**Driving an E-Paper Display with the BW21-CBV-Kit**

### ****Introduction****

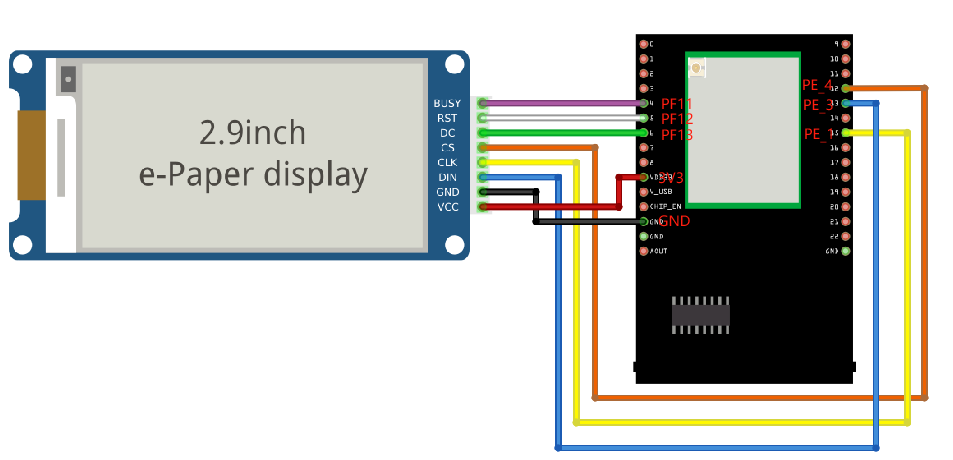
This project was created by **iiv**, a member of the Ai-Thinker community.

Hello, everyone! Today, I’m testing the **BW21-CBV-Kit** to drive a **2.9-inch e-paper display**. The development board comes with a rich set of examples, making it incredibly easy to use, especially when developing with Arduino—just plug and play!

**Hardware Setup**

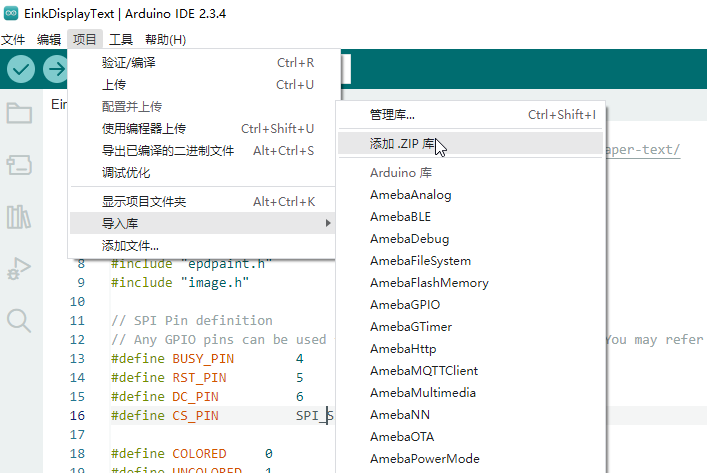


**Wiring & Connection**



Since the driver library for this e-paper display is not included by default in the examples, you’ll need to download it manually from GitHub：

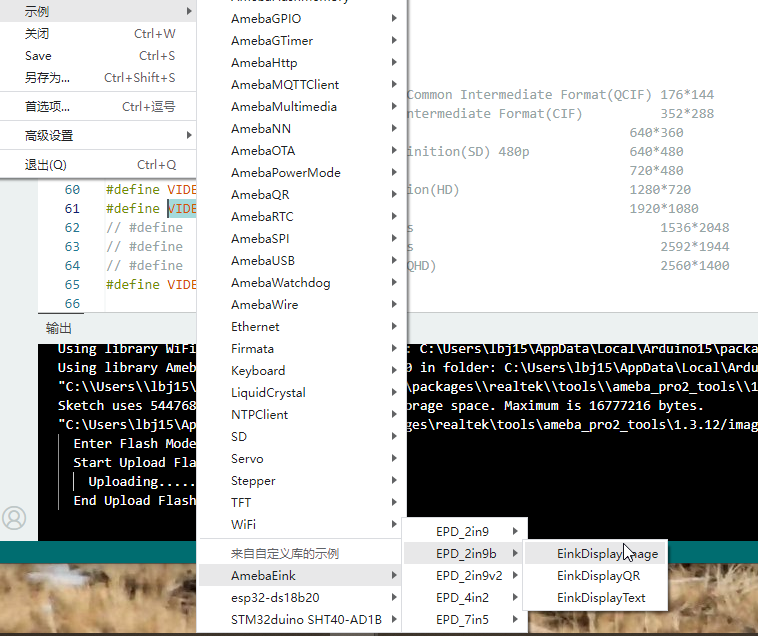
[AmebaEink.zip](https://bbs.ai-thinker.com/forum.php?mod=attachment&aid=MzAwMjZ8NGE0YzFkY2R8MTc0MDY0NDI2NXwxMjYxMHw0NTYyOA==" \t "https://mp.weixin.qq.com/_blank)



**Library Download Link:**

<https://github.com/Ameba-AIoT/ameba-arduino-pro2/tree/dev/Arduino_zip_libraries>

After downloading the library, import it as shown in the image. Once imported, you’ll find an example named **AmebaEink**, which supports **2.9-inch, 4.2-inch, and 7.5-inch e-paper displays**—very convenient!



The library provides sample code for displaying **images, QR codes, and text** on the e-paper display.

### ****Examples & Features****

**Displaying a QR Code**

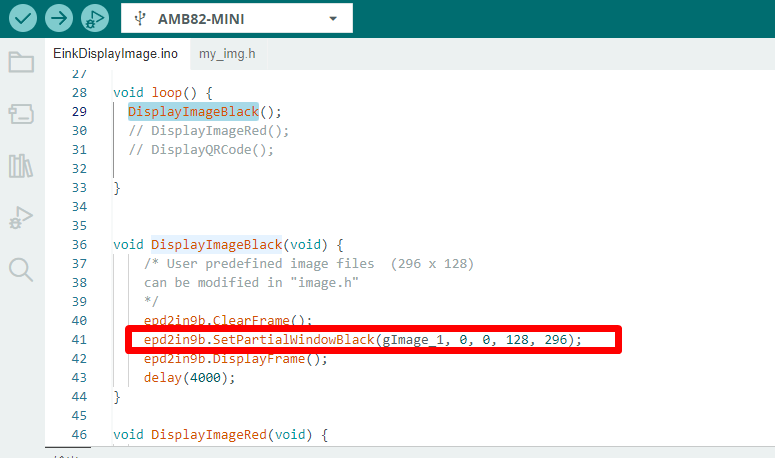


Open the QR code example and modify the URL you want to display.



Once the firmware is uploaded and the board is restarted, a QR code will appear on the screen. Simply scan it with your browser, and it will direct you to the URL you specified—super easy!

**Displaying an Image**



Open the image display example and modify the image you want to display.

Prepare a **296 × 128** image in advance, use an image-to-array conversion tool, and then import the generated array into the code.

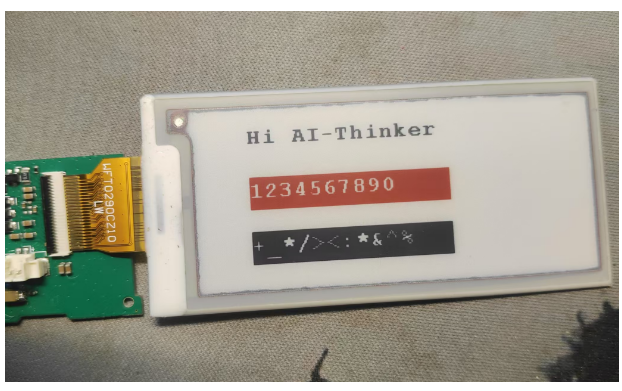


After uploading and restarting, the screen will refresh and display the image you prepared.

**Displaying Text**



Similarly, open the text display example. You’ll see several functions that allow you to display different text colors on a tri-color e-paper screen (black, white, red, or yellow). Simply choose the appropriate function based on your display type.



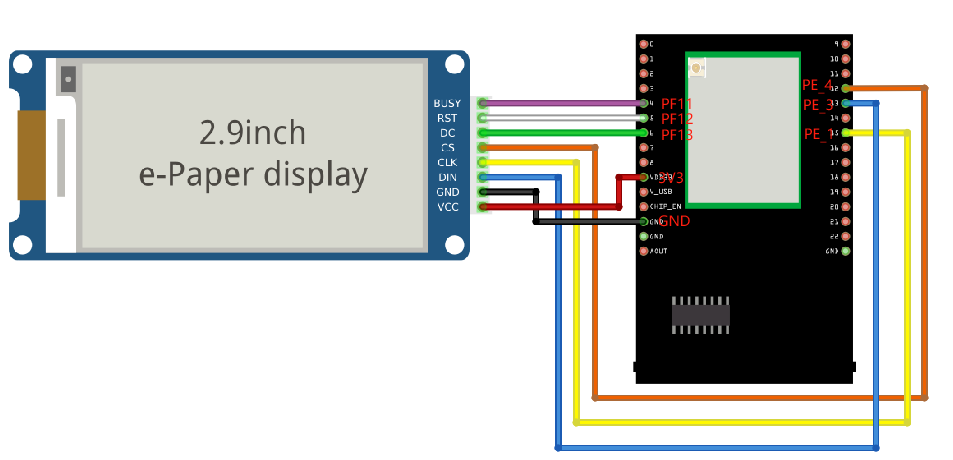
Once compiled, uploaded, and restarted, your desired text will appear on the screen—it’s that simple!

### ****Expanding to a 4.2-inch E-Paper Display****

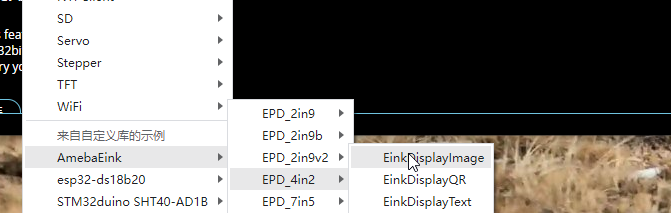
Since I also have a **4.2-inch e-paper display**, I decided to test it as well and explore its library functions. The **BW21-CBV-Kit** already includes support for this display, so you can use the same approach as before.



**Wiring & Connection**



Follow the same wiring method as the 2.9-inch display and open the 4.2-inch e-paper example from the provided library.



Since I already have experience with the **2.9-inch display**, I didn’t go through every single example again—the functions are very similar. Instead, I focused on **exploring the key display functions**.



### ****Basic Drawing Functions****

#### ****Drawing Lines****

*// Draw a straight line*

*// Parameters: x, y, target\_x, target\_y, color*

*paint.DrawLine(10, 10, 100, 100, COLOR\_BLACK);*

*// Draw a horizontal line*

*// Parameters: x, y, length, color*

*paint.DrawHorizontalLine(10, 10, 100, COLOR\_BLACK);*

*// Draw a vertical line*

*// Parameters: x, y, length, color*

*paint.DrawVerticalLine(10, 10, 100, COLOR\_BLACK);*

These three functions all **draw lines**, but with different approaches.

#### ****Drawing Rectangles****

*// Draw an outlined rectangle*

*// Parameters: x, y, target\_x, target\_y, color*

*paint.DrawRectangle(50, 50, 10, 10, COLOR\_BLACK);*

*// Draw a filled rectangle*

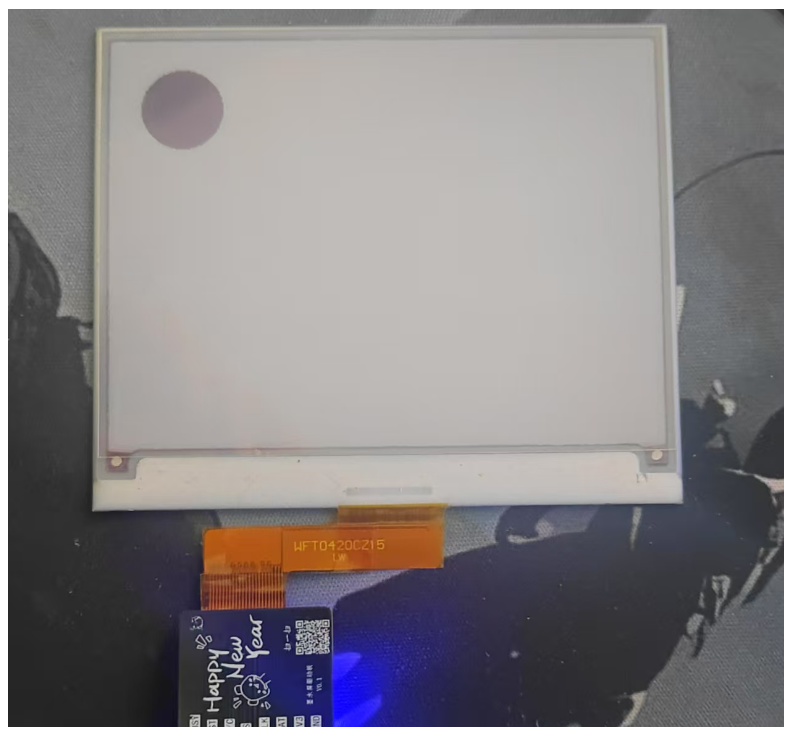
*// Parameters: x, y, target\_x, target\_y, color*

*paint.DrawFilledRectangle(10, 10, 30, 30, COLOR\_BLACK);*



The first function draws a **hollow rectangle**, while the second one draws a **filled rectangle**.

#### ****Drawing Circles****



*// Draw a circle*

*// Parameters: x, y, radius, color*

*paint.DrawCircle(10, 10, 30, COLOR\_BLACK);*

*// Draw a filled circle*

*// Parameters: x, y, radius, color*

*paint.DrawFilledCircle(50, 50, 30, COLOR\_BLACK);*

Similar to rectangles, the first function draws a **hollow circle**, and the second one draws a **filled circle**.

**Displaying Text**



*// Display a string*

*// Parameters: x, y, text, font, color*

*paint.DrawStringAt(20, 20, "HI Ai-Thinker!", &Font24, COLOR\_BLACK);*

*Additional Utility Functions*

*// Set screen orientation*

*// Parameters: rotation angle*

*paint.SetRotate(ROTATE\_0);*

*// Clear the screen*

*// Parameters: background color*

*paint.Clear(COLOR\_BLACK);*

*// Set canvas width and height*

*// Parameters: width, height*

*paint.SetWidth();*

*paint.SetHeight();*

### ****Conclusion****

Thanks to **pre-built libraries and ready-to-use examples**, getting started with the BW21-CBV-Kit is incredibly easy. In **Arduino**, you can **right-click on a function to jump directly to its implementation**, or open the source files in **VS Code** for more convenient editing.

With its **extensive example library, comprehensive tutorials, and seamless peripheral integration**, this module is **one of the best choices** for anyone working with e-paper displays and IoT projects.