

[>> Back to LoRa Main topics](#)

lamaPLC: Waveshare LoRA products

Waveshare offers a diverse range of LoRa and LoRaWAN products, primarily categorized into Gateways for network infrastructure, HATs/Expansion Boards for Raspberry Pi/Pico, Arduino, and standalone Core Modules for embedded integration.

Product Category	Example Models	Primary Chip	Interface	Key Features
LoRaWAN Gateway	SX1302/SX1303 868M Gateway HAT	SX1302 / SX1303	SPI (via Mini-PCIe)	8-10 channels, multi-node concurrency, high sensitivity (-141dBm)
Plug-and-play USB interface	USB-TO-LoRa-xF	SX1262	USB/UART (with AT commands)	LF for 410-510MHz or HF for 850-930MHz
Node Expansion HAT	SX1262 LoRaWAN/GNSS HAT	SX1262	SPI	Standard 40-pin GPIO, integrated GPS (GNSS) on some models, LoRaWAN support
Point-to-Point HAT	SX1262/SX1268 LoRa HAT	SX1262 / SX1268	UART	Proprietary “private” protocol, does not support LoRaWAN, easy AT command config
Development Boards	RP2040-LoRa	SX1262 + RP2040	Internal SPI	All-in-one MCU + LoRa board, USB-C programming, compact form factor
Core Modules	Core 1262 HF/LF	SX1262	SPI	Minimalist stamp-hole design for direct soldering onto custom PCBs
Industrial DTU	SX1262-LoRa-DTU-xF	SX1262	RS232/485/422	Rail-mount, aluminum alloy case, long-range wireless serial bridge

Important Selection Criteria

- **LoRa vs. LoRaWAN:** Many Waveshare HATs are split by protocol. The UART Interface models often use a proprietary protocol and cannot connect to standard LoRaWAN gateways such as The Things Network.
- **Frequency Bands:** Waveshare typically offers each product in two versions: LF (Low Frequency) for 410-510MHz (e.g., CN470) and HF (High Frequency) for 850-930MHz (e.g., EU868, US915).
- **Interface Choice:** Use SPI products if you want full control via libraries (like RadioHead or LMIC); use UART or USB versions for simple plug-and-play serial data transmission.

Waveshare USB-TO-LoRa-xF02

The USB-TO-LoRa-xF is an industrial-grade wireless data transfer unit from Waveshare that provides a plug-and-play USB interface for LoRa communication. It is designed to allow computers or other USB-enabled hosts to communicate over long distances using the Semtech SX1262 chip.



The “xF” in the name refers to the frequency version (e.g., LF for 410–510 MHz or HF for 850–930 MHz), and “02” typically denotes a specific hardware iteration or feature set.

USB-TO-LoRa-xF02 Lora AT commands

AT command	Status query	Type query	Description	Note
+++ AT+EXIT			Enter AT command mode Exit AT command mode	
ATE			Enable/disable AT command echo	
AT+VER			Check the software version number	
AT+HELP			View AT help	
AT+REBOOT			Reboot	
After all parameter changing use AT+EXIT to save the parameters!				
AT+SF=7	AT+SF?	AT+SF=?	Set the spreading factor, the value range is 7~12 SF7: Highest data rate, shortest range, and lowest power consumption per packet. SF12: Lowest data rate, longest range, and highest sensitivity for penetrating obstacles. Default value: 7	
AT+BW=0	AT+BW?	AT+BW=?	Set bandwidth, 0 means 125KHz, 1 means 250KHz, 2 means 500KHz	
AT+CR=1	AT+CR?	AT+CR=?	Set the encoding rate to 1, 1 represents 4/5, 2 represents 4/6, 3 represents 4/7, 4 represents 4/8	
AT+PWR=22	AT+PWR?	AT+PWR=?	Set the RF power, the value range is 10~22dBm Selectable Values: 10, 13, 17, or 22 Higher power levels (e.g., 22 dBm) provide the longest communication range but consume more power. At maximum power (22 dBm), the module's transient transmit current can reach approximately 100 mA to 107 mA. Default: 22	

AT command	Status query	Type query	Description	Note
AT+NETID=0\r\n	AT+NETID?	AT+NETID=?	<p>Network ID assignment, the value range is 0~65535</p> <p>Only modules with the same Network ID, Frequency, and Air Speed can communicate with each other.</p> <p>Default: 0</p>	
AT+LBT=0\r\n	AT+LBT?	AT+LBT=?	<p>Enable/disable LBT (<i>Listen Before Talk</i>) function, 0: disable, 1: enable</p> <p>module enables or disables the Listen Before Talk (LBT) function. This feature is designed to prevent data collisions by ensuring the radio channel is clear before the module begins its own transmission.</p> <p>In some regions (such as parts of Europe for the 868MHz band), LBT is a required alternative to strict "Duty Cycle" limitations to ensure fair access to the spectrum.</p> <p>Default: 0</p>	
AT+MODE=1\r\n	AT+MODE?	AT+MODE=?	<p>DTU working mode, 1: <i>stream mode</i>, 2: <i>packet mode</i>, 3: <i>relay mode</i></p> <p>Stream Mode (AT+MODE=1): The factory default mode. It functions as a "virtual cable," where data sent to the serial port is transmitted immediately in plaintext to all other modules on the same channel.</p> <p>Packet Mode (AT+MODE=2): Used for targeted point-to-point or point-to-multipoint communication. In this mode, you must prefix your data with the target device's Address and Channel (e.g., FF FE 12 + data).</p> <p>Relay Mode (AT+MODE=3): Configures the module to act as a repeater node to extend the transmission range. In this mode, the device only forwards packets and does not output data to its own serial interface.</p>	

AT command	Status query	Type query	Description	Note
AT+TXCH=18\r\n	AT+TXCH?	AT+TXCH=?	<p>Transmit channel, value range 0~80, corresponding frequency point is 850~930MHz or 410~490MHz</p> <p>LF Version (usually labeled “LF” or marked with a 433MHz sticker) Formula: Frequency (MHz) = 410 + Channel Range: Channel 0: 410 MHz, Channel 23: 433 MHz (Default), Channel 80: 490 MHz Default: 23 (corresponding to 433 MHz). - Example: AT+TXCH=23 sets the frequency to 433 MHz (410 + 23). - Example: AT+TXCH=80 sets the frequency to 490 MHz (410 + 80).</p> <p>HF Version (usually labeled “HF” or marked with an 868/915MHz sticker) Formula: Frequency (MHz) = 850 + Channel Range: 850 MHz (Channel 0) to 930 MHz (Channel 80) Default Value: 18 (corresponding to 868 MHz) - Example: 868 MHz: AT+TXCH=18\r\n (Standard for Europe) - Example: 915 MHz: AT+TXCH=65\r\n (Standard for North America/Australia) - Example: 923 MHz: AT+TXCH=73\r\n (Common in Asia)</p>	
AT+RXCH=18\r\n	AT+RXCH?	AT+RXCH=?	<p>Receive channel, value range 0~80, corresponding frequency point is 850~930MHz or 410~490MHz</p> <p>LF Version (usually labeled “LF” or marked with a 433MHz sticker) Formula: Frequency (MHz) = 410 + Channel Range: Channel 0: 410 MHz, Channel 23: 433 MHz (Default), Channel 80: 490 MHz Default: 23 (corresponding to 433 MHz). - Example: AT+TXCH=23 sets the frequency to 433 MHz (410 + 23). - Example: AT+TXCH=80 sets the frequency to 490 MHz (410 + 80).</p> <p>HF Version (usually labeled “HF” or marked with an 868/915MHz sticker) Formula: Frequency (MHz) = 850 + Channel Range: 850 MHz (Channel 0) to 930 MHz (Channel 80) Default Value: 18 (corresponding to 868 MHz) - Example: 868 MHz: AT+TXCH=18\r\n (Standard for Europe) - Example: 915 MHz: AT+TXCH=65\r\n (Standard for North America/Australia) - Example: 923 MHz: AT+TXCH=73\r\n (Common in Asia)</p>	

AT command	Status query	Type query	Description	Note
AT+RSSI=0\r\n	AT+RSSI?	AT+RSSI=?	<p>Enable/disable RSSI (<i>Received Signal Strength Indicator</i>) signal value output, 0: disable, 1: enable.</p> <p>When enabled, the module appends a hexadecimal RSSI value to the end of every received data packet. Signal Quality: Typical LoRa RSSI values range from -30 dBm (very strong) to -120 dBm (very weak).</p> <p>Default:0</p>	
AT+ADDR=0\r\n	AT+ADDR?	AT+ADDR=?	<p>Set DTU (<i>Data Transfer Unit</i>) address, value range 0~65535</p> <p>This address acts as the device's identity within a network. <i>Broadcast Address: 65535 (Hex 0xFFFF).</i></p> <p>Setting a module to this address allows it to receive data from any other module on the same channel, regardless of the sender's target address.</p> <p>Default:0</p>	
AT+PORT=0\r\n	AT+PORT?	AT+PORT=?	<p>The command is used to select the active hardware interface mode on the module.</p> <p>Values: 1: RS485 2: RS232 3: USB (Typically the default for this specific dongle)</p> <p>Typically, you do not need to change it unless you are using the same firmware on a different DTU hardware variant.</p>	
AT+KEY=0\r\n	AT+KEY?	AT+KEY=?	<p>Used to configure AES encryption for wireless data transmission, value range:</p> <p>0: Disables encryption (<i>factory default</i>) 1 to 65535: Sets the specific key value for AES encryption</p>	
AT+BAUD=115200\r\n	AT+BAUD?	AT+BAUD=?	<p>Set COMx port baud rate, value range 1200~115200, 1200, 2400,, 57600,</p> <p>Common Values: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.</p> <p>Default: 115200</p>	
AT+COMM="8N1"\r\n	AT+COMM?	AT+COMM=?	<p>Set COM port parameters, data bits: 8 or 9, parity: N, O, E, stop bits: 0, 1, 2,</p> <p>Default: 8N1</p>	

AT command	Status query	Type query	Description	Note
AT+AIIP=xx\r\n	AT+ALLP?		Set the spreading factor to key multi-parameter [spreading factor 7..12],[bandwidth 125/250/500],[encoding rate 1..4],[RF power 10/13/17/22],[Network ID 0..65535],[LBT function 0/1],[DTU Mode 1/2/3],[Transmit channel 0..80],[Receive channel 0..80],[RSSI signal 0/1],[DTU address 0..65535],[Application Port 1..223],[Baud rate 1200..115200],[COM port "8N1"],[AES Key 0..65535] <i>Example:</i> AT+AIIP=7,125,1,22,0,0,1,18,18,0,0,3,115200,"8N1",0	
AT+RESTORE=0\r\n	AT+RESTORE?		resets all configuration parameters to their factory default settings. 0: disabled, 1: enabled	

2026/04/23 21:51

Waveshare Core 1262 HF/LF

The Waveshare Core1262 is a small LoRa module built around the latest Semtech SX1262 chip. It provides greater power efficiency and extended transmission range compared to the previous SX127x series.



The module is available in two versions based on the operating frequency:

- **Core1262-HF (High Frequency):** Operates between 850 ~ 930MHz. It is primarily used in Europe, North America, and Oceania.
- **Core1262-LF (Low Frequency):** Operates between 410 ~ 510MHz. It is typically used in Asia and Europe.

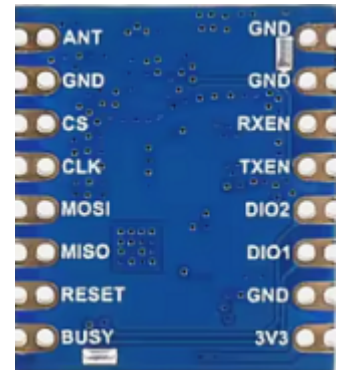
Technical Specifications

Both versions share common core specifications:

- **RF Chip:** Semtech SX1262.
- **Modulation:** LoRa and (G)FSK.
- **Transmit Power:** Up to +22dBm at 3.3V.
- **Data Rate:** LoRa (0.018~62.5Kbps), (G)FSK (0.6~300Kbps).
- **Communication Interface:** SPI (18MHz maximum speed).
- **Operating Voltage:** 3.3V.
- **Key Components:** Features an industrial-grade TCXO (*Temperature Compensated Crystal Oscillator*) for stable operation in harsh conditions (0~85°C).

- **Documentation:** Detailed guides and example codes are available on the [Waveshare Wiki](#).

Waveshare Core 1262 HF/LF Pinout

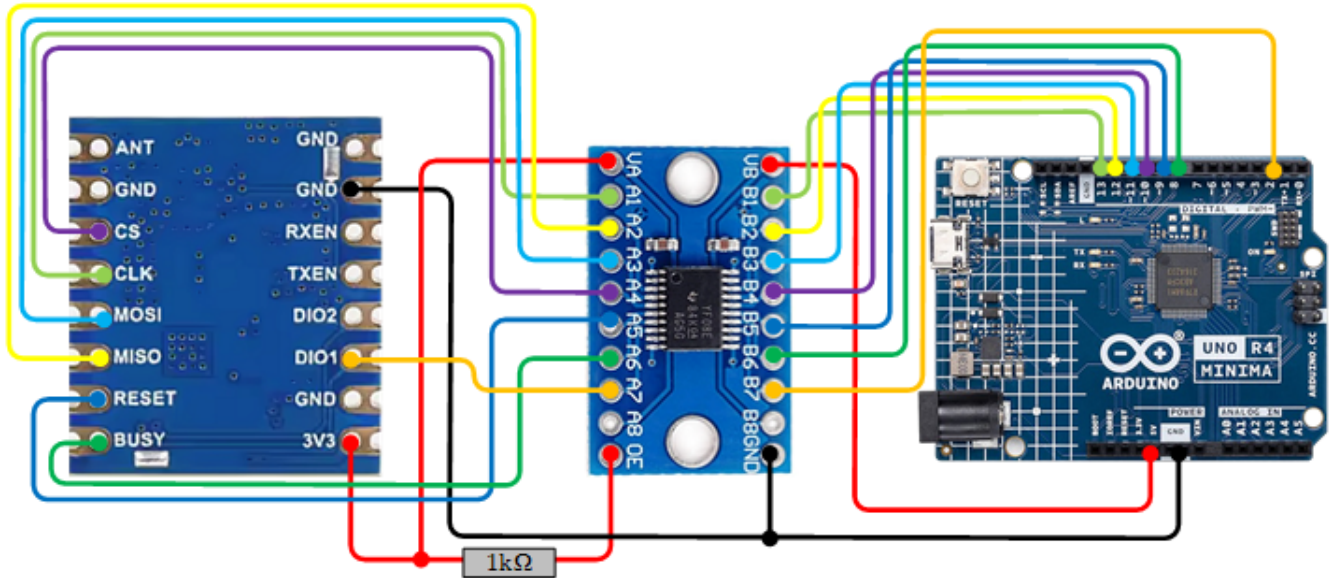


Pin Name	Type	Description
VCC	Power	3.3V power supply
GND	Ground	Ground connection
CS	Input	SPI Chip Select (Active Low)
SCK / CLK	Input	SPI Serial Clock (up to 18MHz)
MOSI	Input	SPI Master Out Slave In (Data to module)
MISO	Output	SPI Master In Slave Out (Data from module)
RESET	Input	External Reset; pull low for 100μs to restore defaults
BUSY	Output	Status pin; High means busy, Low means idle (ready for commands)
DIO1	Output	Configurable IRQ/Interrupt output (often used for RX/TX completion)
RXEN	Input	RF Switch Control; High for Receive mode
TXEN	Input	RF Switch Control; High for Transmit mode
ANT	RF	Antenna connection pin (can be used if not using the IPEX connector)

Waveshare Core 1262 HF/LF Wiring

Because the Core1262 operates at 3.3V, you must use a 3.3V Arduino (*like the MKR series*) or use level shifters if using a 5V Uno to avoid damaging the module.

Ra-01 Pin	Logic level converter	Arduino Pin (Uno/Nano)	Wire Color	Notes
VCC	Va - Vb	5V	♦ red	Power
GND	GND	GND	♦ black	Common ground
CLK	A1 - B1	D13	♦ lightgreen	SPI Clock
MISO	A2 - B2	D12	♦ yellow	SPI Master In Slave Out
MOSI	A3 - B3	D11	♦ lightblue	SPI Master Out Slave In
CS/NSS	A4 - B4	D10	♦ purple	Chip Select (CS)
RESET	A5 - B5	D9	♦ darkblue	Reset Pin
BUSY	A6 - B6	D8	♦ green	Busy Pin
DIO1	A7 - B7	D2	♦ orange	Interrupt Pin (Required for RX)



Waveshare Core 1262 HF/LF

TXS0108E (HW-0108)
8-channel,
bidirectional 5V-3.3V
logic level converter

Arduino UNO Controller

lamaPLC.com

Arduino Example Code (using RadioLib)

Install **RadioLib** via the Arduino Library Manager before uploading this code.

```
#include <RadioLib.h>

// Pin definitions for Waveshare Core1262
// CS, DI01, RESET, BUSY
SX1262 radio = new Module(10, 2, 9, 8);

void setup() {
  Serial.begin(9600);

  // Initialize SX1262
  // Frequency: 868.0 MHz (HF) or 433.0 MHz (LF)
  Serial.print(F("[SX1262] Initializing ... "));
  int state = radio.begin(868.0); // Change to 433.0 for LF version

  if (state == RADIOLIB_ERR_NONE) {
    Serial.println(F("success!"));
  } else {
    Serial.print(F("failed, code "));
    Serial.println(state);
    while (true);
  }
}

void loop() {
```

```
Serial.print(F("[SX1262] Transmitting packet ... "));

// Send a message
int state = radio.transmit("Hello Waveshare!");

if (state == RADIOLIB_ERR_NONE) {
  Serial.println(F("success!"));
  // Print measured data rate
  Serial.print(F("[SX1262] Datarate:\t"));
  Serial.print(radio.getDataRate());
  Serial.println(F(" bps"));
} else {
  Serial.print(F("failed, code "));
  Serial.println(state);
}

// Wait 2 seconds before next transmission
delay(2000);
}
```

Key Implementation Details

HF vs. LF: Ensure you set the *radio.begin()* frequency to match your hardware (e.g., 868.0 for HF in Europe, 915.0 for HF in the USA, or 433.0 for LF).

- **Busy Pin:** Unlike older SX127x modules, the SX1262 requires monitoring the BUSY pin; the library handles this automatically if the pin is correctly assigned in the constructor.
- **TCXO Support:** Waveshare modules often use a TCXO. If the module fails to initialize, you may need to enable TCXO voltage in your setup using *radio.setTCXO(1.6)*;

Waveshare RP2040-LoRa

The Waveshare RP2040-LoRa is a compact development board that combines Raspberry Pi's RP2040 microcontroller with the next-generation SX1262 RF chip for long-range wireless communication. It is designed for low-power IoT applications such as smart homes, industrial monitoring, and data acquisition.



Key Specifications

- **Microcontroller:** Dual-core Arm Cortex M0+ running up to 133 MHz.
- **Memory:** 264KB SRAM and 2MB onboard Flash memory.
- **RF Chip:** SX1262, which offers better power efficiency and range than the older SX127X series.
- **Sensitivity:** High reception sensitivity of -148dBm.
- **Programmable Power:** Up to 22 dBm output power.
- **Modulation:** Supports LoRa, FSK, and GFSK.
- **GPIO:** Up to 20 multi-function GPIO pins.

Frequency Versions

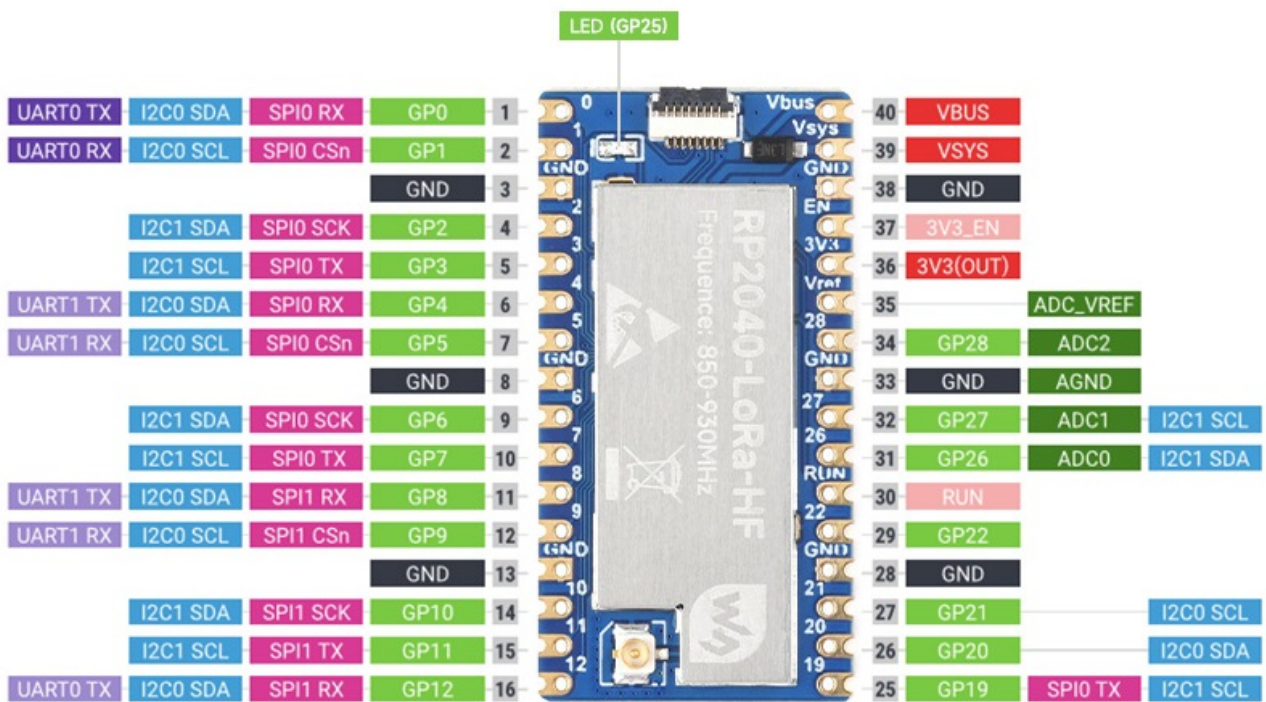
The board is available in two versions tailored for different regional regulations:

- **RP2040-LoRa-LF:** Low-frequency version (410 ~ 525 MHz), typically used in Europe and Asia.
- **RP2040-LoRa-HF:** High-frequency version (850 ~ 930 MHz), used in Europe, North America, and Oceania.

Development Resources

- **Programming:** Supports drag-and-drop programming over USB and can be developed using C/C++ (SDK provided) or Arduino (via libraries like RadioLib).
- **Cloud Integration:** Compatible with LoRa gateways for connecting to cloud servers like *The Things Network (TTN)* and *ChirpStack*.
- **Documentation:** Detailed guides and example codes are available on the [Waveshare Wiki](#).

Waveshare RP2040-LoRa Pinout



Waveshare SX1262/SX1268 LoRa HAT

The **Waveshare SX1262/SX1268 LoRa HAT** is a Raspberry Pi expansion board designed for long-range wireless data transmission. It functions as a wireless serial port module, allowing devices like a Raspberry Pi, Arduino, or STM32 to communicate over several kilometers using LoRa modulation.



Key Specifications

- **Communication Range:** Up to 5 km under ideal conditions.
- **Chipsets & Frequencies:**
 - **SX1262:** Typically used for 868MHz (Europe) and 915MHz (Americas/Asia).
 - **SX1268:** Typically used for 433MHz and 470MHz (China/Asia).
- **Protocol:** Uses a private protocol for point-to-point or point-to-multipoint communication; it does not natively support LoRaWAN.
- **Interface:** Connects via the standard Raspberry Pi 40-pin GPIO or via an onboard **CP2102 USB-to-UART** converter for serial debugging.

Core Features

- **Operating Modes:** Controlled via jumpers (M0 and M1), supporting Transmission, Configuration, Wake-on-Radio (WOR), and Deep Sleep modes.
- **Advanced Networking:** Supports auto multi-level repeating (relay) to extend transmission distance and fixed-point/broadcast transmission.
- **Security & Reliability:** Features a customizable communication key for encrypted transmission and Listen Before Talk (LBT) to monitor channel noise before sending.
- **Indicators:** Includes 4 LED indicators for monitoring power (PWR), RX/TX status, and auxiliary (AUX) activity.

Lora topics on lamaPLC

Page	Date	Tags
• lamaPLC Communication: IoT	2026/04/23 21:51	communication , iot , internet , iomt , 6lowpan , ipv4 , ipv6 , bluetooth , ble , li-fi , nfc , rfid , wi-fi , zigbee , z-wave , lte-advanced , 5g , lora , dash7 , lpwan , lorawan , sigfox , nb-iot , weightless , rpma , mioty , vsat , ethernet , thread , matter
• lamaPLC: AI-Thinker LoRA products	2026/04/23 21:51	ai-thinker , lora manufacturer , communication , lora , modul , ra-01 , ra-02 , spi , arduino
• LamaPLC: Arduino MKR Family	2025/08/24 12:19	arduino , board , mkr fox 1200 , mkr zero , ethernet , shield , m-bus , rs-485 , can , env , sd card , rgb , gps , iot , wifi , sigfox , lora , gsm , lte-m
• lamaPLC: Ebyte LoRA products	2026/04/23 21:51	ebyte , lora manufacturer , communication , lora , modul
• lamaPLC: Waveshare LoRA products	2026/03/07 01:46	waveshare , lora manufacturer , communication , lora , modul , usb-to-lora-xf02 , core 1262 , 1262 , spi , arduino , rp2040-lora , rp2040

[Waveshare](#), [Lora Manufacturer](#), [communication](#), [lora](#), [modul](#), [USB-TO-LoRa-xF02](#), [Core 1262](#), [1262](#), [SPI](#), [Arduino](#), [RP2040-LoRa](#), [RP2040](#)

This page has been accessed for: Today: 1, Until now: 116

From:

<http://lamaplc.com/> - **lamaPLC**

Permanent link:

http://lamaplc.com/doku.php?id=com:waveshare_products

Last update: **2026/04/21 20:46**

