

MYD-YT507H Ubuntu System Migration Guide



File Status: [] Draft [√] Release	FILE ID:	MYIR-MYD-YT507H-SW-DG-ZH-L4.9.170
	VERSION:	V1.0
	AUTHOR:	nico
	CREATED:	2022-09-07
	UPDATED:	2022-09-07

Revision History

VERSION	AUTHOR	PARTICIPANT	DATE	DESCRIPTION
V1.0	nico	nico,licy	2022-09-07	Initial Version

CONTENT

MYD-YT507H Ubuntu System Migration Guide.....	- 1 -
Revision History.....	- 2 -
CONTENT.....	- 3 -
1. Overview	- 5 -
1.1. Software Resources.....	- 5 -
1.2. Document Resources	- 6 -
2. Development Environment Preparation.....	- 7 -
2.1. Development Host Environment.....	- 7 -
3. Ubuntu18.04 File System Migration.....	- 8 -
3.1. Introduction.....	- 8 -
3.2. Get the Source Code	- 8 -
3.2.1. Obtain the Source Code Archive from the Disc Image	- 8 -
3.2.2. Get the Source Code Via Wget	- 9 -
3.3. Prepare the Chroot Environment.....	- 10 -
3.3.1. Install the Emulator.....	- 10 -
3.3.2. Make a Mount Script	- 10 -
3.4. Install the Package Files	- 11 -
3.4.1. Mount the System.....	- 11 -
3.4.2. Base Package Installation	- 12 -
3.4.3. Installation of Desktop Systems	- 13 -
3.4.4. Create a User	- 13 -
3.4.5. Other Configurations.....	- 15 -
3.4.6. Uninstall the System	- 15 -
3.5. Make a System.....	- 16 -
3.5.1. Make Ext4 Format Files	- 16 -
3.5.2. Generate an Image	- 16 -
3.5.3. Programming System	- 17 -
4. XFCE4	- 18 -

4.1. Introduction to XFCE - 18 -

4.2. Startup XFCE - 19 -

5. Resources - 21 -

Appendix A - 22 -

 Warranty & Technical Support Services - 22 -

1. Overview

There are many open source system construction frameworks on the Linux system platform, which are convenient for developers to build and customize embedded systems, and the more common ones are Buildroot, Yocto, OpenEmbedded and so on.

At the same time, more traditional desktop systems have also been added to the embedded environment system, such as Ubuntu, debian, etc., to provide more convenient and practical systems for embedded. This article mainly introduces the complete process of customizing a complete embedded Ubuntu system based on Ubuntu core and Myir core board, including the preparation of the development environment, the acquisition and porting of the ubuntu system, XFCE4 lightweight desktop management, etc. This document does not contain an introduction to the basics of the Linux BSP system and will use the BSP provided by Myir-image-full, which is published by Myir. For information on the production of BSP files, please refer to the MYD-YT507H _Linux Software Development Guide.

1.1. Software Resources

The Ubuntu system used by MYD-YT507H-V10 is based on the Ubuntu-base-18.04.5-base-arm64 .tar.gz version (download address below), adding a wealth of system resources and other software resources on the basis of this version.

<http://cdimage.ubuntu.com/ubuntu-base/releases/>

Table 1-1.Resource list

Feature items	Filename
ubuntu18.04 base	ubuntu-base-18.04.5-base-arm64.tar.gz
Start management	Xinit
Desktop system	xfce4
Network management	network-manager network-manager-gnome
Audio management	pulseaudio
Video player	vlc
Browser	chromium-browser

Note: The login name for the ubuntu system provided by Myir is: myir password: 123456

The root password is: 123456

1.2. Document Resources

slightly

2. Development Environment Preparation

Based on the Ubuntu base system in the development process of some of the software and hardware environment, including the necessary development host environment, necessary software tools, code and data acquisition, the specific preparation will be described in detail below.

2.1. Development Host Environment

This section describes how to build a development environment for the YT507H platform. By reading this chapter, you will learn about the installation and use of relevant hardware tools, software development and debugging tools. And can quickly set up the relevant development environment, in preparation for the later development and debugging. The YT507H processor is a 4-core ARM Cortex-A53, different processor cores will run different systems, will also use different development environments and tools, as described below:

- **Host Hardware**

The transplant of the system is not particularly high for the development host, and the configuration of the host does not make specific requirements. However, the host or VM running the Linux OS must be installed.

- **Host Operating System**

Generally choose to install Fedora, openSUSE, Debian, Ubuntu, RHEL or CentOS and other Linux distributions on the local host, here is recommended is the Ubuntu 20.04 64bit desktop version of the system, follow-up development is also to introduce this system as an example.

- **Installing Required Software Packages**

For the basic package, please refer to the MYD-YT507H_Linux Software Development Guide for installation, where only one package needs to be installed.

```
dth@TH:~$ sudo apt-get update
dth@TH:~$ sudo apt-get install qemu-user-static
```

3. Ubuntu18.04 File System Migration

3.1. Introduction

ubuntu-base is the Ubuntu official build of the Ubuntu smallest file system, including the debain package manager, the base package size is usually only a few tens of megabytes, behind it there is the entire ubuntu software source support, ubuntu software general stability is better, based on ubuntu-base on-demand installation of Linux software, deep customizable, etc., often used for embedded rootfs construction.

There are several common file system building methods embedded: busybox, yocto, buildroot, but Ubuntu is convenient, a powerful package management system, has strong community support, and can directly apt-get install to install new packages. This article describes how to build a complete Ubuntu system based on Ubuntu-base. Ubuntu supports many architectures, arm, X86, powerpc, ppc, etc., this article is mainly based on arm as an example, to build a more complete ubuntu system.

3.2. Get the Source Code

We provide two ways to obtain the source code, one is to obtain the compressed package directly from the Mill disc image 04-sources directory, and the other is to use the wget to obtain the official source code for construction, please choose one of them according to the actual needs to build.

3.2.1. Obtain the Source Code Archive from the Disc Image

The compressed source package is located in the Myir Development Package data 04-Sources/ubuntu-base-18.04.5-base-arm64 .tar.gz. Copy the package to a user-specified directory, such as the /home/dth/ubuntu directory, which will serve as the top-level directory for subsequent builds:

Subsequent operations are best performed in **root** privileges

```
dth@TH:~$ mkdir ubuntu/ubuntu-rootfs -p
dth@TH:~$ su
root@TH:/home/dth# cd ubuntu
```


Extract the ubuntu-base-18.04.5-base-arm64 .tar.gz package to the ubuntu-rootfs directory

```
root@TH:/home/dth/ubuntu# tar -xvf ubuntu-base-18.04.5-base-arm64.tar.gz -C .  
/ubuntu-rootfs
```

The extracted directory reads as follows:

```
PC$ tree -d -L 1 ubuntu-rootfs
```

```
ubuntu  
├── bin  
├── boot  
├── dev  
├── etc  
├── home  
├── lib  
├── media  
├── mnt  
├── opt  
├── proc  
├── root  
├── run  
├── sbin  
├── srv  
├── sys  
├── tmp  
├── usr  
└── var
```

```
18 Directories
```

3.2.2. Get the Source Code Via Wget

Specific operation methods are as follows:

```
dth@TH:~$ sudo wget http://cdimage.ubuntu.com/ubuntu-base/releases/18.04/release/ubuntu-base-18.04.5-base-arm64.tar.gz
```

The extracted directory is the same as above.

3.3. Prepare the Chroot Environment

3.3.1. Install the Emulator

```
root@TH:/home/dth/ubuntu# cp /usr/bin/qemu-aarch64-static ./ubuntu-rootfs/usr/bin/
```

Copy the host DNS configuration file into the arm-architecture Ubuntu file system (must be copied, otherwise the following operation may not work).

```
root@TH:/home/dth/ubuntu# cp /etc/resolv.conf ./ubuntu-rootfs/etc/resolv.conf
```

3.3.2. Make a Mount Script

Copy the following script code into ch-mount.sh file and change the permissions to executable.

```
root@TH:/home/dth/ubuntu# vi ch-mount.sh
root@TH:/home/dth/ubuntu# chmod 777 ch-mount.sh
```

```
#!/bin/bash
function mnt() {
    echo "MOUNTING"
    sudo mount -t proc /proc ${2}proc
    sudo mount -t sysfs /sys ${2}sys
    sudo mount -o bind /dev ${2}dev
    sudo mount -o bind /dev/pts ${2}dev/pts
    sudo chroot ${2}
}
function umnt(){
    echo "UNMOUNTING"
    sudo umount ${2}proc
    sudo umount ${2}sys
    sudo umount ${2}dev/pts
```

```

    sudo umount ${2}dev
}
if [ "$1" == "-m" ] && [ -n "$2" ];
then
    mnt $1 $2
elif [ "$1" == "-u" ] && [ -n "$2" ];
then
    umnt $1 $2
else
    echo ""
    echo "Either 1'st, 2'nd or both parameters were missing"
    echo ""
    echo "1'st parameter can be one of these: -m(mount) OR -u(umount)"
    echo "2'nd parameter is the full path of rootfs directory (with trailing '/)"
    echo ""
    echo "For example: ch-mount -m/media/sdcard/"
    echo ""
    echo 1st parameter : ${1}
    echo 2nd parameter : ${2}
fi

```

3.4. Install the Package Files

3.4.1. Mount the System

Start by mounting the ubuntu file system using ch-mount.sh.

```

root@TH:/home/dth/ubuntu# ./ch-mount.sh -m ubuntu-rootfs/
MOUNTING
root@TH:/#
root@TH:/# ls
bin  dev  home  media  opt  root  sbin  sys  usr
boot  etc  lib  mnt  proc  run  srv  tmp  var

```

After mounting successfully, you can configure the ubuntu file system and install some necessary software.

3.4.2. Base Package Installation

You can install the following package files as needed, and it is recommended to install them all. (Follow the installation sequence to avoid errors during the installation)

```
root@TH:/# apt update
root@TH:/# apt-get install language-pack-zh-hant language-pack-zh-hans
root@TH:/# apt install language-pack-en-base
root@TH:/# apt install dialog rsyslog
root@TH:/# apt install systemd avahi-daemon avahi-utils udhcpc ssh (Must be in
stalled)
root@TH:/# apt install sudo
root@TH:/# apt install vim
root@TH:/# apt install net-tools
root@TH:/# apt install ethtool
root@TH:/# apt install ifupdown
root@TH:/# apt install iputils-ping
root@TH:/# apt install htop
root@TH:/# apt install lrzsz
root@TH:/# apt install gpod
root@TH:/# apt install wpasupplicant
root@TH:/# apt install kmod
root@TH:/# apt install iw
root@TH:/# apt install usbutils
```

Add log, the user debugs the ubuntu system

```
root@TH:/# touch /var/log/rsyslog
root@TH:/# chown syslog:adm /var/log/rsyslog
root@TH:/# chmod 666 /var/log/rsyslog
root@TH:/# systemctl unmask rsyslog
root@TH:/# systemctl enable rsyslog
```

Install network and language pack support

```
root@TH:/# apt-get install synaptic
```

```
root@TH:/# apt-get install network-manager network-manager-gnome
root@TH:/# apt-get install rfc
root@TH:/# apt install -y --force-yes --no-install-recommends fonts-wqy-microhei
root@TH:/# apt install -y --force-yes --no-install-recommends ttf-wqy-zenhei
```

Install Bluetooth-related software

```
root@TH:/# apt-get install bluetooth
root@TH:/# apt-get install bluez
root@TH:/# apt-get install blueman
root@TH:/# apt-get install cheese
```

3.4.3. Installation of Desktop Systems

Xfce4 desktop system installation

```
root@TH:/# apt-get install xinit
root@TH:/# apt-get install xfce4
```

Browser chrome installation

```
root@TH:/# apt-get install chromium-browser chromium-browser-l10n
```

3.4.4. Create a User

Set root password: 123456

```
root@TH:/# passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Be sure to execute the following instructions, Otherwise, sudo will report an error "sudo: /usr/bin/sudo must be owned by uid 0 and have the setuid bit set"

```
root@TH:/# chown root:root /usr/bin/sudo
root@TH:/# chmod 4755 /usr/bin/sudo
```

Create a username with: myir Password: 123456

```
root@TH:/# adduser myir
perl: warning: Setting locale failed.
```

perl: warning: Please check that your locale settings:

```
LANGUAGE = (unset),
LC_ALL = (unset),
LC_TIME = "zh_CN.UTF-8",
LC_IDENTIFICATION = "zh_CN.UTF-8",
LC_TELEPHONE = "zh_CN.UTF-8",
LC_NUMERIC = "zh_CN.UTF-8",
LC_ADDRESS = "zh_CN.UTF-8",
LC_NAME = "zh_CN.UTF-8",
LC_MONETARY = "zh_CN.UTF-8",
LC_PAPER = "zh_CN.UTF-8",
LC_MEASUREMENT = "zh_CN.UTF-8",
LANG = "zh_CN.UTF-8"
```

are supported and installed on your system.

perl: warning: Falling back to the standard locale ("C").

Adding user `myir' ...

Adding new group `myir' (1000) ...

Adding new user `myir' (1000) with group `myir' ...

Creating home directory `/home/myir' ...

Copying files from `/etc/skel' ...

Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

Changing the user information for myir

Enter the new value, or press ENTER for the default

Full Name []: cy

Room Number []: 604

Work Phone []:

Home Phone []:

Other []:

Is the information correct? [Y/n] y

Set permissions

vi /etc/sudoers

```
root ALL=(ALL:ALL) ALL
myir(Add based on the users you add) ALL=(ALL:ALL) ALL
```

When you add a user above, the warning that appears in the middle can use the following command:

```
root@TH:/# export LC_ALL=C
```

3.4.5. Other Configurations

Set the hosts and host name

```
root@TH:/# vi /etc/hosts
```

Add 127.0.0.1 myir

```
root@TH:/# vi /etc/hostname
```

Clear the content

Add myir (based on the username actually added)

Configure the NIC interface and add the following:

```
root@TH:/# vi /etc/network/interfaces
auto eth0
iface eth0 inet dhcp
```

Create a link file init (must be configured), and then modify the permissions to 777

```
root@TH:/# ln -s lib/systemd/systemd init
root@TH:/# chmod 777 init
```

Configure the serial port

```
root@TH:/#systemctl enable getty@ttyS0.service
```

3.4.6. Uninstall the System

After the above steps are completed, you can uninstall the system. Exit the system by entering exit directly into the system and use the command to uninstall

```
root@TH:/# exit
exit
```

```
root@TH:/home/dth/ubuntu#
root@TH:/home/dth/ubuntu# ./ch-mount.sh -u ubuntu-rootfs/
UNMOUNTING
```

At this point, the ubuntu file system has been configured.

Note: Myir also provides a well-made file system that can help users avoid the above steps, source code path: 04_Sourceubuntu18_rootfs.tar.gz

3.5. Make a System

3.5.1. Make Ext4 Format Files

Ext4 format file, make a larger than 6GByte EXT4 empty file, due to the installation of more software, the file system will be very large, here is temporarily set to about 6GByte size, users can change according to the situation (**The file system name must be rootfs.ext4**)

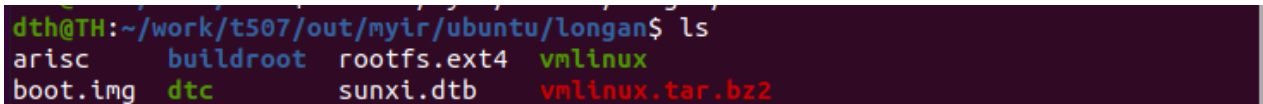
```
root@TH:/home/dth/ubuntu# make_ext4fs -s -l 6144M rootfs.ext4 ubuntu-rootfs/
```

The file system is now finished

Note: Myir has already made a system that can be programmed, system path: 02_imagesmyir_linux_ubuntu_uart0.img

3.5.2. Generate an Image

The YT507H platform will use the files in Figure 3-1 when building the ubuntu image, just need to replace the rootfs.ext4 in this directory with the rootfs.ext4 we made, then the built image will use our own file system. For details and instructions, please refer to the MYD-YT507H_Linux Software Development Guide.



```
dth@TH:~/work/t507/out/myir/ubuntu/longan$ ls
arisc      buildroot  rootfs.ext4  vmlinux
boot.img   dtc        sunxi.dtb   vmlinux.tar.bz2
```

Figure 3-1. rootfs.ext4 File Directory

It then returns to the top-level directory of the build

```
dth@TH:~$ cd /home/dth/work/t507
dth@TH:~/work/t507$ ./build.sh pack
```


The image is created as shown below

```
aultls32.fex Len: 0x2511f
cardtool.fex Len: 0x11e00
cardscript.fex Len: 0x76b
sunxi_gpt.fex Len: 0x2000
sunxi_mbr.fex Len: 0x10000
dlinfo.fex Len: 0x4000
arisc.fex Len: 0x6
vmlinux.fex Len: 0x58964b9
boot-resource.fex Len: 0x753400
Vboot-resource.fex Len: 0x4
env.fex Len: 0x20000
Venv.fex Len: 0x4
boot.fex Len: 0x179f800
Vboot.fex Len: 0x4
rootfs.fex Len: 0x2a982d74
Vrootfs.fex Len: 0x4
BuildImg 0
Dragon execute image.cfg SUCCESS !
-----image is at-----
size:4.8G /home/dth/work/t507/out/myir_linux_ubuntu_uart0.img
pack finish
```

Figure 3-2. Build Success

The finished image is in the /work/t507/out directory

3.5.3. Programming System

For the steps to burn, please refer to the MYD-YT507H_Linux Software Development Guide.

4. XFCE4

4.1. Introduction to XFCE

Xfce is a lightweight desktop environment for UNIX-like operating systems. It aims to be fast and low in system resources while still having a beautiful interface and user-friendly operation. Xfce embodies the traditional UNIX concept of modularity and reusability. It consists of many components that provide all the functionality that a modern desktop environment can expect. They are individually packaged and can be selected from the available packages to configure the best personal environment.

Another priority for Xfce is to comply with standards, especially Freedesktop.org. Xfce can be installed on a variety of UNIX platforms. It is known to compile in Linux, NetBSD, FreeBSD, OpenBSD, Solaris, Cygwin, and many more.

Xfce contains several core components that can be used in a desktop environment to manage overall system resources. The main components are as follows:

- Window Manager: Manages the position of the window on the screen, provides a window view, and manages workspaces and virtual desktops.
- Desktop Manager: Sets the background image and provides window menus, desktop icons or a list of minimized icons and windows, and so on.
- Work Panel: Switch between open windows, launch applications, switch workspaces.
- Session Manager: Controls login and power management for desktops and allows multiple logon sessions to be stored.
- App Finder: Displays applications installed on your system in categories and can quickly find and launch applications.

- Archive Manager: Provides basic file management features and unique utilities such as a large-capacity renamer.
- Settings Manager: Tools for controlling various settings on the desktop, such as keyboard shortcuts, appearance, display settings, and more.

In addition to the basic modules, Xfce offers a number of additional apps and plugins that can extend your desktop in the way you like, such as a terminal emulator, text editor, sound mixer, app finder, image viewer, iCal-based calendar, and CD and DVD burning apps.

4.2. Startup XFCE

The xinit command is the initializer of the X-Window system under Linux, which mainly completes the initialization of the X server. When the startx command is executed, the xinit can be started.

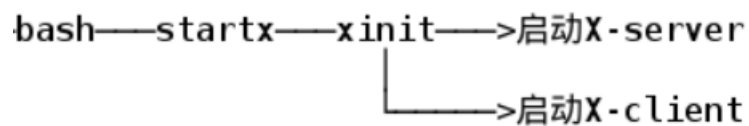


Figure 4-1.xinit Start the process

The system provided by Myir adds the self-start service `/lib/systemd/system/watchdog.service`

This service will start XFCE.

Default Wallpaper: `/usr/share/backgrounds/xfce/xfce-blue.jpg`

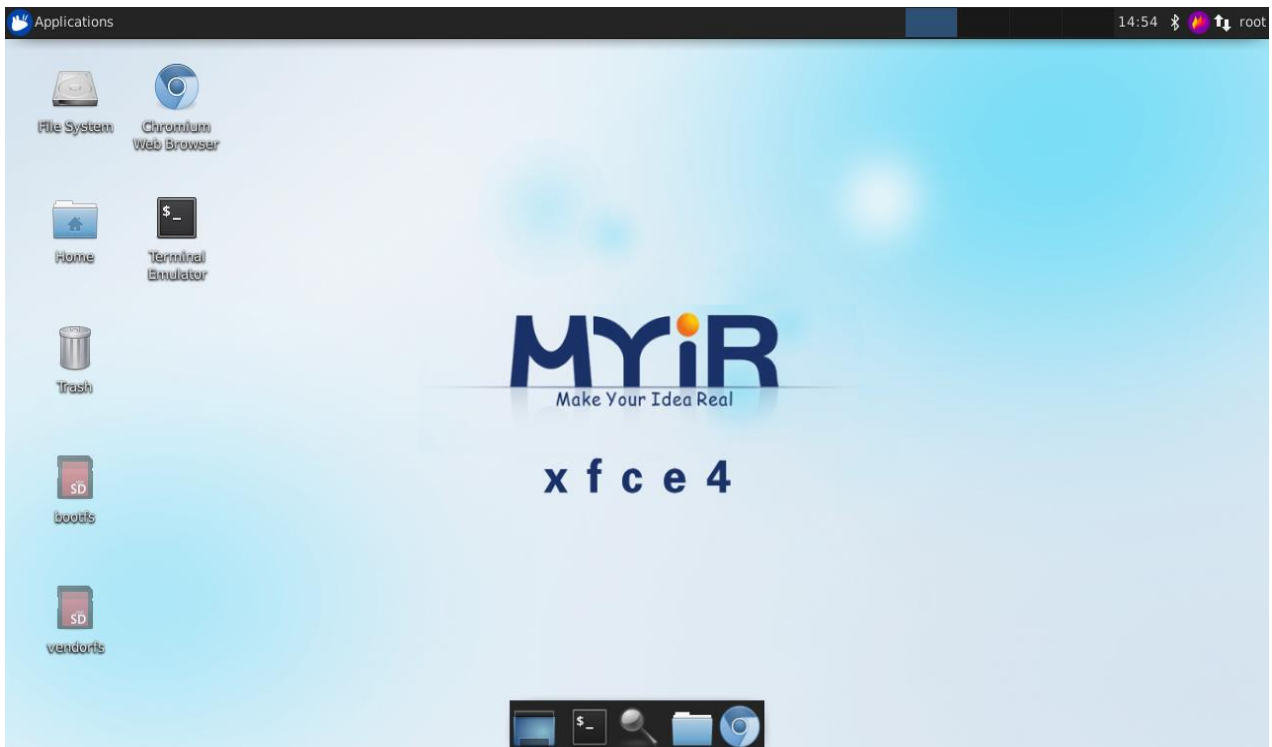


Figure 4-2.XFCE Launch the interface

5. Resources

Linux kernel Open source community

<https://www.kernel.org/>

ubuntu

<http://cdimage.ubuntu.com/ubuntu-base/releases/18.04.5/release/>

xfce4

<https://www.xfce.org/>

freedesktop

<https://www.freedesktop.org/wiki/>

Appendix A

Warranty & Technical Support Services

MYIR Electronics Limited is a global provider of ARM hardware and software tools, design solutions for embedded applications. We support our customers in a wide range of services to accelerate your time to market.

MYIR is an ARM Connected Community Member and work closely with ARM and many semiconductor vendors. We sell products ranging from board level products such as development boards, single board computers and CPU modules to help with your evaluation, prototype, and system integration or creating your own applications. Our products are used widely in industrial control, medical devices, consumer electronic, telecommunication systems, Human Machine Interface (HMI) and more other embedded applications. MYIR has an experienced team and provides custom design services based on ARM processors to help customers make your idea a reality.

The contents below introduce to customers the warranty and technical support services provided by MYIR as well as the matters needing attention in using MYIR' s products.

Service Guarantee

MYIR regards the product quality as the life of an enterprise. We strictly check and control the core board design, the procurement of components, production control, product testing, packaging, shipping and other aspects and strive to provide products with best quality to customers. We believe that only quality products and excellent services can ensure the long-term cooperation and mutual benefit.

Price

MYIR insists on providing customers with the most valuable products. We do not pursue excess profits which we think only for short-time cooperation. Instead, we hope to establish long-term cooperation and win-win business with customers. So we will offer reasonable prices in the hope of making the business greater with the customers together hand in hand.

Delivery Time

MYIR will always keep a certain stock for its regular products. If your order quantity is less than the amount of inventory, the delivery time would be within three days; if your order quantity is greater than the number of inventory, the delivery time would be always four to six weeks. If for any urgent delivery, we can negotiate with customer and try to supply the goods in advance.

Technical Support

MYIR has a professional technical support team. Customer can contact us by email (support@myirtech.com), we will try to reply you within 48 hours. For mass production and customized products, we will specify person to follow the case and ensure the smooth production.

After-sale Service

MYIR offers one year free technical support and after-sales maintenance service from the purchase date. The service covers:

Technical support service

MYIR offers technical support for the hardware and software materials which have provided to customers;

- To help customers compile and run the source code we offer;
- To help customers solve problems occurred during operations if users follow the user manual documents;
- To judge whether the failure exists;
- To provide free software upgrading service.

However, the following situations are not included in the scope of our free technical support service:

- Hardware or software problems occurred during customers' own development;
- Problems occurred when customers compile or run the OS which is tailored by themselves;
- Problems occurred during customers' own applications development;
- Problems occurred during the modification of MYIR's software source code.

After-sales maintenance service

The products except LCD, which are not used properly, will take the twelve months free maintenance service since the purchase date. But following situations are not included in the scope of our free maintenance service:

- The warranty period is expired;
- The customer cannot provide proof-of-purchase or the product has no serial number;
- The customer has not followed the instruction of the manual which has caused the damage the product;
- Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads the defects of appearance/function;
- Due to the power supply, bump, leaking of the roof, pets, moist, impurities into the boards, all those reasons which have caused the damage of the products or defects of appearance;
- Due to unauthorized weld or dismantle parts or repair the products which has caused the damage of the products or defects of appearance;
- Due to unauthorized installation of the software, system or incorrect configuration or computer virus which has caused the damage of products.

Warm tips

1. MYIR does not supply maintenance service to LCD. We suggest the customer first check the LCD when receiving the goods. In case the LCD cannot run or no display, customer should

contact MYIR within 7 business days from the moment get the goods. 2. Please do not use finger nails or hard sharp object to touch the surface of the LCD.

3. MYIR suggests user purchasing a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.

4. Do not clean the surface of the screen with chemicals.

5. Please read through the product user manual before you using MYIR' s products.

6. For any maintenance service, customers should communicate with MYIR to confirm the issue first. MYIR' s support team will judge the failure to see if the goods need to be returned for repair service, we will issue you RMA number for return maintenance service after confirmation.

Maintenance period and charges

- MYIR will test the products within three days after receipt of the returned goods and inform customer the testing result. Then we will arrange shipment within one week for the repaired goods to the customer. For any special failure, we will negotiate with customers to confirm the maintenance period.
- For products within warranty period and caused by quality problem, MYIR offers free maintenance service; for products within warranty period but out of free maintenance service scope, MYIR provides maintenance service but shall charge some basic material cost; for products out of warranty period, MYIR provides maintenance service but shall charge some basic material cost and handling fee.

Shipping cost

During the warranty period, the shipping cost which delivered to MYIR should be responsible by user; MYIR will pay for the return shipping cost to users when the product is repaired. If the warranty period is expired, all the shipping cost will be responsible by users.

Products Life Cycle

MYIR will always select mainstream chips for our design, thus to ensure at least ten years continuous supply; if meeting some main chip stopping production, we will inform customers in time and assist customers with products updating and upgrading.

Value-added Services

1. MYIR provides services of driver development base on MYIR' s products, like serial port, USB, Ethernet, LCD, etc.
2. MYIR provides the services of OS porting, BSP drivers' development, API software development, etc.
3. MYIR provides other products supporting services like power adapter, LCD panel, etc.
4. ODM/OEM services.

MYIR Electronics Limited

Room 04, 6th Floor, Building No.2, Fada Road,
Yunli Intelligent Park, Bantian, Longgang District.

Support Email: support@myirtech.com

Sales Email: sales@myirtech.com

Phone: +86-755-22984836

Fax: +86-755-25532724

Website: www.myirtech.com