

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : TECHNOLOGIE

Département : ELECTRONIQUE

Domaine : SCIENCES ET TECHNIQUES

Filière : Automatique.

Spécialité : Automatique et informatique industrielle.

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème :

Contrôle d'une maison intelligente avec Raspberry-Pi

Présenté par : *Sebti Heithem*

Bouregba Fatima

Encadrant : *Bekaik Mounir*

MCB

Université Badji Mokhtar-Annaba

Jury de Soutenance :

Khaldouna Zahia	Prof	Université Badji Mokhtar Annaba	Président
Bekaik Mounir	MCB	Université Badji Mokhtar Annaba	Encadrant
Ramdani Mesaoud	Prof	Université Badji Mokhtar Annaba	Examineur

Année Universitaire : 2021/2022

Résumé

Ce projet est présenté pour l'obtention du diplôme de master en Automatique et Informatique Industrielle, il s'agit de réaliser une maison intelligente à base de Raspberry Pi. Notre travail rentre dans le cadre de conception d'une solution technologique pour assurer la sécurité des maisons et des habitants, on les incitant à maîtriser non seulement le contrôle de leur maison à distance mais aussi prendre les mesures les plus rapides et appropriés pour résoudre tout événements indésirables.

Dans ce sens, notre solution porte sur l'implémentation d'un control à distance à base de Raspberry Pi des paramètres physiques afin d'assurer la sécurité de la maison ainsi que les individus. Notre objectif est d'être efficace face à des situations dangereuses telles que : incendie, fuite de gaz, intrus. Pour ce faire, nous avons proposé des cahiers des charges pour chaque cas à l'aide des Grafjets et Ladder, puis la programmation des solutions via Raspberry-pi à l'aide de Python.

Mots-clés : Reconnaissance Faciale, Raspberry Pi, Python, Grafjet, Ladder, contrôle

Abstract

This project presented for obtaining the master's degree in Automation and Industrial data consisted of creating a smart house based on Raspberry Pi. Our work falls within the framework of designing a technological solution to ensure the safety of homes and residents, encouraging them to control not only their home but also to take the fastest and most appropriate measures to resolve any adverse events.

In this way, our solution concerns the implementation of a distance control based on Raspberry Pi of the physical parameters in order to ensure the security of the house and the individuals. Our goal is to be effective in the presence of dangerous situations such as : fire, gas leak, intruder. To do this, we proposed specifications for each case using Grafjet and Ladder, then we programme the solutions via Raspberry-pi using Python.

Keywords: Facial Recognition, Raspberry Pi, Python, Grafjet, Ladder, control.

ملخص

يتكون هذا المشروع المقدم للحصول على درجة الماجستير في الأتمتة والبيانات الصناعية من إنشاء منزل ذكي يعتمد على Raspberry Pi.

يندرج عملنا في إطار تصميم حل تقني لضمان سلامة المنازل والمقيمين ، وتشجيعهم على التحكم ليس فقط في منازلهم ولكن أيضاً لاتخاذ الإجراءات الأسرع والأنسب لحل أي أحداث سلبية.

للمعايير المادية من أجل ضمان أمن المنزل والأفراد. Raspberry Pi بهذه الطريقة ، يتعلق حلنا بتنفيذ التحكم عن بعد استناداً إلى هدفنا أن نكون فعالين في وجود المواقف الخطرة مثل: حريق ، تسرب غاز ، دخيل. للقيام بذلك ، اقترحنا مواصفات لكل حالة Python باستخدام Raspberry-pi ، ثم نقوم ببرمجة الحلول عبر Ladder و Grafjet باستخدام

Remerciement

Nous voulons d'abord remercier le Dieu tout-puissant ET miséricordieux, Cela nous a donné la force et la patience pour accomplir Ce modeste travail.

Nous remercions également notre entraîneur **Mr BEKAIK MOUNIR** pour ses bonnes déclarations qui ont éclairé la voie de la recherche et de sa collaboration avec nous dans la réalisation de cette modeste œuvre.

Nous tenons également à remercier chaleureusement les membres du jury pour l'intérêt que vous portez à nos recherches, en acceptant d'examiner notre travail et d'enrichir vos propositions.

Enfin, nous remercions tous ceux qui ont contribué directement ou indirectement à ce travail.

Dédicace

Je me consacre à ce travail

À mon père monsieur BOUREGBA MOHAMED

Aucune dévotion ne peut exprimer amour, estime, dévouement et respect rien ne vaut les efforts faits jour et nuit pour mon éducation et mon bien-être.

Ce travail est le fruit de tes sacrifices pour moi Grâce à ce travail, vous recevrez ma gratitude et sentiments profonds.

Qu'Allah le Tout-Puissant vous protège et vous garde dans son vaste paradis

À ma chère mère Madame BRICHNI AKILA

Pour me donner la vie et la joie ses bénédictions n'ont jamais manqué. La source de la tendresse et l'exemple de la dévotion qui n'a cessé de m'encourager et de prier pour moi. La lumière de mes jours, la source de mes efforts, la flamme de mon cœur, de ma vie et de mon bonheur, la mère que j'aime.

Qu'Allah le tout puissant soit à vos coté.

À ma sœur Chaima et mon frère ali

Qu'Allah vous procure bonne santé et longue vie.

À Mon Collègue du Binôme et Mon frère haithem

Pour Sa Patience et Sa Compréhension.

À mon cher ami et compagnon Manar

En témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passés ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur.

À tous mes camarades de classe sans exception AII Master II Et surtout mon cher collègue qui m'a aidé HACHEM BAYADHI.

FATIMA

Dédicace

À celui qui a sacrifié toute sa vie pour nous, mon symbole de développement et qui m'a donné une bonne éducation. Pour tout cela je dédie ce modeste mémoire à mon Cher PERE
Mr **Youcef**

À ma très chère et tendre MERE la lumière de ma vie qui a veillée jour et nuit sur mon éducation, qui a été présente à chaque instant pour moi, qui m'a toujours en courage et qui m'a soutenu durant toute la longue durée de mes études Ma MERE Mme **Khlefatifairouze**

Je dis à mes parents aujourd'hui merci et mille fois merci et Dieu ALLAH vous Protèges

À mes sœur **Roufaida et haifa et Ritedj**

Qu'Allah le tout puissant soit à vos côtés.

Tous mes remerciements vont également à mes meilleurs amis Rachid, Badr Eddine, Riad Hachmene.

À tous mes camarades de classe sans exception AII Master II.

À tous ceux qui auront l'occasion de lire cet ouvrage.

HAITHEM

Liste des figures

Figure 1:exemple de système domotique d'une maison.	3
Figure 2:détecteur de gaz.	4
Figure 3:détecteur de température.	4
Figure 4:Le chiffre d'affaire dans une sélection de pays en 2017.	8
Figure 5: exemple de l'entreprise « poly-automatic ».	9
Figure 6: Contrôler les lumières des chambres.	9
Figure 7:Contrôler de verrous.	10
Figure 8: Jardins.	11
Figure 9:Raspberry pi.	13
Figure 10 : Le composant d'un Raspberry pi.	14
Figure 11: Les portes GPIO d'une carte Raspberry-pi.	15
Figure 12 : Fond d'écran Raspberry.	16
Figure 13:Module de LEDs.	17
Figure 14:Branchement d'une LED.	18
Figure 15:Module DHT 11.	19
Figure 16: Module LCD 1602 A (V2.0).	19
Figure 17: Module I2C.	20
Figure 18:Module I2C. Branchement DHT11.	20
Figure 19: Servo moteur.	22
Figure 20:Un train d'impulsion de période 20 ms avec une rotation.	23
Figure 21: Branchement du Servomoteur.	25
Figure 22:Gaz Sensor MQ2.	26
Figure 23: Module du MCP3008.	27
Figure 24: branchement du capteur MQ2.	27
Figure 25:Moteur pas à pas.	29
Figure 26: Module ULN 2003.	30
Figure 27: Circuit du moteur pas à pas.	30
Figure 28:Caméra Surveillance.	32
Figure 29 : Branchement Générale.	33
Figure 30:Architecteure d'un réseau de neurone convolutif.	35
Figure 31:Exemple illustrant l'opération de convolution 3x3.	36
Figure 32:Exemple de zéro padding.	37
Figure 33: Exemple simpliste des valeurs du pixel 5x5.	37
Figure 34:Exemple de valeur d'une matrice utilisée comme filtre.	37
Figure 35:Démonstration de convolution.	38
Figure 36:Résultat attendu après une phase de convolution.	38
Figure 37: Représentation graphique d'un perceptron.	39
Figure 38: Fonction Relu.	40
Figure 39: Fonction Sigmoidé.	40
Figure 40: Fonction Soft max.	41

Figure 41: Fonction Tanh.	41
Figure 42:Exemple de calcul du pooling sur une image 4×4.	43
Figure 43 : grafcet de gaz	49
Figure 44: le capteur est détecté.....	52
Figure 45 : Allumé la led.....	52
Figure 46 : ouverture la fenêtre.....	53
Figure 47 : envoyé le message.....	53
Figure 48 : Le cas de non détection de gaz.....	53
Figure 49 : éteindre la led.....	54
Figure 50 : grafcet de température.....	55
Figure 51 : affiche la température.....	58
Figure 52: le message envoyé pour la température.....	59
Figure 53 : allume la led.....	59
Figure 54 : éteindre la led.....	60
Figure 55 : grafcet de la Cam	61
Figure 56 : la Cam détecté la facial Haithem.....	66
Figure 57 : la Cam détecté facial inconnue.....	66
Figure 58 : ouverture de la porte.....	66
Figure 59 : enregistrement des vidéo de camera surveillance.....	67
Figure 60 : le montage général du notre maison intelligente	67

Liste des abréviations

LED: light-emitting diode

GPIO: General –purpose input/output

VNC:Virtual Network Computing

CNN: Convolution neural network

FC: Fully Connected

LCD: Liquide-Crystal display

SDA: Serial Data Line

SCL: Serial Clock Line

VDD: Voltage Drain Drain

GND: Ground

PWM: Pulse –width modulation

Wi -Fi: Wireless Fidelity

ADC:analog-to-digital converter

RGB: Red, green, Blue

CSI: Camera Serial Interface

USB: Universal Serial Bus

IDE: Integrated Development System

OS:Operating System

OSI: Open System Interconnexion

LOT: Internet of Things

DSI: Direction systeminformatique

I2C:Inter Integrated Circuit Bus

VCC : Voltage Common Collector

Liste du tableau :

I.1 tableau d'analyse	52
-----------------------------	----



Sommaire

Abstract	
Liste des figures	
Liste des abréviations	
Liste du tableau	
Introduction générale	1
Chapitre 1: Présentation	1
1.1 Généralités sur la Smart Home	3
1.2 Paramètres à contrôler	3
1.3 L'enjeu du démotique	5
1.4 Concepts de recherche	6
1.5 Construction d'un système d'automatisation domestique	6
1.6 Marché des maisons intelligentes	7
1.6.1 Les statistiques	7
1.6.2 Autonomie domestique en Algérie	8
Chapitre2 : Instrumentation	12
2. Présentation Raspberry pi	13
2.1 Définition	13
2.2 Caractéristiques Principales	14
2.3 Les Portes GPIO	15
2.4 Python	15
2.5 Le matériel utilisés	17
2.5.1 LED	17
2.5.2 Le DHT11	18
2.5.3 L'écran LCD	19
2.5.4 Bus I2C	19
2.5.5 Servomoteur	22

2.5.6 Gaz SensorMQ2	26
2.5.7 MCP3008	26
2.5.8 Les moteurs pas à pas	29
2.5.9 Pilote de moteur ULN2003	30
2.5.10 Caméra Surveillance	32
Chapitre 3 Réseaux de neurone convolutif (CNN)	34
3 Réseaux de neurone convolutif (CNN)	35
3.1 Définition	35
3.2 Architecteur de réseaux de neurones convolutif	35
3.3 La couche de convolution	36
3.4 Le perceptron	38
3.5 Structure de perceptron	39
3.6 La Fonction d'activation	40
3.7 La couche de correction Relu	41
3.8 La couche de pooling	42
3.9 Couche entièrement connectées (Fully Connected (FC))	43
3.10 Liste des fonctions de coût utilisées dans les réseaux de neurones	44
3.11 Code python	44
Chapitre 4 : Simulation et résultat	47
4.1 Tableau d'analyse	48
4.2 Explication les informations (Réceptivité) et les capteurs	48
4.3 Résultat de gaz	49
4.3.1 Cahier de charge	49
4.3.2 Grafctet	49
4.3.3 Ladder	49
4.3.4 Résultat finale	51
4.4 Résultat de température	54
4.4.1 Cahier de charge	54
4.4.2 Grafctet.....	54
4.4.3 Ladder	55
4.4.4 Résultat finale	58
4.5 Résultat de reconnaissance facial	60
4.5.1 Cahier de charge	60
4.5.2 Grafctet	60
4.5.3 Ladder	61
4.5.4 Résultat finale	65

4.6 Résultat de camera surveillance	67
Conclusion générale	68
Bibliographie	69



Introduction générale :

Tout le monde rêve d'avoir une maison encore plus confortable, une bonne température dans chaque pièce, un bon éclairage, une surveillance à distance...

Dans l'automatisation domestique, de nombreuses fonctions sont gérées par un Smartphone ou une tablette, ce qui offre un contrôle permanent à distance afin d'assurer le confort et garantir une meilleure sécurité.

L'automatisation domestique met l'accent sur deux aspects : la sécurité et le confort des habitants. Elle ouvre non seulement de nouvelles possibilités dans le domaine de l'automatisation domestique, de plus, c'est un moyen offert aux individus pour gérer leur environnement.

Notre projet est dédié à offrir une solution basée sur la carte Raspberry Pi pour construire un prototype de Smart Home offrant en permanence des solutions simples dans le domaine de l'automatisation domestique.

Nous commencerons dans un premier lieu par une vue d'ensemble de l'automatisation domestique et de ses fonctions et des différents types de technologies utilisées.

Ensuite, dans le deuxième chapitre nous présentons exhaustivement les différents composants de Raspberry PI ainsi que les différents capteurs et actionneurs utilisées dans notre solution. Pour ce faire, nous avons procédé au montage de chaque partie indépendamment de l'autre avec un test du programme d'exécution.

Dans le troisième chapitre, nous avons évoqué l'algorithme CNN fortement utilisé dans le domaine de l'intelligence artificielle où une étude détaillée de la méthode de détection faciale est montrée.

Afin de donner le résultat de notre solution, nous avons présenté des différents cas d'utilisation de notre carte électronique, pour chaque scénario nous avons donné son diagramme de Grafcet ainsi que son Ladder afin d'expliquer l'utilité de chaque capteurs et actionneurs pour résoudre les différents problèmes existants. Nous illustrons l'efficacité de notre solution avec des tests et des figures des différents cas d'utilisations.

Chapitre 1 : Présentation

Chapitre 1 : Présentation

L'idée de créer une maison automatisée est née à la fin du XIXe siècle : son histoire est longue ! En fait, en 1803, ils imaginaient une maison dans laquelle il y aurait du courant dans toutes les pièces : il suffirait d'appuyer sur un interrupteur pour allumer la lumière. Mais cette maison idéale ne cesse de se développer : aujourd'hui, c'est une maison non seulement automatisée, mais aussi intelligente ! Cette maison est présente dans la littérature et les films de science-fiction depuis de nombreuses années. Pendant de nombreuses années, des chercheurs et des ingénieurs ont fait de cette maison une réalité.

1.1 Généralités sur la Smart Home :

Smart Home ou Domotique, nous en parlons beaucoup, mais donnent-ils la même définition à ce mot ? Ce terme a-t-il toujours été révélateur d'erreurs ? Comment l'évolution des possibilités offertes par les acteurs de ce secteur a-t-elle évolué ?

Une maison devient une maison intelligente lorsqu'elle est équipée d'un réseau de communication dépendant de différents produits ou services électroniques entre eux pour permettre une télécommande ou une surveillance. Ce sont des objets différents : ampoules, détecteurs de gaz, stores, température, ... Grâce à la plateforme connectée, il est possible de contrôler tous les objets à partir d'un seul point.

La Smart Home vise à proposer des solutions techniques répondant aux exigences de confort (gestion de l'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (télécommandes, signaux visuels ou acoustiques, etc.)



Figure 1: exemple de système domotique d'une maison.

1.2 Paramètres à contrôler :

Détecteur de gaz : un détecteur de gaz est un appareil qui détecte la présence de gaz dans une zone, souvent en tant que partie d'un système de sécurité. Un détecteur de gaz peut

Chapitre 1 : Présentation

déclencher une alarme pour les opérateurs dans la zone où se produit la fuite, ce qui leur donne une chance de descendre.



Figure 2 : détecteur de gaz.

Traitement de l'image : une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations dans le but d'améliorer leur qualité ou d'extraire des informations.

Détecteur de température : il s'agit d'un type de capteur de température dont l'une des principales caractéristiques est qu'il fonctionne dans différentes conditions, qu'il est protégé contre les explosions, qu'il est facile à installer et qu'il s'ajuste à une température donnée, Vous pourrez ainsi contrôler la température de votre maison où que vous soyez.

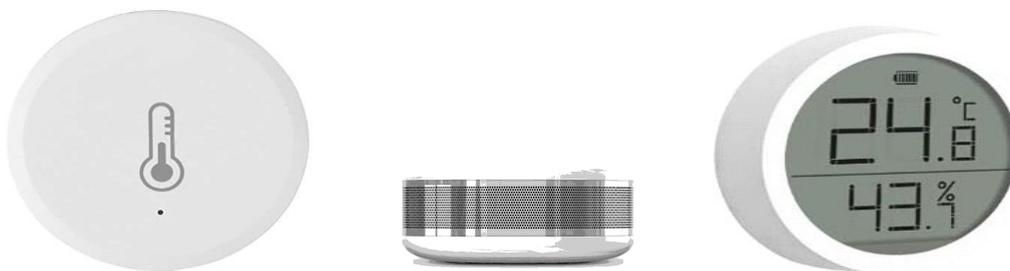


Figure 3 : détecteur de température.

Les solutions proposées :

La Smart Home est l'endroit où chaque endroit peut être commandé numériquement. Votre sonnette, votre éclairage, votre sécurité, le chauffage, le gaz, l'eau...

Ça les rassure aussi, pour qu'ils puissent être sûrs que tout fonctionne de manière optimale. Le système de surveillance intégré de la Smart Home vous permet par exemple de surveiller ce qui se passe partout. Vous pouvez aussi acheter des systèmes qui protègent votre maison des

Chapitre 1 : Présentation

dangers du feu, du gaz et de la température, ou ouvrir la porte avec le visage. Une Smart Home vous permet d'avoir plus de contrôle et de suivre ce qui se passe dans votre maison.

1.3 L'enjeu du démotique :

L'avantage :

- L'avantage principal de l'automatisation domestique est d'améliorer le quotidien domestique en termes de confort, de sécurité et de gestion de l'énergie.
- Ce type d'équipement simplifie votre vie et optimise votre confort en adaptant votre intérieur à différents scénarios du quotidien.
- Il vous permet d'éteindre tous vos appareils électriques et de régler l'alarme lorsque vous quittez votre maison pour régler l'atmosphère de la lumière (atmosphère de lecture, atmosphère de détente avec lumière tempérée) pour vous réveiller dans un environnement climatisé, dans lequel le café est déjà prêt, commencez par verser ou ouvrez automatiquement les stores chaque matin.
- L'automatisation domestique économise également de l'énergie en commandant automatiquement le chauffage, la climatisation et l'éclairage et en planifiant les appareils en dehors des heures de pointe.
- Il présente l'avantage d'améliorer la sécurité par des alarmes, des systèmes automatiques d'ouverture des portes (reconnaissance vocale, carte magnétique, etc.).
- En cas de tentative d'effraction dans la maison, un appel téléphonique automatisé du propriétaire ou de l'entreprise peut être contacté pour l'assurance.
- Enfin, ces différentes technologies constituent une aide précieuse pour les personnes handicapées.

Sécurité : la technologie Smart Home peut vous aider à protéger votre domicile en évitant les dégâts inattendus tels que les inondations ou les incendies. Les capteurs d'eau détectent les fuites du lave-linge et du lave-vaisselle et alertent afin d'éviter d'éventuels dommages au design de votre intérieur. Et les avertisseurs d'incendie envoient une notification en cas de fumée ou de surchauffe, qui peut également préserver votre culture de l'habitat.

Confort : les technologies de la maison intelligente offrent aux utilisateurs un maximum de confort, de luxe et de détente physique et psychique. Il y a d'importants avantages à vivre dans une maison intelligente, qui est le plaisir d'un repos physique et psychique dans la vie quotidienne, où vous pouvez programmer des sources de lumière, de la sécurité, du chauffage et de la ventilation, ainsi que d'autres avantages qui facilitent vos besoins quotidiens, et vous pouvez gérer à distance de nombreux éléments de votre maison et apporter les modifications souhaitées à tout ce qui a été sélectionné pour le système Smart Home. Ce système moderne vous aide également à décider ce qui est le plus important pour vous, puis ajoute des outils supplémentaires si vous le souhaitez. Et la meilleure nouvelle, c'est que ce n'est pas aussi cher que vous le pensiez.

Chapitre 1 : Présentation

1.4 Concepts de recherche :

L'intelligence artificielle (IA) : est une branche d'ordinateur qui s'intéresse à la création de machines et d'appareils intelligents. Il est aussi défini comme l'imitation du cerveau humain dans la performance de certaines de ses fonctions complexes.

L'informatique ubiquitaire : est un paradigme dans lequel le traitement de l'information est lié à chaque activité ou objet trouvé. Il comprend l'interconnexion de dispositifs électroniques, y compris l'intégration de microprocesseurs pour la communication d'informations.

L'intelligence environnementale : le concept d'intelligence environnementale est l'un des concepts techniques modernes qui ont suscité le plus d'intérêt ces dernières années. De nombreux pays développés ont commencé à utiliser des concepts d'intelligence environnementale et leurs applications.

Internet of Things (IOT) : un terme émergent fait référence à la nouvelle génération d'Internet (le réseau) qui permet la compréhension entre les dispositifs interconnectés (via le protocole Internet). Ces appareils Il s'agit d'outils, de capteurs, de divers outils d'intelligence artificielle et autres.

1.5 Construction d'un système d'automatisation domestique :

L'un des aspects les plus importants de la construction d'un modèle d'automatisation domestique est la réflexion sur les protocoles que vos appareils peuvent utiliser pour communiquer avec les passerelles et les serveurs. Et un des éléments les plus importants.

- Capteurs.
- Actionneurs.
- Gestion des interfaces.
- Unité de terrain.
- Protocoles de communication.

Les capteurs :

Un capteur peut être défini comme un appareil ou une unité qui aide à détecter toute variation d'une grandeur physique telle que la pression, la force ou une grandeur électrique telle que le courant ou toute autre forme d'énergie, telle qu'un capteur d'humidité et de température DHT11, MQ2 Señor, etc.

Actionneurs :

Un actionneur est un objet qui transforme l'énergie qui lui est fournie en un phénomène physique qui travaille, modifie le comportement ou l'état d'un système, ces actions pouvant agir sur certains éléments (portes, stores, etc.).

Chapitre 1 : Présentation

Interface pour le pilotage :

Il s'agit d'une série de matériels et de logiciels permettant à l'utilisateur de contrôler, de surveiller ou de surveiller une maison et ses occupants à distance ou à proximité. Exemples : Smartphone, tablette, télécommande, etc.

Unité de terrain :

L'unité de terrain joue le rôle d'une « intelligence centralisée » qui contient relativement peu d'objets mais peut être exigeante. Il se compose de deux éléments principaux :

- **Unité centrale** : c'est une carte de développement qui ressemble en fait à un micro-ordinateur avec des connexions, par exemple Raspberry pi.
- **Logiciel d'automatisation domestique** : un micro-ordinateur a besoin d'un logiciel dont vous avez besoin pour le traitement des tâches, par exemple Jeedom.

Protocoles de communication :

La communication entre les équipes d'un système d'automatisation domestique est assurée par un protocole commun, un protocole étant alors une spécification de règles permettant l'échange entre différents appareils. Il y a deux types de protocoles :

- Protocole propriétaire : les produits sont disponibles dans un seul fabricant avec des spécifications fermées, exemples : Delta Dore X2D, PnP...
- Protocole standard : il s'agit le plus souvent d'une combinaison de plusieurs fabricants utilisant le même protocole, dont les spécifications sont disponibles, exemples : KNX, Zigbee, etc.

1.6 Marché des maisons intelligentes :

1.6.1 Les statistiques :

Afin d'illustrer l'évolution des technologies avancées, nous présentons dans ce rapport quelques statistiques sur la maison intelligente, telles que rapportées par le site web américain "Techjury".

- Les experts du secteur s'attendent à ce que les achats mondiaux de maisons intelligentes dépassent 475 millions de maisons d'ici 2022.
- 66% des consommateurs américains ont au moins un appareil intelligent chez eux.
- Les statistiques montrent que le marché mondial des homes intelligents atteindra 53,5 milliards de dollars d'ici 2022.
- Le marché américain de la domotique intelligente est l'un des plus rentables au monde. La taille du marché américain a augmenté d'environ 15% par rapport à l'année précédente, suivi par la Chine avec une part de marché de 12,9 milliards de dollars. Le Japon occupe la troisième place, avec un volume de marché de 4,9 milliards de dollars.

Chapitre 1 : Présentation

- On estime que les ménages dépenseront 19,4 milliards de dollars en systèmes de sécurité intelligents.
- Les ménages qui ont déjà mis en place des technologies de sécurité domestique intelligente dépenseront jusqu'à 19,4 milliards de dollars d'ici la fin de 2020 pour des systèmes de protection de leurs maisons, et ce chiffre doublera pour atteindre 35 milliards de dollars d'ici 2024.
- Le graphique suivant montre un classement d'une sélection de pays selon le chiffre d'affaires du marché de l'automatisation à domicile en 2017.

Recettes du marché de la domotique dans une sélection de pays en 2017 (en millions de dollars des Etats-Unis)

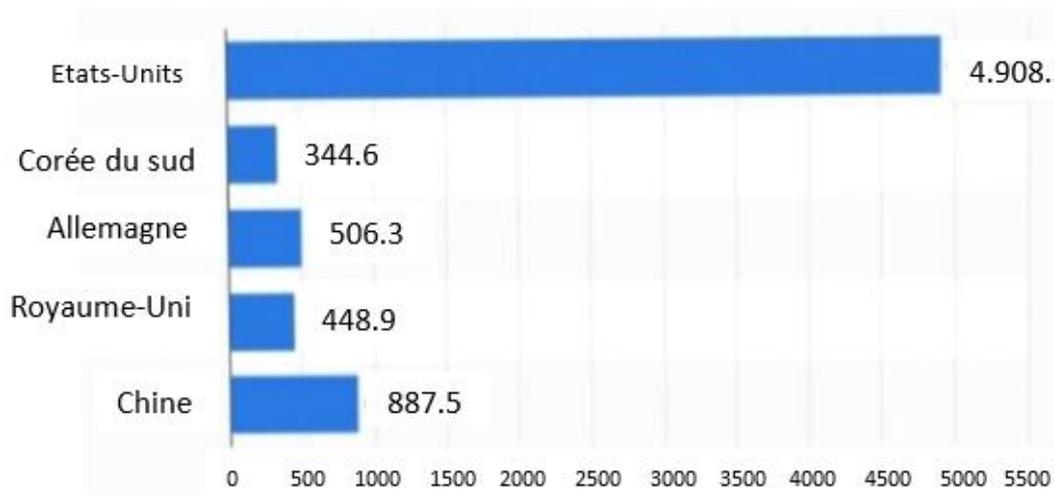


Figure 4 : Le chiffre d'affaire dans une sélection de pays en 2017.

1.6.2 Autonomie domestique en Algérie :

Le concept de Smart City a toujours suscité la controverse en Algérie, tant l'Autorité insiste sur la volonté et la détermination de faire de nos villes L'intelligence numérique, qui a besoin de technologies et de logiciels. Et avant cela, une culture de la citoyenneté et de l'acceptation du numérique. Les obstacles au développement de l'automatisation domestique en Algérie trouvent leur origine dans plusieurs raisons, dont :

- Faible demande, il s'agit donc de donner la priorité aux citoyens.
- L'environnement n'est pas adapté à tous les aspects de l'automatisation domestique.
- Manque d'entrepreneurs et d'investisseurs capables de mettre en œuvre ces projets.
- Manque de compétences pour travailler dans des technologies modernes et intelligentes.

D'autre part, certains pays arabes, comme les Émirats arabes unis et le Royaume d'Arabie saoudite, sont les leaders du monde arabe dans l'utilisation et la diffusion de ces technologies.

- **POLY-AUTOMATIC** : est une entreprise active dans le domaine Smart Home (Smart Home).

Chapitre 1 : Présentation

Avec l'apparition de nouvelles technologies de téléphones intelligents et de tablettes, les gens se sont habitués à tout contrôler et à accéder facilement à toutes les informations.

Leur objectif est d'intégrer ces technologies dans la vie quotidienne de nos Technologies disponibles.



Figure 5 : exemple de l'entreprise « poly-automatic ».

- **Le contrôle dans votre poche :**
 - **Contrôler les lumières des chambres**
 - Allumer et éteindre.
 - Contrôlez la luminosité.



Figure 6 : Contrôler les lumières des chambres.

Chapitre 1 : Présentation

- **Vidéo surveillance**
 - vidéo surveillance intégrée au système domotique via technologie IP
 - Intégrée au système domotique avec ouverture à distance
- **Allumer et éteindre la climatisation et chauffage**
 - Gestion numérique de la climatisation de tout type de marque.
 - Gestion de la température de chauffage individuel pour les maisons équipées de chauffage centralise.
- **Contrôle des verrous**
 - Détection d'ouverture de portes.
 - Installation de systèmes d'ouvertures motorisées pour volets roulants, garages, portiques et tous autres dispositif.
 - Ouverture des portes à distance.

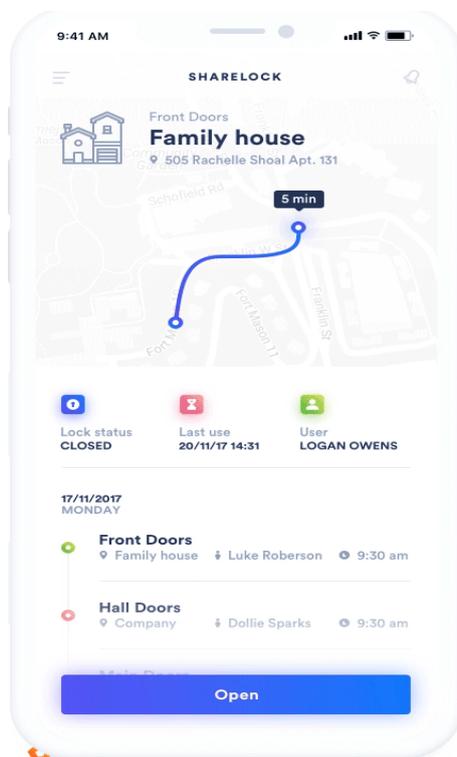


Figure 7 : Contrôler de verrous.

- **Notifications**
 - Détection d'ouverture de porte.
 - Relevé la température et de mouvement.
 - Détection d'inondation.

Chapitre 1 : Présentation

- Système d'alarme complètement intégrée à la domotique avec extension du réseau de capteurs.
- **Jardins**
 - Installation de systèmes d'arrosage jardins et fontaines pour maisons et collectivités.

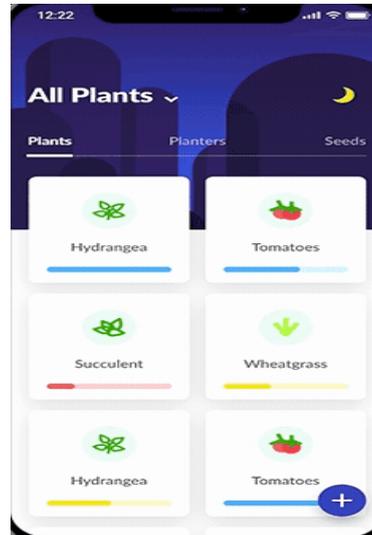


Figure 8 : Jardins.

Dans ce chapitre, nous avons montré l'importance du domaine de domotique dans la vie au quotidien et les différentes solutions existantes dans le marché. Nous avons aussi montré les différents capteurs et actionneurs utilisés dans ce contexte.

Chapitre2 : Instrumentation

Chapitre 2 : Instrumentation

Si nous voulons automatiser une application avec un micro-ordinateur ou un microcontrôleur, nous nous rendons compte rapidement qu'elle communique la plupart du temps avec le monde extérieur via ce qu'on appelle l'entrée/sortie de nom générique. Bien sûr, elles varient fortement d'une application à l'autre. Tous ces I/O pourraient indiquer l'importance du système, puisqu'ils permettent de saisir tous les paramètres d'un lieu donné dont nous voulons mesurer les constantes de cette diversité d'I/O, la plupart d'entre eux utilisent des diagrammes de base classiques et les programmes associés, que nous avons regroupés dans les autres chapitres.

2. Présentation Raspberry pi :

2.1 Définition :

Le Raspberry Pi est un mini-ordinateur de la taille d'une carte de crédit et un appareil aux possibilités infinies, où jeunes et moins jeunes peuvent découvrir le monde de la programmation.

Créez vos propres programmes, jouez à des jeux rétro, visionnez des films et des séries ou construisez une porte interactive ! Tout est possible avec le Raspberry Pi, la seule limite est votre imagination ! Il est équipé du nécessaire :

- Un microprocesseur ARM
- De la mémoire RAM
- Une carte vidéo
- Une carte Ethernet
- Le Wi-Fi
- Le Bluetooth



Figure 9 : Raspberry pi.

Chapitre 2 : Instrumentation

2.2 Caractéristiques Principales :

- Carte mère Raspberry Pi 3 type B+
- Processeur Broad ohm BCM2837B0 Cortex-A53 64-bit Soc @ 1,4 GHz
- RAM : 1 Go LPDDR2 SDRAM
- Connexions sans fil: Bluetooth 4.2 BLE, Wi-Fi Dual Band b/g/n/AC
- Connexion filaire : Gigabit Ethernet
- Décodage et encodage H.264 1080p30
- Support Poe (Poe HAT non fourni)
- Lecteur de carte micro-SD
- Header GPIO 40 broches
- Support du boot PXE
- Port caméra CSI pour connecter la caméra Raspberry Pi
- Port d'affichage DSI pour connecter l'écran tactile Raspberry Pi4 x USB 2.0
- Compatible avec les accessoires du Pi 3 Model B
- Alimentation 5V / 2.5A (transformateur non fourni)

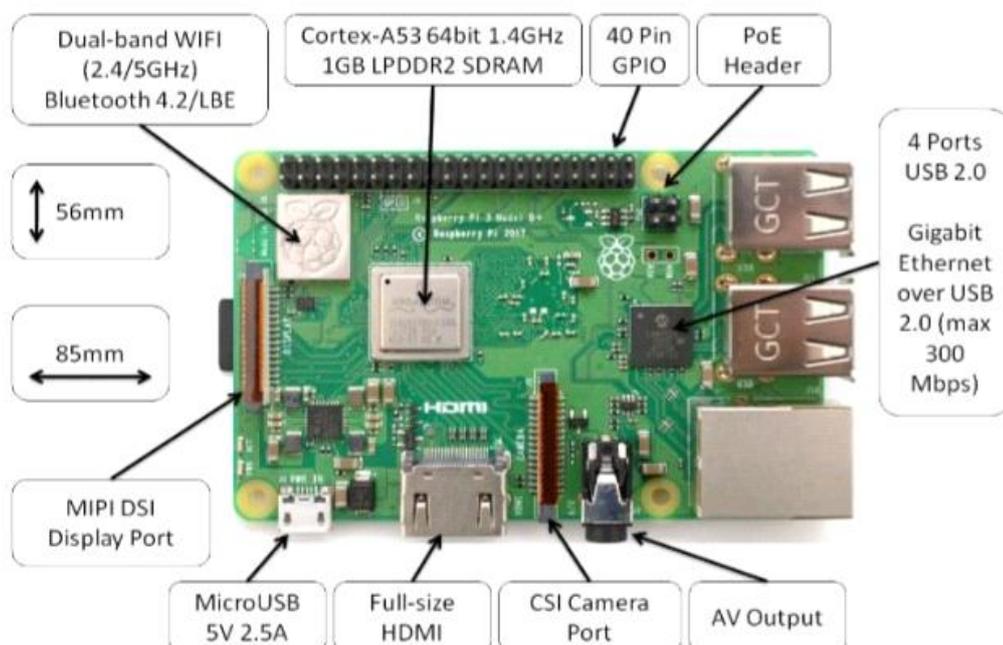


Figure 10 : Le composant d'un Raspberry pi.

Chapitre 2 : Instrumentation

2.3 Les Portes GPIO :

Tous les ports de la connexion GPIO ont un rôle bien défini, avec des blocs d'alimentation 5 V et 3,3 V et des masses.

Les ports GPIO sont identifiés par un numéro ; GPIO1, GPIO2, etc. Cette fonction n'empêche pas l'utilisation classique de la GPIO comme entrée/sortie numérique (0 ou 1).

Il y a 26 GPIO qui peuvent être utilisés comme I/O., qui fonctionnent sur tout ou rien, 0 ou 1, 0V ou 3,3V. Il n'y a pas de port analogique (tension variable continue). Si vous avez besoin de ports supplémentaires ou analogiques, vous devez ajouter des cartes d'extension.

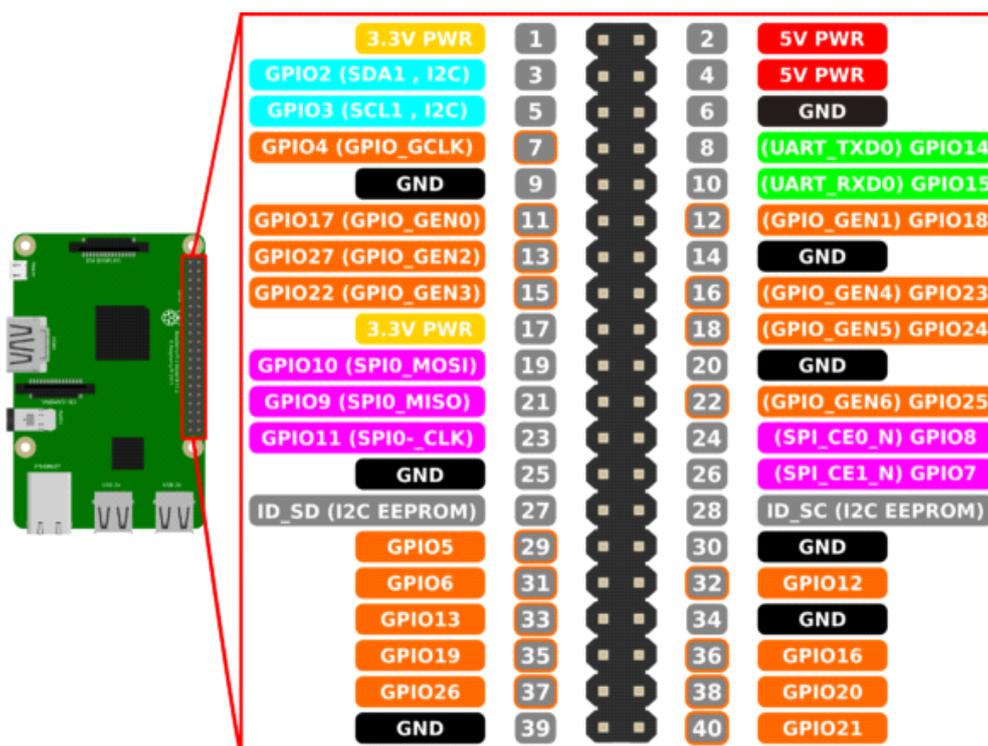


Figure 11 : Les portes GPIO d'une carte Raspberry-pi.

2.4 Python :

Un langage de programmation relativement accessible (par rapport à C ou C++, qui sont beaucoup plus complexes pour un débutant), mais encore très puissant et dont les possibilités sont énormes.

Programmation python à Raspberry Pi :

- Préparez votre pi :

Chapitre 2 : Instrumentation

Pour ce tutoriel, votre Raspberry Pi doit disposer de la dernière version de Raspberry Pi OS, qui contient Tony. Utilisons cet IDE facile à utiliser pour écrire notre premier programme Python sur notre Raspberry Pi.

- **Ouvrez Thonny Python IDE :**

Vous trouverez Thonny dans le menu des applications (coin supérieur gauche de votre écran), voir la figure ci-dessous. Si vous ne trouvez pas Thonny entre vos applications, vous n'avez probablement pas la dernière version de Raspberry Pi OS. Consultez notre tutoriel Comment mettre à jour votre Raspberry Pi avant de continuer.

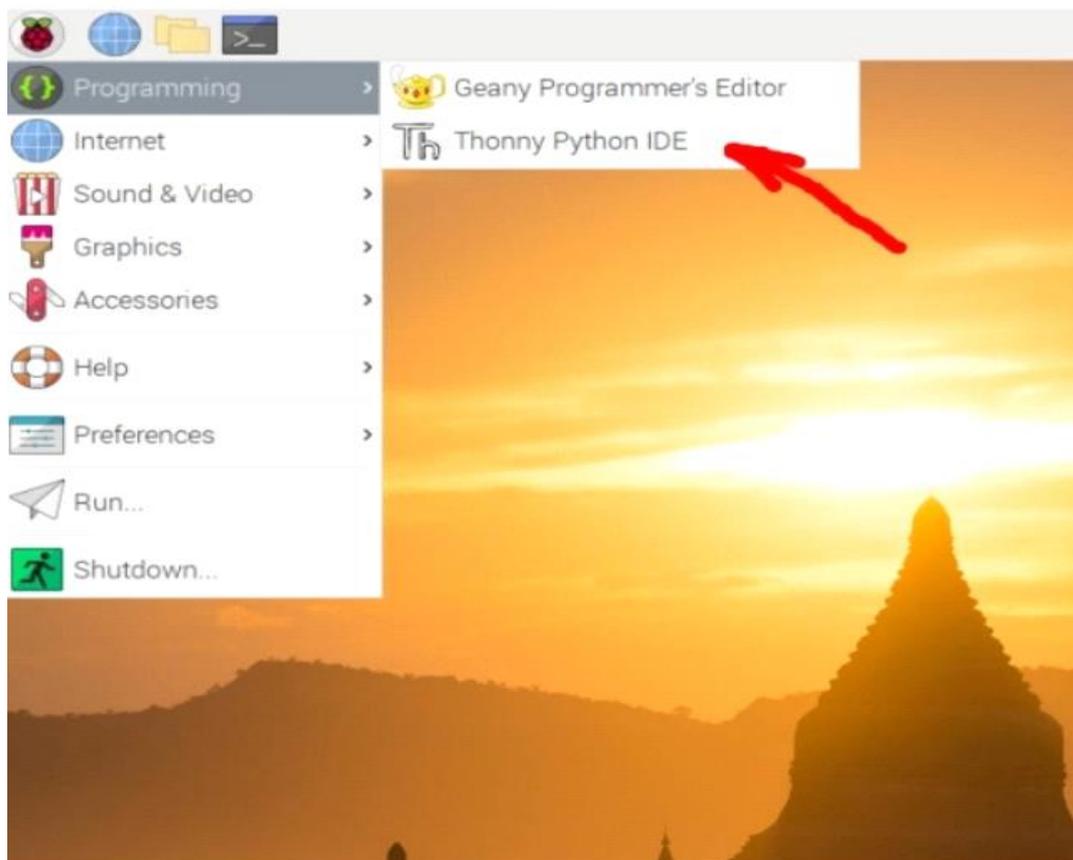


Figure 12 : Fond d'écran Raspberry.

- **Ecrivez le code :**

Au lieu du programme nous allons être légèrement plus ambitieux et choisissons un programme qui demandera votre nom. Votre nom sera sauvegardé en tant que variable et renvoyé sur l'écran immédiatement après.

- **Sauf gardez votre programme**

Avant de lancer le programme, vous devrez lui donner un nom et le sauvegarder. Dans notre cas, nous allons sauvegarder le fichier dans le dossier Documents et le nommer home. Py. Les fichiers Python ont l'extension py. Ici, vous n'avez pas besoin de spécifier l'extension.

Chapitre 2 : Instrumentation

- **Exécutez le programme :**

Maintenant vous pouvez exécuter le code. En cliquant sur le bouton Rune (Exécuter), le programme devrait vous demander votre nom dans la partie inférieure de la fenêtre (Shell). Après l'avoir entré.

- **Exécutez le programme depuis la fenêtre du terminal :**

Il est également possible d'exécuter le programme à partir de la fenêtre du terminal. Pour ce faire, ouvrez une fenêtre de terminal en cliquant sur l'icône du terminal (coin supérieur gauche de votre écran).

Au niveau de la ligne de commande, nous devons d'abord nous déplacer dans le dossier où le fichier Python a été sauvegardé, Documents dans notre cas. Pour ce faire, entrez.

2.5 Le matériel utilisés :

2.5.1 LED :

Une LED ou diode est un dispositif semi-conducteur qui sert principalement de commutateur de courant unidirectionnel. Il permet au courant circulaire de se contracter facilement dans un sens, mais fortement dans le sens opposé.

La diode doit être montée dans le circuit électrique de telle sorte que le courant entrant dans le circuit passe d'abord par l'anode positive (+), puis par la cathode négative (-) et, dans le cas d'une connexion opposée, il coupe chaque composant électronique, qui suit le circuit ou qui est fourni avec un kit.

L'extrémité de la diode est une ligne de couleur au plomb caractéristique qui est la cathode, et l'autre extrémité est l'anode qui reçoit la tension.

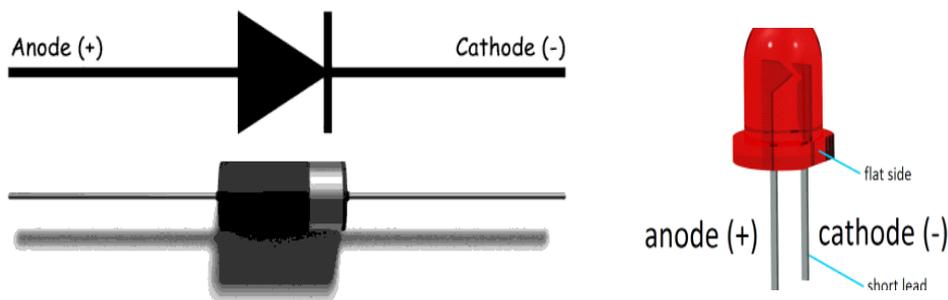


Figure 13 : Module de LED.

Branchement du led :

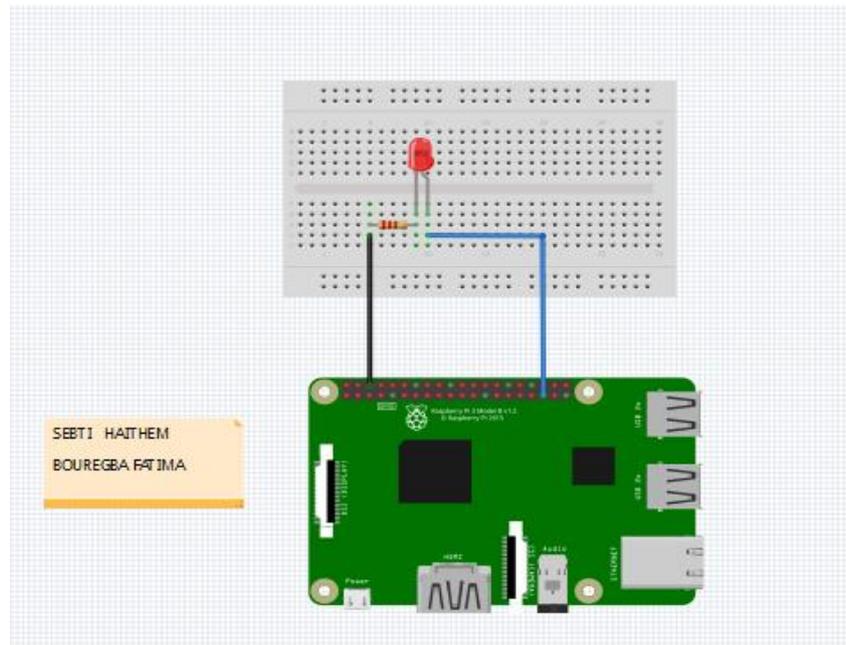


Figure 14 : Branchement d'une LED.

Code Python pour allumer et éteindre la Led (LED.py) :

```
1 import RPi.GPIO as GPIO
2 import time
3
4 def led_on():
5     GPIO.setup(19,GPIO.OUT)
6     GPIO.output(19,GPIO.HIGH)
7     print("LED is ON")
8
9 def led_off():
10    GPIO.setup(19,GPIO.OUT)
11    GPIO.output(19,GPIO.LOW)
12
13 if __name__ == '__main__':
14    GPIO.setmode(GPIO.BCM)
15    GPIO.setwarnings(False)
16    GPIO.setup(19,GPIO.OUT)
17    # While loop
18    while True:
19        # set GPIO14 pin to HIGH
20        GPIO.output(19,GPIO.HIGH)
21        print("LED is ON")
22        # show message to Terminal
23
24        # pause for one second
25        time.sleep(1)
26
27        # set GPIO14 pin to HIGH
28        GPIO.output(19,GPIO.LOW)
29        # show message to Terminal
30        print("LED is OFF")
31        # pause for one second
32        time.sleep(1)
33
34
```

2.5.2 Le DHT11 :

Le capteur DHT11 mesure la température et l'humidité. L'utilisation de ce type de capteur est intéressant pour un débutant, car il permet de mesurer une grandeur physique accessible à tous.

Chapitre 2 : Instrumentation

Le capteur DHT11 a 4 broches, mais il est souvent vendu sur une carte de support avec 3 broches.

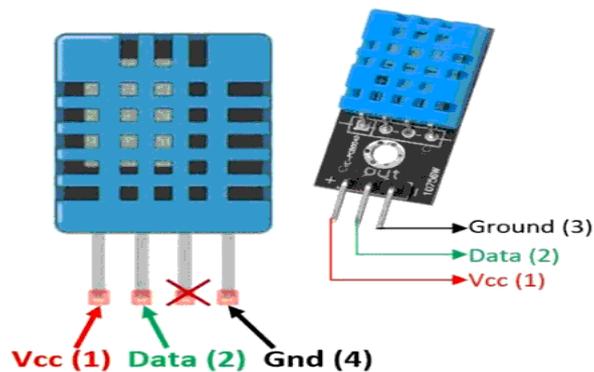


Figure 15 : Module DHT 11.

2.5.3 L'écran LCD :

Il s'agit en particulier d'une interface visuelle entre un système (projet) et l'homme (utilisateur). Sa fonction est de transmettre des informations utiles d'un système à un utilisateur. Par conséquent, les données qui peuvent être utilisées par l'utilisateur d'un système sont affichées.

La première information connue est le nombre de caractères par ligne. Pour ce modèle, il y a 16 caractères en deux lignes, soit un total de 32 caractères. Il est évident que ces informations figurent dans les données de 16 X 02, mais aussi dans la référence 1602 A.



Figure 16 : Module LCD 1602 A (V2.0).

2.5.4 BusI2C :

Le bus I2C (Inter Intégrante Circuit Bus) est le bus historique développé par Philips au début des années 1980 en particulier pour permettre une connexion aisée avec un microprocesseur aux différents circuits d'un téléviseur moderne.

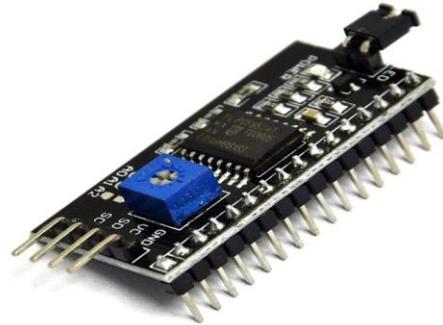


Figure 17 : Module I2C.

Branchement du DHT11 :

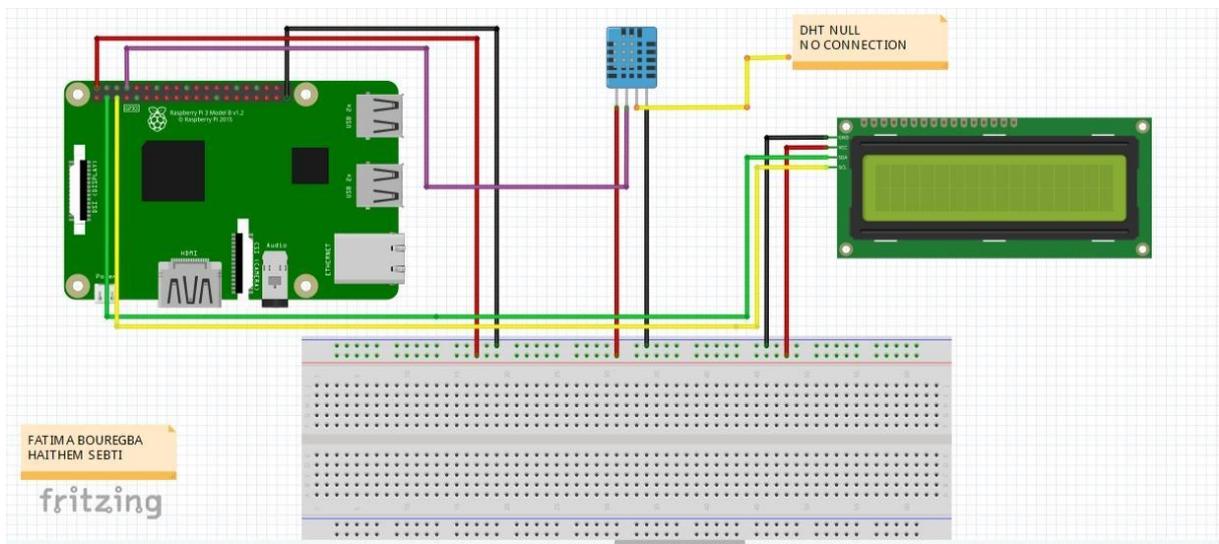


Figure 18 : Branchement DHT11.

Code python de température (lcd_dht.py) et le message email (email.py) :

Code message email :

Chapitre 2 : Instrumentation

```
1 import smtplib
2
3 def send_email(message, reci):
4
5     s = smtplib.SMTP('smtp.gmail.com', 587)
6     s.starttls()
7
8     # Authentication
9     s.login("haithem.sebti15@gmail.com", "haithemsebti23")
10
11     s.sendmail("haithem.sebti15@gmail.com", reci, message)
12
13     s.quit()
14
```

Code de température :

```
1 import smbus
2 import time
3 from datetime import datetime
4 import dht11
5 import RPi.GPIO as GPIO
6 from send_email import send_email
7 from LED import led_on, led_off
8
9 #define GPIO 14 as DHT11 data pin
10 Temp_sensor=14
11
12 # Define some device parameters
13 I2C_ADDR = 0x27 # I2C device address, if any error, change this address to 0x27
14 LCD_WIDTH = 16 # Maximum characters per line
15
16 # Define some device constants
17 LCD_CHR = 1 # Mode - Sending data
18 LCD_CMD = 0 # Mode - Sending command
19
20 LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
21 LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line
22 LCD_LINE_3 = 0x94 # LCD RAM address for the 3rd line
23 LCD_LINE_4 = 0xD4 # LCD RAM address for the 4th line
24 LCD_BACKLIGHT = 0x08 # On
25 #LCD_BACKLIGHT = 0x00 # Off
26
27 ENABLE = 0b00000100 # Enable bit
28
29 # Timing constants
30 E_PULSE = 0.0005
31 E_DELAY = 0.0005
32
33 #Open I2C interface
```

```
34 #bus = smbus.SMBus(0) # Rev 1 Pi uses 0
35 bus = smbus.SMBus(1) # Rev 2 Pi uses 1
36
37 def lcd_init():
38     # Initialise display
39     lcd_byte(0x33,LCD_CMD) # 110011 Initialise
40     lcd_byte(0x32,LCD_CMD) # 110010 Initialise
41     lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
42     lcd_byte(0x0C,LCD_CMD) # 001100 Display On,Cursor Off, Blink Off
43     lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines, font size
44     lcd_byte(0x01,LCD_CMD) # 000001 Clear display
45     time.sleep(E_DELAY)
46
47 def lcd_byte(bits, mode):
48     # Send byte to data pins
49     # bits = the data
50     # mode = 1 for data
51     #       0 for command
52
53     bits_high = mode | (bits & 0xF0) | LCD_BACKLIGHT
54     bits_low = mode | ((bits<<4) & 0xF0) | LCD_BACKLIGHT
55
56     # High bits
57     bus.write_byte(I2C_ADDR, bits_high)
58     lcd_toggle_enable(bits_high)
59
60     # Low bits
61     bus.write_byte(I2C_ADDR, bits_low)
62     lcd_toggle_enable(bits_low)
63
64 def lcd_toggle_enable(bits):
65     # Toggle enable
```

Chapitre 2 : Instrumentation

```
66 time.sleep(E_DELAY)
67 bus.write_byte(I2C_ADDR, (bits | ENABLE))
68 time.sleep(E_PULSE)
69 bus.write_byte(I2C_ADDR, (bits & ~ENABLE))
70 time.sleep(E_DELAY)
71
72 def lcd_string(message,line):
73
74     message = message.ljust(LCD_WIDTH, " ")
75     lcd_byte(line, LCD_CMD)
76
77     for i in range(LCD_WIDTH):
78         lcd_byte(ord(message[i]),LCD_CHR)
79
80 def main():
81     # Main program block
82     GPIO.setwarnings(False)
83     GPIO.setmode(GPIO.BCM)           # Use BCM GPIO numbers.
84     # Initialise display
85     lcd_init()
86     instance = dht11.DHT11(pin=Temp_sensor)
87
88     is_send_temp = False
89
90     while True:
91         #get DHT11 sensor value
92         result = instance.read()
93         c_time = datetime.now().strftime("%H:%M:%S")
94         if result.is_valid():
95             if result.temperature > 35 and not is_send_temp:
96                 is_send_temp = True
97
98                 send_email("temperature superieur a 35C", 'bouregbafatima3@gmail.com')
99                 led_on()
100
101             elif result.temperature < 35:
102                 led_off()
103                 is_send_temp = False
104
105             lcd_string("Temp: " + str(result.temperature)+" C",LCD_LINE_1)
106             lcd_string("Humid: " + str(result.humidity)+"%",LCD_LINE_2)
107             time.sleep(3) # 3 second delay
108             lcd_string("Time: " + str(c_time),LCD_LINE_1)
109             lcd_string("Maison intellegent",LCD_LINE_2)
110             time.sleep(3)
111
112 def start_lcd_dht():
113
114     try:
115         main()
116     except KeyboardInterrupt:
117         pass
118     finally:
119         lcd_byte(0x01, LCD_CMD)
120
121 if __name__ == '__main__':
122     start_lcd_dht()
```

2.5.5 Servomoteur :

Un servomoteur est un système motorisé capable d'atteindre et de maintenir des positions prédéterminées. La position est une valeur angulaire pour un moteur rotatif et une distance pour un moteur linéaire. On utilise des moteurs électriques (continus, asynchrones, à douilles) et des moteurs hydrauliques. La mise en marche et l'entretien de la position par défaut sont commandés par un système de réglage.

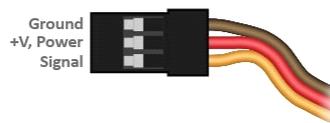


Figure 19 : Servomoteur.

Chapitre 2 : Instrumentation

Caractéristiques du servomoteur SG90 :

- Couple : 25,0 oz-in (1,80 kg.cm) à 4,8 V
- Régime : 0,1 S/60 pF (4,8 V)
- Tension : 4,0 V ou plus mais n'excédant pas 7,2 V, 4,6 V - 5,2 V nominaux
- Version durcis sable avec bloc d'alimentation 5 V (sans shunt mécanique) 220 50 mA
- Stal curent avec bloc d'alimentation 5 V (Horne Locke) 650 80 mA
- Adaptateur Idole avec bloc d'alimentation 5V 6 10 mA
- Dimensions : 0,91 x 0,48 x 1,14 (23 x 12,2 x 29 mm)
- Perruque 0,32 oz (9 g)



- Dead band Wind 10 μ s
- Plage de températures de fonctionnement -22 F à 140 F (-30 C à 60 C)
- Connecteur universel « S » type fismustresas

Fonctionnement du servomoteur :

La commande d'un servomoteur avec un microcontrôleur est simple, pas une commande externe.

Le principe de base est très simple. Il suffit d'envoyer une impulsion, et c'est le temps que prend cette impulsion, qui détermine l'angle du servomoteur.

Ce temps d'impulsion est de quelques millisecondes et doit être répété à intervalles réguliers (50 Hz ou toutes les 20 ms).

Si le temps de pulsation varie d'un fabricant à l'autre, les valeurs de ce schéma sont assez en attente.

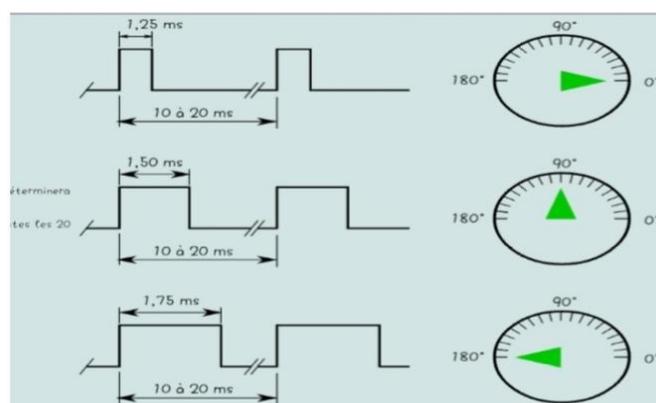


Figure 20 : Un train d'impulsion de période 20 ms avec une rotation.

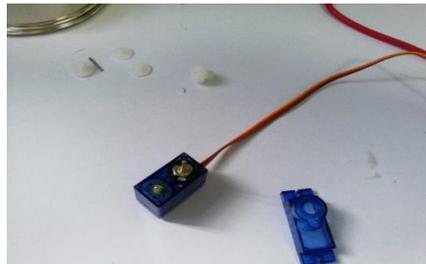
Chapitre 2 : Instrumentation

Modification d'un micro serviteur (SG90) pour une rotation continue :

Un servomoteur normal fonctionne à environ 180 degrés.

Faites des pas pour enlever le limiteur du Servo et faites-le tourner à 360 degrés pour pouvoir l'utiliser comme si vous utilisiez un moteur à courant continu !

Le limiteur de courant comprend deux parties : le potentiomètre et la boîte de vitesses. Ce que nous devons faire, c'est rompre la connexion du potentiomètre avec la carte de circuit interne et retirer un bouton de pignon de la boîte de vitesses.



Retirer de couvercle et de la boîte de vitesses.

Puis retirer le moteur pour

Courant direct et échelle potentielle.



Couper la patte qui se détache de l'engrenage. Remplacement du potentiomètre par les deux résistances.

Chapitre 2 : Instrumentation

Branchement du Servomoteur :

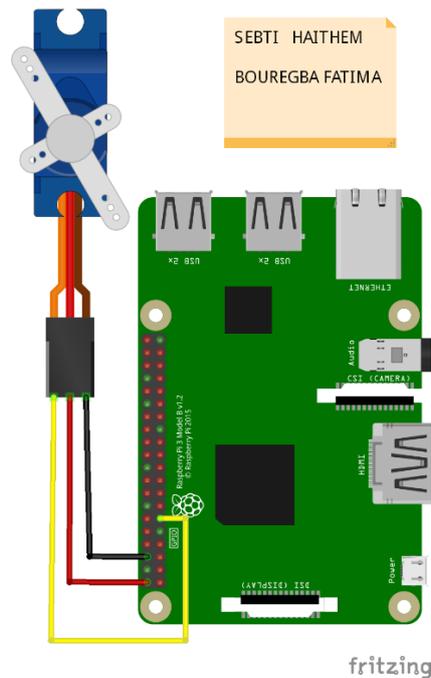


Figure 21 : Branchement du Servomoteur.

Code python d'un servomoteur (servo.py) :

```
1 # Import libraries
2 import RPi.GPIO as GPIO
3 import time
4
5 def servo_start():
6
7
8     # Set pin 17 as an output, and set servol as pin 11 as PWM
9     GPIO.setup(17,GPIO.OUT)
10    servol = GPIO.PWM(17,50) # Note 17 is pin, 50 = 50Hz pulse
11
12    #start PWM running, but with value of 0 (pulse off)
13    servol.start(0)
14    print ("Waiting for 2 seconds")
15    time.sleep(2)
16
17    #Let's move the servo!
18    print ("Rotating 180 degrees in 10 steps")
19
20    # Define variable duty
21    duty = 2
22
23    # Loop for duty values from 2 to 12 (0 to 180 degrees)
24    while duty <= 12:
25        servol.ChangeDutyCycle(duty)
26        time.sleep(1)
27        duty = duty + 1
28
29    # Wait a couple of seconds
30    time.sleep(2)
31
```

Chapitre 2 : Instrumentation

```
31
32 # Turn back to 90 degrees
33 print ("Turning back to 90 degrees for 2 seconds")
34 servol.ChangeDutyCycle(7)
35 time.sleep(2)
36
37 #turn back to 0 degrees
38 print ("Turning back to 0 degrees")
39 servol.ChangeDutyCycle(2)
40 time.sleep(0.5)
41 servol.ChangeDutyCycle(0)
42
43 #Clean things up at the end
44 servol.stop()
45 # GPIO.cleanup()
46
47 if __name__ == '__main__':
48     servo_start()
```

2.5.6 Gaz SensorMQ2 :

Utile pour la détection des fuites de gaz (domestique et industriel). Convient pour la détection de H₂, LPG, CH₄, CO, alcool, fumée ou propane. En raison de sa sensibilité élevée et de son temps de réponse rapide, la mesure peut être effectuée aussi rapidement que possible. La sensibilité du capteur peut être réglée par potentiomètre.



Figure 22 : Gaz Sensor MQ2.

2.5.7 MCP3008 :

Le MCP3008 est un convertisseur analogique-numérique à canal 8 de 10 bits à faible coût. La précision de cet ADC ressemble à celle d'une carte électronique, et avec 8 canaux, vous pouvez lire de nombreux signaux analogiques sur Pi. Cette puce est un bon choix si vous n'avez qu'à reproduire des signaux analogiques simples provenant d'un capteur de température ou de lumière.

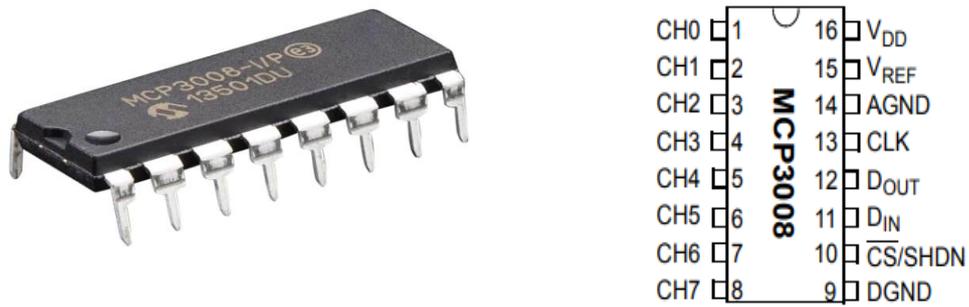


Figure 23 : Module du MCP3008.

Branchement de capteur MQ2 :

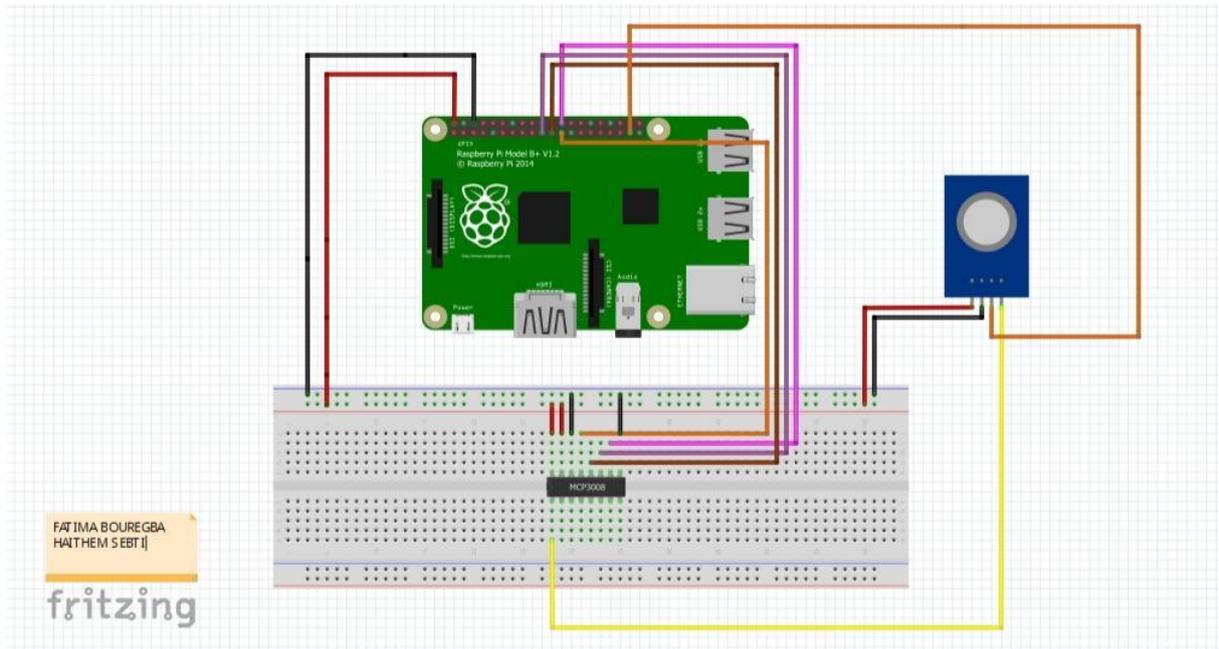


Figure 24 : branchement du capteur MQ2.

Code python mq2 (mq2.py) :

Chapitre 2 : Instrumentation

```
1 import RPi.GPIO as GPIO
2 import time
3 from servo import servo_start
4 from LED import led_on, led_off
5 from send_email import send_email
6
7 # change these as desired - they're the pins connected from the
8 # SPI port on the ADC to the Cobbler
9 SPICLK = 11
10 SPIMISO = 9
11 SPIMOSI = 10
12 SPICS = 8
13 mq2_dpin = 26
14 mq2_apin = 0
15
16 #port init
17 def init():
18     GPIO.setwarnings(False)
19     GPIO.cleanup() #clean up at the end of your script
20     GPIO.setmode(GPIO.BCM)
21
22     # specify which pin numbering system
23     # set up the SPI interface pins
24     GPIO.setup(SPIMOSI, GPIO.OUT)
25     GPIO.setup(SPIMISO, GPIO.IN)
26     GPIO.setup(SPICLK, GPIO.OUT)
27     GPIO.setup(SPICS, GPIO.OUT)
28     GPIO.setup(mq2_dpin,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
29
30 #read SPI data from MCP3008(or MCP3204) chip,8 possible adc's (0 thru 7)
31 def readadc(adcnum, clockpin, mosipin, misopin, cspin):
32     if ((adcnum > 7) or (adcnum < 0)):
```

```
34     GPIO.output(cspin, True)
35
36     GPIO.output(clockpin, False) # start clock low
37     GPIO.output(cspin, False) # bring CS low
38
39     commandout = adcnum
40     commandout |= 0x18 # start bit + single-ended bit
41     commandout <<= 3 # we only need to send 5 bits here
42     for i in range(5):
43         if (commandout & 0x80):
44             GPIO.output(mosipin, True)
45         else:
46             GPIO.output(mosipin, False)
47         commandout <<= 1
48         GPIO.output(clockpin, True)
49         GPIO.output(clockpin, False)
50
51     adcout = 0
52     # read in one empty bit, one null bit and 10 ADC bits
53     for i in range(12):
54         GPIO.output(clockpin, True)
55         GPIO.output(clockpin, False)
56         adcout <<= 1
57         if (GPIO.input(misopin)):
58             adcout |= 0x1
59
60     GPIO.output(cspin, True)
61
62     adcout >>= 1 # first bit is 'null' so drop it
63     return adcout
```

Chapitre 2 : Instrumentation

```
64 #main loop
65 def main():
66     init()
67     print("please wait...")
68     time.sleep(1)
69
70     is_alert = False
71
72     while True:
73         COlevel=readadc(mq2_apin, SPICLK, SPIMOSI, SPIMISO, SPICS)
74
75         if GPIO.input(mq2_dpin):
76             is_alert = False
77             led_off()
78             print("Gas not leak")
79             time.sleep(0.5)
80
81         else:
82             if not is_alert:
83                 is_alert = True
84                 led_on()
85                 send_email('home alert', 'haithem.sbt@gmail.com')
86                 servo_start()
87
88             print("Gas leakage")
89             print("Current Gas AD vaule = " +str("%.2f"%((COlevel/1024.)*3.3))+ " V")
90             time.sleep(0.5)
91
92 #start PWM running, but with value of 0 (pulse off)
93
94
95
96 if __name__ == '__main__':
97     try:
```

```
96 if __name__ == '__main__':
97     try:
98         main()
99     pass
100 except KeyboardInterrupt:
101     pass
102
```

2.5.8 Le moteur pas à pas :

Un moteur pas à pas est un moteur électrique à courant continu sans balais, qui subdivise une rotation complète en plusieurs étapes. Il ne se déplace pas à pas, et chaque pas est de la même taille. Cela nous permet de faire tourner le moteur d'un angle précis à une position précise. Le moteur pas à pas peut tourner dans le sens des aiguilles d'une montre ou dans le sens inverse des aiguilles d'une montre.



Figure 25 : Moteur pas à pas.

Chapitre 2 : Instrumentation

2.5.9 Pilote de moteur ULN2003 :

Pour relier le moteur pas à pas au Raspberry Pi Board, nous utilisons le contrôleur moteur ULN2003, comme illustré sur la figure ci-dessous. Le moteur pas à pas 28BYJ-48 est souvent vendu avec le pilote du moteur ULN2003.

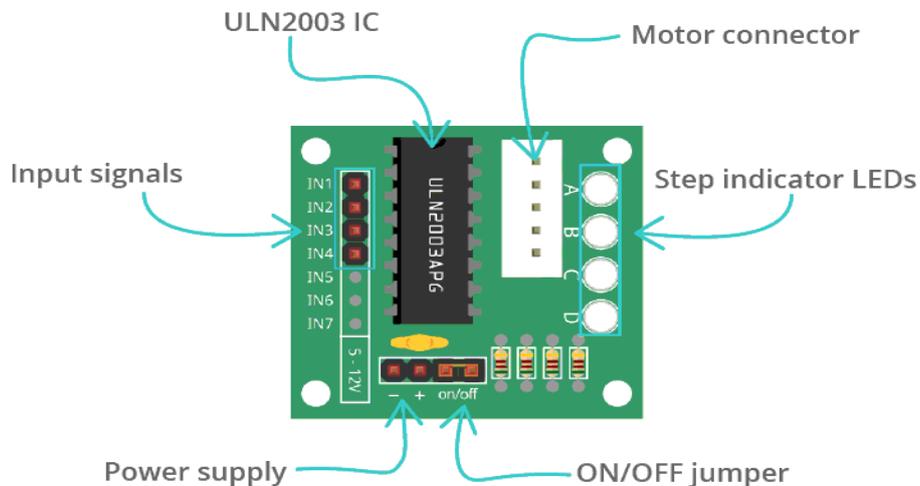


Figure 26 : Module ULN 2003.

Le module est fourni avec un connecteur qui facilite et facilite le raccordement du moteur au module. Il dispose de quatre broches d'entrée pour commander les bobines qui déplacent le moteur pas à pas. Les quatre LED fournissent une interface visuelle pour l'état des bobines.

Il existe des broches pour la connexion de VCC et de GND, ainsi qu'un capuchon de dérivation qui sert de commutateur marche/arrêt pour alimenter le moteur pas à pas. Si vous retirez le pont, il n'y a plus de courant vers le moteur.

Schéma de connexion :

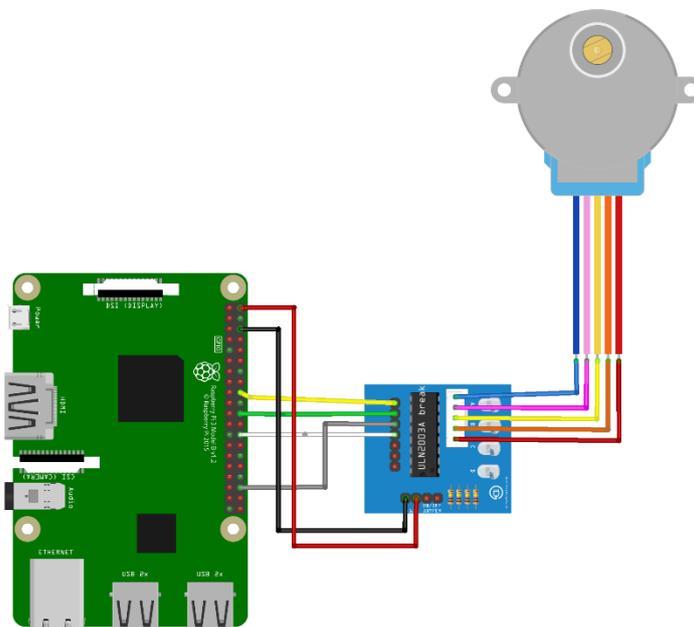


Figure 27 : Circuit du moteur pas à pas.

Chapitre 2 : Instrumentation

Code python d'un moteur pas à pas (engine.py) :

```
1 import time
2 import RPi.GPIO as GPIO
3 # Use BCM GPIO references
4 # Instead of physical pin numbers
5 GPIO.setmode(GPIO.BCM)
6 # Define GPIO signals to use Pins 18,22,24,26 GPIO24,GPI025,GPI08,GPI07
7 StepPins = [24,25,16,7]
8 # Set all pins as output
9 for pin in StepPins:
10     print("Setup pins")
11     GPIO.setup(pin,GPIO.OUT)
12     GPIO.output(pin, False)
13 # Define time delay between steps
14 WaitTime = 0.005 #set speed
15 # Define simple sequence
16 StepCount1 = 4
17 Seq1 = []
18 Seq1 = [i for i in range(0, StepCount1)]
19 Seq1[0] = [1,0,0,0]
20 Seq1[1] = [0,1,0,0]
21 Seq1[2] = [0,0,1,0]
22 Seq1[3] = [0,0,0,1]
23 # Define advanced half-step sequence
24 StepCount2 = 8
25 Seq2 = []
26 Seq2 = [i for i in range(0, StepCount2)]
27 Seq2[0] = [1,0,0,0]
28 Seq2[1] = [1,1,0,0]
29 Seq2[2] = [0,1,0,0]
30 Seq2[3] = [0,1,1,0]
```

```
31 Seq2[4] = [0,0,1,0]
32 Seq2[5] = [0,0,1,1]
33 Seq2[6] = [0,0,0,1]
34 Seq2[7] = [1,0,0,1]
35 # Choose a sequence to use
36 Seq = Seq2
37 StepCount = StepCount2
38 def steps(nb):
39     StepCounter = 0
40     if nb<0: sign=-1
41     else: sign=1
42     nb=sign*nb*2 #times 2 because half-step
43     print("nbsteps {} and sign {}".format(nb,sign))
44     for i in range(nb):
45         for pin in range(4):
46             xpin = StepPins[pin]
47             if Seq[StepCounter][pin]!=0:
48                 GPIO.output(xpin, True)
49             else:
50                 GPIO.output(xpin, False)
51         StepCounter += sign
52     # If we reach the end of the sequence
53     # start again
54     if (StepCounter==StepCount):
55         StepCounter = 0
56     if (StepCounter<0):
57         StepCounter = StepCount-1
58     # Wait before moving on
```

Chapitre 2 : Instrumentation

```
58         # Wait before moving on
59         time.sleep(WaitTime)
60 # Start main loop
61 nbStepsPerRev=2048
62 def start():
63     hasRun=False
64     while not hasRun:
65         steps(nbStepsPerRev)# parcourt un tour dans le sens horaire
66         time.sleep(1)
67         steps(-nbStepsPerRev)# parcourt un tour dans le sens anti-horaire
68         time.sleep(1)
69         hasRun=True
70     print("Stop motor")
71     for pin in StepPins:
72         GPIO.output(pin, False)
73         time.sleep(1)
74     start()
75
```

2.5.10 Caméra Surveillance :

La caméra Raspberrypi a une résolution de 5 méga pixels et est compatible avec les modèles pi 2B, 3,3B, 3B + et 4 Raspberry, elle est également compatible avec les modèles Zéro et Zéro W (avec le bon câble plat). Une bibliothèque Python pour le pi camera permet aux utilisateurs de contrôler l'appareil photo, de prendre des photos et de capturer des vidéos, les utilisateurs peuvent également contrôler la balance des blancs, le déclencheur et chaque pixel RVB directement depuis l'appareil photo avec la bibliothèque.



Figure 28 : Caméra Surveillance.

Caractéristiques :

- Résolution de 5 MP
- Résolution vidéo maximale : 1080p @ 30fps
- Module de caméra PCB dimension : 25 mm x 24 mm (9 mm d'épaisseur)

Code python d'un Caméra Surveillance :

Chapitre 3 :

Réseaux de neurone convolutif (CNN)

Chapitre 3 : Réseau de neurone convolutif(CNN)

Comme pour les réseaux neuronaux ordinaires, les paramètres des couches sont déterminés par rétro propagation à gradient : la radiographie croisée est minimisée pendant la phase d'entraînement. Toutefois, dans le cas de CNN, ces paramètres se rapportent en particulier aux calculs d'images.

Les différents types de postes sur un CNN sont expliqués dans le chapitre suivant.

3 Réseaux de neurone convolutif (CNN) :

3.1 Définition :

Un réseau neural convoluté (CNN) est une sorte de réseau de rétroaction neuronale artificielle utilisé pour l'apprentissage profond. C'est un perceptron multicouche contrôlé d'un type particulier. Les Convoluté Neural Networks (CNN) sont une sorte de modèle de profondeur inspiré par les personnes qui traitent l'information. Chaque neurone du cortex visuel du cerveau a un champ de réception qui recueille des données d'un certain voisinage local dans l'espace visuel. Ils ont été développés pour identifier des données multidimensionnelles avec un degré élevé de biais et d'invariance par rapport à la variation d'escalade.

3.2 Architectur de réseaux de neurones convolutif :

Cette architecture comprend en plus de la couche d'entrée, plusieurs couches de convolution et de pooling. Le sommet du réseau est un réseau complètement connecté (Full Connected FC) qui permet la classification des caractéristiques extraites par les couches de convolutions. La couche d'entrée est constituée de la matrice de données à classifier, représentée généralement par la matrice de l'image, dans le cas des images monochromes, ou les 3 matrices de l'image pour le cas d'images RGB.

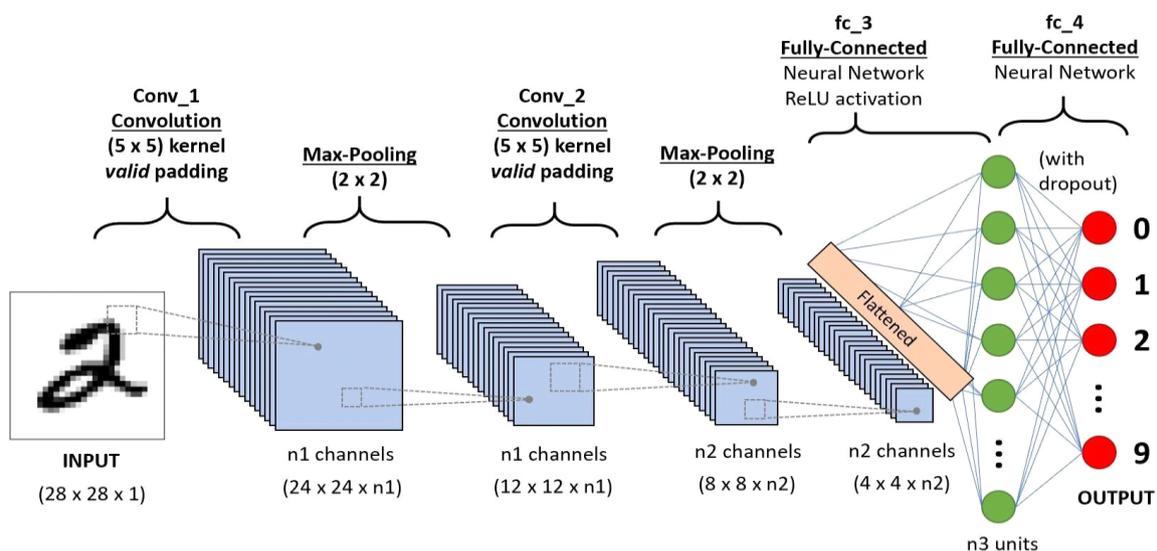


Figure 30 : Architecture d'un réseau de neurone convolutif.

Chapitre 3 : Réseau de neurone convolutif(CNN)

3.3 La couche de convolution :

La couche de convolution est la composante clé des réseaux de neurone convolutif, et constitue toujours au moins leur première couche. Une couche de convolution est caractérisée par la taille et le nombre des filtres de convolution. La figure 30 illustre l'opération de convolution. En considérant une image ou une carte de caractéristiques de 5x5 pixels (matrice bleue) et un filtre de 3x3 (matrice rouge), la convolution est la somme des produits des éléments qui se trouvent aux mêmes positions des deux matrices pour chaque étape où le filtre de convolution est décalé rapport à la matrice de données selon un pas (stride) horizontalement et verticalement. Pour conserver la même dimensionnalité des matrices d'entrée et de sortie, le zéro-padding peut être utilisé (figure 30)

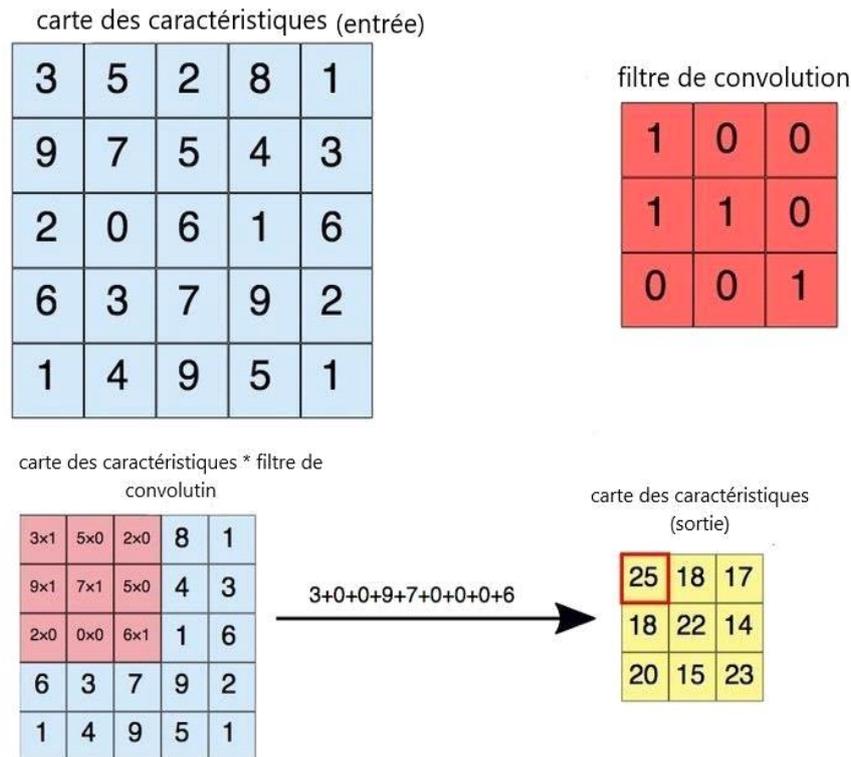


Figure 31 : Exemple illustrant l'opération de convolution 3x3.

Chapitre 3 : Réseau de neurone convolutif(CNN)

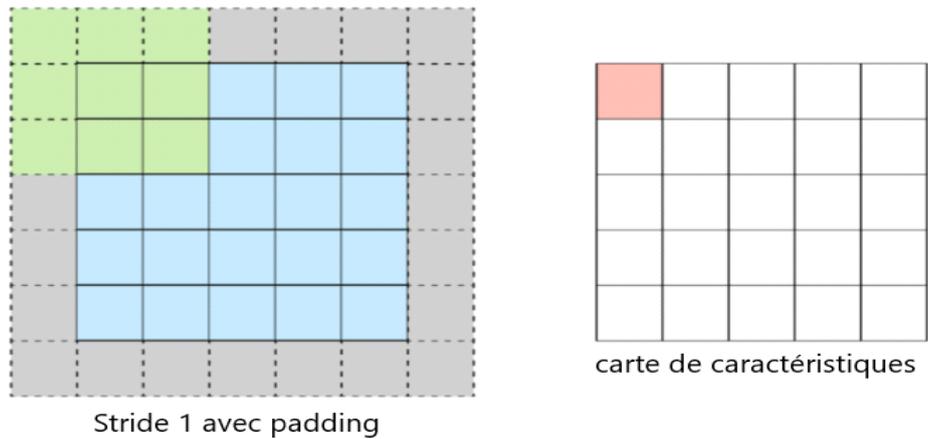


Figure 32 : Exemple de zéro padding.

Par exemple, des valeurs des pixels d'une image 5x5

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Figure 33 : Exemple simpliste des valeurs du pixel 5x5.

1	0	1
0	1	0
1	0	1

Figure 34 : Exemple de valeur d'une matrice utilisée comme filtre.

Dans le cas de la Figure 34, les valeurs sont binaires. Dans un cas réel, les valeurs de varient varier entre 0 et 255. Dans la Figure33, les valeurs sont représentées par des 1 et 0. Dans un cas réel, ces valeurs sont continuées et peuvent être positives ou négatives.

Chapitre 3 : Réseau de neurone convolutif(CNN)

1*1	1*0	1*1	0	0
0*0	1*1	1*0	1	0
0*1	0*0	1*1	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved Feature

Figure 35 : Démonstration de convolution.

Dans la Figure (35) nous pouvons voir que chaque valeur des pixels de l'image est multipliée par chaque valeur correspondante du filtre (Figure35). Cette valeur va générer un nouveau pixel qui fera partie d'une nouvelle image (ujjwalkarn, 2016).

Le filtre doit se déplacer d'une case à chaque itération jusqu'à ce que la première ligne soit finie. Lorsque nous avons fini la première ligne, le filtre « descend » d'une case et la même procédure répétée pour chaque ligne et colonne (ujjwalkarn,2016).

4	3	4
2	4	3
2	3	4

Figure 36 : Résultat attendu après une phase de convolution.

3.4 Le perceptron :

Le perceptron est un neurone artificiel, une unité qui appartient à un réseau de neurones artificiel. Pour mieux comprendre ce concept, il est nécessaire de revenir à la définition de réseau de neurones. Un cerveau humain est constitué de milliards de neurones, des cellules nerveuses connectées entre elles, qui permettent de traiter des informations. Le réseau de neurones artificiels tend à copier le fonctionnement du cerveau humain, en recréant des neurones fictifs. Le perceptron est un algorithme qui reçoit des données avec un certain poids, qui les calcule et qui produit un résultat transmis à d'autres perceptrons par des liens

Chapitre 3 : Réseau de neurone convolutif(CNN)

interconnectés. Le réseau de neurones artificiels avec une capacité d'apprentissage autonome est à la base de la machine Learning et de l'intelligence artificielle (IA).

3.5 Structure de perceptron :

La structure d'un perceptron est très simple

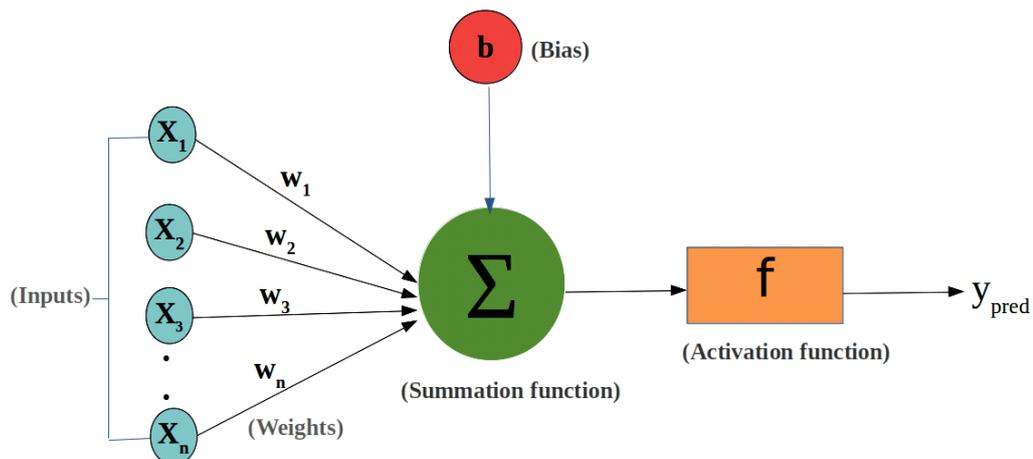


Figure 37 : Représentation graphique d'un perceptron.

- Les entrées ($x_1, x_2, x_3, \dots, x_n$)
 - Les poids ($w_1, w_2, w_3, \dots, w_n$)
 - Une fonction de somme (Σ)
 - Un biais (w_0)
 - Une fonction d'activation
 - Une sortie
-
- **Entrées** : les entrées sont l'ensemble de valeurs pour lesquelles nous devons prédire la valeur de sortie. Ils peuvent être visualisés en tant qu'entités ou attributs dans un data set.
 - **Les poids** : les poids sont les valeurs réelles associées à chaque caractéristique, ce qui indique l'importance de cette caractéristique dans la prédiction de la valeur finale.
 - **Biais** : le biais est utilisé pour déplacer la fonction d'activation vers la gauche ou la droite, il peut être appelé une ordonnée à l'origine dans l'équation de la ligne.
 - **Fonction d'activation** : elle est utilisée pour introduire la non-linéarité dans le

Chapitre 3 : Réseau de neurone convolutif(CNN)

modèle.

- **Fonction de sommation** : Le travail de la fonction de sommation consiste à lier les poids et les entrées ensemble et à trouver leur somme.

3.6 La Fonction d'activation :

La fonction d'activation dans un réseau neuronal est responsable de la conversion du nœud sommé.

Entrée pondérée lors de l'activation ou de la sortie du nœud pour cette entrée.

Les fonctions d'activation couramment utilisées comprennent

- Fonction - Relu : $Max(x, 0)$.
- Fonction - Sigmoidé : $\frac{1}{1+e^{-x}}$
- Fonction - Soft max : $\frac{e^x}{\sum e^{x_i}}$
- Fonction - Tanh : $\frac{e^x - e^{-x}}{e^x + e^{-x}}$

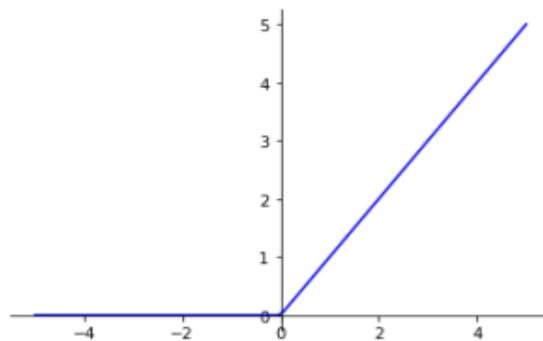


Figure 38 : Fonction Relu.

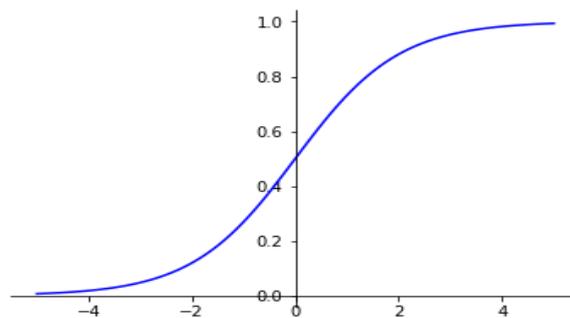


Figure 39 : Fonction Sigmoidé.

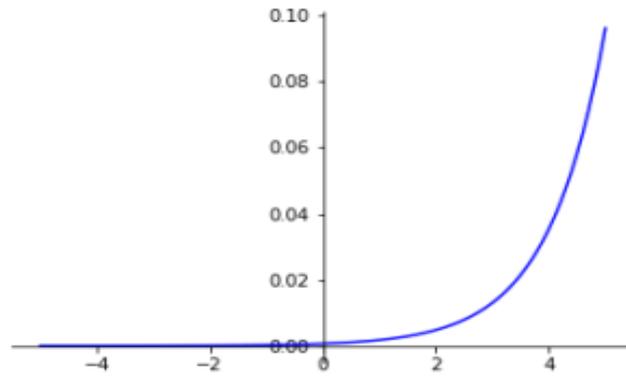


Figure 40 : Fonction Soft max.

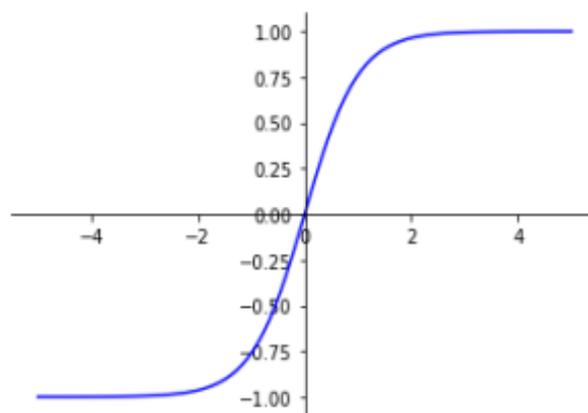
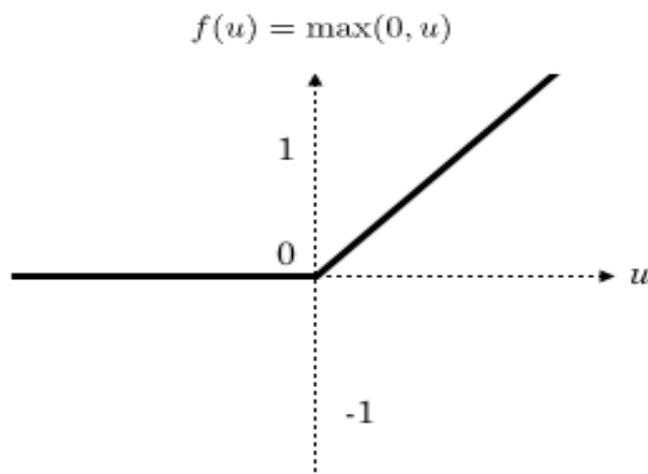


Figure 41 : Fonction Tanh.

3.7 La couche de correction Relu :

Relu (RectifiedLinearUnits) désigne la fonction réelle non-linéaire définie par $Relu(x) = \text{Max}(0, x)$



Chapitre 3 : Réseau de neurone convolutif(CNN)

$$F(x) = \{x \text{ si } x > 00 \text{ Ailleurs } \}$$

La couche de correction Relu remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation.

3.8 La couche de pooling :

Après une opération de convolution, unpooling est généralement effectué pour réduire la dimensionnalité des cartes de caractéristiques. Cela permet deréduire le nombre de paramètres, ce qui à la fois raccourcit et temps d'entraînement et empêchée sur apprentissage (overfitting). Les couches Pooling sous-échantillonnent chaque carte de caractéristiques indépendamment, en réduisant la hauteur et la largeur, tout en conservant la profondeur intacte. Il existe plusieurs types de pooling (figure41):

- *Maxpooling* : qui revient à prendre la valeur maximale de la sélection. C'est le type le plus utilisé, car il est rapide à calculer (immédiat), et permet de simplifier efficacement l'image.
- *Meanpooling* :(ou average pooling) qui revient à calculer la somme de toutes les valeurs et diviser par le nombre de valeurs. On obtient ainsi une valeur intermédiaire pour représenter ce lot de pixels.
- *Sumpooling* : c'est la moyenne sans avoir divisé par le nombre de valeurs (on ne calcule que leur somme)

Chapitre 3 : Réseau de neurone convolutif(CNN)

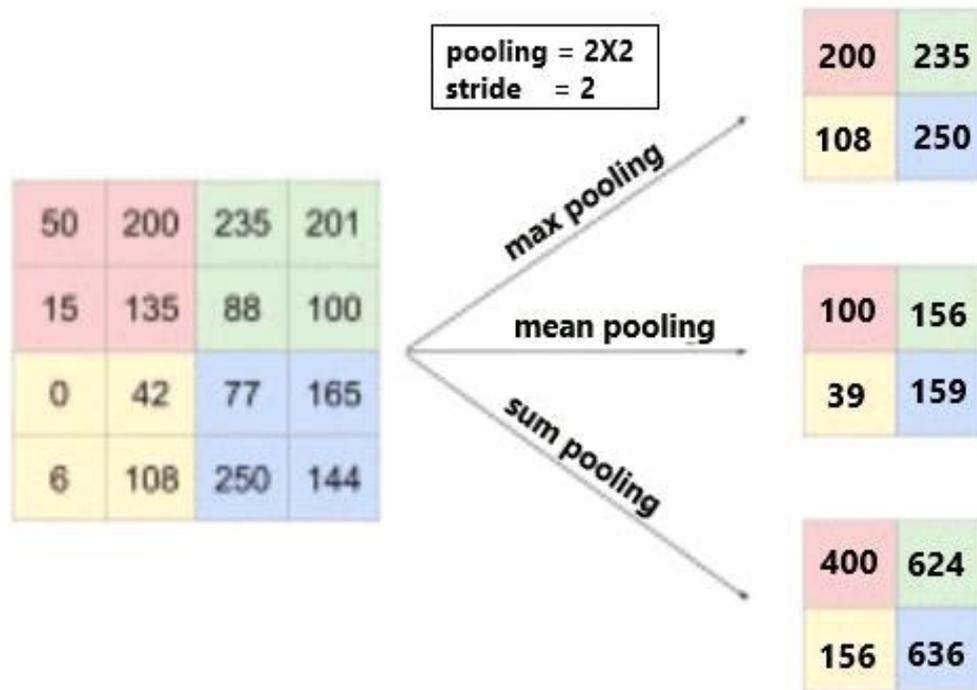


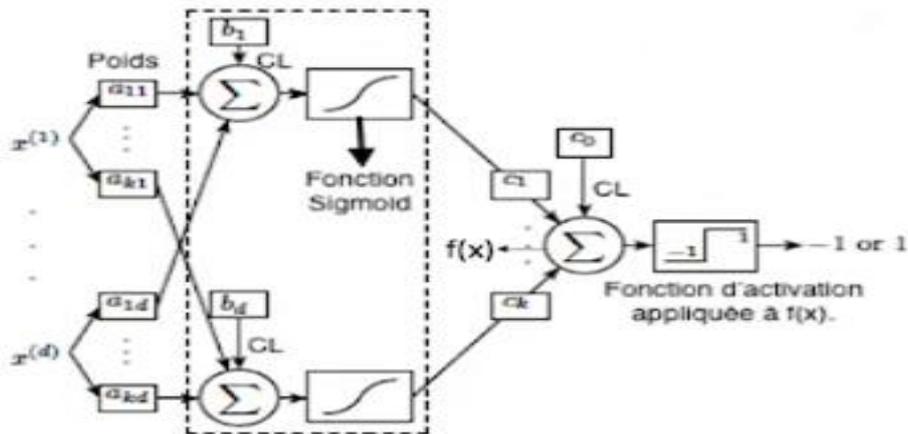
Figure 42 : Exemples de calcul du pooling sur une image 4×4.

Contrairement à l'opération de convolution, l'opération de pooling n'utilise pas des poids et fait glisser une fenêtre sur son entrée et prend simplement une valeur dans cette fenêtre selon le type du pooling.

3.9 Couche entièrement connectées (FullyConnected (FC)) :

Ces couches sont placées en fin d'architecture de CNN et sont entièrement connectées à tous les neurones de sorties (d'où le terme Fully-Connected). Après avoir reçu un vecteur en entrée, la couche FC applique successivement une combinaison linéaire puis une fonction d'activation dans le but final de classifier l'input image (voir schéma suivant). Elle renvoie enfin en sortie un vecteur de taille d correspondant au nombre de classes dans lequel chaque composante représente la probabilité pour l'input image d'appartenir à une classe.

Chapitre 3 : Réseau de neurone convolutif(CNN)



$$F(x) = c_0 + \sum_{i=1}^p c_i \sigma \psi(x)$$

Pour $x \in R^*$

3.10 Liste des fonctions de coût utilisées dans les réseaux de neurones :

- Reculer : $\mathbf{j} = \frac{1}{2} \sum_j (\hat{y}_j - y_j^{pred})^2$
- Classifier : $\mathbf{j} = - \sum_j (\hat{y}_j y_j) + (1 - \hat{y}_j) \log \log(1 - y_j)$
- Coût Exponentiels : $\mathbf{j} = \tau \exp \left[\frac{1}{\tau} \sum_j (\hat{y}_j - y_j^{pred})^2 \right]$
- Distance D'Itakura-Saito: $\mathbf{j} = \sum_j \left(\frac{\hat{y}_j}{y_j^{pred}} - \log \log \frac{\hat{y}_j}{y_j^{pred}} - 1 \right)$
- Divergence Itakura-Leibler: $\mathbf{j} = \sum_j \hat{y}_j \log \log \frac{\hat{y}_j}{y_j^{pred}}$
- Divergence Générale De Kullback-Leibler : $\mathbf{j} = \sum_j \hat{y}_j \log \log \frac{\hat{y}_j}{y_j^{pred}} - \sum_j \hat{y}_j - \sum_j y_j^{pred}$

3.11 Code python :

Chapitre 3 : Réseau de neurone convolutif(CNN)

```
1 # import the necessary packages
2 from imutils.video import VideoStream
3 from imutils.video import FPS
4 import face_recognition
5 import imutils
6 import pickle
7 import time
8 import cv2
9 from engine import start
10
11 #Initialize 'currentname' to trigger only when a new person is identified.
12 currentname = "unknown"
13 #Determine faces from encodings.pickle file model created from train_model.py
14 encodingsP = "encodings.pickle"
15
16 # load the known faces and embeddings along with OpenCV's Haar
17 # cascade for face detection
18 print("[INFO] loading encodings + face detector...")
19 data = pickle.loads(open(encodingsP, "rb").read())
20
21 # initialize the video stream and allow the camera sensor to warm up
22 # Set the ser to the following
23 # src = 0 : for the build in single web cam, could be your laptop webcam
24 # src = 2 : I had to set it to 2 inorder to use the USB webcam attached to my laptop
25 #vs = VideoStream(src=2, framerate=10).start()
26 vs = VideoStream(usePiCamera=True).start()
27 time.sleep(2.0)
28
29 # start the FPS counter
30 fps = FPS().start()
31
32 # loop over frames from the video file stream
33 while True:
34     # grab the frame from the threaded video stream and resize it
35     # to 500px (to speedup processing)
36     frame = vs.read()
37     frame = imutils.resize(frame, width=500)
38     # Detect the fce boxes
39     boxes = face_recognition.face_locations(frame)
40     # compute the facial embeddings for each face bounding box
41     encodings = face_recognition.face_encodings(frame, boxes)
42     names = []
43
44     # loop over the facial embeddings
45     for encoding in encodings:
46         # attempt to match each face in the input image to our known
47         # encodings
48         matches = face_recognition.compare_faces(data["encodings"],
49             encoding)
50         name = "Unknown" #if face is not recognized, then print Unknown
51
52         # check to see if we have found a match
53         if True in matches:
54             # find the indexes of all matched faces then initialize a
55             # dictionary to count the total number of times each face
56             # was matched
57             matchedIdxs = [i for (i, b) in enumerate(matches) if b]
58             counts = {}
59
60             # loop over the matched indexes and maintain a count for
61             # each recognized face face
62             for i in matchedIdxs:
63                 name = data["names"][i]
64                 counts[name] = counts.get(name, 0) + 1
65
66             # determine the recognized face with the largest number
67             # of votes (note: in the event of an unlikely tie Python
68             # will select first entry in the dictionary)
69             name = max(counts, key=counts.get)
70
71             #If someone in your dataset is identified, print their name on the screen
72             if currentname != name:
73                 currentname = name
74                 print(currentname)
75                 start()
76
77             # update the list of names
78             names.append(name)
79
80     # loop over the recognized faces
81     for ((top, right, bottom, left), name) in zip(boxes, names):
82         # draw the predicted face name on the image - color is in BGR
```

Chapitre 3 : Réseau de neurone convolutif(CNN)

```
83     cv2.rectangle(frame, (left, top), (right, bottom),
84                  (0, 255, 255), 2)
85     y = top - 15 if top - 15 > 15 else top + 15
86     cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
87                .8, (0, 255, 255), 2)
88
89     # display the image to our screen
90     cv2.imshow("Facial Recognition is Running", frame)
91     key = cv2.waitKey(1) & 0xFF
92
93     # quit when 'q' key is pressed
94     if key == ord("q"):
95         break
96
97     # update the FPS counter
98     fps.update()
99
100 # stop the timer and display FPS information
101 fps.stop()
102 print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
103 print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))
104
105 # do a bit of cleanup
106 cv2.destroyAllWindows()
107 vs.stop()
108
```

Chapiter4 : Simulation et résultat

Chapitre 4 : Simulation et résultat

4.1 Tableau d'analyse :

Les capteurs	Les actionneurs
- MQ2 -DHT11 -cam	-AL -OF+ -EMP -OVP -FEP -unknown -SV

Table I.1 : Tableau d'analyse

4.2 Explication les informations (Réceptivité) et les capteurs :

Les capteurs :

- MQ2 : capteur qui détecter le gaz.
- DHT11 : capteur détecter la température.
- Cam : détecter le facial.

Les information (Réceptivité) :

- msg : envoyée une message.
- AR : arrivée
- AUT : autorisé pour ouverture le portail.

Explication les actionneurs (les Taches):

- AL : allumée la Led
- OF+ : ouverture la fenêtre
- EMP : le message envoyé via gmail
- OVP : ouverture le portail
- FEP : fermeture le portail
- unknown : le facial qui inconnue
- SV : sauvé la maison des incendies

Chapitre 4 : Simulation et résultat

4.3 Résultat de gaz :

4.3.1 Cahier de charge :

Pour la protection des familles de l'étouffement au gaz :

Nous avons utilisé un détecteur du gaz capable de détecter une concentration anormale de gaz dans l'air. Lors de la détection du gaz une Led s'allume automatiquement puis :

- L'ouverture des fenêtres avec le servomoteur.
- Un message envoyé pour avertir le propriétaire.

4.3.2 Grafctet :

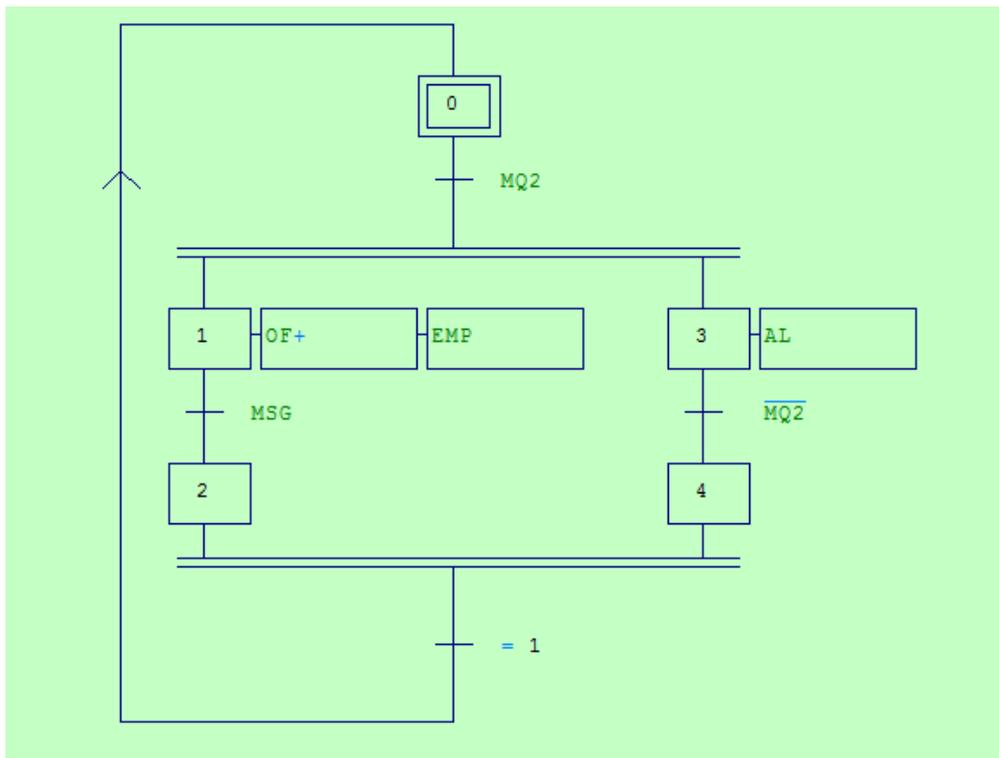
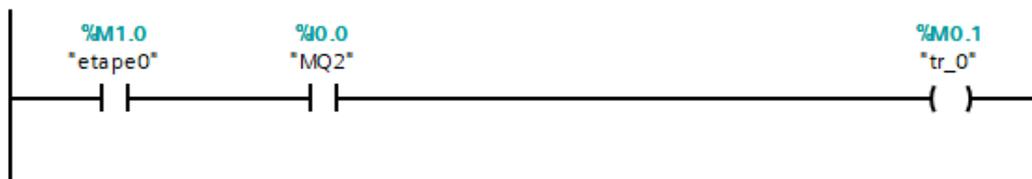
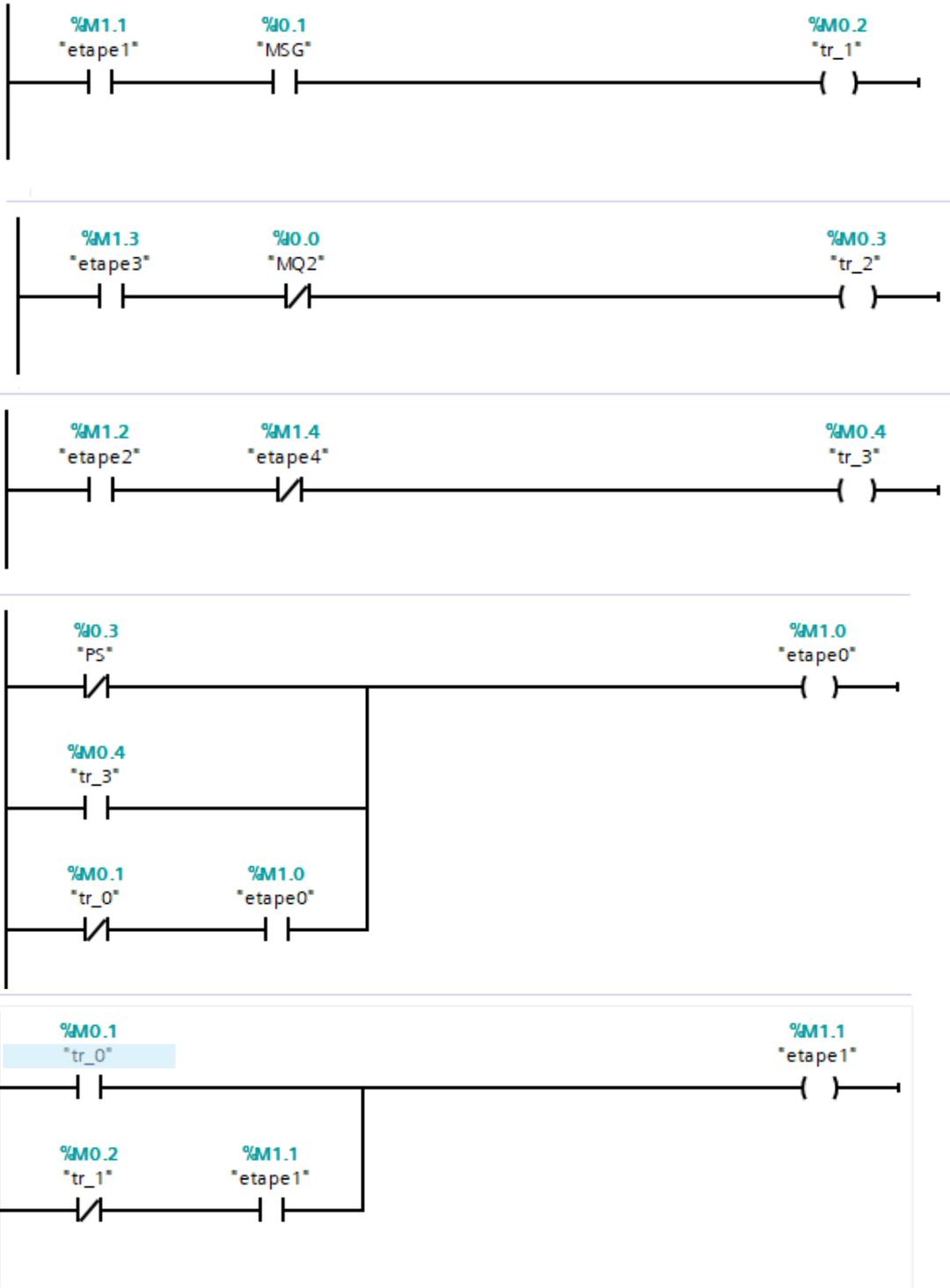


Figure 43 : Grafctet de gaz

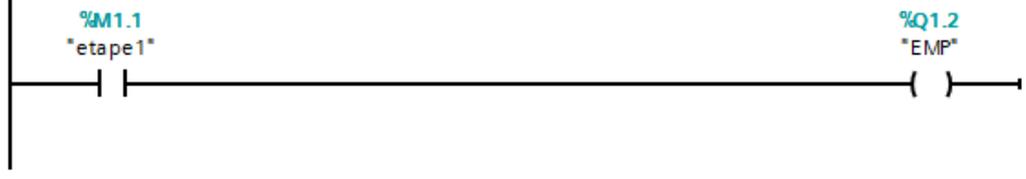
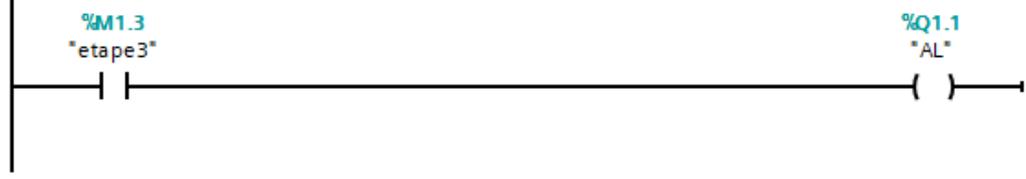
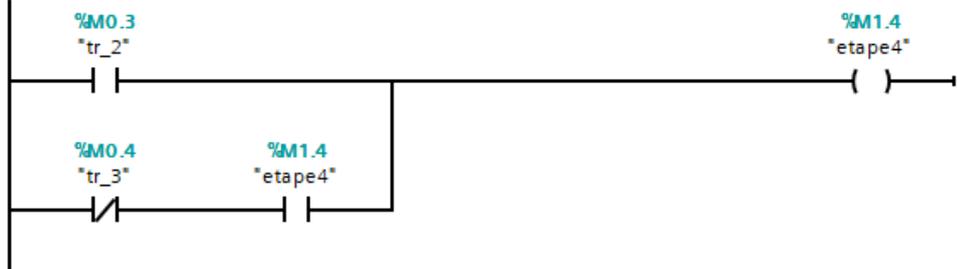
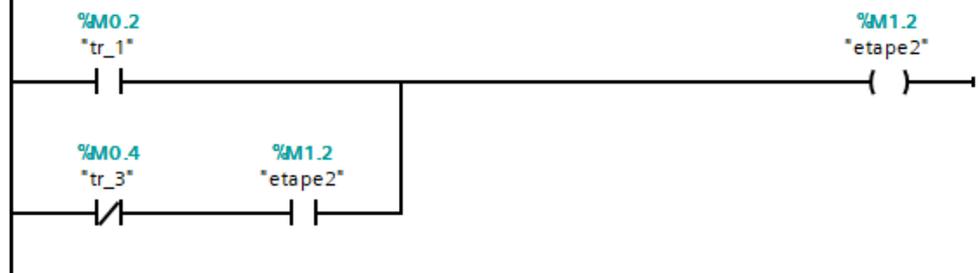
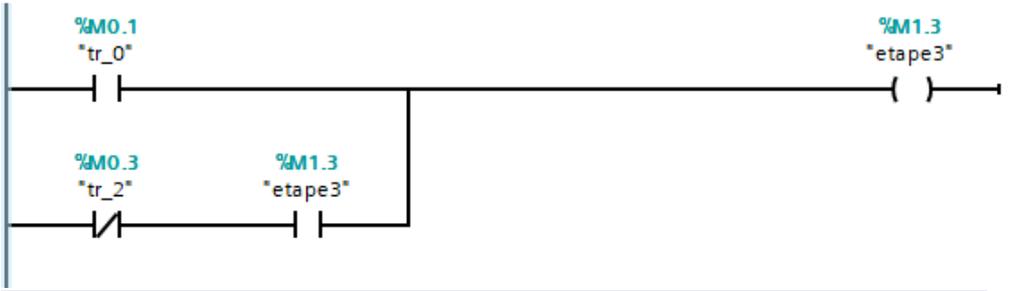
4.3.3 Ladder :



Chapitre 4 : Simulation et résultat



Chapitre 4 : Simulation et résultat



4.3.4 Résultat finale :

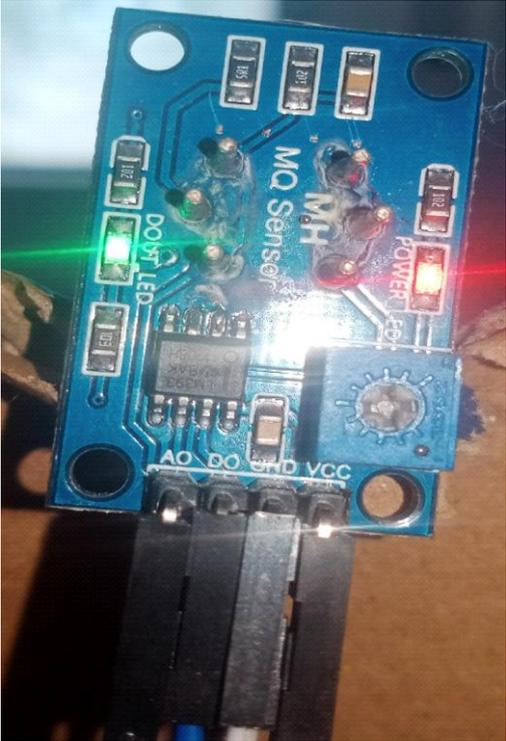


Figure 44 : Le capteur est détecté

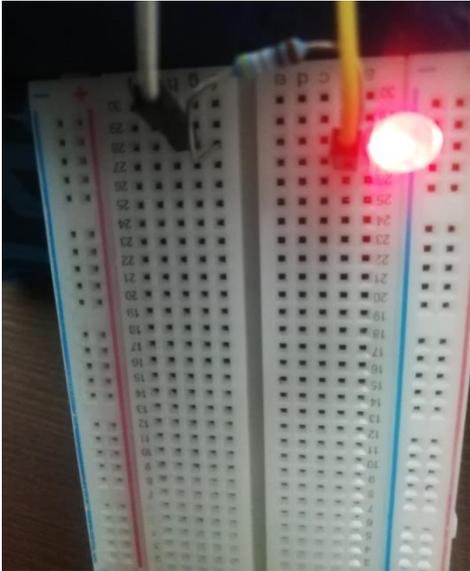


Figure 45 : Allumer la Led



Figure 46 : Ouverture la fenêtre

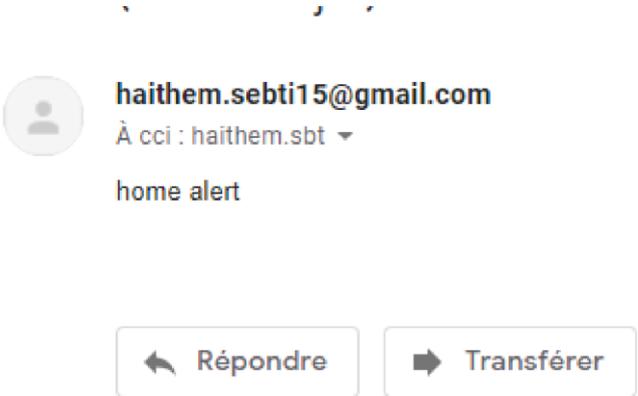


Figure 47: Envoyer le message



Figure 48 : Le cas de non détection de gaz



Figure 49 : Eteindre la Led

4.4 Résultat de température :

4.4.1 Cahier de charge :

Dans ce projet, nous avons mesuré la température de la maison à l'aide d'un DHT11 qui affiche la température et l'humidité

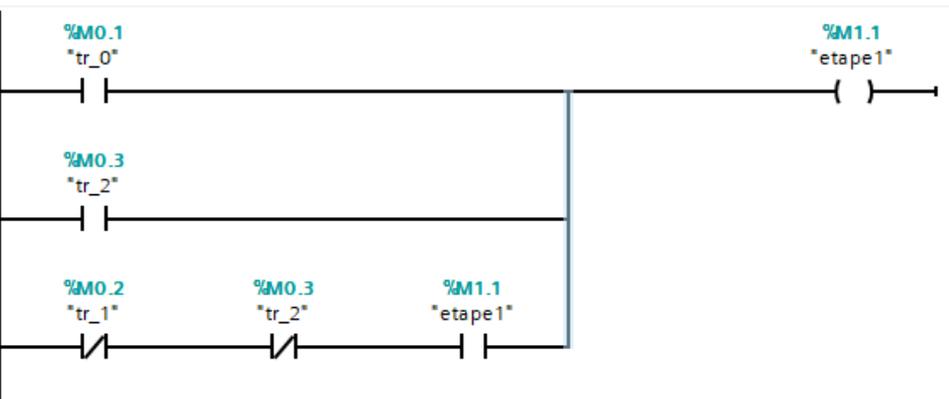
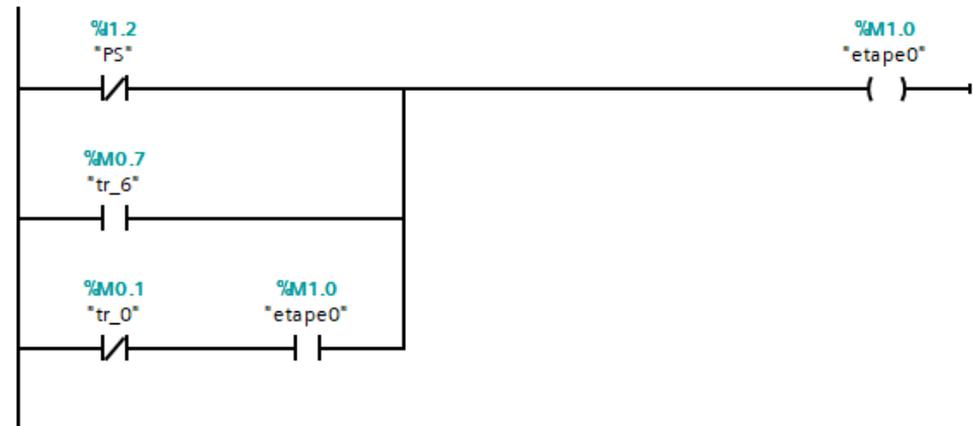
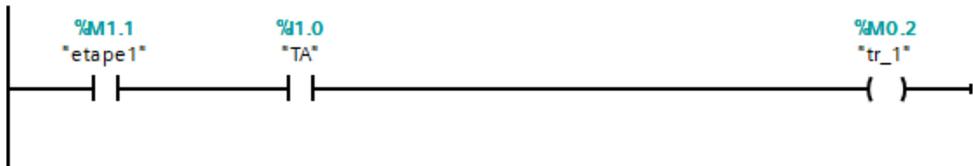
Nous avons considéré deux cas :

- $T < 35(TB)$ la maison affiche une température normale sans danger

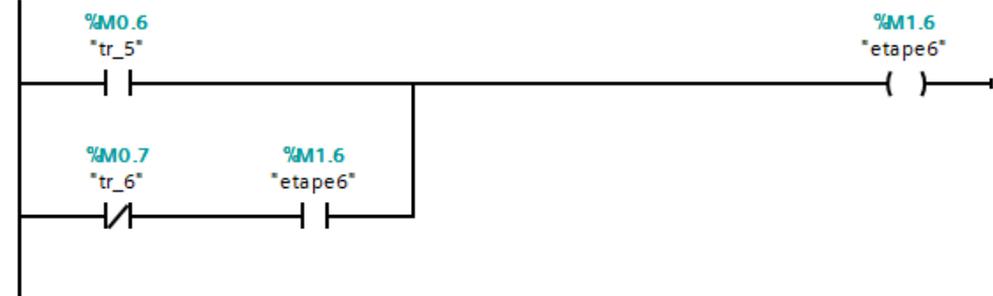
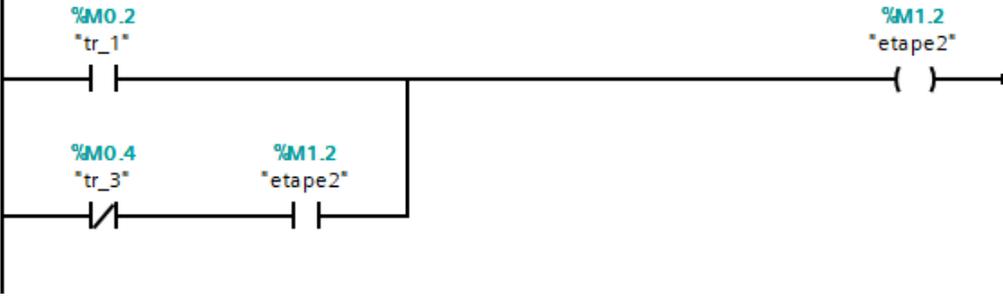
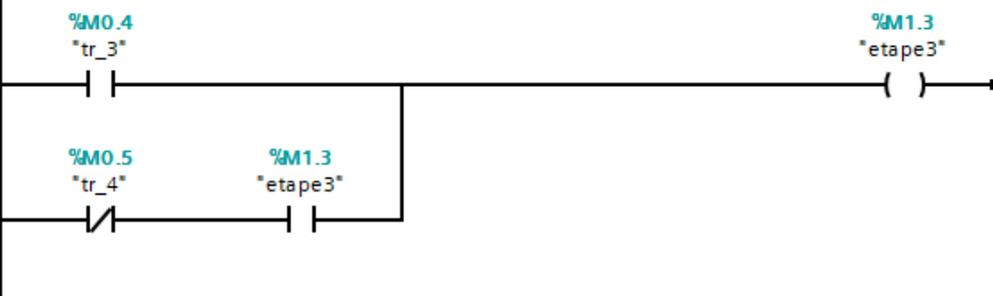
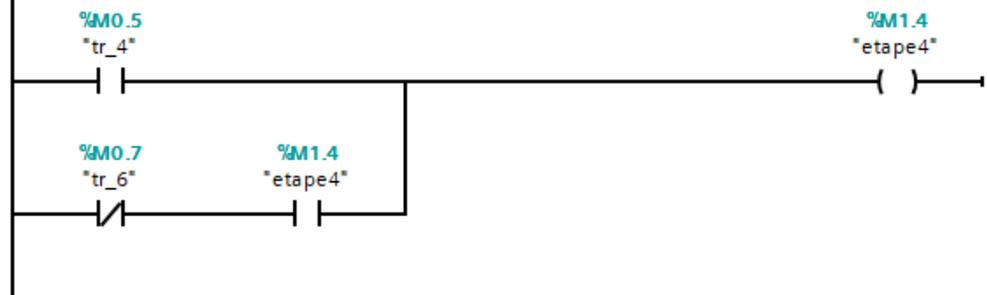
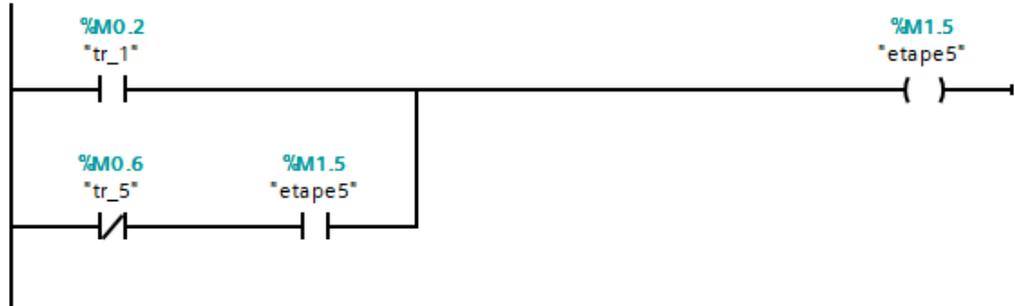
- $T > 35(TA)$ il y a un danger ou un risque d'échauffement donc le Led sera allumé et en même temps un message est envoyé au propriétaire pour l'avertir.

4.4.2 Grafct :

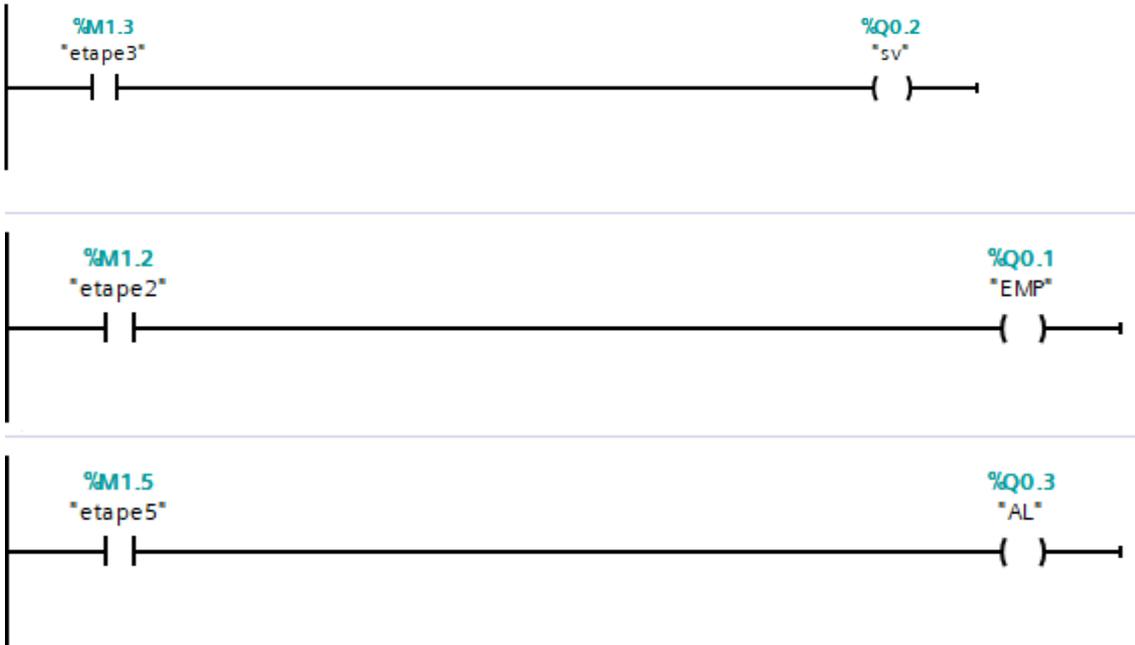
Chapitre 4 : Simulation et résultat



Chapitre 4 : Simulation et résultat



Chapitre 4 : Simulation et résultat



4.4.4 Résultat finale :



Figure 51 : Affichage de la température

Chapitre 4 : Simulation et résultat



Figure 52 : Le message envoyé pour la température

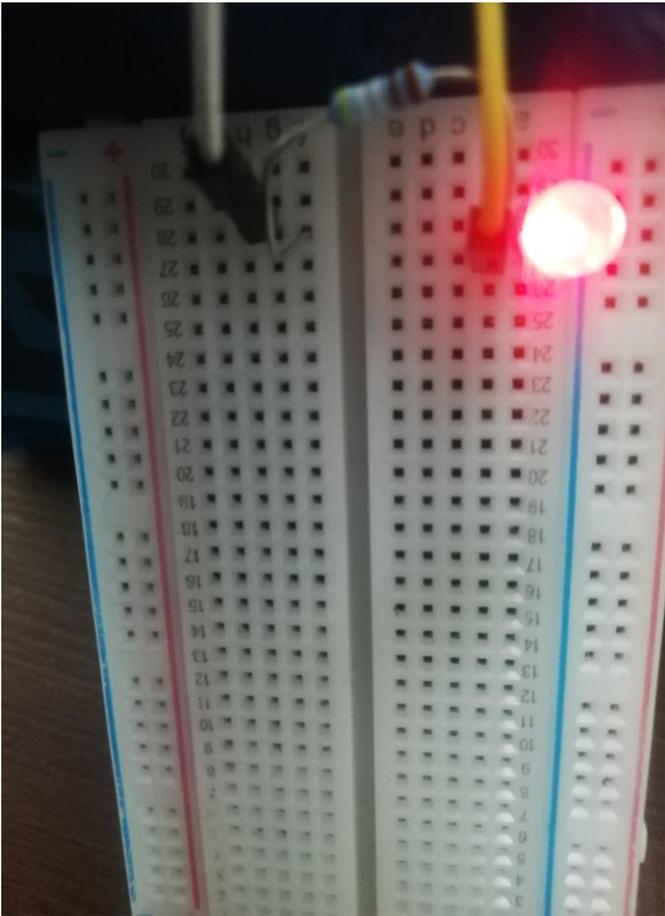


Figure 53 : allumer la Led

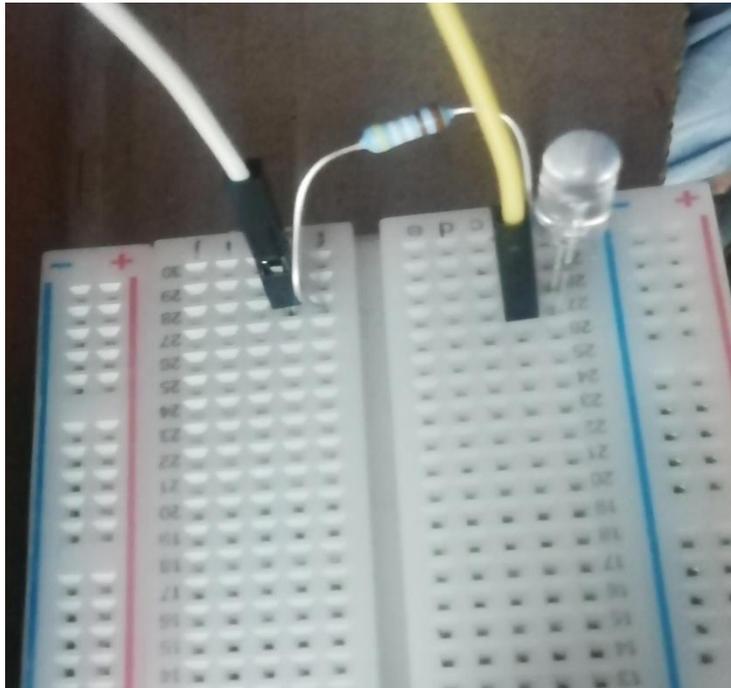


Figure 54 : Eteindre la Led

4.5 Résultat de reconnaissance facial :

4.5.1 Cahier de charge :

Afin d'ouverture les portes de façon sécurisée, nous avons installé une caméra pour vérifier et examiner le visage du propriétaire de la maison par la technique reconnaissance faciale.

Dans le cas où le visage du propriétaire Haithem est exposé, la porte s'ouvre automatiquement puis se referme.

Si le visage n'est pas reconnu, le propriétaire de la maison a le choix soit de donner l'autorisation d'ouverture ou de refuser.

4.5.2 Grafctet :

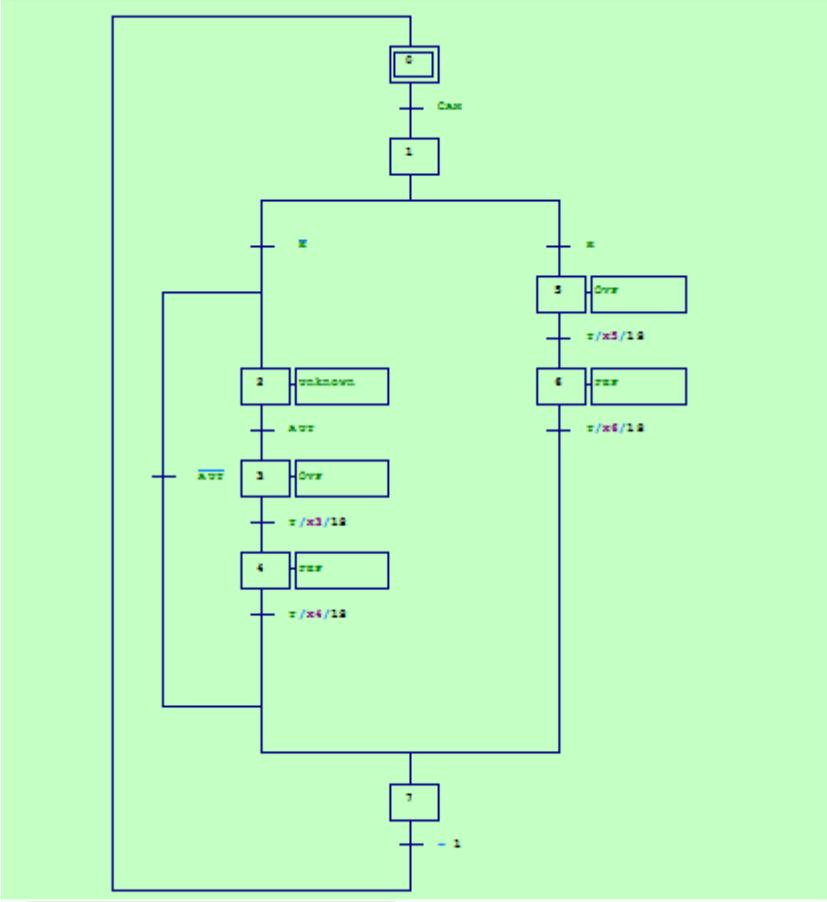
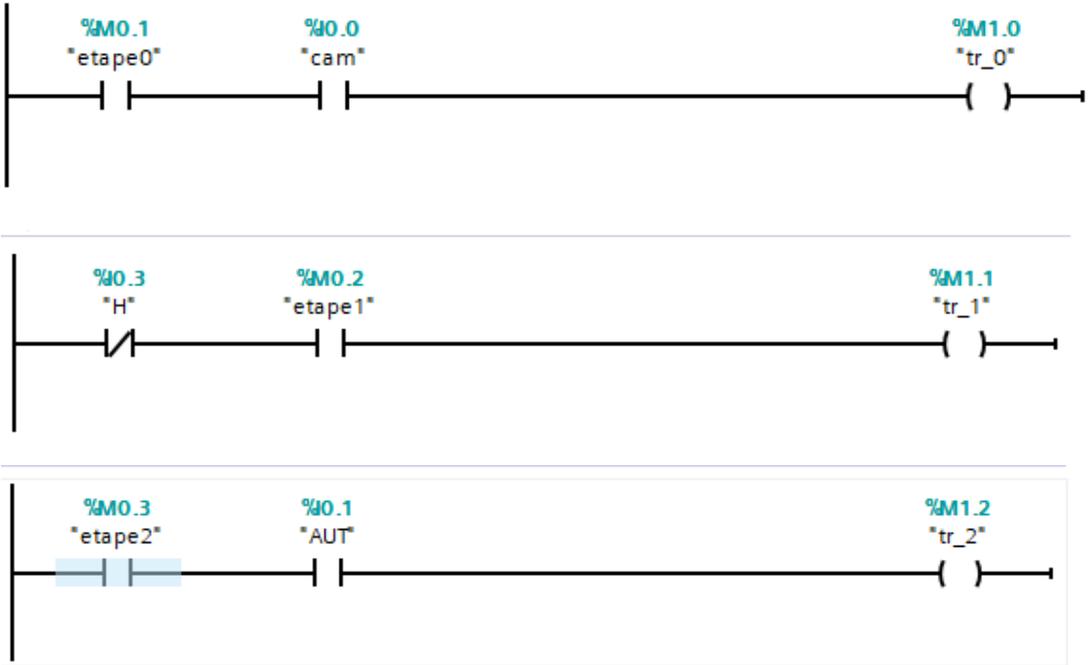
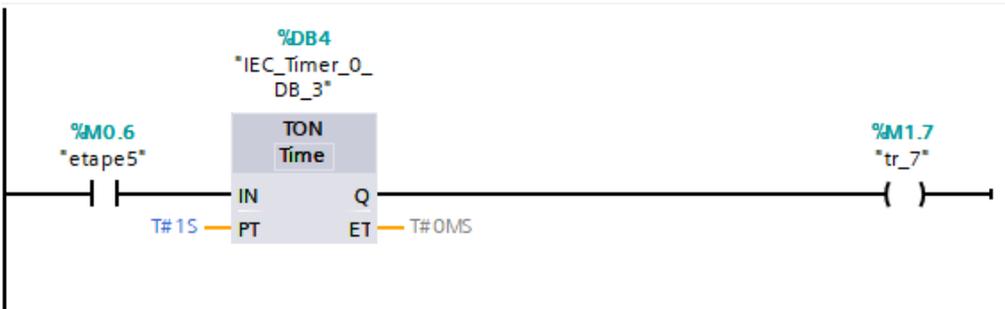
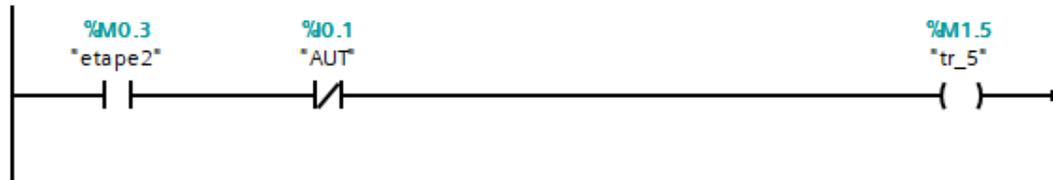
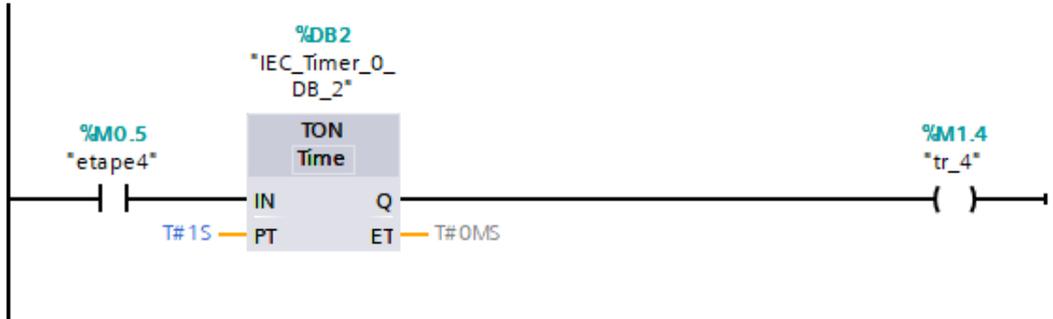
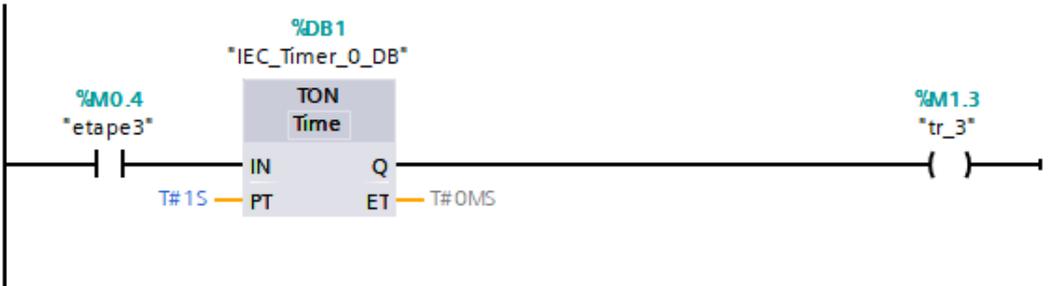


Figure 55 : grafcet de la Cam

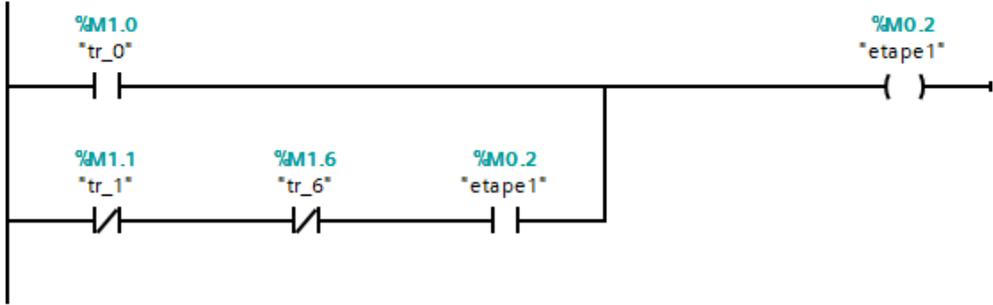
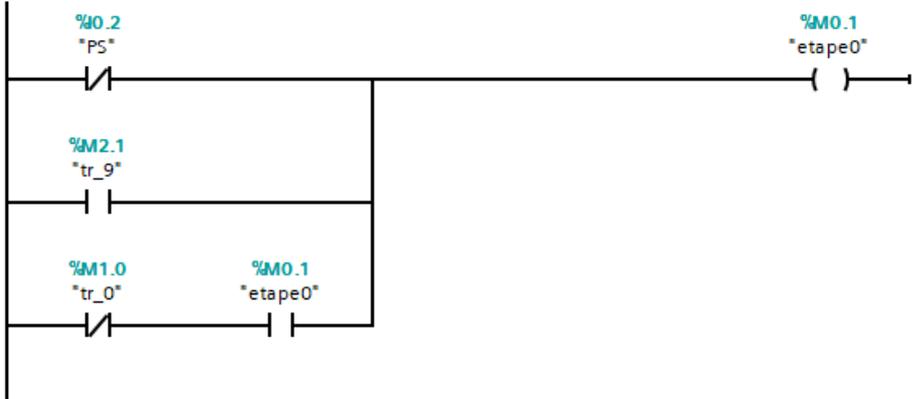
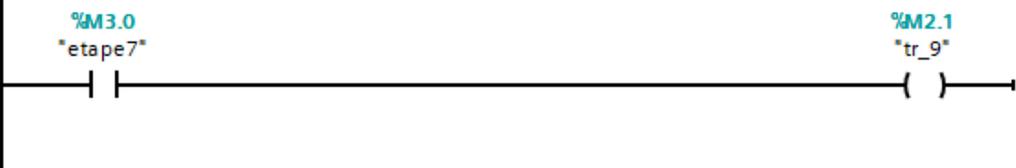
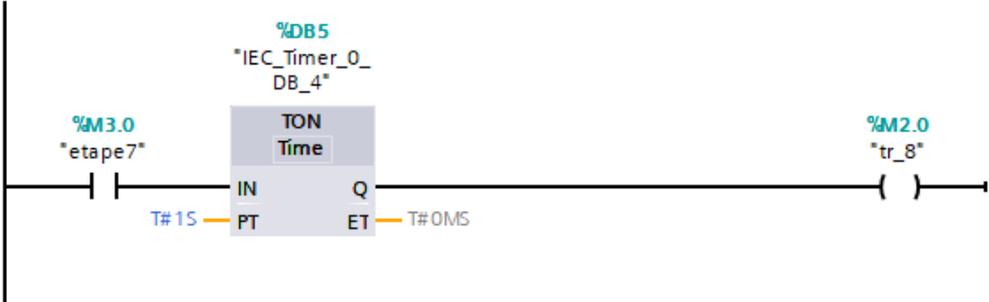
4.5.3 Ladder :



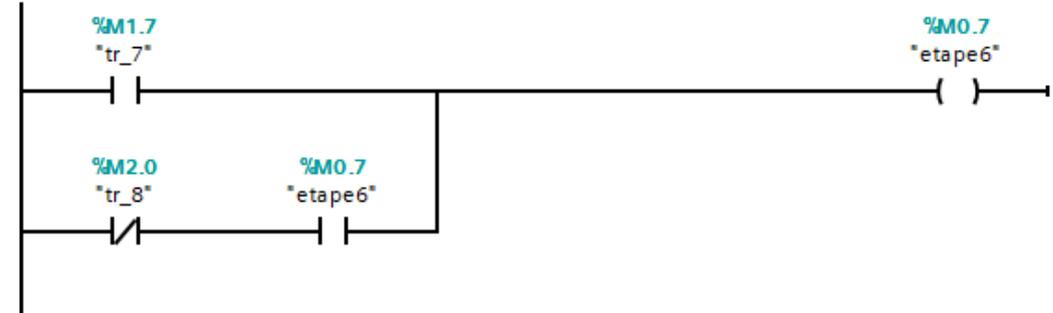
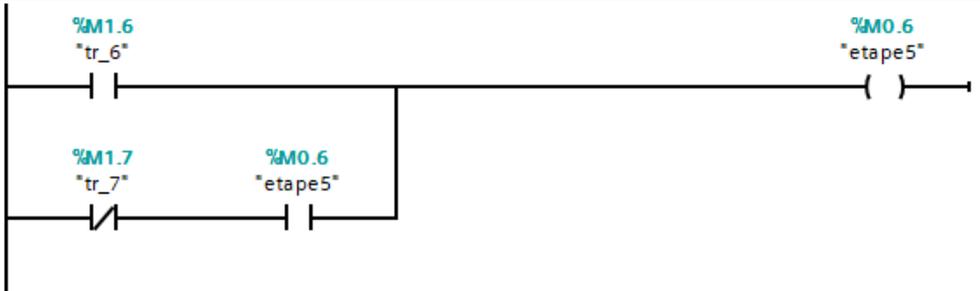
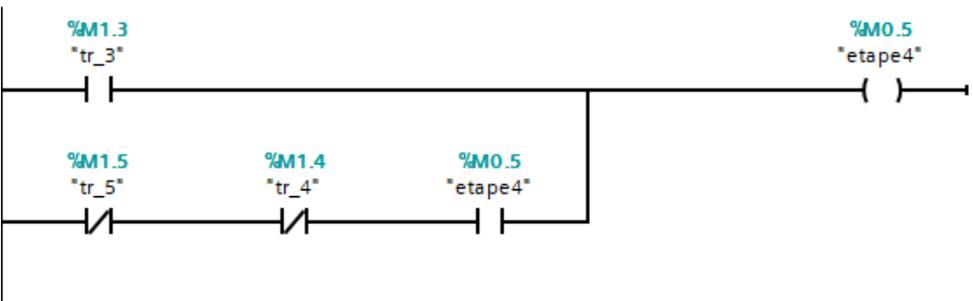
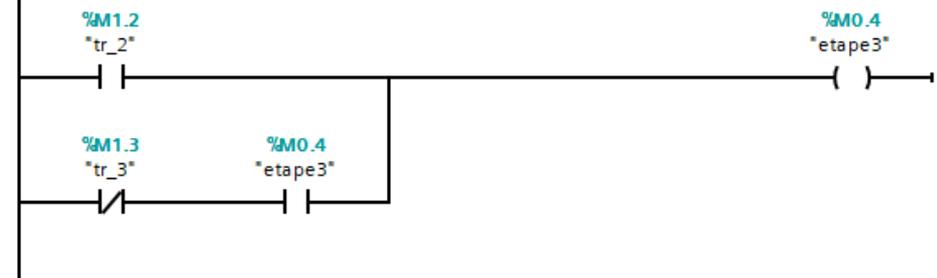
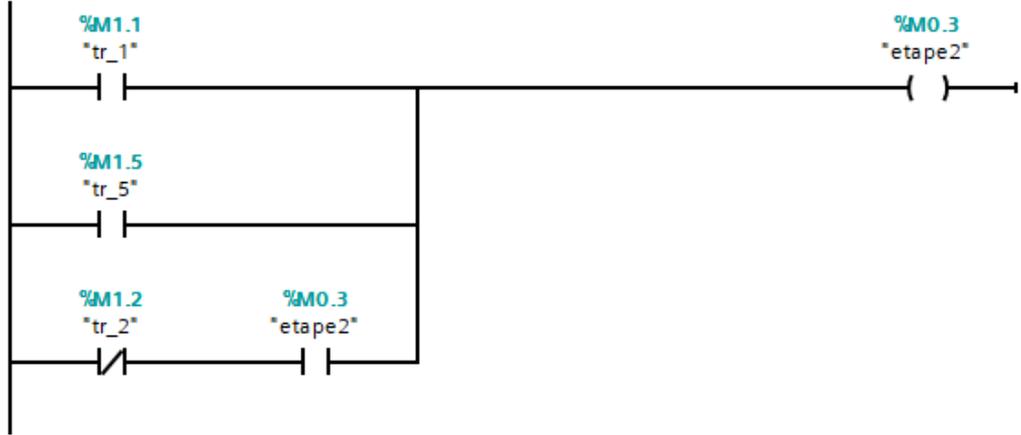
Chapitre 4 : Simulation et résultat



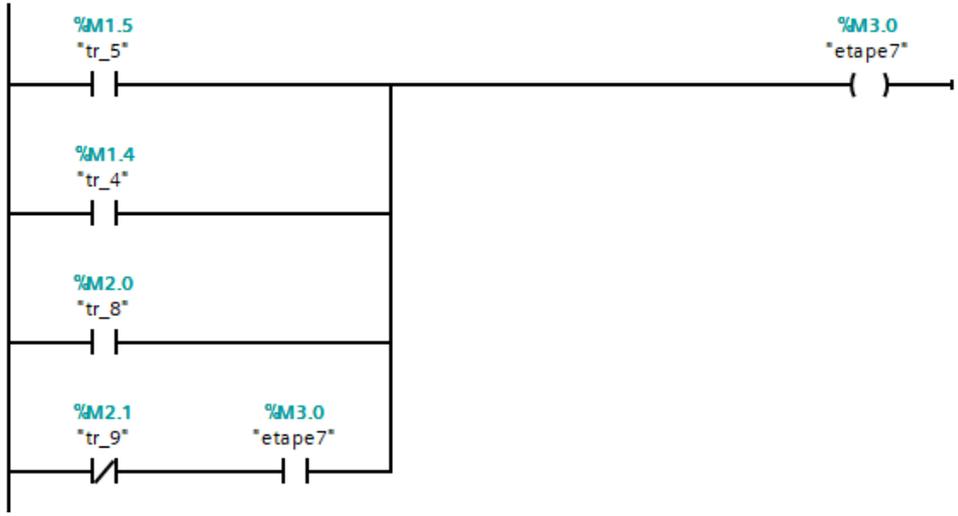
Chapitre 4 : Simulation et résultat



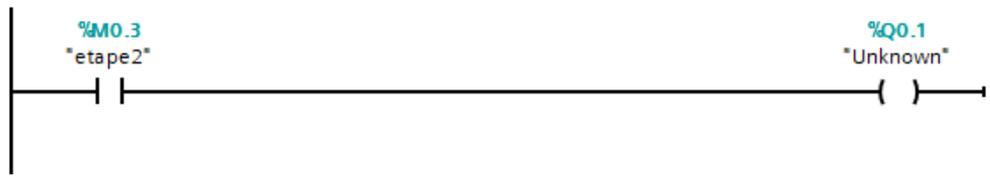
Chapitre 4 : Simulation et résultat



Chapitre 4 : Simulation et résultat



Résumé :



4.5.4 Résultat finale :



Figure 56 : la Cam détecte le visage de Haithem

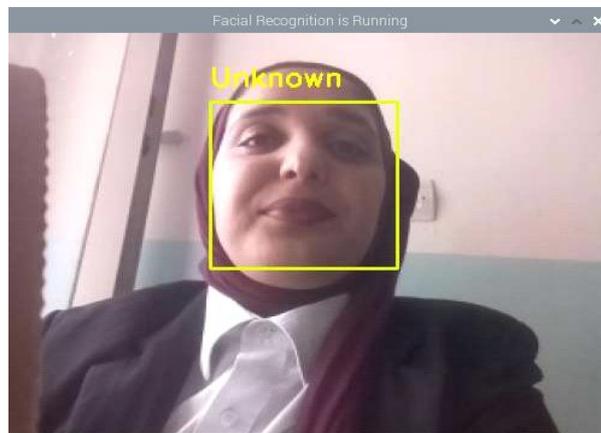


Figure 57 : La Cam détecte un visage inconnu



Figure 58 : Ouverture de la porte

Chapitre 4 : Simulation et résultat

4.6 Résultat de camera surveillance :

Pour la sécurité de la maison nous avons installé une caméra de surveillance avec un système qui conserve des enregistrements vidéos sur un disque dur ou une carte SD selon la durée voulue.

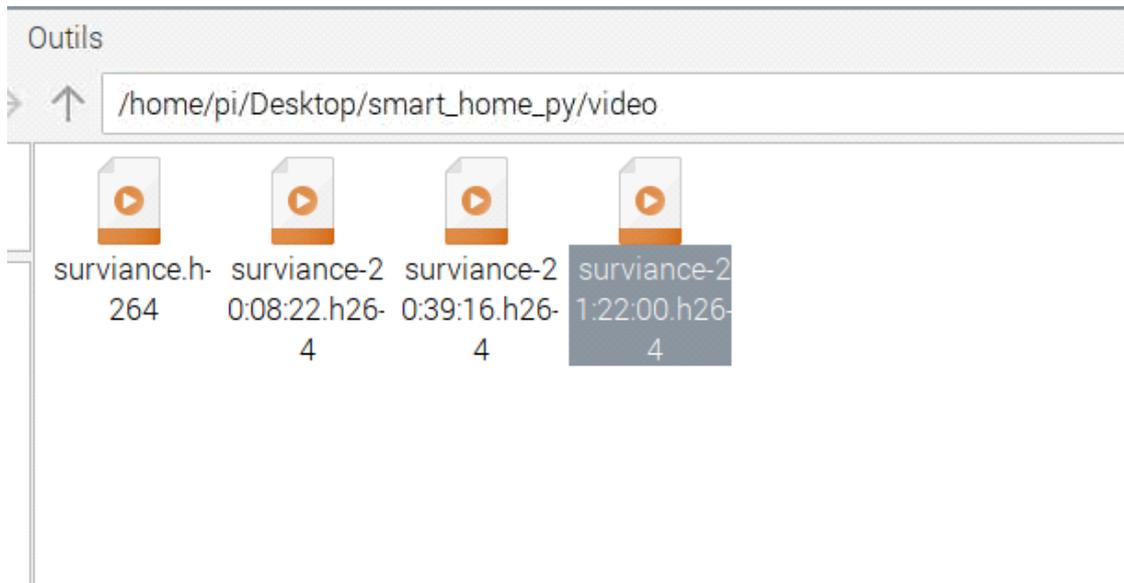


Figure 59 : enregistrement des vidéos de camera surveillance

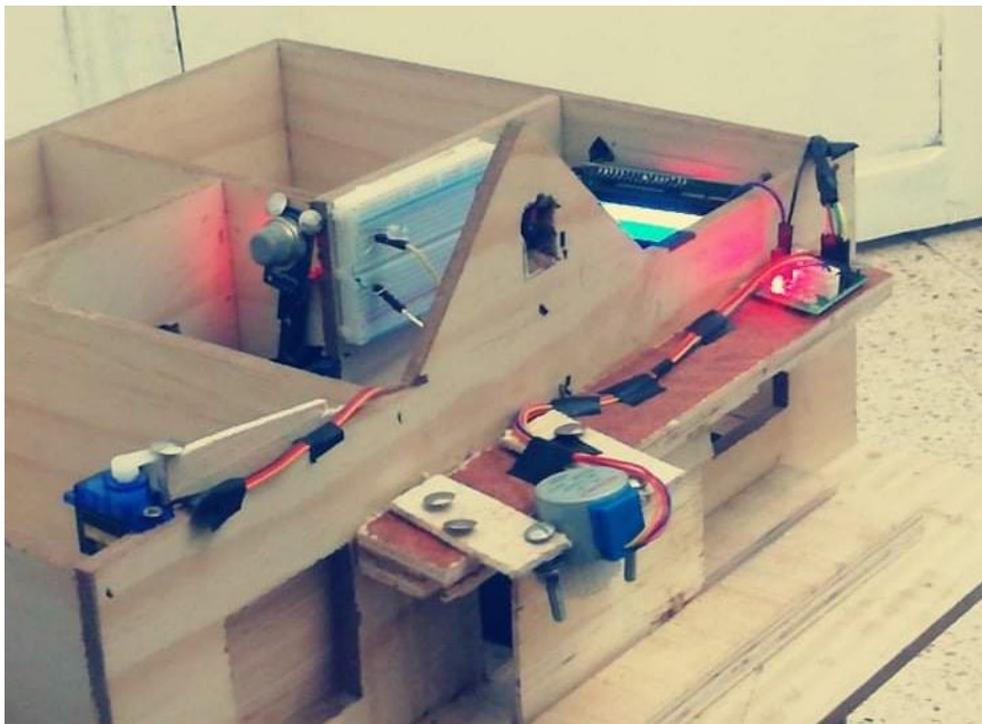


Figure 60 : Le montage général du notre maison intelligente

Conclusion générale

Notre projet consiste à construire et de contrôler la maison intelligente (surveillance, gestion de l'énergie, sécurité, confort, etc.).

Tout d'abord, nous avons donné un aperçu de la carte Raspberry Pi, le langage Python, Avec une description globale des composants utilisés, Ensuite, poursuite du développement de notre modèle (maison miniature) équipé de divers capteurs et actionneurs (moteur pas à pas, chambre, servomoteur, etc., capteur de température et d'humidité).

Dans ce projet, nous avons réussi à construire un modèle « Smart Home » basée sur une carte Raspberry pi3 et son langage de programmation Python.

Notre projet consistait à utiliser :

- Un servomoteurs pour l'ouverture et la fermeture des portes
- Un capteur DHT11 pour contrôler la température et l'humidité de la maison.
- Une caméra de surveillance de notre maison

Le résultat obtenue montre l'efficacité de notre approche grâce à l'implémentation de plusieurs scénarios via Grafset où tout évènement indésirable pouvant affecté la sécurité des maisons étaient pris en compte. Grace au langage Python, la programmation de ces cahiers de charges étaient un succès remarquable car nous avons pu utiliser plusieurs algorithmes en même temps tel que la détection faciale et les différentes instructions suite à une lecture des données en temps réel. En termes de perspectives, nous pourrons remplacer l'algorithme de détection faciale par la voix et aussi l'empreinte digitale. Nous pourrons aussi commander à distance l'éclairage de la maison et démarrer des machines à distance comme par exemple la machine à laver qui offre plus de confort aux individus.

Bibliographie

1. A.HoucineAlgérie Smart-Home, Poly-Automatic ,<https://polyautomatic.com/>.
2. amber ,Kookye , Design a smoke detector through a raspberry pi board and MQ-2 smoke sensor ,2017 ,<https://kookye.com/2017/06/01/design-a-smoke-detector-through-a-raspberry-pi-board-and-mq-2-smoke-sensor/>.
3. B.Gary ,DataScientest , Convolutional neural network ,2020 , <https://datascientest.com/convolutional-neural-network> .
4. .C.Diou, I2C BUS , Introducion au bus I2C , https://les-electroniciens.com/sites/default/files/cours/cours_i2c.pdf.
5. C.Barnatte , Explaining Computers , Raspberry Pi Servo Motor Control , 2020 , https://www.explainingcomputers.com/pi_servos_video.html.
6. Dhanush ,LearnelectronicindiaMaking Electronics Easy , Interfacing DHT11 sensor with Raspberry pi and LCD ,2020, <https://www.learnelectronicindia.com/post/interfacing-dht11-sensor-with-raspberry-pi-and-lcd>.
7. F.MOCQ ,Kubii,Framboise314, GPIO, <https://www.framboise314.fr/scratch-raspberry-pi-composants/gpio/> .
8. Frevra, Comment Débuter à Programmer en Python sur RaspberryPi , 2018, <https://www.frevra.com/fr/debuter-a-programmer-en-python-sur-raspberry-pi/#respond>.
9. .JDN, Perceptron retour sur l'ancêtre du machine learning , 2021 , <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501903-perceptron-retour-sur-l-ancetre-du-machine-learning/#:~:text=Le%20perceptron%20est%20un%20neurone,d%C3%A9finition%20de%20r%C3%A9seau%20de%20neurones>.
10. Hight–Tech Expérience LDLC, RaspberryPi3 Model B+ <https://www.ldlc.com/fiche/PB00246555.html>.
11. . Microcontrollerslab , MCP3008 8-Channel 10-Bit A/D Converters with SPI Serial Interface , <https://microcontrollerslab.com/mcp3008-8-channel->

- 10-bit-adc-converters-with-spi-serial-interface/.
12. P.CLEMENT, Python et Raspberry Pi , 2019 ,
https://www.framboise314.fr/livre-python-pour-le-raspberry-pi-paru-aux-editions-eni/#Lrsquoauteur_Patrice_CLEMENT.
 13. P.Monasse, K.Nadjahi,OPENCLASSROOMS, Découvrez les différentes couches d'un CNN,2021 , <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn>.
 14. Tutorials For RaspberryPi, Configurer et lire le capteur de gaz Raspberry Pi (MQ-X), <https://raspberrypi-tutorials.fr/configurer-et-lire-le-capteur-de-gaz-raspberry-pi-mq-x/>.
 15. . Thomas ,Raspberry ME , Moteur pas à pas ESP8266 NodeMCU (pilote 28BYJ-48 et ULN2003) ,2020 , <https://www.raspberryme.com/moteur-pas-a-pas-esp8266-nodemcu-pilote-28byj-48-et-uln2003/> .
 16. Timmy , Core Electronics , Face Recognition With Raspberry Pi and OpenCV , 2021 ,<https://core-electronics.com.au/guides/face-identify-raspberry-pi/>.
 17. T. Keldenich , Inside Machine Learning , Fonction d'activation, comment ça marche ? – Une explication simple , 2021 , <https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/>

.
..
..
.