TFT SPI Display

Driver development

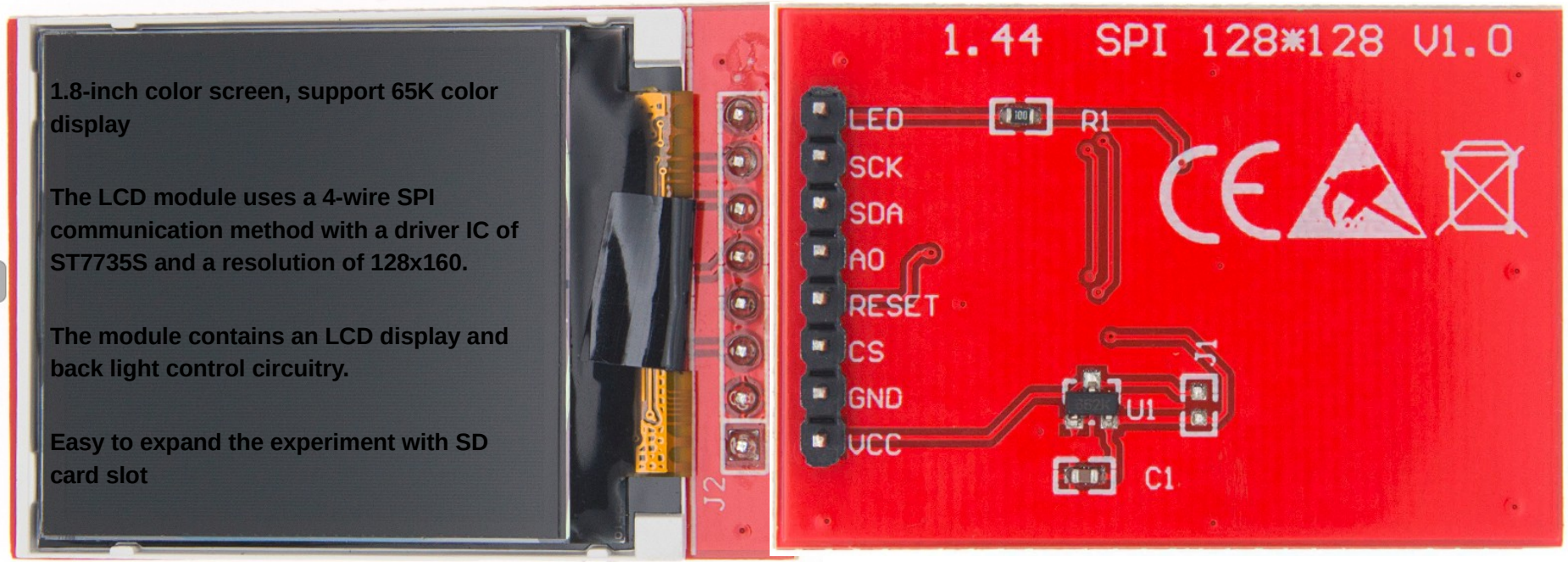
Product Description

1.8-inch color screen, support 65K color display

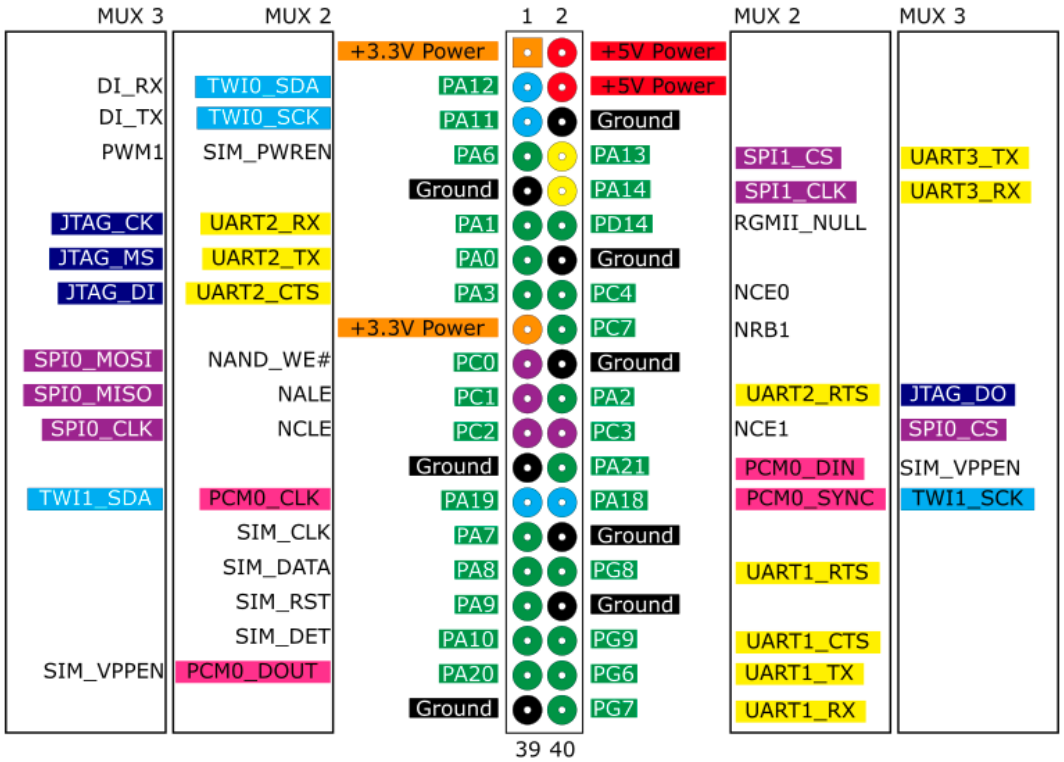
The LCD module uses a 4-wire SPI communication method with a driver IC of ST7735S and a resolution of 128x160.

The module contains an LCD display and back light control circuitry.

Easy to expand the experiment with SD card slot



Orange Pi (H3 SoC) GPIO - pinout



TFT SPI Pins

Number	Pin Label	Description
1	VCC	5V/3.3V power input
2	GND	Ground
3	CS	LCD chip select signal, low level enable
4	RESET	LCD reset signal, low level reset
5	A0	LCD register / data selection signal, high level: register, low level: data
6	SDA	SPI bus write data signal
7	SCK	SPI bus clock signal
8	LED	Backlight control, high level lighting, if not controlled, connect 3.3V always bright

Connection st7735s to Orange Pi One

LED -> PIN17 (+3.3V)

SCK -> PIN23 (SPI0 CLK)

SDA -> PIN19 (SPI0 MOSI)

A0 -> PIN18 (PC7)

RESET -> PIN22 (PA2)

CS -> PIN24 (SPI0 CS)

GND -> PIN6 (GND)

VCC -> PIN2 (+3.3V)

Step 0. Linux kernel sources

1.Build kernel image and modules

```
export CROSS_COMPILE=/bin/arm-linux-gnueabihf-
export ARCH=arm
export KERNELDIR=/work/kernel/linux/orange
make ARCH=arm help
make ARCH=arm O=orange clean
make ARCH=arm O=orange mrproper
make ARCH=arm O=orange sunxi_defconfig
make ARCH=arm O=orange zImage
make ARCH=arm O=${KERNELDIR} modules_install
```

U-boot DTS overlay

```
overlays=spi-add-cs1 spi-spidev
param_spidev_spi_bus=0
param_spidev_spi_cs=1
```

Prerequisites

1. gcc: arm-linux-gnueabi-
2. OPi Kernel compiled.
3. SDcard with OPi rootfs and kernel above.
4. ssh to your OPi
5. https://github.com/DevyatovAndrey/opi

Step 1. Basic I2C kernel module + DeviceTree

git clone https://github.com/DevyatovAndrey/sandbox.git && cd sandbox

git checkout v1.1_sceleton

edit your path in **envsetup.sh**

source ./envsetup.sh

cd ssd1306

./build_on_x86.sh --clean –module

./build_on_x86.sh –deploy

ssh root@orange

sudo dmesg -c

sudo insmod ssd1306.ko

sudo rmmod ssd1306

dmesg

Device Tree overlay

file: sun8i-h3-i2c0-lcd-ssd1306.dts

```
/dts-v1/;
/plugin/;
/ {
        compatible = "allwinner,sun8i-h3";
        fragment@0 {
                target = <&i2c0>;
                __overlay__ {
                        clock-frequency = <400000>;
                        #address-cells = <1>;
                        #size-cells = <0>;
                        lcd_ssd1306@3c {
                                compatible = "DAndy,lcd_ssd1306";
                                reg = <0x3c>;
                                status = "okay";
                        };
                };
        };
};
```

$KDIR/scripts/dtc/dtc -I dts -O dtb -@ sun8i-h3-i2c0-lcd-ssd1306.dts >sun8i-h3-i2c0-lcd-ssd1306.dtbo

Step 2. SYS FS

cd sandbox

git checkout v1.2_sysfs

cd ssd1306

./build_on_x86.sh --clean --module

./build_on_x86.sh --deploy

ssh orangepi

ls -l /sys/class/lcd_ssd1306/

Step 3. Communicate with the device

cd sandbox

git checkout v1.3_LCD_comm

cd ssd1306

./build_on_x86.sh --clean –module

./build_on_x86.sh –deploy

ssh orangepi

cat /sys/class/lcd_ssd1306/paint

cat /sys/class/lcd_ssd1306/clear

Step 4. Graphics primitives

cd sandbox

git checkout v1.4_graphics

cd ssd1306

./build_on_x86.sh --clean –module

./build_on_x86.sh –deploy

ssh orangepi

cat /sys/class/lcd_ssd1306/paint

cat /sys/class/lcd_ssd1306/clear

Step 5. Framebuffer support

The framebuffer device provides an abstraction for the graphics hardware. It represents the frame buffer of some video hardware and allows application software to access the graphics hardware through a well-defined interface, so the software doesn't need to know anything about the low-level (hardware register) stuff.

The device is accessed through special device nodes, usually located in the /dev directory, i.e. /dev/fb*

Step 5. Framebuffer support

cd sandbox

git checkout v1.5_framebuffer

cd ssd1306

./build_on_x86.sh --clean –module

./build_on_x86.sh –deploy

ssh orangepi

Step 6. Speedup!

git checkout v1.6_speedup

git checkout v1.7_wq

cd ssd1306

./build_on_x86.sh --clean –module

./build_on_x86.sh –deploy

ssh orangepi

Step 7. Userspace application

Low-Level Graphics on Linux

```
cd sandbox
git checkout v1.8_userapp
cd userapp_analog_clock
./build_user_app.sh
scp analog_clock orange

ssh orange
./analog_clock
```

Links

- https://randomnerdtutorials.com/guide-to-1-8-tft-display-with-arduino/
- http://www.lcdwiki.com/1.8inch_SPI_Module_ST7735S_SKU:MSP1803
- http://www.lcdwiki.com/res/MSP1803/QDTFT1801_specification_v1.1.pdf
- git@github.com:MaksymPrymierov/st7735s_driver.git
- https://github.com/DevyatovAndrey/opi

Happy coding!