INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

Projet de Physique P6

STPI/P6/2016-08

CRÉATION D'UNE BASE DE DONNÉES POUR L'ANALYSE DE COMPOSANTS ÉLECTRONIQUES

Etudiants :

Matthieu BONNORON

Jérémy CATELAIN

Simon DELECOURT

María BERMÚDEZ JAIME

Enseignant-responsable du projet :

A.ECHEVERRI

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

Cette page est laissée intentionnellement vierge.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

Date de remise du rapport :**13/06/2016**

Référence du projet : ***STPI/P6/2016-08***

Intitulé du projet : Création d'une base de données pour l'Analyse de composants électroniques.

Type de projet : ***(expérimental, simulation, veille technologique,…):Développement d'applications***

Objectifs du projet *(10 lignes maxi)* :

- **Utilisation de Scilab pour l'analyse de données et le développement d'une interface graphique**
- **Visualisation du suivi d'un appareil électronique sous forme de graphique**
- **Création d'une base de données**
- **Gestion de plusieurs fichiers**
- **Création d'un tableau qui récupère les données du suivi**

Mots-clefs du projet *(4 maxi)* :**Visualisation graphique- Récupération des données-Transistors**

Si existant, n° cahier de laboratoire associé : ———

INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE ROUEN
Département Sciences et Techniques Pour l'Ingénieur
avenue de l'Université - 76801 Saint-Etienne-du-Rouvray - tél : +33 (0)2 32 95 97 00 - fax : +33 (0)2 32 95 98 60

Table des matières

Première partie

Introduction

Dans le cadre de notre projet P6 nous nous sommes intéressés aux transistors, système essentiel dans le développement des appareils électroniques modernes.

Plus particulièrement, nous avons cherché à developper une application permettant le suivi d'un objet électronique. L'intérêt de la première partie du projet est d'étudier un transistor à différentes dates et températures ; et tracer des courbes à partir de fichiers .csv pour pouvoir faire d'éventuelles analyses. En d'autres termes, le but est d'optimiser et de faciliter les opérations nécessaires à l'étude des transistors ainsi que d'analyser les résultats en se servant d'une interface graphique. La deuxième partie consiste à récupérer les données dans un tableau récapitulatif et à créer une base de données contenant les composants déjà étudiés.

Afin d'illustrer le travail realisé, nous allons présenter dans une première partie la méthodologie du projet et comment nous nous avons organisé.Dans une deuxième partie nous allons présenter nos recherches et l'explication du code de l'application que nous avons développé sur Scilab.

Deuxième partie

Méthodologie/ Organisation du travail

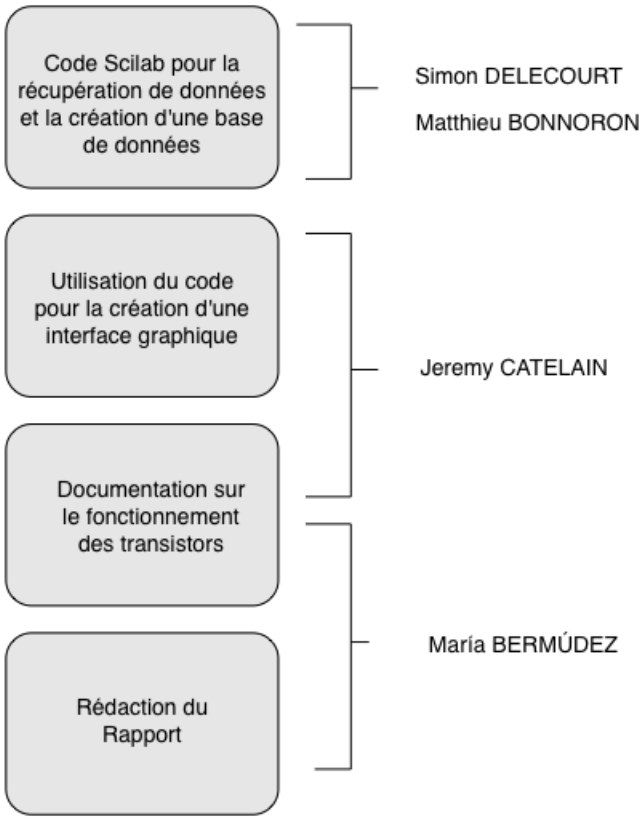
Pour présenter l'évolution du projet au cours de semestre nous pouvons résumer les objectifs, atouts et difficultés rencontrés dans le tableau suivant :

Semaine	Objectif	Difficultés
1er Février	Familiarisation avec le logiciel Scilab, définir les objectifs du projet	-
22 Février	Récupérer les données d'un fichier csv avec scilab, planning du projet	Récuperer seulement les données qui nous intéressent
29 Février	Rechercher des informations sur les transistors	-
7 Mars	Recherche sur le développement d'une interface graphique sur scilab. Récupérer uniquement les données numériques des fichiers	Déterminer le nombre de lignes à lire par Scilab
14 Mars	Commencer la rédaction du rapport	-
21 Mars	Interface graphique : extraire l'information d'un ficher csv pour créer un graphique	Sélection d'un fichier à partir d'un dossier

Semaine	Objectif	Difficultés
18 Avril	Menu sélection abscisse et ordonnée dans l'application Scilab	
25 Avril	Modification et amélioration du rapport.Chercher à ajouter dans la base de données un composant déjà étudié (algorithme scilab)	-
2 Mai	Enregistrement de nouvelles données dans une base de données déjà existante	Debuger le code
9 Mai	Amélioration du code pour l'extraction des données	Intégrer et utiliser le nouveau code pour le développement de la partie graphique
23 Mai	Rédaction du rapport : partie explication du code, continuer le développement de l'interface graphique	Intégrer les algorithmes à l'interface graphique, optimisation des codes
30 Mai	travailler dans l'interface graphique, créer un tableau avec toutes les données ,commenter le code	debuger le code de l'interface graphique

1 Répartition du travail

Afin de travailler plus efficacement on a divisé le travail de la manière suivante :



2 Évolution

Le code sur Scilab et l'interface graphique, ont été réalisés progressivement au cours du semestre afin d'améliorer petit à petit l'application. Plusieurs versions ont ainsi été faites :

Première version de l'application : Dans cette version (*figure 1*) nous

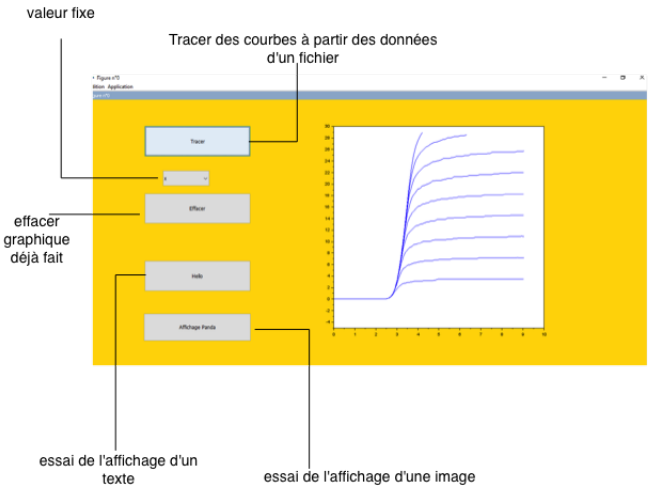


FIGURE 1 – Premier aperçu de l'application scilab

pouvons sélectionner le fichier .csv à étudier mais l'abcisse et l'ordonnée du graphique à tracer sont prédéfinis.

Deuxième version de l'application : Dans cette version (*figure 2*) plusieurs fonctionnalités ont été ajoutées :sélection de l'abscisse , de l'ordonnée et de la température, affichage d'un menu(option pour quitter l'application),etc. Afficher un tableau avec les données se trouve parmi les fonctionnalités à ajouter.

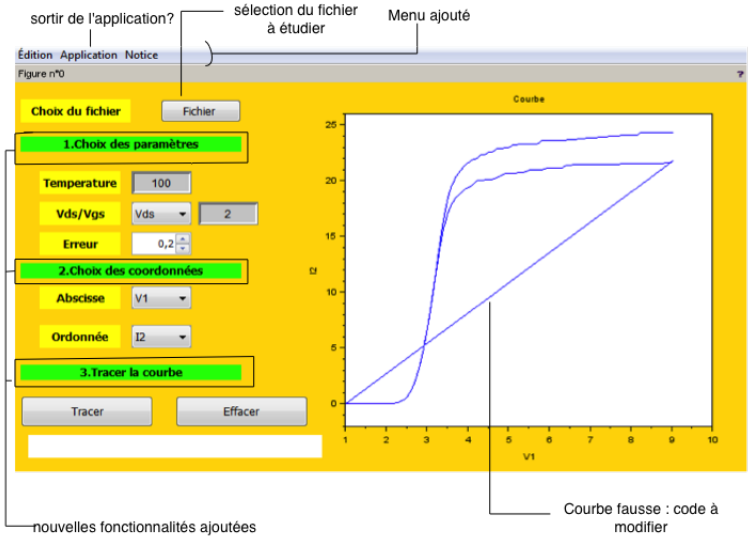


FIGURE 2 – deuxième aperçu de l'application scilab

Troisième partie

Travail réalisé et résultats

Du fait qu'on étudie transistor qui est un composant que nous connaissons peu, il nous a paru important d'approfondir nos connaissances sur son fonctionnement. Ceci nous a permis de mieux comprendre les graphiques et les résultats obtenus.

3 Fonctionnement des transistors

3.1 Définition

Un transistor est un composant électronique constitué de matériaux semi-conducteurs. Il est par exemple utilisé :

— comme interrupteur dans les circuits

— comme amplificateur de signal

— pour stabiliser une tension et moduler un signal

C'est un dispositif à trois électrodes actives, qui permet de contrôler un courant (ou une tension) sur une des électrodes de sortie (le collecteur pour le transistor bipolaire et le drain pour le transistor à effet de champ) grâce à une électrode d'entrée (la base sur un transistor bipolaire et la grille pour un transistor à effet de champ).

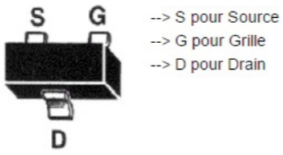


FIGURE 3 – Transistor à effet de champ

Il existe une grande variété de transistors, selon la marque et leurs caractéristiques (taille et puissance, courant continu et maximal etc.).

3.2 Points principaux

Le transistor est un composant qui permet de réguler le courant ou la tension qui circule (rôle d'amplificateur) ou se comporte comme un "switch" pour les signaux électroniques (commute). Ils sont souvent faits en silicium , un matériau semi-conducteur.Le transistor possède aussi un canal semi-conducteur de type N ou P :

— *Type P* : les charges électriques sont portées sous forme de déficiences d'électrons ("trous").

— *Type N* : les électrons portent les charges(courant négatif).

En effet, le dopage de type N consiste à produire un excès d'électrons, alors que le dopage de type P consiste à produire un déficit d'électrons, donc un excès de trous. Ces deux types de dopage permettent de modifier la conductivité des éléments chimiques présents dans les transistors.

Deux sources d'alimentation sont nécessaires pour assurer le bon fonctionnement d'un transistor. Elles sont souvent notées :

— V_{GS} : entre la grille(gate) et la source.

— V_{DS} : entre le "Drain" et la source.

3.3 Transistors FET

Le FET (field-effect transistor) est un type de transistor utilisé pour l'amplification des signaux faibles. Il peut amplifier des signaux analogiques ou numériques.Il s'agit d'un transistor semi-conducteur unipolaire(la circulation du courant dépend uniquement des trous *ou* des électrons mais pas les deux) à trois sorties. Dans le FET on trouve un chemin semi-conducteur (le "canal" type P ou N) .Dans une extrémité du "canal" on trouve l'électrode "source" et de l'autre côté l'électrode appelée le "Drain". La conductivité du FET peut varier grâce à la grille (une petite variation dans le voltage de la grille peut causer un changement important dans le courant). Il existe deux types de transistor FET : les **JFET**(junction FET) et **MOSFET** (metal-oxide- semiconductor FET). Dans le cadre de notre projet on s'intéresse plutôt au MOSFET.

3.3.1 Principe de fonctionnement

Une tension V_{DS} positive se présente entre la grille et la source, si aucune tension V_{GS} n'est appliquée entre la grille et la source. La diode Substrat-Grille (le substrat est relié à la source) est polarisée en inverse et aucun courant ne circule dans le circuit Grille-Source. Lorsque l'on applique une tension V_{GS} positive entre la grille et la source, le champ électrique créé par la tension V_{GS} repousse les porteurs majoritaires présents dans le substrat. Si le substrat est de type P, les porteurs majoritaires sont des trous. Il se forme alors sous la grille une zone de « déplétion », vide de porteurs libres (trous) mais peuplée d'atomes ionisés négativement.

3.3.2 Etude expérimentale

Dans le réseau des caractéristiques de sortie $I_D = f(V_{DS})$, on observe quatre zones différentes. Une zone linéaire dite résistive, un coude, une zone de saturation ($I_D = constant$) et une zone d'avalanche.

zone résistive Il existe une zone isolante qui corespond aux jonctions grille-canal et substrat-canal et où l'épaisseur est fonction de la tension inverse ($e = k\sqrt{V_{GS}}$) .Cette zone diminue la largeur dite "effective" du canal. Si V_{DS}

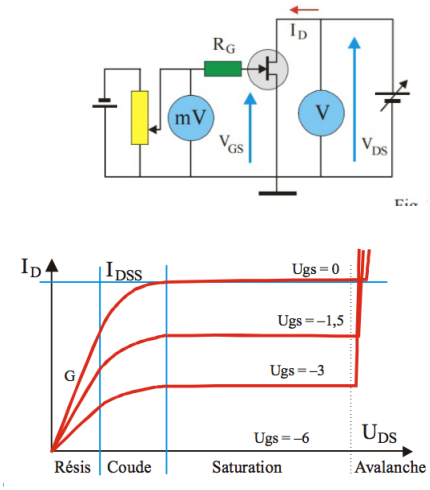


FIGURE 4 – $I_{DS} = f(V_{DS})$

est faible, le cana se comporte comme une résistance ohmique.Pour une valeur V_P suffisamment négative de V_{GS}, la conduction s'annule (V_P est la tension de pincement)

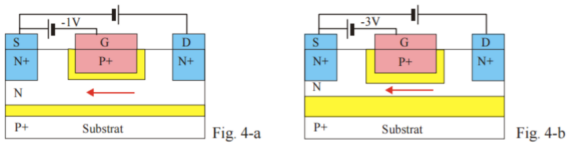


FIGURE 5 – Zone résistive

zone du coude : La tension V_{DS} agit sur la zone isolante.Du côté de la source sa largeur est : $e_1 = k\sqrt{V_{GS}}$. Du côté du drain, elle est :$e_2 = k\sqrt{V_{GD}}$.Quand V_{DS} augmente deux phénomènes entrent en compétition : une croissance (caractère ohmique du canal) et un diminution lié au V_P dans le canal.

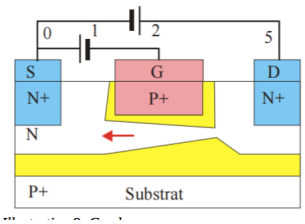


FIGURE 6 – Coude

zone saturation : Dans cette zone si V_{DS} augment I_{DS} et V_P augmentent aussi.La densité du courant augment jusqu'à arriver à la vitesse limite des porteurs.I_{DS} reste constant et le transistor est dit saturé.La valeur maximale de I_D pour $V_{GS} = 0$.

zone avalanche :C'est le résultat du claquage inverse de la jonction drain-grille. Si le courant du drain n'est pas limité le claquage peut détruire du dispositf.

zone température : Le courant du drain augmente à cause de la diminution de la largeaur de la zone de délétion et la température (ainsi que la obilité des porteurs).

3.4 MOSFET

Le MOSFET est un transistor unipolaire à effet de champ à couche d'oxyde de silicium. Il peut fonctionner en commutateur analogique. Ce type de transistor permet de commander un courant "drain" (I_{DS} à l'aide de la tension de la grille V_{GS}).Dans le MOSFET la grille est isolée du canal. Le canal dopé (P ou N) entre Drain et Source dévient plus ou moins conducteur par la jonction PN entre la Grille et la Source (polarisée en inverse). Il utilise un champ électrique pour contrôler la conductivité du canal/

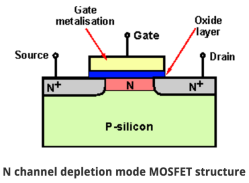


FIGURE 7 – Schéma du MOSFET

3.4.1 type "enrechissement"

Il est bloqué pour V_{GS} nul. Si V_{GS} est positive les trous qui se trovent près

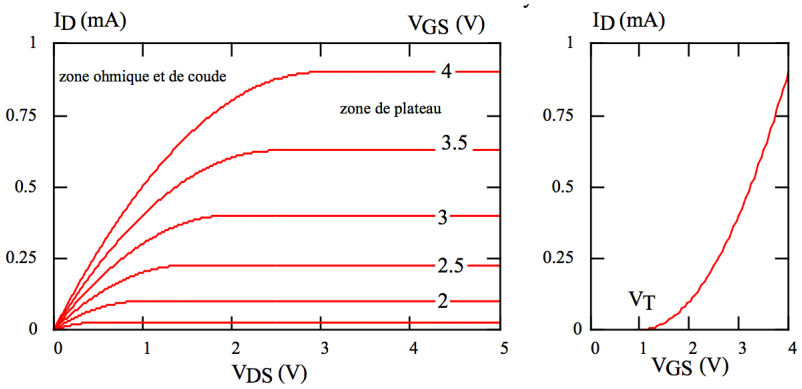


FIGURE 8 – Caractéristiques MOSFET "enrechissement"

du canal sont repousés à cause de l'influence électrostatique.Il existe donc un tension seuil V_T parmi les V_{GS} (*figure8*)dans les trous repoussés sont remplacés par des électrons.Un canal induit de type N apparaît et le courant I_{DS} commence à circuler.Plus $V_{GS} >> V_T$ plus I_{DS} augmente.Mais si on fait augmenter V_{DS},I_{DS} se ralenti.

3.4.2 type "appauvrissement"

La struscture est la même qu'avec le MOSFET à enrechissement sauf pour le canal N qui, dans ce cas, est crée par implantation ionique.

Dans le MOSFET "appauvrissement" on le appauvrit le canal en électrons avec une tension V_{GS} négative.Pour $V_{GS} < V_T$ (*figure9*) le MOSFET est bloqué.

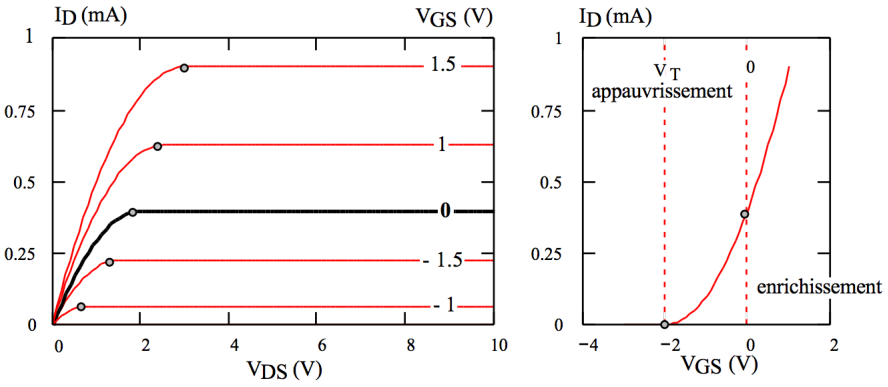


FIGURE 9 – Caractéristiques MOSFET "appauvrissement"

3.4.3 Commutation à charge passive et à charge active

V_{GS} et V_{DS} doivent avoir la même polarité. Seulement les transistors à enrechissement fonctionnent en régime bloqué ou saturé avec ces condition.Les MOSFET utilisés pour la commutation sont donc à enrechissement.

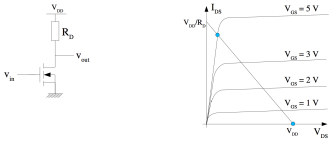


FIGURE 10 – Commutation à charge passive

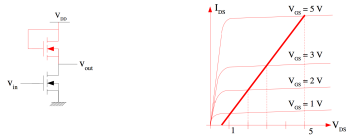


FIGURE 11 – Commutation à charge actives

Commutation à charge passive le circuit à une fonction "d'inverseur"

Commutation à charge active Le transistor (en rouge sur la figure) se comporte comme une résistance

3.4.4 Résumé MOSFET

Une faible tension Drain-Source ($V_{DS} < V$) est appliquée. Le courant I_{DS} passe à travers du canal N.L'itensité de I_{DS} dépend de la densité du canal et donc de la tension V_{GS} (qui a une influence sur la densité d'électrons).

Si les V_{GS} est inferieures ou égales à V_T , I_{DS} est nul ou négligeable.

Si les V_{GS} est supérieures à V_T , les électrons sont attorés dans le canal.

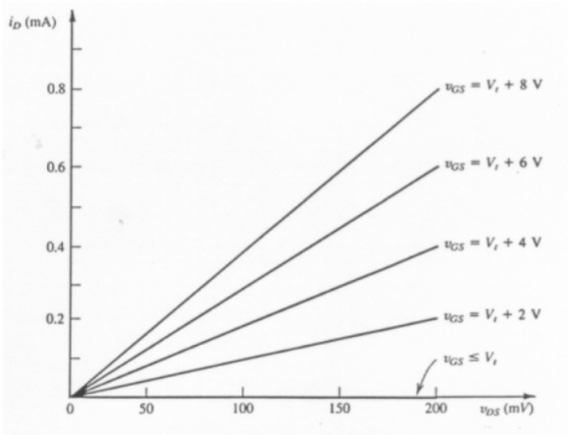


FIGURE 12 – Caractéristiques I_D-V_{DS} d'un MOSFET pour des faibles tensions Drain-Source

L'influence de la tension V_{GS} (au-dessus de V_T) est représentée avec l'augmentation de la profondeur du canal et donc une augmentation de la conductance du canal (diminution de la résistance).L'I_{DS} circulant dans le canal est donc proportionnel à (V_{GS}-V_T) et à la tension Drain-Source.

3.5 Transistors HEMT

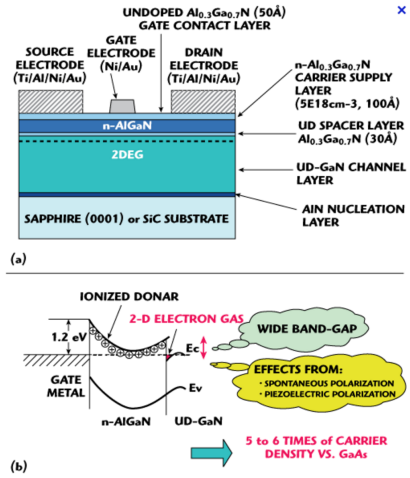


FIGURE 13 – Shéma du HEMT

Le HEMT (High Electron Mobility Transistor), est un type de FET qui permet de travailler avec des fréquences micro-ondes très élevées. La particularité du HEMT c'est sa hétérojonction NP (différents matériaux utilisés dans le deux côtés de la jonction). Le HEMT est différent aux FET et permet d'assurer une plus haute performance, particulièrement dans les applications des ondes radios.

Dans le cadre de notre projet, les données utilisées appartiennent à des HEMT GaN. Il s'agit de transistors récemment mis au marché pouvant remplacer les HEMT à base de Si grâce à ses avantages. Ceci dit, les HEMT GaN disponibles sont encore en-dessous de leurs limite théorique.

Les HEMT peuvent être *"Normally-ON"* ou *"Normally OFF"*

— **Normally ON** (passant) :La tension V_{GS} entre la source et la grille doit être inférieure à celle du seuil V_{th} négative pour être maintenu bloqué.

— **Normally OFF** (bloqué) :La tension V_{GS} doit être en-dessous d'une tension V_{th} pour être maintenu bloqué.

Le HEMT est créé à partir de couches minces de GaN et d'autres matériaux (comme l'ALGaN) sur un substrat compatible. L'hétéro-épitaxie (cristal qui croît dans le substrat) permet la croissance du GaN dans un matériau avec lequel il possède un faible désaccord de maille. Le saphir, par exemple, possède une maille de désaccord avec le GaN de 14% . Mais pour des raisons de prix on utilise le plus souvent le Si comme substrat malgré un désaccord de maille de 20% avec le GaN. Ceci produit une résistivité qui empêche le fonctionnement à haute fréquence du transistor.

À l'hétérojonction, plusieurs électrons de la région type N restent à proximité de la jonction hétérogène et forment ce qu'on appelle un gaz bidimensionel d'électrons(ou 2DEG pour 2-Dimension). À l'intérieur de gaz la résistivité est très faible et par conséquence la mobilité des électrons dans le gaz est importante. Un bias appliquée dans la porte est utilisé pour contrôler le nombre d'électrons dans le canal ce qui permet de contrôler la conductivité du transistor.

3.6 HEMT VS MOSFET

Les MOSFET et HEMT ont des caractéristiques électriques différentes (qu'on observe dans le tableau) :

— La charge totale GQ de la grille des HEMT GaN est inférieure

— La grille des HEMT est plus fragile (limite V_{GS} de -5V/+6V contre 20V pour les MOSFET)
— La température de fonctionnement maximale de HEMT est plus faible et les fuites de courant (grille et drain) sont supérieures à celles du MOSFET

	MOSFET Si IPD33CN10NG [45]	HEMT GaN EPC2007 [39]
Tension de claquage V_{BR}	100 V	100 V
Résistance à l'état passant R_{ON}	33 mΩ	30 mΩ
Charge de grille totale Q_G	18-24 nC (V_{gs} de 0 à 10V)	2.1-2.8 nC (V_{gs} de 0 à 5V)
Tension grille-source min $V_{gs,min}$/max $V_{gs,max}$	± 20 V	-5 V / +6 V
Tension de seuil V_{th}	2-4 V	0.7-2.5 V
Temps de recouvrement diode interne t_{rr} à I_S=27A	77 ns	0 ns
Charge de recouvrement de la diode interne Q_r	154 nC	0 nC
Température max de fonctionnement T_j	175 °C	125 °C
Courant de fuite de grille I_{GSS} à l'état ON à 25°C	1-100 nA (V_{gs}=20V)	0.25-2 mA (V_{gs}=5V)
Courant de fuite de grille I_{GSS} à l'état OFF à 25°C	-	-0.25 mA (V_{gs}=-5V)
Courant de fuite de drain I_{DSS} à V_{gs}=0V,V_{ds}=80V et 25°C	0.1-1 µA	20-60 µA

FIGURE 14 – Tableau de comparaison à 25°C des caractéristiques électroniques du MOSFET et du HEMT GaN

4 Application Scilab

Nous avons développé l'application avec le logiciel Scilab. Le code se divise en quatre grandes parties : l'extraction des données, la création de la base de données, le traçage des fonctions et la création de l'interface graphique. Tous les algorithmes ont été testés avec différentes données pour vérifier leur fonctionnement.

4.1 Présentation générale

4.1.1 Fonctionnalités et algorithmes

Nous allons présenter la documentation/explication des fonctions utilisées :

— **function entete_fichier=extraction_entete(fichier)**

Cette fonction permet de récupérer l'en-tête sous forme d'un tableau.

— **function valeurs_fichier=extraction_valeurs(texte)**

Permet de récupérer les valeurs d'un fichier. Pour cela on vérifie que chaque case du tableau est un bien un nombre. Ceci pour ne pas récupérer les en-têtes à chaque fois.

— **function matrice_valeurs=valeurs_en_matrice(valeurs_fichier)**

Permet de transformer une matrice d'une colonne de type chaine de caractères en une matrice à plusieurs colonnes (on définit les séparateurs entre les colonnes par "**;**") de type réel.

— **function [matrice_valeurs_entete, col_matrice_valeurs, row_matrice_valeurs]=inserer_entete_valeurs(matrice_valeurs_init, entete_fichier)**

Permet d'insérer au début de la matrice les informations contenues dans l'en-tête. Par exemple la température, input/output, la date.

— **function matrice_tempo=creation_fichier(matrice_valeurs_entete, entete_fichier)**

Cette fonction permet de compléter la base de données lors de l'ajout d'un nouveau fichier de mesure. Elle teste si dans le dossier "BASE" il existe déjà un fichier pour le composant ajouté. Si oui, il faut ajouter à la fin de ce fichier les nouvelles mesures. Sinon, on créé un nouveau fichier pour ce composant et on y écrit les mesures.

— **function [X, Y]=tracer_courbes(matrice_valeurs, row_matrice_valeurs, temperature, axeX, axeY, tension_col, tension_valeur, erreur)**

Permet de récupérer une matrice X et une matrice Y pour tracer une courbe. Pour sélectionner les valeurs qui intéressent l'utilisateur, il faut en entrée la grandeur de l'axe X, la grandeur de l'axe Y, une grandeur de tension et sa valeur, et finalement l'erreur permise sur cette valeur. Ceci afin de sélectionner toutes les valeurs des grandeurs axeX et axeY lorsque tension_col=tension_valeur +/- erreur.

4.2 Interface Graphique

On a pu développer l'interface graphique directement sur Scilab en intégrant une partie du code déjà faite. Après avoir créer une figure (fenêtre), nous avons créer des "handles" afin de paramétrer les différents objets que l'on souhaitait insérer dans l'interface graphique. Il a ensuite fallu créer les différentes fonctions associées à ces différents objets de l'interface graphique. Nous avons utilisé pour cela le paradigme de la programmation évènementiel, c'est-à-dire nous avons programmé avec des "callback" (en français : fonctions de rappel), ce qui était la plus grande difficulté. Il s'agit de fonctions qui sont passées en paramètre à d'autres fonctions. Nous avons ainsi pu relier les clics-souris et les valeurs choisis et entrées avec les fonctions du programme. Voici les différentes fonctions :

— **function exit_scilab(handles)**

Si l'on clique sur le bouton "Application", cette fonction est appelée et

permet à l'utilisateur de fermer l'application.

— **function notice_scilab(handles)**

Si l'on clique sur le bouton "Notice", cette fonction est appelée et permet à l'utilisateur de lire quelques instructions à propos de l'application.

— **function pb_fichier_callback(handles)**

Si l'on clique sur le bouton "Fichier", cette fonction est appelée et permet à l'utilisateur de choisir un fichier.

— **function pb_tableau_callback(handles)**

Si l'on clique sur le bouton "Tableau", cette fonction est appelée et permet à l'utilisateur de créer un fichier avec toutes les valeurs qui respectent les paramètres choisis par l'utilisateur avec les autres fonctions callbacks.

— **function pb_tracer_callback(handles)**

Si l'on clique sur le bouton "Tracer", cette fonction est appelée et permet à l'utilisateur de tracer un graphe à partir des différents paramètres définis par l'utilisateur avec les autres fonctions callbacks.

— **function pb_effacer_callback(handles)**

Si l'on clique sur le bouton "Effacer", cette fonction est appelée et permet à l'utilisateur d'effacer la courbe tracée, et les messages d'erreurs affichés.

— **function [x,mpp]=liste_callback(handles)**

Cette fonction permet de renvoyer la valeur sélectionnée, qui correspond à un numéro de colonne dans la base de donnée, par l'utilisateur pour l'abscisse (x) ainsi que le nom de la valeur (mpp - string).

— **function [VX,VV]=liste_vdsvgs_callback(handles)**

Cette fonction permet de renvoyer la valeur entrée, qui correspond à un numéro de colonne dans la base de donnée, par l'utilisateur pour le choix entre Vgs et Vds (VX) ainsi que le nom de la valeur (VV - string). Ce choix permet de prendre uniquement les données ayant comme Vgs/Vds la valeur entrée +- erreur.

— **function [err]=erreur_callback(handles)**

Cette fonction permet de renvoyer la valeur sélectionnée par l'utilisateur pour l'erreur (permettant de définir l'intervalle des Vgs/Vds)

— **function [y,s]=listey_callback(handles)**

Cette fonction permet de renvoyer la valeur sélectionnée, qui correspond à un numéro de colonne dans la base de donnée, par l'utilisateur pour l'abscisse (y) ainsi que le nom de la valeur (s - string).
De plus, nous avons utilisé le mot-clé global en début de programme qui nous a permis de partager certaines variables en lecture/écriture entre certaines fonctions. Toutes les affectations à ces variables sont propagées à toutes les autres fonctions ayant déclaré cette variable globale. Voici les différentes variables globales que nous avons utilisé : **global** s p X Y obj VV VX err Filename. La variable p représente le string du nom de la variable des abscisses, le X et le Y représentent les données (après le paramétrage) et obj représente la valeur de la température entrée. Nous avons choisis de mettre ces variables en globales car elles sont utilisées dans plusieurs fonctions.

Quatrième partie

Conclusions et perspectives

Nous avons rencontré différentes types de difficultés dont la principale fut l'utilisation d'un logiciel auquel nous n'étions pas familiarisés. Ceci a été particulièrement vrai pour la conception de l'interface graphique. Une autre difficulté que nous avons rencontré a été la mise en commun du travail individuel pour la conception de l'application.

Néanmoins, il nous a paru intéressant de développer une application, puisque aucun d'entre nous ne l'avait déjà fait.Nous avons pu acquérir des nouvelles compétences informatiques, en ce qui concerne la programmation par callback et le paradigme de la programmation évènementiel ; ainsi que des nouvelles méthodes de réflexion.

Enfin, nous avons pu prendre concience du travail de recherche d'un doctorant et approfondir nos connaisances sur les transistors.

Références

[1] Junction field effect transistor. http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/fet.html. Accessed : 2016-03-12.

[2] Junction field effect transistor. http://www.electronics-tutorials.ws/transistor/tran_5.html. Accessed : 2016-03-19.

[3] Les transistors à effet de champ, howpublished = http://ressources.univ-lemans.fr/accueslibre/um/pedago/physique/02/cours_elec/jfet.pdf, note = Accessed : 2016-04-16.

[4] Transistors. https://learn.sparkfun.com/tutorials/transistors. Accessed : 2016-02-20.

[5] Transistorsaeffetdechamp, howpublished = http://users.polytech.unice.fr/~cpeter/elec/docs_cours/5_transistorfet.pdf, note = Accessed : 2016-06-01.

[6] What is a mosfet : Basics tutorial. http://www.radio-electronics.com/info/data/semicond/fet-field-effect-transistor/mosfet-basics-tutorial.php. Accessed : 2016-02-25.

[7] Romain Grézaud. *Commande de composants grand gap dans un convertisseur de puissance synchrone sans diodes*. PhD thesis, Energie électrique. Université de Grenoble. Français. <tel- 01121124>, 2014.

[8] Ian poole. Hemt, high electron mobility transistor tutorial. *radio-electronics.com*.

[9] Julio Brandelero, Bernardo Cougo, Thierry Meynard, Nicolas Videau, Olivier Goualard, Xavier Bonnin, Henri Schneider. *valuation des pertes par commutation pour la conception des convertisseurs et applications des composants grand gap*. PhD thesis, Symposium de Génie Electrique 2014, Jul 2014, Cachan, France.<hal-01065310>, 2014.

[8] [1] [2] [6] [3] [4] [5] [?] [7] [9]

Cinquième partie

Annexes

5 Code de l'application

```
1
2  // Corps du programme
3
4  //variable globale Filename=fichier choisi par l' utilisateur .
5  global s p X Y obj VV VX err Filename
6
7  function exit_scilab(handles) //option pour quitter l'application
8      msg = gettext("Voulez vous vraiment quitter l''application ?");
9      answ = messagebox(msg, gettext("Quitter Scilab"), "quextion", [gettext("Oui") gettext
        ("Non")], "modal");
10     if answ==1 then
11         //exit;
12         close(f);
13     end
14 endfunction
15
16 function notice_scilab(handles) //donne quelques indications pour savoir utiliser l'
       application
17     messagebox(["Fichier : extension csv." " Paramtres : Veuillez bien suivre l''ordre
          prdfinis " "Temprature: en degr Celsius." "Tension : en V" "Bouton tableau :
       permet de crer un fichier csv contenant les valeurs paramtres du fichier
          slectionn ."], "NOTICE")
18 endfunction
19
20 function pb_fichier_callback(handles)        //renvoie le fichier choisi
21     global Filename
22     Filename=uigetfile(["*.csv"])
23 endfunction
24
25
26
27 function pb_tableau_callback(handles)  //permet de crer un tableau      partir des
       donnes selectionnes (paramtrage).
28     nom=s+"_contre_"+p+"_T="+obj.string+"_"+VV+"="+string(VX)+"_e="+
       string(err)+".csv";
29     // nom="Tableau"+p+""+s+obj.string+VV+".csv";
```

```
30      filename= fullfile (nom);
31      Z=[p s;string(X) string(Y)]
32     csvWrite(Z,filename);
33          obje=findobj('tag','error');
34          obje.string="Fichier  cre ";
35          obje.BackgroundColor = [0.4 0.2 0.4];
36 endfunction
37
38
39
40 function pb_tracer_callback(handles) //permet de tracer un graphe avec des paramtres
          prdfinis  .
41 global s p X Y obj VV VX err Filename
42
43     pb_effacer_callback(handles);
44
45     // Cration d'une matrice  partir du fichier
46     texte=read_csv(Filename);
47     entete_fichier=extraction_entete(texte);
48     // test
49 if entete_fichier=='input' | entete_fichier=='output' then
50     texte=read_csv(Filename);
51     matrice_valeurs=csvRead(Filename,";",",","string");
52 else
53     disp(entete_fichier)
54     [valeurs_fichier, position_entete]=extraction_valeurs(texte);
55     matrice_valeurs=valeurs_en_matrice(valeurs_fichier);
56     [matrice_valeurs_entete, col_matrice_valeurs, row_matrice_valeurs]=
        inserer_entete_valeurs(matrice_valeurs, entete_fichier);
57     [sortie_fonction, filename]=creation_fichier(matrice_valeurs_entete, entete_fichier);
58     matrice_valeurs=csvRead(filename,";",",","string");
59 end
60     // Variables des paramtres choisis par l' utilisateur
61     [x1,p]=liste_callback(handles)  //x1 prend l' ordonne et p le nom string.
62     [y1,s]=listey_callback(handles) //y1 prend l' ordonne et p le nom string.
63     [err]=erreur_callback(handles) //prend la valeur de l'erreur  entre .
64     [VX,VV]=liste_vdsvgs_callback(handles)  //VX prend la tension choisie (numro de
        colonne) et VV la valeur entre .
65     obj= findobj('Tag','temperature'); //obj prend la valeur de la temprature  entre .
66     objV= findobj('Tag','vdsvgs');  //objV prend le nom Vds ou Vgs  slctionn  ..
67     obj1= findobj('Tag','error'); //obj1 est la case qui est  affich  pour les messages d
        'erreurs.
68     obj1.string="";
```

```
69  obj1.BackgroundColor = [1 1 1];
70
71  // On vrifie si les paramtres sont correctes
72  Verification1 =find((strtod(matrice_valeurs(:,3))==strtod(obj.string)))
73  if Verification1 ==[] then
74      obje=findobj('tag','error');
75      obje.string="Valeur temperature non valide";
76      obje.BackgroundColor = [0.9,0.1,0.1];
77  end
78  Verification2 =find(strtod(matrice_valeurs(:,VX))<=(strtod(objV.string)*(1+err)) &
     strtod(matrice_valeurs(:,VX))>=(strtod(objV.string)*(1-err)))
79  if Verification2 ==[] then
80      obje=findobj('tag','error');
81      obje.string="Valeur tension non valide";
82      obje.BackgroundColor = [0.9,0.1,0.1];
83  end
84  if Verification2 ==[] & Verification1==[] then
85      obje=findobj('tag','error');
86      obje.string="Valeurs entres tension/temprature non valides";
87      obje.BackgroundColor = [0.9,0.1,0.1];
88  end
89
90  // On trace la courbe selon les paramtres (correctes) dfinis .
91  if Verification1~=[] & Verification2~=[] then
92      [X,Y,I]=tracer_courbes(matrice_valeurs,strtod(obj.string), x1,y1,VX,strtod(objV.
     string),err);
93      [r2 c2]=size(I);
94      traceX=X(1);
95      traceY=Y(1);
96      for (i=2:c2-1)
97          if (X(i,1)>-1E-3) & (X(i,1)<1E-3) then
98              X(i,1)=0;
99          end
100         if (Y(i,1)>-1E-3) & (Y(i,1)<1E-3) then
101             Y(i,1)=0;
102         end
103         if (X(i,1)<X(i+1,1)) then
104             traceX=[traceX X(i+1)];
105             traceY=[traceY Y(i+1)];
106         else
107             plot(traceX,traceY);
108             traceX=X(i);
109             traceY=Y(i);
```

```
110             end
111         end
112          xtitle ("Courbe",p,s)
113                 obje=findobj('tag','error');
114         obje.string="Courbe trace";
115         obje.BackgroundColor = [0.4 0.2 0.4];
116     end
117
118
119 endfunction
120
121 function [err]=erreur_callback(handles) //renvoie la valeur de l'erreur choisie.
122     pop = findobj("Tag", "erreur");
123     selected = get(pop, "Value");
124     err=selected
125 endfunction
126
127 function pb_effacer_callback(handles)                    //effacer la courbe
128     delete(handles.obj4.children);
129      obj1= findobj('Tag','error');
130     obj1.string="";
131     obj1.BackgroundColor = [1 1 1];
132 endfunction
133
134 function [x,mpp]=liste_callback(handles)          // le x prend la valeur de l'abscisse et
       le mpp le string de l'abscisse
135     objsd = findobj("Tag", "liste");
136     items = get(objsd, "String");
137     selected = get(objsd, "Value");
138     mpp=items(selected)
139     select mpp,
140     case "Ig" then x=4,
141     case "Id" then x=5,
142     case "Vgs" then x=6,
143     case "Vds" then x=7,
144     end
145 endfunction
146
147
148
149 function [VX,VV]=liste_vdsvgs_callback(handles) //renvoie VX la nom de la tension
       choisie (numro de la colonne) ainsi que VV la valeur choisie.
150     pop = findobj("Tag", "liste_vdsvgs");
```

```
151     items = get(pop, "String");
152
153     selected  = get(pop, "Value");
154     VV=items(selected)
155     select  items(selected),
156     case "Vgs" then VX=6,
157     case "Vds" then VX=7,
158     end
159     drawnow();
160 endfunction
161
162 function [y,s]=listey_callback(handles) // le y prend la valeur de l' ordonne  et le s le
        string de l' ordonne .
163     drawlater(); clear;
164     pop = findobj("Tag", "listey");
165     items = get(pop, "String");
166     selected  = get(pop, "Value");
167     s=items(selected)
168     select  items(selected),
169     case "Ig" then y=4,
170     case "Id" then y=5,
171     case "Vgs" then y=6,
172     case "Vds" then y=7,
173     end
174     drawnow();
175 endfunction
176
177 // Fonctions permettant la  cration  d'une matrice  partir d'un fichier
178
179 function entete_fichier=extraction_entete(fichier)
180     //Extrait l'en- tte du fichier
181     tmp=fichier(2);
182     filename= fullfile (TMPDIR,"data.csv");
183     csvWrite(tmp,filename,ascii(9));
184     entete_tableau=csvRead(filename,";",[],"string");
185
186     info=entete_tableau(2);
187     filename= fullfile (TMPDIR,"data.csv");
188     csvWrite(info,filename, ascii (9));
189     entete_fichier=csvRead(filename,"_",[],"string");
190 endfunction
191
192
```

```
193 function [valeurs_fichier, position_entete]=extraction_valeurs(texte)
194
195     compteur_valeurs=1;
196     compteur_position=1;
197     taille_texte=size(texte,"r");
198
199     for (i=1:taille_texte)
200         ligne_ascii=ascii(texte(i));
201         if(ligne_ascii(1)>47 & (ligne_ascii(1)<58))
202             valeurs_fichier(compteur_valeurs)=texte(i);
203             compteur_valeurs=compteur_valeurs+1;
204         else
205             position_entete_trans(compteur_position)=compteur_valeurs;
206             compteur_position=compteur_position+1;
207         end
208     end
209     position_entete=unique(position_entete_trans);
210 endfunction
211
212
213 function matrice_valeurs=valeurs_en_matrice(valeurs_fichier)
214     filename= fullfile(TMPDIR,"data1.csv");
215     csvWrite(valeurs_fichier,filename, ascii(9));
216
217     matrice_valeurs=csvRead(filename,";",",","string");
218 endfunction
219
220
221 function [matrice_valeurs_entete, col_matrice_valeurs, row_matrice_valeurs]=
        inserer_entete_valeurs(matrice_valeurs_init, entete_fichier)
222
223     col_matrice_valeurs=size(matrice_valeurs_init,"c");
224     row_matrice_valeurs=size(matrice_valeurs_init,"r");
225
226     longueur_temperature=length(entete_fichier(1,4))-1;
227     temperature=part(entete_fichier(1,4),1:longueur_temperature);
228
229     colonne=1;
230     for i=2:3
231         for j=1:row_matrice_valeurs
232             matrice_valeurs_entete(j,colonne)=entete_fichier(1,i);
233         end
234         colonne=colonne+1;
```

```
235     end
236
237     matrice_valeurs_entete(:,colonne)=temperature;
238
239     for i=2:col_matrice_valeurs
240         for j=1:row_matrice_valeurs
241             matrice_valeurs_entete(j,colonne+1)=matrice_valeurs_init(j,i);
242         end
243         colonne=colonne+1;
244     end
245 endfunction
246
247 function [sortie_fonction, filename]=creation_fichier(matrice_valeurs_entete,
        entete_fichier)
248     racine=ls('C:\Users\Jeremy\Documents'); // ATTENTION ADRESSE A MODIFIER
249
250     filename= fullfile (entete_fichier(1)+".csv");
251     if (members(filename,racine)==0) then
252         csvWrite(matrice_valeurs_entete,filename,";");
253         matrice_tempo=[];
254         sortie_fonction=1;
255     else
256         matrice_tempo=csvRead(filename, ";", [], "string");
257         test=%F;
258         row_matrice_tempo=size(matrice_tempo,"r");
259         i=1;
260         longueur_temperature=length(entete_fichier(1,4))-1;
261         temperature=part(entete_fichier(1,4),1:longueur_temperature);
262         while ((i<row_matrice_tempo+1) & (test==%F))
263             test=((matrice_tempo(i,1)==entete_fichier(2)) & (matrice_tempo(i,2)==
        entete_fichier(3)) & (matrice_tempo(i,3)==temperature));
264             i=i+1;
265         end;
266         if (test==%F)
267             matrice_tempo=[matrice_tempo;matrice_valeurs_entete];
268             csvWrite(matrice_tempo,filename,";");
269         end;
270         sortie_fonction=0;
271     end;
272 endfunction
273
274
275 function [X,Y,I]=tracer_courbes(matrice_valeurs,temperature, axeX,axeY,tension_col,
```

```
      tension_valeur,erreur) // renvoie le X  tracer en fonction du Y.
276
277     I=find((strtod(matrice_valeurs(:,3))==temperature)&(strtod(matrice_valeurs(:,
         tension_col))<=(tension_valeur*(1+erreur)))&(strtod(matrice_valeurs(:,tension_col)
         )>=(tension_valeur*(1-erreur))));
278     X=strtod(matrice_valeurs(I,axeX));
279     Y=strtod(matrice_valeurs(I,axeY));
280
281 endfunction
```