Here's the **updated procedure** specifically for the **RAKwireless RAK3112** WisDuo module (ESP32-S3 + SX1262 LoRa), where the exact board selection in the Arduino IDE is **RAKwireless RAK3112** (as required and officially supported in recent versions of the Espressif arduino-esp32 core).

This board entry became available in the official Espressif BSP (arduino-esp32 core version ~3.x or later), and it pre-configures the optimal settings for the RAK3112's hardware: 16MB Flash, 8MB PSRAM, native USB CDC, etc. No separate RAKwireless BSP is needed for basic Arduino sketches (unlike WisBlock nRF-based cores).

Step 1: Install or Update Arduino IDE

Use **Arduino IDE 2.x** (latest version as of 2026) for best compatibility.

Download: https://www.arduino.cc/en/software

Step 2: Add / Update ESP32 Board Support

1. Open Arduino IDE → **File → Preferences** (or **Arduino IDE → Settings** on macOS)
2. In **Additional Boards Manager URLs**, ensure this is present (add if missing):

   ```
   text

   https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
   ```

3. Go to **Tools → Board → Boards Manager**
4. Search for **esp32**
5. Install or update **esp32 by Espressif Systems** → select the **latest version** (3.x.x or newer — this includes the RAKwireless RAK3112 board definition)

Step 3: Select the Exact Board

Go to **Tools → Board → esp32** and choose:

RAKwireless RAK3112

- This is the precise entry you need (it appears after installing/updating the Espressif core)

core).

- It automatically sets Flash size to 16MB, PSRAM enabled, USB CDC, etc., optimized for the RAK3112.

Step 4: Recommended Upload / Configuration Settings (Tools menu)

The board selection usually applies good defaults, but verify/adjust these:

- **USB CDC On Boot** → **Enabled** (required for native USB on RAK3112)
- **CPU Frequency** → **240MHz (WiFi/BT)**
- **Flash Mode** → **QIO 80MHz** (preferred; fallback to DIO if issues)
- **Flash Size** → **16MB (128Mb)**
- **Partition Scheme** → **16M Flash (3MB APP/9.9MB FATFS)** or a large one if your app needs more space
- **PSRAM** → **Enabled** (OPI PSRAM — critical, 8MB on RAK3112)
- **Upload Mode** → **UART0 / Hardware CDC** (or default)
- **USB Mode** → **Hardware CDC and JTAG**
- **Erase All Flash Before Sketch Upload** → **Disabled** (enable only for recovery if stuck)

Step 5: Connect & Select Port

1. Connect your RAK3112 (via breakout board, WisBlock base, or direct USB) with a reliable USB data cable.
2. **Tools → Port** → pick the new port:
    - Windows: COMx
    - macOS: /dev/cu.usbmodem... or /dev/ttyACM...
    - Linux: /dev/ttyUSB... or /dev/ttyACM...
3. No port? Install drivers for CP210x, CH9102, or CH343 (check your breakout/base's USB-to-serial chip).

Step 6: Upload the Sketch

1. Open or write your sketch (use examples from RAK's GitHub for LoRa/BLE/WiFi).
2. Click **Upload** (→ icon).

RAK3112-specific upload trick (common on ESP32-S3):

- If "Connecting...." hangs:
    1. Hold the **BOOT** button (tied to GPIO0).
    2. Press and release **RESET** (or briefly disconnect/reconnect USB power).
    3. Release **BOOT** when upload progress starts (or hold until "Writing flash..." shows).
- Success looks like:

```
text

...
Leaving...
Hard resetting via RTS pin...
```

Quick Test Sketch (Serial Output + GPIO Blink)

Use a safe GPIO (e.g., GPIO2 if available on your breakout; check pinout).

```
C++
```

```
#define LED_PIN 2  // Adjust to a free GPIO on your setup

void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
  Serial.println("RAKwireless RAK3112 Test - ESP32-S3 + SX1262");
  Serial.println("Board selected: RAKwireless RAK3112");
}

void loop() {
  digitalWrite(LED_PIN, HIGH);
  Serial.println("LED ON");
  delay(1000);
  digitalWrite(LED_PIN, LOW);
  Serial.println("LED OFF");
  delay(1000);
}
```

- After upload, open **Serial Monitor** at **115200 baud**.

Extra Notes & Troubleshooting (2026-era)

- **Board not listed?** Update esp32 core to latest — the "RAKwireless RAK3112" entry is in recent Espressif releases.
- **LoRa examples** → Use RAK's repo: https://github.com/RAKWireless/WisBlock/tree/master/examples/RAK3112 (includes SX1262 pin defs via libraries like RadioLib or SX126x-Arduino).
- **Persistent upload issues?** Enable **Erase All Flash** once → retry → disable.
- **PSRAM test** → Try allocating large arrays; if it crashes, double-check PSRAM is Enabled.
- Verbose logs: **File → Preferences → Show verbose output during: Upload** — share errors if needed.

This aligns with RAKwireless docs (e.g., Quick Start Guide) and uses the official **RAKwireless RAK3112** board entry. If your core version doesn't show it yet or you get

RAKwireless RAK3112 board entry. If your core version doesn't show it yet or you get errors, let me know your esp32 core version!