

六叶树产品教程-aarch64 下 Docker 使用教程

类别	内容
关键词	Docker 使用教程
摘要	本教程讲解如何在 Docker 环境下使用六叶树 USBCANFD 产品

相关环境介绍:

类别	内容
Docker 宿主系统	Ubuntu22.04
Docker 版本	27.4.1

文档记录

版本	日期	· 说明
V1.00	2025.01	创建文档





六叶树产品教程-aarch64 下 Docker 使用教程
1、Docker 教程
1.1 Docker 安装
1.2 Docker 镜像的导入
1.3 Docker 容器创建
1.4 Docker 容器启动
1.5 Docker 容器停止
1.6 Docker 进入容器
2、linux 通用用例使用说明
第一步,启动并进入容器
第二步,工程代码下载
第三步,代码编译与运行
常见问题:1
3、CanAssistant Docker 镜像使用说明1
第一步 , 导入镜像1
第二步,创建并启动容器
3 常用链接1
4 联系方式1

1、Docker 教程

1.1 Docker 安装

1.1.1 aarch64 架构

演示安装环境:



项目	版本
CPU 架构	Aarch64
操作系统	Ubuntu22.04

orangepi@orangepi4-lts:~\$ lsb_release -a No LSB modules are available. Distributor ID: Ubuntu Description: Ubuntu 22.04.5 LTS Release: 22.04 Codename: jammy orangepi@orangepi4-lts:~\$

注意:所有指令使用具有 root 权限的账户

1.更新软件包索引

apt-get update apt-get -y upgrade

2. 允许 APT 使用 HTTPS

apt-get install apt-transport-https ca-certificates curl software-properties-common



3.添加 Docker 官方 GPG 密钥

curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -

root@orangepi4-lts:/home/orangepi# root@orangepi4-lts:/home/orangepi# root@orangepi4-lts:/home/orangepi# root@orangepi4-lts:/home/orangepi# root@orangepi4-lts:/home/orangepi# curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -Warning: apt-key is_deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)). OK coot@orangepi4-lts:/home/orangepi#

4.添加 Docker 软件源

add-apt-repository "deb [arch=arm64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu \$(lsb_release -cs) stable"

六叶树教育科技有限公司 © 版权所有



orangepi4-lts:/home/orangepi# add-apt-repository "deb [arch=arm64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu \$ stable statte depository: 'deb [arch=arm64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy stable' nechive for codename: jammy components: stable fore info: http://mirrors.aliyun.com/docker-ce/linux/ubuntu

5.再次更新软件包索引

apt-get update

6.安装 Docker CE (社区版)

apt-get install -y docker-ce docker-ce-cli containerd.io



源配置:

touch /etc/docker/daemon.json vim /etc/docker/daemon.json

{

"registry-mirrors": ["https://registry.docker-cn.com", "https://docker.mirrors.ustc.edu.cn", "https://hub-mirror.c.163.com", "https://mirror.baidubce.com", "https://ccr.ccs.tencentyun.com"]

}



7.验证 Docker 是否安装成功并运行

systemctl status docker

root@orangepi4-lts:/home/orangepi# systemctl status docker
o docker.service - Docker Application Container Engine
Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
Active: active (running) since Wed 2025-01-08 06:10:39 UTC; 1min 15s ago
TriggeredBy: 🗴 docker.socket 🔨
Docs: https://docs.docker.com
Main PID: 4277 (dockerd)
Tasks: 12
Memory: 20.6M
CPU: 1.304s
CGroup: /system.slice/docker.service
4277 /usr/bin/dockerd -H fd://containerd=/run/containerd/containerd.sock
Jan 08 06:10:37 orangepi4-lts systemd[1]: Starting Docker Application Container Engine
Jan 08 06:10:38 orangepi4-lts dockerd[4277]: time="2025-01-08T06:10:38.010600890Z" level=info msg="Starting up"
Jan 08 06:10:38 orangepi4-lts dockerd[4277]: time="2025-01-08T06:10:38.013043795Z" level=info msg="OTEL tracing is not configured, using n
Jan 08 06:10:38 orangepi4-lts dockerd[4277]: time="2025-01-08T06:10:38.268559407Z" level=info msg="Loading containers: start."
Jan 08 06:10:39 orangepi4-lts dockerd[4277]: time="2025-01-08T06:10:39.207715852Z" level=info msg="Loading containers: done."
Jan 08 06:10:39 orangepi4-lts dockerd[4277]: time="2025-01-08T06:10:39.2555429592" level=info msg="Docker daemon" commit=c710b88 container
Jan 08 06:10:39 grangepi4-lts dockerd[4277]: time="2025-01-08T06:10:39.255974026Z" level=infg msg="Daemon has completed initialization"
Jan 08 06:10:39 grangepi4-lts dockerd[4277]: time="2025-01-08T06:10:39.411993613Z" level=info msg="API listen on /run/docker.sock"
Jan 08 06:10:39 grangepi4-lts systemd[11: Started Docker Application Container Engine.
lines 1-21/21 (END)

8.查看版本

docker --version



1.2 Docker 镜像的导入

方式 1:通过 Dockerfile 文件

通过使用六叶树提供的 Dockerfile 文件导入到 Docker 环境。

进入到 Dockerfile 文件所在的目录执行命令: root@orangepi4-lts:/home/orangepi/docker# ls Dockerfile root@orangepi4-lts:/home/orangepi/docker#

语法:

docker build -t "镜像名称:版本" ./

镜像名称和版本是自定义的。

示例:

docker build -t lys.ubuntu.22.04.usbcan.demo:20250106 ./



Sending build context to Docker daemon 2.048kB

导入完成



COLLECT_LTO_WRAPPER=/usr/lib/gcc/aarch64-linux-gnu/ll/lto-wrapper Targetaarch64-linux-gnu
<pre>rarge: aarCrow-intoXynu Configured with:fxrc/configure -vwith-pkgversion='Ubuntu 11.4.0-lubuntu1-22.04'with-bugurl=file:///usr/share/doc/gcc-11/README.Bug configured with:fxrc/configure -vwith-pkgversion='Ubuntu 11.4.0-lubuntu1-22.04'with-bugurl=file:///usr/share/doc/gcc-11/README.Bug configured with:fxrc/configure -vwith-pkgversion='Ubuntu 11.4.0-luburtu1-22.04'with-bugurl=file:///usr/share/doc/gcc-11/README.Bug configured buf-luburtu1-included_gettextenable-linker-build-idlibexecdir=/usr/libwithout-included_gettextenable-threads=posixlib bdir=/usr/libenable-nlsenable-bootstrapenable-clocale=gnuenable-libstdcxx-debugenable-libstdcxx-time=yeswith-default-libs tdcxx-abi=newenable-libends-chocking=releasewith-target-system -Zlib=autoenable-objc-gc=autoenable-multiarchenable-fix-cortex-a 53-843419disable-wrrorenable-chocking=releasebuild=aarch64-linux-gnuhost=aarch64-linux-gnutarget=aarch64-linux-gnuwith-b uild-config=bootstrap-lto-leanenable-link-serialization=2</pre>
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version II.4.0 (Ubuntu II.4.0-IubuntuI~22.04)
root@orangep14.tts:/home/orangep1/docker# ping www.baidu.com
PINg www.a.sniren.com (153.3.238.102) 56(84) bytes of data.
64 bytes from 153.3.238.102 (153.3.238.102): 1cmp_seq=1 ttl=52 fime=22.0 ms
64 bytes from 153.3.238.102 (153.3.238.102): 1cmp_seq=2 ttl=52 time=21.7 ms
D4 Dytes from 153.3.238.102 (153.3.238.102): 1Cmp_seq=3 ttt=52 t1me=22.0 ms ^C
www.a.shifen.com ping statistics
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 21.720/21.889/21.994/0.120 ms
root@orangepi4-lts:/home/orangepi/docker# ping www.liuyeshu.cn 🚊成
PING bjdiy05.qidc.cn (180.76.249.219) 56(84) bytes of data.
bjd1y05.qidc.cn ping statistics
/ packets transmitted, 0 received, 100% packet loss, time 6130ms
查看当前系统里的 Docker 镜像:

命令:

docker images

aborter mages				
root@orangepi4-lt:	s:/home/oran	ngepi/docker#		
root@orangepi4-lt	s:/home/oran	ngepi/docker#		
root@orangepi4-lt	s:/home/oran	ngepi/docker# d	ocker images	
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
lys.ubuntu.22.04	20250108	ca4993eff2fb	12 minutes ago	388MB
ubuntu 🔻	22.04	37f74891464b	18 months ago	69.2MB
root@orangepi4-lt	s:/home/orai	ngepi/docker#		

方式 2:通过 Docker 镜像文件

通过使用六叶树提供的 Docker 镜像文件导入到 Docker 环境。

```
root@orangepi4-lts:/home/orangepi/docker#
root@orangepi4-lts:/home/orangepi/docker#
root@orangepi4-lts:/home/orangepi/docker#
root@orangepi4-lts:/home/orangepi/docker#
Dockerfile lys.aarch64.ubuntu.22.04.usbcan.demo.tar
root@orangepi4-lts:/home/orangepi/docker#
```

进入到 Docker 镜像文件所在的目录执行命令:

语法:

docker load -i "六叶树镜像文件名称"

示例:

dockerdocker load -i lys.aarch64.ubuntu.22.04.usbcan.demo.tar

root@orangep14-lts:/home/orangep1/docker#	
root@orangepi4-lts:/home/orangepi/docker# docker load -i lys.aarch64.ubuntu.22.04	.usbcan.demo.tar
e466a070b652: Loading layer [2.048kB/2.048kB
33550c0f3ff9: Loading layer [====================================	109.5MB/109.5MB
af724512d696: Loading layer [>]	3.353MB/3.353MB
4fc33028593e: Loading layer [====================================	5.12kB/5.12kB
c40fe31e12a5: Loading layer [====================================	666.6kB/666.6kB
c58173773a02: Loading layer [>]	210.5MB/210.5MB
_oaded image: lys.ubuntu.22.04:20250108	
root@orangepi4-lts:/home/orangepi/docker#	

查看当前系统里的 Docker 镜像:



命令:					
docker images					
root@orangepi4-lt	s:/home/oran	ngepi/docker# do	ocker images		
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE	
lys.ubuntu.22.04	20250108	ca4993eff2fb	33 minutes ago	388MB	
ubuntu	22.04 🦘	37f74891464b	18 months ago	69.2MB	
root@orangepi4-lt	s:/home/oran	ngepi/docker# 📘	가는 것 같아요. 가슴에 가슴을 가셨다. 것으라 있다. (국가가) 		

1.3 Docker 容器创建

1.创建容器:

docker create -it --device=/dev/bus/usb:/dev/bus/usb --privileged --name lys-container lys.ubuntu.22.04:20250108

root@orangepi4-lts:/home/orangepi# docker create -<u>it --device=/dev/bus/usb:/dev/bus/usb --privileged</u> --name lys-container lys.ubuntu.22.04: 20250108 1e57c0b.c6689de2528d5e92f60bddc592da038c400ee5a97bfdad602beec6da5

2.查看到上面创建的容器:

命令:

docker ps -a

root@orangepi4	-lts:/home/orangepi#	ns -3				
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
le57c0bc0689	lys.ubuntu.22.04:20250108	"/bin/bash"	About a minute ago	Created		lys-container
root@orangepi4	-lts:/home/orangepi# 🗧					

注意:容器创建指令一定要加参数:"--device=/dev/bus/usb:/dev/bus/usb --privileged",否则程 序访问六叶树设备会报错。

1.4 Docker 容器启动

命令:

docker start lys-container

uocker start	Tys container					
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
1e57c0bc0689	lys.ubuntu.22.04:20250108	"/bin/bash"	About a minute ago	Created		lys-container
root@orangepi4	-lts:/home/orangepi# docker	start lys-con	tainer			
lys-container						
root@orangep14	-lts:/home/orangep1#					

查看容器状态:

命令:

docker ps -a

uberter p3 u									
root@orangepi4	-lts:/home/orangepi#								
root@orangepi4	root@orangepi4-lts:/home/orangepi#								
root@orangepi4	-lts:/home/orangepi# docker	ps -a							
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES			
1e57c0bc0689	lys.ubuntu.22.04:20250108	"/bin/bash"	2 minutes ago	Up 41 seconds	口白井	lys-container			
root@orangepi4	-lts:/home/orangepi#								



Docker 使用教程

1.5 Docker 容器停止

命令: docker stop lys-container

1.6 Docker 进入容器

命令: docker exec -it lys-container /bin/bash

root@orangepi4-lts:/home/orangepi#
root@orangepi4-lts:/home/orangepi#
root@orangepi4-lts:/home/orangepi# docker exec -it lys-container /bin/bash
root@le57c0bc0689:/#

2、linux 通用用例使用说明

六叶树提供的 Dockerfile 和镜像里是不包含通用用例代码的,只是构建后了用例编译以及运行所依赖的环境,代码的下载和编译、运行需要用户自己操作。 以下简要介绍 aarch64 平台下 linux 通用用例工程的使用。

第一步,启动并进入容器

见上面的 Docker 教程。

第二步,工程代码下载

下载工程代码到 Docker 容器。 六叶树 USBCAN 适配器二次开发用例下载地址: <u>http://www.liuyeshu.cn/?page_id=440</u>

IysDemo-arm-linux-gnueabi-gcc.7.4.1		立即下载
IysDemo-arm-linux-gnueabihf-gcc.7.4.1		立即下载
iysDemo-arm-none-linux-gnueabihf-gcc.11.2.1		立即下载
JysDemo-arm-openwrt-linux-gcc.4.8.3.arm_cortex-a7	我们这里演示的平台的orangepi4,是aarch64	立即下载
lysDemaarch64)linux-gnu-gcc.7.4.1		立即下载
IysDemo-aarch64-none-linux-gnu-gcc.11.2.1		立即下载

获取到下载地址:

1

ARM:



回到容器通过 wget 进行下载



下载下来是 zip 压缩包格式。



Zip 解压:

unzip lysDemo-aarch64-linux-gnu-gcc.7.4.1.20240615.tar.zip





gz 解压:

tar -xvf lysDemo-aarch64-linux-gnu-gcc.7.4.1.20240615.tar.gz

root@le5/c0bc0689:/home/lys# tar	-xvf lysDemo-aarch64-linux-gnu-gcc./.4.1.20240615.tar.gz
lysDemo/canframe.h	
lysDemo/config.h	
lysDemo/ControlCAN.h	
lysDemo/libControlCAN.a	
lysDemo/libControlCAN.so	
lysDemo/libusb-1.0.a	
lysDemo/libusb-1.0.la	
lysDemo/libusb-1.0.so	
lysDemo/libusb-1.0.so.0	
lysDemo/libusb-1.0.so.0.1.0	
lysDemo/libusb.h	
lysDemo/lyscan-test	
lvsDemo/lvscan-test.c	
lysDemo/lysfd-test	
lysDemo/lysfd-test.c	
lvsDemo/Makefile	
lvsDemo/tvpedef.h	

进入到工程目录:

cd lysDemo

cu iysbell	110							
root@le57c0bc	0689:/home/l	ys# cd lysDemo						
root@le57c0bc	0689:/home/l	ys/lysDemo# ls						
ControlCAN.h	canframe.h	libControlCAN.a	libusb-1.0.a	libusb-1.0.so	libusb-1.0.so.0.1.0	lyscan-test	lysfd-test	typedef.h
Makefile	config.h	libControlCAN.so	libusb-1.0.la	libusb-1.0.so.0	libusb.h	lyscan-test.c	lysfd-test.c	
root@le57c0bc	0689:/home/l	ys/lysDemo#						

编译:

make

root@le57c0bc	0689:/home/l	ys/lysDemo# make	4	05240				
rm -vf lyscan	-test lysfd-	test	」 直接m	nake				
removed 'lysc	an-test'		EIXI	Tarco				
removed 'lysf	d-test'							
gcc -o lyscan	-test lyscan	-test.c -L ./ -pt	hread -lrt -lCo	ontrolCAN -lusb-1.	Θ	4	1-1-1-	
gcc -o lysfd-	test lysfd-t	est.c -L ./ -pthr	ead -lrt -lCont	rolCAN -lusb-1.0		生成的可执	们程序	
root@le57c0bc	0689:/home/l	ys/lysDemo# ls						
ControlCAN.h	canframe.h	libControlCAN.a	libusb-1.0.a	libusb-1.0.so	libusb-1.0.so.0.1.0	lyscan-test	lysfd-test	typedef.h
Makefile	config.h	libControlCAN.so	libusb-1.0.la	libusb-1.0.so.0	libusb.h	lyscan-test.c	lystd-test.c	
root@le57c0bc	0689:/home/l	ys/lysDemo#						

运行:

export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:./ ./lyscan-test

root@le57c0bc0689:/home/lys/lysDemo# export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:./ root@le57c0bc0689:/home/lys/lysDemo# ./lyscan-test 设置CANO波特率成功 DEMO功能:CAN1波特率1M,接收到什么数据就回发什么数据。 启动成功



常见问题:

1.此设备不存在

root@le57c0bc0689:/home/lys/lysDemo# ./lyscan-test 此设备不存在 root@le57c0bc0689:/home/lys/lysDemo# 🚪

原因:六叶树 USBCANFD 设备没有接到主机或没有接到容器

可以通过 lsusb 指令查看 USB 设备已连接:

命令:

lsusb

rootale57c0bc0689:/home/lvs/lvsDemo# lsusb
Bus 007 Device 001: ID 1d6b:0002 Linux 5.18.5-rk3399 ehci hcd EHCI Host Controller
Bus 002 Device 001: <u>ID 1d6b:0003 Linux 5.18.5-rk33</u> 99 xhci-hcd xHCI Host Controller
Bus 004 Device 003: ID 0471:1200 Co.Ltd. X USBCAN
Bus 004 Device 001: ID 1d6b:0001 Linux,5-18-2-rk3399/ohci_bcd_6eneric_Platform_0HCi_montroller
Bus 006 Device 001: ID 1d6b:0003 Lind之东东东地内公约如外在东东省小公省从在外外的安全公司行行的任何可留占加久
Bus 008 Device 001: ID 1d6b:0001 Linux 5.18.5-rk3399 ohci_hcd Generic Platform OHCI controller
Bus 001 Device 001: ID 1d6b:0002 Linux 5.18.5-rk3399 xhci-hcd xHCI Host Controller
Bus 003 Device 001: ID 1d6b:0002 Linux 5.18.5-rk3399 ehci_hcd EHCI Host Controller
Bus 005 Device 001: ID 1d6b:0002 Linux 5.18.5-rk3399 xhci-hcd xHCI Host Controller
root@le57c0bc0689:/home/lys/lysDemo#

六叶树教育科技有限公司 © 版权所有



3、CanAssistant Docker 镜像使用说明



见上面的 Docker 教程。
oot@orangepi4-lts:/home/orangepi# oot@orangepi4-lts:/home/orangepi# oot@orangepi4-lts:/home/orangepi# oot@orangepi4-lts:/home/orangepi# oot@orangepi4-lts:/home/orangepi# oot@orangepi4-lts:/home/orangepi# ls oot@orangepi4-lts:/home/orangepi# ls oot@orangepi4-lts:/home/orangepi# docker load -i lys.aarch64.ubuntu.22.04.canassistant.tar oot@orangepi4-lts:/home/orangepi# docker load -i lys.aarch64.ubuntu.22.04.canassistant.tar
ooded image: lys.aarch64.ubuntu.22.04.canassistant:20250109 oot@orangepi4-lts:/home/orangepi#

导入成功:

docker images

root@orangepi4-lts:/home/orangepi# root@orangepi4-lts:/home/orangepi# docl	ker images	discussion office	interest and	 Calebrative at r
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
lys.aarch64.ubuntu.22.04.canassistant	20250109	7e667ad2c4de	25 minutes ago	670MB
Lys.ubuntu.22.04	20250108	ca4993eff2fb	18 hours ago	388MB
ubuntu	22.04	37f74891464b	18 months ago	69.2MB
root@orangepi4-lts:/home/orangepi#				

第二步,创建并启动容器

方式 1:通过 VNC 软件连接

要求其他电脑安装 VNC 软件,宿主机可以是不带界面的操作系统,容器里的运行的软件将通过 VNC 软件显示,宿主机和安装 VNC 的主机必须在同一个网络里。

1.安装 VNC 软件

VNC 下载地址: https://www.123684.com/s/elmmTd-QuPX



2.查看宿主机 IP 地址

命令:

Ifconfig -a

docker0: flags=4163 <up,broadcast,running,multicast> mtu 1500</up,broadcast,running,multicast>
inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
inet6 fe80::42:21ff:fe4a:9bbf prefixlen 64 scopeid 0x20 <link/>
ether 02:42:21:4a:9b:bf txqueuelen 0 (Ethernet)
BX nackets 25127 bytes 1398577 (1.3 MB)
BX errors of dropped of overrups of frame of
TV process 42141 bytes 10027661 (100.9 MP)
TX packets 42141 bytes 1096/0041 (109.6 Mb)
TX errors of dropped o overruns of carrier of cottisions of
eth0: flags=4163 <up, ad="" broadcast,="" intuffsorangepi4()="" p,<br="" runing,="" trast="">inet 192.168.1.103 netmack 255,255,255,0 breadcast 192.168.1.255 inet6 fe80::5ce4:65bf:8ca/b7b5 prefixion 64 necopeid 0x20<link/> ether 46:c8:1c:04:5a:84 txqueuelen 1000 (Ethernet) RX packets 154700 bytes 117000882 (117.0 MB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 401680 bytes 544090193 (544.0 MB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 device interrupt 31</up,>

3.启动容器

命令:



docker run -it --rm -e DISPLAY=\$DISPLAY \ -v /tmp/.X11-unix:/tmp/.X11-unix \ --device=/dev/bus/usb:/dev/bus/usb \ -p 5900:5900 \ --privileged lys.aarch64.ubuntu.22.04.canassistant:20250109 \ /home/lys/start.sh vnc root@orangepi4-lts:/home/orangepi# docker run -it --rm -e DISPLAY=\$DISPLAY \
-v /tmp/.Xll-unix:/tmp/.Xll-unix \
--device=/dev/bus/usb:/dev/bus/usb \ -p 5900:5900 \ --privileged lys.aarch64.ubuntu.22.04.canassistant:20250109 \ /home/lys/start.sh vnc lys CanAssistant QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root' "log/." "log/.." "log/dd.log" QVncServer created on port 5900 步通讨VNC连接 自动成功

运行 VNC 软件:



RealVNC Viewer	
e View Help	
192.168.1.103	💄 Sign in 🝷
Address book	92.168.0.110:5900 192.168.1.103

启动界面如下:

192.168.1.103	(Qt for Embedded Li	nux VNC Server) - RealVI	NC Viewer							
		US	BCAN调试助手	=(六叶树)	ii a		主题			Х
选择设备	设置	语言 工具	帮助							
USBCAN1 US		2000 Mini USBCANFD1 US	BCANFD2							
	这里演元	示用的是六叶树	JSBCAN2语	高配版,所	以这里选USB(CAN2				 一 数据清空 500 停止显示
基本发送 高级	8余洋	六叶树C	anAssistar	nt启动成功	1					
发送方式:	正常发送	数据(16进制):			()		- and the latest	20		
帧协议:	CAN	00 11 22 33 44 55 66 77	,		◎ ID递增		₩ 实时存储	fi		
帧类型:	标准帧 👻				每次发送顺	数:	1			
帧格式:	数据帧 👻				发送次数:		1			
帧ID(16进制):	0000000				发送间隔(毫	秒):	10			
数据长度:	8 🔻				开始		停止			
			ডিগ	备信息 发ì	长速率:0帧/秒 接收速率:(0帧/秒 发送	· 频数:0帧 接收(贞数:0帧	清空计	数

六叶树教育科技有限公司 © 版权所有





x	1	/	~	Γ)	
				~	



3 常用链接

.

六叶叔

六叶树官网:<u>www.liuyeshu.cn</u>

六叶树 CanAssistant 调试软件下载地址:<u>http://www.liuyeshu.cn/?page_id=492</u>
六叶树 USBCANFD 产品驱动下载地址:<u>http://www.liuyeshu.cn/?page_id=485</u>
六叶树 USBCANFD 适配器产品二次开发资料下载:<u>http://www.liuyeshu.cn/?page_id=440</u>
六叶树 USBCAN 适配器产品 SOCKETCAN 专栏资料下载:<u>http://www.liuyeshu.cn/?page_id=1256</u>
六叶树 USBCAN 适配器产品 ROS 机器人操作系统专栏:<u>http://www.liuyeshu.cn/?page_id=1893</u>
六叶树 USBCANFD 产品常见软硬件问题解决方案专栏:<u>http://www.liuyeshu.cn/?page_id=593</u>

4 联系方式

- 🐚 电话:15211065817(业务合作咨询)
- 🖂 邮箱:798746621@qq.com(业务咨询+技术支持)
- 👧 微信:18673379565(技术支持)
- 😋 官网:<u>www.liuyeshu.cn(</u>资料下载)
- **痴 网上商城:<u>https://shop112408209.taobao.com</u>(产品购买)**
 - 淘宝店铺搜索:"六叶树教育科技"