

Surveillance Température-Humidité pour réfrigérateur

N° projet Diplôme 2409

Mandataire

Entreprise/Client:	ETML-ES – Ph. Bovey	Département:	SLO
Demandé par (Prénom, Nom):	Ph. Bovey	Date:	14.08.2025

1 Objectif - Cahier des charges

Le but de ce travail de diplôme est de réaliser un système de surveillance pour réfrigérateur, mesurant la température et l'humidité à l'intérieur de celui-ci (surveillance de médicaments ou d'autres denrées périssables).

Le diplômant devra étudier le système d'alimentation : pile (ex : bouton style CR2032) ou superCap, pour permettre à ce que le système de surveillance puisse fonctionner sur une durée d'un à deux mois minimums.

Une communication Wifi, à l'aide d'un **ESP32**¹ ou module équivalent, devra être implémentée pour transmettre une alarme, si un seuil ou un écart de température et/ou d'humidité est détecté, ou si la pile/superCap arrive en fin de vie.

L'utilisateur devra pouvoir se connecter l'ESP32 (celui-ci sera vu comme un web serveur) et transmettre des informations de configuration (écart min-max, seuil limite – éviter la congélation).

Un affiche type ePaper sera implémenté sur le système permettant d'afficher des valeurs de température et d'humidité (actuel, min, max, ...). L'affichage sera mis à jour s'il y a des variations de +/- 1° Celsius ou +/- 5% d'humidité.

Le système développé devra être fixé sur la porte du réfrigérateur, pour cela prévoir des magnets (petits aimants) en guise de pieds de fixation.

Pour simuler le réfrigérateur, une glacière portative fera office de Frigo.

Partie Hardware

Le diplômant devra concevoir une carte électronique basé sur un microcontrôleur de son choix ; cette carte doit contenir au minimum :

- Un microcontrôleur
- Alimentation
- Capteurs : humidité – température
ATTENTION : les capteurs seront déportés
- Affichage type ePaper

¹ ESP32 : <https://www.espressif.com/en/products/modules>

- Module RF ESP32 pour transmission de données
- ...

Il devra soit choisir un boîtier du commerce et l'adapter (emplacement affichage – câbles capteurs, accus, etc...) ou le designer lui-même.

Partie Firmware

L'étudiant réalisera un Firmware (partie microcontrôleur) permettant de faire au minimum les tâches suivantes :

- Lecture Capteurs
 - Température
 - Humidité
 - ...
- Communication
 - uC et module Wifi (ESP ou équivalent)
 - envoi et réception de datas
- Communication UART (extérieur)
 - Configuration module ESP
- Affichage
 - Datas (température – humidité – seuil - ...)
- Algorithme pour alarme
- ...

Le diplômé devra aussi configurer le module ESP32 :

- Wifi – mode web serveur

Partie Software

Le diplômé devra garantir la réception et la transmission de donnée sur :

- Module ESP (Wifi – mode web serveur - ...)
- Page Web simple (html)
- Configuration d'un routeur Wifi

1.1 Données en lien avec l'objectif – les grandes lignes

- Recherche et implémentation de solutions **Hardware** :
 - Microcontrôleur (préférence fabricant Microchip ou fabricant ST Electronics)
 - Alimentation – Etude de cas / recherche de solution :
 - Par pile Bouton
 - Par superCap
 - Pour les deux solutions :
 - Tension et ampérage adéquat
 - Durée de vie
 - ...
 - Attention** : les tensions subsidiaires au montage (interne) devront être réalisées par des alimentations à découpage DC-DC
 - Capteur Température
 - Plage de mesure – Résolution
 - Communication avec le uC
 - Connexion : Le capteur doit pouvoir être déporté
 - ...
 - Capteur d'humidité
 - Plage de mesure – Résolution
 - Communication avec le uC
 - Connexion : Le capteur doit pouvoir être déporté
 - ...
 - Module de communication ESP32 ou équivalent
 - Chip ou module
 - Paramétrage du module (boot)
 - Communication – Configuration – UART
 - Avec le uC
 - Extérieur – configuration
 - ...
 - Affichage
 - Type Epaper – Monochrome
 - Communication avec le uC
 - ...
- Recherche de composants électroniques autour des composants décrits ci-dessus : AOP, résistances, condensateur, self, autres
- Réalisation de schématique(s) complet(s) et d'un PCB - à réaliser sous ALTIUM de préférence
- Recherche (achat) ou réalisation d'un boîtier sous SolidWorks
- Recherche et implémentation de solutions au niveau **Firmware**
 - Lecture de capteurs
 - Température
 - Protocole de communication (SPI, I2C, UART, ...)
 - Conversion en °C
 - ...
 - Humidité
 - Protocole de communication (SPI, I2C, UART, ...)
 - Conversion en °C
 - ...
 - ...
 - Communication entre le uC et l'ESP32 ou équivalent

- Choix protocole de communication (SPI, I2C, UART, ...)
 - Configuration trames
 - Longueur trame – début et fin trame – taux de transfert
 - Datas (alarme – seuils température – humidité - ...)
 - ...
 - Communication entre le uC et l’affichage (ePaper)
 - Choix protocole de communication (SPI, I2C, UART, ...)
 - Affichage Datas : seuils - température – humidité - ...
 - ...
 - Algorithme Alarme
 - Comparaison seuil/écart et valeurs capteurs
 - Température
 - Humidité
 - ...
 - Batterie
 - ...
 - Configuration module ESP32 ou équivalent :
 - Wifi
 - SSID
 - Password
 - Protocole communication
 - ...
 - Serveur Web
 - Datas (alarme – seuils température – humidité - ...)
 - ...
 - ...
- Recherche et implémentation de solutions au niveau **Software**
 - Configuration serveur WEB sur ESP32
 - Wifi
 - Configuration page html simple
 - Réception et émission de datas (alarme – seuils température – humidité - ...)
 - ...
- Démontrer par différentes simulations et mesures que les parties Hardware, Firmware et Software ont été bien implémentées.

2 A l'issue du projet de diplôme, l'étudiant fournira (liste non exhaustive) :

- Fichiers sources de CAO électronique du PCB réalisé (ALTIUM) + configuration logiciel (version utilisées)
- Tout le nécessaire pour fabriquer un exemplaire hardware :
fichiers de fabrication (GERBER) / liste de pièces avec références pour commande (BOM) /
implantation (prototype) / modifications, etc
- Fichiers sources de programmation microcontrôleur (.c / .h) + configuration logiciel (version utilisées)
- Fichiers sources d'application externe au système électronique + configuration logiciel (version utilisées)
- Tout le nécessaire pour programmer le microcontrôleur (logiciel ou fichiers .hex), sous format numérique => utilisation de la structure de projet fourni par l'ES
- Tout le nécessaire pour programmer des applications (logiciel ou fichiers .hex) sous format numérique => utilisation de la structure de projet fourni par l'ES
- Tout le nécessaire à l'installation de programmes sur PC ou autres environnements utilisés durant le travail de diplômes
- Un rapport de diplôme contenant :
 - Les concepts du design **Hardware** :
 - Études des différents systèmes à implémenter (explications)
 - Choix des composants et dimensionnement de ceux-ci (calculs / simulation / autre)
 - Réalisation schématique / PCB / boîtier / montage de la carte
 - Les concepts du design **Firmware**
 - Structogramme / flowchart / Pseudocode
 - Explication des algorithmes mise en place
 - Démonstration par calculs / outils de debug / des résultats obtenus
 - Validation des concepts mis en place
 - Les concepts du design **Software**
 - Structogramme / flowchart / Pseudocode
 - Explication des algorithmes mise en place
 - Démonstration par calculs / outils de debug / des résultats obtenus
 - Validation des concepts mis en place
 - Tests & Validation :
 - Méthodologie de tests
 - Mesure(s)
 - Validation des résultats
 - Correction(s) apportée(s) au design (Hardware / Firmware / Software)
 - Estimation des coûts pour le design développé (un prototype)
 - Etat d'avancement & problème(s) rencontré(s)
 - Conclusion
 - Bibliographie / webographie / autre sources
- Les annexes :
 - Calculs détaillés des concepts
 - Listings complets des parties Firmware & Hardware que vous avez implémenté
 - Schématique + plan d'implémentation complète du PCB
 - Dessin / schématique du boîtier
 - Mesures

- Pages utilisées des différents datasheets ou documentations exploités
 - Mode d'emploi du système développé pendant le diplôme
 - Journal de travail
 - PV de séances hebdomadaires
 - Un prototype

3 Autres demandes / contraintes / conseils

- **Planifier** dans le détail les travaux demandés.
- Se référer au planning régulièrement, **vérifier son avancement**, rédiger son **journal de projet** quotidiennement.
- Commencer à **rédiger le rapport de diplôme le plus tôt possible**, et régulièrement tout au long du travail de diplôme.
- Prendre du temps, préparer sa réflexion, rechercher des apports théoriques et des exemples pratiques, **envisager plusieurs possibilités** avant de finaliser une solution.
- **Numéroter et dater tous les documents**
- En cas de **problème** (retard, objectif à revoir, difficulté rencontrée, etc.), se référer à l'enseignant et au mandataire au plus vite.
- Toutes les **décisions importantes**, tant au niveau technique qu'organisationnel, doivent être posées **par écrit** dans le PV de séance, le rapport de diplôme et /ou figurer dans le journal de projet, après discussion avec l'enseignant / le mandataire.

4 Documents de références

Projet ES

- **Prj 2312 – Badge pour place de travail**
K:\ES\PROJETS\SLO\ 2312_BadgePourPlaceTravail

Vidéo

- **Envoi de donnée via ESP32 (wifi)**
<https://www.youtube.com/watch?v=13Crtvr85IE>
- **ESP32 – Web serveur**
<https://www.youtube.com/watch?v=z-l-r3PX2IU>

Pages web :

- **Articles**

Accès Point (AP) et ESP32

<https://randomnerdtutorials.com/esp32-access-point-ap-web-server/>

Réseau Wifi et ESP32 :

https://www.wikidebrouillard.org/wiki/Configurez_le_r%C3%A9seau_Wifi_sur_un_ESP

Web Serveur – ESP32 :

<http://emery.claude.free.fr/esp32-serveur-web-simple.html>

Epaper – fournisseur :

https://www.mouser.ch/c/optoelectronics/displays/electronic-paper-displays-epaper/?srltid=AfmBOoqdtHntuGWUgNRkvcWUS7Bdp4_K-tzP2blPH2_8cU8vCf7sjIFV&_gl=1*tfnc0*_ga*MjA1MTI3MzQ3Ny4xNzIzNjQwOTEz*_ga_15W4STQT4T*MTcyMzY0MDkxMy4xLjAuMTcyMzY0MDkxNC41OS4wLjA.



Documentation sur l'ESP32

Wifi :

https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_wifi.html



Git

Espressif :

<https://github.com/espressif>