

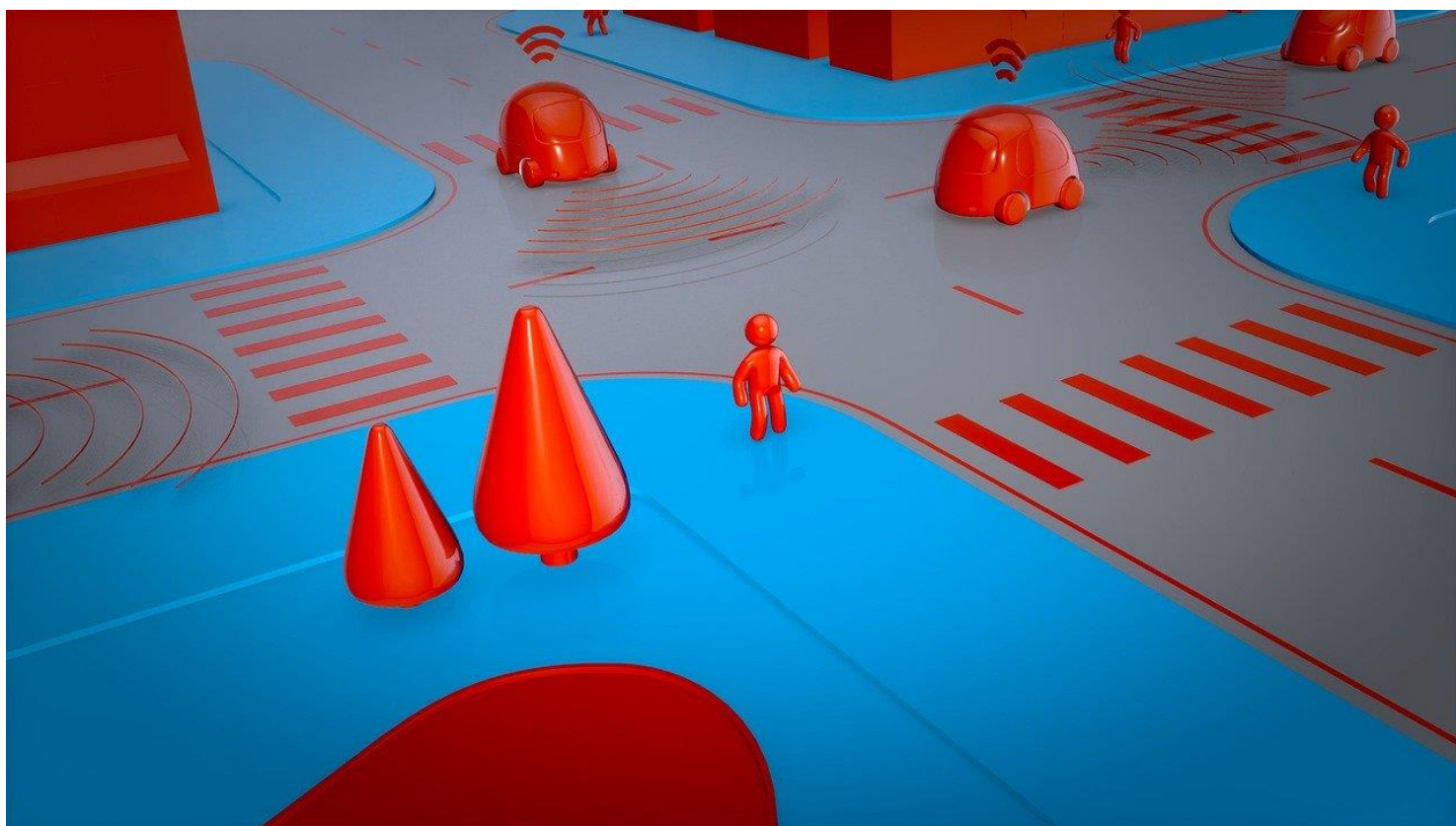


E 6-2 – PROJET TECHNIQUE

Dossier de présentation et de validation du projet (consignes et contenus)

Groupement académique : AIX-MARSEILLE		Session 2022
Lycée : Alphonse BENOIT		
Ville : L'ISLE SUR LA SORGUE		
N° du projet : 3	Nom du projet : SmartCity : Gestion intelligente d'une ville	

Projet nouveau	Oui	<input type="checkbox"/> Non	Projet interne	Oui	<input type="checkbox"/> Non
Délai de réalisation	06/01/2022 au 30/05/2022		Statut des étudiants	<input checked="" type="checkbox"/> Formation initiale	<input type="checkbox"/> Apprentissage
Spécialité des étudiants	EC	IR	<input checked="" type="checkbox"/> Mixte	Nombre d'étudiants 4 étudiants : 2 IR, 2 EC	
Professeurs responsables	ANTOINE / HORTOLLAND / ESCURET				



Sommaire

1	Présentation et situation du projet dans son environnement.....	3
1.1	Contexte de réalisation.....	3
1.2	Présentation du projet.....	3
1.3	Cahier des charges – Expression du besoin	4
2	Spécifications.....	4
2.1	Diagrammes UML / SYSML	4
2.1.1	Diagrammes des cas d'utilisation	5
2.1.2	Diagramme de déploiement	5
2.1.3	Exigences	6
2.1.4	Architectures Matérielle & Logicielle	8
2.1.5	Scénarios des cas d'utilisation	12
2.2	Contraintes de réalisation.....	13
2.3	Ressources mises à disposition des étudiants (logiciels / matériels / documents).....	13
3	Répartition des tâches par étudiant	14
4	Exploitation Pédagogique – Compétences terminales évaluées :.....	16
5	Planification.....	17
6	Conditions d'évaluation pour l'épreuve E6-2.....	17
6.1	Disponibilité des équipements	17
6.2	Atteintes des objectifs du point de vue client.....	17
6.3	Avenants :.....	17
7	Observation de la commission de Validation.....	18
7.1	Avis formulé par la commission de validation :.....	18
7.2	Nom des membres de la commission de validation académique :.....	18
7.3	Visa de l'autorité académique :.....	19

1 Présentation et situation du projet dans son environnement

1.1 Contexte de réalisation

Constitution de l'équipe de projet :	Étudiant 1 EC <input checked="" type="checkbox"/> IR	Étudiant 2 EC <input checked="" type="checkbox"/> IR	Étudiant 3 <input checked="" type="checkbox"/> EC IR	Étudiant 4 <input checked="" type="checkbox"/> EC IR
Projet développé :		Au lycée ou en centre de formation <input type="checkbox"/> En entreprise <input checked="" type="checkbox"/> Mixte		
Type de client ou donneur d'ordre (commanditaire) :		Entreprise ou organisme commanditaire : <input checked="" type="checkbox"/> Oui Non Nom : Philippe GOZLAN / Marc SILANUS Adresse : CERI / Université d'Avignon pôle Agro parc Montfavet Contact : Origine du projet : <ul style="list-style-type: none"> • Idée : Lycée <input checked="" type="checkbox"/> Entreprise • Cahier des charges : Lycée <input checked="" type="checkbox"/> Entreprise • Suivi du projet : <input checked="" type="checkbox"/> Lycée <input checked="" type="checkbox"/> Entreprise 		
Si le projet est développé en partenariat avec une entreprise :		Nom de l'entreprise : CERI / Adresse de l'entreprise : Université d'Avignon pôle Agro parc Adresse site : Montfavet Tél. : Courriel :		

1.2 Présentation du projet

Le CERI est le Centre d'Etudes et de Recherches en Informatique, un département de l'université d'Avignon.

Dans le cadre de la formation Master CMI, l'équipe pédagogique désire une maquette pilotable d'une ville contenant à minima un système d'éclairage des voies composé de plusieurs tronçons, un parking, une intersection, des véhicules, des piétons.

Cette maquette sera utilisée par les étudiants pour développer des applications de gestion intelligente de la ville. Les logiciels clients créés par les étudiants du CERI utiliseront des technologies avancées telles que les réseaux neuronaux, les systèmes experts, l'intelligence artificielle, etc.

Le développement de ce projet a débuté en 2021 et il nécessite de continuer cette année.

Les deux étudiants EC apporteront des améliorations sur les cartes « Feux d'intersection » et « Gestion de parking ».

Les deux étudiants IR devront mettre en œuvre les logiciels client/serveur et s'assurer de leur bon fonctionnement.

Les règles du jeu sont à coder.

Enfin, l'ensemble du système est à valider par tous les étudiants avant livraison au client.

1.3 Cahier des charges – Expression du besoin

Constitution d'une maquette représentant quelques éléments d'une ville : parking, éclairage public, intersection, voitures, piétons.

Les éléments de la maquette peuvent être commandés depuis un client réseau.

Un site WEB de supervision permettra de voir ce qui se passe en temps réel.

Aucun automatisme ne sera développé au sein de la maquette, tout sera contrôlé par le client réseau.

Bien que le client soit développé par les étudiants de Master du CERI, il sera toutefois nécessaire de disposer d'au moins un client fonctionnel pour donner vie à la maquette. Le client contrôlera manuellement la maquette.

2 Spécifications

2.1 Diagrammes UML / SYSML

Le système se décompose en plusieurs sous-systèmes :

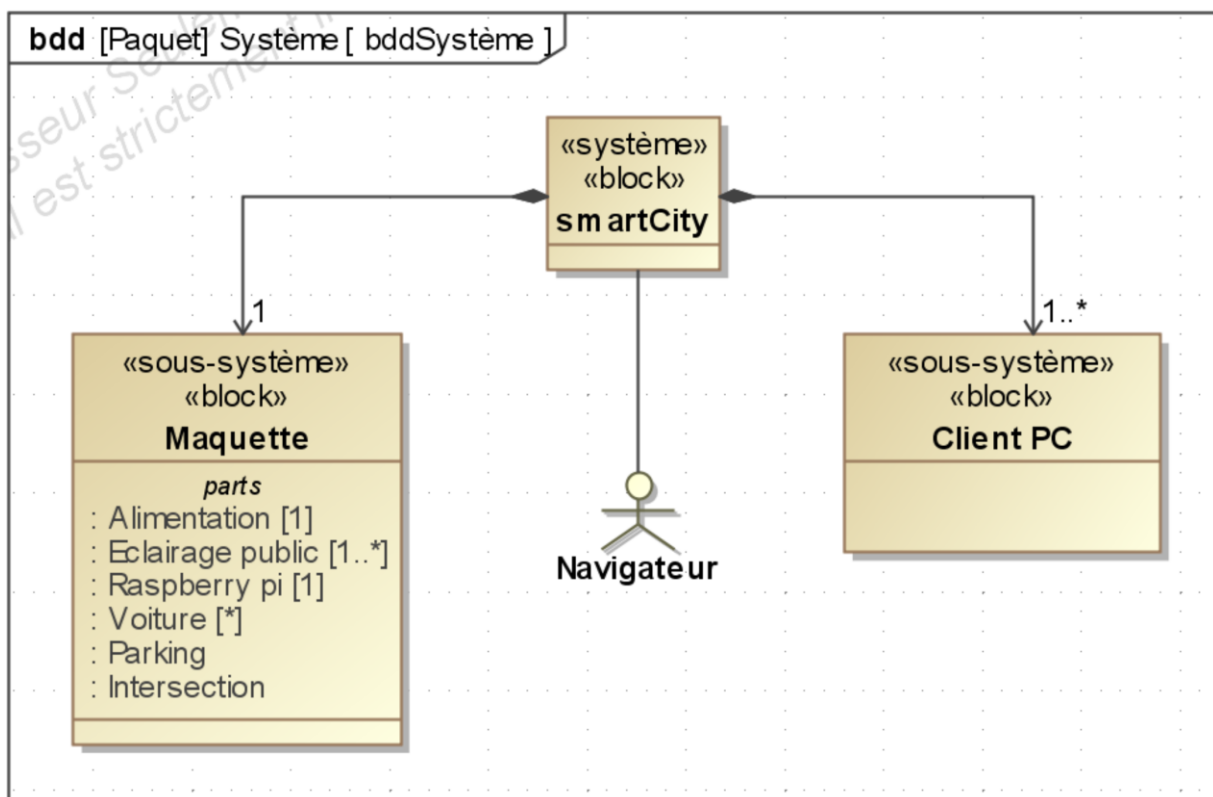


Figure 1 : Diagramme de blocs du système

2.1.1 Diagrammes des cas d'utilisation

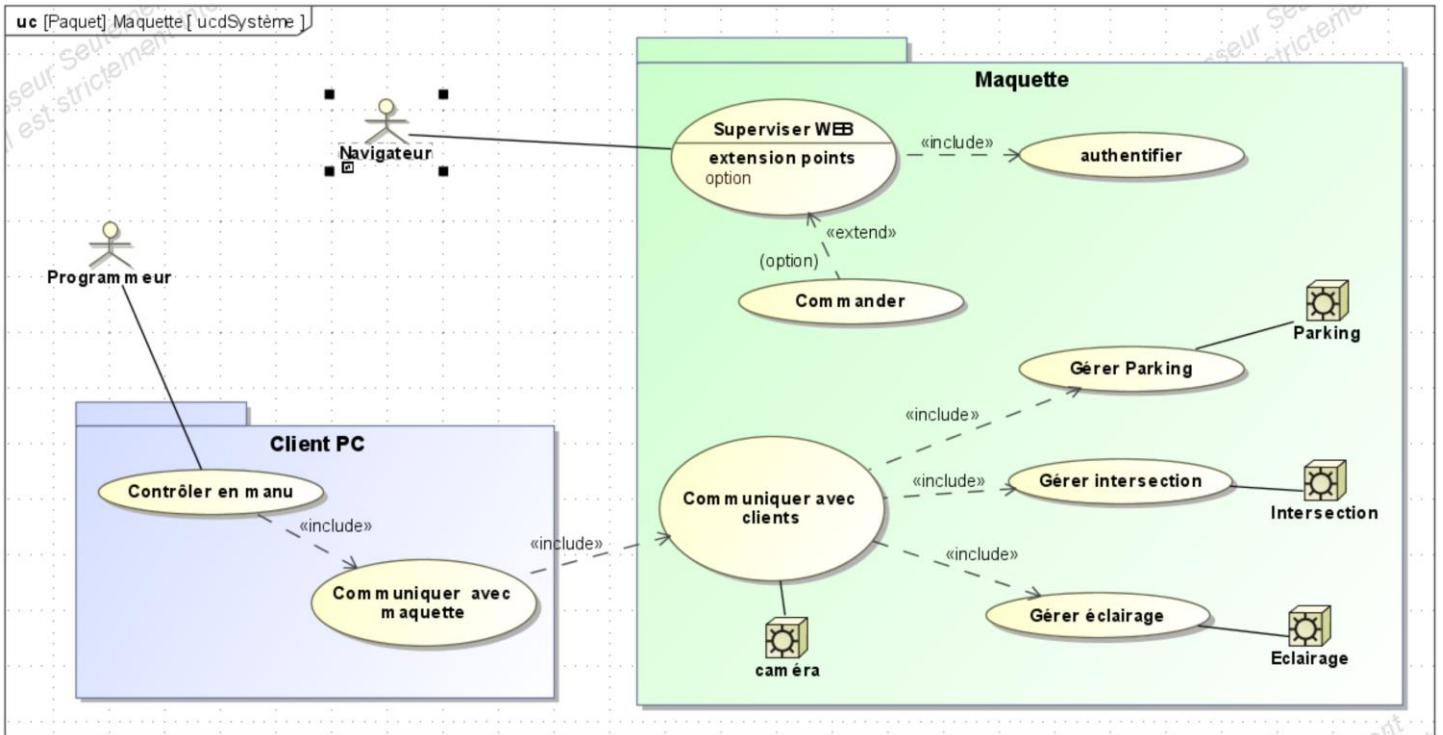


Figure 2 : Diagramme des cas d'utilisation du système

2.1.2 Diagramme de déploiement

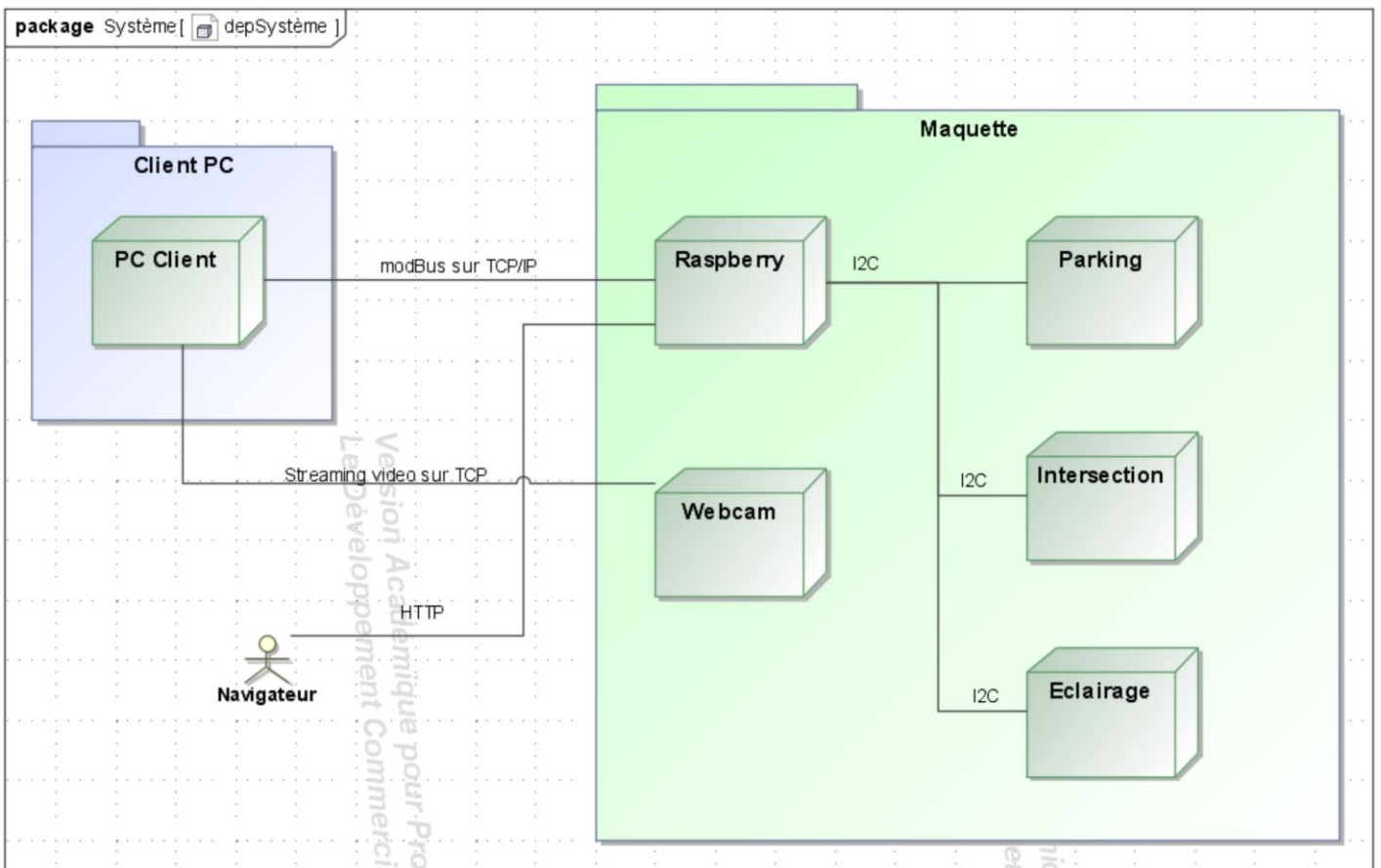


Figure 3 : Diagramme de déploiement du système

2.1.3 Exigences

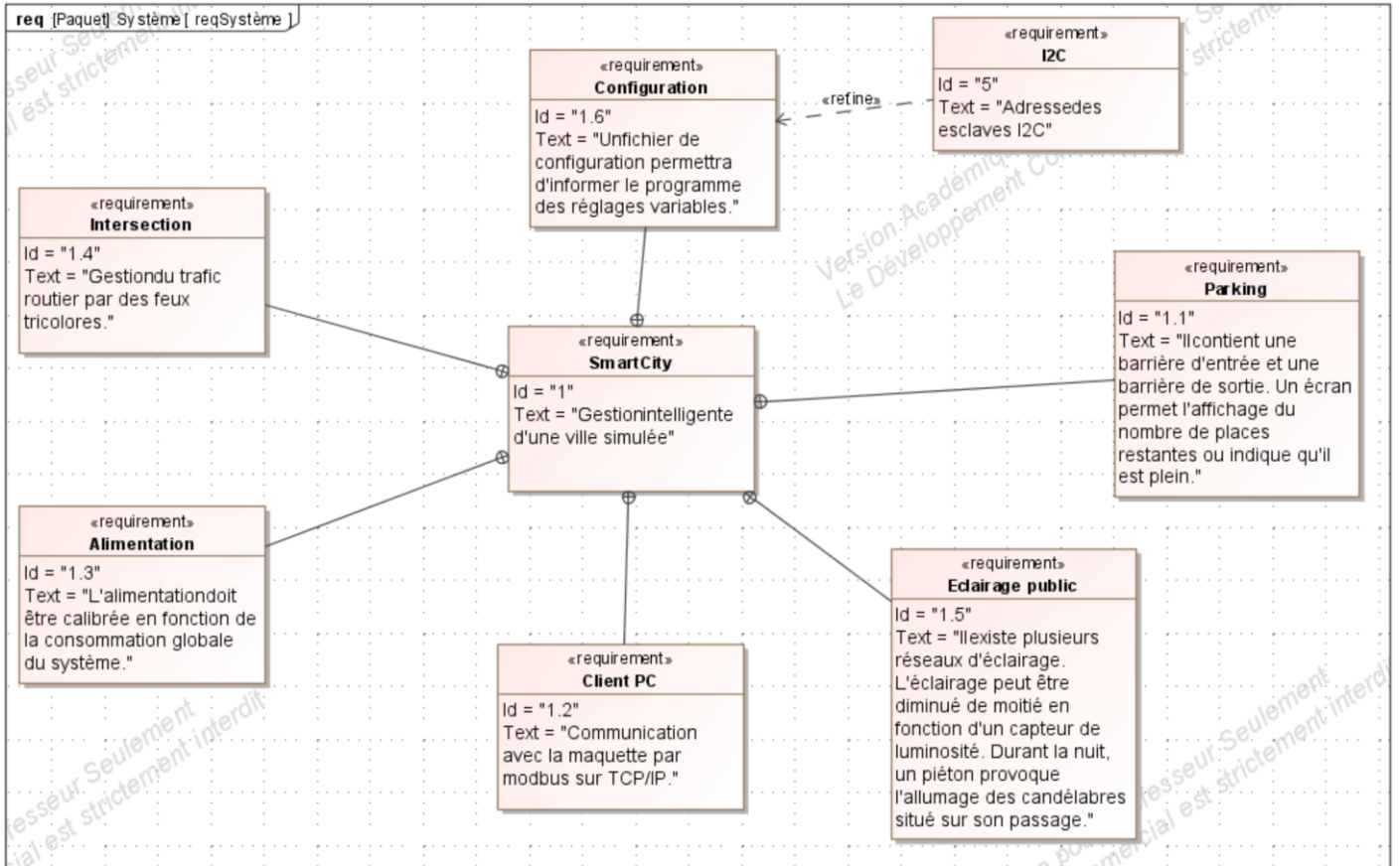


Figure 4 : Diagramme des exigences

#	ID	Name	Text																																																																
1	6	modBus TCP	Le protocole n'utilisera que les codes fonction lecture 1 octet et écriture n octets. Une adresse est codée sur 16 bits et correspond à un octet.																																																																
2	4	Parking	<p>Le parking contient 8 places et 8 places abonnés. A l'entrée du parking se trouve une barrière, un écran indiquant le nombre de places restantes, un bouton poussoir pour appeler de l'aide et un lecteur RFID. Chaque véhicule doit posséder une carte RFID pour entrer dans le parking, soit en tant que client simple, soit en tant qu'abonné. L'inscription d'un client et l'envoi de la carte RFID ne fait pas partie de l'étude. Le véhicule est détecté par la présence de la carte RFID.</p> <p>A la sortie se trouve une barrière et un lecteur RFID pour identifier les véhicules sortants.</p> <table border="1"> <thead> <tr> <th>Adresse modBus (hexa)</th> <th>Désignation</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>0x0010 - 0x001F (Écriture seule)</td> <td>Texte affichage ligne supérieure (16 caractères)</td> <td></td> <td></td> </tr> <tr> <td>0x0020 - 0x002F (Écriture seule)</td> <td>Texte affichage ligne inférieure (16 caractères)</td> <td></td> <td></td> </tr> <tr> <td>0x0030 (Écriture seule)</td> <td>bit 0</td> <td>Ordre monter barrière entrée</td> <td>-</td> </tr> <tr> <td></td> <td>bit 1</td> <td>Ordre descendre barrière entrée</td> <td>-</td> </tr> <tr> <td></td> <td>bit 2</td> <td>Ordre monter barrière sortie</td> <td>-</td> </tr> <tr> <td></td> <td>bit 3</td> <td>Ordre descendre barrière sortie</td> <td>-</td> </tr> <tr> <td>0x0031 (lecture seule)</td> <td>bit 0</td> <td>Barrière montée entrée</td> <td>-</td> </tr> <tr> <td></td> <td>bit 1</td> <td>Barrière descendue entrée</td> <td>-</td> </tr> <tr> <td></td> <td>bit 2</td> <td>Barrière montée sortie</td> <td>-</td> </tr> <tr> <td></td> <td>bit 3</td> <td>Barrière descendue sortie</td> <td>-</td> </tr> <tr> <td></td> <td>bit 4</td> <td>Bouton d'appel appuyé</td> <td>-</td> </tr> <tr> <td>0x0032 (lecture seule)</td> <td>bit0-bit7</td> <td>Place occupée</td> <td>Place libre</td> </tr> <tr> <td>0x0033 (lecture seule)</td> <td>bit0-bit7</td> <td>Place occupée</td> <td>Place libre</td> </tr> <tr> <td>0x0034 (lecture seule)</td> <td colspan="3">Compteur des places libres non abonnés</td> </tr> <tr> <td>0x0035 (lecture seule)</td> <td colspan="3">Compteur des places libres abonnés</td> </tr> </tbody> </table>	Adresse modBus (hexa)	Désignation	1	0	0x0010 - 0x001F (Écriture seule)	Texte affichage ligne supérieure (16 caractères)			0x0020 - 0x002F (Écriture seule)	Texte affichage ligne inférieure (16 caractères)			0x0030 (Écriture seule)	bit 0	Ordre monter barrière entrée	-		bit 1	Ordre descendre barrière entrée	-		bit 2	Ordre monter barrière sortie	-		bit 3	Ordre descendre barrière sortie	-	0x0031 (lecture seule)	bit 0	Barrière montée entrée	-		bit 1	Barrière descendue entrée	-		bit 2	Barrière montée sortie	-		bit 3	Barrière descendue sortie	-		bit 4	Bouton d'appel appuyé	-	0x0032 (lecture seule)	bit0-bit7	Place occupée	Place libre	0x0033 (lecture seule)	bit0-bit7	Place occupée	Place libre	0x0034 (lecture seule)	Compteur des places libres non abonnés			0x0035 (lecture seule)	Compteur des places libres abonnés		
Adresse modBus (hexa)	Désignation	1	0																																																																
0x0010 - 0x001F (Écriture seule)	Texte affichage ligne supérieure (16 caractères)																																																																		
0x0020 - 0x002F (Écriture seule)	Texte affichage ligne inférieure (16 caractères)																																																																		
0x0030 (Écriture seule)	bit 0	Ordre monter barrière entrée	-																																																																
	bit 1	Ordre descendre barrière entrée	-																																																																
	bit 2	Ordre monter barrière sortie	-																																																																
	bit 3	Ordre descendre barrière sortie	-																																																																
0x0031 (lecture seule)	bit 0	Barrière montée entrée	-																																																																
	bit 1	Barrière descendue entrée	-																																																																
	bit 2	Barrière montée sortie	-																																																																
	bit 3	Barrière descendue sortie	-																																																																
	bit 4	Bouton d'appel appuyé	-																																																																
0x0032 (lecture seule)	bit0-bit7	Place occupée	Place libre																																																																
0x0033 (lecture seule)	bit0-bit7	Place occupée	Place libre																																																																
0x0034 (lecture seule)	Compteur des places libres non abonnés																																																																		
0x0035 (lecture seule)	Compteur des places libres abonnés																																																																		
3	2	Éclairage	<p>Chaque lampadaire est commandé par le micro contrôleur en PWM, par une sortie avec un convertisseur 8 bits. Seules les valeurs 0,128, 255 seront retenue. Bien que le micro contrôleur commande indépendamment chaque lampadaire, la gestion sera centralisée par secteur. Pour un secteur donné, un client pourra éteindre, allumer à moitié, allumer à pleine puissance tous les lampadaires.</p> <table border="1"> <thead> <tr> <th>Adresse modBus (hexa)</th> <th>Bit</th> <th>Si 1</th> <th>Si 0</th> </tr> </thead> <tbody> <tr> <td rowspan="3">0x0000 (Écriture seule)</td> <td>0</td> <td>Ordre allumer secteur</td> <td>Ordre éteindre secteur</td> </tr> <tr> <td>1</td> <td>100% (éclairage maxi)</td> <td>50% (éclairage diminué)</td> </tr> <tr> <td>2</td> <td>Avec mode économe</td> <td>Sans mode économe</td> </tr> <tr> <td rowspan="3">0x0001 (Lecture seule)</td> <td>0</td> <td>Présence piéton</td> <td>Absence piéton</td> </tr> <tr> <td>1</td> <td>Détection jour</td> <td>Détection nuit</td> </tr> <tr> <td>2</td> <td>Lampadaire(s) défectueux</td> <td>RAS</td> </tr> </tbody> </table> <p>Note 2 : Ce plan mémoire est valable pour chaque secteur d'éclairage.</p>	Adresse modBus (hexa)	Bit	Si 1	Si 0	0x0000 (Écriture seule)	0	Ordre allumer secteur	Ordre éteindre secteur	1	100% (éclairage maxi)	50% (éclairage diminué)	2	Avec mode économe	Sans mode économe	0x0001 (Lecture seule)	0	Présence piéton	Absence piéton	1	Détection jour	Détection nuit	2	Lampadaire(s) défectueux	RAS																																								
Adresse modBus (hexa)	Bit	Si 1	Si 0																																																																
0x0000 (Écriture seule)	0	Ordre allumer secteur	Ordre éteindre secteur																																																																
	1	100% (éclairage maxi)	50% (éclairage diminué)																																																																
	2	Avec mode économe	Sans mode économe																																																																
0x0001 (Lecture seule)	0	Présence piéton	Absence piéton																																																																
	1	Détection jour	Détection nuit																																																																
	2	Lampadaire(s) défectueux	RAS																																																																
4	3	Intersection	A développer.																																																																

2.1.4 Architectures Matérielle & Logicielle

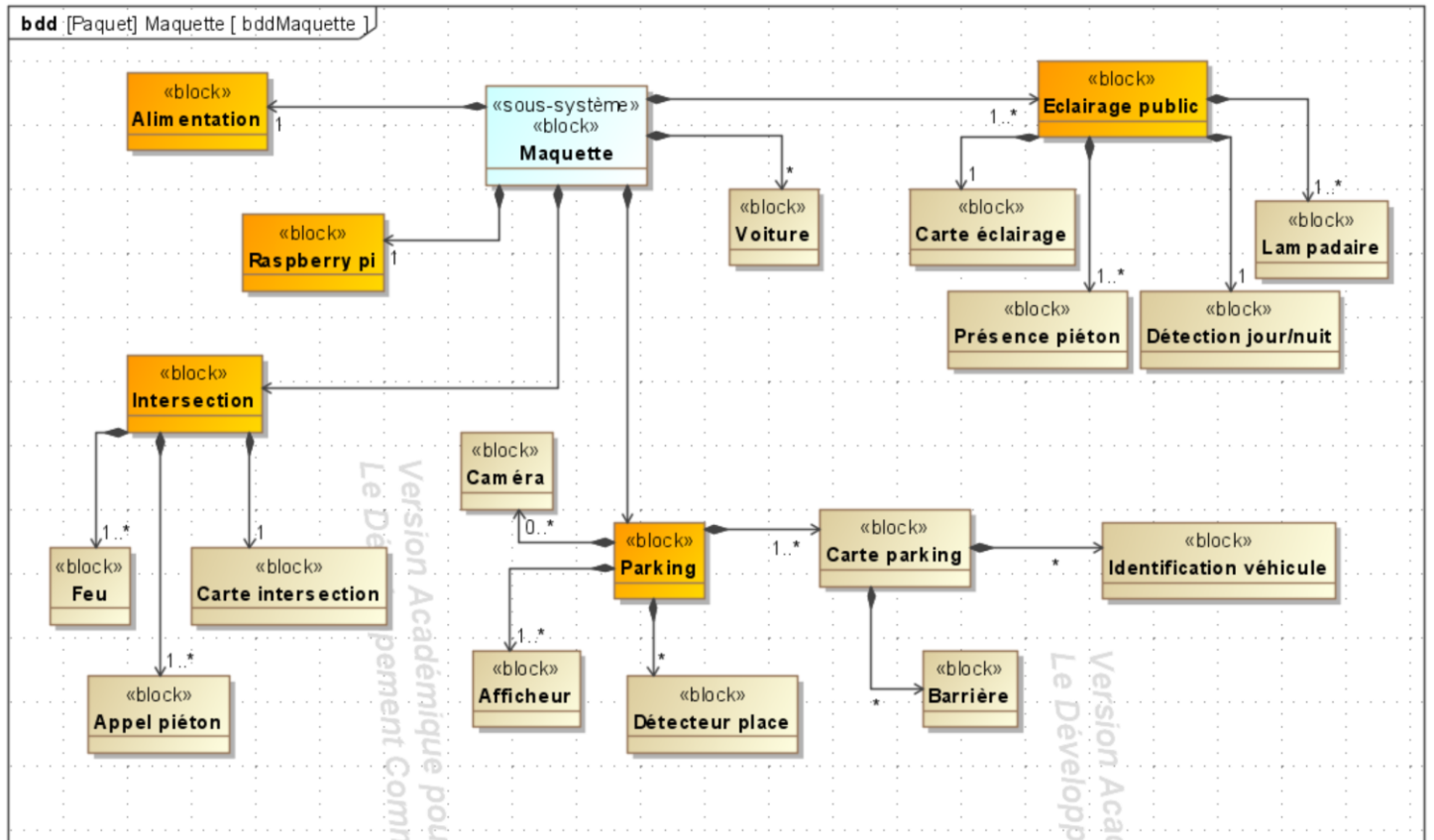


Figure 5 : Diagramme des blocs du SS Maquette

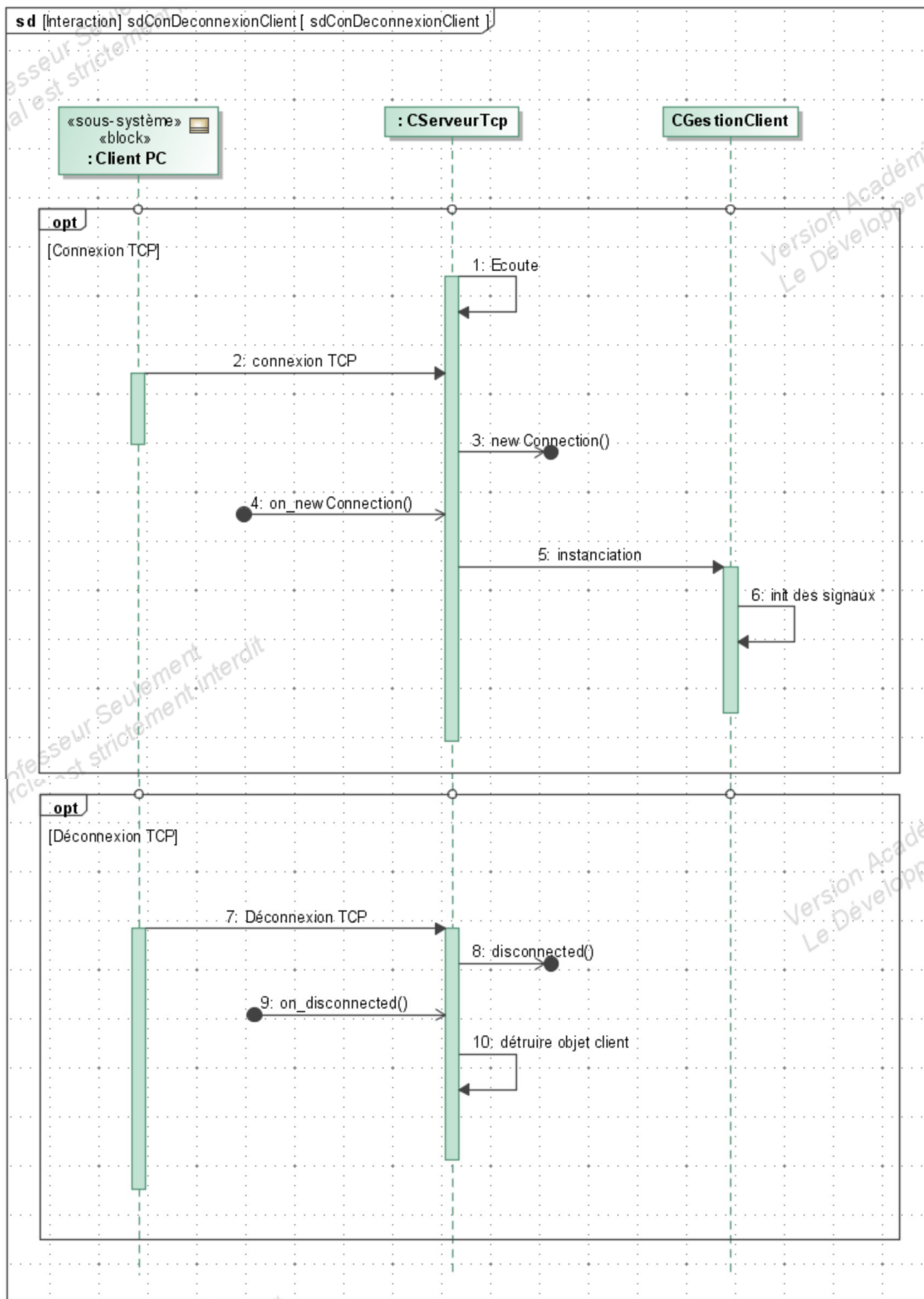


Figure 6 : Diagramme de séquence de la connexion / déconnexion d'un client

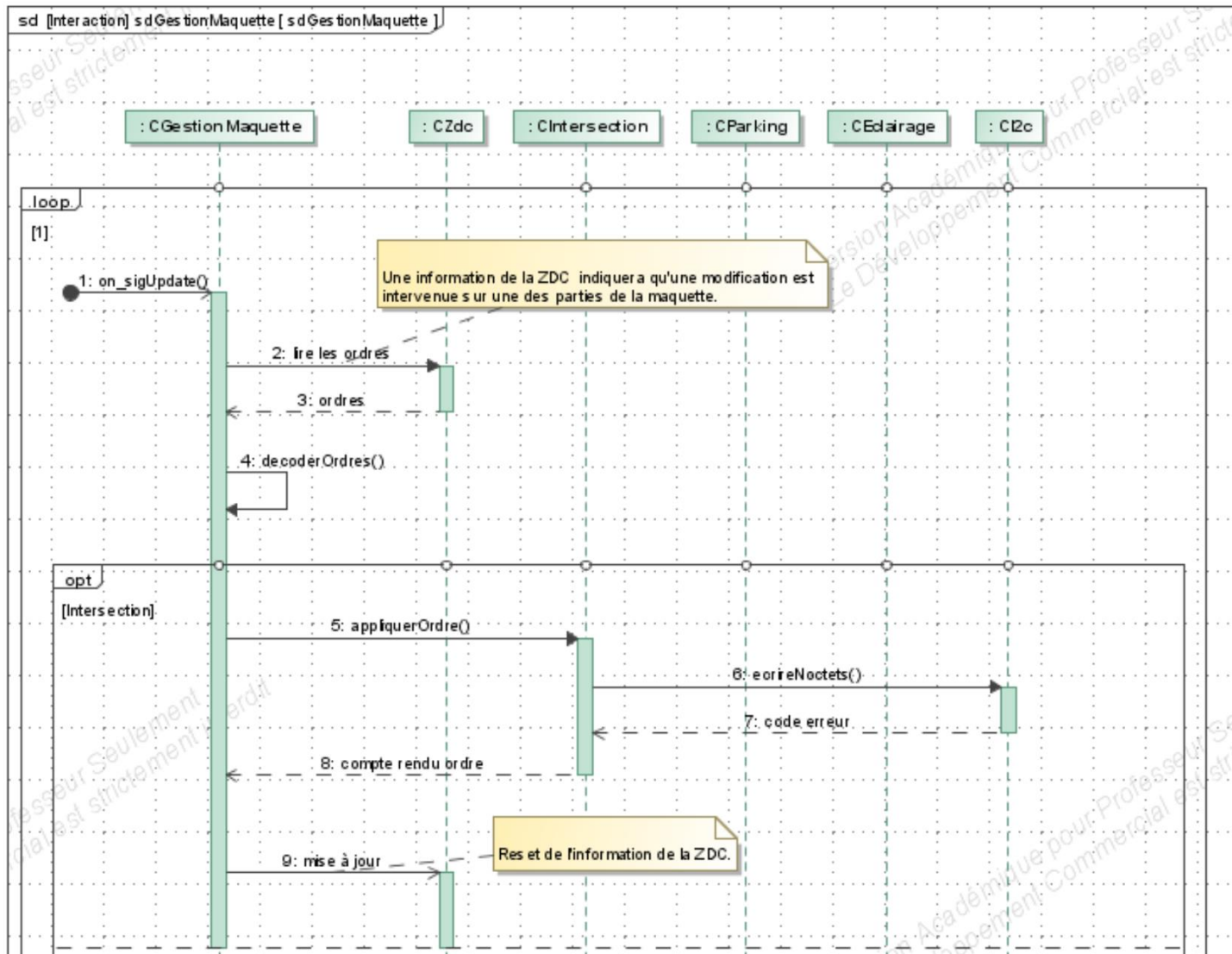


Figure 7 : Diagramme de séquence des échanges I2C

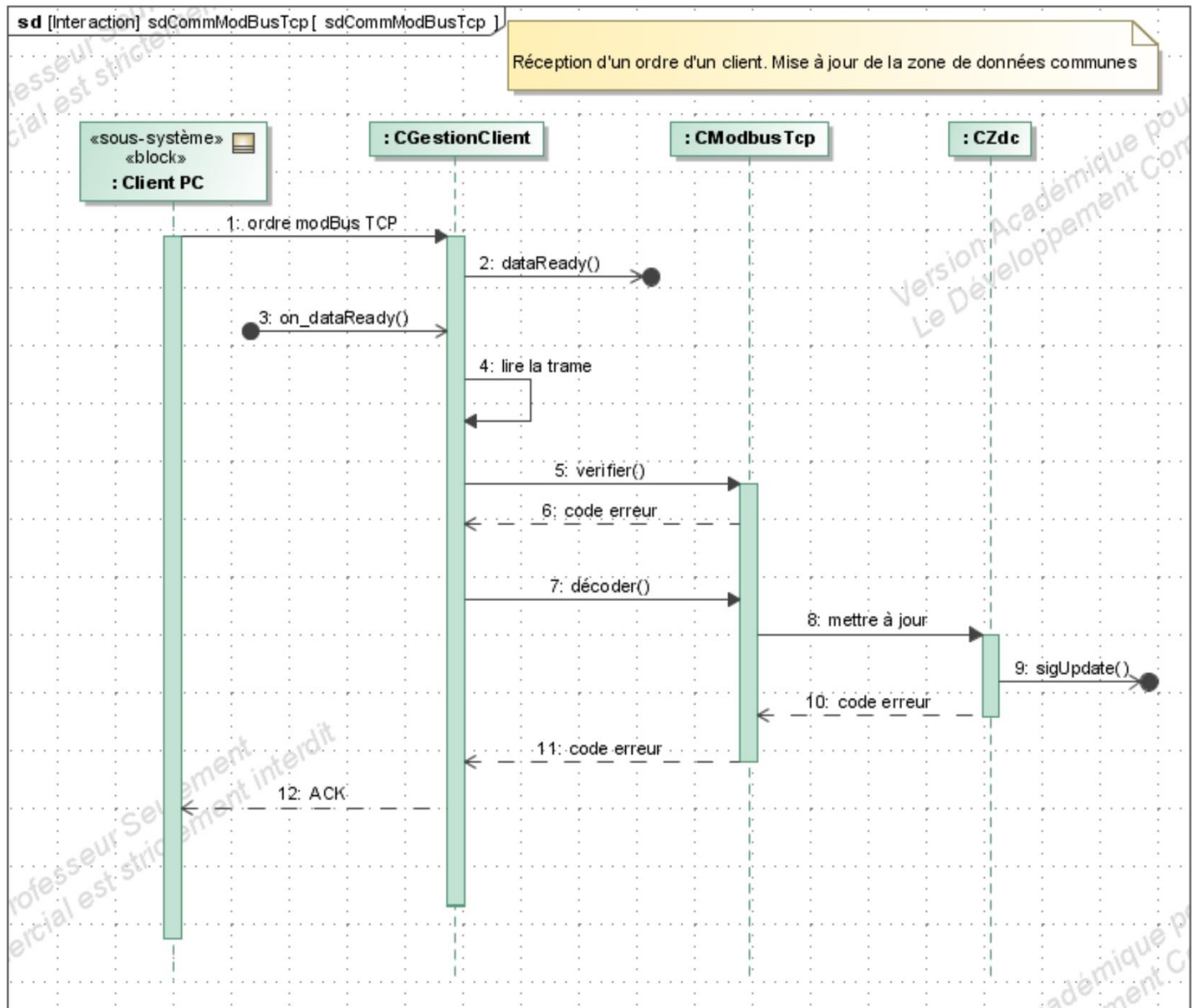


Figure 8 : Diagramme de séquence de la communication modBus

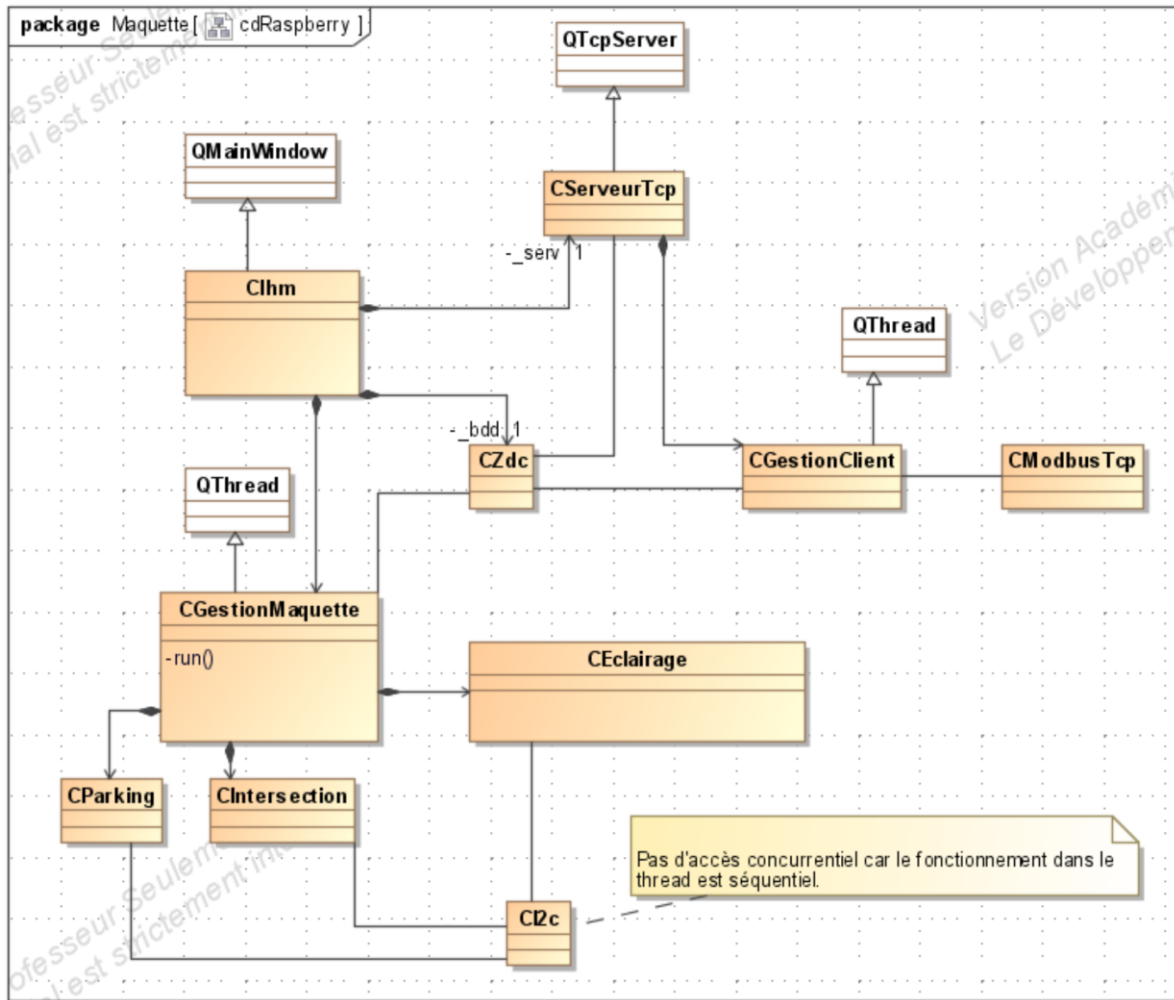


Figure 9 : Diagramme de classes du logiciel Raspberry

2.1.5 Scénarios des cas d'utilisation

Sous système : Maquette	UC : Superviser WEB
Scénario nominal	
Serveur http/https permettant la supervision distante et graphique de la maquette sous forme de page WEB.	
Sous système : Maquette	UC : Commander
Scénario nominal	
Permet d'émettre des ordres depuis le site WEB vers les éléments de la ville. Ce scénario est optionnel. Il sera implémenté que si le reste fonctionne.	
Sous système : Maquette	UC : Authentifier
Scénario nominal	
Demande une authentification nom+mot de passe. Un seul compte sera nécessaire.	
Sous système : Maquette	UC : Communiquer avec clients
Scénario nominal	
Serveur TCP gérant les clients sous forme de threads. Pour chaque requête d'un client, mettre à jour la zone de données commune et envoyer un signal à l'objet concerné. Répondre au client.	
Cas particulier des caméras qui seront directement accessibles par leur adresse IP et numéro de port. Le flux vidéo est généré par chaque caméra.	
Sous système : Maquette	UC : Gérer XXX
Scénario nominal	
XXX : Elément de la ville (intersection, parking, éclairage).	

Dès réception d'un signal, lire les données dans la ZDC.
 Rechercher les modifications à appliquer sur l'élément de la ville.
 Contrôler la communication avec les éléments de la ville par le bus I2C.

Sous système : Client PC	UC : Contrôler en manu
Scénario nominal	
Permet par un logiciel muni d'une IHM (interface homme machine) de superviser et contrôler les éléments de la ville. Note : Bien que cette partie soit normalement à la charge des étudiant de Master 2, nous développerons une version de test des éléments de la ville.	

Sous système : Client PC	UC : Communiquer avec maquette
Scénario nominal	
Permet la communication avec la maquette suivant les protocoles : <ul style="list-style-type: none"> • modBus TCP pour superviser et contrôler les éléments de la ville. • http/https/ftp, etc. pour récupérer les flux des caméras et les intégrer dans l'IHM. 	

2.2 Contraintes de réalisation

Contraintes financières (budget alloué) :

Budget estimé de 500€ selon avancement du projet à charge partielle du demandeur.

Contraintes de développement (matériel et/ou logiciel imposé / technologies utilisées) :

La spécification, conception et codage seront modélisés.

Contraintes qualité (conformité, délais, ...) :

Maintenable, maniable (ergonomie)

Contraintes de fiabilité, sécurité :

Les accès logiciels seront sécurisés.

2.3 Ressources mises à disposition des étudiants (logiciels / matériels / documents)

Matériels :

- Composants pour la réalisation
- Matériel de laboratoire (alimentation, oscilloscope, analyseur logique)
- Cartes de développement pour ATTiny 3217 et 1617

Logiciels :

- Système d'exploitation
- Logiciel de modélisation SysML/UML : MagicDraw v7.02
- Logiciel de conception électronique : KiCAD 5
- Logiciel de conception électronique Fritzing uniquement pour illustrer le prototypage rapide
- Un logiciel de saisie de schéma et de simulation (Proteus ISIS) pourra éventuellement être utilisé pour illustrer des essais de programmation.

Documents :

- Site de la STS SN mettant à disposition les différentes documentations.

3 Répartition des tâches par étudiant

<p>Étudiant 1</p> <p>IR 1</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <ul style="list-style-type: none"> • Mise au point du logiciel RASPBERRY. <ul style="list-style-type: none"> ○ Fonctionnement de l'application ○ Communications I2C. ○ Mémoire partagée 	<p>Installation : Système d'exploitation Raspbian, EDI.</p> <p>Mise en œuvre : Thread, TCP, communication réseau serveur.</p> <p>Configuration : I2c, Thread, mémoire partagée. Gestion de l'afficheur du parking.</p> <p>Réalisation : Logiciel Qt C++ avec Qt Creator</p> <p>Documentation : Manuel d'installation et mise en œuvre.</p>
<p>Étudiant 2</p> <p>IR 2</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <ul style="list-style-type: none"> • Mise au point du logiciel RASPBERRY et du client PC. <ul style="list-style-type: none"> ○ Communications TCP/IP ○ Base de données 	<p>Installation : Système d'exploitation Raspbian, EDI.</p> <p>Mise en œuvre : Modbus, Thread, mémoire partagée. Gestion de l'afficheur du parking. BDD</p> <p>Configuration : EDI, TCP/IP, SGBD</p> <p>Réalisation : Logiciel Qt C++ avec Qt Creator</p> <p>Documentation : Cahier de recettes, défaillances.</p>
<p>Étudiant 3</p> <p>EC 1</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <p style="text-align: center;">Feux d'intersection</p> <ul style="list-style-type: none"> • Analyser et mettre en œuvre la carte développée lors de la session 2021. • Des améliorations doivent être apportées sur cette carte, telles que l'utilisation d'un régulateur à découpage CMS, la conception de feux tricolores qui soient davantage visibles, et une mise en œuvre d'un outil de programmation du microcontrôleur (<i>Bus UPDI</i>) plus simple d'emploi. • Modifier le schéma structurel de la carte en conséquence. Un PCB spécifique avec LED CMS pour les feux sera envisagé. • Effectuer la saisie du schéma et le routage des solutions proposées. Produire les fichiers Gerber afin que la fabrication du PCB soit sous-traitée. • Câbler les cartes et effectuer les essais. • Documenter la mise en service de l'ensemble finalisé. 	<p>Installation : IDE Arduino avec bibliothèques megaTinyCore pour la programmation d'un ATtiny1617 ou 3217.</p> <p>Mise en œuvre : Tester/valider les modifications souhaitées.</p> <p>Réalisation : Après validation des solutions, le circuit imprimé v2021 sera modifié. Il devra être fabriqué industriellement.</p> <p>Documentation :</p> <ul style="list-style-type: none"> • Schéma de câblage rapide (Fritzing) pour documenter la phase d'essais. • Documents de fabrication de la carte (KiCAD). Ces documents devront avoir un niveau de qualité permettant une fabrication industrielle du circuit imprimé. • Schéma structurel avec contours IBD. • Liste complète des composants avec leurs sources d'approvisionnement et leur prix. • Programme en C/C++ de communication sur le bus I2C, et de détection des impacts, accompagné des commentaires et diagrammes nécessaires à sa compréhension. • Fiche de mise en service. • Fiche de dépannage.
<p>Étudiant 4</p> <p>EC 2</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <p style="text-align: center;">Gestion des places libres/occupées du parking</p> <ul style="list-style-type: none"> • Analyser et mettre en œuvre la carte développée lors de la session 2021. • Des améliorations doivent être apportées sur cette carte, telles que l'utilisation d'un régulateur à découpage CMS, l'ajout d'un microcontrôleur (ATtiny1617 ou 3217) 	<p>Installation : IDE Arduino avec bibliothèques megaTinyCore pour la programmation d'un ATtiny1617 ou 3217.</p> <p>Mise en œuvre : Tester/valider les modifications souhaitées.</p> <p>Réalisation : Après validation des solutions, modifier le circuit imprimé v2021. Il devra être fabriqué industriellement.</p>

	<p>communiquant en esclave sur le bus I2C et qui fera remonter l'état d'occupation des places du parking au Raspberry Pi, la carte devra être adaptée pour l'utilisation d'un programmeur simple d'emploi (<i>Bus UPDI</i>) .</p> <ul style="list-style-type: none"> • Modifier le schéma structurel de la carte en conséquence. • Effectuer la saisie du schéma et le routage des solutions proposées. Produire les fichiers Gerber afin que la fabrication du PCB soit sous-traitée. • Câbler la carte et effectuer les essais. • Documenter la mise en service de l'ensemble finalisé. 	<p>Documentation :</p> <ul style="list-style-type: none"> • Schéma de câblage rapide (Fritzing) pour documenter la phase d'essais. • Documents de fabrication de la carte (KiCAD). Ces documents devront avoir un niveau de qualité permettant une fabrication industrielle du circuit imprimé. • Schéma structurel avec contours IBD. • Liste complète des composants avec leurs sources d'approvisionnement et leur prix. • Programme en C/C++ de communication sur le bus I2C, et de détection des impacts, accompagné des commentaires et diagrammes nécessaires à sa compréhension. • Fiche de mise en service. • Fiche de dépannage.
Tous les étudiants	<p>✓ <i>Tâches à traiter par l'ensemble des étudiants de l'équipe projet pour le développement de la solution</i></p> <ul style="list-style-type: none"> • Documents de vie du projet : <ul style="list-style-type: none"> - Fiches de lecture croisée - Comptes rendus de réunion. <p>✓ <i>Domaines de physique à traiter par l'ensemble des étudiants de l'équipe projet :</i></p> <ul style="list-style-type: none"> ○ Traitement du signal analogique ○ Signaux : Puissance et énergie ○ Numérisation de signaux ○ Lignes de transmission 	<p>Intégration de la solution et livraison au client du matériel/logiciel/sources/manuels.</p> <p>Le logiciel sera installable facilement chez le client en suivant une procédure écrite.</p> <ul style="list-style-type: none"> • Un dossier de recettes sera fourni.

4 Exploitation Pédagogique – Compétences terminales évaluées :

Pour chaque équipe :

	Électronique et Communications	Informatique et Réseaux	IR 1	IR 2	EC1	EC2
C2.1	Maintenir les informations		X	X	X	X
C2.2	Formaliser l'expression du besoin		X	X	X	X
C2.3	Organiser et/ou respecter la planification d'un projet		X	X	X	X
C2.4	Assumer le rôle total ou partiel de chef		X	X	X	X
C2.5	Travailler en équipe		X	X	X	X
C3.1	Analyser un cahier des charges		X	X	X	X
C3.3	Définir l'architecture globale d'un prototype ou d'un système		X	X	X	X
C3.5	Contribuer à la définition des éléments de recette au regard des contraintes du cahier des charges		X	X	X	X
C3.6	Recenser les solutions existantes répondant au cahier des charges		X	X	X	X
C3.8	Élaborer le dossier de définition de la solution techniquement				X	X
C3.9	Valider une fonction du système à partir d'une maquette réelle				X	X
C3.10	Réaliser la conception détaillée d'un module matériel et/ou logicielle				X	X
C4.1	Câbler et/ou intégrer un matériel		X	X	X	X
C4.2	Adapter et/ou configurer un matériel		X	X	X	X
C4.3	Adapter et/ou configurer une structure logicielle	Installer et configurer une chaîne de développement	X	X	X	X
C4.4	Fabriquer un sous ensemble	Développer un module logiciel	X	X	X	X
C4.5	Tester et valider un module logiciel et matériel	Tester et valider un module logiciel	X	X	X	X
C4.6	Produire les documents de fabrication d'un sous ensemble	Intégrer un module logiciel	X	X	X	X
C4.7	Documenter une réalisation matérielle / logicielle		X	X	X	X

5 Planification

Début du projet (Dp)	semaine 2	: 04/01/2022.
Revue 1 (R1)	semaine 10	: à partir du 21/02/2022.
Revue 2 (R2)	semaine 18	: à partir du 25/04/2022
Remise du projet (Rp)	semaine 21	: 28/05/2022 (date limite de remise du dossier sur l'espace académique)
Soutenance finale (Sf)	semaine 24	: à partir du 13/06/2022.

6 Conditions d'évaluation pour l'épreuve E6-2

6.1 Disponibilité des équipements

L'équipement sera-t-il disponible ?

Oui

Non

6.2 Atteintes des objectifs du point de vue client

Que devra-t-on observer à la fin du projet qui témoignera de l'atteinte des objectifs fixés, du point de vue du client :

L'étudiant devra être capable de mettre en œuvre les tâches dont il est en charge.

Dans le meilleur des cas : l'intégration et les cas d'utilisation seront opérationnels, en respectant les contraintes.

6.3 Avenants :

Date des avenants : Nombre de pages :

7 Observation de la commission de Validation

Ce document initial : **comprend 19 pages.**

(À remplir par la commission de validation qui valide le sujet de projet)

a été utilisé par la Commission Académique de validation qui s'est réunie à Gardannes, le 30/11/2022

Contenu du projet :	Défini	Insuffisamment défini	Non défini
Problème à résoudre :	Cohérent techniquement		Pertinent / À un niveau BTS SN
Complexité technique : (liée au support ou au moyen utilisés)	Suffisante	Insuffisante	Exagérée
Cohérence pédagogique : (relative aux objectifs de l'épreuve)	Le projet permet l'évaluation de toutes les compétences terminales Chaque candidat peut être évalué sur chacune des compétences		
Planification des tâches demandées aux étudiants, délais prévus, ... :	Projet ... Défini et raisonnable	Insuffisamment défini	Non défini
Les revues de projet sont-elles prévues : (dates, modalités, évaluation)	Oui	Non	
Conformité par rapport au référentiel et à la définition de l'épreuve :	Oui	Non	

Observations :

.....

7.1 Avis formulé par la commission de validation :

Sujet accepté
en l'état

Sujet à revoir :

Conformité au Référentiel de Certification / Complexité
 Définition et planification des tâches
 Critères d'évaluation
 Autres :

Sujet rejeté

Motif de la commission :

.....

7.2 Nom des membres de la commission de validation académique :

Nom	Établissement	Académie	Signature

7.3 Visa de l'autorité académique :

(nom, qualité, Académie, signature)

Nota :

Ce document est contractuel pour la sous-épreuve E6-2 (Projet Technique) et sera joint au « Dossier Technique » de l'étudiant. En cas de modification du cahier des charges, un avenant sera élaboré et joint au dossier du candidat pour présentation au jury, en même temps que le carnet de suivi.