Application Note 502
Interfacing SPI Real Time Clocks With A Microcontroller

www.dalsemi.com

Dallas Semiconductor offers a variety of serial Real Time Clocks (RTCs). A number of these use the Motorola SPI interface to communicate with a microprocessor.

This application note is intended to help customers understand the basics of the SPI interface.

OVERVIEW

Figure 1 shows a typical single byte read and single byte write for the DS1306. Other SPI parts may have a different command byte structure. However, the timing sequence remains the same. Note that data are clocked into the RTC on the rising edge of SCLK. The RTC outputs data on the falling edge of SCLK.

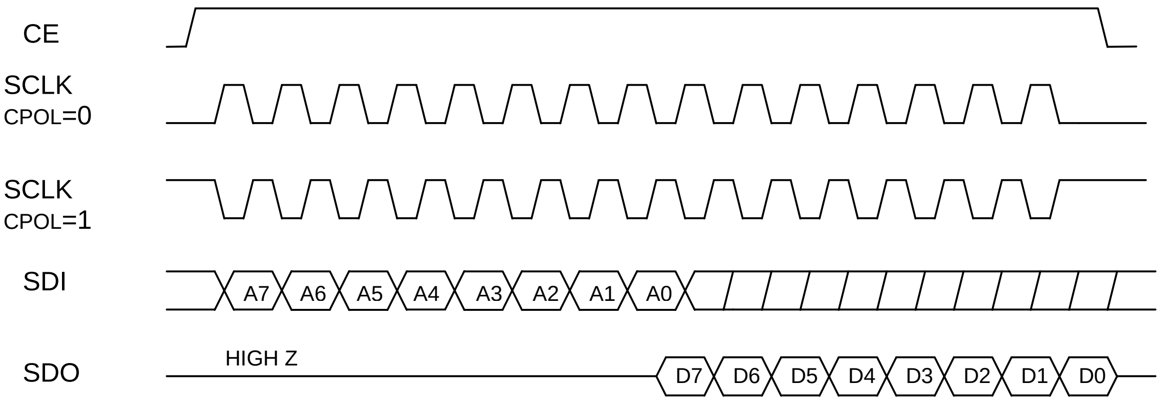
Note that during a read, the clock goes from reading the command byte to outputting data on one SCLK cycle. The processor must release the data line after the rising clock edge (after Tcdh). The RTC takes control of the data line on the falling edge of the clock.

NOTE:

SCLK must be in the proper idle state before the slave is enabled.

DATA TRANSFER SUMMARY Figure 1

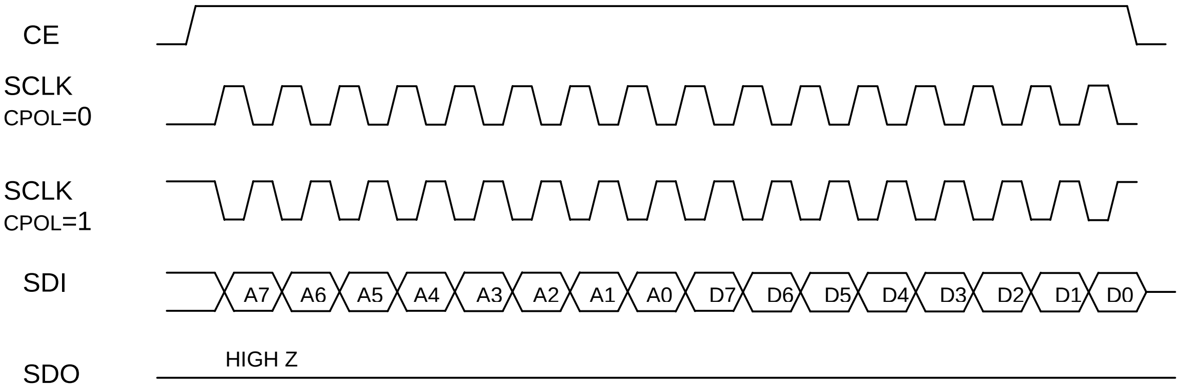
Single Byte Read



NOTE:

CPOL is the “Clock Polarity” bit that is set in the control register of the microcontroller. The CPHA bit polarity in the microcontroller must be set to a “1” (if applicable).

Single Byte Write



NOTE:

In burst mode, CE is kept high and additional SCLK cycles are sent until the end of the burst.

At the beginning of a data transfer, SCLK should be low. If SCLK is high when CE is brought active, the RTC may interpret that as a low to high transition of SCLK.

An example of SPI code, written for the DS1306, is available at:
ftp://ftp.dalsemi.com/pub/timekeeping/SPI_1306C.txt

SUMMARY

Following the information above will help insure that SPI communication routines will function correctly.