



Faculté des Sciences de Tunis



Université Tunis El Manar

Projet Driver GRPS en BM

MATIÈRE : Technologies Interfaces Systèmes
embarqués

ÉLABORÉ PAR : BEL HAJ SAID Mahdi
Fares Dkhili

ANNÉE UNIVERSITAIRE: 2024/2025

SECTION: ICE4

• Contexte du projet

Les modules GPRS, tels que le SIM808, sont largement utilisés dans les systèmes embarqués pour permettre la communication sans fil dans des applications IoT (Internet des objets), telles que la surveillance à distance, le suivi de véhicules, ou la transmission de données en temps réel. Le développement de drivers adaptés à ces modules est crucial pour assurer une communication stable et fiable.

• Objectifs du projet

Le but de ce projet était de développer un driver GPRS pour le microcontrôleur STM32F429ZIT6, permettant de faciliter l'utilisation du module SIM808, et plus particulièrement de sa connexion GPRS. L'objectif principal est de fournir à l'utilisateur un ensemble de fonctions prédéfinies, qu'il pourra intégrer facilement dans son projet, pour réaliser les fonctionnalités de base du GPRS.



- **Technologies et outils utilisés**

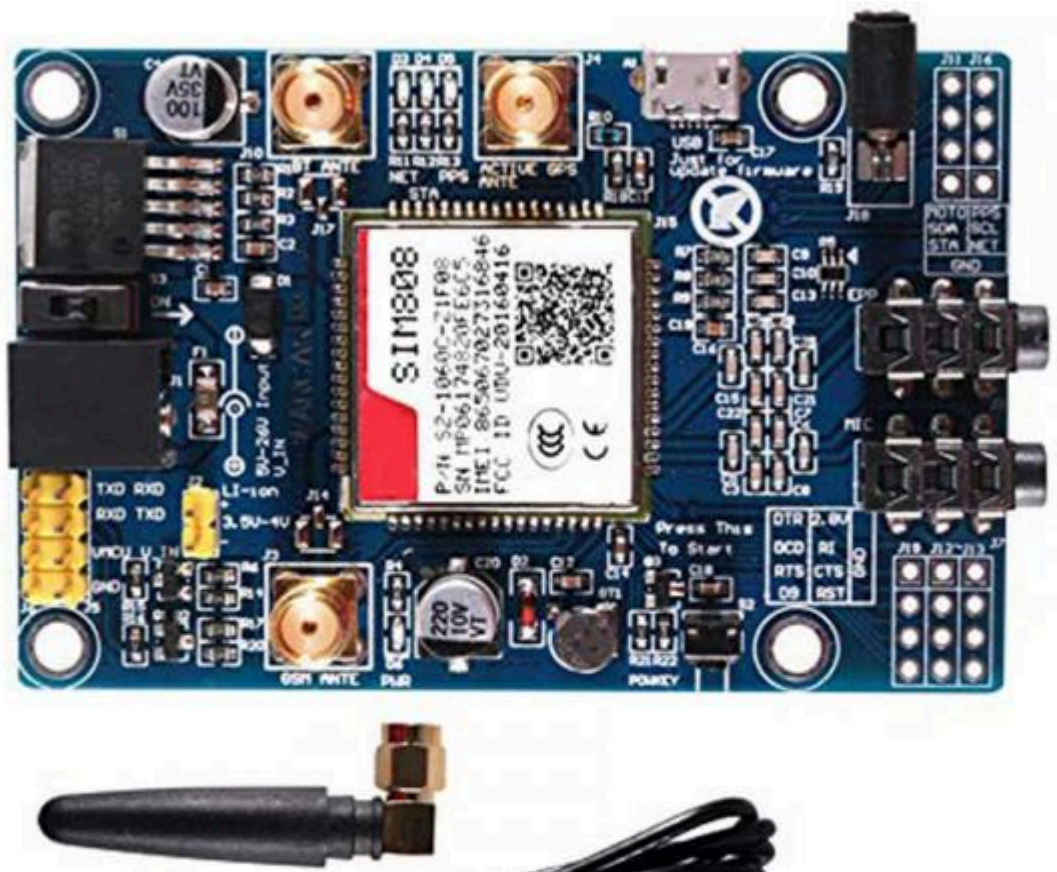
1) Microcontrôleur STM32F429ZIT6

Le STM32F429ZIT6 est un microcontrôleur de la famille STM32F4 développé par STMicroelectronics. Il est basé sur le cœur ARM Cortex-M4 à 180 Mhz et est conçu pour offrir des performances élevées tout en étant capable de gérer des applications embarquées complexes et nécessitant des calculs intensifs.



2) Module SIM808

Le SIM808 est un module polyvalent qui offre des solutions de communication complètes pour des applications embarquées. Il offre les modules GSM , GPS GPRS. Il permet des communications vocales, l'envoi de SMS, et l'accès à Internet via GPRS... tout en étant relativement facile à intégrer grâce à ses interfaces standard comme UART.



3) Keil uVision5

Un environnement de développement intégré (IDE) largement utilisé pour la programmation et le débogage de microcontrôleurs, principalement dans les applications embarquées

4) UART

Un module matériel utilisé dans les systèmes embarqués pour la communication série asynchrone. Il permet l'échange de données entre un microcontrôleur et d'autres périphériques.

5) DMA

Le DMA est un mécanisme qui permet de transférer des données entre la mémoire et les périphériques sans intervention directe du processeur.

• Développement du driver GPRS :

Le développement du driver GPRS pour le microcontrôleur STM32F429ZIT6 se divise en deux parties principales :

1. **Le fichier GPRS_LIB.c** : Ce fichier contient toutes les fonctions qui seront intégrées dans le driver. Il comprend l'implémentation des fonctions

nécessaires pour configurer et interagir avec le module SIM808 via la communication GPRS.

2. **Le fichier GPRS_LIB.h** : Ce fichier contient le prototypage des fonctions et la déclaration des variables globales et externes utilisées dans le driver. Il permet une interface propre entre les différentes parties du code, facilitant l'utilisation du driver dans d'autres projets.

Fonctions du Driver GPRS :

Les fonctions suivantes ont été choisies pour être incluses dans le driver GPRS :

- void GPRS_Init ()
Initialise les ressources nécessaires pour le module GPRS tel que les paramètres du GPIO , du DMA et de l'UART
- void GPRS_Attach()
Effectue l'attachement au réseau mobile pour établir une connexion GPRS.
La commande : AT+CGATT=1
- void GPRS_SetBearerProfile()
Configure le profil de porteur

La commande : AT+SAPBR=3 + mode + Contype + Username + password

- void GPRS_OpenBearer()

Ouvre une connexion GPRS pour l'échange de données.

La commande : AT+SAPBR=1 + mode

- void GPRS_CloseBearer()

Ferme la connexion GPRS après avoir terminé l'échange de données

La commande : AT+SAPBR=0 + mode

- void GPRS_SetAPN()

permettre au module SIM808 de se connecter à Internet via le fournisseur de services mobiles.

La commande : AT+CSTT

- void GPRS_BringUpWirelessConnection()

Elle initie la communication pour accéder à des services Internet, tels que l'envoi ou la réception de données HTTP.

- int SIM808_HTTP_Post()

Permet d'envoyer des données via le protocole HTTP en utilisant une méthode POST

- int SIM808_HTTP_Get()

Permet de recevoir des données via le protocole HTTP en utilisant une méthode GET.

- void SIM808_Init()

Initialise le module SIM808, configure les paramètres de communication série, et prépare le module à recevoir et envoyer des commandes AT.

• Cablage :



- Alimentation 12V pour le module SIM808
- TX du SIM808 associé avec le RX(PA10) du Carte STM32
- RX du SIM808 associé avec le TX(PA9) du Carte STM32
- Antenne branché sur la partie associé au GPRS

● Tests et Validation :

Pour vérifier le bon fonctionnement des méthodes développées, nous allons créer un fichier `main.c` qui fera appel au driver créer et tester chaque fonction une par une. Pour chaque test, nous vérifierons les valeurs de retour pour s'assurer qu'elles sont correctes.

Afin de faciliter les tests, deux méthodes seront utilisées :

1. **Retour entier des méthodes et inspection des valeurs via le Watch Window** : Nous inspecterons les valeurs de retour des fonctions pour valider si elles sont correctes.
2. **Affichage des messages de succès ou d'erreur via la console de débogage série (Serial Debug Console)** : Les résultats des tests seront affichés sous forme de messages indiquant si la fonction a réussi ou échoué. Cela permettra de suivre le processus de test et d'identifier rapidement les éventuelles erreurs.

Étape Primordiale : Test de Connectivité

Une étape clé consiste à envoyer une commande AT simple, comme `AT`, pour tester la connectivité entre le module SIM808 et la carte STM32. Cette commande permet de vérifier si le module SIM808 répond correctement.

- **Si la réponse obtenue est "OK"**, cela signifie que la communication entre le module SIM808 et la carte STM32 fonctionne correctement.
- **Si la réponse est différente**, cela indique un problème de configuration, qu'il faudra corriger avant de continuer les tests plus avancés.

Ainsi, cette étape initiale permet de valider la base de la communication avant d'aller plus loin dans les tests des autres fonctionnalités du driver GPRS.

Et finalement, nous avons testé une API de démonstration en envoyant un nombre binaire à 3 bits à l'aide de la fonction `SIM808_HTTP_Get()`.

Dashboard FST 2024/2025			
Pin Number	Pin Status	Toggle	
9	On	Désactiver	
10	On	Désactiver	
11	Off	Activer	

● Analyse des performances :

Les clés de performance les plus importantes dans ce projet sont la consommation d'énergie, la latence, la fiabilité et le temps de réponse. Ces critères de performance peuvent être optimisés au niveau logiciel grâce à un développement en **bare metal**.

Le reste des performances dépendra principalement de la puissance du matériel utilisé. Cependant, pour garantir une solution logicielle optimale, nous avons choisi de développer le driver GPRS en **bare metal**, ce qui permet un contrôle plus précis et une gestion plus efficace des ressources, assurant ainsi des performances accrues en termes de consommation d'énergie, latence, fiabilité et temps de réponse.