

الجمهورية الجزائرية الديمقراطية الشعبية



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Génie industriel

Spécialité : Ingénierie de la production

Par : **BENALI MOUNIR & CHEBAB MOHAMMED NADJIB**

Thème

**CONCEPTION ET REALISATION D'UNE LUNETTE INTELLIGENTE
POUR LA SECURITE INDUSTRIELLE**

Soutenu le Jeudi 4 Juillet 2019 devant le Jury :

Président : HADRI Abdelkader

MAA – Université de Tlemcen

Examineur : HASSAM Ahmed

MCB – Université de Tlemcen

Examineur : BESSENOUCI Hakim Nadhir

MAA – Université de Tlemcen

Encadrant : GUEZZEN Amine

MCB – Université de Tlemcen

Année Universitaire : 2018 – 2019

Résumé :

Les lunettes intelligentes sont vues comme étant des dispositifs électro-optiques qui servent à exécuter une application ou à projeter une information donnée sur les yeux d'utilisateur sans perdre la vision extérieure.

Ce mémoire présente l'exploitation des lunettes intelligentes au domaine de l'industrie pour servir le confort et la sécurité. Dans ce cadre, nous avons développé certaines fonctionnalités en rendant ces dispositifs compatibles avec de multiples accessoires tels que : outils de mesure (Voltmètre ...), station de surveillance, lecteurs des consignes de sécurité et les applications Android.

Les résultats obtenus démontrent la fiabilité des lunettes intelligentes dans le secteur industriel : le confort d'entretien, la sécurité d'utilisation, la rentabilité et la productivité.

Mots clés : Lunettes intelligentes, Sécurité, Voltmètre, Android, Arduino.

Abstract:

Smart glasses are electro-optical devices that can execute application or project a given information on the user's eyes without losing the external vision.

This study exposes the use of smart glasses in the field of industry to serve comfort and safety. In this context, we have developed some features by making these devices compatible with multiple accessories such as: measurement tools (Voltmeter ...), surveillance station, security instructions reader, Android applications.

The results obtained demonstrate the reliability of smart glasses in the industrial sector: the comfort of maintenance, the safety of use, the profitability and the productivity.

Keywords: Smart Glasses, Security, Voltmeter, Android, Arduino.

ملخص :

النظارات الذكية هي أجهزة كهروضوئية تُستخدم لتنفيذ تطبيق ما أو لعرض معلومات معينة على عين المستخدم دون فقد الرؤية الخارجية.

تقدم هذه المذكرة استخدام النظارات الذكية في مجال الصناعة لتوفير الراحة والأمان. في هذا السياق , قمنا بتطوير بعض الميزات من خلال جعل هذه الأجهزة متوافقة مع العديد من الملحقات مثل: أدوات القياس (مقياس التوتر...), وحدات المراقبة, كاشفات التعليمات الأمنية وتطبيقات الأندرويد.

توضح النتائج التي تم الحصول عليها موثوقية استخدام النظارات الذكية في القطاع الصناعي من أجل توفير راحة الصيانة , سلامة الاستخدام , الربحية والإنتاجية.

الكلمات المفتاحية : النظارات الذكية ، الأمان ، مقياس التوتر ، أندرويد ، اردوينو.

Dédicace

Nous dédions ce projet de fin d'étude à :

- Nos parents qui nous ont soutenu et encouragé tout au long de nos années d'études.
- Nos frères qui nous ont partagé toutes les émotions lors de la réalisation de ce travail.
- Nos proches, à nos amis qui, à tous ceux qui nous ont donné de l'amour, de la persévérance.
- Nos professeurs qui ont veillé de nous encadrer durant notre parcours.
- Tous les collègues de la promotion.
- Tous ceux qui nous aiment ... Merci

CHEBAB Mohammed Nadjib & BENALI Mounir

Remerciement

Après avoir rendu ce projet de fin d'étude, nous tenons à remercier Allah, le tout puissant et miséricordieux, qui nous a donné de la force et de la patience pour conclure ce travail.

Nous tenons à adresser nos profondes gratitudee à notre cher professeur et encadrant M. Guezzen Amine pour son soutien, pour son suivi, pour son temps consacré et pour toutes les informations précieuses pendant la période du projet.

Nos remerciements sont aussi dédiés aux membres de jury d'avoir bien voulu examiner notre travail.

Nous remercions également tout le personnel que nous avons contacté pour ses informations contribuées à la réussite de ce rapport.

Nous tenons de remercier tous les étudiants de la spécialité Génie Industriel de leurs soutiens aussi moraux que financiers et de leurs sacrifices.

Table des matières

Résumé :	i
Dédicace	ii
Remerciement	iii
Liste des figure	ix
Liste des tableaux.....	xi
Liste des Abréviations.....	xii

Introduction générale

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

Chapitre I : Généralités

1.1 Introduction :	3
1.2 Risque industriel :	3
1.3 Sécurité industrielle :	3
1.3.1 Types des effets des risques industriels :	4
1.3.2 Conséquences des risques :	4
1.4 Sûreté de fonctionnement :	4
1.4.1 Définition :	4
1.4.2 Historique :	4
1.4.3 Concepts :	5
1.4.4 Méthodes :	5
1.4.5 Logiciels de sûreté de fonctionnement :	6
1.5 Maintenance :	6
1.5.1 Objectifs de la maintenance :	6
1.5.2 Evolution de la maintenance :	7
1.5.3 Importance de la maintenance :	7
1.5.4 Politiques de la maintenance :	8
1.5.5 Opérations de la maintenance :	9
1.5.6 Centralisation ou décentralisation de la maintenance :	10

1.5.7 Télémaintenance :	10
1.6 Industrie 4.0 :	10
1.6.1 Que signifie l'industrie 4.0 ?	10
1.6.2 Avantage de l'industrie 4.0 :	11
1.7 Réseau industriel :	11
1.7.1 Qu'est-ce qu'un réseau industriel ?	11
1.7.2 Avantages d'un réseau industriel :	11
1.7.3 Connectivité :	11
1.8 Lunettes intelligentes :	12
1.8.1 Un peu d'histoire :	12
1.8.2 Technologie optique à guide transparent :	13
1.9 Proteus :	13
1.9.1 Définition :	13
1.9.2 Présentation de la suite Proteus :	13
1.9.3 Points forts :	13
1.9.4 ARES :	13
1.9.5 ISIS :	14
1.10 Modélisation 3D des objets :	14
1.11 Arduino :	15
1.11.1 Petite définition :	15
1.11.2 Pourquoi Arduino ?	15
1.11.3 Logiciel de programmation :	16
1.12 Android :	16
1.12.1 Définition :	16
1.12.2 Android est-il puissant ?	17
1.12.3 Avantages d'Android :	17
1.12.4 Inconvénients de l'Android Studio :	17
1.13 Etat de l'art :	18
1.13.1 Etat de l'art critique :	19
1 ^{ère} fonctionnalité :	19
2 ^{ème} fonctionnalité :	19
3 ^{ème} fonctionnalité :	20

4 ^{ème} fonctionnalité :.....	20
Pourquoi devons-nous renforcer la sécurité ?.....	21
1.14 Conclusion :	21

Chapitre II : Composants électroniques et outils de développement

2.1 Introduction :.....	22
2.2 Description du problème :	22
2.3 Partie I : Arduino.....	23
2.3.1 Accessoires Arduino :.....	23
2.3.1.1 Arduino UNO :.....	23
2.3.1.2 Arduino Nano :.....	24
2.3.1.3 LCD 2*16 :.....	26
2.3.1.4 Ecran OLED 128*64 :.....	27
2.3.1.5 RFID :	28
2.3.1.6 MQ :.....	30
2.3.1.7 LED :.....	31
2.3.1.8 Résistance :.....	32
2.3.1.9 Module Bluetooth :.....	32
2.3.1.10 Capteur de température et d'humidité :.....	32
2.4 Partie II : Android	33
2.4.1 Historique :.....	33
2.4.2 Etat de la plateforme Android :	34
2.4.3 Architecture de plateforme Android :.....	34
2.4.4 Outils de développement sous Android :.....	34
2.4.4.1 Environnement :	34
2.4.4.2 Langage java :.....	35
2.4.4.3 SDK (software développement kit):	35
2.4.4.4 JDK (java développement kit) :.....	35
2.4.5 Architecture d'un projet Android :.....	36
2.4.6 Création d'une application Android :.....	37
2.4.7 Conception de l'application :.....	39
2.4.8 Développement de l'application :	41

2.5 Conclusion :	46
------------------	----

Chapitre III : Conception et réalisation

3.1 Introduction :	47
3.2 SolidWorks :	47
3.2.1 Définition :	47
3.2.2 Pourquoi SolidWorks ?	47
3.2.3 Mises en plan :	48
3.3 Lunette intelligente	49
3.3.1 Description du produit :	49
3.3.2 Etape 1 : Conception	50
3.3.3 Etape 2 : Mises en plan	51
3.3.4 Etape 3 : Simulation du circuit électrique	52
3.3.5 Etape 4 : Programmation	53
3.3.6 Etape 5 : Montage final	56
3.4 Accessoires développés :	56
3.4.1 Voltmètre :	56
3.4.1.1 Description du prototype :	56
3.4.1.2 Etape 1 : Simulation du circuit électrique	57
3.4.1.3 Etape 2 : Programmation	58
3.4.1.4 Etape 3 : Montage final	58
3.4.2 Station de surveillance :	59
3.4.2.1 Description du prototype :	59
3.4.2.2 Etape 1 : Schéma électrique	59
3.4.2.3 Etape 2 : Programmation	61
3.4.2.3 Etape 3 : Montage final	64
3.4.3 Lecteur des consignes de sécurité (LCS) :	64
3.4.3.1 Description du prototype :	64
3.4.3.2 Etape 1 : Schéma électrique	65
3.4.3.3 Etape 2 : Programmation	66
3.4.3.4 Etape 3 : Montage final	67
3.4.4 Application Android :	68

3.4.4.1 Description de l'application :	68
3.5 Partie résultats et interprétations :	69
3.5.1 Lunette intelligente / voltmètre :	69
3.5.2 Lunette intelligente / station de surveillance :	70
3.5.3 Lunette intelligente / lecteur des consignes de sécurité (LCS) :	71
3.5.4 Lunette intelligente / application Android :	72
3.5.5 Résumé des résultats :	74
3.6 Conclusion :	74

Conclusion générale

Conclusion générale	75
---------------------------	----

Bibliographie

Bibliographie	76
---------------------	----

Liste des figure

Chapitre I

Figure I 1 : Outils de sûreté de fonctionnement	6
Figure I 2 : Evolution de la maintenance	7
Figure I 3 : Types de maintenance	9
Figure I 4 : Révolutions industrielles	10
Figure I 5 : Echelle de la connectivité.....	11
Figure I 6 : Lunette intelligente	12
Figure I 7: Exemple d'un circuit imprimée par ARES	14
Figure I 8: Exemple d'un circuit électrique simulé par ISIS	14
Figure I 9 : Logiciels de modélisation 3D	15
Figure I 10 : Arduino UNO	15
Figure I 11 : Interface de logiciel EDI	16
Figure I 12 : Exemple d'une liaison sans fil entre Arduin et Smart phone	17

Chapitre II

Figure II 1 : Elément de l'Arduino Uno	23
Figure II 2 : Arduino Nano	25
Figure II 3 : Ecran LCD	26
Figure II 4 : Ecran OLED 7 pin	27
Figure II 5 : Exemple d'une connexion RFID	28
Figure II 6 : Tag type porte-clés	29
Figure II 7 : Tag type carte	29
Figure II 8 : Léceteur RFID.....	30
Figure II 9 : Capteur de gaz MQ2.....	31
Figure II 10 : Eléments du capteur MQ2.....	31
Figure II 11 : LED.....	31
Figure II 12 : Résistance.....	32
Figure II 13 : Module Bluetooth HC 05.....	32
Figure II 14 : Module Bluetooth HC 06.....	32
Figure II 15 : Capteur DHT	32
Figure II 16 : Versions d'Android	34
Figure II 17 : Différentes API Android	35
Figure II 18 : Modules d'un projet Android	36
Figure II 19 : Page d'accueil de l'Android Studio	37
Figure II 20 : Choix de l'interface graphique	38
Figure II 21 : Création des paramètres de l'application	38

Figure II 22 : Choix de niveau d'API.....	39
Figure II 23 : Logo de l'application	39
Figure II 24 : Interface d'authentification	40
Figure II 25 : Interface de transmission	40
Figure II 26 : Interface de communication	45

Chapitre III

Figure III 1 : Interface graphique SolidWorks	48
Figure III 2 : Exemple d'un mis en plan	48
Figure III 3 : Lunette intelligente développée	49
Figure III 4 : Fonctionnalités développées	49
Figure III 5 : Diagramme d'exigence.....	50
Figure III 6 : Conception des lunettes intelligentes	50
Figure III 7 : Mise en plan des lunettes intelligentes	51
Figure III 8 : Schéma électrique des lunettes intelligentes	52
Figure III 9 : Simulation des lunettes intelligentes	53
Figure III 10 : Vue interne des lunettes intelligentes	56
Figure III 11 : Voltmètre	56
Figure III 12 : Schéma électrique du voltmètre.....	57
Figure III 13 : Vue interne du voltmètre	58
Figure III 14 : Station de surveillance.....	59
Figure III 15 : Schéma électrique de la station de surveillance	60
Figure III 16 : Vue interne de la station de surveillance	64
Figure III 17 : Lecteur des consignes de sécurité (LCS).....	64
Figure III 18 : Schéma électrique du lecteur des consignes de sécurité (LCS)	65
Figure III 19 : Vue interne du lecteur des consignes de sécurité (LCS)	67
Figure III 20 : Organigramme de l'application Android développée	68
Figure III 21 : Résultat de l'essai N°1 avec le voltmètre	69
Figure III 22 : Résultat de l'essai N°2 avec le voltmètre	69
Figure III 23 : Résultat de l'essai N°3 avec le voltmètre	69
Figure III 24 : Détection du gaz et de température	70
Figure III 25 : Résultat de l'essai N°1 avec la station de surveillance.....	70
Figure III 26 : Résultat de l'essai N°2 avec la station de surveillance.....	70
Figure III 27 : Résultat de l'essai N°3 avec la station de surveillance.....	71
Figure III 28 : Résultat de l'essai N°4 avec la station de surveillance.....	71
Figure III 29 : Résultat de l'essai N°1 avec le lecteur des consignes de sécurité.....	71
Figure III 30 : Résultat de l'essai N°2 avec le lecteur des consignes de sécurité.....	71
Figure III 31 : Résultat de l'essai N°3 avec le lecteur des consignes de sécurité.....	72
Figure III 32 : Résultat de l'essai N°4 avec le lecteur des consignes de sécurité.....	72
Figure III 33 : Interface d'authentification.....	73
Figure III 34 : Interface de transmission	73
Figure III 35 : Résultat de l'essai N°1 avec l'application	73
Figure III 36 : Résultat de l'essai N°2 avec l'application	74

Liste des tableaux

Chapitre I

Tableau I 1 : Types et évènements chronologiques de sûreté de fonctionnement.....	5
Tableau I 2 : Concepts de sûreté de fonctionnement	5

Chapitre II

Tableau II 1 : Fiche technique d'Arduino Uno.....	24
Tableau II 2 : Fiche technique d'Arduino Nano	25
Tableau II 3 : Fiche technique de LCD 2*16	26
Tableau II 4 : Fiche technique d'écran OLED.....	27
Tableau II 5 : Broches de capteur DHT	33
Tableau II 6 : Types de capteur DHT.....	33

Liste des Abréviations

AMDEC : Analyse des Modes de Défaillance, de leur Effet et de leur Criticité

API : Application Programming Interface.

APK : Android Package.

AVD : Android Virtual Device.

BT : Bluetooth.

CAO : Conception Assistée par Ordinateur.

CES : Consumer Electronics Show.

DHT : Détecteur d'Humidité et de Température.

DIMENC : Direction de l'Industrie des Mines et de l'Energie en Nouvelle-Calédonie.

EPC : Electronic Product Code.

EDI : Electronic Data Interchange ou, en français, Echange de Données Electronique.

IIC : Inter Integred Circuit.

JRE : Java Runtime Environment.

JVM : Java Virtual Machine.

LCD : Liquid Crystal Display.

LED : Light Emitting Diode (Diode Electro- Luminescente).

MTP : Multiple Time Programmable.

ODG : Osterhout Design Group.

OLED : Organic Light Emitting Diode.

OTP : One Time Programmable.

PDA : Personal Digital Assistant.

RFID : Radio Frequency Identification.

SDK : Software Development Kit.

SPI : Serial Peripheral Interface.

TDC-FMEA : Telecommunication Company in Denmark-Failure Mode and Effects Analysis

TDC-FTA: Telecommunication Company in Denmark-Fault trees Analysis

WORM : Write Once, Read Multiple.

XML : Extensible Markup Language.

Introduction Générale

Introduction générale

Aujourd'hui, personne ne peut nier le développement technologique que l'industrie a parvenu. Nous parlons, ces jours, des usines connectées, celles de l'industrie 4.0 où les technologies de l'information, de la numérisation, de la réalité augmentée, de la communication mobile ... sont fortement lancées dans le monde entier. Cette transformation a succédé une suite de révolutions telles que les chaînes de montage et l'automatisation ...

L'impact de cet avancement, pour le secteur industriel est assez remarquable. Nous remarquons déjà un passage à la numérisation aux pays développés (Europe, Etat Unis ...) où des entreprises fortement automatisées et aussi bien polyvalentes sont désormais considérées de faibles usines.

Le niveau actuel de l'exploitation des technologies numériques au sein des entreprises de petites et de moyennes tailles (PME) explique, vu à un sondage mondial les raisons pour lesquelles une entreprise devrait hautement investir ces techniques pour améliorer la productivité et la satisfaction de la clientèle.

En parallèle, nous pouvons constater dans notre pays, l'Algérie, un faible investissement de cette gamme technologique provoquant par conséquent des problèmes majeurs qui menacent parfois la vie humaine aux zones industrielles. Parmi ces risques, nous voyons presque chaque jour l'utilisation classique des outils de mesure (Multimètre ...), le contrôle pénible des stations de travail, la non fiabilité de flux d'information et l'ignorance des consignes de sécurité.

Ce présent travail vise à adopter ces nouvelles technologies de l'industrie 4.0 pour la résolution de ces problèmes en indiquant l'importance de ces outils, la connectivité surtout, pour la fiabilité des systèmes, le confort d'entretien et la sécurité humaine. Dans ce cadre, nous allons développer des lunettes intelligentes avec certaines fonctionnalités pour innover la démarche de mesure et de surveillance, l'accès aux fiches techniques et le flux d'information en temps réel.

Nous allons consacrer un premier chapitre pour traiter certaines généralités telles que les risques et les dangers, la sécurité et la sûreté de fonctionnement ... en expliquant dans une partie l'état de l'art de ces dispositifs à l'heure actuelle. Dans le deuxième chapitre, nous allons citer les composants électroniques et les outils de développement qui nous permettront à créer les diverses fonctionnalités de nos lunettes (Arduino, Android ...). Le troisième chapitre est spécifié pour la réalisation du projet où

nous allons montrer, en certaines étapes, la conception, la simulation et la programmation de ces modèles en exposant à la fin du rapport l'interprétation des résultats obtenus.

Chapitre I : Généralités

1.1 Introduction :

Dans ce premier chapitre, nous focalisons sur les notions des risques, sûreté de fonctionnement et de la maintenance. Tout d'abord, nous montrons certaines fonctions qui décrivent ces notions, telles que le risque majeur, le risque industriel, l'arbre de défaillance, la maintenance corrective, la maintenance préventive, les actions de la maintenance...etc. dans la deuxième partie, nous abordons les composants électroniques.

1.2 Risque industriel :

Dans le Petit Robert (*Edition 1996*) on peut lire comme une première définition « *un danger éventuel plus ou moins prévisible* ». Ainsi la définition de Chaline : « *éventualité et probabilité d'un danger* » [10].

Lorsqu'un évènement est très fréquent à apparaître mais avec une gravité importante et sens probabiliste on l'appelle un **risque majeur**, selon Haroun Tazieff : « *c'est la menace sur l'homme et son environnement direct, sur ses installations, la menace dont la gravité est telle que la société se trouve absolument dépassée par l'immensité du désastre* ».

« *Le Risque Majeur est caractérisé par une faible occurrence et une gravité importante engendrant un nombre élevé de victimes et de nombreux dommages matériels et environnementaux* » [6]

En revanche, le terme **catastrophe** ce qui est devenu une certitude peut être définie selon *GEORGE.P* et *VERGER.F* comme un « *Accident violent qui met en cause sur une étendue, la sécurité et l'organisation de l'économie et du peuplement* » [21].

Selon DIMENC [19], le risque industriel se caractérise par un accident se produisant sur un site industriel et pouvant entraîner des conséquences graves pour le personnel, les populations, les biens, l'environnement ou le milieu naturel.

1.3 Sécurité industrielle :

DIMENC [19] a défini la sécurité industrielle comme suit : « *l'application systématique de politique, de procédures et pratiques de gestion visant à analyser, évaluer les conséquences, contrôler (par la mise en œuvre des mesures de prévention, de préparation, d'intervention, de rétablissement et de suivi) et communiquer les risques technologiques majeurs, de façon à protéger les employés, les populations.*

Dans le même contexte, on peut dire que les activités humaines poussées par le développement technologique, ce sont les origines du risque industriel, par exemple :

- Des produits inflammables, explosifs, toxiques...etc.
- Certains produits peuvent devenir dangereux en fonction de leur état, de leur température ou de leur pression.
- Aux installations (choix des matériels, des matériaux, des modes de stockage, ...).

- La majorité des accidents surviennent par négligence, méconnaissance ou erreur d'appréciation.
- Phénomènes extérieurs (séisme, accident d'industrie voisine, malveillance, ...).

1.3.1 Types des effets des risques industriels :

Suivant DIMENC [19], on peut classer les effets des risques industriels en trois catégories :

- **Effets thermiques** : exemple d'une masse électrique dans un magasin du gaz inflammable est par conséquence des : explosions, combustions, brûlures.
- **Effets mécaniques** : causés par une explosion (déflagration ou détonation) des ondes de choc peuvent influencer à la santé des opérateurs au niveau des tympons, poumons...etc.
- **Effets toxiques** : lorsque le processus de préservation des matériaux toxiques ne respecte pas les normes internationales, une fuite des composants toxiques (chlore, acide, etc.) peuvent menacer directement la vie des employés, une atteinte au système nerveux ou des brûlures chimiques cutanées par contact avec la peau ou les yeux.

1.3.2 Conséquences des risques :

DIMENC [19] a cité trois niveaux de conséquences : humaine, économique, environnementale :

- **Conséquence humaines** : toutes les caractères qui sont imposés aux effets du risque de façon directe ou indirecte.
- **Conséquence économique** : défaite de l'argent dans la réparation ou le remplacement des équipements ou le remboursement des salariés endommagés par le dégât.
- **Les conséquences environnementales** : pollution de l'environnement.

1.4 Sûreté de fonctionnement :

1.4.1 Définition :

D'après Alain Villemeur, la sûreté de fonctionnement est définie comme étant une activité qui permet l'évaluation des risques potentiels, l'anticipation des défaillances et la minimisation des conséquences brutales et catastrophiques [1].

Dans les définitions normatives [34], la sûreté de fonctionnement est vue comme la capacité d'un système de remplir une fonction requise, au moment voulu, pendant une durée prévue, sans dommage pour lui-même et son environnement.

1.4.2 Historique :

L'étude de sûreté de fonctionnement n'est pas un phénomène récent, elle a reconnu, depuis diverses périodes, des évènements chronologiques comme le montre le tableau suivant :

Année	Type	Titre	Référence
1600	Approche	Etude de maillon le plus faible d'une chaîne de production : analyse de poste de travail le plus lent de fabrication	[40]
1794	Accident	Explosion de la poudrière de Grenelle	[25]
1842	Accident	Accident de chemin de fer Meudon	[11]
1914	Evènement	Première guerre mondiale (1914 -1918) : les bateaux construits rapidement pour amener les soldats américains sur le sol européen ne résisteront que très peu aux eaux gelées de l'atlantique nord	[40]
2006	Evènement	Création de l'IMDR (Institut pour la Maitrise De Risque)	[22]

Tableau I 1 : Types et évènements chronologiques de sûreté de fonctionnement

1.4.3 Concepts :

Selon les travaux de Algirads et Jean-Claude Lapire [26], la sûreté de fonctionnement s'organise en trois concepts :

- a) **Les attributs** : représentent les caractéristiques des performances de système, ils doivent être quantitatifs et qualitatifs.
- b) **Les entraves** : sont les évènements qui perturbent les performances de système.
- c) **Les moyens** : désignent les techniques et les approches qui améliorent les valeurs des attributs.

Entraves	Moyens	Attributs
Faute Erreur Défaillance	Prévention des risques Tolérance des risques Elimination des risques Prévision des risques	Disponibilité Fiabilité Maintenabilité

Tableau I 2 : Concepts de sûreté de fonctionnement

1.4.4 Méthodes :

Le comportement dysfonctionnel d'un système représente une problématique pour laquelle sont développées de nombreuses méthodes [8], nous présentons, dans ce qui suit, les principales méthodes permettant l'analyse de risques et de sûreté de fonctionnement.

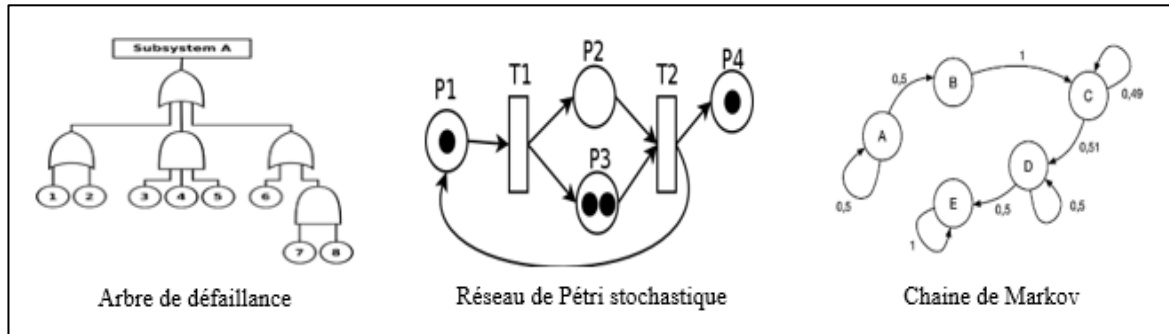


Figure 1.1 : Outils de sûreté de fonctionnement

1.4.5 Logiciels de sûreté de fonctionnement :

Aujourd'hui, personne ne peut nier que pour chacun de grands projets est dédié un logiciel donné. Comme la sûreté de fonctionnement connaît des méthodes d'analyse, elle connaît, de plus, des logiciels qui exécutent des démarches d'analyse de risque telles que : TDC FTA (pour arbre de défaillance), TDC FMEA (pour la démarche AMDEC) ...

1.5 Maintenance :

La Rousse a défini la maintenance comme un ensemble des opérations permettant de maintenir ou de rétablir un matériel, un appareil, un véhicule, etc., dans un état donné, ou de lui restituer des caractéristiques de fonctionnement spécifiées.

La maintenance sert tout simplement à répondre à la question suivante : comment conserver ou remettre un équipement dans état de fonctionnement ?

La maintenance vise toutes les activités destinées à rétablir un bien dans un état de bon fonctionnement, ou dans des conditions de fonctionnement données et spécifiques, pour accomplir une fonction requise; ces activités sont une combinaison d'actions techniques, administratives et de management.

1.5.1 Objectifs de la maintenance :

Les lignes ci-dessous vont bien montrer l'objectif de la maintenance, tel que :

- Améliorer les produits fabriqués (biens ou services) en terme de temps, quantité et de qualité ;
- Minimiser le taux des pannes ;
- Fournir un environnement de travail sécurisé ;
- Augmenter la productivité.

1.5.2 Evolution de la maintenance :

Plusieurs années avant la révolution industrielle, le terme maintenance est limitée sur des actions artisans simple, comme la réparation des : maisons, machines et les wagons de transport (primitives). De plus, la carence des notions des pièces de rechange et des méthodes de mesure de dégradation, la réparation était à l'ordre du jour et les structures de base étaient elles-mêmes réparables, ces activités qui sont caractérisées par une lente évolution dans cette époque, ont commencé à évoluer graduellement au court de la révolution industrielle [41], [35].

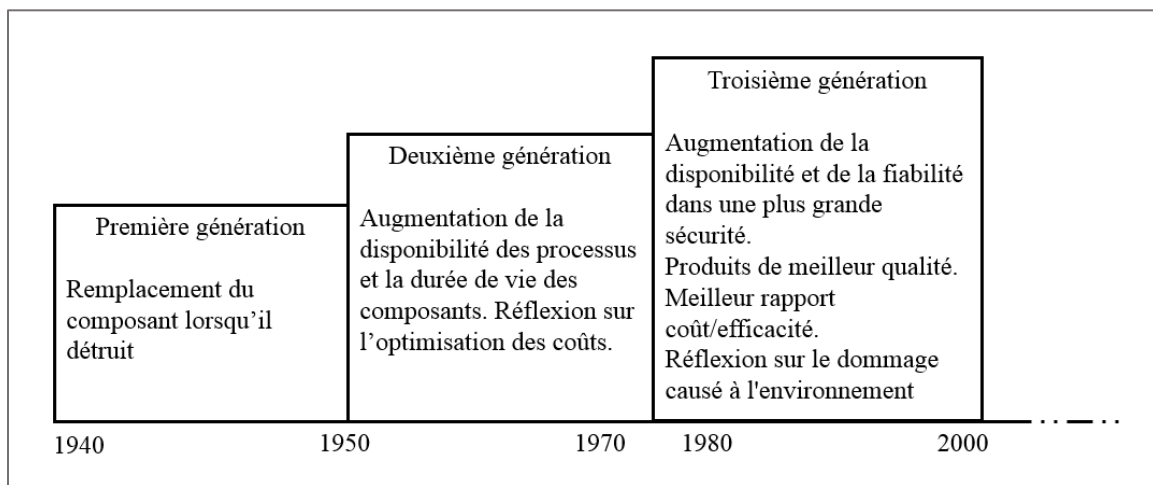


Figure I 2 : Evolution de la maintenance

1.5.3 Importance de la maintenance :

La maintenance est importante pour l'industrie, ce qui paraît clair lors de l'occurrence des pannes provoquant des arrêts non planifiés. Par conséquent, son absence va causer :

- Coût de production très élevé ;
- Perte de profit ;
- Rupture du stock ;
- La non-respect du due date ;
- Heures de travail supplémentaires (panne va retarder la production) ;
- La démotivation des opérateurs (absence de sécurité).

A cette effet, le service de maintenance avec ces actions principales dans le contexte de l'anticipation des problèmes et la planification des entretiens lors de parvenir des pannes, peuvent nous aider à éviter ces conséquences.

1.5.4 Politiques de la maintenance :

La politique de maintenance se répartit en deux piliers, le premier est similaire au principe de là contre force, réparer quand il y a une panne dans le système, on l'appelle la maintenance corrective. En revanche, le deuxième est nommé par la maintenance préventive qui utilise des techniques d'anticipation des problèmes et leur solution alors que le système est en fonctionnement.

a. Maintenance corrective :

La maintenance corrective a pour but de maintenir le système dans un état pour qu'il peut fonctionner à nouveau, il y a deux type de la maintenance corrective [24] :

a.1. Maintenance curative :

La réparation appliquée lors de la confrontation d'une défaillance dans le système elle est complète et définitive [15].

a.2. Maintenance Palliative :

La Maintenance Palliative peut être citée comme un terme provisoire, une solution temporelle, l'objectif est de pousser le système juste pour terminer l'opération en court d'exécution [24], [15].

b. Maintenance préventive :

La mise en place d'une maintenance préventive se fait avant l'apparition de la panne, plusieurs mesures, analyses et précautions sont effectuées afin de minimiser le taux de panne, il y a trois types de la maintenance préventive [33] :

b.1. Maintenance préventive systématique :

On dit maintenance préventive systématique ou périodique car elle suit des écarts du temps prédéfini, sans suivi (au premier lieu) préalable de l'état du système [14].

b.2. Maintenance préventive conditionnelle :

Quand la maintenance préventive se fait sur la base de présence d'un ou plusieurs facteurs qui définisse l'état critique du système on l'appelle conditionnelle. Elle consiste à mise en place des seuils critiques aux signes de dégradation de fonctionnement du système (température, pression...etc.) et contrôler son évolution lors d'une intervention [23].

b.3. Maintenance préventive prévisionnelle :

C'est une maintenance conditionnelle appliquée suivant les résultats obtenus d'après les mesures d'évolution des facteurs significatifs de dégradation du système pour estimer à base du temps la durée qu'un équipement peut fonctionner correctement avant de tomber en panne pour anticiper une intervention [13].

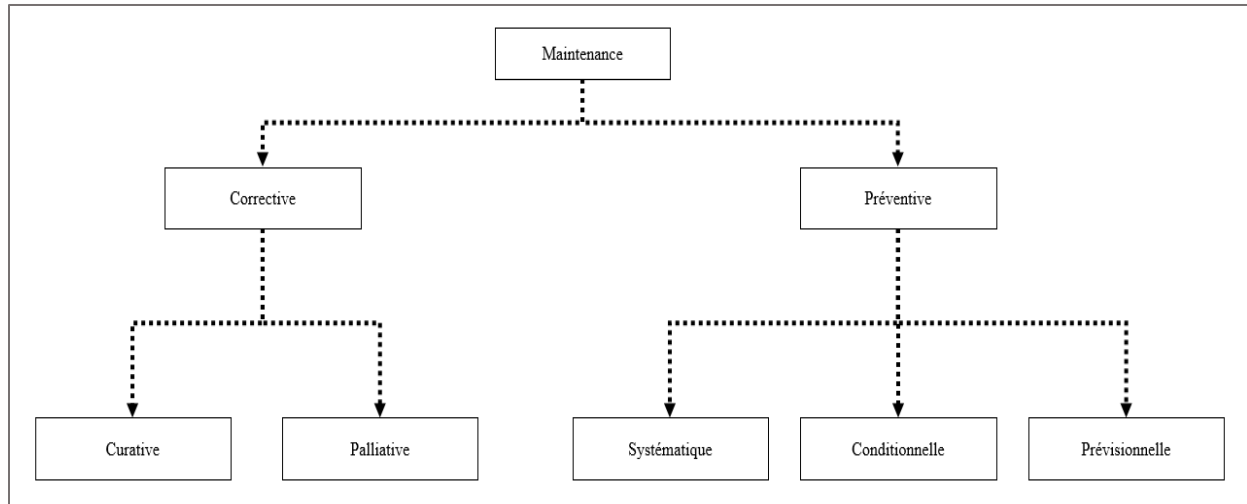


Figure I 3 : Types de maintenance

1.5.5 Opérations de la maintenance :

1. Opérations de la maintenance corrective :

- Test : si les mesures sont égales ou non une référence.
- Détection : la présence d'une défaillance.
- Localisation : identifier précisément les éléments qui manifestent sur le panne
- Diagnostic : analyser et déterminer les causes de la panne.
- Dépannage, réparation : mis du système en état redouté.
- Contrôle : vérifier si le système peut fonctionner après l'intervention.
- Amélioration éventuelle : éviter la réapparition de la panne.
- Historique : archiver les interventions.

2. Opérations de la maintenance préventive :

- Inspection : observation, test et mesure des caractéristiques significatives (pas de démontage, ou arrêt de travail nécessaire)
- Contrôle : vérifier les données lors de l'inspection, on peut prendre une décision corrective, acceptation ou refus.
- Visite : complète ou partielle, détailler et prédéterminer sous forme d'un examen pour les composants du système.
- Test : faire une comparaison des résultats obtenus avec une référence
- Echange standard : placer des pièces nouvelles ou identiques au lieu des pièces défectueuses

- Révision : des examens et des actions pour garantir le niveau de sécurité et de disponibilité des équipements. Elle nécessite un démontage complet ou partiel de l'équipement.

1.5.6 Centralisation ou décentralisation de la maintenance :

D'après Abdou R [2] et Alhouaji Ahmed [3], il y a :

1. La centralisation : il y a un seul service qui garantit le suivi de maintenance.
2. La décentralisation : pas seul mais plusieurs services interagissant sur le suivi de la maintenance.

1.5.7 Télémaintenance :

Des actions de maintenance appliquées sans interaction directe du personnel à l'aide des moyens de télécommunication dans le concept du contrôle à distance.

1.6 Industrie 4.0 :

Depuis l'époque, l'industrie a subi plusieurs révolutions [5] :

- Industrie 1.0 : s'articule autour de la mécanisation et les machines à vapeur.
- Industrie 2.0 : a introduit la production de masse et les chaînes d'assemblage.
- Industrie 3.0 : basée sur l'automatisation, les ordinateurs et l'électronique.
- Industrie 3.5 : la mondialisation de la production.
- Industrie 4.0 : la numérisation, la connectivité et l'internet des objets.

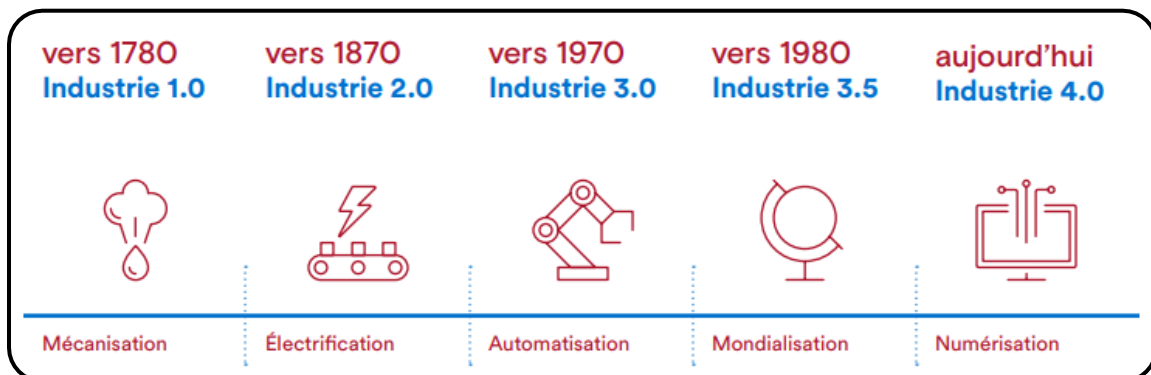


Figure I 4 : Révolutions industrielles

1.6.1 Que signifie l'industrie 4.0 ?

Le terme « industrie 4.0 » signifie le recours aux technologies numériques pour rendre les activités plus souples et mieux adaptées aux besoins de l'industrie [5].

1.6.2 Avantage de l'industrie 4.0 :

Nous pouvons citer les apports de l'industrie 4.0 en quelques points :

- Croissement de la productivité, l'efficacité ...
- Réduction des coûts de production
- Amélioration de la qualité de produit
- Facilitation d'innovation

1.7 Réseau industriel :

Présentons en premier temps ce que désigne le terme réseau :

Un **réseau** désigne un ensemble de moyens permettant la communication et la transmission des signaux entre des tâches ou des processus de type informatique [17].

1.7.1 Qu'est-ce qu'un réseau industriel ?

Un réseau industriel est tout simplement un réseau mis en œuvre dans une zone industrielle pour connecter de différents processus en vue de permettre : la gestion, la commande, la surveillance, la supervision, la maintenance, le suivi de produit ...

1.7.2 Avantages d'un réseau industriel :

- Réduction des coûts des câblages ;
- Amélioration de service de maintenance ;
- Augmentation de performance ;
- Echange de données en temps réel.

1.7.3 Connectivité :

D'après l'internaute, le dictionnaire français, la connectivité signifie une liaison entre des périphériques informatiques. D'une vision générale, la connectivité remplace le terme de « réseau industriel » pour l'industrie 4.0.



Figure I 5 : Echelle de la connectivité

Protocoles de « connectivité » :

L'industrie 4.0 utilise généralement des technologies sans fils pour relier ses appareils et pour transmettre les données. Ces technologies s'appuient fortement sur deux modules majeurs (wifi et Bluetooth) :

Wifi : c'est une transmission sans fil permettant de relier des ordinateurs portables, des machines de bureau ou tout type de périphériques à une liaison haut débit sur un rayon de plusieurs dizaines de mètres en intérieur (de 20 à 50 mètre généralement) à plusieurs centaines de mètres en environnement ouvert.

Bluetooth : la technologie Bluetooth est une connexion sans fil, utilisant des ondes radio et permet à deux appareils situés dans deux pièces différentes de se relier. La portée est limitée à 100 m et diminue suivant les obstacles rencontrés.

L'industrie 4.0 a apporté, au monde industriel, beaucoup d'avantages vus à ses technologies récentes. La numérisation, l'élément de base de cette industrie, a permis la démocratisation de nouveaux produits (objets connectés) exploitant la connectivité comme source de communication.

1.8 Lunettes intelligentes :

D'après Sarayeddine, K [39], une lunette intelligente est vue comme étant un dispositif électro-optique qui sert à projeter une image ou généralement une information donnée sur les yeux de l'utilisateur sans perdre sa vision extérieure. Elle peut de plus exécuter des applications mobiles.

N.B : certaines lunettes sont tellement développées qu'elles peuvent être équivalentes d'une tablette, elles exécutent de vraies applications (la réalité augmentée par exemple).



Figure I 6 : Lunette intelligente

1.8.1 Un peu d'histoire :

Le secteur de l'avionique militaire était le premier à voir l'intérêt des lunettes intelligentes, tout en affichant des informations dans les yeux de ses pilotes en situation de risque ou de danger.

Nokia et Sony ont développé leurs dispositifs en intégrant de nouvelles technologies de telle sorte que leurs lunettes permettent le positionnement optique ainsi que la vision de l'utilisateur.

Google et Microsoft ont apporté à ces dispositifs de nouvelles techniques. L'utilisateur peut encore exécuter des applications ou d'accéder librement à sa connexion internet via ces lunettes.

1.8.2 Technologie optique à guide transparent :

Il y a encore trois méthodes de cette technologie optique (diffractive, polarisante, réflective) :

- La méthode diffractive : elle s'appuie sur un élément optique holographique à l'affichage des données. Cette méthode ne permet pas la visualisation en couleurs.
- La méthode polarisante : cette méthode utilise des réflecteurs polarisants. Son rendement est un peu faible par rapport aux autres méthodes.
- La méthode réflective : elle utilise un seul réflecteur mis entre deux biseaux plastiques. Cette méthode est exploitée par de majeures entreprises telles que Google, Epson ...

La plupart des lunettes intelligentes disposent des circuits électroniques trop sophistiqués. La réalisation de ce genre de dispositif nécessite une simulation virtuelle pour vérifier leurs comportements. Pour cela centaine de logiciels (Proteus, KiCad, MPLAB ...) sont développés afin de rectifier toute sorte d'erreur dès la conception.

1.9 Proteus :

1.9.1 Définition :

Proteus est un ensemble de logiciel dédié à l'électronique, développé par la société labcenter Electronics. Ces logiciels permettent la CAO dans le domaine électrique [43]. Nous parlons de deux logiciels principaux : ISIS et ARES.

1.9.2 Présentation de la suite Proteus :

Il s'agit d'un ensemble de logiciels destiné au domaine de l'électronique. Beaucoup de sociétés, d'organisations exploitent cette suite, les lycées les universités par exemple [43].

1.9.3 Points forts :

Proteus permet les avantages suivants :

- Pack facile à utiliser
- Pack trop performant
- Le prototypage virtuel réduit les coûts matériels lors de la création du projet

1.9.4 ARES :

Il représente un outil d'édition et de routage. Un schéma électrique conçu par ISIS peut être facilement importé sur ARES pour réaliser le circuit imprimé de la carte électronique. ARES permet de placer automatiquement les composants électriques et d'installer leur routage [43].

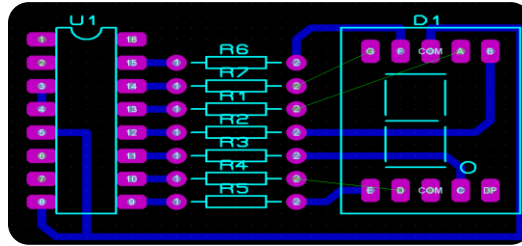


Figure I 7: Exemple d'un circuit imprimé par ARES

1.9.5 ISIS :

C'est un logiciel inclus dans la suite Proteus, destiné principalement à la création des schémas électriques. Il sert, de plus, de stimuler ces schémas en vue de détecter les erreurs avant la réalisation du projet. Ces schémas électriques réalisés par ce logiciel peuvent être intégrés dans des documentations [43].

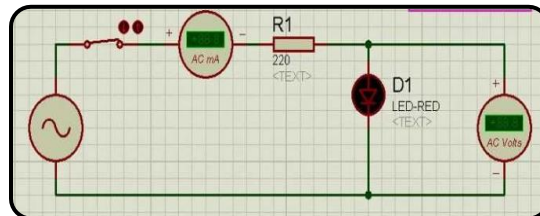


Figure I 8: Exemple d'un circuit électrique simulé par ISIS

1.10 Modélisation 3D des objets :

Comme le pack Proteus sert à concevoir et à simuler les composants électroniques, il y a de plus de multiples logiciels qui permettent la modélisation 3D de ces composants. De nombreuses entreprises utilisent SolidWorks et Catia comme logiciel de modélisation.

Présentation générale de SolidWorks et Catia : Ils sont tous les deux développés par l'entreprise Dassault Système, dédiés à la modélisation 3D des objets, utilisés dans plusieurs domaines tels que l'ingénierie ...

Avantages :

La modélisation 3D est particulièrement prometteuse dans tous les secteurs, l'industrie surtout, grâce aux avantages qui suivent :

- Elle sert à améliorer la conception et sa qualité ;
- Elle augmente la productivité du concepteur ;
- Elle permet la conception complète ;

- Elle réduit le temps de conception ;
- Elle donne une meilleure visualisation aux clients.

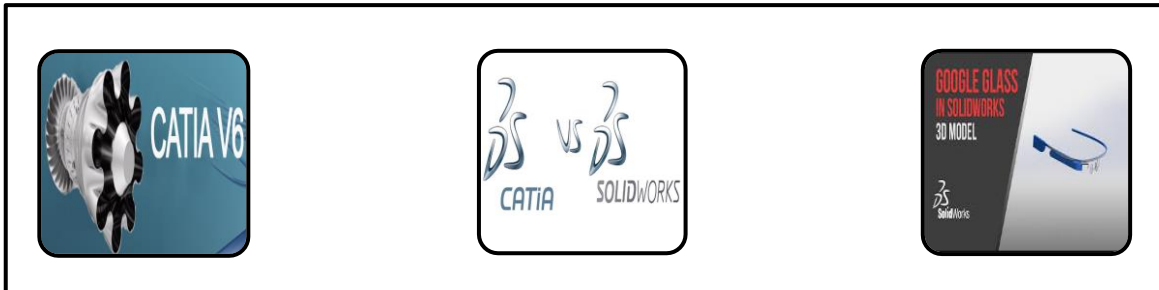


Figure I 9 : Logiciels de modélisation 3D

1.11 Arduino :

1.11.1 Petite définition :

D'après son site officiel, Arduino est une open source électronique. Une plateforme facile à utiliser en terme de logiciel ou de matériel. La carte Arduino peut lire diverses entrées (capteurs, bouton ...), exécuter des sorties (activer un moteur, allumer un LED ...) [44].



Figure I 10 : Arduino UNO

1.11.2 Pourquoi Arduino ?

La carte Arduino a été utilisé en centaine projets et applications vu à ses apports :

- EDI est facile à appréhender pour les débutants, trop flexible pour les développeurs ;
- Il s'exécute sur divers systèmes d'exploitation (Windows, Linux, Mac ...) ;
- Utile pour les étudiants pour créer leurs prototypes en robotique ... ;
- Pas cher.

1.11.3 Logiciel de programmation :

La programmation de la carte Arduino s'effectue sur un logiciel gratuit (EDI), une interface simple à utiliser, plus lisible. Le logiciel de programmation (EDI) exécute des commandes, environ centaines de commandes différentes. L'EDI peut effectuer les fonctionnalités suivantes :

- Ecriture et compilation des programmes ;
- Télé version des programmes compilés dans la carte Arduino ;
- Communication avec la carte Arduino via les portes série.

Voici une figure illustrant le logiciel EDI :

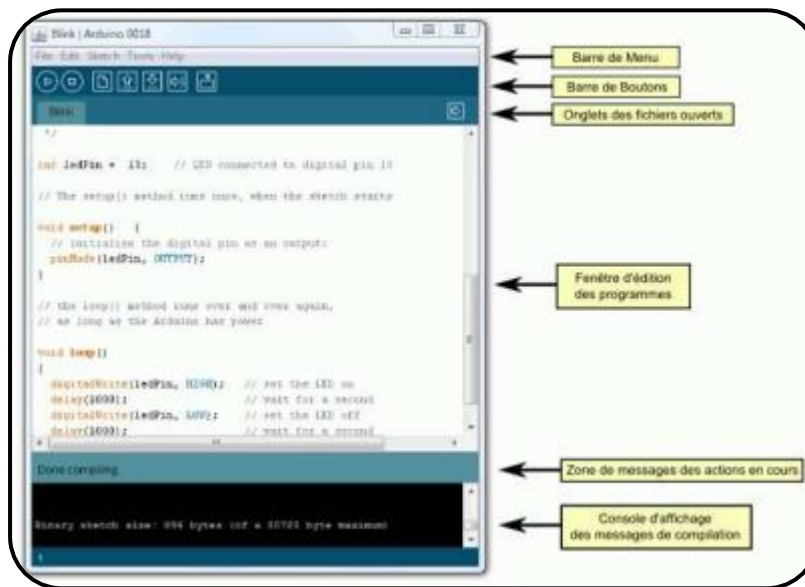


Figure 1 11 : Interface de logiciel EDI

La carte Arduino est compatible non seulement avec le kit précédent mais aussi avec le système d'exploitation Android destiné aux smart phones. Une application développée par Android peut être simplement utilisée comme entrée pour la carte Arduino. Nous pouvons par exemple contrôler un prototype à distance par une application via une connexion Wireless (Wifi, Bluetooth ...).

1.12 Android :

1.12.1 Définition :

Android est un système d'exploitation développé par l'Open Handset Alliance, destiné pour les smart phones, lancé en 2007, basé sur Linux, utilisant Java comme plateforme d'applications [20].

1.12.2 Android est-il puissant ?

- 1) Vu à la simplicité et la flexibilité qui caractérise Android, le système d'exploitation moderne est considéré parmi les systèmes puissants à l'image d'iOS, Windows phone etc. Ce qui montre le développement du système.
- 2) Android est un système open source. Il est possible aux développeurs de créer leurs applications.
- 3) Linux lui génère une mémoire importante, une bonne gestion de processeur ...

1.12.3 Avantages d'Android :

- OS Kernel Robust ;
- Soutient des bibliothèques innovantes ;
- Trop facile au développement d'applications ;
- Il s'exécute rapidement.

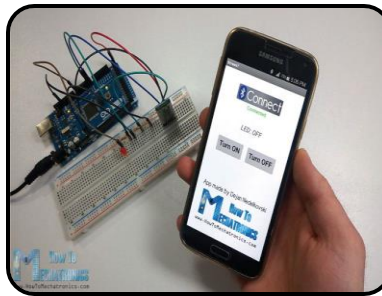


Figure 1 12 : Exemple d'une liaison sans fil entre Arduino et Smart phone

N.B : Dans la figure précédente, nous avons une carte Arduino contrôlé par une application Android. L'application permet via une connexion Bluetooth d'allumer ou d'éteindre le LED rouge.

1.12.4 Inconvénients de l'Android Studio :

Android possède, en parallèle de ces avantages, des inconvénients majeurs, comme par exemple :

- Android nécessite des périphériques trop puissants pour s'exécuter en pleine performance ;
- Il exige une dépendance, de Google, trop difficile à accéder.

1.13 Etat de l'art :

La technologie des lunettes intelligentes est un phénomène trop récent pour le monde industriel. Dans les passages suivants, nous allons présenter certains modèles adoptés, ou qui peuvent être adoptés aux services de l'industrie.

Vuzix Blade :

Vuzix Blade sont des lunettes intelligentes de réalité augmentée, produites par l'entreprise Vuzix en 2019. Elles fonctionnent à base d'un système d'exploitation Android. Elles peuvent donc se connecter aux smart phones via connexion Bluetooth ou Wifi. Elles disposent d'une caméra HD, comme elles peuvent être utilisées pour la protection ultra-violet [53].

Vue Smart Glasses :

Développées par l'entreprise Kickstarter. Un utilisateur pourrait lancer ses ordres, demander un appel téléphonique par exemple, à travers une communication vocale [54].

Epson Moverio BT-300 :

Les lunettes intelligentes Epson s'agissent plutôt des dispositifs de réalité augmentée, conçues en 2016 pour compléter les drones comme fonction majeure. Les caméras intégrées leur permettent de visionner le vol des drones et d'afficher les statistiques concernées. Elles sont utilisées pour la photographie, l'architecture, la sécurité ... [53].

Rokid Glass :

Produites par l'entreprise Rokid en 2019. Ces lunettes fonctionnent sous Android. Grâce à la technologie SLAM (Simultaneous Localisation And Mapping), elles permettent la reconnaissance des objets (visages, formes ...). Ces dispositifs sont contrôlés par les commandes vocales [53].

La réalité augmentée pour détecter les défauts d'un coup d'œil :

Développées dans l'usine de Venissieux, les lunettes intelligentes Active glass permettent de vérifier, grâce à un système de réalité augmentée, que les installations sont conformes aux paramètres établis. Concrètement, lors d'un contrôle de qualité visuel, l'opérateur voit apparaître sur ses verres-écrans une photo de la machine qu'il a devant lui. Au lieu de procéder aux multiples vérifications linéairement, à partir de checklists en papier, il peut directement comparer le modèle de "ce qui devrait être" à la réalité. "Cela permet de gagner du temps. Plus besoin de lire les instructions. L'opérateur embrasse globalement l'objet du regard et détecte immédiatement les défauts [55].

Certains développeurs ont investi ces modèles, citons par exemple :

- **Abdelhafid RACHIDI** en 2015, avec son groupe de travail, ont exploité la technologie de réalité augmentée, à l'aide d'une lunette intelligente, aux services de télémaintenance et de e-maintenance [36].

- **Konica Minolta** en 2018 a intégré des lunettes connectées au service de logistique et de maintenance pour bénéficier de meilleurs flux de travail [51].
- **Philippe Peyrard** en 2018, a inventé des lunettes connectées permettent la prévention de l'endormissement au volant [52].

1.13.1 Etat de l'art critique :

Nos lunettes intelligentes à développer s'agissent plutôt de dispositifs de sécurité au domaine industriel. Elles peuvent apporter aux utilisateurs beaucoup de fonctionnalités que nous allons expliquer dans les lignes suivantes :

1^{ère} fonctionnalité :

Un électricien a pour fonction de mesurer, de temps en temps, les grandeurs électriques des équipements électroniques, les armoires électriques surtout. Un électricien, quel que soit son grade, a fortement besoin d'un multimètre pour mesurer ces grandeurs, en cas de l'armoire par exemple, il doit à chaque moment tourner sa tête pour bien lire la valeur mesurée à l'écran du multimètre, ce qui amène aux erreurs : l'électricien ne fait probablement pas attention aux câbles nus sous tension et par conséquent un état de mort.

Avantages des lunettes intelligentes :

L'existence des lunettes intelligentes dans cette situation assure à l'électricien deux trucs important :

- **Le confort** : certaines douleurs au cou sont aggravées vu à l'utilisation classique du multimètre. L'électricien peut confortablement lire la valeur mesurée depuis les verres de ses lunettes intelligentes.
- **La sécurité** : le fait que l'électricien n'a plus encore besoin de recourir à son multimètre, cela lui permet de bien se concentrer avec son équipement à vérifier en évitant toute source d'accident.

Principe de fonctionnement :

Cela peut être résumé en quelques étapes :

- On doit prendre les mesures via un multimètre ;
- Le multimètre transmet les données via Bluetooth ;
- Les lunettes reçoivent ces données ;
- Les données reçues doivent être traduites et réfléchies sur les yeux d'utilisateur.

2^{ème} fonctionnalité :

La deuxième fonctionnalité touche la contrôle intelligent des stations de travail au sein de l'entreprise. Suivant des capteurs bien définis, un message d'alerte est envoyé aux lunettes intelligentes en cas où ces capteurs détectent des valeurs indésirables. Il n'exige plus encore de mettre en place un employé pour surveiller les stations concernées.

Avantages :

Nous parlons plus précisément de :

- Faciliter l'opération de surveillance ;
- Optimiser le nombre d'employés ;
- Minimiser le salaire généré ;
- Gagner en temps pour d'autres fonctions.

Principe de fonctionnement :

- On doit choisir les stations concernées pour le contrôle ;
- On détermine de plus la marge de bon fonctionnement ;
- En cas de dépassement de la marge, les capteurs indiquent des signes d'alerte ;
- Les lunettes détectent ces signes d'alertes pour les réfléchir aux yeux d'employé.

3^{ème} fonctionnalité :

Grâce aux accessoires intelligents, l'employé a encore la possibilité d'identifier les informations essentielles d'une machine, d'une chaîne de production ou d'une station de travail. Il peut même être averti de certaines utilisations en cas de pénétration des zones critiques. Toutes ces informations sont projetées aux yeux d'utilisateur.

Apports :

- Encourager l'employé à poursuivre les consignes de sécurité de manière innovante ;
- Améliorer l'identification des stations de travail ;
- Exciter l'employé au monde de travail.

Principe de fonctionnement :

Les lunettes devraient être capable, grâce à un système bien établi, de traduire les consignes chiffrées par les codes des stations.

4^{ème} fonctionnalité :

Cela consiste dans l'intégration d'une application Android comme étant un accessoire pour les lunettes. Elle sert à établir la communication à distance entre ces dispositifs et les smart phones. Les employés peuvent librement se communiquer via leurs lunettes, ce qui ramène beaucoup d'apports.

Pourquoi une application Android ?

- Communication en temps réel ;

- Optimisation des coûts de communication ;
- Un message reçu par un mobile peut être non visualisé vu au bruit des machines ;
- La connexion est toujours disponible.

Principes de fonctionnement :

Il s'agit d'une application qui :

- Est compatible aux lunettes pour établir la connexion entre les dispositifs ;
- Est seulement spécifique pour les employés prédéfinis ;
- Assure l'émission des données à distance à travers une connexion sans fil.

Pourquoi devons-nous renforcer la sécurité ?

Selon Leplat, G [28], deux facteurs trainant la violation des règles de sécurité par les opérateurs : acceptation et accessibilité.

- 1) **Acceptabilité** : un opérateur, que ce soit débutant ou expérimenté, peut accepter ou réfuter une règle de sécurité suivant certains facteurs :
 - **Coût d'utilisation** : nous parlons plutôt du rapport coût / bénéfice, un opérateur cherche à accomplir une tâche donnée tout avec un moindre effort.
 - **Cohérence des règles** : en cas où les règles se produisent de façon progressive à chaque accident où incident, il se peut donc créer des conflits entre les articles eux-mêmes (manque de cohérence) [28].

N.B : d'autres facteurs pourraient être aussi remis en cause : la sur-spécification des règles et la pertinence des règles ...).

- 2) **Accessibilité** : s'articule autour de la disponibilité et la lisibilité des règles au cours d'exécution des tâches, ce qui rend facile ou difficile l'accès à l'information. Les règles doivent permettre, aux opérateurs, la logique de sécurité pour bien gérer les risques, ce qui fait appel à la notion de formation comme l'indiquent [38], [32].

1.14 Conclusion :

Dans ce chapitre nous avons présenté les techniques principales utilisées dans la maintenance et dans la sûreté de fonctionnement qui visent à maintenir le fonctionnement des systèmes d'une part et à minimiser les effets (toxiques, chimiques et mécanique) et les conséquences (environnementales, humaines et économiques) des risques industriels d'autre part.

Dans le chapitre suivant, nous allons beaucoup plus expliquer les composants électroniques (Arduino) ainsi que les outils de développement (Android) nécessaire pour la conception et la réalisation de notre projet.

Chapitre II : Composants électroniques et outils de développement

2.1 Introduction :

Dès la création du premier ordinateur, aux années passées, les clés de la programmation et de l'électronique étaient réservées, jusqu'à l'arrivée de « l'open source » où chacun pourrait modifier, développer ... complètement gratuit.

Dans ce contexte, se trouve l'Arduino, qui est une carte électronique englobant l'ordinateur d'une part, et l'automate programmable d'autre part. c'est une espace de prototypage destinée aux développeurs, aux amateurs, aux entrepreneurs ... dans les domaines de robotique, d'automatique, de mécanique ...

En parallèle, nous vivons l'avancement de l'Android, le système d'exploitation qui répondent aux exigences des téléphones intelligents (smart phone). C'est une plateforme trop nécessaire et importante pour le développement des applications grâce à sa puissance d'utilisation ainsi que ses diverses fonctionnalités.

Android studio est une plateforme open source qui permet à n'importe quel développeur de créer gratuitement ses applications, il n'a plus besoin donc d'autorisation du constructeur du téléphone pour développer ses propres projets.

Dans ce chapitre, notre objectif est de définir les composants et les outils de développement, dans une première partie. Dans la deuxième partie, nous allons présenter le système d'exploitation Android ainsi que l'outil de développement Android studio que nous avons mis en œuvre pour créer notre application.

2.2 Description du problème :

Dans les lignes suivantes, nous allons expliquer les problèmes majeurs pour lesquelles nous avons choisi ce sujet. Durant notre parcours universitaire, nous avons effectué beaucoup de stages au sein de différentes entreprises où des problèmes communs, entre ces différents établissements, ont provoqués des risques mortels, des procédés pénibles, des communications ratées ... vu aux faibles investissements des technologies disponibles, citons par exemple :

- L'utilisation classique et pénible des outils de mesure (Multimètre ...) ;
- Le contrôle exhaustif ;
- L'ignorance des consignes de sécurité ;
- La non fiabilité de flux d'information.

Notre problématique vise à adopter et à exploiter les technologies récentes, la connectivité surtout, pour répondre aux problèmes cités ci-dessus pour servir le confort et la sécurité industrielle. Par conséquent, nous allons développer des lunettes intelligentes permettant les fonctionnalités suivantes :

- La réception des données des outils de mesure;
- La projection des informations des stations surveillées ;
- L'affichage des consignes de sécurité ;
- La notification des flux d'information.

2.3 Partie I : Arduino

La carte Arduino est une plateforme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation afin d'étendre les capacités de relations humain/machine ou environnement/machine [44].

2.3.1 Accessoires Arduino :

2.3.1.1 Arduino UNO :

Selon Erik Bartmann [16], La carte Arduino UNO contient des nombreux éléments tell que :

- **Prise jack** : permet de brancher une alimentation (pile, batterie, adaptateur secteur...);
- **Microcontrôleur** : stocke le programme et l'exécute ;
- **Entrées analogiques** : 6 entrées permettent de brancher des capteurs et des détecteurs analogiques ;
- **Entrées et sorties numériques (Digital)** : 14 entrées/sorties permettent de brancher des actionneurs / détecteurs ;
- **Port USB (5V)** : Permet d'alimenter la carte en énergie et de télé-verser le programme dans la carte ;

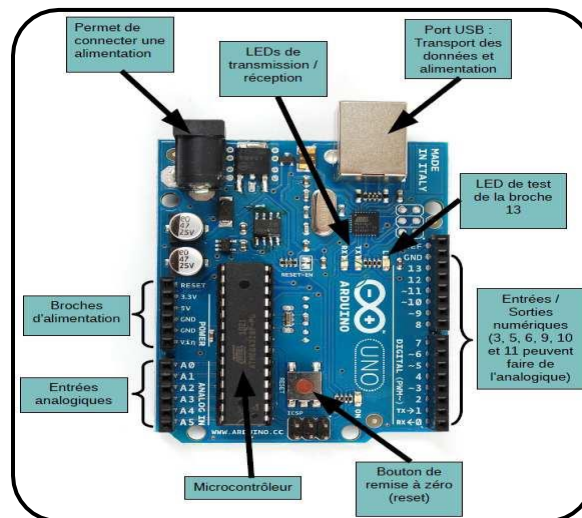


Figure II 1 : Élément de l'Arduino Uno

Entrées et Sorties Numériques : Chacun des connecteurs D0 à D13 peut être configuré par programmation en entrée ou en sortie.

Entrées Analogiques : Contrairement aux entrées/sorties numériques qui ne peuvent prendre que deux états HAUT et BAS, ces six entrées peuvent admettre un millier de valeurs (1024 exactement)

analogiques comprises entre 0 et 5 Volts. Nous pourrions donc avoir des valeurs de tension précises à 5 mV près ($\approx 5V/1024$).

Microcontrôleur : Le microcontrôleur est le cœur de la carte Arduino. L'Arduino UNO est équipée d'un microcontrôleur Atmel AVR de type ATmega328.

Reset : chaque carte Arduino possède un bouton pour redémarrer le microcontrôleur.

Composant	Caractéristique
Microcontrôleur	Atmel ATmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (utilisables en broches E/S numérique)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA max total entrée sortie cumulées)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	500 mA max sur port USB
Mémoire Programme Flash	32 KB dont 0,5 sont utilisé par le bootloader
Mémoire SRAM (mémoire volatile)	2 KB
Mémoire EEPROM (mémoire non volatile)	1 KB
Fréquence d'horloge	16 MHz

Tableau II 1 : Fiche technique d'Arduino Uno

2.3.1.2 Arduino Nano :

La carte Arduino Nano est une version minimisée de la carte Arduino UNO avec moins de porte numérique et analogique, elle est très utile pour créer des petits projets ou tout simplement pour gagner l'espace [18].

On trouve aussi :

- Entrées analogique et numérique ;
- Microcontrôleur ;

- Port USB ;
- Broche reset.

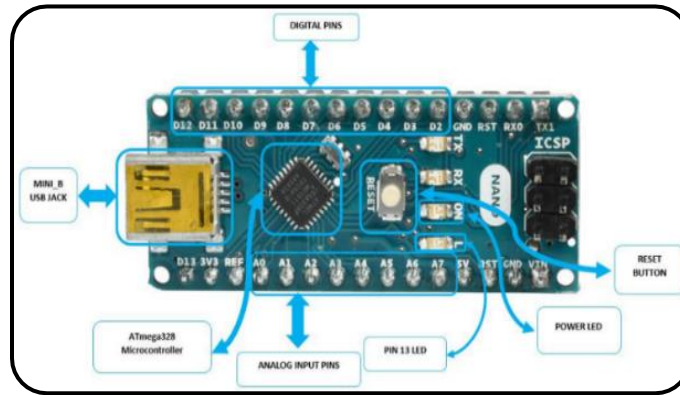


Figure II 2 : Arduino Nano

Microcontrôleur : La version 2.x de l'Arduino Nano est équipée avec ATmega168 contrairement au 3.x qui est basée sur ATmega328 microcontrôleur.

Composant	Caractéristique
Microcontrôleur	Atmel ATmega168 ou ATmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	8 10 bits
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA max total entrée sortie cumulées)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	500 mA max sur port USB
Mémoire Programme Flash	16 KB (ATmega168) ou 32 KB (ATmega328) 2 KB utilisée par le bootloader
Mémoire SRAM (mémoire volatile)	1 KB (ATmega168) ou 2 KB (ATmega328)
Mémoire EEPROM (mémoire non volatile)	512 bytes (ATmega168) or 1 KB (ATmega328)
Fréquence d'horloge	16 MHz

Tableau II 2 : Fiche technique d'Arduino Nano

2.3.1.3 LCD 2*16 :

Le module LCD (Liquid Crystal Display) utilisé dans notre projet est un afficheur à base matricielle avec deux lignes et seize colonnes, chaque colonne peut supporter un seul caractère ou un seul chiffre, pour pouvoir afficher des caractères sur l'écran il nous faudrait activer individuellement chaque pixel de l'écran. Un caractère est représenté par un bloc de 7*5 pixels. Ce qui fait qu'un écran de 16 colonnes et 2 lignes représente un total de $16*2*7*5 = 1120$ pixels [4].

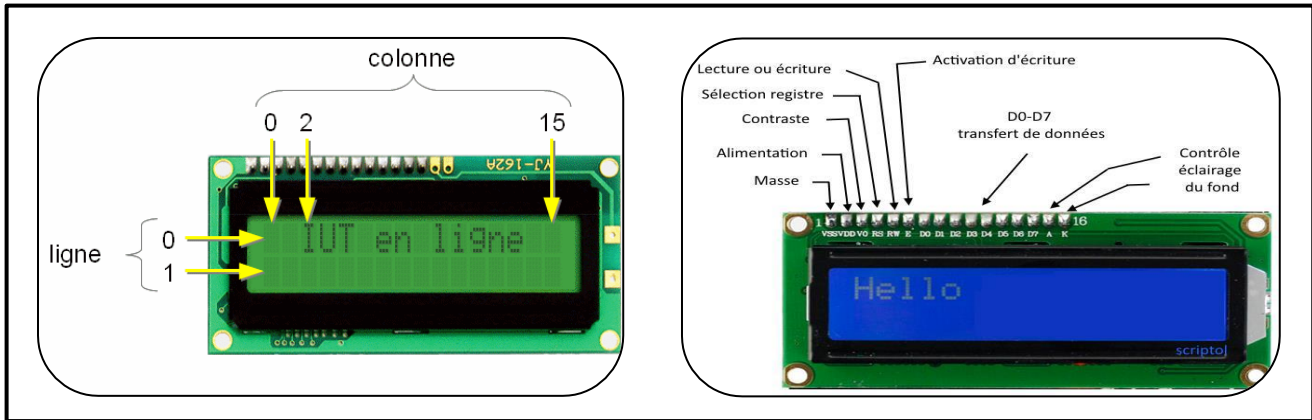


Figure II 3 : Ecran LCD

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

Tableau II 3 : Fiche technique de LCD 2*16

2.3.1.4 Ecran OLED 128*64 :

Grâce à la technologie OLED, on peut maintenant prendre l'avantage d'un afficheur très petit pour les applications ou les projets dont la mobilité et la taille sont des facteurs critiques et essentiels, avec 128 colonnes et 64 lignes poussées par un microcontrôleur SSD1306 et deux moyens de communication IIC ou ISP (par défaut l'écran et en mode ISP).

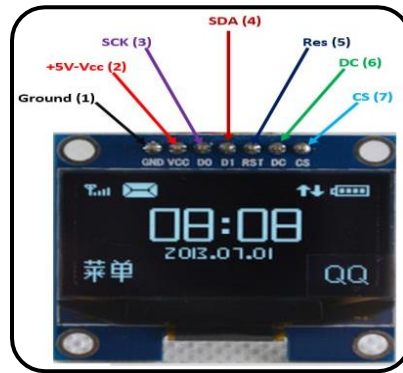


Figure II 4 : Ecran OLED 7 pin

N°	Nom	Rôle
1	GND	Connecté à la terre du circuit
2	VDD, VCC	Peut être alimenté par 3.3V ou 5V
3	SCK (D0, SCL, CLK)	L'affichage prend en charge les IIC et les SPI, pour lesquels l'horloge est fournie par cette broche.
4	SDA (D1, MOSI)	Ceci est la broche de données des deux, il peut être utilisé pour IIC ou pour SPI
5	RES (RST, RESET)	Maintenue momentanément à la terre, cette broche réinitialise le module
6	DC (A0)	Ceci est une broche de commande, peut être utilisé pour SPI ou pour IIC
7	CS	Normalement maintenu bas, utilisé uniquement lorsque plusieurs périphériques SPI sont connectés à la MCU

Tableau II 4 : Fiche technique d'écran OLED

Caractéristiques de l'écran OLED :

- Écran 0,96 ° OLED monochrome à 7 broches (il existe d'autre modèle avec 4,6 ou 8 broches) ;
- Résolution 128 × 64 pixels avec un angle de vision de 160 ° ;
- Tension d'alimentation 3V - 5V (il faut prend en charge les dispositifs logiques 5V et 3,31V) ;
- Utilise SSD1306 pour l'interfaçage et peut donc communiquer via SPI ou IIC.

SPI :

Le SPI est un protocole de communication qui permet à votre Arduino de communiquer très rapidement avec d'autres appareils sur de courtes distances. Il peut s'agir de recevoir des données d'un capteur, d'échanger avec des périphériques tels qu'une carte SD ou un autre microcontrôleur.

IIC :

Le IIC est un protocole de communication destiné à distribuer des signaux sur un grand nombre de sorties.

2.3.1.5 RFID :

La technologie de la RFID est basée sur l'émission de champ électromagnétique par un « lecteur », ou « élément fixe », ou « éléments déportés », qui transmet un signal selon une fréquence déterminée vers une ou plusieurs étiquettes situées dans son champ de lecture. Ce champ électrique ou magnétique sert la liaison entre l'étiquette et son lecteur, et l'activation de ces étiquettes. Une fois "réveillées" par le lecteur, ces étiquettes transmettent alors en retour un signal et un dialogue s'établit selon un protocole de communication prédéfini et les données sont échangées [36].

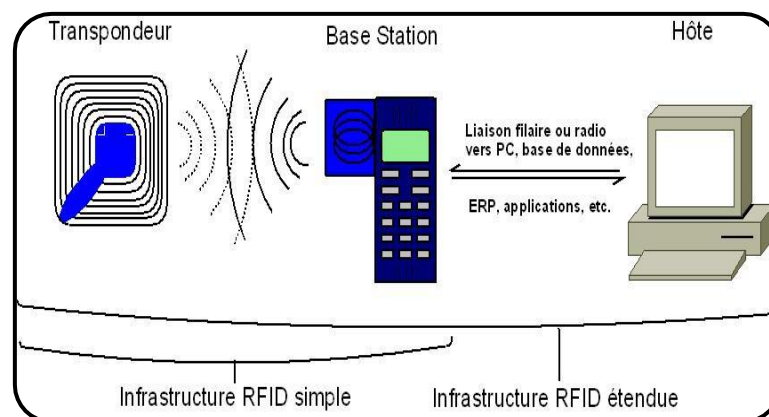


Figure II 5 : Exemple d'une connexion RFID

Composant de RFID :

Le tag (étiquette) :

Une des méthodes d'identification les plus utilisées est d'abriter un numéro de série ou une suite de données dans une puce (chip) et de relier cette dernière à une petite antenne. Ce couple (puce silicium + antenne) est alors encapsulé dans un support (Tag (ou Label) RFID) [36].

Ces "tag" peuvent alors être incorporés dans des objets ou être collés sur des produits. Le tout est alors imprimé sur un support pliable, souvent adhésif. Le format des données inscrites sur les étiquettes est standardisé à l'initiative d'EPC Global (Electronic Product Code) [36].



Figure II 7 : Tag type carte



Figure II 6 : Tag type porte-clés

Le lecteur RFID :

Le lecteur/enregistreur est constitué d'un circuit qui émet une énergie électromagnétique à travers une antenne, et d'une électronique qui reçoit et décode les informations envoyées par le transpondeur et les envoie au dispositif de collecte des données [36].

Le lecteur RFID est l'élément responsable de la lecture des étiquettes radiofréquence et de la transmission des informations qu'elles contiennent (code EPC ou autre, informations d'état, clé cryptographique...) vers le niveau suivant du système (middleware). Cette communication entre le lecteur et l'étiquette s'effectue en quatre temps [36] :

1. Le lecteur transmet par radio l'énergie nécessaire à l'activation du tag ;
2. Il lance alors une requête interrogeant les étiquettes à proximité ;
3. Il écoute les réponses et élimine les doublons ou les collisions entre réponses ;
4. Enfin, il transmet les résultats obtenus aux applications concernées.

La communication entre le lecteur et l'étiquette s'effectue via les antennes qui équipent l'un et l'autre, ces éléments étant responsables du rayonnement radiofréquence.

Mode de fonctionnement :

Il existe plusieurs types de fonctionnement et de communication possibles [36] :

- Lecture seule : il est uniquement possible de lire le transpondeur. Ses informations (et son identité) peuvent avoir été inscrites préalablement par le fabricant, ou le transpondeur peut avoir été livré vierge et c'est l'utilisateur qui en détermine le contenu. Dans ce dernier cas, il s'agit d'une seule écriture, et lecture multiple WORM (Write Once, Read Multiple). Les étiquettes des produits vendus en hypermarchés en sont un exemple.
- Lectures et écritures multiples : l'objectif est la réutilisation du transpondeur et/ou la mise à jour de ses informations. Certaines zones mémoires peuvent n'être programmées qu'un nombre déterminé de fois dans le cas du mode MTP (Multiple Time Programmable), ou programmées de manière illimitée (ou seulement par la technologie) dans le cas du mode R/W (Read/Write). Le suivi sur les chaînes de productions des produits (fabrication, puis mise au point, test, etc.) en est un exemple.
- Lectures et/ou écritures protégées : la protection des données « secrètes » lues ou écrites peut être faite de manière logicielle (mots de passe), ou matérielle (timing particulier, etc.), et appliquée pour tout ou partie de la mémoire. Dans le cas de l'écriture il peut aussi y avoir des zones mémoires programmables de manière unique, dite OTP (One Time Programmable) indépendamment du type de lecture possible.
- Lecture et/ou écriture sécurisées, cryptées : la sécurisation tient à l'authentification des partenaires (station de base – transpondeur) habilités à correspondre ensemble, par des codes évolutifs ou tournants par exemple. Le cryptage des données échangées entre la Station de base et le Transpondeur sert à contrer les écoutes clandestines et pirates.

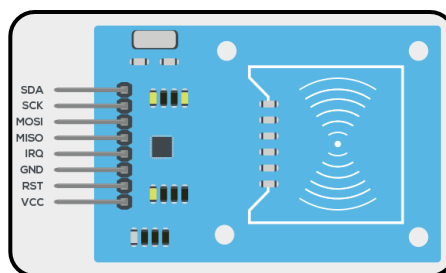


Figure II 8 : Lécuteur RFID

2.3.1.6 MQ :

Le détecteur de gaz est un appareil fait pour donner une alarme lorsque l'atmosphère devient explosible (présence d'un gaz combustible) ou toxique (présence d'un gaz dangereux pour la santé, ou absence d'oxygène) [7].



Figure II 9 : Capteur de gaz MQ2

Les types des capteurs MQ :

1. MQ2 : Capteur de gaz MQ2 est utile pour la détection des fuites de gaz (dans la maison et l'industrie). Il peut détecter le H₂, le GPL, le CH₄, le CO, l'alcool, la fumée, le propane. Basé sur son temps de réponse rapide [48].
2. MQ5 : Le capteur de gaz MQ5 a une grande sensibilité au méthane, au propane et au butane. Le capteur pourrait être utilisé pour détecter différents gaz combustibles, en particulier le méthane, il est à faible coût et convient à différentes applications. [50].
3. MQ7 : Le capteur de gaz monoxyde de carbone MQ7 détecte la concentration de gaz monoxyde de carbone dans l'air et évite sa lecture en tant que tension analogique. Le capteur peut fonctionner à des températures comprises entre -10 et 50 ° C et consomme moins de 150 mA à 5 V [49].

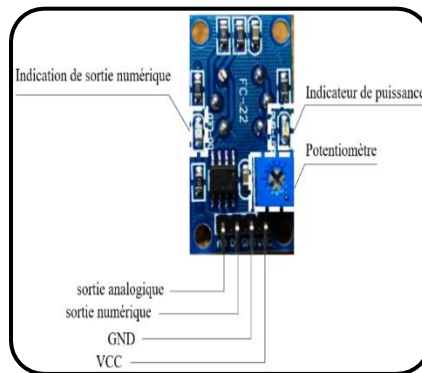


Figure II 10 : Eléments du capteur MQ2

2.3.1.7 LED :

Une DEL / LED : Diode Electro- Luminescente, ou bien " Light Emitting Diode" en anglais. C'est un composant électronique qui crée de la lumière quand il est parcouru par un courant électrique [4].



Figure II 11 : LED

2.3.1.8 Résistance :

La résistance est un composant de base qui s'oppose au passage du courant. On peut l'utiliser pour limiter le courant maximal afin de protéger les composants [4].



Figure II 12 : Résistance

2.3.1.9 Module Bluetooth :

Bluetooth est une norme de communications permettant l'échange bidirectionnel de données à très courte distance en utilisant des ondes radio UHF (ultra haute fréquence) sur une bande de fréquence de 2,4 GHz.

Il existe deux modules de Bluetooth, le HC05 et le HC06. Ils permettent simplement de disposer d'une liaison Bluetooth sur un projet.

Le module HC-05 possède 6 pins. Ce module peut être un maître (il peut proposer à un autre élément Bluetooth de s'appairer avec lui) ou un esclave [9].

Le module HC06 à 4 sorties. Ce module ne peut être qu'un esclave.

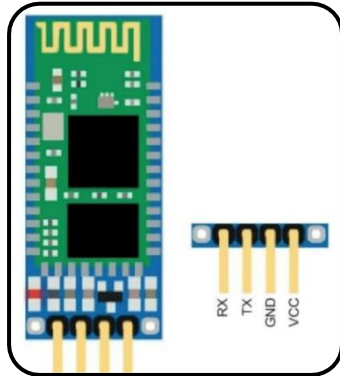


Figure II 14 : Module Bluetooth HC 06

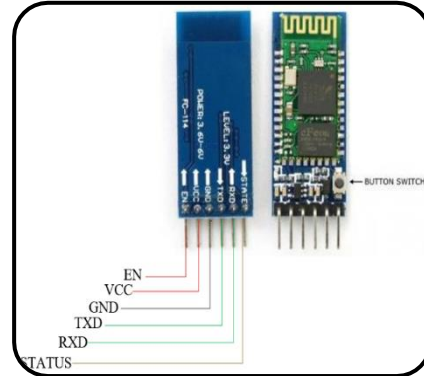


Figure II 13 : Module Bluetooth HC 05

2.3.1.10 Capteur de température et d'humidité :

Capteur de température et d'humidité, Ce composant regroupe à la fois une sonde de température et un capteur d'humidité [45].

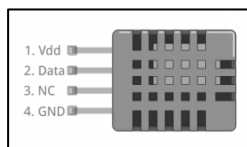


Figure II 15 : Capteur DHT

Broche de DHT :

N°	Nom	Rôle
1	VDD	pour l'alimentation (5 volts ou 3.3 volts).
2	DATA	pour la communication
3	NC	n'est pas utilisée
4	GND	la masse

Tableau II 5 : Broches de capteur DHT

Il y en a deux types, le DHT11 et le DHT22 [45]:

	DHT11	DHT22
Humidité (relative %)	0 ~ 100 %	20 ~ 80%
Précision (humidité)	+/- 2% (+/- 5% aux extrêmes)	+/- 5%
température	-40 ~ +150°C	0 ~ +50°C
Précision (température)	+/- 0.5°C	+/- 2°C
Fréquence mesure max	2Hz (2 mesures par seconde)	1Hz (1 mesure par seconde)
Tension d'alimentation	3 ~ 5 volts	3 ~ 5 volts
Stabilité à long terme	+/- 0.5% par an	+/- 1% par an

Tableau II 6 : Types de capteur DHT

2.4 Partie II : Android

2.4.1 Historique :

L'apparence de l'Android est vue le mois d'octobre de l'année 2003, dès la création de la société *Android Inc.* Elle a commencé en premier temps de fournir des logiciels pour mobiles. Deux années après, elle s'est préparée pour développer un nouveau système d'exploitation pour les téléphones intelligents (smart phone). En 2005, elle est intégrée sous l'entreprise fameuse Google qui lui a permis de lancer la première version *béta* en novembre 2007. Un an après, plus précisément en septembre 2008, elle a sorti sa première version 1.0 sous *HTC dream*. De cet instant là, pas moins de 25 versions Android ont été développées où chacune, de ces dernières, porte comme marque un nom de dessert [20].

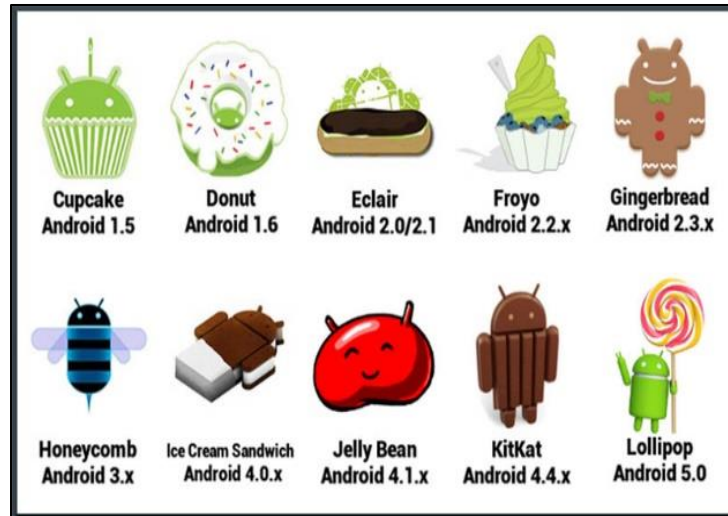


Figure II 16 : Versions d'Android

2.4.2 Etat de la plateforme Android :

D'après Gartner [20], Android est vu comme un système d'exploitation pour smart phone, tablettes tactiles, PDA et terminaux mobiles. C'est une plateforme open source, conçu par Android, utilisant Linux comme noyau. Elle est lancée comme start up, rachetée après par Google, elle a commencé officiellement le 5 novembre 2007. Beaucoup d'appareils utilisent ce système d'exploitation, par exemple : téléviseurs, montres, autoradios, voitures ...

Concrètement ce système est encore gratuit pour les constructeurs ainsi que les développeurs. Android s'articule autour d'un noyau Linux, d'une machine virtuelle Java (JVM), d'un compilateur SDK exécutant les fichiers d'extension (.dex) et non (.class).

L'application est développée en Java sur sa machine virtuelle (JVM), il n'est donc pas possible, à cet effet, d'utiliser toute librairie Java à la création de l'application Android.

2.4.3 Architecture de plateforme Android :

Selon la démarche *botton up*, l'architecture de la plateforme Android se compose de quatre piliers fondamentaux (noyau Linux, librairie et environnement d'exécution, module de développement d'applications, différentes applications) ...

2.4.4 Outils de développement sous Android :

2.4.4.1 Environnement :

Aujourd'hui, beaucoup d'outils sont créés pour permettre aux développeurs de fournir leurs applications sous Android, citons par exemple :

- Android Studio ;

- Eclipse ;
- Netbeans ;
- xPerfect.

2.4.4.2 Langage java :

Java est défini comme étant un langage de programmation orienté, inventé par Sun Microsystème, paru en 1995, racheté en 2009 par Oracle. Une application développée en Java est facilement portable sur plusieurs systèmes d'exploitation exécutant l'extension "APK" [27].

2.4.4.3 SDK (software développement kit):

Le SDK représente un package d'outils mis à disposition, par les éditeurs Android, pour permettre aux développeurs de fournir leurs applications dans un environnement. Le SDK est un kit spécifique pour Android, il est, tout à fait, possible, aux développeurs, d'installer ou de mettre à jour les SDK dont ils ont besoins grâce à Android SDK manager [31]. Le SDK comprend :

- Bibliothèques Java.
- Emulateur pour tester des application AVD.
- Outils de communication avec les vraies tablettes ADB.

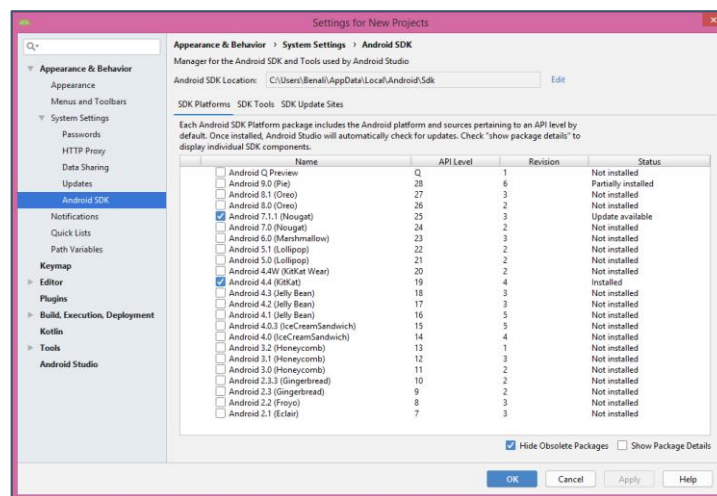


Figure II 17 : Différentes API Android

D'après la figure précédente, il existe plusieurs versions Android. Une application développée sur une version donnée ne fonctionnera pas sur une version inférieure [31], il faut donc bien choisir l'API de la version.

2.4.4.4 JDK (java développement kit) :

Il est noté qu'une application Android développée est officiellement écrite en Java (qui se base sur une machine virtuelle pour se lancer (JVM)). Pour obtenir une JVM sur un ordinateur, il faut plutôt installer le JRE qui comprend la machine virtuelle (JVM) ainsi que les bibliothèques standard de Java.

La JVM ne traite pas le code Java, elle traite un code compilé. Alors, pour compiler un code Java écrit par le développeur, il faut mettre en présence des outils spécifiques, ces outils sont essentiellement compris dans le JDK.

N.B : le JRE est inclus dans le JDK et pratiquement la JVM. Un développeur doit forcément avoir le JDK. Quant à un utilisateur ordinaire, le JRE (Java Runtime Environment) est largement suffisant.

2.4.5 Architecture d'un projet Android :

L'architecture d'un projet Android a, tout à fait, changé depuis que l'outil de développement Android Studio est mis en œuvre. Sous Android, nous utilisons plusieurs projets avec toutes dépendances entre chacune. Un seul projet contient plusieurs modules [46]. Un projet se compose donc de deux parties :

- Un fichier XML déterminant le design de l'application (interface graphique) ;
- Un fichier Java déroulant les fonctionnalités de l'application.

La figure suivante montre les modules d'un projet :

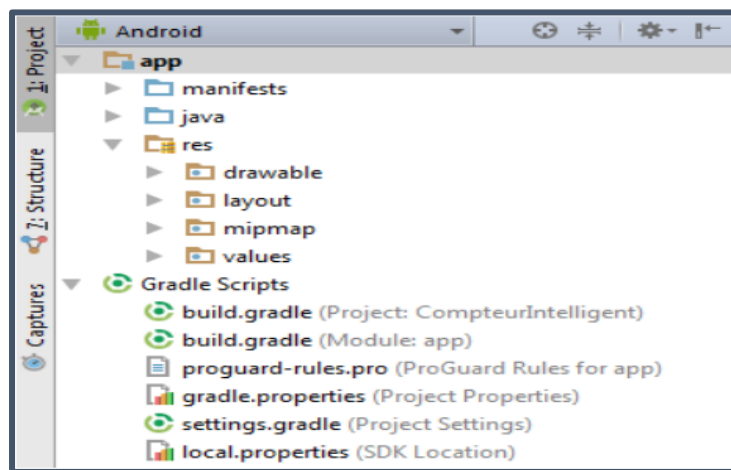


Figure II 18 : Modules d'un projet Android

A la création d'un projet, une liste de modules, représentant les ressources d'application, est automatiquement générée. Une liste bien structurée et précisément définie qu'un développeur doit, un par un, respecter. Cette structure permet de bien simplifier le développement en rendant le projet bien organisé et plus lisible. Voici une brève explication de quelques modules générés :

- **AndroidManifest.xml** : un fichier d'extension xml représentant la carte d'identité de l'application. Un fichier trop nécessaire pour l'application, il autorise certaines opérations, l'utilisation du module Bluetooth par exemple. Ce fichier xml contient de plus le nom de l'application, son logo ...

- **res** : une bibliothèque contenant les ressources décrivant le design de l'application, comme par exemple (les images, les arrières plan ...). La bibliothèque est généralement structurée de telle manière :
 - **res/drawable** : toute source de type image ;
 - **res/layout** : assure la création de l'interface graphique ;
 - **res/values** : traite la taille et les couleurs des textes.
- **build.gradle** : un fichier qui établit la synchronisation du projet avec le moteur de production Gradle. Un nouveau projet doit être bien synchronisé pour permettre aux développeurs d'utiliser l'outil de développement.

Après avoir compilé l'application, un fichier APK est généré, ce fichier doit être ensuite déployé par d'autres outils spécifiques pour utiliser la version réelle de l'application.

2.4.6 Création d'une application Android :

Etape 01 :

Après avoir installé l'outil de développement Android Studio, nous allons confronter la première fenêtre "Welcome to Android Studio" (figure II 19). Pour créer un nouveau projet, nous allons sélectionner "start a new Android Studio project".

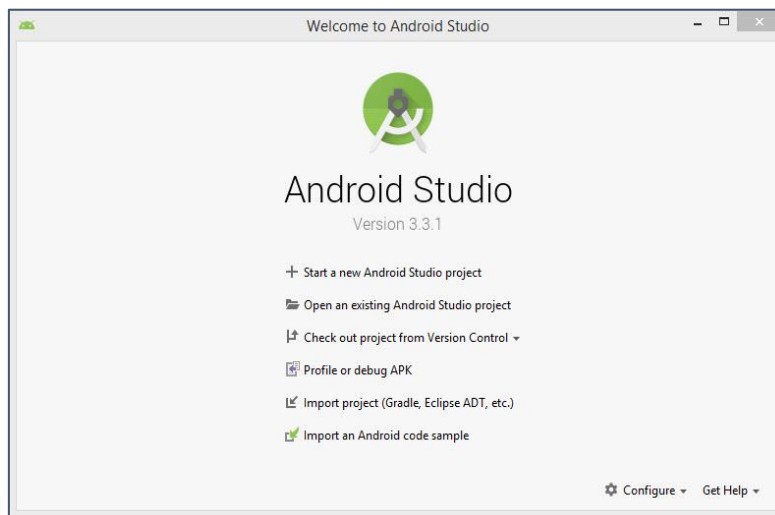


Figure II 19 : Page d'accueil de l'Android Studio

Etape 02 : nous choisissons de plus l'interface graphique depuis cette liste proposée par Android Studio. Pour notre application, nous allons choisir "Empty Activity".

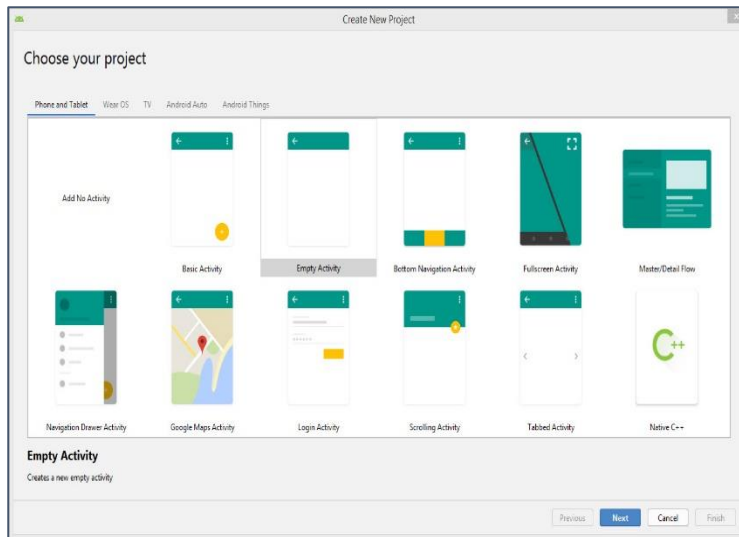


Figure II 20 : Choix de l'interface graphique

Etape 03 : comme la montre la figure II 21, une deuxième fenêtre apparaît :

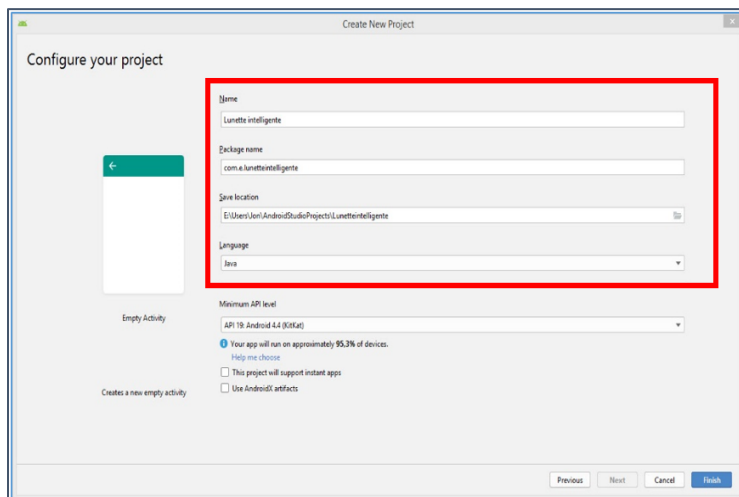


Figure II 21 : Création des paramètres de l'application

Dans cette fenêtre, nous pouvons remplir les informations suivantes :

- **Name** : représente le nom de l'application.
- **Package name** : c'est un nom unique pour identifier l'application en Google Play store.
- **Save location** : choisir l'emplacement d'enregistrement de projet.

Etape 04 : dans cette étape, nous allons terminer la configuration préalable. Nous allons donc choisir la version minimale de l'Android sur laquelle notre application pourra fonctionner.

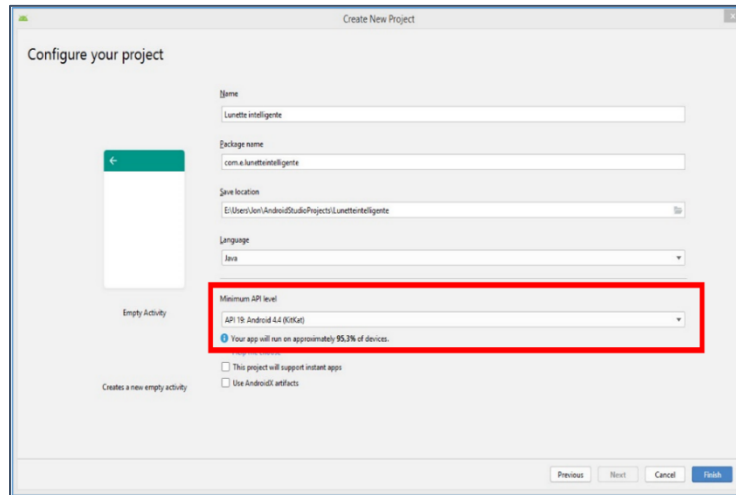


Figure II 22 : Choix de niveau d'API

2.4.7 Conception de l'application :

Dans les lignes qui suivent, nous allons développer, pas à pas, une application qui permet aux superviseurs de contacter leurs employées à travers des lunettes intelligentes. L'application doit servir, via une transmission Bluetooth, l'envoi des messages depuis les smart phone envers les lunettes intelligentes. Pour cela nous avons structuré l'application comme suit :

- Une interface d'authentification pour introduire le nom d'utilisateur ainsi que le mot de passe ;
- Une interface de transmission de message.

La figure suivante représente le logo choisi pour notre application :



Figure II 23 : Logo de l'application

Pour que l'application ne soit utilisée que par l'agent défini à sa base de données, nous avons sécurisé l'utilisation en ajoutant deux champs pour le nom d'utilisateur ainsi que son mot de passe :

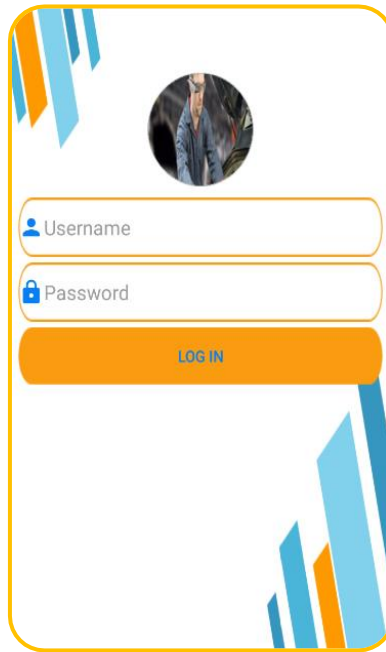


Figure II 24 : Interface d'authentification

La figure N° 38 présente l'interface de transmission des messages :

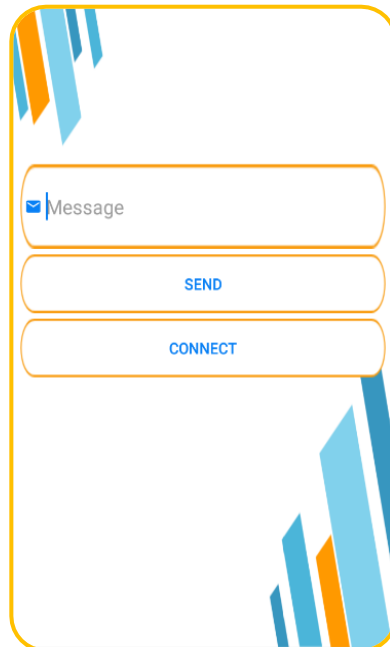


Figure II 25 : Interface de transmission

Cette interface contient trois cases :

- Le bouton : "**Connect**" : pour connecter l'application avec l'adresse Bluetooth de la lunette à laquelle nous voudrions transmettre le message ;
- Le champ : "**Message**" : pour saisir le message à envoyer ;
- Le bouton : "**Send**" : pour envoyer le message.

2.4.8 Développement de l'application :

Dans cette partie, nous allons expliquer les étapes de développement de notre application :

Etape 01 (synchronisation du projet) : pour bien permettre au module Bluetooth de fonctionner, nous devons, à nouveau, synchroniser le projet en ajoutant la dépendance suivante au fichier build.gradle :

```
compile 'com.akexorcist:bluetoothspp:1.0.0'
```

Cela permet au fichier Java d'utiliser la librairie BluetoothSpp que nous avons choisi pour l'émission des données.

Etapes 02 (permission de module Bluetooth) : nous devons maintenant déclarer deux permissions au fichier AndroidManifest en vue d'autoriser l'application de transmettre ou de recevoir des données via Bluetooth. Nous allons ajouter les permissions suivantes :

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Ces étapes expliquées ci-dessus dépendent de tout ce qui autorisation pour l'application. Nous allons par la suite montrer la création des interfaces de notre projet.

Etape 03 (l'interface d'authentification) : comme l'a montré la figure II 24, nous avons quatre éléments principaux (arrière-plan, username, password, bouton login). Voici les instructions dépendant de chacune :

```
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="8dp"
    tools:context=".MainActivity"
    android:background="@drawable/background">
```

```
<EditText
    android:id="@+id/username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:background="@drawable/user"
    android:drawableLeft="@drawable/ic_action_name"
    android:ems="10"
    android:hint="Username"
    android:inputType="textPersonName"
    android:textColor="@color/colorText" />
```

```
<EditText
    android:id="@+id/password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:background="@drawable/user"
    android:drawableLeft="@drawable/ic_action_lock"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    android:textColor="@color/colorText" />
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="5dp"
    android:background="@drawable/button"
    android:onClick="Log"
    android:text="Log in"
    android:textColor="@color/colorText" />
```

Etape 04 (interface d'émission) : dans la figure II 25, nous avons déclaré les trois éléments suivants : la case de message à envoyer "Message", le bouton de connexion "Connect" et le bouton d'envoi "Send". Les instructions, qui suivent, représentent la mise en page de chaque élément :

```

<EditText
    android:id="@+id/message"
    android:layout_width="match_parent"
    android:layout_height="75dp"
    android:layout_marginTop="5dp"
    android:background="@drawable/user"
    android:drawableLeft="@drawable/ic_action_message"
    android:ems="10"
    android:hint="Message"
    android:inputType="textPersonName"
    android:textColor="@color/colorText" />

```

```

<Button
    android:id="@+id/connect"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="5dp"
    android:background="@drawable/user"
    android:text="connect"
    android:textColor="@color/colorText" />

```

```

<Button
    android:id="@+id/ssend"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="5dp"
    android:background="@drawable/user"
    android:text="Send"
    android:textColor="@color/colorText" />

```

Par ces deux dernières étapes, les interfaces de notre application sont, tout à fait, développées, ce qu'il nous reste encore, c'est de développer le programme Java pour donner de sens aux éléments mis en place.

Etape 05 (programmation Java) : dans un premier temps, nous importons les bibliothèques montrées ci-dessous. Ces bibliothèques sont trop utiles pour les instructions restantes :

```

import app.akexorcist.bluetoothspp.library.BluetoothSPP;
import app.akexorcist.bluetoothspp.library.BluetoothState;
import app.akexorcist.bluetoothspp.library.DeviceList;

```

```
import android.support.constraint.ConstraintLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
```

Après, une liste de paramètres doit être générée en faisant appel à la fonction onCreate (). Nous parlons de :

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    bluetooth = new BluetoothSPP(context, this);

    ssend = (Button) findViewById(R.id.ssend);
    connect = (Button) findViewById(R.id.connect);
}
```

Comme notre application exige le module Bluetooth comme moyen de communication, nous devons vérifier sa présence au smart phone en ajoutant le code suivant :

```
if (!bluetooth.isBluetoothAvailable()) {
    Toast.makeText(getApplicationContext(), text: "Bluetooth is not available", Toast.LENGTH_SHORT).show();
    finish();
}
```

Si, encore, ce module Bluetooth est connecté avec d'autre module, le bouton "Connect" affiche ce module auquel nous allons envoyer nos données par exemple "Connected to Glass01". Cette opération est exécutée par le code :

```
public void onDeviceConnected(String name, String address) {
    connect.setText("Connected to " + name);
}
```

Si la connexion n'est pas établie avec un module donné, le bouton doit afficher "Connexion failed" pour avertir l'utilisateur :

```
public void onDeviceConnectionFailed() { connect.setText("Unable to connect"); }
```

En cas où la connexion est perdue, le bouton affiche "Connexion lost" :

```
public void onDeviceDisconnected() { connect.setText("Connection lost"); }
```

Nous avons expliqué les trois situations sur lesquelles peut se trouver le bouton "**Connect**". Mais en cliquant réellement sur ce bouton, nous rencontrons tous les modules auxquelles nous pouvons connecter notre smart phone comme suit

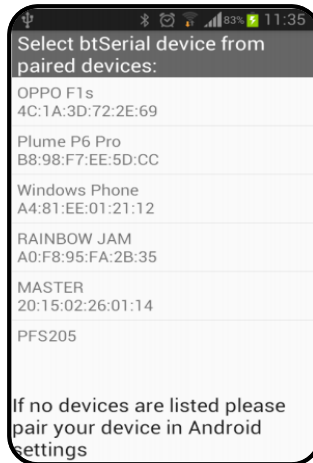


Figure II 26 : Interface de communication

Ces opérations sont contrôlées par le code ci-dessus :

```
connect.setOnClickListener((v) -> {
    if (bluetooth.getServiceState() == BluetoothState.STATE_CONNECTED) {
        bluetooth.disconnect();
    } else {
        Intent intent = new Intent(getApplicationContext(), DeviceList.class);
        startActivityForResult(intent, BluetoothState.REQUEST_CONNECT_DEVICE);
    }
});
```

En cliquant sur le bouton "**Send**", le message saisi est envoyé immédiatement à la lunette connectée, ceci est effectué par la classe suivante :

```
ssend.setOnClickListener((v) -> {
    EditText Message = (EditText) findViewById(R.id.message);
    if (!(Message.getText().toString().isEmpty())) {
        bluetooth.send(Message.getText().toString(), CRLF: true);
    } else {Toast.makeText(getApplicationContext(), text: "You need to write something", Toast.LENGTH_LONG).show();}
});
```

Les codes ci-dessus reviennent à l'interface d'émission, nous passons par la suite aux codes de l'interface d'authentification :

```
public void Log (View view){
    EditText username = (EditText) findViewById(R.id.username);
    EditText password = (EditText) findViewById(R.id.password);
    if ((username.getText().toString().equals("Mounir") && password.getText().toString().equals("1234"))
        || (username.getText().toString().equals("Nadjib") && password.getText().toString().equals("Naj123"))){

        Toast.makeText(getApplicationContext(), text: "Hello "+username.getText().toString(), Toast.LENGTH_LONG).show();
        LinearLayout MessageLayout = (LinearLayout) findViewById(R.id.messageLayout);
        LinearLayout LogLayout = (LinearLayout) findViewById(R.id.logLayout);
        LogLayout.setVisibility(View.INVISIBLE);
        MessageLayout.setVisibility(View.VISIBLE);

    } else {Toast.makeText(getApplicationContext(), text: "wrong password or username", Toast.LENGTH_LONG).show();}
```

Dès que nous cliquons sur "**Login**", l'application compare le nom d'utilisateur introduit ainsi que le mot de passe avec ceux de base de données, en cas d'authentification, l'utilisateur est autorisé à dérouler l'application.

2.5 Conclusion :

Pour ce chapitre, nous avons présenté les différents outils de travail que nous aurons besoin pour concevoir et pour réaliser notre lunette intelligente ainsi que les différents accessoires compatibles. Nous avons de plus expliqué le développement de l'application Android. Un Smart phone peut encore se connecter à la lunette intelligente en envoyant des messages, des ordres, des tâches à exécuter, des démarches à suivre ...

Le développement de la lunette intelligente ainsi que ces accessoires (le voltmètre, la station de surveillance et la lecture de consigne de sécurité) sera bien détaillé dans le troisième chapitre.

Chapitre III : Conception et réalisation

3.1 Introduction :

Dans ce chapitre, nous allons illustrer en détail le fonctionnement de notre lunette intelligente en expliquant la compatibilité de celle-ci avec les stations de sécurité montrées aux chapitres suivante.

Nous allons, tout d'abord, passer par la conception puis la réalisation de notre projet à l'aide de la conception assistée par ordinateur CAO en Solidworks. Nous allons de plus mettre les documentations des dessins techniques. Ainsi, nous allons prouver notre solution pour les problématiques de l'industrie en terme de sécurité et de surveillance.

L'objectif principal de ce chapitre est de réaliser le modèle supposé et d'interpréter les résultats obtenus en comparant ces derniers avec les hypothèses estimées. Nous devons plutôt parler de :

- La simulation des schémas électriques ;
- La programmation des microcontrôleurs ;
- L'interprétation des résultats obtenus.

3.2 SolidWorks :

3.2.1 Définition :

D'après son site officiel, Solidworks est une application de conception mécanique 3D qui sert aux concepteur de concrétiser facilement leurs idées, d'expérimenter des fonctions et des côtes en vue de réaliser des modèles et des mises en plan (dessins techniques) précis [47].

3.2.2 Pourquoi SolidWorks ?

Parmi plusieurs logiciels de conception et de modélisation 3D, nous avons choisi Solidworks pour certaines raisons telles que :

- Une interface simple et un confort visuel ;
- Un accès rapide aux librairies des composants ;
- Un outil de vérification efficace ;
- Une modélisation 3D performante ;
- Une application de mises en plan efficiente.

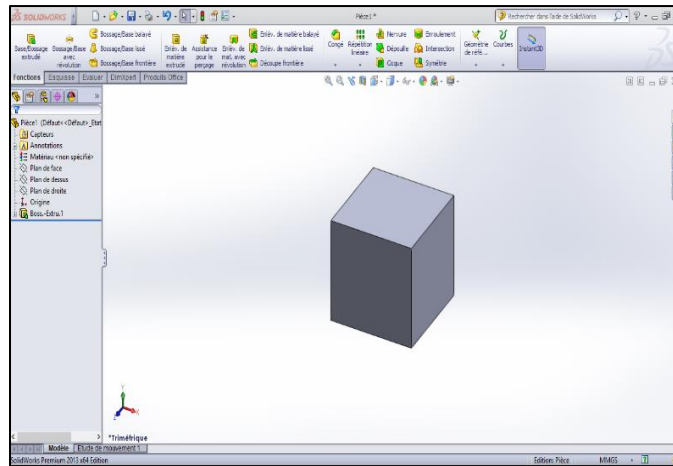


Figure III 1 : Interface graphique SolidWorks

3.2.3 Mises en plan :

Les mises en plan représentent des documentations 2D permettant de transmettre une conception à la fabrication [47].

On peut distinguer deux types :

- Dessin d'ensemble : pour présenter la pièce dans son ensemble ;
- Dessin de définition : pour présenter la pièce de manière détaillée.

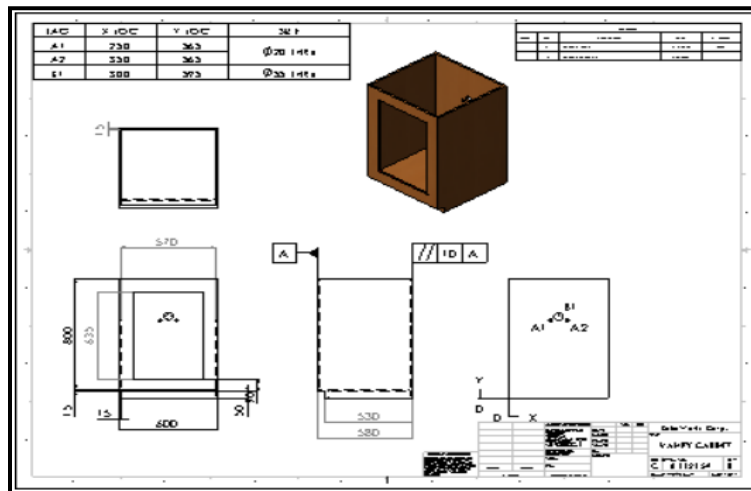


Figure III 2 : Exemple d'un mis en plan

Pour mettre en évidence la réalisation de notre projet, nous allons en certaines étapes, expliquer la démarche suivie dès la conception à la réalisation.

3.3 Lunette intelligente

3.3.1 Description du produit :

Il s'agit d'un dispositif qui :

- Utilise une connexion sans fil (Bluetooth (BT)) ;
- Délivre des informations en temps réel ;
- Sert la lecture de manière sécurisé et confortable ;
- Renforce la sécurité industrielle ;
- Est compatible avec de multiple accessoires (Voltmètre, Android ...) ;
- Contient une expansion Android ;
- Est esthétique et exige peu d'entretien ;
- Est portable et mobile.



Figure III 3 : Lunette intelligente développée

La figure suivante représente la lunette intelligente et les accessoires développés :

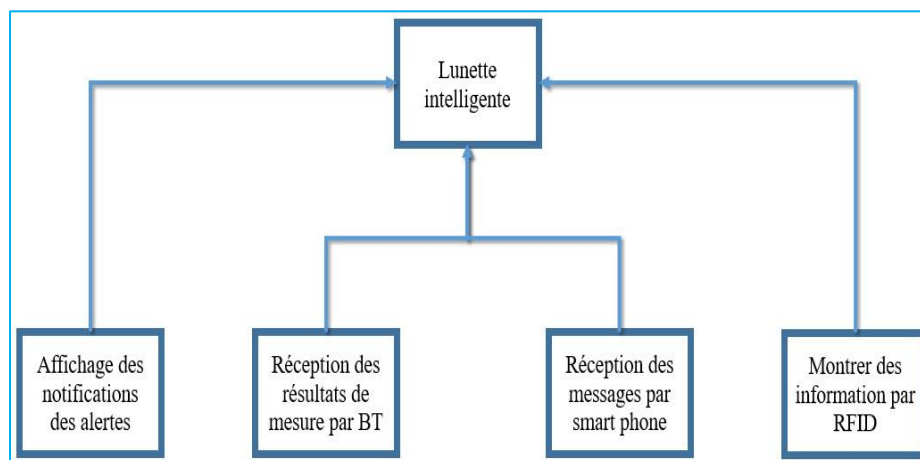


Figure III 4 : Fonctionnalités développées

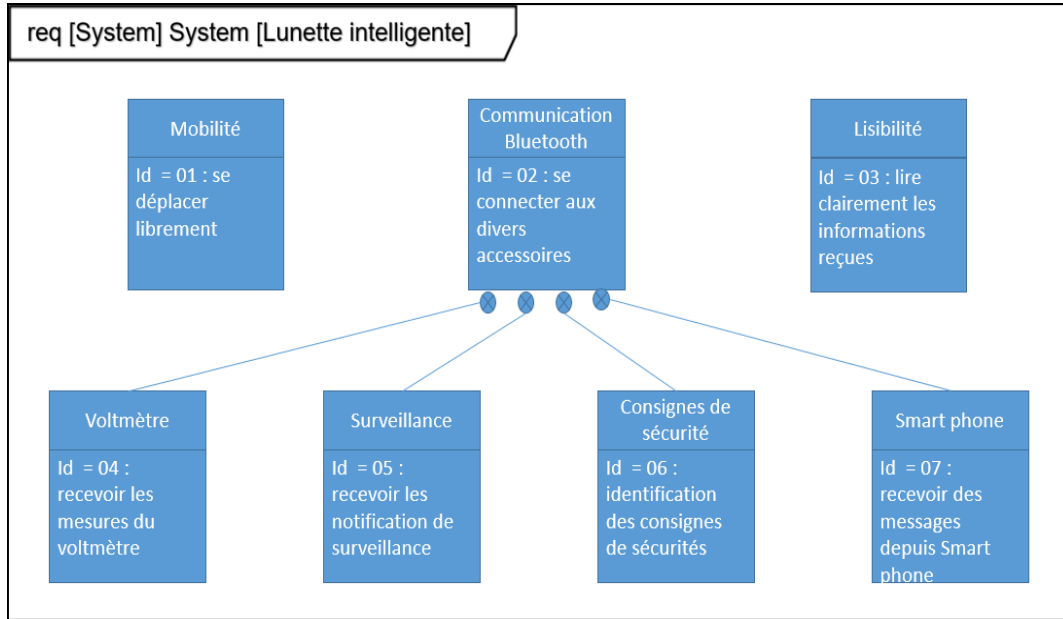


Figure III 5 : Diagramme d'exigence

3.3.2 Etape 1 : Conception

Afin de répondre aux fonctionnalités attribuées à nos lunettes intelligentes, nous avons commencé par concevoir ce modèle en utilisant Solidworks, ceci permet de visualiser ou de concrétiser l'idée initiale avant l'étape de la réalisation du produit. La figure III 6 démontre la conception finale.

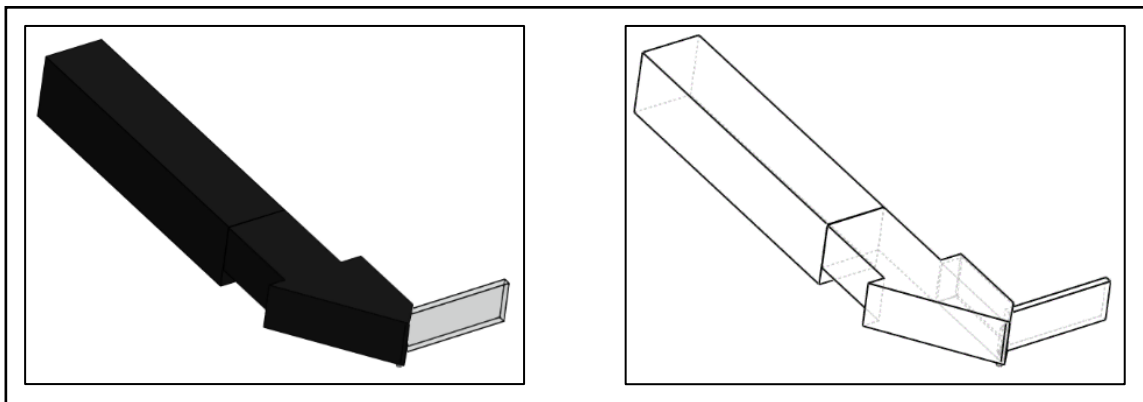


Figure III 6 : Conception des lunettes intelligentes

3.3.3 Etape 2 : Mises en plan

Autrement dit dessin technique ou dessin industriel sert à renforcer la communication entre la phase précédente et celle de la réalisation. Dans cette étape, nous utilisons des représentations normalisées basées sur la projection de la pièce pour bien renseigner la taille et les parties concernées (figure III 7).

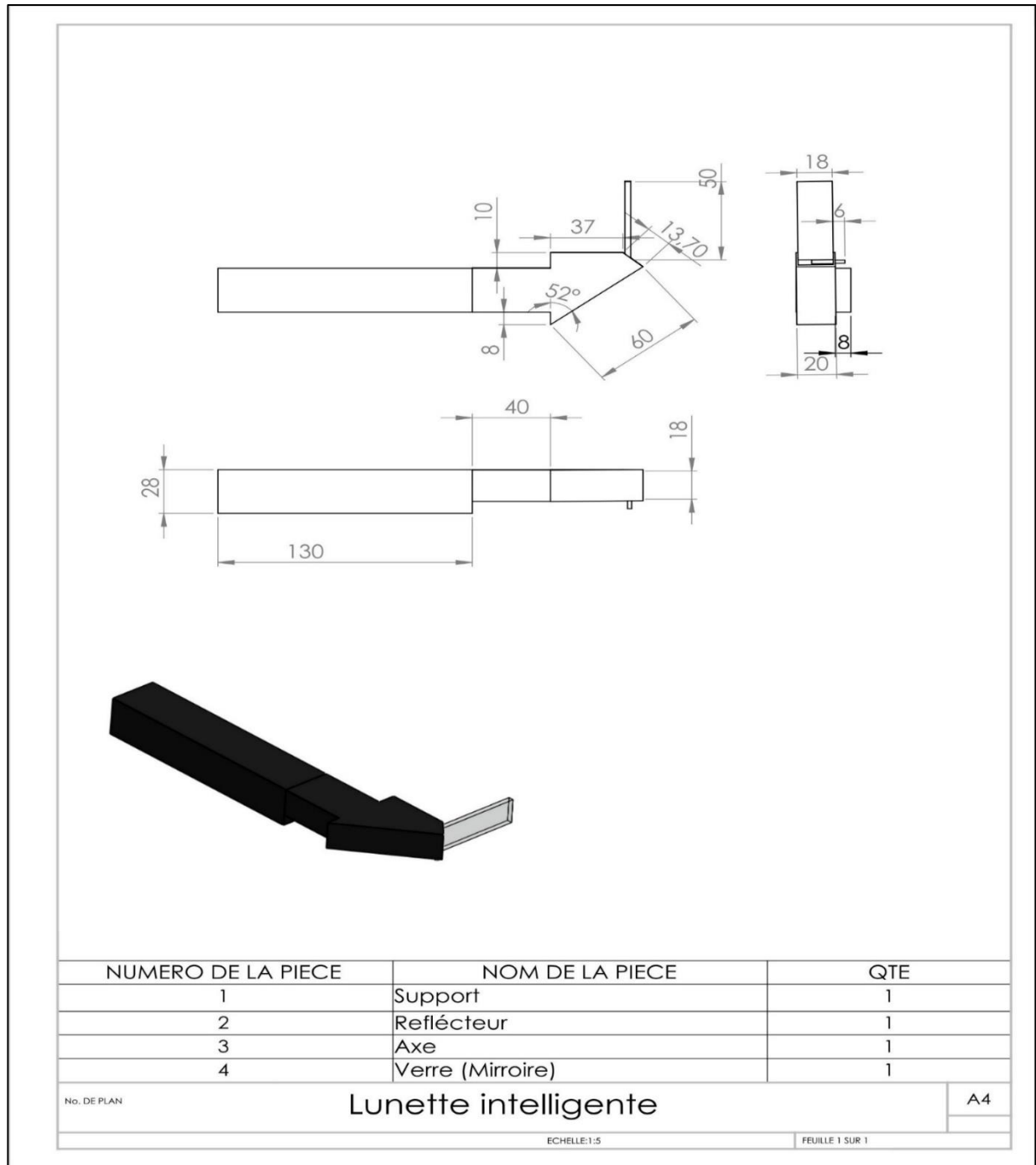


Figure III 7 : Mise en plan des lunettes intelligentes

3.3.4 Etape 3 : Simulation du circuit électrique

Une étape importante dans un projet électronique, c'est de créer et de simuler les schémas électriques des composants choisis tout pour bien vérifier leurs interactions et leurs comportements avec le microcontrôleur.

Pour présenter les schémas électriques, nous avons choisi le logiciel fritzing :

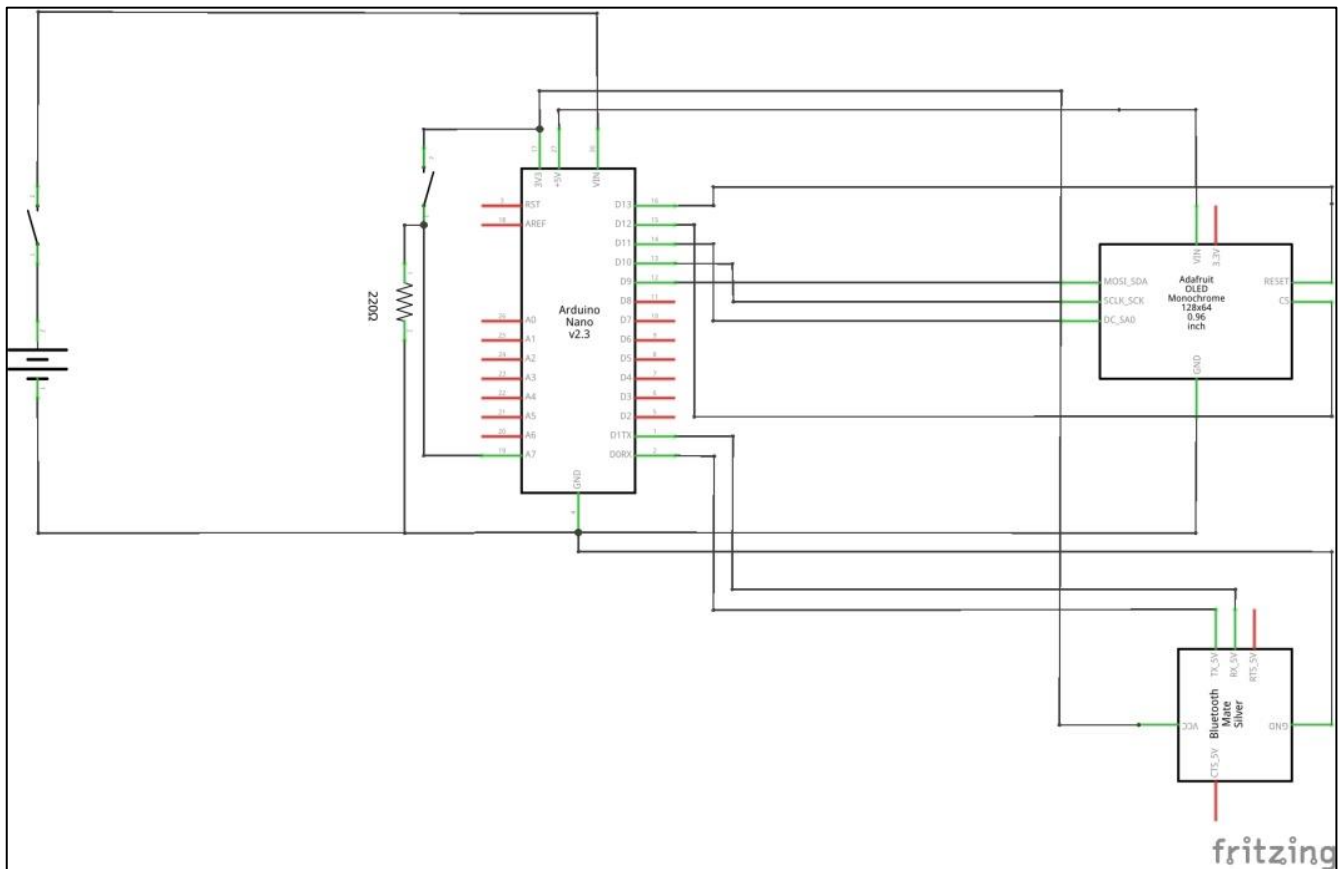


Figure III 8 : Shéma électrique des lunettes intelligentes

Pour la simulation, nous avons choisi le logiciel ISIS :

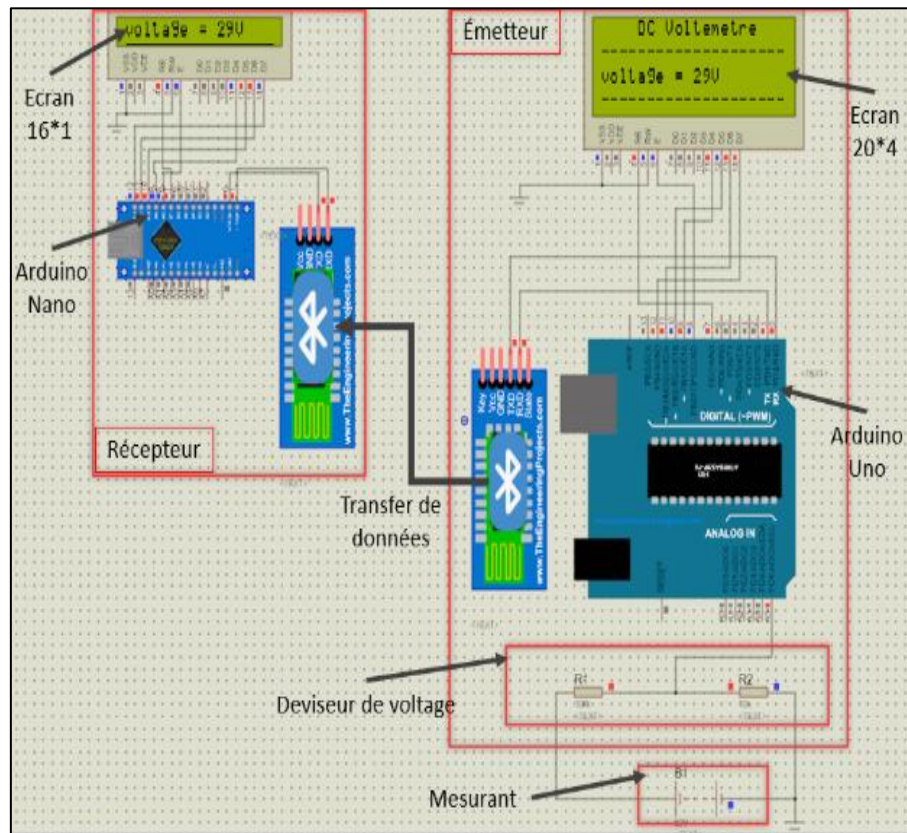


Figure III 9 : Simulation des lunettes intelligentes

La figure précédente montre une simulation graphique du schéma électrique des lunettes intelligentes, nous pouvons remarquer deux parties : une partie émetteur, elle est constituée d'un système de mesure (voltmètre) et un système de transfert d'information, la deuxième partie représente notre lunette intelligente, elle est composée d'un afficheur, d'une carte Arduino et d'un module Bluetooth.

3.3.5 Etape 4 : Programmation

Le programme Arduino contient les commandes à suivre par le microcontrôleur. Comme nos lunettes intelligentes sont conçues pour servir deux fonctionnalités (Mesure et Alerte) nous avons programmé le microcontrôleur en déclarant les instructions suivantes :

- **1^{er} pas** : nous devons définir l'écran OLED pour être reconnue par le microcontrôleur, ceci est effectué par le code suivant :

```

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 6
#define OLED_D1 9
#define OLED_D0 10
#define OLED_DC 11
#define OLED_CS 12
#define OLED_RES 13

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
    OLED_D1, OLED_D0, OLED_DC, OLED_RES, OLED_CS);
#define NUMFLAKES 10

```

- **2^{ème} pas** : Nous allons initialiser les paramètres à utiliser :

```

String vlt = "0.00 V";
int button = 7;

```

- **3^{ème} pas** : les paramètres initialisés doivent être ensuite définis en terme de fonction, tels que les entrées/sorties et la vitesse de communication (Bluetooth) :

```

void setup() {
  Serial.begin(115200);
  if(!display.begin(SSD1306_SWITCHCAPVCC)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
    pinMode(button, INPUT);
  }

  display.clearDisplay();
  display.drawPixel(1, 1, WHITE);
}

```

- **4^{ème} pas** : le code principal que l'Arduino Nano va exécuter est intégré dans la boucle (*void loop*). Puisque nous avons deux fonctionnalités, nous permutons entre elles en utilisant un bouton connecté avec le pin 7, si le pin reçoit un signal électrique, le microcontrôleur exécute la première fonctionnalité (Alerte), sinon il exécute la deuxième (Mesure). Cela est contrôlé par l'instruction :

```

if (digitalRead(button) == HIGH){fonctionnalité 1}else {fonctionnalité 2}

```


Voici le code de la première fonctionnalité :

```

if(Serial.available() > 0){
  vlt = Serial.readString();

}
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(30, 0);
display.println("DC Voltmetre");
display.setCursor(0, 10);
display.println("-----");
display.setTextSize(3);
display.setCursor(10, 20);
display.println(vlt);
display.setCursor(70, 20);
display.setTextSize(1);
display.setCursor(0, 50);
display.println("-----");
display.display();
delay(100);
display.clearDisplay();
vlt="0.00 V";
}

```

Voici le code de la deuxième :

```

if(Serial.available() > 0){
  Message = Serial.readString();
  display.setTextColor(WHITE);
  display.setTextSize(2);
  display.setCursor(20, 20);
  display.println("ALERTE");
  display.display();
  delay(1000);
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(2);
  display.setCursor(0, 0);
  display.println(Message);
  display.display();
  delay(2000);
  display.clearDisplay();
}
else{
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(0, 0);
  display.println(".");
  display.display();
  delay(50);
  display.clearDisplay();
}
}

```

3.3.6 Etape 5 : Montage final

Jusqu'à présent, nous avons réussi la simulation des circuits électriques ainsi que la programmation des microcontrôleurs, nous allons encore fusionner le tout dans un seul système en respectant la conception et le schéma électrique présentés précédemment.

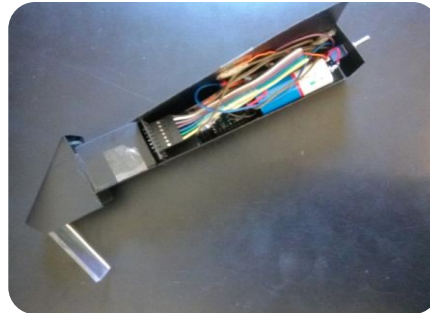


Figure III 10 : Vue interne des lunettes intelligentes

Avant de tester la fiabilité et d'exposer les interprétations des résultats de nos dispositifs, nous allons par la suite présenter les divers accessoires compatibles.

3.4 Accessoires développés :

Au cours de notre projet, nous avons développé quatre accessoires :

3.4.1 Voltmètre :

3.4.1.1 Description du prototype :

Nous avons créé ce prototype pour tester la possibilité de connecter nos lunettes intelligentes avec les outils de mesure (Multimètre par exemple). Notre prototype consiste à :

- Mesurer le voltage d'un équipement donné ;
- Transférer les valeurs mesurées via connexion sans fil (Bluetooth).

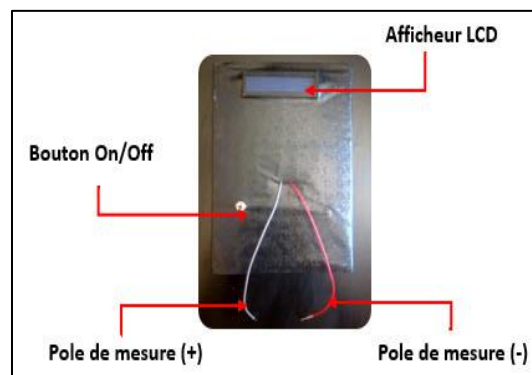


Figure III 11 : Voltmètre

3.4.1.2 Etape 1 : Simulation du circuit électrique

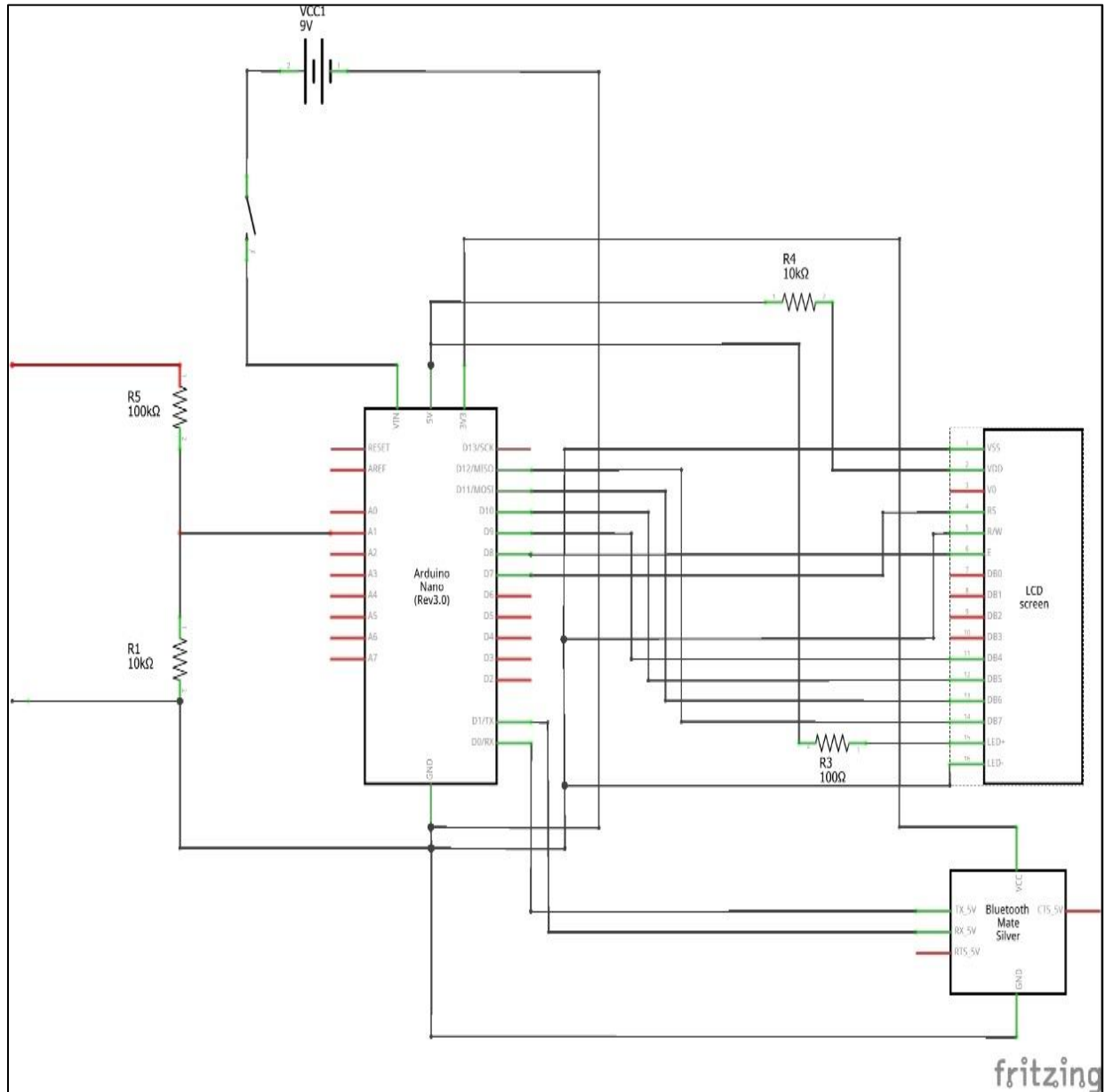


Figure III 12 : Shéma électrique du voltmètre

3.4.1.3 Etape 2 : Programmation

Voici le code utilisé pour mesurer le voltage :

```
include <LiquidCrystal.h>
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
int Vpin=A5;
float voltage;
float volts;
float vin =0;
float R1 = 100000.0;
float R2 = 10000.0;
void setup() {
  lcd.begin(16,2);
  Serial.begin(115200);
}
void loop() {
  lcd.setCursor(2,0);
  lcd.print("DC Voltmeter");
  voltage = analogRead(Vpin);
  volts = voltage*5/1024;
  vin = volts/(R2/(R1+R2));
  if (vin<1) {vin=0.0;}
  lcd.setCursor(3,1);
  lcd.print(vin);
  lcd.print(" Volts");
  Serial.print(vin);
  Serial.print(" V");
  delay(1150);
  lcd.clear();
}
```

3.4.1.4 Etape 3 : Montage final

Après l'assemblage des différents composants, nous obtenons l'accessoire suivant :



Figure III 13 : Vue interne du voltmètre

3.4.2 Station de surveillance :

3.4.2.1 Description du prototype :

La station de surveillance permet de surveiller les facteurs prédéfinis par l'entreprise de manière intelligente. Dans notre cas, nous avons basé sur : la température, l'humidité et la fuite de gaz. Dans le cas où l'un de ces facteurs est détecté hors sa marge de bon fonctionnement, la station de surveillance émet des alertes via une connexion sans fil (Bluetooth) vers la lunette intelligente.

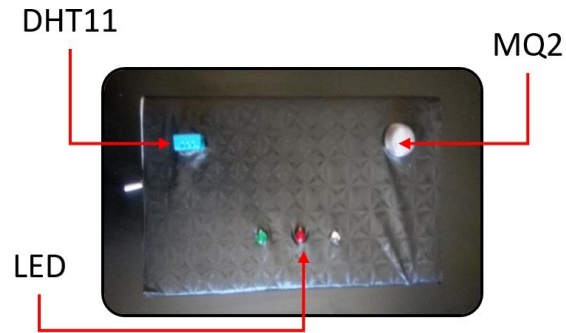


Figure III 14 : Station de surveillance

3.4.2.2 Etape 1 : Schéma électrique

En utilisant le logiciel fritzing, nous obtenons le schéma suivant :

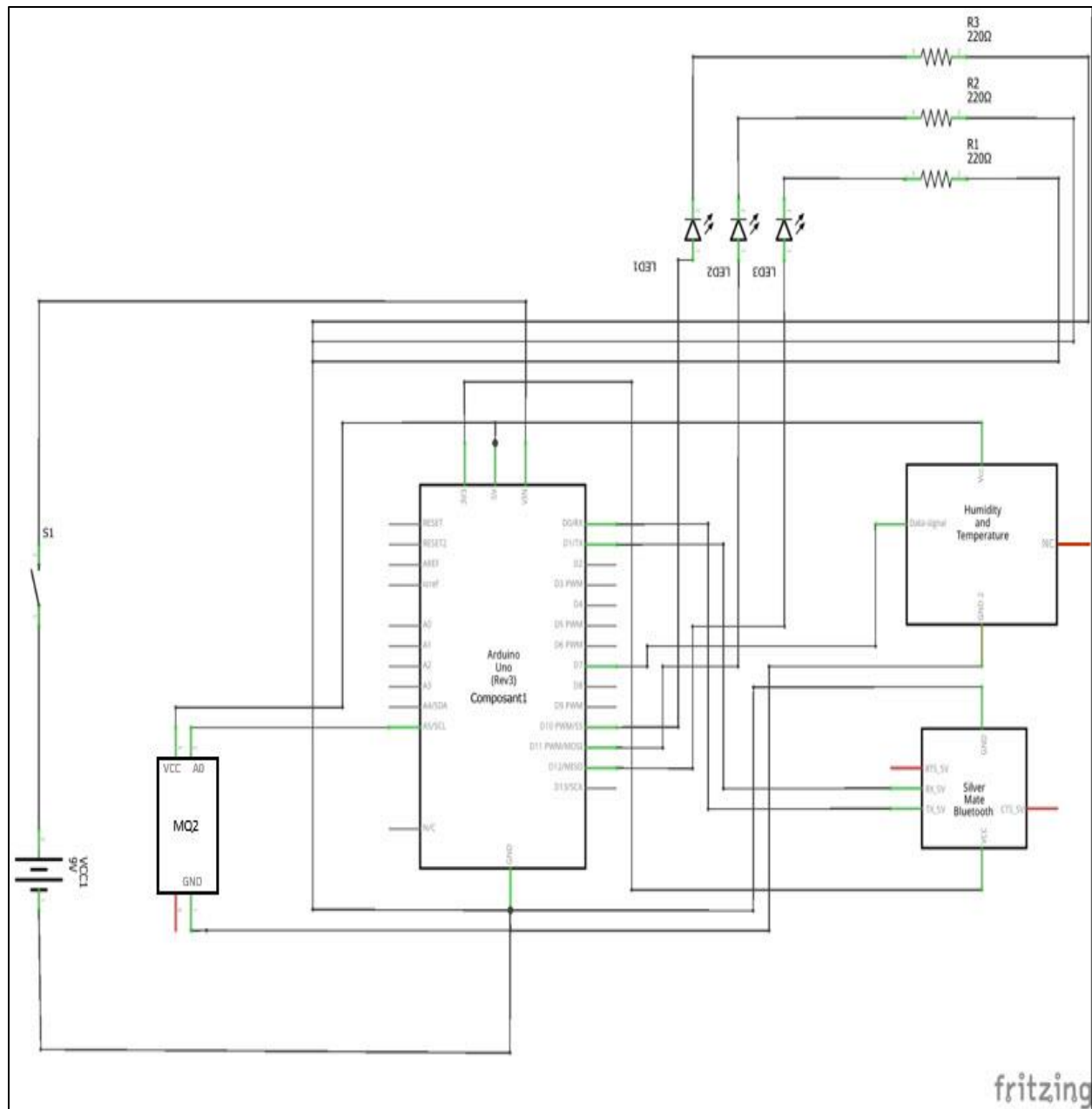


Figure III 15 : Schéma électrique de la station de surveillance

3.4.2.3 Etape 2 : Programmation

Pour l'initialisation des paramètres ainsi que la définition des entrées/sorties, nous avons :

```
#include <TroykaDHT.h>
DHT dht(7, DHT11);
int whiteLed = 12;
int greenLed = 11;
int redLed = 10;
int smokeA0 = A5;
void setup()
{
  Serial.begin(115200);
  dht.begin();
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(whiteLed, OUTPUT);
  pinMode(smokeA0, INPUT);
}
```

Voici de plus la boucle comprenant tous les cas de détection possibles. Comme nous avons trois facteurs à surveiller, nous avons alors 2^3 cas possibles :

```
void loop()
{
  dht.read();
  int analogSensor = analogRead(smokeA0);
  switch(dht.getState()) {
    case DHT_OK:
      if ((dht.getTemperatureC()>23.5) && (dht.getHumidity())>80) && (analogSensor>400)) {
        digitalWrite(redLed, HIGH);
        digitalWrite(greenLed, HIGH);
        digitalWrite(whiteLed, HIGH);
        Serial.print("T=");
        Serial.print(dht.getTemperatureC());
        Serial.println("C");
        Serial.print("Hm=");
        Serial.print(dht.getHumidity());
        Serial.println("%");
        Serial.print("Gaz leak");
      }
  }
}
```

```
else if ((dht.getHumidity()>80)&&(analogSensor>400)){
    digitalWrite(redLed, HIGH);
    digitalWrite(whiteLed,HIGH);
    digitalWrite(greenLed,LOW);
    Serial.print("Hm=");
    Serial.print(dht.getHumidity());
    Serial.println("%");
    Serial.print("Gaz leak");
}
```

```
else if ((dht.getTemperatureC()>23.5)&&(dht.getHumidity()>80)){
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, HIGH);
    digitalWrite(whiteLed,LOW);
    Serial.print("T=");
    Serial.print(dht.getTemperatureC());
    Serial.println("C");
    Serial.print("Hm=");
    Serial.print(dht.getHumidity());
    Serial.println("%");
}
```

```
else if ((dht.getTemperatureC()>23.5)&&(analogSensor>400)){
    digitalWrite(greenLed, HIGH);
    digitalWrite(whiteLed,HIGH);
    digitalWrite(redLed,LOW);
    Serial.print("T=");
    Serial.print(dht.getTemperatureC());
    Serial.println("C");
    Serial.print("Gaz leak");
}
```



```
else if ((dht.getTemperatureC()>23.5)) {
    digitalWrite(greenLed, HIGH);
    digitalWrite(redLed, LOW);
    digitalWrite(whiteLed, LOW);
    Serial.print("T=");
    Serial.print(dht.getTemperatureC());
    Serial.println("C");
}
```

```
else if ((dht.getHumidity()>80)) {
    digitalWrite(redLed, HIGH);
    digitalWrite(whiteLed, LOW);
    digitalWrite(greenLed, LOW);
    Serial.print("Hm=");
    Serial.print(dht.getHumidity());
    Serial.println("%");
}
```

```
else if ((analogSensor>400)) {
    digitalWrite(whiteLed, HIGH);
    digitalWrite(greenLed, LOW);
    digitalWrite(redLed, LOW);
    Serial.print("Gaz leak");
}
```

```
else
{
    digitalWrite(greenLed, LOW);
    digitalWrite(redLed, LOW);
    digitalWrite(whiteLed, LOW);
}
```

3.4.2.3 Etape 3 : Montage final

En respectant le montage électrique, nous arrivons à la phase finale :

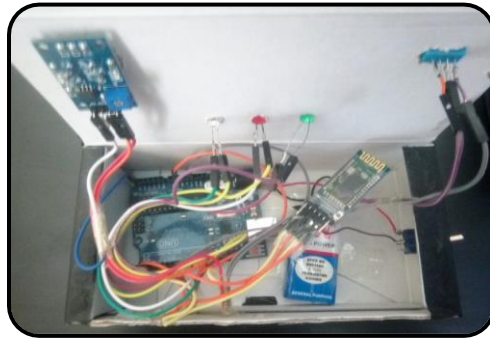


Figure III 16 : Vue interne de la station de surveillance

3.4.3 Lecteur des consignes de sécurité (LCS) :

3.4.3.1 Description du prototype :

Le lecteur de consignes de sécurité est un accessoire trop nécessaire pour l'employé au sein de l'entreprise. Ce dispositif permet l'émission des consignes de sécurité aux lunettes intelligentes tout en passant ce lecteur par les puces installées dans les stations concernées, sans besoin de recourir aux fiches techniques qui sont parfois inaccessibles. A la réalisation de ce modèle, nous avons équipé la carte Arduino par un module RFID qui sert, à travers un champ magnétique, la lecture des informations enregistrées sur les puces



Figure III 17 : Lecteur des consignes de sécurité (LCS)

3.4.3.2 Etape 1 : Schéma électrique

Nous avons réalisé ce modèle en suivant le schéma électrique présenté sur la figure III 17 :

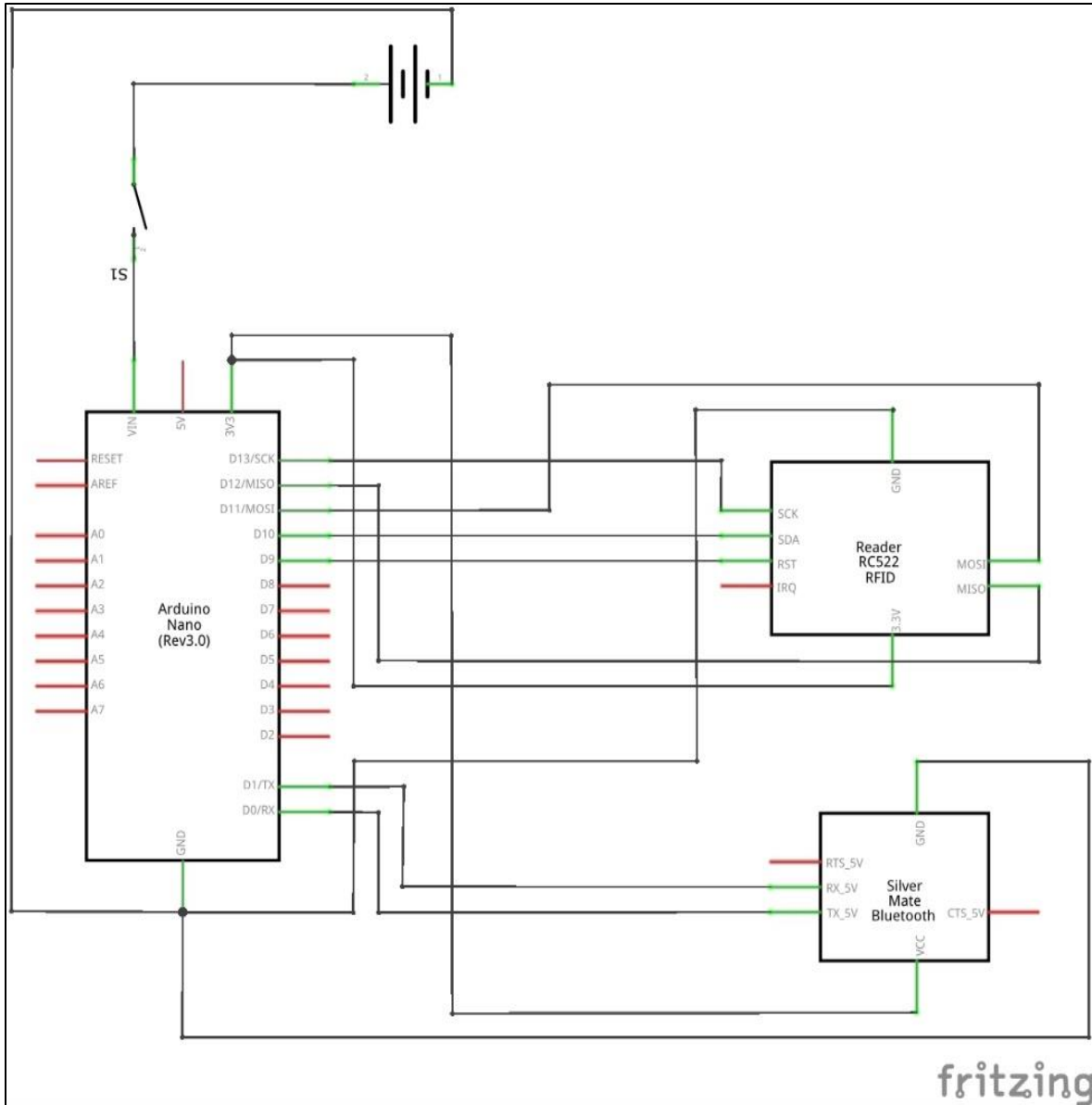


Figure III 18 : Schéma électrique du lecteur des consignes de sécurité (LCS)

3.4.3.3 Etape 2 : Programmation

Pour initialiser les paramètres de la RFID, nous devons déclarer les instructions suivantes :

```

#include <MFRC522.h>
#include <SPI.h>
#define SDAPIN 10
#define RESETPIN 8
byte FoundTag;
byte ReadTag;
byte TagData[MAX_LEN];
byte TagSerialNumber[5];
byte GoodTagSerialNumber_1[5] = {0xBA, 0x1A, 0x2E, 0x49};
byte GoodTagSerialNumber_2[5] = {0xAA, 0x19, 0x2E, 0x49};
byte GoodTagSerialNumber_3[5] = {0x55, 0x3D, 0x2B, 0xD9};
byte GoodTagSerialNumber_4[5] = {0xE3, 0xEF, 0xD5, 0x73};
MFRC522 nfc(SDAPIN, RESETPIN);

void setup() {
  SPI.begin();
  Serial.begin(115200);
  nfc.begin();
  byte version = nfc.getFirmwareVersion();
  if (! version) {
    Serial.print("Didn't find RC522 board.");
    while(1);
  }
}

```

Le Lecteur des consignes de sécurité (LCS) déchiffre les codes stockés dans les puces. Les instructions déclarées ci-dessous explique les codes ainsi que les messages à émettre en cas de détection.

```

void loop() {
  FoundTag = nfc.requestTag(MF1_REQIDL, TagData);

  if (FoundTag == MI_OK) {
    delay(200);
    ReadTag = nfc.antiCollision(TagData);
    memcpy(TagSerialNumber, TagData, 4);
  }
}

```

```

for(int i=0; i < 4; i++){
  if (GoodTagSerialNumber_1[i] != TagSerialNumber[i]) {
    break;
  }
  if (i == 3) {
    Serial.println(" Risque ");
    Serial.println(" gas ");
    Serial.println("inflammable");
  }
}

```

```

for(int i=0; i < 4; i++){
if (GoodTagSerialNumber_2[i] != TagSerialNumber[i]) {
break;
}
if (i == 3) {
Serial.println(" Danger ");
Serial.println(" choc ");
Serial.println("electrique");
}
}

```

```

for(int i=0; i < 4; i++){
if (GoodTagSerialNumber_3[i] != TagSerialNumber[i]) {
break;
}
if (i == 3) {
Serial.println("Station A");
Serial.println("utiliser");
Serial.println("vos casque");
}
}

```

```

for(int i=0; i < 4; i++){
if (GoodTagSerialNumber_4[i] != TagSerialNumber[i]) {
break;
}
if (i == 3) {
Serial.println("Attention");
Serial.println("aux cables");
Serial.println(" nues ");
}
}
}
delay(4000);
}

```

3.4.3.4 Etape 3 : Montage final

Vu à la nécessité de la mobilité, nous avons créé un support très léger avec de petites dimensions.



Figure III 19 : Vue interne du lecteur des consignes de sécurité (LCS)

3.4.4 Application Android :

3.4.4.1 Description de l'application :

Comme quatrième accessoire, nous avons développé une application présentée dans le deuxième chapitre. Cette application nous permet de :

- L'entretien à distance ;
- La maintenance ;
- La communication interne ;
- La surveillance ...

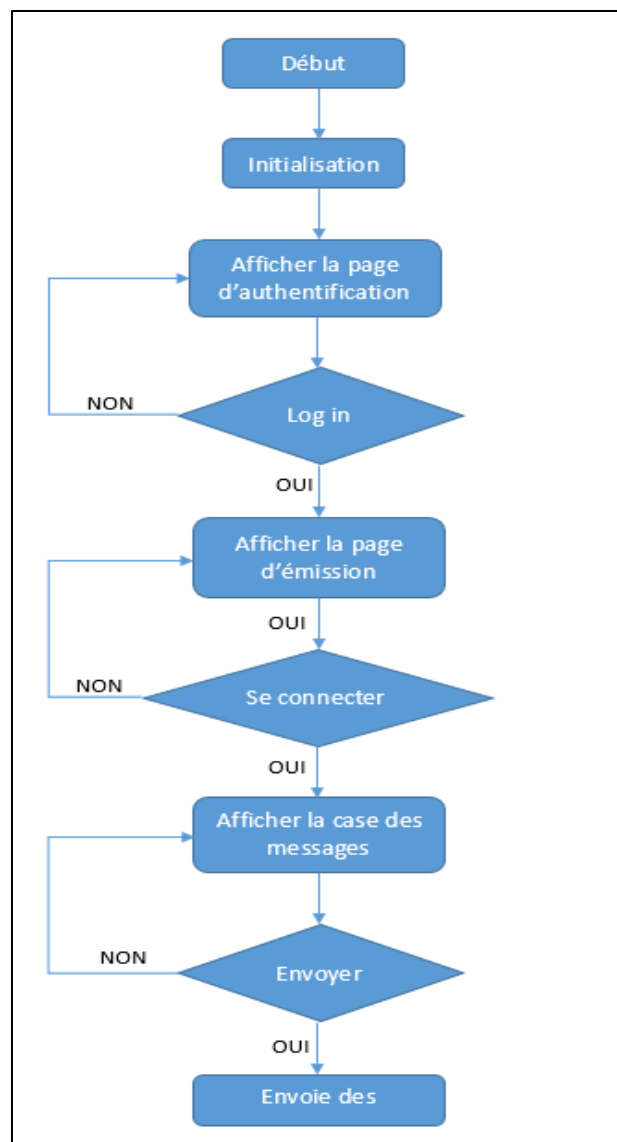


Figure III 20 : Organigramme de l'application Android développée

3.5 Partie résultats et interprétations :

Jusqu'ici, nous avons illustré la conception des lunettes intelligentes ainsi que leurs accessoires. Ce qu'il reste encore, c'est d'interpréter les résultats obtenus, ce qui est mentionné par la suite :

3.5.1 Lunette intelligente / voltmètre :

Après avoir réalisé les deux modèles, on peut librement mesurer le voltage d'un équipement donné en plaçant l'outil de mesure n'importe où, les valeurs mesurées seront affichées à l'écran des lunettes connectées. Les figures III 20, III 21 et III 22 démontrent nos hypothèses.

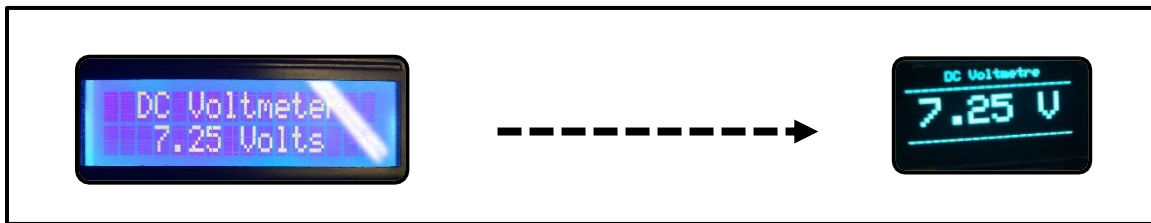


Figure III 21 : Résultat de l'essai N°1 avec le voltmètre

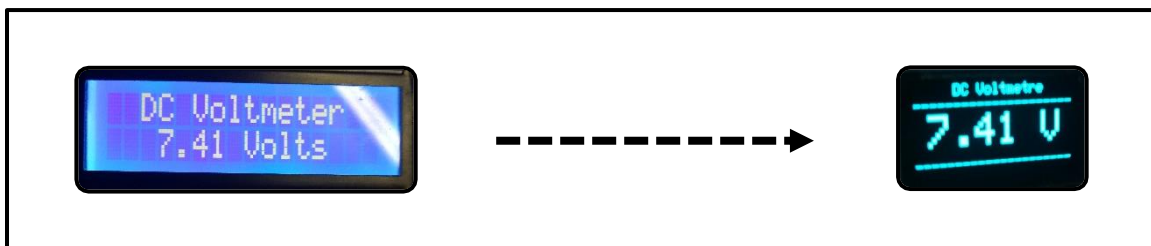


Figure III 22 : Résultat de l'essai N°2 avec le voltmètre

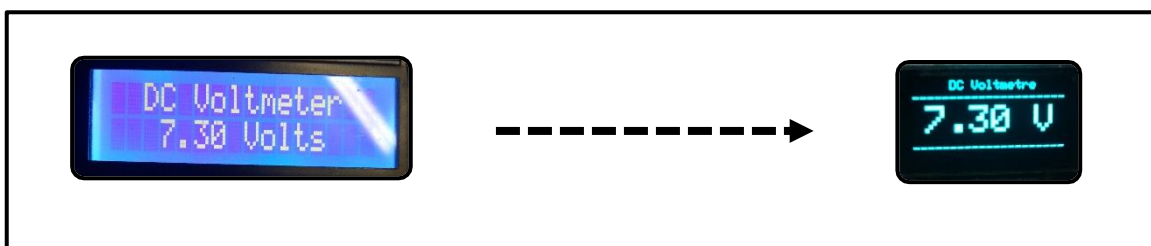


Figure III 23 : Résultat de l'essai N°3 avec le voltmètre

Remarque :

La figure III 20 illustre que la valeur mesurée par le voltmètre vaut 7.25 volts, la figure III 21 montre la valeur 7.41 volts et dans la figure III 22, le voltmètre nous donne la valeur est 7.30 volts. Les trois valeurs sont affichées, l'utilisateur peut clairement lire l'affichage projeté, ce qui explique le confort de lecture et la sécurité de mesure.

3.5.2 Lunette intelligente / station de surveillance :

Dans la station de surveillance, il existe trois LED où chacun mentionne une situation spécifique (le vert pour indiquer une température élevée, le rouge indique une perturbation dans l'humidité et le LED violet pour la fuite de gaz). Dans la figure suivante, nous constatons que le serveur détecte une température élevée ainsi qu'une fuite de gaz.

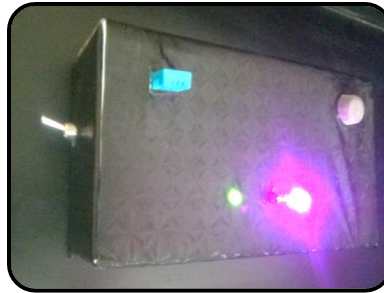


Figure III 24 : Détection du gaz et de température

Puisque nous avons huit cas possibles, nous allons présenter quelques-uns :

Cas N° 1 :

Nous pouvons obtenir cette situation où les trois facteurs sont détectés :

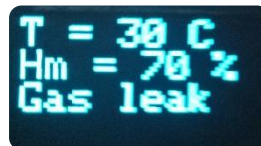


Figure III 25 : Résultat de l'essai N°1 avec la station de surveillance

La lunette indique que la température de la station surveillée est de 30 °C avec une humidité de 70 % ainsi qu'une fuite de gaz.

Cas N° 2 :

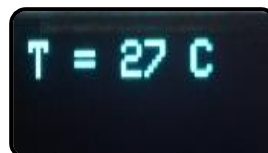


Figure III 26 : Résultat de l'essai N°2 avec la station de surveillance

Cas N°3 :



Figure III 27 : Résultat de l'essai N°3 avec la station de surveillance

Cas N°4 :



Figure III 28 : Résultat de l'essai N°4 avec la station de surveillance

Les trois facteurs sont détectés séparément où la température dans le deuxième cas vaut 27 °C, l'humidité dans le troisième cas vaut 90 % et une fuite de gaz dans le quatrième cas.

3.5.3 Lunette intelligente / lecteur des consignes de sécurité (LCS) :

En traversant un champ magnétique spécifique ou en passant le détecteur par les puces placés sur les machines, nous recevons des messages tels que :

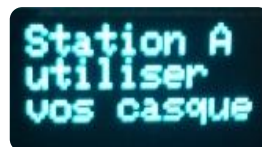


Figure III 29 : Résultat de l'essai N°1 avec le lecteur des consignes de sécurité

La figure précédente indique que l'utilisateur doit porter ses casques pour accéder à la station concernée.

Nous pouvons confronter d'autres cas aussi :

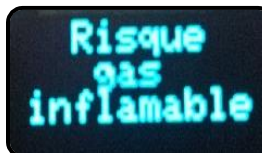


Figure III 30 : Résultat de l'essai N°2 avec le lecteur des consignes de sécurité

Cette alerte est déclenchée lorsqu'un employé pénètre une zone de danger. Comme ça, on puisse avertir notre employé à ne pas utiliser toute source de chaleur pour éviter la possibilité d'explosion ou d'inflammation.

N.B : la majorité des employés ignorent les consignes de sécurité, cette méthode les incite alors à suivre les consignes de manière innovante.

Ce cas est également possible :



Figure III 31 : Résultat de l'essai N°3 avec le lecteur des consignes de sécurité

Ce message est généralement transmis avant l'intervention aux armoires électriques, il met en conscience de l'employé la nécessité d'intervenir avec prudence.

Un autre cas possible :



Figure III 32 : Résultat de l'essai N°4 avec le lecteur des consignes de sécurité

Nous pouvons de plus trouver des situations dans laquelle on tire l'attention de l'employé aux câbles nus.

3.5.4 Lunette intelligente / application Android :

Lorsque l'application est lancée, nous confrontons une interface dans laquelle nous devons déclarer le nom d'utilisateur ainsi que son mot de passe :

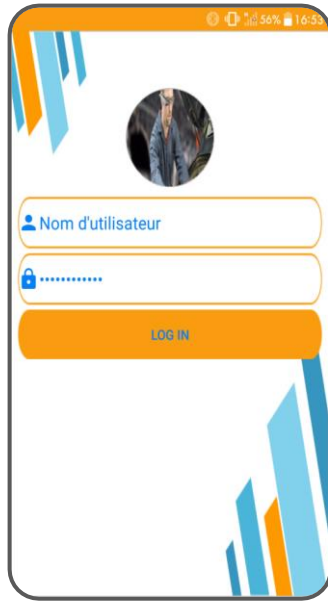


Figure III 33 : Interface d'authentification

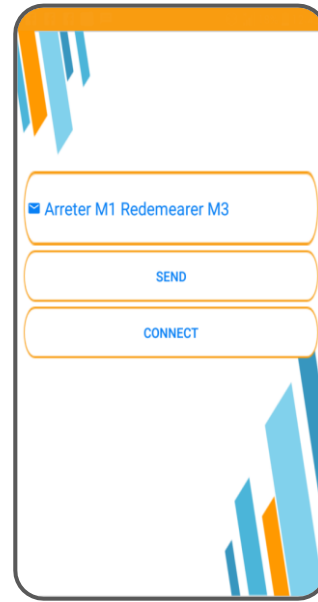


Figure III 34 : Interface de transmission

Si les informations saisies sont prédéfinies dans la base de données de l'application, nous passons à la deuxième interface où :

- Nous pouvons choisir la lunette à laquelle nous voulons connecter l'application ;
- Nous écrivons le message à envoyer.

Comme par exemple :

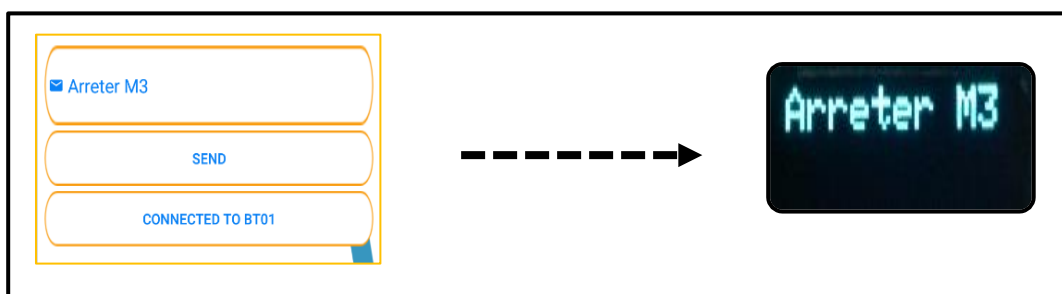


Figure III 35 : Résultat de l'essai N°1 avec l'application

Dans la figure précédente, nous avons envoyé un message à un employé donné contenant l'ordre d'arrêter la machine 3.

Voici un autre essai :

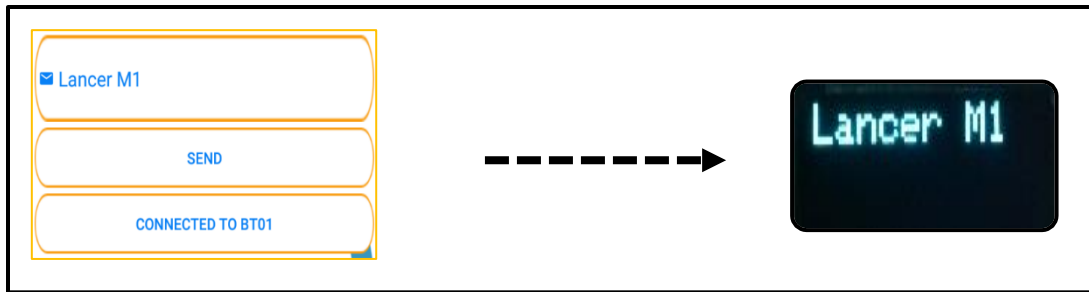


Figure III 36 : Résultat de l'essai N°2 avec l'application

Cet exemple explique le cas où un manager demande de son employé de lancer la machine 1. Cet ordre pourrait effectuer à distance sans besoin d'être sur terrain.

3.5.5 Résumé des résultats :

En achevant ces résultats, nous pouvons répondre à la problématique en confirmant les hypothèses suivantes :

- Développer une lunette intelligente pour renforcer le confort et la sécurité industrielle ;
- Réaliser un voltmètre et réussir la communication sans fil ;
- Exiger aux employés la lecture des consignes de sécurité ;
- Améliorer la surveillance en mettant en places des systèmes intelligents ;
- Exploiter les technologies existantes ;
- Augmenter la productivité des employés vu à l'innovation des systèmes ;
- Dispatcher l'information en temps réel ;
- Faciliter la communication interne ;

3.6 Conclusion :

Dans ce chapitre nous avons illustré la réalisation des lunettes intelligentes en passant par des différentes étapes. Nous avons de plus présenté les divers accessoires compatibles afin de répondre à la problématique et de justifier les hypothèses supposées auparavant.

Nous avons montré les interactions entre les fonctionnalités, la simulation des schémas électriques, la programmation des microcontrôleurs ainsi que les résultats obtenus et leurs interprétations.

Les lunettes intelligentes peuvent apporter beaucoup d'avantages en terme de confort, de sécurité, de productivité, de rentabilité et de fiabilité.

Conclusion générale

Conclusion générale

Les lunettes intelligentes communément dites Smart Glasses, sont aujourd'hui parmi les moyens les plus efficaces et les plus utilisées, dans l'industrie 4.0, pour servir la numérisation, la communication, et les réseaux industriels. Cette technologie est trop récente qu'elle n'est exploitée qu'en pays développés (Japan, Allemagne, Etat Unis ...).

L'étude abordé dans ce projet consiste à développer des lunettes intelligentes pour servir le confort d'utilisation et la sécurité humaine dans le secteur industriel. Nous visons à résoudre les problèmes, de l'industrie algérienne, que nous avons confronté tout au long de nos stages et durant notre parcours universitaire, tout en exploitant les technologies disponibles (Arduino, Android, Bluetooth ...).

Pour atteindre ces objectifs, nous avons développé, en premier temps, une application Android pour optimiser le flux d'information, une station pour la surveillance intelligente, un lecteur des consignes de sécurité et un voltmètre à connexion Bluetooth pour transmettre les mesures sans fils. Ces trois derniers systèmes fonctionnent à base de carte Arduino où les informations traitées sont envoyées aux lunettes intelligentes via connexion Bluetooth.

Nous avons exposé dans un premier chapitre quelques notions telles que les risques, les dangers, la sécurité, la sureté de fonctionnement et la maintenance Nous avons consacré un deuxième chapitre pour bien définir les composants électroniques et les outils de développement (Arduino, Android ...). Dans le troisième chapitre, nous avons illustré la création de nos dispositifs dès la conception à la réalisation.

L'implémentation de notre lunette intelligente peut éviter beaucoup de procédés pénibles au sein de l'entreprise, nous citons par exemple : l'utilisation classique des outils de mesure, le contrôle des stations de travail, l'accès aux fiches techniques et aux consignes de sécurité et le manque de flux d'information en temps réel.

A la fin du projet, nous envisageons certaines perspectives :

- Intégrer des caméras pour la reconnaissance des formes ;
- Elargir le réseau des lunettes en mettant en place une connexion wifi ;
- Optimiser le design en un modèle beaucoup plus léger ;
- Augmenter l'autonomie en installant des batteries rechargeables ;
- Adopter la réalité augmentée pour les services de télémaintenance.

Bibliographie

Bibliographie

- [1] A.Villemeur, Sûreté de fonctionnement des système industriels, vol. 67, Eyrolles, 1988.
- [2] Abbou R, Contribution à la mise en œuvre d'une maintenance centralisée : conception et optimisation d'un atelier de maintenance. Thèse de Doctorat, Université Joseph Fourier, Grenoble, France. 2003.
- [3] Alhouaij Ahmad, Contribution à l'optimisation de la maintenance dans un contexte distribué. Thèse de Doctorat, Institut National Polytechnique de Grenoble - INPG, 2010.
- [4] Astalaseven, Eskimon et olyte. Arduino pour bien commencer en électronique et en programmation. 2012.
- [5] bdc*. L'industrie 4.0 : la nouvelle révolution industrielle, les fabricants canadiens sont-ils prêts ?. 2017.
- [6] BECK Elise. 2006, Approche multirisques en milieu urbain, thèse de Doctorat, Université Louis pasteur, Strasbourg, p 17.
- [7] Boudjedir Imen. Mémoire master sous le titre UN SYSTEME EMBARQUE POUR LA DETECTION DES GAZ DANGEREUX chapitre 2 page 22, 2017.
- [8] C. Pagetti, « Module de sûreté de fonctionnement, space.univtlemcen.dz/bitstream/112/9395/1/cours.pdf, » 2012.
- [9] CHAFAA BELAID ZAKARYA & CHAFA BELAID HAMZA. Mémoire Master (Etude et réalisation d'un système de communication par Bluetooth pour un compteur électrique avancé). Chapitre2 page 45, année 2016-2017.
- [10] CHALINE C et J DUBOIS-MAURY, 1994. La ville et ses dangers. Paris, Masson, p 244.
- [11] Collectif AFIS, «DÉCOUVRIR ET COMPRENDRE L'INGÉNIERIE SYSTÈME, chapitre 11: l'ingénierie système : définition,» *Collection AFIS*, n° %11005, pp. 189-190, 2012.
- [12] Cuny, X. *La fonction sémique dans le travail. L'élaboration et l'utilisation des systèmes non verbaux chez l'adulte*. Doctorat d'État, Université de Bordeaux II, Bordeaux. 1981.
- [13] Deloux E., Castanier B., and Bérenguer C., Optimisation de la politique de maintenance pour un système à dégradation graduelle stressé, 7ème Congrès international pluridisciplinaire Qualité de Sûreté de Fonctionnement, 20, 21 et 22 mars 2007, volume 1, pages 142–149, Tanger (Maroc), 2007.

- [14] Devarun Ghosh, Sandip Roy, Maintenance optimization using probabilistic cost benefit analysis. *Journal of Loss Prevention in the Process Industries* 2009; 22(4): 403-407.
- [15] E. Deloux, B. Castanier, and C. Bérenguer. Construction d'un schéma de maintenance.
- [16] Erik Bartmann. Le grand livre d'Arduino [chapitre 1 p6] 3^{ème} édition.
- [17] Farhat beghil. réseau locaux industriels. Licence automatique, université d'Annaba, Annaba. 2007.
- [18] FEC électronique datasheet.
- [19] Fiche DRM « le risque industriel » DIMENC-2014.
- [20] Gartner Inc Gartner Says Smartphone Sales Grew 46.5 Percent in Second Quarter of 2016 and Eceeded Featur Phone Sales for First Time Gartner. [En ligne]. Disponible sur : <http://www.gartner.com/newsroom/id/2573415>.
- [21] GEORGE.P, VERGER.F (Dictionnaire de la géographie, presse universitaire de France, Paris, 1970, 500p).
- [22] H. Procaccia et J. Procaccia, Genèse du Big Bang à l'art pariétal, 2. Saint Denis : Connaissances et savoirs, Éd., Sciences de l'univers, 2015.
- [23] Hafaifa Ahmed, Attia Daoudi and Kouider Laroussi, Application of fuzzy diagnosis in fault detection and isolation to the compression system protection. *Control and Intelligent Systems ACTA Press*, July 2011, vol. 39, no. 3, pp. 151-158.
- [24] I.W Burr. Statistical quality control methods. Marcel Dekker, 1976.
- [25] J. Wiley et Sons, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 2015.
- [26] J.-C. Laprie, Guide de la sûreté de fonctionnement, 2ème édition éd., Cépaduès, Éd., Broché, 1996.
- [27] KRAMA Abdel basset, GOUGUI Abdelmoumen, le : 08/06/2015, Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde, Master en Electrotechnique Industrielle, UNIVERSITE KASDI MERBAH OUARGLA.
- [28] Leplat, J. About implementation of safety rules. *Safety Science*, 29, 189-204. 1998.
- [29] Leplat, J. Causes et risques dans l'analyse d'accidents. *Revue roumaine de psychologie*, 39, 9-24. 1995.

- [30] Leplat, J. Questions autour de la notion de risque. In R. D. Kouabenan & M. Dubois (Eds.), *Les risques professionnels : évolution des approches, nouvelles perspectives* (pp. 37-52). Toulouse : Octarès. 2003.
- [31] Mark Murphy Beginning Android 3.sl: Apress 2011.
- [32] Mayen, P. (2002). Formation et prescription : une réflexion de didactique professionnelle. In J.-M. Évesque, A.-M. Gautier, C. Revest, Y. Schwartz, & J.-M. Vayssière (Eds.), *Actes du XXXVIIe Congrès de la SELF : Les évolutions de la prescription* (pp. 226-232). Aix-en Provence, septembre 2002.
- [33] Monchy, F. Maintenance Méthodes et Organisations (2e édition). Paris: Dunod (2003).
- [34] NF, Norme Française, *Norme NF X60-500 Terminologie relative à la fiabilité, maintenabilité, disponibilité*, 1988.
- [35] R. Dekker, Applications of maintenance optimization models : a review and analysis, *Reliability Engineering and System Safety* 51(1996), 229–240.
- [36] RACHIDI, A., DAKKAK, B., TALBI, A., & KHATORY, A. “la réalité augmentée au service de Télémaintenance et de E-maintenance industrielle”, 2015.
- [37] Rapport RFID Télécom Sud Paris 2013.
- [38] Samurçay, R., Volkoff, S., & Savoyant, A. (1999). Introduction. *Formation emploi*, 67, 5-7.
- [39] Sarayeddine, K. (2015). Regard sur le futur. Les lunettes intelligentes (smart glasses). *Photoniques*, (77), 24-26.
- [40] Schneider Electric, «La Sûreté de Fonctionnement - revue InterSections oct 2004,» *InterSections*, vol. 1, n° % 11, pp. 1-12, Nov 2004.
- [41] Sherwin, D. (2000). A review of overall models for maintenance management. *Journal of Quality in Maintenance Engineering*, 6(3), 138–164.
- [42] <https://www.realite-virtuelle.com/lunettes-connectees-15-modeles>.
- [43] www.elektronique.fr/logiciels/proteus.php.
- [44] <https://www.arduino.cc/>.
- [45] <https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/>.
- [46] <http://www.tutos-android.com/introduction-a-android-studio>.
- [47] https://my.solidworks.com/solidworks/guide/SOLIDWORKS_Introduction_FR.
- [48] <https://docs.particle.io/assets/datashets/electronsensorkit/MQ-2>.

- [49] <https://fr.hobbytronics.co.uk/mq7-carbon-monoxide-sensor>.
- [50] https://www.hdevbot.com/dataset/dataset_MQ-2.s.
- [51] <https://digital-solutions.konicaminolta.fr/des-lunettes-connectees-pour-lindustrie-4-0>.
- [52] <https://ellcie-healthy.com>.
- [53] www.aniwaa.fr.
- [54] [vue-glasses.com/ ?lang=fr](http://vue-glasses.com/?lang=fr).
- [55] <https://www.industrie-techno.com/article/bracelets-connectes-visseuses-autonomes-cobots-lunettes-intelligentes-visite-d-une-usine-4-0.49738>.

Résumé :

Les lunettes intelligentes sont vues comme étant des dispositifs électro-optiques qui servent à exécuter une application ou à projeter une information donnée sur les yeux d'utilisateur sans perdre la vision extérieure.

Ce mémoire présente l'exploitation des lunettes intelligentes au domaine de l'industrie pour servir le confort et la sécurité. Dans ce cadre, nous avons développé certaines fonctionnalités en rendant ces dispositifs compatibles avec de multiples accessoires tels que : outils de mesure (Voltmètre ...), appareils de sécurité et de contrôle, lecteurs des consignes de sécurité et les applications Android.

Les résultats obtenus démontrent la fiabilité des lunettes intelligentes dans le secteur industriel : le confort d'entretien, la sécurité d'utilisation, la rentabilité et la productivité.

Mots clés : Lunettes intelligentes, Sécurité, Voltmètre, Android, Arduino.

Abstract :

Smart glasses are electro-optical devices that can execute application or project a given information on the user's eyes without losing the external vision.

This study exposes the use of smart glasses in the field of industry to serve comfort and safety. In this context, we have developed some features by making these devices compatible with multiple accessories such as: measurement tools (Voltmeter ...), control and security devices, security instructions reader, Android applications.

The results obtained demonstrate the reliability of smart glasses in the industrial sector: the comfort of maintenance, the safety of use, the profitability and the productivity.

Keywords: Smart Glasses, Security, Voltmeter, Android, Arduino.

ملخص :

النظارات الذكية هي أجهزة كهروضوئية تُستخدم لتنفيذ تطبيق ما أو لعرض معلومات معينة على عين المستخدم دون فقد الرؤية الخارجية.

تقدم هذه المذكرة استخدام النظارات الذكية في مجال الصناعة لتوفير الراحة والأمان. في هذا السياق، قمنا بتطوير بعض الميزات من خلال جعل هذه الأجهزة متوافقة مع العديد من الملحقات مثل: أدوات القياس (مقياس التوتر...), أجهزة المراقبة والأمان, كاشفات التعليمات الأمنية وتطبيقات الأندرويد.

توضح النتائج التي تم الحصول عليها موثوقية استخدام النظارات الذكية في القطاع الصناعي من أجل توفير راحة الصيانة, سلامة الاستخدام, الربحية والإنتاجية.

الكلمات المفتاحية : النظارات الذكية، الأمان، مقياس التوتر، أندرويد، اردوينو.