



Table des matières

Dédicaces	2
Remerciements	2
Résumé	4
Liste des figures	7
Liste des tables	8
Introduction générale	10
Chapitre 1: Introduction.....	12
1. Organisme d'accueil	13
1.1. Mission et valeurs.....	13
1.2. Les secteurs d'activité	14
2. But du projet	15
3. Diagramme de Gantt relatif au projet.....	15
Chapitre 2 :_Comparatif de plates-formes embarquées	18
1. Une approche des systèmes embarqués.....	18
2. Comparatif des Plates formes embarquées.....	19
3. Choix de la plateforme embarquée.....	20
Chapitre 3: Projet Raspberry Pi.....	28
1. Introduction	28
2. Processus d'installation d'une distribution linux.....	31
2.1. Préparer la carte SD.....	31
2.2. Connecter la carte Raspberry Pi aux périphériques.....	32
2.3. Première connexion sur la Raspberry Pi.....	32
3. Mise en place de la solution	33
3.1. Conception	33
3.2. Architecture de la solution	33
3.2.1. Matériels utilisés	34



3.2.2.	Conception du prototype et fabrication du circuit électronique.....	36
3.2.3.	Programmation.....	44
3.2.4.	Développement web.....	47
3.3.	Mise en place du système	58
	Conclusion et perspectives	59
	Références	60
	Annexe	61



Liste des figures

Figure 1: Platine Raspberry Pi	22
Figure 2: BeagleBone.....	23
Figure 3 : Arduino UNO	24
Figure 4: Schéma général du projet	28
Figure 5: Installation du système d'exploitation sur carte SD	30
Figure 6: Paramètres de configurations Raspberry Pi	31
Figure 7: Interface graphique Raspberry Pi	32
Figure 8: Capteur de température LM 335	33
Figure 9: Modules d'affichage à LED RGB 32x16	33
Figure 10: Connecteur d'alimentation	34
Figure 11: Schémas eagle	37
Figure 12: Typon de la carte	38
Figure 13: Carte électronique	38
Figure 14: Schéma eagle	39
Figure 15: Typon de la carte	40
Figure 16: Carte électronique	40
Figure 17 : Montage du BSS138	42
Figure 18: Installation de la bibliothèque BCM2835	43
Figure 19: Base de données sur PhpMyAdmin	50
Figure 20: Page identification.....	51
Figure 21: Message d'erreur	51
Figure 22: Message de validation de champs	51
Figure 23: Page journal de température.....	52
Figure 24: Graphique des dernières trois heures	53
Figure 25: Graphique des dernières 24 heures.....	53
Figure 26: Graphique de température avec des relevés tous les cinq minutes	54
Figure 27: Page gestion des utilisateurs.....	54
Figure 28: Page gestion des événements	55
Figure 29: Mise en place du système partie affichage sur modules à LED graphiques	55



Liste des tables

Tableau 1: Comparaison des caractéristiques techniques.....	20
Tableau 2: Dictionnaire de données.....	46
Tableau 3 : Entité Admin.....	46



Université Sidi Mohamed Ben Abdellah
Faculté des Sciences et Techniques Fès
Département Génie Electrique



Tableau	4	:	Entité
temp.....			46
Tableau	5	:	Entité
humidity.....			47
Tableau	6	:	Entité
événement.....			47



Introduction générale

Le projet comprend un volet sur l'affichage dynamique et il apparaît opportun de commencer ce rapport par une présentation brève de ce concept.

L'affichage dynamique permis par la signalétique assistée par ordinateur est l'ensemble des procédés informatiques (micro-ordinateur, logiciel, serveur, etc.) permettant la création et la programmation de campagne d'affichage numérique. C'est un outil de communication qui permet de diffuser dans les lieux publics sur un écran, depuis un vidéoprojecteur, sur un mur d'image ou même un mur d'écrans, toutes sortes d'informations.

Ces informations peuvent prendre la forme d'un film, de photos, d'animations Flash, de présentation PowerPoint, de fichier (PDF, Excel, Word), de page web, de flux RSS ou streaming.

Toutefois l'appellation affichage dynamique est souvent usurpée par des systèmes qui n'ont rien de dynamique et ne sont que des "cadres photos" (même de grande taille) qui sont parfois pilotables à distances.

Quelques solutions innovantes, permettent d'être pilotées par un simple navigateur web pour un contrôle de la solution quel que soit son lieu et son environnement. La plupart du temps la solution nécessite un client installé sur un ordinateur rendant dépendant la programmation.

Dans notre cas l'affichage sur les modules à LED est géré par une plateforme embarquée qui se chargera d'envoyer des messages text de manière quasi instantanée suite à des événements extérieur tel que: la température et l'humidité.

Les messages Text à afficher sur les matrices sont configurés par l'administrateur via une page Web de gestion d'événement.

Outre l'introduction générale, ce rapport est composé de trois chapitres.



Dans le premier chapitre nous décrivons le contexte général du projet, nous commencerons dans une première partie par décrire brièvement la société INTELIFEX SYSTEMS le promoteur du projet. Le reste du chapitre sera réservé à une description du projet.

Dans le second chapitre nous donnerons une vue comparative des plates-formes embarquées.

Dans le troisième chapitre nous commencerons par présenter l'architecture matérielle et logicielle du système, ensuite nous allons présenter l'environnement de travail en termes d'outils utilisés et de choix technologiques, nous finirons par traiter la mise en place de la solution, et enfin la conclusion.

Chapitre 1 : Introduction

Chapitre 1: Introduction	12
1. Organisme d'accueil.....	13
1.1. Mission et valeurs.....	13



1.2. Les secteurs d'activité	14
2. But du projet.....	15
3. Diagramme de Gantt relatif au projet.....	15

Chapitre 1: Introduction

Ce chapitre a pour objectif de présenter dans un premier temps l'organisme d'accueil dans lequel j'ai effectué mon stage de fin d'études et dans un deuxième temps, je vais mettre le point sur le but du projet.



1. Organisme d'accueil

INTELIFEX SYSTEMS est une entreprise de prestation de services. Spécialisée dans le développement de toute solution électronique, électrique et logicielle ayant trait à l'automatisme, la supervision et l'informatique industrielle et technique.

Depuis sa création, INTELIFEX SYSTEMS s'est imposée comme une référence dans le domaine en travaillant sur des projets toujours plus ambitieux pour ses nombreux clients dans différents secteurs d'activités. Leur offre s'appuie sur des compétences expertes et une méthodologie éprouvée.

En renfort des équipes ou en menant le projet de sa conception jusqu'à son intégration, leur expertise permet la réussite de tous les projets. Plus particulièrement tournés vers les domaines de l'électricité, de l'automatisme et de la robotique, ils répondent à tout besoin industriel nécessitant des solutions matérielles et logicielles embarquées.

INTELIFEX emploie plusieurs collaborateurs incluant des experts qualifiés et qui sont dotés de tous les moyens possibles matériels et humains pour offrir des solutions clé en main à leurs clients. Ils ont développé des relations d'affaires avec un certain nombre de leaders technologique tel PELCO, BOSCH, SONY, KOUKAM, PROXIM, PAXTON etc....

1.1.Mission et valeurs

INTELIFEX SYSTEMS vise à:

Être le partenaire de confiance de ses clients, fournir les meilleurs avis et conseils possibles pour assurer le maximum de profits et gains de productivité à travers l'implémentation des dernières technologies en matière de télécommunications.

Gérez pour leurs clients la complexité de leurs besoins de communications tout en s'occupant d'intégrer les solutions qui répondent à leur niveau d'exigence. Faciliter l'augmentation d'activité en offrant à leurs clients des solutions flexibles et des services à la demande à travers des applications intelligentes basée sous IP. Délivrer le meilleur service industriel à leurs clients à tout moment.

Les Valeurs qui articulent la Mission de INTELIFEX SYSTEMS sont les suivantes :



Avoir un engagement total envers ses Clients, d'investir dans leurs propres ressources humaines, pour conserver leur statut d'entreprise experte et Leader en Afrique de l'Ouest dans les services de Télécommunications.

De rester des passionnés de Nouvelles Technologies, ils sont obsédés par la veille technologique et les bénéfices que peuvent en tirer leurs clients.

1.2. Les secteurs d'activité

AUTOMATISME ET TELEGESTIO:

- Contrôle commande de processus industriels
- Télé-contrôle, Télécommande et Télégestion via Stellite, GSM/GPRS, Radio
- Régulation
- Interface Homme / Machine
- Pilotage E/S décentralisées

Formation : Rénovation, amélioration, fiabilisation de processus, Conception de banc de test, Contrat de maintenance

INFORMATIQUE INDUSTRIELLE:

- Supervision graphique et contrôle commande des processus industriels
- Gestion de production (MES)
- Outils d'aide à la décision
- Gestion entrepôts, magasins
- Gestion commerciale
- Gestion d'énergies, GTC

ELECTRONIQUE :

- Réparation de cartes, systèmes électroniques,
- Réparation radio, relais, mobiles, portatifs PMR,
- Modification, duplication de cartes obsolètes
- Conception et réalisation de cartes et systèmes
- Contrat de maintenance atelier et sur site

ARMOIRES ELECTRIQUE :



- Etude sur cahier des charges,
- Réalisation des schémas et des plans,
- Montage/câblage de tableaux basse tension, pupitres de commande, baies courants faibles.

La société INTELIFEX SYSTEMS dans laquelle j'ai effectué mon stage ne se limite pas seulement à ces secteurs et essaye à chaque fois d'en intégrer d'autres en fonction des besoins de ses clients et de ses moyens humains et matériels.

2. But du projet

Ce projet de fin d'études traite le domaine de l'électronique embarqué il s'agit ainsi de la mise en place de la solution de gestion de la température dans un système embarqué.

3. Diagramme de Gantt relatif au projet

Avant de débiter dans ce travail nous présentons le diagramme de Gantt afin de visualiser ce qui a été fait durant ce stage aussi pour rappeler les étapes suivies lors du travail dans sa globalité.



Université Sidi Mohamed Ben Abdellah
Faculté des Sciences et Techniques Fès
Département Génie Electrique



Système de gestion de la température dans les salles informatiques	Start Date	End Date	Time
Carte Arduino UNO	3-Mar	16-Mar	13
Documentation sur carte Arduino et initiation au langage Arduino			
Protocoles de communications			
Système d'affichage dynamique sur carte FPGA	17-Mar	23-Mar	6
Documentation sur carte de développement ALTERA DE2			
Recherche internet sur les systèmes d'affichages dynamiques			
Etude et reproduction d'une carte de contrôle à base FPGA solution optimisée	24-Mar	13-Apr	20
Décortication de la carte de contrôle FPGA			
Etude et décortication de la carte de contrôle des matrices à LED existante à base de pic			
Etude des matrices d'affichage à LED et du code C existant			
Utilisation de l'arduino pour le contrôle des matrices d'affichage			
Programmation sur pic	14-Apr	20-Apr	6
Développement de programme C d'affichage d'heure de date et de température			
Discussion à proopos du projet Raspberry et validation du sujet	21-Apr	24-Apr	3
Travail sur projet Raspberry Pi	25-Apr	13-Jun	49
Etude comparative des plateformes embarquées			
Choix de la plateforme			
Etude du cahier des charges			
Recherche internet sur les conditions de climatisation dans les salles informatiques			
Choix des outils utilisés et choix technologiques			
Prototypage des cartes électroniques			
Fabrication de la carte			
Travail sur la Raspberry Pi et programmes de test plateforme Adafruit			
Spécification des besoins et attentes pour la réalisation d'une interface web pour la gestion de la température			
Installation du Wampserver et travail en local			
Création de la base de données sur phpMyAdmin			
Initiation sur l'outil Dreamweaver			
Développement de pages Web et test en local			
Installation du système d'exploitation Raspbian et des outils Apache MySQL phpMyadmin sur le Raspberry et du FTP pour le partage de fichier			
Exportation de la base de données sur RPi			
Conception de programme C sur RPi permettant le bouclage sur la table événement de la BD			
Programmation du PIC			
Liaison du système et mise en œuvre de la solution			
Test de fonctionnement			
Conception de boîtier			



Chapitre 2: Comparatif de plates-formes embarquées

Chapitre 2 : Comparatif de plates-formes embarquées	18
1. Une approche des systèmes embarqués.....	18
2. Comparatif des Plates formes embarquées.....	19
3. Choix de la plateforme embarquée.....	20



Chapitre 2

Comparatif de plates-formes embarquées

Sans nous en rendre compte, nous utilisons actuellement tous de nombreux systèmes embarqués. En effet, de nos jours, ces systèmes électroniques sont présents dans de nombreux domaines d'applications tels que les systèmes multimédia, la domotique, la voiture, l'avionique, etc...

Ce chapitre sera d'abord consacré à la présentation des notions des systèmes embarqués ensuite il fera l'objet des concepts de base liés aux plates-formes embarquées.

1. Une approche des systèmes embarqués

Un système embarqué est un système complexe qui intègre du logiciel et du matériel conçu ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires. Le système matériel et l'application (logiciel) sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau (PC). Un système embarqué est autonome et ne possède pas des entrées/sorties standards [Réf].

Le logiciel a une fonctionnalité fixe à exécuter qui est spécifique à une application. L'utilisateur n'a pas la possibilité de modifier les programmes. Bien souvent, il n'a pas conscience d'utiliser un système à base des microprocesseurs. Les systèmes embarqués sont désormais utilisés dans des applications diverses tels que le transport (avionique, espace, automobile, ferroviaire), dans les appareils électriques et électroniques (appareils photo, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables), dans la distribution d'énergie, dans l'automatisation, etc. L'attrait principal des systèmes embarqués vient du fait qu'ils permettent d'implémenter à faible coût des fonctions complexes dont la réalisation était inimaginable il y a quelques années seulement.



Aujourd'hui, la tendance générale est d'utiliser les systèmes embarqués pour incorporer des fonctionnalités complexes, précédemment considérées comme exotiques, dans les produits de tous les jours.

¹Les systèmes embarqués fonctionnent généralement en Temps Réel (TR) : les opérations de calcul sont alors faites en réponse à un événement extérieur (interruption matérielle). La validité et la pertinence d'un résultat dépendent du moment où il est délivré. Une échéance manquée induit une erreur de fonctionnement qui peut entraîner soit une panne du système (plantage), soit une dégradation non dramatique de ses performances.

Lorsque les systèmes embarqués sont utilisés dans les produits de grande consommation, ils sont fabriqués en grande série. Les exigences de coût se traduisent alors en contraintes drastiques sur les différentes composantes du système : utilisation de faibles capacités mémoires et de petits processeurs (4 bits ou 8 bits), mais en grand nombre. Ainsi, les systèmes embarqués sont particulièrement sensibles au coût de production. Il existe des applications dans lesquelles les contraintes de coût de production et de maintenance ont une importance de même niveau que les performances envisagées [Réf].

Dans les systèmes embarqués autonomes, la consommation d'énergie est un point critique pour le coût.

En effet, une consommation excessive augmente le prix de revient du système embarqué, car il faut alors des batteries de forte capacité.

2. Comparatif des Plates formes embarquées

Les fournisseurs de technologies sont en train de développer des plates-formes embarquées qui combinent des composants matériels avec un Framework logiciel intégré pour aider les équipes de conception à construire plus rapidement des systèmes embarqués complexes.

Il existe une multitude de plates-formes embarquées, chacune offrant des environnements de travail orientés vers des domaines en particulier, sans engager dans un comparatif entre toutes les différentes plates-formes existantes, on peut quand même citer les plus célèbres d'entre elles.

Raspberry Pi, BeagleBone, BeagleBoard et Arduino

¹ [Réf] : <http://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-introduction.htm>



Développées à l'origine dans ou pour le monde universitaire, très prisées des amateurs éclairés en électronique, les cartes Raspberry Pi, BeagleBone, BeagleBoard et Arduino ont acquis petit à petit leurs lettres de noblesse dans le monde des applications professionnelles industrielles. Au point qu'elles deviennent des plates-formes de choix pour réaliser du prototypage, des maquettes ou du développement. Une évolution sur laquelle surfent les grands distributeurs, qui utilisent ces cartes électroniques comme têtes de gondole.

Il existe à l'heure actuelle sur le marché, un nombre incalculable de plates-formes, cartes, modules, designs de référence basés sur des microcontrôleurs dont le rôle est toujours le même : pouvoir développer et déboguer rapidement une application, une ébauche d'application ou une maquette avant de passer aux choses sérieuses, l'industrialisation.

Problème, ces plates-formes matérielles, parfois onéreuses, ont surtout l'inconvénient d'être fermées, propriétaires, et parfois mal documentées. Or, dans ce domaine, depuis quelque temps, l'émergence de cartes à bas coûts issues de projets communautaires est en train de bouleverser le paysage. En fait, au même titre que les technologies open source ont profondément modifié l'économie du logiciel dans le secteur des systèmes embarqués, l'émergence d'un mouvement similaire pour le matériel pourrait aboutir à une évolution identique. Trois cartes aujourd'hui très populaires, issues d'origines diverses, symbolisent pour partie cette tendance : Arduino, Raspberry Pi et BeagleBone. Utilisées à leurs débuts par des amateurs éclairés ou pour des projets de petite envergure, elles ont pris aujourd'hui une place très importante sur le marché et sont devenues de véritables points d'entrée pour les grands distributeurs (Digi-Key, Farnell, Mouser, RS Components...) qui n'hésitent pas à les mettre en avant sur leurs catalogues.

3. Choix de la plateforme embarquée

Le tableau numéro 1 présente les principales caractéristiques de ces trois cartes.



Nom de la carte	Arduino Uno	BeagleBone	Raspberry Pi (Model B)
Origine	Interaction Design Institute d'Ivrea (Italie)	Projet de Hardware Open Source piloté par Texas Instruments	Université de Cambridge
Organisation en charge des spécifications	Arduino.cc	BeagleBoard.org	Raspberry Pi Foundation (fondation de droit anglais)
Naissance	2005 (fabrication en Italie par Smart Projects)	2008 (BeagleBoards) - 2011 (BeagleBone) (accord de fabrication/distribution avec Digi-Key)	2008 (accord de fabrication avec RS Components et Farnell/Element 14 en 2011)
Taille	45,43 x 32,34 mm	86,36 x 53,34 mm (bords arrondis)	85,60 x 53,98 mm
Processeur	ATmega328 8 bits d'Atmel à 16 MHz	Sitara 335x de TI basé sur un Cortex-A8 à 720 MHz (1 GHz pour la BeagleBone Black)	BCM2835 de Broadcom basé sur un ARM11 à 700 MHz GPU intégrée (Video Core 4 de Broadcom)
Mémoires	2 Ko Ram, 1 Ko Eeprom	256 Mo DDR2 (512 Mo DDR3 pour la BeagleBone Black)	512 Mo SDRAM
Mémoire Flash	32 Ko	Sur MicroSD (4 Go)	Sur carte SD
Tension d'entrée	7 V - 12 V	5 V - 3,3 V	5 V
Consommation	42 mA (0,5 W)	210 à 450 mA (2,5 W max.)	700 mA (3,5 W)
Ethernet	Non	10/100 Ethernet	10/100 Ethernet
USB	Non	1 USB 2.0	2 USB 2.0
Sorties vidéo	Non	Non (micro HDMI pour la BeagleBone Black)	Composite et HDMI
Développement	Langage de programmation Arduino.	Environnement BoneScript. Langages Phyton, Scratch, Squeak	Langages Scratch, Squeak

Tableau 1: Comparaison des caractéristiques techniques

On a le choix entre le Raspberry, le BeagleBone et l'Arduino, entre autres.

Libre à chacun de choisir celle qui lui convient le mieux, qui est la plus adaptée à ses besoins et cahiers de charges.

D'abord, il faut préciser ce que l'on attend de la plateforme qui fera fonctionner le système.

Méthodologie

Les arguments de choix généralement utilisés sont: la taille, le prix, le poids, la puissance calcul nécessaire, le temps de mise en œuvre raisonnable, la compatibilité Linux, la compatibilité Arduino/Processing, l'utilisation de ressources standards par ailleurs

Taille : Le plus petit est le mieux. Disons qu'un format embarqué acceptable est grosso-modo la taille d'une demi-feuille A4, soit environ 15cmx21cm sur une épaisseur de 3cm...



Prix : Le prix doit être le plus bas possible et disons "raisonnable" pour une application dédiée embarquée, et en tout cas, dans le même ordre de grandeur qu'une base embarquée classique type ARM, de l'ordre de 150 €

Poids : Là encore, le mieux est le poids le plus faible. Un poids de l'ordre du kilogramme semble être un maximum à ne pas trop dépasser. Un autre objectif de référence est l'utilisation à bord d'une plateforme embarquée sur ballon sonde par exemple. Là aussi, on peut difficilement dépasser le kilogramme. L'idée est globalement une mobilisation facile (sur soi, en bateau, en voiture, en camping-car, jardin, etc...).

Basse consommation : La plateforme doit également pouvoir être autonome un temps raisonnable, de quelques heures à plusieurs jours, sur sa source d'alimentation ou alimentée par panneau solaire, ce qui suppose une consommation basse. L'objectif est une base embarquée x86 consommant de l'ordre de 20 à 30 W maximum.

Puissance calcul nécessaire : Avec comme projet de référence un PC embarqué sur un robot mobile capable de réaliser de la reconnaissance visuelle et du traitement d'image ou encore de l'affichage permanent de type "oscilloscope", il est nécessaire de disposer d'une puissance de calcul conséquente. Là encore, le plus sera le mieux, dans la limite d'un coût raisonnable.

Temps de mise en œuvre raisonnable : Même si ce point ne semble pas essentiel à première vue, en pratique il est très important. En effet, il suffit d'avoir tenté d'installer une distribution Gnu/linux dite "embarquée" sur une plateforme de type ARM pour comprendre ce que "temps de mise en œuvre raisonnable" veut dire. Le but premier est de se concentrer sur les développements liés aux fonctions utiles de la plateforme embarquée, pas d'installer le système de base. A ce titre, la distribution Ubuntu et ses variantes donne en pratique d'excellents résultats puisque l'installation du système de base pourra se faire raisonnablement en moins d'une demi-heure sur une plateforme compatible.

Compatible Linux : Dans la lignée du temps raisonnable, le choix se porte tout naturellement sur une distribution Linux et ses variantes qui s'installent simplement à partir d'une clé USB live. De plus, cette distribution propose une quantité de paquets impressionnante qui permet de faire face en quelques minutes à tous les besoins utiles. Le support de la communauté est par ailleurs très efficace.



Compatible Arduino/Processing : Le système embarqué doit pouvoir supporter l'exécution en embarqué des logiciels Arduino et Processing, ce qui suppose l'exécution embarquée de Java également. Avec les 2 logiciels Arduino et Processing en embarqué, on disposera de fonctionnalités très intéressantes :

- Communication par réseau wifi entre interfaces processing embarquée et fixe
- Programmation de la plateforme embarquée à distance et sans fil via réseau wifi grâce à l'accès au bureau distant du PC embarqué depuis un PC fixe.
- Utilisation des fonctions graphiques propres de Processing (traitement d'image, gestion de fichier et datalogging de donnée) et des bibliothèques avancées de Processing, notamment communication Série avec la carte Arduino, openCV pour la reconnaissance visuelle, gestion réseau, gestion vidéo, la gestion programmée en ligne de commande de programmes système pour la synthèse vocale ou la lecture de fichier sons, etc...

Supportant logiciels spécifiques utiles : Le système embarqué devra pouvoir exécuter en embarqué des logiciels spécifiques utiles tels que la synthèse vocale ou la lecture de fichier sons, donnant accès une utilisation simplifiée sur un robot mobile. L'utilisation de logiciels avancés tel que Blender (3D) devra également être possible.

Utilisation de ressources standards par ailleurs (wifi, ethernet, format de fichiers sons, etc...) : l'utilisation des ressources standards devra être supportée, notamment connexion réseau Ethernet, connexion Wifi, lecture des fichiers sons courant, etc... Là aussi, le gain de temps sera appréciable. Par exemple, l'utilisation d'un fichier sonore consistera simplement à télécharger sur internet le fichier voulu et à le lire sur la plateforme embarquée tel quel, sans avoir à l'alléger ou autre...

Le Raspberry Pi correspond à ces critères, c'est le plus répandu et le moins cher, donc on va le prendre comme point de référence, et le comparer aux produits concurrents.

Raspberry Pi



Figure 1: Platine Raspberry Pi

- Elle dispose d'emblée d'une sortie MLI sous la forme d'un connecteur GPIO (General Purpose I/O). On peut avoir besoin de composants additionnels entre le connecteur et l'appareil périphérique.
- L'alimentation par le port micro USB le connecte sur le réseau avec un chargeur (on peut utiliser un chargeur de mobile).
- La sortie RCA le relie à un écran et le connecteur audio supporte une prise de 3.5 V.
- Le groupe de LEDs montre l'état du fonctionnement.
- Deux USB 2 pour un clavier et une souris en mode développement. On peut y insérer différents dongles, dont le Wi-Fi.
- Une prise RJ-45 pour se relier au réseau.
- Une sortie HDMI pour connecter un écran.
- Le connecteur CSI relie la carte à un appareil photo ou tout autre appareil fournissant des images ou des senseurs.
- Le connecteur DSI (Display Serial Interface) relie la carte à un affichage LCD.
- SD pour stocker le système d'exploitation et les données.
- Le JTAG sert au débogage de la carte et n'est pas très utile (ni très documenté).
- Il y a au moins deux connecteurs très utiles pour concevoir un robot, le CSI et le GPIO. Mais il n'est pas exclu d'utiliser l'USB pour connecter des éléments. Tous les autres connecteurs permettent de programmer la carte et contrôler le fonctionnement du programme grâce à l'interface de développement.



BeagleBone Black



Figure 2: BeagleBone

- Dans le même ordre de prix, avec un processeur similaire et une mémoire de 512 MO également.
- Alimenté par un port mini USB ou un jack. Cela permet de réutiliser un chargeur, si l'on en a déjà un de 5 V et 1000 ampères au moins.
- Un USB 2. Connecteur hôte comme ceux du Rpi.
- Un USB client. Peut servir à l'alimentation.
- Un port série Serial Debug UART0. Il permet de communiquer avec la carte. Requiert une interface TTL.
- Connecteur RJ-45.
- Carte micro SD.
- Micro HDMI. Pas de sortie audio dédié, le son peut passer par le HDMI, mais le robot ne se déplace pas avec un écran, donc c'est une limitation.
- Deux connecteurs d'extension CAN. Ils donnent une certaine liberté pour connecter des périphériques.
- La charte affiche 3 accès MLI, passant sans doute par les connecteurs CAN.
- Il n'a pas les mêmes capacités de traitement vidéo que le Raspberry Pi, donc pas le même confort pour l'interface de développement, mais plus de connecteurs GPIO grâce aux ports d'extensions. On trouvera moins facilement des réponses aux problèmes qu'avec la Raspberry Pi, même s'il a une bonne documentation.

Arduino



Figure 3 : Arduino UNO

La carte Arduino est nettement moins puissante que le Rpi. Mais elle a des avantages, elle consomme nettement moins de courant, elle est compacte. On peut la relier à une carte Bluetooth ou Wi-Fi pour recevoir des ordres d'un ordinateur plus puissant qui se chargera de tous les traitements et enverra des commandes élémentaires.

Conclusion

Il y a de plus en plus de nouveaux modèles équivalents. La Raspberry Pi a toujours l'avantage de la popularité, donc plus de logiciels, plus de support, le choix s'est porté sur la plateforme Raspberry Pi.



Chapitre 3: Projet Raspberry Pi

Chapitre 3: Projet Raspberry Pi	28
1. Introduction	28
2. Processus d'installation d'une distribution linux	31
2.1. Préparer la carte SD.....	31
2.2. Connecter la carte Raspberry Pi aux périphériques.....	32
2.3. Première connexion sur la Raspberry Pi	32
3. Mise en place de la solution	33
3.1. Conception.....	33
3.2. Architecture de la solution.....	33
3.2.1. Matériels utilisés.....	34
3.2.2. Conception du prototype et fabrication du circuit électronique	36
3.2.3. Programmation	44
3.2.4. Développement web	47
3.3. Mise en place du système	58
II de l'affichage de la température sur les matrice à LED	59
Conclusion et perspectives	59



Références	60
Annexe.....	61

Chapitre 3: Projet Raspberry Pi

1. Introduction

Le projet sur lequel nous travaillons s'adresse aux salles informatiques.

Une salle informatique est un site physique sur lequel se trouvent regroupés des équipements constituant le système d'information de l'entreprise (ordinateurs centraux, serveurs, baies de stockage, équipements réseaux et de télécommunications, etc.).

C'est un service généralement utilisé pour remplir une mission critique relative à l'informatique et à la télématique.

Il comprend en général un contrôle sur l'environnement (climatisation, système de prévention contre l'incendie, etc.), une alimentation d'urgence et redondante, ainsi qu'une sécurité physique élevée.

Des enjeux environnementaux sont liés à la consommation d'électricité des salles informatiques, et à leur coproduit qu'est la chaleur, souvent dissipée par les serveurs.

Les fabricants d'ordinateurs livrent des appareils qui dégagent d'importantes quantités de chaleur dans un espace de plus en plus réduit en volume. Le fonctionnement de ces



machines n'est garanti que si elles travaillent dans des conditions d'ambiances bien particulières où la température, l'hygrométrie et la pureté de l'air sont maîtrisées.

De ce fait, la climatisation a une fonction industrielle liée intimement au process de l'entreprise sans quoi il ne peut fonctionner correctement.

Le fonctionnement de la climatisation doit répondre à des exigences:

- De précisions
- De fiabilité
- De maintenance rapide

En général, on trouve des salles informatiques avec des niveaux suivants :

Température ambiante : de 20 à 26°C

Humidité ambiante : de 50 à 60 % HR

Le maintien de la climatisation à environ 20 degrés Celsius est essentiel, puisque les appareils électroniques génèrent beaucoup de chaleur et deviennent défectueux lorsque la température s'élève au-delà d'une certaine limite.

Il suffit d'un évènement d'arrêt ou d'une série d'incidents évités de justesse pour se décider d'investir dans l'amélioration de la disponibilité des opérations informatiques. Dans la plupart des cas, cela incite à réaliser de nouveaux projets de modernisation des équipements informatiques.

Ce projet de modernisation sur lequel nous travaillons est l'occasion d'assurer le bon déroulement des opérations informatiques en mettant en place un système embarqué qui permettra la gestion de la température en temps réel avec des interfaces web. Ces interfaces permettront une aisance d'accès à l'information et rendront possible le contrôle à distance même si la personne n'est pas présente physiquement sur le site.

Ce système sera caractérisé par une régulation précise maintenant par exemple le point de consigne en température à +/- 0,3°C.

Pour cela nous proposons le schéma général suivant



Schéma général:

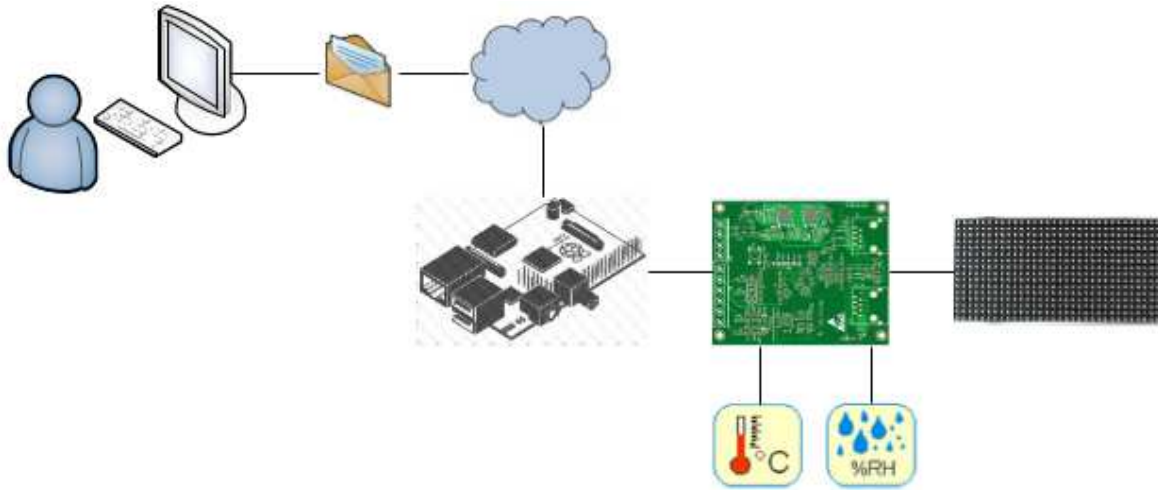


Figure 4: Schéma général du projet

Un capteur de température sera mis en œuvre, ce dispositif permettra de détecter en temps réel la température de la salle et de transformer la sensation du refroidissement ou du réchauffement en signal électrique.

Le traitement de ce signal électrique sera confié à un microcontrôleur pic de la société Microchip qui permettra de communiquer les informations reçues à la plateforme embarquée, nous rappelons que dans notre cas le choix s'est porté sur la Raspberry Pi qui a toujours l'avantage de la popularité, donc plus de logiciels et de support.

Sur la Raspberry sera installé un serveur web qui hébergera les pages web pour permettre la gestion de la température en temps réel.

Au dépassement des limites de tolérance en températures la Raspberry Pi transmettra des mails électroniques de prévention. La Raspberry pourra aussi envoyer des commandes au pic pour l'activation d'un appareil relié à un relais. Ou encore donner au pic l'ordre d'écriture de message text sur les modules d'affichage.



La partie suivante sera consacrée à la conception de notre plateforme Raspberry Pi, nous allons voir comment préparer et installer une distribution GNU/Linux.

2. Processus d'installation d'une distribution linux

Le succès du petit système Raspberry Pi n'est plus à démontrer. Alliant faible prix de revient et un potentiel informatique prometteur, il s'impose comme une base expérimentale incontournable pour Linux embarqué.

Nous présentons par la suite le processus d'installation de Linux.

Tout d'abord avant d'utiliser la Raspberry Pi, il faudra:

- Une carte SD;
- Un câble HDMI;
- Câble Ethernet;
- Un clavier et une souris USB standard;
- Un hub USB avec alimentation séparée.

2.1. Préparer la carte SD

La carte SD est ce qui fera office de mémoire morte sur la Raspberry Pi, tout comme un disque dur d'ordinateur. C'est donc là-dessus que sera stocké le système d'exploitation, et à priori, les documents, photos, musiques, vidéos...

Nous avons commencé par télécharger le système d'exploitation, qui est en fait une distribution de Linux qui s'appelle Raspbian. Cette distribution est une version modifiée (adaptée pour la Raspberry Pi) de Debian, qui est utilisée sur la grande majorité des serveurs de sites Internet.

D'autres distributions de Linux peuvent être installées, par exemple Debian (l'originale). Mais Raspbian optimisée pour la Raspberry Pi, et peut faire gagner de 7% à 41% de performances en plus par rapport à Debian.

Installation de la carte SD du Raspberry Pi



Le but ici est d'expliquer comment installer le système d'exploitation sur la carte SD pour le Raspberry Pi.

Il faut récupérer l'image, c'est à dire un gros fichier contenant le système à installer, généralement un .iso ou un .img ou un fichier compressé de la distribution à installer.

Pour Raspbian, ce sera sur <http://www.raspberrypi.org/downloads> dans la section Raspbian wheezy. C'est un .zip qu'il faut dézipper

Nous avons utilisé le logiciel Win32 Disk Imager sous Microsoft Windows.

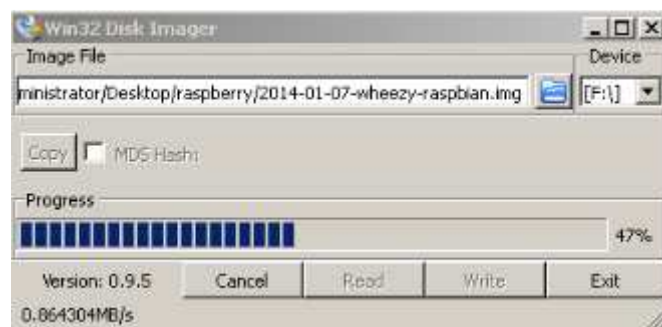


Figure 5: Installation du système d'exploitation sur carte SD

2.2. Connecter la carte Raspberry Pi aux périphériques

On insère la carte SD sur le connecteur Raspberry Pi. On branche le câble HDMI sur l'écran, le clavier et la souris, et pour finir, l'alimentation (on la branche sur le secteur)

Le système démarre automatiquement.

2.3. Première connexion sur la Raspberry Pi

La dernière version du système d'exploitation, Raspbian, simplifie les choses

Lorsqu'on lance pour la première fois la Raspberry Pi on tombe sur cet écran

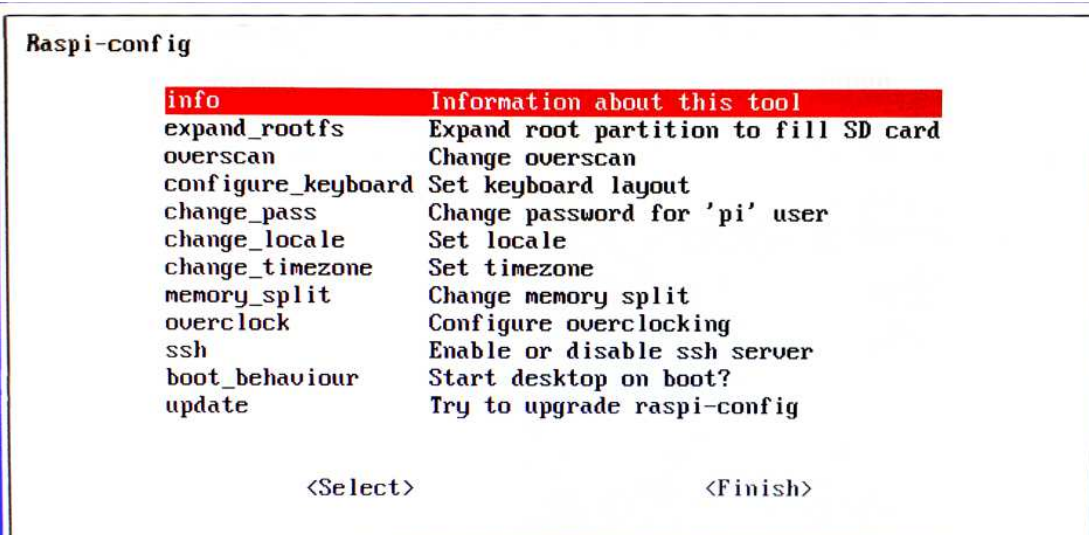




Figure 6: Paramètres de configurations Raspberry Pi

Les paramètres que l'on devait auparavant régler en ligne de commandes sont désormais plus faciles d'accès. Quelques options peuvent être intéressantes à changer:

expand_rootfs: vous permettra d'utiliser tout l'espace de la carte SD

configure_keyboard: on tape setxkbmap fr pour mettre le clavier en français

change_pass: modifier le mot de passe de l'utilisateur 'pi' (par défaut: Raspberry)

change_locale: changer la langue du système

change_timezone: changer l'heure locale

ssh: activer le SSH

boot_behavior: permettra d'atterrir directement sur le bureau, pas sur une console.

On sélectionne les options qui nous intéressent, puis on sélectionne «Finish». Au final, on arrive sur le bureau de LXDE, le gestionnaire d'interface graphique installé:

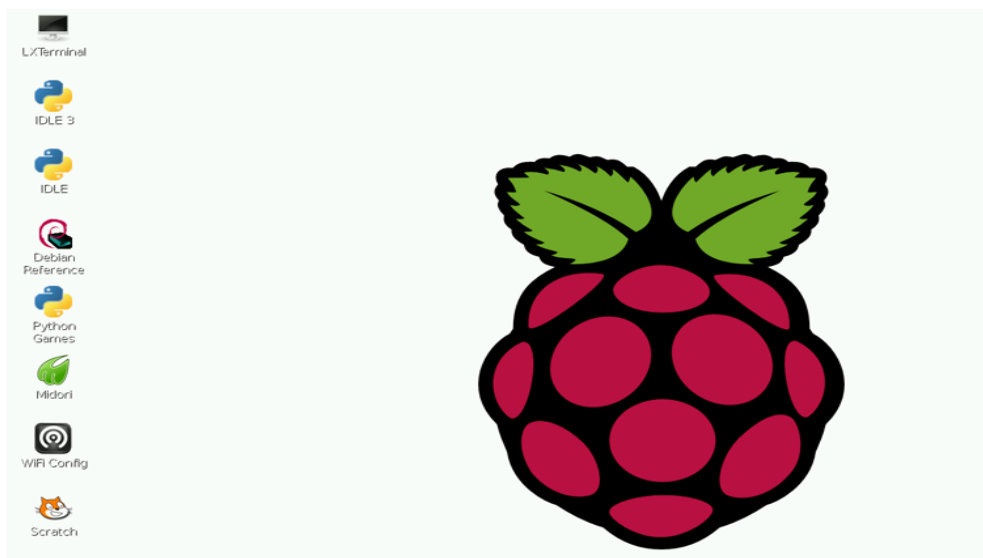


Figure 7: interface graphique Raspberry Pi

3. Mise en place de la solution

3.1. Conception

Dans cette section on va présenter l'architecture générale de notre système ainsi que le matériel utilisé.

3.2. Architecture de la solution

Notre travail peut être décomposé en quatre parties, une partie qui décrit le matériel utilisé dans l'application, une partie portant sur la conception du prototype la fabrication et la



validation du circuit électronique, une partie qui traite de la programmation du pic en C et une dernière partie sur le développement des pages web.

3.2.1. Matériels utilisés

Capteur de température

Le choix du capteur a été la première chose à effectuer dans la conception de notre système de gestion de la température, en effet il constitue la base du fonctionnement du système en récupérant les données physiques de la température.

Nous avons eu un vaste choix de capteurs, que ce soit du point de vue technologique, de la précision de la mesure ou de la récupération des données issues du capteur. Après avoir recherché dans le matériel disponible au laboratoire, deux composants sortent du lot: le DS 1621 et LM 335 car ils sont faciles d'utilisation. Nous avons finalement choisi le LM 335.

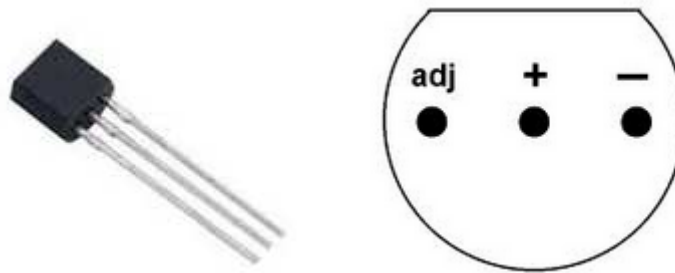


Figure 8: Capteur de température LM 335

Le LM335 est un capteur de température qui délivre entre ses bornes une tension proportionnelle à la température absolue. Pour cela, il doit être traversé par un courant dont l'intensité est telle que :

$$400 \mu A < I < 5 mA$$

On a alors $U = a.T$ avec $a = 10 mV.K^{-1}$

Le LM 335 appartient à une famille de capteurs qui se distinguent par leurs plages d'utilisation :

LM135 : $-55^{\circ}C < T < +150^{\circ}C$

LM235 : $-47^{\circ}C < T < +125^{\circ}C$

LM335 : $-40^{\circ}C < T < +100^{\circ}C$

Modules d'affichage à LED

Les matrices à LED RGB 32x16



Figure 9: Modules d'affichage à LED RGB 32x16

Ces matrices requièrent 12 ou 13 broches numériques (6 bits de données, 6 ou 7 bits de contrôle)

Les bits de données: 2 rouges, deux verts, deux bleus

Les bits de contrôle: A, B, C, LAT.

Il est à noter que ces modules sont conçus de sorte à être piloté par des FPGA ou autres processeurs tournant à grande vitesse.

Il faut rappeler que pour ces matrices le câble d'alimentation est séparé de la connexion de données ce qui rend le câblage plus facile.

Ces panneaux ont besoin d'une tension de 5V et jusqu'à 2A par panneau.

Il existe deux types différents de connecteurs d'alimentation.

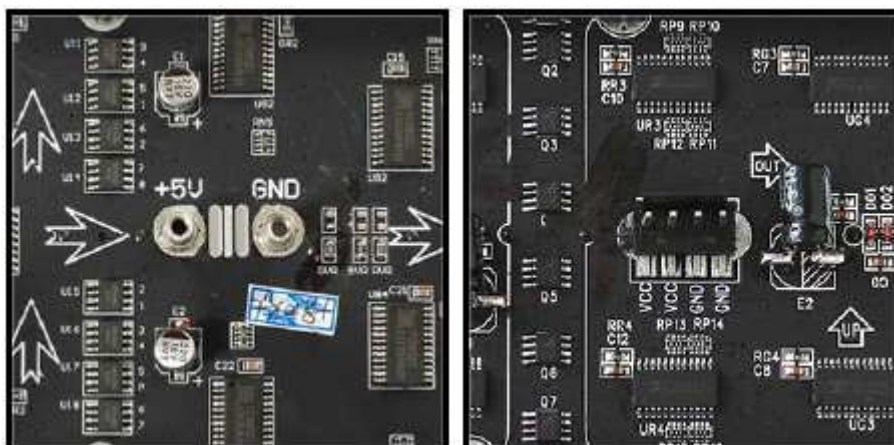


Figure 10: Connecteurs d'alimentation
ANNEE UNIVERSITAIRE 2013- 2014



3.2.2. Conception du prototype et fabrication du circuit électronique

Nous avons réalisé des prototypages de la carte électronique pour élaborer une carte qui répond à nos attentes.

Nous avons travaillé sur le prototypage de la carte de sorte à respecter le cahier des charges que nous avons établi.

Nous présentons par la suite les solutions envisagées jusqu'à l'arrivée au résultat et ce en suivant les différentes étapes de conception puis de fabrication de carte électronique.

Première solution:

Composant mis en oeuvre

PCF8574 ;

PCF8591 ;

PCF8583

Présentation des composants PCF

PCF8574

Ce composant fait le lien entre des entrées ou des sorties de type logique, au niveau TTL et le transport des informations portées par ces entrées ou ces sorties, par un bus I2C

Le composant est dit quasi bidirectionnel car une broche TTL peut jouer soit le rôle d'une entrée soit le rôle d'une sortie alternativement mais pas les deux à la fois.

Le composant réalise donc, suivant les cas :

Une sérialisation, lors de l'entrée, c'est à dire le passage d'informations sous forme parallèle (octet) à une information sous forme série (I2C).

Une dé-sérialisation, lors des sorties, c'est à dire le passage de série (I2C) à parallèle (octet).

Communication avec le composant PCF8574

La communication se fait de maître à esclave par le bus I2C. Le composant ne peut jouer le rôle de maître.

La communication s'opère à l'aide de l'envoi de deux octets sur la liaison I2C:

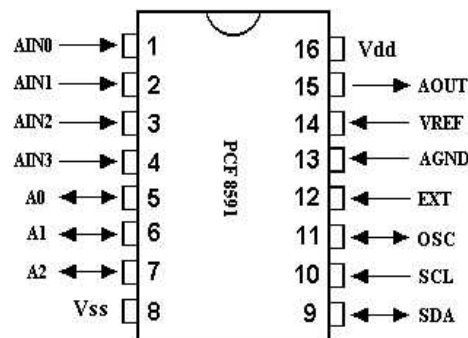
Un octet d'adresse, qui permet la sélection du composant.

Un octet de données qui porte les ordres du maître dans le cas d'une écriture ou l'état des entrées du composant dans le cas d'une lecture.



PCF8591

Le PCF8591 supporte 4 entrées et une sortie analogique, toutes d'une résolution de 8 bits. Les Broches AIN0 à AIN3 de l'image qui suit correspondent aux entrées analogiques et la broche AOUT à la sortie analogique. Les broches VREF et AGND correspondent aux tensions de références pour les conversions dans un sens ou dans l'autre. Ces valeurs peuvent être différentes de la tension d'alimentation Vdd et Vss (qui est généralement de 5V) mais doivent toutefois se tenir dans l'intervalle de ces dernières. La broche EXT à Vss permet d'indiquer l'utilisation de l'oscillateur interne pour les conversions numériques, le signal est alors disponible sur la Broche OSC. Inversement, la broche EXT à VDD indiquera utilisation d'un oscillateur externe qui devra être appliqué à la broche OSC (L'utilisation d'un oscillateur externe est sans intérêt en usage courant)



L'adresse Du PCF 8591 est codifiée sur 7 bits avec un 8em bit (R/W) indiquant le sens du transfert. Cette adresse est constituée d'une partie fixe et d'une partie «programmable» à partir des broches A0 à A2, ce qui permet d'envisager le raccordement de 8 circuits de ce type sur un même bus I2C. Le pilotage du circuit s'effectue par la lecture ou l'écriture d'une série de données sur le Bus I2C, mais les choses sont un peu plus complexes que dans le cas du PCF8574.

1	0	0	1	A2	A1	A0	R/W
---	---	---	---	----	----	----	-----

PCF8591 Plage : 144 a 158 (sur 8 bits)

PCF8583

RTC = Real Time Clock = Horloge temps réel

Une horloge temps réel est un composant de type "oscillateur / base de temps" qui fonctionne de façon autonome, avec un quartz pour une bonne précision de la fréquence d'oscillation.



Ce type de composant équipe entre autres les cartes mère des ordinateurs, et permet de disposer à tout instant (et sur requête) des informations Date et Heure.

Si l'alimentation du composant est assurée lors des coupures secteur (avec une pile, un accu ou un condensateur de forte valeur), les informations sont conservées et continuent d'être régulièrement mises à jour.

Ce type de composant peut être utilisé dans tout système où la date et/ou l'heure sont utilisées.

L'horloge temps réel PCF8583 dont il est question ici dispose d'une interface de type I2C, et le dialogue avec elle s'effectue grâce à un microcontrôleur utilisé en maître.

Schéma eagle

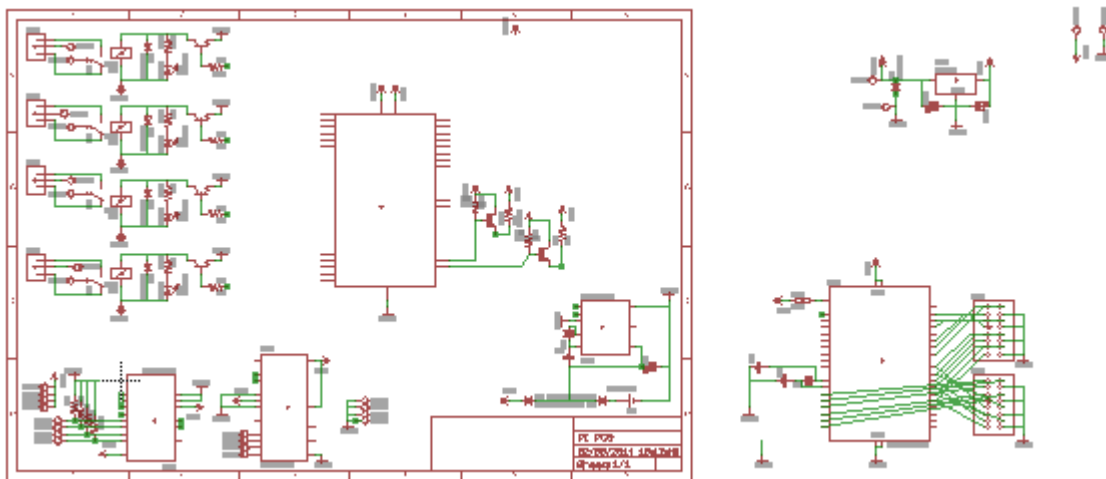


Figure 11: Schémas eagle

Nous avons utilisé quatre relais chacun sera relié à un capteur pour la mesure d'un paramètre particulier de la salle informatique.

Un capteur de température un capteur d'humidité, un capteur de bruit et un capteur d'inondation.

Les capteurs communiquent avec la Raspberry Pi via le composant PCF8574.



Donc la mesure ou la récupération des données issues des capteurs passe à travers la liaison I2C.

Le bus I2C est un bus de type série permettant la communication entre composants électroniques. Le succès de ce bus repose sur le nombre de composants disponibles.

Pour ce faire, ce bus utilise seulement trois fils :

- Un signal de donnée SDA (Sérial Data = Transfert des données bi-directionnel, synchrone avec SCL)
- Un signal d'horloge SCL (Sérial Clock = Horloge de transmission synchrone, fournie par le micro contrôleur et commune à tous les récepteurs)
- Un référentiel (Masse)

Les dispositifs qui viennent se connecter au bus se raccordent en parallèle sur les lignes SDA et SCL.

La discussion sur un bus se fait entre un maître et un esclave. C'est le maître qui demande et l'esclave qui répond. Un esclave peut être maître si sa structure interne le permet. Dans notre cas la Raspberry est le maitre et le PCF 8574 est l'esclave.

Typon

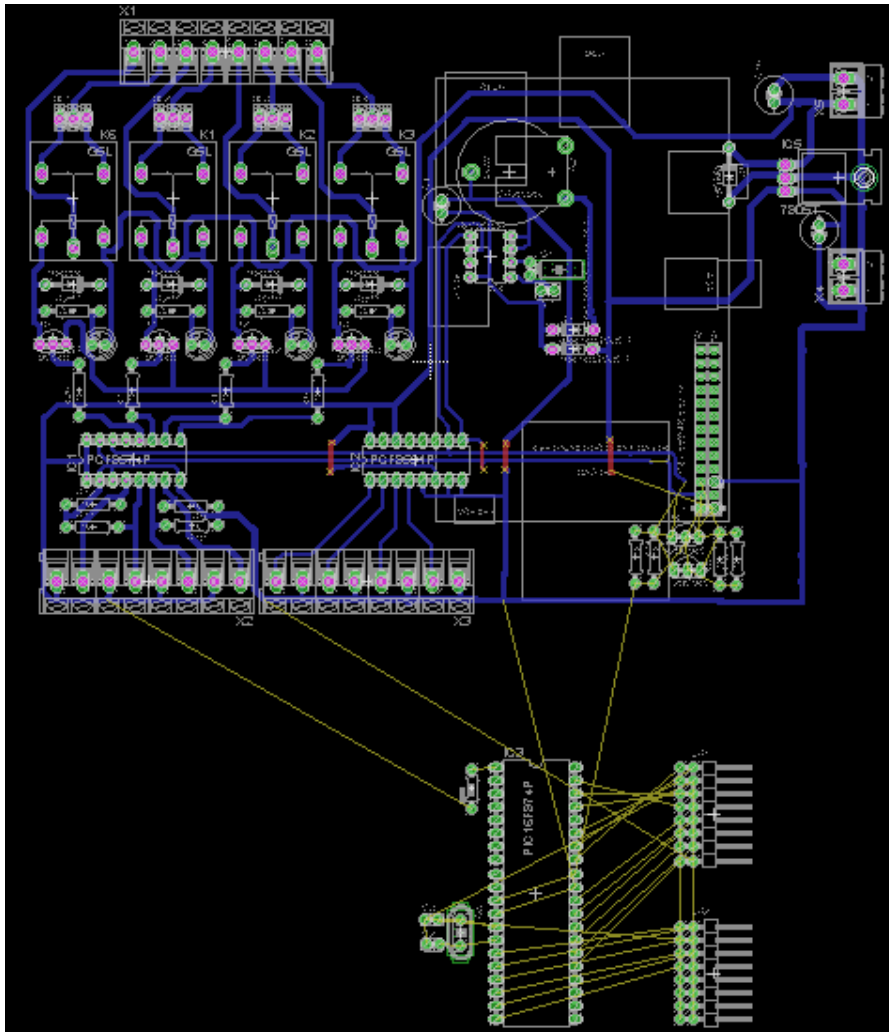


Figure 12: Typon de la carte

Réalisation de la carte

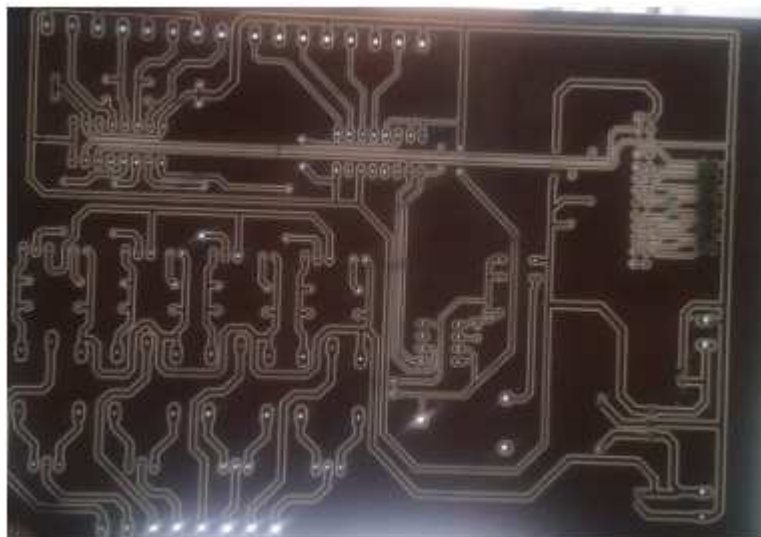




Figure 13: Carte électronique

Le travail avec la carte Raspberry seul ne présente pas des résultats satisfaisants au niveau de l'affichage sur les matrices à LED et il serait préférable d'avoir un module d'affichage à part entière, ce choix s'explique par la vitesse aussi la disponibilité du processeur ARM présent sur la carte Raspberry. Le pic se chargera de ce fait de traiter la partie affichage De plus les composants PCF existent en petite quantité sur le marché.

Deuxième solution:

Composants mis en œuvre

PIC16F874P ;

74AC574N

Schéma eagle

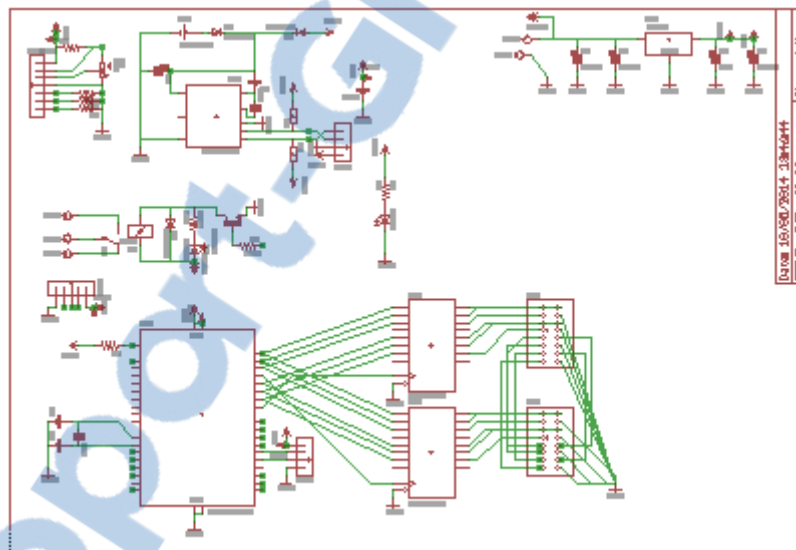


Figure 14: Schéma eagle

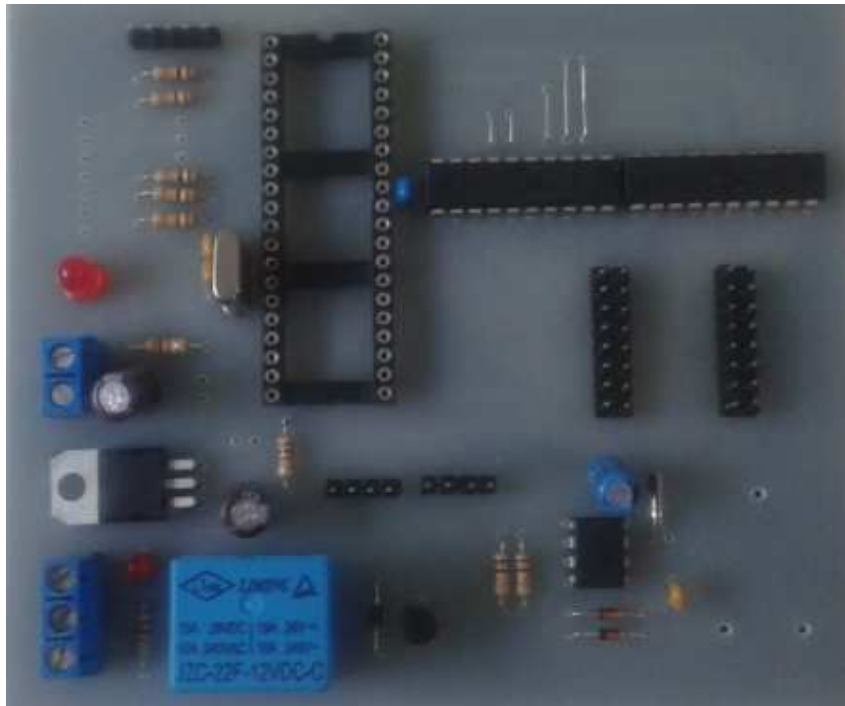


Figure 16: Carte électronique

Ce circuit ainsi conçu permet au PIC16F874P de commander quatre matrices à LED en parallèle. La Raspberry Pi envoie au pic l'ordre d'écriture sur les modules d'affichage via la liaison RS232 aussi les messages à afficher. Le pic reçoit les données et les présente sur son portB auquel est connecté le composant 74AC574.

Ce dernier commande par une horloge CK et une entrée de validation de sortie, sur le front montant de l'horloge la sortie prend l'état logique qui était en entrée.

Le circuit prévient la commande d'un appareil sur le relais, la Raspberry Pi envoie au pic encore une fois les commandes pour l'activation de l'appareil il pourrait s'agir par exemple de l'activation d'un climatiseur.

Adaptation des niveaux électriques

Le problème consiste en la connexion du PIC16F874P fonctionnant en 5 volts sur la carte Raspberry Pi fonctionnant en 3.3 volts. Etant donné que les deux cartes doivent communiquer ensemble, un adaptateur du niveau électrique sera requis puisqu'on envisage une communication RS232 entre pic et la carte Raspberry Pi. Par la suite il faut faire attention à ne pas provoquer d'incident avec un risque de griller les composants les plus sensibles.

Le cas que nous allons présenter est une communication RS232 entre le pic et la carte Raspberry Pi. Il y a un piège: la carte Rpi s'alimente aussi en 5 volts, on pourrait donc croire que l'on va relier les pattes du connecteur d'extension correspondant à l'RS232 aux pattes RS232 du pic.

Les signaux GPIO sont en 3.3 volts, qui est la tension de la puce ARM11 qui équipe la Raspberry Pi et sur laquelle ces signaux sont connectés. Il y a bien une conversion du 5 volts en 3.3 volts pour l'alimentation, mais toute la logique est ensuite en 3.3 volts.

Pour résoudre le problème exposé nous avons choisi d'utiliser le BSS138

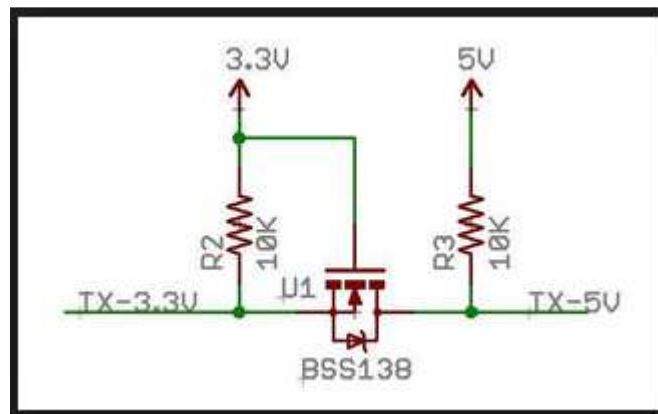


Figure 17 : Montage du BSS138

Ce sont de simples transistors mais avec une belle astuce pour proposer l'adaptation des niveaux électrique.

3.2.3. Programmation

Programmation du pic en C

On va se contenter de donner le sous programme d'affichage.

Sous programme d'affichage

Le sous programme d'affichage sera donné en annexe

Programmation de la Raspberry Pi

Avant toute chose nous avons installé la bibliothèque BCM2835 pour la gestion du GPIO et la bibliothèque Gcc pour la prise en charge de programme C.

Bibliothèque GPIO BCM2835

Il existe plusieurs librairies qui permettent de prendre le contrôle du GPIO. Nous avons utilisé la librairie BCM2835. C'est une bibliothèque C pour la Raspberry Pi donnant accès au GPIO et à d'autres fonctions d'entrée sortie.





Nous avons d'abord commencé par l'installation de la librairie.

download the latest version of the library, say bcm2835-1.xx.tar.gz, ensuite:

```
tar zxvf bcm2835-1.xx.tar.gz
```

```
cd bcm2835-1.xx
```

```
./configure
```

```
make
```

```
sudo make check
```

```
sudo make install
```

```
pi@raspberrypi:~/bcm2835-1.36
pi@raspberrypi ~/bcm2835-1.36 $ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking for style of include used by make... GNU
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking dependency style of gcc... gcc3
checking for clock_gettime in -lrt... yes
checking for doxygen... no
configure: WARNING: Doxygen not found - continuing without Doxygen support
checking for ranlib... ranlib
checking for gcc... (cached) gcc
checking whether we are using the GNU C compiler... (cached) yes
```

Figure 18: Installation de la bibliothèque BCM2835

Gcc 4.8 on Raspberry Pi Wheezy

Nous rappelons les commandes d'installation

```
sudo nano /etc/apt/sources.list
```



```
deb http://mirrordirector.raspbian.org/raspbian/ wheezy main contrib non-free rpi
deb http://archive.raspbian.org/raspbian wheezy main contrib non-free rpi
# Source repository to add
deb-src http://archive.raspbian.org/raspbian wheezy main contrib non-free rpi
deb http://mirrordirector.raspbian.org/raspbian/ jessie main contrib non-free rpi
deb http://archive.raspbian.org/raspbian jessie main contrib non-free rpi
# Source repository to add
deb-src http://archive.raspbian.org/raspbian jessie main contrib non-free rpi
```

```
sudo nano /etc/apt/preferences
```

```
Package: *
Pin: release n=wheezy
Pin-Priority: 900
Package: *
Pin: release n=jessie
Pin-Priority: 300
Package: *
Pin: release o=Raspbian
Pin-Priority: -10
```

```
sudo apt-get update
```

```
sudo apt-get install -t jessie gcc-4.8 g++-4.8
```

```
sudo update-alternatives --remove-all gcc
sudo update-alternatives --remove-all g++
```

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.6 20
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 50
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.6 20
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.8 50
```

Programme



Il s'agit ici de programme C sur la Raspberry Pi qui permettra la gestion d'événements, comme on le verra plus tard dans la partie développement Web une page web sera créée afin de gérer les événements.

Le code C que nous avons écrit va boucler sur la table événement de la base de données et exécutera les actions rencontrées suite à l'activation des événements.

Si l'événement devient vrai alors on passe à l'exécution de l'action associée.

3.2.4. Développement web

Dans cette section il s'agit de la mise en place de pages web dynamiques pour le contrôle en temps réel des paramètres tels que la température et l'hygrométrie dans les salles informatiques, aussi l'application devra permettre la gestion des événements majeurs qui poussent une action à activer un climatiseur, à envoyer un mail ou des messages text par exemple.

Objectif du projet

Le projet a pour but de relever la température via un capteur tous les cinq minutes par exemple, et ces mesures sont enregistrées dans une base de données MySQL avec la table: temp pour la température, cette table est reliée à la Raspberry Pi.

On va faire un graphique permettant d'établir le profil thermique de la salle informatique ou sera installé le système, sur une page web codée en PHP reprenant automatiquement les valeurs dans la base de données MySQL.

Aussi l'application devra prendre en charge la gestion des événements, l'utilisateur doit indiquer l'événement, la valeur de l'événement, l'action et la valeur qui lui est associée.

Le programme C écrit sur la Raspberry se chargera alors de l'exécution des actions si les événements sont vrais. Par rapport à l'événement il pourrait s'agir par exemple du dépassement des limites de fonctionnement en température et l'action correspondante pourrait être l'activation d'un appareil.

Choix technique

Le choix des outils a été une étape décisive dans la réalisation du projet. Le choix s'est porté sur le langage PHP et le gestionnaire de base de données MySQL.

Conception du système

Analyse conceptuelle des données

La création d'une base de données est nécessaire.



La base de données contiendra les différentes données relatives aux: Logs administrateurs, température, humidité, événements.

Dictionnaire de donnée

Attribut	Type	Description
Id	Numéro_auto	Identifiant d'un administrateur
Login	Varchar	Log administrateur
Pass	Varchar	Mot de passe administrateur
nom_pren	Varchar	Nom et prénom administrateur
Id	Numéro auto	Identifiant de la température
val_temp	Varchar	Valeur de la température
Timestamp	Bigint	Date et heure d'enregistrement de la température
id_humidity	Numéro_auto	Identifiant de l'humidité
Date	Date	Date d'enregistrement
humidity_val	Int	Valeur de l'humidité
Id	Numéro auto	Identifiant de l'événement
Event	Int	Evénement
consigne_event	Varchar	Valeur de l'événement
Action	Varchar	Action
consigne_action	Varchar	Valeur de l'action

Tableau 1: Dictionnaire de données

Modèle conceptuel de données

Après avoir classé les données, il a été possible de dégager les différentes entités et leurs attributs:

Entité	Attributs
--------	-----------



Admin	id login pass nom_pren
-------	---------------------------------

Tableau2: Entité Admin

Entité	Attributs
Temp	id val_temp timestamp

Tableau3: Entité temp

Entité	Attributs
Humidity	id_humidity date humidity_val

Tableau 4 : Entité humidity

Entité	Attributs
Evénement	id event consigne_event action consigne_action

Tableau 5 : Entité événement

Outils utilisés

Pour réaliser le projet à l'aide de PHP/MySQL, deux outils principaux ont été utilisés: phpMyAdmin et Dreamweaver 8.

PhpMyAdmin

PhpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP. Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP.



Cette interface pratique permet d'exécuter, très facilement, de nombreuses requêtes comme les créations de table de données, les insertions, les mises à jour, les suppressions, les modifications de structure de la base de données. Ce système est très pratique pour sauvegarder une base de données sous forme de fichier « .sql » et ainsi transférer facilement ses données. De plus celui-ci accepte la formulation de requêtes SQL directement en langage SQL, cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.

Dreamweaver 8

Produit phare de Macromédia, Dreamweaver MX est aujourd'hui le logiciel incontournable de création de sites Web. En outre la nouvelle version 8 de Dreamweaver permet d'exploiter toute la puissance du couple PHP/MySQL. La puissance des éditeurs de codes et la fonction de prévisualisation de Dreamweaver sont les plus appréciés par les développeurs d'applications web.

C'est donc grâce à la plateforme de développement complète qui est offerte par PhpMyAdmin que l'application a été réalisée.

Installer un serveur web sur la Raspberry

Installation et gestion d'un serveur sur Raspbian

Nous avons choisi d'utiliser la Raspberry Pi comme un serveur web.

En choisissant la Raspberry, on a la possibilité de modifier les services à souhait (exemples: la taille du disque, l'hébergement de Database, etc.), ce qui n'est en général pas le cas chez les hébergeurs spécialisés, qui vendent souvent des hébergements mutualisés avec une faible capacité de configuration.

Installation du serveur Apache avec Raspbian

Tout d'abord, nous avons installé Apache, qui est le serveur web en tant que tel. Quand on parle de serveur web, on pense souvent à la machine, mais ce terme désigne aussi le logiciel qui permet à la machine d'analyser les requêtes d'un utilisateur (sous forme http), et de retourner le fichier correspondant à la requête (ou une erreur si le fichier n'est pas trouvé, ou la requête mal formulée).

Dans le cadre d'Apache, c'est donc du logiciel que l'on parle.

À l'heure actuelle, Apache est le serveur web le plus utilisé, avec environ 60% de parts de marché. Apache possède même sa propre licence, utilisée par de nombreux autres projets.



Que ce soit pour la Raspberry Pi et Raspbian, ou pour une machine plus généraliste, Apache est donc un choix sûr.

Installation d'Apache

Pour ce faire nous devons posséder les droits administrateurs, soit en étant connecté en root, soit via la commande sudo.

```
1 | sudo aptitude install apache2
```

Une fois l'installation terminée, nous avons testé qu'Apache fonctionne correctement en rendant sur l'adresse de la Raspberry.

Pour cela, nous avons tenté d'accéder à la Raspberry nous avons ouvert le navigateur web de la Raspberry (par défaut Midori sous Raspbian), et nous sommes allés à l'adresse « <http://127.0.0.1> ». Nous avons obtenu une page avec un message « It works! ».

Installation de PHP sur la Raspberry

Le PHP est un langage interprété. Et comme dans le cas des serveurs, l'acronyme PHP peut avoir plusieurs sens. En fait, quand l'on parle de PHP, on peut parler soit du langage, soit de l'interpréteur.

Ici, quand nous parlons d'installer PHP, cela signifie que nous allons installer l'interpréteur, afin d'utiliser le langage.

PHP (le langage cette fois) est principalement utilisé pour rendre un site dynamique, c'est-à-dire que l'utilisateur envoie des informations au serveur qui lui renvoie les résultats modifiés en fonction de ces infos. A contrario, un site statique ne s'adapte pas aux informations fournies par un utilisateur. Il est enregistré sous forme de fichier une fois pour toute, et livrera toujours le même contenu.

Le PHP est l'un des langages de programmation les plus utilisés, et c'est même le plus utilisé pour la programmation destinée au web, avec environ 79% de parts de marché.

Installer PHP

Nous avons fait appel à l'administrateur pour installer php5 avec la ligne de commande.

```
1 | sudo aptitude install php5
```

Une base de données MySQL

Maintenant nous allons mettre en place un SGBD (Système de Gestion de Base de Données), à savoir MySQL.



MySQL est un SGBD libre, puissant, massivement utilisé (environ 56 % de parts de marché des SGBD libres). Là encore, MySQL est tellement un incontournable du développement, quel qu'en soit la branche.

Installer MySQL

Pour ce faire, nous allons installer mysql-server et php5-mysql (qui servira de lien entre php et mysql)

```
1 | sudo aptitude mysql-server php5-mysql
```

Lors de l'installation de mysql-server, il nous est demandé un mot de passe pour le compte administrateur MySQL (root).

Le mot de passe que nous avons choisit pour notre cas est 2899100

PHPMysqlAdmin, gérer facilement ses bases de données

C'est une application développée en PHP, et qui vise à fournir une interface simplifiée pour MySQL.

Installer PHPMyAdmin

L'installation de PHPMyAdmin se fait très simplement, via le gestionnaire de paquet, en utilisant la commande suivante:

```
1 | sudo aptitude install phpmyadmin
```

PHPMysqlAdmin pose plusieurs questions concernant ses réglages.

Nous avons choisit donc un serveur Apache, puis nous avons utilisé le choix par défaut pour la base (dbconfig-common). Pour le mot de passe root, il s'agit de celui que nous avons utilisé pour MySQL c'est-à-dire 2899100.

La mise en œuvre des interfaces web

La base de données raspberry





Figure 19: Base de données sur PhpMyAdmin

Pages web créés

Page identification

Pour l'identification de l'administrateur

Cette page permettra à l'administrateur de s'identifier en introduisant son login et son mot de passe.

Figure 20: Page identification

Un message d'erreur s'affiche si les informations saisies sont erronées

Figure 21: Message d'erreur

Un message s'affiche si les champs ne sont pas tous remplis



Veillez Remplir tout Les Champs

OK

Figure 22: Message de validation de champs

Page journal de température



Figure 23: Page journal de température

La température actuelle reprend automatiquement de la base de données la valeur du dernier enregistrement

Graphique de température



L'utilisateur de l'application peut afficher le graphique de la température des trois dernières heures

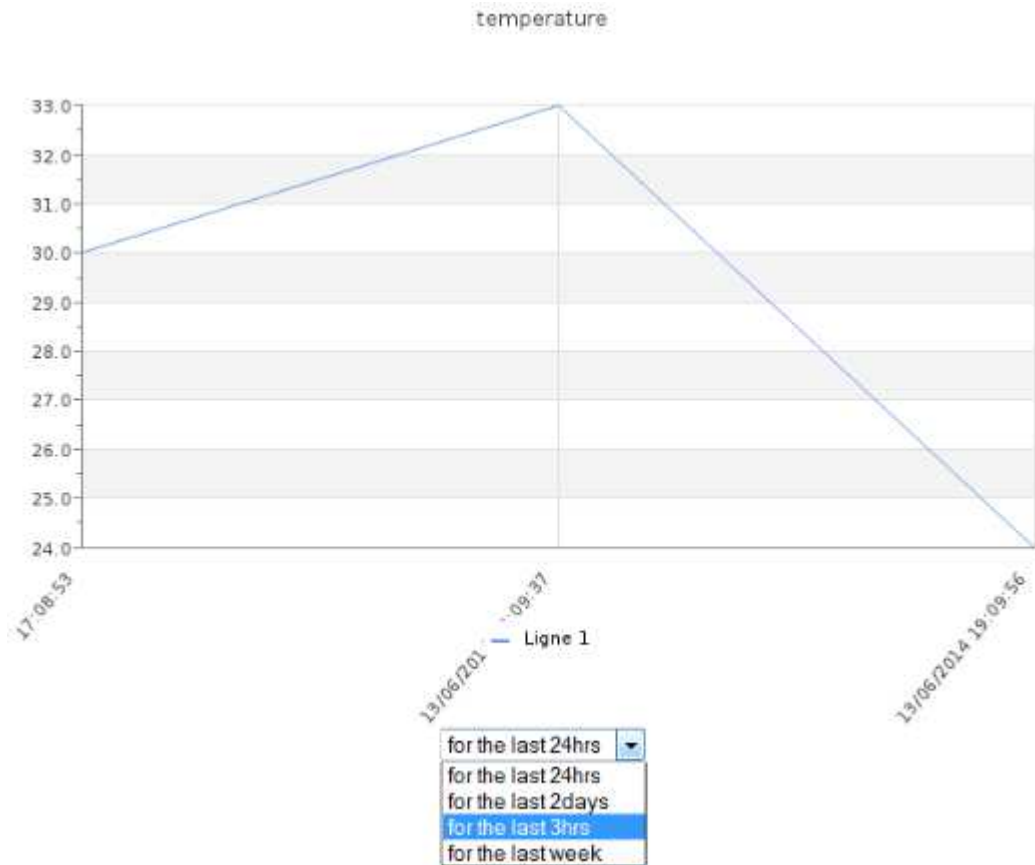


Figure 24: Graphique des dernières trois heures

Dernières 24 heures

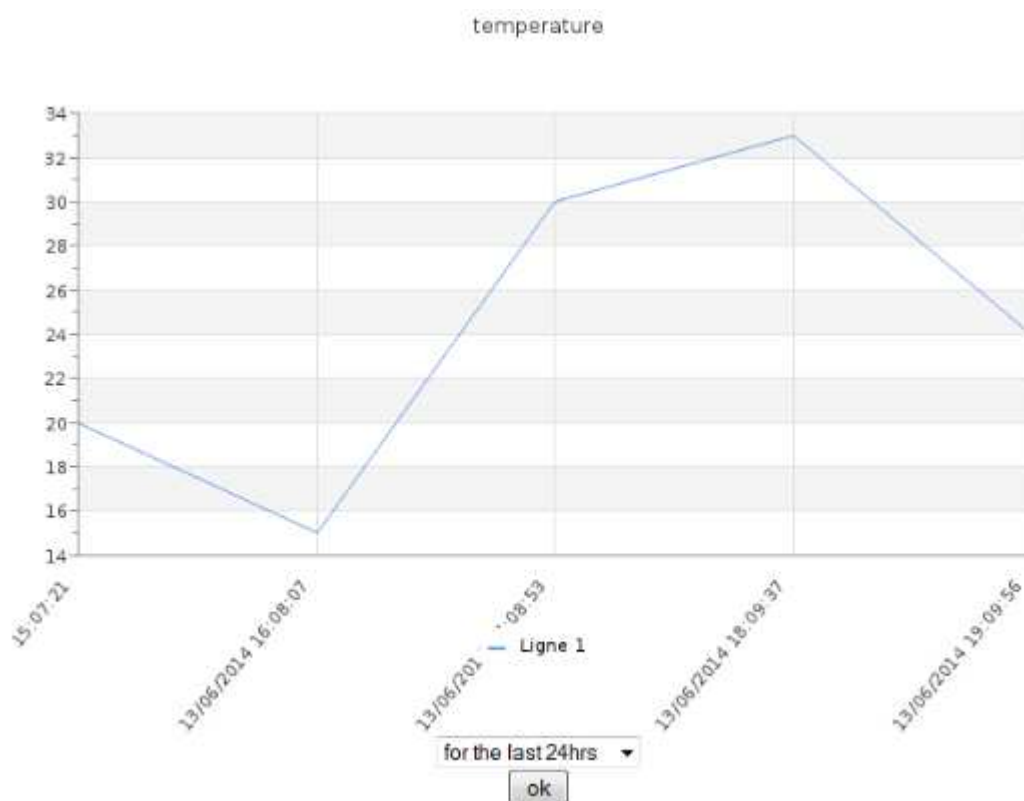


Figure 25: Graphique des dernières 24 heures

Relevés de températures tout les cinq minutes

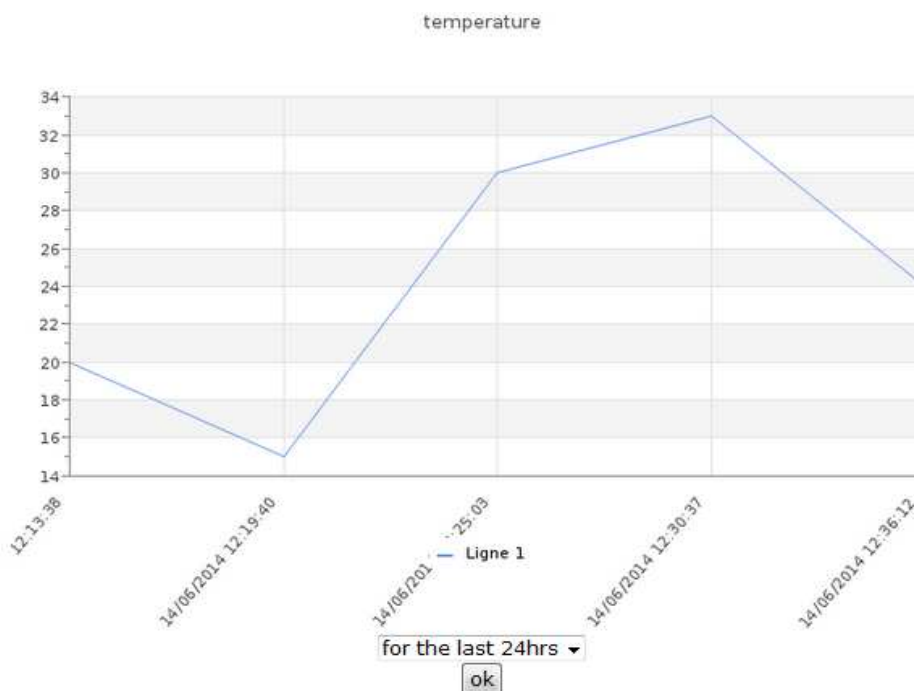


Figure 26: Graphique de température avec des relevés tous les cinq minutes

Gestion des utilisateurs



Liste des administrateurs système

Nom et prénom	Identifiant	Mot de passe	Action
djouri.loubna	Intellifex	djouri.loubna	
yassine.abouch	yassine.abouch	Intellifex	

Ajouter un administrateur

Nom et prénom
Identifiant
Mot de passe

Enregistrer

Figure 27: Page gestion des utilisateurs

Gestion des événements



Page: 1 2 Suivant >

Liste des événements				
Événement	Consigne	Action	Consigne	Gérer
Température inférieure à	xxx	Send email	la température courante est de 40 xxx degré	

Ajouter un nouveau événement

Événement	Sélectionnez...	Valeur
Action	Sélectionnez...	Valeur

Enregistrer

Figure 28: Page gestion des événements

3.3. Mise en place du système





Figure 29: Mise en place du système partie affichage sur modules à LED graphiques
Il s'agit dans la figure 28 de l'affichage de la température sur les matrice à LED

Conclusion et perspectives

Dans ce rapport nous avons suivi les étapes de conception d'un système de gestion de la température dans les salles informatiques. Le choix de la plateforme embarqué c'est posé sur la Raspberry Pi qui a toujours l'avantage de la popularité, donc plus de logiciels et de support.

La méthode étudiée dans ce projet peut être améliorée, ainsi le système conçu peut être remplacé par un système mettant en œuvre un seul processeur au lieu des deux cartes d'extension et par suite des deux processeurs ARM et PIC communicant en RS232.

On pourrait utiliser la plateforme raspberry seul et connecter le capteur directement pour le relever de température. Le projet peut être compléter par le relevé des autres paramètres de la salle informatique citant par exemple l'humidité, le bruit et l'inondation.

On pourrait proposer des modifications au niveau de l'application web.

Dans le cadre d'amélioration toujours on pourrait remédier à une communication wifi entre les deux cœurs pour considérer le système comme étant Client/serveur. On pourrait aussi tenter une communication entre carte Raspberry et Arduino.

Outre l'enrichissement technique, le stage m'a permis de découvrir de près le fonctionnement d'une entreprise, de découvrir sa complexité et son harmonie, de constater



l'importance du travail collectif et le rôle d'une gestion maîtrisée. D'autre part, cette expérience m'a permis d'améliorer mon sens de communication, et surtout de m'aventurer dans le domaine de l'électronique et de l'informatique.

Références

Site Intelifex Systems :

www.intelifex.ma

Site du zéro pour la plate-forme Arduino UNO :

<http://fr.openclassrooms.com/sciences/cours/arduino-pour-bien-commencer-en-electronique-et-en-programmation>

Site Arduino :

www.arduino.cc

<http://fr.openclassrooms.com/sciences/cours/arduino-pour-bien-commencer-en-electronique-et-en-programmation>

Affichage dynamique wikipedia :

http://fr.wikipedia.org/wiki/Affichage_dynamique

Cours sur les systèmes embarqués :

<http://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-introduction.htm>

Présentation du bus I2C.htm

Plate-forme Adafruit 32x16 and 32x32 RGB LED Matrix

Site du zéro pour la conception de site web dynamique

<http://www.raspberrypi.org/forums/viewtopic.php?f=65&t=70060>

Site officiel Raspberry Pi



```
void GetTemp(){
OW_Reset(&PORTA,0); // Onewire reset signal
OW_Write(&PORTA,0,0xCC); // Issue command SKIP_ROM
OW_Write(&PORTA,0,0x44); // Issue command CONVERT_T
Delay_us(120);

OW_Reset(&PORTA,0);
OW_Write(&PORTA,0,0xCC); // Issue command SKIP_ROM
OW_Write(&PORTA,0,0xBE); // Issue command READ_SCRATCHPAD
Delay_ms(400);

j = OW_Read(&PORTA,1); // Get temperature LSB
temp = OW_Read(&PORTA,1); // Get temperature MSB
temp <<= 8; temp += j; // Form the result
}

void Lettre_Print(char ASci,char I_X,char I_Y,char RGB){

char Xii,Yii;
char bChar; char Val;
RGB = RGB<<1;
if (ASci==32){ASci=58;}

for(Xii=0;Xii<8;Xii++){
for(Yii=0;Yii<8;Yii++){
if(ASci>64){
bChar = GetBit(lettre[ASci-65][Yii+8],Xii,RGB)<<4;
Val = bChar + GetBit(lettre[ASci-65][Yii],Xii,RGB) ;

RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F1= Val.F5;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F2= Val.F6;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F3= Val.F7;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F5= Val.F1;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F6= Val.F2;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F7= Val.F3;

}
else
{
bChar = GetBit(chifre[ASci-48][Yii+8],Xii,RGB)<<4;
Val = bChar + GetBit(chifre[ASci-48][Yii],Xii,RGB);

RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F1= Val.F5;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F2= Val.F6;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F3= Val.F7;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F5= Val.F1;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F6= Val.F2;
RAM_AFFICHAGE[Yii+I_Y][Xii+I_X].F7= Val.F3;

}
}
}
int ind;
```



```
int Lint;
int fint;

void SCREEN(char li);
void interrupt(void);

int contt;
char lii=0;

void interrupt () {
    SCREEN(lii);
    if(lii<8)lii++;else lii=0;
    intcon.TMR0IF=0;
}

unsigned int Read_ADC(unsigned short channel)
{
    unsigned int result;

    if (channel > 12 ) {return 0;} //designed for 18F4550
    ADCON0 = channel << 2; // channel set, no go, no start
    ADCON2 = 0B00000111; // left aligned, internal RC
    ADCON0.F0 = 1; // AD module enabled
    delay_us(3); // wait for conversion start
    ADCON0.F1 = 1; // Conversion start
    while(ADCON0.F1){} // wait to finish
    ADCON0.F0 = 0; // disable module
    result = ADRESH;
    result <<= 2;
    result |= ADRESL & 0B00000011; //weed out some bit-noise
    return result;
}

void printTEmp(int Val){
    int D1,D2,D0;
    D0 = VAL/100 ;
    D1= Val/10 - D0 *10 ;
    D2 = Val - D0*100 - D1 * 10 ;

    Lettre_Print(48+D0,0,0,1);
    Lettre_Print(48+D1,8,0,1);
    Lettre_Print(48+D2,16,0,4);

}

void main() {
    TRISB=0x00;
    TRISA=0x01;
    TRISC=0x00;
    TRISD=0x00;
```



```
OE =1;
```

```
INTCON.gie=1;  
intcon.TMR0IE=1;  
TOCON.t0cs=0;  
TOCON.psa=0;  
TOCON.T0PS0=1; //1  
TOCON.T0PS1=1; //1  
TOCON.T0PS2=0; //0
```

```
while(1){
```

```
    printTEmp ( Read_ADC(0)-773);  
    delay_ms(10);
```

```
    }
```

```
}
```

```
void SCREEN(char li){
```

```
    int Xi=0;  
    int Ll=0;  
    for(Xi=0;Xi<32;Xi++){
```

```
        PORTB = RAM_AFFICHAGE[Li][Xi] & 0b11101110;  
        PORTA = RAM_AFFICHAGE[Li][Xi] & 0b11101110;
```

```
        MATRICE_H=1;  
        MATRICE_H=0;
```

```
        PORTB = RAM_AFFICHAGE[Li+16][Xi] & 0b11101110;  
        PORTA = RAM_AFFICHAGE[Li+16][Xi] & 0b11101110;
```

```
        MATRICE_B=1;  
        MATRICE_B=0;
```

```
        S = 0;  
        S = 1;  
        }  
        A = Li.F0;  
        B = Li.F1;  
        C = Li.F2;  
        L = 1;L = 0;  
        OE = 0;
```

```
    }
```

Code PHP page de journal de température

```
<?php
```

```
include '../connect_db.php';
```

```
$messagesParPage=1; //Nous allons afficher 5 messages par page.
```



```
//Une connexion SQL doit être ouverte avant cette ligne...
$retour_total=mysql_query("SELECT COUNT(*) AS total FROM temp "); //Nous
récupérons le contenu de la requête dans $retour_total
$donnees_messages_total=mysql_fetch_assoc($retour_total); //On range retour sous la
forme d'un tableau.
$total=$donnees_messages_total['total']; //On récupère le total pour le placer dans la
variable $total.
if ( $total > 0 ) {
//Nous allons maintenant compter le nombre de pages.
$nombreDePages=ceil($total/$messagesParPage);
if(isset($_GET['page'])) // Si la variable $_GET['page'] existe...
{
$pageActuelle=intval($_GET['page']);
if($pageActuelle>$nombreDePages) // Si la valeur de $pageActuelle (le numro de la page)
est plus grande que $nombreDePages...
{
$pageActuelle=$nombreDePages;
}
}
else // Sinon
{
$pageActuelle=1; // La page actuelle est la n1
}
$premiereEntree=($pageActuelle-1)*$messagesParPage; // On calcul la première entre lire
// La requête sql pour récupérer les messages de la page actuelle.
$retour_messages=mysql_query("SELECT * FROM temp ORDER BY id DESC LIMIT
'.$premiereEntree.', '$messagesParPage.");
$xx=0;
// On récupère les 5 dernières demandes
$retour = mysql_query("SELECT * FROM temp ORDER BY id DESC LIMIT 0, 1 ");
```





```
{
echo "<a href='?page=$prev'> < Pr&eacute;c&eacute;dent </a> "; //Pour l'affichage, on
centre la liste des pages
}
if (empty($_GET['page']) or $prev < 1)
{
echo "";
}
for($i=1; $i<=$nombreDePages; $i++) //On fait notre boucle
{
//On va faire notre condition
if($i==$pageActuelle) //Si il s'agit de la page actuelle...
{
echo ' <a href="#" class="active">'. $i. '</a> ';
}
else
{
echo ' <a href="index.php?page='.$i.">'. $i. '</a> ';
}
}
$next = (isset($_GET['page']) ? $_GET['page'] : 1) + 1;
if ($next > $nombreDePages)
{echo "};}
else
{
echo "<a href='?page=$next'>Suivant ></a>";
}
echo '</div></div></div></div>';

//-----
}

// Fin de la boucle
```



```
?></td>
</tr>
</table>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
</center>
<div align="center"><?php
if(isset($_POST['ok']))
{
echo"<div align='center' height='450px' width='600px'>";
if($_POST['time']==1)
echo"<img src='jpgraph/graph.php'>";
if($_POST['time']==2)
echo"<img src='jpgraph/graph2.php'>";
if($_POST['time']==3)
echo"<img src='jpgraph/graph3.php'>";
if($_POST['time']==4)
echo"<img src='jpgraph/graph4.php'>";
"</div>";
}
echo"<form method=post name='form' action=" "><select name='time'>
<option value='1'>for the last 24hrs</option>
<option value='2'>for the last 2days</option>
<option value='3'>for the last 3hrs</option>
<option value='4'>for the last week</option>
</select><br/><input type='submit' name='ok' value='ok'></form><br/>";
?>
<br class="clearfix" />
</div>
</div>
```



<div id="page-bottom">

<div align="center"></div>

</div>

</div>

</body>

</html>

Rapport-Gratuit.com