



Projet fin d'étude

Réalisation d'une maison intelligente



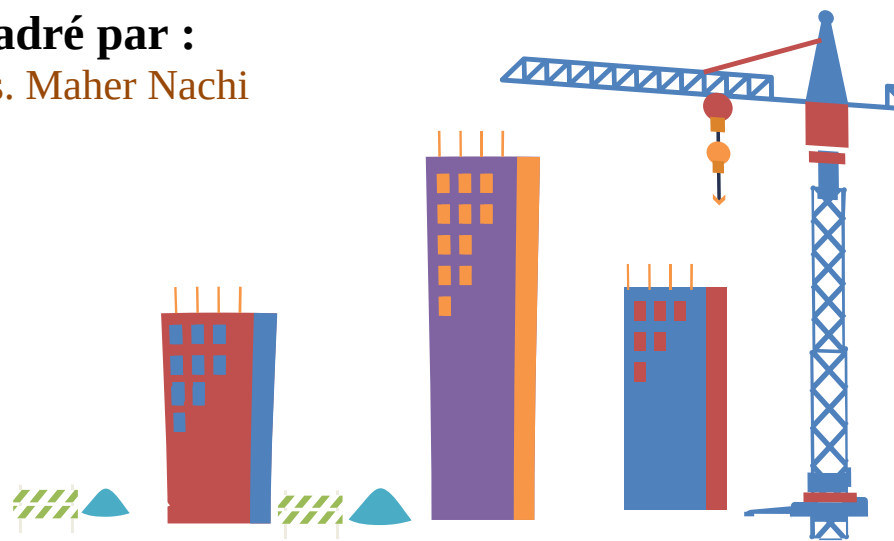
Élaboré par :

Gouider Mohamed ilyes

Nefaa hamdi

Encadré par :

Ms. Maher Nachi



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
souhaitez. 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

**éclairer le monde par leur savoir, A 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, 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 lier à 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

RESUME

- Introduction
- Introduction general
- CHAPITRE 1 :La DOMATIQUE

I. Introduction

II. Définition de la domotique

III. Différents domaines d'application de la domotique

1. Domotique pour le confort
2. Domotique pour l'énergie
3. Domotique pour la sécurité
4. Communication :

IV Maison communicante

V Combien coute la domotique ?

1. But du Projet
2. Conclusion

CHAPITRE 2 : MATERIELS ET LOGICIELS UTILISES

I Schéma synoptique du système à réaliser

II Description des shields utilisés

III Domaine d'utilisation et ces applications

IV Présentation de la carte

1. Qu'est-ce qu'un microcontrôleur
2. Microcontrôleur ATMEL ATMega328
3. Capteur humidité & température DHT11 :
4. Capteur de gaz MQ6 :
5. Module de Relais
6. Description du module de connexion Wifi ESP12
7. Contacts magnétiques

V Logiciels

1. L'Arduino :
2. CADIO

CHAPITRE 3 : CONCEPTION DE LACARTE ELECTRONIQUE

I Introduction

II Principes de réalisation

1. Tracé manuel du circuit imprimé
2. Utilisation d'un typon

II Mode opératoire

III Implantation des composants et soudure

1. Perçage du circuit imprimé
2. Test des pistes du circuit imprimé
3. Préparation des composants
4. Utilisation du fer à souder

IV Conclusion

CHAPITRE 4 : DESCRIPTION DU PROJET

I Introduction

II L'objectif du projet

III Les étapes de la réalisation

L'architecture Générale de notre système

V La conception du notre système

VI Réalisation pratique du système

1. La Réalisation de carte électronique
2. Édition du schéma électrique et simulation du fonctionnement...
3. Montage sur plaque d'essai
4. Souder les composants

VII Fabrication de la maison :

1. Structure de la maison :
2. Périphériques utilisées et leurs positions
3. Automatisation de la maison

Capteur de mouvement PIR :

Capteur d'humidité DHTT11 :

Capteur de gaz et contact magnétique

Capteur RGB :

4. Commander l'éclairage à distance
5. Centralisation des commandes : Interface Web
6. L'interface des capteurs :

Conclusion Générale

ANNEXE ET Pprogrammes

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'arriviez. 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 30000DT 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 1000 à 3000DT environ, en restant dans le « grand public », car il y a beaucoup plus cher. Disonsque pour 1600DT 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 1000DT.

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 : 1600DT
- Gestion des volets : 1500DT
- Gestion du chauffage : 800DT
- Gestion de l'éclairage : 2000DT
- Gestion des appareils : 1200DT
 - Sécurité : 2500DT

Soit un total de 10000DT

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 10000DT 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	15 DT
Capteur d'humidité	X 1	13 DT
Arduino Méga	X 1	45DT
ESP8266 12E	X 1	38DT
VOYANT	X 4	20 DT
LED 5mm	X 3	18 DT
MODUL RELAIS	X 1	35 DT

Tableau 1 : Les périphériques utilisés et leurs prix

Soit un total un peu prés de : 200 DT

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 basée 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.

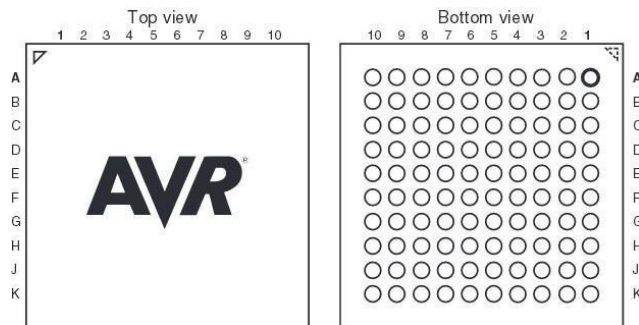


Fig.4 : CBGA-pinout ATmega640/1280/2560

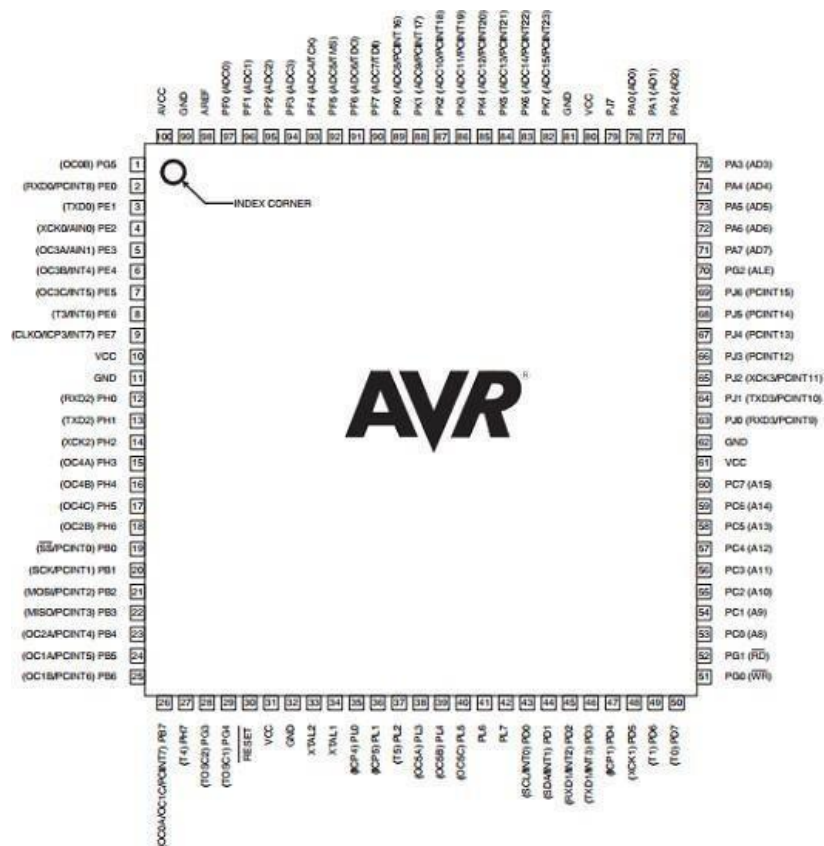


Fig.5 : TQFP-pinout ATmega640/1280/2560

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 :

Il y a deux autres broches disponibles sur la carte :

AREF : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction analogReference ().

Reset : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

2. ESP8622 12E :

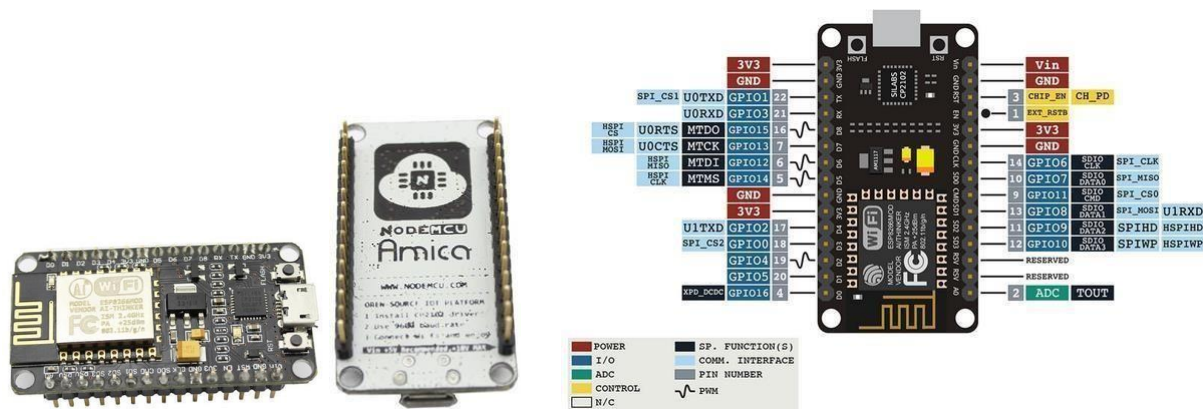


Fig.6 : ESP8266 12E et leur description

Caractéristiques :

- Wi-Fi Module - Module ESP-12E similaire à ESP-12 module, mais avec 6 GPIOs supplémentaires.
 - Module ESP8266 ESP-12E
- USB intégré Adaptateur UART série (SiliconLabs CP2102)
 - Bouton de réinitialisation
- Touche d'entrée (également utilisé pour bootloading)
- Montage en surface, LED rouge contrôlable par l'utilisateur
 - régulateur de tension 500mA 3.3V (LM1117)
- Deux entrées d'alimentation protégée par diode (l'un pour un câble USB, une autre pour une batterie)
- Têtes - 2x 2,54 mm en - tête à 15 broches avec accès à GPIO, SPI, UART, CAN et broches d'alimentation
 - Alimentation - 5V via port micro USB
 - Dimensions - 49 x 24,5 x 13mm

Presentation:

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. Capteur humidité & température DHT11 :

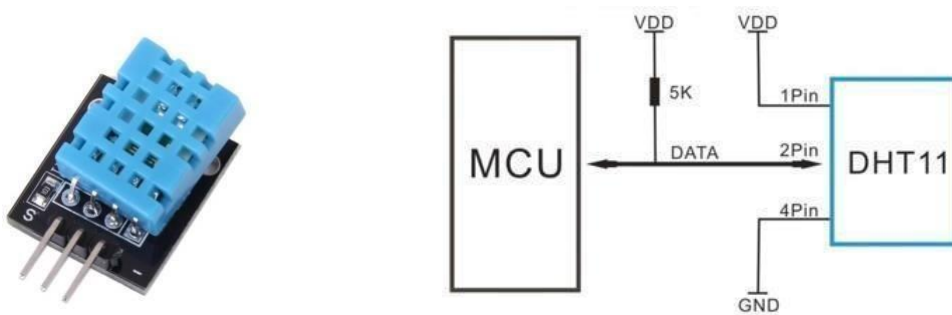


Fig.13 : Capteur d'humidité DHT11/ application typique

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

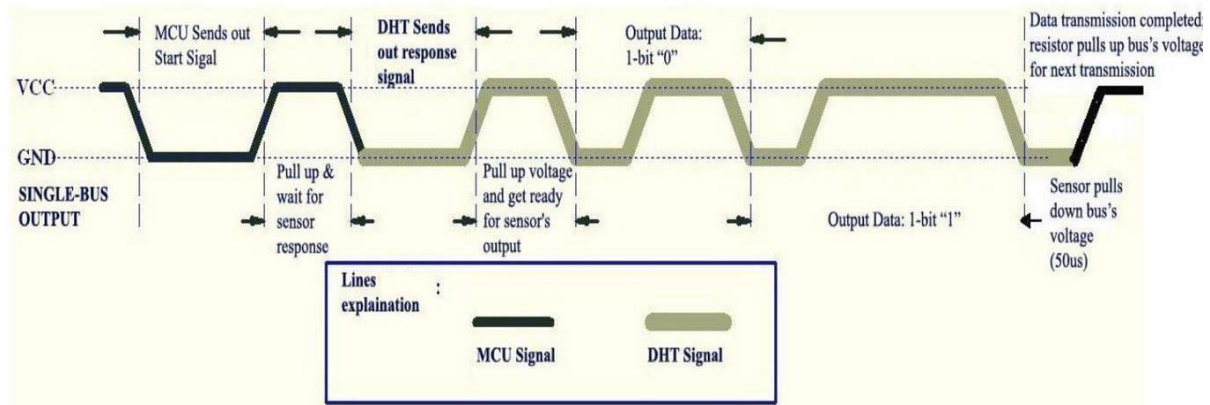


Fig.14 : les bases pour communiquer avec un DHT11

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-qu'il 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.

4. Capteur de gaz MQ6 :

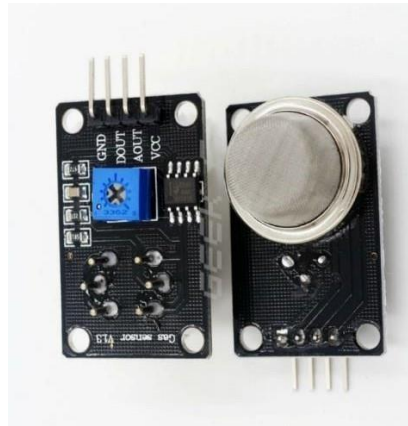


Fig.15 : Capteur de gaz MQ6

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.

5. 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

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 :

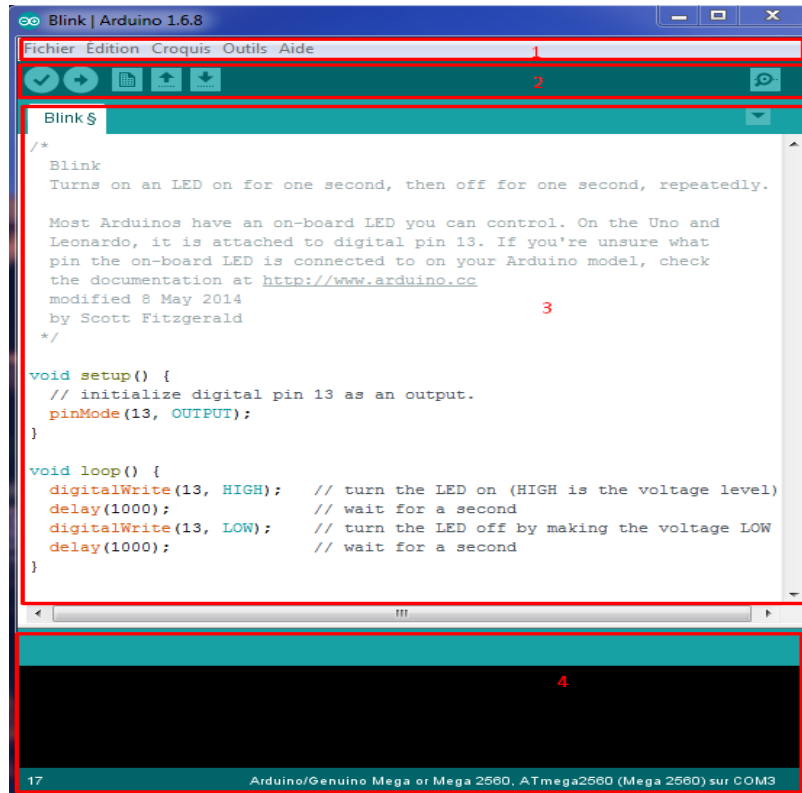


Fig.19 : Présentation de l'interface initiale du logiciel

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

Dans ce chapitre, nous allons développer les étapes nécessaires pour Aboutir à notre objectif et faciliter la réalisation de notre projet « une sécurisation d'une smart home ai

La maquette de la maison est équipée de divers capteurs qui effectuent des mesures telles que le DHTT11 et un système de sécurité.

La particularité de cette technologie c'est qu'elle permet à l'utilisateur de contrôler à distance sa maison en toute sécurité.

Avant de présenter notre solution on va analyser et étudier tout d'abord l'architecture et le fonctionnement de notre système ;ses objectifs, et les différentes étapes de la réalisation pratique...etc.

II. L'objectif du projet

Notre projet vise à créer un système intelligent de type maison intelligente, grâce à des modules et des capteurs moins chers, plus efficaces et plus faciles à utiliser conçus pour: Assurer un grand confort ,La sécurité, une priorité absolue qui est parmi les principales bases de la domotique. Il est considéré comme la priorité des priorités dans la majorité des domiciles. Implique ZERO gaspillage en économie d'énergie.

III. Les étapes de la réalisation

La réalisation de notre dispositif a connu cinq étapes :

Première étape : C'est l'étape la plus difficile et la plus importante car elle consiste en la recherche des informations bibliographiques et techniques à utiliser pour aboutir à l'objectif.

Deuxième étape : Test du montage dans l'environnement « puis programmation de ESP12 et Arduino Mega avec L'IDE par le code obtenu après compilation du programme écrit.

Troisième étape : réalisation du montage complet sur une plaque d'essai puis sur maquette afin d'assembler Et enfin la programmation du système par l'IDE

Quatrième étape : la création d'une application sous Android (création d'interface avec la programmation orienté objet de notre application)

IV. L'architecture Générale de notre système

On propose une architecture d'un système qui nous permet d'atteindre des solutions pour ces problèmes. La figure si dessus représente l'architecture générale de notre système :



Fig 1 : Architecture du système

Cette architecture permet :

D'apporter un confort supplémentaire au sein de nos habitations. Automatiser une maison et l'équiper efficacement simplifient la vie de ses habitants tout en leur offrant un gain de temps optimal au quotidien. Il peut s'agir d'une lumière qui s'allume dès que quelqu'un entre dans une pièce ou d'un appareil électrique qui s'éteint lorsque vous vous absentez.

- Activer une alarme en cas d'absence
 - Régler le chauffage selon la température
 - contrôler l'éclairage à distance

- Automatiser de l'arrosage extérieur
- Assurer une maison pleinement sécurisée.

V. La conception du notre système

Le système conçu se compose de trois parties principales :

- L'application mobile agit comme un serveur local installé dans le smartphone, permettant le contrôle à distance de la maison intelligente.
- Dans notre cas, nous utiliserons le microcontrôleur ATMEGA ET ESP28WIFI et les capteurs intégrés, tels que le capteur de gaz MQ-2, le capteur à ultrasons, le capteur de température DHTT11, etc., pour l'appareil intelligent qui collecte les données.

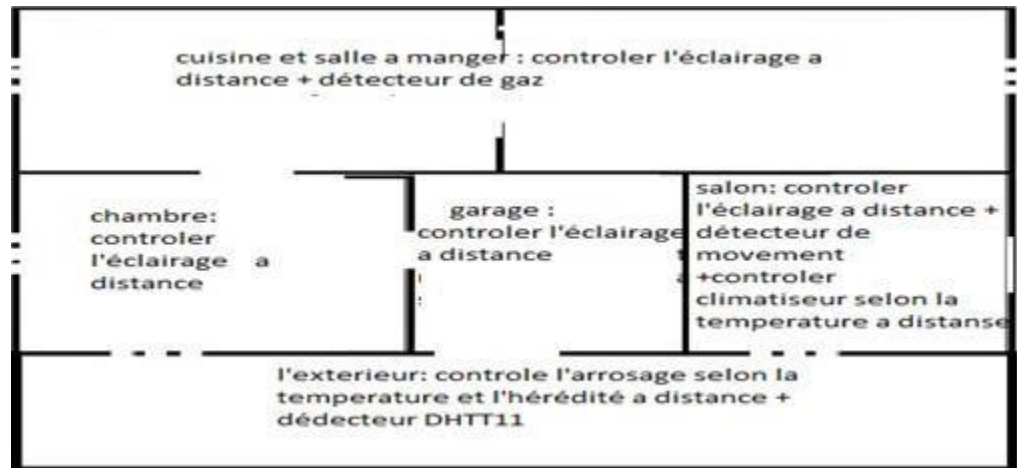
Voici une explication de l'architecture de la conception de notre solution :



Figure 2 : Conception du notre système

VI. Réalisation pratique du système

3. 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 120cm sur 80cm constitue d'une chambre, une cuisine, un garage, s, un salon, et une salle à manger.

4. Périphériques utilisées et leurs positions :

Cette maquette, permettrait de présenter certaines fonctionnalités de la domotique à travers 4 Capteurs (Gaz, Humidité et Température, Mouvement, et contact magnétique.) un RGB et autres périphériques. Ces scénarios seront automatisés via la module WIFI ESP12 Exécutant des programmes informatiques.

LOCALISATION	PERIPHERIQUES
La chambre	- Spote 12V
Salle à manger et cuisine	- Spote 12V
	- Capteur de gaz MQ6
	- Voyant rouge 220V
SALON	- Spote 12V
	- Ventilateur
	- Capteur de mouvement

Garage

- Spote 12V

5. Automatisation de la maison :

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
 - Un spot
 - Un microcontrôleur ATMéga et ESP12

Capteur d'humidité DHTT11 :

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.

Donc on cas ou la température et supérieur à 30° le climatiseur il va déclenche

Si la température et inférieur à 20° la chaudière il va déclenche

Si l'humidité et inferieur a 10 l'arrosage il va déclenche

Nous avons utilisé comme matériel :

- Un capteur DHTT11
- Deux voyant 220V
- Une ventilateur
- Un microcontrôleur ATMEGA et ESP12

Capteur de gaz et contact magnétique :

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 et stopper les voleur.

Ainsi nous avons associé différents composants

Afin d'utiliser un détecteur de gaz dans la cuisine pour une bonne sécurité. Ces deux détecteurs déclenchent une alarme en cas d'une fuite de gaz

Le matériel utilisé pour le gaz est le suivant :

- Un détecteur de gaz
- Un microcontrôleur ATMEGA et ESP12

Commander l'éclairage à distance

La domotique est de plus en plus présente dans les habitations contemporaines et parfois elle est exigée par certains habitants. Il faut reconnaître que la maison domotique (smart house) n'est pas toujours à la portée de tout le monde. Pour cela, cette simple application va nous permettre de réaliser un aspect de « la MAISON INTELLIGENTE ». Les espaces sont éclairés par le biais d'un téléphone portable (un android ou un Smartphone) . Cette application s'appelle KADIOL » et on peut la télécharger du portable à partir du play store :

- / 1 ère lampe pour éclairer le salon
- / 2ème lampe pour éclairer la cuisine
- / 3ème lampe pour éclairer la chambre
- / 4ème lampe pour éclairer Le garage

Dans ce tutoriel, nous allons voir comment réaliser un système d'éclairage connecté.

Pour ce projet nous aurons besoin du matériel suivant :

- Un module wifi ESP12,
 - Un relais,
- Une résistance 220 ohm,
- Une LED qui représente n'importe quel éclairage ou prise de notre maquette,
- Des câbles de connexion.

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 ou smart phone .

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'ESP12 E, dont on a choisi de le configurer autant que serveur avec une page Web accessible via l'adresse IP.

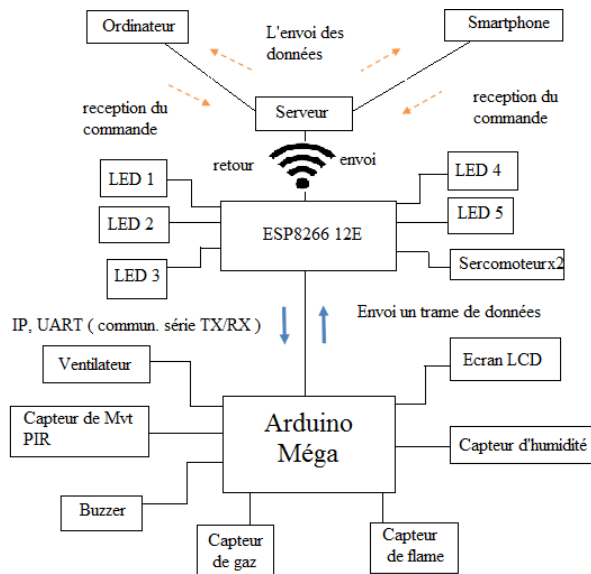
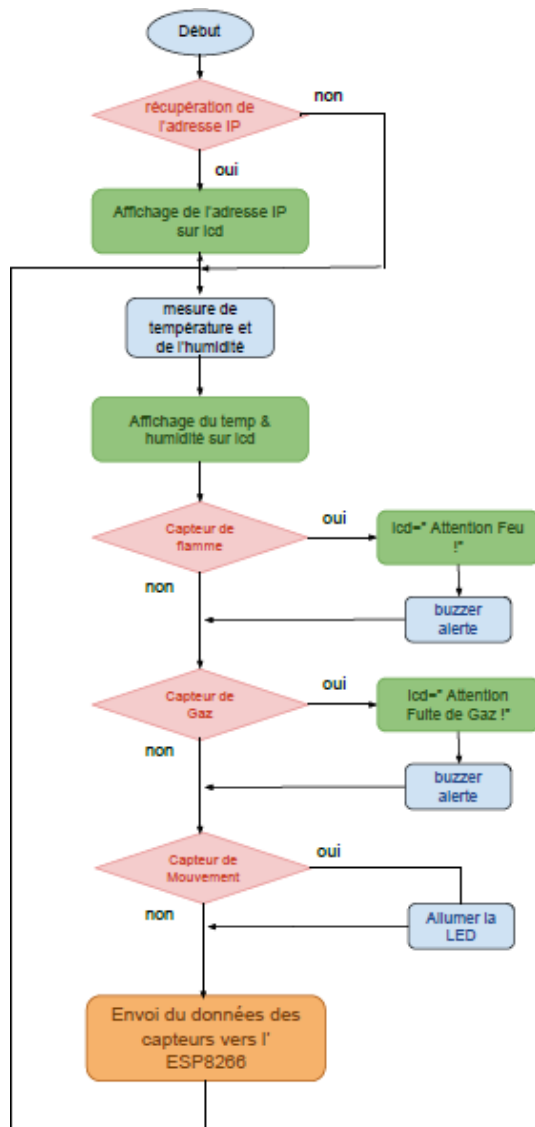


Fig. 21 : L'organigramme représente le chemin des requêtes



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

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

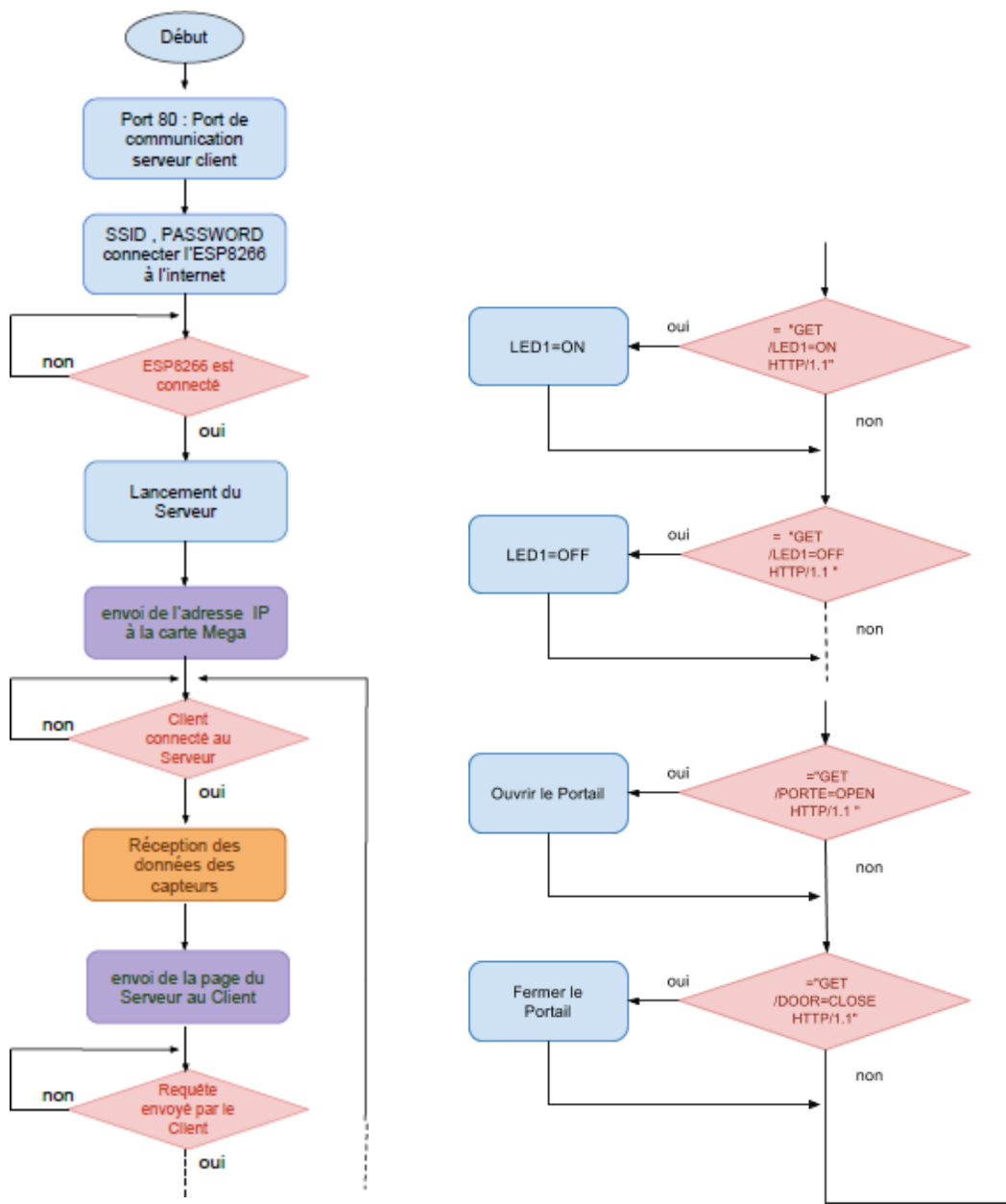
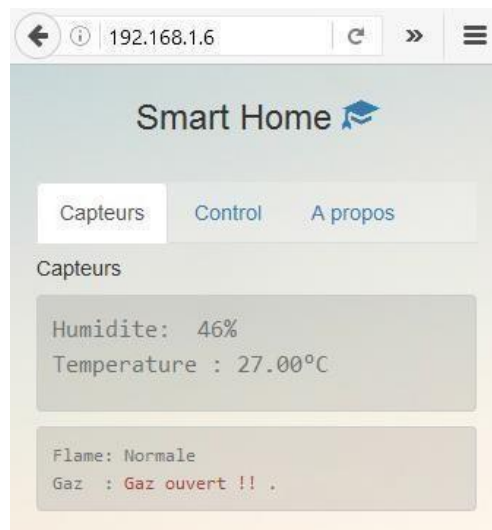


Fig.23 : l'organigramme de la carte ESP8266 12E

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.

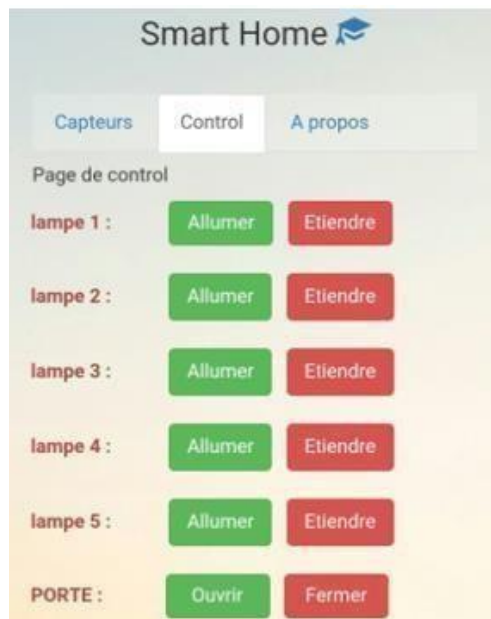
L'interface des capteurs :

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

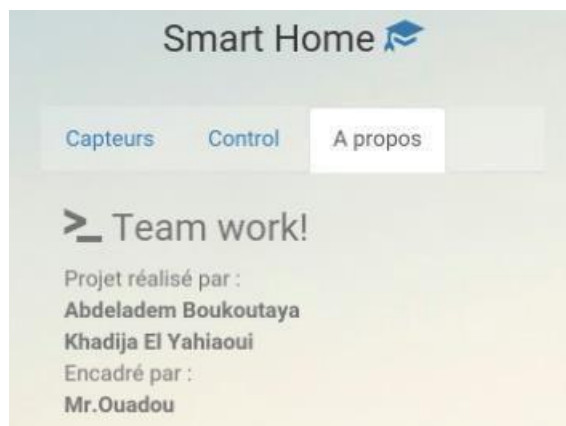


L'interface de commande :

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 qi 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 ...

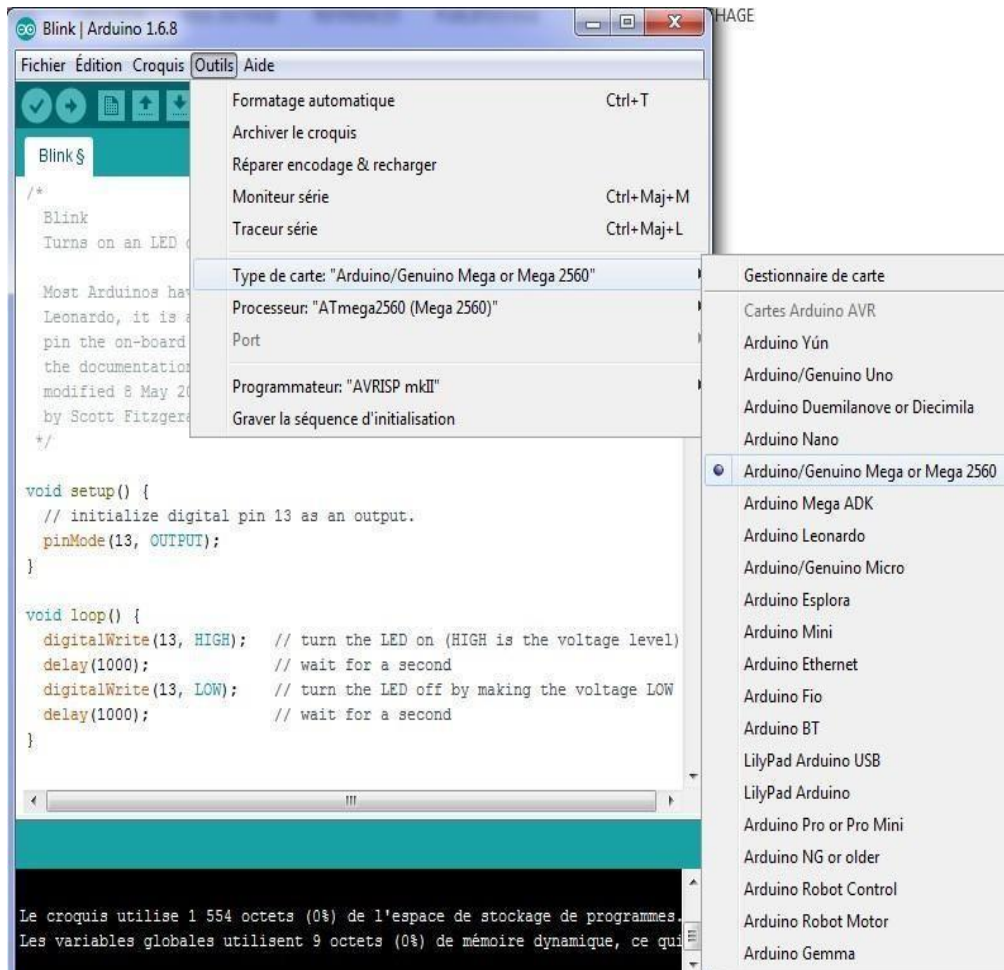
Ce projet a été vivant, entraînant et motivant pour la suite de nos études. Nous pensons avoir entraperçu une partie de notre future vie active.

Annexes et Programmes

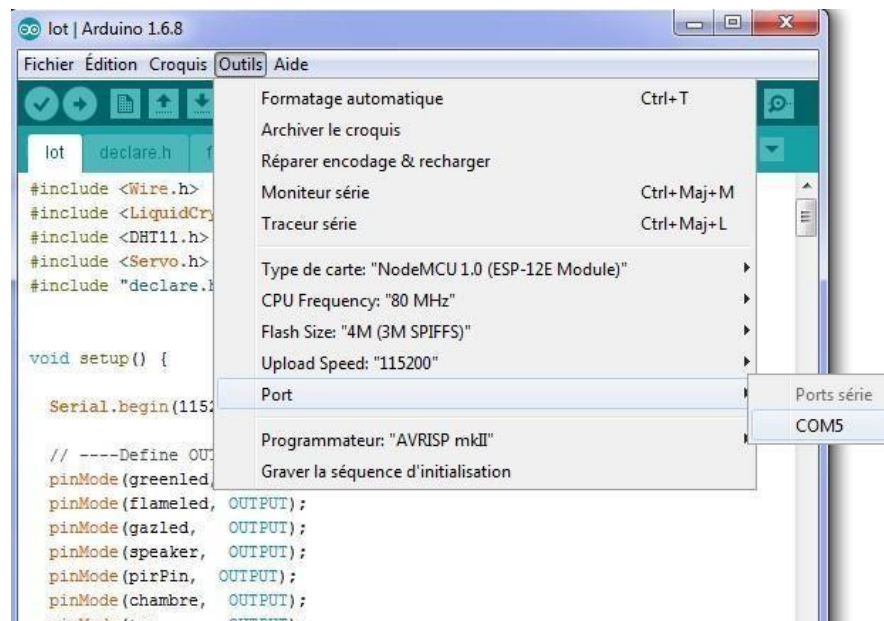
Découvrir Arduino :

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

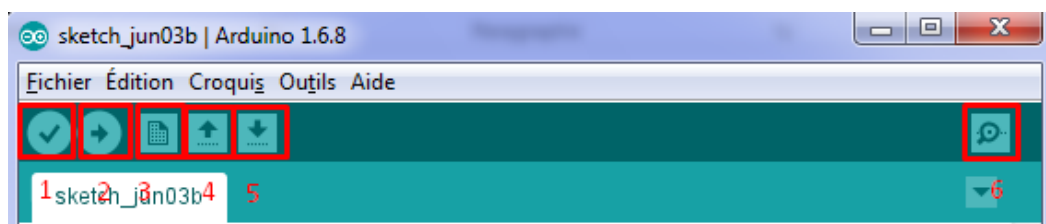
Choisir le type de carte :



Serial Port / COM Port:

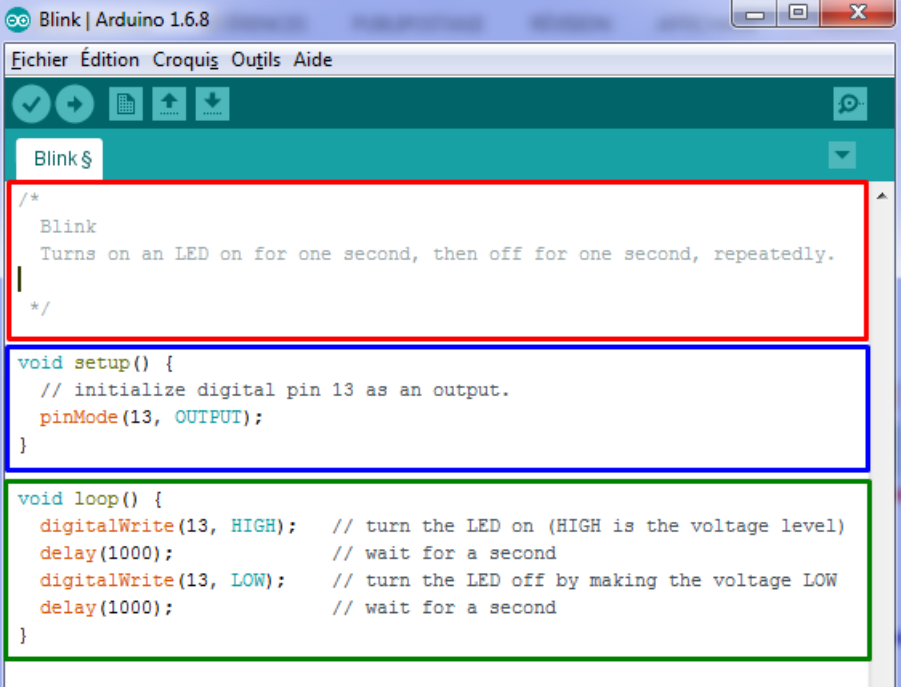


Charger un programme dans la carte :



- 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.
 */

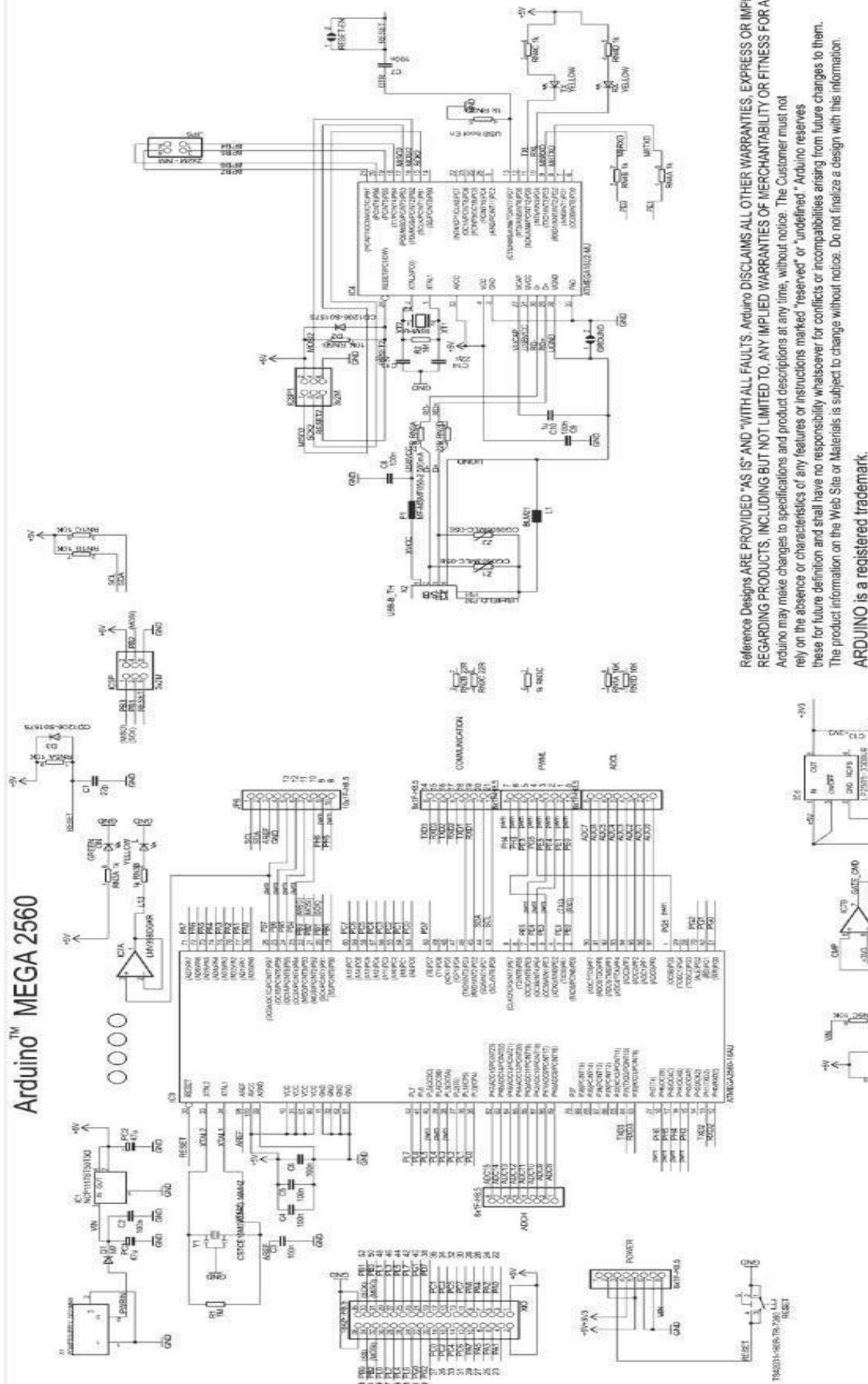
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

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

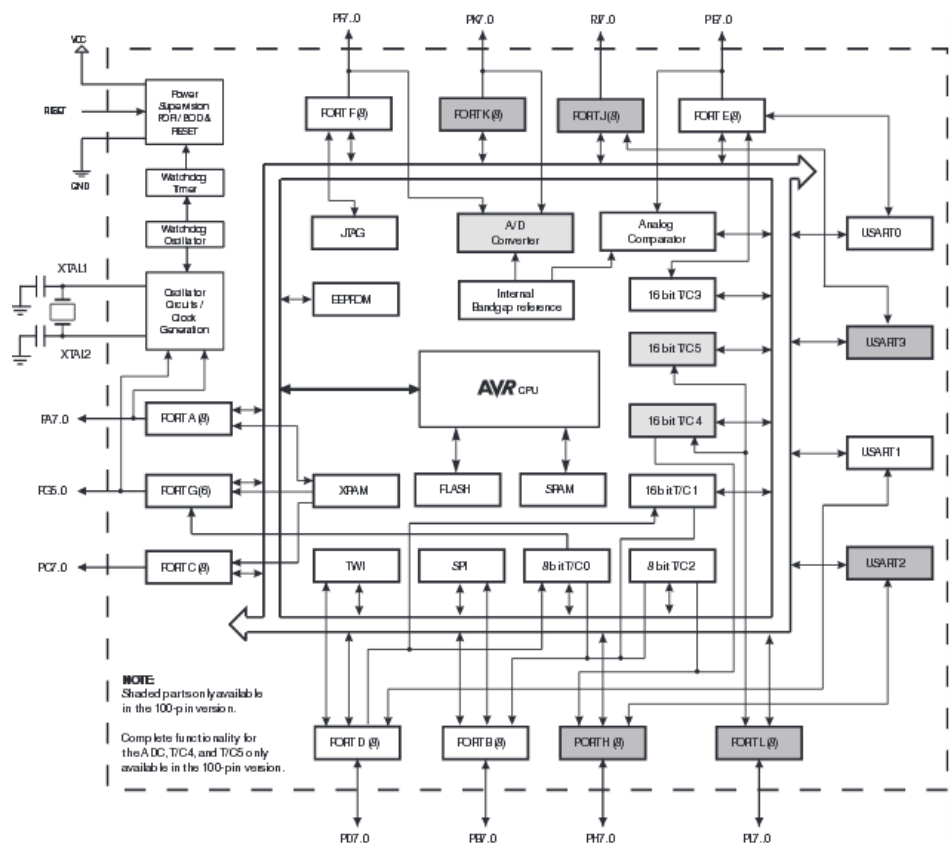
installation /
étirement

boucle /
jouer le jeu

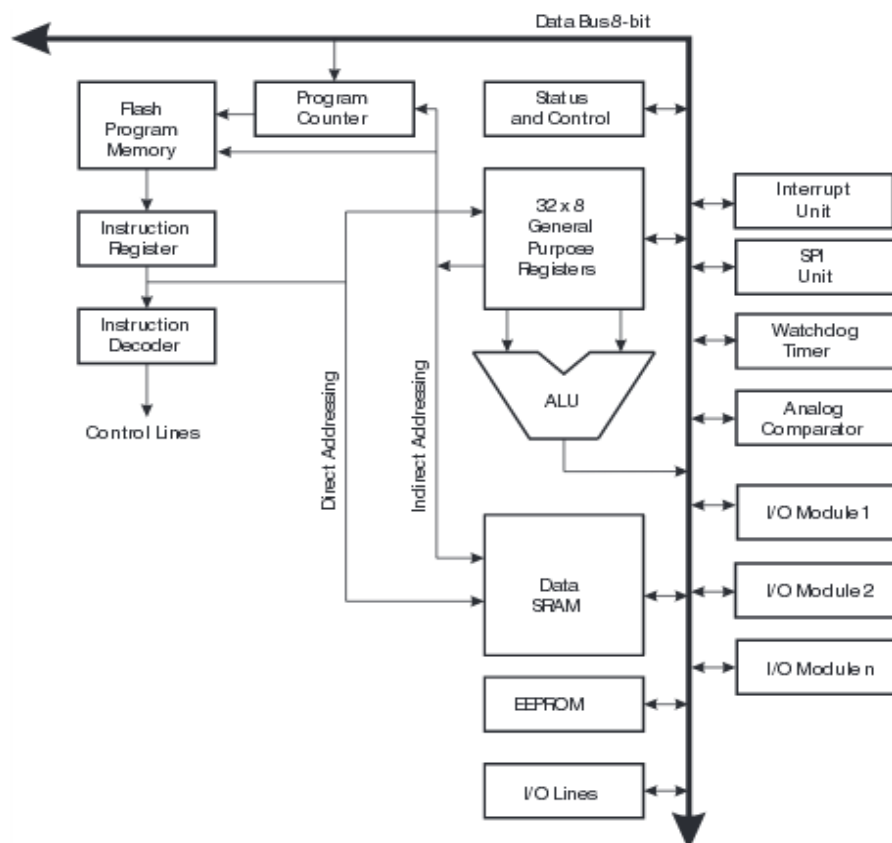


Reference Designs are provided "AS IS" AND "WITH ALL FAULTS." Arduino disclaims all other warranties, express or implied. REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product 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 gazPin = A1;

int dhtPin = 12;
int speaker = 10;

/*----- Leds */
int gazled = 22;
int greenled = 24;

/*----- Var -----*/
long lecture_echo ;
long cm ;
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(gazPin, INPUT);
  pinMode(dhtPin, INPUT);

  // -----Define OUTPUT pins

  pinMode(greenled, OUTPUT);
  pinMode(gazled, OUTPUT);
```

```

pinMode(speaker, OUTPUT);

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

void loop() {
    capteurs();
    }

    void capteurs(){

        Gaz_test();
        dht_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+"|";

    }

```



```

/*****
**
*
*      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 = "domotique";
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="", gaz="", humi="", temp="", us="";

            recu= Serial.readString();

            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: ");

```


[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;  
  
}
```

