LCD SPI DMA transaction time diagram (if FreeRtos used) DMA start at: Actual thread DMA stop fd: LCD Draw function A drawing operation can consist of multiple DMA transactions (if the DMA transfer counter is only 16 bits size). DMA Interrupt wm: LCD IO WriteMultiData 8 or 16 If LCD_DMA_TXWAIT == 0: at the beginning of the last DMA transaction, it return from the drawing function. Semaphore release w: LCD_IO_WriteMultiData - What to watch out for? When drawing a bitmap, do not change the contents of the bitmap until it is actually finished drawing. Semaphore wait ih: DMAX_CHANNEL_IRQHANDLER (LCD_DMA_TX) - How to use it? CS pin on (LOW) wde: WaitForDmaEnd After a longer drawing activity, do other activities (eg file management). CS pin off (HIGH) ot: Other thread (osSemaphoreWait) If LCD DMA TXWAIT == 1: it only returns from the drawing function at the end of the transaction. - When do we use it? LCD IO DmaTransferStatus If you use another thread that can run while the DMA transaction is running. start of end of wde function function LCD IO WriteMultiData time diagram (#if LCD DMA TXWAIT == 0) 1. LCD Draw function 2. LCD Draw function at Actual thread Other thread ot ot I recommend doing several other activities before using the Interrupt ih ih ih drawing function again 2 1 1.DMA transaction 2. DMA transaction 1.DMA transaction LCD IO WriteMultiData time diagram (#if LCD DMA TXWAIT == 1) 1. LCD IO WriteMultiData 8 or 16 2. LCD Draw function Actual thread WaitFor Release Other thread ot ot ot Interrupt ih ih XXO 1.DMA transaction 2. DMA transaction 1.DMA transaction