

Création d'une application mobile Sur Plateforme **WIRELESS - HYDRA**



**TP APPLICATION GSM
SUR WIRELESS - HYDRA**

Introduction

Pilotée par un microcontrôleur PIC 32 programmable en C et en C++, la plateforme **Wireless – Hydra** développée par **ABSA-NT** intègre les composants et modules de communication nécessaires permettant à l'utilisateur de se familiariser avec les principaux protocoles de communication.

Le but de ce TP est de permettre aux étudiants de créer des fonctions visant à mettre en oeuvre une application mobile embarquée. Cette dernière peut émettre et recevoir des appels, gérer un répertoire ... l'étudiant pourra l'améliorer pour émettre des sms etc...

NB : Il faut impérativement lire et se munir du manuel MPLAB à chaque TP. Toute la partie création de projet et ouverture d'un projet existant se trouve dans le manuel

Retrouver tous les informations nécessaires au TP sur le site

<http://www.absa-nt.com> rubrique TP/MPLAB-Manuel-Wireless

Sommaire

Introduction	2
TP GSM :	4
Q1: Ecrire un numéro, appeler et recevoir un appel	5
Q2: Création d'un clavier sur l'afficheur tactile	5
Q3) Répondre ou Refuser un appel téléphonique	8

TP GSM :

Ce TP permet de réaliser un GSM pour émettre et recevoir des *appels téléphoniques* avec notre plateforme **Wireless-Hydra**:

On va réaliser cette partie avec des fonctions simples, chaque exercice comportera un nombre de fonctions à faire.

NB : On représentera chaque sous-programme par un bouton qui devra être affiché sur l'écran. La fonction `addBouton` contient tous les paramètres nécessaires sur le choix de la localisation d'un bouton, sa taille et sa couleur de fond, couleur de contour... Une disposition types des touches sera donnée à la fin.

Le fichier `gsm.h` qui sera donné avec le projet **WirelessDevBoard** fourni avec la plateforme Wireless-Hydra contiendra toutes les bibliothèques, les structures et fonctions qui seront explicitées sur `gsm.c` (ou seront détaillées les fonctions de ce TP)

```
#ifndef MOBILEAPP_H
#define MOBILEAPP_H
#include "../libs/lcd_lib/ssd1963.h" //bibliothèques permettant d'afficher à l'ecran LCD
#include "../libs/menu_lib/application.h"//bibliothèques qui s'occupe du menu principale de la carte
#include "../libs/gsm_lib/sim900.h"//bibliothèques necessaires pour emettre et recevoir des appels
#include "../libs/task_lib/task.h"//bibliothèques pour la gestion des taches
#define TASK_ID_MOBILE "task_mobile_id"// tache qui gere toute l'application

void MOBILEAppInit(struct Task * task);//fonction d'initialisation
void MOBILEAppTask(struct Task * task);//fonction qui gere les taches
void MOBILEAppDestroy(struct Task * task);//fonction pour arreter une ou plusieurs taches

void MOBILEApp(struct Button *butt); //Fonction qui affiche les boutons
void MOBILEExit(struct Button *butt);//permettant de sortir de l'application mobile
//on pourra retourner au menu s'il est déjà fait
void MobileAppNmbr(struct Button *butt);//affiche les
void MobileAppCall(struct Button *butt);
void MobileAppCancel(struct Button *butt);
void MobileAppBackSpace(struct Button *butt);
void MobileOnOff(struct Button *butt);
void MobileAboutNetwork(struct Button *butt);//gere la connection
void MobileReset(struct Button *butt);// remise à zero
void MobilePin(struct Button *butt);// metre un code pin
void MobileDecodeAT(unsigned char *AT_response, int ind);// gere les appels en générale
void MobileIncommingCallApp(void);// gere les appels reçus
#endif /* MOBILEAPP_H */
```

Q1: Ecrire un numéro, appeler et recevoir un appel

Remarques : Toutes les fonctions utiles permettant de mener à bien le TP sont expliquées dans l'exemple donné à la fin du manuel MPLAB destiné à la carte Wireless-Hydra

Créer une touche qui permettra d'effacer tout ce qui est écrit sur l'écran graphique et de revenir au menu principal (inspirez-vous de la fonction `exit` expliquée dans le manuel MPLAB),

Une autre touche permettra de passer un appel.. Les sous-programmes donnés comme aide se trouvent dans l'arborescence `Wirelessdevboard/mobile/mobileapp.c` et les bibliothèques utilisées se trouvent dans l'arborescence `Wirelessdevboard/lib/gsm.lib`

Vous pouvez partir du programme suivant

A l'aide des bibliothèques données, compléter la fonction suivante.

```
void MobileAppCall(struct Button *butt) {
    StdOutSet(gsm_uart_out);
    printf("ATD%s;\n", phone_number);
    /*à compléter
    */
    ssd1963_PutText(240, 100, phone_number, Green, Black);
    ShowButtons();
}
```

Q2: Création d'un clavier sur l'afficheur tactile

Faire une fonction qui permet de créer les touches du clavier pour composer les numéros et les afficher dans la ligne en dessus. Pour ce faire vous allez utiliser :

- `MobileAppNbr` : Fonction permettant d'écrire des numéros

```
. //fonction à utiliser pour appeler|
] void MobileAppNbr(struct Button *butt) {
    phone_number[phone_number_index] = butt->ButtonName[0]; //nom du bouton
    phone_number[++phone_number_index] = 0; //nombre de bouton
    ssd1963_PutText(240, 5, phone_number, Green, Black); //affichage
. }
```

- **MobileAppCall** : Interface de l'application mobile

```
//Function handles Call Button

] void MobileAppCall(struct Button *butt) {
    StdOutSet(gsm_uart_out);
    printf("ATD%s;\n", phone_number);

    DeleteButtons();
    AddButton(20, 80, 80, 100, "Exit", Red, Exit, NULL);
    AddButton(240, 135, 305, 200, "Cancel", Green, MobileCallingAppCancel, NULL);
    ssd1963_PutText(240, 80, "Calling number", Red, Green);
    phone_number[++phone_number_index] = 0;
    ssd1963_PutText(240, 100, phone_number, Green, Black);
    ShowButtons();
- }
```

- **MobileAppCancel** : fonction permettant de supprimer ce qu'il y'a à l'écran LCD

```
//Function handles Cancel Button

] void MobileAppCancel(struct Button *butt) {
    phone_number_index = 0;
    ssd1963_ClearZone(240, 5, 120, 20, Black);
- }
```

- **SetPromtParameters** : Cette fonction est déjà expliquée et détaillée dans le manuel MPLAB de la plateforme Wireless -Hydra
- **ssd1963_PutText** : Cette fonction est déjà expliquée et détaillée dans le manuel MPLAB de la carte Wireless -Hydra
- **AddButton** Cette fonction est déjà expliquée et détaillée dans le manuel MPLAB de la carte Wireless -Hydra

- compléter le programme suivant

//GSM Application

```
Void MOBILEApp (struct Button *butt) {  
  
    GSMDebugEnable (TRUE);  
  
    DeleteButtons ( );  
  
    SetTimePosition (10, 20);  
  
    /* à compléter */  
  
    If (Si on écrit un numéro)  
    {  
  
        else  
  
    /*  
  
    */  
  
    }  
}
```



Q3) Répondre ou Refuser un appel téléphonique

Compléter le programme suivant en partant de l'aide fourni.

```
void MobileIncommingCallApp (void) {
    /* à compléter */
    ShowButtons();
}
```

Il faut utiliser les fonctions suivantes :

- **MobileIncommingCallAppAnswer** : Sert à répondre aux appels reçus et à arrêter la sonnerie

```
//Function handles Answer button then happens incomming call
] void MobileIncommingCallAppAnswer(struct Button *butt) {
    StdOutSet(gsm_uart_out);
    printf("ATA\n");
    BUZZER_OFF;
- }
```

- **MobileIncommingCallAppRefuse** : Sert à refuser les appels reçus et à arrêter la sonnerie

```
//Function handles reffuse button then happens incomming call
[] void MobileIncommingCallAppRefuse(struct Button *butt) {
    StdOutSet(gsm_uart_out);
    printf("ATH0\n");
    MOBILEApp(butt);
    BUZZER_OFF;
}
```


Le menu principal du GSM sur l'écran LCD se présenter comme le montre l'image suivante

