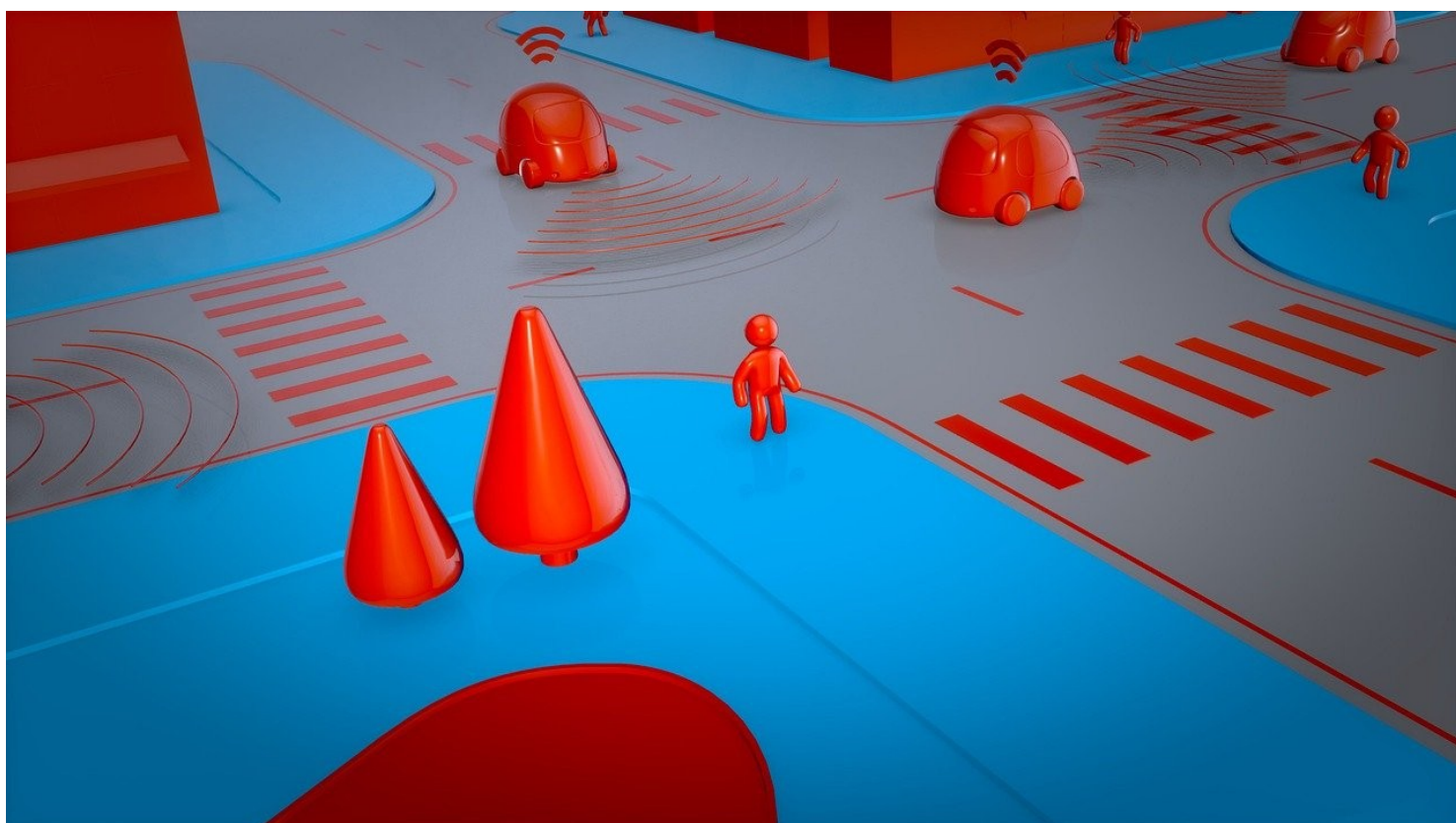




Groupement académique : AIX-MARSEILLE		Session 2021
Lycée : Alphonse BENOIT		
Ville : L'ISLE SUR LA SORGUE		
N° du projet : 2 et 3	Nom du projet : SmartCity : Gestion intelligente d'une ville	

Projet nouveau	<input checked="" type="checkbox"/> Oui	<input type="checkbox"/> Non	Projet interne	<input type="checkbox"/> Oui	<input checked="" type="checkbox"/> Non
Délai de réalisation	06/01/2021 au 30/05/2021		Statut des étudiants	<input checked="" type="checkbox"/> Formation initiale	<input type="checkbox"/> Apprentissage
Spécialité des étudiants	<input type="checkbox"/> EC	<input type="checkbox"/> IR	<input checked="" type="checkbox"/> Mixte	Nombre d'étudiants 1 équipe de 5 et 1 équipe de 4.	
Professeurs responsables	ANTOINE / DEFRANCE / ESCURET / HORTOLLAND / SILANUS				



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.....	15
3.1	Equipe 1.....	15
3.2	Equipe 2.....	17
4	Exploitation Pédagogique – Compétences terminales évaluées :.....	20
5	Planification.....	21
6	Conditions d'évaluation pour l'épreuve E6-2.....	21
6.1	Disponibilité des équipements.....	21
6.2	Atteintes des objectifs du point de vue client.....	21
6.3	Avenants :.....	21
7	Observation de la commission de Validation.....	22
7.1	Avis formulé par la commission de validation :.....	22
7.2	Nom des membres de la commission de validation académique :.....	22
7.3	Visa de l'autorité académique :.....	23

1 Présentation et situation du projet dans son environnement

1.1 Contexte de réalisation

Constitution de l'équipe de projet :	Étudiant 1 EC <input type="checkbox"/> IR <input type="checkbox"/>	Étudiant 2 EC <input type="checkbox"/> IR <input type="checkbox"/>	Étudiant 3 EC <input type="checkbox"/> IR <input type="checkbox"/>	Étudiant 4 <input type="checkbox"/> EC <input type="checkbox"/> IR <input type="checkbox"/>	Étudiant 5 <input type="checkbox"/> EC <input type="checkbox"/> IR <input type="checkbox"/>
Projet développé :	Au lycée ou en centre de formation <input type="checkbox"/> Mixte			En entreprise	
Type de client ou donneur d'ordre (commanditaire) :	Entreprise ou organisme commanditaire : <input type="checkbox"/> Oui Nom : Philippe GOSLAN / Marc SILANUS Adresse : CERI / Université d'Avignon pôle Agroparc Montfavet Contact : Origine du projet : <ul style="list-style-type: none"> • Idée : Lycée <input type="checkbox"/> Entreprise <input type="checkbox"/> • Cahier des charges : Lycée <input type="checkbox"/> Entreprise <input type="checkbox"/> • Suivi du projet : <input type="checkbox"/> Lycée <input type="checkbox"/> Entreprise <input type="checkbox"/> 				
Si le projet est développé en partenariat avec une entreprise :	Nom de l'entreprise : CERI / Adresse de l'entreprise : Université d'Avignon pôle Agroparc 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.

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 :

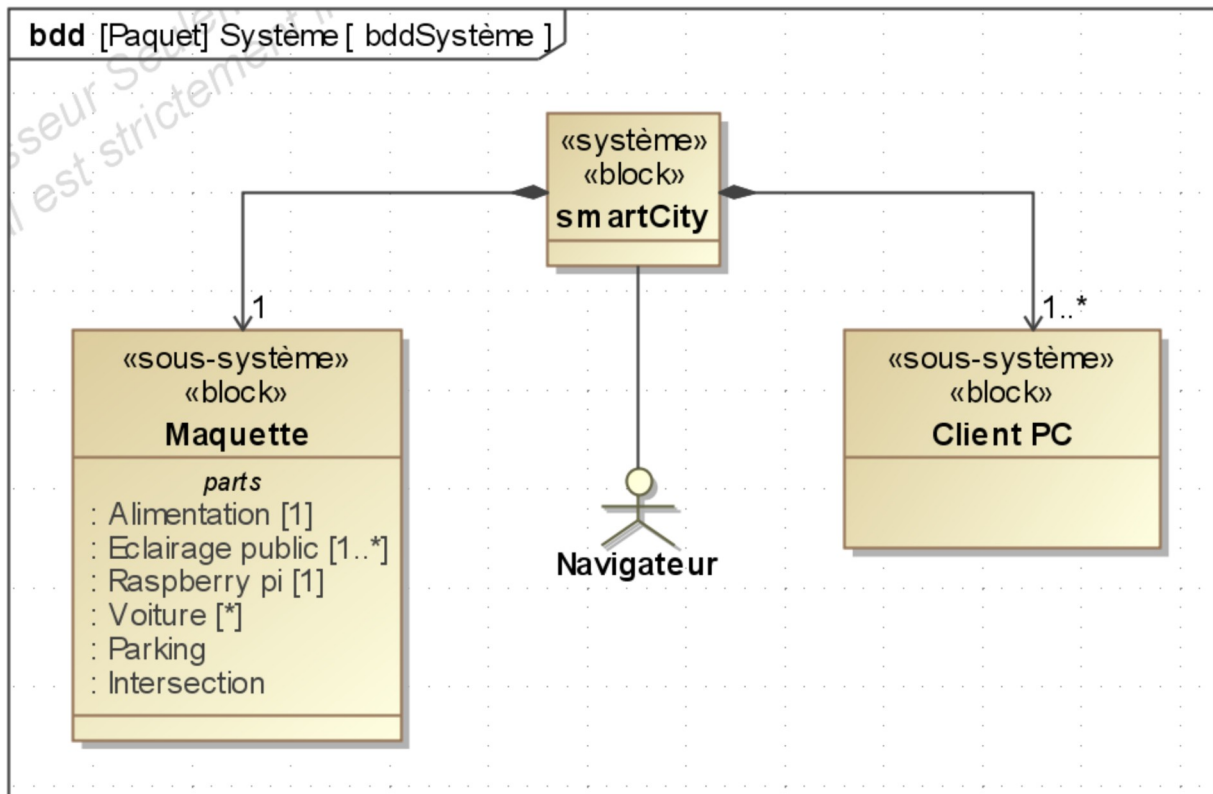


Figure 1 : Diagramme de blocs du système

2.1.1 Diagrammes des cas d'utilisation

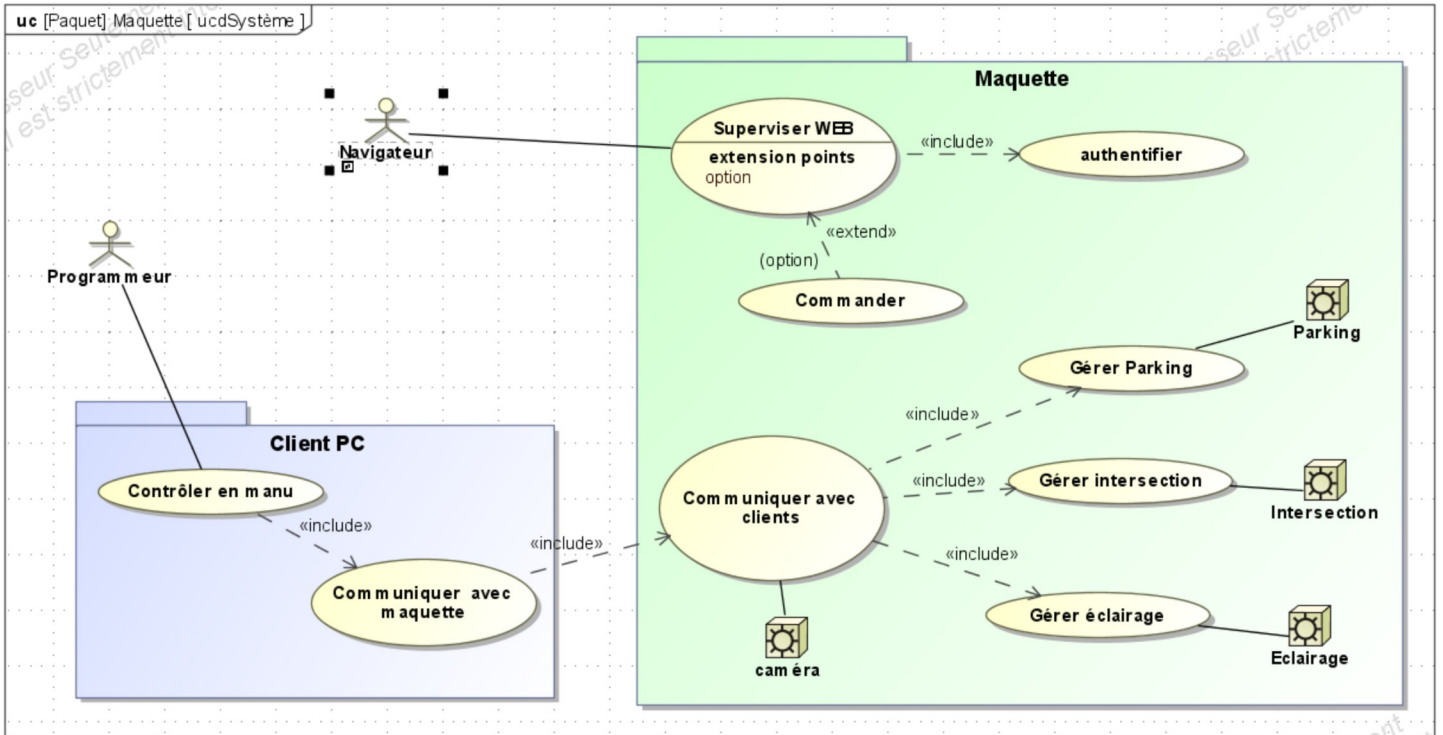


Figure 2 : Diagramme des cas d'utilisation du système

2.1.2 Diagramme de déploiement

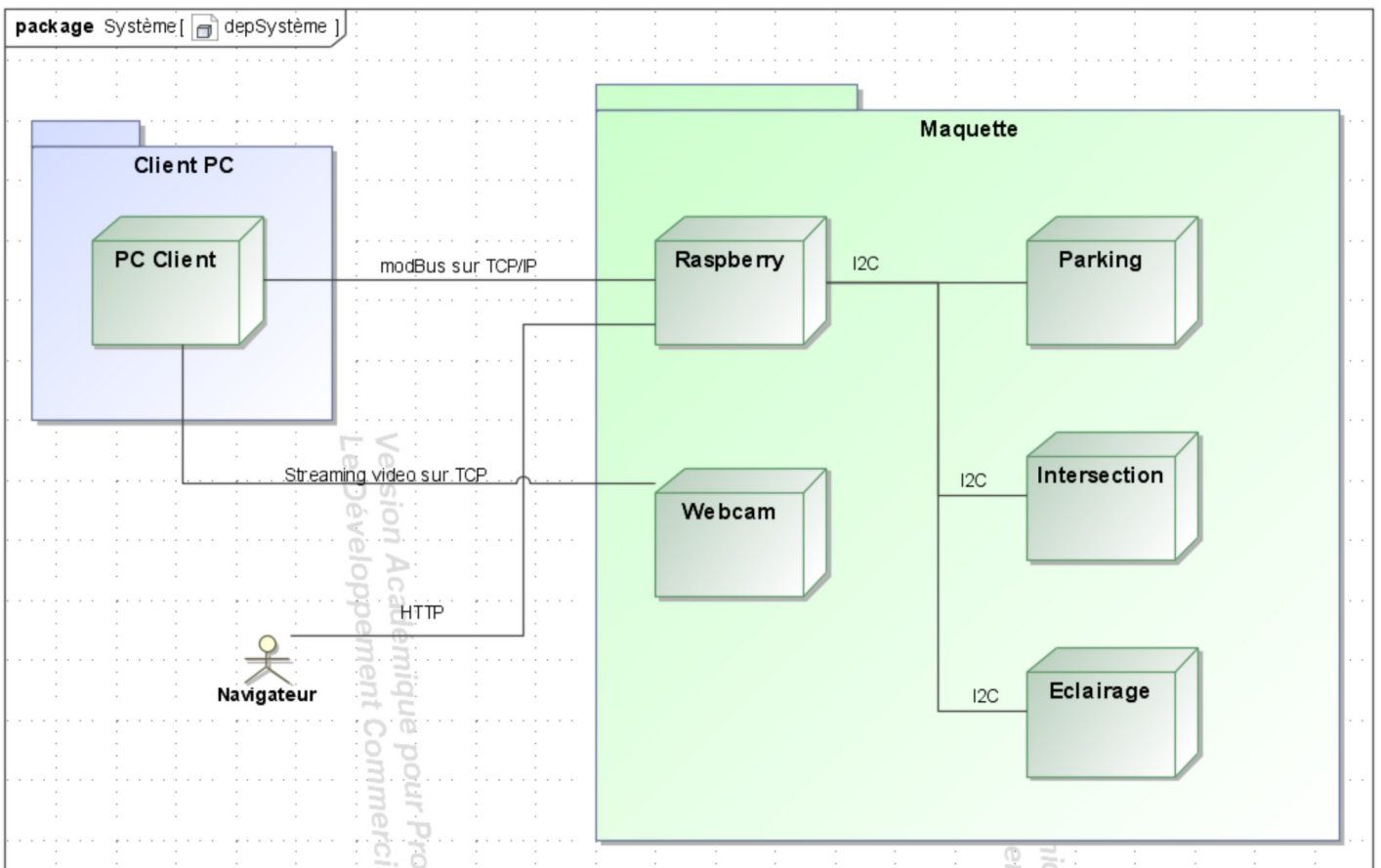


Figure 3 : Diagramme de déploiement du système

2.1.3 Exigences

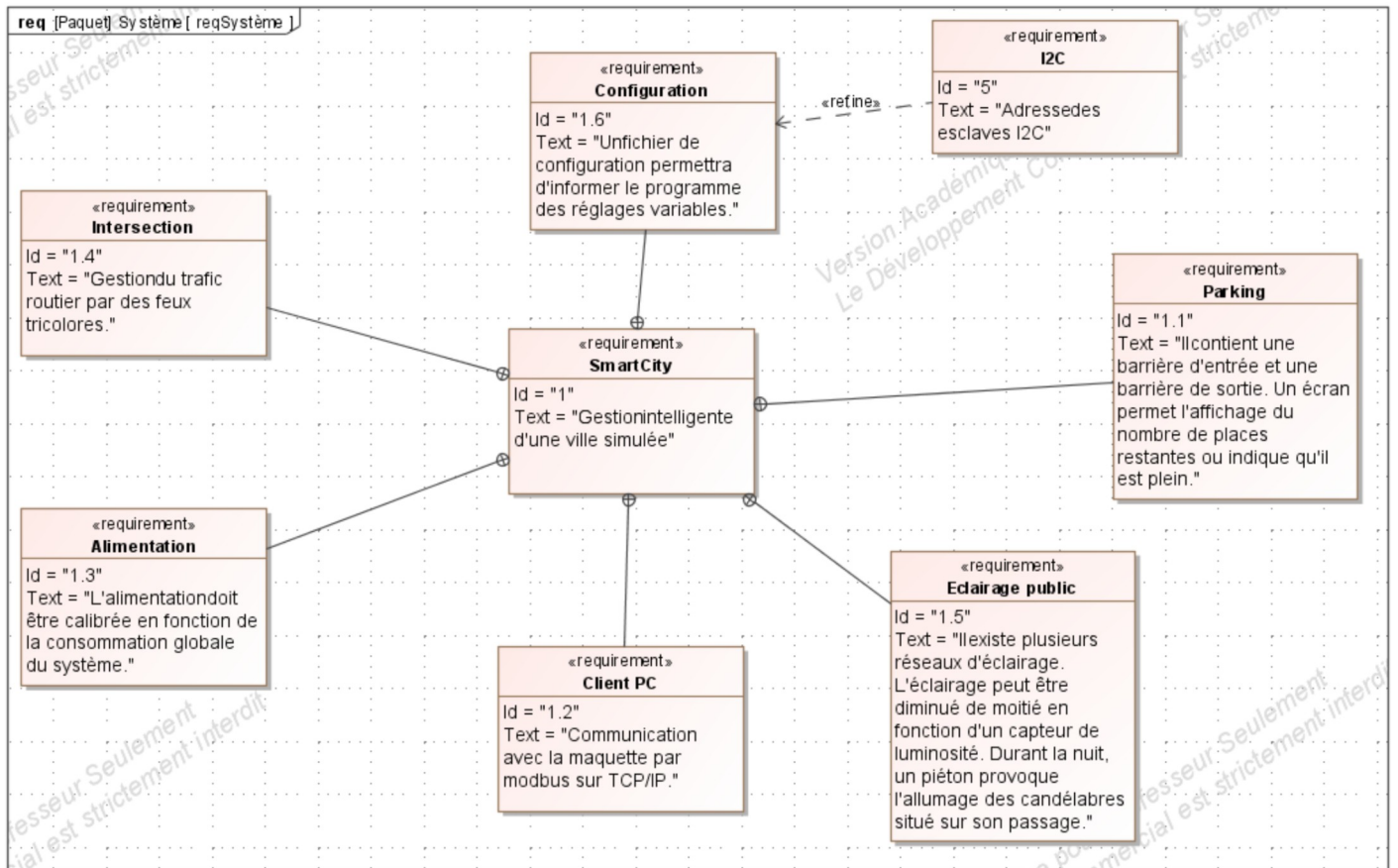


Figure 4 : Diagramme des exigences

#	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

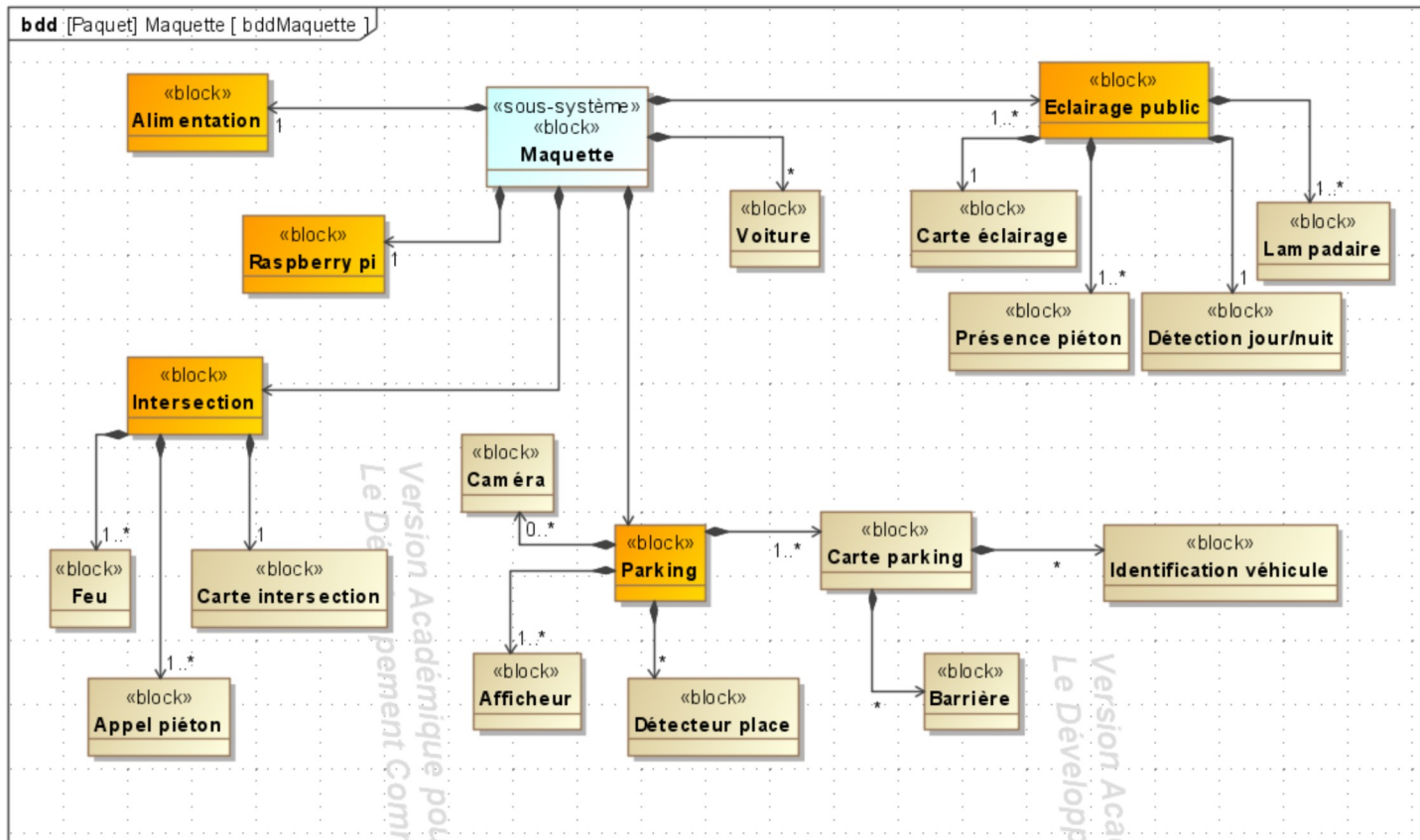


Figure 5 : Diagramme des blocs du SS Maquette

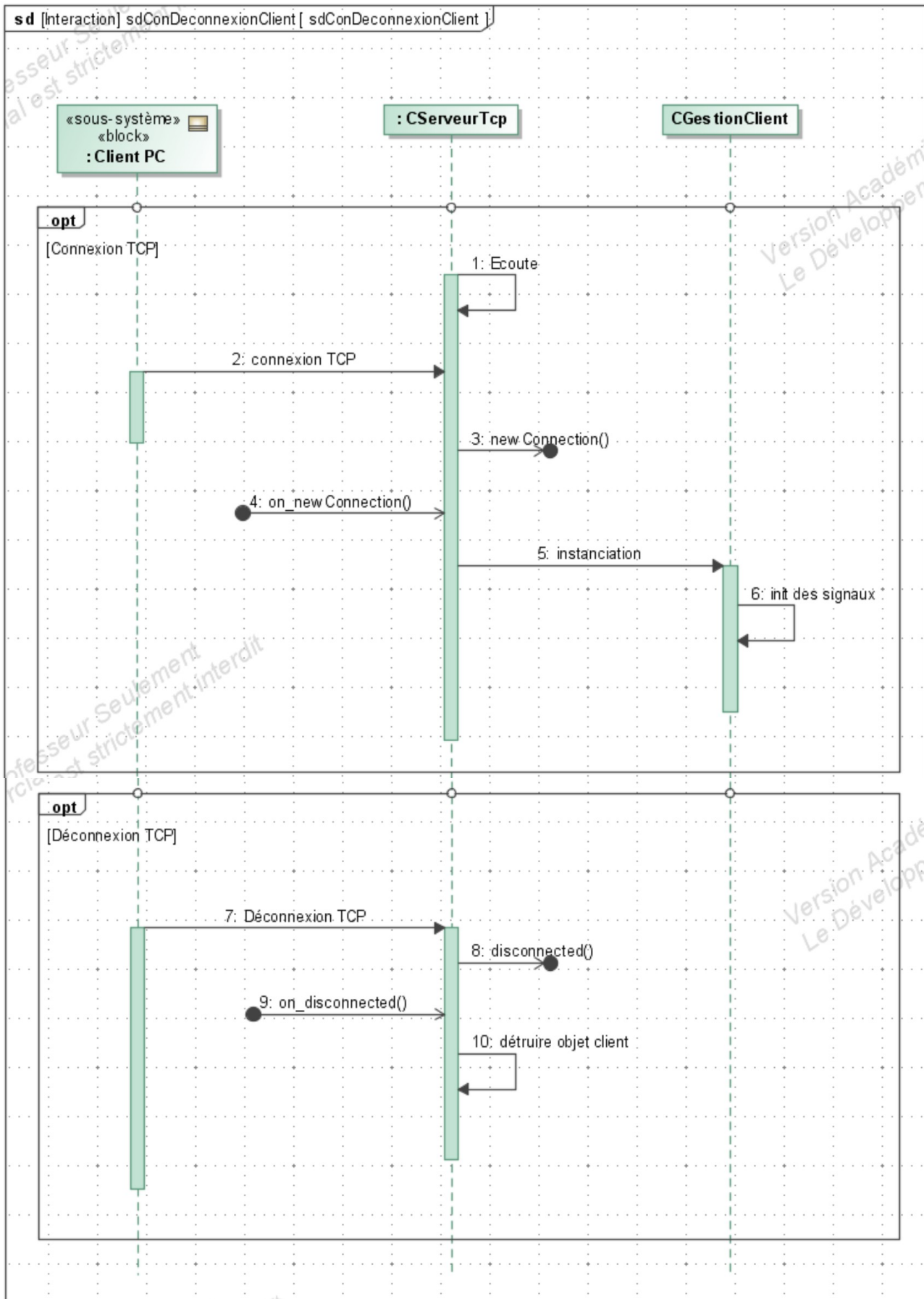


Figure 6 : Diagramme de séquence de la connexion / déconnexion d'un client

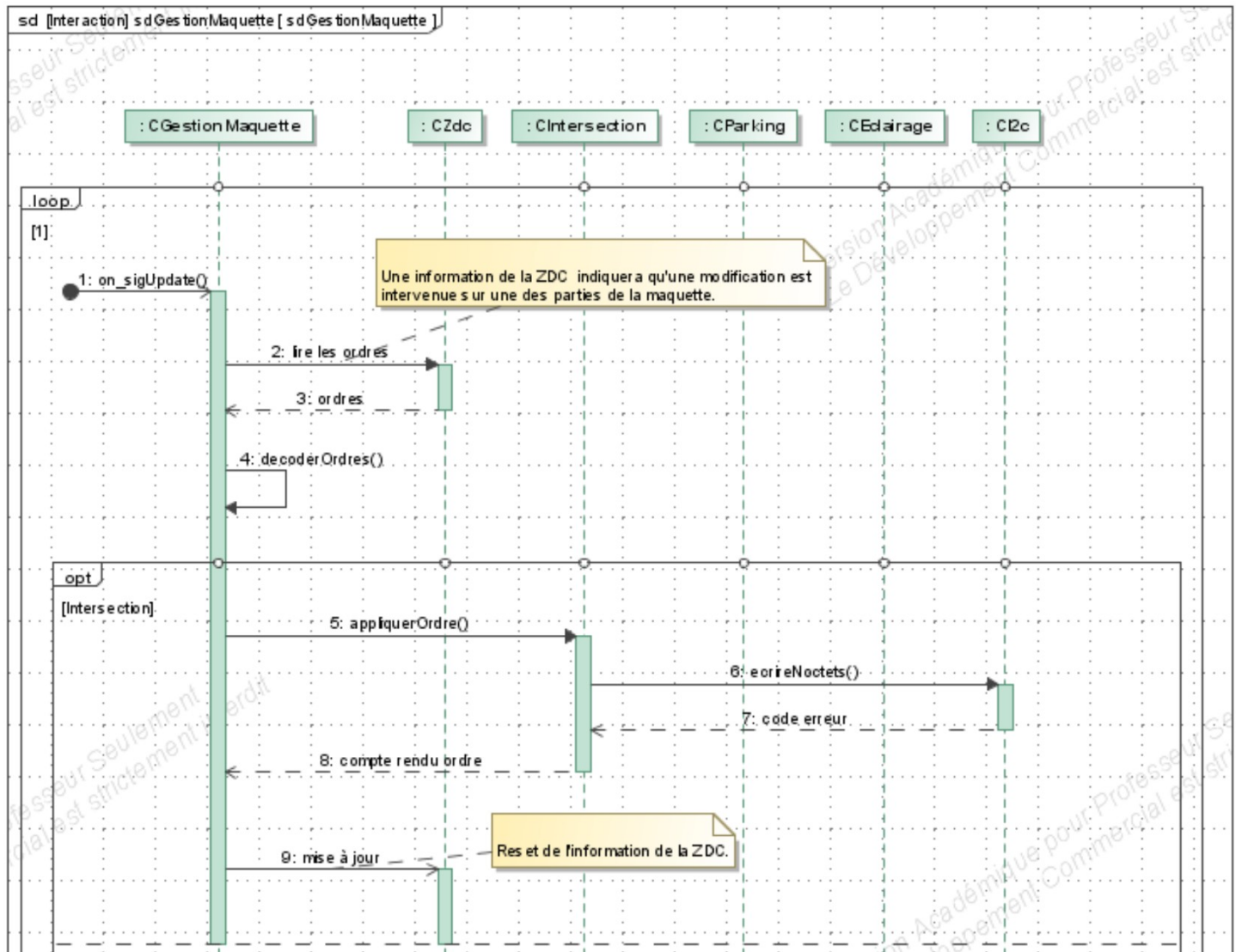


Figure 7 : Diagramme de séquence des échanges I2C

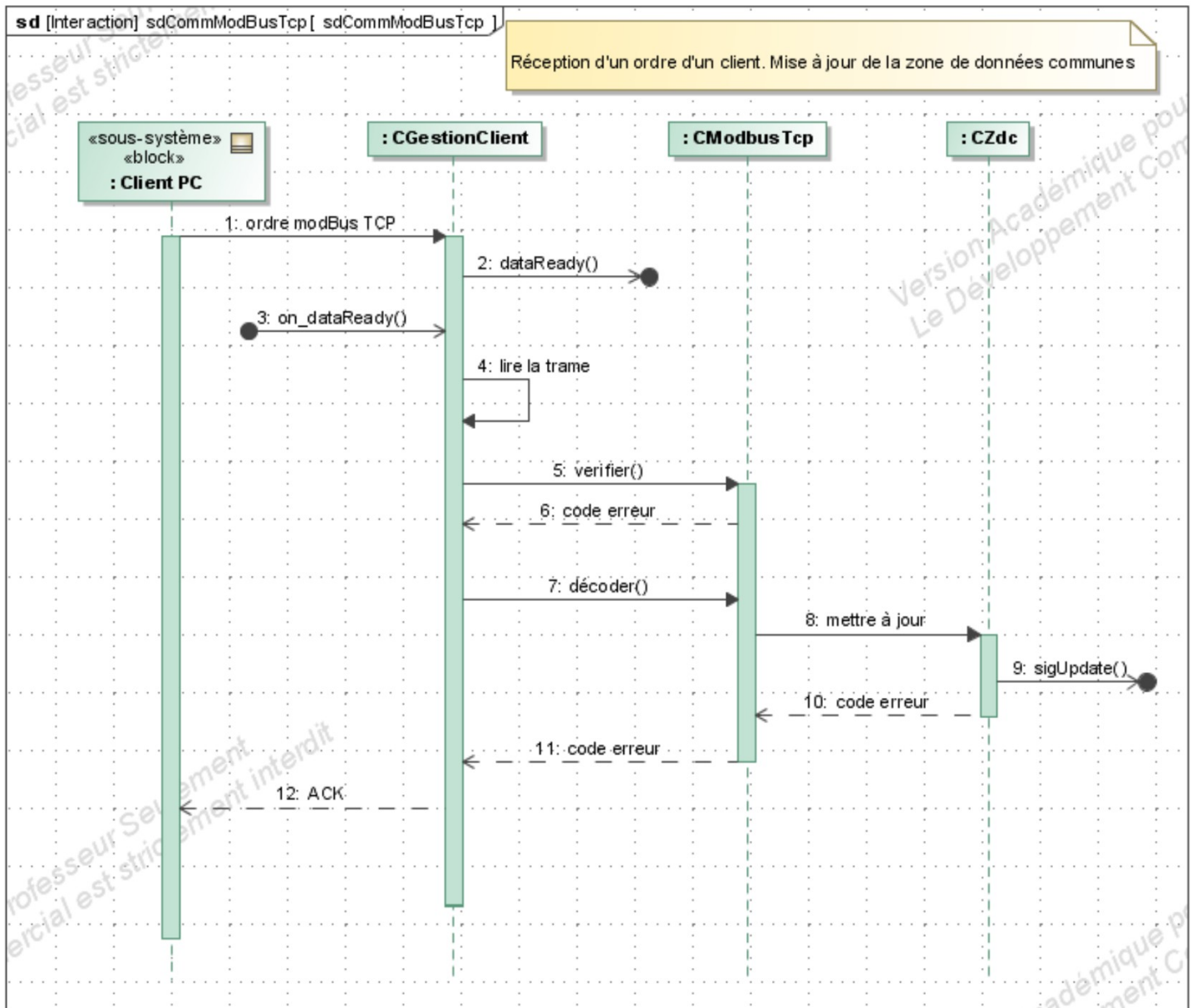


Figure 8 : Diagramme de séquence de la communication modBus

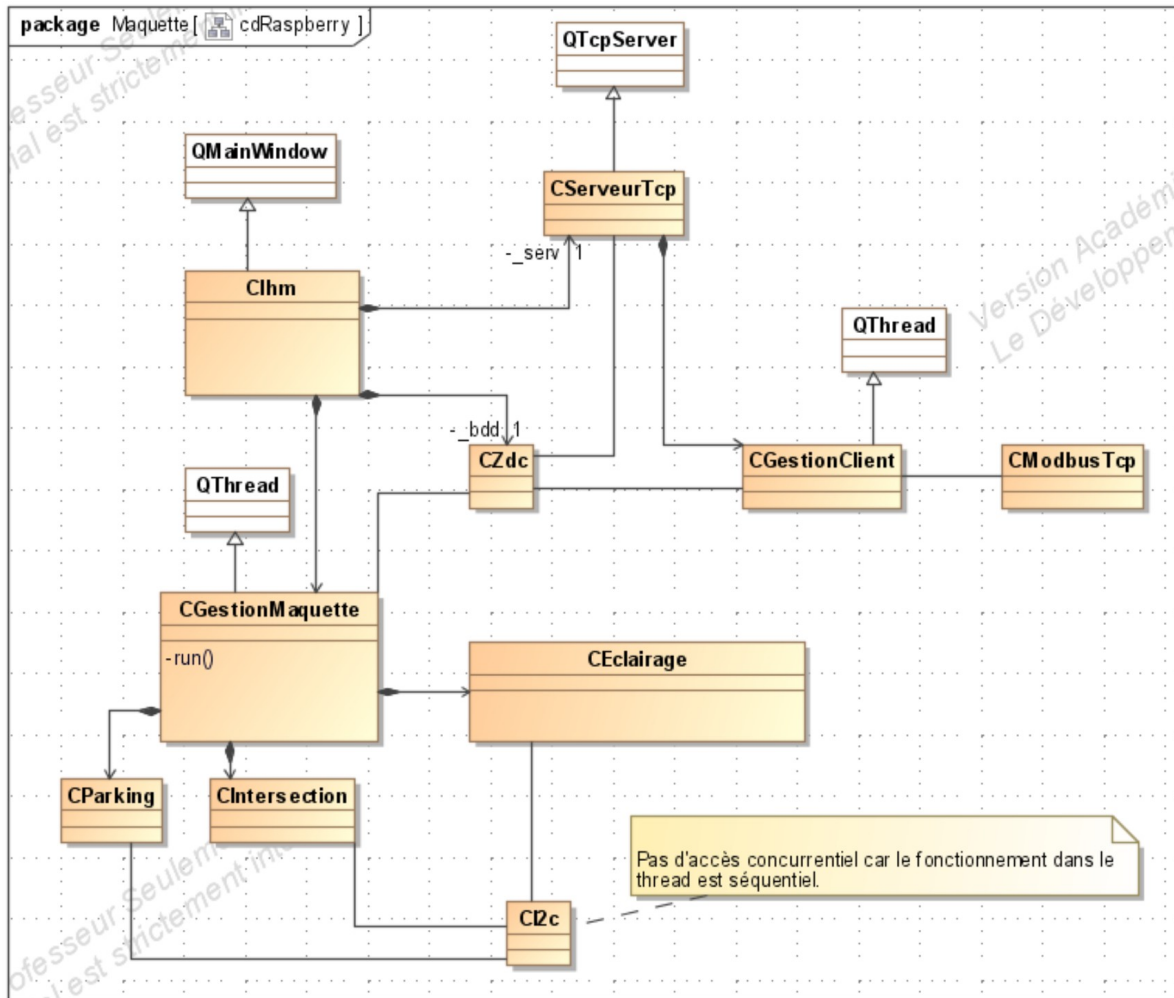


Figure 9 : Diagramme de classes du logiciel Raspberry

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)

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

2 équipes de 5 étudiants vont contribuer à la construction du système :

3.1 Equipe 1

<p>Étudiant 1 IR 11</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <ul style="list-style-type: none"> • Développement partiel du logiciel RASPBERRY. Classes à coder : <ul style="list-style-type: none"> ○ CServeurTcp ○ CGestionClient ○ CModbusTcp ○ CIhm 	<p>Installation : Système d'exploitation Raspbian, EDI. Mise en œuvre : ModBus Thread, TCP, communication réseau serveur. Configuration : EDI, Réseau Réalisation : Logiciel Qt C++ avec Qt Creator Documentation : Manuel d'installation et mise en œuvre.</p>
<p>Étudiant 2 IR 12</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <ul style="list-style-type: none"> • Développement partiel du logiciel RASPBERRY. Classes à coder : <ul style="list-style-type: none"> ○ CGestionMaquette ○ CParking ○ CIntersection ○ CEclairage ○ CI2c ○ CZdc 	<p>Installation : Système d'exploitation Raspbian, EDI. Mise en œuvre : I2c, Thread, mémoire partagée. Gestion de l'afficheur du parking. Configuration : EDI, I2c Réalisation : Logiciel Qt C++ avec Qt Creator Documentation : Cahier de recettes, défaillances.</p>
<p>Étudiant 3 EC 11</p>	<p><i>Liste des tâches assurées par l'étudiant</i> Accès parking version Arduino Nano</p> <ul style="list-style-type: none"> • Concevoir une carte de gestion du parking (<i>identification des usagers, commande des barrières, gestion des appels d'urgence</i>) gérée par une carte Arduino Nano, à partir d'un schéma proposé. • Travail en collaboration avec l'étudiant EC12 dont l'objectif est identique mais avec un microcontrôleur différent. • Effectuer tous les tests nécessaires pour valider la structure, en la modifiant si nécessaire. • Participer à l'élaboration d'un protocole de communication sur le bus I2C avec l'étudiant IR concerné. • Participer à la conception des parties mécaniques du parking. • Effectuer un travail de documentation afin de déterminer une structure permettant de piloter une vraie barrière, et si possible l'intégrer sur la carte. • Effectuer la saisie du schéma et le routage de la solution retenue. 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 la carte finalisée. 	<p>Installation : IDE Arduino. Mise en œuvre : Tester/valider/modifier une structure utilisant une carte Arduino nano pour gérer le parking afin de piloter les servo-moteurs des barrières, et gérer les capteurs de badges RFID pour identifier les usagers. La carte fonctionnera en tant que circuit esclave sur le bus I2C. L'analyse devra être menée conjointement avec l'étudiant EC12 dont la carte est gérée par un autre circuit. Prévoir la disposition de l'afficheur LCD avec lui également. Suite aux essais, modifier si nécessaire le schéma structurel proposé. Participer à une réflexion commune avec tous les étudiants EC du projet SmartCity pour le câblage du faisceau distribuant l'alimentation et le bus I2C sur tout le démonstrateur. Réalisation : Après validation de la solution, concevoir un circuit imprimé devant être fabriqué industriellement. Mettre en œuvre le faisceau de communication/alimentation. Documentation :</p> <ul style="list-style-type: none"> • Avantages / inconvénients de cette solution par rapport à celle d'une carte avec microcontrôleur Attiny3217. • 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

		<p>du circuit imprimé.</p> <ul style="list-style-type: none"> • Schéma structurel avec contours IBD. • Liste complète des composants avec leurs sources d'approvisionnement et leurs prix. • Programme de gestion de la carte, accompagné des commentaires et diagrammes nécessaires à sa compréhension. • Fiche de mise en service. • Fiche de dépannage.
<p>Étudiant 4</p> <p>EC 12</p>	<p>Liste des tâches assurées par l'étudiant</p> <p>Accès parking version ATtiny3217</p> <ul style="list-style-type: none"> • Concevoir une carte de gestion du parking (<i>identification des usagers, commande des barrières, gestion des appels d'urgence</i>) gérée par un microcontrôleur ATtiny3217 de nouvelle génération (<i>mega Core</i>), à partir d'un schéma proposé. • Travail en collaboration avec l'étudiant EC11 dont l'objectif est identique, mais avec un microcontrôleur différent. • Effectuer la mise au point en comparant les IDE MPLAB X et Arduino, ainsi que les cartes de développement correspondantes. • Effectuer tous les tests nécessaires pour valider la structure, en la modifiant si nécessaire. • Participer à l'élaboration d'un protocole de communication sur le bus I2C avec l'étudiant IR concerné. • Participer à la conception des parties mécaniques du parking. • Effectuer un travail de documentation afin de déterminer une structure permettant de piloter une vraie barrière, et si possible l'intégrer sur la carte. • Effectuer la saisie du schéma et le routage de la solution retenue. 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 la carte finalisée. 	<p>Installation : IDE Microchip le mieux adapté au circuit, sous Windows associé à une carte de développement pour microcontrôleur ATtiny de type Curiosity Nano. IDE Arduino avec librairies megaTinyCore.</p> <p>Mise en œuvre : Tester/valider/modifier une structure utilisant un microcontrôleur ATtiny3217 pour gérer le parking afin de piloter les servo-moteurs des barrières, et gérer les capteurs de badges RFID pour identifier les usagers. Le microcontrôleur fonctionnera en tant que circuit esclave sur le bus I2C.</p> <p>L'analyse devra être menée conjointement avec l'étudiant EC11 dont la carte est gérée par un autre circuit. Prévoir la disposition de l'afficheur LCD avec lui également.</p> <p>Suite aux essais, modifier si nécessaire le schéma structurel proposé.</p> <p>Participer à une réflexion commune avec tous les étudiants EC du projet SmartCity pour le câblage du faisceau distribuant l'alimentation et le bus I2C sur tout le démonstrateur.</p> <p>Réalisation :</p> <p>Après validation de la solution, concevoir un circuit imprimé devant être fabriqué industriellement.</p> <p>Mettre en œuvre le faisceau de communication/alimentation.</p> <p>Documentation :</p> <ul style="list-style-type: none"> • Avantages / inconvénients de cette solution par rapport à celle d'une carte avec Arduino Nano. • Avantages / inconvénients des environnements de travail et des cartes de développement pour ATtiny3217. • 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 de gestion de la carte, accompagné des commentaires et diagrammes nécessaires à sa compréhension. • Fiche de mise en service. • Fiche de dépannage.
Tous les étudiants	<ul style="list-style-type: none"> ✓ Tâches à traiter par l'ensemble des étudiants de l'équipe projet pour le développement de la solution ✓ Documents de vie du projet : 	<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</p>

	<ul style="list-style-type: none"> ✓ Fiches de lecture croisée ✓ Comptes rendus de réunion. ✓ <i>Domaines de physique à traiter par l'ensemble des étudiants de l'équipe projet :</i> <ul style="list-style-type: none"> ○ Traitement du signal analogique ○ Signaux : Puissance et énergie ○ Numérisation de signaux ○ Convertisseurs électromécaniques. ○ Antennes. OEM ○ Ondes électromagnétiques ○ Lignes de transmission ○ Transmissions numériques. 	<p>suivant une procédure écrite.</p> <p>Un dossier de recettes sera fourni.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------

3.2 Equipe 2

<p>Étudiant 1</p> <p>IR 21</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <ul style="list-style-type: none"> • Développement sur PC Windows d'une application cliente en C#. Cas d'utilisation : <ul style="list-style-type: none"> ○ Contrôler en manu ○ Communiquer avec maquette 	<p>Installation : Système d'exploitation Windows</p> <p>Mise en œuvre : Visual Studio, Réseau, modbus TCP.</p> <p>Configuration : Réseau, Caméra</p> <p>Réalisation : Application Windows Forms</p> <p>Documentation : Manuel d'installation et de maintenance.</p>
<p>Étudiant 2</p> <p>IR 22</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <ul style="list-style-type: none"> • Développement sur PC Linux d'une application cliente en C++/Qt. Cas d'utilisation : <ul style="list-style-type: none"> ○ Contrôler en manu ○ Communiquer avec maquette 	<p>Installation : Système d'exploitation Linux</p> <p>Mise en œuvre : Qt Creator, Réseau, modbus TCP.</p> <p>Configuration : Réseau, Caméra</p> <p>Réalisation : Application Qt graphique</p> <p>Documentation : Manuel d'installation et de maintenance.</p>
<p>Étudiant 3</p> <p>IR 23</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <ul style="list-style-type: none"> • Développement d'un site Internet hébergé dans la Raspberry et permettant de : <ul style="list-style-type: none"> ○ Représenter graphiquement les éléments de la ville et tout ce qui s'y passe. ○ (Option) Contrôler les éléments de la ville. • Cas d'utilisation : <ul style="list-style-type: none"> ○ Superviser WEB ○ Authentifier ○ (Option) Commander 	<p>Installation : Serveur WEB Apache</p> <p>Mise en œuvre : Https, http, PHP.</p> <p>Configuration : Apache, Linux</p> <p>Réalisation : Application WEB de supervision.</p> <p>Documentation : Manuel d'installation et de maintenance.</p>
<p>Étudiant 4</p> <p>EC 21</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <p>Éclairage public</p> <ul style="list-style-type: none"> • Concevoir une carte d'éclairage public permettant de gérer un tronçon de 6 lampadaires, avec détection de présence à chaque extrémité du tronçon et mesure de 	<p>Installation : IDE Arduino.</p> <p>Mise en œuvre : Tester/valider/modifier une structure utilisant un microcontrôleur ATtiny84 pour piloter une carte permettant de gérer 6 lampadaires, 2 détecteurs de présence et un capteur de luminosité. La carte fonctionnera en tant que circuit esclave sur le bus I2C.</p>

	<p>l'intensité lumineuse pour adapter l'éclairage.</p> <ul style="list-style-type: none"> • Un schéma structurel est proposé, effectuer tous les tests nécessaires pour valider les structures, et les modifier si nécessaire. • Participer à l'élaboration d'un protocole de communication sur le bus I2C avec l'étudiant IR concerné. • Participer à la conception des parties mécaniques du parking. • Effectuer la saisie du schéma et le routage de la solution retenue. 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 la carte finalisée. 	<p>Si, suite aux essais, le microcontrôleur ne convient pas, un autre devra être proposé et testé .</p> <p>Participer à une réflexion commune avec tous les étudiants EC du projet SmartCity pour le câblage du faisceau distribuant l'alimentation et le bus I2C sur tout le démonstrateur.</p> <p>Réalisation : Après validation de la solution, concevoir un circuit imprimé devant être fabriqué industriellement. Participer à la partie réalisation du démonstrateur. Mettre en œuvre le faisceau de communication/alimentation.</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 leurs prix. • Programme de gestion de la carte, accompagné des commentaires et diagrammes nécessaires à sa compréhension. • Fiche de mise en service. • Fiche de dépannage.
<p>Étudiant 5</p> <p>EC 22</p>	<p><i>Liste des tâches assurées par l'étudiant</i></p> <p>Feux d'intersection et signalisation des places libres/occupées du parking</p> <ul style="list-style-type: none"> • Concevoir une carte de gestion de feux d'intersection. Cette carte devra pouvoir fonctionner selon plusieurs modes prédéterminés qui correspondent à la norme en vigueur, la sélection du mode se fera par réception de consignes véhiculées par le bus I2C, sur lequel la carte fonctionnera en esclave. • Concevoir une carte de signalisation de l'état, libre ou occupé, de chacune des places du parking. • Un schéma structurel sera proposé pour chacune des cartes. • Effectuer tous les tests nécessaires pour valider les structures, et les modifier si nécessaire. • Participer à l'élaboration d'un protocole de communication sur le bus I2C avec l'étudiant IR concerné. • Participer à la conception des parties mécaniques du feu d'intersection et du parking. • Effectuer un travail de documentation afin de proposer une structure permettant de piloter de vrais feux et voyants, et si possible les intégrer sur chacune des cartes. • Effectuer la saisie du schéma et le routage des solutions retenue. Produire les fichiers Gerber afin que la fabrication du PCB soit sous-traitée. • Câbler les cartes et effectuer les essais. 	<p>Installation : IDE Arduino.</p> <p>Mise en œuvre : Tester/valider/modifier une structure utilisant un microcontrôleur Attiny pour gérer des feux tricolores. La carte fonctionnera en tant que circuit esclave sur le bus I2C.</p> <p>Tester/valider/modifier une carte permettant de signaler l'état de chacune des places du parking par une LED verte ou rouge à l'emplacement du véhicule. En principe cette carte ne nécessitera pas de circuit programmable.</p> <p>Participer à une réflexion commune avec tous les étudiants EC du projet SmartCity pour le câblage du faisceau distribuant l'alimentation et le bus I2C sur tout le démonstrateur. Mettre en œuvre ce faisceau.</p> <p>Réalisation :</p> <p>Après validation des solutions, concevoir un circuit imprimé pour chacune d'elles. Ces cartes devant être fabriquées industriellement.</p> <p>Mettre en œuvre le faisceau de communication/alimentation.</p> <p>Documentation :</p> <ul style="list-style-type: none"> • Schémas de câblage rapide (Fritzing) pour documenter les phases d'essais. • Documents de fabrication des cartes (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 leurs prix. • Programme de gestion de la carte, accompagné des commentaires et diagrammes nécessaires à sa compréhension. • Fiche de mise en service. • Fiche de dépannage.

	<ul style="list-style-type: none"> • Documenter la mise en service des cartes finalisées. 	
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 ○ Convertisseurs électromécaniques. ○ Antennes. OEM ○ Ondes électromagnétiques ○ Lignes de transmission ○ Transmissions numériques. 	<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> <p>Un dossier de recettes sera fourni.</p>

4 Exploitation Pédagogique – Compétences terminales évaluées :

Pour chaque équipe :

	Électronique et Communications	Informatique et Réseaux	IR 1	IR 2	IR 3	EC 1	EC 2
C2.1	Maintenir les informations		X	X	X	X	X
C2.2	Formaliser l'expression du besoin		X	X	X	X	X
C2.3	Organiser et/ou respecter la planification d'un projet		X	X	X	X	X
C2.4	Assumer le rôle total ou partiel de chef		X	X	X	X	X
C2.5	Travailler en équipe		X	X	X	X	X
C3.1	Analyser un cahier des charges		X	X	X	X	X
C3.3	Définir l'architecture globale d'un prototype ou d'un système		X	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	X
C3.6	Recenser les solutions existantes répondant au cahier des charges		X	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	X
C4.2	Adapter et/ou configurer un matériel		X	X	X	X	X
C4.3	Adapter et/ou configurer une structure logicielle	Installer et configurer une chaîne de développement	X	X	X	X	X
C4.4	Fabriquer un sous ensemble	Développer un module logiciel	X	X	X	X	X
C4.5	Tester et valider un module logiciel et matériel	Tester et valider un module logiciel	X	X	X	X	X
C4.6	Produire les documents de fabrication d'un sous ensemble	Intégrer un module logiciel	X	X	X	X	X
C4.7	Documenter une réalisation matérielle / logicielle		X	X	X	X	X

5 Planification

Début du projet (Dp)	semaine 1	: 06/01/2021.
Revue 1 (R1)	semaine 10	: à partir du 10/02/2021.
Revue 2 (R2)	semaine 15	: à partir du 21/04/2021
Remise du projet (Rp)	semaine 21	: 29/05/2021 (date limite de remise du dossier sur l'espace académique)
Soutenance finale (Sf)	semaine 24	: à partir du 15/06/2021.

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** pages et les documents annexes suivants :

(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 L'Isle sur la Sorgue, le 08/12/2020

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.