



Rapport de Projet de fin d'Etudes

Conception et Développement d'un Système de Supervision d'un Réseau électrique moyenne tension (SMART RTU)

Présenté et soutenu par :

Haj Slimen Rayen

En vue de l'obtention de

Licence Nationale en Électronique Électrotechnique et Automatique
Spécialité : (systèmes électriques)

Sous la Direction de :

Gamra Imed & Zammel Sami

Soutenu le 08/06/2023

Devant le jury composé de :

Président : Zahra Basma

Rapporteur : Elhadj Abdalah Saber

Membres : Zammel Sami

Dédicaces

Du profond de mon cœur, je dédie ce travail :

À mes chers parents

Que ce travail soit l'expression de ma reconnaissance pour vos sacrifices consentis, votre soutien que vous n'avez cessé de prodiguer. Vous avez tout fait pour mon bonheur et ma réussite.

Que dieu vous préserve en bonne santé et vous accorde une longue vie.

 **Rayen**

Avant-propos

Ce présent rapport détaille le travail que nous avons réalisé dans le cadre d'un stage de fin d'études à **Société Tunisienne de l'Electricité et du Gaz**, pour l'obtention du Licence Nationale en Electronique Electrotechnique et Automatique spécialité systèmes électriques.

Au terme de ce travail, nous tenons à exprimer notre profonde reconnaissance à tous ceux dont nous avons eu l'honneur de solliciter la compétence et l'expérience

Remerciements

Nous tenons à exprimer mes vifs remerciements envers **Monsieur Imed Gamra**, chef de service informatique en bureau centrale de conduite STEG KSAR Gafsa. Un grand merci pour son accueil chaleureux, ainsi que pour sa patience, sa disponibilité et ses précieux conseils.

Nous nous ferons un agréable devoir de remercier notre encadrant **Monsieur Sami Zemmel**, enseignant à ISSAT GAFSA pour nous avoir fait bénéficier de son expérience et de ses précieux conseils.

Nous tenons à remercier sincèrement les membres du jury qui nous font le grand honneur d'évaluer ce travail.

Résumé

Titre du projet :

Conception et Développement d'un système de supervision d'un réseau Électrique moyenne tension.

Description :

Ce présent rapport détaille le travail que nous avons réalisé dans le cadre d'un stage de fin d'études à **L'Institut Supérieur des Sciences Appliquées & Technologie de Gafsa**, pour l'obtention du Licence Nationale en Électronique Électrotechnique et Automatique Spécialité (systèmes électriques)

Ce projet consiste à mettre en place une Unité Terminale Télécommandée intelligente à partir des trois coffrets classiques présents au niveau de l'Interrupteur Aérien Télécommandé et une carte programmable. L'envoi et réception des données provenant de ce système se fera à travers une application mobile ou une interface web. Les informations reçus représentent l'état du réseau et plus particulièrement le point télécommandé et les données envoyées représentent les commandes envoyées par l'agent vers le système intelligent.

Mots Clés :

Système de contrôle, SCADA, RTU, IAT, Arduino, Android, commande sans fil, ...

Table des matières

Dédicaces	i
Avant-propos	ii
Remerciements	iii
Résumé	iv
Liste des figures	vii
Liste des tableaux	viii
Liste des Abréviations	ix
Introduction générale.....	1
Chapitre1	4
Etude préalable	4
1. Introduction	4
2. Présentation du projet.....	4
2.1. Présentation de l'organisme d'accueil	4
2.2. Objectifs du projet.....	5
3. Etat de l'art	6
3.1. Présentation des systèmes SCADA.....	6
3.1.1. Architecture d'un système SCADA	6
3.2. Présentation des cartes programmables	11
3.2.1. Les cartes Arduino	12
3.2.2. Les cartes Raspberry Pi	16
1. Conclusion.....	19
Chapitre2	21
Analyse et spécification des besoins.....	21
1. Introduction	21
2. Cahier des charges.....	21
3. Méthodologie de travail.....	22
4. Identification des besoins	25
4.1. Enoncé des besoins fonctionnels.....	25
4.2. Enoncé des besoins non fonctionnels.....	26
5. Conclusion.....	27
Chapitre3	29

Conception	29
1. Introduction	29
2. Présentation de la solution.....	29
3. Côté hardware.....	31
4. Côte software.....	33
2.2. Modèle d'analyse de l'interface utilisateur	34
2.3. Diagramme des classes	35
2.4. Diagrammes des séquences.....	36
2.5. Protocole d'échange	38
3. Conclusion.....	38
Chapitre4 : Réalisation	40
1. Introduction	40
2. Mise en place de la RTU	40
2.1. Réalisation du coffret	40
2.2. Configuration de la base de données.....	42
3. Intégration de la carte programmable.....	46
3.1. Choix de la carte	46
3.2. Mise en place de la carte	47
4. Développement de l'application mobile.....	48
3.1. Etapes de réalisation.....	48
3.2. Présentation des interfaces	49
5. Développement de l'application web	51
5.1. Etapes de réalisation.....	51
5.2. Présentation des interfaces	52
6. Conclusion.....	54
Conclusion générale	55
Webographie	56
Annexes	57
Annexe 1 : Liste des composants utilisés.....	57
Annexe 2 : SAITEL 100.....	68
Annexe 3 : Configuration et chargement de la base des données	70

Liste des figures

Figure 1. Architecture d'un système SCADA	7
Figure 2. Coffret de commande.....	7
Figure 3. Atelier d'énergie.....	8
Figure 4. <i>Connexion des chargeurs d'alimentation</i>	8
Figure 5. RTU Saitel-100.....	9
Figure 6. Procédure d'échange de données RTU-SCADA	10
Figure 7. Exemple des cartes Arduino.....	12
Figure 8. Logiciel Arduino.....	15
Figure 9. La famille des Raspberry Pi.....	16
Figure 10. Cycle de vie en V.....	23
Figure 11. Schéma fonctionnel du système proposé.....	29
Figure 12. Descriptif fonctionnel du programme de contrôle	30
Figure 13. Vue globale sur les cas d'utilisation de l'interface utilisateur.....	31
Figure 14. Conception logiciel du programme embarqué au niveau de la carte programmable (diagramme de déploiement).....	32
Diagramme d'activité du programme Arduino.....	33
Figure 16. Diagramme de package.....	35
Figure 17. Diagramme des classes.....	36
Figure 18. Diagramme de séquence de cas d'utilisation « <i>consulter l'état d'une RTU</i> »	37
Figure 19. Diagramme de séquence de cas d'utilisation « <i>Envoyer une commande à la RTU</i> »	37
Figure 20. Dessin de nouveau schéma de câblage	41
Figure 22. Coffret mixte	42
Figure 23. <i>Schéma de câblage relais SCHRAK</i>	44
Figure 24. <i>Reliage IAT type ABB Après modification</i>	44
Figure 25. Fichiers de base de données modifiés manuellement (1)	45
Figure 26. Fichiers de base de données modifiés manuellement (2)	46
Figure 28. Schéma de montage de la carte Arduino	48

Figure 29. Environnement de développement Android Studio	49
Figure 30. Interface d'authentification.....	50
Figure 31. Menu principal de l'application.....	50
Figure 32. Page d'accueil de l'application web	52
Figure 33. Interface principale du site.....	53
Figure 34. Interface « fiche numérique ».....	53
Figure 35. Interface liste des alarmes.....	54

Liste des tableaux

Tableau 1. Tableau comparatif des différentes cartes Arduino [5]	14
Tableau 2. Protocoles de communication.....	38
Tableau 3. <i>Les matériels utilisés en ancien câblage et nouveau câblage</i>	41
Tableau 4. Arduino VS Raspberry Pi.....	47

Liste des Abréviations

BCC : Bureau Central de Conduite
DMS:**D**istribution**M**anagement**S**ystem
CSS : **C**ascading **S**tyl**S**heets **HTA** :
Haute **T**ension **A**
HTML: **H**yper**T**ext**M**arkup **L**anguage
IAT: **I**nterrupteur**A**erien**T**élécommandé
IDE : **I**ntegrated **D**evelopment**E**nvironment
MT: **M**oyenne **T**ension
PC : **P**oste **C**oupure
PHP: **H**ypertext**P**reprocessor
PS : **P**oste **S**ource
RTU : **R**emote **T**erminal **U**nit
SCADA : **S**uperviser **C**ontrol **A**nd **D**ata **A**cquisition
SGBD : **S**ystème de **G**estion de **B**ase de **D**onnées
SPI : **S**erial **P**eripheral **I**nterface
STEG : **S**ociété**T**unisienned'**E**lectricitéet du**G**az
UML : **U**nified **M**odeling **L**anguage
VCC : **C**ommon **C**ollector **V**oltage

Introduction générale

Il est difficile d'imaginer le monde d'aujourd'hui sans électricité. Les applications de l'électricité sont en effet toujours plus nombreuses, accompagnant les nouvelles inventions et les avancées technologiques. En conséquence, la production et la consommation d'électricité augmentent chaque année, et elle dépend nécessairement de la disponibilité et du coût des combustibles. L'électricité est une énergie non stockable, elle doit être produite, transportée et distribuée pour être consommée. Il nous faut garantir l'équation de production égale à la consommation et par conséquent veiller en temps réel à la continuité et la qualité de fourniture de cette énergie. L'énergie électrique est produite dans des usines génératrices appelées 'Centrales électriques'. Une centrale comporte un ou plusieurs groupes constitués chacun d'un alternateur entraîné par une machine motrice (moteur Diesel, turbine à vapeur, turbine hydraulique.). Le transport de l'énergie produite est assuré par un réseau de transport qui se présente sous la forme d'un maillage de lignes à haute tension couvrant l'ensemble de pays et dont les nœuds sont constitués par des postes. Le transport de l'électricité se fait tout d'abord en augmentant le voltage de l'énergie à la sortie des centrales son voltage de 15 000V à une très haute tension de 225 000V ou 400 000V à l'aide de transformateurs élévateurs de tension. Cette opération est nécessaire car le transport de l'énergie électrique s'accompagne de pertes d'énergie (de l'ordre de 5%). Après, on trouve des postes de transformation de l'énergie aux bords des villes, ces postes sont construits pour baisser la tension (Haute tension) et la rendre moyenne tension MT (HTA). Enfin, on trouve des lignes électriques (HTA) qui sortent de ces postes et qui se dispersent dans les villes, les villages, les zones rurales, et les zones industrielles.

Les réseaux de transport et de distribution de l'électricité, qui représentent un point clef de distribution de la gestion des réseaux électriques, comme ils deviennent de plus en plus complexes à gérer du fait de leurs architectures, du faible nombre de données, des agressions et des perturbations qui peuvent s'y produire. Les lignes de distribution de l'énergie peuvent être aériens ou souterrain, ils sont soumis à plusieurs défauts ce qui représente un obstacle de continuité de fournir de l'électricité aux consommateurs. Puisque l'énergie

Électrique représente un facteur capital à tout développement dans tous les domaines soit économiques, soit social ou culturel, les sociétés de production et de transport de l'électricité ont essayé d'adapter des nouvelles technologies dans le but de mettre à jour et d'optimiser ses techniques d'exploitations du réseau électrique. Les premiers pas vers l'automatisation des réseaux de distribution étaient réalisés avec des systèmes de type SCADA (*Supervisory Control And Data Acquisition*). Ils permettent la conduite des postes électriques importants et la télé conduite de quelques organes de coupure (sectionneurs, interrupteurs). Les fonctions SCADA concernent aussi le traitement des signalisations et des alarmes provenant du réseau de distribution, l'émission de diagnostic vers l'opérateur, l'exécution des informations pour les applications ultérieures.

La rentabilité de ces systèmes de télé conduite est totalement liée à la disponibilité et l'efficacité des organes télécommandés. C'est dans ce sens que s'inscrit notre projet de fin d'études qui vise principalement à l'optimisation des fonctions de système SCADA par le fait de rendre une unité terminale télécommandé plus intelligente et garantir une traçabilité en temps réel des incidents. Il s'agit de concevoir un coffret RTU à partir des trois coffrets présents au niveau de l'interrupteur aérien. Et d'intégrer dedans une carte programmable permettant à l'utilisateur l'interaction avec la RTU à travers une application mobile ou une interface web.

Pour la mise en place d'un tel système, nous allons commencer par une étude préalable de l'existant. Par la suite, nous allons spécifier les besoins par l'analyse du cahier des charges et l'identification des besoins fonctionnels et des besoins non fonctionnels. Puis, nous allons définir l'approche proposée et la conception du système et par suite nous allons détailler les différentes étapes de réalisation.

Chapitre 1

ETUDE PREALABLE



Chapitre1

Etude préalable

1. Introduction

Ce chapitre est décomposé en trois parties. La première partie sera consacrée à la présentation de la société d'accueil et les objectifs principaux du projet. La seconde partie comprendra une description aussi complète que possible du système SCADA. La troisième partie sera consacrée à la présentation des cartes de contrôles à base des microcontrôleurs.

2. Présentation du projet

2.1. Présentation de l'organisme d'accueil

Pour garantir la continuité de service, la STEG a essayé de se servir des nouvelles technologies pour optimiser l'exploitation de son réseau électrique. Dans ce sens, elle a introduit pour la télé conduite du réseau moyenne tension depuis février 1989 par la mise en service d'un premier Bureau Central de Conduite de Tunis ensuite ce projet à évaluer pour toucher toutes les régions moyennant six BCC et permettre de contrôler tout le réseau moyenne tension (HTA).

En fait, le système de contrôle pour les 6 bureaux centraux de conduite des réseaux de distribution de la STEG, est défini comme un système d'exploitation, de contrôle et de supervision de toute l'information (acquisition et traitement) en provenance des systèmes et sous-systèmes (stations distantes, etc.) dont il est responsable.

Pour développer toutes ces fonctions, les BCC disposeront de tous les éléments hardware et software permettant l'échange de données en temps réel entre tous les éléments constituant l'architecture globale du système.

Les éléments hardware permettront le support et la maintenance aussi bien de l'information (temps réel et historique) que de l'exécution et la gestion de programmes, SCADA et DMS, selon le BCC. Les éléments software procureront l'acquisition, le traitement, la gestion, le stockage et la transmission de toute l'information, aussi bien en temps réel qu'historique, tels que les protocoles de communications, la gestion des canaux de communications, la gestion de l'interface homme-machine, etc.

La BCC a pour objectifs de ;

- Diminuer les durées de coupure d'électricité,
- Réduire l'énergie non distribuée (1% d'économie seulement représente énormément lorsque on observe la flambée des prix des combustibles (pétrole et gaz d'aujourd'hui),
- Réduire les pertes sur le réseau,
- Assurer la sécurité des agents et des équipements en coordonnant entre les différents intervenants du réseau,
- Améliorer et optimiser l'exploitation du réseau compte tenu la disponibilité des données temps réel tel que (tension – courant –puissance et état de position des appareils de coupure),
- Disposer d'une base de données complète sur le réseau, ce qui est primordial pour la planification à cours et moyenne terme pour l'assainissement et l'extension du réseau,
- Réduire les coûts d'exploitation en limitant les déplacements des agents et des équipements inutilement sur le terrain.

2.2. Objectifs du projet

Suite à une étude assez complète des différentes fonctionnalités qu'assure la BCC, nous avons pu dégager un certain nombre de problèmes qui empêche la réalisation de ses objectifs. D'ailleurs, la BCC assure la télécommande des équipements par l'intermédiaire d'une unité terminale télécommandée (RTU) menu d'un système de communication radio. Les principaux équipements sont les postes sources (PSs), les postes de coupures (PCs) et les interrupteurs aériens télécommandés (IAT). Nous allons détaillés le fonctionnement de ces équipements dans la deuxième partie de ce chapitre.

En fait, nous avons identifiées problèmes communs pour les RTU au niveau des IAT et des PCs. Et nous avons fixé comme objectifs pour notre projet de fin d'étude de répondre à ces problèmes à savoirs :

- Manque des pièces de rechanges.
- Encombrement au niveau des IAT et des PCs à cause de l'existence de plusieurs composantes dépendantes mais répartis sur trois coffrets.

- Interruption de communication parfois avec les points télécommandés suite à un blocage au niveau des composants ce qui nécessite le déplacement d'un agent sur terrain pour assurer une remise à zéro manuel de RTU.
- En cas d'un incident au niveau de réseau HTA, un passage de courant de défaut au détecté par RTU, déclenche un indicateur visuel qui reste active pendant une durée minimale de trois heures. Ce qui provoque le déchargement des batteries et diminue ainsi leurs durées de vie.

3. Etat de l'art

3.1. Présentation des systèmes SCADA

Le système de contrôle et d'acquisition de données électriques (SCADA) de Schneider Electric est le système SCADA/DMS utilisé au sein de la STEG. Ce système nommé OASyS est un système de surveillance et contrôle ayant les fonctions principales suivantes :

- Acquisition de données depuis des dispositifs sur le terrain
- Contrôle de ces dispositifs sur le terrain
- Affichage et traitement de situations d'alarme
- Automatisation des rapports
- Présentation des données manipulées par le système par l'intermédiaire de fenêtres graphiques
- Traitement de données historiques.

3.1.1. Architecture d'un système SCADA

L'architecture de système SCADA de la BCC est donnée par la figure suivante ;

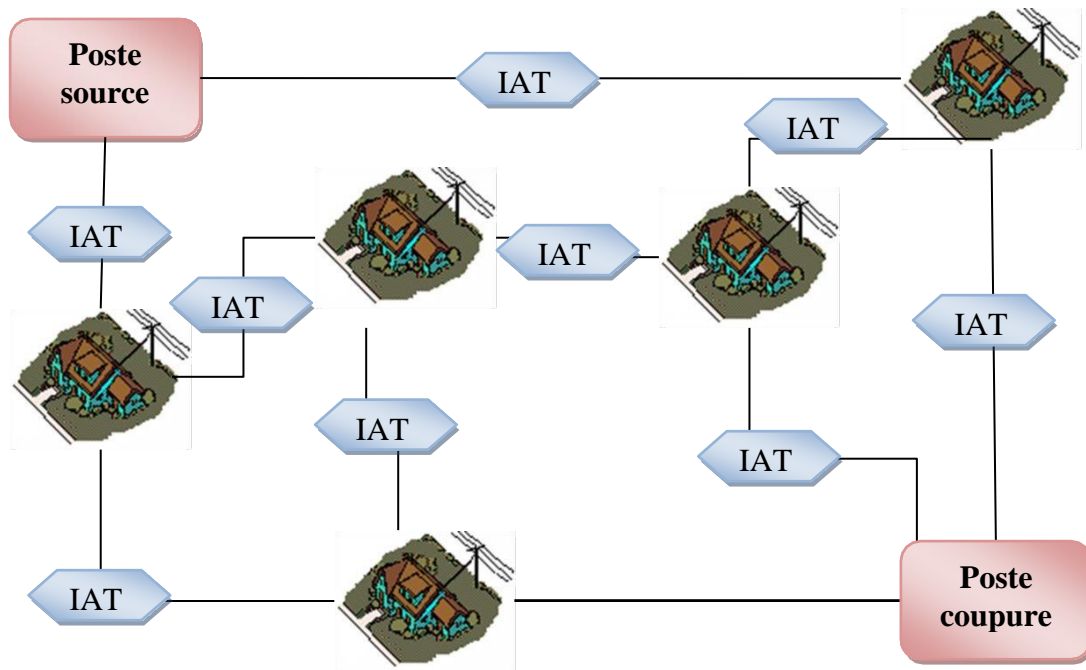


Figure 1. Architecture d'un système SCADA

Les IAT (interrupteur aérien télécommandé) relient les zones de consommation, les postes sources et les postes coupures. Un interrupteur aérien télécommandé est composé de :

3.1.1.1. Coffret de commande

C'est une armoire contenant un ensemble des composants électrique jouent le rôle de contrôle de bloc interrupteur et acquisition des données sur le réseau.



Figure 2. Coffret de commande

3.1.1.2. Atelier d'énergie

Comme le montre figure suivante, il s'agit d'une série qui comprend 3 modèles de chargeur d'alimentation qui, en présence de la tension du secteur, fournissent une tension stabilisée tout en procédant à la charge contrôlée de la batterie.

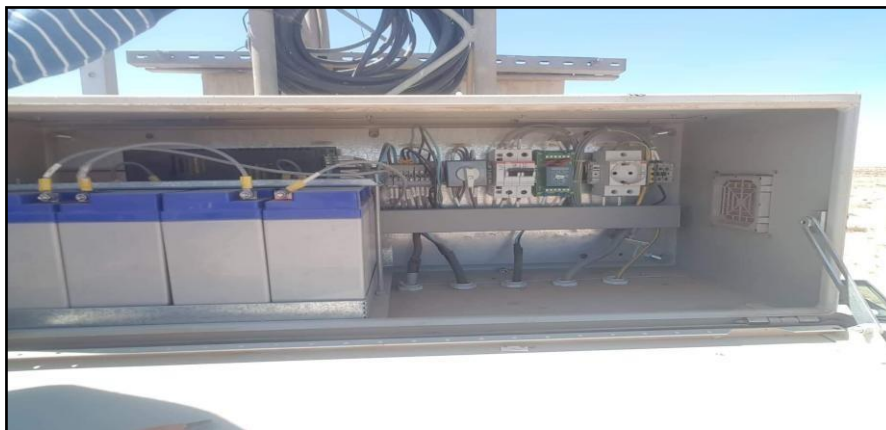


Figure 3. Atelier d'énergie

Cet appareil est constitué par une source d'alimentation commutée et un circuit limiteur du courant de charge qui fournit une charge de batterie à tension constante avec un courant limité. Il comprend également un circuit d'alarmes qui agissent indépendamment aussi bien à cause d'une panne du réseau ou de la source électrique que d'une batterie faiblement chargée. Les sorties des alarmes sont les contacts commutés et libres de potentiel des relais correspondants.

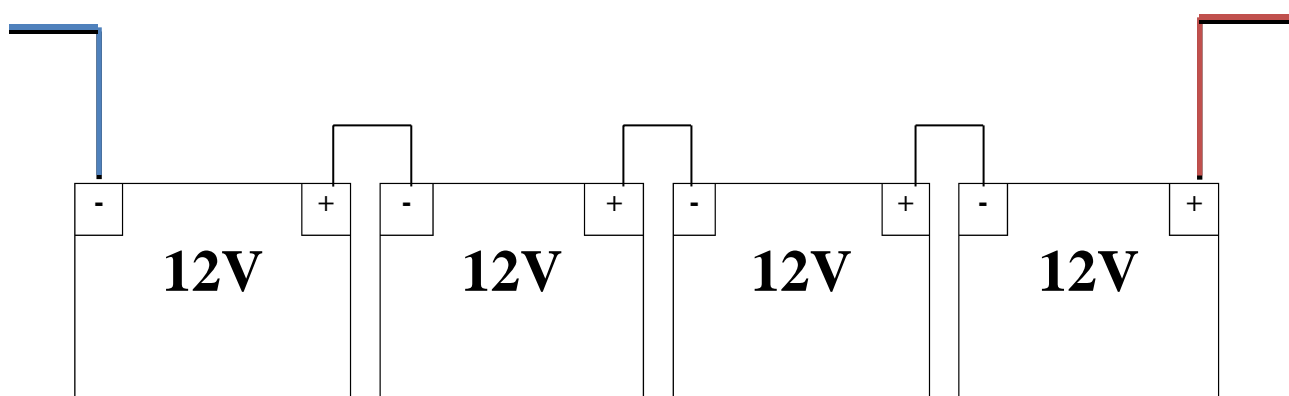


Figure 4. Connexion des chargeurs d'alimentation

3.1.1.3. Unité terminal télécommandé (RTU)

La RTU Saitel-100 est composée des blocs fonctionnels suivants :

- **Unité de commande à distance** : Module d'acquisition et de communications basé sur un microprocesseur HC11 de 8 bits.
- **Signaux d'entrée et de sortie** : La capacité en signaux d'entrée et sortie bruts est de 16 entrées numériques, 4 sorties numériques et 2 mesures analogiques bipolaires ou 4 unipolaires.
- **Communications** : La RTU Saitel-100 dispose de 4 canaux série asynchrones. Deux canaux sont pour les communications avec les couches supérieures et les deux autres pour les communications avec les niveaux inférieurs. Les signaux sont RS-232, y compris ceux de commande de modem. Elle possède en plus un canal série asynchrone par signaux RS-232, utilisé pour la maintenance locale. Le protocole utilisé est IEC 870-101 esclave.
- **Alimentation en énergie** : L'alimentation électrique se fait par courant continu +48 V, +20/-10% (Versions en 24 VCC et 12 VCC). La consommation maximum est de 5 W et les éléments de puissance nécessaires pour fournir les tensions et les isollements des différents blocs la constituant sont disponibles.



Figure 5. RTU Saitel-100

L'unité terminale de télé conduite sera intégrée dans un système pouvant comprendre les équipements suivants :

- Ouvrages électriques à superviser et contrôler.
- Voyants d'alarmes et alarmes sonores.

- Générateur de base de données fonctionnel sur un ordinateur portable (décrire la configuration de l'équipement et son environnement).
- Simulateur d'entrées et sorties.
- Un centre de conduite moyennant une liaison radio.

La synchronisation des équipements terminaux s'effectue via la liaison radio.

Ce type de poste asservi est installé dans les postes de coupures et les organes de coupures réseau (IAT et reclosers).

La procédure d'échange des données entre le RTU et le système SCADA est donnée par la figure suivante ;

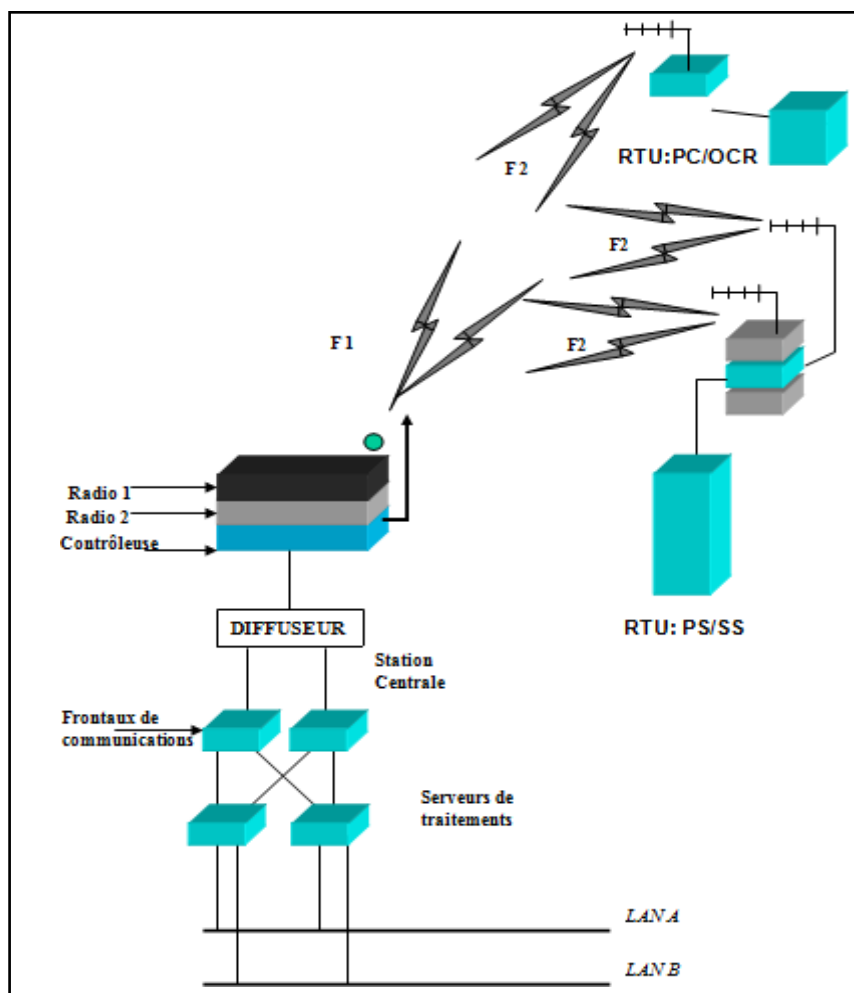


Figure 6. Procédure d'échange de données RTU-SCADA

3.2. Présentation des cartes programmables

Une carte programmable est une carte qui a besoin d'un "programme" pour fonctionner. Elle fonctionne un peu comme un être humain : s'il n'y a rien dans son "cerveau" (le microcontrôleur), elle ne sait rien faire et ne fait donc rien [1].

Elle a un microcontrôleur que l'on peut plus ou moins facilement programmer. C'est-à-dire qu'on peut installer un programme plus ou moins facilement dans ce microcontrôleur.

Si on devait reprendre l'idée qu'une carte programmable est à l'image d'un être humain, on pourrait dire qu'une carte programmable c'est le cerveau et le système nerveux (transmission des informations) d'un être humain. On peut aussi sans doute dire que c'est aussi le système sanguin puisque que de l'électricité la parcourt, source d'énergie pour faire fonctionner l'ensemble.

Le programme qui tourne dans un microcontrôleur est écrit en langage machine, une suite de "0" et de "1" un peu incompréhensible pour nous les humains. Heureusement, nous utilisons un langage informatique, certes différent de notre langage usuel qui nous permet de communiquer entre nous, mais c'est quand même plus simple et si de premier abord cela semble compliqué, très vite, on arrive à faire clignoter des lumières, réagir à un mouvement, actionner un moteur, ...

Pour programmer une carte électronique, on utilise un logiciel qui va s'occuper de transformer un programme informatique en langage machine : ce logiciel est appelé "compilateur"[2].

L'installation d'un programme dans son cerveau ne suffit pas car si on veut qu'elle puisse faire quelque chose, il faut lui donner les moyens de communiquer avec l'extérieur. Il y a donc nécessairement des broches accessibles pour brancher d'autres composants. On dit qu'elle possède un certain nombre d'entrées-sorties sur lesquelles on va pouvoir brancher des cartes ou des composants électroniques qui vont permettre de connaître certains paramètres de l'environnement de la carte programmable, d'agir sur cet environnement et de communiquer avec le monde extérieur (d'autres cartes programmables ou bien par exemple des sites internet).

La programmation du microcontrôleur peut être :

- Déjà faite au moment de l'achat, comme par exemple dans de nombreux jouets électroniques ;
- À faire par l'utilisateur comme sur les cartes *Arduino*, *Picaxe*, *Raspberry Pi* et autres.

3.2.1. Les cartes Arduino

Arduino [3][4] est une plateforme infotronique (Soft et Hard) sur lequel se trouve un microcontrôleur programmable pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses. Par exemple gérer automatiquement l'ouverture d'une porte de garage, envoyer un SMS quand le jardin est trop sec et gérer le système d'arrosage à distance, piloter un nouveau robot, ... Il faut pour cela associer à la carte Arduino des capteurs (de lumière, de température, de position, ...), des actionneurs (moteurs, pompe, ...), des organes de sortie (lampe, chauffage, ...), des circuits de puissance, une alimentation (piles, panneaux solaires, ...), des interfaces de dialogue (boutons, LEDs, écran, ...), des interfaces de communication (réseau filaire, réseau sans fil, ...),

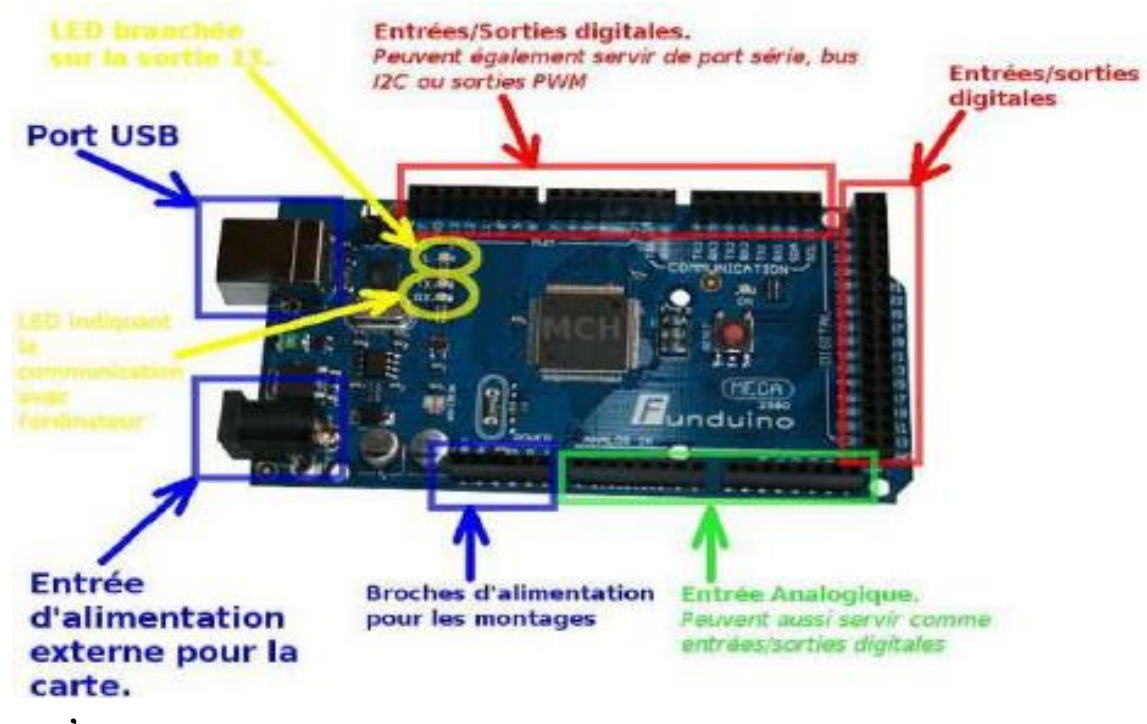


Figure 7. Exemple des cartes Arduino Méga adk

3.2.1.1. Les différents types d'Arduino

On peut classer les cartes Arduino en deux grandes familles :

- Les cartes Arduino officielles (ou « classique »), compatible hardware et software avec le « form factor » et l'IDE Arduino,
- Les cartes dérivées d'Arduino, compatible avec les Shields Arduino classique (mais pas avec l'ide Arduino de base).

Pour tout ce qui est des cartes Arduino dites « classiques » elles sont basées généralement sur le même microcontrôleur à savoir un ATMEGA328P du fabricant ATMEL.

Le tableau suivant donne une étude comparative entre les différents cartes Arduino

Cartes Arduino Caractéristiques	UNO R3 (classique & CMS)	UNO R3 Ethernet (classique & POE)	Leonardo	Mega 2560	Mega ADK	DUE	Esplora	Mini	Nano	Yun (classique & POE)	Zero PRO
Microcontrôleur	ATmega328P	ATmega328P	ATmega32u4	ATmega2560	ATmega2560	AT91SAM3X8E	ATmega32u4	ATmega328P	ATmega328P	ATmega32u4	ATSAMD21G18
Cadencement Horloge	16 MHz	16 MHz	16 MHz	16 MHz	16 MHz	84 MHz	16 MHz	16 MHz	16 MHz	16 MHz	48 Mhz
Tension d'entrée	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 9V	7 - 9V	5V	5V
Tension de fonctionnement	5V	5V	5V	5V	5V	3,3V	5V	5V	5V	5V	3,3V
Entrée/Sortie Numérique	14/6	14/4	20/7	54/15	54/15	54/12	✗	14/6	14/6	20/7	14/12
Entrée-Sortie (PWM) Analogique	6/0	6/0	12/0	16/0	16/0	12/2 (DAC)	✗	8/0	8/0	12/0	6/1 (DAC)
Mémoire vive (Flash)	32 Ko	32 Ko	32 Ko	256 Ko	256 Ko	512 Ko	32 Ko	32 Ko	32 Ko	32 Ko	256 Ko
Mémoire vive (SRAM)	2 Ko	2 Ko	2,5 Ko	8 Ko	8 Ko	96 Ko	2,5 Ko	2 Ko	2 Ko	2,5 Ko	32 Ko
Mémoire morte (EEPROM)	1 Ko	1 Ko	1 Ko	4 Ko	4 Ko	✗	1 Ko	1 Ko	1 Ko	1 Ko	16 Ko
Interface USB	USB-B mâle	USB-B mâle	Micro-USB	USB-B mâle	USB-B mâle & USB-A pour Android	2 ports micro- USB (Native et programming)	Micro-USB	✗	Mini-USB	Micro-USB	2 ports micro- USB (Native et programming)
Port UART	1	1	1	4	4	4	✗	✗	1	1	2
Carte SD	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Ethernet	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Wi-Fi	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Dimensions	68x53mm	68x53mm	68x53mm	101x53mm	101x53mm	101x53mm	165x60mm	30x18mm	45x18mm	68x53mm	68x53mm

Tableau 1. Tableau comparatif des différentes cartes Arduino [5]

3.2.1.2. Outil de programmation Arduino

Le logiciel de programmation des modules Arduino est une application Java multiplateformes, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware (et le programme) au travers de la liaison série (RS232, Bluetooth ou USB selon le module).

L'IDE Arduino permet :

D'éditer un programme : des croquis (sketch en Anglais),
les programmes sont écrits en langage C

De compiler ce programme dans le langage « machine » de l'Arduino,
la compilation est une traduction du langage C vers le langage du microcontrôleur
La console donne des informations sur le déroulement de la compilation et affiche les messages D'erreur.

De téléverser le programme dans la mémoire de l'Arduino,
le téléversement (upload) se passe via le port USB de l'ordinateur
un fois dans la mémoire de l'Arduino, le logiciel s'appelle un micrologiciel.
La console donne des informations sur le déroulement du téléversement et affiche les messages D'erreur.

De communiquer avec la carte Arduino grâce au terminal (ou moniteur série).
pendant le fonctionnement du programme en mémoire sur l'Arduino, il peut communiquer avec L'ordinateur tant que la connexion est active (câble USB, ...)

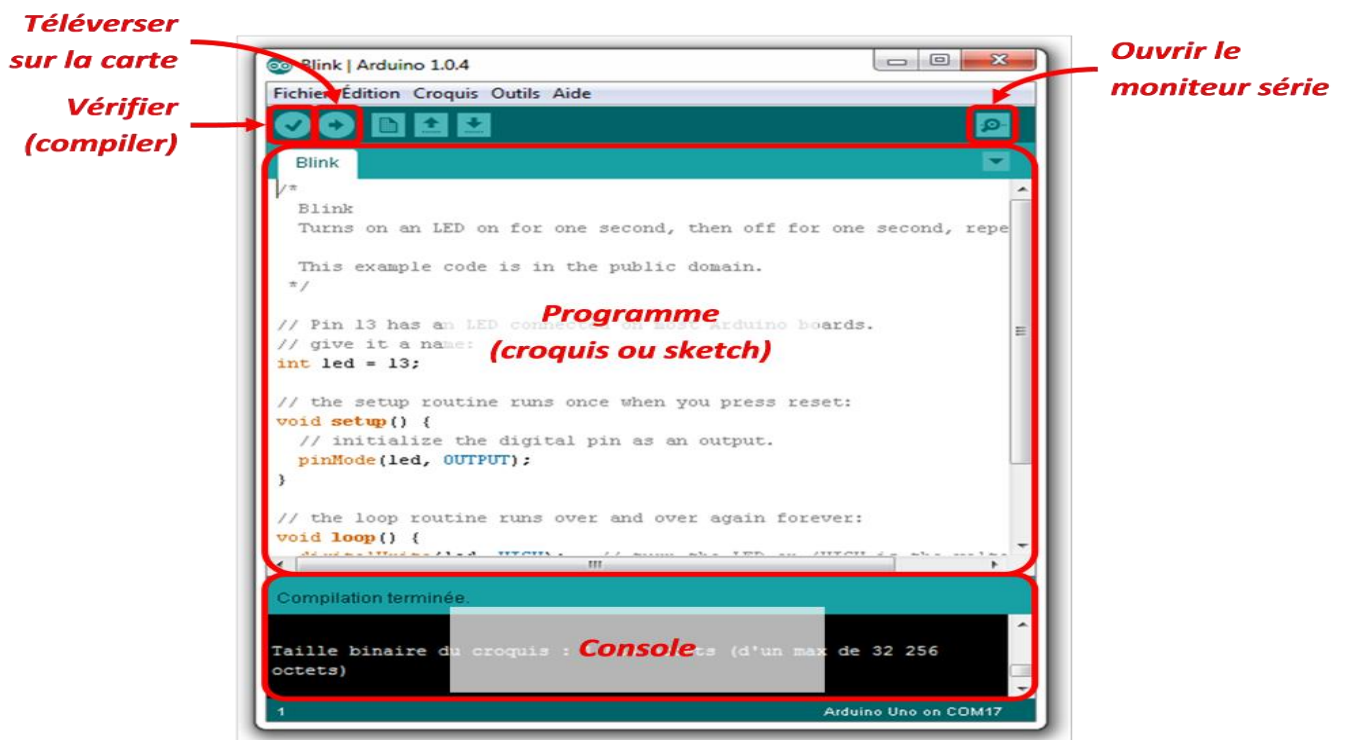


Figure 8. Logiciel Arduino

3.3.2. Les cartes Raspberry Pi

La plateforme *Raspberry Pi* [5] est un microcalculateur ou nano-ordinateur créé par l'anglais David Biraben, dans le cadre de sa fondation Raspberry Pi. Il a été conçu par une fondation éducative à but non-lucratif pour faire découvrir le monde de l'informatique sous un autre angle.



Le système d'exploitation conseillé est l'environnement libre et gratuit GNU/Linux (et principalement une « distribution » (version) dite Raspbian). Le processeur est d'un type « ARM » comme trouvé dans la plupart des smartphones de nos jours. Tout le stockage de données se fait par défaut par le biais d'une carte Micro SD. L'ordinateur est alimenté par un chargeur micro-USB comme celui d'un téléphone portable. Le processeur est capable de traiter les images et vidéos en Haute Définition. Il suffit de le brancher sur un écran HDMI ou téléviseur, clavier, souris par port USB et éventuellement le réseau et nous avons un ordinateur complet et fonctionnel.

A la différence de la plupart des ordinateurs en vente, la carte comporte 40 broches « GPIO » (Broches/picots d'Entrée-Sortie générale). C'est une invitation à l'électronique et l'interaction avec le monde extérieur. En quelques lignes de code, l'informatique dite « physique » devient un jeu d'enfant.








Figure 9. La famille des Raspberry Pi

Les principaux modèles de Raspberry sont le modèle A et le modèle B. Le tableau suivant décrit les spécifications techniques des modèles A et A+ :

Critère	Modèle A	Modèle A+
Carte		
Système sur puce (Soc)	BroadcomBCM2835 (CPU, GPU et SDRAM)	BroadcomBCM2835 (CPU, GPU et SDRAM)
Processeur (CPU)	700 MHz ARM1176JZF-S Core (ARM V6)	700 MHz ARM1176JZF-S core (ARM V6)
Processeur Graphique (GPU)	Broadcom video Core IV OpenGL ES 2.0 OpenVG 1080p30 H.264	Broadcom video Core IV OpenGL ES 2.0 OpenVG 1080p30 H.264
Mémoire vive (SDRAM)	256Mo	256Mo
Alimentation	300 mA (2.5 W) via micro-USB	200 mA (1 W) via micro-USB
Stockage des données	Lecteur de carte SD/MMC/SDIO	Lecteur de carte micro-SD
Sortie vidéo	1 composite RCA et 1 HDMI	1 composite RCA et 1 HDMI
Sortie audio	3.5 mm jack et HDMI	3.5 mm jack et HDMI
Ethernet	0	0
Port USB	1	1
Périphériques bas niveau	26 GPIO connecteurs	40 GPIO connecteurs
Taille	85.60 mm × 56.5 mm	65.00 mm × 56.5 mm
Poids	45g	23g
Prix	\$25	\$20

Et le tableau suivant décrit les différentes composantes du modèles B c'est-à-dire les spécifications techniques des modèles B, B+, 2, 3 et 3B+ :

Critère	Modèle B	Modèle B+	Modèle 2	Modèle 3	Modèle 3B+
Carte					
Système surpuce (Soc)	BroadcomBCM2835 (CPU, GPU et SDRAM)	BroadcomBCM2835 (CPU, GPU et SDRAM)	BroadcomBCM2836 (CPU, GPU et SDRAM)	Broadcom BCM2837 64 bit à quatre cœurs ARM Cortex-A53 à 1,2 GHz,	Broadcom BCM2837B0 64 bit à quatre cœurs ARM Cortex-A53 cadencé à 1,4 GHz
Processeur (CPU)	700 MHz ARM1176JZF-S core (ARM V6)	700 MHz ARM1176JZF-S core (ARM V6)	900 MHz quad-core ARM Cortex-A7 (ARM V7)	1,2 GHz Quadricœur ARM Cortex-A53	1,4 GHz Quadricœur ARM Cortex-A53
Processeur graphique (GPU)	Broadcom Video Core IV OpenGL ES 2.0 OpenGL 1080p30 H.264	Broadcom video Core IV OpenGL ES 2.0 OpenGL 1080p30 H.264	Broadcom video Core IV OpenGL ES 2.0 OpenGL 1080p30 H.264	Broadcom video Core IV OpenGL ES 2.0 OpenGL 1080p30 H.264	Broadcom video Core IV OpenGL ES 2.0 OpenGL 1080p30 H.264
Mémoire vive (SDRAM)	512Mo	512Mo	1Go	1Go	1Go
Alimentation	700 mA (3.5 W) via micro-USB	600 mA (3.0 W) via Micro-USB	800 mA (4.0 W) via micro-USB	800 mA (4 W)	800 mA (4 W)
Stockage des données	Lecteur de carte SD/MMC/SDIO	Lecteur de carte micro-SD	Lecteur de carte Micro-SD	Lecteur de carte Micro-SD	Lecteur de Carte micro-SD
Sortie vidéo	1 composite RCA et 1 HDMI	1 composite RCA et 1 HDMI	1 composite RCA et 1 HDMI	HDMI et Composite (via un connecteur Jack)	HDMI et Composite (via un connecteur Jack)

Sortie audio	3.5 mm jack et HDMI	3.5 mm jack et HDMI	3.5 mm jack et HDMI	stéréo Jack 3,5 mm (sortie son 5.1 sur la prise HDMI) et	stéréo Jack 3,5 mm (sortie son 5.1 sur la prise HDMI) et
--------------	---------------------	---------------------	---------------------	--	--

				Composite	Composite
Ethernet	1	1	1	10/100 Ethernet, Wifi 802.11n, Bluetooth 4.1	10/100/1000 Ethernet (Max. 300Mbps), WIFI 802.11ac, Bluetooth 4.2
Port USB	2	4	4	4	4
Périphériques Bas niveau	26 GPIO Connecteurs	40 GPIO Connecteurs	40 GPIO connecteurs	40 GPIO Connecteurs	40 GPIO Connecteurs
Taille	85.60 mm × 56.5 mm	85.60 mm × 56.5 mm	85.60 mm × 56.5 mm	85,60 mm × 53,98 mm × 17 mm	85,60 mm × 53,98 mm × 17 mm
Poids	45g	45g	45g	45 g	45 g
Prix	\$35	\$35	\$35	\$35	\$35

1. Conclusion

Dans ce chapitre nous avons présenté l'organisme d'accueil, nos objectifs et les outils nécessaires pour le développement de notre code relatif au système à mettre en place. Le chapitre suivant va être dédié à l'étude des besoins de ce système.

Chapitre 2

ANALYSE ET SPECIFICATION DES BESOINS



Chapitre2

Analyse et spécification des besoins

1. Introduction

L'erreur la plus fréquente qui a de lourdes conséquences est de se lancer dans la programmation en ayant une idée superficielle de ce qu'on attend du projet. Pour balayer ces erreurs, une phase d'analyse des données constitue le point de passage obligatoire de toute conception. Il s'agit d'une étape cruciale dans la réalisation du projet.

Cette phase consiste d'abord à collecter toute donnée en rapport direct avec notre sujet. Une fois l'ensemble des besoins est identifié et les spécifications sont établies, il faut exprimer clairement les objectifs du projet à atteindre. C'est la fameuse phase d'analyse préliminaire qui sert à bien expliquer l'architecture du système dès que le cahier des charges est bien formulé.

Ce chapitre sera consacré à détailler les différents besoins attendus du projet c'est-à-dire une analyse détaillée du cahier des charges en fonction des contraintes imposées.

2. Cahier des charges

Comme expliqué dans le chapitre précédent, les objectifs de notre projet sont principalement de trouver des solutions pour le manque des pièces de rechanges des RTU, l'encombrement au niveau des IAT et des PCs à cause de l'existence de plusieurs composantes dépendantes mais répartis sur trois coffrets et aussi certains problèmes de communications.

Notre cahier de charge se résume dans le fait regrouper les trois coffrets répartis à savoir ; le coffret de commande, l'atelier d'énergie et la RTU dans un seul coffret pour résoudre à la fois le problème d'encombrement au niveau des IAT et celui de manque de pièce de rechanges.

Par la suite, on doit trouver une solution pour la remise à zéro de la RTU après un blocage au niveau des composants sans la nécessité de déplacement d'un agent. Et optimiser la durée de vie des batteries au niveau des IAT par le fait de désactiver de l'indicateur visuel indiquant le passage de courant de défaut juste après la localisation de défaut.

3. Méthodologie de travail

Pour la mise en place d'un système permettant de résoudre les problèmes mentionnés dans la section précédentes, UML est idéal pour déployer notre architecture logicielle permettant de réaliser nos objectifs du fait qu'il est un modèle universel permettant de modéliser une application selon une vision objet. Plus concrètement, il possède plusieurs facettes. C'est une norme, un langage de modélisation objet, un support de communication performant et un cadre méthodologique.

L'outil Star UML a été utilisé dans la phase de conception.

Avant de présenter les détails de l'étude conceptuelle, il était nécessaire de commencer par présenter les outils requis pour le développement de l'application.

3.1. *Choix des outils de développement*

Côté configuration de la base de données de la RTU, nous allons utiliser les logiciels fournis par le constructeur.

Côté carte programmable, quel que soit la carte choisie, le langage de programmation est le langage C, langage structuré et évolué permettant d'effectuer des opérations de bas niveau (assembleur) et surtout caractérisé par sa portabilité due à l'emploi de bibliothèques dans lesquelles sont reléguées les fonctionnalités liées à la machine. Pour créer et compiler le programme en C, nous allons utiliser les logiciels disponibles en téléchargement gratuit pour la programmation des cartes avec les bibliothèques nécessaires.

Pour le développement de l'application mobile, nous avons choisi le langage *Android*. Et l'application web, nous avons choisi le langage *php*.

3.2. Cycle de vie et planning prévisionnel du projet*

Pour définir le travail technique à exécuter à chaque phase de ce projet et assurer la cohérence au niveau de sa réalisation ainsi que la qualité du travail, nous nous sommes appuyés sur un cycle de vie en V.

En fait, le modèle en V demeure actuellement le cycle de vie le plus connu et certainement le plus utilisé. Il s'agit d'un modèle en cascade dans lequel le développement des tests et des logiciels sont effectués de manière synchrone.

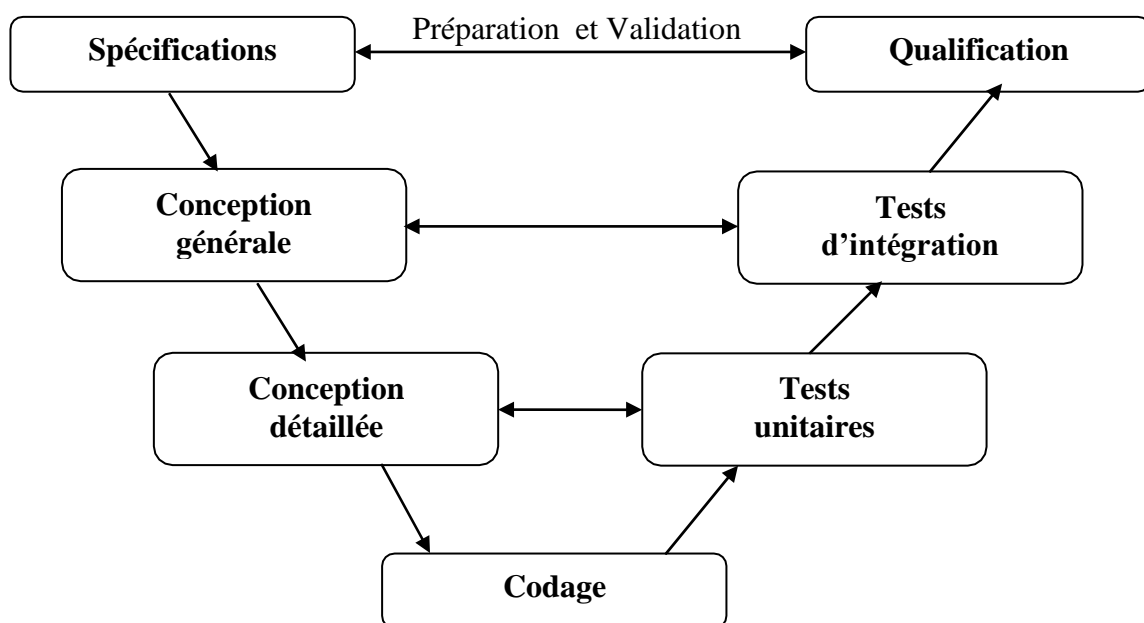


Figure 10. Cycle de vie en V

Le principe de ce modèle est qu'avec toute décomposition doit être décrite la recombinaison et que toute description d'un composant est accompagnée de tests qui permettront de s'assurer qu'il correspond à sa description. Ceci rend explicite la préparation des dernières phases (validation-vérification) par les premières (construction du logiciel), et permet ainsi d'éviter un écueil bien connu de la spécification du logiciel : énoncer une propriété qu'il est impossible de vérifier objectivement après la réalisation.

La représentation en V tient d'avantage compte de la réalité, le processus de développement n'est pas réduit à un enchaînement de tâches séquentielles. Elle montre que:

- C'est en phase de spécification que l'on se préoccupe des procédures de qualification,
- C'est en phase de conception globale que l'on se préoccupe des procédures d'intégration,
- C'est en phase de conception détaillée que l'on prépare les tests unitaires.

Le modèle de cycle de vie en V permet d'anticiper sur les phases ultérieures de développement du produit. En particulier le modèle en V permet de commencer plus tôt:

- Plan de tests de qualification,
- Plan d'évaluation des performances.

En suivant ce modèle de cycle de vie, dès la première semaine de réalisation du projet, nous avons établi un planning prévisionnel afin de bien délimiter le travail. Ce planning permet surtout de se donner des objectifs à réaliser et permet de ne pas se lancer dans plus d'une tâche en même temps. Suite à ça, nous avons pu définir des sous-ensembles cohérents d'activités qui pourraient faire un projet plus structuré.

Voici ci-après le tableau *Gantt* donné par la figure suivante qui récapitule le planning du déroulement du projet donnant ainsi un calendrier prévisionnel. Il nous a permis de prendre conscience de la charge de travail et des délais qui nous ont été fixés.

Nom	▲ Date de début	Date de fin
• Etablissement du cahier des charges détaillés	04/02/23	06/02/23
• Etude de système SCADA	07/02/23	14/02/23
• Etude des cartes programmables	15/02/23	20/02/23
• Conception de système à mettre en place	21/02/23	06/03/23
• Réalisation de coffret RTU	07/03/23	27/03/23
• Développement de code source de la carte progr...	28/03/23	10/04/23
• Développement de l'application web/mobile	11/04/23	22/05/23
• Test final de système	23/05/23	30/05/23
• Rédaction du rapport	24/05/23	28/05/23

Tableau2- 1. Planning prévisionnel du projet

4. Identification des besoins

Cette phase marque le démarrage de l'analyse du système à réaliser. Elle consiste en la reconnaissance des objectifs du projet et l'identification des cas d'utilisation.

4.1. Enoncé des besoins fonctionnels

Cette phase représente un point de vue fonctionnel de l'architecture du système. Par le biais des cas d'utilisation identifiés dans la figure suivante, nous avons pu représenter un ensemble des séquences d'actions à être implémentées. Chaque cas exprime une interaction entre les acteurs de l'application.

L'objectif du projet est de mettre en place une RTU intelligent. Cette dernière doit assurer les fonctions habituelles d'un RTU et de pouvoir permettre à l'agent de ;

- Pouvoir savoir l'état de la RTU,
- Pouvoir envoyer une commande à la RTU via son appareil mobile,
- Pouvoir visualiser les différentes RTU sur le réseau et les chemins actifs et inactifs du réseau,
- Pouvoir consulter l'historique et l'archives du réseau à travers un tableau de bord simple à utiliser,
- Avoir une durée de vie importante,
- Avoir un coût très réduit par rapport aux solutions existantes,
- Être facile à mettre en place.

Dans l'ensemble, notre application est constituée de deux parties : la première est consacré à la mise en place d'un nouveau plan de câblage et une nouvelle configuration des bases de données des équipements pour sortir des trois coffrets un seul assurant les mêmes fonctionnalités, la deuxième partie consiste à l'intégration d'une carte programmable au niveau de la RTU avec un module de communication sans fil et le développement des logiciels nécessaires pour assurer les fonctions que nous avons cité au début de ce titre. Il s'agit de mettre en place ;

- Un programme en langage bas niveau installé au niveau du microcontrôleur pour communiquer avec les différents composants de la RTU et fournir les données nécessaires à l'utilisateur,
- Une application mobile et une interface web permettant la communication avec le microcontrôleur d'une part et d'interface de paramétrage utilisateur d'autre part.

La figure suivante représente les différents cas d'utilisations de système à mettre en place ;

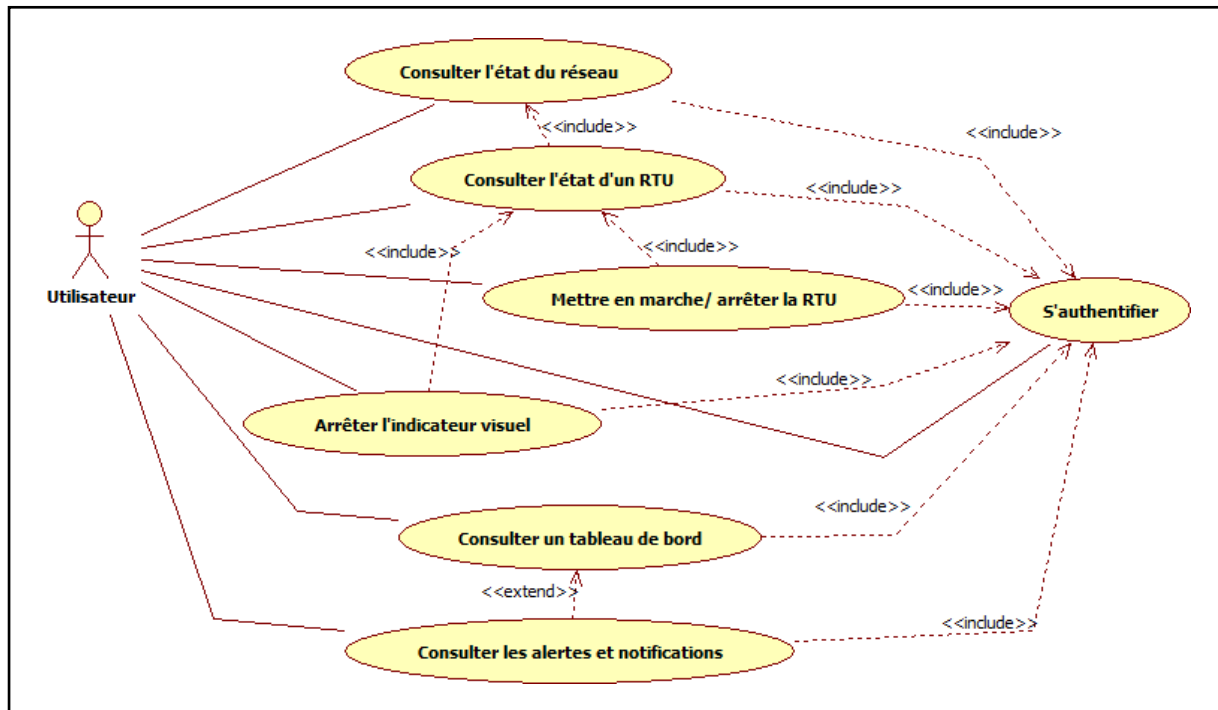


Figure2- 1. Diagramme des cas d'utilisation identifiés

L'utilisateur dans notre cas peut être soit l'agent responsable à l'installation et la maintenance des RTU soit son supérieur.

4.2. Enoncé des besoins non fonctionnels

Les besoins non fonctionnels de système à mettre en place peuvent être décrits comme suit :

a. Contraintes de sécurité :

Le système ne doit pas être accessible qu'à la personne ayant le droit d'accès. Côté Software, cette fonctionnalité doit être assurée par un mécanisme de gestion des droits d'accès. Côté

Hardware, seuls les agents de la STEG ayant le droit de manipuler les coffrets installés sursites.

Contraintes esthétiques et ergonomiques :

Les contraintes esthétiques sont spécifiques à la partie Software. On parle de ;

- La simplicité de l'interface
- La convivialité et l'ergonomie : L'application doit avoir des interfaces conviviales, faciles à utiliser, adaptées à chaque type d'utilisateur. Elles doivent combiner des données textuelles bien claires et des données graphiques bien structurées.
- Une navigation aisée avec des liens simples et personnalisés en assurant l'harmonie graphique de chaque page.
- Privilégier la clarté et la lisibilité.
- L'utilisation des couleurs significatives et qui reflètent l'aspect fonctionnel de l'application.

5. Conclusion

Dans ce chapitre, nous avons présenté le cahier des charges du projet et nous avons ainsi fait un recueil des besoins fonctionnels et des besoins non fonctionnels. Par la suite nous avons défini la méthodologie à suivre et le cycle de vie prévisionnel du projet. Dans ce qui suit de ce rapport nous allons commencer par une conception détaillée du projet.

Chapitre 3

CONCEPTION



Chapitre3

Conception

1. Introduction

Cette partie a pour but de décrire toutes les étapes de l'étude conceptuelle du projet et de la présentation des diagrammes de la partie logicielle. Nous allons procéder par définir l'architecture à mettre en place, et de la détailler par la suite.

2. Présentation de la solution

Il s'agit de donner une vue fonctionnelle représentant le graphe fonctionnel de projet. C'est une architecture centralisée nous permettant de voir le système final qui sera développé. En effet, les parties intervenantes dans la mise en place finale de ce projet sont: un coffret RTU, une application embarquée, une application mobile Android et une interface web.

Le graphe de la figure suivante modélise l'architecture à mettre en place ;

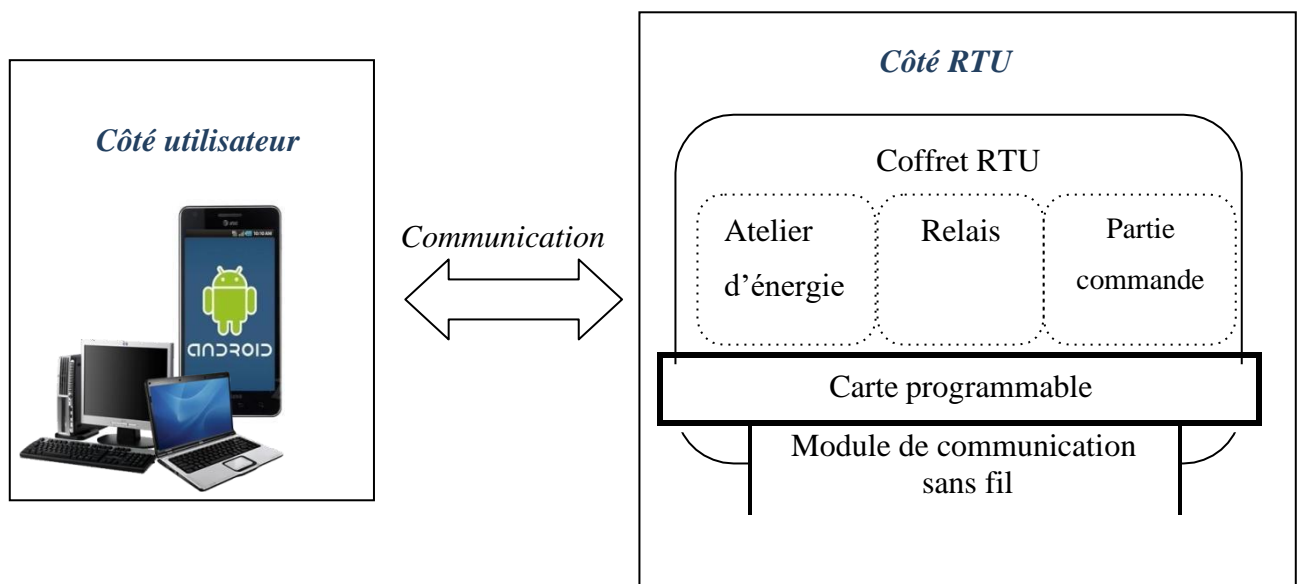


Figure 11. Schéma fonctionnel du système proposé

2.1.1. Côté RTU

La meilleure solution aux manques des pièces de rechanges était le fait de concevoir un nouveau plan de câblage et d'intégrer les coffrets atelier d'énergie et la partie commande dans un seul coffret avec la partie classique d'une RTU. Les détails de ce passage seront exposés dans le chapitre suivant. Ainsi, nous étions face au défi de trouver un bon marché des composants nécessaires assurant les mêmes fonctionnalités et exploiter certains ports non utilisés dans la conception de fournisseur de la solutions SCADA de la BCC.

2.1.2. Côté carte programmable

Pour communiquer avec le reste de système et moyennant son microcontrôleur, la carte programmable va traiter les données qui arrivent sur son Pin de réception. Le programme installé au niveau de microcontrôleur va se charger de transformer les commandes reçues en un format bien défini et les envoyer au serveur dans une boucle et avec un délai d'envoi bien défini. La figure suivante résume le fonctionnement de ce programme.

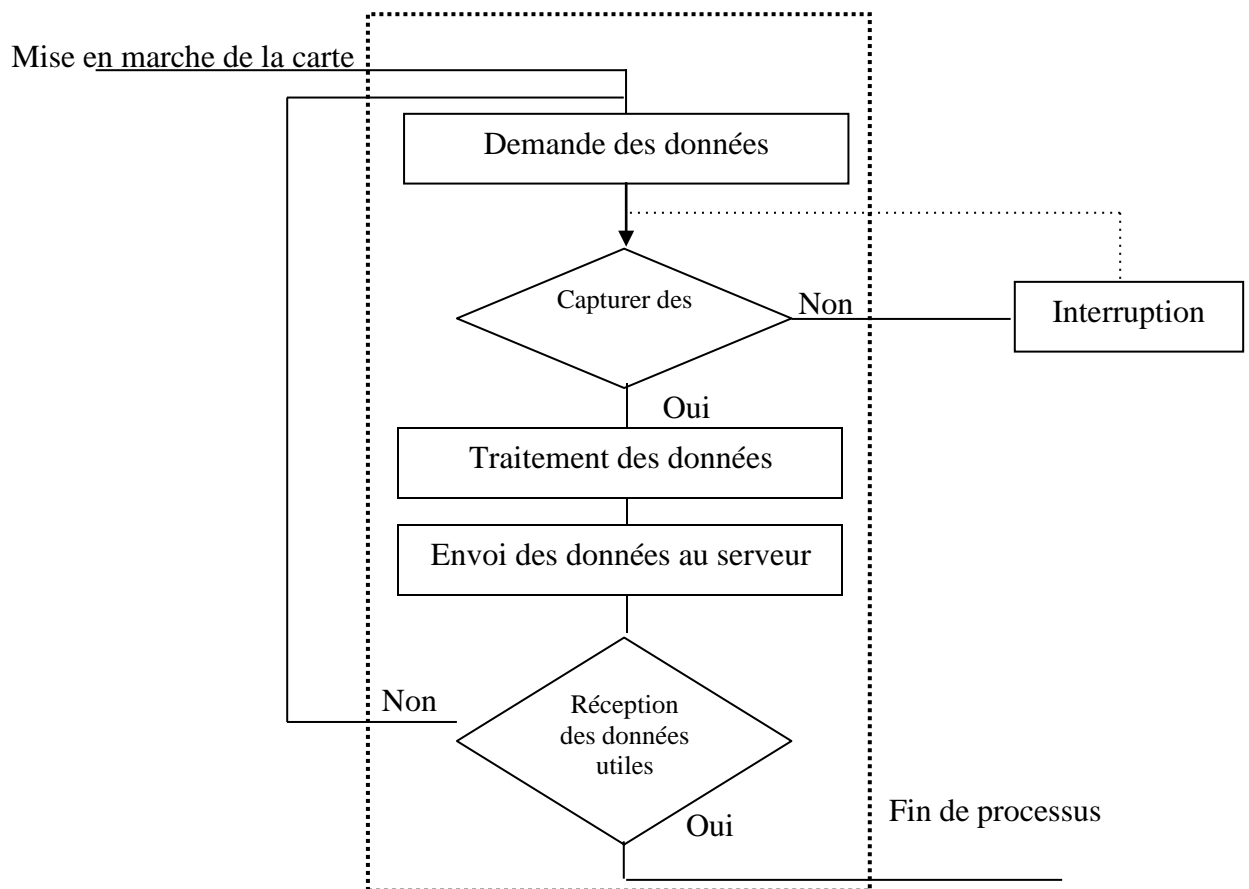


Figure 12. Descriptif fonctionnel du programme de contrôle

2.1.1. Côte interface utilisateur

C'est une interface graphique utilisateur soit installé sur un Smartphone Android communicant ordinateur comme serveur pour les données enregistrées à partir de la carte de contrôle, soit accessible à travers un navigateur web affichant les données à partir d'une base de données recevant les informations à partir de la carte de contrôle.

Une vue globale sur la structure du programme nous a permis de définir trois zones de l'interface à produire :

- Une zone de commande c'est-à-dire une zone pour l'envoi de requête,
- Une zone pour la détection de l'état de la RTU,
- Une zone pour le paramétrage de l'application.

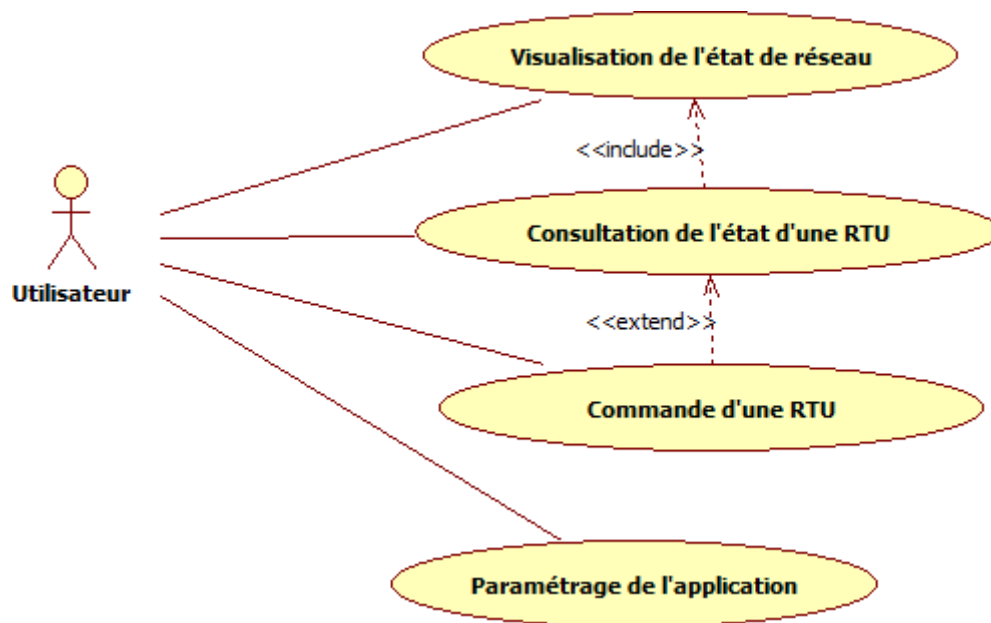


Figure 13. Vue globale sur les cas d'utilisation de l'interface utilisateur

3. Côté hardware

A ce stade, il est nécessaire de définir les différentes parties ou module du programme embarqué au niveau de la carte programmable. Ainsi, il est divisé en un cycle de trois phases comme l'indique la figure suivante ;

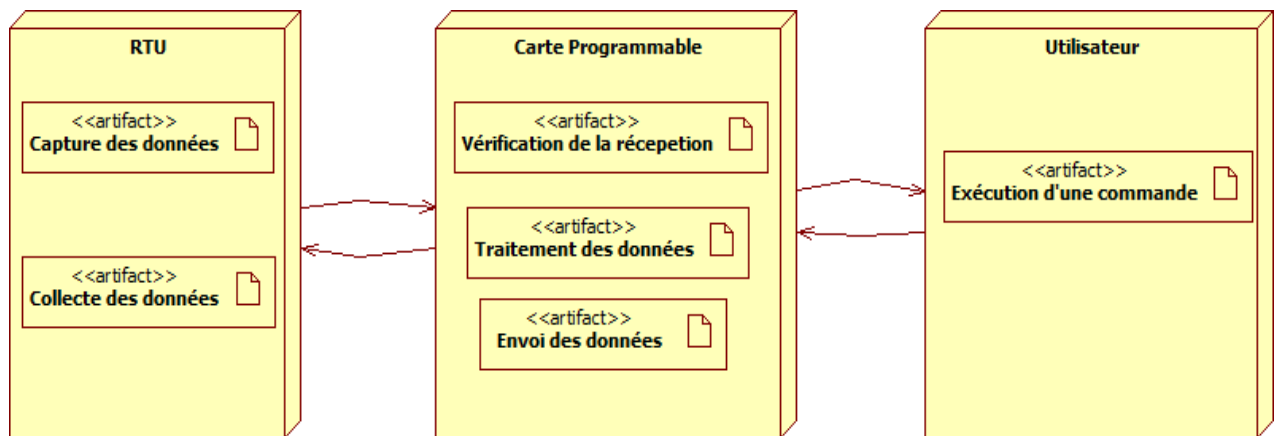


Figure 14. Conception logiciel du programme embarqué au niveau de la carte programmable (diagramme de déploiement)

Le diagramme décrit par la figure suivante donne une vue détaillée sur le fonctionnement interne du programme en C qui sera par la suite installé au niveau du microcontrôleur de la carte programmable. Ainsi, notre programme commence par initialiser le système. Ensuite, il se met à mesurer les niveaux de tension sur l'une de ses broches. Une fois cette séquence est reçue, il exécute le processus de transformation de cette valeur en une mesure, sinon il donne un message d'erreur.

Le programme embarqué traite ces données en temps réel et les envoie au serveur de données. Dans le cas où un ordre de commande est reçu, le microcontrôleur se charge de traiter cette commande en exécutant l'ordre et il envoie un message de bonne réception de l'ordre. Ce processus est modélisé dans le diagramme d'activité suivant ;

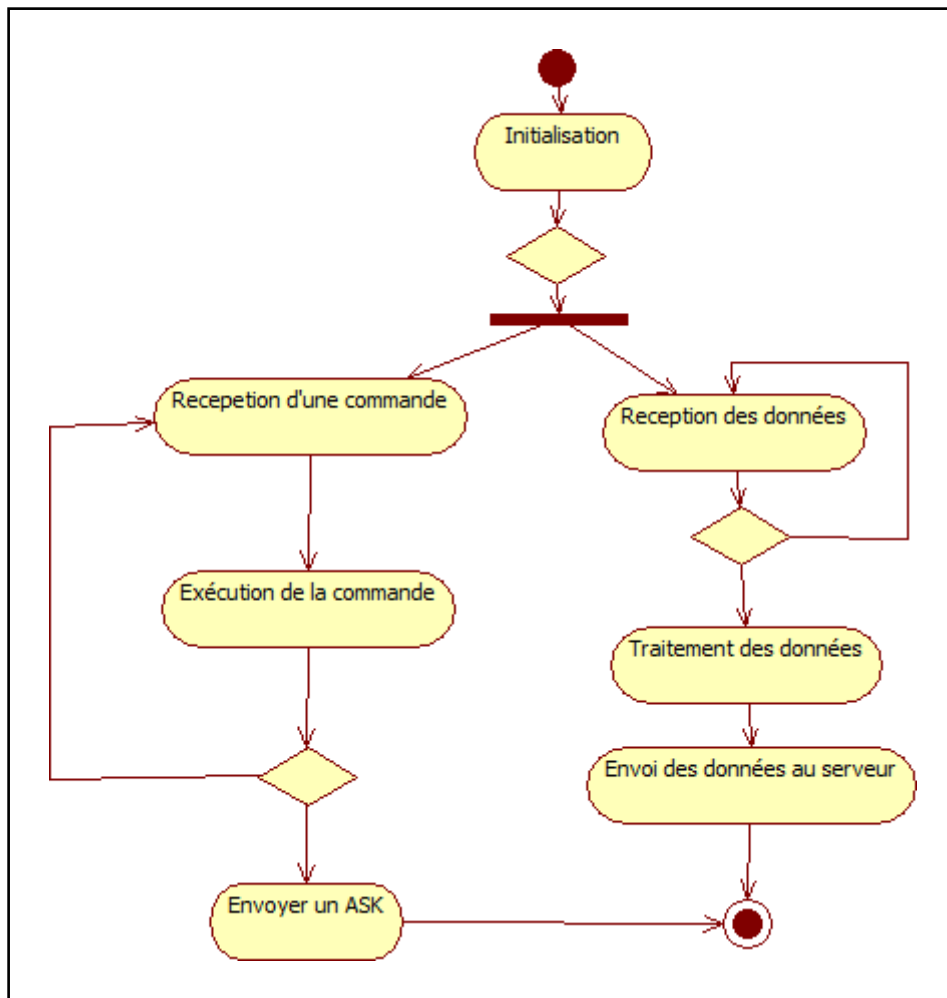


Figure 15. Diagramme d'activité du programme Arduino

4. Côte software

Concernant l'interface graphique de l'application mobile, le développement doit être fait selon les fonctionnalités demandées. Ainsi nous pouvons identifier cinq classes pour notre interface :

- Une classe d'authentification,
- Une classe d'accueil qui permet de donner accès au programme et donne les autorisations nécessaires selon les paramètres choisis pour l'utilisation du logiciel,

- Une classe de configuration et de paramétrage de l'application qui permet la communication initiale avec les différents modules de l'application et leurs configurations,
- Une classe de réception des données envoyées par la carte de contrôle,
- Une classe de contrôle de système permettant de donner l'ordre à ce dernier.
- Une classe de statistique pour un traitement avancé des données.

L'interface web assure les mêmes fonctionnalités que l'application mobile mais avec plus de détails et un petit espace d'administration.

2.2. Modèle d'analyse de l'interface utilisateur

Dans les parties précédentes, nous avons déterminé le diagramme de cas d'utilisation et l'architecture générale de l'application. Dans cette partie et pour assurer une meilleure structuration de la solution finale, nous allons fragmenter notre application en différents *packages* et déterminer les interactions entre eux comme le montre la figure suivante. C'est l'interface utilisateur web ou mobile qui gère tous les autres packages représentant chacune des fonctionnalités spécifiques. Elle est donc responsable de définir les interdépendances entre les différentes classes de l'application. Le diagramme donné par la figure suivante décrit cette interaction.

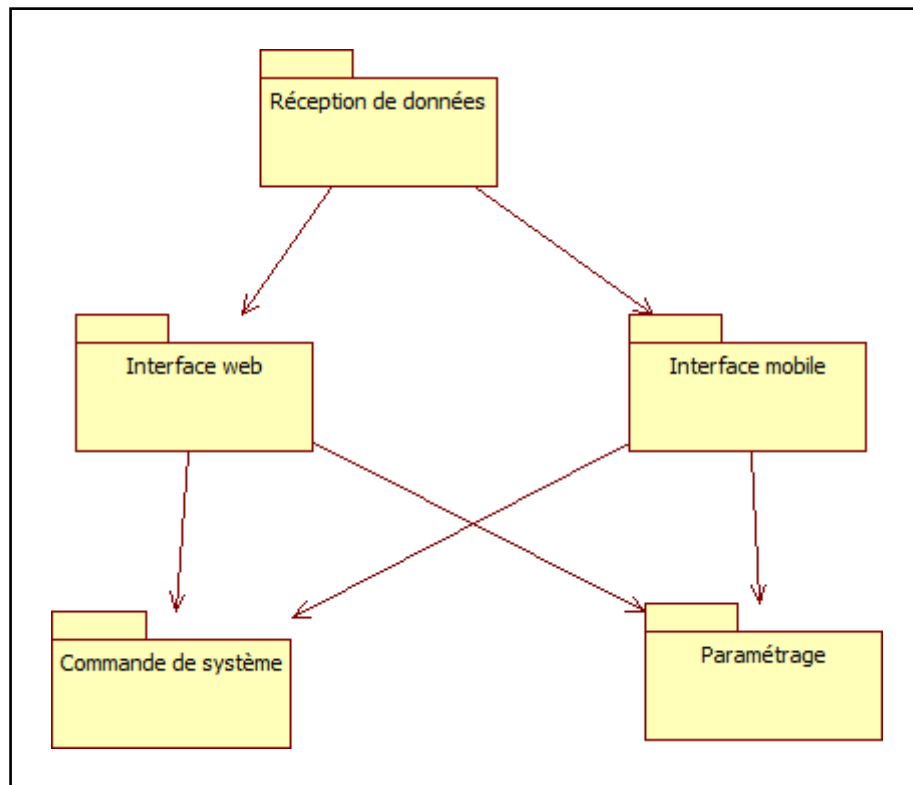


Figure 16. Diagramme de package

2.3. Diagramme des classes

Le diagramme de classe est une collection d'éléments de modèle statique, tels que des classes, des interfaces et leurs relations, connectés entre eux comme un graphe. Il représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. C'est le diagramme pivot de l'ensemble de la modélisation d'un système. La figure suivante illustre le diagramme des classes de notre application ;

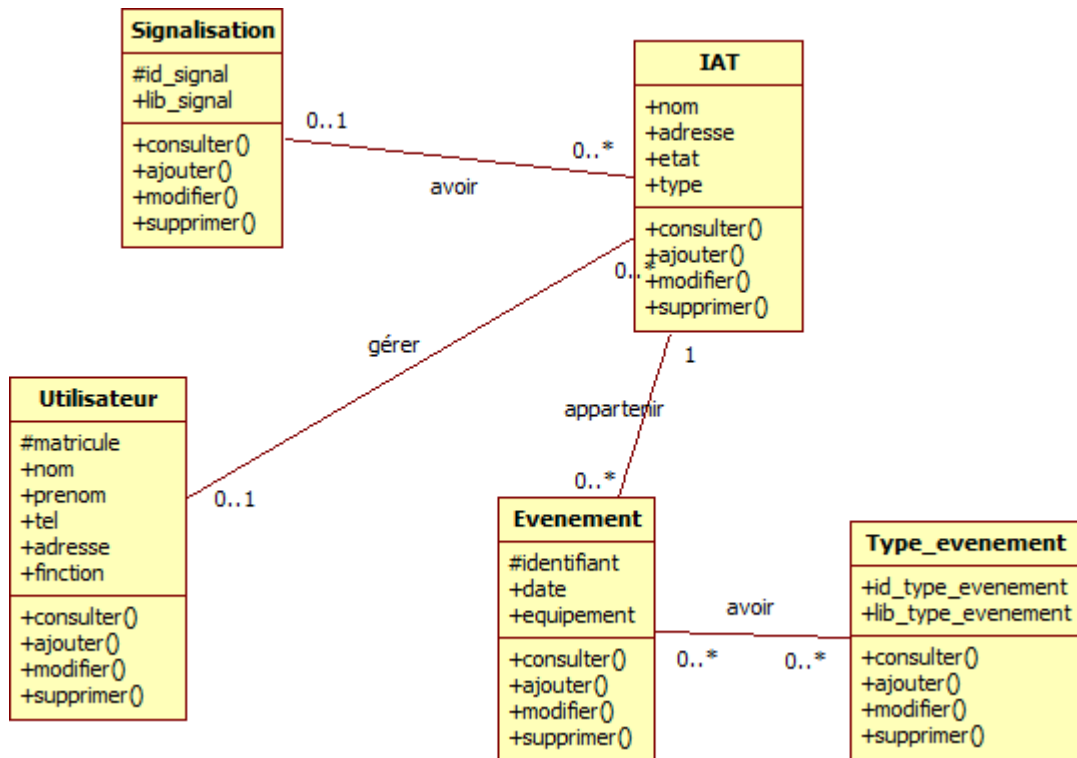


Figure 17. Diagramme des classes

2.4. Diagrammes des séquences

Ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.

Le diagramme de séquence de cas d'utilisation « *consulter l'état d'une RTU* » est donné par la figure suivante ;

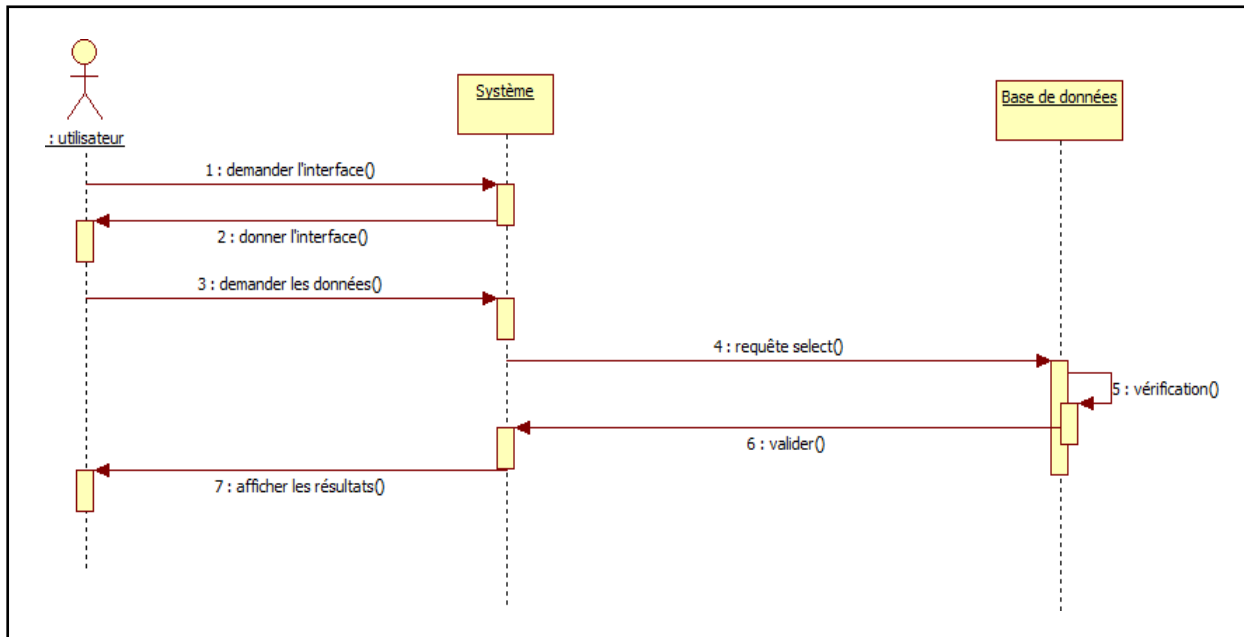


Figure 18. Diagramme de séquence de cas d'utilisation « consulter l'état d'une RTU »

Le diagramme de séquence de cas d'utilisation « *Envoyer une commande à la RTU* » est donné par la figure suivante ;

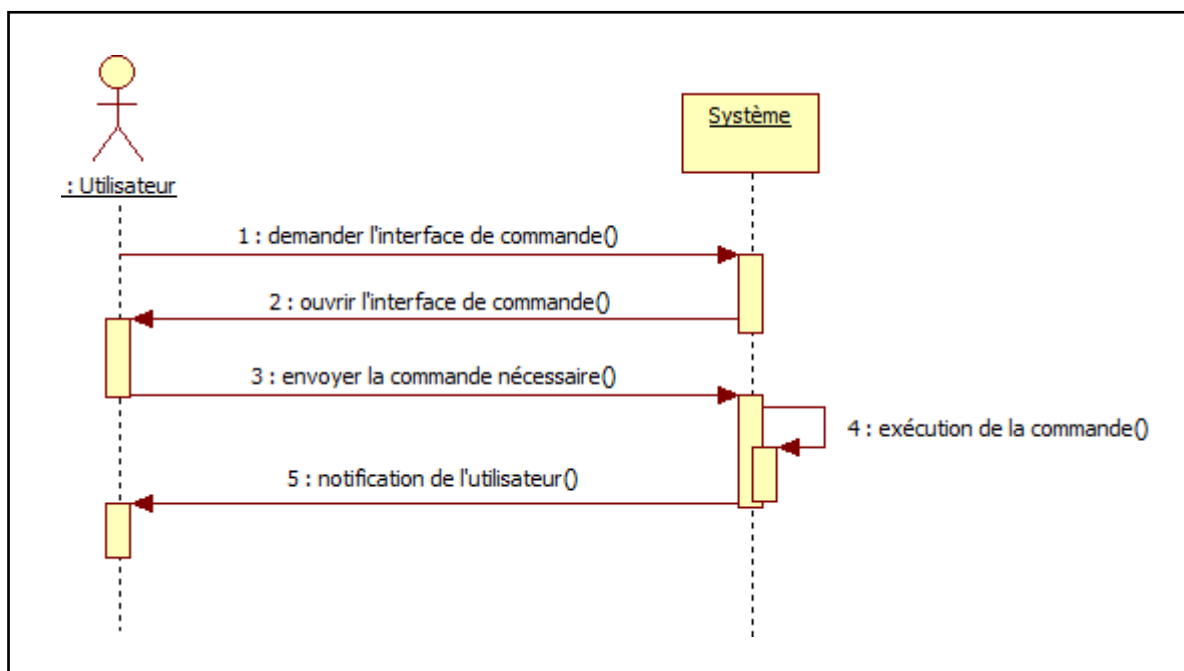


Figure 19. Diagramme de séquence de cas d'utilisation « Envoyer une commande à la RTU »

2.5. Protocole d'échange

La communication entre la RTU, le serveur de données et l'application mobile/web est un système de requête /réponse parfois et un système de transfert de données/acquittement autrefois.

Le rôle de ce protocole de communication ne sera en aucun cas de prendre des décisions, mais uniquement d'assurer une transmission sécurisée d'un côté à l'autre. Nous avons étudié les deux protocoles de communication UMTS et Wi-Fi et nous avons résumé les principales caractéristiques de chaque protocole dans le tableau suivant et notre choix s'est porté sur le Wi-Fi du fait qu'il est plus adapté de point de vue débit de transfert des données.

Standard	UMTS	Wi-Fi
Norme	3G	802.11a/b/g/n/n-draft
Portée (mètres)	300 (et plus)	10-100
Capacité du réseau (nœud)	Un grand nombre de Connexions.	Quelques dizaines d'utilisateurs
Débit de transfert de données	1 Mbit/s	11-54-108-320 Mb/s

Tableau 2. Protocoles de communication

Ce choix est juste pour valider le principe d'une unité terminale télécommandée mais nous allons étudier d'autre protocole en perspective et choisir le plus adapté pour l'élaboration d'un prototype final fonctionnel.

3. Conclusion

Ce chapitre résume les principales étapes de conception. Ainsi, nous avons présenté l'architecture de l'application et les différents diagrammes servant au développement de l'application par la suite.

Chapitre 4 :

REALISATION



Chapitre 4 : Réalisation

1. Introduction

Cette partie décrit toutes les étapes de réalisation du projet. Elle sera donc consacré à décrire pas à pas les étapes de développement de notre application. Nous finirons par étudier les performances de ce travail et la proposition des différentes améliorations possibles.

2. Mise en place de la RTU

Ace stade, nous procédons par décrire les processus de développement de différentes composantes du projet.

2.1. Réalisation du coffret

Pour résoudre au problème de l'encombrement des composants qui cause principalement la difficulté d'identification et de localisation de défaut, nous avons généré un nouveau plan de câblage assez simple et réduit coté matériels.

La reconstruction du nouveau coffret mixte contenant l'atelier d'énergie, le coffret de commande et la RTU a permis de ;

- Faciliter d'identification du défaut
- Simplifier la maintenance et de réparation.
- Réduire les dépenses économiques

Les étapes de réalisation du coffret mixte sont les suivantes :

- 1. Dessin des nouveaux plans de câblage sur AUTOCAD :** Nous avons commencé par le dessin de nouveau plan de câblage sur AUTOCAD comme l'indique la figure suivante. Nous avons pu concevoir un nouveau plan de câblage assez simple et réduit coté matériels. Ce plan est simple à exploiter et permet la réduction des matériels et donc une réduction des dépenses.

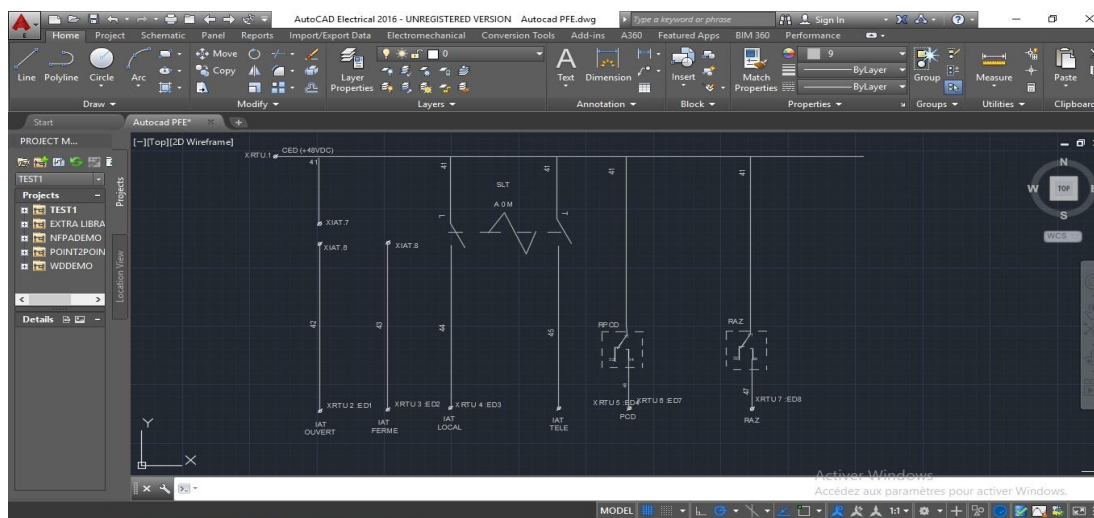


Figure 20. Dessin de nouveau schéma de câblage

- 2. Choix des composants à utiliser :** le tableau suivant décrit la liste exhaustive des composants utilisés pour l'ancien câblage et celle du matériel à utiliser dans le nouveau câblage. (La liste des différents composants avec une photo du composant et le rôle qu'assure chacun est donnée en annexe1)

Ancien câblage	Nouveau câblage
1 disjoncteur 220V	1 disjoncteur 220V
1 disjoncteur 48V	2 disjoncteurs 48V
1 chargeur E/S 220V/48V	1 chargeur E/S 220V/48V
2 contacteurs	2 contacteurs
Convertisseur	Convertisseur
8 batteries	4 batteries
1 radio	1 radio
10 relais	7 relais
1 carte relais	1 détecteur passage courant de défaut (DIN)
1 SAITEL100	1 SAITEL100
1 bouton poussoir marche arrêt	1 bouton poussoir marche arrêt
1 sélecteur (TELE/LOCAL/CCOMDANNEE)	1 sélecteur (TELE/LOCAL)
1 voyant (marche/arrêt/condamné)	1 voyant (marche/arrêt)

Tableau 3. Les matériels utilisés en ancien câblage et nouveau câblage

3. Assemblage de coffret mixte : à ce stade, il s'agit de choisir un coffret conteneur et de mettre en place les différents composants.
4. Réalisation du câblage : En suivant les nouveaux plans de câblage réalisés (voir annexe 2), nous avons pu mettre en place le nouveau « *coffret mixte* » illustré par la figure suivante ;

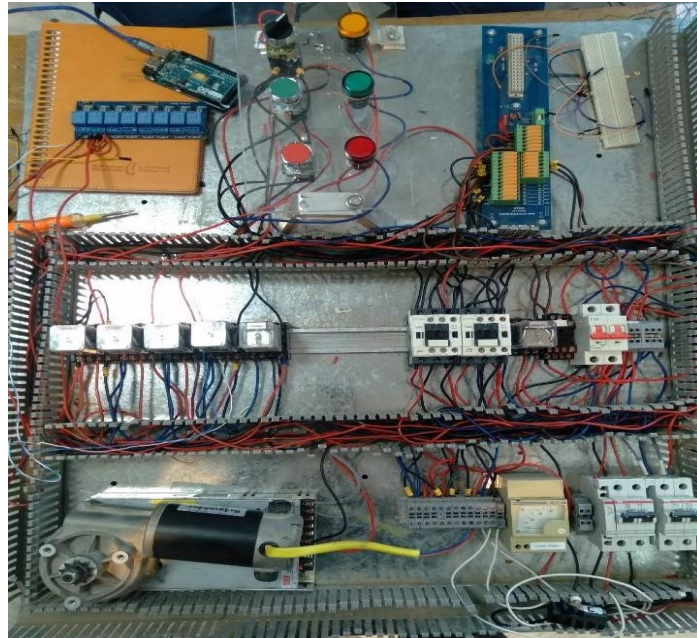


Figure 21. Coffret mixte

Coffret assure la même fonctionnalité mais facile à exploiter et à réparer. Il permet d'identifier facilement une défaillance donnée du fait que le câblage est repéré fil par fil.

2.2. Configuration de la base de données

Dans cette phase, nous allons décrire les étapes de configuration manuelle de la base de données de l'équipement SAITEL 100 (équipement clé dans la RTU) (voir annexe 3) pour permettre d'exploiter les deux ports existants de plus au niveau de ce composant et non exploité par le constructeur. L'objectif de ce travail est d'utiliser ces deux entrées de plus pour envoyer d'autres commandes. Et il nous faut alors une petite modification au niveau de câblage de ce composant.

Ce travail va résoudre deux principaux problèmes à savoir

- Le verrouillage des coffrets de commande des IAT type ABB est un obstacle pour les télécommander, comme il empêche les manipulations de secours et de balance

de charge lors d'un défaut enregistré sur le réseau HTA. Ce type de défaut est dû à la construction. La solution est d'effectuer manuellement la remise à zéro et sous tension de disjoncteur 48v en envoyant un agent sur lieu.

- Le DIN, c'est un élément de détection de passage du courant de défaut, en cas de défaut, une lampe est excitée et reste clignotante pour signaler le défaut jusqu'à le déplacement du contremaitre réseau pour l'éteindre manuellement. L'inconvénient est que ce voyant est alimenté par les batteries, le déchargement des batteries diminue leur durée de vie après trois heures et par conséquent la perte de communication de tout l'ouvrage. La solution adaptée est le déplacement d'un agent STEG sur terrain pour faire l'effacement de courant de défaut et donc l'arrêt de voyant.

En fait, la solution proposée est l'effacement de défaut de coffret de commande et l'effacement de courant de défaut à distance. Cette solution est composée de deux parties :

2.2.1. Partie câblage :

Afin de remédier au problème rencontré ; une solution nécessite la modification de câblage de coffret de commande par changement de commande manuelle RESET par une autre à distance. En effet, l'ancienne commande « RESET » est assurée par un bouton poussoir, fermée en marche normale et assure l'alimentation en +48DC du coffret de commande, une fois le bouton poussoir est activée il coupe l'alimentation du coffret et par conséquent le rafraichissement de tous les alarmes et le déverrouillage.

L'opération d'effacement de défaut à distance nécessite un relais SCHRAK MT221048, ce relais permet, comme l'indique la figure suivante, le passage de l'alimentation +48VCC avec son contact fermé alimenté par : +48VCC à travers le bornier 11 (01) et une +48VCC bornier 12 (04) vers le coffret de commande, la bobine alimentée par -48VCC de bornier A2 (10) reçoit une excitation à partir du relais réserve SD3 du SAITEL 100 sous forme d'un signal électrique +48VCC à travers le bornier A1 (2), une fois la bobine est excitée, le contact est ouvert et par conséquent coupure de courant et effacement de défaut de verrouillage .

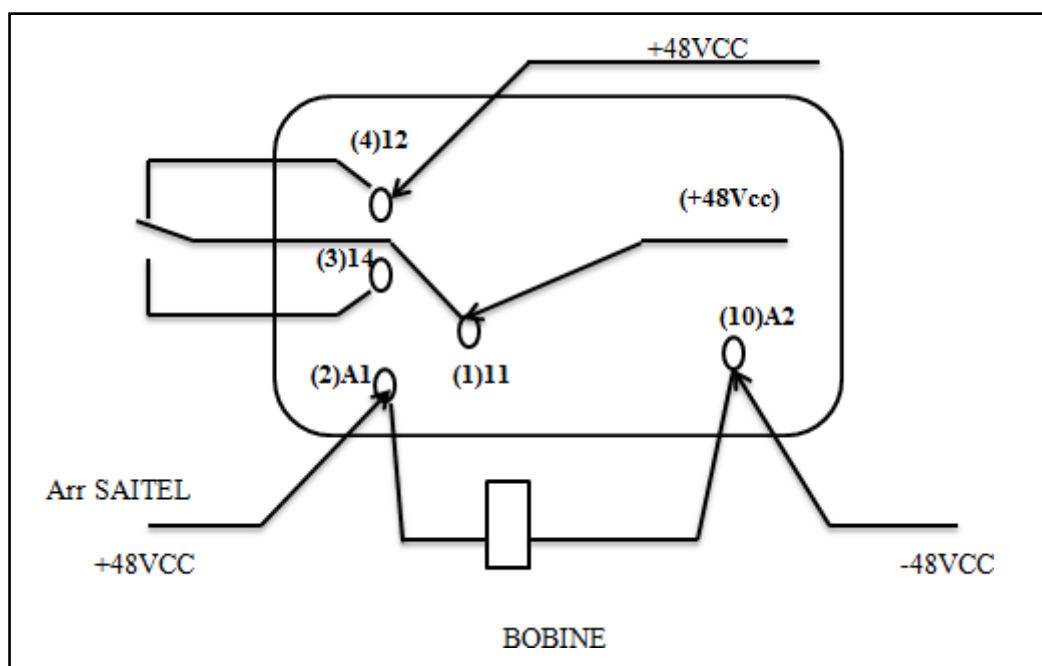


Figure 22. Schéma de câblage relais SCHRAK

La solution pour garder la durée de vie de quatre batteries et donc l'arrêt de voyant en moins de temps est de faire une remise à zéro à distance pour effacer le courant de défaut. Pour cela un relais SCHRAK MT221048 était la solution adéquate. La figure suivante indique le reliage IAT après modification ;

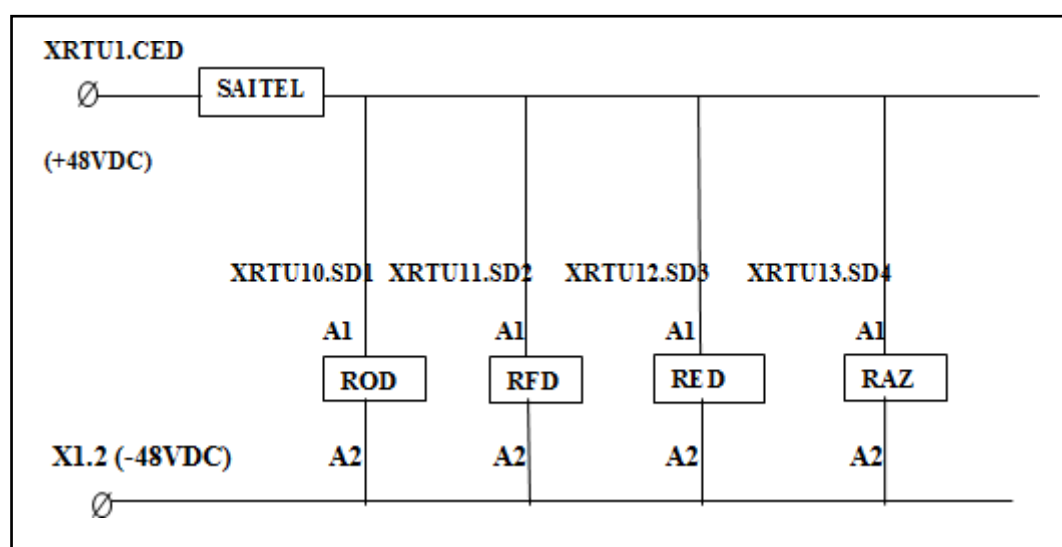


Figure 23. Reliage IAT type ABB Après modification

2.2.2. Partie base de données :

Par défaut, Saitel100 lors de génération de base de données déclare 2 sorties numérique et 2 mesure analogique et 16 entres numérique. L'outil de gestion de base de données déclare juste deux sorties numériques pourtant il avait quatre commandes alors que nous avons besoin de déclarer les deux autres par modification manuelle (les étapes de génération de la base de données sont décrites dans l'annexe 4). Les fichiers générés qu'on doit les modifier sont :

- Adq.bdp : Interface physique. Ce fichier comprend :
 - 12 télésignalisations leurs adresses sont de 1000 au 1012.
 - 02 commandes : 1,1 : « OUVRIR » son adresse 15001 et 2,2 : « FERMER » son adresse 15002
- Adq.frt : Interface logique. Ce fichier comprend 02 commandes : 1,1 : « commande » son adresse 15001 et 1,2 : « commande » son adresse 15002

On commence par déclarer deux ordres simples comme des réserves comme l'illustre les deux figures suivantes ;

```

ADQ - Bloc-notes
Fichier Edition Format Affichage ?

*SS,14
1,1,"PRJ01LPTITS0103","IAT PROJET","IAT PROJET: IAT LOCAL/TELEC",3,"CERO","UNO","SOE"
1,2,"PRJ01LPTITS0104","IAT PROJET","IAT PROJET: IAT TELEC",4,"CERO","UNO","SOE"
1,3,"PRJ01ALMITS0105","IAT PROJET","IAT PROJET: MANQUE CC COFFRET IAT",5,"CERO","UNO","SOE"
1,4,"PRJ01ALMITS0106","IAT PROJET","IAT PROJET: MANQUE CA COFFRET IAT",6,"CERO","UNO","SOE"
1,5,"PRJ01ALMITS0107","IAT PROJET","IAT PROJET: DETECT PASSAGE COURANT DEFAULT",7,"CERO","UNO","SOE"
1,6,"PRJ01ALMITS0108","IAT PROJET","IAT PROJET: BASSE PRESSION SF6",8,"CERO","UNO","SOE"
1,7,"PRJ02ALMITS0109","ATELIER","ATELIER: ALARME BATTERIE BASSE",9,"CERO","UNO","SOE"
1,8,"PRJ02ALMITS0110","ATELIER","ATELIER: MANQUE CA RTU",10,"CERO","UNO","SOE"
1,9,"PRJ02ALMITS0111","ATELIER","ATELIER: BATTERIE EN DECHARGE RTU",11,"CERO","UNO","SOE"
1,10,"PRJ02ALMITS0112","ATELIER","ATELIER: MANQUE 48VCC AUXILIAIRE",12,"CERO","UNO","SOE"
1,11,"PRJ51ALMITS0113","SAIT 01","SAIT 01: RTU LOCAL / TELE",13,"CERO","UNO","SOE"
1,12,"PRJ51ALMITS0114","SAIT 01","SAIT 01: DEFAULT MODULE ENTREES",14,"CERO","UNO","SOE"
1,13,"PRJ51ALMITS0115","SAIT 01","SAIT 01: DEFAULT MODULE DES SORTIES",15,"CERO","UNO","SOE"
1,14,"PRJ51ALMITS0116","SAIT 01","SAIT 01: DEFAULT MODULE DES ENTREES ANALOGIQUES",16,"CERO","UNO","SOE"

*SD,1
1,1,"PRJ01DISOOUV","IAT PROJET","IAT PROJET: POSITION IAT",1,2,"UNO","UNO",200

*MA,2
1,1,"RESERVA","POS","RESERVA",1,1,50
1,2,"RESERVA","POS","RESERVA",2,1,50

*OS,4
1,1,"PRJ01DISOOUVRT","IAT PROJET","IAT PROJET: POSITION IAT",1,"CERO","UNO","EJECUCION DIRECTA"
1,2,"PRJ01DISOOFERME","IAT PROJET","IAT PROJET: POSITION IAT FERMER",2,"CERO","UNO","EJECUCION DIRECTA"
1,3,"RESERVA","POS","RESERVA",3,"CERO","UNO","EJECUCION DIRECTA"
1,4,"RESERVA","POS","RESERVA",4,"CERO","UNO","EJECUCION DIRECTA"

*CI,0
*PT,0
*OC,0
*OD,0
*OR,0
  
```

en doit declarer deux ordre simple comme des reserves

Figure 24. Fichiers de base de données modifiés manuellement (1)

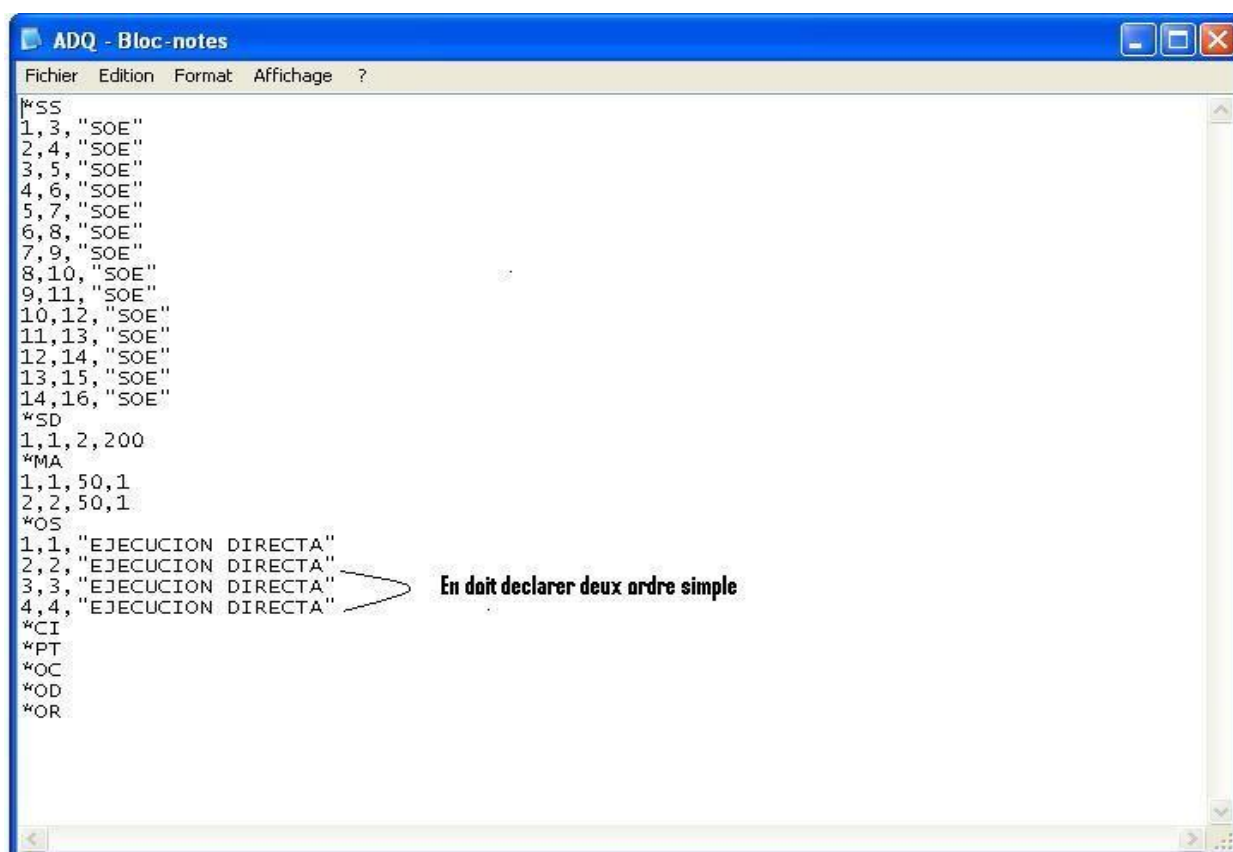


Figure 25. Fichiers de base de données modifiés manuellement (2)

3. Intégration de la carte programmable

3.1. Choix de la carte

Après une étude des différentes cartes programmables existantes sur marché à savoir les cartes Arduino et les cartes Raspberry PI, nous avons résumé les caractéristiques communes de chaque famille dans le tableau suivant ;

Raspberry PI	Arduino
Ordinateur : Utilise un système d'exploitation, pas de temps réel	Carte à base d'un microcontrôleur : Pas de système d'exploitation, Temps réel
Autonome	Semi-autonome: a besoin d'un ordinateur pour le programmer

Quelque entrées/sorties numériques (8 par défaut), aucune analogique	Beaucoup d'entrées/sorties numériques (14 à 54), beaucoup d'entrées analogique (6 à 16)
Sortie audio, vidéo, E/S USB, connecteurs spécialisé écran /caméra, réseau	Possibilité d'extension via des cartes filles
Communauté importante	Communauté importante

Tableau 4. Arduino VS Raspberry Pi

Selon ce tableau, les deux cartes semblent concurrentes mais elles ne répondent pas aux mêmes besoins, donc elles peuvent être complémentaires.

En conclusion, Raspberry pi est plus adapté aux projets qui se basent sur des programmes complexes comme le traitement d'image, reconnaissance vocale. Tandis qu'Arduino est plus adapté au control du matériel (bas niveau) comme le contrôle de capteurs. Donc, la solution idéale dans notre cas est d'utiliser l'Arduino

3.2. Mise en place de la carte

Pour la mise de notre système, nous avons connecté la carte Arduino au module wifi et aux quatre relais pour s'assurer de la communication comme le montre la figure suivante. Par la suite, nous avons intégré le tout avec le coffret mixte.

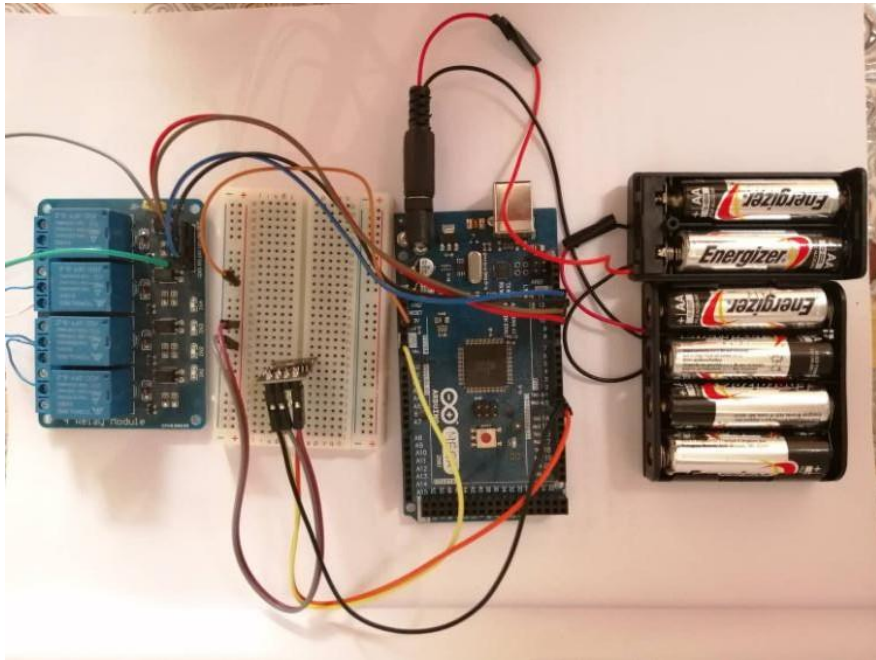


Figure 26. Schéma de montage de la carte Arduino

4. Développement de l'application mobile

3.1. Etapes de réalisation

La plupart des agents à la STEG ont des smartphones avec un noyau Android. Pour ce faire, notre choix s'est porté sur l'environnement de développement Android Studio.

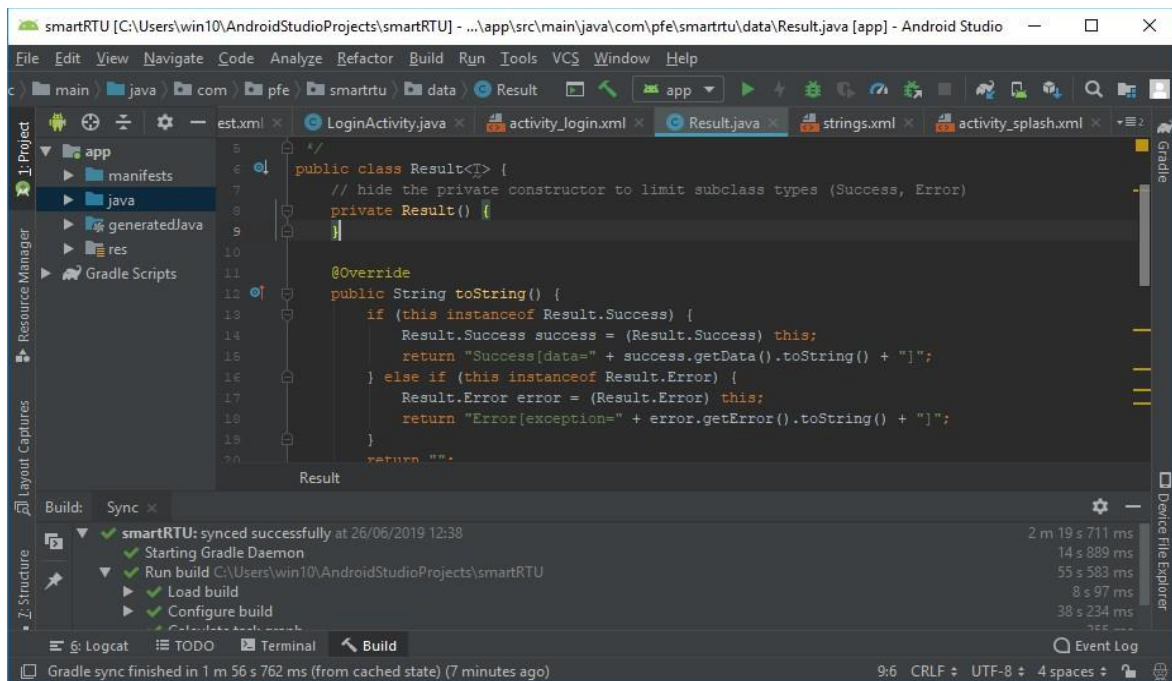


Figure 27. Environnement de développement Android Studio

Une fois l'environnement de développement est installé, nous avons suivi l'architecture de l'application définie dans la partie conception pour coder nos interfaces.

3.2. Présentation des interfaces

L'utilisation de l'application est spécifique pour les utilisateurs autorisés. La première interface donc est celle de l'authentification comme l'indique l'interface suivante ;

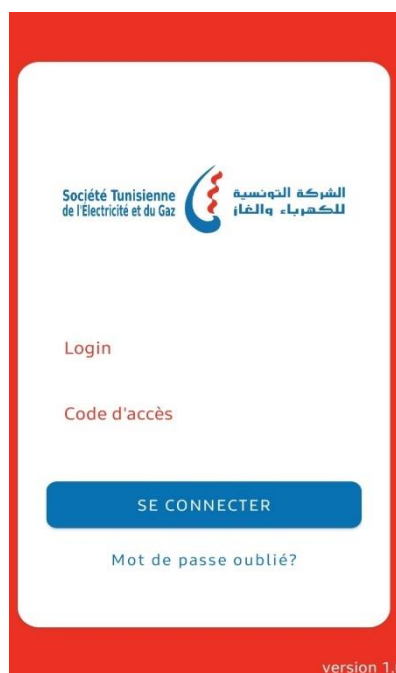


Figure 28. Interface d'authentification

Une fois connecté, l'utilisateur choisit la fonction à exécuter à partir d'un menu citant les principales fonctionnalités de l'application. L'interface suivante illustre ces menus ;

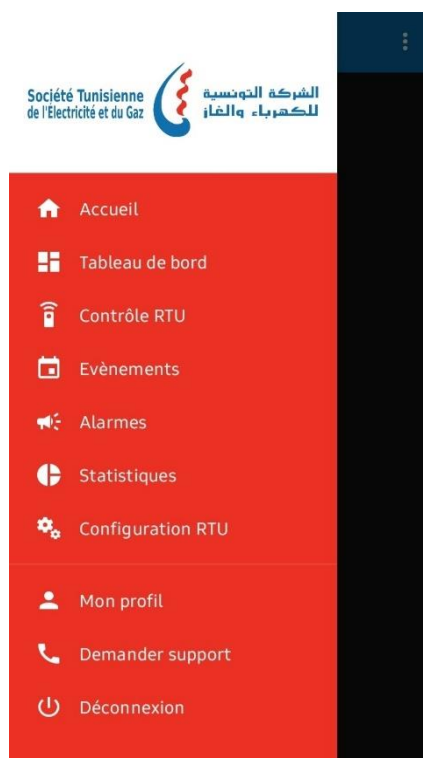


Figure 29. Menu principal de l'application

Cette interface liste les différentes fonctionnalités de l'application.

5. Développement de l'application web

5.1. Etapes de réalisation

La première étape de développement de l'application web est le choix des outils nécessaire à la réalisation.

Pour les langages nous avons choisi ;

- **PHP** : C'est un langage scripts libre de programmation pour produire les applications web dynamiques via un serveur HTTP.
- **MySQL** : MySQL est un système de gestion de base de données « SGBD » gratuit. C'est le logiciel de gestion de base de données les plus utilisés. Le couple PHP/MySQL est très utilisé pour le développement des sites web et par la majorité des hébergeurs web.
- **HTML5/CSS** : Html « HyperText MarkupLanguage 5 » est un langage de présentation, Il fonctionne sur le principe de balises imbriquées pour permettre de présenter l'information au public. Il est simple dont son usage d'utilisation et compatible depuis les différents navigateurs web. Le langage CSS « Cascading Style Sheets » est utilisé pour définir le style des pages d'un site, par exemple les couleurs, les fonds, dimension... Plus concrètement, c'est une feuille de style d'extension. ccs,

Pour implémenter notre solution nous avons choisi comme outils logiciels;

- XAMPP : c'est un ensemble de logiciels permettant de mettre en place facilement un serveur Web local. Il s'agit d'une plateforme de développement Web, qui permet d'installer simplement un espace de développement permettant de faire fonctionner d'une manière locale des codes PHP.
- Notepad++ ; Notepad++ est un éditeur de texte libre générique qui prend en charge plusieurs langages.

5.2. Présentation des interfaces

Une fois le code est développé, l'exécution de notre projet dans sa version web lance en premier lieu une interface de connexion comme l'indique la figure suivante. Cette interface permet à l'utilisateur de s'authentifier, il doit entrer son « nom d'utilisateur » et son « mot de passe » pour accéder au menu principal de l'application.



Figure 30. Page d'accueil de l'application web

L'ajout des comptes utilisateurs se fait par l'administrateur du site. Et une fois connecté l'utilisateur de type agent peut gérer son espace qui se présente comme l'indique la figure suivante ;

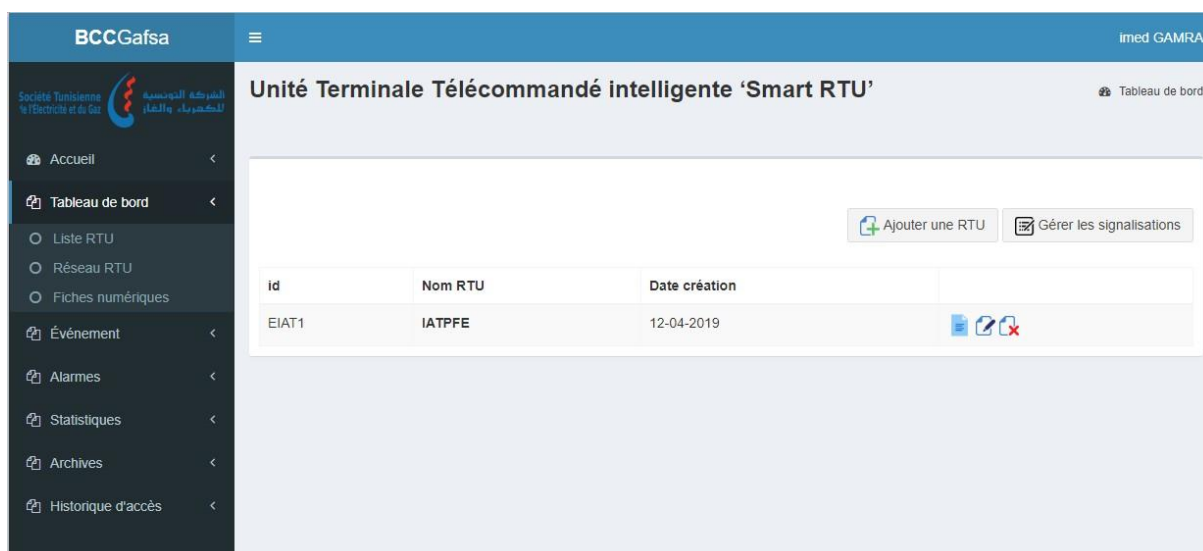


Figure 31. Interface principale du site

Cette interface permet à l'utilisateur de visualiser la liste des RTU dans un tableau ou une carte interactive qui affiche l'état de réseau. On parle ici de menu « Réseau RTU ».

Le menu « Fiche numérique » d'une RTU, donne l'état des différents signaux provenant de la IAT sélectionnée comme l'illustre la figure suivante ;



Figure 32. Interface « fiche numérique »

Le menus « Événement » liste tous les évènements sur le réseau. Dans ce projet, nous avons travaillé avec un seul point donc on parle ici d'une interface similaire à celle de fiche numérique.

Le menu « Alarmes » filtre les alarmes des événements et les affiche dans une liste comme la suivante ;

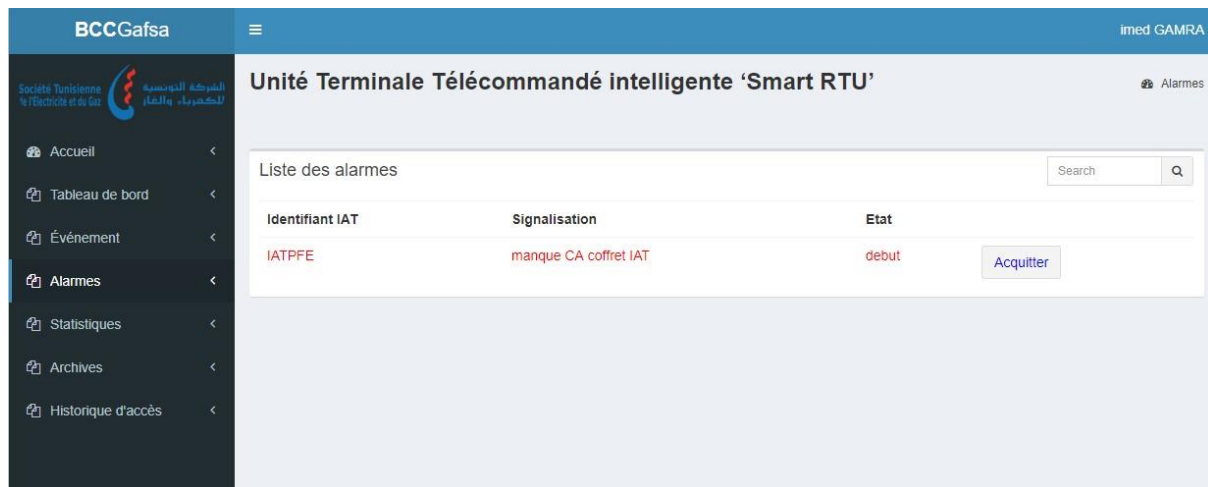


Figure 33. Interface liste des alarmes

Le menu « Statistiques » donne des chiffres clés sur le réseau et le menu « Archives » contient les archives. Le menu « Historique d'accès » donne une traçabilité sur l'accès au système.

6. Conclusion

La partie de réalisation donne une idée plus claire des tâches qui sont réalisées dans ce site par la présentation des interfaces graphiques. Enfin, avec ce chapitre, et malgré toutes les difficultés rencontrées, nous avons réussi à réaliser presque la totalité de fonctionnalités demandées.

Conclusion générale

Ce rapport résume les différentes étapes de conception et d'implémentation d'une unité terminale télécommandée 'RTU' à partir de trois coffrets classiques présents au niveau de l'IAT. Nous avons commencé par une étude préalable pour présenter le cadre du projet et l'état de l'art. Puis, nous nous sommes lancés dans la phase d'identification des besoins fonctionnels et non fonctionnels. Ensuite, nous avons décrit les différentes phases de réalisation et ce après une conception détaillée de système à mettre en place.

Le choix d'une telle application n'est pas anodin du fait que les Smart City et les systèmes commandés à distances sont un sujet d'actualité. En fait, toute la richesse fonctionnelle de cette solution réside dans le fait de sa simplicité et la facilité d'utilisation.

Notre programme est opérationnel bien que certains ajustements soient possibles et pas certainement nécessaires.

Aussi, le stage s'est déroulé en respectant les temps fixés par le planning prévisionnel, et ce, malgré un retard pris lors de la réalisation majoritairement dû à la non disponibilité de certains matériels.

Webographie




- [1] <http://matechnologie.com/spip.php?article509> [Consulté le 23/02/2023]
- [2] <https://lehollandaisvolant.net/tuto/computer/> [Consulté le 23/02/2023]
- [3] <http://blogpeda.ac-poitiers.fr/techno4e-jean-mace/2013/11/16/carte-electronique-programmable-arduino-pour-les-curieux/> [Consulté le 02/03/2023]
- [4] <https://www.wemustbegeeks.com/beginners-guide-arduino/> [Consulté le 02/03/2023]
- [5] <http://blewando.dlinkddns.com/elv/Promo2018/th14/pag6.html> [Consulté le 06/03/2023]
- [6] http://pf-mh.uvt.rnu.tn/621/1/Conception%2C_d%C3%A9veloppement_et_int%C3%A9gration_d%E2%80%99une_application_embarqu%C3%A9_de_t%C3%A9l%C3%A9chargement_des_applications_Andro%C3%AFd_Ftab_Stor.pdf [Consulté le 06/03/2023]
- [7] <http://www.redohm.fr/2014/12/arduino/> [Consulté le 06/03/2023]
- [8] <https://www.fais-le-toi-meme.fr/fr/electronique/materiel/esp8266-arduino-wifi-2-euros> [Consulté le 02/03/2023]
- [9] http://www.journaldunet.com/solutions/0302/030206_wifi.shtml [Consulté le 02/03/2023]

Annexes

Annexe 1 : Liste des composants utilisés

Nom	Définition	Rôle
Disjoncteur  Disjoncteur 48V 220V	Un disjoncteur est un appareil mécanique de connexion capable d'établir, de supporter et d'interrompre un courant dans un circuit électrique.	1 Disjoncteur : permet le passage de 220V pour alimenter le chargeur. 2 Disjoncteurs : permet le passage de 48V vers le coffret de commande et RTU.
Chargeur série EC5-200 	Cette série comprend 3 modèles de chargeur d'alimentation qui, en présence de la tension du secteur, fournissent une tension stabilisée tout en procédant à la charge contrôlée de la batterie.	<ul style="list-style-type: none"> - Chargement de la batterie en courant constante. - Alimenter le coffret de bloc IAT de 48V. - Convertir la tension de 220V à 48V
Contacteur 	Un contacteur est un appareil électrotechnique destiné à établir ou interrompre le passage du courant, à partir d'une commande électrique ou pneumatique.	1 contacteur pour l'ouverture du bloc IAT 1 contacteur pour la fermeture de bloc IAT
Relais	Il s'agit d'un ensemble de bobine et	Relai d'ouverture et relai de fermeture

	<p>Fil de contact de cuivre et alimentation 48 v</p> <p>Deux types : relais contact ouvert</p> <p>Relais contact fermé</p>	<p>Localement.</p> <p>Relai d'ouverture et relai de fermeture à distance.</p> <p>Relai passage courant de défaut.</p> <p>Relai effacement de défaut.</p> <p>Relai de remise à zéro.</p>
<p>Détecteur passage courant de défaut</p> 	<p>Le relais N-DIN-F sont équipés d'une unité ampère métrique biphasée pour la mesure des courants de ligne et d'une unité homopolaire pour la mesure des courants de fuite à la terre.</p>	<ul style="list-style-type: none"> - Protection des transformateur HTA/BT - Protection des départs câbles et ligne - Protection des moteurs de petites puissances.
<p>Convertisseur</p> 	<p>Convertisseur de tension de 24V à 12V</p>	<p>Alimentation le radio.</p>
<p>Radio</p> 	<p>Radio de communication avec antenne de type yagui ou omni</p>	<p>Réception et transmission des paquets analogiques entre le BCC et le point télécommandée</p>

SAITEL	Carte programmable à base d'un microcontrôleur dont en doit charger un tel base des données permettant de communiquer avec la partie logique (radio) et la partie Physique câblage interne	Contrôle à distance et acquisition des données
<p>Bouton poussoir</p> 	Un interrupteur poussoir, ou bouton poussoir, est un interrupteur électrique dont le bouton revient à sa position initiale après avoir été actionné.	<p>1 bouton poussoir marche.</p> <p>1 bouton poussoir arrêt.</p>
<p>Sélecteur</p> 	Sélecteur de choix	Sélection entre télé ou local
<p>Batterie</p> 	Batterie à courant continu de 12 V	Alimentation du coffret et commande et RTU

Annexe 2 : SAITEL 100

Portée

Le poste terminal de télé conduite Saitel-100 a pour objet de contrôler un élément de coupure (interrupteur ou sectionneur), aussi bien à haute qu'à moyenne tension, ainsi que des transformateurs et des éléments auxiliaires, avec un équipement électronique unique, identique pour tous.

Le moniteur de maintenance permet de configurer certains paramètres tels que la définition des liaisons, les temps de pulsation des sorties numériques et les temps de filtrage des entrées numériques. De plus, il permet de surveiller les signaux d'entrée et les positions de mémoire et d'exécuter des commandes.

Caractéristiques techniques

Le poste terminal Saitel-100 est basé sur un microcontrôleur Motorola modèle MC68HC111A1FN, en mode « développé » (expanded-multiplexed), qui requiert l'utilisation d'éléments supplémentaires tels que des mémoires et des portes d'entrée/sortie.

Les caractéristiques de base de cette unité de commande à distance, avec les éléments supplémentaires cités, sont :

- 32 Kb de mémoire EPROM pour programmes d'application
- 8 Kb de mémoire RAM pour variables et tableaux
- 512 bytes de mémoire EEPROM
- Canal série asynchrone (SCI)
- 4 générateurs de rythmes de 16 bits
- Watch-dog

L'unité de commande incorpore en plus une série de dispositifs électroniques auxiliaires (horloge, surveillance de tension, circuit de reset, etc.).

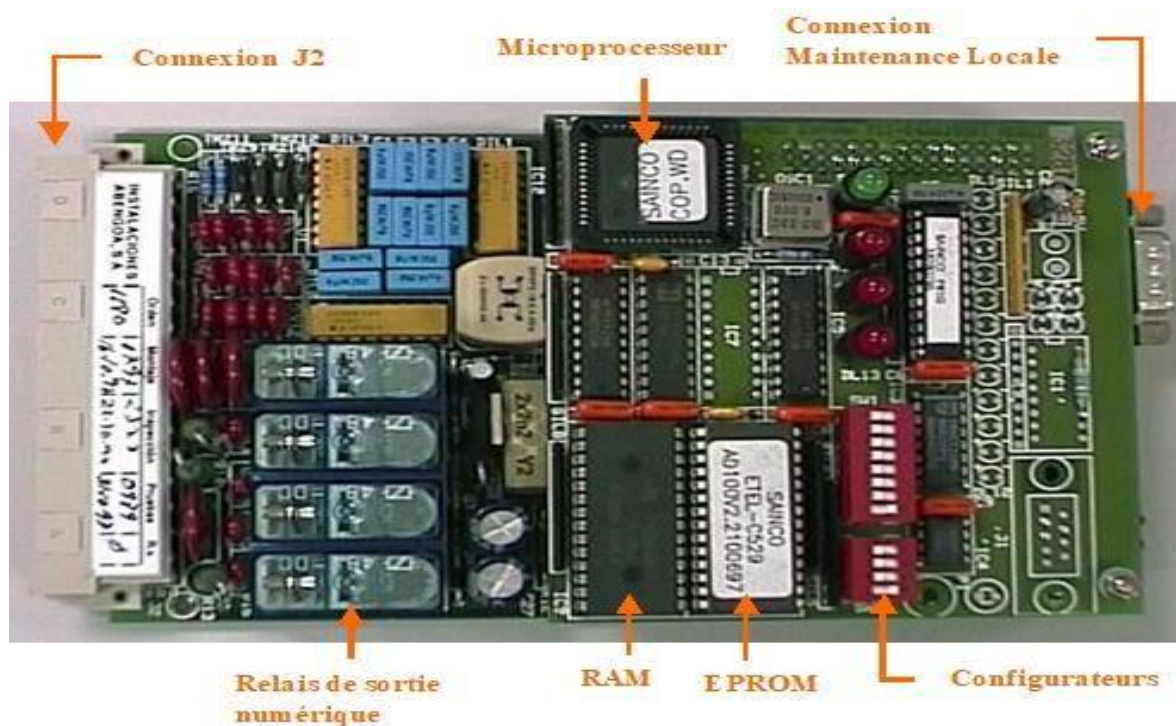


Figure 3.4 : Poste terminal SAITEL100



Bloc fonctionnel

La RTU Saitel-100 est composée des blocs fonctionnels suivants :

- Unité de commande à distance. Module d'acquisition et de communications basé sur un microprocesseur HC11 de 8 bits.
- Signaux d'entrée et de sortie. La capacité en signaux d'entrée et sortie bruts est de 16 entrées numériques, 4 sorties numériques et 2 mesures analogiques bipolaires ou 4 unipolaires.
- Communications : la RTU Saitel-100 dispose de 4 canaux série asynchrones. Deux canaux sont pour les communications avec les couches supérieures et les deux autres pour les communications avec les niveaux inférieurs. Les signaux sont RS-232, y compris ceux de commande de modem. Elle possède en plus un canal série asynchrone par signaux RS-232, utilisé pour la maintenance locale. Le protocole utilisé est IEC 870-101 esclave.
- Alimentation en énergie : L'alimentation électrique se fait par courant continu +48V, +20/-10% (Versions en 24 VCC et 12 VCC). La consommation maximum est de 5 W [10].

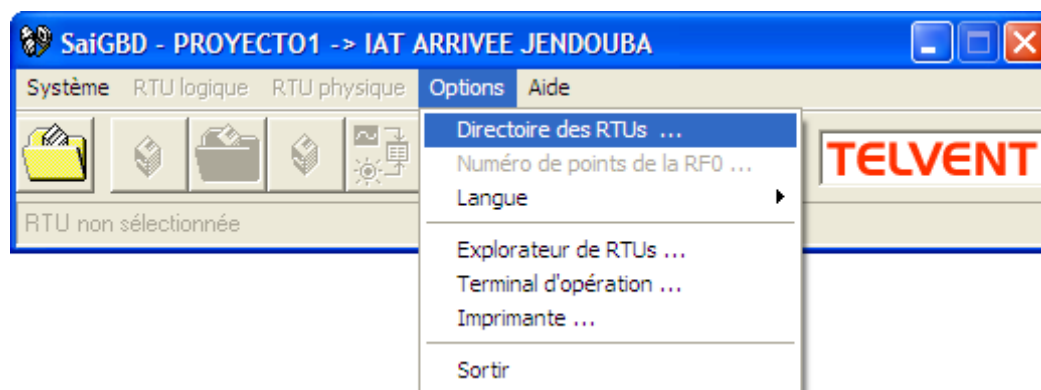
Annexe 3 : Configuration et chargement de la base des données

Téléchargement de base de données

Ouvrir SAIGBD

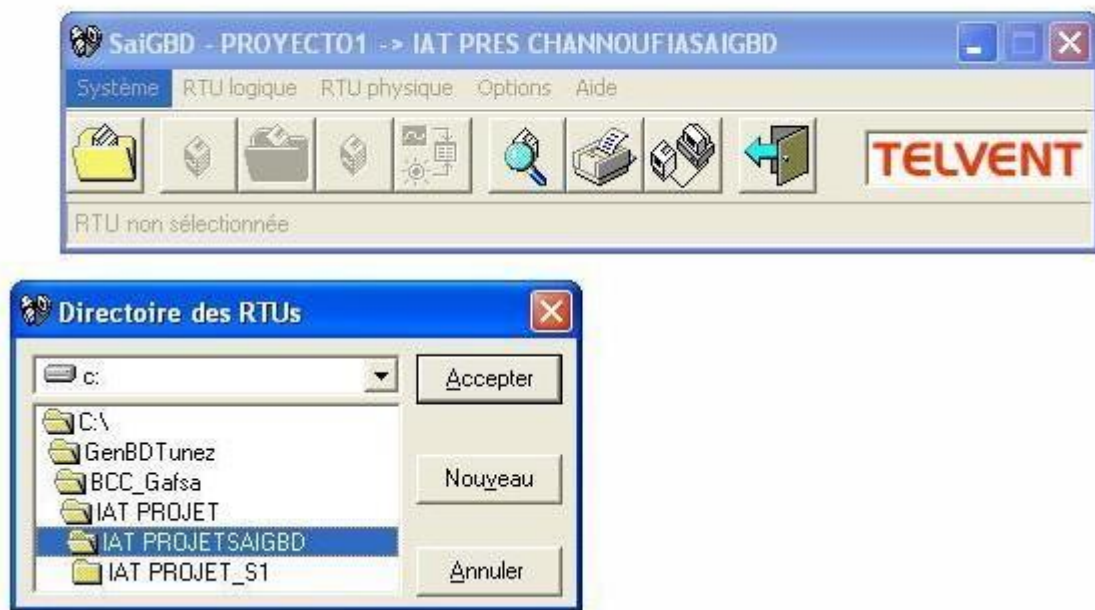


Dans le menu OPTION choisir DIRECTOIRE DES RTUs



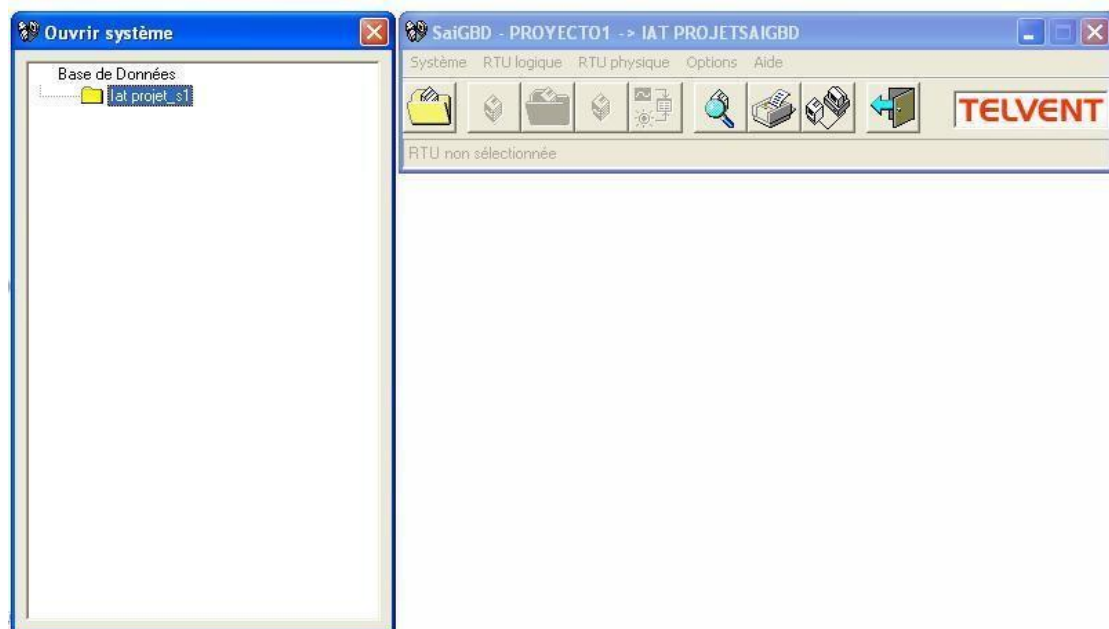
Choisir le dossier GENBDTUNEZ /BCC GAUSA /IAT PROJET

Sélectionner IAT PROJET



Double cliquer sur IAT PROJETAIGBD puis Accepter

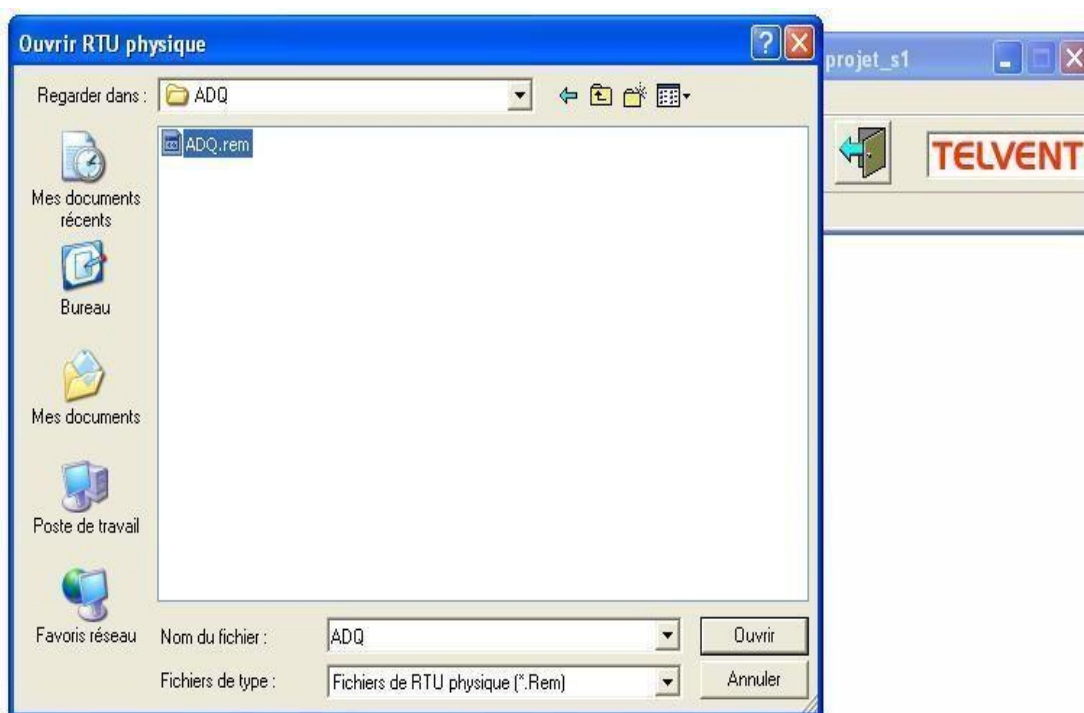
Dans le menu système sélectionner ouvrir



Dans la boîte « ouvrir système » double cliquer sur Iat projet-s1



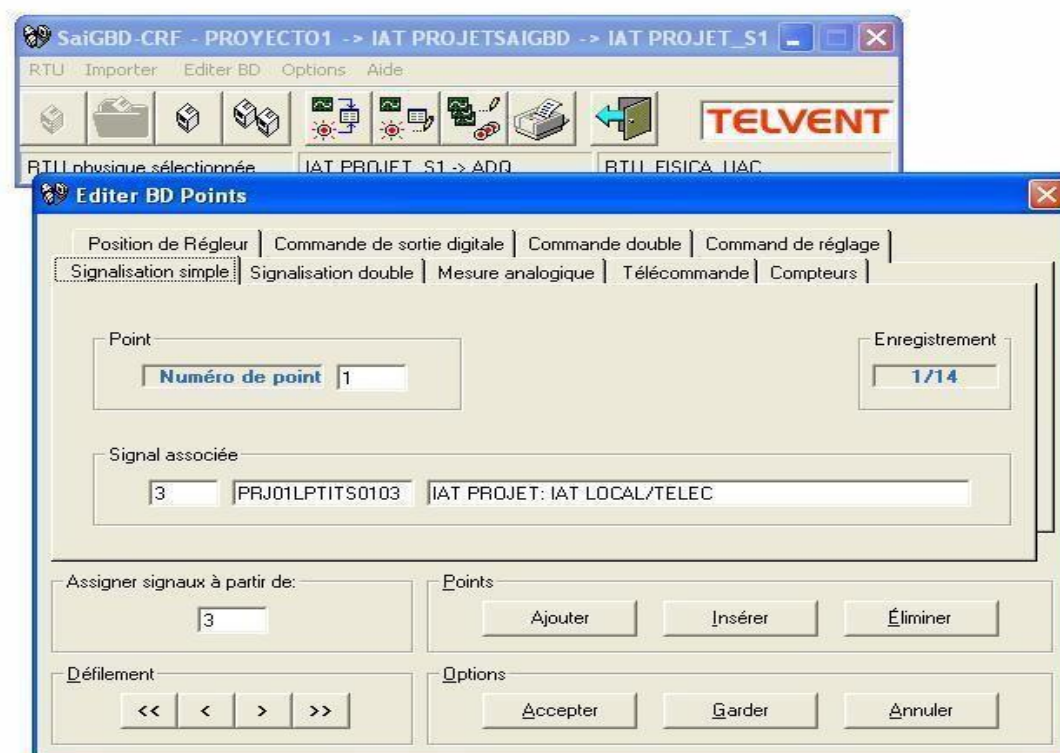
Dans la boîte de dialogue ouvrir RTU physique ouvrir le dossier ADQ puis double cliquer sur le fichier ADQ. Rem (une nouvelle fenêtre apparaît)



- Dans cette nouvelle fenêtre choisir EDITER BD /POINTS /ASSIGNER POINTS



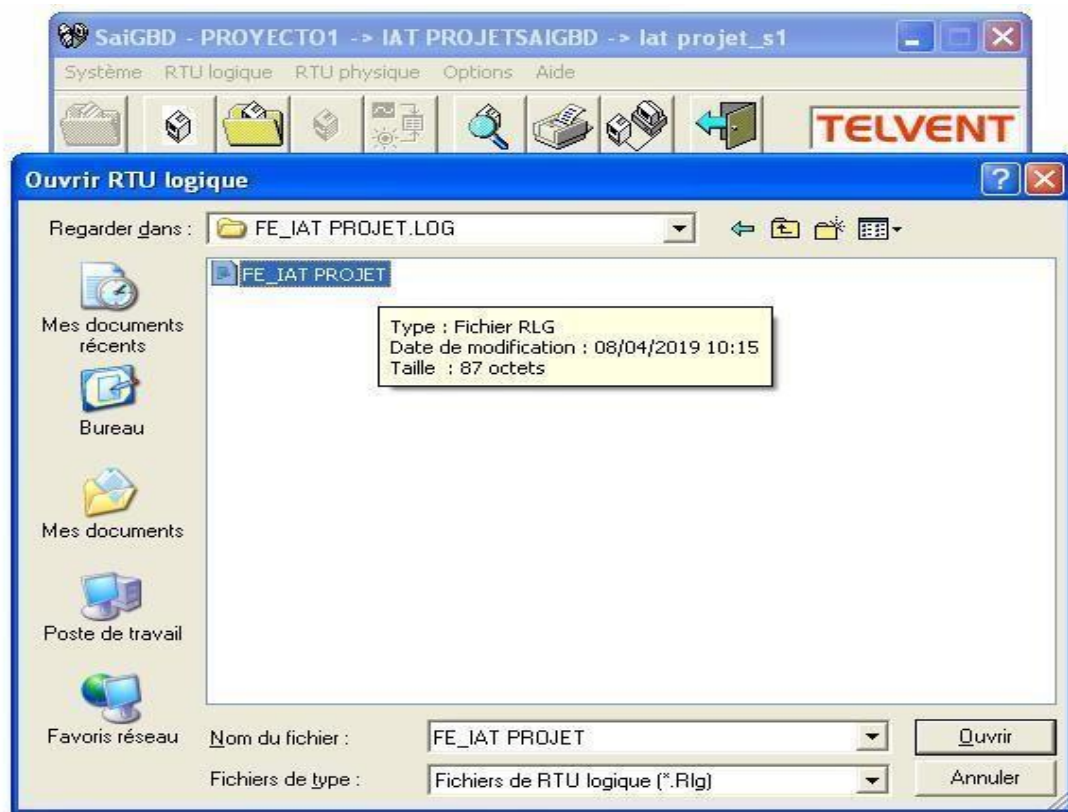
Appuyer sur GARDER /ACCEPTER



Cliquer sur le bouton « sortir »



Dans le menu RTU LOGIQUE choisir « OUVRIR »



Dans la boîte « ouvrir RTU logique » choisir FE-IAT PROJET. Log

Nb, - double cliquer sur le fichier Fe-Iatprojet.rlg

- choisir RTU logique /propriétés

Dans la fenêtre propriété vérifier :

L'adresse du point (8 projet) convertir en binaire pour configurer les Switch de la carte saitel

8 dec. = 00001000bin

Ligne « 0 »

Profil : PERIFEL SAINCO UNBALANCE

Appuyer sur accepter

Propriétés

RTU

Numéro: 1

Nom: FE_IAT PROJET

Description: Remota Logica IAT PROJET

Type: RTU_LOGICA_IEC101

Système: IAT PROJET_S1

Communications

Profil: PERFIL_SAINCO_UNBALANC

Ligne: 0

Adresse: 8

☐ Local

☒ Synchronisme

Annuler Accepter

Points

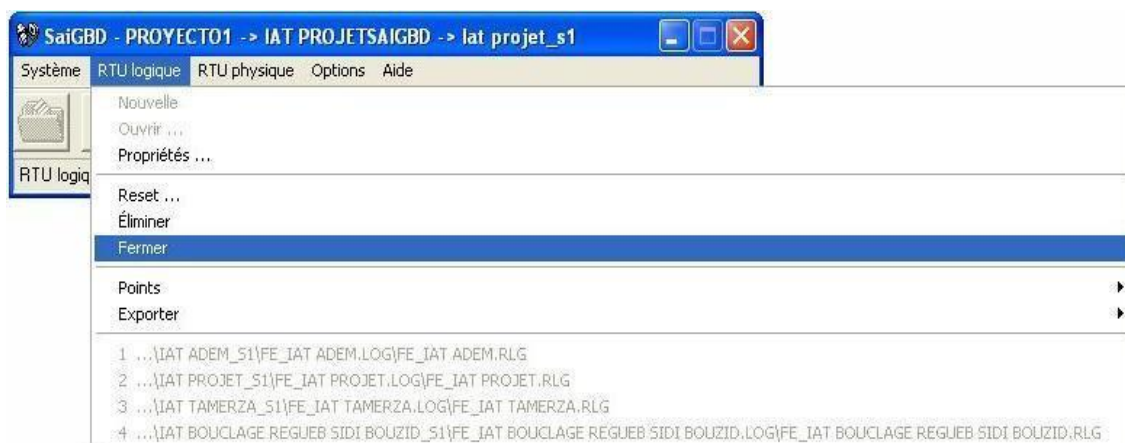
SS	SD	OS	OD	ME	SE	CB	OR	OB32	B32	OC
14	1	4	0	0	0	0	0	0	0	0

Annexer fichiers

Ajouter

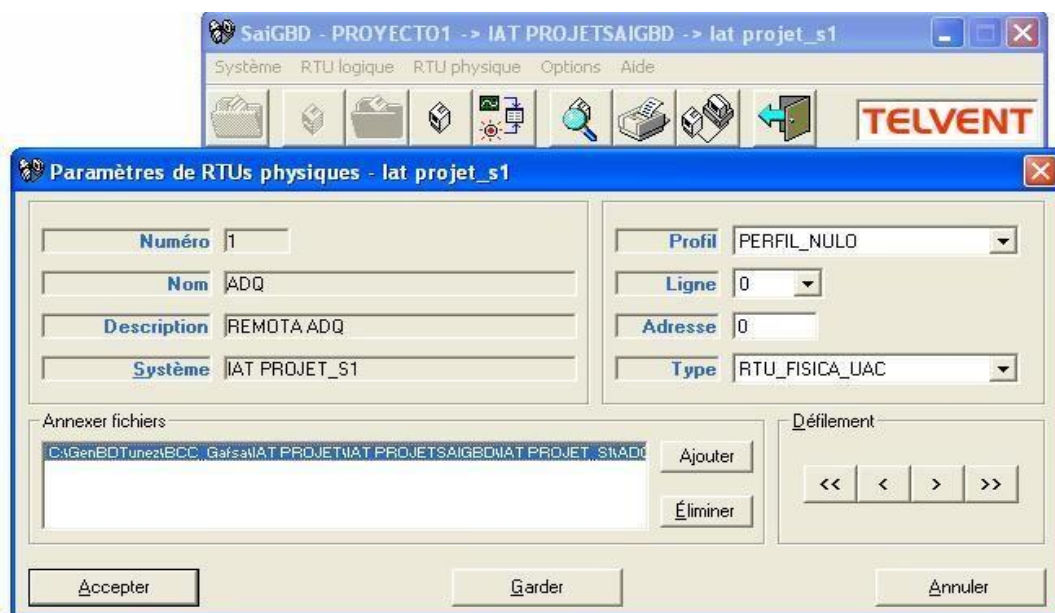
Éliminer

Fermer RTU logique.



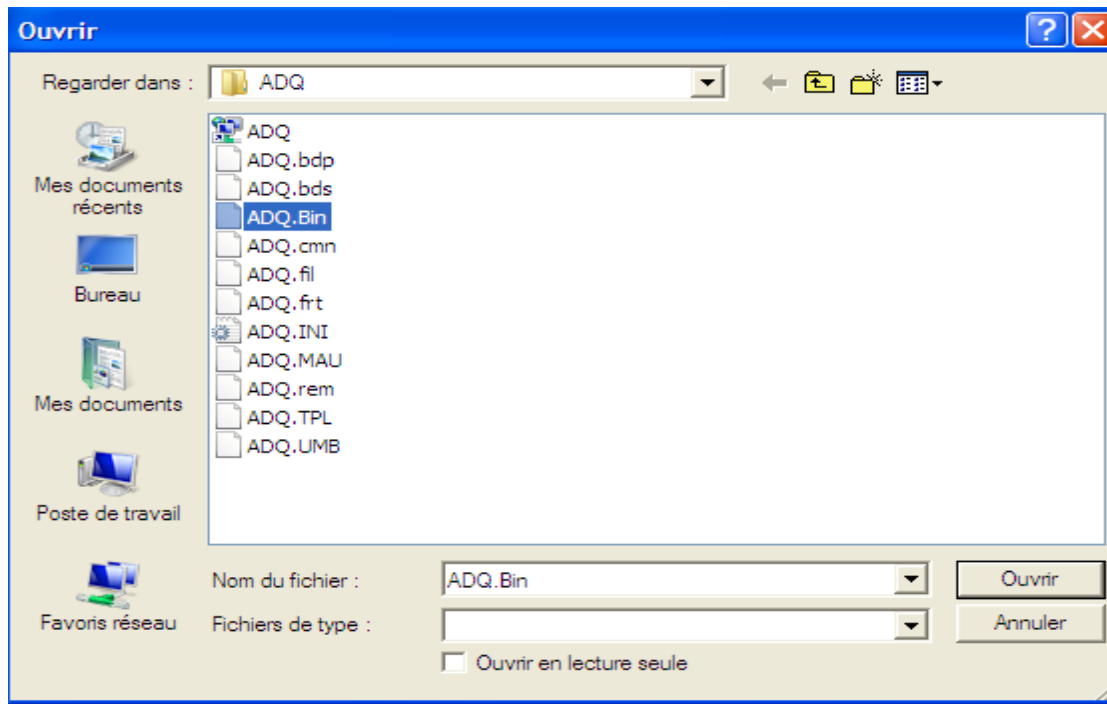


Dans le menu système choisir **paramètre des RTUs physique**



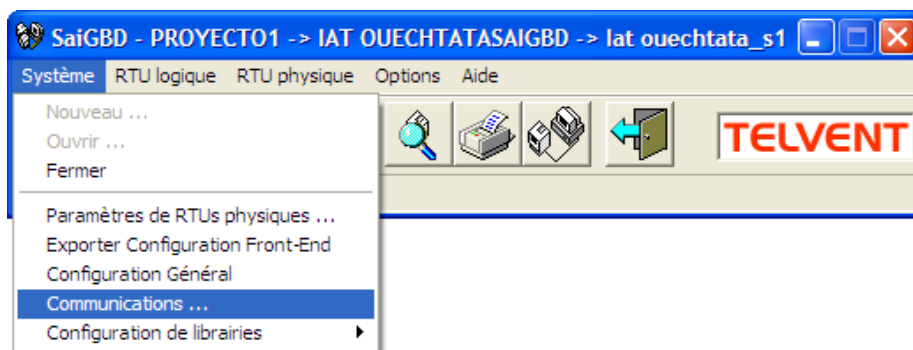
Appuyer sur le bouton éliminé puis ajouter

Double cliquer sur le fichier ADQ.bin

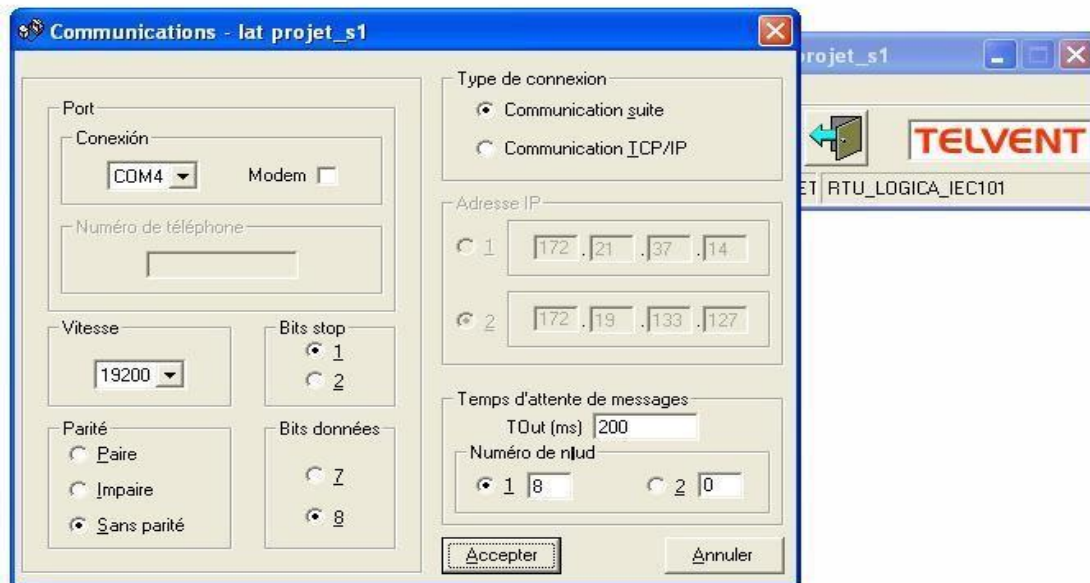


Appuyer sur garder accepter.

Choisir dans le menu Système l'option « **communication** »

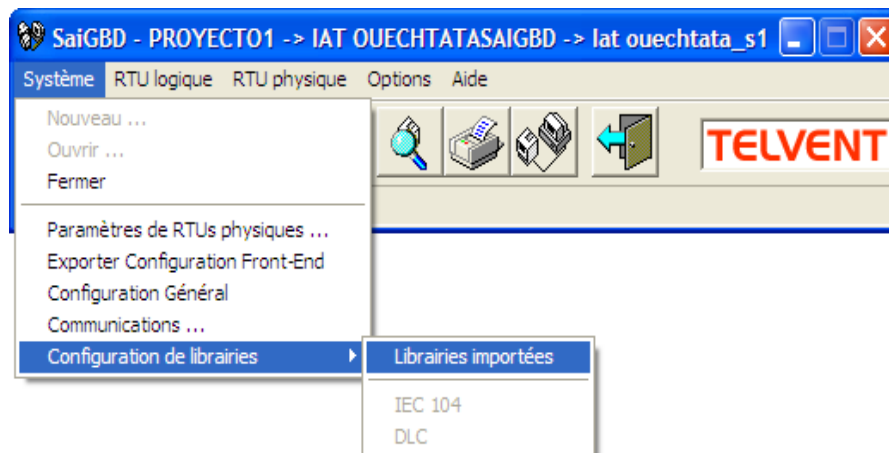


Charger l'adresse de l'IAT dans la case numéro du nœud. (8) et configurer le port de communication du PC.

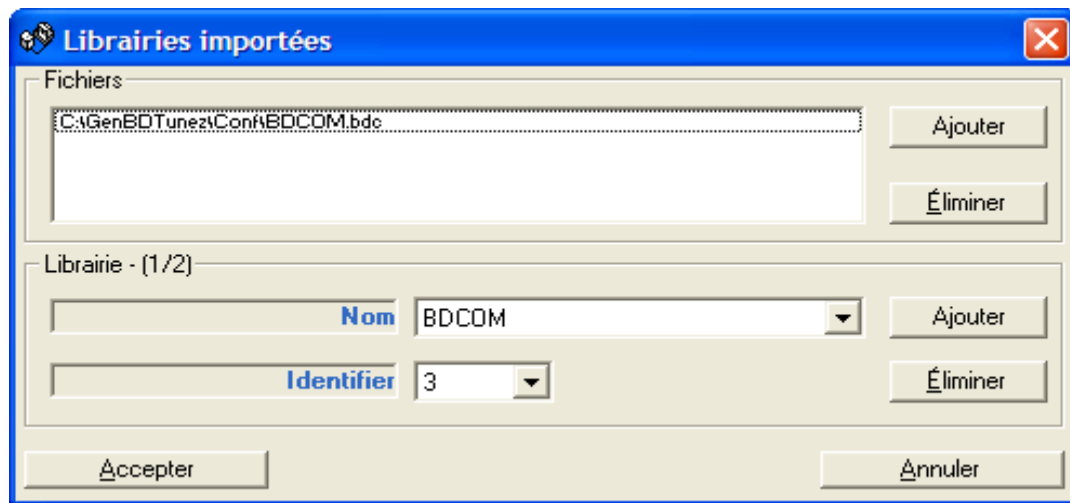


Accepter.

Choisir « configuration de librairies / librairie importées



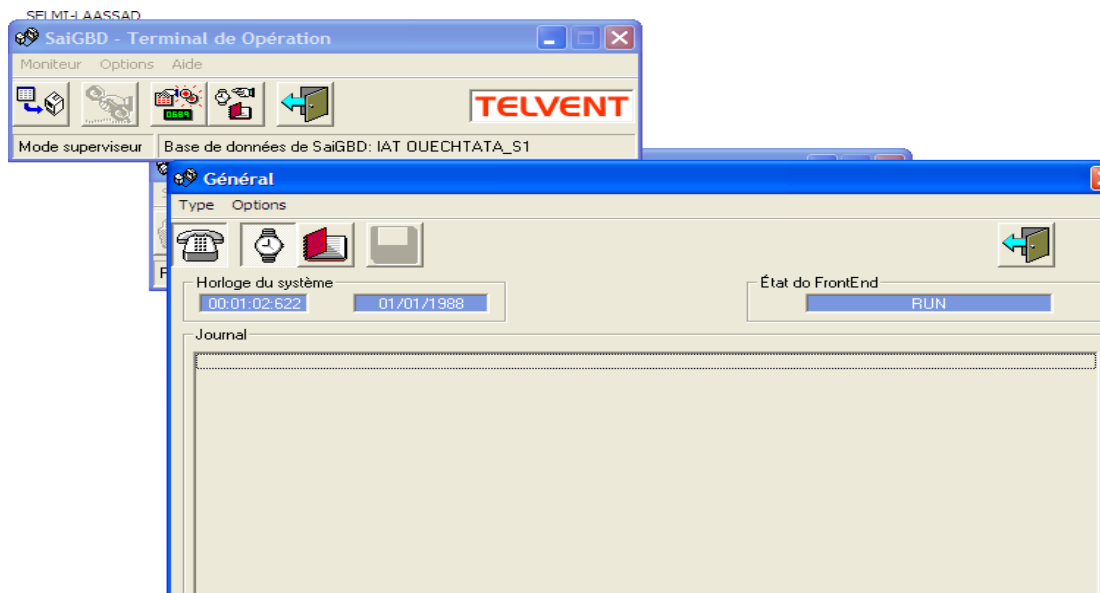
Librairies importé : choisir BDCOM et l'identifiant 3 puis accepter.



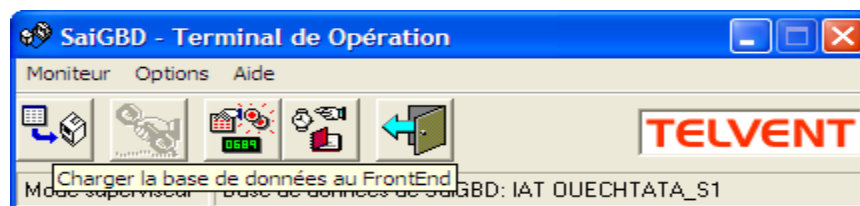
Appuyer sur le bouton terminal d'opération



Appuyer sur moniteur général pour vérifier l'horloge du système. Et état de frontend RUN



Charger la base de données du point



Charger la base de données du saitel et confirmer par accepter.

