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

AppStack ESP	3266 Config V. 1.0.0.0	_ 🗆 🗙	12	Connectify Hotspot 2015	- 🗆 🗡
ESD9266 Config	Port COM9	✓ Disconnect	AT+CWLAP ERROR	Settings Tools Help Upgrade Now	
ESP6200 Coning	Baudrate 9600	 Serial Monitor 	AT+RST	💄 Connectify-sb (1 Client)	Q
Config About				🔒 0123456789 <u>Hide</u>	
Made H.	40		ок	8 No Internet Sharing	
	SSID · Set AP	Get AP	bB????Sb9??nR?H???Hg???	Settings Clients	
O STA	Paceword -	Gurv	[System Ready, Vendor:w	Connected to my Hotspot (1)	-
AP+STA Get Mode O Multiple Get Mux	Chapped : 1		AT+CWLAP		
		A 111		18-fe-34-9c-fc-db	() () ()
FW Version FW Version : 0018000902 Reset	Encryption : Open V	Get join device	ERROR	Download Lisage:	0.36 KB
Enable WD	IP		AT+CWMODE=3	Upload Usage:	961,22 KB
Disable WD			ок	Download Speed: Upload Speed:	0,0 Kbps 0.0 Kbps
Set Baudrate 9600 V			AT+CWLAP +CWLAP:(3,"Fre	_	
Get Baurdrate			+CWLAP:(0,"FreeWifi",-4 +CWLAP:(0,"FreeWifi_sec	Connectify Peers (0)	•
Server (TCP)	STA		+CWLAP:(0,"orange",-85,		
Port : Listen Close	SSID : Connectify-sb Lis	t AP Get AP	+CWLAP:(4,"SFR_NEUFBLUE	Previously Connected (20)	•
Timeout : Set Timeout Get Timeout	Password : 0123456789 Joi	n AP Quit AP	+CWLAP: (3, "NUMERICABLE-		
Client (TCP,UDP)	IP : -	Get IP	+CWLAP:(0,"orange",-90,		
IP::	Name Encryption	Signal ^	+CWLAP:(0,"SFR WiFi Mob		
ID : 1 V Protocol : TCP V	Livebox-06c0 Open	-93	+CWLAP:(0,"SFR WiFi FON		
	orange Open	-90	+CWLAP:(0,"FreeWifi_sec		
	SFR WiFi Mobile Open	-39	+CWLAP:(0,"FreeWiti",-6		
	NEUF_5D1C WPA2_PSK	-70	OK		
	SFR WiFi FON Open	-70	AT+CWJAP="Connectify-sb		
	FreeWifi secure Open	-62			~
Send Close	FreeWifi Open	-62	ок	Speedify for Mobil	e
				Sign up for early access	now >>
http://www.facetagok.com/apptstk.in.t	🖇 🕎 👂 📑 😻 🖊				Send 16:56

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 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(0b1111111); 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()!='l'){}; 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()!='r){};
while(getc()!='n'){};
while(getc()!='k'){};
while(getc()!='k'){};
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 :

SSCOM3.2 (Author: NieXiaoMeng . http:/	//www.mcu51.com, E 🗕 🗖	×
AT+CIPSTART="TCP","192.168.73	Set Multi Char COM DATA	
	HEX DATA S	END
	AT	1
	AT+RST	2
	AT +CIPSERVER=1, 80	3
	AT+CIPSEND=0, 5	4
		5
	AT+CIPMUX=0	6
	AT +CIPSTART="TCP", "192.168.	7
		8
		9
	V Round send, interval: 1000	ms
OpenFile FileNm	SendFile SaveData Clear / He	exDat
ComNum COM9 💌 🛞 CloseCom Help	WWW. MCU51 .COM	Hide
BaudRa 9600 - DTR RTS	DOCTYPE HIML PUBLIC "-//W3C//DTD H</td <td>TML</td>	TML
DataBi 8 - Send eve 1000 ms/Time	<pre>\niml>\nEAD>\TITLE>Not Found\/TITLE> HTML>\HEAD>\TTTLE>Not Found\/TITLE></pre>	
StopBi 1 - SendHEX SendNew	TML> <head><title>Not Found</title></head>	
SEND	MI ACHEAD SCITTER FOR THE BOWERS (PETER)	

Blocage sur "Envoi en cours..."

Après une extinction de la carte

AT +CIPSTART="TCP","192.168.73î↓ v¦Ôxù؇I:ü̇XH	^	Se	t Multi	Char	COM DATA	SEND
`è1)üð∢ ÚñÁð⁼ÿ [System Ready, Vendor:www.ai-thinker.com]			AT	-		1
AT+CIPMUX=0		H	AT +CTPSE	RVFR=	=1.80	- 2
OK		Ē	AT+CIPSH	END=0,	5	4
AT+CIPSTART="TCP","192.168.73.1",80						5
A A		Г	AT+CIPM	лх=о		6
ERROR			AT+CIPST	TART='	TCP", "192.168.	7
Unlink						8
						9
	~	Г	Round s	end, i	nterval: 1000	ms
OpenFile FileNm	Se	ndFi	ile Sav	eData	Clear	HexD
ComNum COM9 💌 🛞 CloseCom Help		W	WW. N	CUS	51 .COM	Hid
BaudRa 9600 - DTR RTS	D</td <td>OCTY</td> <td>PE HTML</td> <td>PUBLI</td> <td>C ~-//W3C//DTD</td> <td>HTML</td>	OCTY	PE HTML	PUBLI	C ~-//W3C//DTD	HTML
DataBi 8 🔻 🗆 Send eve 1000 ms/Time	<hii HTM</hii 		HEAU> <ii RAD>/TTT</ii 	ILE >N	t Found / TITL	E>
StopBi 1 - SendHEX SendNew	TML		AD> <titl< td=""><td>E>Not</td><td>Found (/TITLE)</td><td></td></titl<>	E>Not	Found (/TITLE)	
CTIM		CHER &	DVTTTT	ANT THE O	E	

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 !

В ЭЭСОІVІЭ.2 (Айспол. Міеліаоіvieng . піцр.)	//www.mcubi.com, e	_
AT+CIPSTART="TCP","192.168.73.1",80 OK Linked AT+CIPSEND=159 > Pÿ\®K«Ñ}'‰}µ•ÍÕɕͽdbvisu.php HTTP/1.1Host: 192.168.73.1Content-Type: appliÿ«©É ±•¹½'•'5Content-Length: 23 mesure1=184&mesure2=147AT+CWQAP	Set Multi Char COM DATA HEX DATA AT AT AT+RST AT+CIPSERVER=1, 80 AT+CIPSEND=0, 5 AT+CIPSTART="TCP", "192.168. AT+CIPSTART="TCP", "192.168. Bound send, interval: 1000	SEND 1 2 3 4 5 6 7 8 9 9 ms
OpenFile FileNm	SendFile SaveData Clear	{exDat
ComNum COM9 💌 🛞 CloseCom _ Help	WWW. MCU51 .COM	Hi de
BaudRa 9600 - DTR RTS DataBi 8 - Send eve 1000 ms/Time	<pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD <HTML> <head><title>Not Found</title><head><title>Not Found</title></head></head></pre>	HTIMIL - ;>

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

OK Linked AT+CIPSEND=159 > PÿV®K«Ñ³'‰}µ•ĺÕÉ•ĺ½dbvisu.php HTTP/1.1Host: 192.168.73.1Content-Type: appliÿ«©É ±•¹½'•'5Content-Length: 23 mesure1=184&mesure2=147AT+CWQAP OK Unlink AT+CIPCLOSE	^	Set Multi Char COM D HEX DATA AT AT+RST AT+CIPSERVER=1,80 AT+CIPSEND=0,5 AT+CIPSTART="TCP",'
ERROR OpenFile FileNm	¥ Se	Round send, interve endFile SaveData Cle

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

ОК

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(0b1111111); 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()!='l'){}; while(getc()!='n'){}; while(getc()!='e'){}; while(getc()!='e'){}; 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