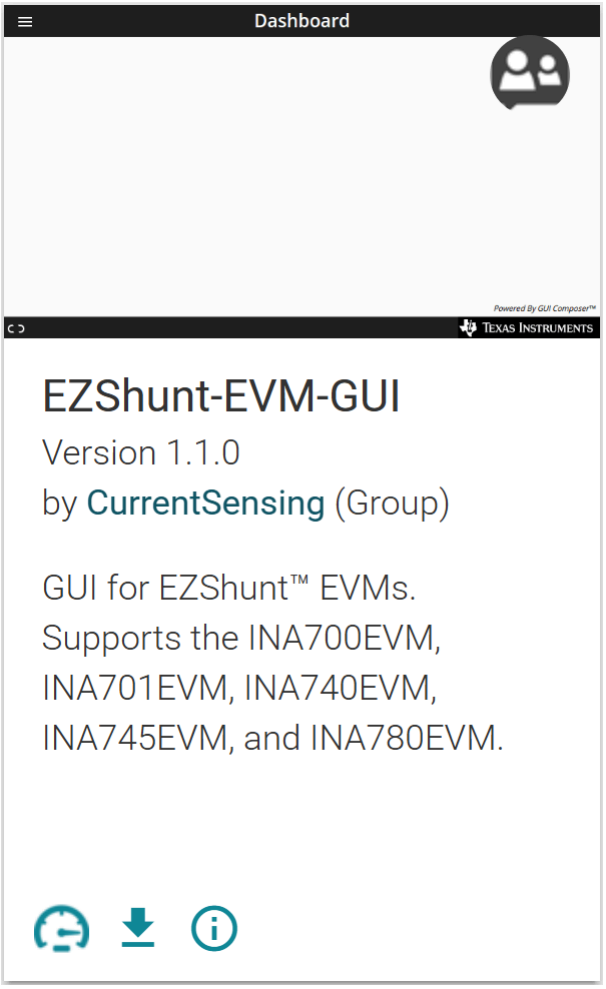


Figure 3-2. GUI Composer Application

a. The GUI link brings up all versions of the GUI. TI recommends launching the newest version available.
b. For first-time GUI Composer setup, follow the prompts to download the *TI Cloud Agent* and browser extension (see Figure 3-3). Close the README.md dialog box to see these prompts.

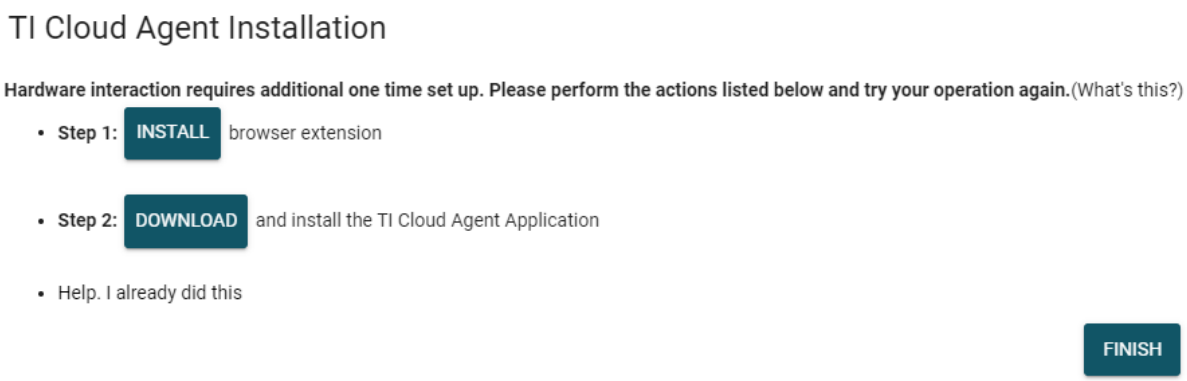


Figure 3-3. TI Cloud Agent

4. Optionally, to download the GUI for offline use, click the icon in the *GUI Composer* application and follow the prompts (see Figure 3-2).

3.1.3.2 GUI to EVM Connection

To connect the GUI to the EVM, follow these steps:

1. Setup and launch the GUI as described in *Initial Setup*.
   a. Connect the EVM to the SCB before powering on.
2. Close the README.md file page to initiate a connection between the EVM and the GUI. If successful, then the *Hardware Connected* message is visible near the bottom-left corner of the GUI.

Figure 3-4. Hardware Connected

   a. A green indicator with the device type and the text *DEVICE CONNECTED* is also visible near the top-left corner of the GUI.

Figure 3-5. Device Connected

   b. If *Hardware Connected* and *DEVICE CONNECTED* do not show in the GUI, then long-press the RESET button on the EVM to try again.
      i. If the long-press reset does not work, then check different hardware COM ports under *Options* > *Serial Port*.

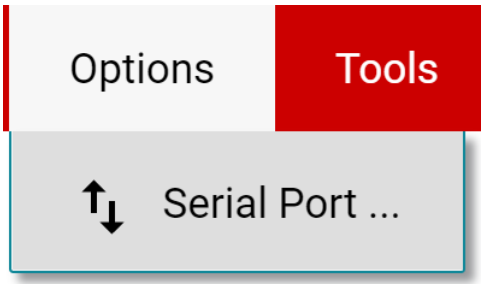


Figure 3-6. Change Serial Port

   c. If the hardware still does not connect, then make sure the correct GUI and EVM combination is being used.
      i. If using the correct GUI and EVM combination, then reprogram the firmware of the SCB as described in *Firmware Debug*.
      ii. Many connectivity issues can be addressed by doing one of the following:
         1. Long-press the RESET button on the EVM with the EVM and SCB connected to each other.
            - Refreshing the GUI can also help.
         2. Connect the EVM to a different USB port.
            - Avoid using long cables and USB hubs.
            - If using a desktop PC, try a USB port on the back.

3.2 GUI Operation

Setup, launch, and connect the GUI to the EVM per *GUI Setup and Connection*. Refer to the sections below for a description on how to use each tab of the GUI.

3.2.1 Homepage Tab

The GUI starts out on the *Homepage* tab. Click the (Homepage) icon on the menu to the left to return to the *Homepage* tab at any time.

From the *Homepage*, you can easily confirm successful GUI to EVM connection (see *GUI to EVM Connection*). There are also helpful resources available through the buttons on the bottom (see Homepage Tab Links).

Figure 3-7. Homepage Tab Links

3.2.2 Registers Tab

To view and edit the device registers, click the (Registers) icon on the menu to the left. The *Registers* tab looks similar to the one shown in Figure 3-8.

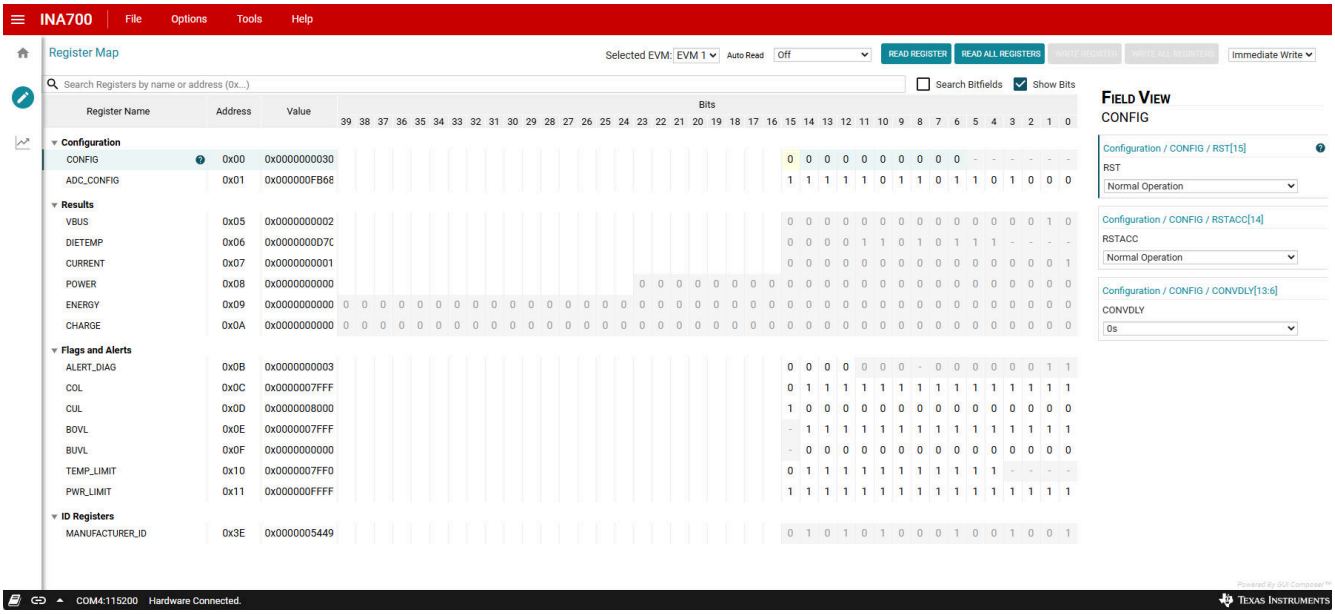


Figure 3-8. GUI Registers Tab

From this tab, the user can read and write device registers on the EVM. Here are some important notes:

- Use the *Selected EVM* drop-down menu at the top to choose which device to work with on the *Registers* tab.
  - Functionally, this button sets the default read and write address in the MCU and then reads all register values back to update the register map. Note that if data is currently collecting at a high frequency, then this can cause a minor delay in the data collection. To prevent this, set the device settings before you starting data collection.
- By default, all changes are automatically written to the device. If desired, then the user can change the *Immediate Write* setting to *Deferred Write* to only allow writing when ready.
  - Users can modify writable register values from any of these methods:
    - Through the widget settings in the *Field View* section on the right.
    - Changing the *Value* field directly with either hex or decimal values.
    - Double-clicking any individual bit.

- Turning on *Auto Read* only updates registers in the *Registers* tab and not the plots in the *Results Data* section.
  - Leaving *Auto Read* on while collecting data for plots can interfere with data collection timing.
- For questions about a register or register bit field, click the icon.
  - For even more questions about registers, check the data sheet.
- For convenience, register settings can be saved and loaded back later to any device with the sam register map. To save or reload register settings, go to *File* > *Register Data*.

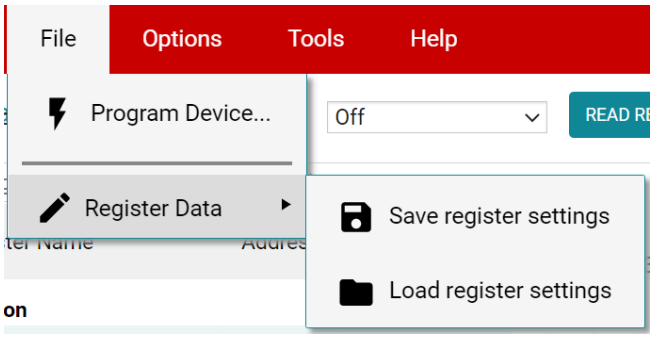


Figure 3-9. Save and Load Register Settings

  - Press the *Read All Registers* button after loading data to update the register map with the actual device values, in case the loaded registers are not compatible with the connected device.

3.2.3 Results Data Tab

To view and collect results data over time, click the (Results Data) icon on the menu to the left. Figure 3-10 shows part of the *Results Data* tab for reference, which can look different depending on the number of connected EVMs.

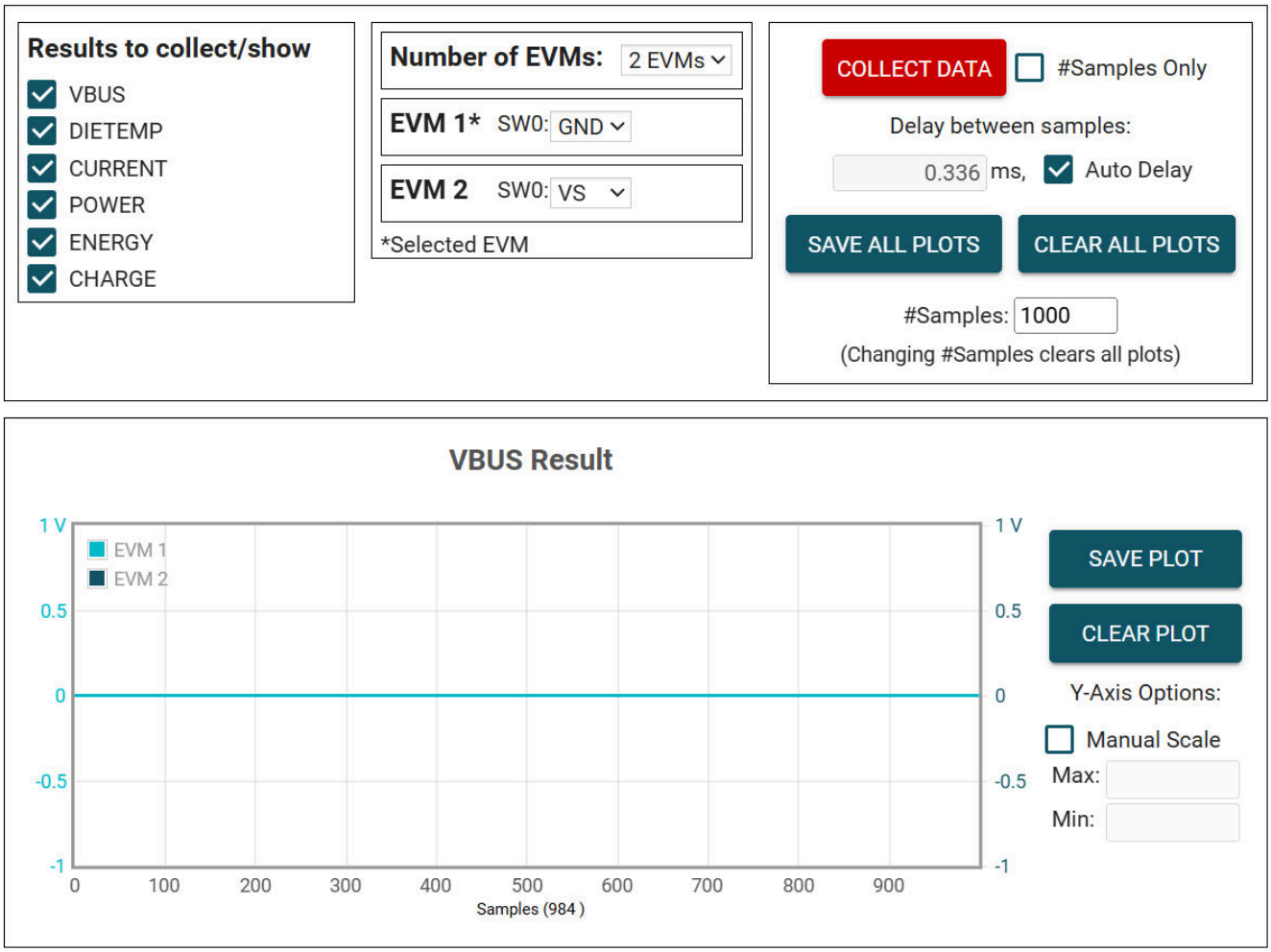


Figure 3-10. Results Data Tab and Settings

A description of how to use the buttons and settings at the top of the *Results Data* tab and next to each plot is below:

- Results to collect/show
  - Use this section to select which register values to collect data for. If a results register is unselected before the *COLLECT DATA* button is pressed, then the plot below is hidden and the EVM does not try to read this register during the collect cycle (even if the conversion is enabled).
  - If the user disables one of these settings while the EVM is collecting data, then the plot does not show, but data is still collected and the plot updates in the background. Reselect to show data.
- Number of EVMs
  - Set the *Number of EVMs* drop-down menu to the number of EVMs currently in use.
    - See Figure 2-2 for how to attach multiple EVMs together.
  - The GUI only supports one EVM and device type at a time, up to four EVMs total.
  - Switch settings
    - Use the onboard switch to select a different address for each EVM.
    - Set the switch settings in the GUI to match the setting for each connected EVM.
      - *EVM 1* automatically populates with the lowest addressed device if the SCB is powered up with the EVM connected, unless a setting has already been selected.
    - If more than one device is being used, an asterisk symbol (*) appears next to the selected EVM that is being used on the *Registers* tab.
      - Changing the switch settings of any EVM sets that EVM as the selected EVM.
- Collect and Plots settings

- Collect Data
  - Press the *COLLECT DATA* button to start data collection.
  - In this mode, the MCU reads and sends the selected result values for each device over a USB BULK channel. All results from one device are read before moving on to the next device.
    - All result values from all EVMs together are considered one "sample set".
  - Although users can read and write to other registers through the *Registers* tab while collecting data, this can add a delay to the data being collected.
  - Press the *STOP COLLECT* button to stop collecting data.
- #Samples Only
  - If this checkbox is selected, then the GUI automatically stops collecting data after the number of samples specified in the #Samples box have been collected.
  - If not selected, then the GUI continues collecting data and only stores the most recent #Samples.
- Delay between samples
  - Sets the delay between the start of each sample set.
  - Desired delay time is not obtainable set faster than the read loop, which depends on the number of results being collected, the number of EVMs, and the CPU.
  - Although users can read and write to other registers through the *Registers* tab while collecting data, this can add a delay to the data being collected.
- Auto Delay
  - Sets delay based on conversion times, averaging, and number of channels being converted.
  - If multiple EVMs are being used, then the time put in the delay box is from the EVM with the shortest calculated delay value.
- Save All Plots
  - Press the *SAVE ALL PLOTS* button to save the data for each currently selected result from the *Results to collect/show* section in a spreadsheet.
  - Press the *SAVE PLOT* button next to each plot to save just the data from that plot in a spreadsheet.
- Clear All Plots
  - Press the *CLEAR ALL PLOTS* button to clear the data from all plots together.
  - Press the *CLEAR PLOT* button next to each plot to clear the data from just that plot.
- #Samples
  - Change the number in this box to change the number of samples shown in each plot.
    - Changing this number clears out the plot buffers, so the plots are cleared on the next read.
- Y-Axis Options
  - Manual Scale
    - Check this box to set all EVM results in this plot to the same scale value specified by the Max and Min fields.
    - When this is not selected, each EVM has a separate Y-axis scale based on the minimum and maximum values of those EVM result values.
    - For the DIETEMP Result, the units can be toggled between °C and °F.
      - This processes the existing data in the plots and converts new data coming in. If receiving new data while changing units, a data point can be missed or duplicated.
  - Max
    - The maximum Y-axis value to use for all EVMs in this plot.
    - If this field is empty when Manual Scale is selected, then the field auto-populates with the maximum value currently in the plot.
  - Min
    - The minimum Y-axis value to use for all EVMs in this plot.
    - If this field is empty when Manual Scale is selected, then the field auto-populates with the minimum value currently in the plot.

3.3 Direct EVM USB Communication

If desired, the EVM can be communicated with directly without the use of the GUI through the USB port. This is done by sending the desired command string over the serial COM port and receiving the results either through

the COM port or the USB BULK channel based on the mode. This is useful for interfacing the EVM with custom setups, scripts, or GUIs.

3.3.1 Standard USB Read and Write Operations

Use the serial COM port to read and write registers through USB commands using the following format:

- Set device address format: setdevice DEVID
  - Where setdevice is always lower case, and DEVID is defined as:
    - The 4 least significant bits of the address, in decimal format. For example, an address of 0x4A is 10.
  - Note, when the SCB is reset while one or more EVMs are connected, the address defaults to the lowest address found.
    - The SCB checks for I^2C or SPI at start-up. If no device is attached, then the SCB defaults to SPI. Reset the SCB with an I^2C EVM connected to use I^2C.
  - For example, to set an I^2C EVM with a register address of 0x4A, send the command: setdevice 10
  - For this example, the EVM returns the acknowledgment and state (idle or collecting) in JSON format:
    {"acknowledge":"setdevice 10"}
    {"evm_state":"idle"}
- Read register format: rreg ADR
  - Where ADR is the address in hex, and rreg is always lower case.
  - Register addresses can be in upper or lower case, and do not need to be led by '0x'. 0 padding register addresses is also optional. For example, to read register address 0xB, some valid commands include:
    - rreg b
    - rreg 0B
    - rreg 0x0B
      - When '0x' is used, the 'x' must be lower case.
  - For this example, the EVM returns the results and state (idle or collecting) in JSON format:
    {"acknowledge":"rreg 0x0B"}
    {"register":{"address":11,"value":3}}
    {"evm_state":"idle"}
- Write register format: wreg ADR VAL
  - Where ADR and VAL are in hex, and wreg is always lower case.
  - Register addresses and values can be in upper or lower case, and do not need to be led by '0x'. 0 padding register addresses and values is also optional. For example, to write register address 0x1 with the value 0xfb68, some valid commands include:
    - wreg 1 fb68
    - wreg 01 0xfb68
    - wreg 0x01 0xFB68
      - When '0x' is used, the 'x' must be lower case.
  - For this example, the EVM returns the results and state (idle or collecting) in JSON format:
    {"acknowledge":"wreg 0x01 0xfb68"}
    {"console":"Writing 0xfb68 to ADC_CONFIG register"}
    {"evm_state":"idle"}

3.3.2 Collect Data Through the USB BULK Channel

The Collect Data function reads the desired result registers and sends the data based on the specified settings. This function works best with continuous conversion mode and does not configure the EVM or associated register settings. Collect mode starts and stops via the serial COM port, however the results are sent over the USB BULK channel. To use this mode, use the following format:

- Start collecting data format: collect timerPeriod collectFlags channelAddressIDs numDevices
  - Where collect is always lower case, and each parameter is the decimal representation of the value in the following format:
    - timerPeriod
      - The timer delay used in the MCU to allow data collection sample sets (in µs, unsigned 32-bit value).
    - collectFlags

        - a byte of data that has a 1 to collect and a 0 to not collect each register value type, according to the following definitions:
            - VBUS = 0b0100000
            - DIETEMP = 0b0010000
            - CURRENT = 0b0001000
            - POWER = 0b0000100
            - ENERGY = 0b0000010
            - CHARGE = 0b0000001
    - channelAddressIDs
        - This is the 4 least significant bits of each I^2C address chained together, starting with bits 3-0.
            - For example, if EVM 1 is on channel 0x44 and EVM 2 is on ox46, the value here is 0b01100100.
    - NumDevices
        - The number of EVMs chained together (1-4).
  - For example, to start data collection for VBUS, and DIETEMP every 3.156ms, for two INA700s with EVM 1 address = 0x44 and EVM 2 address = 0x46, you send: collect 3156 48 100 2.
    - For this example, the EVM returns the acknowledgment and state in JSON format:
      {"acknowledge":"collect 3156 48 100 2"}
      {"evm_state":"collecting"}
  - The USB BULK channel receives data in the format: frameID deviceNumID address registerSize data.
    - Where each parameter is the decimal representation of the value in the following format:
      - frameID (1 byte)
        - Always reads 0. Used to verify that data is aligned.
      - deviceNumID (1 byte)
        - An ID number corresponding to the EVM number.
          - From the above example, this is 1 if reading from EVM 1 which had address set to 0x44, and 2 if reading from EVM 2 which had an address set to 0x46.
      - address (1 byte)
        - The register address that is read from the device.
      - registerSize (1 byte)
        - The number of bytes that the following data has.
      - data (1 byte at a time)
        - The register data value, given in bytes with the most significant byte (MSB) first.
- Stop collecting data format: stop
  - Where stop is always lower case.
  - The EVM returns the acknowledgment and state in JSON format:
    {"acknowledge":"stop"}
    {"evm_state":"idle"}

4 Hardware Design Files

4.1 Schematics

Figure 4-1 and Figure 4-2 show the schematic of the EVM. Figure 4-1 shows the circuitry for the EVM. Figure 4-2 shows the mechanical components included with the EVM.

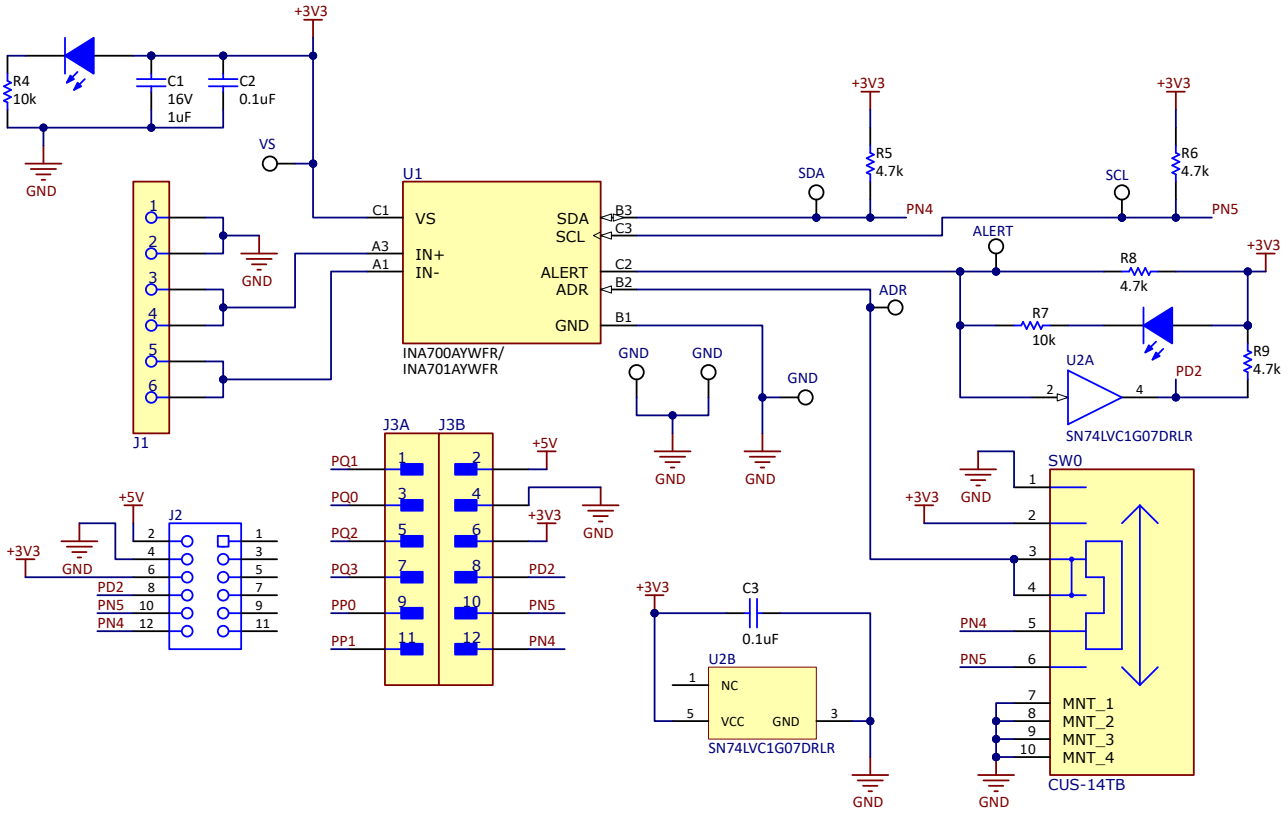


Figure 4-1. SENS113 Schematic

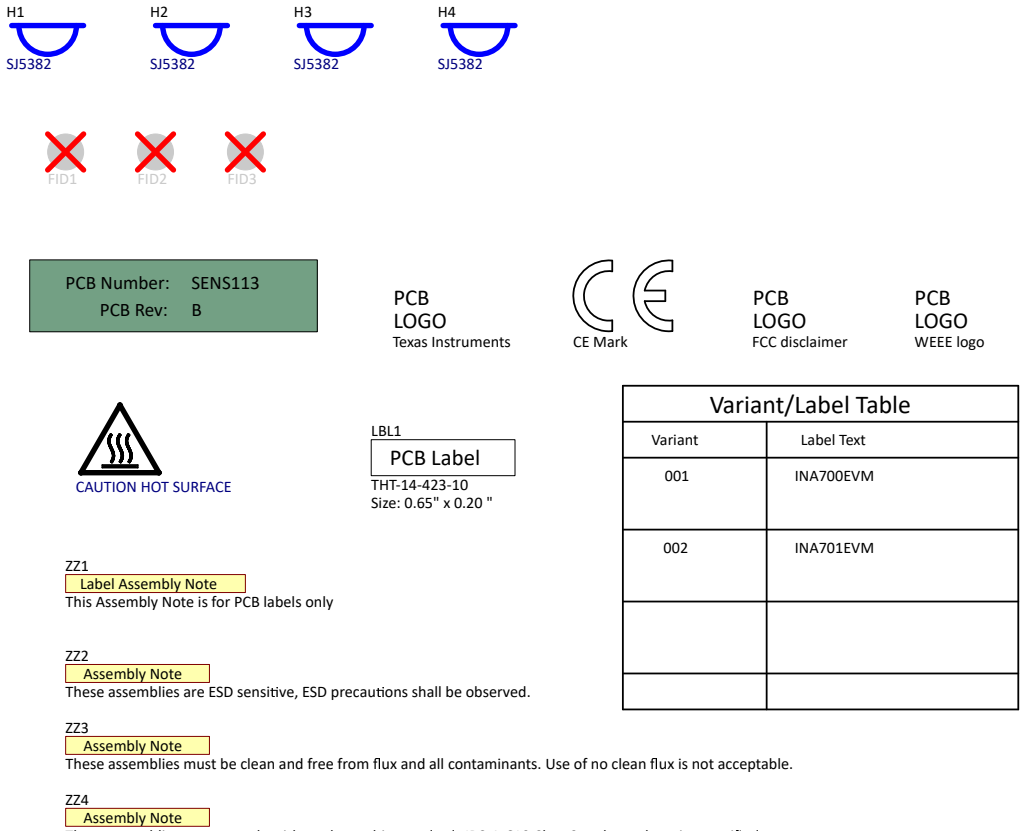


Figure 4-2. SENS113 Hardware Schematic

4.2 PCB Layout

Note

Board layouts are not to scale. These figures are intended to show how the board is laid out. The figures are not intended to be used for manufacturing EVM PCBs.

Figure 4-3 through Figure 4-6 illustrate the PCB layers of the EVM.

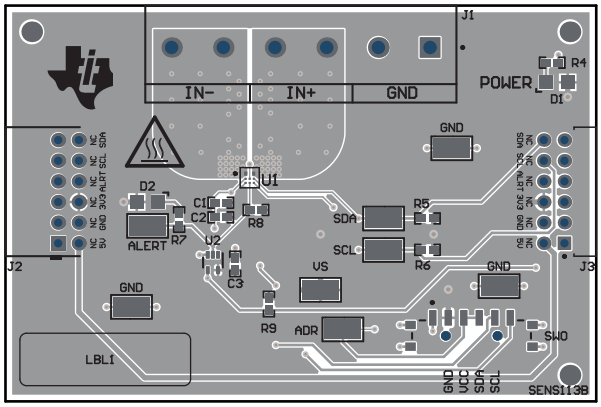


Figure 4-3. SENS113 Top View

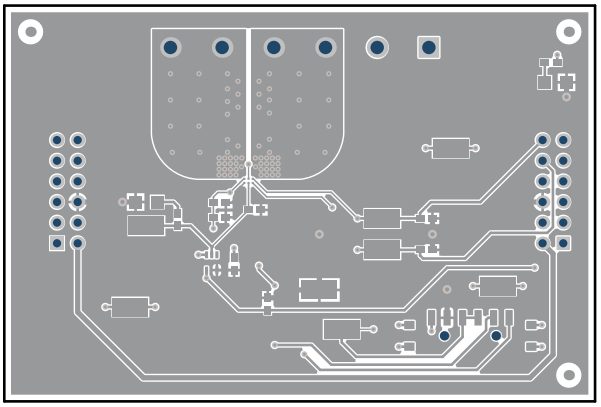


Figure 4-4. SENS113 Top Layer

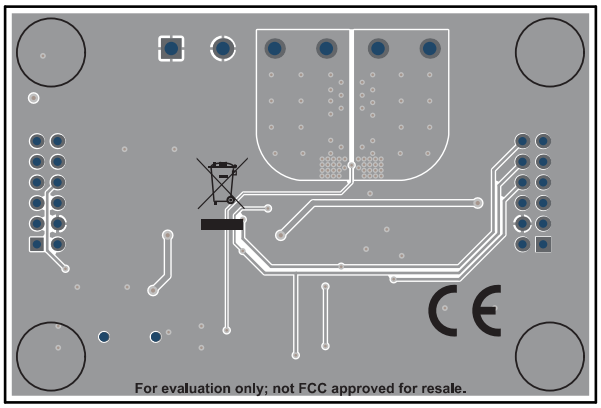


Figure 4-5. SENS113 Bottom View

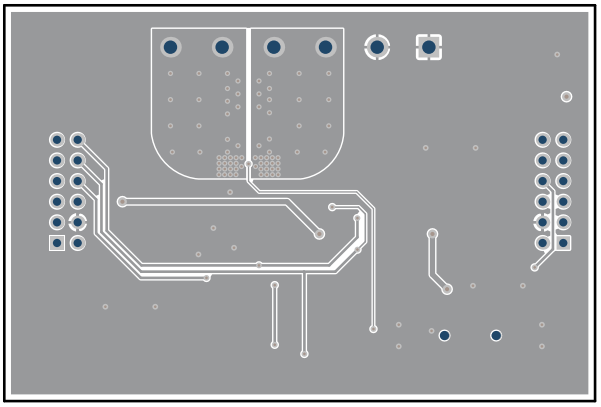


Figure 4-6. SENS113 Bottom Layer

4.3 Bill of Materials

Table 4-1 provides the parts list for the EVM.

Table 4-1. Bill of Materials

DESIGNATOR	QTY	VALUE	DESCRIPTION	PACKAGE REFERENCE	PART NUMBER	MANUFACTURER
C1	1	1uF	CAP, CERM, 1 uF, 16 V, +/- 10%, X6S, 0402	0402	C1005X6S1C105K050BC	TDK
C2, C3	2	0.1uF	CAP, CERM, 0.1 uF, 50 V, +/- 20%, X7R, 0402	0402	GRM155R71H104ME14D	MuRata
D1, D2	2	White	LED, White, SMD	0805	VAOL-S8WR4	Visual Communications Company, LLC
H1, H2, H3, H4	4		Bumpon, Hemisphere, 0.25 X 0.075, Clear	75x250 mil	SJ5382	3M
J1	1		TERM BLK 6POS SIDE ENTRY 5MM PCB	HDR6	6.91138E+11	Wurth Electronics
J2	1		Receptacle, 2mm, 6x2, Gold, R/A, TH	Receptacle, 2mm, 6x2, R/A, TH	NPPN062FJFN-RC	Sullins Connector Solutions
J3	1			HDR12	NRPN062PARN-RC	Sullins Connector Solutions
LBL1	1		Thermal Transfer Printable Labels, 0.650" W x 0.200" H - 10,000 per roll	PCB Label 0.650 x 0.200 inch	THT-14-423-10	Brady
R4, R7	2	10k	RES, 10 k, 5%, 0.063 W, AEC-Q200 Grade 0, 0402	0402	CRCW040210K0JNED	Vishay-Dale
R5, R6, R8, R9	4	4.7k	RES, 4.7 k, 5%, 0.063 W, AEC-Q200 Grade 0, 0402	0402	CRCW04024K70JNED	Vishay-Dale
SW0	1		Slide Switch SP4T Surface Mount, Right Angle	SMT_SW_11MM3_4MM1	CUS-14TB	Nidec Copal Electronics
TP1, TP2, TP3, TP5, TP7, TP8, TP10, TP11	8		Test Point, Miniature, SMT	Testpoint_Keystone_Miniature	5015	Keystone
U1	1		40-V, 16-Bit, Precision Digital Power/Energy/Charge Monitor With Integrated Sense Element	PowerWCSP8	INA700AYWFR/ INA701AYWFR	Texas Instruments
U2	1		Single Buffer/Driver With Open-Drain Output, DRL0005A, LARGE T&R	DRL0005A	SN74LVC1G07DRLR	Texas Instruments

5 Additional Information

5.1 Trademarks

Chrome® is a registered trademark of Google LLC.
Firefox® is a registered trademark of Mozilla Foundation.
Safari® is a registered trademark of Apple Inc.
All trademarks are the property of their respective owners.

6 Related Documentation

This user's guide is available from the TI website under literature number SBOU285. Any letter appended to the literature number corresponds to the document revision that is current at the time of the writing of this document. When ordering, identify the document by both title and literature number. Table 6-1 lists documentation related to the EVM. Click the links in Table 6-1 for further information. The device name links to the product web folder on www.ti.com. The literature number links to the document PDF.

Table 6-1. Related Documentation

DOCUMENT TITLE	DOCUMENT LITERATURE NUMBER
INA700 data sheet	SBOSAB4

7 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision * (April 2023) to Revision A (July 2025)	Page
• Updated the numbering format for tables, figures, and cross-references throughout the document	1
• Added the *Get Started* section	1
• Changed the *Overview* section to the *Introduction* section	2
• Moved the *Device Summary* table to the new *Device Information* section	2

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265