ESP32-3248S035C 3.5" Display Module – Technical Documentation & Setup Guide

Overview & Specifications

The **ESP32-3248S035C** (sold by DIYmalls) is a 3.5-inch TFT touchscreen module that integrates an ESP32 microcontroller with a 320×480 color display. It features an **ESP32-D0WDQ6** dual-core MCU (the standard ESP32, 240 MHz), with **4 MB flash and no PSRAM** on this modelopenhasp.com. The display uses the **ST7796** driver (65K color TFT, 480x320 resolution) and in this "C" variant it has a **capacitive touch panel** (Goodix **GT911** controller)community.home-assistant.ioamazon.com. Key hardware features include:

- **3.5" TFT LCD** (ST7796 driver) at 320×480 resolution, 16-bit color depthamazon.com. The active display area is ~73.4×48.9 mm, and the module PCB measures about 101.5×54.9 mmamazon.com.
- **Touchscreen:** Capacitive 5-point touch (GT911 controller) on the C model; a resistive touch with XPT2046 controller is used on the R model. (Ensure you have the correct model; C = capacitive, R = resistiveopenhasp.com.) The capacitive version comes pre-calibrated, while resistive requires calibration on first useforum.arduino.cc.
- **ESP32 MCU:** Built-in WiFi (802.11 b/g/n) and Bluetooth, with a PCB antenna on board for wireless connectivity. It supports the Arduino IDE and other ESP32 development frameworks out of the box.
- **Storage:** MicroSD **TF card slot** for data storage or firmware update. (The SD slot is present on the boardamazon.com. On this model, it's wired to the ESP32's SDIO/SPI interface; see pinout section. Note that some firmware like openHASP did not yet support the SD slot as of v0.7openhasp.com.)
- **Audio:** Onboard audio amplifier (e.g. SC8002) with a 2-pin speaker header. The ESP32's DAC/GPIO can drive this amp (GPIO 26 is used for audio output or amp shutdown control)community.home-assistant.io. *Note:* Use a proper series resistor or 8Ω+ speaker to avoid overloading the ampcommunity.home-assistant.io.
- **RGB LED:** A multi-color indicator LED on the board, with red, green, blue channels tied to ESP32 GPIO4, GPIO16, and GPIO17 respectivelycommunity.home-assistant.io. These can be used for status indication.
- **Power & Programming:** Powered via a Micro-USB port (5V input regulated to 3.3V on board). The module has a USB-UART for programming (supports auto-boot/flash). It also breaks out many ESP32 GPIOs on a header for expansion. There is a BOOT/EN button (for flashing mode) and the typical ESP32 auto-reset circuit.

Overall, this is a "**smart display**" module aimed at IoT interfaces and home automation control panels. It is sometimes referred to as a "Sunton 3.5-inch ESP32 display" or the "cheap yellow display" in community forumsamazon.comgithub.com. Despite the rich feature set (touch, SD, audio, etc.), note that the lack of PSRAM means memory is limited for graphics buffersopenhasp.com. Complex GUIs (e.g. with LVGL) will work, but may require smaller buffers or lower color depth to fit into RAM.

Pin Mapping & Internal Wiring

One challenge with this module has been the lack of official pinouts in the documentation. However, through community efforts and the available schematics, the **internal wiring** of the ESP32-3248S035C has been mapped as follows:

- **TFT Display (ST7796)** – The LCD is controlled via **SPI interface** (4-wire SPI mode, 16-bit color). The ESP32's VSPI bus is used:
    - **MOSI** – GPIO 13
    - **MISO** – GPIO 12 *(Note: MISO is not needed for basic drawing, but the ST7796 does support reading data; this line is shared with the touch controller and SD card)*
    - **SCLK** – GPIO 14
    - **TFT CS** (Chip Select) – GPIO 15
    - **TFT DC** (Data/Command select) – GPIO 2
    - **TFT RESET** – Not connected to a separate GPIO. The LCD's reset line is tied to the ESP32 reset (EN) so it reboots with the ESP32 (you can define TFT_RST -1 in code)github.com.
    - **Backlight Control** – GPIO 27 (controls a MOSFET for LCD LED power)community.home-assistant.io. This pin can be used for dimming (PWM) or turning the screen on/off. A 3.3V-driven MOSFET (Q2) on the board switches the LED supplyopenhasp.com.

Note: The ST7796 driver supports both SPI and parallel interfaces. In this design it is configured for SPI (to save pins). The TFT shares the SPI bus with the touch controller (resistive version) and possibly the SD card. Maximum SPI clock of 40 MHz is supported for the ST7796 on ESP32community.home-assistant.io.

- **Capacitive Touch (GT911)** – The GT911 touch controller (often integrated on the touch panel's flex cable) communicates via **I²C**:
    - **SDA** – GPIO 33
    - **SCL** – GPIO 32
    - **INT** – GPIO 21 (interrupt pin to signal touch events)
    - **RST** – GPIO 25 (reset pin for the touch controller)
    - **I²C Address:** 0x5D (7-bit) by default for GT911.
      These connections are fixed on the capacitive variantcommunity.home-assistant.io. The GT911 comes pre-calibrated for the 3.5" screen, so no manual alignment is neededforum.arduino.cc.
- **Resistive Touch (XPT2046)** – *[For reference – not present on the "C" model]* The resistive version adds an XPT2046 touch ADC, which is connected to the same VSPI bus as the TFT:
    - **Touch CS** – GPIO 33
    - **Touch IRQ (Pen interrupt)** – GPIO 36

    - **Touch SPI** (SCLK/MOSI/MISO) – shared with TFT (GPIO14, 13, 12)community.home-assistant.io.
    - The XPT2046 (a.k.a. T_TOUCH) reads the 4-wire resistive panel (X+, X-, Y+, Y-) and needs calibration on first use. (This chip is not present on capacitive models, and GPIO33 is repurposed for I2C SDA in that case.)

- **MicroSD Card Slot:** Present on the board (TF slot). In the official schematic, the SD card is wired to the ESP32's SDIO interface (4-bit mode) using the following pins: **GPIO 14** (CLK), **GPIO 15** (CMD), **GPIO 2** (D0), **GPIO 4** (D1), **GPIO 12** (D2), **GPIO 13** (D3). These are the ESP32's HS2 SDMMC pins, which means the SD is intended to use the native SD host controller (not just SPI)openhasp.com. **Important:** GPIOs 2, 4, 12, 15 are strapping pins on ESP32; the board includes proper pull-ups/pull-downs so that using the SD card doesn't interfere with boot mode. In Arduino, you can use the SD library (with SD.begin() on those pins) to access the card. *(If using the SD in SPI mode instead, you would use GPIO14/13/12 and one of the CS pins, but the provided design is for SDIO.)*

- **RGB Status LED:** There is an onboard RGB LED (often near the top edge of the board). Its pins are: **Red = GPIO 4**, **Green = GPIO 16**, **Blue = GPIO 17**community.home-assistant.io. These LEDs are active HIGH (driven directly by the ESP32 GPIOs). You can use PWM signals on those pins to adjust brightness for each color.

- **Audio Amplifier:** A small audio amp chip (e.g. SC8002) is on board, with a 2-pin JST speaker connector. The amplifier's input and enable are connected to **GPIO 26**community.home-assistant.io. In the schematic, GPIO26 feeds the amp (likely as the DAC output for audio, or as a shutdown pin via a resistor network). In practice, to use sound you can attach a speaker (8Ω with a series resistor, or a 4Ω with careful volume control) and output analog audio on GPIO26 (e.g. using DAC1 or I2S DAC output). **Caution:** The amplifier is powered from 3.3V, so driving a low-impedance speaker at high volume can brown out or damage it. It's recommended to use at least an 8Ω speaker and moderate volumecommunity.home-assistant.io.

- **Available GPIOs:** Many of the ESP32's GPIOs are exposed on a header/connector for expansion. However, note that **a lot of pins are pre-assigned** to the display and peripherals as listed above. Usable free pins on this module (for your own sensors, etc.) include, for example: GPIO 0, 5, 18, 19, 22, 23, 34, 35, 39 (and 36/37 if not used by touch in your model). Always double-check if a pin is not already in use: for instance, GPIO0 and GPIO5 are broken out and can be used for outputs/inputs (keeping in mind GPIO0's boot mode behavior), and GPIO22/23 are free (they are not used internally, and correspond to I2C pins on many ESP32 boards – on this module they might be available on a connector labeled for I^2C). According to one source, the board even provides multiple 4-pin JST connectors for peripherals (e.g. labeled for **I2C, UART, SPI, etc.**), making it easier to plug in sensorsforum.arduino.ccforum.arduino.cc. Always refer to any silkscreen or documentation for the pin header to utilize these correctly.

Arduino IDE Setup

You can program the ESP32-3248S035C using the standard ESP32 Arduino environment. Here's how to get started:

- **Board Definition:** In Arduino IDE, install the ESP32 boards package (via Boards Manager) if not already installed. Select **"ESP32 Dev Module"** as the boardm.media-amazon.comm.media-amazon.com. This module uses the typical ESP32-WROOM module, so the generic Dev Module settings work. Leave flash size at "4MB (32Mb)" and Flash Mode "QIO" (default). You can use standard upload speed (921600 baud) for flashing via USB.
- **Drivers & USB:** The device should enumerate as a serial COM port when plugged in (it uses an onboard USB-to-UART converter). Install any necessary drivers (e.g. CH340 or CP210x driver, depending on the USB chip on the board) if the port is not recognized.
- **Power Considerations:** Ensure the board is powered with a stable 5V via USB. The 3.5" TFT with WiFi can draw significant current (especially with backlight on and WiFi transmitting). If you encounter instability or WiFi connection issues, use a high-quality USB cable or power the board's 5V pin directly from a reliable 5V source. Some users reported improved WiFi reliability by **bypassing the USB port and powering the board via the 5V/Vin pin** (this can provide a bit more stable voltage under load)community.home-assistant.iocommunity.home-assistant.io. In other words, if WiFi is "flaky," the issue is often insufficient power – the ESP32's radio is sensitive to voltage drop. Using a short USB cable and a 5V 1–2A supply (or powering VIN directly) can resolve connectivity problems.
- **Arduino Libraries:** Install the necessary libraries for display and touch. We highly recommend using **Bodmer's TFT_eSPI** library for the ST7796 display, as it's optimized for ESP32. For the capacitive touch (GT911), you can use the **TAMC_GT911** Arduino library (or similar GT911 libraries) to read touch input. Both are available via the Library Manager:
    - **TFT_eSPI** by Bodmer – for the TFT display graphics.
    - **TAMC_GT911** (by TAMCTec) – for the GT911 touch controllerforum.arduino.cc.

Optionally, if you plan to create advanced GUIs, you might use **LVGL** (Light and Versatile Graphics Library) along with TFT_eSPI or LovyanGFX. The manufacturer's demo code uses LVGL 7.x with TFT_eSPI, but this adds complexity; you can start with simpler examples first.

- **Pin Configuration (TFT_eSPI):** After installing TFT_eSPI, you must configure it for this module's pin connections. This involves editing the User_Setup.h or using a custom setup file. You can find User_Setup.h in the TFT_eSPI library folder (or use the **User_Setup_Select.h** to point to a custom config). Set the driver and pins as follows (this matches the board's wiring):

```
// User_Setup for ESP32-3248S035C (3.5" ST7796 SPI TFT)

#define ST7796_DRIVER      // Enable ST7796 display driver (320x480 TFT)


#define TFT_WIDTH  320

#define TFT_HEIGHT 480


// Define the ESP32 pins used for the TFT interface
```

```
#define TFT_MOSI 13   // SPI MOSI (TXD)

#define TFT_MISO 12   // SPI MISO (RXD)

#define TFT_SCLK 14   // SPI Clock

#define TFT_CS   15   // TFT CS pin

#define TFT_DC   2    // TFT D/C pin

#define TFT_RST  -1   // TFT reset (not used, tied to ESP32 EN)

#define TFT_BL   27   // TFT backlight control pin


// Touch (XPT2046) – not present in capacitive version, but define if needed:

// #define TOUCH_CS 33    // XPT2046 chip select (resistive touch only)

// #define TOUCH_IRQ 36   // XPT2046 touch interrupt (resistive touch only)
```

These definitions are derived from the module's schematic and have been confirmed by community membersgithub.com. With this setup, you can call tft.begin() (for a TFT_eSPI TFT_eSPI tft instance) in your Arduino sketch to initialize the display. TFT_eSPI will drive the ST7796 in SPI mode. The backlight pin (GPIO27) can be controlled by calling tft.setBrightness(val) if you have configured TFT_eSPI for PWM backlight, or simply digitalWrite to turn it on/off (remember on this board, HIGH turns the backlight **on** via the MOSFET).

- **Touch Configuration (Arduino):** For the capacitive touch, TFT_eSPI does *not* handle GT911 directly (it only has built-in support for XPT2046 resistive touch if configured). Instead, use the GT911 library. With **TAMC_GT911**, initialization is straightforward, as shown below:

```
#include <TAMC_GT911.h>

#define TOUCH_SDA 33

#define TOUCH_SCL 32

#define TOUCH_INT 21

#define TOUCH_RST 25

#define TOUCH_WIDTH  320   // touch active area width (pixels)

#define TOUCH_HEIGHT 480   // touch active area height


TAMC_GT911 touch = TAMC_GT911(TOUCH_SDA, TOUCH_SCL, TOUCH_INT, TOUCH_RST,
TOUCH_WIDTH, TOUCH_HEIGHT);


void setup() {
```

```
  Wire.begin(TOUCH_SDA, TOUCH_SCL);      // Initialize I2C for touch
  touch.begin();                         // Initialize GT911 touch controller
  touch.setRotation(ROTATION_NORMAL);    // Orient touch coordinates (if needed)
  ...
}
```

This uses the known pin assignments (SDA=33, SCL=32, INT=21, RST=25)forum.arduino.cc. After calling touch.begin(), you can periodically call touch.read() to poll for touches. The library provides touch.isTouched flag and an array touch.points with touch coordinatesforum.arduino.cc. For example:

```
touch.read();
if (touch.isTouched) {
  uint16_t x = touch.points[0].x;
  uint16_t y = touch.points[0].y;
  Serial.printf("Touch at (%d, %d)\n", x, y);
}
```

The GT911 supports multi-touch, but you'll typically just use the first touch point (points[0]). Be aware of rotation: if you use tft.setRotation() on the display, you may need to adjust the touch orientation with touch.setRotation() accordingly so that the X/Y match the screen's rotation. By default, ROTATION_NORMAL in the TAMC library assumes (0,0) is top-left.

- **WiFi and Bluetooth:** Enabling WiFi on the ESP32 is done in software as usual. There are no special hardware steps required – just include the WiFi library and use WiFi.begin(ssid, pass). If you run into issues where WiFi won't connect, double-check the power as noted above. The antenna is on-board; make sure you don't have it covered by metal or your hand when testing range. In general, the ESP32 on this board behaves like any dev kit for WiFi/BT. The Bluetooth (Classic and BLE) can also be used with the appropriate libraries (though BLE usage alongside 2.4GHz WiFi can be memory-heavy given no PSRAM).

Example: Display and Touch in Action

Below is a simple example sketch that draws something on the screen and reads touch input. This assumes you have configured TFT_eSPI as described and installed the GT911 library:

```
#include <TFT_eSPI.h>
#include <TAMC_GT911.h>

TFT_eSPI tft = TFT_eSPI();
TAMC_GT911 touch(33, 32, 21, 25, 320, 480);
```

```
void setup() {
  Serial.begin(115200);
  tft.begin();
  tft.setRotation(1);       // e.g. rotate display if needed (0-3)
  tft.fillScreen(TFT_BLACK);
  tft.drawString("Hello ESP32-3248S035C!", 50, 100);  // example text

  Wire.begin(33, 32);
  touch.begin();
  touch.setRotation(ROTATION_NORMAL);  // adjust if using tft.setRotation(...)
}

void loop() {
  // Example: draw a circle at touch location
  touch.read();
  if(touch.isTouched) {
    uint16_t x = touch.points[0].x;
    uint16_t y = touch.points[0].y;
    tft.fillCircle(x, y, 5, TFT_RED);
    Serial.printf("Touch at (%d,%d)\n", x, y);
  }
}
```

This sketch initializes the display and touch, prints a greeting, and then whenever the screen is touched, it draws a small red circle at the touch location and prints the coordinates. This can help verify that both the TFT and touch are working properly.

Troubleshooting: If you see a blank screen: double-check the User_Setup.h configuration (CS, DC pins especially) – a wrong pin define will result in no output. If the serial console shows touch coordinates but they seem inverted or swapped, use touch.setRotation(...) or swap X/Y in code as needed. For example, some users found they needed to swap axes for correct orientation[forum.arduino.cc](forum.arduino.cc) – this depends on the display rotation. You can experiment to align the touch to the display coordinates.

Enabling WiFi and Known Issues

Using WiFi on this board should be straightforward, as it uses a standard ESP32 WiFi/BT module. Start by including the WiFi library and connecting to your network in setup(). For instance:

```
#include <WiFi.h>

...

WiFi.begin("your-ssid", "your-password");

while(WiFi.status() != WL_CONNECTED) {

  delay(100);

}

Serial.println("WiFi Connected, IP = " + WiFi.localIP().toString());
```

If you encounter issues with WiFi (e.g. it resets, fails to connect, or is unstable), consider the following fixes:

- **Power Supply:** As mentioned, ensure the board gets enough power. The WiFi radio can draw short bursts of ~300 mA. If powered via a PC USB port, make sure the port can supply this. If you see the board resetting when WiFi begins, it's likely a brown-out. In that case, use a different USB port or an external 5V 1A supply. Users have reported that powering the board through the 5V pin (bypassing the USB cable/drop) solved WiFi reliability problemscommunity.home-assistant.iocommunity.home-assistant.io.
- **Antenna Placement:** The antenna is a printed trace on the PCB (usually at the top edge of the board). Keep that end of the board away from ground planes or metal. For example, don't press the board flat against a metal surface when using WiFi – it will degrade the signal. In free air, the board's WiFi should perform similarly to a normal ESP32 dev kit.
- **GPIO Interference:** Avoid using high-frequency signals or heavy loads on GPIOs 0, 2, 15 during WiFi operation – those are strapping pins and can sometimes interfere if pulled in the wrong direction at boot. The board design already accounts for this with resistors, but it's good practice not to re-purpose those pins for other functions that fight the built-in pulls. (Typically, you won't need to touch those pins since they're used internally for TFT/SD as noted.)
- **Memory Constraints:** With 4MB flash and no PSRAM, the ESP32 has limited RAM for buffers. Large libraries (like LVGL or running WiFi + TFT updates concurrently) can approach the memory limits. If using **WiFi alongside graphics**, be mindful of memory usage. For example, opening large images from SD to RAM or using very large draw buffers can fail. In ESP32 Arduino you might see "Guru Meditation" errors if out of memory. A tip is to use **smaller update regions** or **lower color depth** when drawing. In ESPHome (Home Assistant), users had to disable continuous display updates and use LVGL for rendering because of memory limitscommunity.home-assistant.iocommunity.home-assistant.io. In Arduino, simply ensure you aren't buffering full 320x480 frames in RAM unless necessary – draw in chunks or use the TFT_eSPI push-color methods which stream pixels without huge buffers.

- **Touch & I2C Conflicts:** The GT911 uses its own I2C (on 32/33). If you use Wire for other devices, you have two options: (1) Use a separate TwoWire instance on different pins for other sensors (e.g. use Wire for the touch and Wire1 for other devices on another set of pins), or (2) Put all I2C devices on the same bus if possible. The GT911 can share the I2C bus with others, *provided* they have different addresses. In the design of this board, there are actually multiple 4-pin JST connectors that likely tie into the same I2C bus lines (one community member noted connectors labeled for "Temperature/Humidity" etc., which are probably wired to GPIO32/33 as well)forum.arduino.cc. So you can potentially put an RTC or sensor on the same SDA=33/SCL=32 lines. Just be cautious: the GT911's INT pin (GPIO21) is also used; that line is separate and won't conflict with other I2C devices, but ensure any additional device on 32/33 doesn't try to use GPIO21. In summary, plan your I2C usage such that the touchscreen's bus is either exclusive or properly shared. If you run into issues (e.g. another device's library wants to use the default Wire on 21/22), you may need to instantiate a custom TwoWire for the GT911 or vice-versa.
- **Serial Console and Logging:** By default, the serial programming port is on UART0 (GPIO1 TX, GPIO3 RX). The USB-UART is connected to these, so Serial.print goes to USB. This is fine for debugging. Just remember GPIO3 (RX0) is also the board's UART receive; avoid using GPIO3 for anything else or you'll lose programming ability. Similarly, do not tie GPIO1 (TX0) to anything that might interfere with the USB serial.

Additional Resources & Documentation

- **DIYmalls Documentation:** DIYmalls provides an *"Installation Manual"* PDF which includes Arduino setup instructions and an example using LVGLm.media-amazon.com. (It basically guides you to replace TFT_eSPI's User_Setup.h with their config and run an LVGL demo.) While somewhat brief, it can be found on the Amazon product page under "Product guides and documents." It includes example code (LVGL_Arduino-3.5CTP-gt911.ino) and wiring screenshots. The core pin definitions from that are the ones we've used above.
- **Schematic & Board Info:** The board's schematic (labeled *ESP32-3248S035 V1.1, 2022-02-26*) has been shared in forumsgithub.com. It confirms all pin connections we listed. Notably it shows the dual touch controller design (XPT2046 vs GT911) and how GPIOs are routed. If interested, community member *Matthias Birkich* shared a zip file with the complete documentation and demo code for this modulegithub.com. (That zip from manufacturer "Jingcai" contains the schematic, usage docs, and demo projects with LovyanGFX/LVGL.)
- **Forum Discussions:** Because official docs were sparse, several forum threads can be insightful:
    - Arduino Forum – *"I need help with understanding the ESP32-3248S035!"* – discusses features and confirms the touch controllers usedforum.arduino.cc.
    - Home Assistant Community – *"Help making ESP32-3248S035 work (ESPHome)"* – users share pin configs for ESPhome, and one post neatly lists all the GPIO assignments (which we cited here)community.home-assistant.iocommunity.home-assistant.io.
    - GitHub Issues – Bodmer's TFT_eSPI and LVGL issues have threads on this board. For example, GitHub issue **#3323 in TFT_eSPI** shows a user attempting to use TFT_eSPI and eventually confirming it works with the right configgithub.com. There is also an LVGL

issue about this board where the manufacturer's example code was discussedgithub.comgithub.com.

- Reddit – The r/esp32 subreddit has a few posts about "ESP32-3248S035C" which can provide tips (e.g. one user asking how to use the touch, etc.). Often, the advice is to use the correct library and pins as we've covered.

- **Performance Notes:** Driving a 480x320 SPI display can be demanding. With our pin config, you can push the SPI clock to 40 MHz (TFT_eSPI default for ILI9488/ST7796). That yields decent performance – for instance, DIYmalls' demo shows ~66 FPS with an optimized LVGL UIm.media-amazon.comm.media-amazon.com, but that is with partial updates. Don't expect to redraw the entire screen at high FPS without flicker – SPI at 40MHz can theoretically do ~10 million pixels/second, which is ~20 full frames/sec for 153,600 pixels, and in practice overhead will lower that. For smoother performance, use **LovyanGFX** library as an alternative to TFT_eSPI – it's highly optimized for ESP32 and this board (the official doc even references LovyanGFX). Also, enabling **8-bit parallel mode** is not applicable here since the hardware is wired for SPI, but if you needed more speed and were designing a custom board, the ST7796 could be used in parallel mode toogithub.com.

- **Known Issues:** Aside from the WiFi power and memory considerations already mentioned, a few quirks have been noted by users:

  - The **boot messages** from the ESP32 (printed at 115200 baud on reset) will appear on the serial monitor or possibly even cause a brief flash on the display's backlight pin (since GPIO2 is used by the LCD and also as a strapping pin that emits a log if booting in debug mode). These are generally harmless.

  - Some boards shipped with **old firmware** pre-loaded (e.g. an Annex32 BASIC environment) which outputs to the screen or UART. When you flash your Arduino sketch, it overwrites this. Just be aware you might see some initial text or have to reset the board after first power-up. One Amazon review mentioned the module comes with no documentation and finding the config was the hardest partamazon.com – which this guide aims to solve!

  - **Silkscreen labeling:** Unfortunately, the board's silk doesn't label all signals (especially those used internally for the display/touch). This means you can't identify the pin functions just by looking at the board. Always refer to a pin map (such as the one above) when programming it – e.g., knowing that "TFT_MOSI=GPIO13" is not obvious from the board itself. The community strongly wished the vendor provided such info in a one-page sheetamazon.com.

  - If using **Bluetooth**, note that the default partition (1.2MB app) might be small if you also include BLE + WiFi + a large display GUI. You may need to adjust the partition scheme to "No OTA (2MB APP)" to get more program space, if your code is large. This is set in Arduino Tools menu when the board is selected.

In summary, the ESP32-3248S035C is a versatile module once set up correctly. With the pin details and library configurations provided here, you should be able to get the display and touch working in Arduino.

From there, you can explore building your UI or IoT application – whether it's a home automation panel, a sensor interface, or a portable device. Despite its initial documentation gaps, the module has a "fabulous" set of features according to early adoptersforum.arduino.cc, combining an ESP32's connectivity with a touchscreen, SD storage, audio, and more. We've gathered the essential info to save you time and get your project up and running with this device. Happy hacking!

Sources: Key references include the openHASP project documentation for Sunton screensopenhasp.comopenhasp.com, community pin mappings from Home Assistant forumscommunity.home-assistant.iocommunity.home-assistant.io, the Arduino Forum and Reddit posts, as well as the DIYmalls/Amazon product infoamazon.comamazon.com. These have been cited in-line above for accuracy. For further details, consult the linked sources or the manufacturer's example code if needed.

Citations

ESP32-3248S035 - openHASP

https://www.openhasp.com/0.7.0/hardware/sunton/esp32-3248s035/

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332

Amazon.com: DIYmall 3.5" ESP32 Display Module ESP32-3248S035C WiFi+BT Dual-core MCU 320x480 3.5inch TFT Capacitive Touch Screen ST7796 4MB Flash for IoT Smart Home : Electronics

https://www.amazon.com/DIYmall-ESP32-3248S035C-Dual-core-320x480-Capacitive/dp/B0C5DB6RHM

Amazon.com: DIYmall 3.5" ESP32 Display Module ESP32-3248S035C WiFi+BT Dual-core MCU 320x480 3.5inch TFT Capacitive Touch Screen ST7796 4MB Flash for IoT Smart Home : Electronics

https://www.amazon.com/DIYmall-ESP32-3248S035C-Dual-core-320x480-Capacitive/dp/B0C5DB6RHM

ESP32-3248S035 - openHASP

https://www.openhasp.com/0.7.0/hardware/sunton/esp32-3248s035/

I need help with understanding the ESP32-3248S035! - General Guidance - Arduino Forum

https://forum.arduino.cc/t/i-need-help-with-understanding-the-esp32-3248s035/1281502

Amazon.com: DIYmall 3.5" ESP32 Display Module ESP32-3248S035C WiFi+BT Dual-core MCU 320x480 3.5inch TFT Capacitive Touch Screen ST7796 4MB Flash for IoT Smart Home : Electronics

https://www.amazon.com/DIYmall-ESP32-3248S035C-Dual-core-320x480-Capacitive/dp/B0C5DB6RHM

ESP32-3248S035 - openHASP

https://www.openhasp.com/0.7.0/hardware/sunton/esp32-3248s035/

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332

Amazon.com: DIYmall 3.5" ESP32 Display Module ESP32-3248S035C WiFi+BT Dual-core MCU 320x480 3.5inch TFT Capacitive Touch Screen ST7796 4MB Flash for IoT Smart Home : Electronics

https://www.amazon.com/DIYmall-ESP32-3248S035C-Dual-core-320x480-Capacitive/dp/B0C5DB6RHM

ESP32-3248S035C with GT911 touch chip not working · Issue #3323 · Bodmer/TFT_eSPI · GitHub

https://github.com/Bodmer/TFT_eSPI/issues/3323

ESP32-3248S035 - openHASP

https://www.openhasp.com/0.7.0/hardware/sunton/esp32-3248s035/

ESP32-3248S035C with GT911 touch chip not working · Issue #3323 · Bodmer/TFT_eSPI · GitHub

https://github.com/Bodmer/TFT_eSPI/issues/3323

ESP32-3248S035 - openHASP

https://www.openhasp.com/0.7.0/hardware/sunton/esp32-3248s035/

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332

I need help with understanding the ESP32-3248S035! - General Guidance - Arduino Forum

https://forum.arduino.cc/t/i-need-help-with-understanding-the-esp32-3248s035/1281502

ESP32-3248S035: use with an external RTC - General Guidance - Arduino Forum

https://forum.arduino.cc/t/esp32-3248s035-use-with-an-external-rtc/1314183

https://m.media-amazon.com/images/I/A1DphRgKG9L.pdf

https://m.media-amazon.com/images/I/A1DphRgKG9L.pdf

Something is very wrong with the WiFi on the ESP32 boards I ordered from AliExpress. Is there any way to salvage these and fix the WiFi? - Hardware - Home Assistant Community

https://community.home-assistant.io/t/something-is-very-wrong-with-the-wifi-on-the-esp32-boards-i-ordered-from-aliexpress-is-there-any-way-to-salvage-these-and-fix-the-wifi/477631

Something is very wrong with the WiFi on the ESP32 boards I ordered from AliExpress. Is there any way to salvage these and fix the WiFi? - Hardware - Home Assistant Community

https://community.home-assistant.io/t/something-is-very-wrong-with-the-wifi-on-the-esp32-boards-i-ordered-from-aliexpress-is-there-any-way-to-salvage-these-and-fix-the-wifi/477631

ESP32-3248S035: use with an external RTC - General Guidance - Arduino Forum

https://forum.arduino.cc/t/esp32-3248s035-use-with-an-external-rtc/1314183

ESP32-3248S035: use with an external RTC - General Guidance - Arduino Forum

https://forum.arduino.cc/t/esp32-3248s035-use-with-an-external-rtc/1314183

ESP32-3248S035: use with an external RTC - General Guidance - Arduino Forum

https://forum.arduino.cc/t/esp32-3248s035-use-with-an-external-rtc/1314183

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332

https://m.media-amazon.com/images/I/A1DphRgKG9L.pdf

ESP32-3248S035C with GT911 touch chip not working · Issue #3323 · Bodmer/TFT_eSPI · GitHub

https://github.com/Bodmer/TFT_eSPI/issues/3323

ESP32-3248S035C with GT911 touch chip, arduino demo compile error · Issue #6214 · lvgl/lvgl · GitHub

https://github.com/lvgl/lvgl/issues/6214

ESP32-3248S035C with GT911 touch chip, arduino demo compile error · Issue #6214 · lvgl/lvgl · GitHub

https://github.com/lvgl/lvgl/issues/6214

https://m.media-amazon.com/images/I/A1DphRgKG9L.pdf

ESP32-3248S035C Support · Bodmer TFT_eSPI · Discussion #3122 · GitHub

https://github.com/Bodmer/TFT_eSPI/discussions/3122

Amazon.com: DIYmall 3.5" ESP32 Display Module ESP32-3248S035C WiFi+BT Dual-core MCU 320x480 3.5inch TFT Capacitive Touch Screen ST7796 4MB Flash for IoT Smart Home : Electronics

https://www.amazon.com/DIYmall-ESP32-3248S035C-Dual-core-320x480-Capacitive/dp/B0C5DB6RHM

Amazon.com: DIYmall 3.5" ESP32 Display Module ESP32-3248S035C WiFi+BT Dual-core MCU 320x480 3.5inch TFT Capacitive Touch Screen ST7796 4MB Flash for IoT Smart Home : Electronics

https://www.amazon.com/DIYmall-ESP32-3248S035C-Dual-core-320x480-Capacitive/dp/B0C5DB6RHM

Help making ESP32-3248S035 work - ESPHome - Home Assistant Community

https://community.home-assistant.io/t/help-making-esp32-3248s035-work/748332