TP ESP8266 CLIENT : TP PRISE EN MAIN

Objectifs :
Envoyer des données mesurées par la PICV2 vers un serveur

---

1. CONFIGURATION DU MODULE ESP8266

1.1. Liaison série

A l'aide de ESP8266Config configurer le module ESP01 avec les valeurs suivantes :
Choisir le COMxx correspondant à la liaison série USB/TTL5V
Bauds : 9600 (parfois 115200 sur un module neuf)

1.2. firmware

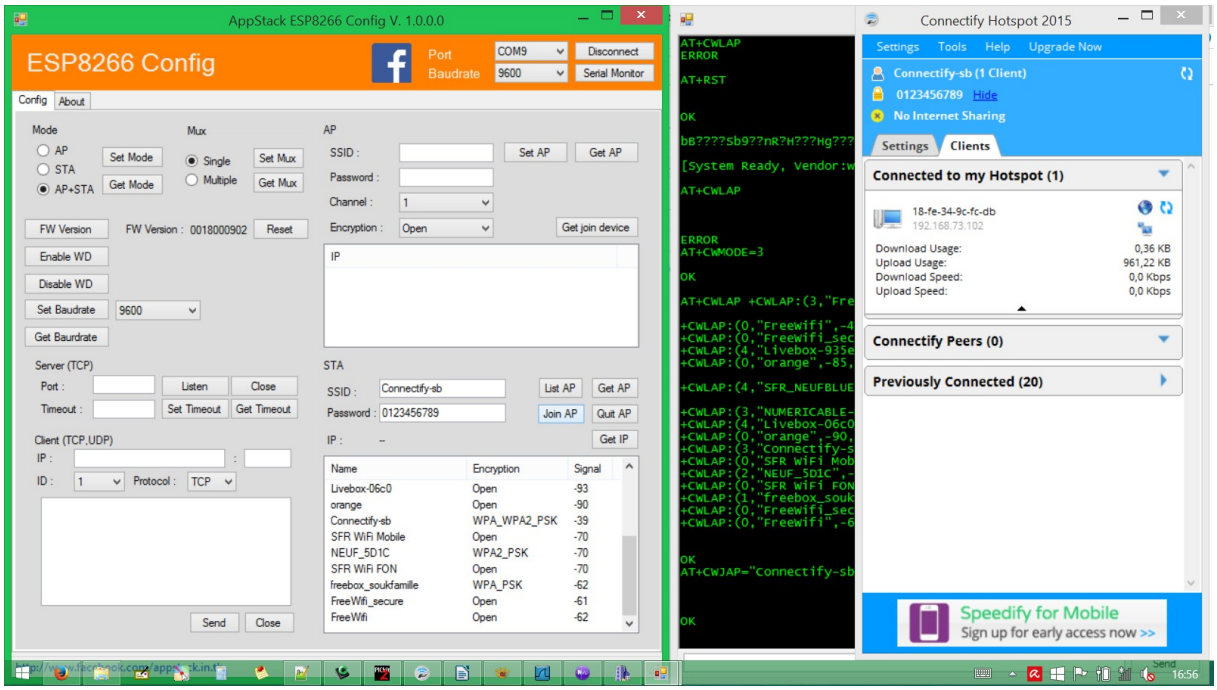
IMPORTANT : vérifier le firmware.
Les programmes proposés fonctionnent avec le firmware 0.924 et 0.922 (fichier .bin fournit sur le site stssnsb.free.fr/iot). Les firmwares plus récents ne fonctionnent pas car les réponses ou les commandes AT sont modifiées : exemple de modification :

- pas nécessité d'envoyer \r\n mais uniquement \r après chaque commande AT.
- Pas de réponse "Linked" à la commande AT+CIPSTART...

Pour vérifier le firmware : cliquer sur FW version

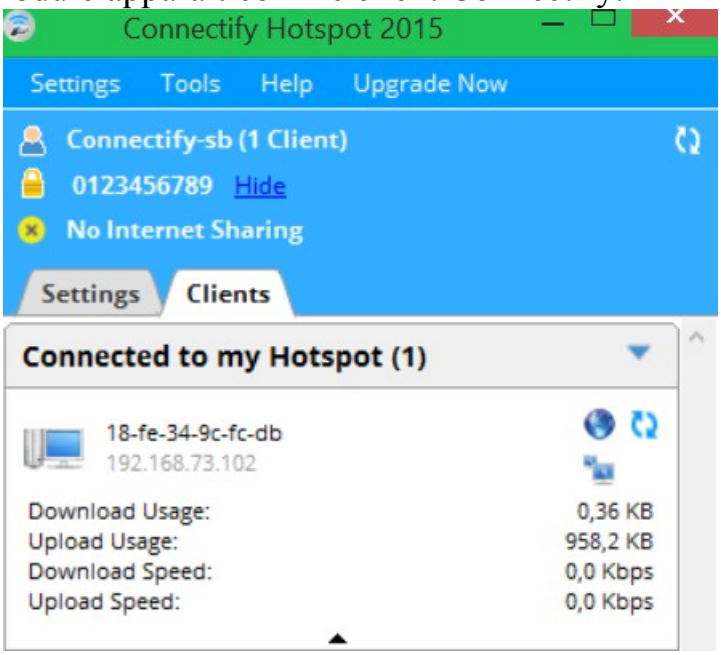
1.3. configuration Wifi

Mode AP+STA, MUX single connexion, SSID : Connectify-sb /pass : 0123456789



Pour voir et choisir le SSID faire :
List AP + Choisir le SSID dans la liste + Join AP

Vérifier que le module apparaît comme client Connectify.



Ici l'IP du module est 192.168.73.102 et le MAC est 18-fe-34-9c-fc-db

2. PROGRAMME EN C POUR PIC

Le programme suivant permet d'envoyer en TCP vers un serveur web à l'adresse 192.168.73.1:80 des données POST correspondant à 2 mesures (mesure1 et mesure2) . Les mesures seront récupérées par un programme php qui remplira une table (table_mesures) d'une base de données (db_mesures).

2.1. Programme en C

```
/*des problèmes : ne fonctionne que de temps en temps
Programme envoyant des données mesurées par le pic vers une base de données web
en utilisant le module ESP8266
*/
/*
Lorsque le message "appuyer pr envoi " apparaît il faut appuyer sur le bouton central du joystick.
Le CIPSTART est alors envoyé et on attend la réponse de ESP8266 : "linked "
Le texte : "Linked : réappuyer" apparaît.
En appuyant de nouveau la requête POST est envoyé par la commande CIPSEND.
Les données envoyant sont  les résultats des convertisseurs CAN0 = mesure1 et CAN1 ) mesure2.
Le programme tourne en boucle.
*/

/*
La procedure a suivre pour créer un client web avec envoi d'une donnée :

Configuration de l'esp8266 :
il faut que l'esp soit en mode STA et connaitre l'adresse IP (IPscan32) ou utilisation du soft
ESPconfig.exe pour le placer dans le bon mode.

activer le lien TCP vers l'adresse du serveurWeb et le port par l'envoi de l'AT :
AT+CIPSTART=\"TCP\",\"192.168.0.10\",80\r\n") le\ permet d'envoyer le caractere "
vérifier la réponse OK et Linked
envoyer le texte suivant qui correspond à la commande AT d'envoi suivi de la requete POST :
"AT+CIPSEND=159\r\n" puis
"
POST /projet_db_mesures/dbvisu.php HTTP/1.1
Host: 192.168.0.10
Content-Type: application/x-www-form-urlencoded
Content-Length: 21

mesure1=11&mesure2=11
"
*/
#include <16F876.h>

#device adc=8



#FUSES NOWDT                 //No Watch Dog Timer
#FUSES HS                   //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES NOPUT                //No Power Up Timer
#FUSES NOPROTECT            //Code not protected from reading
#FUSES NOBROWNOUT           //No brownout reset
#FUSES NOLVP                //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
```

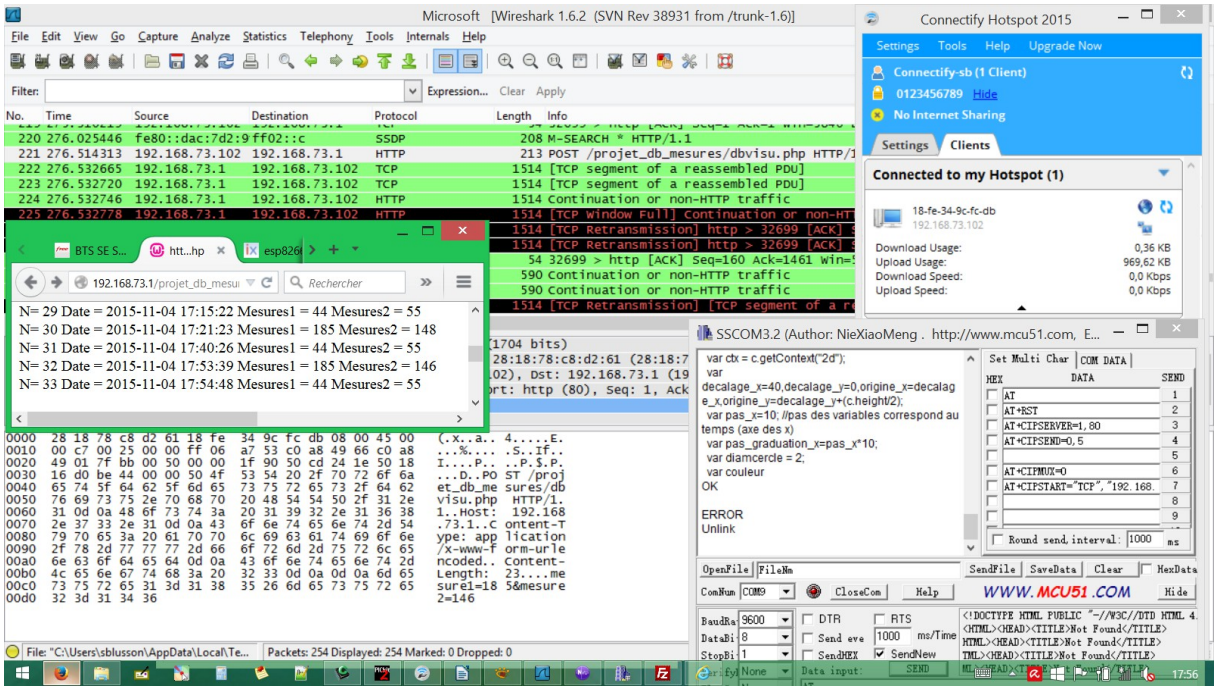
```
#FUSES NOCPD                    //No EE protection
#FUSES NOWRT                    //Program memory not write protected
#FUSES NODEBUG                  //No Debug mode for ICD

#use delay(clock=20000000)
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,ERRORS)
#use rs232(baud=9600,parity=N,xmit=PIN_C5,rcv=PIN_C0,bits=8,stream=DEBUG,ERRORS)
#use i2c(Master,Fast,sda=PIN_C4,scl=PIN_C3)

#include <PCF2119_Driver_LCDI2C.c>
#include <stdlib.h>

//#define ADRWEBSERVEUR 192.168.0.10  //adresse du serveur web (PC sur lequel WAMP est en marche)
#define ADRWEBSERVEUR 192.168.73.1  //adresse du serveur web (PC sur lequel WAMP est en marche)


#define allume_LEDVERTE output_high(PIN_C2);

#define LEDROUGE PIN_C0
#define LEDJAUNE PIN_C1
#define LEDVERTE PIN_C2
#define BP_MILIEU PIN_B4 //BP du milieu du joystick


#int_RDA
void  RDA_isr(void)
{
}

/*****************************************************************************/
boolean testOK(){

int8 i;
//attente OK
  while(getc()!='O') {};
  while(getc()!='K'){return(true);};

//fin attente OK

}//fin testOK
/*****************************************************************************/
/*************************************/
// initialisation du CAN
void initCAN(){
// initialise le CAN//
   setup_port_a( RA0_RA1_RA3_ANALOG );
   setup_adc( ADC_CLOCK_DIV_32 );
}

/*************************************/
// fonction d'acquisition : la valeur retournée est la valeur convertie du canal0
int8 acquerirCAN(int lcanal){

   int lvaquire;

   set_adc_channel( lcanal );
   delay_us(100);
   lvaquire=read_adc();
   return lvaquire;
```

```
}

/**************************************************************************/
void main()
{
int8 i=0;
int8 channel=0;
char strbuffer[20];
int8 lchar;
int8 mesure1=0,mesure2=0;

   setup_adc_ports(AN0_AN1_AN3);
   setup_adc(ADC_CLOCK_INTERNAL);
   setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
   setup_timer_1(T1_DISABLED);
   setup_timer_2(T2_DISABLED,0,1);

   // TODO: USER CODE!!

   set_tris_a(0b00111111);
   set_tris_b(0b11111111);
   set_tris_c(0b10010000);

   initCAN();//initialise le CAN


   Init_Ecran();//initialise l'écran LCD
   Efface_Ecran();printf(Affiche_caractere,"ESP en client ");
   Lcd_Place_Curseur(2,1);printf(Affiche_caractere,"Connectify");
   delay_ms(1000);
   allume_LEDVERTE;
   //config ESP8266
   printf("AT+CIPMUX=0\r\n");testOK();
   Efface_Ecran();printf(Affiche_caractere,"CIPMUXok");delay_ms(2000);


   //acquisition des valeurs de CAN0 et CAN1
   mesure1=acquerirCAN(0);
   mesure2=acquerirCAN(1);

   Efface_Ecran();printf(Affiche_caractere,"Appuyer pr envoi");delay_ms(500);
   while(input(BP_MILIEU)){}//attente appui sur BPMILIEU

//etablir une connexion TCP : avec choix du IP destination et du port
    Efface_Ecran();printf(Affiche_caractere,"Envoi en cours...");delay_ms(500);

//     printf("AT+CIPSTART=\"TCP\",\"192.168.0.10\",80\r\n");
     printf("AT+CIPSTART=\"TCP\",\"192.168.73.1\",80\r\n");
     testOK();  //Efface_Ecran(); printf(Affiche_caractere,"IPSTARTok");
      //test Linked
      while(getc()!='L'){};
      while(getc()!='i'){};
      while(getc()!='n'){};
      while(getc()!='k'){};
      while(getc()!='e'){};
      while(getc()!='d'){};
      Efface_Ecran(); printf(Affiche_caractere,"Linked:...");

     //while(input(BP_MILIEU)) {}; //attente
```

```
//envoyer les donnees en indiquant la taille
delay_ms(1000);
printf("AT+CIPSEND=159\r\n");
printf("POST /projet_db_mesures/dbvisu.php HTTP/1.1\r\n");//47octets-2\=45
printf("Host: 192.168.73.1\r\n");//22-2\=20
printf("Content-Type: application/x-www-form-urlencoded\r\n");//51-2\ = 49
printf("Content-Length: 23\r\n\r\n");//26-4\ = 22 //21=taille de la  ligne suivante
printf("mesure1=%3u&mesure2=%3u",mesure1,mesure2);//23 donc total = 159


delay_ms(10000);//
printf("AT+CWQAP\r\n");testOK();//désactive la connexion
delay_ms(1000);
printf("AT+CIPCLOSE\r\n");//fermeture connection pas de test car retour varié !



Efface_Ecran();printf(Affiche_caractere,"Fin...");delay_ms(500);



}//main
```

2.2. Résultats

Les données apparaissent bien sur la page web du serveur de base de données.



2.3. Remarques :

Attention il faut bien vérifier que le module soit connecté à Connectify.
Allumer le module par l'intermédiaire du PICKIT2 avec MCLR actif

Lancer le programme en désactivant le MCLR du PICKIT2

2.4. Exemple de fonction d'envoi de requête POST :

```
/**************************************/
void EnvoiRequetePOST(int8 mesure1,mesure2){
//Envoi la requete POST au module par commande AT : la requete contient les données mesure1 et mesure2 (ici
mesure1 = NUM_VOTEUR mesure2 = VAL_VOTE
//ok testé

     printf("AT+CIPSTART=\"TCP\",\"192.168.73.1\",80\r\n");
     testOK();
     //test LINKED
     while(getc()!='L'){};
     while(getc()!='i'){};
     while(getc()!='n'){};
     while(getc()!='k'){};
     while(getc()!='e'){};
     while(getc()!='d'){};

    delay_ms(5000);

    printf("AT+CIPSEND=159\r\n");
    printf("POST /projet_db_mesures/dbvisu.php HTTP/1.1\r\n");//47octets-2\=45
//    printf("Host: 192.168.0.10\r\n");//22-2\=20
    printf("Host: 192.168.73.1\r\n");//22-2\=20
    printf("Content-Type: application/x-www-form-urlencoded\r\n");//51-2\ = 49
    printf("Content-Length: 23\r\n\r\n");//26-4\ = 22 //21=taille de la  ligne suivante
    printf("mesure1=%3u&mesure2=%3u",mesure1,mesure2);//23 donc total = 159

    printf("AT+CIPCLOSE\r\n");//fermeture connection pas de test car retour varié !
    delay_ms(1000);//
    printf("AT+CWQAP\r\n");testOK();//désactive la connexion


}//fin EnvoiRequetePOST()
```

3. CRÉER LA BASE DE DONNÉES

Cf TP créer un objet connecté

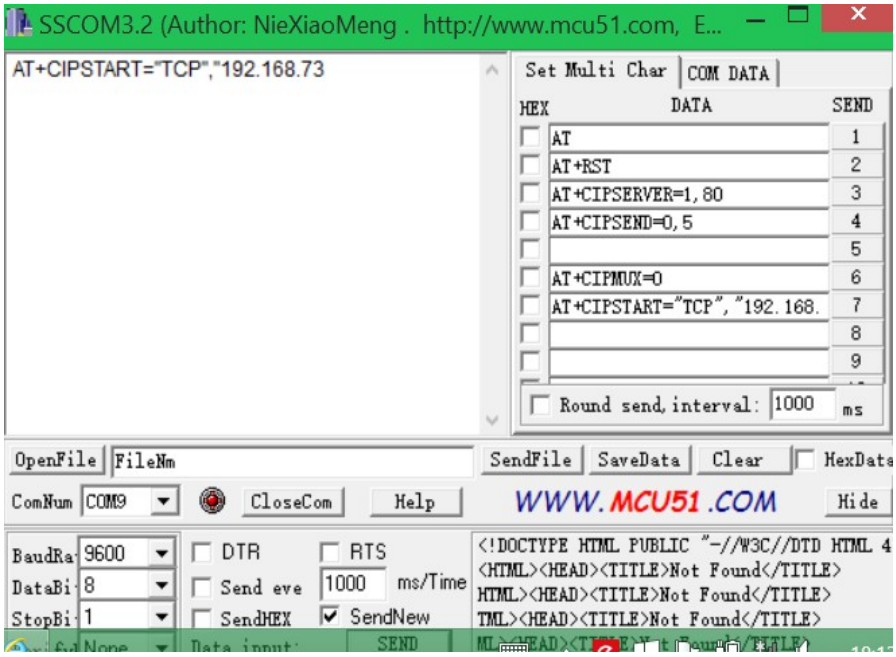
4. PROGRAMMES PHP

Cf TP créer un objet connecté

5. LES PROBLÈMES

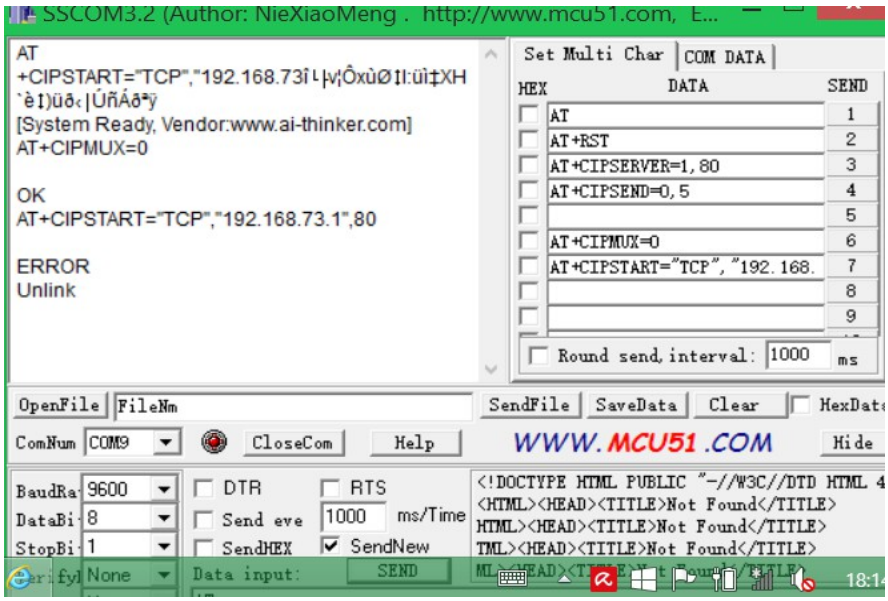
5.1. Premier essai

Blocages après CIPSTART :



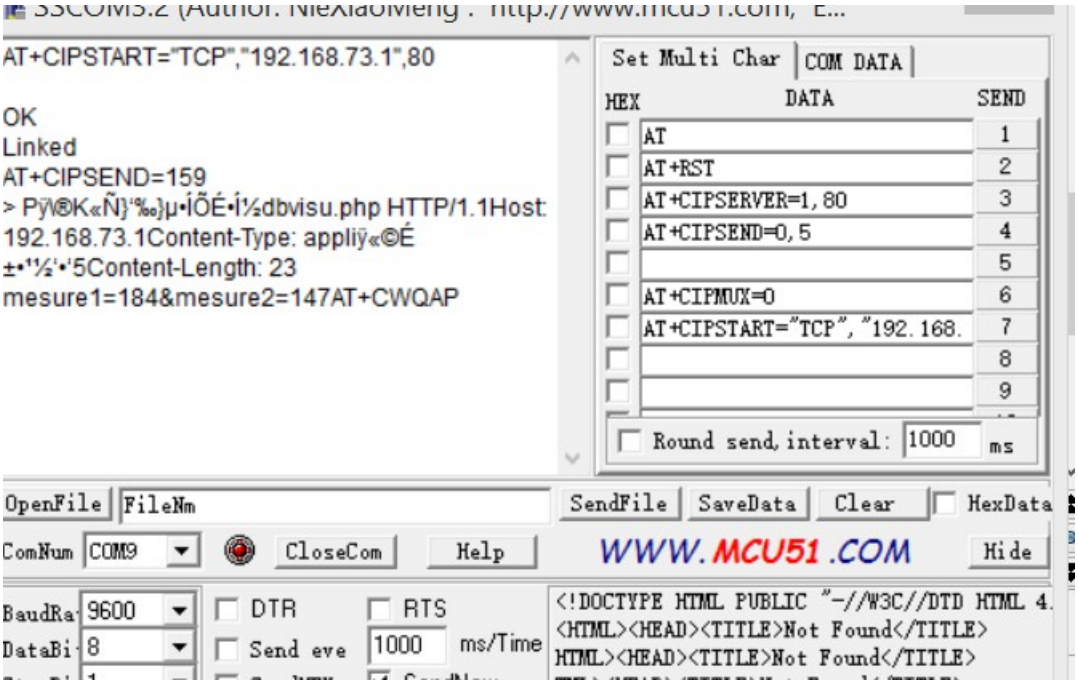
Blocage sur "Envoi en cours..."

Après une extinction de la carte

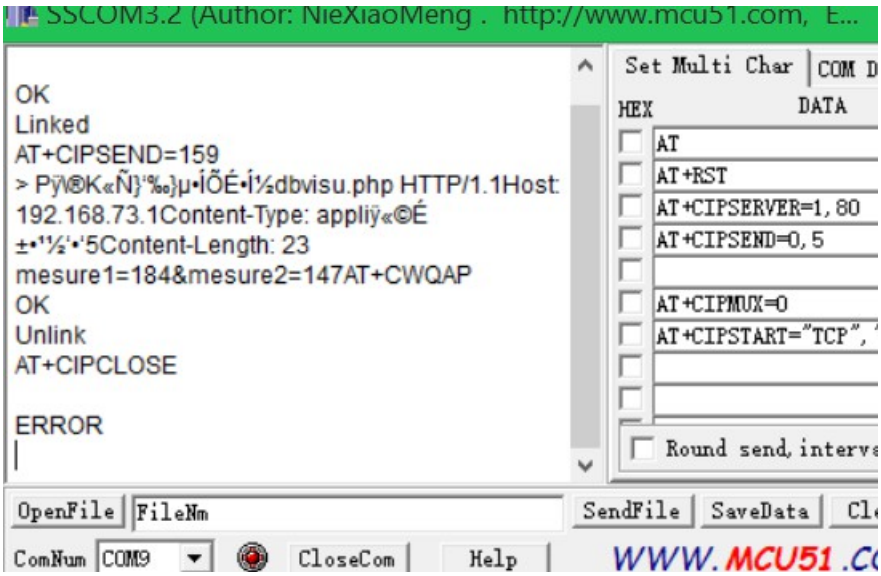


Toujours un blocage sur "Envoi en cours"

Redémarrage de Connectify qui s'était arrêté
Vérification de la connexion du module à Connectify
Reset de la carte PICV2 par le bouton ROUGE
18h19 appui pour "Appuyer pr envoi"
Envoi ok mais incomplet donc pas de données écrites dans la base !



L'écran finit par afficher "Fin..." car un ok a été renvoyé par le module



5.2. Deuxième essai

Appui RST bouton rouge carte PICV2
Affichage "Appuyer pr envoi"

Resultat : NE FONCTIONNE PAS

AT+CIPSTART="TCP","192.168.73.1",80

OK
Linked
AT+CIP159

ERROR
POST /projet_db_mesures/dbvisu.php HTTP/1.1
Error
Host: 192.168.73.1
Error
Content-Type: application/x-www-form-urlencoded
Content-Length: 23
busy p...

busy p...

ERROR
AT+CWQAP

OK

ERROR
Unlink
AT+CIPCLOSE

ERROR

affichage LCD : "Fin..."

5.3. Troisième essai

Attention comme on coupe la connection par programme il faut réalimenter pour que connectify reconnecte le module !!!!

CA MARCHE !
Et effectivement le module se déconnecte de connectify !!

Quatrième essai
Après redémarrag et reset :

AT+CIPStART="TCP","192.168.73.1",80

ERROR

Blocage sur "Envoi en cours"

5.4. Modification

Ne plus fermer la connexion
Faire un RST

Programme testé

FONCTIONNE MIEUX

```
/*****************************************************************************/
void main()
{
int8 i=0;
int8 channel=0;
char strbuffer[20];
int8 lchar;
int8 mesure1=0,mesure2=0;

  setup_adc_ports(AN0_AN1_AN3);
  setup_adc(ADC_CLOCK_INTERNAL);
  setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DISABLED,0,1);

  // TODO: USER CODE!!

  set_tris_a(0b00111111);
  set_tris_b(0b11111111);
  set_tris_c(0b10010000);

  initCAN();//initialise le CAN


  Init_Ecran();//initialise l'écran LCD
  Efface_Ecran();printf(Affiche_caractere,"ESP en client ");
  Lcd_Place_Curseur(2,1);printf(Affiche_caractere,"Connectify");
  delay_ms(1000);
  allume_LEDVERTE;
  //config ESP8266
  printf("AT+CIPMUX=0\r\n");testOK();
  Efface_Ecran();printf(Affiche_caractere,"CIPMUXok");delay_ms(2000);

while(1){

  //acquisition des valeurs de CAN0 et CAN1
  mesure1=acquerirCAN(0);
  mesure2=acquerirCAN(1);

  Efface_Ecran();printf(Affiche_caractere,"Appuyer pr envoi");delay_ms(500);
```

```
   while(input(BP_MILIEU)){}//attente appui sur BPMILIEU

//etablir une connexion TCP : avec choix du IP destination et du port
    Efface_Ecran();printf(Affiche_caractere,"Envoi en cours...");delay_ms(500);

//     printf("AT+CIPSTART=\"TCP\",\"192.168.0.10\",80\r\n");
     printf("AT+CIPSTART=\"TCP\",\"192.168.73.1\",80\r\n");
     testOK();  //Efface_Ecran(); printf(Affiche_caractere,"IPSTARTok");
      //test Linked
      while(getc()!='L'){};
      while(getc()!='i'){};
      while(getc()!='n'){};
      while(getc()!='k'){};
      while(getc()!='e'){};
      while(getc()!='d'){};
      Efface_Ecran(); printf(Affiche_caractere,"Linked:...");

     //while(input(BP_MILIEU)) {}; //attente
     //envoyer les donnees en indiquant la taille
     delay_ms(1000);
     printf("AT+CIPSEND=159\r\n");
     printf("POST /projet_db_mesures/dbvisu.php HTTP/1.1\r\n");//47octets-2\=45
     printf("Host: 192.168.73.1\r\n");//22-2\=20
     printf("Content-Type: application/x-www-form-urlencoded\r\n");//51-2\ = 49
     printf("Content-Length: 23\r\n\r\n");//26-4\ = 22 //21=taille de la  ligne suivante
     printf("mesure1=%3u&mesure2=%3u",mesure1,mesure2);//23 donc total = 159


     delay_ms(1000);//
     printf("AT+CWQAP\r\n");testOK();//désactive la connexion
//     delay_ms(1000);
//     printf("AT+CIPCLOSE\r\n");//fermeture connection pas de test car retour varié !
    printf("AT+RST\r\n");




   Efface_Ecran();printf(Affiche_caractere,"Fin...");delay_ms(500);


}//while

}//main
```