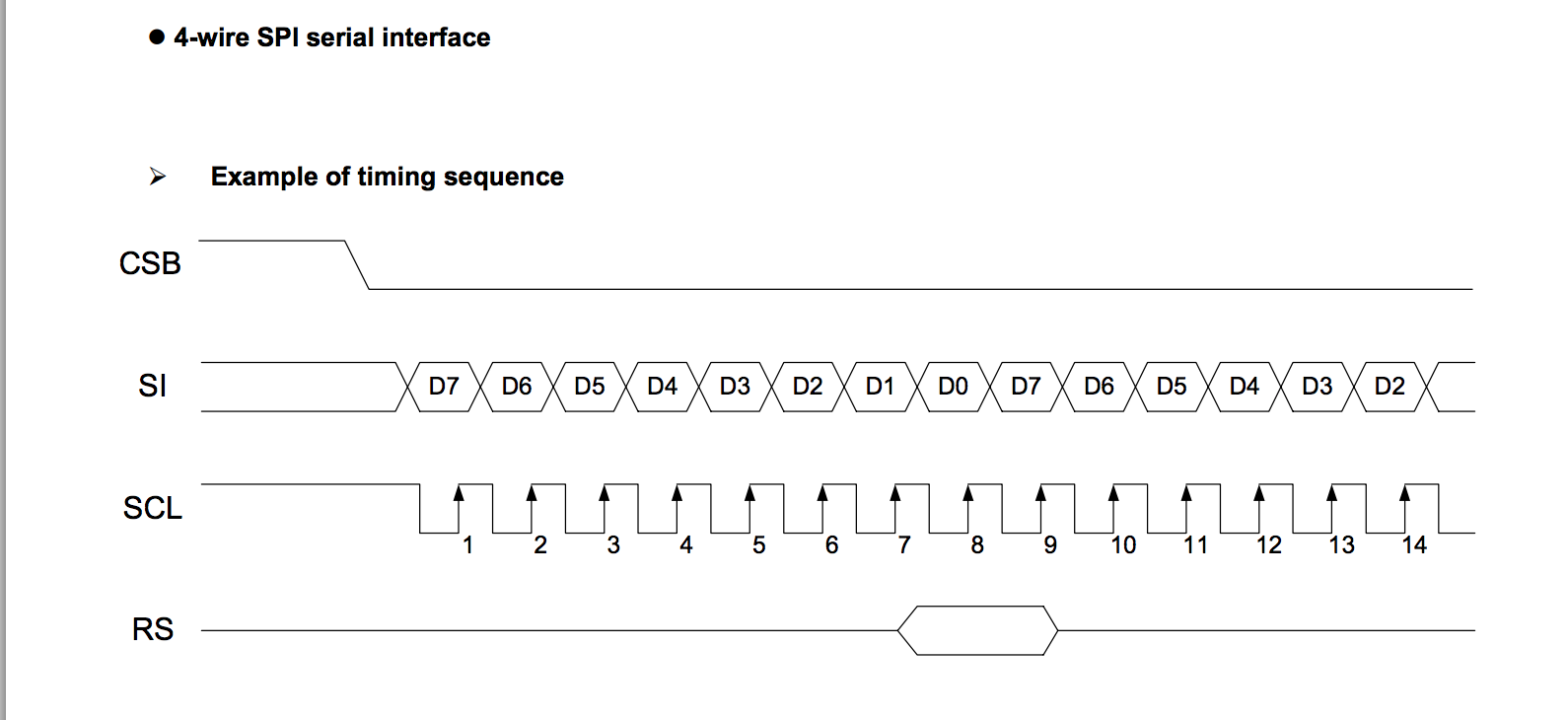
SPI Interface LCD driver and ADC driver

The software of this application relies on Serial Peripheral Interface (SPI) to transmit data. There are two approaches to implement the SPI in this system. The LCD driver uses the built-in SPI hardware from the microcontroller to send data to the LCD. The other part, the ADC driver, utilizes the software SPI to receive data from the ADC. Both approaches have important influences on the system.

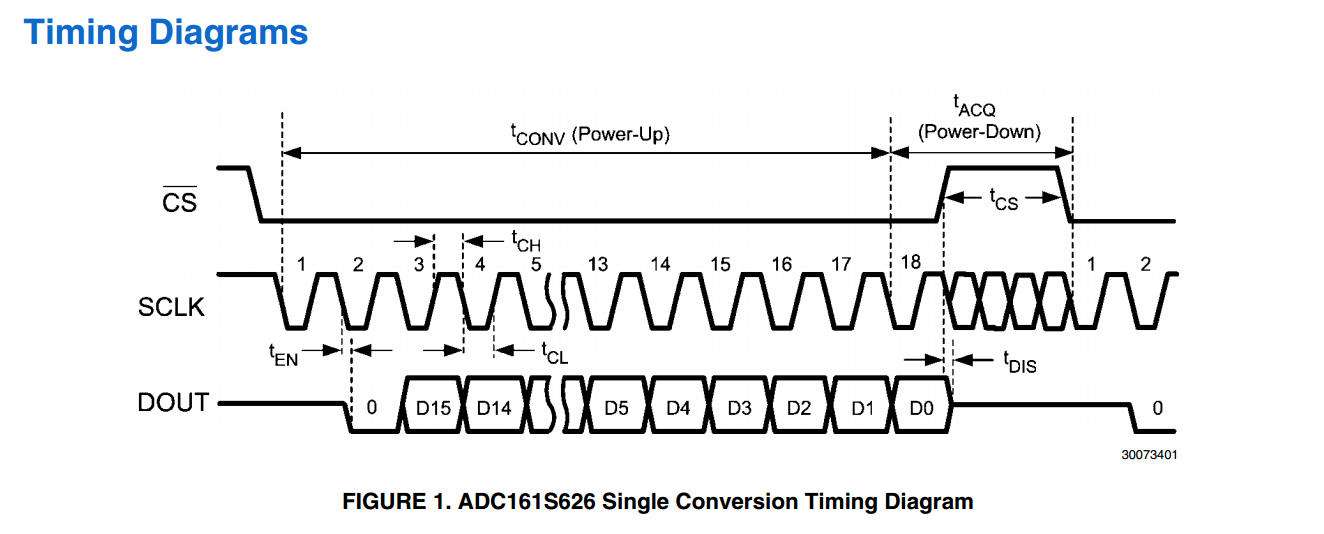
SPI hardware (LCD driver)



Diagram#1 Reference: ST7036 Dot Matrix LCD Controller/Driver Data Sheet.

The diagram above shows the timing of the SPI data transmission. The commands or data to be sent to the LCD have the size of one byte and transmit time of 8 clocks. Therefore, the SPI hardware can be used to communicate with the LCD efficiently.

To configure the SPI hardware: the clock polarity is 1 since the clock is high when CSB is idle (SPOL = 1). The clock phase is also 1 because the data is sample at the trailing edge of the clock (SPOH = 1). The minimum clock period is 100ns from the datasheet, which indicates that the 16MHz clock of the microcontroller needed to be divided by a factor of 2 to provide the fastest system clock to the LCD controller. See the file LCD\_driver.c for the full configuration of SPI hardware.



Diagram#2 Reference: ADC161626 16-bit Analog-to-Digital Converter datasheet

SPI software:

The software SPI approach is the so-called “bit-banging”, which uses the microcontroller to generate one clock with the proper period to transmit one bit of data. In the case of ADC161S626, the slave ADC needs to transmit 16 bits data to the microcontroller in 18 clock cycles, which violates the SPI hardware protocol that only generates 8 system clocks to receive one byte of data. Therefore, the “bit-banging” approach is used to transmit each bit of the data individually. The system will start with 2 dummy clocks to initiate the data transmission, then start receiving data at the third clock. See Flowchart 7 for a visual implementation and ADC161S626.c for the detailed implementation.