

---

**T-TWR**

***Release master***

**lewin**

**Jun 21, 2023**



# CONTENTS

<b>1</b>	<b>Get Started</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	What You Need . . . . .	1
1.3	Install prerequisites . . . . .	1
1.4	Install dependent libraries . . . . .	2
1.5	Examples . . . . .	2
1.6	Pinout . . . . .	3
1.7	Datasheet . . . . .	3
1.8	Troubleshooting . . . . .	3
<b>2</b>	<b>Examples</b>	<b>5</b>
2.1	AT Debug . . . . .	5
2.2	Walkie Talkie . . . . .	7
<b>3</b>	<b>Troubleshooting</b>	<b>11</b>
3.1	1. How to enter download mode? . . . . .	11
3.2	2. SA868 power problem . . . . .	11
<b>4</b>	<b>Copyrights and Licenses</b>	<b>13</b>
4.1	Software Copyrights . . . . .	13
4.2	Third Party . . . . .	13
4.3	Documentation . . . . .	13



## GET STARTED

This document is intended to guide users to build a software environment for T-TWR hardware development.

### 1.1 Introduction

T-TWR has the following hardware resources:

- ESP32-S3-WROOM-1-N16R8
- OLED display (128\*64)
- SA868 Embedded small size wailie talkie module
- TP4054 battery charging chip
- 18650 battery holder
- Encoder

### 1.2 What You Need

Hardware:

- T-TWR
- Computer running Windows, Linux, or macOS

Software:

- arduino
- platformio

### 1.3 Install prerequisites

Please complete the installation of the tool first. The specific steps are as follows:



## 1.4 Install dependent libraries

T-TWR supports the following versions of arduino-esp32:

- `arduino-esp32 2.0.5`

T-TWR needs the support of the following dependent libraries, users can use the library management to install:

- `RotaryEncoder 1.5.3`
- `OneButton 2.0.3`
- `U8g2 2.34.5`
- `ArduinoJson 6.19.4`

<b>Warning:</b> If there is a compilation error, please check whether the above dependent library versions are consistent.
--

## 1.5 Examples

- *AT Debug*
- *Walkie Talkie*

## 1.6 Pinout

Name	Pin	Note
SA868_TX_PIN	47	Serial transmit data pin
SA868_RX_PIN	48	Serial receive data pin
SA868_PTT_PIN	41	SA868 Transmitting/receiving control, “0” force the module to enter TX state; and “1” to Rx state.
SA868_PD_PIN	40	Power Down control, “0” for power down mode; “1” for the normal work
SA868_RF_PIN	39	high/low output power control; Leave open for high output power, low level to low output power.
BUT- TON_PTT_PIN	38	transmit/receive button
BUTTON_UP_PIN	3	volume+
BUT- TON_DOWN_PIN	0	volume1
ENCODER_A_PIN	9	Encoder Pin1
ENCODER_B_PIN	5	Encoder Pin2
EN- CODER_OK_PIN	7	Encoder OK button
BAT- TERY_ADC_PIN	6	Battery voltage detection pin
OLED_POWER_PIN	21	OLED screen power control pin
LED_PIN	1	User-Defined Indicators

## 1.7 Datasheet

- [ESP32-S3-WROOM-1-N16R8 \(Datasheet\)](#)
- [OLED Display Module \(Datasheet\)](#)
- [SA868 \(Datasheet\)](#)
- [Slide and Rotary Encoder Switches \(Datasheet\)](#)

## 1.8 Troubleshooting

If something goes wrong, first see [Troubleshooting](#). If [Troubleshooting](#) doesn't cover the issue you're having, please get technical support via [GitHub Issue](#).





## EXAMPLES

### 2.1 AT Debug

Communication protocol for debugging SA868.

The esp32s3 is used as a serial port transponder, and the PC sends the SA868 command to the esp32s3 through the serial port tool, and the esp32s3 forwards it to the SA868.

In the same way, the response command of SA868 is sent to esp32s3, and then forwarded to PC.

#### 2.1.1 Arduino

The following configuration is recommended:

Auto Format	Ctrl+T
Archive Sketch	
Fix Encoding & Reload	
Manage Libraries...	Ctrl+Shift+I
Serial Monitor	Ctrl+Shift+M
Serial Plotter	Ctrl+Shift+L
ESP32 Sketch Data Upload	
WiFi101 / Wi-FiNINA Firmware Updater	
Board: "ESP32S3 Dev Module"	>
Upload Speed: "921600"	>
USB Mode: "Hardware CDC and JTAG"	>
USB CDC On Boot: "Enabled"	>
USB Firmware MSC On Boot: "Disabled"	>
USB DFU On Boot: "Disabled"	>
Upload Mode: "UART0 / Hardware CDC"	>
CPU Frequency: "240MHz (WiFi)"	>
Flash Mode: "QIO 80MHz"	>
Flash Size: "16MB (128Mb)"	>
Partition Scheme: "16M Flash (2MB APP/12.5MB FAT)"	>
Core Debug Level: "None"	>
PSRAM: "OPi PSRAM"	>
Arduino Runs On: "Core 1"	>
Events Run On: "Core 1"	>
Port	>
Get Board Info	
Programmer	>
Burn Bootloader	

## 2.1.2 PlatformIO

See PlatformIO IDE for VSCode.

### 2.1.3 Burning

The T-TWR is linked to the computer via USB. First press and hold **Volume-**, then lightly press the reset button, and finally release **Volume-** to enter the download mode.

### 2.1.4 Pinout

Name	Pin	Note
SA868_TX_PIN	47	Serial transmit data pin
SA868_RX_PIN	48	Serial receive data pin
SA868_PTT_PIN	41	SA868 Transmitting/receiving control, “0” force the module to enter TX state; and “1” to Rx state.
SA868_PD_PIN	40	Power Down control, “0” for power down mode; “1” for the normal work
SA868_RF_PIN	39	high/low output power control; Leave open for high output power, low level to low output power.
BUT- TON_PTT_PIN	38	transmit/receive button

### 2.1.5 Datasheet

- [ESP32-S3-WROOM-1-N16R8](#) (Datasheet)
- [SA868](#) (Datasheet)

### 2.1.6 Troubleshooting

If something goes wrong, first see [Troubleshooting](#). If [Troubleshooting](#) doesn’t cover the issue you’re having, please get technical support via [GitHub Issue](#).

## 2.2 Walkie Talkie

### 2.2.1 Arduino

The following configuration is recommended:

Auto Format	Ctrl+T
Archive Sketch	
Fix Encoding & Reload	
Manage Libraries...	Ctrl+Shift+I
Serial Monitor	Ctrl+Shift+M
Serial Plotter	Ctrl+Shift+L
ESP32 Sketch Data Upload	
WiFi101 / Wi-FiNINA Firmware Updater	
Board: "ESP32S3 Dev Module"	>
Upload Speed: "921600"	>
USB Mode: "Hardware CDC and JTAG"	>
USB CDC On Boot: "Enabled"	>
USB Firmware MSC On Boot: "Disabled"	>
USB DFU On Boot: "Disabled"	>
Upload Mode: "UART0 / Hardware CDC"	>
CPU Frequency: "240MHz (WiFi)"	>
Flash Mode: "QIO 80MHz"	>
Flash Size: "16MB (128Mb)"	>
Partition Scheme: "16M Flash (2MB APP/12.5MB FAT)"	>
Core Debug Level: "None"	>
PSRAM: "QIO PSRAM"	>
Arduino Runs On: "Core 1"	>
Events Run On: "Core 1"	>
Port	>
Get Board Info	
Programmer	>
Burn Bootloader	

## 2.2.2 PlatformIO

See PlatformIO IDE for VSCode.

### 2.2.3 Install dependent libraries

T-TWR supports the following versions of arduino-esp32:

- [arduino-esp32 2.0.5](#)

T-TWR needs the support of the following dependent libraries, users can use the library management to install:

- [RotaryEncoder 1.5.3](#)
- [OneButton 2.0.3](#)
- [U8g2 2.34.5](#)
- [ArduinoJson 6.19.4](#)

**Warning:** If there is a compilation error, please check whether the above dependent library versions are consistent.

### 2.2.4 Band selection

---

**Note:** Because the AT command cannot read the frequency band information of the SA868 module, it is necessary to manually select the frequency band in the source code.

---

The `UHF_BAND` macro is used to select the UHF band, and the supported frequency range is 400 ~ 480 MHz

The `_350_BAND` macro is used to select the VHF band, the supported band range is 320 ~ 400 MHz

The `VHF_BAND` macro is used to select the VHF band, the supported band range is 134 ~ 174 MHz

### 2.2.5 Datasheet

- [ESP32-S3-WROOM-1-N16R8 \(Datasheet\)](#)
- [OLED Display Module \(Datasheet\)](#)
- [SA868 \(Datasheet\)](#)
- [Slide and Rotary Encoder Switches \(Datasheet\)](#)

### 2.2.6 Burning

The T-TWR is linked to the computer via USB. First press and hold **Volume-**, then lightly press the reset button, and finally release **Volume-** to enter the download mode.

## 2.2.7 Detailed Operation

pending upgrade.

## 2.2.8 Pinout

Name	Pin	Note
SA868_TX_PIN	47	Serial transmit data pin
SA868_RX_PIN	48	Serial receive data pin
SA868_PTT_PIN	41	SA868 Transmitting/receiving control, “0” force the module to enter TX state; and “1” to Rx state.
SA868_PD_PIN	40	Power Down control, “0” for power down mode; “1” for the normal work
SA868_RF_PIN	39	high/low output power control; Leave open for high output power, low level to low output power.
BUT- TON_PTT_PIN	38	transmit/receive button
BUTTON_UP_PIN	3	volume+
BUT- TON_DOWN_PIN	0	volume-
ENCODER_A_PIN	9	Encoder Pin1
ENCODER_B_PIN	5	Encoder Pin2
EN- CODER_OK_PIN	7	Encoder OK button
BAT- TERY_ADC_PIN	6	Battery voltage detection pin
OLED_POWER_PIN	21	OLED screen power control pin
LED_PIN	1	User-Defined Indicators

## 2.2.9 Troubleshooting

If something goes wrong, first see *Troubleshooting*. If *Troubleshooting* doesn’t cover the issue you’re having, please get technical support via [GitHub Issue](#).

## TROUBLESHOOTING

### 3.1 1. How to enter download mode?

First press and hold the **Volume-** button, then lightly press the reset button, and finally release the **Volume-** button to enter the download mode.

### 3.2 2. SA868 power problem

The SA868 module supports low (1.6W) / high (1.8W) power; the power can be selected by controlling the level of Pin#7 of SA868. The program uses low power by default, if you choose high power, it will cause the temperature of T-TWR to be too high, please use it with caution.

<p><b>Warning:</b> When there is no program in esp32s3, Pin#39 of esp32s3 is floating, and the power of SA868 will be set to high power.</p>
--





## COPYRIGHTS AND LICENSES

### 4.1 Software Copyrights

All original source code in this repository is Copyright (C) 2022 LILYGO.

### 4.2 Third Party

- [RotaryEncoder](#) Copyright (c) 2005-2014 by Matthias Hertel and licensed under the BSD license.
- [OneButton](#) Copyright (c) 2005-2014 by Matthias Hertel and licensed under the BSD license.
- [U8g2\\_Arduino](#) Copyright (c) 2016, [olikraus@gmail.com](mailto:olikraus@gmail.com) and licensed under new-bsd license.
- [ArduinoJson](#) Copyright © 2014-2022, Benoit BLANCHON and licensed under MIT License.

### 4.3 Documentation

- HTML version of the T-Wristband Programming Guide uses the Sphinx theme `sphinx_idf_theme`, which is Copyright (c) 2013-2020 Dave Snider, Read the Docs. It is licensed under the MIT license.