

**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MOULOUD MAMMÈRI, TIZI-OUZOU**



**FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE**

Mémoire de Fin d'Etudes DE MASTER ACADEMIQUE

**Filière : Electronique
Spécialité : Instrumentation**

Présenté par :

M^{lle} KRAZEM Chabha

M^{lle} LACETE Malha

Thème

Conception et réalisation d'un gilet intelligent.

Mémoire soutenu publiquement le 27/09/2018 devant le jury composé de :

Mr Mohia Y.	professeur	U. M. M. Tizi-Ouzou	Président
Mr Ziani R.	professeur	U. M. M. Tizi-Ouzou	Encadreur
Mr Attaf	professeur	U. M. M. Tizi-Ouzou	Examineur

Promotion 2017/2018

Remerciements

Nous remercions d'abord le bon Dieu qui nous a donné le courage, la patience, la santé et la volonté d'arriver à la fin de ce travail.

*Nous tenons à remercier notre promoteur **Monsieur ZIANI** pour son aide, sa disponibilité et les conseils qu'il n'a cessé de nous prodiguer pour l'aboutissement de ce modeste travail, qu'il trouve ici notre profonde gratitude et toute notre reconnaissance.*

*Nos remerciements s'adressent également à **Monsieur MOULOUDJ** professeur et formateur à l'école technique de Tizi-Ouzou, pour sa générosité et la grande patience dont il a su faire preuve malgré ses charges.*

*Nos vifs remerciements vont également aux **membres du jury** pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner ce travail et de l'enrichir par leurs propositions.*

Enfin à toutes et tous ceux qui ont contribué au déroulement de ce travail.

*Que tous les **enseignants** ayant contribué et participé à notre formation trouvent ici notre profond respect.*

Dédicace

Je dédie ce modeste travail :

À mes chers parents aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices qu'ils ont consenti pour mon instruction et mon bien être. Puisse Dieu, le très Haut, leur accorder santé, bonheur et longue vie.

À ma très chère grand-mère qui m'a accompagné avec ses prières, sa douceur, puisse Dieu lui prêter longue vie et beaucoup de santé.

À mes très chères sœurs Melissa, Nora, Melina, pour leurs soutiens infinis et leurs aides incessantes, à qui je souhaite un meilleur avenir.

À toutes ma famille mes oncles, mes tantes, mes cousins, mes cousines.

À tous mes amis(es), en particulier Doria.

À mon binôme Malha.

À monsieur ZIANI.

À monsieur MOULOUDI.

*À tous mes camarades et la promotion Instrumentation électronique
2017/2018.*



Chabha

Dédicace

Je dédie ce modeste travail :

*À mes chers parents aucune dédicace ne saurait exprimer mon respect,
mon amour éternel et ma considération pour les sacrifices qu'ils ont
consenti pour mon instruction et mon bien être. Puisse Dieu, le très Haut,
leur accorde santé, bonheur et longue vie.*

À ma très chère tante puisse Dieu lui prêter longue vie.

À mes très chères sœurs Mélissa, Nassima, Sonia, Thiziri,

*À toutes ma famille mes oncles, mes tantes, mes cousins, mes cousines en
particulier Fazia, Fétouma, Mhand et Amazigh.*

À tous mes amis(es), en particulier Youva, Sabrina, Yasmine,

À mon binôme Chabha.

À monsieur ZIANI.

À monsieur MOULOUDI.

À tous mes camarades et la promotion Instrumentation électronique

2017/2018.



Maïha

Glossaire

API : Application Programming Interface : Interface de programmation d'applications.

CAN : Convertisseur Analogique Numérique.

DC : Direct Current : courant continu.

EDI : l'Espace Développement Intégré.

EEPROM : Electrically-Erasable Programmable Read-Only Memory : mémoire morte Effaçable électriquement et programmable.

E/S : Entrées/Sorties.

GND : Ground : la masse ou 0V.

GSM : Global System for Mobile Communications : Système Global pour les Communications Mobiles.

Hex : hexa décimale.

ICSP : In Circuit Serial Programming : Programmation série en circuit.

IDE : Integrated Development Environment : Environnement de développement intégré.

INT : The interrupt : L'interruption.

I/O : Input/Output : Entrée sortie.

IOS : (iphone ou ipad)open source.

ISP : Microcontrôleur préprogrammé avec un bootloader.

LED : Light Emitting Diode : Diode électroluminescente.

LQR : Linear Quadratic Regulator : Régulateur linéaire quadratique.

MHZ : Mega Hertz.

MIT : Massachusetts Institute of Technology.

MTBF : la fiabilité.

ON /OFF : commande marche-arrêt.

OS : Open Source.

PC : Personal Computer.

PD : Proportionnelle Dérivée.

Pic : Peripheral Interface Controller : Contrôleur d'interface périphérique.

PME : Petite ou Moyenne Entreprise.

PWM : Pulse Width Modulated : modulation de largeur d'impulsions.

RAM : Random Access Memory : la mémoire vive.

RX : Receiver : récepteur.

RXD : Receive data : réception de donnée par l'ordinateur.

SCL : Serial Clock : la ligne d'horloge.

SDA : Serial Daata.

SMS : Short Message Service.

SRAM : Mémoire volatile.

TCI : Tracé du Circuit Imprimé.

Tmax : Température maximum.

Tmin : Température minimum.

TTL : Transistor- Transistor Logic.

TX : Transmission.

TXD : Transmit data, émission de donnée vers l'appareil à piloter.

USB : Universal Serial Bus : bus universel en série.

VCC : Common Collector Voltage : Alimentation Tension Continue.

Liste de figures

Figure I.1 : Schéma d'un capteur	03
Figure I.2 : Exemple de constitution de chaînes de mesure : chaîne contrôlée par un microprocesseur	05
Figure I.3 : Effet piézoélectrique	10
Figure I.4 : schéma d'un capteur de pression	10
Figure I.5 : schéma d'un capteur de pression.....	11
Figure I.6 : Effet Hall	11
Figure I.7 : La structure d'un capteur de champ magnétique	12
Figure I.8 : Photodiode $I=f(U)$	13
Figure I.9 : Transmission de données	14
Figure I.10 : Roue codeuse.....	14
Figure I.11 : Capteurs potentiométriques de déplacement.....	15
Figure I.12 : Mesure de débit de fluide	15
Figure I.13 : Fonctionnement d'une jauge (Corps au repos)	16
Figure I.14 : Fonctionnement d'une jauge (Corps ayant subi un étirement).....	16
Figure I.15 : pont de Wheatstone	17
Figure I.16 : Thermomètre à thermocouple	18
Figure I.17 : DS1621 8-PIN DIP (300-MIL.....	19
Figure I.18 : Les capteurs 1 Wire ou i-button	20
Figure II.1 : Les différentes cartes Arduino selon leurs tailles	24
Figure II.2 : Les composants de la carte Arduino UNO	25
Figure II 3 : Les différentes parties de la fenêtre principale du logiciel Arduino.....	26
Figure II 4 : la barre de menu.....	26
Figure II.5 : Barre de boutons pour le programme C Arduino	27
Figure II.6 : Différents composants du programme Arduino	28

Figure II.7 : L'écriture de programme	32
Figure II.8 : Compilation du programme	32
Figure II.9: Console d'affichage	32
Figure II.10 : Sélection de la cible	33
Figure II.11 : Sélection du port	33
Figure II.12 : Bouton de transfert du programme	34
Figure II.13 : Port Rx, Tx communication série	34
Figure II.14 : Communication carte Arduino et PC	35
Figure III.1 : Vue de la carte Arduino Nano	37
Figure III.2 : Vue du Module Bluetooth HC-05	38
Figure III.3 : Câblage d'un capteur DHTxx	40
Figure III.5 : Le GY-521	41
Figure III.6 : Le buzzeur	42
Figure III.7 : Le régulateur de tension 7805	42
Figure III.8 : logiciel Arduino	43
Figure III.9 : L'interface logicielle Arduino	44
Figure III.10 : Les boutons d'interphase de programmation Arduino	44
Figure III.11 : La structure d'un programme Arduino	45
Figure III.12 : Le concept d'App Inventor	48
Figure III.13 : L'interface d'app inventor	49
Figure III.14 : La fenêtre de programmation graphique	49
Figure III.15 : L'émulateur pour tester le programme	50
Figure III.16 : Les menus de la fenêtre de travail androïde	50
Figure III.17 : La liste des éléments et des medias	51
Figure III.18 : La palette des éléments	52
Figure III.19 : L'écran d'affichage des éléments	52
Figure III.20 : Liste des propriétés	52

Figure III.21 : L'interface de programmation de l'application	53
Figure III.22 : Les blocks de l'algorithme	53
Figure III.23 : Test de fonctionnement de l'application	54
Figure III.24 : Téléchargement de l'application	54
Figure IV.1 : Schéma explicatif du fonctionnement du gilet intelligent	56
Figure IV.2 : Schéma synoptique général de la réalisation	57
Figure IV.3 : Circuit électrique de la carte électronique	58
Figure IV.4 : Circuit sur le logiciel TCI	59
Figure IV.5 : Typon imprimé	60
Figure IV.6 : Carte électronique finale personnalisée	61
Figure IV.7 : La communication du capteur DHT11 avec Arduino Nano	62
Figure IV.8 : Câblage du GY 521 a une carte Arduino Nano	63
Figure IV.9 : Câblage du HC-05 a une carte Arduino Nano	64
Figure IV.10 : Câblage du buzzeur a une carte Arduino Nano	64
Figure IV.11 : Câblage de la batterie avec carte Arduino Nano	65
Figure IV.12 : Organigramme de fonctionnement de l'application	66
Figure IV.13 : Le sous organigramme de fonctionnement de l'application	67
Figure IV.14 : Programme déclaration sur le logiciel Arduino	68
Figure IV.15 : Programme d'action	69
Figure IV.16 : Application Smart Vest	70
Figure IV.17 : Les différents éléments de l'application	71
Figure IV.18 : Programme de configuration de Bluetooth	72
Figure IV.19 : Code pour activer /désactiver la communication GSM	73
Figure IV.20 : Code pour l'envoi du SMS	73
Figure IV.21 : L'application fonctionne	74

Liste des tableaux

Tableau I.1 : Grandeurs d'entrée et de sortie et effet utilisé pour les capteurs actifs	08
Tableau I.2 : Types de matériaux utilisés et caractéristiques électriques des capteurs passifs ...	09
Tableau I.3 : Temperature / data relation ships.....	19
Tableau II.1 : Types et caractéristiques des cartes Arduino	23
Tableau III.1 : Caractéristiques de la carte Arduino Nano	37
Tableau III.2 : Les caractéristiques du capteur DHT11	39

Table des matières

Remerciements

Dédicace

Glossaire

Liste des figures

Liste des tableaux

Table des matières

Introduction générale..... 01

Chapitre I : Les capteurs en instrumentation.

1- Introduction..... 03

2- Définitions 03

3- La chaîne de mesure 04

4- Grandeurs d'influence 06

5- Classification des capteurs..... 07

6- Les différents types de capteurs..... 07

 6.1-Capteurs actifs 07

 6.2-Capteurs passifs 09

7- Capteur a effet piézoélectrique 10

 7.1-Effet piézoélectrique 10

 7.2-Capteur de pression..... 10

 7.3-Capteur d'accélération..... 11

 7.4-Récepteur à ultrason 11

8- Capteur a effet Hall..... 11

 8.1-L'effet Hall 11

 8.2-Capteur de champ magnétique..... 12

9- Capteurs à effet photoélectrique 12

 9.1-L'effet photoélectrique 12

 9.2-Les photorésistances 12

9.3- Les photodiodes	13
10- Capteurs à résistance variable par déformation.....	14
10.1- Capteurs potentiométrique de déplacement.....	14
10.2- Capteurs à jauges d'extensomètre	16
a- Fonctionnement d'une jauge simple	16
b- Conditionneur de signal (pont de Wheatstone).....	17
11- Capteurs de température	18
11.1- Thermomètre à thermocouple.....	18
11.2 - Thermistance.....	18
11.3 - Capteurs à sortie numérique directe	19
a- Les capteurs à sortie I2C (2 fils) DS1621	19
b- Les capteurs 1 Wire ou i-button (1 fil) DS1820	20
12- Choix d'un capteur.....	21
13- Conclusion.....	21

Chapitre II : Présentation d'Arduino.

1- Introduction.....	22
2- Historique.....	22
3- La description d'Arduino.....	22
3.1- La description matérielle	22
3.2- La description matérielle	22
4- Les différentes cartes Arduino	23
5- Le fonctionnement de la carte Arduino	24
6- Langage de programmation	25
6.1- Présentation de l'espace développement Intégré (EDI) Arduino.....	25
6.2- Différents compositions de l'espace de développement intégré (EDI).....	26
7- Structure d'un programme Arduino.....	27
8- Description de la structure d'un programme Arduino.....	27
8.1 - Le code minimal	29
8.2 - Jeux d'instructions du langage Arduino.....	29

9- Commandes du programme Arduino	29
10- Etapes à suivre pour programmer la carte Arduino	31
a- Ecriture du programme	31
b- Compilation du programme	32
c- Sélection de la cible (type de carte).....	33
d- Sélection du port	33
e- Transfert du programme	34
11- La communication entre le PC et la carte	34
12- Bonnes raisons de choisir Arduino	35
13-Conclusion.....	35

Chapitre III : Etude de la partie matérielle et logicielle.

1- Introduction.....	36
2- Conception matériel.....	36
a- La carte Arduino Nano	36
b- Bluetooth HC-05.....	38
c- Le capteur DHT 11	39
d- GY521	40
e- Le buzzeur	41
f- Le régulateur de tension 7805	42
3- La conception logicielle.....	43
a- Logiciel Arduino	43
b- Logiciel TCI	46
c- Application androïde	46
1-L'androïde	47
2-App inventor 2	47
2.1- Le concept d'App Inventor	47
2.2- La fenêtre de la création de l'interface	49
a- La fenêtre de programmation graphique	49
b- L'émulateur pour tester le programme.....	50
c- Présentation des différents menus	50

2.3- Les étapes de développement d'application androïde avec appinventor	51
2.3.1- La création de son interface	51
2.3.2- La conception de l'écran se partage en 4 parties	51
2.3.3- Le comportement de l'application	53
4-Conclusion	54

Chapitre IV : Réalisation pratique du projet

1- Introduction.....	55
2-Schéma explicatif du fonctionnement dugilet intelligent.....	55
3-Schéma synoptique global.....	57
4- La réalisation du projet.....	58
A- Partie électronique	58
1- Realisation de la Carte électronique	58
2- Test du fonctionnement de la carte Arduino	62
3- Branchement des différents composants avec la carte Arduino Nano	62
4- L'organigramme principal.....	65
5- Le sous organigramme de fonctionnement.....	67
B- Partie logicielle	68
1- Programme de déclaration	68
2- Programme d'action	69
3- Connexion entre la carte électronique et le Smartphone	69
4- Conception de l'application	70
4.1 La programmation événementielle de l'application Smart Vest	72
5- Teste l'application sur le Smartphone	74
5-Conclusion.....	75
Conclusion générale.....	76
Bibliographie	77



Introduction générale

Introduction générale

Après l'indépendance de l'Algérie, trois générations cohabitaient sous un même toit, c'est de moins en moins le cas aujourd'hui. Ils sont désormais de plus en plus nombreux, ces grands-parents à se retrouver seuls sans assistance et souffrant pour les plus mal lotis de pathologies lourdes. Cette solitude est encore plus difficile pour les personnes vivant en milieu rural où les structures sanitaires sont rares. La transition épidémiologique, c'est à dire la multiplication de nouvelles maladies, comme le diabète, l'hypertension artérielle ou encore le cancer, est venue fragiliser cette catégorie de la population dont les besoins médicaux sont spécifiques et nécessaires. Pour l'heure, les structures spécialisées, comme les maisons de retraite et foyers pour personnes âgées sont trop peu nombreuses en Algérie.

Alors que la population Algérienne vieillit massivement et que l'espérance de vie augmente avec des conséquences importantes sur la qualité et les conditions de vie des personnes âgées (en termes de dégradations, de maladies, d'incapacités diverses...), il devient nécessaire de réfléchir à de nouveaux moyens de prise en charge qui peuvent suppléer ou compléter les modes d'assistance traditionnels (aide-soignante, infirmière...), notamment en raison de leur coût et d'un déploiement encore insuffisant (manque de personnel, formation insuffisante et /ou inadaptée...).

Le développement des technologies de l'information et de la communication pour l'aide à la personne âgée représente une piste prometteuse. Elles laissent en effet augurer de nouvelles opportunités pour l'assistance et la prise en charge des personnes âgées à domicile ou en institutions spécialisées (hôpitaux...). Déjà présentes dans de nombreux secteurs médicaux et sanitaires (télémédecine, télédiagnostic...), ces technologies se répandent de plus en plus aux domiciles même des individus. Elles fournissent des moyens de surveillance et d'assistance qui ont pour but d'améliorer le confort de vie des personnes âgées.

La question qui se pose par ailleurs est comment exploiter le développement des technologies pour mettre en sécurité la santé de la personne âgée ?

Les risques que court les personnes âgées vivant seuls et sans assistance et le développement des technologies nous ont motivés à la réalisation d'un gilet intelligent qui servira à la surveillance de ces personnes à distance.

Introduction générale

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

Le premier chapitre de ce mémoire est consacré aux capteurs en instrumentation.

Dans le second chapitre, nous avons fait une description détaillée de la carte Arduino avec ses différents types et la programmation y afférente.

Nous avons consacré le troisième chapitre à l'étude de tout le matériel et les logiciels que nous avons utilisé pour la commande et le fonctionnement du gilet intelligent.

Le dernier chapitre de ce mémoire est consacré à la réalisation pratique avec illustration des schémas de branchement des différents composants qu'on a utilisé avec la carte Arduino nano ainsi que les codes de programmes utilisés pour cette carte et enfin les résultats du test effectué avec notre application.

Chapitre I

Les capteurs en instrumentation.

1- Introduction :

Nous savons que seule la mesure permet de conduire le progrès scientifique et d'accroître la connaissance de ce qui nous entoure ; les limites ne sont pas atteintes et ne le seront jamais. Il nous apparait, à ce jour, que la mesure joue un rôle de plus en plus fondamental dans le développement des activités industrielles avec la sophistication des automatismes, la robotique, le contrôle de la qualité, les économies d'énergie, la lutte contre la pollution, etc. De plus, nous trouvons maintenant des applications nombreuses dans la conception et la réalisation des biens de consommation, l'électroménager, l'automobile, les jouets, etc.

« Ce monde est pénétré des applications de la mesure ; toute connaissance, non mesurable, est frappée d'un jugement de dépréciation. Le nom de « science » se refuse de plus en plus à tout savoir intraduisible en chiffre ». Cette affirmation de Paul Valéry en 1982 est plus que d'actualité. Notre monde est, et va être de plus en plus, pénétré des applications de la mesure. Dans de nombreux domaines (industrie, recherche scientifique, services, loisirs ...), nous avons besoin de contrôler de nombreux paramètres physiques (température, force, position, vitesse, luminosité, ...).

Rien ne se fera sans le « capteur », qui est à la base même de l'édifice « mesure ». Lui seul permet de prolonger et d'affiner les sens de l'homme.

2- Définitions :

Capteur : Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique, une autre grandeur physique de nature différente (très souvent électrique). Cette dernière représentative de la grandeur prélevée est utilisable à des fins de mesure ou de commande.

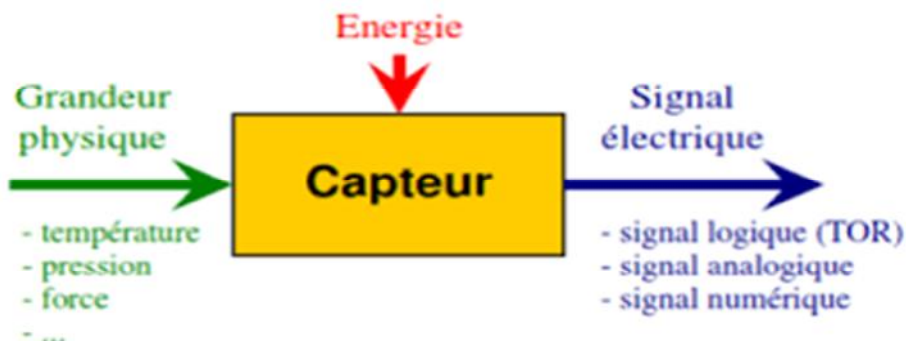


Figure I.1 : Schéma d'un capteur.

Etendue de mesure : Valeurs extrêmes pouvant être mesurées par le capteur.

Résolution : Plus petite variation de grandeur mesurable par le capteur.

Sensibilité : Variation du signal de sortie par rapport à la variation du signal d'entrée.

Exemple : Le capteur de température LM35 a une sensibilité de 10mV / °C.

Précision : Aptitude du capteur à donner une mesure proche de la valeur vraie.

Rapidité : Temps de réaction du capteur. La rapidité est liée à la bande passante.

3- La chaîne de mesure :

La chaîne de mesure est constituée de l'ensemble des dispositifs, y compris le capteur, rendant possible, dans les meilleures conditions, la détermination précise de la valeur du mesurande.

À l'entrée de la chaîne, le capteur soumis à l'action du mesurande permet, directement s'il est actif ou par le moyen de son conditionneur s'il est passif, d'injecter dans la chaîne le signal électrique.

À la sortie de la chaîne, le signal électrique qu'elle a traité est converti sous une forme qui rend possible la lecture directe de la valeur cherchée du mesurande :

- Déviation de l'aiguille d'un instrument analogique.
- Affichage sur un écran.
- Enregistrement sous forme de courbe.

C'est l'étalonnage de la chaîne de mesure dans son ensemble qui permet d'attribuer à chaque indication en sortie la valeur correspondante du mesurande agissant à l'entrée.

Sous sa forme la plus simple, la chaîne de mesure peut se réduire au capteur et à son conditionneur éventuel, associé à un appareil de lecture :

- Thermocouple et voltmètre.
- Jauge de contrainte placée dans un pont de Wheatstone, avec un afficheur numérique.

Cependant les conditions pratiques de mesure telles qu'elles sont imposées par l'environnement d'une part et par les performances exigées pour une exploitation satisfaisante

du signal d'autre part amènent à introduire dans la chaîne des blocs fonctionnels destinés à optimiser l'acquisition et le traitement du signal :

- Circuit de linéarisation du signal délivré par le capteur.
- Amplificateur d'instrumentation ou d'isolement destiné à réduire les tensions parasite de mode commun.
- Multiplexeur, amplificateur d'instrumentation programmable, échantillonneur bloqueur, convertisseur analogique pour que l'information soit traitée par un microprocesseur ou un microcontrôleur (figure I.2).

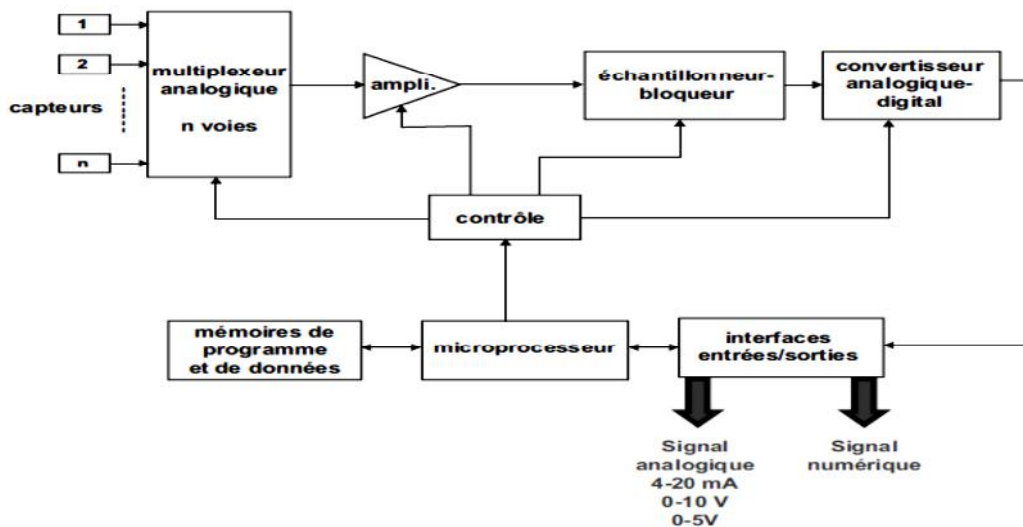


Figure I.2 : Exemple de constitution de chaînes de mesure : chaîne contrôlée par un microprocesseur.

Les fonctions multiples et importantes du système de traitement numérique (microprocesseur ou microcontrôleur) associé à la chaîne de mesure peuvent être regroupées sous deux rubriques :

- Gestion de l'acquisition d'une part ;
- Traitements du signal requis par la précision et par la nature de l'information cherchée d'autre part.

Le système de traitement numérique est le chef d'orchestre de la chaîne d'acquisition; il délivre les séquences de signaux de commande activant de façon ordonnée les divers dispositifs concourant à l'obtention de la valeur du mesurande particulier dont la connaissance à un instant donné est nécessaire au déroulement de l'application :

- Sélection d'une voie d'entrée par envoi d'adresse au multiplexeur ;
- Fixation du gain de l'amplificateur programmable ;
- Echantillonnage puis blocage du signal ;
- Déclenchement de la conversion analogique-numérique ;
- Lecture de la donnée numérique à la réception du signal de fin de conversion délivré par le convertisseur analogique-numérique.

De nos jours, compte tenu des possibilités offertes par l'électronique et l'informatique, les capteurs délivrent un signal électrique et la quasi-totalité des chaînes de mesure sont des chaînes électroniques et informatiques.

Certains capteurs, par exemple le thermomètre DALLAS DS1621, délivrent directement un mot binaire, image de la température, en leur sortie. Ils intègrent, dans un seul boîtier (DIL 08) le capteur + le circuit de mise en forme + le CAN.

4- Grandeurs d'influence :

Les grandeurs d'influence sont des grandeurs étrangères qui, selon leur nature et leur importance, peuvent provoquer des perturbations sur le capteur. C'est donc une cause d'erreurs agissant sur le signal de sortie. Les principales grandeurs d'influence sont :

- La température qui modifie les caractéristiques électriques, mécaniques et dimensionnelles des composants du capteur ;
- La pression, l'accélération et les vibrations susceptibles de créer dans certains éléments constitutifs du capteur des déformations et des contraintes qui altèrent la réponse ;
- L'humidité à laquelle certaines propriétés électriques comme la constante diélectrique ou la résistivité peuvent être sensibles et qui risque de dégrader l'isolation électrique entre composants du capteur ou entre le capteur et son environnement ;
- Les champs magnétiques variables ou statiques ; les premiers créent des f.é.m. d'induction qui se superposent au signal utile, les seconds peuvent modifier une propriété électrique ;
- La tension d'alimentation.

5- Classification des capteurs :

La classification se fait par :

- Le mesurande qu'ils traduisent (capteur de température, de pression, ...)
- Leur rôle dans un processus industriel (contrôle de produits finis, de sécurité, ...)
- Le signal qu'ils fournissent (capteur analogique, capteur logique, capteurs digitaux)
- Leur principe de traduction du mesurande (capteur résistif, à effet hall,...)
- Leur principe de fonctionnement : les capteurs fonctionnent selon deux principes de base suivant l'origine du signal électrique de sortie.

6- Les différents types de capteurs :

Si l'on s'intéresse aux phénomènes physiques mis en jeu dans les capteurs, on peut classer ces derniers en deux catégories :

- ✓ Capteurs actifs
- ✓ Capteurs passifs

6.1-Capteurs actifs :

Un capteur actif est un capteur qui fonctionne en générateur, il est généralement fondé dans son principe sur un effet physique qui assure la conversion de la grandeur physique prélevé (énergie thermique, mécanique ou de rayonnement en une grandeur électrique.

Les effets physiques les plus rencontrés en instrumentation sont :

- **Effet thermoélectrique :** Un circuit formé de deux conducteurs de nature chimique différente, dont les jonctions sont à des températures **T1** et **T2**, est le siège d'une force électromotrice d'origine thermique **e (T1, T2)**.
- **Effet piézo-électrique :** L'application d'une contrainte mécanique à certains matériaux dits piézo-électriques (le quartz par exemple) entraîne l'apparition d'une déformation et d'une même charge électrique de signe différent sur les faces opposées.

- **Effet d'induction électromagnétique** : La variation du flux d'induction magnétique dans un circuit électrique induit une tension électrique (détection de passage d'un objet métallique).
- **Effet Hall** : Un champ magnétique B et un courant électrique I créent dans le matériau une différence de potentiel
- **Effet photovoltaïque** : Des électrons et des trous sont libérés au voisinage d'une jonction PN illuminée, leur déplacement modifie la tension à ses bornes.

Tableau I.1 : Grandeurs d'entrée et de sortie et effet utilisé pour les capteurs actifs

Grandeur physique Mesurée	Effet utilisé	Grandeur de sortie
Température	Thermoélectricité	Tension
Flux de rayonnement Optique	Photoémission	Courant
	Effet photovoltaïque	Tension
	Effet photoélectrique	Tension
Force	Piézo-électricité	Charge électrique
Pression		
Accélération	Induction	Tension
Vitesse	Electromagnétique	
Position (aimant)	Effet Hall	Tension
Courant		

6.2-Capteurs passifs :

Il s'agit généralement d'impédance dont l'un des paramètres déterminants est sensible à la grandeur mesurée. La variation d'impédance résulte :

- Soit d'une variation de dimension du capteur, c'est le principe de fonctionnement d'un grand nombre de capteurs de position, potentiomètre, inductance à noyaux mobile, condensateur à armature mobile.
- Soit d'une déformation résultant d'une force ou d'une grandeur s'y ramenant, pression accélération (armature de condensateur soumise à une différence de pression, jauge d'extensomètre liée à une structure déformable).

Tableau I.2 : Types de matériaux utilisés et caractéristiques électriques des capteurs passifs.

Grandeur mesurée	Caractéristique électrique	Type de matériau utilisé
Température	Résistivité	Platine, nickel, cuivre ...
Très basse température	Constante diélectrique	Verre
Flux de rayonnement optique	Résistivité	Semi-conducteur
Déformation	Résistivité	Alliage de Nickel, silicium dopé
	Perméabilité	Alliage ferromagnétique
Position (aimant)	Résistivité	Matériaux magnéto résistants : bismuth, antimoine d'indium
Humidité	Résistivité	Chlorure de lithium

7- Capteur a effet piézoélectrique :

7.1-Effet piézoélectrique :

Une force appliquée à une lame de quartz induit une déformation qui donne naissance à une tension électrique.

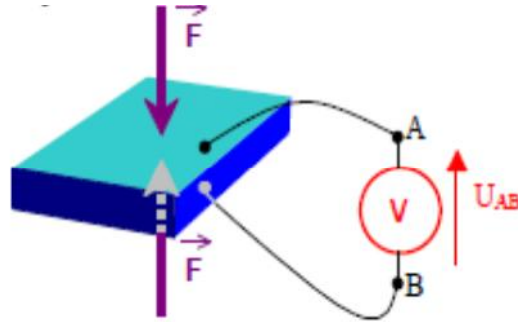


Figure I.3 : Effet piézoélectrique

7.2-Capteur de pression :

Définition : Lorsqu'un corps (gaz, liquide ou solide) exerce une force F sur une paroi S (surface); on peut définir la pression P exercée par ce corps avec la relation ci-dessous :

$$P = \frac{F}{S} \text{ Avec les unités : } 1 \text{ Pascal} = \frac{1 \text{ Newton}}{1 \text{ m}^2} \text{ ou } 1 \text{ Pa} = \frac{1 \text{ N}}{1 \text{ m}^2}$$

On rappelle que $1 \text{ kg} = 9,81 \text{ N}$.

Unites: $1 \text{ bar} = 105 \text{ Pa} = 100\,000 \text{ N} / \text{m}^2 \gg 10\,000 \text{ kg} / \text{m}^2 \gg 1 \text{ kg} / \text{cm}^2$

Le capteur de force est inséré dans la paroi d'une enceinte où règne une pression P .

Une face du capteur est soumise à la force F (pression P) et l'autre face est soumise à la force F_0 (pression extérieure P_0).

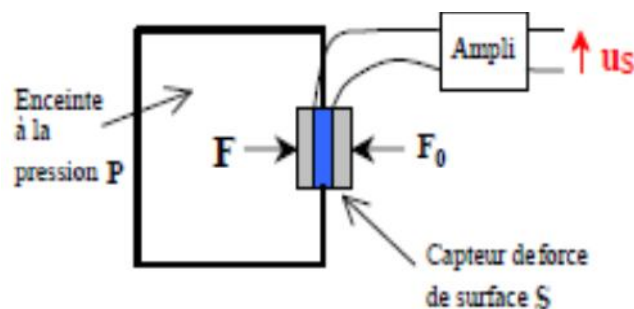


Figure I.4: Schéma d'un capteur de pression.

7.3-Capteur d'accélération :

L'augmentation de vitesse V du véhicule donne une accélération « a » qui induit une force F exercée par la masse sur le capteur.

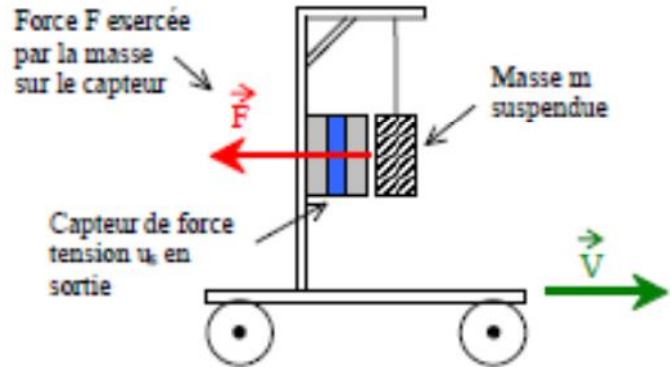


Figure I.5 : Schéma d'un capteur de pression.

7.4- Récepteur à ultrason :

La réception d'un son engendre une variation de pression à la surface du récepteur. Un capteur de pression sur cette surface donnera donc une tension image du signal ultrasonore.

8- Capteur a effet Hall :

8.1-L'effet Hall

Un barreau de semi-conducteur soumis à un champ magnétique uniforme B et traversé par un courant I , est le siège d'une force électromotrice U_H sur deux de ses faces.

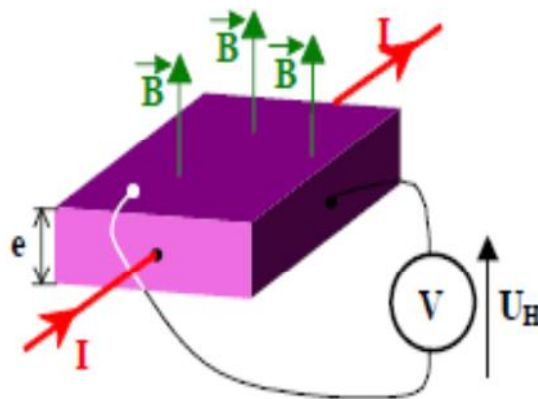


Figure I.6 : Effet Hall.

La tension de Hall U_H est définie par la relation ci-dessous :

$$U_H = R_H \frac{IB}{e} \quad \text{avec :}$$

R_H : constante de Hall (dépend du semi-conducteur)

I : intensité de la source de courant (A)

B : intensité du champ magnétique (T)

e : épaisseur du barreau de silicium.

8.2- Capteur de champ magnétique :

La structure typique d'un capteur de champ magnétique est la suivante :

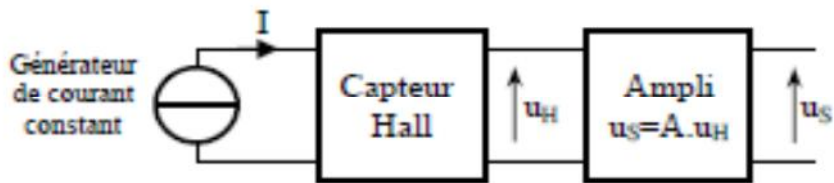


Figure I.7: La structure d'un capteur de champ magnétique.

La sensibilité de ce capteur pourra être ajustée en agissant sur I et sur A .

9- Capteurs a effet photoélectrique :

9.1-L'effet photoélectrique :

Un semi-conducteur est un matériau pauvre en porteurs de charges électriques (isolant).

Lorsqu'un photon d'énergie suffisante excite un atome du matériau, celui-ci libère plus facilement un électron qui participera à la conduction.

9.2-Les photorésistances :

Une photorésistance est une résistance dont la valeur varie en fonction du flux lumineux qu'elle reçoit.

Avantages : - Bonne sensibilité.

- Faible coût et robustesse.

Inconvénients : - Temps de réponse élevé.

- Bande passante étroite.

- Sensible à la chaleur.

Utilisation :

- Détection des changements obscurité-lumière (éclairage public).

9.3-Les photodiodes :

Une photodiode est une diode dont la jonction PN peut être soumise à un éclairement lumineux.

Le graphe $I = f(U)$ pour une photodiode dépend de l'éclairement (Lux) de la jonction PN.

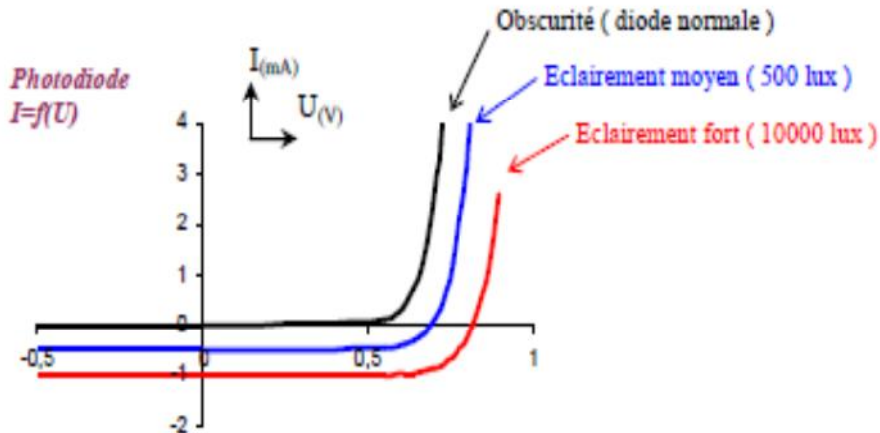


Figure I.8 : Photodiode $I=f(U)$

On constate que lorsque la diode est éclairée, elle peut se comporter en générateur ($I = 0 \quad U \gg 0,7V$ pour 1000lux). On a donc affaire à une photopile (effet photovoltaïque).

Avantages : - Bonne sensibilité.

- Faible temps de réponse (bande passante élevée).

Inconvénients :- Coût plus élevé qu'une photorésistance.

- Nécessite un circuit de polarisation précis.

Utilisations :

- Transmission de données



Figure I.9 : Transmission de données.

- Télécommande IR
- Transmission de données par fibre optique
- Détection de passage

- Roue codeuse

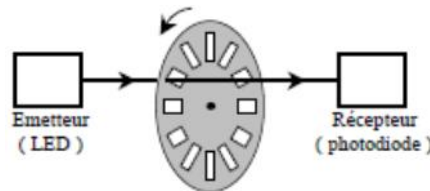


Figure I.10 : Roue codeuse

- Mesures d'angle et de vitesse
- Comptage d'impulsions (souris de PC)

10- Capteurs a résistance variable par déformation :

10.1- Capteurs potentiométriques de déplacement :

Pour mesurer la position d'un objet, il suffit de le relier mécaniquement au curseur C d'un potentiomètre (figure I.11).

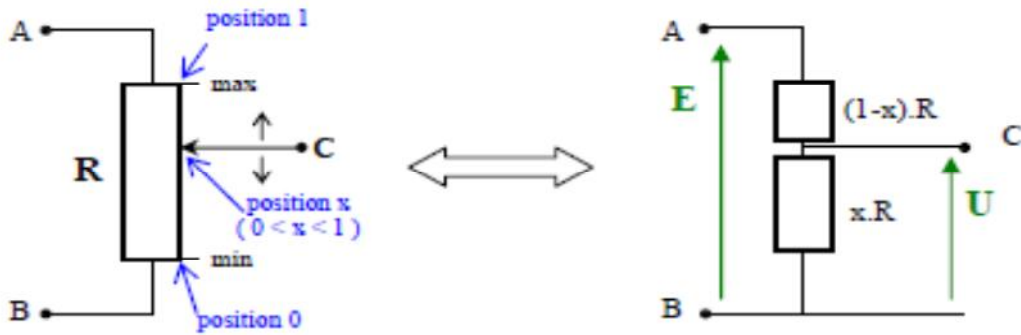


Figure I.11 : Capteurs potentiométriques de déplacement.

On applique une tension continue E entre les extrémités A et B du potentiomètre.

La tension U en sortie aura l'expression suivante : $U = E \frac{xR}{R} = xE$

La tension U en sortie est donc proportionnelle à la position x du curseur.

Avantages :- simplicité d'utilisation

- faible coût.

Inconvénient : - usure mécanique (utilisation déconseillée dans les asservissements très dynamiques)

b- Utilisations :

- Mesures de déplacements rectilignes (potentiomètre rectiligne).
- Mesures d'angles de rotations
- Mesure de débit de fluide :

Le débit du fluide exerce une force sur un clapet relié au curseur d'un potentiomètre.

La tension en sortie du potentiomètre augmente avec la vitesse d'écoulement.

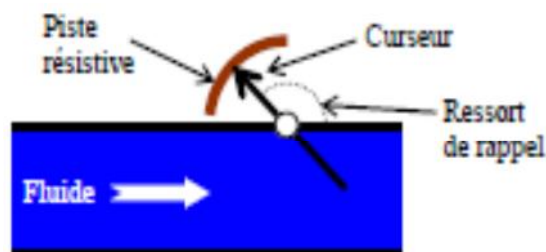


Figure I.12: Mesure de débit de fluide.

10.2 Capteurs à jauges d'extensomètre :

La résistance d'un conducteur est donnée par la relation :

$$R = \rho \frac{l}{S} \quad \text{avec } \rho : \text{résistivité } (\Omega.m)$$

l : longueur (m)

S : surface (m^2)

La déformation du conducteur (jauge) modifie la longueur l entraînant une variation de la résistance R.

La relation générale pour les jauges est : $\frac{\Delta R}{R_0} = K \frac{\Delta l}{l}$ où K est le facteur de jauge.

a- Fonctionnement d'une jauge simple :

La jauge est constituée d'une piste résistive collée sur un support en résine. Le tout est collé sur le corps dont on veut mesurer la déformation.

- Corps au repos (pas d'allongement) :

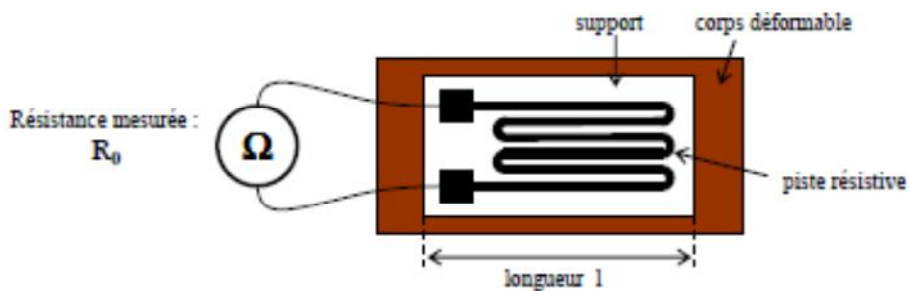


Figure I.13 : Fonctionnement d'une jauge (Corps au repos).

- Corps ayant subi un étirement (effort de traction) :

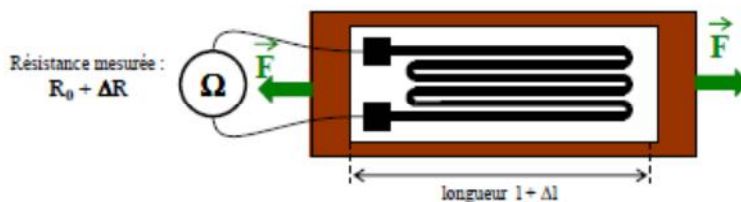


Figure I.14 : Fonctionnement d'une jauge (Corps ayant subi un étirement)

b- Conditionneur de signal (pont de Wheatstone)

La jauge étant un composant purement résistif, il faut l'associer à un circuit électrique pour obtenir une tension image de la déformation.

Le circuit souvent utilisé est appelé "pont de Wheatstone". Il est ici constitué d'un générateur de tension associé à 4 résistances dont l'une est la jauge (Figure I.15) :

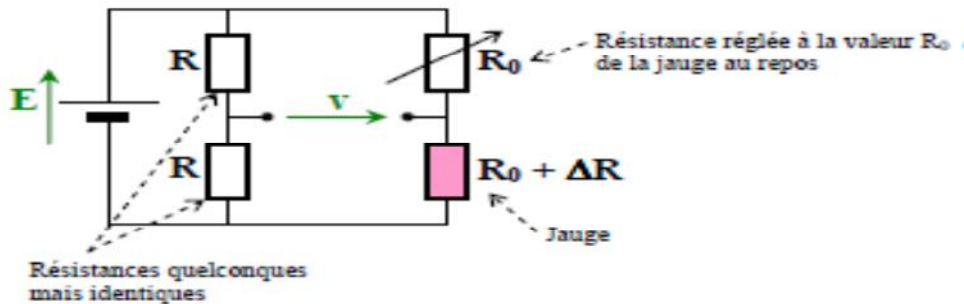


Figure I.15 : Pont de Wheatstone.

La tension de sortie v du pont a l'expression suivante :

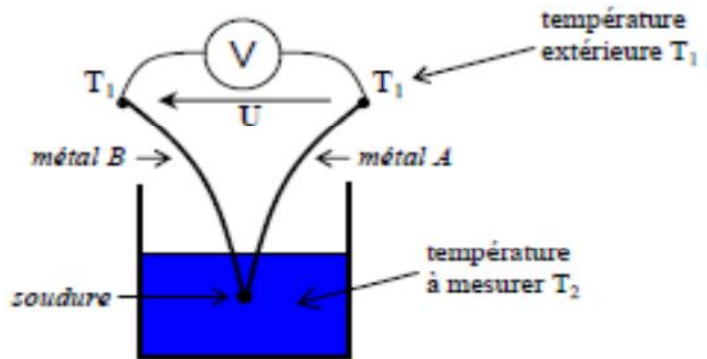
$$v = E \left[\frac{R_0 + \Delta R}{R_0 + R_0 + \Delta R} - \frac{R}{R + R} \right] = E \left[\frac{R_0 + \Delta R}{2R_0 + \Delta R} - \frac{1}{2} \right] = E \left[\frac{2R_0 + 2\Delta R - 2R_0 - \Delta R}{4R_0 + 2\Delta R} \right]$$

$$\rightarrow v = E \frac{\Delta R}{4R_0 + \Delta R}$$

En général, la variation ΔR est petite devant R_0 ; la relation se simplifie alors pour devenir quasi-linéaire :

$$v = E \frac{\Delta R}{4R_0}$$

Remarque : On peut améliorer la sensibilité et la linéarité du dispositif en utilisant un pont à 2 résistances et 2 jauges symétriques $R_0 + \Delta R$ et $R_0 - \Delta R$. Il est même possible d'utiliser un pont à 4 jauges symétriques pour avoir une parfaite linéarité.

11-Capteurs de température :**11.1-Thermomètre à thermocouple :****Figure I.16 :** Thermomètre à thermocouple.

On constate que si la température T_2 est différente de T_1 alors il apparaît une tension U aux bornes des deux fils soumis à la température T_1 .

Le phénomène inverse est aussi vrai : si on applique une tension, alors il y aura un échauffement ou un refroidissement au point de liaison des deux conducteurs (modules à effet Peltier).

Application: Mesure des hautes températures ($900 \rightarrow 1300^\circ\text{C}$).

11.2-Thermistance :

Une thermistance est un composant dont la résistance varie en fonction de la température.

En première approximation, la relation entre résistance et température est la suivante :

$$R_\theta = R_0 (1 + a\theta)$$

- R_θ est la résistance à la température θ
- R_0 est la résistance à la température 0°C
- a est le coefficient de température.

Utilisation : On insère la thermistance dans un pont de jauge.

On obtient ainsi une tension V en sortie du pont $V = k (\theta - \theta_0)$.

Si on prend $\theta_0 = 0^\circ\text{C}$, on obtient $V = k.\theta$.

On peut aussi alimenter la thermistance avec un générateur de courant.

La tension à ses bornes sera donc proportionnelle à la résistance.

11.3-Capteurs à sortie numérique directe :

On trouve actuellement sur le marché, des capteurs de température à sortie numérique directe de type série. Il s'agit notamment des capteurs *DALLAS* qui sont classés en deux catégories :

a- Les capteurs à sortie I2C (2 fils) DS1621 :

Ce capteur DS1621 peut mesurer une température variant de -55°C à 125°C avec une précision de $0,5^\circ\text{C}$.

Pour transmettre la mesure (9 bits), il utilise la norme I2C qui consiste à transmettre en série les bits de mesure sur la ligne SDA en synchronisation avec la ligne SCL (horloge).

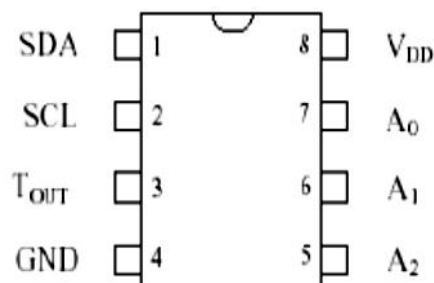


Figure I.17: DS1621 8-PIN DIP (300-MIL)

Tableau I.3: Temperature / data relation ships.

Température	Sortie numérique (binaire).	Sortie numérique (Hexa)
+125°C	01111101 00000000	7D00h
+25°C	00011001 00000000	1900h
+0°C	00000000 00000000	0000h
-25°C	11100111 00000000	E700h
-55°C	11001001 00000000	C900h

Le DS1621 possède aussi d'autres fonctions :

- Il est adressable physiquement sur 3 bits (A0, A1 et A2), ce qui permet d'en utiliser 8 sur la même ligne SDA-SCL.
- Il possède une fonction thermostat qui permet de commander un chauffage (températures TH et TL) par l'intermédiaire de la ligne TOUT même lorsque le capteur est déconnecté du matériel informatique.

b- Les capteurs 1 Wire ou i-button(1 fil) DS1820

Ce capteur DS1820 peut mesurer une température variant de -55°C à 125°C avec une précision maximale de $0,125^{\circ}\text{C}$.

Pour transmettre la mesure (résolution réglable de 9 à 12 bits), il utilise la norme i-button qui consiste à transmettre en série sur un seul fil, le résultat de la mesure.

La ligne VD peut être connectée à la masse GND et la ligne DQ supportera à la fois l'alimentation et la transmission des données, d'où l'appellation 1 Wire.

Il suffit donc de deux fils (DQ et GND) pour alimenter et communiqué avec ce capteur.

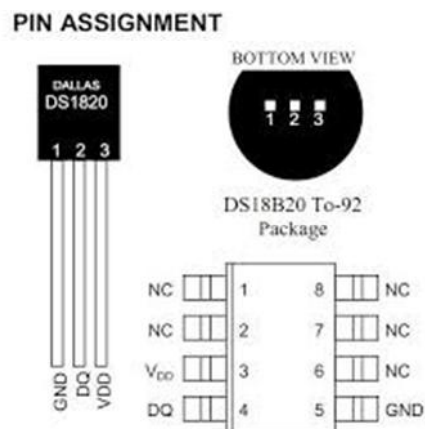


Figure I.18: Les capteurs 1 Wire ou i-button.

Le DS1820 possède aussi d'autres fonctions :

- Il est doté d'une adresse (numéro de série) affectée en usine et définitive. Elle est codée sur 8 octets ce qui permet d'utiliser, en théorie, un très grand nombre de DS1820 sur la même ligne.
- Une alarme de température peut être paramétrée et la consultation de celle-ci se fait par lecture d'une zone mémoire (adresse – donnée).

12- Choix d'un capteur :

Tous les capteurs dont les fonctionnements ont été décrits précédemment présentent deux parties distinctes, une première partie qui a pour rôle de détecter un événement et une deuxième partie qui a pour rôle de traduire un événement en un signal compréhensible d'une manière ou d'une autre par une partie PC.

Pour choisir correctement un capteur, il faudra définir tout d'abord :

- Le type événement à détecter,
- La nature de l'événement,
- La grandeur de l'événement,
- L'environnement de l'événement.

En fonction de ces paramètres on pourra effectuer un ou plusieurs choix pour un type de détection. D'autres éléments peuvent permettre de cibler précisément le capteur à utiliser :

- Ses performances,
- Son encombrement,
- Sa fiabilité (MTBF)
- La nature du signal délivré par le capteur (électrique, pneumatique)
- Son prix...

13- Conclusion :

L'industrie du capteur est aujourd'hui en évolution rapide. Cette mutation orientée vers les techniques de fabrication collectives issues de la technologie du circuit intégré silicium conduit à des produits de faible coût et intégrant un prétraitement de l'information. Plusieurs facteurs restent néanmoins limitant :

- La difficulté d'accès pour les PME à des moyens de développement et de fabrication impliquant des investissements assez élevés ;
- Un marché en constante évolution et souvent peu enclin à assumer les risques associés aux technologies innovantes ;
- Un déficit en concepteurs possédant à la fois une bonne maîtrise de la technologie et de la physico-chimie des phénomènes utilisés par les capteurs ;
- L'énorme disproportion existant entre le nombre de réalisations expérimentales et celles atteignant le stade industriel traduit un décalage certain.

Chapitre II

Présentation d'Arduino.

1- Introduction :

Pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités des relations humain/machine ou environnement/machine. Arduino est un projet open source sous licence libre.

2- Historique :

Le projet « Arduino » a été initié par un groupe d'enseignants et d'étudiants d'une école de design italienne en 2004-2005.

3- La description d'Arduino :

L'Arduino est une plateforme de prototypage électronique open-source, basée d'une part sur du matériel et d'autre part sur un ensemble de logiciels.

3.1- La description matérielle :

Le module Arduino est généralement construit autour d'un microcontrôleur AtmelAVR (ATmega328, ATmega32u4 ou ATmega2560 pour les nouvelles versions et ATmega168, ATmega1280 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5 V et un oscillateur à 16 MHz.

L'Arduino utilise la plupart des entrées/sorties du microcontrôleur pour l'interfaçage avec les autres circuits. Le modèle Diecimila par exemple, possède quatorze entrées/sorties numériques.

3.2- La description logicielle :

Le logiciel de programmation des modules Arduino est une application Java, libre et multi-plateforme, servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série Bluetooth ou USB, et de compiler les programmes via l'interface en ligne de commande.

Le langage de programmation utilisé est le C++, compilé avec Avr g++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties.

4- Les différentes cartes Arduino :

Il existe plusieurs types de cartes Arduino et le tableau (II.1) nous montre ces différents types et caractéristiques de chacune.

Tableau II.1 : Types et caractéristiques des cartes Arduino.

Arduino	Micro-contrôleur	EEPROM Ko	RAM Ko	Broches d'E/S numériques	Dimension (mm)
Mega 2560	AT mega 2560	4	8	54	101.6*53.3
Uno	AT mega 328	1	2	14	68.6 *53.3
Mega	ATmega 1280	4	8	54	101.6*53.3
Diecimila	ATmega 168	0.5	1	14	68.6*53.3
Nano	ATmega 328	0.5/1	1/2	14	43*18
LilyPad	ATmega 168	0.5	1	14	50
Fio	ATmega328	1	2	14	40.6*27.9
Duemilanove	ATmega158	0.5	1/2	14	68.6*53.3



Figure II.1 : Les différentes cartes Arduino selon leurs tailles.

5- Le fonctionnement de la carte Arduino :

Les différentes versions des Arduino fonctionnent sous le même principe:

- Le microcontrôleur : considéré comme le cerveau de la carte.
- L'alimentation : C'est celle du microcontrôleur : 5V régulée (port USB) ou 7 à 12V provenant d'alimentation externe.
- Visualisation : par des LED de taille millimétrique. Elles sont là pour tester le matériel avec le branchement du microcontrôleur et les autres LED sont pour émission et réception du programme dans le microcontrôleur.
- Les connecteurs sont :
 - **A** : 0 à 13 ce sont les broches E/S numériques, elles offrent en sortie du 5 V et acceptent en entrée du 5 V sur le même principe.
 - **B** : A0 à A5 ce sont les broches E/S analogiques, valeur entre 0 V et 5 V.
 - **C** : les différentes broches d'alimentation :
 - Jaune : entrée reliée à l'alimentation (7 V-12 V).
 - Noire : les masses (-).
 - Rouge : sortie 5 V (+).
 - Orange : sortie 3,3 V (+).

- Il y a des variations entre les différentes cartes. (Par exemple : UNO, la patte 13 est équipée d'une résistance).

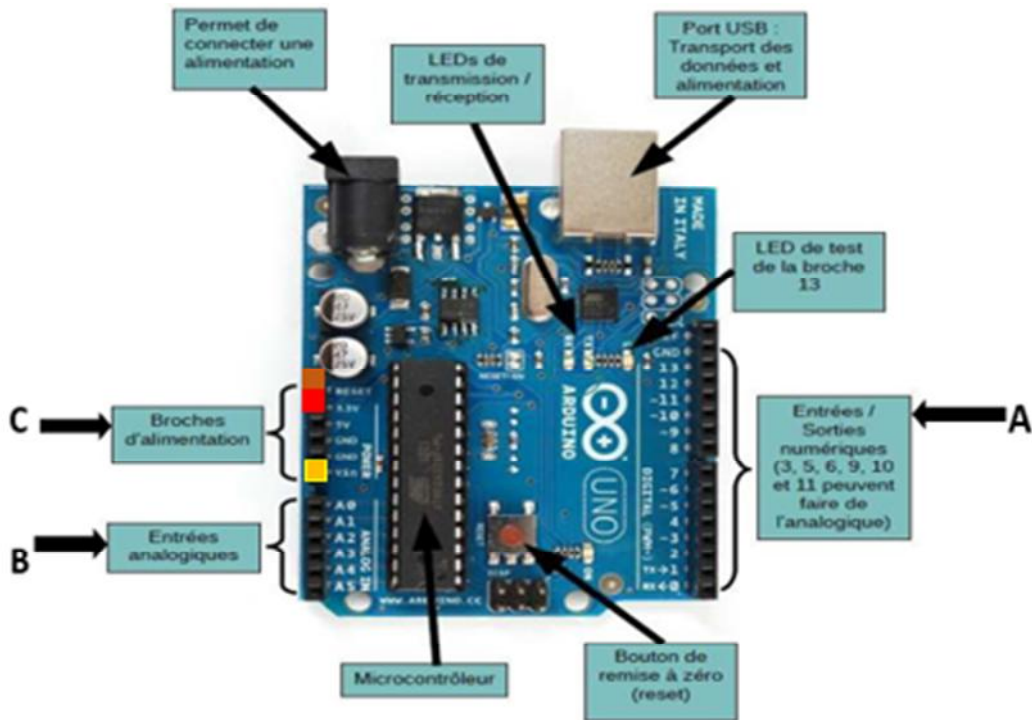


Figure II.2 : Les composants de la carte Arduino UNO.

6- Langage de programmation :

6.1- 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 intégré) 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.

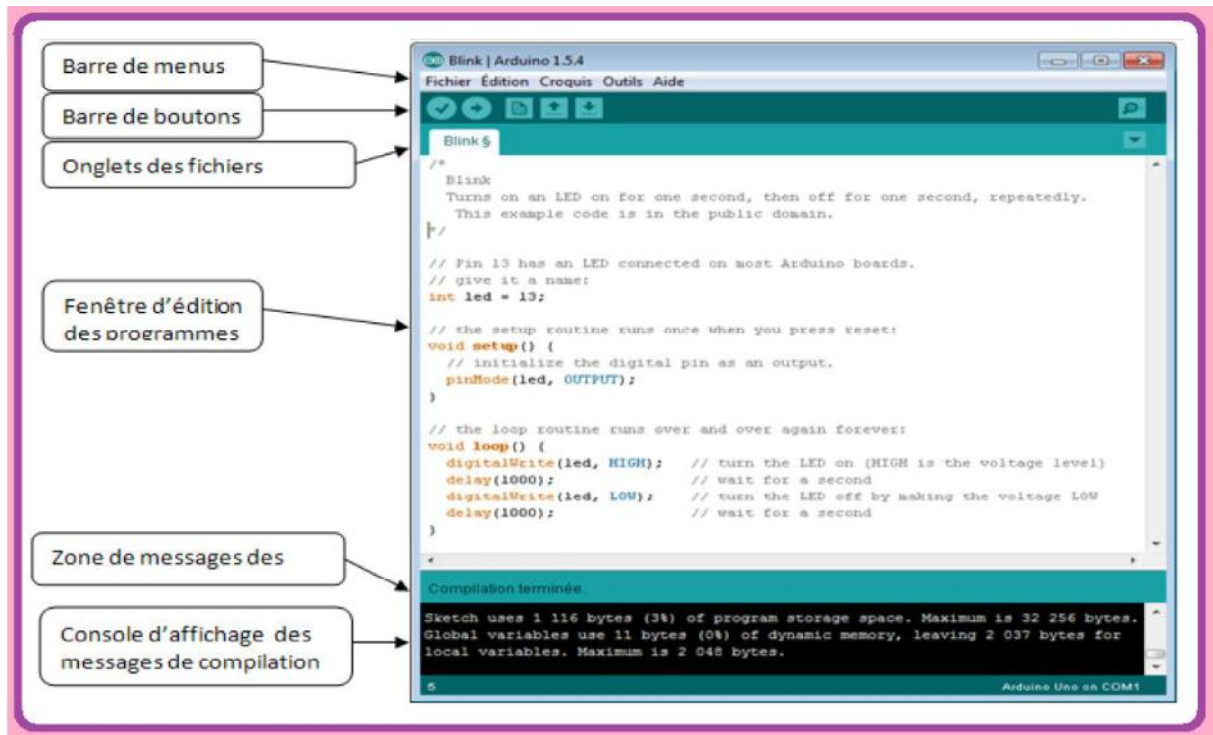


Figure II 3 : Les différentes parties de la fenêtre principale du logiciel Arduino.

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

➤ Une barre de menu:

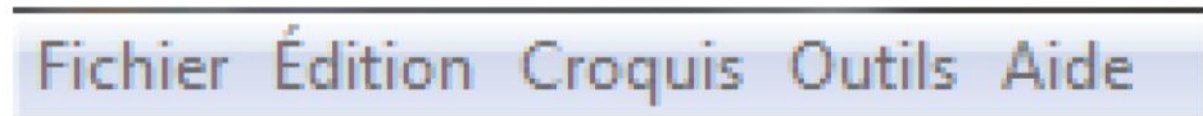


Figure II 4 : La barre de menu.

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 (Editer) :** 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 de boutons, ainsi que les options d'ajouts de bibliothèques 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 du logiciel Arduino.



Figure II.5: Barre de boutons pour le programme C Arduino.



Vérifier / compiler : vérifie le code à la recherche d'erreur .



Transférer vers la carte : compile votre code et le transfère vers la carte Arduino.



Nouveau : crée un nouveau code (nouvelle fenêtre).



Ouvrir : ouvre la liste de tous les programmes dans le « livre de programme ».

Enregistrer : enregistre un programme.



Moniteur série : ouvre la fenêtre du moniteur série.

7- 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).

8- 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.

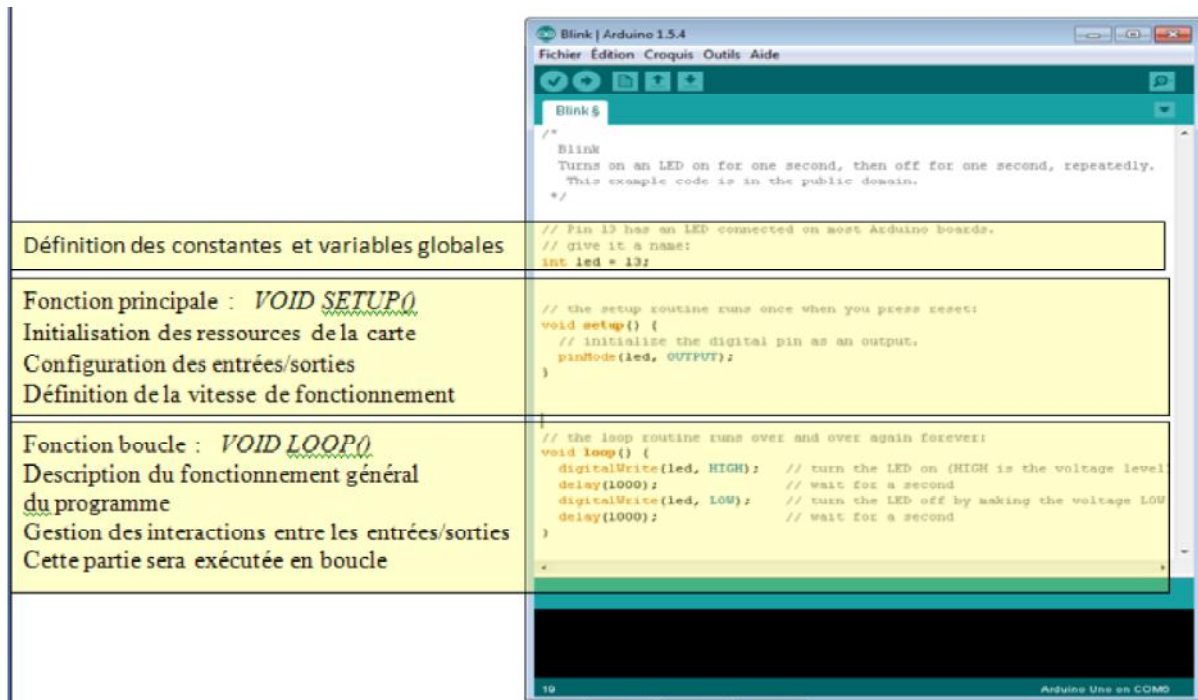


Figure II.6 : Différents composants du programme Arduino.

```
voidsetup() //fonction d'initialisation de la carte
{
    //contenu de l'initialisation
}

voidloop() //fonction principale qui se répète (s'exécute) à
l'infini
{
    //contenu du programme
}
```

8.1 - 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 `voidsetup()` ou `voidloop()` décrites en haut.

8.2- 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".

9- Commandes du programme Arduino:

➤ **Structure générale :**

- **voidsetup()** (configuration-préparation).
- **voidloop()** (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).

➤ Operations 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 'chaine de caractères').

- **array** = (tableau de 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 :

- **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).

10-Etapes à suivre pour programmer la carte Arduino:

a- Ecriture du programme :

Cette première étape consiste à saisir le programme.

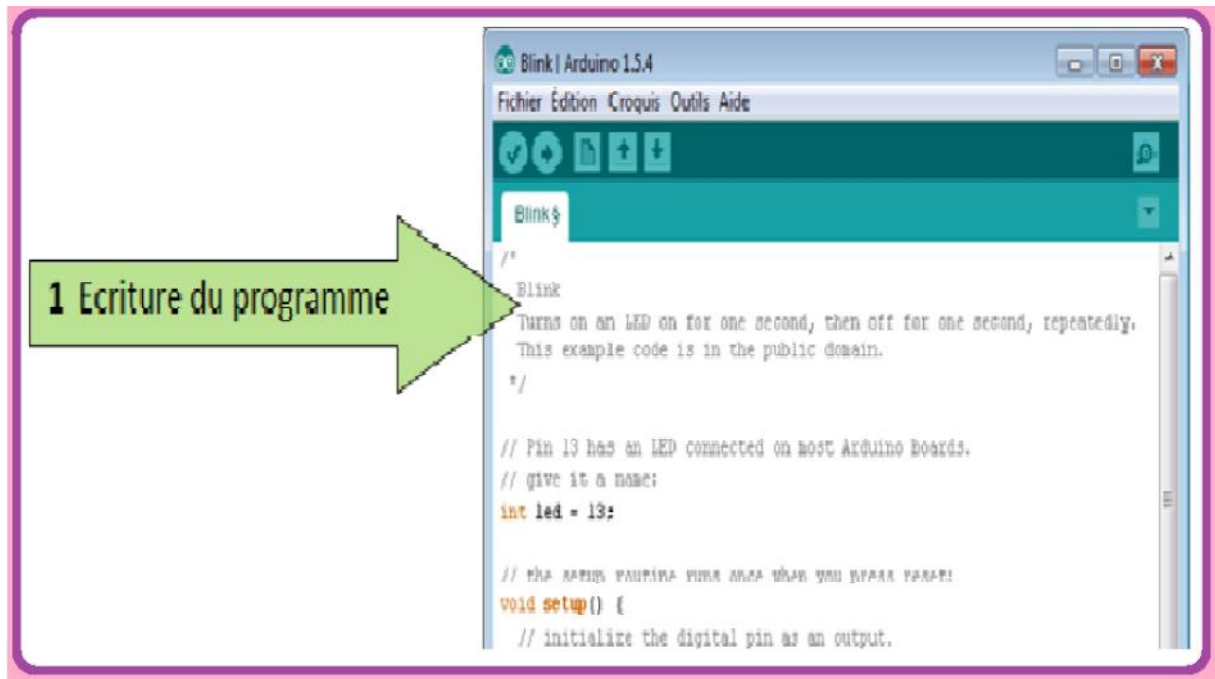


Figure II.7: L'écriture du programme.

b- 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 où elle se trouve, (Pour la correction des erreurs il suffit de consulter l'aide disponible dans la barre de menus).

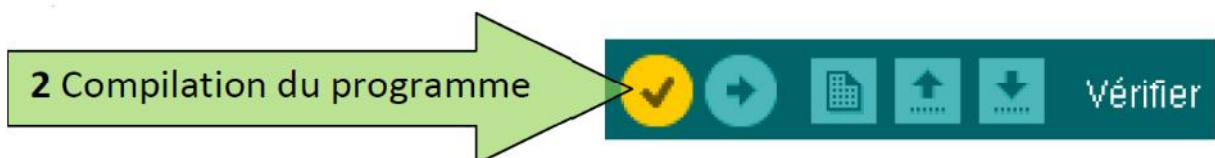


Figure II.8: Compilation du programme.

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

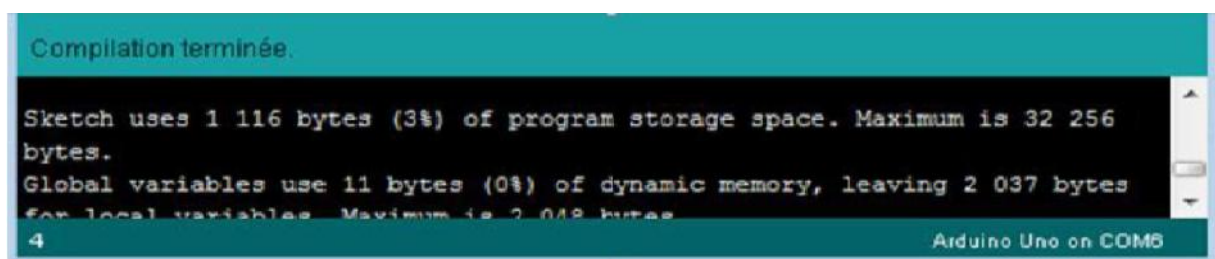


Figure II.9: Console d'affichage.

c- 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.

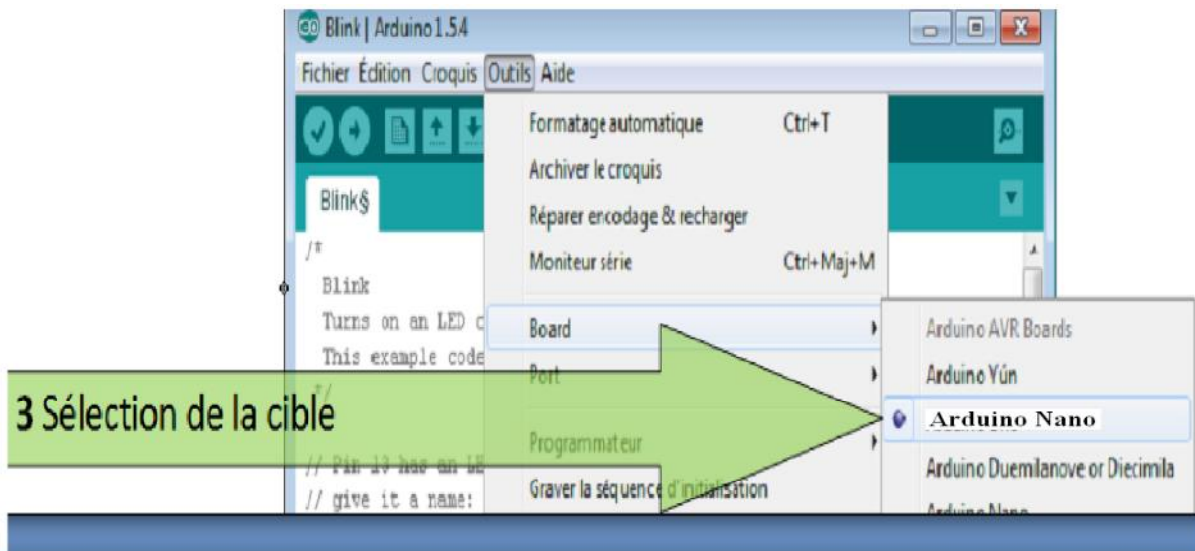


Figure II.10 : Sélection de la cible.

d- Sélection du port :

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

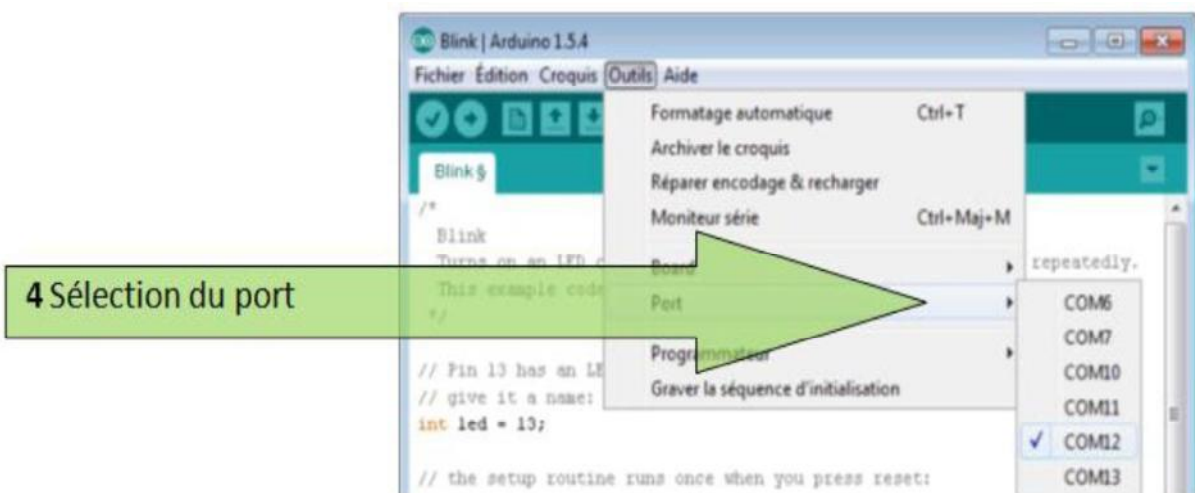
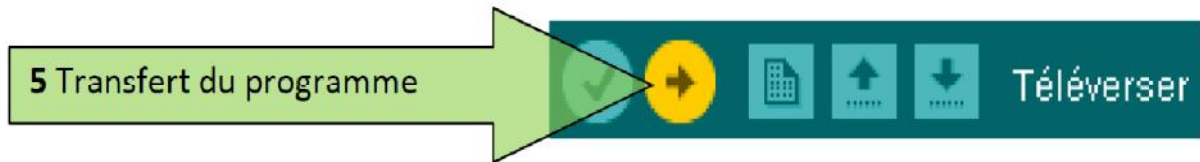


Figure II.11: Sélection du port.

e- **Transfert du programme :**

Cela se fait en cliquant sur le bouton flèche.



FigureII.12 : Bouton de transfert du programme.

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

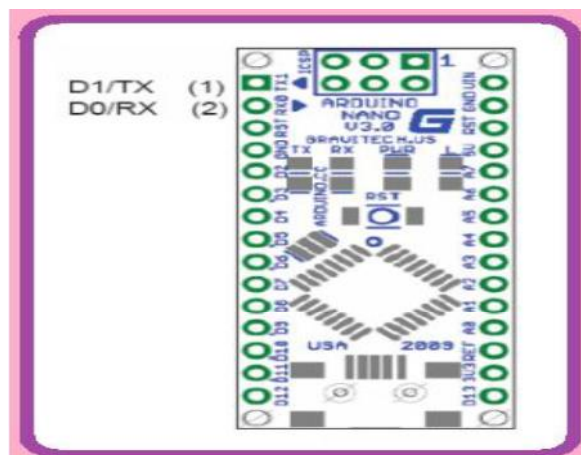


Figure II.13 : Port Rx, Tx communication série.

Sur la plupart des cartes, nous devons voir les **LEDs** des lignes **RX** et **TX** clignoter 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.

11- La communication entre le PC et la carte :

La carte Arduino UNO intègre un fusible qui protège le port USB de l'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 appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.



Figure II.14 : Communication carte Arduino et PC.

12 -Bonnes raisons de choisir Arduino :

Il existe pourtant dans le marché, une multitude de plateformes qui permettent de faire la même fonction. Notamment les microcontrôleurs « PIC » du fabricant Microchip. Nous allons voir pourquoi nous avons choisi d'utiliser Arduino :

- L'ATMEGA est assez complet facile à utiliser et plutôt, solide Grosse communauté
- Beaucoup de bibliothèques pour interfacier avec d'autres modules.
- 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.

13-Conclusion :

Dans ce chapitre nous avons présenté la carte Arduino qui est un mélange d'électronique et de programmation, en citant ces types, son langage de programmation, ainsi que les raisons qui nous ont poussé à l'utiliser pour la réalisation de notre projet.

Chapitre III

**Etude de la partie matérielle et
logicielle.**

1- Introduction :

Dans ce chapitre, nous allons définir et expliquer les caractéristiques des éléments électroniques que nous allons utiliser pour la commande et le fonctionnement du gilet intelligent, ainsi que les différents logiciels nécessaires pour la programmation de ce dernier.

Pour la réalisation du gilet nous allons utiliser l'ensemble de matériels suivant :

- Une carte arduino nano.
- Un module Bluetooth HC-05.
- Capteur de température et humidité DHT11.
- Un Accéléromètre GY 521.
- Un Buzzer.
- Un régulateur de tension 7805.

Et les logiciels suivant :

- Arduino.
- Le TCI.
- Application Android.

2- Conception matériel :**a- La carte Arduino Nano :**

La carte Arduino Nano est basée sur le microcontrôleur ATmega328 en version 3.0, c'est une version compacte et réduite de l'Arduino UNO.

Il dispose de 14 entrées/sorties numériques (dont 6 peuvent être utilisées comme sorties PWM), 8 entrées analogiques, d'un oscillateur à quartz de 16 MHz, d'une connexion USB Mini-B, un connecteur ICSP, et un bouton de reset.

Il contient tout le nécessaire pour piloter le microcontrôleur, il suffit simplement de le connecter à un ordinateur avec un câble USB pour l'utiliser.

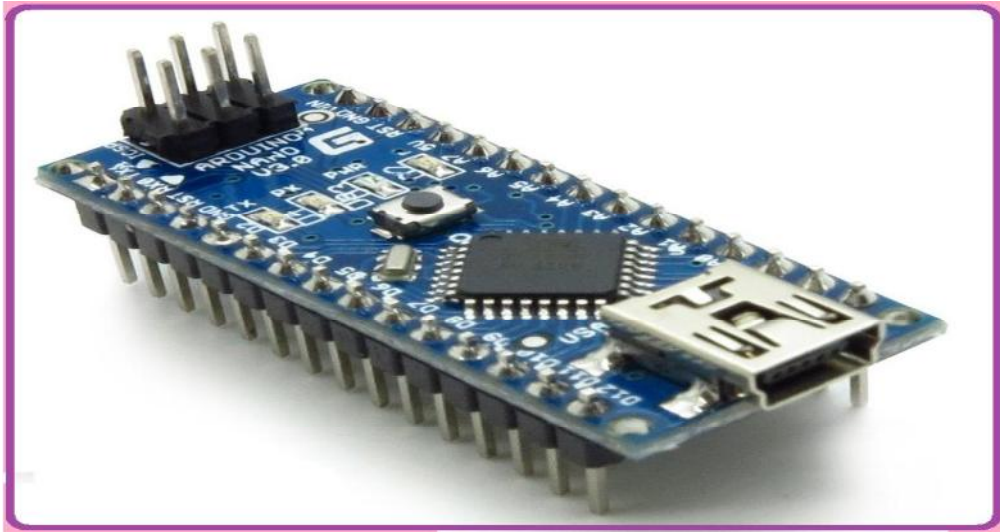


Figure III.1 : Vue de la carte Arduino Nano

➤ **Caractéristiques de la carte Arduino Nano :**

Tableau III.1 : Caractéristiques de la carte Arduino Nano

Caractéristique	NANO version3
Microcontrôleur	ATMEGA 328
Tension d'alimentation normale	5V
Tension d'alimentation (recommandée)	7----12V
E/S digitales	14 (dont 6 dispose d'une sortie PWM)
Entrées analogiques	8
Courant continu pour les E/S	40 mA
Courant continu pour la pin 3.3V	50mA
Mémoire Flash	32KB (ATMEGA 328)
SRAM	2KB
EEPROM	1KB
Fréquence d'horloge	16MHz

➤ Choix de la carte Arduino NANO :

Puisque dans notre réalisation nous n'allons pas dépasser trois capteurs, l'Arduino nano est la solution la plus optimale, car elle comporte huit (8) entrées analogique (plus que la UNO) avec un bas prix, ce qui facilitera la duplication sans oublier que sa taille est largement plus petite que la Méga et la UNO, ce qui la rend plus pratique pour notre réalisation.

b- 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.



Figure III.2 : Vue du Module Bluetooth HC-05

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

➤ **Caractéristique principale :**

- Fonctions maître / esclave.
- Tension de fonctionnement de 3.3V
- Tension d'entrée 3.3 à 6V.
- Taille 39 x 15 mm
- Intensité 30 mA en fonctionnement, 8mA pour établir la communication
- Utilisation pour communication par Bluetooth avec téléphones portables, tablettes, ordinateurs
- Une LED indique le fonctionnement : clignote si non connecté; éclairage fixe quand il est connecté.
- Utilise un régulateur 150mA 3.3V
- Brochage 6 pins : Vcc, Gnd, TxD, RxD

c- Le capteur DHT 11 :

Le DHT11 est un capteur de Température et d'Humidité très apprécié pour sa simplicité de mise en œuvre et son coût peu élevé. Il ne requiert qu'une résistance de tirage et une alimentation 3V ou 5V pour fonctionner.

Le tableau (III.2) montre les caractéristiques du capteur DHT11 :

Tableau III.2 : Les caractéristiques du capteur DHT11

Caractéristiques	DHT11
Humidité (relative %)	20 ~ 80%
Précision (humidité)	+/- 5%
Température	0 ~ +50°C
Précision (température)	+/- 2°C
Fréquence mesure max	1Hz (1 mesure par seconde)
Tension d'alimentation	3 ~ 5 volts
Stabilité à long terme	+/- 1% par an

➤ **Câblage d'un capteur DHTxx :**

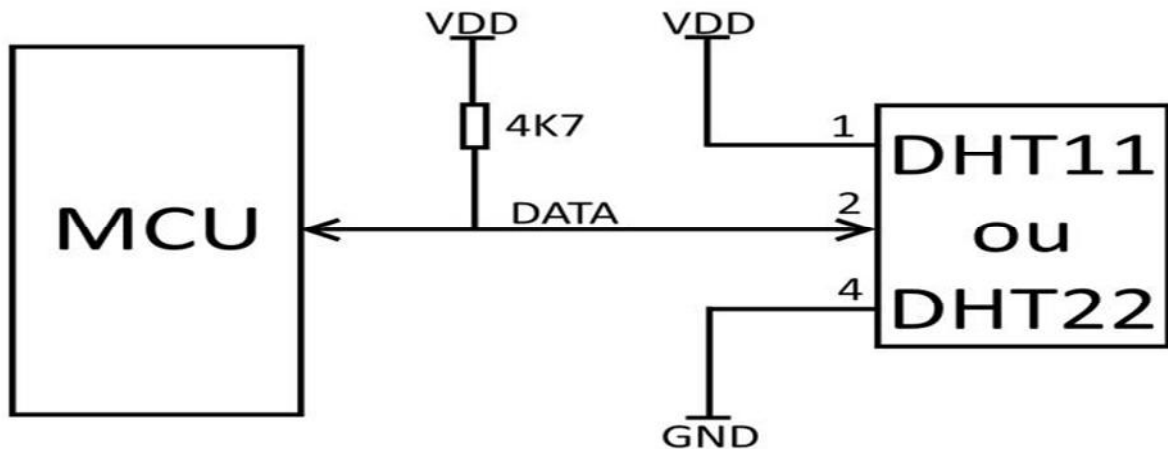


Figure III.3 : Câblage d'un capteur DHTxx

Le brochage du capteur est le suivant :

- La broche n°1 est la broche d'alimentation (5 volts ou 3.3 volts).
- La broche n°2 est la broche de communication. Celle-ci doit impérativement être reliée à l'alimentation via une résistance de tirage de 4.7K ohms (il s'agit d'une sortie à collecteur ouvert).
- La broche n°3 n'est pas utilisée et ne doit pas être câblée.
- La broche n°4 est la masse du capteur (GND).

➤ **Le protocole de communication :**

Les capteurs DHTxx ont la particularité de communiquer avec le microcontrôleur via une unique broche d'entrée / sortie.

d- GY521 :

Le module GY-521 est une carte de dérivation pour les MEMS MPU-6050 (systèmes microélectromécaniques) qui comprend un gyroscope à trois axes, un accéléromètre à trois axes, un processeur de mouvement numérique (DMP). Le processeur de mouvement numérique peut être utilisé pour traiter des algorithmes complexes directement sur la carte. Habituellement, le DMP traite les algorithmes qui transforment les valeurs brutes des capteurs en données de position stables.

➤ **Constitution du GY 521:**

La casse GY-521 a huit broches :

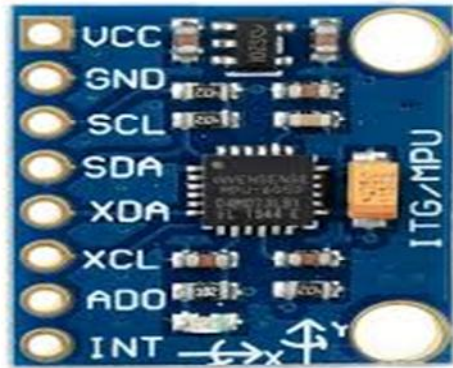


Figure III.5 : Le GY-521

VCC : (La carte de dérivation est équipée d'un régulateur de tension, ce qui permet la connexion de la carte à des sources de 3,3 V et 5 V).

GND : la masse

SCL : (Ligne d'horloge série du protocole I2C.)

SDA :(Ligne de données série du protocole I2C.)

XDA : (données auxiliaires => données série maître I2C pour la connexion du module à des capteurs externes.)

XCL :(Horloge auxiliaire => Horloge série maître I2C pour la connexion du module aux capteurs externes.)

AD0 : (Si cette broche est BAS, l'adresse I2C de la carte sera 0x68, sinon, si la broche est HAUTE, l'adresse sera 0x69.)

INT :(Sortie numérique d'interruption).

e- Le buzzeur :

Un buzzeur ou un bippeur est un élément électromécanique ou piézoélectrique qui produit un son caractéristique quand on lui applique une tension : le bip. Certains nécessitent une tension continue, d'autres nécessitent une tension alternative.



Figure III.6 : Le buzzer.

f- Le régulateur de tension 7805 :

Le régulateur de tension 7805 est l'un des circuits intégrés populaire qui peut être trouvé dans presque tous les circuits électroniques. Comme son nom l'indique il est utilisé pour la régulation de tension. Il fournit une tension de sortie constante pour une large gamme de tensions d'entrée des variables.

➤ Signification de l'appellation du régulateur de tension 7805 :

Le régulateur de tension 7805 appartient à la famille 78xx. Il est aussi appelé LM 7805, L 7805, IC 7805 etc.

Les deux derniers chiffres (05) désignent la tension de sortie du circuit intégré, ce qui veut dire qu'à la sortie du régulateur 7805 on aura une tension constante qui sera égale à 5V quelles que soient les tensions des variables d'entrée.

La valeur maximale de la tension d'entrée du circuit intégré qui peut supporter est de 24 volts.

➤ Constitution du régulateur 7805:

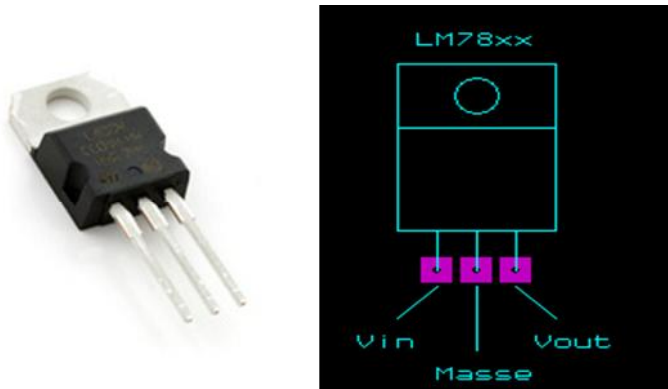


Figure III.7 : Le régulateur de tension 7805

Le régulateur de tension 7805 a trois broches :

- 1- Tension d'entrer (5V—24V)
- 2- GND (la masse 0V)
- 3- Tension de sortie 5V (4,8V—5,2V)

➤ **L'importance du régulateur de tension 7805:**

Parce qu'il est nécessaire d'ajuster la tension dans un circuit électronique le 7805 est utilisé pour maintenir un niveau de tension de l'état stationnaire dans un circuit électronique. Les fluctuations de tension dans un circuit peuvent endommager les composants de circuits et de réduire la durée de vie de l'équipement. Il est donc essentiel de fournir une tension constante au circuit.

3- **La conception logicielle:**

a- **Logiciel Arduino :**

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateforme, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série. Il est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande.



Figure III.8 : Logiciel Arduino.

➤ **Le langage de programmation :**

Le langage de programmation utilisé est le C++, compilé avec avr-g++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino, à toute personne maîtrisant le C ou le C++.

➤ **Les éléments de l'interface logicielle Arduino :**

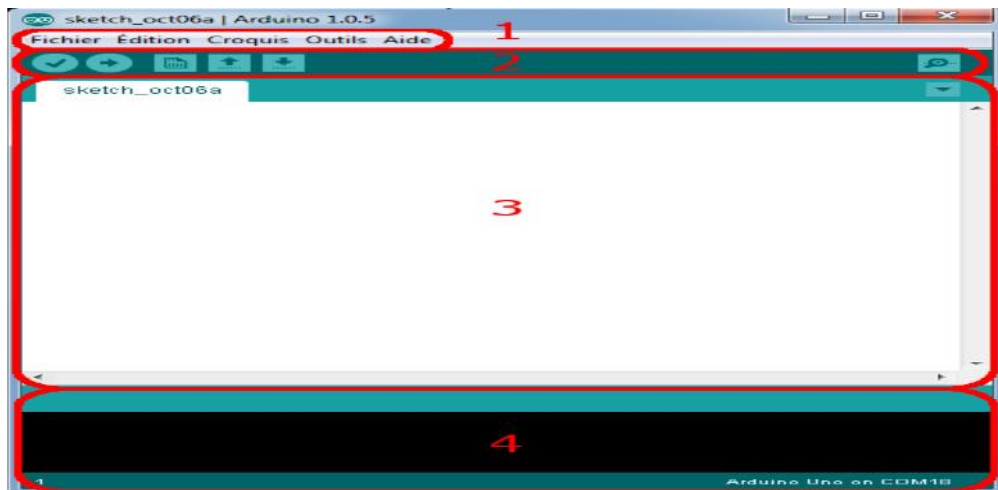


Figure III.9 : L'interface logicielle Arduino

- 1- : Ce sont les options de configuration du logiciel.
- 2- : Les boutons qui nous servir lorsque on programmer la cartes.
- 3- : Ce bloc va contenir le programme que nous allons créer.
- 4- : Il indique les erreurs pour les corriger les fautes dans notre programme.

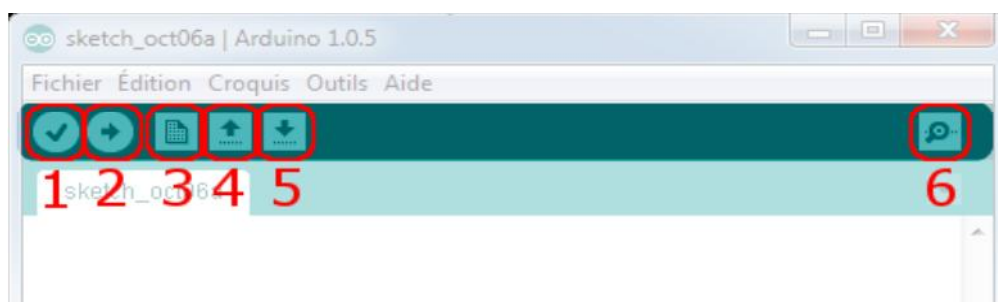


Figure III.10 : Les boutons d'interphase de programmation Arduino.

- Bouton 1 : Compiler le programme, et actionne un module qui cherche les erreurs dans votre programme.
- Bouton 2 : Charger le programme dans la carte Arduino.
- Bouton 3 : Créer un nouveau fichier.
- Bouton 4 : Ouvrir un fichier.
- Bouton 5 : Enregistrer le fichier.
- Bouton 6 : Ouvrir le moniteur série.

➤ **La structure d'un programme Arduino :**

Elle est composée de 3 zones:

- **Zone globale :** Elle permet d'appeler les bibliothèques, créer les variables globales, créer les nouvelles fonctions ou procédures pour votre programme.
- **Zone setup:** Une fonction "*void setup ()*" permet d'initialiser des éléments, variables. Cette fonction n'est appelée qu'une seule fois au boot par l'arduino.
- **Zone loop:** Une fonction "*void loop ()*" est appelée constamment par l'arduino et après la fonction setup. Le cœur du programme est là.

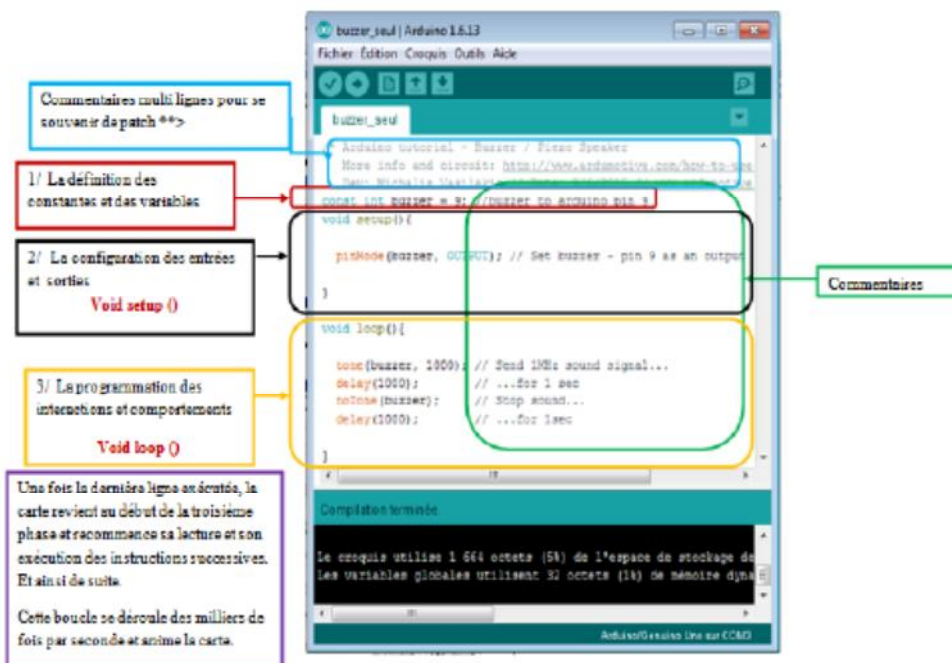


Figure III.11 : La structure d'un programme Arduino

➤ Applications d'Arduino :

Le système Arduino nous permet de réaliser un grand nombre d'applications dans tous les domaines, l'étendue de l'utilisation de ce système est gigantesque. Pour quelques exemples, nous pouvons:

- ✓ Contrôler les appareils domestiques ;
- ✓ Fabriquer un robot ;
- ✓ Faire un jeu de lumières ;
- ✓ Communiquer avec l'ordinateur ;
- ✓ Télécommander un appareil mobile (modélisme) ;
- ✓ Construire un appareil de mesure de tension/courant ;
- ✓ Fabriquer une imprimante 3D ;
- ✓ Etc.

b- Logiciel TCI :

Logiciel pour les électroniciens, destinés à la conception de circuits imprimés. Ce logiciel permet de tracer son circuit simple ou double face rapidement et d'imprimer chacune des couches sur papier ou sur transparent bureautique (pour obtenir un film directement exploitable). Fini le collage de pastilles et de rubans sur mylar. La taille du circuit est limitée par la feuille d'imprimante.

c- Application androïde :

Les Smartphones sont des objets techniques très présents dans le quotidien de la personne.

Un terminal mobile a besoin d'un système d'exploitation pour fonctionner.

Androïde, est un système d'exploitation mobile Pour Smartphones, tablettes tactiles. Il utilise un noyau Linux qui est un système d'exploitation libre.

1. L'androïde :

Androïde est un système d'exploitation développé initialement pour les Smartphones. Il utilise un noyau Linux qui est un système d'exploitation libre pour PC. Il intègre tous les utilitaires nécessaires à un Smartphone. Il est optimisé pour les outils Google.

Les applications sont exécutées par un processeur de type ARM à travers un interpréteur Java.

2. App inventor 2:

App Inventor est un outil de développement en ligne pour les téléphones et les tablettes sous Android. App Inventor est un OS créé par Google, et concurrent de l'Ios d'Apple qui équipe l'iPad, iPodTouch et iPhone. La plateforme de développement est offerte à tous les utilisateurs possédant un compte Gmail. Elle rappelle certains langages de programmation simplifiés des années 80 et s'inspire des travaux d'une étudiante en thèse au MIT, Ricarose Roque. Le projet a été dirigé par Hal Abelson, professeur au MIT qui a mis au point le portage du langage LOGO sur Apple 2 au début des années 80.

2.1- Le concept d'App Inventor :

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.

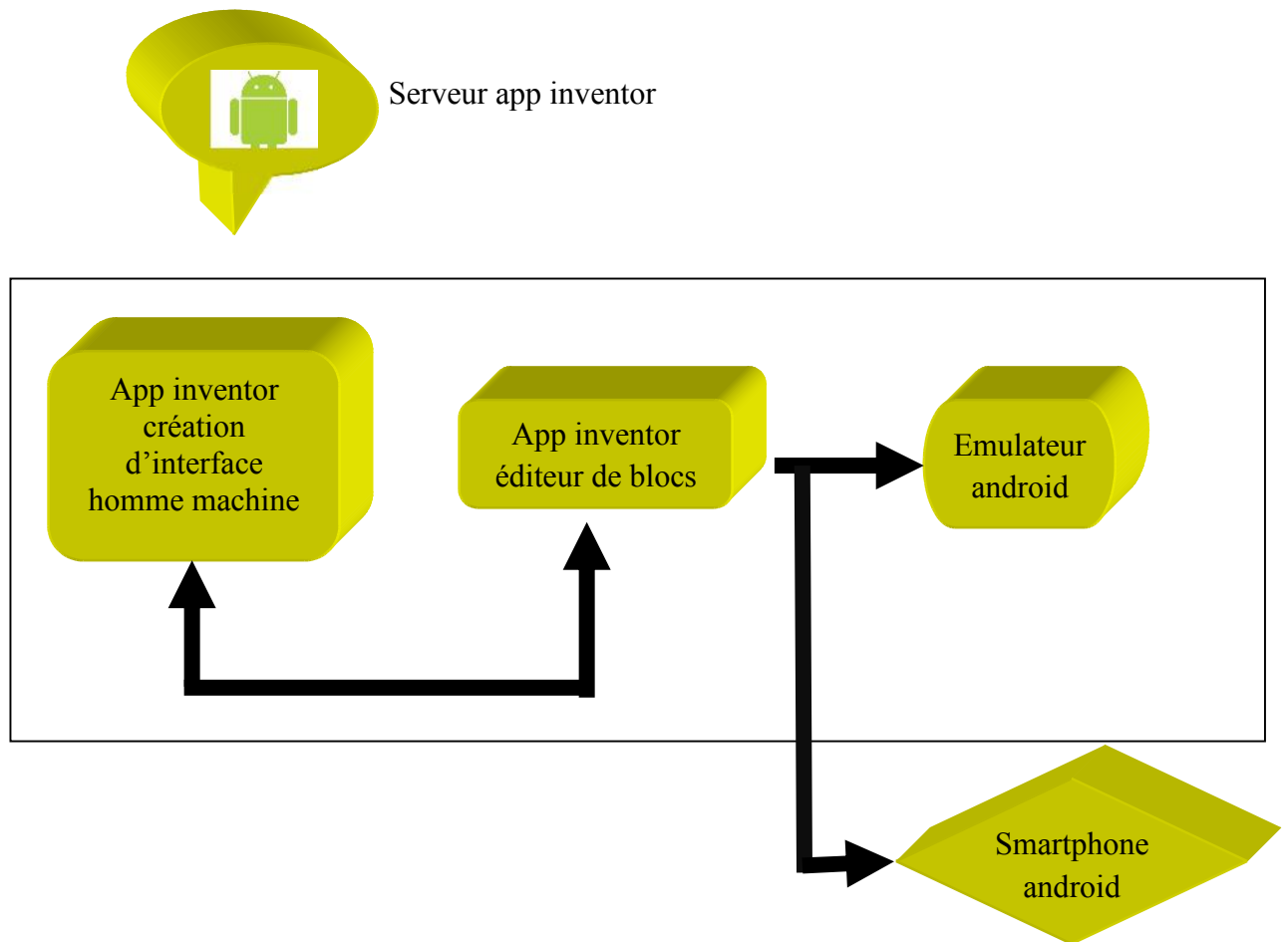


Figure III.12 : Le concept d'App Inventor

Trois fenêtres sont proposées pendant le développement :

- ✓ Une pour la création de l'interface homme machine : 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 ;
- ✓ Et 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 pour un test réel. Ce terminal pourra aussi bien être un téléphone qu'une tablette, le comportement du programme sera identique.

2.2- La fenêtre de la création de l'interface :

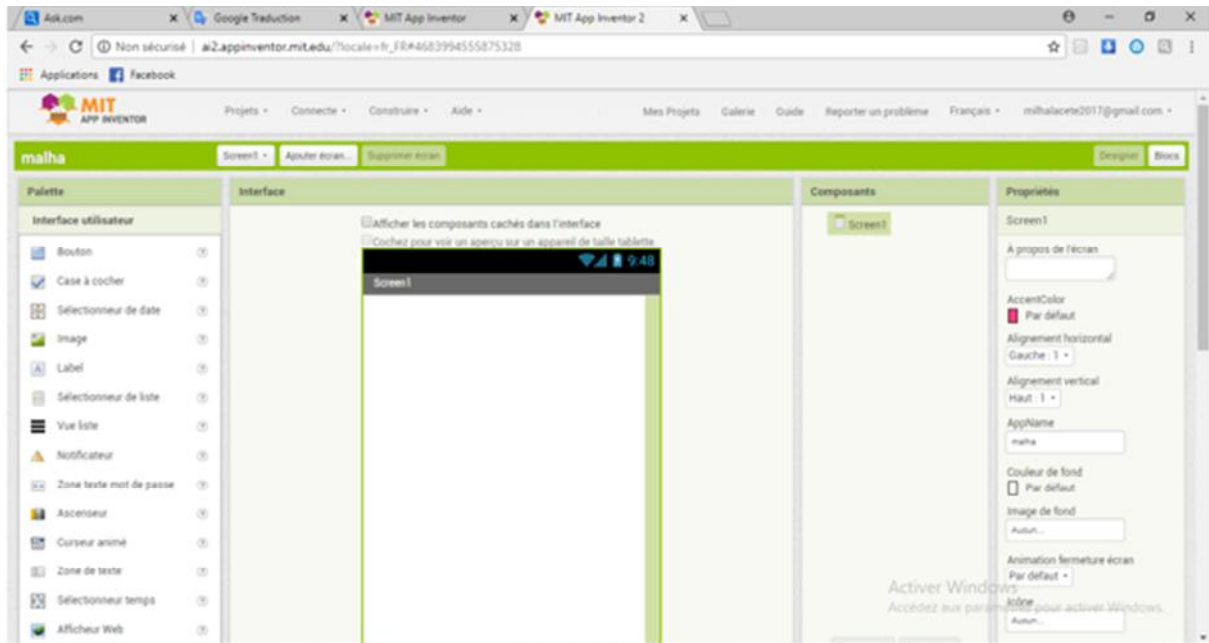


Figure III.13 : L'interface d'app inventor.

a- La fenêtre de programmation graphique

Cette fenêtre est l'espace de programmation on glisse les blocs et on les rassembler



Figure III.14 : La fenêtre de programmation graphique.

b- L'émulateur pour tester le programme

L'émulateur qui permettra de tester l'application et il permet de remplacer un terminal réel pour vérifier le bon fonctionnement du programme.

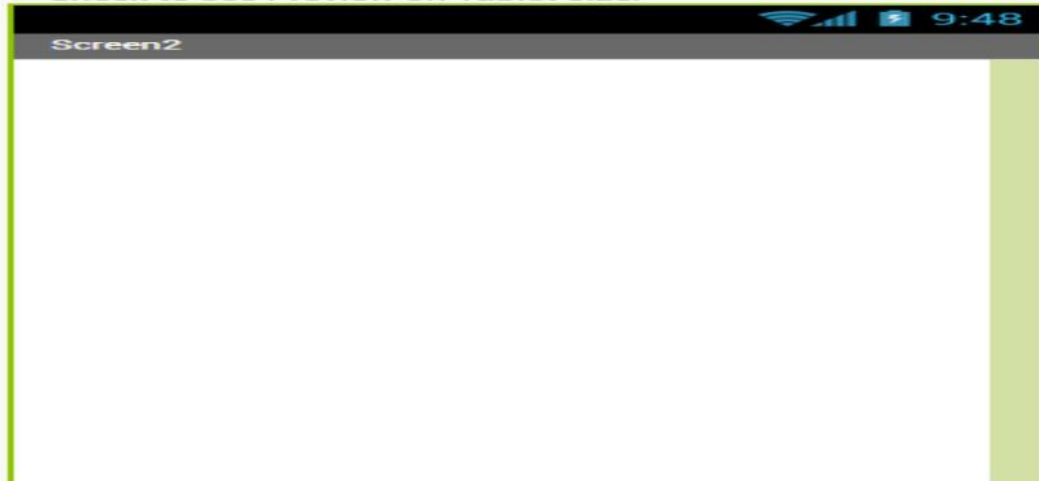


Figure III.15 : L'émulateur pour tester le programme

c- Présentation des différents menu :

La table menue donne l'ensemble des outils et des éléments qu'on peut utiliser dans la création d'une application.

Le menu « Project »	Le menu « Connect »	Le menu « Build »	Le menu « Help »
<ul style="list-style-type: none"> Project - Conne My Projects New ... Import ... Delete Save Save As... Checkpoint ... Export Export all Import Keystore Export Keystore Delete Keystore 	<ul style="list-style-type: none"> Connect - Build - AI Companion Emulator USB Reset Connection 	<ul style="list-style-type: none"> Build - Help - App (provide QR code) App (save to my computer) 	<ul style="list-style-type: none"> Help - About Library Get Started Tutorials Troubleshooting Forums Report an Issue

Figure III.16 : Les menus de la fenêtre de travail Androïde

- **Le menu Project** : Permet de visualiser les projets réalisés (My Projects), de réaliser un nouveau projet (New), d'importer un projet, de sauvegarder un projet ...
- **Le menu connecte** : Permet de visualiser le résultat du projet soit sur l'émulateur, soit sur le Smartphone ou la tablette par l'intermédiaire du Wifi, soit par l'intermédiaire d'un câble USB.
- **Le menu build** : Permet de construire le QR code de l'application ou de sauvegarder l'application sur votre ordinateur.
- **Le menu help** : Permet de retrouver un certains nombres de ressources d'aides.

2.3- Les étapes de développement d'application androïde avec app inventor :

2.3.1- La création de son interface :

Pour cela, le site affiche un écran de téléphone dans lequel nous pouvons placer les éléments que nous voulons utiliser.

Il y a des éléments graphiques comme des boutons, des éléments non graphiques comme des capteurs ou des fonctions permettant d'effectuer des actions : lecteur audio, appareil photo...

2.3.2- La conception de l'écran se partage en 4 parties :

➤ **La liste des éléments et des medias :**

Dans cette liste on trouve les éléments et les médias utilisés sur l'écran

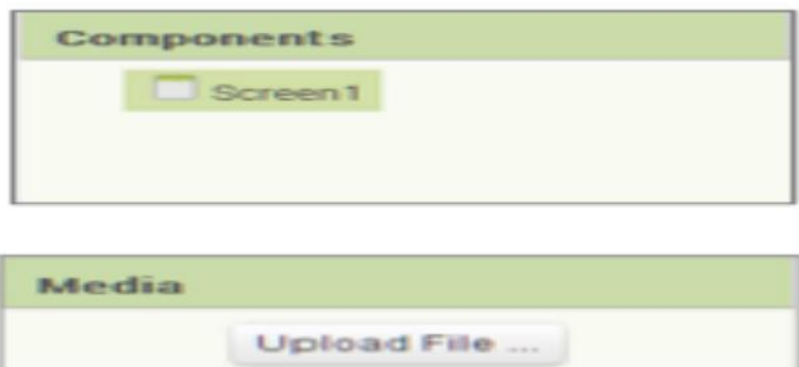


Figure III.17 : La liste des éléments et des medias.

➤ **La palette des éléments :**

La palette contenant tous les éléments qui peuvent être positionnés sur l'écran du téléphone.

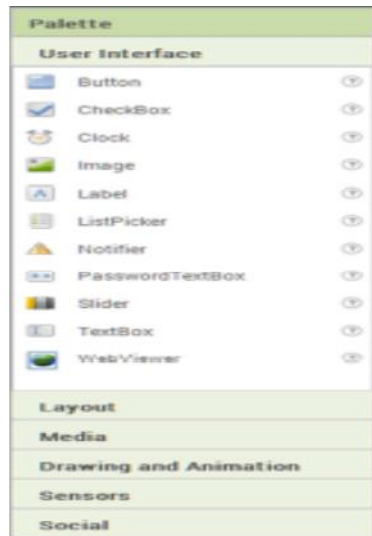


Figure III.18 : La palette des éléments.

➤ **L'écran d'affichage des éléments :**

Cet écran permet l'affichage des éléments nécessaires dans l'application.



Figure III.19 : L'écran d'affichage des éléments.

➤ **Les propriétés des différents éléments :**

Cette liste représente les différentes propriétés des éléments d'application.



Figure III.20 : Liste des propriétés

2.3.3- Le comportement de l'application :

Une fois l'allure de notre application créée, on va décrire le comportement de chaque élément. Pour cela, il faut cliquer sur « Blocks » en haut et à droite de la page :



Figure III.21 : L'interface de programmation de l'application.

Nous avons un système d'onglets. Nous y retrouvons les différents composants que nous avons placé sur l'écran ainsi que des blocks utilities « Built in ». Dans l'onglet «Screen1» nous retrouvons les éléments de l'interface.

➤ Réalisation de l'algorithme à l'aide des blocks :

On clique sur les éléments dans l'onglet « Screen » et on décrit le comportement de chaque élément.

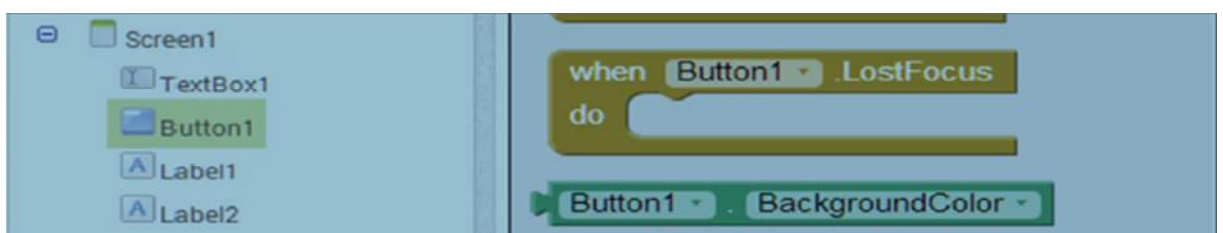


Figure III.22 : Les blocks de l'algorithme.

- ✓ Une fois que l'algorithme a été entièrement réalisé, on sauvegarde le programme.

➤ **Tester l'application sur l'émulateur ou le Smartphone :**

Afin de tester l'application on choisit le test sur :

AI Companion : dans ce cas, la connexion se fera directement sur le Smartphone en wifi,

L'émulateur : un écran s'affichera sur l'ordinateur,

USB : la connexion se fera sur le Smartphone via un Câble USB.

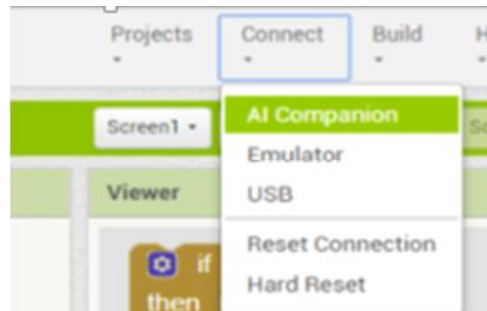


Figure III.23 : Test de fonctionnement de l'application

➤ **Installation :**

Sur le Smartphone, l'application MIT AI2 Companion doit être téléchargée et installée sur le playStore.



Figure III.24 : Téléchargement de l'application

4- Conclusion :

Dans ce chapitre, nous avons donné quelques définitions sur le matériel électronique en présentant leurs logiciels de fonctionnement que nous allons utiliser dans la partie pratique qui est la réalisation de ce gilet intelligent, un projet qui sera détaillé dans le chapitre qui suit (chapitre 4).

Chapitre IV

Réalisation pratique du projet.

1- Introduction :

Dans ce chapitre nous allons procéder à la réalisation du « gilet intelligent ». Le gilet est réalisé en deux parties, une partie électronique à base d'une carte Arduino Nano qui va récolter l'information par un capteur DHT11 et l'accéléromètre/Gyroscopie GY521, et une application installée sur le Smartphone (le système Android).

2- Schéma explicatif du fonctionnement du gilet intelligent :

La carte électronique se trouve à l'intérieur du gilet. Elle est composée d'une carte Arduino Nano et de capteurs reliés à un Bluetooth. Avec lequel la carte va se connecter au smart phone de la personne âgée.

Lorsque le gilet capte une température élevée ou baissée ou en cas d'une chute, il donnera l'information au Smartphone de la personne âgée et se connectera au GSM pour envoyer un message au médecin.

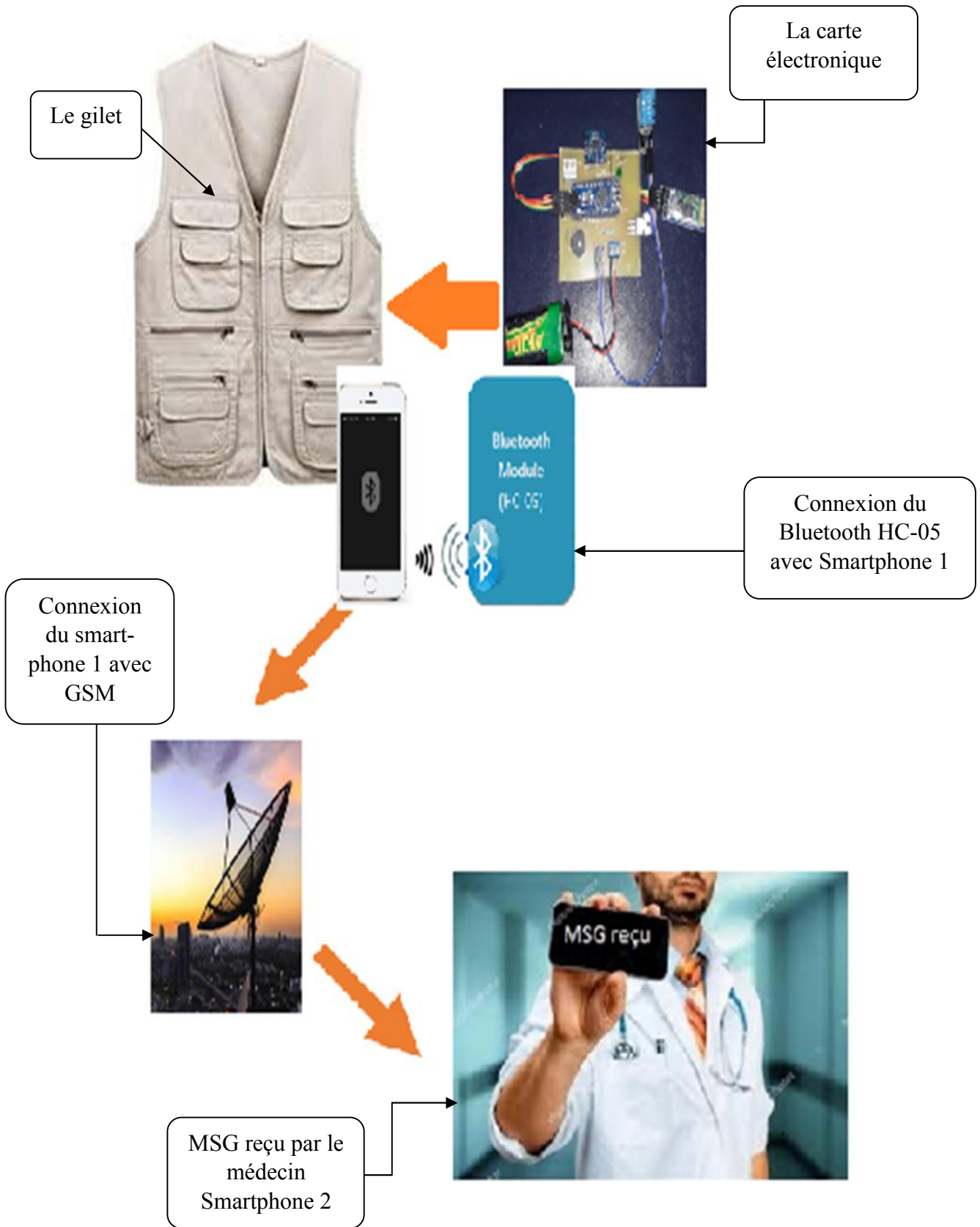


Figure IV.1: Schéma explicatif du fonctionnement du gilet intelligent.

3- Schéma synoptique global :

La figure (IV.2) montre le schéma synoptique du fonctionnement de notre application.

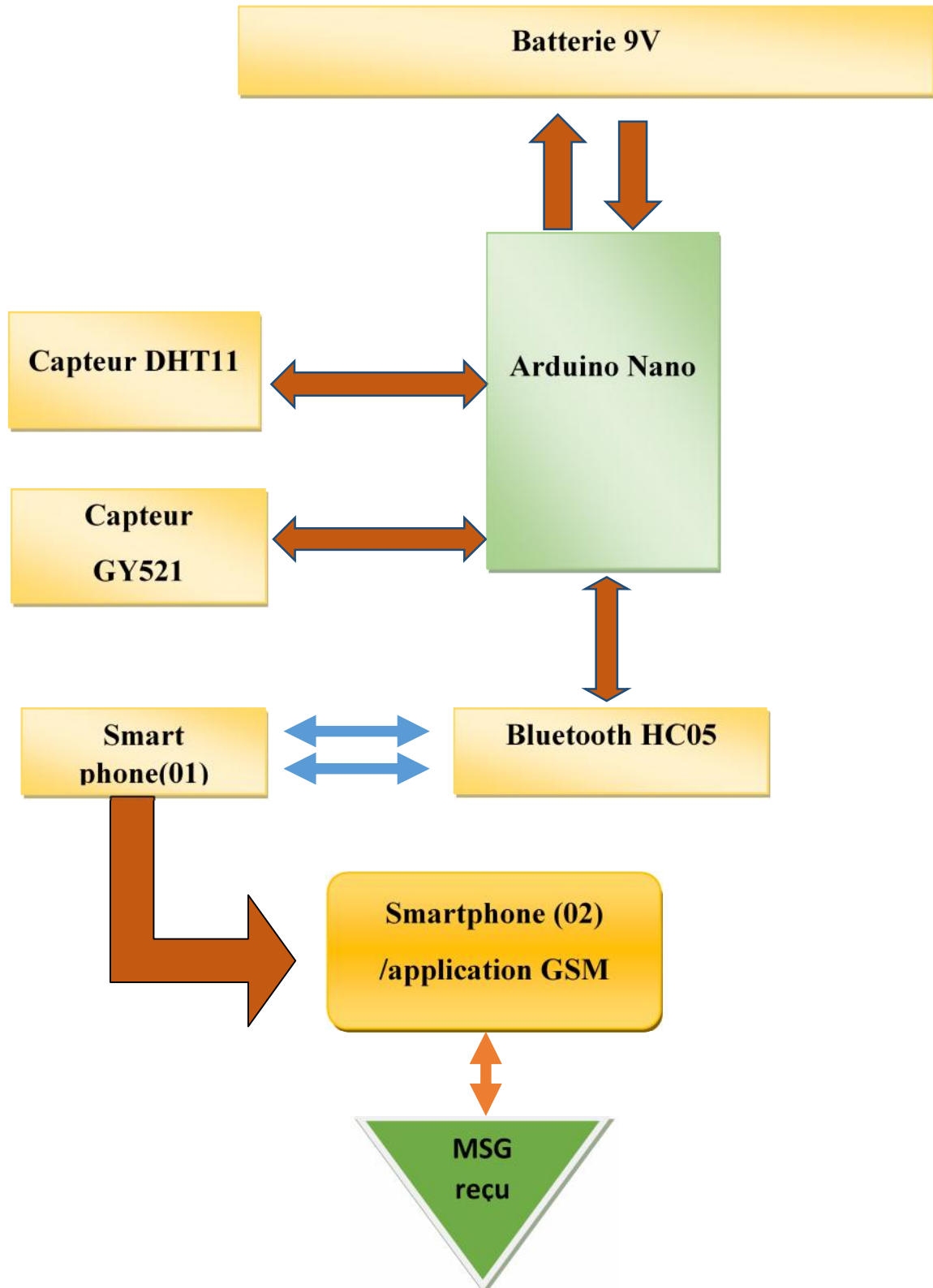


Figure IV.2 : Schéma synoptique général de la réalisation

La batterie 9v alimente la carte électronique, cette dernière est un socle pour la carte Arduino Nano, DHT11, GY521 et le Bluetooth.

Le Bluetooth va se connecter au Smartphone 1 qui avec le réseau GSM va envoyer un message au médecin (Smartphone 2).

4- La réalisation du projet :

A- Partie électronique :

1-Realisation de la Carte électronique :

➤ **Etapes de fabrication de la carte :**

Le circuit électrique que nous avons réalisé avec le logiciel TCI :

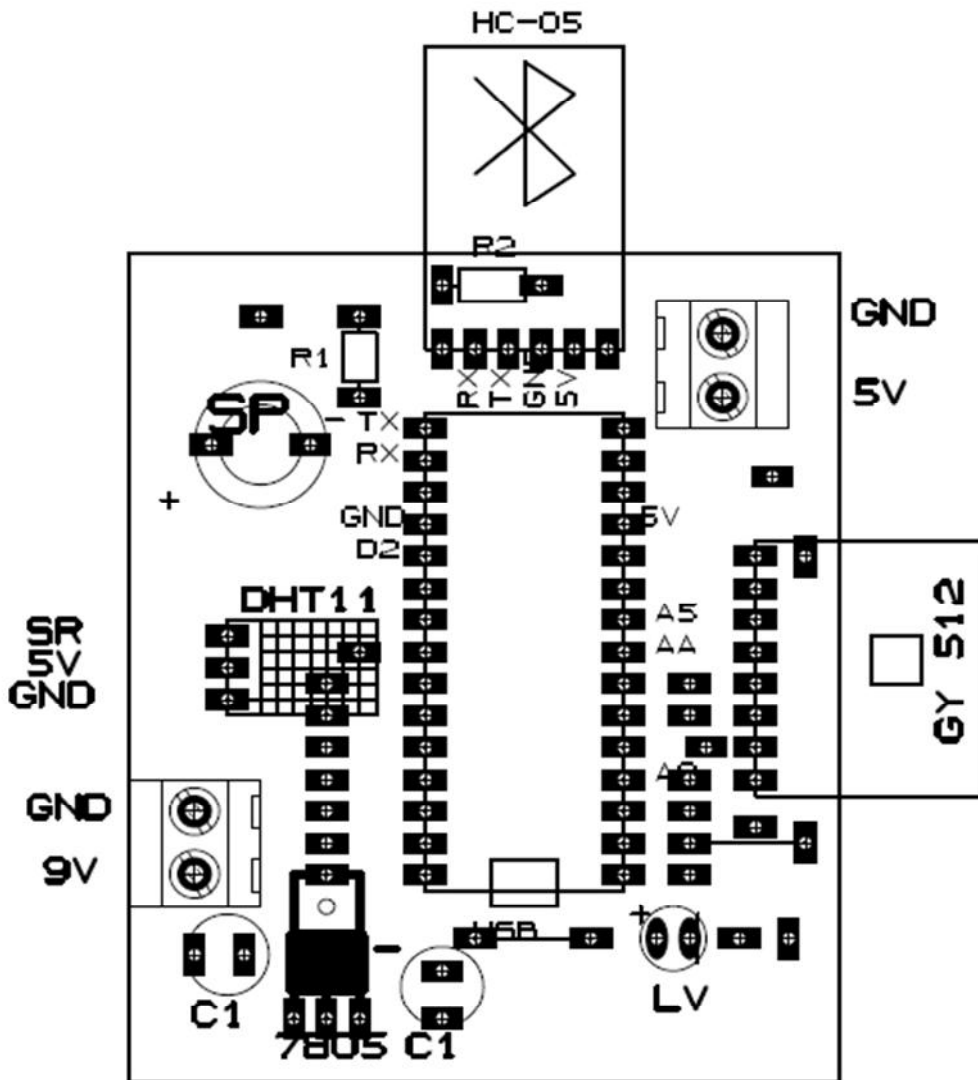


Figure IV. 3 : Circuit électrique de la carte électronique.

Le logiciel TCI (tracé du circuit imprimé) est conçu pour dessiner manuellement des circuits imprimés en vue de les imprimer sur papier ou sur le transparent

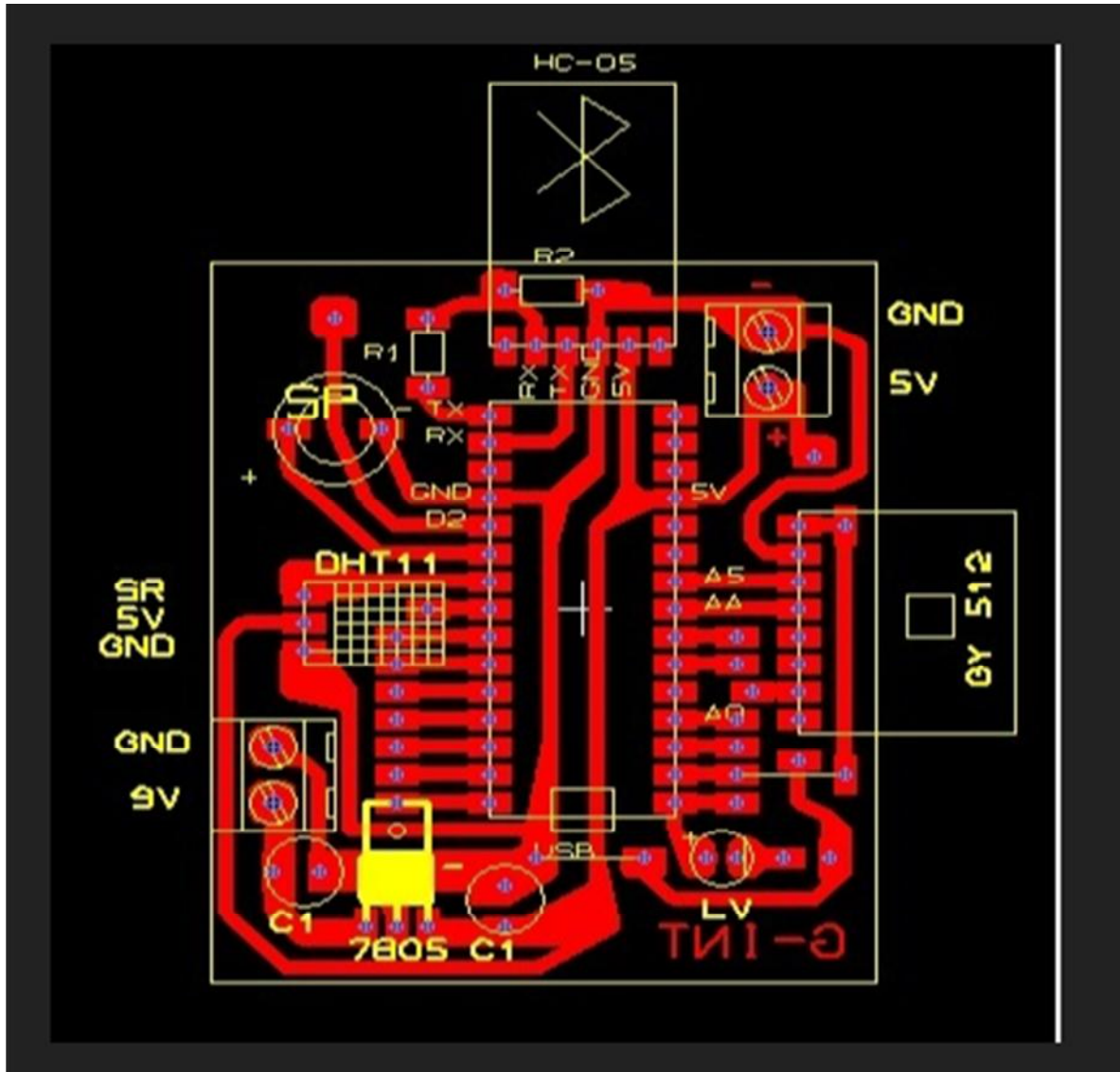


Figure IV.4 : Circuit sur le logiciel TCI.

Nous obtenons notre typon que nous avons imprimé sur le transparent ainsi que la carte ci-après :

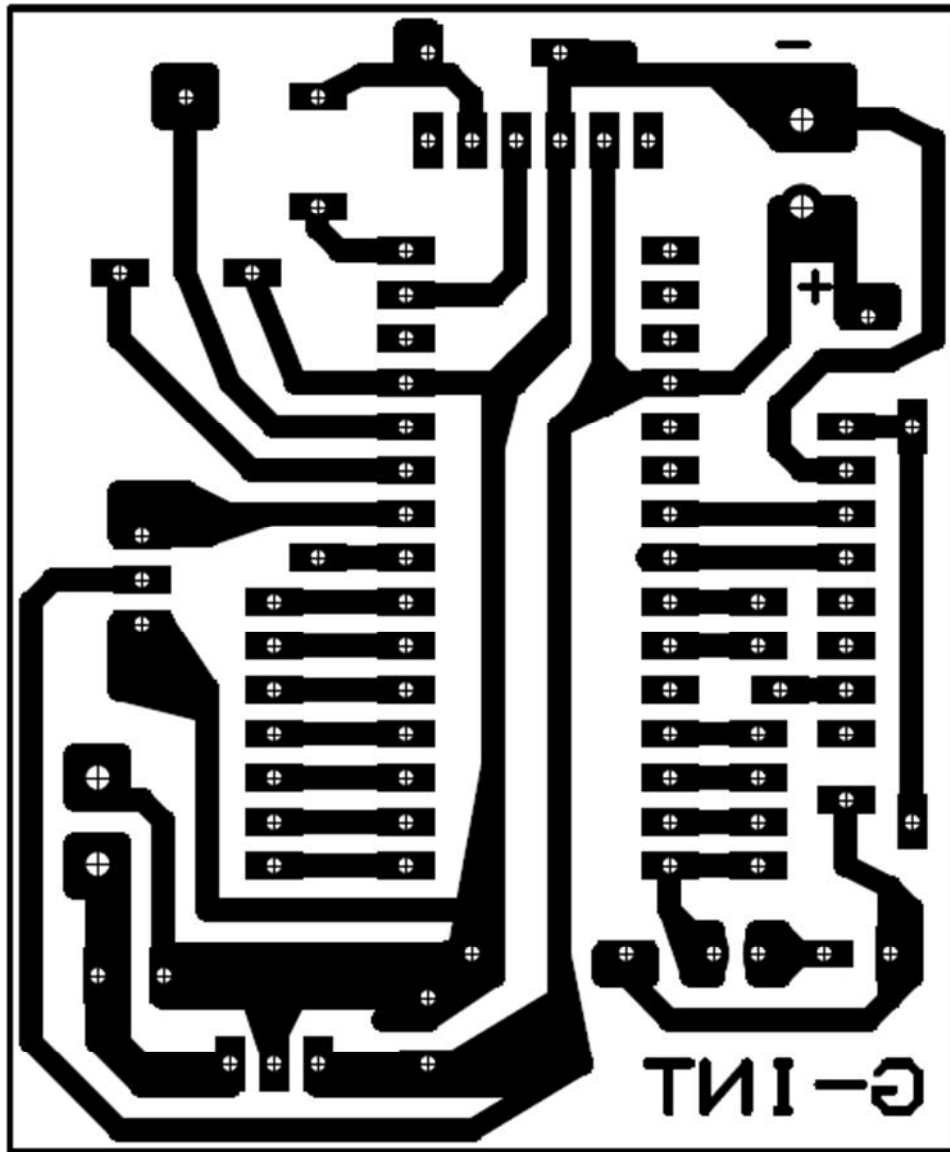
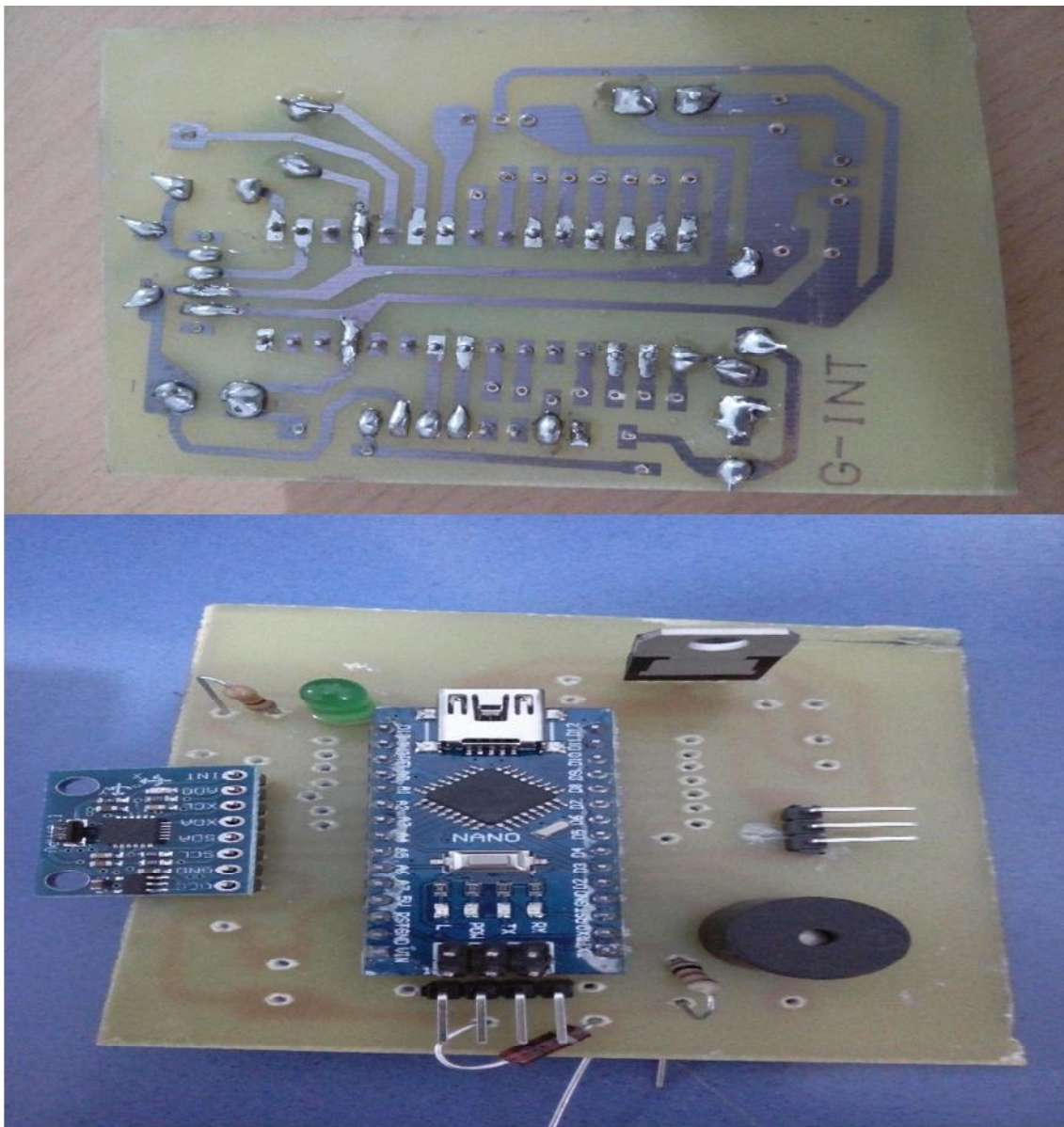


Figure IV.5 : Typon imprimé.

Pour la réalisation de notre projet, nous avons suivi les procédures si dessous:

- ❖ Nous avons découpé la plaque Epoxy /cuivre aux dimensions du futur circuit imprimé 7,8*6 cm
- ❖ Nous avons enlevé la feuille en plastique protégeant la face de la plaque en cuivre sensible aux rayons ultra-violet de la lumière ambiante pour plaquer le typon sur le cuivre et mettre tout dans l'insoleuse en vérifiant bien son sens.
- ❖ Ensuite nous avons trempé la plaque pré-sensibilisée dans le bain révélateur en suivant bien les consignes de sécurité. Si le révélateur est récent ou très actif, l'opération risque d'être très courte.

- ❖ Après nous avons bien rincé avec de l'eau la plaque pré-sensibilisée et nous l'avons placé dans la graveuse en suivant les consignes de sécurité. Par la suite nous avons réglé la minuterie sur 20 min. Le perchlore de fer décompose le cuivre qui n'est pas protégé par la couche photosensible, le cuivre sera donc éliminé sur toute la plaque sauf sur les pistes tracées sur le typon.
- ❖ Enfin, nous terminons notre réalisation par une opération de soudure et cela après le perçage des trous où on implante les différents composants sur leurs emplacements, On obtient notre carte de commande finale.



FigureIV.6 : Carte électronique finale personnalisée.

2-Test du fonctionnement de la carte Arduino :

Initialement, on teste le fonctionnement de la carte Arduino, en connectant cette dernière avec le port USB du PC. Si la LED power s'allume donc la carte est bonne.

Ensuite, on clique sur le bouton Reset de la carte, pour supprimer tout ancien programme afin de la réinitialiser.

3-Branchement des différents composants avec la carte Arduino Nano :

➤ Branchement de la carte Arduino Nano au module DHT 11 :

Les étapes de câblage du DHT11 à la carte Arduino Nano sont:

1. Nous connectons le VCC (broche 1) du DHT11 à la broche 5V de l'Arduino.
2. La broche 2 va être connectée à la broche D4 de l'Arduino.
3. Une résistance de 10K est branchée entre la broche 1 et la broche 2.
4. La broche 4 est connectée à l'une des broches GND de l'Arduino.

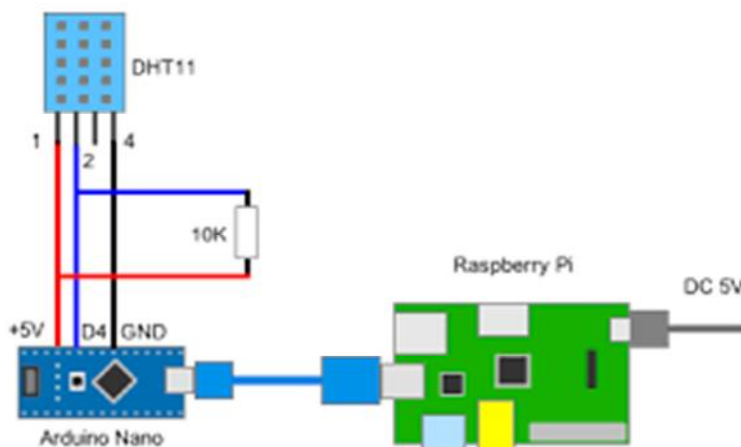


Figure IV.7 : La communication du capteur DHT11 avec Arduino Nano.

➤ **Branchement de la carte Arduino Nano au module GY 521 :**

Les étapes de câblage du GY 521 à la carte Arduino Nano sont :

1. Nous connectons le VCC du module à la broche 5V de l'Arduino.
2. Le module GND est connecté à l'une des broches GND de l'Arduino.
3. La broche SCL va être connectée à la broche A5 de l'Arduino.
4. La broche SDA va être connectée à la broche A4 de l'Arduino.
5. La broche INT va être connectée à la broche D2 de l'Arduino.

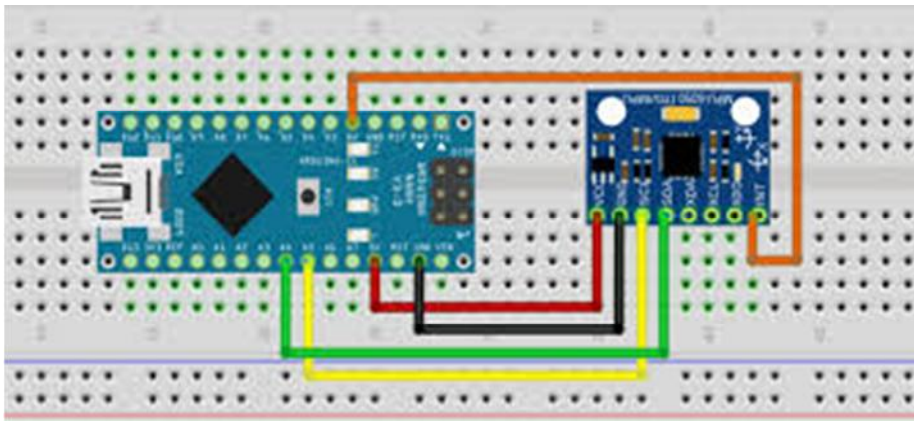


Figure IV.8 : Câblage du GY 521 a une carte Arduino Nano.

➤ **Branchement de la carte Arduino Nano au module HC-05 :**

Les étapes de câblage du HC-05 à la carte Arduino Nano sont :

1. Nous connectons le VCC du HC-05 à la broche 5V de l'Arduino.
2. Le module GND est connecté à l'une des broches GND de l'Arduino
3. La broche TXD va être connectée à la broche D2 de l'Arduino.
4. La broche RXD va être connectée à la broche D3 de l'Arduino.
5. Les broches Key et State restent non connectées.

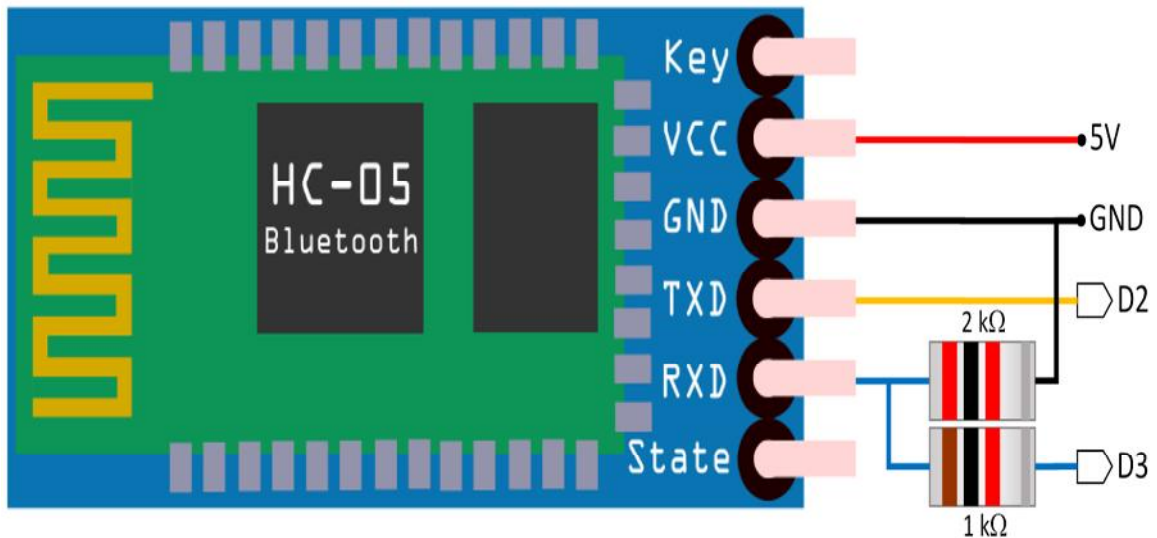


Figure IV.9 : Câblage du HC-05 a une carte Arduino Nano.

➤ **Branchement de la carte Arduino Nano au buzzer:**

Les étapes de câblage du buzzer à la carte Arduino Nano sont :

1. La broche plus (+) va être connectée à la broche D10 de l'Arduino.
2. La broche moins (-) va être connectée à une résistance de 1K qui a son tour va être connectée à l'une des broches GND de l'Arduino.

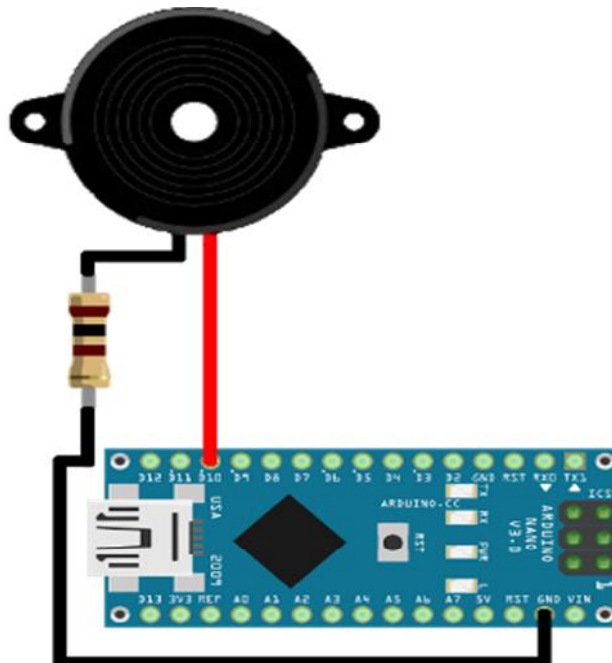


Figure IV.10 : Câblage du buzzer a une carte Arduino Nano

➤ **Branchement de la carte Arduino Nano au module régulateur de tension 7805 :**

Les étapes de câblage du régulateur 7805 à la carte Arduino Nano sont :

1. Le plus (+) de la batterie est connecté avec l'entrée du régulateur 7805.
2. Le moins (-) de la batterie est connecté à la masse du régulateur qui est à son tour connectée à l'une des broches GND de l'Arduino.
3. La sortie du régulateur 7805 va être connectée à la broche +5V de l'Arduino.

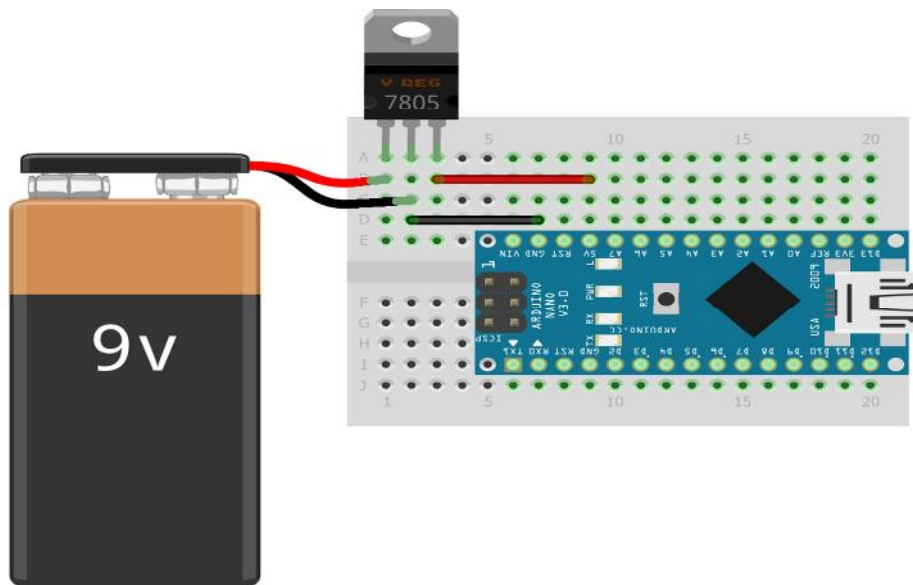


Figure IV.11 : Câblage de la batterie avec la carte Arduino Nano

4-L'organigramme principal :

La programmation va se dérouler en quatre étapes :

- ✓ Programme d'initialisation
- ✓ Programme scan
- ✓ Programme de traitement
- ✓ programme d'action

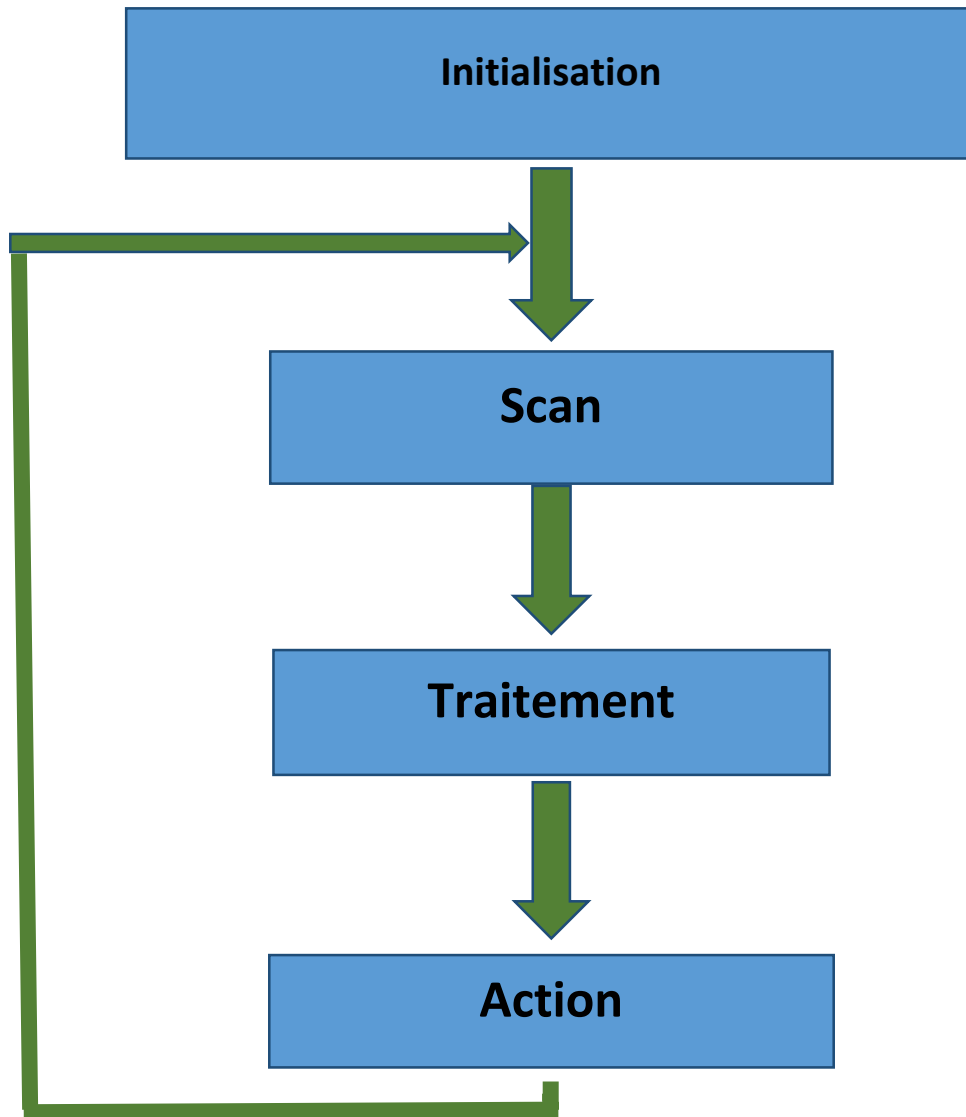


Figure IV.12 : Organigramme de fonctionnement de l'application.

5-Le sous organigramme de fonctionnement :

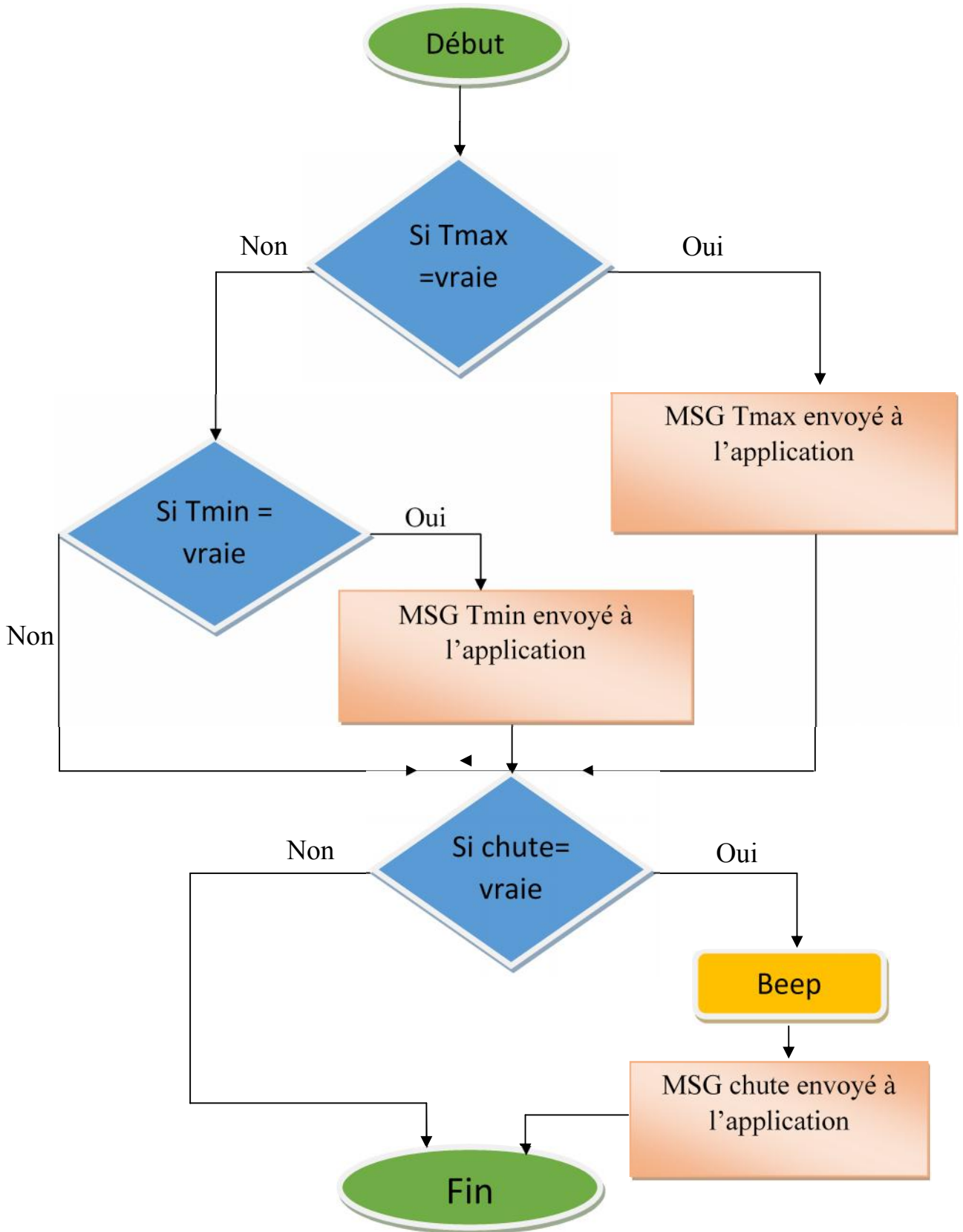
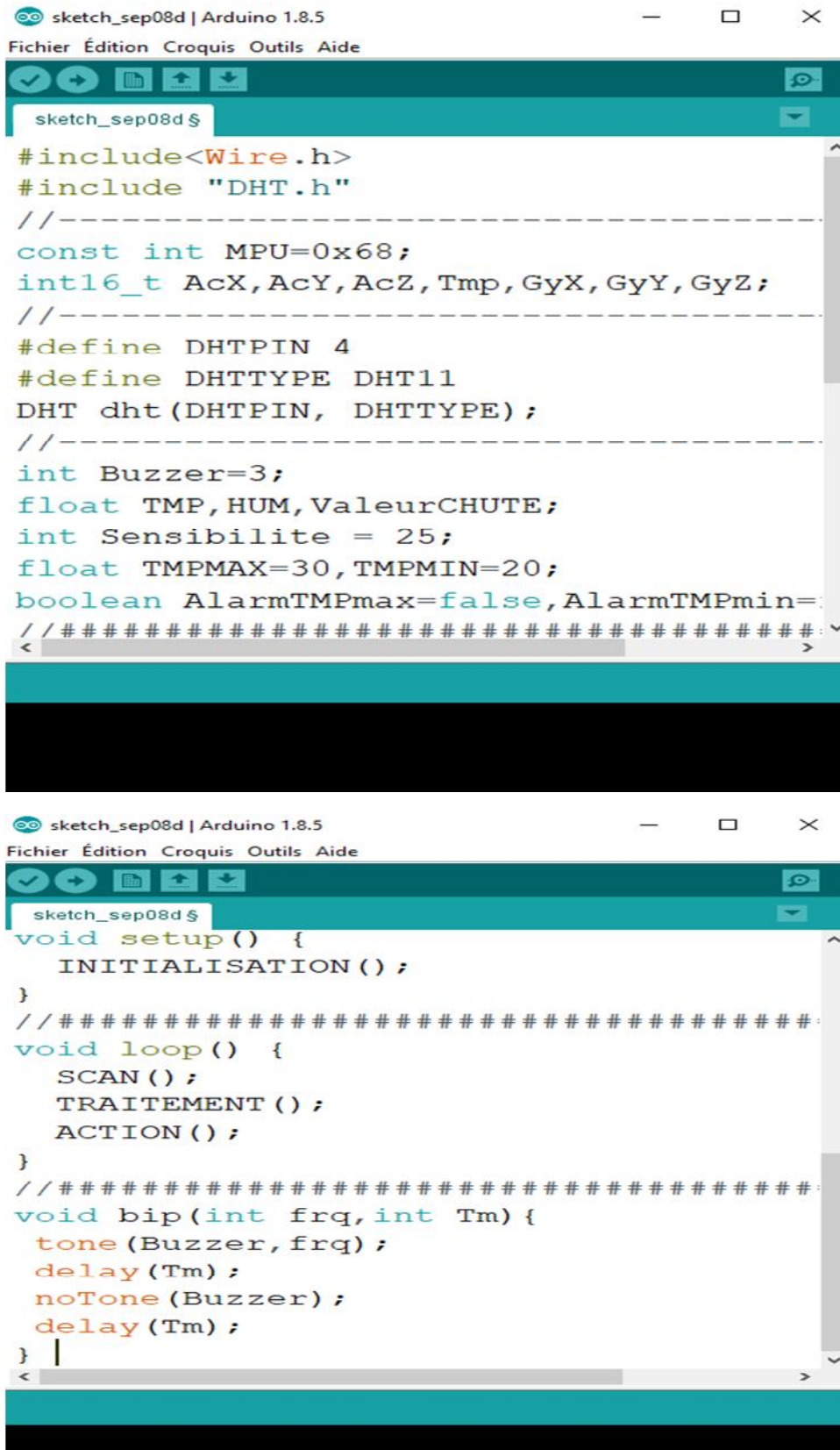


Figure IV.13 : Le sous organigramme de fonctionnement de l'application.

B- Partie logicielle :1- Programme de déclaration :


The figure shows two screenshots of the Arduino IDE interface. The top screenshot displays the declaration part of the sketch, including header files, constants, pin definitions, and variable declarations. The bottom screenshot displays the main logic functions: setup, loop, and a custom bip function.

```

sketch_sep08d | Arduino 1.8.5
Fichier Édition Croquis Outils Aide
sketch_sep08d $
#include <Wire.h>
#include "DHT.h"
//-----
const int MPU=0x68;
int16_t AcX,AcY,AcZ, Tmp, GyX, GyY, GyZ;
//-----
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
//-----
int Buzzer=3;
float TMP,HUM,ValeurCHUTE;
int Sensibilite = 25;
float TMPMAX=30,TMPMIN=20;
boolean AlarmTMPmax=false,AlarmTMPmin=
//#####
<

```

```

sketch_sep08d | Arduino 1.8.5
Fichier Édition Croquis Outils Aide
sketch_sep08d $
void setup() {
  INITIALISATION();
}
//#####
void loop() {
  SCAN();
  TRAITEMENT();
  ACTION();
}
//#####
void bip(int frq,int Tm) {
  tone(Buzzer, frq);
  delay(Tm);
  noTone(Buzzer);
  delay(Tm);
} |
<

```

Figure IV.14 : Programme de déclaration sur le logiciel Arduino

2- Programme d'action :

The image shows a screenshot of the Arduino IDE interface. At the top, the window title is 'sketch_may25a | Arduino 1.8.5'. Below the title bar, there is a menu bar with 'Fichier', 'Édition', 'Croquis', 'Outils', and 'Aide'. A toolbar with various icons is visible below the menu bar. The main workspace contains the following C++ code:

```
void ACTION() {  
  //#####  
  if (AlarmTMPmax) {  
    Serial.println("TMPSUPOK");bip(800,150);  
    delay(50);  
  }  
  //#####  
  if (AlarmTMPmin) {  
    Serial.println("TMPINFOK");bip(800,150);  
    delay(50);  
  }  
  if (ValeurCHUTE > Sensibilite) {  
    Serial.println("MSGCHUTE");bip(800,150);  
    Serial.println(ValeurCHUTE);  
  }  
}
```

Figure IV.15 : Programme d'action

3- Connexion entre la carte électronique et le Smartphone:

➤ Principe de fonctionnement du circuit :

Notre travail était de réaliser une application sur le Smartphone permettant d'aider les personnes âgées à distance.

Pour que le gilet intelligent fonctionne ; tous les équipements matériels et logiciels doivent fonctionner ensemble. La carte de commande est la liaison entre la carte et le Smartphone, qui est faite via le Bluetooth.

Le programme Arduino gère des ordres de mouvements et de températures qui permettent de contrôler les personnes âgées.

4- Conception de l'application :

Dans ce qui suit on va se focaliser sur l'application qu'on a conçue pour contrôler les personnes âgées avec le Smart phone.

Après avoir suivi les étapes détaillées dans le chapitre III nous allons pouvoir créer l'application, et lui donner le nom de « SMART VEST ».

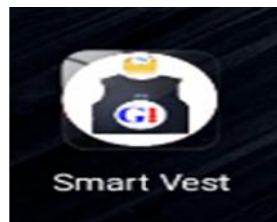


Figure IV.16 : Application Smart Vest

La première phase est la création de son interface. Pour cela, nous allons utiliser un écran de téléphone dans lequel nous pouvons les placer, en les faisant glisser, les éléments que nous voulons utiliser. Ensuite nous allons répartir les différents blocs sur l'écran.

➤ 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 Smart Vest. Il suffit de cliquer sur les boutons pour régler

➤ Commande Bluetooth :

Le module Bluetooth HC-05 assure la communication entre l'application **Android** que nous avons installé sur notre téléphone **Android** et notre carte électronique.

La figure (IV.17) représente les éléments de l'application Smart Vest :



Figure IV.17 : Les différents éléments de l'application.

4.1 La programmation événementielle de l'application Smart Vest :

➤ Programme de Bluetooth :

Premièrement, nous avons commencé par programmer le Bluetooth.

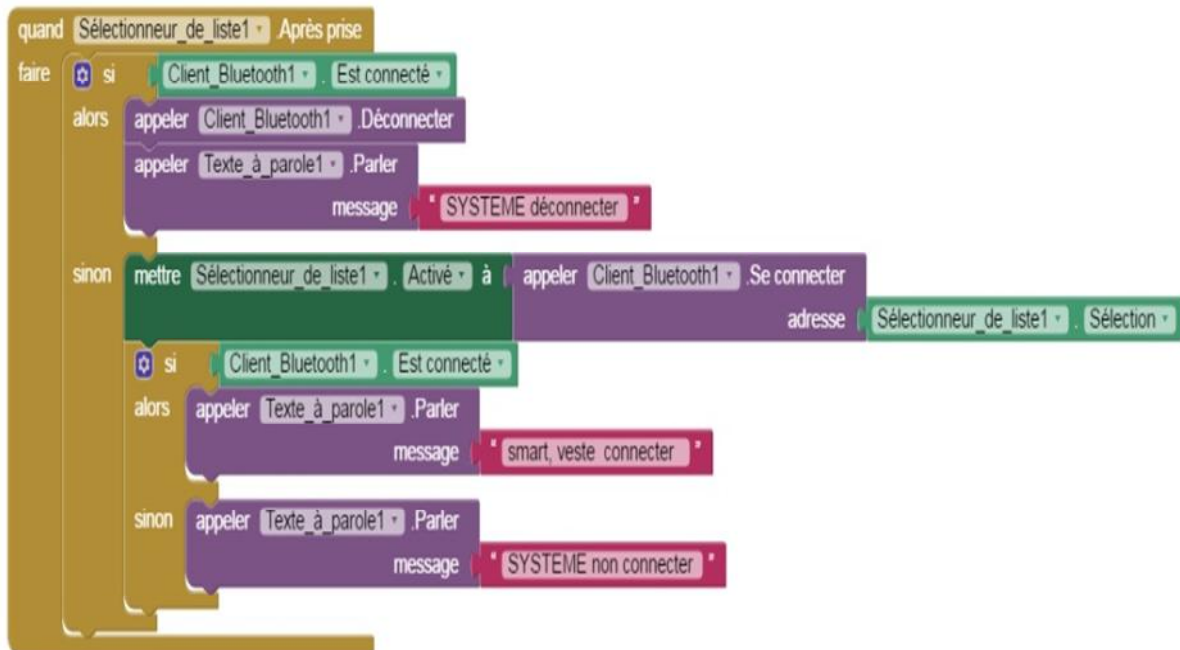


Figure IV.18 : Programme de configuration de Bluetooth

1 : Récupération de la liste des objets connectables au Bluetooth.

2 : Se connecter à l'objet en Bluetooth.

3 : on teste si la connexion Bluetooth est activée et si on est connectée à l'Arduino via le Bluetooth.

➤ Code pour activer /désactiver la communication GSM :

```

initialise global GSMACTIVE à faux
quand Bouton2 .Clic
faire
  si Client_Bluetooth1 .Est connecté
  alors
    si obtenir global GSMACTIVE = faux
    alors
      mettre global GSMACTIVE à vrai
      mettre global TMPINFOK à vrai
      mettre global TMPSUPOK à vrai
      appeler Texte_à_parole1 .Parler
      message "Communication GSM Activé"
      mettre global Nbr à 0
      appeler Client_Bluetooth1 .Envoyer texte
      texte "GSMON"
    sinon
      mettre global GSMACTIVE à faux
      mettre global TMPINFOK à faux
      mettre global TMPSUPOK à faux
      appeler Texte_à_parole1 .Parler
      message "Communication GSM désactivé"
      appeler Client_Bluetooth1 .Envoyer texte
      texte "GSMOFF"
  
```

Figure IV.19 : Code pour activer /désactiver la communication GSM

➤ Code pour l'envoi du SMS :

```

à SMS_SEND NT MSG
faire
  mettre SMS1 . Numéro téléphone à obtenir NT
  mettre SMS1 . Message à obtenir MSG
  appeler SMS1 .Envoyer message

```

Figure IV.20 : Code pour l'envoi du SMS.

- 1- Message Température élevée : **prudence la personne est en danger, la température s'élève.**
- 2- Message température baissé : **prudence la personne est en danger, la température se baisse.**
- 3- Message Chute : **Danger, la personne s'évanouie.**

5- Teste de l'application sur le Smartphone

Nous allons télécharger l'application et l'installer sur Smart phone

Premièrement nous allons configurer le Bluetooth pour qu'il soit reconnu sur le Smartphone, une fois qu'il est connecté on passe à l'application Smart Vest.

Notre projet fonctionne comme le montre la figure (IV.21)



Figure IV.21: L'application fonctionne.

5- Conclusion

Dans ce chapitre, nous avons présenté en détail la réalisation de notre projet. La programmation a été faite avec IDE Arduino, l'application Androïde a été réalisée avec l'environnement MIT APP Inventor et a été testée sur Smartphone.

Conclusion générale

Conclusion générale

Dans notre travail mené tout au long de ce mémoire, nous avons conçu et réalisé un gilet intelligent à base d'une carte Arduino NANO avec une transmission de données via le Bluetooth vers un Smart phone qui avec le GSM va envoyer des informations sous forme de message au médecin (personne qui s'en occupe).

La problématique de ce travail était de trouver des solutions pour mettre en sécurité la santé des personnes âgées en exploitant le développement des technologies.

Dans ce travail nous avons proposé deux solutions, la première est liée à la température du lieu où se trouve la personne âgée. Dans le cas où la température dépasse l'intervalle que nous avons fixé sur l'application, un message d'urgence va être envoyé automatiquement à la personne chargée de la surveillance de cette personne âgée.

La deuxième solution est préconisée en cas de chute de la personne âgée (épilepsie, hypoglycémie...), dans ce cas aussi un message va être envoyé à la personne responsable.

Comme travaux futurs :

- ✓ Dans notre projet on a utilisé le DHT11, il ya un développement qui doit se faire à l'intérieur de l'application Android et le programme de la carte Arduino pour ajouter l'humidité comme option dans le cas où la personne a des problèmes respiratoires par exemple.
- ✓ Nous avons pensé aux personnes amnésiques, dans ce cas l'ajout d'un module GPS est nécessaire.
- ✓ La même carte électronique va être posée dans le berceau d'un bébé, nous enlevons le capteur DHT11, nous le remplaçons par un capteur sous forme d'une petite puce que nous poserons sur le bras du bébé qui détectera sa température, au cas où cette température s'élèvera, elle sera signalée par un message d'urgence à la maman.

Sites et liens consultés :

- ✓ www.arduino.cc
- ✓ www.fr.wikipedia.org
- ✓ www.fritzing.org
- ✓ www.alldatsheet.cm
- ✓ <https://www.dunod.com>
- ✓ <http://sitelyceejdarc.org>
- ✓ <https://www.les-electroniciens.com>
- ✓ <http://www.incertitudes.fr>
- ✓ <http://ai2.appinventor.mit.edu>

Ouvrage :

- ✓ Simon Landrault (Eskimon), Hippolyte Weisslinger (Olyte), Arduino : Premiers pas en informatique embarquée, le blog d'eskimon, Edition du 01 juin 2014.
- ✓ Jean-Noël Montagné, Atelier Arduino, initiation à la mise en œuvre matérielle et logicielle de l'arduino, centre de ressources art sensitif, novembre 2006.

Thèses :

- ✓ Mr BELKESSAM Moussa, étude et évaluation des systèmes de communication sans fil MIMO-OFDM, Master Electronique, 2012/2013, UMMTO.
- ✓ Mr GRIBISSA Aghilas, Commande domotique d'un climatiseur, Automatique et informatique industrielles, 2015/2016, UMMTO.
- ✓ Mr Omar ICHEROUFENE, Conception et réalisation d'un Robot mobile télécommandé à base de la PCDUINO V3, Télécommunication et Réseaux, 2015/2016, UMMTO.
- ✓ Mr Oualid MESSAOUI, Karima MEZAGUER, Conception et réalisation d'un système Intelligent et Autonome de distribution d'eau potable, 2015/2016, UMMTO.