FREQUENTLY ASKED QUESTIONS

Q1. What is the impedance range for the AD5940 high-speed loop?

A. The measurable impedance range = 0.1 Ohms <=Z<= 10M Ohms.
Ensure HSTIA RTIA selection that generates an output voltage of <±900mV with PGA Gain =1.
For example, if the expected input current max is 50uA, RTIA must be = 900mV/50uA = 18KOhm

Q2. I am implementing bootloader. During the implementation I need to know how to use below: "ReadProtectedKeyhash" and "NumCRC pages" in startup_ADuCM355.c. Could you let me know some detail manual, or examples?

A. This is part of the flash user space metadata.
For more details, you may check out
"User Space Metadata" section of Hardware Reference Manual of ADuCM355,
"Metadata Configuration" section of Hardware Reference Manual of ADuCM355,
"WRITE PROTECTION REGISTER (WRPROT)" and "SIGNATURE REGISTER" section of Hardware Reference Manual of ADuCM355.

Q3. In ADuCM355, is it possible to access the signal that is sent to the ADC?

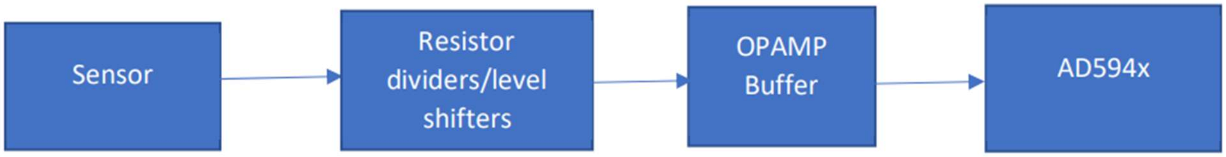
A. The ADC input can't be monitored directly. But you can use the ADC raw data , use ADC transfer function to obtain the input to ADC.

```
 uiIntSta = AfeAdcIntSta();
if (uiIntSta & BITM_AFE_ADCINTSTA_ADCRDY) {
szADCSamples[i]= AfeAdcRd(RAWADC); //Raw input to ADC is read
}
```

To calculate the input voltage (VIN), use the ADC Transfer function equation in AD5940/1 datasheet to convert ADC codes to a voltage.

Q4. Our application requires that the ADuCM355 can see >3.3V. What would you suggest for input signal limiting? I would normally use Zeners and resistors, but don't want to corrupt the AINx inputs with additional parasitics.

A. It is recommended to restrict the analog inputs to between 0.2V and 2.1V, to avoid saturation and erratic results. Resistor dividers along with OPAMP buffers may be used in the input path. It will not cause any problem.



Q5. How can we deinitialize the ad5940 in normal operation mode to deinit gracefully before turning LDO off?

A. Graceful stopping can be done by adding the line below before turning off the AFE blocks:
AppEDACtrl(APPCTRL_STOPSYNC);

Q6. In AD5940, how is the bandwidth and Rtia value calculated for LPTIA and HSTIA?

A. The BW and RTIA values are decided based on the following parameters:
LPTIA gain bandwidth product (GBW) is 20KHz. LPTIA current limit is 10mA.
HPTIA gain bandwidth product (GBW) is 30MHz. HPTIA ramp current limit is 3mA.
Note: GBW can be changed by changing the power modes.

Q7. How do we physically set the I2C address of each ADuCM355 in the system?

A. I2C0SIDCfg() function in M355_I2C_Slave code sets I2C slave address for ADuCM355.
M355 I2C slave can support 4 slave addresses. Eg: I2C0SIDCfg(0xA0,0xA6,0xA8,0xA0);

Q8. How to update the AFE parameters so that it gets updated after each measurement?

A. You may refer to

- AppAMPRegModify() Pull requests · analogdevicesinc/ad5940-examples (github.com)
  Which tells how to update the AFE parameters after each measurement.

Q9. In AD5940 examples, without the interrupt trigger read, dataprocess and datafifothresh, What should I do to have a fixed number of samples to 20Hz ?

A. To achieve this, you may change the firmware to run without using timers and sequences.

Q10. I want to use a display device such as an OLED(via I2C) with uCM355. Also planning to interface a SD card via SPI. Kindly suggest if there are any dedicated libraries for interfacing the above devices with ADuCM355.

A. we are not supporting third party hardware in our AD5940/aducm355 examples pack, and user can re-use low level I2C and SPI configuration and read/write functions, together with the device driver functions for the chosen OLED.

Q11. How to disable AD5940 watchdog timer?

A. /*TURNING OFF THE WATCHDOG*/

```
uint32_t wdtcon = AD5940_ReadReg(REG_AFEWDT_WDTCON);
wdtcon &= !BITP_AFEWDT_WDTCON_EN;
AD5940_WriteReg(REG_AFEWDT_WDTCON, wdtcon);
```

Watchdog timer is disabled in the AD5940_Initialize() function (defined in AD5940.c).

Q12. In AD5940_EDA example, is there an equation which is used to choose RTIA point from many sampled points (how is threshold fixed ?)

A. There is a feature implemented in the code to automatically select the optimum value of RTIA. Ensure RtiaAutoScaleEnable is set to bTRUE to enable the feature. Check out the EDARtiaAutoScaling() function to see how this feature is implemented.

Q13. what is the relationship between sampling rate, dft num, sin freq.

A. The SampleFreq defines how often the ADC makes a measurement. The DftNum parameter defines the number of ADC samples used to calculate the DFT. Increasing the DftNum will reduce ODR and decreasing DftNum will increase ODR. Sin Freq is the frequency of the excitation sine wave.

Q14. What is the ideal Sin frequency / Excitation frequency for doing eda measurements?

A. 100Hz is accepted as the best frequency for EDA measurement. If the frequency is higher the signal will penetrate too far below the skin and will not give an indication of the actual skin impedance.

Q15. How are range of impedances determined for EDA/BIA measurements ?
(valid range which gives accurate result)

A. Impedance range is determined by the allowed current through RTIA, that ensures that TIA doesn't reach saturation. Typical impedances for EDA measurements are from 20KOhms to 10MOhms.

Q16. In AD5940_EDA example, how to know the maximum allowed ODR for a given configuration.

A. You may add the code snippet below inside AppEDASeqMeasureGen() and add to watch "AppEDACfg.MaxODR", to know the maximum ODR allowed for the set configuration.

```
AD5940_SEQGenCtrl(bFALSE); /* Stop sequencer generator */
 error = AD5940_SEQGenFetchSeq(&pSeqCmd, &SeqLen);
 AppEDACfg.MeasSeqCycleCount = AD5940_SEQCycleTime();
 AppEDACfg.MaxODR = 1/(((AppEDACfg.MeasSeqCycleCount + 10) / 16.0)* 1E-6) ; //If Sysclk is16MHz
 if(AppEDACfg.EDAODR > AppEDACfg.MaxODR)
 {
 /* We have requested a sampling rate that cannot be achieved with the time it takes to acquire a
sample.*/
 AppEDACfg.EDAODR = AppEDACfg.MaxODR;
 }
```

Also, do the below modification in AD5940_SEQCycleTime() function:

```
uint32_t AD5940_SEQCycleTime(void)
{
 uint32_t i, Cycles, Cmd;
 Cycles = 0;
 for(i=0;i< SeqGenDB.SeqLen;i++)
 {
………..
```

Q17. Is there is power supply sequence for AD5940?

A. The is no power supply sequence for AD5940.

Q18. What are the states of GPIOs when AD5940 is in Hibernate mode

A. The GPIOs retain their last settings in hibernate mode.

Q19. In AD5940, what is the key difference between 4 wire and 2 wire impedance measurement?

A. In 2-wire impedance measurement example available in Github, connections are as shown in the picture below.
Current is measured across CE0 and AIN2.
Voltage is measured across CE0 and excitation buffer's N node.
The above two measurements are used to find the impedance.

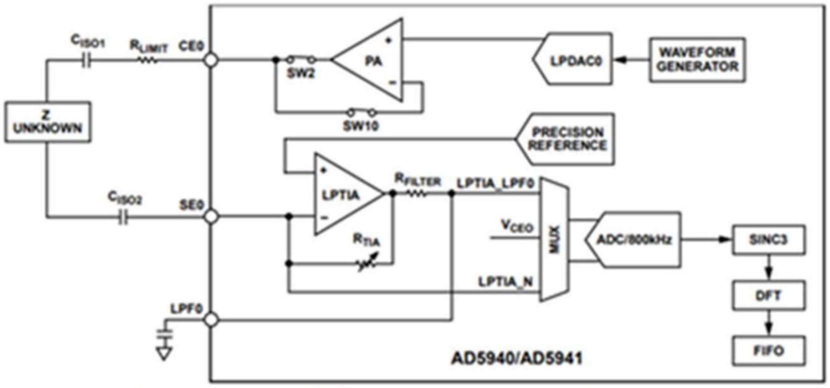


Figure 54. Low Frequency, 2-Wire, Bioimpedance Loop (Maximum Bandwidth = 300 Hz)

In 4-wire impedance measurement example,
Connections are as shown in the picture below.
The first measurement is done across CE0 and AIN1.
Second measurement is done across AIN2 and AIN3.

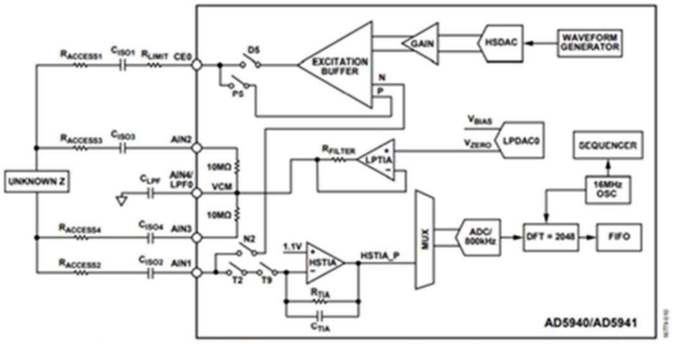


Figure 21. Low Power DACs Used in a 4-Wire Impedance Measurement (HSTIA_P = Positive Output of High Speed TIA)

Q20. Is it that for both 4-wire and 2-wire impedance measurements, we are connecting the sense pin of the sensor to AINx pins and following the two step measurement process as explained on page 12 of AN1563?

A. Yes. In both the techniques, Sense pin of Sensor is connected to AINx pins, and both follow 2-step measurement as shown in AN-1563 pg.12.

Q21. In AD5940, how to tune to ADC filter parameters to get correct measurement for all frequencies?

A. You may refer to

- **AppBATCheckFreq() function here:** ad5940-examples/BATImpedance.c at master · analogdevicesinc/ad5940-examples (github.com)
- **AppAMPRegModify() Pull requests · analogdevicesinc/ad5940-examples (github.com)**

Which tells how to update the AFE parameters after each measurement.

Q22. In AD5940, how to implement bidirectional ramp in Square Wave Voltammetry example?

A. You may refer to AD5940_SqrWaveVoltammetry_Bidirectional_ramp , Pull requests · analogdevicesinc/ad5940-examples (github.com)