

Practica Sistemas Empotrados, Dispositivos y Buses

Construcción de un juego retro

Los videojuegos de principios de los 90 estaban basados en hardware extremadamente simple, donde programando muchas veces en ensamblador se conseguía construir juegos sencillos pero extremadamente adictivos. Uno de los ejemplos más claros de este tipo de consolas es “Game Boy”, algunas de las características de este dispositivo era:

CPU: 8 bit a 4.19 Mhz basado en un microcontrolador [Intel 8080](#)

Memoria Ram: 8KB SRAM Total

Ram de video: 8KB

Pantalla LCD Monocromo.

Software: Cartuchos con bus de conexión rápido a memoria. Se trata de una memoria ROM ejecutable directamente desde el procesador.

Precio de venta 90 \$, precio venta equivalente actual 180 \$.

Un microcontrolador actual tiene un hardware de características parecidas e incluso mejores que el hardware antiguo a un precio muy bajo, con lo que en esta práctica se plantea la construcción de un pequeño juego inspirado en juegos de plataformas antiguas.



Material disponible:

Pantalla TFT

Una pantalla TFT a diferencia de una LCD es una pantalla gráfica que permite escribir píxeles de colores. La generación de texto se realiza a través de la unión de píxeles, siendo responsable el código del programa de realizar esta labor. La labor de colocar los píxeles en la pantalla es del controlador de pantalla, en este caso el circuito ST7735, que recibe por una conexión serial spi de 2 hilos la información a pintar en pantalla, y se encarga de manejar en este caso las 160 columnas y 128 filas. Es decir el controlador de pantalla tiene una conexión a cada fila y columna de forma matricial, y de esta forma seleccionar cada píxel de forma específica.

Un LCD es un panel de cristal que se vuelve opaco y su opacidad se controla eléctricamente. Los paneles LCD no generan su propia luz sino que necesitan una luz trasera (backlight) proporcionada por alguna fuente de luz (por ejemplo LED). El panel LCD tapa la luz emitida en ciertos puntos, lo que permite generar una imagen.

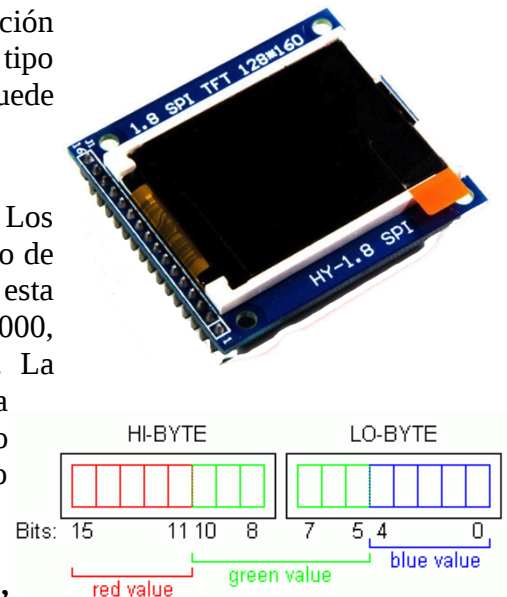
La luz emitida se polariza a través de un filtro para que las ondas de luz tengan una única dirección, esta luz se pasa a través de otros filtros polarizados que eléctricamente pueden controlar la cantidad de luz que pasa a través de ellos. De esta forma, se puede controlar el grado de opacidad del panel LCD. En una pantalla TFT RGB cada píxel tiene tres pequeñas celdas LCD con filtros de color que tapan uno de los componentes de la luz trasera, lo que genera la imagen formada por sub-píxeles de color Rojo, Verde y Azul. La combinación de estos colores primarios pueden generar cualquier otro.

La tecnología TFT (Thin Transistor Layer) hace referencia a la electrónica necesaria para conmutar el LCD. Los sub-píxeles se distribuyen en matriz, de forma que cada píxel se controla a través de un

transistor que a su vez se controla encendiendo la fila y columna correspondiente a ese píxel.

En nuestro caso utilizamos una pantalla TFT de 1.8 pulgadas de 128x160 píxeles, a color. La pantalla utiliza un protocolo SPI para la comunicación y tiene su propia memoria intermedia que se puede utilizar con todo tipo de micro controladores. El controlador de esta TFT (ST7735R) puede mostrar el color a 16 bits. También soporta tarjeta microSD.

Se utiliza como si fuera una memoria indicando dirección y valor. Los píxel tienen direcciones consecutivas y el valor se trata de un elemento de 16 bits en formato RGB 565 (Red 5 bits, Green 6 bits, Blue 5 bits). De esta manera si escribimos en la dirección 0 la palabra 1111100000000000, estaremos pintando el primer píxel de rojo, y así sucesivamente. La pantalla también admite comandos directos al controlador, de esta forma si queremos cubrir una zona de un color determinado, no tenemos que escribir todos los píxeles, sino darle un comando específico a la pantalla, lo cual es mucho más rápido.

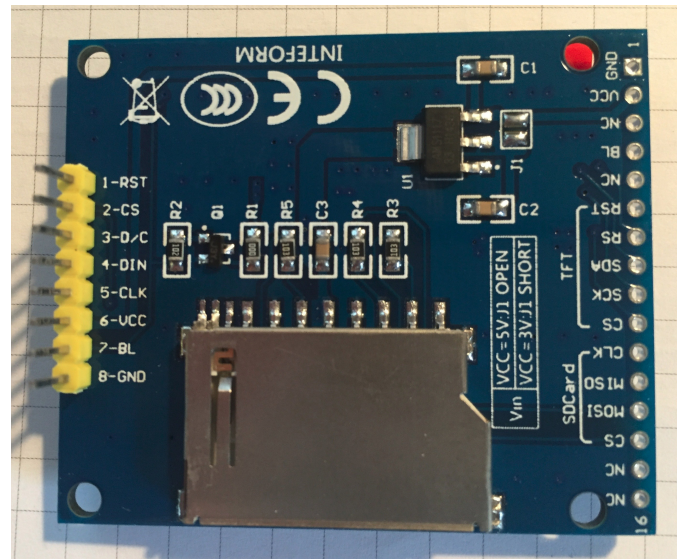


En nuestro caso la utilizaremos a través de la librería de **Arduino**, **Adafruit_GFX.h** y **Adafruit_ST7735.h**. Estas librerías incluyen una serie de subrutinas que permiten realizar dibujos sobre la pantalla de forma más sencilla y rápida. Consultar la documentación de la librería para consultar el listado de funciones disponibles.

La conexión se realizará a través del controlador de bus SPI de Arduino, conectado físicamente a los pines 11,12,13 (11 Salida de datos SPI, 12 entrada de datos SPI, 13 Reloj SPI). Para enviar se utiliza un buffer interno que vamos rellenando con los bytes a enviar para ir cubriendo los píxeles de la pantalla. Si no se envía nada, la pantalla mantiene su configuración.

La conexión se realiza utilizando estos pines, y otros que podemos elegir a voluntad para las señales,

- RST Reset
- BL led de Back Light
- CS Chip Select por si hay varias pantallas
- D/C Display Data Command selection
- DIN Data in de SPI
- CLK Reloj SPI



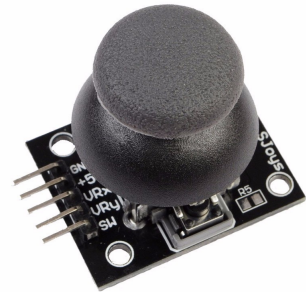
Para que la conexión funcione correctamente tenemos que colocar **en serie con cada pin, menos alimentación y tierra una resistencia de 1K**.

Imágenes cargadas en el código: Ya que el microcontrolador tiene poca memoria RAM y mayor cantidad de memoria flash de programa, las imágenes que ocupan más espacio normalmente se guardan como parte del código, para ello típicamente se incluyen en el código como: "const unsigned char", con

lo que se almacenarán como parte del programa y no gastarán memoria RAM. Existen programas que permiten a partir de una imagen convencional, generar la variable `const unsigned` para mostrar en el código, un ejemplo es LCD Image Converter <https://github.com/riuson/lcd-image-converter> que permite realizar esto de forma automática. Pondremos el código como variable “`const unsigned int logo [] PROGMEM = {`” para imágenes en color y “`const unsigned char logo [] PROGMEM = {`” para imágenes monocromo. De esta forma el compilador Arduino las incluirá en la compilación del código.

Para mostrar una imagen monocromo podemos utilizar el comando `tft.drawBitmap`, en niveles de gris, `drawGrayscaleBitmap`, o en color `drawRGBBitmap`. Dentro de la librería ST7735.h

Mando potenciómetro, se trata de un control mando X-Y, compuesto por dos resistencias variables que proporcionarán un voltaje asociado a la posición del mando. Entradas 5V y Gnd, salidas Vrx, Vry que es el voltaje entre 0-5 voltios que marca la posición. Algunos modelos incluyen un botón sw para seleccionar. Necesitamos usar el conversor analógico digital para utilizar este dispositivo.



Realización práctica:

Programar un juego utilizando pantalla y mando, le juego debe incluir

1. Uso de [sprites](#) y generación gráfica dinámica, para acelerar el entorno gráfico. La velocidad de la pantalla es demasiado lenta para en cada cambio borrar completamente y generar desde cero la pantalla, con lo que tendremos que llevar un registro de los cambios a realizar para evolucionar el juego.
2. Función de calculo de números aleatorios y escenarios con comportamiento aleatorio. Podemos utilizar un Conversor Analógico digital sin conectar a nada y tomar el último bit como aleatorio.
3. Gestión del mando como interfaz de usuario, incluyendo ejeX ejeY y botón central.
4. Detector de colisiones (dependiente del juego), Detector de final de juego Game Over y puntuación.
5. Presentación de gráficos tipo logo inicial, o dinámicos basados en BMP almacenado en memoria.

Partes opcionales:

1. Posibilidad de guardar de manera indefinida las mejores puntuaciones a través de la EEPROM no volátil del microcontrolador.
2. Niveles de dificultad y menú de gestión para seleccionar opciones del juego.
3. Calidad en la programación que puede incluir
 - Nivel de optimización en las funciones
 - Calidad del juego
 - Complejidad del juego
 - Aprovechamiento del Hardware del microcontrolador.