

This thread has been locked.

If you have a related question, please click the "[Ask a related question](#)" button in the top right corner. The newly created question will be automatically linked to this question.

ADS1262: SPI transfer speed



Lucas Deruyter

Prodigy 10 points

Community Member

Part Number: [ADS1262](#)

I have two questions about the SPI communication of the ADS1262.

1) I'm not able to sample at 38ksps, the SPI communication is too slow for that.

I'm running at 8 MHz clock which is the maximum for the ADC.

Receiving the 6 data bytes takes around 65 μ s, because of that i can only sample at 15ksps max.

Any solutions for this?

2) Can I use a clock frequency higher than 8 MHz for the communication with other devices on the SPI bus?

[over 6 years ago](#)



Christopher Hall *over 6 years ago*

[TI_Mastermind](#) 44600 points

Hi Lucas,

Welcome to the TI E2E Forums!

Running the ADS1262 at a faster data rate indeed allows for less time to clock out the data. However, an 8 MHz clock should be fast enough to clock out 48 bits (~ 6 us) within the 38.4 kSP conversion period (~26 us). Do you have other digital delays that make the data read out take 65 us to complete?

Some recommendations I can offer:

- If you don't need the 38.4kSPS data rate, I would strongly recommend using lower data rates on the ADS1262. The noise level at the fastest data rate is significantly higher than all of other data rates so your effective resolution will not be all that great at 38.4kSPS.
- To reduce the time required to read data you could consider disabling the STATUS and CRC bytes so that you only need to read 32 or 40 bytes of data. Due to the increased noise at 38.4kSPS, chances are that the last data byte will be mostly noise, so you could also consider ignoring the last data byte and only reading 24-bits of data.
- Do make use of the /DRDY pin to interrupt your microcontroller. If your microcontroller is busy with other tasks or is polling /DRDY than it may not respond quickly to retrieve the data in time. Using the /DRDY interrupt will help you service the data read operation faster.
- Some microcontrollers will allow you to increase the SPI frame size. Using a larger frame may reduce the dead time between SPI byte transmissions and allow you to read the data a bit faster.
- As much as possible, try to optimize your code so that the processor is not performing unnecessary operations during the read data operation, or make use of DMA to directly read the SPI data into memory without disturbing the processor.