

Manuals+

[Q & A](#) | [Deep Search](#) | [Upload](#)

manuals.plus /

- › [Waveshare](#) /
- › [Waveshare 4.2inch E-Ink Display Module User Manual](#)

Waveshare 4.2inch E-Ink Display Module

Waveshare 4.2inch E-Ink Display Module User Manual

Model: 4.2inch E-Ink Display Module

1. INTRODUCTION

The Waveshare 4.2inch E-Ink Display Module features a 400x300 resolution and supports partial refresh. It is designed for low power consumption, offers a wide viewing angle, and provides a paper-like display effect. This module includes an embedded controller and communicates via an SPI interface, making it compatible with various controller boards such as Raspberry Pi, Jetson Nano, Arduino, and STM32.

Its ultra-low power consumption and ability to retain display content without continuous power make it suitable for applications like electronic shelf labels and industrial instrumentation.

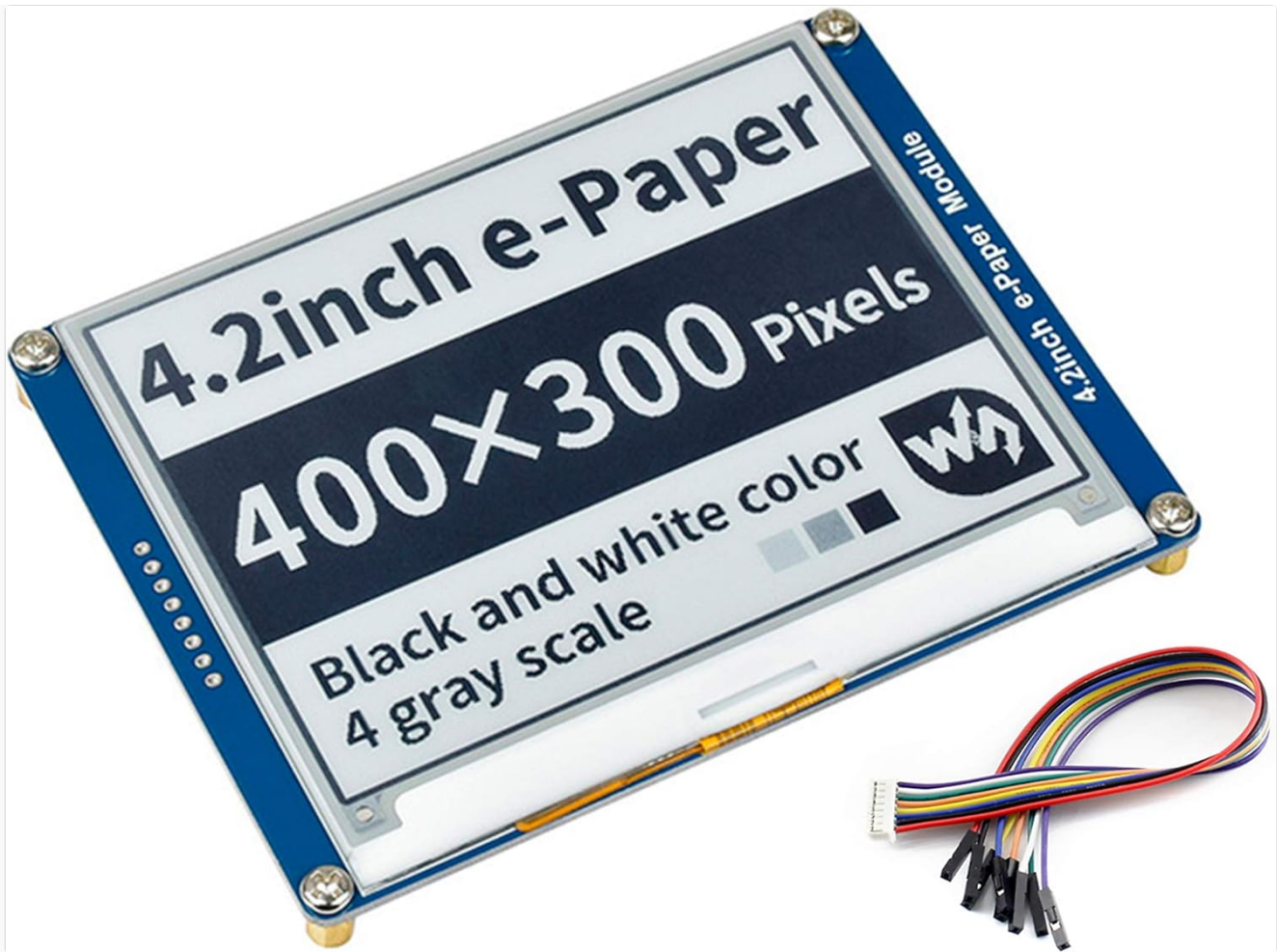


Image 1.1: Waveshare 4.2inch E-Ink Display Module with connection wires.

2. KEY FEATURES

- **Display Technology:** 4.2inch E-Ink (e-Paper) with 400x300 pixel resolution.
- **Power Efficiency:** Ultra-low power consumption; power is primarily required only during display refresh cycles.
- **Persistent Display:** No backlight required. Retains the last displayed content indefinitely, even when power is removed.
- **Interface:** SPI interface (3-wire or 4-wire) for broad compatibility.
- **Controller Compatibility:** Designed for use with Raspberry Pi, Jetson Nano, Arduino, STM32, and other microcontrollers.
- **Voltage Translation:** Onboard voltage translator supports both 3.3V and 5V MCUs.
- **Refresh Capability:** Supports partial refresh for faster updates of specific screen areas.
- **Viewing Angle:** Wide viewing angle, offering clear visibility from over 170 degrees.

4.2" E-Paper Module

Partial Refresh Support, Low Power, Wide Viewing Angle, Paper-Like Effect

Ideal for price tags, shelf labels, industrial instruments...




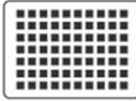

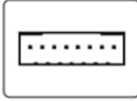
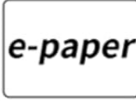





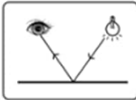
| | | | | | |
|---|--|--|---|--|--|
| Size  4.2" | Resolution  400x300 | Viewing Angle  >170° | Communication  SPI | Display Panel  E-paper | Experience  Paper-like |
| Refreshing  Partial Refresh Support | Display Color  Black and White | Grey Scale  4 | Environment  Ambient Light Required | Display Type  Passively Reflective | |

Image 2.1: Overview of the 4.2-inch E-Paper Module's key characteristics.

3. PACKAGE CONTENTS

The following items are included in your product package:

- 1x Waveshare 4.2inch e-Paper Module
- 1x PH2.0 20cm 8-pin cable

Product Show



Outline Dimensions

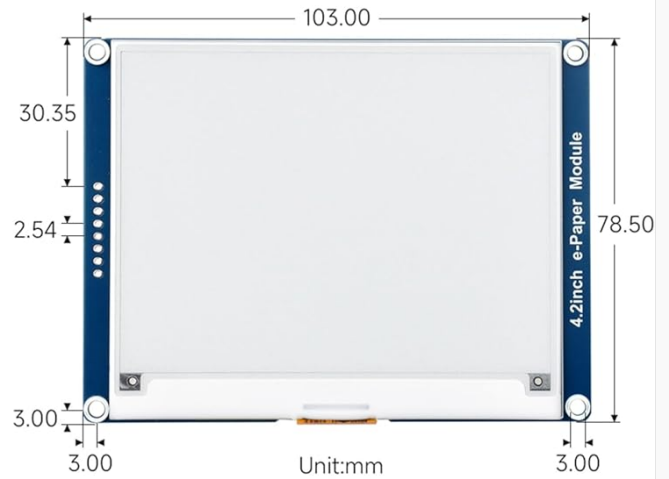


Image 3.1: Contents of the product package, including the e-Paper module and connection cable.

4. SPECIFICATIONS

Detailed technical specifications for the 4.2inch E-Ink Display Module are provided below:

| Parameter | Value | Parameter | Value |
|--------------------|------------------------|----------------------|-----------------------|
| Operating Voltage | 3.3V/5V | Display Color | Black, White |
| Interface | 3-wire SPI, 4-wire SPI | Grey Level | 4 |
| Outline Dimensions | 103.0 × 78.5mm | Full Refresh Time | 5s |
| Display Size | 84.8 × 63.6mm | Fast Refresh Time | 1s/1.5s* |
| Dot Pitch | 0.212 × 0.212mm | Partial Refresh Time | 0.4s |
| Viewing Angle | >170° | Refresh Power | 26.4mW (typ.) |
| Resolution | 400 × 300 pixels | Standby Current | <0.01uA (almost none) |

*Fast refresh function is only available in V2 version.

Features At A Glance

- No backlight, keeps displaying last content for a long time even when power down
- Ultra low power consumption, basically power is only required for refreshing
- SPI interface, for connecting with controller boards like Raspberry Pi/Jetson Nano/Arduino/STM32, etc.
- Onboard voltage translator, compatible with 3.3V / 5V MCUs
- Comes with online development resources and manual (driver board circuit diagram, examples for Raspberry Pi/Jetson Nano/Arduino/STM32)

Specifications

| | | | |
|---------------------------|------------------------|-----------------------------|-----------------------|
| OPERATING VOLTAGE | 3.3V/5V | DISPLAY COLOR | Black/White |
| INTERFACE | 3-wire SPI, 4-wire SPI | GREY SCALE | 4 |
| OUTLINE DIMENSIONS | 103.0 × 78.5mm | FULL REFRESH TIME | 5s |
| DISPLAY SIZE | 84.8 × 63.6mm | FAST REFRESH TIME | 1s/1.5s* |
| DOT PITCH | 0.212 × 0.212mm | PARTIAL REFRESH TIME | 0.4s |
| VIEWING ANGLE | >170° | REFRESH POWER | 26.4mW(typ.) |
| RESOLUTION | 400 × 300 pixels | STANDBY CURRENT | <0.01uA (almost none) |

* Fast refresh function is only available in V2 version

Image 4.1: Summary of features and detailed specifications for the e-Paper module.

Hardware Connection

1. Working With Raspberry Pi

When connecting the Raspberry Pi, you can directly insert the board into the 40PIN pin header of the Raspberry Pi, and pay attention to the correct pins. If you choose to connect with an 8PIN cable, please refer to the pin correspondence table below:

| e-Paper | Raspberry Pi | |
|---------|--------------|-------|
| | BCM2835 | Board |
| VCC | 3.3V | 3.3V |
| GND | GND | GND |
| DIN | MOSI | 19 |
| CLK | SCLK | 23 |
| CS | CE0 | 24 |
| DC | 25 | 22 |
| RST | 17 | 11 |
| BUSY | 24 | 18 |

2. Working With Arduino

Use an 8PIN cable to connect, please refer to the pin correspondence table below:

| e-Paper | Arduino UNO | Mega2560 |
|---------|-------------|----------|
| VCC | 5V | 5V |
| GND | GND | GND |
| DIN | D11 | D51 |
| CLK | D13 | D52 |
| CS | D10 | D10 |
| DC | D9 | D9 |
| RST | D8 | D8 |
| BUSY | D7 | D7 |

3. Working With Jetson Nano

The 40PIN pin of Jetson Nano is compatible with the 40PIN pin of RPi and provides a Jetson.GPIO library with the same API as the Raspberry Pi.GPIO library of RPi, so the serial number connected here is the same as that of Raspberry Pi. The module can be directly inserted into the 40Pin headers of the Jetson Nano when using the 40PIN interface.

If you choose to connect with an 8PIN cable, please refer to the pin correspondence table below:

| e-Paper | Jetson Nano Developer Kit | |
|---------|---------------------------|-------|
| | BCM2835 | Board |
| VCC | 3.3V | 3.3V |
| GND | GND | GND |
| DIN | 10 (SPI0_MOSI) | 19 |
| CLK | 11 (SPI0_SCK) | 23 |
| CS | 8 (SPI0_CS0) | 24 |
| DC | 25 | 22 |
| RST | 17 | 11 |
| BUSY | 24 | 18 |

Image 4.2: Detailed outline dimensions of the 4.2inch E-Ink Display Module.

5. SETUP AND HARDWARE CONNECTION

The 4.2inch E-Ink Display Module communicates via an SPI interface. Ensure proper pin connections to your chosen microcontroller board.

5.1 Onboard SPI Control Interface

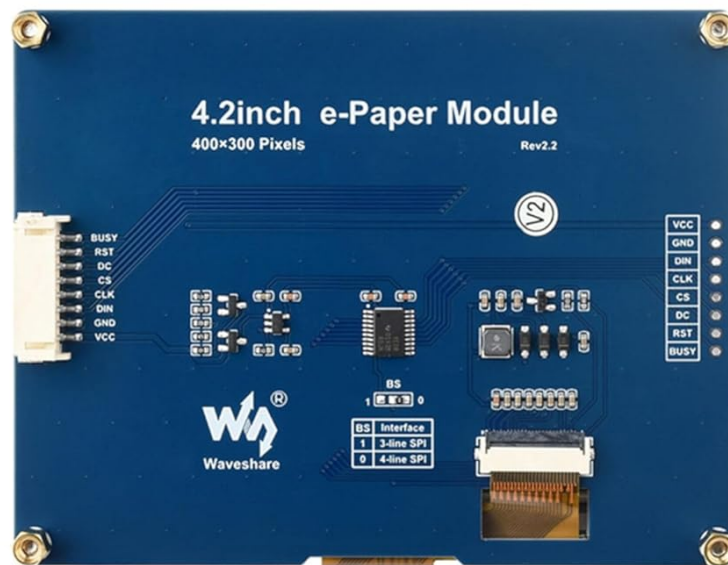
The module features an onboard SPI control interface. The pin definitions are crucial for correct operation.

| Pin | Description |
|-----|----------------------|
| VCC | 3.3V/5V Power Supply |
| GND | Ground |
| DIN | SPI MOSI pin |
| CLK | SPI SCK pin |

| Pin | Description |
|------|---|
| CS | SPI chip selection, low active |
| DC | Data / Command selection (high for data, low for command) |
| RST | External reset, low active |
| BUSY | Busy status output |

Onboard SPI Control Interface

For Use With Other Controller Boards Like Raspberry Pi/Jetson Nano/Arduino/STM32



| | |
|-------------|---|
| VCC | 3.3V/5V |
| GND | Ground |
| DIN | SPI MOSI pin |
| CLK | SPI SCK pin |
| CS | SPI chip selection, low active |
| DC | Data / Command selection (high for data, low for command) |
| RST | External reset, low active |
| BUSY | Busy status output |

Image 5.1: Diagram of the onboard SPI control interface and its pin definitions.

5.2 Onboard Voltage Translator

The module includes an onboard voltage translator, ensuring compatibility with both 3.3V and 5V microcontrollers. This simplifies integration by eliminating the need for external level shifters.

Onboard Voltage Translator

Compatible With 3.3V / 5V MCUs

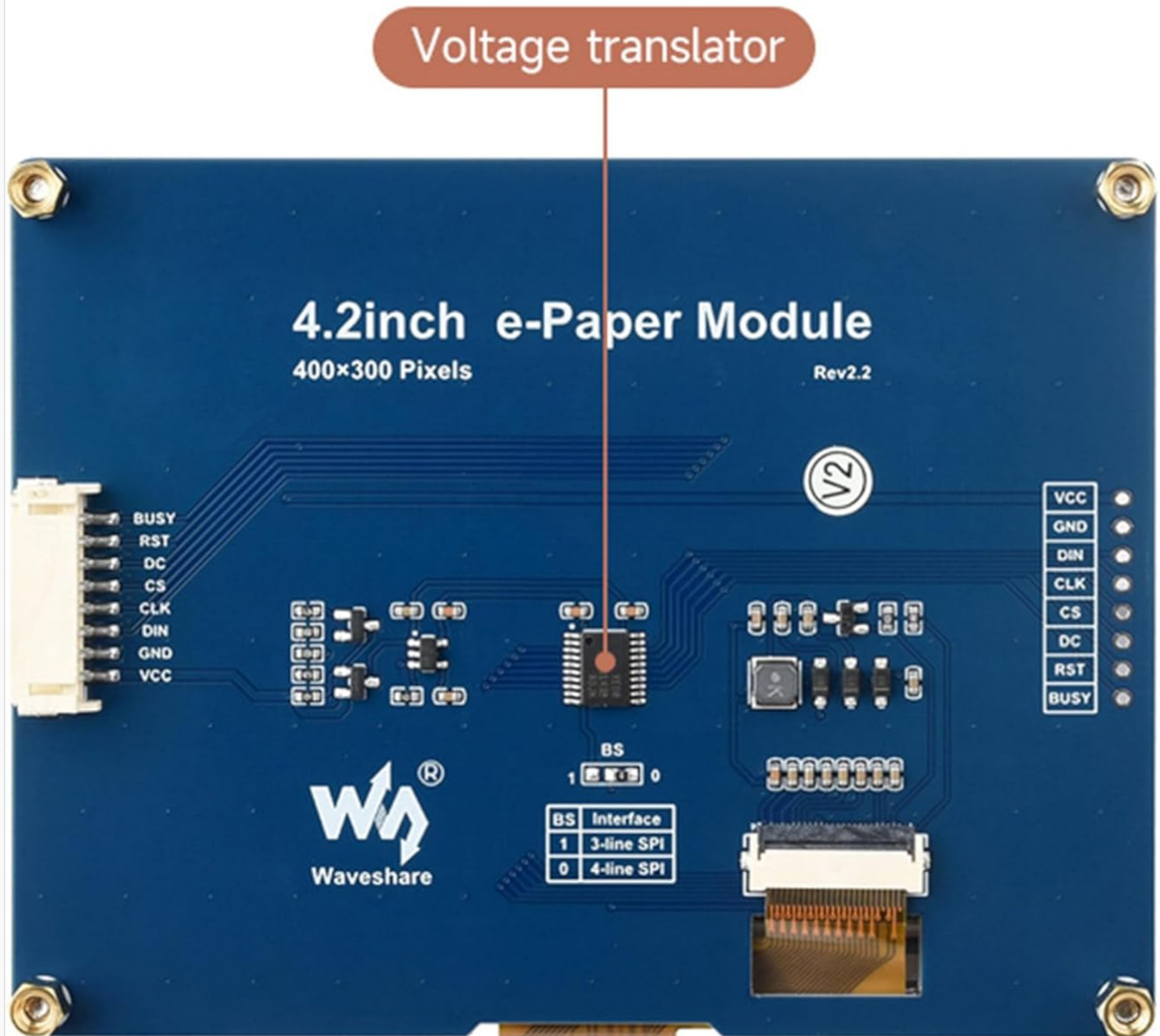


Image 5.2: Location of the onboard voltage translator on the module.

5.3 Connection with Raspberry Pi

When connecting to a Raspberry Pi, you can directly insert the module into the 40-pin GPIO header. If using an 8-pin cable, refer to the pin correspondence table below:

| e-Paper | BCM2835 | Board |
|---------|---------|-------|
| VCC | 3.3V | 3.3V |
| GND | GND | GND |
| DIN | MOSI | 19 |

| e-Paper | BCM2835 | Board |
|---------|---------|-------|
| CLK | SCLK | 23 |
| CS | CE0 | 24 |
| DC | 25 | 22 |
| RST | 17 | 11 |
| BUSY | 24 | 18 |

5.4 Connection with Arduino

Use an 8-pin cable to connect the module to an Arduino board. Refer to the pin correspondence table below for Arduino UNO and Mega2560:

| e-Paper | Arduino UNO | Mega2560 |
|---------|-------------|----------|
| VCC | 5V | 5V |
| GND | GND | GND |
| DIN | D11 | D51 |
| CLK | D13 | D52 |
| CS | D10 | D10 |
| DC | D9 | D9 |
| RST | D8 | D8 |
| BUSY | D7 | D7 |

5.5 Connection with Jetson Nano

The 40-pin GPIO of Jetson Nano is compatible with the 40-pin RPi GPIO. The module can be directly inserted into the 40-pin headers of the Jetson Nano when using the 40-pin interface. If you choose to connect with an 8-pin cable, refer to the pin correspondence table below:

| e-Paper | BCM2835 | Board |
|---------|----------------|-------|
| VCC | 3.3V | 3.3V |
| GND | GND | GND |
| DIN | 10 (SPI0_MOSI) | 19 |
| CLK | 11 (SPI0_SCK) | 23 |
| CS | 8 (SPI0_CS0) | 24 |

| e-Paper | BCM2835 | Board |
|---------|---------|-------|
| DC | 25 | 22 |
| RST | 17 | 11 |
| BUSY | 24 | 18 |

Image 5.3: Pin connection tables for Raspberry Pi, Arduino, and Jetson Nano.

6. OPERATING PRINCIPLES (E-INK TECHNOLOGY)

E-paper displays utilize microcapsule electrophoretic technology for displaying images. The principle involves charged particles suspended in a clear fluid moving to the sides of microcapsules when an electric field is applied. This process makes the microcapsule content visible by reflecting ambient light, similar to traditional printed paper. E-paper displays offer clear visibility under lamplight or natural light, do not require a backlight, and provide a wide viewing angle of nearly 180 degrees. This paper-like effect makes them an ideal choice for applications requiring readability and low power consumption.

Advantages Of EINK

E-paper display utilizes microcapsule electrophoretic technology for displaying, the principle is: charged particles suspended in clear fluid will move to sides of microcapsule when electric field is applied, making the microcapsule become visible by reflecting ambient light, just as traditional printed paper.

E-paper display will clearly display images/texts under lamplight or natural light, requires no backlight, and features nearly up to 180° viewing angle. It is usually used as e-reader due to its paper-like effect.



Application Examples

Suitable For Price Tags, Asset/Equipment Tags, Shelf Labels, Conference Name Tags...



Image 6.1: Comparison of E-Paper display with traditional printed paper and LCD monitors, highlighting E-Ink advantages.

7. APPLICATION EXAMPLES

The Waveshare 4.2inch E-Ink Display Module is versatile and suitable for various applications where low power,

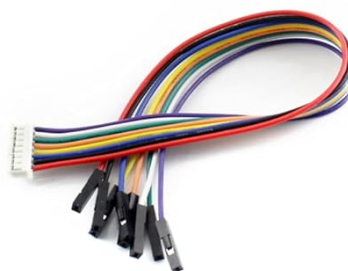
clear display, and persistent content are beneficial. Common uses include:

- Price Tags in retail environments
- Asset and Equipment Tags for inventory management
- Electronic Shelf Labels
- Conference Name Tags
- Industrial instrumentation displays

PACKAGE CONTENT



1. 4.2inch e-Paper Module x1



2. PH2.0 20cm 8Pin x1

Image 7.1: Visual examples of the e-Paper module used in various settings such as supermarkets, hospitals, and conference venues.

8. MAINTENANCE

E-Ink displays are generally low-maintenance. To ensure longevity and optimal performance:

- **Handling:** Handle the module by its edges to avoid touching the display surface directly.
- **Cleaning:** If necessary, gently wipe the display surface with a soft, dry, lint-free cloth. Avoid using harsh chemicals or abrasive materials.
- **Storage:** Store the module in a dry, dust-free environment within its specified operating temperature range.
- **Power:** Ensure stable power supply within the specified voltage range (3.3V/5V).

9. TROUBLESHOOTING

If you encounter issues with your Waveshare 4.2inch E-Ink Display Module, consider the following common troubleshooting steps:

- **No Display/Blank Screen:**
 - Verify all power connections (VCC, GND) are correct and stable.
 - Check SPI data lines (DIN, CLK, CS, DC, RST, BUSY) for proper connection and continuity.
 - Ensure your microcontroller code is correctly initializing the display and sending data.
 - Confirm the correct voltage (3.3V or 5V) is being supplied to the module.
- **Incorrect/Garbled Display:**
 - Double-check your SPI communication protocol and timing in your code.
 - Ensure the correct display resolution (400x300) is configured in your software.
 - Verify that the data format (e.g., black/white, grey levels) matches the display's capabilities and your code's output.
- **Slow Refresh Rate:**
 - E-Ink displays inherently have slower refresh rates than LCDs. Refer to the specifications for expected refresh times.
 - If using partial refresh, ensure your code is correctly implementing it for faster updates of specific areas.
- **Module Not Responding:**
 - Perform a hard reset by momentarily disconnecting and reconnecting power.
 - Check the BUSY pin status in your code to ensure the display is not busy before sending new commands.

For more detailed troubleshooting or specific error codes, refer to the online development resources provided by Waveshare.

10. SUPPORT AND RESOURCES

Waveshare provides comprehensive online development resources and manuals to assist with the integration and programming of this module. These resources typically include driver board circuit diagrams and example code for Raspberry Pi, Jetson Nano, Arduino, and STM32.

Access the official online development resources here: bit.ly/4btDibU

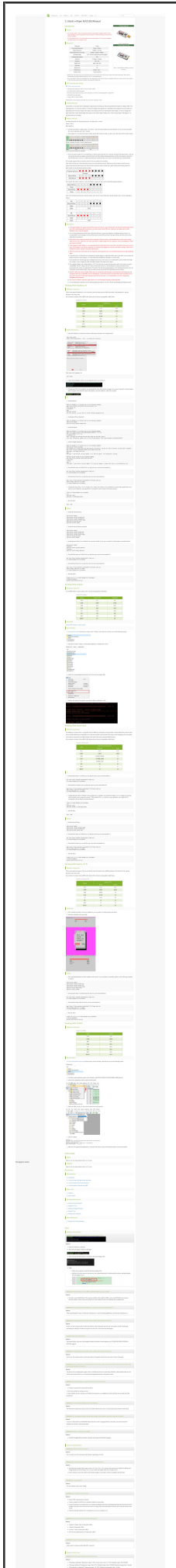
For further assistance, please visit the Waveshare official website or contact their customer support.

11. WARRANTY INFORMATION

Warranty information for the Waveshare 4.2inch E-Ink Display Module is typically provided by the seller or manufacturer at the time of purchase. Please retain your proof of purchase for any warranty claims.

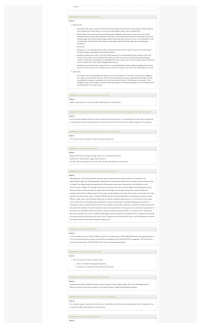
For specific details regarding warranty terms, duration, and claim procedures, refer to the documentation included with your product or contact the retailer/manufacturer directly.

Related Documents - 4.2inch E-Ink Display Module



[Waveshare 2.13inch e-Paper HAT \(B\) User Manual and Technical Guide](#)

Comprehensive guide for the Waveshare 2.13inch e-Paper HAT (B), covering hardware connections, software setup, programming principles, and troubleshooting for Raspberry Pi, Arduino, Jetson Nano, and STM32.



7-Track e-Page HST Manual

LUCKFOX

Introduction

This manual provides detailed instructions for the operation and maintenance of the 7-Track e-Page HST. It covers safety precautions, installation procedures, and troubleshooting steps. The user is advised to read this manual thoroughly before using the device to ensure safe and effective operation.

Specifications

| Parameter | Value |
|-----------------------|--------------------|
| Model | 7-Track e-Page HST |
| Power Supply | 24VDC |
| Operating Temperature | 0°C to 40°C |
| Storage Temperature | -20°C to 60°C |
| Humidity | 10% to 90% RH |

Installation

The device should be installed in a well-ventilated area, away from direct sunlight and moisture. The mounting surface should be clean and dry. The power and data cables should be connected according to the wiring diagram provided in the manual.

Operation

The device is controlled via a web interface. The user should access the interface using a web browser and enter the device's IP address. The interface allows for configuration of various parameters, including network settings, user access, and system status monitoring.

Maintenance

Regular maintenance is required to ensure the device's longevity and performance. This includes cleaning the device, checking the power supply, and updating the firmware as needed. The manual provides detailed instructions for each of these tasks.

Troubleshooting

The manual includes a comprehensive troubleshooting section that addresses common issues such as power problems, network connectivity, and device malfunctions. The user is encouraged to follow the steps provided to identify and resolve the problem.

Appendix

The appendix contains additional information, including a glossary of terms, a list of accessories, and contact information for technical support.



[Waveshare 7.5-inch E-Paper HAT User Manual and Guide](#)

This comprehensive user manual provides detailed information on the Waveshare 7.5-inch E-Paper HAT (V1/V2), an 800x480 resolution display module utilizing Microencapsulated Electrophoretic Display technology. It covers hardware connections, SPI communication, working principles, and integration with Raspberry Pi, Arduino, Jetson Nano, Sunrise X3 Pi, STM32, ESP32, and ESP8266. Essential precautions, resources, and FAQs are included for optimal use.


```
root@kali:~# apt-get install libx11-dev libxext-dev libxrender-dev libxrandr-dev libxss-dev libxft-dev libxkbfile-dev libxv-dev libxvmc-dev libxinerama-dev libxi-dev libxt-dev libxtst-dev libxvuln-dev libxvmc-dev libxinerama-dev libxi-dev libxt-dev libxtst-dev libxvuln-dev
root@kali:~# apt-get install libx11-dev libxext-dev libxrender-dev libxrandr-dev libxss-dev libxft-dev libxkbfile-dev libxv-dev libxvmc-dev libxinerama-dev libxi-dev libxt-dev libxtst-dev libxvuln-dev
root@kali:~# apt-get install libx11-dev libxext-dev libxrender-dev libxrandr-dev libxss-dev libxft-dev libxkbfile-dev libxv-dev libxvmc-dev libxinerama-dev libxi-dev libxt-dev libxtst-dev libxvuln-dev
```

Reboot Raspberry Pi:
sudo reboot

After the reboot the Pi is not occupied by other devices, you can check in the middle of booting by:

Install Libraries

Open the Raspberry Pi terminal and run the following command:
sudo apt-get install libx11-dev libxext-dev libxrender-dev libxrandr-dev libxss-dev libxft-dev libxkbfile-dev libxv-dev libxvmc-dev libxinerama-dev libxi-dev libxt-dev libxtst-dev libxvuln-dev

Install xorg-driver

Open the Raspberry Pi terminal and run the following command:
sudo apt-get install libx11-dev libxext-dev libxrender-dev libxrandr-dev libxss-dev libxft-dev libxkbfile-dev libxv-dev libxvmc-dev libxinerama-dev libxi-dev libxt-dev libxtst-dev libxvuln-dev

Install Xorg-server

Open the Raspberry Pi terminal and run the following command:
sudo apt-get install libx11-dev libxext-dev libxrender-dev libxrandr-dev libxss-dev libxft-dev libxkbfile-dev libxv-dev libxvmc-dev libxinerama-dev libxi-dev libxt-dev libxtst-dev libxvuln-dev

Download Drivers

Open Raspberry Pi terminal and run the following command:
sudo apt-get install libx11-dev libxext-dev libxrender-dev libxrandr-dev libxss-dev libxft-dev libxkbfile-dev libxv-dev libxvmc-dev libxinerama-dev libxi-dev libxt-dev libxtst-dev libxvuln-dev

Run the driver code

There is a list of Raspberry Pi drivers called that are not the commands in terminal

Code

To compile the driver code

```
gcc -o main.c
```

The first program of all drivers can be used directly by entering the corresponding driver

Depending on the OS, one of the following commands should be entered:

```
1. sudo ./main.c
2. sudo ./main.c
3. sudo ./main.c
4. sudo ./main.c
5. sudo ./main.c
6. sudo ./main.c
7. sudo ./main.c
8. sudo ./main.c
9. sudo ./main.c
10. sudo ./main.c
```

python

Enter the python program directory and run the command:

```
python main.py
```

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

FBPC Porting

The FBPC is currently not compatible with 64-bit Raspberry Pi system, it is recommended to use 32-bit system.

FBPCPorter uses a video output device to drive a video display device from a memory device containing compressed data. This device is a video output device that can be used to drive a video output, and the device content can be changed by changing the data in the memory.

There is an open source project on GitHub: FBPCPorter. Compared with other FBPCPorter, this project uses a video output device to drive a video output of a video display device.

Download Drivers

Open the Raspberry Pi terminal and run the following command:
sudo apt-get install libx11-dev libxext-dev libxrender-dev libxrandr-dev libxss-dev libxft-dev libxkbfile-dev libxv-dev libxvmc-dev libxinerama-dev libxi-dev libxt-dev libxtst-dev libxvuln-dev

Method 1: Use a script (recommended)

Use the first program of all drivers that can be used directly by entering the corresponding driver

Depending on the OS, one of the following commands should be entered:

The script will update the corresponding drivers to all 64-bit Raspberry Pi and 32-bit Raspberry Pi.

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

Method 2: Manual Configuration

Environment Configuration

Raspberry Pi can be used to drive a video display device, so we need to clean up the video output before installing the driver.

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

Set up FBPC automatically

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

main.py

After installing the system, the Raspberry Pi OS user interface will be displayed.



The screenshot shows the 'Properties' dialog for a C++ file. The 'Compiler' tab is active, displaying various compiler options and settings. The 'Include Paths' section is expanded, showing a list of include directories. The 'Compiler' section shows the compiler version and other settings.

Program Description

Document Introduction

This module (and controlling I/O) is an example, open the hardware's user's manual.

| File | Size | Time |
|------------|------|---------|
| main.cpp | 1024 | 0:00:00 |
| main.o | 1024 | 0:00:00 |
| main.exe | 1024 | 0:00:00 |
| main.pdb | 1024 | 0:00:00 |
| main.res | 1024 | 0:00:00 |
| main.lib | 1024 | 0:00:00 |
| main.dll | 1024 | 0:00:00 |
| main.exp | 1024 | 0:00:00 |
| main.man | 1024 | 0:00:00 |
| main.hlp | 1024 | 0:00:00 |
| main.ico | 1024 | 0:00:00 |
| main.rc | 1024 | 0:00:00 |
| main.rc2 | 1024 | 0:00:00 |
| main.rc6 | 1024 | 0:00:00 |
| main.rc7 | 1024 | 0:00:00 |
| main.rc8 | 1024 | 0:00:00 |
| main.rc9 | 1024 | 0:00:00 |
| main.rc10 | 1024 | 0:00:00 |
| main.rc11 | 1024 | 0:00:00 |
| main.rc12 | 1024 | 0:00:00 |
| main.rc13 | 1024 | 0:00:00 |
| main.rc14 | 1024 | 0:00:00 |
| main.rc15 | 1024 | 0:00:00 |
| main.rc16 | 1024 | 0:00:00 |
| main.rc17 | 1024 | 0:00:00 |
| main.rc18 | 1024 | 0:00:00 |
| main.rc19 | 1024 | 0:00:00 |
| main.rc20 | 1024 | 0:00:00 |
| main.rc21 | 1024 | 0:00:00 |
| main.rc22 | 1024 | 0:00:00 |
| main.rc23 | 1024 | 0:00:00 |
| main.rc24 | 1024 | 0:00:00 |
| main.rc25 | 1024 | 0:00:00 |
| main.rc26 | 1024 | 0:00:00 |
| main.rc27 | 1024 | 0:00:00 |
| main.rc28 | 1024 | 0:00:00 |
| main.rc29 | 1024 | 0:00:00 |
| main.rc30 | 1024 | 0:00:00 |
| main.rc31 | 1024 | 0:00:00 |
| main.rc32 | 1024 | 0:00:00 |
| main.rc33 | 1024 | 0:00:00 |
| main.rc34 | 1024 | 0:00:00 |
| main.rc35 | 1024 | 0:00:00 |
| main.rc36 | 1024 | 0:00:00 |
| main.rc37 | 1024 | 0:00:00 |
| main.rc38 | 1024 | 0:00:00 |
| main.rc39 | 1024 | 0:00:00 |
| main.rc40 | 1024 | 0:00:00 |
| main.rc41 | 1024 | 0:00:00 |
| main.rc42 | 1024 | 0:00:00 |
| main.rc43 | 1024 | 0:00:00 |
| main.rc44 | 1024 | 0:00:00 |
| main.rc45 | 1024 | 0:00:00 |
| main.rc46 | 1024 | 0:00:00 |
| main.rc47 | 1024 | 0:00:00 |
| main.rc48 | 1024 | 0:00:00 |
| main.rc49 | 1024 | 0:00:00 |
| main.rc50 | 1024 | 0:00:00 |
| main.rc51 | 1024 | 0:00:00 |
| main.rc52 | 1024 | 0:00:00 |
| main.rc53 | 1024 | 0:00:00 |
| main.rc54 | 1024 | 0:00:00 |
| main.rc55 | 1024 | 0:00:00 |
| main.rc56 | 1024 | 0:00:00 |
| main.rc57 | 1024 | 0:00:00 |
| main.rc58 | 1024 | 0:00:00 |
| main.rc59 | 1024 | 0:00:00 |
| main.rc60 | 1024 | 0:00:00 |
| main.rc61 | 1024 | 0:00:00 |
| main.rc62 | 1024 | 0:00:00 |
| main.rc63 | 1024 | 0:00:00 |
| main.rc64 | 1024 | 0:00:00 |
| main.rc65 | 1024 | 0:00:00 |
| main.rc66 | 1024 | 0:00:00 |
| main.rc67 | 1024 | 0:00:00 |
| main.rc68 | 1024 | 0:00:00 |
| main.rc69 | 1024 | 0:00:00 |
| main.rc70 | 1024 | 0:00:00 |
| main.rc71 | 1024 | 0:00:00 |
| main.rc72 | 1024 | 0:00:00 |
| main.rc73 | 1024 | 0:00:00 |
| main.rc74 | 1024 | 0:00:00 |
| main.rc75 | 1024 | 0:00:00 |
| main.rc76 | 1024 | 0:00:00 |
| main.rc77 | 1024 | 0:00:00 |
| main.rc78 | 1024 | 0:00:00 |
| main.rc79 | 1024 | 0:00:00 |
| main.rc80 | 1024 | 0:00:00 |
| main.rc81 | 1024 | 0:00:00 |
| main.rc82 | 1024 | 0:00:00 |
| main.rc83 | 1024 | 0:00:00 |
| main.rc84 | 1024 | 0:00:00 |
| main.rc85 | 1024 | 0:00:00 |
| main.rc86 | 1024 | 0:00:00 |
| main.rc87 | 1024 | 0:00:00 |
| main.rc88 | 1024 | 0:00:00 |
| main.rc89 | 1024 | 0:00:00 |
| main.rc90 | 1024 | 0:00:00 |
| main.rc91 | 1024 | 0:00:00 |
| main.rc92 | 1024 | 0:00:00 |
| main.rc93 | 1024 | 0:00:00 |
| main.rc94 | 1024 | 0:00:00 |
| main.rc95 | 1024 | 0:00:00 |
| main.rc96 | 1024 | 0:00:00 |
| main.rc97 | 1024 | 0:00:00 |
| main.rc98 | 1024 | 0:00:00 |
| main.rc99 | 1024 | 0:00:00 |
| main.rc100 | 1024 | 0:00:00 |

Underlying hardware interface

The hardware interface is defined in the file `HW_Compiled.cpp` and functions such as `read` and `write` are used. The hardware interface is implemented in the file `HW_Compiled.h`.

```

// HW_Compiled.h
#ifndef HW_COMPILED_H
#define HW_COMPILED_H

// ...

#endif

```

The main application

The main application is defined in the file `main.cpp`. It includes the hardware interface and the `main` function.

```

// main.cpp
#include "HW_Compiled.h"

// ...

int main() {
    // ...
}

```

HW_Compiled.cpp

```

// HW_Compiled.cpp
#include "HW_Compiled.h"

// ...

void HW_Compiled::read() {
    // ...
}

void HW_Compiled::write() {
    // ...
}

```

HW_Compiled.h

```

// HW_Compiled.h
#ifndef HW_COMPILED_H
#define HW_COMPILED_H

// ...

#endif

```

main.cpp

```

// main.cpp
#include "HW_Compiled.h"

// ...

int main() {
    // ...
}

```

HW_Compiled.cpp

```

// HW_Compiled.cpp
#include "HW_Compiled.h"

// ...

void HW_Compiled::read() {
    // ...
}

void HW_Compiled::write() {
    // ...
}

```

HW_Compiled.h

```

// HW_Compiled.h
#ifndef HW_COMPILED_H
#define HW_COMPILED_H

// ...

#endif

```

main.cpp

```

// main.cpp
#include "HW_Compiled.h"

// ...

int main() {
    // ...
}

```

```
void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}

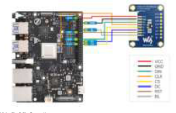
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

VideoFive2

Adaptive Video

- 3.5 inch LCD Module
- 3.5 inch LCD Module
- 3.5 inch LCD Module
- 3.5 inch LCD Module
- 3.5 inch LCD Module
- 3.5 inch LCD Module
- 3.5 inch LCD Module

Hardware Connection



| Pin | Function |
|-----|----------|
| 1 | VCC |
| 2 | GND |
| 3 | VCC |
| 4 | GND |
| 5 | VCC |
| 6 | GND |
| 7 | VCC |
| 8 | GND |
| 9 | VCC |
| 10 | GND |
| 11 | VCC |
| 12 | GND |

Install Corresponding Libraries

```
sudo apt-get install python3-pip
sudo pip3 install RPi.GPIO
```

Clone Download

```
git clone https://github.com/adafruit/Adafruit_CircuitPython_ILI9341.git
cd Adafruit_CircuitPython_ILI9341
python3 setup.py install
```

Run the Corresponding Demo According to the Screen You Purchase

```
python3 ili9341_demo.py
python3 ili9341_demo.py
python3 ili9341_demo.py
python3 ili9341_demo.py
python3 ili9341_demo.py
python3 ili9341_demo.py
```

Resource

- Documents
- GitHub
- GitHub

Software

- Raspbian

Demo codes

- Demo codes

3D Drawing

- 3.5 inch LCD Module 3D Drawing

3D Drawing

- 3.5 inch LCD Module 3D Drawing

FAQ

Question 1: The LCD is blank when using with Raspberry Pi?

Answer: Please make sure you have installed the RPi.GPIO library. Check the output of the pip3 if it doesn't have any value, please try to download it.

Question 2: How to change the display orientation?

Answer: If you use the C code, you can use the PinLib SoftwareSerial function to set the display orientation, and the SPI, I2C, and the I2C are available. If you use Python code, you can use the Adafruit_ILI9341 class to change the orientation in any ways.

Question 3: Why does the screen display become blurry?

Answer: Please make sure when copying the software example please try to install the Pi3. Blurred by command: sudo apt-get install python3-pip

Question 4: Why does the screen display become blurry when connected to an Arduino?

Answer: Please try to use the software example in the GitHub repository.

Question 5: Is it possible to use of Raspberry Pi controls?

Answer: If you use the software example, then you can use the RPi.GPIO module to control the display. If you use the C code, you can use the PinLib SoftwareSerial function to control the display. If you use the Python code, you can use the Adafruit_ILI9341 class to control the display. If you use the C code, you can use the PinLib SoftwareSerial function to control the display. If you use the Python code, you can use the Adafruit_ILI9341 class to control the display. If you use the C code, you can use the PinLib SoftwareSerial function to control the display. If you use the Python code, you can use the Adafruit_ILI9341 class to control the display.

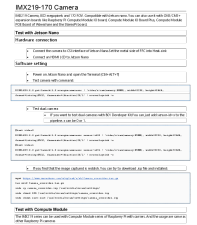
Support

If you require technical support, please go to the page and open a ticket.



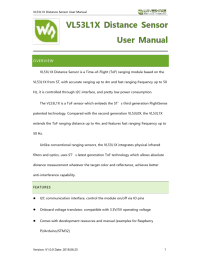
[ArduCam MEGA SPI Camera Getting Started Guide](#)

A comprehensive guide to getting started with the ArduCam MEGA SPI Camera, detailing connection, setup, and operation with Arduino microcontrollers and other platforms.



[IMX219-170 Camera User Guide for Jetson Nano and Compute Module](#)

A guide to using the IMX219-170 camera with Jetson Nano and Raspberry Pi Compute Modules, including hardware connection, software setup, and troubleshooting.



[VL53L1X Distance Sensor User Manual and Integration Guide](#)

A comprehensive user manual for the Waveshare VL53L1X Time-of-Flight (ToF) distance sensor. It details the sensor's specifications, features, pinouts, and provides step-by-step guides for integration with popular development platforms like Raspberry Pi, Arduino, and STM32, including demo code instructions.