

RAPPORT DE STAGE

Mise en œuvre d'un tableau électrique de test piloté par Raspberry PI

Année 2020 – 2021

Date du stage : 12/04/2021 – 02/07/2021

Jalil BENHIBA

DUT GEII – Aix-Marseille Université

SmartFuture

Tuteur de l'entreprise : **BERTRAND Paul**

Tuteur académique : **MOREAU Mathieu**

Table des matières

Remerciements	4
Introduction.....	5
I/ Présentation de l'entreprise	6
1. Présentation générale	6
2. Quelques chiffres	6
3. Recherche & Développement	6
II/ Le Stick'n Sense.....	7
1. Fonctionnement	7
2. Problématique.....	7
III/ Présentation du projet.....	8
1. Résumé de la mission	8
2. Cahier des charges	8
3. Planning Prévisionnel	9
IV/ Mes travaux :	9
1. Mes recherches	9
1.1) Présentation Modbus	9
1.2) Les modes de transmission	9
1.3) Les Supports physiques de transmission	10
1.4) Les variations du protocole.....	11
2. Création du générateur de scénario en CSV	11
3. Configuration du matériel	13
3.1) Liste du matériel	13
3.2) Configuration	14
4. Programmation	15
4.1) Les librairies	16
4.2) Structure du programme CSV	17
4.3) Structure du programme relais.....	18
4.4) Structure du programme complet	19
5. Montage de l'armoire	20
V/ Conclusion	23
1. Bilan technique.....	23
2. Bilan personnel.....	23

Index des Illustrations	24
Bibliographie	25
Annexes	26
Résumé & Abstract.....	27
1) Résumé	27
2) Abstract	27

Remerciements

Tout d'abord, je tiens à adresser mes remerciements au département GEII de l'IUT Aix-Marseille ainsi que tous ses enseignants pour toutes les choses qu'ils ont pu nous apporter durant ces deux dernières années.

Je remercie aussi le département de nous avoir laissé accès à des salles ainsi que du matériel afin que notre stage se déroule dans les meilleures conditions.

Je remercie particulièrement :

Monsieur **VAUCHE, professeur du département**, qui en plus de nous avoir permis d'avoir effectué ce stage, était là dès lors que nous avions une question ou que nous étions bloqué. Je le remercie également de sa confiance quand il m'a sélectionné pour ce stage.

Paul BERTRAND, Président de SmartFuture, d'avoir accepté de nous laisser une chance d'effectuer notre stage dans son entreprise, de nous avoir fait confiance, ainsi que de nous avoir consacré du temps lorsque nous en avons besoin.

Monsieur MOREAU, notre tuteur académique qui va lire ce rapport de stage.

Samuel MIGNUCCI, pour le travail qu'il a fourni dans la réalisation de ce projet.

Introduction

Afin de valider le DUT en Génie électrique et Informatique Industrielle, un stage en entreprise doit être effectué. Le stage doit durer de 8 à 12 semaines et il est là pour nous faire découvrir le monde de l'entreprise, nous faire mettre en pratique nos acquis académiques et ainsi nous faire acquérir une expérience professionnelle.

J'ai effectué mon stage avec la société SmartFuture qui est une entreprise qui aide les Directions générales d'autres sociétés à « maîtriser leurs évolutions vers le plus de technologie ».

La société SmartFuture développe des capteurs Stick'n Sense, ces capteurs mesurent la puissance apparente des disjoncteurs d'un tableau électrique. Notre objectif à nous était de câbler un tableau électrique et réaliser un programme qui joue un scénario d'allumage des relais et mesure la puissance apparente des disjoncteurs. Notre travail servira donc à mesurer la précision des capteurs que développent SmartFuture.

Tout d'abord, je présenterais la société avec qui j'ai effectué mon stage SmartFuture ainsi que les projets sur lesquels ils travaillent. Ensuite, je présenterais le projet sur lequel nous avons travaillé avec mon binôme, les tâches qui m'ont été attribuées. Pour finir, je ferais part de mon ressenti sur ce stage avec un bilan personnel et technique.

I/ Présentation de l'entreprise

1. Présentation générale

SmartFuture est une société fondée le 1^{er} Avril 2011 par Monsieur **Paul BERTRAND**. Le monde d'aujourd'hui évolue toujours plus rapidement ce qui impose aux entreprises de suivre cette évolution vers le toujours plus technologique. SmartFuture a été créé afin d'aider les Directions Générales des entreprises à effectuer le pas vers l'évolution technologique en offrant ses compétences en « ingénierie de l'innovation technologique ».

2. Quelques chiffres

SmartFuture, basée à PIERREFEU-DU-VAR (83390), est spécialisée dans le secteur du conseil. Cette entreprise est une SAS familiale et compte un seul employé mais elle travaille avec un réseau de collaborateurs qui participent au développement des capteurs Stick'n Sense.

Le réseau de collaborateurs s'occupe :

- Du développement de logiciel
- De la rédaction des brevets
- Du développement Hardware.

Ses principaux clients sont Inovadea et la société Stick'n Sense (qui est une société à part entière créée par Monsieur **Paul BERTRAND**).

En plus de ça SmartFuture conseil les entreprises en innovation, parmi ces entreprises on peut retrouver le groupe EDF pour l'ISO et la CEI dans le domaine des protocoles de communication pour les véhicules électriques.

3. Recherche & Développement

Les systèmes qui permettent d'améliorer l'efficacité énergétique des bâtiments commerciaux et tertiaires peuvent s'avérer très coûteux car il faut prendre en compte l'achat, l'installation mais aussi la maintenance.

SmartFuture a donc décidé de développer un système que l'utilisateur pourra installer lui-même sans avoir à ouvrir le tableau et donc sans avoir besoin de l'aide d'un professionnel.

Cette technologie se nomme le Stick'n Sense et est en lien direct avec la mission qui nous a été confiée durant ce stage.

II/ Le Stick'n Sense

La technologie Stick'n Sense est développée par SmartFuture qui possède les droits d'utilisation de cette technologie.

Stick'n Sense est une technologie permettant aux entreprises de manager l'énergie plus facilement et à bas coûts.

Ces capteurs offrent deux fonctionnalités :

- La mesure de l'énergie apparente qui circule dans chaque disjoncteur
- Surveillance de la santé des disjoncteurs.

1. Fonctionnement

Le capteur Stick'n Sense est une technologie non invasive, il se présente sous cette forme (*voir figure 1*). Le capteur est capable de mesurer la température afin de vérifier l'état des relais mais il est aussi capable de déduire le courant qui passe dans le disjoncteur en mesurant le champ électromagnétique généré par ceux-ci. En déduisant le courant, on peut donc connaître la puissance apparente car nous connaissons déjà la tension, qui est celle de secteur, soit environ 230V selon les normes Européennes.

2. Problématique

Un problème se pose avec la mesure du champ électromagnétique. En effet, dans une armoire électrique, les disjoncteurs sont très proches les uns des autres alors les champs électromagnétiques des disjoncteurs proches rentrent en collision, interfèrent entre eux et perturbent les mesures réalisées par les capteurs.

Le projet sur lequel nous travaillons est de jouer plusieurs scénarios d'allumage des relais afin de voir l'influence d'un disjoncteur sur un autre lors des mesures. Ces données serviront à améliorer la précision de la technologie Stick'n Sense.



Figure 1 - Capteur Stick'n Sense

III/ Présentation du projet

1. Résumé de la mission

Durant ce stage, mon binôme et moi avons à étudier, câbler et programmer un système de contrôle automatique de charges dans un tableau électrique. Ce système a pour but de mesurer la précision des capteurs Stick'n Sense en comparant les valeurs mesurées par notre programme avec les valeurs mesurées par les capteurs Stick'n Sense.

La mission était composée en 2 grandes parties :

- Programmation de la boîte à relais afin de jouer un scénario d'allumage des relais
- Programmation des 2 compteurs afin de mesurer et sauvegarder des données dans un fichier csv

2. Cahier des charges

Le tableau à câbler doit être composé de 6 disjoncteurs magnéto thermique (*voir figure 2*), qui servent à détecter les surcharges de courant dans le circuit ainsi que la présence de court-circuit, un ensemble de relais (mbh88) commandés en Modbus (*voir figure 6*), qui serviront d'interrupteur entre la source de tension et les prises, des compteurs d'énergie (rail310V) commandés en Modbus (*voir figure 3*), qui serviront à mesurer la tension, le courant, le déphasage entre courant et intensité, la puissance active/apparente, d'un Raspberry Pi3 qui servira à contrôler notre système(*voir figure 5*) et d'un adaptateur Modbus/port Com afin d'effectuer la connexion entre le Raspberry et les différents appareils(*voir figure 4*).

Le programme du Raspberry devait être capable de jouer un scénario d'allumage et d'extinction des relais mais aussi de récupérer les valeurs des tensions, courant et puissance active mesurée sur chaque disjoncteur par les compteurs.



Figure 4 - Boîte à relais mbh88



Figure 2 - Disjoncteur magnéto thermique



Figure 3 - Compteur Rail310V



Figure 5 - Raspberry Pi 3



Figure 6 - Adaptateur Modbus

3. Planning Prévisionnel

Tableau 1 - Planning prévisionnel

Semaines	15	16	17	18	19	20	21	22	23	24	25	26
Connaissance du projet	X											
	V											
Recherches et lecture de documentation technique	X											
	V											
Création de scénario en CSV grâce à Excel		X										
		V										
Configuration Raspberry Pi		X										
		V										
Programmation du mbh88		X	X	X								
					V							
Implémentation matériel armoire électrique							X					
								V				
Optimisation du programme					X	X						
						V						
Cablage armoire électrique								X				
									V			
Assemblage des 2 programmes mbh88 et Rail310V												
Optimisation du programme final												
Test de l'armoire électrique avec les capteurs Stick'n Sense											X	X
Préparation Soutenance							X	X	X			
										V		
Soutenance Oral										X		
										V		

X = Prévisionnel | V = Réalisé |

IV/ Mes travaux :

1. Mes recherches

1.1) Présentation Modbus

L'objectif premier de ce stage était de faire communiquer le Raspberry et la boîte de relais/compteurs en Modbus.

Le Modbus au même titre que le HTTP (HyperText Transfer protocole) et le SSH (Secure Shell) est un protocole de communication. Il est le plus souvent utilisé pour programmer un API (Automate programmable industriel) ou encore avec les équipements industriels.

1.2) Les modes de transmission

En Modbus, il existe 3 modes de transmissions :

- Le mode unidirectionnel : cela signifie que les données partent dans un seul sens émetteur -> récepteur
- Le mode bidirectionnel : cela signifie que les données peuvent circuler dans les 2 sens mais pas en même temps
- Le mode full duplex : enfin ce mode signifie que les données peuvent circuler dans les 2 sens et ce en simultanée.

1.3) Les Supports physiques de transmission

On peut dénombrer 4 supports physiques de transmission pour le Modbus :

- Nous avons d'abord le RS-232 qui permet de faire communiquer seulement un maître et un esclave, le RS-232 fonctionne en Full Duplex (voir figure 7).

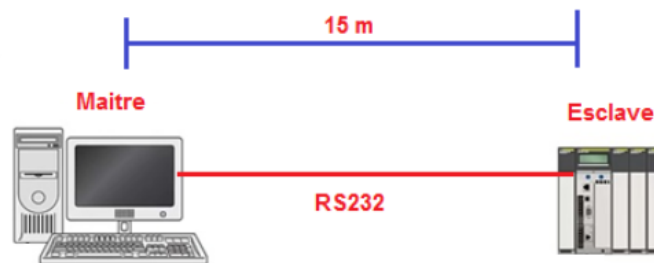


Figure 7 - Transmission RS232

- Ensuite nous avons le RS-422 qui fonctionne aussi en Full Duplex mais qui peut supporter plusieurs récepteurs à l'inverse du RS-232 (voir figure 8).



Figure 8 - Transmission RS422

- Nous avons aussi le RS-485 qui fonctionne le plus souvent en Half Duplex et donc nécessite 2 fils pour la transmission et réception de données (voir figure 9).

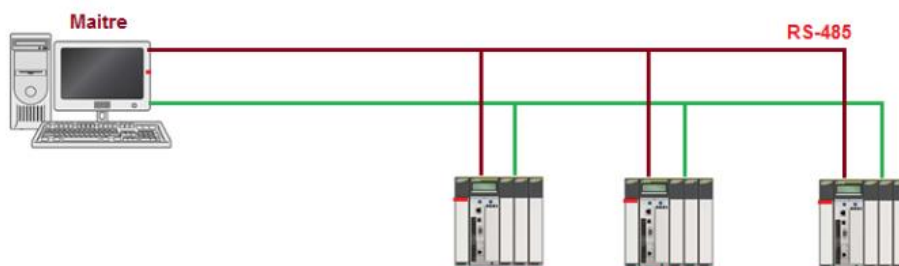


Figure 9 - Transmission RS485

- Pour finir, nous avons le support TCP/IP qui nécessite l'accès à un réseau Ethernet. L'échange de données s'effectue par des mots, chaque mot est composé de plusieurs Bits et un Bit représente l'état d'une variable.

1.4) Les variations du protocole

Il y a trois variations du protocole de communications, on peut y retrouver :

- Le Modbus RTU

La communication Modbus RTU se fait via les interfaces que nous avons vues précédemment (RS-232, RS-422, RS-485). Les Informations transmises en binaires sont codées en binaire.

Les trames que nous envoyons en RTU sont composés :

- De l'adresse du dispositif Modbus (composée d'un octet)
- Du Code de fonction (composé d'un octet)
- Des données à envoyer (composées de n octets)
- Des bits de Contrôle servant à vérifier l'intégrité du message (composés de 2 octets)

- Le Modbus TCP/IP

A l'image du protocole RTU, le protocole TCP/IP permet de faire communiquer deux ou plusieurs appareils.

Afin de communiquer les équipements doivent être connectés sur le même réseau.

Les données transitent sur le réseau via le protocole TCP/IP.

- Les Modbus ASCII

Le protocole Modbus ASCII est semblable au protocole RTU, les seules choses qui divergent sont les informations transmises codées en ASCII à la place du binaire et le contrôle de redondance cyclique qui est moins efficace en ASCII.

2. Création du générateur de scénario en CSV

Un fichier CSV (Common Separate Value) est un format de texte qui représente les données d'un tableau séparé par des virgules, chaque ligne du texte correspond à une ligne du tableau et chaque colonne du tableau est séparée par une virgule ou point-virgule selon notre choix. *(voir figure 10) .*

Le format de fichier csv a été choisie car il est universel, tout le monde peut lire ce fichier mais ce format de fichier a aussi été choisi car nous pouvons l'exploiter avec Microsoft Excel ou bien Calc de Libre Office.

Fichier au format .csv	Représentation tabulaire		
<pre> Sexe,Prénom,Année de naissance M,Alphonse,1932 F,Béatrice,1964 F,Charlotte,1988 </pre>	Sexe	Prénom	Année de naissance
	M	Alphonse	1932
	F	Béatrice	1964
	F	Charlotte	1988

Figure 10 - Exemple de fichier CSV

Nous avons créé à l'aide de Microsoft Excel un générateur de fichier csv qui est présenté sous la forme d'un tableau où l'on peut directement choisir l'état 1 (relais fermé) ou 0 (relais ouvert) pour le relais que l'on souhaite utiliser à l'instant souhaité. ([voir tableau 2](#))

Tableau 2 - Etat des relais

durée (min)	relais 5	relais 4	relais 3	relais 2	relais 1	relais 0
1	0	0	0	0	0	0
3	0	0	0	1	1	1
7	0	1	0	1	0	1
8	0	0	1	0	1	0
9	1	1	0	0	1	1
10	0	1	0	1	0	0
17	1	1	1	1	1	1
25	0	0	1	1	0	0
30	1	1	1	1	1	1
32	1	1	1	1	1	1
35	1	1	1	1	1	1

Nous avons décidé d'ajouter un code couleur en fonction de ce que l'on remplit dans la case et afin de faciliter la visualisation et la compréhension du scénario.

Tableau 3 - Scénario en CSV exploitable

temps (minutes)	relais
0	0
1	0
3	7
7	21
8	10
9	51
10	20
17	63
25	12
30	63
32	63
35	63

Pour réaliser ce tableau, nous avons utilisé la fonction CONCAT afin de combiner le texte de plusieurs cellules dans une seule case. Une fois le texte combiné, nous avons utilisé la fonction BIN2DEC qui sert à convertir un nombre binaire en nombre décimal. L'ensemble des 6 relais sont codés sur 63 bits donc nous utilisons un nombre entier allant de 0 à 63 pour coder l'ensemble des relais. *(voir tableau 3)*

3. Configuration du matériel

Afin de mener à bien notre projet, la société SmartFuture et le département GEII nous ont fournis tout le matériel nécessaire. Une partie du matériel était destiné au montage de l'armoire uniquement et l'autre partie du matériel était d'avantage destinée à la réalisation des deux programmes (mbh88 et Rail310V).

3.1) Liste du matériel

Matériel commun :

- Armoire électrique *(voir figure 12)*
- Disjoncteur magnéto thermique x6
- Prises x6 *(voir figure 13)*
- Disjoncteur différentiel x1 *(voir figure 11)*

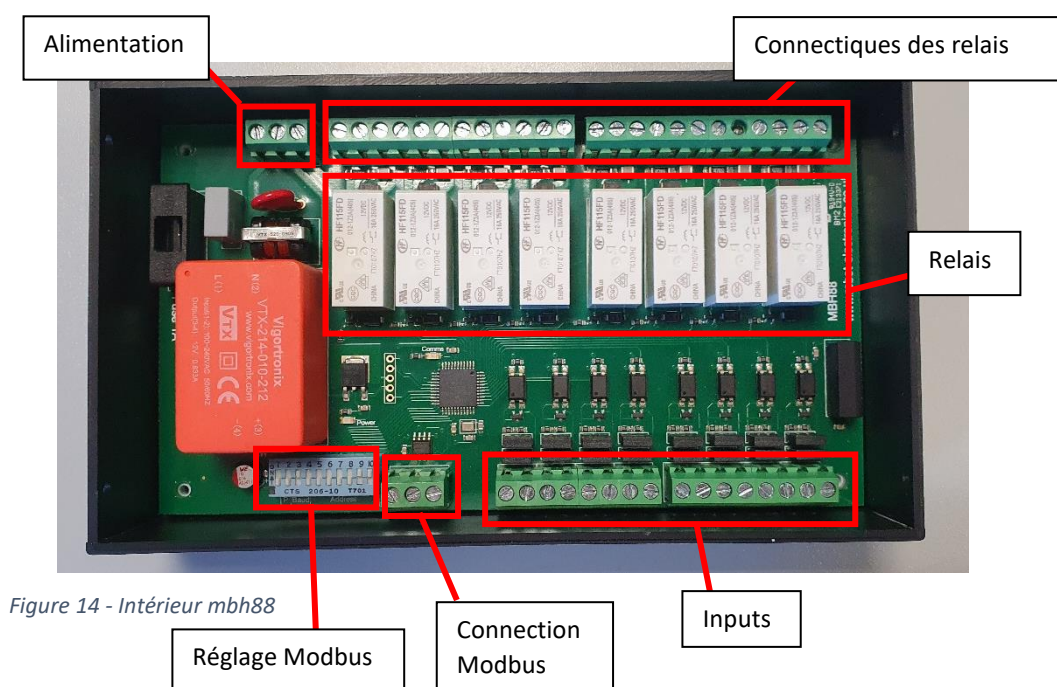
Matériel pour mon travail :

- Boîte à relais mbh88 composé de 8 relais x1
- Convertisseur USB vers RS485
- Raspberry Pi3



3.2) Configuration

Configuration de la boîte à relais mbh88 (voir figure 14) :



La configuration du mbh88 en Modbus s'effectue via les petits leviers à lever ou baisser.

Il a fallu configurer l'adresse 3, pas de parité et un baud rate de 19200 afin d'avoir le même que les deux Rail310V.

3.2.1) Fichier de Configuration

Après avoir configuré les paramètres de la boîte à relais, nous avons besoins d'avoir ces paramètres dans notre programme afin d'établir la connexion en Modbus.

Nous avons décidé de renseigner tous ces paramètres dans un fichier JSON.

Le JSON (Javascript Objet Notation) est un format de stockage de données. Un fichier JSON est composé d'une clé associée à une valeur et lors de la programmation nous pourrions donc appeler une clé et lire directement la valeur qui lui est associée. *(Voir figure 15)*

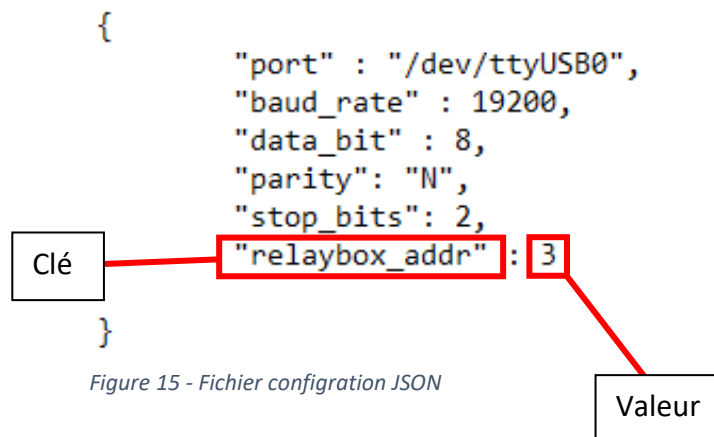


Figure 15 - Fichier configuration JSON

4. Programmation

Afin de réaliser un programme complet qui permet de jouer un scénario d'allumage et d'extinction des relais en fonction de ce qu'on a rempli dans le fichier Excel générateur de csv, il fallait diviser le travail en deux grandes parties.

La première partie servait donc à ouvrir le fichier csv, le lire ainsi que l'interpréter.

La deuxième partie servait quant à elle à communiquer avec la boîte à relais notamment en choisissant et en lisant l'état des relais.

Mon camarade lui s'est occupé des power meter en réalisant un programme qui mesure la puissance active/apparente, tension, intensité et déphasage.

Pour se faire nous avons utilisé le logiciel Code blocks qui est un logiciel de programmation Open Source orienté langage C/C++. Code blocks est compatible Windows, Linux et Mac qui est peut s'avérer important si l'on souhaite faire tourner notre programme sur d'autres machines.

4.1) Les librairies

Pour réaliser nos deux programmes nous avons eu recours à la librairie de fonction Modbus, ce qui permet de bien faciliter le travail car les appareils compatibles Modbus sont compatibles avec certaines fonctions pré-faites Modbus.

On dénombre selon l'annexe 21 fonctions standardisée (voir annexe)

La boîte à relais (mbh88) est compatible avec cinq de ces fonctions :

- 0x01 Read Coils :
 - Cette fonction a pour utilité de lire l'état actuel des relais.
- 0x02 read Discrete inputs :
 - Cette fonction sert à lire l'état des entrées du mbh88.
- 0x03 Read Holding Register :
 - Cette fonction sert à obtenir les informations de l'appareil en lisant des registres, nous avons le registre 1 qui correspond au nombre du model (qui est 0x188 pour le mbh88), le registre 2 qui correspond à la révision matérielle et le registre 3 qui correspond à la révision logicielle.
- 0x05 Write Single Coil :
 - Cette fonction est utilisée afin d'ouvrir ou de fermer un seul relais
- 0x0F Write Multiple Coils :
 - Cette fonction sert quant à elle à ouvrir ou bien fermer plusieurs relais en même temps.

Les fonctions Write single Coil et Write Multiples Coils se ressemblent mais pour notre programme nous utiliserons uniquement Write Multiple Coils car si nous utilisons Write Single Coils plusieurs fois de suite la commutation des relais risque de ne pas être synchrone.

4.2) Structure du programme CSV

Ce programme s'occupe de la partie exploitation du fichier CSV

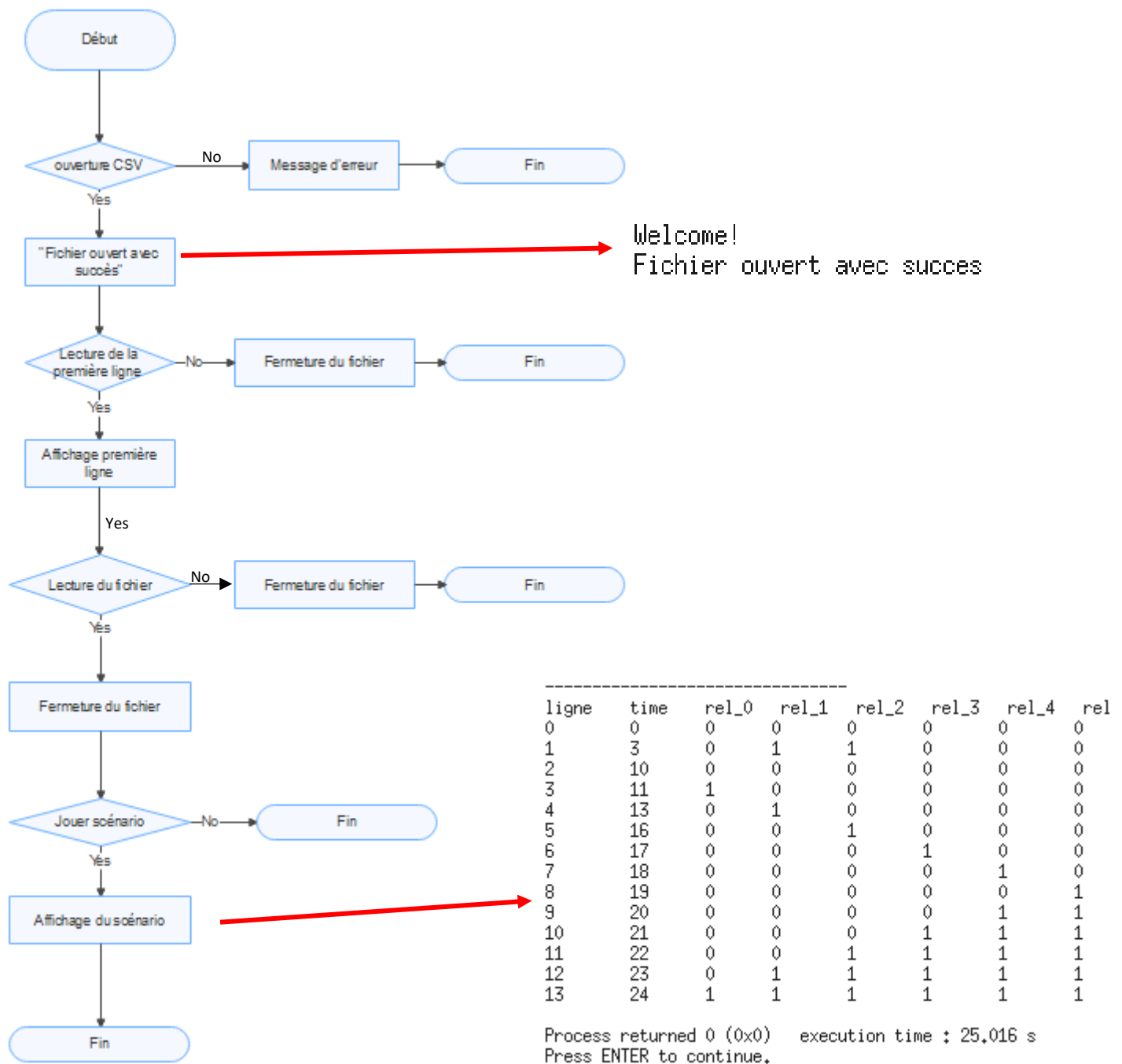


Figure 16 - Organigramme programme exploitation CSV

4.3) Structure du programme relais

Nous avons réalisé ce programme afin de tester la connexion en Modbus de la boîte à relais mais aussi tester les fonctions Modbus notamment de lecture et d'écriture des relais. (Voir figure 17)

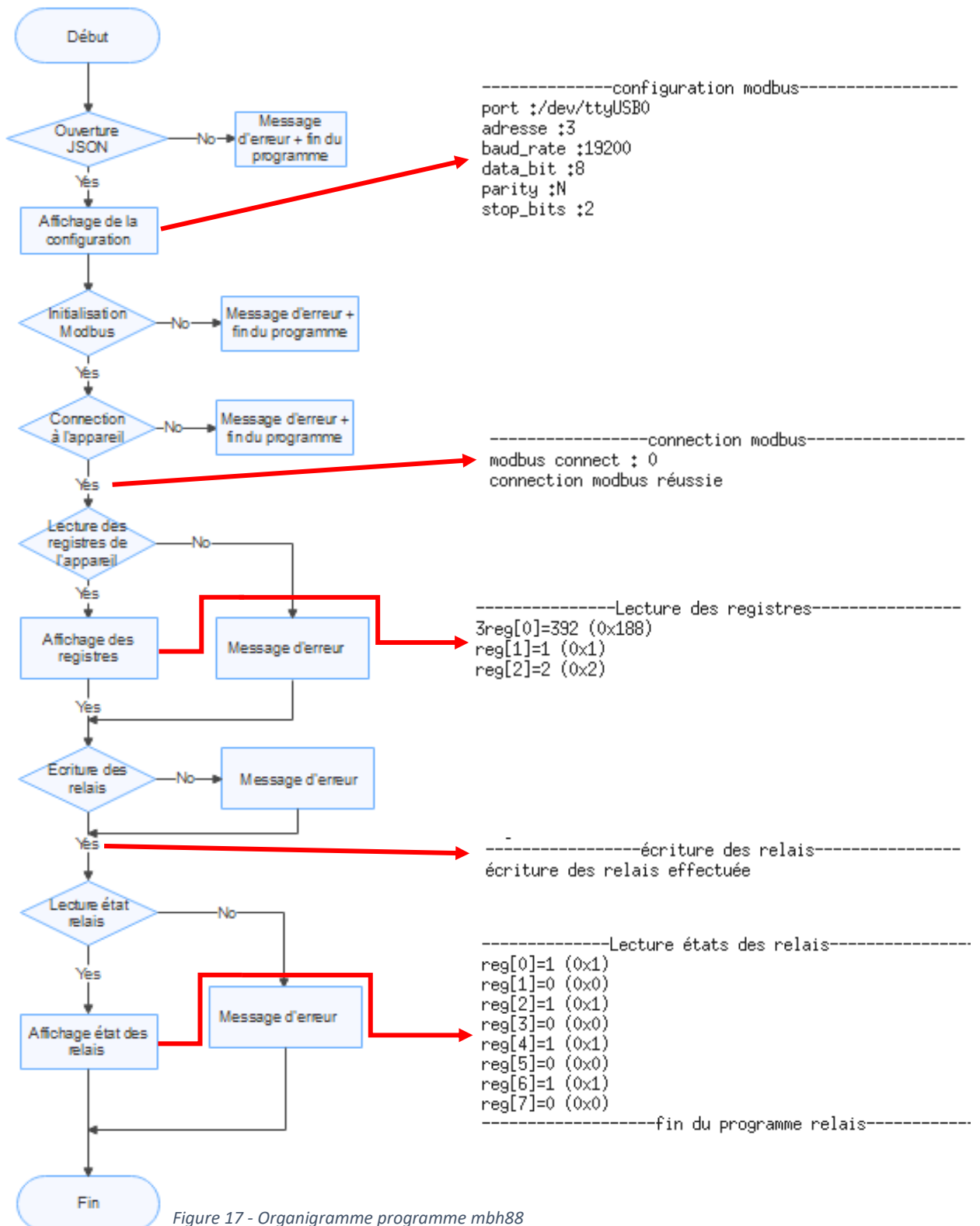
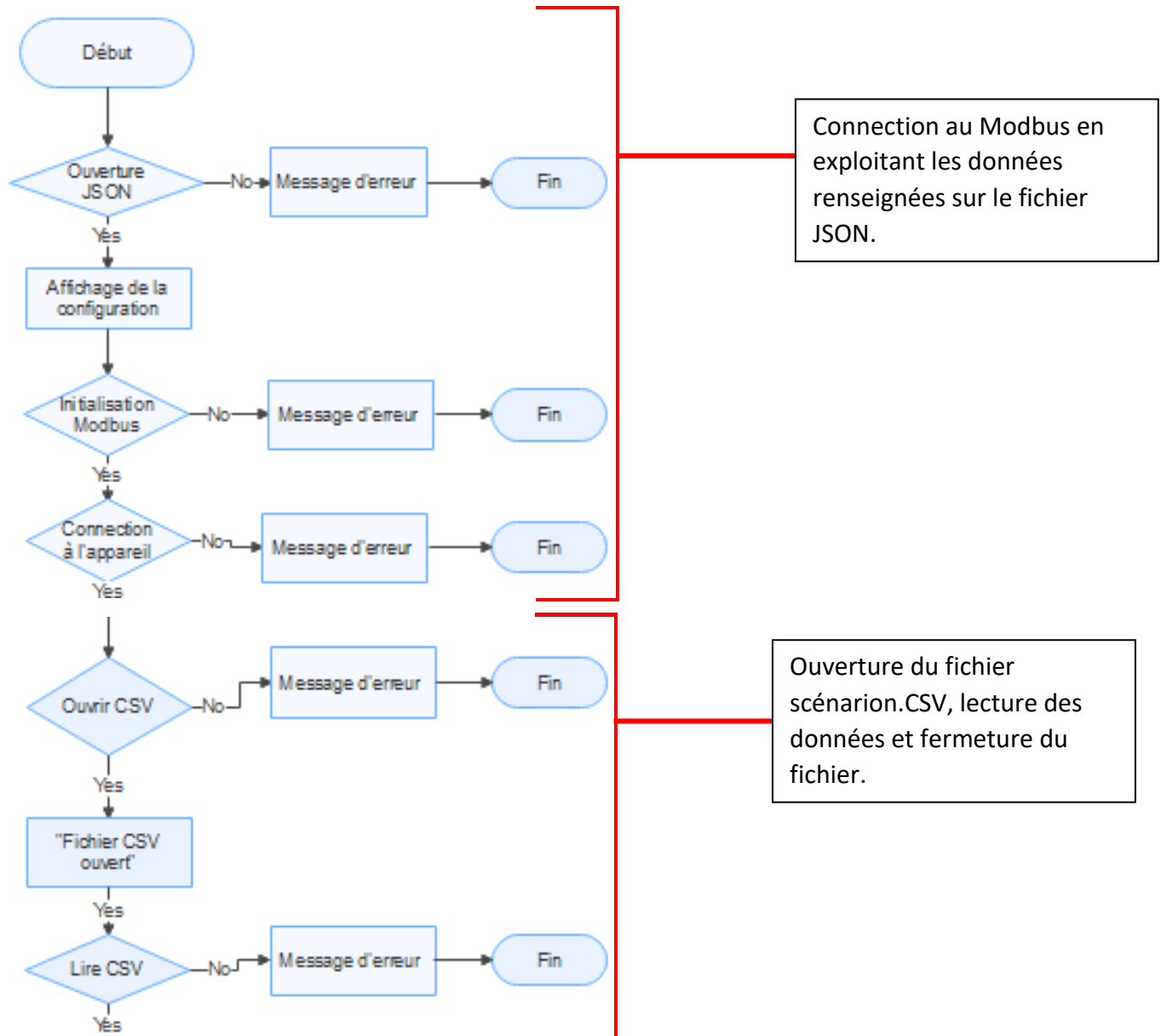


Figure 17 - Organigramme programme mbh88

4.4) Structure du programme complet

Le programme complet qui sert à jouer les scénarios d'allumage est un mélange des deux programmes montrés précédemment et il peut se découper en trois grandes parties.



Nous avons ensuite la partie qui s'occupe d'ouvrir ou de fermer les relais en fonction de ce que nous avons renseigné dans le fichier du scénario.

Pour se faire nous avons implémenter la fonction qui sert à l'écriture des relais ainsi qu'à la lecture des relais dans la fonction qui s'occupe d'afficher le scénario

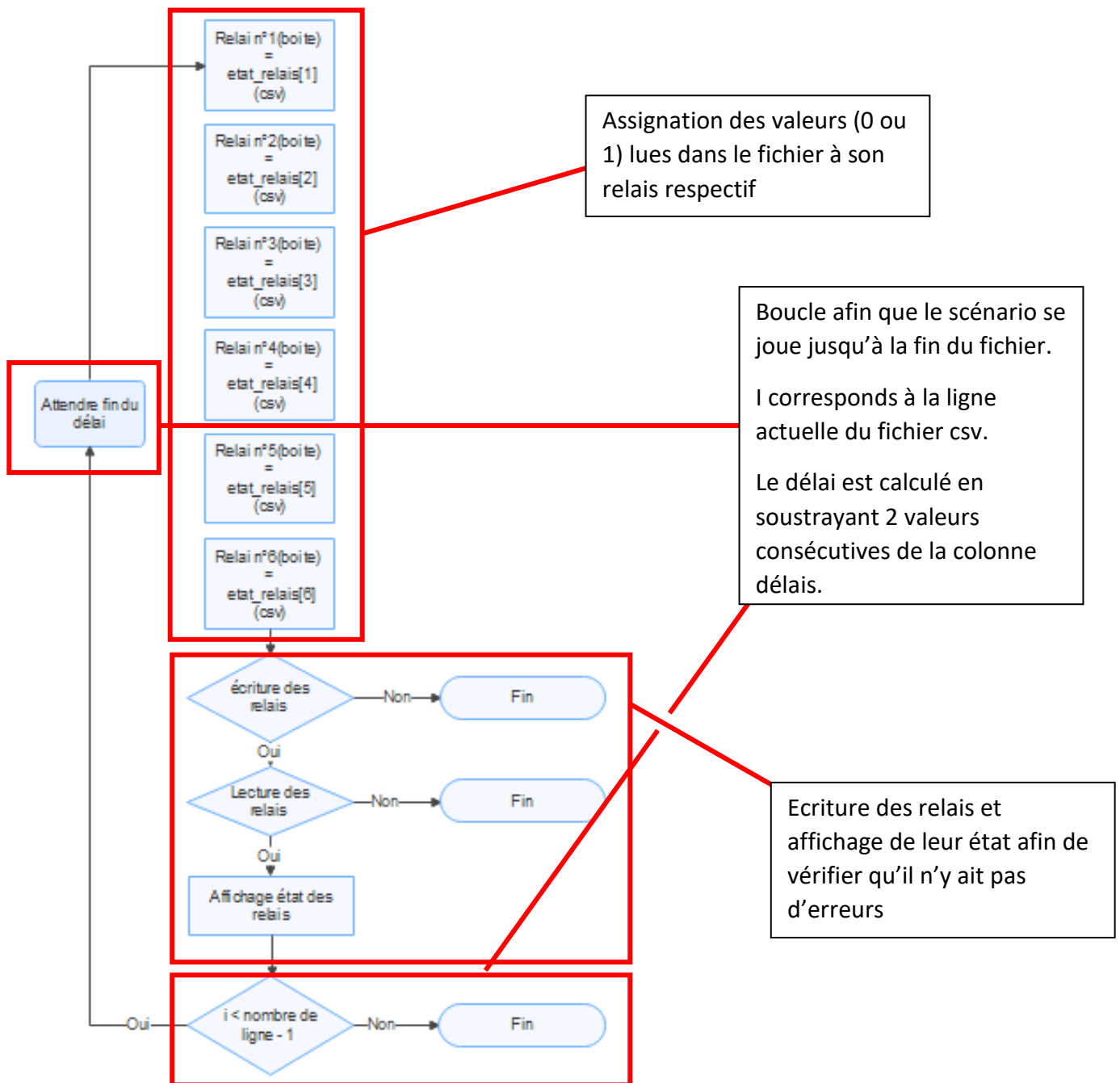
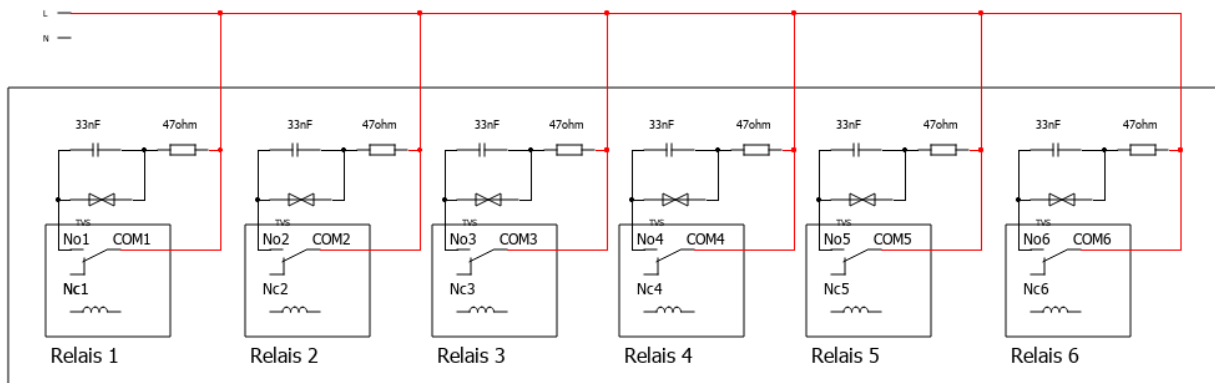


Figure 18 - Fonction scénario relais

5. Montage de l'armoire

Afin de connecter la boîte à relais mbh88 dans le montage il fallait connecter la phase au contacteur NO (normally open) car nous voulons qu'à l'état par défaut les relais ne laissent pas passer de courant.

Nous avons ponté les NO à l'aide de borniers de raccordement et nous les avons connectés à la phase. (Voir figure 18)



Boîte à relais mbh88

Figure 19 - Schéma montage boîte à relais

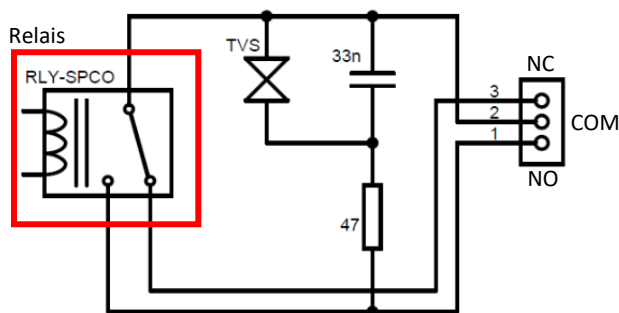


Figure 20 - Schéma interne relais

Lorsque nous mesurons la tension à la borne NO, nous nous rendons compte que peu importe l'état du relais nous relevons une tension alors que le relais est censé être ouvert.

Nous nous sommes rendu compte que la cause de ce phénomène était le circuit RC. (Voir figure 20)

Ce circuit est plus communément appelé « protection Snubber » ou encore « réseau de protection des contacts ». Il est le plus souvent placé sur les contacts des relais ou des interrupteurs car il arrive qu'à la fermeture ou bien l'ouverture des contacts un arc électrique se crée et donc puisse abimer le matériel dans le temps.

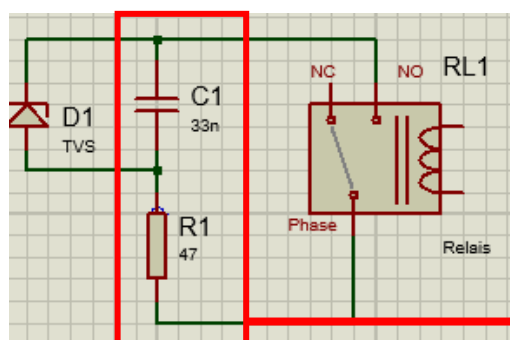


Figure 21 - Schéma interne relais Proteus

Équation 1 - Calcul impédance Z

$$Z = \frac{1}{jCw} + R = 47 - j \frac{1}{100\pi * 33 * 10^{-9}}$$

Lorsque nous venons prendre les mesures sans charges, nous avons donc l'impédance interne du multimètre qui vient s'ajouter.

Nous avons mesuré une résistance interne de 10 MΩ pour le multimètre.

Lorsque nous mesurons, nous avons donc :

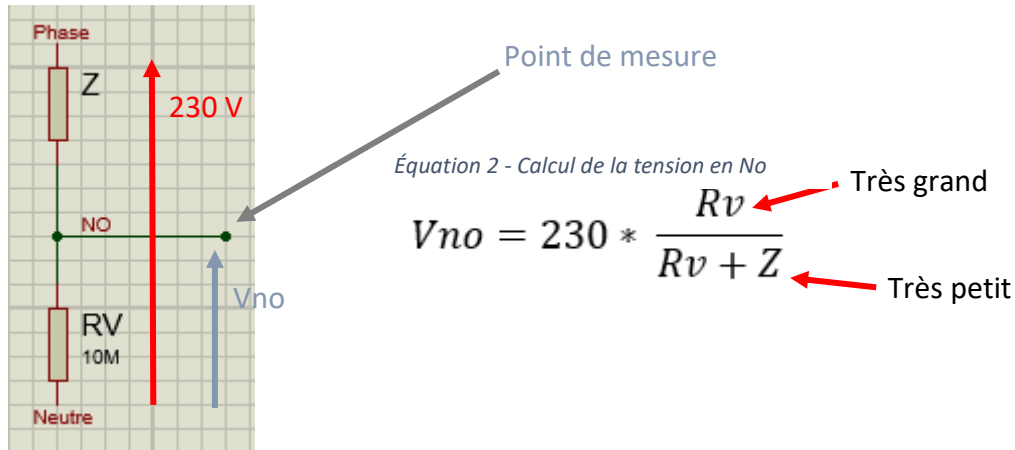


Figure 22 - Pont diviseur

Lorsque nous mesurons sans charge, il est donc normal qu'on obtienne 230V.

Nous avons retesté les branchements de la boîte à relais en connectant une charge et nous avons pu voir que la tension s'écroulait.



Figure 23 - Tableau électrique câblé

V/ Conclusion

1. Bilan technique

Le projet qui nous a été confié comprenait beaucoup de programmation mais aussi pas mal de pratique. Durant ce stage j'ai découvert un nouveau protocole de communication qui est le Modbus, cette découverte peut s'avérer utile pour la suite car ce protocole de communication est très utilisé dans l'industrie. La formation du DUT m'a beaucoup aidé tout au long du stage, cela m'a aidé à mieux comprendre à pouvoir m'adapter à programmer en C++. J'ai développé mes connaissances en informatique avec l'apprentissage du type de fichier JSON mais aussi le CSV

Sur la partie montage de l'armoire j'ai appris à construire un schéma électrique et à câbler une armoire.

2. Bilan personnel

Le stage que j'ai effectué s'est déroulé en grande partie au département GEII de l'IUT, je n'ai pas eu l'occasion de découvrir ce qu'est réellement le monde de l'entreprise mais en contrepartie j'ai réellement pu apprendre de nouvelles choses qui me seront fortement utiles pour la poursuite de mes études. Car, à l'inverse d'un stage plutôt classique où l'on m'aurait confié quelques tâches qui n'ont pas forcément de rapport les unes avec les autres, j'ai pu là travailler en collaboration sur un réel projet avec une réelle utilité. Ce stage m'aura aussi de me conforter dans mon choix de poursuivre mes études (Licence Sciences de l'Ingénieur).

Index des Illustrations

<i>Figure 1 – Capteur Stick’n Sense</i>	<i>7</i>
<i>Figure 2 – Disjoncteur magnéto thermique</i>	<i>8</i>
<i>Figure 3 – Compteur rail 310V</i>	<i>8</i>
<i>Figure 4 – Adaptateur Modbus</i>	<i>8</i>
<i>Figure 5 – Raspberry Pi3</i>	<i>8</i>
<i>Figure 6 – Boite à relais mbh88</i>	<i>8</i>
<i>Figure 7 – Liaison RS232</i>	<i>10</i>
<i>Figure 8 – Liaison RS422</i>	<i>10</i>
<i>Figure 9 – Liaison RS485</i>	<i>10</i>
<i>Figure 10 – Exemple de fichier CSV</i>	<i>12</i>
<i>Figure 11 – Disjoncteur différentiel</i>	<i>14</i>
<i>Figure 12 – Armoire électrique</i>	<i>14</i>
<i>Figure 13 – Prise électrique</i>	<i>14</i>
<i>Figure 14 – Intérieur mbh88</i>	<i>14</i>
<i>Figure 15 – Fichier Configuration JSON</i>	<i>15</i>
<i>Figure 16 – Organigramme programme exploitation CSV</i>	<i>17</i>
<i>Figure 17 - Organigramme programme mbh88</i>	<i>18</i>
<i>Figure 18 - Fonction scénario relais</i>	<i>20</i>
<i>Figure 19 - Schéma montage boite à relais</i>	<i>21</i>
<i>Figure 20 - Schéma interne relais</i>	<i>21</i>
<i>Figure 21 - Schéma interne relais Proteus</i>	<i>21</i>
<i>Figure 22 - Pont diviseur</i>	<i>22</i>
<i>Figure 23 - Tableau électrique câblé</i>	<i>22</i>
<i>Tableau 1 - Planning prévisionnel</i>	<i>9</i>
<i>Tableau 2 - Etat des relais</i>	<i>12</i>
<i>Tableau 3 - Scénario en CSV exploitable</i>	<i>13</i>

Bibliographie

Figure 1 – Capteur Stick'n Sense : <https://stick-n-sense.com/>

Figure 2 – Disjoncteur magnéto thermique :

<https://www.materielelectrique.com/disjoncteurmagnetotherm-6000-lexic-agrave-courbe-p-1341.html>

Figure 3 – Compteur Rail 310V : <https://www.ndmeter.co.uk/rail310.html>

Figure 4 – Adaptateur Modbus : <https://www.antarc-automation.com/imaster/33-adaptateur-usb-rs485.html>

Figure 5 – Raspberry Pi3 : <https://urlz.fr/fRTV> (<https://fr.rs-online.com/web/>)

Figure 6 – Boite à relais mbh88 : <https://www.antratek.de/modbus-rtu-rs485-module-8-relays-8-inputs>

Figure 7 – Liaison RS232 : <https://www.automation-sense.com/medias/files/guide-du-modbus-pour-les-nuls-extrait.pdf>

Figure 8 – Liaison RS4322 : <https://www.automation-sense.com/medias/files/guide-du-modbus-pour-les-nuls-extrait.pdf>

Figure 9 – Liaison RS485 : <https://www.automation-sense.com/medias/files/guide-du-modbus-pour-les-nuls-extrait.pdf>

Figure 10 – Exemple de fichier CSV : https://fr.wikipedia.org/wiki/Comma-separated_values

Figure 11 – Disjoncteur différentiel : <https://www.conrad.fr/p/interrupteur-differentiel-legrand-411560-40-a-003-a-230-v-1-pcs-1435411>

Figure 12 – Armoire électrique : <https://www.amazon.fr/%C3%9868-encast%C3%A9-IP40-locaux-distribution-36-modules-Transparentee/dp/B01BSMHQUA>

Figure 13 – Prise électrique : <https://urlz.fr/fRUq> (<https://fr.rs-online.com/web/>)

Sources – SmartFuture : <https://www.smartfuture.fr/>

Sources – Stick'n Sense : <https://stick-n-sense.com/>

Sources – Modbus : : <https://www.automation-sense.com/medias/files/guide-du-modbus-pour-les-nuls-extrait.pdf>

Sources – CSV : <https://debitor.fr/termes-comptables/fichier-csv>

Sources – JSON : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445308-json-definition-et-presentations-de-ce-format-de-donnees/>

Sources – Circuit RC (Snubber) : <http://philippe.demerliac.free.fr/Articles/Snubber/Snubber.htm>

Annexes

Code de Fonction	Fonction
0x01	Read Coils
0x02	Read Discrete Inputs
0x03	Read Holding Registers
0x04	Read Input Registers
0x05	Write Single Coil
0x06	Write Single Register
0x07	Read Exception Status (Serial Line only)
0x08	Diagnostics (Serial Line only)
0x0B	Get Comm Event Counter (Serial Line only)
0x0C	Get Comm Event Log (Serial Line only)
0x0F	Write Multiple Coils
0x10	Write Multiple registers
0x11	Report Server ID (Serial Line only)
0x14	Read File Record
0x15	Write File Record
0x16	Mask Write Register
0x17	Read/Write Multiple registers
0x18	Read FIFO Queue
0x2B	Encapsulated Interface Transport
0x2B/ 0x0D	CANopen General Reference Request and Response PDU
0x2B/ 0x0E	Read Device Identification

Annexe 1 – Code de fonction Modbus

Résumé & Abstract

1) Résumé

Les capteurs Stick'n Sense développés par la société SmartFuture mesurent la puissance apparente dans chaque disjoncteur à l'aide du champ électromagnétique généré par ceux-ci. Les mesures sont perturbées à cause de la proximité des champs électromagnétiques.

Notre objectif était de réaliser un système de contrôle automatique de charge dans un tableau électrique. Mon camarade s'est occupé lui de la partie mesure de la tension, du courant, du déphasage entre courant et intensité, et de la puissance active/apparent. J'étais en charge de la partie de la boîte à relais. Mon objectif était de réaliser un programme qui permet de jouer un scénario d'allumage et d'extinction de relais selon ce qu'on renseignait dans un tableau Excel. Il a fallu que nous réalisons tous les deux notre programme, ensuite nous nous sommes occupés de fusionner nos programmes et enfin nous avons monté le tableau électrique.

2) Abstract

The Stick'n Sense sensors developed by SmartFuture measure the apparent power in each circuit breaker using the electromagnetic field generated by them. The measurements are disturbed due to the proximity of the electromagnetic fields.

Our objective was to realize a system of automatic control of load in an electrical panel. My classmate took care of the measurement of the voltage, the current, the phase shift between current and intensity, and the active/apparent power. I was in charge of the part of the relay box, my objective was to carry out a program to play a scenario of lighting and extinction of relay according to what one informed in a table Excel. We had to realize both our program, then we took care of merging our programs and finally we assembled the electrical panel.