Communication LR entre 2 ESP32

Librairie à installer :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Gestionnaire de carte :

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Pour une communication entre 2 esp32 il faut établir un émetteur et un récepteur. Il faut que l’émetteur connaisse l’adresse MAC du récepteur pour cela, il faut utiliser le programme ci-dessous pour la récupérer :

#include "WiFi.h"

void setup(){

Serial.begin(115200);

WiFi.mode(WIFI\_MODE\_STA);

Serial.println(WiFi.macAddress());

}

void loop(){

}

Lors du lancement du programme, il se peut que Arduino affiche une erreur il suffit de relancer le programme et d’appuyer sur le bouton boot longtemps le temps que Arduino lance les % d’écritures.

Suite à ça, il faut écrire un code émetteur et un code récepteur en mettant bien l’adresse mac trouvé dans le code émetteur.

**Récepteur :**

//Recepteur

#include "esp\_now.h"

#include "WiFi.h"

uint8\_t HELTECAdresse []={0x24, 0x0A, 0xC4, 0x0D, 0x8D, 68};

void setup(){

// Init Serial Monitor

Serial.begin(115200);

// Set ESP32 as a WIFI Station

WiFi.mode(WIFI\_STA);

// Init ESP-NOW

esp\_now\_init();

//Fonction callback

esp\_now\_register\_recv\_cb(OnDataRecv);

}

typedef struct struct\_message

{

char a[32];

int b;

float c;

String d;

bool e;

}struct\_message;

struct\_message myData;

void OnDataRecv(const uint8\_t \*mac, const uint8\_t \*incomingData, int len){

memcpy(&myData, incomingData, sizeof(myData));

Serial.println("Donnees recues");

}

void loop(){

Serial.print("b : ");

Serial.print(myData.b,DEC);

Serial.print("\t");

}

**Emetteur :**

//Emetteur

#include "esp\_now.h"

#include "WiFi.h"

uint8\_t HELTECAdresse []={0x24, 0x0A, 0xC4, 0x0D, 0x8D, 68};

void setup(){

// Init Serial Monitor

Serial.begin(115200);

// Set ESP32 as a WIFI Station

WiFi.mode(WIFI\_STA);

// Init ESP-NOW

if(esp\_now\_init()==ESP\_OK)

{

Serial.println("Initialisation ESP OK");

}

else{

Serial.println("Erreur initialisation ESP");

ESP.restart();

}

//Fonction callback

esp\_now\_register\_send\_cb(OnDataSent);

//Strucute Appairage INFO

struct esp\_now\_peer\_info peerInfo;

memcpy(peerInfo.peer\_addr, HELTECAdresse, 6);

peerInfo.channel=0;

peerInfo.encrypt=false;

// test

if(esp\_now\_add\_peer(&peerInfo)==ESP\_OK)

{Serial.println("Fonction peer OK");}

else

{Serial.println("Erreur fonction peer");}

}

//Information sur l'envoi de la data

void OnDataSent(const uint8\_t \*mac\_addr, esp\_now\_send\_status\_t status){

if(status == ESP\_NOW\_SEND\_SUCCESS)

{

Serial.println("livraison OK");}

else{Serial.println("OK");}

}

typedef struct struct\_message

{

char a[32];

int b;

float c;

String d;

bool e;

}struct\_message;

struct\_message myData;

void loop(){

strcpy(myData.a,"SALUT TWAAA");

myData.b = random(1,20);

myData.c = 1.2;

myData.d = "HELLO";

myData.e= false;

esp\_err\_t result=esp\_now\_send(HELTECAdresse,(uint8\_t\*)&myData,sizeof(myData));

if (result==ESP\_OK)

{Serial.println("Envoi OK");}

else{Serial.println("erreur Envoi");}

delay(500);

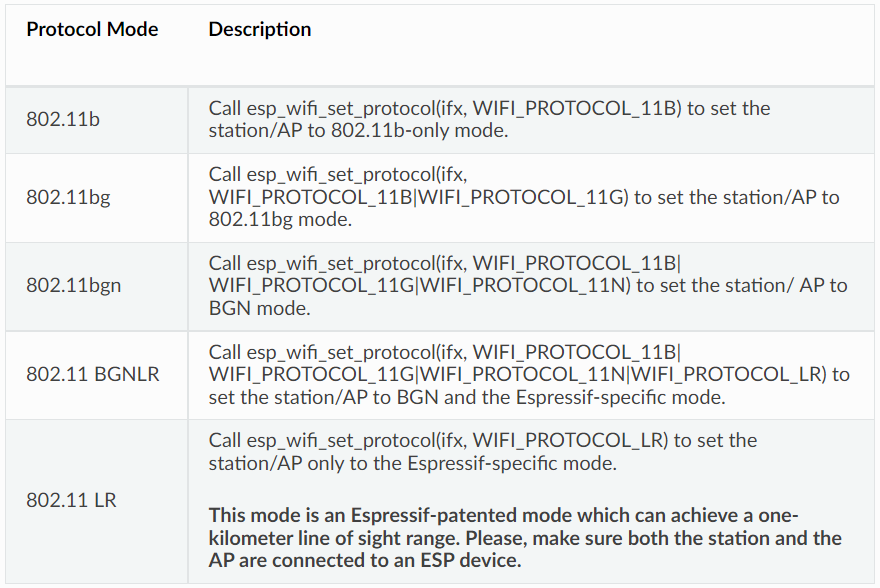
}

Voila les programmes qu’on à effectuer qui ne fonctionne pas complétement, cependant on arrive à écrire sur le moniteur série, il ne doit pas manquer grands choses pour avoir la communication entre les 2 ESP32.

Après avoir réussi à faire communiquer les ESP32, il faudrait suivre la partie qui suit pour effectuer la communication LR :

Partie Communication LR :

La communication LR va avec le mode 802.11, ce sont des modes de protocoles wifi :



Le mode Longue Portée (LR) est un mode Wi-Fi unique d’Espressif cela veut dire que seul les appareils ESP32 peuvent transmettre et recevoir les données. De plus, le mode peut atteindre une portée d’environ 1 Km.

Le **débit LR** a un débit très limité car le débit de données PHY brut LR est de ½ Mbits et ¼ Mbits.

Pour utiliser LR :

* Le point d’accès et la station sont tous deux des appareils.
* Une connexion Wi-Fi longue distance et une transmission de données sont nécessaires
* Les exigences de débit de données sont très faibles, telles que le contrôle a distance des périphériques, etc

L’activation du mode est un simple appel de fonction pour basculer votre ESP32 en mode LR voici un exemple :

void init\_wifi(wifi\_mode\_t mode)

{

const uint8\_t protocol = WIFI\_PROTOCOL\_LR;

tcpip\_adapter\_init();

ESP\_ERROR\_CHECK( esp\_event\_loop\_init(event\_handler, NULL) );

wifi\_init\_config\_t cfg = WIFI\_INIT\_CONFIG\_DEFAULT();

ESP\_ERROR\_CHECK( esp\_wifi\_init(&cfg) );

ESP\_ERROR\_CHECK( esp\_wifi\_set\_storage(WIFI\_STORAGE\_RAM) );

ESP\_ERROR\_CHECK( esp\_wifi\_set\_mode(mode) );

wifi\_event\_group = xEventGroupCreate();

if (mode == WIFI\_MODE\_STA) {

ESP\_ERROR\_CHECK( esp\_wifi\_set\_protocol(WIFI\_IF\_STA, protocol) );

wifi\_config\_t config = {

.sta = {

.ssid = ap\_name,

.password = pass,

.bssid\_set = false

}

};

ESP\_ERROR\_CHECK( esp\_wifi\_set\_config(WIFI\_IF\_STA, &config) );

ESP\_ERROR\_CHECK( esp\_wifi\_start() );

ESP\_ERROR\_CHECK( esp\_wifi\_connect() );

xEventGroupWaitBits(wifi\_event\_group, CONNECTED\_BIT,

false, true, portMAX\_DELAY);

ESP\_LOGI(TAG, "Connected to AP");

} else {

ESP\_ERROR\_CHECK( esp\_wifi\_set\_protocol(WIFI\_IF\_AP, protocol) );

wifi\_config\_t config = {

.ap = {

.ssid = ap\_name,

.password = pass,

.ssid\_len = 0,

.authmode = WIFI\_AUTH\_WPA\_WPA2\_PSK,

.ssid\_hidden = false,

.max\_connection = 3,

.beacon\_interval = 100,

}

};

ESP\_ERROR\_CHECK( esp\_wifi\_set\_config(WIFI\_IF\_AP, &config) );

ESP\_ERROR\_CHECK( esp\_wifi\_start() );

}

}