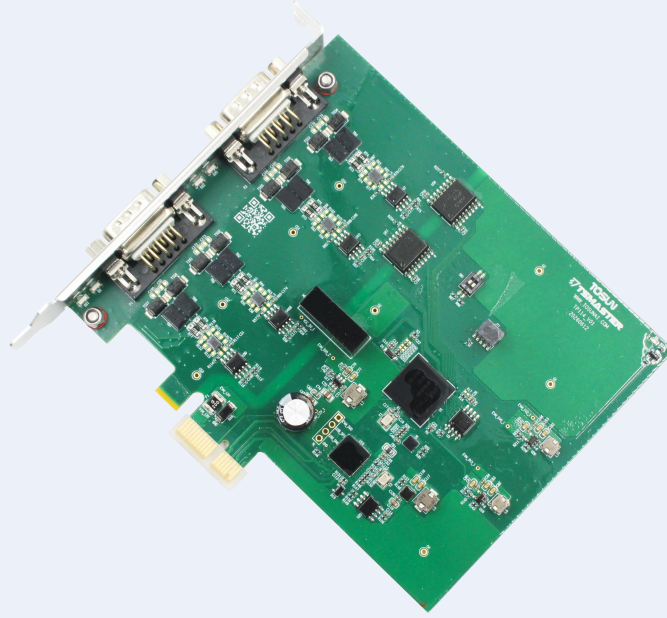




TOSUN-TP114

User Manual

Product Features & Interface Overview

SocketCAN to PCIe Interface

Product Name	Channel
TP114	CAN FD × 4

Copyright Information

Shanghai TOSUN Technology Ltd.

No. 9 Building, 1288 Jiasong North Road, Jiading District, Shanghai (Headquarters)

Buildings 14-17, Lane 4849 Cao'an Highway (Shanghai Research Institute)

In an effort to provide users with the best possible service, Shanghai TOSUN Technology Ltd. (hereinafter referred to as “TOSUN Technology”) has made every attempt to present accurate and detailed product information as possible in this manual. However, due to the time-sensitive nature of the content, TOSUN Technology cannot guarantee the timeliness and applicability of the information at all times.

The information and data contained in this manual are subject to change without prior notice. For the latest updates, please visit the [official website of TOSUN Technology](#) or contact our support team directly. We appreciate your understanding and continued support!

No part of this manual may be reproduced in any form or by any means without prior written permission from TOSUN Technology.

@ Copyright 2026-2027, Shanghai TOSUN Technology Ltd. All rights reserved.

Typical Application

- Vehicle-level CAN (FD) bus data acquisition and analysis
- Domain Controller Test
- Various automated test systems

Product Highlights

- Seamless Compatibility: Official Linux drivers are provided, with compatibility across mainstream Linux distributions.
- Flexible Configuration: Adjustable CAN channel bit rates from 125 kbps to 1 Mbps; CAN FD bit rates up to 5 Mbps.
- Broad Ecosystem Support: Supports Linux and Loongson Kylin operating systems, among others.
- Secure and Reliable: Uses a domestically developed main controller and supports fully domestically developed configurations.
- Open and Flexible: Provides cross-platform APIs for secondary development.

Contents

Product Features & Interface Overview	2
Typical Application	3
Product Highlights	3
1. Introduction	5
1.1. Technical Specifications	5
1.2. Pin Definitions	8
1.3. LED Indicators	9
1.4. System Requirements	9
1.5. Packing List	10
2. Application Example in Linux	11
2.1. Driver Installation	11
2.1.1. Environment Dependencies	11
2.1.2. Driver Installation Steps	11
2.2. Hardware Connection	13
2.2.1. CAN	13
2.3. Usage Example	14
2.3.1. SocketCAN Example	14
2.3.2. can-utils Example	14
2.3.3. Windows Usage	16
3. Appendix	17
3.1. Software Installation	17
4. Inspection and Maintenance	22

1. Introduction

The TP114 is a multi-channel CAN (FD) communication tool based on SocketCAN. It is designed to make CAN communication as simple and efficient as network programming.

The TP114 supports CAN FD bus speeds up to 5 Mbps and connects to the host via PCIe. It is fully compatible with SocketCAN and comes with a complete Linux driver for excellent system compatibility. The TP114 adopts driver-free design for Windows with plug-and-play support, and CAN transmission and reception can be implemented quickly via the Windows API.

Powered by a China-made main control chip, TP114 supports a fully domestic hardware configuration, making it suitable for a wide range of embedded and industrial communication applications.

The product package includes:

- Linux driver installation package
- Cross-platform secondary development library (with a dedicated programming manual)

1.1. Technical Specifications

➤ Device Specifications

Parameter	Description
Channel	CAN FD × 4
PC Interface	PCIe
Driver	Linux driver support
Buffer	Hardware buffer
Connector	Standard D-Sub, 9-pin
CAN	Supports CAN 2.0 A/B protocols (ISO 11898-1 compliant); baud rate 125 kbps ~ 1 Mbps
CAN FD	Supports both ISO and non-ISO CAN FD standards; baud rate 125 kbps ~ 5 Mbps

Power Supply	Powered via PCIe
Power Consumption	1.5 W
Sample Point Range	Adjustable from 70% ~ 90%
Dimensions	120 × 21.5 × 24 mm
Weight	106.2 g
Operating Humidity	10% ~ 90% (no condensing)

➤ **CAN/CAN FD Specifications**

Parameter	Description
Supported Protocols	CAN 2.0 A/B (ISO 11898-1), CAN FD (ISO and non-ISO)
CAN Baud Rate	125 kbps ~ 1 Mbps
CAN Frame Data Length	Up to 8 bytes
CAN FD Baud Rate	125 kbps ~ 5 Mbps
CAN FD Frame Data Length	Up to 64 bytes; supports BRS frames
Channel Count	CAN/CAN FD × 4
Buffer	Hardware buffer

➤ Mechanical Dimensions

Unit: mm

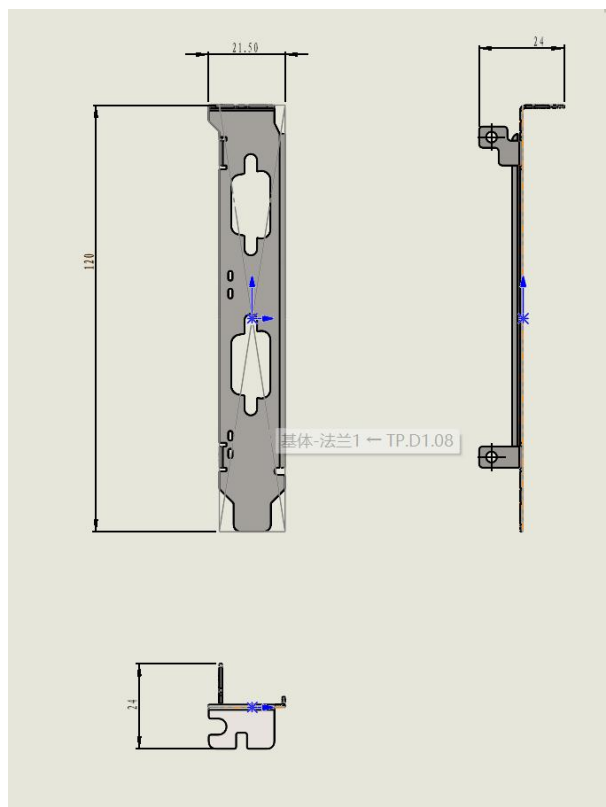


Figure 1-1 Mechanical Dimensions

1.2. Pin Definitions

➤ Hardware Interface



Figure 1-2 Hardware Interface (CAN FD)

1.3. LED Indicators

➤ LED Definitions

Indicator	Description
CAN FD 1~4	Status of CAN FD Channel1~4

➤ LED Color Description

Indicator	Color	Description
CAN FD	Green	Normal frame transmission/reception
	Red	Frame transmission/reception error — configuration, protocol, or wiring fault

1.4. System Requirements

➤ PC Requirements

- Operating System: Linux (official driver provided; compatible with mainstream distributions and Loongson Kylin OS)

➤ Build & Runtime Dependencies

- Kernel header package for the corresponding version
- GCC compiler
- Makefile build environment
- Shell (supported sh scripts)

➤ Downloads

- TP114 Linux driver source code package:

https://gitee.com/xujinpeng120/tsdev_release/tree/v20260128/05_socketcan/driver

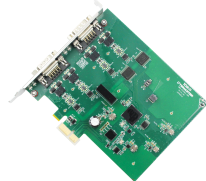
(download the latest version)

- PDF user manual
- Programming library (for secondary development)



The download link is available on the official website of Shanghai TOSUN Technology Ltd.: <https://www.tosunai.com/>

1.5. Packing List

Item	Qty.	Illustration	Standard/Optional
TP114 Main Device	1		Standard

2. Application Example in Linux

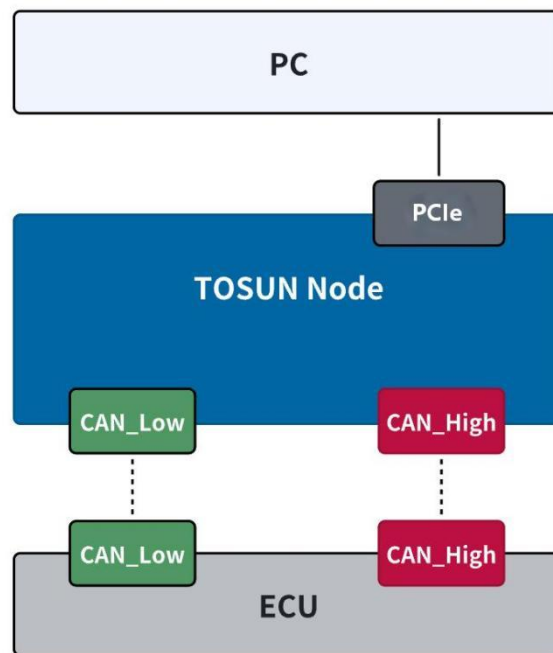


Figure 2-1 Example in Linux

2.1. Driver Installation

The TP114 provides a Linux driver with excellent system compatibility. Before installation, make sure you have downloaded the latest driver source code package as described in Section 1.4.

2.1.1. Environment Dependencies

1. Install the kernel header package for your system version (with CAN device support).
2. Install the GCC compiler.
3. Ensure Makefile support is available.
4. Enable shell script (.sh) execution environment.

2.1.2. Driver Installation Steps

1. Copy the user directory to the Linux environment and enter the directory.

```
tosun@tosun-HP-ProBook-455-15-6-inch-G9-Notebook-PC:~/Desktop/tsdev_release/05_socketcan$ ls
app  doc  driver  README.md
```

Figure 2-2 User Directory

2. Enter the driver folder and run make to compile the driver. The generated file is `tosun_socket_can.ko`.

```
tosun@tosun-HP-ProBook-455-15-6-inch-G9-Notebook-PC:~/Desktop/tsdev_release/05_socketcan$ cd driver
tosun@tosun-HP-ProBook-455-15-6-inch-G9-Notebook-PC:~/Desktop/tsdev_release/05_socketcan/driver$ make
-e .....
-e  Checking build environment...
-e .....
# make
# sudo
# kernel build dir
# 必要头文件
# 必要源文件
-e  Using gcc-12, version 12
-e  Building kernel module tosun_socket_can.ko for 6.8.0-65-generic...
Using CC=gcc-12
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-65-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
You are using: gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
CC [M] /home/tosun/Desktop/tsdev_release/05_socketcan/driver/drivers_net_can_usb_tosun.o
CC [M] /home/tosun/Desktop/tsdev_release/05_socketcan/driver/drivers_net_can_usb_cdev.o
CC [M] /home/tosun/Desktop/tsdev_release/05_socketcan/driver/drivers_net_can_usb_tsmassage.o
LD [M] /home/tosun/Desktop/tsdev_release/05_socketcan/driver/tosun_socket_can.o
MODPOST /home/tosun/Desktop/tsdev_release/05_socketcan/driver/Module.symvers
CC [M] /home/tosun/Desktop/tsdev_release/05_socketcan/driver/tosun_socket_can.mod.o
LD [M] /home/tosun/Desktop/tsdev_release/05_socketcan/driver/tosun_socket_can.ko
BTF [M] /home/tosun/Desktop/tsdev_release/05_socketcan/driver/tosun_socket_can.ko
Skipping BTF generation for /home/tosun/Desktop/tsdev_release/05_socketcan/driver/tosun_socket_can.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-65-generic'
-e  Build complete: tosun_socket_can.ko
tosun@tosun-HP-ProBook-455-15-6-inch-G9-Notebook-PC:~/Desktop/tsdev_release/05_socketcan/driver$
```

Figure 2-3 Example Figure 1

3. Run `sudo make install` to load `can_dev` and `hid` dependencies, then install the driver.

```
tosun@tosun-HP-ProBook-455-15-6-inch-G9-Notebook-PC:~/Desktop/tsdev_release/05_socketcan/driver$ make install
-e  Installing kernel module...
[sudo] password for tosun:
-e  Installed. Load with: sudo modprobe tosun_socket_can
```

Figure 2-4 Example Figure 2

4. The driver does not produce console output during loading. After connecting the device, run `ip link show` to verify the CAN interface.

```

tosun@tosun-HP-ProBook-455-15-6-inch-G9-Notebook-PC:~/Desktop/tsdev_release/05_socketcan/driver$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 7c:57:58:19:12:f9 brd ff:ff:ff:ff:ff:ff
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DORMANT group default qlen 1000
    link/ether 38:d5:7a:99:a1:19 brd ff:ff:ff:ff:ff:ff
4: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 72 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can
5: can1: <NOARP,UP,LOWER_UP,ECHO> mtu 72 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can
6: can2: <NOARP,UP,LOWER_UP,ECHO> mtu 72 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can
7: can3: <NOARP,UP,LOWER_UP,ECHO> mtu 72 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can

```

Figure 2-5 Example Figure 3

5. To uninstall the driver, run `make uninstall`.

2.2. Hardware Connection

2.2.1. CAN

Use the included DB9 Female–Dual Male Signal Cable (CAN) to access two independent channels via two D-Sub 9-pin connectors.

The following diagram shows the pin mapping of the “DB9 Female–Dual Male Signal Cable (CAN)” harness:

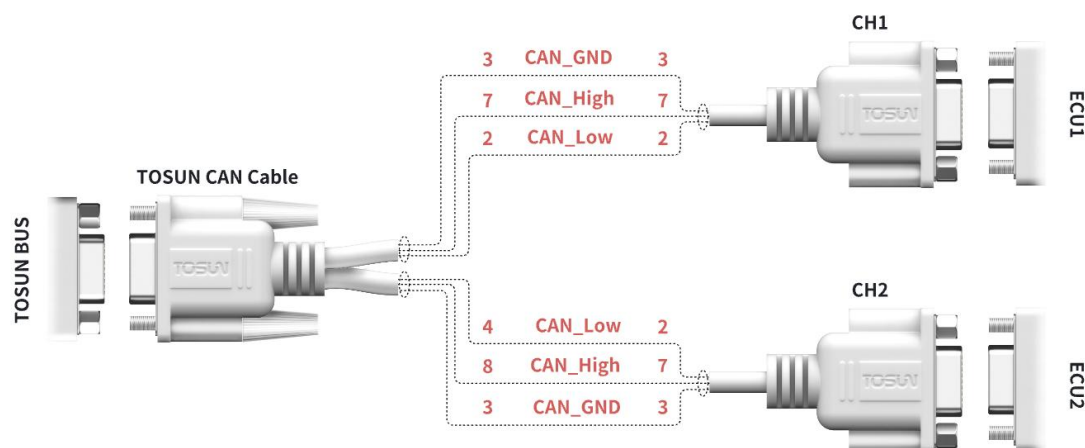


Figure 2-6 DB9 Female-Dual Male Signal Cable (CAN)

2.3. Usage Example

2.3.1. SocketCAN Example

TP114 functions as a standard SocketCAN device, fully compatible with the Linux SocketCAN subsystem API.

- Official Documentation: For complete SocketCAN APIs, concepts, and usage, refer to the official Linux kernel documentation: Linux source tree →
/Documentation/networking/can.rst

(Online version: <https://www.kernel.org/doc/html/latest/networking/can.html>.)

- Practical Tutorial: For beginners, third-party resources such as SocketCAN – Wikipedia provide a quick overview of key concepts.

The examples in this document assume a basic understanding of SocketCAN.

2.3.2. can-utils Example

Below is an example showing how to use all four channels of the TP114.

1. Install the open-source CAN tools can-utils:

```
sudo apt update
```

```
sudo apt install can-utils -y
```

2. Bring down all CAN network interfaces before configuration:

```
sudo ip link set can0 down
```

```
sudo ip link set can1 down
```

```
sudo ip link set can2 down
```

```
sudo ip link set can3 down
```

3. Configure baud rate and sample points (example: arbitration phase 500 kbps with 87.5% sample point, data phase 2,000 kbps with 75% sample point on all four channels):

```
sudo ip link set can0 type can bitrate 500000 sample-point 0.875  
dbitrate 2000000 dsample-point 0.75 fd on
```

```
sudo ip link set can1 type can bitrate 500000 sample-point 0.875
```

```
dbitrate 2000000 dsample-point 0.75 fd on

sudo ip link set can2 type can bitrate 500000 sample-point 0.875
dbitrate 2000000 dsample-point 0.75 fd on

sudo ip link set can3 type can bitrate 500000 sample-point 0.875
dbitrate 2000000 dsample-point 0.75 fd
```

4. Bring the configured interfaces up:

```
sudo ip link set can0 up

sudo ip link set can1 up

sudo ip link set can2 up

sudo ip link set can3 up
```

5. Send CAN frames:

```
cansend <interface> <id>#<data>

cansend can0 123#DEADBEEF

cansend can1 123#DEADBEEF

cansend can2 123#DEADBEEF

cansend can3 123#DEADBEE
```

6. Receive CAN frames:

```
candump <interface>

candump can0

candump can1

candump can2

candump can3
```

The figure below shows the received frame data.

```
tosun@tosun-HP-ProBook-455-15-6-inch-G9-Notebook-PC:~/Desktop/tsdev_release$ candump can0  
can0 123 [4] DE AD BE EF  
can0 123 [4] DE AD BE EF  
can1 123 [4] DE AD BE EF
```

Figure 2-7 Received Frame Data

2.3.3. Windows Usage

Obtain the TOSUN open-source component package. On Windows, device access is provided via the `libTSDevBase` library, which exposes the API for interacting with the hardware.

3. Appendix

3.1. Software Installation

The section describes the steps for installing the TSMaster software on a Windows PC.

➤ TSMaster Software Download

<https://www.tosunai.com/downloads/>

If the site is unavailable, contact your sales representative or visit the TOSUN official website.

You may also scan the QR code below to follow the official WeChat account and obtain download links.



Figure 3-1 TOSUN Official WeChat Account

➤ Software Installation

1. Double-click the TSMaster installer and select the installation language.

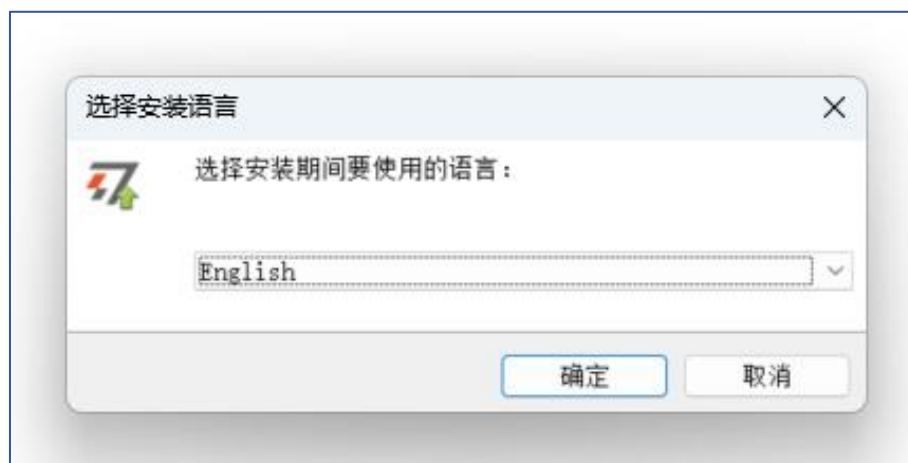


Figure 3-2 TSMaster Installation

2. Accept the license agreement and click “Next”.

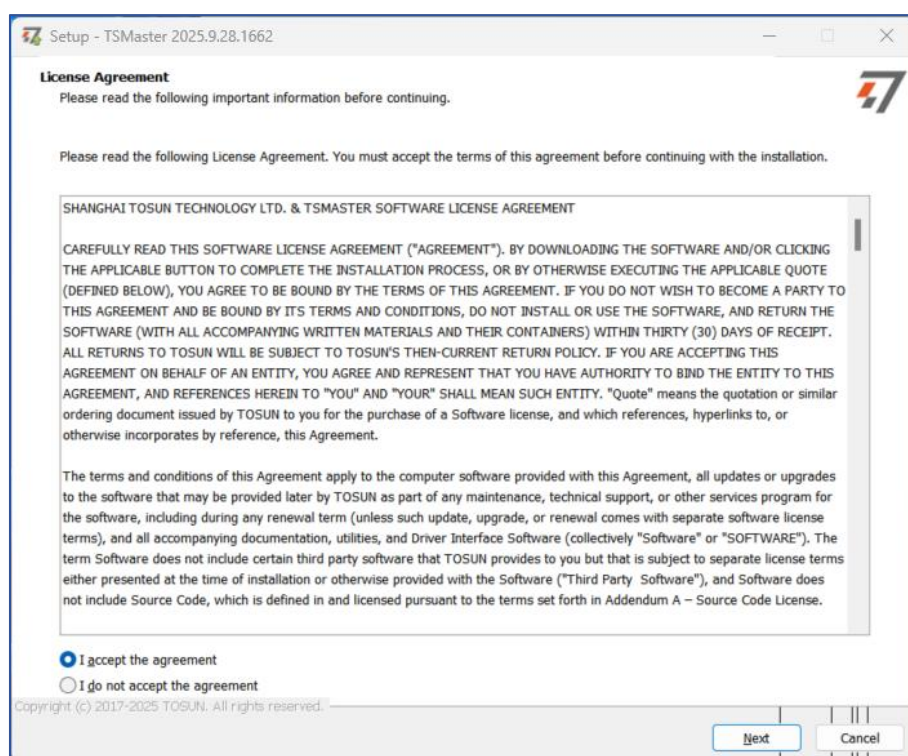


Figure 3-3 TSMaster Installation

3. Choose an installation directory and click “Next”.

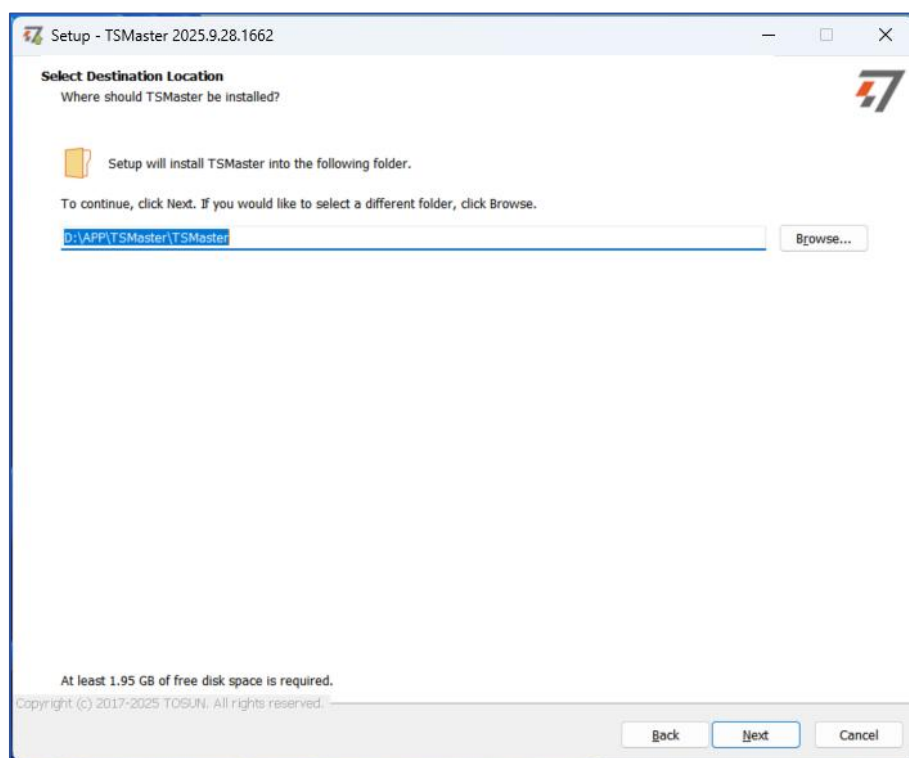


Figure 3-4 TSMaster Installation

4. Select additional tasks as needed and click “Next”.

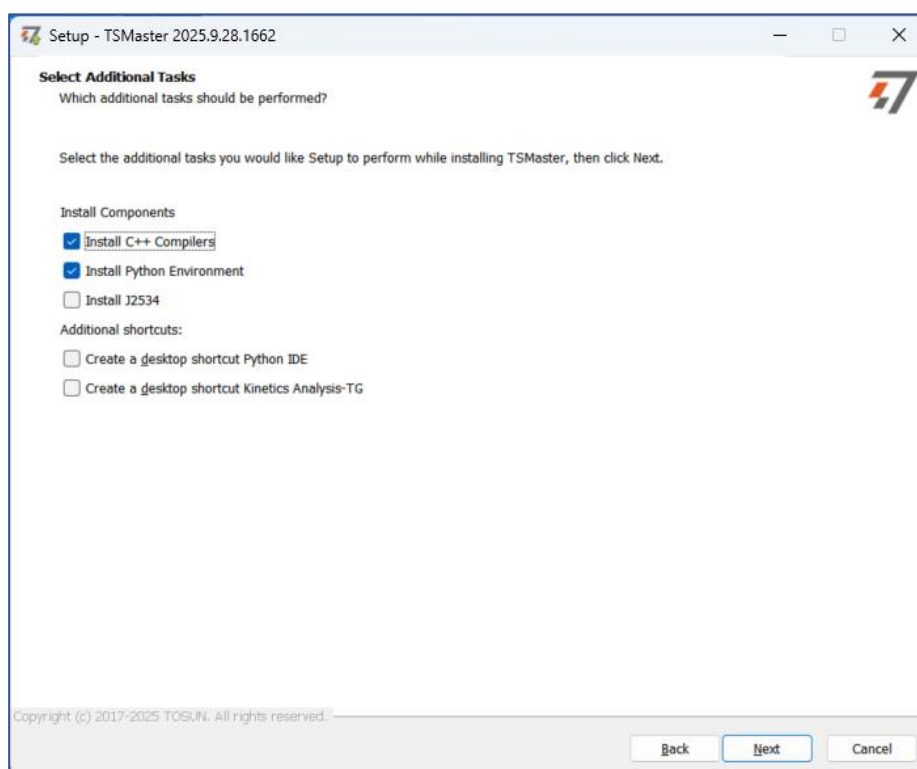


Figure 3-5 TSMaster Installation

5. Click “Install” and wait for completion.

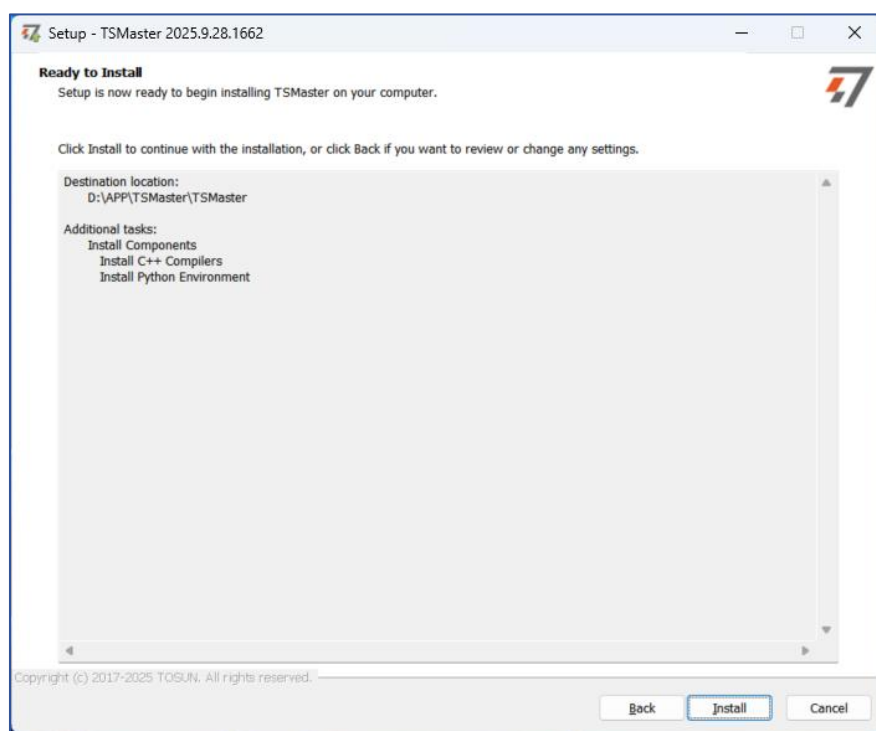


Figure 3-6 TSMaster Installation

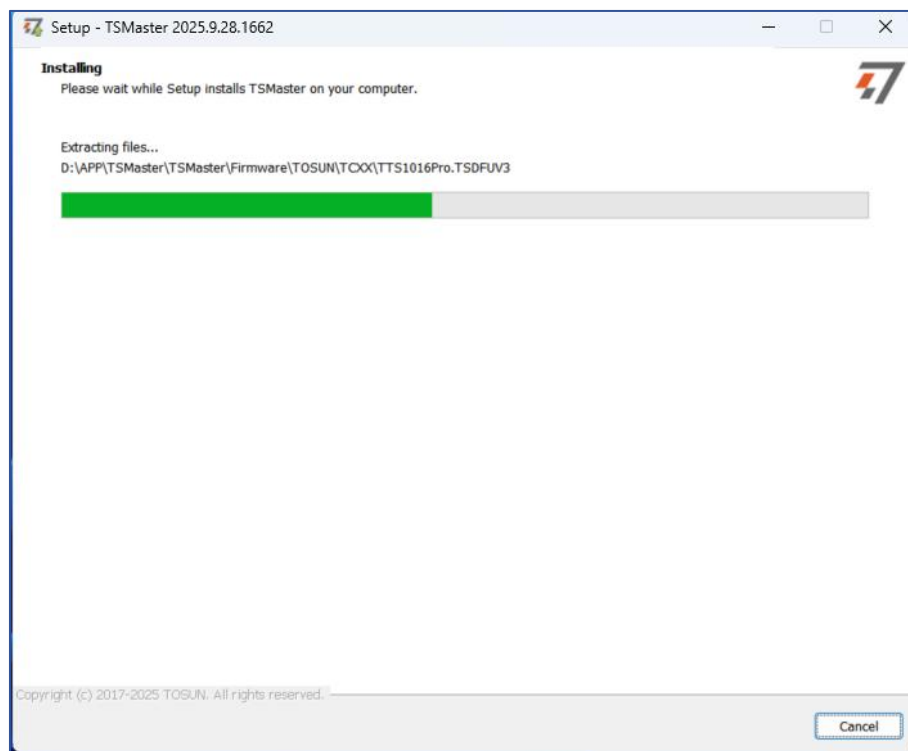


Figure 3-7 TSMaster Installation

6. Click “Finish” to complete installation.

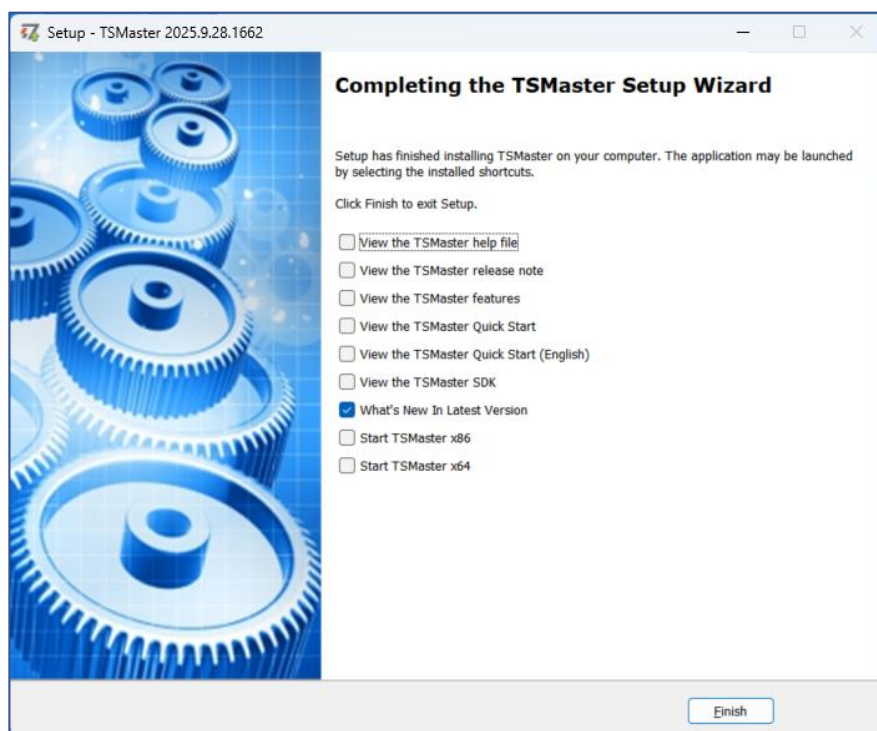


Figure 3-8 TSMaster Installation

4. Inspection and Maintenance

The TP114 primarily contains semiconductor components, which typically have a long service life. However, adverse environmental conditions may accelerate aging and degrade performance. To ensure proper operation, regular inspections are recommended to maintain the required environmental conditions.

It is recommended to inspect the device at least once every 6 to 12 months. In harsher environments, inspections should be performed more frequently. Refer to the table below for inspection criteria and recommended actions. If issues persist, please contact Shanghai TOSUN Technology Ltd.

➤ Power Environment Inspection

Item	Check Content	Standard/ Range	Action/Measure
Ambient Conditions	Check ambient temperature (including internal temperature within enclosures)	-40°C ~ +80°C	Use a thermometer to ensure temperature is within specified range
	Check the ambient humidity (including internal humidity within enclosures)	10% ~ 90% RH	Use a hygrometer to ensure humidity is within specified range

➤ Contamination & Protection Check

Item	Check Content	Standard/ Range	Action/Measure
Contamination	Check for accumulation of dust, powder, salt, and metal debris	None	Clean the device and prevent future contamination

	Check for exposure to water, oil, or chemicals	None	Clean and shield if necessary
Hazardous Gases	Check for corrosive or flammable gases	None	Use sensors or odor detection to verify

➤ **Mechanical Stress & EMI Check**

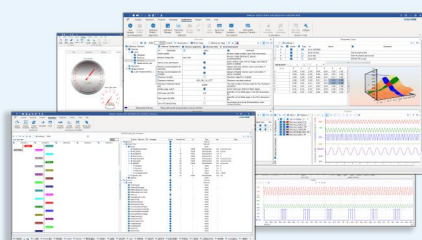
Item	Check Content	Standard/Range	Action/Measure
Mechanical Stress	Check vibration and shock levels	Within specified limits	Install padding or vibration isolation measures if necessary
Electromagnetic Environment	Check for noise sources near the device	No significant noise sources	Isolate or shield the device from noise sources

➤ **Installation & Wiring Check**

Item	Check Content	Standard/Range	Action/Measure
Wiring	Check crimped connectors in external wiring	Adequate clearance between connectors	Visually inspect and adjust as needed
	Check for damage to external wiring	No damage	Visually inspect and replace damaged cables if necessary

Software

Support CAN(FD)/LIN/FlexRay/SOME/IP and DoIP
 UDS diagnostics/ECU flashing/CCP/XCP calibration
 Embedded code generation/Application builder
 Encrypted release/Logging and bus replay
 Graphical programming/Residual bus simulation
 C and Python scripting
 Bus monitoring/Transmitting/Automated testing



TSMaster

Hardware

1/2/4/8/12-channel CAN FD/CAN to USB/PCIe device
 1/2/6-channel LIN to USB/PCIe device
 Multi channel FlexRay/CAN FD to USB/PCIe device
 Multi channel automotive Ethernet/CAN FD to USB/PCIe device
 Automotive Ethernet media conversion device (T1 to Tx)
 Multi-channel CAN FD/Ethernet/LIN datalogger



TTS test systems

- CAN FD/CAN/FlexRay/LIN communication boards
- Relay and fault injection boards
- Resistors for sensor simulation
- Digital I/O, Analog I/O boards available



Solutions

- Bus Conformance
- Network Automation Testing System
- Charging Testing System
- EMB Calibration Testing Equipment
- Information Security Solutions
- Steer-by-Wire Chassis Testing Solutions
- EOL Testing Equipment
- Motor Performance
- Durability Testing Solutions
- FCT



About TOSUN

The core product, TSMaster, is a comprehensive tool for automotive R&D, testing, production, and after-sales. It integrates essential functions with hardware support to streamline processes and ensure precision, making it ideal for automotive professionals.

International Organization



Quality Assurance

ISO9001:2015

CE Certification



Contact Us :

+86 21-5956 0506
 sales@tosunai.com

website :

www.tosunai.com

