

Mini
Alarme de bureau connectée
Sous-système : Connexion au serveur
Node.js via USB

Dossier de Tests de Validation

Référence : Mini Alarme de bureau connectée/Connexion au serveur Node.js via USB-V5.0

Date : 20/12/2024

HISTORIQUE DES RÉVISIONS DU DTV

Version	Date	Commentaires
5.0	20/12/2024	Version initiale

SOMMAIRE

5.1 Test Connexion USB Arduino-Node.js

1.INTRODUCTION

1.1Objet

Le présent document constitue le dossier de test du sous-système : Contrôlé l’alarme via une interface web

1.2Documents de référence

Référence	Titre
cdch-v1	Cahier des charges fonctionnelles du client
DUP-V1	Diagramme des cas d'utilisation du projet
DEP-V1	Diagramme des exigences du projet

2. DESCRIPTION DE L'ENVIRONNEMENT DE TESTS

2.1 Configuration matérielle et logicielle

2.1.1 Généralités

- Matériel :
 - Arduino (ex. Arduino Uno)
 - Capteur magnétique pour la détection d'ouverture/fermeture
 - Buzzer pour l'alarme sonore
 - LED pour indiquer l'état de l'alarme
 - Câbles et connecteurs pour le montage
 - Notre ordinateur pour la communication série et l'hébergement du serveur Node.js
- Logiciel :
 - IDE Arduino (pour écrire et téléverser le programme Arduino)
 - Node.js (pour le serveur et la gestion des communications série)
 - Navigateur web (ex. Google Chrome) pour accéder à l'interface web

2.1.2 Configuration de la Mini alarme de bureau connectée de l'application

- Arduino :
 - Code écrit en langage C/C++ pour la communication.
 - Gestion des entrées/sorties numériques pour les capteurs, LED et buzzer.
- Serveur Node.js :
 - Mise en place d'un serveur HTTP/HTTPS pour héberger l'interface web.
 - Utilisation de la bibliothèque `serialport` pour la communication entre le serveur et l'Arduino.
 - Gestion des requêtes HTTP/HTTPS pour synchroniser l'état de l'alarme et du tiroir.
- Interface Web :
 - Créée en HTML, CSS, et JavaScript.
 - Affichage en temps réel de l'état du tiroir (ouvert/fermé).
 - Boutons pour armer et désarmer l'alarme.

5.0 FICHES DE TESTS

Le chapitre présent contient les fiches de tests suivantes :

5.1 Test Connexion USB Arduino-Node.js

Les tests suivants permettent de tester les fonctionnalités générales de Connexion au serveur Node.js via USB.

Réf. : FE1.1 : Maintenir la connexion / CAS_01		
ENVIRONNEMENT DU TEST		
<p>Arduino connecté à un ordinateur via un câble USB.</p> <p>Serveur Node.js démarré et prêt à gérer la communication série.</p> <p>Outil de débogage (ex. Moniteur série) disponible pour vérifier la transmission des données.</p>		
DESCRIPTION	CRITÈRES D'ACCEPTATION	RÉSULTAT
1 Connecter l'Arduino à l'ordinateur via USB	Le périphérique est détecté sur un port série	Réussie / Ratée
2 Démarrer le serveur Node.js	Le serveur affiche "Connexion série établie"	Réussie / Ratée

<u>3</u> Vérifier l'échange initial de données	L'Arduino envoie un message "Prêt" au serveur	Réussie / Ratée
<u>4</u>		
<u>5</u>		
ETAT DU TEST		
Réserve	Etat du test : Accepté Refusé	Accepté sous
Observations : (À compléter après le test)		

Réf. : FE1.2 : Maintenir la connexion / CAS_02		
ENVIRONNEMENT DU TEST		
Même environnement que CAS_01.		
DESCRIPTION	CRITÈRES D'ACCEPTATION	RÉSULTAT
<u>1</u> Maintenir la connexion série pendant environ 10 min	Aucun message d'erreur sur le serveur Node.js	Réussie / Ratée
<u>2</u> Simuler des échanges de données réguliers	Les données sont reçues sans perte ni décalage	Réussie / Ratée
<u>3</u> Vérifier l'état sur l'Arduino et le serveur	La connexion est toujours active à la fin du test	Réussie / Ratée
<u>4</u>		

<u>5</u>		
ETAT DU TEST		
Réserve	Etat du test : Accepté Refusé	Accepté sous
Observations : (À compléter après le test)		