

LCD SPI DMA transaction time diagram (if FreeRtos used)

A drawing operation can consist of multiple DMA transactions (if the DMA transfer counter is only 16 bits size).

If LCD_DMA_TXWAIT == 0: at the beginning of the last DMA transaction, it return from the drawing function.

- What to watch out for?
When drawing a bitmap, do not change the contents of the bitmap until it is actually finished drawing.
- How to use it?
After a longer drawing activity, do other activities (eg file management).

If LCD_DMA_TXWAIT == 1: it only returns from the drawing function at the end of the transaction.

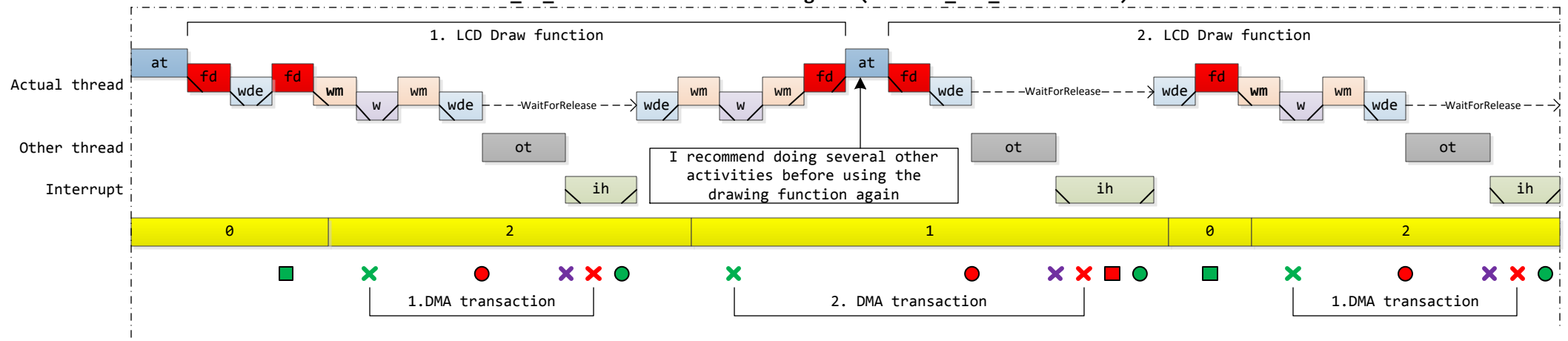
- When do we use it?
If you use another thread that can run while the DMA transaction is running.

- ✗ DMA start
- ✗ DMA stop
- ✗ DMA Interrupt
- Semaphore release
- Semaphore wait
- CS pin on (LOW)
- CS pin off (HIGH)

at: Actual thread
fd: LCD Draw function
wm: LCD_IO_WriteMultiData 8 or 16
w: LCD_IO_WriteMultiData
ih: DMAX_CHANNEL_IRQHANDLER (LCD_DMA_TX)
wde: WaitForDmaEnd
ot: Other thread (osSemaphoreWait)
LCD_IO_DmaTransferStatus

wde start of function
wde end of function

LCD_IO_WriteMultiData time diagram (#if LCD_DMA_TXWAIT == 0)



LCD_IO_WriteMultiData time diagram (#if LCD_DMA_TXWAIT == 1)

