Client : **ELSYS-DESIGN**

Projet : **ROBOT HOLONOME V3**

Titre : **SPECIFICATIONS TECHNIQUES**

Référence : **22-9999**

Version : **A**

Nombre de pages :

Langage : **FR**

**OBJET :** L’objet de ce document est de décrire les spécifications matérielles et logicielles du robot de surveillance du sujet de stage ‘Robot holonome V3’.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Auteur** | **Auteur** | **Auteur** | **Validé** | **Validé** | **Approuvé** |
| Tristan CORNIERE | Lenny LAFFARGUE | Isabelle VAN LEEUWEN |  |  |  |
| **Signature** | **Signature** | **Signature** | **Signature** | **Signature** | **Signature** |

**HISTORIQUE DES MODIFICATIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Version*** | ***Date*** | ***Auteur(s)*** | ***Description des modifications*** |
| A |  | Tristan CORNIERE  Lenny LAFFARGUE  Isabelle VAN LEEUWEN | Création du document |

Table of contents

|  |  |
| --- | --- |
| **Acronyme** | **Définition** |
| **ELSYS** | **ELSYS DESIGN** |
| **ADVANS** | **ADVANS Group** |
| **AVISTO** | **AVISTO TELECOM** |
| **GPIO** | **General Purpose Input/Output** |
| **I2C** | **Inter Integrated Circuit** |
| **SPI** | **Serial Peripheral Interface** |
| **UART** | **Universal Asynchronous Receiver Transmitter** |
| **USB** | **Universal Serial Bus** |
| **HW** | **Hardware** |
| **SW** | **Software** |
| **FPGA** | **Field-Programmable Gate Array** |
| **AMCL** | **Adaptive Monte Carlo Localisation** |
| **SLAM** | **Simultaneous Localisation And Mapping** |
| **EKF** | **Extended Kalman Filter** |
| **IMU** | **Inertial Measurement Unit** |

[1. Introduction 6](#_Toc100822908)

[1.1. Énoncé du besoin 6](#_Toc100822909)

[1.2. Contexte, moyens et outils 6](#_Toc100822910)

[1.3. Existant 7](#_Toc100822911)

[1.4. Le projet 7](#_Toc100822912)

[2. Spécification des cas d’usage 9](#_Toc100822913)

[2.1. Fonctionnement classique 9](#_Toc100822914)

[2.2. Cas particuliers 9](#_Toc100822915)

[3. Spécifications détaillées du système 11](#_Toc100822916)

[3.1. Fonctionnement général 11](#_Toc100822917)

[3.2. Fonctionnement SOFTWARE 11](#_Toc100822918)

[3.2.1. SW bas niveau carte châssis 11](#_Toc100822919)

[3.2.2. SW haut niveau carte Zybo Z7-20 11](#_Toc100822920)

[3.2.2.1. Localisation 11](#_Toc100822921)

[3.2.2.2. Navigation 13](#_Toc100822922)

[3.2.2.3. Détecteur d’alertes 13](#_Toc100822923)

[3.2.2.4. Gestionnaire 13](#_Toc100822924)

[3.2.2.5. IHM web 13](#_Toc100822925)

[3.2.2.6. Gestionnaire d’alertes 14](#_Toc100822926)

[3.2.2.7. Evitement d’obstacles 14](#_Toc100822927)

[3.3. Fonctions électroniques 15](#_Toc100822928)

[3.3.1. Station de charge 15](#_Toc100822929)

[3.3.2. Balises d’alertes 15](#_Toc100822930)

[3.3.3. Carte d’interface 16](#_Toc100822931)

[3.3.4. Hautparleur / microphone 16](#_Toc100822932)

[3.4.1. Acquisition vidéo 16](#_Toc100822933)

[3.4.2. Interface 17](#_Toc100822934)

[3.5.1. Boîtier carte Zybo Z7-20 17](#_Toc100822935)

[3.5.2. Boîtier de balise d’alerte 17](#_Toc100822936)

[3.5.3. Station de charge 18](#_Toc100822937)

[4. Annexes 19](#_Toc100822938)

# Introduction

## Énoncé du besoin

Cause : L’entreprise Elsys Design veut surveiller ses bureaux dans l’agence de Toulouse sans qu’une personne ne soit présente.

Besoin : Il est nécessaire de pouvoir prévenir le directeur lorsqu’il y a une personne qui s’introduit illégalement dans les locaux de Elsys Design.

Objectif : Développer un robot de « sécurité » autonome à partir d’une « plateforme holonome » existante

## Contexte, moyens et outils

Le système est un robot autonome à roues holonomes ayant pour but d’accomplir des missions de surveillance et de sécurité.

Le système doit respecter les exigences suivantes :

* Développement du système complet avec un budget de 3000€,
* Déplacement autonome,
* Détection d’alerte.

Pour réaliser ce projet, ELSYS DESIGN a engagé trois stagiaires, responsables de toutes les phases du projet et répartis dans trois corps de métier :

* Un développeur Hardware, responsable du développement de la carte électronique qui viendra s’encastrer sur la carte FPGA. La carte développée servira d’interface avec la plateforme Holo32 et les autres éléments ajoutés.
* Un développeur Software, responsable du contrôle du robot (localisation, génération et suivi de trajectoire)
* Un développeur FPGA, responsable de la programmation de la carte SoC. Le développeur doit développer la partie FPGA (architecture, conception, code, test, vérification et routage).

## Existant

Elsys-Design a fourni plusieurs éléments :

* Une base roulante holonome conçue lors d’un précédent projet datant de 2019.

Cette base roulante est contrôlée par un microcontrôleur stm32 et possède un module Bluetooth Low Energy NRF52. A savoir qu’il y a également un asservissement en vitesse des moteurs implémenté.

* Du matériel électronique : Des fers à souder, une alimentation stabilisée et un oscilloscope sont à notre disposition pendant le stage.
* Une imprimante 3D





Figure : base roulante holonome

## Le projet

Le but du projet est de concevoir et développer un système de surveillance pour l’agence Elsys Design de Toulouse.

Ce système est composé de quatre éléments principaux :

* Le robot autonome : Ce robot doit pouvoir se déplacer dans l’agence de Toulouse en autonomie. Il doit savoir se repérer dans l’agence et éviter les murs et obstacles (chaises, personnes, portes...). Il doit aussi être capable de communiquer avec l’utilisateur et les balises de détection.
* La station de recharge : Entre chaque ronde que le robot devra effectuer, le robot devra retourner à son point de départ, qui sera la station de recharge, ce qui permettra notamment au robot de tenir toute la nuit.
* Les balises de détection : Pour détecter l’intrusion de quelqu’un dans l’agence, des balises seront placées sur chaque ouverture. Ces balises seront équipées d’un capteur de mouvement, d’un capteur de température et d’un module de communication pour appeler le robot en cas de problème. Ces balises doivent être autonomes en énergie pendant au moins un an afin d’éviter une maintenance trop fréquente. Elles se déclenchent dès qu’une porte ou fenêtre est ouverte ou fracturée.

# Spécification des cas d’usage

Liste des fonctions principales du robot :

* Se déplacer

**Cahier des charges initial**

* Détection d’alertes
* Navigation en Autonomie
* Contrôle à distance via une IHM
* Se situer dans un espace donné
* Eviter les obstacles
* Accès à distance en directe au flux vidéo
* Autonome en énergie

## Fonctionnement classique

Une fois allumé le robot vérifie qu’il est bien sur sa station de charge et attend un ordre : Contrôle manuel, détection d’anomalie par une balise ou routine de surveillance.

Cas contrôle manuel : Le robot exécute les commandes reçues.

Cas détection d’anomalie par une balise :

* + Envoi d’un avertissement (signal visuel sur l’IHM et SMS)
  + Le robot localise la balise concernée
  + Il génère une trajectoire entre la balise et sa position (donc la station de charge)
  + Il suit cette trajectoire tout en évitant les obstacles sur son chemin
  + Une fois arrivé dans la pièce concernée
    - Au bout de 1 minute, le robot retourne à sa station de charge

Cas Routine de surveillance :

* + Le robot suit un trajet prédéfini (une ronde) une fois par heure

## Cas particuliers

Si le robot est mis marche ailleurs qu’à la station de charge :

* + Il essaye de se localiser. Si réussite de la localisation :
    - Il génère une trajectoire entre sa localisation et la station de charge,
    - Il suit cette trajectoire en évitant les obstacles
    - Une fois arrivé, reprise du fonctionnement normal
  + Si Echec de la localisation :
    - Attente du mode manuel par l’utilisateur

Si ordre de contrôle manuel pendant un déplacement (Gestion d’anomalie ou ronde) :

* + Le contrôle manuel est prioritaire. L’ordre est exécuté sans délais et le robot passe en mode manuel.

Si problème de connexion internet : Retourne ou reste à la station de charge et attend.

Si détection d’anomalie pendant une ronde :

* + Interruption de la ronde
  + Gestion de l’anomalie comme en fonctionnement normal

Si détection d’une autre balise pendant la gestion d’une première détection :

* + Interrompt la première et va à la deuxième détection (priorité à la plus récente)

Si détection d’anomalie pendant le contrôle manuel :

* + Informe via l’IHM d’une détection

Si le robot n’arrive pas à se rendre dans une pièce (porte fermée, obstacle non contournable) :

* + Message d’erreur via l’IHM, reste sur place et attend un ordre manuel

# Spécifications détaillées du système

Dans le cadre de la [conception de produit](https://fr.wikipedia.org/wiki/Conception_de_produit), le produit minimum viable (ou MVP, de l'[anglais](https://fr.wikipedia.org/wiki/Anglais) : *minimum viable product*), il désigne aussi la stratégie utilisée pour fabriquer, tester et mettre sur le marché ce produit.

L'intérêt ici est donner un degré d’importance aux exigences afin d’en tirer un produit minimum viable.

|  |  |
| --- | --- |
| MVP 1 | 1 |
| MVP 2 | 2 |
| MVP 3 | 3 |

## Fonctionnement général

Le système en fonctionnement nominal doit assurer les fonctions suivantes :

|  |  |
| --- | --- |
| SYS-000 | Déplacement manuel |
| SYS-001 | Déplacement autonome |
| SYS-002 | Détection d’alertes |

## Fonctionnement SOFTWARE

### SW bas niveau carte châssis

|  |  |  |
| --- | --- | --- |
| SW-001 | Le SW de la carte châssis doit pouvoir échanger des informations avec une ou plusieurs cartes de contrôle haut niveau | 1 |
| SW-002 | Le SW de la carte châssis doit pouvoir assurer un asservissement fiable des moteurs | 1 |
| SW-003 | Le SW de la carte châssis doit pouvoir calculer et envoyer les données odométriques du déplacement du robot | 1 |
| SW-004 | Le SW de la carte châssis doit pouvoir exécuter un arrêt d’urgence s’il en reçoit l’ordre | 1 |

### SW haut niveau carte Zybo Z7-20

### Localisation

|  |  |  |
| --- | --- | --- |
| SW-101 | Le SW de localisation doit pouvoir échanger des informations avec le FPGA (RFID, UART, …) | 1 |
| SW-102 | Le SW de localisation doit pouvoir recevoir et traiter les données odométriques du robot (localisation relative) | 1 |
| SW-103 | Le SW de localisation doit pouvoir recevoir et traiter les données d’une IMU (loc relative) | 1 |
| SW-104 | Le SW de localisation doit pouvoir fusionner les données sensorielles (via EKF) | 1 |
| SW-105 | Le SW de localisation doit être en mesure de localiser le robot dans son environnement | 1 |
| SW-106 | Le SW de localisation doit pouvoir détecter lorsqu’il est en échec et le communiquer via l’IHM | 3 |

### Navigation

|  |  |  |
| --- | --- | --- |
| SW-201 | Le SW de navigation doit pouvoir échanger des informations avec le FPGA (RFID, UART) | 1 |
| SW-202 | Le SW de navigation doit être en mesure de recevoir des destinations depuis le gestionnaire de mission ou le gestionnaire d’alertes | 1 |
| SW-203 | Le SW de navigation doit être en mesure de recevoir la localisation du robot | 1 |
| SW-204 | Le SW de navigation doit pouvoir générer des commandes de déplacement |  |
| SW-205 | Le SW de navigation doit pouvoir générer une trajectoire exécutable par le robot (rotation + translation) | 1 |
| SW-206 | Le SW de navigation doit pouvoir interrompre la navigation en cours | 1 |
| SW-207 | Le SW de navigation doit pouvoir rectifier la navigation en cours | 3 |
| SW-208 | Le SW de navigation doit être capable de laisser la main au software d’évitement d’obstacles lorsqu’un obstacle est détecté | 3 |
| SW-209 | Le SW de navigation doit pouvoir corriger l’erreur de localisation odométrique à chaque passage intermédiaire grâce au RFID | 1 |

### Détecteur d’alertes

|  |  |  |
| --- | --- | --- |
| SW-301 | Le détecteur d’alertes doit pouvoir échanger des informations avec le gestionnaire | 2 |
| SW-302 | Le détecteur d’alertes doit pouvoir recevoir une alerte envoyée par une balise | 2 |
| SW-303 | Le détecteur d’alertes doit pouvoir identifier la balise à l’origine de l’alerte | 2 |

### Gestionnaire

|  |  |  |
| --- | --- | --- |
| SW-401 | Le gestionnaire doit pouvoir échanger des informations avec l’IHM | 1 |
| SW-402 | Le gestionnaire doit pouvoir échanger des informations avec le détecteur d’alertes | 2 |
| SW-403 | Le gestionnaire doit pouvoir organiser la procédure de démarrage du robot | 2 |
| SW-404 | Le gestionnaire doit être en mesure d’envoyer des destinations au Gestionnaire d’alertes | 1 |
| SW-405 | Le gestionnaire doit pouvoir exécuter des rondes : envoyer une série de destination au SW de navigation | 1 |
| SW-406 | Le gestionnaire doit pouvoir recevoir une nouvelle mission (rondes) depuis l’IHM et la garder en mémoire. | 3 |
| SW-407 | Le gestionnaire doit pouvoir lancer une mission à un horaire précis | 1 |
| SW-408 | Le gestionnaire doit pouvoir interrompre toute mission en cours si le mode manuel est demandé | 2 |
| SW-409 | Le gestionnaire doit pouvoir interrompre une ronde en cas d’alertes d’une balise | 2 |
| SW-410 | Le gestionnaire doit pouvoir enregistrer le flux audio du micro lorsqu’il exécute sa ronde. | 3 |

### IHM web

|  |  |  |
| --- | --- | --- |
| SW-501 | L’IHM web doit pouvoir échanger des informations avec le gestionnaire | 1 |
| SW-502 | L’IHM web doit pouvoir autoriser à l’utilisateur de choisir entre mode manuel ou Auto | 2 |
| SW-503 | L’IHM web doit prévenir l’utilisateur en cas de nouvelle alerte détectée | 2 |
| SW-504 | L’IHM web doit pouvoir communiquer l’état du robot : Manuel, Traitement alerte, ronde, repos | 2 |
| SW-505 | Si le mode manuel est activé, l’IHM web doit afficher les commandes clavier du robot | 2 |
| SW-506 | Si le mode manuel est activé, l’IHM web doit lire le clavier et transmettre les ordres au gestionnaire | 2 |
| SW-507 | Si le mode Auto est activé, l’IHM doit pouvoir offrir à l’utilisateur le choix de lancer une ronde manuellement | 2 |
| SW-508 | L’IHM web doit pouvoir émettre le flux audio du micro | 3 |

## Gestionnaire d’alertes

|  |  |  |
| --- | --- | --- |
| SW-601 | Le gestionnaire d’alertes doit pouvoir échanger des informations avec le gestionnaire | 1 |
| SW-602 | Le gestionnaire d’alertes doit lancer le software de navigation en transmettant ce trajet | 1 |
| SW-603 | Le gestionnaire d’alertes doit pouvoir interrompre l’intervention en cours pour intervenir sur une détection plus récente | 2 |
| SW-604 | Le gestionnaire d’alerte doit pouvoir générer le trajet pour retourner à la station e charge après une intervention et lancer la navigation sur ce même trajet. | 2 |
| SW-605 | Le gestionnaire d’alerte doit enregistrer et diffuser le flux audio lors de l’intervention | 3 |

## Evitement d’obstacles

|  |  |  |
| --- | --- | --- |
| SW-701 | Le SW d’évitement d’obstacles doit pouvoir interrompre la navigation lorsqu’un obstacle est détecté sur le chemin | 1 |
| SW-702 | Le SW d’évitement d’obstacles doit pouvoir calculer si un contournement est possible | 2 |
| SW-703 | Le SW d’évitement d’obstacles doit pouvoir rentrer à la station s’il n’est pas possible de passer | 2 |
| SW-704 | Le SW d’évitement d’obstacles doit pouvoir communiquer avec le SW de LOCALISATION | 1 |
| SW-705 | Une fois l’obstacle passé et le robot orienté, le SW d’évitement d’obstacles redonne la main au SW de Navigation pour terminer le trajet. | 3 |

## Fonctions électroniques

### Station de charge

|  |  |  |
| --- | --- | --- |
| HW-001 | La station de charge est en mesure de recharger les batteries en intégralité.  Batterie que nous utiliserons : batterie lithium LiFePO4 12.8V - 3.8Ah (protection BMS interne)  Voir [annexe 4.3](#_heading=h.6mdiwkjef252) | 1 |
| HW-002 | La station de charge doit indiquer lorsque la batterie est en charge ou chargée. | 2 |
| HW-003 | La station de charge doit avoir un IP20 (indice de protection) au minimum et respecter la norme CE. | 1 |
| HW-004 | La charge se fait par contact ou induction. | 2 |
| HW-005 | La station de charge doit protéger la batterie en cas de disfonctionnement. | 1 |
| HW-006 | La station de charge doit arrêter automatiquement la charge lorsque celle-ci est terminée. | 1 |
| HW-007 | La station de charge doit pouvoir être branchée via le réseau EDF. | 1 |

### Balises d’alertes

|  |  |  |
| --- | --- | --- |
| HW-101 | Les Balises communiquent entre elles via un réseau Wi-Fi Mesh, l’une des balises sert de routeur aux autres et est reliée directement avec le robot via une liaison I²C | 1 |
| HW-102 | Le réseau de balise devra couvrir l’ensemble des locaux d’Elsys Design, la distance entre les balises doit pouvoir être d’au moins 20m. Voir [annexe 4.4](#_heading=h.qgg9yta66cg4) | 1 |
| HW-103 | Une balise est positionnée sur la porte de chaque entrée des locaux de l’entreprise. | 3 |
| HW-104 | Il y a une balise dans chaque pièce avec son capteur spécifique. Exemple : La balise située proche de la baie serveur possède un capteur de température. | 1 |
| HW-105 | L’utilisateur ne peut pas changer le ou les capteurs présents sur la balise | 1 |
| HW-107 | La balise est en mesure de donner son identité au robot | 1 |
| HW-108 | La balise sera alimentée par une batterie ou pile avec une autonomie d’au moins 1 mois. | 1 |
| HW-109 | Un témoin lumineux rouge clignotant s’allume sur la balise en cas de batterie faible. | 2 |
| HW-104 | Il y a une balise dans chaque pièce avec son capteur spécifique. Exemple : La balise située proche de la baie serveur possède un capteur de température. | 1 |

### Carte d’interface

|  |  |  |
| --- | --- | --- |
| HW-201 | La carte d’interface permet de convertir les niveaux hauts et bas de chaque périphérique vers la carte Zybo Z7-20 et inversement sans dégrader la qualité du signal transmis. | 1 |
| HW-202 | La carte d’interface laisse accessible des éléments essentiels présent sur la carte Zybo Z7-20 (tel que le bouton d’allumage) | 1 |
| HW-203 | Il ne doit pas y avoir de doute sur la manière de brancher la carte | 1 |
| HW-204 | La carte doit posséder des emplacements dédiés afin d’être facilement testée et validée. Il doit être possible de la tester partie par partie. | 1 |
| HW-207 | La carte permet si nécessaire de convertir le protocole des périphériques en un protocole exploitable par Zybo Z7-20. | 1 |
| HW-206 | La carte doit prévoir des points de connexion en cas d’ajout de périphériques | 1 |

### Hautparleur / microphone

|  |  |  |
| --- | --- | --- |
| HW-401 | Le module Haut parleur / microphone est une carte périphérique qui viendra soit se monter en mezzanine ou directement via un connecteur qui lui est propre. | 3 |
| HW-401 | Le module Haut parleur / microphone est une carte périphérique qui viendra soit se monter en mezzanine ou directement via un connecteur qui lui est propre. | 3 |
| HW-402 | Le module haut parleur / microphone est activé en cas d’alerte | 3 |
| HW-403 | Le microphone enregistre l’ambiance sonore au moment de l’alerte (retour son + image disponible) | 3 |
| HW-404 | En cas d’alerte le haut parleur permet de diffuser un message préenregistré stocké dans la mémoire de l’Zybo Z7-20 | 3 |

* 1. Fonctionnement FPGA

### Interface

|  |  |  |
| --- | --- | --- |
| FPGA-101 | La partie FPGA doit communiquer avec la partie PS. | 1 |
| FPGA-102 | La partie FPGA doit pouvoir interagir avec la carte d’interface via GPIO. | 1 |
| FPGA-103 | La partie FPGA doit gérer un protocole UART pour communiquer avec la base Holonome. | 1 |
| FPGA-104 | La partie FPGA doit gérer un protocole I2C pour communiquer avec les différents périphériques reliés à travers la carte interface (Balise) | 1 |
| FPGA-105 | La partie FPGA doit pouvoir interagir avec le microphone | 3 |
| FPGA-106 | La partie FPGA doit pouvoir interagir avec les hauts parleurs | 3 |

### Gestion des ultrasons

|  |  |  |
| --- | --- | --- |
| FPGA-201 | La partie FPGA doit communiquer avec les ultrasons. | 1 |
| FPGA-202 | La partie FPGA doit sélectionner l’ultrason utilisé. | 1 |
| FPGA-203 | La partie FPGA doit générer un trigger d’au minimum 10 µs. | 1 |
| FPGA-204 | La partie FPGA doit pouvoir calculer la distance entre les obstacles et les ultrasons | 1 |
| FPGA-205 | La partie FPGA doit transformer les distances en zones et déterminer si l’objet s’éloigne ou non et le transmettre au PS. | 2 |
| FPGA-206 | La partie FPGA doit manipuler les ultrasons de manière à éviter les interférences. | 2 |

### Gestion du NFC

|  |  |  |
| --- | --- | --- |
| FPGA-301 | La partie FPGA doit communiquer avec le module NFC par IIC. | 1 |
| FPGA-302 | La partie FPGA doit pouvoir initialiser le module NFC. | 1 |
| FPGA-303 | La partie FPGA doit pouvoir activer et désactiver le champ NFC. | 1 |
| FPGA-304 | La partie FPGA doit pouvoir lire les modules NFC et en déduire une coordonnée. | 1 |
| FPGA-305 | La partie FPGA doit transmettre les données au PS. | 2 |

### Gestion de l’IMU

|  |  |  |
| --- | --- | --- |
| FPGA-401 | La partie FPGA doit communiquer avec l’IMU par IIC. | 1 |
| FPGA-402 | La partie FPGA doit pouvoir initialiser l’IMU. | 1 |
| FPGA-403 | La partie FPGA doit pouvoir lire les données de l’IMU de manière régulière (3 registres par adresse). | 1 |
| FPGA-404 | La partie FPGA doit pouvoir transmettre les données au PS. | 1 |

### Gestion de l’algorithme de navigation

|  |  |  |
| --- | --- | --- |
| FPGA-501 | La partie FPGA doit pouvoir calculer la trajectoire la plus courte entre deux points. | 1 |
| FPGA-502 | La partie FPGA doit envoyer cette trajectoire au PS. | 1 |

### Gestion des timers

|  |  |  |
| --- | --- | --- |
| FPGA-601 | La partie FPGA doit pouvoir signaler au PS quand 2 heures se sont écoulés. | 2 |
| FPGA-602 | La partie FPGA doit gérer tous les timers nécessaires au bon fonctionnement des ultrasons | 1 |

* 1. Spécifications mécaniques

Tout d’abord un boîtier sera nécessaire pour accueillir la carte Zybo Z7-20 ainsi que la carte d’interface.

|  |  |  |
| --- | --- | --- |
| MEC-001 | Le boîtier de la carte Zybo Z7-20 devra se fixer sur le châssis déjà existant du robot holonome |  |
| MEC-002 | Le boîtier devra être en capacité d'accueillir la carte électronique, son capteur et sa batterie |  |
| MEC-003 | Le boîtier devra être en capacité d'accueillir la carte électronique et le transformateur |  |
| MEC-004 | Le boîtier doit permettre de protéger l’électronique de l’extérieur et inversement (IP20) |  |

# Annexes