

Projet de fin d'études

**Filière : La licence fondamentale en Informatique,
Electronique et Automatique « IEA »**
Département de physique

Réalisation d'une maison intelligente à base d'Arduino

Réalisé par :

EL YAHIAOUI Khadija

BOUKOUTAYA Abdel Adem

Encadré par :

Le président des encadrants Mr. OUADOU Mohammed

Soutenue devant les jurys :

Mr. OUADOU Mohammed

Professeur à la faculté des sciences Rabat

Mr MINAOUI Khalid

Professeur à la faculté des sciences Rabat

Mr. OUADOU Mourad

Doctorant à la faculté des sciences Rabat

Dédicace

A nos chers parents

Pour leur soutien, leur patience, leur sacrifice

Et leur amour, vous méritez tout éloge,

Vous qui avez fait de nous ce que nous sommes maintenant.

Nous espérons être comme vous l'avez souhaité.

Que dieu vous garde et vous bénisse.

Nous dédions aussi ce travail à nos chers frères et sœurs,

Pour leurs affections et leur encouragement

Qui ont toujours été pour nous des Plus précieux.

A ceux qui ont sacrifié, patienté, donné tout pour continuer
leurs études.

Pour être là entre nous, apprendre plus et plus pour éclairer le
monde par leur savoir,

À tous mes amis à tous ce qui nous a aidés.

A tous ceux que nous aimons nous dédions ce travail.

RESUME

La domotique est l'ensemble des objets connectés qui rendant la maison elle-même connectée, ou communicante. On parle même parfois de maison intelligente, les différents objets connectés permettant à la maison de réagir automatiquement en fonction d'événements (ouvrir/fermer les portes, allumée/éteindre la lumière à distance lors d'une absence imprévue...).

Dans le cadre de notre projet de fin d'étude en Informatique, Electronique, Automatique, nous avons choisi de travailler sur la création d'un système domotique sans fil.

L'objectif est de créer ce système à l'aide d'Arduino et de petits systèmes embarqués à base de microprocesseur et d'antenne pour permettre une production low-cost, reproductible chez soi et modifiable à souhait, tout ça avec un budget convenable.

Ce projet se découpe en deux parties faisant appel à nos compétences en Informatique et en Electronique. La première se focalise sur la construction d'une maquette avec les modules utilisés (Capteur, Modules WIFI ...) et la deuxième sur la création et l'implémentation du système et le rendre commandable via un smartphone ou bien un ordinateur.

Dans un premier temps, nous expliquerons la domotique brièvement. Dans un second temps, nous préciserons tous les modules utilisés ainsi les logiciels nécessaires. Enfin, nous terminerons sur la création d'une maquette de maison connectée et le déplacement des modules en commandant tout ça grâce à une page WEB.

Table des matières

RESUME.....	2
INTRODUCTION GENERALE.....	5
CHAPITRE 1 : LA DOMOTIQUE	6
I. Introduction de la domotique :.....	7
II. Définition de la domotique :	7
III. Que peut-on faire grâce à la domotique ?	7
1. Le confort :.....	7
2. Les économies d'énergie :	8
3. Communication :	8
4. La sécurité :.....	8
IV. Combien coute la domotique ?	9
V. But du projet :.....	10
CHAPITRE 2 : MATERIELS ET LOGICIELS UTILISES	11
I. Introduction :.....	12
II. Matériels :	12
1. Carte Arduino Méga :.....	12
2. ESP8622 12E :.....	15
3. Servomoteur :	16
4. Capteur de flamme :.....	18
5. Capteur humidité & température DHT11 :.....	21
6. Capteur de gaz MQ6 :.....	23
7. Capteur de mouvement PIR :.....	24
8. Ecran LCD avec un I2C :	25
9. Buzzer :.....	26
I. Logiciels :	26
1. L'Arduino :	26
CHAPITRE 3 : DESCRIPTION DU PROJET	28
I. Introduction :.....	29
II. Fabrication de la maison :	29
1. Structure de la maison :.....	29
2. Périphériques utilisées et leurs positions :.....	29
III. Automatisation de la maison :.....	30
1. Capteur de mouvement PIR :.....	30
2. Capteur d'humidité :.....	30
3. Capteur de flamme / de gaz :.....	30
4. Servomoteur :	31

IV. Centralisation des commandes : Interface Web	31
Conclusion Générale :	35
Annexes et Programmes	36

INTRODUCTION GENERALE

La pensée à la technologie de télécommunication, nous ramène immédiatement à l'esprit les techniques de la communication vocale, l'envoi et la réception des SMS, l'internet, et tout ce qui fonctionne en utilisant cette technologie moderne introduite dans notre vie quotidienne.

Cette technologie bouleverse notre mode de vie, pourtant nos maisons continuent d'être conçues comme il y a trente ans, sans tenir compte de ces évolutions, comme si l'endroit où nous passons plus de la moitié de notre temps n'en valait pas la peine, l'électronique n'y a pas sa place.

Qui accepterait d'acheter une voiture neuve qui obligerait à fermer à clé une à une toutes les portières ou à actionner les vitres manuellement? Qui voudrait d'un véhicule dépourvu de système de sécurisation des passagers? Ce que nous refusons pour nos voitures, nous semblons l'accepter pour nos maisons.

Nos maisons sont dépourvues de système de sécurité, le taux de cambriolage des habitats est élevé. Le nombre de victime de gaz est important, c'est là où la Dominique fait surface pour trouver des solutions à ces problèmes.

Notre but à Travers ce projet de la Maison Intelligente est de permettre d'évaluer l'apport de la domotique dans la gestion d'énergie et l'optimisation du confort dans l'habitat.

Le but principal réside dans la possibilité de réaliser une pièce domotique avec des modules simples, modifiable à souhait et à la portée de tout le monde.

CHAPITRE 1 : LA DOMOTIQUE

I. Introduction de la domotique :

La gestion centralisée des équipements techniques et multimédia d'un logement

La Domotique désigne la **gestion centralisée des équipements techniques** (chauffage, sécurité, éclairage, volet roulant, etc.) et du **multimédia** dans le résidentiel. Elle vise à apporter des fonctions de confort, de sécurité, d'économie d'énergie et de communication aux maisons ou appartements équipés.

Le terme Smart Home est de plus en plus utilisé. Il a deux définitions. Il peut s'agir simplement de la traduction anglophone de la domotique. D'autres lui donne une portée supplémentaire, le logement devient intelligent. Cette intelligence se caractérise par des objets et des services connectés qui viennent s'ajouter aux équipements techniques gérés par la domotique.

Les scénarios envisagés peuvent ainsi dépasser la mise en relation de capteurs et d'actionneurs du logement. Ils s'ouvrent d'avantage au monde de l'Internet des Objets, de l'informatique ubiquitaire ou à l'Intelligence ambiante. Ils permettent de mettre en relation des services extérieurs avec les objets du logement et ses occupants.

II. Définition de la domotique :

La domotique est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans les bâtiments, plus ou moins « interopérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prises électriques, etc.). La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut retrouver dans les maisons, les hôtels, les lieux publics, etc.

III. Que peut-on faire grâce à la domotique ?

On peut dire que la domotique trouve sa place dans trois domaines principaux en particulier.

1. Le confort :

Bien sûr, le fait d'automatiser sa maison a un véritable apport sur le confort qu'on y trouve. Plus besoin de se prendre une averse pour ouvrir le portail en rentrant à la maison, plus besoin de prendre froid en ouvrant les volets le matin, et fini les retours de week-end dans une maison toute froide.

Aujourd'hui, une maison intelligente est capable de savoir quand vous rentrez à la maison (grâce à votre smartphone par exemple), et donc d'ouvrir le portail avant même que vous

n'arrivez. Les volets peuvent s'ouvrir et se fermer au rythme du soleil, et peuvent même aller jusqu'à s'adapter à la saison et la température pour laisser entrer la lumière et la chaleur du soleil l'hiver, ou au contraire conserver le frais l'été en fermant les volets des fenêtres exposées au soleil. De la même façon, votre maison sait quand vous êtes présent, et peut ainsi adapter elle-même votre chauffage pour que la maison soit toujours à la température idéale pour vous. Il est même possible de diffuser automatiquement votre Play List musicale préférée à votre réveil, ou quand vous rentrez à la maison. Pendant ce temps, un robot peut passer l'aspirateur dans la maison à votre place, et le système d'arrosage automatique arrosera votre jardin, tout en tenant compte des prévisions météo des prochains jours, histoire de ne pas arroser inutilement.

2. Les économies d'énergie :

En gérant les volets selon la saison, ainsi que le chauffage, le système domotique vous permet d'économiser de l'énergie, et donc de l'argent, même si au départ on ne recherchait que le confort en plus. La consommation d'énergie peut être suivie très finement, qu'il s'agisse de votre consommation d'électricité, d'eau, ou même de gaz.

Vous partez de la maison ? Le simple fait d'activer l'alarme en partant va passer le chauffage en mode éco, et éteindre toutes les lampes et les appareils restés en veille, réduisant ainsi votre consommation d'énergie en votre absence. Et ceci sans aucune action de votre part. C'est cela la maison intelligente !

3. Communication :

Un système domotique permet la communication non seulement à l'intérieur de la maison, mais aussi à l'extérieur. La technologie Internet interviendra de plus en plus pour la commande à distance par La certains utilisateurs. Vous ne devez même pas être à la maison pour commander vos appareils. Un simple coup de fil ou un sms vous permettra par exemple de régler le chauffage à distance, d'activer une simulation de présence ou de lancer le lave-vaisselle ou le lave-linge. Pratique, non ?

4. La sécurité :

Les automatismes que nous avons vus plus haut peuvent tout à fait contribuer à la sécurité de vos biens, en réalisant ce qu'on appelle une simulation de présence : même en votre absence, les volets continuent de s'ouvrir, de la musique peut être diffusée dans la maison, et des lumières allumées aléatoirement. Ainsi, de l'extérieur, il devient très difficile de savoir si la maison est inoccupée, ce qui dissuade de nombreux cambrioleurs.

Une détection de fuite d'eau peut couper automatiquement l'arrivée d'eau afin d'éviter de gros dégâts.

Mais la sécurité, c'est également la sécurité des personnes : en cas de détection d'incendie, par exemple, il est tout à fait possible d'ouvrir automatiquement les volets, déverrouiller les portes, et éclairer le chemin de la sortie pour faciliter l'évacuation. Exemple d'une mise en scène d'un « simple » détecteur de fumée, couplé à un système domotique.

Dans le cas de personnes dépendantes, telles que des seniors ou des personnes à mobilité réduite, il est également possible par exemple de prévenir un proche ou les services de secours en cas de chute ou tout évènement anormal.

IV. Combien coute la domotique ?

Pour les marques connus dans le domaine des maisons connectées, l'installation domotique ça commence par un budget de 10000€ au minimum.

Après une recherche sur internet on a trouvé des chercheurs dans ce domaine qui font des efforts pour la réalisation des maisons intelligentes avec un budget moins cher.

Exemple : << Extrait d'un ouvrage >>

Pour commencer il faudra un box domotique pour piloter la maison. On trouve des box allant de 150 à 600€ environ, en restant dans le « grand public », car il y a beaucoup plus cher. Disons que pour 200€ on trouve des box très corrects, le prix n'étant pas spécialement un gage de qualité.



Fig.1 : La VeraEdge, de Vera Control LTD, bien que petite, est très complète et ne coute que 190€.

Ensuite, le budget va surtout dépendre du nombre de périphériques qu'on va mettre en place, et de la technologie utilisée, certaines étant plus couteuses que d'autres.

Pour récapituler :

- Box domotique : 200€
- Gestion des volets : 200€
- Gestion du chauffage : 200€
- Gestion de l'éclairage : 500€
- Gestion des appareils : 250€
- Sécurité : 550€

Soit un total de 1900€

Notre remarque :

Enfin, le prix des périphériques cité ici est une moyenne, il est possible de trouver moins cher. Bref, tout ça pour vous dire que même 1900€, dans le type d'installation domotique dont nous parlons dans notre projet, est aussi un budget qui est chère quand même, mais il est tout à fait possible de commencer avec budget moins chère tout en ayant des fonctions déjà

intéressantes, qu'on pourra faire évoluer plus tard, et c'est le gros avantage des solutions sans fil.

V. But du projet :

Notre projet a pour but de créer un système à l'aide d'Arduino pour réaliser une maison connectée grâce à des modules moins chère par rapport à ce qu'on a vu au-dessus aussi par rapport au prix que l'on peut voir présenter à la TV par exemple (publication des marques de la domotique [Home Auto, Samsung, IAM et Orange...]), et plus simple qui sont disponible partout avec des prix convenable.

Partie financière :

Pour notre projet le budget dépendre du nombre de périphériques qu'on a utilisés.

Modules	Nombres	Prix
Capteur de mouvement PIR	X 1	45 DH
Capteur de gaz MQ6	X 1	70 DH
Capteur de flamme	X 1	40 DH
Capteur d'humidité	X 1	50 DH
Arduino Méga	X 1	230DH
ESP8266 12E	X 1	150DH
Servomoteur 9g	X 1	50 DH
LCD 1602 I2C	X 1	75 DH
LED 5mm	X 5	05 DH
Buzzer	X 1	30 DH

Tableau 1 : Les périphériques utilisés et leurs prix

Soit un total un peu prés de : 800 DH

CHAPITRE 2 : MATERIELS ET LOGICIELS UTILISES

I. Introduction :

Comment faire des montages électroniques, simplement, en utilisant un langage de programmation? La réponse, c'est le projet.

Arduino qui l'apporte, celui-ci a été conçu pour être accessible à tous par sa simplicité. Mais il peut également être d'usage professionnel, tant les possibilités d'application sont nombreuses. Ces cartes polyvalentes sont donc parfaites pour nous, débutants, qui ne demandent qu'apprendre et progresser. Grâce à cette carte ces fonctions sont réalisées par des capteurs, des actionneurs, des automates et plus généralement par des équipements électriques et/ou électroniques. Elle permet de contrôler son logement en local ou à distance depuis un ordinateur, un téléphone, une tablette tactile ou une télévision.

II. Matériels :

1. Carte Arduino Méga :



Fig.2 : Carte Arduino Méga 2560



Fig.3 : Carte microcontrôleur ATmega2560

Arduino désigne un écosystème libre comprenant des cartes (Arduino Uno, Arduino Leonardo, Arduino Méga, Arduino Nano...), des logiciels (notamment l'IDE Arduino), ou encore des bibliothèques. Ces systèmes d'électronique programmable permettent de construire des projets facilement, et d'aborder tant l'approche électronique que l'approche logicielle.

La carte Arduino Méga 2560 est une carte à microcontrôleur basée sur un ATmega2560. Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur; Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

La carte Arduino Méga 2560 est compatible avec les circuits imprimés prévus pour les cartes Arduino Uno, Duemilanove ou Diecimila.

N.B : le Datasheet de la carte Arduino Méga se trouve dans l'annexe

La carte microcontrôleur ATmega2560 :

Le ATmega640 / 1280/1281 / 2560/2561 est un CMOS à faible puissance microcontrôleur 8 bits basé sur l'AVR amélioré l'architecture RISC. En exécutant des instructions puissantes dans

un seul cycle d'horloge, le ATmega640 / 1280/1281 / 2560/2561 atteint des débits approchant 1 MIPS par MHz permettant au concepteur du système pour optimiser la consommation d'énergie par rapport à la vitesse de traitement.

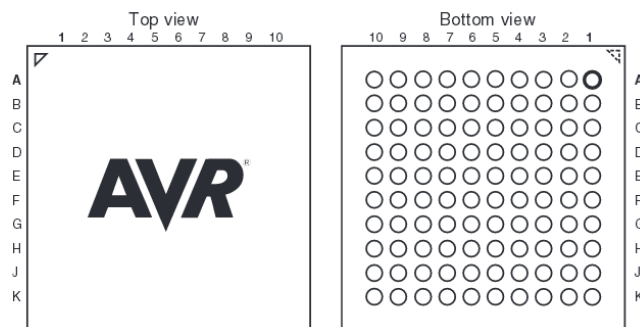


Fig.4 : CBGA-pinout ATmega640/1280/2560

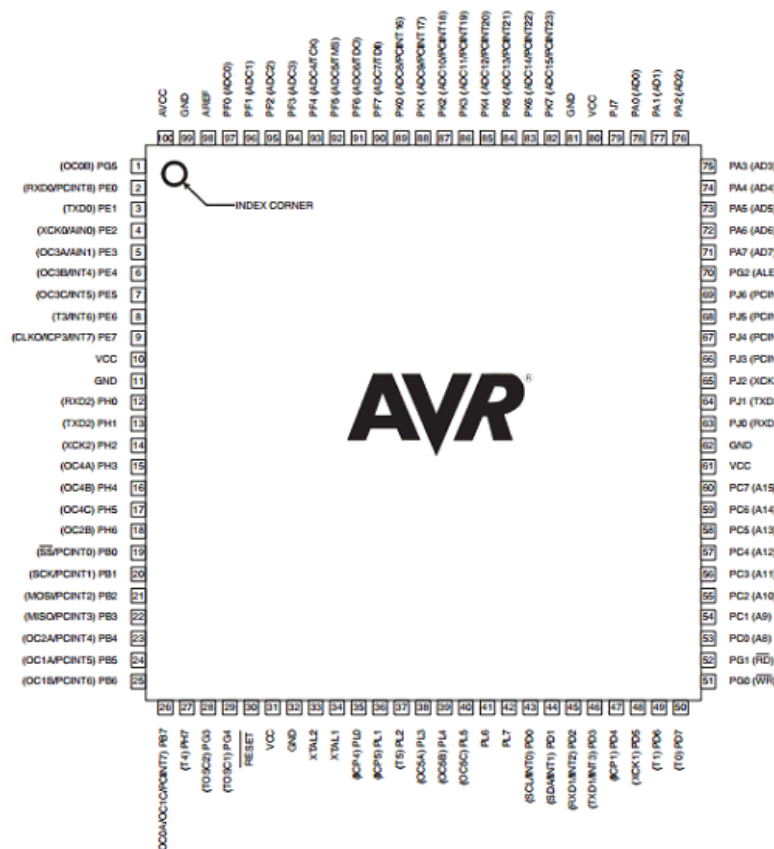


Fig.5 : TQFP-pinout ATmega640/1280/2560

Synthèses caractéristiques :

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)

Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 Ma
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont 8 KB sont utilisés par le bootloader
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

Tableau 2 : fiche technique de l'Arduino Méga

Mémoire :

L'ATmega 2560 à 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader). L'ATmega 2560 a également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

Pour info : Le bootloader est un programme préprogrammé une fois pour toute dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

Entrées et sorties numériques :

Chacune des 54 broches numériques de la carte Méga peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode ()`, `digitalWrite ()` et `digitalRead ()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite (broche, HIGH)`.

Broches analogiques :

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (c.à.d. sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead ()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference ()` du langage Arduino.

Note : les broches analogiques peuvent être utilisées en tant que broches numériques.

Autres broches :

La puce ESP8266 nécessite 3.3V tension d'alimentation. Il ne doit pas être alimenté avec 5 volts comme les autres cartes Arduino.

NodeMCU ESP-12E carte de Dev peut être connecté à 5V en utilisant le connecteur micro USB ou une broche Vin disponible à bord.

Les broches d' E / S de ESP8266 communiquer ou entrée / sortie max 3.3V seulement. Dire que les broches ne sont pas 5V entrées tolérantes.

Si vous avez à l'interface avec 5V broches d' E / S, vous devez utiliser le système de conversion de niveau (soit construit vous - même en utilisant la tension de résistance diviseur.

3. Servomoteur :



Fig.7 : Servomoteur 9g

Description :

Servomoteur miniature économique. Livré avec palonniers, visserie et connecteur JR.

- Alimentation: 4,8 à 6 Vcc
- Course: 2 x 60°
- Couple: 1,6 kg.cm à 4,8 Vcc
- Vitesse: 0,12 s/60°
- Dimensions: 24 x 13 x 29 mm

Présentation :

Le moteur produit un déplacement angulaire qui s'étend de -45° à +45°. Le servomoteur est utilisé en modélisme afin de produire les mouvements nécessaires aux déplacements des oranges de direction. Il est commandé par l'intermédiaire d'un récepteur radio. Cette particularité justifie les principes de commande dont nous allons parler ci-dessous.

Nous allons nous intéresser au principe de commande puis au principe de production de déplacement angulaire.

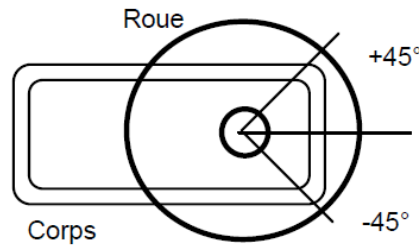


Fig.8 : déplacement angulaire allant de -45° à $+45^\circ$

Le corps contient un moteur à courant continu, un réducteur et une électronique de commande.

La roue possède une série de trous, elle permet la transmission du mouvement par une tringlerie.

Principe de commande :

Il s'agit de commander un déplacement angulaire allant de -45° à $+45^\circ$.

Le moteur doit donc être capable de tourner dans les deux sens de rotation et de suivre les consignes de position avec la contrainte supplémentaire d'avoir un déplacement proportionnel à la commande. La capacité de suivre une consigne est obtenue par des techniques regroupées sous le terme d'asservissement.

Un des moyens possibles est d'utiliser une tension variant de $-9V$ à $+9V$ ce qui permet de déterminer le sens et l'angle de rotation. Cette technique est possible avec une commande filaire mais trop compliquée pour une transmission des informations par radio.

L'architecture interne :

Un servomoteur contient un moteur à courant continu, un réducteur à roues dentées à axes parallèles et une électronique de commande.

L'alimentation et la commande se font par un câble de trois fils, un commun, un fil d'alimentation et un fil de commande. Les couleurs sont conventionnelles pour un constructeur.

Inversion du sens de rotation du moteur :

La tension d'alimentation étant unipolaire, il faut une utiliser une disposition classique modélisée ci-contre.

En fermant H1 et H3, on obtient un sens de rotation, la fermeture de H2 et de H4 donne l'autre sens. Les interrupteurs sont réalisés par des transistors.

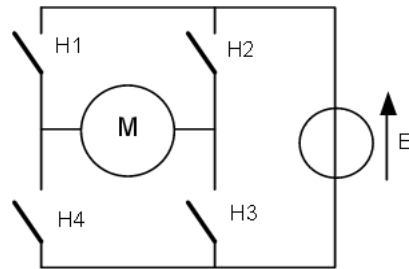


Fig.9 : Inversion du sens de rotation du moteur

4. Capteur de flamme :

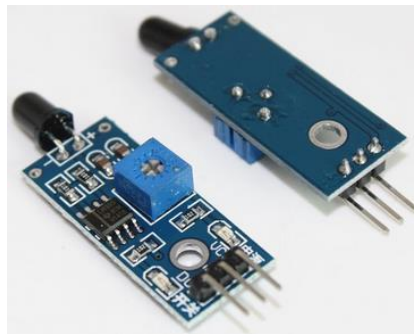


Fig.10 : Capteur de flamme

Présentation du module :

- Nom du produit : Module Capteur IR Infrarouge Flamme Lumière 760nm-1100nm pour Arduino.
- Catégorie : Module de détection

Caractéristiques:

Module capteur de détection de flamme Capteur le plus sensible pour des longueurs d'onde infrarouge de la flamme entre 760 nm et 1100 nm. Il a deux sorties:

AO: sortie analogique, signaux de tension de sortie sur la résistance thermique en temps réel,

DO: lorsque la température atteint à un certain seuil, signaux de seuil de sortie haute et basse est réglable par potentiomètre.

- Capteur de détection de 60 degrés Convient pour projet Arduino DIY
- Tension: DC 3 ~ 5.5V

- Matériel: PCB
- Couleur: bleu + rouge + gris argent
- Dimension du produit: 3,5 x 1,5 x 1,2 cm
- Dimension de l'emballage: 80 x 41 x 15mm
- Poids: 5

Principe De La Détection De Flamme :

Le détecteur de flamme détecte toute élévation de température ou présence de produits issus d'une combustion.

Les flammes produisent des rayonnements caractérisés par une fréquence de scintillement plus ou moins intense dans des bandes spectrales spécifiques. Le principe du détecteur de flamme est de répondre aux rayonnements électromagnétiques émis par une flamme, en les distinguant des rayonnements interférents présents dans l'environnement d'utilisation. Les détecteurs de flamme optiques sont constitués de capteurs UV et/ou IR pour détecter ces rayonnements.

Il existe trois catégories d'appareils pour détecter une flamme :

- Les détecteurs dotés de capteurs Infra-Rouge (IR)
- Les détecteurs composés de capteurs Ultra-Violet (UV)
- Et les détecteurs combinant IR et UV (en général, ils sont constitués de deux capteurs IR et d'un capteur UV)

Les détecteurs IR se divisent eux-mêmes en deux familles, à fréquence unique ou multi-spectre. Le plus souvent les détecteurs multi-IR identifient le spectre du dioxyde de carbone pour les feux carbonés et/ou celui de l'eau pour les feux non carbonés.

La vidéosurveillance associée à un traitement d'image peut également servir à la détection de flamme. Il s'agit de l'imagerie thermique dans le domaine de l'IR ou du visible.

Détection Mono-IR :

Les détecteurs de flamme mono-IR sont sensibles à une bande de rayonnement située autour de 4,4 μm , caractéristique du spectre d'émission des feux d'hydrocarbures. Le rayonnement dans cette bande est issu de la relaxation du CO₂ produit pendant la combustion des feux carbonés.

Les détecteurs mono-IR mettent en œuvre un capteur pyroélectrique dont le principe est basé sur l'effet thermique des rayonnements. Un courant est généré par polarisation du cristal pyroélectrique exposé à une variation de température. Les scintillements de la flamme sont perçus par le cristal qui génère un signal traité par un filtre passe-bande basse fréquence.

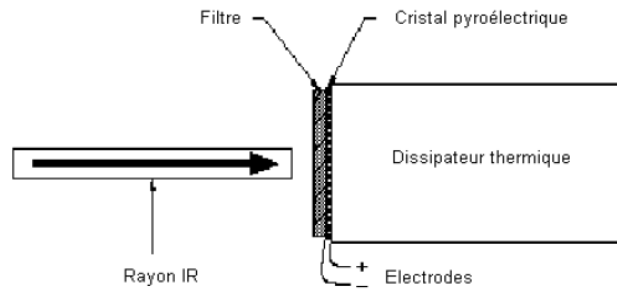


Fig.11 : Schéma de principe d'un capteur pyroélectrique (technologie IR)

Détection Multi-IR :

Les détecteurs multi-IR sont constitués de deux, trois ou quatre capteurs (IR2, IR3, IR4), chacun sensible à une fréquence de rayonnement spécifique dans le domaine infrarouge.

Le principe de fonctionnement des détecteurs multi-IR repose sur :

- Une analyse spectrale qui identifie la signature infrarouge des feux liée à la relaxation des molécules des substances émises : signature caractéristique du CO₂, dans la bande 4.2-4.7 μm , et/ou signature caractéristique de H₂O, dans la bande 2.7-3.0 μm . Des bandes spectrales complémentaires (au-dessus ou en-dessous de ces bandes) sont également analysées pour distinguer le bruit de fond.
- Une analyse des scintillements dans les bandes spectrales caractéristiques

L'alarme feu n'est déclenchée que si tous les seuils des paramètres de l'analyse spectrale et de l'analyse des scintillements sont atteints.

Détection UV :

La détection UV se fait à l'aide d'un photomultiplicateur. Les photons arrivent sur une cathode et provoquent l'émission d'électrons dans un photo-tube contenant un gaz inerte (cf. figure 2) dans un champ électrique. Les électrons, propulsés vers l'anode, ionisent les molécules de gaz créant ainsi une réaction en chaîne. Le capteur génère alors une série d'impulsions de tension qui sont converties électroniquement en un signal d'alarme de sortie. La plage de sensibilité du détecteur au rayonnement UV est généralement très étroite, de 185 à 265 nm, afin de garantir une bonne immunité contre le rayonnement solaire.

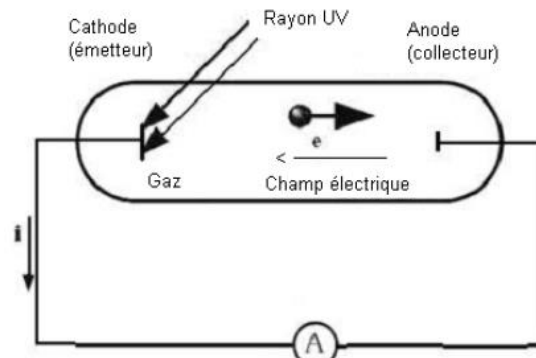


Fig.12 : Schéma de principe d'un capteur UV

Détection UV/IR :

Les détecteurs UV/IR sont constitués d'un capteur UV associé à un ou deux capteurs IR. Les circuits électroniques traitent les signaux issus des deux types de capteurs afin de confirmer le signal de feu de manière plus sûre.

Imagerie Thermique (Ir Ou Visible) :

Actuellement, c'est l'imagerie visible qui est principalement utilisée, elle s'appuie sur du traitement d'image issue de matrices CCD (Charge Coupled Device – capteur à transfert de charges) «classique».

L'imagerie IR, dans les bandes spectrales adéquates, reste encore peu utilisée notamment pour des raisons économiques.

5. Capteur humidité & température DHT11 :

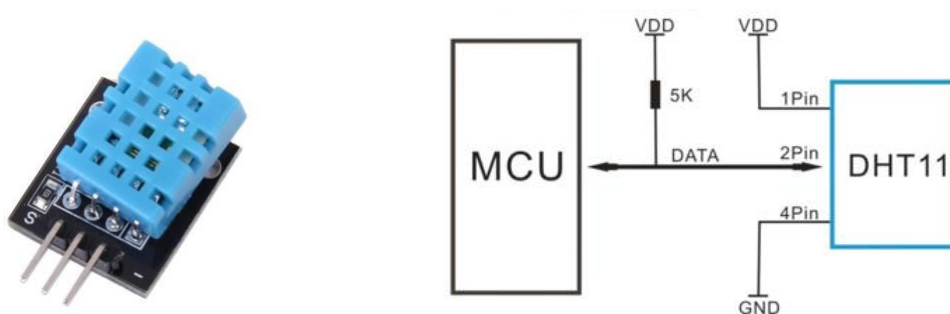


Fig.13 : Capteur d'humidité DHT11/ application typique

Ne pas oublier le condensateur de 100nF entre VCC et GND (il n'est pas sur le schéma mais il est nécessaire).

Caractéristiques :

- Alimentation +5V (3.5 - 5.5V)
- température : de 0 à 50°C, précision : +/- 2°C
- Humidité : de 20 à 96% RH, précision +/- 5% RH

Le protocole de communication :

Le DHT11 a la particularité d'utiliser une seule broche pour communiquer avec son maître. La communication se fait donc au moyen d'un protocole propriétaire demandant des timings bien précis.

Explication des bases pour communiquer avec un DHT11

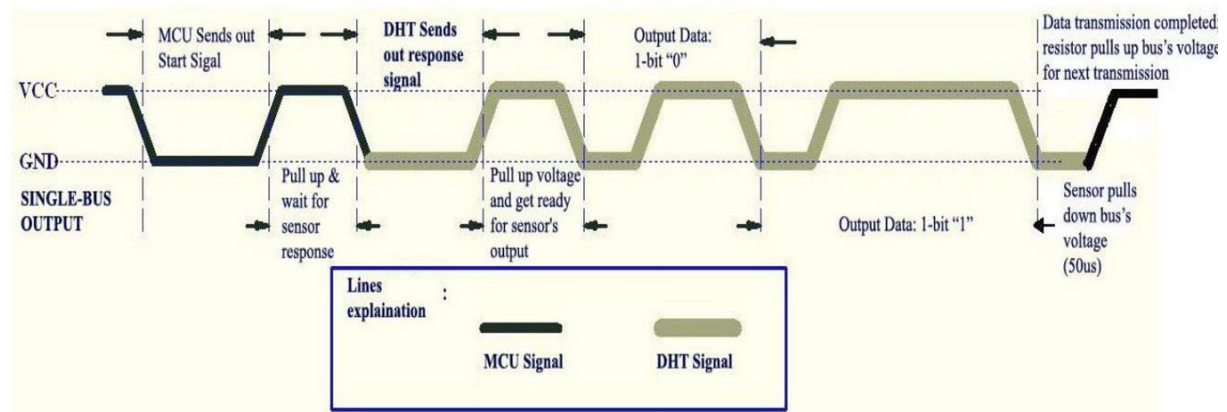


Fig.14 : les bases pour communiquer avec un DHT11

Une trame « type » se compose de trois grandes parties :

- Le maître réveille le capteur au moyen d'un signal de « Start ».
- Le capteur sort de son sommeil et envoie une réponse pour dire qu'il est vivant.
- Le capteur envoie 40 bits de données (5 octets) contenant la température, le taux d'humidité et une checksum.

Ce qu'il faut noter c'est que le capteur a une sortie dite à « collecteur ouvert », cela signifie que le capteur ne sort pas des niveaux logiques 0v/5v mais utilise un transistor en sortie. Si le capteur veut envoyer un « 0 » il force le signal à 0v en rendant son transistor (entre DATA et GND) passant.

Si le capteur veut envoyer un « 1 » il laisse le signal tel-quiel en rendant son transistor bloquant.

D'où l'important de la résistance de tirage que l'on peut voir sur le schéma, pas de résistance = pas de signal

Réveil du capteur :

Le DHT11 est une grosse féniasse, il passe son temps à dormir. Si vous voulez qu'il vous envoie ses données il faut le secouer en lui envoyant un signal de Start.

Le signal de Start consiste en un état bas (0v) de 18 millisecondes et d'un état haut (5v) de 40 microsecondes. Jusque-là rien de bien compliqué !

Ensuite le DHT11 va répondre pour dire qu'il est bien réveillé. Pour ce faire il va placer le signal à l'état bas pendant 80 microsecondes puis à l'état haut pendant 80 autres microsecondes. A ce moment précis le capteur est prêt, les mesures de la température et de l'humidité sont faites et le capteur va pouvoir commencer à envoyer les données.

6. Capteur de gaz MQ6 :

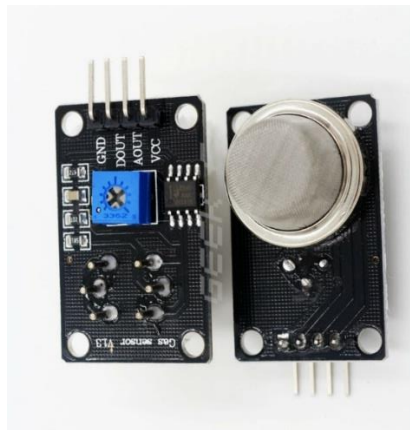


Fig.15 : Capteur de gaz MQ6

Description produit :

- Alimentation électrique : 5 V
- Capteur de gaz (GPL) simple à utiliser
- Peut être utilisé dans les équipements de détection de fuites de gaz dans les applications grand-public et industrielles
- Type d'interface : analogique
- Haute sensibilité au GPL, isobutane, propane
- Faible sensibilité à l'alcool, la fumée
- Réponse rapide
- Stable et durable
- Circuit de pilotage simple

Le MQ-6 Capteur de Gaz GPL, Isobutane et de Propane est un semi-conducteur capteur de gaz qui détecte la présence du gaz GPL, Isobutane et de Propane à des concentrations de 300 ppm à 10000 ppm, une gamme appropriée de détection des fuites de gaz. La simple interface de tension analogique du capteur ne nécessite qu'une seule broche d'entrée analogique de votre microcontrôleur.

Le capteur de gaz GPL, Isobutane et de Propane MQ-6 détecte la concentration de gaz dans l'air et sort le résultat comme une tension analogique. La concentration de détection gamme

de 300 ppm à 10000 ppm est appropriée pour la détection des fuites. Le capteur peut fonctionner à des températures allant de -10 à 50 ° C et consomme moins de 150 mA à 5 V.

Connexions

Raccordement de 5 volts à travers le chauffage (H) broches conserve le capteur assez chaud pour fonctionner correctement. Raccordement de 5 volts, soit au broches A ou B provoque le capteur d'émettre une tension analogique sur les autres broches. Une charge ohmique entre les broches de sortie et la terre, définit la sensibilité du détecteur. La charge résistive doit être calibrée pour votre application particulière en utilisant les équations de la fiche technique, mais une bonne valeur de départ pour la résistance est de 20 kQ.

7. Capteur de mouvement PIR :



Fig.16 : Capteur de mouvement

Description produit :

Les applications d'un senseur PIR sont multiples:

- Détection de mouvement et activation d'une œuvre interactive.
- Détection de passage/intrusion.
- Commande d'une chaudière automatisée.
- Détecteur pour commande d'installation domotique.

Le capteur de mouvement PIR (Passive Infrared Sensor) est un senseur électronique qui mesure la lumière infrarouge (IR) rayonnant à partir d'objets dans son champ de vision. Ils sont très souvent utilisés dans les systèmes d'alarmes ou de détection de présence pour leur faible coût et leur efficacité.

Caractéristique :

- Dimensions: 32 x 24 x 27H mm
- Voltage: 5-12VDC
- Output: 3,3V TTL
- Detection Distance: 3-7mt (approx, adjustable)
- Delay Time: 5-200s (adjustable)

- Trigger: L: non repeatable trigger - H: repeatable trigger

Principe du capteur :

Le fonctionnement d'un capteur PIR n'est pas compliqué à comprendre. Il possède 3 pins : l'alimentation, la masse et le pin data qui va nous intéresser. Par défaut, celui-ci est à l'état bas. Quand le capteur détectera un mouvement cette sortie passera à l'état haut pendant une durée comprise entre 5s et 2min réglable grâce à un des deux petits potentiomètres situés à l'arrière du capteur, l'autre permettant de régler la sensibilité de 3 à 7m.

Pour les détails techniques, ce capteur est basé sur un circuit-intégré BISS001, en regardant le datasheet on s'aperçoit que c'est un CI spécialisé pour cette application qu'il opère entre 3 et 5V et qu'il consomme très peu de courant, il est donc idéal pour une utilisation à piles.

8. Ecran LCD avec un I2C :



Fig.17 : Ecran LCD avec un I2c en arrière

Ce 16 caractères par 2 lignes d'affichage a un contraste texte blanc très clair et très haut sur un fond bleu / rétro-éclairage. Il comprend également une série I2C / IIC carte adaptateur pré-soudé à l'arrière de l'écran LCD. Cela signifie qu'il peut être contrôlé avec seulement 2 I2C broches de données série (SDA et SCL) et nécessite donc beaucoup moins broches numériques IO lorsqu'il est commandé à partir d'un microcontrôleur. Au total, le module ne nécessite 4 fils dont la puissance 5V et GND. Le réglage du contraste est également fourni par la carte fille par l'intermédiaire d'un potentiomètre. Si vous prévoyez d'utiliser avec une carte Arduino.

Note : Ces modules sont actuellement fournis avec une adresse I2C par défaut soit 0x27 ou 0x3F. Pour déterminer quelle version vous avez vérifié l'adaptateur I2C tableau noir sur la face inférieure du module. S'il y a 3 jeux de patins étiquetés A0, A1, A2 et puis l'adresse par défaut sera 0x3F. S'il n'y a pas pads l'adresse par défaut sera 0x27.

Le module est équipé d'un potentiomètre de réglage de contraste sur la face inférieure de l'écran. Cela peut nécessiter de réglage pour l'écran pour afficher le texte correctement. Si la pression est appliquée à la carte fille I2C il est possible de se plier et de venir en contact avec le module LCD. S'il vous plaît assurer lorsque l'écran LCD est installé dans votre application qu'aucun objet externe n'applique une pression sur le dos du module.

Les pins de LCD (I2C) sont : GND, VCC (+5V), SDA et SCL

9. Buzzer :

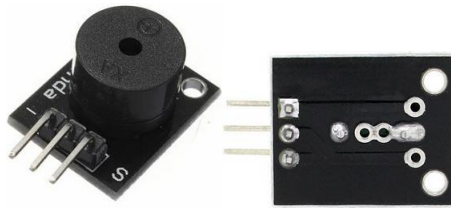


Fig.18 : Buzzer passive

Description :

Type : Buzzer passive

Tension de travail: 3.5-5.5v

Courant de travail: < 25mA

Dimension PCB: 18.5mm x 15mm (L x P)

Fonction de Buzzer : buzz

Fonctionnalité :

Aucune source d'oscillation, besoin onde carrée (fréquence 2k-5k) à conduire.

Le contrôle de la fréquence du son, vous pouvez faire une " Do Re Mi Fa Sol La SI " effet. Buzzer est une structure intégrée de transducteurs électroniques, alimentation en courant continu, largement utilisé dans les ordinateurs, les imprimantes, les photocopieurs, les alarmes, jouets électroniques, matériel électronique automobile et d'autres produits électroniques pour les appareils sonores. Buzzer passif module d'alarme utilisé pour l'Arduino

Le Buzzer interne active avec la source de choc, de sorte que sera appelé à une tension. Les sources internes passives sans chocs, donc si un signal continu ne peut pas faire tweet. Doit 2K ~ 5K avoir place pour le conduire. Buzzer souvent coûteux passif, car il circuit oscillateur multiple. Avantages Buzzer passifs sont les suivants:

1. Bon marché,
2. contrôle de la fréquence sonore, vous pouvez faire un "plus que l'efficacité d'un cheveu mètre Suola Xi 'Fruit.
3. Dans certains cas particuliers, vous pouvez réutiliser un contrôle et un port LED Buzzer actif

Avantages sont les suivants: contrôle de processus, pratique.

I. Logiciels :

1. L'Arduino :

L'open-source Arduino Software (IDE), il est facile d'écrire du code et de le transférer à la carte. Il fonctionne sur Windows, Mac OS X et Linux. L'environnement est écrit en Java et basé sur le traitement et d'autres logiciels open-source.

Ce logiciel peut être utilisé avec toute carte Arduino.

Présentation de logiciel Arduino :

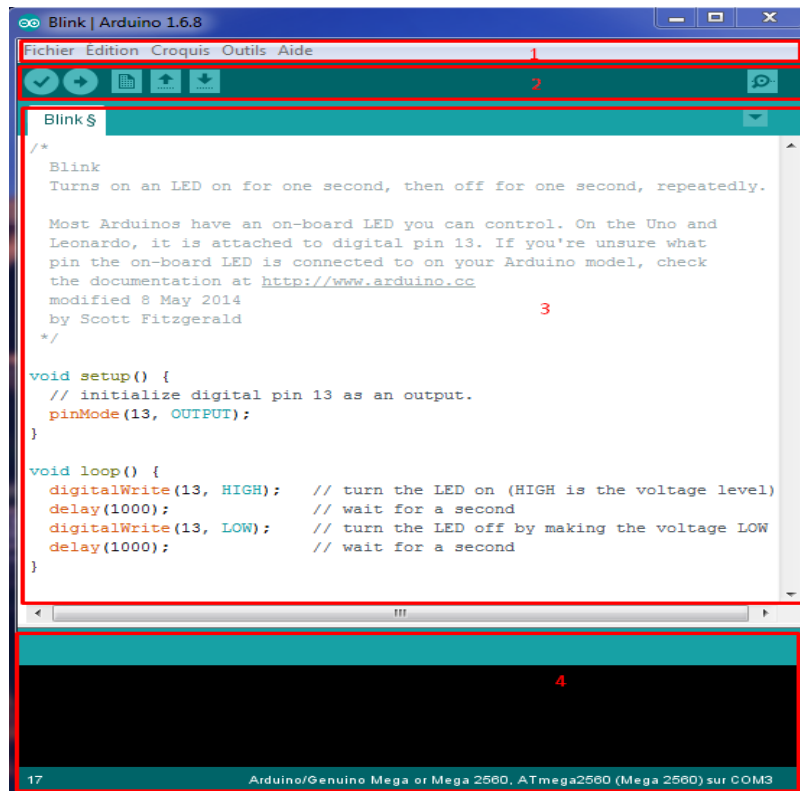


Fig.19 : Présentation de l'interface initiale du logiciel

Correspondance :

- Le cadre numéro 1 : ce sont les options de configuration du logiciel
- Le cadre numéro 2 : il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes
- Le cadre numéro 3 : ce bloc va contenir le programme que nous allons créer

Le cadre numéro 4 : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le **débogueur**.

CHAPITRE 3 : DESCRIPTION DU PROJET

I. Introduction :

L'ordinateur est relié au microcontrôleur par un port USB qui sert à transmettre le code ainsi qu'à alimenter en électricité la carte Arduino. Les données extérieures des capteurs sont envoyées à la carte Arduino, qui envoi à son tour les données reçus à la carte ESP8266 12E. De plus, les données relevées par les capteurs peuvent être visibles sur une page WEB.

Pour pouvoir faire tout cela, il faut créer un programme permettant de contrôler les appareils électriques en fonction des données reçues. Pour cela plusieurs langages de programmation sont utilisables tels que : Python, Java, C++, etc. Dans le cadre du projet nous avons utilisé l'environnement de développement Arduino. En effet, nous avons choisis ce langage suite à l'achat des différents composants dont le principal est une carte Arduino nous permettant d'agir au niveau des appareils électriques.

II. Fabrication de la maison :

1. Structure de la maison :

La première étape consiste à créer une maison. Pour cela, nous avons dessiné la structure principale, c'est-à-dire les murs extérieurs et le sol à l'aide d'une découpeuse laser. La maquette est de taille de 31cm sur 28cm constitue d'une chambre, une cuisine, un garage, salle de bains, un salon, et une salle de matérielles.

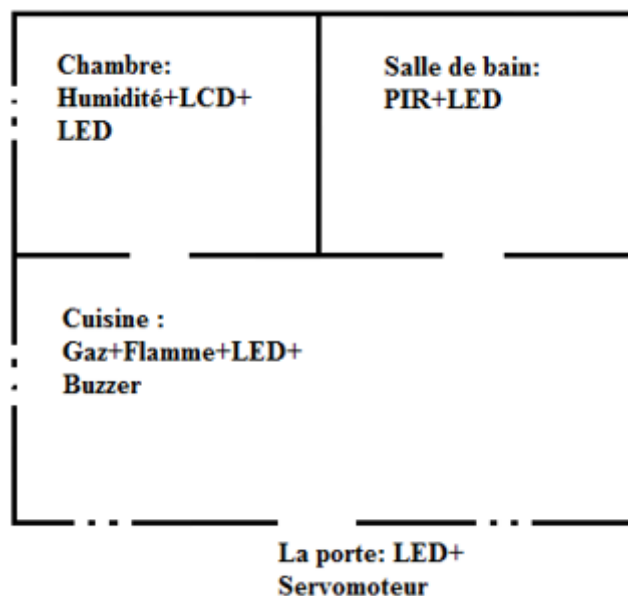


Fig.20 : illustration de la maquette utilisée dans le projet

2. Périphériques utilisées et leurs positions :

Cette maquette, permettrait de présenter certaines fonctionnalités de la domotique à travers 4 Capteurs (Gaz, Humidité et Température, Mouvement, Ultrason et de Flamme.) 2 Servomoteur et autres périphériques. Ces scénarios seront automatisés via la carte « arduino » Exécutant des programmes informatiques.

LOCALISATION	PERIPHERIQUES
La chambre	-LED - Capteur d'humidité -LCD 1602 I2C
Salle de bains	-Capteur de mouvement PIR -LED
La cuisine	-Capteur de gaz MQ6 -Capteur de flamme -LED -Buzzer
La porte	-servomoteur -LED

Tableau 3 : Localisation des périphériques utilisés dans la maquette

III. Automatisation de la maison :

1. Capteur de mouvement PIR :

La gestion de la consommation d'énergie est une tendance actuelle, c'est pourquoi nous avons respecté cette tendance à une échelle réduite. Notre dispositif permet d'allumer automatiquement la lumière en cas de présence dans la salle de bain [mais permet aussi d'éteindre celle-ci en l'absence de mouvement] pendant (10 min).

Le matériel que nous avons utilisé est le suivant :

- Un capteur de mouvement PIR
- Une LED (lampe)
- Une Carte Arduino Méga

2. Capteur d'humidité :

Ce dispositif permet d'améliorer la gestion d'énergie de la maison. En effet nous avons utilisé un dispositif prenant en charge la détection de la température et de l'humidité dans l'entourage de la maison.

Nous avons utilisé comme matériel :

- Un capteur DHT 11
- Ecran LCD 1602 avec I2C
- Une carte Arduino

3. Capteur de flamme / de gaz :

La sécurité est devenue un élément primordial dans le choix d'une maison. Et l'une des plus grande crainte d'accident reste l'incendie. Ainsi nous avons associé différents composants

afin d'utilisé un détecteur de flamme et de gaz dans la cuisine pour une bonne sécurité. Ces deux détecteurs déclenchent une alarme.

Le matériel utilisé pour la flamme est le suivant :

- Un détecteur de flammes
- Un Buzzer
- Une carte Arduino Méga

Le matériel utilisé pour le gaz est le suivant :

- Un détecteur de gaz
- Un Buzzer
- Une carte Arduino Méga

4. Servomoteur :

Le confort est placé au premier plan pour cela pour éviter les actes répétitifs et pour une bonne sécurité de la maison nous avons pensé à ce dispositif. On va commander à distance la fermeture et l'ouvrir de la porte initiale de la maison.

Le matériel que nous avons utilisé est le suivant :

- servomoteurs 9g
- Une Carte ESP 8266 12E

Le programme et le schéma global sont situés en **Annexe**.

IV. Centralisation des commandes : Interface Web

La domotique peut également permettre à des fonctions d'être contrôlés à distance de partout dans le monde en utilisant tout appareil avec un navigateur.

Pour rendre la maison connectée et commandable à distance nous avons le choix de travailler soit avec une carte wifi pour Arduino ou un module Bluetooth, ce dernier a une zone de commande inférieure à 15 mètres, ce qui nous a obligé d'utiliser le module (WI-FI) **l'ESP8266 12E**, dont on a choisi de le configurer autant que serveur avec une page Web accessible via l'adresse IP.

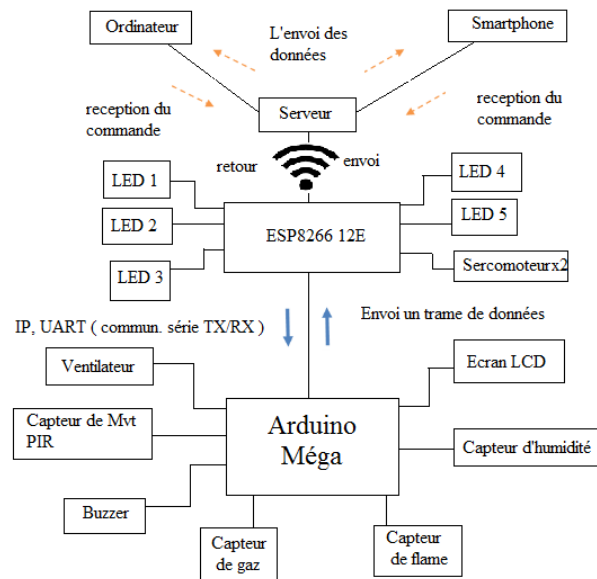
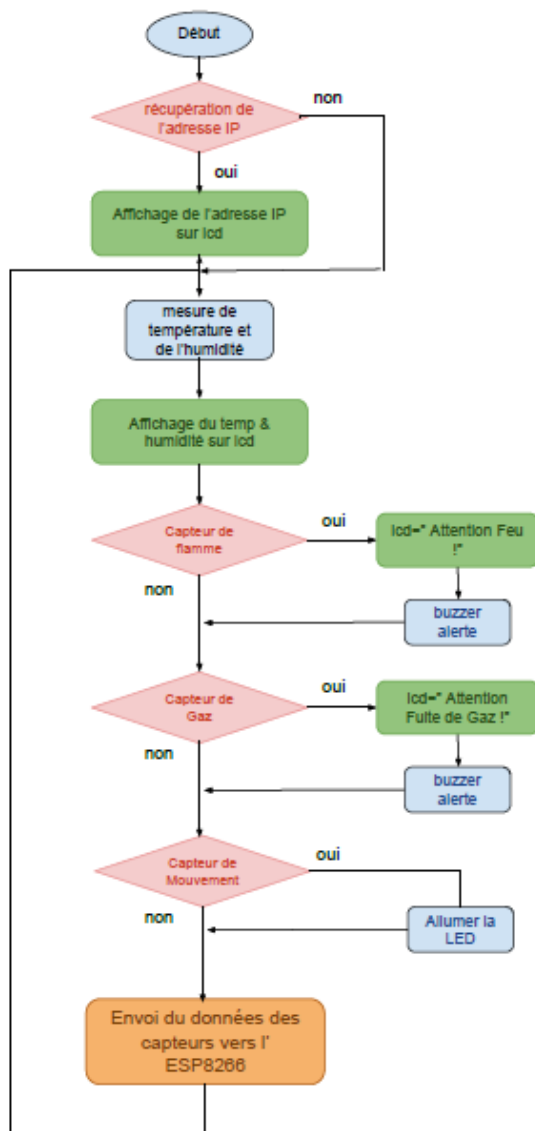


Fig. 21 : L'organigramme représente le chemin des requêtes



Cette figure représente le fonctionnement des capteurs liés à la carte Arduino Méga, et l'envoi des données mesurés par les capteurs.

Fig.22 : l'organigramme de la carte Arduino Méga

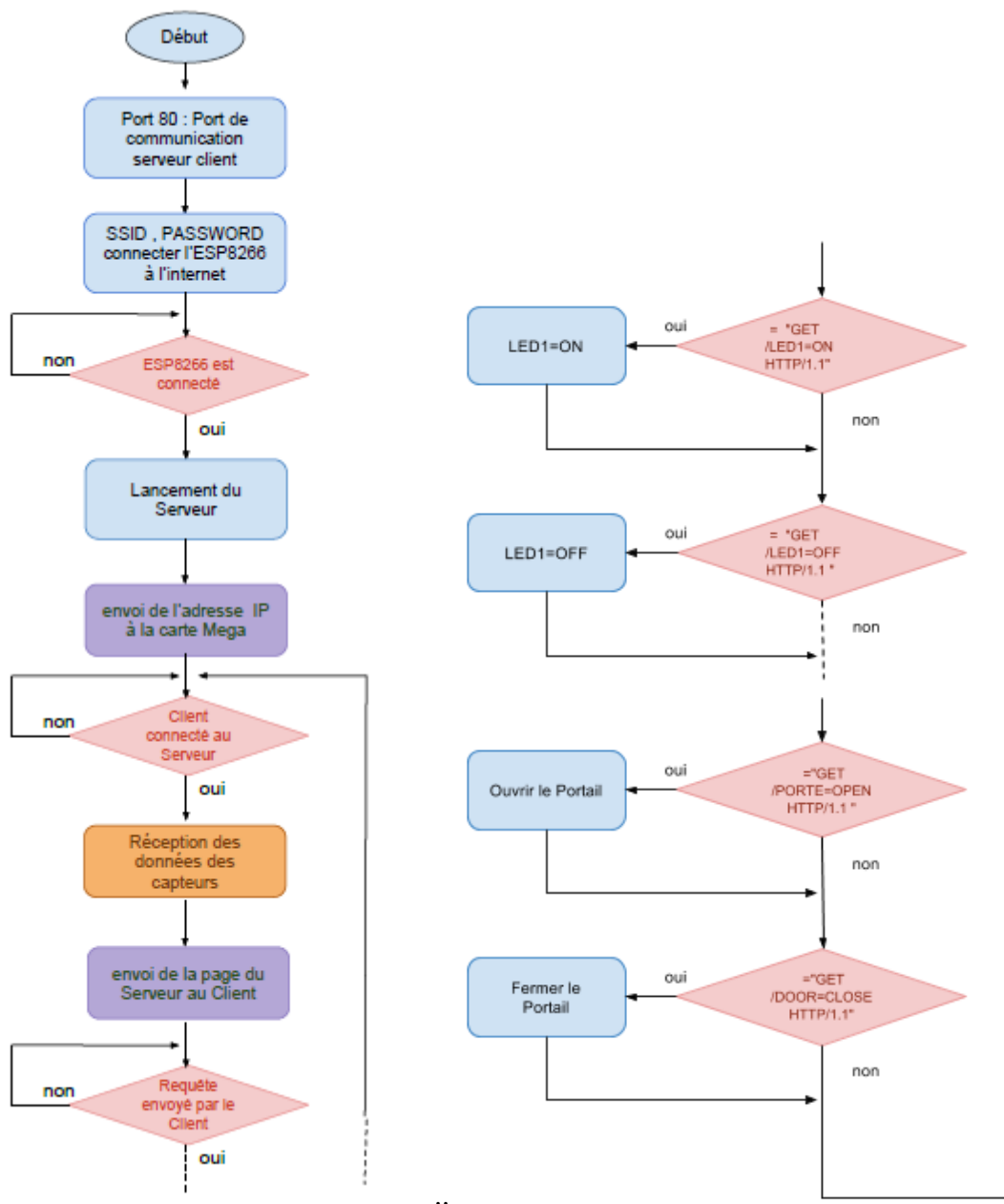
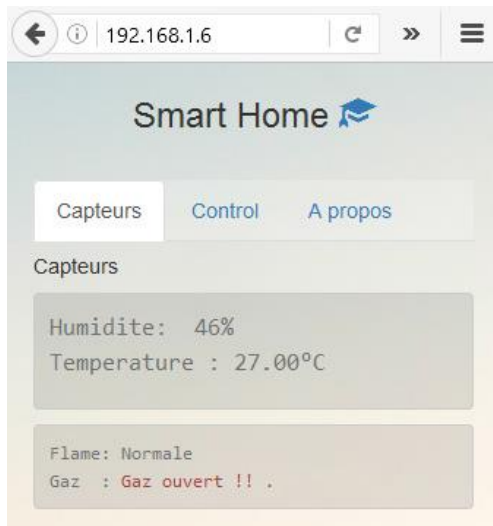


Fig.23 : l'organigramme de la carte ESP8266 12E

La page web va être sous forme d'une interface qui doit être en mesure de permettre à l'utilisateur de visualiser les données des capteurs et aussi de contrôler les lampes et les portes de manière simple et intuitive.

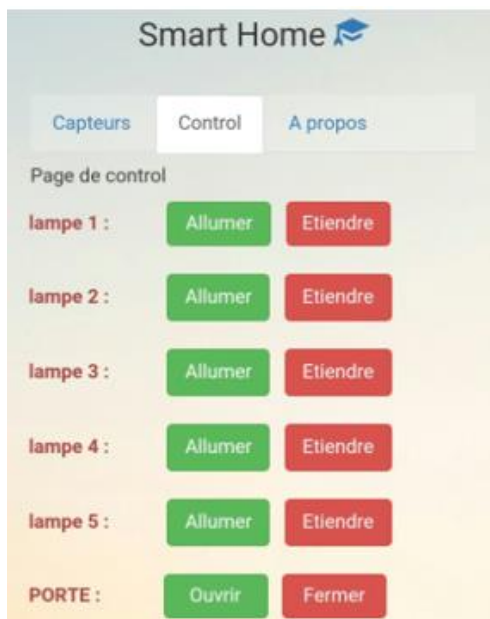
L'interface des capteurs :

Dans cette interface vous trouvez tous les capteurs qu'on a utilisés (Humidité, gaz, flamme, PIR) chacun à ses états.

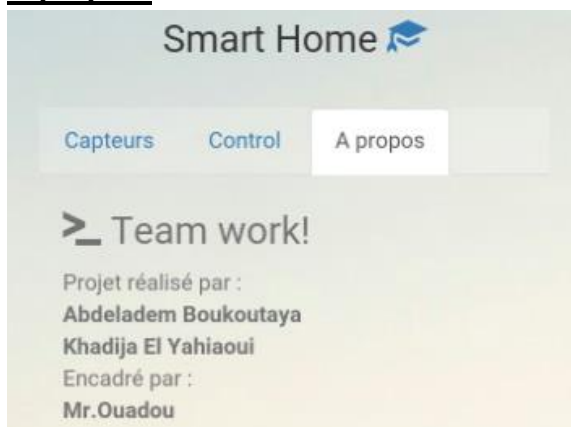


L'interface de commande :

La deuxième interface pour les modules qu'on souhaite contrôler (LEDS, Servomoteur)



A propos :



Conclusion Générale :

Nous avons fabriqué une maison dite « intelligente ». Effectivement, dans le cadre du thème de la domotique, nous avons conçu une maison automatisée. Elle est capable de gérer l'éclairage des chambres et les portes, de prévenir en cas d'incendie ou dans le cas de fuite de gaz aussi surveillé l'état de la température et l'humidité. Cependant, les fonctionnalités ne se restreignent pas à celles-ci et d'autres peuvent être ajoutées grâce à un système de centralisation.

Nous sommes unanimes pour dire que ce projet nous a permis de nous amuser grâce à la manipulation du matériel, tout en acquérant de meilleures connaissances des applications de la domotique, ce qui pourrait nous être fortement utile pour notre vie professionnelle future. Bien sûr tout ce travail s'est déroulé dans les meilleures conditions possible, en effet une bonne cohésion et une bonne entente ont permis l'obtention d'un travail abouti et satisfaisant.

Ce projet nous a fait découvrir un secteur que nous ne connaissions pas vraiment et qui nous a intéressés de plus en plus au fur et à mesure que nous approfondissions nos recherches. En plus de l'expérience humaine, la rencontre avec des professionnels travaillant dans la domotique nous a permis de recueillir des informations techniques et des explications nécessaires à la compréhension du principe de fonctionnement de certaine technologie.

Le seul point « négatif », serait sûrement le manque de temps pour pouvoir encore approfondir ce travail, car ce dernier ne s'arrête pas ici il a encore plusieurs tache q i peut être amélioré. En effet, beaucoup de possibilités s'offrent aux passionnés de domotique, tant sur le matériel disponible que sur les actions à réaliser. Cependant rien ne nous empêche de continuer sur cette voie de notre propre côté ...

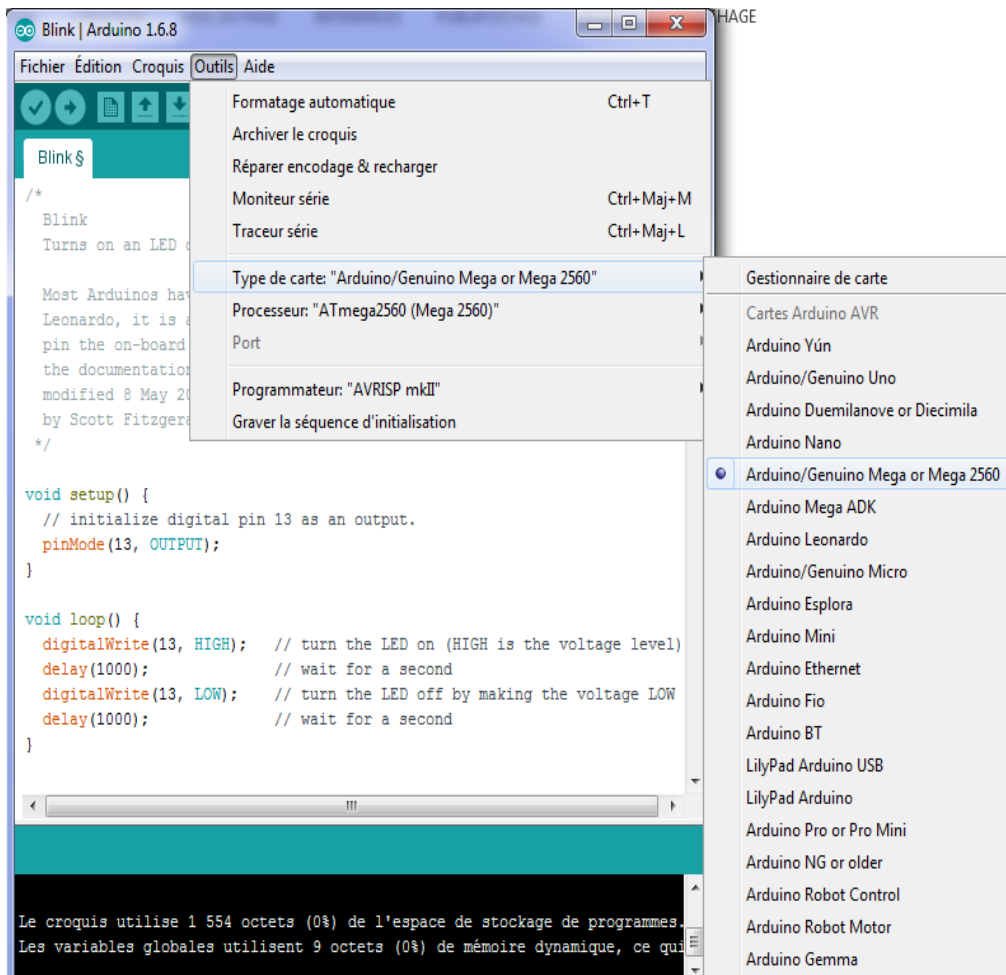
Ce projet a été vivant, entraînant et motivant pour la suite de nos études. Nous pensons avoir entraperçu une partie de notre future vie active.

Annexes et Programmes

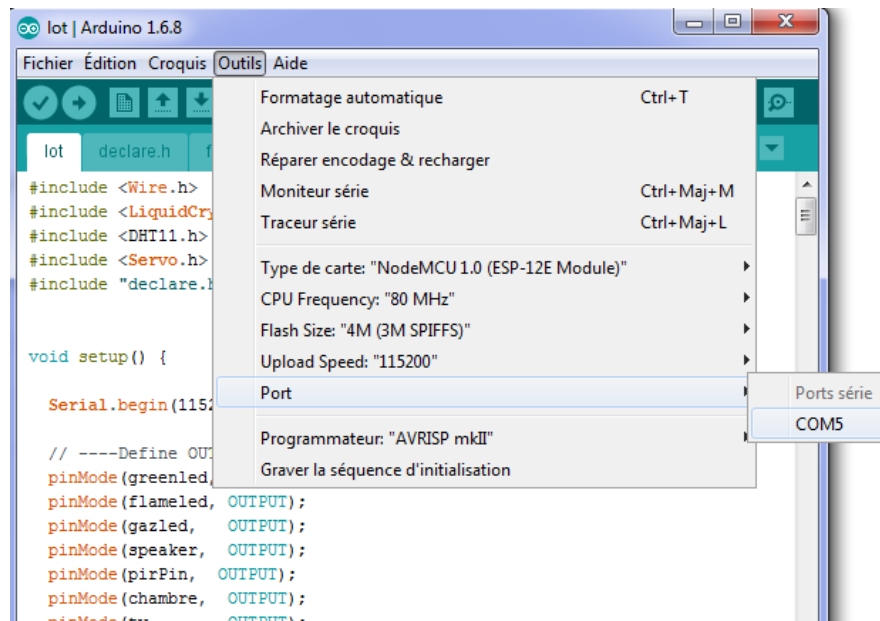
Découvrir Arduino :

Dans un premier temps on va Télécharger le logiciel Arduino gratuitement depuis le site arduino.cc, ensuite on va installer le driver pour la carte Arduino

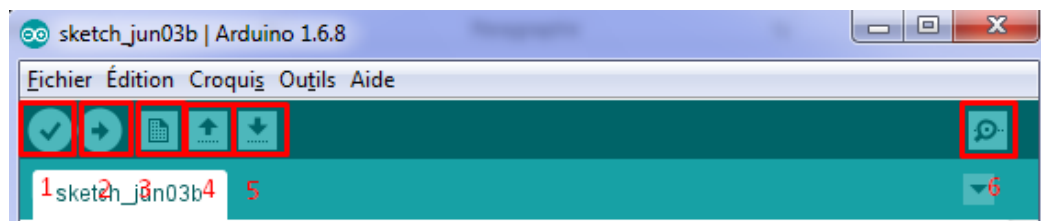
Choisir le type de carte :



Serial Port / COM Port:

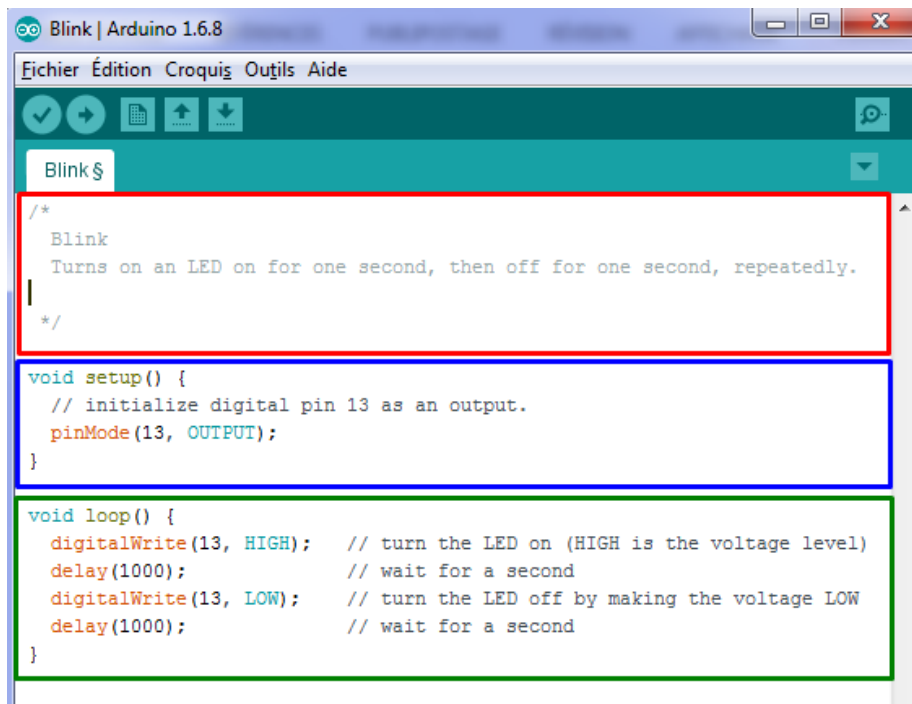


Charger un programme dans la carte :



- Bouton 1 : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.
- Bouton 2 : Charge (téléverser) le programme dans la carte Arduino.
- Bouton 3 : Crée un nouveau fichier.
- Bouton 4: Ouvre un fichier.
- Bouton 5 : Enregistre le fichier.
- Bouton 6 : Ouvre le moniteur série.

Principales parties d'un code:



```
Blink | Arduino 1.6.8
Fichier Édition Croquis Outils Aide

Blink $

/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 */

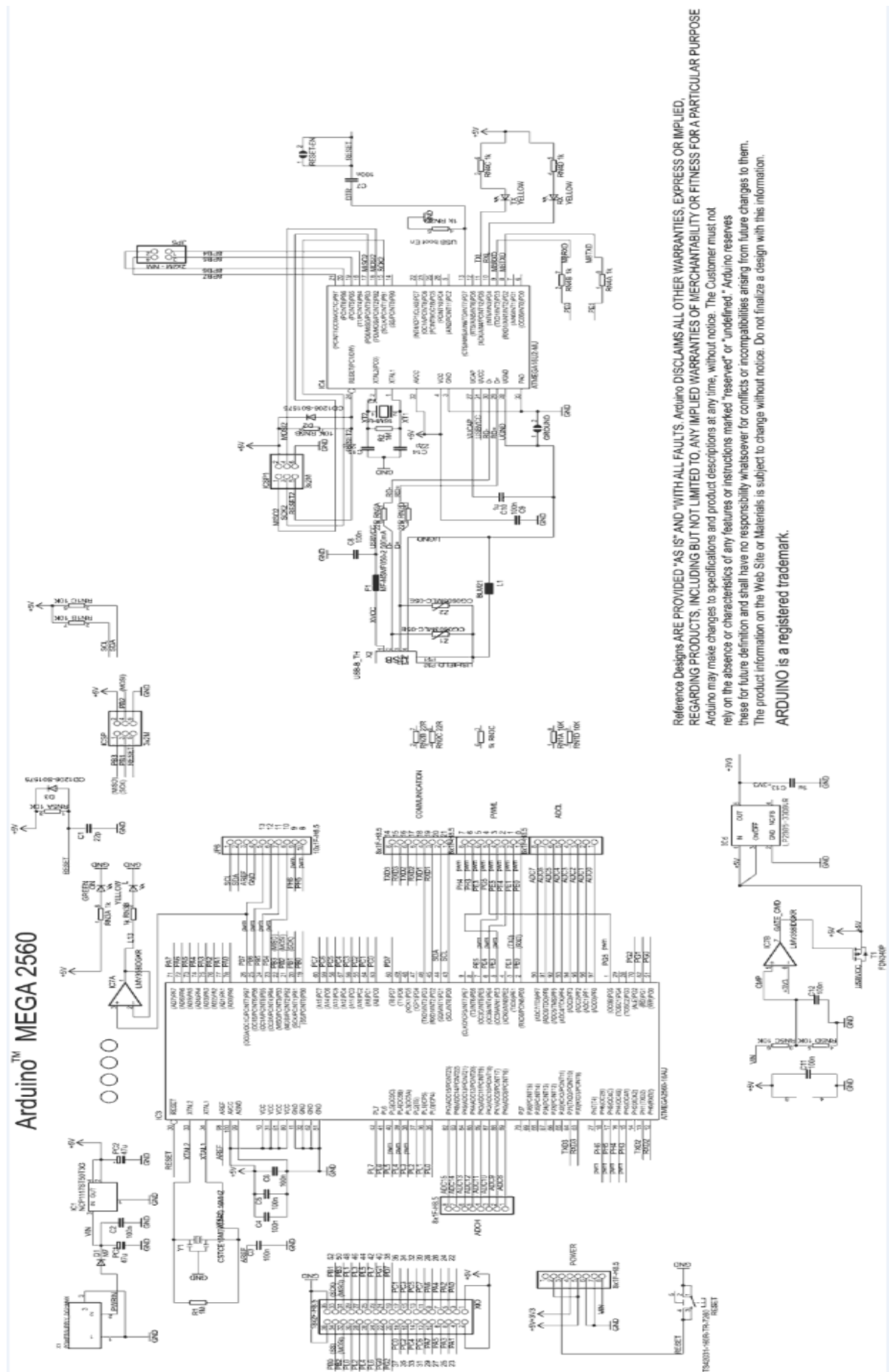
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

**Commentaire /
expliquer le jeu**

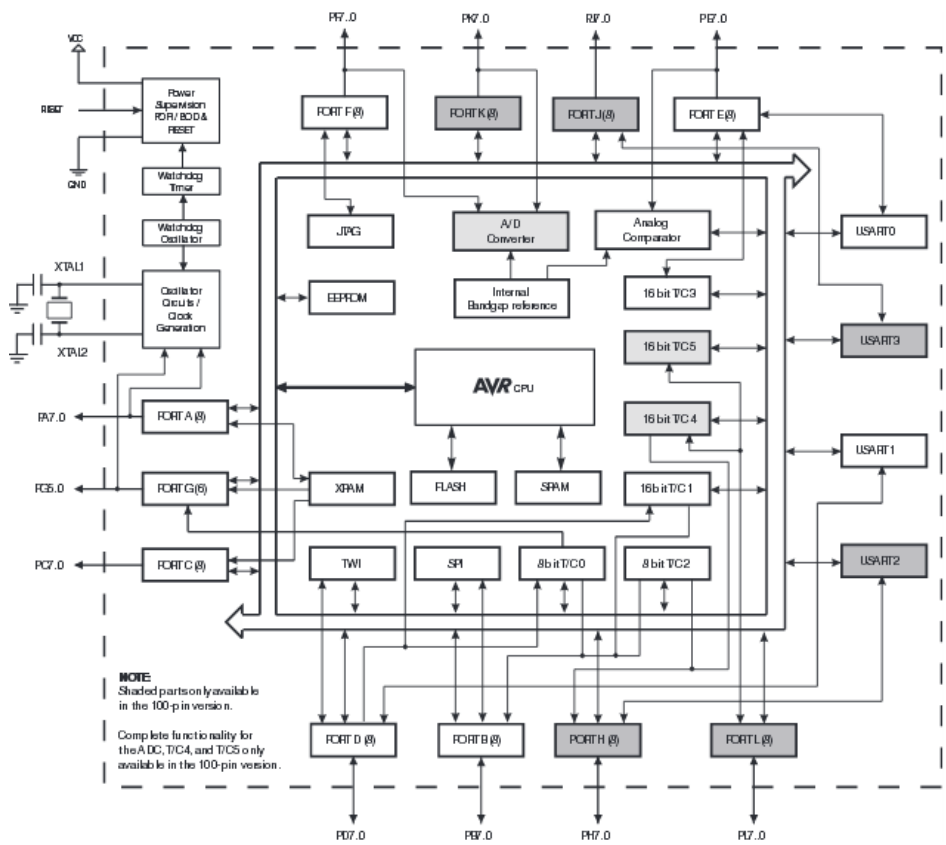
**installation /
étirement**

**boucle /
jouer le jeu**

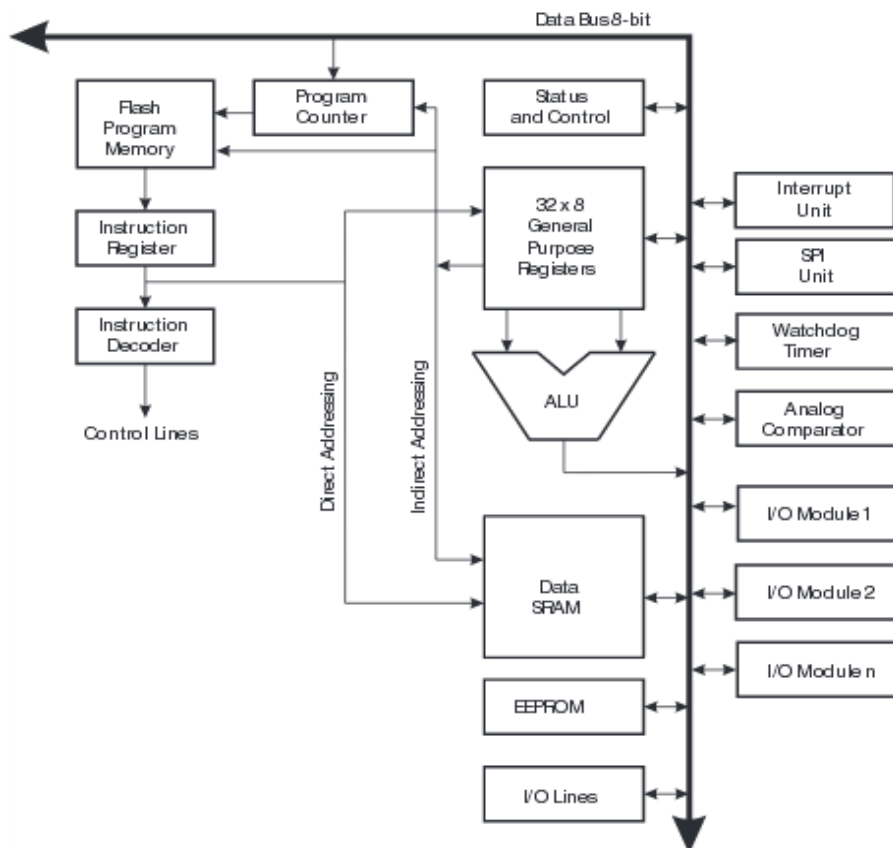


Reference Designs are PROVIDED "AS IS" AND "WITH ALL FAULTS." Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.



Datasheet de la carte ATmega2560



Block diagram of the AVR

Les programmes:

⇒ Le code de l'Arduino Méga :

```
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include "DHT.h"
#include <Servo.h>

#define DHTTYPE DHT11
#define Esp8266 Serial1
#define Debug Serial

/***** declaration des variables *****/
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

/*---- Capteur ----*/
int flamePin    = A0;
int gazPin      = A1;
int dhtPin      = 12;
int pirPin      = 3;
int speaker     = 10;

/*---- Leds -----*/
int flamed      = 23;
int gazed       = 22;
int greenled    = 24;

/*----- Var -----*/
long lecture_echo ;
long cm           ;
int Pirr          ;
int flame         = 0;
int gaz           = 0;
String tram       = "";
float t, h        ;
String recu="" , ip="" , ssid="" ;

DHT dht(dhtPin, DHTTYPE);

void setup() {

  Debug.begin(9600);
  Esp8266.begin(9600);
  dht.begin();

  pinMode(flamePin, INPUT);
  pinMode(gazPin, INPUT);
  pinMode(pirPin, INPUT);
  pinMode(dhtPin, INPUT);

  // ----Define OUTPUT pins

  pinMode(greenled, OUTPUT);
  pinMode(flamed, OUTPUT);
  pinMode(gazed, OUTPUT);
```

```
pinMode(speaker, OUTPUT);
pinMode(pirPin, OUTPUT);

lcd.begin(16,2);
lcd_game(400);
delay(4000);
}

void loop() {
  capteurs();
}

void capteurs(){

  Flame_test();
  Gaz_test();
  dht_test();
  Pir_test();
  Debug.println();
  Esp8266.print(tram);
  Debug.print(tram);
  Debug.println();
  delay(1000);
  tram= "";

}

/*****
*
*      Capteur de Temperature
*
*****/

/
void dht_test(){
float h = dht.readHumidity();
float t = dht.readTemperature();

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temperature:");
  lcd.setCursor(13, 0);
  lcd.print((int)t);
  lcd.print("C");

  lcd.setCursor(0,1);
  lcd.print("Humidite: ");
  lcd.setCursor(13,1);
  lcd.print((int)h);
  lcd.print("%");

  tram+=(String)t+"|";
  tram+=(String)h+"|";

}
```

```

/*****
**
*
*      Detecteur de mouvement
*
*****/

/
void Pir_test(){

    Debug.print("mouvement detecte : ");
    Pirr=digitalRead(pirPin);
    Debug.println(Pirr);
    delay(100);

    tram+=Pirr;
    tram+="|";

}

/*****/
**
*
*      Gaptteur de flame
*
*****/

/
void Flame_test(){
    int fval = analogRead(flamePin);
    int stat= 0;
    if (fval <=80) {
        stat= 1;
        Debug.println(" Warning Fire is On");
        tone(speaker, 750, 200);
        digitalWrite(flameled, HIGH);
        digitalWrite(greenled, LOW);
        lcd.clear();
        lcd.print("Attention !!");
        lcd.setCursor(0,1);
        lcd.print("!!   Feu   !!");
        delay(500);
        lcd.clear();
    }

    else {
        stat= 0;
        noTone(speaker);
        digitalWrite(flameled, LOW);
        digitalWrite(greenled, HIGH);
    }
    Debug.print("fire distance is :");
    Debug.println(fval);
    tram+=stat;
    tram+="|";
}

```

```

/*****
**
*
*           Capteur de gaz
*
*****/

/
void Gaz_test(){
    int gval=analogRead(gazPin);
    int stat= 0;

    if (gval >=250 ){
        stat= 2;
        Debug.println("Gaz opened");
        tone(speaker, 600, 300);
        digitalWrite(gazled, HIGH);
        digitalWrite(greenled, LOW);
        lcd.clear();
        lcd.setCursor(1,0);
        lcd.print("Attention !!");
        lcd.setCursor(0,1);
        lcd.print("!!! -Gaz- !!!");
        delay(500);
        lcd.clear();
    }

    else{
        stat= 0;
        noTone(speaker);
        digitalWrite(gazled, LOW);
        digitalWrite(greenled, HIGH);
    }
    Debug.print("gaz distance is :");
    Debug.println(gval);
    tram+=stat;
    tram+="|";
}

/*****
**
*
*           LCD
*
*****/

/
void lcd_game(int delai){
    // Quick 3 blinks of backlight
    for(int i = 0; i< 3; i++)
    {
        lcd.backlight();
        delay(200);
        lcd.noBacklight();
        delay(200);
    }
    lcd.backlight();

    lcd.setCursor(0,0);

```

```
lcd.print("PFE :");
lcd.setCursor(6,0);
lcd.print(" Domotique");
delay(3000);
lcd.setCursor(0,1);
lcd.print("A");delay(150);
lcd.setCursor(1,1);
lcd.print("c");delay(150);
lcd.setCursor(2,1);
lcd.print("t");delay(150);
lcd.setCursor(3,1);
lcd.print("i");delay(150);
lcd.setCursor(4,1);
lcd.print("v");delay(150);
lcd.setCursor(5,1);
lcd.print("a");delay(150);
lcd.setCursor(6,1);
lcd.print("t");delay(150);
lcd.setCursor(7,1);
lcd.print("i");delay(150);
lcd.setCursor(8,1);
lcd.print("o");delay(150);
lcd.setCursor(9,1);
lcd.print("n");delay(150);
for(int s=10;s<17;s++)
  lcd_print('.',s,1);

for(int s=0;s<11;s++)
  lcd_print('.',s,1);

lcd.setCursor(10,1);
lcd.print("active");
delay(3000);
  lcd.clear();
lcd.setCursor(0,0);
  lcd.print("connection  ");
while (!(Esp8266.available() > 0 )){

  for(int i=1; i<17 ; i++){
    lcd_print('>' ,i,1);
    delay(1);
    int j =i-1;
    lcd_print(' ' ,j,1);
    delay(50);
  }

  for(int i=16; i>=1 ; i--){
    lcd_print('<' ,i,1);
    delay(1);
    int j= i+1;
    lcd_print(' ' ,j,1);
    delay(50);}

}
recu= Esp8266.readString();
Debug.print(recu);

ssid =  getStr(recu,1);
ip = getStr(recu,2);
lcd.clear();
```

```
        lcd.setCursor(0,0);
        lcd.print("ssid: ");
        lcd.setCursor(7,0);
        lcd.print(ssid);
        lcd.setCursor(0,1);
        lcd.print("ip: ");
        lcd.setCursor(3,1);
        lcd.print(ip);
        delay(1000);

    }

    void lcd_print(char a ,int c,int l){
        lcd.setCursor(c,l);
        lcd.print(a);
        delay(100);
    }

    /******* Read recieved data *****/

    String getStr(String data, int index)
    {
        int stringData = 0;
        String dataPart = "";
        for(int i = 0; i<data.length()-1; i++){
            if(data[i]=='|'){
                stringData++;}

            else if(stringData==index){
                dataPart.concat(data[i]);}

            else if(stringData>index){
                return dataPart;
                break;
            }
        } return dataPart;
    }
```


⇒ Le code de l'ESP 8266 12E :

```
##include <ESP8266WiFi.h>
#include <Servo.h>

/***** Déclaration des variables *****/
Servo porte;

String tram="";
const char* ssid = "Eloctro-lab";
const char* password = "12345678";

int led1=16;
int led2=5;
int led3=4;
int led4=0;
int led5=2;

int pos =0;

WiFiServer server(80);

void setup(){

Serial.begin(9600);
delay(10);

pinMode(led1,OUTPUT);
pinMode(led2,OUTPUT);
pinMode(led3,OUTPUT);
pinMode(led4,OUTPUT);
pinMode(led5,OUTPUT);

digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
digitalWrite(led5, LOW);

porte.attach(14);
porte.write(pos);
```

```
Serial.print("|");
Serial.print(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(100);
}

server.begin();

Serial.print("|");
Serial.print(WiFi.localIP());
Serial.print("|");

}

void loop() {

WiFiClient client = server.available();
if (!client) {
return;
}

client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("");

client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<title>Domotique</title>");
client.println("<meta charset=\"utf-8\">");
client.println("<meta http-equiv=\"refresh\" content=\"20\">");
client.println("<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
client.println("<link rel=\"stylesheet\" href=\"http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css\">");

//----- JavaScript -----
client.println("</div></div>");
client.println("<script src=\"https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js\"></script>");
```

```
client.println("<script
src=\"http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js\">
</script>");

client.println("<script
type=\"text/javascript\">$(document).ready(function(){ $('a[data-
toggle=\"tab\"]').on"
"('show.bs.tab',function(e){localStorage.setItem('activeTab',
$(e.target).attr('href'))});});"
"var activeTab = localStorage.getItem('activeTab');if(activeTab){ $('#myTab
a[href=\"' + activeTab + '\"]').tab('show');});});</script>");

//----- CSS -----
client.println("<style type=\"text/css\">.custom{ width: 75px ; padding:
5px;}.bs-example{margin: 20px;alignment:center;}"
"pre{max-width:500px; background-color: rgba(0,0,0,0.1); color: #737373; }"
"body{position: relative;margin-left: auto;margin-right: auto;width:
100%;text-align: left;}</style>");

client.println("</head>");
client.println("<body style=\"background:
url(http://www.wallpaperup.com/uploads/wallpapers/2013/10/28/166644/a526e84
ad1dcafaa6fe06ddda28ba029.jpg) no-repeat center center fixed;\">");
client.println("<div class=\"container\" style=\"text-align:
center;\"><h3>Smart Home <a class=\"glyphicon glyphicon-education\"
href=\"\"#sectionC\"></a></h3></div>");
client.println("<div class=\"bs-example\"><ul class=\"nav nav-tabs\"
style=\"background-color: "
"rgba(255,255,255,0.2)\" id=\"myTab\"><li class=\"active\"
id=\"capteurs\"><a data-toggle=\"tab\" "
"href=\"\"#sectionA\">Capteurs</a></li><li><a data-toggle=\"tab\"
href=\"\"#sectionB\">Control</a></li>"
"<li><a data-toggle=\"tab\" href=\"\"#sectionC\">A propos</a></li></ul>");

// ----- section A -----
client.println("<div class=\"tab-content\"><div id=\"sectionA\"
class=\"tab-pane fade in active\" ><h5>Capteurs</h5>");

if (Serial.available() > 0 ){
    String recu="" , flame="", gaz="", humi="",temp="", us="";

    recu= Serial.readString();
    flame = getStr(recu,0);
    gaz   = getStr(recu,1);
    temp  = getStr(recu,2);
    humi  = getStr(recu,3);

    //***** temperature & humidité *****
    client.print("<pre style=\"background-color: rgba(0,0,0,0.1); color:
#737373;height: 80px; font-size: 120%;\" >Humidite:  ");
```

[Réalisation d'une maison intelligente à base d'Arduino]

[illegible]

[illegible]

[illegible]

```
}
else if ((String)request == ("GET /L4N HTTP/1.1")) {
digitalWrite(led4, HIGH);
}
else if ((String)request == ("GET /L4F HTTP/1.1")) {
digitalWrite(led4, LOW);
}
else if ((String)request == ("GET /L5N HTTP/1.1")) {
digitalWrite(led5, HIGH);
}
else if ((String)request == ("GET /L5F HTTP/1.1")) {
digitalWrite(led5, LOW);
}

//----- Porte -----
else if ((String)request == ("GET /DO HTTP/1.1")) {
  for (pos = 0; pos <= 180; pos += 1) {
    porte.write(pos);
    delay(5);
  }
}
else if ((String)request == ("GET /DC HTTP/1.1")) {
  for (pos = 180; pos >= 0; pos -= 1) {
    porte.write(pos);
    delay(5);
  }
}

delay(20);
Serial.println("Client disconnected");
Serial.println("");
}

String getStr(String data, int index)
{
  int stringData = 0;
  String dataPart = "";
  for(int i = 0; i<data.length()-1; i++){
    if(data[i]=='|'){
      stringData++;
    }

    else if(stringData==index){
      dataPart.concat(data[i]);
    }
  }
}
```

```
        else if(stringData>index){  
            return dataPart;  
            break;  
        }  
    } return dataPart;  
}
```


Les références :

<https://skyduino.wordpress.com/2013/04/27/tuto-capteur-temperature-humidite-dht11/>

<http://www.micro4you.com/files/sensor/DHT11.pdf>

<http://playground.arduino.cc/main/DHT11Lib>

<http://fr.hobbytronics.co.uk/mq6-propane-sensor>

http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielMega2560

<http://wiki.t-o-f.info/Arduino/CapteurPIRDeMouvement>

<http://lecafedugeek.fr/electronique-detectez-les-intrus-avec-le-capteur-de-mouvement-pir/>

<https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/testing-a-pir>

<https://www.sparkfun.com/products/13678>

http://shop.mchobby.be/breakout/647-module-wifi-esp8266-breakout-3232100006478.html?search_query=ESP8266&results=11

<http://www.maison-et-domotique.com/47940-combien-coute-la-domotique/>

<http://eskimon.fr/79-arduino-103-le-logiciel>

<https://translate.google.com/translate?hl=fr&sl=en&u=https://acrobotic.com/acr-00018/&prev=search>

<https://translate.google.com/translate?hl=fr&sl=en&u=http://www.hotmcu.com/nodemcu-lua-wifi-board-based-on-esp8266-cp2102-module-p-265.html&prev=search>

<http://hobbycomponents.com/displays/74-i2c-serial-lcd-1602-module>

Les figures:

- Fig.1 : La VeraEdge, de Vera Control LTD, bien que petite, est très complète et ne coûte que 190€.
- Fig.2 : Carte Arduino Méga 2560
- Fig.3 : Carte microcontrôleur ATmega2560
- Fig.4: CBGA-pinout ATmega640/1280/2560
- Fig.5: TQFP-pinout ATmega640/1280/2560
- Fig.6 : ESP8266 12E et leur description
- Fig.7 : Servomoteur 9g
- Fig.8 : déplacement angulaire allant de -45° à $+45^\circ$
- Fig.9 : Inversion du sens de rotation du moteur
- Fig.10 : Capteur de flamme
- Fig.11 : Schéma de principe d'un capteur pyroélectrique (technologie IR)
- Fig.12 : Schéma de principe d'un capteur UV
- Fig.13 : Capteur d'humidité DHT11/ application typique
- Fig.14 : les bases pour communiquer avec un DHT11
- Fig.15 : Capteur de gaz MQ6
- Fig.16 : Capteur de mouvement
- Fig.17 : Ecran LCD avec un I2c en arrière
- Fig.18 : Buzzer passive
- Fig.19 : Présentation de l'interface initiale du logiciel
- Fig.20 : illustration de la maquette utilisée dans le projet
- Fig. 21 : L'organigramme représente le chemin des requêtes
- Fig.22 : l'organigramme de la carte Arduino Méga
- Fig.23 : l'organigramme de la carte ESP8266 12E

Les tableaux :

- Tableau 1 : Les périphériques utilisés et leurs prix
- Tableau 2 : fiche technique de l'Arduino Méga
- Tableau 3 : Localisation des périphériques utilisés dans la maquette