

Rapport de SA  - GEII

SA  CUVE CONNECT E



Encadrant : Monsieur Lucidarme PHILIPPE

Tonin OLAGNON-Louis BETOU-Layla QUEYSSAC

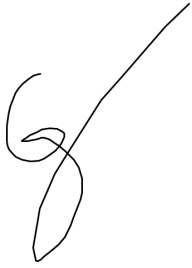
Ann e 2023-2024

Décharge de responsabilités

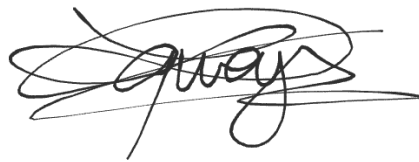
Ce rapport présente le travail réalisé par un groupe d'étudiants dans le cadre d'un projet pédagogique. Les auteurs et l'Université d'Angers ne garantissent pas que l'information, les documents, la méthodologie et le matériel présentés dans ce document soient complets, conformes à l'état de l'art et exacts ni n'assurent en toutes circonstances la sécurité des biens, des personnes et des utilisateurs. Les auteurs et l'Université d'Angers ne seront pas tenus responsables des dommages éventuels qui pourraient résulter de l'utilisation du contenu du présent rapport.

Licence

OLAGNON TONIN, QUEYSSAC LAYLA et BETOU LOUIS auteurs du présent rapport, publions et divulguons celui-ci sous la Licence Creative Commons suivante « CC BY » pour le monde entier et pendant la durée légale de protection des droits d'auteur. Cette licence autorise la représentation, la reproduction, la modification, la création d'œuvres dérivées et l'utilisation y compris à des fins commerciales sous réserve de mentionner les noms et prénoms des auteurs.



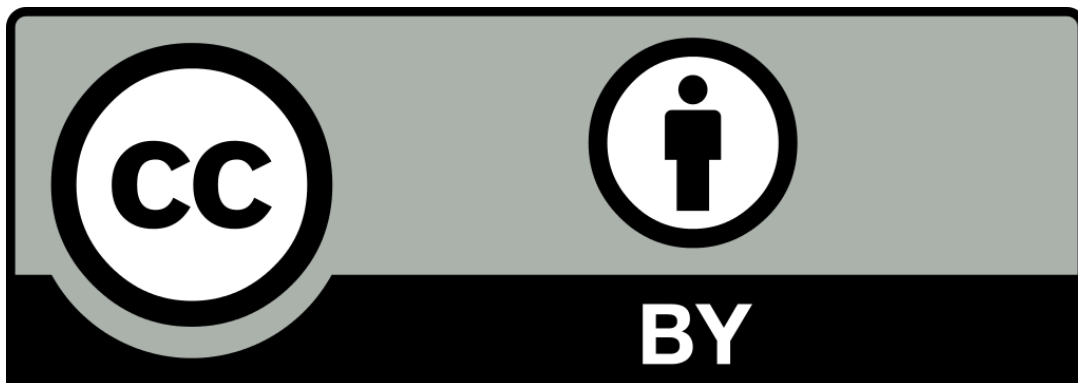
Tonin OLAGNON



Layla QUEYSSAC



Louis BETOU



Remerciements :

Tout d'abord, nous tenons à remercier tout particulièrement et à témoigner toute notre reconnaissance aux personnes suivantes, pour leur soutien dans la concrétisation de ce projet :

- M. Philippe LUCIDARME, responsable projet, pour ses conseils éclairés, sa patience, sa disponibilité et sa confiance qu'il nous a accordée lors de ce projet ; ainsi que pour sa mise à notre disposition de tous les moyens disponibles afin de mener à bien ce projet.
- Mme Géraldine DENECHÉAU, aidante à l'enseignement GEII, pour son aide et son temps lors de la réalisation du projet.
- Le département GEII de l'IUT d'Angers-Cholet pour sa coopération professionnelle tout au long de cette expérience et pour s'être mis à notre disposition lors de la réalisation de ce projet.

Introduction

Chaque été, les températures en France connaissent une hausse significative, entraînant des restrictions d'eau malgré des précipitations annuelles variables allant de 500 à 2000 millimètres selon la localisation géographique. Face à ces défis, l'utilisation de récupérateurs d'eau de pluie émerge comme une solution viable pour les particuliers, que ce soit pour l'arrosage des jardins pendant les périodes de restriction de consommation d'eau ou pour réduire la dépendance à l'eau potable.

Cependant, l'efficacité des cuves enterrées est limitée par la difficulté à surveiller leur niveau d'eau. C'est dans ce contexte que nous proposons une solution novatrice : un télémètre intelligent connecté à une interface web. Ce dispositif se compose de deux capteurs distincts, le premier étant un capteur à ultrasons installé à l'intérieur de la cuve, permettant de mesurer précisément le niveau d'eau, et le second étant un capteur de température. Ces capteurs sont intégrés à une unité de contrôle, le microcontrôleur, située dans une zone protégée de l'humidité.



Le microcontrôleur joue un rôle central dans le projet en collectant les données des capteurs, en gérant l'ensemble du système et en les transmettant à une base de données dédiée. Cette base de données est essentielle car elle stocke les informations recueillies et alimente une interface web consultable par l'utilisateur. De plus, pour étendre les fonctionnalités du projet, nous avons intégré un relais au système. Ce relais offre la possibilité de contrôler d'autres équipements en lien avec la gestion de l'eau, tels que des pompes d'arrosage ou des vannes d'irrigation. Ainsi, l'utilisateur peut automatiser davantage le processus et optimiser l'utilisation de l'eau collectée. En résumé, notre solution allie technologie, praticité et flexibilité pour répondre aux défis saisonniers en France tout en offrant des options avancées de contrôle et d'automatisation.

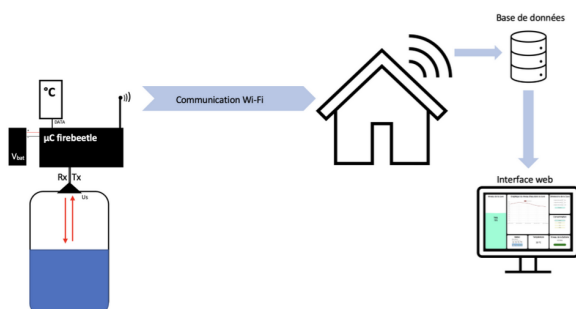


Table des matières

Introduction.....	5
I . Cahier des Charges.....	7
II . Choix technologiques.....	8
A. Carte Commande Capteur.....	8
1. Microcontrôleur.....	8
2. Capteur Ultrasons.....	10
3. Capteur de Température.....	11
B. Carte Commande Relais.....	12
1. Microcontrôleur.....	12
2. Convertisseur AC/DC.....	12
3. Relais 1RT.....	12
4. Transistor.....	13
III . Réalisation.....	14
A. Carte Commande Capteur.....	14
1. Schéma Structurel.....	14
2. Routage.....	15
3. Tests.....	16
B. Carte Commande Relais.....	21
1. Schéma Structurel.....	21
2. Routage.....	22
3. Tests.....	23
IV . Conclusion et Perspectives.....	26
V . Bibliographie.....	28
VI . Résumé.....	29

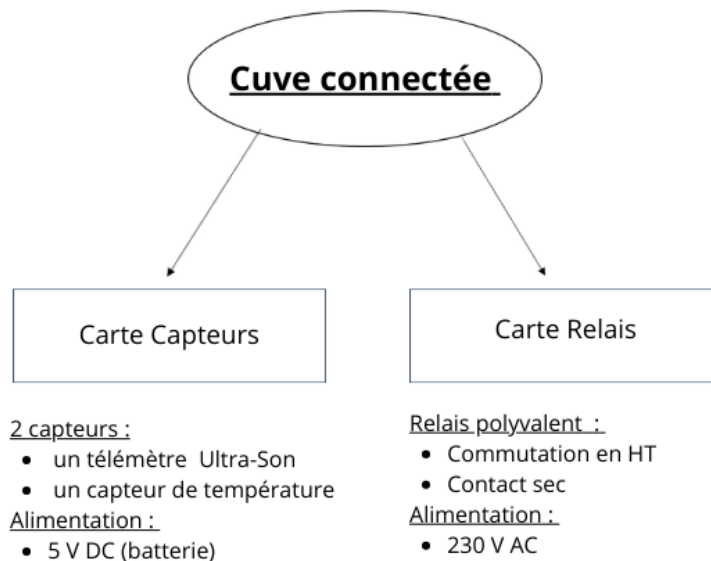
I . Cahier des Charges

Ce projet se divise en deux parties, l'une à été réalisée par un groupe d'étudiants pendant l'année 2022-2023. Elle consiste à faire fonctionner le télémètre ultrason qui s'occupe de relever la distance entre le haut de la cuve et la surface de l'eau et le capteur de température à proximité du microcontrôleur qui permet de vérifier si la batterie au lithium n'est pas en surchauffe. Ce groupe à aussi choisi certains composants technologiques du projet que nous allons vous présenter par la suite. Enfin, il ont réalisé une interface web qui récolte et affiche les données des capteurs par wifi pour qu'elles soient facilement consultables par l'utilisateur.

La deuxième partie du projet a été réalisée par notre équipe. Elle doit se charger de concevoir deux cartes électroniques, chacune jouant un rôle spécifique dans le bon fonctionnement du système. La première carte regroupe le télémètre à ultrasons, le capteur de température et le microcontrôleur préalablement sélectionnés. Ces composants ont des fonctions bien définies : le télémètre à ultrasons mesure la distance entre le haut de la cuve et la surface de l'eau, le capteur de température relève les données thermiques pour surveiller l'état de la batterie au lithium, tandis que le microcontrôleur est le cerveau du dispositif, chargé de collecter, traiter et transmettre ces informations.

La seconde carte, elle, comprend le microcontrôleur et le relais. Le microcontrôleur dans cette carte est responsable de la gestion du relais, tandis que le relais introduit une fonctionnalité supplémentaire, il offre la possibilité de contrôler d'autres équipements liés à la gestion de l'eau, tels que des pompes d'arrosage ou des vannes d'irrigation.

L'objectif majeur de cette phase est d'optimiser le routage des cartes, en minimisant le nombre de composants et la surface occupée. Après avoir finalisé les schémas électroniques, nous passerons à la commande des cartes, à la brasure des composants, puis à une vérification minutieuse pour assurer la fiabilité et l'efficacité globale du système.



II . Choix technologiques

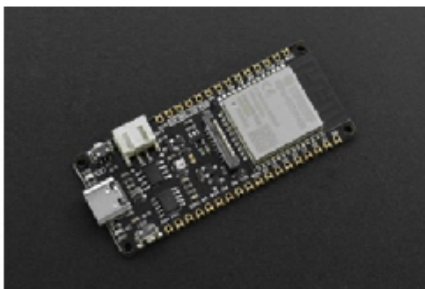
Pour réaliser ces deux cartes, nous avons dû choisir les composants les plus adaptés à notre cahier des charges. Pour cela, nous avons étudié plusieurs références et comparé les différences afin de trouver le meilleur composant pour répondre à notre cahier des charges.

A. Carte Commande Capteur

Tout d'abord, pour réaliser la carte capteur, nous devons utiliser un capteur ultrasons, un capteur de température, ainsi qu'un microcontrôleur.

1. Microcontrôleur

Le choix du microcontrôleur a été réalisé par le groupe de l'année dernière. Leur choix a été fait grâce à deux études. En effet, dans un premier temps ils ont comparé cinq microcontrôleurs à partir de leurs documentation constructeur. Le point commun entre ces cinq microcontrôleurs étant la possibilité de communiquer en wifi mais également d'avoir un mode veille appelé deepsleep. Parmi ces cinq microcontrôleurs on peut retrouver l'ESP32-firebeetle (DFR0654), l'ESP32-firebeetle (DFR0478) l'ESP32-Devkitc, l'ESP32-AZ-Delivery LOLIN32 ou encore l'ESP32 Thing Plus.



Esp32 firebeetle



Esp32 Devkitc



Esp32 AZ-Delivery LOLIN32



Esp32 thing plus

Tableau comparatif des différents microcontrôleurs à partir des documentations constructeurs :

	Esp32 Devkitc (V4)	Esp32 firebeetle (DFR0654)	Esp32 firebeetle (DFR0478)	Esp32 AZ-Delivery LOLIN32	Esp32 Thing Plus
Deepsleep	✓	✓	✓	✓	✓
Wifi	✓	✓	✓	✓	✓
Consommation en mode normal	entre 30mA et 68mA	80mA	80 mA	80mA	40mA
Consommation en Deepsleep	10µA	10µA	10 µA	10µA	10µA

La consommation de courant étant le plus important, c'est cette caractéristique qui a été décisive. Cependant, d'après ce tableau, on peut observer que la consommation de courant en mode Deepsleep et en mode normal est similaire pour les cinq microcontrôleurs. Alors, ils ont réalisé une seconde étude en mesurant eux-même cette consommation :

Microcontrôleur	Esp32 firebeetle (DFR0478)	Esp32 firebeetle (DFR0654) (low power pad connecté)	Esp32 firebeetle (DFR0654) (low power pad déconnecté)	Esp32 Devkitc (V4)	Esp32 AZ-Delivery LOLIN32	Esp32 Thing Plus
Consommation en Deepsleep	10.5 µA	520 µA	11.6 µA	3.84 mA	12 µA	4.1 mA
Consommation en mode normal	42 mA	40 mA	40 mA	43.6 mA	39 mA	41 mA

Suite à cette nouvelle étude, on peut remarquer que les informations de la première étude ne sont pas forcément justes. Ainsi, après cette seconde étude, on peut donc dire que le microcontrôleur le plus adapté est l'ESP32 Firebeetle DFR0654 avec le mode low power pad déconnecté.

2. Capteur Ultrasons

Le choix du capteur ultrason a également été réalisé par le groupe de l'année dernière. En effet, ils ont, comme pour le microcontrôleur, comparé plusieurs capteurs afin de trouver celui qui répond le mieux au cahier des charges. Une étude de documentations constructeur a donc été menée sur cinq capteurs différents, avec comme point commun l'étanchéité. En effet, ce capteur ayant pour but d'être dans une cuve à eau, il doit pouvoir fonctionner même au contact de l'eau. Les capteurs qui ont été étudiés sont donc, le SEN0208, le Seeed Studio, le JSN-SR04T, le A01NYUB et le SEN0311



SEN0208



Seeed Studio



JSN-SR04T



A01NYUB



SEN0311

Nous obtenons donc ce tableau final :

	SEN0208	Seed Studio	JSN-SR04T	A01NYUB	SEN0311
Distance de fonctionnement	0.25 à 4.5 m	0.03 à 3.5 m	0.23 à 6 m	0.28 à 4 m	0.03 à 4.5 m
Étanche	oui	non	oui	oui	oui
Alimentation	3.3 à 5 Vcc	3.3 à 5 Vcc	3.3 à 5 Vcc	3.5 à 5 Vcc	3.3 à 5 Vcc
Consommation	30 mA	8 mA	5 mA (statique) 30 mA (pdt la mesure)	15 mA	< 8 mA

Suite à cette étude, nous pouvons remarquer que le SEN0311 est le capteur qui consomme le moins, et qui a la distance de fonctionnement la plus adaptée à ce projet. En effet, il possède la plus petite distance minimale, permettant ainsi de mesurer la profondeur de la cuve même si celle-ci est seulement un tout petit peu remplie. Le capteur le plus adapté est donc le SEN0311.

3. Capteur de Température

Le choix du capteur de température SHT41 nous a été proposé par l'enseignant car, le capteur utilisé par le groupe de l'année dernière n'était pas assez précis. En effet, le capteur choisi l'année dernière est un DHT22. Grâce à la documentation constructeur, nous pouvons remarquer que la mesure de température se fait à une précision de $\pm 0.5^{\circ}\text{C}$ et le taux d'humidité à $\pm 2\% \text{RH}$. Alors que, lors de l'étude de la documentation constructeur du capteur SHT41, on a pu observer que la précision de la mesure de température était de $\pm 0.1^{\circ}\text{C}$ et que celle du taux d'humidité était de $\pm 1\% \text{RH}$. Ainsi, nous pouvons donc confirmer que le SHT41 est plus précis que le DHT22.



Capteur de température SHT41

B. Carte Commande Relais

1. Microcontrôleur

Pour le choix du microcontrôleur de la carte commande relais, nous avons décidé de reprendre la même étude que pour le choix sur la carte commande capteur. Ainsi nous avons utilisé l'ESP32 Firebeetle DFR0654 avec le mode low power pad déconnecté pour cette carte.

2. Convertisseur AC/DC

Le choix du convertisseur AC/DC nous a été imposé par l'enseignant. Il s'agit du convertisseur HLK-PM03, AC/DC 100-240 V AC, 3.3 V DC, 1A qui permet donc de convertir une tension alternative 230V en une tension continue 3.3V



Convertisseur HLK-PM03

3. Relais 1RT

Pour le choix du relais, le but était de trouver un relais pouvant avoir une tension de 230V pour le contact et une tension de 3.3V pour la bobine. Cependant, nous devons utiliser les relais déjà présents au magasin. Ainsi, nous avons hésité entre 2 relais : le Finder 40.61 6V ou le Finder 30.22 5V. En effet, le Finder 40.61 6V possède une tension nominale pour le contact de 250V mais, a une tension nominale pour la bobine de 6V.

Contact specification	
Rated voltage/Maximum switching voltage V AC	250/400

Coil specification		
Nominal voltage (U _N)	V AC (50/60 Hz)	6 - 12 - 24 - 48 - 60 - 110 - 120 - 230 - 240
	V DC	***See table

*** Nominal voltage (U_N):
5 - 6 - 7 - 9 - 12 - 14 - 18 - 21 -
24 - 28 - 36 - 48 - 60 - 90 -
110 - 125 V DC

Extrait documentation constructeur du Relais Finder 40.61 6V

Or, nous devons utiliser la broche 3.3V de l'ESP32. Nous avons donc cherché un autre relais avec une tension nominale de la bobine plus faible, mais, en essayant de garder une tension nominale pour le contact la plus élevée possible. Nous avons donc trouvé le relais Finder 30.22 5V.

En effet, ce relais possède une tension nominale pour le contact de seulement 125V mais, a une tension nominale pour la bobine de 5V.

Contact specification		
Rated voltage/ Maximum switching voltage	V AC	125/250
Coil specification		
Nominal voltage (U_N)	V AC (50/60 Hz)	—
	V DC	5 - 6 - 9 - 12 - 24 - 48

Extrait documentation constructeur du Relais Finder 30.22 5V



Relais Finder 40.61 6V



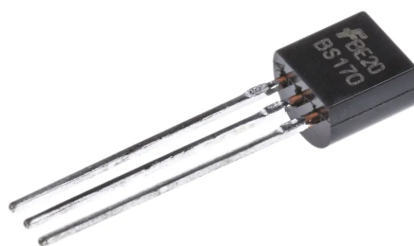
Relais Finder 30.22 5V

4. Transistor

Nous avons décidé d'utiliser un transistor sur cette carte pour commander le relais. Nous nous sommes plutôt tournés vers un transistor MOSFET car il y a moins de caractéristiques à prendre en compte pour choisir un modèle de mosfet en particulier. Nous devons regarder la tension max de drain que nous allons lui soumettre qui est de 5V et le courant maximum de drain que nous allons également lui soumettre qui était de 42 mA. Nous avons donc opté pour le BS170 car il possède une tension de drain maximum de 60 V et un courant drain maximum de 500 mA. Cela respecte donc notre cahier des charges.

Symbol	Parameter	BS170	MMBF170	Units
V_{DSS}	Drain-Source Voltage	60	60	V
V_{DGR}	Drain-Gate Voltage ($R_{GS} \leq 1M\Omega$)	60	60	V
V_{GSS}	Gate-Source Voltage	± 20		V
I_D	Drain Current - Continuous	500	500	mA
	- Pulsed	1200	800	

Extrait documentation constructeur du BS170



Transistor BS170

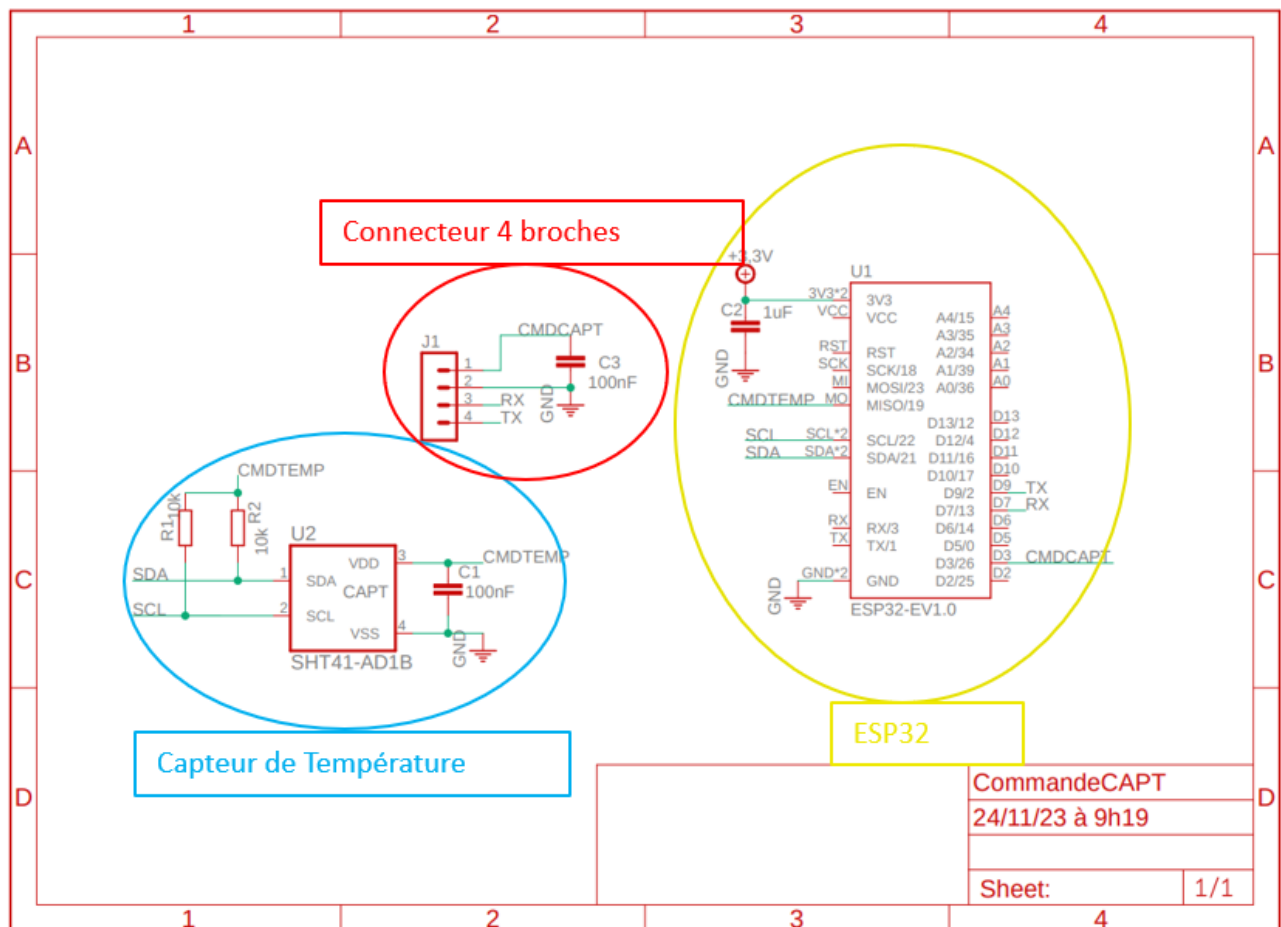
III. Réalisation

A. Carte Commande Capteur

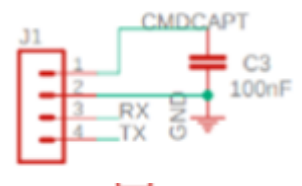
Comme expliquée dans le cahier des charges, cette carte va s'occuper de récupérer la distance entre le fond de la cuve et la surface de l'eau et relever la température de la carte. Les données seront traitées par un microcontrôleur et récupérées par le site web.

1. Schéma Structurel

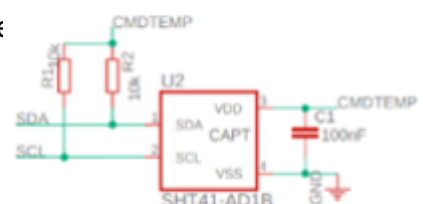
Voici le schéma structurel de la carte :



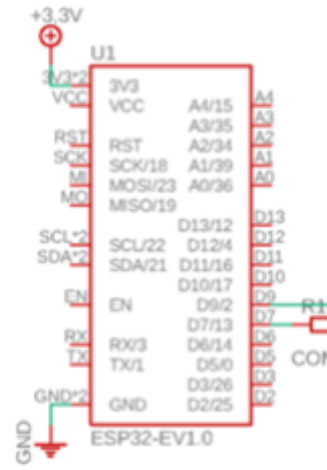
- **Connecteur 4 broches** : Connecteur permettant de relier le capteur US à l'



- **Capteur de Température** : Capteur de Température pour mesure température et le taux d'humidité qui est commandé par l'ESP32



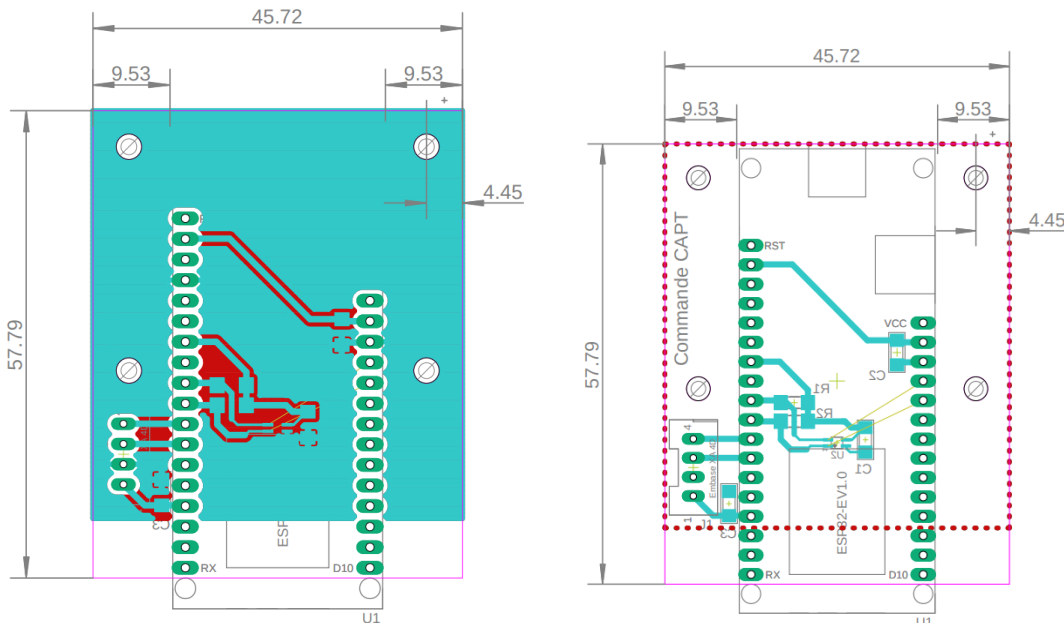
- **ESP 32** : Microcontrôleur permettant de contrôler le relais



2. Routage

Pour réaliser le routage de la carte commande capteur, notre but principal était de réaliser une carte à laquelle on pouvait connecter le télémètre ultrasons et le commander à l'aide du microcontrôleur. Nous nous sommes rendus compte cependant que la broche du microcontrôleur qui était censé alimenter le capteur de température ne fournissait pas assez de tension, nous avons donc réfléchi à deux solutions. La première était de placer un transistor alimenté en 5V par le microcontrôleur, dont l'activation serait commandée par la broche qui anciennement était censée alimenter le circuit. Nous avons utilisé cette méthode pour les tests du capteur. Cependant, cela complexifie le routage et ajoute des composants supplémentaires, c'est pourquoi nous pensons que la deuxième solution, qui consiste à simplement se connecter sur un PIN Digitale du microcontrôleur, est plus intéressante.

Enfin l'antenne WI-FI de l'ESP 32 impose qu'aucun composant, piste ou plan de masse ne soit présent à proximité de celle-ci.



Schémas de routage de la carte avec la PIN qui ne fournit pas assez de tension au capteur

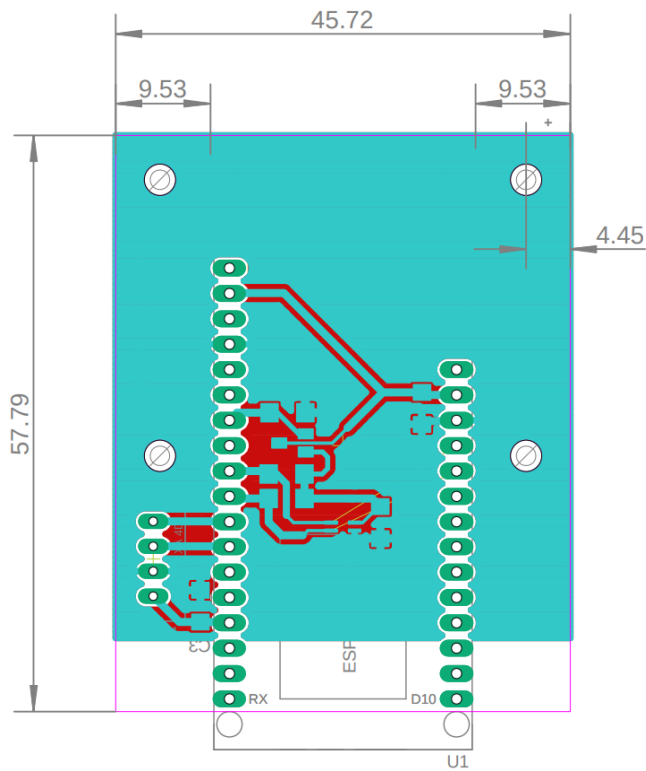


Schéma de routage de la carte avec le capteur de température commandé par un transistor.

3. Tests

Programme de Tests Carte Cuve

T1 Test de continuité à la réception du PCB

	OK / KO
GND	OK
3V3	OK
CMDTEMP	OK
CMDCAPT	OK

T2 Mesure de valeurs de composants

	Mesure	Valeur du composant	OK / KO
C1		F	OK
C2		F	OK
C3		F	OK
R1		1MΩ	OK
R2		1MΩ	OK

T3 Tests des Alimentations

A l'aide du programme Code_capteurUS.ino et des librairies associées :

```
1 // Inclusion des bibliothèques nécessaires
2 #include "Arduino.h"
3 #include "sen0311.h"
4
5 // Configuration du capteur ultrasonique (US)
6 sen0311 us(2, 2, 13);
7
8 void setup() {
9 // Initialisation de la communication série à une vitesse de 9600 bps
10 Serial.begin(9600);
11
12 // Attente de 1 seconde après le démarrage du programme
13 delay(1000);
14
15 // Affichage d'un message dans le moniteur série
16 Serial.println("US sensor");
17
18 // Configuration de la broche 26 en mode de sortie
19 pinMode(26, OUTPUT);
20
21 // Attente de 500 millisecondes
22 delay(500);
23 }
24
25 void loop() {
26 // Activation du capteur ultrasonique en envoyant un signal sur la broche 26
27 digitalWrite(26, HIGH);
28
29 // Variable pour stocker la distance mesurée par le capteur ultrasonique
30 int distance;
31
32 // Nettoyage des données du capteur ultrasonique
33 us.flushUs();
34
35 // Obtention de la distance mesurée par le capteur ultrasonique
36 distance = us.getDistance();
37
38 // Affichage de la distance dans le moniteur série
39 Serial.print("distance = ");
40 Serial.print(distance);
41 Serial.println("mm");
42 }
43
```

-3.1 Mesure des tensions d'alimentations des capteurs

	Mesure	Valeur de la Tension	OK / KO
CMDTEMP		0V	OK
CMDCAPT		3,3V	OK

A l'aide du programme Code_temperature.ino et des bibliothèques associées :

```
Code_temperature.ino
1  #include "Adafruit_SHT4x.h"
2
3  // Instanciation d'un objet Adafruit_SHT4x appelé sht4
4  Adafruit_SHT4x sht4 = Adafruit_SHT4x();
5
6
7  void setup()
8  {
9      // Configuration de la broche 23 en mode sortie
10     pinMode(23, OUTPUT);
11
12     // Initialisation de la communication série à une vitesse de 115200 bauds
13     Serial.begin(115200);
14
15     // Attente de l'ouverture de la console série (utile pour certaines cartes comme Zero, Leonardo, etc.)
16     while (!Serial)
17         delay(10);
18
19     // Affichage du message de test du capteur
20     Serial.println("Test du capteur Adafruit SHT4x");
21
22     // Vérification de la présence du capteur, arrêt du programme sinon
23     if (!sht4.begin())
24     {
25         Serial.println("Capteur SHT4x non trouvé");
26         while (1) delay(1);
27     }
28
29     // Affichage du message de détection du capteur
30     Serial.println("Capteur SHT4x détecté");
31
32     // Affichage du numéro de série du capteur en format hexadécimal
33     Serial.print("Numéro de série 0x");
34     Serial.println(sht4.readSerial(), HEX);
35
36     // Configuration de la précision du capteur (haute précision)
37     sht4.setPrecision(SHT4X_HIGH_PRECISION);
38
39     // Configuration du chauffage du capteur (pas de chauffage)
40     sht4.setHeater(SHT4X_NO_HEATER);
41 }
42
43 // Fonction principale du programme, exécutée en boucle
44 void loop()
45 {
46     // Activation de la sortie sur la broche 23 (HIGH)
47     digitalWrite(23, HIGH);
48
49     // Déclaration d'objets pour stocker les données de température et d'humidité
50     sensors_event_t humidity, temp;
51
52     // Mesure du temps d'exécution de la fonction getEvent
53     uint32_t timestamp = millis();
54     sht4.getEvent(&humidity, &temp); // Remplissage des objets temp et humidity avec les nouvelles données
55     timestamp = millis() - timestamp;
56
57     // Affichage des valeurs de température et d'humidité
58     Serial.print("Température : ");
59     Serial.print(temp.temperature);
60     Serial.println(" degrés C");
61     Serial.print("Humidité : ");
62     Serial.print(humidity.relative_humidity);
63     Serial.println("% HR");
64
65     // Affichage de la durée de lecture en millisecondes
66     Serial.print("Durée de lecture (ms) : ");
67     Serial.println(timestamp);
68
69     // Pause d'une seconde
70     delay(1000);
71 }
```

-3.2 Mesure des tensions d'alimentations des capteurs

	Mesure	Valeur de la Tension	OK / KO
CMDTEMP		3,3V	OK
CMDCAPT		0V	OK

T5 Test des mesures

A l'aide du programme Code_capteurUS.ino et des bibliothèques associées :

-5.1 Observer dans le terminal série les valeurs de distance en mm retournés par le capteur US

```
17:49:53.606 -> distance = 313mm
17:49:53.698 -> distance = 230mm
17:49:53.778 -> distance = 136mm
17:49:53.903 -> distance = 93mm
17:49:53.983 -> distance = 89mm
17:49:54.076 -> distance = 93mm
17:49:54.187 -> distance = 85mm
17:49:54.281 -> distance = 111mm
17:49:54.374 -> distance = 87mm
17:49:54.438 -> distance = 86mm
17:49:54.563 -> distance = 129mm
17:49:54.642 -> distance = 247mm
17:49:54.752 -> distance = 128mm
17:49:54.845 -> distance = 236mm
```

A l'aide du programme Code_temperature.ino et des bibliothèques associées :

-5.2 Observer dans le terminal série les valeurs de température en °C et d'hygrométrie en % retournés par le capteur SHT41

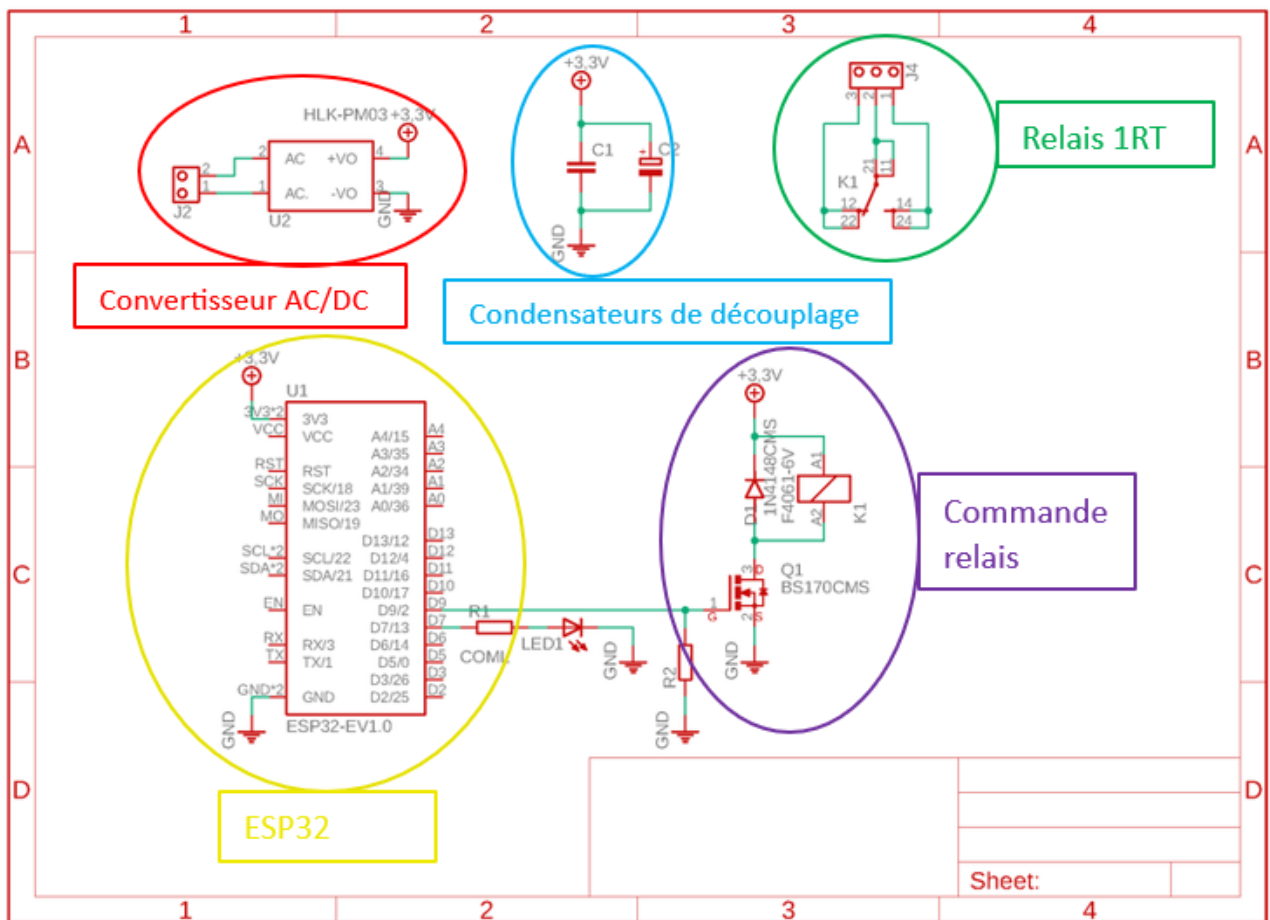
```
16:06:45.948 -> humidity: 44.8% rH
16:06:45.948 -> Read duration (ms): 10
16:06:46.946 -> Temperature: 22.36 degrees C
16:06:46.946 -> Humidity: 45.03% rH
16:06:46.946 -> Read duration (ms): 10
16:06:47.972 -> Temperature: 22.42 degrees C
16:06:47.972 -> Humidity: 45.26% rH
16:06:47.972 -> Read duration (ms): 10
```

B. Carte Commande Relais

Comme expliquée dans le cahier des charges, cette carte a pour but d'être polyvalente. En effet, cette carte peut permettre de contrôler d'autres équipements liés à la gestion de l'eau, tels que des pompes d'arrosage ou des vannes d'irrigation.

1. Schéma Structurel

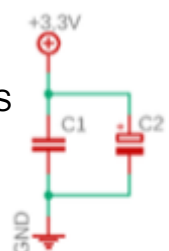
Voici le schéma structurel de la carte :



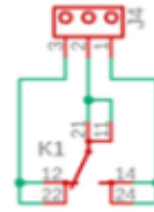
- **Convertisseur AC/DC** : Permet de convertir une tension sinusoïdale 230V en une tension continue de 3.3V



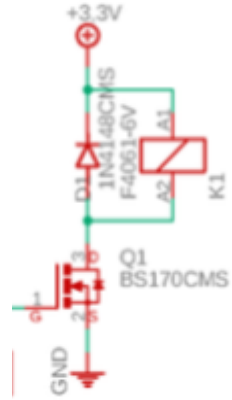
- **Condensateurs de découplage** : Permet de stabiliser la tension en entrée de l'ESP32



- Relais 1RT : Relais avec une position Repos et Travail



- Commande Relais : Commande du relais électroniquement par un transistor MOSF canal N à enrichissement



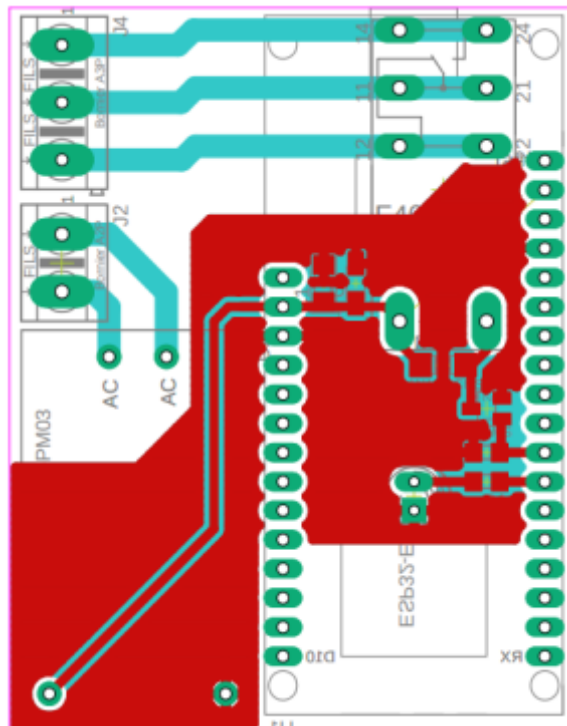
- ESP 32 : Microcontrôleur permettant de contrôler le relais



2. Routage

Pour réaliser le routage de la carte relais notre principale contrainte était la taille, dans un premier temps nous avons pensé réaliser un PCB de la taille de l'ESP 32. Pour cela nous aurions dû placer tous les autres composants du même côté, ce qui est impossible puisque les composants sont assez volumineux, de plus cela ne nous permettait pas d'éloigner de l'ESP 32 les pistes de l'alimentation de la carte en 230V qui aurait pu générer du bruit et des perturbations. Enfin l'antenne WI-FI de l'ESP 32 impose qu'aucun composant, piste ou plan de masse ne soit présent à proximité de celle-ci.

Sur ce routage on peut donc observer en haut et à gauche avec les pistes vertes la partie pour la haute tension, toute la partie rouge représente le plan de masse avec la partie électronique de commande et enfin en bas à droite on constate une partie dénuée de plan de masse correspondant à l'antenne WI-FI.



3. Tests

Programme de Tests Carte relais

T1 Test de continuité à la réception du PCB

	OK / KO
GND	OK
3V3	OK

T2 Mesure de valeurs de composants

	Mesure	Valeur du composant	OK / KO
C1		F	OK
C2		F	OK
R1		4,7K Ω	OK
R2		1M Ω	OK

T3 Tests des Alimentations

	Mesure	Valeur de la Tension	OK / KO
3V3		3,3V \approx	OK
230V	Pas mesuré	230V	

T4 Tests du transistor

A l'aide du programme Code_relais.ino :

```
Relais.ino
1 // Définition de la broche pour la LED
2 #define broche_LED LED_BUILTIN
3
4 void setup() {
5 // Configuration de la broche de la LED en mode de sortie
6 pinMode(broche_LED, OUTPUT);
7 }
8
9 void loop() {
10 // Allumer la LED
11 digitalWrite(broche_LED, HIGH);
12 // Attendre pendant 2 secondes
13 delay(2000);
14 // Éteindre la LED
15 digitalWrite(broche_LED, LOW);
16 // Attendre pendant 2 secondes
17 delay(2000);
18 }
```

- 4.1 Mesurer La tension de la broche D9

	Mesure	Valeur de la Tension	OK / KO
LED Eteinte		0V	OK
LED Allumée		3,3V	OK

- 4.2 Mesurer la tension aux bornes de la bobine

	Mesure	Valeur de la Tension	OK / KO
LED Eteinte		0V	OK
LED Allumée		3,3V	OK

T5 Test de la commutation du relais

A l'aide du programme Code_relais.ino :

-5.1 Test entre J1 et J2

	Mesure	Continuité	OK / KO
LED Eteinte		NON	OK
LED Allumée		OUI	OK

-5.2 Test entre J1 et J3

	Mesure	Continuité	OK / KO
LED Eteinte		NON	OK
LED Allumée		NON	OK

-5.3 Test entre J2 et J3

	Mesure	Continuité	OK / KO
LED Eteinte		OUI	OK
LED Allumée		NON	OK

IV. Conclusion et Perspectives

Nous avons réussi, à la fin du temps imparti, à avoir deux cartes fonctionnelles. Nous avons pu tester les cartes avec les différents composants pour valider le bon fonctionnement de celles-ci. Cependant, il y a certaines conditions du cahier des charges qui n'ont pas été remplies. En effet, les dimensions de la carte relais n'ont pas les mêmes dimensions que le microcontrôleur, il n'a pas été possible de réaliser un routage de cette carte plus condensé en raison de la taille des composants qui se trouvaient dessus.

Concernant la carte capteurs, nous avons commandé une première version de carte avec des connexion sur les broches RX TX pour commander le télémètre ultrasons, cependant nous nous sommes rendus compte, après la commande de celle-ci, que ses broches ne pouvaient pas transmettre les informations. Nous avons donc recommandé une carte qui permet de commander ce télémètre avec les broches D7 et D9 du microcontrôleur. De plus, la broches censée alimenter le capteur de température ne fournissait pas assez d'énergie, nous avons donc alimenté le capteur par la broche 5V du microcontrôleur et commandé son alimentation avec la broches MOSI qui était censée l'alimenter précédemment. Il a cependant été retenu de simplement changer la broche d'alimentation du microcontrôleur, nous n'avons pas pu réaliser ce routage mais avons fait un test sur maquette et la broche D10 apporte assez d'énergie pour alimenter le capteur de température.

Il aurait fallu, avant la commande des cartes, tester nos routages avec des maquettes afin d'éviter de commander plusieurs cartes.. Cependant, compte tenu du temps imparti et du délais de réception des cartes, nous ne pouvions pas nous permettre de procéder ainsi.

Nous aurions aussi pu améliorer notre routage en mettant par exemple une isolation galvanique sur la carte de commande du relais, pour éviter les perturbations entre les différents domaines de tension (230V/5V) mais aussi éloigner ses pistes. Il aurait aussi été judicieux de mettre des bords ronds sur les cartes pour éviter de se blesser avec, que ce soit lors de la fabrication ou de l'installation.

Une alimentation de la carte de commande des capteurs serait aussi envisageable par des panneaux photovoltaïques, ce qui permettrait de ne plus avoir besoin de surveiller la température de la batterie avec un capteur de température, mais aussi de ne pas avoir besoin de recharger la batterie après sa décharge, ce qui permettrait de diminuer la maintenance du système. Le modèle ci-dessous correspond à notre domaine d'utilisation.



Source :

https://www.amazon.fr/Hiluckey-Batterie-Solaire%EF%BC%9A26800mAh-T%C3%A9%9A9pho%9A9pho-Tablettes/dp/B09V4FJY6X/ref=sr_1_6

Note : Ce modèle de panneau photovoltaïque a été trouvé par le groupe de l'année dernière.

Nous avons également trouvé un relais qui possède les caractéristiques recherchées dans le cahier des charges, en effet celui-ci fonctionne dans les domaines de tension voulus. Il s'agit du relais "Finder 36.11-4001". En effet, ce relais possède une tension nominale pour le contact de 250 V et peut avoir une tension nominale pour la bobine de 3V ou de 5V.



Relais Finder 36.11-4001

Lien de la documentation technique du composant : <https://manualsbrain.com/fr/manuals/570236/>

Il n'est cependant pas disponible au magasin de l'IUT et il faudrait donc le commander, implémenter ses dimensions sur le logiciel eagle ainsi que le tester en remplaçant l'ancien relais par celui-ci.

Pour protéger la carte de commande des capteurs (de l'humidité et des chocs extérieurs) ainsi que celle du relais (pour protéger les personnes car cette carte fonctionne sous 230V ainsi que de protéger la carte des chocs extérieurs), il serait envisageable de réaliser un boîtier 3D, réalisable soit en interne avec un logiciel de modélisation 3D comme solidworks ou fusion 360 en prenant en compte les types de matériaux nécessaires ainsi que les dimensions, ou réalisable en externe.

V. Bibliographie

Voici ci-dessous les liens des sites et documents que nous avons consultés pour mener à bien notre projet.

- <https://wiki.dfrobot.com/A02YYUW-Waterproof-Ultrasonic-Sensor-SKU-SEN0311> , Informations sur le capteur US
- <https://www.dfrobot.com/product-1588.html> , informations, et schématique de l'ESP32
- <https://chat.openai.com/> , aide à la réalisation du présent rapport
- <https://manualsbrain.com/fr/manuals/570236/> , documentation technique du relais "Finder"
- https://www.amazon.fr/Hiluckey-Batterie-Solaire%EF%BC%9A26800mAh-T%C3%A9C3%A9phone-Tablettes/dp/B09V4FJY6X/ref=sr_1_6 , modèle du panneau photovoltaïque possible
- Rapport_Final_SAE_Cuve_Connectée , rapport du groupe qui a travaillé sur le projet l'année dernière

VI. Résumé

Dans le cadre de notre projet visant à améliorer l'utilisation des récupérateurs d'eau de pluie, ce rapport détaille les étapes cruciales pour sa réalisation. Nous mettons en avant les défis du projet et les solutions que nous avons élaborées pour surmonter les obstacles rencontrés. Notre approche s'appuie grandement sur l'analyse approfondie du rapport antérieur et des documents fournis par le groupe d'étudiants précédemment impliqué dans ce projet. Nous avons testé les cartes à l'aide de ligne de programme sur le logiciel "Arduino" et avons réalisé les schémas structurels et les routages des cartes sur le logiciel "Eagle".

L'accès à l'intégralité du rapport et des documents connexes est ouvert à tous, favorisant ainsi la possibilité de reprise et d'amélioration. Les perspectives d'évolution du projet sont également abordées dans le présent rapport.

As part of our project aimed at improving the use of rainwater harvesters, this report outlines the crucial steps for its implementation. We highlight the challenges faced in the project and the solutions we developed to overcome these obstacles. Our approach relies heavily on a thorough analysis of the previous report and the documents provided by the group of students previously involved in this project. We tested the cards using a program line in the "Arduino" software and created the structural diagrams and card routings in the "Eagle" software.

Access to the entire report and related documents is open to everyone, facilitating the opportunity for continuation and improvement. The potential developments of the project are also discussed in this report.

