

HOS03 - INSTALLATION MODULE CAN/WIFI NODE-RED

1. Introduction.....	2
1.1 Objectif :.....	2
1.2 Compétences :.....	2
1.3 Ressources :.....	2
1.4 Compte rendu.....	2
1.5 Sécurité.....	2
2. Activités.....	3
2.1 Préparation.....	3
2.2 Etude du JoyIt bus CAN.....	5
2.3 Etude du ESP32 Adafruit Huzzah.....	5
2.4 Etude du SSD1306.....	5
2.5 Etude de l'alimentation.....	5
3. Assemblage du module CAN/WiFi.....	5
4. Mise en œuvre en Wifi.....	6
4.1 Programme pour l'ESP32.....	6
4.2 Mise en œuvre réseau WiFi rapide.....	6
4.3 Installation serveur.....	6
5. Correction.....	11
5.1 Etude du JoyIt bus CAN.....	11
5.2 Etude du ESP32 Adafruit Huzzah.....	11
5.3 Etude du SSD1306.....	11
5.4 Etude de l'alimentation.....	11

1. Introduction

1.1 Objectif :

Être capable d'installer et configurer le module électronique permettant l'envoi des données CAN vers un serveur SCADA.

1.2 Compétences :

1.3 Ressources :

Doc. Technique des modules AS. (NAS 172.19.2.129/sembat/...)

Schéma du module CAN/WiFi – IDE arduino – Code source INO.

Wireshark : sniffer logiciel.

Site internet : stssnsb.free.fr :Fiche bus CAN...

1.4 Compte rendu

Un compte rendu informatique sera réalisé sous LibreOffice et rendu en PDF.

Il contiendra toutes les réponses aux questions. Le compte rendu doit contenir tous les graphiques, schémas, tableaux, explications nécessaires. Il doit aussi contenir le rappel du numéro des questions.

ATTENTION : Votre cahier 'système' et les prises de notes manuscrites qu'il contient, vous permet de prendre toutes les notes nécessaire à la répétition de l'activité sans difficultés. Il faut donc le compléter régulièrement en plus du comte rendu.

1.5 Sécurité.

Le panneau comporte des pièces en mouvement. Pour votre sécurité et celle du matériel rester vigilant et REFLECHISSEZ avant d'agir.

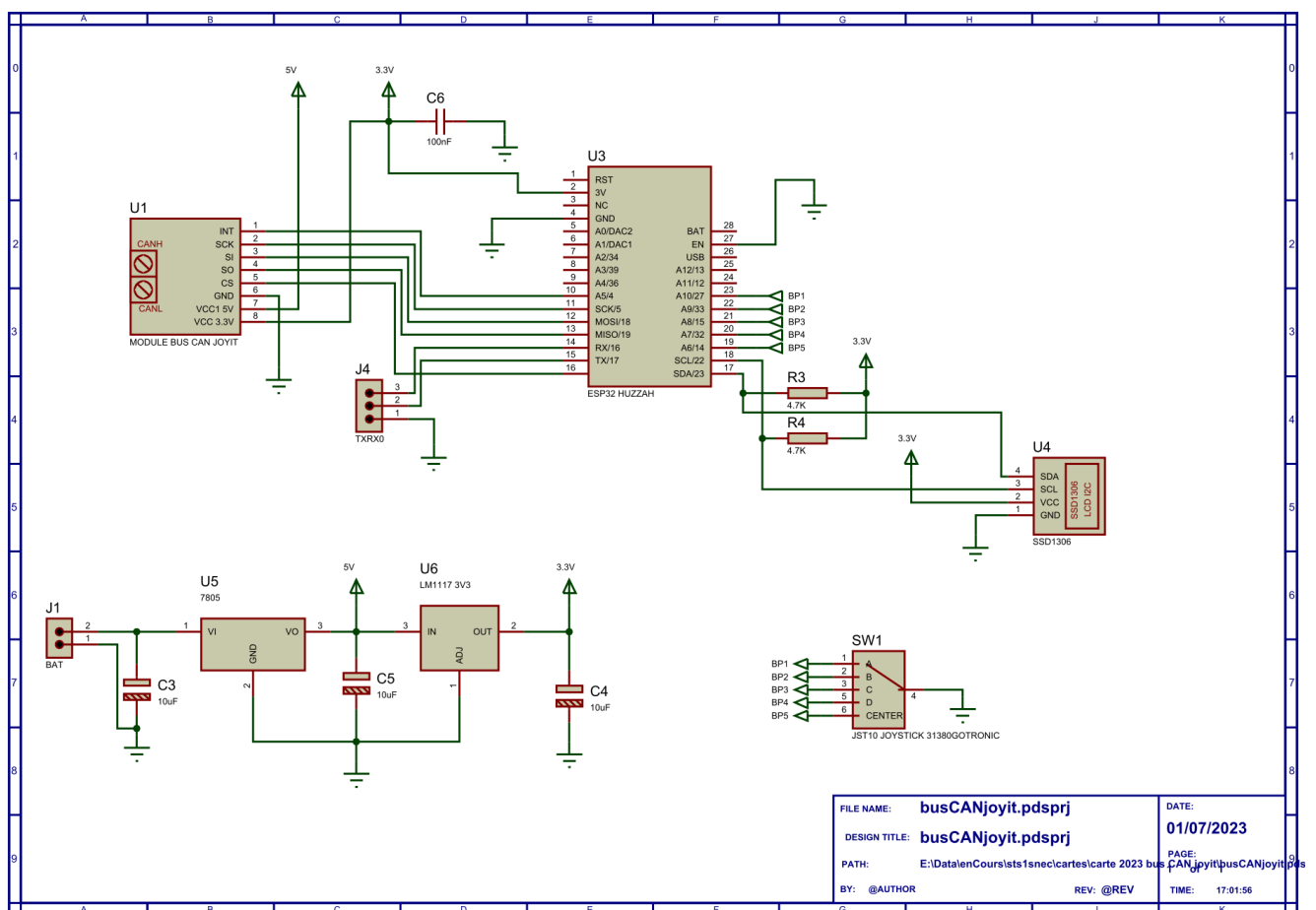
2. Activités

Le sous-système 'panneau soalrie' n'est pas connecté.

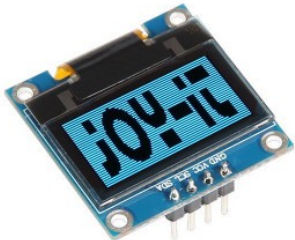
Votre mission est de le connecter afin de récupérer les mesures IPAN, UPAN, IBAT, UBAT sur un serveur distant. Les données seront alors stockées et exploitées (affichage graphique avec historisation des mesures)

2.1 Préparation

A partir du schéma du module CAN/WiFi répondre aux questions



2.1.1 Identifier les éléments présents dans la boîte module CAN/Wifi

Element réel	Nom	Référence sur schéma/rôle
	Module JoyIt bus CAN gotronic 35987	
	ESP32 Adafruit Huzzah gotronic 35911	
	SSD1306 gotronic 36688	
	LM1117 3.3V	
 	LM7805	

2.2 Etude du Joylt bus CAN

2.2.1 *A partir du schéma et du module réel, identifier les entrées et les sorties : type de bus, nom des broches*

2.2.2 *Donner le rôle de ce module*

2.3 Etude du ESP32 Adafruit Huzzah

2.3.1 *Identifier les entrées et les sorties utilisées dans le schéma :*

type de bus, nom des broches

2.3.2 *Donner le rôle de ce module*

2.4 Etude du SSD1306

2.4.1 *Identifier les entrées et les sorties :*

type de bus, nom des broches, ...

2.4.2 *Donner le rôle de ce module*

2.5 Etude de l'alimentation

2.5.1 *Quelle est la fonction des LM1117 et LM7805.*

3. Assemblage du module CAN/WiFi

A partir des informations précédentes :

3.1.1 *Assembler les différents éléments afin de concevoir le module CAN/WiFi*

3.1.2 *Faire vérifier à l'enseignant*

3.1.3 *Ouvrir l'IDE Arduino (l'installer s'il ne l'est pas (version 1.8 pas de version 2.xx)*

3.1.4 *Ouvrir le programme 'BUS_CAN_MCP2515_CAN_Receiver.ino' , le parcourir et le comprendre et configurer le débit, l'adresse SSD1306 et la broche CS du MCP2515.*

3.1.5 *A l'aide d'un cordon USB fournit par l'enseignant téléverser le programme.*

3.1.6 *Noter le comportement du programme. Vérifier qu'il satisfait à votre besoin.*

4. Mise en œuvre en Wifi

L'ESP32 se charge, avec un programme approprié, d'envoyer les données en WiFi vers un serveur.

Il faut donc :

- Mettre un programme d'envoi dans l'ESP32. (Code fournit ou conçu par vous)
- Mettre en œuvre un réseau Wifi (TP point accès WiFi)
- Installer un serveur récupérant les données et les traitant (SCADA) : TP futur

4.1 Programme pour l'ESP32

Ce programme :

lit les données CAN (UPAN, IPAN, IBAT, UBAT) et les envoie en UDP vers le serveur Node-red (PC WiFi ou PC + APWifi) .

Il affiche quelques informations utiles sur le LCD.

4.1.1 Vous devez adapter le programme à votre situation :

configuration du WiFi, débit du CAN, adresse LCD I2C...

4.1.2 Téléverser le programme modifié avec IDE arduino

4.1.3 Tester le bon fonctionnement avec le terminal série

4.2 Mise en œuvre réseau WiFi rapide

Pour gagner du temps, nous allons créer un réseau WiFi avec 1PC Wifi et un logiciel MyPublicWifi (gratuit)

4.2.1 Utiliser un PC AllInOne de la salle EN3.

4.2.2 Installer MyPublicWifi

4.2.3 Lancer le logiciel SSID : WIFI_AS1 ou WIFI_AS2 et motdepasse : WIFI_HOS1 et WIFI_HOS2 et configurer le point d'accès (pas d'accès internet).

4.2.4 Tester le fonctionnement avec VOTRE téléphone pour voir si la connexion est possible et fonctionne.

4.2.5 Lancer Wireshark sur le PC afin de visualiser les communications du réseau WiFi. (Fiche Wireshark – stssnsb.free.fr)

4.2.6 Visualiser la réception des données envoyées par le module CAN/WiFi

4.3 Installation serveur

On utilise le logiciel gratuit Node-red. (voir TP utilisation de Node-red)

4.3.1 Installer-le et configurer le en suivant le TP approprié

4.3.2 Créer un flow simple afin de recevoir les données Modbus. (exemple de flow fournit)

4.3.3 Vérifier que les données envoyées par le module CAN/WiFi se retrouvent bien sur Node-red.

4.3.4 Faire vérifier par l'enseignant.

Code source : BUS_CAN_MCP2515_CAN_Receiver.ino

```
/*
 * Adafruit MCP2515 FeatherWing CAN Receiver Example
 */
#include <Adafruit_MCP2515.h>
#define CS_PIN ??
// Set CAN bus baud rate
#define CAN_BAUDRATE ( ????)
Adafruit_MCP2515 mcp(CS_PIN);

//affichage
#include "SSD1306Wire.h" //pour OLEDSSD3306
SSD1306Wire display(0x ??, SDA, SCL);

/*****/
void setup() {
  Serial.begin(115200);
  while(!Serial) delay(10);
//init du oled LCD
  display.init();
  Serial.println("MCP2515 Receiver test!");

  if (!mcp.begin(CAN_BAUDRATE)) {
    Serial.println("Error initializing MCP2515.");
    while(1) delay(10);
  }
  Serial.println("MCP2515 chip found");
}
/*****/
void loop() {
  //LCD
  display.setFont(ArialMT_Plain_16);//font pour oled
  display.setTextAlignment(TEXT_ALIGN_CENTER); //aligne le texte
  display.drawString(40, 0, "BUS CAN RX\n\r"); //ecrit la phrase dans le buffer
  display.display(); //affiche sur l'Ã©cran le buffer
```



```
// try to parse packet
int packetSize = mcp.parsePacket();

if (packetSize) {
  // received a packet
  Serial.print("Received ");

  if (mcp.packetExtended()) {
    Serial.print("extended ");
  }

  if (mcp.packetRtr()) {
    // Remote transmission request, packet contains no data
    Serial.print("RTR ");
  }

  Serial.print("packet with id 0x");
  Serial.print(mcp.packetId(), HEX);

  if (mcp.packetRtr()) {
    Serial.print(" and requested length ");
    Serial.println(mcp.packetDlc());
  } else {
    Serial.print(" and length ");
    Serial.println(packetSize);

    // only print packet data for non-RTR packets
    while (mcp.available()) {
      //Serial.print((char)mcp.read());
      Serial.print(mcp.read());
    }
    Serial.println();
  }

  Serial.println();
}
} //fin loop
/*****/
```

--

5. Correction

5.1 Etude du Joylt bus CAN

5.1.1 A partir du schéma et du module réel, identifier les entrées et les sorties : type de bus, nom des broches

Bus CAN CAN+ , CAN- en entrée

bus SPI en sortie

5.1.2 Donner le rôle de ce module

Adapter et décoder le bus CAN pour interprétation par ESP32.

5.2 Etude du ESP32 Adafruit Huzzah

5.2.1 Identifier les entrées et les sorties utilisées dans le schéma :

type de bus : SPI vers module busCAN, RXTX, I2C vers SSD1306

SPI : CS = broche 21.

5.2.2 Donner le rôle de ce module

C'est le cerveau de la carte (microcontrôleur WiFi / BTH)

5.3 Etude du SSD1306

5.3.1 Identifier les entrées et les sorties :

type de bus, nom des broches, ...

bus I2C SDA, SCL – adresse 0x3C

5.3.2 Donner le rôle de ce module

Afficher des informations en provenance de l'ESP32.

5.4 Etude de l'alimentation

5.4.1 Quelle est la fonction des LM1117 et LM7805.

RIT : régulateur de tension : permet d'avoir une tension stable et fixe pour alimenter les composants électroniques . A partir du 12V batterie on fournit du 5V (module busCAN)et du 3V3 (uC + SSD)