

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE Mouloud MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes De MASTER PROFESSIONNEL

Domaine : Sciences et Technologies

Filière : Génie électrique

Spécialité : **Automatique et informatique
industrielles**

Présenté par
Mr GRIBISSA Aghilas

Thème

Commande domotique d'un climatiseur.

Mémoire soutenu publiquement le 29 /09/ 2016 devant le jury composé de :

KASRI Ahmed	M A A	Président	U. M. M. Tizi-Ouzou
BENSIDHOUM	M C A	Encadreur	U. M. M. Tizi-Ouzou
MOULOUDJ Kamel	INGENIEUR	Co-encadreur	U. M. M. Tizi-Ouzou
BOUDEMAA Fadhila	M A A	Examineur	U. M. M. Tizi-Ouzou
YOUSFI Safia	M A A	Examineur	U. M. M. Tizi-Ouzou

Remerciements

Je souhaite exprimer mes vifs remerciements à :

-Monsieur BENSIDHOUM Mohand Outahar, mon promoteur, sans qui, ce mémoire n'aurait pas eu lieu. Je le remercie aussi pour son soutien scientifique et hors scientifique et la confiance qu'il m'a accordée depuis que je suis au département d'automatique. Je me souviendrai toujours de lui comme d'un homme patient, compréhensif et très sage avec tout le monde surtout avec les étudiants. En fin, rien de ce que j'écrirai ici ne sera suffisant pour lui exprimer ma gratitude.

-Monsieur MOULOUDJ Kamel, professeur à l'école technique de Tizi-Ouzou. Je tiens à lui témoigner toute ma gratitude pour toute la confiance qu'il m'avait accordée dès le premier jour à l'école technique de Tizi-Ouzou en mettant à ma disposition les moyens nécessaires au bon déroulement de mon travail. Sa sympathie 'légendaire', sa disponibilité malgré les nombreuses sollicitations dont il est l'objet, son expérience et ses compétences m'ont été très bénéfiques et cela dans une ambiance très chaleureuse. A lui aussi, rien de ce que j'écrirai ici ne serait exprimer ma profonde reconnaissance.

-Monsieur LAGHROUCHE Mourad que je remercie pour m'avoir orienté et pour toute l'aide qu'il m'a fournie.

-J'exprime aussi tout ma gratitude à tous les enseignants qui ont contribué à ma formation depuis mon jeune âge.

-Enfin, mes remerciements les plus sincères à tout le personnel administratif, ingénieurs, techniciens supérieurs, et ouvrier de ENIEM pour m'avoir encadré tout le long du stage que j'ai passé dans leur entreprise qui a duré deux mois.

-Je remercie aussi les membres du jury qui ont su estimer ce mémoire de fin d'études à sa juste valeur.

-Ma gratitude va tout particulièrement aussi à mes chers parents et surtout à ma mère qui a été toujours derrière moi et sans laquelle je ne serai pas ce que je suis devenu.

Table des matières

Introduction générale	1
-----------------------------	---

Chapitre I : Cahier des charges

I.1-Introduction	2
I.2-Composants d'un système de réfrigération	2
I.2.1-Compresseur	2
I.2.2-Condenseur et Evaporateur	2
I.2.2.1-Condenseur	3
I.2.2.2-Evaporateur	3
I.2.3-Détendeur	3
I.2.4-Fluide frigorigène	3
I.3-Fonctionnement d'un système réfrigérant a compression de vapeur	4
I.4-CLIMATISEURS DIT «MONO SPLIT»	6
I.4.1-Emplacement de l'unité extérieure.....	8
I.4.2-Composition du CLIMATISEUR SPLIT SYSTEME.....	8
I.4.2.1-Unité extérieure	8
I.4.2.2-Unité intérieure	9
I.4.3-Fiche technique sur différent composants d'un climatiseur	11
I.4.3.1-Moto-ventilateur + hélice.....	11
I.4.3.2-Turbine	11
I.4.3.3-Compresseur	12
I.4.3.4-Fluide frigorigène.....	13
I.4.4-Inversion de cycle.....	14
I.4.4.1-Vannes quatre (4) voies	16
I.4.4.2-Dégivrage.....	17

I.5-Contrôle électronique du climatiseur	18
I.5.1-Cheema détailiez de la Catre électronique et c'est composent	18
I.5.2-Rôle de la carte électronique	20
I.5.3-Télécommande à infrarouge	20
I.6-Fonctionnement du système à réaliser	20
I.6.1-Mode ventilateur	21
I.6.2-Mode froid	22
I.6.3-Mode chaud	22
I.6.4-Mode antigivrage	23
I.7-Cahier de charge	24
I.7.1-Mesure de température d'ambiance	24
I.7.2-Mesure de température des échangeurs thermiques	24
I.7.3-capteurs de donnes numériques	24
I.7.4-Unité d'acquisition et de traitement	24
I.7.5-Unité de commande et interface graphique	24
I.7.6-Du point de vue technique, le système doit être	25
I.8-conclusion	25

Chapitre II : Choix des composants et des langages de programmations

II.1-Introduction	26
II.2-Structure du système	26
II.3-Module d'acquisition et de traitement de données	28
II.3.1-Présentation de la carte Arduino Nano	28
II.3.2-Présentation du capteur thermique LM35 :	28
II.3.3-Récepteur infrarouge	31
II.3.4 Bluetooth HC-05	32
II.4-Module de commande et d'affichage	33
II.4.1- afficheur LCD 4*20 (JHD 204A)	33
II.4.2- Carte relais huit canaux	34

II.4.3- Buzzer piézo-électrique	34
II.5-Module serveur WEB	35
II.5.1-Carte réseau arduino	35
II.6-Choix de langage de programmation	36
II.6.1-C Arduino	37
II.6.2-App Inventor	38
II.6.2.1-Présentation d'Android.....	38
II.6.2.2-Présentation App inventeur 2.....	38
II.6.3-Programme HTML et CSS	39
II.7-Conclusion	40

Chapitre III : Réalisation du système de contrôle

III.1-Introduction	41
III.2-Réalisation	41
III.2.1-Réalisation matérielle	41
III.2.1.1-Premier partie	42
III.2.1.1.1-Carte Arduino-Nno	43
III.2.1.1.2-Récepteur infrarouge	43
III.2.1.1.3-Module Bluetooth HC05	45
III.2.1.1.4-Capteur LM35	47
III.2.2.1-Deuxième partie de la carte électronique.....	51
III.2.2.1.1-Carte Arduino-Uno	51
III.2.2.1.2-Buzzer	52
III.2.2.1.3-Carte relais	52
III.2.2.1.4-Afficheur LCD 4*20.....	53
III.2.3.1-Troisième parité	55
III.2.3.1.1- carte arduino nano	55
III.2.3.1.2-Carte réseauRJ45	56
III.3-Méthode de contrôle du système par la carte électronique	58
III.3.1-Mode ventilateur	58
III.3.2-Mode froid	58

III.3.3-Mode chaud	58
III.3.4-Vitesse de la turbine	58
III.3.5-GRAF7.....	59
III.4-Différent modes de contrôle du climatiseur	59
III.4.1-Commande infrarouge	60
III.4.1.1-Fonctionnement de la télécommande	60
III.4.2-Commande Bluetooth	61
III.4.2.1-Mode bouton.....	62
III.4.2.2-Mode vocal.....	63
III.4.3 -Commande réseau	64
III.5- installation	66
III.6-Conclusion	67
III.7-Conclusion	68
Conclusion générale et perspective	69

Introduction générale :

Le froid est devenu un phénomène physique important dans notre société, utilisé dans un grand nombre de secteurs et sous de nombreuses formes (conservation de denrées périssables, climatisation, refroidissement de procédés industriels etc..). Généralement sa production s'appuie essentiellement sur des techniques classiques de compression et de détente des fluides frigorigènes.

Toutefois l'électronique moderne et la technologie informatique ont apporté à la production de froid de nouveaux champs d'application. Dans le domaine de la climatisation, les climatiseurs actuels sur le marché algérien sont assez performants côté consommation énergétique, grâce à une technologie qui permet un mode réversible sans surplus de consommation d'énergie. Ils proposent aussi une large gamme d'options pour le fonctionnement des climatiseurs.

L'objectif de notre travail, grâce à l'utilisation de la carte Arduino-uno, nous allons réaliser de nouvelles options au climatiseur split-système au climatiseur 12000 BTU de l'ENIEM.

Ce mémoire est organisé en trois chapitres à travers lesquels, nous décrivons le travail effectué pour la conception et la réalisation de notre système.

Dans le premier chapitre nous présentons le fonctionnement général d'un climatiseur Split-système et les spécificités du cahier de charges sur lequel nous nous sommes basés pour la réalisation de notre projet.

Le second chapitre décrit la phase de conception du système. On y décrit les principaux composants constituant notre carte électronique en justifiant leurs choix.

Le troisième et dernier chapitre présente les étapes de réalisation du système. On y découvre les schémas électriques détaillés de différentes parties du système, ses modes de fonctionnement ainsi que la mise en place de la partie logicielle et les résultats obtenus.

Nous concluons notre travail par une conclusion générale et les perspectives.

Chapitre I

Fonctionnement général

Du climatiseur

I.1-Introduction :

La climatisation :

Le conditionnement d'air consiste à traiter l'air dans un espace où les gens résident afin d'avoir une agréable température, un taux d'humidité optimal et assurer une bonne circulation d'air purifié.

Cela est le cas pour le refroidissement, plus spécifiquement le refroidissement de confort, et le chauffage. Dans le refroidissement de confort (ou le chauffage) on envisage une température de consigne comprise entre 20 et 25 °C. Outre le refroidissement de confort, on parle également du refroidissement tout court lorsqu'il s'agit de refroidir avec une température de consigne en dessous de 18°C. Ceci est une distinction importante. En chauffage on a une température cible fixée de 21 °C.

Il existe des climatiseurs à la base des pompes à chaleur (réversible en climatisation). Ils fournissent également, outre la possibilité du chauffage, le refroidissement.

Les 5 caractéristiques suivantes sont essentielles à un environnement idéal et sain :

Température : Un climatiseur maintient la température à un niveau constant et idéal pour l'organisme. Moins le corps est soumis aux fluctuations de température mieux est pour le fonctionnement de l'organisme.

Humidité : Non seulement le climatiseur contrôle la température mais il permet également, pendant les jours chauds de l'été, de déshumidifier l'air afin de créer un climat intérieur agréable. Généralement on détermine le taux d'humidité relatif idéal entre 40 et 60%.

Pureté : Chaque jour, notre organisme est exposé à de l'air contaminé, grâce aux filtres du climatiseur, les substances nocives, telles que la poussière, les pollens, la fumée et les bactéries, sont filtrées de l'air pollué. Un climatiseur veille à ce que nous ayons de l'air frais à tout moment.

Circulation d'air : Des courants d'air sont causés par la circulation trop rapide de l'air dans une pièce. Ces courants causent un refroidissement partiel ou entier du corps, en conséquence, des rhumes se produisent facilement. Les systèmes actuels de climatisation n'entraînent plus de courants d'air.

Ventilation : Une unité de conditionnement d'air peut également fonctionner en simple ventilation. L'air intérieur pollué est remplacé par de l'air purifié, sans rajout de chaud ou de froid. Cela constitue en soi un sentiment de fraîcheur, en raison du mouvement de l'air.

I.2-Composants d'un système de réfrigération :

Il y'a plusieurs composants mécaniques nécessaires pour un système de réfrigération. Nous allons présenter notre travail sur les composants les plus essentiels au fonctionnement d'un système de réfrigération à compression de vapeur qui sont :

- Compresseur,
- Condenseur,
- Evaporateur,
- Détendeur,
- Fluide frigorigène

.

I.2.1-Compresseur :

La fonction du compresseur est de fournir de l'énergie au fluide pour lui permettre de passer de la basse pression à la haute pression, ainsi de rejeter dans le condenseur, la chaleur absorbée de l'évaporateur. La contrepartie est l'absorption d'une quantité de chaleur supplémentaire par le fluide durant la compression.

Il a également pour tâche d'aspirer, hors de l'évaporateur, le fluide frigorigène évaporé et de le refouler dans le condenseur.

D'un autre point de vue le compresseur sépare le côté basse pression du côté haute pression.

I.2.2-Condenseur et l'évaporateur :

Le condenseur et l'évaporateur d'une machine frigorifique sont essentiellement des échangeurs de chaleur. Nous pouvons les classer d'ailleurs tous les deux dans la rubrique d'échangeur thermique, car leur but essentiel est d'assurer le passage du flux calorifique du fluide frigorigène au milieu extérieur (condenseur) ou bien du milieu à refroidir au fluide frigorigène (évaporateur).

Dans le cas particulier où on sélectionne le mode froid dans un climatiseur, l'échangeur thermique de l'unité intérieure va jouer le rôle de l'évaporateur et celui de l'unité extérieure va jouer le rôle du condenseur.

Dans le cas plus particulier où on sélectionne le mode chaud dans un climatiseur l'échangeur thermique de l'unité intérieure va jouer le rôle du condenseur et celui de l'unité,

extérieure va jouer le rôle de l'évaporateur.

I.2.2.1- Le condenseur :

Le condenseur a pour fonction de transporter et d'expulser la chaleur hors du circuit du fluide frigorigène pour ne pas l'absorber. Cette chaleur est cédée à l'air. La température de condensation doit toujours être supérieure à la température d'entrée de l'air ambiant.

Le fluide frigorigène est condensé à une température située au-dessus de la température ambiante (pression plus élevée). Lors de ce processus, toute la chaleur absorbée dans l'évaporateur et le compresseur est cédée à l'environnement extérieur.

I.2.2.2-Evaporateur :

L'évaporateur peut être considéré comme point de capture de la chaleur ambiante dans un système de réfrigération. L'évaporateur a pour fonction de prélever la chaleur à son environnement et de la céder au fluide frigorigène. La température d'évaporation doit être située lors de ce processus, au-dessous de la température ambiante. Le flux de chaleur généré entre l'évaporateur et le milieu ambiant grâce à la différence de température, entraîne dans l'évaporateur une évaporation du fluide frigorigène.

I.2.3-Détendeur :

Il permet de détendre le fluide frigorigène et de l'amener de la pression de condensation à la pression d'évaporation. Ce dernier organe boucle ainsi le cycle du circuit frigorigène. Une partie du fluide frigorigène s'évapore lors de cette opération et prélève de la chaleur à la température d'évaporation. Il permet aussi de réguler l'écoulement du liquide réfrigérant à l'évaporateur.

I.2.4-Fluide frigorigène :

Un fluide frigorigène est un fluide qui présente des particularités physiques permettant de l'exploiter dans un cycle physique de compression/détente pour transférer des calories.

Les fluides frigorigènes sont choisis pour leurs températures de passage de l'état liquide à l'état gazeux, la quantité d'énergie nécessaire pour provoquer ce changement d'état gazeux, la quantité d'énergie nécessaire pour provoquer ce changement d'état et la différence de température provoquée par ce changement d'état.

On appelle souvent improprement les fluides frigorigènes sous le terme de fréon (marque déposée), comme on parle de Frigidaire pour un réfrigérateur. Les fluides

frigorigènes les plus employés actuellement sont les HFC (Hydro Fluoro Caradones).

I.3-Fonctionnement d'un système réfrigérant a compression de vapeur :

En pratique, les systèmes de réfrigération à compression de vapeur sont généralement les plus utilisés et chaque système utilise un compresseur. Dans un cycle de réfrigération de base à compression de vapeur quatre processus thermiques principaux ont lieu comme suit :

- Compression
- Condensation
- Détente
- Evaporation

Comme dessiné sur ce schéma, le compresseur met en circulation le réfrigérant. Il sort de l'orifice HP (haute pression) du compresseur à l'état gazeux et très chaud. Il traverse alors le coté HP en commençant par le condenseur.

Lors de son passage dans le condenseur, le réfrigérant perd beaucoup de calories et se condense. On observe donc un changement d'état. La tuyauterie située entre le condenseur et le détendeur s'appelle à juste titre : **ligne liquide**.

Le détendeur est donc alimenté en réfrigérant à l'état **liquide**. Celui-ci **crée une restriction** c'est-à-dire une chute de pression du réfrigérant pour alimenter le cote BP (basse pression). A la sortie du détendeur, on observe une chute de pression importante ainsi qu'une chute de température de réfrigérant. Le réfrigérant est alors en mélange c'est-à-dire 15% en gaz et 85% en liquide. (Ces valeurs de pourcentage sont données à titre d'exemple et à prendre avec précaution. l'important est de retenir qu'ici le gaz est un mélange).

Le fluide mélangé traverse l'évaporateur. Dans ce dernier, on observe à nouveau un changement d'état. En effet la partie liquide du fluide **rentre en ébullition** et absorbe les calories qui sont au voisinage de l'évaporateur.

Le fluide à l'état gazeux est alors aspiré par l'orifice BP du compresseur et le cycle se renouvelle.

Remarque1 : Du refoulement du compresseur au détendeur, la pression HP est identique (aux pertes de charges près). Seule la température varie.

Remarque2 : Du détendeur à l'aspiration du compresseur, la pression BP est identique (aux pertes de charges près). Seule la température varie.

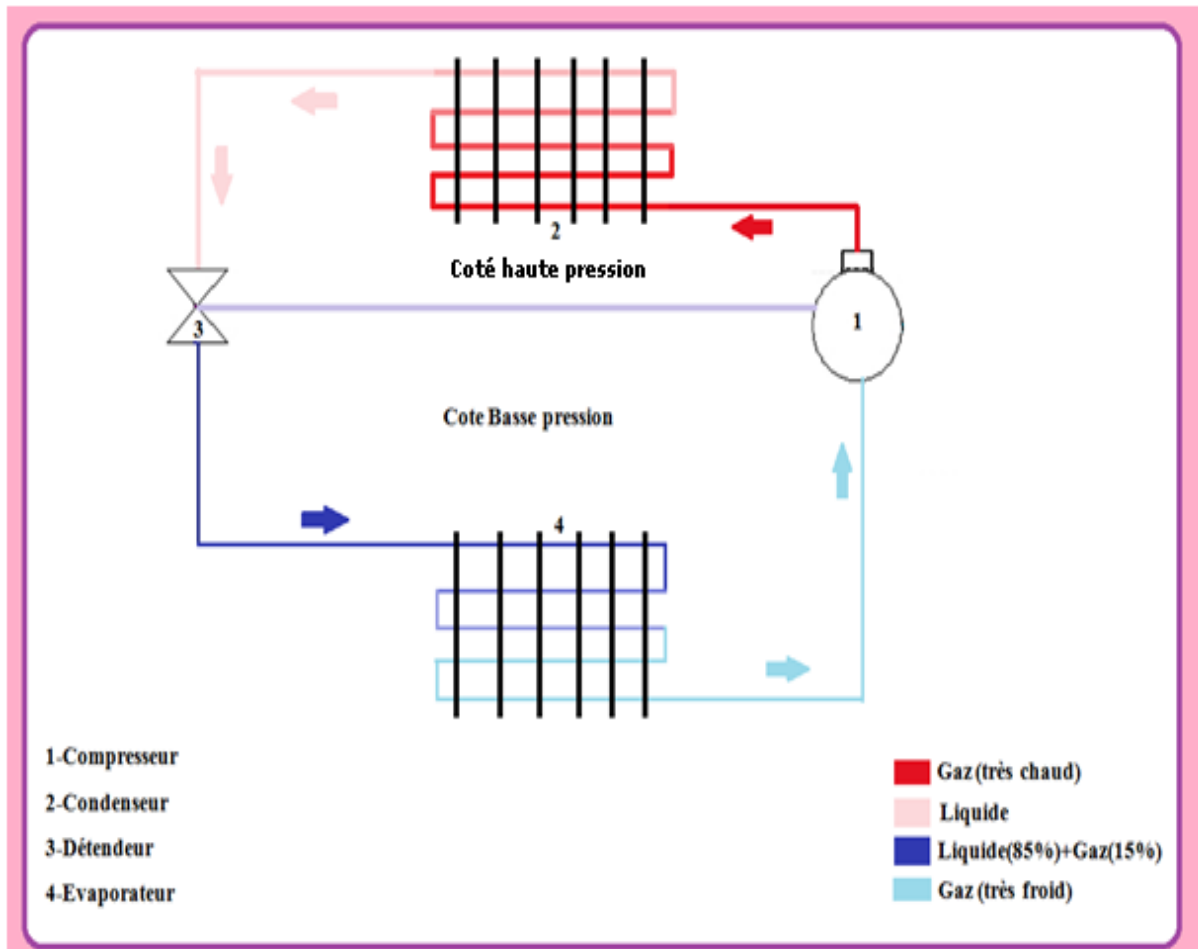


Figure1.1 : Schéma du fonctionnement d'un système de réfrigération.

I.4-Climatiseurs dit "MONO-SPLIT» :

Un Mono split est composé de 2 parties raccordées par des tubes de liaison frigorifique. L'évaporateur et son ventilateur sont installés dans le local à climatiser (unité intérieure). Le compresseur, le condenseur, le détendeur et les accessoires sont installés à l'extérieur du local à refroidir (unité extérieure). Les modèles de type split sont donc moins bruyants que les climatiseurs monoblocs.

On trouve des modèles de puissance frigorifique de 2 à 10 kW.

Les unités intérieures sont disponibles en plusieurs versions selon leurs emplacements.

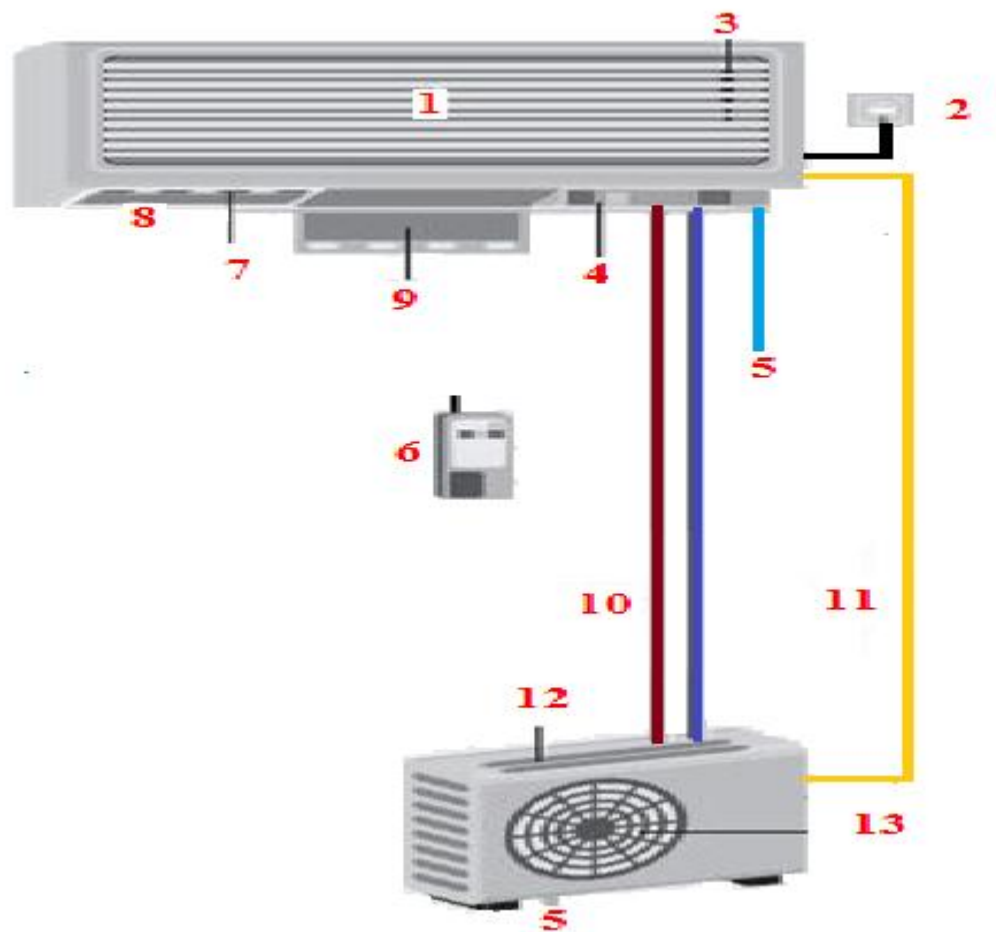


Figure1.2 : Schéma d'installation murale du climatiseur
"MONO-SPLIT» :

- | | |
|-------------------------------|---------------------------------------|
| 1- Air repris dans le local | 2- Alimentation électrique |
| 3- Sonde de température | 4- Commande unité intérieure |
| 5- Evacuation des condensats | 6- Télécommande |
| 7- Volet réglable | 8- Air soufflé dans le local |
| 9- Filtre à air | 10- Liaison du fluide frigorigène R22 |
| 11- Raccordement électrique | 12- Air repris à l'extérieur |
| 13- Air soufflé à l'extérieur | |

I.4.1-Emplacement de l'unité extérieure :

L'emplacement du condenseur jouera un rôle négatif sur l'aspect extérieur du bâtiment, mais jouera surtout sur l'efficacité énergétique du climatiseur.

La position du condenseur conditionnera la consommation énergétique du climatiseur, ainsi que sa fiabilité et sa longévité.

Le condenseur ne doit pas être placé dans une zone fortement ensoleillée. Plus la température de l'air extérieur est élevée, plus le condenseur aura des difficultés pour évacuer la chaleur.

Remarque : Il est recommandé de bien respecter les préconisations des constructeurs concernant les longueurs des tuyauteries entre l'unité extérieure et l'unité intérieure.

I.4.2-Composition du Climatiseur "SPLIT SYSTEME» :

Comme nous avons vu, le climatiseur **split-système** se compose de deux parties liées par des tubes de liaison pour fluide frigorigène

- Unité intérieure
- Unité extérieure

I.4.2.1-Unité extérieure :

Elle contient les composants suivants :

- 1- Compresseur
- 2- Echangeur thermique
- 3- Tube capillaire
- 4- Bouteille anti coups de liquide
- 5- Vannes à 4 voies
- 6- Un moteur-ventilateur + hélice
- 7- Deux tubes (un pour l'aspiration et l'autre pour le refoulement)
- 8- Un boîtier électrique

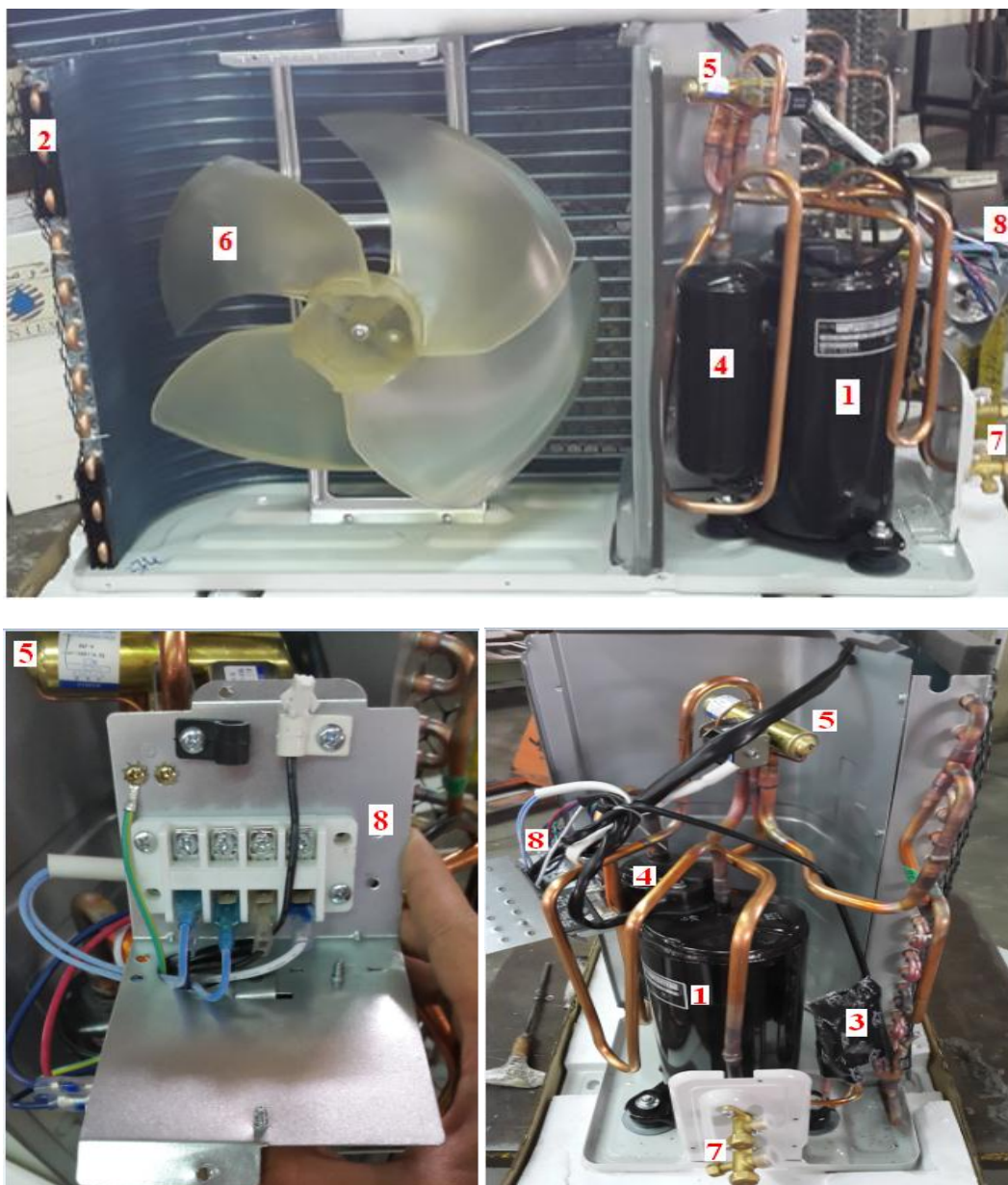


Figure 1.3 : Composons de l'unité extérieure du climatiseur Split-système

I.4.2.2-Unité intérieure :

Elle contient les composants suivants :

- 1) Turbine,
- 2) Filtre,
- 3) Echangeur thermique,
- 4) Bac à condensant,
- 5 Moteur électrique,
- 6) Boîtier électrique qui contient la carte électrique.

La turbine de l'unité intérieure installée dans une pièce peut être considérée comme un aspirateur.

L'air ambiant (A) est aspiré par l'effet de la turbine (1) qui tourne, ensuite il passe à travers des filtres (2). Il se rafraîchit ou se réchauffe suivant le mode de fonctionnement en passant à travers l'échangeur thermique (3). En mode froid la condensation produite par l'échangeur thermique coule dans le bac à condensant (4), l'air ainsi refroidi ou chauffé et pulsé par la turbine vers les ouvertures prévues (B).

Remarque : Il est primordial que les filtres soient nettoyés et désinfectés régulièrement.

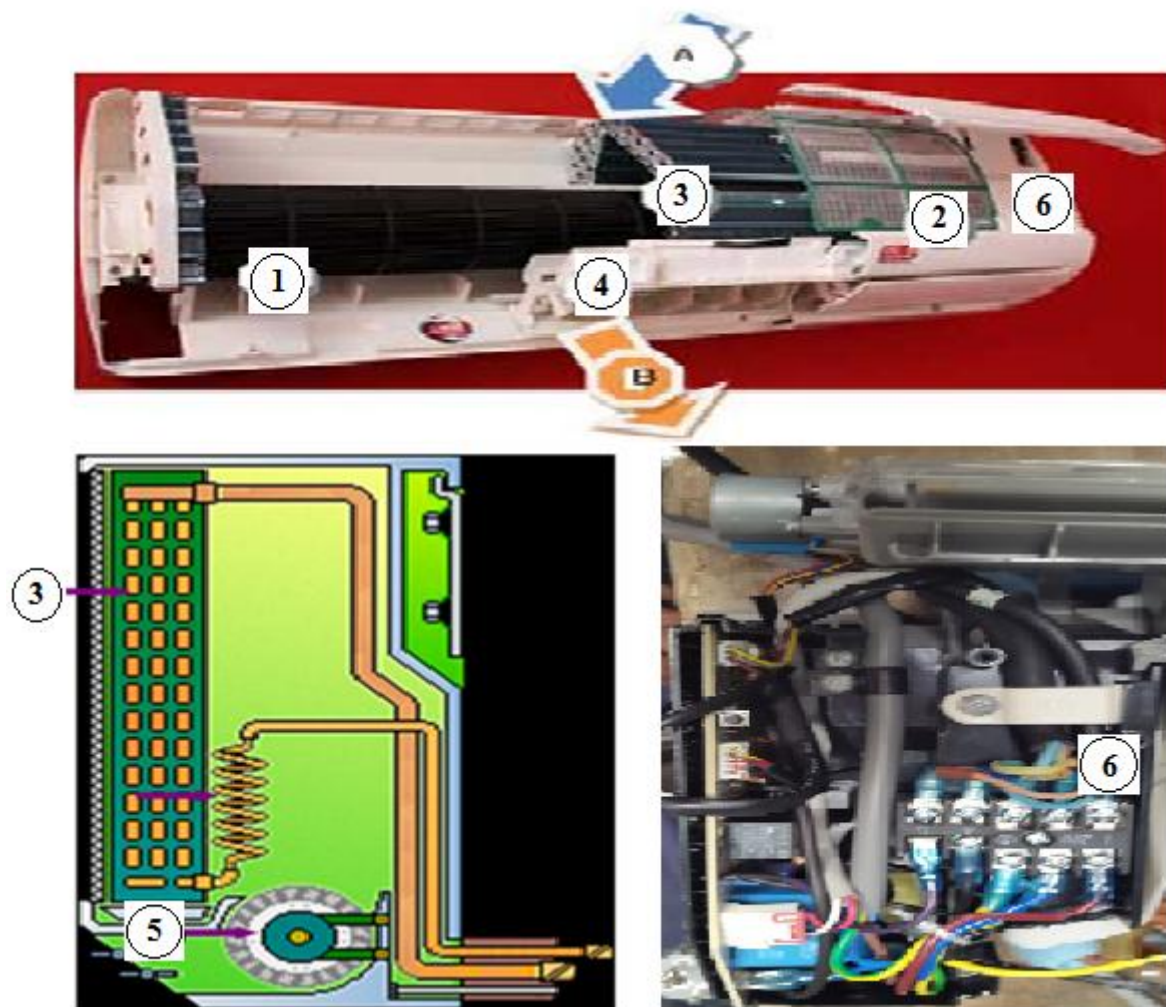


Figure1.3 : Composants de l'unité intérieure du climatiseur

Split-système :

- 1) Turbine 2) Filtre 3) Echangeur thermique, 4) Bac à condensant
 5) Moteur électrique 6) Boîtier électrique qui contient la carte électrique ;
 (A) Air aspiré (B) Air éjecté

I.4.3-Fiches techniques des différents composants d'un climatiseur :

I.4.3.1-Moto-ventilateur + hélice :

Model	Y55613B550
Voltage	208-240 V
Phase	50/60 HZ
Puissance	31 W
Ampérage	0.30 A
Nombre de pole	6 P
Classe	B
Capacité	2 UF/450V

Tableau 1.1 : Fiche technique du moteur-ventilateur

I.4.3.2-Turbine :

Model	RA12A
Voltage	208-240 V
Phase	50/60 HZ
Puissance	15 W
Ampérage	0.18 A
Nombre de pole	4 P
Classe	E
Capacité	1.5 UF/450 V

Tableau 1.2 : Fiche technique de la turbine

I.4.3.3-Compresseur :

Model	44R273A
Voltage	220/240 V
Phase	50 HZ
Puissance	1055 +/- 5% W
Ampérage démarrage	24.5 A +/- 10% max
Ampérage régime permanent	4.6 A
Nombre de pole	2
Configuration	K-SES
Capacité	35 uf/ 370 V

Tableau 1.3 : Fiche technique du compresseur

I.4.3.4-Fluide frigorigène :

Fiche Technique et Caractéristiques physiques :

Nom du Produit	Chlorodifluorométane
Formule chimique	CHF ₂ CL
Poids Moléculaire	86,84 g/mol
Point d'ébullition de liquide à 760 Torr	-40°C
Température critique	96°C
Pression critique	50 bars
Chaleur spécifique du liquide à 760 tours	0.426 K Cal/Kg
Densité du liquide à 20°C	1.213 Kg/L
Chaleur spécifique du gaz à 0°C	0.145 K cal/Kg.°C
Solubilité dans l'eau à 30°C	0.15 g/ 100g
Ininflammable Non explosif	1.19 (cp/cv)
Non corrosif	
Incolore	
Inodore	

Tableau 1.4 : Fiche technique et caractéristiques physiques du fluide frigorigène.

I.4.4-Inversion de cycle :

La climatisation réversible est une technique fiable et reconnue, avec un rendement important.

Dans un cycle frigorifique normal, l'écoulement du fluide frigorifique s'effectue toujours dans un même sens bien déterminé, à savoir : compresseur (refoulement) - condenseur (liquéfaction) - détendeur (détente) - évaporateur (vaporisation) - compresseur (aspiration).

Il est possible grâce à la vanne à quatre voies dont les climatiseurs "**MONO-SPLIT**", sont équipés, d'inverser une partie de ce circuit de telle sorte que le compresseur, au lieu de refouler dans le condenseur, puisse refouler le gaz surchauffé dans l'évaporateur (Ce dernier devenant ainsi condenseur) pour ensuite, après la liquéfaction des vapeurs, provoquer la détente (et la vaporisation) dans le condenseur (devenu évaporateur).

On s'aperçoit qu'il s'agit d'une inversion thermodynamique, non mécanique car le compresseur continue de tourner dans le même sens et de refouler et d'aspirer exactement dans les mêmes directions.

Remarque 1 : Il est important de signaler que plus la température extérieure est basse, plus le rendement du climatiseur réversible est faible, donc, fournit moins de chaleur.

Remarque 2 : Certains climatiseurs ont des résistances additionnelles dans les unités, permettant de fournir un chauffage complémentaire même par très basse température.

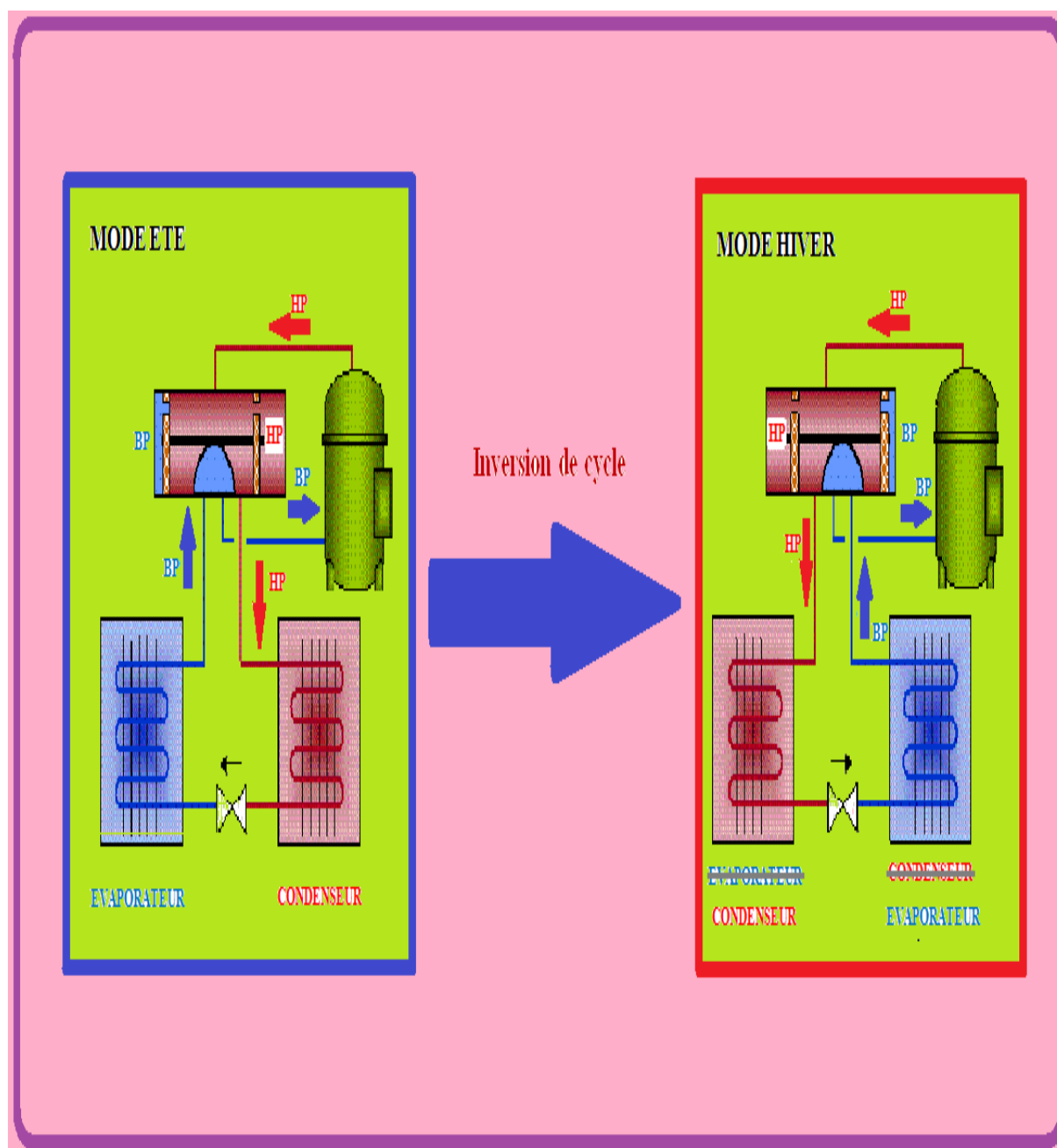


Figure1.4 : Inversion de cycle dans un climatiseur "MONO-SPLIT" :

I.4.4.1- Vanne quatre (4) voies :

Elle permet de réaliser l'inversion de cycle. La vanne possède une bobine et quand la bobine est à l'état repos c'est le mode froid, quand la bobine est excitée c'est le mode chaud.

Elle est uniquement utilisée sur les unités réversibles. Leur fonction est d'inverser la position de l'évaporateur et du condenseur.

Dans le cas de système réversible, le détendeur est de type capillaire car le fluide doit pouvoir circuler dans les deux sens.

La vanne quatre voies est de type à tiroir.



Figure1.5 : Vanne à quatre (4) voies

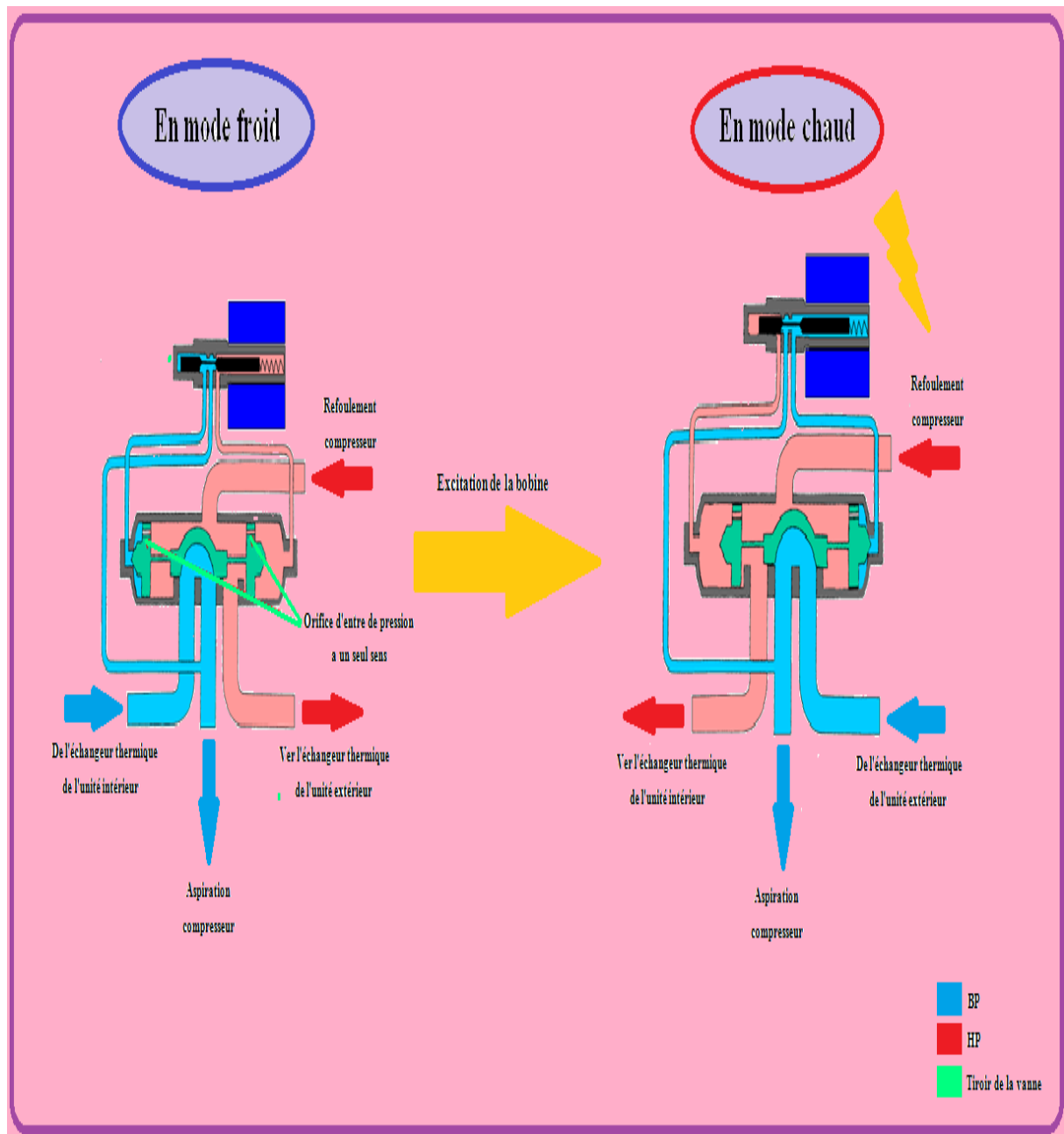


Figure1.6 : Schéma fonctionnel de la vanne à quatre voies du climatiseur "MONO-SPLIT".

I.4.4.2- Dégivrage :

La vanne à quatre voies a un deuxième rôle. L'inversion de cycle permet par ailleurs d'assurer le dégivrage de l'unité extérieure en hiver et l'unité intérieure en été.

En effet, pour des températures inférieures à 5°C, la vapeur d'eau se condense sur l'évaporateur qui peut alors "prendre de la glace". L'inversion momentanée du cycle frigorifique permet le dégivrage.

I.4-Contrôle électronique du climatiseur :

Le contrôle du climatiseur est assuré par une boîte électronique qui se situe dans l'unité intérieure. Elle contient la carte électronique qui gère le fonctionnement du climatiseur.

I.4.1-Schéma détaillé de la Carte électronique et ses composants :

- 1- Relais compresseur,
- 2- Relai électrovanne,
- 3- Condensateur,
- 4- Fusible,
- 5- Relais moteur ventilateur,
- 6- Microcontrôleur un qui control l'afficheur,
- 7- Microcontrôleur deux qui control le fonctionnement du système,
- 8- Premier fiche du moteur turbine,
- 9- Deuxième fiche du moteur turbine,
- 10- Fiche de la sonde de l'échangeur thermique de l'unité extérieur,
- 11- Fiche de la sonde d'ambiance,
- 12- Fiche du moteur pas à pas,
- 13- Fiche de l'afficheur,
- 14- Buzzer,
- 15- Transformateur

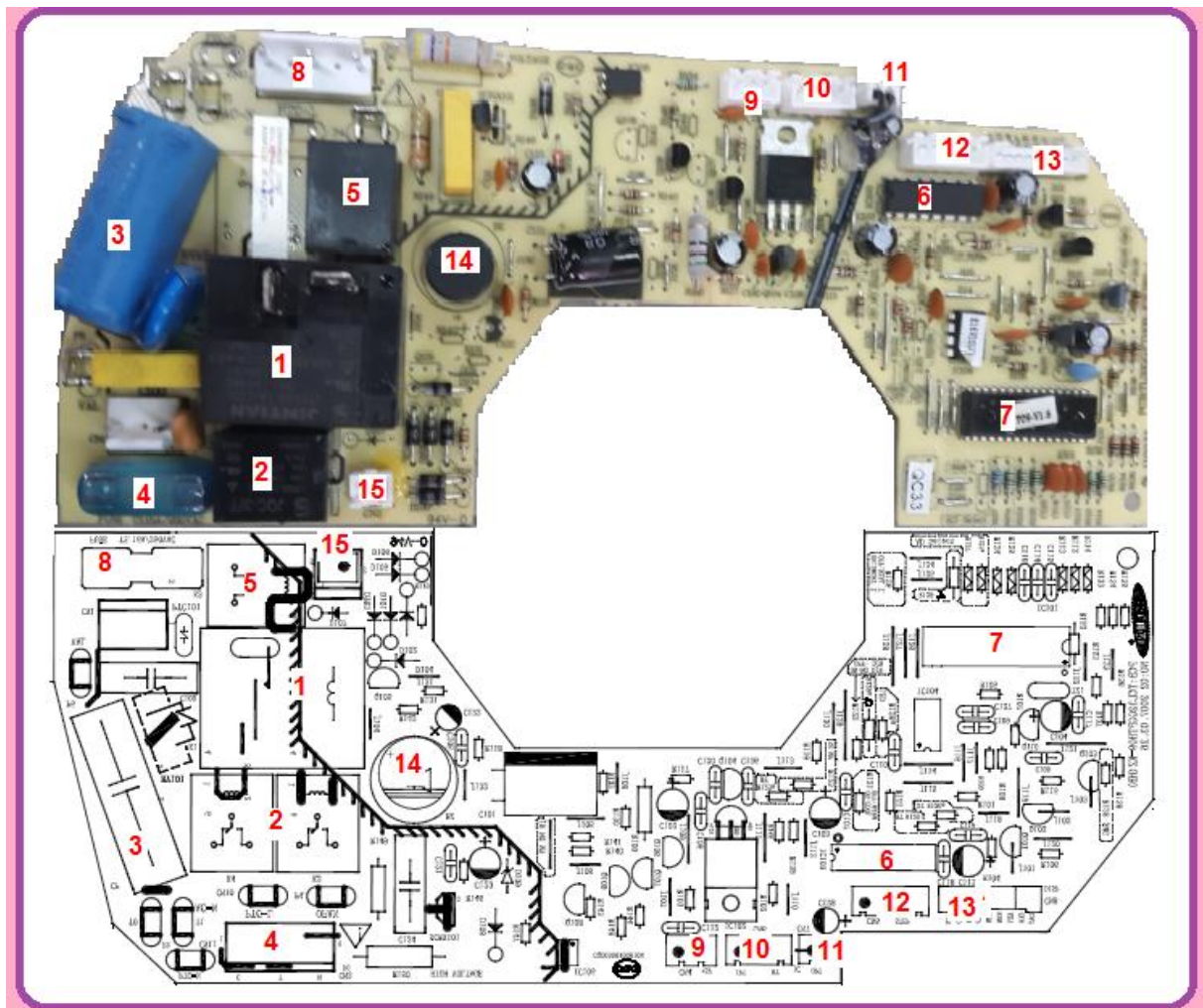


Figure1.7 : Schéma d'implantation des composants de la carte électronique du climatiseur "MONO-SPLIT" ENIEM 12000 BTU

I.4.2-Rôle de la carte électronique :

Elle a pour fonction d'assurer :

- Le bon fonctionnement général de la turbine, du moteur ventilateur, du compresseur, de l'électrovanne du climatiseur,
- Le fonctionnement du moteur déflecteur,
- Le contrôle de l'afficheur,
- Le basculement automatique du mode froid vers le mode chaud,
- La Protection antigivre des échangeurs thermiques par inversion de cycle automatique,
- La Protection anti-court circuit et anti surchauffe.

I.4.3-Télécommande à infrarouge :

La carte électronique est commandée par l'opérateur grâce à une télécommande à infrarouge qui permet le pilotage du climatiseur en agissant sur les paramètres suivants :

- Sélection du mode fonctionnement et des vitesses de soufflage,
- Programmation horaire et choix du mode fonctionnement,
- Fonction déshumidification,
- Témoins d'encrassement des filtres.

I.6-Fonctionnement du système à réaliser :

Pour assurer la bonne marche du climatiseur, la carte électronique que l'on veut concevoir doit assurer le bon fonctionnement du climatiseur dont ses différents modes de fonctionnement.

I.6.1-Mode ventilateur :

Dans le mode ventilateur il n'y a que la turbine de l'unité intérieure qui est actionnée, comme c'est illustré dans le tableau suivant :

Composant	Action
Compresseur	Off
Moteur ventilateur	Off
Electrovanne	Off
Moteur pas à pas	On
Turbine	On

Tableau1.5 : Fonctionnement de différent composent du climatiseur au Mode

ventilateur :

La turbine possède trois vitesses :

Vitesse	Pourcentage (%) de la vitesse de la turbine
1	33%
2	66%
3	100%

Tableau1.6 : vitesses en pourcentage de la turbine

I.6.2-Mode froid :

Dans le mode froid la turbine, le moteur ventilateur, le compresseur est actionné. Seule l'électrovanne n'est pas actionnée, comme cela est illustré dans le tableau suivant :

Composant	Action
Compresseur	On
Moteur ventilateur	On
Electrovanne	Off
Moteur pas à pas	On
Turbine	On

Tableau1.7 : Fonctionnement de différent composent du climatiseur au Mode Froid

I.6.3- Mode chaud :

Dans le mode chaud, la turbine, le moteur ventilateur, le compresseur sont actionnés et aussi l'électrovanne, comme c'est illustrée dans le tableau suivant :

Composant	Action
Compresseur	On
Moteur ventilateur	On
Electrovanne	On
Moteur pas à pas	On
Turbine	On

Tableau1.8 : Fonctionnement de différent composent du climatiseur au Mode chaud

Remarque : La turbine n'est actionnée qu'après 30s pour éviter un courant d'air froid à l'utilisateur.

I.6.4-Mode antigivrage :

Le mode antigivrage est une sécurité qui assure le dégivrage de l'évaporateur, quand il y a du givre qui se forme sur lui. Le givre diminue significativement le rendement du climatiseur et empêche un bon échange thermique avec le milieu extérieur qui fait que le gaz frigorigène ne s'évapore pas bien dans l'évaporateur.

Le givre se forme sur l'évaporateur car il est soumis à des températures très faibles de l'ordre de -40°C , c'est la température d'évaporation du gaz frigorigène.

Dans le mode froid les climatiseurs mon-split l'échangeur thermique de l'unité intérieure joue le rôle de l'évaporateur comme c'est expliqué précédemment. Quand le givre se forme, la carte électronique bascule le climatiseur sur **le mode chaud** pour dégager le givre qui s'est formé.

Dans le mode chaud l'échangeur thermique de l'unité extérieure joue le rôle de l'évaporateur comme c'est expliqué précédemment. Quand le givre se forme sur lui, la carte électronique bascule le climatiseur sur **le mode froid** pour dégivrer.

Remarque : Ce mode n'est pas contrôlé par l'utilisateur. C'est une sécurité automatique qu'assure la carte électronique du climatiseur pour assurer une bonne évaporation du gaz

frigorigène dans les échangeurs thermiques dans les différents modes.

I.7-Cahier de charge :

I.7.1-Mesure de température d'ambiance à l'aide d'un capteur thermique :

- Le capteur doit détecter la température réelle du milieu,
- Le capteur doit être capable de détecter des variations de température très faibles pour garder une température ambiante désirée optimale,
- Le capteur doit avoir une plage de variation comprise entre $[-20^{\circ}, 100^{\circ}]$,

I.7.2-Mesure de température des échangeurs thermiques :

- Le capteur doit détecter la température réelle à l'intérieur des échangeurs thermiques,
- Le capteur doit être capable de détecter des températures très faibles entre $[-20^{\circ}, -40^{\circ}]$ pour assurer le dégivrage des échangeurs thermiques.

I.7.3-Récepteurs de données numériques :

- Récepteur infrarouge pour un contrôle avec télécommande,
- Récepteur Bluetooth pour un contrôle vocal avec portable,
- Carte réseau pour un contrôle à distance avec portable.

I.7.4-Unité d'acquisition et de traitement :

- Centrale d'acquisition et de traitement à base d'un microcontrôleur,
- Développement d'un programme d'acquisition et de traitement de données des différents capteurs et cartes électroniques installées sur le climatiseur.

I.7.5-Unité de commande et interface graphique :

- Centrale de commande à base d'un microcontrôleur,
- Visualisation des données en temps réel,
- Développement d'un programme sur portable avec une interface graphique,
- Développement d'un programme pour contrôle des différents moteurs, compresseur et afficheur.

I.7.6-Du point de vue technique, le système doit être :

- Facile à installer, le logiciel comme le matériel,
- Partie électronique compacte et peu encombrante, en utilisant un microcontrôleur qui

assurera la gestion des capteurs, des moteurs et de l'afficheur,

- Fiable, avec une probabilité de fausse mesure infiniment petite,
- Cout relativement abordable par rapport aux systèmes disponibles.

I.8-Conclusion :

Pour répondre à ce cahier de charges, le système de contrôle du climatiseur sera divisé principalement en trois modules :

- 1- Le module d'acquisition et de traitement de donnée,
- 2- Le module de contrôle et d'affichage.
- 3- Le module serveur Web.

Avant de montrer les détails de notre réalisation, nous présentons dans le chapitre suivant les composants et les langages de programmation que l'on utilisera pour la conception de notre carte électronique.

Chapitre II

Choix des composants et
des langages de
programmations

II.1-Introduction :

Après que nous avons donné la description de la machine et du système à contrôler et rédiger le cahier de charges, il y a lieu d'étudier la structure du système pour pouvoir traiter, visualiser et contrôler les différents moteurs, cartes électroniques et afficheurs. Dans ce chapitre nous allons présenter la conception du système de contrôle et les critères de choix de chaque composant.

II.2-Structure du système :

Le système de contrôle que nous avons conçu est constitué de deux modules essentiels. Chacun accomplit une ou plusieurs fonctions qui lui sont propres (figure).

Le premier est le module d'acquisition de données et de traitement, le second est le module de commande et d'affichage. On commence par le premier module qui est réalisé à base d'une carte **Arduino-nano**, du moment que c'est lui qui s'occupe en premier lieu de l'acquisition et du traitement données des différents capteurs et composants électroniques, puis on passe au second module.



Figure 2.1 : Schéma bloc du système de contrôle

Module d'acquisition et traitement de données :

Il a une partie software à base de logiciel IDE Arduino et d'une partie hardware qui est à base d'une carte Arduino-nano. Il prend en charge la réception et l'envoi des différentes instructions de fonctionnements aux modules de commande et d'affichage à partir d'une commande infrarouge et Bluetooth. Il reçoit aussi les données des capteurs thermiques.

Module de commande et d'affichage :

Il a aussi une partie software à base de logiciel IDE Arduino et d'une partie hardware qui est à base d'une carte Arduino-nano. Il assure la commande des moteurs, afficheur et buzzer. Il a le rôle aussi de communiquer avec les deux autres modules pour recevoir les instructions de fonctionnement, les deux autres modules sont connectés à lui.

Module serveur WEB :

Il a aussi une partie software à base de logiciel IDE Arduino et d'une partie hardware qui est à base d'une carte Arduino-nano. Il prend en charge la réception des différentes instructions de fonctionnements aux modules de commande et d'affichage à partir d'une commande réseau internet.

II.3-Module d'acquisition et de traitement de données :

Pour la réalisation de cette partie du projet, nous avons opté pour les composants suivants :

- Une carte Arduino-nano.
- Capteur thermique LM35.
- Récepteur infrarouge.
- Module Bluetooth HC-05.

II.3.1-Présentation de la carte Arduino Nano :

Le modèle Arduino Nano est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega328. Le microcontrôleur de la famille AVR dont la programmation peut être réalisé en langage C Arduino. Cette carte possède 14 entrées/sorties numériques (dont 6 peuvent être utilisées comme étant des PWM (pulse Width modulation), 6 entrées analogiques avec un convertisseur Analogique/numérique de 10 bits de résolution, 1 résonateur céramique (quartz) de 16 Mhz. 1 connecteur ICSP (in-circuit serial

Chapitre II

programing) qui permet la programmation du microcontrôleur sur le circuit sans savoir à l'enlever, 1 connecteur jack pour une alimentation extérieure, un bouton de reset pour mettre le processus a zéro.

L'avantage de cette carte c'est qu'elle n'a pas besoin de pilote pour faire la conversion FTID USB/série. Elle a juste un petit microcontrôleur ATMega programmé comme convertisseur USB/série.

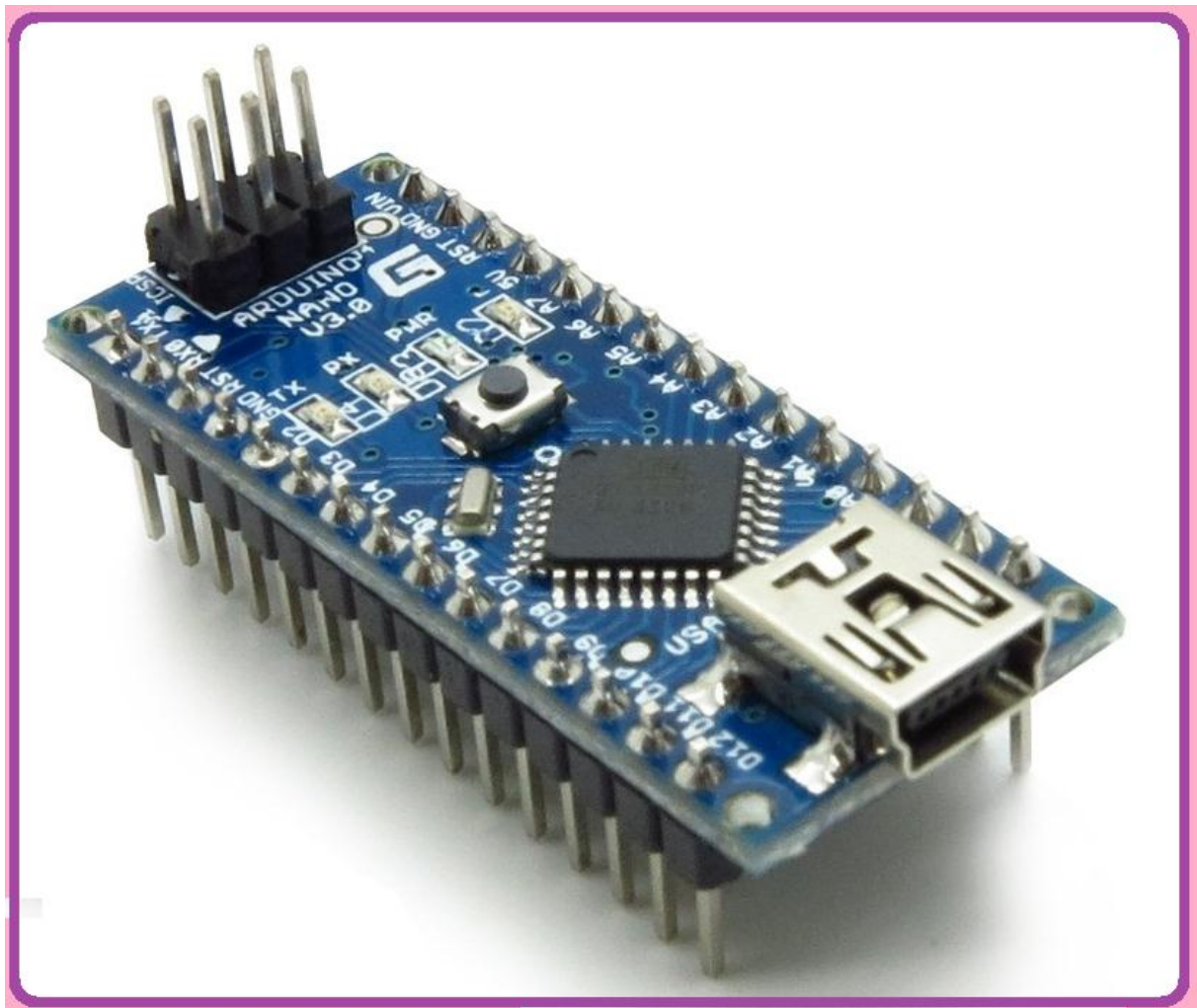


Figure2.2 : Vue de la carte Arduino nano

II.3.2-Présentation du capteur thermique LM35 :

Le capteur de température LM35 est un capteur de température analogique fabriqué par **Texas Instruments**. Il est extrêmement populaire en électronique, car précis, peu coûteux, très simple d'utilisation et d'une fiabilité à toute épreuve. Il est capable de mesurer des températures allant de -55°C à $+150^{\circ}\text{C}$ dans sa version la plus précise et avec le montage adéquat, de quoi mesurer n'importe quelle température.

N.B : Les versions grand public sont capables de mesurer des températures comprises entre -40°C à 110°C .

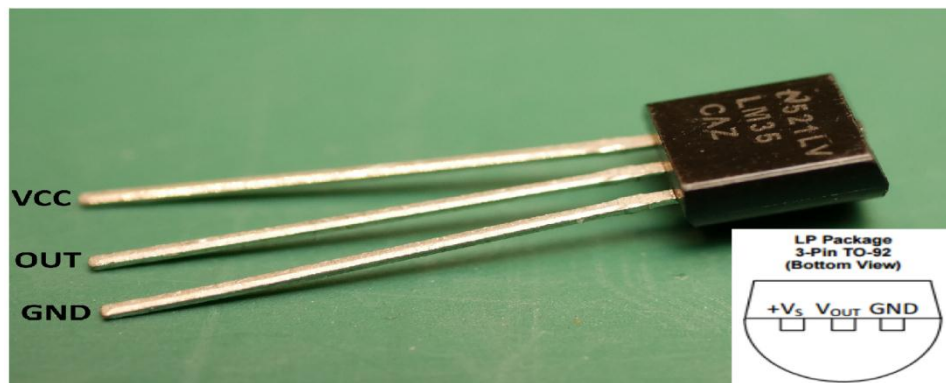


Figure2.3 : Vue du Capteur LM35

II.3.3-Récepteur infrarouge :

Le récepteur **IR TSOP** délivre un signal constant lorsqu'il est actif et qu'il reçoit les données.

Les signaux de commande à distance sont souvent des octets de données qui sont codés et transmis par impulsions (commutation ON et OFF **LED IR** à une fréquence spécifique). La plupart des télécommandes de télévision fonctionnent à des fréquences de 32-40 kHz et la plupart des récepteurs peuvent recevoir cette gamme.

Le protocole **SIRC** utilise un codage de largeur d'impulsion de bits. L'impulsion représentant un état logique "1" est une salve de 1.2ms de long de la porteuse de 40 kHz, tandis que la largeur de salve à un état logique "0" est 0.6ms de long, toutes les salves sont séparés par un intervalle de temps de 0.6ms espace.



Figure 2.4 : Vue du Récepteur infrarouge IR TSOP

II.3.4 Bluetooth HC-05 :

Le module HC-05 n'est pas plus gros qu'un pouce. Il est en fait un montage d'un module **Bluetooth** sur un petit **P.C.B**, cela permet de s'affranchir de certaines contraintes comme la soudure du module (qui est très délicate), la conversion 5V -> 3.3V, la régulation de l'alimentation (3.3V de nouveau) ou encore l'ajout de **LED** de signal, tout cela est déjà intégré.



Figure2.5 : Vue du Module Bluetooth HC-05

Chapitre II

Le **HC-05** agit comme un port série à travers lequel vous pouvez envoyer et recevoir des données. Donc en utilisant un terminal série ou une application Bluetooth personnalisée sur votre ordinateur ou téléphone, vous pouvez contrôler et surveiller votre projet.

Nous trouvons sur ce petit module, un ensemble de broches, VCC et GND pour l'alimentation **5V**, **Rx/Tx** pour la communication, on y voit aussi une broche "Key" qui servira à envoyer des commandes de configuration au module, la dernière broche nommée "Led" permet de brancher une LED pour obtenir un signal sur l'état du module.

II.4-Module de commande et d'affichage :

Pour la réalisation de cette partie du projet, nous avons opté pour les composants suivants :

- Afficheur LCD 4*20
- Carte relais huit canaux
- Buzzer

II.4.1- afficheur LCD 4*20 (JHD 204A)

Afficheur LCD 4*20 (**JHD 204A**) : c'est un afficheur à écran Christo-liquide alpha numérique qui a 4 quatre lignes de 20 caractères.



Figure 2.6 : Afficheur LCD 4*20 (JHD 204A)

II.4.2- Carte relais huit canaux :

Une **Carte relais** est un organe électrique permettant de dissocier la partie puissance de la partie commande. Elle permet l'ouverture et la fermeture d'un circuit électrique par un second circuit complètement isolé.



Figure 2.7 : Relais huit canaux

II.4.3- Buzzer piézo-électrique :

C'est un composant électronique qui émet un signal sonore d'une fréquence comprise entre 20 Hz et 20 kHz.



Figure 2.8 : Vue Buzzer piézo-électrique

II.5-Module serveur WEB :

Pour la réalisation de cette partie du projet, nous avons opté pour les composants suivants :

- Une carte arduino-nano
- Une Carte réseau-arduino

II.5.1-Carte réseau arduino :

Elle permet de connecter la carte Arduino à Ethernet. Elle permet d'envoyer et de recevoir des données de partout dans le monde grâce à une connexion internet. Nous pouvons l'utiliser pour contrôler des moteurs et même des robots à partir d'un serveur Web.

Cette carte nous ouvre une quantité infinie de possibilité en nous permettant de connecter notre projet à l'internet en un temps court.



Figure 2.9 : Carte réseau arduino

Le shield Arduino Ethernet est un module Arduino additionnel aux cartes Uno, Duemilanove ou Mega qui permet de les rendre communicantes sur un réseau filaire Ethernet.

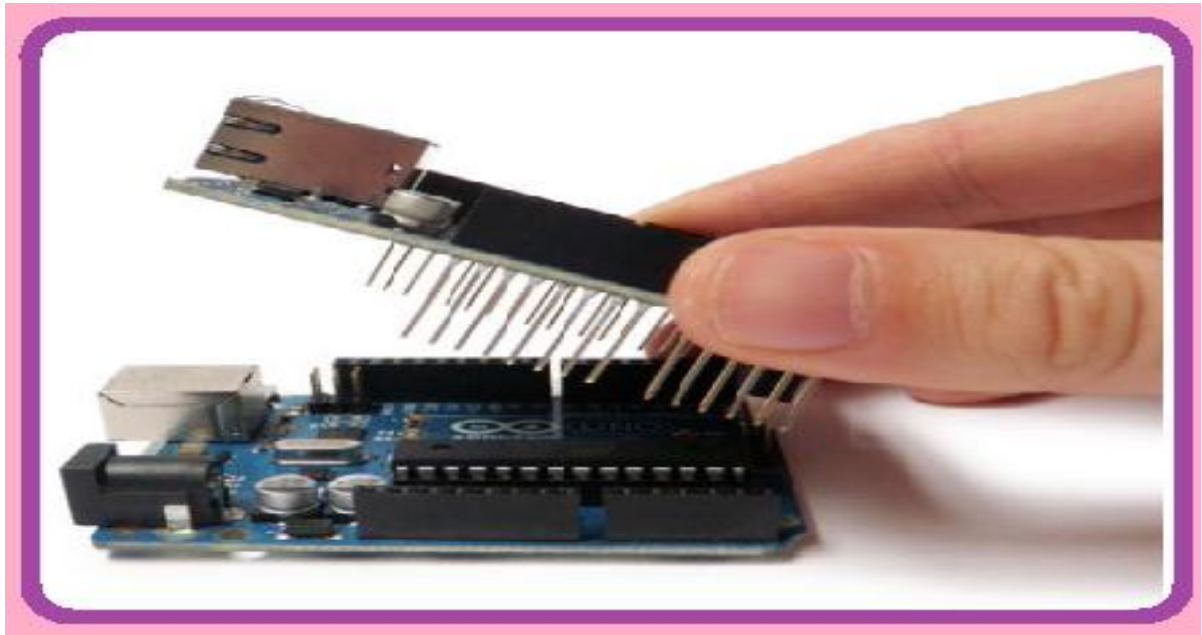


Figure 2.10 : Shield Arduino Ethernet additionne a la carte arduino UNO

On pourra ainsi créer une interface Homme Machine pour piloter à distance ou visualiser l'état de notre carte **arduino** en utilisant un câble réseau relié à un box internet.

➤ **Caractéristique de la carte Rj45 :**

La carte contient un certain nombre de LEDs d'information :

PWR : s'allume lorsque l'alimentation de la carte est présente,

LINK : s'allume lorsqu'un réseau est connecté et clignote lors de l'échange de données.

FULLD : s'allume lorsque la connexion réseau est en full duplex,

100M : s'allume lorsque la connexion est en 100Mbps/s

RX : s'allume lors d'une réception de données,

TX : s'allume lors d'une transmission de donnée,

COLL : s'allume lors d'une collision sur le réseau.

II.6-Choix de langage de programmation :

- C Arduino
- App Inventor
- HTML et CSS

II.6.1-C Arduino

Comme cela a été dit, la carte Arduino présente le noyau de notre système. Elle est programmée par un logiciel compatible appelé **Arduino EDI** (Espace de développement intégré) qui est une application écrite en Java inspirée du langage Processing.

L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensible pour la carte.

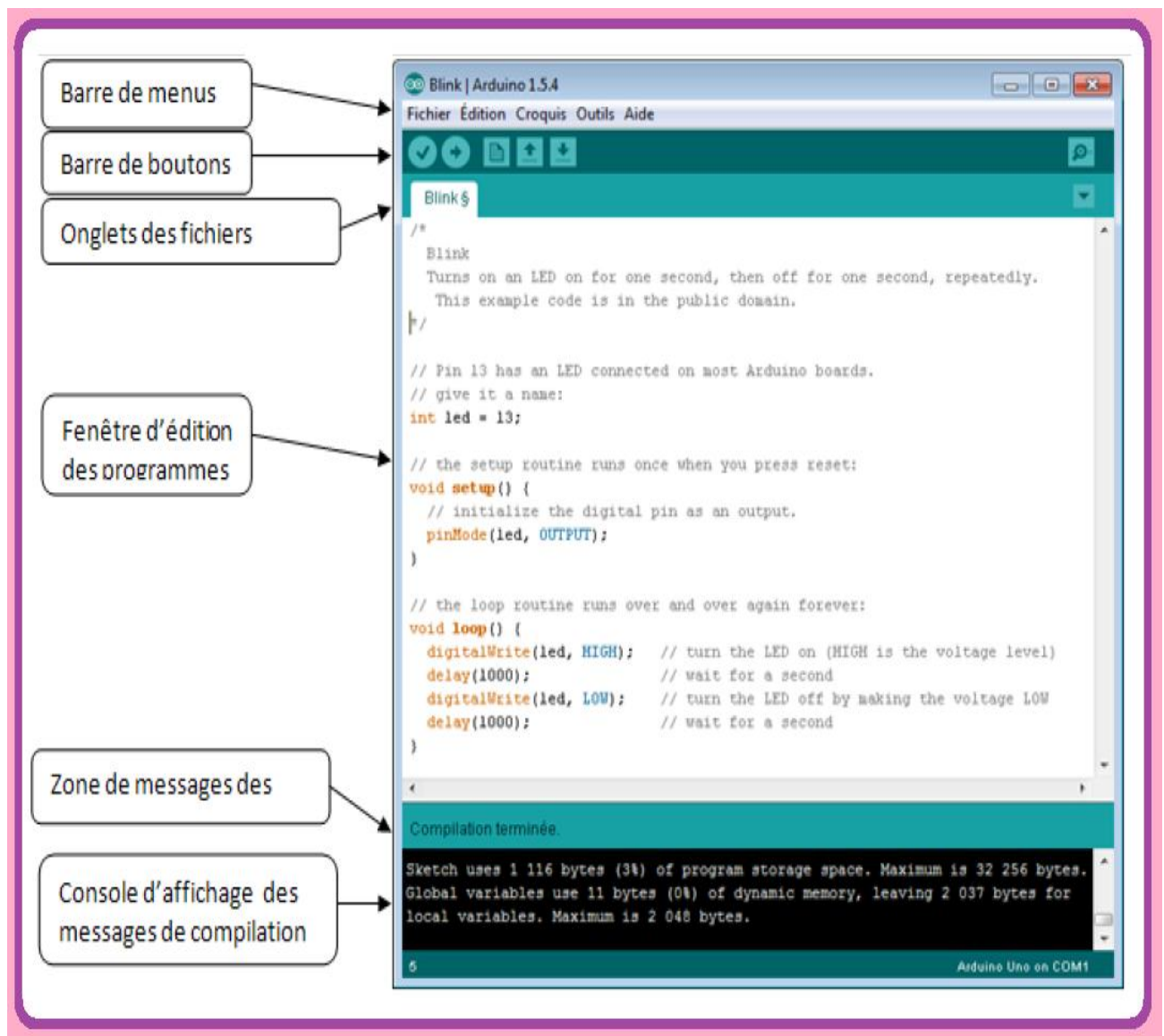


Figure 2.11 : Les différentes parties de la fenêtre principale de logiciel Arduino

II.6.2-App Inventor :

II.6.2.1-Présentation d'Androïde :

Androïde est un système d'exploitation libre qui utilise les téléphones mobiles modernes nommés (Smartphones). Il est basé sur linux, Androïde est détenu par Google. Sur le marché de la téléphonie mobile Androïde représente 78% des systèmes d'exploitation des téléphones mobiles.

L'Androïde vous pouvez facilement programmer et effectuer des programmes de formules mathématiques, contrôle de moteurs à distance, des jeux, emplacement sur Google Maps, le déplacement d'un robot via Bluetooth, réponse automatique SMS...

II.6.2.2-Présentation App inventeur 2 :

Est une application développée par Google. Elle est actuellement entretenue par le Massachusetts Institute of Technology (MIT).

Elle simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont plus familiers avec les langages de programmation. Elle est basée sur une interface graphique similaire, grâce à son interface entièrement graphique et à l'absence totale de ligne de code. Elle est particulièrement adaptée à l'initiation des enfants à la programmation, et ce des écoles primaires.

La programmation est réalisée sans taper une seule ligne de code, mais simplement en associant et en paramétrant des briques logicielles réalisées.

Google publie l'application le 15 décembre 2010. C'est le centre d'études mobiles au MIT qui gère le support technique de cette application sous le nouveau nom "MIT app Inventor".

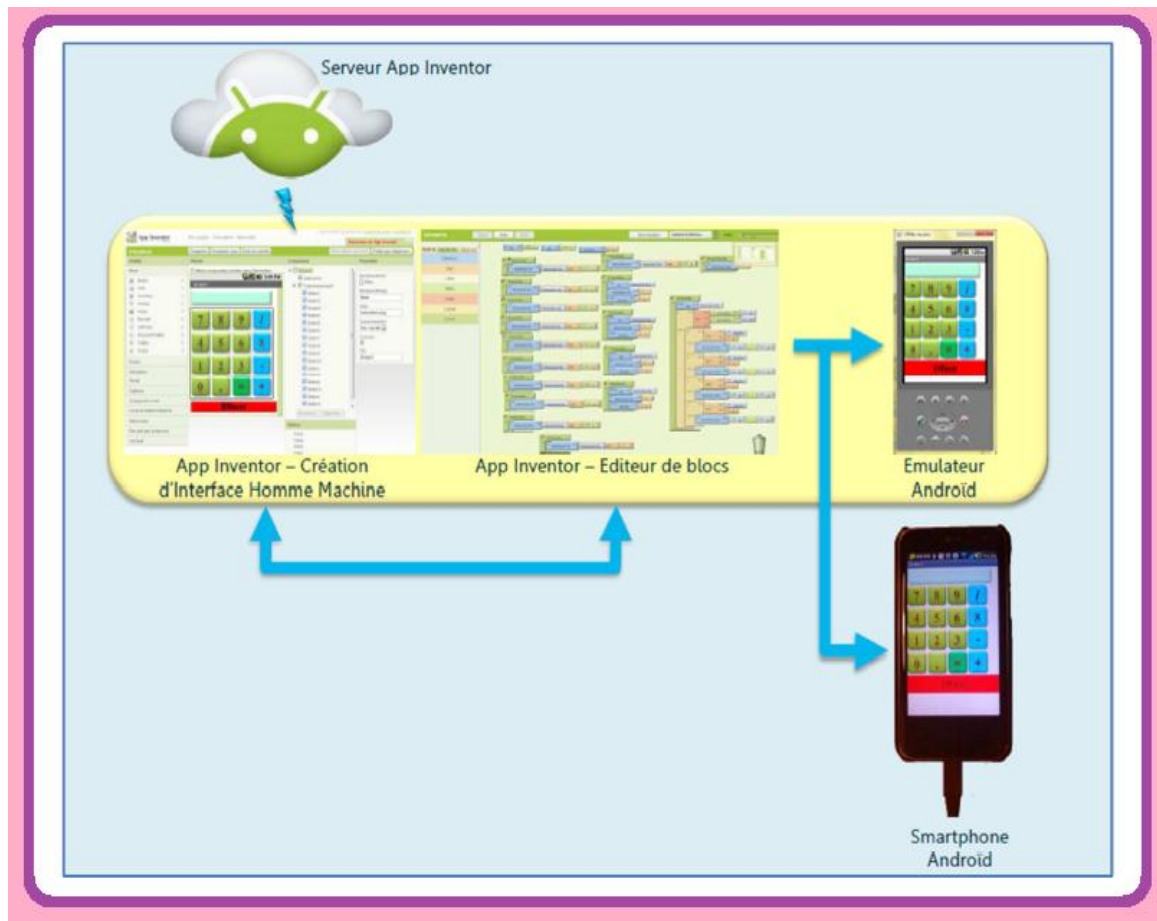


Figure 2.12 : Concept d'App Inventer 2

II.6.3-Programme HTML et CSS :

Il s'agit de langages informatiques qui permettent de créer des sites web. Tous les sites web sont basés sur ce genre de langages. Ils sont incontournables et universels aujourd'hui, ils sont à la base même du web.

Le langage HTML et CSS est à la base du fonctionnement de tous les sites web. Quand vous consultez un site avec votre navigateur, il faut savoir que, en coulisse, des rouages s'activent pour permettre au site web de s'afficher, l'ordinateur se base sur ce qu'on lui a expliqué en HTML et CSS pour savoir ce qu'il doit afficher, comme le montre la figure suivante.



Figure 2.13 : Défiance entre le langage HTML, CSS et le Résultat a l'écran

Pour créer un site web, on doit donner des instructions à l'ordinateur. Il ne suffit pas simplement de taper le texte qui devra figurer dans le site (comme on le ferait dans un traitement de texte Word par exemple), il faut aussi indiquer où placer ce texte, insérer les images, faire des liens entre les pages, etc.

II.7-Conclusion :

Dans ce chapitre, nous avons choisi et passé en revue l'architecture matérielle et logicielle de notre système de contrôle du climatiseur ainsi que les principales caractéristiques des composants électroniques et langages de programmations qui constitue notre système. De même, nous avons présenté leurs configurations externes et internes et la manière avec laquelle, chacun d'eux, va être connecté avec les autres.

Chapitre III

Réalisation du système de
contrôle

III.1-Introduction :

Dans ce chapitre on définira les composants électroniques et les langages de programmations que l'on utilisera pour concevoir notre carte électronique. La réalisation pratique et les tests du système font aussi l'objet de ce chapitre.

III.2-Réalisation :

III.2.1-Réalisation matérielle :

Chaque partie est réalisée et testée séparément. Les montages sont d'abord construits sur des "breadboard" ou lab d'essai. Après les avoir expérimentés et adoptés séparément, nous les avons regroupés et réalisés sur un circuit imprimé qui s'emboîte parfaitement.

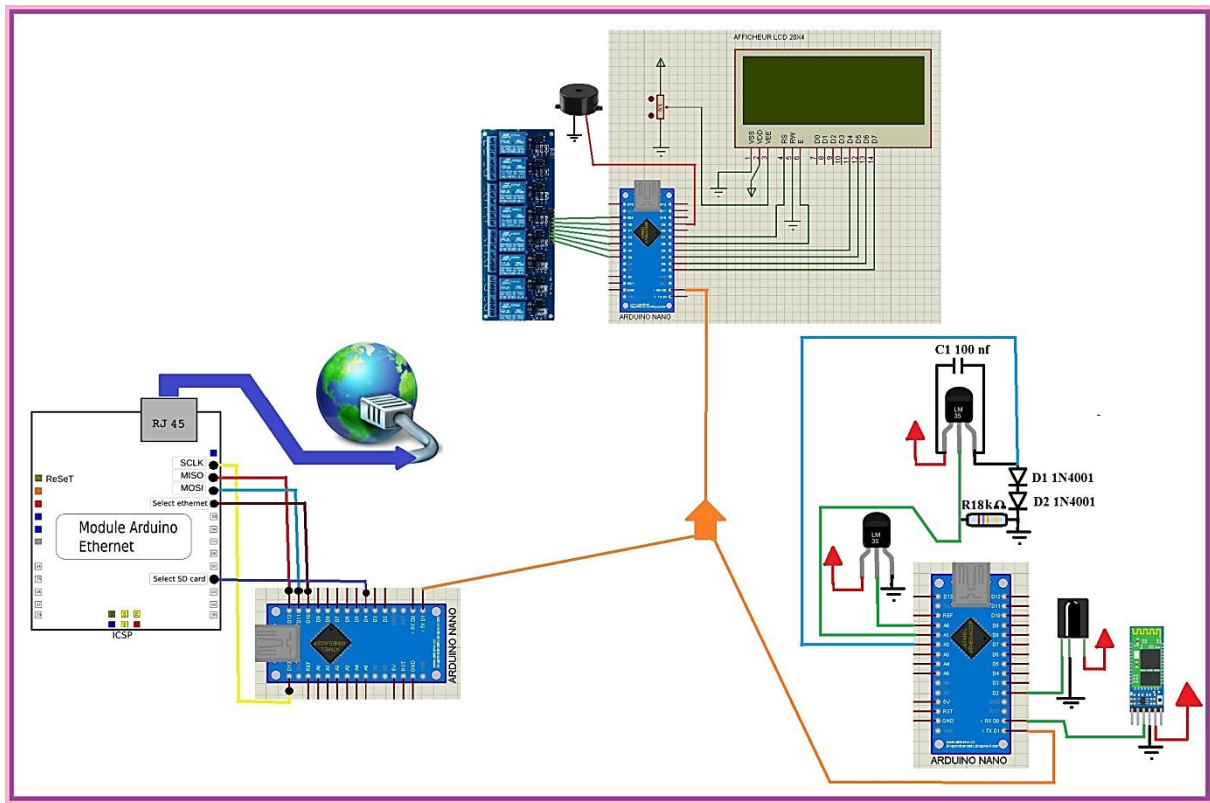
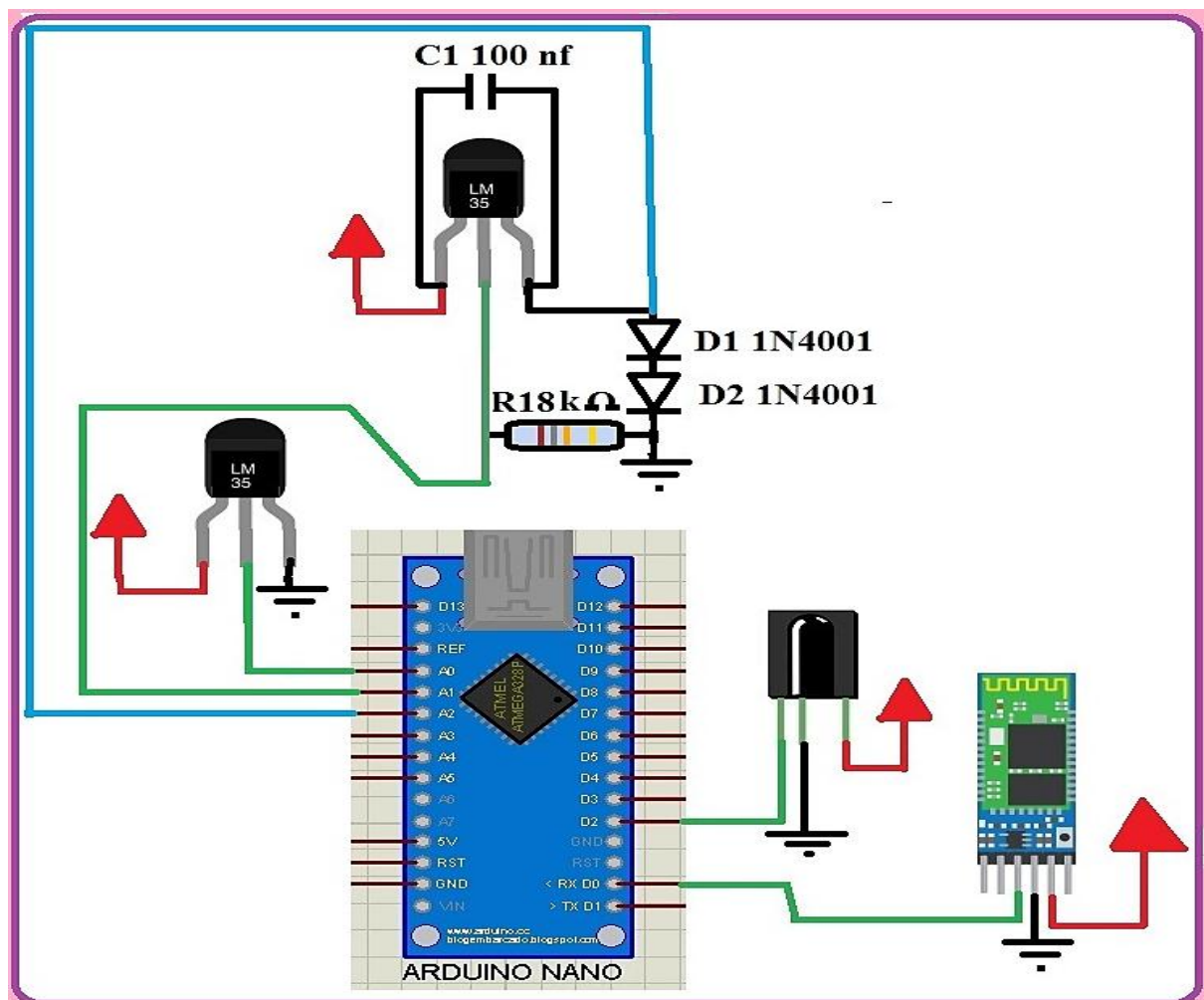


Figure 3.1 : Schéma de conception du circuit

Le circuit électronique qui englobe le système de contrôle est divisé en trois parties principales :

Partie une se compose :

- une carte Arduino-Nano
- un récepteur infrarouge



Chapitre III

III.2.1.1.1-Carte Arduino-Nno :

La carte Arduino-Nano dont elle est composée la première partie de la carte, reçoit et traite les données du récepteur infrarouge, module Bluetooth et des capteurs LM35 afin de commander la carte Arduino-Nano de la deuxième partie qui contrôle l'afficheur et qui actionne la carte relais grâce à une communication série (RX, TX) entre les deux cartes Arduino-Nano.

La carte est à base de microcontrôleur Atmega 328 cadencé par un oscillateur externe de fréquence de 16 Mhz.

- Les entrées/sorties numériques ont été configurées,
- La broche D2 est réservée au récepteur infrarouge,
- La broche A0 est réservée au capteur de température ambiante,
- Les broches A1 et A2 sans réservées à la sonde thermique,
- LA broche D0(Rx) est réservée à la communication série au module Bluetooth HC-05.

III.2.1.1.2-Récepteur infrarouge :

Permet un contrôle classique du climatiseur grâce à une télécommande ENIEM que l'on a pu adapter à notre carte électronique.

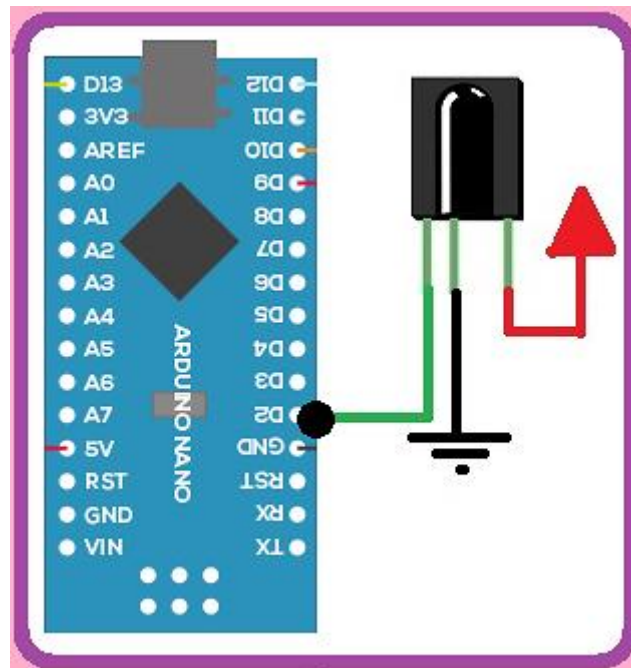


Figure 3.3 : Schéma de brochage récepteur IR

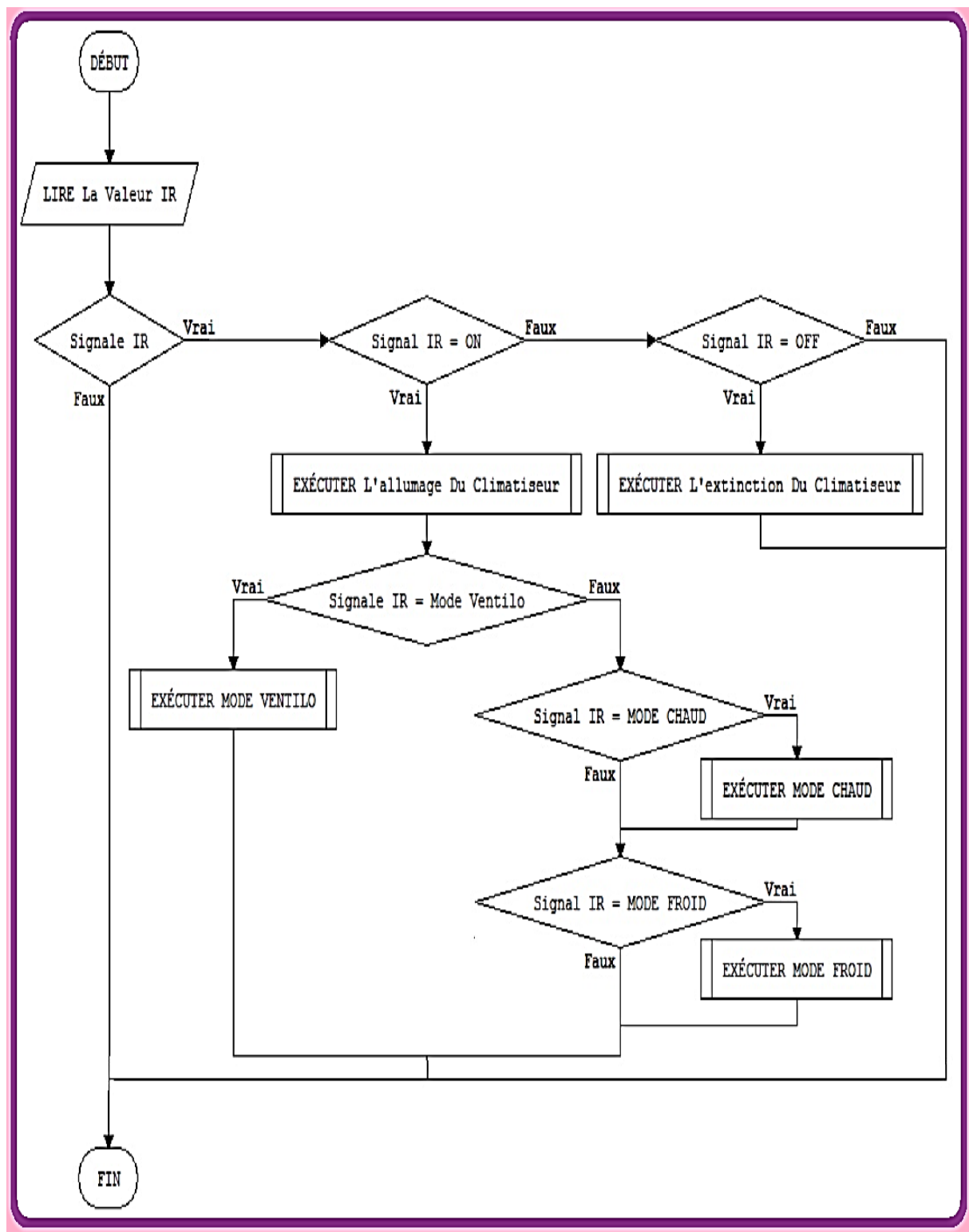


Figure 3.4 : organigramme du fonctionnement du récepteur infrarouge

III.2.1.1.3-Module Bluetooth HC05 :

Ce module est à faible consommation d'énergie et facile à mettre en œuvre. Il va nous assurer une communication unidirectionnelle avec la carte Arduino-nano et sans fil grâce à une application installée sur téléphone portable **Androïde** qui va nous assurer deux modes de contrôles :

- Commande manuelle
- Commande vocale

Le Bluetooth HC 05 est configuré comme maître sur le module de traitement et d'acquisition.

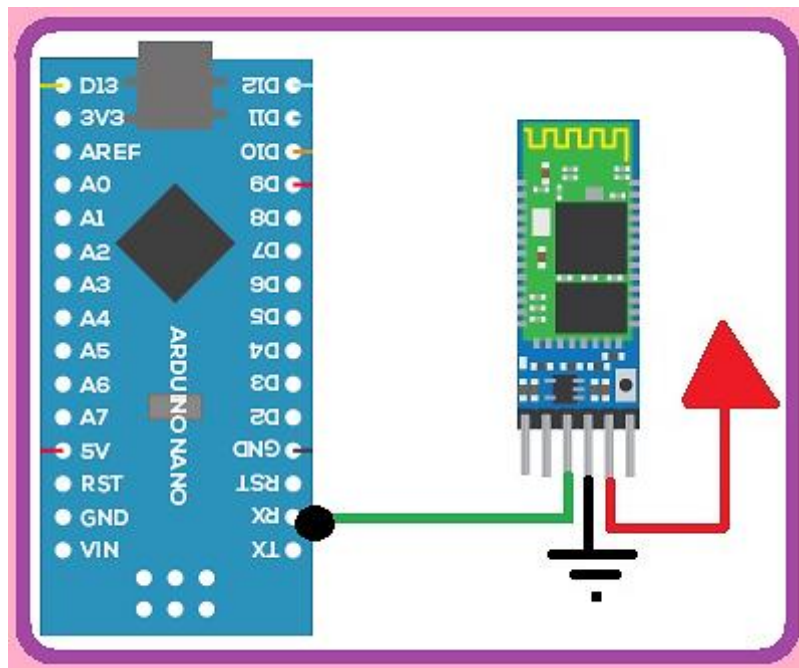


Figure 3.5 : Schéma de brochage du HC-05

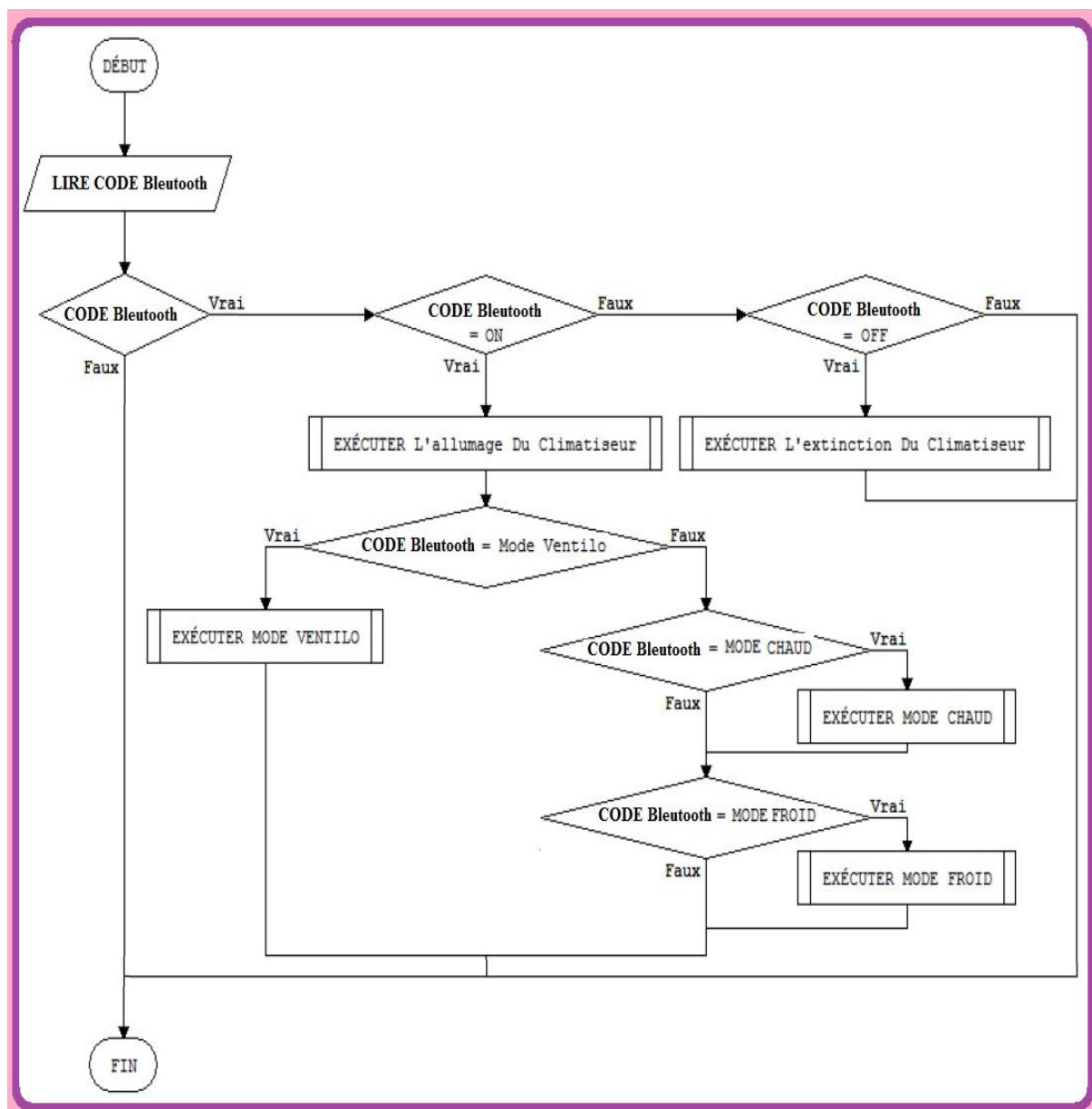


Figure 3.6 : Organigramme du fonctionnement du Module Bluetooth HC-05.

III.2.1.1.4-Capteur LM35 :

Le premier LM35 est utilisé comme capteur de température ambiante et affiché sur l'afficheur LCD 4*20.

Dans le mode froid, si cette température captée par LM35 est plus élevée que la consigne donnée par l'utilisateur, le climatiseur démarre et s'arrête quand la température d'ambiance affiche 1°C de moins que la consigne donnée. Une fois arrêté, il redémarre quand le capteur détecte une température plus élevée de 1°C par rapport à la consigne.

Dans le mode chaud si cette température captée par LM35 est moins élevée que la consigne donnée par l'utilisateur, le climatiseur démarre et s'arrête quand la température d'ambiance affiche 1°C de plus que la consigne donnée. Une fois arrêté il redémarre quand le capteur détecte une température moins élevée de 1°C par rapport à la consigne.

Remarque : Le démarrage et l'arrêt du climatiseur est conditionné par une régulation TOR (tout ou rien) +/-1°C par rapport à la consigne.

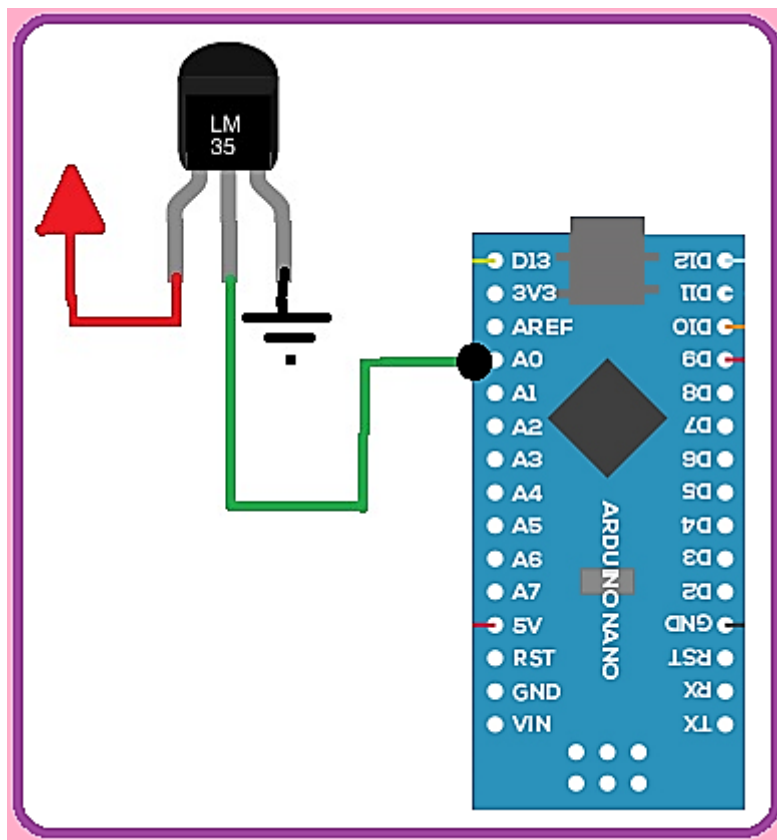


Figure 3.7 : Schéma de brochage du LM35

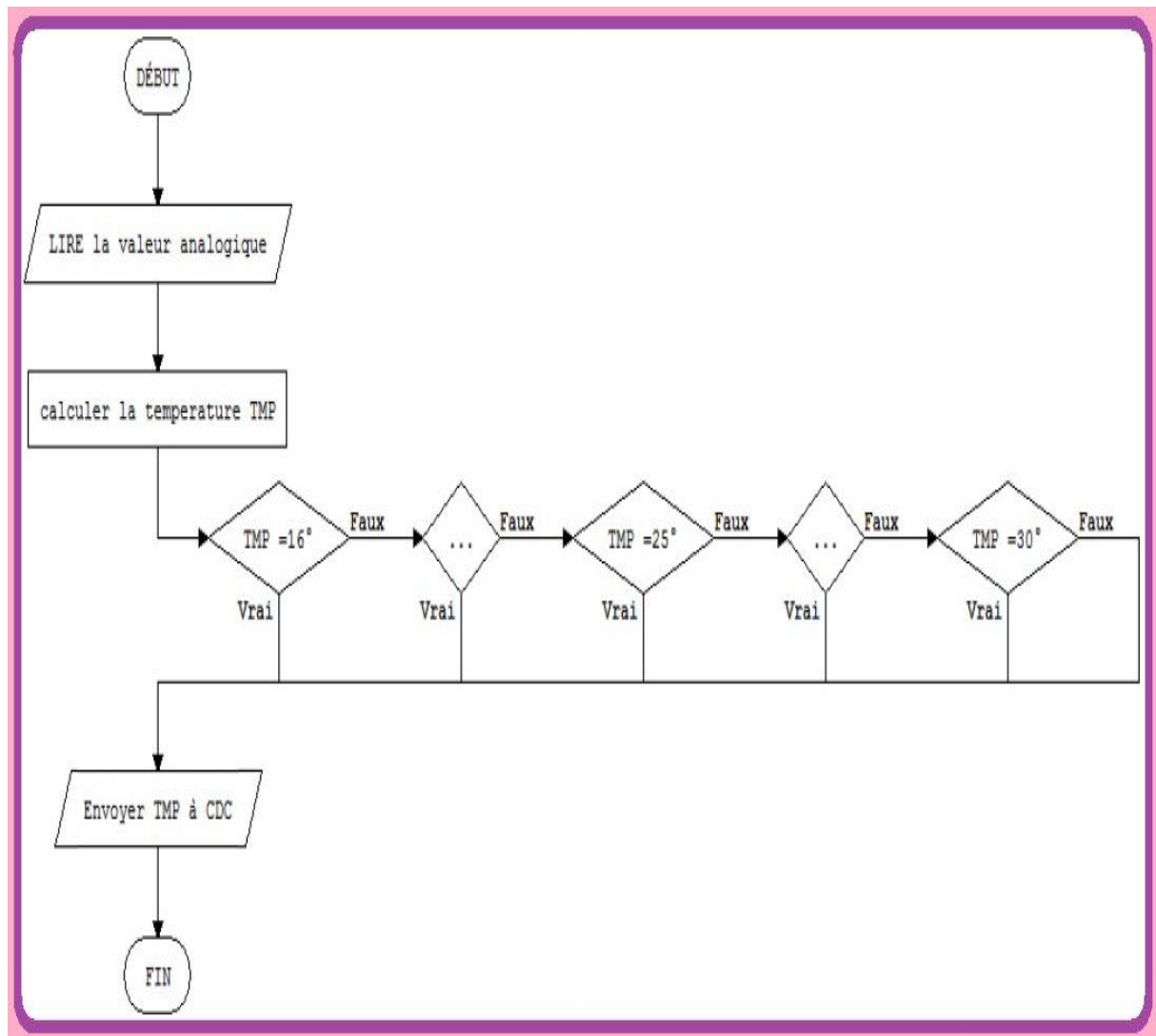


Figure 3.8 : Organigramme du fonctionnement du capteur de température ambiant

Quand les deux LM35 captent une température inférieure à -40°C à l'intérieur des échangeurs thermiques qui est la température d'ébullition du gaz frigorigène, la carte déclenche l'inversion de cycle pour dégivrer le givre formé sur l'évaporateur.

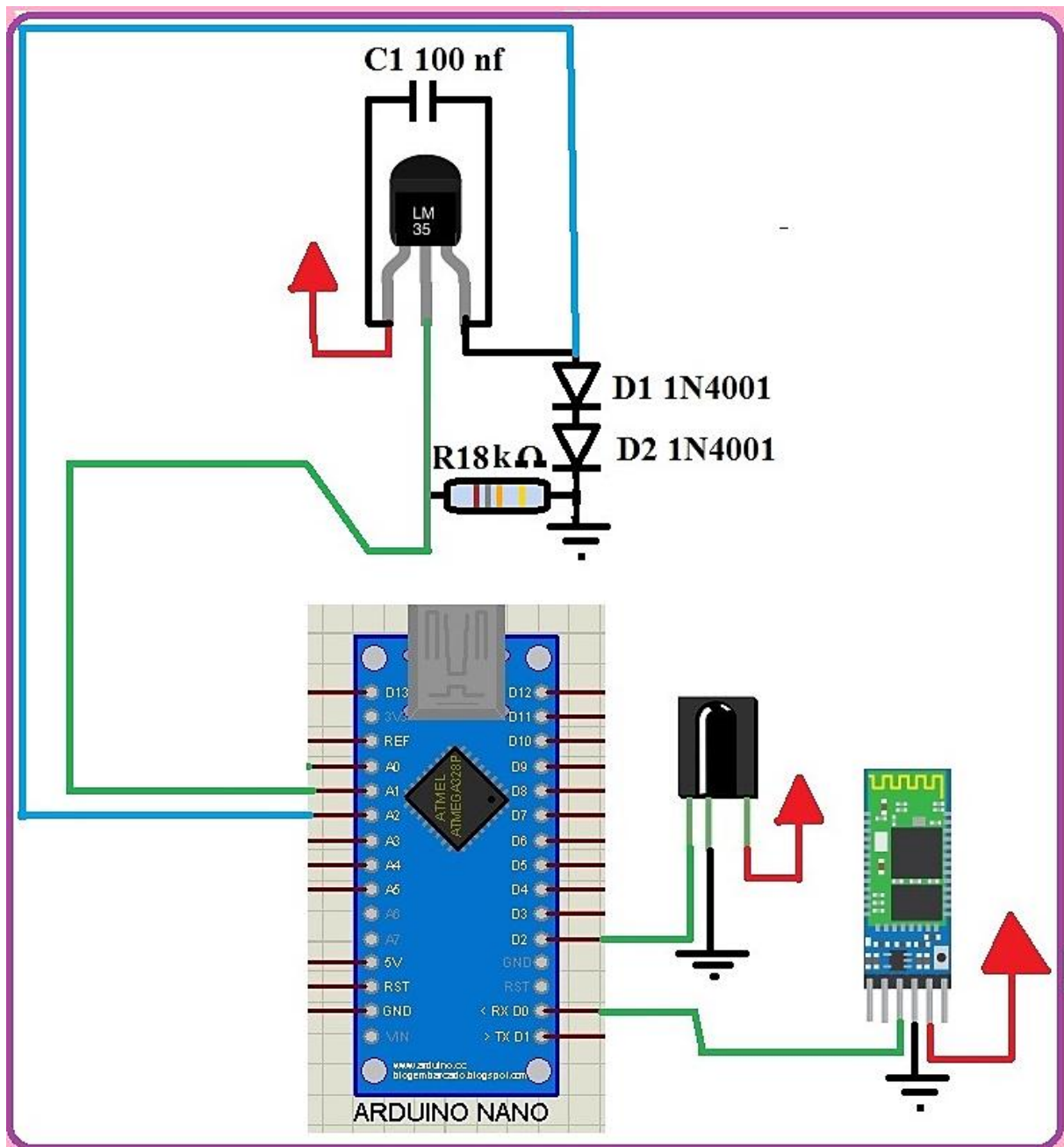


Figure 3.9 : Schéma de brochage du deuxième capteur LM35

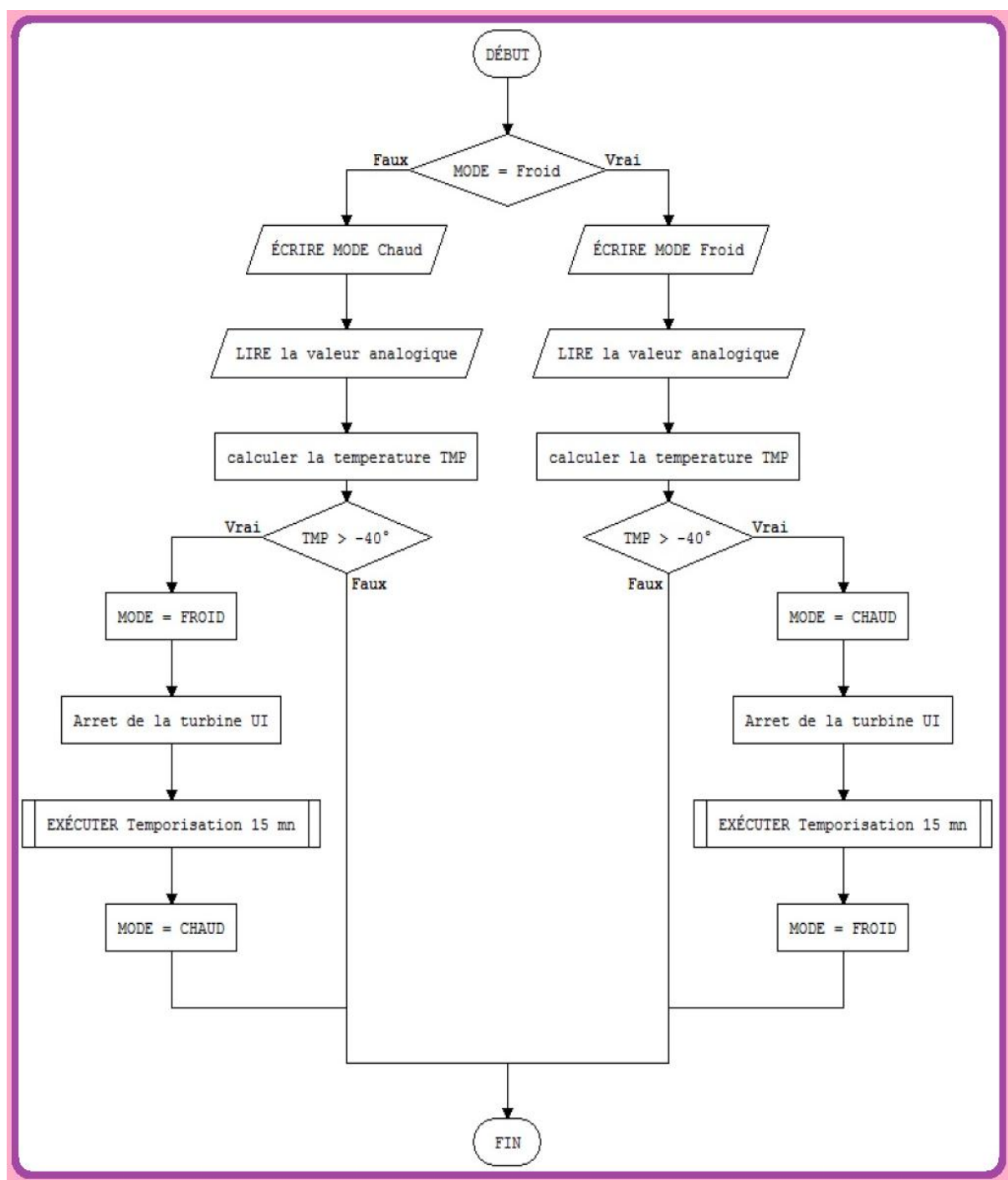


Figure 3.10 : Organigramme du fonctionnement de la sonde entier givrage

III.2.2.1-Deuxième partie de la carte électronique :

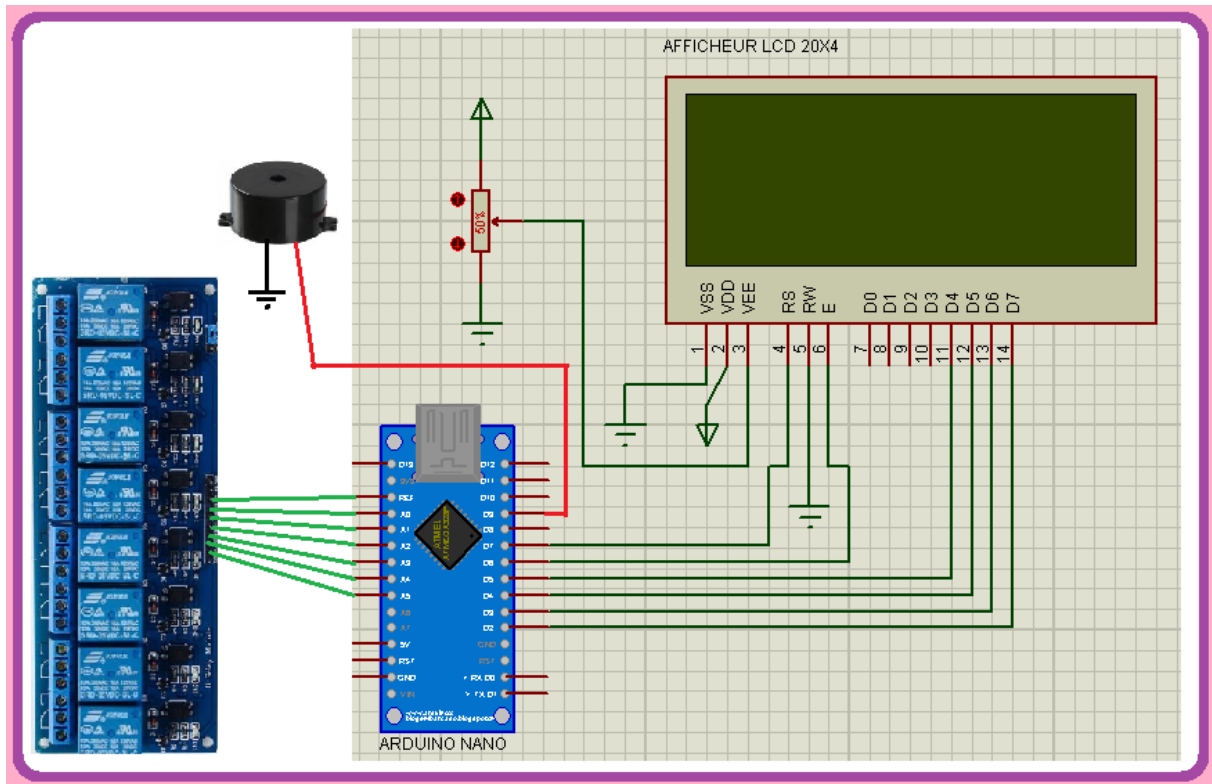


Figure 3.11 : Module de commande et d'affichage

III.2.2.1.1-Carte Arduino-Uno :

La carte Arduino-nano est composée du module de commande et d'affichage, reçoit les données du module d'acquisition et du module serveur Web grâce à une communication série (TX, RX). Elle va actionner ou non l'afficheur ou le buzzer ou un des relais qui sont reliés aux moteurs et à l'électrovanne.

La carte est à base de microcontrôleur Atmega 328 cadencé par un oscillateur externe de fréquence de 16 Mhz.

- Les entrées/sorties numériques ont été configurée,
- La broche D2 est réservée à l'entrée D7 de l'afficheur,
- La broche D3 est réservée à l'entrée D6 de l'afficheur,
- La broche D4 est réservée à l'entrée D5 de l'afficheur,
- La broche D5 est réservée à l'entrée D6 de l'afficheur,
- La broche D6 est réservée à l'entrée E de l'afficheur,
- La broche D7 est réservée à l'entrée RS de l'afficheur,
- La broche D9 est réservée à l'entrée du buzzer.

Chapitre III

- La broche A0 est réservée à l'entrée du relais qui actionne **R1 liée à la turbine**.
- La broche A1 est réservée à l'entrée du relais qui actionne **l'électrovanne**.
- La broche A2 est réservée à l'entrée du relais qui actionne le **moteur ventilateur**.
- La broche A3 est réservée à l'entrée du relais qui actionne le **relais au compresseur**.
- LA broche D0(Rx) est réservée à la communication en série avec les deux autres broches D1(Tx) des deux autres cartes arduino nano de la carte électronique.

III.2.2.1.2-Buzzer :

Délivre un bip sonore bien distinct pour exprime la validation de la commande de l'utilisateur

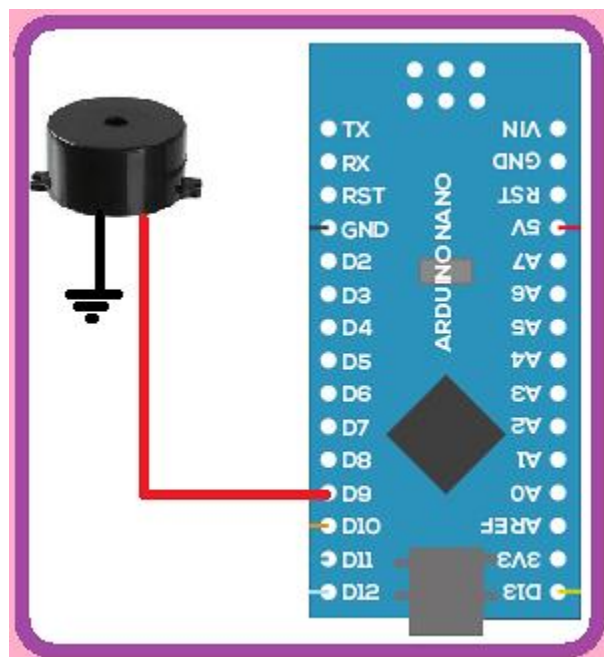


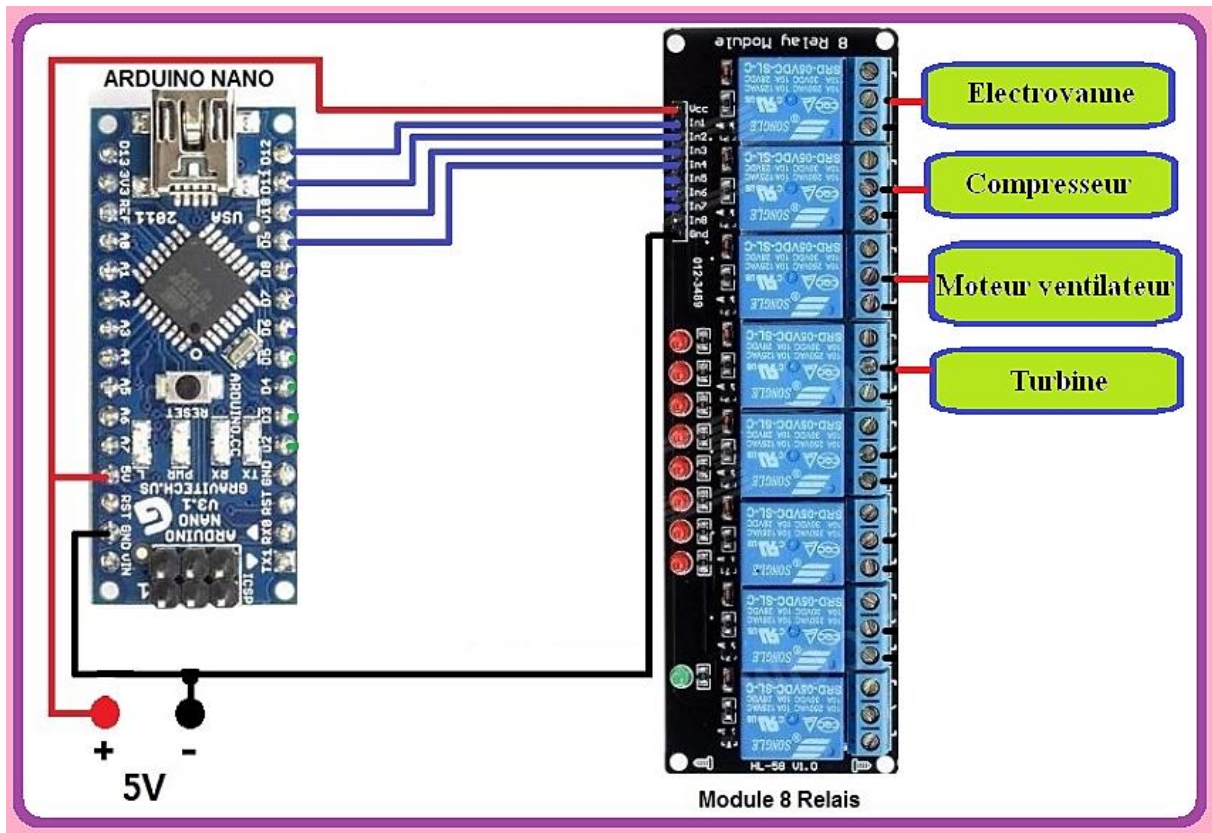
Figure 3.12 : Schéma de brochage du Buzzer

III.2.2.1.3-Carte relais :

Vu que la carte Arduino-nano fait la fonction d'une carte de commande donc on n'aura pas besoin d'une carte de puissance. Pour cela nous avons opté pour une carte relais pour pouvoir donner suffisamment d'énergie pour fonctionner les moteurs et l'électrovanne qui ont besoin d'une tension de 220 Volts pour leurs fonctionnement.

Le principe est simple, selon le relai qui est actionné par l'arduino nano, le moteur qui

est relié à ce même relai s'actionne.



**Figure 3.13 : Schéma fonctionnel de connexion de la carte électronique
Au climatiseur à travers la carte relais :**

Chaque relai est excité par une sortie numérique d'Arduino. La sortie du relai R1 liée à la turbine, R2 est liée à l'électrovanne, R3 et relai au moteur du ventilateur, R4 et relai au compresseur.

III.2.2.1.4-Afficheur LCD 4*20 :

Il permet d'afficher des informations diverses. Il s'agit d'un afficheur 20 caractères sur 4 lignes avec affichage rétro éclairé bleu.

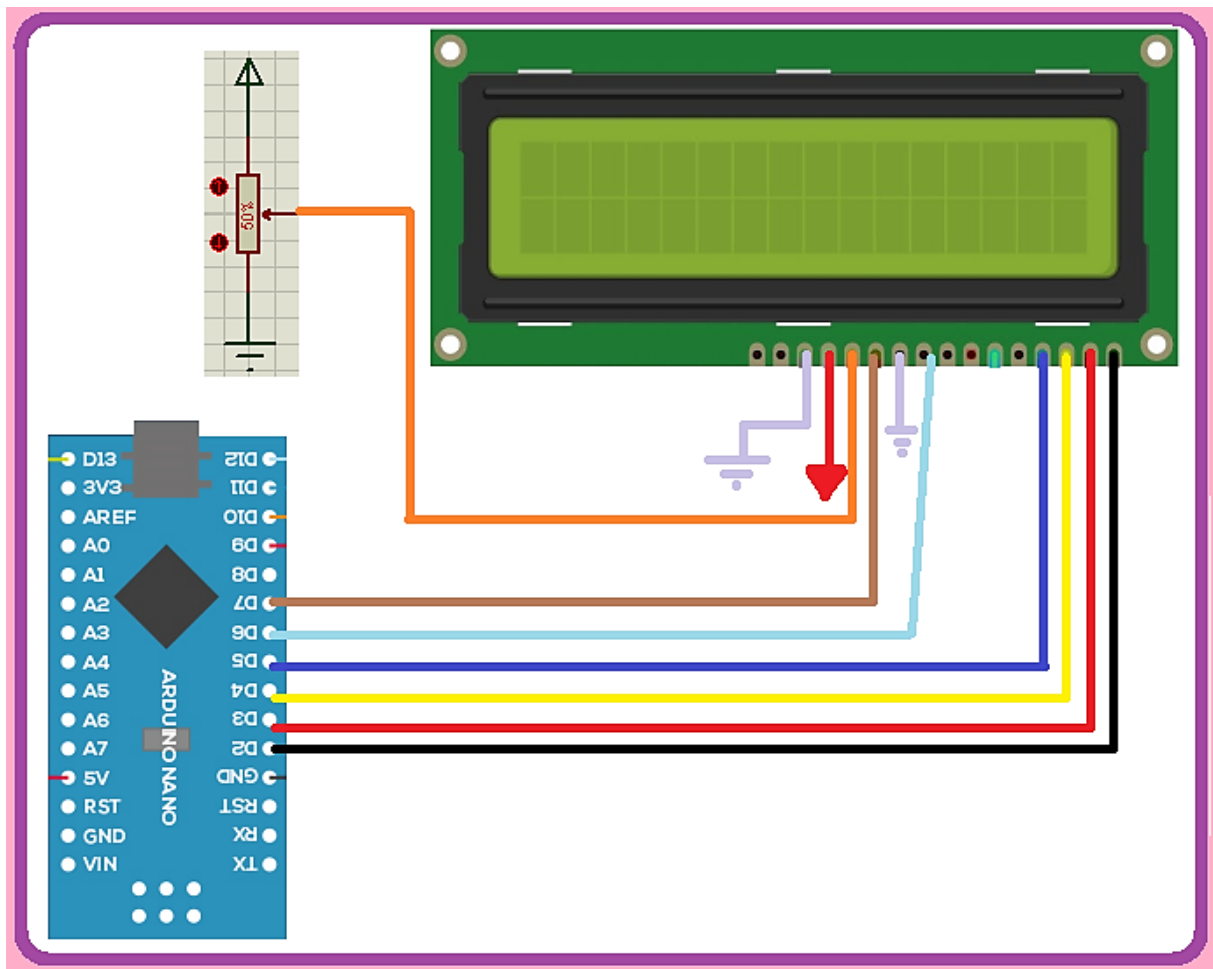


Figure 3.14 : Schéma de brochage afficheur LCD 4*20

On utilise l'écran LCD pour afficher les paramètres suivants :

1. Message d'accueil soit :
 - climatiseur **Eteint**
 - climatiseur **Allumé**
2. Le fonctionnement de notre système
 - Mode de fonctionnement du climatiseur
 - Mode ventilateur
 - Mode froid
 - Mode chaud
 - Niveau de vitesse de la turbine
 - Niveau 1
 - Niveau 2
 - Niveau 3

- Températures
- Température d'ambiance
- Température demandée

III.2.3.1-Troisième parité :

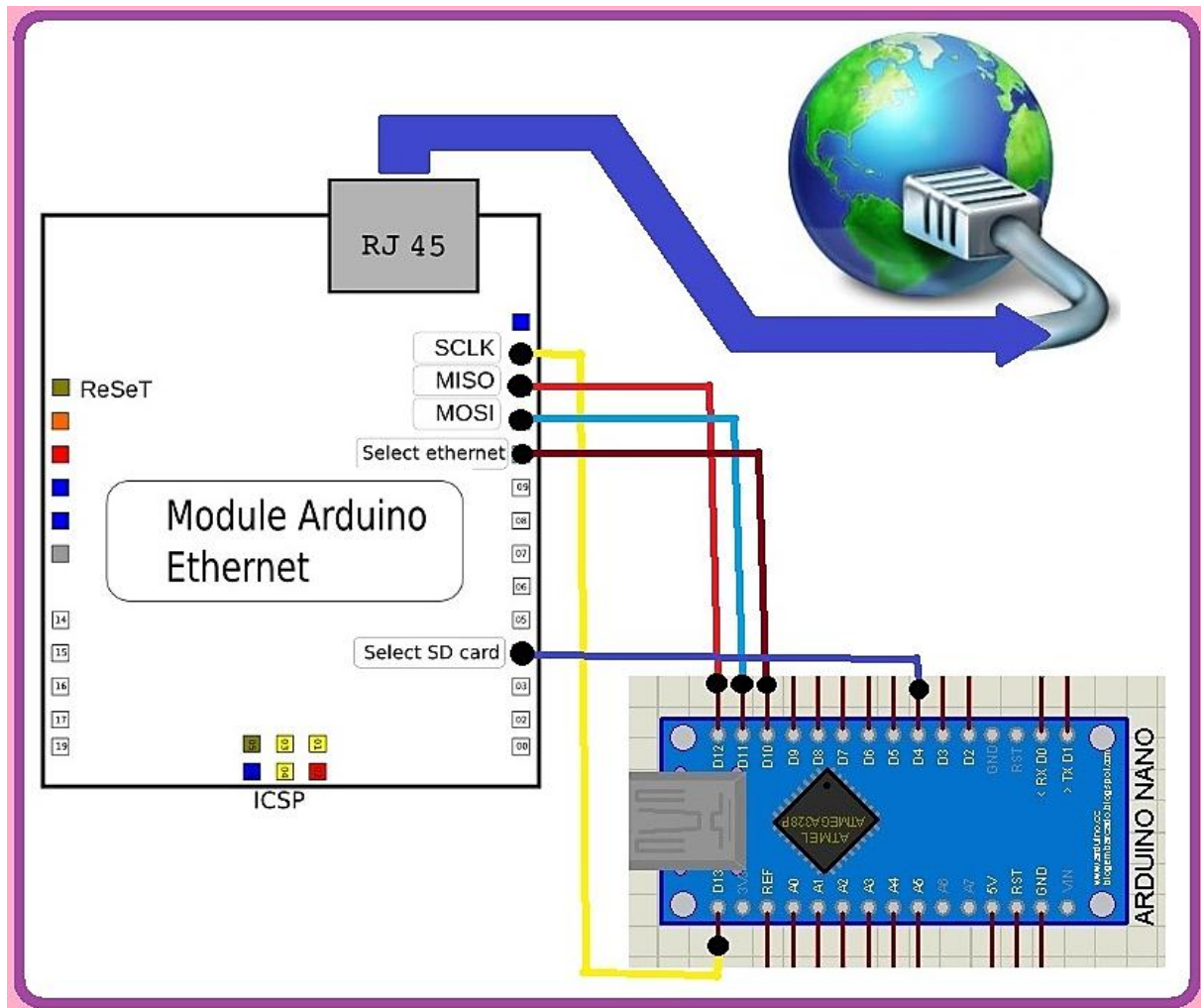


Figure 3.15 : Schéma module serveur Web

III.2.3.1.1- carte arduino nano :

Notre module est classé dans les catégories des projets IOT (internet of things) qui veut dire internet des objets. Pour faire fonctionner notre module nous avons programmé notre carte arduino nano en serveur Web, (générateur des pages Web on HTML). Lorsqu'un utilisateur tape l'adresse IP de notre serveur web dans n'importe quel navigateur internet, la carte arduino-nano lira les pages qui sont enregistrées dans la carte mémoire de shield arduino puis les envoie à

Chapitre III

L'utilisateur. Lorsque ce dernier clique sur les boutons de commande de notre page Web, L'arduino va traiter l'ordre et elle va envoyer les commandes demandées au module de commande et d'affichage. Ce dernier, va traiter la commande et actionner le climatiseur.

La carte est à base de microcontrôleur Atmega 328 cadencé par un oscillateur externe de fréquence de 16 Mhz.

- Les entrées/sorties numériques ont été configurées.
- La broche D4 est réservée à l'entre D4 de la carte réseau,
- La broche D10 est réservée à l'entre D10 de la carte réseau,
- La broche D11 est réservée à l'entre D11 de la carte réseau,
- La broche D12 est réservée à l'entre D12 de la carte réseau,
- La broche D13 est réservée à l'entre E13 de la carte réseau.

III.2.3.1.2-Carte réseau RJ45 :

Le shield arduino Ethernet est un module Arduino additionnel à notre carte arduino-nano qui lui permet de relier le programme serveur Web qui est flaché dans la mémoire flash de notre carte nano au réseau locale ou internet.

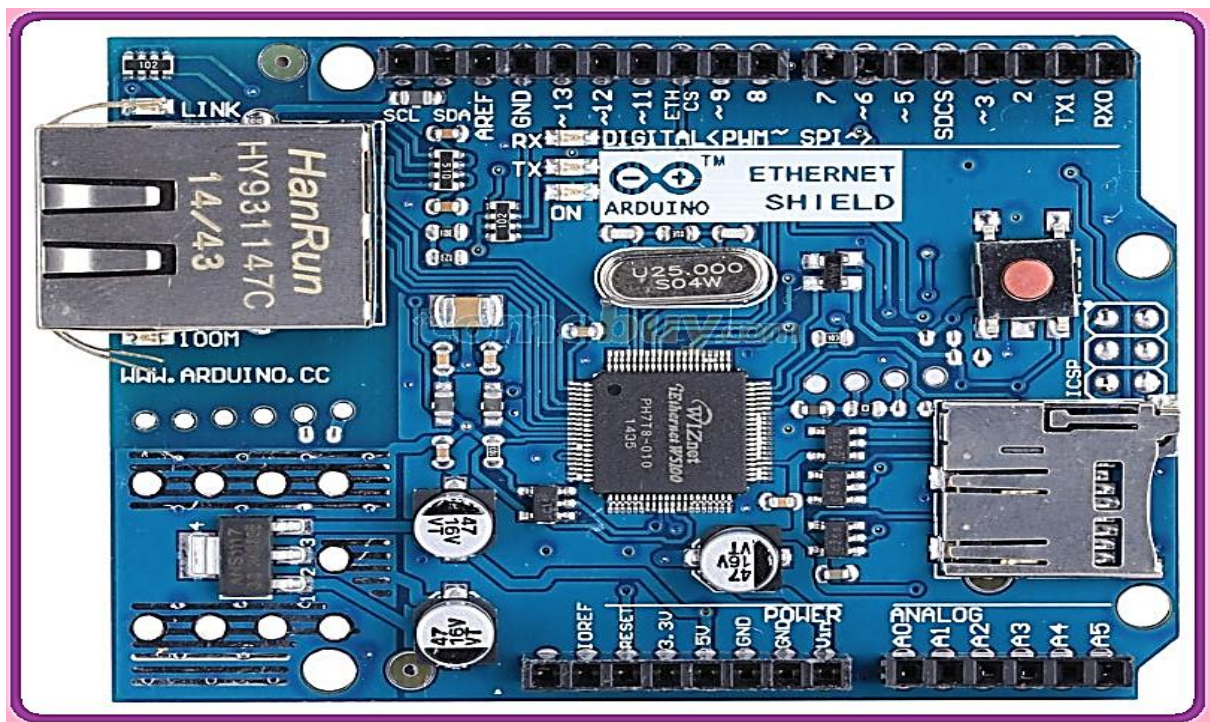


Figure 3.16 : Carte Réseau Arduino

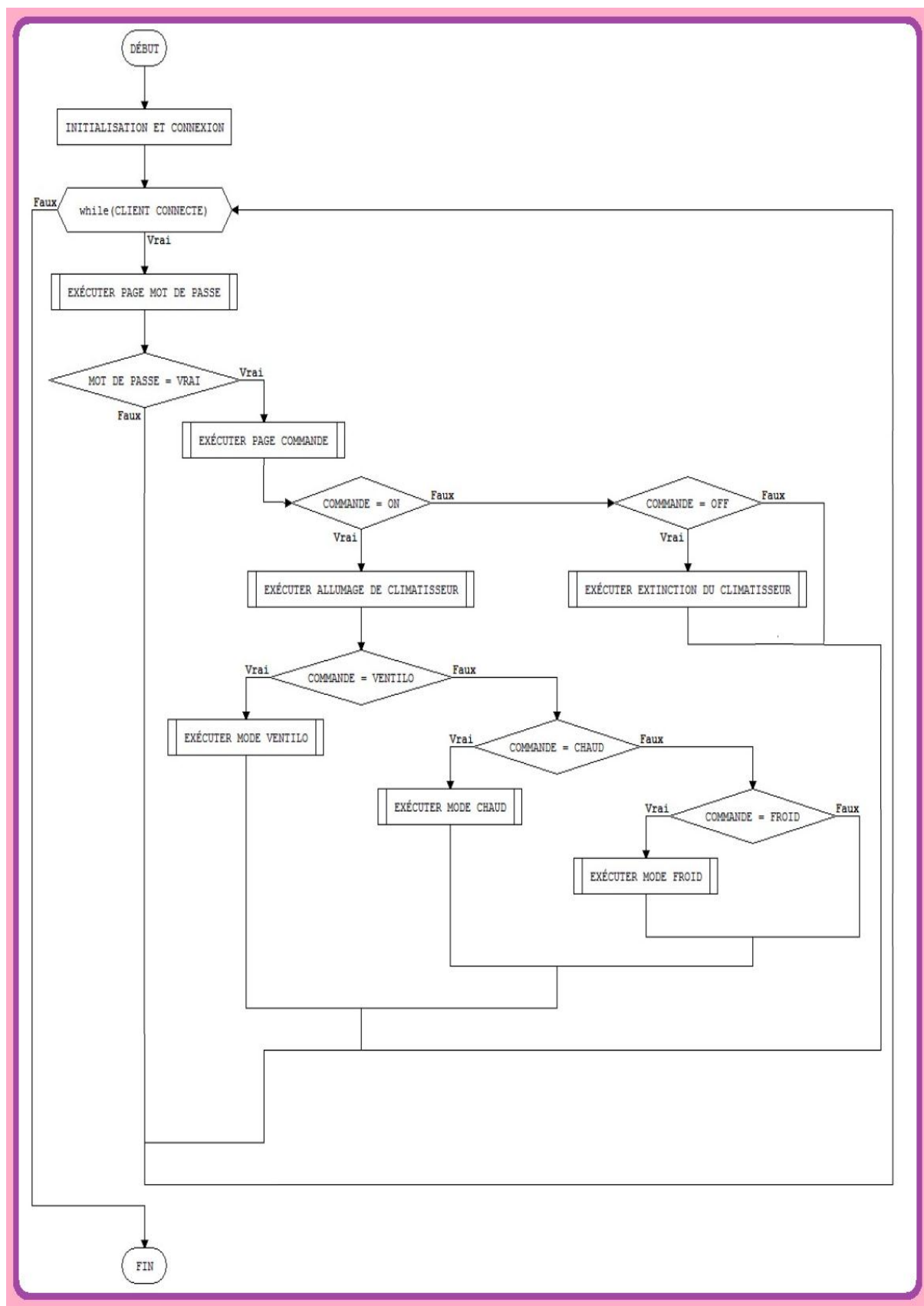


Figure 3.17 : Organigramme du fonctionnement Carte réseauRJ45

III.3-Méthode de contrôle du système par la carte électronique :

Le système à l'état initial est au repos. Dès que l'opérateur donne l'ordre de démarrage par l'une des commandes (infrarouge, Bluetooth au réseau), le système par défaut se met au mode ventilateur et la vitesse de la turbine est 33% de sa vitesse Max.

Quand l'opérateur choisie l'un des modes de fonctionnement :

III.3.1-Mode ventilateur :

La carte électronique vérifie que tant le bouton de changement de mode ou le bouton d'arrêt n'est pas actionné, le mode ventilateur continu de fonctionner.

III.3.2-Mode froid :

La carte électronique vérifie que tant que la température demandée **Td** est inférieure à la température d'ambiance **Ta**, la carte va commander le système de continuer de fonctionner dans le mode froid. Si la température **Td** est supérieure à la température **Ta** ou le bouton de changement de mode ou le bouton d'arrêt est actionné alors le mode froid arrête de fonctionner.

III.3.3-Mode chaud :

La carte électronique vérifie que tant que la température demandée **Td** est supérieure à la température d'ambiance **Ta** la carte va commander le système de continuer de fonctionner dans le mode chaud. Si la température **Td** est inférieure à la température **Ta** ou le bouton de changement de mode ou le bouton d'arrêt est actionné le mode chaud arrête de fonctionner.

III.3.4-Vitesse de la turbine :

La vitesse de la turbine est indépendante des modes de fonctionnements. L'opérateur peut augmenter la vitesse de la turbine dans les trois modes de fonctionnement.

La vitesse varie selon le nombre de fois que l'opérateur aura cliqué sur le bouton de variation (n : nombre de clique)

3*n : bouton de vitesse de la turbine sélectionne 3*n fois c'est la vitesse1=33% V-MAX,

3*n+1 : bouton de vitesse de la turbine sélectionne 3*n+1 fois c'est la vitesse2=33% V-MAX,

3*n+2 : bouton de vitesse de la turbine sélectionne 3*n+2 fois c'est la vitesse3=100% V-MAX,

III.3.5-GRAF7 :

Vitesse1 : vitesse de la turbine à 33% de la vitesse Max,

vitesse2 : vitesse de la turbine à 66% de la vitesse Max,

vitesse3 : vitesse de la turbine à 100% de la vitesse Max,

3*n : bouton de vitesse de la turbine sélectionne 3*n fois,

3*n+1 : bouton de vitesse de la turbine sélectionne 3*n+1 fois,

3*n+2 : bouton de vitesse de la turbine sélectionne 3*n+2 fois,

Ta : température d'ambiance captée par le capteur thermique LM35,

Td : température demandée par l'utilisateur.

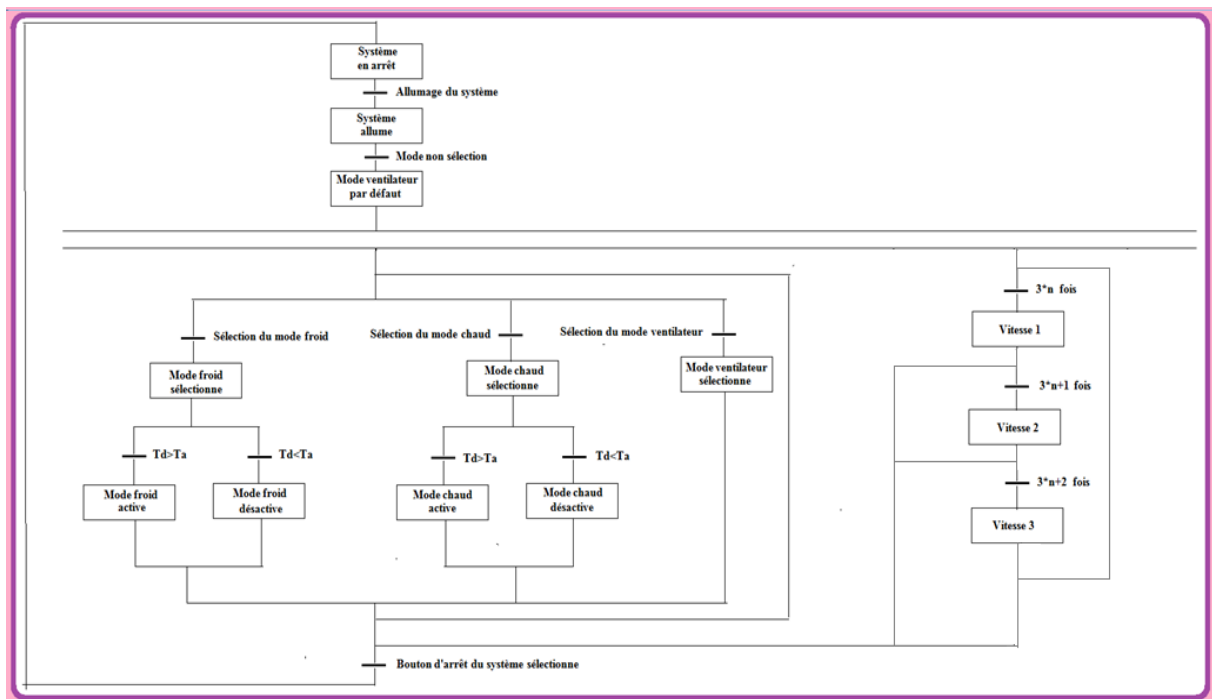


Figure 3.18 : Méthode de contrôle du système par la carte électronique

III.4-Différents modes de contrôle du climatiseur :

Notre carte électronique à base de arduino Nano va permettre de commander la carte relais de trois manières différentes avec une :

1. Commande infrarouge
2. Commande Bluetooth
3. Commande réseau

III.4.1-Commande infrarouge :

Grace au récepteur infrarouge nous pouvons contrôler notre climatiseur avec une télécommande infrarouge ENIEM.

Le principe de fonctionnement de la télécommande est simple. Lorsque vous appuyez sur un bouton, une information est codée en binaire suivant un protocole, ensuite un circuit électrique à l'intérieur de la télécommande va se charger de moduler cette information binaire pour transmettre des ondes infrarouges. On utilise pour cela une Led sauf que l'ouïe humaine ne peut pas visualiser la lumière émise. Par contre un récepteur infrarouge va d'abord la transformer en un signal électrique puis amplifie le signal, le filtre et le numérise pour restitue l'information binaire qui avait été codé par la télécommande. Tout ceci s'appelle **une chaine d'acquisition et de restitution de signal**.

Ce mode de contrôle très efficace prend en charge (l'arrêt/démarrage), les modes de fonctionnements (chaud, froid, ventilateur), les différents vitesses de la turbine et toutes les variations de températures de consignes possibles qui sont entre [16,30] °C du climatiseur.

Les seuls points négatifs est que la distance de contrôle que permet la commande infrarouge est très limite pas plus de quinze mètres et il faut être face au climatiseur si non l'ordre n'est pas transmis.

III.4.2-Fonctionnement de la télécommande :

- Pour allumer ou éteindre le climatiseur il faut cliquer sur le bouton : 4,
- Pour changer le mode de fonctionnement du climatiseur il faut cliquer sur le bouton: 5,
- Pour changer la vitesse de la turbine il faut cliquer sur le bouton : 6,
- Pour diminuer la température demande il faut cliquer sur le bouton : 7,
- Pour augmenter la température demande il faut cliquer sur le bouton : 8.

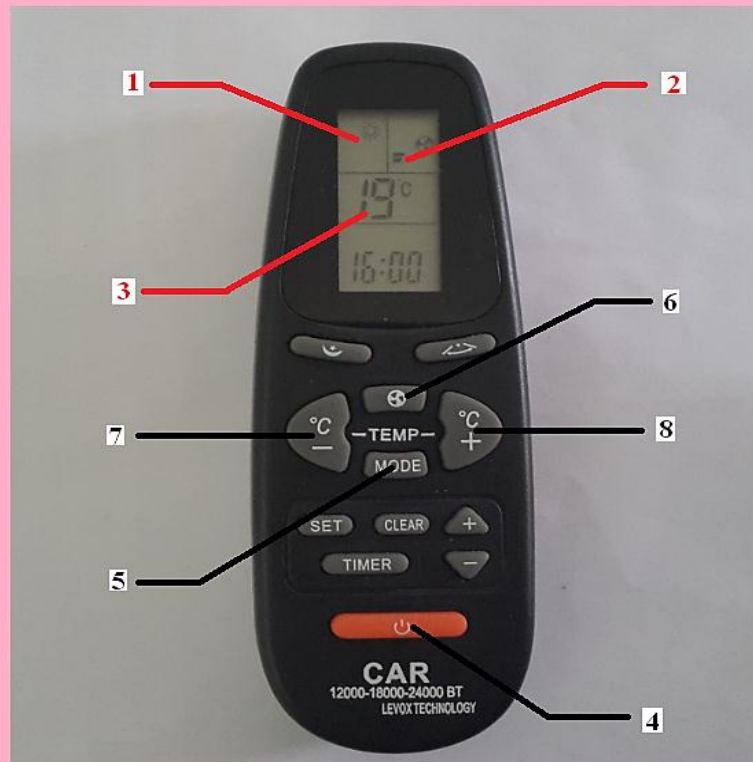


Figure 3.19 : Télécommande ENIEM adapte à notre carte électronique

- 1-Mode de fonctionnement,
- 2-Vitesse de la turbine,
- 3-Température demande,
- 4-Bouton allumage ou arrêt,
- 5-Bouton de mode de fonctionnement,
- 6-Bouton de variation de la vitesse de turbine,
- 7-Bouton de diminution de ta température demandée,
- 8- Bouton d'augmentation de ta température demandée.

III.4.2-Commande Bluetooth :

Le module Bluetooth HC-05 assure la communication entre l'application **Android** que l'on a installé sur notre téléphone **Android** et notre carte électronique qui elle assure la commande et le bon fonctionnement du climatiseur. Cette communication 'est une communication série (TX/RX) avec une vitesse de 9600 bit par seconde.

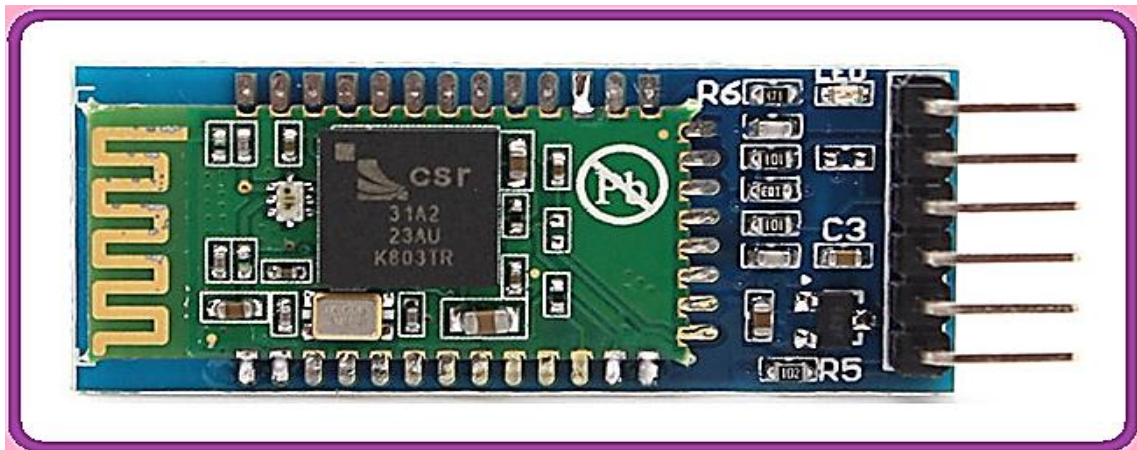


Figure 3.20 : Module Bluetooth HC-05

L'application Clim commande.APK est une application que l'on a développée sur le site APP inventer2. Grâce à celle-ci qui est installée sur notre téléphone **Android**, on peut contrôler le climatiseur en deux modes :

- Mode bouton qui joue le rôle d'une télécommande infrarouge
- Mode vocale

III.4.2.1-Mode bouton :

Dans le mode bouton il suffit d'activer le Bluetooth du téléphone et de le relier au module Bluetooth du HC-05. Après que cette liaison soit effectuée, il faut activer la connexion Bluetooth de l'application de la page de commande manuelle. Il suffit de cliquer sur les boutons voulus pour contrôler le climatiseur. L'application va générer des codes qu'elle va envoyer à notre carte qui elle va contrôler le climatiseur via la liaison Bluetooth.



Figure 3.21 : Ecran de contrôle de la commande manuelle

III.4.2.2-Mode vocal :

En mode vocal, il suffit que l'utilisateur active le Bluetooth du téléphone et de le relier au module Bluetooth du HC-05, après que cette liaison soit effectuée, il faut activer la connexion Bluetooth de l'application de la page vocale. Il suffit alors à l'utilisateur de cliquer sur l'image du milieu qui apparaît et prononcer les commandes vocales précises pour contrôler le climatiseur. L'application va faire la synthèse vocale de ces mots et de les envoyer au climatiseur.

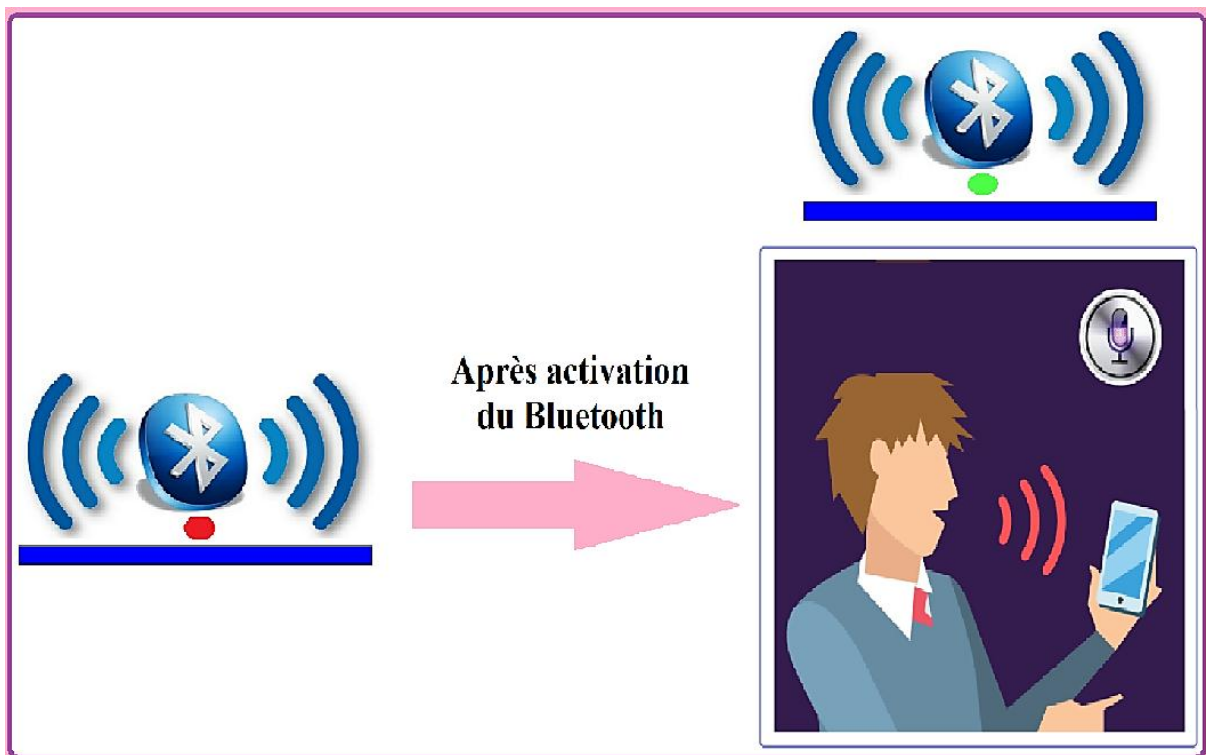


Figure 3.22 : Ecran de contrôle de la commande vocale

Les commandes vocales qu'il faut prononcer pour contrôler le climatiseur sont :

- Pour éteindre le climatiseur : Climatiseur éteint,
- Pour allume le climatiseur : Climatiseur allumer,
- Pour active le mode ventilateur : Réglage ventilateur,
- Pour active le mode froid : Réglage froid,
- Pour active le mode chaud : Réglage chaud,
- Pour règle la température : Exemple (température 16 degrés..., température 30 degrés),
- Pour règle les niveaux de vitesse : Vitesse faible, vitesse moyenne, vitesse rapide de la turbine.

III.4.3 -Commande réseau :

Pour cette commande, lorsqu'on lance n'importe quel navigateur internet sur PC ou téléphone connecté au réseau, on tape l'adresse IP de notre serveur Web (192.168.1.200). Le navigateur envoie cette requête à notre Module serveur Web. Ce dernier va charger les pages Web login (page pour identification) enregistrée dans la carte mémoire du schild arduino RJ45 et de les renvoyés à l'utilisateur. L'utilisateur doit, pour continuer l'opération, introduire le nom de l'utilisateur et le mot de passe :

The image shows a web browser window with the title 'Commande Climatisseur'. The address bar contains '192.168.1.200'. The page content includes a login form with the following elements:

- A horizontal line of hash symbols (#####) at the top.
- A label 'Utilisateur :' followed by a text input field containing the text 'admin'.
- A label 'Mot de passe :' followed by an empty password input field.
- A 'Se connecter' button centered below the password field.
- Another horizontal line of hash symbols (#####) below the button.
- Footer text: 'Réalisé par : GRIBISSA AGHILAS' and '2015/2016'.

Figure 3.23 : Page Web nom d'utilisateur et mot de passe

Chapitre III

Le serveur va traiter cette information (nom d'utilisateur et mot de passe) et il va vérifier l'authentification de l'utilisateur. Si l'utilisateur a tapé les bons mots de passe et nom d'utilisateur, le serveur Web va charger la page principale de la carte SD et l'envoyer à l'utilisateur et à base de cette page principale que l'utilisateur peut contrôler et commander notre climatiseur via un réseau.

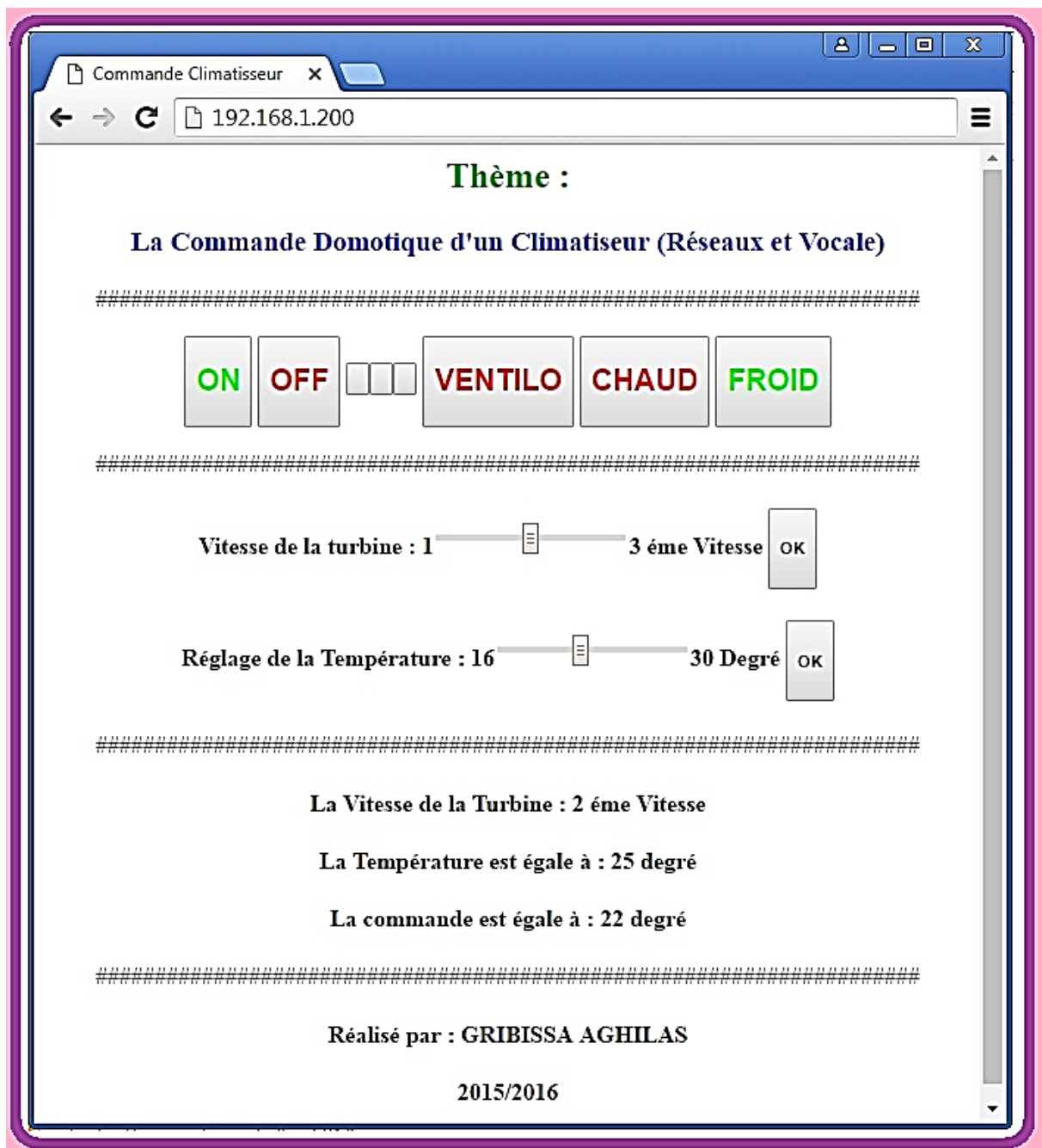


Figure 3.24 : Page Web de la page principale.

III.5- installation :

La carte relais permet de donner l'énergie électrique nécessaire à l'électrovanne, la turbine, le moteur ventilateur et le compresseur et de les relier à la carte électronique qui va les commander.

Voici le schéma de connexion de la carte électronique à la carte relais et du branchement électrique du système à la prise 220V à travers la carte relais :

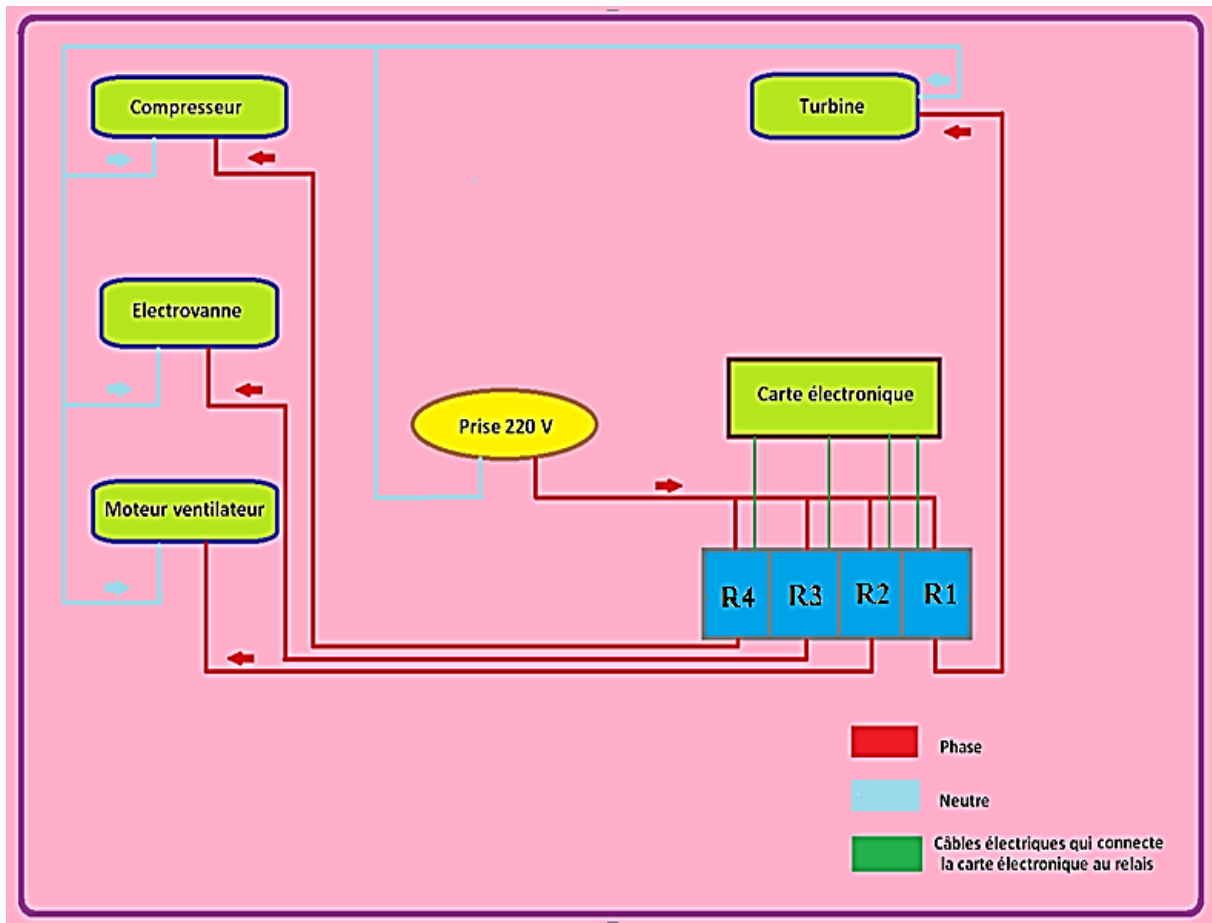


Figure 3.25 : Branchement électrique de notre système à la prise 220V à travers la carte relais qui est commandée par la carte électronique.

III.6-Réalisation pratique :

Vu que on n'a pas sous les mains un vrai climatiseur. Nous avons dû opter pour une maquette qui va représenter la partie puissance et la partie commande du climatiseur qui est illustrée comme suite sur l'image :

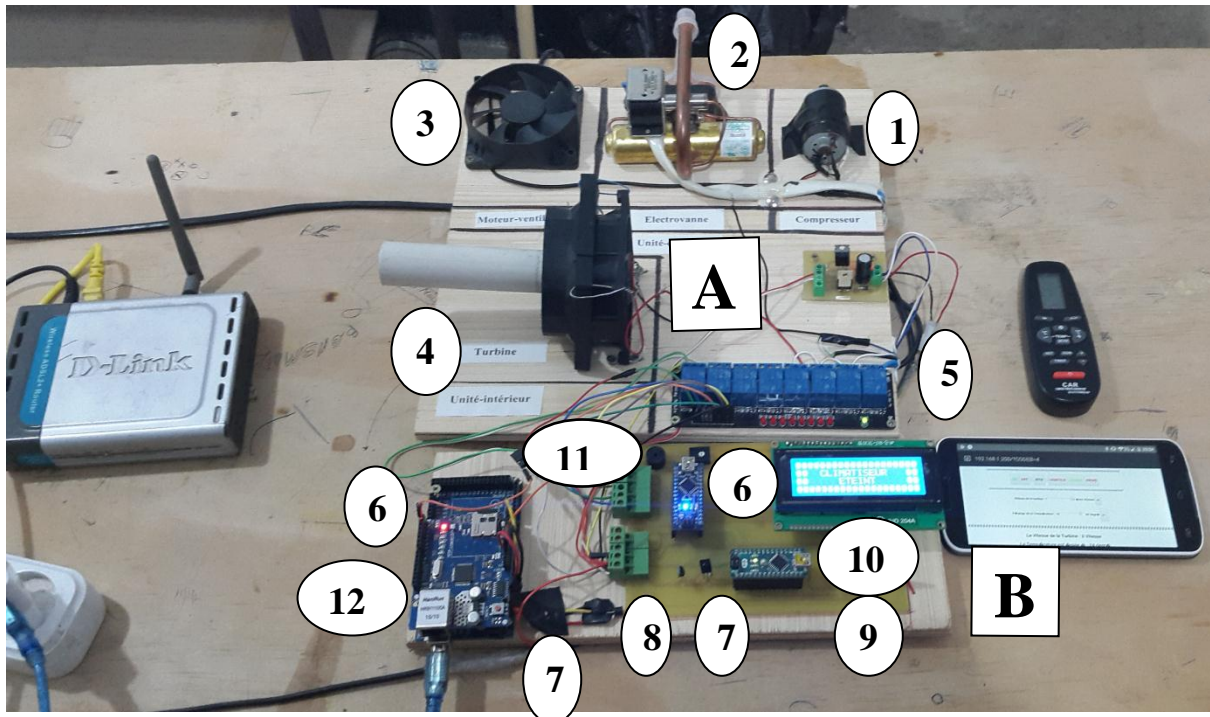


Figure 3.26 : Maquette de la partie commande et la partie puissance

Partie puissance "A" se compose de :

- 1) Un petit moteur à courant continu de 7V qui représente le compresseur,
- 2) Une vraie électrovanne d'un climatiseur,
- 3) Un moteur à courant continu de 12V qui représente le moteur-ventilateur,
- 4) Un moteur à courant continu de 12V qui représente la turbine,

Partie commande "B" se compose de :

- | | |
|------------------------------------|--------------------------------------|
| 5) Une carte relais à huit canaux. | 10) Un afficheur LCD 4x20, |
| 6) Trois cartes Arduino-nano, | 11) Un buzzer, |
| 7) Deux capteurs LM35 | 12) Une carte réseau shield entière, |
| 8) Un récepteur infrarouge, | |
| 9) Un module Bluetooth, | |

III.7-Conclusion :

Dans ce chapitre, nous avons décrit le processus et les étapes de la réalisation matérielle et logicielle de notre carte électronique qui va contrôler le climatiseur ENIEM 12 000BTU ainsi que son installation et branchement au climatiseur.

Conclusion Générale

Conclusion générale et perspective :

Dans ce travail, nous avons conçu et réalisé une carte électronique qui contrôle le climatiseur ENIEM 12 000BTU de trois manières différentes avec :

- Une commande infrarouge,
- Une commande Bluetooth,
- Une commande Réseau.

La commande infrarouge est assurée par un récepteur infrarouge et une télécommande ENIEM que l'on a adapté à notre carte électronique.

La commande Bluetooth est assurée par le module Bluetooth **HC-05** et une application développe sur le site **APP inventer2**. Cette application va être installée sur un téléphone **Android**.

La commande réseau est assurée par le **shiled arduino ethernet** qui va relier le programme serveur Web qui est flache dans la "mémoire flash" de notre carte nano via le réseau local ou internet.

Nous nous sommes intéressés à travers ce projet à développer une carte électronique permettant un meilleur confort dans le contrôle de notre climatiseur.

Exemple : Le mode Bluetooth nous assure une commande vocale du climatiseur qui peut être très intéressante pour les non-voyants.

Le mode réseau peut assurer un contrôle du climatiseur par internet qui veut dire un contrôle sur une longue distance et à tout moment.

La réalisation matérielle et logicielle de ce projet a donné des résultats satisfaisants. En effet, elle apporte une souplesse et facilite d'utilisation du climatiseur ENIEM 12 000BTU.

En perspective, notre système peut être amélioré :

Par la qualité du programme en englobant les trois programmes qui sont destinés pour chaque carte arduino nano dans une seule carte arduino Mega. L'objectif étant d'élever la qualité du programme en ajoutent les interruptions et en diminuant la consommation d'énergie (au lieu de faire fonctionne trois micro contrôleurs on se retrouve avec un seul microcontrôleur qui est géré par un programme plus complet et mieux structuré).

Toujours dans le cadre de l'amélioration du système, il y aurait été souhaitable de compléter par l'étude du moteur pas à pas. Cependant il n'est pas prépondérant dans le fonctionnement du climatiseur.

ANNEXES

Présentation de la carte Arduino Nano :

Description:

Le modèle Arduino Nano est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega328. Le microcontrôleur de la famille AVR dont la programmation peut être réalisée en langage C Arduino. Cette carte possède 14 entrées/sorties numériques (dont 6 peuvent être utilisées comme étant des PWM (pulse Width modulation), 6 entrées analogiques avec un convertisseur Analogique/numérique de 10 bits de résolution, 1 résonateur céramique (quartz) de 16 Mhz. 1 connecteur ICSP (in-circuit serial programming) qui permet la programmation du microcontrôleur sur le circuit sans avoir à l'enlever, 1 connecteur jack pour une alimentation extérieure, un bouton de reset pour mettre le processus à zéro.

L'avantage de cette carte c'est qu'elle n'a pas besoin de pilote pour faire la conversion FTID USB/série. Elle a juste un petit microcontrôleur ATmega programmé comme convertisseur USB/série.

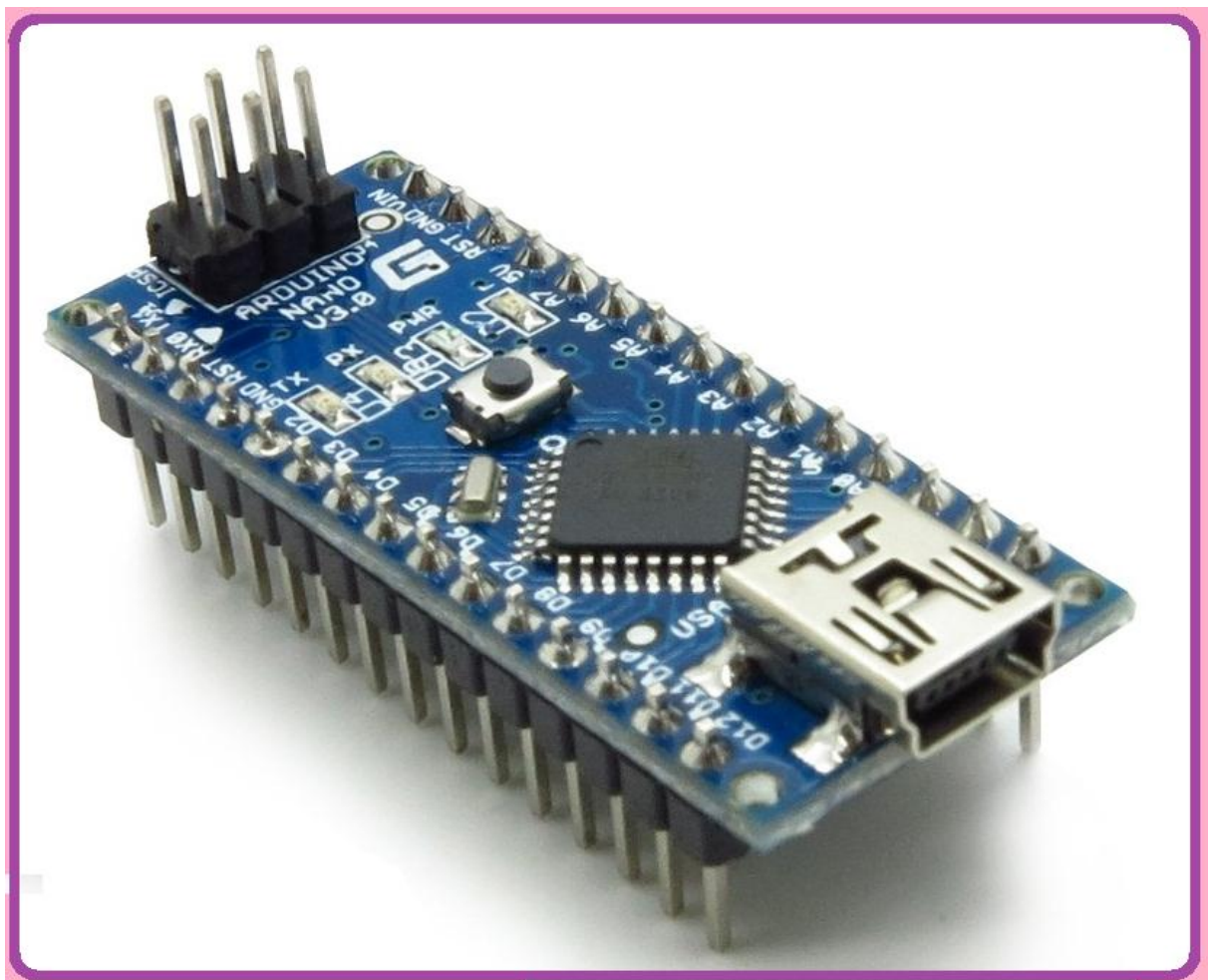


Figure4.1: Description de la carte Arduino nano.

L'intérêt principal des cartes Arduino est leur facilité de mise œuvre. Arduino fournit un environnement de développement s'appuyant sur des outils open source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB.

En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties courantes : gestion des E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus I2C, exploitation de servomoteurs...ect.

Alimentation de la Carte arduino :

La carte Arduino nano peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte. L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 5V à 12V sous 500mA) ou des piles (ou des accumulateurs)

L'adaptateur secteur peut être connecté en branchant une prise de 2.1mm, dédié du pôle positif au centre dans le connecteur jack de la carte, les fils en provenance d'un bloc de piles ou d'accumulateurs peuvent être insérés dans les connecteurs des broches de la carte appelées GND (masse ou 0V) et vin (tension positive en entrée) de connecteur d'alimentation. La carte peut fonctionner avec une alimentation externe de 6 à 20V. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Ainsi, la plage idéale recommandée pour alimenter la carte Nano est entre 7V et 12V.

Il est à noter qu'il est strictement dangereux d'utiliser une alimentation externe via la prise jak et avoir le câble USB connecté.

Les broches d'alimentation sont les suivantes:

- **VIN:** (à distinguer du 5V de la connexion USB ou autre source 5V régulée). on peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, on peut accéder à la tension d'alimentation sur cette broche
- **5V:** la tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (un régulateur de tension est intégré dans la carte). la 5V fournie par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit aussi 5V régulé) ou de toute autre source d'alimentation régulée.

- **3V3:** une alimentation de 3.3V fournie par le régulateur de 3.3V de la carte est disponible: ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu de 5V), l'intensité maximale disponible sur cette broche est 50mA.

Protection du port USB contre la surcharge en intensité:

La carte Arduino nano intègre un poly-fusible réinitialisable qui protège le port USB de votre ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité), Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection, si plus de 500mA sont appliqué au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

Gestion des mémoires dans la carte Arduino:

L'ATMega 328 a 32Ko de mémoire FLASH pour stocker le programme (dont 0.5Ko également utilisé par le bootloader), l'ATMega328 a également 2Ko de mémoire RAM (volatile) et 1Ko d'EEPROM (nom volatile: mémoire qui peut être lue l'aide de la librairie EEPROM).

Entrées/sorties numériques:

Chacune des 14 broches numériques de la carte Nano (numérotées de 0 à 13) 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 40 mA d'intensité et dispose d'une résistance interne "résistance de rappel" (déconnectée par défaut) de 20-50 Kohms.

Cette résistance interne s'active sur broche en entrée à l'aide de l'instruction `digitalWrite(broche,HIGH)`.

- **Communication série:** broche 0 (RX) et 1 (TX). Utilisée pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. ces broches sont connectées aux broches correspondantes du circuit intégré ATMega 16U2 programmé en USB/Série de la carte (composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur). On fait appel à la transmission série à travers ces broches avec l'instruction `serialprint ()`, à condition que le câble USB soit déconnecté, sinon il va y avoir un chevauchement.

- **Interruptions externes:** broches 2 et 3, ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un chargement de valeur, voir l'instruction `attachInterrupt()` pour plus de détails.
- **Impulsion PWM (largeur d'impulsion modulée):** broches 3,5,6,9,10 et 11. Fournissent une impulsion PWM 8 bits à l'aide de l'instruction `analogWrite()`.
- **SPI (interface série périphérique):** broches 10(SS), 11(MOSI), 12(MISO), 13(SCK). ces broches supportent la communication SPI disponible avec une librairie pour la communication SPI. Les broches SPI sont également connectées sur le connecteur ISCP.
- **12C:** broche 4 (SDA) et 5 (SCL), supportent les communications de protocole 12C, disponible en utilisant la librairie `Wire/12C`.
- **LED:** broche 13, il y'a une LED incluse dans la carte connectée à la broche 13, lorsque la broche est au niveau HAUT, la LED est allumée, lorsqu'elle est au niveau BAS, la LED est éteinte.

Entrées analogiques:

La carte Nano dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c.à.d sur 1024 niveau soit de 0 à 1023) à l'aide de la très utilisée fonction `analogRead()` du langage Arduino.

NB: les broches analogiques peuvent être utilisées en tant que broches numériques : Elles sont numérotées en tant que broches numérique de 14 à 16, aux cas où le nombre de broches numériques ne suffise pas.

- **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()`. Elle est utilisée pour comparer la valeur d'une tension d'entrée par rapport à la valeur d'une tension de référence choisie.
 - **Reset:** mettre cette broche au niveau BAS entraîne la réinitialisation du microcontrôleur, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

On peut voir les différentes broches de la carte et leurs fonctions à travers la figure suivante:

Arduino	Microcontroller
1(TX)	PD1(TXD)
0(RX)	PD0(RXD)
D2	PD2(INT0)
D3	PD3(INT1)
D4	PD4
D5	PD5
D6	PD6
D7	PD7
D8	PB0
D9	PB1
D10	PB2(SS')
D11	PB3(MOSI)
D12	PB4(MISO)
D13	PB5(SCK)
A0	PC0
A1	PC1
A2	PC2
A3	PC3
A4	PC4(SDA)
A5	PC5(SCL)
A6	ADC6
A7	ADC7

Figure4.2: Brochage de la carte Arduino nano

Arduino et la communication avec l'extérieur:

La carte Arduino Nano dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs.

L'ATMega 328 dispose d'une UART (universal asynchronous receiver transmitter) ou émetteur-récepteur universel asynchrone, pour la communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX).

Un circuit intégré ATMega 16U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparaît comme un port COM virtuel pour les logiciels qui utilisent les ports séries virtuels sur l'ordinateur. Le code utilisé pour programmer l'ATMega 16U2 utilise driver standard USB COM, et aucun autre driver externe n'est nécessaire.

L'IDE gratuit d'Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignotent lorsque les données sont transmises via le circuit intégré USB/Série et la connexion USB vers l'ordinateur (mais pas pour les communications séries sur la broches 0 et 1). Une librairie série logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte Nano.

L'ATMega 328P supporte également la communication par protocole I2C/ SPI:

- L'IDE d'arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C.
- Pour utiliser la communication SPI, la librairie SPI est disponible, il suffit de l'inclure dans le programme au niveau de l'IDR lors de la programmation.

- **Courant max broches E/S**=40mA.
- **Courant max sur sortie 3.3V**=50mA
- **Fréquence horloge** = 16MHz.
- **Dimensions**= 68,6mm*53.3mm
- **Timers/courters:** Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits).
Chaque timer peut être utilisé pour générer deux signaux PWM (6 broches OCxA/OCxB)
- **Plusieurs broches multifonctions:** Certaines broches peuvent avoir plusieurs fonctions différentes choisies par l'utilisateur.
- **PWM :** 6 broches OC0A(PD6), OC0B(PD5), OC1A(PB1), OC1B(PB3), OC2A(PB3), OC2B(PD3).
- **Analog to Digital Converter (resolution 10 bits) :** 6 entrées multiplexes ADC0(PC0) a ADC5(PC5)
- **Gestion bus 12C (TWI Two Wire Interface):** le bus est exploité via les brooches SDA (PC5)/SCL(PC4).
- **Port série (USART) :** Emission/réception via les broches TXD(PD)/RXD(PD0).
- **Comparateur Analogique :** broches AIN0(PD6) et AIN1 (PD7) peuvent déclencher l'interruption.
- **Watchdog Timer programmable :** l'ATMega possède un compteur dit de chien de garde programmable pour générer des interruptions à la fin de son comptage et il peut être utilisé comme étant un simple compteur.
- **Gestion d'interruptions (24 sources possibles):**
 - ✓ Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3).
 - ✓ Interruption sur changement d'état des broches PCINT0 a PCINT23.
 - ✓ Interruption liées aux Timers 0,1 et 2 (plusieurs causes configurables).
 - ✓ Interruption de fin de conversion ADC.
 - ✓ Interruption du port série USART.
 - ✓ Interruption du bus 12C.

➤ **Raisons du choix de l'Arduino:** Un microcontrôleur d'une autre famille comme PIC 16f887, qui est un puissant microcontrôleur de famille Microchip aurai pu être choisi pour le projet, mais l'arduino Nano a été choisi pour les avantages suivants:

- L'Atmega328 répond au cahier des charges, en effet le nombre des entrées/sorties dont on

aura besoin est de six entrées/sorties sans VCC et GND.

- S'épargner le travail de conception du circuit de configuration du microcontrôleur, de circuit de conversion Port série USB.
- Microcontrôleur préprogrammé avec un bootloader (ISP), donc le programme dédié n'est pas nécessaire.
- Une communauté active avec un site officiel et un forum officiel.
- Environnement de programmation claire, simple et Multiplateforme.

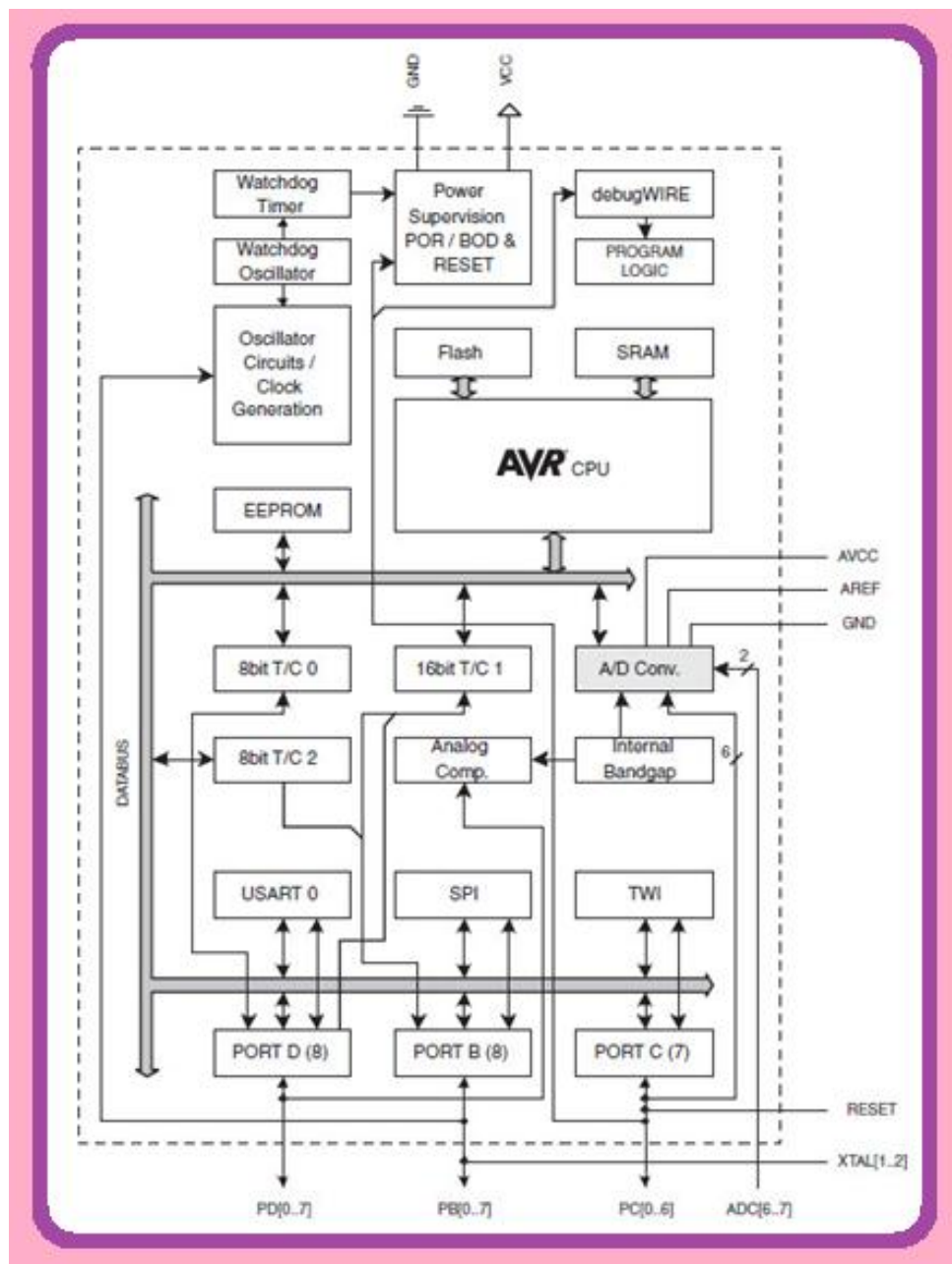


Figure4.4: Architecture interne ATmega328

Présentation du capteur thermique LM35:

Description:

Le capteur de température LM35 est un capteur de température analogique fabriqué par **Texas Instruments**. Il est extrêmement populaire en électronique, car précis, peu coûteux, très simple d'utilisation et d'une fiabilité à toute épreuve. Il est capable de mesurer des températures allant de -55°C à $+150^{\circ}\text{C}$ dans sa version la plus précise et avec le montage adéquat, de quoi mesurer n'importe quelle température.

N.B: Les versions grand public sont capables de mesurer des températures comprises entre -40°C à $+110^{\circ}\text{C}$.

Point positif:

La sortie analogique du capteur est proportionnelle à la température. Il suffit de mesurer la tension en sortie du capteur pour en déduire la température. Chaque degré Celsius correspond à une tension de $+10\text{mV}$. Une des grandes forces du capteur LM35, qui fait sa popularité, c'est sa pré-calibration en sortie d'usine.

Tous les capteurs LM35 sont calibrés en degré Celsius lors de la fabrication. La précision garantie par le fabricant est de $\pm 1^{\circ}\text{C}$ de -10°C à 25°C et $\pm 1.5^{\circ}\text{C}$ de -55°C à 150°C pour la version la moins précise, ce qui est largement suffisant la plupart des applications.

Chaque degré Celsius correspond à 10mV (soit 0.01 Volt) et que la sortie du capteur est (quasi) parfaitement linéaire.

Pour en finir avec les compliments, le capteur LM35 fonctionne avec n'importe quelle tension d'alimentation comprise entre 4 volts à 30 volts , ce qui permet de l'utiliser dans virtuellement n'importe quel montage numérique ou analogique.

Point négatif:

Le capteur LM35 ne fonctionne pas en dessous de 4 volts , donc ne s'utilise pas avec des cartes Arduino 3.3 volts , sans alimentation 5 volts externe.

Le capteur LM35 nécessite une alimentation négative pour mesurer des températures en dessous de 0°C . La sortie est proportionnelle à la température, par conséquent -10°C équivaut à -0.1 volt . De fait, avec une simple alimentation 5v , comme celle des cartes Arduino, il n'est possible de mesurer que des températures positives.

N.B: on verra qu'il ya une astuce pour contourner le problème.

Mesurer une température avec un capteur LM35:

Le but de ce montage sera de tout simplement mesurer la température ambiante de l'atelier et d'envoyer la température en degré Celsius vers l'ordinateur via le câble USB.

Montage:

Pour réaliser ce montage, il va nous falloir:

1. Une carte Arduino UNO (et son câble USB),
2. Un capteur LM35,
3. Un condensateur de 100nF,
4. Une plaque d'essai et des fils pour câbler notre montage.

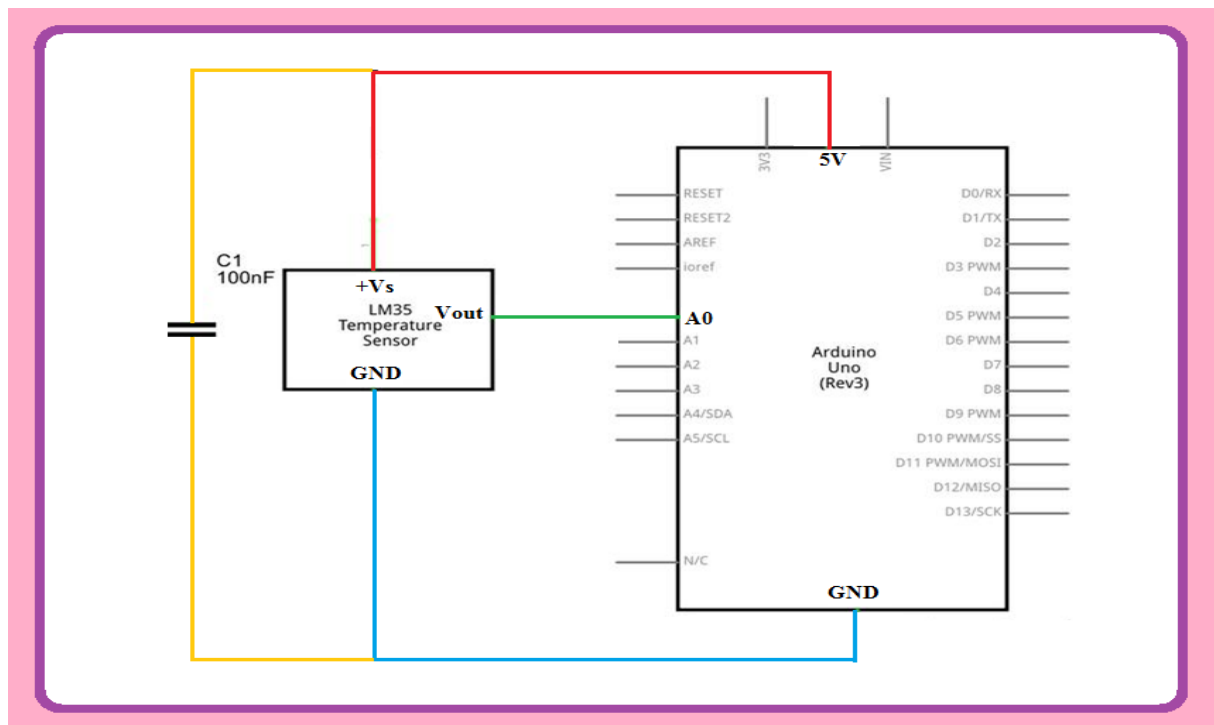


Figure4.5: Schématique du montage pour mesurer une température avec un capteur LM35

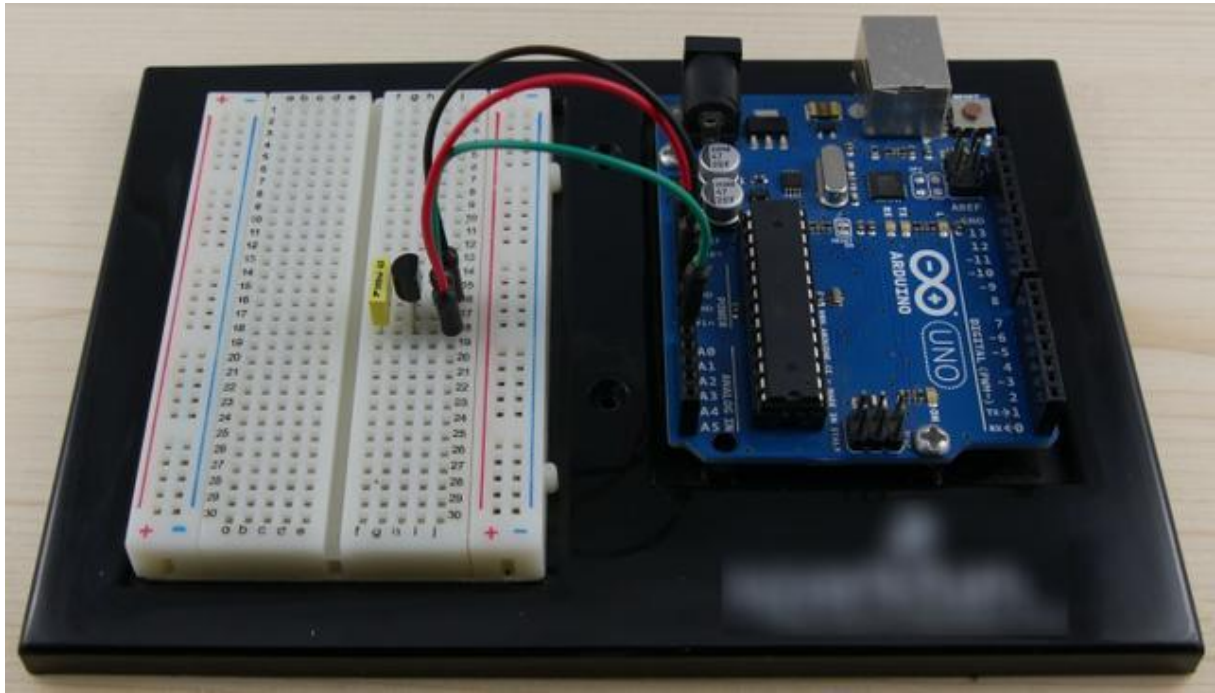


Figure4.7: Montage pour mesurer une température avec un capteur LM35 fini

Le code de démonstration:

Maintenant que nous avons notre montage, passons au code :

Le but de notre code va être de:

1. Lire la tension sur la broche **A0**
2. Convertir la valeur mesurée en une température (pour l'affichage)
3. Envoyer la valeur au PC (pour l'affichage)
4. Recommencer au point 1.

Pour réaliser ce morceau de code, nous allons utiliser la fonction **analogRead()** .

```
1 void setup() {  
2   Serial.begin(9600);  
3 }
```

Nous allons commencer notre programme de démonstration avec la fonction **Serial.begin(9600)** qui va simplement initialiser la communication avec le **PC**.

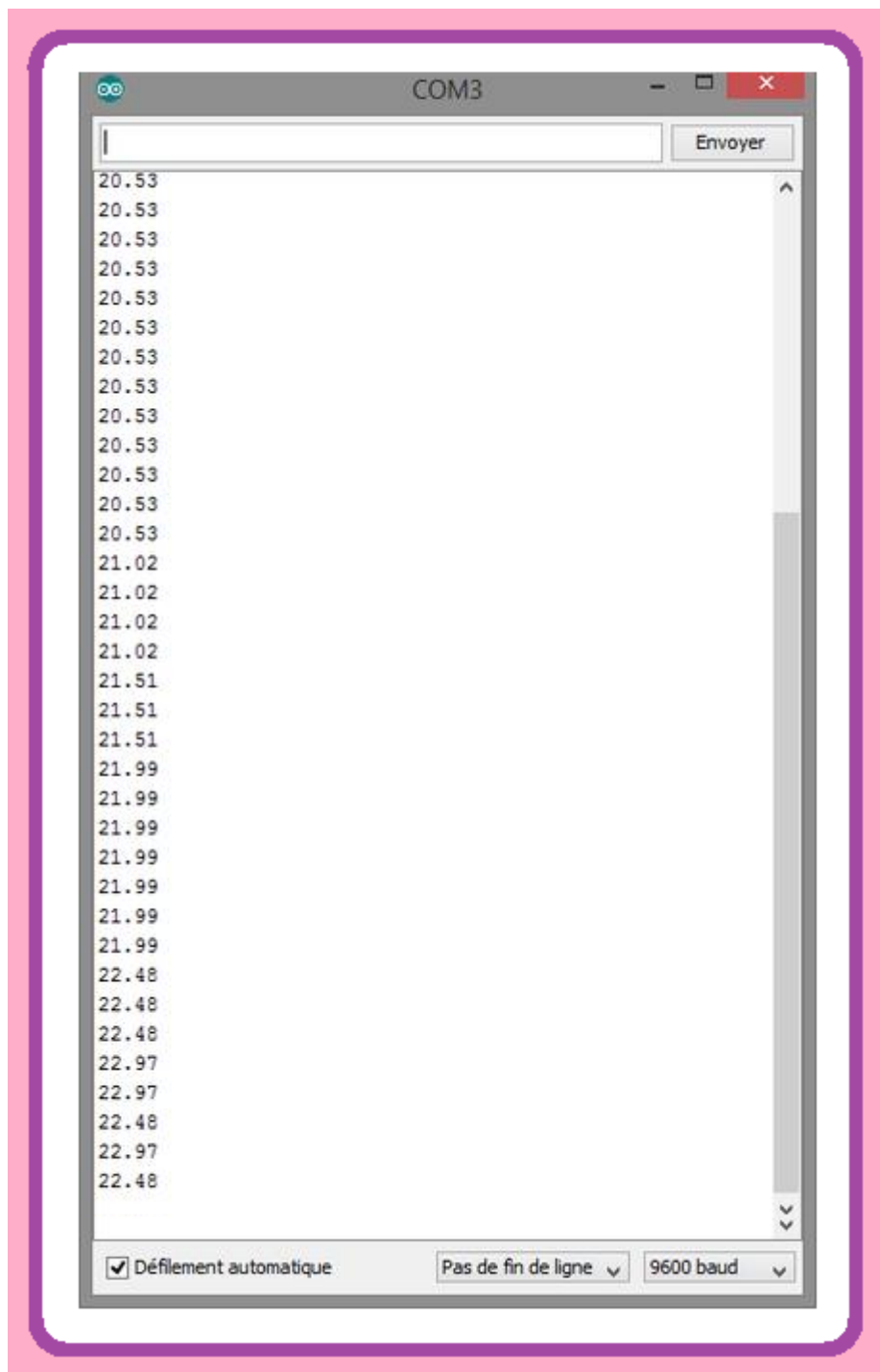
```
1 void loop() {  
2   int valeur_brute = analogRead(A0);  
3  
4   float temperature_celcius = valeur_brute * (5.0 / 1023.0 * 100.0);  
5  
6   Serial.println(temperature_celcius);  
7   delay(250);  
8 }
```

Dans la fonction Loop () nous allons faire trois choses :

1. Mesurer la tension sur la broche **A0** avec **analogRead()**,
2. Transformer le résultat de la mesure en un nombre à virgule (type float) en faisant un simple produit en croix : **5V= 5000mV= 1023** en sortie de **analogRead()**, **10mv= 1°C**, par conséquent, **température= valeur mesurée * (5.0 / 1023.0 * 100.0)**,
3. Envoyer la valeur au **PC** et attendre quelques millisecondes pour avoir le temps de lire ce qui se passe côté **PC**.

```
1  /*
2   * Code d'exemple pour le capteur LM35 (2°C ~ +110°C).
3   */
4
5  // Fonction setup(), appelée au démarrage de la carte Arduino
6  void setup() {
7
8      // Initialise la communication avec le PC
9      Serial.begin(9600);
10 }
11
12 // Fonction loop(), appelée continuellement en boucle tant que la carte Arduino est alimentée
13 void loop() {
14
15     // Mesure la tension sur la broche A0
16     int valeur_brute = analogRead(A0);
17
18     // Transforme la mesure (nombre entier) en température via un produit en croix
19     float temperature_celcius = valeur_brute * (5.0 / 1023.0 * 100.0);
20
21     // Envoi la mesure au PC pour affichage et attends 250ms
22     Serial.println(temperature_celcius);
23     delay(250);
24 }
```


Résultat:



Mesurer des températures négatives:

Il arrive régulièrement qu'on ait besoin de mesurer des températures négatives, tout le monde n'habite pas dans les pays où il fait chaud.

Montage:

Afin de comprendre comment mesurer des températures négatives avec une carte **Arduino**, nous allons reprendre le montage de démonstration précédent et le modifier.

Matériel nécessaire:

Pour réaliser ce second montage, il va nous falloir:

- Une carte **Arduino uno**, un capteur LM35, un condensateur de 100 nF, deux diodes 1N4148, une résistance de 18K ohms, une plaque d'essai et des fils pour câbler notre montage.

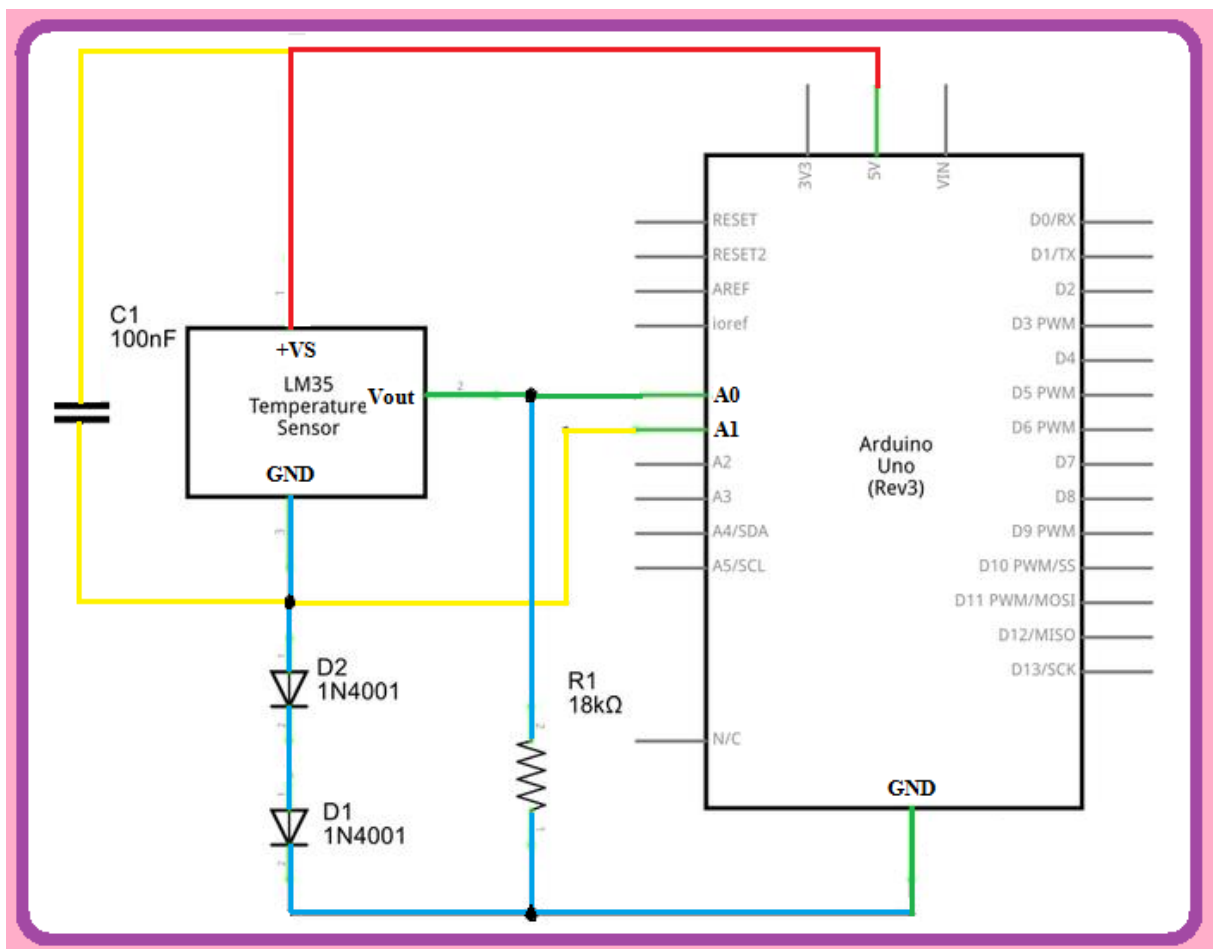


Figure 4.8: Schéma pour mesure de températures négatives avec le capteur LM35

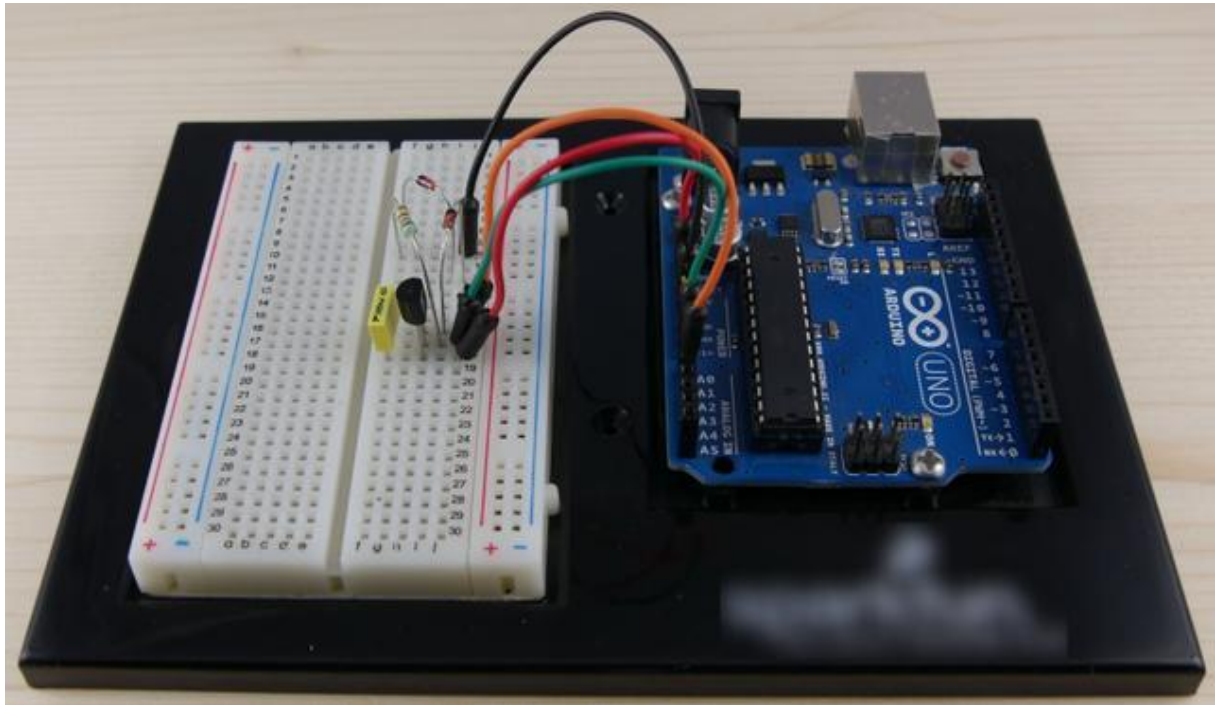


Figure 4.10: Montage pour mesurer une température avec un capteur LM35 finie

Comment ça marche le circuit de mesure de températures négatives ?:

Les deux diodes 1N4148 induisent une chute de tension d'environ 0.6 Volt à leurs bornes chacune (c'est une caractéristique physique des diodes), soit 1.2 volt au total entre la masse réelle du montage et la broche **GND** du capteur.

Avec une masse à 1.2 volt au niveau du capteur, il est possible pour le capteur d'aller en dessous de "son" 0 volt vu qu'il n'est en réalité pas égal à 0 volt, mais à 1.2 volt.

Il y'a cependant un petit souci avec ce montage : la tension aux bornes des diodes varie en fonction de la température, ce qui est embêtant pour un montage censé mesurer des températures précises. C'est un comportement normal pour une diode, c'est même un comportement qui est utilisé pour mesurer des températures, sauf que dans notre cas, c'est un comportement parasite.

L'astuce pour "annuler" cette variation de tension en fonction de la température consiste à mesurer la tension au niveau de la broche **GND** du capteur, puis de corriger la mesure finale en faisant une soustraction dans le code.

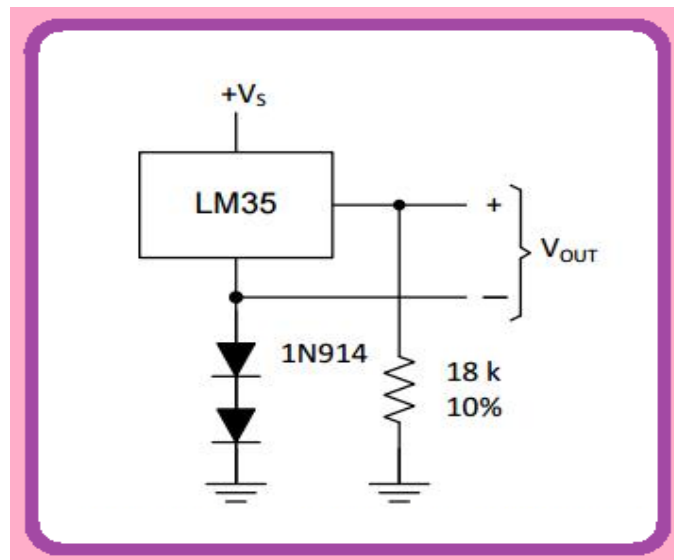


Figure4.11: Schéma de la sonde (-55°C a $+150^{\circ}\text{C}$)

Programme:

Pour réaliser ce second morceau code, nous allons reprendre le code du montage de démonstration du chapitre précédent et lui apporter quelques modifications.

```

1 void setup() {
2     Serial.begin(9600);
3 }

```

La partie setup() reste inchangée.

On notera seulement qu'il n'est pas possible d'utiliser l'astuce **analogReference()**

```

1 void loop() {
2     int valeur_brute = analogRead(A0);
3     int valeur_offset = analogRead(A1);
4
5     float temperature_celcius = (valeur_brute - valeur_offset) * (5.0 / 1023.0 * 100.0);
6
7     Serial.println(temperature_celcius);
8     delay(250);
9 }

```

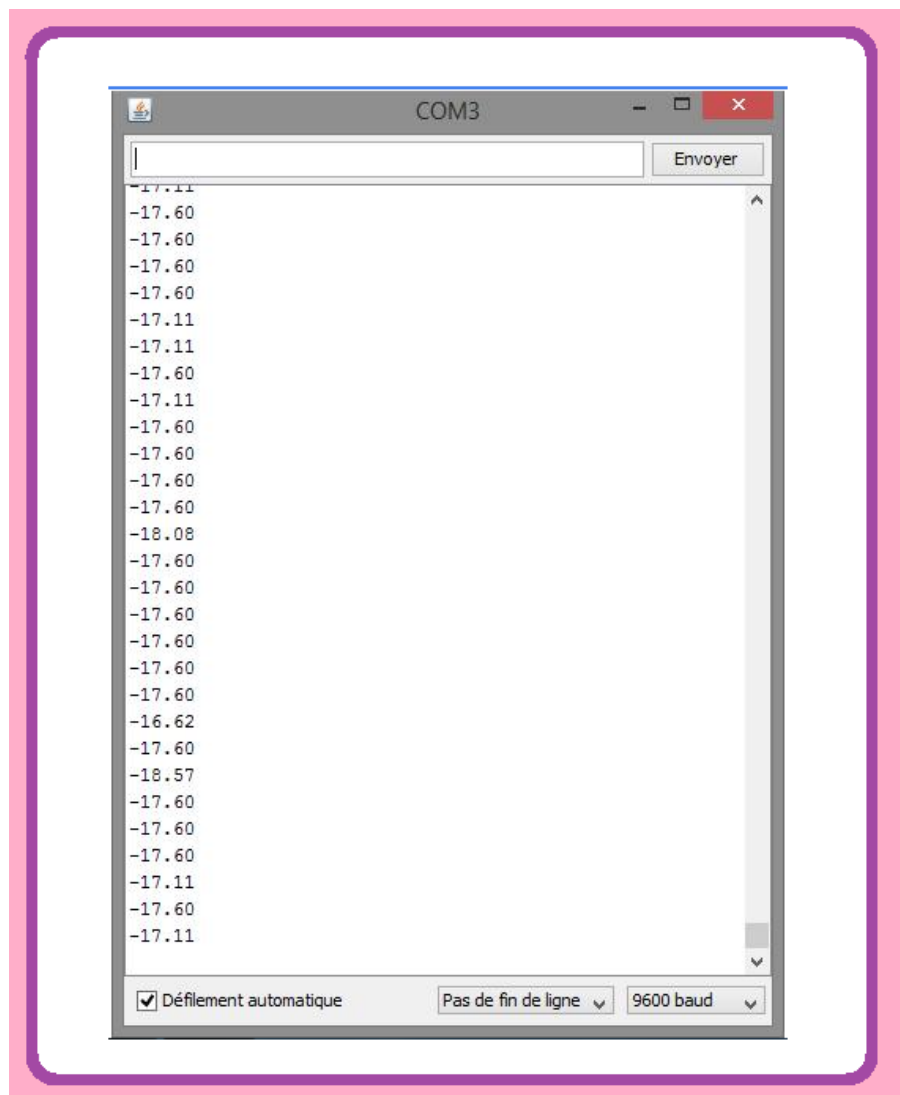
Dans la fonction **Loop()**, nous allons faire deux modifications:

1. Nous allons mesurer la tension sur la broche **A1** **analogRead()** pour savoir de combien la valeur brute doit être compensée.
2. Nous allons modifier le produit en croix pour que la valeur brute soit corrigée.

Le code complet avec commentaires:

```
1  /*
2   * Code d'exemple pour le capteur LM35 avec support des températures négatives (-40°C ~ +110°C).
3   */
4
5  // Fonction setup(), appelée au démarrage de la carte Arduino
6  void setup() {
7
8      // Initialise la communication avec le PC
9      Serial.begin(9600);
10 }
11
12 // Fonction loop(), appelée continuellement en boucle tant que la carte Arduino est alimentée
13 void loop() {
14
15     // Mesure la tension sur la broche A0 (sortie capteur) et A1 (référence du point zéro)
16     int valeur_brute = analogRead(A0);
17     int valeur_offset = analogRead(A1);
18
19     // Transforme la mesure (nombre entier) en température via un produit en croix
20     float temperature_celcius = (valeur_brute - valeur_offset) * (5.0 / 1023.0 * 100.0);
21
22     // Envoi la mesure au PC pour affichage et attends 250ms
23     Serial.println(temperature_celcius);
24     delay(250);
25 }
```

Résultat:



Récepteur infrarouge:

Définition:

Le récepteur **IR TSOP** délivre un signal constant lorsqu'il est actif et qu'il reçoit les données.

Les signaux de commande à distance sont souvent des octets de données qui sont codés et transmis par impulsions (commutation ON et OFF **LED IR** à une fréquence spécifique). La plupart des télécommandes de télévision fonctionnent à des fréquences de 32-40 khz et la plupart des récepteurs peuvent recevoir cette gamme.

Le protocole **SIRC** utilise un codage de largeur d'impulsion de bits. L'impulsion représentant un état logique "1" est une salve de 1.2ms de long de la porteuse de 40 KHz, tandis que la largeur de salve a un état logique "0" est 0.6ms de long. Toutes les salves sont

séparées par un intervalle de temps de 0.6ms espace.



Figure 4.12: Récepteur infrarouge IR TSOP

Exemple de fonctionnement d'un récepteur IR:

Voici un schéma de base de la façon dont les données sont envoyées, chaque fois que vous appuyez sur un bouton sur une télécommande "SONY", il envoie des données de 13Bits, le premier bit est un bit de départ et il y a 12 bits de données qui suivent.

Les 7 bits suivants sont les bits de commande qui varient en fonction des touches pressées. Les 5 derniers bits sont les bits qui seront les mêmes pour tous les boutons d'adresse, mais varient pour les télécommandes de différents appareils

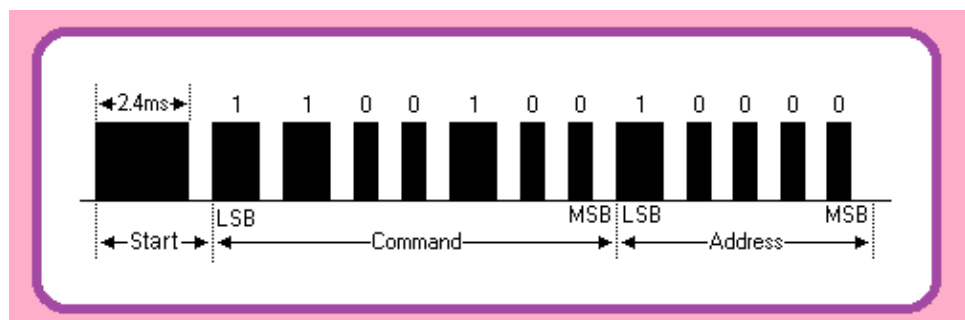


Figure 4.13: Exemple de fonctionnement d'un récepteur IR

Récupération d'un signal:

Nous avons utilisé le port 11 de l'**arduino** mais nous pouvons choisir un autre.

Pour notre montage nous avons besoins:

1. Une carte Arduino
2. un récepteur infrarouge IR type TSOP1736
3. Une télécommande IR

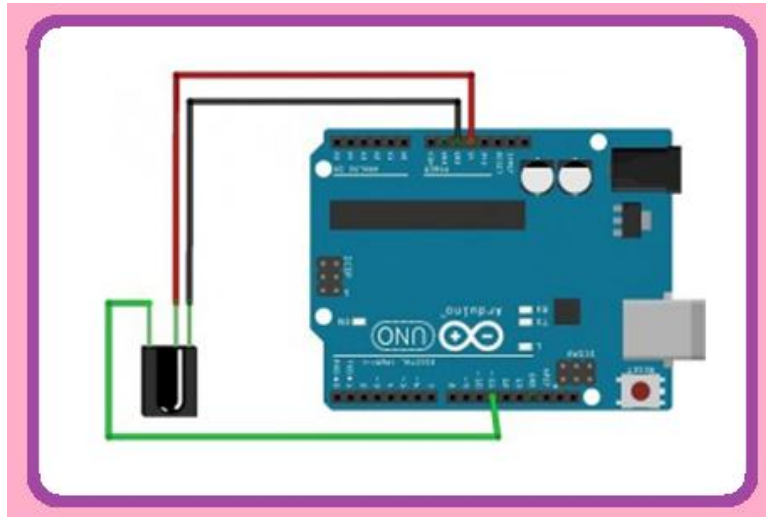
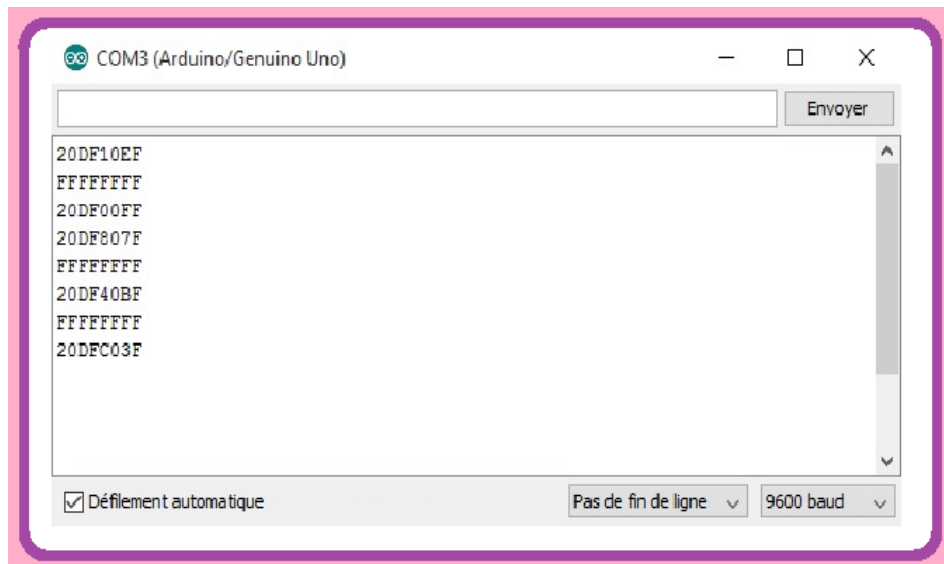


Figure 4.14: Branchement pour réception d'un signal avec récepteur IR

Télé-versez le programme suivant dans votre Arduino:

```
1 #include <IRremote.h>
2
3 int pin_recept = 11; // On définit le pin 11
4 IRrecv ir_recept(pin_recept);
5 decode_results ir_decode; // stockage données reçues
6
7 void setup()
8 {
9   Serial.begin(9600);
10  ir_recept.enableIRIn(); // Initialisation de la réception
11 }
12
13 void loop()
14 {
15   if (ir_recept.decode(&ir_decode))
16   {
17     Serial.println(ir_decode.value, HEX); // On affiche le code en hexadecimal
18     ir_recept.resume();
19   }
20 }
```

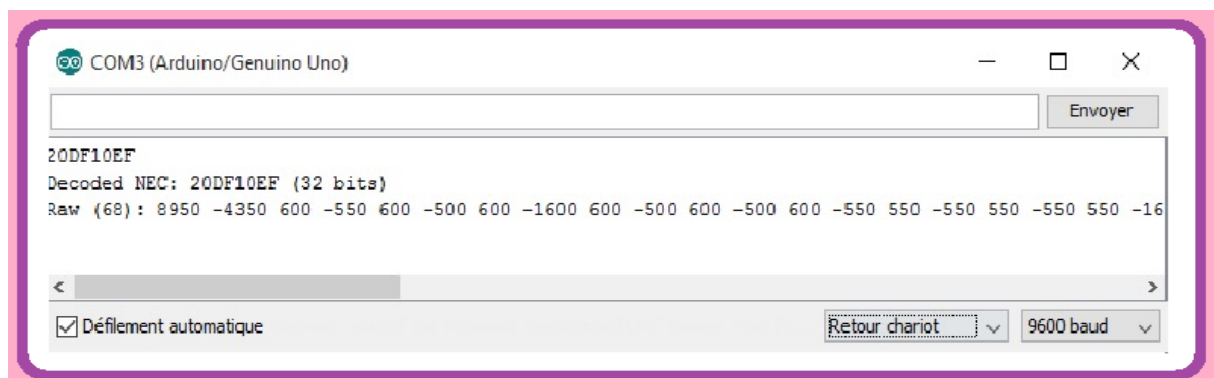
Pour le cas présent, nous avons utilisé la télécommande d'une TV (SONY). Si le protocole est reconnu nous devrions recevoir ce genre de code:



Cela correspond au code **hexadecimal** qui est envoyé par la télécommande. A noter que "FFFFFFFF" est à négliger. Nous pensons qu'il correspond au fait que nous sommes resté trop longtemps sur le bouton. Notons quelques uns de ces codes avec leur correspondance car nous allons les refaire plus tard.

D'ailleurs pour les rejouer nous avons besoin de savoir quel protocole est utilisé par notre télécommande. Pour cela rien de plus simple, il suffit de charger le **sketch Arduino "IRrecvDump"**.

Télé-versions-le dans **Arduino**, ouvrons à nouveau notre moniteur série et appuyons sur une touche



On retrouve le code envoyé et on voit que le protocole utilisé est **NEC**, gardons-le dans un coin car on en aura besoin plus tard.

Emission d'un signal avec une LED:

Pour les branchements il suffit de connecter la LED avec une résistance 100ohms au pin 3.

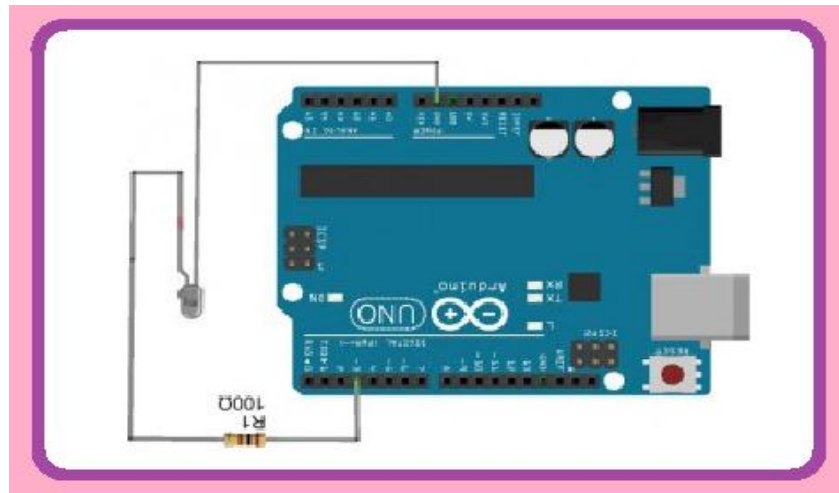


Figure 4.15: Schéma du branchement pour émission du signal avec une LED

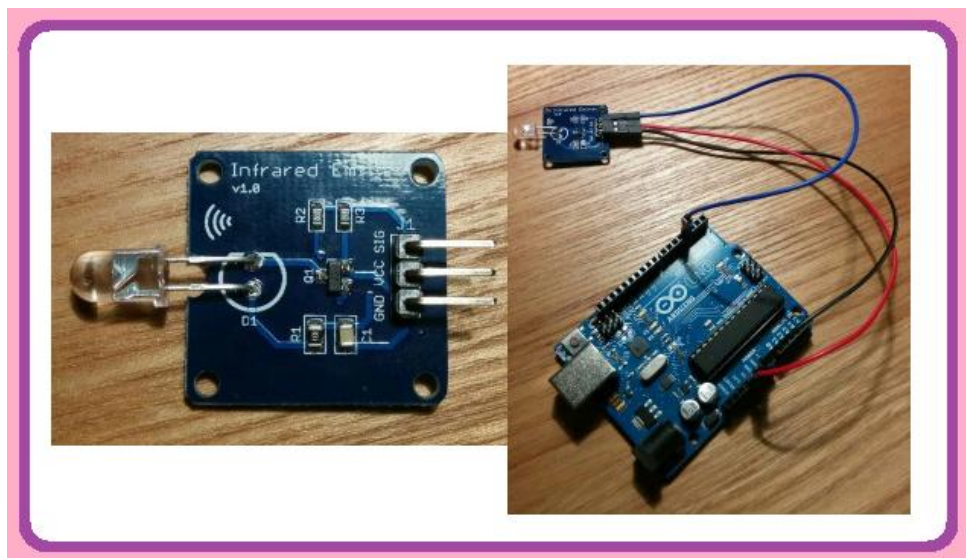


Figure 4.16: Branchement réel pour émission du signal avec une LED

Enfin on charge le programme suivant:



```

1 #include <IRremote.h>
2
3 IRsend ir_send; // crée une instance
4
5 void setup()
6 {
7 }
8
9 void loop()
10 {
11   ir_send.sendNEC(0x20DF10EF, 32); // code télécommande et nombre de bits
12   delay(10000); / On attends 10s avant de rejouer le code
13 }

```

Bluetooth HC-05:

Définition du Bluetooth:

C'est un protocole de communication sans fil. Il a vu le jour à la fin des années 1990 et n'a vraiment percé que dans les années 2000. Il a subi de nombreuses révisions et évolutions.

A l'origine, la société **Ericsson** cherchait à développer un moyen d'utiliser une voie série sans passer par un fil, petit à petit, ses utilisations se sont étendues pour ce que l'on connaît aujourd'hui, un moyen de connecter sans fil de nombreux appareils, allant d'une **Arduino** et sa voie série à un ordinateur ou pour utiliser un casque audio ou encore une manette de jeu sur console de salon .

Ce protocole est un cousin du **Wi-Fi**. En effet, ils respectent tous deux une même spécification **IEEE** et utilisent la même gamme de fréquences : 2.4 GHz (tout comme les téléphones portables par exemple), c'est une communication bidirectionnelle, deux modules peuvent communiquer ensemble en même temps, le comportement utilisé est "maître/esclave" un esclave pourra parler avec un seul maître, mais un maître pourra dialoguer avec plusieurs esclaves. Pour son utilisation elle se passe en plusieurs étapes:

1. Le maître se met en mode "reconnaissable",

2. L'esclave trouve le maître et demande à s'y connecter,
3. Le maître accepte la connexion,
4. Les périphériques sont alors appariés (ou associés),
5. La communication peut commencer.

Ensuite, selon le type de composant que nous utilisons (une oreillette **Bluetooth**, une manette de jeu-vidéo, ...) la communication pourra se faire selon un protocole ou un autre.

Présentation du module Bluetooth HC-05:

Le module HC-05 n'est pas plus gros qu'un pouce. Il est en fait un montage d'un module **Bluetooth** sur un petit **P.C.B**, cela permet de s'affranchir de certaines contraintes comme la soudure du module (qui est très délicate), la conversion 5V -> 3.3V, la régulation de l'alimentation (3.3V de nouveau) ou encore l'ajout de **LED** de signal, tout cela est déjà intégré.

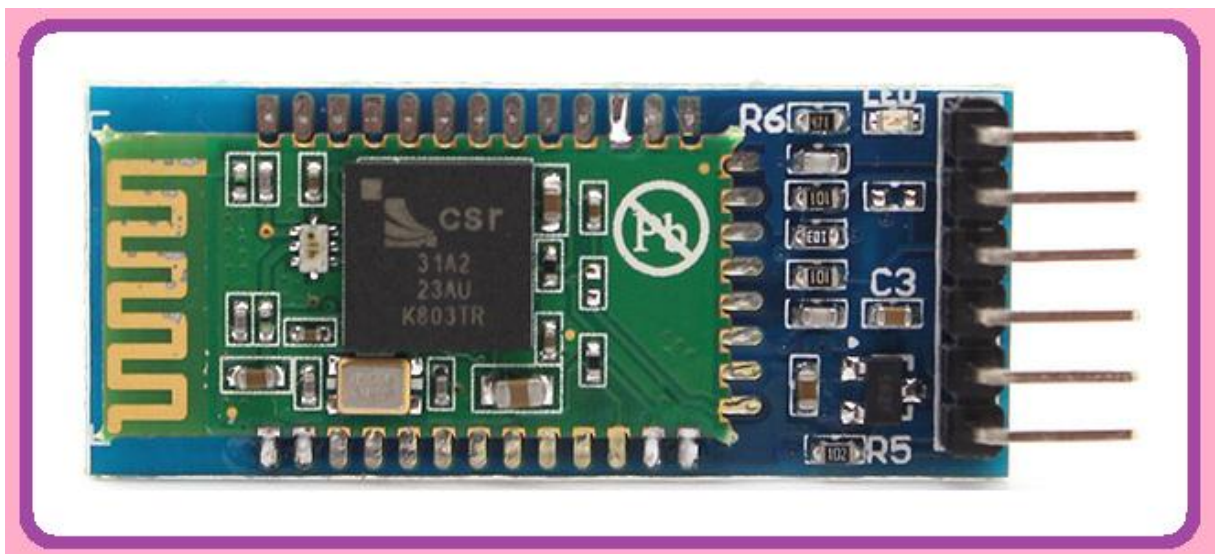


Figure4.17: Module Bluetooth HC-05

Le **HC-05** agit comme un port série à travers lequel vous pouvez envoyer et recevoir des données. Donc en utilisant un terminal série ou une application Bluetooth personnalisée sur votre ordinateur ou téléphone, vous pouvez contrôler et surveiller votre projet.

Nous trouvons sur ce petit module, un ensemble de broches, VCC et GND pour l'alimentation **5V**, **Rx/Tx** pour la communication, on y voit aussi une broche "Key" qui servira à envoyer des commandes de configuration au module, la dernière broche nommée "Led" permet de brancher une LED pour obtenir un signal sur l'état du module.

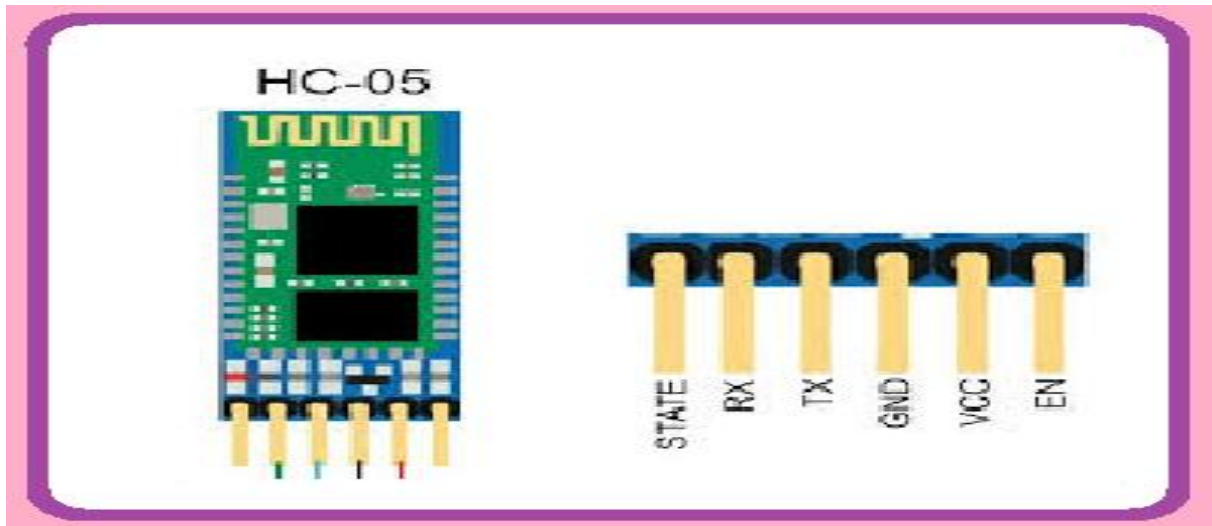


Figure4.18: Schéma module Bluetooth HC-05

Comment brancher le module a l'Arduino:

Le VCC sera relié au 5V de l'**Arduino** et le GND à la masse. Ensuite, viennent les broches de transmission de données, Rx et Tx. Nous pouvons, au choix, les connecter aux broches de liaison série de l'Arduino (0 et 1) pour utiliser la voie série native, ou alors les brancher sur n'importe quelle autre pin afin de garder la voie série disponible pour du debug avec l'ordinateur.

Il ne reste alors que la broches Key et Led. Led sert à connecter une LED de statut. Nous pouvons la laisser déconnecter cela n'influencera pas le comportement du module, Key sert à utiliser le Mode "commande" du module. Avec cette dernière nous pouvons configurer la voie série (vitesse, parité etc...) et d'autres options liées au Bluetooth (nom du module, mot de passe d'appairage, mode esclave ou maitre...). Cette broche est à connecter à n'importe quelle sortie numérique de l'Arduino.

Montage du module HC-05 sur Arduino:

Pour le diviseur de tensions, on pourra utiliser deux résistances :

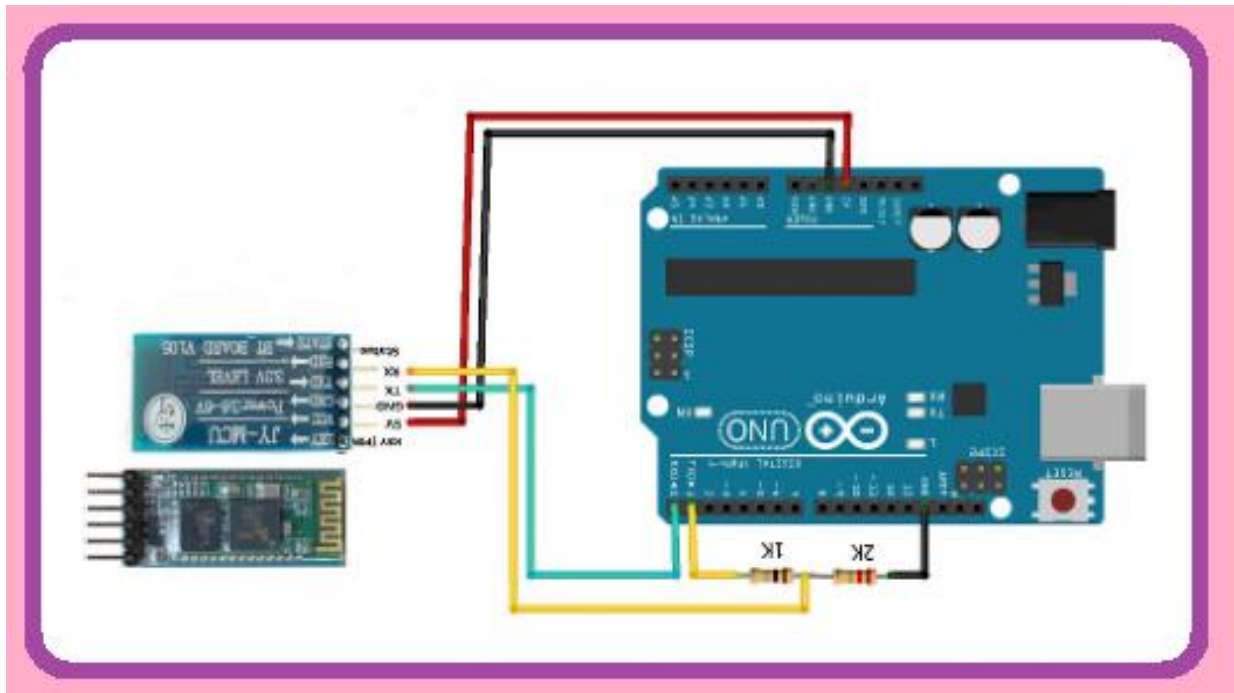


Figure4.19: Schéma de montage du module HC-05 sur Arduino

Comment utiliser HC-05 :

Une seule précaution : Lors des téléchargements des sketches, mieux vaut déconnecter TX et RX pour ne pas créer de conflit avec la liaison série.

Ensuite, c'est très simple, puisque les commandes sont celles de la liaison série, le nom par défaut ressemble ou HC05 et le mot de passe par défaut est 1234.

Avant toute chose, téléchargeons le programme suivant dans lequel nous devrions choisir le nom du module, le mot de passe et la vitesse.

```

char NOMBRE[10] = "BTDENIS01";    // le nom du module
char BPS       = '4';              // 1=1200 , 2=2400, 3=4800, 4=9600, 5=19200,
6=38400, 7=57600, 8=115200
char PASS[10]   = "1234";          // le mot de passe

void setup()
{
    Serial.begin(9600);
    pinMode(13,OUTPUT);
    digitalWrite(13,HIGH);
    delay(10000);
    digitalWrite(13,LOW);

    Serial.print("AT");
    delay(1000);

    Serial.print("AT+NAME");
    Serial.print(NOMBRE);
    delay(1000);

    Serial.print("AT+BAUD");
    Serial.print(BPS);
    delay(1000);

    Serial.print("AT+PIN");
    Serial.print(PASS);
    delay(1000);
}

void loop()
{
    digitalWrite(13, !digitalRead(13));
    delay(500);
}

```

Il reste plus qu'à mettre en route le module et d'attendre quelques secondes que la LED 13 clignote.

Module de commande et d'affichage:

Pour la réalisation de cette partie du projet, nous avons opté pour les composants suivants:

- Afficheur LCD 4*20
- Carte relais huit canaux

- Buzzer
- Carte réseau Arduino

Afficheur LCD 4*20 (JHD 204A)

Définition:

Afficheur LCD 4*20 (**JHD 204A**) : C'est un afficheur à écran Christo-liquide alpha numérique qui a 4 quatre lignes de 20 caractères.



Figure 4.20: Afficheur LCD 4*20 (JHD 204A)

Commande du LCD:

Normalement, pour pouvoir afficher des caractères sur l'écran il faudrait activer individuellement chaque pixel de l'écran. Un caractère est représenté par un bloc de 7*5 pixels, ce qui fait qu'un écran de 20 colonnes et 4 lignes représente un total $20*4*7*5=2800$ pixels.

Décodeur de caractères:

Tout comme il existe un driver vidéo pour notre carte d'ordinateur, il existe un driver "LCD" pour votre afficheur.

Aucun composant ne s'ajoute à notre liste d'achat puisqu'il est intégré dans notre écran. Ce driver va servir à décoder un ensemble "simple" de bits pour afficher un caractère à une position précise ou exécuter des commandes comme déplacer le curseur par exemple. Ce

composant est fabriqué principalement par HITACHI et ce nomme JHD 204A.

Il sert de **décodeur de caractères**. Ainsi, plutôt que de devoir multiplier les signaux pour commander les pixels un à un, il nous suffira d'envoyer des octets de commandes pour lui dire "écris-moi <zéros> à partir de la colonne 3 sur la ligne 1.

Ce composant possède 16 broches que je vais brièvement décrire:

N°	Nom	Rôle
1	VSS	Masse
2	Vdd	+5V
3	V0	Réglage du contraste
4	RS	Sélection du registre (commande ou donnée)
5	R/W	Lecture ou écriture
6	E	Entrée de validation
7 à 14	D0 a D7	Bits de données
15	A	Anode du rétro-éclairage (+5V)
16	K	Cathode rétro-éclairage (masse)

Tableau 2.1: Broches de l'afficheur LCD 4*20 (JHD204A)

Remarque: Normalement, pour tous les écrans LCD (non graphiques) ce brochage est le même, donc pas d'inquiétude lors des branchements

Par la suite, les broches utilisées qu'il faudra relier à **l'Arduino** sont les broches 4, 5 (facultatives), 6 et les données (7 à 14 pouvant être réduite à 8 à 14) en oubliant pas l'alimentation et la broche de réglage du contraste. Ce composant possède tout le système de traitement pour afficher les caractères. Il contient dans sa mémoire le schéma d'allumage des pixels pour afficher chacun d'entre eux.

Voici la table des caractères affichables:

Higher 4bit Lower 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000			0	Q	P	`	F		-	タ	エ	×	ρ
xxxx0001		!	1	A	Q	a	9	μ	7	チ	4	3	q
xxxx0010		"	2	B	R	b	r	「	イ	ツ	×	ρ	θ
xxxx0011		#	3	C	S	c	s	」	ウ	チ	モ	ε	×
xxxx0100		\$	4	D	T	d	t	、	エ	ト	ト	μ	Ω
xxxx0101		%	5	E	U	e	u	・	オ	ナ	ユ	δ	Q
xxxx0110		&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111		'	7	G	W	g	w	フ	チ	ズ	ウ	q	π
xxxx1000		<	8	H	X	h	x	イ	ク	ホ	リ	フ	又
xxxx1001		>	9	I	Y	i	y	ウ	ケ	ル	ル	フ	4
xxxx1010		*	:	J	Z	j	z	エ	コ	ハ	レ	i	*
xxxx1011		+	:	K	L	k	l	オ	サ	ロ	ロ	×	天
xxxx1100		,	<	L	*	1	1	ホ	ミ	フ	フ	Φ	天
xxxx1101		-	=	M	I	m	>	ユ	ズ	ハ	コ	ト	÷
xxxx1110		μ	>	N	^	n	+	ヨ	セ	ホ	ハ	ん	
xxxx1111		/	?	O	_	o	+	ウ	リ	マ	マ	る	■

Tableau 2.2: Caractères affichés de l'afficheur LCD 4*20 (JHD204A)

Communication avec l'écran:

Communication Parallèle:

De manière classique, on communique avec l'écran de manière **Parallèle**, cela signifie que l'on envoie des bits (Par Blocs) en utilisant plusieurs broches en même temps (opposée à une transmission série où les bits sont envoyés un par un sur une seule broche).

Nous utilisons 10 broches différentes, 8 pour les données en parallèle et 2 pour de la

commande (E: Enable et RS: Register Selecter). La ligne R/W peut être connectée à la masse si l'on souhaite uniquement faire de l'écriture.

Pour envoyer des données sur l'écran, il suffit de suivre un ordre logique et un certain timing. Tout d'abord, il faut placer la broche RS à 1 ou à 0 selon que l'on veut envoyer une commande, comme par exemple : "déplacer le curseur à la position (1, 1)" ou que l'on veut envoyer une donnée : "écris le caractère 'a' ".

Ensuite, on place sur les 8 broches de données (D0 à D7) la valeur de donnée à afficher.

Enfin, il suffit de faire une impulsion d'au moins 450 ns pour indiquer à l'écran que les données sont prêtes. Cependant, la communication parallèle prend beaucoup de broches, il existe un autre mode "semi-parallèle". Ce dernier se contente de travailler avec seulement les broches de données D4 à D7 (en plus de RS et E) et il faudra mettre les quatre autres (D0 à D3) à la masse.

Il libère donc quatre broches. Dans ce mode, on fera donc deux fois le cycle "envoi des données puis impulsion sur E" pour envoyer un octet complet.

Communication série:

Lorsque l'on ne possède que très peu de broches disponibles sur notre **Arduino**, il peut être intéressant de faire appel à un composant permettant de communiquer par voie série avec l'écran.

Un tel composant se chargera de faire la conversion entre les données envoyées sur la voie série et ce qu'il faut afficher sur l'écran. Le gros avantage de cette solution est qu'elle nécessite seulement un seul fil de donnée (avec une masse et le V_{CC}) pour fonctionner là où les autres méthodes ont besoin de presque une dizaine de broches.

Comment on s'en sert ?

Branchement:

L'afficheur LCD utilise 6 à 10 broches de données (D0 à D7) ou (D4 à D7) + RS + E) et deux alimentations (+5V et masse). La plupart des écrans possèdent aussi une entrée analogique pour régler le contraste des caractères. Les 10 broches de données peuvent être placées sur n'importe quelles entrées/sorties numériques de **l'Arduino**, En effet, nous l'indiquerons ensuite à la librairie **LiquidCrystal**.

Le montage à 8 branches de données:

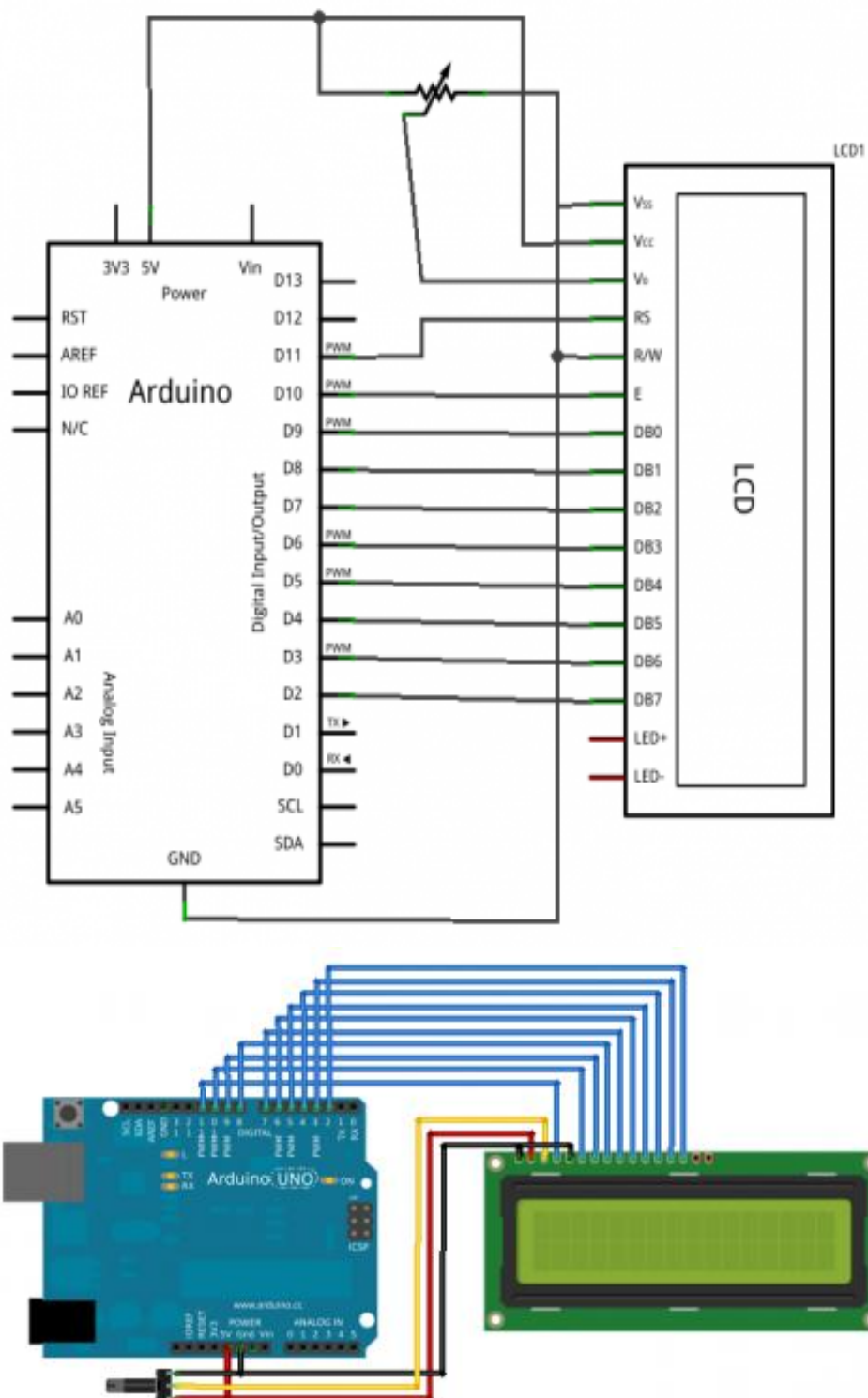


Figure 4.21: Montage à 8 branches de données

Montage à 4 branches de données:

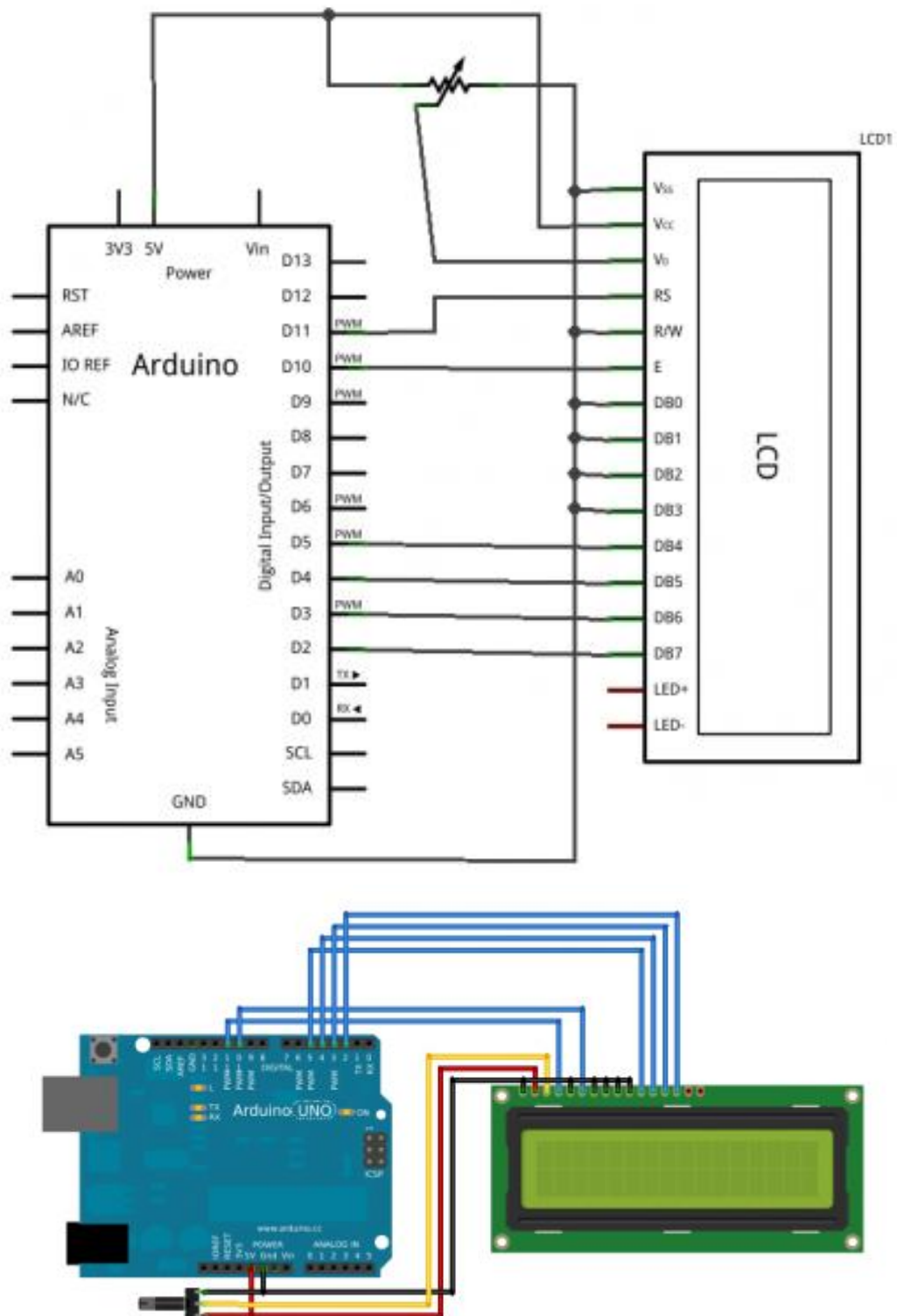


Figure 4.22: Montage à 4 branches de données

Démarrage de l'écran avec Arduino :

Comme écrit plus tôt, nous allons utiliser la librairie "**LiquidCrystal**". Pour l'intégrer, il suffit de cliquer sur le menu "Import Library" et d'aller chercher la bonne. Une ligne `#include "LiquidCrystal.h"` doit apparaître en haut de la page de code.

Ensuite, il ne nous reste plus qu'à dire à notre **Arduino** où est branché l'écran ? (sur quelles broches) et quelle est la taille de ce dernier (nombre de lignes et de colonnes). Nous allons donc commencer par déclarer un objet (c'est en fait une variable évoluée, de type `LiquidCrystal` et qui sera global à notre projet. La déclaration de cette variable possède plusieurs formes

- `LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)` où `rs` est le numéro de la broche où est branché "RS", "enable" est la broche "E" et ainsi de suite pour les données.
- `LiquidCrystal(rs, enable, d4, d5, d6, d7)` ou `rs` est le numéro de la broche où est branché "RS", "enable" est la broche "E" et ainsi de suite pour les données.

Ensuite, dans le `setup()` il nous faut démarrer l'écran en spécifiant son nombre de colonnes puis de lignes. cela se fait grâce à la fonction `begin(cols,rows)`.

Voici un exemple complet de code correspondant au deux branchements.

```
1 #include "LiquidCrystal.h" //ajout de la librairie
2
3 //Vérifier les broches !
4 LiquidCrystal lcd(11,10,9,8,7,6,5,4,3,2); //liaison 8 bits de données
5 LiquidCrystal lcd(11,10,5,4,3,2); //liaison 4 bits de données
6
7 void setup()
8 {
9     lcd.begin(16,2); //utilisation d'un écran 16 colonnes et 2 lignes
10    lcd.write("Salut les Zeros !"); //petit test pour vérifier que tout marche
11 }
12
13 void loop() {}
```

Carte relais huit canaux:

Définition:

Une **Carte relais** est un organe électrique permettant de dissocier la partie puissance de la partie commande. Elle permet l'ouverture et la fermeture d'un circuit électrique par un second circuit complètement isolé.

Pourquoi avoir envie de piloter un relais:

La tension sur les pins **I/O** de l'**Arduino** est de **5V**, un peu limite quand on sait que la tension de l'appareillage électrique de la maison est de **240V**, voir **400V** si vous disposez du triphasé.

Les relais sont utilisés très régulièrement, par exemple pour démarrer un moteur, commander une prise, une lumière, réguler le chauffage

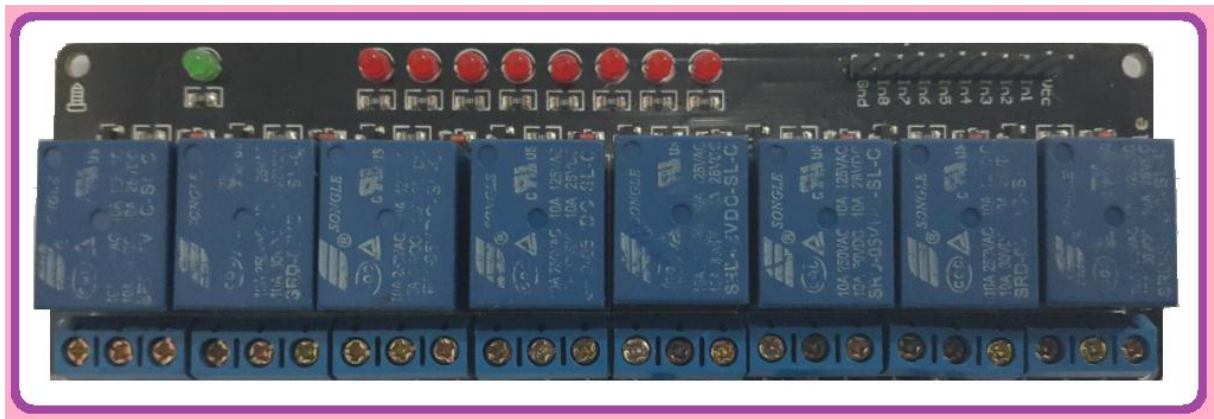


Figure 4.23: Relais huit canaux

Relais et l'arduino:

Une bobine de relais demande un certain ampérage, parfois important. Les sorties **Arduino** sont données à 40 mA maxi, voir 20 mA. Donc on ne pilotera pas directement la bobine avec la sortie sous peine de griller notre carte. Il ya deux possibilités, soit nous utilisons une alimentation extérieure pour piloter la carte, soit nous utilisons le pin **VIN**.

Tout dépend de ce que nous voulons faire. Si nous avons plusieurs relais, il faudrait surement utiliser une alimentation extérieure, si vous en avons un faible nombre, nous pourrons probablement utiliser le pin **VIN**.

Exemple pour faire fonctionne une carte relais:

Nous avons pris une carte relais à quatre canaux pour nos tests. Après, si cela fonctionne on pourra appliquer facilement à la carte un relais à huit canaux. Nous avons réalisé un petit montage simple avec 4 **LEDS** en sortie des 4 **relais** dont voici le montage pratique :

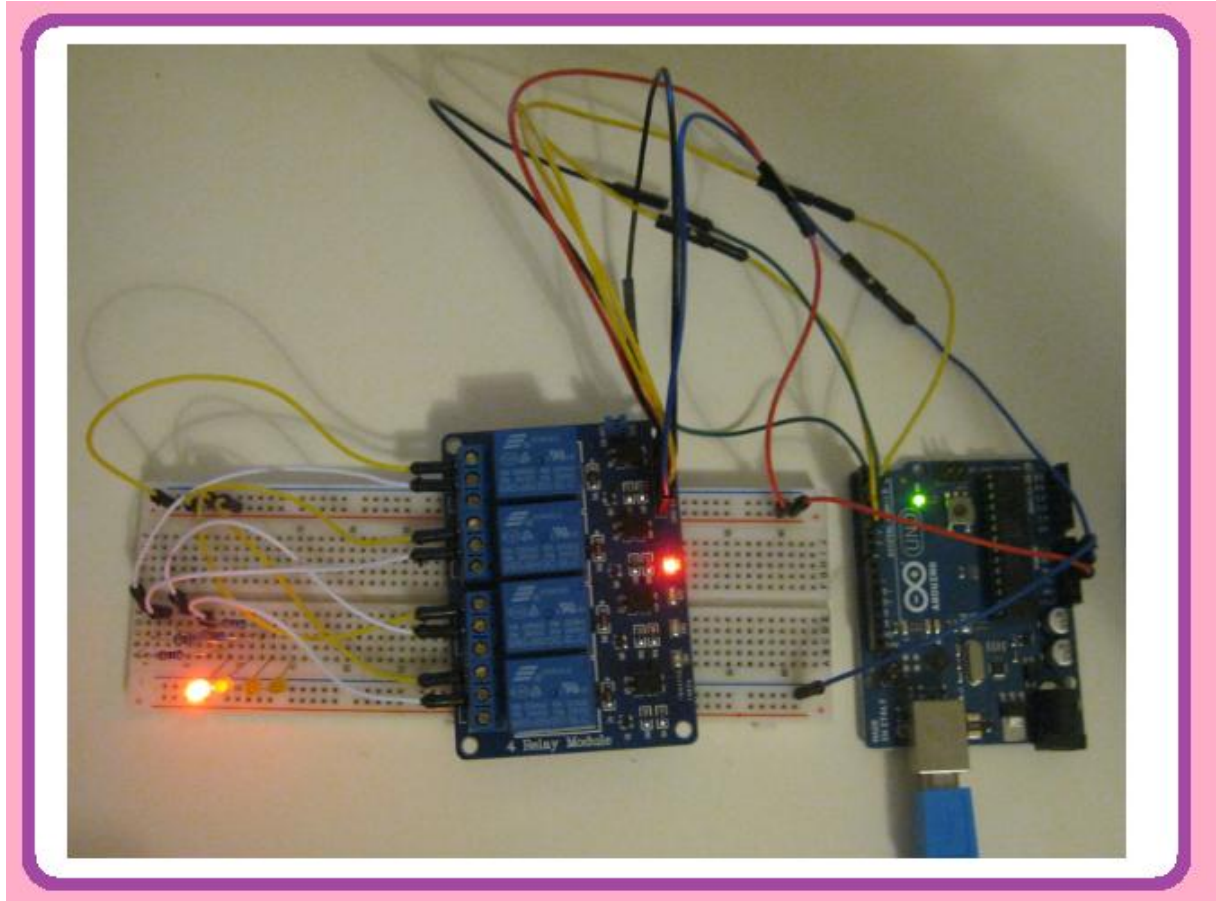


Figure 4.24: Montage teste pour le fonctionnement de la carte relais à quatre canaux.

Programme pour commande la carte relais:

```
#define RELAY_1 2
#define RELAY_2 3
#define RELAY_3 4
#define RELAY_4 5

void setup() {
    // on initialise les sorties et on éteind les leds connectées
    pinMode(RELAY_1, OUTPUT);
    digitalWrite(RELAY_1, HIGH);
    pinMode(RELAY_2, OUTPUT);
    digitalWrite(RELAY_2, HIGH);
    pinMode(RELAY_3, OUTPUT);
    digitalWrite(RELAY_3, HIGH);
    pinMode(RELAY_4, OUTPUT);
    digitalWrite(RELAY_4, HIGH);
}

void loop() {
    // on allume la led de chaque relai chacune à son tour
    pulseRelay(RELAY_1);
    pulseRelay(RELAY_2);
    pulseRelay(RELAY_3);
    pulseRelay(RELAY_4);
}

void pulseRelay(int pin) {
    // pendant 3s on allume la led du montage
    digitalWrite(pin, LOW);
    delay(3000);
    // et on l'éteind
    digitalWrite(pin, HIGH);
}
```

Buzzer piézo-électrique:

Définition:

C'est un composant électronique qui émet un signal sonore d'une fréquence comprise entre 20 Hz et 20 kHz.



Figure 4.25: Buzzer piézo-électrique

Comment ça marche:

- Connecter le fil noir du buzzer piézo-électrique au 0V (masse ou GND)
- Connecter le fil rouge du buzzer piézo-électrique à la carte Arduino

Branchement du buzzer:

L'instruction `tone ()` applique une impulsion carrée de fréquence précise. Les fréquences audibles sont comprises entre 20 Hz et 20.000 Hz. Ce buzzer piézo-électrique transforme l'impulsion électrique en une onde sonore de fréquence identique et audible.

Module serveur WEB:

Carte réseau arduino:

Description:

Elle permet de connecter la carte Arduino à Ethernet. Elle permet d'envoyer et de recevoir des données de partout dans le monde grâce à une connexion internet. Nous pouvons l'utiliser pour contrôler des moteurs et même des robots à partir d'un serveur Web.

Cette carte nous ouvre une quantité infinie de possibilité en nous permettant de connecter notre projet à l'internet en un temps court.



Figure 4.26 : Carte réseau arduino

Le shield Arduino Ethernet est un module Arduino additionnel aux cartes Uno, Duemilanove ou Mega qui permet de les rendre communicantes sur un réseau filaire Ethernet.

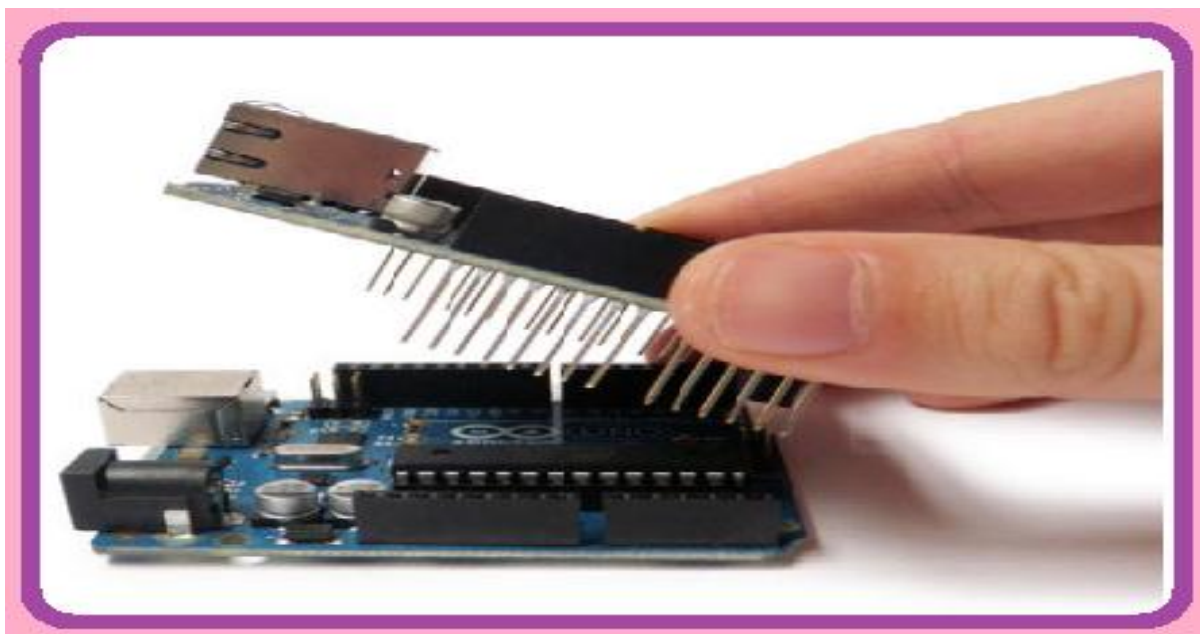


Figure 4.27: Shield Arduino Ethernet additionne a la carte arduino UNO

On pourra ainsi créer un interface Homme Machine pour piloter à distance ou visualiser l'état de notre carte **arduino** en utilisant un câble réseau relié à un box internet.

Caractéristique de la carte Rj45:

La carte contient un certain nombre de LEDs d'information :

PWR : s'allume lorsque l'alimentation de la carte est présente,

LINK : s'allume lorsqu'un réseau est connecté et clignote lors de l'échange de données.

FULLD : s'allume lorsque la connexion réseau est en full duplex,

100M : s'allume lorsque la connexion est en 100Mbps/s,

RX : s'allume lors d'une réception de données,

TX : s'allume lors d'une transmission de donnée,

COLL : s'allume lors d'une collision sur le réseau.

Choix de langage de programmation :

- C Arduino
- App Inventor
- HTML et CSS

C Arduino

Présentation de l'espace développement Intégré (EDI) Arduino:

Comme cela a été dit, la carte Arduino présente le noyau de notre système. Elle est programmée par un logiciel compatible appelé **Arduino EDI** (Espace de développement intégrée) qui est une application écrite en Java inspiré du langage Processing.

L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte.

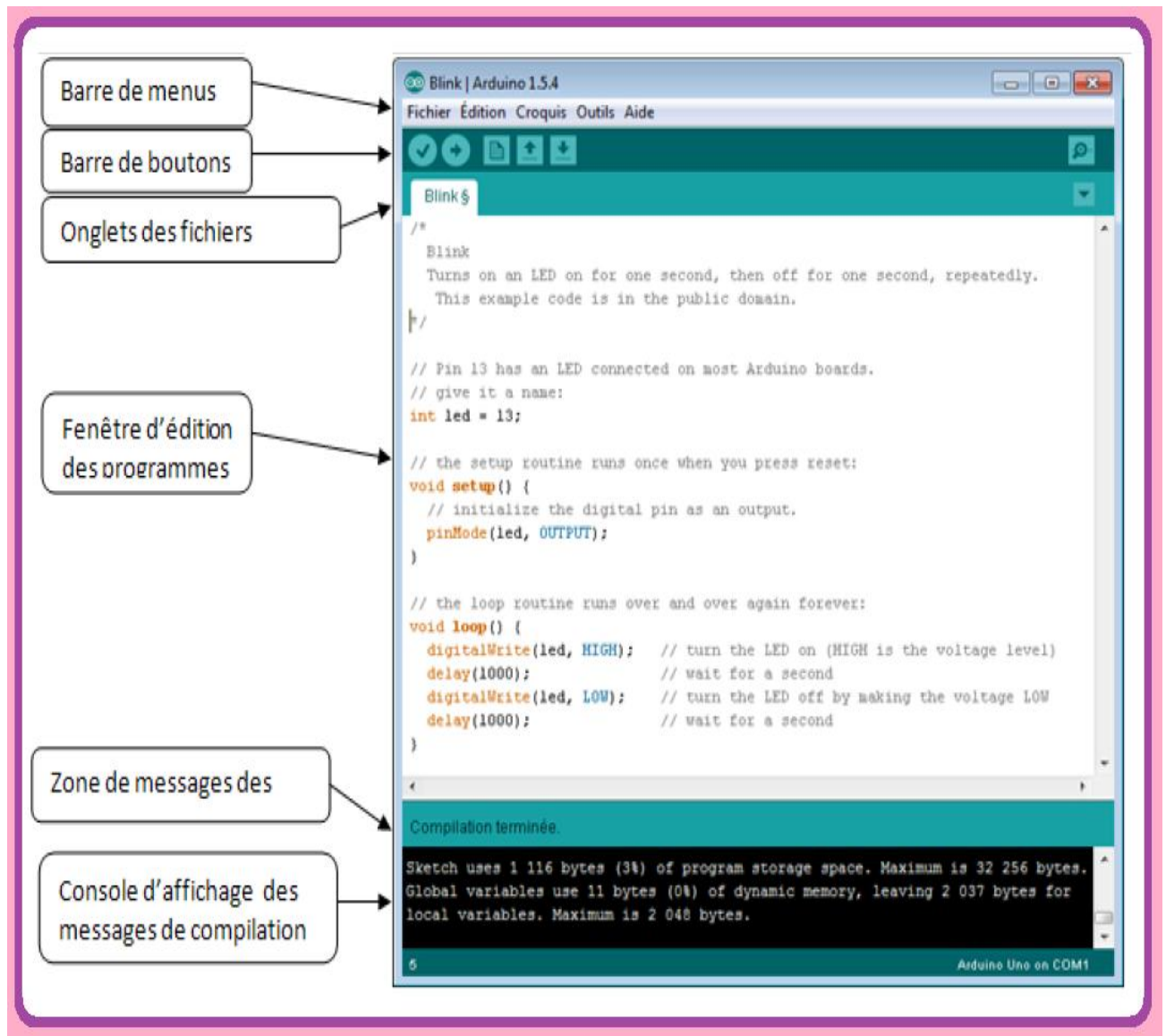
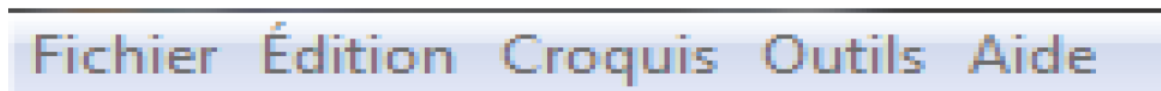


Figure 4.28: Les différentes parties de la fenêtre principale de logiciel Arduino.

Différents compositions de l'espace de développement intégré (EDI):

Une barre de menus:






Cette barre contient les icônes suivantes:

- ✓ **File (Fichier):** ce menu contient les différentes options de créations d'ouverture de sauvegarde, d'impression du programme, ou l'ouverture d'un exemple parmi les exemples qui accompagnent le logiciel Arduino.

- ✓ **Edite (Editor)** : ce menu contient les options de copier, coller et sélectionner les options de recherche.
- ✓ **Sketch (programme ou séquence)** : ce menu contient les différentes fonctions de la barre des boutons, ainsi que les options d'ajouts de librairies ou de fichiers.
- ✓ **Tools (outils)** : c'est dans ce menu qu'on sélectionne le type de carte à programmer, et le port série utilisé.
- ✓ **Help (Aide)** : ce menu est fait pour donner de l'aide concernant les différents problèmes rencontrés au niveau de logiciel Arduino.



Figure 4.29: Barre de boutons pour le programme C Arduino

- ✓ **Vérifier/compiler** : Vérifie le code a la recherche d'erreur,
- ➔ **Transférer vers la carte:** compile votre code et le transférer vers la carte Arduino,
-  **Nouveau:** Crée un nouveau code (ouvre une nouvelle fenêtre),
-  **Ouvrir:** ouvre la liste de tous les programmes dans le "livre de programmes",
Enregistrer: Enregistre un programme,
-  **Moniteur série:** Ouvre la fenêtre du moniteur (on terminal) série.

Structure d'un programme Arduino:

- ✓ **Un éditeur:**
C'est l'espace utilisé pour écrire le programme à réaliser, comme il dispose aussi des onglets de navigation.
- ✓ **Une zone de message:**
Comme son nom l'indique, cet espace est utilisé par l'EDI pour communiquer à l'utilisateur des informations sur l'état du programme et des actions en cours.
- ✓ **Une console texte:**
Affiche le message concernant le résultat de la compilation du programme (il indique si le programme comporte des erreurs).

Programme Arduino:

Description de la structure d'un programme Arduino:

Le langage Arduino est basé sur le langage C/C++ :

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne, la carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code, comme lors d'une programmation classique. Cette structure se décompose en trois parties :

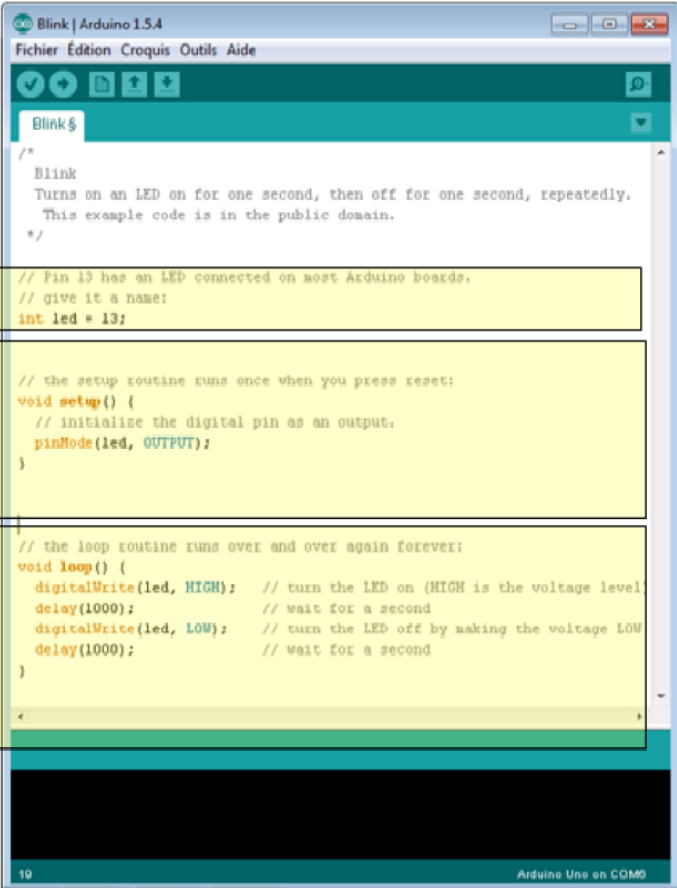
	
Définition des constantes et variables globales	<pre>// Pin 13 has an LED connected on most Arduino boards, // give it a name: int led = 13;</pre>
Fonction principale : <i>VOID SETUP()</i> Initialisation des ressources de la carte Configuration des entrées/sorties Définition de la vitesse de fonctionnement	<pre>// the setup routine runs once when you press reset: void setup() { // initialize the digital pin as an output. pinMode(led, OUTPUT); }</pre>
Fonction boucle : <i>VOID LOOP()</i> Description du fonctionnement général du programme Gestion des interactions entre les entrées/sorties Cette partie sera exécutée en boucle	<pre>// the loop routine runs over and over again forever: void loop() { digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level) delay(1000); // wait for a second digitalWrite(led, LOW); // turn the LED off by making the voltage LOW delay(1000); // wait for a second }</pre>

Figure 4.30: Différentes composantes du programme Arduino

Le code minimal:

Avec Arduino, nous devons utiliser un code minimal lorsque l'on crée un programme. Ce code permet de diviser le programme que nous allons créer en deux parties, soit void setup() ou void loop() décrites en haut.

Jeux d'instructions du langage Arduino:

Syntaxes du Programme:

- Chaque instruction se termine par un ";",
- Les **accolades** "{" et "}" sont les "conteneurs" du code du programme, elles sont propres aux fonctions, aux conditions et aux boucles. Les instructions du programme sont écrites à l'intérieur de ces accolades,
- Les **commentaires** sont des lignes de texte incluses dans le programme et qui ont pour but de nous informer de la façon dont le programme fonctionne. Ces lignes ajoutées sont ignorées par le compilateur. Les commentaires sont précédés des caractères "//" ou bien encadrés par " /* " et " */ ".
- Il est formellement interdit de mettre des accents en programmation, sauf dans les commentaires,
- Un nombre en binaire doit être précédé de "B",
- un nombre en hexadécimal doit être précédé par les caractères "0x".

Commandes du programme Arduino:

❖ Structure générale

- **void setup()** (configuration-préparation)
- **void loop()** (exécution)

❖ Contrôle et conditions

- **if** (si...)
- **if...else** (si...alors...)
- **for** (pour...)
- **switch case** (dans le cas ou...)
- **while** (pendant que...)

❖ Opérations de comparaison

- **=** (équivalent à)
- **!=** (différent de)
- **<=** (inférieur ou égal à)
- **>=** (supérieur ou égal à)

❖ Opérations booléennes

- **&** (Et)
- **!** (et pas)

- `||` = (ou)

❖ Autres commandes

- `//` = (Commentaire simple ligne)
- `/**/` = (Commentaire multi-lignes),
- `#define` = (donner une valeur à un nom)

❖ Variables

- `char` = (variable 'caractère')
- `int` = (variable 'nombre entier')
- `long` = (variable 'nombre entier de très grande taille')
- `string` = (variable 'chaîne de caractères')
- `array` = (tableau de variables)

Variables:

❖ Niveaux logiques des connecteurs numériques

- `HIGH` = (état 1),
- `LOW` = (état 0),
- `INPUT` = (configuré en entrée),
- `OUTPUT` = (configuré en sortie).

Fonctions:

❖ Entrées-sorties numériques

- `pinMode(broche, état)` (configuration des broches en entrée ou en sortie),
- `digitalWrite(broche, état)` (écrire un état sur une broche numérique),
- `digitalRead(broche)` (lire un état sur une broche numérique),
- `unsigned long pulseIn (broche, état)` (lire une impulsion sur broche numérique).

❖ Entrées analogiques

- `int analogRead(broche)` (lire la valeur d'une broche analogique)
- `analogWrite(broche, valeur)` (PWM: écrire une valeur analogique sur les broches 5, 6, 9, 10 et 11)

❖ Gestion du temps

- `unsigned long millis()` (temps de fonctionnement du programme),
- `delay(ms)` (attente, en millisecondes),

- **delayMicroseconds(us)** (attente, en microsecondes).

Étapes à suivre pour programmer la carte:

Ecriture du programme:

Cette première étape consiste à saisir le programme.

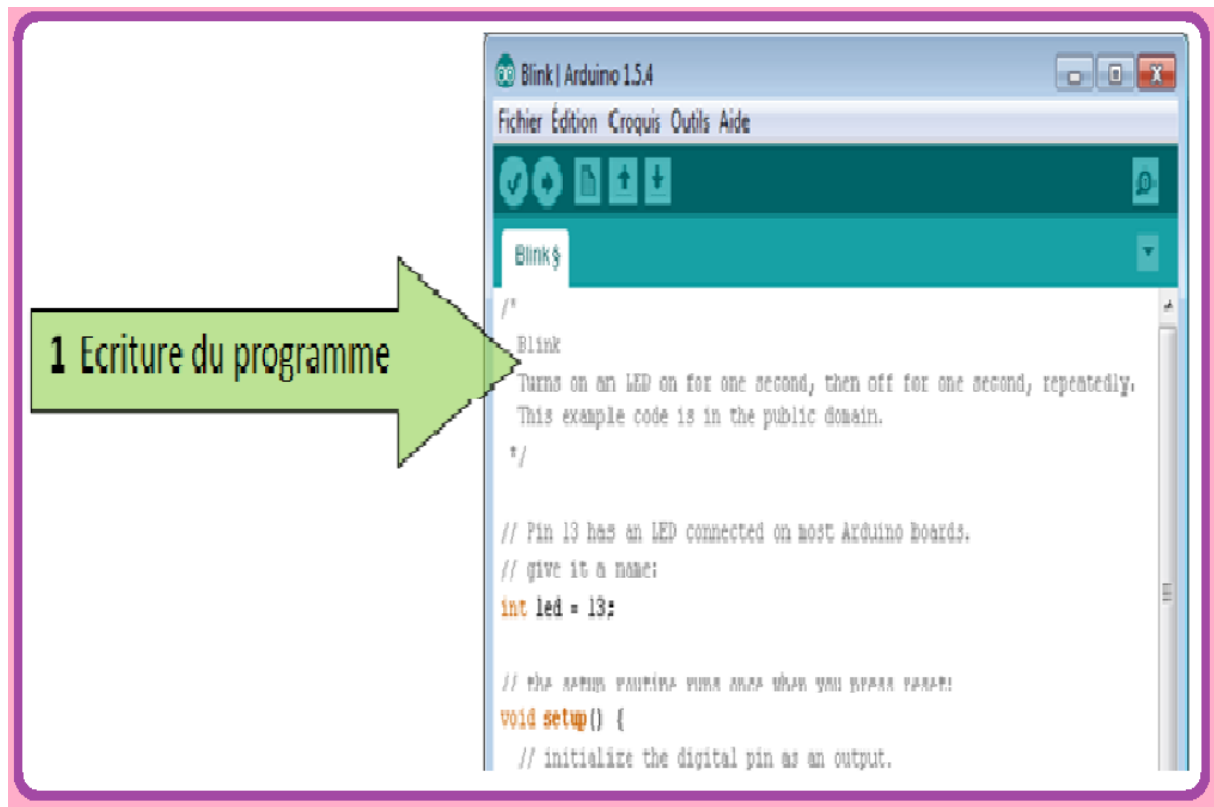


Figure 4.31: L'écriture de programme

Compilation du programme:

Dans cette partie, on vérifie si le code contient des erreurs de syntaxes, en cas d'anomalie dans le programme, le compilateur nous renseigne sur le type d'erreur et la ligne ou elle se trouve, (Pour la correction des erreurs il suffit de consulter l'aide disponible dans la barre de menus).

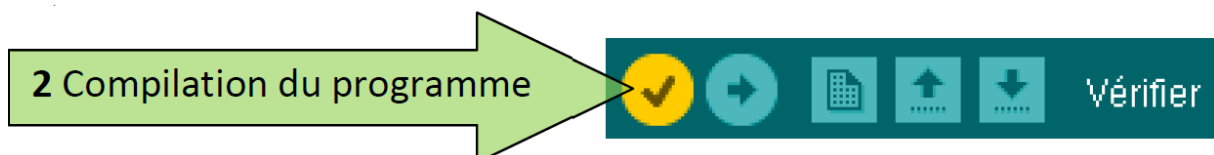


Figure 4.32: compilation du programme

Si aucune erreur n'est détectée, le programme est compilé et le message "Compilation terminée" apparaît, suivi de taille du programme.

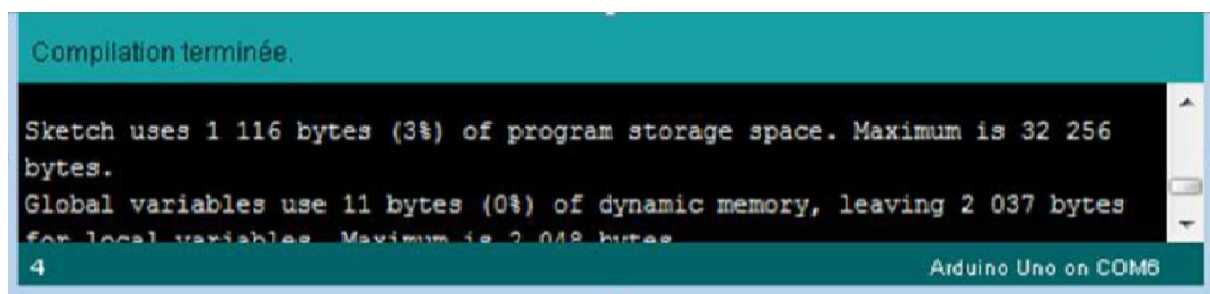


Figure 4.33: Console d'affichage

Sélection de la cible (type de carte):

Avant de transférer le programme vers la carte Arduino, il faut sélectionner le type Arduino (cible) depuis le menu **Outils>Board**.

La carte doit évidemment être connectée à l'ordinateur via un câble USB.

Vérifions que c'est bien le nom "Arduino nano" qui est coché, si ce n'est pas le cas, on le coche.

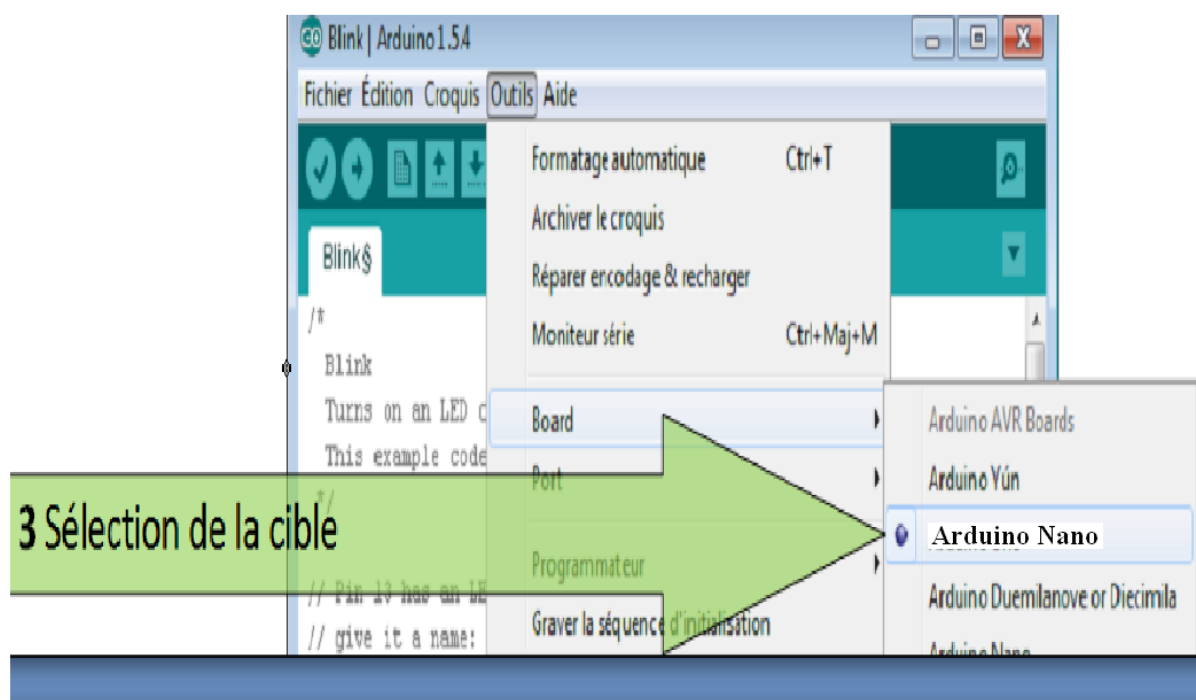


Figure 4.34: Sélection de la cible

Sélection du port:

Sélectionner le port série sur lequel est branché cette carte (depuis le menu **Outils>Port**)

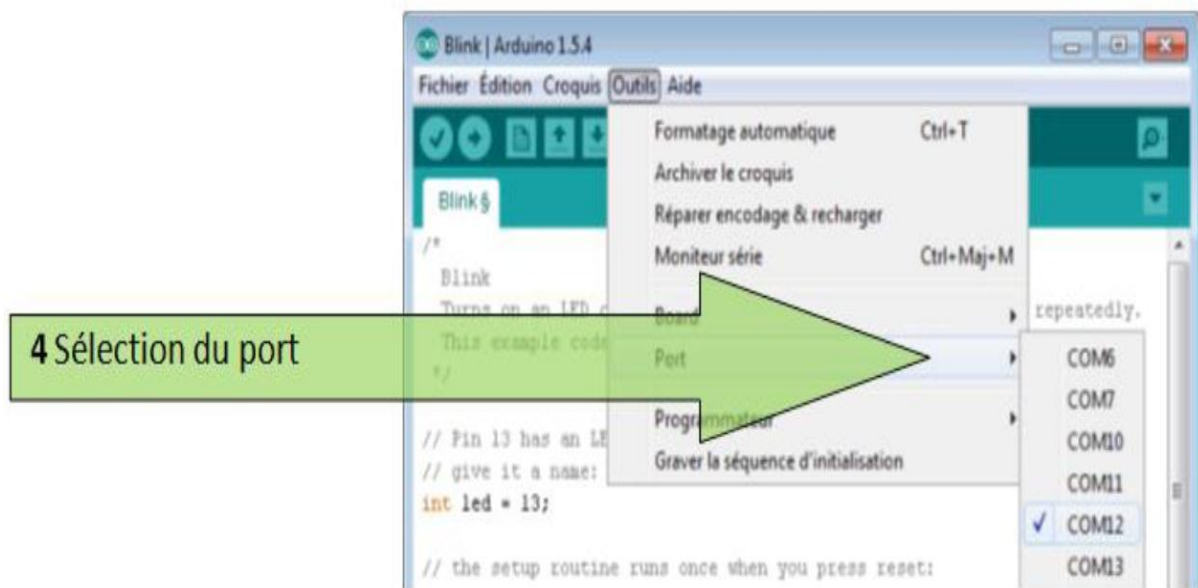


Figure 4.35: Sélection du port

Transfert du programme:

Cela se fait en cliquant sur le bouton.

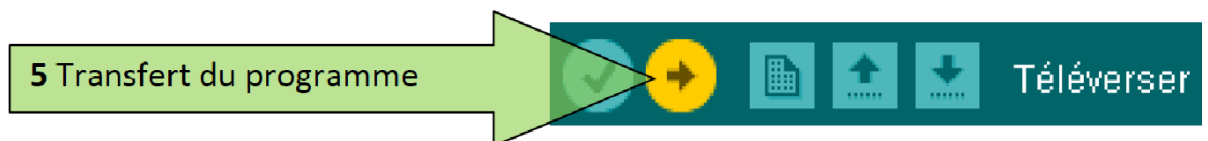


Figure 4.36: Bouton de transfert du programme

Le programme est téléchargé vers la carte, le message "télé versement terminé" s'affiche une fois l'opération terminée.

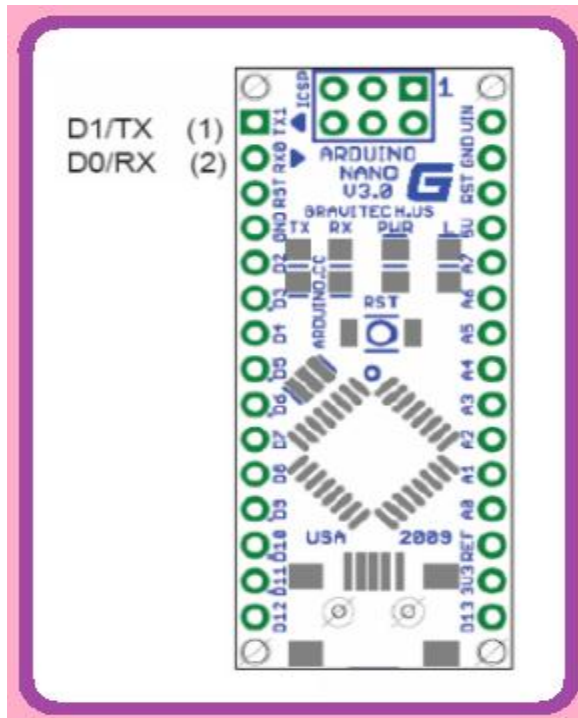


Figure 4.37: Port Rx, Tx communication série

Sur la plupart des cartes, nous devons voir les **LEDs** des lignes **RX** et **TX** clignotées rapidement, témoignant que le programme est bien transféré. Durant le transfert, le bouton devient jaune et le logiciel Arduino affiche un message indiquant que le transfert est en cours.

App Inventor:

Présentation d'Android:

Android est un système d'exploitation libre qui utilise les téléphones mobiles modernes nommés (Smartphones). Il est basé sur linux. Android est détenu par Google. Sur le marché de la téléphonie mobile, Android représente 78% des systèmes d'exploitation des téléphones mobiles.

Avec Android on peut facilement programmer et effectuer des programmes de formules mathématiques, contrôler de moteurs à distance, des jeux, emplacement sur Google Maps, le déplacement d'un robot via Bluetooth, réponse automatique SMS.....

Présentation App inventeur 2:

Est une application développée par Google, elle est actuellement entretenue par le Massachusetts Institute of Technology(MIT).

Elle simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont plus familiers avec les langages de programmation.

Elle est basée sur une interface graphique similaire, grâce à son interface entièrement graphique et à l'absence totale de ligne de code. Elle est particulièrement adaptée à l'initiation des enfants à la programmation, et ce, dès l'école primaire.

La programmation est réalisée sans taper une seule ligne de code, mais simplement en associant et en paramétrant des briques logicielles réalisées.

Google publie l'application le 15 décembre 2010. C'est le centre d'études mobiles au MIT qui gère le support technique de cette application sous le nouveau nom "MIT app Inventor".

A quoi ressemble un programme App inventeur 2:

La programmation se réalise en ligne à l'aide de son navigateur préféré, et sous l'environnement logiciel de son choix (Mac, Linux ou Windows). Seules contraintes, avoir un compte Gmail pour pouvoir y accéder, et un accès à internet évidemment. Les informations sont stockées sur des serveurs distants.

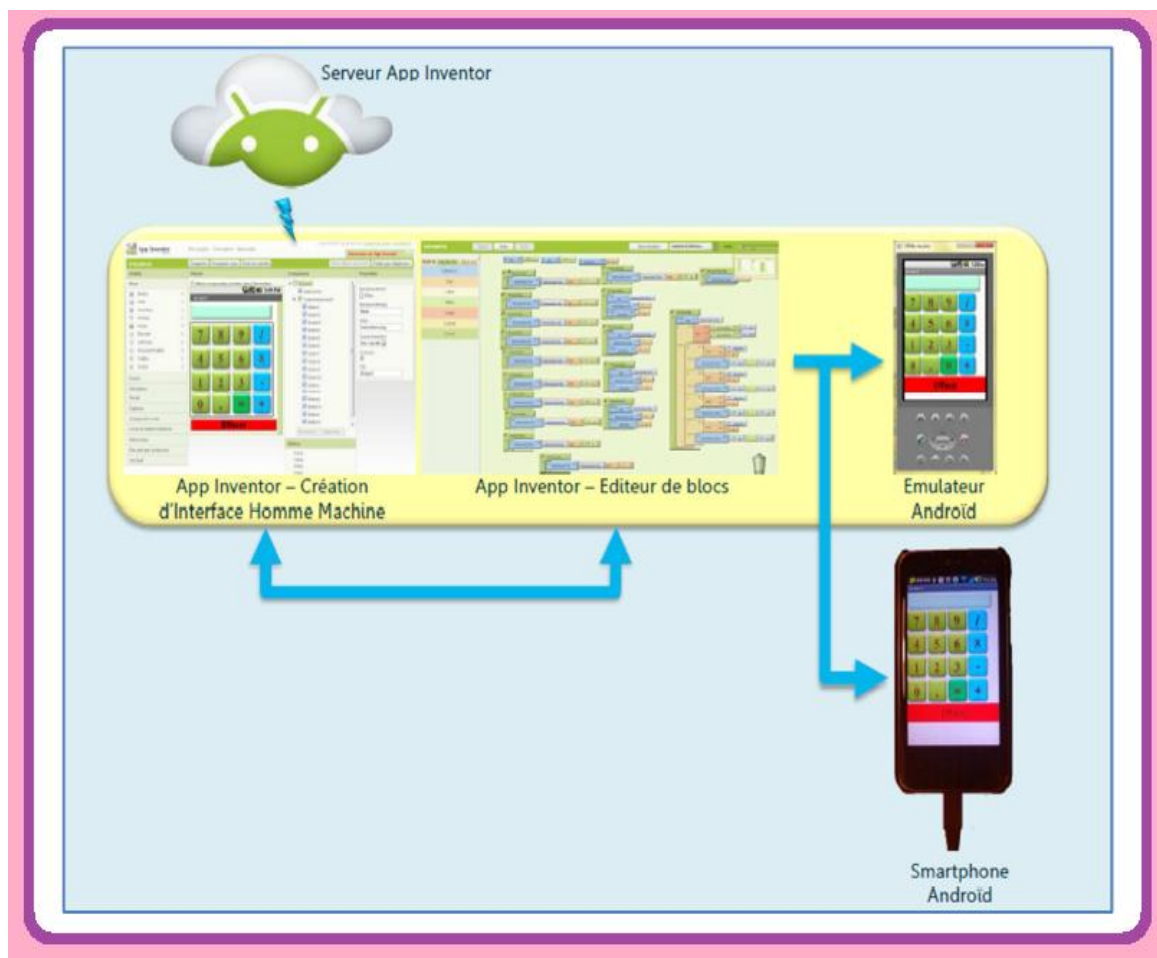


Figure 4.38: Concept d'App Inventer 2

Fenêtres proposées pendant le développement:

Trois (3) fenêtres sont proposées pendant le développement :

- Une pour la création de l'interface homme machine : ce sera l'allure de notre application,
- Une pour la programmation par elle-même : elle permettra, par l'assemblage de blocs de créer le comportement de l'application,
- Une pour l'émulateur qui permettra de tester l'application. L'émulateur permet de remplacer un terminal,
- Réel pour vérifier le bon fonctionnement du programme,
- La connexion d'un terminal réel sous Android permettra ensuite d'y télécharger le programme.

Premier Fenêtre: fenêtre de création de l'interface

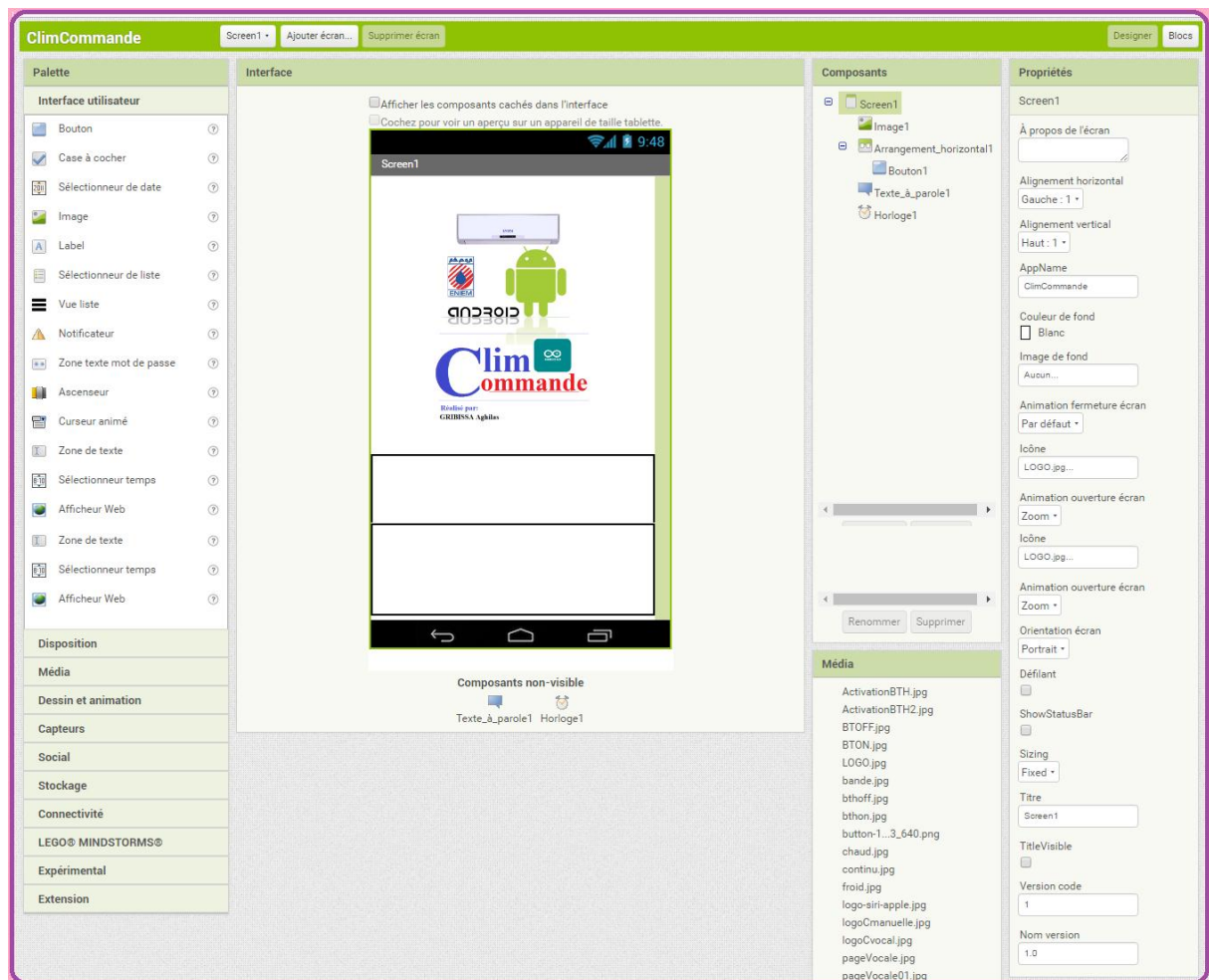


Figure 4.39: Fenêtre de création de l'interface

Deuxième fenêtre : la fenêtre de programmation graphique



Figure 4.40: Fenêtre de programmation graphique

Troisième fenêtre: l'émulateur pour tester le programme



Figure 4.41: Emulateur pour tester le programme

L'approche graphique ne remplace pas la réflexion et l'analyse préalable nécessaire à tout projet de développement.

La logique de programmation événementielle reste également un préalable avant la mise en œuvre d'app inventées.

Par contre, elle a le mérite de masquer toute la complexité imposée par l'apprentissage d'un langage de programmation objet (ici java), à savoir une syntaxe et des concepts parfois subtils. Cet apprentissage restera obligatoire pour aller plus loin notamment dans le domaine

Le langage HTML et CSS est à la base du fonctionnement de tous les sites web. Quand on consulte un site avec votre navigateur, il faut savoir que, en coulisses, des rouages s'activent pour permettre au site web de s'afficher, l'ordinateur se base sur ce qu'on lui a expliqué en HTML et CSS pour savoir ce qu'il doit afficher, comme le montre la figure suivante.



Figure 4.42 : Déférence entre le langage HTML,CSS et le Résultat a l'écran

Pour créer un site web, on doit donner des instructions à l'ordinateur. Il ne suffit pas simplement de taper le texte qui devra figurer dans le site (comme on le ferait dans un traitement de texte Word par exemple), il faut aussi indiquer où placer ce texte, insérer les images, faire des liens entre les pages, etc.

Différence entre internet et le web:

De nombreuses personnes confondent (à tort) internet et le Web. Il faut savoir que le Web fait partie d'internet, internet est un grand ensemble qui comprend, entre autres: le Web, les emails, la messagerie instantanée, etc.

Rôles de HTML et CSS:

Pour expliquer à l'ordinateur ce que l'on doit faire, il va falloir utiliser un langage qu'il comprend, et il va falloir apprendre deux langages.

- **HTML** (Hyper Text Markup Language): il a fait son apparition dès 1991 lors du lancement du web. Son rôle est de gérer et organiser le contenu, c'est donc en HTML que vous écrivez ce qui doit être affiché sur la page : du texte, des liens, des images.
- **CSS** (Cascading Style Sheets, aussi appelées Feuilles de style): le rôle du CSS est de gérer l'apparence de la page Web (agencement, positionnement, décoration, couleurs, taille du texte...). Ce langage est venu compléter le HTML en 1996.

Vous pouvez très bien créer un site web uniquement en HTML, mais celui-ci ne sera pas très beau : l'information apparaîtra <brute>, c'est pour cela que le langage CSS vient toujours le compléter.

Pour vous donner une idée, la figure suivante montre ce que donne la même page sans CSS puis avec le CSS:

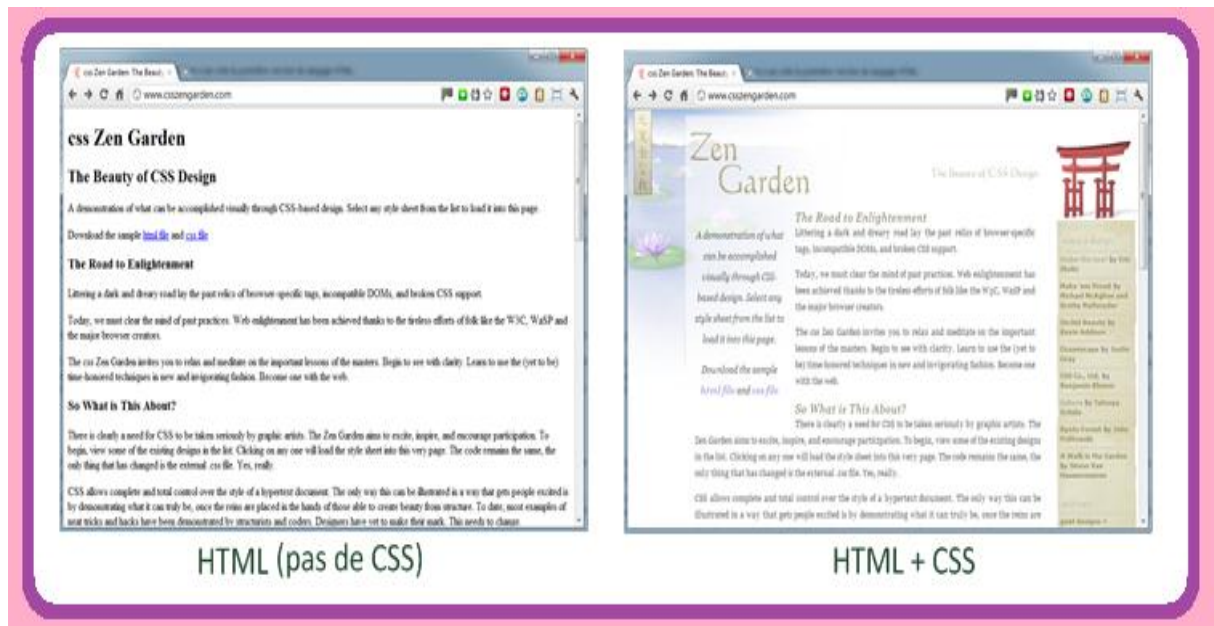


Figure 4.43: Page Web sans et avec sans CSS

Logiciel pour création de site web:

Il existe effectivement de nombreux logiciels dédiés à la création de sites Web.

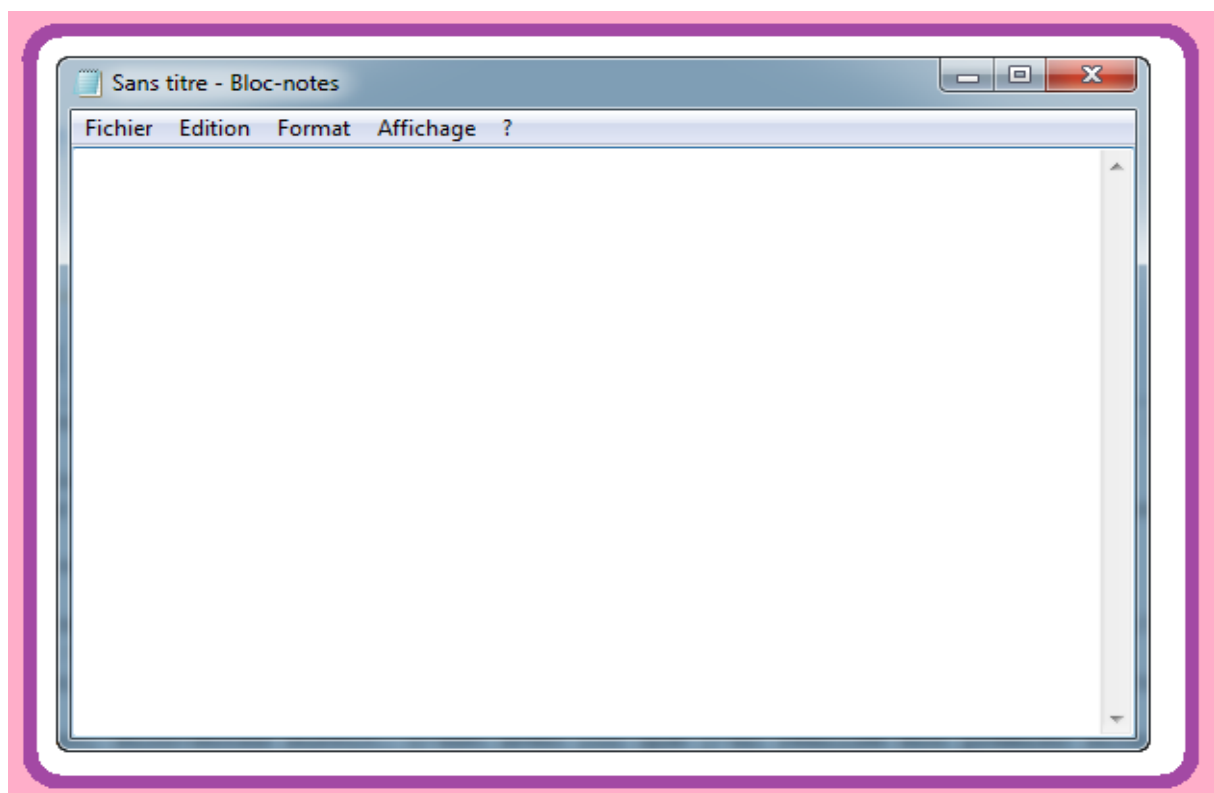


Figure 4.44: Bloc-Note

Il suffi juste d'enregistre le fichier en lui donnant le nom qu'on veulent en terminant

par ex (**.html**), par exemple (**test.html**), comme indiqué à la figure suivante:

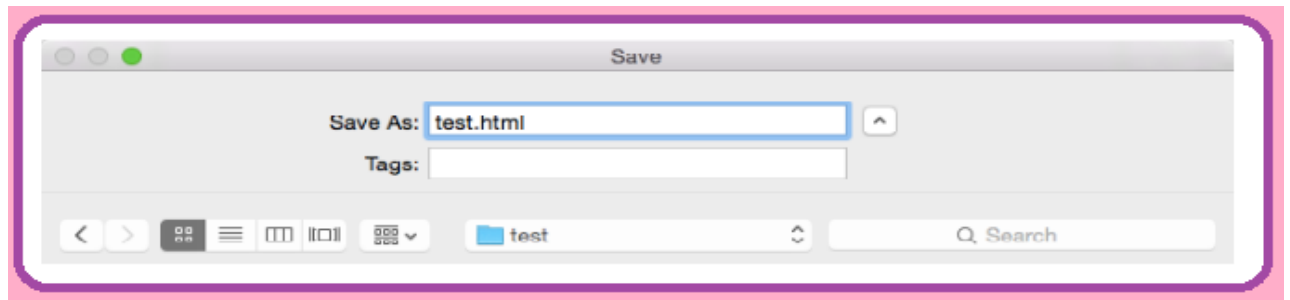


Figure 4.45: Enregistrement du programme

Bibliographie:

➤ **Sites et liens consultés :**

- www.arduino.cc
- www.abcelectronique.com
- www.willowgarage.com/pages/software/opencv.com
- www.fr.wikipedia.org
- www.opencv.org
- www.fr.openclassrooms.com
- www.developpez.net
- www.sourceforge.net
- www.github.com
- www.microsoft.com/france/visual-studio.com
- www.msdn.microsoft.com
- www.fritzing.org
- www.mon-club-elec.fr
- www.alldatsheet.com
- <http://ai2.appinventor.mit.edu/?locale=en#4963171621404672>

➤ **Ouvrage:**

- Simon Landrault (Eskimon), Hippolyte Weisslinger(Olyte), Arduino : Premiers pas en informatique embarquée, le blog d'eskimon, Edition, juin 2014.
- Jean-Noel Montagné , Atelier Arduino, initiation à la mise en œuvre matérielle et logicielle de l'arduino, centre de ressources art sensitif, novembre 2006.
- Maik Schmidt, Arduino a quick start guide, the paragmatic Programmes, janvier 20011.
- Jacques BOUTELOUP, Michel LE GUAY et Jean UGEN : Production de chaud et de froid dans un système de climatisation et de conditionnement d'air.

➤ **Thèses :**

- Mr BELKESSAM Moussa, étude et évaluation des systèmes de communication sans fil MIMO-OFDM, Master ELN, 2012/2013 , ummto
- Mr. MAHIOUT Samy, conception et réalisation d'un robot autonome mobile, Master ELN, UMMTO, 2014/2015,
- Mr .KROUBI Conception et réalisation d'un système de mesure de flambement et de déformation à base d'une Arduino Uno, Master ELN, UMMTO, 2014/2015,

- Mr ZERBOUT Lyes, Mr BOUGUERIDA Med. Etude d'un système de réfrigération simple de Linde, M2 option Energétique.
- Mr : R.NEBBALI, M elle : KHALFOUNI Louiza. Etude Paramétrique d'une tour de refroidissement d'eau, M2 option Energétique.