ALX-9x91
User Manual

V1.0.1 Dec. 2025

ALX-905/ALX-9251/ALX-9651

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, not for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Contact US

If you have any problem, please feel free to contact us.

You can count on us for quick response.

Email: service@icpdas.com

Contents

1. Introduction

This chapter provides an overview of the ALX-9x51 and its components, introducing basic concepts to users familiar with the ALX-9x51.

The ALX-9x51 are Linux-based PACs that combine computing, I/O, and operator interface into a single unit, and provide the perfect solution for data acquisition and control in an individual PAC.
It is equipped with an Intel Core i5-8365UE CPU, 0/2/6 I/O expansion slots and a variety of connectives including dual Gigabit Ethernet, VGA, HDMI, USB port 2.0/3.0, RS-232 and RS-485 interface. Local I/O slots are available to use our I-9K, I-97K and e-9K series I/O modules and remote I/O expansions are available to use our Ethernet I/O modules and RS-485 I/O modules.

1.1. Features

The ALX-9x51 offers the most comprehensive configuration and remote system upgrade solutions to meet specific application requirements. The following list shows the hardware and software features designed to simplify installation, configuration and application.

Powerful CPU Module

1. i5-8365UE (1.6 ~ 4.1 GHz, 4C8T) for ALX-9x51

Built-in VGA and HDMI Port

A built-in VGA and HDMI port can be directly connected to a regular display.

Memory Size:

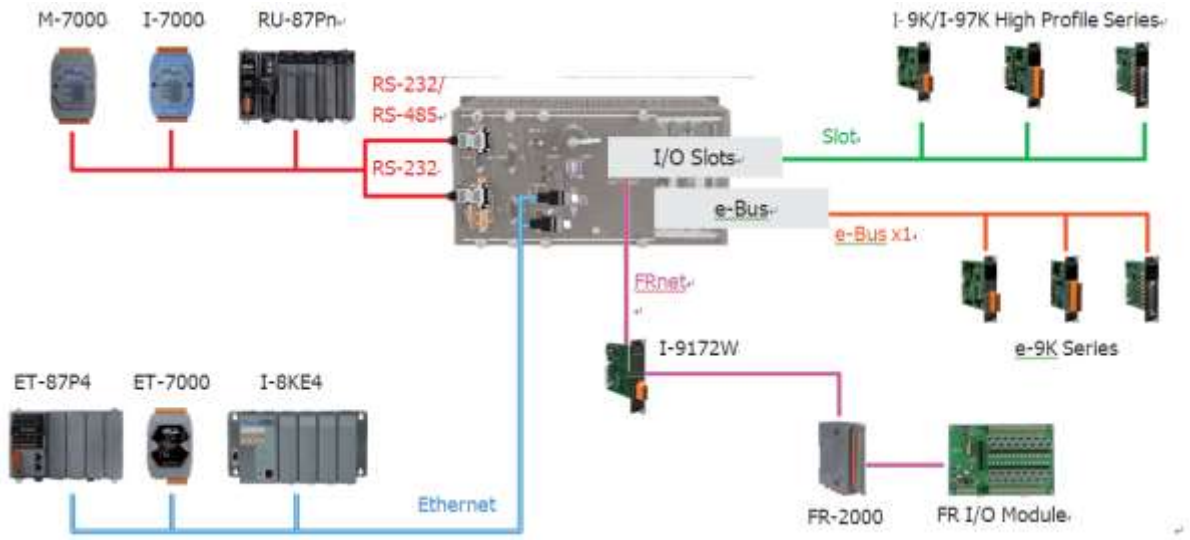
- SDRAM (16 GB DDR4) for ALX-9051/ALX-9251/ALX-9651
- Flash (128 GB SSD)
- The device features a built-in CFast slot for an optional CFast card.

64-bit Hardware Serial Number

The 64-bit hardware serial number is unique and individual. Every serial number of ALX-9x51 PAC is different. Users can add a checking mechanism to their AP to prevent software from pirating.

Rich I/O Expansion Ability(connect with serial/USB I/O and Ethernet I/O devices)

- I/O Slots
- RS-232/RS-485
- FRnet
- USB



Besides the local I/O slots, ALX-9x51 also provides several RS-232/RS-485/USB ports and two Ethernet ports to connect with serial/USB I/O and Ethernet I/O devices.

Dual Ethernet Ports

ALX-9x51 provides two Gigabit Ethernet ports. The two Ethernet ports can be used to implement redundant Ethernet communication and separate Ethernet communication (one for a global Internet, one for private Ethernet).

Dual Watchdog Timer

A system could be hanged up when the OS or the AP fails. There are two watchdogs (CPU watchdog and Backplane watchdog) designed to automatically reset the CPU/Backplane when the situations happen. The design will increase the reliability of the system.

Redundant Power Inputs

To prevent the ALX-9x51 from failing by the power loss, the power module is designed with two inputs. The ALX-9x51 can keep working even one power input fails, and meanwhile, there is a relay output for informing the power failure.

Operating Temperature：

-25°C to +60°C

Rugged RJ-45 connector

LAN1

The ALX-9000 is equipped with a rugged and dustproof RJ-45 connector on the LAN1 port that can protect against both liquid and dust contaminants in harsh industrial environments, as well as protection against vibration and shock.

LAN2

Screw-lockable RJ-45 connector on the LAN2 port and there are screw holes (spacing 20mm) on both sides of the RJ45 connector, you can lock the RJ45 cable connector with them, to reduce the risk of the Ethernet cable falling off due to vibration occurring. Meanwhile, you can also choose a general RJ45 cable connector for your needs

▲Screw-lockable RJ45 Connector

▲Regular Ethernet Cable

▲Ethernet Cable with Screw Lock

Power/Communication spring clamp terminal connector

The spring clamp terminal block that is used for the Power/Communication connector in the ALX-9x51 offers advantages including anti-vibration, stable clamping, and easier installation.

Metal Enclosure

The ALX-9x51 features a durable metal casing to provide high levels of impact resistance and flame resistance and higher protection on EMS or other electromagnetic noises.

Fan less Design for ALX-9x91 series

With the fan less design for embedded applications, the operating temperature range is -25°C to +60°C for the ALX-9x91 series.

Advanced CPU Heat Sink Design for AXP-9x5x-IoT series

With the new heat sink design using a larger heat sink and fan, the specific metal mechanism design can reduce the temperature of the entire CPU by nearly 10°C; moreover, it can also extend the service life of electronic parts. We specially selected the new long-life type of cooling fan, and the fan can work for 180,000 hours by test. (180,000 hrs ≒ 20 years)., the operating temperature range is -25°C to +60°C for the ALX-9x51 series.

▲Long Life CPU Heat Sink Design

▲Long Life Design

▲Regular Design

1.2. Specifications

The table below summarizes the specifications of ALX-9x51 series.

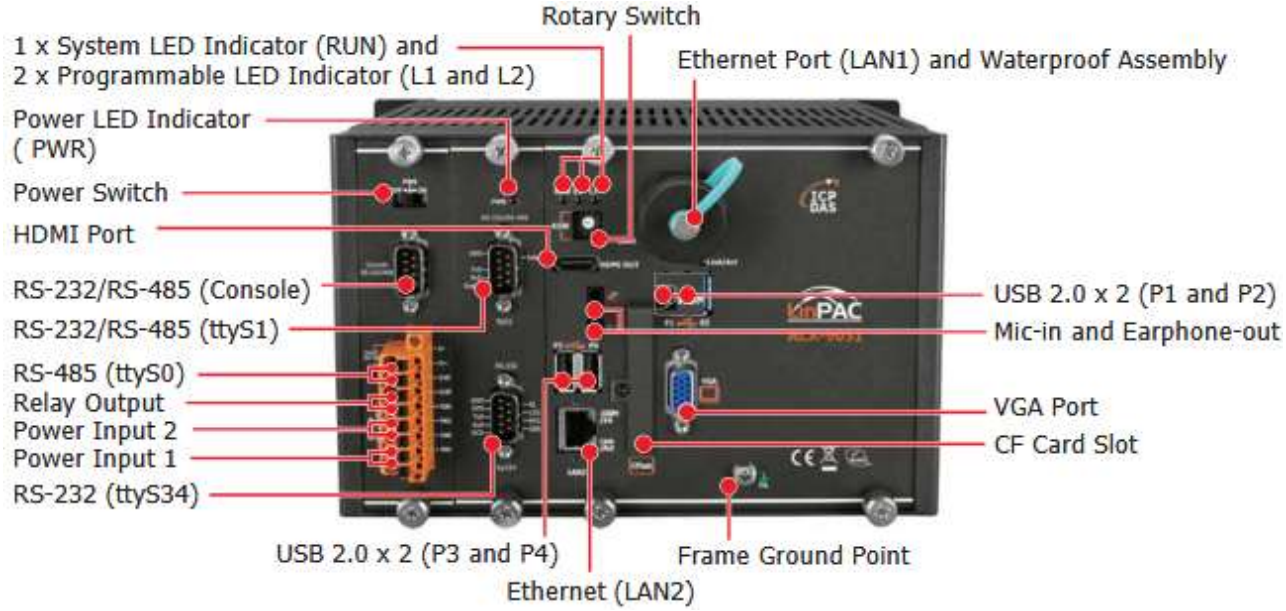
Models	ALX-9051	ALX-9251	ALX-9651
System Software			
OS	Ubuntu 24.04 with kernel 6.9.0-rt5		
Embedded Service	SFTP server, Web server, SSH		
SDK Provided	Standard LinPAC SDK for Linux by GNU C language		
Multilanguage Support	English, Russian, Korean, Japanese, Simplified Chinese, Traditional Chinese		
CPU Module			
CPU	Intel® Core™ i5-8365UE Processor (1.6 ~ 4.1 GHz, 4C8T)		
SDRAM	16 GB DDR4 SDRAM		
MRAM	128 KB		
Flash(SSD)	mSATA slot with one 128 GB SSD		
EEPROM	16 KB		
Memory Expansion	Built-in CFast slot for optional CFast card (up to 64 GB)		
RTC (Real Time Clock)	Provide second, minute, hour, date, day of week, month, year		
64-bit Hardware Serial Number	Yes, for software copy protection		
Dual Watchdog Timers	Yes		
Programmable LED Indicator	3 (RUN, L1, L2)		
Rotary Switch	Yes (0 ~ 9)		
VGA, HDMI & Communication Ports			
Signal	VGA, HDMI		
Resolution	**VGA** 1920 x 1200 @ 60Hz, **HDMI** 4096 x 2160 @ 30P		
Ethernet Port	RJ-45, 10/100/1000M Base-TX (Auto-negotiating, Auto MDI/MDI-X, LED indicators)		
USB port	USB 2.0 x 2, USB 3.0 x 2		

ttySA0	Internal communication with the I-97K series modules in slots		
Console	RS-232/485 (RxD, TxD and GND for RS-232; Data+, Data- for RS-485); 3000 V_{DC} isolated		
ttyS0	RS-485 (Data+, Data-); 3000 V_{DC} isolated		
ttyS1	RS-232/485 (RxD, TxD, CTS, RTS and GND for RS-232; Data+, Data- for RS-485); 3000 V_{DC} isolated		
ttyS34	RS-232 (RxD, TxD, CTS, RTS, DSR, DTR, CD, RI and GND); 3000 V_{DC} isolated		
Audio	Mic-in and Earphone-out		
I/O Expansion I-9K, I-97K series	N/A	Slot *2	Slot *6
I/O Expansion e-9K series	N/A	1 x eBusx1, 1 x eBusx4	2 x eBusx1, 1 x eBusx4
Mechanical			
Dimensions (W x H x D, unit: mm)	239 x 164 x 133	300 x 164 x 133	422 x 164 x 133
Installation	Wall mounting / DIN-rail mounting		
Environmental			
Operating Temperature	-25 °C to +60 °C		
Storage Temperature	-30 °C to +80 °C		
Ambient Relative Humidity	10 % to 90 % RH (non-condensing)		
Power			
Input Range	+19 V_{DC} to +30 V_{DC}		
Isolation	2 kV		
Redundant Power Inputs	Yes, with one power relay (1 A @ 24 V_{DC}) for alarm		
Capacity	80 W supply to CPU and backplane, 40 W supply to I/O expansion slots and USB ports, 120 W in total		
Consumption	30.5 W (Typical) 71 W (CPU Full Load)	31.5 W (Typical) 72 W (CPU Full Load)	32.6 W (Typical) 73 W (CPU Full Load)
Cooling			
Cooling Type	CPU Heat Sink and 180,000 hour long-life fan with Smart Fan function		

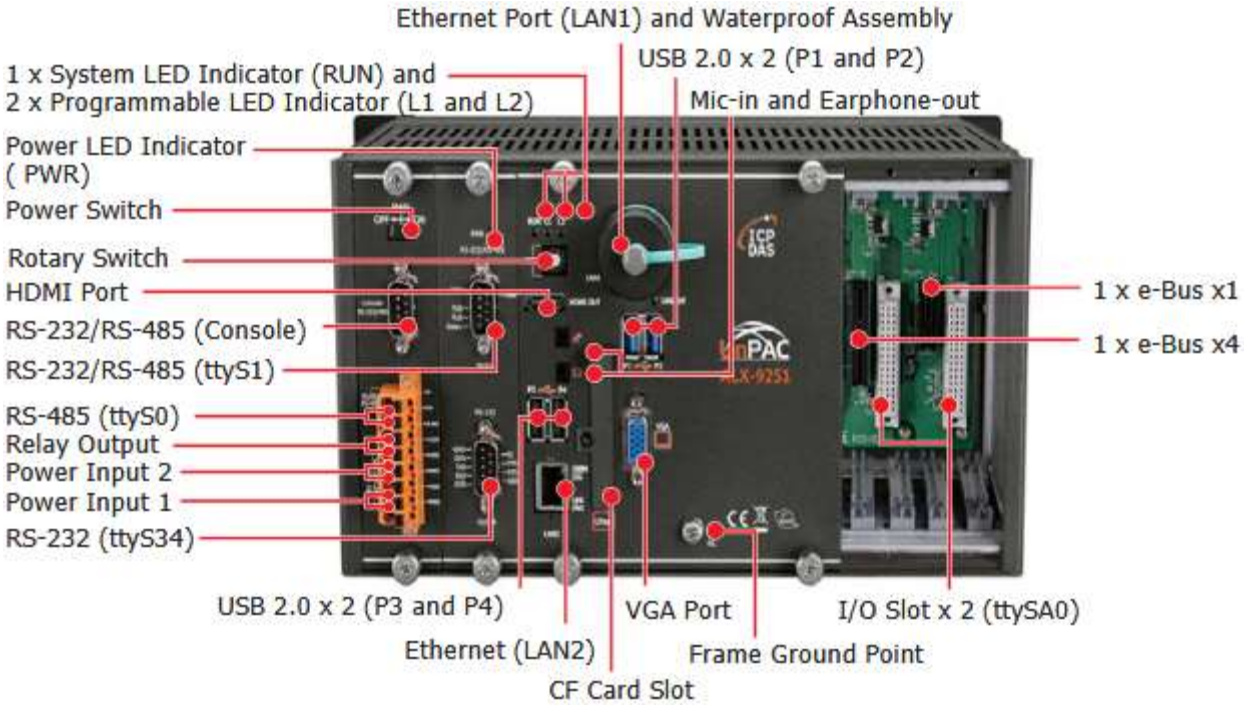
1.3. Overview

The ALX-9x51 Series modules are equipped with several interfaces and peripherals that can be integrated with external systems. Here is an overview of the components and its descriptions.

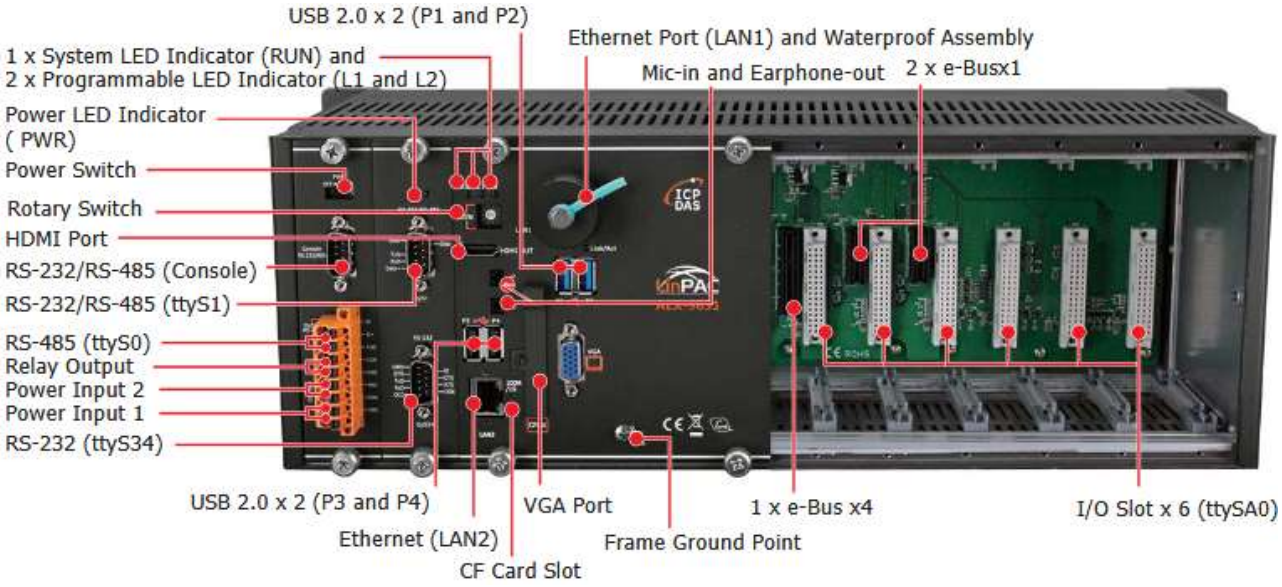
ALX-9051



ALX-9251



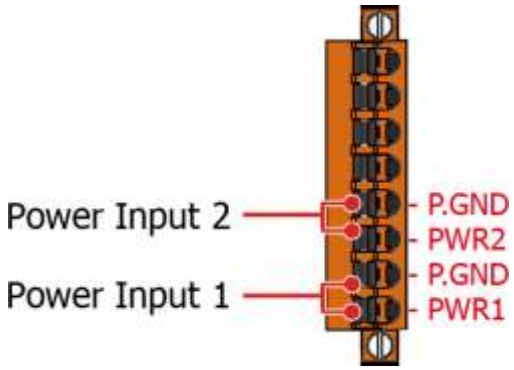
ALX-9651



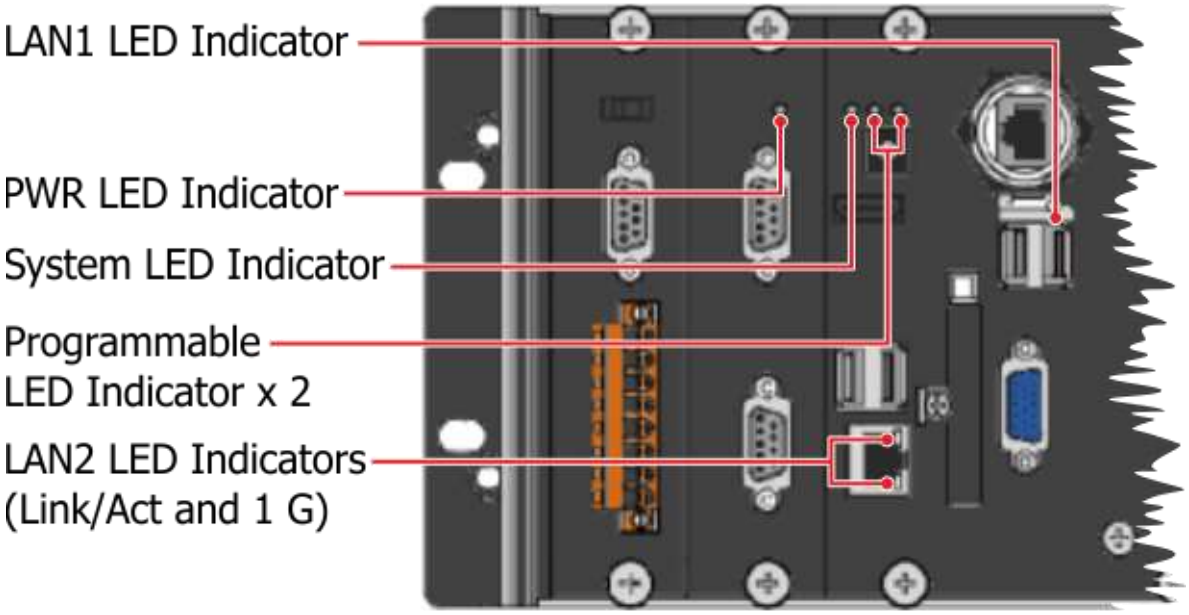
The details of these items are as follows:

Redundant Power (PWR1 and PWR2)

The ALX-9x51 has a terminal with 8-wire; there are 4-wire for redundant power inputs, the details of the redundant power are shown to the side.

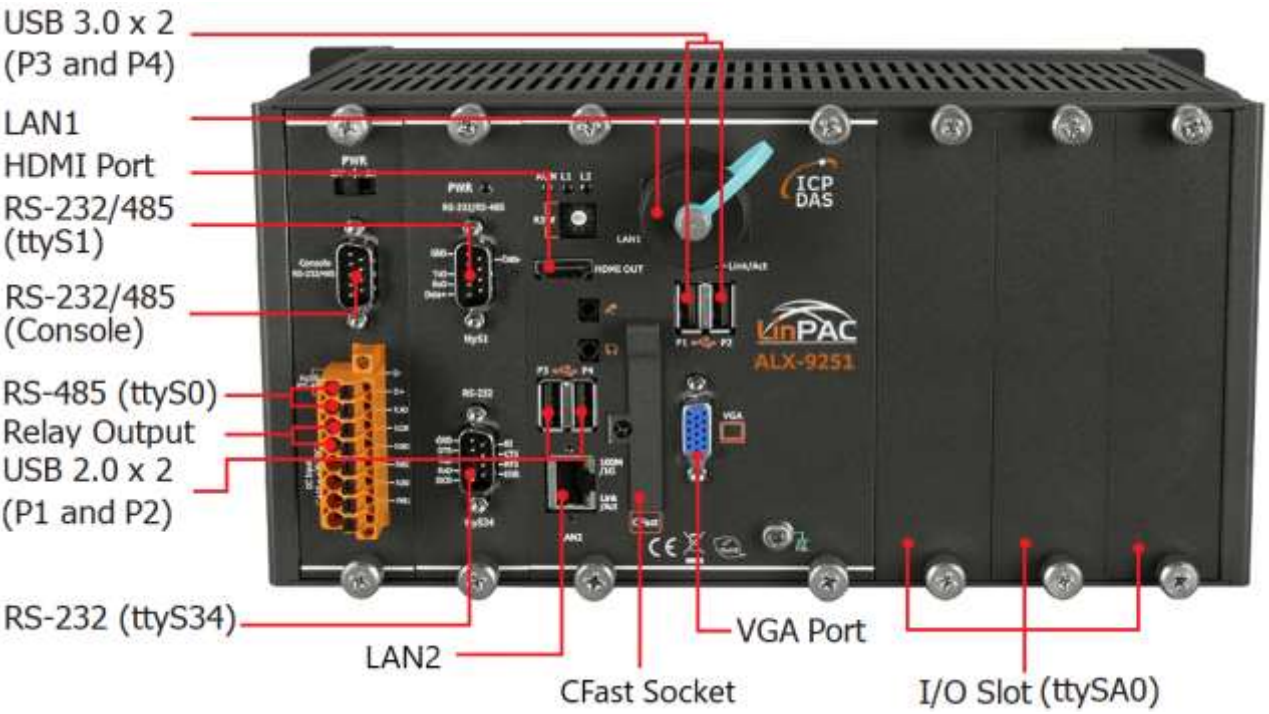


LED Indicators



LED Indicator	Label	State (Color)	Meaning
Programmable LED Indicators	L1 and L2	-	Programmable LED indicators
System LED indicator	RUN	Orange	Programmable LED indicators
PWR LED Indicator	PWR	Green	Power is on
LAN1 LED indicator	Link/Act	Green	The Link is active
		Blinking	Network activity
	1G	Yellow	The network speed is 1 G
LAN2 LED indicator	Link/Act	Green	The Link is active
		Blinking	Network activity
	1G	Orange	The network speed is 1 G

Communication Ports



- **CFast Socket**

  The ALX-9x51 comes with the CFast slot can be used with an optional CFast card to expand storage capacity.

- **LAN Ports, LAN1 and LAN2**

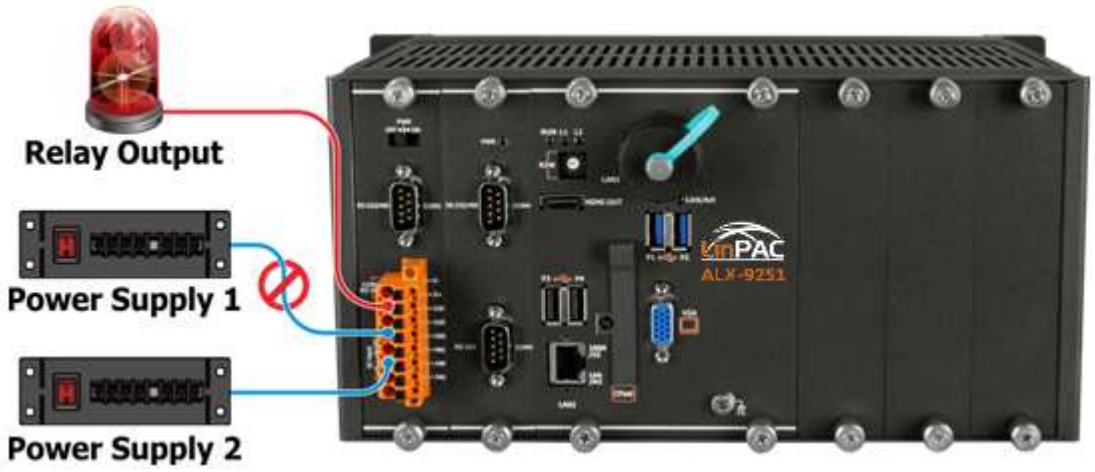
  The ALX-9x51 has two Ethernet ports that can be used to connect the router to the Internet or to other devices.

- **USB 2.0 Ports, P1, P2, USB 3.0 Ports, P3 and P4**

  The ALX-9x51 has four USB 2.0/3.0 ports that can be used to connect the USB devices such as mouse, keyboard or an external USB hard drive.

- **Relay Output**

  The AXP-9x51 has a relay output that can be used to control a light, siren, or other low voltage device when an alarm occurs.



- **VGA Port**

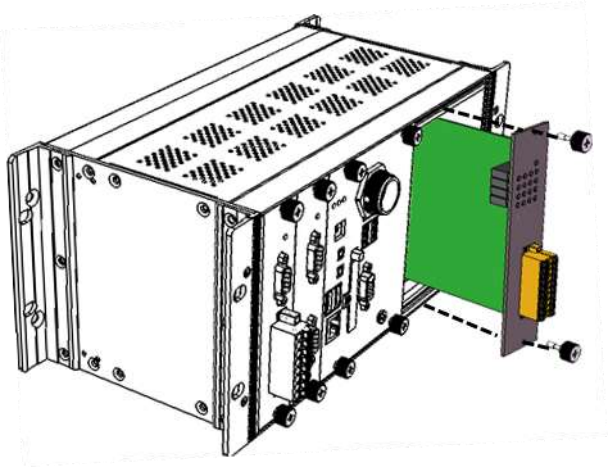
  The ALX-9x51 has a VGA port that can be used with a variety of supported VGA resolutions, and the output resolution covers, 1920 x 1200.

- **HDMI Port**

  The ALX-9x51 has a HDMI port that can be used with a variety of supported HDMI resolutions, and the output resolution covers, 4096 x 2160.

- **ttySA0, Expansion I/O Slot**

  The ALX-9x51 has 0/2/6 I/O slots that can be used to integrate high performance parallel I/O modules (I-9K Series) or serial I/O modules (I-97K series).

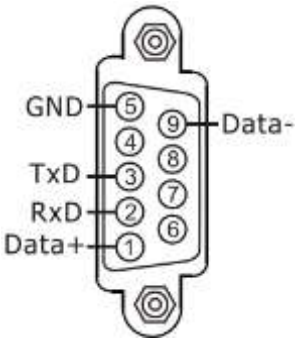


● **Console (RS-232)**

The Console port is a 9-pins RS-232/485 connector that can be configured as either RS-232 or RS-485, that only can select one at a time and its configuration depends on the pin connections as follows:

RS-232 (RXD, TXD and GND)

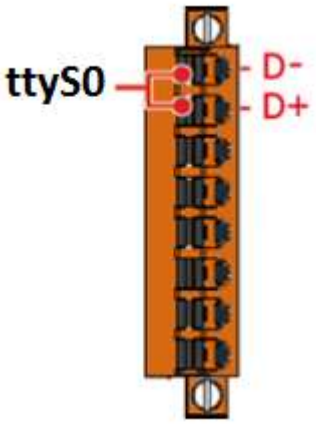
RS-485 (Data+ and Data-)



There is no software configuration or hardware jumper needed.
The details of the Console port specifications are shown to the side.

Note: 16C550 compatible
Port Type: Male
Baud Rate: 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps
Data Bits: 5,6,7, 8
Parity: None, Even, Odd, Mark (Always 1), Space (Always 0)
Stop Bits: 1, 2
FIFO: 64 bytes

● **ttyS0 (2-wire RS-485)**



Note: 16C550 compatible
Port Type: Terminals
Baud Rate: 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps
Data Bits: 5, 6, 7, 8
Parity: None, Even, Odd, Mark (Always 1), Space (Always 0)
Stop Bits: 1, 2
FIFO: 128 bytes

● **ttyS1 (RS-232/RS-485)**

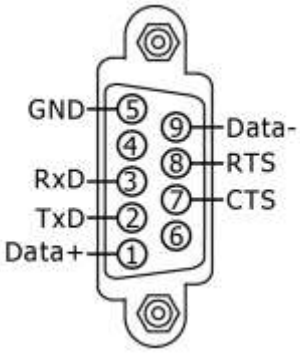
The ttyS1 port is a 9-pins RS-232/RS-485 connector that can be configured as either RS-232 or RS-485, that only can select one at a time and its configuration depends on the pin connections as follows:

RS-232 (RXD, TXD, RTS, CTS and GND)

RS-485 (Data+ and Data-)

There is no software configuration or hardware jumper needed.

The details of the ttyS1 port specifications are shown to the side.



Note: 16C550 compatible

Port Type: Male

Baud Rate: 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps

Data Bits: 5, 6, 7, 8

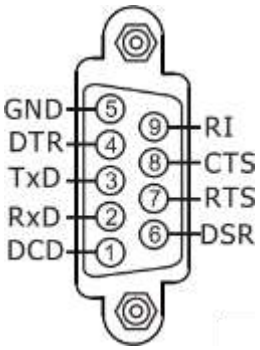
Parity: None, Even, Odd, Mark (Always 1), Space (Always 0)

Stop Bits: 1, 2

FIFO: 128 bytes

● **ttyS34 (RS-232)**

The ttyS34 port is a 9-pins RS-232 connector. The details of the ttyS34 port specifications are shown to the side.



Note: 16C550 compatible

Port Type: Male

Baud Rate: 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps

Data Bits: 5, 6, 7, 8 Parity: None, Even, Odd, Mark (Always 1), Space (Always 0)

Stop Bits: 1, 2

FIFO: 128 bytes

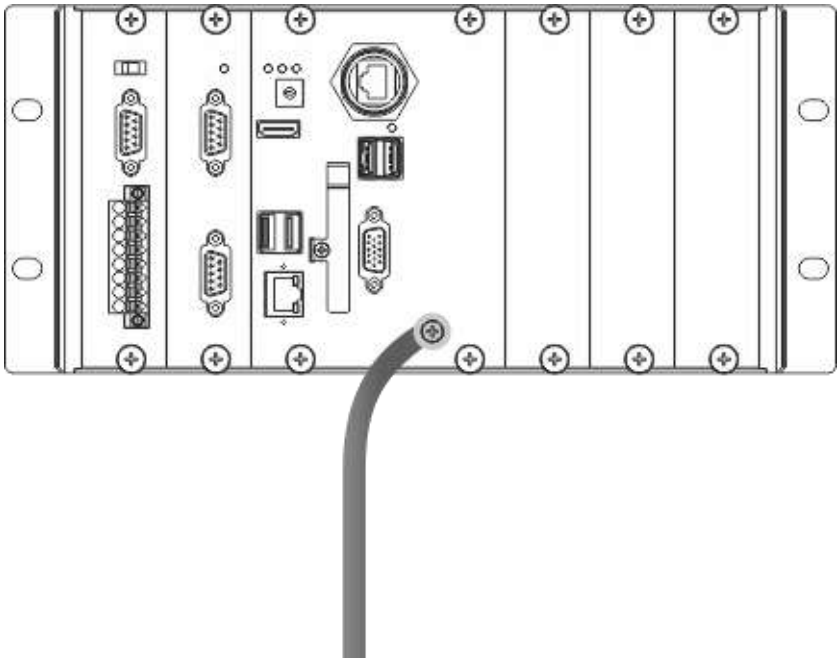
Tips & Warnings

The table below shows the data bit and their corresponding stop bit for Console, ttyS0, ttyS1 and ttyS34.

Word Length	Number of Stop Bits
5, 6, 7, 8	1
5	1.5
6, 7, 8	2

Frame Ground Point

The frame ground point is a small piece of metal that can be used to terminate the shield.

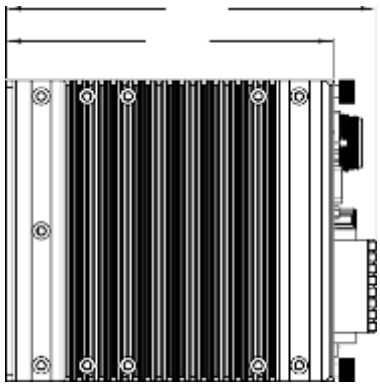


1.4. Dimensions

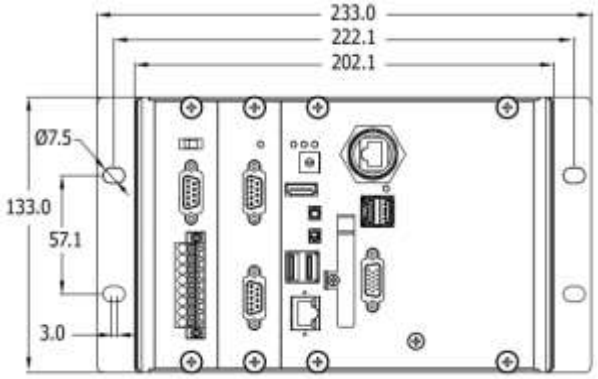
The diagrams below provide the dimensions of the ALX-9x51 to use in defining your enclosure specifications. Remember to leave room for potential expansion if you are using other components in your system.

The height dimension is the same for all ALX-9x51. The width depending on your choose of I/O expansion slots. All dimensions are in millimeters.

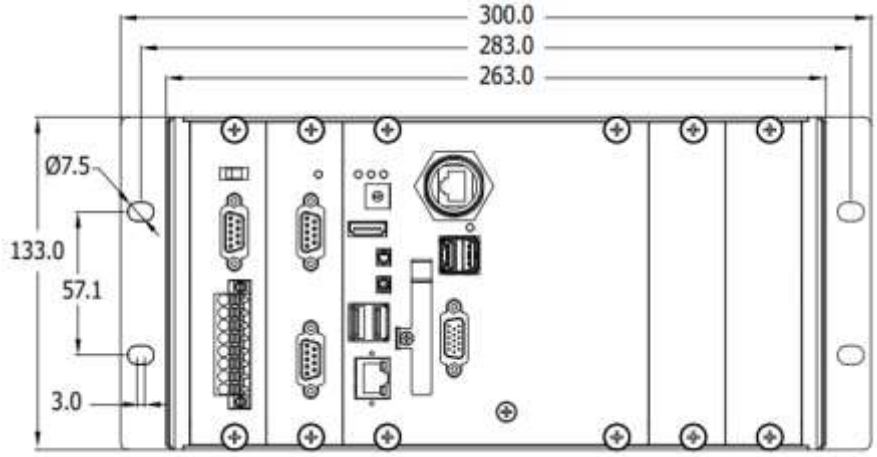
ALX-9051/ALX-9251/ALX-9651

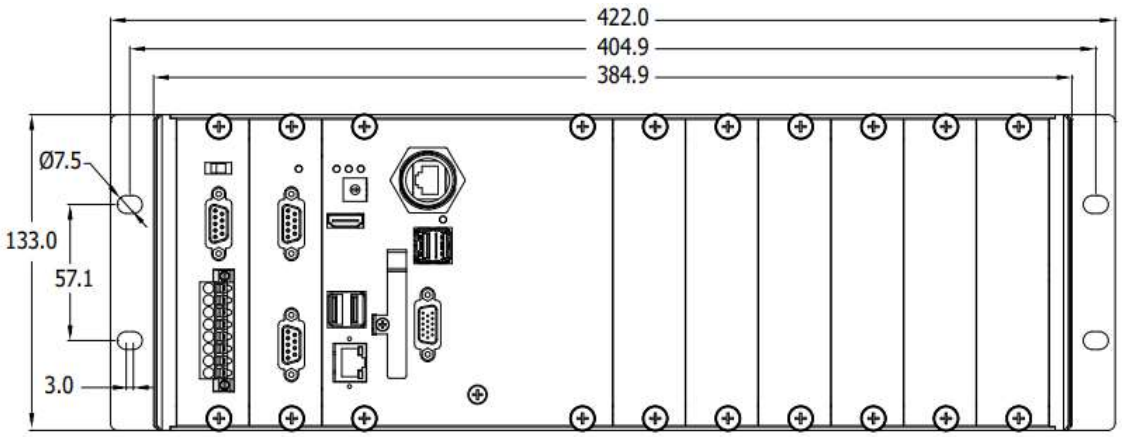


ALX-9051



ALX-9251

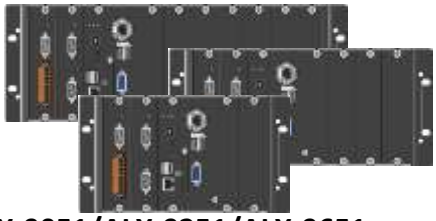




2. Getting Started

This chapter provides a guided tour of the ALX-9x51 installation and configuration that describes the steps needed to download, install, configure, and run the basic procedures for user working with the ALX-9x51 for the first time.

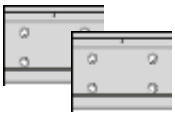
Before starting any task, please check the package contents. If any of the following package contents are missing or damaged, contact your dealer, distributor.



ALX-9051/ALX-9251/ALX-9651

Quick Start Guide

RJ-45 Waperproof Assembly



60 mm DIN-Rail Clip * 2

M3x6L Screw * 8

Screw Driver (1C016) 2.4 mm

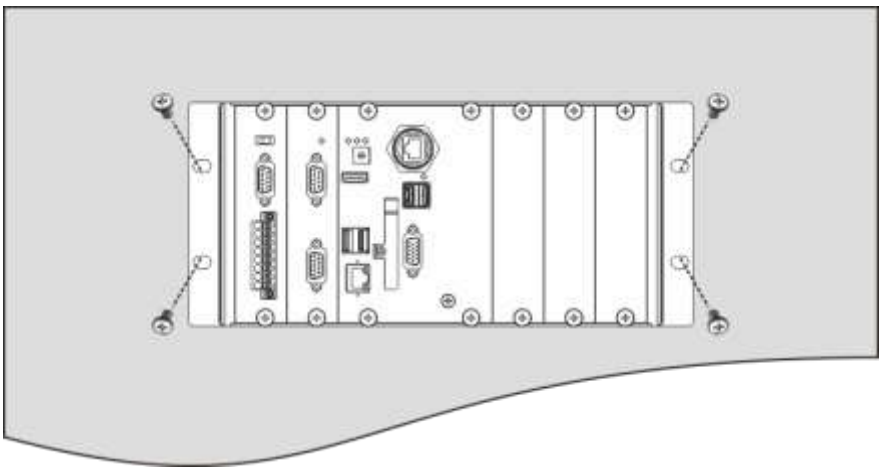
2.1. Mounting the ALX-9x51

The ALX-9x51 can be mounted either directly to a wall/panel, or onto a stainless 35mm DIN rail.

Wall/Panel mounting

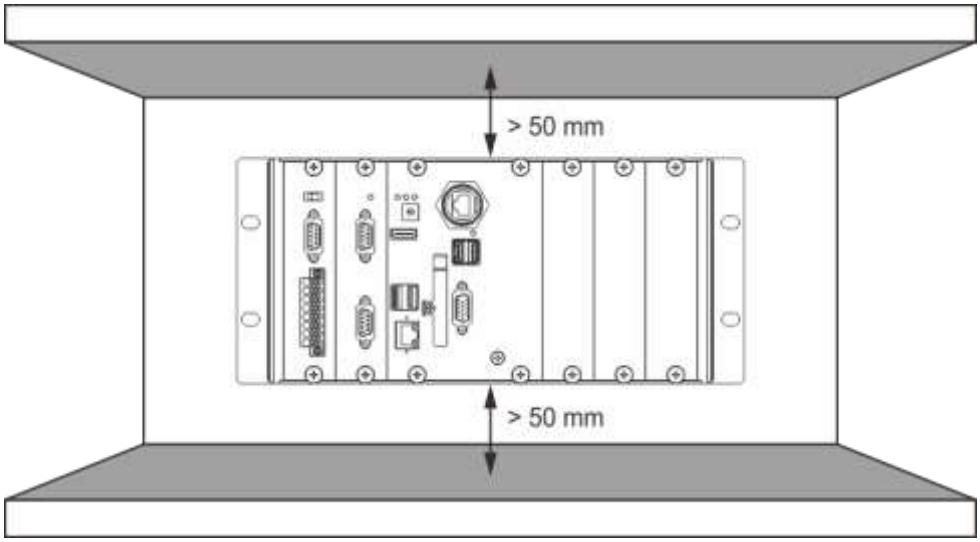
Step 1: Install the four mounting screws into the 4 keyhole mounting holes

Step 2: Fasten the screws securely

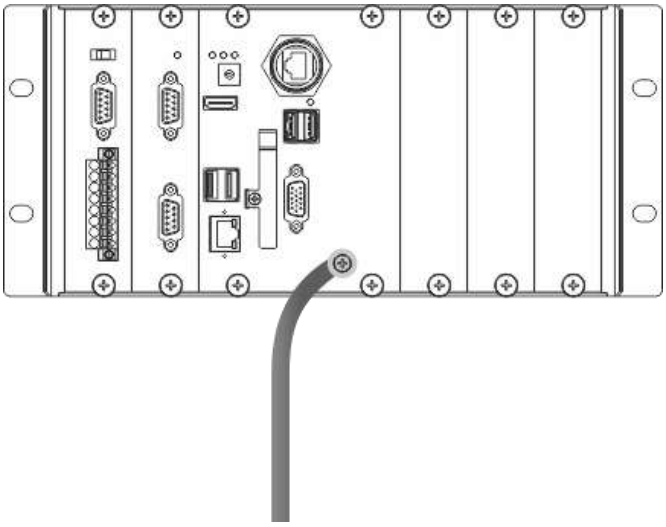


Tips & Warnings

There must be a minimum clearance of 50mm between the ALX-9x51 and the top and bottom side of the enclosure panel.



Step 3: Connect the ground lead to the frame ground point

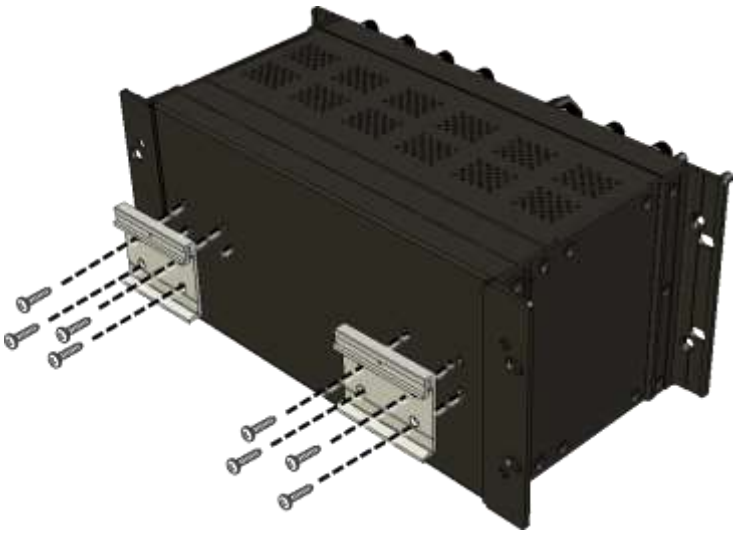


Tips & Warnings

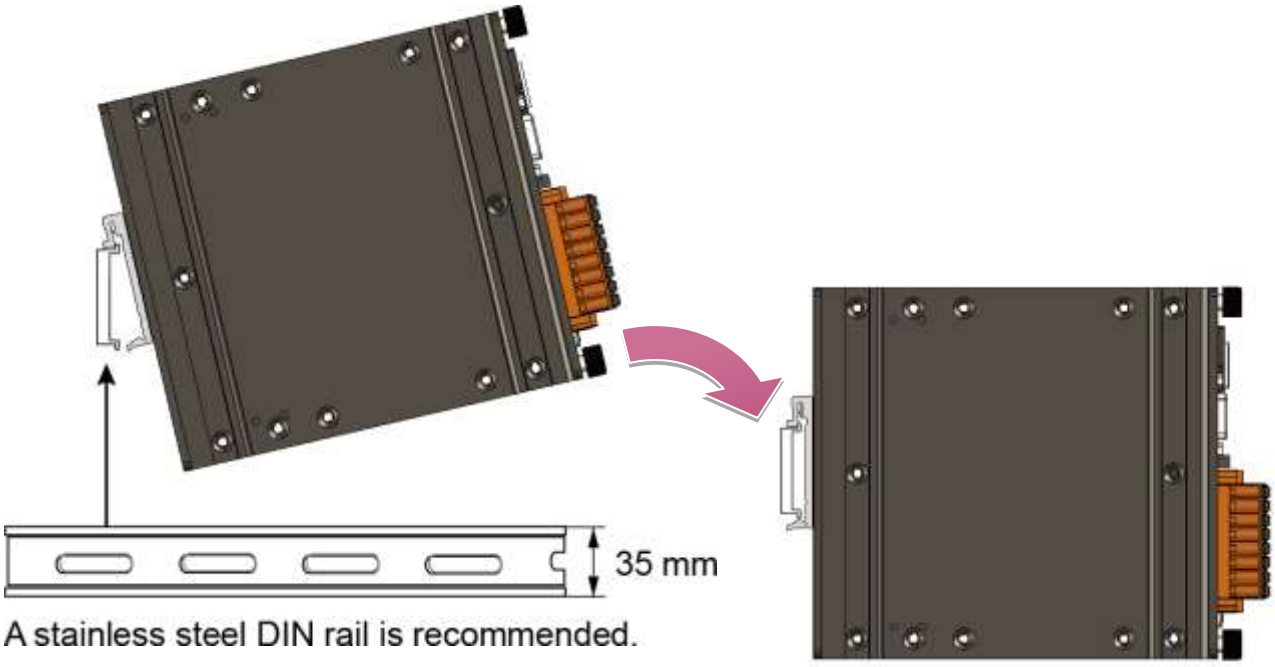
A good common ground reference (earth ground) is essential for proper operation of the ALX-9x51. One side of all control circuits, power circuits and the ground lead must be properly connected to earth ground by either installing a ground rod in close proximity to the enclosure or by connecting to the incoming power system ground. There must be a single-point ground (i.e. copper bus bar) for all devices in the enclosure that require an earth ground.

DIN Rail mounting

Step 1: Fasten the DIN rail clip to the ALX-9x51

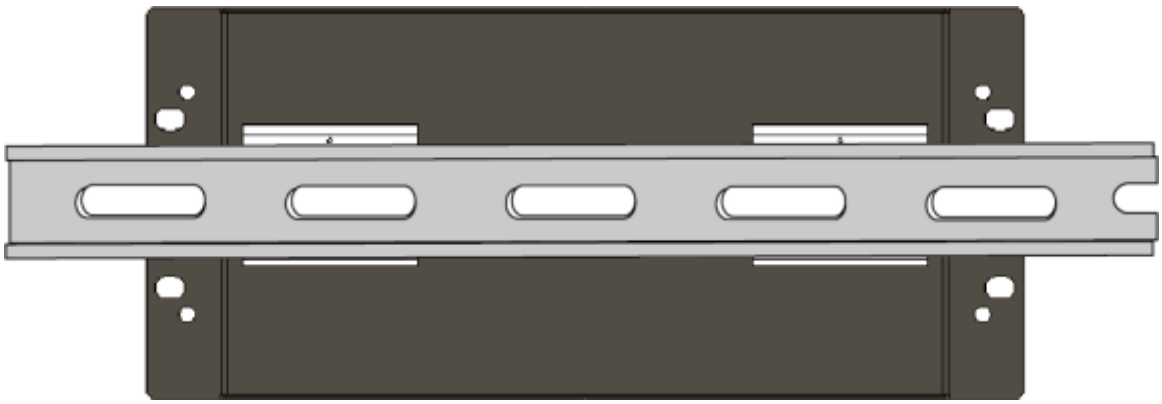


Step 2: Clip the device onto a stainless DIN rail

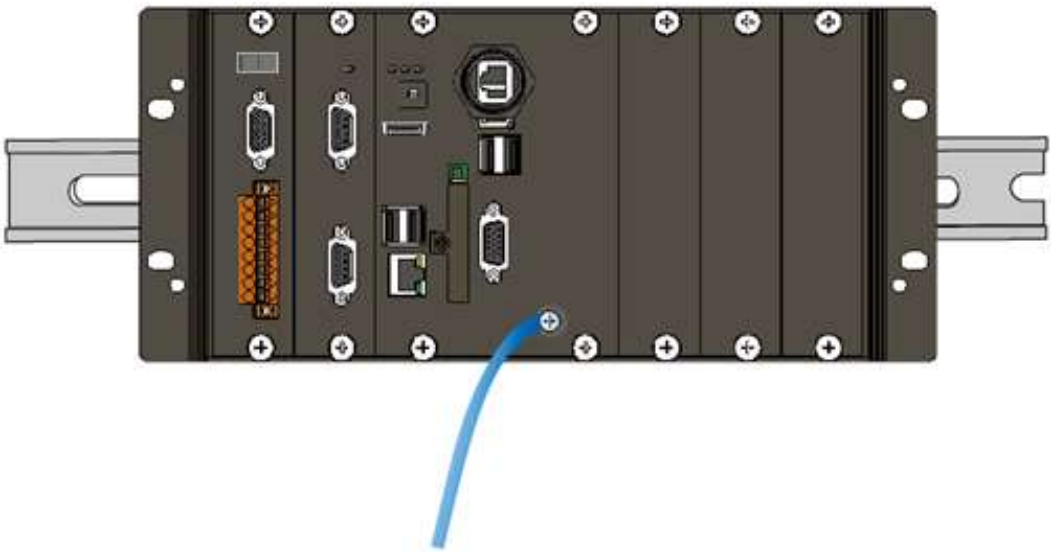


Tips & Warnings

For DIN rail mounting, it is strongly recommended that only a stainless steel DIN rail be used to support the weight of ALX-9x51 system, providing stability and preventing ALX-9x51 from leaning



Step 3: Connect the ground lead to the frame ground point



Tips & Warnings

A good common ground reference (earth ground) is essential for proper operation of the ALX-9x51. One side of all control circuits, power circuits and the ground lead must be properly connected to earth ground by either installing a ground rod in close proximity to the enclosure or by connecting to the incoming power system ground. There must be a single-point ground (i.e. copper bus bar) for all devices in the enclosure that require an earth ground.

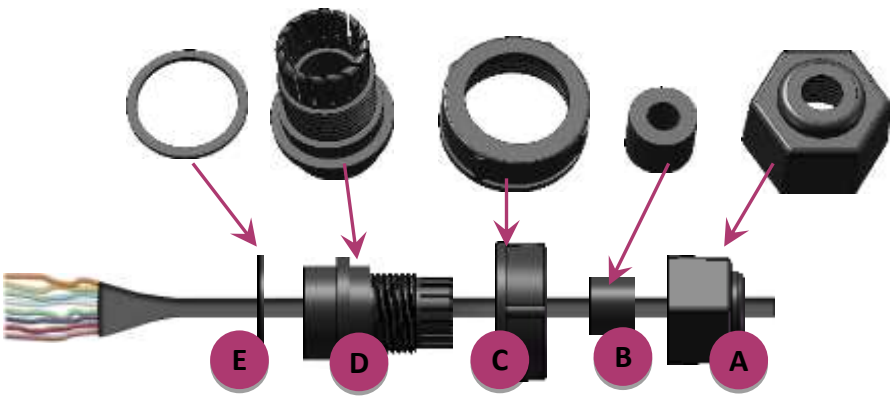
2.2. Installing the RJ-45 waterproof connector assembly

The ALX-9x51 is equipped with an RJ-45 waterproof connector to protect the connection in vibrate environment.
The RJ-45 waterproof connector is optional for use with LAN1 port. If you do not need the RJ-45 waterproof connector, you can remove the cap and just plug in a regular Ethernet cable.

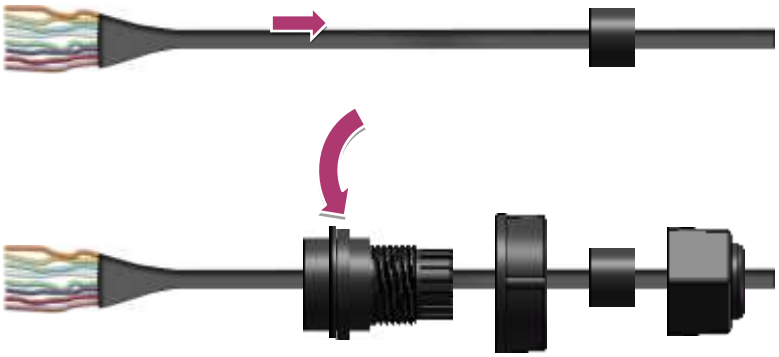
If you want to use the RJ-45 waterproof connector for protecting the connection, follow the instructions below.

Step 1: Remove the RJ-45 connector from the RJ-45 cable

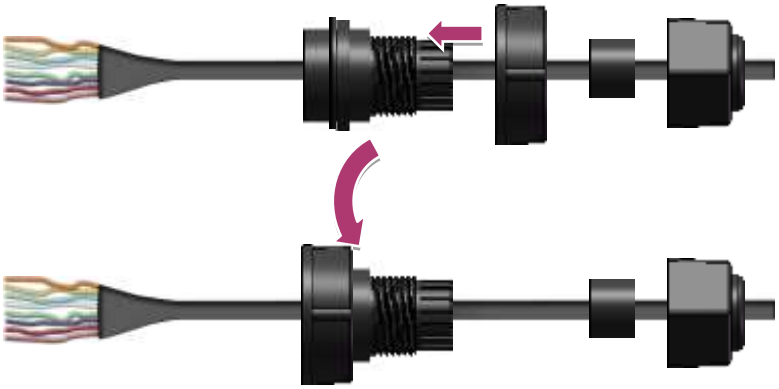
Step 2: Feed the end of the RJ-45 cable through the (A) sealing nut, (B) rubber sealing insert, (C) cable gland base, (D) clamping ring and (E) panel gasket



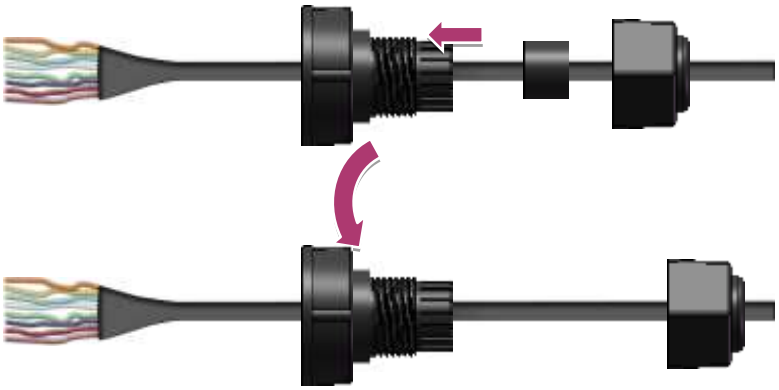
Step 3: Wrap the (E) panel gasket around the (D) clamping ring



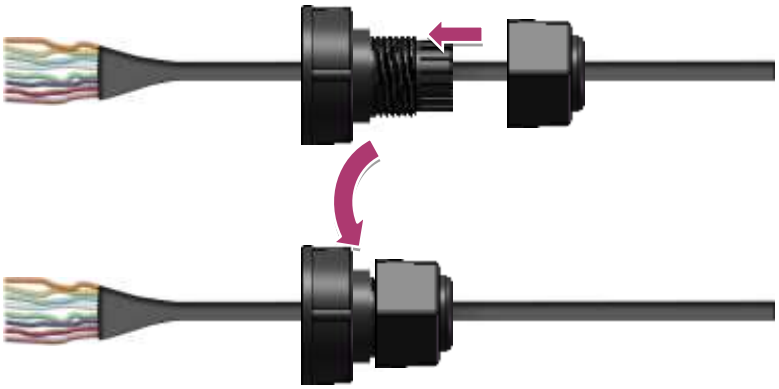
Step 4: Wrap the (C) cable gland base around the (D) clamping ring



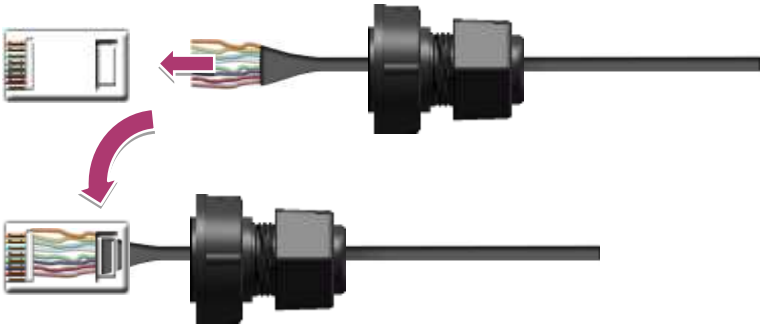
Step 5: Insert the (B) rubber sealing insert into the (D) clamping ring



Step 6: Push the (E) sealing nut forward and Hand-tighten it to seal the assembly



Step 7: Insert the RJ-45 cable into the RJ-45 connector



Step 8: Push the RJ-45 waterproof connector assembly forward

Step 9: Insert the Ethernet cable and screw the RJ-45 waterproof into the receptacle

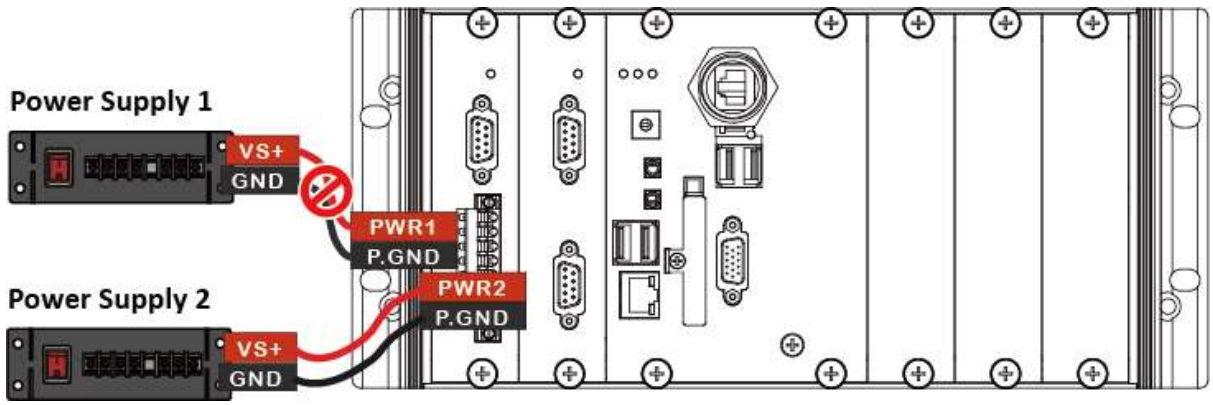
2.3. Deploying a Basic ALX-9x51 System

The ALX-9x51 provides a variety of communication interface to suit a range of application. Here is a simple application for using the ALX-9x51.

Step 1: Connect the positive terminal (+) of the power supply to the terminal PWR1/2 and the negative terminal (-) of the power supply to the P.GND

Tips & Warnings

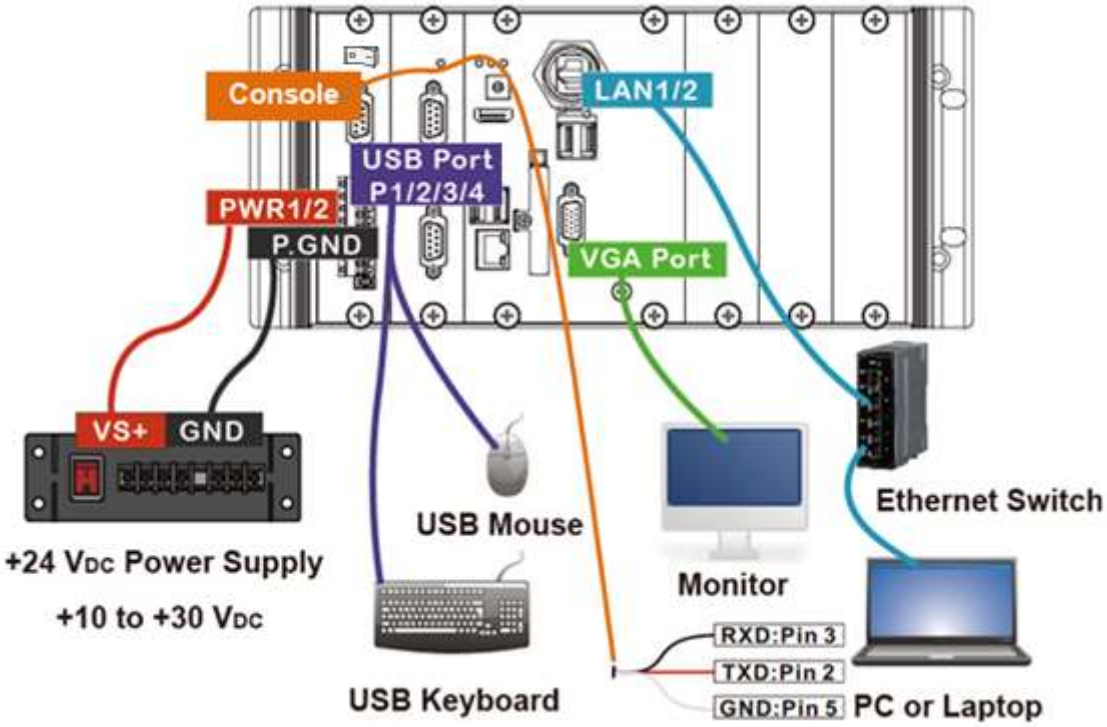
1. The input range of power supply is +10 to +30 V_{DC}.
2. The ALX-9x51 have two power inputs that can be connected simultaneously to the two independent power sources. If one power source fails, the other source takes over automatically. Redundant power input help assure non-stop operation of the ALX-9x51.



Step 2: Connect the USB mouse or the USB keyboard to the USB port

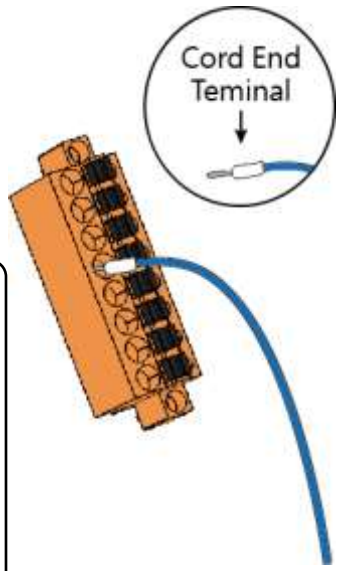
Step 3: Connect the monitor to the VGA or HDMI port

Step 4: Connect to PC or the laptop to the LAN port via an Ethernet switch



Tips & Warnings

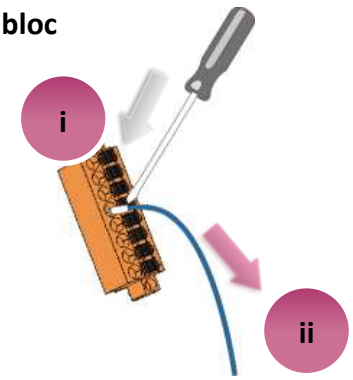
The metal part of the cord end terminal on the wire can be direct wired to the terminal.



Remove the wiring from the terminal bloc

i. Use the screwdriver to push the black clip in

ii. Remove the wiring



2.4. Inserting the I/O Modules

ALX-9x51 has 0/2/6 I/O expansion slots to support I-9K and I-97K series I/O modules.
ALX-9x51 also has 2/3 e-Bus I/O slots to support e-9K series I/O expansion modules.
Before choosing the right I/O modules, you first need to know the I/O expansion capacities in order to choose the best expansion module for achieving maximal efficiency. For more information about the I/O expansion modules that are compatible with the ALX-9x51, please refer to:

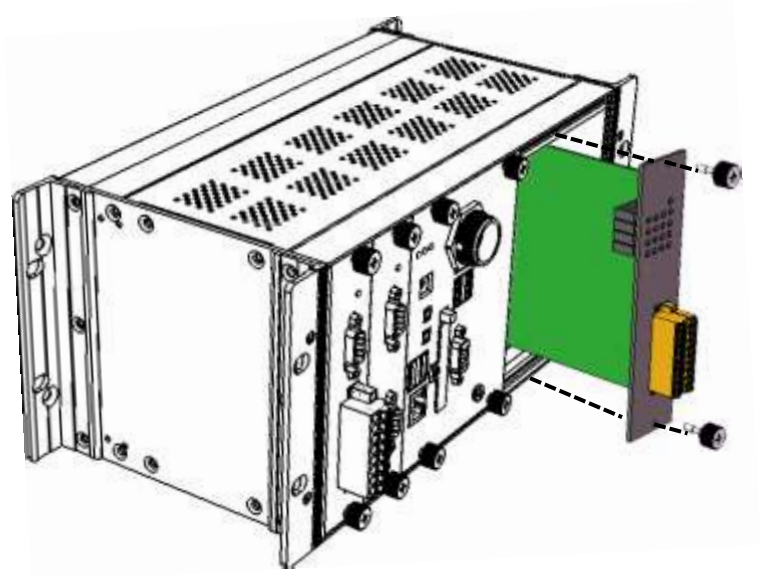
I-9K and I-97K series

https://www.icpdas.com/en/product/guide+Remote_I_O_Module_and_Unit+PAC_%EF%BC%86amp;_Local_I_O_Modules+I-9K_I-97K_Series

e-9K series

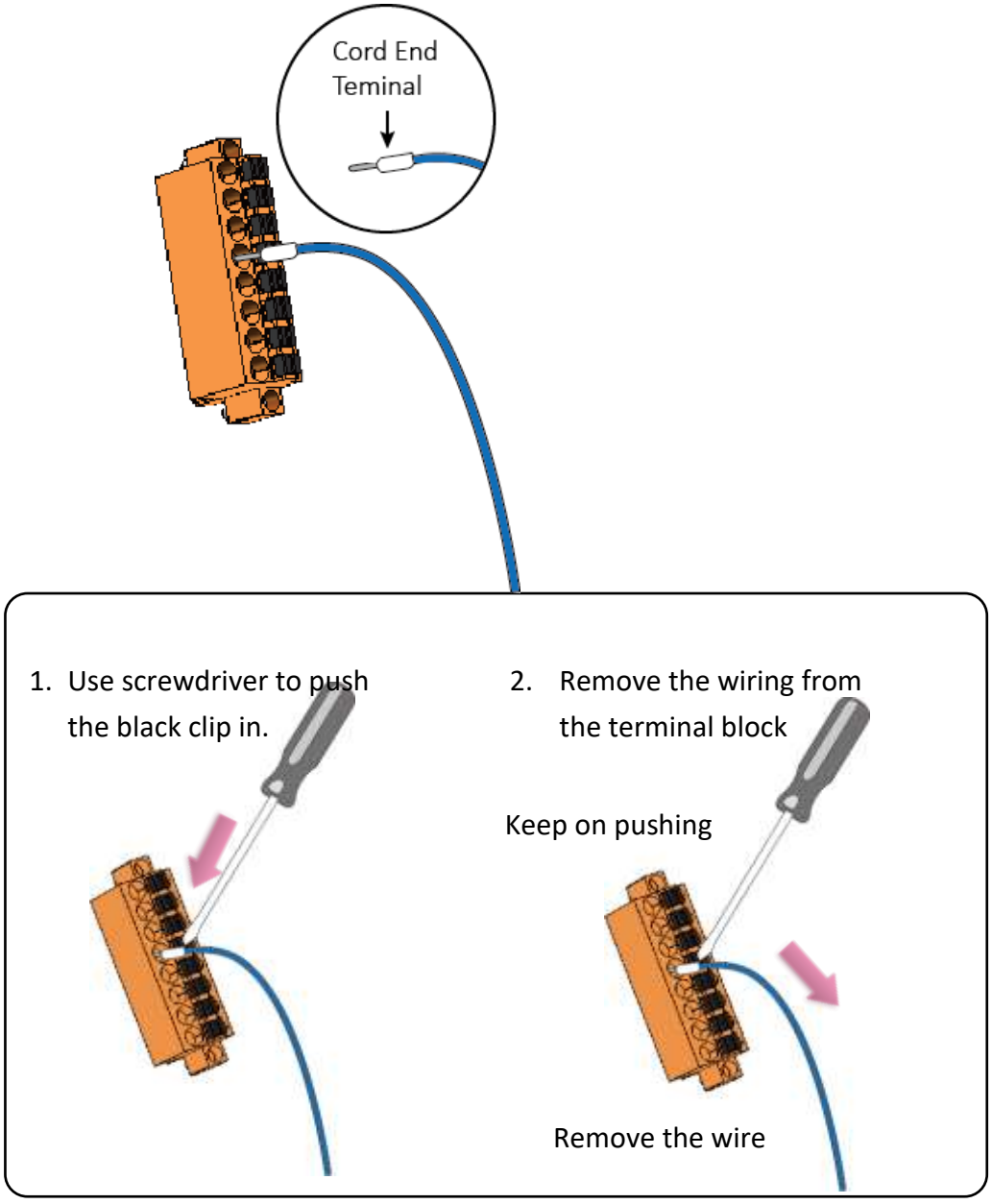
https://www.icpdas.com/en/product/guide+Remote_I_O_Module_and_Unit+PAC_%EF%BC%86amp;_Local_I_O_Modules+e-9k_Series#384

Step 1: Insert the I/O module



Step 2: Wiring connection

The metal part of the cord end terminal on the wire can be direct wired to the terminal of ALX-9x51.



Tips & Warnings

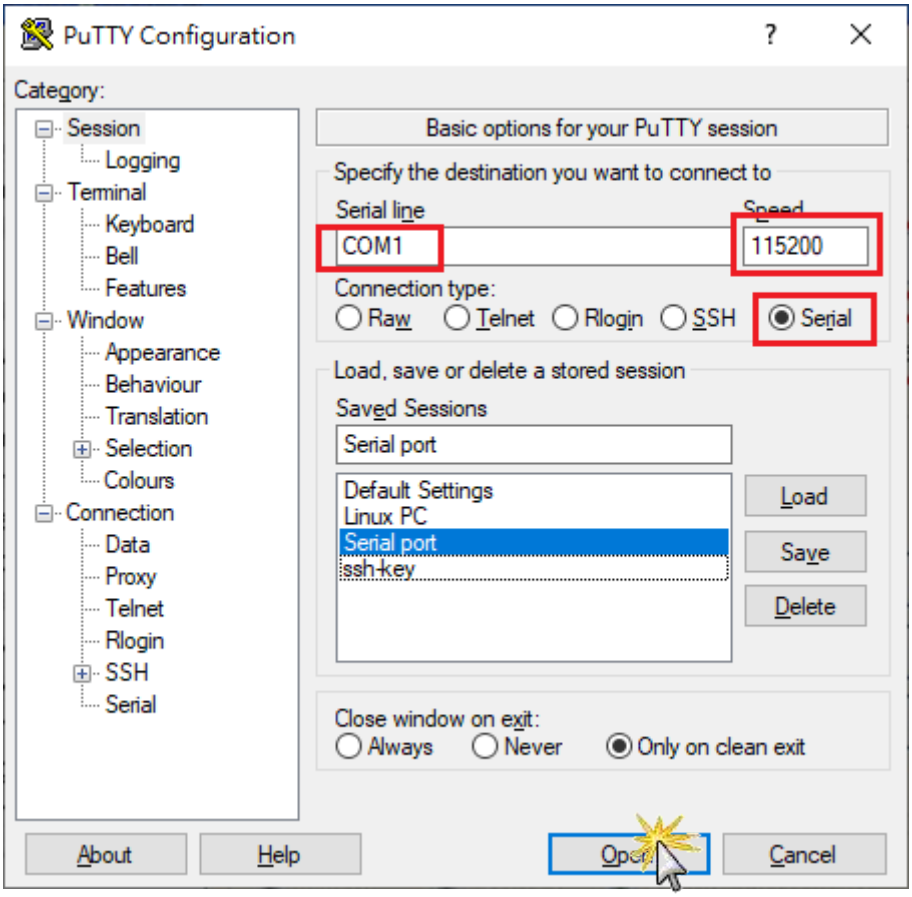
If you do not expand the I/O module full, please keep the top case of the unused slot to protect the backplane from dirt, dust and damage from foreign objects.

2.5. Console Port Connection

ALX-9x51 supports remote connection through the "Console" port without the need for the internet. Users can connect to ALX-9x51 by following the steps below:

Step 1: Users can connect to ALX-9x51 by selecting software (such as Putty or others) through the "Console" port.

Step 2: If user chose the “Console”, user can set the baud rate “115200” to connect the device.



Step 3: After user connect to the ALX-9x51 from the “Console” port, user can input default ID “root” and password “icpdas” to login.

2.6. LAN1/LAN2 Network Configuration

If the user successfully logs into ALX-9x51 through the Console port, the user can use the "ifconfig" command to obtain the IP address for LAN1/LAN2.

2.6.1. Set Static IP

The LAN1/LAN2 of factory setting use DHCP. If user wants to set the static IP address, user can refer to below steps:

Step 1: Using the Linux command "vim" to modify the file

"/etc/netplan/ 01-netcfg.yaml".

Step2: Set a static IP. The configuration method is shown in the figure below.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      addresses:
        - 192.168.2.207/24
      routes:
        - to: default
          via: 192.168.2.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
    eth1:
      addresses:
        - 192.168.2.208/24
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Step3: After user save the file and use "netplan apply" command to restart internet interface, Users can use new network configurations on ALX-9x51.

2.6.2. Configuring network bonding

A network bond is a method to combine or aggregate physical and virtual network interfaces to provide a logical interface with higher throughput or redundancy. In a bond, the kernel handles all operations exclusively. You can create bonds on different types of devices, such as Ethernet devices or VLANs.

To configure network bonding on ALX-9x51 with static and dynamic IP, follow these steps:

DHCP:

Step 1: Using the Linux command "vim" to modify the file "/etc/netplan/01-netcfg.yaml".

Step2: Set up a DHCP IP as shown in the diagram below.

```
network:
  version: 2
  ethernets:
    eth0: {}
    eth1: {}
  bonds:
    bond0:
      interfaces: [eth0, eth1]
      dhcp4: true
      parameters:
        mode: active-backup
        primary: eth0
        mii-monitor-interval: 100
```

Step3: After user save the file and use "netplan apply" command to restart internet interface, Users can use new network configurations on ALX-9x51.

```
root@icpdas:~# ifconfig
bond0: flags=5187<UP,BROADCAST,RUNNING,MASTER,MULTICAST>  mtu 1500
        inet 192.168.2.232  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::1074:8ff:feeb:b7b0  prefixlen 64  scopeid 0x20<link>
        ether 12:74:08:eb:b7:b0  txqueuelen 1000  (Ethernet)
        RX packets 5791  bytes 8579009 (8.5 MB)
        RX errors 0  dropped 13  overruns 0  frame 0
        TX packets 3402  bytes 262730 (262.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST>  mtu 1500
        ether 12:74:08:eb:b7:b0  txqueuelen 1000  (Ethernet)
        RX packets 12419223  bytes 1295147151 (1.2 GB)
        RX errors 0  dropped 1715968  overruns 0  frame 0
        TX packets 2843422  bytes 3737471391 (3.7 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xa1300000-a1320000

eth1: flags=6147<UP,BROADCAST,SLAVE,MULTICAST>  mtu 1500
        ether 12:74:08:eb:b7:b0  txqueuelen 1000  (Ethernet)
        RX packets 11732341  bytes 1161119344 (1.1 GB)
        RX errors 0  dropped 1715939  overruns 0  frame 0
        TX packets 157548  bytes 8839839 (8.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xa1200000-a127ffff
```

Step4: If the user doesn't know the IP, they can log in through the console port to retrieve it.

Static IP:

Step 1: Using the Linux command “vim” to modify the file

“/etc/netplan/01-netcfg.yaml”.

Step2: Set up a static IP as shown in the diagram below.

```
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: no
    eth1:
      dhcp4: no
  bonds:
    bond0:
      interfaces: [eth0, eth1]
      addresses: [192.168.2.240/24]
      gateway4: 192.168.2.1
      nameservers:
        addresses: [8.8.8.8]
      parameters:
        mode: active-backup
        primary: eth0
        mii-monitor-interval: 100
```

Step3: After user save the file and use "netplan apply" command to restart internet interface, Users can use new network configurations on ALX-9000.

```
root@icpdas:~# ifconfig
bond0: flags=5187<UP,BROADCAST,RUNNING,MASTER,MULTICAST>  mtu 1500
        inet 192.168.2.240  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::1074:8ff:feeb:b7b0  prefixlen 64  scopeid 0x20<link>
        ether 12:74:08:eb:b7:b0  txqueuelen 1000  (Ethernet)
        RX packets 33271  bytes 10737783 (10.7 MB)
        RX errors 0  dropped 1247  overruns 0  frame 0
        TX packets 4095  bytes 363976 (363.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST>  mtu 1500
        ether 12:74:08:eb:b7:b0  txqueuelen 1000  (Ethernet)
        RX packets 12434124  bytes 1296411027 (1.2 GB)
        RX errors 0  dropped 1715968  overruns 0  frame 0
        TX packets 2844090  bytes 3737570145 (3.7 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xa1300000-a1320000

eth1: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST>  mtu 1500
        ether 12:74:08:eb:b7:b0  txqueuelen 1000  (Ethernet)
        RX packets 11744920  bytes 1162014242 (1.1 GB)
        RX errors 0  dropped 1715939  overruns 0  frame 0
        TX packets 157573  bytes 8842331 (8.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xa1200000-a127ffff
```

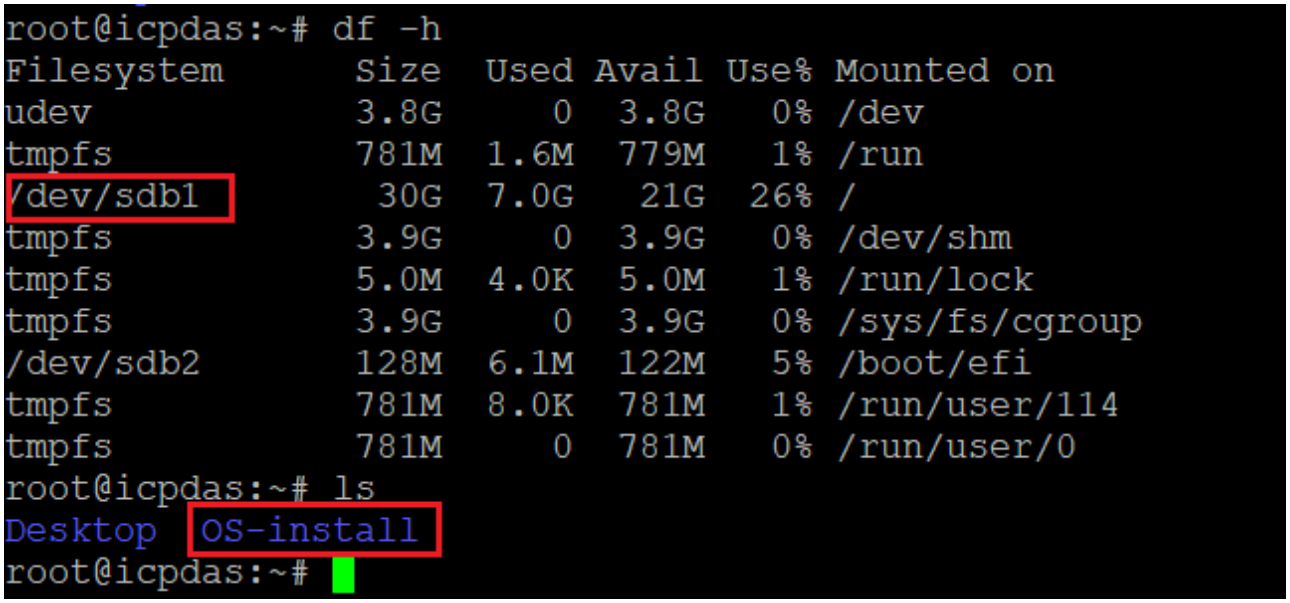
2.7. Recovering factory OS from USB disk to Flash

If users want to restore the factory operating system on ALX-9x51, they can follow the steps below to restore the Linux operating system from the Flash:

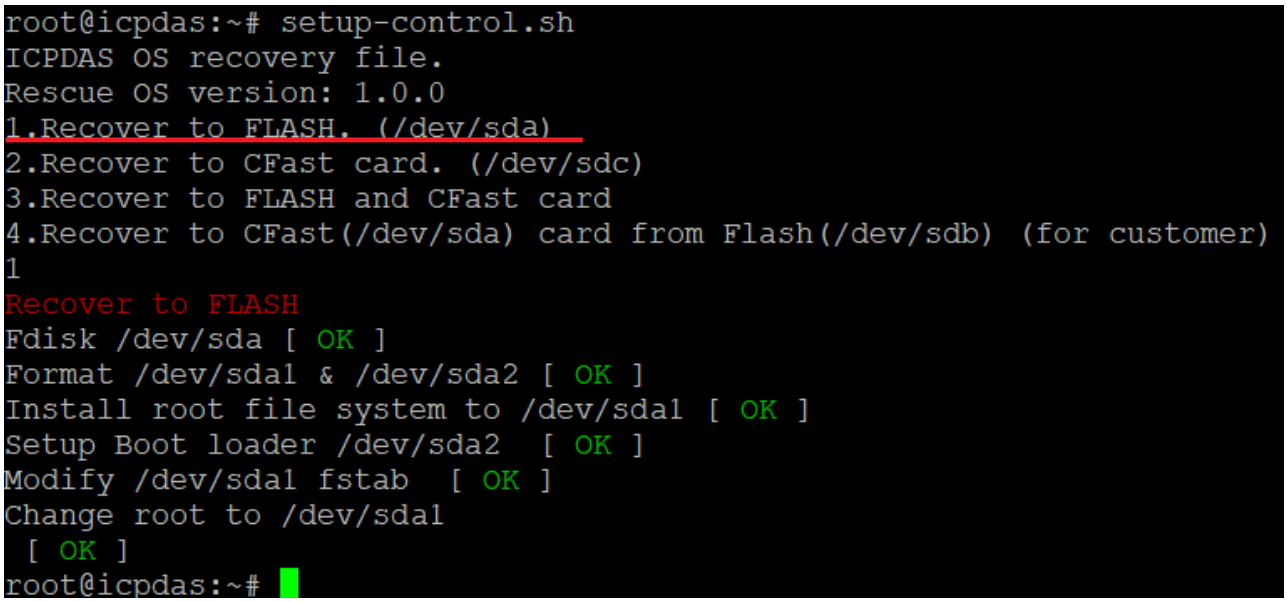
Step 1: You can download the OS image from the following URLs, then use the tool **HDDRawCopy (please download it from the internet) to write the OS image to a USB drive (with a capacity of at least 32 GB):**

ALX-9x51: https://www.icpdas.com/tw/download/show.php?num=9704&model=ALX-9051
Then select OS image file to download.

Step 2:
Boot up ALX-9x51 with USB disk OS. Ensure that the system boots from USB OS (/dev/sdb). Use the `ls` command to confirm that the "OS-install" directory is in the "/root" directory.



Step 2: Executing the command “setup-control.sh” and choose option ‘1’ to recover the factory Linux OS to flash disk.

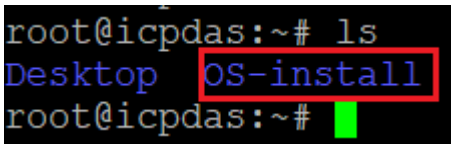


2.8. Update USB disk recover file version

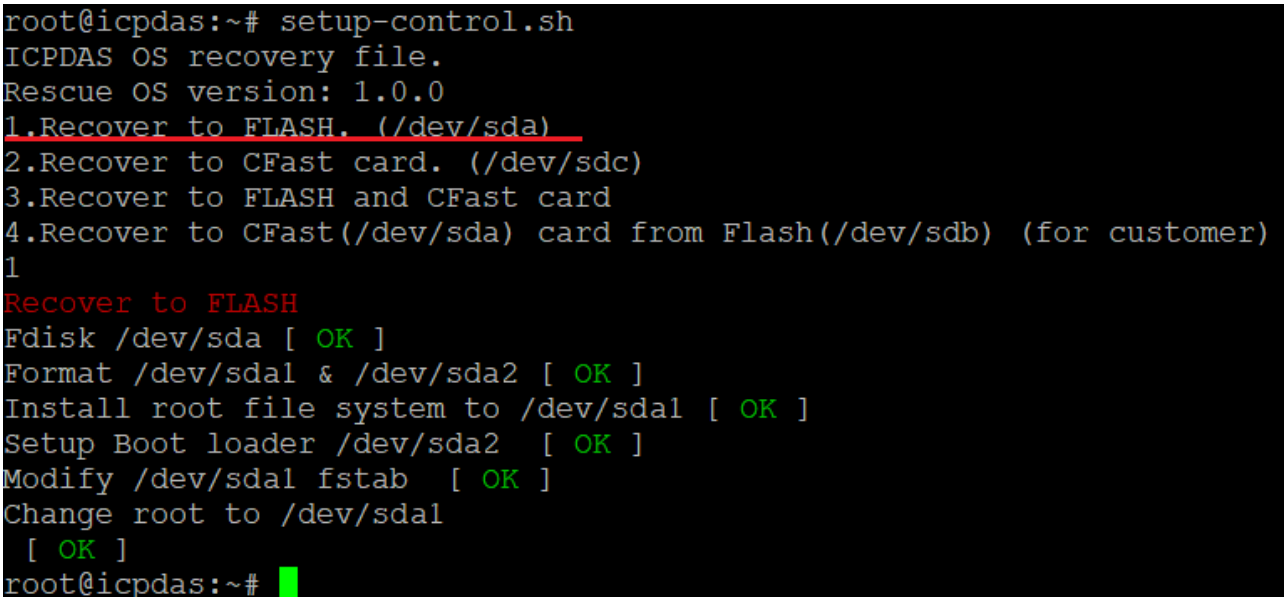
User can follow below steps to update recover file to recent version.

1. Boot up with USB disk and download recover file by below link.
   https://www.icpdas.com/web/product/download/pac/linux/alx-9000/software/os/OS-install-alx9x51.tgz
   Please download “ALX-9x51 OS rootfs” option.

2. Delete old recover folder (“OS-install”) and unzip downloaded file (OS-install-ALX_9x51-1.0.0).



3. Execute the command "setup-control.sh" and choose option '1' to restore the latest version of the factory Linux operating system to the Flash Disk.



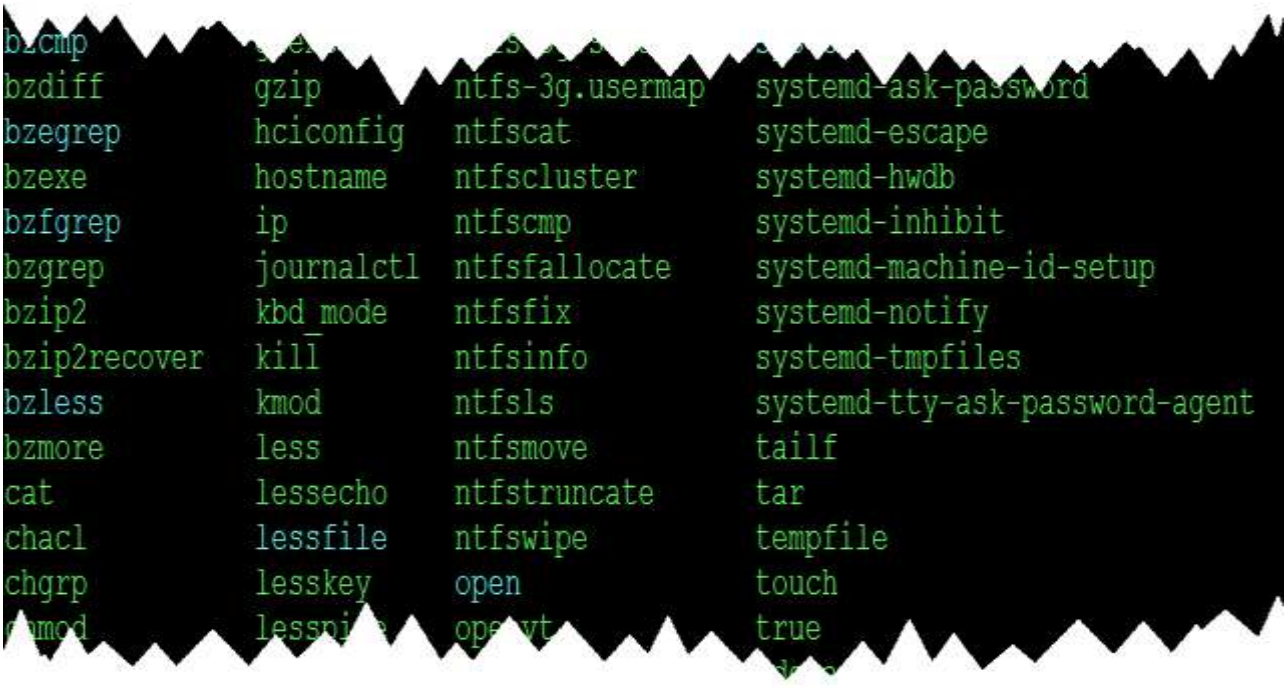
3. Instructions for ALX-9x51

This chapter provides a brief introduction of the ALX-9x51 service tools and its benefits.

There are several tools and utilities built-in and designed for use with ALX-9x51.
Some of these are pre-installed on ALX-9x51 and can work directly on ALX-9x51, and some of these are supporting tools and can help you to manage the ALX-9x51 remotely on a PC.

3.1. Basic Linux Command

In ALX-9x51, you can use basic Linux commands to configure the Linux operating system or obtain system information.



3.1.1. Linux Command “ls”

ls: list the file information -> (like dir in DOS)

Parameter：

(1) -l：list detailed information of file

```
root@icpdas:/# ls -l
total 100
drwxr-xr-x     2    root root 4096     Mar 21 03:56  bin
drwxr-xr-x     3    root root 4096     Apr  8 08:41  boot
drwxrwxr-x     2    root root 4096     Jan  5 09:02  cdrom
drwxr-xr-x     19   root root 6640     Apr  10 18:05 dev
drwxr-xr-x     136 root root  12288    Apr  10 14:13 etc
drwxr-xr-x     3    root root 4096     Apr  13 2016  home
…ignore below message ….

root@icpdas:/#
```

(2) -a：list all files including hidden files

```
root@icpdas:/# ls -a
.      boot   etc          lib         mnt    root   tmp
..     cdrom  home         lost+found  opt    run    srv    usr
bin    devinitrd.img media  proc        sbin   sys    var

root@icpdas:/#
```

3.1.2. Linux Command “cd”

cd directory: Change directory -> (like cd in DOS)

Parameter：

(1) **..**：move to the upper directory

```
root@icpdas:/tmp# cd ..
root@icpdas:/#
```

(2) ~：move back to the root directory

```
root@icpdas:/# cd ~
root@icpdas:~#
```

(3) /：divided sign

```
root@icpdas:~# cd /tmp
root@icpdas:/tmp#
```

3.1.3. Linux Command “mkdir”

mkdir: create the subdirectory -> (like md in DOS)

mkdir –parameter subdirectory

```
root@icpdas:/tmp# mkdir mytest
root@icpdas:/tmp# ls mytest
mytest
root@icpdas:/tmp#
```

3.1.4. Linux Command “rm”

rm: delete file or directory -> (like del or deltree in DOS)

rm -parameter file (or directory)

Parameter：

(1) i: it will show the warning message when deleting

```
root@icpdas:/tmp# rm -i test.log
rm: remove regular file 'test.log'? y
root@icpdas:/tmp#
```

(2) r: delete directory despite that it isn’t empty

```
root@icpdas:/tmp# rm -r mytest/
root@icpdas:/tmp#
```

(3) f: it will not show a warning message when deleting (Example: rm -f test.exe)

```
root@icpdas:/tmp# rm -f test.log
root@icpdas:/tmp#
```

3.1.5. Linux Command “cp”

cp: copy file -> (like copy in DOS)

cp –parameter source destination

```
root@icpdas:/tmp# cp /root/test.log /tmp/test.log
root@icpdas:/tmp# ls test.log
test.log
root@icpdas:/tmp#
```

3.1.6. Linux Command “mv”

mv: move or rename file or directory -> (like move or ren in DOS)

mv -parameter source file (or directory) destination file (or directory)

```
root@icpdas:/tmp# mv /root/test.log /tmp/mytest.log
root@icpdas:/tmp# ls mytest.log
mytest.log
root@icpdas:/tmp#
```

3.1.7. Linux Command “pwd”

pwd: show the current path

```
root@icpdas:/tmp# pwd
/tmp
root@icpdas:/tmp#
```

3.1.8. Linux Command “chmod”

chmod: change authority of file

chmod ??? file -> ??? means owner: group: all users

For example：

7 5 4 -> 111(read, write, execute) 101(read, write, execute) 100(read, write, execute)

The first number 7: **owner** can read and write and execute files

The second number 5: **group** can only read and execute files

The third number 4: **all users** can only read files

```
root@icpdas:/tmp# chmod 754 test.log
root@icpdas:/tmp# ls -al test.log
-rwxr-xr-- 1 root root 7 Apr 11 10:34 test.log
root@icpdas:/tmp#
```

3.1.9. Linux Command “uname”

uname: show the version of Linux kernel

```
root@icpdas:/tmp# uname -a
Linux icpdas 5.4.161-rt67 #12 SMP PREEMPT_RT Wed Feb 22 10:54:16 CST 2023 x86_64
x86_64 x86_64 GNU/Linux
root@icpdas:/tmp#
```

3.1.10. Linux Command “ps”

ps: show the procedures that execute now

```
root@icpdas:/tmp# ps -a
  PID TTY          STAT      TIME COMMAND
 2751 ttySX2      00:00:00 bash
30734 pts/0       00:00:00 ps
root@icpdas:/#
```

3.1.11. Linux Command “date”

date: show date and time

```
root@icpdas:/tmp# date
Fri 2 Feb 2024 04:39:11 PM CST
root@icpdas:/tmp#
```

3.1.12. Linux Command “netstat”

netstat: show the state of network

Parameter [-a]: list all states

```
root@icpdas:/tmp# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 icpdas:ssh              192.168.2.195:59561     ESTABLISHED
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
…ignore below message ….

root@icpdas:/#
```

3.1.13. Linux Command “ifconfig”

ifconfig: show the ip and network mask (like ipconfig in DOS)

```
root@icpdas:/tmp# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.186  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::290:fbff:fe74:dad3  prefixlen 64  scopeid 0x20<link>
        ether 00:90:fb:74:da:d3  txqueuelen 1000  (Ethernet)
        RX packets 14609  bytes 1430664 (1.4 MB)
        RX errors 0  dropped 1756  overruns 0  frame 0
        TX packets 702  bytes 104533 (104.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0x91180000-911fffff

eth1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:0d:e0:6e:30:1e  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0x91500000-9157ffff

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 171  bytes 14078 (14.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 171  bytes 14078 (14.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
root@icpdas:/tmp#
```

3.1.14. Linux Command “wget”

wget: get the file from the web link.

```
root@icpdas:# wget https://www.icpdas.com/en/download/file.php?num=19223
--2024-02-2 17:04:56--  https://www.icpdas.com/en/download/file.php?num=19223
Resolving www.icpdas.com (www.icpdas.com)... 50.87.226.148
Connecting to www.icpdas.com (www.icpdas.com)|50.87.226.148|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location:
https://www.icpdas.com/web/product/download/pac/linux/lx-9000/software/sdk/linpac_x86_sdk.tgz [following]
--2024-02-2 17:04:57--
https://www.icpdas.com/web/product/download/pac/linux/lx-9000/software/sdk/linpac_x86_sdk.tgz
Reusing existing connection to www.icpdas.com:443.
HTTP request sent, awaiting response... 200 OK
Length: 1177554 (1.1M) [application/x-tar]
Saving to: â€˜file.php?num=19223â€™

file.php?num=19223
100%[==========================================================================
=============>]      1.12M      499KB/s       in 2.3s

2024-02-2 17:05:00 (499 KB/s) - â€˜file.php?num=19223â€™ saved [1177554/1177554]

root@icpdas:~#
```

3.1.15. Linux Command “ping”

ping: check to see if the host in the network is alive

ping IPAddress (Example: ping 192.168.0.1)

```
root@icpdas:/tmp# ping www.google.com
PING www.google.com (142.251.43.4) 56(84) bytes of data.
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=1 ttl=57 time=7.70 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=2 ttl=57 time=7.76 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=3 ttl=57 time=8.13 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=4 ttl=57 time=7.85 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=5 ttl=57 time=7.63 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=6 ttl=57 time=8.03 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=7 ttl=57 time=7.63 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=8 ttl=57 time=7.63 ms
^C
--- www.google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7007ms
rtt min/avg/max/mdev = 7.625/7.793/8.132/0.183 ms
root@icpdas:/tmp#
```

3.1.16. Linux Command “clear”

clear: clear the screen

```
root@icpdas:/tmp# ping www.google.com
PING www.google.com (142.251.43.4) 56(84) bytes of data.
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=1 ttl=57 time=7.70 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=2 ttl=57 time=7.76 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=3 ttl=57 time=8.13 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=4 ttl=57 time=7.85 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=5 ttl=57 time=7.63 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=6 ttl=57 time=8.03 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=7 ttl=57 time=7.63 ms
64 bytes from tsa03s08-in-f4.1e100.net (142.251.43.4): icmp_seq=8 ttl=57 time=7.63 ms
^C
--- www.google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7007ms
rtt min/avg/max/mdev = 7.625/7.793/8.132/0.183 ms
root@icpdas:/tmp# clear
```

3.1.17. Linux Command “passwd”

passwd: change the password

passwd ??? －＞ ??? means the users that you want to change password

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

3.1.18. Linux Command “reboot”

reboot: reboot the LinPAC

```
root@icpdas:~# reboot
```

3.2. i-Talk Utility

The **i-Talk utility** can make the convenient for users to access the modules and hardware in the ALX-9x51.

Instruction	Function Description
getlist	List all module name in the ALX-9x51
setdo	Set digital output value to I9k module
setao	Set analog output value to I9k module
getdi	Get digital input value from I9k module
getai	Get analog input value from I9k module
setexdo	Set digital output value to I97k module
setexao	Set analog output value to I97k module
getexdi	Get digital input value from I97k module
getexai	Get analog input value from I97k module
setport	Set port value by offset to a module
getport	Get port value by offset from a module
setsend	Send string from LinPAC COM port
getreceive	Receive string from LinPAC COM port
getsendreceive	Send/Receive string from LinPAC COM port
read_sn	Get Hardware Serial Number of ALX-9x51
rotary_sw	Get Rotary Switch ID of ALX-9x51
led	Set all led indicator status
led_single	Set one led indicator status
OS_version	Get your LinPAC OS version

Below Table lists the demos that show how to use the I-talk utility. In the demo, the **I-9024** (AO Module) 、**I-9017** (AI Module) and **I-9054P** (DIO Module) are all used and they are plugged into the slots 1、2 and 3 of the LinPAC separately.

Instruction	Demo
getlist	**Command:** getlist → getlist **Description**: List all module names in the ALX-9x51 Series.
setdo	**Command:** setdo [slot] [data] → setdo 3 3 **Description**: Set i-9054P channel 1 and 2 on.
setao	**Command:** setao [slot] [channel] [data] → setao 3 0 2.2 **Description**: Set i-9024 channel 0 output 2.2V.
getdi	**Command:** getdi [slot] [type] → getdi 3 8 **Description**: Get the 8 bits DI value from i-9054P.
getai	**Command:** getai [slot] [channel] [gain] [mode] → getdi 3 0 0 0 **Description**: Get the AI value from i-9017.
setexdo	**Command:** (1) setexdo [slot] 1 [data] → setexdo 3 1 55 (2) setexdo [slot] [comport] [data] [baudrate] [address]

	→ setexdo 0 ttyS1 55 9600 2 **Description**: (1)Set the dec digital output value to the module at slot 3 at ttySA0. (2)Set the dec digital output value to the module at slot 0 at ttyS1.
setexao	**Command:** (1) setexao [slot] 1 [value] [channel] → setexao 3 1 6.7 5 (2)setexao [slot] [comport] [value] [channel] [baudrate] [address] → setexao 0 ttyS1 6.7 5 9600 2 **Description**: (1)Set channel 5 analog value 6.7 to the module at slot 3. (2)Set channel 5 analog value 6.7 to the module at ttyS1.
getexdi	**Command:** (1)getexdi [slot] 1 → getexdi 3 1 (2)getexdi [slot] [comport] [baudrate] [address] → getexdi 0 ttyS1 9600 2 **Description**: (1)Get the dec digital input value from the module at slot 3. (2)Get the dec digital input value from the module at ttyS1.
getexai	**Command:** (1)getexai [slot] 1 [channel] → getexai 3 1 5 (2)getexai [slot] [comport] [channel] [baudrate] [address] → getexai 0 ttyS1 5 9600 2 **Description**: (1)Get channel 5 analog value from the module at slot 3. (2)Get channel 5 analog value from the module at ttyS1.
read_sn	**Command:** read_sn → read_sn **Description**: Show the serial number.

rotary_sw	**Command:** rotary_sw → rotary_sw **Description**: Read rotary switch ID of ALX-9x51
led	**Command:** led → led **Description**: Print a menu to set all led indicator
led_single	**Command:** led_single [led] [status] → led_single 0 1 **Description**: Led: 0 -> Run status: 0 -> OFF 1 -> L1 1 -> ON 2 -> L2 Set “Run” led indicator status to “ON”
OS_version	**Command:** OS_version → OS_version **Description**: Read LinPAC OS version

3.3. ALX-9x51 SDK

ALX-9x51 SDK consists of the following major items.

- LinPAC SDK library files
- LinPAC SDK include files
- Demo files

From https://www.icpdas.com/en/download/file.php?num=19223, you can download the latest version of ALX-9x51 SDK. And then follows the below steps in order to get the development toolkit which has been provided by ICP DAS for the easy application of the ALX-9x51 embedded controller platform.

(1)User can connect to ALX-9x51 through communication port (**Console**, **LAN1**, **LAN2)** by using **"putty"** software (refer to "2.5 Console Port Connection ").

(2)After connecting to ALX-9x51, user could type "wget" command to get the latest version of ALX-9x51 SDK.

Tips & Warnings

Please check the network can connect to the ICP DAS official website.

ALX-9000 series

Related Resources

▶ I-9K/I-97K Software Support of LinPAC

FILE NAME	VERSION	FILE DATE	SIZE	NOTE	
LinPAC x86 SDK	0.9.0	2023-03-05	1.1 MB	For ALX-9000, LX-8000 and LX-9000 series	

(2) To type "**tar zxf linpac_x86_sdk.tgz**" to decompress tar file and type "make" to compile demo code.

```
root@icpdas:~# tar zxf linpac_x86_sdk.tgz
root@icpdas:~# ls linpac_x86_sdk
linpac_x86_sdk
root@icpdas:~# cd linpac_x86_sdk/
root@icpdas:~/linpac_x86_sdk# make
```

Once user decompresses the SDK file, user can find the files for the library and demo in the following paths.

The libPAC_x86.a path is "**LinPAC_X86_SDK/lib**".

The include files path is "**LinPAC_X86_SDK/include**".

The ALX-9x51 demo path is "**LinPAC_X86_SDK/examples/lx-series**".

4. Your First ALX-9x51 Program

This chapter provides a guided tour that describes the steps needed to set-up a development environment, download, install, configure for user programming with the ALX-9x51. The default ALX-9x51 support below development tools:

Default Development Tools

Linux OS is a mature embedded operating system which supports rapid development. Standard development tools are list as follows which are highly integrated, with comprehensive support for developing applications of ALX-9x51.

- Scripts Language (shell script, perl, python and php)
- Standard C language and GNU gcc compiler

ALX-9x51 SDKs

The x86 LinPAC SDK is a Software Development Kit (SDK) that contains C header files, C libraries.

The ALX-9x51 SDK is classified by development tools that can be downloaded from the web link

"https://www.icpdas.com/en/download/file.php?num=19223"

4.1. Your First ALX-9x51 Program with C Language

In this section, we will introduce how to compile the helloworld.c to helloworld and executes this file on the ALX-9x51 Series. In this example, no ICP DAS modules are used. To create a demo program with C that includes the following main steps:

1. Find demo “helloworld.c” in SDK
2. Compile and Execute the demo on ALX-9x51
3. Execute the application on ALX-9x51 at boot time

All main steps will be described in the following subsection.

4.1.1. Find Demo “helloworld.c” in SDK

The x86 LinPAC SDK had added the demo “helloworld.c”. User can follow below steps to find the demo in SDK.

1. Download SDK

Please get SDK with link “https://www.icpdas.com/en/download/file.php?num=19223”

2. Find the demo in SDK

Using command “cd” move to demo’s directory.

3.Use “vim” command to modify/create demo “helloworld.c” in SDK

```
root@icpdas:~#cd LinPAC_SDK/examples/lx-8k_9k/common
root@icpdas:/tmp/LinPAC_X86_SDK/examples/lx-series/common# ls helloworld*
helloworld    helloworld.c
root@icpdas:/tmp/LinPAC_X86_SDK/examples/lx-series/common#vim helloworld.c
```

4.1.2. Compile/Execute Demo “helloworld”

User can refer to below steps to compile and execute demo “helloworld”.

1. Add script to SDK Makefile

User can refer to the SDK Makefile (LinPAC_X86_SDK/examples/lx-8k_9k/common/Makefile) that we had added the script for demo “helloworld.c”.

2. Type “make helloworld” command to compile demo “helloworld.c”

3. Execute binary file “helloworld”

```
root@icpdas:/tmp/LinPAC_X86_SDK/examples/lx-8k_9k/common# make helloworld
gcc -I. -I../../../include      -c -o helloworld.o helloworld.c
gcc -I. -I../../../include -o helloworld helloworld.o ../../../lib/libPAC_x86_64.a -lm
rm -f helloworld.o
root@icpdas:/tmp/LinPAC_X86_SDK/examples/lx-8k_9k/common# ./helloworld
Hi ~ Welcome to x86 LinPAC
root@icpdas:/tmp/LinPAC_X86_SDK/examples/lx-8k_9k/common#
```

4.1.3. Execute Demo at Boot Time

User can refer to below steps to auto-execute demo “helloworld” at boot time in ALX-9000.

1. Create script file in “/etc/init.d”

User can use “vim” command to create the script file in “/etc/init.d” and add below script language to the file.

```
root@icpdas:~# vim /etc/init.d/helloworld
```

```
#! /bin/bash

/root/LinPAC_SDK/examples/lx-8k_9k/common/helloworld >
/tmp/test.log
```

Please refer to your own path for executing the helloworld demo.

2. Create service file in “/lib/systemd/system”.

```
root@icpdas:~# vim /lib/systemd/system/helloworld.service
```

```
[Unit]
Description=Icpdas init routine
After=syslog.target

[Service]
ExecStart=/bin/sh /etc/init.d/helloworld
Type=oneshot
```

3. Create soft link to link /lib/systemd/system/helloworld.service

```
root@icpdas:~# ln -s /lib/systemd/system/helloworld.service
helloworld.service
```

4. After these setting , the ALX-9000 will execute binary “helloworld” at boot time

```
root@icpdas:~# cat /tmp/test.log
Hi ~ Welcome to LinPAC SDK
root@icpdas:~#
```

5. Application for ALX-9x51

In this chapter, ICP DAS provides extra module supported and instructions to enhance LX-8000/9000 functionality and affinity.

5.1. Advanced Package Tool

“apt-get” command is a management system for software packages on the Linux OS of LX-9000.

- To install a package run the following commands:

```
apt-get update
apt-get install <package>
```

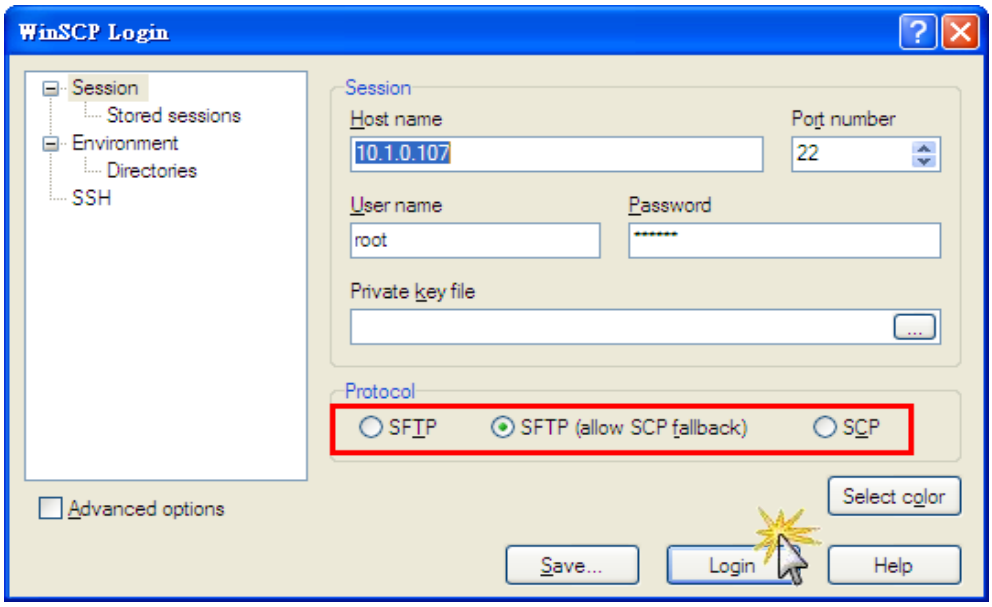
- To search available package run the following commands:

```
apt-cache search <package name>
```

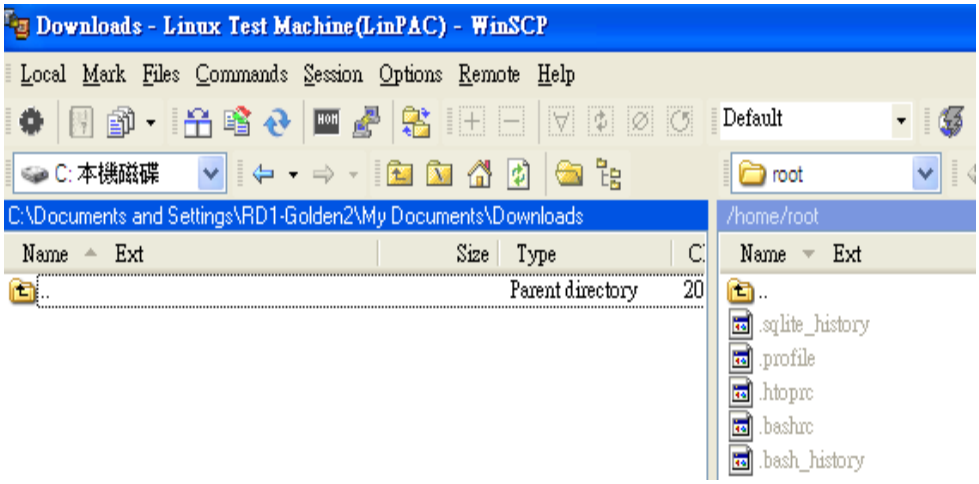
5.2. SFTP (secure file transfer program)

The ALX-9x51 series had supported SFTP(or SCP), user can transfer the file from Windows(or Linux). For examples, using Windows Program “WinSCP” to access the device over network, please follow below steps:

1. Choosing the “SFTP” or “SCP” protocol and type IP address, default ID (root) and password (icpdas) to login

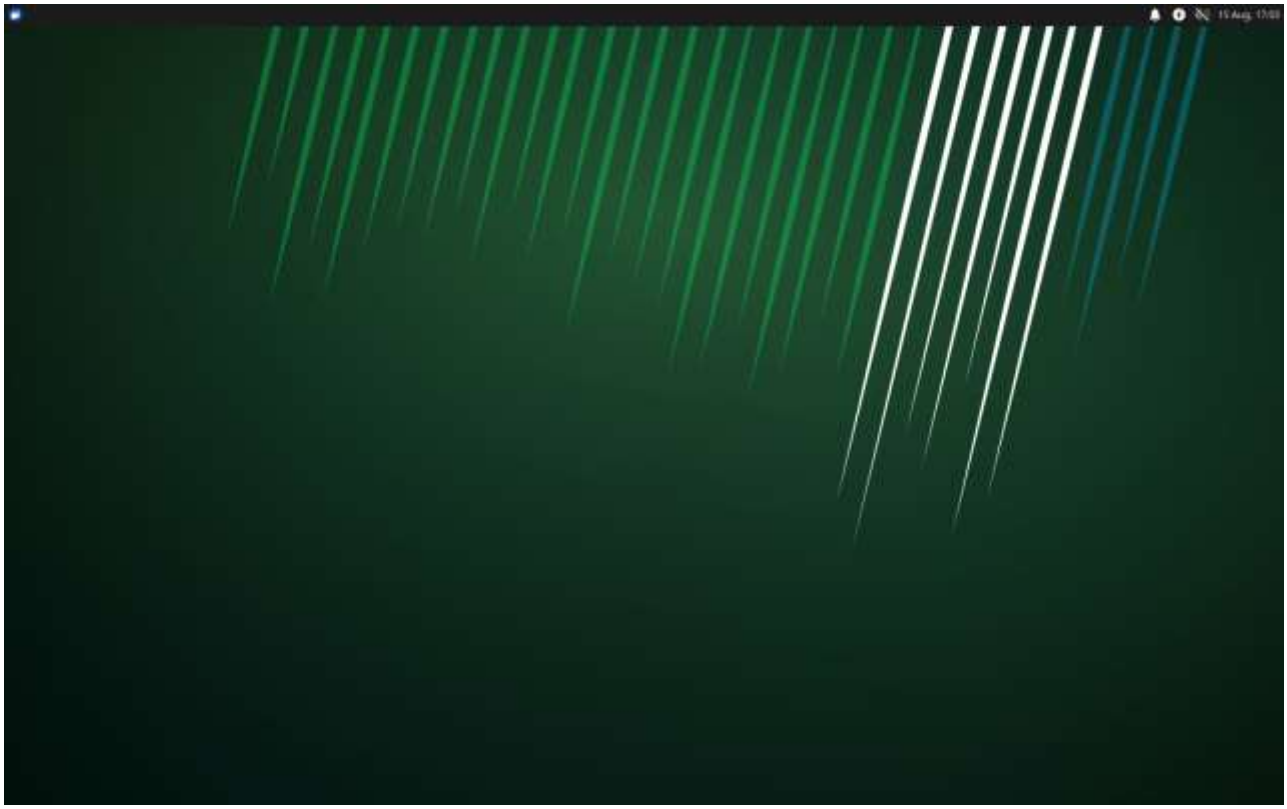


2. Drag and drop file



5.3. XFCE

XFCE is a lightweight desktop environment for UNIX-like operating systems. It aims to be fast and low on system resources, while still being visually appealing and user friendly. The ALX-9x51 series support the XFCE package, after user type “root” and password “icpdas” to login, the local terminal would execute the XFCE Desktop.



ALX-9x51 XFCE Desktop screen shot

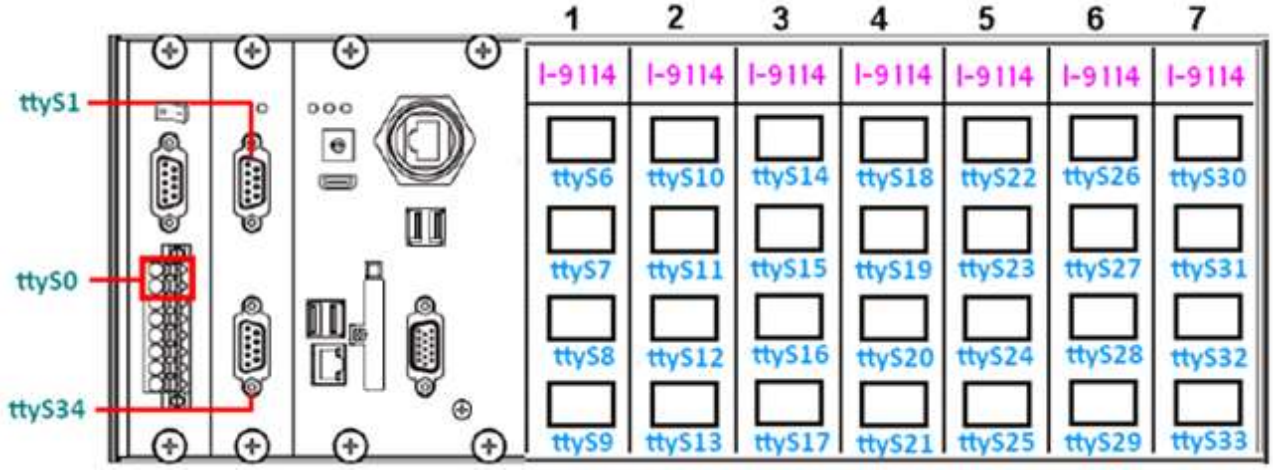
6. Additional Support

In this chapter, ICP DAS provides extra module supported and instructions to enhance ALX-9x51 functionality and affinity.

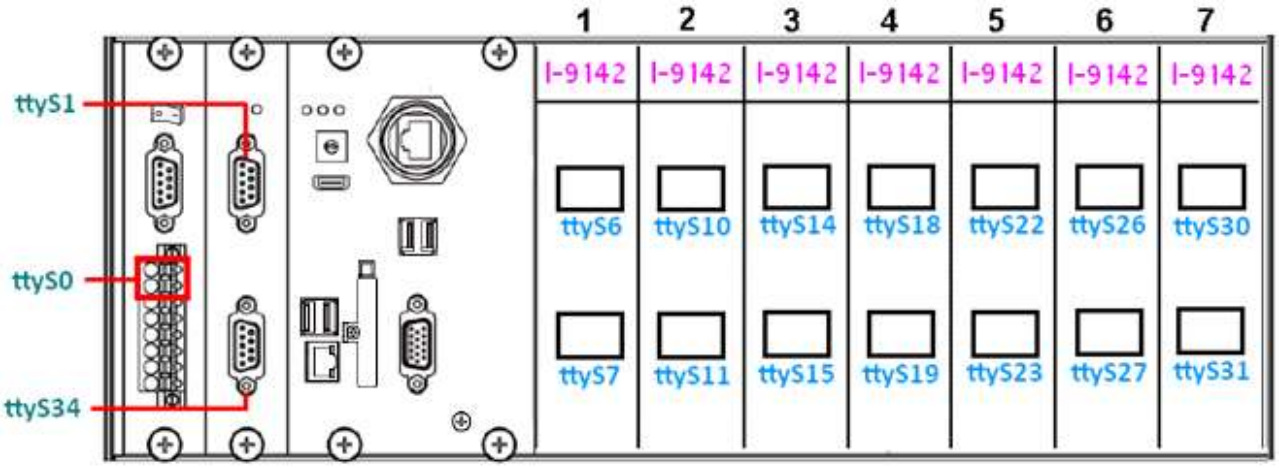
6.1. N-Port Modules Support

N-port communication modules provide two or four serial ports and can be inserted into the slot of an ALX-9x51 embedded controller. In this way, additional serial ports can be used on the ALX-9x51 embedded controller. The ALX-9x51 embedded controller is a multi-tasking unit, meaning that all the serial ports can be controlled simultaneously. The number of each serial port on the N-Port Serial modules (two or four serial ports) are presented in below figures:

- ALX-9x51 N-Port Modules support (I-9114W, I-9144iW, i-9142W, etc.)



ALX-9x51 and Serial Module (4-Ports)



ALX-9x51 and Serial Module (2-Ports)

7. ALX-9x51 Internet Service

The internet service provided by ICP DAS will be satisfied and it includes Technical Support, OS_Image, LinPAC_SDK and User's Manual Download etc. Users can refer to the following web site to get more information:

1. ICP DAS Web Site :

http://www.icpdas.com/

2. ALX-9x51 Product Page:

https://www.icpdas.com/en/product/guide+PACs+Linux__PACs+ALX-9000

3. E-mail for Technical Support: service@icpdas.com

Appendix

A. I-9K Modules and I-97K Modules

This chapter provides a brief overview of the different between I-9K series modules and I-97K series modules.

There are two types of I/O modules provided for supporting ALX-9x51. One is high communication speed I-9K series modules with parallel interface; the other is I-97K series modules with serial interface. The differences between the two series are listed as follows:

The differences between the I-9K and I-97K series I/O modules are as follows.

Item	I-9K Series	I-97K Series
Communication Interface	Parallel Bus	Serial Bus
Protocol	-	DCON
Communication Speed	Fast	Slow
DI with latched function	-	Y
DI with counter input	-	Y (100 Hz)
Power on value	-	Y
Safe Value	-	Y
Programmable slew-rate for AO module	-	Y

B. e-9K Modules

e-9K series modules are provided for combining a variety of I/O functions within the AXP-9000 and ALX-9000 programmable automation controllers (PAC). e-9k series is based on a high-speed bus interface (e-Bus) with speeds up to 2GB/s (e-Bus x4) and 500MB/s(e-Bus x1), and also supports DMA (Direct memory access) data transfer without the need for a central Processing unit intervention. e-Bus new generation high-speed bus interface.

ICP DAS fully supports the new generation of high-speed bus interface (e-Bus) in the new generation flagship controller (AXP/ALX9000). In addition to the maximum speed of up to 2GB/s (e-Bus x4) and 500MB/s(e-Bus x1), it also supports DMA (Direct memory access) data transmission without the need for the central processing unit to intervene in processing. Under the burden of the processor at the same level, DMA is a fast data transfer method, which can provide a large amount of data transmission and high-speed data comparison of the expansion module, and improve the working efficiency of the controller.

ICP DAS utilizes the powerful performance of the e-Bus bus interface to fully develop a new generation of e-9K I/O expansion modules, and fully demonstrate the performance of e-Bus on the new generation flagship controller (AXP/ALX9000) to provide customers with the high-cost performance ratio.

The differences between the two series are listed as follows:

Model	e-9K Series	I-9K Series	I-97K Series
Communication Interface	e-Bus	I-9K Bus (Parallel)	I-97K Bus (Serial)
Protocol	-	-	DCON
Communication Speed	2GB/s (e-Bus x4) 500MB/s(e-Bus x1)	200 ~ 500 KB/s	115 kbit/s
DI with Latched Function		-	Y
DI with Counter Input		-	Y (100 Hz)
Power on Value		Y	Y
Safe Value		Y	Y
Programmable Slew-Rate for AO Module		-	Y

C. Revision History

This chapter provides revision history information to this document.

The table below shows the revision history.

Revision	Date	Description
V1.0.1	Dec. 2025	Restoring the OS from the CFast card has been changed to restoring the OS from a USB hard drive.
V1.0.0	Sep. 2025	Initial issue