

Evaluating the **ADuCM355** Precision Analog Microcontroller with Chemical Sensor Interface

FEATURES

Debug and programming capability of the **ADuCM355**
 Evaluation capability with electrochemical gas sensors
ADT7420 0.5 C accurate temperature sensor via I²C
 USB power option and connection to PC

EQUIPMENT NEEDED

PC
 Electrochemical gas sensor (not provided)

DOCUMENTS NEEDED

ADuCM355 hardware reference manual
ADuCM355 data sheet

SOFTWARE NEEDED

IAR Embedded Workbench

GENERAL DESCRIPTION

The **ADuCM355** system on a chip provides the features needed to bias and to measure a range of different electrochemical sensors. The EVAL-ADuCM355QSPZ allows users to evaluate the performance of the **ADuCM355** when implementing a range of different electrochemical techniques, including chronoamperometry, voltammetry, and electrochemical impedance spectroscopy (EIS).

EVALUATION BOARD PHOTOGRAPH

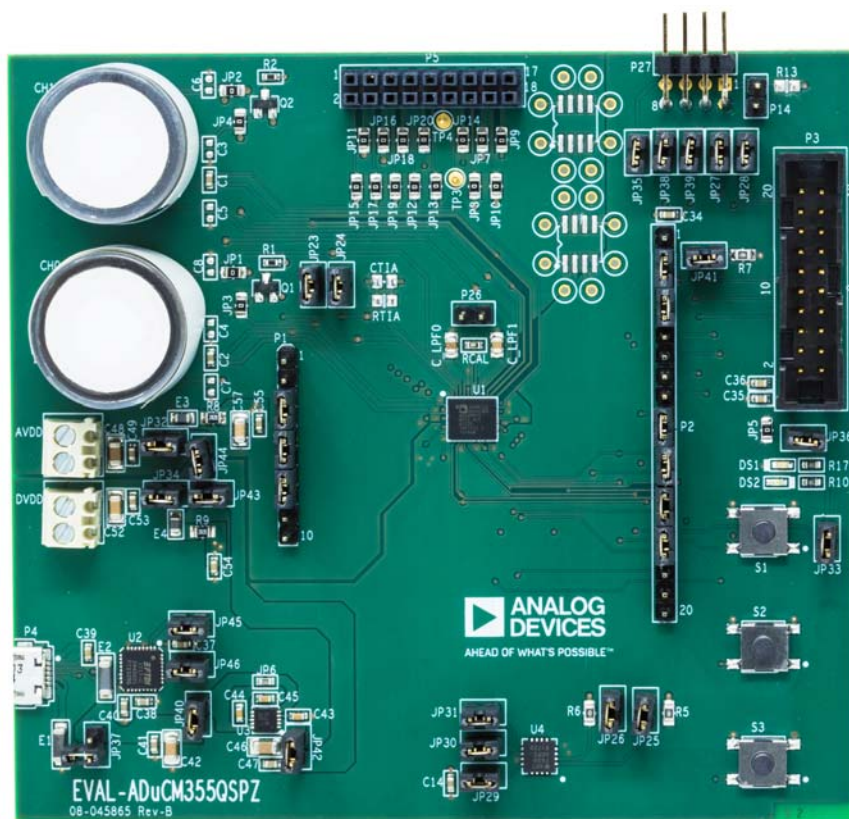


Figure 1.

TABLE OF CONTENTS

Features	1	Installing the ADuCM355 Support Package	7
Equipment Needed	1	Running GPIO Example	7
Documents Needed	1	Application Examples	10
Software Needed	1	Cyclic Voltammetry Example	10
General Description	1	Electrochemical Impedance Spectroscopy (EIS) Example ...	11
Evaluation Board Photograph	1	Chronoamperometry Example	11
Revision History	2	DC Current Example	12
Power Configurations	3	HSTIA Gain Resistor Calibration	13
USB Direct via P4	3	Low Power TIA0/TIA1 Gain Resistor Calibration	16
3.3 V Direct via AVDD and DVDD Connectors	5	Connecting an External Gain Resistor Across the High Speed	
Power via USB from 8-Pin Connector (P27)	5	TIA	19
Power via External 5 V Supply to 2-Pin Connector (P37)	5	AFE Die Watchdog Timer Example	19
Connecting an Electrochemical Sensor	6	4-Lead Electrochemical Sensor Example	19
Getting Started with the IAR Embedded Workbench Tool	7	Mass Erase a Device Not Responding to SWD Commands	21
Downloading IAR	7		

REVISION HISTORY

10/2018—Revision 0: Initial Version

POWER CONFIGURATIONS

This section lists and describes the four different options to power the evaluation board. The different power options include the following:

- Power via a microUSB connector, P4, which is the default power option.
- Connect 3.3 V to the AVDD and DVDD connectors. This setup is useful for measuring current consumption of the evaluation board via the current meter.
- Power via Debug Header P27 (ultimately a different USB connection option to the PC). This setup is useful if using the older serial wire debug (SWD) and universal asynchronous receiver transmitter board (UART), the EVAL-SWD/UART-EMUZ setup.
- Power via the [ADP7158](#) low dropout (LDO) regulator. The USB connector sources 5 V for the LDO regulator.

USB DIRECT VIA P4

MicroUSB Connector, P4 Setup

To power the evaluation board via the microUSB connector, take the following steps:

1. Close JP40, JP42, JP43, JP44, JP45, and JP46.
 - JP45 and JP46 connect the UART pins from the [ADuCM355](#) to the UART-USB transceiver chip (U2) (see Figure 1 and Figure 2).
 - JP40 connects the 5 V USB supply to the input of the LDO (U3) (see Figure 1, Figure 3, and Figure 4).
 - JP42 connects the LDO output (3.3 V) to the board power supply filters (see Figure 1, Figure 3, and Figure 4).
 - JP43 and JP44 connect the DVDD rail to filters for the AVDD_DD and AVDD analog supplies to the [ADuCM355](#) (see Figure 1, Figure 3, and Figure 4).
2. Open JP37.

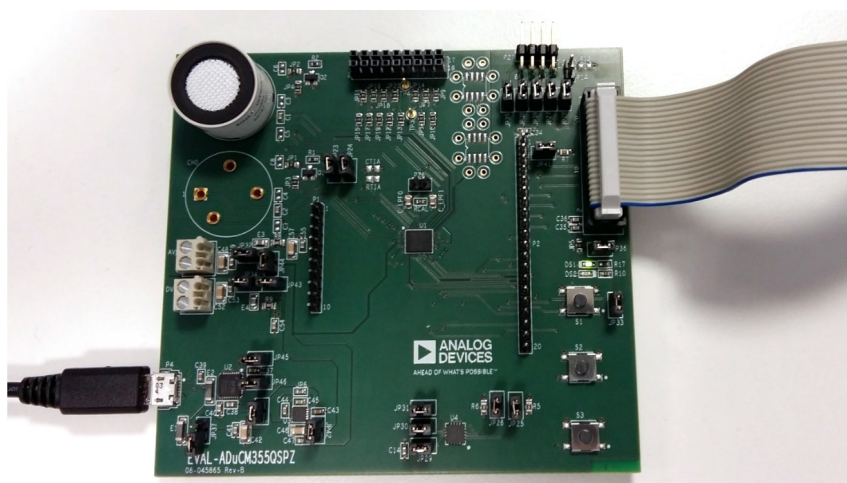
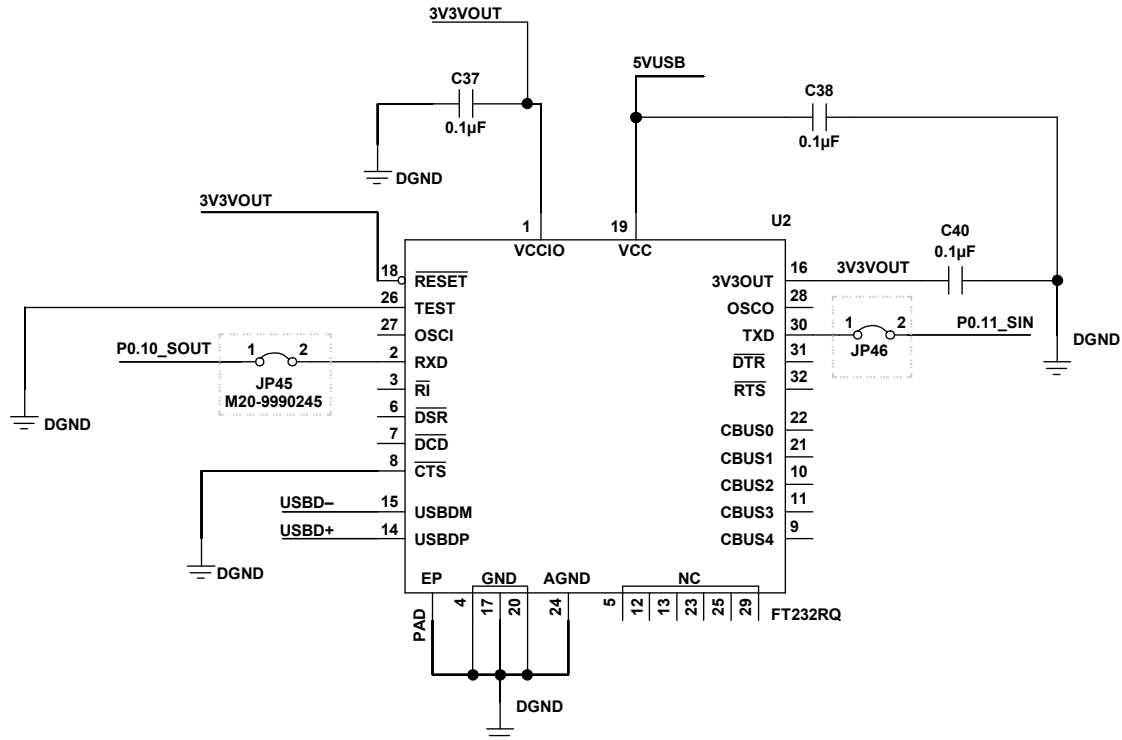
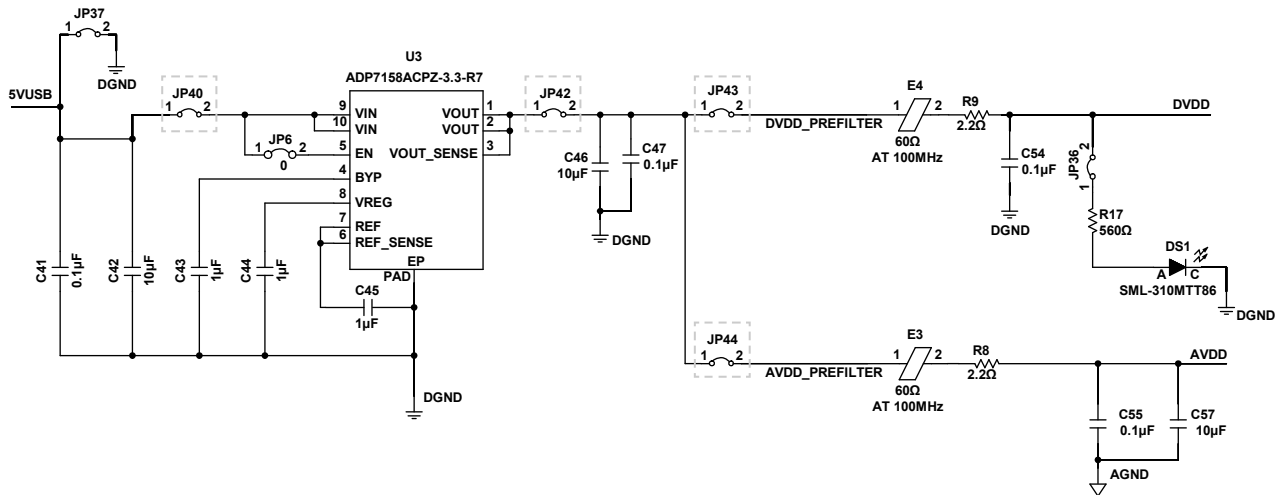


Figure 2. Power via MicroUSB Cable Directly



16887-003



16887-004

3.3 V DIRECT VIA AVDD AND DVDD CONNECTORS

The 3.3 V direct connection to the AVDD and DVDD connectors set up is useful for measuring the current consumption (I_{DD}) of the [ADuCM355](#) itself.

To power the EVAL-ADuCM355QSPZ evaluation board, apply 3.3 V supply directly to Pin 1 on AVDD and DVDD.

Jumper Setup

To set up the jumper, take the following steps:

1. Close JP32, JP34, JP43, and JP44.
2. Open JP37 and JP42.

For additional information, see Figure 6.

POWER VIA USB FROM 8-PIN CONNECTOR (P27)

If using the older USB-SWD/UART connector and debug interface, the [ADuCM355](#) can also be powered from the USB. The UART-USB interface is handled by the USB-SWD/UART board.

Jumper Setup

To set up the jumper, close JP35, JP40, JP42, JP43, and JP44 (see Figure 5).

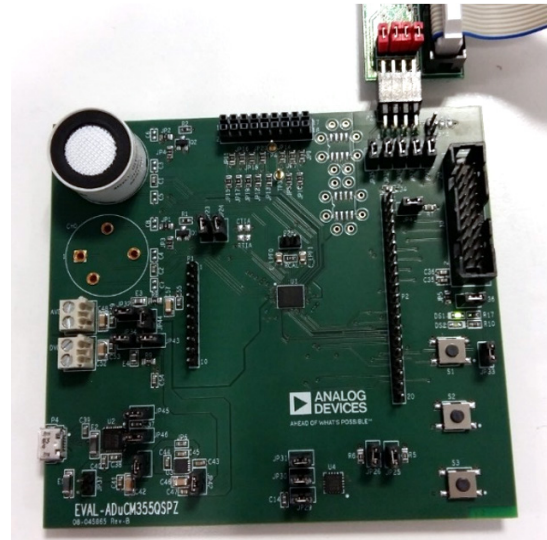


Figure 5. Power via 8-Pin P27 Connector

POWER VIA EXTERNAL 5 V SUPPLY TO 2-PIN CONNECTOR (P37)

The last option is to connect 5 V to the 2-pin header (P37). This 5 V is the input to the [ADP7158](#) LDO regulator that has a 3.3 V output voltage. Do not connect the microUSB cable to P4. This option is a debug and/or test option only.

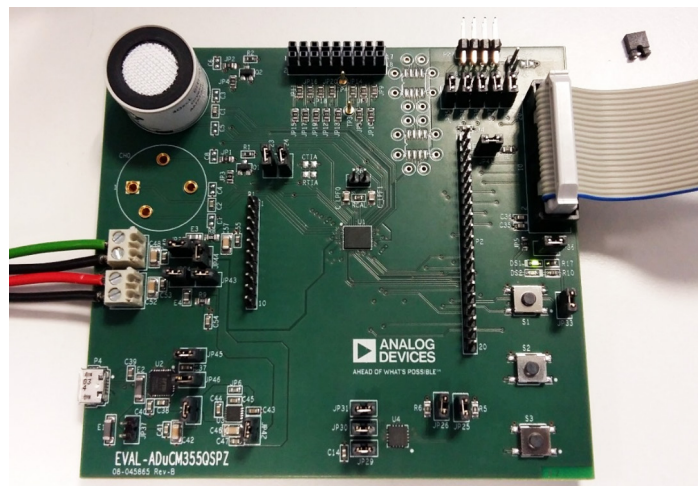


Figure 6. Power DVDD and AVDD Directly via the Power Header Blocks

CONNECTING AN ELECTROCHEMICAL SENSOR

The ADuCM355 has two measurement channels for electrochemical sensors. A 2-lead, 3-lead, or 4-lead sensor can be connected to either CH0 or CH1. Figure 7 shows an electrochemical sensor connected to CH1.

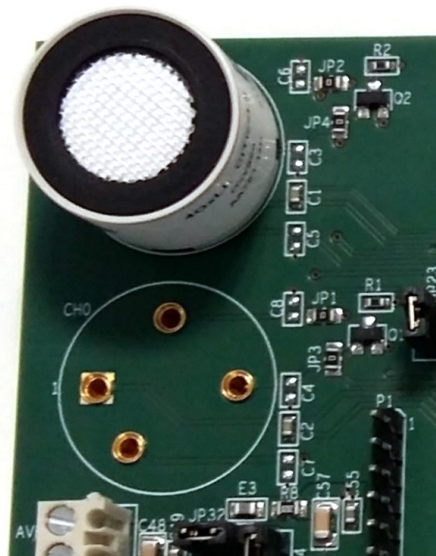


Figure 7. Sensor Connector

GETTING STARTED WITH THE IAR EMBEDDED WORKBENCH TOOL

DOWNLOADING IAR

Ensure you have downloaded and installed a full or evaluation version of the IAR Embedded Workbench for ARM first. Version 8.32.1 or later of the Embedded Workbench for ARM fully support the ADuCM355.

INSTALLING THE ADuCM355 SUPPORT PACKAGE

Download the [ADuCM355 support package](#) and extract the contents. Run the executable and follow the installation instructions to install the required drivers.

Download the ADuCM355 support package and extract the contents. Run the executable and follow the installation instructions to install the required drivers.

Only check the **ADuCM355 IAR EWARM Support** option if you are using an older version than Version 8.32.1 of the IAR Embedded Workbench for ARM, as shown in Figure 8.

When the **ADuCM355 IAR EWARM Support** option is checked, the installer adds a patch to EWARM to add ADuCM355 support. This is not necessary for Version 8.32.1 and later.

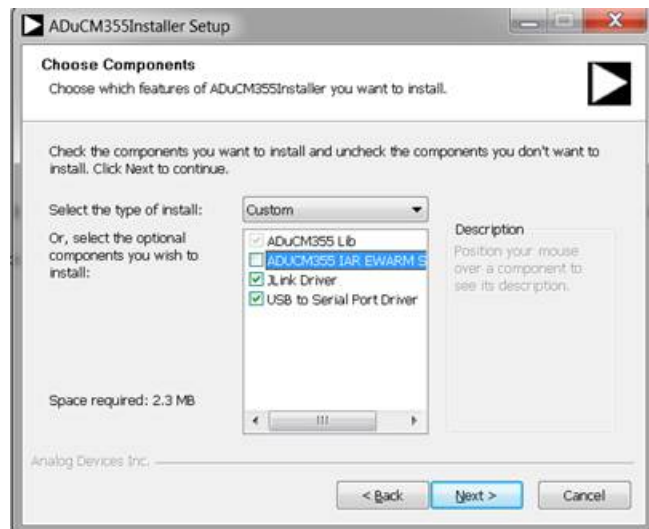


Figure 8. Evaluation Software Installation Options

By default, the sample firmware installs in **C:\Analog Devices\ADuCM355Vx.x.x.x**.

The sample firmware contains the following folders:

- The **Common** folder contains all library files common to every application.
- The **Examples** folder contains specific example projects (see the Running GPIO Example section).
- The **Inc** folder contains some included files for the microprocessor.

RUNNING GPIO EXAMPLE

To run the general purpose input/output (GPIO) example, open the **ADuCM355Vx.x.x.x** directory and go to **examples > M355_GPIO.iar**. Double click **M355_GPIO.eww** to open the project in the IAR Embedded Workbench (see Figure 9).



Figure 9. M355_GPIO.eww File Location

IAR Project Folder Structure

The IAR project folder structure is shown to the left of the IAR Embedded Workbench window (see Figure 10). The **app** folder contains files specific to the open application. In Figure 10, **M355_GPIO.c** is the example shown. The **common** folder contains the required library files for the open application. For the GPIO example, the library files are **AfeWdtLib.c**, **ClkLib.c**, **DioLib.c**, and **UrtLib.c**. The start-up folder contains start-up files for the microprocessor, and the output folder contains the **M355_GPIO.out** file. All subsequent firmware examples follow this folder structure in the IAR Embedded Workbench.

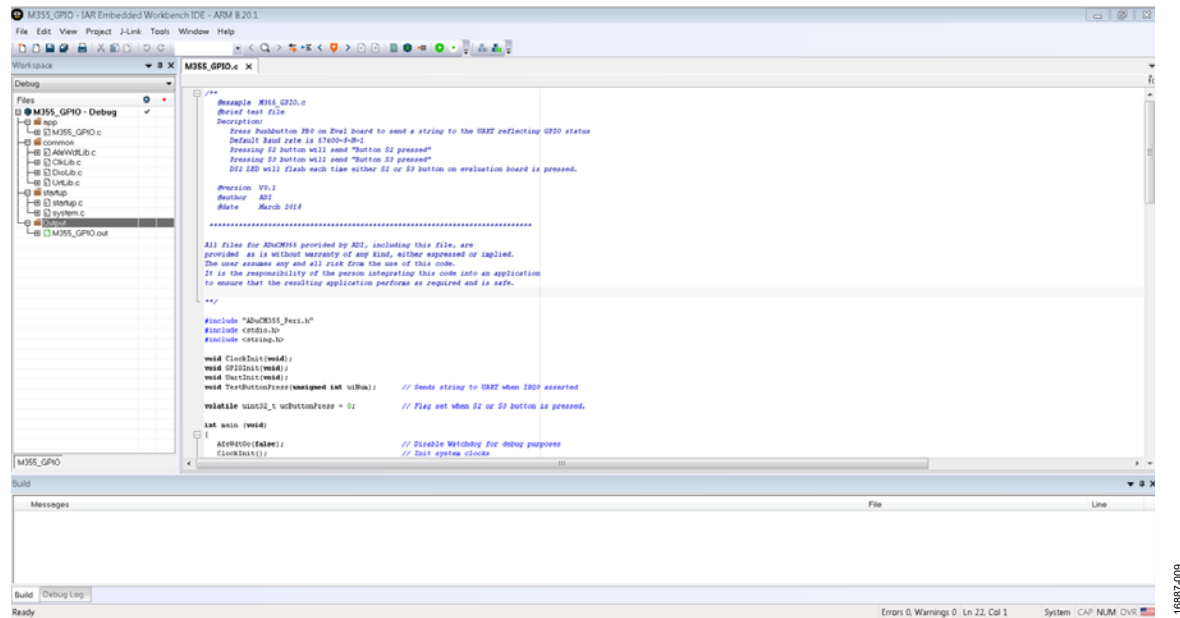


Figure 10. IAR Embedded Workbench

Compiling and Running Firmware

To compile and build the firmware, take the following steps:

1. Go to **Project > Rebuild All** (see Figure 11).

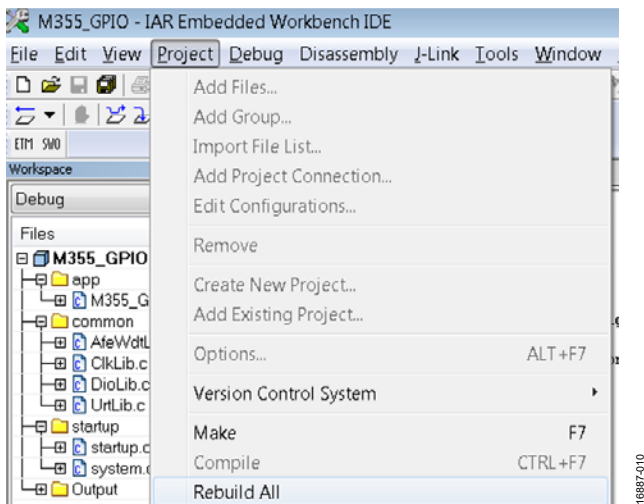


Figure 11. Project > Rebuild All

2. The message shown in Figure 12 appears in the **Build** window.

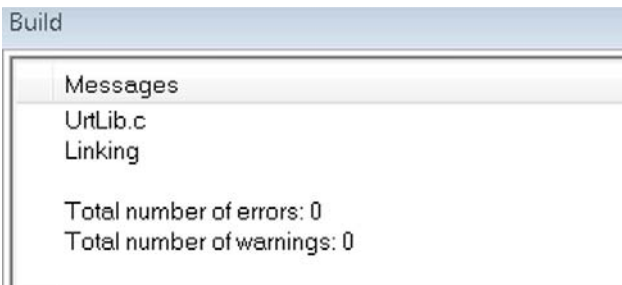


Figure 12. Project > Build

3. To run the firmware on the **ADuCM355**, ensure the evaluation board is powered on and the J-Link debugger is connected. Press **Download and Debug** to load the firmware to the **ADuCM355** and launch the debugger (see Figure 13). Launching and downloading usually takes a few seconds, but the delay can be longer.

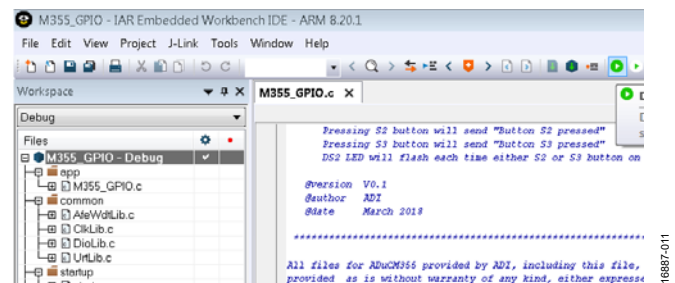


Figure 13. Launching the Debugger

4. Open a terminal program such as RealTerm to view the UART data from the **ADuCM355** (see Figure 14). The baud rate is 57600 bps.

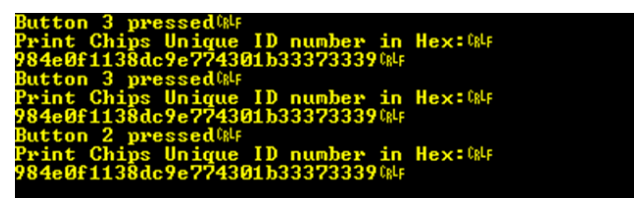


Figure 14. UART Data in RealTerm

5. Figure 15 shows the debug interface. Click on the blue arrow (shown in the red box) to begin code execution. The UART then prompts the user to press either S2 or S3. The LED (DS2) toggles on and off with each button press.

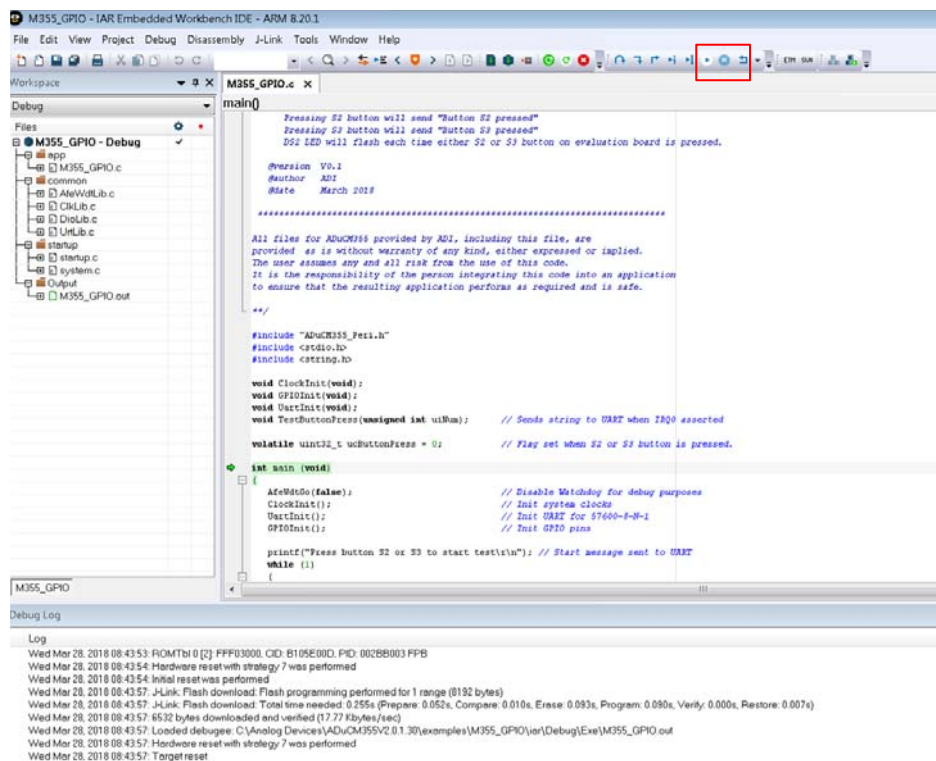


Figure 15. Debug Interface

16887-014

APPLICATION EXAMPLES

CYCLIC VOLTAMMETRY EXAMPLE

Cyclic voltammetry is a common electrochemical measurement in which the current on the sense electrode is measured in response to a ramp like the voltage applied on the counter electrode.

Figure 16 shows a typical, stepped differential voltage between the REx and WEx electrodes of the sensor.

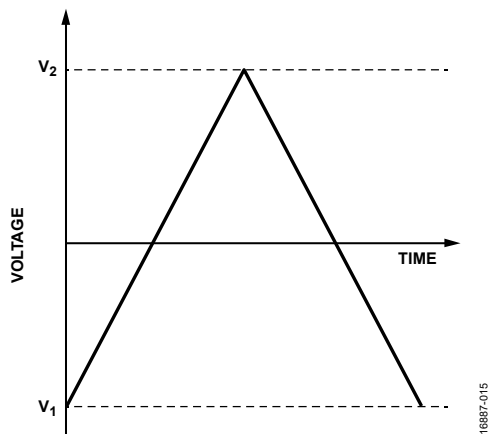


Figure 16. Typical Cyclic Voltammetry Waveform

In the [ADuCM355](#) firmware package, the **M355_Cyclic-Voltammetry** project demonstrates how to implement a cyclic voltammetry measurement on the [ADuCM355](#). In the **M355_Ramp.c** file, there is a description of the measurement. There are also a number of macro definitions that define the measurement.

For this example project, the selected options are as follows:

```
#define SENSOR_DUAL_CHANNEL_ENABLE    0
//Only run test on Channel 0

#define SENSOR_RAMP_SE                1
//Select SE0 electrode for measurement

#define REC_OPTION                    REC_ADCDAT
//Record ADC Data

#define OPT_RAMP_MEAS.                1
//Measure current and pin voltages

#define SNS_DC_VBIAS_DEFAULT0 sets the
Channel 0 VBIAS default voltage

#define SNS_DC_VZERO_DEFAULT0 sets the
Channel 0 VZERO default voltage

#define SNS_DC_RLOAD_DEFAULT0 sets the
Channel 0 low power TIA (LPTIA) RLOAD
default value

#define SNS_DC_RGAIN_DEFAULT0 sets the
Channel 0 gain resistor (RTIA) of the LPTIA
gain default value
```

To test the firmware, constructing a dummy electrochemical cell using 1 kΩ resistors in a star network is recommended (see Figure 17). Connect each pin of the resistor network to CE0, RE0, SE0, and DE0 on P5.

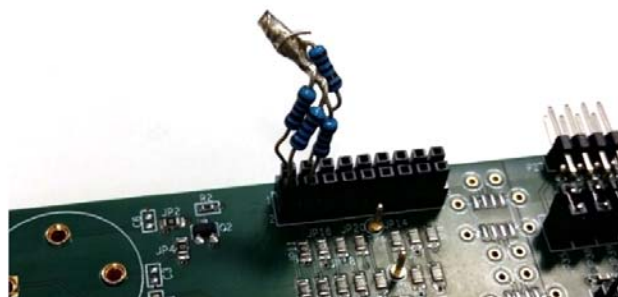


Figure 17. Resistor Star

To begin measuring and gathering data, open a terminal program such as RealTerm. Configure the baud rate for 115200. Compile and build the project in the IAR Embedded Workbench and launch the debugger. Run the measurement and save the data to a **.csv** file for processing. If **OPT_RAMP_MEAS** is set to 1, there are six measurements for each channel, which includes the following:

- Current on Channel 1
- Voltage on DE0
- Voltage on RE0
- VZERO voltage
- VBIAS0 voltage
- Current

To plot the current response of the ramp test, open the saved **.csv** file in Microsoft® Excel. The first set of data points under **RAMP_DATA** is the current measurements. Figure 18 shows the **RAMP_DATA** data points after being plotted.

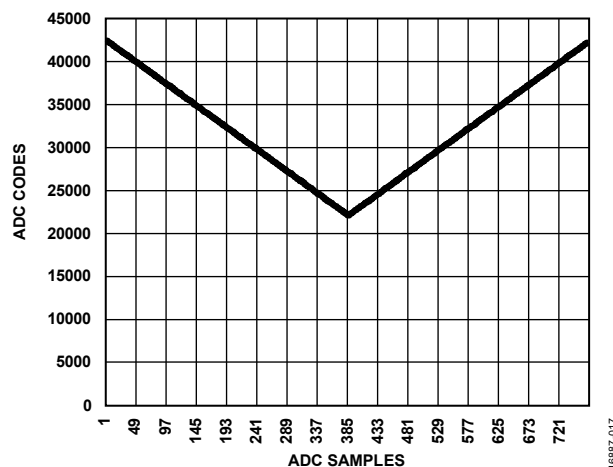


Figure 18. Example SE0 Channel Current Measurement—ADC Codes on the Y-Axis, and ADC Sample Number on X-Axis

ELECTROCHEMICAL IMPEDANCE SPECTROSCOPY (EIS) EXAMPLE

EIS is a common electrochemical measurement in which an ac excitation signal is applied to an electrochemical cell. The response current is measured, and the impedance is calculated.

On the ADuCM355, the EIS measurement is a three step process. The response current in each step is measured using a high speed transimpedance amplifier (HSTIA).

The EIS measurement process is as follows:

1. Signal is applied across R_{CAL} .
2. Signal is applied across R_{LOAD} .
3. Signal is applied across $Z_{SENSOR} + R_{LOAD}$.

R_{CAL} is a precision resistor connected to the RCAL0 pin and the RCAL1 pin, R_{LOAD} is the internal load resistor on the SE0 path, and Z_{SENSOR} is the impedance under test.

Use the following equation to calculate the actual impedance:

$$Z_{SENSOR} = Z_{SENSOR+R_{LOAD}} - Z_{R_{LOAD}}$$

where:

$Z_{SENSOR+R_{LOAD}}$ is the impedance of $R_{SENSOR} + R_{LOAD}$ connected together.

$Z_{R_{LOAD}}$ is the impedance of R_{LOAD} .

For the purpose of this initial test, a dummy electrochemical cell was used. Connect three 1 k Ω resistors in a star network and connect to the resistors to the CE0, RE0, and SE0 pins on P5 (see Figure 19).



Figure 19. 1 k Ω Resistor Star Connected on P5

The two user options in the M355_ECSns_EIS.c file are as follows:

- The #define EIS_DCBIAS_EN macro defines whether there is a dc bias applied across the sensor as is required for oxygen sensors.
- Select the frequency of the sine wave excitation signal. This is defined in the ImpResult [] array. By default, 12 frequencies are already populated. These frequencies can be modified as required.

To run the impedance measurement, take the following steps:

1. Launch the debugger in the IAR Embedded Workbench.
2. Open a terminal program with a 57600 baud rate.
3. Start to execute the code.
4. A prompt to press S2 is sent over the UART and displayed in the terminal. Press S2 to begin the impedance test. It

takes up to 20 seconds because there are 12 frequency points to measure.

5. When the impedance measurement completes, the results are sent to the UART (see Figure 20). Optionally, save the results in a Microsoft Excel file for further analysis.

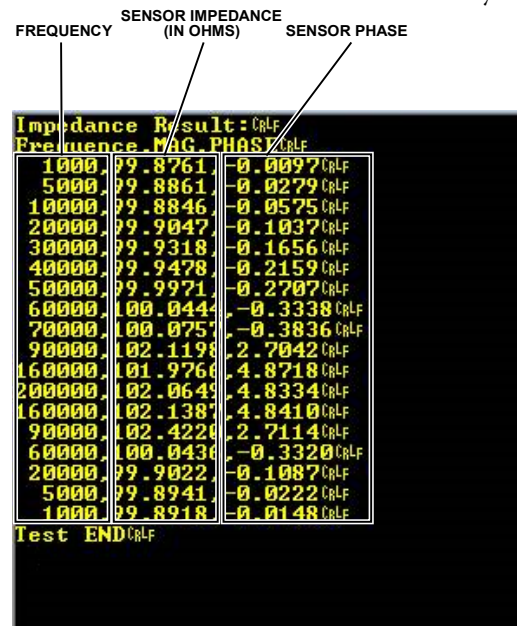


Figure 20. Impedance Results

CHRONOAMPEROMETRY EXAMPLE

Chronoamperometry is an electrochemical technique in which the voltage applied to an electrochemical cell is stepped. The response current on the sense electrode is measured. Figure 21 and Figure 22 show a typical chronoamperometric measurement and sensor response.

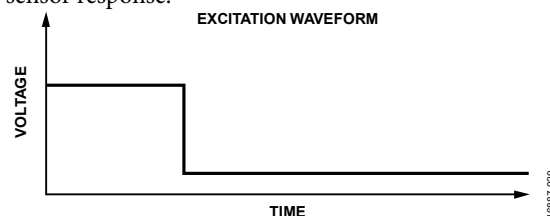


Figure 21. Typical Chronoamperometric Voltage Stimulus Waveform

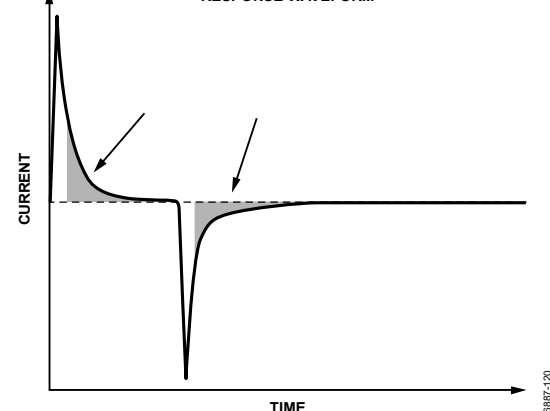


Figure 22. Typical Chronoamperometric Current Response Waveform

In the [ADuCM355](#) firmware development package, the **M355_ECSns_CapaTest** project implements a chrono-amperometric measurement.

The **M355_ECSns_CapaTest.c** file contains a data structure defined to select measurement options. The user can change these measurement options to modify measurements.

There is also a structure (**SNS_CFG_Type**) that sets up a data structure to configure each sensor channel. For this demonstration, only Sensor Channel 0 is used.

All default values are used. The resistor star model is connected to P5 as was the case in the examples described in the Cyclic Voltammetry Example section and the Electrochemical Impedance Spectroscopy (EIS) Example section.

Load the project in the IAR Embedded Workbench and open a terminal program. Compile and build the project and launch the debugger interface. Start the code execution and save the UART data to a .csv file for processing.

The example code sends three arrays of results to the UART, 8191 of each (see Figure 23):

- First 8191 values are the current measurement results for the SE0 channel in μA .
- The next 8191 values are the voltage measurement results for the RE0 channel in mV.
- The final 8191 values are the voltage measurement results for the SE0 channel in mV.

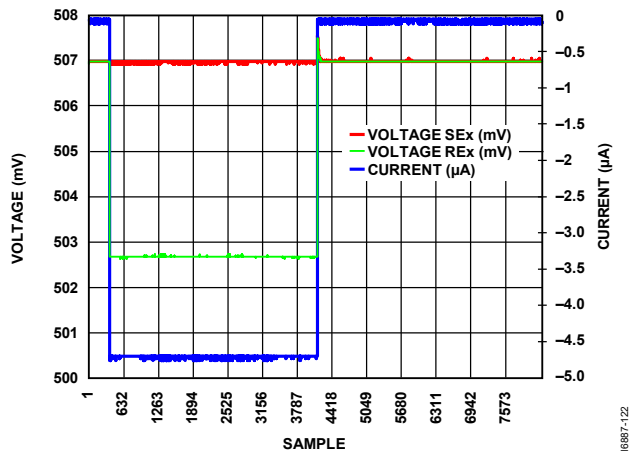


Figure 23. Output Data Using **M355_ECSns_CapaTest** Example with Three 100 Ω Resistors

```
CO Sensor Initializing OK
O2 Sensor Initializing OK
1. CH0, 21819, CH1, 25227, OK
2. CH0, 23912, CH1, 25780, OK
3. CH0, 23958, CH1, 26238, OK
4. CH0, 23995, CH1, 26617, OK
5. CH0, 24025, CH1, 26932, OK
6. CH0, 24052, CH1, 27195, OK
7. CH0, 24072, CH1, 27415, OK
```

Figure 24. CO Sensor on Channel 0 Output, Channel 1 Floating

DC CURRENT EXAMPLE

The dc current is a standard electrochemical measurement. Depending on the sensor type, a bias voltage is applied between the reference and sense electrodes. The current output on the sense electrode is measured.

In the [ADuCM355](#) firmware package, the **M355_ECSns_DCTest** project implements a dc current measurement on two electrochemical cells connected to Channel 0 (CH0) and Channel 1 (CH1). CH0 is configured for a zero biased sensor, $V_{\text{BIAS}} - V_{\text{ZERO}} = 0$. A carbon monoxide (CO) sensor is an example of a zero biased sensor. CH1 is configured for a biased sensor, $V_{\text{BIAS}} - V_{\text{ZERO}} = 600 \text{ mV}$. An oxygen (O_2) sensor is an example of a biased sensor.

For the purpose of testing, connect the 1 k Ω resistor star network to CH0 and CH1. Ensure that the [ADuCM355](#) board is powered, and that the debugger is connected to the PC. Then, open the project in the IAR Embedded Workbench, build, and run the debugger. Open a terminal program to view the results. The output is the raw analog-to-digital converter (ADC) codes (see Figure 24).

HSTIA GAIN RESISTOR CALIBRATION

HSTIA has three different programmable gain resistor options (see Figure 25).

The user adjusts the gain resistors shown in Figure 25 to convert the current from the SE0, SE1, and DE0 inputs or the DE1 input to a differential voltage across the R_{TIA2} resistor, R_{TIA2_03} resistor, or R_{TIA2_05} resistor.

The R_{TIA2} , R_{TIA2_03} , and R_{TIA2_05} resistors have an initial accuracy range as specified in the ADuCM355 data sheet. The resistors also vary with temperature as specified in the ADuCM355 data sheet.

If the HSTIA is uncalibrated for the selected gain resistor and the ADC PGA setting, an error will be present when measuring an absolute input current.

By using the high speed DAC (HS DAC) to create a differential voltage across an external precision R_{CAL} resistor that is connected to the ADuCM355 R_{CAL0} pin and R_{CAL1} pin, a precision calibration current can generate, which is routed through any of the three HSTIA gain resistors.

Because the calibration current value is known, and the ADC can measure the voltage drop across the R_{TIA2} resistor, R_{TIA2_03} resistor, and R_{TIA2_05} resistor, the exact $R_{TIA2}/R_{TIA2_03}/R_{TIA2_05}$ value can be determined.

Figure 26 shows the setup and switch settings that connect the HS DAC output to the external R_{CAL} resistor and the current flow into the HSTIA and R_{TIA2} gain resistor.

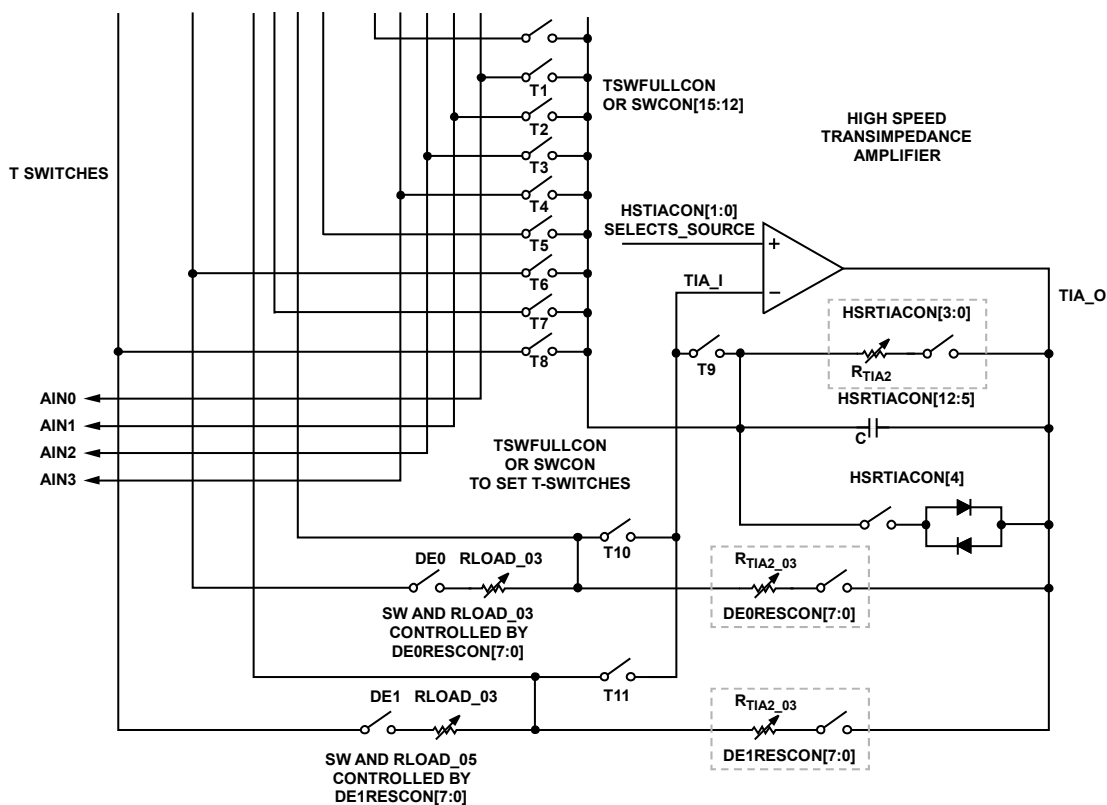
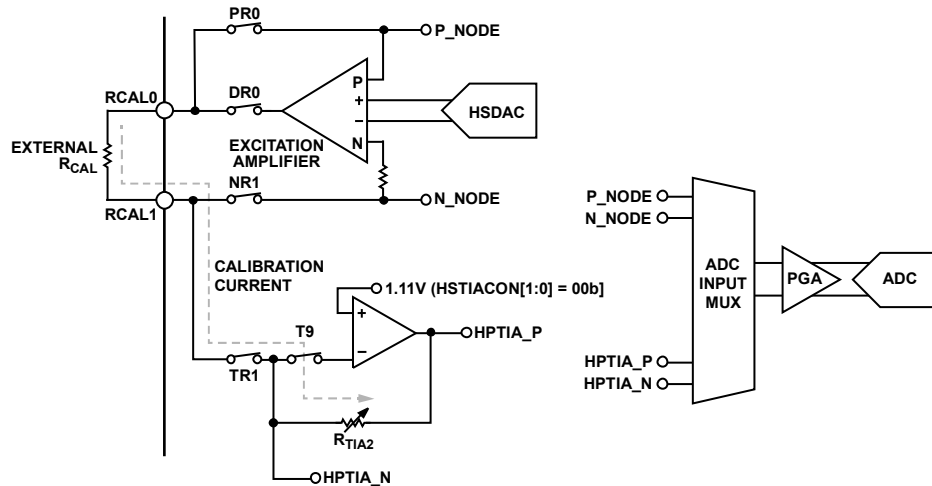
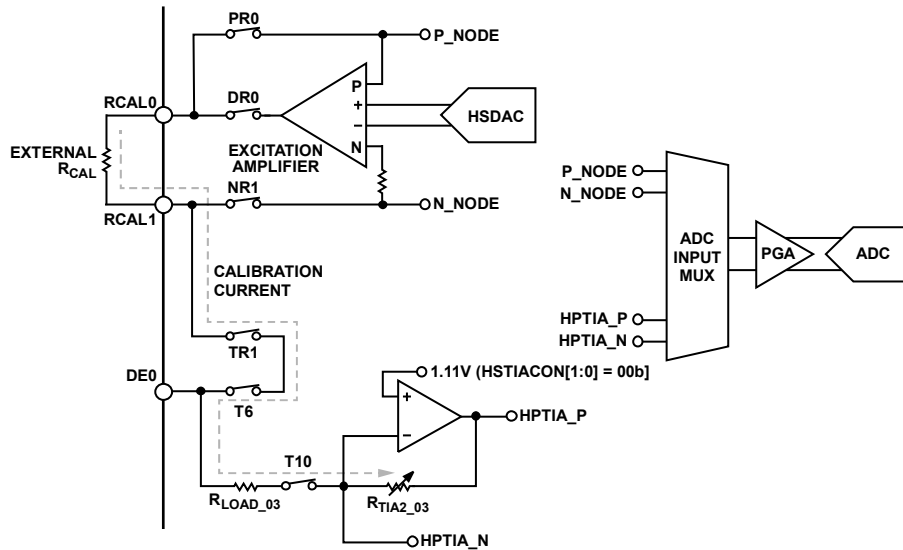
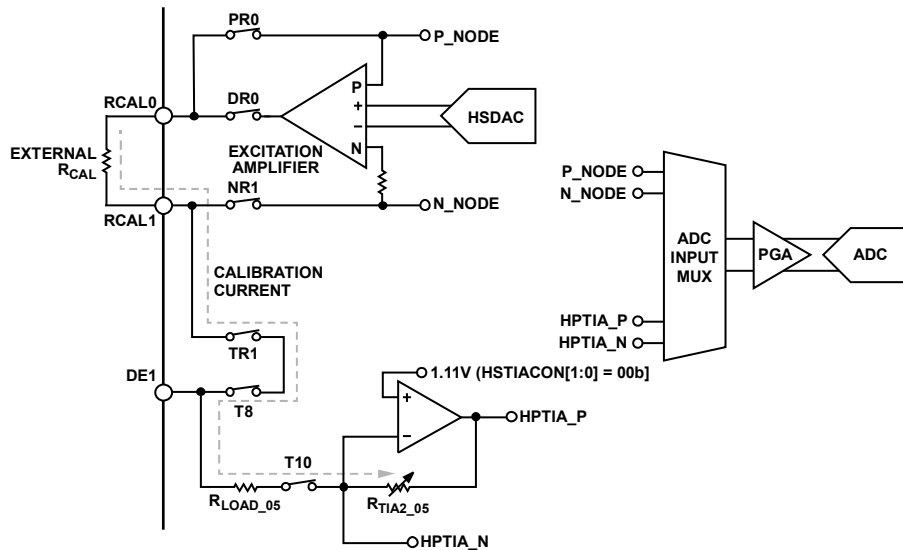


Figure 25. HSTIA Programmable Gain Resistors

16887-002

Figure 26. HS DAC, HSTIA, and Switch Matrix Settings for R_{TIA2} CalibrationFigure 27. HS DAC, HSTIA, and Switch Matrix Settings for R_{TIA2_03} CalibrationFigure 28. HS DAC, HSTIA, and Switch Matrix Settings for R_{TIA2_05} Calibration

The **M355_HSRTIA_Calibration** code example project in the **examples** folder demonstrates how to configure the device to calibrate the 5 kΩ setting for R_{TIA2} , R_{TIA2_03} , and R_{TIA2_05} .

The example code sends text to the UART similar to that shown in the Figure 29. The baud rate used is 57600-8-N-1. The text can be viewed by a viewing terminal such as RealTerm or HyperTerminal. To use the **M355_HSRTIA_Calibration** example code, take the following steps:

1. Press the **S3** button on the EVAL-ADuCM355QSPZ evaluation board to start the calibration sequence.
 - The program calibrates the ADC offset and gain errors for LP mode with a programmable gain amplifier (PGA) gain setting of 1.5.
 - The HS DAC offset error is calibrated out.
 - The required HS DAC differential voltage is generated across the external R_{CAL} resistor
2. After calibration, apply a current source to any of the SE0, DE0, or DE1 inputs. Press the **S2** button to trigger a measurement of the input currents to the SE0 pin, DE0 pin, and DE1 pin. **S2** may also be pressed before calibration to allow the user to compare the precalibration and post-calibration results.

```

ADC Offset Pre-Calibration 7d7 0x112000 mV
ADC Offset Post-Calibration 7cb 0x037333 mV
Old ADC Gain Register Pre-Calibration 3f73
Voltage Urefs 1.82-01.11 Pre-Calibration
709.890259 mV
ADC Gain Register Post-Calibration 3f74
ADC Full-scale voltage Post Calibration:
709.927612 mV
HSDAC LP Atten Offset calibration Old Value: 0
-0.746663 mV
HSDAC LP Atten Offset calibration New Value: 38
-0.485331 mV
14.671937 mV
Voltage Across RCAL for RTIA calibration in mV 14.671937
*** HPTIA 0 Calibration Current: 73.359680uA
*** Voltage Drop across RTIA 0.414361
*** HPTIA 0 Resistor 5648.346191 ohms
14.671937 mV
Voltage Across RCAL for RTIA calibration in mV 14.671937
*** HPTIA 1 Calibration Current: 73.359680uA
*** Voltage Drop across RTIA 0.417310
*** HPTIA 1 Resistor 5688.550293 ohms
14.671937 mV
Voltage Across RCAL for RTIA calibration in mV 14.671937
*** HPTIA 2 Calibration Current: 73.359680uA
*** Voltage Drop across RTIA 0.419886
*** HPTIA 2 Resistor 5723.664063 ohms
  
```

Figure 29. **M355_HSRTIA_Calibration** Output Text to UART Port

10887-026

LOW POWER TIA0/TIA1 GAIN RESISTOR CALIBRATION

The ADuCM355 contains two independent, low power TIA channels.

Each TIA has an independent, programmable gain resistor to scale the input current from the SE0 pin and the SE1 pin to a voltage that the ADC can measure.

Figure 30 shows the gain resistor for the low power Channel 0 (LPTIA0). A similar diagram is valid for low power Channel 1 (LPTIA1).

Similar to the example described in the HSTIA Gain Resistor Calibration section, the user adjusts the gain resistor to convert the current from the SE0 input pin and the SE1 input pin to a differential voltage across the RTIA resistors.

These resistors have an initial accuracy range as specified in the ADuCM355 data sheet. These resistors also vary with temperature as specified in the ADuCM355 data sheet.

When uncalibrated, an error is present when trying to measure an absolute input current.

By using the low power DAC to create a differential voltage across an external precision R_{CAL} resistor that is connected to the ADuCM355 RCAL0 pin and RCAL1 pin, a precision calibration current can be generated and routed through either the LPTIA0 gain resistor or the LPTIA1 gain resistor.

Because the calibration current value is known and the ADC can measure the voltage drop across each RTIA resistor, the exact RTIA value can be determined.

Figure 31 and Figure 32 show the setup and switch settings used to connect the low power DAC outputs to the external R_{CAL} resistor and the current flows into the LPTIA gain resistors (LPRTIAx).

The **M355_LPTIA_Calibration** code example project in the **examples** folder demonstrates how to configure the device to calibrate the 4 k Ω setting for RTIA for the both LPTIA0 and LPTIA1.

The example code sends text to the UART similar to that shown in Figure 33. The baud rate used is 57600-8-N-1. View the text in a viewing terminal such as RealTerm or HyperTerminal.

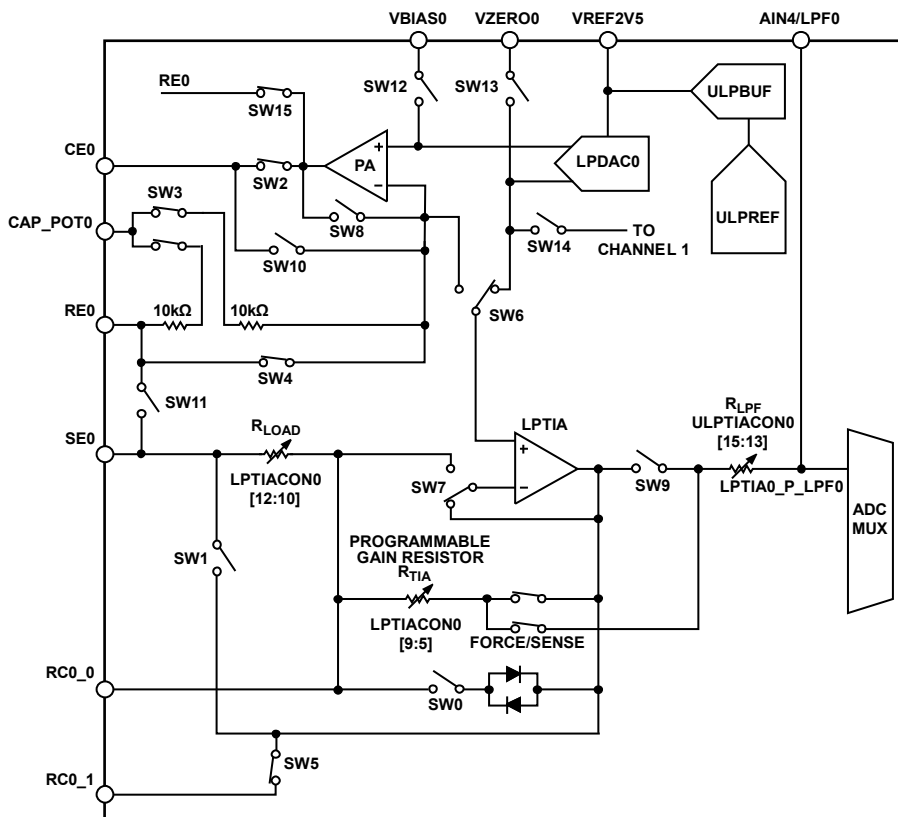


Figure 30. LPTIA0 Gain Calibration Resistor

18687-027

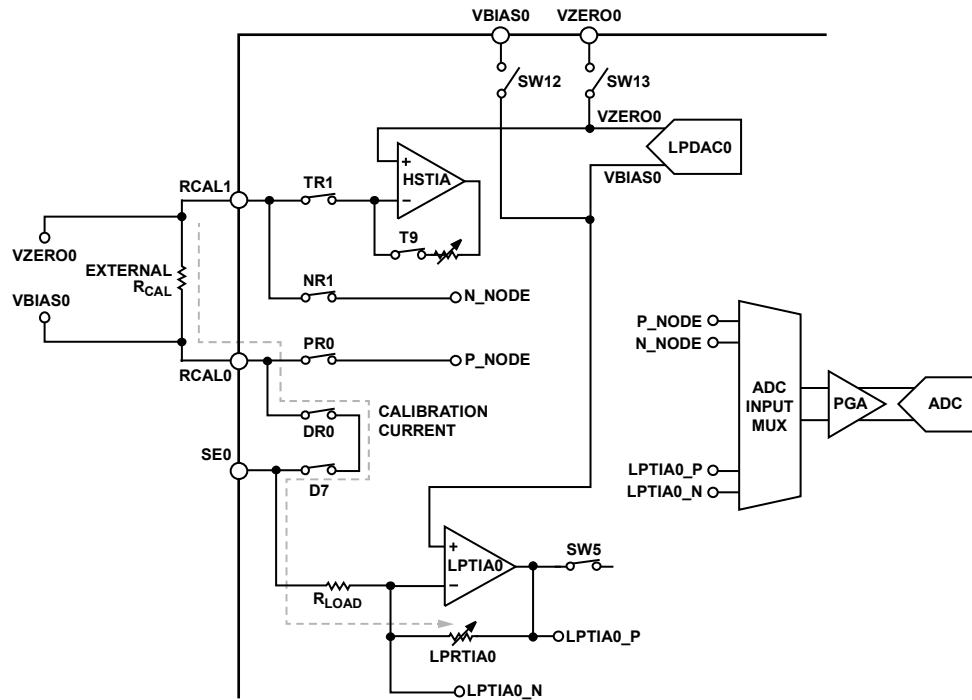


Figure 31. HSTIA, LPTIA0, and Switch Matrix Settings for LPRTIA0 Calibration

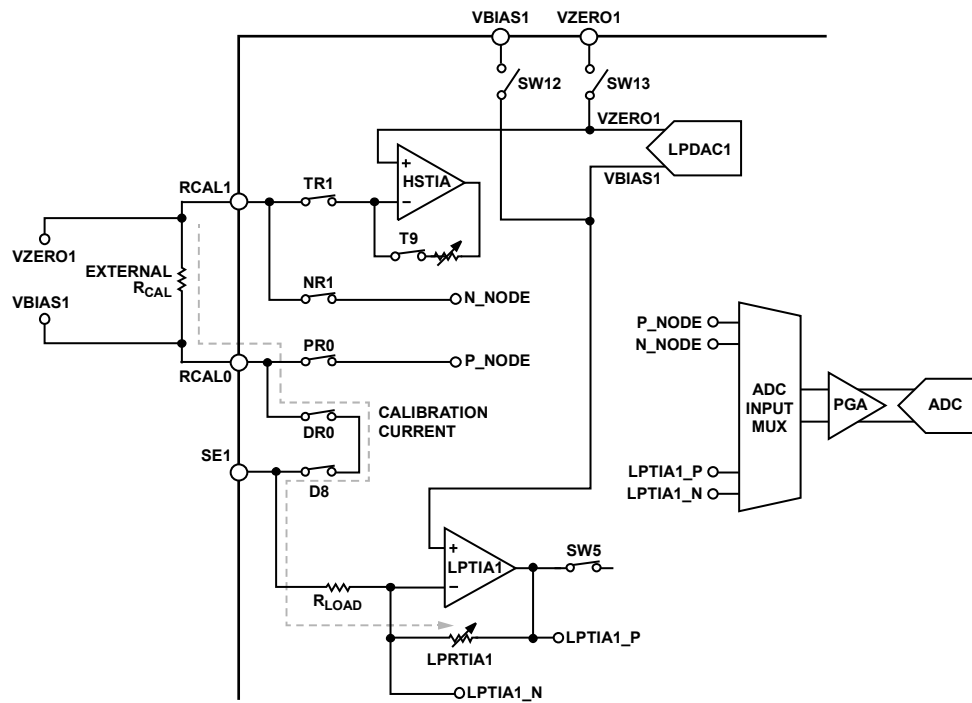


Figure 32. HSTIA, LPTIA1, and Switch Matrix Settings for LPRTIA1 Calibration

```

LP LPTIA Gain resistor Calibration (uA)
Press S2 button to measure Current from SE0/SE1 pins .. (uA)
Or, Press S3 button to measure LPTIA0/LPTIA1.. (uA)

ADC Offset Pre-Calibration 7d7 (uA)
0.112000 mU (uA)
ADC Offset Post-Calibration 7cb (uA)
0.032333 mU (uA)
Old ADC Gain Register Pre-Calibration 3f73 (uA)
Voltage Urefs 1.82-01.11 Pre-Calibration (uA)
-1104.053833 mU (uA)
ADC Gain Register Post-Calibration 14000 (uA)
ADC Full-scale voltage Post Calibration: (uA)
-1113.648560 mU (uA)
24.117229 mU (uA)
Voltage Across RCAL for RTIA calibration in mU 24.117229 (uA)
*** LPTIA 0 Calibration Current: 120.586151uA (uA)
*** Voltage Drop across RTIA 0.524270 (uA)
*** LPTIA 0 Resistor 4347.677246 ohms (uA)
24.303894 mU (uA)
Voltage Across RCAL for RTIA calibration in mU 24.303894 (uA)
*** LPTIA 1 Calibration Current: 121.519470uA (uA)
*** Voltage Drop across RTIA 0.528675 (uA)
*** LPTIA 1 Resistor 4350.537598 ohms (uA)
LPTIA Calibration completed... (uA)
Press S2 button to measure current into SE0 & SE1 pins (uA)
LP Current Channel measurement in progress (uA)
Wait a moment for results .... (uA)
*** Voltage Drop across RTIA -0.088368 U (uA)
*** LPTIA 0 Current -20.325247 uA (uA)
*** Voltage Drop across RTIA 0.000075 U (uA)
*** LPTIA 1 Current 0.017163 uA (uA)

```

ADC OFFSET CALIBRATION

ADC GAIN CALIBRATION

LPTIA0 GAIN RESISTOR VALUE

LPTIA1 GAIN RESISTOR VALUE

Figure 33. M355_LPTIA_Calibration Output Text to UART Port

```

if (LPTIAL > DEFAULT_RTIA) // Demonstrate how to write to LPTIAL Gain calibration register
{
    fCalAdjust = (float)(pADI_AFE->ADCGNLPTIAL * DEFAULT_RTIA)/LPTIAL;
    u32CalAdjust = (uint32_t)fCalAdjust;
    pADI_AFE->CALDATLOCK = 0x0687A5AF; // Write to unlock key to gain access to Calibration registers
    pADI_AFE->ADCGNLPTIAL = u32CalAdjust; // Write to unlock key to gain access to Calibration registers
}
else
{
    fCalAdjust = (float)(pADI_AFE->ADCGNLPTIAL * LPTIAL)/DEFAULT_RTIA;
    u32CalAdjust = (uint32_t)fCalAdjust;
    pADI_AFE->CALDATLOCK = 0x0687A5AF; // Write to unlock key to gain access to Calibration registers
    pADI_AFE->ADCGNLPTIAL = u32CalAdjust; // Write to unlock key to gain access to Calibration registers
}

```

Figure 34. Loading the ADC Gain Calibration Register

To use the **M355_LPTIA_Calibration** sample code, take the following steps:

- Press the **S3** button on the EVAL-ADuCM355QSPZ evaluation board to start the calibration sequence.
 - The program first calibrates the ADC offset and gain errors for LP mode with a PGA gain setting of 1.5.
 - The required LP DAC differential voltage is generated across the external R_{CAL} resistor
 - The voltage across R_{CAL} is measured via the ADC P_NODE input channel and N_NODE input channel. This method of measurement allows the user to determine the calibration current i .
 - The calibration current i is then switched into each of the RTIAs across the LPTIA0 amplifier and LPTIA1 amplifier and the voltage drop across each measured in turn.
 - The ADC LPTIAx_P input channel and LPTIAx_N input channel are measured for each LPTIA (see Figure 31).

- After calibration, apply a current source to either the SE0 input or SE1 input and press **S2** to trigger a measurement of the input currents to the SE0 pin and the SE1 pin. The **S2** button can also be pressed before calibration to allow the user to compare the precalibration and postcalibration results.

Do not apply a current to the SE0 pin or the SE1 pin while the calibration is taking place.

Figure 34 shows how to load the ADC gain calibration register for the LPTIA1 channel (ADCGNLPTIA1) in the main while() loop.

CONNECTING AN EXTERNAL GAIN RESISTOR ACROSS THE HIGH SPEED TIA

The internal high speed transimpedance amplifier (TIA) has a programmable gain resistor that allows users to configure a high speed current measurement channel for different input current ranges. However, there is an option to connect an external gain resistor instead.

The EVAL-ADuCM355QSPZ supports the connection of an external RTIA resistor across the AIN0 pin and DE0 pin, which are labelled RTIA on the top side of the printed circuit board (PCB).

The current into the HSTIA flows from AIN0 into the HSTIA inverting input with the HSTIA connected to DE0.

The ADC measures the voltage drop across the external RTIA by selecting the input channels HPTIA_P and HPTIA_N (see Figure 35).

When the external gain resistor is populated by the customer, this gain resistor can be used instead of the internal R_{TIA2} resistor, R_{TIA2_03} resistor, and R_{TIA2_05} resistor.

The **M355_ExternalRTIA** code example project in the **examples** folder shows how to setup the HSTIA for an external gain resistor.

AFE DIE WATCHDOG TIMER EXAMPLE

The **ADuCM355** supports a watchdog timer on the analog front end (AFE) die. The watchdog timer clocks via an oscillator that is completely independent of the clocks in the Cortex-M3 core and thus meets the IEC 61508 requirement of an independent watchdog timer for a microcontroller and saves the need for an external watchdog timer chip.

The **M355_AfeWdt** code example project in the **examples** folder shows how to configure the windowed watchdog mode.

The **WDT_INTERRUPT_EN** #define parameter configures the project to generate a reset or an interrupt.

The default timeout period used by the project is 16 sec with a minimum period of 4 sec required before a watchdog refresh is allowed, refreshing the watchdog within 4 sec, which causes a reset or interrupt to occur.

The refresh of the watchdog timer is triggered by sending the ASCII Character 1 from a PC.

4-LEAD ELECTROCHEMICAL SENSOR EXAMPLE

Many electrochemical sensors come in 4-lead packages. The 4-lead packages referred to have a counter, reference, and two sensing electrodes. The **ADuCM355** supports biasing and measuring of these sensor types.

This example project is based on the CiTiceL 4COSH dual gas sensor (carbon monoxide, CO, and hydrogen sulphide, H2S). The current flowing from the SE0 electrode indicates H2S levels. Current flowing from the SE1 electrode indicates the CO level.

This example project configures the low power, potentiostat Channel 0 to bias the sensor. The current flowing to and from SE0 is measured via the low power TIA0 channel. The current flowing from the SE1 electrode is measured via the low power TIA1 channel.

The TIA amplifiers convert the current to a voltage that is measured via the ADC. The source code then calculates the parts per million (PPM) of the gas for each channel based on this measurement. The results are sent to the UART in ASCII format as shown in Figure 36.

The **M355_ECSns_DualSensor** code example project is in the **examples** folder.

Figure 37 shows the connection details between the 4-lead sensor and the **ADuCM355**.

For most accurate absolute measurements, combine this example with the **Low Power TIA0 Gain resistor Calibration** example project.

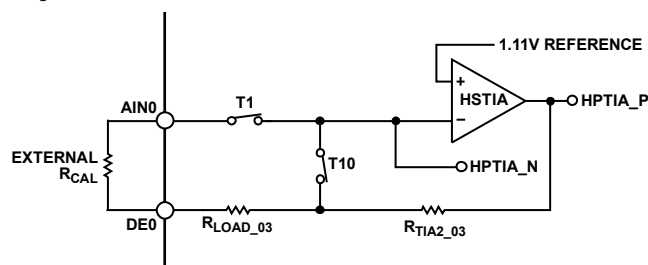
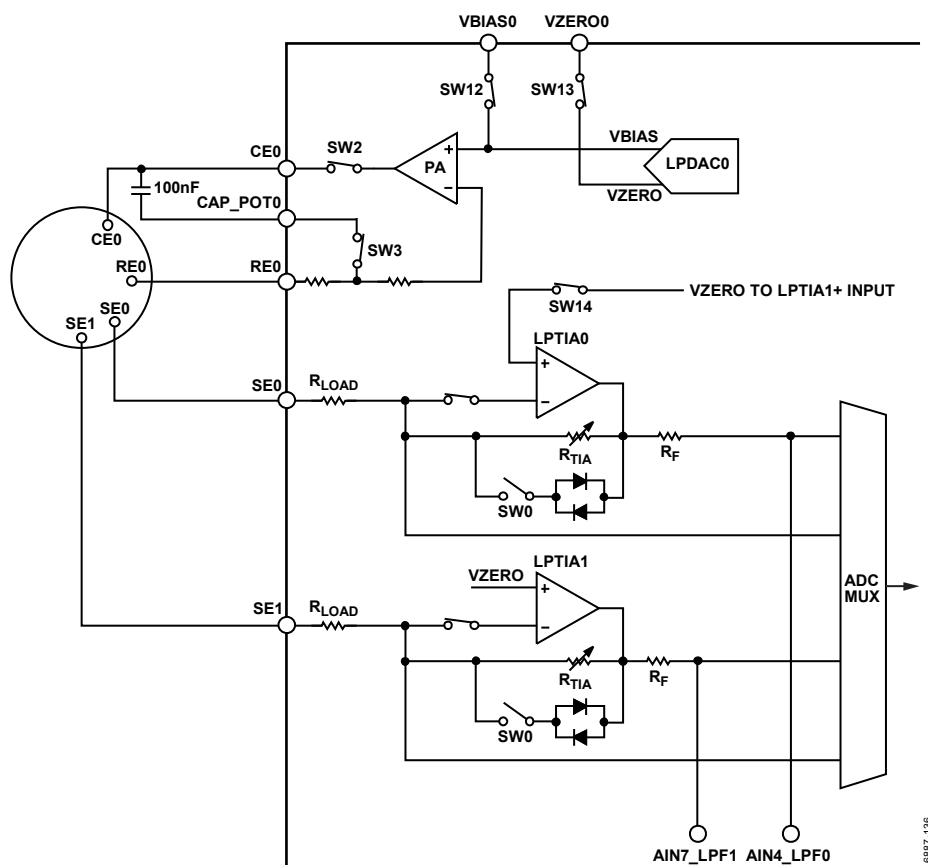


Figure 35. **ADuCM355** External RTIA Connection to the HSTIA

```
ADC Samples: H2S - CHAN0: 32419, CO - CHAN1: 32806 t@f
H2S: -2.521873 ppm t@f
CO: 0.222091 ppm t@f
```

Figure 36. **M355_ECSns_DualSensor** Output Text to UART Port



MASS ERASE A DEVICE NOT RESPONDING TO SWD COMMANDS

The SWD debug tools can only communicate with the microcontroller when the device is in active mode.

Similarly, watchdog or software resets when a debug session is starting result in the debug session ending with errors.

To recover a device that is locked like this, mass erase the user flash.

To mass erase the user flash, take the following steps:

1. Put the device into boot mode by holding the S3 button down.
2. While holding the S3 button down, press and release the reset button (S1).
3. The device is now locked in a loop in the kernel space and does not execute user code.
4. In the IAR Embedded Workbench, navigate to **Project > Download > Erase memory** (see Figure 38).
5. The window shown in Figure 39 then appears. Press **OK**.

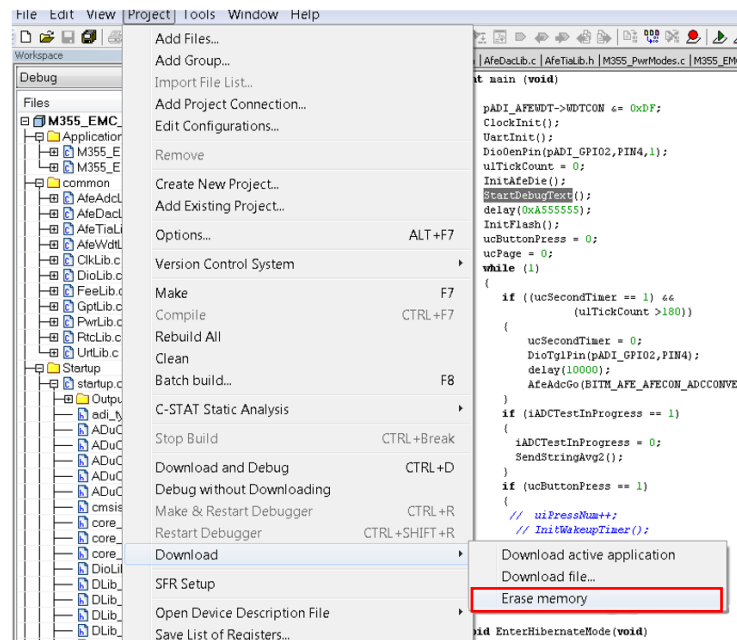


Figure 38. Embedded Workbench Erase Flash Memory Option

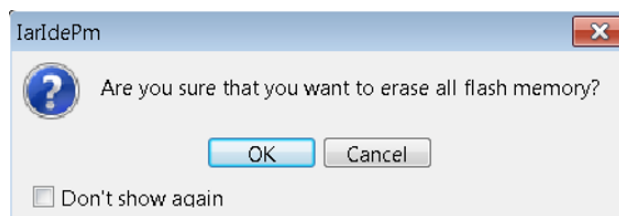


Figure 39. Erase All Flash Memory

NOTES

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

**ESD Caution**

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

Legal Terms and Conditions

By using the evaluation board discussed herein (together with any tools, components documentation or support materials, the "Evaluation Board"), you are agreeing to be bound by the terms and conditions set forth below ("Agreement") unless you have purchased the Evaluation Board, in which case the Analog Devices Standard Terms and Conditions of Sale shall govern. Do not use the Evaluation Board until you have read and agreed to the Agreement. Your use of the Evaluation Board shall signify your acceptance of the Agreement. This Agreement is made by and between you ("Customer") and Analog Devices, Inc. ("ADI"), with its principal place of business at One Technology Way, Norwood, MA 02062, USA. Subject to the terms and conditions of the Agreement, ADI hereby grants to Customer a free, limited, personal, temporary, non-exclusive, non-sublicensable, non-transferable license to use the Evaluation Board FOR EVALUATION PURPOSES ONLY. Customer understands and agrees that the Evaluation Board is provided for the sole and exclusive purpose referenced above, and agrees not to use the Evaluation Board for any other purpose. Furthermore, the license granted is expressly made subject to the following additional limitations: Customer shall not (i) rent, lease, display, sell, transfer, assign, sublicense, or distribute the Evaluation Board; and (ii) permit any Third Party to access the Evaluation Board. As used herein, the term "Third Party" includes any entity other than ADI, Customer, their employees, affiliates and in-house consultants. The Evaluation Board is NOT sold to Customer; all rights not expressly granted herein, including ownership of the Evaluation Board, are reserved by ADI. **CONFIDENTIALITY.** This Agreement and the Evaluation Board shall all be considered the confidential and proprietary information of ADI. Customer may not disclose or transfer any portion of the Evaluation Board to any other party for any reason. Upon discontinuation of use of the Evaluation Board or termination of this Agreement, Customer agrees to promptly return the Evaluation Board to ADI. **ADDITIONAL RESTRICTIONS.** Customer may not disassemble, decompile or reverse engineer chips on the Evaluation Board. Customer shall inform ADI of any occurred damages or any modifications or alterations it makes to the Evaluation Board, including but not limited to soldering or any other activity that affects the material content of the Evaluation Board. Modifications to the Evaluation Board must comply with applicable law, including but not limited to the RoHS Directive. **TERMINATION.** ADI may terminate this Agreement at any time upon giving written notice to Customer. Customer agrees to return to ADI the Evaluation Board at that time. **LIMITATION OF LIABILITY.** THE EVALUATION BOARD PROVIDED HEREUNDER IS PROVIDED "AS IS" AND ADI MAKES NO WARRANTIES OR REPRESENTATIONS OF ANY KIND WITH RESPECT TO IT. ADI SPECIFICALLY DISCLAIMS ANY REPRESENTATIONS, ENDORSEMENTS, GUARANTEES, OR WARRANTIES, EXPRESS OR IMPLIED, RELATED TO THE EVALUATION BOARD INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, TITLE, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. IN NO EVENT WILL ADI AND ITS LICENSORS BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES RESULTING FROM CUSTOMER'S POSSESSION OR USE OF THE EVALUATION BOARD, INCLUDING BUT NOT LIMITED TO LOST PROFITS, DELAY COSTS, LABOR COSTS OR LOSS OF GOODWILL. ADI'S TOTAL LIABILITY FROM ANY AND ALL CAUSES SHALL BE LIMITED TO THE AMOUNT OF ONE HUNDRED US DOLLARS (\$100.00). **EXPORT.** Customer agrees that it will not directly or indirectly export the Evaluation Board to another country, and that it will comply with all applicable United States federal laws and regulations relating to exports. **GOVERNING LAW.** This Agreement shall be governed by and construed in accordance with the substantive laws of the Commonwealth of Massachusetts (excluding conflict of law rules). Any legal action regarding this Agreement will be heard in the state or federal courts having jurisdiction in Suffolk County, Massachusetts, and Customer hereby submits to the personal jurisdiction and venue of such courts. The United Nations Convention on Contracts for the International Sale of Goods shall not apply to this Agreement and is expressly disclaimed.