

Tableau des tests des cartes ESP32_LoRa_OLED en émission/réception :

Sélection de "Esp32 Dev Module" dans l'IDE d'Arduino.

Un sketch "ESP32_LoRa_Initialisation" a été mis au point pour purger la carte entre deux tests différents. Il affiche : ESP32 LoRa Device//Device initialised !

Origine	Sketches	Corrections	Résultat
Site AliExpress (1)	868mhzOledLoRa Utilisation de "Wire.h"	Pour les deux cartes : LoRa.begin (BAND) au lieu de LoRa.begin(868) et suppression de image,h Pour le receveur : correction String au lieu de string.	L'émetteur affiche : Sending packet n avec n qui croît de 1 en 1. La LED bleue clignote. Le récepteur affiche : RSSI - n1//Received8bytes /hello n. Avec n1 = taille du paquet.
Site AliExpress (2)	"OLED_LoRa_Sender" et "OLED_LoRa_Reciver" Utilisation de "Wire.h"	Pour les deux cartes, nombre de bauds de 9600 à 115200. Le sender : la ligne « display.drawXbm(0,5,logo_width,logo_height,logo_bits); » sort en erreur à la compilation avec « Invalid conversion from char to const uint8_t ... ». On met la ligne en commentaire.	L'émetteur affiche : Sending packet: n avec n qui croît de 1 en 1. La LED bleue clignote. Le récepteur affiche : RSSI - n1//Received8bytes/H ello n. Avec n1 = taille du paquet.
Article "LoRa and the ESP32 SX1276.doc" Ligne "ZIP of Send and Receive Sketches for testing". On télécharge LoRa.zip	"ESP32_LoRa_Send" et "ESP32_LoRa_Rec_OLED" Utilisation de "arduino.h"	Pour les deux cartes : correction BAND 868E6. Le récepteur : quelques corrections, le programme n'affichant que les unités des nombres qui sont émis. Corrections faites dans le loop() : . Paragraphe "try to parse packet", supprimer la ligne "u8x8..."; . Paragraphe "read packet", corriger la ligne : "receivedText = LoRa.readString()" corriger la dernière ligne : "u8x8.drawString(0, 4, currentid)" ajouter "u8x8.clearLine(4);" en avant dernière ligne.	L'émetteur affiche : Sending packet/n avec n croissant de 1 en 1. Le récepteur affiche : LoRa Receiver//Hello n/PackageRS -n1. Avec n1 = taille du paquet. Le récepteur affiche "LoRa Receiver" dès le branchement à une source d'énergie.
Tests avec breadBoard			
Sketch Blink de la bibliothèque Arduino	On le renomme : "ESP32_LoRa_allume_Led" Test avec Led + résistance 220 ohms.	Variable LED_ BUILTIN renommée ledTest et branchée : a) sur port 2 (borne 4à droite). b) sur port 23 (borne 8d) (int ledTest = 23 ou #define LedTest 23 au choix).	Clignotement de la LED. + clignotement de la LED bleue si port 2 branché (GPIO LED). (3)

Sketch DS18x20_Temperature de la bibliothèque Arduino	On le renomme : "ESP32_DS18x20_Temp" Montage avec capteur filaire DS18B20. Sortie jaune su GND, rouge sur 3v3 et verte sur le port 21. Résistance de 4,7 k ohms.	delay(1000) au lieu de 250 et Serial.begin à 115200. OneWire ds(21) sur borne 1g.	Affichage, sur le moniteur uniquement, de la température que l'on fait varier au toucher.
Le sketch précédent complété par ajout du sketch ESP32_LoRa_Send	On le nomme : "ESP32_LoRa_TempSender" Test avec "ESP32_LoRa-Rec-OLED"	néant	L'émetteur affiche : Sending packet dd.cc (exemple 22.56) ESP32_LoRa-Rec-OLED affiche : "LoRa Receiver//Hello nn.pp/PackageRS - n1" avec, par exemple, nn.pp = 22.56 et - n1 = -50 (4)

1. Ces sketches ont été fournis avec le descriptif des cartes. Le second "sender" est, en réalité, un receveur. On les nomme 868mhzOledLoRaSender et 868mhzOledLoRaReceiver.

2. AliExpress propose, en complément de ceux du descriptif, un couple de sketches de test à l'adresse : <https://github.com/LilyGO/TTGO-LORA32-V2.0/tree/LilyGO-868-V1.0>

On y télécharge TTGO-LORA32-V2.0-LilyGO-868-V1.0.zip d'où l'on extrait les deux sketches (à noter que ce couple est aussi fourni dans l'article "LoRa and the ESP32 SX1276.doc").

3. Avec le même montage et le sketch "ESP32_LoRa_Initialisation" la LED s'allume (feu fixe) sur les bornes 3 (port 4 OLED SDA), 5 (port 15 OLED SCL), 6 (port 5 LoLa SCK), 11 (port 9 CLK1), 12 (port RST), 13 (port TX CLK3), 14 (port RX CLK2). Allumage vérifié aux bornes 16 (port 3V3) et 17 (port 5v) de chaque face.

4. Exemple d'affichage sur le moniteur, en défilement :

```
ROM = 28 D9 E5 ...
Chip = DS18B20
Data = 1 +69 1 4B 46
Temperature = 22.56 Celsius 72.61 Fahrenheit
Sending packet : 22.56
No more addresses
```