ideaspark ESP32 1.9 inch LCD

ideaspark ESP32 Development Board with 1.9-inch TFT LCD Display

Model: ESP32 1.9 inch LCD

1. Introduction

The ideaspark ESP32 Development Board integrates an ESP32-WROOM-32 module with a 1.9-inch ST7789 TFT LCD display. This board offers WiFi and Bluetooth wireless connectivity, 16MB Flash memory, and a USB Type-C interface for power and data. Designed for ease of use, it eliminates the need for complex wiring between the ESP32 module and the display, making it suitable for various Internet of Things (IoT) projects, including data monitoring, weather stations, and graphic plotters.



Figure 1: The ideaspark ESP32 Development Board with integrated 1.9-inch TFT LCD display.

2. Features

- **Integrated 1.9-inch TFT LCD:** Features a 170x320 resolution full-color display with ST7789 driver, compatible with I2C interfaces.
- **ESP32-WROOM-32 Module:** Incorporates 2.4 GHz Wi-Fi and BLE dual-mode chip with 16MB Flash memory.
- **USB Type-C Connectivity:** Utilizes a Type-C USB cable for power and data connection.
- **CH340 Driver:** Requires CH340 driver for USB communication.
- **Versatile Application:** Ideal for IoT projects, displaying network status, sensor data, power levels, Internet Weather Stations, Graphic Plotters, and Data Monitors.
- **Compact Design:** The entire solution occupies minimal PCB area due to integrated components.



Figure 2: The board integrates an ESP32 module with a 1.9-inch LCD display.

3. Package Contents

The product package includes:

- 1 x ideaspark ESP32 1.9 inch LCD Development Board

4. Setup Guide

4.1. CH340 Driver Installation

The board uses a CH340 chip for USB-to-serial communication. Before using the board, you must install the appropriate driver for your operating system. For Windows 10, search for "CH340 Driver" on a search engine like Google.com or Bing.com and follow the provided installation instructions.

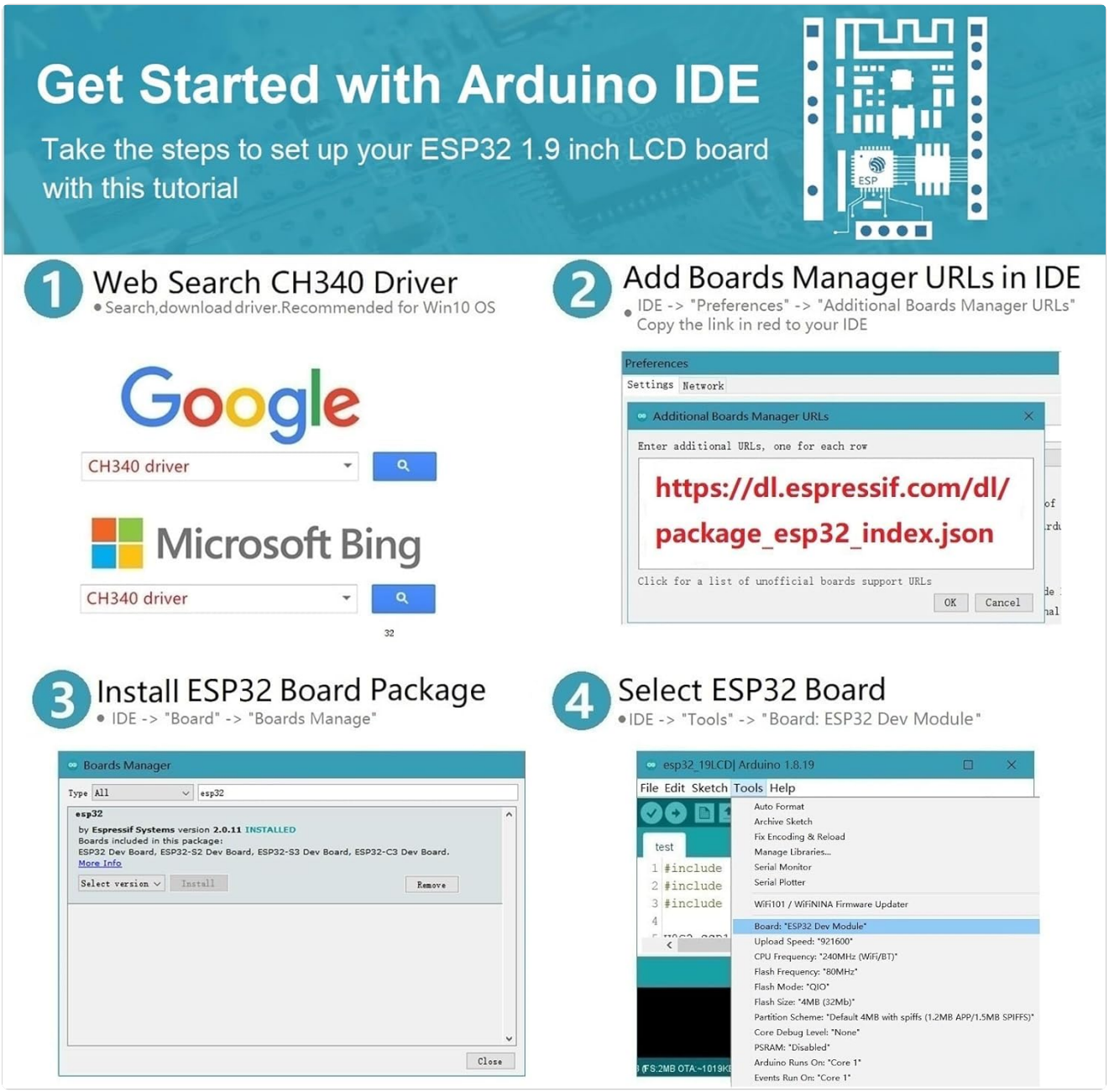


Figure 3: Step 1 of Arduino IDE setup: Searching for CH340 driver.

4.2. Arduino IDE Configuration

To program the ESP32 board using the Arduino IDE, follow these steps:

1. **Add Boards Manager URL:** In the Arduino IDE, go to *File > Preferences*. In the "Additional Boards Manager

URLs" field, add the following URL: `https://dl.espressif.com/dl/package_esp32_index.json`

2. **Install ESP32 Board Package:** Go to *Tools > Board > Boards Manager*. Search for "esp32" and install the "ESP32 by Espressif Systems" package.
3. **Select ESP32 Board:** After installation, go to *Tools > Board > ESP32 Arduino* and select "ESP32 Dev Module".
4. **Install ST7789 Library:** Go to *Sketch > Include Library > Manage Libraries*. Search for "Adafruit ST7789" and install the library. Ensure all related dependencies are also installed.
5. **Configure Board Settings:** Under *Tools*, configure settings such as Upload Speed, Flash Mode, and Partition Scheme as appropriate for your project.

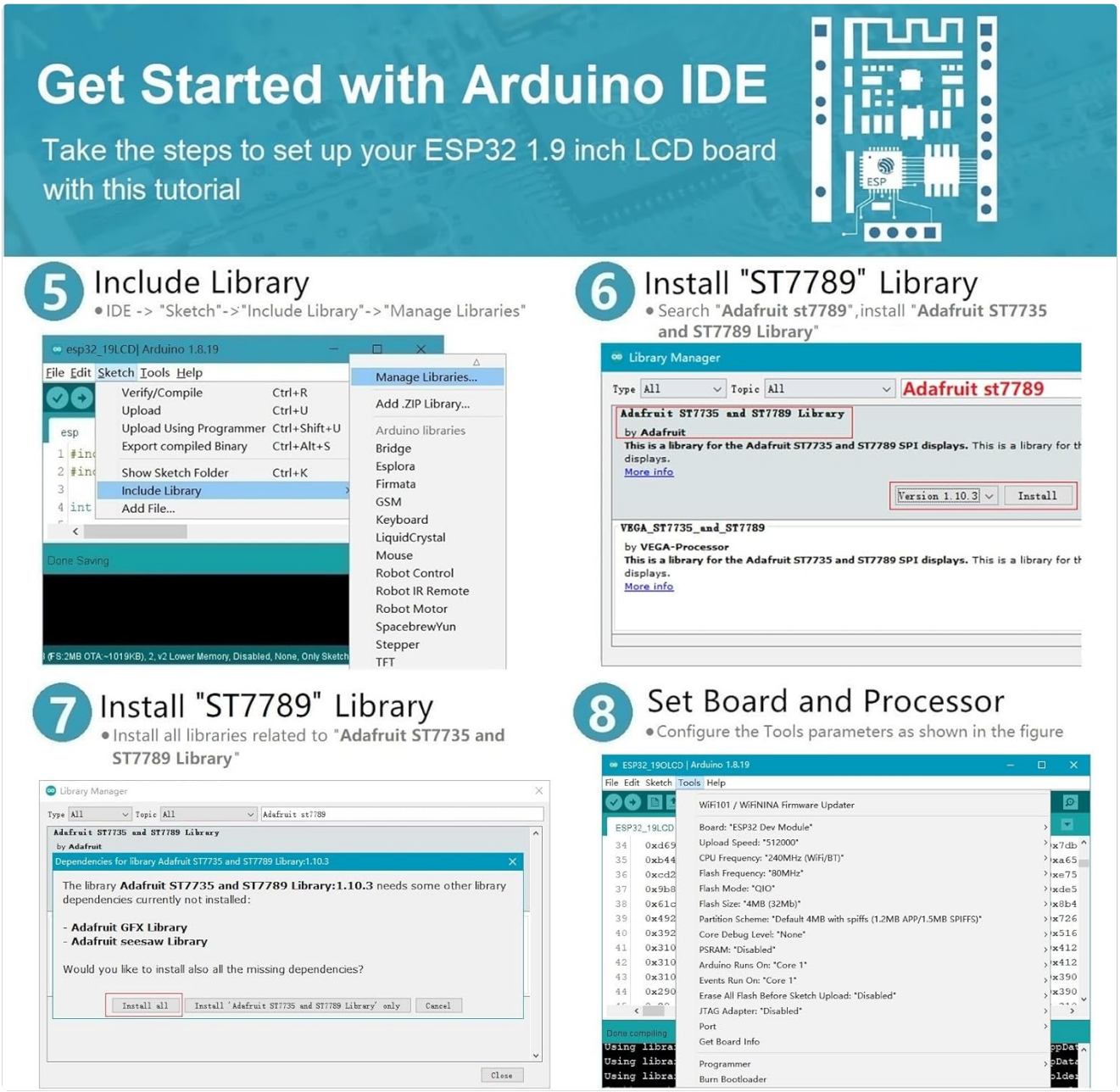


Figure 4: Steps 5-8 of Arduino IDE setup for libraries and board configuration.

4.3. Pinout Diagram

Understanding the pinout is crucial for connecting external components. The following diagram illustrates the pin assignments for the ideaspark ESP32 1.9 inch LCD Development Board.

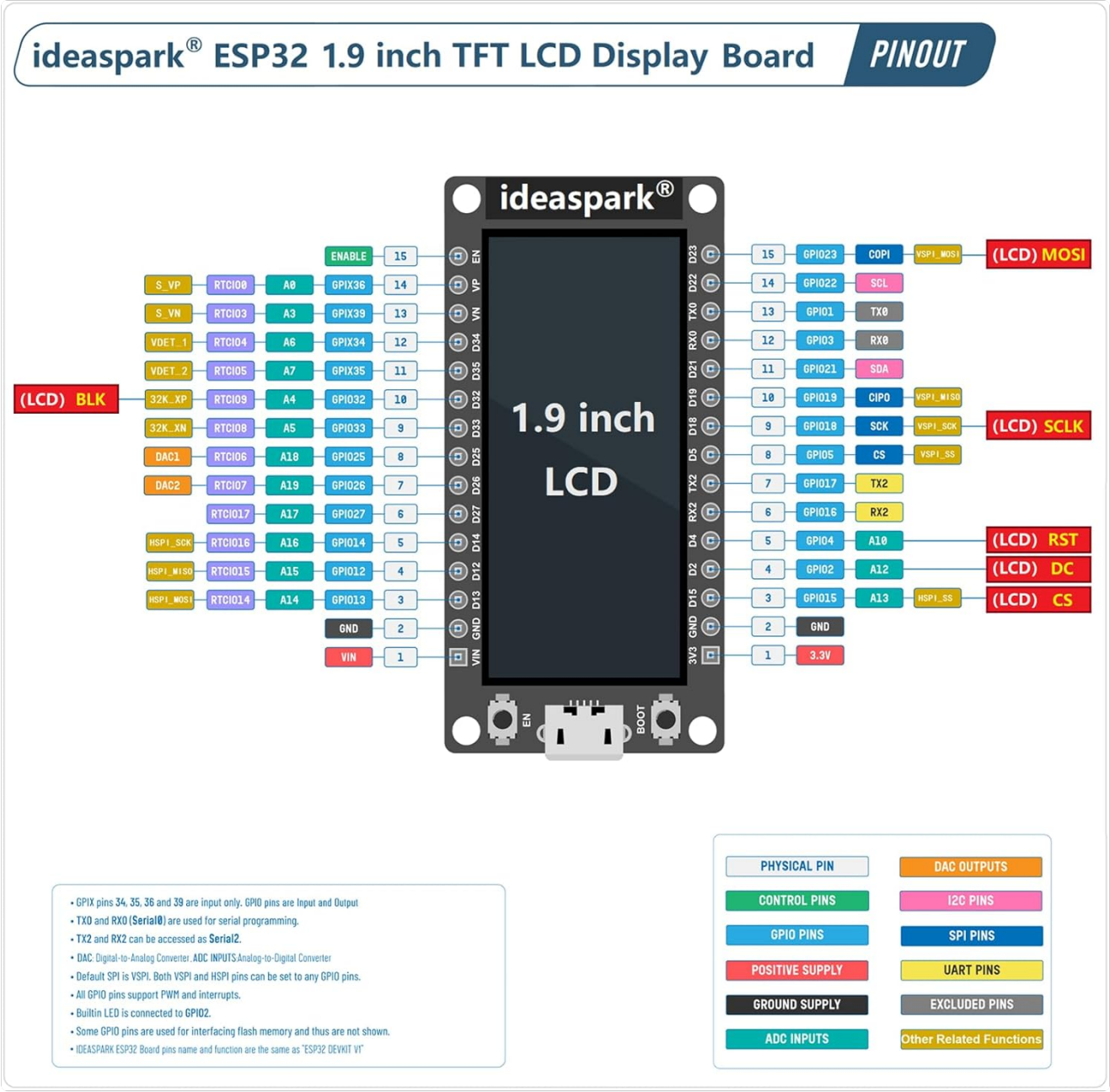


Figure 5: Pinout diagram of the ESP32 1.9 inch LCD Development Board.

Key Pin Assignments for LCD (SPI):

- **MOSI:** D23 / GPIO23
- **SCLK:** D18 / GPIO18
- **CS:** D15 / GPIO15
- **DC:** D2 / GPIO2
- **RST:** D4 / GPIO4
- **BLK (Backlight):** D32 / GPIO32

These pins are pre-wired for the integrated LCD, simplifying display integration.

5. Operating Instructions

5.1. Basic Display Operation

The integrated LCD uses SPI for communication. You can use libraries like Adafruit_GFX and Adafruit_ST7789 to

easily control the display. The pre-defined pin connections simplify programming.

5.2. Sample Code

A basic example to initialize the display and print text is provided below. This code demonstrates how to set up the display and output information.

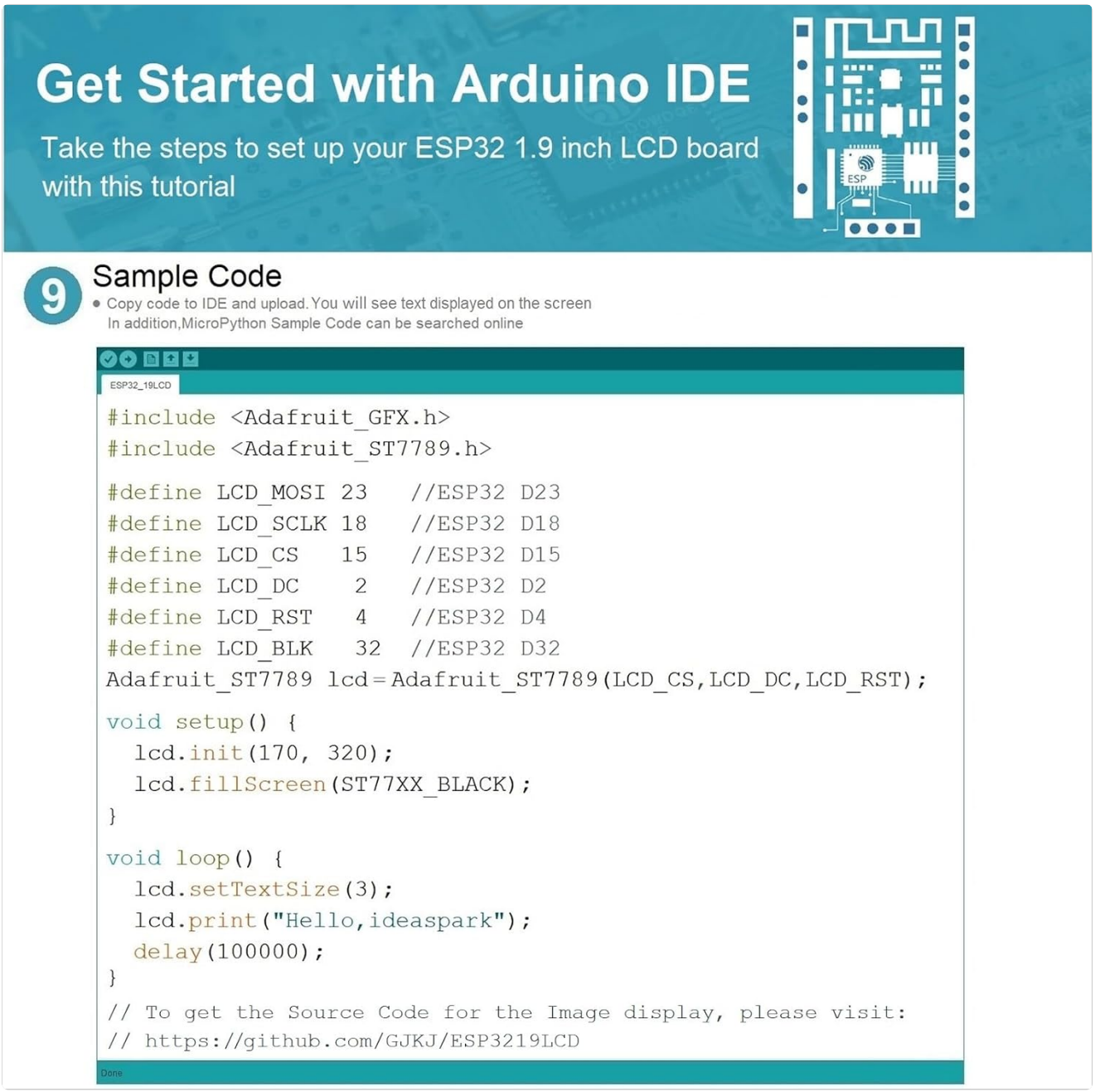


Figure 6: Sample Arduino code for basic display functionality.

```
#include <Adafruit_GFX.h>
#include <Adafruit_ST7789.h>

#define LCD_MOSI 23 // ESP32 D23
#define LCD_SCLK 18 // ESP32 D18
#define LCD_CS   15 // ESP32 D15
#define LCD_DC    2 // ESP32 D2
#define LCD_RST   4 // ESP32 D4
#define LCD_BLK  32 // ESP32 D32

Adafruit_ST7789 tft = Adafruit_ST7789(LCD_CS, LCD_DC, LCD_RST);

void setup() {
  tft.init(170, 320);
  tft.fillScreen(ST77XX_BLACK);

  tft.setTextColor(TFT_YELLOW, TFT_BLACK);
  tft.setTextSize(2);
  tft.setCursor(10, 10);
  tft.println("Hello, ideaspark");
  delay(10000);
}

void loop() {
  // Your main program loop
}
```

For more advanced examples and source code for image display, refer to online resources or the manufacturer's GitHub repository.

5.3. WiFi and Display Coexistence (Important)

Due to shared GPIO pins, specifically GPIO2 (D2) for both WiFi and the display's DC line, a specific programming sequence is recommended to prevent conflicts, display lock-ups, or constant reboots. It is crucial to initialize and connect to WiFi completely before performing display operations.

Recommended Code Structure:

```
//  WiFi FIRST (CRITICAL for your board)
WiFi.mode(WIFI_STA);
WiFi.setSleep(false);
WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println();
Serial.println("WiFi Connected");

//  Lock GPIO2 AFTER WiFi is done with it
pinMode(2, OUTPUT);
digitalWrite(2, HIGH);

//  Backlight
pinMode(32, OUTPUT);
digitalWrite(32, HIGH);

//  Display init AFTER WiFi
tft.init();
tft.setRotation(1);
tft.fillScreen(TFT_BLACK);

tft.setTextColor(TFT_YELLOW, TFT_BLACK);
tft.setTextSize(2);
tft.setCursor(10, 10);
tft.println("WiFi OK");
tft.println(WiFi.localIP());
```

This sequence ensures that GPIO2 is stabilized after WiFi initialization, preventing interference with the display's data/command line. Some users have also found that adding a 10K Ohm pull-up resistor between the 3.3V pin and D2 can help stabilize the line during WiFi startup, though the software workaround is often sufficient.

6. Maintenance

To ensure the longevity and proper functioning of your ideaspark ESP32 Development Board, observe the following maintenance guidelines:

- **Handle with Care:** Avoid dropping the board or subjecting it to excessive physical stress.
- **Keep Dry:** Protect the board from moisture and liquids.
- **Clean Gently:** If cleaning is necessary, use a soft, dry cloth. Avoid harsh chemicals.
- **Static Discharge:** Take precautions against electrostatic discharge (ESD) when handling the board.
- **Proper Storage:** Store the board in a cool, dry environment away from direct sunlight and extreme temperatures.

7. Troubleshooting

If you encounter issues with your ideaspark ESP32 Development Board, consider the following troubleshooting steps:

- **CH340 Driver Not Recognized:** Ensure the CH340 driver is correctly installed for your operating system. Try

reinstalling the latest version.

- **WiFi and Display Conflicts:** If the display freezes or the board reboots when WiFi is active, refer to Section 5.3 for the recommended software workaround to initialize WiFi before display operations.
- **Display Not Working/Garbled Output:**
  - Verify that the correct ESP32 board and ST7789 library are installed and selected in the Arduino IDE.
  - Check your code for correct pin definitions and display initialization parameters (e.g., `tft.init(170, 320)`).
  - Inspect for any physical damage or poor solder joints, particularly around the display connections.
- **I2C Sensor Conflicts:** If you experience issues with I2C sensors when the display is active, be aware that some boards might have manufacturing variations or shared pin issues (e.g., GPIO22 and GPIO23 being connected). Try using alternative I2C pins if available, or carefully inspect the board for unintended shorts.
- **Power Issues (Battery/USB):** Note that display brightness may vary between battery (3.7V) and USB (5V) power. If using a battery, ensure it is properly charged and connected. The board may not include an integrated battery charging circuit or voltage booster, so external solutions might be required for battery charging or consistent display brightness.
- **Board Not Responding:** Ensure the USB Type-C cable is functional and properly connected. Some users report issues with USB-C to USB-C cables, suggesting USB-A to USB-C might be more reliable for power.

8. Specifications

Feature	Specification
Brand	ideaspark
Model Name	ESP32 Development Board
Item Model Number	ESP32 1.9 inch LCD(Solder PIN)
Processor Brand	Espressif
Number of Processors	1
RAM	LPDDR
Memory Storage Capacity	16 MB
Wireless Type	802.11ac (Wi-Fi), Bluetooth Low Energy (BLE)
Connectivity Technology	USB Type-C
Operating System	FreeRTOS
Display Size	1.9 inches
Display Resolution	170x320
Display Driver	ST7789
Item Weight	1.13 ounces (approx. 32g)
Product Dimensions (LxWxH)	2.83 x 2.05 x 0.63 inches (approx. 7.19 x 5.21 x 1.60 cm)

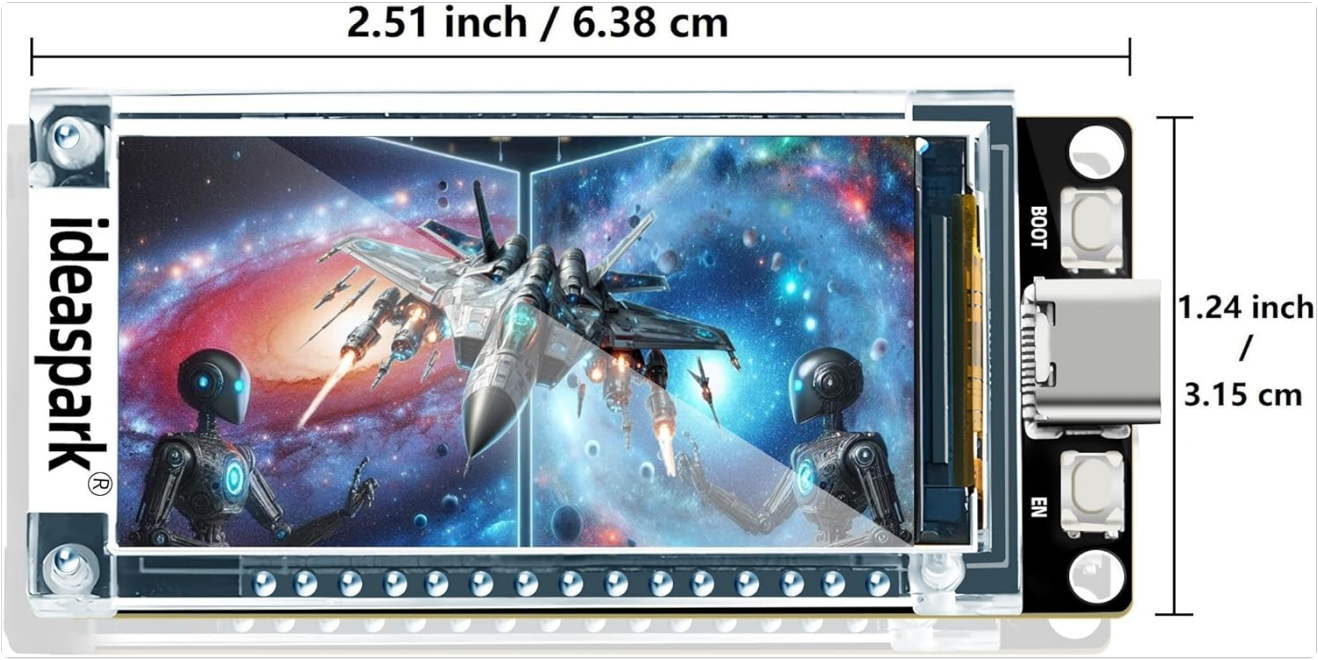


Figure 7: Dimensions of the ideaspark ESP32 Development Board.

9. Warranty and Support

For warranty information, technical support, or further assistance, please refer to the product packaging or contact ideaspark customer service through their official channels. Keep your purchase receipt for warranty claims.

Ask a question about this manual

Ask about setup, troubleshooting, compatibility, parts, safety, or missing instructions. Manuals+ will review the question and use this page's manual context to help answer it.

Name

Anonymous

Email

you@example.com

After a successful submission, we securely hash the supplied account or form email before sending it to Microsoft Advertising for conversion attribution. Privacy details.

Question

Example: How do I reset this device or fix this error?

Details

Model number, symptoms, what you tried, or the section of the manual you are using.

Submit question

Answered questions about this manual

On the ideaspark ESP32 1.9 inch LCD board, what is the physical button on the rear labeled KEY21 connected to?

On the ideaspark ESP32 1.9 inch LCD development board, the physical button labeled KEY21 is connected to GPIO 21 of the ESP32 module.While the primary manual focuses on the display pins (GPIO 23, 18,...

Read full answer

Can the LCD display on the ideaspark ESP32 1.9-inch development board be controlled using MicroPython?

Yes, you can control the display using MicroPython, although the provided manual focuses primarily on Arduino (C++) examples. Here are the specific details regarding your technical questions:LCD Controller and Interface: The board uses the...

Read full answer