

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MOULOUD MAMMERI DE TIZI-OUZOU



FACULTÉ DE GÉNIE ÉLECTRIQUE ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Études de **MASTER ACADÉMIQUE**

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Réseaux, Mobilité et Systèmes
Embarqués**

Présenté par
Cylia BACHA
Lynda KAJITE

Thème

Générateur d'interfaces tactiles pour la carte LPC 3250

Mémoire soutenu publiquement le 22/09/2016. devant le jury composé de :

Présidente : Mme Rachida AOUDJIT - Université Mouloud MAMMERI de Tizi-Ouzou

Encadreur : M Mehammed DAOUI - Université Mouloud MAMMERI de Tizi-Ouzou

Examinatrice: Mme Rebiha HADDAOUI - Université Mouloud MAMMERI de Tizi-Ouzou

Examinatrice : Mme Karima OUKFIF - Université Mouloud MAMMERI de Tizi-Ouzou

Remerciements

A Monsieur DAOUI, notre encadreur,

Nous vous sommes extrêmement reconnaissantes d'avoir accepté de nous encadrer. Nous avons eu la chance de bénéficier de vos conseils éclairés pour la réalisation de notre travail. Nous vous adressons ici nos sincères remerciements et nous vous prions de croire en notre plus haute considération.

Aux membres du jury,

Nous tenons à vous remercier d'avoir accepté d'être membre de notre jury. Nous sommes sensibles à l'attention que vous avez bien voulu porter à ce travail. Nous vous exprimons ici l'assurance de notre profond respect.

A toute personne ayant contribué au bon déroulement de notre travail,

Nous tenons à vous remercier d'avoir été à nos côtés. Nous vous présentons ici nos sincères remerciements.

Dédicaces

A nos parents,

Que dieu vous donne santé et longue vie, je vous remercie pour votre continuel soutien et vos sacrifices.

A nos frères et sœurs,

Que dieu soit à vos côtés pour illuminer vos chemins, je vous remercie pour vos conseils et encouragements.

A tous nos amis(es),

En particulier **MASSI, SEDDIK, IMENE, LYNDA, CELIA, AMEL, LYCIA, TASSASITE, NACERA, ...**

A nos familles.

LYNDA et CELIA

SOMMAIRE

Introduction générale.....	2
----------------------------	---

Chapitre I : Généralités sur les technologies des écrans LCD

1. Introduction.....	5
2. Historique.....	5
3. Quelques notions de base concernant les LCD	6
3.1. Définition d'un pixel	6
3.2. Profondeur des couleurs	6
3.3. Ecrans monochromes	7
4. Types de panneaux LCD.....	7
4.1. L'écran LCD à matrice active ou passive	7
a. Ecran LCD-TFT	8
b. Ecran LCD-STN.....	9
c. Ecran TN	9
d. Ecran PLASMA	10
5. Caractéristiques de l'écran LCD	10
6. Les contrôleurs LCD.....	10
6.1. Paramétrage d'un contrôleur LCD	11
7. Frame buffer.....	13
Conclusion	15

Chapitre II : Généralités sur les technologies tactiles

1. Introduction	17
2. Ecran tactile.....	17
2.1. Définition	17
2.2. Principe de fonctionnement.....	18
2.3. Terminologie	19
2.4. Technologies des interfaces tactiles	20

2.4.1. Technologie capacitive	20
2.4.2. Technologie résistive	21
2.4.3. Technologie infrarouge	22
2.4.4. Technologie à onde de surface	22
2.4.5. Technologie optique.....	23
2.5. Contrôleur de l'interface tactile	24
2.5.1. Fonctionnement du contrôleur TSC de la LPC3250	24
2.5.2. Paramétrage du contrôleur TSC	26
Conclusion	28
 Chapitre III : Analyse et conception	
1. Introduction	30
2. Analyse et conception	30
2.1. Analyse.....	30
2.1.1. Idée générale sur notre travail	30
2.2. Conception	33
2.2.1. Comportement de l'application	33
2.2.2. Aperçu de quelques algorithmes	37
Conclusion	41
 Chapitre IV : Réalisation et tests	
1. Introduction	43
2. Outils de développement utilisés.....	43
2.1. Outils logiciels	43
2.1.1.Keil- μ vision	43
2.1.2. Le langage C.....	44
2.1.3. L'Extensible Markup Language(XML)	45
2.2. Outils matériels	46
3. Fonctionnement de notre application.....	46
3.1. Description du générateur	46
1. Ecriture	50

2. Suppression	51
3. L'action des boutons	51
Conclusion.....	56
Conclusion générale et perspectives	58
Références bibliographiques	59
Annexe.....	62

TABLE DES FIGURES

Chapitre I : Généralités sur les technologies des écrans LCD

Figure I.1-Le codage des couleurs sur 16 bits	6
Figure I.2-Représentation d'un caractère dans le frame buffer et son affichage sur l'écran	14

Chapitre II : Généralités sur les technologies tactiles

Figure II.1-Principe de fonctionnement des interfaces tactiles	18
Figure II.2-Technologie capacitive	20
Figure II.3-Technologie résistive	21
Figure II.4-Technologie à infrarouge	22
Figure II.5-Technologie à onde de surface	23
Figure II.6-Technologie optique	24
Figure II.7-Fonctionnement du contrôleur TSC	25

Chapitre III : Analyse et conception

Figure III.1-Fonctionnement de notre système	31
Figure III.2-Le diagramme de séquence des interactions entre l'utilisateur et le générateur	34
Figure III.3-Le diagramme de séquence des interactions entre l'utilisateur et une zone de texte	35
Figure III.4-Le diagramme de classe de l'interface graphique	36

Chapitre IV : Réalisation et tests

Figure IV.1-La plateforme Keil- μ vision	45
Figure IV.2-Exemple de fichier XML	47
Figure IV.3-Interface tactile	48
Figure IV.4-Appuis sur l'une des zones	50
Figure IV.5-Saisie dans la zone de texte	51
Figure IV.6-Avant la suppression	52

Figure IV.7-Après la suppression	52
Figure IV.8-Remplissage de toutes les zones de textes	52
Figure IV.9-Opération effectué avec sucée	53
Figure IV.10-Signalisation d'un champ non remplis.....	54
Figure IV.11-Action du bouton calculer	55

Introduction générale

Introduction Générale

Introduction générale

Aujourd'hui, beaucoup de grandes avancées scientifiques ont été enregistrées dans le domaine des technologies embarquées et mobiles. En effet, au cours des dix dernières années, les systèmes embarqués ont envahis notre vie au quotidien. Des plus petits (étiquettes électroniques) jusqu'aux plus gros (satellites), en passant par les téléphones portables et appareils électroménagers. Ces avancées ont profondément modifié notre comportement avec leurs divers degrés d'utilité et de nécessité tant dans les pays avancés que dans les pays en voie de développement.

Ces systèmes sont des dispositifs intelligents qui ont des puissances de calcul de plus en plus grandes, et dont les moyens de communication leur permettent d'interagir avec le monde environnant. A l'origine, ce sont des systèmes matériels et logiciels intégrés dans des avions militaires ou des missiles, mais actuellement ils se sont étendus à des domaines divers tels que : le domaine publique, les moyens de transport, les équipements médicaux, la communication et l'industrie.

L'évolution des technologies de miniaturisation offre aujourd'hui des perspectives d'intégration importantes. Ces systèmes complexes peuvent être intégrés ou embarqués dans des microcontrôleurs, exécutant des opérations en temps réel. La miniaturisation de toutes ces fonctions permet aux objets d'interagir, de communiquer entre eux (wifi, Bluetooth) sans intervention humaine. Les objets communicants sont devenus un des axes stratégiques de développement de l'économie mondiale. [1]

De nombreux systèmes embarqués sont équipés d'interfaces tactiles. Les interfaces tactiles sont dotées d'un écran LCD (Liquid Crystal Display) et un film tactile permettant de détecter la position du doigt ou d'un stylet, les Smartphones et les tablettes tactiles sont les meilleurs exemples de ces systèmes.

Bien que le développement d'interfaces graphiques soit acquis dans les ordinateurs conventionnels, grâce à la standardisation des moniteurs graphiques et aux nombreux outils de génération d'interfaces existants, la situation dans les systèmes embarqués est plus complexe. Il existe de nombreuses technologies d'interfaces et leur gestion au niveau bas (drivers) nécessite une grande maîtrise des paramètres et la documentation est parfois limitée aux manuels des constructeurs. Toutefois, un effort de standardisation est en cours et les générateurs d'interfaces présents dans les outils de développement pour Smartphone (Android, iOS, Windows mobile,... etc.) sont les résultats de cet effort. Toutefois ces outils ne génèrent d'interfaces que pour des applications fonctionnant sur des systèmes appropriés.

Introduction Générale

Dans le cadre de notre projet de fin d'étude, nous allons concevoir et développer un générateur d'interfaces graphiques pour la carte Phycore LPC 3250 disponible au niveau de notre département. Cette carte de développement dispose d'un écran LCD (HITACHI) résistif de type TFT, 320*240 pixels, nombre de couleurs 262^K (R, V, B), alimentation 3.3v et d'un clavier tactile. Cet outil permettra de faciliter la création d'interfaces tactiles pouvant être intégrées dans des projets de développements pour un system embarqué. Ce générateur pourra créer des interfaces contenant des composants usuels tels que les boutons, zones de textes, graphiques...etc.

Ce mémoire s'organise en quatre chapitres, à travers lesquels nous décrivons le travail effectué pour la conception et la réalisation de notre système.

Le chapitre 1 sera consacré aux technologies des écrans LCD dans le contexte des systèmes embarqués.

Le chapitre 2 est consacré aux technologies tactiles. On y présente d'abord le principe général du fonctionnement de ces interfaces puis on donnera un état de l'art des technologies existantes. On terminera le chapitre par le contrôleur tactile de la LPC 3250 et son utilisation.

Le chapitre 3 présentera l'approche adoptée pour l'analyse et la conception de notre projet. On décrira d'abord les objectifs de notre système puis on décrira ces différents composants illustrés par des figures et diagrammes.

Le chapitre 4 présentera les outils de développement utilisés et décrira les étapes de réalisation de notre application et les résultats des tests de quelques interfaces générées par notre système.

Nous terminerons notre projet par une conclusion générale et des perspectives. Nous avons ajouté à la fin du mémoire, une annexe présentant la carte Phycore LPC 3250.

Chapitre I :

Généralités sur les

technologies des écrans LCD

Chapitre I : généralités sur les technologies des écrans LCD

1. Introduction

Depuis maintenant plus de 5 ans, l'écran plat gagne de plus en plus en part de marché au dépend des écrans à tube. Si à leur sortie les écrans plats furent beaucoup critiqués, il est apparu une envie de réaliser des écrans capables d'être moins encombrants, plus légers, et une consommation réduite tout en gardant une bonne qualité d'image. L'écran LCD représente aujourd'hui l'écran plat type, il est donc le plus utilisé dans tous les domaines tel que l'informatique, la TV et les systèmes embarqués.

Grâce à la miniaturisation de différents composants, ces écrans sont omniprésents. On les trouve dans de nombreux appareils électroniques disposant d'afficheurs : les montres, le tableau de bord des voitures, les calculatrices, les téléphones portables ...etc. L'utilisation d'un afficheur à cristaux liquides ou LCD est assez courante dans le domaine des technologies embarquées et mobiles. Ces afficheurs issus des technologies de miniaturisation sont employés pour informer l'utilisateur sur l'état du système, lancer des applications dédiées aux fonctionnalités spéciales ou afficher les mesures effectuées par différents capteurs. [2]

Dans ce chapitre, nous allons commencer par présenter un bref historique sur les écrans LCD et leur développement, nous détaillerons ensuite les types de panneaux LCD puis nous allons voir le contrôleur LCD de la carte Phycore LPC 3250 utilisé pour communiquer avec l'afficheur à cristaux liquides. Enfin nous terminerons par quelques caractéristiques de l'écran LCD disponible sur la carte LPC 3250.

2. Historique

En 1964, l'ingénieur américain, George Harry Heilmeyer invente l'affichage à cristaux liquides. L'écran à cristaux liquides utilise la polarisation de la lumière via des filtres polarisants. En fait il n'émet pas de lumière, il doit la recevoir. Ces premiers types d'affichages, ont été présentés en 1971.

En 1984, Thomson développe le premier écran à cristaux liquides en couleur. Dans les années 90 les téléphones portables, les ordinateurs et les téléviseurs sont équipés d'afficheur LCD en noir et blanc.

Les écrans couleurs arrivent sur les marchés dans les années 2000. Aujourd'hui ces écrans sont utilisés dans presque tous les affichages électroniques. [3]

Chapitre I : généralités sur les technologies des écrans LCD

3. Quelques notions de base concernant les LCD

3.1. Définition d'un pixel

Un pixel est souvent l'assemblage électronique de trois "points": un rouge + un vert + un bleu. La couleur est produite par l'addition de ces trois couleurs à des intensités variables, ce qui permet de produire toute une palette de couleurs différentes.

- Si on allume seulement le rouge, alors le pixel sera rouge. Si on allume aucun point (R, V, B) = (0, 0, 0) le pixel sera noir. Si on allume tous les points, le pixel sera blanc.
- Un point peut être à moitié allumé, ce qui permet d'obtenir des nuances. Par exemple, pour obtenir un pixel gris intermédiaire entre le blanc et le noir, on allumera à moitié les points rouge, vert et bleu.
- Ensuite, les pixels sont eux-mêmes assemblés sur ce qu'on appelle une dalle. Cette dernière est en fait la base de l'écran. [4]

3.2. Profondeur des couleurs

La profondeur de couleur, dont l'unité est le bit par pixel (bpp), décrit le nombre de bits utilisés pour représenter la couleur d'un pixel.

Une plus grande profondeur de couleurs nécessite un plus grand nombre de bits et permet une plus grande échelle de nuances dans les couleurs :

8-bit Color (256 Couleurs).

15/16 bit: Highcolor (Haute couleur, plus de 65 000 couleurs).

24/32 bit: Truecolor (Vraie couleur, plus de 16 millions de couleurs).[5]

Si on prend l'exemple du codage des couleurs sur 16 bits, chaque pixel sera présenté par un mot de 16 bits comme indiqué dans la figure I.1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rouge					Vert						bleu				

Figure I.1-Le codage des couleurs sur 16 bits

Chapitre I : généralités sur les technologies des écrans LCD

5 bits sont consacrés à la teinte primaire rouge.

6 bits sont consacrés à la teinte primaire verte.

5 bits sont consacrés à la teinte primaire bleue.

3.3. Ecrans monochromes

Le mot « monochrome » signifie littéralement une seule couleur. Par conséquent une image monochrome ne comporte qu'une seule couleur, mais peut contenir de nombreuses nuances.

Un écran monochrome utilise une couleur pour l'arrière plan et une autre pour le texte d'affichage sur l'écran. [6]

On peut trouver ce type d'écrans dans quelques appareils électroniques, tels que les montres, les minuteriers et horloges numériques.

4. Types de panneaux LCD

Il existe actuellement plusieurs technologies de panneaux LCD. Nous décrivons dans ce qui suit quelques types de panneaux les plus utilisés dans les systèmes embarqués.

4.1. L'écran LCD à matrice active ou passive

Un écran à cristaux liquides est un écran plat utilisant un mode d'affichage numérique où les cristaux liquides sont soumis à un courant électrique fournissent l'image désirée. [7]

Le gros avantage de cette technologie est le contraste élevé, l'augmentation de l'angle de vue et de luminosité. C'est la technologie utilisée actuellement par les écrans standards et les télévisions de petites tailles.

Il existe divers types de panneaux LCD, chacun d'eux offre une performance différente. Un écran LCD est doté d'une grille d'affichage à matrice passive ou active. Ces deux types utilisent deux façons de mettre en ou hors service les pixels de la dalle LCD.

Les écrans à matrice active est la technologie d'écran la plus récente. Ces écrans intègrent des transistors qui accompagnent chaque pixel. Grâce aux transistors, les pixels peuvent s'allumer et s'éteindre très rapidement. La présence de ces transistors élimine aussi tout phénomène d'interférences, qui diminue le contraste des écrans.

Chapitre I : généralités sur les technologies des écrans LCD

La matrice active est de meilleure qualité d'image mais également plus gourmande en électricité.

Les écrans à matrice passive désignent une technologie d'écran plat à cristaux liquides (LCD) fondée sur une grille (ou matrice) de fils conducteurs horizontaux et verticaux. A chaque intersection d'un fil horizontal et vertical se trouve un élément à cristaux liquides qui constitue un pixel. Lorsque ces deux fils sont alimentés, le pixel correspondant s'allume.

Bien que la technologie à matrice passive coûte moins chère, elle a quelques inconvénients, notamment un effet de rémanence (les objets en mouvement à l'écran laissent une " trace ") et une visibilité contestable lorsqu'on ne se trouve pas face à l'écran. [8]

a. Ecran LCD-TFT :

La technologie LCD-TFT (écran à cristaux liquides et film transistor fin) est un écran à matrice active, il permet de contrôler chaque pixel d'une dalle LCD avec un ou plusieurs transistors. Les transistors TFT¹ sont constitués de plusieurs couches minces, ils jouent le rôle d'interrupteur dans le circuit de commande des pixels d'un écran LCD. Ils sont utilisés généralement dans les ordinateurs portables, système de navigation automobile, etc. [9]

Ces écrans LCD-TFT nécessitent une valeur numérique appliquée sur la couleur de chaque pixel. [10]

- 1 bpp, 2 couleurs choisies parmi les couleurs disponibles. -

2 bpp, 4 couleurs choisies parmi les couleurs disponibles. - 4

bpp, 16 couleurs choisies parmi les couleurs disponibles. - 8

bpp, 256 couleurs choisies parmi les couleurs disponibles. -

12 bpp, 4:4:4 RGB.

- 16 bpp, 5:6:5 RGB.

- 24 bpp, 8 : 8 : 8 RGB, plus de 16 millions de couleurs.

¹ Thin-film transistor

Chapitre I : généralités sur les technologies des écrans LCD

Son principal avantage est le coût de production relativement bas, son inconvénient est la consommation excessive d'énergie, et donc une durée de vie réduite de la batterie pour les dalles de grande taille.

b. Ecran LCD-STN

Les écrans à cristaux liquides en technologies STN² est un type à matrice passive monochrome. Les écrans LCD STN nécessitent moins d'énergie et sont moins coûteux à fabriquer que le LCD TFT. L'affichage sur les écrans STN souffre généralement de baisse de qualité et un temps de réponse plus lent que les écrans TFT. Cependant, les écrans LCD STN utilisent des écrans purement réfléchissants pour une meilleure visualisation en plein soleil. Cette technologie d'écran est utilisée dans certains téléphones mobiles, écrans d'information et produits numériques. [11]

L'écran STN supporte un ou plusieurs modes de couleurs : [10]

- 1 bpp, 2 couleurs sont sélectionnés de 3375 niveaux de gris (15x15x15).
- 2 bpp, 4 couleurs sont sélectionnés de 3375 niveaux de gris (15x15x15).
- 4 bpp, 16 couleurs sont sélectionnés de 3375 niveaux de gris (15x15x15).
- 8 bpp, 256 couleurs sont sélectionnés de 3375 niveaux de gris (15x15x15).

c. Ecrans TN

Les écrans TN³ ont la particularité de faire varier la position de leurs cristaux liquides très rapidement, ce qui leur permet d'afficher des mouvements rapides sans effet de flou. Cet écran est la seule technologie capable de descendre sous les 5 ms de temps de réponse. Seul problème, ses angles de visions sont mauvais et son rendu d'image très moyen. [12]

² Super Twisted Nematics

³ Twisted Nematic

d. Ecran PLASMA

La technologie plasma (PDP⁴) est basée sur une émission de lumière grâce à l'excitation d'un gaz qui est utilisé comme un mélange d'argon (90%) et de xénon (10%).

Le gaz est contenu dans des cellules correspondant aux pixels, dans lesquelles sont adressées une électrode ligne et une électrode colonne permettant d'exciter le gaz qui produit un rayonnement lumineux ultraviolet. Grâce à des luminophores respectivement bleus, verts et rouges répartis sur les cellules, le rayonnement lumineux ultraviolet est converti en lumière visible, ce qui permet d'obtenir des pixels (composés de 3 cellules) de 16 millions de couleurs (256 x 256 x 256). [13]

5. Caractéristiques de l'écran LCD

Définition (ou nombre de pixels): Un écran est divisé en de nombreux pixels. La largeur et la hauteur d'un écran en pixels représentent sa définition. Si l'écran est composé de 320 px en largeur et 240 px de hauteur, la définition de l'écran est de 320x 240 px.

Dimension: c'est la diagonale qui est indiquée en pouces (1 pouce= 2,54 cm).

Résolution Ppi (pixel per inch): La résolution est un rapport de densité, elle exprime le nombre de pixels que peut afficher l'écran dans un pouce (2,54 cm). Ne pas confondre avec la définition qui donne le nombre total de pixels. [13]

6. Les contrôleurs LCD

La multitude des technologies d'écran LCD et le nombre important de paramètres nécessaires à son fonctionnement rendent sa gestion par un driver logiciel extrêmement complexe. Heureusement, des dispositifs matériels spécialisés, dits contrôleurs LCD sont utilisés pour faciliter la gestion de ces écrans.

Un contrôleur LCD est un dispositif matériel qui gère et dirige le flux de données entre le processeur et l'écran LCD. Il est réalisé avec un processeur (ou microcontrôleur) spécialisé et peut gérer plusieurs types d'écrans différents. Il dispose d'un certain nombre de registres internes permettant au programme (le driver) d'introduire les différents paramètres

⁴ Plasma Display Panel

Chapitre I : généralités sur les technologies des écrans LCD

correspondant à l'écran utilisé. Il dispose en sortie de tous les signaux nécessaires pour gérer un écran LCD. Les contrôleurs sophistiqués disposent également de mémoire propre permettant d'introduire directement les objets à afficher (image, texte, ...etc.) facilitant ainsi la programmation.

Le contrôleur LCD de la carte LPC 3250 (voire l'annexe pour les détails de la carte) que nous allons utiliser permet de piloter un ensemble de dispositifs d'affichage par cristaux liquide. Développé par HITACHI, il a été commercialisé en 1987 .Il fournit tous les signaux de contrôles nécessaire pour interfacier les écrans couleurs ainsi que les écrans monochromes.

Le contrôleur LCD de la carte LPC 3250 prend en charge les types de panneaux LCD suivants : [14]

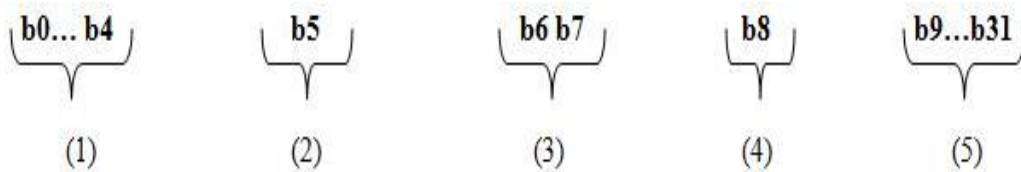
- Les panneaux LCD-TFT avec une matrice active (une interface de bus de 24 bits).
- Les écrans STN monochrome avec une interface de 4 bits et 8 bits.
- Un seul panneau monochrome STN (une interface de 4 bit et 8 bit).
- Double panneau monochrome STN (interface de bus de 4-bit et 8-bit par panneau)
- Un seul panneau couleur STN, interface de bus 8 bits.
- Double- panneau couleur STN, interface de bus 8 bits par panneau.
- 15 niveaux de gris monochrome, 3375 couleurs pour la dalle STN.
- Résolution d'affichage, y comprise, mais non limité : 320×200 , 320×240 , 640×200 , 640×240 , 640×480 , 800×600 et 1024×768 .

6.1. Paramétrage d'un contrôleur LCD

Pour utiliser correctement un écran LCD, il est nécessaire d'introduire un ensemble de paramètres de contrôle et d'affichage correspondants. Nous citons dans ce qui suit quelques paramètres et registres.

Chapitre I : généralités sur les technologies des écrans LCD

LCD Configuration register (LCD_CFG, RW - 0x4000 4054) C'est le registre de configuration de l'écran à gérer. C'est un registre de 32 bits mappé à l'adresse mémoire 0x4000 4054. Il contrôle la sélection des broches de sortie nécessaires pour différentes configurations de l'écran LCD, ainsi que le pré-échelonnage de l'horloge utilisée pour la génération de données LCD.



- (1) :CLKDIV sélection du pré-diviseur de l'horloge.
- (2) : HCLK_ENABLE activer l'horloge HCLK.
- (3) :MODE_SELECT sélectionner le mode de couleur.
- (4) :DISPLAY_TYPE sélectionner le type d'affichage.
- (5) : RESERVED, bits non utilisés.

A titre d'exemple, pour configurer un écran de type LCD-TFT qui prend en charge le mode de couleur 16 bits par pixel (RGB : 565) il faut :

- sélectionner le mode (5 :6 :5) ce qui correspond à la valeur 01 sur la partie (3) (bits 6 et 7).
- Sélectionnez type LCD-TFT en mettant la partie (4) à 0 (bit 8).

Pour cet exemple, il faut enregistrer donc par programme (par le driver de l'écran exactement) la valeur 0x60 (la valeur en hexadécimal) en mémoire à l'adresse 0x4000 4054 correspondant à l'adresse du registre LCD_CFG. Toutes les valeurs à utiliser doivent être tirées du manuel d'utilisation de la carte LPC 3250. [6]

LCD Control register (LCD_CTRL, RW - 0x3104 0018): contrôle le mode de fonctionnement du LCD et les paramètres des panneaux.

-Le bit 7 du registre LCD_CTRL correspond au paramètre LCD Dual qui permet de sélectionner le type du panneau, s'il s'agit d'un panneau simple ou double.

Chapitre I : généralités sur les technologies des écrans LCD

-le bit 5 du registre LCD_CTRL correspond au paramètre LcdTFT qui permet de sélectionner le type du panneau LCD s'il s'agit d'un panneau STN utilisant des niveaux de gris ou TFT n'utilisant pas de niveaux de gris.

Par exemple pour un écran TFT simple on doit choisir les valeurs suivantes :

- Le bit 7 est mis à 0.
- Le bit 5 est mis à 1.

Ce qui correspond à une valeur 0x0000012C dans le registre LCD_CTRL se trouvant à l'adresse 0x3104 0018. [10]

7. Le Frame buffer

Un frame buffer (en français tampon de trame ou mémoire d'image) commande un affichage à partir d'un tampon mémoire contenant une trame complète de données. Les informations dans le tampon mémoire sont en général les valeurs des composantes de couleurs pour chaque pixel. La quantité de mémoire nécessaire à la commande d'un frame buffer dépend de la résolution.

L'écran est un périphérique analogique et le frame buffer est un dispositif numérique. Chaque pixel dans la mémoire tampon de trame doit être consulté et converti par le contrôleur avant qu'il ne soit visible sur l'écran. [15]

Exemple de l'affichage d'un caractère :

On prend l'exemple simple d'un écran monochrome de dimension 9x7 pour illustrer comment se fait la représentation d'un caractère dans le frame buffer et son affichage sur l'écran.

Pour afficher un caractère (par exemple un T) on utilise le codage suivant : 0 pour un pixel éteint, 1 pour un pixel allumé comme le montre la figure I.2.

On crée ensuite une image mémoire du caractère qu'on envoie dans le frame buffer.

Si on suppose que l'image mémoire du caractère se trouve à l'adresse *Image_T et que le frame buffer se trouve à l'adresse *FB, l'affichage du caractère peut se faire de la manière suivante :

Chapitre I : généralités sur les technologies des écrans LCD

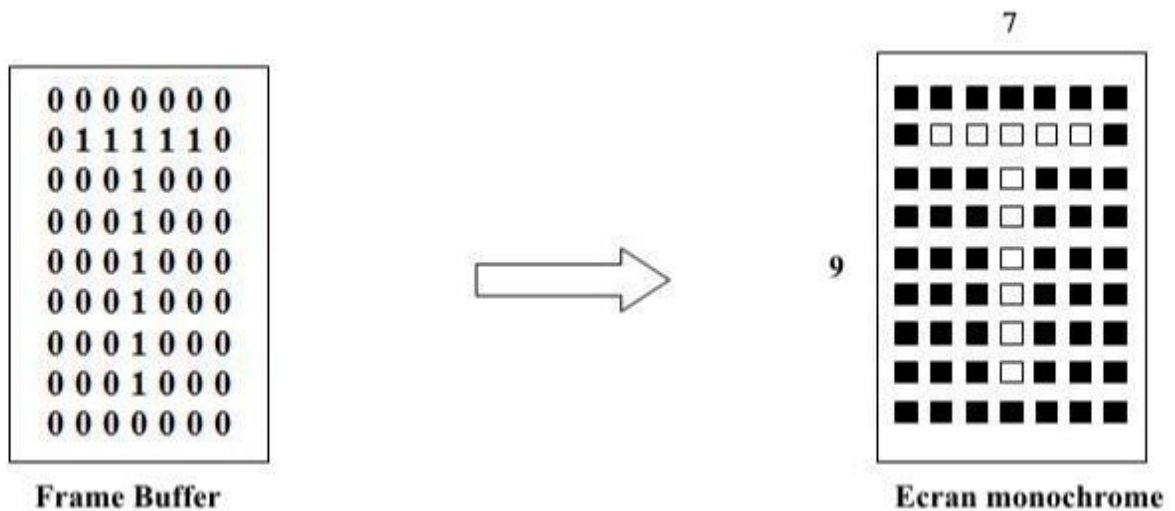


Figure I.2-représentation d'un caractère dans le frame buffer et son affichage sur l'écran

Voici un programme simple qui permet d'afficher le caractère T :

debut

```
byte IMG_T[9]={0X00, 0X3E, 0X08, 0X08, 0X08, 0X08, 0X08, 0X08, 0X00} ; int i ;
```

```
for (i=0 ; i < 9 ; i++){
```

```
    *FB = IMG_T[i];
```

```
    *FB++;
```

```
}
```

fin

Pour des situations plus complexes, pour prendre en charge les couleurs par exemple, chaque pixel sera défini, non pas par un bit, mais par un mot de 8, 16 ou 24 bits contenant chacun les codes de couleur RVB associé.

Chapitre I : généralités sur les technologies des écrans LCD

Conclusion

Dans ce chapitre nous avons décrit les technologies d'affichage graphique les plus utilisées actuellement dans les systèmes embarqués. Nous avons constaté qu'il existe de nombreux types d'écrans, chaque type utilise une technologie différente avec des caractéristiques variées en termes de qualité d'affichage, de durée et de consommation d'énergie. Cependant, on s'est focalisé sur la technologie LCD qui s'est répandue grâce à ses caractéristiques répondant aux exigences des utilisateurs ; faible consommation d'électricité et visualisation très agréable.

Nous avons également introduit les principes de gestion des écrans LCD notamment l'utilisation des contrôleurs et des Frames buffer qui sont à la base des composants manipulés par les drivers logiciels.

Dans le chapitre qui suit nous allons nous consacrer aux technologies tactiles en présentant le principe général de leur fonctionnement et illustrer la relation existant entre les écrans LCD et ces technologies tactiles.

Chapitre II :

Généralités sur les technologies tactiles

1. Introduction

Les technologies tactiles sont de plus en plus présentes dans le monde d'aujourd'hui, ces technologies étaient peu répandues et seulement disponibles sur des bornes extérieures telles que les stations services, les gares ou aéroports.

Ces dernières années, de nouvelles technologies ont souvent eu recours à des écrans et des dispositifs tactiles. La sortie de la console DS de Nintendo ou du téléphone iPhone d'Apple sont des exemples de cette expansion. Le coût abordable et la résistance à l'utilisation régulière de l'écran tactile sont les principales causes du développement de cette technologie.

Cet élan impacte indirectement le monde de l'informatique et des interfaces homme / machine. En effet, les écrans tactiles apportent un nombre infini de nouvelles possibilités d'interactions entre un utilisateur et un matériel. Les interfaces graphiques des dispositifs tactiles évoluent donc pour offrir aux utilisateurs l'occasion d'exploiter ces nouveautés. C'est ainsi que les grandes interfaces graphiques populaires, comme Windows, sont vouées à disparaître en même temps que le tactile se développe.

C'est ce constat qui nous amène à nous interroger sur les technologies tactiles et l'avenir qu'elles nous réservent. En tant qu'informaticiens, nous sommes tous confrontés à ce nouveau type de technologie, et donc à de nouvelles méthodes de développement d'applications. [16]

Comme pour le LCD, l'objectif de notre travail inclut la création d'un gestionnaire de l'interface tactile de la carte LPC 3250. Pour cela, dans un premier temps, nous allons définir la technologie du tactile ainsi que le contrôleur tactile de la LPC 3250 et son utilisation.

2. Ecran tactile

2.1. Définition :

L'écran tactile est un écran de visualisation réactif au toucher .Il cumule les fonctions d'affichage d'un écran (pour l'affichage) et d'une souris (pour le pointage). Ce dispositif d'entrée reconnaît un ou plusieurs touchers simultanés et permet aux utilisateurs d'interagir avec les applications soit à travers divers gestes créés par les doigts ou bien avec un stylet optique sur une surface. [17]

Les écrans tactiles sont utilisés, par exemple, pour les PDA⁵, les GPS⁶, les Smartphones, les tablettes, les consoles portables, les billetteries automatiques, les ordinateurs...etc.

2.2. Principe de fonctionnement

Pour faciliter la compréhension, nous présentons tout d'abord le principe général abstrait du fonctionnement d'une interface tactile. Elle peut être réalisée par deux films (en plastique ou en verre) collés sur la dalle d'un LCD. Les deux films correspondent à des résistances dépendantes de la pression exercées par le doigt ou le stylet auxquels on applique une tension de référence. Quand on exerce une pression sur l'interface, chaque film retourne une tension proportionnelle à la position (en x et en y). Cette tension est ensuite lue par un convertisseur analogique numérique et convertie en coordonnées en fonction de la dimension de l'écran LCD. La figure II.1 illustre ce principe.

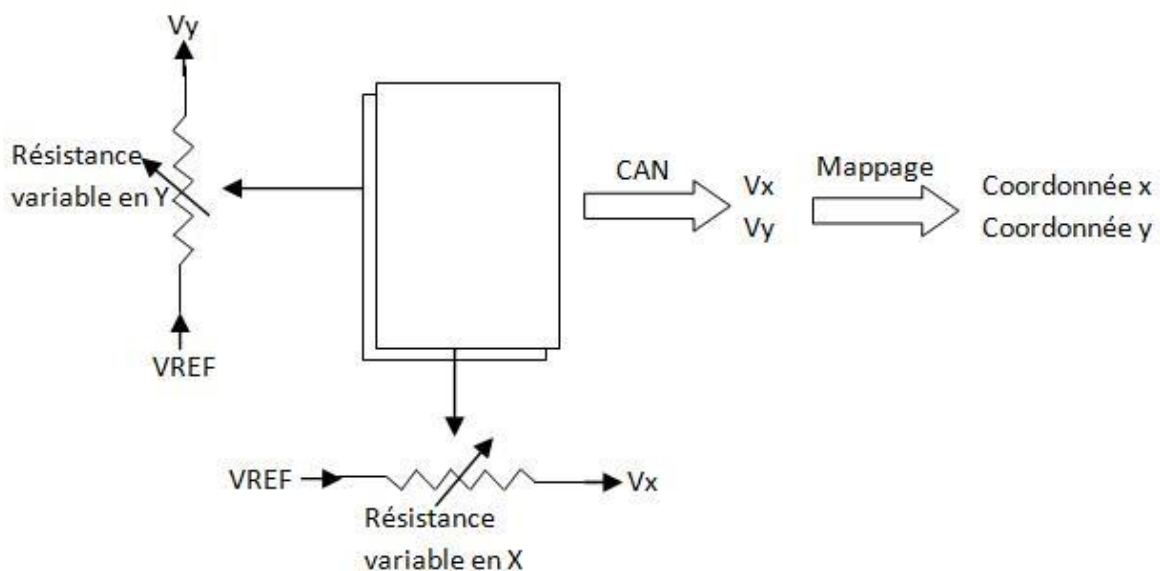


Figure II.1-Principe de fonctionnement des interfaces tactiles

⁵ Personal Digital Assistant(en français un assistant personnel.)

⁶ Global Positioning System (en français Système mondial de positionnement)

Pour gérer une interface tactile, le driver doit donc réaliser les tâches suivantes:

1. Initialiser le convertisseur analogique numérique avec la tension de référence V_{REF} ⁷ appliquée à l'interface.
2. Lire les tensions des résistances en X et en Y et les convertir en numérique.
3. Mapper les tensions en coordonnées en fonction de la dimension de l'écran.

Ce procédé est toutefois basique et différentes technologies tactiles existent maintenant mais le principe reste le même. Nous citons dans ce qui suit quelques technologies d'écrans tactiles existants mais avant, nous donnerons quelques explications sur les termes rencontrés souvent dans ce domaine.

2.3. Terminologie

Single-touch ou multi-touch

Un dispositif single-touch n'est capable d'interpréter qu'un seul et unique point de contact à la fois, et donc par extension, il n'accepte qu'un seul utilisateur.

La technologie multi-touch, permet d'analyser plusieurs points de contacts en même temps sur le même appareil. Ceci offre la possibilité de faire des gestes complexes, et également d'interagir simultanément avec plusieurs utilisateurs. [16]

Multipoints ou multi-touch

Ces deux termes sont souvent confondus, mais ils ne reflètent pas exactement la même notion.

Un système multipoints est un système qui interprète plusieurs points de contacts (coordonnée en x et y). Il s'agit d'un événement équivalent à plusieurs clics de souris simultanés. Cela permet l'introduction de gestes limités.

La technologie multi-touch offre des gestes plus riches. Elle ne se contente pas de simples coordonnées en abscisse et ordonnée, mais elle interprète des zones de contacts (également appelées régions). [16]

⁷ Voltage référence

2.4. Technologies des interfaces tactiles :

2.4.1. Technologie capacitive

La technologie capacitive est la plus ancienne de toutes. C'est aussi la plus répandue. Son fonctionnement est très simple.

Elle est composée d'une plaque de verre sous laquelle se trouve une grille électriquement chargée.

Lorsque l'utilisateur touche la plaque de verre (la surface du doigt étant conductrice), les charges électriques sont attirées vers le doigt ce qui crée un vide au point de contact dans le champ. Ce vide est calculé grâce aux différentes valeurs obtenues par les quatre capteurs disposés à chaque coin de la plaque.

L'inconvénient de cette technologie, en plus du prix, est la nécessité d'utiliser un élément conducteur, un doigt ganté par exemple n'aura aucun effet sur l'écran et sa sensibilité à l'humidité. Elle permet par contre d'avoir un écran généralement plus lumineux par rapport aux autres technologies, une meilleure réactivité et une utilisation multi-touch. Ce type d'écran est le plus utilisé de nos jours sur les tablettes et les téléphones portables. [18]

Les meilleurs exemples de cette technologie sont : iPhone (apple), touch pad etc. [3]

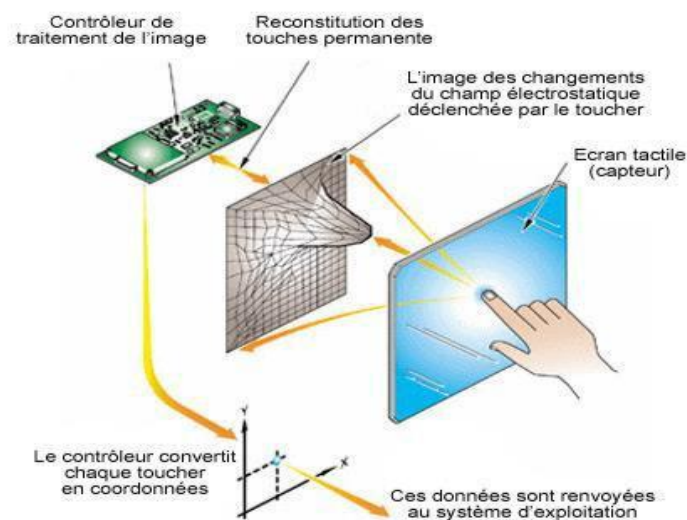


Figure II.2-Technologie capacitive

2.4.2. Technologie résistive

La technologie résistive est la seconde technologie la plus répandue. Le dispositif est composé de deux couches conductrices séparées par une couche d'isolant. La couche supérieure, du côté de l'utilisateur, est flexible.

Lorsque l'utilisateur exerce une pression sur un point précis de l'écran, la surface en plastique souple entre en contact avec l'écran en verre, ce qui entraîne une variation dans les champs électriques des deux faces, qui sera retransmis en information et traitée par l'algorithme intégré dans le contrôleur de l'interface. Une fois cette information traitée, la partie commande de l'appareil va déterminer les coordonnées du point de contact.

L'avantage de cette technologie est qu'elle fonctionne avec tout type de pointeur tout en étant moins cher que la capacitive et offre une meilleure précision (stylet est plus précis qu'un doigt). Cependant, elle a tendance à s'user plus rapidement du fait des décharges électriques et nécessite des rééquilibrages pour gommer ce défaut. [19]

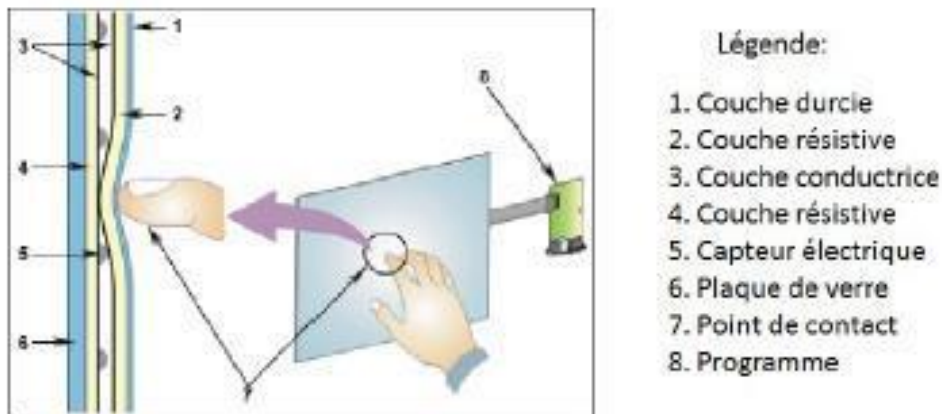


Figure II.3-Technologie résistive

2.4.3. Technologie infrarouge

La technologie infrarouge consiste en un quadrillage infrarouge, réalisé à l'aide de DEL⁸ (LED en anglais), posé sur l'écran. Quand on touche l'écran, des faisceaux infrarouges sont coupés, ce qui permet de connaître les coordonnées du point de contact. Ces écrans gardent une bonne luminosité et une bonne durée de vie. Cependant, la poussière peut interférer dans les signaux infrarouges, les exemples d'application de cette technologie sont : TouchSmart(HP) et AlbatronTouch. [19]

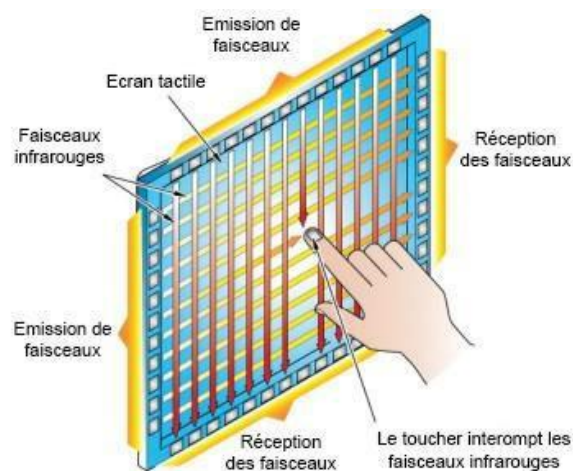


Figure I.4-Technologie à infrarouge

2.4.4. Technologie à ondes de surface

Cette technologie possède le même fonctionnement que la technologie à infrarouges, sauf qu'elle utilise des ondes ultrasoniques.

Des ondes ultrasoniques inaudibles parcourent la surface de l'écran et créent un quadrillage. Chaque onde est émise par un émetteur à une extrémité de l'écran et reçue par un récepteur à l'autre extrémité. Lorsque l'utilisateur touche l'écran, il vient perturber la propagation des ondes.

Les écrans tactiles à onde de surface ont plutôt une bonne luminosité, cependant l'inconvénient de cette technologie est que la minime rayure ou même une poussière sur l'écran perturbe la propagation des ondes, Un exemple de cette technologie : iTouch «Touch-on-Tube », les bornes publiques...etc. [19]

⁸ Diode électroluminescente (LED en anglais *Light-Emitting Diode*)

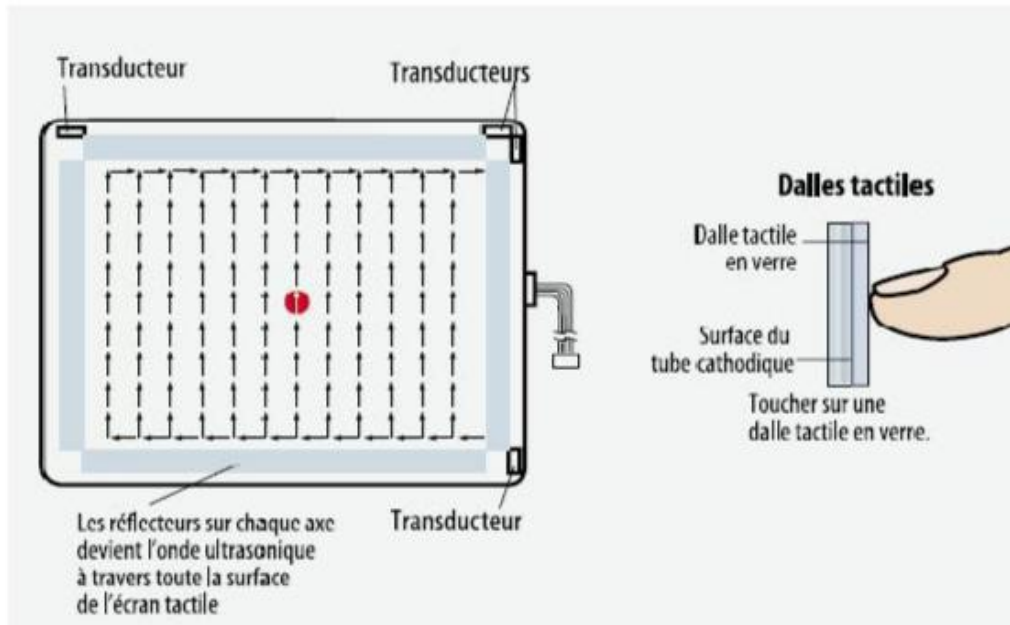


Figure II.5 -Technologie à onde de surface

2.4.5. Technologie optique

C'est une récente technologie dans laquelle sont placées deux cameras au bord de l'écran. Chaque camera est équipée d'une diode infrarouge et l'écran est doté de retro-rélecteurs. La diode infrarouge émet de la lumière, qui est réfléchiée par les retro-rélecteurs. Une ombre due au doigt est alors formée sur l'écran, et cette ombre est captée par les cameras qui peuvent définir la position de cette pression.

Cette technologie est beaucoup utilisée aujourd'hui. De plus, elle s'adapte très bien aux écrans haut format. [20]

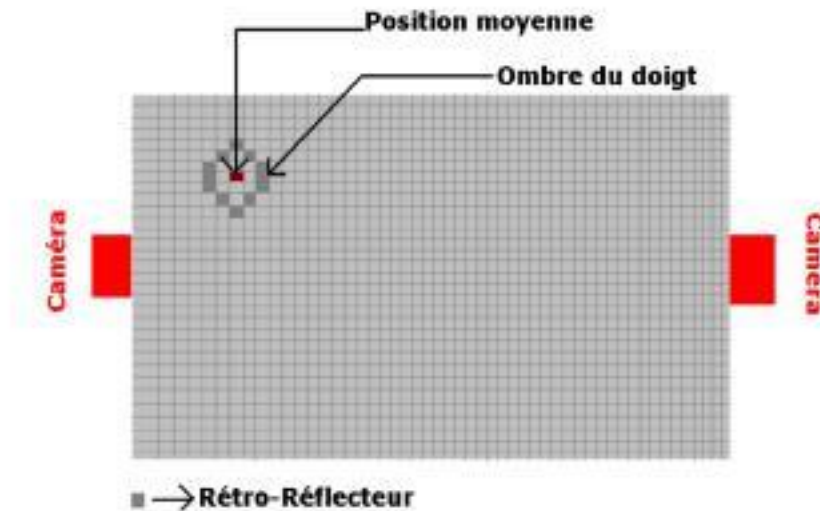


Figure II.6-Technologie optique

2.5. Contrôleur de l'interface tactile

Comme pour l'afficheur LCD, les interfaces tactiles modernes sont complexes à gérer et sont conçus en divers technologies. Les contrôleurs d'interfaces tactiles (ou TSC) sont des dispositifs matériels facilitant la gestion d'interfaces tactiles. Le TSC de la carte LPC 3250 est un exemple de ces contrôleurs. [21]

Le contrôleur de l'interface tactile prend en charge tous les signaux nécessaires à la connexion d'une interface tactile, dispose en général, d'un convertisseur analogique numérique pour prendre des mesures et effectuer toutes les opérations nécessaires pour restituer au système les coordonnées utilisables directement ou qui doivent être mappés en fonction de l'écran (comme c'est le cas pour le TSC de la LPC 3250). Nous présentons dans ce qui suit le contrôleur TSC de la PLC 3250, ainsi qu'un aperçu sur son mode d'exploitation.

2.5.1. Fonctionnement du contrôleur TSC de la LPC 3250

Le dispositif tactile est un dispositif résistif à 4 fils qui est constitué de deux films souples de polyester dont la surface est conductrice recouverte d'ITO⁹, L'un des films est placé comme axe des X et l'autre comme axe des Y.

Ces deux surfaces sont isolées par des petites billes isolantes afin d'éviter de fausser les courants électriques passant entre les deux plaques.

⁹ Indium tin oxide

Chapitre II : Généralités sur les technologies tactiles

Le contrôleur applique la tension de +5v sur l'axe des X à gauche de l'écran puis la tension diminue uniformément jusqu'à atteindre la tension nulle vers la droite de l'axe X, le même processus pour l'axe Y, une tension de +5v est appliquée vers le haut, et elle diminue jusqu'à atteindre la tension nulle vers le bas de l'axe.

Lorsque l'écran n'est pas touché, la tension est nulle, dans le cas contraire les deux films se rapprochent juste assez pour établir un contact électrique entre eux, le contrôleur détectera la valeur des tensions des résistances des axes X et Y puis il fait appel au convertisseur analogique-numérique codé sur 10 bits pour convertir le signal analogique en une valeur numérique. [22]

Vient ensuite l'étape de calibration qui est utilisée pour traduire les données de la dalle tactile mesurées en véritables coordonnées de l'écran LCD en prenant en considération le décalage et de la rotation de l'écran tactile. Ce mode de fonctionnement est illustré dans la figure II.7.

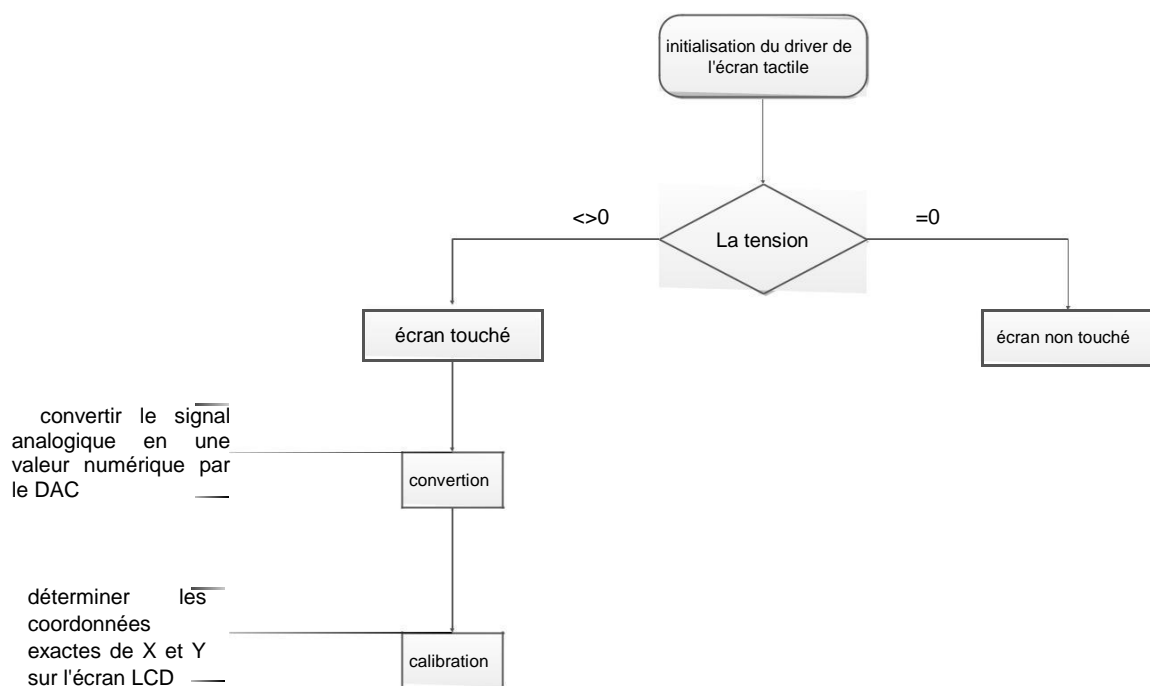


Figure II.7-Fonctionnement du contrôleur tactile TSC

Chapitre II : Généralités sur les technologies tactiles

Pour déterminer si l'écran est touché, le contrôleur TSC fonctionne en deux modes, le mode automatique et le mode manuel.

Lorsque le mode automatique est activé, le contrôleur gère automatiquement la séquence de mesure pour récupérer les coordonnées X et Y de la position pressée et les stocke dans une file d'attente TSC_SAMPLE_FIFO. Le but principal de l'automatisation est de réduire le nombre d'interruptions nécessaires pour déterminer la position du touché.

Tandis que dans le mode manuel (inactif), le TSC peut être séquencé entièrement par logiciel pour collecter les coordonnées de la position pressé. [10]

2.5.2. Paramétrage du contrôleur TSC

Pour utiliser correctement l'écran tactile, il est nécessaire d'introduire un ensemble de paramètres de contrôle. Nous citons dans ce qui suit quelques paramètres et registres.

ADC Clock Control register (ADCLK_CTRL - 0x4000 40B4):

Ce registre du contrôleur implémenté à l'adresse 0x4000 40B4 permet d'activer ou de désactiver l'horloge de l'ADC. L'horloge de l'ADC étant désactivée par défaut pour l'économie d'énergie.

Bien que le registre soit de 32 bits, un seul bit est utilisé pour le TSC qui est le bit 0. Pour l'activer on doit donc le mettre à 1, ce qui est réalisé par l'initialisation du registre ADCLK_CTRL à 0x1.

Touchscreen controller sample FIFO register (TSC_SAMPLE_FIFO - 0x4004800C):

Ce registre permet d'accéder aux valeurs des coordonnées récoltés par le contrôleur. Il est accessible uniquement en lecture, il contient à la fois les valeurs X et Y et si l'écran est pressé ou non pressé. Le registre est structuré de la manière suivante:

Les bits **0 à 9** : **TS_Y_VALUE**, représentent la valeur ADC de la coordonnée Y.

Les bits **10 à 15** : **RESERVED**. Non utilisés .

Les bits **16 à 25** : **TS_X_VALUE**, représentent la valeur ADC de la coordonnée X.

Les bits **26 à 29** : **RESERVED**. Non utilisés .

Le bit **29** : **FIFO_OVERRUN** indique si la FIFO est pleine : si elle est pleine le bit est mis à 1 sinon à 0.

Chapitre II : Généralités sur les technologies tactiles

Le bit **30** : **FIFO_EMPTY** indique si La FIFO est vide : si elle est vide le bit est mis à 1 sinon à 0.

Le bit **31** : **TSC_P_LEVEL** indique si l'écran est touché ou non : si il n'est pas pressé alors le bit est mis à 1 sinon à 0.

Si la FIFO est pleine et que, de nouveaux échantillons sont ignorés par manque de place, Cet événement donnera le bit dépassement 8 dans le registre ADC_STAT. **[10]**

Chapitre II : Généralités sur les technologies tactiles

Conclusion

Dans ce chapitre nous avons vu des généralités sur la technologie tactile incluant ses différents types, les composants d'un dispositif tactile qui s'est imposé dans notre vie quotidienne grâce à sa fiabilité et son utilisation facile.

Nous avons montré en particulier comment utiliser un contrôleur TSC (celui de la LPC-3250) pour gérer une interface tactile.

Dans le chapitre qui suit, nous allons nous concentrer sur la description de notre approche adoptée pour l'analyse et la conception de notre projet. On décrira d'abord les objectifs de notre système puis on décrira ses différents composants illustrés par des figures et diagrammes.

Chapitre III :

Analyse et conception

1. Introduction

Comme nous l'avons vu dans les chapitres précédents, les interfaces tactiles ne cessent de prendre de l'ampleur dans le marché de l'informatique. Avec cette progression le monde du tactile occupe désormais une place très importante dans notre vie quotidienne ; nos maisons, places de travail et places publiques sont toutes dotées de technologie tactile. La plupart des utilisateurs estiment que cette technologie est utile et favorable ; ce qui pousse les développeurs à initier de nouvelles vues et idées en utilisant le moyen du toucher.

Dans ce chapitre nous allons exposer le problème et faire une analyse des différents composants qui constituent notre interface tactile. Nous entamerons la partie conception logicielle afin de modéliser le système de manière simplifiée et abstraite, enfin nous terminerons par exposer quelques algorithmes.

2. Analyse et conception

2.1. Analyse

La phase d'analyse permet de lister les objectifs et étudier le système c.à.d. fournir une compréhension des exigences, des concepts et du comportement de l'application.

2.1.1. Idée générale sur notre travaille

Notre objectif est de concevoir un générateur d'interface graphique tactile pour la carte LPC 3250 qui doit être intégré dans n'importe quel projet de développement embarqué pour cette dernière. Le développeur du système ne va pas se soucier de la programmation de l'interface adéquate. Il va juste écrire un fichier XML¹⁰ contenant les composants dont il a besoin : boutons, zones de textes ... etc. Ce fichier va être traduit ensuite par le générateur en un programme nécessaire à l'affichage de cette interface graphique tactile incluant les routines de gestion des contrôleurs LCD et tactile. La figure III.1 illustre le fonctionnement général de notre application.

¹⁰

Extensible Markup Language (en français langage de balisage extensible)

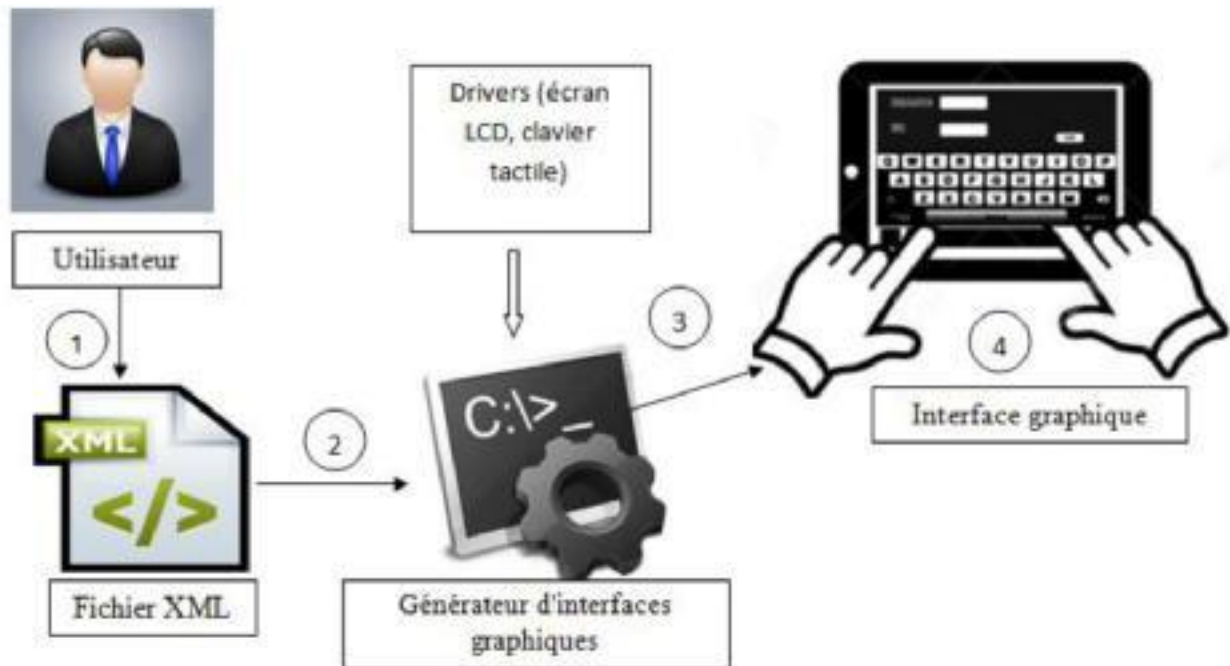


Figure III.1-Fonctionnement de notre système

1 : l'utilisateur écrit son fichier XML contenant les composants selon ses besoins : boutons, zones de textes... etc.

2 : extraire du fichier XML les informations nécessaires et les envoyer au générateur.

3 : Le générateur traduit le fichier XML en une interface graphique tactile en ajoutant les drivers nécessaires.

4 : L'affichage de l'interface sur l'écran.

Afin de mieux cerner ce travail il est recommandé de définir les composants nécessaires à une interface graphique tactile qui est composé de trois différentes couches : couche d'affichage des interfaces graphiques, dispositif de pointage et la couche logiciel.

Interface graphique :

Les interfaces graphiques sont souvent appelées GUI (Graphical User Interface). L'interface graphique d'une application constitue la couche externe de cette dernière, avec laquelle l'utilisateur interagit et communique. [23]

Par ailleurs l'interface graphique utilisateur est une plateforme de contrôleur LCD compact non seulement difficiles à concevoir (du fait de la multiplicité des aspects humains et logiciels qu'il faut prendre en compte) mais aussi à implémenter.

La majorité des interfaces graphiques sont composées de divers composants tels que (boutons, champs et zones de saisies... etc.).

- Fenêtre :

Objet d'une interface graphique, se présentant sous la forme d'un rectangle sur l'écran, contenant des informations (texte, graphique... etc.). L'origine du terme « fenêtre » vient de la traduction littérale du mot anglais Windows. Les fenêtres d'un écran sont organisées en arbre: la fenêtre racine occupe l'écran; chaque application a une fenêtre principale et des sous-fenêtres. [24]

Plusieurs fenêtres peuvent être affichées et gérer simultanément, on parle alors de multifenêtrage. Toutefois, ce genre d'affichage est souvent évité dans les systèmes embarqués du fait des contraintes de dimensions des écrans.

Par ailleurs on peut saisir des données et exécuter des tâches à l'aide de boutons, clavier et champs de texte.

- Zone de texte :

Graphiquement une zone de texte ressemble à un rectangle de taille quelconque à l'intérieur duquel se trouve le texte saisi par l'utilisateur. La zone de texte est le contrôle standard utilisé pour afficher et modifier des données dans des formulaires et des états. [25]

Le contenu du champ de saisie peut être rempli et modifié par l'utilisateur en utilisant un clavier alphanumérique.

- Bouton :

Un bouton est un dessin : rond, carré, rectangle, qui symbolise une zone de commande d'un programme. [26]

Le dispositif de pointage :

Dans notre cas nous avons utilisé un dispositif de pointage avec une technologie résistive, lorsqu'un événement externe arrive correspondant au toucher de l'utilisateur une interruption se déclenche.

Une interruption est un mécanisme qui permet d'interrompre l'exécution d'un processus suite à un événement extérieur ou intérieur et de passer le contrôle à une routine dite **routine d'interruption** ou **traitement d'interruption**. [27]

Dans notre programme nous avons opté pour la programmation en mode interruption car l'interruption évite au processeur d'être en permanence en scrutation sur une entrée et donc la perte de temps inutile qui n'est pas toléré dans des applications temps réel.

Lorsque l'utilisateur appuie sur un objet (zone de texte, clavier ou bouton) le gestionnaire d'interruption déterminera la position du toucher (retourner les coordonnées x et y), la couche logicielle reçoit ces informations et prend en charge leur interprétation et leur traitement pour ensuite envoyer le résultat à la couche d'affichage afin d'identifier l'objet sur lequel l'utilisateur a appuyé. Une comparaison de la position retournée avec les objets existant est effectuée.

2.2. Conception

Dans cette partie du travail nous allons nous focaliser sur les composants logiciels permettant de réaliser ce projet et définir le fonctionnement du système afin de réaliser les différentes tâches vues lors de l'analyse.

2.2.1. Comportement de l'application

La figure III.2 montre le comportement de notre générateur :

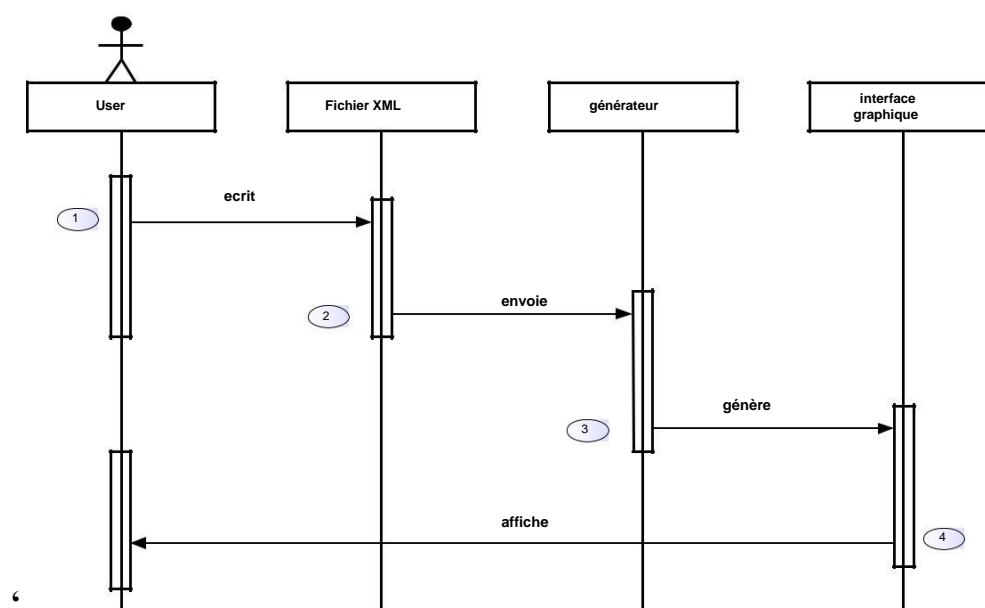


Figure III.2-Le diagramme de séquence des interactions entre l'utilisateur et le générateur

1 : L'utilisateur écrit un fichier XML contenant les composants selon son besoin : boutons, zones de textes... etc.

2 : extraire du fichier XML les informations nécessaires et les envoyer au générateur.

3 : Le fichier XML va être traduit par le générateur en une interface graphique

tactile. 4 : l'affichage de l'interface graphique tactile adéquate.

Diagramme de séquence 2:

La figure III.3 montre les interactions entre l'utilisateur et un objet de l'interface tactile (exemple d'une zone de texte) :

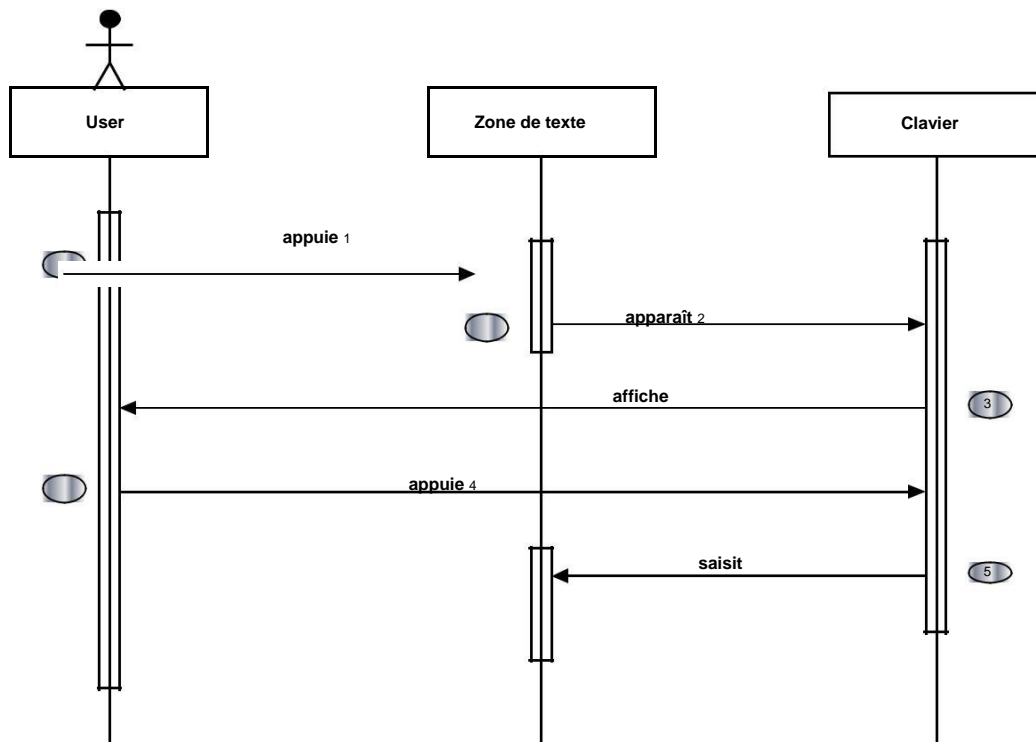


Figure III.3-Le diagramme de séquence des interactions entre l'utilisateur et la zone de texte

1 : L'utilisateur appuie sur une des zones de textes.

2 : En appuyant sur la zone de texte un clavier virtuel apparaît pour l'utilisateur.

3 : Un clavier est affiché pour l'utilisateur.

4 : L'utilisateur utilise le clavier pour remplir le formulaire, il peut (écrire un caractère, supprimer un caractère, espacer entre les caractères).

5 : La saisie sur la zone de texte.

Diagramme de classe

Afin de mieux cerner le fonctionnement de notre interface graphique, nous allons faire une brève interprétation des interactions entre les différents éléments qui composent cette interface. Comme l'illustre la figure III.4.

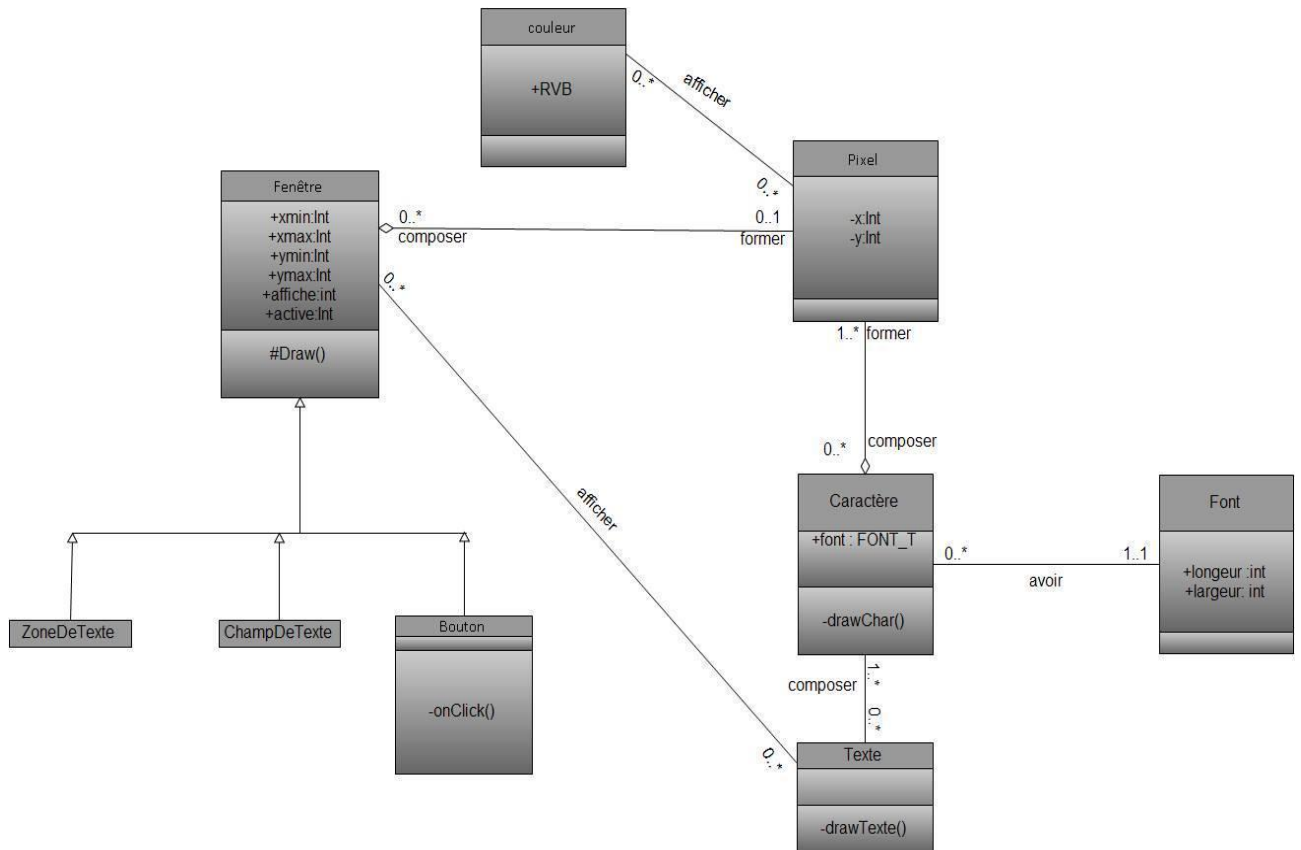


Figure III.4-diagramme de classe de l'interface graphique

-Chaque pixel est déterminé par ses coordonnées X et Y, ce pixel fait appel à la classe couleur pour afficher sa propre couleur.

-Une fenêtre est composée de plusieurs pixels, chaque fenêtre a ses propres attributs ainsi qu'une méthode draw() qui permet de la dessiner.

-La fenêtre fait appel à la classe Texte qui contient la méthode drawText() permettant d'afficher le texte.

-La classe texte fait appel à la classe caractère contenant la méthode drawChar() pour l'affichage d'un caractère.

- Un caractère est composé de plusieurs pixels.
- Les boutons, zone et champ de textes héritent de la classe fenêtre.

Déploiement de notre application

1) Générateur :

Main.c : génère le fichier de configuration pour l'interface en spécifiant le fichier d'entrée <fichier XML> et de sortie <fichier header >.

XML.h : contient les déclarations des structures, et prototypes des fonctions du fichier.

XML.c : permet de parser le fichier XML.

2) Interface graphique tactile

USELCD.c :

Contient la classe principale de notre application **main ()** et qui fait appel au fichier LCD.h, LPC3250.h, colors.h et lpc_x6x13.h,

- Elle contient la déclaration des objets.
- Le traitement a effectué sur les objets.
- Contient la routine d'interruption.

Inclure le header généré par le parseur.

lpc_x6x13.h:

Cette classe contient la définition des données permettant d'afficher tous les caractères sur notre écran.

Color.h :

Cette classe contient une déclaration de couleurs sur 16-bit (RVB 5 6

5). 5 bits sont consacrés à la teinte primaire rouge.

6 bits sont consacrés à la teinte primaire verte.

5 bits sont consacrés à la teinte primaire bleue.

LCD.h :

Cette classe contient des déclarations de constantes, registres et prototypes des fonctions.

LCD.c :

Cette classe fait appel aux fichiers LCD.h, LPC3250.h et colors.h, contient :

- La déclaration des paramètres d'affichage (nombre de pixels par ligne, type du panneau LCD...).

- La Configuration du contrôleur LCD, contrôleur d'écran tactile et l'initialisation de l'interruption.
- Les fonctions permettant l'affichage d'une fenêtre, d'un caractère, d'un texte...

LPC3250.h

Configuration de la carte LPC3250 des registres du contrôleur d'interruption, GPIOs, contrôleur d'interface clavier, convertisseur A/D, contrôleur LCD ...etc.

2.2.2. Aperçu de quelques procédures

2.2.2.1. Déclaration d'une fenêtre

La déclaration d'une fenêtre se fait comme suit :

```
//déclaration de la fenêtre win de type WINDOWS_T
WINDOW_T win[50];
//déclaration des coordonnées en hexa selon l'axe x
win[0].xmin = 0;
win[0].xmax = 0xF0;
//déclaration des coordonnées en hexa selon l'axe y
win[0].ymin = 0;
win[0].ymax = 0x140;
//déclaration de la couleur de la fenêtre
win[0].fill = BLACK;
/* la variable affiche permet de manipuler l'affichage
   de la fenêtre sur l'écran
   si affiche=0 la fenêtre ne s'affichera pas
   si affiche=1 la fenêtre s'affichera */
win[0].affiche = 0;
```

2.2.2.2. Création de caractères

Dans notre cas on a défini la liste des caractères dans un fichier header.

Le dessin d'un caractère est défini sur 13 lignes. Chaque ligne comprend 16 chiffres binaires.

Première étape :

Pour dessiner un caractère on utilise le codage suivant : 0 pour la couleur d'arrière plan éteint, 1 pour la couleur d'avant plan.

Numéro de ligne :	Représentation binaire	dessin correspondant
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
4	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
5	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
6	0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	* * *
7	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	*
8	0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	* * * *
9	1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	* *
10	1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	* *
11	0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	* * * *
12	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
13	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Deuxième étape :

Conversion des treize lignes du binaire en hexadécimal comme suit :

{0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x0800, 0x7800, 0x8800, 0x8800, 0x7800, 0x0000, 0x0000}

2.2.2.3. Ecriture d'un caractère

Après avoir déclaré la liste des caractères dans le fichier header, pour l'afficher sur l'écran on fait appel à la fonction **drawChar()** se trouvant dans un fichier.c.

Drawchar prend comme argument :

win de type WINDOW_T : la fenêtre ou s'affichera le caractère saisi.

C de type char : le caractère à écrire.

X et y de type int : les coordonnées où s'affichera le caractère.

```
void drawChar(WINDOW_T win, char c, int x, int y){
```

```

int i, j;

int charindex;

unsigned short *field, data;

COLOR_T *position;

charindex = (int)c;

field = win.font->font_table +(charindex*win.font-
>font_height); for (i=0; i< win.font->font_height; i++){

    position = (COLOR_T*)(int)(fb)+x+y*ligne;

    data = field[i];

    for(j=0;j<6; j++){

        if ((data&0x8000) !=0){

            *position=win.pen;

        }

        data=data<<1;

        position++;

    }

    y ++; }

}

```

2.2.2.4. La routine d'interruption

La routine d'interruption vérifie d'abord si une interruption a eu lieu en consultant la valeur du 6^{ème} bit du registre SIC1_ER, si cette valeur est égale à 1 on effectue un deuxième test pour vérifier si la file d'attente est pleine ou non, si elle n'est pas pleine on enregistre les coordonnées X et Y lue par le contrôleur tactile.

Ensuite, elle détermine la position réelle en calculant à chaque fois la moyenne des positions valides, puis effectue le calibrage et la détermination des coordonnées de l'endroit pressé sur l'écran LCD.

Enfin, on doit effacer les sources d'interruption dans les deux registres MIC_RSR et SIC1_RSR afin de prendre en compte la prochaine interruption.

Conclusion

Dans ce chapitre, nous avons tout d'abord décrit notre approche pour la réalisation des objectifs de notre projet.

Suite au travail accompli au niveau de cette partie, il reste les étapes d'implémentation et des tests qui vont faire l'objet du prochain chapitre. Nous allons décrire les outils qui nous ont permis de développer notre projet. Nous allons ainsi donner les résultats des tests effectués sur notre application.

Chapitre IV :

Réalisation et Tests

1. Introduction

Après avoir explicité dans le chapitre précédent la conception de notre application, nous allons présenter dans ce chapitre l'environnement de développement et d'implémentation de notre application et les langages de programmations qui nous ont servi d'appui pour le développement de notre application.

2. Outils de développement utilisés

2.1. Outils logiciels

Pour développer notre application, nous avons eu recours à divers éléments notamment le langage de programmation C et pour environnement de développement keil- μ vision qui nous ont permis l'implémentation de notre application.

2.1.1. Keil - μ vision

Keil Uvision est un outil de développement spécialement conçu pour les systèmes embarqués. Initialement prévu pour les microcontrôleurs de type MC51, il intègre aujourd'hui de nombreuses technologies notamment celle reposant sur le processeur ARM. L'IDE μ Vision combine la gestion de projet, l'environnement d'exécution, l'édition de code source et le programme de débogage dans un environnement unique et puissant. μ Vision est facile à utiliser et accélère le développement de logiciel embarqué.

Le μ Vision Debugger fournit un environnement unique dans lequel nous pouvons tester, vérifier et optimiser notre code d'application. Il inclut des fonctionnalités traditionnelles comme des points d'arrêt simples et complexes, le contrôle de l'exécution et fournit une visibilité complète des périphériques.

Keil-uvison intègre également des outils de programmation permettant d'envoyer le code en mémoire du microcontrôleur et un simulateur reconnaissant de nombreuses cartes pour simuler l'exécution du code et tester son fonctionnement même si nous ne disposons pas de matériel physique. [28]

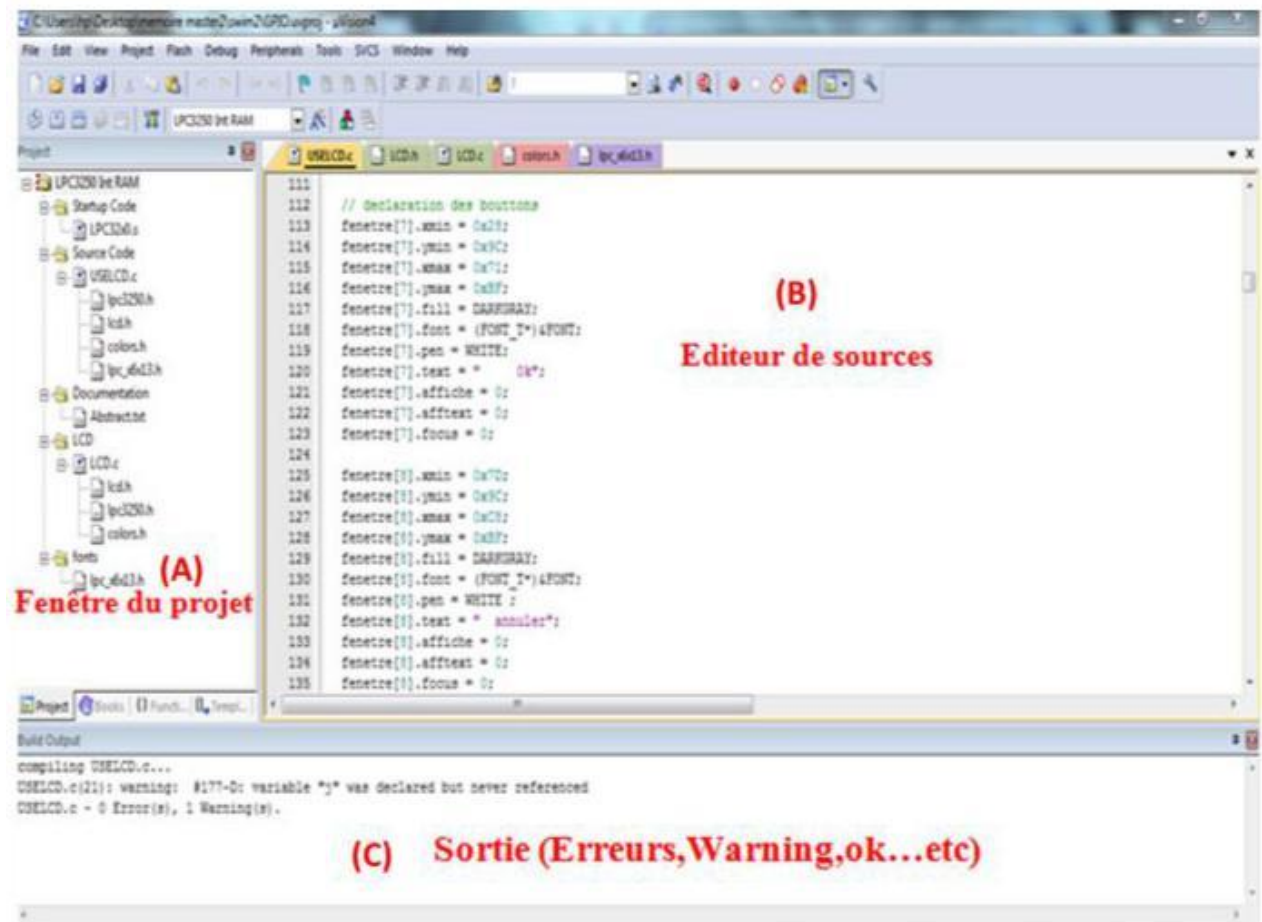


Figure IV.1-La plateforme Keil- µvision

La fenêtre A : contient l’interface « Projets » qui affiche tous les projets ouverts dans Keil-µvision à un instant donné.

La fenêtre B : est un Editeur de texte utilisé pour la déclaration des variables, des structures et des fonctions doté d’une colorisation de syntaxe.

La fenêtre C : est utilisée pour afficher les résultats de compilation : des messages d’erreurs ou de warning...etc.

2.1.2. Le langage C

Nous avons choisit le langage C parce qu’il permet la programmation de bas niveau très populaire, Il est portable, libre, faiblement typé (peu de types de variables différents : son fonctionnement est donc proche de l’ordinateur (gain en rapidité), ce qui le rend plus adapté aux systèmes embarqués.

Le C est un langage complexe (notamment à cause de l'adoption de notions avancées tels que les pointeurs) mais ses qualités font de lui un langage incontournable en matière de programmation pour systèmes embarqués. [29]

2.1.3. L'Extensible Markup Language (XML)

L'Extensible Markup Language (**XML**, « langage de balisage extensible » en français) est un métalangage informatique de balisage générique qui dérive du SGML (Standard Generalized Markup Language) et de HTML (HyperText Markup Language). Comme ces derniers, il s'agit d'un langage orienté texte et formé de balises (markup) qui permettent d'organiser les données de manière structurée. [30]

Cette syntaxe est dite « extensible » Cela signifie qu'on accole aux données des « étiquettes » qui qualifient leur contenu. Elle est reconnaissable par son usage des chevrons (<,>) encadrant les balises.

XML peut être considéré comme un métalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte.

Nous avons utilisé l'XML pour permettre à un utilisateur de décrire facilement son interface graphique. Par exemple le code de la figure IV.2 décrit la fenêtre principal d'un écran de taille 320 X 240 pixels contenant un texte écrit en blanc sur font noir et un champ de texte de couleur noir portant la mention désignation en blanc.

```
<fenetre>
    <id> 0 </id>
    <xmin> 0 </xmin>
    <ymin> 0 </ymin>
    <xmax> 0xF0 </xmax>
    <ymin> 0x140 </ymin>
    <fill> BLACK </fill>
    <affiche> 0 </affiche>
    <font> (FONT_T*)&FONT </font>
    <pen> WHITE </pen>
    <text> "" </text>
    <active> 0 </active>
</fenetre>

<champ_text>
    <id> 1 </id>
    <xmin> 0x7 </xmin>
    <ymin> 0xF </ymin>
    <xmax> 0x32 </xmax>
    <ymin> 0x28 </ymin>
    <fill> BLACK </fill>
    <affiche> 0 </affiche>
    <font> (FONT_T*)&FONT </font>
    <pen> WHITE </pen>
    <text> "designation" </text>
    <active> 0 </active>
</champ_text>
```

Figure IV.2-Exemple de fichier XML

2.2. Outils matériels

Notre projet concerne la carte de développement Phycore LPC3250 dotée d'un microcontrôleur ARM926EJ-S (voir l'annexe pour les détails de la carte).

Pour injecter le code dans l'EPROM de la carte, nous avons utilisé le JTAG. C'est une sonde de marque ULINK2 qui permet également le débogage de notre système. Nous avons également utilisé l'écran LCD TFT fournit avec la carte, branché sur l'interface de connexion LCD de la Phycore LPC3250 pour afficher notre interface graphique.

3. Fonctionnement de notre application

Dans ce qui suit, nous allons présenter des captures d'écrans de notre application et expliquer les actions réalisées. Pour illustrer l'utilité de notre application, nous donnerons un exemple illustratif à la fin de ce chapitre.

3.1. Description du générateur

La syntaxe du générateur : générateur <fichier XML> <fichier header généré>.

Cette étape se résume à l'exécution du générateur d'interface graphique tactile, pour cela l'utilisateur doit écrire son fichier XML indiquant des composants selon ses exigences :

zones de textes, boutons...etc. Par la suite, Le générateur extrait les informations du fichier XML, les transforme en un fichier header qui, par la suite, sera inclut dans le code de l'interface en question. Pour valider le bon fonctionnement de notre générateur, nous avons conçu deux exemples d'interfaces graphiques tactiles.

Exemple 1 : Affichage d'un formulaire de base

Pour créer un formulaire, une fenêtre de base est nécessaire dans laquelle les éléments graphiques (objets) seront insérés.

Les principales propriétés d'une fenêtre graphique sont : la taille, le texte, la dimension dans l'écran et la couleur de fond. La taille et la position de la fenêtre (ou d'un objet) se fixent par spécification des coordonnées(x, y) : (xmin, ymin, xmax, ymax). On peut notamment indiquer la valeur d'une couleur par son nom: "WHITE", "BLACK", "RED", "YELLOW"... aux fenêtres ainsi que celle du texte déclaré dans le fichier header color.h.

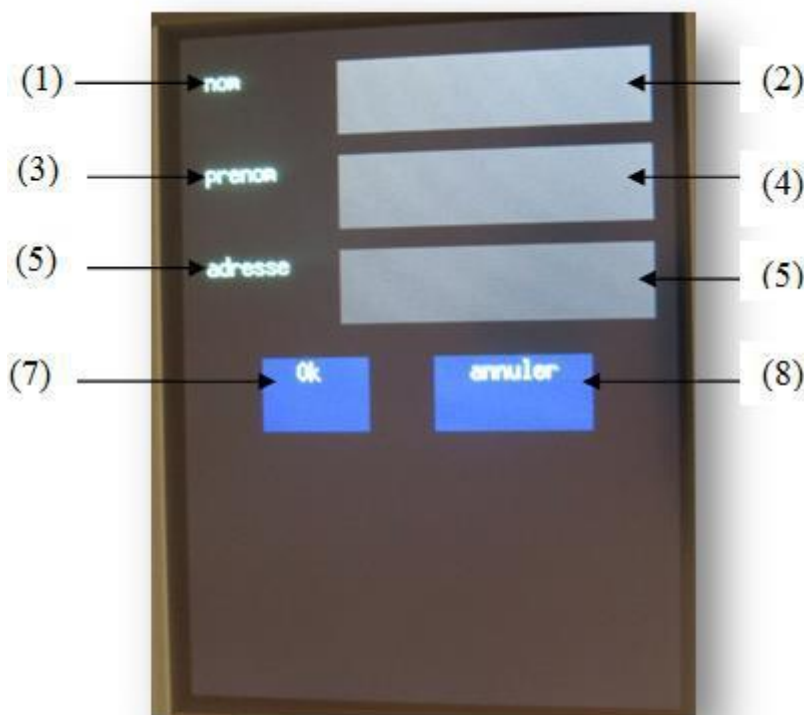


Figure IV.3-Interface tactile

Notre interface se présente comme suit :

Fenêtre de base (0, 0, 240, 320): ces valeurs permettent de fixer la taille de la fenêtre par rapport à la dimension physique de l'écran, en effet cette fenêtre prend l'intégralité de la surface et de la dalle.

Zone de texte: elle s'affiche en forme d'un rectangle pouvant contenir différents types de données, chaque zone de texte est une fenêtre incluse sur la fenêtre de base. L'interface graphique affiche deux types de zones de textes :

Zone de texte verrouillée : elle est l'objet le plus simple; cette zone permet de placer un texte dans la fenêtre ; toutes les propriétés de cet objet ne peuvent être modifiées ou supprimées.

Zone de texte non verrouillé (champ de saisie): contrairement aux zones de textes verrouillées, l'utilisateur peut saisir un texte dans le champ de saisie.

1 : est une zone de texte verrouillée dont les coordonnées sont comme suit : (7, 15, 50, 40) de couleur noir contenant le texte « nom » de couleur blanche.

2 : est une zone de texte non verrouillé dont les coordonnées sont comme suit: (80, 15, 230, 50) de couleur Grise, l'utilisateur peut saisir dans cette zone son nom par exemple.

3 et 5 : sont des zones de textes verrouillées, elles ont des propriétés identiques à celle de 1 la seule différence, est leurs texte et leurs coordonnées : prénom = (7, 60, 50, 75), adresse = (7, 105, 50, 188).

4 et 6 : constituent aussi les zones de textes non verrouillées, elles ont des propriétés similaires à la zone 2, seule leurs coordonnées les différencient. (Zone de texte 4 = (80, 60, 230, 95), (zone de texte 5 = (80, 105, 230, 140)).

Bouton : est une fenêtre qui symbolise une zone de commande, dans le cas de notre interface les fenêtres 7 et 8 sont des boutons qui ont des propriétés identiques mis à part leurs textes, leurs traitements effectués et leurs coordonnées (bouton **ok** = (40, 156, 113, 191)) (bouton **annuler** = (125, 156, 200, 191)).

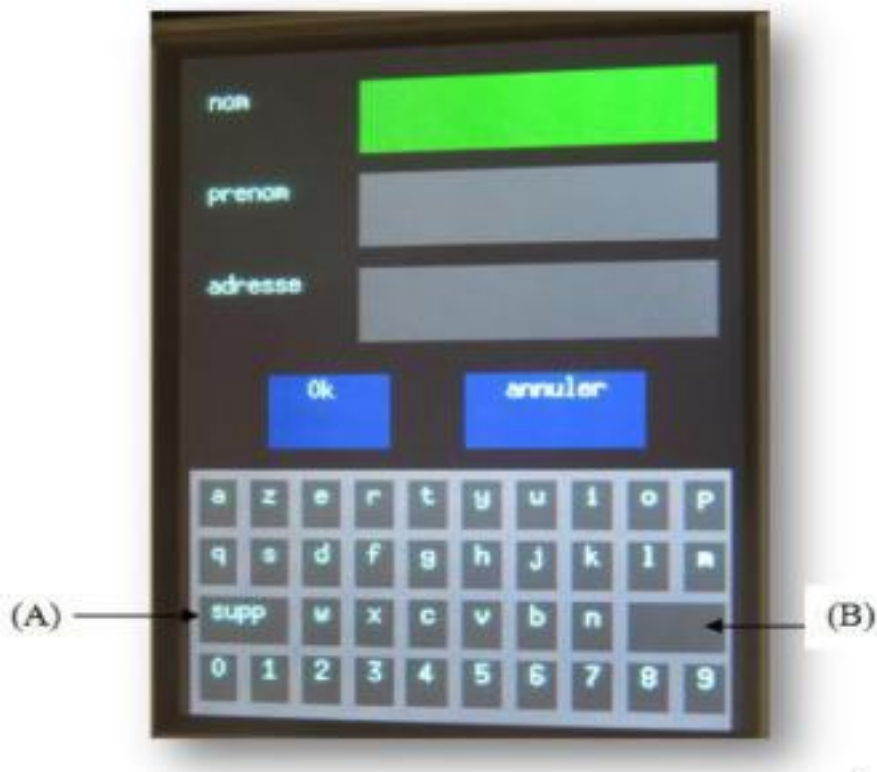


Figure IV.4-Appuis sur l'une des zones

Dès lors qu'un utilisateur appuie sur l'une des zones de textes, la couleur de la zone change du gris vers le vert, un clavier alphanumérique apparait comportant 36 caractères :

Dix chiffre allant de 0 jusqu'à 9 et vingt-six lettre de (a à z).

A : permet de laisser un espace entre les mots.

B : permet la suppression des caractères saisis.

1. Ecriture

La figure IV.5 illustre la saisie sur le formulaire :

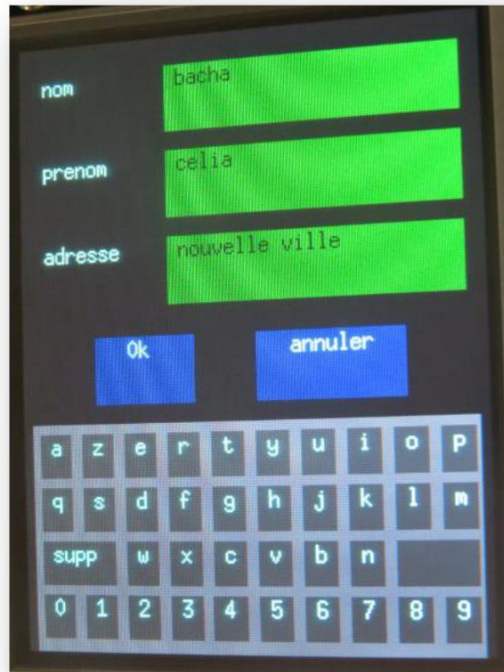


Figure IV.5 saisie dans la zone de texte

Pour la saisie, l'utilisateur clique sur une des zones de textes, une fois que le clavier virtuel apparait et ainsi appuyer sur un des caractères désirés, ce dernier change de couleur puis s'affiche sur le champ de saisie.

2. Suppression

Lorsque l'utilisateur désire effacer un caractère, il appuie sur la zone « supp » présente sur le clavier virtuel. Exemple : supprimer le 'a' de la zone de texte (celia → celi) comme le montre les figures IV.6 :



Figure IV.6 avant la suppression



Figure IV.7 Après la suppression

3. L'action des boutons

Il revient au développeur de l'application d'associer l'action adéquate aux boutons selon l'objectif pour le quel l'interface sera employée.

Dans se qui suit on va présenter un exemple simple de l'action du bouton OK selon le formulaire précédant.



Figure IV.8-Remplissage de toutes les zones de textes

Lorsque l'utilisateur saisit ses informations et appuie sur le bouton **OK** un test sera effectué :

Si l'intégralité des champs sont remplis alors une nouvelle fenêtre s'affiche indiquant que l'opération s'est effectuée avec succès. Comme l'illustre la figure IV.9.

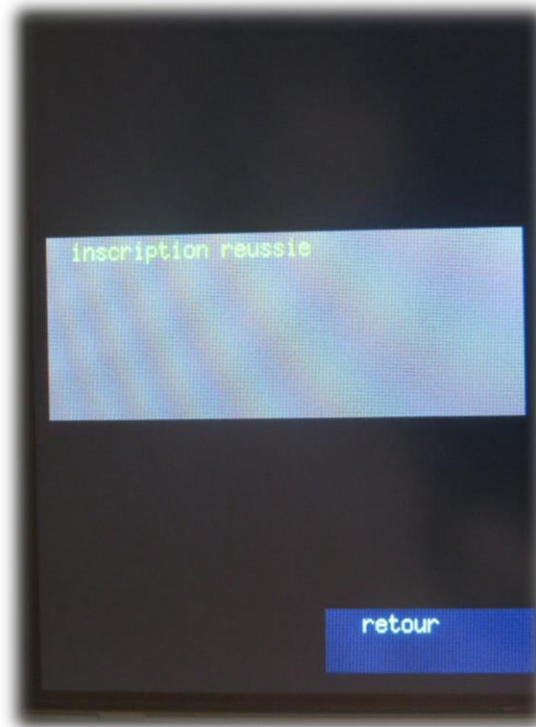


Figure IV.9-opération effectuée avec succès

Pour revenir à la fenêtre principale, l'utilisateur appuie sur le bouton « retour ».

Si l'une des zones de textes est vide, une nouvelle fenêtre s'affiche contenant le message «veuillez remplir tous les champs» demandant de remplir tous les champs de saisies comme l'expose la figure IV.10.

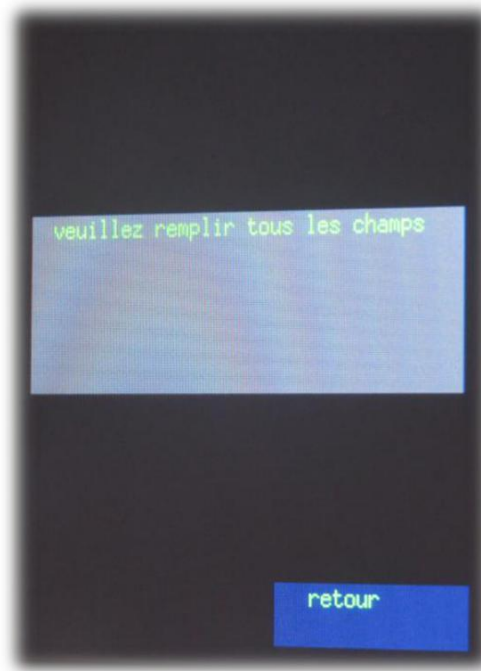


Figure IV.10-signalisation d'un champ non remplis

Pour retourner à la fenêtre principale et continuer la saisie des zones, l'utilisateur appuie sur le bouton « retour ».

Exemple 2 : utilisation du formulaire dans le cadre d'un projet de développement

L'objectif de notre travail étant la création de l'interface graphique uniquement, comme le font tous les générateurs d'interfaces existants, il revient aux développeurs d'application d'ajouter les différents traitements associés aux éléments de commande de l'interface (boutons, menus, ... etc.).

Nous donnons dans ce qui suit un exemple d'une application simple qui consiste à calculer le montant d'un produit spécifié par son prix unitaire et la quantité. Dans ce cas, notre application lui génère l'interface contenant le nom du produit, son prix unitaire, sa quantité ainsi que deux boutons calculer et réinitialiser. Le développeur n'a qu'à ajouter la fonction nécessaire au calcul et l'associer à l'action du bouton calculer.

1- L'action du bouton calculer :

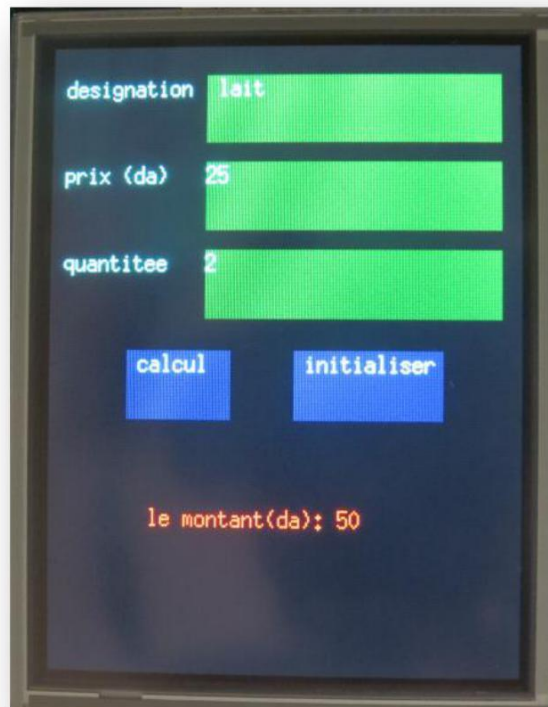


Figure IV.11-Action du bouton calculer

Après la saisie des informations sur le formulaire et que l'utilisateur appuie sur le bouton calculer, la fonction appliqué sur ce bouton va récupérer en premier lieu le contenu de la deuxième zone de texte puis celui de la troisième et effectue le calcul suivant :

Montant = prix * quantité

Une fois que le calcul est résolu le montant en dinars (DA) s'affiche sur notre écran. Pour effectuer le calcul sur le contenu des zones de texte de type « chaîne de caractère », une conversion de type s'impose. Comme l'illustre le code suivant :

```
int conv_char_int(char *tab){  
    int nbre = 0, i = 0, len = 0;  
    char c;  
    /*calculer le nombre de caractères à convertir*/  
    while ((c = tab[i]) != '\0'){  
        i++;  
        len++;  
    }  
    for(i=0; i<len; i++){  
        nbre = nbre * 10 + ( tab[i] - '0' );  
    } return nbre;  
}
```

Vu que les valeurs ASCII sont consécutives, donc, en soustrayant la valeur ASCII de 0 (valeur ASCII 48) à partir d'un caractère numérique donne la valeur du nombre.

Par exemple, convertir le caractère '5' en un entier on procède comme suit :

Le code ASCII de '5'=53 donc si on soustrait la valeur 53 de 48 on aura : 5.

Nous avons souhaité compléter notre interface pour générer d'autres objets comme les menus, les graphs et les conteneurs d'images, mais nous n'avons pas pu le faire faute de temps. Toutefois ces objets auront la même structure de base que les objets existants et pourront être intégrés facilement au générateur d'interface. Nous les laissons comme perspectives de notre travail.

Conclusion

Au cours de ce dernier chapitre, nous avons décrit l'étape de réalisation de notre projet. En premier lieu, nous avons présenté les outils matériels et logiciels utilisés pour développer et implémenter notre application. En second lieu nous avons exposé le fonctionnement de notre application à travers de nombreuses figures.

Conclusion générale et perspectives

Conclusion générale et perspectives

Le principal objectif de notre travail consiste en la conception et la réalisation d'un générateur pouvant traduire un fichier XML écrit par un utilisateur en une interface graphique tactile en rajoutant tout le code nécessaire pour la gestion au niveau bas de l'écran et de l'interface tactile. Cette interface est affichée sur un écran LCD connecté à une carte PHYCOR LPC 3250 disponible au niveau de notre département. Cette dernière fournit une interface de connexion flexible pour supporter divers panneaux LCD :

Ce travail nous a permis de participer à une expérience intéressante qui nous a renforcé et enrichit nos connaissances et ce à travers :

L'acquisition d'une précieuse expérience relative au fonctionnement des écrans LCD et tactile.

L'amélioration de nos connaissances sur le langage de programmation (langage C) sous l'environnement de développement keil- μ vision.

L'approfondissement de nos connaissances sur la carte de développement (Phycore LPC 3250).

L'initiation au domaine complexe de développement au niveau bas, notamment les drivers de périphériques.

En guise de perspective, nous souhaitons étendre notre travail pour inclure d'autres interfaces comme : des menus, des graphes ainsi que des conteneurs d'images.

Ce projet n'est qu'une étape permettant de doter la carte LPC-3250 d'un moyen de création d'interfaces graphiques qui sera utilisés dans de nombreux projets envisagés pour cette dernière.

Bibliographiques

Références bibliographiques

- [1] : Alain LEBEGUE, systèmes embarqués, mobilité et objets communicants, PARIS, 2012 P.1
- [2] : BAPTISTE OSSOLA – PIERRE PATTARD, séminaire tv écrans plats, FEVRIER 2007, P.3
- [3] : histoire des premiers écrans aux écrans tactiles, [online] :
<http://www.histoire-cigref.org/blog/histoire-des-premiers-ecrans-aux-ecrans-tactiles/>
- [4] : ELIE ZAGURY, comment fonctionnent les pixels sur les écrans et à quoi servent ils ?,
<http://www.rts.ch/decouverte/sciences-et-environnement/5703174-comment-fonctionnent-les-pixels-sur-les-ecrans-et-a-quoi-servent-ils.html>
- [5] : [online] : https://fr.wikipedia.org/wiki/Profondeur_des_couleurs
- [6] : ANTOINE MALAISE, informatique de base, Bruxelles 2015 – 2016
- [7] : [online] :
<http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/technologie-lcd-1835/> [8]: Matrice passive, <http://www.01net.com/actualites/matrice-passive-191812.html> [9] : CHRISTIAN WEISS, Téléviseurs à écrans plats, Bordeaux, P .14, 15.
- [10]: NXP founded by Philips, draft LPC32x0 User manual, LPC32x0_UM, 20 November 2008 P.204
- [11]: [Online]: https://issuu.com/bilbaxo/docs/configuration_et_maintenance_pc P.275
- [12]: Écrans OLED, IPS, VA, TN... Quelles sont les différences? [Online]:
<http://www.01net.com/actualites/ecrans-oled-ips-va-tn-queelles-sont-les-differences-640957.html>
- [13] : JEAN-FRANÇOIS PILLOU, écran plat, Septembre 2015, P.2.
- [14]: Manuel d'utilisation NXP Semi-conducteurs N.V, LPC3220/30/40/50 16/32-bit ARM microcontrôleurs ,24 Juin 2014, P.33.
- [15]: Displays and Frame buffers, University of Texas at Austin, 2010.
- [16] : TOM MIETTE, Les technologies tactiles, Filière Informatique et Réseaux, Université Paris-Est de Marne-la-Vallée 3 février 2009, P. 4
- [17] : Les écrans tactiles [Online] : <https://prezi.com/vjutkigfipjq/ecrans-tactiles/#>
- [18] : Interface tactile, 2011, [Online] <http://interfacetactile.com/ecran-tactile-capacitif>
- [19] : Les solutions tactiles et leurs applications, [Online] :
<http://pagesperso.lina.univ-nantes.fr/~prie-y/archives/VEILLE-2009-2012/2010/interfaces-tactiles/axe1.html#>
- [20] : DUBOIS ALEXIS, HUYNH Quang Nghi, MILETIC Nikolas, Comment fonctionne un écran tactile ? P.20
- [21]: [Online]: <http://www.lemagit.fr/definition/Controleur>

Bibliographiques

- [22]: The Working Principle of 4 Wire Resistive Touchscreen [Online]:
<http://www.onetouch.com.tw/en/images/tech/Eng-4W.pdf>
- [23] : RICHARD GRIN, interface graphique – introduction, université de Nice-Sofia Antipolis, 13/3/11
- [24] : JEAN BERSTEL - Introduction au système X, Institut Gaspard Monge, Université Marne-la-Vallée, P.14
- [25] : Zone de texte [Online] https://fr.wikipedia.org/wiki/Zone_de_texte
- [26] : Docteur ordinateur, Assistance informatique à domicile, Bouton, [Online] :
<http://www.docteurordinateur.com/dictionnaire-de-termes-techniques-informatiques.html>
- [27] : S. AROUSSI, interruptions, Université Saad Dahlab de Blida Faculté des Sciences Département d'Informatique Licence Génie des Systèmes Informatiques <https://sites.google.com/a/esi.dz/s-aroussi/>
- [28] : M^r DAOUI, Keil μ vision, Outil de développement des systèmes embarqués, 2015-2016
- [29] : Introduction au langage C. [Online] : <http://www.depannetonpc.net/>
- [30] : JEAN-FRANÇOIS PILLOU, Introduction à XML PDF, Septembre 2015
- [31]: KAOHSIUNG HITACHI, Customer's acceptance specifications TX09D70VM1CCA, kaohsiung taiwan, 17 Février 2006
- [32]: USB-JTAG ADAPTER [Online]:
HTTPS://WWW.EMBEDDEDARTISTS.COM/PRODUCTS/TOOLS/TOOL_ULINK2.PHP
- [33]: PHYTEC, phyCORE-LPC3250 System on Module and Carrier Board Hardware Manual, 27 Janvier 2009 P. 66

Annexe

1. Présentation de La carte Phycore LPC3250

Le module Phycore LPC3250 (figure A.1) est une plateforme de développement conçu par la société NXP¹¹ semi-conducteur pour les applications embarquées complexes nécessitant de hautes performances et une faible consommation d'énergie. La carte LPC3250 est basée sur le processeur ARM926EJ-S qui utilise une architecture de type Harvard avec un pipeline à 5 étages et doté d'une vitesse de 266MHZ. [10]

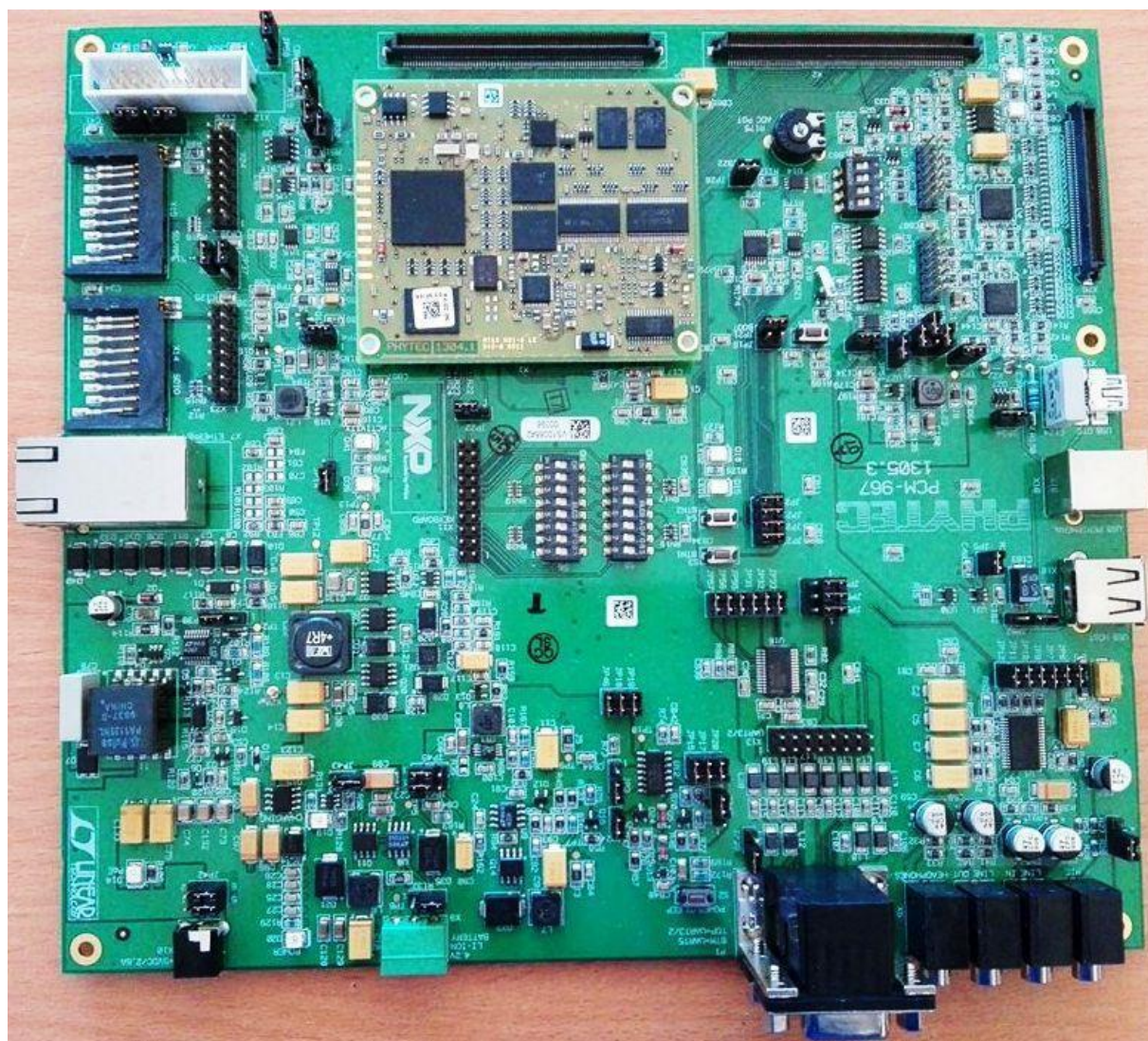


Figure A.1 Carte LPC3250

¹¹ NXP : est un fabricant de semi-conducteurs, fondé le 31 août 2006, NXP Semi-conducteur a conçu la carte LPC32x0 pour les applications embarquées

1.1. Composants de la carte LPC3250

La carte Phycore LPC3250 est composé de plusieurs éléments dont:

Différents types de mémoires :

Une mémoire Nand Flash de 16MB¹² à 128MB. Une RAM¹³ de 128 MB à 256 MB.

Une SRAM¹⁴ de 64 MB.

Une mémoire SDRAM¹⁵ de 16MB Random Access Memory à 128MB.

7 interfaces séries UART.

Une UART est une interface classique de communication série en mode asynchrone.

2 interfaces séries SSP¹⁶.

2 interfaces séries I2C¹⁷.

2 interfaces séries I2S¹⁸.

2 interfaces séries SPI¹⁹.

1 interface USB²⁰.

4 ports d'entrées sorties, chacun est composé de plusieurs broches.

12 GPI²¹, 24 GPO²², 51 GPIO²³.

1 convertisseur analogique/numérique (ADC).

Contrôleur LCD.

permet de supporter divers panneaux LCD (TFT, STN).

Contrôleur tactile. [10]

12 mégabyte

13 Random Access Memory

14 Static Random Access Memory

15 Synchronous Dynamic Random Access Memory(en français mémoire dynamique synchrone à accès aléatoire)

16 Synchronous Serial Port

17 Inter-Integrated Circuit

18 Inter-IC Sound, Integrated Interchip Sound (un standard d'interface électrique de type serial bus pour connecter des matériels audio numériques ensemble)

19 Serial Peripheral Interface (est un bus de données série synchrone)

20 Universal Serial Bus (USB, en français Bus Universel en Série)

21 General Purpose Input (Entrée pour un Usage Général)

22 General Purpose Output (sortie pour un Usage Général)

23 General Purpose Input/Output, (Entrée/Sortie pour un Usage Général)

Annexe

1.2. Branchement d'un écran LCD sur la LPC3250

La carte Phycore-LPC3250 fournit une interface de connexion LCD flexible pour supporter divers panneaux LCD. [10]

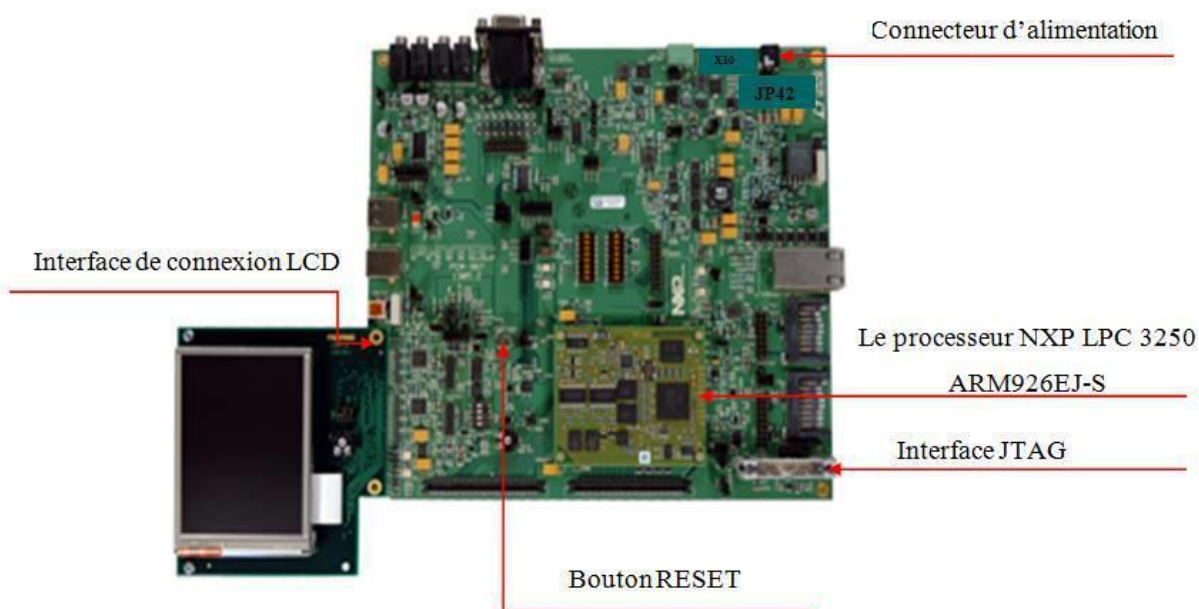


Figure A.2 Branchement de l'écran LCD sur la carte lpc3250

Caractéristiques techniques de l'écran

- Panel Marque : HITACHI²⁴
- Panel model : TX09D71VM1CCA
- Diagonale d'afficheur 3.5 pouces
- Résolution : 320*240
- Tension d'alimentation 3.3 v
- Temps de réponse 30 ms
- Apparence de l'affichage RVB²⁵
- Panel type : TFT-LCD
- La fréquence : 60 HZ [31]

²⁴ Entreprise japonaise d'électronique, elle a été fondée en 1910 par l'ingénieur électrique Namihei Odaira

²⁵ Rouge Vert Bleu

Interface JTAG

Le JTAG²⁶ est une sonde de marque ULINK2 qui connecte un PC à la carte LPC 3250, permettant ainsi d'injecter le code dans l'EPROM²⁷ de la carte ainsi que le débogage du système. [32]

Connecteur d'alimentation

La puissance d'entrée primaire de la carte LPC 3250 est située au niveau du connecteur X10 et sélectionné par JP42 à partir de la prise mural de l'adaptateur .Un adaptateur avec une alimentation minimum de 2600mA est recommandé. [33]

Bouton RESET

Le bouton nous permet une réinitialisation du système via le signal RESET_SYS.

²⁶ Joint Test Action Group

²⁷ Erasable Programmable Read-Only Memory