الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــة

Faculté : la faculté des sciences

Département : électronique

Domaine : Sciences et Technologie

Filière : Automatique

Spécialité : Automatique et informatique industrielle

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Iot industrial application with M5Stick

Présenté par : *MEZGHACHE Imad*

Encadreur : *Fezari Mohamed* *Professeur* *université badji mokhtar annaba*

Jury de Soutenance :

ARBAOUI Faycal	Pr	UBM Annaba	Président
FEZARI Mohamed	Pr	UBM Annaba	Encadreur
SAIDI Med Larbi	Pr	UBM Annaba	Examinateur

Année Universitaire : 2019/2020

Résumé :

Aujourd'hui, l'industrie joue un rôle essentiel dans notre société d'où toute une économie peut être établie sur elle. Là où le secteur est sensible on trouve des défaillances inexpliquées qui conduisent à une baisse de la production. Afin de permettre à l'industrie de grandir sans problème nous allons concevoir et réaliser un système à base d'objets connectés pour surveiller la production, ce système permettra de faire un contrôle non destructif préventif direct sur les appareils avec le moindre coût possible et sans l'aide d'un opérateur rendant la production efficace.

ملخص:

تلعب الصناعة اليوم دورًا أساسيًا في مجتمعنا حيث قد يكون الاقتصاد بأكمله هنا اين يكون القطاع حساسًا نجد هناك إخفاقات غير مبررة تؤدي إلى تدهور الإنتاج ، فللسماح للصناعة بالنمو بسلاسة سنقوم بتصميم وبناء نظام يعتمد على انترنت الأشياء لمراقبة الإنتاج ، سيسمح هذا النظام بإجراء اختبار وقائي مباشر غير مدمر على المعدات بأقل تكلفة ممكنة ودون مساعدة المشغل ، مما يجعل الإنتاج فعالاً.

Abstract:

Today, industry plays an essential role in our society from where an entire economy can be established on it, where the sector is sensitive, there are unjustified failures that lead to the deterioration of production, so to allow the industry to grow smoothly we will design and build a system that depends on the Internet of things to monitor production. This system will allow direct preventive non-destructive testing on the equipment with the lowest possible cost and without the help of an operator, which makes production efficient.

Remerciements

Ce mémoire est le résultat d'un travail de recherche qui a duré toute la période de confinement.

En préambule, je tiens tout particulièrement à remercier Monsieur le professeur Fezari qui, en tant que directeur de mémoire, s'est toujours montré à l'écoute et qui m'a permis de croire en moi pour faire ce travail, ainsi que pour l'inspiration, l'effort fourni et le temps qu'il nous a consacré depuis le début de l'année, et sans qui ce mémoire n'aurait jamais vu le jour.

Je remercie également mes parents qui m'ont soutenu, encouragé et qui ont toujours répondu présent, et sans qui je n'aurai jamais pu arriver à ce stade. Enfin, j'adresse mes plus sincères remerciements à mes proches, amis, et toute personne ayant participé à l'élaboration de ce mémoire.

Je dédie ce mémoire,

A mes chers parents qui m'ont fourni les conditions nécessaires pour la poursuite de mes études et qui se sont sacrifiés pour me voir réussir, leur présence et leur soutien tout au long de mon parcours universitaire ont été ma source d'inspiration et de motivation.

A mon frère et ma sœur, qui seuls savent me redonner le sourire et répondent présents à chaque fois que j'ai besoin d'eux. Et enfin à toute ma famille que j'aime ainsi qu'à mes amis qui m'ont toujours soutenu et encouragé durant ces années d'études. Ce travail vous est dédié.

Table des tableaux :

Table des figures :

Table des matières

Introduction générale

Avant la révolution industrielle le monde ne connaissait pas la vitesse, il y avait des ateliers de fabrication où les ingénieurs avaient des problèmes simples à résoudre d'une complexité primaire sans avoir à interrompre la fabrication.

Après la révolution industrielle, tous les procédés ont été automatisés, d'une simple machine de remplissage à une infrastructure compliquée pour créer des voitures, d'un simple bloque de métal à un moteur à 4 cylindres. Tout ceci prenait de l'ampleur mais aussi de la complexité. Tout est lié, du début de la fabrication jusqu'à la sortie du produit. Chaque machine qui entrait dans le procédé de fabrication avait une importance majeure, un simple arrêt engendrant une pause générale au sein de la fabrication.

De nos jours, il existe plusieurs façons pour détecter les défauts sans passer par l'ouverture de la machine. Certes, ces méthodes ne sont pas appliquées tout le temps sur la machine et ne possèdent pas une disponibilité infinie à cause de leur prix, leur taille ou leur protocole de communication qui est restreint, ce qui laisse la surveillance partielle et non complète. Ceci nous conduit à voir les nouvelles technologies, l'internet des objets ou L'IoT, un objet qui n'a pas de limite dans la communication du fait qu'il est connecté à internet avec une très grande modularité et un léger coût pour une large application sur un vaste territoire.

Chapitre I : l’état de l’art (Contrôle non destructif)

I.1. Introduction du contrôle non destructif

Le contrôle non destructif (CND) est un vaste groupe de techniques d'analyse utilisées dans l'industrie scientifique et technologique pour évaluer les propriétés d'un matériau, d'un composant ou d'un système sans causer de dommages. Les termes examen non destructif (END), inspection non destructive (IND) ou évaluation non destructive (END) sont également couramment utilisés pour décrire cette technologie. Étant donné que les CND ne modifient pas de manière permanente l'article à inspecter, il s'agit d'une technique très précieuse qui peut économiser de l'argent et du temps dans l'évaluation, le dépannage et à la recherche du problème. [1]

Les phénomènes physiques sur lesquels repose chacune de ces techniques sont très différents et le choix de l'une d'entre elles dans une application dépend d'un certain nombre de paramètres, principalement : la nature des matériaux à contrôler, la nature de l'information recherchée (détection ou mesure, position et forme du défaut), l'environnement du contrôle, le type de contrôle à effectuer (contrôle en ligne, sur pièce mobile, possibilité de contact ou non) et le contexte économique (coût, …). Pour que l'une de ces techniques soit appliquée, il est bien sûr nécessaire que les propriétés des matériaux à contrôler soient compatibles avec le phénomène physique mis en jeu par la méthode de CND choisie et que les grandeurs mesurées soient significatives afin d'être correctement interprétées.

L'analyse et la documentation d'un mode de défaillance non destructive peuvent également être effectuées à l'aide d'une caméra à haute vitesse enregistrant en continu (boucle vidéo) jusqu'à ce que la défaillance soit détectée. La détection de la panne peut être effectuée à l'aide d'un détecteur de son ou d'une jauge de contrainte qui produit un signal pour déclencher la caméra à grande vitesse. Ces caméras haute vitesse ont des modes d'enregistrement avancés pour capturer certaines défaillances non destructives. Après l'échec, la caméra rapide arrêtera l'enregistrement. Les images capturées peuvent être lues au ralenti montrant précisément ce qui s'est passé avant, pendant et après l'événement non destructif, image par image.[2]

I.2. Les méthodes du Contrôle non destructif (CND).

Le CND a évolué au cours de ces dernières années avec lequel de nouvelles méthodes et procédés sont créés afin de permettre de contrôler une machine, un moteur ou un système complexe.

Les six méthodes les plus fréquemment utilisées sont les tests par courants de Foucault, par particules magnétiques, par ressuage, par radiographie, par ultrasons et visuels. Le CND est couramment utilisé en génie judiciaire, génie mécanique, génie pétrolier, génie électrique, génie civil, génie des systèmes, génie aéronautique, médecine et art. Les innovations dans le domaine des tests non destructifs ont eu un impact profond sur l'imagerie médicale, y compris sur l'échocardiographie, l'échographie médicale et la radiographie numérique.

Il existe plusieurs types sur le marché, et l'exigence varie selon la fonction et le besoin, pour être plus direct il existe beaucoup de solutions qui diffèrent dans le volume, la maniabilité, la mobilité et le coût.

I.2.1. Contrôle par rayon

Le plus connu dans le monde, du fait qu'il est utilisé dans le milieu médical. Des rayons X pénètrent l'objet sans avoir à l'ouvrir ou le démonter, ce qui donne une vision à l'intérieur d'objet en marche sans interférer le mécanisme ou le process.

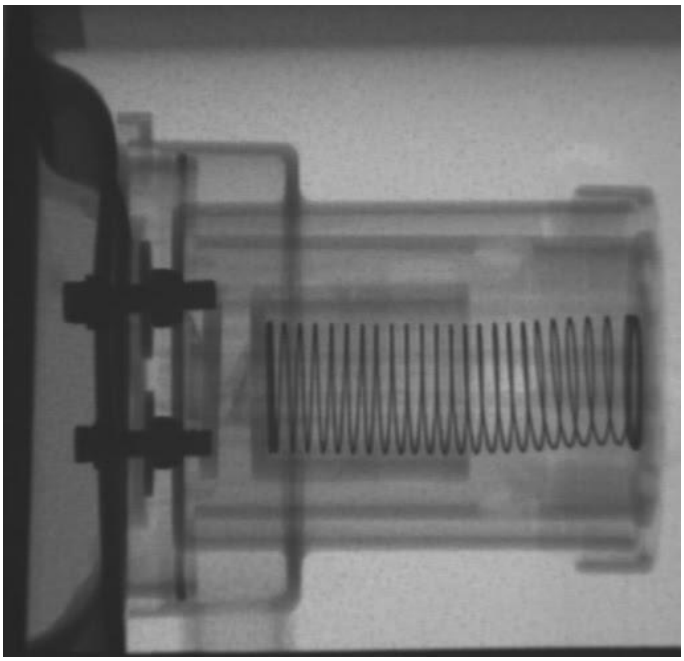


Figure 1 imagerie à rayon x d'un ressort fin dans un capteur

Le phénomène repose sur le type de rayonnements utilisés, puisque les rayons X pénètrent presque tous les matériaux, les rayons percutent le film ou le capteur qui est surfacique pour révéler l'image. Le taux de pénétration est lié à la puissance du rayonnement permettant de sélectionner telle ou telle partie à voir avec un léger contraste.

Les machines à rayons X sont d'une taille moyenne mais difficiles à transporter et mettre en place puisque qu'il y a deux parties (l'émetteur et le capteur surfacique) qu'il faut aligner pour la faire marcher.

Figure 2 Imagerie médicale

La surface d'émission du capteur et du récepteur joue un rôle essentiel dans l'analyse des objets de taille élevée ce qui diminue la maniabilité dans un milieu industriel là où toutes les machines occupent déjà un volume considérable.

Il existe une technique qui consiste à utiliser gamma comme rayonnement pour aboutir à une meilleure pénétration que le rayon X

I.2.2. Contrôle par courants de Foucault

La méthode est simple, une induction est faite afin de créer un courant alternant dans une surface métallique, comme la loi de Lenz Faraday le flux magnétique varie avec la variation du courant, ce courant peut être dévié avec des fissures ce qui rend la détection assez simple dans des coques ou des tôles.[3]

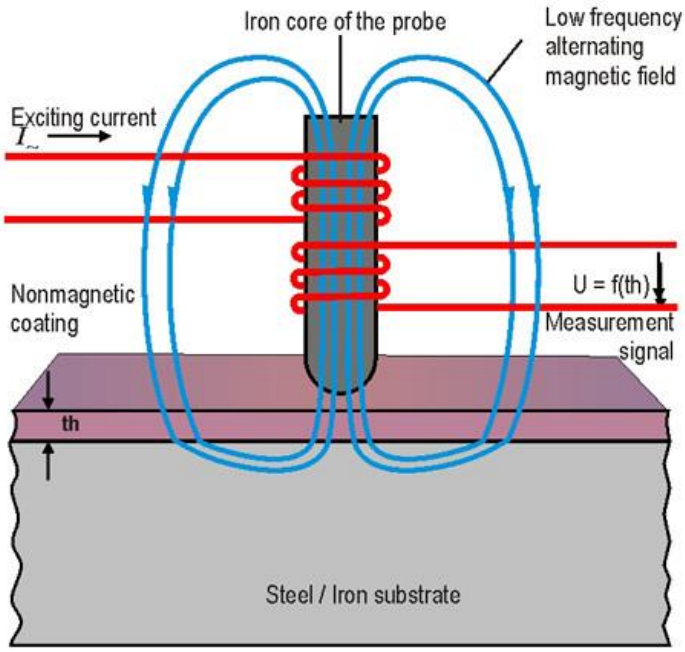


Figure 3 Procédé d'induction magnétique

I.1.1. Contrôle par ressuage

C'est une méthode destinée à révéler la présence de discontinuités ouvertes en surface de pièces métalliques, essentiellement, mais aussi en céramique. Elle consiste à badigeonner (par immersion ou par pulvérisation électrostatique, parfois mais rarement, au pinceau) la cible avec un liquide fluorescent ou coloré en rouge, qui pénètre dans les discontinuités. Après nettoyage de la cible, un révélateur est appliqué et, en faisant « ressuer » le liquide resté dans les fissures, va les révéler.

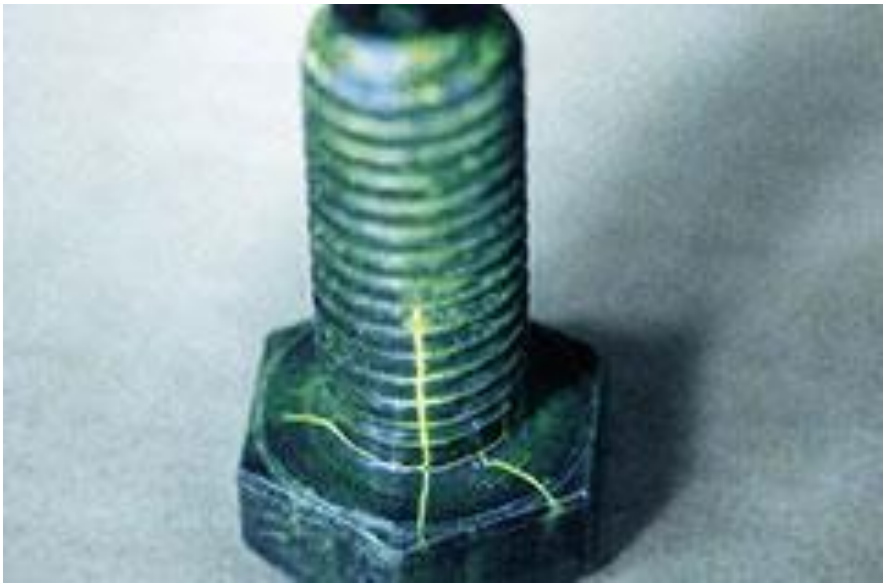


Figure 4 un boulon qui révèle des défauts après ressuage

I.1.2. Contrôle par ultrasons

Le contrôle par ultrasons (UT) est une famille de techniques de contrôle non destructif basées sur la propagation d'ondes ultrasonores dans l'objet ou le matériau testé. Dans les applications UT les plus courantes, des ondes ultrasonores très courtes avec des fréquences centrales allant de 0,1 à 15 MHz, et parfois jusqu'à 50 MHz, sont transmises dans les matériaux pour détecter les défauts internes ou pour caractériser les matériaux. Un exemple courant est la mesure d'épaisseur par ultrasons, qui teste l'épaisseur de l'objet d'essai, par exemple, pour surveiller la corrosion de la tuyauterie. [4]

Les tests par ultrasons sont souvent effectués sur l'acier et d'autres métaux et alliages, bien qu'ils puissent également être utilisés sur le béton, le bois et les composites, bien qu'avec une résolution moindre. Ils sont utilisés dans de nombreuses industries, notamment la construction en acier et en aluminium, la métallurgie, la fabrication, l'aérospatiale, l'automobile et d'autres secteurs du transport.

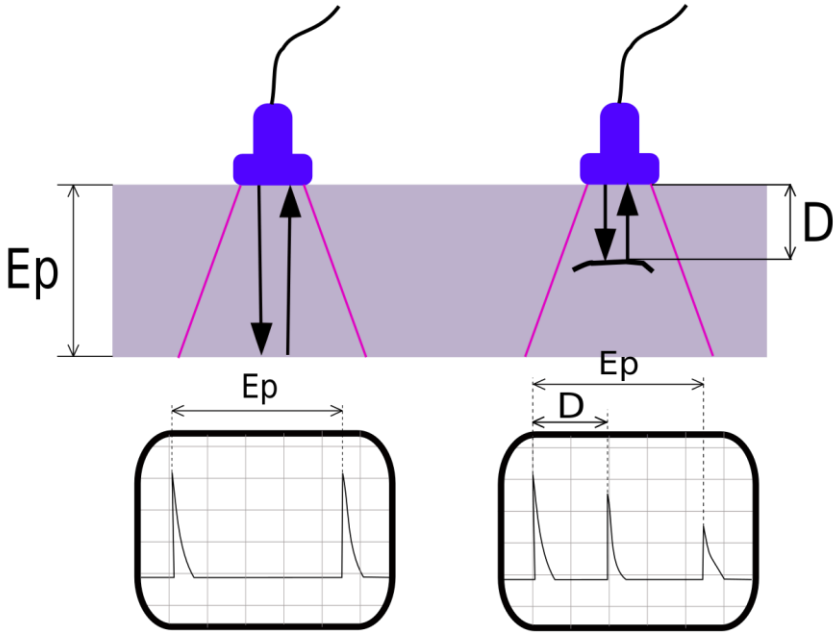


Figure 5 illustration qui explique le contrôle par ultrason

I.1.3. Contrôle par analyse vibratoire

L'analyse des vibrations est définie comme un processus permettant de mesurer les niveaux et les fréquences de vibration des machines, puis d'utiliser ces informations pour analyser l'état de santé des machines et de leurs composants. Alors que le fonctionnement interne et les formules

utilisées pour calculer diverses formes de vibrations peuvent se compliquer, tout commence par l'utilisation d'un accéléromètre pour mesurer les vibrations. Chaque fois qu'une machine fonctionne, elle produit des vibrations. Un accéléromètre attaché à la machine génère un signal de tension qui correspond à la quantité de vibration et à la fréquence de vibration que la machine produit, généralement combien de fois par seconde ou par minute la vibration se produit.

I.2. L'application real du CND sur terrain

Le CND a plusieurs façons pour être mis en place, d'un simple capteur à une machine toute entière jusqu'à un hangar dédié uniquement pour lui. Afin d'établir une idée claire, un tableau est nécessaire pour avoir une vue générale.

Type	Efficacité	Taille	Illustration
Analyse vibratoire	Permet une analyse complète sur un spectre de 10 Hz jusqu'à 10 KHz	$30cm^3$	
Rayon X	permet une analyse approfondie dans un système complexe à base mécanique	De $5m^3$ a $10m^3$	

Tableau 1. Les différentes machines de contrôle non destructif

Chapitre II : le Contrôle Non destructif et les objets connectés

II.1. Introduction des objets connectés.

L'Internet des objets (IoT) est un système de dispositifs informatiques interdépendants, de machines mécaniques et numériques dotées d'identifiants uniques (UID) et de la capacité de transférer des données sur un réseau sans nécessiter d'interaction d'homme à homme ou d'homme à ordinateur.

La définition de l'Internet des objets a évolué en raison de la convergence de plusieurs technologies, de l'analyse en temps réel, de l'apprentissage automatique, des capteurs de produits de base et des systèmes embarqués. Les domaines traditionnels des systèmes embarqués, des réseaux de capteurs sans fil, des systèmes de contrôle, de l'automatisation (y compris la domotique) et d'autres contribuent tous à permettre l'Internet des objets. Sur le marché grand public, la technologie IoT est le plus souvent synonyme de produits appartenant au concept de « maison intelligente », y compris les appareils (tels que les luminaires, les thermostats, les systèmes de sécurité domestique et les caméras, et d'autres appareils ménagers) qui prennent en charge un ou des écosystèmes plus courants et peuvent être contrôlés via des appareils associés à cet écosystème, tels que les smartphones.

II.1.1. Tendances

La tendance majeure et significative de l'IoT ces dernières années est la croissance explosive des appareils connectés et contrôlés par Internet. Le large éventail d'applications de la technologie IoT signifie que les spécificités peuvent être très différentes d'un appareil à l'autre, mais il existe des caractéristiques de base partagées par la plupart.

L'IoT crée des opportunités pour une intégration plus directe du monde physique dans des systèmes informatiques, ce qui se traduit par des améliorations de l'efficacité, des avantages économiques et une réduction des efforts humains.

Le nombre d'appareils IoT a augmenté de 31% d'une année sur l'autre pour atteindre 9,6 milliards en 2017 et on estime qu'il y aura 30 milliards d'appareils d'ici 2021. La valeur marchande mondiale de l'IoT devrait atteindre 9,2 billions de dollars d'ici 2021.

II.1.2. Les caractéristiques D'IoT

Intelligence :

L'intelligence ambiante et le contrôle autonome ne font pas partie du concept original de l'Internet des objets. L'intelligence ambiante et le contrôle autonome ne nécessitent pas nécessairement non plus de structures Internet. Cependant, il y a un changement dans la

recherche (par des entreprises comme Intel) pour intégrer les concepts de l'IoT et du contrôle autonome, avec les premiers résultats vers cette direction considérant les objets comme le moteur de l'IoT autonome. Une approche prometteuse dans ce contexte est l'apprentissage par renforcement profond où la plupart des systèmes IoT fournissent un environnement dynamique et interactif. Former un agent (c'est-à-dire un appareil IoT) à se comporter intelligemment dans un tel environnement ne peut pas être abordé par des algorithmes d'apprentissage automatiques conventionnels tels que l'apprentissage supervisé. Grâce à une approche d'apprentissage par renforcement, un agent d'apprentissage peut détecter l'état de l'environnement (par exemple, détecter la température de la maison), effectuer des actions (par exemple, activer ou désactiver le système CVC) et apprendre en maximisant les récompenses accumulées qu'il reçoit à long terme.

Architecture (réseau) :

L'architecture du système IoT, dans sa vue simpliste, se compose de trois niveaux :

Niveau 1 : appareils
Niveau 2 : la passerelle Edge
Niveau 3 : le cloud

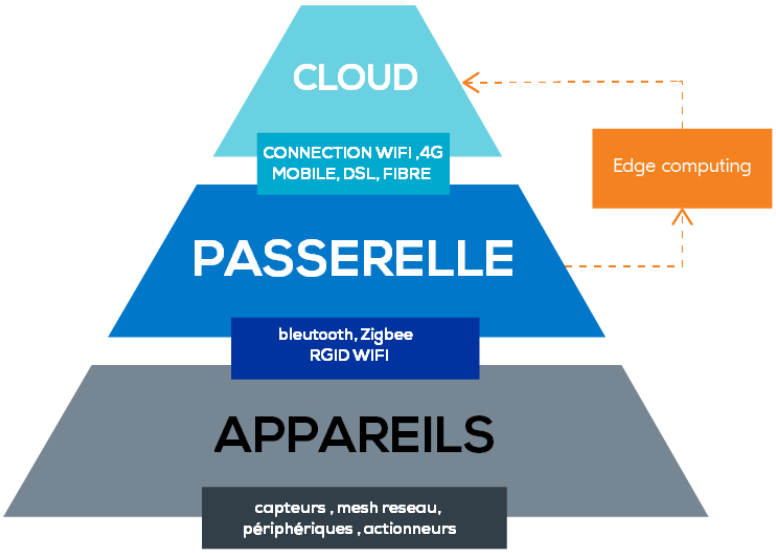


Figure 6 Les différents niveaux dans un réseau D'IoT

Les appareils comprennent des éléments en réseau, tels que les capteurs et les actionneurs présents dans les équipements D'IoT, en particulier ceux qui utilisent des protocoles tels que Modbus, Bluetooth, Zigbee ou des protocoles propriétaires, pour se connecter à une passerelle Edge. La passerelle Edge se compose de systèmes d'agrégation de données de capteur appelés passerelles Edge qui fournissent des fonctionnalités, telles que le prétraitement des données, la sécurisation de la connectivité au cloud, à l'aide de systèmes tels que WebSockets, le hub

d'événements et, même dans certains cas, des analyses de périphérie ou calcul du brouillard. La couche Edge Gateway est également nécessaire pour donner une vue commune des périphériques aux couches supérieures afin de faciliter la gestion. Le dernier niveau comprend l'application cloud conçue pour L'IoT à l'aide de l'architecture des microservices, qui sont généralement polyglottes et de nature intrinsèquement sécurisée à l'aide de HTTPS / OAuth. Il comprend divers systèmes de base de données qui stockent des données de capteur, telles que des bases de données chronologiques ou des magasins d'actifs utilisant des systèmes de stockage de données d'arrière-plan (par exemple Cassandra, PostgreSQL). Le niveau cloud de la plupart des systèmes IoT basés sur le cloud comprend un système de mise en file d'attente d'événements et de messagerie qui gère la communication qui se déroule à tous les niveaux. Certains experts ont classé les trois niveaux du système L'IoT comme étant la périphérie, la plate-forme et l'entreprise et ceux-ci sont connectés respectivement par un réseau de proximité, un réseau d'accès et un réseau de service. [5]

Considérations de taille :

L'Internet des objets encoderait entre 50 et 100 billions d'objets. Les êtres humains dans les environnements urbains étudiés sont chacun entourés de 1 000 à 5 000 objets traçables.

Considérations d'espace :

Dans l'Internet des objets, bien que l'emplacement géographique précis ainsi que les dimensions géographiques précises sont critiques, la personne qui traite les informations peut décider si ces dernières sont importantes ou non pour l'action entreprise.[6]

II.1.3. Les technologies pour l'IoT

Il existe de nombreuses technologies qui permettent l'IoT. Le réseau utilisé pour communiquer entre les appareils d'une installation IoT est crucial pour le terrain, un rôle que plusieurs technologies sans fil ou filaire peuvent remplir :

Sans fil à courte portée

Réseau maillé Bluetooth : Spécification fournissant une variante de réseau maillé à Bluetooth Low Energy (BLE) avec un nombre accru de nœuds et une couche d'application normalisée.

Light-Fidelity (Li-Fi) : Technologie de communication sans fil similaire à la norme Wi-Fi, mais utilisant la communication par lumière visible pour une bande passante accrue.

Communication en champ proche (NFC) : Protocoles de communication permettant à deux appareils électroniques de communiquer dans un rayon de 4 cm.

Identification par radiofréquence (RFID) : Technologie utilisant des champs électromagnétiques pour lire les données stockées dans des étiquettes intégrées dans d'autres éléments.

Wi-Fi : Technologie de mise en réseau locale basée sur la norme IEEE 802.11, où les appareils peuvent communiquer via un point d'accès partagé ou directement entre des appareils individuels.

ZigBee : Protocoles de communication pour les réseaux personnels basés sur la norme IEEE 802.15.4, offrant une faible consommation d'énergie, un faible débit de données, un faible coût et une portée de communication.

Z-Wave : Protocole de communication sans fil utilisé principalement pour les applications domotiques et de sécurité.

Sans fil à moyenne portée

LTE-Advanced(4G) : Spécification de communication à haut débit pour les réseaux mobiles. Fournit des améliorations à la norme LTE avec une couverture étendue, un débit plus élevé et une latence plus faible.

Les réseaux sans fil 5G : peuvent être utilisés pour répondre aux exigences de communication élevées de l'IoT et connecter un grand nombre d'appareils IoT, même lorsqu'ils sont en déplacement.

Sans fil longue portée

Réseau étendu à faible consommation (LPWAN) : Réseaux sans fil conçus pour permettre une communication longue portée à un faible débit de données, réduisant la puissance et les coûts de transmission. Technologies et protocoles LPWAN disponibles: LoRaWan, Sigfox, NB-IoT, Weightless, RPMA.

Terminal à très petite ouverture (VSAT) : Technologie de communication par satellite utilisant des petites antennes paraboliques pour les données à bande étroite et large bande.

LoRa :

Long Range est un protocole de réseau étendu à faible puissance (LPWAN) développé par Semtech. Il repose sur des techniques de modulation à spectre étalé dérivées de la technologie CSS (Chirp Spread Spectrum).

Filaire

Ethernet : Norme de mise en réseau à usage général utilisant des paires torsadées et des liaisons à fibre optique en conjonction avec des concentrateurs ou des commutateurs.

Communication sur courant porteur (PLC) : Technologie de communication utilisant un câblage électrique pour transporter l'alimentation et les données. Les spécifications telles que HomePlug ou G.hn[1] utilisent des API pour la mise en réseau des périphériques IoT.

II.1.4. Espressif Systems

Espressif Systems est une multinationale publique de semi-conducteurs sans usine créée en 2008, dont le siège social est à Shanghai avec des bureaux en Chine, à Singapour, en Inde, en République Tchèque et au Brésil. L'équipe est constituée de passionnée, d'ingénieurs et de scientifiques du monde entier, concentrés sur le développement de solutions IoT de pointe WiFi et Bluetooth à faible consommation d'énergie. Ils ont créé les célèbres séries de puces, modules et cartes de développement ESP8266, ESP32 et ESP32-S. En tirant parti de l'informatique sans fil, ils proposent des chipsets écologiques, polyvalents et économiques. Ils se sont toujours engagés à offrir des solutions IoT sécurisées, robustes et écoénergétiques. Leur vision est de permettre aux développeurs d'utiliser la technologie d'Espressif dans le monde entier et de créer des appareils connectés intelligents.

II.1.4.1. ESP32 Espressif

ESP32 est une série de systèmes à faible coût et à faible consommation d'énergie sur des microcontrôleurs à puce avec Wi-Fi intégré et Bluetooth double mode. La série ESP32 utilise un microprocesseur Tensilica Xtensa LX6 dont les variantes double cœur et monocœur et comprend des commutateurs d'antenne intégrés, un balun RF, un amplificateur de puissance, un amplificateur de réception à faible bruit, des filtres et des modules de gestion de l'alimentation. Fabriqué par TSMC en utilisant leur procédé 40 nm, on peut le considérer comme un successeur du microcontrôleur ESP8266.

[1] **G.hn** est une spécification pour les réseaux domestiques avec des débits de données allant jusqu'à 2 Gbit/s.

II.1.4.2. Les Caractéristiques techniques de ESP32

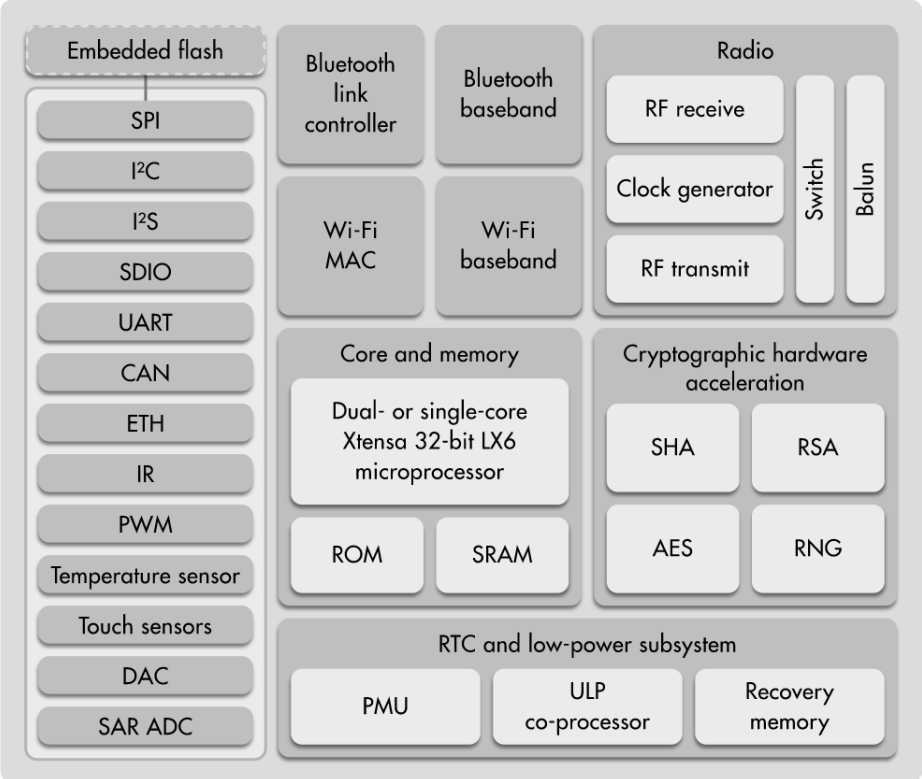


Figure 7 Diagramme de block de fonction d'ESP32

Processeur : microprocesseur Xtensa double cœur (ou monocœur) LX6 32 bits, fonctionnant à 160 ou 240 MHz et fonctionnant jusqu'à 600 DMIPS

Coprocesseur ultra faible puissance (ULP)

Mémoire : 520 KiB SRAM

Wi-Fi : 802.11 b / g / n

Bluetooth : v4.2 BR / EDR et BLE (partage la radio avec le Wi-Fi)

Interfaces périphériques :

ADC SAR 12 bits jusqu'à 18 canaux

2 × DAC 8 bits

10 × capteurs tactiles (GPIO à détection capacitive)

4 × SPI ,2 × interfaces I²S, 2 × interfaces I²C, 3 × UART

Contrôleur hôte SD / SDIO / CE-ATA / MMC / eMMC

Contrôleur esclave SDIO / SPI

Interface MAC Ethernet avec prise en charge du protocole DMA dédié et IEEE 1588 Precision Time

Bus CAN 2.0

Télécommande infrarouge (TX / RX, jusqu'à 8 canaux)

Capteur à effet Hall

Préamplificateur analogique ultra faible puissance

Gestion de l'alimentation :

Régulateur interne à faible décrochage

Domaine de puissance individuel pour RTC

Courant de sommeil profond de 5 µA

Réveil après interruption GPIO, minuterie, mesures ADC, interruption du capteur tactile capacitif

II.1.4.3. Le ESP32 dans le milieu industriel

Le ESP32 est utilisé pour la première fois comme IoT industriel par la firme TECHBASE qui est experte dans le domaine industriel avec la série Modiano qui se dote d'un CPU ESP32-WROVER pour des utilisations comme Capteurs De Points Finaux (CDPF) ou des actionneurs dans un milieu non câblé.[7]

Figure 8 MODUINO X 1 (un automate à base de ESP32)

II.1.4.4. Le module M5STICKC

M5StickC est un mini M5Stack, alimenté par ESP32. Il s'agit d'une carte de développement IoT portable, facile à utiliser et open source, M5stickC est l'un des principaux périphériques de la série des produits M5Stack.

Figure 9 détails sur M5StickC

II.2. Le contrôle non destructif et les objets connectés aujourd'hui.

Du point de vue des tests non destructifs (CND), toutes les possibilités de l'IoT n'ont pas encore été réalisées, mais les détecteurs de défauts qui peuvent se connecter à un réseau de protocole défini permettent de transférer les données pour une analyse complète à la base d'une façon légère mais sans avoir une infrastructure dédiée à l'utilisation à grande échelle, il n'existe pas de solution qui utilise l'infrastructure native d'IoT comme (MQTT[2] JSON[3]) pour la transmission des données au serveur pour permettre la mise en place d'une architecture communicant avec n'importe quel IoT quelque soit son type ou sa taille.[8]

II.3. L'approche dédiée pour la réalisation

Ces dernières années, l'utilisation de l'IoT dans l'industrie n'a pas encore de standard à suivre du fait qu'il n'est pas assez utilisé. Ceci nous conduit à élaborer un système semblable à l'IoT démotique avec des changements touchant particulièrement son utilisation finale.

[2] **M**essage **Q**ueuing **T**elemetry **T**ransport : protocole de messagerie publish-subscribe basé sur le protocole TCP/IP

[3] **J**ava**S**cript **O**bject **N**otation : un format de données textuelles dérivé de la notation des objets du langage JavaScript.

La stratégie élaborée est de programmer un M5STICKC, de prendre des mesures sur un moteur et de les envoyer à un serveur sur internet pour permettre d'une part une analyse complexe de l'échantillon sans gaspillage énergétique dans les calculs localement. Et d'autre part, de délimiter le réseau à un plus large rayon pour avoir plusieurs objets connectés afin d'avoir une vue constante sur les moteurs à surveiller.

II.3.1. L'architecture du réseau élaborée :

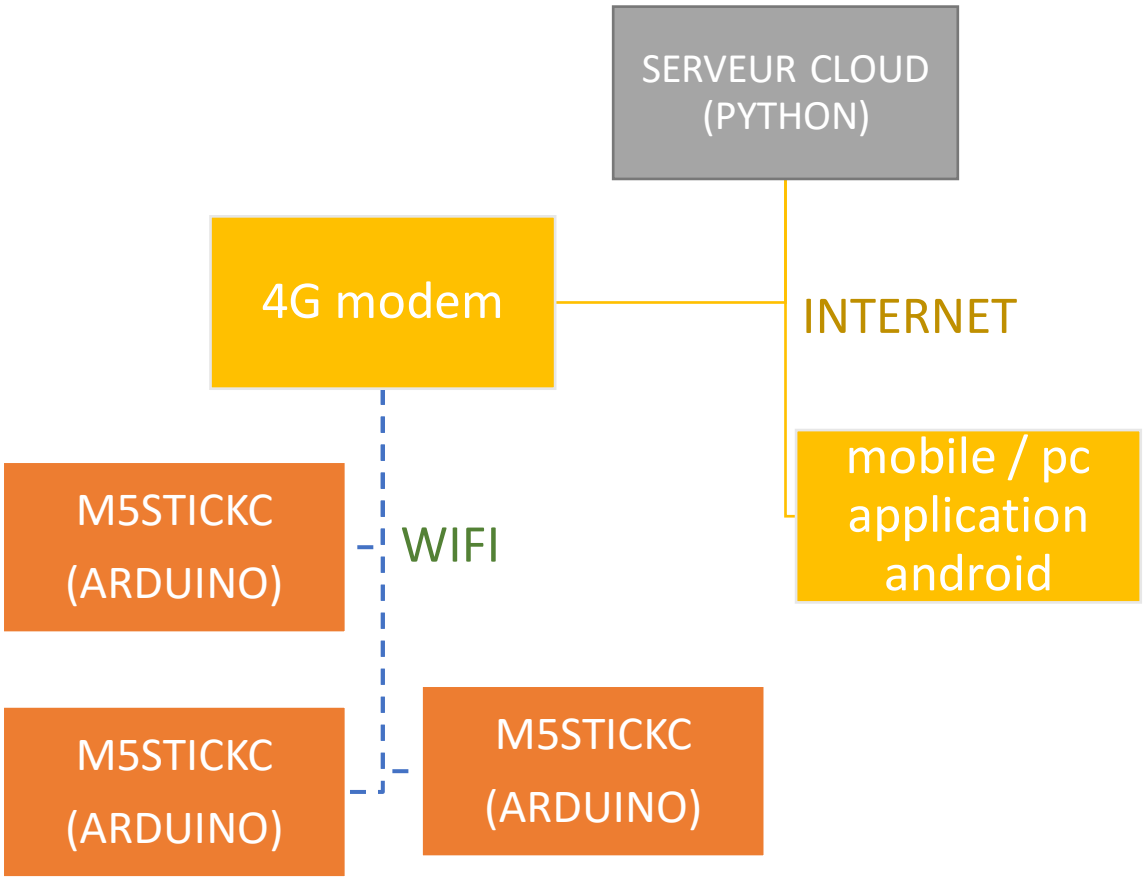


Figure 10 architecture du réseau

L'élaboration du réseau se fera comme l'IoT domotique avec un serveur en PYTHON[4] dans le cloud qui permet une liaison directe avec les objets connectées passant par une borne de

[4] **Python** est un langage de programmation interprété, de haut niveau et polyvalent.

connexion wifi qui donne l'accès à internet, utilise notamment une application pour voir les données traitées et voir l'avancement des choses.

II.3.2. Le privilège du M5STICKC dans le contrôle non destructif.

Pour que le M5STICKC soit utilisé à sa toute-puissance il faudra que le code élaboré exploite les 2 capteurs embarqués, le microphone SPM1423 connecté en I2S et l'accéléromètre MPU6886 connecté en I2C avec L'ESP32 tout en se connectant au wifi pour transmettre les données et économiser de l'énergie avec un sommeil profond qui est natif dans les options du microprocesseur.

L'avantage du M5STICKC est de faciliter l'affichage d'une information. Du fait qu'il y ait un écran intégré, il peut permettre une interaction directe dans sa configuration ou permettre d'afficher le statut de l'objet connecté en mode enregistrement, arrêt ou envoi d'information.

La taille de la batterie est considérablement petite ce qui nous amène à faire des tests distants en longe durée.

L'M5STICKC effectuera les 3 tâches suivantes en boucle pour permettre une large couverture du contrôle non destructif dans une très longue durée.

Le protocole d'envoi utilisé sera le HTTP[5] dans TCP avec des données JSON pour sa légère charge sur le réseau et pour une facilité d'intégration dans n'importe quel serveur.

[5] Le protocole HTTP (HyperText Transfer Protocol) est un protocole d'application pour les systèmes d'information distribués, collaboratifs et hypermédia.

II.3.2.1. La modularité du M5STICKC

N5StickC est une plateforme de développement modulaire basée sur le SoC ESP32 d'Espressif. Les modules (capteur, actionneur, module de communication) viennent avec les chapeaux sur le M5StickC pour permettre d'ajouter des nouvelles fonctionnalités.

Les modules qui peuvent être importants dans le CND sont :

- M5StickC ADC HAT ADS1100 : un autre type de C-HAT spécialement conçu pour le contrôleur M5StickC. Identique à l'unité ADC, il s'agit d'un composant convertisseur ADC pour M5stickc. Emballé avec une puce de convertisseur ADC ADS1100, qui est un convertisseur A / N delta-sigma entièrement différentiel, 16 bits, auto-calibrant. Extrêmement simple à concevoir et à configurer, l'ADS1100 permet d'obtenir des mesures précises avec un minimum d'effort.

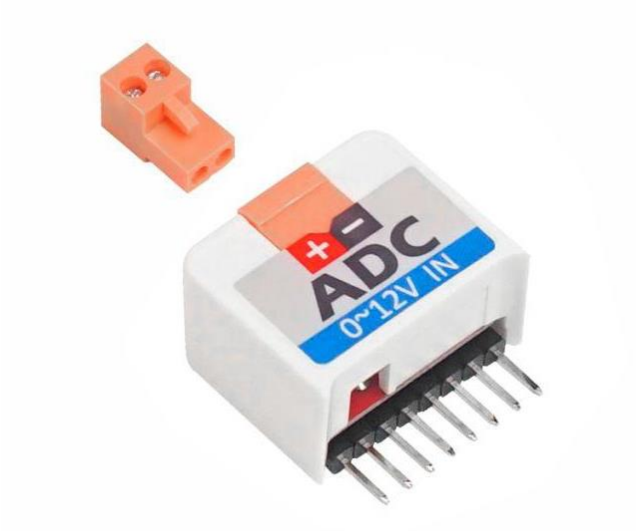


Figure 11 Capteur ADC pour M5STICKC

- M5StickC NCIR Hat(MLX90614): NCIR HAT est un capteur infrarouge compatible M5StickC. Identique à M5Unit NCIR, ce module stickC HAT intègre MLX90614 qui peut être utilisé pour mesurer la température de surface d'un corps humain ou d'un autre objet. Maintenant qu'il a une couverture d'un cas de M5StickC HAT, à peu près toutes les implémentations peuvent être déplacées vers le contrôleur basé sur M5StickC peu coûteuse et hautement productive.

Figure 12 Capteur de température infrarouge

Chapitre III : Le contrôle non destructif avec le langage python en RESTful (serveur)

III.1. Python, un langage de grande flexibilité

L'un des langages de programmation qui a pu attirer l'attention de nombreux développeurs est Python, un langage de programmation à usage général qui est connu pour sa flexibilité et sa simplicité.

Il y a de nombreux avantages à utiliser Python, en particulier en ce qui concerne la science des données. Le langage de codage et l'environnement de développement facilitent la création d'algorithmes complexes sur lesquels le traitement des données peut être effectué. Le code nécessaire le rend évolutif pour réaliser une analyse rapide des données et créer de nouvelles informations à partir des données existantes.

Toute la puissance du Python tient de la diversité des bibliothèques ne laissant aucun problème pour s'intégrer dans différents domaines comme le contrôle non destructif ou notamment dans les objets connectés. Ceci laisse le python comme un langage de premier choix pour un usage qui mélange deux univers différents.

III.1.1. Python est open-source

La meilleure chose qui peut faire évoluer un langage est d'avoir les droits de l'utiliser et de le rénover sans la moindre restriction. Comme Python est open source, son évolution et son intégration a été rapide dans le domaine professionnel. Il a permis à des développeurs d'ajouter, modifier et échanger du contenu contribuant dans son développement ce qui en fait aujourd'hui un langage riche, puissant et flexible, tout en étant gratuit. [9]

III.1.2. Les deux versions de Python

Python existait en version 2 avant que les développeurs décident de passer à la version 3. Puisque la syntaxe a été quelque peu compliquée dans la première version la nouvelle version a simplifié la syntaxe en ajoutant les parenthèses dans les paramètres des fonctions réduisant le risque d'erreur.

La mort de la version 2.7 a été programmée le 1 janvier 2020 avec la dernière mise à jour qui signa son arrêt. À cet instant, le Python 3 est la version officielle de Python. [10]

III.2. Un serveur entier en Python RESTful

La flexibilité dont nous avons parlée plus haut nous donne le pouvoir de créer un serveur qui non seulement peut communiquer avec un objet connecté mais aussi possède la capacité d'avoir des algorithmes à sa disposition. Ceci lui permet d'effectuer des calculs directs sur l'échantillon sonore ou vibratoire sans recourir à la transmission à un serveur dédié ou avoir un autre logiciel

de traitement complexe comme MATLAB, ce qui donne une facilité dans le développement sans recourir à des solutions privées.

Dans cet ensemble, le serveur est un avantage majeur du fait qu'il peut posséder des algorithmes complexes à base de Python ou de Matlab, ce qui laisse l'avantage à des tests de nouvelles approches plus complexes dans un milieu réel.

Avec l'ajout du REST, le serveur aura les propriétés d'un API RESTful, avec des entrées d'information pour permettre à l'objet connecté d'interagir avec le serveur en HTTPS

III.2.1. Définition de REST

Le transfert d'état de représentation (REST) est un style d'architecture logicielle qui définit un ensemble de contraintes à utiliser pour créer des services Web. Les services Web conformes au style architectural REST, appelés services Web RESTful, assurent l'interopérabilité entre les systèmes informatiques sur Internet. Les services Web RESTful permettent aux systèmes demandeurs d'accéder aux représentations textuelles des ressources Web et de les manipuler en utilisant un ensemble uniforme et prédéfini d'opérations sans état. D'autres types de services Web, tels que les services Web SOAP, exposent leurs propres ensembles arbitraires d'opérations.[11]

III.2.1.1. L'architecture REST

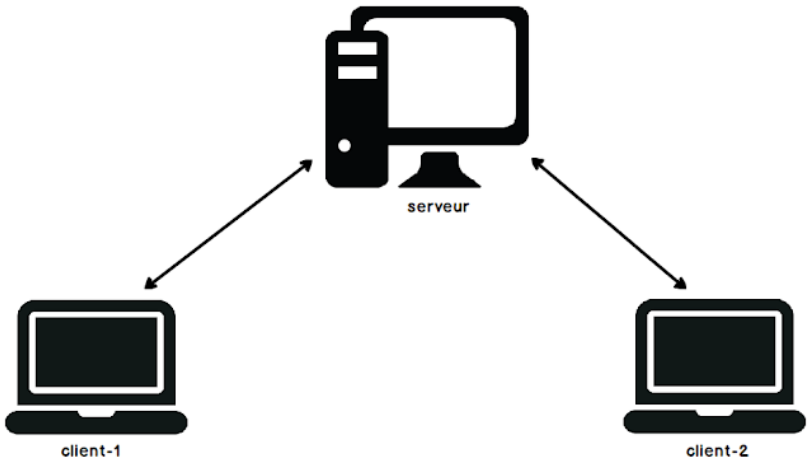
Les contraintes du style architectural REST affectent les propriétés architecturales suivantes :

- Les performances : dans les interactions des composants, qui peuvent être le facteur dominant des performances perçues par l'utilisateur et de l'efficacité du réseau.
- Évolutivité : permettant la prise en charge d'un grand nombre de composants et d'interactions entre les composants. Roy Fielding décrit l'effet de REST sur l'évolutivité.
- Simplicité : d'une interface uniforme.
- Modificabilité : des composants pour répondre aux besoins changeants (même pendant que l'application est en cours d'exécution).
- Visibilité : de la communication entre les composants par les agents de service.
- La portabilité des composants en déplaçant le code du programme avec les données.
- Fiabilité : de la résistance aux pannes au niveau du système en présence de pannes dans les composants, les connecteurs ou les données.

III.2.1.2. Contraintes architecturales du REST

Six contraintes de guidage définissent un système RESTful. Ces contraintes limitent les façons dont le serveur peut traiter et répondre aux demandes des clients de sorte qu'en fonctionnant dans ces limites, le système acquière des propriétés non fonctionnelles souhaitables, telles que les performances, l'évolutivité, la simplicité, la modifiabilité, la visibilité, la portabilité et la fiabilité. Si un système enfreint l'une des contraintes requises, il ne peut pas être considéré comme REST. [12]

- Architecture client-serveur : Le principe des contraintes client-serveur est la séparation des préoccupations. Séparer les préoccupations d'interface utilisateur des préoccupations de stockage de données améliore la portabilité des interfaces utilisateur sur plusieurs plates-formes. Il améliore également l'évolutivité en simplifiant les composants du serveur. Le plus important pour le Web est peut-être que la séparation permet aux composants d'évoluer indépendamment.



Réseau client-serveur

- Apatridie(statelessness) : La communication client-serveur est limitée par le fait qu'aucun contexte client n'est stocké sur le serveur entre les requêtes. Chaque requête de n'importe quel client contient toutes les informations nécessaires pour traiter la demande et l'état de la session est conservé dans le client. L'état de session peut être transféré par le serveur vers un autre service tel qu'une base de données pour maintenir un état persistant pendant une période et permettre l'authentification. La représentation de chaque état d'application contient des liens qui peuvent être utilisés la prochaine fois que le client choisit d'initier une nouvelle transition d'état.

- Cachéabilité : Comme sur le World Wide Web, les clients et les intermédiaires peuvent mettre en cache les réponses. Les réponses doivent, implicitement ou explicitement, se définir comme pouvant être mises en cache ou non pour empêcher les clients de fournir des données périmées ou inappropriées en réponse à d'autres demandes. Une mise en cache bien gérée élimine partiellement ou complètement certaines interactions client-serveur, améliorant encore l'évolutivité et les performances.
- Système en couches : Un client ne peut généralement pas dire s'il est connecté directement au serveur final ou à un intermédiaire en cours de route. Si un proxy ou un équilibreur de charge est placé entre le client et le serveur, cela n'affectera pas leurs communications et il ne sera pas nécessaire de mettre à jour le code client ou serveur. Les serveurs intermédiaires peuvent améliorer l'évolutivité du système en activant l'équilibrage de charge et en fournissant des caches partagés
- Interface uniforme : La contrainte d'interface uniforme est fondamentale pour la conception de tout système RESTful. Il simplifie et découple l'architecture, ce qui permet à chaque pièce d'évoluer indépendamment.

Chapitre IV : Réalisation

Chapitre IV : Réalisation

IV.1. Introduction

Dans ce chapitre, nous évoquerons la réalisation complète, commençant par le serveur jusqu'à l'objet connecté. Pour que le projet soit structuré, il faudra découper le travail en plusieurs étapes pour en visualiser la réalisation mais aussi pour que le projet devienne plus facile à gérer.

La première phase s'intéressera au serveur, la mise en place et la création d'un serveur en Python mais aussi comment le mettre online dans un serveur dédié ou local.

L'installation est différente selon que le serveur soit local ou dédié car dans le cloud il y a un Cpanel avec un Python intégré.

La deuxième phase consistera à programmer l'objet connecté afin qu'il se connecte au serveur pour envoyer les échantillons pour un traitement direct.

IV.2. Initialisation du serveur

IV.2.1. L'installation de Python

IV.2.1.1. L'installation de Python dans le pc serveur comme serveur local

L'installation de Python est simple et facile. Il faut juste avoir l'accès au site officiel de **python https://www.python.org/** pour télécharger la dernière version.

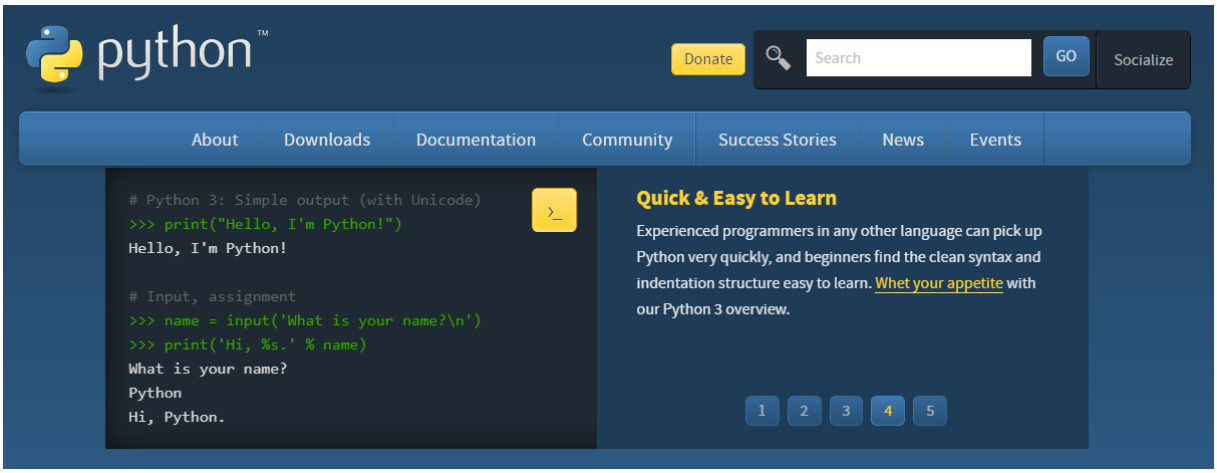


Figure 13 site de Python

Après l'ouverture du site, il faut accéder au téléchargement (Downloads) dans les rubriques en haut pour avoir la dernière version en téléchargement (Download python **x.x.x**),

La dernière version actuelle est 3.8.5. Elle est stable et compatible avec Windows 10, Linux, ou MacOS sans problème.

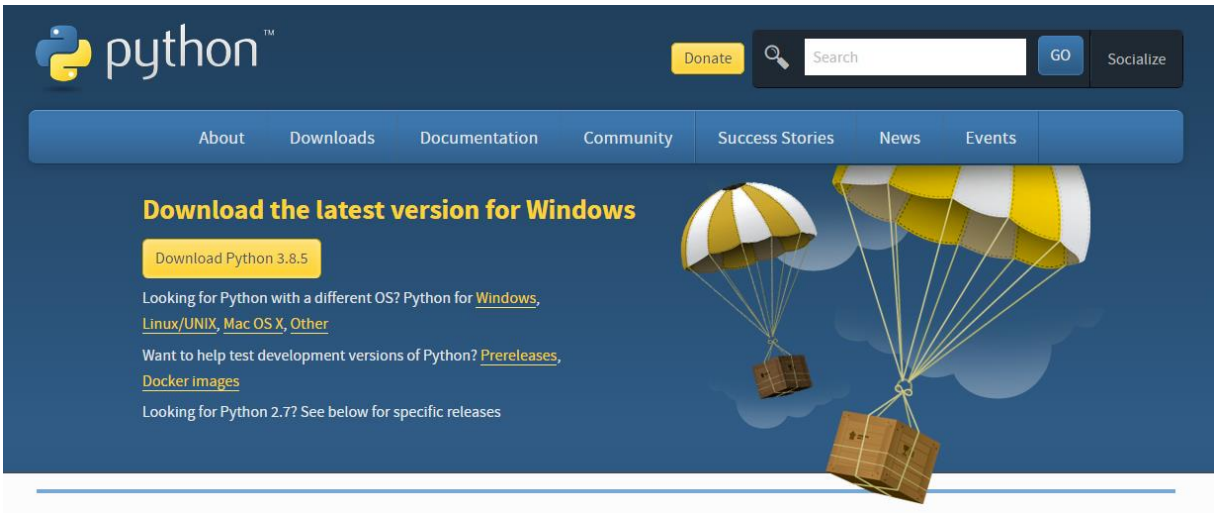


Figure 14 page de téléchargement de Python

Après l'ouverture du fichier d'installation, une fenêtre s'affiche pour permettre de faire des choix, il est nécessaire de l'ajouter dans le PATH Windows en cochant (Add Python 3.8 to PATH)

Figure 15 fenêtre d'installation de Python

Après l'installation, il faut vérifier que Python est accessible de « l'invite de commande » en écrivant **python** dans **un terminal CMD**, le retour sera la version installée dans le pc ainsi que la date de la dernière mise à jour.

Si Python ne s'affiche pas c'est qu'il n'a pas été ajouté à PATH. Il faut refaire toute l'installation pour résoudre ce problème avec l'ajout **ouvrir en mode administrateur** pour que l'installateur ait le privilège d'effectuer des modifications dans le système

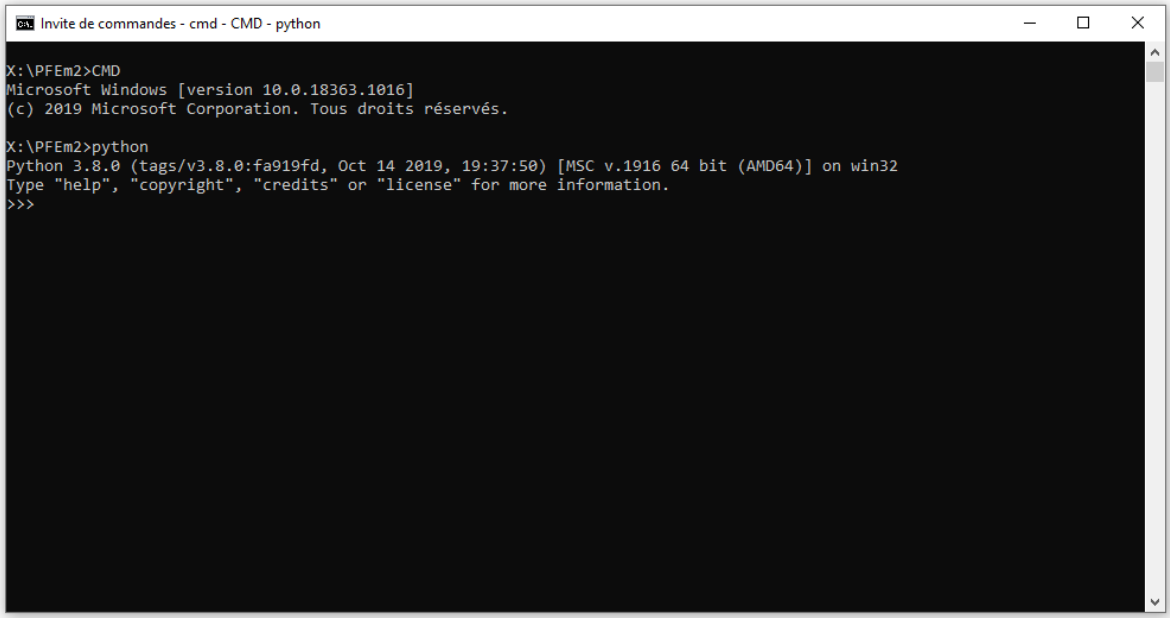


Figure 16 Python executé dans « l'invite de commande »

IV.2.1.2. L'initialisation de python dans le serveur dans le Cloud

L'initialisation de Python dans un serveur diffère totalement de l'initialisation sur un pc en local. Pour faire simple, après l'ouverture du Cpanel, on trouve le bouton **Setup Python App** qui nous conduira à la page des applications Python ajouter au serveur.

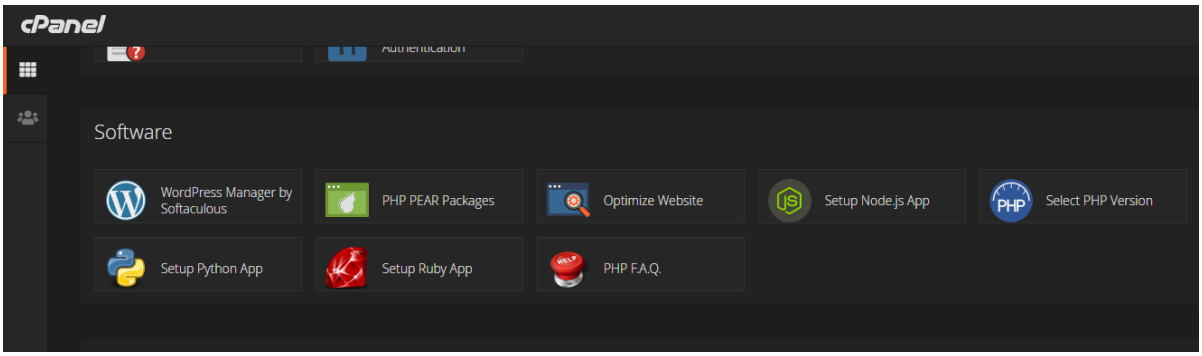


Figure 17 panneau de contrôle Cpanel

Dans cette page, on peut créer une application Python simplement avec un clic sur **CREATE APPLICATION**

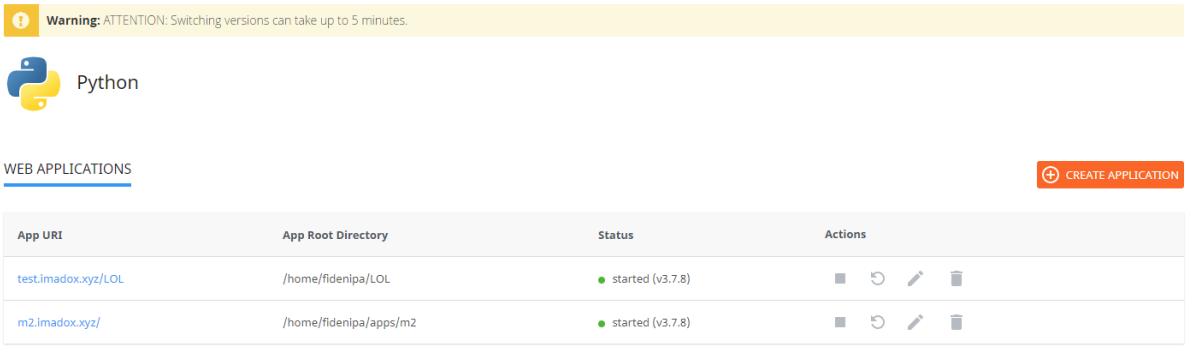


Figure 18 listes des serveurs Python online avant la création

Dans la fenêtre de création d'application Python on retrouve une information à remplir, l'emplacement du programme, la version de Python, l'url utilisé pour le Cloud online ainsi que le ficher de point d'entrée WSGI qui est nécessaire dans la sécurité du serveur.

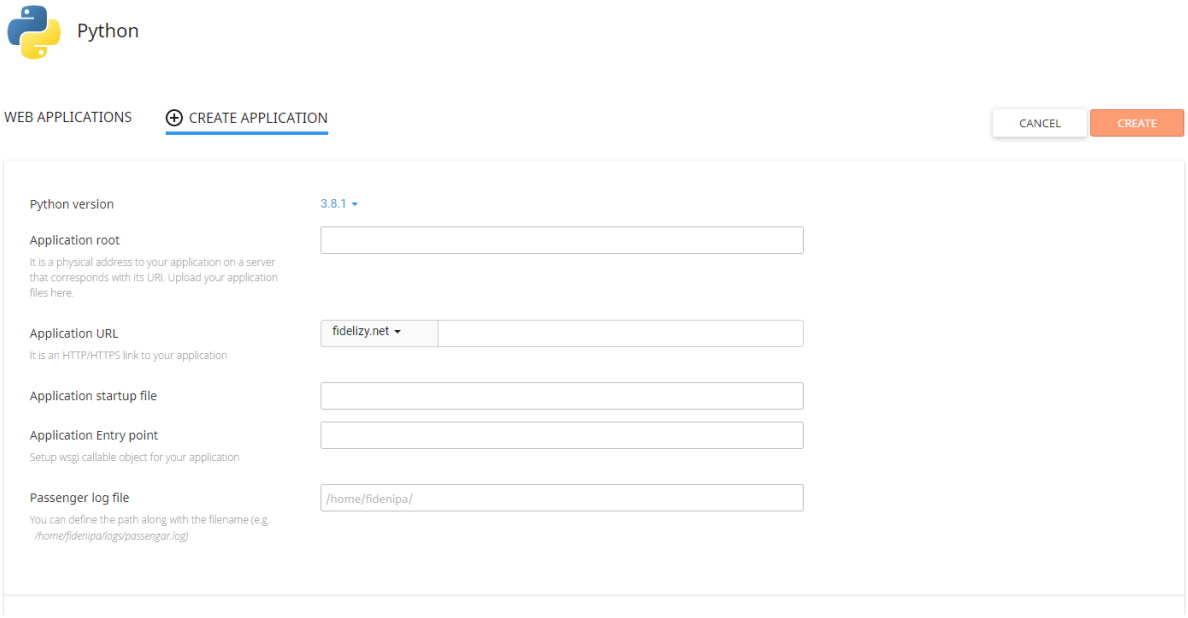


Figure 19 page de création d'une application Python

Après avoir lancé l'application, on peut vérifier qu'elle a bien été lancée en regardant dans la liste des applications Python dans le Cpanel. On trouvera l'application lancée avant avec le lien pour configurer les objets connectés.

On peut contrôler ce serveur Python à partir de cette page, l'arrêter pour une maintenance ou le redémarrer pour appliquer de nouveaux paramètres.

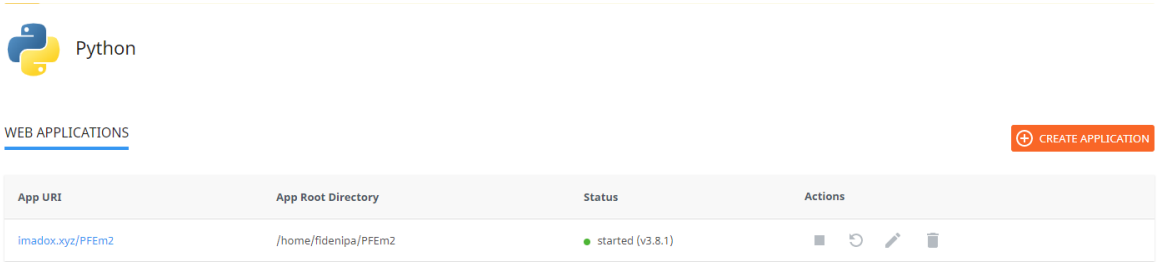


Figure 20 Listes des serveurs Python online après la création

Si on accède au lien, un texte s'affichera confirmant le fonctionnement de Python et la version ajoutée de la part du serveur.

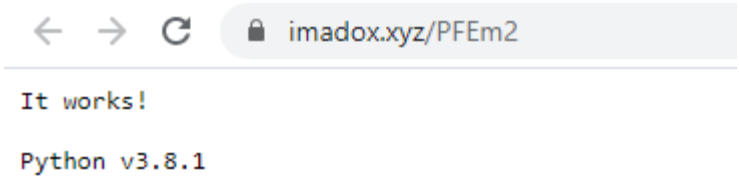


Figure 21 la réponse du serveur Python

IV.2.2. Installation des bibliothèques nécessaires dans Python

Dans notre projet, il y aura six bibliothèques ou modules nécessaires pour permettre à notre serveur de communiquer et de faire une analyse fréquentielle pour avoir le résultat.

La liste des bibliothèques qui vont nous permettre d'établir notre serveur est la suivante :

FLASK : ce module, qui nous donne le pourvoir de créer des API RESTful dynamiques sans recourir à PHP, est nécessaire à la réalisation du serveur.

NUMPY : est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

SCIPY : elle comporte la transformée de Fourier qui va permettre une analyse fréquentielle sur échantillon sans recourir à d'autre langage.

JSON : l'ajout de cette bibliothèque permet la gestion des données JSON pour lire les vecteurs de données envoyés par l'objet connecté.

MATPLOTLIB : permet d'afficher les graphes des fréquences après une analyse de Fourier

Dans le terminal ou SSH si c'est le serveur, il faut écrire **PIP INSTALL** suivi du nom de la Library afin de permettre à Python de chercher la bibliothèque et de l'ajouter automatiquement sans avoir à la télécharger manuellement.

THREADING : permet de lancer un thread (processus) sans recourir a un autre code python.

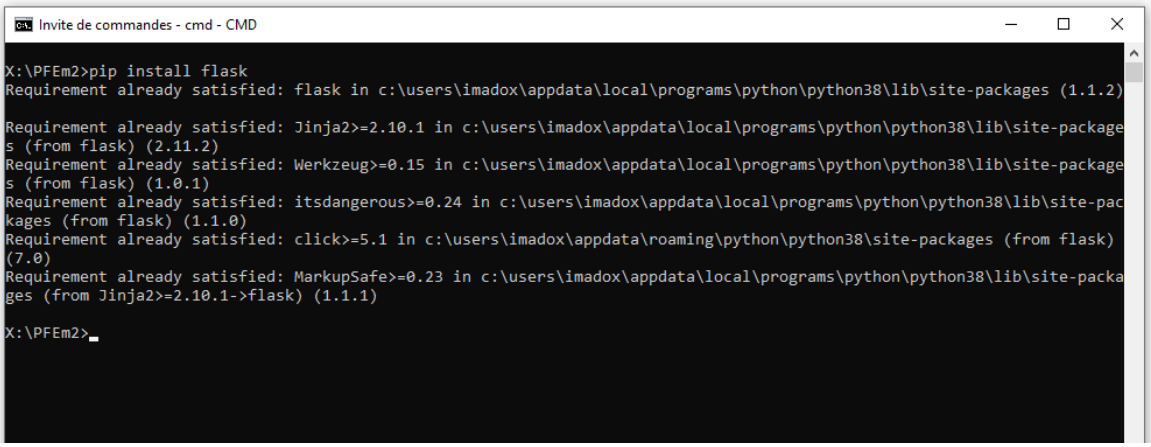


Figure 22 : l'installation d'une bibliothèque Python

Pour avoir la liste des modules installés, une simple commande affichera tous les modules installés avec leurs versions **PIP FREEZE.**

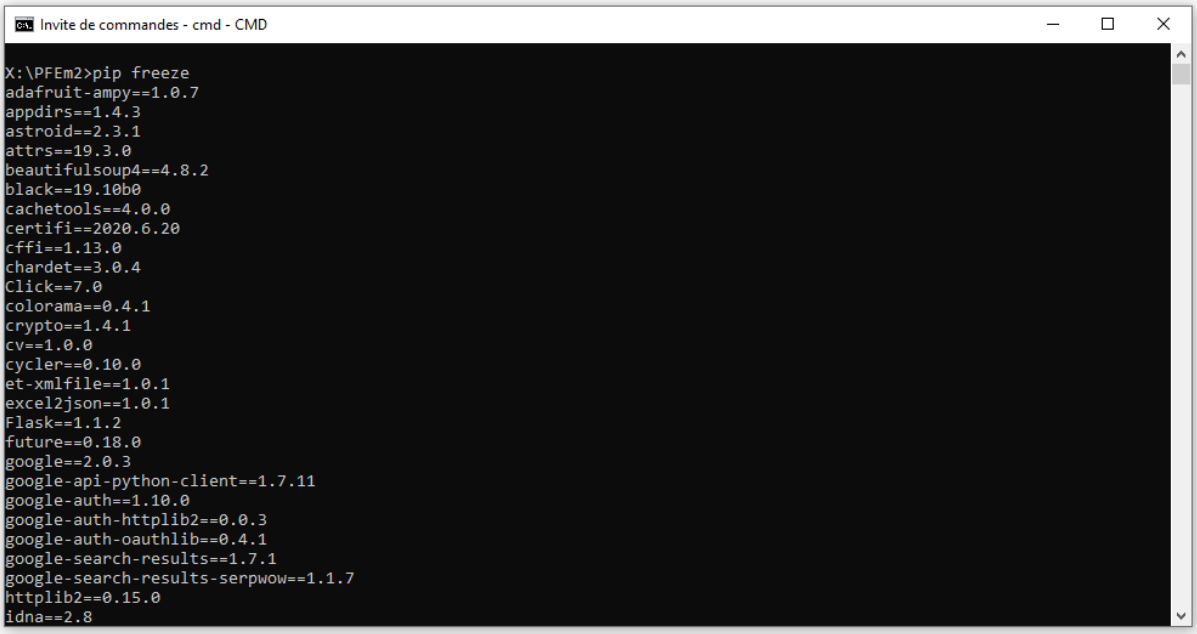


Figure 23 l"affichage des modules Python

IV.2.3.Le code dédier pour le traitement non destructif pour le serveur

```
# importation des modules ou bibliotheques
from __future__ import print_function
from __future__ import division
from flask import *
import json
import logging
from datetime import date
import numpy as np
import matplotlib.pyplot as plt
import scipy.fftpack
from threading import Thread

# fin d'importation des modules ou bibliotheques


app = Flask(__name__)
log = logging.getLogger("test")
log.disabled = True


@app.route("/")
def hello_world():
    return "SERVEUR EN MARCHE"


@app.route("/<user>/<data>")  # declaration de l'entrée d'information
def DATA(data, user):
    y = json.loads("[" + data + "]")
    thread = Thread(target=traitement, args=(10, y, user))
    thread.start()
    return "OK"

# fonction de traitement

def traitement (alpha, jsondata, user):
    # temps d'echentiaunage
    T = 0.001
    # Configuration.
    fS = 1 / T  # Taux d'échantillonnage.
    fH = 10  # frequence de coupure.
    Ns = 400  # Filter length, must be odd.
    # calcule du filtre passe haut.
    h = np.sinc(2 * fH / fS * (np.arange(Ns) - (Ns - 1) / 2))
    # Normaliser pour obtenir un gain unitaire.
    h /= np.sum(h)
    # L'application d'un filtre passe-haut pour filtrer le bruit
    h = -h
    h[(Ns - 1) // 2] += 1
```

```
    print(h)
    jsondata = np.convolve(jsondata, h)
    N = 1000
    x = np.linspace(0.0, N * T, N)
    yf = scipy.fftpack.fft(jsondata)
    xf = np.linspace(0.0, 1.0 / (2.0 * T), N / 2)

    plt.figure(user + str(date.today()))
    plt.plot(xf, 2.0 / N * np.abs(yf[: N // 2]))
    plt.show()


if __name__ == "__main__":
    app.run(host="192.168.10.200")
```

IV.3. Programmation des objets connectés M5StickC

M5StickC est un objet connecté qui supporte l'ARDUINO comme lagunage de base, ce qui nous permet de le programmer facilement sans recourir à ESP-IDF qui est le langage de base du ESP32.

Dans notre programmation on aura 2 variantes. La première variante est le code qui fera une acquisition de type vibration avec l'accéléromètre. La deuxième variante est l'acquisition de son avec le microphone intégré pour l'envoyer au serveur.

IV.3.1. Le code d'acquisition pour M5SttickC avec accéléromètre

Dans la plupart des cas dans l'Arduino il suffit de déclarer une bibliothèque du module pour le faire marcher, mais pas dans le cas ou la carte est un object connecté puisque dans le M5StickC on retrouve une gestion d'alimentation, pour faire simple il suffit de donner l'ordre pour que le circuit de gestion d'alimentation alimente les modules.

```
//déclaration de la bibliothèque M5StickC

#include <M5StickC.h>

// (il ce peu que le choix de l'axe a un effet sur l'acquisition)

float accX = 0;      //acceleration axe X
float accY = 0;      //acceleration axe Y
float accZ = 0;      //acceleration axe Z

float gyroX = 0;      //Gyroscope axe X
float gyroY = 0;      //Gyroscope axe Y
float gyroZ = 0;      //Gyroscope axe Z

//l'initialisation du M5StickC
void setup() {
```

Après la déclaration de la bibliothèque, il faudra ordonner l'alimentation des modules internes

```
  M5.begin(); //L'ordre pour que une alimentation complète des modules

  M5.MPU6886.Init();
  M5.Axp.ScreenBreath(0);


  for (uint8_t t = 4; t > 0; t--) {
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();

  }
  wifiMulti.addAP("L", "ASCII@45");
}
// le démarrage d'acquisition
void loop() {
  if ((wifiMulti.run() == WL_CONNECTED)) {

    if (   M5.BtnA.isReleased() == 0) {
      M5.Axp.ScreenBreath(100);
      M5.Lcd.setTextSize(4);
      M5.Lcd.setCursor(20, 70);
      delay(1000);
      M5.Lcd.println("2");
      M5.Lcd.setCursor(20, 70);
      delay(1000);
      M5.Lcd.println("1");
      delay(1000);
      M5.Lcd.fillScreen(WHITE);

      HTTPClient http;
      Serial.print("[HTTP] begin...\n");
      int dataN [1000];
      String gyrox1 = "";
      for (int i = 0; i < 1000; i++) {
        M5.MPU6886.getGyroData(&gyroX, &gyroY, &gyroZ);
        float aarr = gyroX;
        dataN[i] = aarr * 100;

      }
      for (int i = 0; i < 1000; i++) {
        gyrox1 = gyrox1 + dataN[i];
        if (i < 999) {
          gyrox1 = gyrox1 + ",";
        }
      }

      // connection au serveur
      M5.Lcd.fillScreen(GREEN);
      http.begin("http://192.168.10.103:5000/taha/" + gyrox1); //HTTP
      Serial.print("[HTTP] GET...\n");

      Serial.println( gyrox1);
```

```
      int httpCode = http.GET();

      if (httpCode > 0) {

        Serial.printf("[HTTP] GET... code: %d\n", httpCode);
        // file found at server
        if (httpCode == HTTP_CODE_OK) {
          String payload = http.getString();
          Serial.println(payload);
        }
      } else {
        Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(
httpCode).c_str());
      }
      http.end();
      M5.Lcd.fillScreen(BLACK);
      M5.Axp.ScreenBreath(0);
    }
  }

  M5.update();
}
```

IV.3.2.Le code d'acquisition pour M5SttickC avec microphone

```
#include <Arduino.h>
#include <M5StickC.h>
#include <WiFi.h>
#include <WiFiMulti.h>
#include <HTTPClient.h>

WiFiMulti wifiMulti;
#include <M5StickC.h>

#include <driver/i2s.h>

#define PIN_CLK  0
#define PIN_DATA 34
#define READ_LEN (2048)
#define GAIN_FACTOR 3
uint8_t BUFFER[READ_LEN] = {0};
int16_t *adcBuffer = NULL;


unsigned long previousMillis = 0;


void i2sInit()
{
  i2s_config_t i2s_config = {
    .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX | I2S_MODE_PDM),
    .sample_rate =  10000,
    .bits_per_sample = I2S_BITS_PER_SAMPLE_16BIT, // is fixed at 12bit, stereo
, MSB
    .channel_format = I2S_CHANNEL_FMT_ALL_RIGHT,
```

```
    .communication_format = I2S_COMM_FORMAT_I2S,
    .intr_alloc_flags = ESP_INTR_FLAG_LEVEL1,
    .dma_buf_count = 2,
    .dma_buf_len = 128,
  };
  i2s_pin_config_t pin_config;
  pin_config.bck_io_num   = I2S_PIN_NO_CHANGE;
  pin_config.ws_io_num    = PIN_CLK;
  pin_config.data_out_num = I2S_PIN_NO_CHANGE;
  pin_config.data_in_num  = PIN_DATA;
  i2s_driver_install(I2S_NUM_0, &i2s_config, 0, NULL);
  i2s_set_pin(I2S_NUM_0, &pin_config);
  i2s_set_clk(I2S_NUM_0, 10000, I2S_BITS_PER_SAMPLE_16BIT, I2S_CHANNEL_MONO);
}

void mic_record_task (void* arg)
{
  size_t bytesread;
  while (1) {
    i2s_read(I2S_NUM_0, (char*) BUFFER, READ_LEN, &bytesread, portMAX_DELAY);
    adcBuffer = (int16_t *)BUFFER;
    vTaskDelay(100 / portTICK_RATE_MS);
  }
}

void setup() {

  M5.begin();
  M5.MPU6886.Init();
  M5.Axp.ScreenBreath(0);

  Serial.begin(115200);

  for (uint8_t t = 4; t > 0; t--) {
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();
  }
  wifiMulti.addAP("L", "ASCII@45");
  i2sInit();
  xTaskCreate(mic_record_task, "mic_record_task", 2048, NULL, 1, NULL);

}

void loop() {
  vTaskDelay(1000 / portTICK_RATE_MS); // otherwise the main task wastes half
of the cpu cycles
  if ((wifiMulti.run() == WL_CONNECTED)) {

    if (   M5.BtnA.isReleased() == 0) {
      M5.Axp.ScreenBreath(100);
      M5.Lcd.setTextSize(4);
      M5.Lcd.setCursor(20, 70);
      delay(1000);
      M5.Lcd.println("2");
      M5.Lcd.setCursor(20, 70);
      delay(1000);
```

```
      M5.Lcd.println("1");
      delay(1000);
      M5.Lcd.fillScreen(WHITE);



      HTTPClient http;
      Serial.print("[HTTP] begin...\n");
      int dataN [1000];
      String gyrox1 = "";
      int i = 0;

      while (i < 1000) {

        unsigned long currentMillis = millis();

        if (currentMillis - previousMillis >= 1 ) {
          // save the last time you blinked the LED
          previousMillis = currentMillis;

          dataN[i] =  adcBuffer[i];
          i++;
        }
      }

      for (int i = 0; i < 1000; i++) {
        gyrox1 = gyrox1 + dataN[i];
        if (i < 999) {
          gyrox1 = gyrox1 + ",";
        }
      }

      M5.Lcd.fillScreen(GREEN);
      http.begin("http://192.168.10.107:5000/taha/" + gyrox1); //HTTP
      Serial.print("[HTTP] GET...\n");

      Serial.println( gyrox1);
      int httpCode = http.GET();

      if (httpCode > 0) {

        Serial.printf("[HTTP] GET... code: %d\n", httpCode);
        // file found at server
        if (httpCode == HTTP_CODE_OK) {
          String payload = http.getString();
          Serial.println(payload);
        }
      } else {
        Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(
httpCode).c_str());
      }
      http.end();
      M5.Lcd.fillScreen(BLACK);
      M5.Axp.ScreenBreath(0);
    }
  }
```

```
  M5.update();
}
```

IV.4. La mise en marche

Avant de voir la mise en marche, il faudra expliquer un point essentiel dans un système à base d'objet connecté, le fait d'ajouter ou de supprimer un objet ne nuit pas à la stabilité du système, ce qui rend l'expansion le changement ou la suppression inoffensive.

On entre dans la mise en marche avec l'allumage du serveur, après l'exécution du code et la mise en réseau de serveur, on retrouve l'adresse IP que nous utiliserons pour permettre aux objets connectés de contacter le serveur.

```
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://192.168.10.200:5000/ (Press CTRL+C to quit)
```

Figure 24 le lancement du serveur Python/FLASK

Après avoir lancé le serveur, on allume les M5StickC pour qu'ils se connectent directement au serveur, puis on les place sur la machine souhaitée pour une analyse afin permettre une acquisition vibratoire ou sonore sans avoir à rajouter des modules.

Le M5StickC se colle directement sur les parois métalliques puisqu'un aimant intégré se retrouve à l'intérieur pour permettre de l'utiliser sans attache.

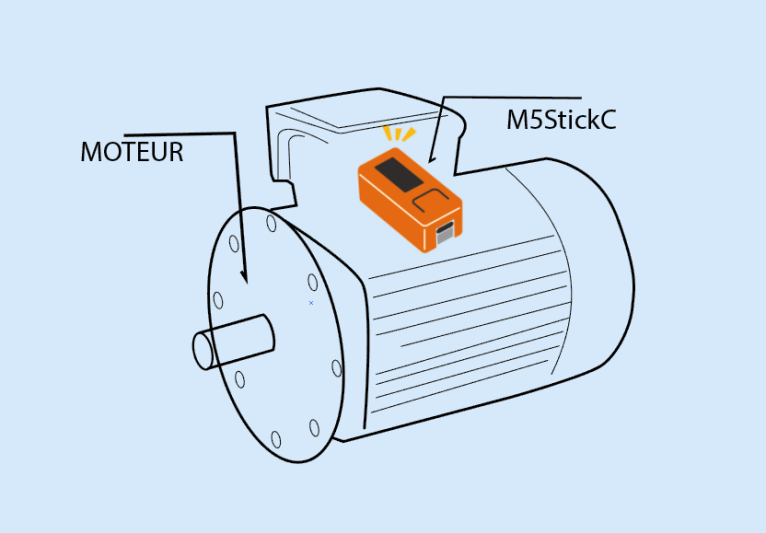


Figure 25 M5StickC collé à un moteur

Après avoir collé le M5StickC, il commence à envoyer des échantillons de capture vibration ou sonore au serveur, l'écran s'allume en blanc pour informer qu'il est en mode acquisition.

Figure 26 le mode acquisition de M5StickC

Après la fin de l'acquisition, une couleur verte s'allume dans l'envoi des informations au serveur.

Figure 27 le mode d'envoi de M5StickC

Du côté du serveur, le traitement procèdera automatiquement après l'envoi avec l'apparition de la Transformée de Fourier automatiquement à l'écran.

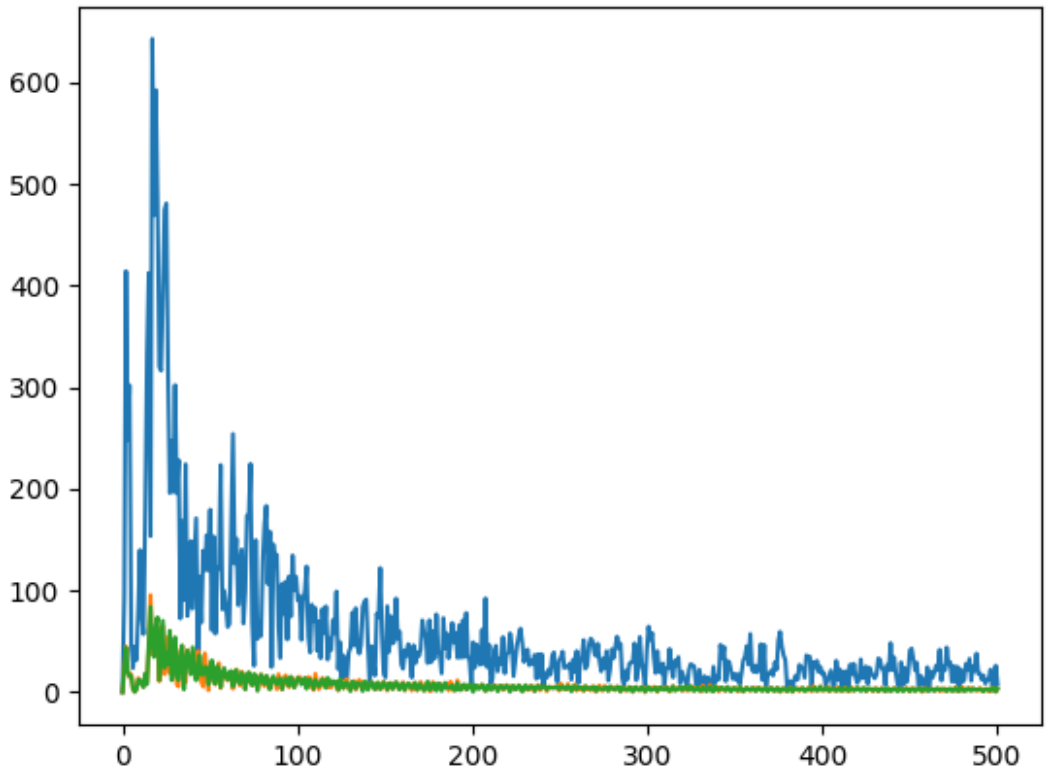


Figure 28 graphe de la transformé de fourier après analyse

On peut faire en sorte que s'il y a un excès dans une fréquence ou si les fréquences du moteur ont changé, un email ou un sms sera envoyé afin de prévenir le chef du secteur de maintenance pour faire une analyse approfondie sur le moteur.

Conclusion générale :

La finalité de ce projet est la conception d'un système permettant la mise en place d'un contrôle non destructif préventif avec des objets connectés. Les connaissances acquises lors de notre parcours universitaire en électronique et en automatique ont pu faire de ce projet une réalité. Par le biais de plusieurs technologies, nos idées ont fait place à un nouveau concept.

Pour une bonne finition du projet, nous avons acquis des compétences dans le Python, la compréhension de l'architecture réseau des objets connectés, ainsi que la mise en place d'un serveur dans le cloud. Nous avons ajouté une flexibilité au système de maintenance lequel malgré sa simplicité peut être d'application difficile sans informations concrètes sur le défaut à régler.

En conclusion, grâce aux données récoltées lors de la mise en marche du serveur, cette conception peut être améliorée par l'ajout d'une couche d'intelligence artificielle. Cette dernière, une fois entraînée, détectera les débuts d'une défaillance, reconnaîtra la partie touchée et décidera s'il est nécessaire de faire une maintenance avant même l'apparition des indicateurs permettant la détection d'un défaut.

Bibliographie

[1] Cartz, Louis (1995). Nondestructive Testing.A S M International. ISBN 978-0-87170-517-4.

[2] Bridges, Andrew. "High Speed Cameras for Non-Destructive Testing". NASA TechBriefs. Retrieved 1 November 2013.

[3] Joseph M. Buckley. "An Introduction to Eddy Current Testing Theory and Technology", retrieved July 1, 2015

[4] Blitz, Jack; G. Simpson (1991). Ultrasonic Methods of Non-Destructive Testing. Springer-Verlag New York, LLC. ISBN 978-0-412-60470-6.

[5] Chauhuri, Abhik (2018). Internet of Things, for Things, and by Things. Boca Raton, Florida: CRC Press. ISBN 9781138710443.

[6] Minteer, A. (2017). "Chapter 9: Applying Geospatial Analytics to IoT Data". Analytics for the Internet of Things (IoT). Packt Publishing. pp. 230–57. ISBN 9781787127579.

[7] Techbase Series (2005) "Energy-ecient ESP32-based Industrial Automation Controller " .moduino.techbase.eu

[8] "OGC SensorThings API". Open Geospatial Consortium. Retrieved 2018-02-20 , Standard in IoT approved as official OGC standard

[9] "History and License". Retrieved 5 December 2016. "All Python releases are Open Source" https://docs.python.org/3/license.html

[10] "PEP 373 -- Python 2.7 Release Schedule". Python.org. Retrieved 22 September 2019. https://www.python.org/dev/peps/pep-0373/

[11] "Web Services Architecture". World Wide Web Consortium. 11 February 2004. 3.1.3 Relationship to the World Wide Web and REST Architectures. Retrieved 29 September 2016.

[12] Erl, Thomas; Carlyle, Benjamin; Pautasso, Cesare; Balasubramanian, Raj (2012). "5.1". SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST. Upper Saddle River, New Jersey: Prentice Hall. ISBN 978-0-13-701251-0.