

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلكايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



## MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

**En** : Électronique

**Spécialité** : Instrumentation

**Par** : BERRALEM Asmaa Halima

MOKHDAR Hanane

**Sujet**

**Étude et réalisation d'un nettoyeur flottant télécommandé pour la collecte des déchets flottants sur l'eau.**

Soutenu publiquement, le 23 / 06 / 2026, devant le jury composé de :

|                           |     |                       |              |
|---------------------------|-----|-----------------------|--------------|
| Mme RAHOU Fatima Zohra    | MCA | Université de Tlemcen | Présidente   |
| Mr KERAI Nabil            | MAA | Université de Tlemcen | Examineur    |
| Mr BOUANATI Sidi Mohammed | MCB | Université de Tlemcen | Encadrant    |
| Mr SLAMI Ahmed            | MCB | Université de Tlemcen | Co-Encadrant |

Année universitaire : 2025 / 2026

# REMERCIEMENTS

Louange à Dieu, par la grâce duquel s'accomplissent les bonnes œuvres, lui qui nous a accordé patience et persévérance et nous a guidés jusqu'à l'achèvement de ce travail.

Nous tenons à exprimer nos sincères remerciements et notre profonde gratitude à notre encadrant de mémoire, Monsieur BOUANATI Sidi Mohammed, pour ses orientations éclairées et ses précieux conseils, qui ont constitué un appui essentiel dans la réalisation de ce travail, ainsi que pour son accompagnement et son suivi constant tout au long de la préparation de ce mémoire

Nous adressons également notre reconnaissance à notre co-encadrant Monsieur SLAMI Ahmed, pour son soutien, sa collaboration et ses multiples orientations, dont l'impact a été déterminant dans l'aboutissement de cette recherche.

Nous adressons les plus vifs remerciements à Madame RAHOU Fatima Zohra, d'avoir accepté de d'évaluer notre travail de mémoire.

Nous remercions également Monsieur KERAI Nabil, d'avoir accepté d'examiner notre travail.

Nos remerciements s'étendent à l'ensemble de nos honorables professeurs, pour le savoir et les connaissances qu'ils nous ont transmis, ainsi que pour les efforts qu'ils ont déployés tout au long de notre parcours académique.

Enfin, nous souhaitons exprimer nos vifs remerciements à nos familles pour leur patience, leur soutien indéfectible, leur confiance et leurs encouragements permanents, qui ont représenté pour nous une source essentielle de motivation afin de mener à bien ce travail.



## *Dédicace*

*Je remercie Dieu de m'avoir guidé et aidé à achever ce travail.*

*Je suis honorée de dédier ce travail*

*A mes chers parents,  
en reconnaissance de leurs sacrifices et de leur soutien constant,  
ainsi que de leurs prières qui ont toujours été la raison de ma réussite.*

*A tous mes professeurs respectés,  
en signe de gratitude pour leurs efforts et leurs pertinents conseils tout au long de  
mon parcours scolaire*

*Et à mes amis, pour leur soutien et leurs encouragements constants.*

*Et à tous les personnes qui m'ont soutenue et ont contribué à mon arrivée à ce  
stade, et que j'ai rencontrées au cours de ma vie et qui ont laissé une belle  
empreinte dans mon parcours.*

*Berralem Asmaa Halima*

*Mokhdar Hanane*



# Table des matières

|                                                                           |            |
|---------------------------------------------------------------------------|------------|
| <b>REMERCIEMENTS.....</b>                                                 | <b>I</b>   |
| <i>Dédicace.....</i>                                                      | <b>II</b>  |
| <b>Table des matières .....</b>                                           | <b>III</b> |
| <b>Liste des figures .....</b>                                            | <b>VI</b>  |
| <b>Liste des tableaux .....</b>                                           | <b>IX</b>  |
| <b>Liste des abréviations.....</b>                                        | <b>X</b>   |
| <b>Introduction générale.....</b>                                         | <b>1</b>   |
| <b>Introduction générale .....</b>                                        | <b>2</b>   |
| <b>Chapitre I   Etat de L’art.....</b>                                    | <b>3</b>   |
| <b>I. Introduction .....</b>                                              | <b>4</b>   |
| <b>II. Différents types de pollution .....</b>                            | <b>4</b>   |
| <b>II. 1. Pollution atmosphérique .....</b>                               | <b>4</b>   |
| <b>II. 1. 1. Définition de la pollution atmosphérique.....</b>            | <b>4</b>   |
| <b>II. 1. 2. Origines et facteurs de la pollution atmosphérique .....</b> | <b>4</b>   |
| <b>II. 1. 3. Impacts de la pollution atmosphérique .....</b>              | <b>5</b>   |
| <b>II. 1. 4. Illustration de la pollution atmosphérique .....</b>         | <b>5</b>   |
| <b>II. 2. Pollution des sols .....</b>                                    | <b>6</b>   |
| <b>II. 2. 1. Définition de la Pollution des sols .....</b>                | <b>6</b>   |
| <b>II. 2. 2. Origines et facteurs de la pollution des sols .....</b>      | <b>6</b>   |
| <b>II. 2. 3. Impacts de la pollution des sols .....</b>                   | <b>6</b>   |
| <b>II. 2. 4. Illustration de la pollution des sols .....</b>              | <b>6</b>   |
| <b>II. 3. Pollution hydrique .....</b>                                    | <b>7</b>   |
| <b>II. 3. 1. Définition de la Pollution hydrique .....</b>                | <b>7</b>   |
| <b>II. 3. 2. Origines et facteurs de la pollution hydrique .....</b>      | <b>8</b>   |
| <b>II. 3. 3. Impacts de la pollution hydrique .....</b>                   | <b>8</b>   |
| <b>II. 3. 4. Illustration de la pollution hydrique .....</b>              | <b>8</b>   |
| <b>III. Présentation des solutions existantes.....</b>                    | <b>9</b>   |
| <b>III.1. Méthodes classiques .....</b>                                   | <b>9</b>   |
| <b>III.1.1. Nettoyage manuel .....</b>                                    | <b>9</b>   |
| <b>III.1.1.1. Avantages du nettoyage manuel.....</b>                      | <b>9</b>   |
| <b>III.1.1.2. Limites du nettoyage manuel .....</b>                       | <b>10</b>  |

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| <b>III.2. Méthodes Technologiques .....</b>                                 | <b>10</b> |
| <b>III.2.1. Les systèmes filaires.....</b>                                  | <b>10</b> |
| <b>III.2.1.1. Avantages des systèmes filaires .....</b>                     | <b>11</b> |
| <b>III.2.1.2. Limites des systèmes filaires.....</b>                        | <b>11</b> |
| <b>III.2.2. Les systèmes autonomes .....</b>                                | <b>11</b> |
| <b>III.2.2.1. Avantages des systèmes autonomes .....</b>                    | <b>12</b> |
| <b>III.2.2.2. Limites des systèmes autonomes .....</b>                      | <b>12</b> |
| <b>III.2.3. Les systèmes semi-automatisés .....</b>                         | <b>13</b> |
| <b>III.2.3.1. Avantages des systèmes semi-automatisés.....</b>              | <b>13</b> |
| <b>IV. Description générale du système .....</b>                            | <b>14</b> |
| <b>V. Conclusion .....</b>                                                  | <b>14</b> |
| <b>Chapitre II Description des composants utilisés dans le projet .....</b> | <b>15</b> |
| <b>I.Introduction .....</b>                                                 | <b>16</b> |
| <b>II. Module de contrôle basé sur l'ESP32 .....</b>                        | <b>16</b> |
| <b>II.2. L'architecture interne de l'ESP32 .....</b>                        | <b>17</b> |
| <b>II.2.1. Unité Centrale de Traitement CPU .....</b>                       | <b>17</b> |
| <b>II.2.2. Mémoire de l'ESP32.....</b>                                      | <b>17</b> |
| <b>II.2.3. Modules de communication sans fil .....</b>                      | <b>18</b> |
| <b>II.2.4. Modules d'entrée et de sortie sur la carte ESP32.....</b>        | <b>18</b> |
| <b>II.3. Mécanisme de fonctionnement de l'ESP32.....</b>                    | <b>21</b> |
| <b>III. Logiciel de programmation Arduino IDE .....</b>                     | <b>22</b> |
| <b>IV. Module de puissance driver l298N .....</b>                           | <b>23</b> |
| <b>IV.2. Description des principales broches du L298N .....</b>             | <b>24</b> |
| <b>IV.3. Architecture interne du module L298N .....</b>                     | <b>25</b> |
| <b>IV.4. Principe de fonctionnement.....</b>                                | <b>26</b> |
| <b>IV.4.1. Les configurations possibles .....</b>                           | <b>28</b> |
| <b>IV.4.2. Pulse Width Modulation (PWM) .....</b>                           | <b>29</b> |
| <b>IV. Plateforme de communication Blynk.....</b>                           | <b>30</b> |
| <b>IV.2. Configuration de l'ESP32 avec l'application Blynk .....</b>        | <b>30</b> |
| <b>V.Moteurs à courant continu (DC) .....</b>                               | <b>31</b> |
| <b>VI. Module de prototype .....</b>                                        | <b>32</b> |
| <b>VI.2. Les fils de connexion.....</b>                                     | <b>32</b> |

|                                                                  |           |
|------------------------------------------------------------------|-----------|
| <b>VII. Conclusion .....</b>                                     | <b>32</b> |
| <b>Chapitre III Simulation &amp; réalisation du projet .....</b> | <b>33</b> |
| <b>I. Introduction .....</b>                                     | <b>34</b> |
| <b>II. Partie simulation.....</b>                                | <b>34</b> |
| <b>II.1. Logiciel de simulation PROTEUS.....</b>                 | <b>35</b> |
| <b>II.1.1. Interface ISIS.....</b>                               | <b>35</b> |
| <b>II.1.2. Interface ARES.....</b>                               | <b>35</b> |
| <b>II.2. Description du système simulé.....</b>                  | <b>37</b> |
| <b>II.2.1. Bloc de commande.....</b>                             | <b>38</b> |
| <b>II.2.2. Bloc de traitement.....</b>                           | <b>39</b> |
| <b>II.2.3. Bloc de puissance.....</b>                            | <b>41</b> |
| <b>II.2.4. Bloc d'action .....</b>                               | <b>42</b> |
| <b>II .3. Les résultats de simulation.....</b>                   | <b>42</b> |
| <b>III. Partie réalisation .....</b>                             | <b>46</b> |
| <b>III.1. Principe de fonctionnement .....</b>                   | <b>48</b> |
| <b>III.2. Validation des résultats .....</b>                     | <b>55</b> |
| <b>IV. Conclusion.....</b>                                       | <b>56</b> |
| <b>Conclusion générale et perspectives .....</b>                 | <b>57</b> |
| <b>Conclusion générale et perspectives.....</b>                  | <b>58</b> |
| <b>Résumé .....</b>                                              | <b>59</b> |
| <b>Abstract.....</b>                                             | <b>59</b> |
| <b>ملخص.....</b>                                                 | <b>59</b> |
| <b>Références bibliographiques .....</b>                         | <b>60</b> |
| <b>Annexes .....</b>                                             | <b>62</b> |

## Liste des figures

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| <b>Figure I.1.</b> Pollution atmosphérique .....                             | 4  |
| <b>Figure I.2.</b> Les sources naturelles de pollution .....                 | 5  |
| <b>Figure I.3.</b> la plantation d'arbres .....                              | 5  |
| <b>Figure I.4.</b> Pollution des sols .....                                  | 6  |
| <b>Figure I.5.</b> Le ramassage des déchets .....                            | 7  |
| <b>Figure I.6.</b> Pollution hydrique .....                                  | 7  |
| <b>Figure I.7.</b> Les différents types des rejets .....                     | 8  |
| <b>Figure I.8.</b> Méthode traditionnelle de nettoyage des piscines .....    | 9  |
| <b>Figure I.9.</b> Robot de nettoyage à système filaire .....                | 10 |
| <b>Figure I.10.</b> Technologie des systèmes autonomes de nettoyage .....    | 11 |
| <b>Figure I.11.</b> Drone semi-autonome de nettoyage des eaux » LAUTUS ..... | 13 |
| <b>Figure II.1.</b> Le module ESP32.....                                     | 16 |
| <b>Figure II.2.</b> Schéma de broches de ESP32 .....                         | 18 |
| <b>Figure II.3.</b> Interface du logiciel Arduino IDE .....                  | 23 |
| <b>Figure II.4.</b> Le driver l298n .....                                    | 23 |
| <b>Figure II.5.</b> Les broches de driver l298n .....                        | 24 |
| <b>Figure II.6.</b> Schéma électronique le l298N .....                       | 25 |
| <b>Figure II.7.</b> Le schéma interne du L298N .....                         | 27 |

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| <b>Figure II.8.</b> Contrôle d'un moteur utilisant le driver L298N dans la sens direct .....   | 27 |
| <b>Figure II.9.</b> Contrôle d'un moteur utilisant le driver L298N dans la sens inverse .....  | 28 |
| <b>Figure II.10.</b> TT moteur .....                                                           | 31 |
| <b>Figure II.11.</b> Plaques d'essai .....                                                     | 32 |
| <b>Figure III.1.</b> Schéma synoptique du système .....                                        | 34 |
| <b>Figure III.2.</b> Interface du logiciel Proteus .....                                       | 36 |
| <b>Figure III.3.</b> Interface du logiciel Arduino IDE .....                                   | 37 |
| <b>Figure III.4.</b> Le circuit électronique global du système simulé sous ISIS .....          | 37 |
| <b>Figure III.5.</b> Représentation des différents étages du système simulé sous Proteus ..... | 38 |
| <b>Figure III.6.</b> Schéma du système simulé en mode avant .....                              | 43 |
| <b>Figure III.7.</b> Schéma du système simulé en mode recul .....                              | 44 |
| <b>Figure III.8.</b> Schéma du système simulée en mode rotation vers la gauche .....           | 45 |
| <b>Figure III.9.</b> Schéma du système simulée en mode rotation vers la droite .....           | 45 |
| <b>Figure III.10.</b> Schéma de commande des moteurs M3 et M4 du convoyeur .....               | 46 |
| <b>Figure III.11.</b> Schéma de câblage et d'interconnexion du système .....                   | 47 |
| <b>Figure III.12.</b> Prototype final du système.....                                          | 47 |
| <b>Figure III.13.</b> Interface de commande du système via l'application Blynk.....            | 48 |
| <b>Figure III.14.</b> Paramètres de configuration Blynk pour la liaison avec l'ESP32.....      | 49 |
| <b>Figure III.15.</b> Paramétrage de l'interface Blynk AQUA-NET.....                           | 50 |
| <b>Figure III.16.</b> Architecture de communication entre l'ESP32 et l'application Blynk ..... | 50 |
| <b>Figure III.17.</b> Commande d'un moteur DC via driver L298N .....                           | 51 |
| <b>Figure III.18.</b> Architecture de contrôle du deux moteurs utilisant le driver L298N ..... | 52 |

|                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------|----|
| <b>Figure III.19.</b> Variation de la tension et la vitesse appliqué au moteur par signal PWM.....  | 54 |
| <b>Figure III.20</b> Déplacement du nettoyeur vers l'avant via l'application Blynk.....             | 55 |
| <b>Figure III.21</b> Déplacement du nettoyeur vers l'arrière via l'application Blynk.....           | 55 |
| <b>Figure III.22</b> Activation du système de convoyeur pour la collecte des déchets flottants..... | 56 |

## Liste des tableaux

|                                                                                                  |           |
|--------------------------------------------------------------------------------------------------|-----------|
| <b>Tableau II.1.</b> Les broches de ESP32 et leur fonctionnement.....                            | <b>19</b> |
| <b>Tableau II.2.</b> Table de vérité de fonctionnement de L298N.....                             | <b>29</b> |
| <b>Tableau III.1.</b> Connexion des broches des drivers L298N avec la carte Arduino Uno.....     | <b>39</b> |
| <b>Tableau III.2.</b> Les entrées et sorties du driver L298N pour les différents mouvements..... | <b>42</b> |
| <b>Tableau III.3.</b> États des entrées INT1 et INT2 pour le contrôle du moteur A.....           | <b>52</b> |
| <b>Tableau III.4.</b> Commandes des moteurs de déplacement du nettoyeur.....                     | <b>53</b> |

# Liste des abréviations

Wi-Fi: Wireless Fidelity.

IOT: Internet of Things.

RAM: Random Access Memory.

DRAM: Data RAM.

IRAM: Instruction RAM.

ROM: Read Only Memory.

RTC: Real Time Clock.

GPIO: General Purpose Input/Output.

ADC: Analog-to-Digital Converter.

SPI: Serial Peripheral Interface.

I2C: Inter-integrated circuit.

SPI: Serial Peripheral interface.

SCL: Serial Clock Line.

SDA: Serial Data Line.

MOSI: Master Out Slave In.

MISO: Master In Slave Out.

SCK: Serial Clock.

CS: Chip Select.

UART: Universal Asynchronous Receiver Transmitter.

ISIS: Intelligent Schematic Input System.

LED: Light Emitting Diode.

# **Introduction générale**

### Introduction générale

Avec le développement industriel et les différentes activités humaines de notre époque, les problèmes environnementaux se sont aggravés de manière remarquable, notamment la pollution des milieux aquatiques par les déchets flottants, en particulier les déchets plastiques qui représentent environ 85 % des déchets marins. Cette situation a conduit à la recherche et au développement de plusieurs solutions visant à limiter ce phénomène.[1]

En effet, grâce aux avancées technologiques, il est désormais possible d'utiliser des systèmes performants qui facilitent les opérations de nettoyage, améliorent leur efficacité et réduisent les efforts humains, parmi lesquels figurent les nettoyeurs flottants. C'est dans ce contexte que s'inscrit notre projet. En effet, notre projet porte sur l'étude et la réalisation d'un nettoyeur flottant contrôlé à distance à l'aide de la technologie Wi-Fi.

Ce système permet de déplacer le nettoyeur dans différentes directions afin de collecter les déchets flottants présents à la surface de l'eau.

Il représente ainsi une solution pratique et technologique contribuant à la protection de l'environnement aquatique, avec la possibilité d'être développée et améliorée grâce à l'intégration de technologies plus intelligentes et plus efficaces.

Nous avons divisé ce mémoire en trois chapitres. Dans le premier chapitre, nous avons élaboré un état de l'art consacré à la problématique de la pollution des eaux, en présentant ses causes, ses impacts et les solutions proposées pour la réduire, notamment le système intelligent de collecte des déchets flottants développé dans ce projet.

Ensuite, nous avons présenté les différents modules et logiciels utilisés dans la réalisation du système.

Enfin, le troisième chapitre est consacré à la simulation du système et à sa réalisation pratique. Il met en évidence les différents blocs fonctionnels et présente l'évaluation des résultats obtenus.

# Chapitre I

## Etat de L'art

## I. Introduction

De nos jours, la pollution environnementale constitue l'un des problèmes majeurs qui représente un danger pour l'environnement et la santé humaine.

L'augmentation des activités industrielles, des activités urbaines et agricoles a entraîné une dégradation progressive des ressources naturelles. Cette situation a favorisé l'apparition de plusieurs formes de pollution qui affectent la qualité de l'air, du sol et de l'eau. [2]

## II. Différents types de pollution

L'environnement est exposé à différents types de pollution, notamment atmosphérique, terrestre et aquatique.

### II. 1. Pollution atmosphérique

#### II. 1. 1. Définition de la pollution atmosphérique

La pollution de l'air correspond à la présence des particules et des substances nocives dans l'atmosphère qui dégradent la qualité de l'air. [3]



**Figure I.1.** Pollution atmosphérique [4]

#### II. 1. 2. Origines et facteurs de la pollution atmosphérique

Les principales sources de la pollution atmosphérique sont les gaz nocifs causés par les émissions des transports, les fumées rejetées par les usines, les activités industrielles et énergétiques, ainsi que les sources naturelles telles que les incendies de forêt, les tempêtes de sable, ...etc. [3]



**Figure I.2.** Les sources naturelles de pollution [3]

### **II. 1. 3. Impacts de la pollution atmosphérique**

Cette pollution engendre plusieurs conséquences sur la santé humaine et sur les organismes vivants, notamment la dégradation de l'environnement, le changement climatique, ainsi que l'apparition de maladies respiratoires et cardiovasculaires...etc. [3]

### **II. 1. 4. Illustration de la pollution atmosphérique**

La lutte contre la pollution atmosphérique peut s'appuyer sur diverses mesures, telles que le reboisement, la réduction de l'usage des véhicules motorisés au profit des énergies renouvelables comme le solaire ou l'éolien, ainsi que la sensibilisation des populations aux pratiques respectueuses de l'environnement. [5]



**Figure I.3.** La plantation d'arbres [5]

## II. 2. Pollution des sols

### II. 2. 1. Définition de la Pollution des sols

La pollution des sols correspond à l'introduction de substances de diverses natures, issues de procédés industriels, de micro-organismes ou de phénomènes naturels, qui perturbent les organismes vivants et compromettent les fonctions naturelles du sol. La présence de ces contaminants entraîne une dégradation de sa qualité et de ses propriétés physiques, biologiques et chimiques. [6]



**Figure I.4.** Pollution des sols [7]

### II. 2. 2. Origines et facteurs de la pollution des sols

La dégradation des sols provient essentiellement des activités humaines, notamment l'accumulation de déchets domestiques et industriels, le déversement d'huiles usagées ainsi que l'élimination d'eaux non traitées. [8]

### II. 2. 3. Impacts de la pollution des sols

Les impacts de la pollution des sols sont multiples : atteinte aux plantes et aux animaux, menace pour la sécurité alimentaire et la santé publique, mais aussi dégradation de l'environnement et déséquilibre des écosystèmes. [9]

### II. 2. 4. Illustration de la pollution des sols

Parmi les mesures mises en œuvre pour lutter contre la pollution des sols figurent la collecte et le traitement des déchets, le recyclage des matériaux ainsi que l'application de techniques de dépollution. [10]



**Figure I.5.** Le ramassage des déchets [11]

La mauvaise gestion des déchets laissés dans le sol ne se limite pas à la pollution terrestre ; en effet, ces résidus peuvent être entraînés par le vent ou les eaux de pluie vers les barrages et les mers, contribuant ainsi à la contamination des milieux aquatiques. [12]

La richesse hydrique est une ressource naturelle fondamentale pour la vie, elle est essentielle pour l'être humain, les animaux et les plantes donc elle est nécessaire pour la préservation de l'équilibre écologique. Ainsi, elle joue un rôle fondamental dans plusieurs domaines tels que l'agriculture, l'alimentation et l'industrie

## **II. 3. Pollution hydrique**

### **II. 3. 1. Définition de la Pollution hydrique**

La pollution hydrique correspond à l'altération de la qualité de l'eau ainsi que de ses principales caractéristiques physiques, chimiques et biologiques. [13]



**Figure I.6.** Pollution hydrique [13]

### II. 3. 2. Origines et facteurs de la pollution hydrique

La pollution hydrique trouve principalement son origine dans les activités humaines. Les rejets industriels contenant des substances chimiques, les eaux usées domestiques insuffisamment traitées, ainsi que les pratiques agricoles reposant sur l'utilisation d'engrais et de pesticides constituent les sources majeures de cette dégradation environnementale. [13]

À ces causes directes s'ajoutent d'autres facteurs aggravants, tels que le dépôt de déchets solides dans les cours d'eau, les déversements accidentels d'hydrocarbures, l'urbanisation croissante et l'augmentation continue de la population. [12]



**Figure I.7.** Les différents types des rejets [14].

### II. 3. 3. Impacts de la pollution hydrique

Les résultats et les conséquences de la pollution des milieux aquatiques sont multiples. Tout d'abord, elle provoque des dommages importants sur les organismes marins, notamment la mort massive de poissons et d'autre espèces aquatiques. Ensuite, ces polluants peuvent affecter la santé humaine à travers la consommation d'eau contaminée ou de produits de la mer pollués, entraînant diverses maladies. Enfin, la pollution des milieux aquatiques perturbe l'équilibre écologique, l'extinction de certaines espèces, la perturbation des chaînes alimentaires, la diminution du nombre d'espèces vivantes. [15]

### II. 3. 4. Illustration de la pollution hydrique

Le nettoyage des milieux aquatiques est devenu une nécessité afin de préserver l'équilibre écologique et garantir une eau de qualité. Dans le cadre de ce projet, l'attention se porte plus particulièrement sur la pollution de l'eau.

Ainsi, une problématique essentielle se pose : quelles solutions efficaces peuvent être mises en œuvre pour assurer la dépollution des milieux aquatiques et protéger durablement les ressources hydriques ? [12]

### III. Présentation des solutions existantes

Les opérations de maintien de la propreté des plages et des piscines peuvent être réalisées selon diverses modalités. Elles comprennent notamment le ramassage des déchets flottants à la surface de l'eau, effectué par des méthodes classiques, telles que le nettoyage manuel ou par des procédés technologiques, qu'ils soient filaires, automatisés ou semi-automatisés. [12]

#### III.1. Méthodes classiques

##### III.1.1. Nettoyage manuel

Le nettoyage manuel représente l'une des méthodes les plus traditionnelle utilisées pour l'élimination des déchets flottants à la surface des plans d'eau. Cette technique basée sur la collecte directe des déchets par l'homme, qui consiste à enlever les déchets manuellement en utilisant des gants et à l'aide d'un outil simple tels que, les sacs et les filets.



**Figure I.8.** Méthode traditionnelle de nettoyage des piscines [16].

##### III.1.1.1. Avantages du nettoyage manuel

Cette méthode présente plusieurs avantages, parmi lesquels le coût relativement abordable des équipements et des produits de nettoyage, la simplicité de leur utilisation ainsi que leur large disponibilité, ces dispositifs étant déjà connus et couramment employés.

### **III.1.1.2. Limites du nettoyage manuel**

Bien que cette méthode présente certains avantages, elle comporte également des limites notables, telles que sa forte dépendance à l'intervention humaine, son rendement insuffisant lorsqu'il s'agit de couvrir de vastes surfaces, ainsi que les contraintes rencontrées dans les zones caractérisées par une pollution élevée ou difficilement accessibles.

### **III.2. Méthodes Technologiques**

Le nettoyage des milieux aquatiques constitue une opération complexe, car les déchets peuvent être dispersés, flotter à la surface ou être immergés, tandis que certaines zones polluées demeurent dangereuses ou difficiles d'accès. Ces contraintes limitent l'efficacité des méthodes traditionnelles de collecte et rallongent considérablement le temps nécessaire à leur mise en œuvre. Dans ce contexte, le recours à des solutions technologiques adaptées, qu'elles soient filaires, automatisées ou semi-automatisées, apparaît indispensable pour assurer la collecte des déchets dans toutes les conditions et faciliter leur élimination. L'utilisation de ces technologies permet de gagner en efficacité tout en améliorant l'organisation du travail.

#### **III.2.1. Les systèmes filaires**

Les systèmes filaires ont constitué une première étape importante dans l'évolution des technologies de nettoyage des piscines. Les premiers modèles apparus sur le marché étaient reliés directement à une source d'alimentation externe par un câble électrique. Ce câble fournissait l'énergie nécessaire au fonctionnement du moteur et des dispositifs de filtration intégrés.

Le déplacement du robot est assuré par des moteurs électriques, commandés par des circuits simples, permettant une couverture partielle de la surface. [17]

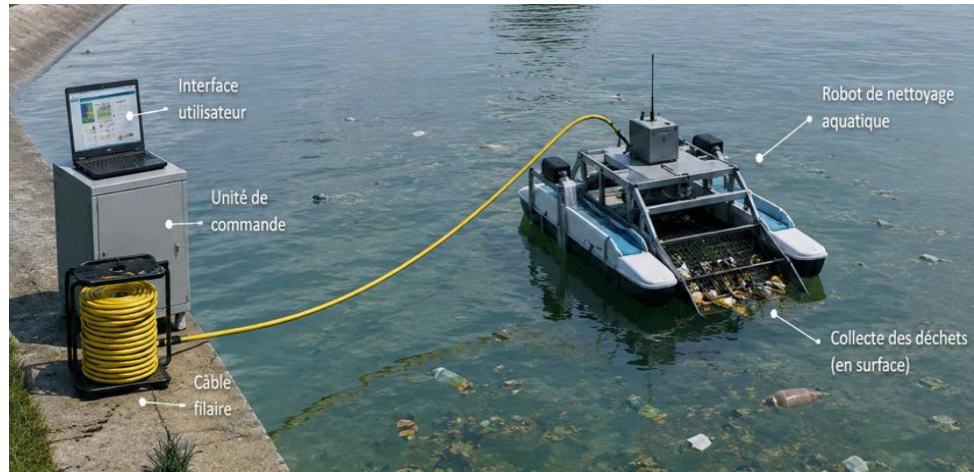


**Figure I.9.** Robot de nettoyage à système filaire [18]

### III.2.1.1. Avantages des systèmes filaires

Les systèmes filaires demeurent attractifs pour certaines applications, car ils garantissent une fiabilité énergétique et une durée de fonctionnement illimitée. Ainsi, leur connexion permanente à une alimentation externe assure une continuité de service, ce qui représente un avantage notable dans le contexte des premières générations de robots de nettoyage.

### III.2.1.2. Limites des systèmes filaires

Malgré leur efficacité initiale, les systèmes filaires présentent plusieurs limites. Tout d'abord, la présence du câble réduisait la liberté de mouvement du robot et pouvait entraîner des enchevêtrements ou des blocages, diminuant ainsi la performance du nettoyage. Ensuite, la dépendance à une alimentation externe limitait la flexibilité d'utilisation, car le robot ne pouvait fonctionner qu'à proximité d'une prise électrique, ce qui se traduisait par une couverture incomplète et parfois répétitive de certaines zones, ces limitations ont conduit au développement des systèmes autonomes sans fil, plus flexibles et plus performants.

### III.2.2. Les systèmes autonomes

Ces systèmes constituent la génération la plus avancée des robots de nettoyage des milieux aquatiques. Leur fonctionnement repose sur l'intégration de capteurs intelligents

(ultrasons, gyroscopes, caméras, capteurs de pression) et d'algorithmes de navigation capables de cartographier la zone ciblée et d'optimiser les trajectoires de nettoyage. [19]



**Figure I.10.** Technologie des systèmes autonomes de nettoyage [20].

Ces dispositifs sont généralement alimentés par une batterie haute capacité et disposent de systèmes de filtration intégrés. Ils peuvent détecter les obstacles et adapter leur vitesse. [19]

La figure 1.10 représente une technologie innovante utilisée pour la collecte des déchets dans les milieux aquatiques. Il s'agit d'un robot autonome capable de ramasser les rejets flottants sur la surface de l'eau.

### **III.2.2.1. Avantages des systèmes autonomes**

Le principal avantage des systèmes autonomes est l'autonomie totale, permettant de réduire significativement l'intervention humaine et de garantir une couverture opérationnelle complète et optimisée. [20]

### **III.2.2.2. Limites des systèmes autonomes**

Malgré leur efficacité démontrée, les systèmes autonomes présentent certaines limites. Leur premier inconvénient réside dans le coût élevé lié à l'intégration de capteurs sophistiqués, d'algorithmes de navigation avancés et de batteries de grande capacité. Ensuite, leur complexité technique qui se traduit par une maintenance délicate. En cas de panne, les réparations nécessitent des composants spécifiques et souvent coûteux, ainsi que des compétences techniques avancées, ce qui limite leur accessibilité pour une large partie des utilisateurs. [19]

### III.2.3. Les systèmes semi-automatisés

Les systèmes semi-automatisés reposent sur une combinaison entre commande manuelle et automatisation partielle. Concrètement, l'utilisateur pilote le dispositif à distance, généralement via une télécommande, tandis que certaines fonctions comme la propulsion ou la collecte des déchets sont motorisées et automatisées. [21]

Leur fonctionnement repose sur un microcontrôleur embarqué, qui reçoit les instructions de l'utilisateur et les traduit en signaux de commande pour les moteurs. Ces derniers assurent à la fois le déplacement du système et l'action de nettoyage (convoyeur). L'alimentation est généralement assurée par une batterie rechargeable, ce qui supprime la contrainte du câble et offre une plus grande liberté de mouvement.



**Figure I.11.** Drone semi-autonome de nettoyage des eaux » LAUTUS » développé par NAVTEK [22]

#### III.2.3.1. Avantages des systèmes semi-automatisés

Les avantages de cette approche résident dans la simplicité de conception, la réduction des coûts et la flexibilité d'utilisation, puisque l'utilisateur conserve un contrôle direct sur le dispositif. [23] Ce constat justifie l'importance de développer un système flottant innovant, commandé via une application Android et reposant sur des composants électroniques accessibles tels que l'ESP32 et les drivers L298N. En réunissant les avantages des solutions existantes : autonomie, simplicité et efficacité.

Cette méthode a une efficacité élevée dans le ramassage des déchets flottants sur l'eau, récupération directe des déchets papiers, métalliques et plastiques, moins de contact humain

direct avec les rejets, une solution écologique si le système fonctionne à une énergie électrique ou solaire. [24]

## IV. Description générale du système

Le système de nettoyage réalisé repose sur une combinaison cohérente de principes physiques, de mécanismes mécaniques et de technologies électroniques. La flottabilité assure la stabilité de la structure, tandis que la propulsion est générée par des moteurs électriques, dont la puissance et la direction sont régulées par des drivers L298N, garantissant un déplacement contrôlé. La collecte des déchets flottants est assurée par un convoyeur motorisé, dont l'efficacité dépend de la synchronisation entre le mouvement du dispositif et l'action de récupération. Sur le plan électronique, le microcontrôleur ESP32 reçoit les instructions transmises via l'application Android Blynk et pilote les moteurs, assurant ainsi la coordination et la fiabilité du système. Enfin, l'interface utilisateur, connectée en Wi-Fi, permet un contrôle intuitif en temps réel de la vitesse, de la direction et du convoyeur, illustrant l'intégration des technologies numériques dans un dispositif mécanique semi-automatisé.

## V. Conclusion

En conclusion, ce chapitre nous a permis de présenter les notions générales relatives à la pollution environnementale, en mettant l'accent sur la pollution des eaux, ses principales sources ainsi que ses impacts sur l'environnement, la santé humaine et l'équilibre écologique.

Face à l'augmentation des déchets flottants dans les milieux aquatiques, il est indispensable de développer des solutions de nettoyage efficaces.

Dans ce contexte, notre projet propose la conception d'un système de collecte des déchets flottants, capable d'améliorer les opérations de nettoyage tout en réduisant l'intervention humaine.

Cette étude préliminaire constitue ainsi une base essentielle pour la suite du travail, consacrée à l'analyse, à la conception et à la réalisation du prototype.

# **Chapitre II**

## **Description des composants utilisés dans le projet**

## I.Introduction

Notre projet consiste à concevoir un système avancé de nettoyage des milieux aquatiques. Ce système est basé sur un robot flottant télécommandé, équipé d'un convoyeur permettant de récupérer les déchets, de les transporter et de les acheminer vers une zone de stockage dédiée. Afin de garantir un fonctionnement optimal, le système intègre plusieurs modules matériels et logiciels assurant la coordination et la fiabilité de l'ensemble.

## II. Module de contrôle basé sur l'ESP32

L'ESP32 est un microcontrôleur conçu par la société chinoise Espressif Systems en 2016, spécialisée dans le développement de circuits intégrés destinés aux applications de communication sans fil. Ce composant se distingue par plusieurs caractéristiques notables

- Connectivité Wi-Fi intégrée
- Support du Bluetooth
- Processeur puissant à double cœur
- Large éventail de broches GPIO disponibles.

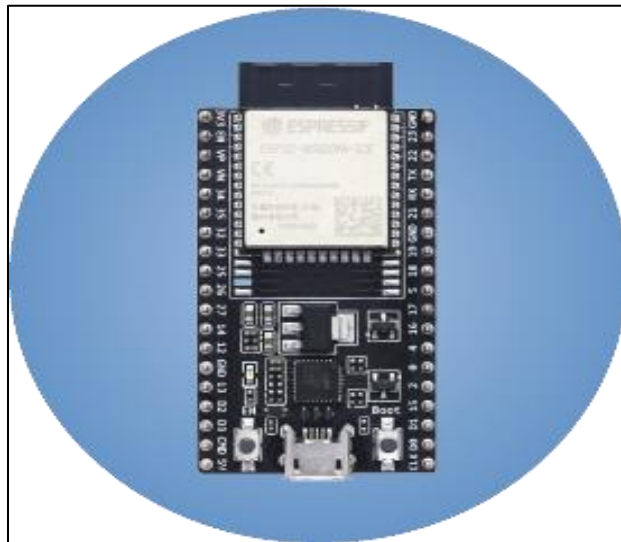


**Figure II.1** Le module ESP32 [25]

## II.2. L'architecture interne de l'ESP32

### II.2.1. Unité Centrale de Traitement CPU

L'ESP32 intègre deux cœurs indépendants de type Xtensa LX6. Le premier cœur (core 0) est généralement dédié à la gestion des connexions Wi-Fi et Bluetooth, tandis que le second (core 1) est réservé à l'exécution et au traitement des programmes. Ces deux unités de calcul peuvent fonctionner à une fréquence allant jusqu'à 240 MHz, ce qui confère au microcontrôleur la capacité de réaliser simultanément plusieurs tâches avec une grande rapidité.

Elles partagent en outre certaines ressources essentielles telles que la mémoire, les périphériques et le bus de données.

### II.2.2. Mémoire de l'ESP32

Le microcontrôleur ESP32 intègre plusieurs types de mémoire, chacun ayant un rôle spécifique

- **RAM interne** : environ 520 KB, organisée en deux parties distinctes. La première, appelée DRAM, qui est utilisée pour le stockage des variables ainsi que des données générées durant l'exécution. Aussi, la IRAM, qui est réservée au stockage de certaines instructions du code, permettant ainsi une exécution plus rapide et efficace des programmes.
- **Mémoire Flash** : d'une capacité de 4 MB ou plus, qui sert à conserver le programme téléchargé via l'IDE Arduino.
- **ROM** : mémoire interne fixe, contenant le bootloader et des fonctions essentielles au démarrage. Elle est immuable et constitue la première mémoire sollicitée lors de l'exécution initiale.
- **Mémoire Cache** : permettant d'accélérer l'accès au code en stockant temporairement des parties du programme, évitant ainsi la lecture directe depuis la Flash, plus lente. On distingue le Data Cache et l'Instruction Cache.
- **Mémoire RTC** : Permet de stocker des données même en mode deep sleep, comme l'état du système, Les compteurs et des variables importantes.

### II.2.3. Modules de communication sans fil

L'ESP32 intègre plusieurs modules de communication sans fil qui lui permettent d'assurer la connectivité avec Internet ou d'autres dispositifs. Le module Wi-Fi offre la possibilité de contrôler l'appareil à distance, notamment via des serveurs web ou des applications liées à l'IoT qui est une technologie permettant aux appareils de transférer et de recevoir des données entre eux via le réseau. Le module Bluetooth est quant à lui utilisé pour le transfert de volumes importants de données, comme l'audio, ou pour établir une liaison directe avec des périphériques tels que les téléphones portables. Enfin, la variante Bluetooth Low Energy (BLE) est particulièrement adaptée aux équipements fonctionnant sur batterie, car elle permet l'échange de petites quantités de données tout en garantissant une consommation énergétique minimale.

### II.2.4. Modules d'entrée et de sortie sur la carte ESP32

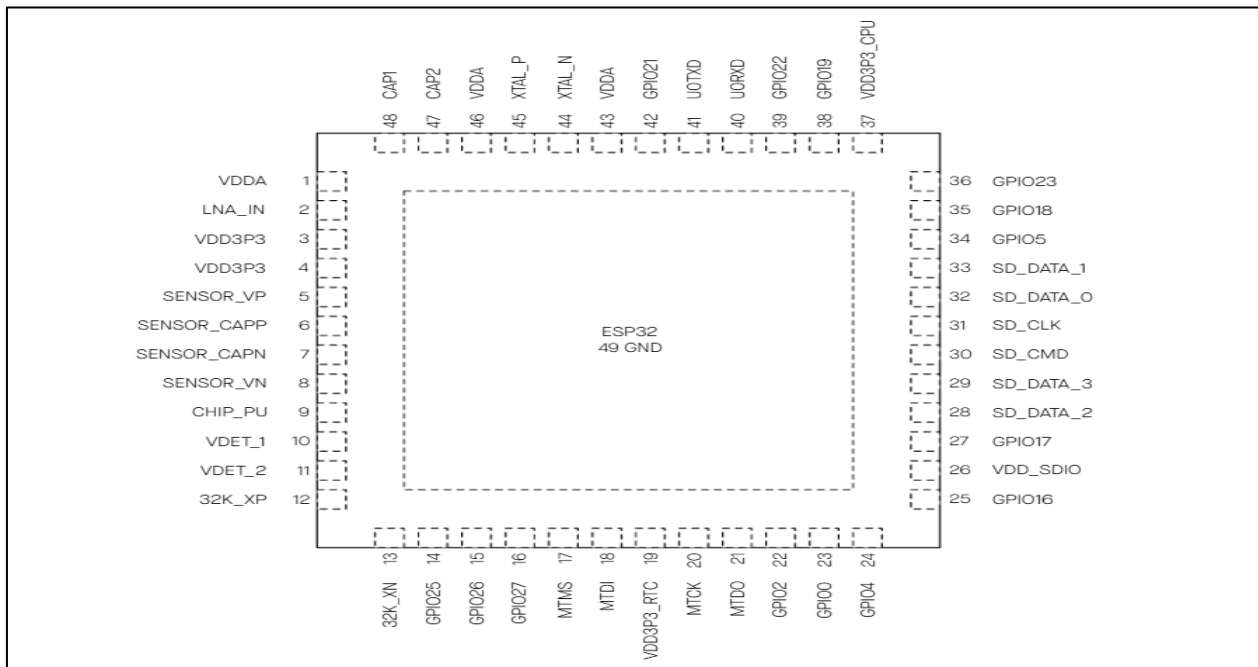


**Figure II.2** Schéma de broches de l'ESP32[26].

Le microcontrôleur ESP32 dispose d'un large éventail de broches GPIO, chacune offrant des fonctionnalités spécifiques.

**Tableau II.1.** Les broches de la carte ESP 32 et leur fonctionnement

| Pin GPIO      | Type  | Fonctionnement                                |
|---------------|-------|-----------------------------------------------|
| 0             | I/O   | Boot                                          |
| 1             | Tx    | Utiliser pour la communication serial UART    |
| 3             | Rx    |                                               |
| 2 12 15       | I/O   | Numérique, PWM, ADC2<br><br>Boot              |
| 4 13 14       | I/O   | Numérique, PWM, ADC2                          |
| 5             | I/O   | Numérique, PWM, SPI CS                        |
| 6 7 8 9 10 11 |       | SPI Flash<br><br>Ne doivent pas être utiliser |
| 16 17         | I/O   | Numérique, PWM                                |
| 18            | I/O   | SPI SCK, PWM<br><br>SPI CLK                   |
| 19            | I/O   | SPI, PWM<br><br>SPI MISO                      |
| 21            | I/O   | I2C SDA, PWM                                  |
| 22            | I/O   | I2C SCL, PWM                                  |
| 23            | I/O   | SPI MOSI, PWM                                 |
| 25 26         | I/O   | ADC2, DAC<br><br>Sorties analogiques          |
| 27            | I/O   | ADC2, PWM                                     |
| 32 33         | I/O   | ADC1, PWM                                     |
| 34 35 36 39   | Input | ADC1<br><br>Entrées seulement                 |

Le **Tableau II.1** illustre que le module ESP32 comprend plusieurs broches GPIO, dont chacune peut être configurée pour assurer des fonctionnalités spécifiques.

### **Boot pins**

Les broches dites Boot pins, également appelées strapping pins, jouent un rôle essentiel lors du démarrage ou du téléversement du code. Le microcontrôleur lit leur état afin de déterminer le mode de démarrage. La broche GPIO0 doit être maintenue à l'état bas (LOW) pour activer le mode programmation, raison pour laquelle un bouton boot est prévu afin de forcer cette condition lors du téléversement. À l'inverse, la broche GPIO2 doit être à l'état haut (HIGH) ou laissée non connectée, car un état bas pourrait empêcher le démarrage. La broche GPIO12 définit la tension de fonctionnement de la mémoire Flash : si elle est à l'état haut, la tension est fixée à 1,8 V, ce qui n'est pas approprié et entraîne des erreurs. Enfin, la broche GPIO15, lorsqu'elle est à l'état haut, bloque le démarrage de l'ESP32 et provoque des dysfonctionnements.

### **Analog to Digital converter ADC pins**

Les broches dites ADC permettent de lire des signaux analogiques et de les convertir en valeurs numériques exploitables par le microcontrôleur. Elles sont réparties en deux groupes distincts. Le premier, ADC1, est le plus utilisé car il fonctionne de manière indépendante et peut être mobilisé en parallèle avec le module Wi-Fi sans interférences. Le second, ADC2, partage certaines ressources avec le module Wi-Fi, ce qui limite son utilisation simultanée et peut entraîner des contraintes lors de la communication sans fil.

### **SPI flash pins**

Ces broches sont exclusivement réservées à la liaison entre la mémoire Flash et le processeur de l'ESP32. Elles assurent la lecture du programme lors du démarrage de la carte et ne doivent donc pas être utilisées pour d'autres fonctions, afin de garantir le bon fonctionnement du système.

**I2C pins**

Les broches de communication I2C sont utilisées pour établir une liaison série avec plusieurs périphériques tout en n'employant que deux lignes. La ligne SDA est dédiée à la transmission des données, tandis que la ligne SCL assure la synchronisation grâce au signal d'horloge.

**SPI pins**

Le protocole de communication SPI est une liaison série offrant une vitesse supérieure à celle de l'I2C. Il repose sur l'utilisation de quatre lignes principales: MOSI, qui permet à l'ESP32 d'envoyer des données vers le périphérique, MISO, qui assure la réception des données en provenance du périphérique, SCK, utilisé pour la synchronisation grâce au signal d'horloge, et enfin CS, qui sert à sélectionner le périphérique cible lors de la communication.

**UART pins**

Les broches UART de l'ESP32 permettent d'établir une communication série asynchrone avec d'autres dispositifs en utilisant uniquement deux lignes, sans signal d'horloge. La ligne Tx (Transmit) est dédiée à l'envoi des données depuis l'ESP32 vers le périphérique externe, tandis que la ligne Rx (Receive) assure la réception des données en provenance de ce périphérique.

**II.3. Mécanisme de fonctionnement de l'ESP32**

Le fonctionnement de l'ESP32 repose sur une séquence bien définie dès la mise sous tension. Au démarrage, l'exécution commence automatiquement à partir de la mémoire ROM, qui contient le programme de démarrage de base chargé d'initialiser les composants essentiels. Une fois cette étape est réalisée, le système transfère le contrôle au bootloader secondaire présent dans la mémoire Flash, lequel charge ensuite les parties nécessaires du programme principal vers la RAM afin de préparer l'exécution. Le CPU, composé de deux cœurs indépendants, assure l'exécution des instructions en parallèle et exploite la mémoire cache pour accélérer l'accès aux données et aux instructions, garantissant ainsi une performance optimale. Pour préserver certaines informations critiques, telles que l'état du système ou des variables importantes, même en cas de redémarrage ou lors du passage en mode Deep Sleep, l'ESP32 utilise la mémoire RTC .

En général, le fonctionnement de l'ESP32 repose sur une architecture intégrée où le processeur, les mémoires et les modules de communication (Wi-Fi, Bluetooth, BLE) collaborent pour offrir une plateforme polyvalente. Cette organisation permet au microcontrôleur de gérer efficacement des tâches multiples, de maintenir une connectivité sans fil stable et de garantir une faible consommation énergétique, ce qui en fait une solution adaptée aussi bien aux applications IoT qu'aux systèmes embarqués complexes.

### III. Logiciel de programmation Arduino IDE

La programmation de l'ESP32 consiste à rédiger des instructions permettant à la carte d'exécuter une grande variété de tâches. Parmi les fonctions les plus courantes, on peut citer l'allumage et l'extinction de lampes, la lecture des données issues de différents capteurs, le contrôle de moteurs, ainsi que la connexion à Internet via le module Wi-Fi intégré. Grâce à cette flexibilité, l'ESP32 s'impose comme une plateforme polyvalente pour le développement d'applications embarquées et de projets IoT.

La méthode la plus courante et simple pour programmer l'ESP32 est via le logiciel Arduino IDE qui utilise le langage C++.

Le programme Arduino se compose de deux fonctions principales : **setup()** initialise les réglages, comme la direction des broches, et ne s'exécute qu'une seule fois au démarrage ; puis **loop()** s'exécute de manière répétée afin que les commandes fonctionnent en continu.

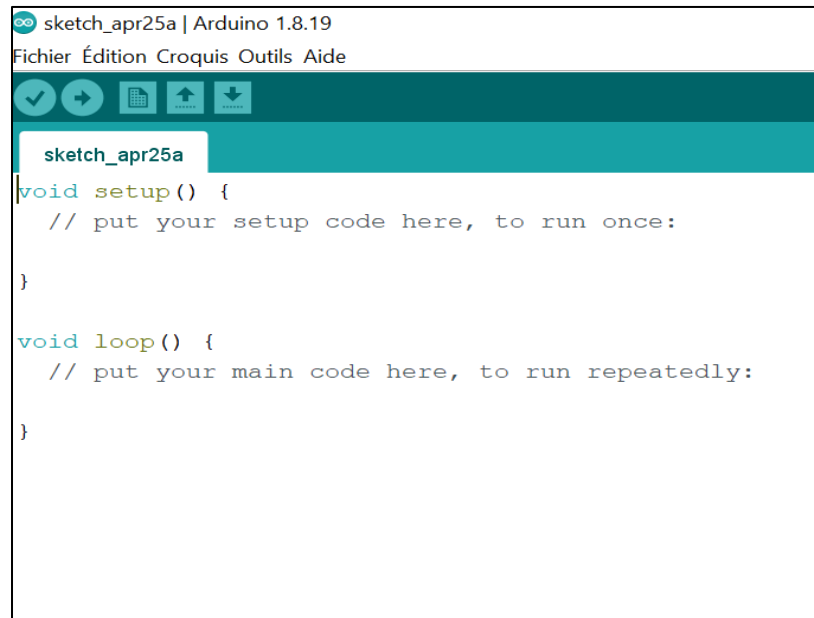


**Figure II.3** Interface du logiciel Arduino IDE

#### IV. Module de puissance Driver l298N

Le L298N est un driver moteur utilisé pour le pilotage de charges inductives telles que les moteurs à courant continu, les moteurs pas à pas ainsi que les relais. Il s'agit d'un circuit de commande qui permet de contrôler le fonctionnement des moteurs à l'aide d'une carte de développement comme l'Arduino ou l'ESP32

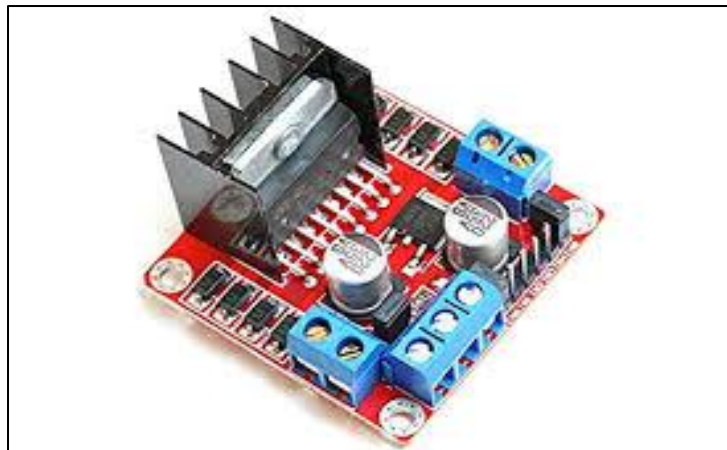


**Figure II.4.** Le driver l298N[27]

Il assure principalement la commande bidirectionnelle des moteurs, en permettant leur rotation Dans les deux sens (avant et arrière), ainsi que le contrôle de leur vitesse de rotation de manière précise.

Ainsi, le L298N joue le rôle d'interface intermédiaire entre le microcontrôleur (par exemple l'ESP32) et les moteurs, en assurant la gestion de la direction et de la vitesse.

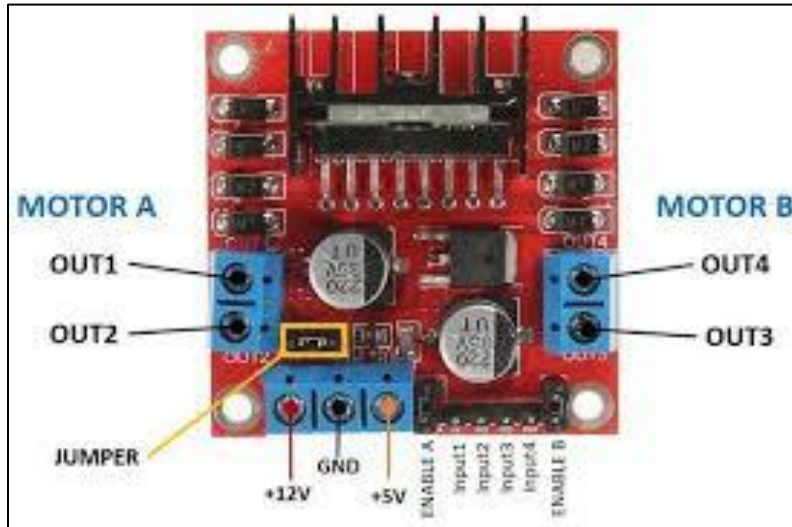


Figure II.5. Les broches de driver l298N[28]

## IV.2. Description des principales broches du L298N

Le L298N est un driver de moteur double basé sur une architecture dual H-Bridge, permettant de contrôler simultanément deux moteurs de manière indépendante.

### Broches d'alimentation

- **VCC** : alimentation destinée aux moteurs, généralement comprise entre 7 V et 12 V.
- **GND** : masse commune reliant le driver et la carte de contrôle (Arduino ou ESP32).
- **VSS** : alimentation logique, utilisée pour fournir 5 V au microcontrôleur.

### Broches de contrôle du sens de rotation

- **IN1 / IN2** : commande du sens de rotation du moteur A.
- **IN3 / IN4** : commande du sens de rotation du moteur B.

### Broches de contrôle de la vitesse par PWM

- **ENA** : permet de réguler la vitesse du moteur A grâce à un signal PWM.
- **ENB** : assure le même rôle pour le moteur B.

### Sorties vers les moteurs

- **OUT1/OUT2** : reliées aux bornes du moteur A.
- **OUT3/OUT4** : reliées aux bornes du moteur B.

## IV.3. Architecture interne du module L298N

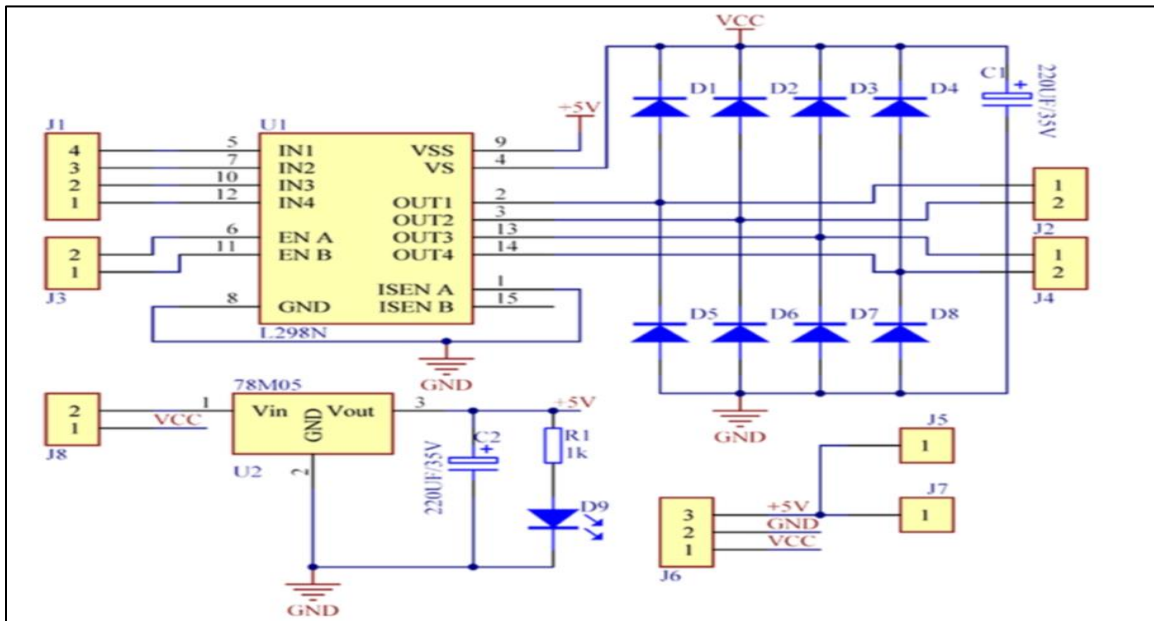


Figure II.6. Le schéma électronique du driver L298N [29].

Le pont en H intégré assure le pilotage de deux moteurs à courant continu. L'entrée J1 reçoit quatre signaux issus du microcontrôleur, tel qu'un ESP32, permettant de contrôler la direction des moteurs. L'entrée J3 est réservée au pilotage des broches ENA et ENB, qui régulent la vitesse de rotation par modulation de largeur d'impulsion (PWM).

Les sorties moteurs sont accessibles via J2 et J4 : les broches OUT1/OUT2 alimentent le moteur A, tandis que OUT3/OUT4 alimentent le moteur B. Chaque moteur est donc relié par deux fils. Les diodes D1 à D8 jouent un rôle de protection en bloquant les surtensions générées par les moteurs, évitant ainsi d'endommager le circuit intégré.

L'entrée J8 fournit l'alimentation principale, comprise entre 7 V et 12 V. Un régulateur 78M05 convertit cette tension en une alimentation stable de 5 V, nécessaire au fonctionnement du L298N et des circuits logiques associés. Les condensateurs C1 et C2 assurent la stabilisation en filtrant le bruit électrique. Une LED associée à une résistance sert d'indicateur d'alimentation : lorsqu'elle est allumée, le circuit est opérationnel. Enfin, les sorties d'alimentation J6, J5 et J7 fournissent respectivement le 5 V, la masse (GND) et la tension d'entrée (VCC).

#### **IV.4. Principe de fonctionnement**

Le module L298N intègre deux ponts en H totalement indépendants, qui repose sur quatre commutateurs électroniques, généralement des transistors bipolaires, disposés de manière à former la lettre « H ».

Pour assurer le contrôle du sens de rotation d'un moteur à courant continu, il est nécessaire d'activer simultanément deux commutateurs opposés du pont en H.

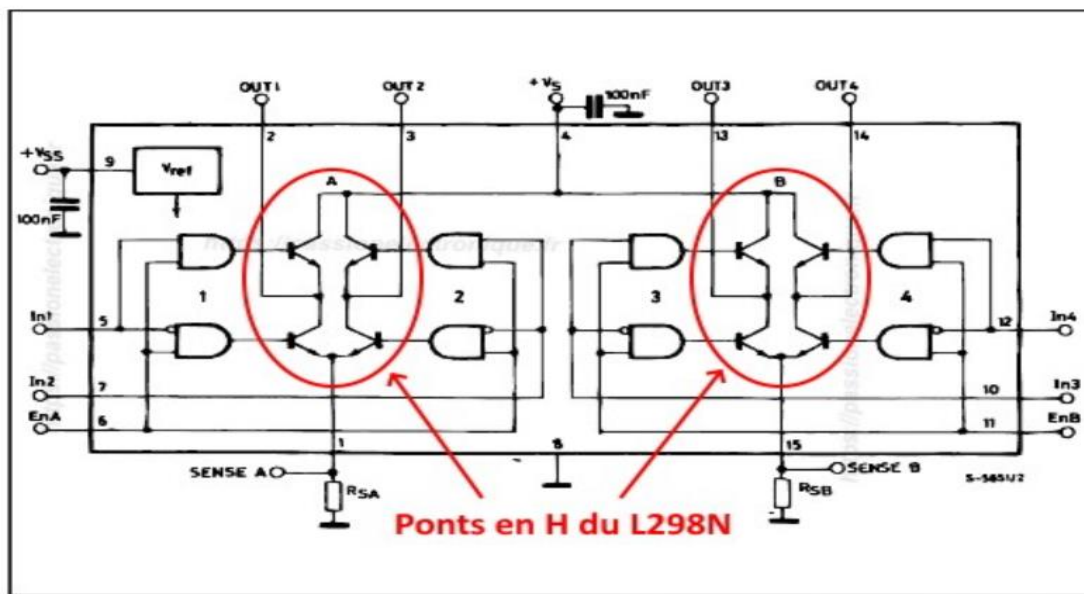


Figure II.7. Le schéma interne du L298N[30]

- En marche avant, le courant circule dans une direction donnée, ce qui entraîne la rotation du moteur dans le même sens.

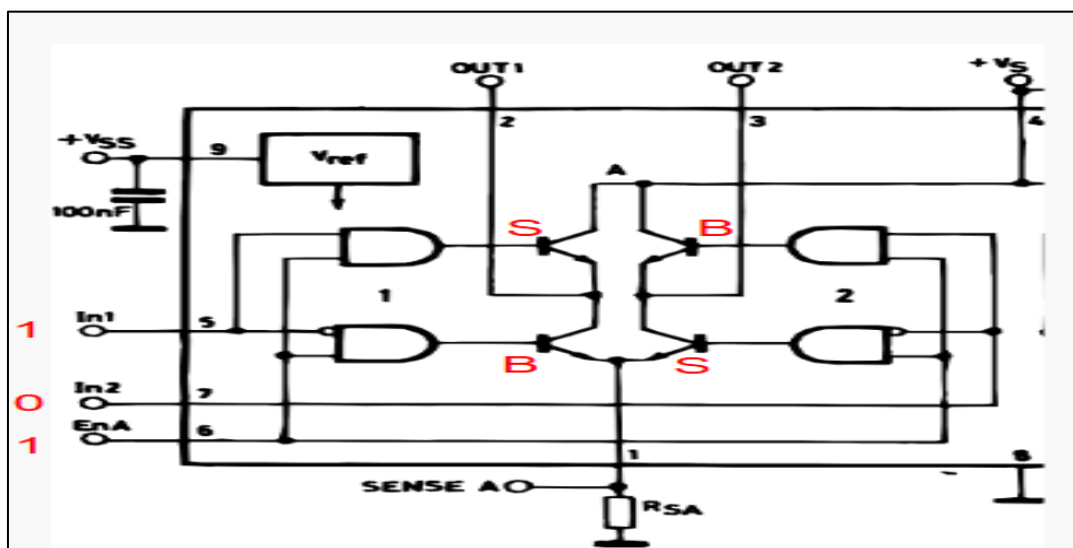


Figure II.8. Contrôle d'un moteur utilisant le driver L298N dans la sens direct [31]

- En marche arrière, on inverse l'état des commutateurs : le courant change de polarité et le moteur tourne dans le sens opposé.

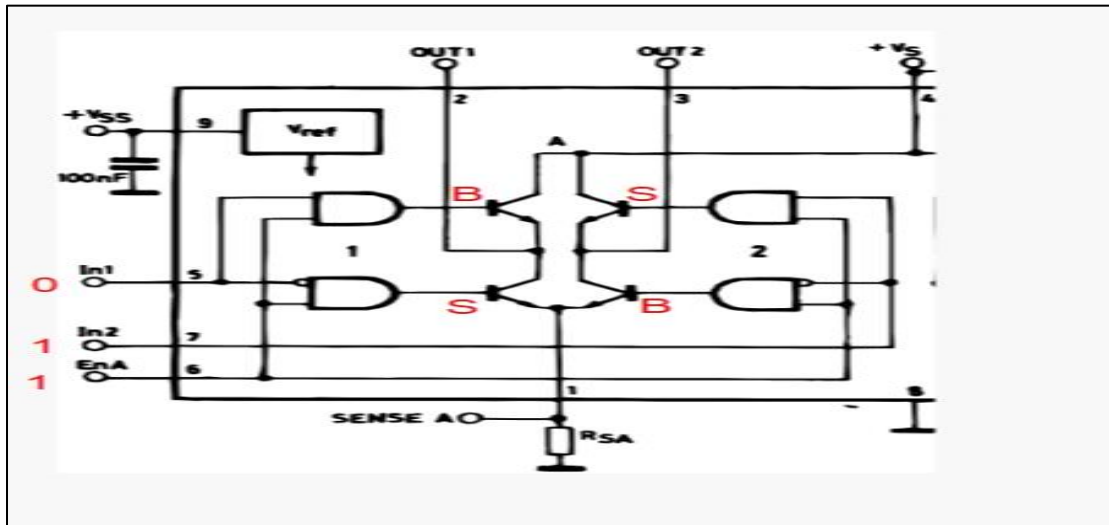


**Figure II.9.** Contrôle d'un moteur utilisant le driver L298N dans la sens inverse [32]

Ainsi, la direction du moteur dépend directement du sens du courant appliqué à ses bornes.

#### IV.4.1. Les configurations possibles

En marche avant, la borne positive du moteur est reliée à la borne positive de l'alimentation, ce qui entraîne une rotation dans le sens horaire. Les transistors sont pilotés par des portes logiques AND connectées aux différentes entrées numériques. En se concentrant sur le pont de gauche, dédié au moteur A, on constate que si  $ENA = 0$ , les transistors restent bloqués et le moteur est à l'arrêt. Pour permettre la rotation, ENA doit être activé ( $ENA = 1$ ), dans ce cas, le fonctionnement dépend des signaux appliqués à IN1 et IN2. Plus précisément, IN1 commande un côté du pont et IN2 l'autre : lorsque  $IN1 = 1$ , le moteur démarre et tourne en marche avant.

En marche arrière, la borne positive du moteur est reliée à la borne positive de l'alimentation, ce qui provoque une rotation en sens inverse. Pour le moteur A, lorsque  $IN2 = 1$ , le courant change de polarité et le moteur tourne en marche arrière.

L'arrêt du moteur se produit lorsque l'une des entrées d'activation, ENA ou ENB, est placée à l'état bas (0). Quelle que soit la configuration des broches IN1, IN2, IN3 et IN4 HIGH ou LOW.

**Tableau II.2.** Table de vérité de fonctionnement de L298N

|     |     |     | Résultat, en sortie        |
|-----|-----|-----|----------------------------|
| ENA | IN1 | IN2 | du 1 <sup>er</sup> pont A  |
| ENB | IN3 | IN4 | du 2 <sup>ème</sup> pont B |
| L   | X   | X   | L'arrêt du moteur          |
| H   | L   | L   | Blocage de moteur          |
| H   | L   | H   | marche recule              |
| H   | H   | L   | marche avant               |
| H   | H   | H   | Blocage de moteur          |

X : peu importe

H : état haut (par exemple 5V)

L : état bas (par exemple 0V)

Pour assurer un contrôle complet d'un moteur à courant continu, il est indispensable de réguler sa vitesse de rotation à l'aide d'une technique fondamentale : la modulation de largeur d'impulsion (PWM – Pulse Width Modulation).

#### IV.4.2. Pulse Width Modulation (PWM)

La modulation de largeur d'impulsion (PWM) constitue une technique essentielle pour le contrôle des moteurs à courant continu. Elle repose sur la variation de la tension moyenne appliquée au moteur, obtenue par l'envoi d'une succession d'impulsions de type ON/OFF.

En ajustant la largeur de ces impulsions, on modifie le cycle de service

Un cycle de service élevé correspond à une tension moyenne plus importante, ce qui entraîne une augmentation de la vitesse de rotation du moteur.

## IV. Plateforme de communication Blynk

Blynk est une application pour l'internet des objets IOT , elle permet notamment de concevoir une application mobile (Android et iOS) pour contrôler les systèmes électroniques à distance via un smartphone et visualiser les données d'un système embarqué (Arduino, ESP32, Raspberry pi, ...) via un serveur cloud public ou privé.

La conception de l'application mobile Blynk IoT repose sur l'utilisation d'éléments graphiques appelés widgets (tels que le Slider, le Button, le Switch, ou encore le Joystick). Le principe de conception s'appuie sur la méthode du glisser-déposer, ce qui permet de construire facilement l'interface utilisateur sans avoir à écrire de code.

Le système Blynk est constitué de trois éléments fondamentaux sont

- Application Blynk : l'utilisateur peut créer une interface graphique basé sur des widgets à l'application mobile Blynk IOT pour contrôler des systèmes embarqués à distance.
- Serveur Blynk : nous avons deux types serveur cloud public et serveur privé local, il garantir la connexion entre l'application mobile et le microcontrôleur.
- Bibliothèque Blynk : permettent la communication avec le serveur et traiter les commandes pour tous les plateformes matérielles populaires.

### IV.2. Configuration de l'ESP32 avec l'application Blynk

La configuration de l'ESP32 avec l'application Blynk IoT s'effectue de manière simple et intuitive. Il faut tout d'abord télécharger l'application sur son smartphone, puis créer un nouveau compte ou utiliser directement une adresse e-mail existante. Une fois connecté, l'utilisateur lance la création d'un nouveau projet en sélectionnant la carte ESP32. L'application génère alors automatiquement une fenêtre contenant le jeton d'authentification, indispensable pour relier le code de l'ESP32 au projet Blynk. À partir de cette interface, il est ensuite possible d'ajouter les

différents widgets nécessaires (boutons, sliders, interrupteurs, etc.) afin de concevoir l'interface utilisateur et de personnaliser le contrôle du système sans avoir à écrire de code.

## V. Moteurs à courant continu (DC)

Un moteur DC est un dispositif électromécanique qui convertit l'énergie électrique en énergie mécanique de rotation. Dans nos applications, on utilise des TT Motors, qui sont des moteurs à courant continu fonctionnant sous une faible tension comprise entre 3 V et 6 V. Ces moteurs sont intégrés dans un boîtier de réduction en plastique jaune, permettant de diminuer la vitesse de rotation tout en augmentant le couple, c'est-à-dire la force de rotation disponible.

Le sens de rotation de ce type de moteur est contrôlé simplement en inversant la polarité de l'alimentation (positif et négatif). Quant à la vitesse, elle peut être ajustée soit en modifiant le courant d'alimentation, soit de manière plus efficace en utilisant une commande par PWM .



**Figure II.10.** TT moteur [33].

## VI. Module de prototype

### VI.1. Définition du Plaque d'essai

Appelée Breadboard ou protoboard, est un support constitué d'une plaque comportant des connexions internes., permet de placer les composants et les fils, utiliser en électronique pour tester les circuits rapidement sans fixer définitivement le circuit.

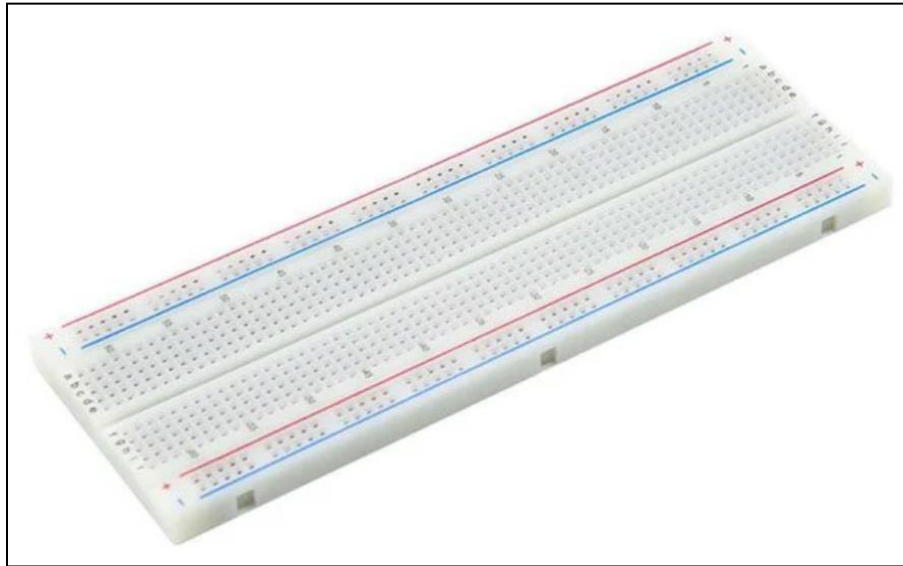


**Figure II.11.** Plaque d'essai[34]

### VI.2. Les fils de connexion

Ces fils sont des câbles utilisés pour connecter différents appareils et composants électroniques entre eux, permettent de faire circuler le courant dans le circuit. Dans ce projet on les utilise pour la connexion de l'ESP32 avec le driver, et pour l'alimentation.

## VII. Conclusion

Après avoir exposé le projet et mis en évidence son importance dans la préservation des surfaces aquatiques, ce chapitre a été consacré à la présentation détaillée des différents modules intégrés utilisés dans la réalisation ainsi qu'à l'explication de leur fonctionnement. Cette démarche offre une vision d'ensemble du système final, en soulignant la cohérence de son architecture et la contribution de chaque module à la performance globale.

# **Chapitre III**

## **Simulation & réalisation du projet**

## I. Introduction

L'objectif principal de ce projet est la conception et la mise en œuvre d'un système flottant destiné au nettoyage des plans d'eau, des piscines et autres surfaces aquatiques.

Le système est conçu pour piloter quatre moteurs à courant continu : deux assurent le déplacement du dispositif flottant dans les différentes directions, et deux autres commandent la rotation ainsi que le convoyeur destiné au transport des déchets flottants.

Ce chapitre est consacré à la présentation de l'architecture du prototype et à la simulation du circuit, constituant une phase essentielle pour valider son bon fonctionnement et concrétiser sa réalisation.

Le schéma synoptique de notre système est illustré dans la figure III.1

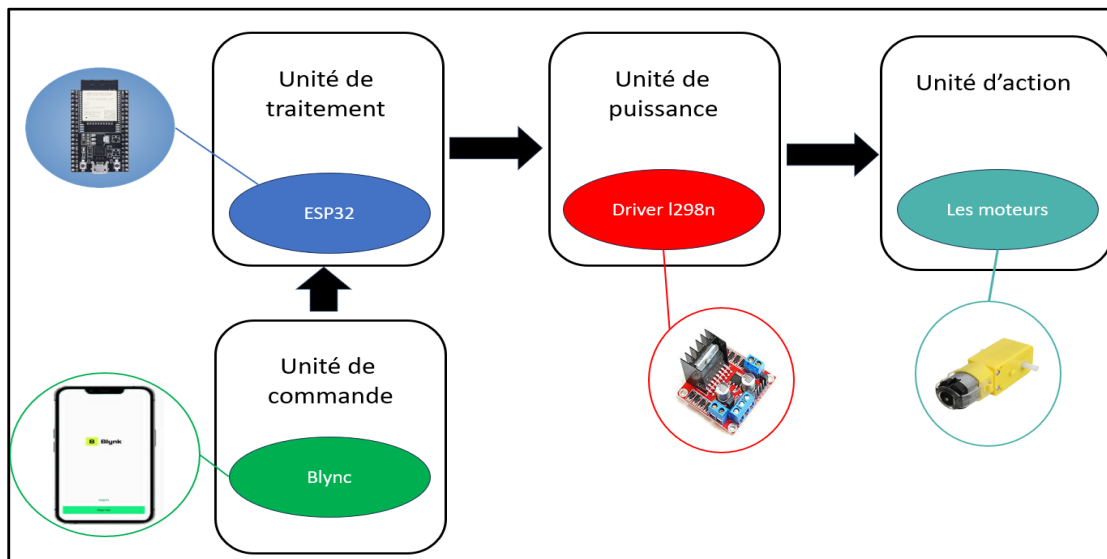


**Figure III.1.**Schéma synoptique du système

## II. Partie simulation

Avant d'entamer la phase de réalisation pratique, nous procédons d'abord à une simulation du système global.

En effet, la simulation constitue une phase essentielle dans le développement scientifique et technologique, car elle permet d'étudier et de modéliser les systèmes avant leur réalisation.

Elle offre la possibilité de comprendre le fonctionnement global, de prévenir les risques de détérioration des composants ou les erreurs de câblage, et de réduire les coûts liés aux essais expérimentaux.

Il existe plusieurs logiciels permettant la simulation des circuits électroniques et des systèmes embarqués, tels que Multisim, Tinkercad ou MATLAB avec Simulink.

Toutefois, le logiciel le plus utilisé reste Proteus Professionnel, que nous avons choisi pour la simulation du système de notre projet.

## **II.1. Logiciel de simulation PROTEUS**

Proteus est un logiciel de conception et de simulation électronique développé par Labcenter Electronics, permettant de réaliser et de tester virtuellement des circuits électroniques. Cet outil dispose d'une bibliothèque complète de composants électroniques et de microcontrôleurs, incluant notamment PIC et Arduino.

Proteus se compose de deux modules principaux : ISIS, dédié à la création de schémas et à la simulation des circuits électroniques, et ARES, destiné à la conception des cartes électroniques imprimées (PCB).

### **II.1.1. Interface ISIS**

L'interface ISIS est principalement utilisée pour l'édition de schémas électriques et la simulation du comportement des composants électroniques constituant un circuit. Elle permet ainsi de détecter d'éventuelles erreurs dès la conception et d'assurer la saisie du schéma structurel, comprenant l'assemblage des composants avec leurs valeurs et références, reliés par des connexions électriques . [35]

### **II.1.2. Interface ARES**

L'interface ARES est un logiciel d'édition et de routage qui complète parfaitement ISIS. en effet, un schéma électrique conçu sous l'environnement ISIS peut être importé dans ARES afin de réaliser le circuit imprimé (PCB). Cet outil offre des fonctionnalités de placement automatique des composants, de routage automatique, ainsi qu'une visualisation en 3D avant la fabrication.

### II.1.3. Création et simulation du circuit

La simulation de notre système nécessite l'utilisation de deux logiciels : Proteus pour la conception et la simulation du circuit électronique, et l'IDE Arduino pour la programmation du microcontrôleur.

La démarche consiste à créer un nouveau projet, sélectionner les composants nécessaires depuis la bibliothèque intégrée, puis établir le schéma souhaité.

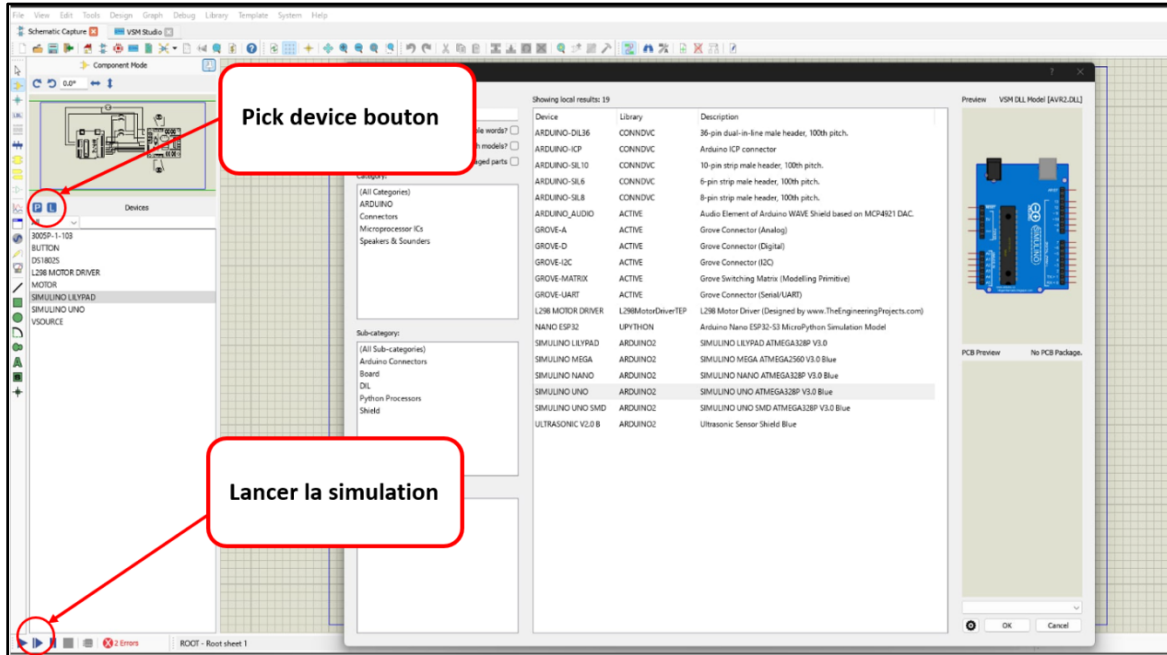


Figure III.2. Interface du logiciel Proteus

La partie programmation est ensuite réalisée dans l'environnement Arduino IDE afin de programmer l'ATmega328 de la carte Arduino.

Pour ce faire, il suffit de charger le code de programmation dans la carte Arduino afin qu'elle puisse exécuter les instructions. Ce code, est vérifié pour s'assurer de l'absence d'erreurs, puis converti en fichier au format HEX avant d'être transféré vers la carte Arduino intégrée dans Proteus.

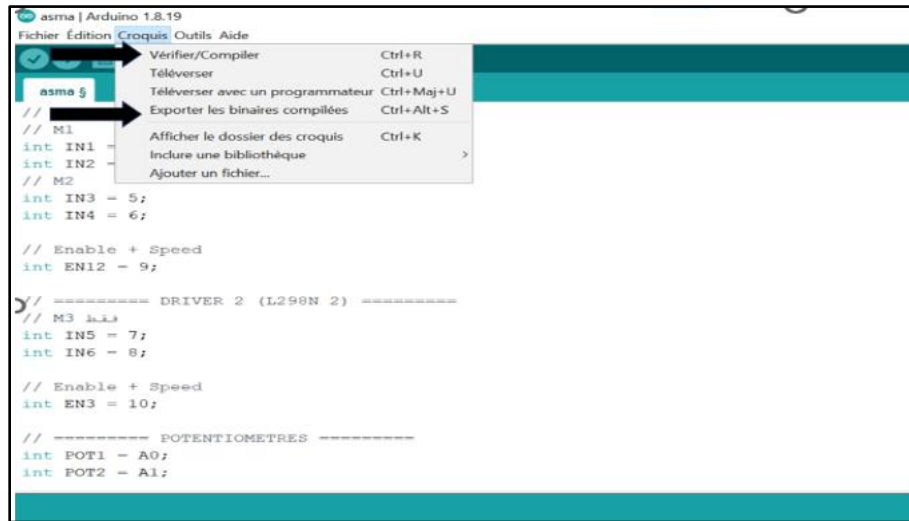


Figure III.3. Interface du logiciel Arduino IDE

## II.2. Description du système simulé

Le système simulé sous Proteus est constitué d'une carte Arduino Uno associée à des interrupteurs et des potentiomètres, tout en conservant les drivers et les quatre moteurs. Cette configuration permet d'illustrer le principe de fonctionnement du système, sans introduire la complexité liée à la communication sans fil.

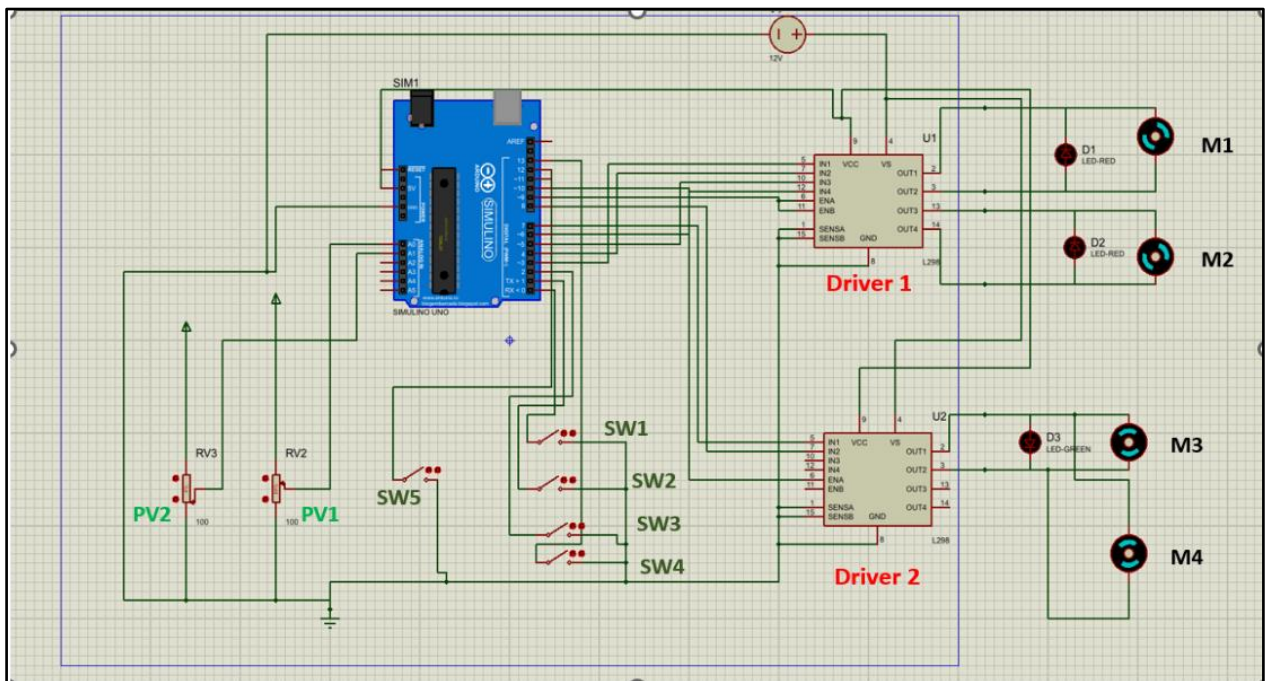


Figure III.4. Le circuit électronique global du système simulé sous ISIS

La figure III.4 illustre le schéma du système flottant conçu pour le nettoyage des eaux. Dans ce schéma notre système est structuré en quatre étages, qui sont : le bloc de commande, l'unité de traitement qui contient le programme chargé d'exécuter les tâches nécessaires à la réalisation des opérations demandées, le bloc de puissance et finalement le système d'action.

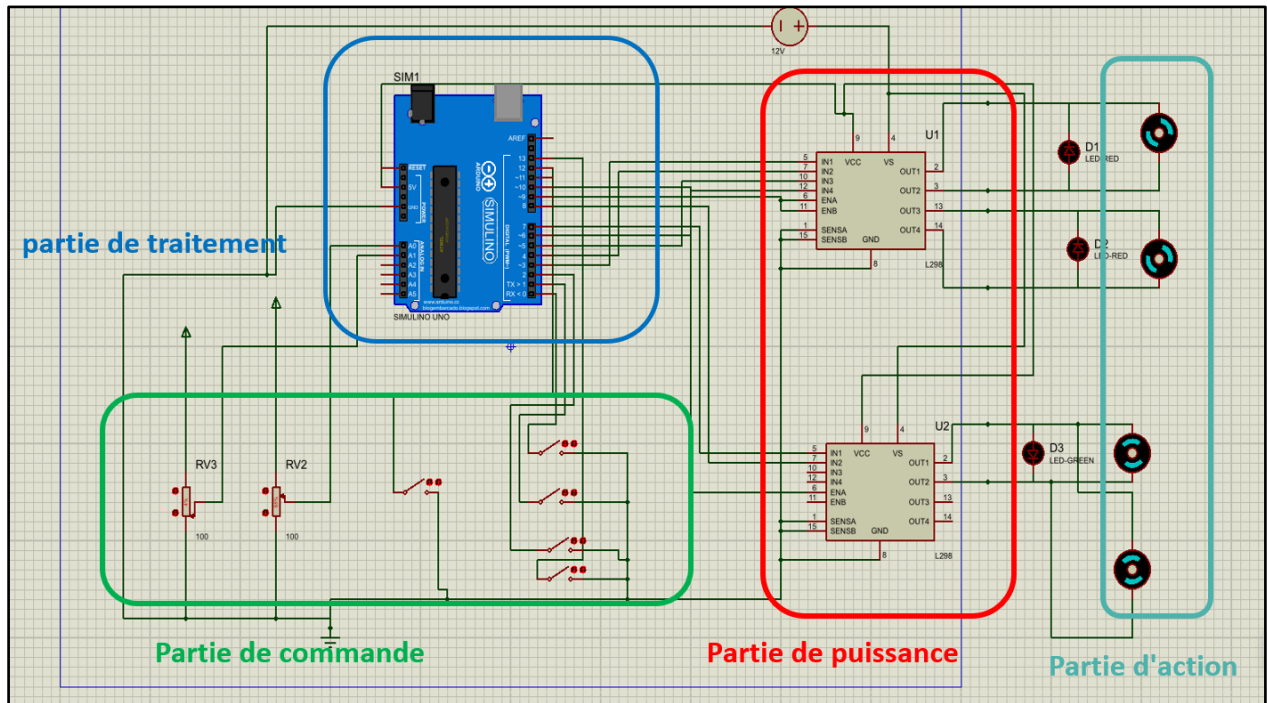


**Figure III.5.** Représentation des différents étages du système simulé sous Proteus

### II.2.1. Bloc de commande

Le bloc de commande est chargé d'introduire les commandes de contrôle dans le système. Il se compose de cinq interrupteurs (switches) et de deux résistances variables (potentiomètres). Le contrôle des moteurs M1 et M2, reliés au premier driver, s'effectue à l'aide des interrupteurs SW1, SW2, SW3, SW4 ainsi que du potentiomètre PV1. L'interrupteur SW1 active la rotation des deux moteurs dans le même sens direct, tandis que SW2 les fait tourner dans le sens inverse. SW3 permet de faire fonctionner uniquement le moteur M1, et SW4 uniquement le moteur M2. Le potentiomètre PV1, quant à lui, règle la vitesse de rotation des deux moteurs simultanément. Concernant le second driver, qui commande les moteurs M3 et M4, son contrôle repose sur l'interrupteur SW5 et le potentiomètre PV2. L'interrupteur SW5 active ou arrête les deux moteurs, tandis que PV2 ajuste leur vitesse de rotation commune.

Les deux potentiomètres PV1 et PV2 sont reliés à la carte Arduino via des broches dédiées à la lecture des signaux analogiques (ADC), en l'occurrence les pins A0 à A5. Dans notre système, la broche A0 est utilisée pour PV1 et la broche A1 pour PV2.

### II.2.2. Bloc de traitement

L'unité de traitement du système est représentée dans la simulation par la carte Arduino Uno. Elle reçoit les signaux transmis par le bloc de commande, les analyse selon le programme stocké dans sa mémoire, puis les envoie vers les drivers L298N. La connexion entre l'Arduino et les deux drivers L298N s'effectue à l'aide des broches suivantes :

**Tableau III.1.** Connexion des broches des drivers L298N avec la carte Arduino Uno

|                 |       |              |
|-----------------|-------|--------------|
| <b>Driver 1</b> | Pin4  | Int1         |
|                 | Pin5  | Int2         |
|                 | Pin6  | EnA et EnB   |
|                 | Pin 7 | Int3         |
| <b>Driver 2</b> | Pin8  | Int1 et Int3 |
|                 | Pin9  | Int2 et Int4 |
|                 | Pin10 | EnA et EnB   |

L'activation ou la désactivation des interrupteurs est convertie en niveaux logiques 1 ou 0, ce qui permet de contrôler le fonctionnement des moteurs ainsi que leur sens de rotation. Les moteurs M1 et M2 peuvent être commandés selon cinq états distincts :

L'activation du commutateur SW1 applique un niveau logique haut (HIGH) aux entrées Int1 et Int3, tandis que les entrées Int2 et Int4 restent à un niveau logique bas (LOW). Par conséquent, les deux moteurs tournent dans le même sens direct. Ce fonctionnement synchronisé entraîne le déplacement du dispositif vers l'avant.

Lorsque le commutateur SW2 est activé, les broches reliées à Int 2 et Int 4 fournissent un niveau logique haut (HIGH), tandis que celles connectées à Int1 et Int3 délivrent un niveau logique bas (LOW).

Par conséquent, les deux moteurs tournent dans le même sens, mais en direction opposée par rapport à l'état précédent. Ce fonctionnement synchronisé a pour résultat un mouvement vers l'arrière, caractérisé par un déplacement en marche arrière, permettant au dispositif de reculer de manière stable et contrôlée.

Lorsque la valeur de SW3 est égale à 1, seul le moteur M1 est activé : la broche reliée à Int1 fournit alors un niveau logique haut (HIGH), tandis que les autres restent à l'état bas.

Par conséquent, seul M1 entre en rotation, ce qui entraîne un mouvement de l'ensemble vers la gauche. Ce fonctionnement asymétrique crée une différence de poussée entre les deux côtés, orientant le dispositif dans une trajectoire de rotation latérale.

Lorsque la valeur de SW4 est égale à 1, seul le moteur M2 est activé : la broche reliée à Int 3 fournit alors un niveau logique haut (HIGH), tandis que les autres restent à l'état bas. Par conséquent, seul M2 entre en rotation, ce qui entraîne un mouvement de l'ensemble vers la droite. Ce fonctionnement asymétrique permet d'orienter le dispositif dans une trajectoire de rotation latérale droite, favorisant un changement de direction précis et contrôlé, adapté aux manœuvres de correction ou de virage.

État d'arrêt : cet état se produit lorsque tous les commutateurs sont désactivés, c'est-à-dire qu'aucun signal n'est transmis aux broches de commande. Par conséquent, aucun des moteurs n'est activé et le dispositif demeure immobile.

Les moteurs M1 et M2 assurent les déplacements principaux (avant, arrière, gauche, droite) du dispositif et fonctionnent de manière totalement indépendante des moteurs M3 et M4. Quant à ces deux derniers, ils sont exclusivement réservés au convoyeur et ne disposent que de deux états de fonctionnement. Lorsque la valeur de SW5 est égale à 1, M3 et M4 sont activés simultanément, produisant une rotation coordonnée. En revanche, lorsque le commutateur est désactivé, aucun signal n'est transmis et les moteurs s'arrêtent.

Ainsi, les vitesses des moteurs M1, M2, M3 et M4 sont contrôlées à l'aide de résistances variables (potentiomètres) reliées à la carte Arduino. Ces composants produisent un signal analogique proportionnel à la position du curseur. Ce signal est ensuite traité par la carte et converti en un signal PWM (Pulse Width Modulation), permettant d'ajuster la largeur des impulsions envoyées aux moteurs.

Ainsi, la vitesse de rotation de chaque moteur peut être réglée de façon précise et progressive, assurant un contrôle optimal des déplacements du dispositif ainsi que du fonctionnement du convoyeur.

### II.2.3. Bloc de puissance

Cet étage est constitué de deux drivers de type L298N. Leur rôle principal est d'alimenter les moteurs en 9 V tout en assurant une séparation entre le circuit de commande et le circuit de puissance.

Cette isolation protège la carte Arduino Uno contre les courants élevés et les fluctuations générées par le fonctionnement des moteurs, notamment lors des phases critiques de démarrage ou lors des inversions de sens de rotation.

Les drivers contribuent également au contrôle des moteurs, puisqu'ils permettent

- **Démarrage et arrêt** : les moteurs ne s'activent que si le driver reçoit un signal sur les broches de commande (INT) et si les broches d'activation ENA et ENB sont différentes de zéro.
- **Changement de sens** : grâce aux signaux appliqués sur les broches INT, l'inversion de l'ordre entre HIGH et LOW modifie la polarité d'alimentation, ce qui entraîne une rotation dans le sens opposé.
- **Variation de vitesse** : les broches ENA et ENB, responsables de l'activation des deux canaux du driver, reçoivent le signal PWM généré par l'Arduino. En modulant ce signal, la tension appliquée aux moteurs varie, ce qui permet d'ajuster leur vitesse de manière progressive et précise.

**Tableau III.2.** Les entrées et sorties du driver L298N pour les différents mouvements

|                 | Les entrées du 1er driver L298N |       |       |       | Les sorties du 1er driver L298N |       |       |       |
|-----------------|---------------------------------|-------|-------|-------|---------------------------------|-------|-------|-------|
|                 | Int1                            | Int 2 | Int 3 | Int 4 | Out 1                           | Out 2 | Out 3 | Out 4 |
| <b>Forward</b>  | 1                               | 0     | 1     | 0     | 1                               | 0     | 1     | 0     |
| <b>Backward</b> | 0                               | 1     | 0     | 1     | 0                               | 1     | 0     | 1     |
| <b>Left</b>     | 1                               | 0     | 0     | 0     | 1                               | 0     | 0     | 0     |
| <b>Right</b>    | 0                               | 0     | 1     | 0     | 0                               | 0     | 1     | 0     |

#### II.2.4. Bloc d'action

Ce dernier bloc est constitué de quatre moteurs à courant continu, chargés d'exécuter les commandes et de les transformer en mouvement réel et en résultats concrets. Ces moteurs reçoivent les instructions transmises par le bloc de puissance, telles que le démarrage, l'arrêt, l'inversion du sens de rotation ou la variation de vitesse via le driver, puis les mettent en œuvre afin d'assurer le fonctionnement effectif du système.

### II .3. Les résultats de simulation

Après avoir présenté l'ensemble des blocs du système et précisé le rôle de chacun, cette section est consacrée à l'exposition des résultats obtenus lors de la simulation. Afin de rendre plus explicite le sens de rotation des moteurs dans cette simulation, nous avons utilisé des LEDs de deux couleurs : vert et rouge.

La LED verte correspond à une rotation des moteurs dans le sens horaire, tandis que la LED rouge indique une rotation dans le sens antihoraire, c'est-à-dire l'inverse du premier sens.

Comme le montre la figure III.6, lorsque le switch SW1 est fermé, les deux LEDs vertes, montées en direct et reliées aux moteurs s'allument, indiquant que ceux-ci tournent dans le sens des aiguilles d'une montre. Dans cette configuration (forward), le nettoyeur se déplace vers l'avant.

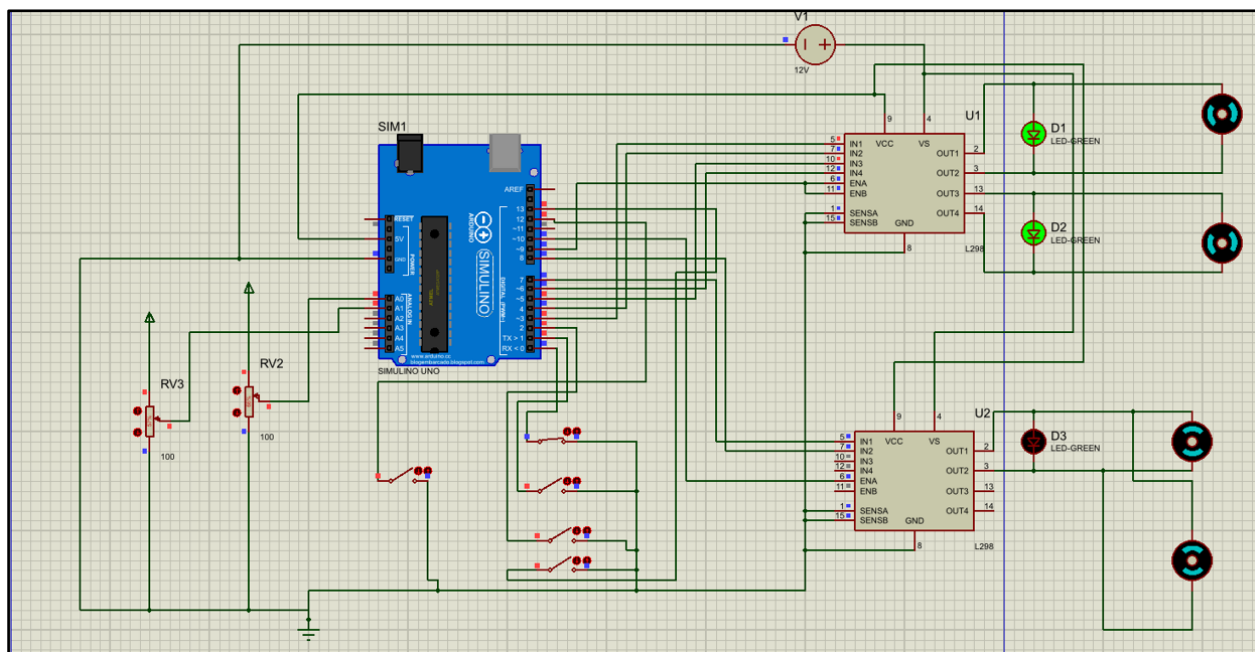


**Figure III.6.** Schéma du système simulé en mode avant

Lorsque le switch SW2 est fermé, les deux LEDs rouges, montées en inverse et reliées aux moteurs s'allument, indiquant que ceux-ci tournent dans le sens inverse des aiguilles d'une montre. Dans cette configuration (backward), le nettoyeur se déplace vers l'arrière, correspondant à la marche arrière (voir la figure III.7)

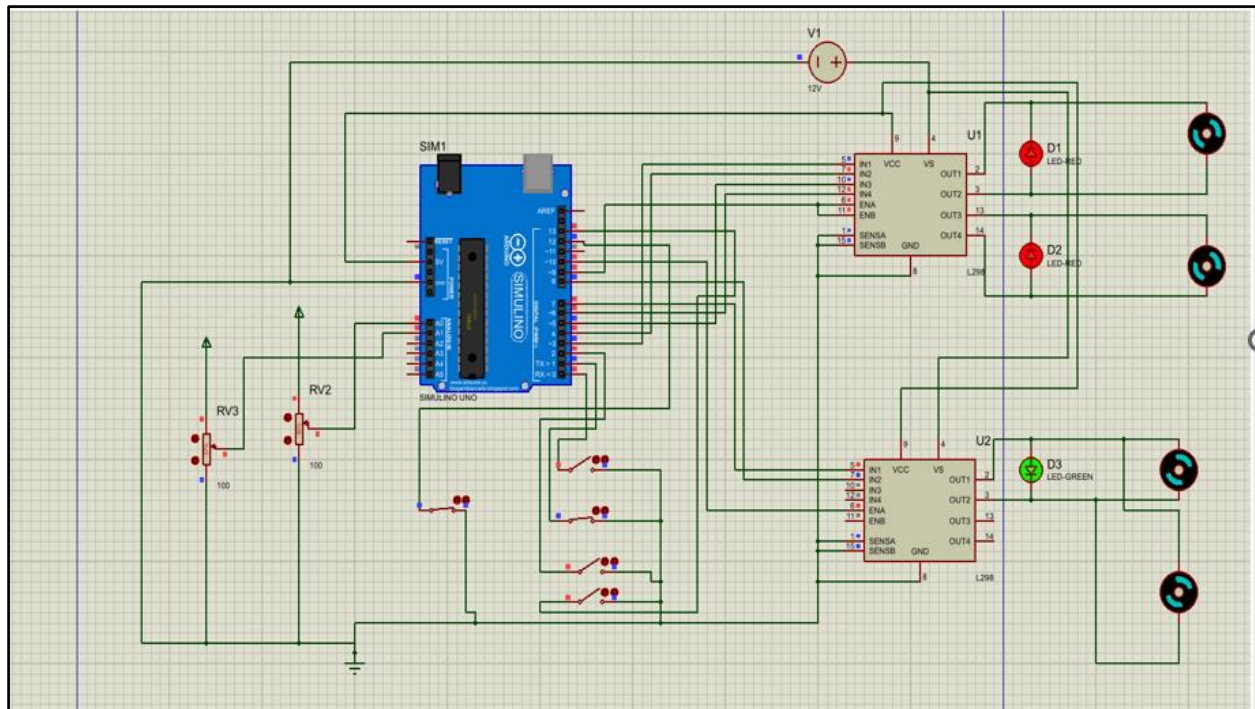


**Figure III.7.** Schéma du système simulé en mode recul

Comme le montre la Figure III.8, lorsque le switch SW3 est fermé, seule la LED verte reliée au moteur M1 s'allume, indiquant que ce dernier est en fonctionnement, tandis que le moteur M2 demeure à l'arrêt. Dans cette configuration, le système effectue une rotation vers la gauche.

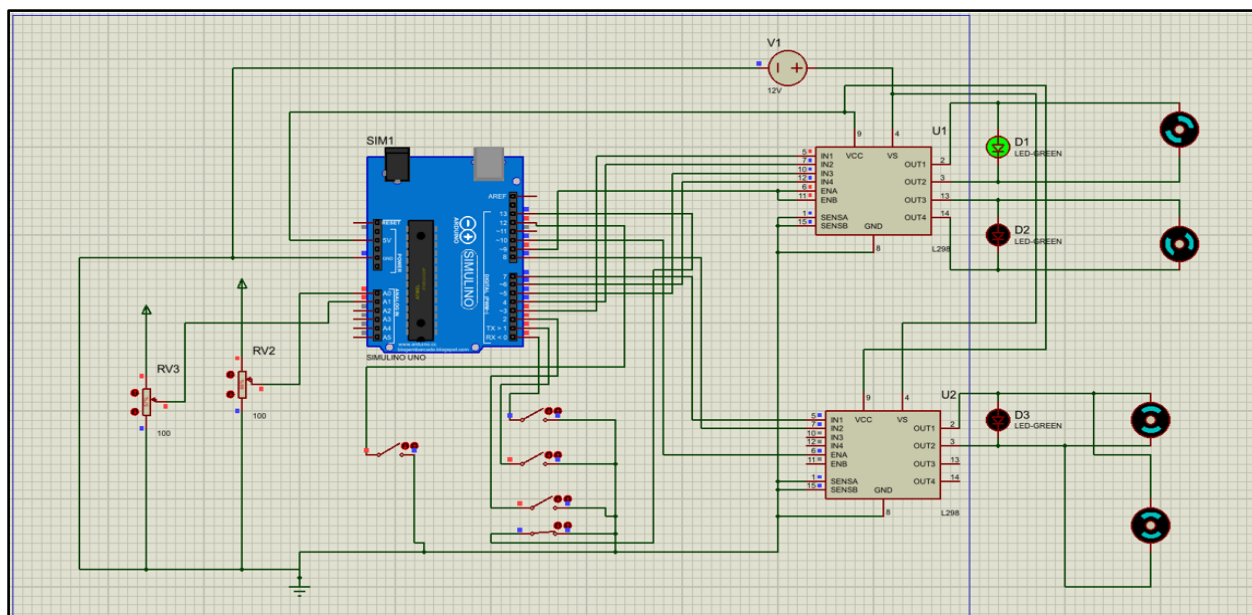


**Figure III.8.** Schéma du système simulée en mode rotation vers la gauche.

Lorsque le switch SW4 est fermé, seule la LED verte reliée au moteur M2 s'allume, indiquant que ce dernier est en fonctionnement, tandis que le moteur M1 demeure à l'arrêt (voir Figure III.9). Dans cette configuration, le **système** effectue une rotation vers la droite.

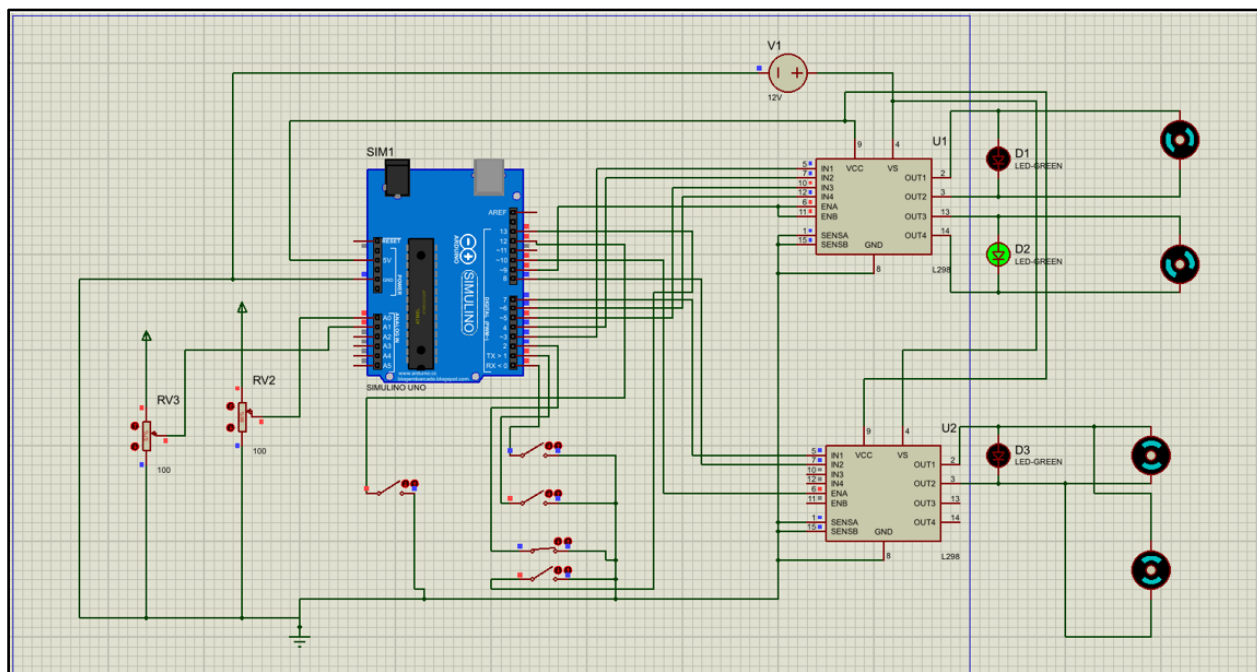


**Figure III.9.** Schéma du système simulée en mode rotation vers la droite.

Enfin, comme illustre la figure III.10, lorsque le switch SW5 est fermé, la LED verte reliée aux moteurs M3 et M4 s'allume, indiquant que ces deux moteurs sont en fonctionnement. Cette activation entraîne la rotation du convoyeur. À l'inverse, lorsque l'interrupteur SW5 est ouvert, la LED s'éteint, ce qui traduit l'arrêt des moteurs et, par conséquent, l'arrêt du convoyeur.

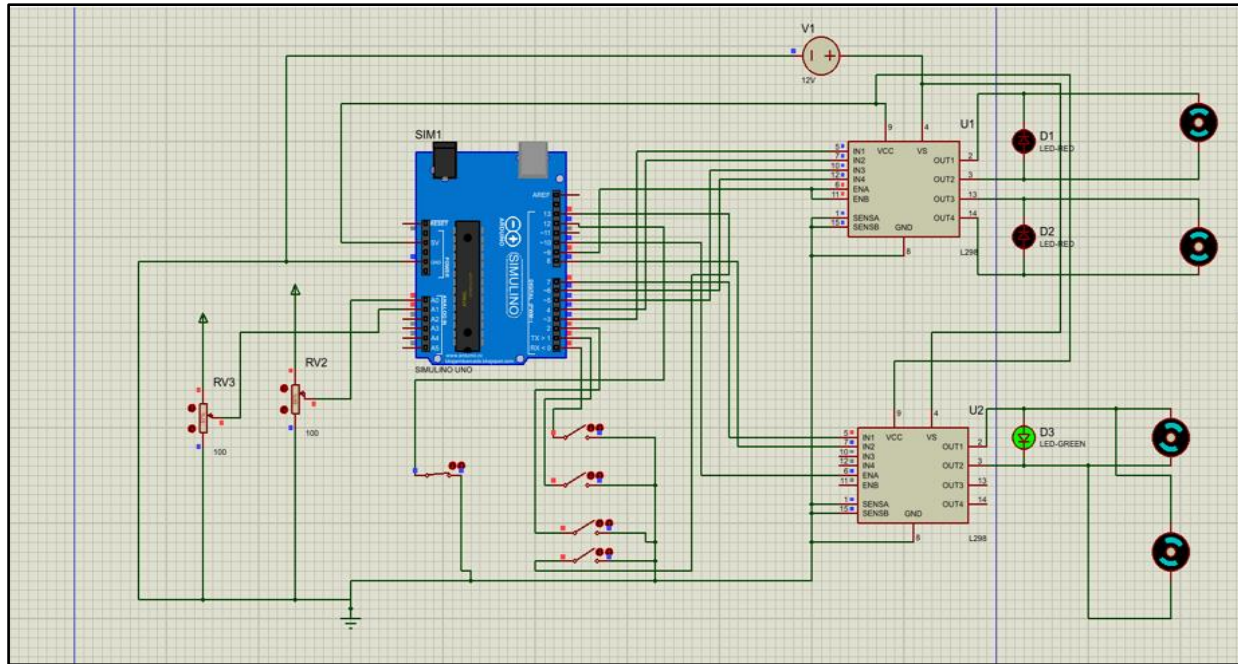


Figure III.10. Schéma de commande des moteurs M3 et M4 du convoyeur

### III. Partie réalisation

Une fois que la simulation est validée sous l'environnement ISIS de Proteus, nous allons entamer la phase de la réalisation pratique. Cette étape consiste à concrétiser le modèle simulé en un prototype fonctionnel, afin de vérifier le comportement du système en conditions réelles et d'évaluer ses performances opérationnelles.

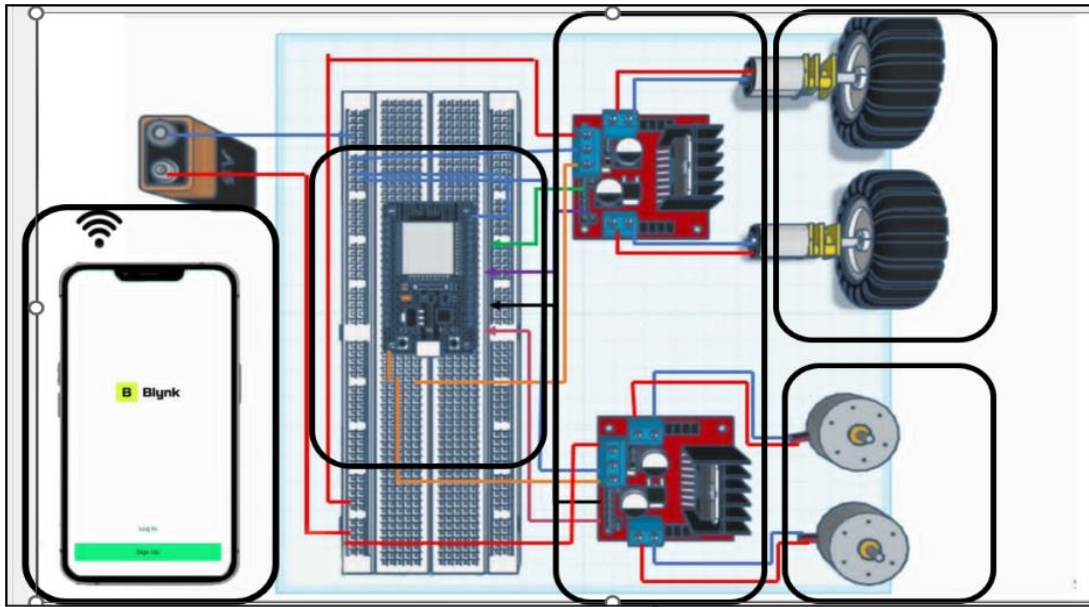


**Figure III.11.** Schéma de câblage et d'interconnexion du système[36]

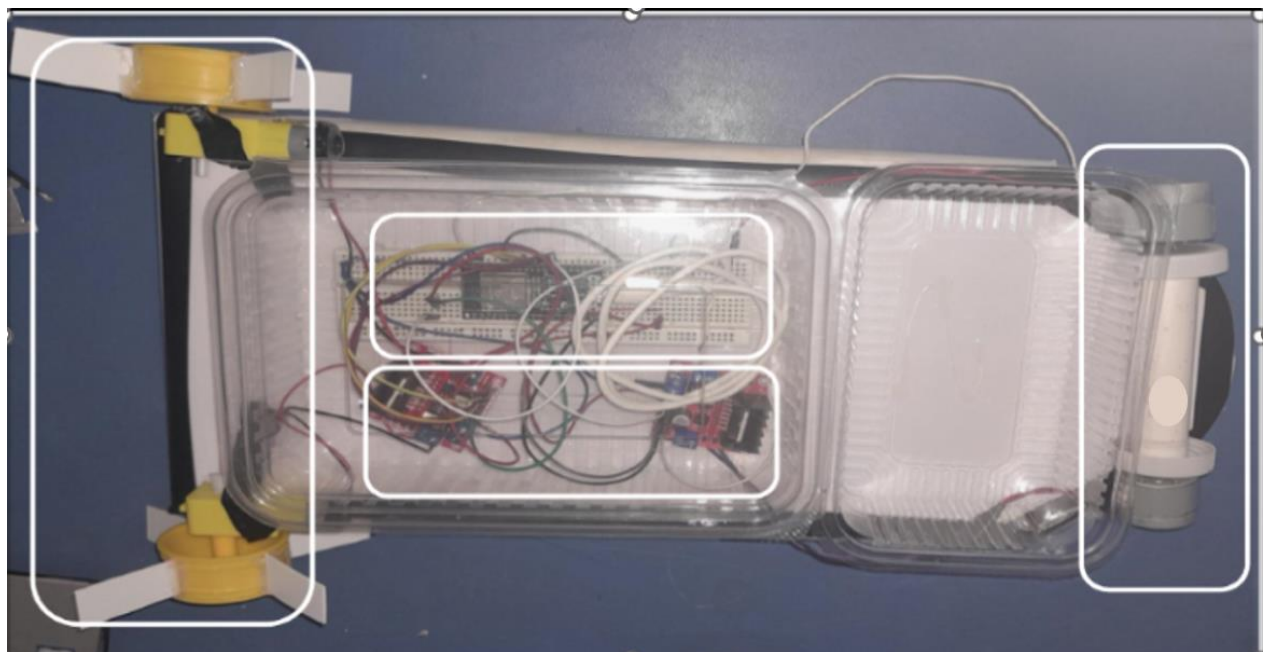


**Figure III.12.** Prototype final

### III.1. Principe de fonctionnement

Notre prototype repose sur une organisation en plusieurs blocs fonctionnels qui coopèrent pour assurer à la fois le nettoyage et le déplacement du système flottant. La partie de contrôle et de communication est assurée par l'application Android Blynk, qui permet à l'utilisateur d'envoyer des commandes et de superviser le fonctionnement en temps réel.



**Figure III.13.** Interface de commande du système via l'application Blynk

La communication entre le robot et le smartphone s'effectue sans fil grâce au module Wi-Fi intégré dans l'ESP32.

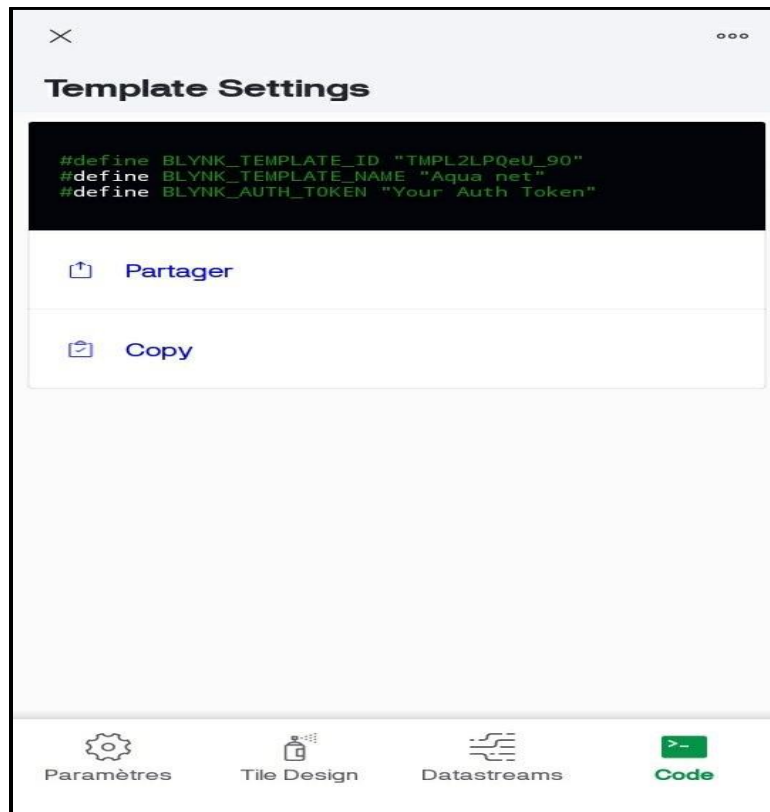


**Figure III.14.** Paramètres de configuration Blynk pour la liaison avec l'ESP32

L'application Blynk, installée sur le smartphone, permet de contrôler à distance les mouvements du robot ainsi que le fonctionnement du convoyeur.

Dans notre projet, nommé «*Aqua net*», l'interface Blynk propose un tableau de bord avec plusieurs commandes : quatre interrupteurs pour diriger le robot (avance, recule, gauche, droite), un curseur pour régler la vitesse, ainsi qu'un interrupteur et un curseur dédiés au convoyeur.

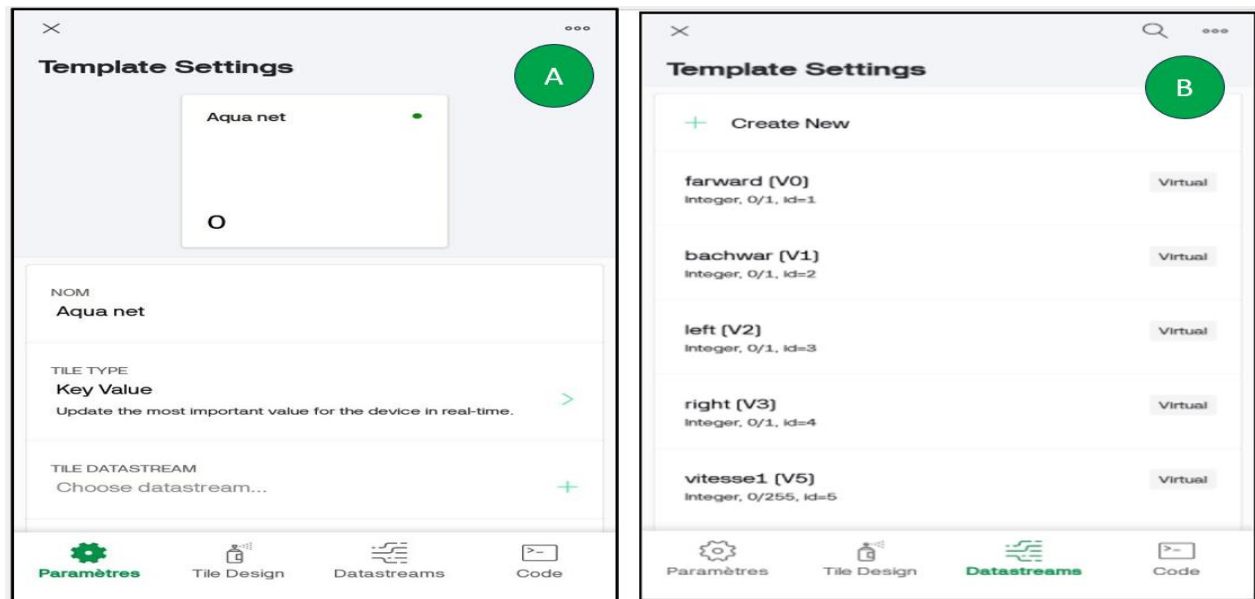


**Figure III.15.** Paramétrage de l'interface Blynk AQUA-NET

A. Paramètres du template Blynk    B. Création des datastreams de commande

En d'autres termes, lorsque l'utilisateur interagit avec l'interface Blynk (en appuyant sur un bouton ou en déplaçant un curseur), l'application envoie une requête vers le serveur Blynk. Ce serveur agit comme intermédiaire et transmet ensuite l'information au microcontrôleur ESP32 connecté au même réseau.

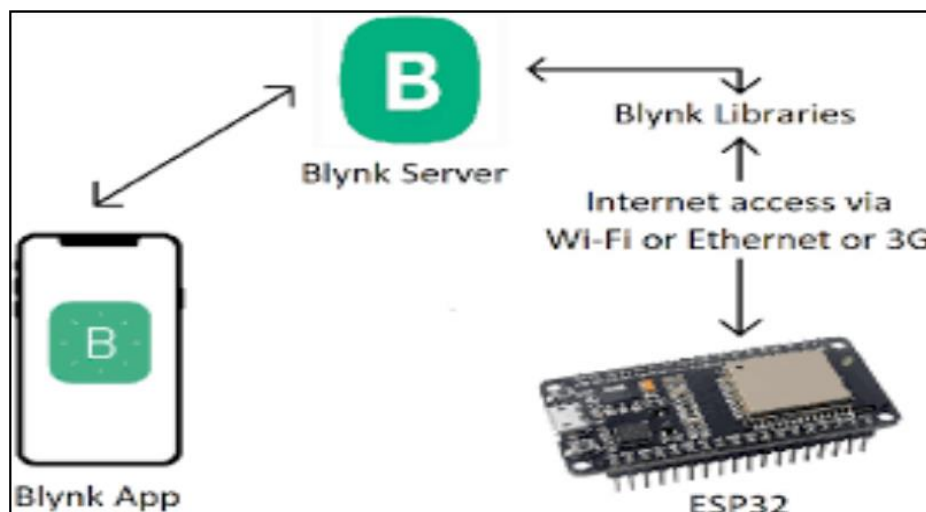


**Figure III.16.** Architecture de communication entre l'ESP32 et l'application Blynk [37]

Sur l'ESP32, chaque commande est associée à un Virtual Pin (V0, V1, V2, V3, V5, V6, V7). Ces broches virtuelles reçoivent les données envoyées par l'application et les traduisent en instructions binaires (0 pour arrêt, 1 pour activation) ou en signaux PWM pour la variation progressive de la vitesse.

Ces instructions sont ensuite traitées par le microcontrôleur ESP32, qui analyse les données reçues et les traduit en signaux exploitables. Ces signaux sont ensuite transmis aux drivers qui jouent le rôle d'amplificateurs de puissance et permettent d'alimenter correctement les moteurs.

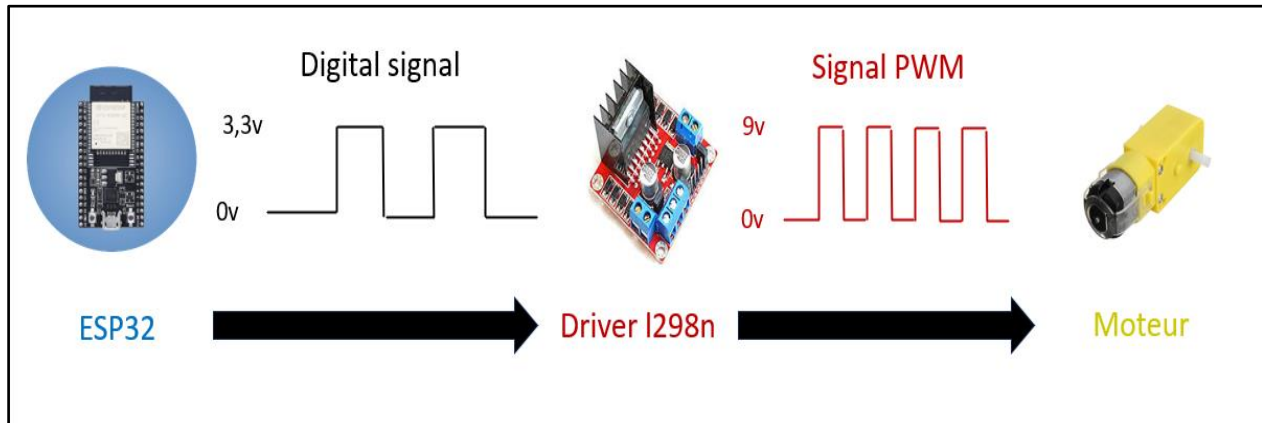


**Figure III.17.** Commande d'un moteur DC via driver L298N

Le composant principal de cette partie est le driver L298N. En effet, les signaux numériques envoyés par l'ESP32 sont limités à une tension logique de 3,3 V et un courant faible, alors que les moteurs nécessitent une alimentation plus importante (entre 100 mA et 300 mA, avec une tension de 9 V).

Le driver permet donc d'adapter et de fournir l'énergie nécessaire au fonctionnement des moteurs.

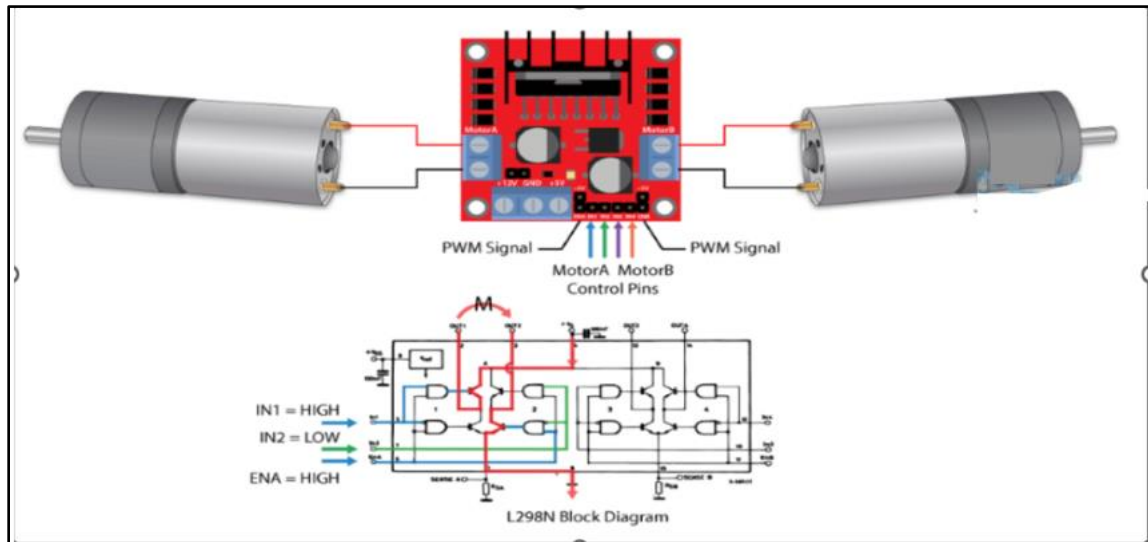


**Figure III.18.** Architecture de contrôle des deux moteurs utilisant le driver L298N[38]

Pour le contrôle, le L298N dispose de broches de commande : IN1 et IN2 déterminent le sens de rotation du moteur A, tandis que IN3 et IN4 assurent la même fonction pour le moteur B.

**Tableau III.3.** États des entrées INT1 et INT2 pour le contrôle du moteur A

| Int1 | Int2 | Sens de rotation du moteur A |
|------|------|------------------------------|
| 1    | 0    | Sens direct                  |
| 0    | 1    | Sens inverse                 |
| 0    | 0    | Arrêt                        |

Les broches **ENA** et **ENB** sont dédiées à la régulation de la vitesse de chaque moteur. Lorsqu'elles sont activées, elles reçoivent un signal **PWM** généré par l'ESP32.

Ce signal correspond à une variation de la tension moyenne appliquée au moteur, ce qui permet d'ajuster progressivement sa vitesse.

**Tableau III.4.** Commandes des moteurs de déplacement du nettoyeur

| <b>La commande</b> | <b>Int1</b> | <b>Int2</b> | <b>Int3</b> | <b>Int4</b> | <b>Sens de rotation</b>                       | <b>Mouvement du robot</b>       |
|--------------------|-------------|-------------|-------------|-------------|-----------------------------------------------|---------------------------------|
| <b>Forward</b>     | <b>1</b>    | <b>0</b>    | <b>1</b>    | <b>0</b>    | <b>Les deux moteurs tournent vers l'avant</b> | <b>Le robot avance</b>          |
| <b>Backward</b>    | <b>0</b>    | <b>1</b>    | <b>0</b>    | <b>1</b>    | <b>Les deux moteurs tournent en arrière</b>   | <b>Le robot recule</b>          |
| <b>Left</b>        | <b>0</b>    | <b>0</b>    | <b>1</b>    | <b>0</b>    | <b>Moteur B avancé, moteur A s'arrête</b>     | <b>Le robot tourne à gauche</b> |
| <b>Right</b>       | <b>1</b>    | <b>0</b>    | <b>0</b>    | <b>0</b>    | <b>Moteur A avancé, moteur B s'arrête</b>     | <b>Le robot tourne à droite</b> |
| <b>Stop</b>        | <b>0</b>    | <b>0</b>    | <b>0</b>    | <b>0</b>    | <b>Les moteurs sont arrêtés</b>               | <b>Le robot s'arrête</b>        |

Ainsi, le système peut contrôler à la fois la direction et la vitesse des moteurs de manière précise et efficace.

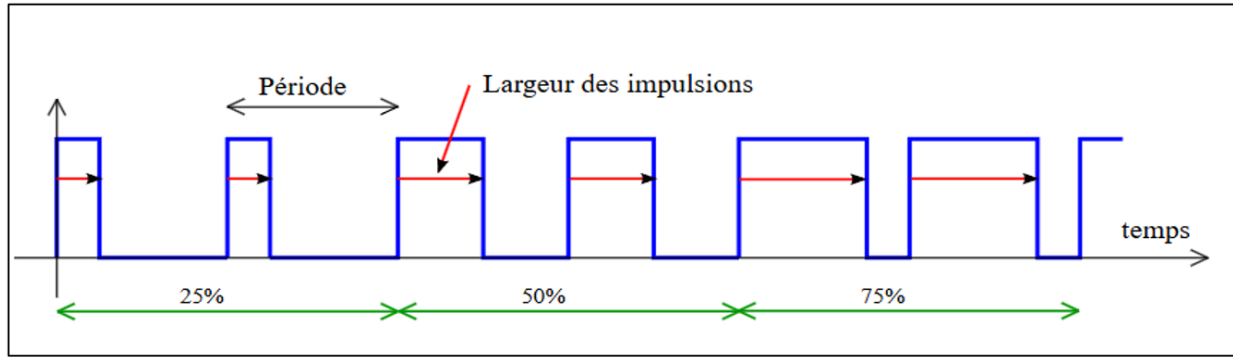


**Figure III.19.** Variation de la tension et la vitesse appliqué au moteur DC par signal PWM.[39]

Les moteurs se répartissent en deux catégories : ceux placés à l'avant, activent le convoyeur et assurent le nettoyage, en collectant les déchets présentes à la surface de l'eau, et ceux situés à l'arrière, responsables du déplacement et de la manœuvrabilité du système en exécutant les commandes de direction transmises par l'utilisateur. Ainsi, lorsque les moteurs avant sont activés, le convoyeur se met en rotation pour collecter les impuretés, tandis que les moteurs arrière assurent la propulsion et permettent au dispositif de se diriger selon les commandes reçues.

Cette organisation permet au dispositif de réaliser simultanément deux tâches complémentaires : avancer ou se déplacer sur l'eau tout en maintenant le processus de nettoyage actif.

Grâce à cette coordination, le prototype optimise son efficacité en couvrant une zone plus large en un temps réduit, tout en garantissant un fonctionnement continu et fluide. Enfin, l'ensemble de ces blocs est alimenté par la batterie, qui constitue la source principale d'énergie. Elle fournit une alimentation continue et stable, garantissant l'autonomie et la fiabilité du prototype lors de son fonctionnement en conditions réelles. Grâce à cette organisation, chaque bloc joue un rôle complémentaire : la commande est transmise par l'utilisateur, traitée par l'ESP32, amplifiée par les drivers, exécutée par les moteurs, et soutenue par l'énergie de la batterie, ce qui assure une cohérence globale et un fonctionnement efficace du système.

### III.2. Validation des résultats

Lorsque nous appuyons sur le bouton virtuel « Forward » dans l'application Blynk Aqua Net installée sur le smartphone, nous constatons que le nettoyeur se déplace vers l'avant, comme le montre la figure III.20 ci-dessous.



**Figure III.20** Déplacement du nettoyeur vers l'avant via l'application Blynk

Après l'activation du bouton virtuel « Backward » dans l'application Blynk Aqua Net installée sur le smartphone, le nettoyeur se déplace vers l'arrière, comme l'illustre la figure III.21.



**Figure III.21** Déplacement du nettoyeur vers l'arrière via l'application Blynk

Après avoir positionné le nettoyeur à proximité des déchets flottants, les moteurs de déplacement sont arrêtés, puis le convoyeur est activé en appuyant sur le bouton virtuel « Convoyeur » de l'application Blynk afin d'assurer leur ramassage efficace, comme le montre la figure III.22.

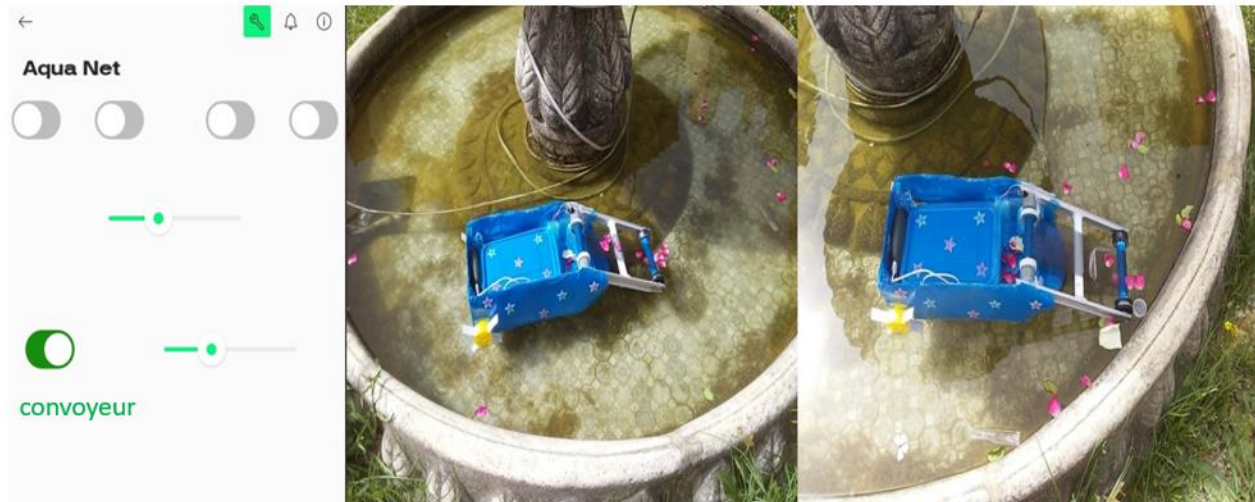


**Figure III.22** Activation du système de convoyeur pour la collecte des déchets flottants

## IV. Conclusion

Dans ce chapitre, nous avons présenté en détail le concept de notre système de nettoyeur flottant. Nous avons débuté notre travail par la présentation du schéma bloc du système, étape qui a permis de définir clairement l'organisation fonctionnelle des différents composants. Cette phase a été suivie par une simulation du système sous l'environnement Proteus ISIS.

Cette étape a permis de valider le schéma de principe, de vérifier la cohérence des signaux et d'anticiper le comportement global du système. Une fois la simulation confirmée, le prototype a été construit en suivant une organisation en blocs complémentaires : la commande transmise par l'application Blynk, le traitement assuré par l'ESP32, l'amplification de puissance réalisée par les drivers L298N, l'action des moteurs avant et arrière pour le nettoyage et le déplacement.

Enfin, plusieurs essais pratiques ont été réalisés afin d'évaluer les performances du prototype. Ces tests ont confirmé la fiabilité du système et ont permis d'obtenir des résultats satisfaisants, validant ainsi le bon fonctionnement global du projet.

# **Conclusion générale et perspectives**

### Conclusion générale et perspectives

Actuellement, le développement technologique a contribué à l'intégration de la robotique, de la programmation et de l'électronique afin de proposer des solutions innovantes aux problèmes environnementaux.

En effet, ce projet présenté une solution technologique innovante consiste à la résolution du problème de pollution de l'eau. Dans ce cadre, nous avons réaliser un robot capable de collecter les déchets flottants à la surface de l'eau.

La conception et la concrétisation de notre projet nécessite une coordination entre plusieurs étages matériels et logiciels.

D'abord nous avons étudié les différents composants électroniques nécessaires pour la réalisation de notre circuit tels que la carte Esp32, le driver l298N et les moteurs DC. Ensuite nous avons procédé à la simulation du système sous PROTEUS. Après cela, nous avons abordé la phase pratique pour la réalisation du prototype.

Ce dernier est basé sur une architecture composée de plusieurs étages fonctionnels. Tout d'abord, l'étage de commande, où la carte l'ESP 32 reçoit les commandes transmises par l'application mobile Blynk.

En effet, la création de l'application mobile 'Aqua Net' nous a permis de contrôler les différents mouvements du robot.

Ensuite vient l'étage de traitement qui traduit ces instructions en signaux exploitables. Le troisième étage est celui de puissance, dans lequel l'énergie nécessaire est régulée et adaptée aux besoins du système. Finalement, l'étage d'action vient finaliser le cycle en mettant les moteurs en mouvement, ce qui permet au robot d'avancer, de reculer ou de changer le sens de rotation afin de se diriger vers les déchets et de les collecter efficacement.

Enfin, les différents tests réalisés ont démontré que le prototype fonctionne de manière fiable et qu'il répond efficacement aux commandes transmises depuis l'application Android.

Enfin, plusieurs améliorations peuvent être envisagées, telles que l'intégration de capteurs intelligents pour que le système soit plus performant et autonome.

### Résumé

Nous avons étudié, conçu et fabriqué un nettoyeur flottant destiné à collecter les déchets flottants à la surface des plans d'eau, en particulier les déchets plastiques qui provoquent une grande pollution de l'environnement. Le système vise la réduction de cette pollution grâce à une solution simple et pratique basée sur le contrôle à distance.

Le nettoyeur est commandé par la technologie Wi-Fi en utilisant l'application Blynk installée sur le téléphone ou l'ordinateur. Les instructions sont envoyées à la carte de contrôle **ESP32**, qui reçoit les ordres, les traite puis les convertit en signaux permettant de contrôler le mouvement de quatre moteurs, responsables de déplacer et de diriger le nettoyeur dans l'eau pour collecter les déchets flottants. Ce projet aide à présenter une idée initiale où des tests ont été effectués montrant l'efficacité et la rapidité du système, qui peut être développé à l'avenir pour améliorer son efficacité.

**Mots clés :** Nettoyeur flottant - Collecte des déchets flottants - Wi-Fi – Blynk - Commandé à distance - Pollution de l'eau - ESP32.

---

### Abstract

We have studied, designed and manufactured a **floating cleaner** intended to collect floating waste on the surface of water bodies, especially plastic waste which causes significant environmental pollution. The system aims to contribute to the reduction of this pollution through a simple and practical solution based on **remote control**.

The system is controlled by **Wi-Fi technology** using the **Blynk** application installed on the Smartphone or computer. The commands are sent to the **ESP32** control board, which receives the orders, processes them and then converts them into signals that control the movement of four motors responsible for moving and directing the cleaner in the water to collect the floating waste.

This project helps to present an initial idea where tests have been carried out showing the efficiency and speed of the system, which can be further developed in the future to improve its effectiveness.

**Keywords:** Floating cleaner - Floating waste collection - Wi-Fi - Blynk - Remote control - Water pollution - ESP32.

---

### ملخص

قمنا في هذا المشروع بدراسة وتصميم وصنع منظم عائم مخصص لجمع النفايات الطافية على سطح المسطحات المائية، وخاصة النفايات البلاستيكية التي تسبب تلوثاً كبيراً للبيئة. يهدف هذا النظام إلى المساهمة في الحد من هذا التلوث من خلال حل بسيط وعملي يعتمد على التحكم عن بعد.

يتم التحكم في المنظم عبر تقنية الواي فاي باستخدام تطبيق Blynk، المثبت على الهاتف أو الحاسوب، حيث تُرسل الأوامر إلى لوحة التحكم ESP32. تقوم هذه اللوحة باستقبال الأوامر ومعالجتها ثم تحويلها إلى إشارات تتحكم في حركة أربع محركات مسؤولة عن تحريك المنظم وتوجيهه داخل الماء لجمع النفايات.

يساعد هذا المشروع على تقديم فكرة أولية حيث تم إجراء اختبارات تظهر فعالية وسرعة النظام يمكن تطويره مستقبلاً لتحسين فعاليته.

الكلمات المفتاحية : منظم عائم – جمع النفايات الطافية – شبكة الواي فاي – تطبيق Blynk – التحكم عن بعد – تلوث المياه .  
ESP32 -

## Références bibliographiques

- [1] Programme des Nations Unies pour l'Environnement (PNUE). (2021). De la pollution à la solution : une évaluation mondiale des déchets marins et de la pollution plastique. Nairobi, Kenya : Programme des Nations Unies pour l'Environnement.
- [2] United Nations Environment Programme. (2019). Global Environment Outlook 6: Healthy Planet, Healthy People. Cambridge University Press
- [3] <https://www.camfil.com/fr-fr/infos/qualite-de-l-air/pollution-de-lair>
- [4] [https://fr.wikipedia.org/wiki/Pollution\\_de\\_l%27air](https://fr.wikipedia.org/wiki/Pollution_de_l%27air)
- [5] <https://www.projetecolo.com/comment-eviter-la-pollution-atmospherique-175.html>
- [6] [https://fr.wikipedia.org/wiki/Pollution\\_des\\_sols](https://fr.wikipedia.org/wiki/Pollution_des_sols)
- [7] <https://pixelfarandole.com/expose-sur-la-pollution/>
- [8] Pollution des sols et agrosystèmes M1/ KARA Karima HDR
- [9] <https://www.scribd.com/document/475401878/Pollution-Des-Sols>
- [10] <https://www.alcor-controles.fr/6-solutions-contre-la-pollution-des-sols/>
- [11] <https://www.chatellaillonplage.fr/actualites/une-operation-de-nettoyage-de-la-plage-soutenue-par-un-geste-pour-chatel>
- [12] United Nations Environment Programme. (2009). Marine Litter: A Global Challenge. Nairobi : United Nations Environment Programme
- [13] <https://www.coloriageetdessins.com/apprendre/qu-est-ce-que-la-pollution-de-l-eau-354>
- [14] [https://www.choix-de-vie.com/blog/leau-la-pollution-n3?srsId=AfmBOor5cAgpTuNXpc5\\_8gzd\\_7TUHG\\_xcyYFMQLPsaBsX9a5jGobgAEh](https://www.choix-de-vie.com/blog/leau-la-pollution-n3?srsId=AfmBOor5cAgpTuNXpc5_8gzd_7TUHG_xcyYFMQLPsaBsX9a5jGobgAEh)
- [15] <https://www.eaufrance.fr/les-impacts-de-la-pollution-de-leau>
- [16] <https://piscines-deluxe.fr/vers-rouges-piscine-dangereux/>
- [17] United Nations Environment Programme. (2019). Global Environment Outlook 6: Healthy Planet, Healthy People. Cambridge University Press.
- [18] <https://www.chasingcleaner.com/products/hydro-6-corded-pool-cleaner>
- [19] Springer Handbook of Ocean Engineering, Springer, 2016
- [20] <https://www.airzen.fr/le-jellyfishbot-un-robot-autonome-qui-depollue-les-eaux/>
- [21] Chang, H.C., et al. (2021). Autonomous Water Quality Monitoring and Water Surface Cleaning for Unmanned Surface Vehicle. Sensors.

- [22] <https://newsatsea.com/environment/navtek-delivers-first-semi-autonomous-unmanned-water-cleaning-drone>
- [23] Takawy, M. Y., et al. (2023). Design and Implementation of an Autonomous Water Surface Cleaning Robot
- [24] RanMarine Technology. WasteShark – Manual Control and Efficient Water Cleaning.
- [25] <https://www.espressif.com/en/products/devkits/esp32-devkitc>
- [26] [https://documentation.espressif.com/esp32\\_datasheet\\_en.pdf](https://documentation.espressif.com/esp32_datasheet_en.pdf)
- [27] <https://powerlab.dz/product/l298-module-driver-moteur/>
- [28] <https://arduinoakitproject.com/esp32-dc-motor-l298n-motor-driver-control-speed-direction/>
- [29] <https://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf>
- [30] <https://passionelectronique.fr/tutoriel-l298n/>
- [31] <https://www.f-legrand.fr/scidoc/docmml/sciphys/arduino/pontL298/pontL298.html>
- [32] <https://passionelectronique.fr/tutoriel-l298n/>
- [33] <https://www.dfrobot.com/product-100.html>
- [34] <https://store.arduino.cc/products/breadboard-830-contacts?queryID=undefined>
- [35] BRIXI NIGASSA Abdellatif Elias, (2023). Étude d'un système d'achat et de facturation automatique en utilisant le module RFID. Mémoire de Master, Université Abou Bekr Belkaid de Tlemcen, Faculté de Technologie, Département de Génie Électrique et Électronique (GEE), Spécialité : Instrumentation.
- [36] Autodesk. Tinkercad Circuits : plateforme de conception et de simulation des circuits électroniques : <https://www.tinkercad.com/>
- [37] [https://link.springer.com/chapter/10.1007/978-3-031-23582-5\\_9](https://link.springer.com/chapter/10.1007/978-3-031-23582-5_9)
- [38] [https://link.springer.com/chapter/10.1007/978-3-031-23582-5\\_9](https://link.springer.com/chapter/10.1007/978-3-031-23582-5_9)
- [39] [https://link.springer.com/chapter/10.1007/978-3-031-23582-5\\_9](https://link.springer.com/chapter/10.1007/978-3-031-23582-5_9)

# Annexes

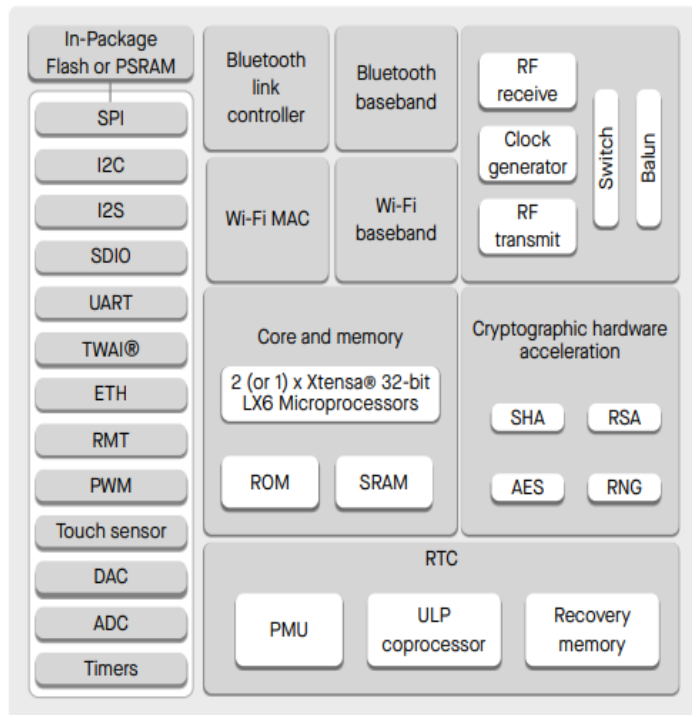
## Annexe 1: datasheet de l'ESP32

### Product Overview

ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC low-power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios.

For details on part numbers and ordering information, please refer to [Section 1 ESP32 Series Comparison](#). For details on chip revisions, please refer to [ESP32 Chip Revision v3.0 User Guide](#) and [ESP32 Series SoC Errata](#).

The functional block diagram of the SoC is shown below.



ESP32 Functional Block Diagram

## 2 Pins

### 2.1 Pin Layout

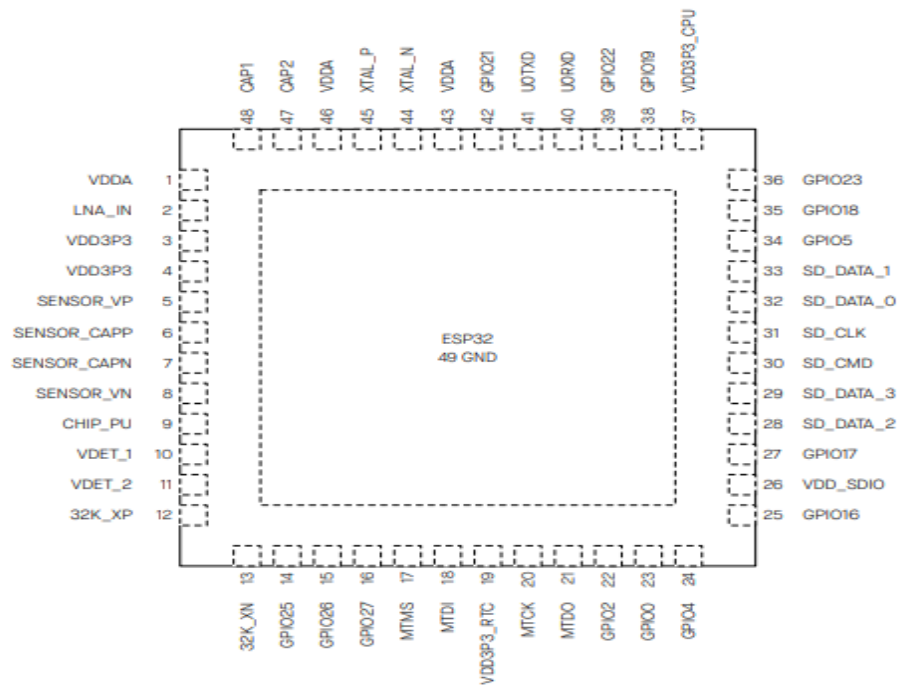


Figure 2-1. ESP32 Pin Layout (QFN 6\*6, Top View)

## 2.2 Pin Overview

Table 2-1. Pin Overview

| Name        | No. | Type | Function                                                                                                    |
|-------------|-----|------|-------------------------------------------------------------------------------------------------------------|
| Analog      |     |      |                                                                                                             |
| VDDA        | 1   | P    | Analog power supply (2.3 V ~ 3.6 V)                                                                         |
| LNA_IN      | 2   | I/O  | RF input and output                                                                                         |
| VDD3P3      | 3   | P    | Analog power supply (2.3 V ~ 3.6 V)                                                                         |
| VDD3P3      | 4   | P    | Analog power supply (2.3 V ~ 3.6 V)                                                                         |
| VDD3P3_RTC  |     |      |                                                                                                             |
| SENSOR_VP   | 5   | I    | GPIO36, ADC1_CH0, RTC_GPIO0                                                                                 |
| SENSOR_CAPP | 6   | I    | GPIO37, ADC1_CH1, RTC_GPIO1                                                                                 |
| SENSOR_CAPN | 7   | I    | GPIO38, ADC1_CH2, RTC_GPIO2                                                                                 |
| SENSOR_VN   | 8   | I    | GPIO39, ADC1_CH3, RTC_GPIO3                                                                                 |
| CHIP_PU     | 9   | I    | High: On; enables the chip<br>Low: Off; the chip shuts down<br>Note: Do not leave the CHIP_PU pin floating. |
| VDET_1      | 10  | I    | GPIO34, ADC1_CH6, RTC_GPIO4                                                                                 |
| VDET_2      | 11  | I    | GPIO35, ADC1_CH7, RTC_GPIO5                                                                                 |
| 32K_XP      | 12  | I/O  | GPIO32, ADC1_CH4, RTC_GPIO9, TOUCH9, 32K_XP (32.768 kHz crystal oscillator input)                           |
| 32K_XN      | 13  | I/O  | GPIO33, ADC1_CH5, RTC_GPIO8, TOUCH8, 32K_XN (32.768 kHz crystal oscillator output)                          |
| GPIO25      | 14  | I/O  | GPIO25, ADC2_CH8, RTC_GPIO6, DAC_1, EMAC_RXD0                                                               |
| GPIO26      | 15  | I/O  | GPIO26, ADC2_CH9, RTC_GPIO7, DAC_2, EMAC_RXD1                                                               |
| GPIO27      | 16  | I/O  | GPIO27, ADC2_CH7, RTC_GPIO17, TOUCH7, EMAC_RX_DV                                                            |
| MTMS        | 17  | I/O  | GPIO14, ADC2_CH6, RTC_GPIO16, TOUCH6, EMAC_TXD2, HSPICLK, HS2_CLK, SD_CLK, MTMS                             |
| MTDI        | 18  | I/O  | GPIO12, ADC2_CH5, RTC_GPIO15, TOUCH5, EMAC_TXD3, HSPIQ, HS2_DATA2, SD_DATA2, MTDI                           |
| VDD3P3_RTC  | 19  | P    | Input power supply for RTC IO (2.3 V ~ 3.6 V)                                                               |
| MTCK        | 20  | I/O  | GPIO13, ADC2_CH4, RTC_GPIO14, TOUCH4, EMAC_RX_ER, HSPID, HS2_DATA3, SD_DATA3, MTCK                          |
| MTDO        | 21  | I/O  | GPIO15, ADC2_CH3, RTC_GPIO13, TOUCH3, EMAC_RXD3, HSPICSO, HS2_CMD, SD_CMD, MTDO                             |

Espressif Systems

Submit Documentation Feedback

14

ESP32 Series Datasheet v5.2

2 Pins

| Name       | No. | Type | Function                                                                    |
|------------|-----|------|-----------------------------------------------------------------------------|
| GPIO2      | 22  | I/O  | GPIO2, ADC2_CH2, RTC_GPIO12, TOUCH2, HSPIW, HS2_DATA0, SD_DATA0             |
| GPIO0      | 23  | I/O  | GPIO0, ADC2_CH1, RTC_GPIO11, TOUCH1, EMAC_TX_CLK, CLK_OUT1                  |
| GPIO4      | 24  | I/O  | GPIO4, ADC2_CH0, RTC_GPIO10, TOUCH0, EMAC_TX_ER, HSPIH, HS2_DATA1, SD_DATA1 |
| VDD_SDIO   |     |      |                                                                             |
| GPIO16     | 25  | I/O  | GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT                                      |
| VDD_SDIO   | 26  | P    | Output power supply: 1.8 V or the same voltage as VDD3P3_RTC                |
| GPIO17     | 27  | I/O  | GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180                                  |
| SD_DATA_2  | 28  | I/O  | GPIO9, HS1_DATA2, U1RXD, SD_DATA2, SPIHD                                    |
| SD_DATA_3  | 29  | I/O  | GPIO10, HS1_DATA3, U1TXD, SD_DATA3, SPIWP                                   |
| SD_CMD     | 30  | I/O  | GPIO11, HS1_CMD, U1RTS, SD_CMD, SPICSO                                      |
| SD_CLK     | 31  | I/O  | GPIO6, HS1_CLK, U1CTS, SD_CLK, SPICLK                                       |
| SD_DATA_0  | 32  | I/O  | GPIO7, HS1_DATA0, U2RTS, SD_DATA0, SPIQ                                     |
| SD_DATA_1  | 33  | I/O  | GPIO8, HS1_DATA1, U2CTS, SD_DATA1, SPID                                     |
| VDD3P3_CPU |     |      |                                                                             |
| GPIO5      | 34  | I/O  | GPIO5, HS1_DATA6, VSPICSO, EMAC_RX_CLK                                      |
| GPIO18     | 35  | I/O  | GPIO18, HS1_DATA7, VSPICLK                                                  |
| GPIO23     | 36  | I/O  | GPIO23, HS1_STROBE, VSPID                                                   |
| VDD3P3_CPU | 37  | P    | Input power supply for CPU IO (1.8 V ~ 3.6 V)                               |
| GPIO19     | 38  | I/O  | GPIO19, UOCTS, VSPIQ, EMAC_TXD0                                             |
| GPIO22     | 39  | I/O  | GPIO22, UORTS, VSPIW, EMAC_TXD1                                             |
| UORXD      | 40  | I/O  | GPIO3, UORXD, CLK_OUT2                                                      |
| UOTXD      | 41  | I/O  | GPIO1, UOTXD, CLK_OUT3, EMAC_RXD2                                           |
| GPIO21     | 42  | I/O  | GPIO21, VSPIH, EMAC_TX_EN                                                   |
| Analog     |     |      |                                                                             |
| VDDA       | 43  | P    | Analog power supply (2.3 V ~ 3.6 V)                                         |
| XTAL_N     | 44  | O    | External crystal output                                                     |
| XTAL_P     | 45  | I    | External crystal input                                                      |
| VDDA       | 46  | P    | Analog power supply (2.3 V ~ 3.6 V)                                         |
| CAP2       | 47  | I    | Connects to a 3.3 nF (10%) capacitor and 20 kΩ resistor in parallel to CAP1 |

Espressif Systems

Submit Documentation Feedback

15

ESP32 Series Datasheet v5.2

2 Pins

| Name | No. | Type | Function                                       |
|------|-----|------|------------------------------------------------|
| CAP1 | 48  | I    | Connects to a 10 nF series capacitor to ground |
| GND  | 49  | P    | Ground                                         |

**Notes for Table 2-1 Pin Overview:**

## 1. Function names:

|             |                                                                                                                                                                                                   |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CLK_OUT...  | clock output                                                                                                                                                                                      |
| SPICLK      | SPI clock signal                                                                                                                                                                                  |
| HSPICLK     |                                                                                                                                                                                                   |
| VSPICLK     |                                                                                                                                                                                                   |
| HS..._CLK   | SDIO Master clock signal                                                                                                                                                                          |
| SD_CLK      | SDIO Slave clock signal                                                                                                                                                                           |
| EMAC_TX_CLK | EMAC clock signal                                                                                                                                                                                 |
| EMAC_RX_CLK |                                                                                                                                                                                                   |
| U..._RTS    | UART0/1/2 hardware flow control signals                                                                                                                                                           |
| U..._CTS    |                                                                                                                                                                                                   |
| U..._RXD    | UART0/1/2 receive/transmit signals                                                                                                                                                                |
| U..._TXD    |                                                                                                                                                                                                   |
| MTMS        | JTAG interface signals                                                                                                                                                                            |
| MTDI        |                                                                                                                                                                                                   |
| MTCK        |                                                                                                                                                                                                   |
| MTDO        |                                                                                                                                                                                                   |
| GPIO...     | General-purpose input/output with signals routed via the GPIO matrix. For more details on the GPIO matrix, see <a href="#">ESP32 Technical Reference Manual</a> > Chapter 10 MUX and GPIO Matrix. |

## 2. Regarding highlighted cells, see Section 2.3.1 Restrictions for GPIOs and RTC\_GPIOs.

3. For a quick reference guide to using the IO\_MUX, Ethernet MAC, and GPIO Matrix pins of ESP32, please refer to Appendix [ESP32 Pin Lists](#).