
Projet n°1814 TotemLumineux

ETML-ES

Cours : POBJ

Réaliser par : Joao Marques (joao.marques@eduvaud.ch)

Classe : SLO2 2022-2023

Introduction

Pour notre deuxième année, pendant le cours de POBJ, il nous a été demandé de réaliser un petit projet au milieu du semestre qui avait pour but de mettre en pratique ce que nous avons vu en théorie. Ce projet nous permettait de concrétiser nos connaissances et de mieux les comprendre grâce à sa dimension réaliste.

Table des matières

1	Cahier des charges	2
1.1	Cahier des charges initial	2
1.2	Cahier des charges mise a jour	2
2	Tâches réalisées	2
3	Programmation	4
3.1	Flow chart	4
4	Test	5
5	Etat d'avancement	6
6	Journal de travail	6
7	Conclusion	7
8	Annexe	8

1 Cahier des charges

Avant d'établir le cahier des charges, voici l'état d'avancement actuel du projet : le Totem fonctionne correctement et affiche des messages sur les matrices. Cependant, la communication par WiFi est manquante et il est recommandé de changer de module, selon les conseils de Nicola Fürst. Malgré cette lacune, le projet respecte les principales exigences du cahier des charges initial, notamment l'affichage de messages, d'animations, la surveillance du niveau de la batterie et l'ajustement de la luminosité en fonction de la lumière ambiante. La partie liée à la batterie et au panneau solaire a été mise de côté en raison du coût élevé des composants et du manque de temps disponible.

1.1 Cahier des charges initial

À la reprise du projet, l'objectif était de faire fonctionner le module WiFi de Microchip afin de pouvoir envoyer et récupérer des messages à travers ce module. Une fois la partie WiFi réussie, il était demandé de réaliser un programme permettant de modifier les configurations de l'affichage, notamment le texte, la couleur et les animations.

1.2 Cahier des charges mise a jour

Suite à des contraintes de temps, une mise à jour du cahier des charges a été effectuée. La réalisation du programme permettant de modifier l'affichage a été retirée, et seule la partie de communication avec le module Microchip reste à accomplir.

2 Tâches réalisées

Tout d'abord, il a fallu prendre connaissance du projet et réaliser la fiche de modification. Pour ce projet, il s'agit d'un Totem qui a la capacité d'afficher un message coloré ou une animation à une vitesse et luminosité variables sur deux matrices souples composées de 512 LEDs RGB. Il peut également mesurer le niveau de son alimentation et possède un boîtier qui peut être imperméabilisé. En ce qui concerne la fiche de modification, j'ai répertorié les modifications nécessaires qui correspondent à mon cahier des charges, la fiche de modification se trouve en Annexe.

Ensuite, il a fallu effectuer des recherches sur le module Wi-Fi. L'objectif était de trouver des informations au niveau matériel afin de pouvoir vérifier si cela a été correctement réalisé.

La Figure 1 montre les connexions minimales nécessaires pour communiquer avec le module. Maintenant, il reste à rechercher les connexions relatives à l'interface SPI.

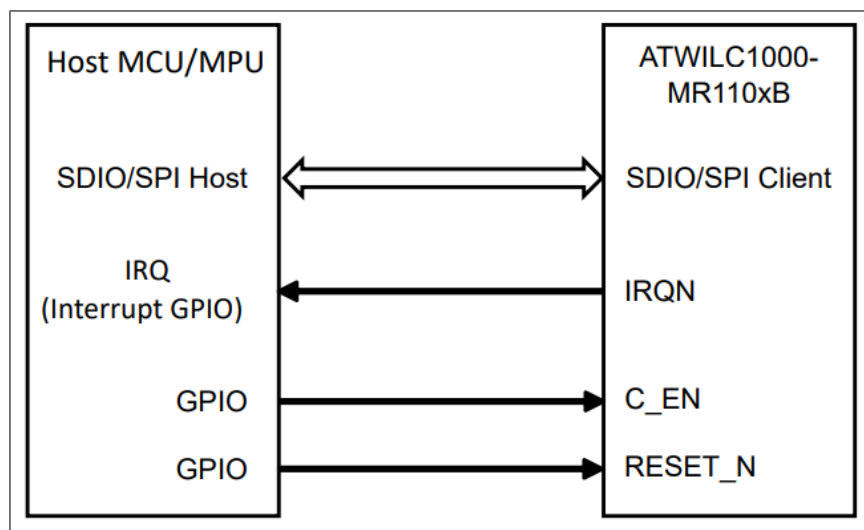


FIGURE 1 – Connexion utile selon datasheet

Avec les Figures 2 et 3, on peut identifier les broches qui doivent être utilisées. Dans notre cas, puisque nous souhaitons communiquer en SPI, nous aurons besoin des broches 4, 13, 22, 15-18.

Module Pin	Function ⁽¹⁾
4	RESET_N
13	IRQ_N
22	CHIP_EN
16	SPI_SSN/SD_DATA1
15	SPI_MOSI/SD_DATA2

FIGURE 2 – Tableaux Pin du ATW

.....continued	
Module Pin	Function ⁽¹⁾
17	SPI_MISO/SD_DATA0
18	SPI_SCK/SD_CMD
14	SD_DATA3
19	SD_CLK

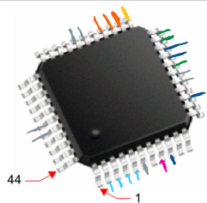
FIGURE 3 – Tableaux Pin du ATW

Maintenant que nous avons trouvé les informations nécessaires au niveau du matériel, j'ai vérifié la partie matérielle du module WiFi sur le PCB. En utilisant la Figure 4, nous pouvons voir les broches utilisées sur le PIC32. Nous constatons qu'il y a bien 4 broches pour la communication SPI, ainsi que 4 broches GPIO utilisées, correspondant aux broches IRQN, C EN, RESET N, et une connexion supplémentaire sur la broche WAKE du module ATW.

TABLE 11: PIN NAMES FOR 44-PIN GENERAL PURPOSE DEVICES

44-PIN TQFP (TOP VIEW)^(1,2,3,5)

**PIC32MX110F016D
PIC32MX120F032D
PIC32MX130F064D
PIC32MX130F256D
PIC32MX150F128D
PIC32MX170F256D**



Pin #	Full Pin Name	Pin #	Full Pin Name
1	RPB9/SDA1/CTED4/PMD3/RB9	23	AN4/C11NB/C2IND/RPB2/SDA2/CTED13/RB2
2	RPC6/PMA1/RC6	24	AN5/C11NA/C2INC/RTCC/RPB3/SL2/RB3
3	RPC7/PMA0/RC7	25	AN6/RPC0/RC0
4	RPC8/PMA5/RC8	26	AN7/RPC1/RC1
5	RPC9/CTED7/PMA6/RC9	27	AN8/RPC2/PMA2/RC2
6	VSS	28	VDD
7	VCAP	29	VSS
8	PGED2/RPB10/CTED11/PMD2/RB10	30	OSC1/CLK1/RPA2/RA2
9	PGEC2/RPB11/PMD1/RB11	31	OSC2/CLK0/RPA3/RA3
10	AN12/PMD0/RB12	32	TD0/RPA8/PMA8/RA8
11	AN11/RPB13/CTPLS/PMRD/RB13	33	SOSCI/RPB4/RB4
12	PGED4 ⁽⁴⁾ /TMS/PM10/RA10	34	SOSCO/RPA4/T1CK/CTED9/RA4
13	PGEC4 ⁽⁴⁾ /TCK/CTED8/PM7/RA7	35	TDI/RPA9/PM9/RA9
14	CVREFOUT/AN10/C3INB/RPB14/SCK1/CTED5/PMWR/RB14	36	RPC3/RC3
15	AN9/C3INA/RPB15/SCK2/CTED6/PMCS1/RB15	37	RPC4/PMA4/RC4
16	AVSS	38	RPC5/PMA3/RC5
17	AVDD	39	VSS
18	MCLR	40	VDD
19	VREF+/CVREF+/AN0/C3INC/RPA0/CTED1/RA0	41	PGED3/RPB5/PMD7/RB5
20	VREF-/CVREF-/AN1/RPA1/CTED2/RA1	42	PGEC3/RPB6/PMD6/RB6
21	PGED1/AN2/C1IND/C2INB/C3IND/RPB0/RB0	43	RPB7/CTED3/PMD5/INT0/RB7
22	PGEC1/AN3/C1INC/C2INA/RPB1/CTED12/RB1	44	RPB8/SL1/CTED10/PMD4/RB8

IO Module Wifi

SPI Module Wifi

Alimentation

SV Enable

Pin de programmation

Entrée analogique

Sortie Led

FIGURE 4 – Choix pin de Nicola Fürst

Pour mieux vérifier si les broches nécessaires sont utilisées, j'ai récupéré une image du datasheet qui montre les différentes broches du module ATW, et j'ai marqué les broches utilisées.

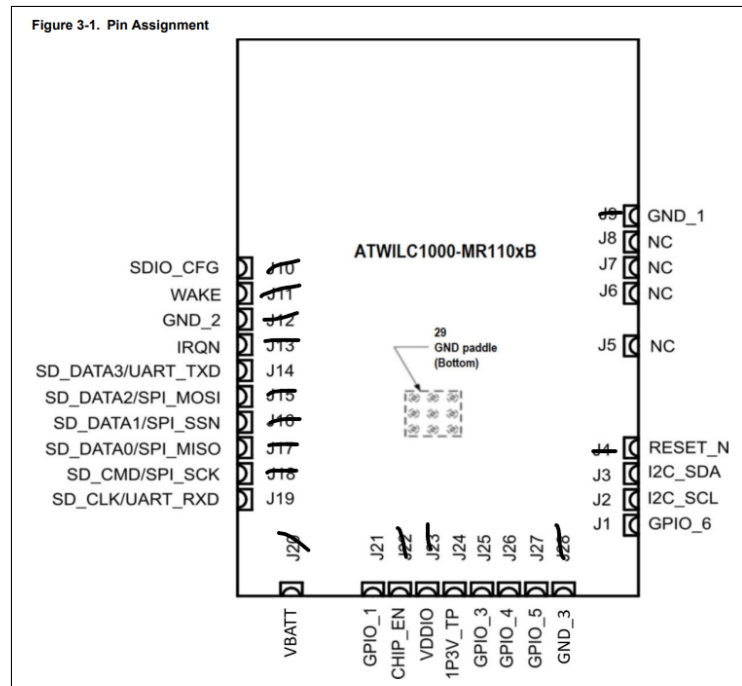


FIGURE 5 – Pin utilisé du module wifi

Avec la figure 5, nous pouvons constater que toutes les connexions dont nous avons besoin sont bien utilisées, ce qui nous laisse penser que la communication devrait être possible.

Ensuite, après avoir vérifié la partie matérielle du module, j'ai entrepris des recherches sur la communication. À cet effet, j'ai trouvé des documents User Guide qui présentent des exemples de communication sous Linux, que vous trouverez en annexe.

Enfin, en raison du manque de temps, j'ai décidé de créer un organigramme (flow chart) pour la partie programmation, en m'appuyant sur les documents User Guide.

3 Programmation

Comme mentionné précédemment et compte tenu du temps restant, j'ai décidé de privilégier la réalisation de Flow chart plutôt que la programmation.

3.1 Flow chart

Dans le diagramme de flux (voir Figure 7 en annexe), nous pouvons voir les étapes impliquées dans l'utilisation du module Wi-Fi ATWILC1000. Voici une explication détaillée de chaque étape :

Initialiser le microcontrôleur PIC32 : Cette étape consiste à initialiser le microcontrôleur avant de pouvoir utiliser le module Wi-Fi.

1. Configurer les broches pour la communication avec le module Wi-Fi : On doit configurer les broches du microcontrôleur pour permettre la communication avec le module Wi-Fi, telles que les broches de données, d'alimentation, de contrôle, etc.
2. Initialiser le module Wi-Fi ATWILC1000 : Cette étape implique l'initialisation du module Wi-Fi lui-même en configurant les paramètres de communication et les modes de fonctionnement.
3. Tenter la connexion au réseau Wi-Fi : Nous tentons d'établir une connexion avec le réseau Wi-Fi en utilisant les paramètres appropriés tels que le SSID (nom du réseau) et la clé de sécurité.
4. Vérifier l'état de connexion : Une fois la tentative de connexion effectuée, nous vérifions si la connexion a été établie avec succès.
5. Si la connexion réussit : Si la connexion est établie, nous passons à l'étape suivante. Sinon, nous gérons les erreurs de connexion.

6. Envoyer/recevoir des données via Wi-Fi : Si la connexion est réussie, nous pouvons envoyer et recevoir des données via Wi-Fi selon les besoins de notre application.
7. Gérer les interruptions Wi-Fi : Nous devons mettre en place une gestion appropriée des interruptions pour traiter les événements Wi-Fi en temps réel.
8. Si besoin de déconnexion : Si notre application nécessite une déconnexion du réseau Wi-Fi à un moment donné, nous pouvons inclure cette étape.
9. Se déconnecter du réseau Wi-Fi : Si nous avons besoin de nous déconnecter du réseau Wi-Fi, nous effectuons cette étape pour mettre fin à la connexion.
10. Arrêt du module Wi-Fi ATWILC1000 : Une fois que nous avons terminé d'utiliser le module Wi-Fi, nous pouvons l'arrêter pour économiser de l'énergie ou libérer des ressources.
11. Fin : Nous avons terminé l'utilisation du module Wi-Fi ATWILC1000 et nous pouvons passer à d'autres tâches ou arrêter complètement le microcontrôleur.

En annexe, vous trouverez également les flowcharts des différentes fonctions utiles, tels que la fonction d'initialisation (voir Figure 8), les fonctions de connexion (voir Figure 9), et les fonctions de traitement de données (voir Figure 10).

4 Test

Étant donné que l'implémentation du code n'a pas été réalisée, voici certains tests qui auraient pu être effectués pour l'utilisation du module Wi-Fi ATWILC1000 :

1. Test de connexion au réseau Wi-Fi :
Configurez les paramètres de connexion Wi-Fi, tels que le SSID (nom du réseau) et le mot de passe, dans le code. Assurez-vous que le module Wi-Fi est capable de se connecter au réseau en utilisant la fonction `connectToWiFi()`. Vérifiez si la connexion est réussie en surveillant le retour de la fonction et en affichant un message approprié.
2. Test d'envoi de données via Wi-Fi :
Configurez les paramètres d'envoi de données, tels que l'adresse IP et le port du serveur, dans le code. Utilisez la fonction `sendDataViaWiFi()` pour envoyer des données au serveur distant. Vérifiez si les données sont envoyées avec succès en surveillant les confirmations d'envoi ou en vérifiant si les données sont reçues par le serveur distant.
3. Test de réception de données via Wi-Fi :
Configurez les paramètres de réception de données, tels que l'adresse IP et le port du serveur, dans le code. Attendez les données entrantes en utilisant une boucle de réception ou une interruption. Lorsque des données sont reçues, traitez-les et affichez-les pour vérifier leur intégrité.
4. Test de déconnexion du réseau Wi-Fi :
Utilisez la fonction `disconnectFromWiFi()` pour vous déconnecter du réseau Wi-Fi. Assurez-vous que la déconnexion est réussie en surveillant le retour de la fonction et en affichant un message approprié.
5. Test de gestion des erreurs :
Simulez des scénarios d'erreur, tels qu'une mauvaise connexion, une perte de signal ou une déconnexion inattendue. Implémentez des mécanismes de gestion des erreurs appropriés, tels que des tentatives de reconnexion automatiques, des messages d'erreur et des actions de récupération.
6. Test de performances :
Mesurez le débit de données en envoyant des quantités variables de données et en chronométrant le temps de transmission. Évaluez la stabilité de la connexion en effectuant des tests de longue durée ou en soumettant le module Wi-Fi à des conditions environnementales difficiles.

5 Etat d'avancement

j'ai effectué des recherches pour faire fonctionner le module WiFi, et d'après ces recherches, le module devrait être fonctionnel. J'ai également passé du temps à comprendre son fonctionnement et à étudier sa documentation.

Ensuite j'ai également passé en revue la partie hardware du projet, en vérifiant les connexions et en m'assurant que tout était en ordre.

En ce qui concerne la communication, j'ai réalisé des recherches pour définir la meilleure approche, mais je n'est pas encore mis en place la partie pratique. Cependant, j'ai élaboré un flow chart détaillé pour planifier la logique du programme. Cela permettra par la suite de voir plus claire la manière dont la communication doit être réalisée.

À l'étape actuelle, il reste principalement le travail d'implémentation du logiciel pour permettre la communication avec le module WiFi. Et également le développement d'une application qui permettra la communication avec le Totem à l'avenir.

6 Journal de travail

Pour le journal de travail, les heures de travail ont été principalement consacrées pendant les cours de POBJ, en parallèle avec la réalisation des exercices de POBJ. La plage horaire était principalement de 9h50 à 11h25. Vous pouvez consulter mon journal de travail dans la figure 6.

DATE	HEURE	ÉVÉNEMENT
16.mars	9:50	Prendre connaissance du projet.
30.mars	9:50	Réaliser la fiche de modification.
06.avr	9:50	Recherche d'information pour le module wifi.
04.mai	9:50	Contrôlée la partie hardware du module wifi.
11.mai	9:50	Recherches sur la communication avec le module wifi.
25.mai	9:50	Réalisation du structogramme et début rapport.
01.juin	9:50	Réalisation du rapport.

FIGURE 6 – Journal de travail

7 Conclusion

En conclusion, ce projet avait pour but de reprendre un projet existant et d'apporter des modifications ou améliorations, voire dans mon cas, de terminer certaines parties du projet qui n'avaient pas été réalisées. Dans mon cas, il m'était demandé de réaliser la communication avec le module WiFi et, par la suite, si le module WiFi fonctionnel était réalisé, de créer un programme permettant la communication avec le totem lumineux. Malheureusement, en raison du manque de temps, je n'ai pas pu implémenter le code pour la

communication avec le module WiFi. Cependant, j'ai proposé des tests qui auraient pu être réalisés pour vérifier le bon fonctionnement du module en cas de réussite de la communication. J'ai également créé des flowcharts qui pourraient aider à la réalisation du code, ainsi que découvert des documents User Guide qui seront utiles au prochain élève. En conclusion, le projet m'a permis d'approfondir mes connaissances

théoriques en les appliquant à un cas pratique. Malgré le peu de temps disponible, j'ai quand même pu réaliser des avancées grâce à des recherches et j'ai également pu acquérir une meilleure compréhension du fonctionnement du module WiFi ATWILC1000.

8 Annexe

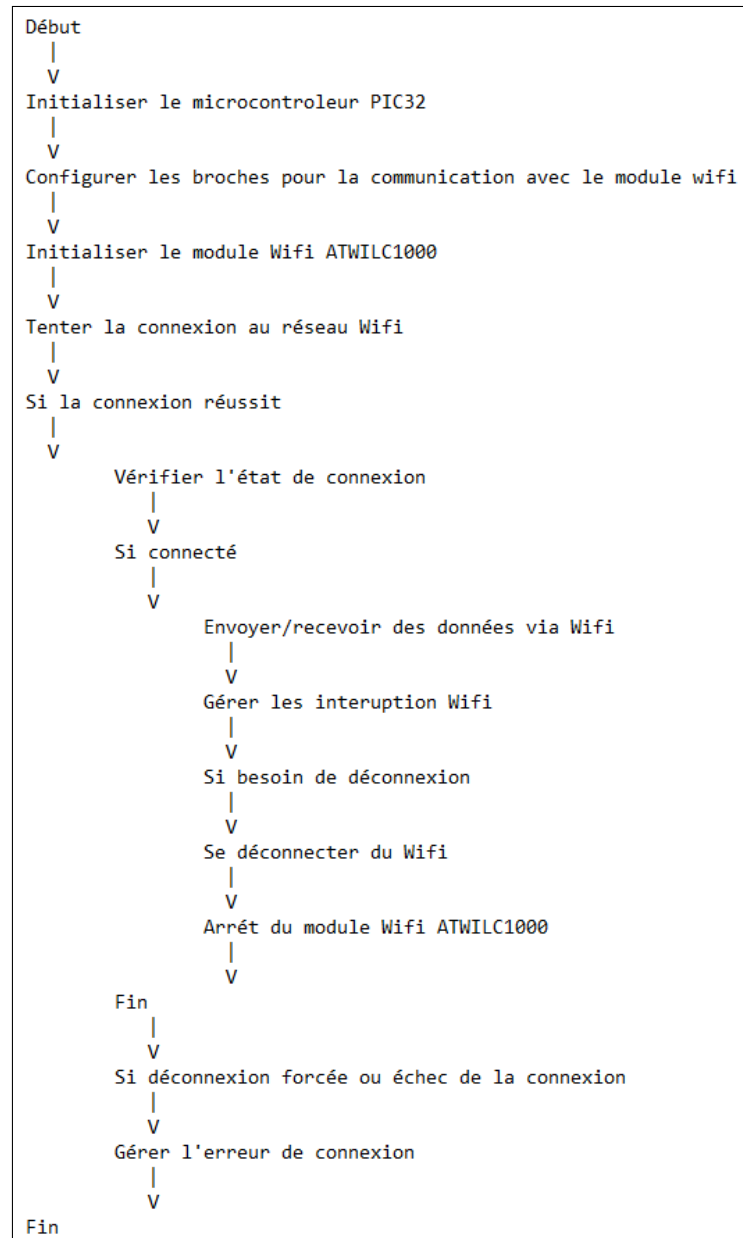


FIGURE 7 – Code principal

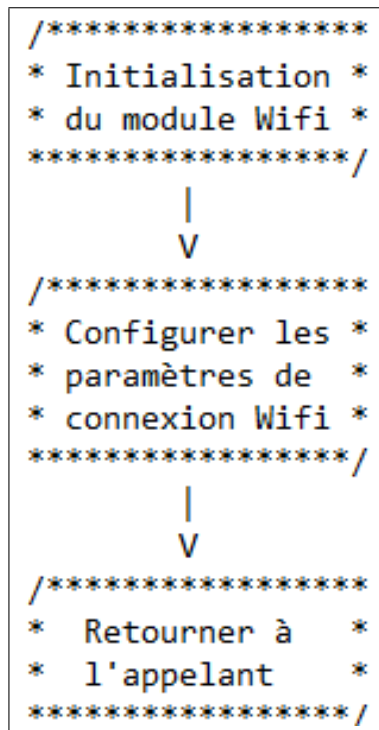
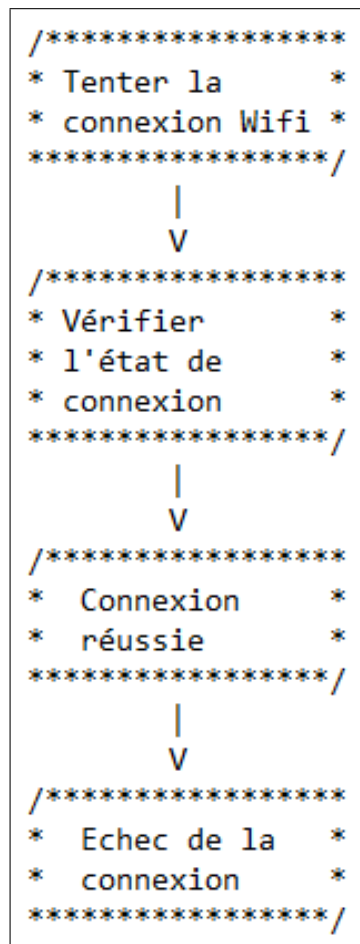
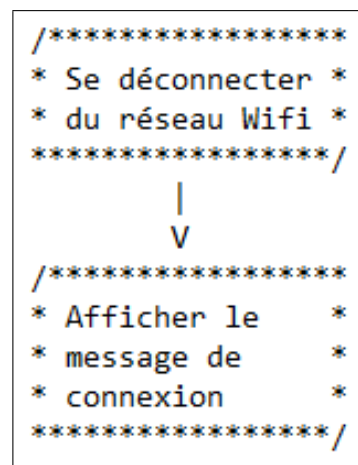


FIGURE 8 – Fonction initWifiModule



(a) Fonction connectToWifi



(b) Fonction disconnectFromWifi

FIGURE 9 – Fonction par rapport a la connexion

```

/*****
* Envoyer les      *
* données via      *
* Wifi             *
*****/
      |
      V
/*****
* Attendre la      *
* confirmation     *
* d'envoi          *
*****/
      |
      V
/*****
* Afficher les     *
* données          *
* envoyées         *
*****/

```

(a) Fonction sendDataViaWiFi

```

/*****
* Recevoir les     *
* données via      *
* Wifi             *
*****/
      |
      V
/*****
* Vérifier         *
* l'état de        *
* réception        *
*****/
      |
      V
/*****
* données reçus    *
*****/
      |
      V
/*****
* Afficher les     *
* données          *
* reçues           *
*****/

```

(b) Fonction receiveDataViaWiFi

FIGURE 10 – Fonction pour traitement de data