Control de gráficos sin controlador

Artículo cedido por Microchip

www.microchip.com

Autor: Steve Rogerson

Cómo configurar un controlador virtual para ahorrar costes y tiempo de CPU al representar gráficos

Una de las formas más habituales de crear aplicaciones con gráficos embebidos consiste en añadir un controlador gráfico interno o externo. El problema es que aumenta el coste, puede hacer que los diseños sean innecesariamente complejos y, en la mayoría de casos, para un interface gráfico de usuario sencillo estos controladores no se necesitan.

Una alternativa consiste en utilizar los periféricos del microcontrolador para obtener un controlador de gráficos virtual que represente gráficos sin necesidad de una gran cantidad de tiempo de CPU, que de hecho puede ser inferior al 5%.

En general, un sistema gráfico sin controlador necesita enviar una trama de información de píxel a un cristal de visualización a una determinada velocidad. Esta velocidad de refresco suele ser de 60 Hz. Para llevarlo a cabo, el sistema debe enviar constantemente los datos de trama al visualizador LCD. De entrada parece que esta tarea vaya a necesitar la mayoría del tiempo de CPU en un microcontrolador. Sin embargo, no ocurre así para los microcontroladores como los de la familia PIC32 de Microchip, que integran un periférico de acceso directo a memoria (direct memory access, DMA) para la transferencia de datos. Con un DMA que transfiere los datos del píxel se utiliza menos del 5% del tiempo de la CPU para obtener un controlador gráfico virtual.

El periférico DMA puede transferir datos de un punto a otro sin que intervenga la CPU. En un método gráfico sin controlador, el DMA se puede ajustar para transferir una línea de datos de trama a través del puerto maestro paralelo (PMP). Cada línea está formada por muchos puntos. El DMA enviaría una parte del buffer de trama durante una transferencia. Un PMP o solicitud de interrupción de temporización activa a continuación la siguiente transferencia DMA hasta que se ha transferido una línea. En dispositivos con interrupciones no persistentes se puede utilizar un temporizador como fuente de disparo de DMA.

Para dispositivos con un interface de bus externo (external bus interface, EBI), este módulo se puede utilizar como fuente de reloj de píxel. Una fuente de reloj de este tipo puede alcanzar unas velocidades de reloj de píxel más altas que el periférico PMP, aunque el EBI comparta las mismas patillas que el PMP.

Durante las transferencias de datos, el PMP o EBI emite una señal de lectura o escritura tras la transferencia de cada píxel. Las señales de lectura y escritura actúan como reloj de píxel para el cristal del visualizador. Una vez transferida cada línea de datos de píxel, el DMA interrumpe la CPU y ciertas señales de temporización, como HSYNC, VSYNC y DEN (data enable line), que son necesarias para actualizar que los visualizadores LCD. Esto se repite constantemente hasta que se ha dibujado una trama completa. La trama se almacena en una memoria volátil, por lo que la imagen puede ser dinámica.

En esta disposición se utiliza memoria SRAM y la configuración es la base para un sistema gráfico sin controlador. El sistema puede configurarse para que utilice memoria SRAM interna o externa como muestran las Figuras 1 y 2.

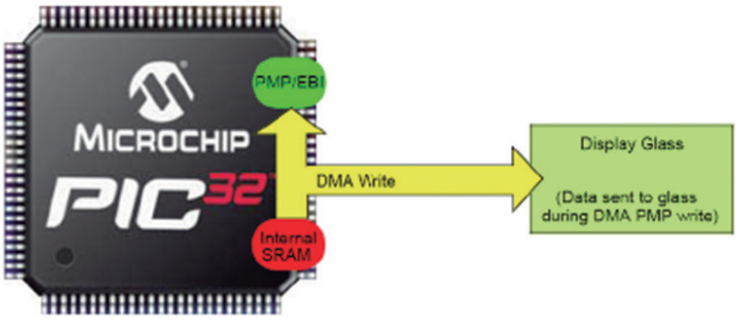

Figura 2. Método con memoria interna.

Visualizadores LCD TFT

Aunque el método gráfico sin controlador se diseñó para trabajar con visualizadores LCD TFT, también puede trabajar con cristales CSTN o MSTN introduciendo pequeñas modificaciones.

Las líneas de datos están formadas por información sobre el color del píxel. Los visualizadores LCD suelen tener de 8 a 24 líneas de datos de color, dependiendo de la profundidad de color del LCD. Estas líneas de datos alimentan el LCD con los datos de color en bruto de cada píxel.

Las señales de reloj HSYNC, VSYNC, DEN y PCLK sincronizan los datos del píxel y la trama de gráficos y el LCD. Las líneas de sincronización indican al LCD si los datos están al principio o al final de una línea (HSYNC) o trama (VSYNC). El DEN permite que el LCD conozca cuándo se está enviando un dato de píxel válido al LCD y algunos visualizadores LCD TFT debido al tiempo necesario para configurar el LCD para las siguientes ubicaciones del píxel.

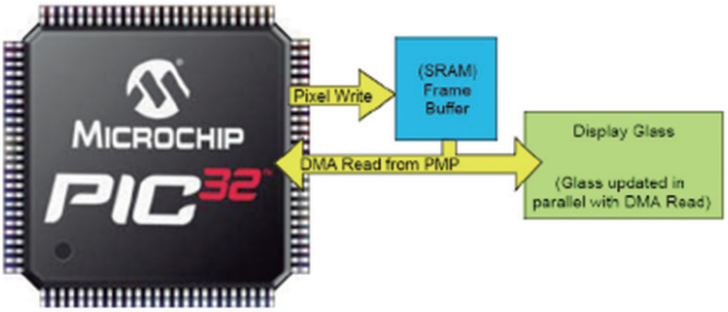

Figura 1. Método con memoria externa.

Los datos se envían línea a línea hasta dibujar la trama por completo. La señal PCLK es la fuente de reloj para todo el sistema. Un pulso de reloj de PCLK actualiza el LCD.

Las restantes líneas de reloj deben sincronizarse al reloj de píxel para obtener una salida de imagen adecuada. Los LCD que no contienen señales HSYNC y VSYNC también se pueden utilizar con la configuración gráfica sin controlador.

La tarjeta hija PICtail Plus para gráficos sin controlador de bajo coste de Microchip (tarjeta gráfica LCC) se diseñó para demostrar esta técnica y trabaja con muchos de los kits de iniciación PIC32 existentes. El driver de software de LCC puede ayudar con la sincronización que necesitan ciertos parámetros temporales, como la anchura de pulso, el tramo anterior y posterior del pulso en pulsos horizontales y verticales. Una vez compilados estos valores en el controlador gráfico LCC, el LCD visualiza la trama.

La Fig. 3 muestra lo que sucede dentro del microcontrolador PIC32 cuando se envía una trama gráfica al visualizador. El bloque DMA y PMP indica que los periféricos DMA y PMP que comparten el bus de datos con la CPU están en funcionamiento.

El bloque CPU indica las tareas necesarias para la representación de gráficos. La rutina de servicio de interrupción (interrupt service routine, ISR) de DMA es el único código que debe escribirse además de la configuración de los periféricos de DMA y PMP para enviar una trama de gráficos al visualizador.

Representación de nuevos píxels

La representación de nuevos píxels en el buffer de trama es tan importante como la actualización de la pantalla. Esto se lleva a cabo mediante la CPU, que escribe en el buffer del visualizador. Si la trama se almacena externamente, la transferencia DMA se suspende suspendido se esté actualizando la trama. Esto es necesario porque solo hay un periférico PMP y éste está siendo compartido por el controlador gráfico virtual o la transferencia DMA. Este método afecta a la velocidad de actualización de la pantalla. Es preciso supervisar la cantidad de actualizaciones de píxel para evitar un cambio de velocidad de actualización demasiado largo; de lo contrario el cambio será perceptible por el ojo humano. Esto se efectúa utilizando un contador de píxels variable incluido en el controlador gráfico virtual que se actualiza con cada escritura de píxel y se borra durante cada interrupción de DMA.

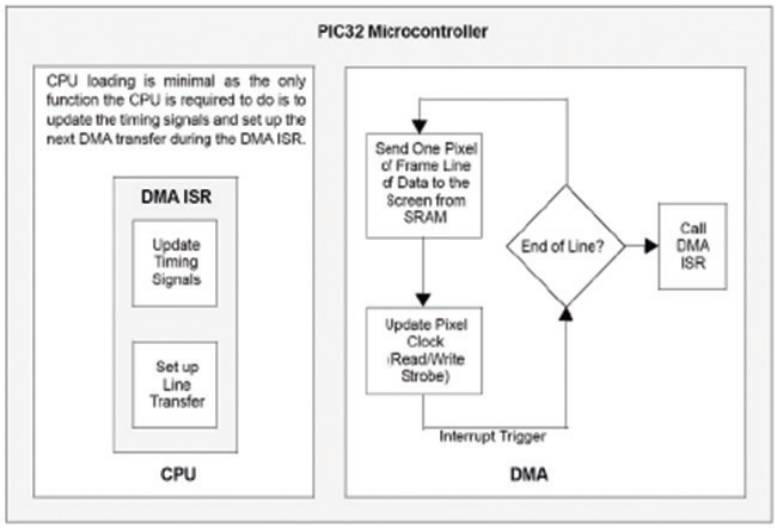

Figura 3. Organigrama de gráficos LCC en un microcontrolador PIC32.

El periférico EBI permite reducir enormemente el tiempo de suspensión necesario por actualización de trama. Cuando el EBI se utiliza para escribir se necesitan menos datos para almacenar y restaurar ya que los registros del PMP no se comparten entre lectura y escritura. Además, el módulo EBI es un periférico más eficiente cuando se comunica con una memoria SRAM externa.

Software

El método de memoria SRAM interna utiliza la señal de escritura del PMP para el reloj de píxel. Para ello se han de configurar las filas de conectores uno y dos de la tarjeta gráfica LCC. Con esta disposición, todos los colores son 8BPP y no se utiliza SRAM externa. La SRAM integrada en el microcontrolador está escribiendo continuamente sus valores de píxel en el PMP.

Para el color 8BPP se utiliza un formato de color 332 RGB, que tiene tres valores de color para el rojo, tres para el verde y dos para el azul. Se trata de un formato de color habitual ya que el rojo es un color más fácilmente detectable por el ojo humano que el azul.

El método de SRAM externa utiliza una señal de lectura del PMP para el reloj de píxel. Se han de configurar las filas dos y tres de la tarjeta gráfica LCC. En esta disposición todos los colores son 16BPP y la SRAM externa contiene la trama de gráficos que se está leyendo continuamente. Para un color 16BPP se utiliza un formato 565 RGB con cinco valores de color para el rojo, seis para el verde y cinco para el azul.

En ambos métodos, cuando se conecta a un LCD con más de 16 líneas de color, las líneas de color no utilizadas se corresponden con los bits más significativos del último bit de color utilizado. Esto asegura que se pueda obtener una escala completa de colores del blanco al negro.

Conclusión

Este artículo ha mostrado cómo implementar un sistema gráfico sin controlador de bajo coste con periféricos de microcontrolador para crear un controlador gráfico virtual utilizando solo una pequeña cantidad del tiempo de CPU.