
Manual Pantallas TFT Touch

DST-80048070-10

AG Electrónica S.A de C.V



Descripción del producto

Hoy en día, las pantallas de color definido se utilizan ampliamente en la industria de control. Estas pantallas con resolución 16 * 02, 128*64 y 240*128 no podían satisfacer la experiencia del usuario y las exigencias funcionales de más productos. Mientras tanto, la mayoría de los procesadores de control industrial en China están siendo construidos principalmente en plataformas como 51, AVR y ARM7 / Cortex-M3, etc. La mayoría de estos procesadores no podrían apoyar dispositivos de control TFT directamente mientras que pocos ARM7 y ARM9 están equipados con interfaces de control LCD; por lo tanto, ciertas dificultades en el desarrollo y un ciclo de investigación a mayor plazo permanecen si se supone que asuntos tales como pantalla de alta definición, descarga de imagen y sonido, GUI y materiales de diseño de arte estén aún por resolver. La creación de un puerto serie HMI separa por completo el control de usuario de la pantalla. Los usuarios no tendrán que modificar sus códigos de control básicos además de añadir funciones para la transmisión y recepción de puerto serie sobre la base de los códigos de origen, es decir, que podrían actualizar sus productos rápidamente en pantallas a color definido. Sin embargo, este tipo de funciones para HMI podrían ser generados automáticamente por medio de la utilización del software Visual TFT instalado en los equipos de control, por lo tanto realiza la reducción de las dificultades en el desarrollo y la mejora de la eficiencia de trabajo y un rendimiento de productos. DST80048070- es una pantalla con puerto serie de 7 pulgadas táctil HMI con una resolución de 800*480, está integrado con una memoria flash no volátil de 1GB, lo que podría permitir a los usuarios almacenar texto, imágenes fijas, imágenes GIF y archivos de sonido, etc. Para ello, los usuarios deben utilizar el software de PC para descargar imágenes requeridas en la memoria flash del panel de operador por medio de USB, y luego controlar funciones como la visualización de imágenes mediante el envío de instrucciones en serie en el puerto correspondiente a través de MCU. Todo el proceso es fácil de entender y operar.

Características del producto

- **Unidad de Procesamiento Central**

Adopta un procesador de 32 bits ARM + marco de control de doble núcleo FPGA de alta velocidad, que refuerzan la capacidad de procesamiento gráfico.

- **Configuración de almacenamiento**

Disfruta de una capacidad de almacenamiento de 1GB, que es capaz de almacenar alrededor de 140 800 * 480 imágenes en color definido.

- **Características de la interfaz**

-Equipado Con una interfaz de imagen y descarga de máxima velocidad USB a 600 KB/S.
-Una interfaz de comunicación: RS232 o nivel / TTL 3.3V CMOS.

- **Características del software**

-Es entorno Potente llamado TFT Visual y con descarga de IDE de compilación; la interface visual es buena y eficaz.

-Este IDE ha integrado una gran cantidad de iconos, botones, vistas en 3D y otros diagramas de flechas en los sectores de control industriales, lo que podría ayudar a crear vistas de diseño-arte de profesionales.

-Soporte en la creación de páginas. Después de que se han creado, el software generará automáticamente la función de guiar para cada página; por lo tanto, no es necesario comprobar sus instrucciones.

-Soporte para funciones tales como el dibujo de la imagen, la transparencia, la edición de texto, copiar un objeto y guardar la página como una imagen.

-Soporte para el software de PC y exhibición síncrona HMI, que gozan de funciones como la de un solo paso de depuración.

-Apoyo en quema de archivo binario; la producción en masa es aún más conveniente y segura. El IDE generará archivos de proyecto binarios de forma automática después de que se compilan los proyectos.

- **Características del Hardware**

-Pantalla RGB color real de 16-Bit (65536 colores).

-Soporte en resolución de 800* 480.

-Construido bajo estándares y existencias de palabras ASIC 8*12, 8*16, 12*24, 16*32, 12*12, 16*16, 24*24GBK palabras existentes y 32*32GB2312 palabras existentes.

-Soporte para la visualización del cursor.

- Apoyo en puntos de dibujo, círculos, línea recta, rectángulo, y datos lineales ajustados, así como otras funciones de GUI.

- Soporta la descarga de formatos de imagen: JPG, BMP, JPEG, WMF, PNG y GIF etc.

- Soporte para la visualización de las imágenes en cualquier zona, visualización de la imagen cortada, actualizar la región, visualización flash de imagen GIF.

- Relleno y Soporte DMA de un solo color, visualización de la imagen transparente.

- Soporte en funciones tales como el tacto.

- Control de luz de fondo, manejable hasta 255 niveles.

- El rango de velocidad de transmisión: 1200-15200bps.

- Soporte en línea para Actualizaciones de hardware.

- Personalización de acuerdo a las necesidades para satisfacer todas las demandas de los usuarios.

- **Características LCD**

- Monitor de tipo: TFT LCD.
- Brillo: 350 cd/m2.
- Tubo de luz de fondo: LED.
- Vida útil de luz de fondo (en promedio): 20.000 horas.
- Dimensiones: 190 x 110 mm

- Contraste: 500: 1
- Ángulo visual (L/R/T/B): 60/60/40/60.

- **Fuente de alimentación.**

- Tensión de entrada: 9~26V@2A. (Recomendada 12V)

- **Temperatura de funcionamiento**

- 20 °C ~ 70 °C con un sufijo W o 0°C~70°C con un sufijo C.

Campos de aplicación

- Menú electrónico
- Casa inteligente
- Equipo de diagnóstico para automóviles
- Panel de control
- Electrocardiógrafo portátil
- Maquina plana



Manual de aplicación del DST80048070-10

Versión	Descripción	Fecha	Revisión
V1.0	Primer Versión	2011-03-24	

Contenido

SECCIÓN 1 AVISO DE USO	7
1.1 Tratamiento de película protectora.....	7
1.2 Evitar la electricidad estática durante el estañado.....	7
1.3 Avisos para montaje de operaciones.....	8
1.4 Mantenimiento de productos y almacenamiento.....	8
1.5 Experiencia del producto.....	8
 SECCIÓN II INTRODUCCION DEL PRODUCTO.....	 9
2.1 Significados de Creación de productos DST80048070-.....	9
2.2 Introducción a DST80048070-.....	9
2.3 Introducción al software visual TFT.....	11
 SECCIÓN III ASPECTO DEL PRODUCTO Y DIMENSIÓN.....	 12
3.1 Dimensión del producto.....	12
 SECCIÓN IV DEFINICIÓN DE INTERFACES DE COMUNICACIÓN.....	 13
 SECCIÓN V REGLAS DE NOMENCLATURA Y MANUAL DE SELECCIÓN DE MODELO.....	 14
5.1 Normas de denominación de operador.....	14
5.2 Manual de selección de modelo.....	15
 SECCIÓN VI DESCRIPCIÓN DEL CONJUNTO DE INSTRUCCIONES HMI por DAS HMI.....	 16
6.1 Formato de instrucciones.....	16
6.2 Formato de color.....	16
6.3 Lista de instrucciones.....	17
6.4 Explicaciones detalladas para aplicación de instrucciones.....	22
6.4.1 Comando Hand-Shake.....	22
6.4.2 Asignar y obtener el color de fondo.....	22
6.4.3 Limpiar Pantalla.....	23
6.4.4 Ajuste de tono en filas y columnas para las palabras.....	23
6.4.5 Ajuste de color Filtrado.....	24
6.4.6 Visualización de texto.....	25
6.4.7 Visualización de cursor.....	26
6.4.8 Visualización de imagen.....	27
6.4.9 Cortar la imagen.....	28
6.4.10 Visualización de imágenes Flash.....	29
6.4.11 Dibuja un punto.....	30
6.4.12 Dibujar líneas.....	30
6.4.13 Dibuja un círculo hueco.....	31
6.4.14 Dibuja un círculo sólido.....	32
6.4.15 Dibuja un rectángulo hueco.....	32
6.4.16 Dibuje un rectángulo sólido.....	33
6.4.17 Dibuja una elipse hueca.....	33
6.4.18 Dibuja una elipse sólida.....	34

6.4.19 Ajuste el backlight.....	34
6.4.20 Control del buzzer.....	34
6.4.21 Carga valor táctil.....	34
6.4.22 Leer valor táctil.....	35
6.4.23 Calibración de la pantalla táctil.....	35
6.4.27 Ajuste de la velocidad de transmisión	3
SECCIÓN VII CIRCUITOS Y PROGRAMAS DE REFERENCIA.....	37
7.1Diagrama de referencia del MCU y conexión HMI.....	37
7.2Controles del MCU 8051 y HMI, Procedimientos del Display y programa de referencia....	37
Descargas.....	49

SECCIÓN 1 AVISO DE USO

Estamos dedicados a brindar todo el apoyo y post venta de servicios técnicos. Los productos LCD pertenecen al instrumento de precisión, por lo que son perecederos y del cuál podría ser rayado fácilmente.

Aunque se han hecho todas las precauciones necesarias antes de salir de fábrica, todavía recomendamos leer siguientes avisos cuidadosamente antes de utilizar los productos para que la pérdida innecesaria pueda evitarse.

1.1 Tratamiento de película protectora

Antes de salir de la fábrica, la pantalla de visualización o el panel táctil tiene su superficie cubierta con un protector de pantalla para evitar manchar la pantalla durante el transporte, el montaje y la depuración.

Despojar a tal película, podría generar electricidad estática y provocar una visualización de imagen anormal pero será considerado como un fenómeno normal. En poco tiempo, HMI se recuperará automáticamente.

Sin embargo, la película sobre la pantalla táctil necesita ser despojada durante su uso o se reducirá su sensibilidad.

1.2 Evitar la electricidad estática durante el estañado

HMI ofrece a los usuarios dos almohadillas de soldadura de cortos circuitos para poder escoger, se utilizan para recuperar los ajustes antes de salir de la fábrica y el interruptor de Nivel de Comunicaciones RS232 / TTL (COMS), respectivamente. Todos los productos HMI desarrollados por INDUSTRY DAS LIMITED adoptan un nivel RS232 con una velocidad de 9600 Kbps por defecto antes de salir de la fábrica. Si los usuarios necesitan modificar esta configuración, se debe prestar atención a los siguientes puntos antes de soldar el producto.

1. No toque las almohadillas de soldadura de cortocircuito deliberadamente con la mano para evitar el cortocircuito.
2. Al tocar los elementos, los usuarios deben mantener su cuerpo y módulos en el mismo potencial eléctrico o tierra de su cuerpo correctamente a fin de evitar daños de electricidad estática corporal a los productos.
3. Las conexiones eléctricas empleadas durante la soldadura deben conectarse a tierra correctamente. Una gran cantidad de enchufes de baja calidad sólo tienen un cable nulo y un cable de alta tensión, pero no tienen el cable de tierra. En ambiente seco, es bastante probable que la descarga de elementos de electricidad y circuitos perfore la cabeza de soldadura.
4. Los cautines y herramientas electrónicas de funcionamiento deben conectarse a tierra correctamente a fin de evitar fugas eléctricas.
5. La humedad en los talleres o lugares de trabajo será de alrededor de 60% RH.

6. Requisitos para la temperatura a la que se va a soldar: soldar sin plomo: por debajo de 320 °C; soldar con plomo: por debajo de 280 °C.

7. El tiempo para la soldadura tiene una duración de menos de 3-4 segundos, y no utilice polvo de escalamiento ácido.

8. Asegúrese de que los productos se manejan con cuidado en el momento de sacarlos o ponerlos de nuevo en paquete.

1.3 Avisos para montaje de operaciones

Todos los productos HMI están diseñados y ensamblados cuidadosamente. Por lo tanto, los usuarios no podrán fabricar o modificarlos por su cuenta. Además, los usuarios deberán prestar atención a los siguientes puntos al realizar operaciones de montaje.

1. La estructura metálica del marco no deberá ser torcida o desmontada deliberadamente.
2. Los cuatro tornillos se aprietan correctamente. Evite apretar los tornillos en exceso o soltarlos.
3. Las pantallas deberán colocarse correctamente. Evite cualquier inclinación por lo que no se verá afectado el alcance visual.
4. Todo el equipo deberá ser fijado a fin de reducir sacudidas.

1.4 Mantenimiento de productos y almacenamiento

Como las pantallas están expuestas durante mucho tiempo, van a ser fácilmente que se tenga suciedad y polvo, por lo que se requiere la limpieza no periódica. Se recomienda la utilización de alcohol isopropílico o alcohol acetaldehído. Por ciertas razones, si necesita ser almacenada durante un largo plazo, le aconsejamos los siguientes métodos:

5. Ponga el producto en una bolsa antiestática y séllelo.
6. Colóquelo en una zona oscura y evite el sol directo.
7. La temperatura ambiente durante el almacenamiento debe oscilar entre -10 °C a 35 °C.
8. No coloque ningún objeto sobre su superficie.

1.5. Experiencia del producto

Después de haber terminado de leer los avisos mencionados, te invitamos a observar las experiencias del software y hardware:

9. Sacar cables de alimentación y conexión HMI.
10. Prueba de encendido. Normalmente, después de que la alimentación está encendida, la pantalla mostrará una imagen fija antes de su salida de la fábrica; si no, por favor, vuelva a comprobar la conexión de hardware.
11. Instalar el entorno de compilación Visual TFT IDE desarrollado por INDUSTRY DAS LIMITED.
12. Lea y comience con el software Visual TFT, y lleve a cabo las pruebas de acuerdo con los procedimientos especificados en el mismo.
13. ¡Enhorabuena! Se le ha informado de procedimiento de desarrollo del producto.

14. Durante su uso, no dude en contactarnos si usted encuentra cualquier problema.
15. Contacto: + 8621-51082662 Correo electrónico: sales@das-tek.com.

SECCIÓN II INTRODUCCION DEL PRODUCTO

2.1 Significados de Creación de productos DST80048070-10

En los últimos años, la tecnología LCD se ha desarrollado rápidamente. Las pantallas de color definido TFT aparecen por todas partes en la vida de las personas, por ejemplo, el teléfono celular, la entrada de la puerta del metro o un vehículo de navegación / AV y ATM etc. Con la creciente competencia de productos entre empresas de la misma industria, las pantallas de color simple tradicionales no podían satisfacer la experiencia del usuario y las exigencias funcionales de productos, especialmente la visualización de imágenes de alta definición y la imagen Flash. Además, Las pantallas TFT, favorecen en la elección de pantallas en color definido, por lo tanto, en algunas industrias, las pantallas de color definido tienen una tendencia a ser sustituidas. Las pantallas de color simple en el futuro. La mayoría de los procesadores de control industrial en China todavía se construyen principalmente en plataformas como 51, AVR y ARM7 / Cortex-M3, etc. La mayoría de estos procesadores no podrían soportar los periféricos de control TFT directamente, mientras que pocos ARM7 y ARM9 están equipados con interfaces de control LCD; por lo tanto, ciertas dificultades en el desarrollo y un ciclo de investigación y desarrollo a más largo plazo permanecen si suponemos cuestiones como la pantalla de alta definición, descarga de imagen y sonido GUI y materiales de diseño de arte están por resolver. Los ingenieros no necesitan pasar un buen rato construyendo la nueva plataforma ARM 9, el desarrollo de software para PC de supervisión o de la realización de trasplante de código, ya que esta práctica aumentará los riesgos de la prueba. La creación de DST80048070-10 separa por completo el control de usuario de la pantalla. Los usuarios no tendrán que modificar sus códigos de control básicos, además de añadir funciones para la transmisión y recepción de puerto serie sobre la base de los códigos originales, es decir, que podrían actualizar sus productos rápidamente en las pantallas a color definido.

2.2 Introducción a DST80048070-10

DST80048070-10 es una interfaz hombre-máquina integrada con interfaz gráfica de usuario, descarga de imágenes, almacenamiento y visualización. Los usuarios pueden controlar la visualización de las palabras, las imágenes y las fotografías con flash y otros en las pantallas a color definido utilizando el puerto serie de 8 bits MCU en la parte de puertos serie. El dibujo esquemático se muestra en la figura:

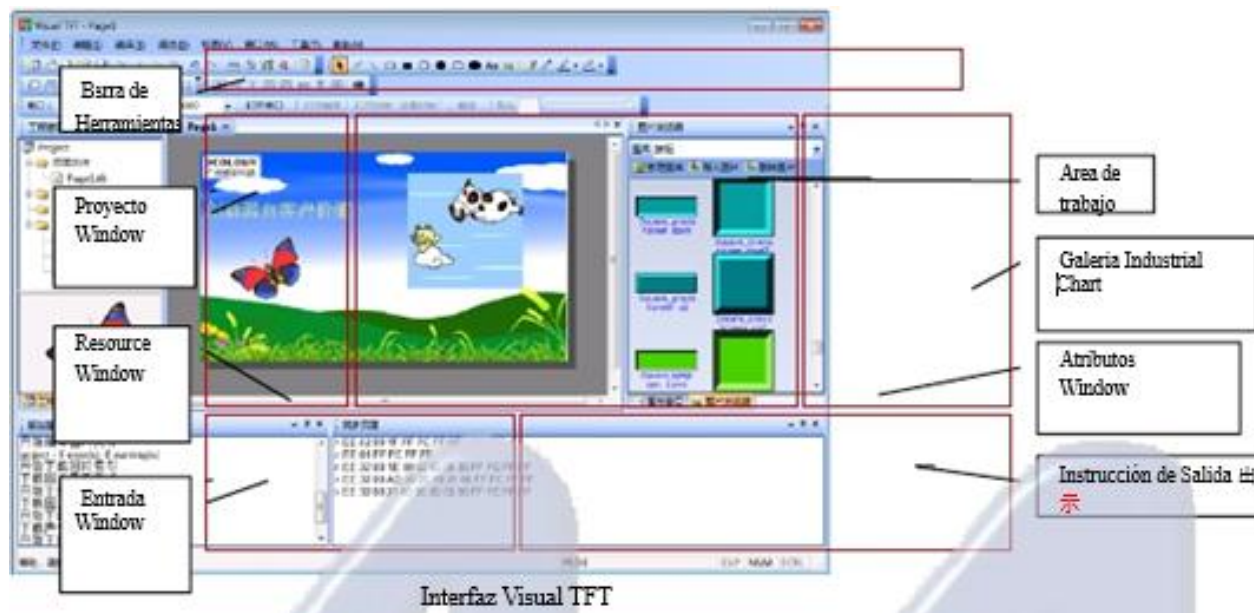


Las funciones principales del hardware se muestran a continuación:

- Pantalla RGB de color definido de 16 bits (65.536 colores).
- Soporte de resolución de 800 * 480.
- Construido bajo estándares de 8 * 12, 8 * 16, 12 * 24, 16 * 32 de palabras existentes ASIC, 12*12, 16*16, 24*24GBK palabras existentes y 32*32GB2312 de palabras existentes.
- Visualización del cursor de la ayuda.
- Soporte en puntos de dibujo, círculos, línea recta, rectángulo, y datos lineales ajustados, así como otras funciones de GUI.
- Descarga de formatos de imagen en: JPG, BMP, JPEG, WMF, PNG y GIF etc.
- Soporte en la visualización de las imágenes en cualquier zona, visualización de la imagen cortada, actualizar la región, visualización de la imagen de flash GIF.
- Soporte DMA llenado de un solo color, visualización de la imagen transparente.
- Soporte para funciones tales como el tacto, este modelo no soporta RTC.
- Control de luz de fondo manejable hasta 255 niveles.
- El rango de velocidad de transmisión: 1200- -15200bps.
- Soporte en línea para actualizaciones de hardware.
- Personalización de acuerdo a las necesidades para satisfacer todas las demandas de los usuarios.

2.3 Introducción al software visual TFT

Visual TFT, es una poderosa recopilación y descarga de software gráfico desarrollado por INDUSTRY DAS LIMITED, es compatible con todos sus productos HMI. Después de que los usuarios están conectados correctamente, se deberán realizar funciones tales como la exhibición síncrona en el PC y HMI, y de un solo paso de depuración. Este software cuenta con una gran cantidad de funciones de galerías gráfico vectorial IC, lo que podría ayudar a resolver los problemas relacionados con el diseño de arte para los ingenieros de desarrollo. La interfaz de software se muestra en la siguiente figura:

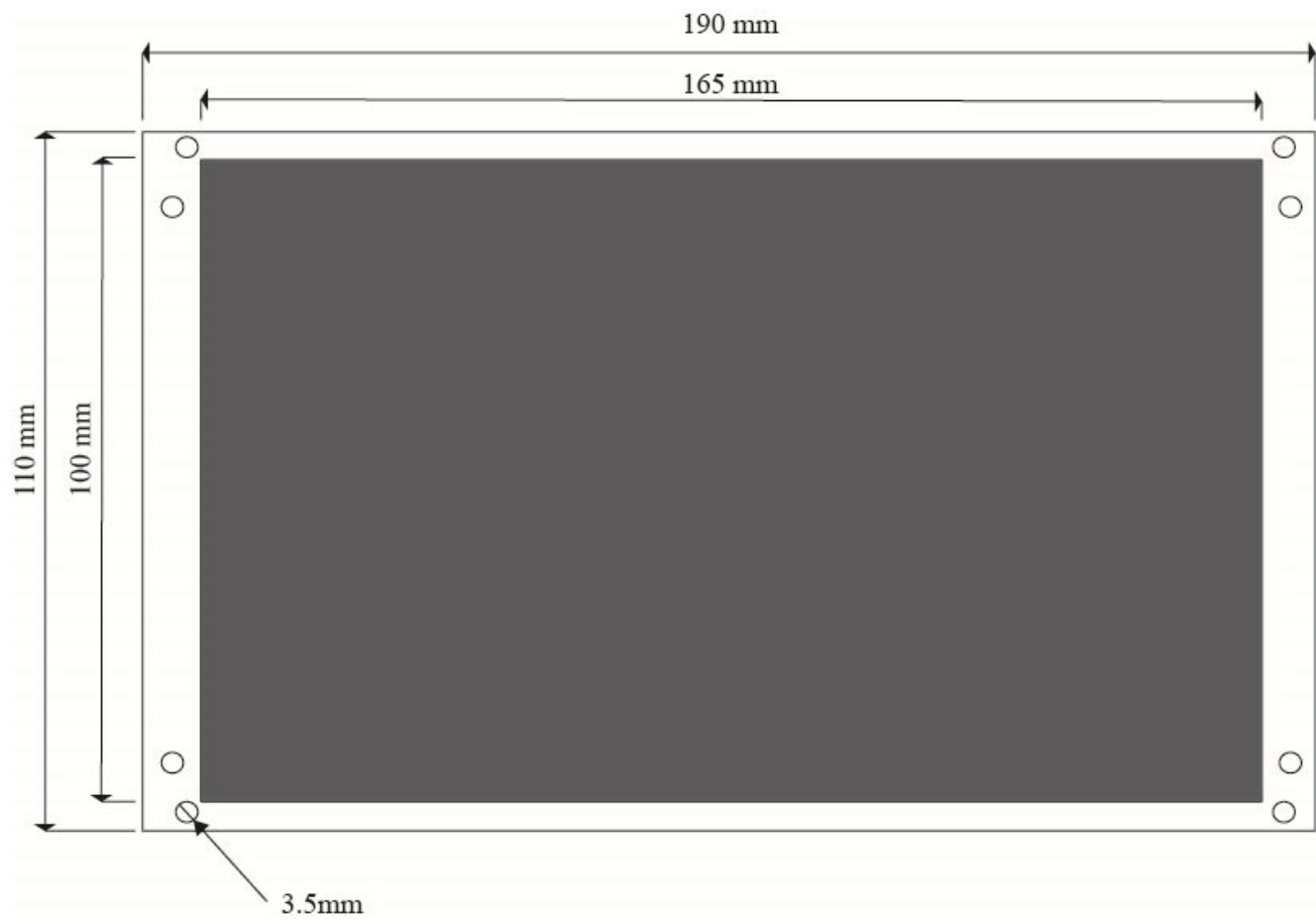


Las principales funciones del software se muestran a continuación:

- Entorno de descarga compilación IDE Potente Visual TFT, ventana visual, buena y eficaz interfaz.
- Este IDE ha integrado una gran cantidad de iconos, botones, vistas en 3D y otros diagramas de flecha en los sectores industriales de control, lo que podría ayudar a reducir las dificultades en el diseño de arte.
- Soporte en funciones tales como el dibujo de la imagen, la transparencia, la edición de texto y copia mutua de los elementos en las páginas.
- Soporte en la creación de varias páginas. Después de que se ha cumplido, el software generará automáticamente la función de manejo para cada página, por lo tanto, no es necesario comprobar el conjunto de instrucciones.
- Soporte de exhibición síncrona en el software de PC y HMI, que gozan de funciones como la de un solo paso de depuración.
- Soporte de quema archivo binario, la producción en masa es aún más conveniente y segura. El IDE generará archivos de proyecto automáticamente binarios después se cumplan los proyectos.

SECCIÓN III ASPECTO DEL PRODUCTO Y DIMENSIÓN

El diagrama de dimensiones de DST80048070-10 se muestra en la siguiente figura:



SECCIÓN IV DEFINICIÓN DE INTERFACES DE COMUNICACIÓN

Todos los productos HMI fabricados por INDUSTRY DAS LIMITED cumplen con las siguientes especificaciones de interfaz. Las definiciones de la instrucción puerto de comunicación e imagen del puerto de descarga se muestran en la Tabla 4.1 y 4.2, respectivamente. La imagen del producto se muestra en la siguiente figura:

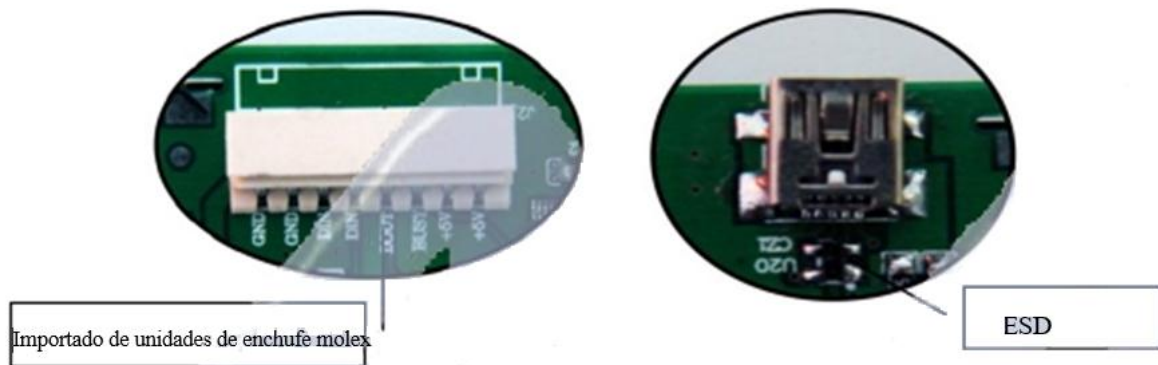


Tabla 4.1. Instrucciones del puerto de comunicación

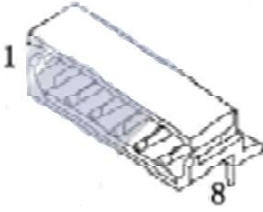

	No.	Nombre	Tipo	Descripción
	1.2	VCC	Entrada	Alimentación de entrada de 5V
	3	BUSY	Salida	Señal que ocupa de salida el puerto serie HMI
	4	DOUT	Salida	Puerto de salida HMI (TXD), utiliza el MCU Entrada (RXD)
	5.6	DIN	Entrada	Entrada de puerto serial HMI (RXD), se usa MCU de salida (TXD)
	7.8	GND	Entrada	Alimentacion a tierra

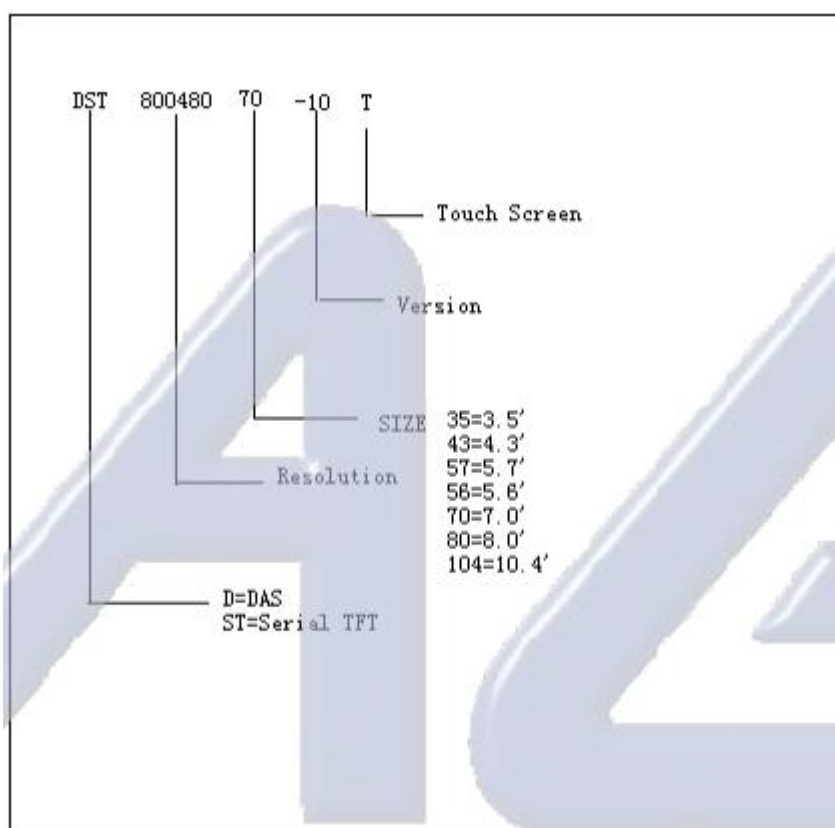
Tabla 4.2. Definición Descargar Puerto de imagen

	Nombre	Descripción
	Interface Mini USB	Imagen, interfaz de descarga de sonido, alta velocidad USB2.0 estandar

SECCIÓN V REGLAS DE NOMENCLATURA Y MANUAL DE SELECCIÓN DE MODELO

5.1 Normas de denominación de operador

Los detalles sobre nombrar DST80048070-10se muestran a continuación:



5.2 Manual de selección de modelo

Tabla 5.1: Tabla de selección de modelo para los productos HMI Desarrollado por INDUSTRY DAS LIMITED.

Tabla de Selección de Modelo para los productos HMI Desarrollado por INDUSTRY DAS LIMITED										
Dimensión	Modelo	Resolución	Ratio	Color	Almacenamiento (Bit)	Táctil	Audio	RTC	Temperatura de Operación	Voltaje de Operación
3.5	DST32024035-20	320*240	4:3	65K	1G	x	X	x	-20~+70°C/0~+70°C	5V
	DST32024035-20T	320*240	4:3	65K	1G	√	x	x	-20~+70°C/0~+70°C	5V
4.3	DST480272043-20	480*272	16:9	65K	1G	x	x	x	-20~+70°C/0~+70°C	5V
	DST480272043-20T	480*272	16:9	65K	1G	√	x	x	-20~+70°C/0~+70°C	5V
5.6	DST64048056-20	640*480	4:3	65K	1G	√	√	√	-20~+70°C/0~+70°C	9~12V
	DST64048056-20T	640*480	4:3	65K	1G	x	√	√	-20~+70°C/0~+70°C	9~12V
	DST64048056-30	640*480	4:3	16.7M	1G	√	√	√	-20~+70°C/0~+70°C	9~12V
	DST64048056-30T	640*480	4:3	16.7M	1G	x	√	√	-20~+70°C/0~+70°C	9~12V
7.0	DST80048070-10	800*480	16:9	65K	1G	√	√	√	-20~+70°C/0~+70°C	9~12V
	DST80048070-10T	800*480	16:9	65K	1G	x	√	√	-20~+70°C/0~+70°C	9~12V
	DST80048070-20	800*480	16:9	16.7M	1G	√	√	√	-20~+70°C/0~+70°C	9~12V
	DST80048070-20T	800*480	16:9	16.7M	1G	x	√	√	-20~+70°C/0~+70°C	9~12V
8.0	DST80060080-10	800*600	4:3	65K	1G	√	√	√	-20~+70°C/0~+70°C	9~12V
	DST80060080-10T	800*600	4:3	65K	1G	x	√	√	-20~+70°C/0~+70°C	9~12V
	DST80060080-20	800*600	4:3	16.7M	1G	√	√	√	-20~+70°C/0~+70°C	9~12V
	DST80060080-20T	800*600	4:3	16.7M	1G	x	√	√	-20~+70°C/0~+70°C	9~12V
10.4	DST1024768104-10	1024*768	4:3	65K	1G	√	√	√	-20~+70°C/0~+70°C	9~12V
	DST1024768104-10T	1024*768	4:3	65K	1G	x	√	√	-20~+70°C/0~+70°C	9~12V
	DST1024768104-20	1024*768	4:3	16.7M	1G	√	√	√	-20~+70°C/0~+70°C	9~12V
	DST1024768104-20T	1024*768	4:3	16.7M	1G	x	√	√	-20~+70°C/0~+70°C	9~12V
Módulos de puerto serial VGA										
Modelo		Resolución	color	Almacenamiento (Bit)	Táctil	Audio	RTC	Temperatura de Operación		Voltaje de Operación
DSTVGA-10		1024*768, others	65K	1G	x	√	√	0~+70°C/-40~+85°C		9~12V
DSTVGA-20		1024*768, others	16.7M	1G	x	√	√	0~+70°C/-40~+85°C		9~12V

SECCIÓN VI DESCRIPCIÓN DEL CONJUNTO DE INSTRUCCIONES DAS HMI

6.1 Formato de instrucciones

El formato de un comando completo del usuario se muestra en la Figura 6.1. Si el número de parámetros de la instrucción sobre pasa de un 1 byte, la comunicación se hará todo por MSB (byte alto es el primero y el byte bajo sigue después). Los datos de instrucciones están representados por el sistema hexadecimal.

Tabla 6.1 Formato de Marcos de Instrucción

Instrucción	EE	XX	XX	FF FC FF FF
Descripción	Marco de cabecera	Instrucción	Parametros de instrucción	Fin de marco

Por ejemplo, mostrar una imagen se especifica en la coordenada (100, 50) como se muestra en la figura 6.1 a continuación:

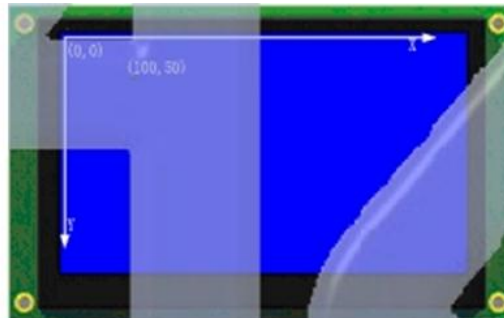


Figura 6.1 Mostrar una imagen en el área especificada (100,50)

El Comando de puerto serie para la transmisión de los usuarios de MCU se muestra a continuación:

Envía Comando: EE **【32 00 64 00 32 00 02 00】**FF FC FF FF

Análisis de Comando: EE representa el inicio de la trama.

32 representa la instrucción de pantalla para visualización de imágenes regionales.

00 64 00 32 representa la coordenada (x, y) = (100, 50), los bytes altos son primero.

00 02 representa el número de la imagen (un PC de supervisión distribuirá la identificación (ID) automáticamente a todas las imágenes).

00 representa ninguna necesidad de filtrado de color.



FF FC FF FF representa el fin de marco.

6.2 Formato de color

DST80048070-10soporta $2^{16} = 65.536$ colores (65K colores para abreviar), los bytes altos y bajos RGB se distribuyen como se muestra en la Tabla 6.2.

Tabla 6.2 Formato de distribución de color RGB

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Distribución de color	R					G					B					

Por ejemplo,  rojo puro=F800H,  azul puro=001FH.

6.3 Lista de instrucciones

La lista de instrucciones consta de dos partes, las definiciones de los parámetros de instrucciones y funciones de manejo. Cada instrucción tiene su función de manejo correspondiente. Los archivos de código fuente de la biblioteca de controladores incluyen hmi_driver.c y hmi_driver.h, consultar archivo adjunto 2. Actualmente, las plataformas de bibliotecas de controladores adoptan un tradicional chip de la máquina 8051. Por lo tanto, los usuarios que no utilizan la plataforma de 8051 se sustituyen por "Avisos de uso" en el archivo hmi_driver.c en primer lugar.

Tabla 6.3 lista de instrucciones

Tipo	Instrucción	Parámetros de instrucción	Descripción
Comando hand-shake	0x00	AA	Después HMI recibe el comando AA, enviará 55 al host con el fin de mostrar con éxito hand-shake. Función: void SetFcolor(uint16 fcolor);
Establecer color en primer plano.	0x41	Fcolor	Fcolor(2 bytes) se utiliza para especificar colores de puntos, líneas, círculos, gráficos y palabras. Función: void SetFcolor(uint16 fcolor);
Establecer color de fondo.	0x42	Bcolor	Bcolor(2 bytes) se utiliza para especificar los colores de fondo para la pantalla de limpieza y palabras. Función: void SetBcolor(uint16 bcolor);
Limpiar pantalla.	0x01	NA	El color de la pantalla de limpieza depende de la configuración de la paleta de colores de fondo. Función: void GUI_CleanScreen();
Ajuste de tono de fila y columna para textos.	0x43	X_W+ Y_W	X_W(1 byte) El tono de columna toma un punto como unidad y tiene un valor de 00~1F Y_W(1 byte) El tono de fila toma un punto como unidad y tiene un valor de 00~1F Función: void SetTextSpace(uint8 x_w, uint8 y_w);
Establecer el filtro del color.	0x44	FillColor_Dwon + FillColor_UP	FillColor_Dwon(2 bytes): valor limite inferior para el filtrado de colores. FillColor_UP(2 bytes): valor limite superior para el filtrado de colores. Observaciones: cuando los dos valores son idénticos, simplemente se filtra un color. Función: void SetFilterColor(uint16 fillcolor_dwon, uint16 fillcolor_up);

Texto de pantalla.	0x20	X+Y+Back+Font+Strings	<p>Ajustes de texto de pantalla</p> <p>X (2 bytes) : X-axis coordina el valor que toma un punto como unidad.</p> <p>Y (2 bytes) : Y-axis coordina el valor que toma un punto como unidad.</p> <p>Back (color de fondo, 1 byte)</p> <p>0x01: mostrar color de fondo 0x00: no mostrar el color de fondo</p> <p>Fuente (codificación de palabra,1 byte)</p> <p>0x00: 8x12 matriz de puntos (ASCII)</p> <p>0x01: 8x16 matriz de puntos (ASCII)</p> <p>0x02: 12x24 matriz de puntos (ASCII)</p> <p>0x03: 16x32 matriz de puntos (ASCII)</p> <p>0x04 12x12 matriz de puntos (GBK)</p> <p>0x05: 16x16 matriz de puntos (GBK)</p> <p>0x06: 24x24 matriz de puntos (GBK)</p> <p>0x07: 32x32 matriz de puntos (GB2312)</p> <p>0x08: 32x64 matriz de puntos (ASCII)</p> <p>0x09: 64x64 matriz de puntos (GB2312)</p> <p>Cadenas: se refieren a cadenas de caracteres escritos en los usuarios (byte alto es primero).</p> <p>Observaciones: el color de la fuente de texto es coherente con la del primer plano, y el color de fondo sirve como color de fondo.</p> <p>Función: void DisText(uint16 x, uint16 y,uint8 back,uint8,font,uchar *strings);</p>
Cursor de pantalla.	0x21	Enable+X+Y + Length+Width	<p>Habilitar/Inhabilitar un cursor</p> <p>Habilitar(1 byte) 0x00:inhabilitado 0x01:habilitado</p> <p>X(2 bytes): X-axis coordina el valor que toma un punto como unidad.</p> <p>Y(2 bytes): Y-axis coordina el valor que toma un punto como unidad.</p> <p>Largo(2 bytes): cursor largo</p> <p>Ancho(2 bytes): cursor ancho</p> <p>Observaciones: El color del cursor es coherente con las palabras.</p> <p>Por defecto, el tiempo de parpadeo dura 1s.</p> <p>Función: void DisCursor(uint8 enable,uint16 x, uint16y,uint16 length,uint16 width);</p>
Visualizacion de imagen.	0x31	Image_ID+MaskEn	<p>Visualizar imagen completa Image_ID (imagen No. ,2 bytes)</p> <p>MaskEn(1 byte) 0x00: no mostrar el filtro de color ; 0x01: muestra el filtro de color</p>
	0x32		<p>Observaciones: el color a filtrar depende de la configuración para el filtrado de color.</p> <p>Función: void DisFull_Image(uint16 image_id,uint8 masken);</p>
		X+Y+ Image_ID+MaskEn	<p>Visualización de la imagen Regional</p> <p>X (2 bytes): X-axis coordina el valor que toma un punto como unidad.</p> <p>Y(2 bytes) : Y-axis coordina el valor que toma un punto como unidad.</p> <p>Image_ID (imagen No.,2 bytes)</p> <p>MaskEn(1 byte) 0x00: no mostrar el filtro del color; 0x01: mostrar filtro del color</p> <p>Observaciones: el color a filtrar depende de la configuración para el filtrado de color</p>
			Función: void DisArea_Image(uint16 x,uint16 y,uint16 image_id,uint8 masken);

Cortar imagen.	0x33	X+Y+ Image_ID+ Image_X +Image_Y+ Image_L+ Image_W+ MaskEn	<p>Mostrar imágenes cortadas en el área especificada</p> <p>X(2 bytes): X-axis coordina el valor que toma un punto como unidad.</p> <p>Y(2 bytes): Y-axis coordina el valor que toma un punto como unidad.</p> <p>Image_ID(2 bytes): número de imagen</p> <p>Image_X(2 bytes): coordinar imagen interna X</p> <p>Image_Y(2 bytes): coordinar imagen interna Y</p> <p>Image_L(2 bytes): largo de corte</p> <p>Image_W(2 bytes): ancho de corte</p> <p>MaskEn(1 byte) 0x00: no mostrar filtro de color; 0x01: mostrar filtro de color</p> <p>Observaciones: el color a filtrar depende de la configuración para el filtrado de color</p> <p>Función: void DisCut_Image(uint16 x,uint16 y, uint16 image_id,uint16 image_x, uint16 image_y, uint16 image_l, uint16 image_w,uint8 masken);</p>
Visualización de imagen en flash.	0x80	X+Y+ FlashImage_ID + Enable+Play num	<p>X (2 bytes): X-axis coordina el valor que toma un punto como unidad.</p> <p>Y (2 bytes): Y-axis coordina el valor que toma un punto como unidad.</p> <p>FlashImage_ID (2 bytes)</p> <p>Observaciones: El formato de imagen flash, sólo soporta * gif.</p> <p>Función: void DisFlashImgae(uint16 x,uint16 y,uint16 flashimage_id);</p>
Dibujar punto.	0x50	X+Y	<p>X (2 bytes) : X-axis coordina el valor que toma un punto como unidad.</p> <p>Y (2 bytes) : Y-axis coordina el valor que toma un punto como unidad.</p> <p>Observaciones: valor del color depende de la configuración de la paleta de color de primer plano.</p> <p>Función: void GUI_Dot(uint16 x,uint16 y);</p>
Dibujar linea.	0x51	X0 +Y0+X1+ Y1	<p>X0 (2 bytes): inicio coordina valor del eje X rectilínea que tiene un punto como unidad</p> <p>Y0 (2 bytes): inicio coordina valor del eje X rectilínea que tiene un punto como unidad</p> <p>X1 (2 bytes):Fin coordina valor del eje X rectilínea que tiene un punto como unidad</p> <p>Y1 (2 bytes):Fin coordina valor del eje Y rectilíneas que toma un punto como unidad</p> <p>Observaciones: El valor del color depende de la configuración de la paleta de color de primer plano.</p> <p>Función: void GUI_Line(uint16 x0, uint16 y0, uint16 x1,uint16 y1);</p>
Dibuja un círculo.	0x52	X0+Y0+R	<p>X0 (2 bytes) : El centro del círculo Y toma un punto como unidad de coordenadas</p> <p>Y0 (2 bytes) : El centro del círculo Y toma un punto como unidad de coordenadas</p> <p>R (2 bytes) :Radio de un circulo</p> <p>Observaciones: El valor del color depende de la configuración de la paleta de color de primer plano.</p> <p>Función: void GUI_Circle(uint16 x0, uint16 y0, uint16 r);</p>

Dibuja un círculo sólido.	0x53	X0+Y0+R	<p>X0(2 bytes): El centro del círculo X toma un punto como unidad de coordenadas.</p> <p>Y0(2 bytes): El centro del círculo Y toma un punto como unidad de coordenadas.</p> <p>R (2 bytes): Radio de un círculo sólido.</p> <p>Observaciones: valor del color depende de la configuración de la paleta de color de primer plano.</p> <p>Función: void GUI_CircleFill(uint16 x0, uint16 y0, uint16 r);</p>
Dibuja un rectángulo.	0x54	X0+Y0+X1+Y1	<p>X0(2bytes):X coordina valor en la esquina superior izquierda de un rectángulo que tiene un punto de unidad.</p> <p>Y0(2bytes):Y coordina valor en la esquina superior izquierda de un rectángulo que tiene un punto de unidad.</p> <p>X1(2bytes):X coordina valor en la esquina superior derecha de un rectángulo que tiene un punto de unidad.</p> <p>Y1(2bytes):Y coordina valor en la esquina superior derecha de un rectángulo que tiene un punto de unidad.</p> <p>Observaciones: El alor del color depende de la configuración de la paleta de color de primer plano.</p> <p>Función: void GUI_Rectangle(uint16 x0, uint16 y0, uint16 x1,uint16 y1);</p>
Dibuja un rectángulo sólido.	0x55	X0+Y0+X1+Y1	<p>X0(2bytes):X coordina valor en la esquina superior izquierda de un rectángulo sólido que tiene un punto de unidad.</p> <p>Y0(2bytes):Y coordina valor en la esquina superior izquierda de un rectángulo sólido que tiene un punto de unidad.</p> <p>X1(2bytes):X coordina valor en la esquina superior derecha de un rectángulo sólido que tiene un punto de unidad.</p> <p>Y1(2bytes):Y coordina valor en la esquina superior derecha de un rectángulo sólido que tiene un punto de Unidad.</p> <p>Observaciones: El valor del color depende de la configuración de la paleta de color de primer plano.</p> <p>Función: void GUI_Rectangle(uint16 x0, uint16 y0, uint16 x1,uint16 y1);</p>
Dibuja una elipse.	0x56	X0+Y0+X1+Y1	<p>X0(2bytes): X coordina valor en el extremo superior de una elipse que tiene un punto como su unidad</p> <p>Y0(2bytes): Y coordina valor en el extremo superior de una elipse que tiene un punto como su unidad</p> <p>X1(2bytes): X coordina valor en el extremo inferior de una elipse que tiene un punto como su unidad</p> <p>Y1(2bytes): Y coordina valor en el extremo inferior de una elipse que tiene un punto como su unidad</p> <p>Descripción: valor del color depende de la configuración de color.</p> <p>Función: void GUI_Ellipse (uint16 x0, uint16 y0, uint16 x1,uint16 y1);</p>
Dibuja una elipse sólida.	0x57	X0+Y0+X1+Y1	<p>X0(2bytes):X coordina valor en el extremo superior de una elipse sólida que tiene un punto como su unidad.</p> <p>Y0(2bytes):Y coordina valor en el extremo superior de una elipse sólida que tiene un punto como su unidad.</p> <p>X1(2bytes): X coordina valor en el extremo inferior de un elipse sólida que tiene un punto como su unidad</p> <p>Y1(2bytes): Y coordina valor en el extremo inferior de un elipse sólida que tiene un punto como su unidad</p> <p>Descripción: valor del color depende de la configuración de la paleta de color de primer plano.</p> <p>Función: void GUI_EllipseFill (uint16 x0, uint16 y0, uint16 x1,uint16 y1);</p>
Ajuste de luz de fondo.	0x60	Light_level	<p>Ajusta el brillo de 255 niveles de luz de fondo;</p> <p>0x00: Habilitar luz de fondo 0xFF: brillo máximo de luz de fondo.</p> <p>Función: void SetBackLight(uint8 light_level);</p>
Control de timbre.	0x61	Enable + Time	<p>Habilitado(1 byte) 0x00:inhabilitado 0x01:habilitado</p> <p>Time(1 byte): la longitud de tiempo de timbre es 10 ms * Tiempo.</p> <p>Función: SetBuzzer(uint8 Enable, uint8 Time);</p>

Control de pantalla táctil.	0x70	Enable	Cargar valor 1: Activar "Cargar coordenadas de pantalla táctil" 0: Desactivar "Cargar coordenadas de pantalla táctil" Función: SetTouchScreen_Release(uint8 Enable)
	0x71	NA	Leer valor táctil Después de la pantalla táctil se suelta, los datos de la última área serán subidos. Cargar formato: EE (encabezado de la trama) + 0x01 (coordenada valor de marca de la actualización) + coordenada X + Y de coordenadas (2 bytes, bytes altos vienen primero) + trama final Función: SetTouchScreen_Press(void)
	0x72	NA	Ajuste de calibración de la pantalla táctil Función: SetTouchScreen_Adjust (void)
Operación de tiempo	0x85	Enable + Dis_Mode+ Text_Mode +Color + Xpoint+Ypoint	Ajustes de Atributos RTC Enable(1 byte) 0x00: inhabilitar RTC 0x01: Habilitar RTC Dis_Mode(1 byte) 0x00 : formato HH:MM:SS 0x01 : formato 20XX-MM-DD HH:MM:SS Text_Mode(1 byte) 0x00: fuente 0 0x01: fuente 1 0x02: fuente 2 0x03: fuente 3 Color (2 bytes) : Color de pantalla Xpoint(2 bytes) : coordenada X en la pantalla Ypoint(2 bytes) : coordenada Y en la pantalla Función: SetRTCEn(uint8 enable, uint8 Dis_Mode, uint8 Text_Mode, uint16 Color, uint16 Xpoint, uint16 Ypoint);
	0x81	Sec+Min+ Hour+Day+ Week+Mon+ Year	Ajuste de tiempo RTC Sec: Ajustar segundos; Min: ajustar minutos; Hour: Ajustar horas; Day: Ajustar día; Week: Ajustar mes semana; Month: Ajustar mes; Year: Ajustar año Observaciones: cada byte se representa por un código BCD. Ejemplo, lunes se establece como 0x001 Función: SetRTCtime(uint8 sec,uint8 min,uint8 hour,uint8 day,uint8 week,uint8 mon,uint8 year)
Leer RTC	0x82	NA	Formato de salida de datos: 0x82 + Sec+ Min+ Hour+ Day+ Week+ Mon+ Year Observaciones: cada byte es representado por un código BCD. Función: ReadRTCtime(void)
Ajuste de la velocidad de transmisión	0xA0	Baudset	Baudset(1 byte) : baud rate sequencing 0x00: 1200bps 0x01: 2400 0x02: 4800 0x03: 9600 0x04: 19200 0x05: 38400 0x06: 57600 0x07: 115200 Función: void SetCommBps(uint8 Baudset);

6.4 Explicaciones detalladas para la aplicación de las instrucciones

Las explicaciones detalladas para la aplicación de las instrucciones se muestran en los siguientes capítulos y secciones. En cuanto a las librerías de controladores, por favor consulte el Apéndice 2.

6.4.1 Comando HAND-SHAKE

- Transmisión de usuario: : EE 【00 AA】 FF FC FF FF
- Retorno de equipo: EE 【00 55】 FF FC FF FF

El comando HAND-SHAKE se usa para juzgar si HMI ha finalizado la inicialización de energía o la comunicación que sucede normalmente. Después de que el usuario ha enviado los parámetros de la instrucción AA, HMI devolverá el comando 55, lo que sugiere una conexión con éxito. En la aplicación práctica, este comando podría ser ignorado.

6.4.2 Asignar y obtener el color de fondo

Formato de instrucción: Ajuste color frontal: EE 【41 Fcolor】 FF FC FF FF

Ajuste color de fondo: EE 【42 Bcolor】 FF FC FF FF

La declaración de parámetros: FColor, Bcolor (2 bytes) son valores RGB de color frontal y color de fondo respectivamente.

Color frontal se utiliza para especificar el color de puntos, líneas, círculos, imágenes y palabras, mientras que el color de fondo se emplea para especificar el color de lavado y la palabra de fondo como se muestra en la Figura 6.2.

Programa de referencia:

```
{  
    SetBcolor(31); // Limpieza de la pantalla, color de fondo azul  
    GUI_CleanScreen( ); //Limpia pantalla  
    SetFcolor(65516); // Ajusta las palabras en color azul de la parte frontal  
    SetBcolor(63488); // Ajusta las palabras en color rojo de la parte de fondo  
    DisText(50, 50, 1, 6, "Hello, hello !"); //Escribe las palabras en las coordenadas (50,50)  
background  
    DisText(50, 90, 0, 6, "Hello,lucky !"); //Escribe las palabras en las coordenadas (50,90).  
background  
    SetFcolor(65400); //Reinicia el color frontal, especifica el color frontal para un elipse  
    GUI_Ellipse(22, 35, 219, 135); // Dibuja un elipse hueco  
    SetFcolor(2016); // Especifica el color de un rectángulo en azul como color frontal  
    GUI_RectangleFill(256, 57, 370, 116); //Dibuja un rectángulo solido  
}
```

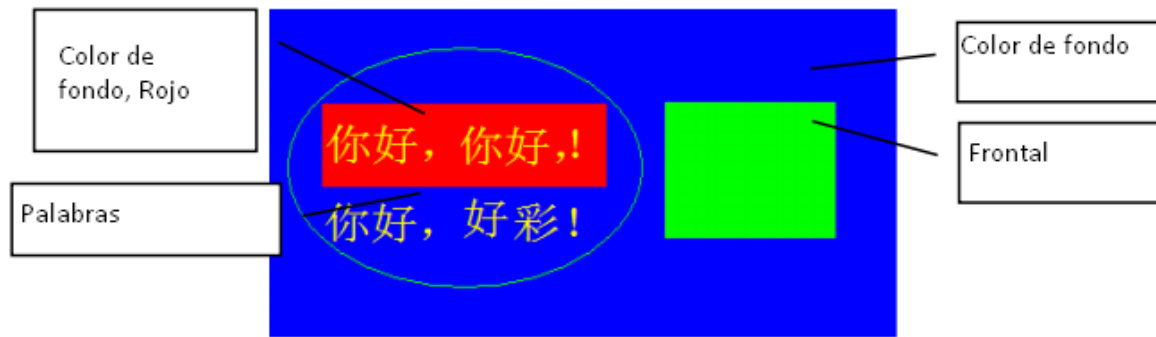


Figura 6.2- Explicación del color de fondo/frontal

6.4.3 Limpiar pantalla

Formato de instrucción: EE **【01】** FF FC FF FF

Declaración de parámetros: No hay parámetros

La instrucción se utiliza para especificar colores de la pantalla limpia. El color de lavado depende de la configuración de paleta de color de fondo.

6.4.4 Ajuste de tono en filas y columnas para las palabras

Formato de instrucción: EE **【43 X_W Y_W】** FF FC FF FF

Declaración de parámetros: X_W(1 byte) es un campo de la columna que tiene un punto como su unidad con un valor que va de 00 a 1F;
Y_W(1 byte) es un campo de fila que toma un punto como su unidad con un valor que va de 00 a 1F.

La instrucción se utiliza para establecer la columna de tono entre las palabras. Si sólo hay una columna de palabras, entonces el tono de la columna será 0. Por ejemplo, mostramos 2 columnas de 32 * 32 cadenas de caracteres, establece el tono de la columna como 24 y el tono de la fila como 16. El efecto de la exhibición será como se muestra en la figura 6.3.

Programa de referencia:

```
{
    SetBcolor(31); // Ajusta el color de fondo azul
    GUI_CleanScreen( ); //Limpia pantalla
    SetTextSpace(24, 16); // Ajuste de tono de la columna como 24 y el tono de la fila como 16
    SetFcolor(63488); // Ajusta color frontal de las palabras
    SetBcolor(65516); //Ajusta color de fondo de las palabras
    DisText(20, 50, 1, 7, "DAS IDUSTRY LIMITED Serial-port Display Terminal");
}
```

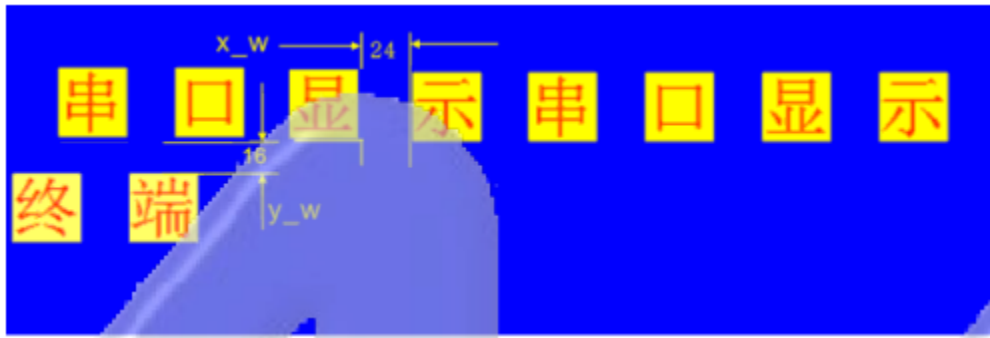



Figura 6.3- Pantalla de tono de la fila y el tono de la columna de las palabras

6.4.5 Ajuste de color Filtrado

Formato de instrucción: EE **【44 FillColor_Dwon FillColor_UP】** FF FC FF FF

Declaración de parámetros: FillColor_Dwon (2 bytes): valor límite inferior del filtrado de color, es decir, el valor mínimo;

FillColor_UP (2 bytes): valor límite superior de filtrado de color, es decir, el valor máximo.

Este comando se utiliza principalmente para especificar el filtrado de color en un área determinada. Cuando ambos valores son idénticos, un color será filtrado.

Principio de Filtrado: Después de que el valor límite inferior y el valor límite superior se establece correctamente, si algún valor píxel en una imagen podría cumplir con el área de filtración, este punto será blindado, y está prohibido escribir en la memoria de vídeo. Por lo tanto, el color de fondo no será cubierto. En cuanto al diagrama de comparación realizado antes y después de la filtración del color, por favor vea la figura 6.4

Programa de referencia:

```
{
    DisArea_Image(0, 0, 0, 0); // Mostrar una imagen como fondo de pastizales en la
    coordenada (0,0)
    DisArea_Image(61,130, 1, 0); // Mostrar una imagen mariposa sin filtrar en la coordenada
    (61,130)
    SetFilterColor(0, 0); // Ajuste valor de filtrado de color como 0 y el valor RGB blanco como
    0
    DisArea_Image(258,68, 2, 1); //Mostrar una imagen filtrada de mariposa en la coordenada
    (258,68)
}
```



Figura 6.4- Comparación Diagrama antes y después del filtrado de color

6.4.6 Visualización de Texto

Formato de instrucción: EE **【20 X Y Back Font Strings】** FF FC FF FF

Declaración de parámetros: X(2bytes): Eje X el valor que toma un punto es la unidad;

Y(2bytes): Eje Y el valor que toma un punto es la unidad;

Volver (color de fondo, 1 byte)

0x01: pantalla color de fondo 0x00: no muestra el color de fondo

Fuente (palabra stock codificación, 1 byte)

0x00: 8x12 matriz de puntos (ASCII)

0x01: 8x16 matriz de puntos (ASCII)

0x02: 12x24 matriz de puntos (ASCII)

0x03: 16x32 matriz de puntos (ASCII)

0x04: 12 x 12 matriz de puntos (GBK)

0x05: 16 x 16 matriz de puntos (GBK)

0x06: 24 x 24 matriz de puntos (GBK)

0x07: 32 x 32 matriz de puntos (GB2312)

Strings refieren a cadenas de caracteres escritos por los usuarios, los bytes altos son los primeros.

La instrucción se utiliza para mostrar los textos especificados en cualquier área de la pantalla. Stocks de palabras chinas incluyen, GBK y GB2312. Sin embargo, cadenas de palabras ASCII no pueden mostrar caracteres chinos.

En las aplicaciones prácticas, los usuarios ponían caracteres solo de entrada después de haber determinado el color frontal, color de fondo y la codificación de stock de palabra de textos. Entonces, el equipo establecerá nuevas líneas de forma automática o caracteres y caracteres chinos serán emparejados y se mostrarán alternativamente. En cuanto a la referencia del programa y el efecto real de la imagen, por favor consulte la Figura 6.5

Programa de referencia:

```
{
    DisFull_Image(0,0); // Mostrar el cielo azul como imagen de fondo
    SetFcolor(65516); // Ajusta color frontal de las palabras
    DisText(0, 4, 0, 5, " DST80048070-10, a 3.5 pulgadas con puerto serie HMI terminal de pantalla
    con una resolución de 320*240, disfruta de actuaciones bastante altas. Su posicionamiento en el
    mercado está muy dirigido y explícito, es decir, se dedica principalmente a la sustitución de un solo
    chip LCD en el mercado tradicional.");
    SetFcolor(63488); // Ajusta color frontal de las palabras
    SetBcolor(65504); //Ajusta color de fondo de las palabras
    DisText(8, 208, 1, 7, " Usted podría desarrollar verdaderas pantallas a color si usted sabe acerca de
    los puertos serie 32*32"); // Fuente 32*32
    DisText(88, 140, 1, 6, " Las innovaciones se originaron de DAS INDUSTRY LIMITED 24*24"); //Fuente
    24*24
    DisText(93, 89, 1, 5, " La personalización de acuerdo a las necesidades podrían satisfacer a todos los
    clientes 16*16 "); // Fuente 16*16
}
```



Figura 6.5- Imagen de texto en la pantalla

6.4.7 Visualización de Cursor

Formato de instrucción: EE 【21 Enable X Y Length Width】 FF FC FF FF

Declaración de parámetros: Enable (1 byte): 0x00: deshabilitado , 0x01: Habilitado

X (2bytes): Eje X el valor que toma un punto es la unidad;

Y (2bytes): Eje Y el valor que toma un punto es la unidad;

Length (2 bytes): Longitud del cursor

Width (2 bytes): Ancho del cursor

Con este comando los usuarios podrían habilitar y deshabilitar que el cursor parpadee. Por ejemplo, un usuario visualiza un cursor 16X8 con un carácter 24 * 24 chino al final. En cuanto a la referencia del programa y el efecto real de la imagen, por favor consulte la Figura 6.6

Código de programa de referencia:

```
{
    SetBcolor(31); //Ajusta el color de fondo azul
    GUI_CleanScreen( ); // Fondo azul para limpiar la pantalla
    SetFcolor(65516); // Ajusta color frontal de las palabras
    DisText(23, 24, 0, 6, " DAS ofrece a sus clientes servicios de diseño de arte libre "); //14
    24*24
    Caracteres chinos
    DisCursor(1,359,40,16,8); // Cursor parpadea pone un cursor 16 * 8 en las coordenadas
    (359,40)
}
```



Figura 6.6 Explicación de los parámetros del cursor.

6.4.8 Visualización de imagen

1. Visualización de la imagen completa

Formato de instrucción: EE 【31 Image_ID MaskEn】 FF FC FF FF

Declaración de parámetros: Image_ID (2 bytes): Número de la imagen

MaskEn (1 byte)

0x00: No filtra el color; 0x01: ejecuta el filtrado de color

Remarks: el color a ser filtrado depende de la configuración de color

2. Visualización de la imagen en una región

Formato de instrucción: EE 【32 X Y Image_ID MaskEn】 FF FC FF FF

Declaración de parámetros: X (2bytes): Eje X el valor que toma un punto es la unidad;

Y (2bytes): Eje Y el valor que toma un punto es la unidad;

Image_ID(2 bytes) : Numero de la imagen

MaskEn (1 byte)

0x00: No filtra el color; 0x01: ejecuta el filtrado de color

La instrucción se utiliza para realizar la visualización de imágenes en cualquier área.

Una vez que el tamaño de la imagen supera los límites de la pantalla, se mostrarán errores.

6.4.9 Cortar la imagen

Formato de instrucción: EE 【 33 X Y Image_ID Image_X Image_Y Image_L Image_WMaskEn】
FF FC FF FF

Declaración de parámetros: X (2bytes): Eje X el valor que toma un punto es la unidad;

Y (2bytes): Eje Y el valor que toma un punto es la unidad;

Image_ID (2 bytes): Número de la imagen

Image_X (2 bytes): imagen interna de coordenadas X

Image_Y (2 bytes): imagen interna de coordenadas Y

Image_L (2 bytes) : longitud de corte

Image_W (2 bytes) : Ancho de corte

MaskEn (1 byte)

0x00: No filtra el color;

0x01: ejecuta el filtrado de color

Remarks: el color a ser filtrado depende de la configuración de color

La instrucción se utiliza para realizar la visualización de imágenes cortadas en coordenadas especificadas. Los usuarios podrían cortar las imágenes almacenadas en Flash. Por ejemplo, para mostrar el plano 0-60, sólo tiene que cortar y combinar números correspondientes en una imagen, como se muestra en la Figura 6.7, a continuación, el color azul se filtra. Además, los comandos se aplican ampliamente en la recopilación de datos, mover, barra de proceso y dispositivos. Por ejemplo, un ave está volando en una imagen. En la operación práctica, cuando un pájaro vuela a una nueva posición de coordenadas, los usuarios deben actualizar el fondo de la última posición de coordenadas. Actualizaciones de corte por región, en comparación con la actualización de pantalla completa, tienen menos tiempo y sus imágenes aparecen mucho más suaves.

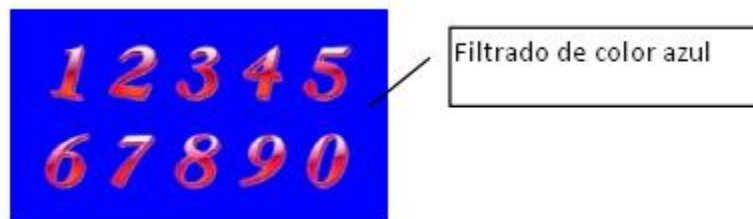


Figura 6.7- imágenes de número

En cuanto al contraste de parámetros de corte de la imagen, por favor refiérase a la Figura 6.8. El punto inicial (0,0) de la imagen interna de coordenadas (X, Y) se encuentra en la esquina superior izquierda.

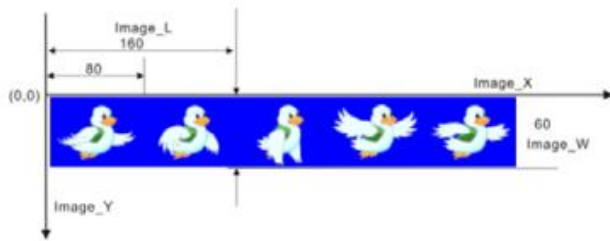


Figura 6.8

6.4.10 Visualización de imágenes Flash

Formato de instrucción: EE **【80 X Y FlashImgae_ID】** FF FC FF FF

Declaración de parámetros: X (2bytes): Eje X el valor que toma un punto es la unidad;

Y (2bytes): Eje Y el valor que toma un punto es la unidad;

FlashImgae_ID (2 bytes): Número de imagen de imagen flash

La instrucción se utiliza para realizar la visualización de imágenes flash * gif en cualquier área. Otros formatos de imagen flash de forma temporal no son compatibles. En cuanto a los efectos de las fotografías con flash, por favor consulte la Figura 6.9.

Código de referencia:

```
{
    DisFull_Image(0 , 0); // Imagen de bebe en pantalla complete, No.0
    DisFlashImgae(330,5, 1); // Inserte una imagen flash de vaca en la coordenada (330,5)
    DisArea_Image(13,8, 2, 0); // Mostrar el logotipo de DAS INDUSTRYLIMITED en la
    coordenada (13,8)
    DisCut_Image(343,137,0,95,30,92,116,0); // Cortar una imagen de 92X116 de la imagen 0
    en coordenadas (95,30),
    // A continuación, lo convierten en la exhibición en coordenadas (343, 137) de la pantalla
}
```



Figura 6.9- Visualización de efectos reales de imágenes flash

6.4.11 Dibujar un punto

Formato de instrucción: EE 【 50 X Y 】 FF FC FF FF

Declaración de parámetros: X (2bytes): Eje X el valor que toma un punto es la unidad;

Y (2bytes): Eje Y el valor que toma un punto es la unidad;

Este comando se utiliza principalmente para realizar el dibujo de un punto en cualquier área de la pantalla. El valor de color de los puntos está sujeto a ajustes de color frontal.

6.4.12 Dibujar líneas

Formato de instrucción: EE 【 51 X₀ Y₀ X₁ Y₁ 】 FF FC FF FF

Declaración de parámetros: X₀ (2 bytes): Comenzando el valor de las coordenadas en el eje X en forma rectilínea tiene un punto como unidad.

Y₀ (2 bytes): Comenzando el valor de las coordenadas en el eje Y en forma rectilínea tiene un punto como su unidad.

X₁ (2 bytes): Terminando el valor de las coordenadas en el eje X en forma rectilínea tiene un punto como su unidad.

Y₁ (2 bytes): Terminando el valor de las coordenadas en el eje Y en forma rectilínea tiene un punto como su unidad.

Este comando se utiliza sobre todo para realizar las líneas de dibujo entre dos puntos cualesquiera de la pantalla. El valor de color de las líneas está sujeto a ajustes de color frontal. En cuanto a las explicaciones de los parámetros, por favor consulte la Figura 6.10. El efecto real de las imágenes se muestra en la Figura 6.11.

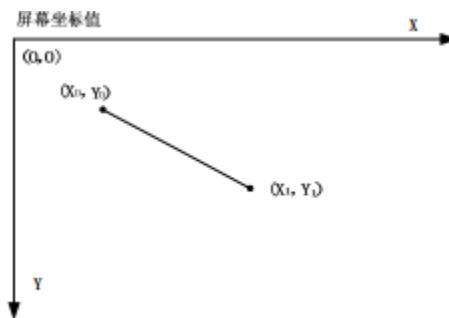


Figura 6.10- Explicación de los parámetros para dibujar líneas.

Código de referencia para el programa:

```
{  
    SetBcolor(31); // Ajusta el color de fondo en azul  
    GUI_CleanScreen( ); //Limpia pantalla  
    SetFcolor(65523); // Ajusta el color frontal de las líneas en amarillo  
    GUI_Rectangle(20, 10, 420, 70); // Dibuja un rectángulo vacío en la parte superior  
    // izquierda de coordenadas (20, 10) e inferior derecha de coordenadas (420, 70)
```



```

GUI_Line(20, 30, 420, 30); // Dibuja una línea recta desde la coordenada (20,30) a la
                             coordenada (420,30)
GUI_Line(20, 50, 420, 50); //Dibuja una línea recta desde la coordenada (20,50) a la
                             coordenada (420,50)
GUI_Line(61, 10, 61, 70); //Dibuja una línea recta desde la coordenada (61,10) a la
                             coordenada (61,70)
GUI_Line(211, 10, 211, 70); //Dibuja una línea recta desde la coordenada (210,10) a la
                             coordenada (211,70)
DisText(30, 13, 0, 4, "No."); // Escribe 12*12 caracteres chinos
DisText(38, 32, 0, 4, "1"); // Escribe 12*12 caracteres chinos
DisText(38, 52, 0, 4, "2"); // Escribe 12*12 caracteres chinos
}

```

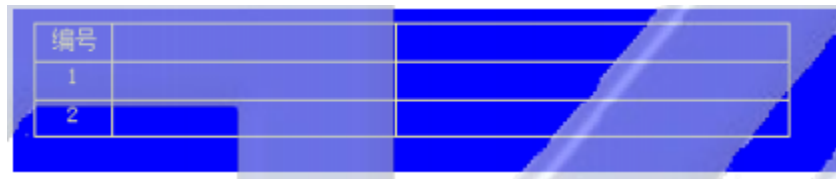


Figura 6.11- Efectos reales al dibujar una línea

6.4.13 Dibuja un círculo hueco

Formato de instrucción: EE 【 52 X0 Y0R 】 FF FC FF FF

Declaración de parámetros: X_0 (2 bytes): Valor de las coordenadas X para el centro del círculo
toma un punto como unidad

Y_0 (2 bytes): Valor de las coordenadas Y para el centro del círculo
toma un punto como unidad.

R (2 bytes): radio de un círculo hueco

La instrucción se utiliza para dibujar un círculo hueco con un radio R en determinadas coordenadas.

El color de las líneas del círculo está sujeto a ajustes de color frontal. Por favor, consulte la figura 6.12.

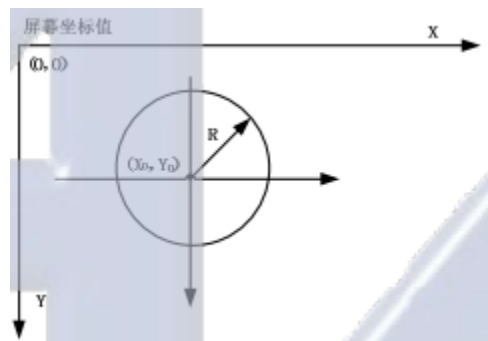


Figura 6.12- Declaración de parámetros para dibujar un círculo hueco

6.4.14 Dibujar un círculo solido

Formato de instrucción: EE 【 67 X₀ Y₀ R SA EA】 FF FC FF FF

Declaración de parámetros: : X₀ (2 bytes): Valor de las coordenadas X para el centro del círculo
toma un punto como unidad

Y₀ (2 bytes): Valor de las coordenadas Y para el centro del círculo
toma un punto como unidad.

R (2 bytes): radio de un círculo solido

La instrucción se utiliza para dibujar un círculo sólido con un radio R en determinadas coordenadas. El color de relleno está sujeto a ajustes de color frontal.

6.4.15 Dibujar un rectángulo hueco

Formato de instrucción: EE 【 54 X₀ Y₀ X₁ Y₁】 FF FC FF FF

Declaración de parámetros: X₀ (2 bytes): valor de coordenadas X en la esquina superior izquierda
de un rectángulo vacío que tiene un punto como unidad

Y₀ (2 bytes): Valor de coordenadas Y en la esquina superior izquierda
de un rectángulo vacío que tiene un punto como unidad

X₁ (2 bytes): Valor de coordenadas X en la esquina superior derecha
de un rectángulo vacío que tiene un punto como unidad

Y₁ (2 bytes): Valor de coordenadas Y en la esquina superior derecha de
un rectángulo vacío que tiene un punto como unidad

La instrucción se utiliza para dibujar un rectángulo vacío en cualquier área de la pantalla. El color de los marcos del rectángulo está sujeto a ajustes de color frontal. Por favor vea la figura 6.13 para las definiciones de los parámetros.

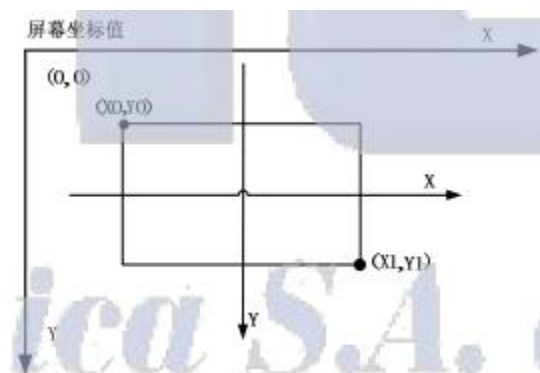


Figura 6.13 Explicación de los parámetros para dibujar un rectángulo vacío.

6.4.16 Dibujar un rectángulo sólido

Formato de instrucción: EE 【 55 X₀ Y₀ X₁ Y₁ 】 FF FC FF FF

Definición de parámetros: X₀ (2 bytes): valor de coordenadas X en la esquina superior izquierda de un rectángulo vacío que tiene un punto como unidad

Y₀ (2 bytes): Valor de coordenadas Y en la esquina superior izquierda de un rectángulo vacío que tiene un punto como unidad

X₁ (2 bytes): Valor de coordenadas X en la esquina superior derecha de un rectángulo vacío que tiene un punto como unidad

Y₁ (2 bytes): Valor de coordenadas Y en la esquina superior derecha de un rectángulo vacío que tiene un punto como unidad

La instrucción se utiliza para dibujar un rectángulo sólido en cualquier área de la pantalla. El color de relleno del rectángulo está sujeto a ajustes de color frontal.

6.4.17 Dibujar una elipse hueca

Formato de instrucción: EE 【 56 X₀ Y₀ X₁ Y₁ 】 FF FC FF FF

Declaración de parámetros: X₀ (2 bytes): Valor máximo de coordenadas X en el extremo izquierdo de un elipse hueco que tiene un punto como unidad

Y₀ (2 bytes): Valor máximo de coordenadas Y en el extremo superior de un elipse hueco que tiene un punto como unidad

X₁ (2 bytes): Valor máximo de coordenadas X en el extremo derecho de un elipse hueco que tiene un punto como unidad

Y₁ (2 bytes): Valor máximo de coordenadas Y en el extremo inferior de un elipse hueco que tiene un punto como unidad

La instrucción se utiliza para dibujar una elipse hueca en cualquier área de la pantalla. El color de los marcos de una elipse está sujeto a ajustes de color frontal. En cuanto a las definiciones de los parámetros, por favor consulte la Figura 6.14.

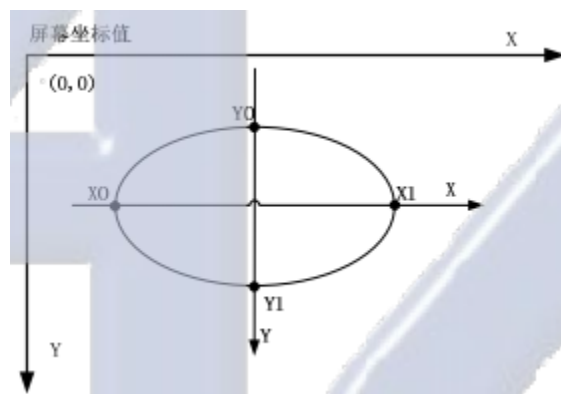


Figura 6.14- Explicación de los parámetros para dibujar una elipse hueca

6.4.18 Dibujar una elipse sólida.

Formato de instrucción: EE 【 57 X₀ Y₀ X₁ Y₁ 】 FF FC FF FF

Declaración de parámetros: X₀ (2 bytes): Valor máximo de coordenadas X en el extremo izquierdo de un elipse solido que tiene un punto como unidad

Y₀ (2 bytes): Valor máximo de coordenadas Y en el extremo superior de un elipse solido que tiene un punto como unidad

X₁ (2 bytes): Valor máximo de coordenadas X en el extremo derecho de un elipse solido que tiene un punto como unidad

Y₁ (2 bytes): Valor máximo de coordenadas Y en el extremo inferior de un elipse solido que tiene un punto como unidad

La instrucción se utiliza para dibujar una elipse sólida en cualquier área de la pantalla. El color de relleno de una elipse está sujeto a ajustes de color frontal

6.4.19 Ajuste del backlight

Formato de instrucción: EE 【 60 Light_level 】 FF FC FF FF

Declaración de parámetros: Light_level (1 byte) : 255-level ajustar luz de fondo.

Esta instrucción se utiliza principalmente para ajustar el brillo de la pantalla LCD con un valor que va desde 00H a FFH. 00H: activar de nuevo la luz, FFH: luz de fondo tiene su máxima brillantes.

6.4.20 Control del Buzzer

Formato de instrucción: EE 【 61 Enable Time 】 FF FC FF FF

Declaración de parámetros: Enable (1 byte):0x00: inhabilita buzzer , 0x01:Habilita buzzer

Time (1 byte): la longitud de tiempo del buzzer es 10 ms * Tiempo.

La instrucción se utiliza para controlar un Buzzer. El sonido de un timbre en diferentes frecuencias se podría establecer por parámetros de tiempo

6.4.21 Cargar valor táctil

Formato de instrucción: EE 【 70 Eneble】 FF FC FF FF

Declaración de parámetros: Enable (1 byte):0x00: desactivar la carga de coordenadas de la pantalla táctil.

0x01: permitir que la carga de coordenadas de la pantalla táctil.

La instrucción se utiliza para activar o desactivar la carga de coordenadas de la pantalla táctil. Una vez está habilitada la subida, HMI coordinará la salida de pantalla táctil con el actual valor de los puertos serie. Si no, se prohibirá dicha salida.

6.4.22 Leer valor táctil

Formato de instrucción: EE 【 71 】 FF FC FF FF

La instrucción de toque activada, activar / desactivar el timbre, y la carga de datos. Para cargar el formato de valores táctil: headers+ 0x01 + coordenada X + coordenada Y + extremo del bastidor. X y Y cada 2 bytes, cuanto más alto es en el frente. Al igual que la imagen que se muestra en la figura 6.15, el usuario pulsa la posición de la pantalla (50, 100), la matriz del puerto serie externo HMI es: EE 01 【00 32 00 64】 FF FC FF FF. MCU del usuario es determinar el tamaño del valor de coordenadas y el rango punto de contacto actual. Durante dicho proceso, HMI ha realizado el muestreo y el cálculo sobre el contacto táctil para varias veces, MCU no será necesario llevar a cabo un segundo cómputo

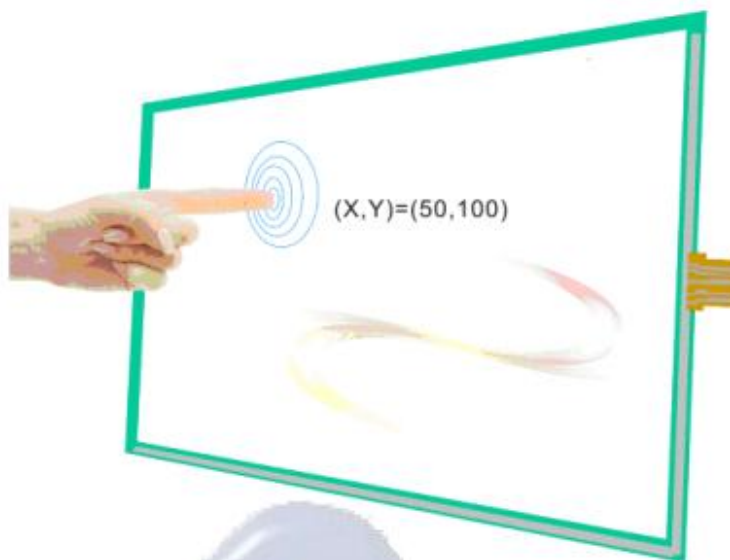


Figura 6.15- Lee el valor de las coordenadas de la pantalla táctil

6.4.23 Calibración de la pantalla táctil

Formato de instrucción: EE 【 72 】 FF FC FF FF

Declaración de parámetros: NA

La instrucción se utiliza para la calibración de la pantalla táctil. Cuando se opera la pantalla táctil, por primera vez, los usuarios deben realizar la calibración. Después se envía este comando, haga clic en el punto del cursor correspondiente según lo notificado en la pantalla. Por favor refiérase a la Figura 6.21, después de hacer clic en 3 puntos diferentes, el dispositivo le dirá si la calibración es exitosa, o hay necesidad de volver a calibrar. Todos los productos HMI producidos por DAS INDUSTRY LIMITED se han instalado con la calibración táctil, el usuario también puede volver a calibrar con el menú de la barra de herramientas del Visual TFT software.

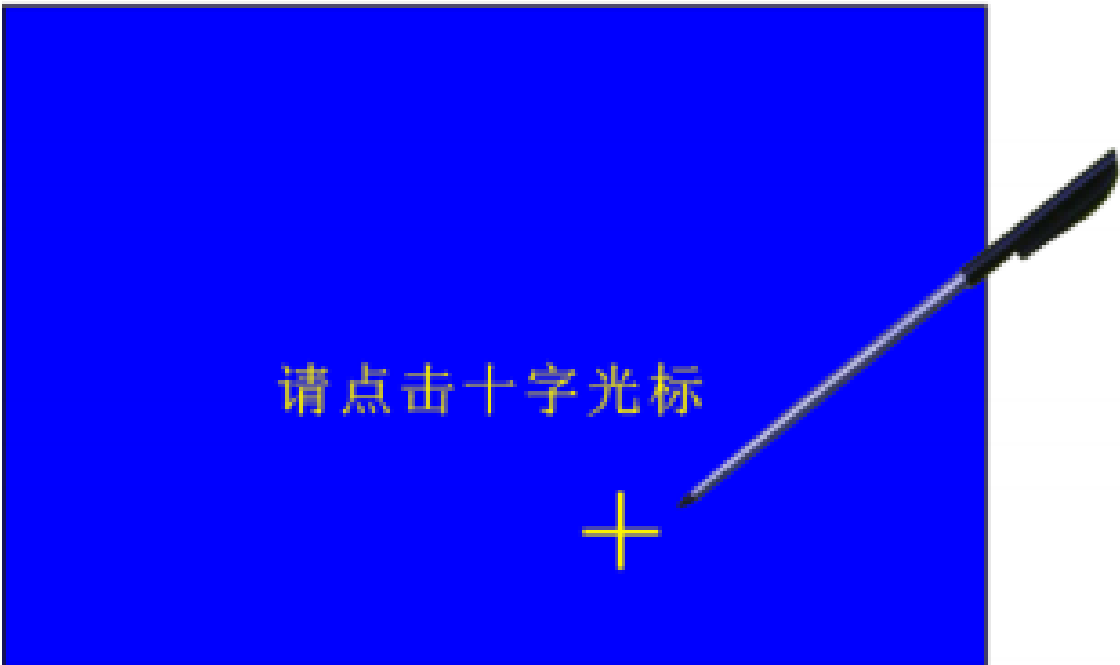


Figura 6.21- Calibración de la pantalla táctil

SECCION VI Circuitos y Programas de Referencia

7.1 Diagrama de referencia del MCU y conexión HMI

Si un usuario adopta una comunicación de nivel TTL / CMOS, podría pasar directamente 232 niveles de conmutación al chip, y el punto de soldadura cortocircuito para las comunicaciones en la parte posterior del panel de operador. Tomemos el circuito RS232 como un ejemplo, por favor consulte la Figura 7.1

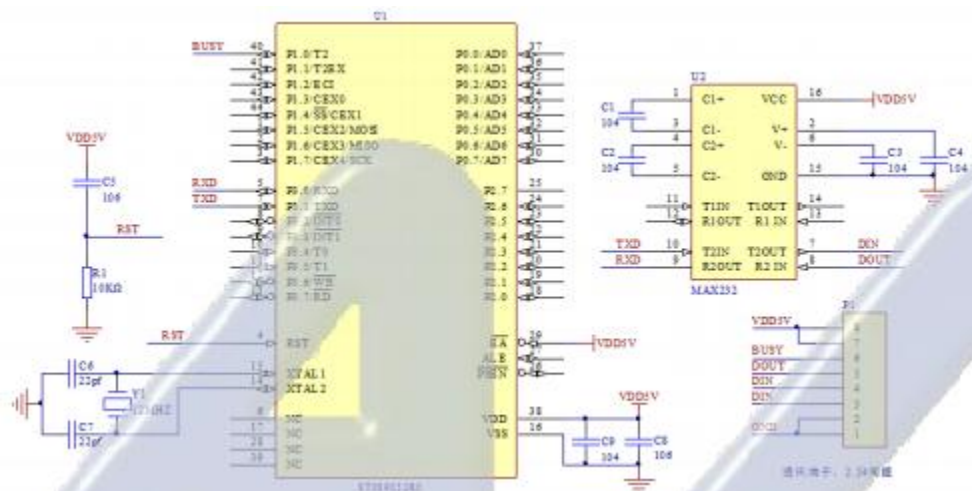


Figura 7.1- Diagrama de referencia del MCU y conexión HMI

7.2 MCU 8051 controla los procedimientos de visualización HMI y referencia del programa

Por ejemplo, si un usuario quiere mostrar una imagen en la pantalla, como se muestra en la Figura 7.2, se deberá cumplir con los procedimientos siguientes:



Figura 7.2- La imagen de un usuario que desea mostrar

3. Ejecute el software de Visual TFT, y terminar de editar el contenido en una página como se muestra en la Figura 7.3 En las aplicaciones prácticas, si se requieren las interfaces multi-nivel, tiene que crear varios archivos de página. En cuanto a la introducción detallada de las operaciones de software, por favor consulte el manual Comienza con Visual TFT.

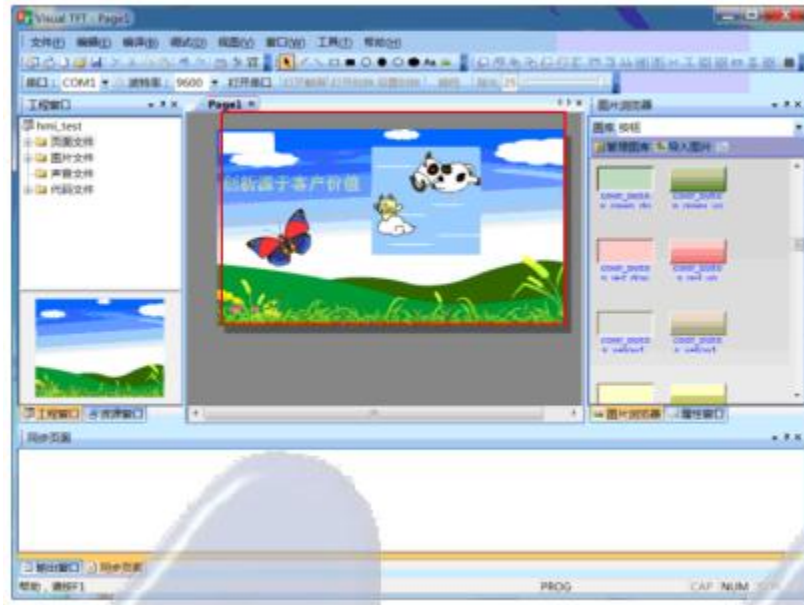


Figura 7.3- Pantalla de edición de imágenes

Habilidades arte-diseño: a excepción de la imagen (vaca.GIF) que es un recurso flash dinámico en la página, el resto son imágenes estáticas incluyendo pastizales, mariposa, palabras y logotipo. Los usuarios pueden hacer clic derecho en la página y guardar estos cuatro objetos como una imagen. Para ello, se podría ahorrar capacidad de almacenamiento y reducir los puertos serie para enviar instrucciones.

4. Haga clic en Compilar y botón de descarga para descargar todas las imágenes en HMI. Después de hacer la compilación, el software generará código del controlador page.c para el archivo de página. Por favor refiérase a la figura 7.4



Figura 7.4- El software genera la función page1.c

5. Añada las bibliotecas de controladores hmi_driver.c, hmi_driver.h y archivo de página page1.c al proyecto de aplicación, como se muestra en la Figura 7.5, a continuación, hacer modificaciones de acuerdo a las necesidades reales.

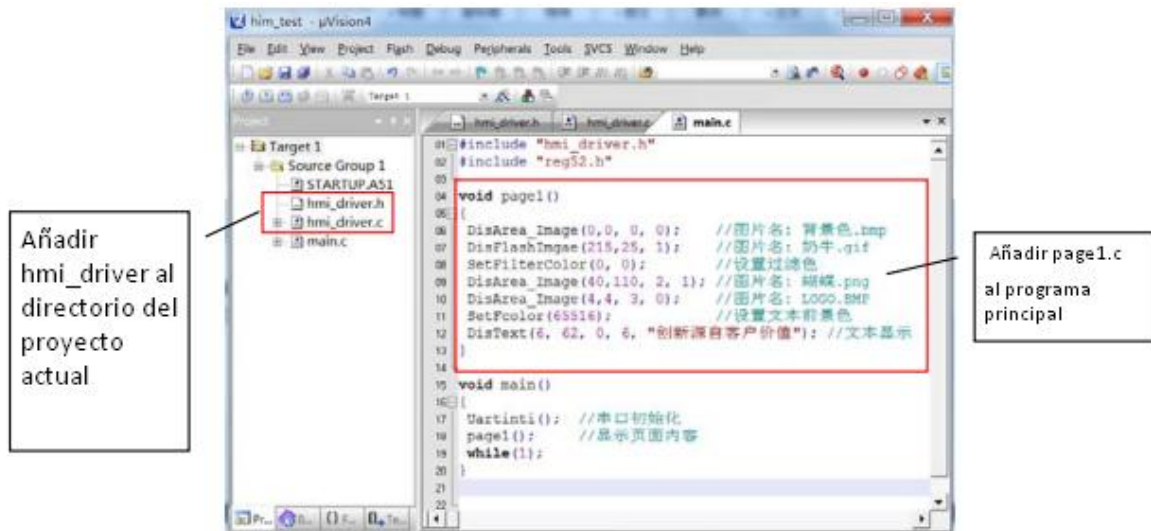


Figura 7.5- Agrega controladores al proyecto

6. Hasta el momento, el conjunto de procedimientos se han terminado. En cuanto a los programas de referencia, por favor consulte el Anexo A.

MCU 8051 Controlador pantalla HMI (Apéndice A)

```
##include "hmi_driver.h"
#include "reg52.h"
void page1()
{
SetBcolor(31); /* Set background color of the page*/
GUI_CleanScreen( ); /*The newly-created page will be, by default, in blue and have a command for page
cleaning, users could delete them*/
DisArea_Image(0,0, 0, 0); /*Display an image (ID=0) at coordinate (0,0), no filtering*/
DisFlashImgae(215,25, 1); /*Display a gif flash image (ID=1) at coordinate (215, 25)*/
SetFilterColor(0, 0); /*Set color filtering, pure white RGB=0*/
DisArea_Image(40,110, 2, 1); /*Display an image (ID=2) at coordinate (40,110), filter out the white color*/
DisArea_Image(4,4, 3, 0); /*Display an image (ID=3) at coordinate (4,4) , no filtering*/
SetFcolor(65516); /*Set words and foreground color in yellow*/
DisText(6, 62, 0, 6, "Innovations are originated from client value"); /*Display 24*24 character strings at
coordinate (6,62), words do not have its background color*/
}
void main( )
{
Uartinti( ); // Initialize serial port
page1(); //Display page contents
while(1);
}
/*-----File Information-----
** File name: hmi_driver.c
** Modification date: 2011-02-19
** File description: DAS INDUSTRY LIMITED HMI driver library file
** Technical support: Tel:8621-51082662 Email:sales@das-tek.com
**-----Welcome to use HMI products developed by DAS INDUSTRY LIMITED
-----**/
/*-----
Notices for Use
hmi_driver.c contains serial port transmission function and "DAS" driver library function. The serial-port
transmission function, a typical case of 8051, four functions altogether, namely Initialize serial port (Uartinti()),
Send 1 byte (SendChar()), Send an array with a specified length (SendDatas()), and Send character strings
(SendStrings()). If user's MCU does not belong to a 8051 platform, serial-port functions need to be modified to
correspond with a register, but function name shall not be modified. If yes, it could not match HMI driver libraries.
-----**/

#include "hmi_driver.h"
/*****
* Name: Uartinti()
* Function: Initialize serial port
* Inlet Parameter: NA
* Outlet Parameter: NA
*****/
void Uartinti()
{
SCON=0x50; //8-bit UART and BPS are variable. They allow control bit for serial-port reception
TMOD |=0x20; //Timer 1 operation mode 2 (8-bit auto-reload mode)
PCON |=0x80; // Double baud rate
TH1=0xF4;
TL1=0xF4 //9600 baud rate , 22.1184M crystal
IE |=0x90; //Enable Serial Interrupt
TR1=1;
ES=0;
}
/*****
* Name: SendChar()
* Function: send 1 byte
* Inlet Parameter: t bytes sent
*****/
```

```

* Outlet Parameter: NA
*****/
void SendChar(uchar t)
{
SBUF=t;
while(TI==0);
TI=0;
}
/*****

* Name: SendDatas()
* Function: send an array with a specified length
* Inlet Parameter: sdata arrays sent
* lenth length of arrays
* Outlet Parameter: NA
*****/
void SendDatas(uchar *sdata,uchar lenth)
{
uchar i;
for(i=0;i<lenth;i++){
SBUF=*sdata;
while(TI==0);
TI=0;
sdata++;
}
}
/*****

* Name: SendStrings()
* Function: send character strings
* Inlet Parameter: str character strings sent
* Outlet Parameter: NA
*****/
void SendStrings(uchar *str)
{
while(*str)
{
SendChar(*str);
str++;
}
}
/****

Driver libraries of HMI Products Developed by DAS INDUSTRY LIMITED
Suitable for All "DAS" HMI Product Series
*****

* Name: SendBeginCMD()
* Function: send frame header command
* Inlet Parameter: NA
* Outlet Parameter: NA
*****/
void SendBeginCMD()
{
SendChar(0xEE);
}
/*****

* Name: SendEndCmd()
* Function: send frame end command
* Inlet Parameter: NA
* Outlet Parameter: NA
*****/
void SendEndCmd()
{

```

```

SendChar(0xFF);
SendChar(0xFC);
SendChar(0xFF);
SendChar(0xFF);
}
/*****
* Name: SetHandShake()
* Function: handshake
* Inlet Parameter: NA
* Outlet Parameter: 55 suggests successful handshake, or out-of-order
*****/
uint8 SetHandShake(void)
{
SendBeginCMD();
SendChar(0xAA);
SendEndCmd();
}
/*****
* Name: SetFcolor()
* Function: Set foreground color
* Inlet Parameter: fcolor RGB parameters of foreground color
* Outlet Parameter: NA
*****/
void SetFcolor(uint16 fcolor)
{
SendBeginCMD();
SendChar(0x41);
SendChar((fcolor>>8)&0xff);
SendChar(fcolor&0xff);
SendEndCmd();
}
/*****
* Name: SetBcolor()
* Function: Set background color
* Inlet Parameter: bcolor RGB parameters of background color
* Outlet Parameter: NA
*****/
void SetBcolor(uint16 bcolor)
{
SendBeginCMD();
SendChar(0x42);
SendChar((bcolor>>8)&0xff);
SendChar(bcolor&0xff);
SendEndCmd();
}
/*****
* Name: GUI_CleanScreen()
* Function: clean screen
* Inlet Parameter: NA
* Outlet Parameter: NA
*****/
void GUI_CleanScreen()
{
SendBeginCMD();
SendChar(0x01);
SendEndCmd();
}
/*****
* Name: SetTextSpace()
* Function: set column pitch and column pitch of words
* Inlet Parameter: x_w row pitch that takes a point as its unit, has a value of 00~1F

```

```

y_w column pitch that takes a point as its unit, has a value of 00~1F
* Outlet Parameter: NA
*****/
void SetTextSpace(uint8 x_w, uint8 y_w)
{
SendBeginCMD();
SendChar(0x43);
SendChar(x_w);
SendChar(y_w);
SendEndCmd();
}
*****/
* Name: SetFilterColor()
* Function: set color filtering
* Inlet Parameter: fillcolor_dwon lower-limit value for filtering out colors
fillcolor_up upper-limit value for filtering out colors
* Outlet Parameter: NA
*****/
void SetFilterColor(uint16 fillcolor_dwon, uint16 fillcolor_up)
{
SendBeginCMD();
SendChar(0x44);
SendChar((fillcolor_dwon>>8)&0xff);
SendChar(fillcolor_dwon&0xff);
SendChar((fillcolor_up>>8)&0xff);
SendChar(fillcolor_up&0xff);
SendEndCmd();
}
*****/
* Name: DisText()
* Function: text display
* Inlet Parameter: x X-axis coordinate value that takes a point as its unit
y Y-axis coordinate value that takes a point as its unit
back enable background color
font word stock encoding
strings strings of Chinese characters written
* Outlet Parameter: NA
*****/
void DisText(uint16 x, uint16 y, uint8 back, uint8 font, uchar *strings )
{
SendBeginCMD();
SendChar(0x20);
SendChar((x>>8)&0xff);
SendChar(x&0xff);
SendChar((y>>8)&0xff);
SendChar(y&0xff);
SendChar(back);
SendChar(font);
SendStrings(strings);
SendEndCmd();
}
*****/
* Name: DisCursor()
* Function: cursor display
* Inlet Parameter: enable
x X-axis coordinate value that takes a point as its unit
y Y-axis coordinate value that takes a point as its unit
length cursor length
width cursor width
* Outlet Parameter: NA
*****/

```

```

void DisCursor(uint8 enable,uint16 x, uint16 y,uint16 length,uint16 width )
{
SendBeginCMD();
SendChar(0x21);
SendChar(enable);
SendChar((x>>8)&0xff);
SendChar(x&0xff);
SendChar((y>>8)&0xff);
SendChar(y&0xff);
SendChar((length>>8)&0xff);
SendChar(length&0xff);
SendChar((width>>8)&0xff);
SendChar(width&0xff);
SendEndCmd();
}
/*****
* Name: DisFull_Image()
* Function: Full image display
* Inlet Parameter: image_id image ID number
masken enable mask color
* Outlet Parameter: NA
*****/
void DisFull_Image(uint16 image_id,uint8 masken)
{
SendBeginCMD();
SendChar(0x31);
SendChar((image_id>>8)&0xff);
SendChar(image_id&0xff);
SendChar(masken);
SendEndCmd();
}
/*****
* Name: DisArea_Image()
* Function: Regional image display
* Inlet Parameter: x X-axis coordinate value that takes a point as its unit
y Y-axis coordinate value that takes a point as its unit
image_id image ID number
masken enable mask number
* Outlet Parameter: NA
*****/
void DisArea_Image(uint16 x,uint16 y,uint16 image_id,uint8 masken)
{
SendBeginCMD();
SendChar(0x32);
SendChar((x>>8)&0xff);
SendChar(x&0xff);
SendChar((y>>8)&0xff);
SendChar(y&0xff);
SendChar((image_id>>8)&0xff);
SendChar(image_id&0xff);
SendChar(masken);
SendEndCmd();
}
/*****
* Name: DisCut_Image()
* Function: cut image
* Inlet Parameter: x X-axis coordinate value that takes a point as its unit
y Y-axis coordinate value that takes a point as its unit
image_id image number
image_x image's internal X coordinate
image_y image's internal Y coordinate
*****/

```

```

image_l cutting length
image_w cutting width
masken enable mask color
* Outlet Parameter: NA
*****/

void DisCut_Image(uint16 x,uint16 y,uint16 image_id,uint16 image_x,uint16 image_y,
uint16 image_l, uint16 image_w,uint8 masken)
{
SendBeginCMD();
SendChar(0x33);
SendChar((x>>8)&0xff);
SendChar(x&0xff);
SendChar((y>>8)&0xff);
SendChar(y&0xff);
SendChar((image_id>>8)&0xff);
SendChar(image_id&0xff);
SendChar((image_x>>8)&0xff);
SendChar(image_x&0xff);
SendChar((image_y>>8)&0xff);
SendChar(image_y&0xff);
SendChar((image_l>>8)&0xff);
SendChar(image_l&0xff);
SendChar((image_w>>8)&0xff);
SendChar(image_w&0xff);
SendChar(masken);
SendEndCmd();
}
*****/

* Name: DisFlashImgae()
* Function: flash image display
* Inlet Parameter: x X-axis coordinate value that takes a point as its unit
y Y-axis coordinate value that takes a point as its unit
flashimage_id ID number of GIF flash image
* Outlet Parameter: NA
*****/

void DisFlashImgae(uint16 x,uint16 y,uint16 flashimage_id)
{
SendBeginCMD();
SendChar(0x80);
SendChar((x>>8)&0xff);
SendChar(x&0xff);
SendChar((y>>8)&0xff);
SendChar(y&0xff);
SendChar((flashimage_id>>8)&0xff);
SendChar(flashimage_id&0xff);
SendEndCmd();
}
*****/

* Name: GUI_Dot()
* Function: draw point
* Inlet Parameter: x X-axis coordinate value that takes a point as its unit
y Y-axis coordinate value that takes a point as its unit
* Outlet Parameter: NA
*****/

/
void GUI_Dot(uint16 x,uint16 y)
{
SendBeginCMD();
SendChar(0x50);
SendChar((x>>8)&0xff);
SendChar(x&0xff);

```

```

SendChar((y>>8)&0xff);
SendChar(y&0xff);
SendEndCmd();
}
/*****
* Name: GUI_Line()
* Function: draw lines
* Inlet Parameter: x0 X-axis coordinate value that takes a point as its unit
y0 Y-axis coordinate value that takes a point as its unit
x1 X-axis coordinate value that takes a point as its unit
y1 Y-axis coordinate value that takes a point as its unit
* Outlet Parameter: NA
*****/
void GUI_Line(uint16 x0, uint16 y0, uint16 x1, uint16 y1)
{
SendBeginCMD();
SendChar(0x51);
SendChar((x0>>8)&0xff);
SendChar(x0&0xff);
SendChar((y0>>8)&0xff);
SendChar(y0&0xff);
SendChar((x1>>8)&0xff);
SendChar(x1&0xff);
SendChar((y1>>8)&0xff);
SendChar(y1&0xff);
SendEndCmd();
}
/*****
* Name: GUI_Circle()
* Function: draw a hollow circle
* Inlet Parameter: x0 X-axis coordinate value that takes a point as its unit
y0 Y-axis coordinate value that takes a point as its unit
r radius of a hollow circle
* Outlet Parameter: NA
*****/
void GUI_Circle(uint16 x0, uint16 y0, uint16 r)
{
SendBeginCMD();
SendChar(0x52);
SendChar((x0>>8)&0xff);
SendChar(x0&0xff);
SendChar((y0>>8)&0xff);
SendChar(y0&0xff);
SendChar((r>>8)&0xff);
SendChar(r&0xff);
SendEndCmd();
}
/*****
* Name: GUI_CircleFill()
* Function: draw a solid circle
* Inlet Parameter: x0 X-axis coordinate value that takes a point as its unit
y0 Y-axis coordinate value that takes a point as its unit
r radius of a solid circle
* Outlet Parameter: NA
*****/
void GUI_CircleFill(uint16 x0, uint16 y0, uint16 r)
{
SendBeginCMD();
SendChar(0x53);
SendChar((x0>>8)&0xff);
SendChar(x0&0xff);

```

```

SendChar((y0>>8)&0xff);
SendChar(y0&0xff);
SendChar((r>>8)&0xff);
SendChar(r&0xff);
SendEndCmd();
}
/*****
* Name: GUI_Rectangle()
* Function: draw a hollow rectangle
* Inlet Parameter: x0 X coordinate value in top left corner of a hollow rectangle that takes a point as its unit
y0 Y coordinate value in top left corner of a hollow rectangle that takes a point as its unit
x1 X coordinate value in top right corner of a hollow rectangle that takes a point as its unit
y1 Y coordinate value in top right corner of a hollow rectangle that takes a point as its unit
* Outlet Parameter: NA
*****/
void GUI_Rectangle(uint16 x0, uint16 y0, uint16 x1, uint16 y1 )
{
SendBeginCMD();
SendChar(0x54);
SendChar((x0>>8)&0xff);
SendChar(x0&0xff);
SendChar((y0>>8)&0xff);
SendChar(y0&0xff);
SendChar((x1>>8)&0xff);
SendChar(x1&0xff);
SendChar((y1>>8)&0xff);
SendChar(y1&0xff);
SendEndCmd();
}
/*****
* Name: RectangleFill()
* Function: draw a solid rectangle
* Inlet Parameter: x0 X coordinate value in top left corner of a solid rectangle that takes a point as its unit
y0 Y coordinate value in top left corner of a solid rectangle that takes a point as its unit
x1 X coordinate value in top right corner of a solid rectangle that takes a point as its unit
y1 Y coordinate value in top right corner of a solid rectangle that takes a point as its unit
* Outlet Parameter: NA
*****/
void GUI_RectangleFill(uint16 x0, uint16 y0, uint16 x1, uint16 y1 )
{
SendBeginCMD();
SendChar(0x55);
SendChar((x0>>8)&0xff);
SendChar(x0&0xff);
SendChar((y0>>8)&0xff);
SendChar(y0&0xff);
SendChar((x1>>8)&0xff);
SendChar(x1&0xff);
SendChar((y1>>8)&0xff);
SendChar(y1&0xff);
SendEndCmd();
}
/*****
* Name: GUI_Ellipse()
* Function: draw a hollow ellipse
* Inlet Parameter: x0 X coordinate value at maximum left end of a hollow ellipse that takes a point as its unit
y0 Y coordinate value at maximum top end of a hollow ellipse that takes a point as its unit
x1 X coordinate value at maximum right end of a hollow ellipse that takes a point as its unit
y1 Y coordinate value at maximum lower end of a hollow ellipse that takes a point as its unit
* Outlet Parameter: NA *****/
void GUI_Ellipse (uint16 x0, uint16 y0, uint16 x1, uint16 y1 )

```



```

{
SendBeginCMD();
SendChar(0x56);
SendChar((x0>>8)&0xff);
SendChar(x0&0xff);
SendChar((y0>>8)&0xff);
SendChar(y0&0xff);
SendChar((x1>>8)&0xff);
SendChar(x1&0xff);
SendChar((y1>>8)&0xff);
SendChar(y1&0xff);
SendEndCmd();
}
/*****
* Name: GUI_EllipseFill()
* Function: draw a solid ellipse
* Inlet Parameter: x0 X coordinate value at maximum left end of a solid ellipse that takes a point as its unit
y0 Y coordinate value at maximum top end of a solid ellipse that takes a point as its unit
x1 X coordinate value at maximum right end of a solid ellipse that takes a point as its unit
y1 Y coordinate value at maximum lower end of a solid ellipse that takes a point as its unit
* Outlet Parameter: NA
*****/
void GUI_EllipseFill (uint16 x0, uint16 y0, uint16 x1, uint16 y1 )
{
SendBeginCMD();
SendChar(0x57);
SendChar((x0>>8)&0xff);
SendChar(x0&0xff);
SendChar((y0>>8)&0xff);
SendChar(y0&0xff);
SendChar((x1>>8)&0xff);
SendChar(x1&0xff);
SendChar((y1>>8)&0xff);
SendChar(y1&0xff);
SendEndCmd();
}
/*****
* Name: SetBackLight()
* Function: adjust back light
* Inlet Parameter: light_level
* Outlet Parameter: NA
*****/
void SetBackLight(uint8 light_level)
{
SendBeginCMD();
SendChar(0x60);
SendChar(light_level);
SendEndCmd();
}
/*****
* Name: SetCommBps()
* Function: set baud rate of communication
* Inlet Parameter: Baud
* Outlet Parameter: NA
*****/
void SetCommBps(uint32 Baud)
{
SendBeginCMD();
SendChar(0xA0);
SendChar((Baud>>16)&0xff);
SendChar((Baud>>8)&0xff);

```

```

SendChar(Baud&0xff);
SendEndCmd();
}

```

Descargas:

[Guía de Inicio Rápido TFT Touch](#)

[Conectividad de las pantallas TFT touch](#)

[Carga de archivos a la pantalla TFT de forma manual](#)

[Software Visual TFT](#)

[Manual en Ingles](#)

Nota: Antes de instalar el software consulte la Guía de inicio Rápido TFT Touch

	AG Electrónica S.A. de C.V. República del Salvador N° 20 Segundo Piso Teléfono: 5130 - 7210		
Acotación: NA	http://www.agelectronica.com/	Escala NA	Rev 1.BAAB Rev 2.
Tolerancia: NA	Descripción: Pantalla TFT Touch		
Tolerancia: NA 5U	Fecha: 02/09/2015	Número de parte: DST80048070-10	