



E-paper 24P Shield

User Manual · Arduino Compatible E-Paper Shield

Product Name	E-paper 24P Shield
Version	Rev 1.0
Compatible Interface	Arduino standard interface (UNO R3 pin definition)
Screen Interface	FPC 24P (0.5 mm pitch)
Communication	SPI (CS / DC / RES / BUSY + SCK / SDA shared I/O)
Operating Voltage	3.3 V (regulated by the Shield from the Arduino mainboard)

This manual applies to the E-paper 24P Shield (Rev 1.0)

1. Product Overview

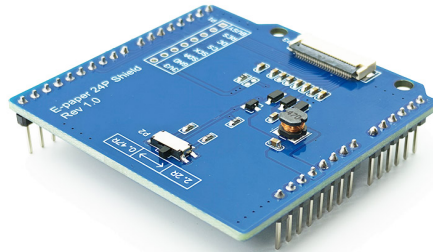


Fig. 1-1 E-paper 24P Shield physical photo (side view)

E-paper 24P Shield is an **e-paper shield compatible with the Arduino standard interface**. Through the onboard FPC 24P connector (0.5 mm pitch), it accepts various FPC 24P interface e-paper screens, and pairs with any mainboard supporting the Arduino interface via standard Arduino pin headers.

Key Features

- **Plug and Play:** using the universal Arduino pin-header interface, it stacks directly onto Arduino UNO / Leonardo / Mega and other standard mainboards.
- **Multi-Screen Compatible:** via the FPC 24P interface, it supports various 24P e-paper modules (black/white / black/white/red / black/white/red/yellow etc.).



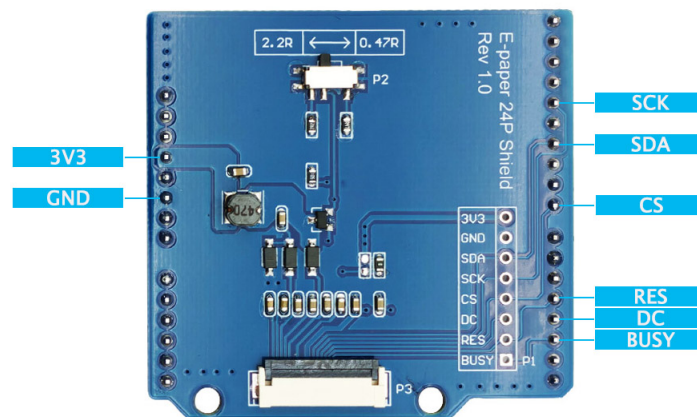
- **Vendor Compatible:** via the P2 jumper, select $2.2\ \Omega$ / $0.47\ \Omega$ resistance to match different vendors' e-paper screens.
- **Low-Power Friendly:** an e-paper screen consumes zero power for static display, ideal for battery-powered IoT labels, price tags, reminder boards and similar scenarios.

About the Arduino Interface

The Arduino header interface is a universal standard in the open-source hardware world and is not exclusive to any particular mainboard. Any mainboard following the Arduino UNO R3 pin definition (such as Arduino UNO, Leonardo, Mega, ESP32-Arduino compatible boards, etc.) can be used directly with this Shield.



2. Hardware Interfaces



All unmarked are: NC

Fig. 2-1 E-paper 24P Shield I/O layout (unlabeled pins are NC)

The shield connects to the outside through two interface groups: the **Arduino pin headers** (to the mainboard) and the **FPC 24P socket** (to the e-paper screen). The table below lists the pins actually used by this Shield; do not connect the remaining unlabeled (NC) pins.

Pin Description

Pin	Direction	Description
3V3	Power input	Supplied from the 3.3 V rail of the Arduino mainboard; powers the e-paper driver circuit.



GND	Power ground	Common ground, connected to the GND of the Arduino mainboard.
SCK	SPI clock	SPI bus clock line, output by the Arduino host.
SDA	SPI data / MOSI	Data line sent by the SPI host and received by the e-paper screen.
CS	Chip select	Active low; selects the current e-paper screen for communication.
DC	Data Command	High: data transfer; low: command transfer.
RES	Reset	Low level resets the e-paper driver; a falling edge after power-up is recommended.
BUSY	Status input	E-paper busy flag; high means a refresh is in progress and the MCU should wait.

Onboard Jumpers / Components

- **P1 (FPC 24P Socket):** connects the e-paper's FPC flat cable; mind the orientation (contacts facing up, press the latch closed).
- **P2 Resistance Jumper:** choose between $2.2\ \Omega$ and $0.47\ \Omega$ to match screens from different vendors.
- **Onboard Boost Inductor:** provides the high drive voltage required by the e-paper screen; no external power supply needed.

3. Application Combinations

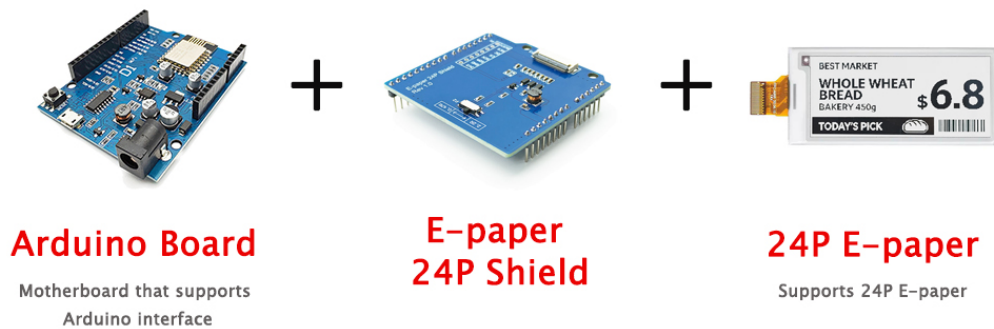


Fig. 3-1 Arduino Mainboard + E-paper 24P Shield + 24P E-Paper Screen — the Three-Module Combo

E-paper 24P Shield plays the role of a **"universal bridge"** in this solution. It is **not bound to any specific mainboard or screen** — as long as both interfaces meet the standard, they combine freely.

Arduino Mainboard: Freely Replaceable

Arduino interface is the open-source hardware world's **universal standard interface**. Any mainboard whose pinout follows the Arduino UNO R3 standard can host this Shield, for example:

- Arduino UNO R3 / Leonardo / Mega 2560 and other official mainboards;
- ESP32 / ESP8266 and other Arduino compatible-header Wi-Fi IoT development boards;
- Raspberry Pi Pico and other open-source Arduino form-factor MCU boards.



24P E-Paper Screen: Freely Replaceable

Any e-paper module whose pins are defined as FPC 24P and controlled via SPI can be connected; common sizes and color options include:

- 1.54" / 2.13" / 2.9" / 4.2" and other common e-price-tag sizes;
- Monochrome (BW), black/white/red (BWR), black/white/red/yellow (BWRY) and other display color combinations.

In One Sentence

With one E-paper 24P Shield, **any Arduino mainboard and any 24P e-paper screen**,

you can quickly build a low-power e-paper display solution, reusable across multiple projects.



4. Usage Instructions

Hardware Connection Steps

1. Align the E-paper 24P Shield pin headers with the matching sockets of the Arduino mainboard and, **after confirming the orientation**, press straight down.
2. According to the selected e-paper screen, choose P2 jumper value $2.2\ \Omega$ or $0.47\ \Omega$.
3. Insert the e-paper's FPC 24P flat cable with its **metal contacts facing up** into the P1 slot, then press the connector latch firmly closed.
4. Supply power via USB or an external source to the Arduino mainboard; the screen automatically enters the refresh-standby state.

Software Example (Arduino UNO)

The following example uses the GxEPD library to drive a 2.9" black/white/red e-paper screen; the pin mapping matches this Shield:

```
#include <GxEPD2_BW.h>

#define EPD_CS 10
#define EPD_DC 9
#define EPD_RST 8
#define EPD_BUSY 7
#define EPD_SCK 13
#define EPD_SDA 11
```



```
GxEPD2_BW<GxEPD2_290_BS,          GxEPD2_290_BS::HEIGHT>          display(  
    EPD_SDA,      EPD_SCK,      EPD_CS,      EPD_DC,      EPD_RST,      EPD_BUSY);  
  
void                                setup()                                {  
    display.init();  
    display.setRotation(1);  
    display.setFullWindow();  
    display.firstPage();  
    display.fillScreen(GxEPD_WHITE);  
    display.setTextColor(GxEPD_BLACK);  
    display.setCursor(20,                                20);  
    display.print("Hello,                                E-paper!");  
    display.nextPage();  
}  
  
void    loop()    {    /*    static    image,    no    loop    needed    */    }
```

Usage Notes

- ① Power-on order: seat the FPC cable first, then power the Arduino, to avoid damage from hot-plugging.
- ② Do not cut power during a refresh, otherwise ghosting may remain; run one full refresh to recover.
- ③ When not in use for long periods, it is recommended to switch off the power supply to reduce static power consumption.



6. FAQ

Q1: No display at all after power-on?

Troubleshoot in this order: ① Check whether the FPC 24P cable is fully seated and correctly oriented; ② Measure 3V3 and GND with a multimeter for 3.3 V; ③ Confirm the pin definitions in your program match the table in this manual; ④ Inside `setup()`, before calling `display.hibernate()`, wait for BUSY to go low. You can also refer to the provided example sketch and flash it directly.

Q2: Garbled display / ghosting?

Usually the refresh cycle did not complete. Call `display.refresh()` once outside the loop for a full refresh; if the problem persists, check whether the P2 jumper resistance matches the e-paper screen specification.

Q3: Can a 5 V powered Arduino mainboard be used?

Yes. The Shield's 3V3 is supplied by the regulator on the Arduino mainboard itself; a 5 V mainboard (e.g. Arduino UNO) has an onboard 3.3 V LDO that can power the Shield directly.

Q4: Are other communication protocols supported (e.g. I2C screens)?

This Shield is designed for SPI interface 24P e-paper screens. If you need an I2C screen, you can use another I2C-supporting shield, or rewire to SDA / SCL.

