

Internet des Objets

TRAVAIL

Date d’échéance : Le 11 Mars 2024

**RAPPORT DE LA REALISATION DU PROJET DE TEST D’UN CABLE RJ45**

**Programme** : Internet des objets

**Cours** : PROG1354

**Niveau** : **1**ère année

Rédigé par :

**CHIMI KOTIEU Jospin**

Matricule : 2074777

Encadreur :

**M. Madrani YAHYA**

**Année académique : 2023 - 2024**

**Table des matières**

[**I.** **Introduction** 3](#_Toc158384698)

[**II.** **Matériels et logiciel utilisés** 3](#_Toc158384699)

[1. Matériels utilisés 3](#_Toc158384700)

[2. Logiciels utilisés 4](#_Toc158384701)

[3. Circuit et diagramme du circuit 4](#_Toc158384702)

[**III.** **Choix de conception** 5](#_Toc158384703)

[**IV.** **Défis rencontrés** 5](#_Toc158384704)

[**V.** **Conclusion** 6](#_Toc158384705)

# **Introduction**

Dans le but de pouvoir tester un câble afin de déterminer si c’est un RJ45 droit, croisé ou indéterminé, j’ai conçu un programme pour se faire.

Ce pendant mon programme se base sur la communication entre les fils et l’ordre de cette communication.

Cet ordre de communication est enregistré dans un tableau et si ce tableau sort étant déjà classé alors on a un câble RJ45 droit ; dans le cas contraire on test les dispositions avec la configuration physique du RJ45 croisé et si cela map, on a un RJ45 croisé dans le cas contrait on à un câble de type indéfini.

Le principe de communication est le suivant : je configure une extrémité de chaque fil en sorti et j’envoie un état 1 en suite je configure l’autre extrémité de chaque fil et je lis la valeur qu’elle a reçu si cette valeur est 1 alors il y’a communication dans le cas contraire il y’a pas communication.

# **Matériels et logiciel utilisés**

Pour la réalisation de ce projet, nous avons utilisé un certain nombre de matériels et de logiciel.

## Matériels utilisés

Il s’agit ici entre autres des composants physiques de notre projet :

* ESP32

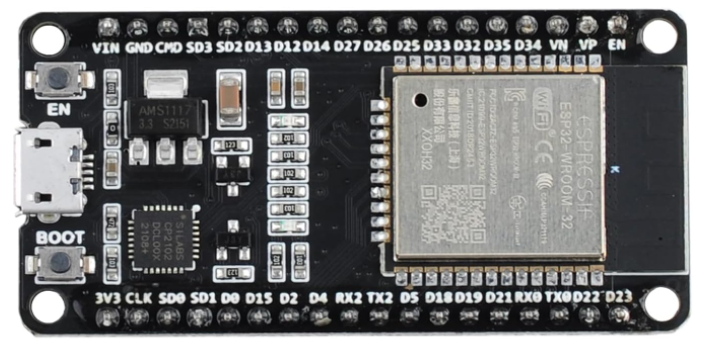
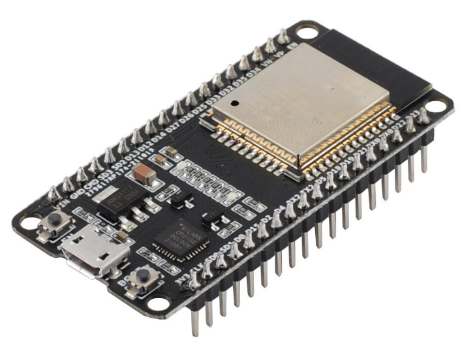


Figure 1 : ESP32

Figure 2 : Ecran LCD 16X2 I2CFigure 3 : Arduino MKR1000

Figure 4 : Ecran LCD 16X2 I2C

Figure 5 : Ecran LCD 16X2 I2C

Figure 6 : Ecran LCD 16X2 I2C

Figure 7 : Ecran LCD 16X2 I2C

Figure 8 : Capteur de température LM35Figure 9 : Ecran LCD 16X2 I2CFigure 10 : Arduino MKR1000

Figure 11 : Ecran LCD 16X2 I2CFigure 12 : Arduino MKR1000

* Les files de branchement

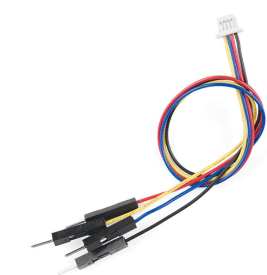


Figure 2 : fils de branchement

Figure 13 : fils de branchement

Figure 14 : fils de branchement

Figure 15 : fils de branchement

Figure 16 : Plaquette de prototypageFigure 17 : fils de branchement

Figure 18 : fils de branchement

Figure 19 : fils de branchement

Figure 20 : fils de branchement

Figure 21 : fils de branchement

Figure 22 : fils de branchement

Figure 23 : fils de branchement

Figure 24 : fils de branchement

Figure 25 : Plaquette de prototypageFigure 26 : fils de branchement

Figure 27 : fils de branchement

Figure 28 : fils de branchement

Figure 29 : fils de branchement

## Logiciels utilisés

Il s’agit ici des logiciels et plateformes qui nous au permis de monter et circuit de notre projet et de la codé : Visual studio code (avec l’extension Plateforme IO)

# **Choix de conception**

Pour réaliser ce projet en orienté objet, j’ai identifié 2 entités :

* **Fil** : qui a les attributs suivants :
  + **LeftPin** : qui représente le pin qui fait référence à l’extrémité gauche du fil)
  + **RightPin** : qui représente le pin qui fait référence à l’extrémité gauche du fil.
* **Cable**: qui a les attributs suivants :
* **Fils** : qui représente un pointeur qui va pointer sur les 8 fils que contient le câble
* **Type** : qui représente le type du câble (droit, croisé, ou indéfini)

Ci-dessous se présente le diagramme de classe :

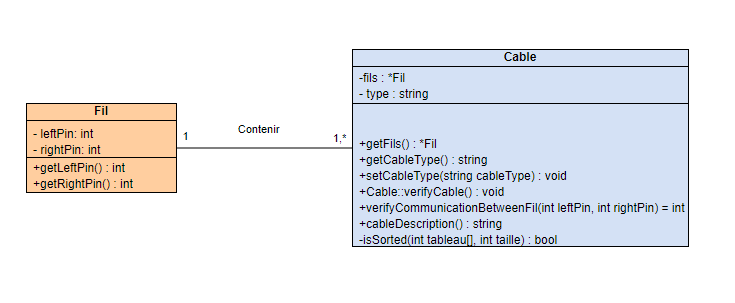


Figure 3 : Diagramme de classe du projet

# **Défis rencontrés**

Durant ce projet j’ai rencontré des problèmes dont le plus difficile et celui dont je n’ai pas trouvé de solution est le suivant :

* Rendre le programme autonome pour qu’il puisse fonctionner sans qu’on ne doit relancé le programme chaque fois.

# **Conclusion**

Parvenu à la fin de ce projet, j’ai beaucoup appris notamment :

* La compréhension des entré flotantes qui m’a causé beaucoup de tord durant le projet,
* Le fonctionnement du microcontrôleur esp32
* Les bonnes pratiques de la programmation orienté objet en C++