

République Algérienne Démocratique et populaire
Ministre de l'Enseignement supérieur
Et de la recherche scientifique
Université Mouloud Mammeri Tizi Ouzou
Faculté de Génie électrique et Informatique
Département Informatique



Mémoire présenté en vue de l'obtention du diplôme de Master

Option : Informatique

Spécialité : Réseaux mobilité et système embarqués

Intitulé :

**Réalisation d'un système de
gestion des files d'attente**

Encadreur : Mlle Rebiha HADAoui

Co-encadreur : M. DJIOUA

Présenté par :

Mohamed RAHMANI
Khiredine REZZIK

Membres du jury :

Président : M. DAOUI

Examinatrice : Mme. BELKADI

Examinatrice : Mme. AOUIJIT

Année universitaire

2015-2016

Liste des figures

Chapitre I

Figure I.1 Structure générale d'un système de file d'attente.....	8
FigureI.1 Système de file d'attente avec un serveur unique.....	9
Figure I.3 Système de file d'attente à serveurs parallèles.....	10
Figure I.4 Système de file d'attente à serveurs en série.....	10

Chapitre II

Figure II.1 Elément d'une liaison entre systèmesinformatiques.....	17
FigureII.2 RS485.....	18
FigureII.3 RS232.....	19
FigureII.4 I2C.....	19
FigureII.5 Bus CAN.....	21
Figure II.6 comparaison entre les bus filaire.....	21
FigureII.7 ZigBee.....	22
FigureII.8 Bluetooth.....	23
FigureII.9 Comparaison entre les bus sans fil.....	24
FigureII.10 Tension des ligne de la liaison RS485.....	25
FigureII.11 Schéma de la structure du message envoyer a l'aide de la liaison rs485.....	27
FigureII.12 Echange de maitre vers 1 esclave.....	28
FigureII.13 Echange de maitre vers tout les esclaves.....	28
FigureII.13 Mode RTU.....	28
FigureII.15Oode ASC II.....	29

Chapitre III

FigureIII.1Arduino	32
FigureIII.2 Interface du logiciel ARDUINO.....	33
FigureIII.3Raspberry Pi 2.....	35

Chapitre VI

Figure VI.1 Cycle de vie d'un produit.....	41
Figure VI.2 Architecture général de notre application.....	42
Figure VI.3 Diagramme d'appel du module de traitement.....	43
Figure VI 4 Diagramme de flux de donnée du module de traitement.....	44
Figure VI.5 Diagramme de tâche du module de traitement.....	45
Figure VI.6 Diagramme d'appel du module central.....	46
Figure VI 7 Diagramme de flux de donnée du module central.....	47
Figure VI.8 Diagramme de tâche du module central.....	48

Chapitre V

FigureV.1 GPIO de la Raspberry PI.....	52
FigureV.1 MAX485.....	53
FigureV.2 Ecran LCD.....	54
FigureV.3 Configurer de l'adresse ip en mode statique.....	58
FigureV.4 Détermination de la passerelle par défaut.....	58
FigureV.5 Interface de la page d'accueil du site web.....	60
FigureV.6 Interface de l'espace client dans le site web.....	61
FigureV.7 Interface général de notre application.....	61
Figure V.8Câblagede l'arduino et la raspberry pi.....	62
Figure V.9Affichage du numéro du client sur l'application web et sur l'écranLCD.....	63
Figure V.10 Incrémentation du numéro du client sur l'application web et sur l'écran LCD...	63

Sommaire

Introduction général.....	1
----------------------------------	----------

Chapitre I : généralité sur les files d'attente.

I. Introduction	5
I.1 Développement de la théorie des files d'attentes	5
I.2 définitions générales.....	6
I.3 les composant de Files d'attente	7
I.3.1 La population "clients"	8
I.3.2 Le nombre de serveurs "guichés"	8
I.3.2.1 Systèmes de files d'attente avec un serveur unique :	8
I.3.2.2 Systèmes de files d'attente avec plusieurs serveurs	9
I.4 discipline de service	11
Conclusion	12

Chapitre II : les système embarquées.

I.1 Introduction	14
I.2 Généralités sur les systèmes embarqués	14
I.3 Historique	14
I.4 Contrainte	14
I.5 Caractéristique	15
II les bus de communication.....	16
II.2 Normes de bus	17
II.3 Constitution d'une liaison	17
II.3.1 Modes d'exploitation.....	18
II.4 Type de liaison	18
II.4.1 Liaison filaire	18
II.4.2 Liaison sans fils	22
III RS485	
III.1Historique	24
III.2 Le Bus RS485	24
III.3 Principales caractéristiques électriques de la norme RS48.....	25
III.4Le protocole	26
III.4.1 Structure des message	27
III.4.2 Adressage	27
III.4.3 Echange maître vers 1 esclave.....	27
III.4.4 Echange Maître vers tous les esclaves	28
III.4.5 Format général d'une trame.....	28
III.4.6 Trame d'échange question/réponse	29
Conclusion	29

Chapitre 3 : Raspberry & Arduino

Introduction	31
I.2 Carte electioneer.....	31
I.3 Microcontroller	31
II Ardouino.....	31
II.2 Matériel	32

II.3 L'environnement de developementr	33
II.4 domaine d'application de Arduino	34
II.5 Avantage	34
III.LaRaspberry Pi	
III.1 presentation de la raspberry pi	35
III.2 Historique	36
III.2.1 Conception	36
III.2.2 Prototype	36
III.2.3 Lancement	36
III.3 Architecture matérielle	37
III.4 Spécifications	37
III.5 Équipement supplémentaire :.....	37
III.6 les système d'exploitation compatible avec la raspberry pi	38
Conclusion	39

ChapitreVI : Analyse et conception

Introduction.....	41
I Analyse et Conception	41
I.1 Cycle de vie d'un produit	41
II Etape Analyse.....	42
II.1 Architecture général de notre système	42
II.2Fonctionnement de notre système.....	42
III Conception.....	43
III.1 Module de traitement	43
III.1.1Diagramme d'appel..	43
III.1.2 Diagramme de flux de données	43
III.1.3Diagramme de tache	44
III.2 Module central	46
III.2.1 Diagramme d'appe.....	46
III.2.2Diagramme de flux de données	46
III.2.3 Diagramme de tache	47
Conclusion	49

Chapitre 5 Realisation& mise en œuvre.

I Description de l'environnement du développement	51
I.1 Matériels	51
I.1.1 Arduino	51
I.1.2Raspberry Pi	51
I.1.3 MAX485	52
I.I.4 Ecran LCD	53
I.2 Logiciels	55
I.1.1 Le serveur web Apache2	55
I.2.2 Sqlite3	55
I.2.3 Adobe Dreamweaver CC 2014	56
I.3 Langages utilisés	56
I.3.1 HTML5 (hypertextMarkupLanguage 5)	56
I.3.2 PHP5	56

I.3.3 CSS3	56
I.3.4 Python...	56
I.3.5 Arduino.....	57
II Configuration du matériel	57
II.1 Configuration du Raspberry Pi	57
II.1.1 Attribuer une adresse IP statique pour le Raspberry PI	57
II.1.2 Installer un serveur Web sur la Raspberry Pi.....	58
II.1.3 Libérer le port série de la Raspberry PI.....	59
II.2 Création d'un site web dynamique :	60
II.3 L'Architecture de notre application	61
Conclusion	64
 Conclusion général	66
Bibliographia	67

Introduction générale :

L'utilisation des files d'attente remonte au début des années 1950. Ce formalisme a été essentiellement utilisé pour la modélisation des systèmes industriels. Mais, de nos jours, il représente un outil de modélisation analytique de plusieurs types de systèmes et notamment les systèmes de service tels que dans l'optimisation de trafic. La théorie des files d'attente est une technique de la Recherche Opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les gestionnaires dans leurs prises de décisions.

Les files d'attentes se forment lorsque les clients arrivent de façon aléatoire pour se faire servir. Les exemples les plus courants de la vie de tous les jours sont les caisses des supermarchés, les établissements de restauration rapide, les billetteries des aéroports, les cinémas, les bureaux de poste, les banques, etc. Toutefois, lorsqu'on parle d'attente, on pense souvent à des personnes. Or, les « clients » en attente sont aussi des commandes en attente de traitement, des camions en attente de chargement ou de déchargement, des machines en attente de réparation, des programmes d'ordinateur qui attendent d'être exécutés, des avions qui attendent l'autorisation de décoller, des bateaux qui attendent les remorqueurs pour accoster, les voitures aux panneaux d'arrêt, les patients dans les salles d'urgence, etc. si les clients attendent trop longtemps, ils associent cette perte de temps à une mauvaise qualité de service. Pour éviter ces situations, la majorité des entreprises ont mis en place des processus d'amélioration continue dans le but ultime est l'élimination de toute forme de gaspillage, notamment l'attente. Et c'est dans ce cadre que se situe notre projet de fin d'étude intitulé « réalisation d'un système de gestion des files d'attente »

Pour cela, notre mémoire va être structuré autour de cinq chapitres organisés comme le suivant :

- ✓ Le premier chapitre consiste en premier lieu à introduire les notions de base exprimant les phénomènes d'attente, nous allons définir les éléments importants à considérer dans la mesure de performance d'un système de file d'attente.
- ✓ Ensuite, dans le deuxième chapitre nous mettrons l'accent sur les systèmes embarqués et les bus de communication d'où nous allons spécifier la liaison rs485 qui jouera un rôle important dans notre projet.
- ✓ Le troisième chapitre sera consacré aux deux cartes électroniques utilisées Raspberry Pi et Arduino.
- ✓ Dans le quatrième chapitre, nous procéderons à l'analyse et la conception de notre système où nous allons spécifier le déroulement de notre système, le rôle des composants matériels et les algorithmes utilisés.

- ✓ Le chapitre quatre illustre l'environnement de développement et les divers composants implémentés dans l'architecture de notre système.

Enfin on achève ce mémoire par une conclusion générale récapitulant tout notre travail ainsi que ses perspectives.

Introduction

La Théorie des files d'attente est une technique de la Recherche opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les gestionnaires dans leurs prises de décisions.

Les files d'attente peuvent être considérées comme un phénomène caractéristique de la vie actuel. On les rencontre dans les domaines d'activité les plus divers spécialement devant les guichés. L'étude mathématique des phénomènes d'attente constitue un champ d'application important des processus stochastiques. C'est un phénomène d'attente chaque fois que certaines unités appelées "clients" se présentent d'une manière aléatoire à des guichés de traitement afin de recevoir un service dans la généralement aléatoire.

Il n'y a pas de théorie des files d'attente en tant que telle. La théorie des files d'attente est un formalisme mathématique qui permet de mener des analyses quantitatives à partir de la donnée des caractéristiques du flux d'arrivées et des temps de service.

I.1 Développement de la théorie des files d'attentes

Les origines du formalisme des files d'attente datent du début du 20ème siècle et principalement des travaux de deux mathématiciens : le mathématicien [danois **A.K.Erlang**] avec ses travaux sur les réseaux téléphoniques et le russe [**A.A. Markov**] avec la création des modèles markoviens.

C'est en 1909 que les bases de ce formalisme sont jetées, grâce à l'article du mathématicien [danois **A.K. Erlang**], "The theory of probabilities and telephone conversations". Les premiers résultats sont variés : il observe le caractère poissonnier des arrivées des appels à un central téléphonique, et le caractère exponentiel

des durées des appels; il réussit à calculer de manière relativement simple la probabilité d'avoir un appel rejeté. La notion d'équilibre stationnaire d'un système d'attente est introduite.

À partir des années 30, les travaux de plusieurs mathématiciens tels que [Molina,Fry, Pollaczek] aux États-Unis, [Kolmogorov et Khintchine] en Russie, Palm en Suède, ou Crommelin en France permettent à la théorie des files d'attente de se développer lentement. Ce sont ensuite les années 50 qui verront l'essor important de la théorie.

Les applications de ces travaux sont alors très pratiques, et concernent les disciplines de recherche opérationnelle et génie industriel. Il existe deux flux : flux de trafic (véhicule, avion, personnes, télécommunications), l'ordonnancement, c'est-à-dire la planification, par exemple les patients dans les hôpitaux, les programmes d'un ordinateur, ou encore le dimensionnement (banque, poste, réseaux, téléphone, ordinateur).

Dans les années 80, cette discipline devient beaucoup plus mathématique, et la littérature regorge d'articles décrivant des techniques ou des astuces mathématiques permettant de trouver des solutions exactes aux modèles.

Au cours de la décennie suivante, les chercheurs s'intéressent d'avantage à la création de modèles, et au calcul scientifique associé pour résoudre ces modèles. En effet, le développement de la puissance des ordinateurs permet maintenant d'obtenir des solutions approches des modèles suffisamment fiables pour être utilisées.[1].

Actuellement ce sont les applications dans le domaine de l'analyse de performance des réseaux (téléphone mobile, Internet, multimédia, qui suscitent le plus de travaux.

I.2 définitions générale

Généralement une file d'attente est la donnée d'une (ou plusieurs) unités de services où arrivent des clients qui demandent une certaine durée d'utilisation de cette unité (le service demandé par les clients) .Quand les clients peuvent accéder à cette unité de service, ils patientent dans une file d'attente en attendant d'être servis .La file d'attente peut éventuellement n'accepter qu'un nombre fini des clients, dans ce cas les clients trouvant la file pleine à leur arrivée sont rejetés par le système. Un client peut être servi pendant une certaine période puis abandonné par le serveur. Le service résiduel d'un client est la durée du service qui reste à effectuer, quand celui-ci est nul, le client quitte la file d'attente.

La charge de la file d'attente est la somme de tous les services résiduels de tous les clients présents[2].

I.3 les composants d'une file d'attente :

Tous les exemples de phénomènes d'attente ont des caractéristiques communes que l'on peut résumer ainsi : Des entités circulent dans un système et utilisent des ressources communes. Le système, les entités ou les ressources peuvent avoir un comportement imprévisible. dans le contexte d'une modélisation mathématique, aléatoire.

Une file d'attente se compose des éléments suivants :

1. La population "client".
2. Le nombre de serveurs "guiché".
3. Les tendances quant à l'arrivée et au service.
4. L'ordre de traitement des clients.
5. Une salle d'attente : un lieu où les clients attendent quand aucun des serveurs n'est disponible pour les servir.

Tout les systèmes de file d'attente peut être représenté par le schéma suivant :

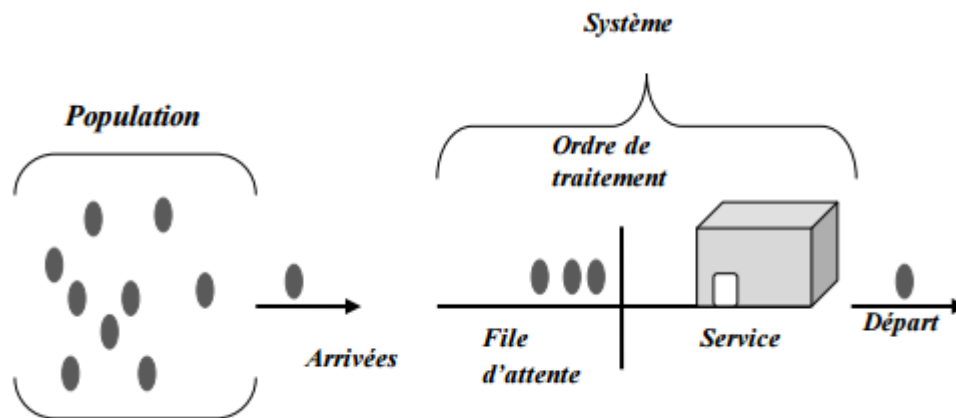


Figure I.1: Structure générale d'un système de file d'attente[20]

I.3.1 La population "clients":

Dans la théorie des files d'attente, la population est la source de clients potentiels. Il y a deux situations possibles. Dans le premier cas, la population est infinie, c'est-à-dire que le nombre potentiel de clients est infiniment grand en tout temps. C'est le cas des clients des supermarchés, des banques, des restaurants, des cinémas, des centres d'appels, trafic urbain, etc... De plus, les clients proviennent de toutes les régions possibles. Dans la deuxième situation, la population est finie, ce qui signifie que le nombre de clients potentiels est limité. Un bon exemple est le nombre de machines, d'avions, etc., en réparation dans le centre de maintenance d'une entreprise.

I.3.2 Le nombre de serveurs "guichés":

La capacité de service dépend de la capacité de chaque serveur et du nombre de serveurs disponibles. Le terme « serveur » représente ici la ressource et, en général, on suppose qu'un serveur ne traite qu'un client à la fois. Les systèmes de files d'attente fonctionnent avec serveur unique ou serveurs multiples.

I.3.2.1 Systèmes de files d'attente avec un serveur unique :

Ce type de système comprend un seul serveur qui offre le service. Les exemples de ce système sont nombreux : les petits magasins avec une seule caisse,

tels que les dépanneurs, certains cinémas, certains lave-autos et établissements de restauration rapide avec guichet unique.

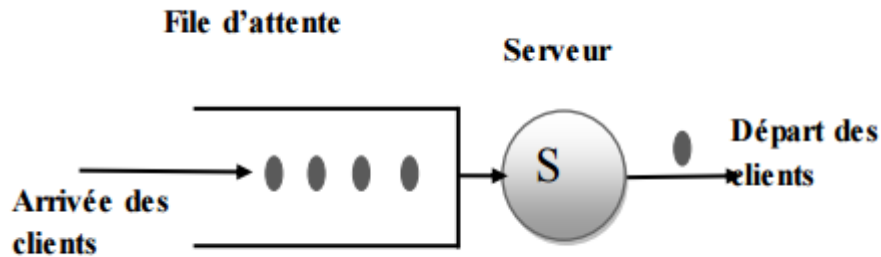


Figure I.2 : Système de file d'attente avec un serveur unique[20]

I.3.2.2 Systèmes de files d'attente avec plusieurs serveurs :

Ce type de systèmes comprend plusieurs serveurs qui fournissent le service. Les serveurs sont en parallèle ou en série.

a. Système de files d'attente avec plusieurs serveurs en parallèles :

Chaque client ne requiert le service que d'un seul serveur et tous les serveurs sont capables de fournir ce service. Dans ce type de files d'attente, on a ' S ' serveurs en parallèle. Le client entrant au système n'est pas obligé de visiter tous les ' S ' serveurs. Si chaque serveur est doté d'une file, au moment de son arrivée le client choisit l'une d'entre elles ; bien sûr il doit choisir la file la moins longue ; ou bien il rejoint une position donnée dans la file si elle est unique, puis il sera sélectionné de cette file selon la politique adoptée dans le système. Si n , le nombre de clients présentes dans le système est inférieur à S , ce qui revient à dire tant que les serveurs ne sont pas tous occupés, alors la file ne se constitue pas et tout client arrivant est immédiatement pris en charge par l'un des serveurs libres. Si la file commence à se constituer et tout client qui arrive doit rejoindre la file.

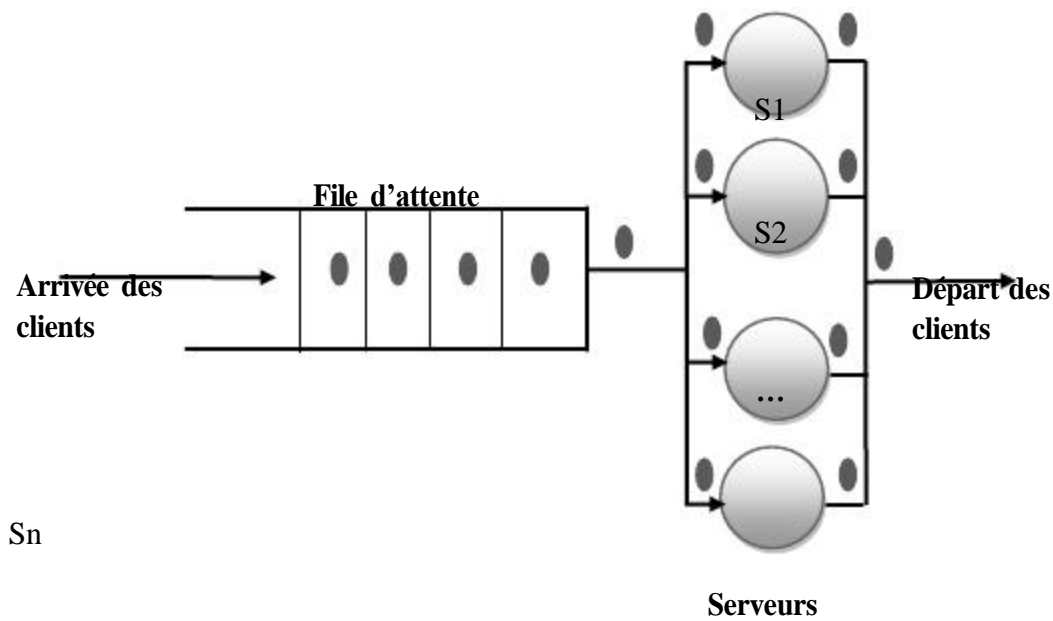


Figure I.3 : Système de file d'attente à serveurs parallèles [20]

b. Système de files d'attente avec plusieurs serveurs en série

Dans ce type de files d'attente, on a S serveurs en série. Le client entrant au système doit visiter plusieurs serveurs successifs dans un ordre fixe pour recevoir le service. Ce type est appelé aussi système de files d'attente en cascade, on les rencontre dans une chaîne de fabrication, dans la circulation des dossiers dans une administration ect[2]

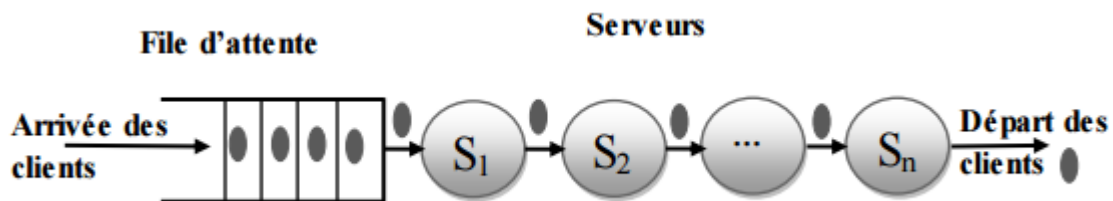


Figure I.4: Système de file d'attente à serveurs en série

I.4 Discipline de service

La discipline de service détermine l'ordre dans lequel les clients sont rangés dans la file et y sont retirés pour recevoir un service. Les disciplines les plus courantes sont :

- ✓ **FIFO** : (first in, first out) ou FCFS (first come first served) ou PAPS (premier arrivé, premier servi) : c'est la file standard dans laquelle les clients sont servis dans leur ordre d'arrivée. Notons que les disciplines FIFO et FCFS ne sont pas équivalentes lorsque la file contient plusieurs serveurs. Dans la première, le premier client arrivé sera le premier à quitter la file alors que dans la deuxième, il sera le premier à commencer son service. Rien empêche alors qu'un client qui commence son service après lui, dans un autre serveur, termine avant lui. En français, le terme PAPS comporte une ambiguïté, puisqu'il ne peut différencier une file "premier arrivé, premier servi" d'une file "premier arrivé, premier sorti".
- ✓ **LIFO** : (last in, first out) ou LCFS (last come, first served) ou DAPS (dernier arrivé, premier servi). Cela correspond à une pile, dans laquelle le dernier client arrivé (donc posé sur la pile) sera le premier traité (retiré de la pile). A nouveau, les disciplines LIFO et LCFS ne sont pas équivalentes que pour une file mono-serveur.
- ✓ **RANDOM** : (aléatoire). Le prochain client qui sera servi est choisi aléatoirement dans la file d'attente.
- ✓ **Round-Robin** : (cyclique). Tous les clients de la file d'attente entrent en service à tour de rôle, effectuant un quantum Q de leur temps de service et sont replacés dans la file, jusqu'à ce que leur service soit totalement accompli. Cette discipline de service a été introduite afin de modéliser des systèmes informatiques .
- ✓ **PS** (Processor Sharing) : C'est le cas limite de la distribution Round-Robin lorsque le quantum de temps Q tend vers 0. Tous les clients sont servis en même temps, mais avec une vitesse inversement proportionnelle au nombre de clients simultanément présents. Si le taux du serveur est égal à U et que à un instant donné il y a N clients à la station, tous les clients sont donc servis simultanément avec un taux u/n .
- ✓ **Priorité**: Il existe plusieurs classes de clients de priorité différente; le client avec

la priorité la plus haute est servi en premier, et la discipline de service est FIFO au sein d'une même classe; Pour la discipline „Priorité", on distingue deux variantes, selon que le client en service est interrompu lorsqu'un client plus prioritaire arrive ou non. La discipline est une discipline de priorité non préemptive si le client en service n'est pas interrompu lorsqu'un client plus prioritaire entre dans le système. Dans le cas contraire, le client en service est interrompu lorsque qu'un client plus prioritaire arrive, afin que ce dernier puisse commencer son service aussitôt. Une fois le service du client prioritaire terminé, le client interrompu reprend son service, et on distingue deux politiques :

- **Priorité préemptive** avec recommencement: le client interrompu reprend son service au début.
- **Priorité préemptive** avec continuation: le client interrompu reprend son service là où il avait été interrompu.

Conclusion

dans ce chapitre nous avons expliqué le phénomène d'attente et présenter les notions de base concernant les systèmes de files d'attente , avec lesquels nous allons réaliser notre application système embarqué pour gérer une files d'attente qui va augmenter la performance de la file en renforsson l'organisation et en améliorent l'orientation des clients.

Chapitre 2

Les systèmes embarquées.

Introduction :

Dans le chapitre précédent nous avons parlé sur les files d'attente et sur ces différentes notions ,il existe plusieurs solution pour gerer les files d'attente, et dans notre cas , nous allons utiliser les systèmes embarqués , ces derniers Prennent une place de plus en plus importante dans notre société, ils servent à contrôler, réguler des dispositifs électroniques grâce à des capteurs embarqués dans des robots, des véhicules spatiaux, ...etc. Ces systèmes embarqués sont souvent utilisés par le public dans la vie de tous les jours sans même qu'on ne s'en rende compte, par exemple dans les systèmes de freinage d'une voiture, le pilot automatique d'un avion,. dans ce chapitre,nous allons faire une brève présentation sur les systèmes embarqués, puis on étudiera les bus de communication

I. Les systèmes embarqués :

Un système embarqué peut être défini comme un système électronique et informatique autonome dédié à une tâche bien précise et répondant souvent à des contraintes temps réel. Ses ressources disponibles sont généralement limitées, elles sont d'ordre spatial (taille limitée) et énergétique (consommation restreinte).

La majorité de ces systèmes, centralisés ou distribués sont critiques pour la sécurité, soit parce qu'ils sont au coeur du comportement du système, soit parce qu'ils interagissent avec l'être humain dans des situations critiques. Assurer leur fiabilité et leur sûreté de fonctionnement est un défi majeur.

La complexité croissante des systèmes embarqués nécessite des méthodes de conception globale qui tiennent compte des fonctionnalités et des constituants (capteurs, actionneurs, contrôleurs, réseaux), des perturbations et des fortes contraintes sur coût et l'environnement (retards, incertitudes,...).[3]

I.1 Historique :

Les premiers systèmes embarqués sont apparus en 1971 avec l'apparition de l'Intel 4004, développé en 1971. Ce premier microprocesseur, était le premier circuit intégré incorporant Tous les éléments d'un ordinateur dans un seul boîtier: unité de calcul, mémoire, contrôle des entrées / sorties. Alors qu'il fallait auparavant plusieurs circuits intégrés différents, chacun dédié à une tâche particulière, avec ce type de microprocesseur un seul pouvait assurer autant de travaux différents que possible. Très rapidement, des objets quotidiens tels que les télévisions et automobiles à moteur à injection électronique ne tardèrent pas à être équipés de microprocesseurs.

Ce sont alors les débuts de l'informatique embarquée.

I.2 Contraintes :

Les systèmes embarqués exécutent des tâches prédéfinies et ont un cahier des charges contraignant à remplir, qui peut être:

- **De coût :** Le prix de revient doit être le plus faible possible surtout s'il est produit en grande série.
- **D'espace mémoire:** ayant un espace mémoire limité de l'ordre de quelques Go maximum (bien que la taille vienne à être de moins en moins limitée grâce à la miniaturisation des

éléments). 11 convient de concevoir des systèmes embarqués qui répondent aux besoins au plus juste pour éviter un surcoût.

- **De puissance de calcul :** Il convient d'avoir la puissance de calcul juste nécessaire pour répondre aux besoins et aux contraintes temporelles de la tâche prédéfinie. Ceci en vue d'éviter un surcoût de l'appareil et une consommation excédentaire d'énergie (courant électrique).
- **D'autonomie :** la consommation énergétique doit être la plus faible possible, due à l'utilisation de batteries et/ou, de panneaux solaires voire de pile à combustible pour certains prototypes.
- **Temporel :** dont les temps d'exécution et l'échéance temporelle d'une tâche sont déterminés (les délais sont connus ou bornés *a priori*). Cette dernière contrainte fait que généralement de tels systèmes ont des propriétés temps réel.
- **De sûreté de fonctionnement :** Car s'il arrive que certains de ces systèmes embarqués subissent une défaillance, ils mettent des vies humaines en danger ou mettent en périls des investissements importants. Ils sont alors dits « critiques » et ne doivent jamais faillir. Il faut comprendre toujours donner des résultats justes, pertinents et ce dans les délais attendus par les utilisateurs.
- **De sécurité :** Ces systèmes peuvent se révéler être porteurs d'informations confidentielles pour leur(s) utilisateur(s), qu'il convient de conserver et de protéger.[3]

I.3-Caractéristiques :

- Plutôt que des systèmes universels effectuant plusieurs tâches, les systèmes embarqués sont étudiés pour effectuer des tâches précises. Certains doivent répondre à des contraintes de temps réel pour des raisons de fiabilité et de rentabilité. D'autres ayant peu de contraintes au niveau performances permettent de simplifier le système et de réduire les coûts de fabrication.
- Les systèmes embarqués ne sont pas toujours des modules indépendants. Le plus souvent ils sont intégrés dans le dispositif qu'ils contrôlent.
- Le logiciel créé pour les systèmes embarqués est appelé *firmware*. Il est stocké dans la mémoire en lecture seule ou dans la mémoire flash plutôt que dans un disque dur. Il fonctionne le plus souvent avec des ressources matérielles limitées : un petit écran et peu de mémoire ...etc.

Interface utilisateur : Certains systèmes embarqués peuvent ne pas avoir d'interface utilisateur (ils sont alors spécialisés dans une seule tâche). Mais cette interface peut également être similaire à celle d'un système d'exploitation d'ordinateur (par exemple un PDA).

Les systèmes les plus simples comportent uniquement des boutons, des LED.

-Les systèmes les plus complexes peuvent avoir un écran tactile ou encore un écran comportant des boutons de façon à minimiser l'espace. -La signification des boutons change selon l'écran et la sélection se fait naturellement en pointant la fonction désirée.

- Les ordinateurs de poche possèdent en général un bouton au style de « joystick » pour la navigation.

• Avec l'explosion du web, les fabricants de systèmes embarqués ont proposé une nouvelle option : une interface au style d'une page web sur une connexion au réseau. Cela permet d'éviter le coût d'un système sophistiqué tout en conservant une interface complète sur un autre ordinateur, quand cela est nécessaire. Interface couronnée de succès pour les installations permanentes à distance, les routeurs en particulier.

Fiabilité : Les systèmes embarqués sont la plupart du temps dans des machines qui doivent fonctionner en continu pendant de nombreuses années, sans erreurs et, dans certains cas, réparer eux-mêmes les erreurs quand elles arrivent. C'est pourquoi les logiciels sont toujours développés et testés avec plus d'attention que ceux pour les PC. Les pièces mobiles non fiables (par exemple les lecteurs de disques, boutons ou commutateurs) sont proscrites.

La question de la fiabilité peut inclure :

- Le système ne peut pas être éteint pour des réparations ou ce sont des réparations inaccessibles
- La solution peut être des pièces détachées supplémentaires ou un "mode mou" du logiciel qui fournit un fonctionnement partiel.
- Le système doit rester en marche pour des raisons de sécurité. Souvent, les sauvegardes sont effectuées par un opérateur.
- Dans ce cas, le « mode mou » est toléré.
- Un arrêt du système peut provoquer des pertes monétaires énormes s'il s'éteint.

Par exemple : les systèmes de ponts ou d'ascenseurs, les transferts de fond, les salles de bourse, les ventes ou services automatiques...

Domaines d'applications :

- Astronautique : fusée, satellite artificiel, sonde spatiale, etc.
 - Automate programmable industriel, contrôle-commande
- Electroménager : télévision, four à micro-ondes
- Environnement
 - Équipement médical
 - Guichet automatique bancaire (GAB)
 - impression : imprimante multifonctions, photocopieur, etc.
 - informatique : disque dur. Lecteur de disquette, etc.
 - Métrologie
 - Militaire : missile
 - Multimédia : console de jeux vidéo, assistant personnel
 - Télécommunication : Set-top box, téléphonie, routeur, pare-feu serveur de temps..etc
- Transport automobile , Aéronautique , Ferroviaire .etc..

II Bus de communication :

les éléments d'un système embarqué sont tous interconnectés à travers des bus, ce dernier est une collection de fils véhiculant des informations entre le processeur, mémoire et les périphériques ,On distingue en général trois bus:

- Le bus de données
 - Véhiculant les données
 - Bidirectionnel

- Le bus d'adresses
 - Véhiculant l'adresse de la location mémoire ou du périphérique
- Unidirectionnel
- Le bus de commande
 - Véhiculant la commande à effectuer (lecture, écriture)
- Fournit d'autres informations de contrôle
 - Horloge
 - Taille de l'information (octet, demi-mot, mot)

II.1-Normes de bus

bus n'est pas qu'une collection de fil mais il est défini par plusieurs caractéristiques

- Le nombre de bits du bus d'adresse
- Le nombre de bits du bus de données
- Le nombre de bits du bus de commande
- Règles d'accomplissement des transactions, Règles d'arbitrage
- Synchrone ou asynchrone
- Débit, Vitesse, ...etc.

Plusieurs normes de bus existent ayant chacune leurs propres caractéristiques

- ISA
- PCI
- AMBA
- ...etc

Tous les éléments (dont le processeur) doivent être compatibles avec la norme du bus

II-2-Constitution d'une liaison

La communication entre systèmes informatiques s'effectue grâce à des liaisons dont les principaux éléments sont définis par des normes dans le cadre du CCITT (Comité Consultatif International des Téléphones et Télégraphes). La Figure 1 montre ces éléments.

ETTD ETCD ETCD ETTD



Figure II.2 élément d'une liaison entre systèmes informatiques

ETTD : Equipement Terminal de Traitement de données (ou DTE : Data Terminal Equipment).

ETCD : Equipement de Terminaison de Circuit de Données (ou DCE : Data Communication Equipment).

L'ETTD est un élément qui ne se connecte pas directement à la ligne de transmission. Par exemple : un ordinateur, un terminal, une imprimante...

L'ETCD assure la transmission des données. Par exemple : un modem, un multiplexeur... Il a deux fonctions principales :

- l'adaptation du signal de l'ETTD à la ligne (codage et modulation en émission, démodulation et décodage en réception) ;
- la gestion de la liaison (établissement, maintien et libération de la

ligne).

II.21-Modes d'exploitation

Il existe trois modes d'exploitation d'une liaison. L'un est choisi en fonction de l'application.

- Simplex : Un système est émetteur, l'autre récepteur. C'est par exemple le cas des émissions radio ou télévision.
- Semi-duplex (half duplex) : Les deux systèmes peuvent être à la fois émetteur et récepteur, mais les émissions ne peuvent avoir lieu en même temps.
- Duplex (full duplex) : Les deux systèmes peuvent être à la fois émetteur et récepteur, les émissions pouvant avoir lieu en même temps.

II.3-Type de liaison

II.3-1 :Liaison filaire

a) bus RS485 :

EIA-485 (souvent appelée RS-485) est une norme qui définit les caractéristiques électriques de la couche physique d'une interface numérique série. La transmission se fait sur une ligne électrique, en pratique une paire torsadée, par des variations de tension en mode différentiel.

Citons, entre autres avantages de l'implémentation de cette norme, l'établissement de réseaux filaires de communications numériques sur de longues distances et/ou en environnement électrique bruité (milieu industriel, ...). De plus, la topologie réseau peut être de type point à point ou multipoints, le débit pouvant atteindre plusieurs Mbit/s, selon la longueur du bus (de 100 kbit/s sur ligne de 1200 m à 35 Mbit/s sur ligne de 10 m).

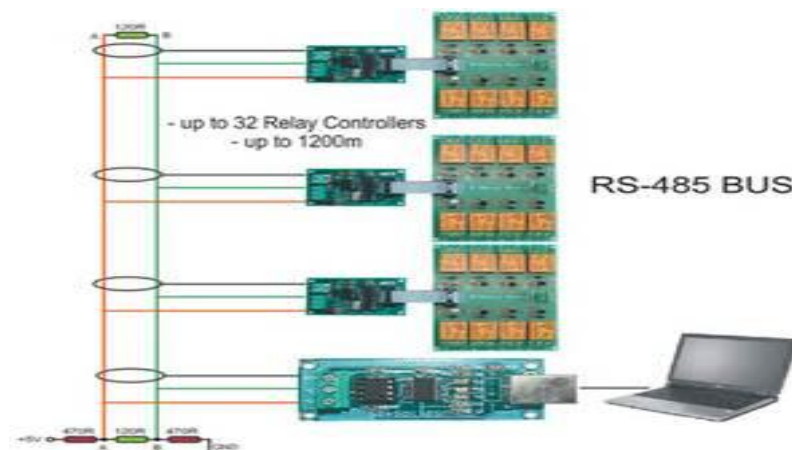


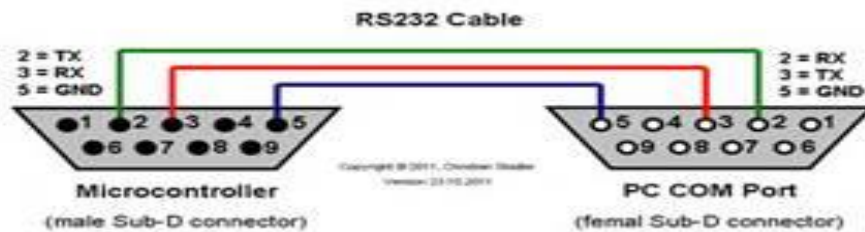
Figure II.2 RS485

b) Liaison RS-232

RS-232 (parfois appelée EIA RS-232, EIA 232 ou TIA 232) est une norme standardisant un bus de communication de type série sur trois fils minimum (électrique, mécanique et protocole). Disponible sur presque tous les PC depuis 1981 jusqu'au milieu des années 2000, il est communément appelé le « port série ». Sur les systèmes d'exploitation MS-DOS et Windows, les ports RS-232 sont désignés par les noms COM1, COM2, etc. Cela leur a valu le surnom de « ports COM », encore utilisé de nos jours. Cependant, il est de plus en plus remplacé par le port USB. Le standard RS-232 recouvre plusieurs autres standards : les

recommandations UIT-T V.24 (définition des circuits) et V.28 (caractéristiques électriques), ainsi que la norme ISO 2110 pour la connectique.

Les liaisons RS-232 sont fréquemment utilisées dans l'industrie pour connecter différents appareils électroniques (automate, appareil de mesure, etc.).



Figurell.3 RS232

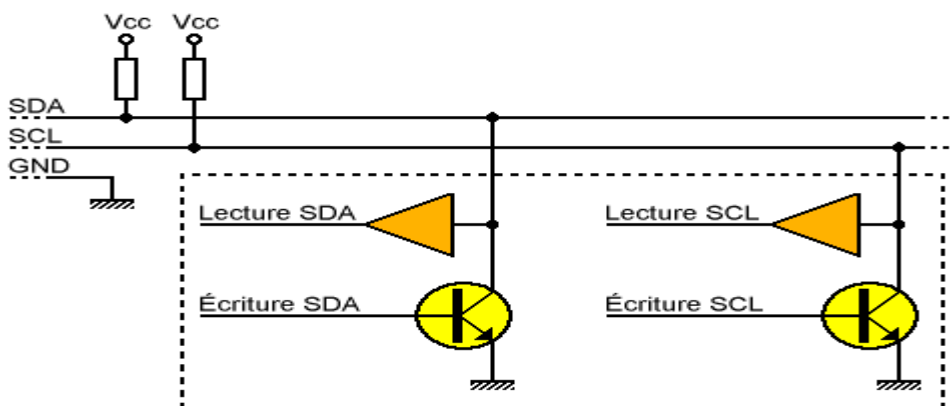
c) Liaison RS-422

EIA-422, appelé communément **RS-422**, est un standard technique qui spécifie les caractéristiques électriques d'un circuit de transmission numérique.[4]

d) Le bus I²C

I²C (signifie : Inter-Integrated Circuit, en anglais) est un [bus informatique](#) qui émergé de la « guerre des standards » lancée par les acteurs du monde électronique. Conçu par [Philips](#) pour les applications de [domotique](#) et d'électronique domestique, il permet de relier facilement un [microprocesseur](#) et différents circuits, notamment ceux d'une télévision moderne : récepteur de la télécommande, réglages des amplificateurs basses fréquences, [tuner](#), horloge, gestion de la prise [péritel](#), etc.

Il existe d'innombrables périphériques exploitant ce bus, il est même implémentable par logiciel dans n'importe quel [microcontrôleur](#). Le poids de l'industrie de l'électronique grand public a permis des prix très bas grâce à ces nombreux composants. Ce bus porte parfois le nom de TWI (Two Wire Interface) chez certains constructeurs.



Figurell.4 I2C

e) Le bus can

Le bus CAN est un bus de communication qui permet le dialogue entre plusieurs cartes. Contrairement à une liaison série classique, le bus CAN n'est pas une liaison émetteur /récepteur ou les cartes sont connectées une à une. Les cartes sont toutes connectées sur un même bus.

Les principales propriétés

Les principales propriétés de la structure du protocole du bus CAN sont :

- hiérarchisation des messages
- souplesse de configuration
- fonctionnement multi-maître
- détection et signalisation des erreurs
- retransmission automatique des messages altérés dès que le bus est à nouveau libre
- déconnexion automatique des noeuds défectueux

La norme CAN couvre les couches 1 et 2 du modèle OSI. Le bus CAN utilise une liaison physique différentiel

- Utilisé dans les réseaux embarqués car plus fiable qu' ethernet.
- Inventé par Bosch pour l'automobile.
- Débit de 1Mbps, trames de 8 octets au maximum, longueur maxi de 40m. Le débit décroît si la longueur augmente.
- Media : paire torsadée, fibre, infrarouge.
- Méthode d'accès au média : CSMA / CA : collision avoidance.
- CSMA / CA : 1 bit permet de couvrir la capacité du réseau. Les liaisons sont du type « wired-AND ». Si une machine A émet 001 et une machine B émet 011, on retrouve sur le réseau 001. B ne lit pas sur le réseau ce qu'il a émis et arrête donc d'émettre. A n'est pas perturbé et continue à émettre. Ceci rend CSMA/CA plus efficace que CSMA/CD d'ethernet.

Contrairement à ce dernier, le débit d'un réseau CAN ne s'écroule pas si le nombre de connexion importe.

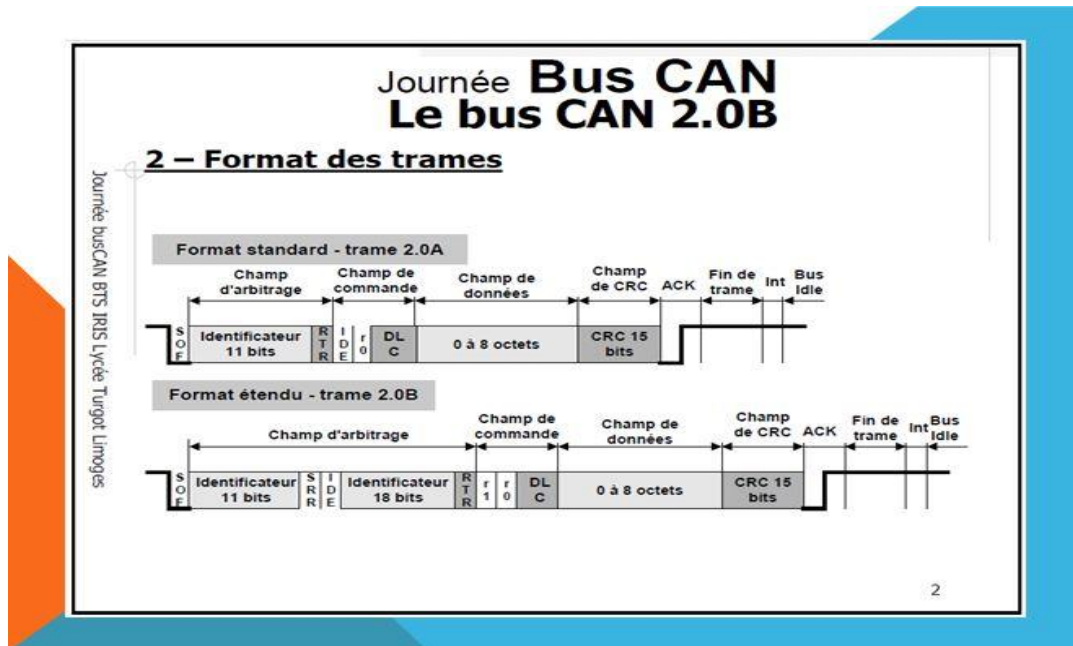


Figure II.5 bus CAN

TABLEAU COMPARATIF ENTRE LES PRINCIPAUX BUS FILAIRE

	RS232	RS423	RS422	RS485
Differential	no	no	yes	yes
Max number of drivers	1	1	1	32
Max number of receivers	1	10	10	32
Modes of operation	half duplex full duplex	half duplex	half duplex	half duplex
Network topology	point-to-point	multidrop	multidrop	multipoint
Max distance (acc. standard)	15 m	1200 m	1200 m	1200 m
Max speed at 12 m	20 kbs	100 kbs	10 Mbs	35 Mbs
Max speed at 1200 m	(1 kbs)	1 kbs	100 kbs	100 kbs
Max slew rate	30 V/ μ s	adjustable	n/a	n/a
Receiver input resistance	3..7 k Ω	≥ 4 k Ω	≥ 4 k Ω	≥ 12 k Ω
Driver load impedance	3..7 k Ω	$\geq 450 \Omega$	100 Ω	54 Ω
Receiver input sensitivity	± 3 V	± 200 mV	± 200 mV	± 200 mV
Receiver input range	± 15 V	± 12 V	± 10 V	-7..12 V
Max driver output voltage	± 25 V	± 6 V	± 6 V	-7..12 V
Min driver output voltage (with load)	± 5 V	± 3.6 V	± 2.0 V	± 1.5 V

Figure II.6 comparaison entre les bus filaire

II.3.2 liaison sans fils

a) ZigBee

La spécification initiale de ZigBee propose un protocole lent dont le rayon d'action est relativement faible, mais dont la fiabilité est assez élevée, le prix de revient faible et la consommation considérablement réduite.

On retrouve donc ce protocole dans des « environnements embarqués » où la consommation est un critère de sélection. Ainsi, la [domotique](#) et les nombreux capteurs et télécommandes qu'elle [implémente](#) apprécient particulièrement ce protocole en plein essor et dont la configuration du réseau maillé se fait automatiquement en fonction de l'ajout ou de la suppression de nœuds. On retrouve aussi ZigBee dans les contrôles industriels, les applications médicales, les détecteurs et d'[intrusion](#) et dans la télécommande de la freebox v6.

Les nœuds sont conçus pour fonctionner plusieurs mois (jusqu'à dix ans pour les plus économes) en autonomie complète grâce à une simple pile de 1,5 V.[5]



Figurell.7 ZigBee

b) bus LIN (Local Interconnect Network)

Le LIN (Local Interconnect Network) est la spécification pour un système de communication série qui est aussi appelé "**bus LIN**". Le LIN représente un système déterministe à prix avantageux pour la connexion de contrôleurs intégrant des capteurs, actionneurs et éléments de réglage intelligents. Basé sur un bus unifilaire, ce standard peut être classé parmi les bus de terrain. Le LIN a été développé notamment pour la communication avantageuse de capteurs et actionneurs intelligents dans les automobiles. Son utilisation pour les interconnexions dans une portière ou un siège auto est une application typique de ce genre de bus.

Le LIN (Local Interconnect Network) décrit un standard de communication avantageux pour véhicules pour l'électronique de confort, pour les capteurs et actionneurs intelligents et pour les composants de commande pour moteurs et transmissions à sécurité non critique. La communication est basée sur une ligne unifilaire en série par bit selon le format SCI (UART). La spécification correspond à la norme ISO 9141. La différenciation entre le maître et un ou

plusieurs esclaves permet d'éviter toute collision de messages sur la ligne de données (seul le maître pouvant déclencher une communication).

Le standard LIN est utilisé où que l'on n'ait pas besoin de toute la bande et polyvalence du CAN. Les spécifications LIN 1.0 et LIN 2.0 tiennent compte de formats uniformes pour la description d'un LIN complet et l'interface entre un LIN et l'application.

c) Bluetooth

C'est un standard de [communication](#) permettant l'échange bidirectionnel de données à très courte distance et utilisant des ondes radio [UHF](#). Son objectif est de simplifier les connexions entre les appareils électroniques en supprimant des liaisons filaires. Elle peut remplacer par exemple les câbles entre [ordinateurs](#), tablettes, téléphones mobiles entre eux ou avec des [imprimantes](#), [scanneurs](#), [claviers](#), [souris](#), manettes de [jeu](#) [vidéo](#), [téléphones portables](#), [PDA](#), systèmes et kits [mains libres](#) micro et/ou écouteurs, [autoradios](#), appareils photo numériques, lecteurs de [code-barres](#) et bornes publicitaires interactives.[6]



Figure II.8 Bluetooth

Tableau comparatif entre les principaux reseaux sans fils :

Tableau 1. Comparatif des 3 technologies sans fil les plus utilisées actuellement.

	Wi-Fi	Bluetooth	ZigBee
Bandes de fréquence	2,4 GHz	2,4 GHz	2,4 GHz, 868/915 MHz
Taille pile	~1 Mbit	~1 Mbit	~20 Kbits
Taux de données brut	11 Mbps	11 Mbps	250 kbps (2,4 GHz) 40 kbps (915 MHz) 20 kbps (868 MHz)
Nombre de canaux	11 - 14	79	16 (2,4 GHz) 10 (915 MHz) 1 (868 MHz)
Types de données	Numérique	Numérique, Audio	Numérique, Paires Key-Value
Portée inter-node	100 m	10_100 m	10_100 m
# de systèmes	32	8	255/65 535
Consommation	Moyenne – plusieurs heures sur 1 pile	Moyenne – plusieurs jours sur 1 pile	Très faible – plusieurs années sur 1 pile
Pénétration marché	Haute	Moyenne	Nulle
Architecture	Étoile	Étoile	Étoile, Arborescence, Cluster
Applications idéales	Internet dans bâtiments	PC et périphériques téléphoniques	Contrôle et monitoring à faible coût

Figure II.9 comparaison entre les bus sans fil

III-RS-485

III.1 Historique

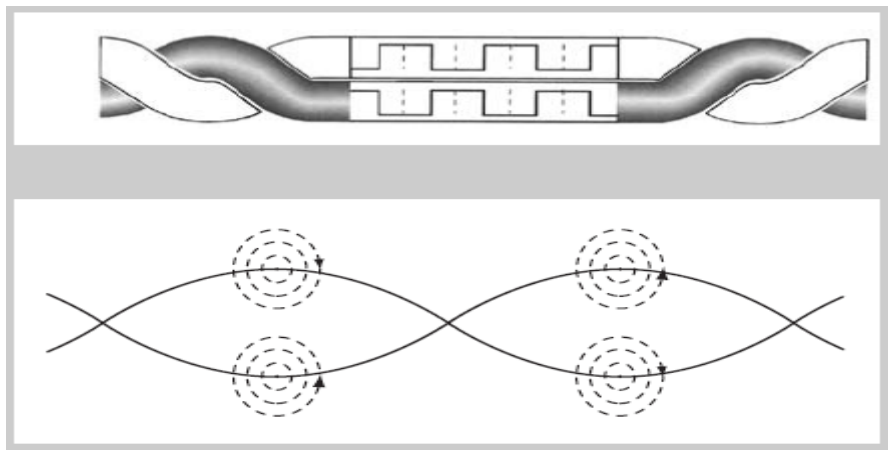
La norme RS485 définie par l'EIA, Electronic Industries Association, a été publiée en Avril 1983. Ce standard précise les caractéristiques électriques des émetteurs et des récepteurs pour une utilisation dans les systèmes multipoints en mode symétrique. La transmission se fait sur une ligne électrique, pratiquement une paire torsadée, par des variations de tension en mode différentiel[7]

III.2. Le Bus RS485 :

Le bus RS485 est un bus normalise par la norme EIA-485 et le guide d'application TSB-89-A utilisant une paire torsadée pour transmettre des données par des variations de tension en mode différentiel. Il permet une communication de type liaison série avec un maximum de 32

appareils, sur une distance allant jusqu'au kilometre selon la vitesse de transmission(peut atteindre 10 megabits/sec). La norme definit les trois points d'interface du generateur (lignes de signaux); «A», «B» et «C». Les donnees sont transmises sur "A" et "B". "C" est une reference de masse. L'etat logiques 1 (off) et 0 (on), est donne par la polarite entre les bornes A et B (differentiel de tension). Si A est negatif par rapport a B, l'etat binaire est a 1 .

Un des principaux problèmes des liaisons séries est l'absence d'immunité pour le bruit sur les lignes de signal. L'émetteur et le récepteur comparent les tensions par rapport à une masse commune en ligne (exemple RS232). Un changement dans le niveau du potentiel de terre peut avoir des effets désastreux. Le bruit limite à la fois la distance maximale et la vitesse de communication. Avec l'**RS485** il n'y a pas une masse commune comme signal de référence. La transmission est différentielle Le récepteur compare la *différence de tension* entre les deux lignes,



Figurell.30 tention les ligne de la liaison RS485

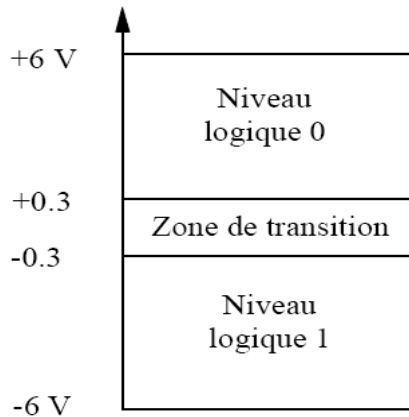
Le support de transmission est ici différentiel. Deux fils correspondant à des niveaux complémentaires sont donc utiles pour chaque signal ce qui limite l'influence des bruits extérieurs et des masses. Des circuits trois états permettent des liaisons multipoints.

III.3 Principales caractéristiques électriques de la norme RS485

Caractéristiques	Min.	Max.
Tension du générateur en circuit ouvert (VAB)	-	±6 V
Courant de sortie en court-circuit (IO)	-	150 mA
Résistance interne du générateur	50 Ω	100 Ω
Résistance de charge (RL)	100 Ω	-

Capacité parallèle (CL)	-	2500 pF
Vitesse	-	10 Mbit/s
Longueur de câble	-	1km

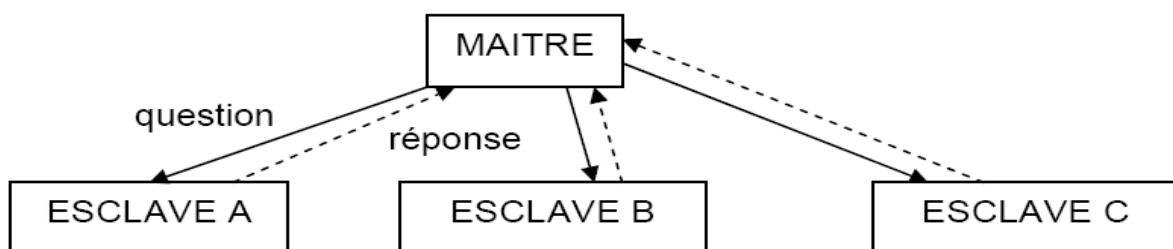
Niveaux significatifs de Tension :



[<http://www.technologuepro.com/cours-systemes-embarques>]

III.4 Le protocole :

Un protocole consiste en la définition de trames d'échange. Plusieurs protocoles en ont été définies. Le protocole le plus connu est le protocole **Modbus** (marque déposée par **MODICON**) ce dernier est un protocole de communication non-propriétaire, créé en 1979, utilisé pour des réseaux d'automates programmables, relevant du niveau 7 (applicatif) du Modèle OSI. C'est un protocole standard de dialogue basé sur une structure hiérarchisée entre un maître et plusieurs esclaves. .



- Le maître envoie une **demande** et attend une **réponse**.
- Deux esclaves ne peuvent dialoguer ensemble.
- Le dialogue maître – esclave peut être schématisé sous une forme successive de liaisons point à point.
- Mode de communication : **half-duplex**. (2 fils ou 4 fils)

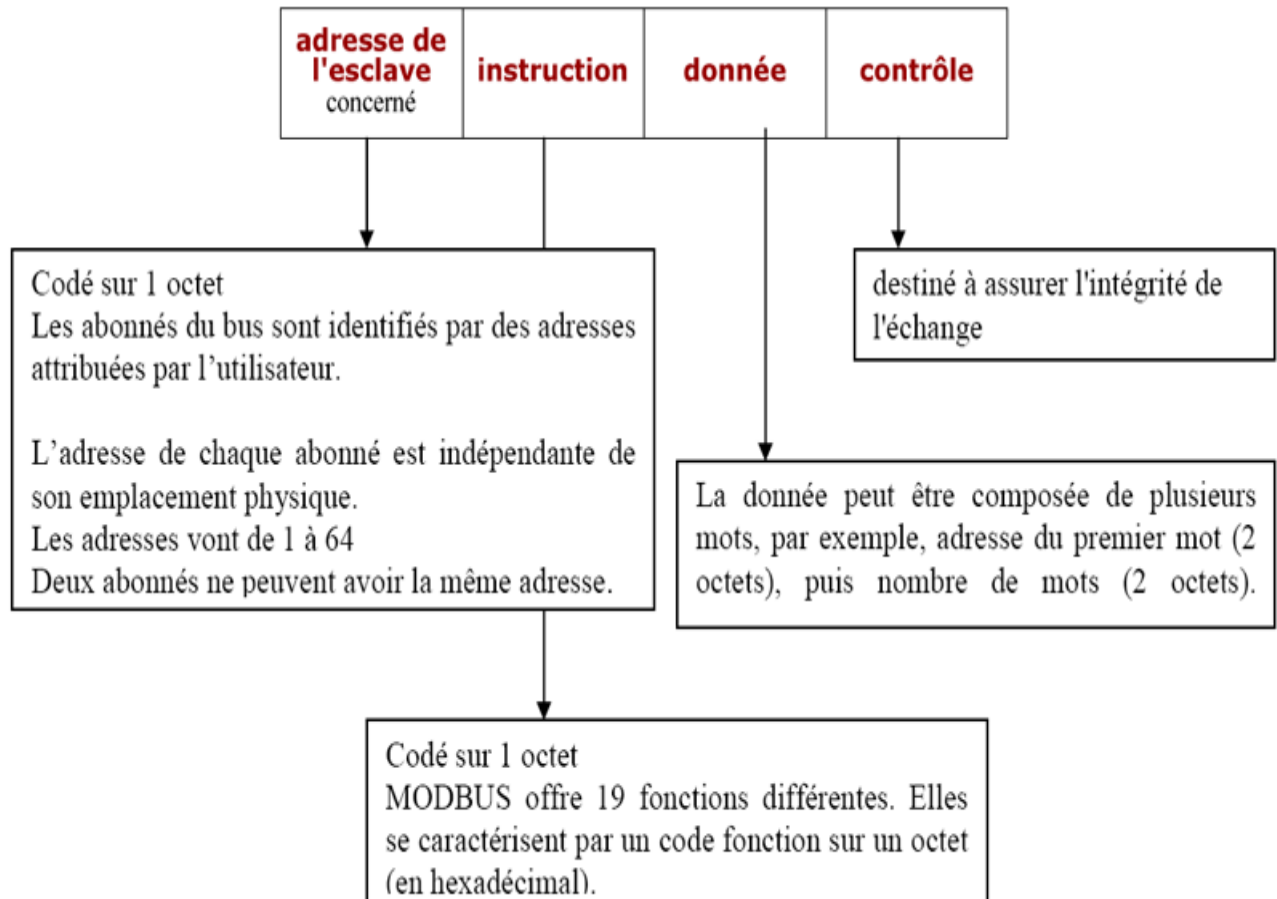
[Tapez le titre du document]

- le maître parle à l'ensemble des esclaves, sans attente de réponse (diffusion générale).

Il ne peut y avoir sur la ligne qu'un seul équipement en train d'émettre. Aucun esclave ne peut envoyer un message sans une demande préalable du maître. Le dialogue direct entre les esclaves est impossible.[<http://www.technologuepro.com/cours-systemes-embarques>]

III.4.1 structure du message

Le maître envoie un message constitué de la façon suivante:



Figurell.11 shema de la structure du message envoyer a l'aide de la liaison rs485

La détection de fin de message est réalisée sur un **silence** de durée supérieure ou égale à la durée de transmission de 3 octets.

L'esclave répond par un message du même type que le message question.

Le protocole Modbus est très utilisé pour les communications des automates programmables industriels et des variateurs de vitesse des moteurs électriques.[8]

III.4.2. Adressage

Les abonnés du bus sont identifiés par des adresses attribuées par l'utilisateur. L'adresse de chaque abonné est indépendante de son emplacement physique. Les adresses vont de 1 à 64 pour le protocole Modbus et de 1 à 255 en général et ne doivent pas obligatoirement être attribuées de manière séquentielle. Deux abonnés ne peuvent avoir la même adresse.

III.4.3. Echange maître vers 1 esclave

Le maître interroge un esclave de numéro unique sur le réseau et attend de la part de cet esclave une réponse.

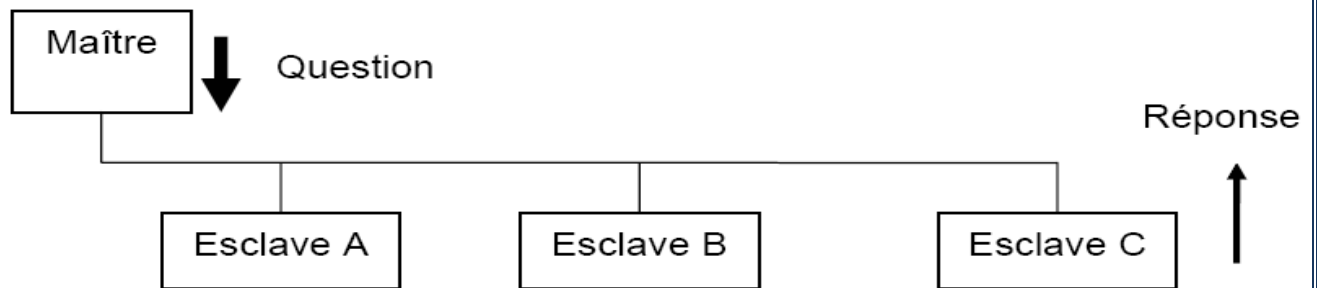


Figure II.12 échange de maître vers 1 esclave

III.4.4 Echange Maître vers tous les esclaves

Le maître diffuse un message à tous les esclaves présents sur le réseau, ceux-ci exécutent l'ordre du message sans émettre une réponse.

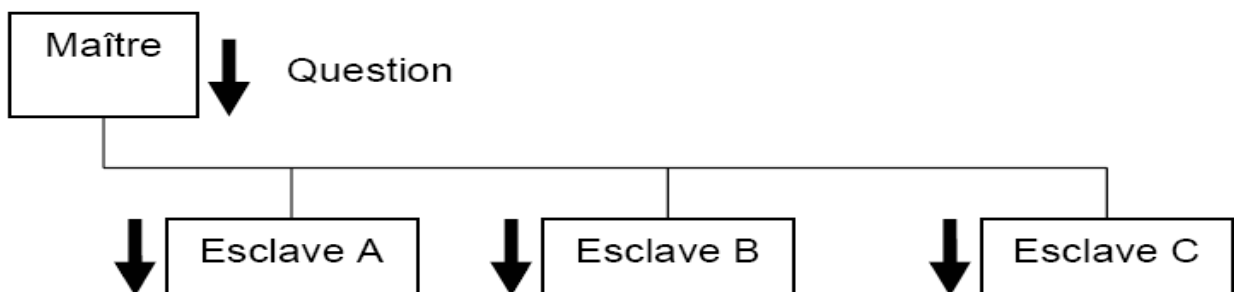


Figure II.13 échange de maître vers tout les esclaves

III.4.5 Format général d'une trame

Deux types de codage peuvent être utilisés pour communiquer sur un réseau Modbus. Tous les équipements présents sur le réseau doivent être configurés selon le même type.

- **Mode RTU (Unité terminale distante):**

La trame ne comporte ni octet d'en-tête de message, ni octets de fin de message. Sa définition est la suivante :

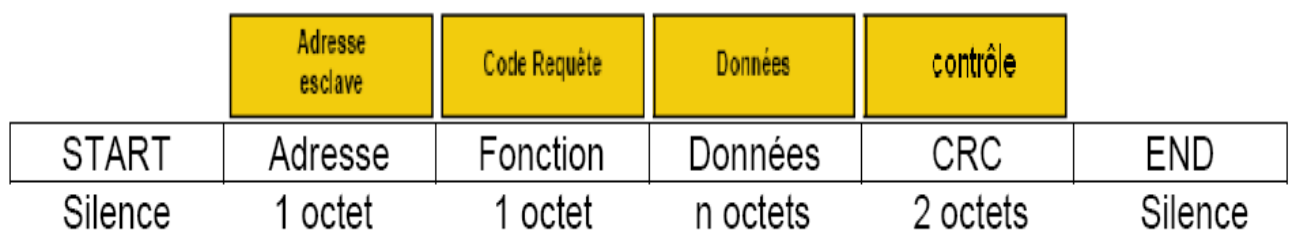


Figure II.13 mode RTU

CRC16 : paramètre de contrôle polynomial (cyclical redundancy check). La détection de fin de trame est réalisée sur un silence supérieur ou égal à 3 octets.

- **Mode ASCII :**

Chaque champ composant une trame est codé avec 2 caractères ASCII (2 fois 8 bits).

[Tapez le titre du document]

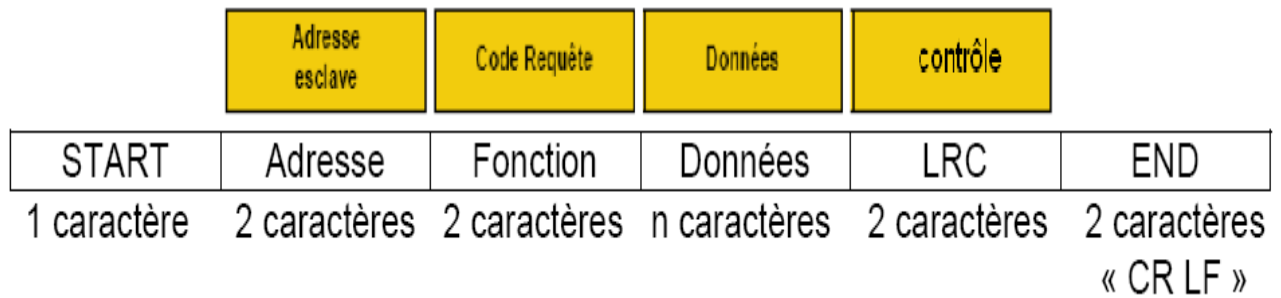


Figure II.15 mode ASCII

LRC : C'est la somme en hexadécimal modulo 256 du contenu de la trame hors délimiteurs, complémentée à 2 et transmise en ASCII.

III.4.6. Trame d'échange question/réponse

La question

Elle contient un code fonction indiquant à l'esclave adressé quel type d'action est demandé. Les données contiennent des informations complémentaires dont l'esclave a besoin pour exécuter cette fonction. Le champ octets de contrôle permet à l'esclave de s'assurer de l'intégralité du contenu de la question.

La réponse

Si une erreur apparaît, le code fonction est modifié pour indiquer que la réponse est une réponse d'erreur. Les données contiennent alors un code (code d'exception) permettant de connaître le type d'erreur. Le champ de contrôle permet au maître de confirmer que le message est valide. [7]

Conclusion

Dans ce chapitre nous avons introduit les systèmes embarqués, nous avons spécifier le bus RS485 que nous allons utilisé pour réalisé notre application qui consiste a gérer les files d'attente , Dans le chapitre suivant nous allons traité les deux carte « Raspberry pi et Arduino »

Chapitre 3

Raspberry pi & Arduino

Introduction

Dans le chapitre précédent nous avons traité les système embarqué que nous utiliseront pour gérer les files d'attente , dans ce chapitre s nous allons parlé sur les deux carte « Raspberry pi & Arduino » que nous allons utilisé dans notre application .

I. Carte électronique

Une carte électronique est un support plan, flexible ou rigide, généralement composé d'époxy ou de fibre de verre. Elle possède des pistes électriques disposées sur une, deux ou plusieurs couches (en surface et/ou en interne) qui permettent la mise en relation électrique des composants électroniques. Chaque piste relie tel composant à tel autre, de façon à créer un système électronique qui fonctionne et qui réalise les opérations demandées.

I.1 Microcontrôleur

Un micro-contrôleur est un petit processeur informatique relié à des entrées et des sorties numériques (0 ou 1) ou analogiques (tension variable). Il est capable de mémoriser et d'exécuter un programme visant à interpréter les entrées pour agir sur les sorties. Il se programme en général à l'aide d'un ordinateur mais peut fonctionner de manière autonome. En milieu industriel, les automates programmables qui servent à gérer et piloter des machines en sont une illustration. Ce type de machine intervient sur la commande d'un système mais ne peut délivrer beaucoup de puissance. Pour cela, on relaye ses commandes avec des transistors ou des relais. Il existe de nombreux types et tailles de micro-contrôleurs avec plus ou moins d'entrées /sorties et d'interfaces. À petite échelle, les PIC (produits par la société Microchip) sont des circuits intégrés bien connus des électroniciens depuis de nombreuses années. De nombreuses autres marques comme AT MEL, ST Microelectronics, Parallax ou Motorola produisent également des machines de ce type. Ils sont en général programmés en C/C++ et nécessitent des connaissances approfondies en électronique. La plate-forme Arduino a apporté une avancée majeure dans l'accessibilité de cette technologie au plus grand nombre par sa simplicité d'utilisation et son coût abordable. [9]

II-Arduino

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation.

Sans tout connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

Pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités de relations humain/machine ou environnement/machine.

Arduino est un projet en source ouverte : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions. [10]

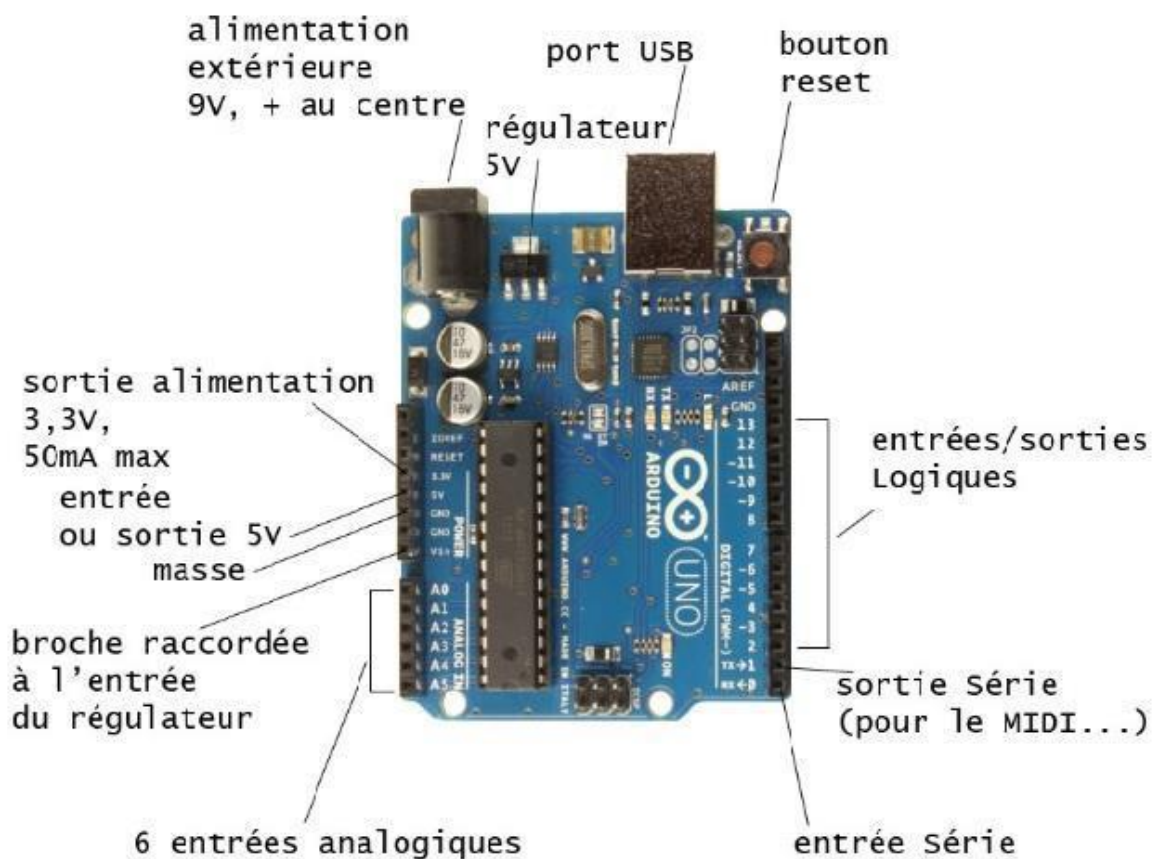


Figure III.1 Arduino

II.2 Matériel

La carte Arduino repose sur un circuit intégré (un mini ordinateur appelé également microcontrôleur) associée à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes :

- Côté entrées, des **capteurs** qui collectent des informations sur leur environnement comme la variation de température via une sonde thermique, le mouvement via un détecteur de présence ou un accéléromètre, le contact via un bouton-poussoir, etc.
- Côté sorties, des **actionneurs** qui agissent sur le monde physique telle une petite lampe qui produit de la lumière, un moteur qui actionne un bras articulé, etc.

Comme le logiciel Arduino, le circuit électronique de cette plaquette est libre et ses plans sont disponibles sur internet. On peut donc les étudier et créer des dérivés. Plusieurs constructeurs proposent ainsi différents modèles de circuits électroniques programmables et utilisables avec le logiciel Arduino .

Il existe plusieurs variétés de cartes Arduino (UNO , MEGA , DUE..etc)[10]

II.3 l'environnement de programmation

L'environnement de programmation Arduino (IDE en anglais) est une application écrite en Java inspirée du langage Processing [<http://processing.org/>]. L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte.

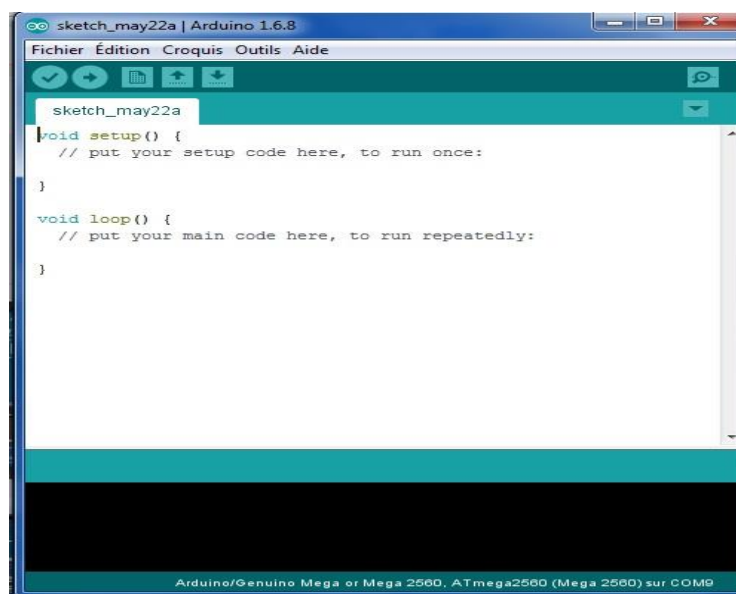


Figure III.2 Interface du logiciel ARDUINO

II.4 Domaine d'application de Arduino

Voici une liste non exhaustive des applications possibles réalisées grâce à Arduino :

- contrôler des appareils domestiques
- donner une "intelligence" à un robot
- réaliser des jeux de lumières
- permettre à un ordinateur de communiquer avec une carte électronique et différents capteurs
- télécommander un appareil mobile (modélisme)
- ..etc

II.5 Les avantages :

- Pas cher !
- Environnement de programmation clair et simple.
- Multiplateforme : tourne sous Windows, Macintosh et Linux.
- Nombreuses librairies disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso etc...)
- Existence de « shield » (boucliers en français) : ce sont des cartes supplémentaires qui se connectent sur le module Arduino pour augmenter les possibilités comme par exemple : afficheur graphique couleur, interface ethernet, GPS, etc...[11]

Par sa simplicité d'utilisation, Arduino est utilisé dans beaucoup d'applications comme l'électronique industrielle et embarquée, le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain ou le spectacle !

III. La Raspberry Pi



Figure III.3 Raspberry Pi 2

Le Raspberry Pi est un ordinateur dont les particularités sont la très petite taille (la taille d'une carte de crédit). Il a été créé par l'anglais David Braben, dans le cadre de sa fondation Raspberry Pi, dans le but d'encourager l'apprentissage de la programmation informatique. Pour la petite histoire, *raspberry* signifie *framboise* en anglais. Il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux et des logiciels compatibles. Il est fourni nu (carte mère seule, alimentation, sans boîtier, sans clavier, sans souris ni écran) dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

Son prix de vente était estimé à 25 \$, soit 19,09 €, début mai 2011. Les premiers exemplaires ont été mis en vente le 29 février 2012 pour environ 25 €. Début 2015, plus de cinq millions de Raspberry Pi ont été vendus. [8]

III.2 Historique :

III.2.1 Conception

En 2006, les premiers prototypes du Raspberry Pi sont développés sur des HDMI.[8] Ur3 500 £. [8]microcontrôleurs Atmel *ATmega 644*. Le schéma et le plan du circuit imprimé sont rendus publics. Cet ordinateur s'inspire du *BBC Micro* d'*Acorn Computer* (1981) et est destiné à encourager la jeunesse à la programmation. Le premier prototype ARM est intégré dans un boîtier de la même taille qu'une clé USB avec un port USB d'un côté et un port HDMI de l'autre.

III.2.2 Prototype

En août 2011, 50 cartes version Alpha sont construites, ces cartes étant identiques du point de vue fonctionnel au modèle B prévu mais elles sont plus grandes pour faciliter le débogage. Une démonstration montre la carte exécutant une distribution Debian avec un bureau LXDE, *Quake 3* en 1080p et une vidéo en Full HD MPEG-4 par En octobre 2011, une version de *RISC OS 5* tournant sur la carte est présentée. Après une année de développement la version grand public sera terminée en novembre 2012. En décembre 2011, 25 cartes modèle B ont été construites et testées. [14] Le design des cartes version Beta est le même que les cartes grand public. Une seule erreur a été découverte dans le design, certaines broches du CPU ne fonctionnaient pas correctement, l'erreur a été corrigée avant la première production.

La première semaine de l'année 2012, 10 premières cartes sont mises aux enchères sur eBay. L'une est achetée anonymement et donnée au Centre « for Computing History », dans le *Suffolk* en Angleterre. Les 10 cartes qui représentaient un prix de 220 £ ont été vendues pour un total de 16 000 £. La carte possédant le numéro de série 01 est achetée po

III.2.3 Lancement

La première série de 10 000 cartes est produite en Taïwan et en Chine. Les livraisons de la première série sont annoncées pour mars 2012 en raison de l'installation d'un mauvais connecteur Ethernet, mais la fondation annonce qu'elle s'attend à augmenter la production des futures séries sans difficulté.

Les ventes débutent le 29 février 2012 à 06:00 UTC. Au même moment est annoncé un modèle A à 256 MB de RAM au lieu des 128 MB prévus. Le site web de la fondation affiche : « *Six ans après le début du projet, nous sommes presque à la fin de la première session de développement - cependant ce n'est que le début de l'histoire de Raspberry Pi.* ». [12]

En septembre 2012, 500 000 cartes ont été vendues. Le 6 avril 2012, la fondation annonce que le Raspberry Pi a obtenu la certification CE, demandée par les distributeurs pour pouvoir lancer la distribution auprès des premiers acheteurs. [13]

Au 22 mai 2012, 20 000 cartes ont été envoyées. En juillet 2012, 4 000 unités sont produites chaque jour.

En septembre 2012, la Fondation Raspberry Pi annonce une deuxième révision du modèle B. De plus, les futures séries seront fabriquées au Royaume-Uni, dans les usines Sony de *Pencoed*, au *Pays de Galles*. Il est estimé que 30 000 unités seraient produites par mois, créant 30 emplois. la fréquence du processeur est passé de 700 à 1 000 MHz.

En octobre 2013, c'est un million de Raspberry Pi qui a été produits au Royaume-Uni. Le deux millionième kit est envoyé entre le 24 et le 31 octobre.

En avril 2014, une nouvelle version est annoncée, elle divise la carte en deux parties : une partie calcul et une partie interface d'entrées-sorties. La partie calcul *Compute Module* comporte 512 Mio de mémoire vive et 4 Gio de mémoire flash. La dimension de la carte est réduite au format SO-DIMM (environ 68 × 30 mm). La partie interface d'entrées-sorties *Compute Module IO Board* comporte des connecteurs HDMI et USB. Au mois de juin, trois millions de Raspberry Pi ont été vendus. [14]

III.3 Architecture matérielle :

Le Raspberry Pi possède un processeur ARM11 à 700 MHz. Il inclut 1, 2 ou 4 ports USB, un port RJ45 et 256 Mo de mémoire vive pour le modèle d'origine (512 Mo sur les dernières versions). Son circuit graphique BMC Videocore 4 en particulier permet de décoder des flux Blu-Ray full HD (1080p 30 images par seconde), d'émuler d'anciennes consoles et d'exécuter des jeux vidéo relativement récents. [12]

III.4 Spécifications :

Modèle Pi 2 :

Le 2 février 2015, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 2, plus puissant, il est équipé d'un processeur Broadcom BCM2836, quatre cœurs ARMv7 à 900 MHz, accompagné de 1 Go de RAM.

Il possède les mêmes dimensions et connectiques que le modèle B+.

III.5 Équipement supplémentaire :

De base, la Raspberry-Pi est fourni sans boîtier, alimentation, clavier, souris ni écran.

Ceci dans le but de minimiser les coûts et de pouvoir recycler d'autres matériels. Des boîtiers adaptés à la carte mère (dont certains originaux) sont disponibles sur la plupart des sites commercialisant la Raspberry-Pi. Un chargeur de Smartphone (micro-USB vers USB) convient parfaitement pour alimenter la carte. En utilisant la prise USB, on peut connecter de nombreux périphériques : disques durs externes, lecteur Blue-ray, clé wifi, webcam, imprimante...

Il est possible de connecter la Raspberry-Pi avec :

- une carte réseau Ethernet ou Wi-Fi, pour établir une connexion en réseau, notamment avec Internet pour le modèle A. Le modèle B introduit durant l'été 2011 comporte une carte Ethernet intégrée.
- Des unités de stockage externe, clés USB ou disques durs alimentés séparément.
- Tout autre périphérique USB disposant d'un pilote informatique compatible : clé Bluetooth, clé Wi-Fi, mémoire flash, imprimante, webcam[12]

III.6 Les systèmes d'exploitation compatibles avec la Raspberry-Pi:

Les systèmes d'exploitation compatibles Raspberry Pi / ARM incluent : [15]

- Diverses distributions Linux - La plateforme par défaut du Raspberry Pi ;
 - Debian « wheezy » est recommandée par la fondation Raspberry Pi avec sa version dédiée *Raspbian* (Raspberry & Debian). L'image est disponible sur le site officiel de la fondation.
 - Fedora « Raspberry Pi Fedora remix ». L'image est disponible sur le site officiel de la fondation.
 - Arch Linux est fonctionnel avec sa version ARM pour Raspberry Pi. L'image est disponible sur le site officiel de la fondation.
 - Gentoo est utilisable classiquement.
 - Slackware est également utilisable classiquement.
 - Suse est fonctionnelle avec sa version ARM pour Raspberry Pi. L'image est disponible sur le site officiel dédié à cette carte.
 - Kali Linux, un système dédié au pentest (Évolution de BackTrack).

III Conclusion :

Après avoir présenté les deux carte Arduino et Raspberry-Pi que nous avons contribuer pour réalisé notre système qui consiste a gérer plusieurs files d'attentes dans le chapitre suivant nous allons entamé la conception de notre système ou nous spécifions les différente étape que nous allons utiliser pour la réalisation de notre système

Chapitre 4

Analyse et conception

Introduction

Dans le chapitre précédent, nous avons présenté une vue générale du système embarqué ainsi que quelques notions des files d'attente, ou avons parlé de la Raspberry pi et Arduino, nous passerons à travers ce chapitre à décrire la conception de notre système qui consiste à la réalisation d'un système de gestion de files d'attente, pour décrire notre conception, en premier lieu nous donnons une analyse du fonctionnement de notre système, par la suite nous détaillerons à l'aide des diagrammes d'appel de flux de données et de tâches.

I. Analyse et Conception

L'analyse et la conception sont primordiales dans tout système embarqué, elles doivent être traitées avec rigueur et précision, car elles constituent la base du système à développer.

Avant de s'engager dans la conception, il est impératif de passer par la phase d'analyse. La conception vient par la suite pour s'appuyer sur les résultats de la phase d'analyse. Ce faisant, nous allons nous inspirer d'une méthode plus récente destinée spécialement aux systèmes embarqués faite par M. Jonathan Valvano (professeur à l'université de Texas) afin de mieux expliquer les caractéristiques structurelles et comportementales de notre système.

Avant d'aborder les deux parties d'analyse et de conception, la section suivante précise la notion du cycle de vie d'un produit.

I.1 Cycle de vie d'un produit

Le cycle de vie d'un produit est généralement illustré comme une série d'étapes qui retracent toutes les phases que traverse un produit depuis son analyse jusqu'à son implémentation (validation) en passant par la conception et le développement.

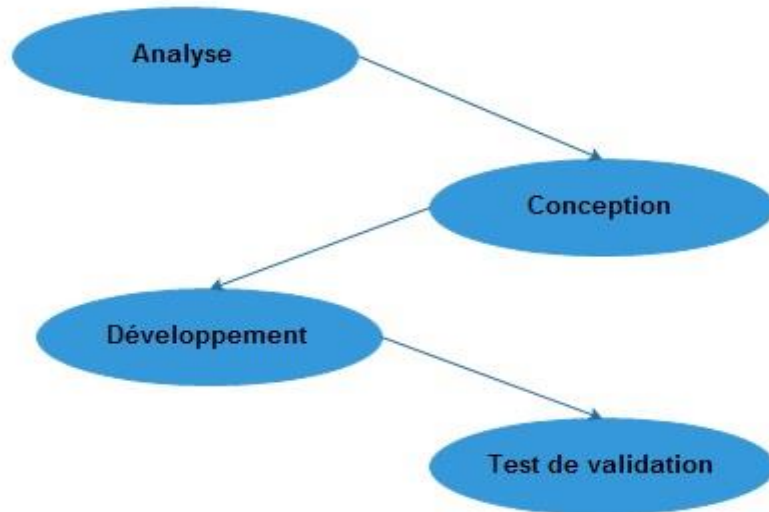


Figure VI.1 Cycle de vie d'un produit

Objectifs visés

L'objectif principal de ce chapitre est de faire une analyse de notre système, suivi d'une conception qui spécifie les différentes étapes de notre système.

II.Etape Analyse

II.1 Architecture générale de notre application

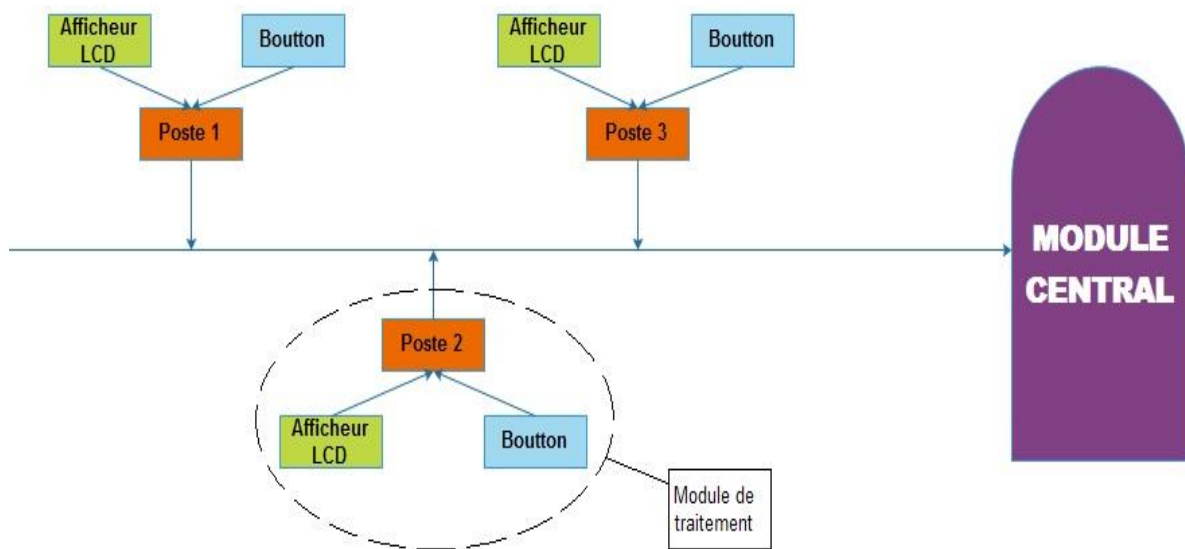


Figure VI.2 Architecture général de notre application.

II.2 Fonctionnement du système

- **Le module central** : c'est une Raspberry Pi, joue le rôle du serveur, sa tâche est de :
 - Demander aux Arduinos l'état du bouton.
 - D'incrémenter la base de données.
- **RS485** : c'est le bus de communication qui permet de relier les Arduinos(module de traitement) à la Raspberry Pi .
- **Base de données** : est incluse dans le module central contenant des données de la file d'attente.
- **L'Ecran principal** affiche les données suivantes :
 - Le numéro de ticket qui est demandé au guichet.
 - Le numéro du guichet libre.
 - Le temps moyen d'attente d'un client.
- **Les postes (Arduino)** : récupérer l'état du bouton et l'envoyer quand le module central le demande.
- **Application web** : elle permet à un client à distance de consulter le dernier numéro appelé de la file d'attente et le temps moyen restant, et aussi prendre un rendez-vous.

III. Conception

III.1 Module de traitement

III.1.1 Diagramme d'appel

Le diagramme d'appel représente la structure globale hiérarchique descendante du système et le mécanisme d'accès aux données par l'interconnexion directe entre l'ensemble (matériel/logiciel).

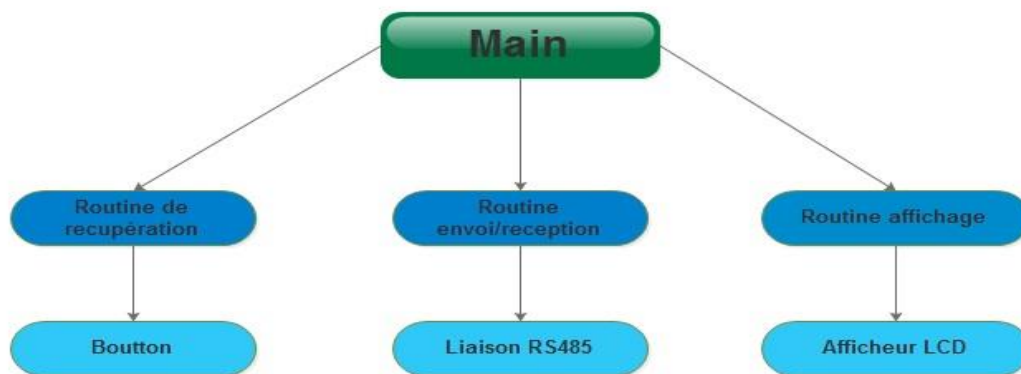


Figure VI.3 Diagramme d'appel du module de traitement

III.1.2 Diagramme du flux de données

Un graphe de flux de données illustre le fonctionnement global du système tout en cachant les détails, il montre la circulation du flux interne des données de la source à la destination.

Les représentations graphiques logiciel/matériel du diagramme du flux de données sont pareilles avec le diagramme d'appel sauf qu'ici les flèches représentent le sens de circulation des données.

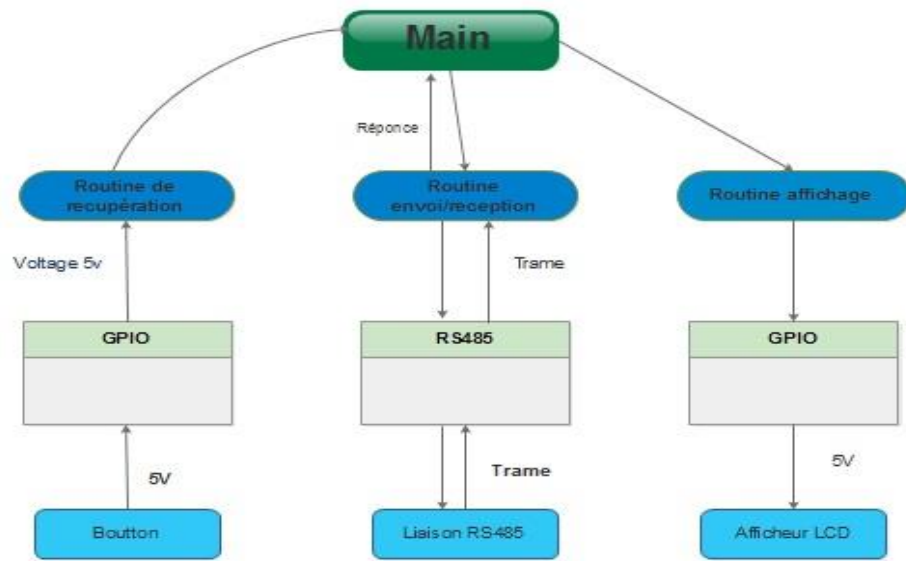


Figure VI 4 Diagramme de flux de donnée module de traitement

III.1.3 Diagramme de tâches

Un diagramme de tâche est une représentation comportementale de la partie logicielle d'un système. Il présente le déroulement séquentiel et les différentes routines qu'effectue le système pour réaliser ses tâches et cela par la décomposition successive d'une tâche en sous tâche pour atteindre une tâche plus simple. La décomposition d'une tâche se fait par l'une de ces méthodes qu'elles soient séquentielles, conditionnelles, itératives ou une interruption.

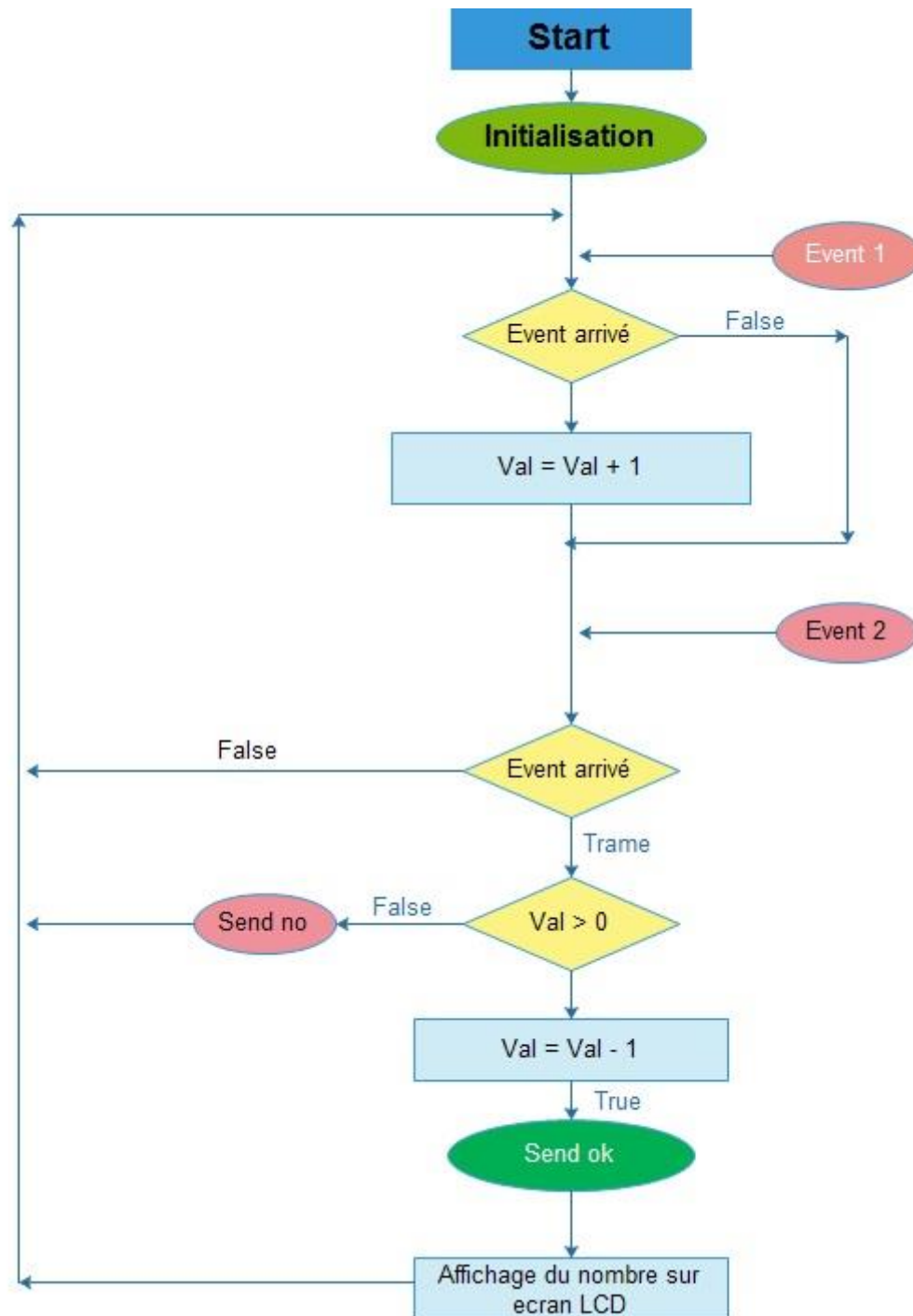


Figure VI.5 Diagramme d' de tâche du module de traitement

Description

Event 1 : l'appui sur le bouton.

Event 2 : la réception d'une trame du module central, elle contient un @ qui doit être égal à l'@ de l'Arduino.

Val : est une variable incrémentée à l'appui du bouton et décrémentée à l'envoi du message OK.

N.B : est le numéro du ticket pour appeler au poste.

La trame ressue du module central contient le N.B.

III.2 Module central

III.2.1 Diagramme d'appel

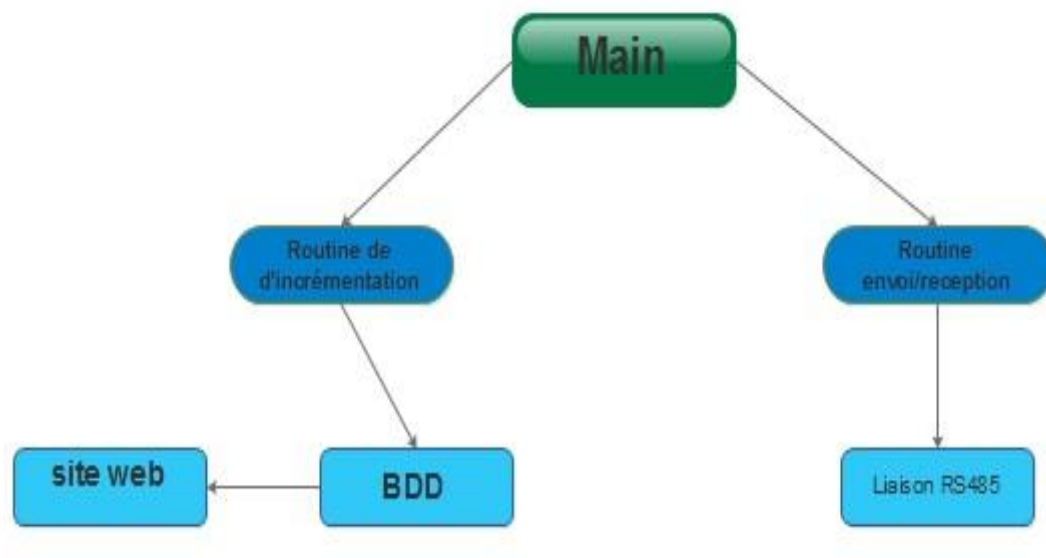


Figure VI.6 Diagramme d'appel du module central

III.2.2 Diagramme du flux de données

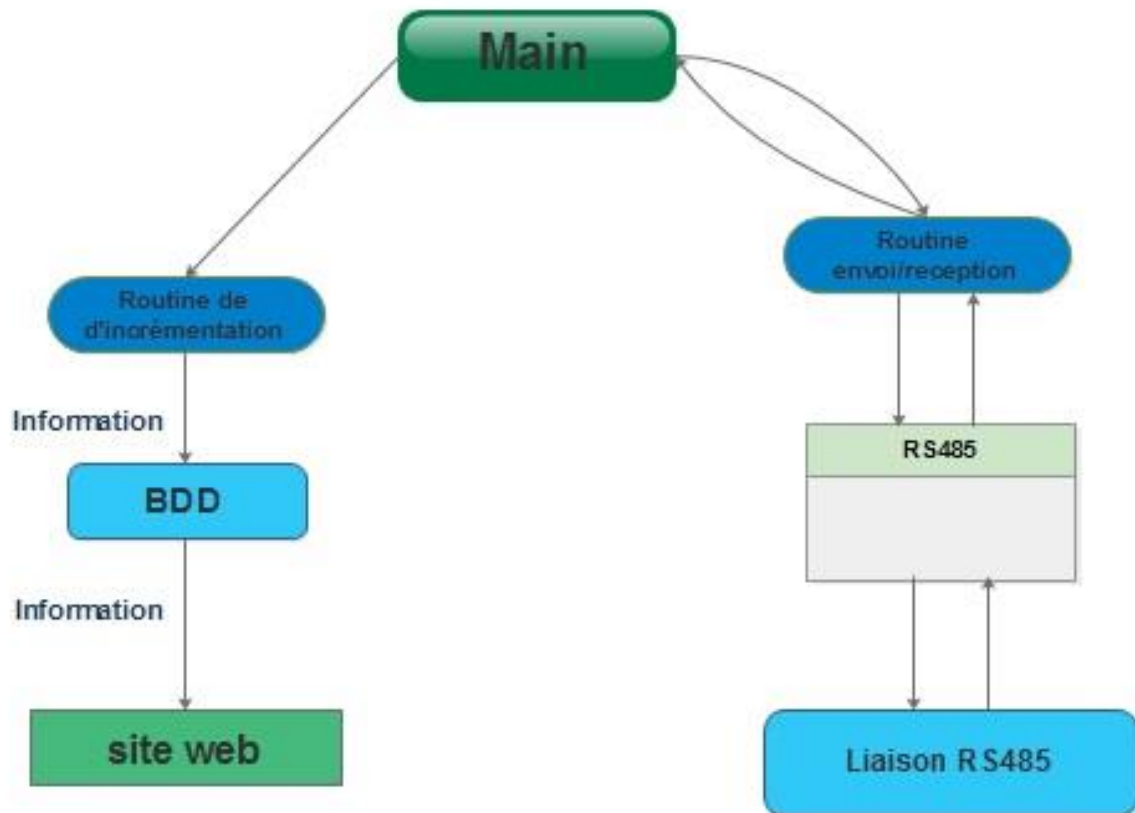


Figure VI 7 Diagramme de flux de donnée du module central

III.2.3 Diagramme de tâches



Figure VI.8 Diagramme de tâche du module central

Description

Event 1: la réponse de l'Arduino qui a l'@ = val soit un **OK** soit un **NO**

Val : est une variable qui contient l'@ de l'Arduino à qui envoyer le message .

Conclusion :

La phase de conception était importante pour pouvoir visualiser le fonctionnement de notre système qui consiste a la réalisation d'un système de gestion des files d'attente d'une façon abstraite , dans le chapitre qui suit nous passons a la réalisation de notre système qu'on a utilisé pour une agence intitulé « CNAS », nous détaillerons les étape mise en œuvre ,nous présenterons les différents logiciels utilisés, leurs installationset leurs configurations afin de mettre de notre système en service .

Chapitre5

Réalisation & mise en œuvre

Introduction

Ce présent chapitre a pour objectif de présenter la réalisation de notre système de gestion de files d'attente. Nous entamons ce chapitre par la présentation de l'environnement de développement dont le matériel, et les outils logiciels utilisés. Ensuite, nous allons présenter les fonctions de nos applications.

I Description de l'environnement du développement :

I.1 Matériels :

I.1.1 Arduino :

Une carte électronique est un support plan, flexible ou rigide, généralement composé d'époxy ou de fibre de verre. Elle possède des pistes électriques disposées sur une, deux ou plusieurs couches (en surface et/ou en interne) qui permettent la mise en relation électrique des composants électroniques. Chaque piste relie tel composant à tel autre, de façon à créer un système électronique qui fonctionne et qui réalise les opérations demandées.

I.1.2 Raspberry Pi :

Nous avons utilisé pour la réalisation de notre application le **Raspberry Pi** modèle B+ sur lequel est installé le système d'exploitation **Linux(Raspbian)**. Ce modèle est composé d'une carte mère.

✓ Caractéristiques techniques :

- Carte mère Raspberry Pi Type B2
- Processeur ARM1176JZF-S core (ARM11) de 700 MHz
- Mémoire RAM intégrée de 512 Mo
- Contrôleur graphique Broadcom VideoCore IV
- Ports disponibles : 4 x USB, 1 x HDMI, 1 x RJ45
- Lecteur de cartes mémoire microSD
- Sortie audio Jack (3.5 mm)
- Puissance : 330 mA
- GPIO 40 broches
- Alimentation

✓ **Les GPIO :**

C'est l'abréviation de General Purpose Input/Output, ou plus simplement entrées/sorties à usage général. Ces entrées sorties permettent d'étendre les fonctionnalités du Raspberry Pi en lui donnant la possibilité d'agir sur des leds ou des afficheurs LCD par exemple, lire l'état d'un interrupteur, d'un capteur...etc. Dans notre cas, nous avons utilisé ces GPIO pour la commande des différents éléments de notre centrale domotique. Ce connecteur GPIO dispose de différents types de connexion :

- des broches(Pins) utilisables en entrée ou sortie numérique.
- des broches pour une interface I2C (permettant de se connecter sur du matériel en utilisant uniquement 2 broches/pins de contrôle.)
- une interface SPI pour les périphériques SPI,
- les broches Rx et Tx pour la communication avec les périphériques séries.
- de broches pouvant être utilisé en PWM ("Pulse Width Modulation") permettant le contrôle de puissance ou PPM ("Pulse Position Modulation") permettant de contrôler des servo moteurs par exemple. [16]

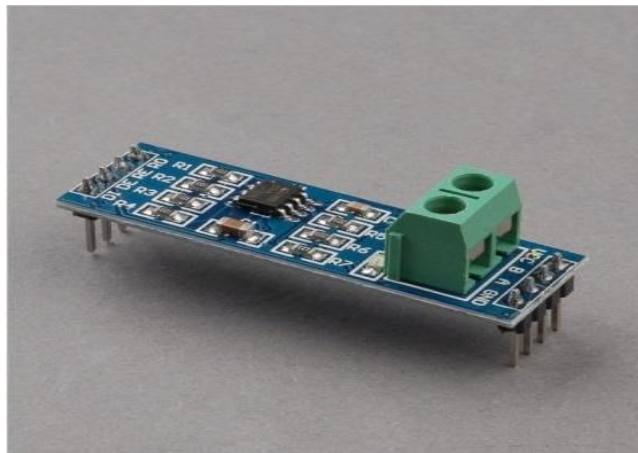
Raspberry Pi B2			
Pin#	NAME		NAME Pin#
01	3.3v DC Power		DC Power 5v 02
03	GPIO02 (SDA1 , I2C)		DC Power 5v 04
05	GPIO03 (SCL1 , I2C)		Ground 06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14 08
09	Ground		(RXD0) GPIO15 10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18 12
13	GPIO27 (GPIO_GEN2)		Ground 14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23 16
17	3.3v DC Power		(GPIO_GEN5) GPIO24 18
19	GPIO10 (SPI_MOSI)		Ground 20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25 22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08 24
25	Ground		(SPI_CE1_N) GPIO07 26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC 28
29	GPIO05		Ground 30
31	GPIO06		GPIO12 32
33	GPIO13		Ground 34
35	GPIO19		GPIO16 36
37	GPIO26		GPIO20 38
39	Ground		GPIO21 40

FigureV.1GPIO de la Raspberry PI

I.1.3 MAX485 :

MAX485 est un émetteur-récepteur à faible puissance et la vitesse de montée limitée utilisée pour communication RS-485 . Il fonctionne à une seule alimentation + 5V et le courant nominal est de 300 μ A . L'adoption de la communication semi-duplex pour mettre en œuvre la fonction de conversion de niveau TTL en RS-485 niveau , il peut atteindre une vitesse de transmission maximale de 2.5Mbps . MAX485 émetteur-récepteur consomme un courant d'alimentation comprise entre 120 μ A et 500 μ A dans les conditions non chargées ou complètement chargées lorsque le conducteur est désactivé. Le pilote est limité pour le courant de court-circuit et les sorties du pilote peut être placé à un état de haute impédance à travers le circuit d'arrêt thermique. L' entrée du récepteur dispose d'une fonction fail-safe qui garantit la logique haut rendement si l'entrée est en circuit ouvert . En outre, il dispose d'une solide performance anti- interférences .

[17]



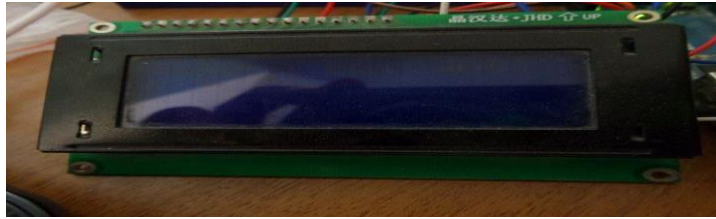
FigureV.4MAX485

I.1.4 Ecran LCD

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité.

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétroéclairage

de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (de 80 à 250 mA).



FigureV. 5Ecran LCD

Ils sont très utilisés dans les montages à microcontrôleur, et permettent une grande convivialité. Ils peuvent aussi être utilisés lors de la phase de développement d'un programme, car on peut facilement y afficher les valeurs de différentes variables.

Commande d'un afficheur LCD

Deux modes de fonctionnement de l'afficheur sont disponibles, le mode 4 bits et le mode 8 bits, modes que l'on choisira à l'initialisation de l'afficheur. (dans notre cas on utilise les commandes a 4bits)

Mode 8 bits : Dans ce mode 8 bits, les données sont envoyées à l'afficheur sur les broches **D0** à **D7**. On place la ligne **RS** à 0 ou à 1 selon que l'on désire transmettre une commande ou une donnée. Il faut aussi placer la ligne **R/W** à 0 pour indiquer à l'afficheur que l'on désire effectuer une écriture. Il reste à envoyer une impulsion d'au moins 450 ns sur l'entrée **E**, pour indiquer que des données valides sont présentes sur les broches **D0** à **D7**. L'afficheur lira la donnée sur le front descendant de cette entrée.

Si on désire au contraire effectuer une lecture, la procédure est identique, mais on place cette fois la ligne **R/W** à 1 pour demander une lecture. Les données seront valide sur les lignes D0 à D7 lors de l'état haut de la ligne **E**.

Mode 4 bits :

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches d'entrées sorties disponibles sur un microcontrôleur. Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur LCD. Dans ce mode, seuls les 4 bits de poids fort (**D4** à **D7**) de l'afficheur sont utilisées pour transmettre les données et les lire. Les 4 bits de poids faible

(**D0** à **D3**) sont alors connectés à la masse. On a donc besoin, hors alimentation de sept fils pour commander l'afficheur. Les données sont alors écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible. Une impulsion positive d'au moins 450 ns doit être envoyée sur la ligne E pour valider chaque demi-octet ou nibble.

Dans les deux modes, on peut, après chaque action sur l'afficheur, vérifier que celui-ci est en mesure de traiter l'information suivante. Pour cela, il faut demander une lecture en mode commande, et tester le flag Busy **BF**. Lorsque BF=0, l'afficheur est prêt à recevoir une nouvelle commande ou donnée.

Il se peut qu'on dispose encore de moins de broches disponibles dans l'application envisagée. Dans ce cas, on peut alors relier la ligne **R/W** à la masse de façon à forcer l'afficheur en écriture. On a alors besoin, hors alimentation de seulement six fils en mode 4 bits, et dix fils en mode 8 bits, pour commander l'afficheur, mais on ne peut alors plus relire l'afficheur. Ceci n'est pas gênant dans la mesure où on sait ce qu'on a écrit sur l'afficheur, mais on ne peut alors plus relire le flag Busy. Il faut alors utiliser des temporisations après chaque écriture sur l'afficheur. On perd alors un peu en temps d'affichage, mais on gagne une broche d'entrée sortie.[18]

I.2 Logiciels :

I.2.1 Le serveur web Apache2 :

Apache2 est un logiciel qui permet de mettre à disposition sur le réseau, d'un site web (pages html, PHP). Plus précisément apache2 est un serveur http. Les utilisateurs utilisent quant à eux un client http pour afficher à l'écran ce site, comme par exemple Firefox, chrome, Microsoft Internet Explorer etc...

I.2.2 SQLite3 :

SQLite est un système de gestion de base de données embarqué qui a la particularité de fonctionner sans serveur. On peut l'utiliser avec beaucoup de langages : PHP, Python, Java, C/C++, Delphi, Ruby...

L'intérêt est qu'il est très léger et rapide à mettre en place, on peut s'en servir aussi bien pour stocker des données dans une vraie base de données sur une application pour smartphone (iPhone ou Android), pour une application Windows, ou sur un serveur web. Une base de données SQLite est bien plus performante et facile à utiliser que de stocker les données dans

des fichiers XML ou binaires, d'ailleurs ces performances sont même comparables aux autres SGBD fonctionnant avec un serveur comme MySQL, Microsoft SQL Server ou PostgreSQL. [19]

I.2.3 Adobe Dreamweaver CC 2014:

Adobe Dreamweaver est un éditeur HTML professionnel destiné à la conception, au codage et au développement de sites, de pages et d'applications web. Il constitue aujourd'hui une plateforme de développement d'applications internet de haut niveau. Vu les multiples possibilités qu'il offre, nous l'avons utilisé pour créer nos interfaces.

I.3 Langages utilisés :

I.3.1 HTML5 (HyperText Markup Language 5) :

C'est la dernière révision majeure d'HTML (format de données conçu pour représenter les pages web). Il permet notamment d'implanter de l'hypertexte dans le contenu des pages et repose sur un langage de balisage. HTML permet aussi de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont les images, les formulaires de saisie et des éléments programmables tels que des applets.

I.3.2 PHP5 :

C'est un langage de programmation compilé à la volée libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. ...

I.3.3 CSS3 :

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelées fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

I.3.4 Python

Python est un langage de script de haut niveau, structuré et open source. Il est multi-paradigme et multi-usage.

Développé à l'origine par Guido Van Rossum en 1993, il est, comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde.

Conçu pour être orienté objet, il n'en dispose pas moins d'outils permettant de se livrer à la programmation fonctionnelle ou impérative; c'est d'ailleurs une des raisons qui lui vaut son appellation de « langage agile ».

Parmi les autres raisons, citons la rapidité de développement (qualité propre aux langages interprétés), la grande quantité de modules fournis dans la distribution de base ainsi que le nombre d'interfaces disponibles avec des bibliothèques écrites en C, C++ ou Fortran. Il est également apprécié pour la clarté de sa syntaxe, ce qui l'oppose au langage Perl.

[20]

C'est le langage qu'on a utilisé pour programmer sous la **Raspberry PI**

I.3.5 Arduino

Fournit un environnement de développement avec un éditeur de source, les opérations de compilation et de chargement dans la mémoire du microcontrôleur étant ramenés à des clicks sur des boutons dans l'IHM (très simple). La communication entre le PC et la carte se fait via le port USB, moyennant installation d'un driver adapté (fourni par ARDUINO).

II configuration du matériel :

II.1 Configuration du Raspberry Pi :

II.1.1 Attribuer une adresse IP statique pour le Raspberry PI :

L'adresse IP est configurée dans le fichier `/etc/network/interfaces` et nous aurons besoin de modifier ce fichier pour la configuration de dynamique à statique. Pour cela, on utilise la commande **ifconfig**


```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:38:85:dd
          inet addr:192.168.1.7  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:24532 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9427 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18343996 (17.4 MiB)  TX bytes:866048 (845.7 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2241 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2241 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:102696 (100.2 KiB)  TX bytes:102696 (100.2 KiB)

pi@raspberrypi ~ $
```

FigureV. 6 configurer de l'adresse ip en mode statique

Dans la partie eth0, à la deuxième ligne celle-ci affiche l'adresse IP, l'adresse de diffusion (Bcast) et le masque de réseau (Mask). Ensuite il nous faut déterminer la passerelle par défaut. C'est à dire l'adresse du modem qui permet d'y accéder. (192.168.1.1).

Et pour cela on doit utiliser la commande **route -n** et puis on doit l'enregistrer dans le fichier `/etc/network/interfaces` en utilisant cette commande

sudo nano /etc/network/interfaces

```
pi@raspberrypi ~ $ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1    0.0.0.0         UG    0     0      0 eth0
192.168.1.0     0.0.0.0        255.255.255.0   U     0     0      0 eth0

pi@raspberrypi ~ $

pi@raspberrypi ~ $ sudo nano /etc/network/interfaces
GNU nano 2.2.6

auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

FigureV.7 détermination de la passerelle par défaut

II.1.2 Installer un serveur Web sur la Raspberry Pi

Pour se faire, on doit installer les modules suivants sur la Raspberry pi : Apache2, PHP5, SQLite3, WiringPi.

✓ Installation du serveur Apache :

Avant d'installer le serveur on met à jour le Raspbian, avec les commandes suivantes

sudo aptitude update:

sudo aptitude upgrade

Une fois Raspbian mis à jour, nous allons installer le serveur Apache, avec la commande suivante : **sudo aptitude install apache2**

✓ **Installation de PHP5**

On installe php5 avec la ligne de commande suivante : **sudo aptitude install php5**

- **Vérifier que PHP fonctionne :**

Pour savoir si PHP fonctionne correctement, la méthode est relativement proche de celle employée pour Apache. En premier lieu, supprimer le fichier « index.html » dans le répertoire « /var/www ». **sudo rm /var/www/index.html**

En deuxième lieu, on crée un fichier « index.php » dans ce répertoire, avec cette ligne de commande **Sudo echo "< ?php phpinfo.php ?>" /var/www/index.php**

Pour vérifier le bon fonctionnement, on saisit l'adresse suivante dans notre navigateur web : <http://192.168.1.250>

✓ **Installer un serveur de gestion de base de données SQLite 3**

On installe SQLite3 avec la ligne de commande suivante :

sudo apt-get install sqlite3

II.1.3 Libérer le port série de la Raspberry PI

Pour libérer le port série non pas pour ouvrir un terminal mais pour envoyer et recevoir des trames, nous allons d'abord configurer le linux embarqué pour qu'il ne se dirige plus vers un terminal mais vers le port série.

Avant de faire cela, il nous faut un autre moyen de se connecter à notre Raspberry pi en utilisant un câble ethernet et ouvrir une session **ssh**

Pour désactiver le bind du terminal sur le port série il nous faut éditer deux fichiers. Le premier fut le plus important et est : /etc/inittab

Ce fichier contient la commande qui active le bind du terminal (**type vt100**) sur le port série à une vitesse de 115200 baud.

Après cela, nous procédons à la fin du fichier et pour commenter la ligne suivante on ajoute un '#' devant.

T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100

La simple modification de ce fichier peut être suffisante. Cependant, le Raspberry pi redirige automatiquement les informations du boot sur le port série. Si vous avez un système branché sur le port série lors de son démarrage.

II.2 Création d'un site web dynamique :

La gestion de file d'attente mobile permet aux clients de rejoindre la file d'attente avant d'arriver et de gagner ainsi un temps précieux. D'où le but du site qui permet au client de consulter la file d'attente à distance et de connaître le dernier numéro du ticket.



Figure V.8 interface de la page d'accueil du site web

Espace client

CONTACT

Nom et prénom

Email

Sujet

Message

ENVOYER

LA POSTE CHIKHI(TIZI-OUZOU)

Adresse : Rue Boulila Amar, Tizi Ouzou

Heur d'ouverture

lundi	• 08:00-16:00
mardi	• 08:00-16:00
mercredi	• 08:00-16:00
jeudi	• 08:00-12:00
vendredi	• Fermé
samedi	• 08:00-16:00
dimanche	• 08:00-16:00

Tél : 026-22-47-44

LA CNAS

7 RUE CHAFAI AHMED

Tizi Ouzou, Tizi Ouzou (15), Algérie

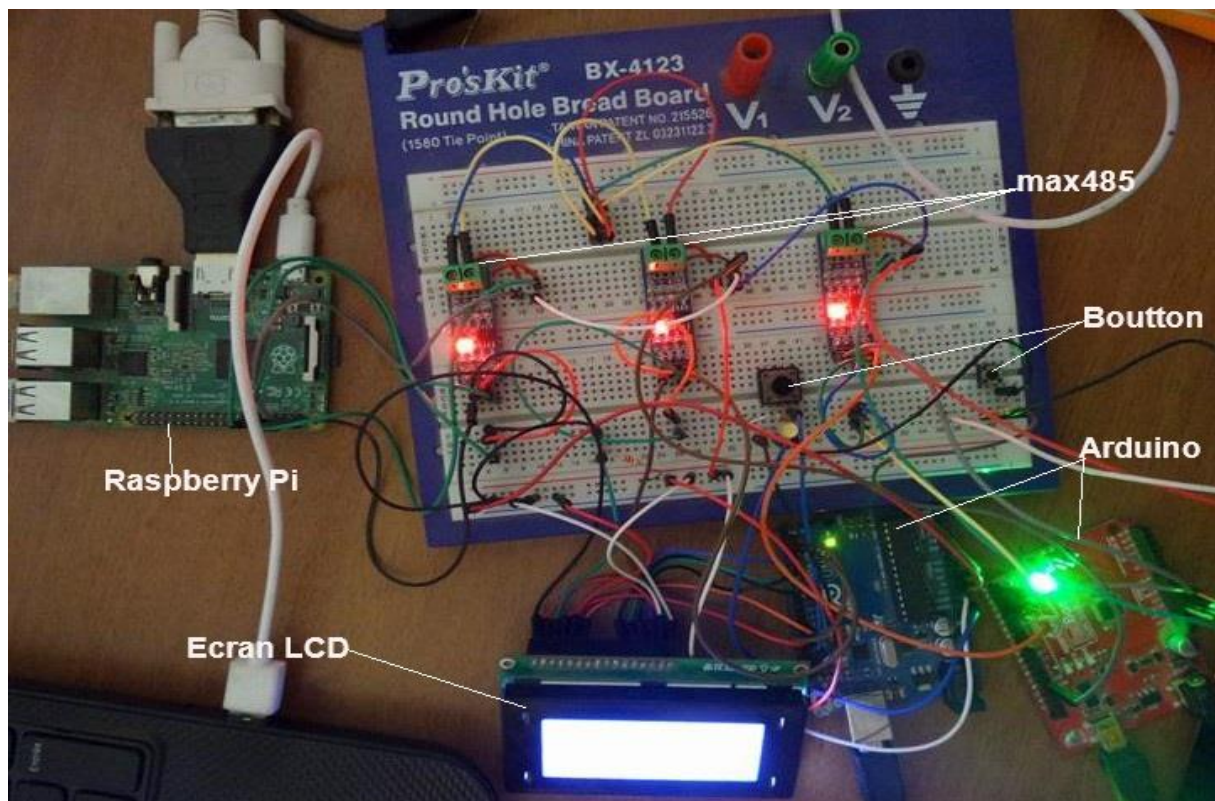
Tél : +213 (0) 26 22 11 44

Fax : +213 (0) 26 22 11 44

FigureV.9 interface de l'espace client dans le site web

II.3 L'Architecture de notre application

Nous vous présentons dès lors l'architecture de notre application :



FigureV.10 interface général de notre application

Cablage de l'arduino et la Raspberry pi

Ce schéma suivant indique le câblage entre l'arduino et la Raspberry pi via la max485

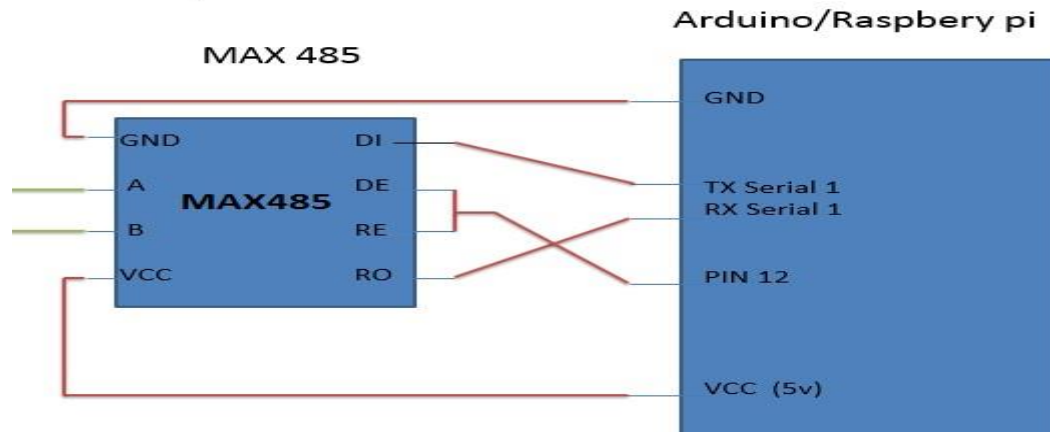


Figure V.11 cablage de l'arduino et la raspberry pi

MAX 485	Arduino
GND	GND
DI	TX Serial 1
DE	PIN 12
RE	
RO	RX Serial 1
VCC	VCC

Affichage du numéro du client en cours sur l'écran LCD et sur l'écran LCD et sur le site web

[Tapez le titre du document]

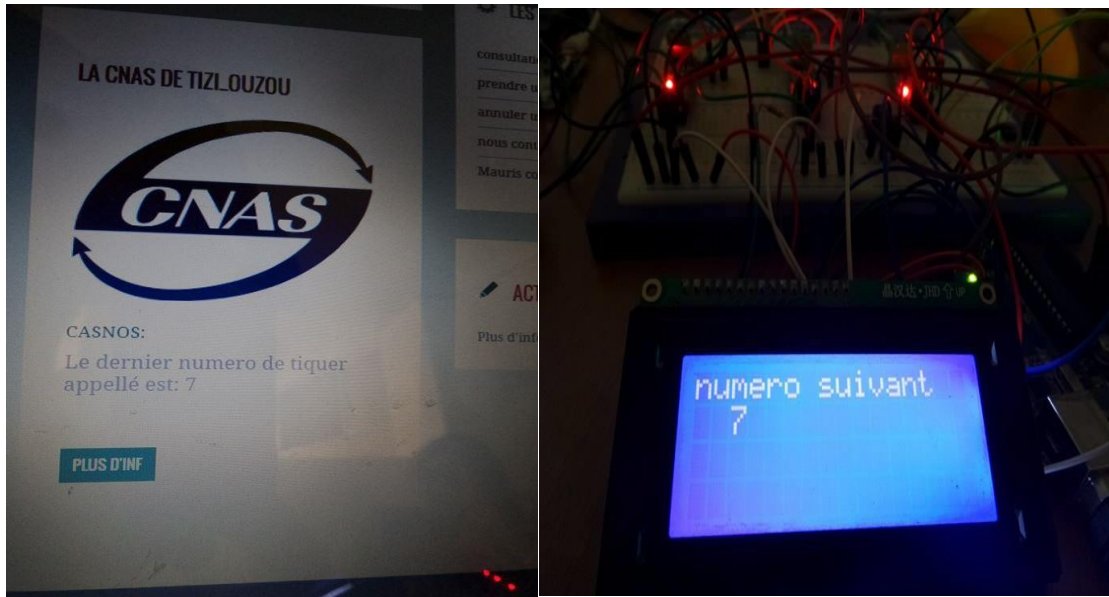


Figure V.12 Affichage du numéro du client sur l'application web et sur l'écran LCD

Après avoir appuyé sur le bouton, on voit tout de suite l'incrémentation du numéro sur l'écran LCD et sur le site web

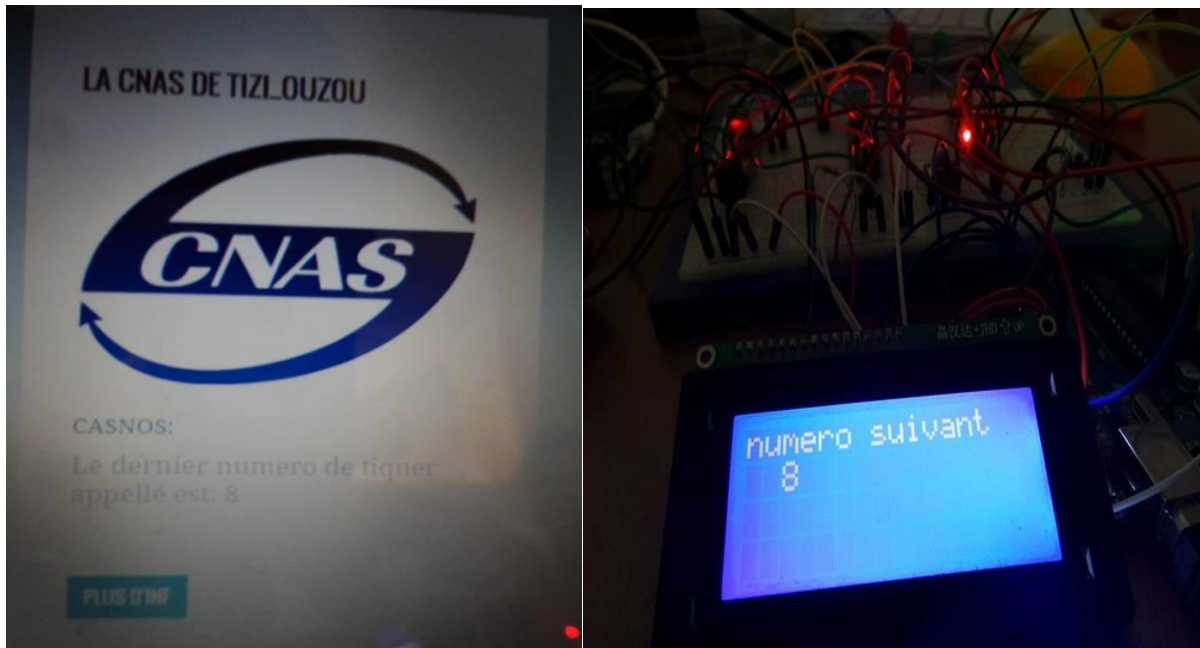


Figure V.13 Incrémentation du numéro du client sur l'application web et sur l'écran LCD

Conclusion

Dans ce dernier chapitre, nous avons explicité les différentes étapes qui nous ont permis de réaliser ce système de gestion des files d'attente, nous avons présenté les différents composants (Raspberry Pi et Arduino) et Language utilisé ainsi nous vous avons montré les différentes étapes qu'on a suivi pour mieux comprendre l'architecture de notre application.

Conclusion générale

Depuis leur apparition, les systèmes embarqués ont connu un franc succès, c'est un domaine vaste et encore fertile pour les chercheurs et les développeurs. Dans le cadre de notre projet , nous avons développé un système embarqués, qui permet de gérer une files d'attente distance en utilisant la Raspberry pi et Arduino , Ce système peut être utilisé dans plusieurs domaines ,(banque , poste , CNAS..etc).

La mise en œuvre de notre projet a exigé des connaissances très approfondies en programmation, ainsi qu'une bonne maîtrise de l'environnement linux .

l'élaboration de ce travail nous a permet d'approfondir les connaissances et le savoir faire acquis durant les années de notre formation au sein de l'université de Mouloud MAMMERI et plus précisément au département informatique, et en arrivant au terme de notre travail, on peut dire que nous avons atteint un double objectif, le premier est la réalisation de notre système très complexe, et le second est enrichissement de nos connaissances en la matière en découvrant les carte Raspberry pi et Arduino , ainsi que la finalisation de nos études par un stage au niveau de laboratoire LABO-DZ, ce stage nous permettras d'avantage d'intégrer le domaine professionnel

ce travail ne constitue qu'un début dans le domaine des système de gestion des files d'attente , des fonctionnalité importante peuvent être rajouter , par exemple une imprimante au lieux d'un distributeur de ticket. elle permet aux client de réservé un ticket directement sur le site web de l'agence , ou bien en ajoutant un module d'appelle par synthèse vocale qui permet d'annoncé vocalement le numéro appelé.

Bibliographie

- [1] A. Bartoli. Le management dans les organisations publiques, Ed. Dunod, 1997.
- [2] K.Salah , thèse de magister « Contribution à la Correction et l'Amélioration de la Qualité de Service dans une Entreprise Publique, en utilisant les Réseaux de Files d'Attente »2013
- [3] .[TIDAF YOUVA , HAFID SALIM Réalisation d'un hotspot WIFI sous une raspberry pi , MEMOIR de l'obtention du diplôme de master 2 en informatique , MOULOUD MAMMERI TIZI OUZOU
- [4] officiel de l'American National Standards Institute *ANSI/TIA/EIA-422-B Electrical Characteristics of Balanced Voltage Differential Interface Circuits*, et sa forme internationale *ITU-T Recommendation T-REC-V.11*, connue aussi sous le nom « X.27 »¹. La révision B, publiée en 1994, a été confirmée en 2005.
- [5] <https://fr.wikipedia.org/wiki/ZigBee>
- [6] <https://fr.wikipedia.org/wiki/Bluetooth>
- [7] <http://www.technologuepro.com/cours-systemes-embarques>
- [8] S4.2 reseau de terrain interface serie modbus rs485
- [9] [www.arduino.cc/cgi-bin/yabb2/YaBB.pl? board=francais.](http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?board=francais.)]
- [10] Livret Arduino en français par Jean-Noël Montagné, Centre de Ressources Art Sensitif, novembre 2006, sous licence CC
- [11] Louis REYNIER – C'est quoi Arduino ?
- [12] <http://www.raspberrypi.org,2015>
- [13] http://sourceforge.net/p/hotcakes/wiki/yfi_setup_nas_coova/
- [14] https://fr.wikipedia.org/wiki/Raspberry_Pi
- [15]<http://lealinux.org/documentations/Pr%C3%A9sentationduRaspberryPi>
- [16]<http://lealinux.org/documentations/Pr%C3%A9sentationduRaspberryPi>
- [17][http://www.ebay.fr/itm/MAX485-Module-RS-485-TTL-to-RS485-MAX485CSA- Converter-Module-1-2-3-5-pcs-/161829909126](http://www.ebay.fr/itm/MAX485-Module-RS-485-TTL-to-RS485-MAX485CSA-Converter-Module-1-2-3-5-pcs-/161829909126)
- [18]<https://www.aurel32.net/elec/lcd.php>
- [19] <http://www.finalclap.com/faq/180-sqlite-definition>
- [20]https://fr.wikibooks.org/wiki/Programmation_Python/Introduction

[Tapez le titre du document]
