



7" HDMI Display with Capacitive Touchscreen(Compatible with Raspberry Pi)

SKU:DFR0506

Introduction

This is a 7" HDMI display with capacitive touchscreen. It comes with a 7" LCD screen with capacitive touch panel overlay on it. The touch panel supports up to 5 touch points. And it adopts DFRobot USB free-driver technology, no special requirement of drivers. You can use it as easy as keyboard or mouse. It can be directly plugged to PC for touching control. In combination with HDMI HD screen, it can turn a large PC to a tablet immediately.

The screen supports Windows, Linux and MAC OS. It can be used on LattePanda, Raspberry Pi and other HDMI device. We also make some special hole to make it compatible with Arduino screw hole. You can mount them on the back of the display directly.

There is no need for external power supply for the screen, it can be powered by USB port and HDMI port. The display also supports back light adjustment. It is very convenient to adjust backlight by potentiometer on the back of the display.



Note: Please guarantee the external power supply current is above 1.5A@5V when it is driven by raspberry Pi. The lower current will reduce the touching sensitivity.

Features

- Capacitive Touchscreen, Support 5 Touch Points
- HDMI Interface
- USB Power Supply, Driver-free Installation

Specification

- Operating Voltage: 5V
- Operating Current: 160mA (with HDMI connection)
- Maximum Operating current: 1200mA
- Refresh Frequency: 43Hz (not recommended 60Hz, it will not be full screen)
- Screen Resolution: 1024x600
- Viewing Angle: 60°/70°/70°/70°
- Number of Touch Points: 5
- RGB888-16 Mega True Color
- Video Interface: HDMI
- Touch Interface: Micro-USB
- Outline Dimensions: 165mm x 100mm/6.50" x 3.94"
- Mounting Hole Size: M2.5
- Outside Mounting Size: 155mm x 90mm/6.10" x 3.54"
- Inside Mounting Size: 58mm x 49mm/2.83" x 1.93"
- Weight: to be continued...

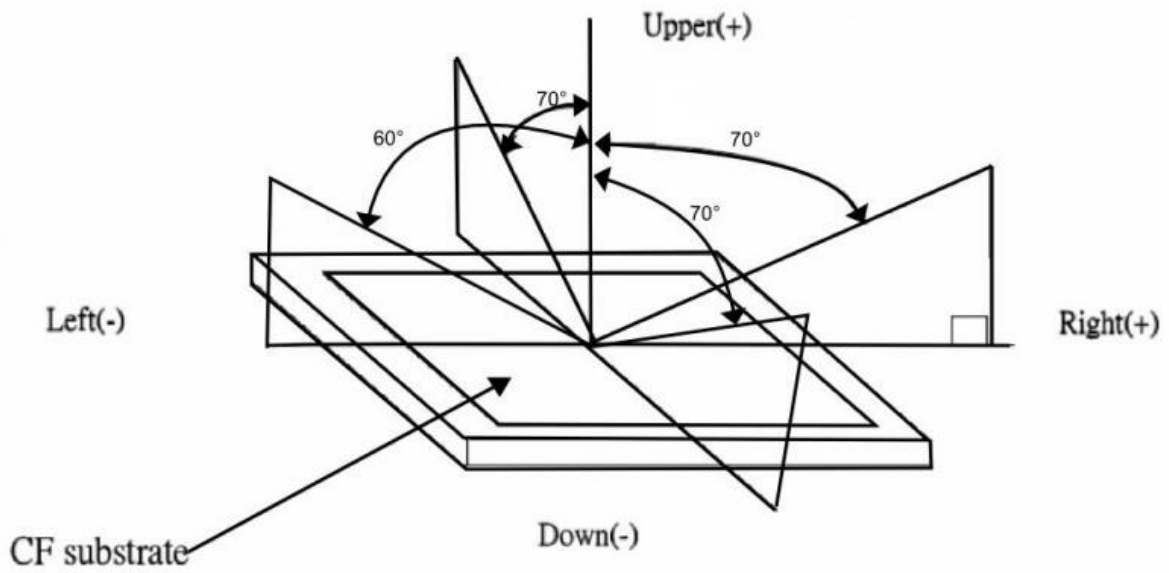
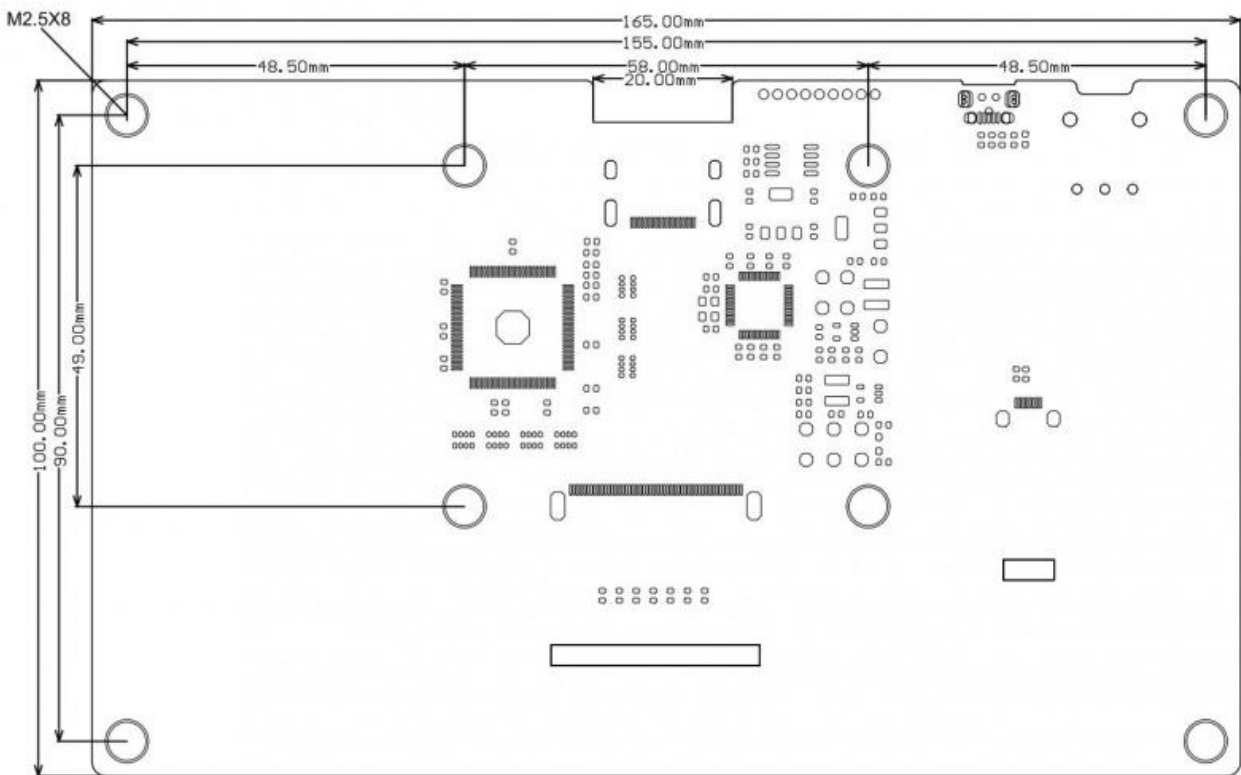
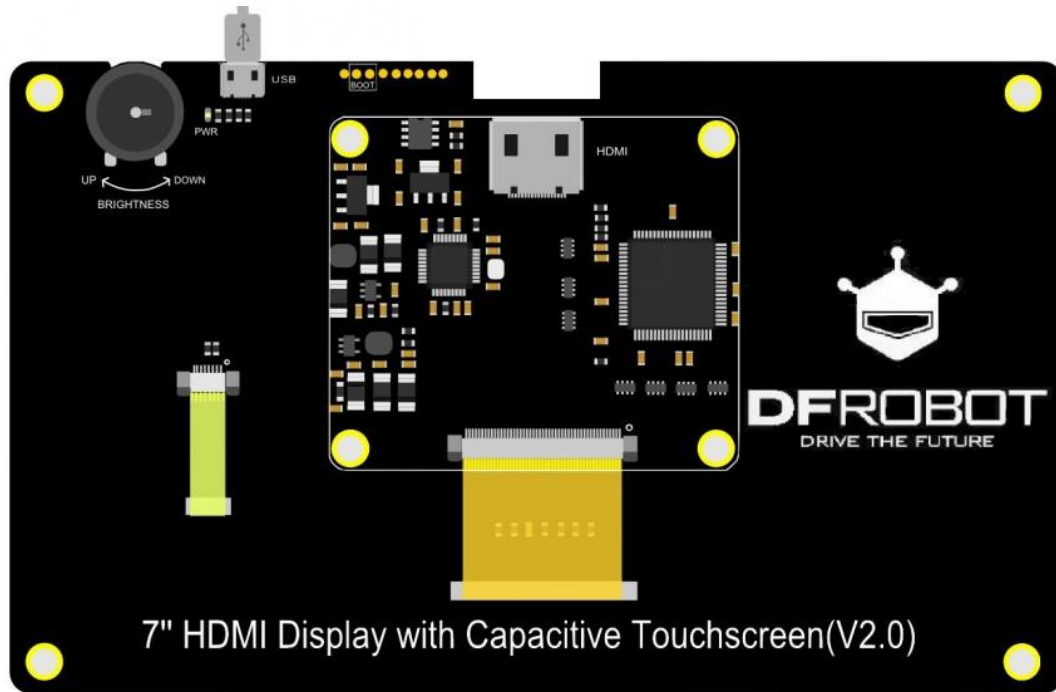


Fig.1 Definition of Viewing Angle



Screen Overview

- Standard HDMI
- Standard Micro USB



Tutorial

Touchscreen with Raspberry Pi

You need to reset resolution to avoid touch differentiation caused by resolution difference between Raspberry Pi and the Touchscreen.

- Enter `/boot/config.txt`, modify or add the project as below:

```
#### remove black borders
disable_overscan=1

#### set specific CVT mode
hdmi_cvt 1024 600 60 6 0 0 0

#### set CVT as default
hdmi_group=2
hdmi_mode=87
```

NOTE: Windows and MAC operating system can capture resolution automatically, so you don't need to modify any parameter.

Firmware Update

Dfu-util (Device Firmware Upgrade Utilities) is a free, open-source, cross-platform firmware update program, it is easy to update firmware for devices with microcontrollers without platform limitations. More information: <http://dfu-util.sourceforge.net>

In Windows

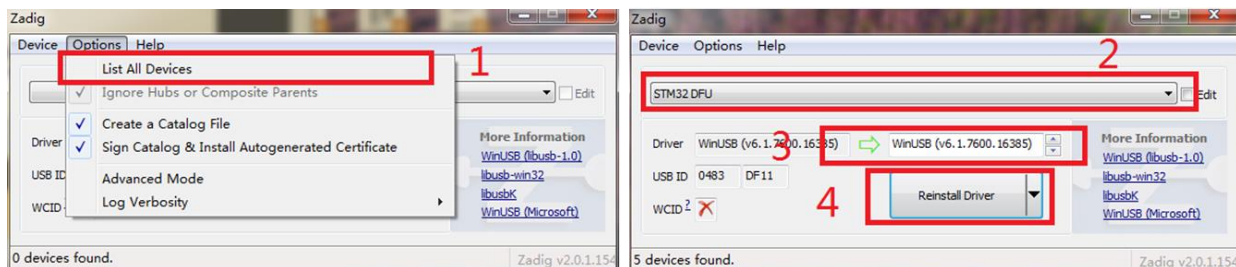
1. Download the USB installation tool zadig.exe

Address: http://zadig.akeo.ie/downloads/zadig_2.2.exe

2. Enter dfu mode Short-circuit the boot's two pins, and then connect it to the computer via USB



3. Install winUSB



Click Options, check All Devices(1)

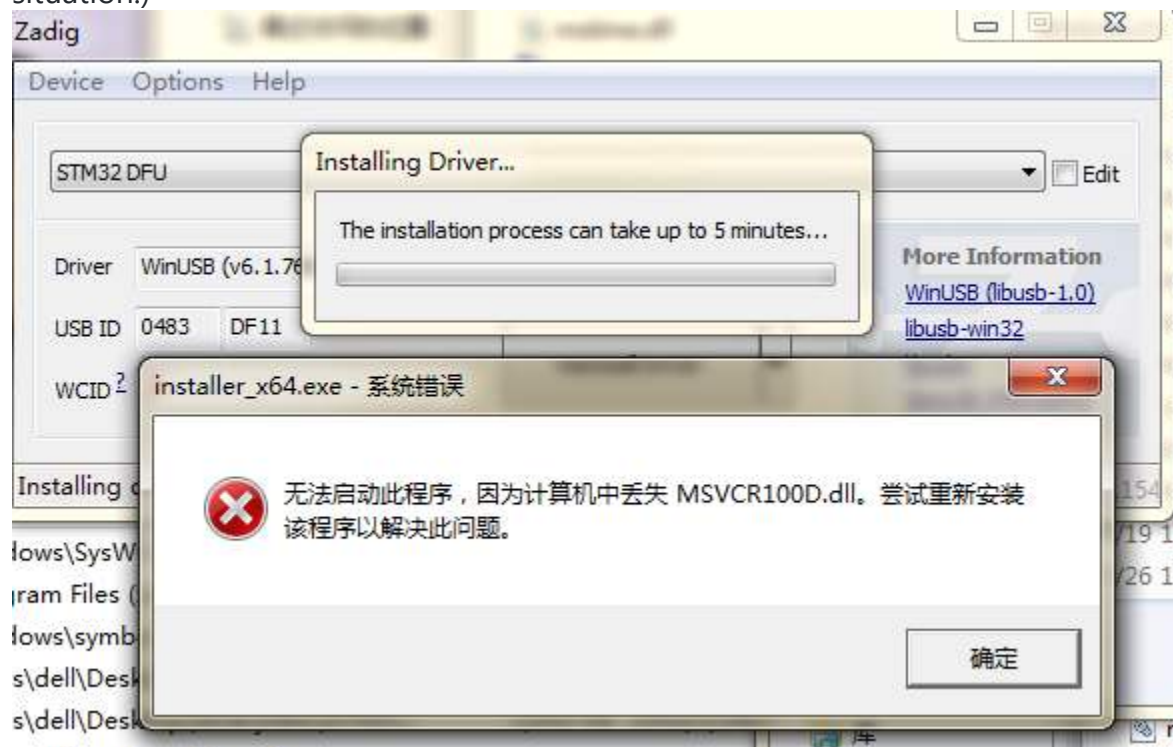
Select "STM32 DFU" in the drop-down box (2)

Select "WinUSB" (3)

Click Install Driver(4)

If there is a dialog box, click Continue to install.

When installing winUSB with zadig fails, the following interface will appear. (Win10 does not have the same situation.)



Win 7 (32-bit/64-bit) Download [msvcr100d.dll](#), place it in the C:\Windows\System32 directory, and then re-install the driver. There will be no mistakes.

Win 8 32-bit Download [msvcr100d.dll](#), place it in the C:\Windows\System32 directory, and then re-install the driver.

64-bit Download [msvcr100d.dll](#), place it in the same directory as zadig.exe. (There will be a new error in the C:\Windows\System32 directory)

| | | | |
|---------------|-----------------|--------|----------|
| zadig.exe | 2012/5/30 18:12 | 应用程序 | 6,203 KB |
| msvcr100d.dll | 2010/3/18 9:36 | 应用程序扩展 | 1,816 KB |

- 4. Download the dfu-util tool 32-bit [Click to download](#) 64-bit [Click to download](#)
- 5. Extract the file to your computer, such as "Desktop > dfu-util-0.9-win64", and use it without installation

- 6. Firmware update, [Click to download](#), unzip and copy the xxxx.dfu, erase.bin, and upload.py from the windows directory to the "desktop > dfu-util-0.9-win64".
- 7. Run upload.py (Windows need to install python firstly. [Click to download](#).)
- 8. The following interface appears, indicating that the program is being burned.

```

dfu-util 0.9

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2016 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

Opening DFU capable USB device...
ID 0483:df11
Run-time device DFU version 011a
Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 1024
DfuSe interface name: "Internal Flash  "
File contains 1 DFU images
parsing DFU image 1
image for alternate setting 0, <1 elements, total size = 14220>
parsing element 1, address = 0x08008000, size = 14212
Download      [=====] 100%          14212 bytes
Download done.
done parsing DfuSe file

```

- 9. After the burning is completed, replug the USB device.

In Linux

- 1. Enter dfu mode Short-circuit the boot's two pins, and then connect it to the computer via USB



- 2. Use apt-get for installation of dfu-util Apt-get install dfu-util
- 3. Use upload.py for firmware upgrade (requires root privileges)
- 4. Firmware update. [Click to download](#).

- **3.** Install dfu-util brew install dfu-util
- **4.** Use upload.py for firmware upgrade (requires root privileges)
- **5.** Firmware update. **Click to download** Open the terminal and enter into the newly downloaded firmware directory (DFRobot_RaspberryPi_TouchScreen_Firmware/linux) Use sudo python upload.py to run the program.
- **6.** Replug the USB device