

## Conception et réalisation d'un laboratoire téléopérable intégrant des palettes de processus dynamiques accessible via une plateforme mobile

### [ Design and implementation of a teleoperable laboratory integrating dynamic process palettes accessible via a mobil platform ]

*Mendjemen Yamedjeu Bertrand, Paune Felix, and Kom Charles Hubert*

Laboratoire de Génie Informatique et Automatique, Université de Douala, Cameroon

---

Copyright © 2022 ISSR Journals. This is an open access article distributed under the *Creative Commons Attribution License*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**ABSTRACT:** In this work, we propose a low-cost teleoperable laboratory architecture, in which we integrate a palette of dynamic processes accessible via a mobile platform. In the current health context, our goal is to promote learning in engineering in developing countries via remote laboratories with a low-cost architecture. In this architecture accessible via wifi on a mobile platform, we use an ESP32 microcontroller as master, an ESP-CAM for visual feedback on our processes and several ESP32 microcontrollers as slave depending on the number of processes on our palette. We have chosen the ARDUINO NANO microcontroller to manage each process of the pallet. To test our architecture, we have integrated a process of visualization of the curves of variation of the current and the voltage of the brushless direct current motor according to the speed level. Using experimental methods, the error variations between the values measured with the tachometer and those found by calculation are between 0.00094 and 0.0350. Thus, the PWM setpoint delivered by our architecture is indeed equivalent to the output speed.

**KEYWORDS:** Remote laboratories, ESP32, dynamic processes, ESP NOW, ARDUINO NANO, Experimental methods.

**RESUME:** Dans ce travail, nous proposons une architecture de laboratoire téléopérable à faible coût, dans laquelle nous intégrons une palette de processus dynamiques accessible via une plateforme mobile. Dans le contexte sanitaire actuel, notre but est de favoriser les apprentissages en ingénierie dans les pays en voie de développement via des laboratoires distants avec une architecture à faible coût. Dans cette architecture accessible en wifi sur plateforme mobile, nous utilisons un microcontrôleur ESP32 comme maître, un ESP-CAM pour le retour visuel sur nos processus et plusieurs microcontrôleurs ESP32 comme esclave selon le nombre de processus de notre palette. Nous avons choisi le microcontrôleur ARDUINO NANO pour gestion de chaque processus de la palette. Pour éprouver notre architecture, nous avons intégré un processus de visualisation des courbes de variation du courant et de la tension du moteur à courant continu brushless en fonction du niveau de vitesse. A l'aide des méthodes expérimentales, les variations d'erreurs entre les valeurs mesurées avec le tachymètre et celles retrouvées par calcul, sont comprises entre 0,00094 et 0,0350. Ainsi, la consigne PWM délivré par notre architecture est bien équivalente à la vitesse en sortie.

**MOTS-CLEFS:** Laboratoires distants, ESP32, processus dynamiques, ESP NOW, ARDUINO NANO, Méthodes expérimentales.

## 1 INTRODUCTION

De nos jours l'utilisation des laboratoires distants ou laboratoire téléopérable en ingénierie, est devenu une norme dans la pédagogie [1], [2]. Nous précisons que le concept de laboratoire distant se divise en deux catégories à savoir le laboratoire

distant réel virtuel et le laboratoire distant réel [3]. Comme exemple de laboratoires distants réel, nous avons le WebLab-DEUSTO de l'université de Deusto, qui est une plateforme qui a favorisé le déploiement plusieurs processus téléopérable distinct [2], [4], [5]. Nous avons également les laboratoires réels proposés par [6], [7] dans le domaine de l'ingénierie industrielle. La pluralité de ces laboratoires distant à amener le laboratoire REMLABNET à concevoir un système de gestion de plusieurs laboratoires distants dont l'architecture repose sur la communication plusieurs serveurs (serveurs de base de données, serveur de mesure, serveur [8], [9]. Dans la même logique, LABNET a également proposé une architecture qui permettrait d'unifier l'accès à plusieurs processus distants réels via un serveur. Au regard de ces différents travaux effectués, nous avons remarqué que tous ces laboratoires qui fédèrent les processus, sollicitent des matériels coûteux et consommant une grande quantité d'énergie électrique. En outre, à l'heure actuelle la pandémie Coronavirus (COVID-19) a obligé les universités à vulgariser les cours et les travaux pratiques en ligne [10], mais malheureusement cette pandémie a affaibli les économies des pays en voie de développement, plus précisément les pays africains [11], [12].

Dans le but de favoriser les apprentissages dans un laboratoire distant dans un contexte économique difficile afin d'offrir une alternative aux institutions éducationnelles des pays en voie de développement, nous proposons une architecture à faible coût dans laquelle nous intégrons une palette de processus dynamiques dans un laboratoire télé opérable. Dans la suite de notre document, nous présentons les outils et les technologies sollicités dans notre travail, ensuite nous procédons à la description de notre architecture suivie de la présentation des résultats obtenus. Nous clôturons notre document par une discussion et des perspectives sur les résultats obtenus.

## 2 MATERIELS ET METHODES

Nous présentons la conception de notre architecture, la modélisation de notre processus ainsi que les méthodes expérimentales mise en application.

### 2.1 SYSTEME PROPOSE

Le travail proposé ici, est inspiré des travaux REMLABNET, REMLABNET II, REMLABNET III. Dans l'ensemble, il s'agit de laboratoire télé-opérable permettant à des classes virtuelles d'effectuer des expériences depuis leur PC via des pages web (par exemple REMLABNET traite l'expérience à distance INCLINE pour l'enseignement de la mécanique et la formation des compétences pour l'expérimentation à distance et la construction d'expériences à distance [13]), l'expérience est construite sur le principe client-serveur et machines a états finis [13].

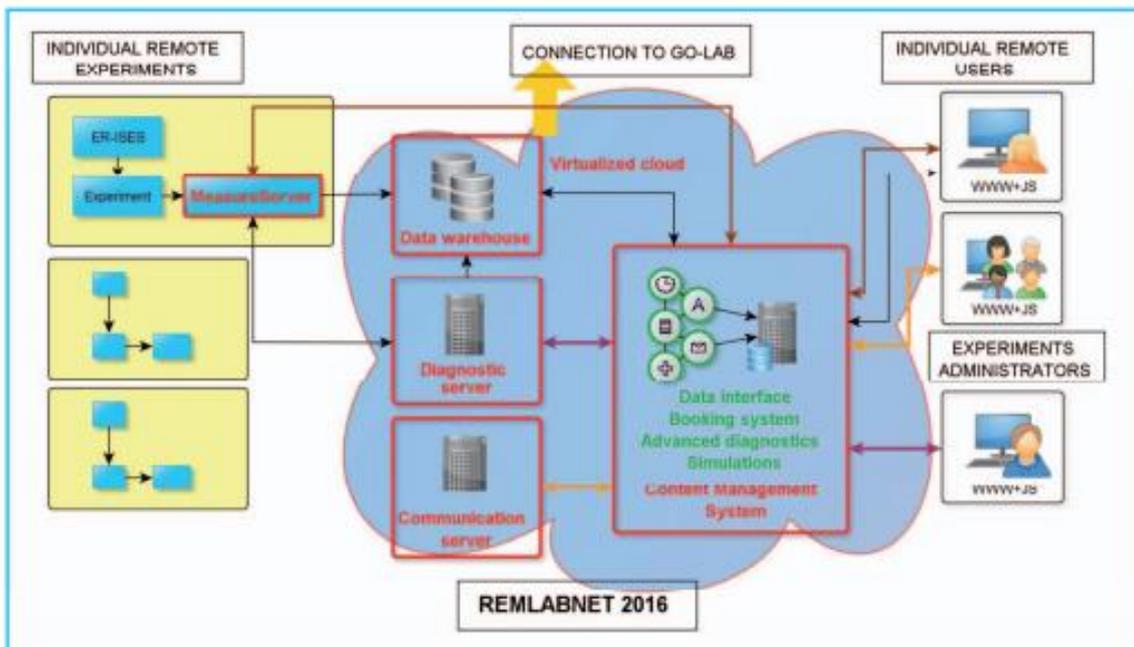


Fig. 1. Fonction mode 1 du REMLABNET INCLINE [13]

Sur cette image nous remarquons la présence de trois serveurs et une base de données, en tenant compte uniquement de ces détails, nous estimons déjà une très grande consommation en ressources énergétiques et en coûts financiers. Dans ce travail, nous allons proposer un laboratoire télé-opérable en local permettant aux étudiants au sein d'un campus universitaire de contrôler des processus. Pour pallier à la grande consommation de ressources citées plus haut, nous proposons un système centré sur l'ESP 32 qui coûte environ 17 \$USD, avec une très faible consommation en énergie électrique. La communication sera basée sur le protocole ESP-NOW, qui est propre à l'ESP 32, mais nous utiliserons un microcontrôleur Arduino Nano pour la gestion des processus et l'ESP CAM pour le retour d'image depuis le laboratoire.

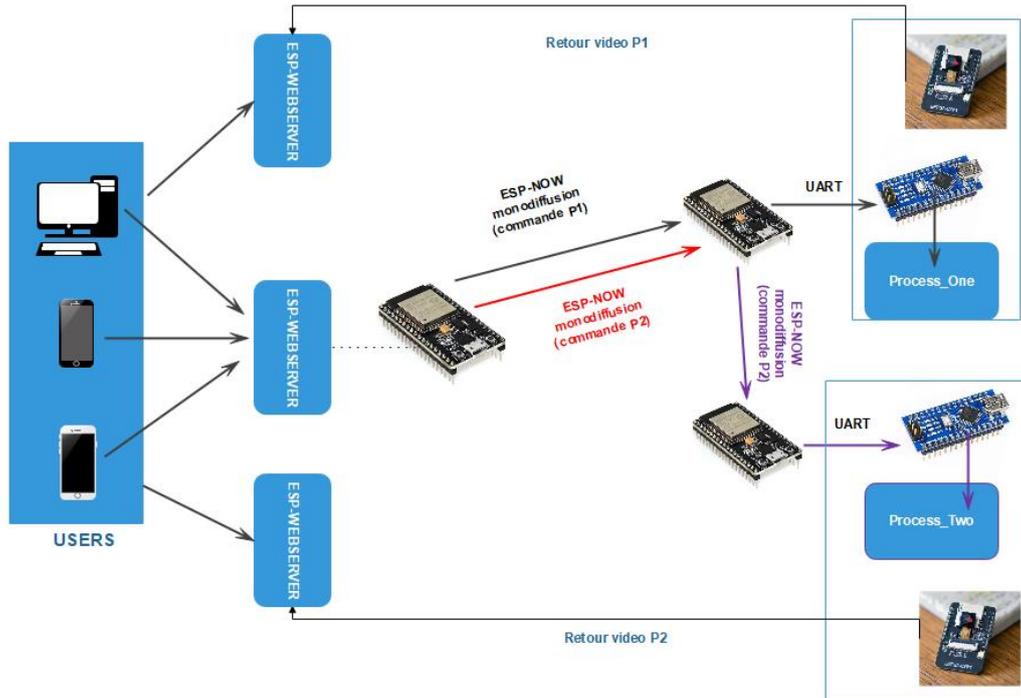


Fig. 2. Architecture du système proposé

## 2.2 CONCEPTION DE L'ARCHITECTURE

La conception du système est assez simple pour la chaîne allant de l'ESP web serveur vers les cartes ARDUINO responsable de la gestion des processus et les retours vidéo ESP-CAM. L'élément qui influence le choix du matériel est fortement basé sur le prix et la disponibilité dans les magasins locaux.

### 2.2.1 MODELISATION DU PROCESSUS

Etape importante car, il faudrait choisir un processus qui va permettre à l'étudiant d'obtenir des résultats efficaces en temps réel pour des éventuels recherches. Nous avons choisi le processus suivant: **Visualisation des courbes de variation du courant et de la tension des moteurs à courant continu en fonction du niveau de vitesse: cas d'un moteur brushless**. Le moteur «brushless», moteur sans balais, tire son nom par opposition au moteur à courant continu qui dispose de bobinages sur le rotor et nécessite un collecteur pour alimenter ces bobinages. Le moteur Brushless entre dans la catégorie des machines alternatives synchrones. Dans la figure 3, nous vous proposons un schéma de commande d'un moteur BRUSHLESS.

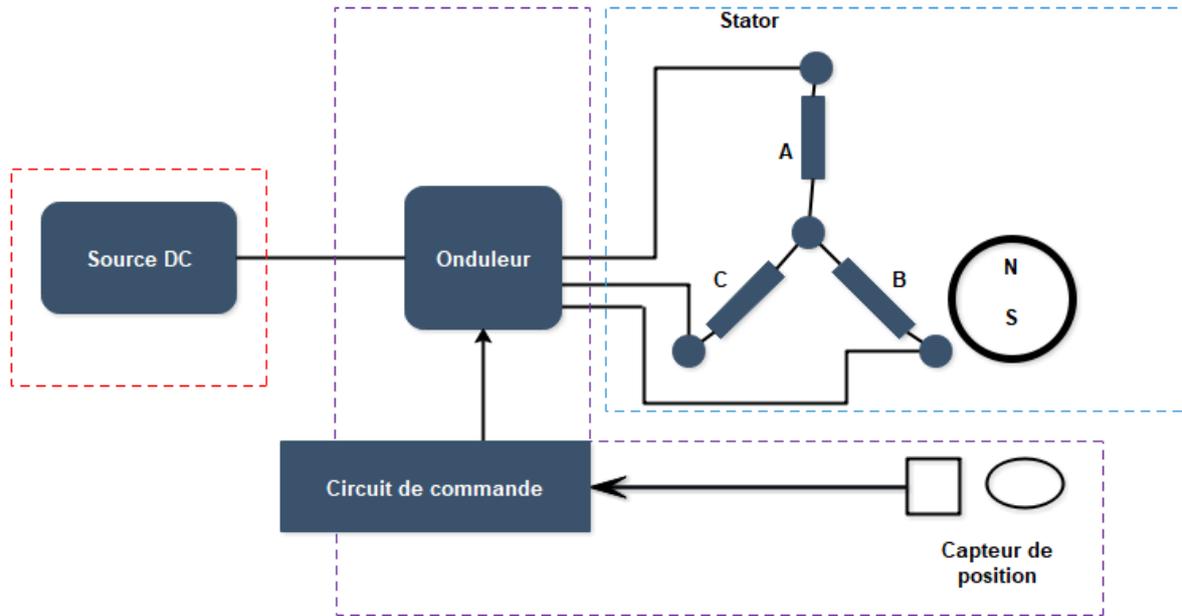


Fig. 3. Schéma de commande d'un moteur BRUSHLESS

Le moteur brushless fonctionne à partir de trois sources de tensions variables, fournies par un onduleur, et permettant de générer un champ magnétique tournant. Le rotor, généralement équipé d'un aimant permanent, tend à suivre le champ magnétique tournant. Les équations électriques régissant le fonctionnement d'un moteur BRUSHLESS sont les suivantes:

$$v_A(t) = R * i_A(t) + L * \frac{di_A(t)}{dt} + e_A(t) \quad (1)$$

$$v_B(t) = R * i_B(t) + L * \frac{di_B(t)}{dt} + e_B(t) \quad (2)$$

$$v_C(t) = R * i_C(t) + L * \frac{di_C(t)}{dt} + e_C(t) \quad (3)$$

Vu que le système est équilibré:

$$i_A(t) + i_B(t) + i_C(t) = 0$$

$$i_C(t) = -(i_A(t) + i_B(t)) \quad (4)$$

(4) dans (3) donne:

$$v_C(t) = -R * (i_A(t) + i_B(t)) - L * \frac{di_A(t)}{dt} - L * \frac{di_B(t)}{dt} + e_C(t) \quad (5)$$

(1), (2) et (5) impliquent:

$$v_{AB} = R * i_A(t) + L * \frac{di_A(t)}{dt} + e_A(t) - R * i_B(t) + L * \frac{di_B(t)}{dt} + e_B(t) \quad (6)$$

$$v_{BC}(t) = R * i_B(t) + L * \frac{di_B(t)}{dt} + e_B(t) + R * (i_A(t) + i_B(t)) + L * \frac{di_A(t)}{dt} + L * \frac{di_B(t)}{dt} - e_C(t) \quad (7)$$

$$v_{BC}(t) = 2 * R * i_B(t) + 2 * L * \frac{di_B(t)}{dt} + e_B(t) + R * (i_A(t)) + L * \frac{di_A(t)}{dt} - e_C(t) \quad (8)$$

(8) implique:

$$R * i_B(t) = \frac{1}{2} \left( -v_{BC}(t) + 2 * L * \frac{di_B(t)}{dt} + e_B(t) + R * (i_A(t)) + L * \frac{di_A(t)}{dt} - e_C(t) \right)$$

$$R * i_B(t) = \frac{1}{2} * v_{BC}(t) - L * \frac{di_B(t)}{dt} - \frac{1}{2} * e_B(t) - \frac{1}{2} * R * (i_A(t)) - \frac{1}{2} * L * \frac{di_A(t)}{dt} + \frac{1}{2} * e_C(t) \quad (9)$$

(9) dans (6) donne:

$$\frac{di_A(t)}{dt} = \frac{2}{3 * L} * v_{AB}(t) - \frac{R}{L} * i_A(t) - \frac{2}{3 * L} e_{AB}(t) + \frac{1}{3 * L} v_{BC}(t) - \frac{1}{3 * L} * e_{BC}(t) \quad (10)$$

(10) dans (8):

$$\frac{di_B(t)}{dt} = \frac{1}{3 * L} * v_{BC}(t) - \frac{R}{L} * i_B(t) - \frac{1}{3 * L} e_{BC}(t) - \frac{1}{3 * L} v_{AB}(t) + \frac{1}{3 * L} * e_{AB}(t) \quad (11)$$

Ainsi (10) et (11) illustrent le fonctionnement du moteur BRUSHLESS sur la commutation des phases A et B comme le montre la figure 4 suivante.

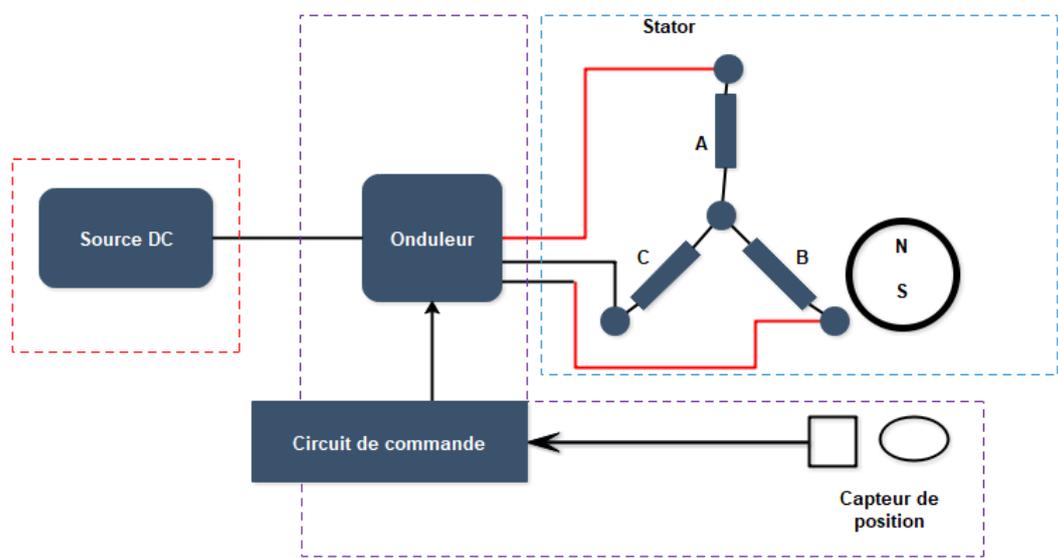


Fig. 4. Commutation sur les phases A et B d'après les formules (10) et (11)

## 2.3 MATERIELS ET TECHNOLOGIES

### 2.3.1 LES OUTILS MATERIELS

- **ESP 32**

ESP 32 est au cœur de l'architecture de notre travail. L'ESP 32 est un microcontrôleur bas prix, successeur de l'ESP8266. Conçu et produit en 2016 par la société Espressif située à Shanghai [14]. Cette technologie a prouvé sa capacité dans les solutions de travaux avec Wi-Fi autonome. L'ESP 32 propose une variété de fonctionnalités compatible avec énormément de capteurs pour une large gamme d'applications partant de l'IOT au streaming audio. En conséquence, sa spécification donne un arsenal extrême pour programmeur avec double cœur 240 MHz, 520 Ko de RAM et périphérique comprenant ADC, DAC, I2S, I2C, UART, SPI, GPIO et ne coûte que 10.000 FCFA. Les fonctions générales de l'ESP32 sont illustrées dans le diagramme bloc à la De plus, l'unité de gestion de l'alimentation et le coprocesseur à très faible puissance permettent à ESP32 de fonctionner à moins de 1mA en mode veille profonde. Cette caractéristique fait ESP32 un excellent choix pour les applications à faible puissance [14].

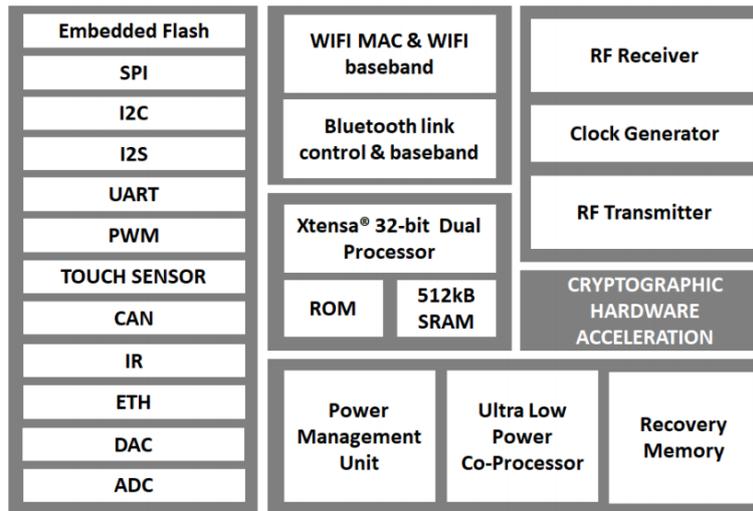


Fig. 5. Diagramme bloc ESP 32 [14]

- **ESP 32 CAM**

C'est une carte de développement à faible coût avec caméra Wi-Fi. Il permet de créer des projets de caméra IP pour le streaming vidéo avec différentes résolutions. Il prend en charge la caméra OV2640 et la caméra OV7640. La carte dispose d'un flash intégré, d'un processeur 32 bits à faible consommation, d'une vitesse d'horloge de 160 MHz, de plusieurs modes de veille et d'une mémoire SRAM externe intégrée. La carte ESP-32 CAM est largement utilisée dans les projets où la surveillance est nécessaire. Sa connectivité Wi-Fi avec microcontrôleur en tant que processeur offre un avantage dans la visualisation des données en direct sous forme d'image à partir d'une zone distante [14]. Le module ESP-32 CAM doit permettre de générer un serveur Web local à travers lequel les flux vidéo en direct peuvent être observés, mais uniquement dans le réseau local. Pour permettre l'accès à distance, nous pouvons utiliser ngrok, qui est une application multiplateforme qui nous permet d'exposer un réseau local à Internet. [15].

- **ARDUINO NANO**

Arduino Nano est une petite carte totalement conviviale basé sur l'architecture d'ATmega328P avec un ATmega328 microcontrôleur avec une tension de fonctionnement de 5V. Il utilise un flash type de mémoire 32 Ko dont 2 Ko sont utilisés par le chargeur de démarrage. La vitesse d'horloge du processeur d'Arduino Nano est de 16 MHz. Il contient 16 broches analogiques IN et 22 broches E/S, son PCB la taille est de 18x45 mm avec un poids de seulement 7g. [16]

### 2.3.2 LES TECHNOLOGIES (PROTOCOLES)

#### 2.3.2.1 ESP NOW

ESP-NOW est une technologie de communication rapide et sans connexion avec transmission de paquets courts. ESP-NOW est idéal pour les éclairages intelligents, les télécommandes, les capteurs et autres applications. ESP-NOW applique la technologie de trame IEEE802.11 Action Vendor, ainsi que la fonction IE développée par Espressif et la technologie de cryptage CCMP, créant ainsi une solution de communication sécurisée et sans connexion [17]. En 2018, un protocole P2P appelé ESP-NOW a été publié pour la première fois pour ESP32. Comme on peut le voir sur la figure 2, la structure globale du protocole ne nécessite aucun routeur ou état de jonction pour un réseau hiérarchique. Au lieu de cela, il permet un P2P ainsi qu'une communication basse consommation entre plusieurs appareils. En comparaison avec le Wi-Fi normal, c'est plus économe en énergie et plus rapide à déployer. À l'intérieur du réseau ESP-NOW, tous les appareils peuvent communiquer par 3 méthodes principales: diffusion, monodiffusion et multidiffusion avec un débit de données de 1 Mbps et plus [p3]. En monodiffusion et la multidiffusion nécessite un couplage initial, une réponse d'accusé de réception et un nombre limité de récepteurs jusqu'à 20, la diffusion est plus efficace pour les applications de conversation en rafale car il n'a pas besoin de jumeler avec d'autres et peut avoir un nombre illimité de receveur. Dans la figure 6, nous vous proposons le schéma de fonctionnement de l'ESP NOW.

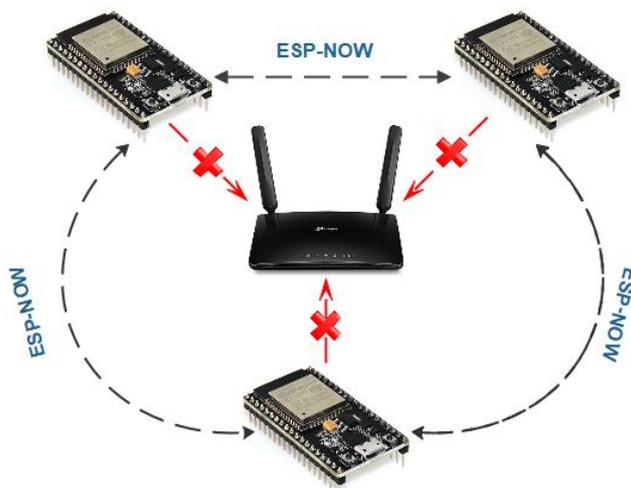


Fig. 6. Schéma de fonctionnement de la technologie ESP NOW

Dans ce projet, la monodiffusion est la principale méthode de la communication. Comme il n'y a pas de connexion avec le routeur Wi-Fi, l'en-tête d'ESP-NOW est extrêmement court. Comme on peut le voir sur le tableau 1, ESP-NOW utilise une trame d'action spécifique au fournisseur pour transmettre les données ESP-NOW [18].

Tableau 1. Format spécifique de trame

MAC HEADER	Category code	Organization Identifier	Random Values	Vendor FCS	Specific content
24 bytes	1 byte	3 bytes	4 bytes	7~255 bytes	4 bytes

ESP-NOW utilise la méthode CCMP, qui est décrite dans IEEE Std. 802.11-2012, pour protéger le cadre d'action spécifique au fournisseur. Le périphérique Wi-Fi conserve une clé principale (PMK) et plusieurs clés principales locales (LMK). Les longueurs de PMK et LMK sont de 16 octets [18]

### 2.3.2.2 UART

L'UART (Universal Asynchronous Receiver Transmitter ou Emetteur récepteur Asynchrone Universel en français) est un composant clé du sous-système de communication série d'un ordinateur. Dans la plupart des microcontrôleurs, UART est également intégré comme fonctionnalité (il est en fait un protocole de communication). L'UART prend des octets de données et transmet les bits individuels de manière séquentielle. A destination, un deuxième UART réassemble les bits en octets complets. UART est asynchrone donc un signal d'horloge n'est pas requis mais les données transmises à la structure se composent du début et de la fin d'un message. Les deux extrémités de la liaison doivent fonctionner avec le même débit en bauds (vitesse de communication). UART fonctionne normalement à des niveaux TTL (Transistor Transistor Logic) de 3,3 V à 5 V [19].

### 2.3.3 LES OUTILS LOGICIELS

La mise sur pieds de notre système nécessite un certain nombre de logiciels, il est important de noter que la principale interface de programmation est **Platformio**. Il s'agit d'un package de **Visual studio code** qui permet de programmer les microcontrôleurs. Il embarque de façon basique la bibliothèque **arduino.h**.

## 2.4 LES METHODES

### 2.4.1 DESCRIPTION DU SYSTEME PROPOSE

Il s'agit d'un processus dynamique qui va permettre aux étudiants en fonction de la commande du niveau de vitesse envoyé (depuis l'interface utilisateur) de visualiser en temps réel (via le retour vidéo) les courbes de variation du courant et de la tension du moteur à partir d'une plateforme mobile.

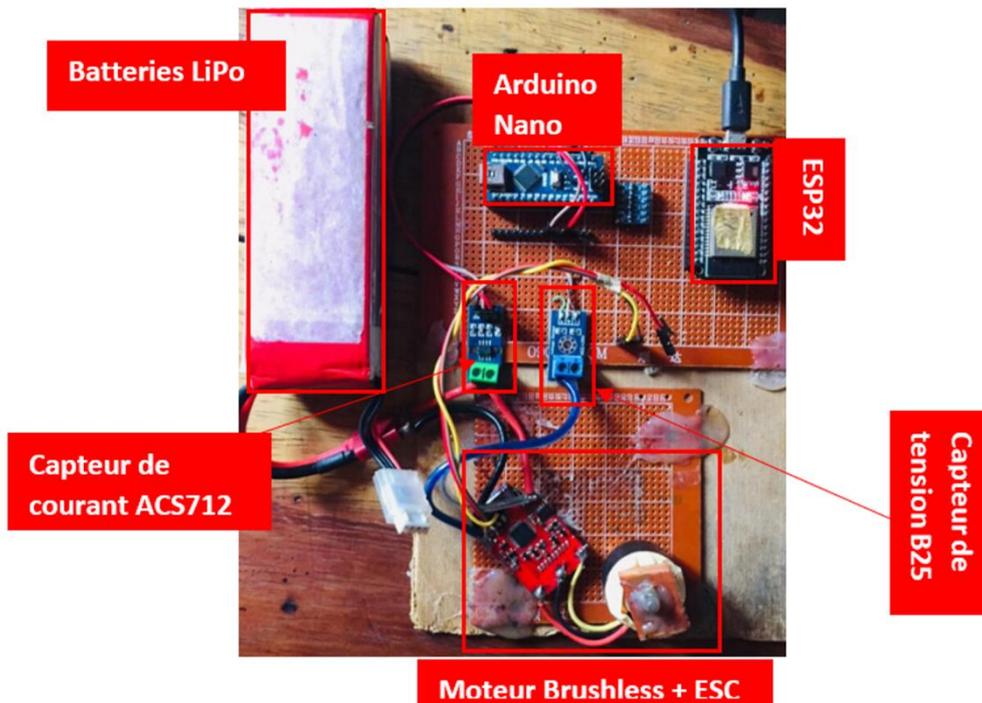


Fig. 7. Schéma électrique de notre système

#### 2.4.1.1 LA COMMANDE D'UN MOTEUR BRUSHLESS

La commande de notre moteur se fera via un microcontrôleur combiné à un ESC (Electronic Speed Control). ESC est un variateur de vitesse des moteurs à courant continu. Le module est compatible avec les moteurs Brushless de type BLDC. Le microcontrôleur génère un signal PWM reconnu par l'ESC qui commande le moteur. Ainsi pour faire tourner le moteur, la valeur du signal PWM généré sera proportionnelle à la valeur de la puissance de fonctionnement du moteur. Le but de cette commande par PWM est de faire varier la vitesse de rotation du moteur qui s'exprime en RPM (Revolutions Per Minuts). La bibliothèque arduino Servo.h avec sa fonction.writeMicroseconds () nous ont permis de facilement générer les PWM vers l'ESC tout en sachant que la plage de valeur va de 1000 à 2000 pour cette fonction.

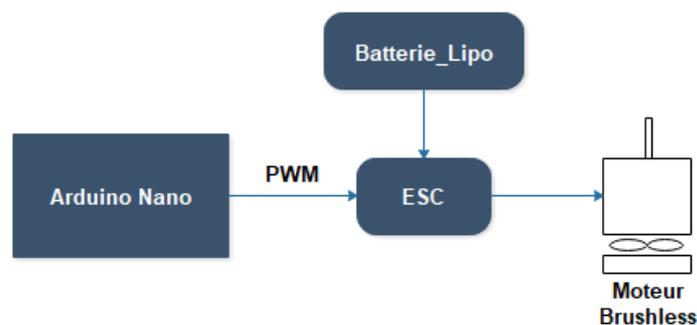


Fig. 8. Schéma de la commande avec PWM

### 2.4.1.2 LES CAPTEURS DE COURANT ET TENSION

Pour obtenir les valeurs de retour, nous avons utilisé Le capteur B25 qui nous permet de mesurer la tension du moteur, et le capteur de courant ACS712 qui nous a permis de mesurer le courant consommé par le moteur.

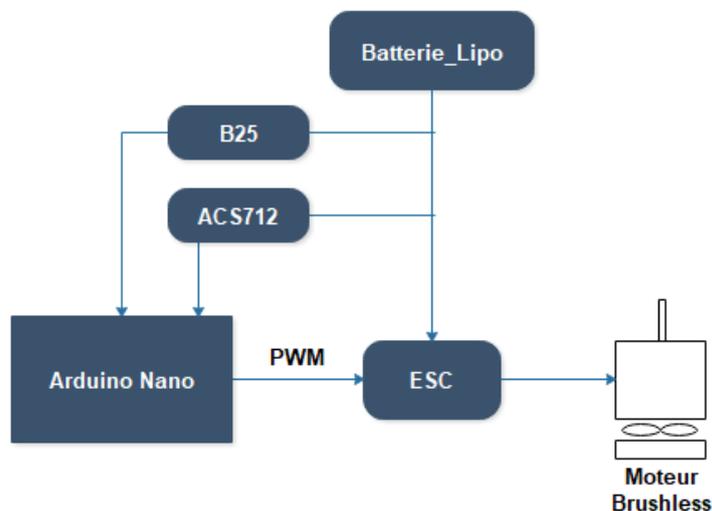


Fig. 9. Schéma de commande avec les capteurs de courant et tension de notre système

### 2.4.2 DESCRIPTION DE LA METHODE EXPERIMENTALE

Pour mesurer la vitesse de rotation du moteur en sortie et établir la proportion par rapport à la PWM générée, nous procéderons en deux étapes. La première se fera à l'aide d'un tachymètre numérique, la seconde par une méthode expérimentale utilisant une formule.

#### 2.4.2.1 ÉTAPE 1: UTILISATION DU TACHYMETRE NUMERIQUE

Le **tachymètre** est un instrument de mesure permettant de déterminer la vitesse de déplacement d'un objet en mouvement. Nous avons utilisé le modèle MULTIMETRIX RPM82.

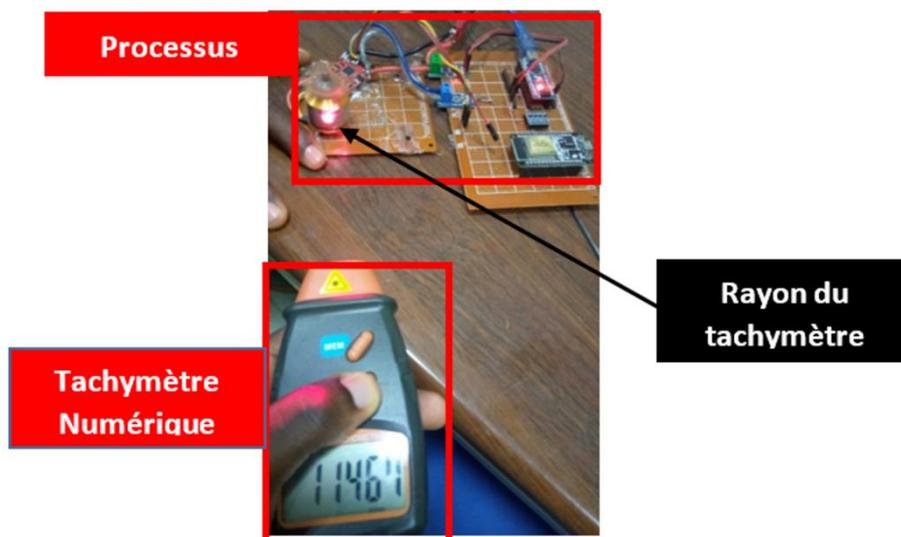


Fig. 10. Mesure de la vitesse grâce au tachymètre numérique

Tableau 2. Mesure tension, courant et vitesse (tachymètre) en fonction des valeurs de PWM

PWM (writeMicroseconds ())	Tension (V)	Courant (A)	Vitesse moteur mesuré avec le tachymètre (RPM)
1100	12.34	0.90	11 240
1200	12.12	1.04	11 942
1300	12.26	1.18	12 305
1400	12.26	1.22	12 592
1 500	12.59	1.18	12 624
1530	12.46	1.65	12 422
1650	12.56	1.26	12 622
1700	12.54	1.18	12 354
1800	12.46	1.85	12 690

#### 2.4.2.2 ÉTAPE 2: CAS EXPERIMENTAL (UTILISATION DE LA FORMULE)

Partant de la formule:

$$N = Kv * (U - I * Ri) \quad (1)$$

Où  $\left\{ \begin{array}{l} N \text{ est la vitesse de rotation en RPM} \\ Kv \text{ est la constante de vitesse du moteur} \\ I \text{ l'intensité de courant qui traverse le moteur} \\ U \text{ la tension d'alimentation du moteur} \\ Ri \text{ la résistance interne du moteur} \end{array} \right.$

Nous pouvons calculer la vitesse de rotation du moteur, ayant ces différentes informations. Le moteur que nous utilisons étant un A2212/13T de 1000KV, ces informations sont contenues dans [20]. En se référant au tableau 2, nous obtenons le tableau 3.

Tableau 3. Mesure de la vitesse (formule) et évaluation de l'erreur

PWM (writeMicroseconds ())	Tension (V)	Courant (A)	Vitesse moteur mesuré avec le tachymètre (RPM)	Vitesse moteur par calcul (RPM)	Ecart	Erreur en %
1100	12.34	0.90	11 240	11229.4	10.6	0.094
1200	12.12	1.04	11 942	11936.4	5.6	0.046
1300	12.26	1.18	12 305	12153.8	151.2	1.22
1400	12.26	1.22	12 592	12150.2	441.8	3.50
1 500	12.59	1.18	12 624	12 483.8	140.2	1.1
1650	12.56	1.26	12 622	12 444.08	177.92	1.41
1 700	12.54	1.18	12 354	12 433.8	-79.8	0.6
1800	12.46	1.85	12 690	12293.5	396.5	3.12

### 3 RESULTATS

#### 3.1 PRESENTATION DES INTERFACES

Les figure 11 et 12 nous présente l'interface par lequel l'utilisateur étant l'étudiant qui fait son TP, mais de manière distante, peut se connecter via un téléphone, une tablette ou un PC. Dans le cas présent l'étudiant se connecte sur un smartphone. Ce dernier doit se trouver dans le rayon d'action du wifi de l'ESP 32 serveur. Il se connecte au serveur via une adresse IP éditée dans l'URL de son navigateur que lui donne l'administrateur.

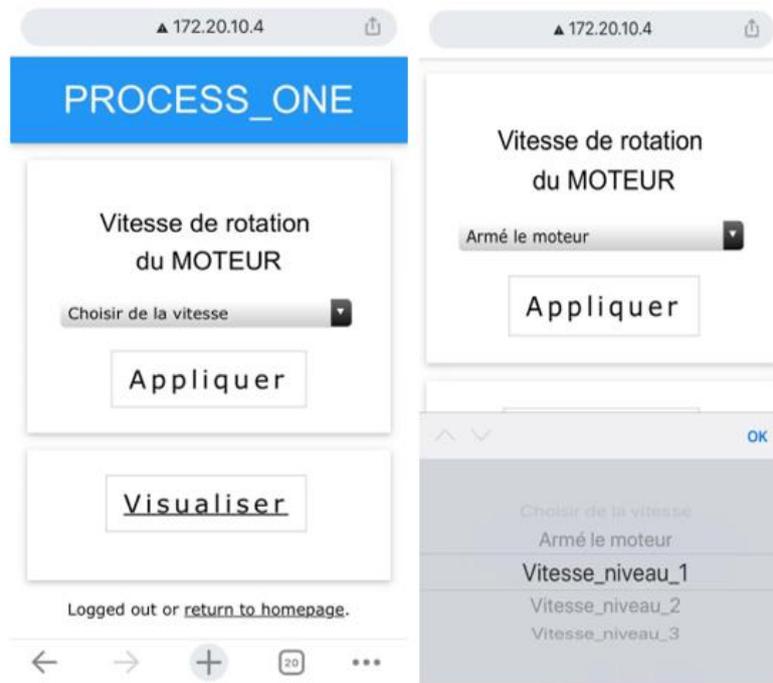


Fig. 11. L'interface de connexion

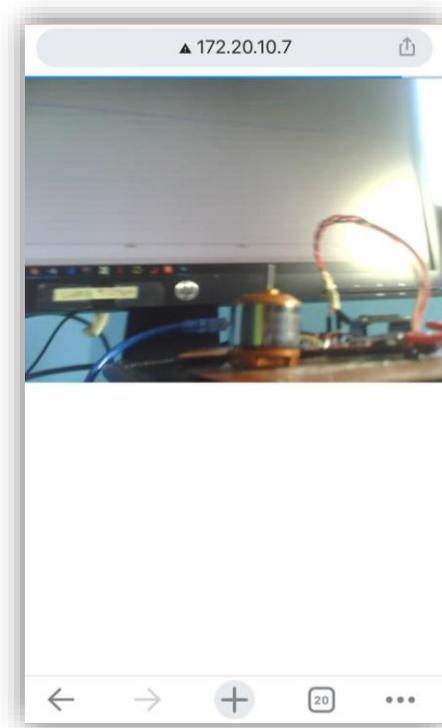


Fig. 12. Page d'accueil

L'étudiant a ainsi la possibilité de choisir un processus, ainsi après avoir sélectionné le premier processus nous obtenons une interface (figure 13) qui donne la possibilité de choisir une vitesse de rotation d'un moteur et de visualiser sa la courbe de variation du courant consommé (figure 14).



**Fig. 13.** Interface du processus 1



**Fig. 14.** Visualisation du processus en temps reel sur une interface (courbe de tension)

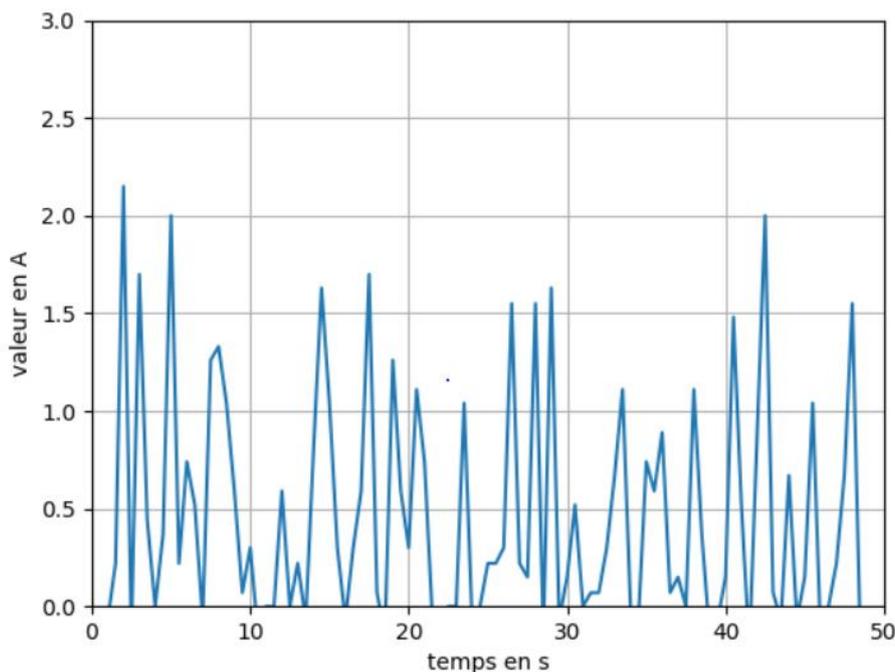


Fig. 15. Courbe de variation du courant en fonction de la vitesse

### 3.2 BILAN ENERGETIQUE

Tableau 4. Consommation énergétique

Composants	Voltage	Courant max
ESP 32	3,3v et 5v	1200 mA
Arduino Nano	3.3v et 5v	500mA (USB)
ESP 32 cam	3,3v et 5v	310mA

En additionnant les niveaux max de courant nous obtenons **2010mA** ce qui est un gain très important par rapport à l'utilisation d'un serveur.

## 4 DISCUSSIONS

On remarque que l'écart entre les valeurs mesurées respectivement avec le tachymètre et celle retrouvées par calcul à l'aide de la tension et de l'intensité du courant consommé par le moteur n'est pas très grand, d'où on retrouve de petites variations d'erreurs. N'ayant pas fait ces tests dans les conditions de laboratoire, ces erreurs peuvent être négligés et on peut conclure que la consigne de PWM délivrée par la ARDUINO nano est bien équivalente à la valeur de vitesse en sortie.

### 4.1 COMMENTAIRES

- Cette architecture que nous proposons est moins couteuse qu'une architecture à base de serveur.
- Cette architecture de laboratoire téléopérable donne la possibilité au pays en voie de développement de favoriser l'accès à distance à des équipements de laboratoire. le plus souvent ces équipements sont en nombre très réduits par rapport aux nombres d'apprenants.
- Notre architecture ne nous donne pas encore la possibilité de stocker les resultats de l'apprentissage pour favoriser la revision de l'apprenant.

## 5 CONCLUSION

Notre objectif dans cet article était de proposer une architecture de laboratoire téléopérable à faible coût, dans lequel serait intégré plusieurs processus téléopérable via une plateforme mobile. Nous avons testé la commande d'un moteur brushless pour évaluer l'erreur entre la consigne et les valeurs en sortie. Nous avons obtenu des erreurs de faible marge entre 0,00094 et 0,0350. L'exploitation de ces résultats nous amène à conclure qu'il est possible d'exploiter notre architecture pour favoriser l'apprentissage à distance, vu le contexte sanitaire, qui requiert la distance entre les étudiants.

## 6 REMERCIEMENTS

Mes remerciements vont à l'endroit de:

- **Dr PAUNE félix**, pour son encadrement et ses orientations dans nos travaux
- **Pr KOM Hubert**, pour ses conseils dans les techniques d'analyse de la documentation
- **MINKOULOU Ulrich**, pour son apport inestimable sur le plan technique et sa disponibilité.

## REFERENCES

- [1] M. Kalúz, L. Čirka, R. Valo, and M. Fikar, ArPi Lab: A low-cost remote laboratory for control education, vol. 19, no. 3. IFAC, 2014.
- [2] J. Garcia-Zubia et al., "Application and user perceptions of using the WebLab-Deusto-PLD in technical education," Proc. - Front. Educ. Conf. FIE, 2011, doi: 10.1109/FIE.2011.6143127.
- [3] Heinz-Dietrich Wuttke, Karsten Henke ET Nadine Ludwig International, CompSysTech'.Conference on Computer Systems and Technologies - Remote Labs versus Virtual Labs for Teaching Digital System Design", 2005.
- [4] P. Orduña, A. Gómez-Goiri, L. Rodríguez-Gil, J. Diego, D. López-de-Ipiña, and J. Garcia-Zubia, "wCloud: automatic generation of WebLab-Deusto deployments in the Cloud," 2015 12th Int. Conf. Remote Eng. Virtual Instrum., pp. 223–229, 2015.
- [5] J. García-Zubia et al., "An integrated solution for basics digital electronics: Boole-DEUSTO and WebLab-DEUSTO," 2013 10th Int. Conf. Remote Eng. Virtual Instrumentation, REV 2013, pp. 1–5, 2013, doi: 10.1109/REV.2013.6502890.
- [6] P. R. Pradyumna, C. K. S. Tarun, and S. Bhanot, "Remote experimentation of 'no-load tests on a transformer' in electrical engineering," AICERA 2012 - Annu. Int. Conf. Emerg. Res. Areas Innov. Pract. Futur. Trends, no. 1, 2012, doi: 10.1109/AICERA.2012.6306740.
- [7] G. A. Prieto and J. P. Mendoza, "Low cost didactic robotic platform based on Player/Stage software architecture and la fonera hardware," Rev. Iberoam. Technol. del Aprendiz., vol. 8, no. 3, pp. 126–132, 2013, doi: 10.1109/RITA.2013.2273112.
- [8] F. Schauer, M. Krbecek, P. Beno, M. Gerza, L. Palka, and P. Spilakova, "REMLABNET II-Open remote laboratory management system for university and secondary schools research based teaching," Proc. 2015 12th Int. Conf. Remote Eng. Virtual Instrumentation, REV 2015, no. February, pp. 109–112, 2015, doi: 10.1109/REV.2015.7087273.
- [9] F. Schauer et al., "REMLABNET III - Federated remote laboratory management system for university and secondary schools," Proc. 2016 13th Int. Conf. Remote Eng. Virtual Instrumentation, REV 2016, no. February, pp. 238–241, 2016, doi: 10.1109/REV.2016.7444471.
- [10] K. A. A. Gamage, D. I. Wijesuriya, S. Y. Ekanayake, A. E. W. Rennie, C. G. Lambert, and N. Gunawardhana, "Online delivery of teaching and laboratory practices: Continuity of university programmes during COVID-19 pandemic," Educ. Sci., vol. 10, no. 10, pp. 1–9, 2020, doi: 10.3390/educsci10100291.
- [11] S. A. Lone and A. Ahmad, "COVID-19 pandemic—an African perspective," Emerg. Microbes Infect., vol. 9, no. 1, pp. 1300–1308, 2020, doi: 10.1080/22221751.2020.1775132.
- [12] P. Ozili, "COVID-19 in Africa: socio-economic impact, policy response and opportunities," Int. J. Sociol. Soc. Policy, 2020, doi: 10.1108/IJSSP-05-2020-0171.
- [13] F. Schauer, M. Ožvoldová2L. Tkáp «REMLABNET - open remote laboratory management system for e-experiments'» 2015 3rd Experiment International Conference (exp.at'15).
- [14] A. Shukla and R. Diwan, "IOT Based load Automation with Remote Access Surveillance Using ESP 32 CAMand ESP 8266 Module," vol. 25, no. 3, pp. 6904–6914, 2021.
- [15] T. Ahmed, A. T. Bin Nuruddin, A. B. Latif, S. S. Arnob, et R. Rahman, « A Real-Time Controlled Closed Loop IoT Based Home Surveillance System for Android using Firebase", in 2020 6th International Conference on Control, Automation and Robotics (ICCAR), Singapore, Singapore, avr. 2020, p. 601-606. doi: 10.1109/ICCAR49639.2020.9108016.
- [16] "Arduino - Home," Arduino.cc, 2019. <https://www.arduino.cc>. (consulté le juin 27,2021).
- [17] « ESP-NOW User Guide - Recherche Google ».

[https://www.google.com/search?q=ESP-NOW+User+Guide&rlz=1C1RLNS\\_frCM809CM816&oq=ESP-NOW+User+Guide&aqs=chrome..69i57.1089j0j7&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=ESP-NOW+User+Guide&rlz=1C1RLNS_frCM809CM816&oq=ESP-NOW+User+Guide&aqs=chrome..69i57.1089j0j7&sourceid=chrome&ie=UTF-8) (consulté le juin 27, 2021).

[18] ESP-NOW — ESP- IDF Programming Guide v4.0-dev-11912019, pp.2-4.

[19] Design and development of daughter board for USB-UART communication between Raspberry Pi and PC, Manisha Sharma; Nidhi Agarwal; S R N Reddy, 15-16 May 2015.

[20] "A2212/13T 1000KV Moteur brushless – a2itronic." <https://a2itronic.ma/product/moteur-brushless-a2212-1000kv/> (accessed JUN. 11, 2021).