



Mémoire de Projet de fin d'étude

Préparé par

BOUAMOR Boujema

OUCHA Mohamed

Pour l'obtention du diplôme

Ingénieur d'Etat en

SYSTEMES ELECTRONIQUES & TELECOMMUNICATIONS

Intitulé

Conception et réalisation d'un système de surveillance d'une salle des serveurs à base de raspberry pi

Encadré par :

Pr RAZI Mouhcine

Mr AADIL Alaa (Groupe MANAGEM)

Soutenu le **21 Juin 2016**, devant le jury composé de :

Pr RAZI Mouhcine :Encadrant

Pr GHENNIQUI Hicham..... : Examineur

Pr LAMHAMDI Tijani..... :Examineur

DEDICACE

A nos chères mères qui ont toujours été près de nous pour nous écouter et nous soutenir.

Vous n'avez jamais épargné le moindre effort pour nous aider et nous encourager.

Veillez trouver en ce travail la preuve de notre amour et de notre estime pour vous.

A nos chers pères. Vous êtes toujours pour nous les exemples à suivre.

Quisse ce travail exprimer le respect et l'amour que nous vous portons.

A nos frères et sœurs pour l'esprit de fraternité et de communion et le soutien physique et moral, Que dieu vous protèges.

A toutes les personnes chères à nos cœurs, pour leurs amour, leurs disponibilités, voire pour tout l'aide et le support perpétuels que vous nous avez prodigués.

A nos familles, nos amis, toutes celles et tous ceux qui ont contribué de près ou de loin à cet humble travail. Que ceci puisse être le témoignage de notre plus haute reconnaissance

BOUAMOR Boujemaa & OUCHA Mohamed

REMERCIEMENTS

Tout d'abord, nous tenons à remercier toute l'équipe du service de la Direction des Systèmes d'Information pour leur accueil chaleureux et leur collaboration.

Nous remercions également notre encadrant professionnel Monsieur **AADIL Alaa** ingénieur de développement informatique au sein de la direction des systèmes d'information, pour nous avoir dirigé et fournis toutes les informations nécessaires, pour sa disponibilité, malgré leur nombreuses responsabilités, ainsi que les efforts et les conseils qu'il nous a apportés lors des différents suivis afin que notre stage se déroule dans des bonnes conditions.

Nos remerciements sont adressés aussi à **Mr. RAZI Mouhcine** notre encadrant au sein de la **Faculté des Sciences et Techniques de Fès** pour son soutien et suivi tout au long de la durée de stage.

Enfin, nous tenons, au terme de ce travail, à présenter nos vifs remerciements à toutes les personnes qui ont contribué, de près ou de loin, à son bon déroulement.

RESUME

Le présent rapport décrit le travail réalisé en guise de **Projet de Fin des Etudes** durant quatre mois au sein de la société Groupe MANAGEM site de GUEMASSA Marrakech dans le cadre de la formation d'ingénieur Systèmes Electroniques et Télécommunication à la **Faculté des Sciences et Techniques de Fès**.

La salle des serveurs du site Backup de GUEMASSA est la salle de gestion de parc informatique de tout le site GUEMASSA, l'importance de cette salle nécessite la mise en place d'un système de surveillance qui permettra d'informer l'administrateur du parc informatique de l'état de la salle des serveurs, et de l'alerter en cas de dysfonctionnement des climatiseurs qui assurent une température idéale de fonctionnement tout au long de l'année.

C'est dans ce cadre que notre projet de fin d'études vient pour répondre à ce besoin, la solution qui a été mise en place, est une carte électronique embarquée dotée des capteurs suivants : Température, Humidité, capteur de mouvement, et caméra embarquée. La carte électronique embarquée sera dotée d'un serveur web qui contiendra une base de données qui sera alimentée en permanence par l'état des capteurs.

LISTES DES FIGURES

Figure 1 : Fiche technique de MANAGEM.....	15
Figure 2: Structure du groupe MANAGEM.....	17
Figure 3: Schéma descriptif du site Backup.....	22
Figure 4 : Les composants standards d'un Raspberry.....	27
Figure 5: Différents modèles du Raspberry Pi.....	29
Figure 6: Les caractéristiques des modèles.....	29
Figure 7: Les caractéristiques du Raspberry Pi 2 Modèle B.....	30
Figure 8: Description des broches GPIO.....	31
Figure 9: Le module Grove Pi.....	33
Figure 10: Capteur température et d'humidité.....	34
Figure 11: Brochage du capteur SHC-SR501.....	34
Figure 12: schéma explicatif du fonctionnement du Jumper Set.....	35
Figure 13: Raspberry Pi Camera Rev 1.3.....	37
Figure 14 : Architecture de connexion client-serveur avec PHP.....	41
Figure 15: Adresse IP de la carte Raspberry Pi.....	44
Figure 16: fichier de configuration réseau.....	46
Figure 17: Logiciel Putty.....	47
Figure 18: Invite de commande via SSH.....	47
Figure 19: Logiciel TightVNC VIEWER.....	48
Figure 20 : Interface graphique sur logiciel TightVNC VIEWER.....	49
Figure 21: Brochage du Grove Pi sur le Raspberry.....	50
Figure 22 : Version de firmware de Grove Pi.....	50
Figure 23: Test de la communication Grove et Raspberry.....	51
Figure 24: Clignotement d'une LED.....	51
Figure 25: Programme de test du DHT11.....	52
Figure 26: Résultat de test du DHT11.....	52
Figure 27: Programme test du PIR.....	53
Figure 28: Résultat de test du PIR.....	53
Figure 29: Test de camera par motion.....	54
Figure 30: Programme test camera.....	54
Figure 31: Test de camera par Picamera.....	55
Figure 32: Schéma descriptif du système.....	56
Figure 33: Montage du système de collection de données et contrôle.....	57
Figure 34: Organigramme du programme principal.....	58
Figure 35 : Organigramme de la routine d'interruption.....	59
Figure 36: Les répertoires du chemin /home/pi/Desktop/pfe.....	60
Figure 37: Photos prises au moment des détections.....	60
Figure 38: Installation de mysql-serveur et php5 mysql.....	61
Figure 39: Structure de la base de données.....	61
Figure 40: Table « sensorInfo » de base de données « DB_MANAGEM ».....	62

Figure 41: Table « sensorData » de base de données « DB_MANAGEM ».....	62
Figure 42: Table « sensorData » remplis	63
Figure 43: Structure de tableau « utilisateur».....	63
Figure 44: Installation de serveur Apache2	66
Figure 45: Installation de l'interpréteur PHP	66
Figure 46: Répertoire de serveur apache sur Raspberry	67
Figure 47: Fichier de connexion	68
Figure 48:Appelle de la fonction de connexion.....	68
Figure 49: Page d'authentification	69
Figure 50: Page de Surveillance	70
Figure 51: Fonction PHP pour faire l'affichage du menu déroulant	70
Figure 52: Extrait de la page de supervision.....	71
Figure 53: Fonction PHP défini la plage de temps disponible sur la base de données.....	71
Figure 54: Extrait de fichier « sensor_data.php »	72
Figure 55: Graphe affiché	72
Figure 56: Graphe de déférentes grandeurs (température et humidité)	73
Figure 57: Interface du NetBeans	74
Figure 58: Bibliothèques importer	75
Figure 59: Fonction de connexion à la base donnée	75
Figure 60: Fenêtre d'authentification de L'application JAVA.....	76
Figure 61: Fenêtre principale de choix	76
Figure 62: La fenêtre de gestion de l'utilisateur	77
Figure 63: La fenêtre de Surveillance.....	78
Figure 64: Plage de date disponible dans la base de données.....	78
Figure 65: Graphe d'humidité salle serveur.....	79
Figure 66: D'autres systèmes d'exploitation compatibles avec Pi 2 Modèle B	82
Figure 67: le fichier (.zip) de Raspbian For Robots.....	83
Figure 68: écriture de l'image Raspbian For Robots sur la carte SD	84
Figure 69: interface de configuration au premier démarrage.....	85
Figure 70: La réussite de l'opération élargissement des fichiers du système	85
Figure 71: les options du choix «Internationalisation options ».....	86
Figure 72: interface graphique du Raspbian For Robots	87

LISTES DES TABLEAUX

Tableau 1: Les caractéristiques du capteur DHT11	33
Tableau 2 : Tableau comparatif entre Swing et JFace	38

LISTE DES ACRONYMES

DSI: Direction des Systèmes d'Informations

PIR : Passive Infra Red

AWT: Abstract Windowing Toolkit

OS: Operating System (Système d'exploitation)

GPLv2: Common Development and Distribution License

IDE: Integrated Development Environment

JDK: Java Development Kits

API: Application Programming Interface

SOMMAIRE

DEDICACE	2
REMERCIEMENTS.....	3
RESUME	4
LISTES DES FIGURES	5
LISTES DES TABLEAUX	7
LISTE DES ACRONYMES	7
SOMMAIRE.....	8
INTRODUCTION GENERALE	12
Chapitre I :	13
Introduction.....	14
I. Présentation de groupe MANAGEM.....	14
II. Liste signalétique du groupe MANAGEM.....	15
III. Historique.....	16
IV. Organigramme du groupe MANAGEM	17
V. Présentation de La Compagnie de Tifnout Tighanimine (CTT).....	18
VI. Présentation du service d'accueil au sein de la CTT	20
1. Service des systèmes d'information.....	20
2. Service informatique de CTT.....	20
VII. Cahier des charges du projet.....	21
1. Contexte	21
2. Objectif	21
3. Etude de l'existant.....	22
4. Description de la solution	22
5. Spécifications fonctionnelles	23
6. Spécifications techniques.....	24
7. Analyse des risques.....	24
Conclusion	25
Chapitre II :	26
Introduction.....	27
I. Raspberry Pi.....	27

1.	Les composants standards de Raspberry Pi	27
2.	Modèles et caractéristiques	28
3.	Caractéristiques du Raspberry Pi2 modèle B.....	30
4.	Le port GPIO.....	31
II.	Choix de système d'exploitation.....	32
III.	Matériels de surveillance	32
1.	Platine d'interface "Grove Pi"	32
2.	Capteur de température et d'humidité.....	33
3.	Capteur de mouvement PIR	34
4.	Caméra	36
IV.	Outils informatiques utilisés	37
1.	Base de données	37
2.	Interface de supervision par java	37
2.1.	Choix du langage de programmation.....	38
2.2.	Choix de la bibliothèque graphique	38
2.3.	Choix de l'outil de développement.....	39
2.4.	Les graphes de supervision	39
3.	Interface de supervision Application web.....	39
3.1.	Les langages utilisés	40
3.2.	Les outils de travail.....	42
	Conclusion	42
	Chapitre III :.....	43
	Introduction.....	44
I.	Configuration du système à base du Raspberry Pi.....	44
1.	Accès à distance au Raspberry Pi	44
1.1.	Choix de l'adresse IP du Raspberry Pi	44
1.2.	Connexion à distance via SSH.....	46
1.3.	Connexion à distance via VNC.....	47
2.	Langage de programmation	49
3.	Test du Grove Pi et capteurs	49
3.1.	Grove Pi	50
3.2.	Capteur de température et d'humidité DHT11	51

3.3. Capteur PIR.....	52
4. Test du camera	53
II. Le système de surveillance de la salle serveurs	55
1. Schéma descriptif du système.....	55
2. Système de collection de données et de contrôle.....	56
3. L’organigramme du programme principal.....	57
4. L’organigramme de la routine d’interruption	59
5. Création de la base de données MySQL.....	61
5.1. Installer MySQL	61
5.2. Création de base de données.....	61
5.3. Structure de la base de données	61
Conclusion	63
Chapitre IV :	64
Introduction.....	65
I. Le développement de l’application web	65
1. Installation du serveur web sur le Raspberry	65
2. Installation de PHP sur le Raspberry	66
3. Configuration de serveur.....	67
4. Plan de l’application	67
5. La connexion.....	68
6. Description de l’application.....	69
6.1. La page d’authentification	69
6.2. Page de Surveillance :.....	69
II. Développement de l’application Java	73
1. L’environnement de travail « Netbeans ».....	73
2. Importation des bibliothèques.....	74
3. Accès à la base de données	75
4. Description de l’interface.....	75
Conclusion	79
Conclusion Générale.....	80
Bibliographie.....	81
Webographie.....	81

Annexe	82
I. Installation du système d'exploitation.....	82
1. Choix de système d'exploitation.....	82
2. Les étapes de l'installation.....	83
2.1. Télécharger l'image Raspbian For Robots	83
2.2. Ecrire l'image sur la carte SD.....	83
2.3. Installer Raspbian For Robots.....	84
II. Langage de programmation python	87
1. Introduction.....	88
2. Les différents types :.....	88

INTRODUCTION GENERALE

La sécurité des informations est devenue un élément indispensable de l'entreprise quel que soit son secteur d'activité. Une indisponibilité d'un système de contrôle à des impacts très préjudiciables sur l'activité et sur la notoriété d'une entreprise.

Il existe deux enjeux majeurs pour la **Direction des Systèmes d'Informations** du Groupe **MANAGEM**, le premier est de garantir la disponibilité et les niveaux de service du système en cas de panne, et le second est de tenter de prévenir en cas de problème et, le cas échéant, garantir une remontée d'information rapide et une durée d'intervention minimale.

Ce rapport présentera le travail effectué durant notre stage de fin d'étude. Dans le premier chapitre nous présenterons l'organisme d'accueil, son historique, sa structure, ainsi une description du cahier des charges. Le deuxième chapitre est consacré à donner toutes les bases théoriques et techniques logiciel et matériel nécessaire à la réalisation de notre projet. Le troisième chapitre présente les différentes étapes de la mise en œuvre de la solution, les configurations, et les tests du système de surveillance. Enfin le dernier chapitre présentera une description des applications de supervision que nous avons développé.

Chapitre I :

Présentation de la société et Description de projet

Introduction

Le présent chapitre présente une vue globale sur l'organisme d'accueil. Ainsi l'étude du cahier des charges avec la DSI, dont l'intention de faire le point sur les grands axes de ce projet.

I. Présentation de groupe MANAGEM

Le Groupe MANAGEM est un acteur marocain dans le secteur des mines et de l'hydrométallurgie. Il opère depuis plus de 85 ans dans l'extraction, la valorisation et la commercialisation des métaux de base, des métaux précieux, du cobalt et d'autres minerais, au Maroc et en Afrique [1].

Le Groupe MANAGEM filiale de la société nationale d'investissement (SNI) opère au niveau de :

- L'exploitation et la valorisation minière à travers 5 sociétés filiales (**CMG**, SMI, AGM, CTT, SAMINE).
- Les services support aux activités minières à travers deux sociétés filiales (TECHSUB et REMINEX).
- Le trading de produits miniers du groupe à travers une filiale de commercialisation à l'international de la production du Groupe.

Son activité s'articule autour de trois branches, à savoir les métaux de base, les métaux précieux ainsi que le cobalt et spécialités. En effet l'activité minière de MANAGEM consiste à explorer, extraire, concentrer, valoriser et commercialiser différents concentrés :

- **Métaux de base**: cuivre, zinc, plomb
- **Métaux précieux** : or et argent
- **Métaux spéciaux** : cobalt et dérivés, fluorine

II. Liste signalétique du groupe MANAGEM

Raison sociale	
Nature	Société Anonyme
Président Directeur Général	M. Imad TOUMI
Siège social	Twin center, Tour A, Angle boulevards Zerktouni et Al Massira Alkhadra, BP : 5199 CASABLANCA, MAROC
Téléphone	05 22 95 65 65
Fax	05 22 95 64 64
Effectif	5600 collaborateurs
Site web	http://www.managemgroup.com/
Immatriculation	C.N.S.S. : 1284281 Patente : 35502445 IF. : 01085047 de TVA : 851001
Filiales	SMI, AGM, CMG, CTT, SAMINE, REMINEX, TECHSUB
Unités	21 unités industrielles
Chiffre d'affaire	439 millions USD en 2015

Figure 1 : Fiche technique de MANAGEM

III. Historique

- 1928 : Découverte du gisement de Cobalt de Bou-Azzer.
- 1930 : Création de la société CTT et démarrage de la mine de Bou-Azzer spécialisée dans la production de concentré de cobalt.
- 1932 : Démarrage des travaux de prospection de manganèse à Aoulouz et de fer à Tanguerfa.
- 1942 : Lancement d'exploitation par CTT du gisement de manganèse de Tiouine.
- 1949 : Création de la société des mines de Bouskour (SMBS) pour l'exploitation d'un gisement de cuivre situé dans la région d'Ouarzazate.
- 1961 : Création de la société SAMINE.
- 1974 : Démarrage de la mine d'El Hammam spécialisée dans la production de fluorine.
- 1981 : Démarrage de l'exploitation de la mine de Bleida par la société SOMIFER.
- 1983 : Création de REMINEX, filiale spécialisée dans la R&D, l'ingénierie et l'exploration.
- 1992 : Démarrage de la mine polymétallique de Hajar.
- 1996 : Création du holding Managem regroupant les participations minières du Groupe ONA.
- 2000 : Introduction en bourse de Managem.
- 2001 : Démarrage de la mine d'or d'Akka, la mine d'or de Kiniero en Guinée, et l'unité de production d'oxyde de zinc sur le site de Guemassa.
- 2004 : Démarrage de la mine polymétallique de Draa Sfar, et l'exploitation de la mine d'or Samira Hill au Niger
- 2006 : Création de la société LAMIKAL en République démocratique du Congo.

- 2008 : Création de MANAGEM INTERNATIONAL, holding détenant l'ensemble des participations du Groupe à l'étranger.
- 2012 : Lancement de la production d'or à Bakoudou au Gabon.
- 2013 : Démarrage de l'extension de SMI.
- 2013 : Démarrage de l'usine d'acide sulfurique à Guemassa.
- 2014 : Réalisation des études de faisabilité de deux projets cuprifères : Bouskour au Maroc et Pumpi en RDC

IV. Organigramme du groupe MANAGEM

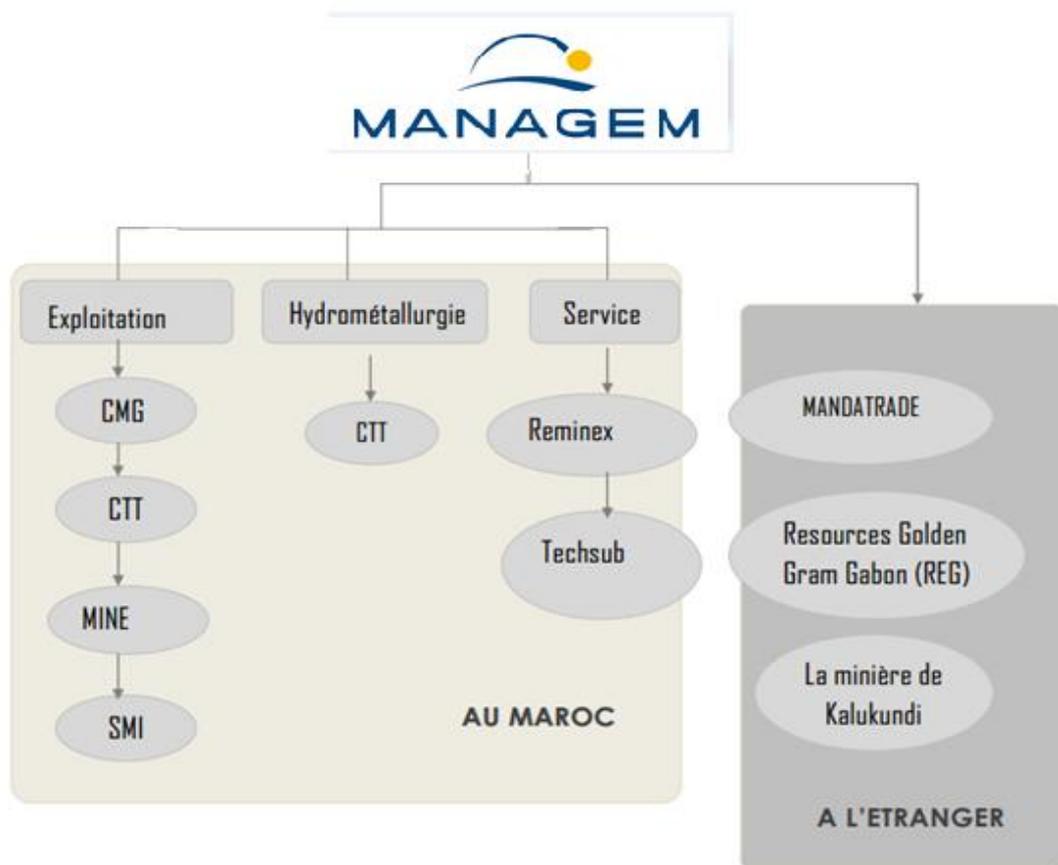


Figure 2: Structure du groupe MANAGEM

V. Présentation de La Compagnie de Tifnout Tighanimine (CTT)

CTT est la filiale de MANAGEM qui gère son complexe Hydrométallurgie, qui est devenu son deuxième grand métier. Située dans la Région de Guemassa, à 35 km au sud-ouest de Marrakech allant à Amizmiz, elle est le fruit des efforts du centre de recherche REMINEX (filiale recherche de MANAGEM) pour développer des processus de valorisation de cobalt, du Cuivre...les activités démarrèrent avec l'installation d'une première unité l'hydro 1 (H1) en 1996 de production de cobalt à partir des rejets de la mine de Bouazzer Au sud de Ouarzazate. Suivra en 1999 l'implantation des unités de grillage (grillé Concentre de cobalt) et l'hydro 2, deuxième unité de production de cobalt [1].

En 2000 l'unité du cuivre sommital fut implantée pour produire du sulfate de cuivre hydraté à partir d'extrait Hajjar (région de Guemassa). En 2002 l'unité de production d'oxyde de zinc (la calamine) est faite leur démarrage.

Aujourd'hui, grâce au savoir-faire de REMINEX, les déchets de l'exploitation du site C.T.T à Bouazzer est valorisée.

CTT Guemassa est la société en charge de la gestion de l'ensemble des activités hydro métallurgiques du groupe. Elle gère plusieurs structures opérationnelles, notamment :

Unité Oxyde Cobalt :

Première unité d'hydrométallurgie de MANAGEM, elle produit des cathodes de cobalt de haute pureté, à partir des rejets de la mine de Bouazzer. Ces cathodes trouvent des applications dans plusieurs domaines .superalliages, batteries, pigment, chimie... Les étapes du procédé sont :

- Lixiviation acide des haldes contenant 0,4% Co
- Elimination de l'arsenic du minerai
- Précipitation des hydroxydes de cobalt

- Concentration du Cobalt à 10/12%
- Séparation Cobalt / Nickel par extraction liquide
- Electrolyse
- Production de cobalt cathodes (99,8%).

Hydro II :

Ce complexe permet de traiter le concentré de cobalt de la mine de BouAzzer pour le transformer en cathode de haute pureté. Les étapes du procédé sont :

- Grillage du concentré de cobalt pour éliminer l'arsenic
- Production de trioxyde d'arsenic pure
- Lixiviation acide
- Extraction liquide
- Electrolyse
- Production de cathodes de cobalt (99,8% Co)

L'unité cuivre sommitale :

Produit depuis 2000, des cristaux de sulfate de cuivre à partir d'un minerai riche en cuivre extrait de la mine de Hajjar, située sur le site de Guemassa. Elle produit de l'oxyde de zinc suivant le procédé suivant :

- Préparation mécanique du minerai
- Lixiviation à l'acide sulfurique pour mettre le zinc en solution
- Deux étapes de purification
- Précipitation du zinc en deux zincs.

✚ L'unité de grillage :

Qui a démarré en 1999 et produit aujourd'hui le trioxyde d'arsenic qui entre dans les industries de traitement du bois, verrerie, et le concentré de cobalt grillé qui alimente l'usine Hydro.

✚ L'unité calamine :

L'unité calamine, démarrée en 2002, à travers laquelle MANAGEM produit de l'oxyde de zinc de très haute pureté à partir de concentrés riches en zinc acheminés depuis des carrières situées dans la région d'Errachidia.

VI. Présentation du service d'accueil au sein de la CTT

1. Service des systèmes d'information

Nous avons eu l'occasion d'effectuer un stage au sein de la Compagnie Tifnout Tighanimine et exactement au **Service des systèmes d'information** avec M. Abdelaziz EL AMRANI, M. AMINE Mohamed Moslih, Melle. ABARRO Kawtar, Mr. AADIL Alaa au but de mettre mes connaissances théoriques en pratique, d'avoir un esprit de travail collectif et de savoir comment je peux les mettre en œuvre au milieu du travail. M. El AMRANI et M. AMINE sont les gérants du service de l'information et les responsables de la maintenance informatique et téléphonique, qui font la réparation de tout le matériel informatique et les réseaux du site minier de Guemassa. Mr.Alaa est le responsable de développement au sein de service.

2. Service informatique de CTT

Le réseau de la Compagnie Tifnout Tighanimine fonctionne avec plusieurs matières notamment :

- Les commutateurs Catalyst Cisco (4503, 2960, 2948,3948....)
- Des serveurs comme :
 - Un serveur Reflexe 132

- Un serveur Report 132
 - Un serveur AD21
 - Un serveur SEP21
 - Un serveur de messagerie.
- Des fibres optiques
 - Des autocommutateurs
 - Des panneaux de brassages
 - Des armoires de brassages
 - Des onduleurs

VII. Cahier des charges du projet

1. Contexte

Dans le cadre du renforcement de la sécurité du site de BACKUP de la Direction des Systèmes d'Information du Groupe MANAGEM, localisé au niveau du site de GUEMASSA, la mise en place d'une application de surveillance à distance de la salle de SERVEUR est indispensable, pour assurer les différents contrôles.

2. Objectif

Après discussion avec l'encadrant, sur les besoins de la société, il nous a été demandé d'accomplir les objectifs suivants :

- ✚ Collection et enregistrement des données captées (température, humidité et état du capteur PIR) dans un fichier ou une base de données.
- ✚ La visualisation sous la forme graphique des données captées (température, humidité et état du capteur PIR) et leur évolution dans le temps.
- ✚ L'envoi des alertes par email à l'administrateur de la salle en cas d'anomalie par

exemple : dépassement de seuil autorisé ou détection d'intrusion.

3. Etude de l'existant

La salle des serveurs du site Backup de GUEMASSA est la salle de gestion du parc informatique de tout le site, le schéma suivant décrit l'état actuel de cette salle :

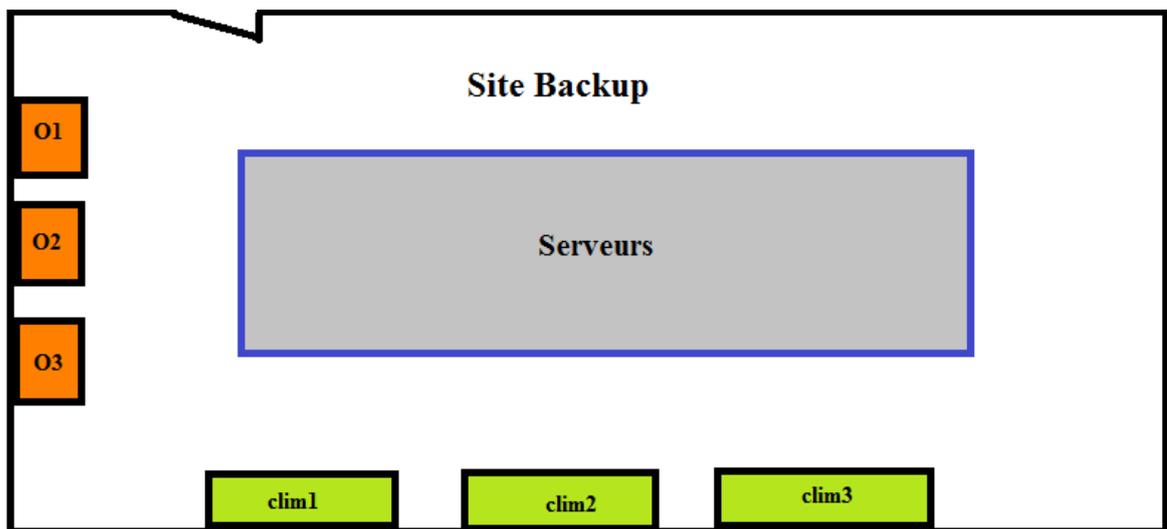


Figure 3: Schéma descriptif du site Backup

La salle contient :

- ✚ Trois onduleurs qui alimentent la salle en cas de coupure d'électricité.
- ✚ Trois climatiseurs qui servent à fixer la température de la salle à un niveau convenable.

D'après cette description, l'idée de mettre en œuvre un système de surveillance de la salle apparaît nécessaire. Ce système va servir au contrôle d'intrusion et la supervision de la température et d'humidité.

4. Description de la solution

Pour réaliser les objectifs déjà cités précédemment, nous avons fait une étude détaillée du projet. Suite à cette étude nous avons décidé de travailler avec la carte électronique embarquée dotée d'un processeur de famille ARM, c'est le Raspberry Pi 2 model B.

Pour répondre aux autres objectifs nous avons opté pour les solutions suivantes :

- ✚ Pour la collection des données nous avons choisi d'utiliser le module d'interfaçage Grove Pi qui permet d'adapter les capteurs avec la carte Raspberry pi.
- ✚ Le capteur qui va nous permettre de mesurer simultanément la température et l'humidité est le capteur Grove DHT 11.
- ✚ Le détecteur de présence que nous avons choisi est le capteur PIR de référence SHC-SR501.
- ✚ Développement de la couche basse logicielle en utilisant le langage python sous LINUX pour la gestion du système (collection et enregistrement des données, détection de présence, prise des photos).
- ✚ L'enregistrement des données dans une base de données MySQL.
- ✚ Développement d'une interface graphique avec Java pour la visualisation des données captées.
- ✚ Développement d'une application Web.
- ✚ Utilisation du protocole SNMP/SMTP pour l'envoi des alertes par Email.

5. Spécifications fonctionnelles

Le système de supervision que nous avons proposé doit répondre aux fonctions suivantes :

- ✚ La carte Raspberry Pi doit exécuter un programme python qui va gérer la collection des données (température et humidité) pour des temps prédéfinis et les stocker dans une base de données MySQL. Le même programme doit être capable de détecter la présence (ou l'intrusion à la salle de serveur) d'une personne et à ce moment le programme doit prendre deux ou trois photos (ou plus selon notre choix), les enregistrer, puis envoyer un alerte email.
- ✚ Utilisation d'une application Java pour la visualisation des données déjà stockées

dans la base de données sous forme des graphes.

- ✚ L'accès à distance à une application web pour la visualisation des données, qui doit être sécurisé.

6. Spécifications techniques

Les spécifications techniques de notre système sont les suivantes :

- ✚ Un seul programme qui gère la collection et le stockage de données dans une base de données, l'envoi d'un alerte email en cas d'anomalie.
- ✚ Utilisation d'une base de données MySQL version 5.5 (**server version 5.5.49 for Debian**) sur le Raspberry sécurisée par un mot de passe.
- ✚ Utilisation d'un **serveur web apache 2**.
- ✚ Utilisation d'un **serveur PHP 5**.
- ✚ L'autorisation de l'accès à la base de données à distance non seulement au réseau local.
- ✚ Accès à le Raspberry Pi à distance avec le protocole SSH et VNC.
- ✚ L'application Java baser sur JDK de Oracle version « **jdk-8u91-windows-x64** ».
- ✚ Installer le système d'exploitation « **Raspbian for Robots** » sur la carte Raspberry Pi.

7. Analyse des risques

Comme tout projet, toujours il peut y avoir des risques lors de sa réalisation, pour cela cette partie a pour vocation de déceler les risques pour les éviter, parmi ces risques :

- ✚ Avoir des problèmes lors de l'installation de système d'exploitation.
- ✚ Un dysfonctionnement des packages et bibliothèques.
- ✚ Interruption lors d'accès aux serveurs des packages.
- ✚ Avoir des problèmes de non compatibilité logiciel (version Linux...).
- ✚ Confondre entre la syntaxe de différentes versions de même langage.
- ✚ Utilisation d'un outil dans un rôle non souhaitable.

- ✚ Ne pas mettre les paramètres de sécurisation.
- ✚ Ne pas prendre en compte la différence entre références.

Conclusion

L'objectif de ce chapitre était de présenter aux lecteurs l'environnement de stage, d'exposer une vue générale du contexte du travail, et aussi une description brève de cahier des charges.

Chapitre II :

Fondements théoriques

Introduction

Après la présentation de la société et la description du cahier des charges, ce deuxième chapitre vient pour donner aux lecteurs toutes les bases théoriques et techniques nécessaires à la réalisation de ce projet.

I. Raspberry Pi

Raspberry est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (C, C++, python, scratch...) et d'être utilisé dans les systèmes embarqués. Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo [2].

1. Les composants standards de Raspberry Pi

La figure suivante représente un Raspberry standard :

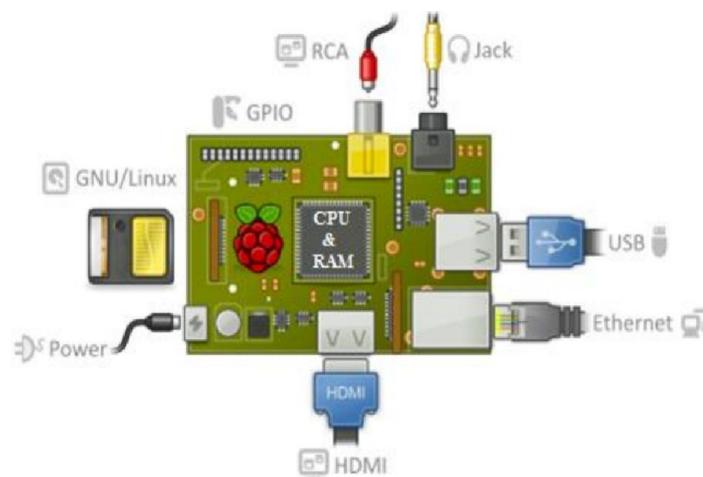


Figure 4 : Les composants standards d'un Raspberry

- ✚ **Processeur ARM** : Les architectures ARM sont des architectures de processeurs, à faible consommation, introduites à partir de 1983 par « Acorn Computers » et développées depuis 1990 par « ARM Ltd ».

- ✚ **Mémoire vive RAM** : C'est la mémoire dans laquelle le Raspberry place les données lors de son traitement.

Une connectique variée :

- ✚ **HDMI** : « High Definition Multimedia Interface » permet de relier le Raspberry PI à un dispositif compatible : écran LCD ou un vidéoprojecteur...
- ✚ **Port USB 2.0** : Le port « Universal Serial Bus » est un port série qui sert à connecter le Raspberry aux autres périphériques.
- ✚ **Port Ethernet** : C'est un port qui correspond au protocole international ETHERNET de réseau local à commutation de paquets.
- ✚ **Prise RCA** : « Radio Corporation of America » est un connecteur électrique utilisé dans le domaine audio/vidéo.
- ✚ **un slot les cartes SD** : Le Raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce slot permet de connecter la mémoire externe.
- ✚ **une prise jack** : C'est une connectique audio-vidéo.
- ✚ **GPIO** : « General Purpose Input/Output » sont des ports d'Entrée/Sortie.

2. Modèles et caractéristiques

Dans ce paragraphe on va faire une brève présentation des modèles et les caractéristiques du Raspberry Pi, les modèles les plus couramment diffusés et ferons l'impasse sur les tout premiers modèles, ainsi que sur la gamme professionnelle.



Figure 5: Différents modèles du Raspberry Pi

Chacun de ces modèles ayant des caractéristiques bien spécifiques pour cela nous avons décidé de présenter les caractéristiques de chaque modèle dans un schéma qui fait une comparaison entre ces différents modèles :

Modèle	A	A+	B	B+	B2
CPU	Monocœur ARM 700 MHz				Quadricœur ARM 900 MHz
GPU	Décodeur vidéo Broadcom VideoCore IV				
RAM	256 MO		512 MO		1 GO
USB	1 * USB2.0		2 * USB2.0	4 * USB2.0	
Audio/vidéo	Jack 3.5, composite et HDMI	HDMI et jack audio/vidéo	Jack 3.5, composite et HDMI	HDMI et jack audio/vidéo	
Ethernet	0	10/100	0	10/100	
Entrées/sorties	GPIO 26 pts	GPIO 40 pts	GPIO 26 pts	GPIO 40 pts	
OS	Officiel : Raspbian Tiers : Fedora, XBMC/Kodi, OSMC				
Stockage	SD	Micro SD	SD	Micro SD	
Dimensions	86*54*17	65*54*17	86*54*17		
Poids	45g	23g	45g		
Consommation	1.5W	1W	3.5W	3W	

Figure 6: Les caractéristiques des modèles

3. Caractéristiques du Raspberry Pi2 modèle B

« Nous vous recommandons le Raspberry Pi 2 Modèle B pour une utilisation dans les écoles: il offre une plus grande souplesse pour les apprenants que les plus maigres (Pi 1) Modèle A +, ce qui est plus utile pour les projets et les projets intégrés qui nécessitent une très faible puissance » **fondation du Raspberry.**

Le Raspberry Pi 2 Modèle B (figure 7) est la deuxième génération Raspberry Pi et la carte sur laquelle **on va développer notre système.**

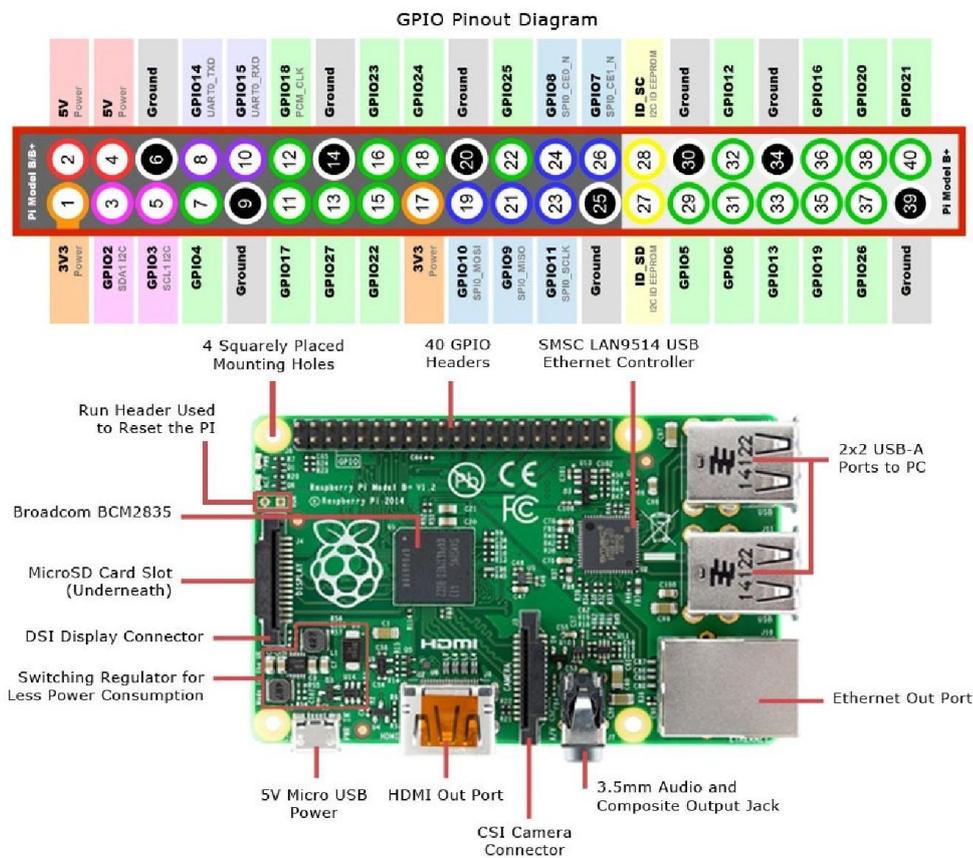


Figure 7: Les caractéristiques du Raspberry Pi 2 Modèle B

Le Raspberry Pi 2 Modèle B dispose d'un processeur ARMv7, donc il peut fonctionner toute la gamme des distributions ARM GNU / Linux, y compris Snappy Ubuntu Core, ainsi que **Microsoft Windows 10** voir plus d'autres systèmes d'exploitation.

4. Le port GPIO

Le Raspberry pi 2 possède différents types de GPIO (General Purpose Input/Output) chacun ayant une fonction bien définie comme on peut le configurer soit en entrée soit en sortie, noter bien que ces broches d'entrées/sorties (I/O) travaillent avec des tensions comprises entre 0 et 3,3 V et NON PAS 0 et 5V. Le schéma suivant montre la description des broches GPIO :

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I ² C)		DC Power 5v	04
05	GPIO03 (SCL1, I ² C)		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 (I ² C ID EEPROM)		(I ² C 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

Figure 8: Description des broches GPIO

Pour la configuration des broches, le Raspberry Pi 2 autorise deux numérotations: celle de la sérigraphie du connecteur de la carte (**GPIO.BOARD** les numéros encadrés en jaune dans la figure précédente), ou la numérotation électronique de la puce (**GPIO.BCM** les numéros encadrés en rouge dans la figure précédente). Le choix de l'une de ces numérotations est laissé à l'utilisateur (ou programmeur) de la carte.

II. Choix de système d'exploitation

La distribution de système d'exploitation choisie c'est « Raspbian For Robots » fournis par la société « Dexter Industries », qui est une distribution Raspbian modifiée et qui contient des logiciels Grove Pi entièrement installés pour qu'on puisse utiliser le Raspberry Pi 2 avec la carte des capteurs Grove Pi (comme on va voir dans la suite de ce rapport) sans qu'il y ait des problèmes de communication et de traitement de données, et pour plus d'informations voir annexes.

III. Matériels de surveillance

1. Platine d'interface "Grove Pi"

Grove Pi est un module d'adaptation des capteurs de type Grove avec la carte Raspberry Pi. Grâce à Grove Pi, nous pouvons connecter simplement et sans soudure de très nombreux capteurs Grove à notre carte Raspberry Pi et les utiliser avec une manière simple et bien organisée.

Le module Grove Pi (figure 9) est basé sur un microcontrôleur ATmega328P Arduino permettant d'interfacer les capteurs Grove avec la carte Raspberry Pi. Ce microcontrôleur est livré avec un firmware standard et peut être reprogrammé directement via la carte Raspberry.

Une librairie est téléchargeable et compatible avec la plupart des capteurs Grove disponibles. Le module peut se programmer en langage Python ou en langage C sur la carte Raspberry Pi. Voici les caractéristiques de ce module :

- ✚ Alimentation: via la carte Raspberry (non incluse)
- ✚ Connecteurs Grove:
 - 3 connecteurs analogiques (A0-A1-A2)
 - 7 connecteurs digitaux (D2-D3-D4-D5-D6-D7-D8)
 - 3 connecteurs I2C
 - 2 connecteurs sériels (RPISER et SERIAL)
- ✚ Dimensions: 88 x 58 x 24 mm

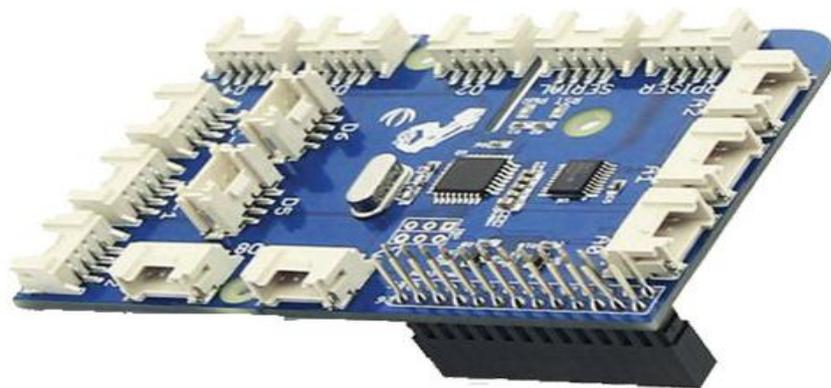


Figure 9: Le module Grove Pi

2. Capteur de température et d'humidité

Le capteur de température et d'humidité (figure 10) est un circuit comprend le capteur DHT11 qui fournit une information numérique proportionnelle à la température et l'humidité mesurées. La technologie utilisée par ce capteur garantit une grande fiabilité, une excellente stabilité à long terme et un temps de réponse très rapide.

Les caractéristiques de ce capteur sont les suivantes :

	Alimentation	Plage de mesure	Précision	Sensibilité
Humidité	3.3V ~ 5V	20% - 90%	± 5%	± 1%
Température	3.3V ~ 5V	0 ~ 50 ° C	± 2 ° C	1 ° C

Tableau 1: Les caractéristiques du capteur DHT11



Figure 10: Capteur température et d'humidité

3. Capteur de mouvement PIR

Le capteur PIR (Passive Infra Red) est un capteur qui détecte une source d'infrarouge placée dans son champ de "vision" et détecte ses mouvements. Il permet de détecter une présence humaine ou animale dans un périmètre et déclencher une réponse. Le capteur utilisé dans ce projet est SHC-SR501. Le brochage de ce dernier est présenté dans la figure suivante :

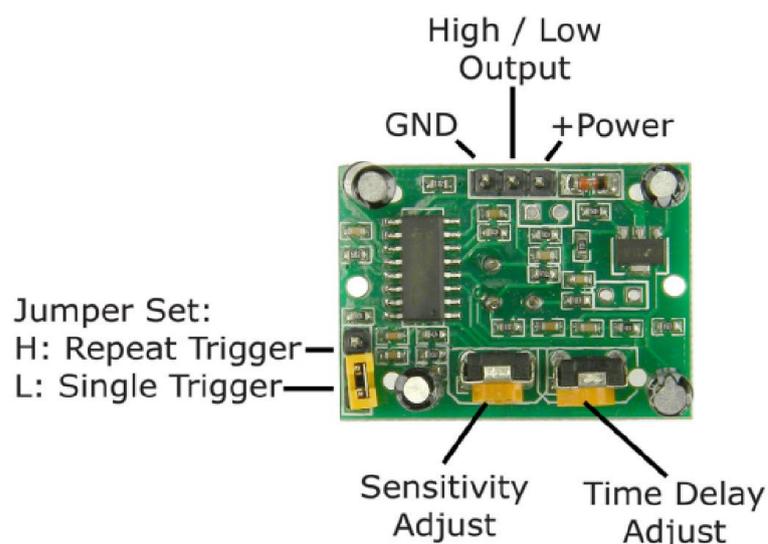


Figure 11: Brochage du capteur SHC-SR501

Le capteur possède deux potentiomètres :

- ✚ Time Delay Adjust = Ajustement du délai pendant lequel la sortie(Output) reste verrouillée sur HIGH après une détection de mouvement.
- ✚ Sensitivity = Ajustement de la sensibilité du capteur entre 3 et 7m.

Le **jumper** permet de choisir le mode : avec ou sans re-déclenchement (voir le schéma suivant) :

- ✚ Sans re-déclenchement (Single Trigger), la sortie passe à l'état HIGH lors du premier contact, y reste durant toute la durée réglée sur le potentiomètre, pendant ce temps le capteur ne "regarde plus". Donc la durée de sortie HIGH est alors fixe.
- ✚ Avec re-déclenchement (Repeat Trigger), la sortie passe à l'état HIGH lors du premier contact, y reste durant toute la durée réglée sur le potentiomètre, mais chaque nouvelle détection remet le chronométrage de durée de sortie à zéro.

Ceci permet une détection en continu, le capteur répond tant que la source d'infrarouge est mobile.

Par défaut, le Jumper Set sur H = Repeat Trigger. La durée de sortie HIGH peut être prolongée tant qu'il y a détection.

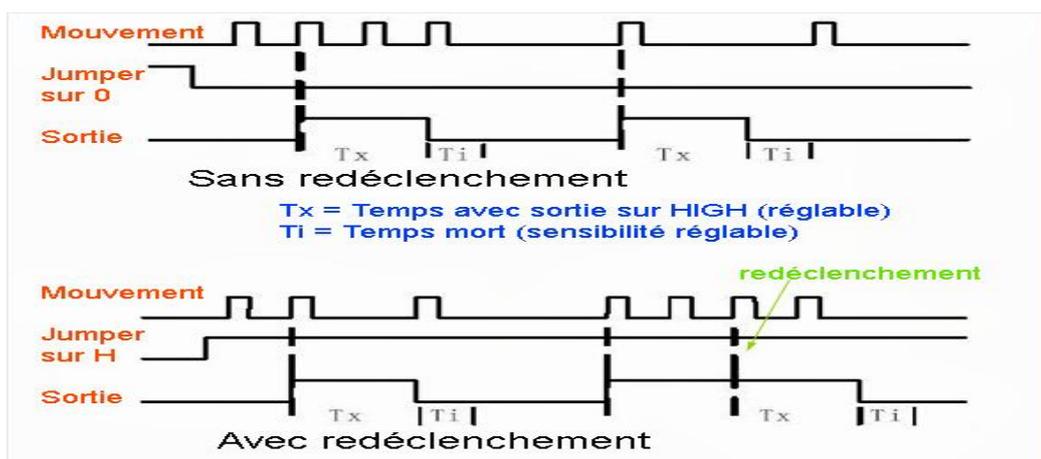


Figure 12: schéma explicatif du fonctionnement du Jumper Set

Citons maintenant les caractéristiques de ce capteur :

- ✚ Alimentation de 4.5 V à 20 V DC
- ✚ Sortie high 3.3V, low 0v
- ✚ Délai de sortie ajustable par potentiomètre de 3Sec à 5Min
- ✚ Déclenchement L sans répétition, H avec répétition (mode par défaut)
- ✚ Portée de 7 m et angle de 120°
- ✚ Sensibilité réglable par potentiomètre de 3m à 7m
- ✚ Dimension du circuit 32mm*24mm

4. Caméra

Dans notre projet on va utiliser une caméra pour prendre des photos en cas d'une intrusion à la salle du serveur, le meilleur choix est d'utiliser le module de caméra dédiée au Raspberry Pi ; La caméra achetée est « Raspberry Pi Camera Rev 1.3 » dispose d'un capteur de 5 méga pixels Sony IMX219.

Le module de caméra peut être utilisée pour prendre la vidéo haute définition (HD), ainsi que de prendre des photos. Il prend en charge 1080p30, 720p60 et modes vidéo VGA90. L'appareil fonctionne avec tous les modèles de Raspberry Pi 1, 2 et 3, et ce qui est très important c'est qu'il y a de nombreuses bibliothèques déjà construites pour la commande de cette caméra. Parmi elles, on trouve le **Picamera** bibliothèque Python qu'on va utiliser par la suite.

Maintenant on va résumer les caractéristiques de la camera :

- ✚ La caméra se branche sur le connecteur CSI existant sur la carte Raspberry Pi par câble plat à l'interface 15-pin.
- ✚ Capteur 5 Mégapixels.
- ✚ Résolutions photo 2592 x 1944
- ✚ Résolutions vidéo 1080p à 30fps (30 images/s), 720p à 60fps et 640x480p à 60 fps ou 90fps
- ✚ Dimension du capteur 3280 x 2464 pixels

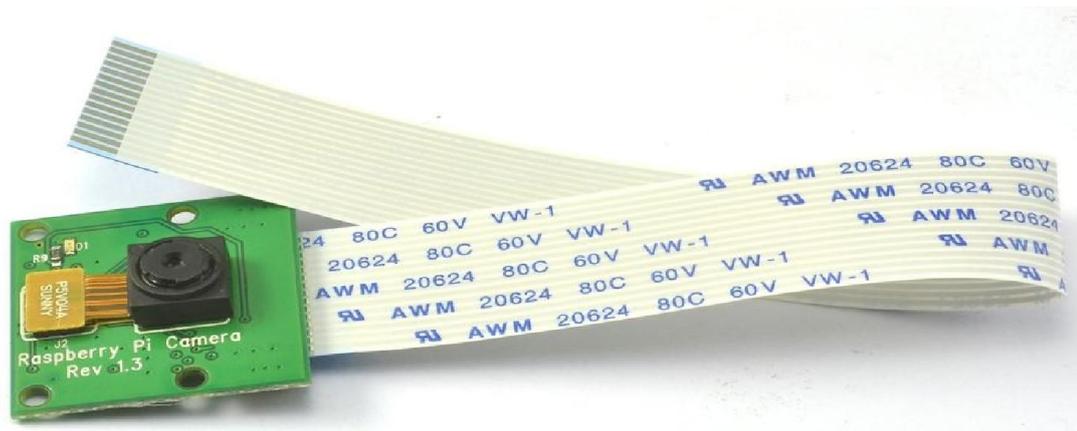


Figure 13: Raspberry Pi Camera Rev 1.3

IV. Outils informatiques utilisés

La collecte et la visualisation de données est devenue une préoccupation de plus en plus majeure pour les entreprises, tant sur les individus que sur l'environnement de travail. Pour notre travail le plus important est de collecter les données sur un environnement et faire le stockage de ces derniers sur un espace sécurisée et finalement les visualiser.

1. Base de données

La base de données est un service qui propose de stocker des données de façon structurée, cohérente, organisée et avec le moins de redondance possible afin de pouvoir les manipuler et les mettre à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour. Ce service est placé sur un serveur qui va être interrogé par les diverses applications « site web, logiciel, etc. ».

Le SGBD (Système de Gestion de Bases de Données) qui couvrira notre base de données est MySQL. Son choix se justifie par sa simplicité d'utilisation. Aussi le côté standard de MySQL, nos connaissances de celui-ci et sa relative rapidité ont rapidement orienté notre choix vers lui.

2. Interface de supervision par java

2.1. Choix du langage de programmation

Avant tout il nous fallut choisir un langage de programmation qui allait être utilisé pour développer notre interface de supervision. Nous focalisons nos choix sur des langages orientés objet et entre deux en particulier : C++ et Java. Nous toutefois préférons Java à C++, notamment à cause de la portabilité de Java, permettant d'utiliser une application sur de multiples systèmes d'exploitation sans avoir à la recompiler. Nous avons également choisi ce langage car il possède des bibliothèques graphiques en standard et qu'il existe des outils facilitant le développement d'interfaces.

2.2. Choix de la bibliothèque graphique

Après notre choix de langage de programmation pour notre application, on a besoin de choisir la meilleure bibliothèque pour construire le corps de l'interface, pour cela nous avons comparé les deux principales bibliothèques graphiques de Java pour déterminer laquelle utiliser. Car il est préférable de n'en utiliser qu'une seule à la fois, tant les risques d'erreurs peuvent survenir entre les deux API graphiques - leurs syntaxes étant assez proches. Ces deux bibliothèques sont Swing et JFace. Chacune de ces bibliothèques se reposent sur une autre bibliothèque, AWT pour Swing et SWT pour JFace. La grande différence entre Swing et JFace est que Swing crée ses propres éléments en Java, ce qui le rend donc indépendant de l'OS (Operating System), par contre SWT demande au système d'exploitation de lui créer des éléments externes à Java et qu'il gèrera par la suite.

De cette différence découle différents avantages et inconvénients (Tableau 2) :

	Performances	Portabilité	Apparence
Swing	Assez lourd Gestion automatique des ressources	Indépendant de l'OS	Indépendant de l'OS
JFace	Rapide et fluide Gestion manuelle des ressources	Installation de bibliothèques en fonction de l'OS	Apparence similaire à celle de l'OS

Tableau 2 : Tableau comparatif entre Swing et JFace

Après cette étude comparative, nous sommes plutôt tournés vers Swing que JFace, car on l'a trouvé plus pratique que JFace pour développer l'interface. Le nombre de données à afficher étant important, Swing sera également plus performant que JFace. De

plus, Swing, plus récent que JFace, est également plus optimisé, et donc plus rapide.

2.3.Choix de l'outil de développement

Avant de commencer le développement de l'interface, on doit rechercher un environnement de travail permettant de travailler avec la bibliothèque qui nous avons choisi précédemment. Les environnements de développement par java est nombreux mais les plus populaire sont Eclipse, NetBeans, Spring Framework, qui sont open source (code source libre).

Eclipse et NetBeans sont deux environnements de développement qui ont été codés en Java pour produire des programmes en Java, mais la différence la plus importante entre ces deux Framework est que NetBeans utilise **Swing** pour la partie graphique tandis que celui d'Eclipse utilise **JFace**, et puisque nous avons décidé de travailler dans la partie précédant de travail avec la bibliothèque **Swing**, donc on va travailler avec NetBeans. De plus, cela nous permettra de découvrir un nouvel environnement et de ne pas nous limiter à Eclipse.

2.4.Les graphes de supervision

Après le choix d'IDE et la bibliothèque graphique, nous avons besoin d'un outil de dessiner les graphes de nos données collectées depuis les capteurs, le problème est que le JDK sur laquelle se base NetBeans IDE ne contient aucune bibliothèque et classe qui va nous permettre de dessiner les graphes. Pour rendre compte à ce problème, nous avons trouvé qu'on besoin d'un API plus précisément **JFreeChart**.

JFreeChart est une API Java permettant de créer des graphiques et des diagrammes de très bonne qualité. Cette API est open source et sous licence LGPL, mais en revanche ces documentations sont payantes.

3. Interface de supervision Application web

L'utilisation de l'application Java pose deux problèmes : le premier chaque superviseur doit avoir l'application et le deuxième le superviseur accède à la supervision seulement s'il est sur le réseau de Managem. Pour cela on a crée l'application web qui permet aux administrateurs la supervision de n'importe quel endroit dans le monde.

3.1. Les langages utilisés

Pour le développement, de l'application on a besoin des langages suivants:

- HTML
- PHP
- CSS
- JavaScript et jQuery
- Ajax

On va maintenant vous dire quelques mots sur les langages afin de vous expliquer dans quelle situation on utilise tel ou tel langage.

HTML (Hyper Text Markup Language)

HTML est un langage de description et non pas un langage de programmation, ce langage est basé sur le principe de balisage de texte qui permet, d'inclure des informations variées (textes, images, sons, animations etc.), la création de documents hypertextes qui permettant d'établir des relations cohérentes entre ces informations affichables par un navigateur.

CSS3 (Cascading Style Sheets)

Les feuilles de style en cascade (CSS) sont un langage qui permet de contrôler facilement la présentation visuelle des pages web, ajouter facilement du style (polices, couleurs, espacement, etc.) à ces pages.

Le but de CSS est séparer la structure d'un document HTML et sa présentation, en effet, avec HTML on peut définir à la fois la structure et la présentation de la page, mais cela pose quelques problèmes. Avec le couple HTML/CSS, on peut créer des pages web où la structure du document se trouve dans le fichier HTML tandis que la présentation se situe dans un fichier CSS externe, pour **éviter les redondances** de classes identiques, tout en clarifiant le code. De plus, CSS ajoute des fonctionnalités nouvelles par rapport à HTML au point de vue du style.

Langage PHP (Hypertext Preprocessor)

Le langage PHP est principalement un langage de programmation web côté

serveur, utilisé pour produire des pages Web dynamiques via un serveur http, ce qui veut dire que c'est le serveur qui va interpréter le code PHP et générer du code (HTML, CSS, et parfois de JavaScript) qui pourra être interprété par un navigateur. PHP peut être installé sur les principaux serveurs web actuels, les plus répandus étant IIS et Apache.

Pour profiter de ce couplage qui nous permet de récupérer des informations issues de notre base de données, ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées.

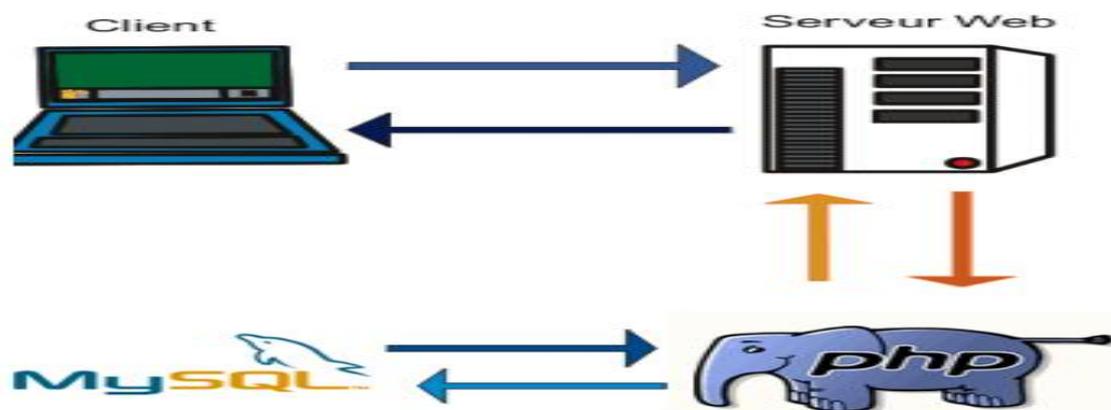


Figure 14 : Architecture de connexion client-serveur avec PHP

✚ JavaScript

JavaScript est un langage de programmation objet et événementiel utilisé pour la gestion des interactions dans les navigateurs web. JavaScript est lié au langage de balisage HTML, on peut insérer du code JavaScript dans des pages HTML grâce à la balise <script> ou par l'appel des propriétés et des méthodes, qui existent dans un fichier script (.js) et leur associer des actions en fonction d'événements déclenchés par le visiteur.

Généralement, nous avons décidé d'utiliser du JavaScript pour contrôler les données saisies dans mes formulaires HTML, par exemple pour vérifier si le mot de passe saisi par l'utilisateur est bien correct.

✚ JQuery

JQuery est une bibliothèque JavaScript qui facilite le développement d'applications web et s'appuie sur l'interaction entre JavaScript et HTML, afin de simplifier des commandes communes de JavaScript et d'ajouter des fonctionnalités

supplémentaires aux sites Web. Au cours des dernières années JQuery est devenue la bibliothèque JavaScript la plus populaire utilisée dans le **développement web**.

Ajax (Asynchronous JavaScript and XML)

Ajax « XML et JavaScript asynchrones », est une architecture qui permet de construire des sites web dynamiques interactifs à base de différentes technologies ajoutées aux navigateurs web.

L'idée de d'AJAX est de faire communiquer une page Web avec un serveur Web en évitant les téléchargements de pages complètes et en réduisant le trafic entre le client et le serveur. Un moteur Ajax est en fait un objet ou une fonction JavaScript qui est appelée chaque fois qu'une information est sollicitée auprès du serveur, car ce modèle traditionnel fournit un lien vers une autre ressource (une autre page Web par exemple), chaque lien fait appel au moteur Ajax, lequel planifie et exécute la requête. Cette dernière est établie de manière asynchrone, ce qui signifie que l'exécution du code n'attend pas de réponse avant de se poursuivre.

3.2. Les outils de travail

XAMPP SERVEUR

XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web confidentiel. Il s'agit d'une distribution de logiciels libres (Apache, MySQL, Perl PHP), ainsi que phpMyAdmin pour l'administration web des bases MySQL qui offrent une bonne souplesse d'utilisation. Nous avons utilisé XAMPP pour les tests sur Windows avant d'implémenter notre travail sur le Raspberry Pi.

RGraph

RGraph est une bibliothèque de graphiques HTML5, gratuite, créée en utilisant des scripts JavaScript. Elle supporte plus de vingt différents types graphiques.

Conclusion

Dans ce chapitre nous avons étudié la partie matérielle (Raspberry, Grove PI, capteurs...) et les outils logiciels utilisés dans notre système proposé.

Chapitre III :

Tests et Configurations du système de surveillance

Introduction

Ce chapitre présente l'architecture générale du système de surveillance de la salle serveurs, configurations et les tests du système.

I. Configuration du système à base du Raspberry Pi

1. Accès à distance au Raspberry Pi

On peut accéder à la carte Raspberry Pi via le protocole SSH et/ou VNC, si on connaît l'adresse IP de la carte.

1.1.Choix de l'adresse IP du Raspberry Pi

Pour trouver l'adresse IP de notre Raspberry Pi, nous avons utilisé **Advanced IP Scanner** qui est un scanner de réseau local utilisé sous Windows. La figure suivante montre l'interface de cet outil et l'information fournie par ce dernier concernant le Raspberry Pi :

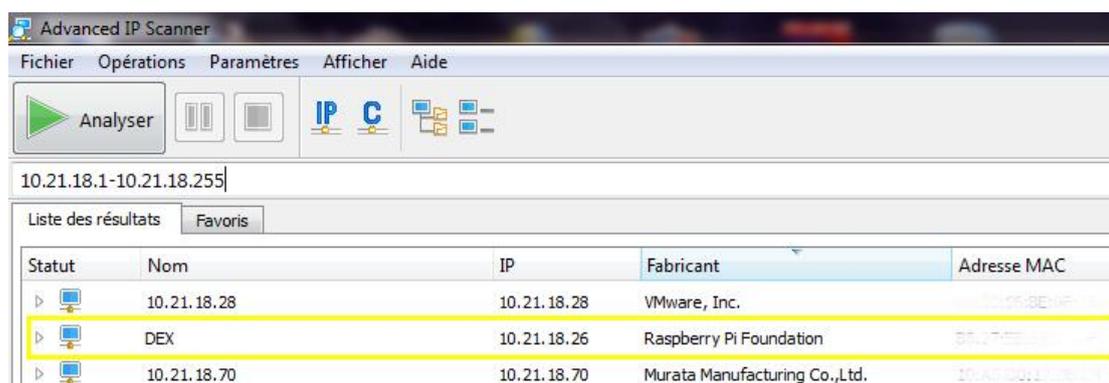


Figure 15: Adresse IP de la carte Raspberry Pi

Après que nous avons trouvé l'adresse IP, il faut la fixer afin de nous permettre l'accès à distance au Raspberry Pi, pour cela on va suivre les étapes suivantes :

- ✚ Configuration réseau sous Raspbian For Robots qui se passe dans le fichier : `/etc/network/interfaces`. Par défaut, il contient les valeurs suivantes :

```
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
```

La signification de chaque ligne de ce fichier est la suivante :

- ❖ **auto lo** : va démarrer l'interface automatiquement lors de la séquence de boot.
 - ❖ **iface lo inet loopback** : définition de l'interface loopback (interface virtuelle indispensable qui permet de supprimer les paquets erronés)
 - ❖ **iface eth0 inet dhcp** : l'interface eth0 (le port RJ45 du Raspberry) sera configurée en envoyant une requête DHCP sur le réseau. Donc IP dynamique.
 - ❖ Les quatre lignes suivantes permettent de configurer le wifi via le fichier de configuration de **wpa_supplicant**.
- ✚ Changement de fichier de configuration, cette configuration permet de placer une adresse IP statique pour cela nous allons changer l'interface Ethernet (eth0) en mettant à jour la configuration de l'interface : IP, masque et passerelle comme indique la figure suivante :

```

root@dex:/etc# cd network/
root@dex:/etc/network# ls
if-down.d      if-pre-up.d  interfaces    run
if-post-down.d if-up.d      interfaces.bkp
root@dex:/etc/network# cat interfaces
auto lo

iface lo inet loopback
iface eth0 inet static
address 10.21.18.26
netmask 255.255.248.0
gateway 10.21.16.100
dns-nameservers 8.8.8.8

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

```

Figure 16: fichier de configuration réseau

La ligne après gateway (passerelle) sert à définir les serveurs de noms de domaine (DNS) pour qu'on puisse se connecter à une ressource réseau externe (convertir un nom alphanumérique « par exemple `www.google.com` » en une adresse réseau numérique « `172. 217.19.164`»). Le DNS `8.8.8.8` est un DNS Google.

✚ Finalement le redémarrage de Raspberry Pi.

Alors à cette étape nous avons donné à notre carte une adresse IP fixe qui va nous servir à la manipulation de la carte à distance via le protocole **SSH** et/ou **VNC**.

1.2.Connexion à distance via SSH

Secure **Shell** est un protocole de réseau crypté pour initier des sessions Shell textuelles sur des machines distantes de manière sécurisée. Cela permet à un utilisateur d'exécuter des commandes sur l'invite de commande d'une machine sans qu'ils soient physiquement présents à proximité de la machine.SSH a été créé en 1995 pour le principal but est de permettre la prise de contrôle à distance d'une machine à travers une interface en lignes de commande.

Pour faire le contrôle à distance sur le Raspberry Pi on doit installer le SSH du côté d'un serveur SSH sur notre Raspberry Pi et d'un autre d'un client SSH sur notre ordinateur, mais heureusement le serveur SSH est déjà installé et activé par défaut sur la Raspberry Pi, ce qui reste seulement d'installer un client SSH sur notre ordinateur ; pour

cela on va installer le logiciel **Putty** sous Windows qui est un client SSH.

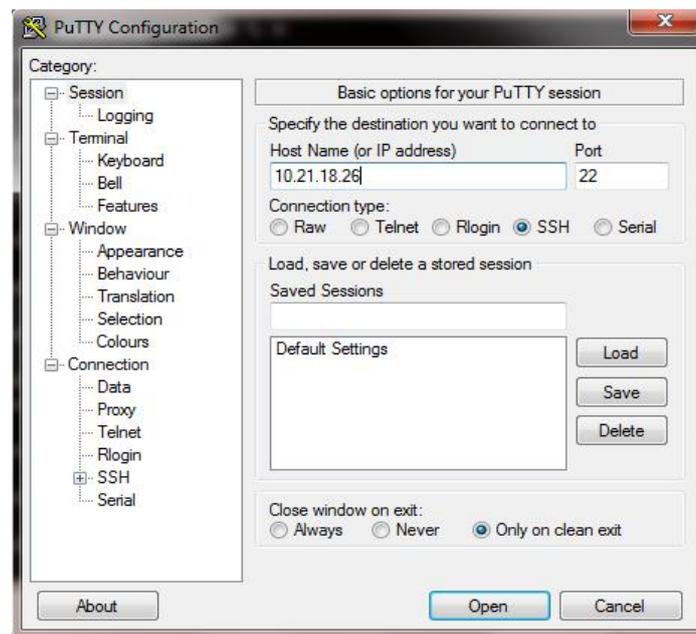


Figure 17: Logiciel Putty.

Après son installation on entre l'adresse IP de la carte puis on clique sur « **Open** », et après l'authentification (en entrant le nom d'utilisateur **pi** et notre mot de passe), on se connectera directement sur l'invité des commandes de notre système d'exploitation Raspbian For robots comme l'indique la figure suivante :

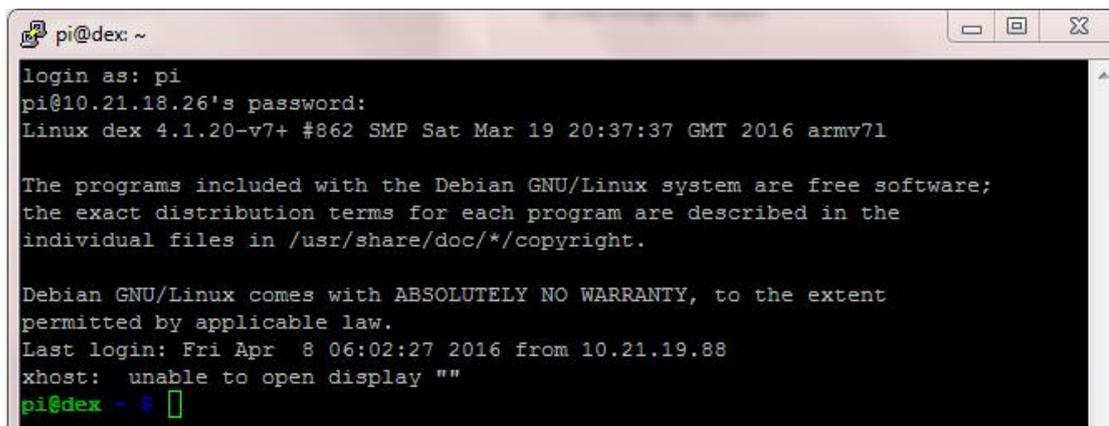


Figure 18: Invite de commande via SSH

1.3. Connexion à distance via VNC

Virtual Network Computing est un système de visualisation et de contrôle d'un ordinateur distant. Il permet au logiciel client VNC de transmettre les informations de

saisie du clavier et de la souris à l'ordinateur distant, possédant un logiciel serveur VNC à travers un réseau informatique. Il utilise le protocole RFB qui est un protocole simple pour l'accès à distance aux interfaces graphiques des utilisateurs.

Comme pour le SSH, le contrôle à distance sur le Raspberry Pi nécessite une installation du VNC du côté d'un serveur sur notre Raspberry Pi et d'un autre d'un client VNC sur notre ordinateur, la distribution de Dexter Industries « Raspbian For Robots » contient déjà un serveur VNC, il ne reste seulement que d'installer un client VNC sur notre ordinateur, le logiciel qui va jouer le rôle d'un client VNC est **TightVNC VIEWER**. La connexion est très simple en entrant l'adresse IP suivie de : 1 indiquant le port de l'interface graphique, puis en cliquant sur « **Connect** » comme montre la figure suivante :

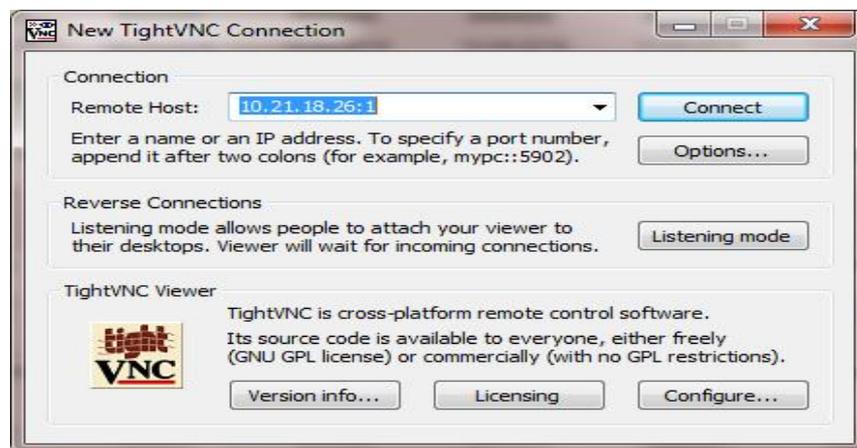


Figure 19: Logiciel TightVNC VIEWER

Après l'authentification (en entrant notre mot de passe) le logiciel **TightVNC VIEWER** affiche l'interface graphique du système d'exploitation (voir figure suivante), donc à ce moment on a tous les droits et toute la liberté d'utiliser l'interface graphique de notre système d'exploitation sur notre ordinateur en bénéficiant du clavier, souris et écrans de notre propre ordinateur, comme si le Raspberry est branchée à un écran par le câble HDMI et utilise un clavier et souris via les ports USB.

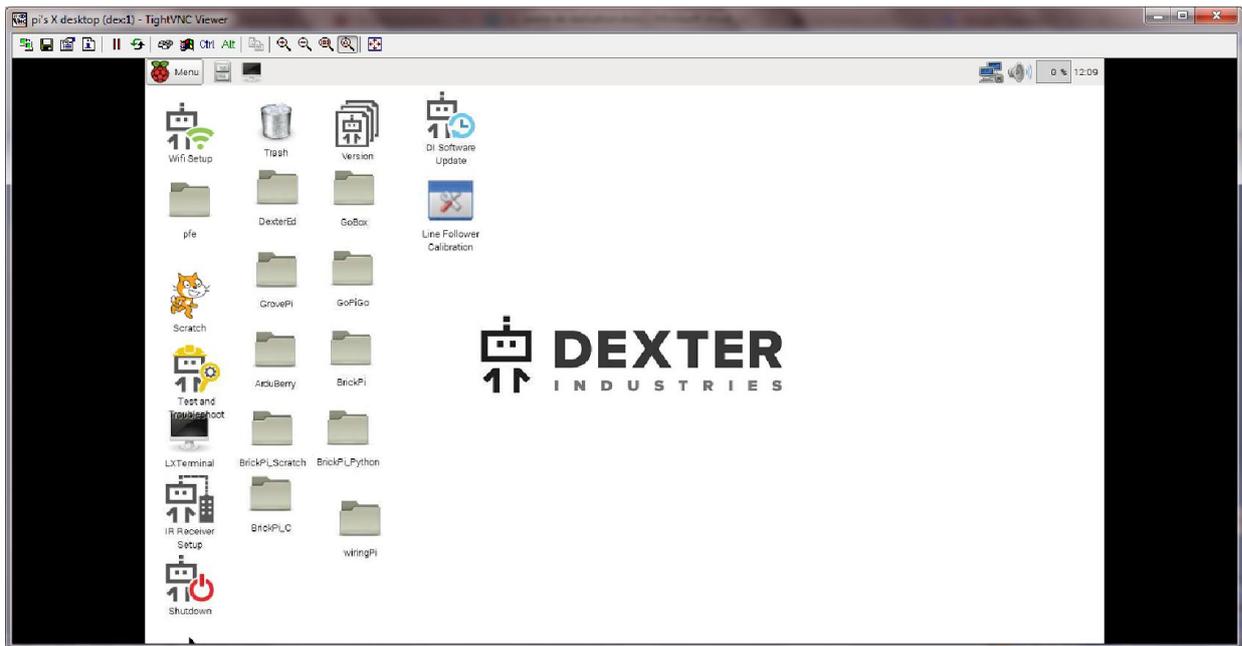


Figure 20 : Interface graphique sur logiciel TightVNC VIEWER

2. Langage de programmation

Le langage de programmation que nous avons choisi c'est le **python** qui est un langage recommandé pour les nouvelles technologies comme par exemple le Raspberry, beaglebone, arduino...

Ce choix n'est pas arbitraire car il a des raisons parmi eux : la société Dexter Industries recommande ces utilisateurs de la Grove Pi de programmer avec le python, l'existence des bibliothèques facilitant la programmation du Raspberry Pi, la richesse documentaire de ce langage, l'existence des exemples fournis avec le système d'exploitation Raspbian For Robots aidant le programmeur à apprendre rapidement la main sur la programmation des nouvelles cartes technologique, la souplesse de ce merveilleux langage avec la gestion des fichiers et bases de données... pour plus d'informations concernant python voir l'annexes.

3. Test du Grove Pi et capteurs

Chaque système dans le monde industriel suit une procédure scientifique et technologique pour qu'il puisse sortir au marché sans aucun problème, et l'étape la plus importante c'est l'étape des tests pour cela on a décédé dans ce paragraphe de tester

chaque élément tout seul afin de les faire fonctionner dans un seul système.

3.1.Grove Pi

Comme nous avons présenté dans le chapitre précédent le Grove Pi est un module d'adaptation des capteurs de type Grove avec la carte Raspberry Pi, donc il est indispensable de tester ce module en suivant le guide de la société Dexter Industries et comme on a fait le bon choix d'utiliser le système d'exploitation Raspbian For Robots créée par cette société, on a déjà sur le bureau de notre Raspberry Pi des programme qui nous aide à faire ce test:

- ✚ On Branche le Grove Pi sur le Raspberry Pi comme illustré dans cette image :



Figure 21: Brochage du Grove Pi sur le Raspberry

- ✚ Après le branchement on alimente la carte Raspberry puis on accède à notre système d'exploitation via VNC.
- ✚ Pour vérifier la version du micro logiciel (firmware) on exécute le programme **grove_firmware_version_check.py** situé à **Desktop/Grove Pi/Software/Python/** comme montre la figure suivante :

```
root@dex:/home/pi/Desktop/GrovePi/Software/Python# python grove_firmware_version_check.py
('GrovePi has firmware version:', '1.2.2')
root@dex:/home/pi/Desktop/GrovePi/Software/Python#
```

Figure 22 : Version de firmware de Grove Pi

- ✚ Vérification de la connexion entre la Grove Pi et le Raspberry : toute la communication entre le Grove Pi et le Raspberry se passe par l'intermédiaire du port I2C du Raspberry, alors on lance la commande **i2cdetect** comme montre la figure suivante :

```
root@dex:/home/pi/Desktop/GrovePi/Firmware# i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  04  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@dex:/home/pi/Desktop/GrovePi/Firmware# █
```

Figure 23: Test de la communication Grove et Raspberry

Comme il est indiqué dans le guide « Si vous pouvez voir un "04" dans la sortie, cela signifie que le Raspberry Pi est capable de détecter la Grove Pi », la sortie de notre commande montre qu'on a le « 04 » donc notre Raspberry Pi est capable de communiquer avec la Grove Pi.

- ✚ Finalement pour tester les sorties de la Grove Pi on va exécuter un programme qui fait clignoter une LED connectée sur le port 4 de la Grove Pi comme montre cette figure :

```
root@dex:/home/pi/Desktop/GrovePi/Software/Python# python grove_led_blink.py
This example will blink a Grove LED connected to the GrovePi+ on the port labeled D4. If you're having trouble seeing the LED blink, be sure to check the LED connection and the port number. You may also try reversing the direction of the LED on the sensor.

Connect the LED to the port labeled D4!
LED ON!
LED OFF!
```

Figure 24: Clignotement d'une LED

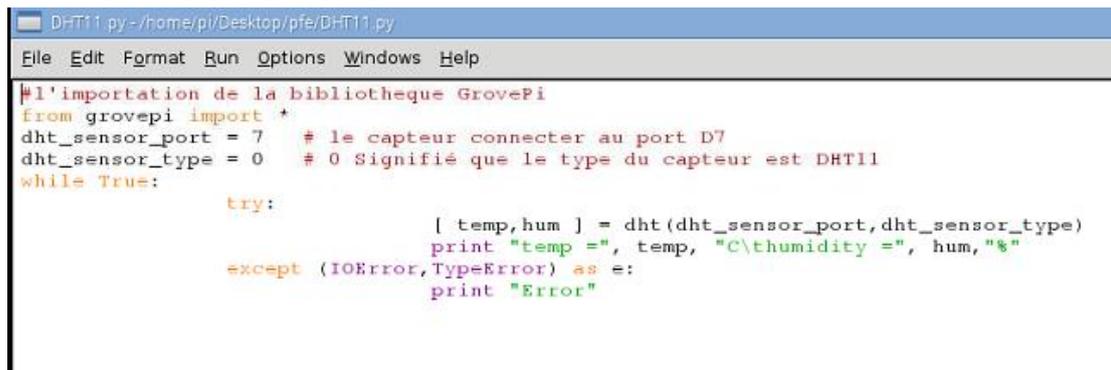
Le programme exécuté marche très bien et la LED branchée clignote donc notre Grove Pi fonctionne correctement sans aucun doute.

3.2. Capteur de température et d'humidité DHT11

Le capteur DHT11 nous permet de mesurer la température et l'humidité de la

salle des serveurs, pour cela on va faire un petit programme en python qui nous permet de mesurer ces données physique et les afficher sur le terminal.

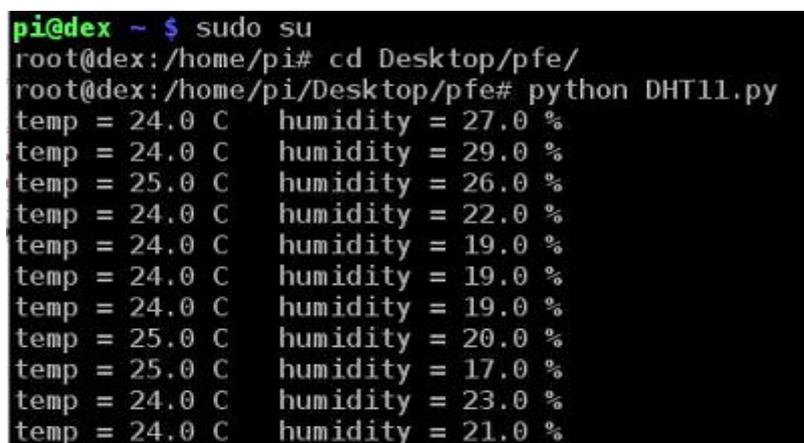
Premièrement on va brancher notre capteur sur l'un des 7 connecteurs digitaux de la Grove Pi par exemple le connecteur D7, puis on va écrire notre programme et l'exécuté afin de vérifier le fonctionnement du capteur DHT11. Le programme qu'on a créé pour le test est le suivant :



```
DHT11.py - /home/pi/Desktop/pfe/DHT11.py
File Edit Format Run Options Windows Help
#l'importation de la bibliotheque GrovePi
from grovepi import *
dht_sensor_port = 7 # le capteur connecter au port D7
dht_sensor_type = 0 # 0 Signifié que le type du capteur est DHT11
while True:
    try:
        [ temp,hum ] = dht(dht_sensor_port,dht_sensor_type)
        print "temp =", temp, "C\thumidity =", hum,"%
    except (IOError,TypeError) as e:
        print "Error"
```

Figure 25: Programme de test du DHT11

L'exécution de ce programme affiche sur le terminal les valeurs de la température et d'humidité donc le capteur DHT11 fonctionne sans aucun problème.

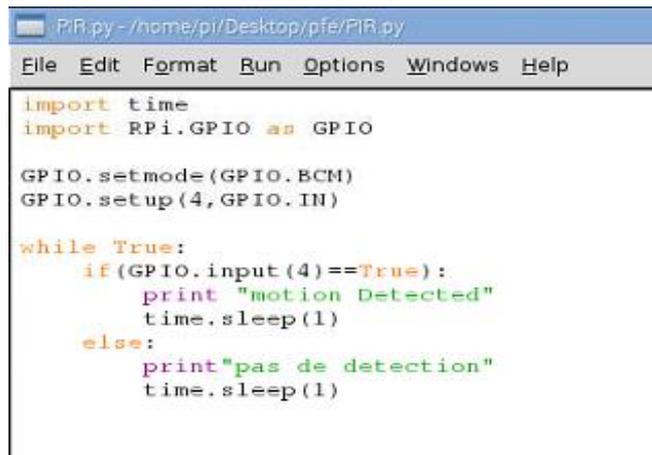


```
pi@dex ~ $ sudo su
root@dex:/home/pi# cd Desktop/pfe/
root@dex:/home/pi/Desktop/pfe# python DHT11.py
temp = 24.0 C humidity = 27.0 %
temp = 24.0 C humidity = 29.0 %
temp = 25.0 C humidity = 26.0 %
temp = 24.0 C humidity = 22.0 %
temp = 24.0 C humidity = 19.0 %
temp = 24.0 C humidity = 19.0 %
temp = 24.0 C humidity = 19.0 %
temp = 25.0 C humidity = 20.0 %
temp = 25.0 C humidity = 17.0 %
temp = 24.0 C humidity = 23.0 %
temp = 24.0 C humidity = 21.0 %
```

Figure 26: Résultat de test du DHT11

3.3.Capteur PIR

Le capteur PIR est le capteur qui nous permet de détecter la présence d'une personne ou d'une intrusion à la salle des serveurs. Après le brochage du capteur avec le Raspberry on va tester le programme suivant :

A screenshot of a text editor window titled 'PIR.py - /home/pi/Desktop/pfe/PIR.py'. The window contains a Python script that imports the 'time' module and the 'GPIO' module from the 'RPI' package. It sets the GPIO mode to BCM and configures pin 4 as an input. A while loop continuously checks the state of pin 4. If it is high (True), it prints 'motion Detected' and sleeps for 1 second. If it is low (False), it prints 'pas de detection' and sleeps for 1 second.

```
PIR.py - /home/pi/Desktop/pfe/PIR.py
File Edit Format Run Options Windows Help

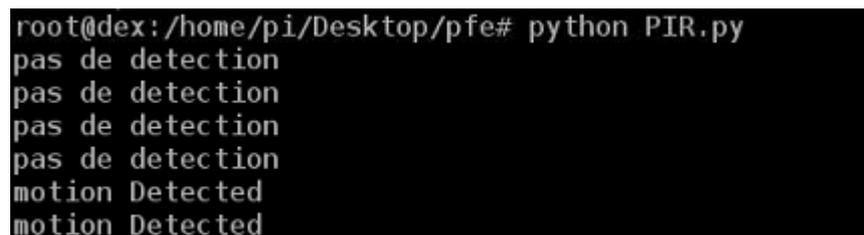
import time
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.IN)

while True:
    if(GPIO.input(4)==True):
        print "motion Detected"
        time.sleep(1)
    else:
        print "pas de detection"
        time.sleep(1)
```

Figure 27: Programme test du PIR

Ce programme nous permet à chaque détection d'écrire « motion Detected » et si on n'a pas de détection d'écrire « pas de détection », le test du programme donne les résultats prévus comme suit :

A terminal window showing the execution of the PIR test program. The prompt is 'root@dex:/home/pi/Desktop/pfe#'. The command 'python PIR.py' has been executed, resulting in the following output: 'pas de detection', 'pas de detection', 'pas de detection', 'pas de detection', 'motion Detected', and 'motion Detected'.

```
root@dex:/home/pi/Desktop/pfe# python PIR.py
pas de detection
pas de detection
pas de detection
pas de detection
motion Detected
motion Detected
```

Figure 28: Résultat de test du PIR

4. Test du camera

La camera est notre élément de surveillance de la salle, elle prend des photos en cas d'intrusion et les enregistrer dans un répertoire qui sera nommé par la date de la détection et les photos seront nommées par l'heure de la détection. Le test de cette caméra est fait en deux manières différentes :

Test avec Motion :

Motion est un logiciel pour la surveillance vidéo et à accès en ligne, disponible pour le Raspberry Pi. Après l'installation et la configuration de ce logiciel on peut accéder à la vidéo diffusée sur le port 8080 (par défaut) du Raspberry Pi, la figure suivante montre le résultat du test :

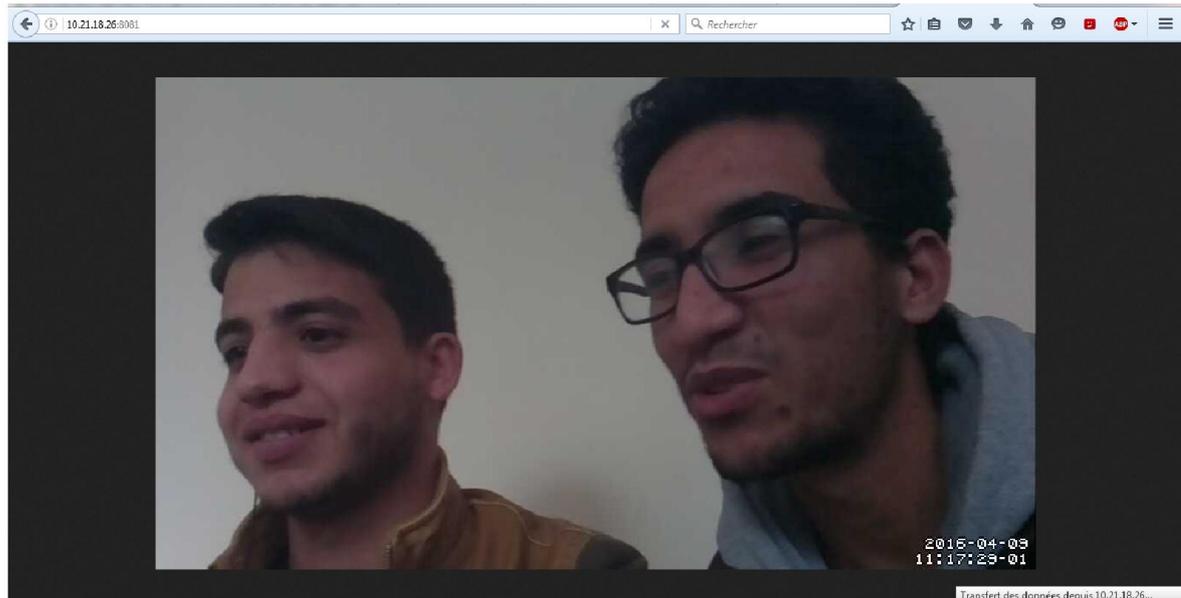


Figure 29: Test de camera par motion

Test avec Picamera

Picamera est une bibliothèque python dédiée au camera « **Raspberry Pi Camera** » et qui est utilisée dans la version Python 2.7 ou supérieure.

Après l'installation de la bibliothèque **python-picamera** avec la commande « `sudo apt-get install python-picamera` », nous avons crée un programme python pour tester la camera :

```
image.py ~/home/pi/Desktop/pfe/image.py
File Edit Format Run Options Windows Help
import picamera
from time import sleep

camera = picamera.PiCamera()
camera.capture('pi_image.jpg')
sleep(2)
camera.capture('pi_image1.jpg')
```

Figure 30: Programme test camera

Ce programme permet de prendre deux photos nommées successivement, « `pi_image.jpg` » et `pi_image1.jpg` » et les enregistrer dans le même répertoire du programme qui est « `/home/pi/Desktop/pfe/` » voici le résultat obtenu :

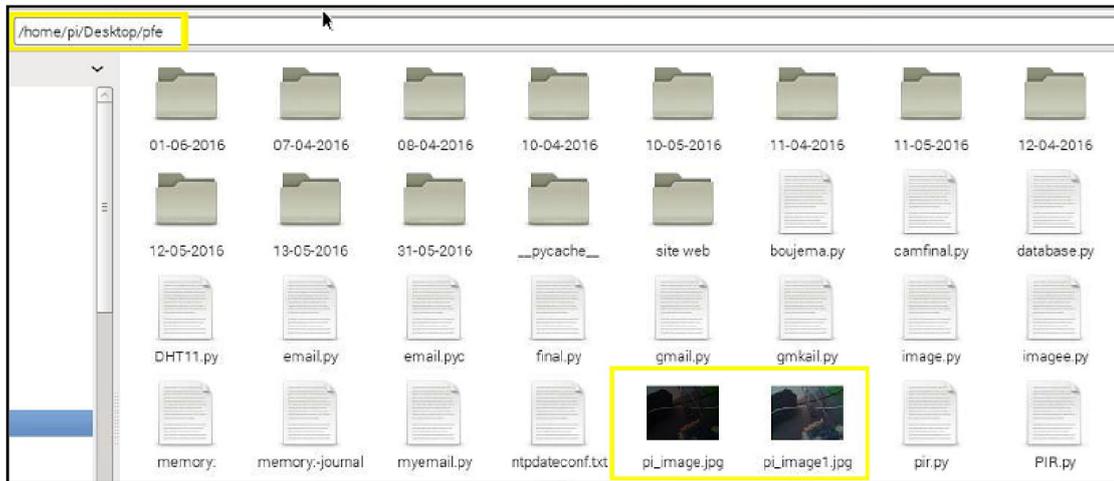


Figure 31: Test de camera par Picamera

II. Le système de surveillance de la salle serveurs

1. Schéma descriptif du système

Notre projet consiste à réaliser les trois fonctions suivantes:

- ✚ La collection des données (température, humidité et état du capteur PIR) pour des temps prédéfinies et les stocker dans une base de données MySQL, cette dernière doit être créée sur le Raspberry Pi.
- ✚ Le contrôle de l'accès à la salle des serveurs et en cas de détection d'intrusion, le système doit prendre des photos nommées par l'heure de la détection et les enregistrer dans un répertoire nommé par la date de détection.
- ✚ La visualisation des données enregistrées sur la base de données MySQL par deux applications, la première est une application Web et la deuxième est une application Java.

Le système de surveillance de la salle serveurs peut être représenté par le schéma suivant:

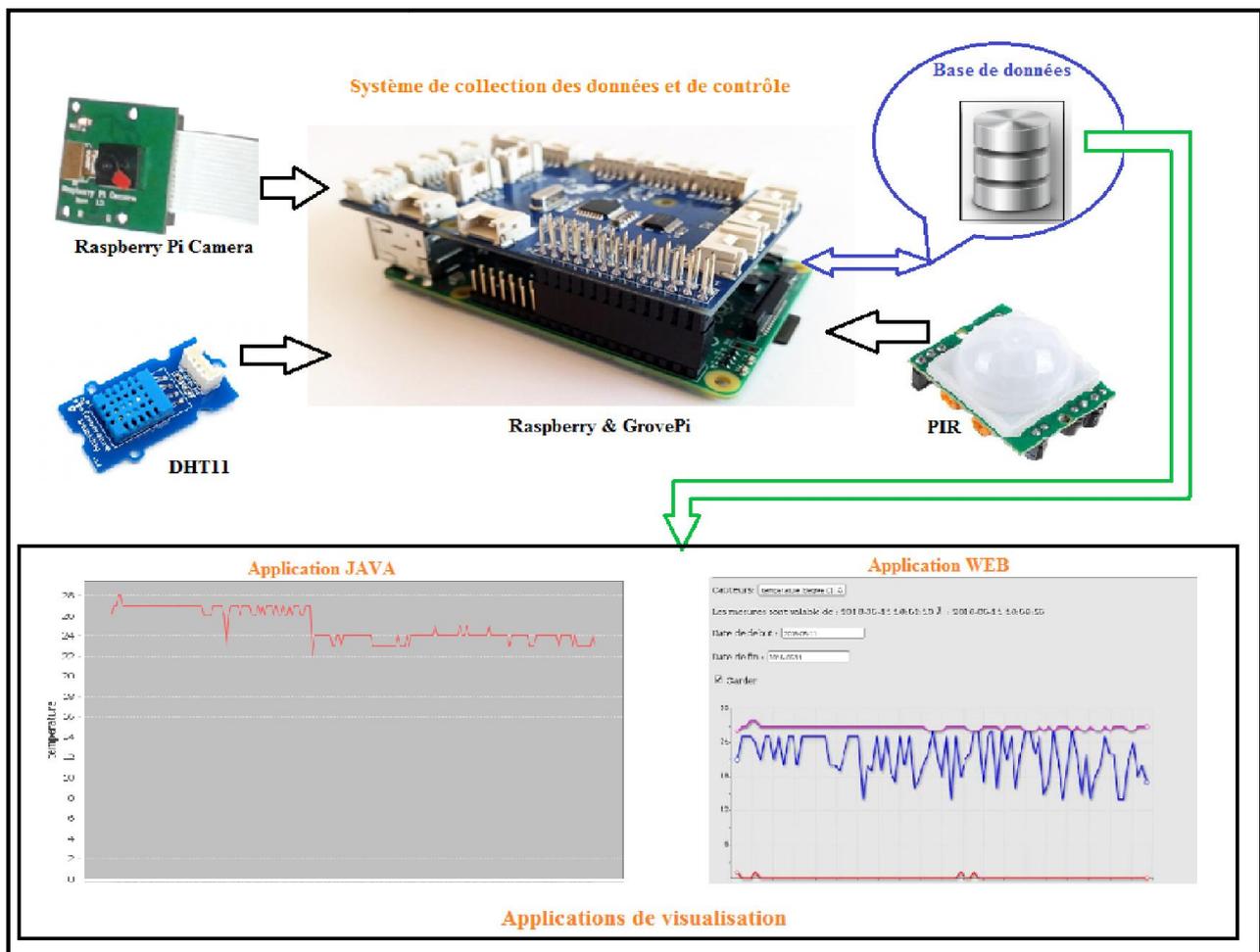


Figure 32: Schéma descriptif du système

2. Système de collection de données et de contrôle

Le Système de collection de données et de contrôle est constitué des éléments déjà testés (voir Configuration du système à base du Raspberry Pi) : le Raspberry Pi, le Grove Pi, le capteur de température et d'humidité DHT11, le capteur PIR et la camera « Raspberry Pi Camera Rev 1.3 », la figure suivante montre le montage réalisé:

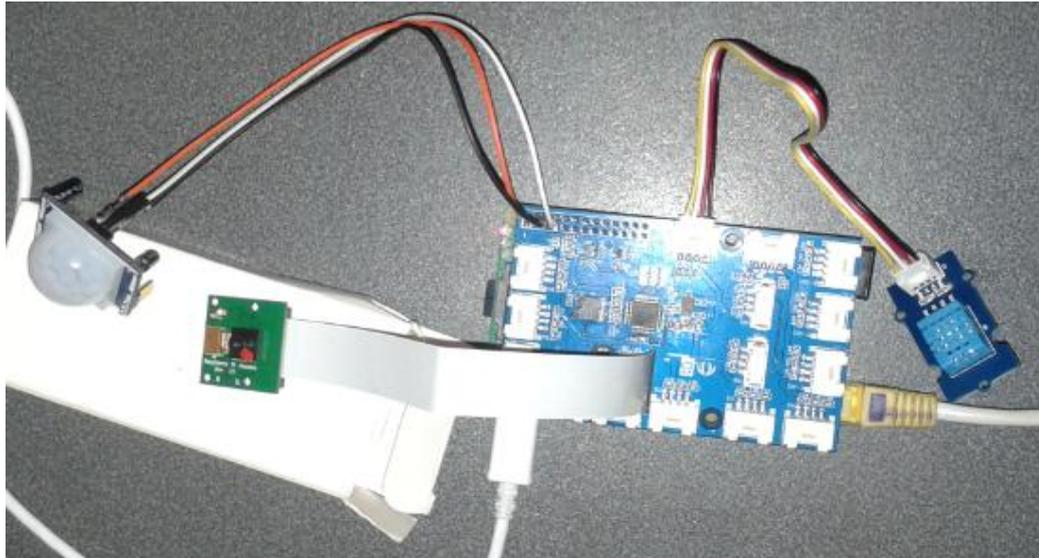


Figure 33: Montage du système de collection de données et contrôle

Comme on a déjà montré dans la partie des tests, le Grove Pi est inséré sur le Raspberry Pi, le capteur DHT11 connecté au port D7 du Grove Pi, le capteur PIR relié au port GPIO et la camera connectée au connecteur CSI du Raspberry Pi.

Le système permet de prendre les valeurs de la température, d'humidité et l'état du capteur PIR à chaque 5 minute (cette période de collection est modifiable selon le rythme de variation de ces données), et les enregistrer dans la base de données MySQL nommée DB_MANAGEM. A chaque détection d'une personne par le capteur PIR une interruption est générée afin de nous permettre de prendre des photos et les enregistrer.

3. L'organigramme du programme principal

Le programme global du système de collection et d'enregistrement des données dans la base de données est décrit dans l'organigramme suivant :

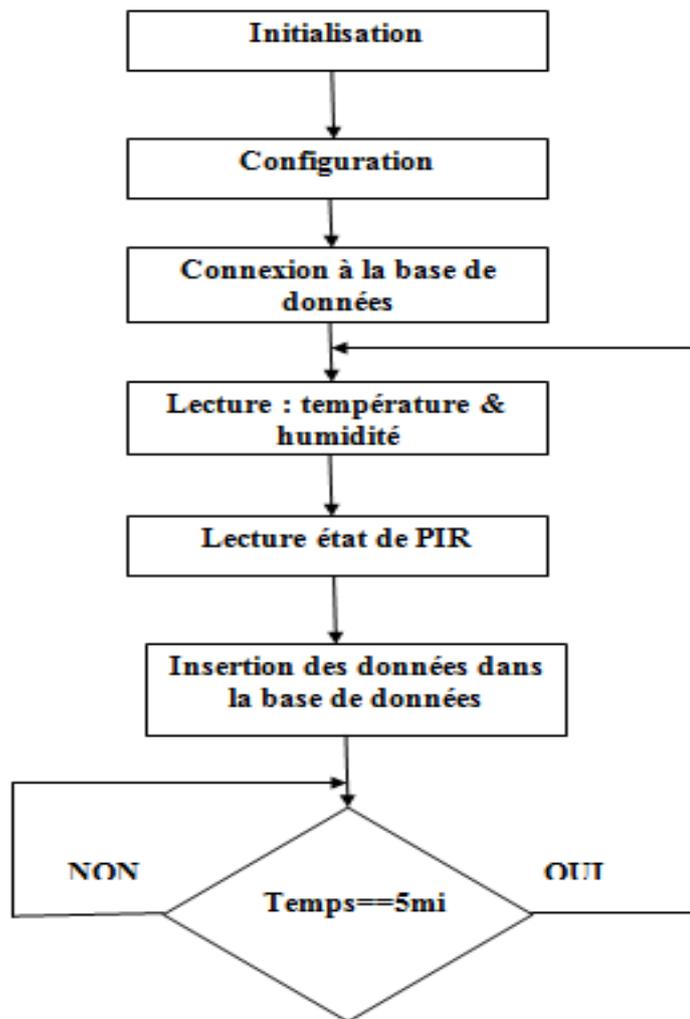


Figure 34: Organigramme du programme principal

Le principe de fonctionnement du programme est le suivant:

- ✚ Initialisation : dans cette étape, on importe toutes les bibliothèques nécessaires pour le bon fonctionnement du programme python (Grove Pi, RPi.GPIO, time, os, picamera, MySQLdb).
- ✚ Configuration : à ce moment on choisit le port de connexion du capteur DHT11 et les pins utilisés par le capteur PIR et ces configurations (GPIO.setmode= type de numérotation, GPIO.setup=configuration soit en entrée eu en sortie).
- ✚ Connexion à la base de données : en choisissant le nom de la base de données et son utilisateur et mot de passe.
- ✚ Lectures des données : température, humidité et l'état du capteur PIR (haut ou

bas).

- ✚ Insertion des données dans la base de données : chaque donnée est enregistrée dans une ligne de la base donnée avec la date et l'heure d'enregistrement.

- ✚ Temporisation de 5 minutes détermine la période de la collection de données.

Notre programme principal contient une **routine d'interruption** « **callback_up(PIR)** » qui s'exécute à chaque front montant de la sortie du capteur PIR (c'est-à-dire que le capteur a détecté la présence d'une personne).

4. L'organigramme de la routine d'interruption

Le programme de la routine d'interruption qui sert à la détection d'intrusion est décrit dans l'organigramme suivant :

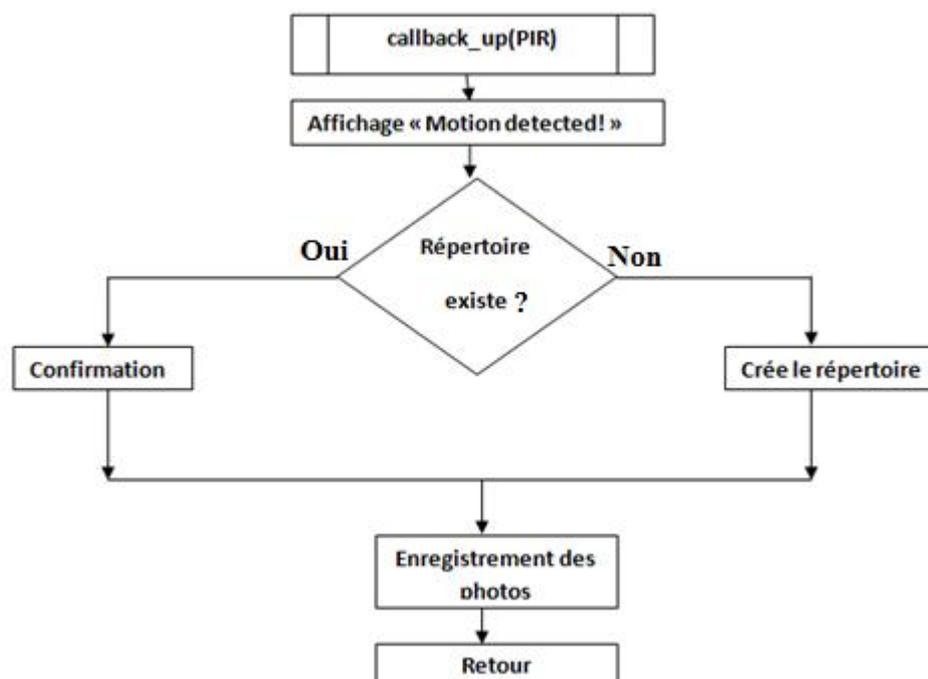


Figure 35 : Organigramme de la routine d'interruption

Le fonctionnement de cette routine est le suivant:

- ✚ Afficher sur le terminal le message « Motion detected! » pour indiquer la détection d'une personne.

- ✚ Tester l'existant du répertoire nommé par le jour de la détection, sinon le programme doit le créer.
- ✚ Enregistrer les deux photos dans le répertoire déjà créé nommées par l'heure de la capture.
- ✚ Retour au programme principal.

A la fin, on trouve dans notre chemin `/home/pi/Desktop/pfe` des répertoires indiquant les jours où il y a une détection d'une personne, et chaque répertoire contient les photos de ces personnes détectées, comme montre les figures suivantes :

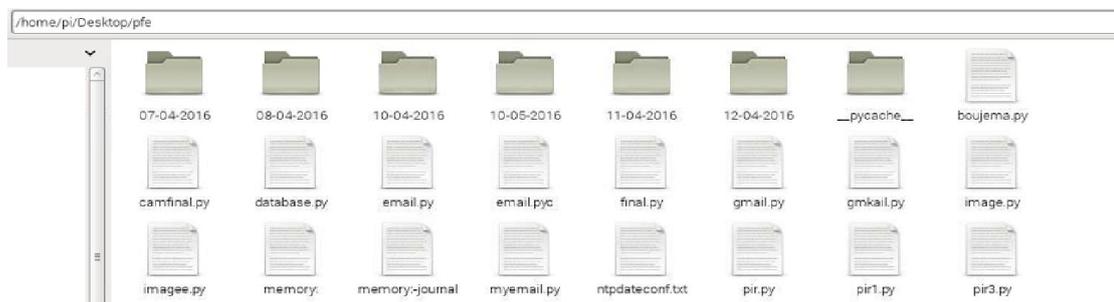


Figure 36: Les répertoires du chemin `/home/pi/Desktop/pfe`

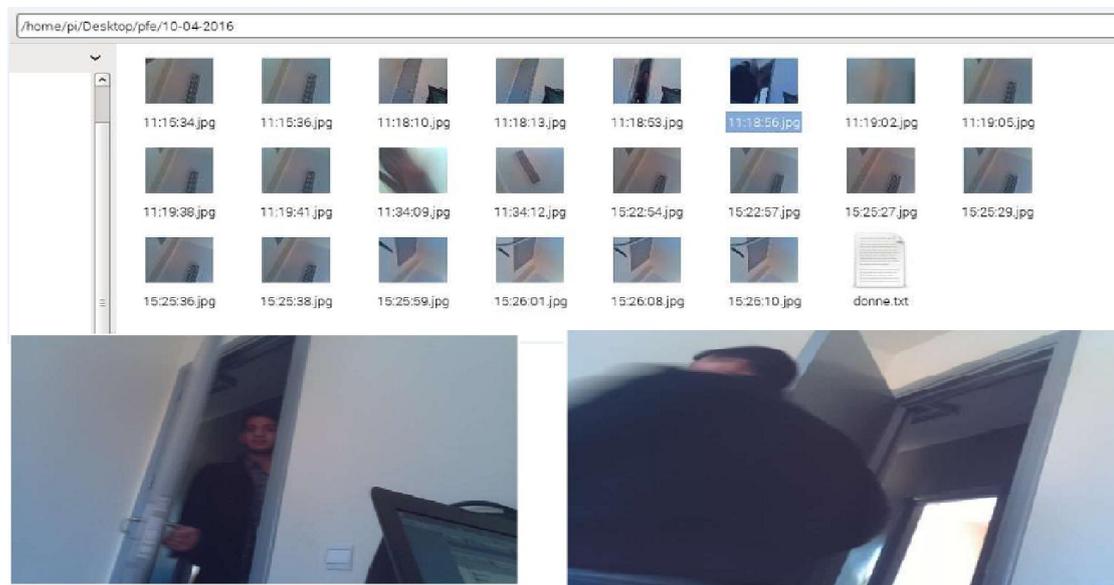


Figure 37: Photos prises au moment des détections

5. Création de la base de données MySQL

5.1. Installer MySQL

Pour ce faire, nous allons installer mysql-server et php5-mysql. La commande suivante installe mysql-server, avec le paquet qui fournit le module pour les connexions de base de données MySQL directement à partir de scripts PHP.

```
root@dex:/home/pi# sudo apt-get install mysql-server php5-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
mysql-server is already the newest version.
The following NEW packages will be installed:
  php5-mysql
0 upgraded, 1 newly installed, 0 to remove and 26 not upgraded.
Need to get 69.2 kB of archives.
After this operation, 180 kB of additional disk space will be used.
Do you want to continue [Y/n]? y
```

Installation de mysql-server et php5-mysql

Figure 38: Installation de mysql-serveur et php5 mysql

Lors de l'installation de mysql-server, il nous demande un mot de passe pour le compte administrateur MySQL, qui sera utilisé plus tard dans l'accès au serveur.

5.2. Création de base de données

Pour la création de la base de données, on a utilisé la commande SQL :
CREATE DATABASE nom_de_base ;

Pour assurer qu'on a créé la base de données, on doit exécuter la commande suivante: **SHOW DATABASES ;**

5.3. Structure de la base de données

Après la création de la base de données, nous avons créé les tables nécessaires au bon fonctionnement du système, ces trois tables sont les suivantes:

```
mysql> SHOW TABLES;
+-----+
| Tables_in_DB_MANAGEM |
+-----+
| SERVEUR_BACKUP        |
| SERVEUR_BACKUP1      |
| sensorData            |
| sensorInfo            |
| temp_hum              |
| utilisateur           |
+-----+
```

Figure 39: Structure de la base de données

- ✚ Une table « **sensorInfo** » qui contient les informations sur le type de capteurs du système.

```
mysql> SELECT * FROM sensorInfo;
+-----+-----+-----+
| sensor_id | type_sensor | unit      |
+-----+-----+-----+
|          1 | temperature | degree C |
|          2 | humidite    | %         |
|          3 | PIR         | booleene  |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Figure 40: Table « sensorInfo » de base de données « DB_MANAGEM »

- « **sensor_id** » chaque grandeur à un « **sensor_id** » spécifique par exemple si la valeur de **sensor_id=1** automatiquement c'est la valeur de température.
 - « **type_sensor** » indique le type de capteur (par exemple la température, l'humidité, ou PIR).
 - « **unit** » l'unité de la grandeur mesurée (comme degrés C, %, booléenne).
- ✚ Une table « **sensorData** » stocke les mesures réelles qui sont périodiquement lues à partir des capteurs, en fonction de leur date et leur temps de collection ou de détection.

```
mysql> DESCRIBE sensorData;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| measur_id | int(11)       | NO   | PRI | NULL    | auto_increment |
| sensor_id | int(11)       | NO   |     | NULL    |                |
| date      | date          | NO   |     | NULL    |                |
| time      | time         | NO   |     | NULL    |                |
| value     | varchar(50)  | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Figure 41: Table « sensorData » de base de données « DB_MANAGEM »

- « **measur_id** » est la clé primaire de la table « **sensorData** » chaque valeur à son id.
- « **sensor_id** » est la clé étrangère qui relie les deux tableaux, « **sensorData** » et « **sensorInfo** »
- « **date** » est la date de collection de l'information depuis les capteurs.
- « **time** » est l'heure de collection de l'information ou de la détection depuis les capteurs.
- « **value** » la valeur physique de la grandeur en fonction de « **sensor_id** », par exemple c'est sensor_id=1, donc la valeur dans « **value** » est de la température (2 pour l'humidité est 3 pour PIR de détection).

```
mysql> SELECT * FROM sensorData;
```

measur_id	sensor_id	date	time	value
13	1	2016-05-11	16:52:10	26.0
14	2	2016-05-11	16:52:10	21.0
15	3	2016-05-11	16:52:10	1

Température
Humidité
PIR de détection

Figure 42: Table « **sensorData** » remplis

- Une table « **utilisateur** » contient les utilisateurs ayant l'accès aux applications que nous avons créé (java et web). La structure de la table est la suivante :

```
mysql> SELECT * FROM utilisateur;
```

id	id_user	username	password	type
1	2016	MANAGEM	PASS	admin

Figure 43: Structure de tableau « **utilisateur** »

Conclusion

Dans ce chapitre nous avons cité toutes les configurations, les tests des composants de notre système et nous avons expliqué le fonctionnement du programme python.

Chapitre IV :

Applications de supervision

Introduction

Ce dernier chapitre présentera les deux applications de supervision des données et les outils nécessaires pour développer ces derniers.

I. Le développement de l'application web

1. Installation du serveur web sur le Raspberry

Un **serveur Web** est un **serveur informatique** qu'on va utiliser pour publier ou héberger des **sites web** sur **Internet**, on va héberger notre application sur le Raspberry comme serveur web.

Un serveur web n'est seulement une machine, mais désigne aussi le **logiciel** qui permet à la machine d'analyser les requêtes des utilisateurs (requêtes http par exemple), et de retourner le fichier correspondant à la requête.

Dans ce cadre **Apache** est le logiciel qu'on va utiliser, cela est dû à plusieurs raisons, à l'heure actuelle, Apache est le serveur web le plus utilisé, avec environ 60 % de parts de marché, ainsi il possède même **sa propre licence**, utilisée par de nombreux autres projets, et le plus important l'utilisation massive d'Apache (Apache est devenu le standard des serveurs web), et sa forte popularité.

Installation d'Apache

Avant d'installer le serveur, on doit assurer que la machine est bien à jour. Pour ce faire nous devons posséder les droits d'administrateur, soit en étant connecté en **root**, soit via la commande **sudo**.

```
sudo aptitude update
```

```
sudo aptitude upgrade
```

Une fois le Raspberry Pi est à jour, nous avons installé, le serveur Apache.

```

root@dex:/home/pi# sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version.
libapache2-mod-php5 is already the newest version.
php5 is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 2
root@dex:/home/pi# sudo service apache2 restart
[....] Restarting web server: apache2apache2: Co
... waiting apache2: Could not reliably determi
. ok
root@dex:/home/pi# sudo service apache2 restart
[....] Restarting web server: apache2apache2: Co
... waiting apache2: Could not reliably determi
. ok
root@dex:/home/pi# █

```

Figure 44: Installation de serveur Apache2

2. Installation de PHP sur le Raspberry

Après l'installation de serveur Apache, on a besoin d'installer un interpréteur de langage PHP, afin d'utiliser le langage, qui est principalement utilisé pour rendre notre application dynamique, c'est-à-dire que l'utilisateur envoie des informations au serveur qui lui renvoie les résultats modifiés en fonction de ces informations, contrairement à un site statique qui ne s'adapte pas aux informations fournies par un utilisateur.

```

root@dex:/home/pi# sudo apt-get install mysql-server php5-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
mysql-server is already the newest version.
The following NEW packages will be installed:
  php5-mysql
0 upgraded, 1 newly installed, 0 to remove and 26 not upgraded.
Need to get 69.2 kB of archives.
After this operation, 180 kB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main php5-m
Fetched 69.2 kB in 7s (8,915 B/s)
Selecting previously unselected package php5-mysql.
Reading database ... 95074 files and directories currently installed
Unpacking php5-mysql (from ../php5-mysql_5.4.45-0+deb7u2_armhf.deb)
Processing triggers for libapache2-mod-php5 ...
[....] Reloading web server config: apache2apache2: Could not reliabl
. ok
Setting up php5-mysql (5.4.45-0+deb7u2) ...
Creating config file /etc/php5/mods-available/mysql.ini with new vers
Creating config file /etc/php5/mods-available/mysqli.ini with new ver
Creating config file /etc/php5/mods-available/pdo_mysql.ini with new
Processing triggers for libapache2-mod-php5 ...
[....] Reloading web server config: apache2apache2: Could not reliabl
. ok
root@dex:/home/pi# █

```

Figure 45: Installation de l'interpréteur PHP

L'idée d'utiliser PHP est pour obtenir les données stockées dans la base de données, puis utiliser JavaScript, Ajax et la bibliothèque **Rgraph** pour dessiner les graphiques.

3. Configuration de serveur

✚ Démarrer le serveur Apache2

Le serveur démarre automatiquement, mais nous pouvons le faire manuellement avec cette commande

```
service apache2 start
```

✚ Répertoire du serveur

Le répertoire du serveur sur le Raspberry est le suivant : `/var/www` (figure 46). Sur ce répertoire nous pouvons placer nos fichiers PHP et HTML de notre application. Pour ensuite y accéder, il faut aller à `ip_du_raspberry/nom_du_fichier.html`.

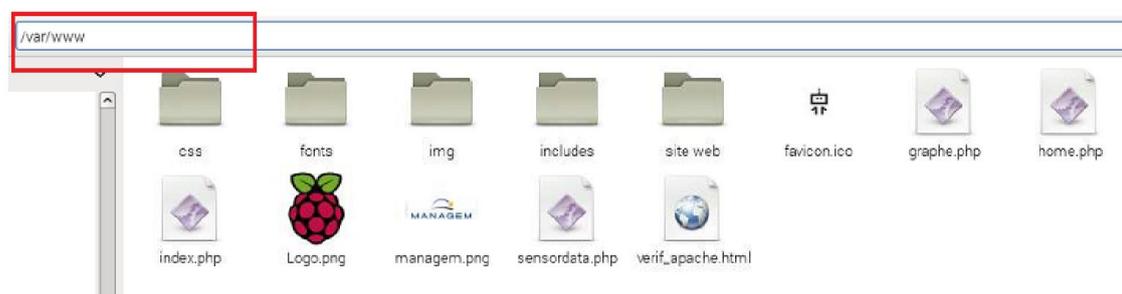


Figure 46: Répertoire de serveur apache sur Raspberry

4. Plan de l'application

Généralement l'application est constituée de deux pages principales, la page d'authentification et la page de surveillance.

- **Page d'authentification** : permet à l'utilisateur de s'identifier. Ce dernier devra donc saisir son login et son mot de passe. D'une manière générale, cette page représente le point d'entrée de toute l'application.
- **Page de surveillance** : permet de superviser l'état de salle de serveur, sous la

forme des graphes, ainsi permet à l'utilisateur de choisir un laps de temps et la grandeur qu'il désire afficher.

5. La connexion

Afin de pouvoir se connecter à la base de données, nous avons créé un fichier de connexion « **connection.php** » dans lequel on a écrit le code qui nous permet d'ouvrir la base de données et la connexion avec le serveur web. En effet, celui-ci est indispensable, puisqu'il nous permettra d'effectuer toutes les requêtes qui seront utiles par la suite. Le code permettant d'ouvrir la base de données et la connexion avec le serveur web Apache est le suivant :

```
<?php
// Serveur MySQL
$cfgHote="10.21.18.26";
// Utilisateur
$cfgUser="USER_DB";
// Mot de passe
$cfgPass="PASS"
mysql_connect('$cfgHote','$cfgUser','$cfgPass')or die('couldn\'t connect to the serveur!');
////Connexion à la base de données de l'application
mysql_select_db('db_managem') or die('couldn\'t find the database!! ');
?>
```

Figure 47: Fichier de connexion

En revanche, nous avons ajouté une ligne sur toutes les pages de l'application de manière à ce que celles-ci appellent la page de connexion. Cet appel est indispensable pour qu'on puisse interroger notre base de données et qui se présente de la manière suivante :

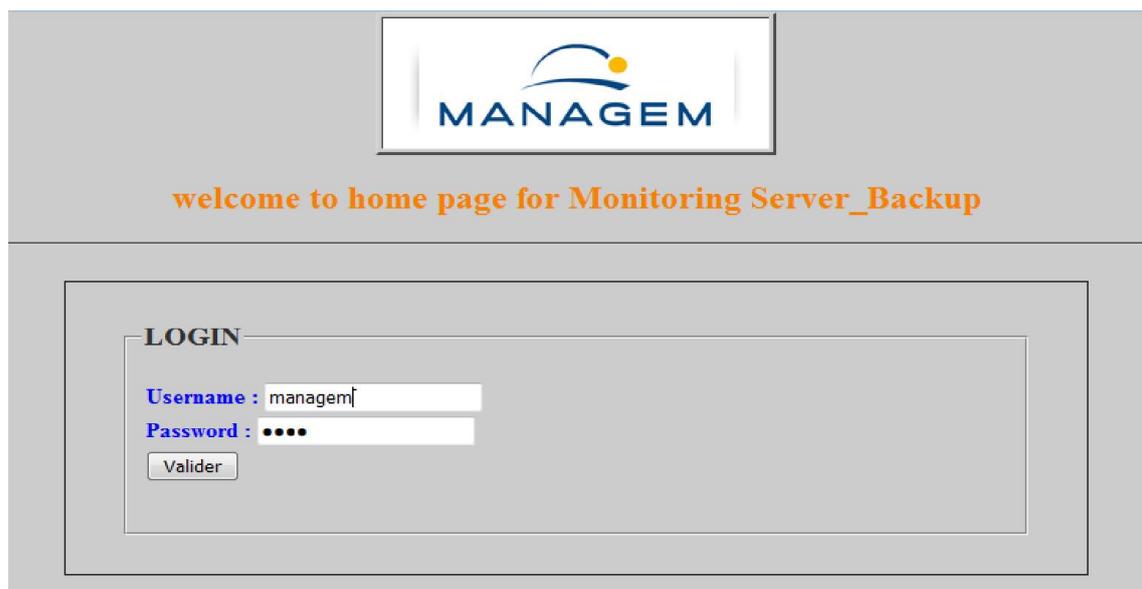
```
connection.php index.php
1 <?php
2 // Permet d'appeler la page de configuration pour accéder à la base de données
3 include('includes/connection.php');
4
5
6 ?>
7 <!doctype html>
8
9 <html>
10 <head>
```

Figure 48: Appelle de la fonction de connexion

6. Description de l'application

6.1. La page d'authentification

Comme la majorité des sociétés, Managem ne souhaite pas que son application soit consultable sur le web par tous les internautes. Pour cette raison, nous avons créé une page d'authentification qui demande à l'utilisateur de saisir son login et son mot de passe. Pour illustrer ces premières notions, vous trouverez ci-dessous la page d'authentification de notre application :



The screenshot shows a web page with a grey background. At the top center is the Managem logo, which consists of a blue arc above the word 'MANAGEM' in blue capital letters. Below the logo is the text 'welcome to home page for Monitoring Server_Backup' in orange. In the center of the page is a white rectangular box containing a login form. The form is titled 'LOGIN' in bold black text. It has two input fields: 'Username : managem' and 'Password : ●●●●'. Below the password field is a button labeled 'Valider'.

Figure 49: Page d'authentification

6.2. Page de Surveillance :

Sur cette page Web, l'utilisateur doit être en mesure de sélectionner l'un des capteurs disponibles avec le menu déroulant présenté sur la page, et le laps de temps souhaité pour lequel les lectures seront affichées.

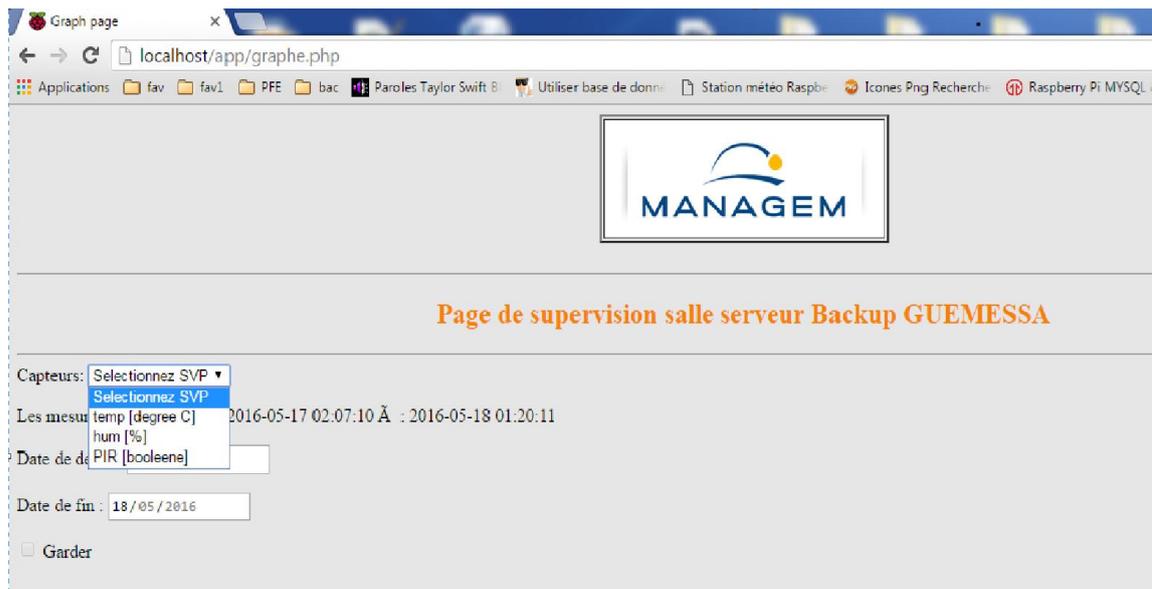


Figure 50: Page de Surveillance

Pour sélectionner l'un des capteurs disponibles, nous avons créé une fonction PHP, permet d'extraire les données de base de données MySQL. Ces informations sont utilisées dans la génération de menu déroulant, à partir du quel l'utilisateur sélectionne le capteur désiré.

```
function sensorInfo(){
    //connexion à MySQL et base de données
    include('includes/connection.php');
    //requête et de faire le menu déroulant en html
    $sql=mysql_query("SELECT * FROM sensorinfo");

    echo '<label for="selectedSensor">Capteurs: </label>';
    if(mysql_num_rows($sql)){
        $select= '<select id="selectedSensor"><option value=""> Selectionnez SVP </option>';
        while($rs=mysql_fetch_array($sql))
        {
            $select.='<option value="'. $rs['sensor_id'].'">'. $rs['type_sensor']. ' ['.$rs['unit'].'] </option>';
        }
    }
    $select.='</select>';
    echo $select;
}
```

Figure 51: Fonction PHP pour faire l'affichage du menu déroulant

Après le choix du capteur, l'utilisateur doit choisir notamment laps de temps où il désire afficher les données. Cependant, nous devons permettre à l'utilisateur de sélectionner uniquement la plage qui est disponible dans la base de données.

Capteurs:

Les mesures sont valable de : 2016-05-17 02:07:10 Ã 2016-05-18 01:20:11

Date de debut :

Date de fin : 18 mai 2016

Garder

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Figure 52: Extrait de la page de supervision

La fonction PHP suivante détermine l'intervalle des temps disponible dans la table "sensorData".

```
function getFirstLastTimeDate($what){
    //connection avec la base de données
    include('includes/connection.php');
    //la requete SQL pour obtenir la premier date dans la base de données
    $sql1=mysql_query("SELECT date, time from sensordata order by  measur_id asc limit 1");
    $rs1=mysql_fetch_array($sql1);
    //la requete SQL pour obtenir la derniere date dans la base de données
    $sql2=mysql_query("SELECT date, time from sensordata order by  measur_id desc limit 1");
    $rs2=mysql_fetch_array($sql2);
    //l'impression d'informations sur la gamme disponible sur le site
    if($what==0){
        $info = '<p> Les mesures sont valable de : '.$rs1['date'].' '.$rs1['time'].' à : '.$rs2['date'].' '.$rs2['time'].'</p>';
        echo $info;
    }
    //return la première date de entrée
    if($what==1)
        echo $rs1["date"];
    //return la dernier date de entrée
    if($what==2)
        echo $rs2["date"];
}
```

Figure 53: Fonction PHP défini la plage de temps disponible sur la base de données

Les fonctions ci-dessus fournissent les informations sur le capteur et le laps de temps disponible des mesures dans la base de données MySQL. Maintenant nous avons besoin d'obtient réellement les mesures demandées à partir de la base de données. Pour cela, nous avons créé un fichier PHP « sensor_data.php ».

Les graphes des mesures doit être mis à jour à chaque fois l'utilisateur change le

laps de temps ou un capteur, pour cela le code de « **sensor_data.php** » est appelé à partir de **JavaScript** sur la demande de **Ajax**.

```
$result = mysql_query("SELECT * FROM sensordata WHERE sensor_id=".$sensor ." AND date BETWEEN ".$startdate ." AND ".$stopdate ." ");

$myjsons = array();
while($row = mysql_fetch_assoc($result))
{
    $myjsons[] = $row;
}
echo json_encode($myjsons);

mysql_close($dbhandle);
```

Figure 54: Extrait de fichier « sensor_data.php »

Après que l'utilisateur sélectionne l'un des capteurs et choisi le laps de temps désiré, les graphes apparaissent automatiquement sous la forme suivante :

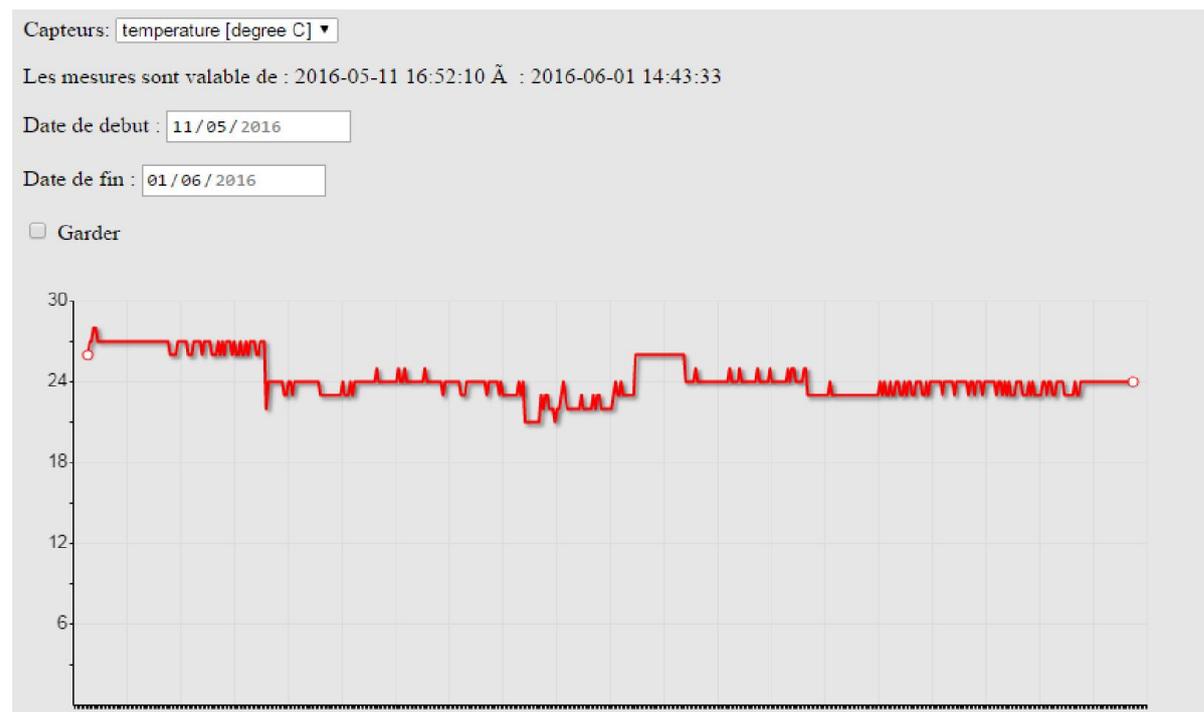


Figure 55: Graphe affiché

Finalement, nous avons ajouté une petite case à cocher sur le site. Si l'utilisateur veut garder l'ancien graphe, et afficher l'autre grandeur (ce qui est utile si vous voulez comparer les mesures des deux capteurs ou plus), les résultats se présentent comme suit:

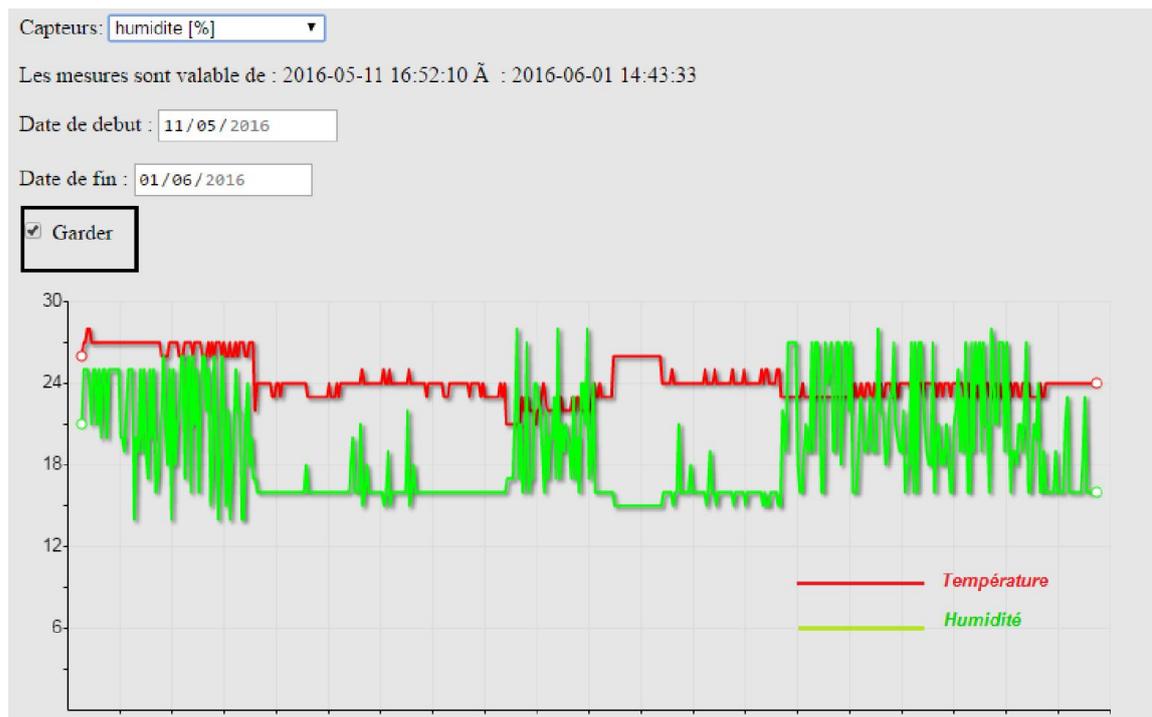


Figure 56: Graphe de différentes grandeurs (température et humidité)

II. Développement de l'application Java

Comme l'application web que nous avons présentée dans la partie précédente, l'application Java a le même rôle, c'est la supervision de la salle serveurs Backup de MANAGEM.

La conception de l'application Java s'est déroulée sur 4 étapes distinctes :

- Installation de l'environnement de travail.
- Téléchargement et l'importation des bibliothèques qu'on devra utiliser.
- Création de la structure de l'application en Java.
- Ouvrir une connexion entre l'application et la base de données.

1. L'environnement de travail « Netbeans »

L'outil de développement qu'on a utilisé pour la réalisation et le développement de cette interface est le Netbeans qui embarque les outils de la création des interfaces graphiques, facilitant la création de la partie visuelle de l'application, un éditeur de code

et un débogueur.

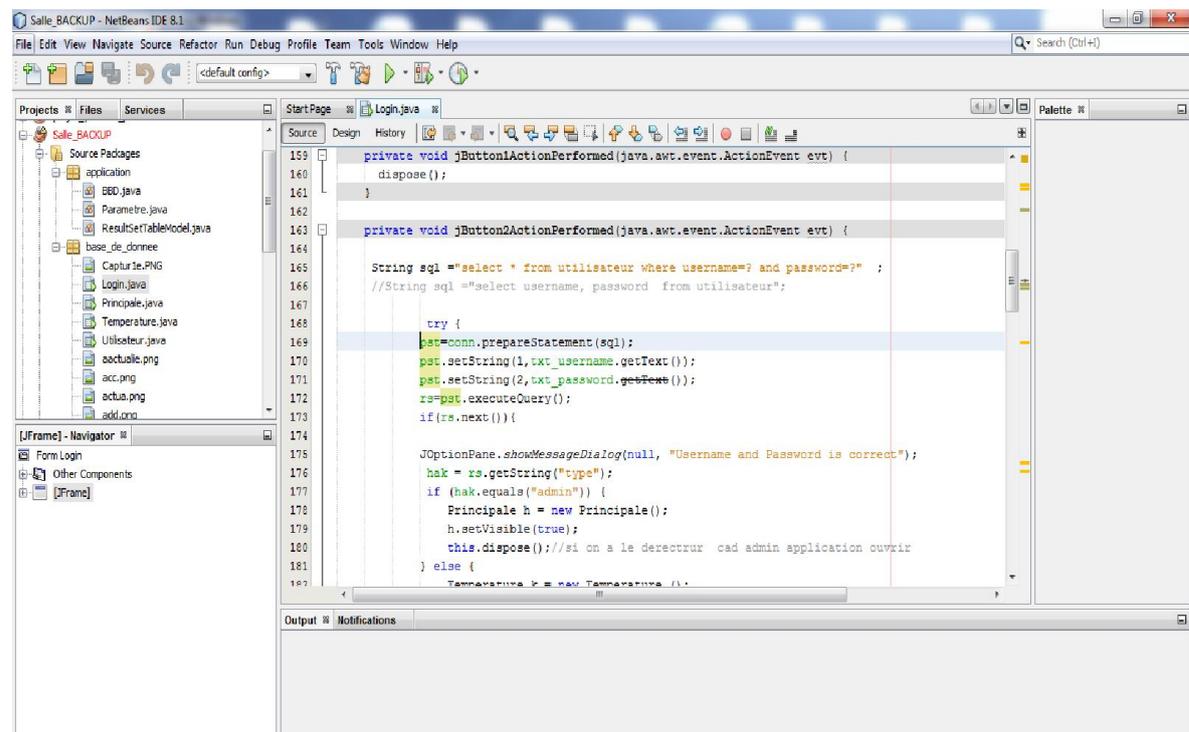


Figure 57: Interface du NetBeans

2. Importation des bibliothèques

Après l'installation de l'environnement NetBeans et la création de notre projet, on a besoin de faire des importations à des bibliothèques qui va nous aider par la suite. Parmi ces bibliothèques :

- ✚ « **mysql jdbc driver mysql-connector-java** » est un API pour le langage de programmation Java, qui permet de définir comment un client peut accéder à une base de données, interroger les données et faire les mises à jour dans la base de données. Tous ceci à l'aide des classes et des méthodes déjà développées.
- ✚ « **JfreeChart** » est un API qui va nous permettre de tracer les graphes de différente grandeur à partir de la base de données.



Figure 58: Bibliothèques importer

3. Accès à la base de données

La connexion avec la base de données se fait avec une fonction qu'on a écrit, pour exécuter des requêtes SQL. Ci-dessous un extrait de code java qui permet la connexion.

```

////////la fonction pour fiare la connexion ala base de donnee
public static Connection connexionDatabase() {
    try {
        Class.forName("com.mysql.jdbc.Driver"); |
        Connection conn= DriverManager.getConnection("jdbc:mysql://10.21.18.26/DB_MANAGEM","user","PASS");
        //JOptionPane.showMessageDialog(null, "connection correcte avec BD");

        return conn;
    }
    catch (Exception e) {
        System.err.println(e);//
    }
    return connection;
}

```

Figure 59: Fonction de connexion à la base donnée

4. Description de l'interface

L'interface est composée de plusieurs fenêtres présentées comme suit :

✚ La fenêtre d'authentification

Comme la page d'authentification de l'application web, cette fenêtre permet de contrôler l'accès à l'application, elle demande à l'utilisateur de saisir son login et son mot de passe. De plus cette fenêtre d'authentification permet d'identifier le type de l'utilisateur, si l'utilisateur de type « **admin** », donc il a l'accès à toutes les autres fenêtres qui suit (fenêtre de gestion des utilisateurs, fenêtre de surveillance), par contre si l'utilisateur est de type « **superviseur** » il a l'accès seulement à la fenêtre de surveillance.



Figure 60: Fenêtre d'authentification de L'application JAVA

Après l'identification de l'utilisateur une autre fenêtre doit être apparaître (fenêtre principale), cette fenêtre montre à l'utilisateur les choix principaux, la gestion de l'utilisateur ou la supervision de l'état de la salle serveur.



Figure 61: Fenêtre principale de choix

La fenêtre de gestion de l'utilisateur

Cette fenêtre a plusieurs fonctions :

- Ajouter un utilisateur.
- Modifier les données d'un utilisateur.

- Supprimer un utilisateur.
- Faire les recherches dans la base de données.

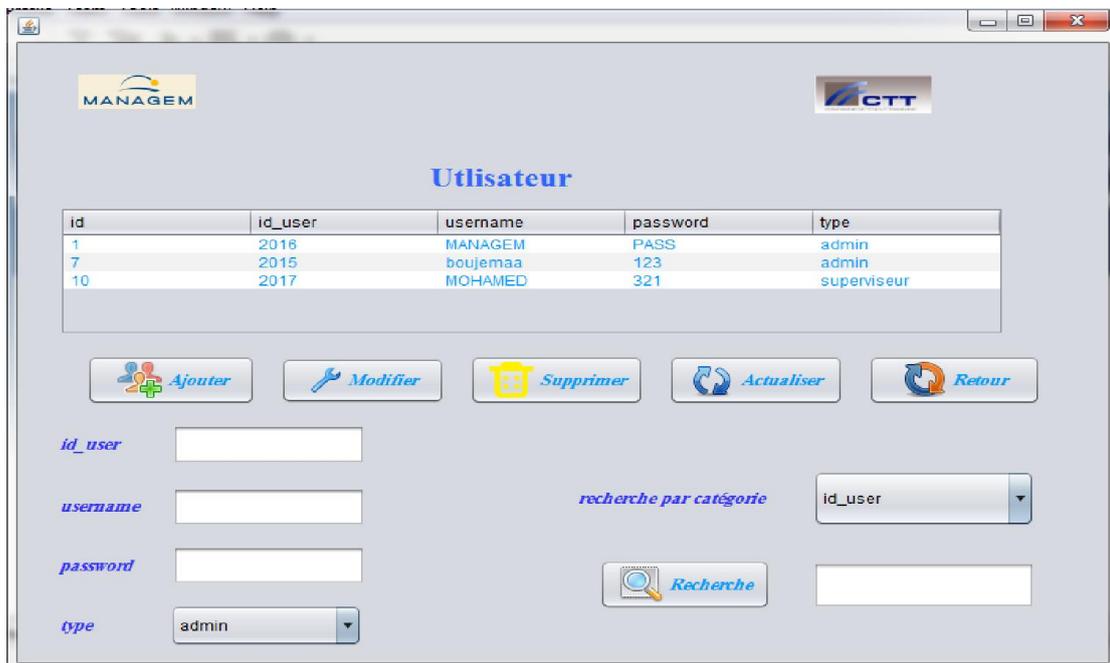


Figure 62: La fenêtre de gestion de l'utilisateur

La fenêtre de supervision de la salle

Cette fenêtre permet l'affichage des données et leurs évolutions en fonction du temps.

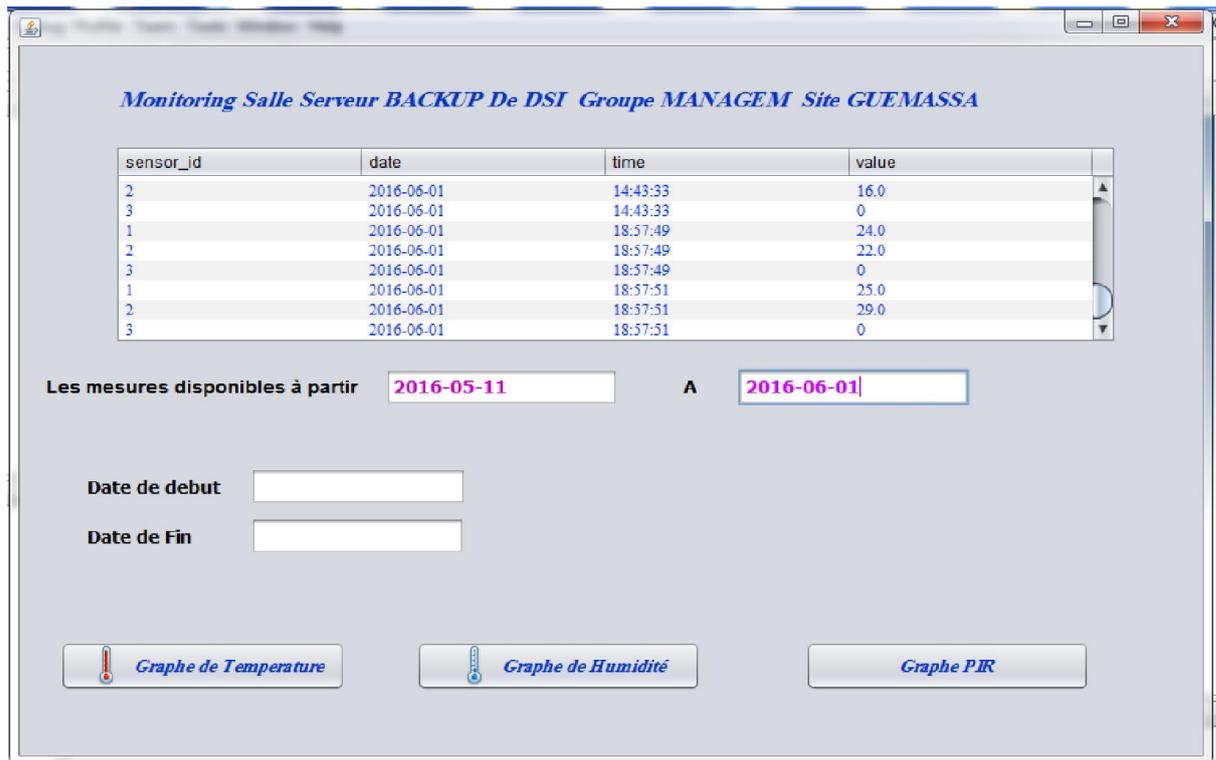


Figure 63: La fenêtre de Surveillance

Le tableau montre le contenu de la table **sensorData** sur laquelle les données captées par les capteurs (température, humidité, PIR) sont déjà stockées, en fonction de la date et l'heure.

La fenêtre permet aussi d'afficher la plage de date disponible dans la base de données, pour que l'utilisateur puisse choisir notamment laps de temps qu'il désire afficher.

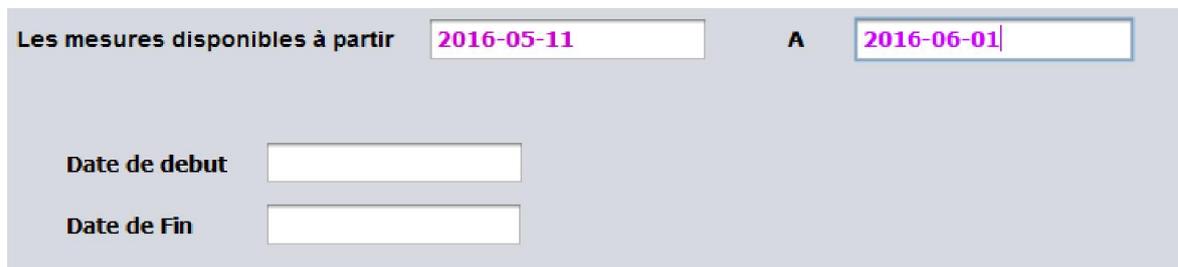


Figure 64: Plage de date disponible dans la base de données

Les trois boutons en-dessous permet chacun à dessiner le graphe de grandeur choisi, en fonction de laps de date sélectionné. Les graphes de visualisation est de la

forme suivante:

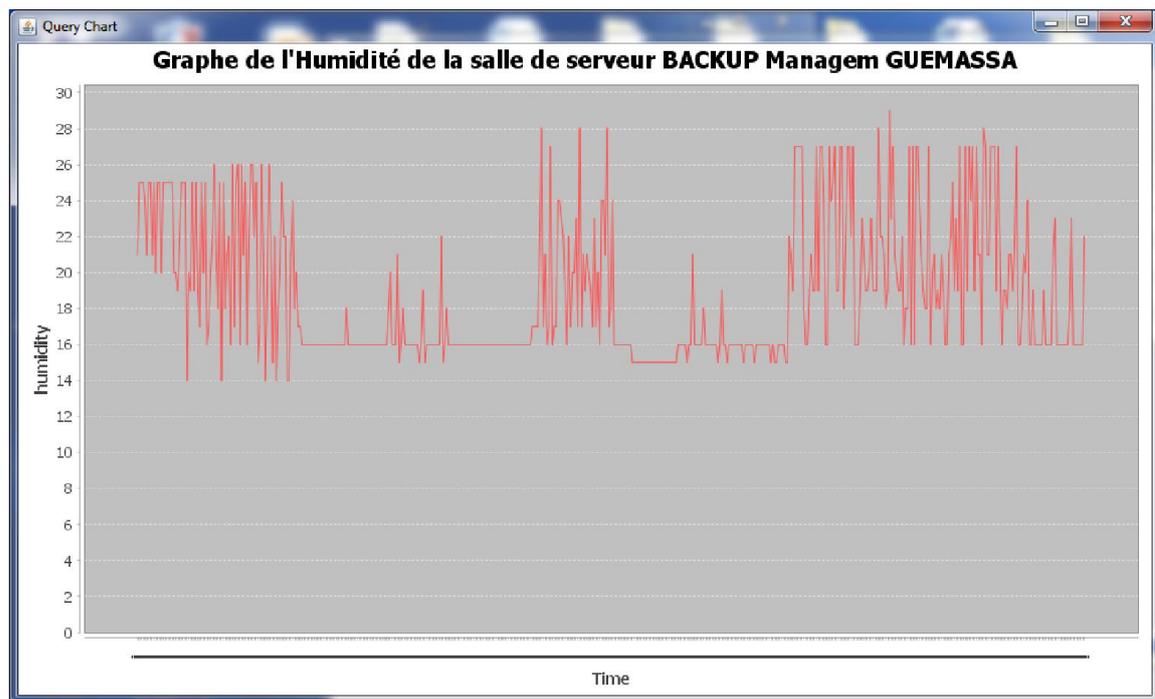


Figure 65: Graphe d'humidité salle serveur

Conclusion

Ce dernier chapitre décrit le fruit de notre travail, il présente les deux applications de supervision des données, déjà stockées dans la base de données, sous forme des graphes.

Conclusion Générale

Le thème de ce stage de fin d'étude consiste à la réalisation d'un système de surveillance de la salle serveurs Backup en utilisant la carte Raspberry Pi comme le cœur de notre système embarqué.

Tout au long de notre période de stage de 4 mois, nous avons réussi à atteindre les objectifs, décrit dans le cahier des charges ; collection et enregistrement des données captées (température, humidité et état de capteur PIR) dans une base de données MySQL, la visualisation sous la forme graphique des données captées et leur évolution dans le temps, cette visualisation est réalisée de deux manières, la première via une application Web et la deuxième via une interface graphique Java, et la détection d'intrusion à la salle en prenant des photos à la personne intrus.

Comme la compréhension et la maîtrise du domaine de l'électronique et des systèmes embarqués était notre premier objectif dans cette formation d'ingénieur, ce stage nous a été une occasion d'or pour avancer avec un grand pas vers cet objectif en prenant des nouvelles notions dans d'autres domaines et en appliquant nos acquis dans un travail concret et en affrontant plusieurs défis au cours de cette réalisation que ce soit technique ou organisationnel.

Bibliographie

[1] : Jerry Placide GODOUA « implémentation d'un réseau d'entreprise multi-sites », rapport de projet de fin d'étude 2011/2012 Faculté des Sciences, Meknès.

[2] : The Raspberry Pi Education Manual, Version 1.0 December 2012.

[3] : Apprenez à programmer en python, Vincent LE GOFF.

Webographie

<http://raspberrypi.org/> : le 17/02/2016

<http://dexterindustries.com/> : le 02/03/2016

<https://docs.python.org/> : le 25/02/2016

<http://apprendre-python.com/> : le 10/03/2016

<http://raspbian-france.fr/> : le 01/05/2016

<http://github.com/> : le 20/02/2016

<http://www.developpez.com/> : le 25/03/2016

Annexe

I. Installation du système d'exploitation

1. Choix de système d'exploitation

La plupart des systèmes qui fonctionnent sur Raspberry Pi sont des versions du système d'exploitation Linux. Parce que Linux est open source, les développeurs peuvent l'adopter pour des buts spécifiques. Dans le cas du Raspberry Pi, le matériel léger nécessite un système réduit et privé d'applications inutiles.

Raspbian est l'officiel distribution de système d'exploitation soutenu par la fondation du Raspberry et il existe plusieurs distributions compatibles avec Le Raspberry Pi 2 Modèle B que nous pouvant les télécharger depuis le site officiel <https://www.raspberrypi.org/downloads/>, et la figure suivante présente des exemples de ces distributions.

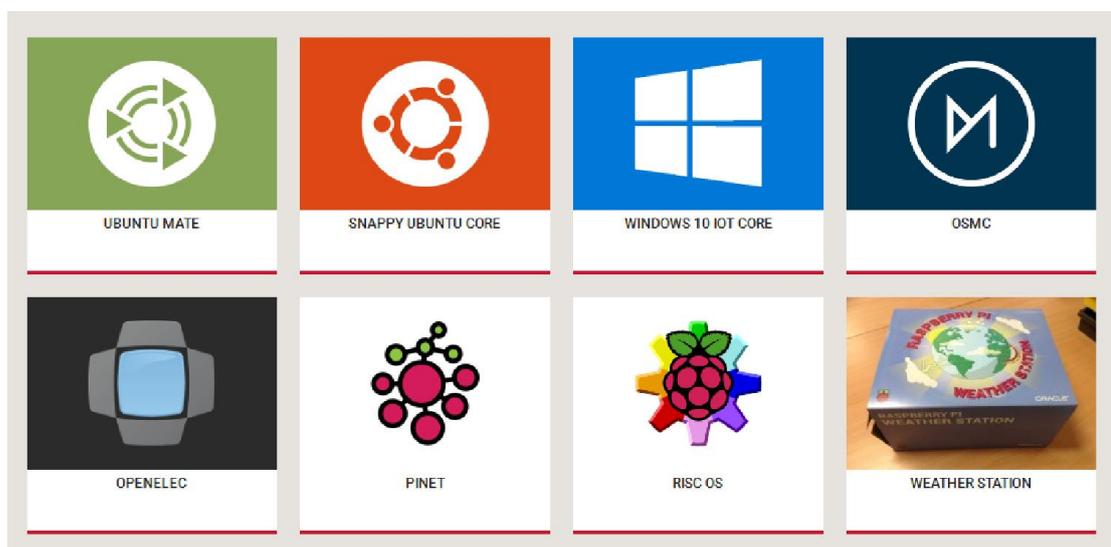


Figure 66: D'autres systèmes d'exploitation compatibles avec Pi 2 Modèle B

La distribution de système d'exploitation choisie c'est « Raspbian For Robots » fournis par la société « Dexter Industries », qui est une distribution Raspbian modifiée et qui contient des logiciels Grove Pi entièrement installés pour qu'on puisse utiliser le Raspberry Pi 2 avec la carte des capteurs Grove Pi (comme on va voir dans la suite de ce

rapport) sans qu'il y ait des problèmes de communication et de traitement de données.

2. Les étapes de l'installation

Comme on a montré précédemment que le Raspberry Pi 2 ne possède pas de disque dur qui va contenir le système d'exploitation, la solution optée par la fondation est d'utiliser la carte SD comme un disque dur ; dans notre cas (en utilisant le Raspberry Pi 2 Model B) la carte SD c'est une MicroSD. Donc pour installer le système d'exploitation au-dessus de la carte SD on va suivre les étapes suivantes :

2.1. Télécharger l'image Raspbian For Robots

Pour télécharger l'image Raspbian For Robots on utilise le site suivant : <https://sourceforge.net/projects/dexterindustriesraspbianflavor/> puis on clique sur **download**.

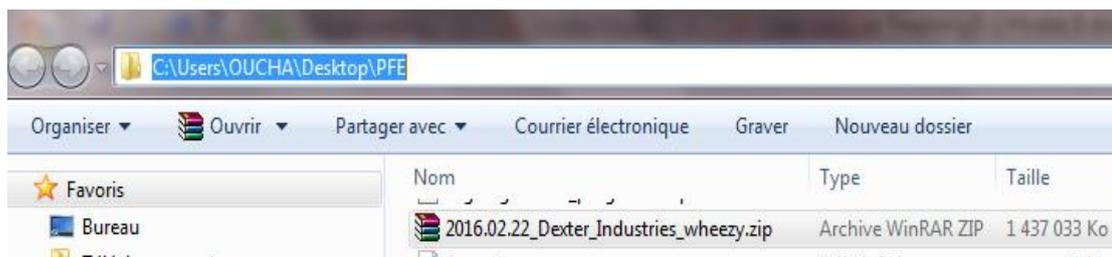


Figure 67: le fichier (.zip) de Raspbian For Robots

Après avoir téléchargé le fichier (.zip), on le décompresse pour obtenir le fichier d'image (.img) qu'on va l'écrire sur notre carte SD.

2.2. Ecrire l'image sur la carte SD

Avant de commencer l'écriture de l'image sur la carte SD il faut au début s'assurer que notre carte SD ayant au minimum 4 Go de taille, nous utilisons une carte de 16 Go qui est très satisfaisante.

Avec le fichier image de la distribution Raspbian For Robots, nous devons utiliser un outil d'écriture d'image pour l'installer sur la carte SD.

On connecte la carte microSD à l'ordinateur, et pour écrire l'image sur la carte SD on utilise le logiciel « **Win32DiskImager** », puis on choisit le fichier d'image (.img)

depuis son répertoire et le dispositif « Device » qu'est la carte SD.

Après ça on écrit l'image sur la carte SD en cliquant sur « Write », et on attend que l'écriture termine. À la fin de l'écriture on éjecte la carte SD et donc notre système d'exploitation **Raspbian For Robots** est prêt d'être utilisé sur la carte Raspberry Pi 2 Modèle B.

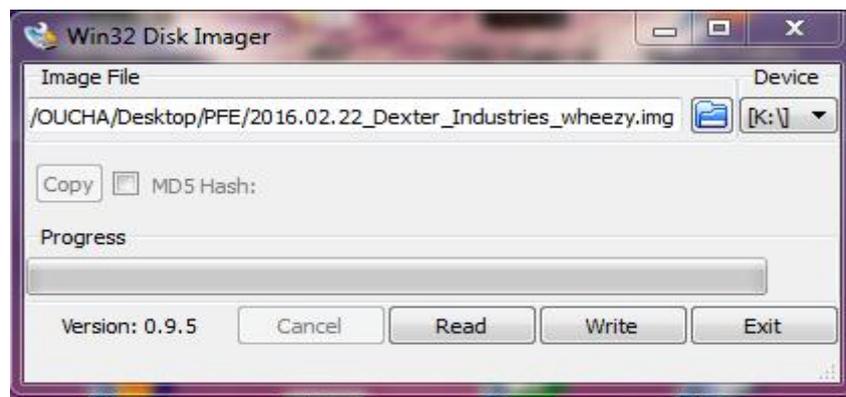


Figure 68: écriture de l'image Raspbian For Robots sur la carte SD

2.3. Installer Raspbian For Robots

Après l'insertion de la carte SD sur le Raspberry Pi, du clavier, de la souris et de l'écran maintenant on peut alimenter le Raspberry Pi qui va démarrer le système d'exploitation pour **la première fois** qui est un peu exceptionnel car ce démarrage peut-être un peu long, et à ce moment le Raspberry va installer le système Raspbian For Robots.

A la fin de démarrage, le système nous demande l'authentification par défaut l'identifiant est « pi » et le mot de passe est « robots1234 » (le clavier par défaut est en QWERTY) que l'on peut changer. Après l'authentification l'interface de configuration suivante apparaît:

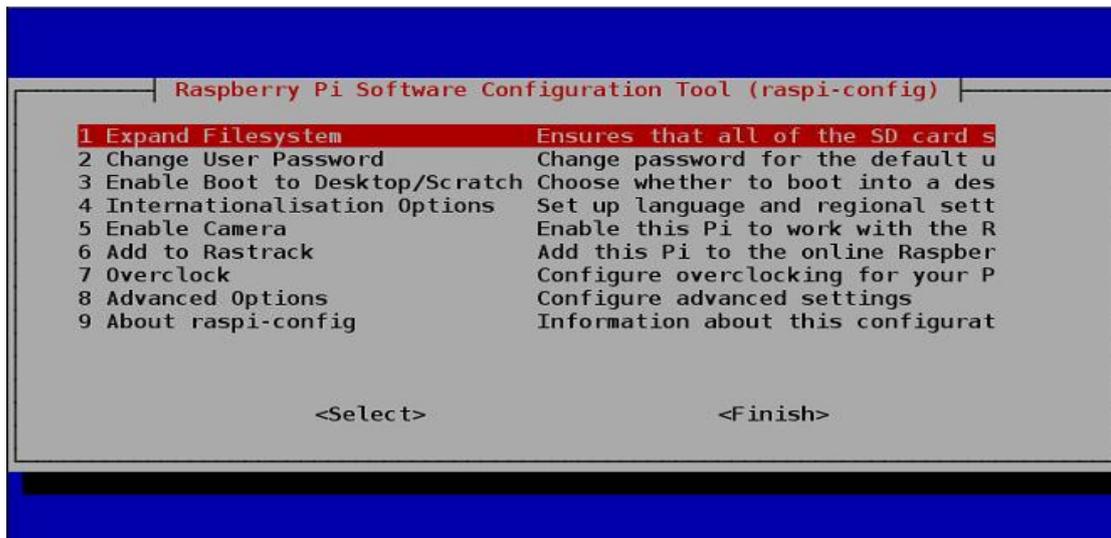


Figure 69: interface de configuration au premier démarrage

L'image a seulement pris environ 3,5 Go de notre carte SD et pour que le système d'exploitation utilise la totalité de la taille de la carte SD on doit obligatoirement faire cette configuration indispensable, pour cela on clique (avec le bouton de clavier « Entrée ») sur le premier choix « **Expand Filesystem** », la nouvelle fenêtre apparaît nous informe de la réussite de l'opération.

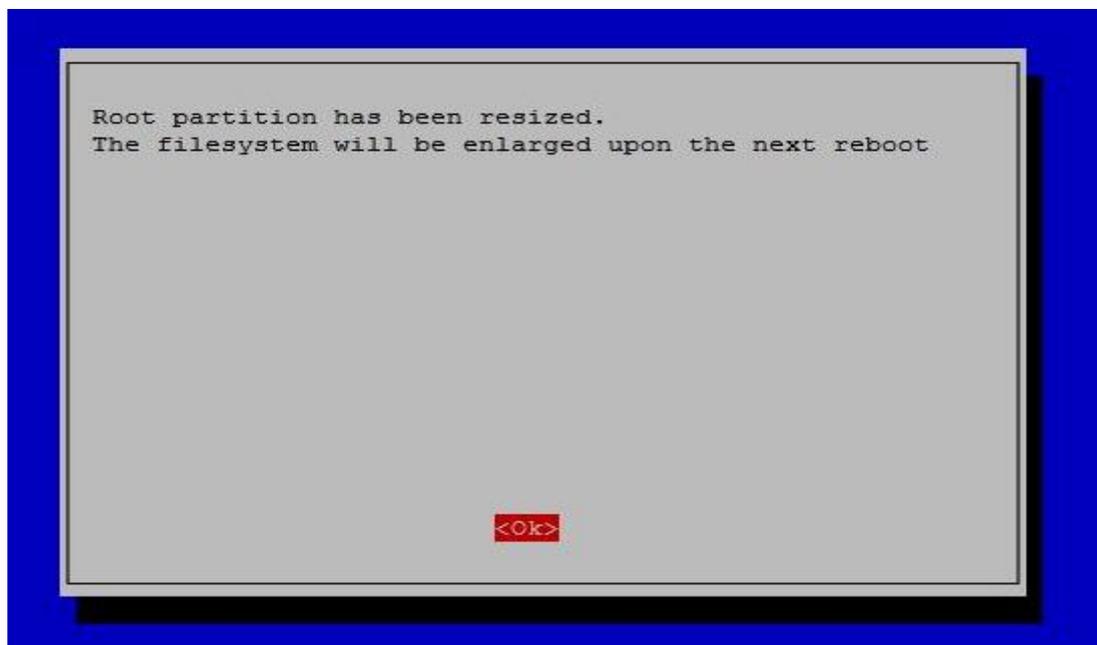


Figure 70: La réussite de l'opération élargissement des fichiers du système

La deuxième configuration que l'on peut faire, et qui est nécessaire pour la sécurisation de notre carte Raspberry, est de changer le mot de passe, pour cela on sélectionne le deuxième choix de l'interface de configuration « **Change User Password** » puis on clique « Entrée » une fenêtre nous informe que nous allons devoir entrer un nouveau mot de passe pour l'utilisateur « pi » en valide puis on entre notre nouveau mot de passe et sa vérification puis on valide les modifications.

La troisième configuration qu'on a fait c'est de passer le clavier en « AZERTY » car le clavier par défaut est en « QWERTY » et pour faire cette configuration on choisit le quatrième choix « **Internationalisation options** » en cliquant sur « Entrée » une nouvelle fenêtre apparaît on choisit « **Change Keyboard Layout** » puis sur la nouvelle fenêtre, on valide directement sans changer le type de clavier. Une autre fenêtre apparaît, on choisit « **Autre** ». Sur la nouvelle fenêtre, on choisit le clavier « **Français** », puis la disposition « **Par défaut** », « **Pas de touche « compose »** », et enfin « **Non** ».

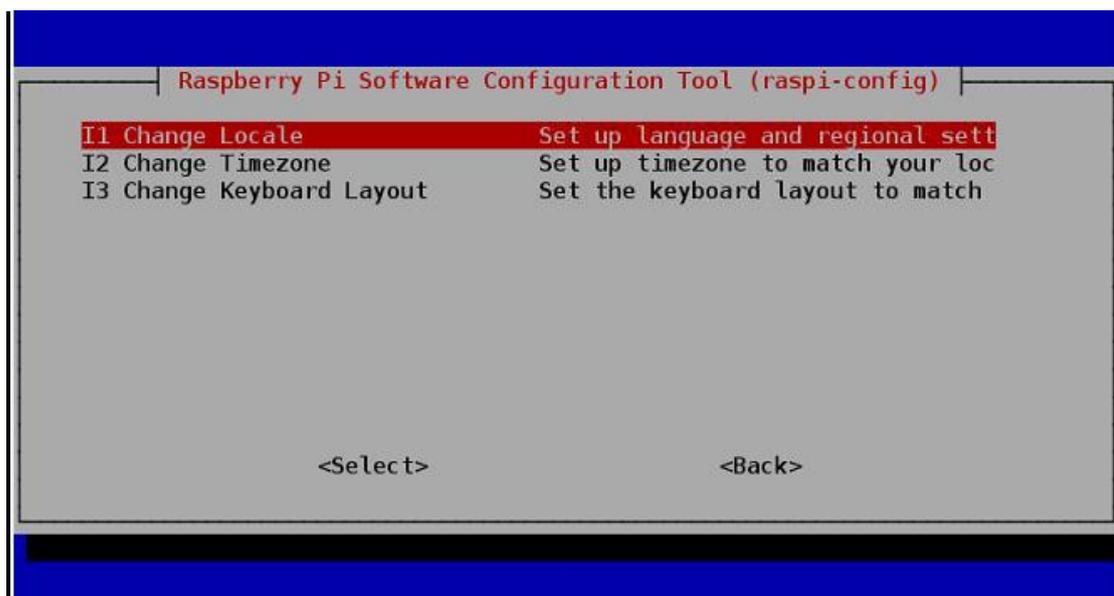


Figure 71: les options du choix «Internationalisation options »

Après la fin de la configuration et que nous sommes sur le menu principal de configuration on choisit « **Finish** », pour valider toutes les configurations faites.

Notre système d'exploitation maintenant est bien configuré et donc il ne reste

qu'à mètre à jours, pour cela on tape les trois commandes suivantes :

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo reboot
```

La dernière commande redémarre le Raspberry, au redémarrage l'interface graphique se lance automatiquement comme suivant :

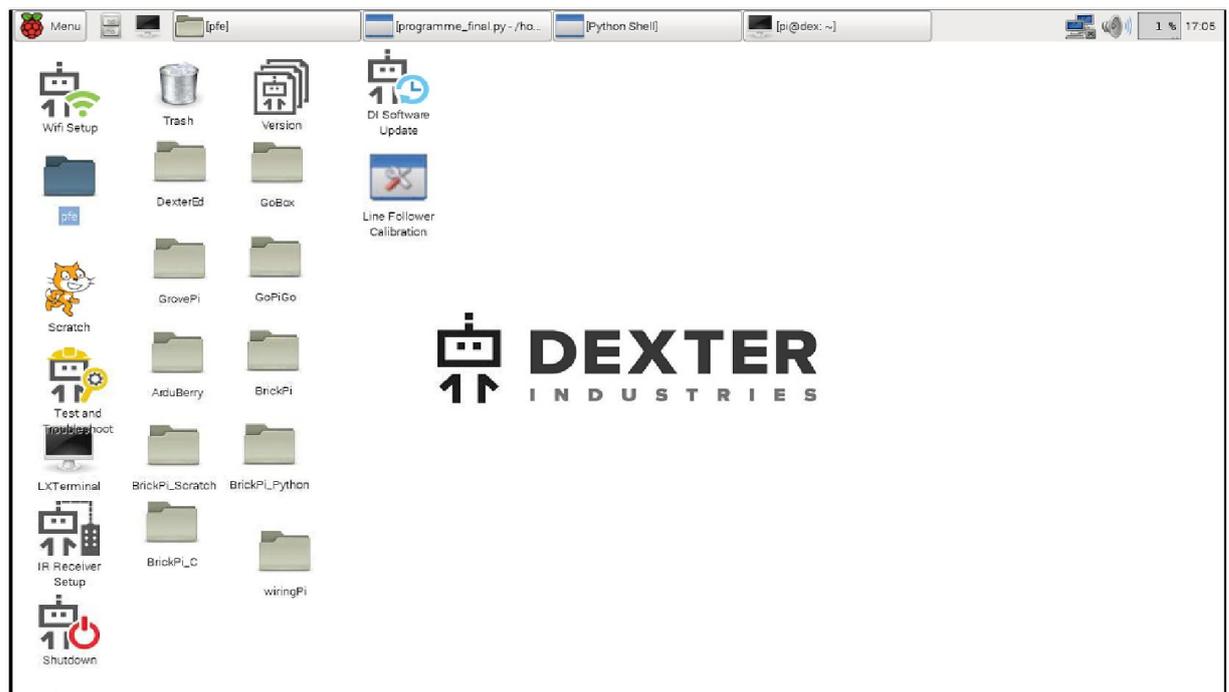


Figure 72: interface graphique du Raspbian For Robots

Alors le Raspbian For Robots est installé sur notre Raspberry Pi 2 et tout est bien mis à jour, il reste seulement de l'utiliser dans notre système de surveillance de salle de serveur.

II. Langage de programmation python

Python est un merveilleux langage de programmation. Moderne, portable, puissant, facile à apprendre.

Dans cette partie on va donner seulement l'essentiel de ce langage :

1. Introduction

Python est un langage interprété, ce qui peut vous faire gagner un temps considérable pendant le développement du programme car aucune compilation ni édition de liens n'est nécessaire.

Python permet d'écrire des programmes compacts et lisibles. Les programmes écrits en Python sont généralement beaucoup plus courts que l'équivalent en C, C++, ou Java, pour plusieurs raisons :

- les types de données de haut niveau vous permettent d'exprimer des opérations complexes en une seule instruction ;
- les instructions sont regroupées entre elles grâce à l'indentation, plutôt que par l'utilisation d'accolades ;
- aucune déclaration de variable ou d'argument n'est nécessaire.

2. Les différents types :

Le type des variables utilisées a une grande importance dans le code utilisé car leurs utilités et leurs manipulations sont bien différentes. Pour reconnaître le type d'une variable, il existe une fonction intégrée à Python, la fonction `type ()` [3].

❖ Les numériques

Un nombre diffère par l'espace mémoire alloué pour celui-ci par l'ordinateur.

✚ **Les entiers** : tous les nombres entiers en langage C correspondent au type **int**.

✚ **Les flottants** : Ce sont les nombres réels (dits à virgule flottante).

✚ **Les complexes** : De façon native, Python contient déjà l'écriture des complexes. La partie imaginaire s'écrit `j`. Ainsi, le nombre complexe $1+3j$ s'écrit `1+3j`.

❖ Les booléens

Les booléens ne peuvent prendre que 2 valeurs : **Vrai** (**True** attention à la majuscule) ou **Fausse** (**False** attention à la majuscule).

Exemple :

```
>>>1==2# 1 est-il égal à 2 ?
False
>>>1==1# 1 est-il égal à 1 ?
True
```

On peut encore définir un booléen en affectant directement une valeur booléenne à une variable.

```
>>>b=False
```

```
>>>type(b)
<type 'bool'>

>>>c=false# Attention à la majuscule
NameError: name 'false' is not defined
```

✚ Les comparaisons :

Une comparaison entre deux nombre conduit à un résultat binaire, donc à un booléen.

Les comparaisons codées en python peuvent se résumer dans le tableau suivant :

<i>test</i>	<i>maths</i>	<i>Vrai quand</i>
<code>x == y</code>	$x = y ?$	x est égal à y
<code>x != y</code>	$x \neq y ?$	x est différent de y
<code>x > y</code>	$x > y ?$	x est strictement supérieur à y
<code>x < y</code>	$x < y ?$	x est strictement inférieur à y
<code>x >= y</code>	$x \geq y ?$	x est supérieur ou égal à y
<code>x <= y</code>	$x \leq y ?$	x est inférieur ou égal à y

On peut aussi combiner deux inégalités :

<i>test</i>	<i>maths</i>	<i>Vrai quand</i>	<i>Remarque</i>
<code>a <= x and x <= b</code>	$a \leq y \leq b ?$	x est compris entre a et b inclus	lisible
<code>a <= x <= b</code>	$a \leq y \leq b ?$	x est compris entre a et b inclus	syntaxe propre à Python

❖ Les chaînes de caractères

Une chaîne de caractères (**string** ou **str** en abrégé) est une suite de données de lettres ou de chiffres (ou les deux) encadrée par des guillemets ou des apostrophes.

La fonction **print()** permet d'afficher la chaîne de caractères en question.

Exemple :

```
>>>test1='essai de chaine de caractères'
>>>print(test1)
essai de chaine de caractères
>>>type (test1)
```

```
<type 'str'>
```

❖ Les structures (boucles et instructions conditionnels) :

❖ Les boucles :

➤ Les boucles for

Les boucles **for** sont appropriées lorsqu'une opération doit être réalisée un nombre de fois donnée. Le bloc indenté est exécuté pour toutes les valeurs de la ligne d'instruction **for**.

```
>>>for i in range(6):# pour i allant de 0 à 5
... print(i)# Afficher i
...
0
1
2
3
4
5
```

Remarque :

- ✓ L'oubli des deux points ":" après la première ligne est une erreur courante.
- ✓ L'oubli de l'indentation de la boucle à répéter est aussi une erreur courante.

➤ Les boucles while

Les boucles **while** seront répétées **tant que** la condition est vraie. Dans l'exemple suivant, on affiche la valeur de *i* tant que *i* est inférieur ou égal à 5.

```
>>>i=1
>>>while i<=5:
... print(i)
... i=i+1
...
1
2
3
4
5
```

- ✓ Attention une condition **jamais réalisée** conduira à une **boucle sans**

fin.

❖ Les instructions conditionnelles :

❖ Les tests (Instruction if)

Une instruction **if** n'est exécutée que si le test est vrai. Toutes les lignes indentées correspondent au bloc exécuté.

```
>>>for I in range(10):
... if 4<=i<=6:
... print(i)
... print(i**2)
4
16
5
25
6
36
```

❖ Création d'un répertoire

Pour la création des dossiers en python on va utiliser le **module** `os` qui est une bibliothèque dédié aux besoins de gestion de fichiers et de dossiers.

Pour python un chemin (ou **path**), c'est une chaîne de caractères donc pour créer un répertoire ayant pour nom la date de la détection on utilise les fonctions suivantes :

- ❖ `date=time.strftime('%d-%m-%Y',time.localtime())` : pour retourner la date de la détection.
- ❖ `rep="/home/pi/Desktop/pfe/"+date` : création du chemin du répertoire.
- ❖ `os.path.exists(rep)` : Test si un chemin existe.
- ❖ `os.mkdir(rep)` : création du répertoire nommé par **date** situé à `"/home/pi/Desktop/pfe/»`.
- ❖ `assert os.path.isdir(rep)` : confirmation que le chemin `"/home/pi/Desktop/pfe/date"` est un dossier.

❖ Ecriture dans un fichier

Un fichier stocke des informations sur un support physique (disque dur, clé USB, CD, DVD, carte mémoire SD...).

Ouvrir un fichier consiste à le charger dans la mémoire vive (RAM) de

l'ordinateur (c'est une mémoire volatile : elle s'efface quand on éteint l'ordinateur).

Enregistrer un fichier consiste à l'écrire sur un support physique de stockage (l'information est alors conservée de manière permanente).

L'écriture dans un fichier se fait avec la fonction **open()** en mode écriture :

```
NomFichier = 'test.txt'
# création et ouverture du fichier test.txt en mode write 'w' (écriture)
# si le fichier test.txt existe déjà, il est écrasé
Fichier = open(NomFichier,'w') # instanciation de l'objet Fichier de la classe file
# écriture dans le fichier avec la méthode write()
Fichier.write('Bonjour à tous !')
# fermeture du fichier avec la méthode close()
Fichier.close()
```

❖ Stocker des données dans une base de données MySQL

Pour stocker les données dans une base de données (dans notre cas DB_MANAGEM) en python on va utiliser la bibliothèque MySQLdb pour cela on suit les étapes suivantes :

- ❖ `db = MySQLdb.connect(host = "localhost", user = "USER_DB", passwd = "****", db = "DB_MANAGEM")` : connection à la base DB_MANAGEM.
- ❖ `cur = db.cursor()` : utilisation du curseur pour se déplacer dans la base.
- ❖ `cur.execute("""INSERT INTO sensorData (sensor_id,date,time,value) VALUES (%s, %s, %s, %s)""",(1,time.strftime("%Y-%m-%d %H:%M:%S"),t))` : insertion de la données dans la table **sensorData** déjà créée.