Application Note

Detecting Wire Breaks in Industrial PLC Systems Using the ADS125H18 Open Wire Current Source (OWCS) Feature

Bryan Lizon

ABSTRACT

This application note provides a comprehensive analysis of the ADS125H18 open wire current source (OWCS) feature that enables the user to detect wire breaks in industrial automation systems. This application note delves into several important topics related to the OWCS feature, including how wire break detection differs for single-ended versus differential inputs, theoretical OWCS behavior, and verified performance using real-world measurements across a variety of conditions. Additionally, this application note provides a suggested OWCS algorithm with accompanying example code, as well as a corollary discussion regarding using the OWCS feature in dual, redundant industrial automation systems. Use the information in this document to design a robust, reliable analog input module using the ADS125H18 and its unique open-wire detection feature.

Table of Contents

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

Modern industrial control systems often use process-level voltage (±10V) signaling to monitor and control a factory or plant. An analog input module continuously receives these signals and converts them to digital so the control system can make quick, accurate decisions. However, random faults such as wire breaks between transmitters and analog input modules can disrupt this signaling. Figure 1-1 shows an example of such a system with a wire break occurring on the signal line of a single-ended analog input.

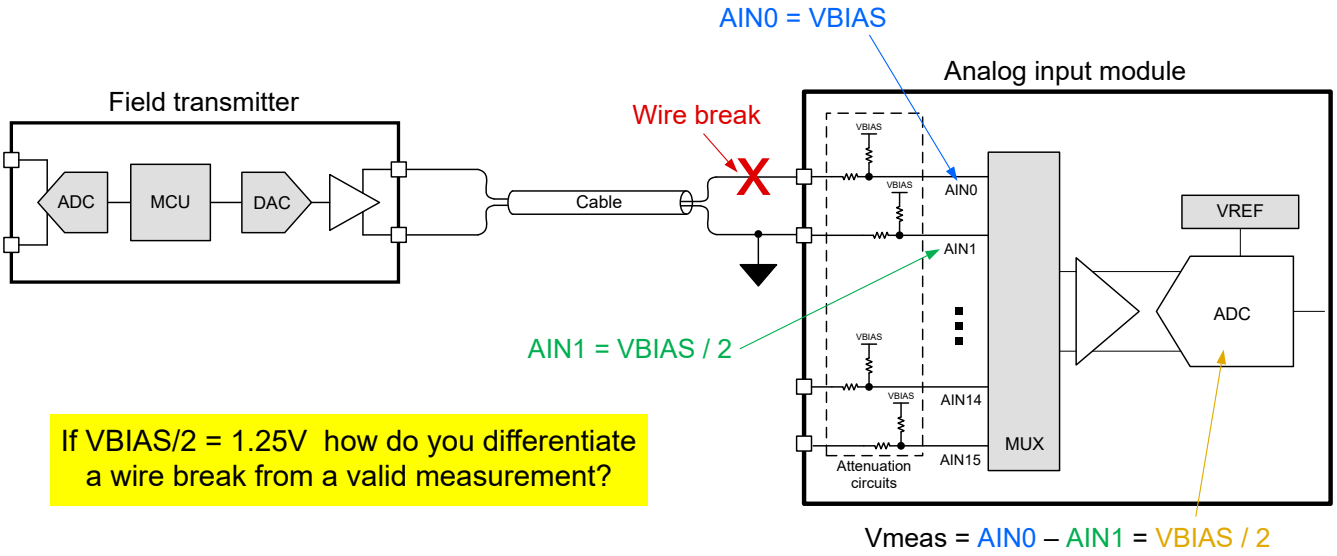

Figure 1-1. Wire Break Between a Field Transmitter and PLC Analog Input Module

Figure 1-1 shows that a wire break causes the AIN0 input to get pulled to the VBIAS voltage while the AIN1 input gets pulled to VBIAS / 2 in this example system. The ADC then measures a voltage difference of VBIAS / 2 = 1.25V that is well within the ±10V valid input range. How does the system distinguish this fault condition from a legitimate input voltage? What happens if this system was measuring the pressure in a gas pipeline that is now either unmonitored or providing a false negative? In these cases, detecting faults and failures is critical to enable the system to be put into a known, secure state and alert the operator that something has gone wrong.

To meet the critical needs of industrial systems, this application note describes the operation and verification of the open wire current source (OWCS) feature in the ADS125H18, a precision, 24-bit, 1MSPS delta-sigma (ΔΣ) analog-to-digital converter (ADC) designed for use in analog input modules. This document covers the following topics in detail:

- Understanding ADS125H18 OWCS operation
- Using OWCS with single-ended inputs signals
- Using OWCS with differential inputs signals
- Using the ADS125H18 channel sequencer to implement an OWCS algorithm

This document also includes an appendix that discusses how the OWCS behavior changes for redundant systems using two ADS125H18 devices to measure the same input signal.

2 Detailed Description

2.1 Understanding ADS125H18 OWCS Operation

The ADS125H18 includes 16 configurable analog inputs. Each analog input is comprised of a high-impedance voltage divider with integrated precision matched resistors that scale down the input voltage into the ADC input range. The OWCS feature operates by injecting a small current into the node between the ADS125H18 resistor divider and input multiplexer. The ADS125H18 includes a buffer after the multiplexer such that the current only flows through the resistor divider and the source impedance, R_{SOURCE}. Comparing data with OWCS disabled and OWCS enabled allows the user to calculate a threshold value that correlates to R_{SOURCE}. A large change in impedance can indicate an open wire. Figure 2-1 shows a simplified ADS125H18 block diagram measuring a single-ended input with the OWCS path highlighted in red:

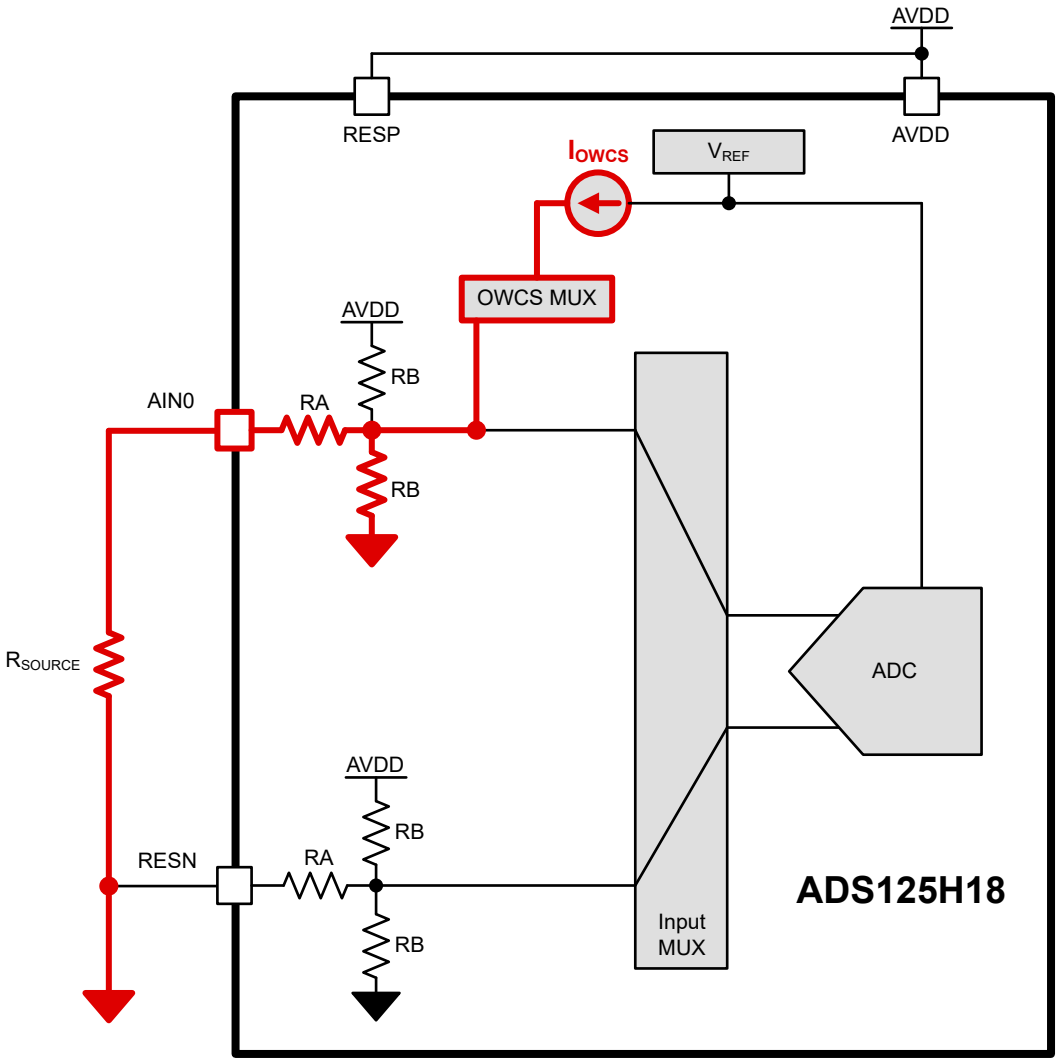

Figure 2-1. Using the ADS125H18 OWCS Feature with a Single-Ended Input

The following list provides several key points regarding open wire detection using the OWCS:

- Predicting a break requires calculating the difference (delta) between a baseline measurement (OWCS off) and a detection measurement (OWCS on). Equation 1 shows the general form of the OWCS delta calculation:

$$\mathrm{OWCS_{Delta}} = \frac{\left(\mathrm{V_{OWCS\_on}} - \mathrm{V_{OWCS\_off}}\right)}{\mathrm{V_{REF}}} \quad (1)$$

- The OWCS detection scheme assumes that the input voltage and source impedance do not change between the OWCS-disabled and OWCS-enabled measurements. It is recommended to measure these signals back-to-back and as quickly as possible to avoid detection errors.
- Detecting an open wire for a differential input requires two separate single-ended measurements: one baseline and one detection measurement for the positive input channel, and one baseline and one detection measurement for the negative input channel. However, performing the detection routine on a differential input should theoretically yield the same result between the positive and negative inputs channels
- The open wire detection deltas differ between:
  - a single-ended measurement and a differential measurement
  - different versions of the ADS125H18 (V12, V20, or V40)

Table 2-1 lists important OWCS parameters for the different ADS125H18 versions:

Table 2-1. Important OWCS Parameters

ADS125H18 Version	Attenuation factor	RA (MΩ)	RB (MΩ)	I_{OWCS_NOM} (µA/V)
V12	7	1.125	0.375	1
V20	10	1.125	0.250	2
V40	19	1.125	0.125	4

Figure 2-1 shows that the ADC voltage reference, V_{REF}, powers the OWCS current source. Table 2-1 reflects this behavior as well because the nominal OWCS current units are µA/V such that the OWCS current is ratiometric to V_{REF}. Equation 2 expresses this relationship with respect to the OWCS output current, I_{OWCS}:

$$I_{OWCS} = I_{OWCS\_NOM} \times V_{REF} \quad (2)$$

Ultimately, the reference voltage value does not affect the detection threshold. This behavior exists whether the user selects the internal voltage reference or applies an external reference voltage. Refer to the ADS125H18 datasheet for more information regarding the OWCS behavior as well as the device register map.

2.2 Using OWCS with Single-Ended Input Signals

The following sections describe the OWCS behavior when the user applies a single-ended input to the ADS125H18:

- Analyzing OWCS behavior for single-ended input signals
- Verifying OWCS operation with a single-ended input using the ADS125H18EVM

2.2.1 Analyzing OWCS Behavior for Single-Ended Input Signals

This section analyzes the behavior of the OWCS when a single-ended input with source impedance R_{SOURCE} is applied to the ADS125H18. Figure 2-2 shows the circuit used to analyze a single-ended input with the OWCS disabled and a single-ended R_{SOURCE} impedance:

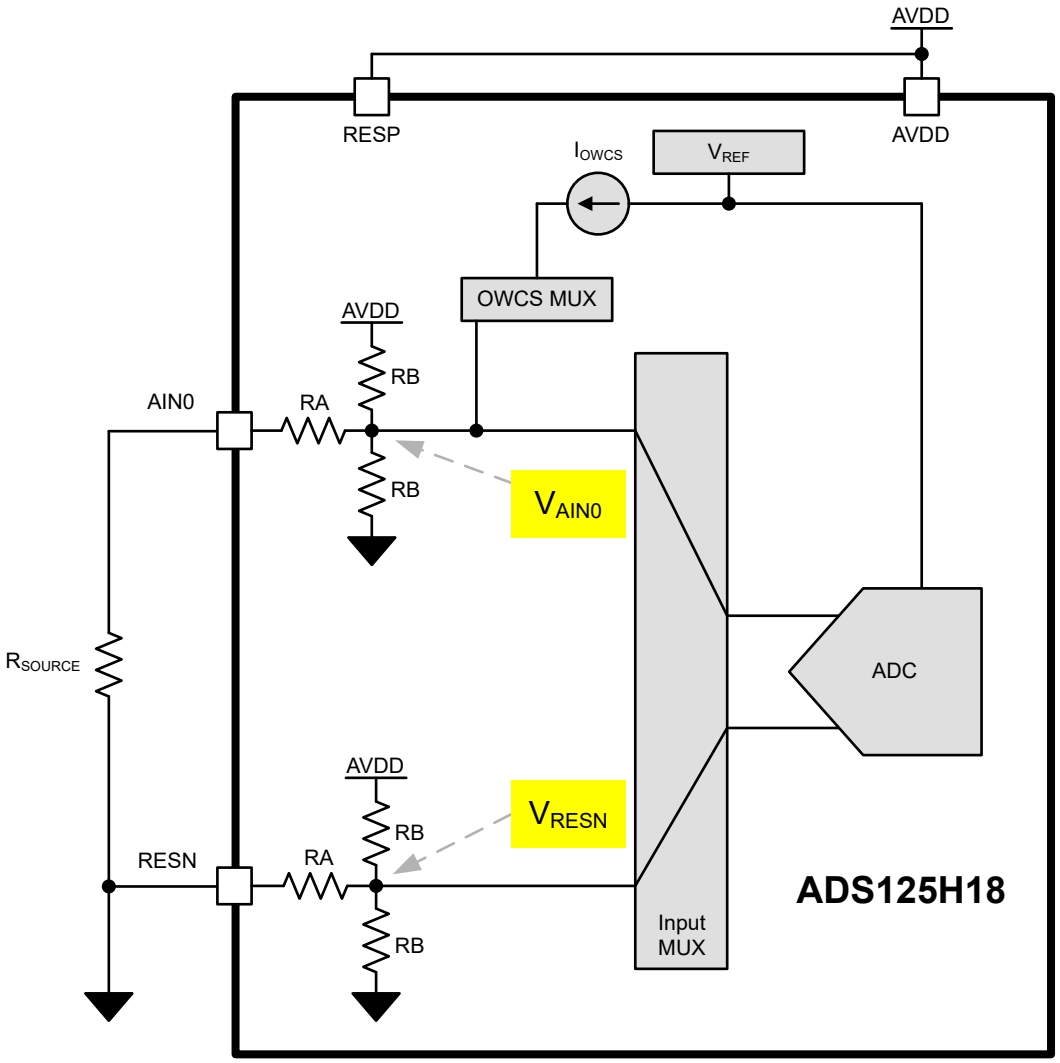

Figure 2-2. Analyzing OWCS Behavior for Single-Ended Input Signals

The ADC measures the single-ended voltage difference between nodes V_{AIN0} and V_{RESN} in Figure 2-2 such that Equation 1 can be expanded to Equation 3:

$$\mathrm{OWCS_{Delta\_SE}} = \frac{\left(\left[\mathrm{V(AIN0)_{OWCS\_on}} - \mathrm{V(RESN)_{OWCS\_on}}\right] - \left[\mathrm{V(AIN0)_{OWCS\_off}} - \mathrm{V(RESN)_{OWCS\_off}}\right]\right)}{\mathrm{V_{REF}}} \quad (3)$$

Use superposition to analyze the voltage contribution from each source in the circuit by shorting voltage sources or opening current sources. For example, Figure 2-3 shows that calculating the total voltage at AIN0 with the OWCS enabled, $V(AIN0)_{OWCS\_SE}$, requires three different superposition configurations:

- $V(AIN0)_{AIN0\_AVDD\_SE}$ = AVDD (AIN0) connected, all other sources removed (Figure 2-3, left)
- $V(AIN0)_{RESN\_AVDD\_SE}$ = AVDD (RESN) connected, all other sources removed (Figure 2-3, middle)
- $V(AIN0)_{AIN0\_OWCS\_SE}$ = I_{OWCS} connected, all other sources removed (Figure 2-3, right)

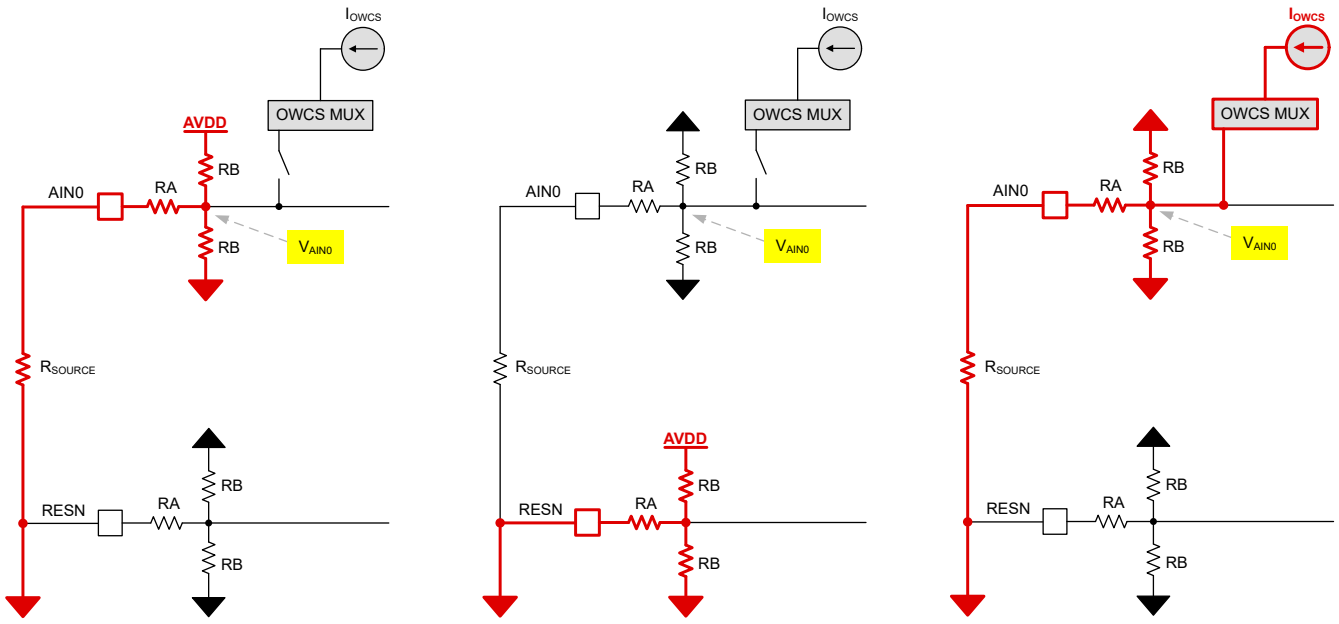

Figure 2-3. Example Superposition Networks for a Single-Ended Input

Table 2-2 describes each nodal voltage after applying this same analysis to every term in Equation 3:

Table 2-2. Nodal Voltages for Single-Ended Inputs

Voltage term	Nodal voltages
$V(AIN0)_{OWCS\_on}$	= $V(AIN0)_{AIN0\_AVDD\_SE}$ + $V(AIN0)_{RESN\_AVDD\_SE}$ + $V(AIN0)_{AIN0\_OWCS\_SE}$ where • – $V(AIN0)_{AVDD\_AIN0\_SE}$ = AVDD (AIN0) connected, all other sources removed – $V(AIN0)_{AVDD\_RESN\_SE}$ = AVDD (RESN) connected, all other sources removed – $V(AIN0)_{OWCS\_SE}$ = I_{OWCS} connected, all other sources removed
$V(RESN)_{OWCS\_on}$	= $V(RESN)_{AIN0\_AVDD\_SE}$ + $V(RESN)_{RESN\_AVDD\_SE}$ + $V(RESN)_{AIN0\_OWCS\_SE}$ where • – $V(RESN)_{AVDD\_AIN0\_SE}$ = AVDD (AIN0) connected, all other sources removed – $V(RESN)_{AVDD\_RESN\_SE}$ = AVDD (RESN) connected, all other sources removed – $V(RESN)_{OWCS\_SE}$ = I_{OWCS} connected, all other sources removed
$V(AIN0)_{OWCS\_off}$	= $V(AIN0)_{AIN0\_AVDD\_SE}$ + $V(AIN0)_{RESN\_AVDD\_SE}$ where • – $V(AIN0)_{AVDD\_AIN0\_SE}$ = AVDD (AIN0) connected, all other sources removed – $V(AIN0)_{AVDD\_RESN\_SE}$ = AVDD (RESN) connected, all other sources removed
$V(RESN)_{OWCS\_off}$	= $V(RESN)_{AIN0\_AVDD\_SE}$ + $V(RESN)_{RESN\_AVDD\_SE}$ where • – $V(RESN)_{AVDD\_AIN0\_SE}$ = AVDD (AIN0) connected, all other sources removed – $V(RESN)_{AVDD\_RESN\_SE}$ = AVDD (RESN) connected, all other sources removed

Importantly, Table 2-2 reveals that virtually all nodal voltages are identical such that Equation 3 can be reduced to Equation 4:

$$OWCS_{Delta\_SE} = \frac{V(AIN0)_{OWCS\_SE} - V(RESN)_{OWCS\_SE}}{V_{REF}} \quad (4)$$

Figure 2-4 shows the superposition configurations for the terms in Equation 4:

- $V(AIN0)_{OWCS\_SE}$ = I_{OWCS} connected, all other sources removed (Figure 2-4, left)

- $V(RESN)_{OWCS\_SE}$ = I_{OWCS} connected, all other sources removed (Figure 2-4, right)

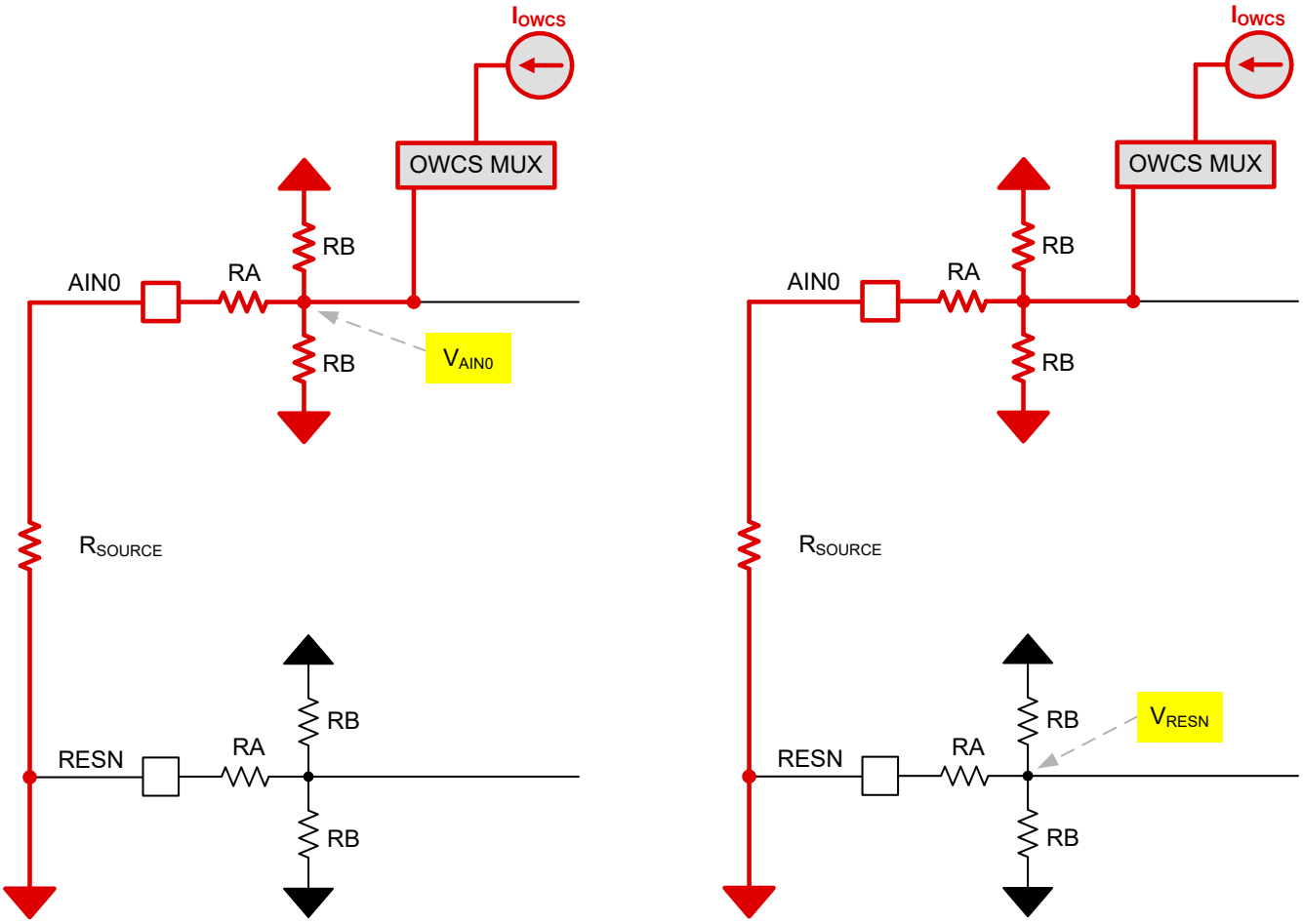

Figure 2-4. Superposition Networks for $V(AIN0)_{OWCS\_SE}$ and $V(RESN)_{OWCS\_SE}$

Figure 2-4 also shows that the voltage at RESN with OWCS enabled, $V(RESN)_{OWCS\_SE}$, is 0V because the RESN pin is grounded. Therefore, Equation 4 reduces to Equation 5, which is represented by Figure 2-4 (left):

$$OWCS_{Delta\_SE} = \frac{V(AIN0)_{OWCS\_SE}}{V_{REF}} \quad (5)$$

Equation 6 rewrites $V(AIN0)_{OWCS\_SE}$ in Figure 2-4 (left) in terms of resistances RA, RB, and R_{SOURCE} as well as current I_{OWCS}:

$$V(AIN0)_{OWCS\_SE} = I_{OWCS} \times (RB \,||\, RB \,||\, [RA + R_{SOURCE}]) \quad (6)$$
$$V(AIN0)_{OWCS\_SE} = I_{OWCS} \times \left(\frac{RB}{2} \,||\, [RA + R_{SOURCE}]\right)$$

Recall from Equation 2 that the actual OWCS current, I_{OWCS}, is relative to the reference voltage, VREF. Therefore, the I_{OWCS} term in Equation 6 can be rewritten in terms of the nominal OWCS current, I_{OWCS_NOM}, and simplified as shown in Equation 7:

$$OWCS_{Delta\_SE} = \frac{(I_{OWCS\_NOM} \times V_{REF}) \times \left(\frac{RB}{2} \,||\, [RA + R_{SOURCE}]\right)}{V_{REF}} \quad (7)$$
$$OWCS_{Delta\_SE} = I_{OWCS\_NOM} \times \left(\frac{RB}{2} \,||\, [RA + R_{SOURCE}]\right)$$

Equation 7 can also be rewritten to predict the single-ended R_{SOURCE_OWCS} value from the measured $OWCS_{Delta}$ threshold, as shown in Equation 8:

$$R_{SOURCE\_OWCS\_SE} = \frac{OWCS_{Delta\_SE} \times (RB + 2 \times RA) - RA \times RB \times I_{OWCS\_NOM}}{RB \times I_{OWCS\_NOM} - 2 \times OWCS_{Delta\_SE}} \quad (8)$$

Table 2-1 includes RB and I_{OWCS_NOM} values for all ADS125H18 versions. Figure 2-5 applies all possible parameters from Table 2-1 to Equation 7 and sweeps R_{SOURCE} from 1kΩ to 1GΩ to generate the ideal $OWCS_{Delta\_SE}$ curves:

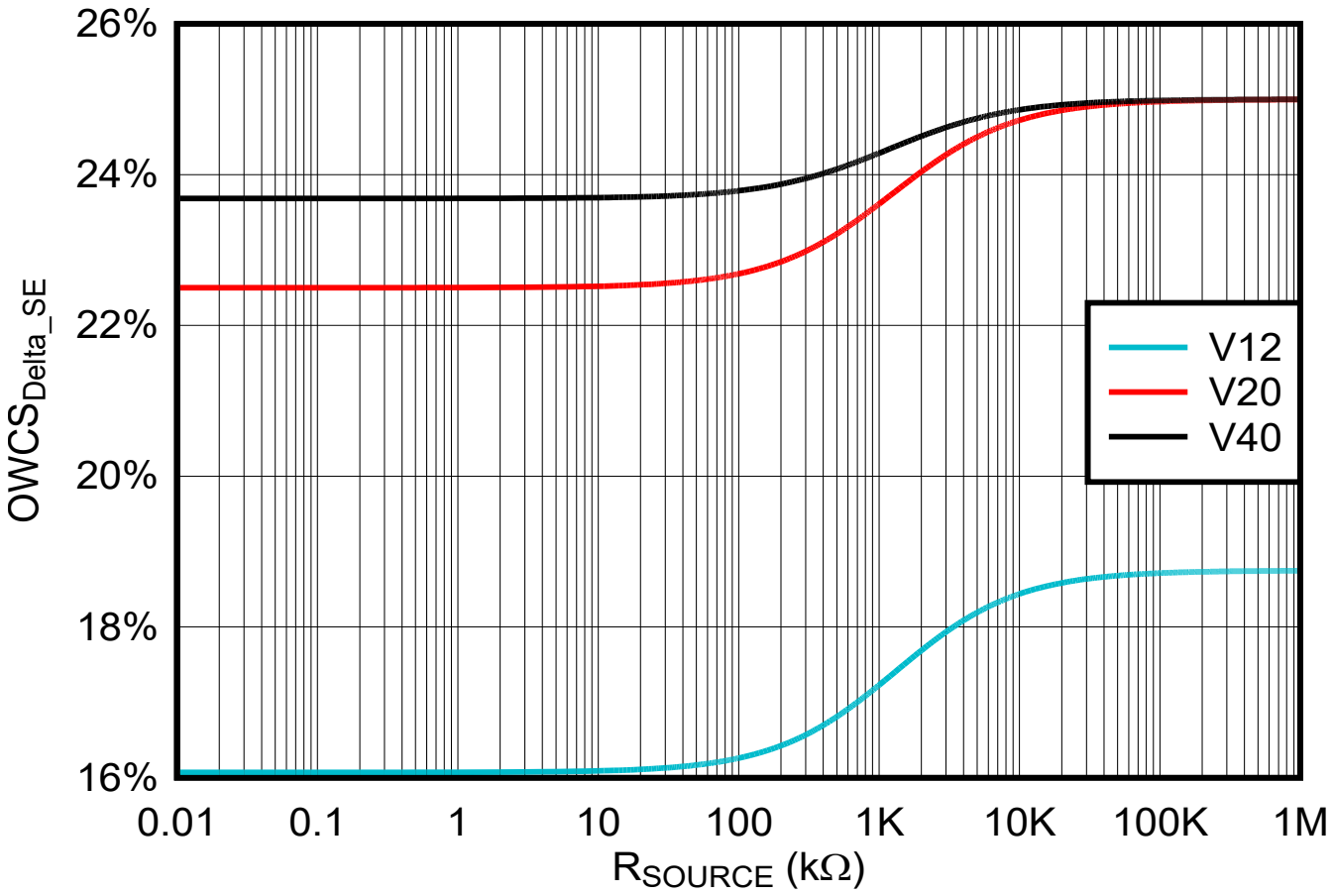

Figure 2-5. $OWCS_{Delta}$ Behavior for all ADS125H18 Variants Measuring a Single-Ended Input (R_{SOURCE} = 1kΩ to 1GΩ)

Figure 2-5 reveals two important takeaways for single-ended OWCS measurements:

1. The significant change in $OWCS_{Delta\_SE}$ occurs in the range from 100kΩ to 10MΩ. This range indicates where a change in input impedance is most detectable
2. Each $OWCS_{Delta\_SE}$ curve has a theoretical minimum and maximum at R_{SOURCE} = 0Ω and R_{SOURCE} = ∞, respectively. The difference between the minimum and maximum decreases as the attenuation factor increases, resulting in a narrower margin to indicate an OWCS fault. Table 2-3 summarizes these results.

Table 2-3. Minimum and Maximum $OWCS_{Delta}$ Values for a Single-Ended Input

ADS125H18 version	Attenuation factor	$OWCS_{Delta\_SE}$ Min	$OWCS_{Delta\_SE}$ Max	Difference
V12	7	16.07%	18.75%	2.68%
V20	10	22.50%	25.00%	2.5%
V40	19	23.68%	25.00%	1.32%

2.2.2 Verifying OWCS Operation with a Single-Ended Input Using the ADS125H18EVM

Testing was performed to verify OWCS operation using the ADS125H18 evaluation module (EVM). The ADS125H18 EVM is a platform for evaluating the functionality and performance of the ADS125H18. The

ADS125H18EVM-PDK eases the evaluation of the device with hardware, software, and computer connectivity through the universal serial bus (USB) interface. Figure 2-6 shows the ADS125H18 EVM.

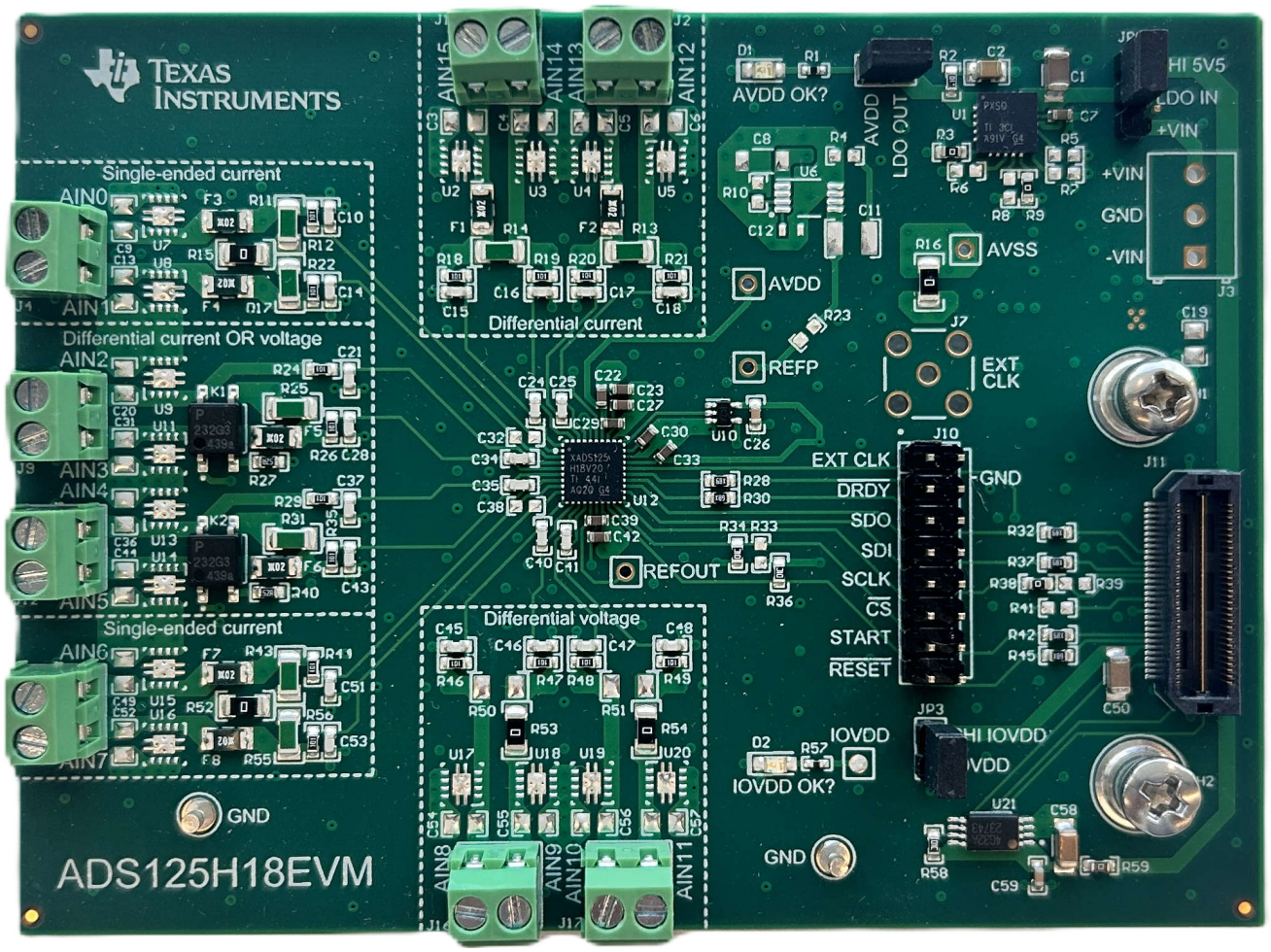

Figure 2-6. ADS125H18EVM Board Image

The following ADC and EVM conditions were used to perform the OWCS verification:

- ADS12H18V20
- Internal 25.6MHz oscillator
- Speed mode 3 (f_{MOD} = 12.8MHz)
- Internal 2.5V reference voltage
- Single-shot sequence mode
- Two enabled steps (same input channel):
  - Step 1 = OWCS disabled
  - Step 2 = OWCS enabled
- Three output data rates (ODR):
  - 1.6kSPS (sinc4+sinc1 filter, OSR = 8000)
  - 12.5kSPS (sinc4 filter, OSR = 1024)
  - 50kSPS (sinc4+sinc1 filter, OSR = 256)
- 512 conversions per OWCS step (to capture possible settling behavior)
- No external passive (R or C) components

2.2.2.1 Verifying OWCS Operation with a Single-Ended Input

Figure 2-7 shows the simple OWCS verification setup using a single-ended input on the ADS125H18 EVM. No passive components were installed between the terminal block and the ADC input pins. Various 0.1% R_{SOURCE}

resistors were connected between the EVM terminal blocks and ground to verify performance across a wide range of input impedances. This configuration results in a 0V input signal to the ADC to simplify the analysis.

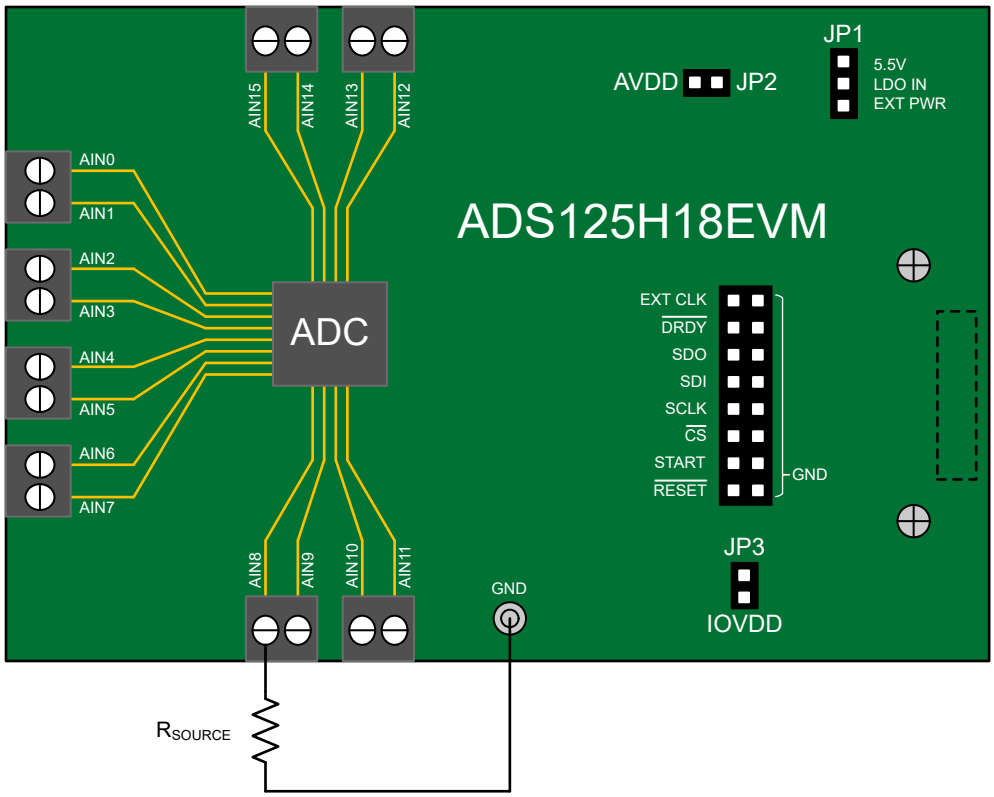

Figure 2-7. OWCS Verification Setup for a Single-Ended Input

The input was measured by taking 512 conversions with the OWCS disabled followed by 512 conversions with the OWCS enabled. The $OWCS_{Delta\_SE}$ value was then calculated using Equation 1 and converted to a predicted R_{SOURCE_OWCS} value by rearranging the terms in Equation 7. Table 2-4 shows the measured, averaged results across 3 different ODRs and 6 different R_{SOURCE} values:

Table 2-4. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} Values Using Different ODRs and R_{SOURCE} Inputs (Single-Ended Input)

R_{SOURCE} (kΩ)	ODR = 1.6kSPS		ODR = 12.5kSPS		ODR = 50kSPS	
	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)
25.5	22.57%	35.95	22.57%	35.72	22.57%	35.61
99.9	22.70%	110.23	22.70%	110.29	22.70%	108.94
499.9	23.22%	508.51	23.22%	509.42	23.22%	507.46
1000.0	23.62%	1011.33	23.62%	1014.42	23.62%	1010.50
1999.9	24.04%	2015.60	24.04%	2016.88	24.04%	2013.53
∞ (Open)	25.01%	>1G	25.01%	>1G	25.00%	>1G

Table 2-4 reveals that the OWCS correctly determined the applied single-ended source impedance across different ODRs for virtually all R_{SOURCE} values. However, Table 2-4 also shows an approximate 10kΩ offset between the predicted R_{SOURCE_OWCS} values and the actual R_{SOURCE} values. Fortunately, this seemingly large error in kΩ translates to a very small 0.05% average input code variation that can be attributed to ADS125H18 resistor divider matching. The end result is that the OWCS fault detection behaves as intended: this feature accurately identified an open circuit as well as significant changes in the source impedance.

2.2.2.2 Using a Fewer Number of Conversions for OWCS with a Single-Ended Input

Table 2-4 shows an average $OWCS_{Delta\_SE}$ value calculated using 512 samples with OWCS disabled and 512 samples with OWCS enabled. Section 2.2.2.1 states that 512 conversions per sequence step were used to capture any possible settling behavior. However, taking a total of 1024 conversions to detect a single wire break might require an impractical amount of time in many industrial systems. For example, 1024 conversions at 1.6kSPS takes approximately 640ms. Analyzing the sampled data points helps indicate how many conversions are actually necessary for accurate fault detection.

Figure 2-8 plots the $OWCS_{Delta\_SE}$ values point-by-point instead of a single value averaged over 512 conversions as listed in Table 2-4. Figure 2-8 uses R_{SOURCE} = 500kΩ and 3 different ODR values: 1.6kSPS, 12.5kSPS, and 50kSPS

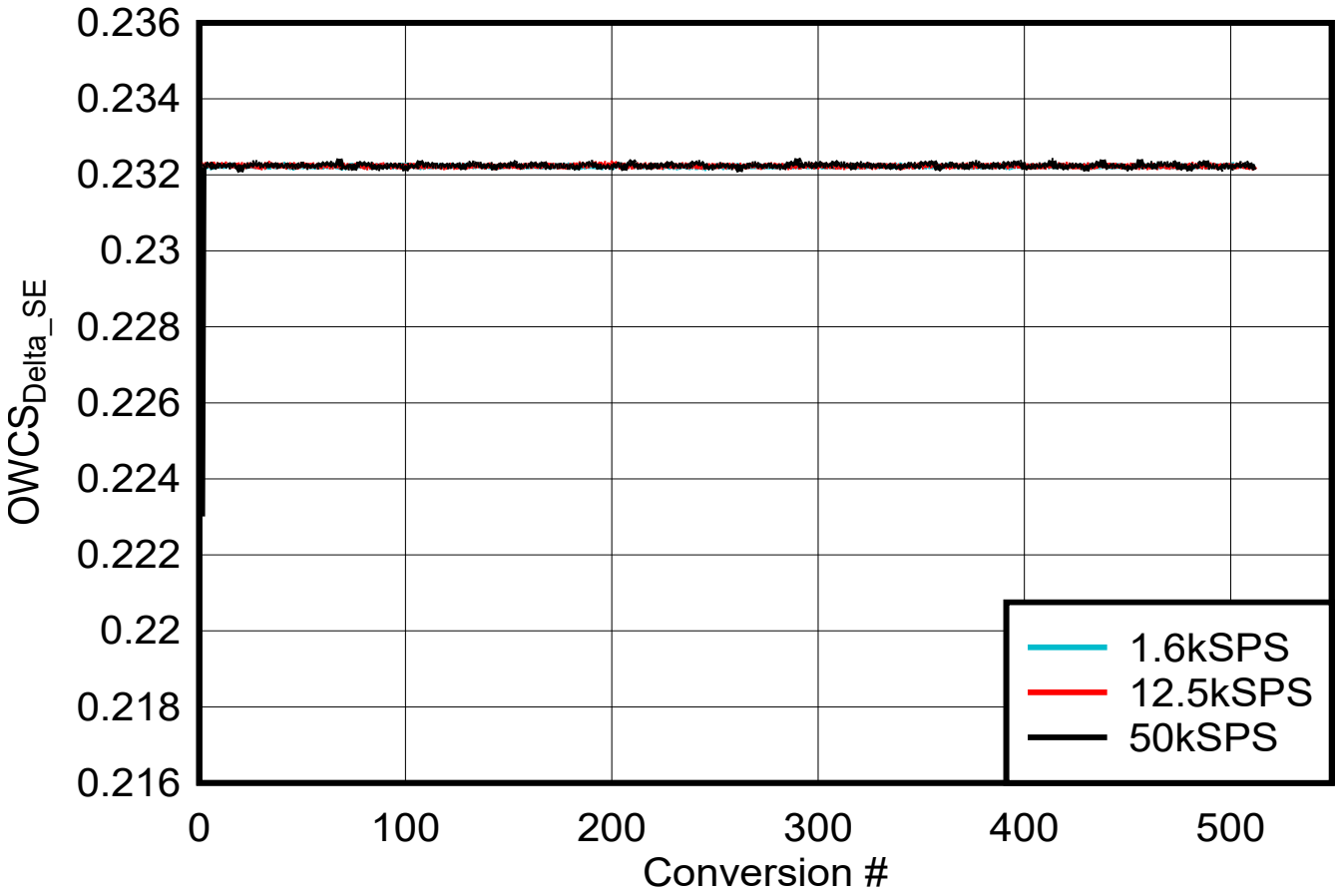

Figure 2-8. Measured $OWCS_{Delta\_SE}$ Values Plotted Point-by-Point to Show Settling Behavior (Single-Ended Input, R_{SOURCE} = 500kΩ)

Figure 2-8 reveals that some settling behavior exists in the data, but otherwise very little useful information due to the unsettled outlier. Breaking this data into unsettled and settled components helps clarify why fault detection errors might occur. Also, note that this behavior is shown at R_{SOURCE} = 500kΩ but exists across all values of R_{SOURCE}.

2.2.2.2.1 Unsettled Data for Single-Ended Inputs

Table 2-5 includes the measured $OWCS_{Delta\_SE}$ and predicted R_{SOURCE_OWCS} values calculated from only the first conversion data at 3 different ODRs.

Table 2-5. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} Value Using Only First Conversion Data Point (Single-Ended Input, R_{SOURCE} = 500kΩ)

ODR	First conversion only	
	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)
1.6kSPS	23.22%	509.01

Table 2-5. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} Value Using Only First Conversion Data Point (Single-Ended Input, R_{SOURCE} = 500kΩ) (continued)

ODR	First conversion only	
	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)
12.5kSPS	23.23%	513.03
50kSPS	21.73%	0 (short)

Importantly, Table 2-5 shows that the unsettled data at ODR = 50kSPS indicates a short instead of R_{SOURCE} = 500kΩ. Using a single conversion result to determine a wire break under these conditions could provide a false negative to the user. Comparatively, the other two ODR settings correctly predict the R_{SOURCE} value. These combined results indicate that faster data rates can result in incorrect OWCS calculations. The user must account for this possibility when they choose both the OWCS data rate and the number of conversions for the system.

2.2.2.2.2 Settled Data for Single-Ended Inputs

Figure 2-9 shows the same data as Figure 2-8 but with the first conversion removed for all ODRs. In other words, Figure 2-9 plots conversions 2 through 512 point-by-point instead of a single value averaged over many conversions.

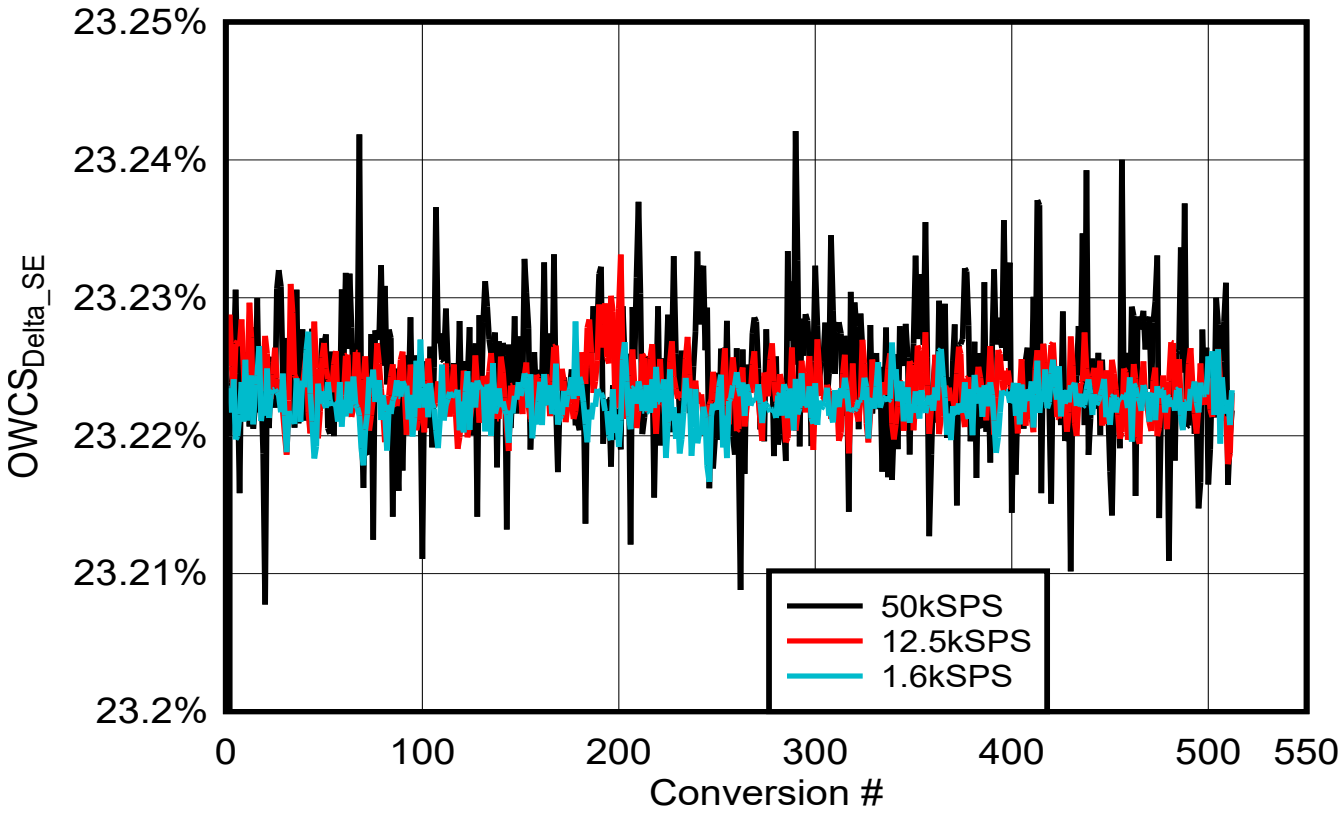

Figure 2-9. Measured $OWCS_{Delta\_SE}$ Values – Excluding First Conversion – Plotted Point-by-Point Across ODR (Single-Ended Input, R_{SOURCE} = 500kΩ)

Figure 2-9 shows the desired behavior for OWCS calculations. The $OWCS_{Delta\_SE}$ values generally have a Gaussian (broadband) noise distribution. Additionally, the noise distribution gets tighter as the ODR decreases because the ADC digital filter bandwidth also decreases. Both of these factors indicate typical ADC data dominated by thermal noise – with no settling effects – that can correctly predict R_{SOURCE_OWCS} for any conversion.

Table 2-6 analyzes the distribution of $OWCS_{Delta\_SE}$ values from Figure 2-9 to determine the minimum and maximum variation in the predicted R_{SOURCE_OWCS}.

Table 2-6. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} Variance (Single-Ended Input, R_{SOURCE} = 500kΩ)**

ODR	Minimum**		Maximum**		Difference (Max – Min) (kΩ)
	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)	
1.6kSPS	23.22%	502.80	23.23%	514.20	11.40
12.5kSPS	23.22%	503.83	23.23%	519.38	15.55
50kSPS	23.21%	494.00	23.24%	528.70	34.70

**excluding the first sample

Table 2-6 shows that the difference between the minimum and maximum is smaller at the lowest ODR (1.6kSPS) and increases as the ODR increases. However, the minimum and maximum values still correctly approximate the 500kΩ R_{SOURCE} impedance in all cases. Therefore, it is possible to conclude that using significantly fewer conversions (<<512) under these conditions could have enabled faster wire break detection without sacrificing reliability.

2.2.2.3 How Input Capacitance Affects OWCS Performance for Single-Ended Inputs

Section 2.2.2 describes the OWCS performance using the standard ADS125H18 EVM with no passive components installed between the terminal block and the ADC input pins. However, many industrial systems include capacitors on the inputs for noise rejection or other purposes. Figure 2-10 shows an example of a system with a differential capacitor in red installed between the AIN8 and AIN9 analog inputs as well as common-mode (single-ended) capacitors in blue installed between each input and ground. The ADS125H18 EVM includes footprints for these components that are not populated by default.

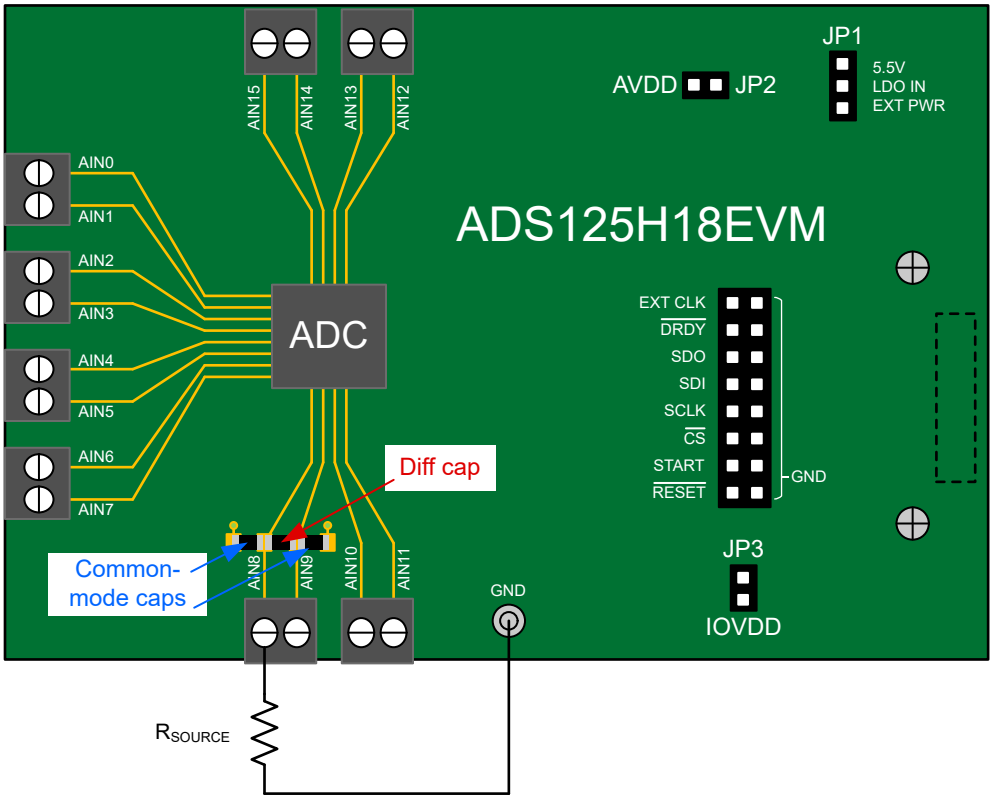

Figure 2-10. Measuring a Single-Ended Input on the ADS125H18EVM with Input Capacitance

Adding capacitance on the analog inputs can help reduce noise at the expense of increased settling time due to capacitor charging. This behavior affects the measurement when the current sources (OWCS) switch on because the capacitor cannot charge up to the required voltage instantaneously. This effect is also measurable when the OWCS turn off because the capacitor cannot discharge instantaneously.

Multiple capacitors were installed on the ADS125H18 EVM to measure how input capacitance affects the OWCS detection behavior. Otherwise, the same settings were used as listed at the beginning of Section 2.2.2. The input was then measured by taking 512 conversions with the OWCS disabled followed by 512 conversions with the OWCS enabled. Next, the $OWCS_{Delta\_SE}$ value was calculated using Equation 1 and converted to a predicted R_{SOURCE_OWCS} value by rearranging the terms in Equation 7. Table 2-7 shows the measured, averaged results when R_{SOURCE} = 800kΩ across 3 different ODRs and 4 different input capacitance values:

Table 2-7. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} for Different ODRs and Input Capacitance (Single-Ended Input, R_{SOURCE} = 800kΩ)

Input capacitance (nF)	ODR = 1.6kSPS		ODR = 12.5kSPS		ODR = 50kSPS	
	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)
1	23.46%	781.15	23.45%	769.40	23.41%	716.45
10	23.42%	731.64	23.33%	616.85	23.19%	480.98
100	23.32%	614.03	23.11%	406.91	22.84%	196.04
1000	23.06%	359.71	22.65%	80.54	22.53%	13.18

Table 2-7 reveals that the $OWCS_{Delta\_SE}$ values vary greatly across data rate and input capacitance. For example, the predicted R_{SOURCE_OWCS} value when ODR = 1.6kSPS and input capacitance = 1nF is 781.15kΩ. This result is approximately equal to the ideal 800kΩ R_{SOURCE} value. However, the predicted R_{SOURCE_OWCS} value when ODR = 50kSPS and input capacitance = 1000nF is only 13.18kΩ! Reviewing the data point-by-point helps illuminate why such a large discrepancy exists across the different parameters.

Figure 2-11 plots the $OWCS_{Delta\_SE}$ values point-by-point for R_{SOURCE} = 800kΩ and ODR = 12.5kSPS for 4 different input capacitance values: 1nF, 10nF, 100nF, and 1000nF.

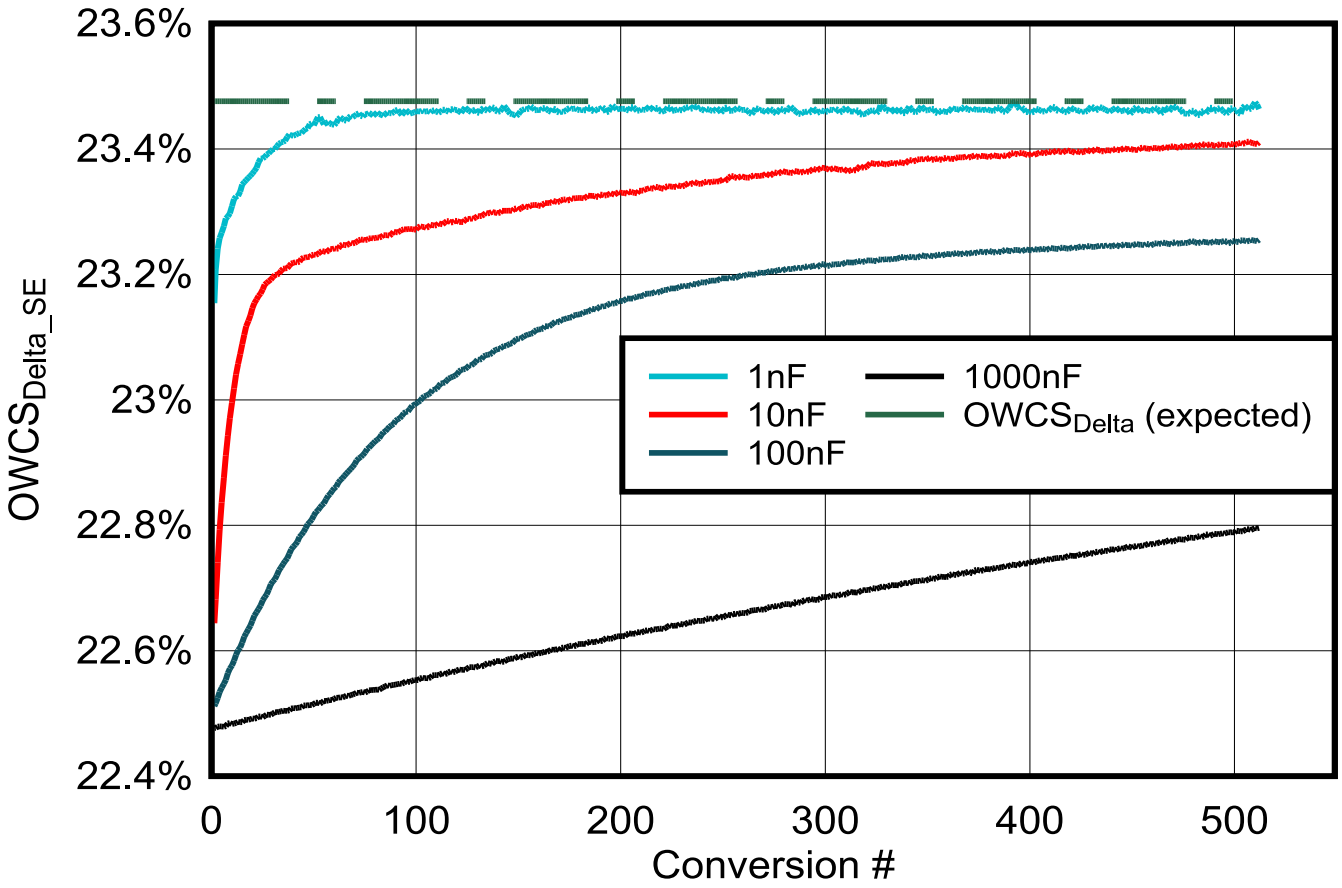

Figure 2-11. $OWCS_{Delta}$ Behavior Across Different Input Capacitance Values (Single-Ended Input, R_{SOURCE} = 800kΩ, ODR = 12.5kSPS)

Figure 2-11 reveals the significant effect input capacitance has on the OWCS behavior as well as why there is such extreme variation in the results listed in Table 2-7:

- The 1-nF plot reaches the expected $OWCS_{Delta\_SE}$ value of 23.48% after approximately 75 conversions
- The other three (3) plots never reach the expected $OWCS_{Delta\_SE}$ value
- The other three (3) plots never reach a settled state even after 512 conversions

These conclusions make sense because the capacitor time constant increases as the input capacitance increases, resulting in longer settling times.

Changing the ODR with input capacitors installed also affects the OWCS performance as mentioned previously. Figure 2-12 plots the $OWCS_{Delta\_SE}$ values for R_{SOURCE} = 800kΩ and input capacitance = 10nF point-by-point for 3 different ODRs: 1.6kSPS, 12.5kSPS, and 50kSPS. The plot x-axis units are in milliseconds to clearly indicate the charging behavior captured by the ADC. Also, all plots start at the same point and overlay each other.

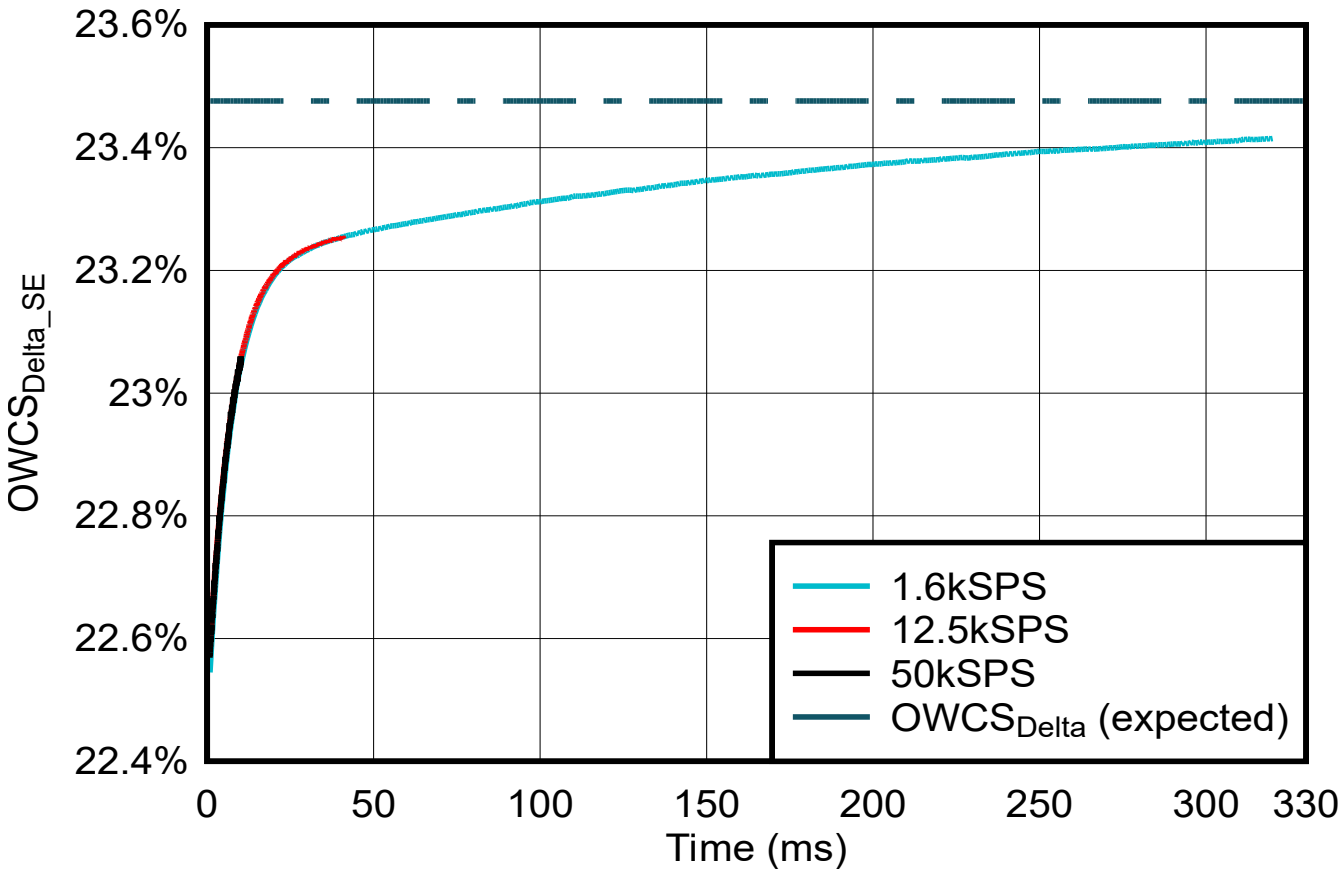

Figure 2-12. $OWCS_{Delta}$ Behavior Across Different ODR Values (Single-Ended Input, R_{SOURCE} = 800kΩ, Input Capacitance = 10nF)

The results in Figure 2-12 make sense because the ADC sampling time decreases as ODR increases, even though the capacitor time constant remains unchanged. Therefore, the ADC captures more of the capacitor charging behavior and less of the steady-state behavior as the data rate increases, leading to less stable measurements.

The input capacitance does help reduce noise as stated previously. Figure 2-13 plots the measured ADC code point-by-point with OWCS disabled for R_{SOURCE} = 800kΩ and ODR = 12.5kSPS for 4 different values of input capacitance: 1nF, 10nF, 100nF, and 1000nF.

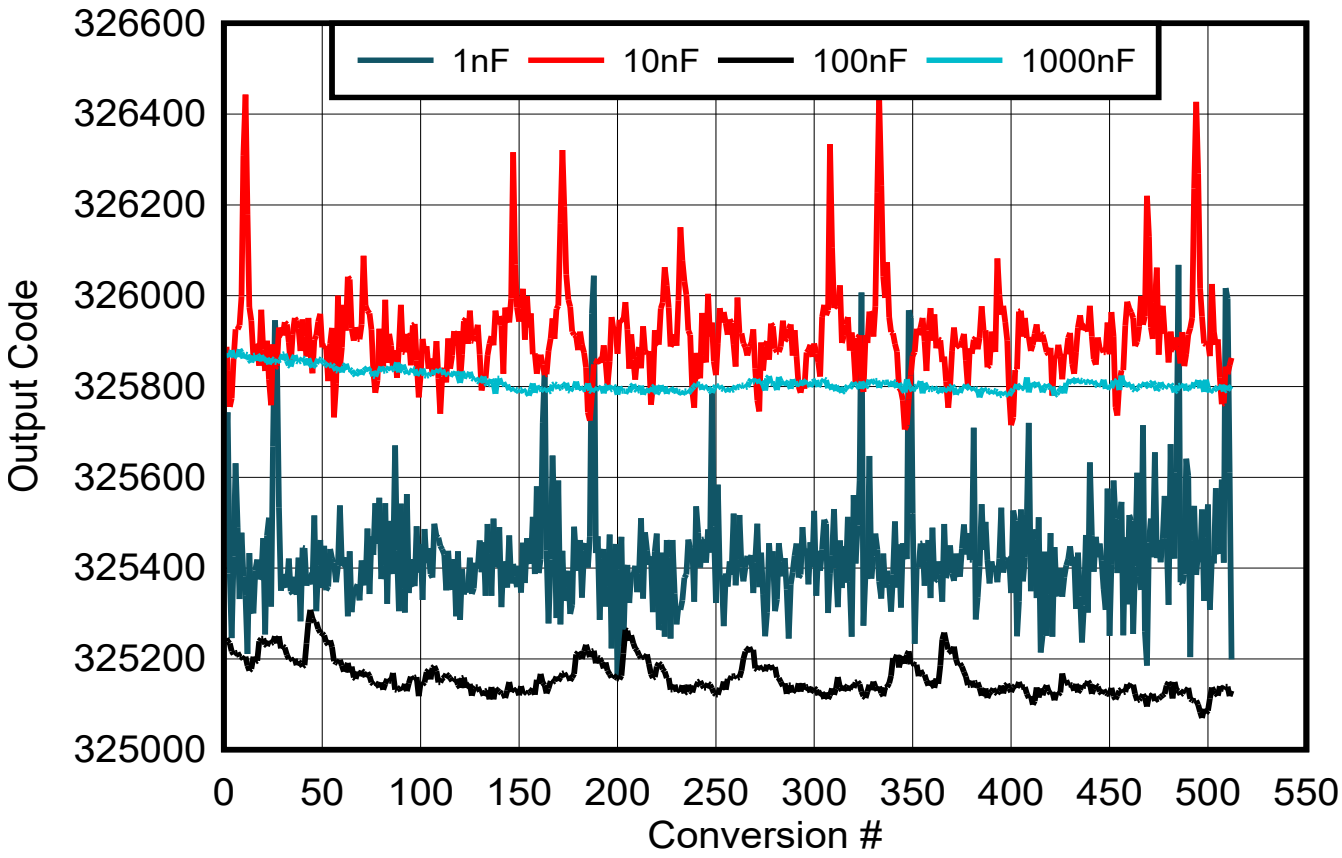

Figure 2-13. ADC Code Behavior Across Different Input Capacitance Values (Single-Ended Input, OWCS disabled, R_{SOURCE} = 800kΩ, ODR = 12.5kSPS)

Figure 2-13 qualitatively shows that the noise spread decreases as the input capacitance increases. Specifically, the RMS noise when input capacitance = 1nF is 160.38 codes, while the RMS noise when the input capacitance = 1000nF is only 14.09 codes. Designers must take care to balance the noise reduction benefits of an input capacitor with the challenges introduced by capacitor settling.

2.2.2.4 Using the ADS125H18 Programmable Delay to Mitigate Settling Effects for Single-Ended Inputs

The ADS125H18 integrates a programmable delay feature that can be used to wait for any external settling behavior to stabilize before the ADC starts converting. This delay occurs before the first conversion after a conversion START as well as at the beginning of every new sequence step when the ADC sequencer is active. The delay value can range from 0 to 65535 modulator periods, t_{MOD}, and is independently programmable per step.

For example, Equation 9 and Equation 10 calculate one t_{MOD} in microseconds when f_{MOD} = 12.8MHz and the number of t_{MOD} periods required for a 1ms delay, respectively:

$$t_{MOD}\ (\mu s) = \frac{1}{f_{MOD}} = \frac{1}{12.8\text{MHz}} = 0.078125\ \mu s \quad (9)$$

$$t_{MOD}\ (\text{periods}) = \frac{\text{Delay}\ (\mu s)}{\text{tMOD}\ (\mu s)} = \frac{1000\ \mu s}{0.078125\ \mu s} = 12800 \quad (10)$$

Figure 2-14 shows a logic analyzer capture of the ADS125H18 data ready (DRDY) signal in yellow. DRDY indicates when new data is ready to be clocked out of the ADC. The time between DRDY pulses measures the sample-to-sample conversion latency. The green box indicates when the ADS125H18 sequencer automatically transitions from Step 1 (OWCS disabled) and starts converting on Step 2 (OWCS enabled). The ADC uses the sinc4 filter, operates at ODR = 12.5kSPS, and has DELAY = 0 such that the first conversion latency is approximately 320µs highlighted in red. The blue box indicates that all subsequent conversions are available at 1 / ODR.

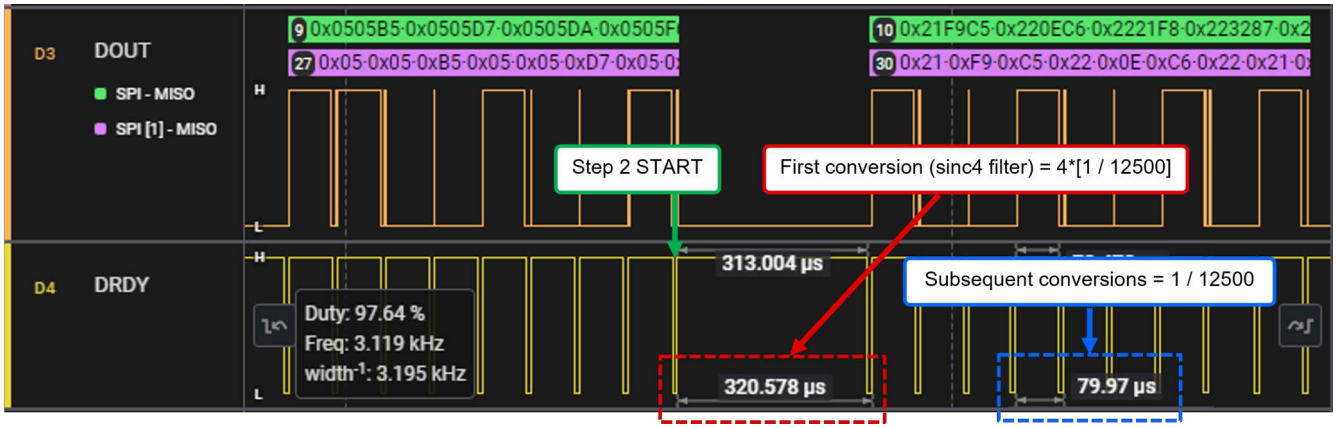

Figure 2-14. ADS125H18 DRDY Timing with Programmable Delay = 0d (Single-Ended Input, Sinc4 Filter, ODR = 12.5kSPS)

Comparatively, Figure 2-15 shows the same behavior as Figure 2-14 except that DELAY = 12800 (1ms). The red highlighted box indicates that the first conversion latency is one millisecond longer as a result. The subsequent conversion latency in blue remains unchanged.

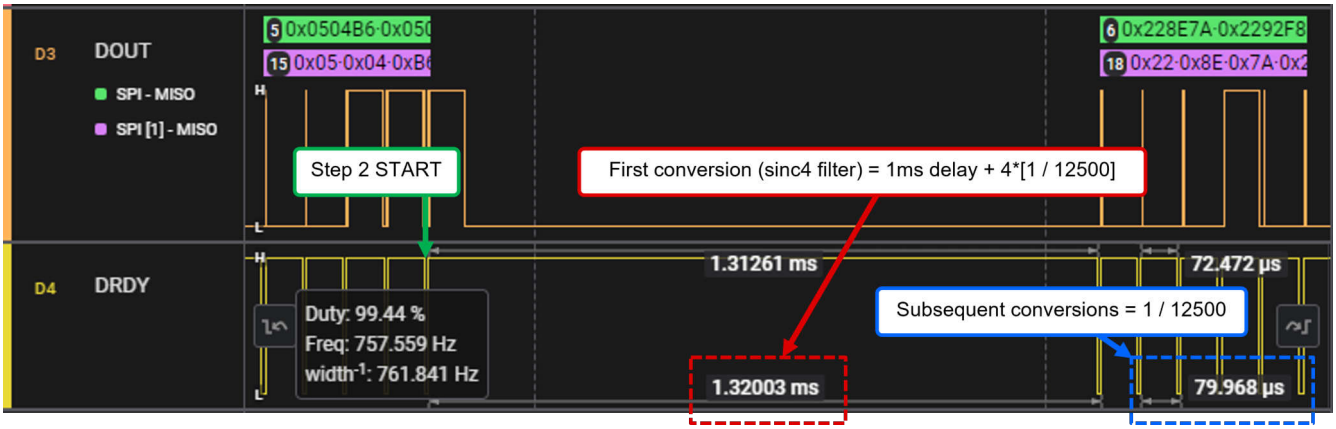

Figure 2-15. ADS125H18 DRDY Timing with Programmable Delay = 12800d (Single-Ended Input, Sinc4 Filter, ODR = 12.5kSPS)

Importantly, the delay feature only postpones the ADC sampling process such that the sequencer immediately enables the OWCS feature at "Step 2 START". This delay provides additional time for the OWCS feature to settle before the ADC conversion process begins, resulting in higher-accuracy measurements. Table 2-8 compares the measured and predicted results when R_{SOURCE} = 800kΩ, ODR = 12.5kSPS, and input capacitance = 10nF for two DELAY values: 0ms and 1ms

Table 2-8. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} for Programmable Delay = 0ms and 1ms (Single-Ended Input, R_{SOURCE} = 800kΩ, ODR = 12.5kSPS, Input Capacitance = 10nF)

ODR	DELAY = 0ms			DELAY = 1ms		
	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)	Error	$OWCS_{Delta\_SE}$	R_{SOURCE_OWCS} (kΩ)	Error
12.5kSPS	23.19%	480.91	40%	23.34%	632.30	21%

Table 2-8 shows that the programmable delay helps reduce the error between the actual and predicted R_{SOURCE} values by approximately 20%. This delay provides more stable measurements by enabling the ADC to capture less of the capacitor charging behavior and more of the steady-state behavior. Figure 2-16 shows this behavior by plotting the $OWCS_{Delta\_SE}$ values point-by-point instead of the single value averaged over 512 conversions shown in Table 2-8. Figure 2-16 uses R_{SOURCE} = 800kΩ, ODR = 12.5kSPS, input capacitance = 10nF, and two different DELAY values: 0ms and 1ms

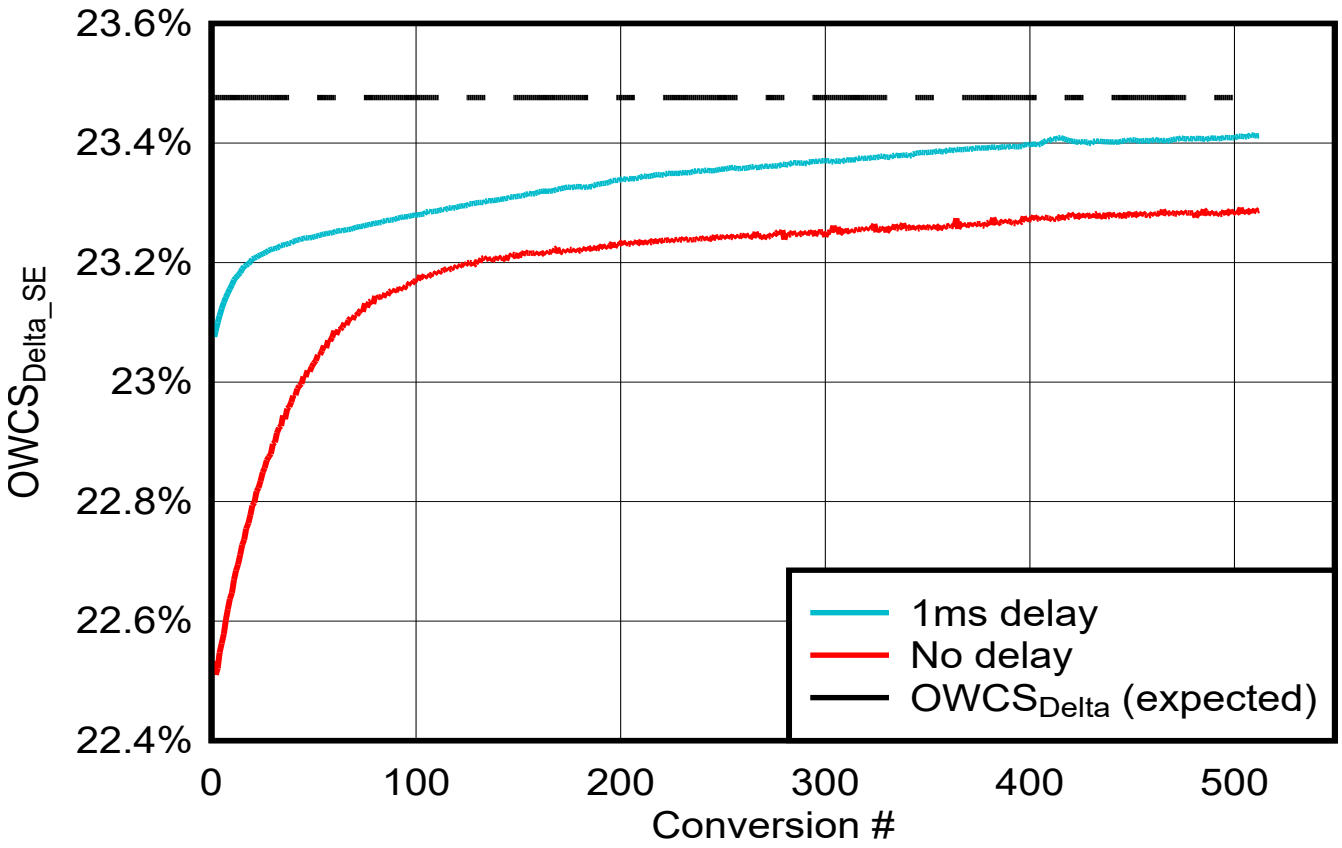

Figure 2-16. $OWCS_{Delta}$ Behavior when Programmable Delay = 0ms and 1ms (Single-Ended Input, R_{SOURCE} = 800kΩ, Input Capacitance = 10nF, ODR = 12.5kSPS)

Figure 2-16 shows how the programmable delay avoids most of the initial capacitor charging behavior to yield an average value closer to the expected. Additional delay could improve this behavior further. Finally, Figure 2-16 shows how a specific delay time affects the OWCS performance for a specific set of ODR and input capacitance parameters. A different delay value might be required if the user changes any of these system parameters.

2.3 Using OWCS with Differential Input Signals

The following sections describe the OWCS behavior when the user applies a differential input to the ADS125H18:

- Analyzing OWCS behavior for differential signals
- Verifying OWCS operation with a differential input using the ADS125H18EVM

2.3.1 Analyzing OWCS Behavior for Differential Input Signals

This section analyzes the behavior of the OWCS when a differential input with source impedance R_{SOURCE} is applied to the ADS125H18. Figure 2-17 shows the circuit used to analyze a differential input with the OWCS disabled and a differential R_{SOURCE} impedance:

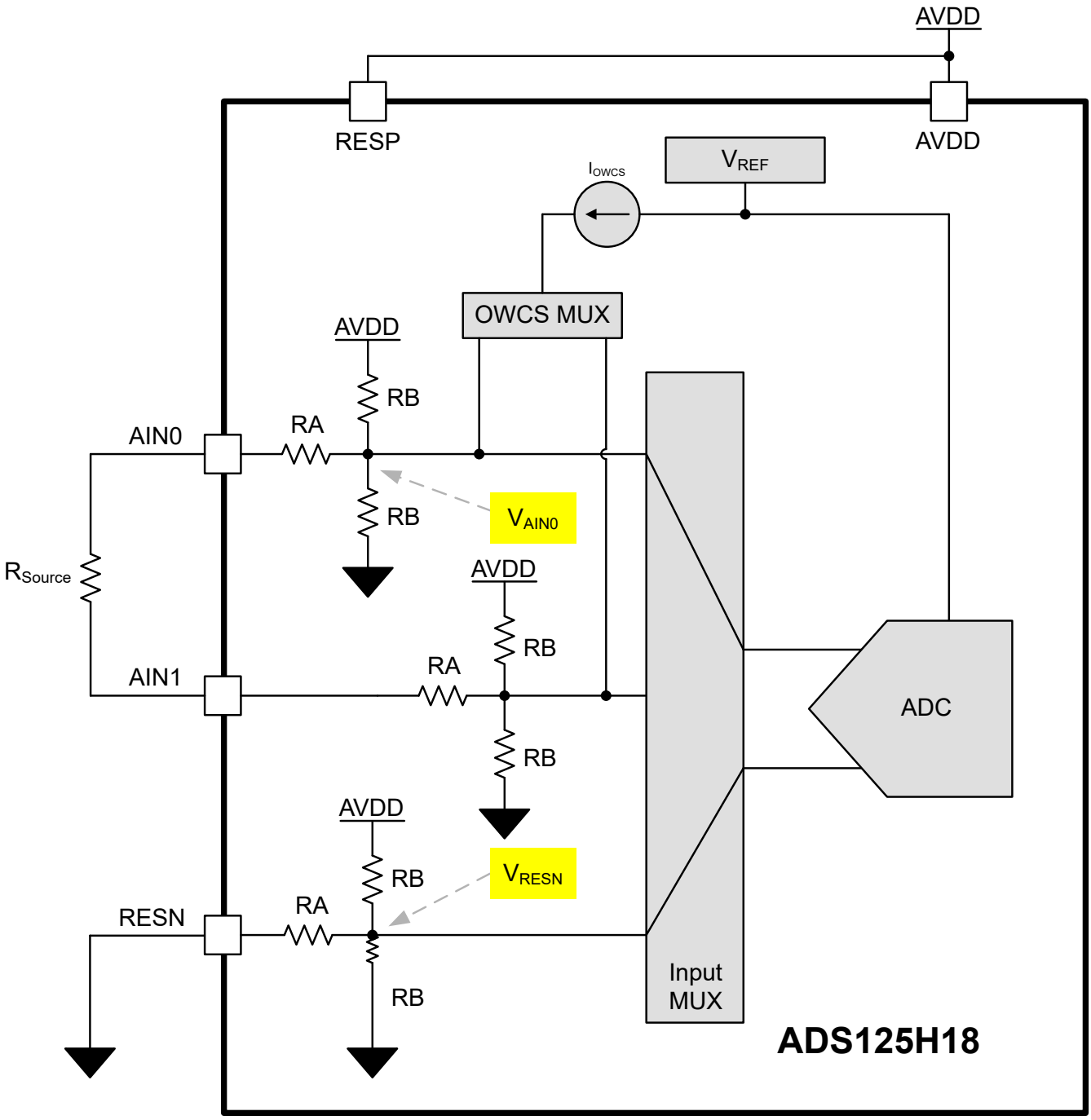

Figure 2-17. Analyzing OWCS Behavior for Differential Input Signals

Recall from Section 2.1 that the OWCS measurement is always single-ended with respect to RESN, even for a differential input signal. Therefore, detecting if either wire is open at AIN0 or AIN1 requires four measurements: one baseline measurement and one OWCS measurement per pin. However, this document only analyzes the voltage at AIN0 because the same methodology applies to AIN1 and should ideally provide the same threshold result in either case. Therefore, use the same analysis techniques described in Section 2.2.1. Equation 11 calculates the OWCS delta for a differential input between nodes V_{AIN0} and V_{RESN} in Figure 2-17:

$$\mathrm{OWCS_{Delta\_Diff}} = \frac{\left(\left[\mathrm{V(AIN0)_{OWCS\_on}} - \mathrm{V(RESN)_{OWCS\_on}}\right] - \left[\mathrm{V(AIN0)_{OWCS\_off}} - \mathrm{V(RESN)_{OWCS\_off}}\right]\right)}{\mathrm{V_{REF}}} \quad (11)$$

Equation 11 for differential measurements is virtually identical to Equation 3 for single-ended measurements; however, the additional AVDD voltage at AIN1 adds another source to include in the analysis. As in Section 2.2.1, use superposition to analyze the voltage contribution from each source in the circuit by shorting voltage sources or opening current sources. For example, Figure 2-18 shows that calculating the total voltage at AIN0 with the OWCS enabled, $V(AIN0)_{OWCS\_Diff}$, requires four different superposition configurations:

- $V(AIN0)_{AIN0\_AVDD\_Diff}$ = AVDD (AIN0) connected, all other sources removed (Figure 2-18, top left)
- $V(AIN0)_{AIN1\_AVDD\_Diff}$ = AVDD (AIN1) connected, all other sources removed (Figure 2-18, top right)
- $V(AIN0)_{RESN\_AVDD\_Diff}$ = AVDD (RESN) connected, all other sources removed (Figure 2-18, bottom left)
- $V(AIN0)_{AIN0\_OWCS\_Diff}$ = I_{OWCS} connected, all other sources removed (Figure 2-18, bottom right)

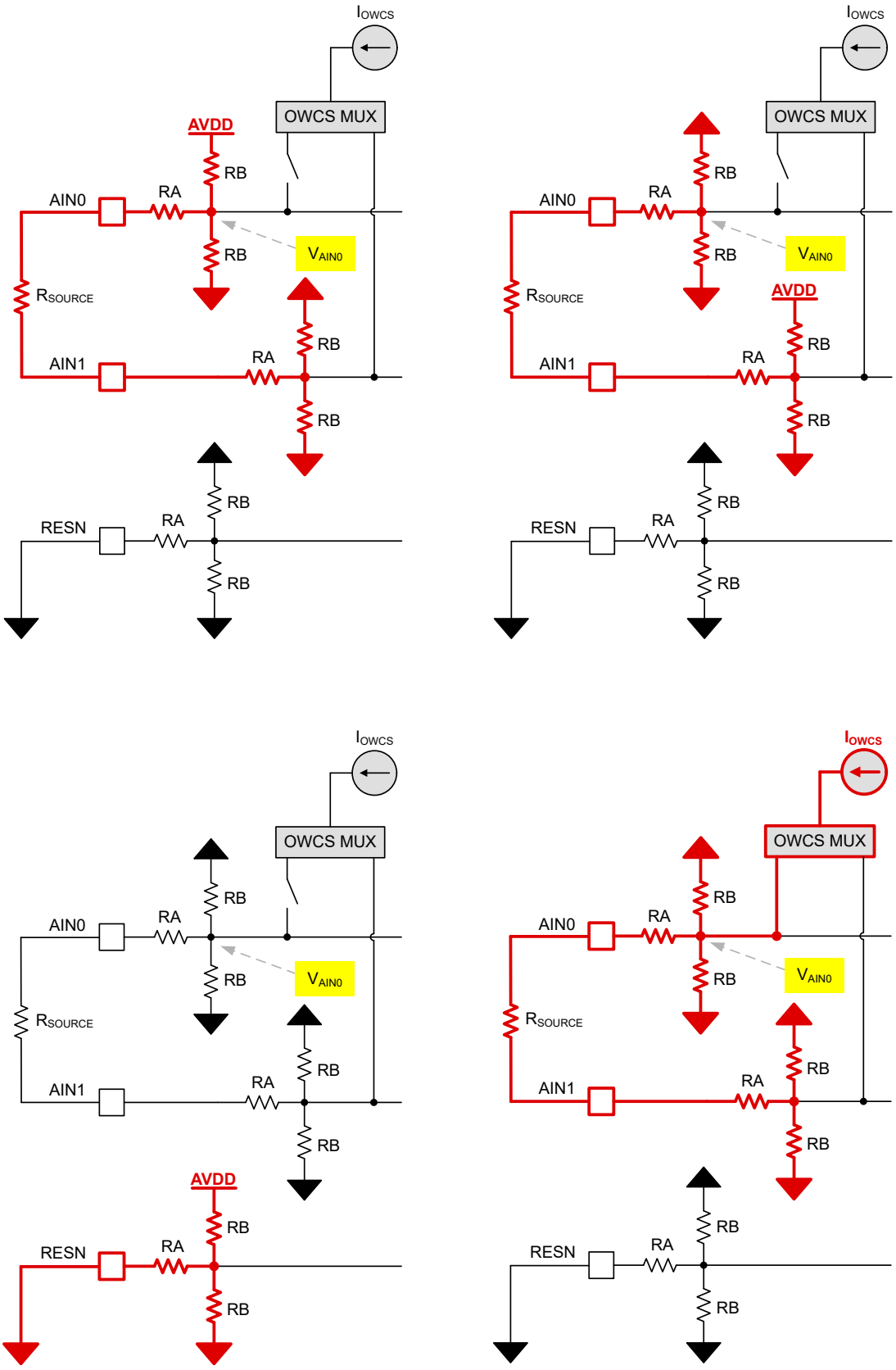

Figure 2-18. Example Superposition Networks for a Differential Input

Table 2-9 describes each nodal voltage after applying this same analysis to every term in Equation 11:

Table 2-9. Nodal Voltages for Differential Inputs

Voltage term	Nodal voltages
$V(AIN0)_{OWCS\_on}$	= $V(AIN0)_{AIN0\_AVDD\_Diff}$ + $V(AIN0)_{AIN1\_AVDD\_Diff}$ + $V(AIN0)_{RESN\_AVDD\_Diff}$ + $V(AIN0)_{AIN0\_OWCS\_Diff}$ where • – $V(AIN0)_{AVDD\_AIN0\_Diff}$ = AVDD (AIN0) connected, all other sources removed – $V(AIN0)_{AVDD\_AIN1\_Diff}$ = AVDD (AIN1) connected, all other sources removed – $V(AIN0)_{AVDD\_RESN\_Diff}$ = AVDD (RESN) connected, all other sources removed – $V(AIN0)_{OWCS\_Diff}$ = I_{OWCS} connected, all other sources removed
$V(RESN)_{OWCS\_on}$	= $V(RESN)_{AIN0\_AVDD\_Diff}$ + $V(RESN)_{AIN1\_AVDD\_Diff}$ + $V(RESN)_{RESN\_AVDD\_Diff}$ + $V(RESN)_{AIN0\_OWCS\_Diff}$ where • – $V(RESN)_{AVDD\_AIN0\_Diff}$ = AVDD (AIN0) connected, all other sources removed – $V(RESN)_{AVDD\_AIN1\_Diff}$ = AVDD (AIN1) connected, all other sources removed – $V(RESN)_{AVDD\_RESN\_Diff}$ = AVDD (RESN) connected, all other sources removed – $V(RESN)_{OWCS\_Diff}$ = I_{OWCS} connected, all other sources removed
$V(AIN0)_{OWCS\_off}$	= $V(AIN0)_{AIN0\_AVDD\_Diff}$ + $V(AIN0)_{AIN1\_AVDD\_Diff}$ + $V(AIN0)_{RESN\_AVDD\_Diff}$ where • – $V(AIN0)_{AVDD\_AIN0\_Diff}$ = AVDD (AIN0) connected, all other sources removed – $V(AIN0)_{AVDD\_AIN1\_Diff}$ = AVDD (AIN1) connected, all other sources removed – $V(AIN0)_{AVDD\_RESN\_Diff}$ = AVDD (RESN) connected, all other sources removed
$V(RESN)_{OWCS\_off}$	= $V(RESN)_{AIN0\_AVDD\_Diff}$ + $V(RESN)_{AIN1\_AVDD\_Diff}$ + $V(RESN)_{RESN\_AVDD\_Diff}$ where • – $V(RESN)_{AVDD\_AIN0\_Diff}$ = AVDD (AIN0) connected, all other sources removed – $V(RESN)_{AVDD\_AIN1\_Diff}$ = AVDD (AIN1) connected, all other sources removed – $V(RESN)_{AVDD\_RESN\_Diff}$ = AVDD (RESN) connected, all other sources removed

Fortunately, like Section 2.2.1, almost all terms in Table 2-9 cancel out such that the final equation is just the voltage at AIN0 with OWCS enabled, $V(AIN0)_{OWCS\_Diff}$. Equation 12 rewrites $V(AIN0)_{OWCS\_Diff}$ in terms of resistances RA, RB, and R_{SOURCE} as well as current I_{OWCS}:

$$V(AIN0)_{OWCS\_Diff} = I_{OWCS} \times (RB||RB||[2 \times RA + R_{SOURCE} + (RB||RB)]) \quad (12)$$
$$V(AIN0)_{OWCS\_Diff} = I_{OWCS} \times \left(\frac{RB}{2}||\left[2 \times RA + R_{SOURCE} + \frac{RB}{2}\right]\right)$$

Finally, recall from Equation 2 that the actual OWCS current, I_{OWCS}, is relative to the reference voltage, VREF. Therefore, the I_{OWCS} term in Equation 12 can be rewritten in terms of the nominal OWCS current, I_{OWCS_NOM}, and simplified as shown in Equation 13:

$$OWCS_{Delta\_Diff} = \frac{(I_{OWCS\_NOM} \times V_{REF}) \times \left(\frac{RB}{2}||\left[2 \times RA + R_{SOURCE} + \frac{RB}{2}\right]\right)}{V_{REF}} \quad (13)$$
$$OWCS_{Delta\_Diff} = I_{OWCS\_NOM} \times \left(\frac{RB}{2}||\left[2 \times RA + R_{SOURCE} + \frac{RB}{2}\right]\right)$$

Equation 13 can also be rewritten to predict the differential R_{SOURCE_OWCS} value from the measured $OWCS_{Delta}$ threshold, as shown in Equation 14:

$$R_{SOURCE\_OWCS\_Diff} = \frac{OWCS_{Delta\_Diff} \times 4 \times (RB + 2 \times RA) - 4 \times RA \times RB \times I_{OWCS_{NOM}} - RB^2 \times I_{OWCS\_NOM}}{2 \times RB \times I_{OWCS\_NOM} - 4 \times OWCS_{Delta\_Diff}} \quad (14)$$

Table 2-1 includes RB and I_{OWCS_NOM} values for all ADS125H18 versions. Figure 2-19 applies all possible parameters from Table 2-1 to Equation 13and sweeps R_{SOURCE} from 1kΩ to 1GΩ to generate the ideal $OWCS_{Delta\_Diff}$ curves:

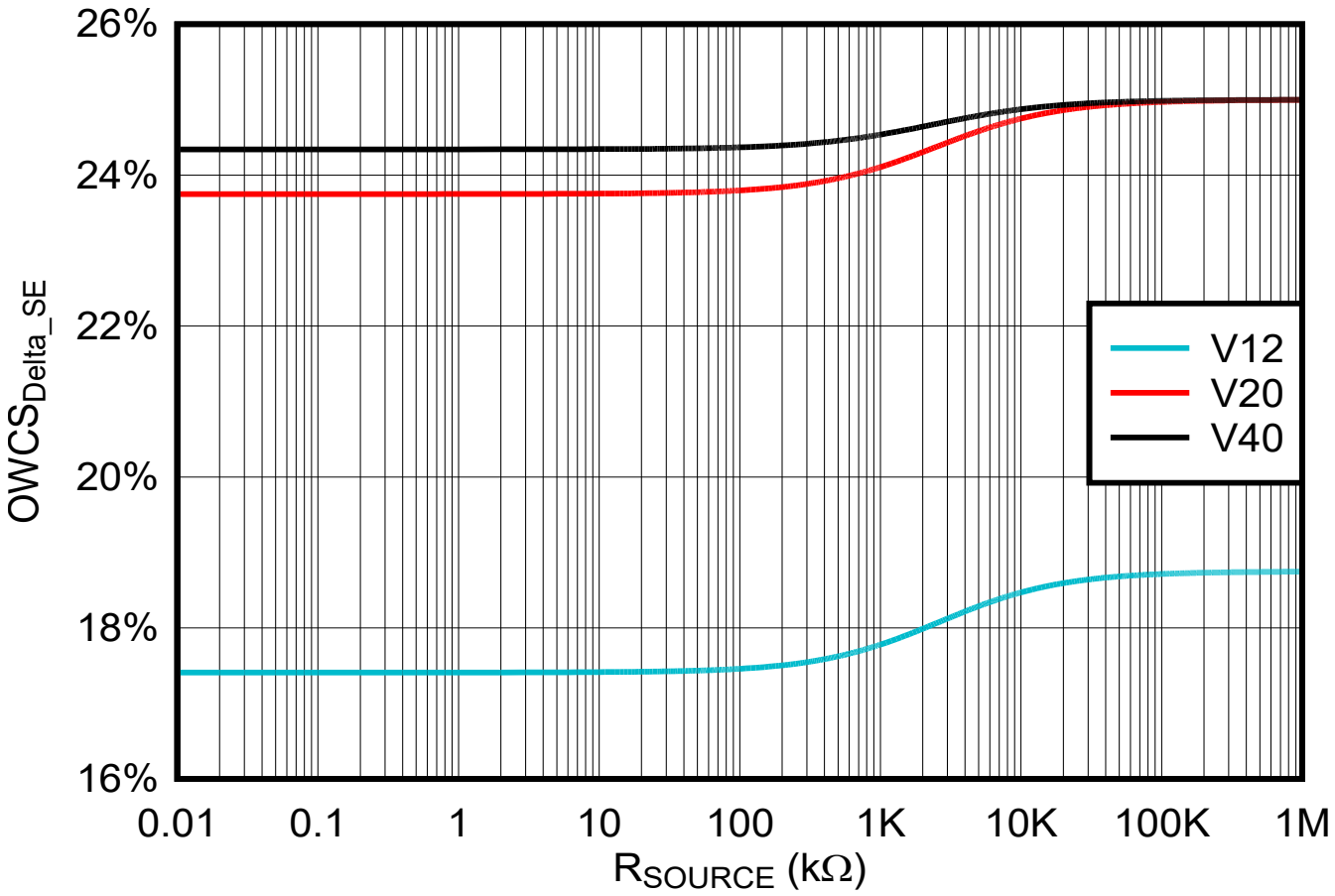

Figure 2-19. $OWCS_{Delta}$ Behavior for all ADS125H18 Variants Measuring a Differential Input (R_{SOURCE} = 1kΩ to 1GΩ)

Figure 2-19 reveals two important takeaways for differential OWCS measurements:

1. The significant change in $OWCS_{Delta\_Diff}$ occurs in the range from 100kΩ to 10MΩ. This range indicates where a change in input impedance is most detectable
2. Each $OWCS_{Delta\_Diff}$ curve has a theoretical minimum and maximum at R_{SOURCE} = 0Ω and R_{SOURCE} = ∞, respectively. The difference between the minimum and maximum decreases as the attenuation factor increases, resulting in a narrower margin to indicate an OWCS fault. Table 2-10 summarizes these results.

Table 2-10. Minimum and Maximum $OWCS_{Delta}$ Values for a Differential Input

ADS125H18 version	Attenuation factor	$OWCS_{Delta\_Diff}$ Min	$OWCS_{Delta\_Diff}$ Max	Difference
V12	7	17.41%	18.75%	1.34%
V20	10	23.75%	25.00%	1.25%
V40	19	24.34%	25.00%	0.66%

Comparing Table 2-10 and Table 2-3 shows that the OWCS delta range for each ADS125H18 version is smaller for differential input versus single-ended inputs. This difference results from the additional resistor divider (RA and RB) introduced by the other input in the differential pair. Therefore, detecting an OWCS fault is comparatively more difficult with a differential input compared to a single-ended input.

2.3.2 Verifying OWCS Operation with a Differential Input Using the ADS125H18EVM

Testing was performed to verify OWCS operation using the ADS125H18 evaluation module (EVM). The ADS125H18 EVM is a platform for evaluating the functionality and performance of the ADS125H18. The ADS125H18EVM-PDK eases the evaluation of the device with hardware, software, and computer connectivity through the universal serial bus (USB) interface. Figure 2-20 shows the ADS125H18 EVM.

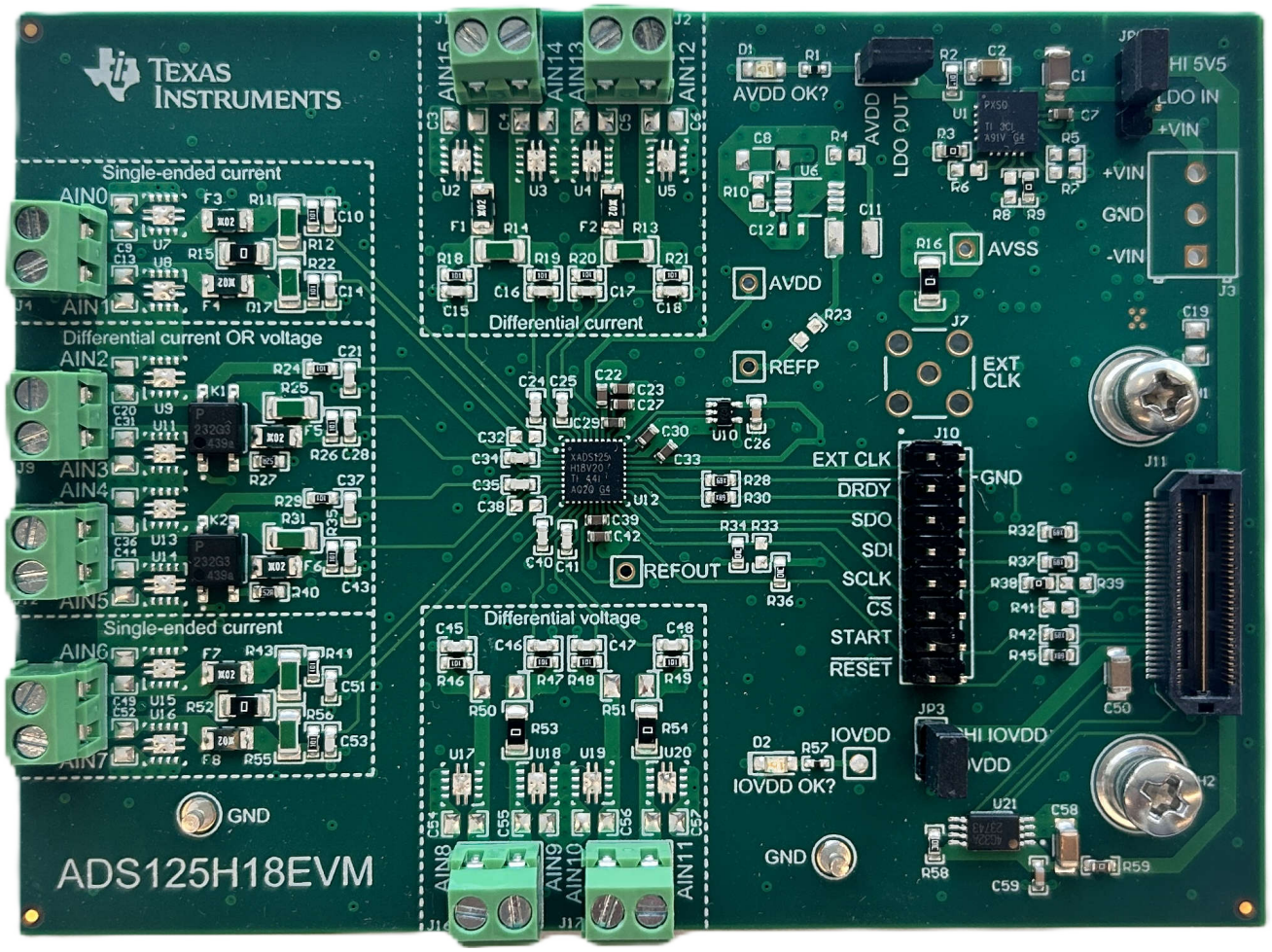

Figure 2-20. ADS125H18EVM Board Image

The following ADC and EVM conditions were used to perform the OWCS verification:

- ADS12H18V20
- Internal 25.6MHz oscillator
- Speed mode 3 (f_{MOD} = 12.8MHz)
- Internal 2.5V reference voltage
- Single-shot sequence mode
- Four enabled steps:
  - Positive input channel
    - Step 1 = OWCS disabled
    - Step 2 = OWCS enabled
  - Negative input channel
    - Step 3 = OWCS disabled
    - Step 4 = OWCS enabled
- Three output data rates (ODR):
  - 1.6kSPS (sinc4+sinc1 filter, OSR = 8000)
  - 12.5kSPS (sinc4 filter, OSR = 1024)
  - 50kSPS (sinc4+sinc1 filter, OSR = 256)
- 512 conversions per OWCS step (to capture possible settling behavior)
- No external passive (R or C) components

2.3.2.1 Verifying OWCS Operation with a Differential Input

Figure 2-21 shows the simple OWCS verification setup using a differential input on the ADS125H18 EVM. No passive components were installed between the terminal block and the ADC input pins. Various 0.1% R_{SOURCE} resistors were connected between the positive and negative inputs on the EVM terminal blocks – for example, between AIN8 and AIN9 – to verify performance across a wide range of input impedances. This configuration results in a 0V input signal to the ADC to simplify the analysis.

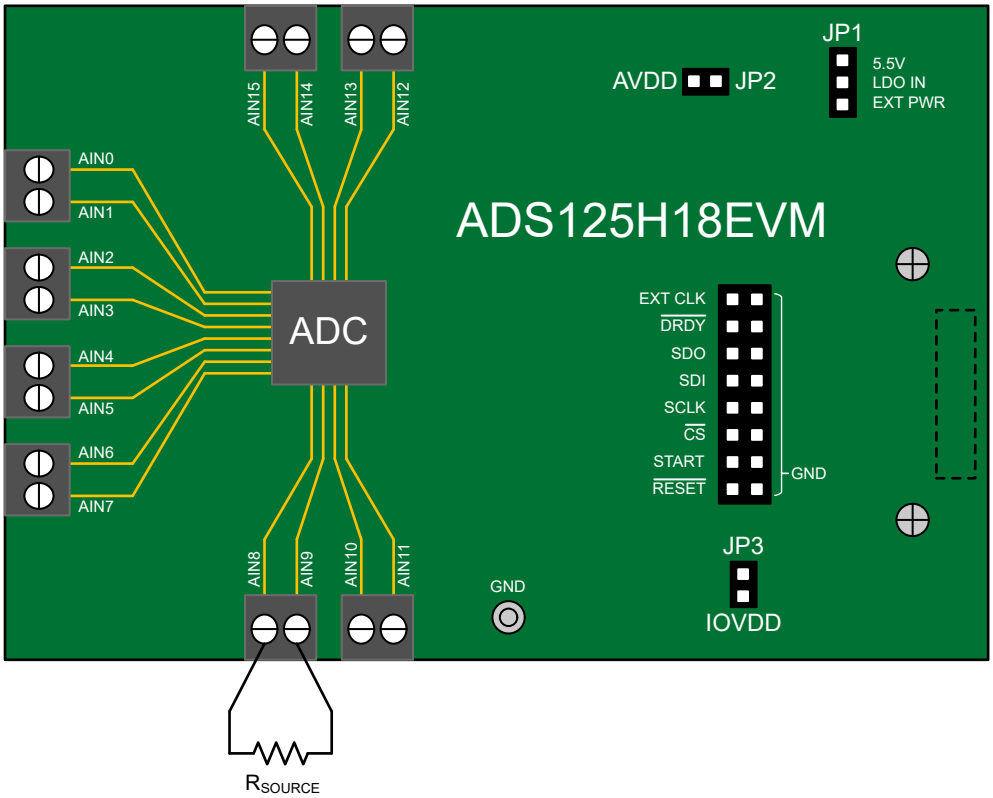

Figure 2-21. OWCS Verification Setup for a Differential Input

Recall from Section 2.1 that detecting an open wire for a differential input requires four measurements: one baseline and one detection measurement per wire. First, the positive input channel was measured by taking 512 conversions with the OWCS disabled followed by 512 conversions with the OWCS enabled. Next, the negative input channel was measured by taking 512 conversions with the OWCS disabled followed by 512 conversions with the OWCS enabled. The 1024 conversions with OWCS disabled were averaged and the 1024 conversions with OWCS enabled were averaged. Two $OWCS_{Delta\_Diff}$ values were then calculated using Equation 1 and converted to a predicted R_{SOURCE_OWCS} value by rearranging the terms in Equation 13. Table 2-11 shows the measured, averaged results across 3 different ODRs and 6 different R_{SOURCE} values:

Table 2-11. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} Values Using Different ODRs and R_{SOURCE} Inputs (Differential Input)

R_{SOURCE} (kΩ)	ODR = 1.6kSPS		ODR = 12.5kSPS		ODR = 50kSPS	
	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)
25.5	23.76%	29.33	23.76%	29.68	23.76%	26.18
99.9	23.80%	102.80	23.80%	105.24	23.80%	99.04
499.9	23.96%	502.89	23.96%	504.66	23.96%	492.91
1000.0	23.41%	999.91	24.11%	1003.13	24.01%	938.98
1999.9	24.31%	1994.77	24.30%	1991.76	24.30%	1970.80
∞ (Open)	25.01%	>1G	25.01%	>1G	25.00%	>1G

Table 2-11 reveals that the OWCS correctly determined the applied differential source impedance across different ODRs for virtually all R_{SOURCE} values. However, Table 2-11 also shows an approximate 4kΩ offset between the predicted R_{SOURCE_OWCS} values and the actual R_{SOURCE} values. Fortunately, this seemingly large error in kΩ translates to a very small 0.05% average input code variation that can be attributed to ADS125H18 resistor divider matching. The end result is that the OWCS fault detection behaves as intended: this feature accurately identified an open circuit as well as significant changes in the source impedance.

2.3.2.2 Using a Fewer Number of Conversions for OWCS with a Differential Input

Table 2-11 shows an average $OWCS_{Delta\_Diff}$ value calculated from 2048 samples. Section 2.3.1 states that 512 conversions were used per sequence step to capture any possible settling behavior. However, taking a total of 2048 conversions to detect a wire break might require an impractical amount of time in many industrial systems. For example, 2048 conversions at 1.6kSPS takes approximately 1280ms. Analyzing the sampled data points helps indicate how many conversions are actually necessary for accurate fault detection.

Figure 2-22 plots the $OWCS_{Delta\_Diff}$ values point-by-point instead of a single, averaged value shown in Table 2-11. Figure 2-22 uses R_{SOURCE} = 500kΩ and 3 different ODR values: 1.6kSPS, 12.5kSPS, and 50kSPS. Note that Figure 2-22 only plots the results from the positive input channel because the results from the negative input channel were effectively identical.

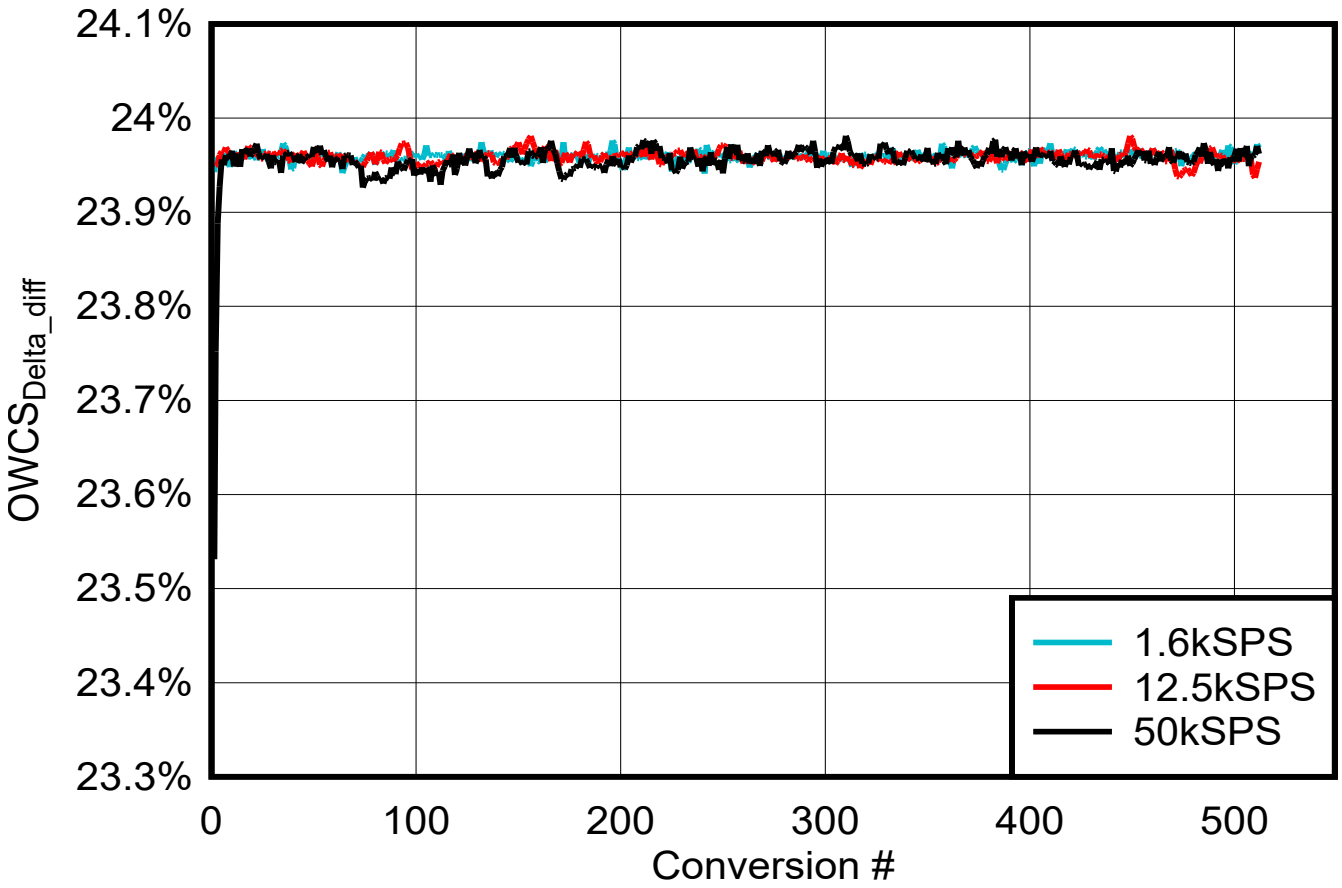

Figure 2-22. Measured $OWCS_{Delta}$ Values Plotted Point-by-Point to Show Settling Behavior (Differential Input, R_{SOURCE} = 500kΩ)

Figure 2-22 reveals that some settling behavior exists in the data, but otherwise very little useful information due to the unsettled outliers. Breaking this data into unsettled and settled components helps clarify why fault detection errors might occur. Also, note that this behavior is shown at R_{SOURCE} = 500kΩ but exists across all values of R_{SOURCE}.

2.3.2.2.1 Unsettled Data for Differential Inputs

Table 2-12 includes the measured $OWCS_{Delta\_Diff}$ and predicted R_{SOURCE_OWCS} values calculated from only the first three conversion data at three different ODRs:

Table 2-12. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} Value Using Only First Conversion Data Point (Differential Input, R_{SOURCE} = 500kΩ)

ODR	First three conversions only	
	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)
1.6kSPS	23.95%	474.32
12.5kSPS	23.95%	478.92
50kSPS	23.68%	0 (short)

Importantly, Table 2-12 shows that the unsettled data at ODR = 50kSPS indicates a short instead of R_{SOURCE} = 500kΩ. Using the first three conversion results to determine a wire break under these conditions could provide a false negative to the user. Comparatively, the other two ODR settings correctly predict the R_{SOURCE_OWCS} value. These combined results indicate that faster data rates can result in incorrect OWCS calculations. The user must account for this possibility when they choose both the OWCS data rate and the number of conversions for the system.

2.3.2.2.2 Settled Data for Differential Inputs

Figure 2-23 shows the same data as Figure 2-22 but with the first three conversions removed for all ODRs. In other words, Figure 2-23 plots conversions 4 through 512 point-by-point instead of a single value averaged over multiple conversions:

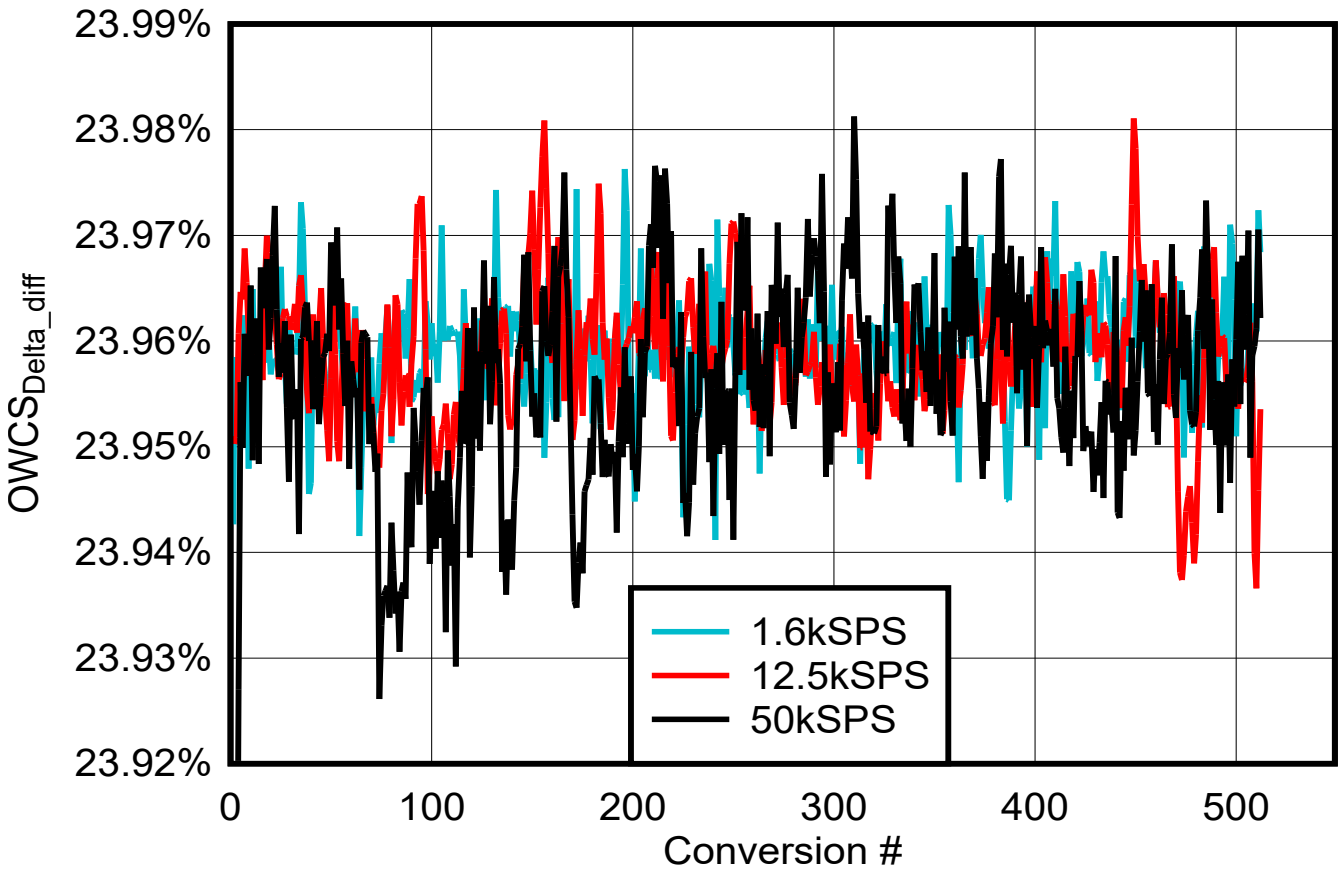

Figure 2-23. Measured $OWCS_{Delta}$ Values – Excluding First 3 Conversions – Plotted Point-by-Point Across ODR (Differential Input, R_{SOURCE} = 500kΩ)

Figure 2-23 shows the desired behavior for OWCS calculations. The $OWCS_{Delta\_Diff}$ values generally have a Gaussian (broadband) noise distribution. Additionally, the noise distribution gets tighter as the ODR decreases because the ADC digital filter bandwidth also decreases. Both of these factors indicate typical ADC data dominated by thermal noise – with no settling effects – that can correctly predict R_{SOURCE_OWCS} for any conversion.

Table 2-13 analyzes the distribution of $OWCS_{Delta\_Diff}$ values from Figure 2-23 to determine the minimum and maximum variation in predicted R_{SOURCE_OWCS}:

Table 2-13. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} Variance (Differential Input, R_{SOURCE} = 500kΩ)

ODR	Minimum**		Maximum**		Difference (Max – Min) (kΩ)
	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)	
1.6kSPS	23.93%	451.34	23.98%	553.54	102.20
12.5kSPS	23.94%	438.78	23.98%	567.73	128.95
50kSPS	23.93%	410.18	23.98%	568.34	158.15

**excluding the first three conversions

Table 2-13 shows that the difference between the minimum and maximum is smaller at the lowest ODR (1.6kSPS) and increases as the ODR increases. However, the minimum and maximum values still correctly approximate the 500kΩ R_{SOURCE} impedance in all cases. Therefore, it is possible to conclude that using significantly fewer conversions (<<512) under these conditions could have enabled faster wire break detection without sacrificing reliability.

2.3.2.3 How Input Capacitance Affects OWCS Performance for Differential Inputs

Section 2.3.2.1 describes the OWCS performance using the standard ADS125H18 EVM with no passive components installed between the terminal block and the ADC input pins. However, many industrial systems include capacitors on the inputs for noise rejection or other purposes. Figure 2-24 shows an example of a system with a differential capacitor in red installed between the AIN8 and AIN9 analog inputs as well as common-mode (single-ended) capacitors in blue installed between each input and ground. The ADS125H18 EVM includes footprints for these components that are not populated by default.

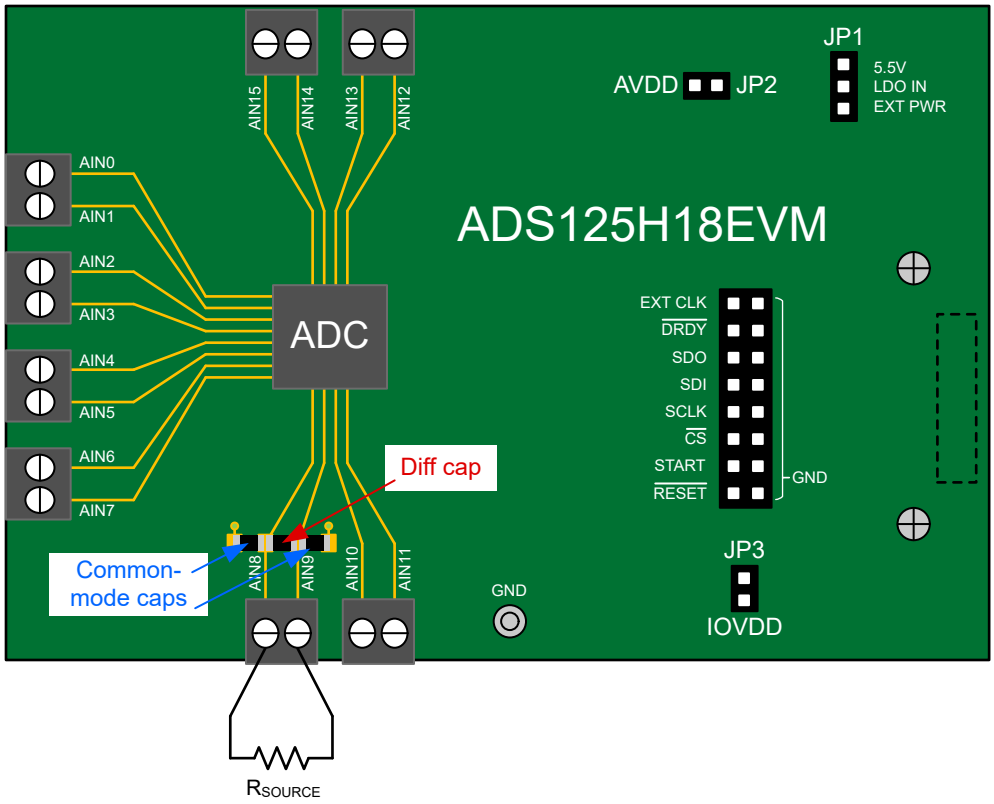

Figure 2-24. Measuring a Differential Input on the ADS125H18EVM with Input Capacitance

Adding capacitance on the analog inputs can help reduce noise at the expense of increased settling time due to capacitor charging. This behavior affects the measurement when the current sources (OWCS) switch on because the capacitor cannot charge up to the required voltage instantaneously. This effect is also measurable when the OWCS turn off because the capacitor cannot discharge instantaneously.

Multiple capacitors were installed on the ADS125H18 EVM to measure how input capacitance affects the OWCS detection behavior. Otherwise, the same settings were used as listed in the beginning of Section 2.3.2. First, the positive input channel was measured by taking 512 conversions with the OWCS disabled followed by 512 conversions with the OWCS enabled. Next, the negative input channel was measured by taking 512 conversions with the OWCS disabled followed by 512 conversions with the OWCS enabled. The 1024 conversions with OWCS disabled were averaged and the 1024 conversions with OWCS enabled were averaged. Two $OWCS_{Delta\_Diff}$ values were then calculated using Equation 1 and converted to a predicted R_{SOURCE_OWCS} value by rearranging the terms in Equation 13. Table 2-14 shows the measured, averaged results when R_{SOURCE} = 800kΩ across 3 different ODRs and 4 different input capacitance values:

Table 2-14. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} for Different ODRs and Input Capacitance (Differential Input, R_{SOURCE} = 800kΩ)

Input capacitance (nF)	ODR = 1.6kSPS		ODR = 12.5kSPS		ODR = 50kSPS	
	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)
1	24.00%	616.53	24.00%	611.37	23.97%	528.68
10	23.96%	318.14	23.89%	324.16	23.67%	0.00
100	23.89%	515.54	23.41%	0.00	22.86%	0.00
1000	23.29%	0.00	22.62%	0.00	22.48%	0.00

Table 2-14 reveals that the $OWCS_{Delta\_Diff}$ values vary greatly across data rate and input capacitance. For example, the predicted R_{SOURCE_OWCS} value when ODR = 1.6kSPS and input capacitance = 1nF is 616.53kΩ. This result is approximately 77% of the ideal 800kΩ R_{SOURCE} value. However, the predicted R_{SOURCE_OWCS}

value when ODR = 50kSPS and input capacitance = 1000nF is 0Ω, indicating a complete short! Reviewing the data point-by-point helps illuminate why such a large discrepancy exists across the different parameters.

Figure 2-25 plots the $OWCS_{Delta\_Diff}$ values point-by-point for R_{SOURCE} = 800kΩ and ODR = 12.5kSPS for 4 different input capacitance values: 1nF, 10nF, 100nF, and 1000nF.

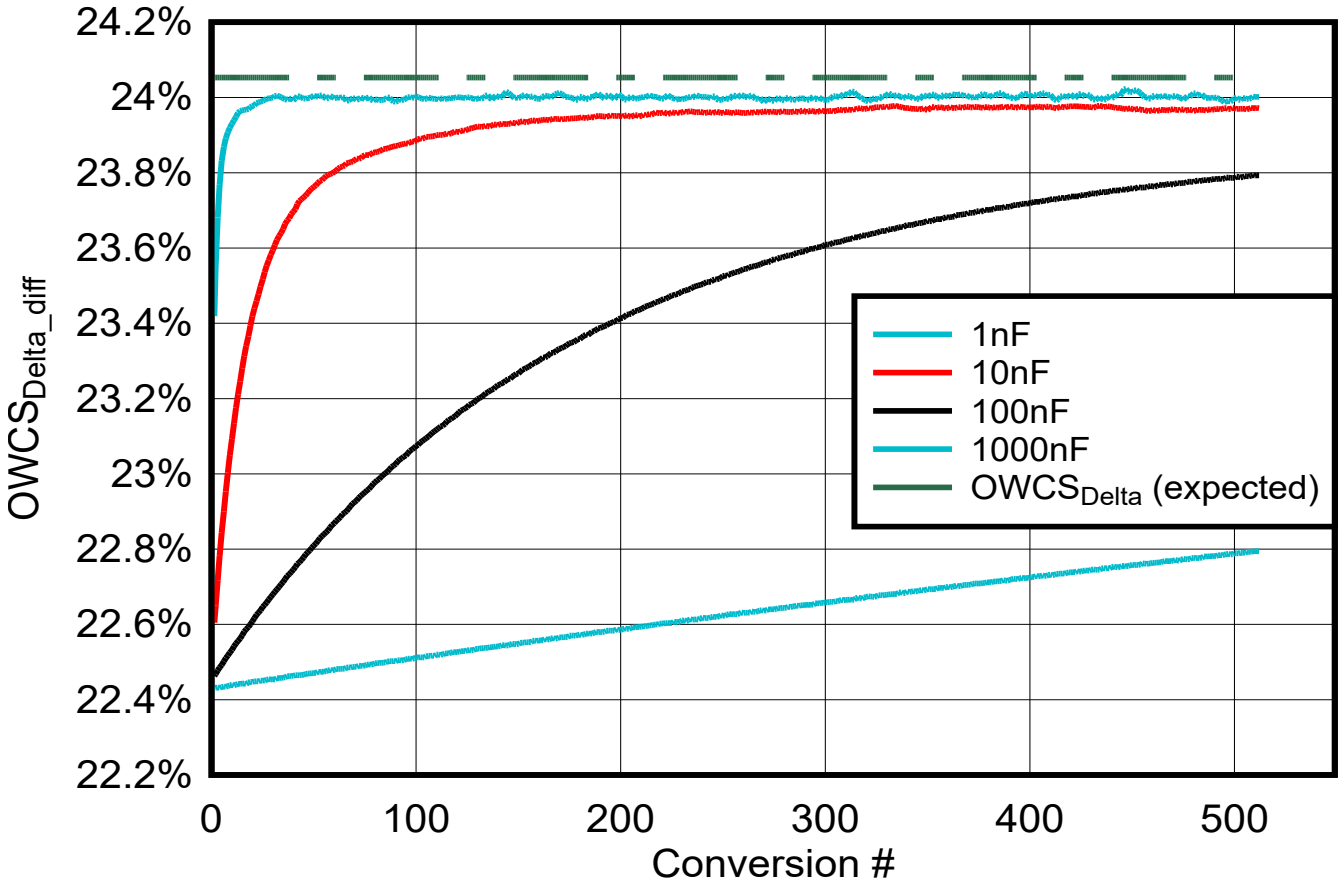

Figure 2-25. $OWCS_{Delta}$ Behavior Across Different Input Capacitance Values (Differential Input, R_{SOURCE} = 800kΩ, ODR = 12.5kSPS)

Figure 2-25 reveals the significant effect input capacitance has on the OWCS behavior as well as why there is such extreme variation in the results listed in Table 2-14:

- The 1-nF plot reaches a steady-state value approximately equal to the expected $OWCS_{Delta\_Diff}$ value of 24.05% after approximately 35 conversions
- The 10-nF plot reaches a steady-state value approximately equal to the expected $OWCS_{Delta\_Diff}$ value of 24.05% after approximately 200 conversions
- The other two (2) plots never reach the expected $OWCS_{Delta\_SE}$ value
- The other two (2) plots never reach a settled state even after 512 conversions

These conclusions make sense because the capacitor time constant increases as the input capacitance increases, resulting in longer settling times.

Changing the ODR with input capacitors installed also affects the OWCS performance as mentioned previously. Figure 2-26 plots the $OWCS_{Delta\_Diff}$ values for R_{SOURCE} = 800kΩ and input capacitance = 10nF point-by-point for 3 different ODRs: 1.6kSPS, 12.5kSPS, and 50kSPS. The plot x-axis units are in milliseconds to clearly indicate the charging behavior captured by the ADC. Also, all plots start at the same point and overlay each other.

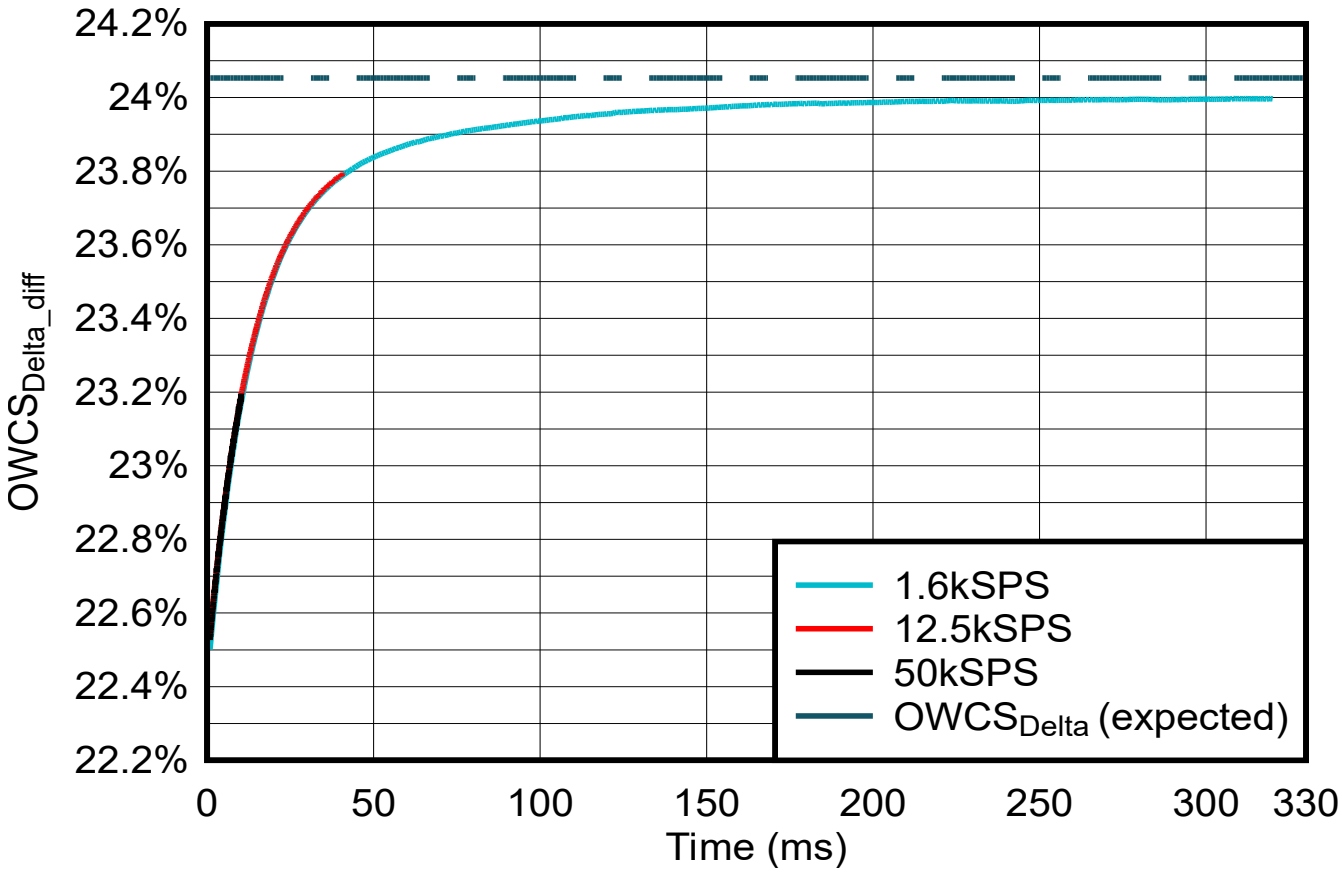

Figure 2-26. $OWCS_{Delta}$ Behavior Across Different ODR Values (Differential Input, R_{SOURCE} = 800kΩ, Input Capacitance = 10nF)

Again, these results make sense because the ADC sampling time decreases as ODR increases, even though the capacitor time constant remains unchanged. Therefore, the ADC captures more of the capacitor charging behavior and less of the steady-state behavior as the data rate increases, leading to less stable measurements.

The input capacitance does help reduce noise as stated previously. Figure 2-27 plots the measured ADC code point-by-point with OWCS disabled for R_{SOURCE} = 800kΩ and ODR = 12.5kSPS for 4 different values of input capacitance: 1nF, 10nF, 100nF, and 1000nF.

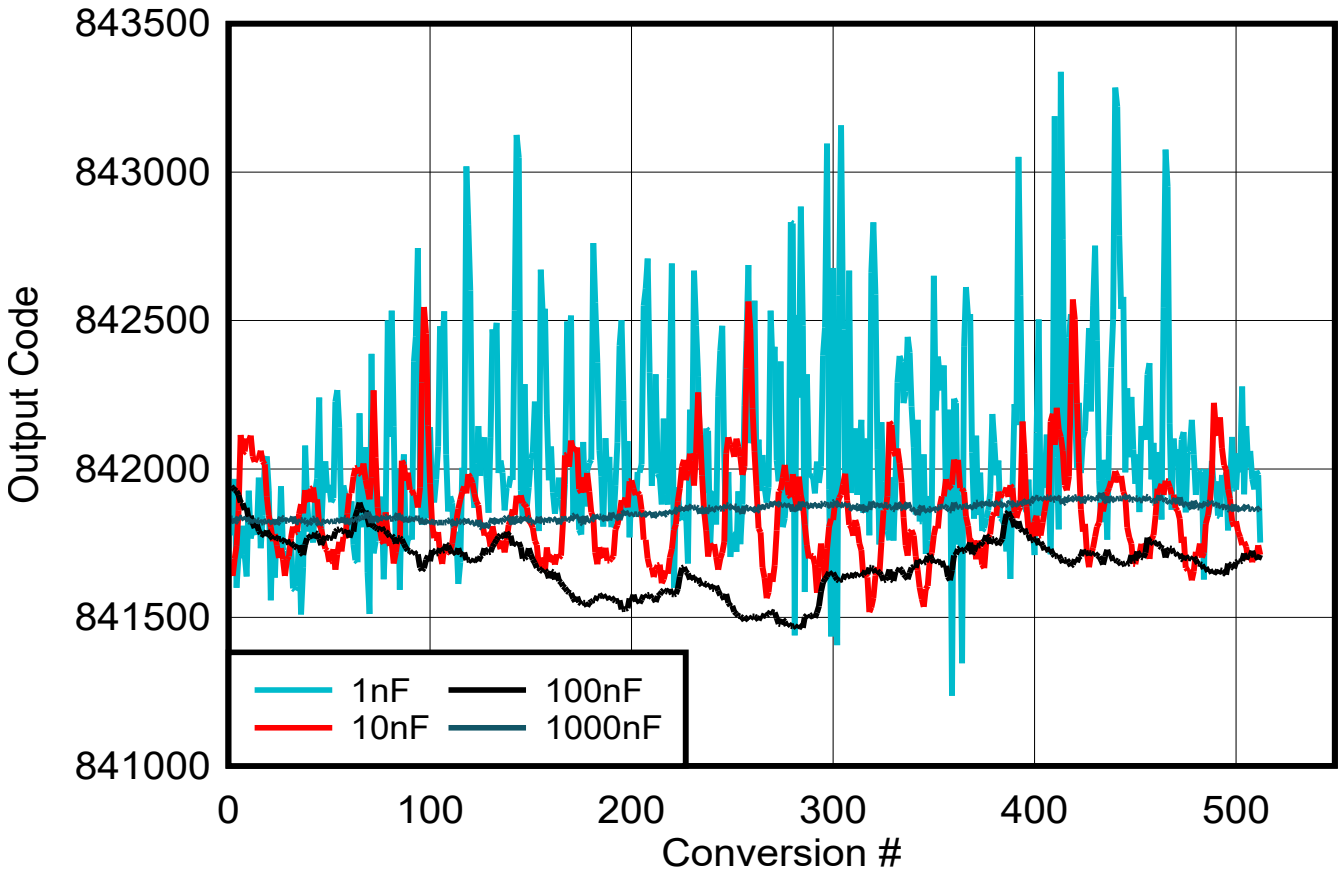

Figure 2-27. ADC Code Behavior Across Different Input Capacitance Values (Differential Input, OWCS disabled, R_{SOURCE} = 800kΩ, ODR = 12.5kSPS)

Figure 2-27 qualitatively shows that the noise spread decreases as the input capacitance increases. Specifically, the RMS noise when input capacitance = 1nF is 265.39 codes, while the RMS noise when the input capacitance = 1000nF is only 16.97 codes. Designers must take care to balance the noise reduction benefits of an input capacitor with the challenges introduced by capacitor settling.

2.3.2.3.1 Back-to-Back, Differential OWCS Measurements with Input Capacitance

One additional consideration regarding how input capacitance affects OWCS performance for differential inputs is the order in which the OWCS measurements are made. Specifically, back-to-back measurements can result in settling behavior even when the OWCS are disabled. As an example, assume that back-to-back measurements use the following sequence:

1. Measure AIN(n)-RESN with OWCS off
2. Measure AIN(n)-RESN with OWCS on
3. Measure AIN(n+1)-RESN with OWCS off
4. Measure AIN(n+1)-RESN with OWCS on

The user might expect that steps 1 and 3 exhibit the stable behavior shown in Figure 2-27 while steps 2 and 4 exhibit the settling behavior shown in Figure 2-25. However, the differential input capacitance continues to hold charge even after step 2 completes such that step 3 also exhibits settling behavior. Figure 2-28 plots the ADC output codes during all four steps of a differential OWCS measurement between AIN12 and AIN13 where R_{SOURCE} = 100kΩ, ODR = 12.5kSPS, and the input capacitance = 10nF:

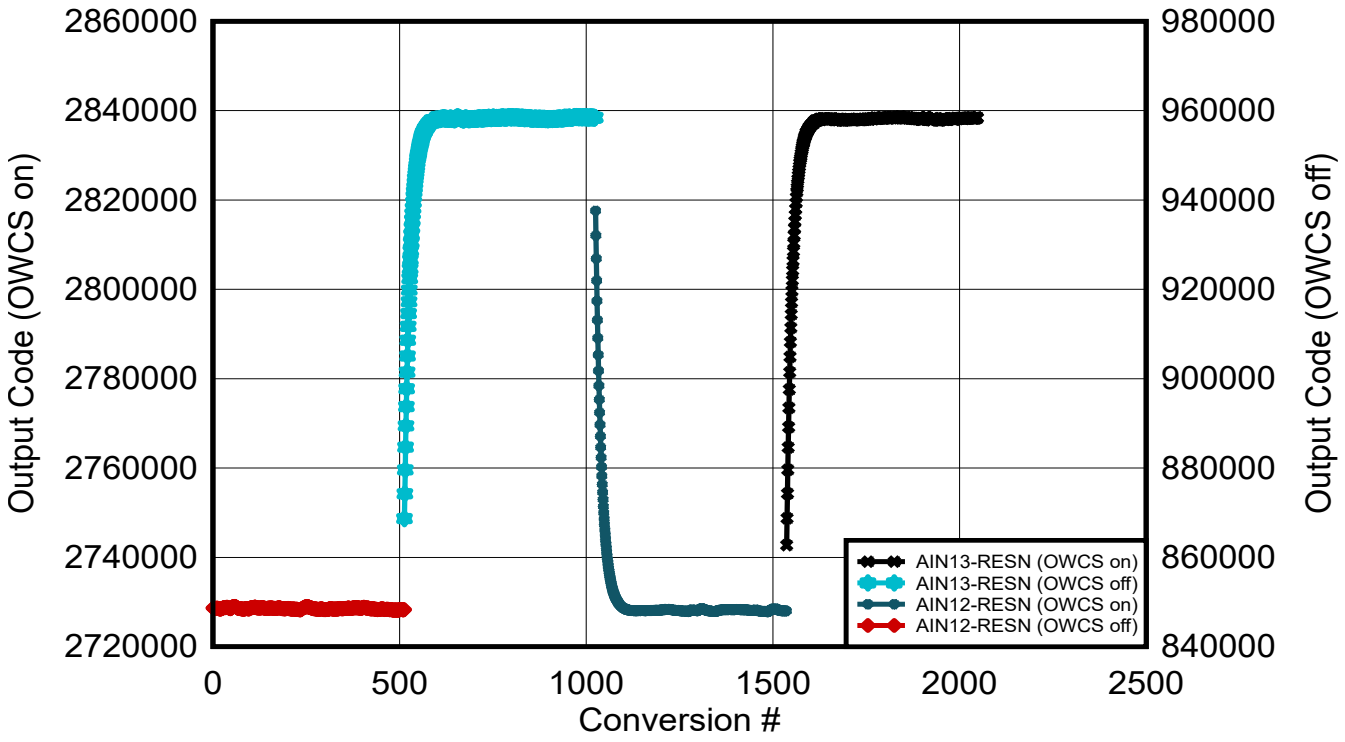

Figure 2-28. ADC Code Behavior for Back-to-Back OWCS Measurements (Differential Input, R_{SOURCE} = 800kΩ, ODR = 12.5kSPS, Input Capacitance = 10nF)

Figure 2-28 shows the expected behavior for steps 1, 2, and 4: stable codes during step 1 and settling behavior during steps 2 and 4. However, step 3 exhibits settling behavior even though the OWCS are disabled because the input capacitor charge accumulated during step 2 takes an equivalent amount of time to drain. The user can employ the techniques outlined in Section 2.3.2.4 to avoid or reduce this settling behavior.

2.3.2.4 Using the ADS125H18 Programmable Delay to Mitigate Settling Effects for Differential Inputs

The ADS125H18 integrates a programmable delay feature that can be used to wait for any external settling behavior to stabilize before the ADC starts converting. This delay occurs before the first conversion after a conversion START as well as at the beginning of every new sequence step when the ADC sequencer is active. The delay value can range from 0 to 65535 modulator periods, t_{MOD}, and is independently programmable per step.

For example, Equation 15 and Equation 16 calculate one t_{MOD} in microseconds when f_{MOD} = 12.8MHz and the number of t_{MOD} periods required for a 1ms delay, respectively:

$$t_{MOD}\ (\mu s) = \frac{1}{f_{MOD}} = \frac{1}{12.8\text{MHz}} = 0.078125\ \mu s \quad (15)$$

$$t_{MOD}\ (\text{periods}) = \frac{\text{Delay}\ (\mu s)}{tMOD\ (\mu s)} = \frac{1000\ \mu s}{0.078125\ \mu s} = 12800 \quad (16)$$

Figure 2-29 shows a logic analyzer capture of the ADS125H18 data ready (DRDY) signal in yellow. DRDY indicates when new data is ready to be clocked out of the ADC. The time between DRDY pulses measures the sample-to-sample conversion latency. The green box indicates when the ADS125H18 sequencer automatically transitions from Step 1 (OWCS disabled) and starts converting on Step 2 (OWCS enabled). The ADC uses the sinc4 filter, operates at ODR = 12.5kSPS, and has DELAY = 0 such that the first conversion latency is approximately 320µs highlighted in red. The blue box indicates that all subsequent conversions are available at 1 / ODR.

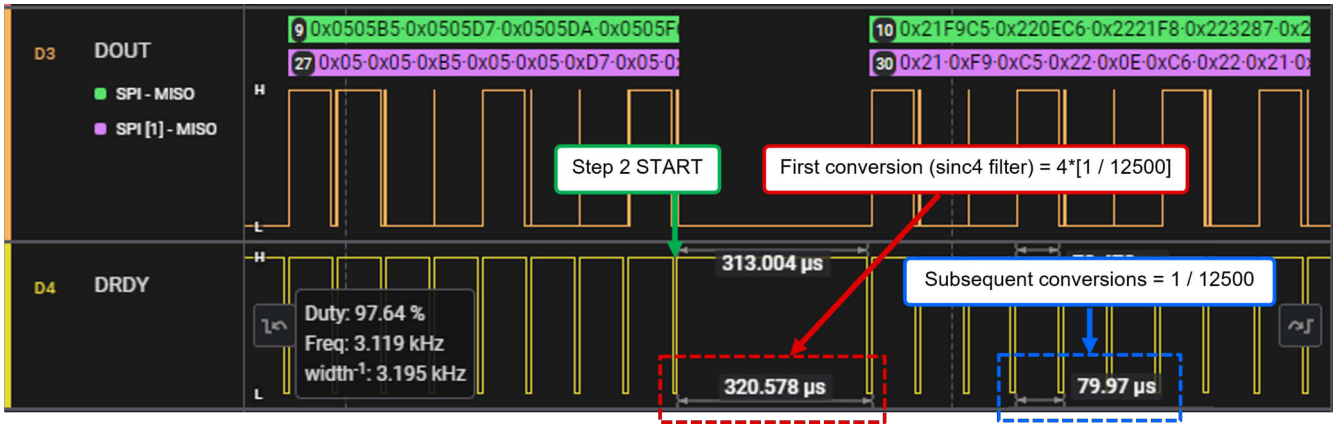

Figure 2-29. DS125H18 DRDY Timing with Programmable Delay = 0d (Differential Input, Sinc4 Filter, ODR = 12.5kSPS)

Comparatively, Figure 2-30 shows the same behavior as Figure 2-29 except that DELAY = 12800 (1ms). The red highlighted box indicates that the first conversion latency is one millisecond longer as a result. The subsequent conversion latency in blue remains unchanged.

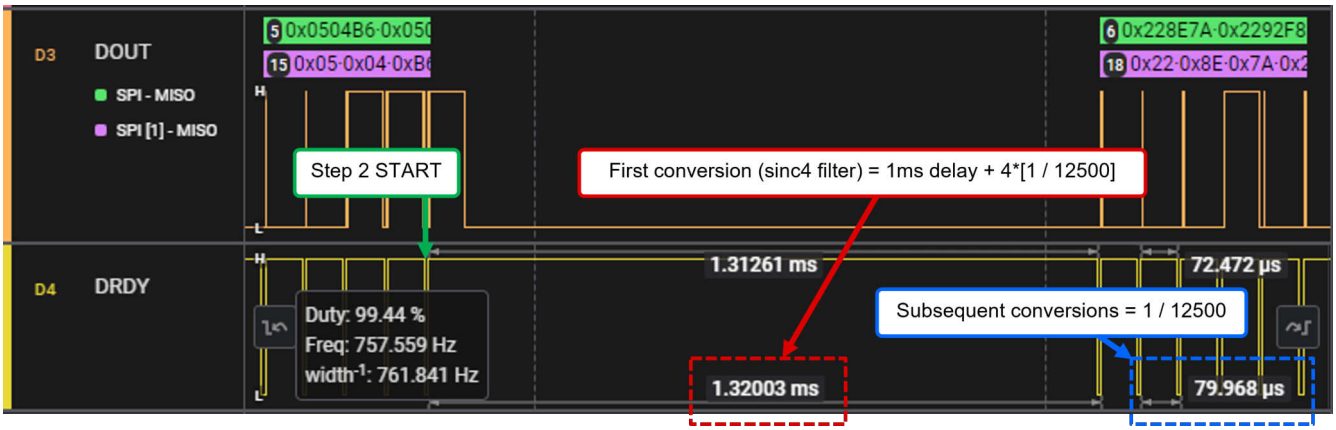

Figure 2-30. ADS125H18 DRDY Timing with Programmable Delay = 12800d (Differential Input, Sinc4 Filter, ODR = 12.5kSPS)

Importantly, the delay feature only postpones the ADC sampling process such that the sequencer immediately enables the OWCS feature at "Step 2 START". This delay provides additional time for the OWCS feature to settle before the ADC conversion process begins, resulting in higher-accuracy measurements. Table 2-15 compares the measured and predicted results when R_{SOURCE} = 800kΩ, ODR = 12.5kSPS, and input capacitance = 10nF for two DELAY values: 0ms and 1ms

Table 2-15. Measured $OWCS_{Delta}$ and Predicted R_{SOURCE_OWCS} for Programmable Delay = 0ms and 1ms (Differential Input, R_{SOURCE} = 800kΩ, ODR = 12.5kSPS, Input Capacitance = 10nF)

ODR	DELAY = 0ms			DELAY = 1ms		
	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)	Error	$OWCS_{Delta\_Diff}$	R_{SOURCE_OWCS} (kΩ)	Error
12.5kSPS	23.89%	323.28	60%	23.92%	384.00	52%

Table 2-15 shows that the programmable delay helps reduce the error between the actual and predicted R_{SOURCE} values by approximately 8%. This delay provides more stable measurements by enabling the ADC to capture less of the capacitor charging behavior and more of the steady-state behavior. Figure 2-31 shows this behavior by plotting the $OWCS_{Delta\_Diff}$ values point-by-point instead of the single value averaged over 512 conversions shown in Table 2-15. Figure 2-31 uses R_{SOURCE} = 800kΩ, ODR = 12.5kSPS, input capacitance = 10nF, and two different DELAY values: 0ms and 1ms

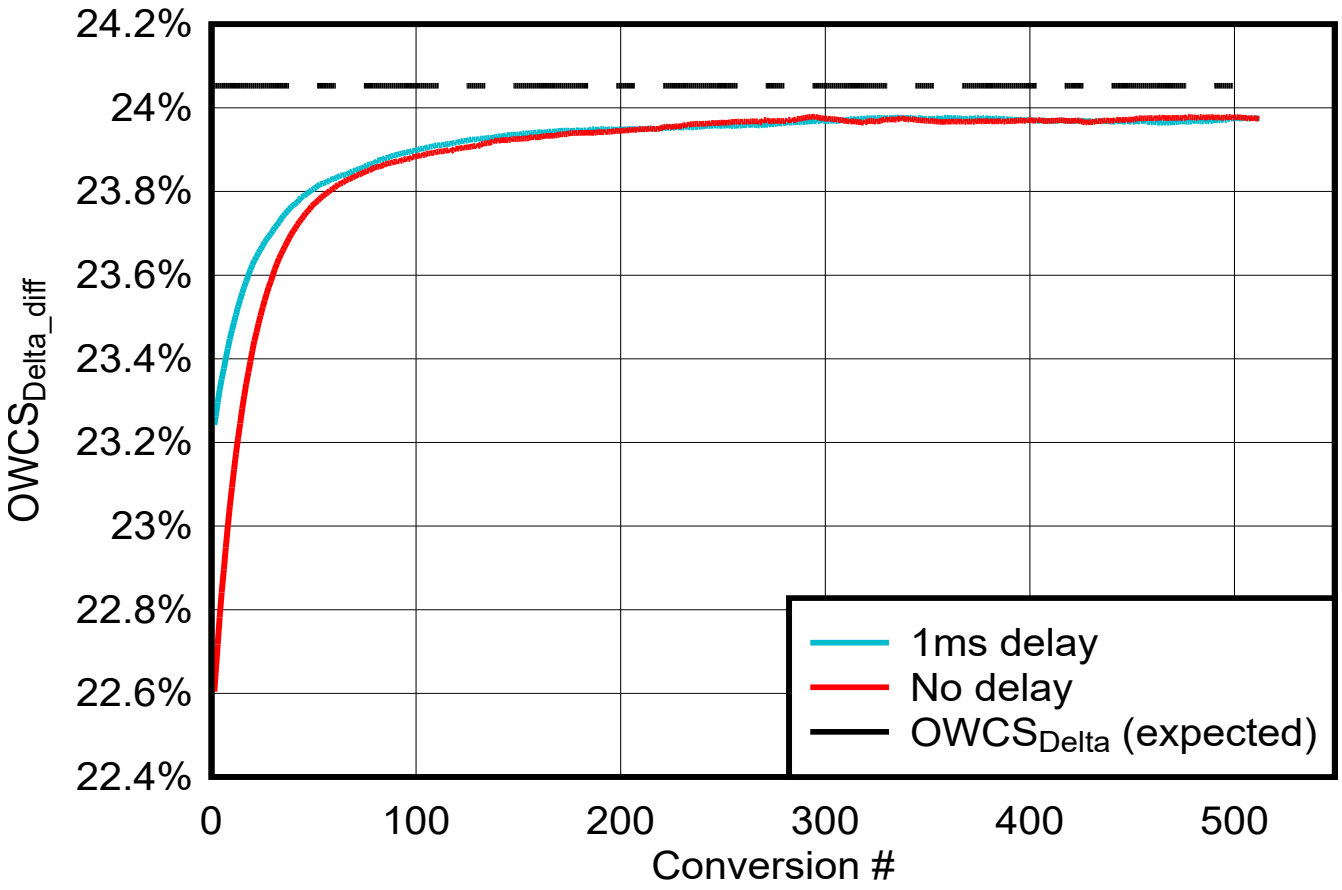

Figure 2-31. $OWCS_{Delta}$ Behavior when Programmable Delay = 0ms and 1ms (Differential Input, R_{SOURCE} = 800kΩ, Input Capacitance = 10nF, ODR = 12.5kSPS)

Figure 2-31 shows how the programmable delay avoids most of the initial capacitor charging behavior to yield an average $OWCS_{Delta}$ value closer to the expected. Additional delay could improve this behavior further. Finally, Figure 2-31 shows how a specific delay time affects the OWCS performance for a specific set of ODR and input capacitance parameters. A different delay value might be required if the user changes any of these system parameters.

2.4 Using the ADS125H18 Channel Sequencer to Implement an OWCS Algorithm

The ADS125H18 features a powerful channel sequencer that simplifies communication to and from the ADC. The sequencer uses multiple copies of the same page – or step – that enable the user to store up to 32 different (volatile) measurement configurations. Example parameters that are programmable on a per-step basis include different input channels, monitoring functions, output data rates, number of conversions, GPIO settings, conversion START delay, and many more. The sequencer automatically indexes to the next enabled step, which allows the ADS125H18 channel sequencer to greatly simplify the OWCS algorithm implementation by using the following guidelines:

- Enable the OWCS rarely – for example, at the end of each complete measurement cycle – to maximize the amount of time spent measuring the signals of interest
- Reserve two or four steps to perform open-wire detection on one single-ended or one differential input, respectively, to maximize the number of steps available for other measurements
- Choose consecutive steps for the OWCS measurements because the OWCS calculations rely on the input signal remaining constant when the OWCS are disabled and enabled. Additional delays between the OWCS-disabled measurements and OWCS-enabled measurement can lead to incorrect threshold calculations
- Scan through all input and monitoring steps in a single sequence. Scan one OWCS channel per sequence using the aforementioned two or four steps. Increment the OWCS step measurement channel after each sequence completes
- Operate the sequencer in single-sequence mode because the OWCS steps need to be reconfigured after each sequence

Figure 2-32 visualizes how to set up the ADS125H18 sequencer steps according to these guidelines:

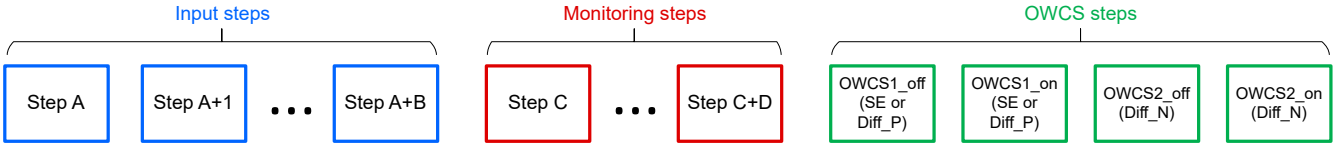

Figure 2-32. Recommended Sequencer Step Configuration to Implement the OWCS Algorithm

Figure 2-32 has several important features that are critical to implementing the OWCS algorithm:

- Ensure the step numbers increase monotonically from left to right because the ADS125H18 sequencer always measures the enabled steps in increasing order. In other words, the step number for the first input step ("Step A") should be less than next input step ("Step A+1), which should be less than "Step A+B", which should be less than the first monitoring step ("Step C"), and so on
- The steps do not have to be consecutive because the ADS125H18 sequencer only indexes through enabled steps. For example, the user could select input steps 1, 3, and 27. This behavior is also shown in the example in the next section
- Always set up the two or four OWCS steps shown in green at the end of the sequence
- A single-ended input uses only the two leftmost OWCS steps in green while a differential input uses all four OWCS steps
- Only apply the OWCS algorithm to input steps in blue. Figure 2-32 also shows monitoring steps in red that are useful for general diagnostics but cannot be used with the OWCS feature. Therefore, monitoring steps are not strictly required to implement the OWCS algorithm.

2.4.1 Example: Configuring the ADS125H18 Sequencer for the OWCS Algorithm

Figure 2-33 shows an example system that measures both input signals and monitoring signals as follows:

- Five input channels in blue, including a combination of single-ended and differential voltage and current inputs
- Three monitoring channels in red, including temperature, AVDD, and IOVDD

Moreover, Figure 2-33 highlights the ADS125H18 OWCS feature and associated multiplexer in green:

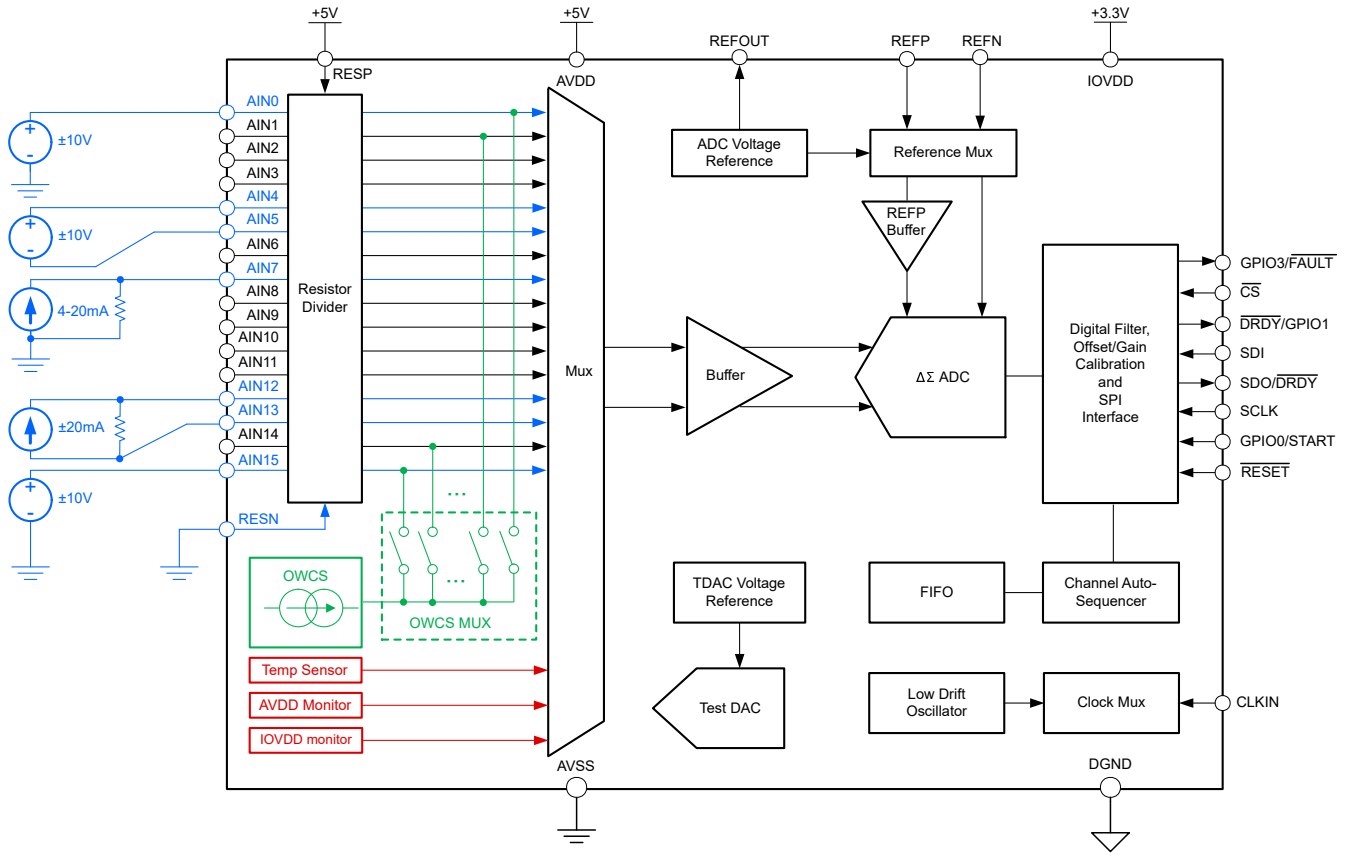

Figure 2-33. ADS125H18 Input Connections for the OWCS Algorithm Example

Figure 2-34 uses the format shown in Figure 2-32 to demonstrate how to setup the sequencer for the example system in Figure 2-33. All information shown in Figure 2-34 can be stored in the ADS125H18 sequencer and was arbitrarily chosen for this example.

Step #	4	7	10	11	13	16	21	22	25	26	27	28
Channel	AIN0-RESN	AIN4-AIN5	AIN7-RESN	AIN12-AIN13	AIN15-RESN	Temp sensor	AVDD	IOVDD	*tbd*	*tbd*	*tbd*	*tbd*
# of conv	6	8	8	24	10	3	3	3	32	32	12	12
OWCS	off	off	off	off	off	off	off	off	off	ON	off	ON

Figure 2-34. Example Sequencer Step for the OWCS Algorithm

Interpret Figure 2-34 as follows:

- Row 1 = the step number used for that step
- Row 2 = the channel measured by that step
- Row 3 = the number of conversions to measure on that step.
- Row 4 = if the OWCS are enabled (on) or disabled (off) for that step

For example, Step 13 measures ten conversions between inputs AIN15 and RESN. Note that the ADS125H18 sequencer allows the user to configure many other parameters on a per-step basis; however these parameters are not necessarily relevant to the OWCS algorithm and are not discussed further in this document.

Figure 2-34 also shows that the "Channel" for the OWCS steps in green are *to be determined*, or TBD. These settings are initially TBD because they change from one sequence to the next. However, the input steps and monitoring steps remain unchanged. Figure 2-35 shows how to configure the OWCS steps during the first sequence: the first and second OWCS steps (#25 and #26, respectively) are both configured with AIN0-RESN because this is the same channel used in the first sequence step (#4). Important settings for the first sequence step are highlighted in yellow.

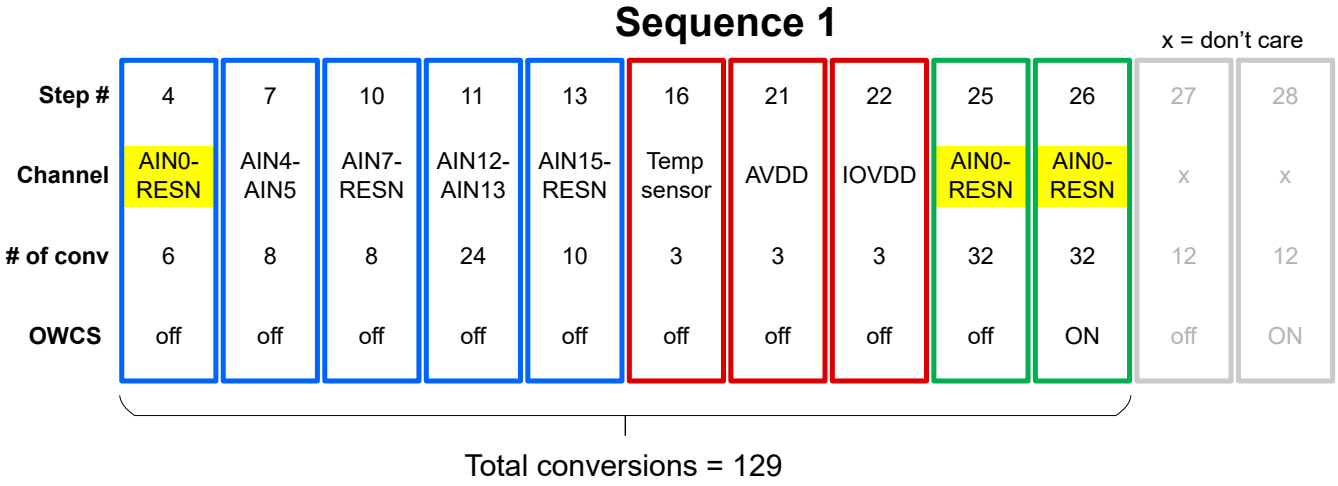

Figure 2-35. Example: Sequence 1 Characteristics

Start the sequencer by setting the START bit or toggling the START pin. The ADS125H18 sequencer indexes through each step in Figure 2-35 from left to right and measures 129 total conversions. The user must decide how to use the information received from each input and monitoring step, as well calculate the OWCS threshold.

Reconfigure the OWCS steps before the next sequence starts if necessary. Figure 2-36 shows how to configure the OWCS steps during the second sequence: the first and second OWCS steps (#25 and #26, respectively) are both configured with AIN4-RESN while the third and fourth OWCS steps (#27 and #28, respectively) are both configured with AIN5-RESN. These settings are required because the second step (#7) measures a differential input but the OWCS feature must always measure single-ended channels. Important settings for the second sequence step are highlighted in yellow.

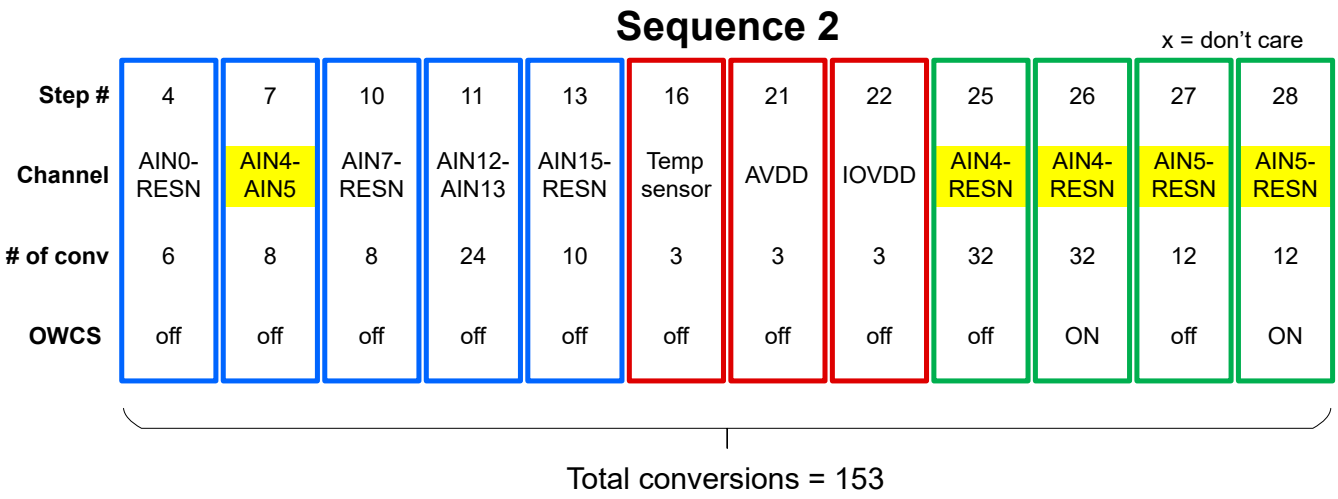

Figure 2-36. Example: Sequence 2 Characteristics

Again, start the sequencer by setting the START bit or toggling the START pin. The ADS125H18 sequencer indexes through each step in Figure 2-36 from left to right and measures 153 total conversions because the last two OWCS steps are enabled in this sequence. The user must decide how to use the information received from the input and monitoring steps, as well as calculate the OWCS threshold for each single-ended input. Ideally, the

calculated OWCS result for AIN4-RESN should be identical to the OWCS result for AIN5-RESN because this is a differential measurement.

Figure 2-37 follows the same logic to show how to reconfigure the OWCS channels before starting each of the next three sequences such that all input step channels in blue are analyzed for wire breaks. Important settings per sequence are highlighted in yellow.

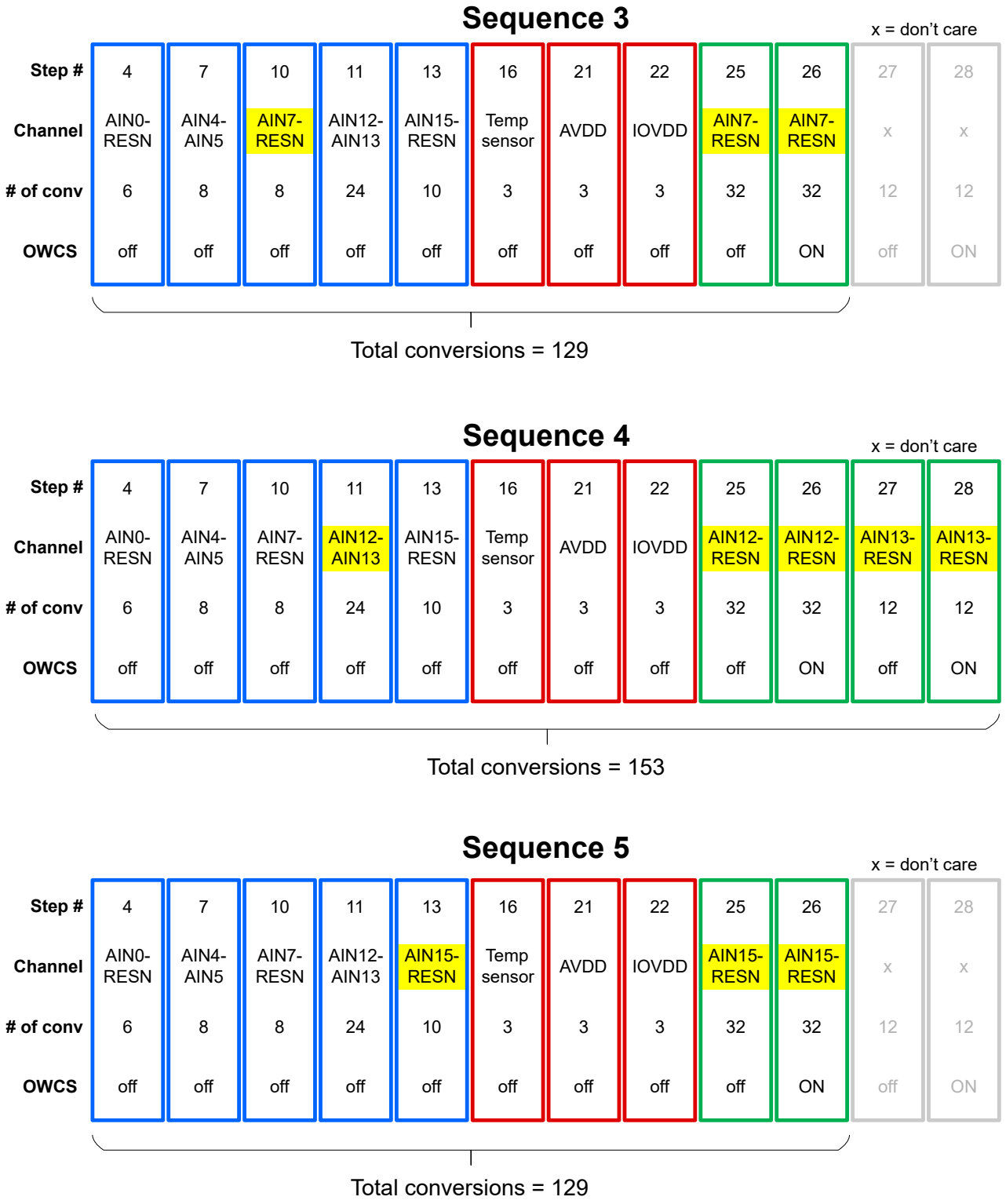

Figure 2-37. Example: Sequences 3 through 5 Characteristics

All input step channels have been diagnosed for wire breaks using the OWCS after sequence 5 completes. At this point, the user should reconfigure the OWCS channels to wrap back around to the first input step channel as shown in Figure 2-35, restarting the complete process. The user then decides how many total sequences to run as well as what to do if they detect a wire break.

2.4.2 Understanding the OWCS Algorithm in the ADS125H18 Example Code

The ADS125H18 product folder provides example code in the *Software development* section that was designed and tested using an ADS125H18EVM connect to an MSPM0G3507 Launchpad. Table 2-16 shows the hardware connections required between the ADS125H18EVM and the MSPM0 Launchpad that enable the code to work as intended:

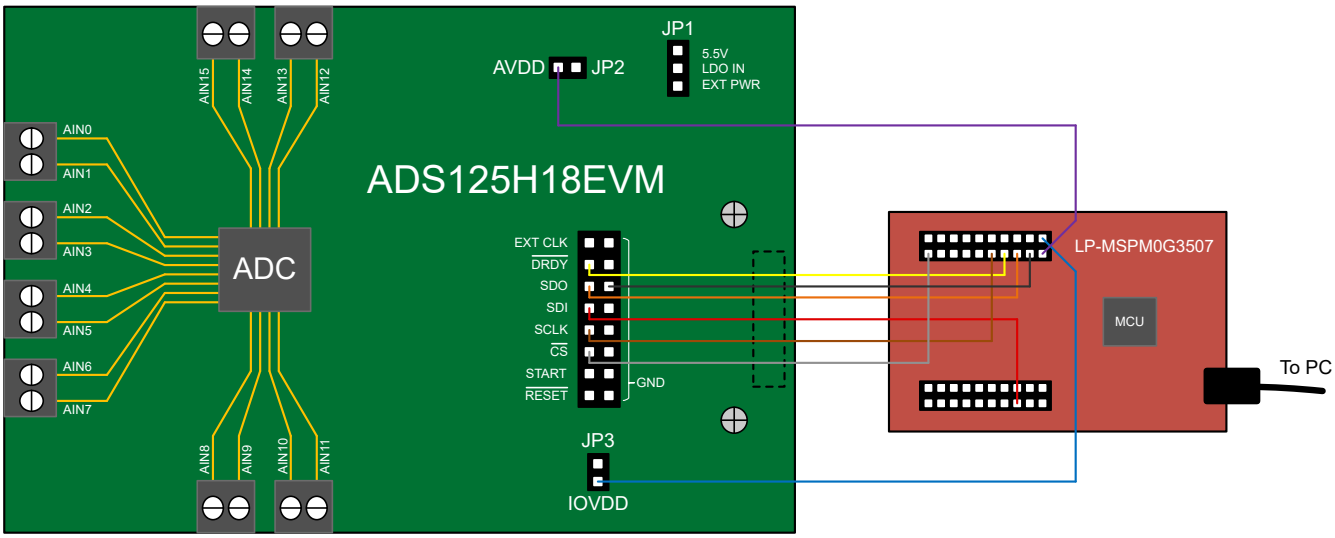

Figure 2-38. Connection Diagram Between the ADS125H18EVM and LP-MSPM0G3507

Table 2-16 summarizes the connections between the two boards:

Table 2-16. Pin Name and Pin Number Connections Between the ADS125H18EVM and LP-MSPM0G3507

ADS125H18EVM		LP-MSPM0G3507	
Pin name	**Pin number**	**Pin name**	**Pin number**
AVDD	JP2-2	5V	J3-21
IOVDD	JP3-1	3V3	J1-1
GND	J10-6	GND	J3-22
DRDY	J10-3	PA22	J3-24
SDO	J10-5	PB19	J3-23
SDI	J10-7	PB17	J2-18
SCLK	J10-9	PB18	J3-25
CS	J10-11	PA15	J3-30

The ADS125H18 example code includes an OWCS algorithm based on the guidelines from Section 2.4. Figure 2-39 shows a flow chart of the algorithm operation. This flow chart can be broken down into two general sections: *initialization* and *conversion + data processing*.

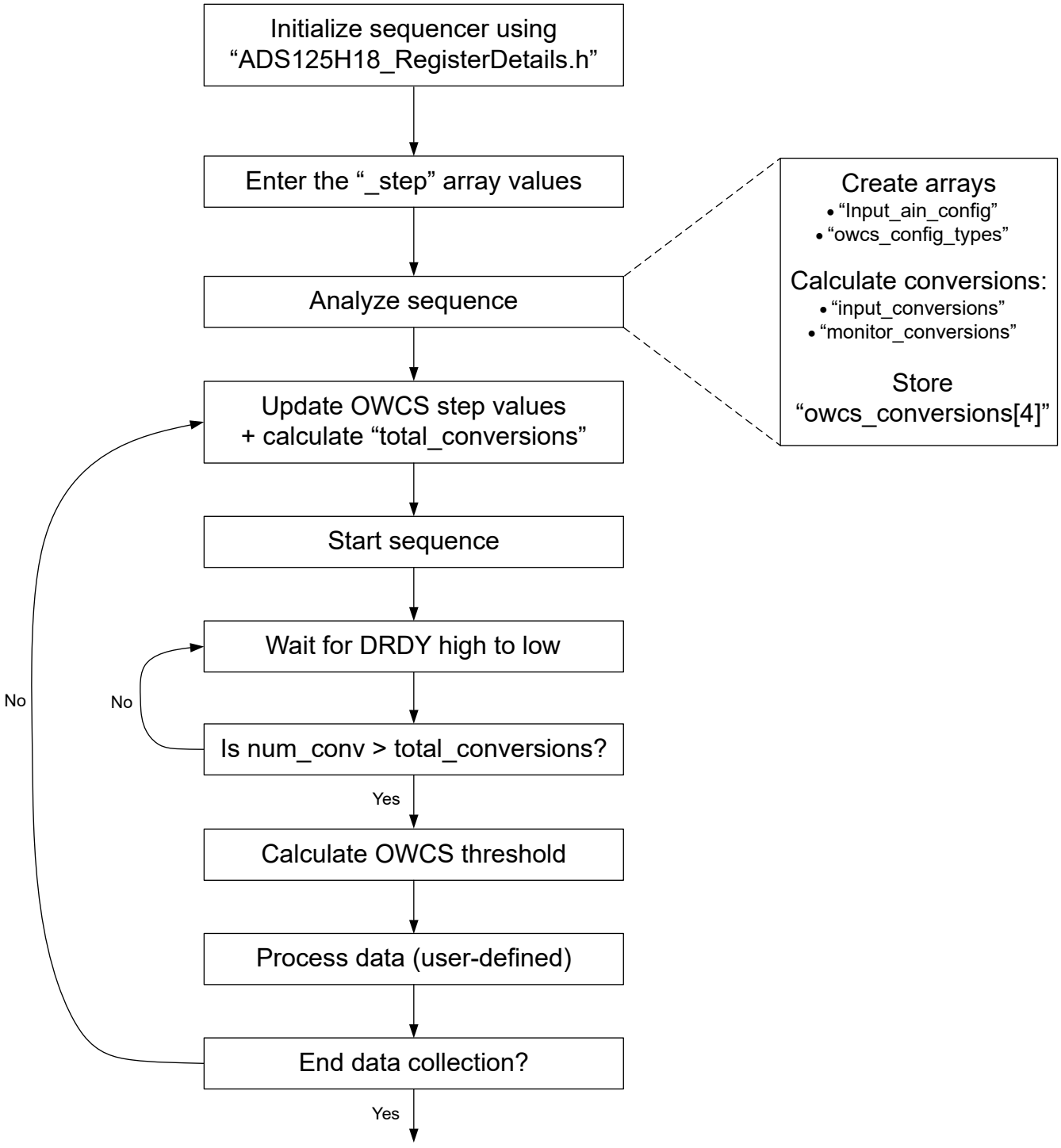

Figure 2-39. OWCS Algorithm Flow Chart

The subsequent sections provided detail the user requirements and algorithm operation for each step in the flow chart.

2.4.2.1 Initialization

The OWCS algorithm requires some information from the user that must be analyzed and stored for the remaining code to execute correctly:

- Initializing the sequencer using the "ADS125H18_RegisterDetails.h" file (user)
- Populating the "_step" arrays (user)
- Analyzing the sequence using the retrieveStepConfig() function (algorithm)

The following subsections describe these actions in more detail:

2.4.2.1.1 Initialize the sequencer

Locate the *Driver* folder after importing the example code project into the desired integrated development environment (IDE). Figure 2-40 shows the contents of the *Driver* folder using Texas Instruments Code Composer Studio IDE. The "ADS125H18_RegisterDetails.h" file is highlighted.

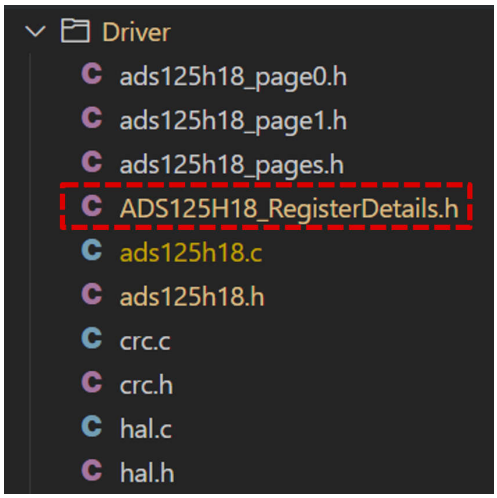

Figure 2-40. Driver Folder in the OWCS Code Project

Ensure that the "ADS125H18_RegisterDetails.h" file includes the proper configuration settings for the desired application. The user can enter these settings manually into the file or use the online *Sequencer Configuration Tool* shown in Figure 2-41. The *Sequencer Configuration Tool* is part of the ADC-DESIGN-CALC located in the *Design tools & simulation* section of the ADS125H18 product folder

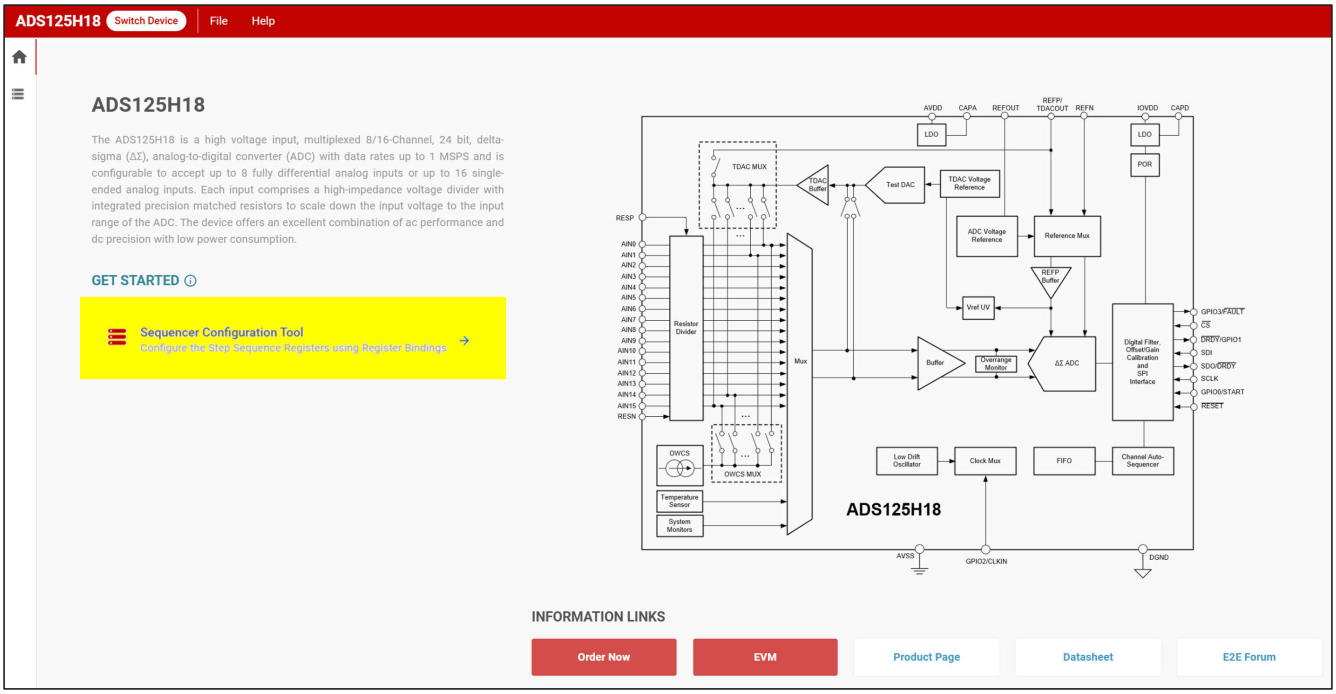

Figure 2-41. ADS125H18 Online Calculator Home Page

The *Sequencer Configuration Tool* guides the user through selecting the ADS125H18 sequencer settings. Figure 2-42 shows an example of the "Configure the Clock" page in the tool that allows the user to select an internal or external oscillator, the ADC speed mode, and the external oscillator properties when that option is selected.

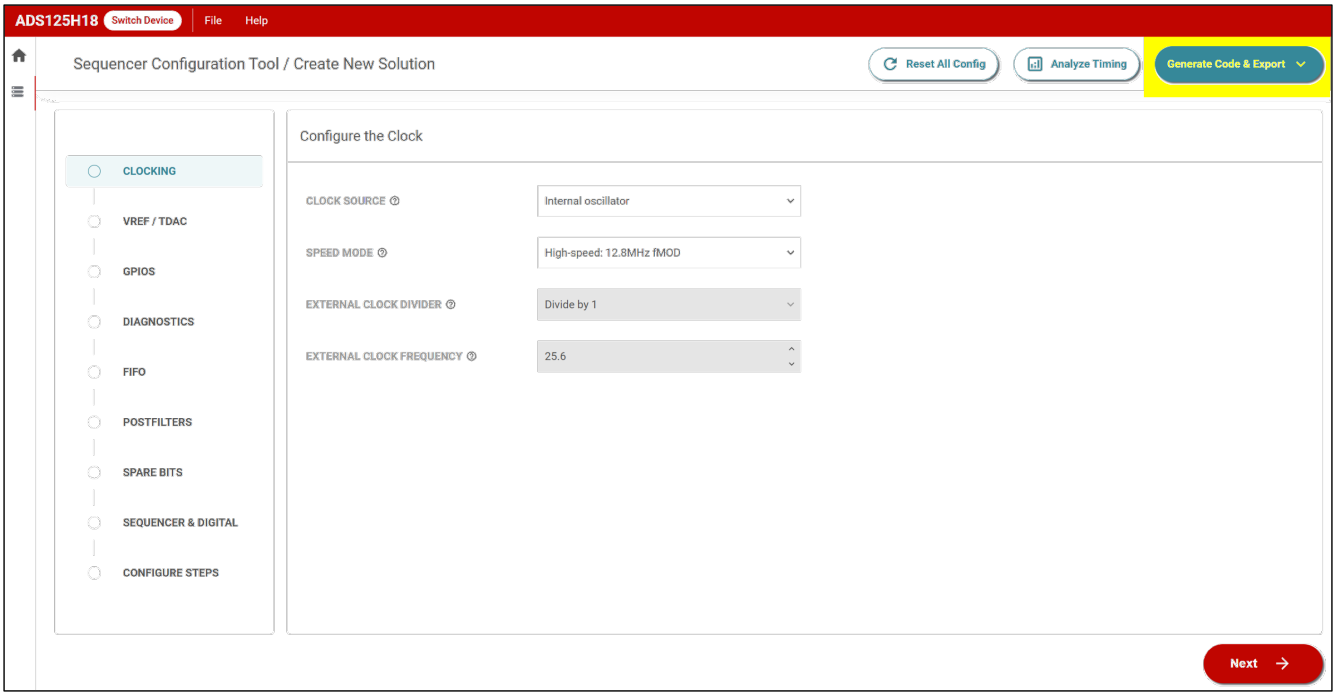

Figure 2-42. Sequencer Configuration Tool in the ADS125H18 Online Calculator

Figure 2-42 also shows how the user can export all selected settings using the "Generate Code & Export" button in the top right. Clicking this button generates a header file for either all register settings or only the non-default register settings. Replace the contents of the "ADS125H18_RegisterDetails.h" file with the generated header file output to initialize the sequencer.

2.4.2.1.2 Enter the "_step" Array Values

The user must enter the correct values into the "input_steps[]", "monitor_steps[]", and "owcs_steps[]" arrays. The values entered into these arrays must match the steps programmed in the previous section. For example, set input_steps[] = {0, 2, 4} if step 0, step 2, and step 4 are programmed as the input channel steps in the "ADS125H18_RegisterDetails.h" file. Also recall that the monitoring steps are not strictly required to implement the OWCS algorithm such that monitor_steps[] can be an empty array, assuming there are no monitoring steps enabled in the "ADS125H18_RegisterDetails.h" file.

Figure 2-43 shows these arrays in the code. Specifically, Figure 2-43 shows how to enter the array values to match the sequencer example shown in Figure 2-34.

```
// Define step arrays for different measurement types
// Input step#, not page# e.g. for step 0, input "0" even though step 0 is on page 1
// monitor_steps[] can be an empty array. Input_steps[] and owcs_steps[] arrays should be entered exactly as in RegisterDetails.h
uint8_t input_steps[] = {4, 7, 10, 11, 13};
uint8_t monitor_steps[] = {16, 21, 22};
uint8_t owcs_steps[] = {25, 26, 27, 28};    // always use four steps for OWCS -> first pair is for single-ended
                                            // measurements only, first and second pairs are for differential measurements
```

Figure 2-43. Defining the Step Arrays in the OWCS Code

2.4.2.1.3 Analyze the Sequence

The retrieveStepConfig() function analyzes the information in the "ADS125H18_RegisterDetails.h" file to determine the sequence properties. Specifically, the algorithm must identify and store:

- The input channel for each input step ("input_ain_config[]")
- Whether each input step measures a single-ended or differential input ("owcs_config_types[]")
- The total number of conversions for all input steps ("input_conversions")
- The total number of conversions for all monitoring steps ("monitor_conversions")
- The number of conversions for each OWCS step ("owcs_conversions[4]")

The algorithm uses this information to reconfigure the OWCS steps every time a new sequence starts

2.4.2.2 Starting the Sequencer and Processing Data

The OWCS algorithm starts ADC conversions as well as collects and processes data after device initialization using the processOWCSMeasurements() function. This function performs several operations:

- Writing to the OWCS steps to change to the next input channel
- Recalculating the total number of conversions
- Starting the sequencer
- Taking the appropriate number of conversions
- Processing the OWCS step data to return a predicted R_{SOURCE_OWCS} value

The following subsections describe these operations in more detail:

2.4.2.2.1 Update OWCS Step Values and Calculate "total_conversions"

The first step in the processOWCSMeasurements() function requires setting the first two OWCS steps to the appropriate input channel and calculating the number of conversions for that sequence. Figure 2-44 shows two register writes – one for the first OWCS step (OWCS disabled) and one for the second OWCs step (OWCS enabled) – and how the "total_conversions" variable is calculated assuming the OWCS operates on a single-ended input.

```
// Update channels on first and second OWCS steps to next value every time
// add one to "page" because page 0 is the global page so step0 is actually page 1, etc.
writeSingleRegister(user_steps.owcs_steps[0] + 1, 0x00, step_result->input_ain_cfg[ain_index]);
writeSingleRegister(user_steps.owcs_steps[1] + 1, 0x00, step_result->input_ain_cfg[ain_index]);

// calculate the total conversions
// if the input channel is differential then we also need to add the third and fourth OWCS conversions to the total
uint16_t total_conversions = step_result->input_conversions + step_result->monitor_conversions + step_result->owcs_conversions[0] + step_result->owcs_conversions[1];

if (step_result->owcs_config_types[ain_index] == DIFFERENTIAL){
    total_conversions = total_conversions + step_result->owcs_conversions[2] + step_result->owcs_conversions[3];

    // update channels on the third and fourth OWCS channels if DIFFERENTIAL
    // add one to "page" because page 0 is the global page so step0 is actually page 1, etc.
    // add one to "data" because we are only storing the AINP value for a differential measurement
    //      for example, if AIN = AIN8-AIN9, we only store AIN8, but we know the other channel must be AIN9
    writeSingleRegister(user_steps.owcs_steps[2] + 1, 0x00, (step_result->input_ain_cfg[ain_index] + 1));
    writeSingleRegister(user_steps.owcs_steps[3] + 1, 0x00, (step_result->input_ain_cfg[ain_index] + 1));
```

Figure 2-44. Updating the OWCS Step Values and Calculating Total Conversions in the OWCS Code

Figure 2-44 also shows an "if" statement that includes two additional register writes – one for the third OWCS step (OWCS disabled) and one for the fourth OWCs step (OWCS enabled) – and a "total_conversions" recalculation when the OWCS operates on a differential input. Refer to Figure 2-36 to visually understand why this behavior is required.

Also note that the OWCS algorithm never disables the third and fourth OWCS steps; instead, the algorithm simply does not store the data from these steps when the OWCS operates on a single-ended input. The user can alter the algorithm to disable these steps when they are not in use.

2.4.2.2.2 Starting the Sequencer and Taking Data

The next step in the processOWCSMeasurements() function starts ADC conversions, waits for the data ready ($\overline{DRDY}$) pin to toggle from high to low, then captures each data into an array. Figure 2-45 shows how this function runs until the routine captures a number of conversions equal to "total_conversions" and stores the results in the "all_sequences_data" array.

```
/* Collect all ADC data directly into allocated buffer */
if (!collectADCData(total_conversions, all_sequences_data)) {
    free(all_sequences_data);
    return result;
}
```

Figure 2-45. Starting the Sequencer and Taking Data in the OWCS Code

Importantly, the collectADCData() function uses a data-capture interrupt that assumes the $\overline{\text{DRDY}}$ pin toggles high to low after each conversion. However, the ADS125H18 allows the user to set different $\overline{\text{DRDY}}$ behaviors using the DRDY_CFG bits in the SEQUENCER_CFG register. For example, the user can set the $\overline{\text{DRDY}}$ pin to toggle high to low after each sequence completes instead of each conversion. These non-default $\overline{\text{DRDY}}$ pin options are intended to be used with the ADS125H18 first-in, first-out (FIFO) buffer. Modify the OWCS algorithm accordingly if the a non-default $\overline{\text{DRDY}}$ pin option is selected in the "ADS125H18_RegisterDetails.h" file. Refer to the ADS125H18 datasheet for more information.

2.4.2.2.3 Processing the OWCS Data

The final step in the processOWCSMeasurements() function processes the OWCS data to determine the threshold value (delta) as well as predict R_{SOURCE_OWCS}. Figure 2-46 shows that this final step is broken into three main actions:

1. Calculate the average value of all data collected for each OWCS step using calculateStepAverage()
2. Calculate the OWCS delta using Equation 1
3. Calculate R_{SOURCE_OWCS} using the OWCSgetRsource() function

```
// Update channels on first and second OWCS steps to next value every time
// add one to "page" because page 0 is the global page so step0 is actually page 1, etc.
writeSingleRegister(user_steps.owcs_steps[0] + 1, 0x00, step_result->input_ain_cfg[ain_index]);
writeSingleRegister(user_steps.owcs_steps[1] + 1, 0x00, step_result->input_ain_cfg[ain_index]);

// calculate the total conversions
// if the input channel is differential then we also need to add the third and fourth OWCS conversions to the total
uint16_t total_conversions = step_result->input_conversions + step_result->monitor_conversions + step_result->owcs_conversions[0] + step_result->owcs_conversions[1];

if (step_result->owcs_config_types[ain_index] == DIFFERENTIAL){
    total_conversions = total_conversions + step_result->owcs_conversions[2] + step_result->owcs_conversions[3];

    // update channels on the third and fourth OWCS channels if DIFFERENTIAL
    // add one to "page" because page 0 is the global page so step0 is actually page 1, etc.
    // add one to "data" because we are only storing the AINP value for a differential measurement
    //      for example, if AIN = AIN8-AIN9, we only store AIN8, but we know the other channel must be AIN9
    writeSingleRegister(user_steps.owcs_steps[2] + 1, 0x00, (step_result->input_ain_cfg[ain_index] + 1));
    writeSingleRegister(user_steps.owcs_steps[3] + 1, 0x00, (step_result->input_ain_cfg[ain_index] + 1));
```

Figure 2-46. Processing the OWCS Data in the OWCS Code

The OWCSgetRsource() function uses a modified version of Equation 8 to predict the single-ended R_{SOURCE_OWCS} value from a measured OWCS threshold. Equation 17 uses fixed coefficients instead of the RB, RA, and I_{OWCS_NOM} terms because these are all known, constant quantities relative to the selected ADS125H18 variant (see Table 2-1):

$$R_{SOURCE\_OWCS\_SE} = \frac{OWCS_{Delta\_SE} \times A - B}{C - 2 \times OWCS_{Delta\_SE}} \quad (17)$$

Figure 2-47 shows the lookup table for the single-ended A, B, and C coefficients in the OWCS.h file as a function of the different ADS125H18 variants:

```
/* Single-ended variables (Rows: A, B, C, MaxRsource) */
static float SingleEnded_variables[4][3] = {
    /* V12          V20          V40 */
    {2625000.0f,    2500000.0f,    2375000.0f},    /* A_SE */
    {421875.0f,     562500.0f,     562500.0f},     /* B_SE */
    {0.375f,        0.5f,          0.5f},          /* C_SE */
    {10e9f,         10e9f,         10e9f}          /* MaxRsource */
};
```

Figure 2-47. Single-Ended Input Look-Up Table in the OWCS Code

Figure 2-47 includes a value for *MaxRsource* in the look-up table. This parameter avoids the "divide-by-zero" error when $C = 2 \times OWCS_{Delta\_SE}$ in Equation 17 and limits the output to 1GΩ. Figure 2-48 shows the simple

function to predict R_{SOURCE_OWCS} from the variables A, B, and C as well as the variable *value* corresponding to the measured OWCS threshold:

```
/* Retrieve variables for single-ended input calculations */
if (config == SINGLE_ENDED) {
    A = SingleEnded_variables[0][col_idx];
    B = SingleEnded_variables[1][col_idx];
    C = SingleEnded_variables[2][col_idx];
    max_rsource = SingleEnded_variables[3][col_idx];

    /* Check if there is a divider / 0 error such that the input impedance is infinite */
    if (value >= C / 2.0f) {
        return max_rsource;
    }

    // Calculate the rsource value in ohms for single-ended inputs
    // return 0.0f if the result is negative
    float result_se = (value * A - B) / (C - 2 * value);
    return (result_se < 0) ? 0.0f : result_se;
```

Figure 2-48. Predicting R_{SOURCE_OWCS} for a Single-Ended Input in the OWCS code

The same logic is true for differential inputs. Equation 18 uses fixed coefficients for the RB, RA, and I_{OWCS_NOM} terms in Equation 14 to predict the differential R_{SOURCE_OWCS} value from a measured OWCS threshold:

$$R_{SOURCE\_OWCS\_Diff} = \frac{OWCS_{Delta\_Diff} \times A - B}{C - 4 \times OWCS_{Delta\_Diff}} \quad (18)$$

Figure 2-49 shows the lookup table for the differential A, B, and C coefficients in the OWCS.h file as a function of the different ADS125H18 variants:

```
/* Differential variables (Rows: A, B, C, MaxRsource) */
static float Differential_variables[4][3] = {
    /* V12          V20             V40 */
    {10500000.0f,   10000000.0f,    9500000.0f},   /* A_Diff */
    {1828125.0f,    2375000.0f,     2312500.0f},   /* B_Diff */
    {0.75f,         1.0f,           1.0f},         /* C_Diff */
    {10e9f,         10e9f,          10e9f}         /* MaxRsource */
};
```

Figure 2-49. Differential Input Look-Up Table in the OWCS Code

Figure 2-49 includes a value for *MaxRsource* in the look-up table. This parameter avoids the "divide-by-zero" error when $C = 4 \times OWCS_{Delta\_Diff}$ in Equation 18 and limits the output to 1GΩ. Figure 2-50 shows the simple function to predict R_{SOURCE_OWCS} from the variables A, B, and C as well as the variable *value* corresponding to the measured OWCS threshold:

```
/* Retrieve variables for differential inputs calculations */
} else {
    A = Differential_variables[0][col_idx];
    B = Differential_variables[1][col_idx];
    C = Differential_variables[2][col_idx];
    max_rsource = Differential_variables[3][col_idx];

    /* Check if there is a divider / 0 error such that the input impedance is infinite */
    if (value >= C / 4.0f) {
        return max_rsource;
    }

    // Calculate the rsource value in ohms for differential inputs
    // return 0.0f if the result is negative
    float result_diff = (value * A - B) / (C - 4 * value);
    return (result_diff < 0) ? 0.0f : result_diff;
}
```

Figure 2-50. Predicting R_{SOURCE_OWCS} for a Differential Input in the OWCS Code

2.4.2.2.4 Using the OWCS Results

The OWCS algorithm requires the user to write code that determines how to use the data received from the ADC. Figure 2-51 shows the USER TODO statements in the code

```
/* collect and process the OWCS data */
while (1) {
    OWCSProcessResult OWCS_result = processOWCSMeasurements(step_result, user_steps, ain_index);

    // Reset the ain_index if we have reached the end of the input_ain_cfg array
    ain_index = (ain_index + 1) % user_steps.num_input_steps;

    /* USER TODO: Process rsource values (result->rsource_values[0], result->rsource_values[1]) */
    /* USER TODO: Process delta values (result->delta_values[0], result->delta_values[1]) */
    /* USER TODO: Process raw ADC data (result->all_sequences_data, result->total_samples) */

    free(OWCS_result.all_sequences_data);
}
```

Figure 2-51. User Instructions to Process the OWCS Results

For example, certain industrial systems might shut down and reset to a known state after detecting an open wire. Other systems might simply log this information so that action can be taken at a later time. It is left to the user to decide how to proceed.

Figure 2-51 also shows how the ain_index variable increments up to the number of input steps set by the user once the processOWCSMeasurements() function completes. The ain_index variable resets once the algorithm indexes through all elements in the input_ain_cfg[] array as described in Section 2.4.1 and after Figure 2-37.

3 Summary

Detecting wire breaks in modern industrial control systems presents a significant challenge for many designers. This application note describes the operation and verification of the open wire current source (OWCS) feature in the ADS125H18, a precision, 24-bit, 1MSPS delta-sigma (ΔΣ) analog-to-digital converter (ADC) designed for use in analog input modules. Key topics covered in the application note include:

1. Analyzing and verifying the performance of the OWCS feature for single-ended inputs
2. Analyzing and verifying the performance of the OWCS feature for differential inputs
3. A detailed dicussion of how to use the ADS125H18 channel sequencer to implement an OWCS algorithm
4. An introduction to the OWCS algorithm in the ADS125H18 example code
5. How the OWCS behavior changes in a dual, redundant system

Use the information in this document to design a robust, reliable analog input module using the ADS125H18 and its unique open-wire detection feature.

4 References

1. Texas Instruments, ADS125H18, product page for the ADS125H18 device
2. Texas Instruments, ADS125H18EVM-PDK, product page for the ADS125H18 evaluation module

5 Appendix A

5.1 Using OWCS with Redundant Systems

Some industrial applications require a redundant configuration where multiple input modules connect to the same input signal. Figure 5-1 shows an example of a redundant system where two independent modules using the ADS125H18 measure the same single-ended input.

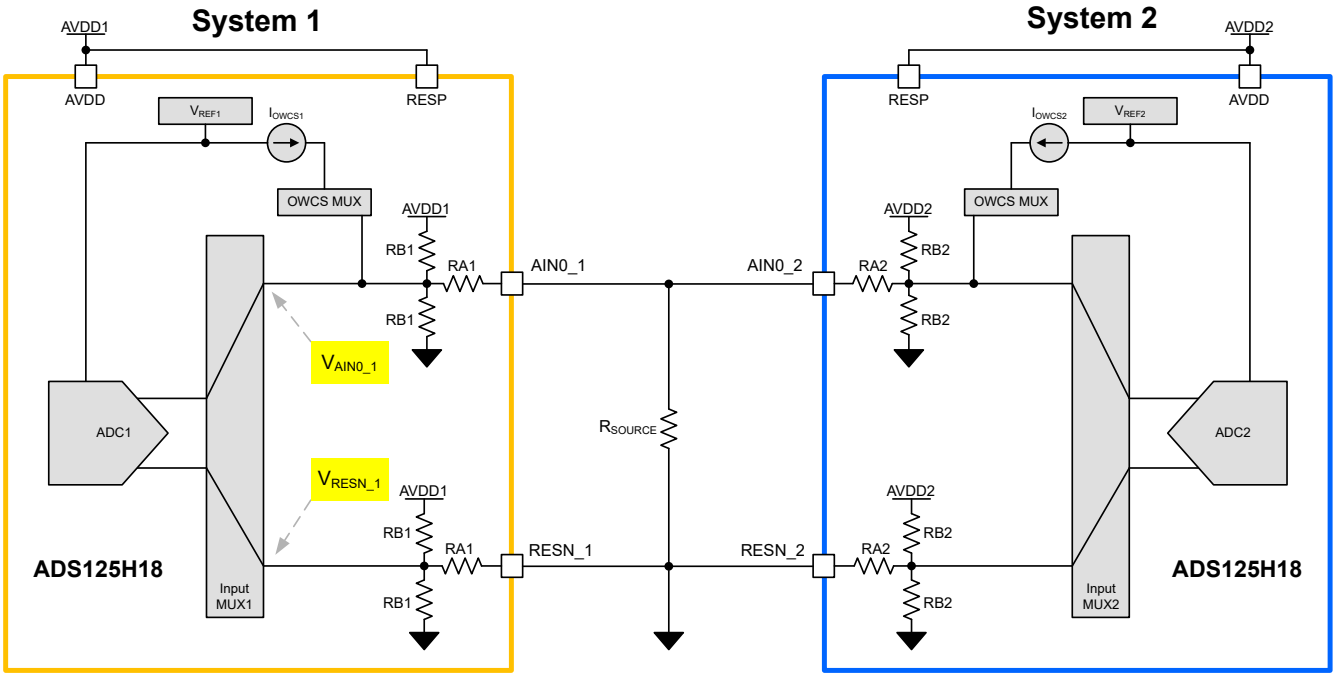

Figure 5-1. Using 2x ADS125H18 and 2x OWCS in a Dual, Redundant System (Single-Ended Input)

Redundancy enables the user to compare measurements between two independent systems to confirm that both signal chains are operating properly. Redundancy might also be used to maintain continuous measurements during a fault by providing a backup for the main system. For example, the user might use System 1 in Figure 5-1 under normal conditions. The user switches to System 2 and powers down System 1 upon detecting an error. Then, the user troubleshoots System 1 while System 2 provides uninterrupted measurement data back to the controller.

However, the redundant system in Figure 5-1 requires some additional considerations when using the OWCS feature because the current sources must flow through additional input resistance. Figure 5-2 shows how both current sources should be enabled for a redundant system as well as how the current flows from each source through the ADS125H18 resistor divider, through R_{SOURCE}, and to ground. The thick red line indicates where the two current sources combine such that the current flowing through $R_{SOURCE} = 2*I_{OWCS}$.

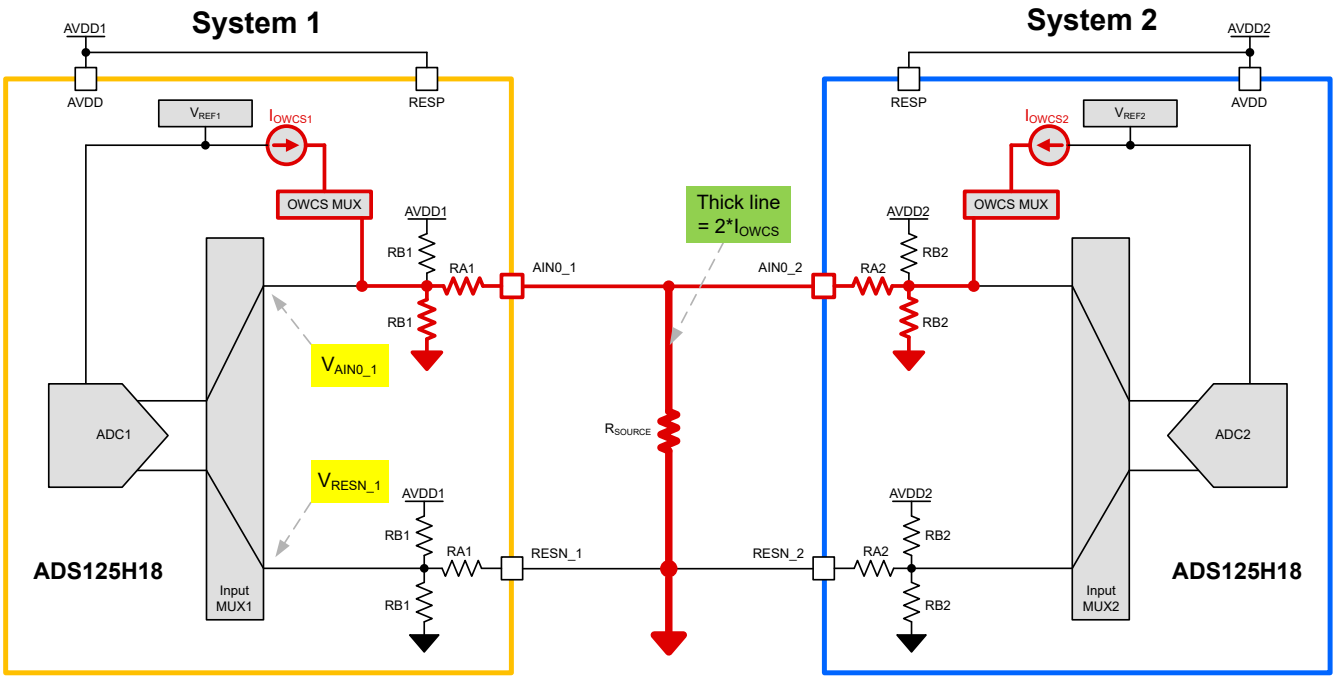

Figure 5-2. OWCS Current Path in a Dual, Redundant System (Single-Ended Input)

Fortunately, superposition and nodal analysis can also be used to analyze the redundant system shown in Figure 5-2. A dual, redundant system includes five sources that need to be analyzed independently: two voltage sources and one current source from System 1 as well as one voltage source and one current source from System 2. The user does not need to consider the RESN AVDD source in the idle system (2) because this source has no impact on the active system (1) measurement. However, this document forgoes a detailed analysis because this effort is just an extension of the principles described throughout this document. A detailed derivation is left to the user. Instead, this appendix only discusses OWCS behaviors across the following topics:

- Dual, redundant systems versus single-device systems
- Using 1x OWCS versus 2x OWCS in a dual, redundant system
- Error sources in dual, redundant systems
- Measuring differential inputs in a dual, redundant system
- Dual, redundant system summary

The subsequent sections use the V20 variant of the ADS125H18 as an example, though these results generally apply to all ADS125H18 device variants.

5.1.1 Dual, Redundant Systems versus Single-Device Systems

Figure 5-3 plots the value of $OWCS_{Delta\_SE}$ versus R_{SOURCE} for both the standard, single ADS12518 system using 1x OWCS from Figure 2-5 as well as the dual, redundant system using 2x OWCS from Figure 5-2:

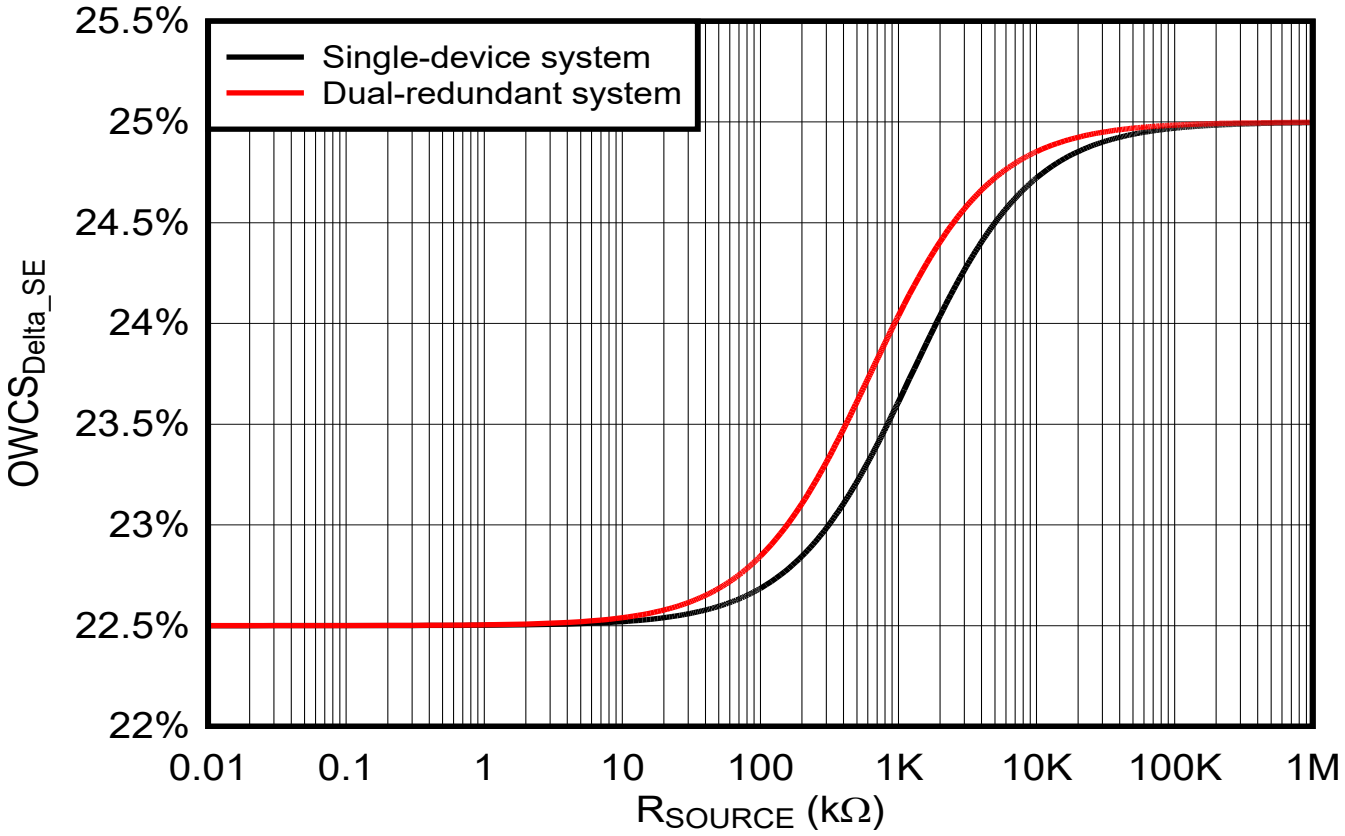

Figure 5-3. $OWCS_{Delta}$ Behavior for a Dual, Redundant System versus a Single-Device System (Single-Ended Input, ADS125H18V20)

The minimum and maximum limits are the same as the standard implementation using a single ADS125H18 with 1x OWCS. Refer to those limits in Table 2-3. However, the redundant system curve shifts slightly to the left in the R_{SOURCE} range from 10kΩ to 10MΩ. This shift results from the additional input resistance, current source, and voltage source from the ADS125H18 in System 2. It is therefore not possible to use Equation 8 to predict any arbitrary R_{SOURCE_OWCS} value from the measured $OWCS_{Delta\_SE}$ threshold in a dual, redundant system.

5.1.2 Using 1x OWCS versus 2x OWCS in a Dual Redundant System

Figure 5-4 demonstrates why the user should enable the OWCS in *both* ADS125H18 devices in a dual, redundant system instead of only enabling the OWCS in the *active* system.

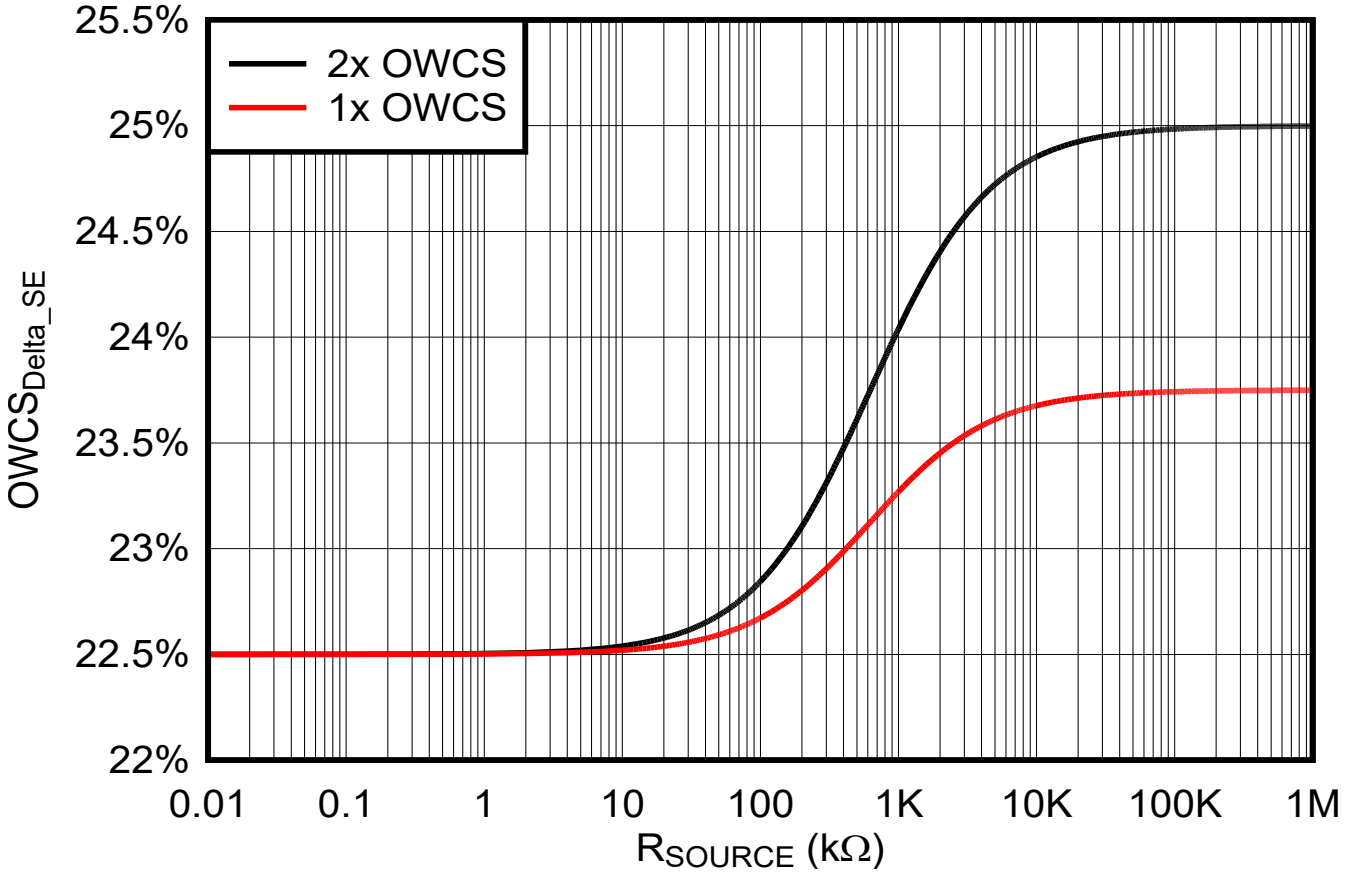

Figure 5-4. $OWCS_{Delta}$ Behavior Using 1x OWCS versus 2x OWCS in a Dual, Redundant System (Single-Ended Input, ADS125H18V20)

Figure 5-4 shows how using 1x OWCS reduces the maximum threshold limit by half compared to using 2x OWCS in a dual, redundant system. Table 5-1 summarizes the minimum and maximum $OWCS_{Delta\_SE}$ values for all ADS125H18 device variants using 1x OWCS versus 2x OWCS in a dual, redundant system.

Table 5-1. $OWCS_{Delta}$ Minimum and Maximum Values for 1x OWCS versus 2x OWCS in a Dual, Redundant System (Single-Ended Input, ADS125H18V20)

Variant	1x OWCS			2x OWCS		
	$OWCS_{Delta\_SE}$ min	$OWCS_{Delta\_SE}$ max	Difference	$OWCS_{Delta\_SE}$ min	$OWCS_{Delta\_SE}$ max	Difference
V12	16.07%	17.41%	1.34%	16.07%	18.75%	2.68%
V20	22.50%	23.75%	1.25%	22.50%	25.00%	2.50%
V40	23.68%	24.34%	0.66%	23.68%	25.00%	1.32%

Table 5-1 clearly indicates that the user should enable the OWCS in both ADS125H18 devices in a dual, redundant system to maximize the $OWCS_{Delta\_SE}$ range.

5.1.3 Error Sources in Dual, Redundant Systems

The redundant response in Figure 5-2 assumes nominal conditions such that all corresponding elements in System 1 are identical to System 2. For example, RA1 = RA2, RB1 = RB2, and VREF1 = VREF2 such that IOWCS_1 = IOWCS_2. However, each of these elements has some tolerance that affects the system behavior. Specifically, the input resistor divider elements can vary ±15% from device to device while the VREF has an initial tolerance of ±0.1% that affects the OWCS output. This document demonstrates how each component independently impacts the $OWCS_{Delta\_SE}$ range and assumes nominal conditions for all other components.

5.1.3.1 RA1 and RB1 Errors

Figure 5-5 plots the effect that a ±15% RA1 and RB1 resistance variation has on a dual, redundant system assuming RA2, RB2, VREF1, and VREF2 maintain their nominal values:

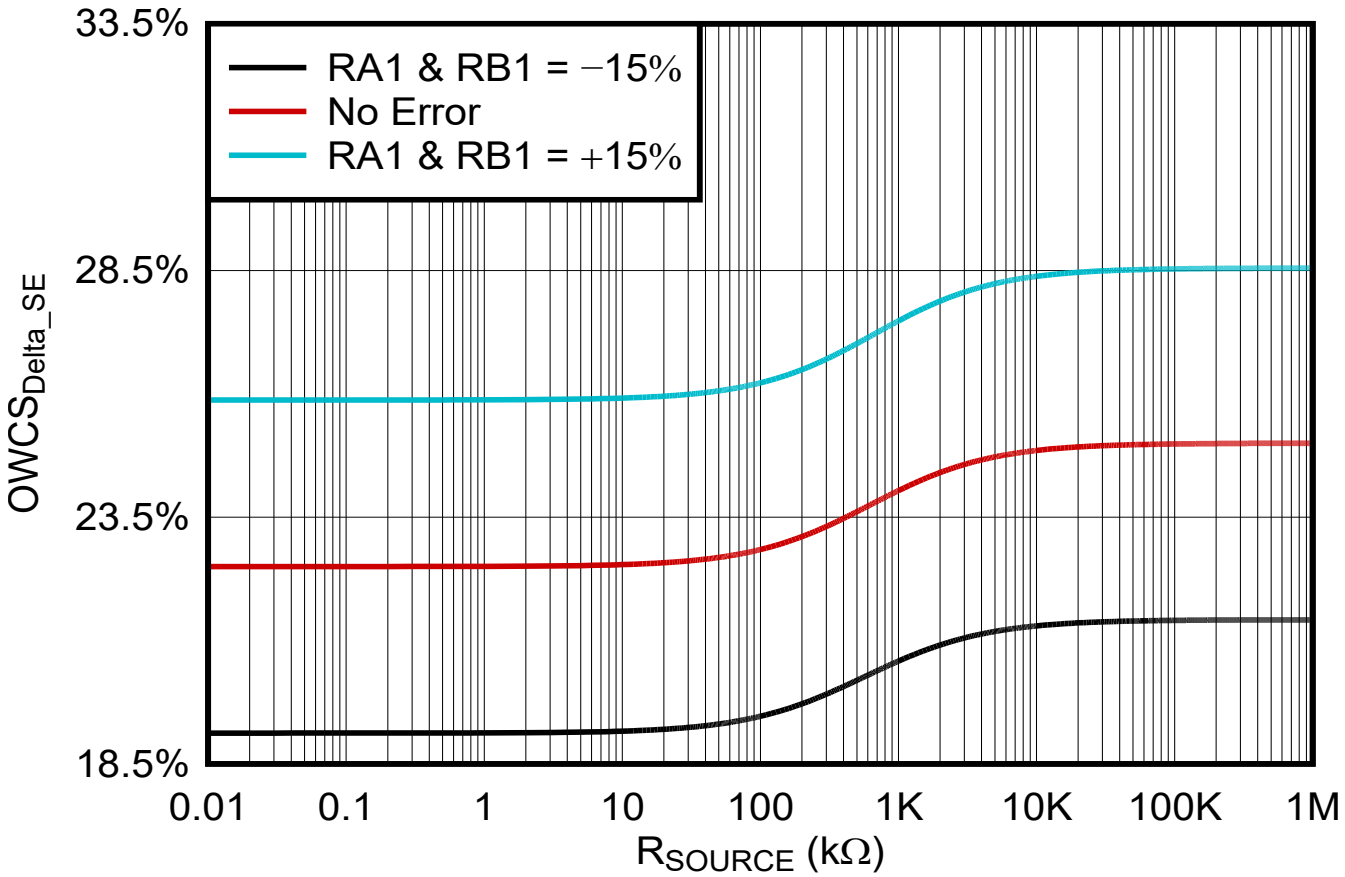

Figure 5-5. $OWCS_{Delta}$ Behavior due to RA1 and RB1 Errors in a Dual, Redundant System (Single-Ended Input, ADS125H18V20)

Figure 5-5 shows that a positive RA1 and RB1 resistance variation causes a significant positive shift (offset) in the threshold curve while a negative resistance variation causes a significant negative shift. However, Table 5-2 reveals that only a small difference exists between the minimum and maximum $OWCS_{Delta\_SE}$ values despite the large offset. In other words, varying the RA1 and RB1 resistors causes a large shift in the $OWCS_{Delta\_SE}$ *values* but a negligible change in the $OWCS_{Delta\_SE}$ *range*.

Table 5-2. $OWCS_{Delta}$ Minimum and Maximum Values Due to RA1 and RB1 Errors in a Dual, Redundant System (Single-Ended Input, ADS125H18V20)

RA1 & RB1 variation	$OWCS_{Delta\_SE}$ (min)	$OWCS_{Delta\_SE}$ (max)	Difference
-15%	19.13%	21.42%	2.30%
Nominal	22.50%	25.00%	2.50%
+15%	25.88%	28.55%	2.67%

5.1.3.2 RA2 and RB2 Errors

Figure 5-6 plots the effect that a ±15% RA2 and RB2 resistance variation has on a dual, redundant system assuming RA1, RB1, VREF1, and VREF2 maintain their nominal values:

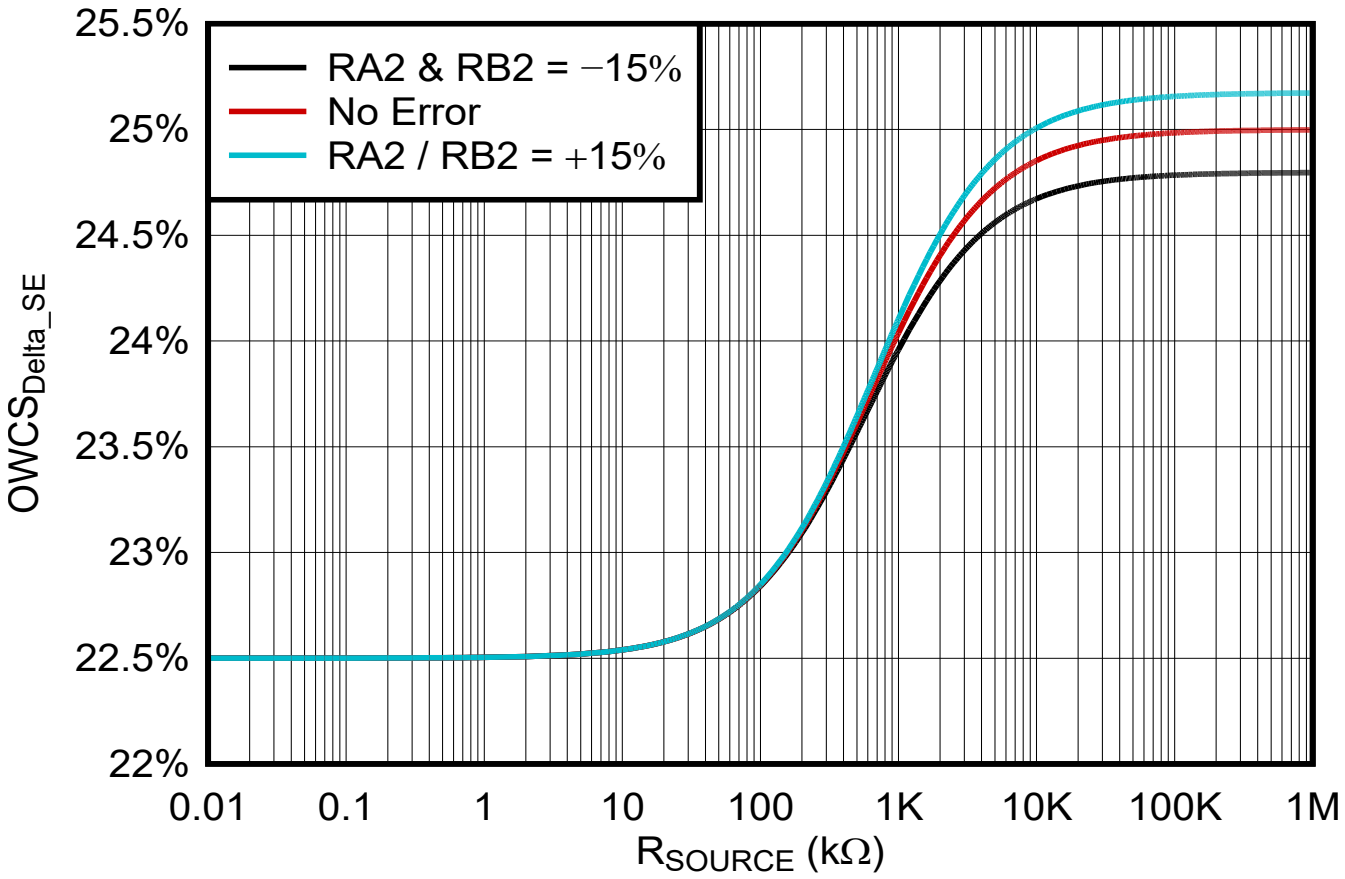

Figure 5-6. $OWCS_{Delta}$ Behavior due to RA2 and RB2 Errors in a Dual, Redundant System (Single-Ended Input, ADS125H18V20)

Figure 5-6 shows how variation in resistors RA2 and RB2 noticeably impacts the $OWCS_{Delta\_SE}$ value only when R_{SOURCE} is large (>1MΩ). This behavior occurs because the AIN0_1 node in Figure 5-2 shorts to ground as R_{SOURCE} approaches 0Ω. Therefore, the AVDD2 and I_{OWCS2} sources have no effect on the system until R_{SOURCE} is sufficiently large. Table 5-3 provides the minimum and maximum $OWCS_{Delta\_SE}$ values from Figure 5-6:

Table 5-3. $OWCS_{Delta}$ Minimum and Maximum Values Due to RA2 and RB2 Errors in a Dual, Redundant System (Single-Ended Input, ADS125H18V20)

RA2 & RB2 variation	$OWCS_{Delta\_SE}$ (min)	$OWCS_{Delta\_SE}$ (max)	Difference
-15%	22.50%	24.80%	2.30%
Nominal	22.50%	25.00%	2.50%
+15%	22.50%	25.17%	2.67%

5.1.3.3 VREF1 and VREF2 Errors

Figure 5-7 plots the effect that a ±0.1% VREF1 and VREF2 voltage variation has on a dual, redundant system assuming RA1, RB1, RA2, and RB2 maintain their nominal values.

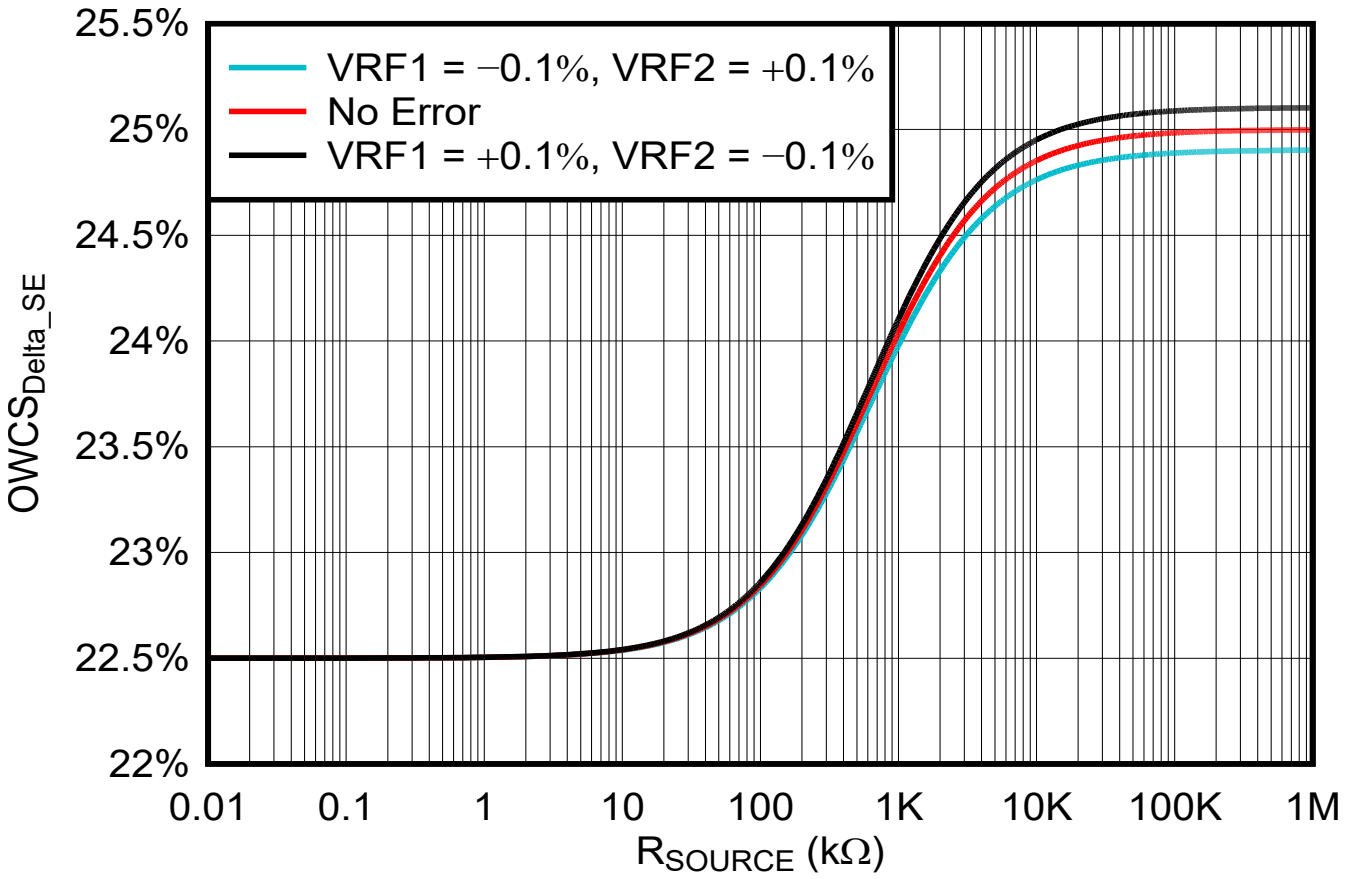

Figure 5-7. $OWCS_{Delta}$ Behavior due to VREF Error in a Dual, Redundant System Measuring (Single-Ended Input, ADS125H18V20)

Figure 5-7 shows how variation in voltages VREF1 and VREF2 noticeably impacts the $OWCS_{Delta\_SE}$ value when R_{SOURCE} is large (>1MΩ), similar to Figure 5-6. This behavior occurs because the I_{OWCS1} current is effectively constant across R_{SOURCE} and the I_{OWCS2} current source generates a negligible voltage contribution at AIN0_1. Therefore, a small change in either VREF1 or VREF2 yields a small change in the $OWCS_{Delta\_SE}$ values. Table 5-4 provides the minimum and maximum $OWCS_{Delta\_SE}$ values from Figure 5-7:

Table 5-4. $OWCS_{Delta}$ Minimum and Maximum Values Due to VREF Errors in a Dual, Redundant System (Single-Ended Input, ADS125H18V20)

VREF1 and VREF2 variation	$OWCS_{Delta\_SE}$ (min)	$OWCS_{Delta\_SE}$ (max)	Difference
VREF1 = 2.6V, VREF2 = 2.4V	22.50%	24.90%	2.40%
VREF1 = VREF2 = 2.5V	22.50%	25.00%	2.50%
VREF1 = 2.4V, VREF2 = 2.6V	22.50%	25.10%	2.60%

5.1.4 Measuring Differential Inputs in a Dual, Redundant System

This appendix has focused on describing dual, redundant system behavior while measuring *single-ended* inputs. However, Figure 5-8 shows that these systems can also measure *differential* inputs:

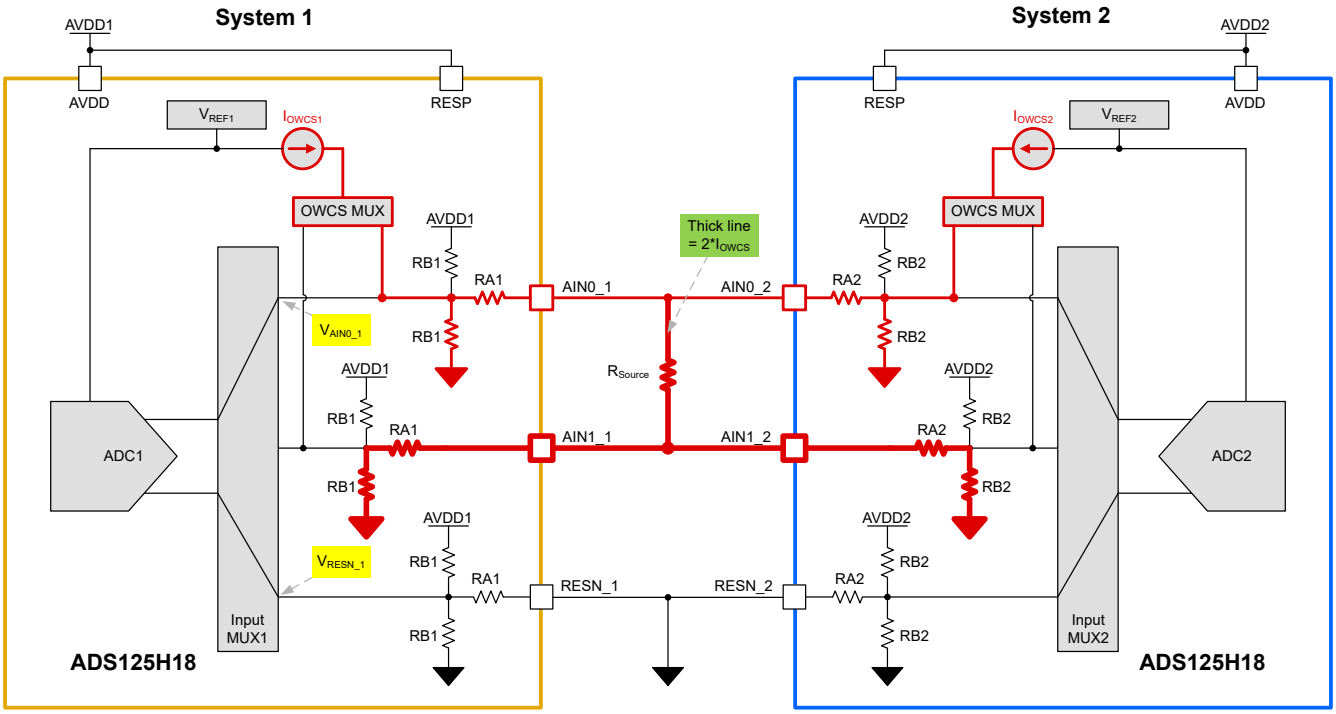

Figure 5-8. OWCS Current Path in a Dual, Redundant System (Differential Input)

Figure 5-8 indicates that the user should enable both current sources similar to the single-ended input example. Current flows from each source through the ADS125H18 positive input resistor divider (AIN0_x), through R_{SOURCE}, through the negative input resistor divider (AIN1_x), and to ground. The thick red line indicates where the two current sources combine such that the current flowing through R_{SOURCE} and is the negative input resistors dividers is $2*I_{OWCS}$.

A dual, redundant system measuring differential inputs results in the same general behaviors compared to a single-ended input. Therefore, no additional discussion is included in this document about differential inputs and any further analysis is left to the user.

5.1.5 Dual, Redundant System Summary

The following list summarizes important conclusions about dual, redundant systems:

- Enable both OWCS to provide the widest threshold range in a dual, redundant system. Both devices must therefore be powered and configured even if one device is not actually measuring the input
- Use ADS125H18s with the same voltage rating. For example, use 2x ADS125H18V20 devices for a dual, redundant system. Mixing different voltage ratings complicates the analysis and can lead to narrower threshold ranges
- Use the same reference voltage value for all devices to ensure the IOWCS current magnitudes are as close as possible. For example, set the internal reference voltage to 2.5V for both devices. If possible, use a single external voltage reference for all ADS125H18s to maintain ratiometric behavior
- The OWCS code uses Equation 17 and Equation 18 to predict the R_{SOURCE_OWCS} value from the measured $OWCS_{Delta}$ for single-ended and differential inputs, respectively. However, this section clearly shows that a dual, redundant system does not exhibit the same behavior as a system using a single ADS125H18 under any conditions. Therefore, the user must modify the OWCS code to work with redundant systems. It is recommended to generate an $OWCS_{Delta}$ lookup-table (LUT) based on the specific use case as well as a function to search the LUT for the corresponding R_{SOURCE_OWCS} value. Again, this action is left to the user to complete.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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 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, TI’s General Quality Guidelines, 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. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

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

Last updated 10/2025