

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE**



**UNIVERSITE MOHAMED BOUDIAF - M'SILA**

**FACULTE DE TECHNOLOGIE**

**DEPARTEMENT DE GENIE ELECTRIQUE**

**MEMOIRE DE FIN D'ETUDE EN VUE DE L'OBTENTION DU DIPLOME DE  
MASTER EN AUTOMATIQUE**

**SPECIALITE: AUTOMATIQUE**

**THEME**

---

**"Réalisation d'un IMU (Inertial Measurement Unit)"**

---

**Proposé et dirigé par :**  
**Mr. KHODJA Mohammed Abdallah**

**Présenté par**  
**RAHMANI Nadir**

**Année Universitaire : 2015/2016**

**N° d'ordre : AUT\_195**

## Remerciements

---

*Je remercie Dieu, le tout puissant, pour m'avoir donné, le courage, la patience, la volonté et la force nécessaire, pour affronter toutes les difficultés et les obstacles, qui se sont hissés au travers de mon chemin, durant toutes mes années d'études.*

*J'adresse mes sincères remerciements à tous mes enseignants de L'université de M'sila, notamment à mon promoteur Mr. KHODJA Mohammed Abdallah, d'avoir proposé le sujet sur lequel j'ai travaillé, et qui a assuré la direction et l'encadrement du travail présenté dans ce mémoire.*

*Je tiens aussi à exprimer mes remerciements à Mekki AbdElmounim (mon collègue), d'avoir bien voulu de me faire profiter pleinement de ces compétences scientifiques et de ses idées pour l'accomplissement de ce travail.*

*Je tiens notamment, à adresser l'expression de ma sincère reconnaissance aux membres du jury d'examen, pour l'intérêt qu'ils ont porté à ce travail.*

*Cette page ne serait être complète sans remercier mes meilleurs collègues et frères : benzahia ayoub ,hassani azzedine , sans leurs soutiens je n'aurais sûrement pas pu mener à bien ce mémoire. Enfin, mes remerciements vont à tous ceux qui m'ont soutenu de près ou de loin à réussir ce travail.*

*Mes vives salutations.*

## Dédicaces

---

Je dédie ce modeste travail

A ma très chère mère source de tendresse

A mon très cher père, qui m'encourage dans les instants délicats

A mes chers frères

A mes tantes et oncles

A toute ma famille

A tous mes amis

---

# Table des matières

---

<b>Table des matières</b> .....	I
<b>Liste des figures</b> .....	IV
<b>Liste des tableaux</b> .....	V
<b>Nomenclatures</b> .....	VI
<b>Introduction générale</b> .....	1

## Chapitre I

### Présentation de l'unité de mesure inertielle

I.1. Introduction .....	3
I.2. Centrales inertielle (IMU : Inertiel Measurement Unit) .....	3
I.3. Configurations système inertielle .....	4
I.4. Gyroscopes .....	5
I.4.1. Définition de gyroscope.....	5
I.4.2. Types de Gyroscope .....	6
I.4.2.1. Mécanique .....	6
I.4.2.2. Optique .....	6
I.4.3. MEMS Gyroscopes.....	6
I.4.4. Les Caractéristiques d'erreur de MEMS Gyro .....	7
I.4.4.1. Biais constant.....	7
I.4.4.2. Effets de la température .....	8
I.5. Accéléromètres linéaires.....	8
I.5.1. Définition de l'accéléromètre.....	8
I.5.2. Types d'Accéléromètre.....	9
I.5.2.1. Mécanique .....	9
I.5.2.2. État solide .....	9
I.5.3. MEMS Accéléromètres .....	9
I.5.4. Accéléromètre MEMS erreur Caractéristiques.....	10
I.5.4.1. Biais constant.....	10
I.5.4.2. Flicker bruit / Biais stabilité .....	10
I.5.4.3. Effets de la température .....	10
I.5.4.4. Erreurs d'étalonnage .....	10
I.6. Filtre de kalman .....	11
I.6.1. introduction.....	11
I.6.2. Définition.....	11
I.6.3. Orientation .....	13
I.6.4. Position .....	14
I.7. Conclusion .....	16

---

**Chapitre II****Arduino**

II.1.Introduction .....	17
II.2. Qu'est-ce qu'un microcontrôleur ? .....	17
II.3. À quoi sert-il ?.....	18
II.4.Structure d'un microcontrôleur .....	18
II.4.1. L'unité centrale (CPU) .....	19
II.4.2. Le bus de données .....	19
II.4.3. Les zones de mémoire .....	20
II.5. Arduino.....	21
II.5.1. Arduino est-il un microcontrôleur ?.....	21
II.5.2. Les différentes cartes Arduino .....	21
II.5.2.1. Arduino Uno .....	22
II.5.2.2. Arduino Mega 2560 .....	23
II.5.3. Arduino Nano.....	24
II.5.4. Structure de la carte Arduino .....	26
II.5.4.1. L'alimentation électrique .....	28
II.5-4-2- Les modes de communication.....	28
II-5-4-3- Le port USB .....	28
II-5.4.4. Les ports d'entrée ou de sortie (E/S) .....	29
II.5.4.5. Les langages de programmation C/C++ .....	30
II.6. Protocoles de communication .....	31
II.6.1. Le protocole I2C .....	31
II.6.1.1. Le support physique utilisé .....	31
II.6.1.2. Structure d'I2C.....	32
II.6.1.3. Communication d' I2C .....	33
II.6.1.4. La prise de contrôle du bus .....	33
II.6.2.Le protocole SPI .....	34
II.6.2.1. Aperçue de SPI .....	34
II.6.2.2. SPI Détails .....	34
II.6.2.3. Les données et les contrôles des lignes de la SPI .....	36
II.6.2.4. SPI Configuration .....	36
II.7. Conclusion.....	36

**Chapitre III****Exemple d'IMU dans matériel réel**

III.1. Introduction.....	39
III.2. Présentation.....	39
III.2.1. Qu'est-ce qu'un quadri-rotor ?.....	39
III.2.2. Comment le piloter ?.....	40
III.2.3. Applications possibles .....	40
III.3. Modèle dynamique.....	40

III.4. les téléphones intelligents .....	41
III.4.1. Système de coordonnées des capteurs .....	42
III.5. conclusion .....	43
<b>Conclusion générale et perspective.....</b>	<b>48</b>
<b>Références bibliographiques .....</b>	<b>49</b>

---

# Liste des figures

---

## Chapitre I

Figure .I.1. principe d'orientation et d'accélération linéaire.....	3
Figure .I.2. Une plate-forme stable IMU.....	3
Figure .I.3. Principe de fonctionnement de gyroscope.....	4
Figure .I.4. Flux d'information pour estimation d'orientation.....	14
Figure .I.5. Flux d'information pour estimation de la position.....	16

## CHAPITRE II

Figure .II.1. Microcontrôleur ATmega328.....	17
Figure .II.2. Schéma fonctionnel d'un microcontrôleur .....	19
Figure .II.3. Sur l'autoroute des données : « Prochain arrêt, mémoire .....	20
Figure .II.4. Ports d'entrée et de sortie .....	21
Figure .II.5. La carte Arduino Uno .....	22
Figure .II.6. La carte Arduino Mega 2560.....	23
Figure .II.7. La carte Arduino Nano.. .....	25
Figure .II.8. La carte Arduino Nano .....	26
Figure .II.9. Que trouve-t-on sur la carte Arduino ?.....	27
Figure .II.10. Entrées et sorties de la carte Arduino.....	27
Figure .II.11. Alimentation de la carte Arduino .....	28
Figure .II.12. Entrées et sorties numériques .....	29
Figure .II.13. Alimentation et entrées analogiques .....	29
Figure .II.14. Le compilateur sert de traducteur .....	30
Figure .II.15. Présentation de Communication Serial .....	32
Figure .II.16. l'adressage d'I2C .....	33
Figure .II.17. seul maître, esclave seule implémentation SPI.. .....	34
Figure .II.18. seul maître, esclaves multiples implémentations SPI. ....	35

## CHAPITRE III

Figure .III.1. structure générale d'un quadrirotor.....	39
Figure .III.2. Configuration du quadrotor.....	41
Figure .III.3. Capteurs sur un système Android basé sur Intel® Atom™.....	41
Figure III.4. Système de coordonnées de capteur d'un téléphone.....	43
Figure .III.5. Système de coordonnées de capteur d'une tablette.....	43

## Liste des tableaux

---

Table (II.1): Tableau II-1: La carte Arduino Uno.....	22
Table (II.2): Tableau II-2: La carte Arduino Uno.....	24
Table (II.3): Tableau II-3: La carte Arduino.....	25
Table (II.4): Table II-4 : SPI Modes.....	36
Tableau III.1. Types de capteurs pris en charge par la plate-forme Android.....	42

## Nomenclatures

---

IMU	:	Inertiel Measurement Unit
MEMS	:	Micro-Electro-Mechanical Systems
FOG	:	fibre optic gyroscope
RLG	:	Ring-Laser Gyroscope
CPU	:	central processing unit
EEPROM	:	Electrically Erasable Programmable Read-Only Memory
SRAM	:	Static random-access memory

# **Introduction Générale**

# Introduction Générale

---

Haute précision de navigation à inertie utilisant les capteurs inertiels coûteux est contenue à des applications de haute technologie comme les satellites, sondes, des missiles. Capteurs inertiels tels que les MEMS peuvent fournir position, la vitesse et l'accélération. [1] Compte tenu de la capacité de mesurer l'accélération du véhicule, il serait possible de calculer la variation de la vitesse et la position en réalisant des intégrations mathématiques successives de l'accélération par rapport au temps, afin de naviguer par rapport à notre cadre de référence inertiel, il est nécessaire de garder une trace de la direction dans laquelle les accéléromètres sont dirigés, le mouvement de rotation du corps par rapport au référentiel inertiel peut être détectée en utilisant des capteurs gyroscopiques qui sont utilisés pour déterminer l'orientation des accéléromètres à tout moment. Compte tenu de cette information, il est possible de résoudre les accélérations dans le cadre de référence avant que le processus d'intégration se déroule [2].

Dans ce mémoire, nous explorons des capteurs et une série de tutoriels pour vous donner une vue d'ensemble complète des capteurs et comment les utiliser. Ces types de capteurs avec des capacités pour mesurer l'accélération, l'inclinaison, vitesse angulaire, et d'autres phénomènes mécaniques sont utilisés dans différents dispositifs, les smartphones, les consoles de jeux ou des jouets.

Si un capteur accéléromètre est conçu pour mesurer l'accélération et de l'inclinaison, le capteur gyroscopique mesure la vitesse et l'orientation angulaire. Le capteur de IMU est un spécial conçu pour combiner les caractéristiques d'un accéléromètre et gyroscope afin d'afficher toutes les infos sur l'accélération, la position, l'orientation, la vitesse, etc. pour un robot.

L'unité de mesure inertielle UMI ou est un capteur qui héberge deux types de capteurs. Un capteur IMU peut être utilisé pour remplacer un capteur accéléromètre ou gyromètre, mais d'abord vous devez définir la tolérance pour les erreurs. Le plus grand inconvénient de ce capteur est l'erreur de mesure. Tous ces capteurs sont minuscules et pas cher que des composants pour un large éventail de mesures.

Les capteurs IMU sont l'un des types les plus inévitables de capteurs utilisés aujourd'hui dans toutes sortes de gadgets. Ils sont considérés dans les smartphones, vestimentaires, les contrôleurs de jeu, etc. Capteurs IMU nous aident à obtenir l'attitude d'un objet, attaché à la sonde dans l'espace tridimensionnel. Ces valeurs habituellement dans les angles, contribuent ainsi à déterminer son attitude. Ainsi, ils sont utilisés dans les téléphones intelligents pour détecter son orientation. Et aussi dans les gadgets portables comme la bande de carburant de Nike ou peu en forme, qui utilisent des capteurs pour suivre le mouvement IMU.

Arduino, et son récent synonyme Genuino1, sont des cartes matériellement libres sur lesquelles se trouve un microcontrôleur (d'architecture Atmel AVR comme par exemple l'Atmega328p). Les schémas de ces cartes sont publiés en licence libre, cependant, certains composants, comme le microcontrôleur par exemple, ne sont pas en licence libre.

Le microcontrôleur peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques - éclairage, chauffage...), le pilotage d'un robot, de l'informatique embarquée, etc [3].

Ce mémoire est composé de trois chapitres, est organisé de la manière suivante :

- Dans le premier chapitre, on définit IMU une technique de navigation, et sa fonctionnement et on a vue utilisation de l'accéléromètre et gyroscope et on termine par le rôle de filtre de kalman qui est nécessaire de le projet.
- Dans le deuxième chapitre nous avons parlé sur l'Arduino et ces types en général et son rôle dans la conception du quadrotor
- Dans le troisième chapitre nous avons présenté la modélisation du système dynamique de quadrotor et nous avons essayé de Synthèse une loi de commande PID à gains adaptatifs .

# Chapitre I

# Chapitre I

## Présentation de l'unité de mesure inertielle

---

### I.1. Introduction

La navigation inertielle est une technique de navigation autonome dans lequel les mesures fournies par accéléromètres et gyroscopes. Sont utilisés pour suivre la position et l'orientation d'un objet par rapport à un point de départ, l'orientation et la vitesse connue. Unités de mesure inertielle (IMU) contiennent généralement trois taux gyroscopes orthogonaux et trois accéléromètres orthogonaux, mesurer la vitesse angulaire et l'accélération linéaire respectivement. En traitant des signaux de ces dispositifs, il est possible de suivre la position et l'orientation d'un dispositif.

Navigation inertielle est utilisée dans une large gamme d'applications, y compris la navigation des avions, des missiles tactiques et stratégiques, engins spatiaux, sous-marins et les navires. Les progrès récents dans la construction de dispositifs MEMS ont permis de fabriquer des petits et légers systèmes de navigation inertielle. Ces progrès ont élargi la gamme des applications possibles pour inclure des domaines tels que la capture de mouvement humain et animal [4].

Un IMU fonctionne en détectant le taux actuel d'accélération, ainsi que cela change dans les attributs de rotation, y compris tangage, de roulis et de lacet. Ces données sont ensuite introduits dans un ordinateur qui calcule la vitesse et la position actuelle, compte tenu de la vitesse initiale et la position connue [5].

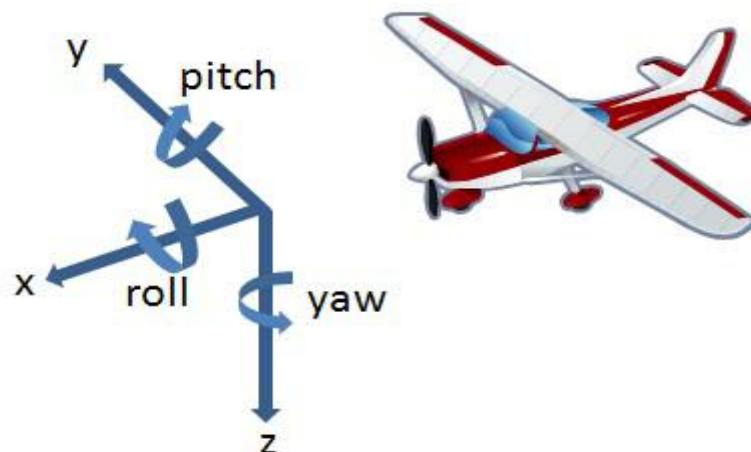


Figure .I.1. Principe d'orientation et d'accélération linéaire [5].

### I.2. Centrales inertielles (IMU : Inertial Measurement Unit)

Une centrale inertielle (IMU) est un système complet, composé au minimum de 3 accéléromètres et de 3 gyroscopes permettant de mesurer les composantes selon les 3 axes de l'accélération non gravitationnelle et de la vitesse instantanée de rotation du véhicule par rapport à un référentiel inertielle (qui est confondu avec le repère terrestre dans la plupart des cas). Les centrales inertielles sont des systèmes complexes et chers. Elles intègrent une électronique permettant de corriger les données capteurs : compensation de l'accélération au niveau de la mesure des gyroscopes,

auto-compensation en température, orthogonalisation des axes de mesures, etc. On distingue deux types principaux de centrales inertielles : les centrales strap-down et les centrales à plate-forme stabilisées [6].

### I.3. Configurations système inertiel

Presque tous IMU tombent dans l'une des deux catégories décrites ci-dessous. La différence entre les deux catégories est le cadre de référence dans lequel les taux des gyroscopes et des accéléromètres fonctionnent.

Tout au long de ce rapport, nous allons consulter le cadre de référence que le châssis et le cadre de référence dans lequel nous naviguons que le cadre global, comme le montre la figure 1 du système de navigation.

Dans les systèmes de type plate-forme stable, les capteurs inertiels sont montés sur une plate-forme qui est isolé de tout mouvement de rotation externe. En d'autres termes, la plate-forme est maintenue en alignement avec le cadre global, ceci est réalisé par le montage de la plate-forme à l'aide cardans (frames) qui permettent la liberté de la plate-forme dans les trois axes, comme le montre la figure 2. Les gyroscopes de plate-forme monté détecter toute rotation de la plate-forme. Ces signaux sont renvoyés à moteurs couples qui tournent les cardans afin d'annuler de telles rotations, donc garder la plate-forme alignée avec le cadre global.

Pour suivre l'orientation de l'appareil les angles entre cardans adjacentes peuvent être lus avec angle pick-offs. Pour calculer la position de l'appareil les signaux provenant de la plate-forme montée accéléromètres sont à double intégrée. On notera qu'il est nécessaire de soustraire l'accélération due à la gravité à partir du canal vertical avant d'effectuer l'intégration. L'algorithme de navigation inertielle de plate-forme stable est illustré sur la figure(3) [7].

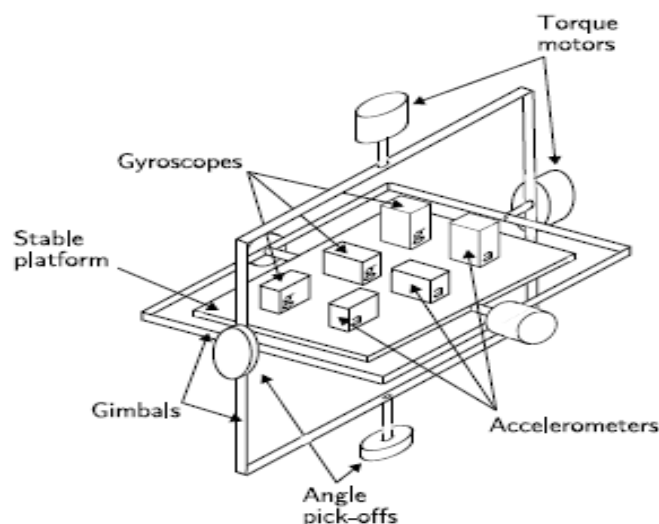


Figure .I.2. Une plate-forme stable IMU. [6]

## I.4. Gyroscopes

### I.4.1. Définition de gyroscope

Un gyroscope est un appareil permettant d'effectuer une mesure de la rotation absolue de son boîtier. On retrouve deux types les gyroscopes mécaniques et les gyroscopes à laser (fibres optiques). Il faut tenir compte de la dérive des mesures au cours du temps et effectuer régulièrement des recalages absolus (on ne les utilise pas seuls, mais en composants intégrés de centrales inertielle) [7].

Le gyroscope est défini comme étant un objet en rotation rigide, symétrique par rapport à un axe. Des générations d'enfants, remonte au moins à l'antiquité grecque, ont trouvé la fascination dans le comportement des sommets, pour donner le gyroscope son nom commun. Un certain nombre d'éminents physiciens ont également trouvé le comportement complexe de filature objets une question d'intérêt et un bon sujet pour une analyse détaillée.

Plus récemment, gyroscopes très soigneusement conçus ont été utilisés pour la navigation car l'axe de points de spin dans une direction presque fixe lorsque couples externes sont de petite taille. Cela rend le gyroscope un bon remplacement pour un compas magnétique, en particulier dans les régions où les compas magnétiques ne sont pas fiables [8].

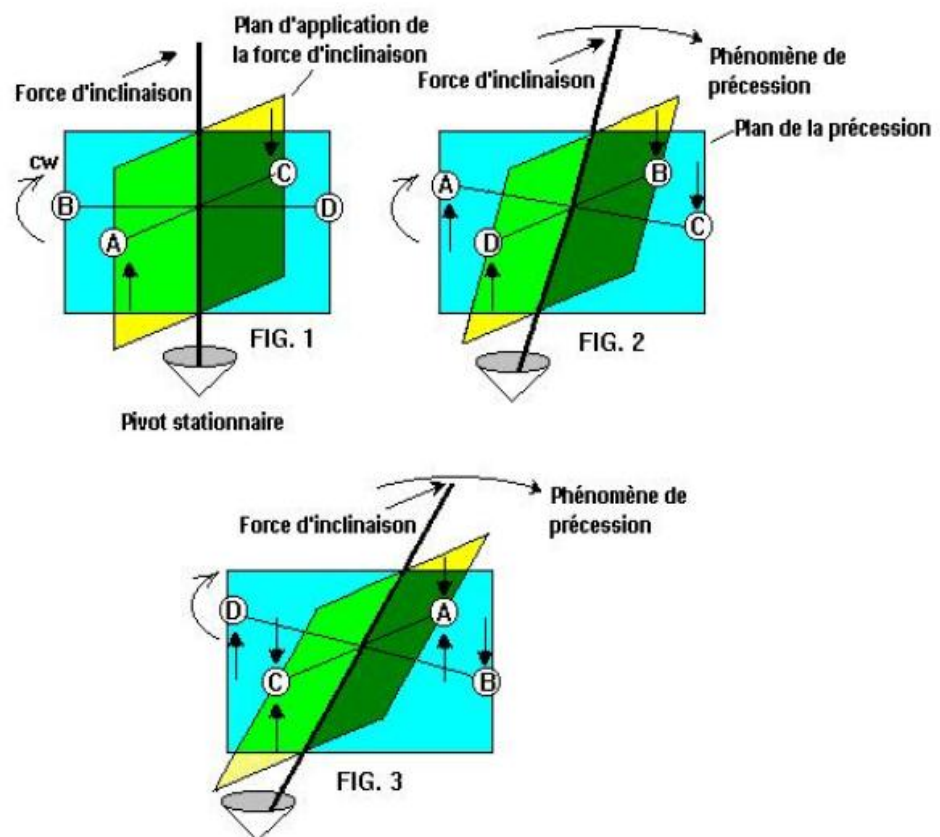


Figure .I.3. Principe de fonctionnement de gyroscope.

## I.4.2. Types de Gyroscopie

Les principaux types de gyroscopie sont présentés. Notez que ceci est loin d'être une liste exhaustive. En particulier, il existe de nombreuses variétés de gyroscopie mécanique qui ne sont pas décrits:

### I.4.2.1. Mécanique

Gyroscopie classique se compose d'une roue à filer montée sur deux balanciers qui lui permettent de pivoter dans les trois axes. Un effet de la conservation du moment cinétique est que le rouet résiste à des changements d'orientation. Par conséquent, quand un gyroscopie mécanique est soumise à une rotation de la roue restera dans une orientation globale constant et que les angles entre les balanciers adjacents va changer. Pour mesurer l'orientation de l'appareil les angles entre cardans adjacents peuvent être lus avec angle pick-offs. A noter qu'une mesure classique gyroscopie d'orientation. En revanche presque tous les gyroscopes sont modernes taux gyroscopes, qui mesurent la vitesse angulaire.

Le principal inconvénient de gyroscopes mécaniques est qu'ils contiennent des pièces en mouvement. Les pièces mobiles causées des frictions, ce qui entraîne à son tour la sortie à la dérive au fil du temps. Pour minimiser la friction des roulements de haute précision et de lubrifiants spéciaux sont utilisés, en ajoutant au coût de l'appareil. Gyroscopes mécaniques exigent également quelques minutes pour se réchauffer, ce qui est idéal dans de nombreuses situations.

### I.4.2.2. Optique

Un gyroscopie à fibre optique (FOG) utilise l'interférence de la lumière pour mesurer la vitesse angulaire. Un FOG se compose d'une grande bobine de fibre optique. Pour mesurer la rotation deux faisceaux lumineux sont déclenchés dans la bobine dans des directions opposées. Si le capteur est l'objet d'une rotation alors le faisceau circulant en sens de rotation connaîtra un trajet plus long à l'autre extrémité de la fibre que le faisceau voyage contre la rotation. Ceci est connu comme l'effet Sagnac. Lorsque les faisceaux sortent de la fibre, ils sont combinés. Le déphasage introduit par l'effet Sagnac amène les faisceaux à interférer, ce qui entraîne un faisceau combiné dont l'intensité dépend de la vitesse angulaire. Il est donc possible de mesurer la vitesse angulaire en mesurant l'intensité du faisceau combiné. Gyroscopes laser en anneau (de RLGS) sont également basés sur l'effet Sagnac. La différence entre un FOG et RLG est que, dans un RLG faisceaux laser sont dirigés dans un circuit fermé en utilisant des miroirs plutôt que la fibre optique.

Contrairement gyroscopes mécaniques, gyroscopes optiques contiennent aucune pièce mobile et ne nécessitent que quelques secondes pour le démarrage. La précision d'un gyroscopie optique est en grande partie fonction de la longueur du trajet de transmission de la lumière (plus grand est mieux), qui est limitée par la taille de l'appareil.

## I.4.3. MEMS Gyroscopes

Malgré des années de développement, gyroscopes mécaniques et optiques ont encore nombre de partie haute et un aiment besoin de pièces avec des tolérances de haute précision et des techniques d'assemblage complexes. En conséquence, ils restent coûteux. En revanche capteurs MEMS construits

en utilisant des techniques de micro-usinage du silicium ont un faible nombre de pièces (un gyroscope MEMS peut consister en aussi peu que trois parties) et sont relativement pas cher à fabriquer.

Gyroscopes MEMS faire usage de la force de Coriolis, qui stipule que dans un cadre de référence tournant à angulaire  $\omega$  de vitesse, une masse  $m$  se déplaçant à la vitesse  $v$  subit une force:

$$F_c = -2m(\omega \times v) \quad (I.1)$$

Gyroscopes MEMS contiennent des éléments vibrant pour mesurer l'effet de Coriolis. Beaucoup de géométries d'éléments vibrants existent, comme vibrants roues et diapason gyroscopes. La géométrie la plus simple consiste en une masse unique qui est entraîné à vibrer le long d'un axe d'entraînement. Lorsque les gyroscopes en rotation une vibration secondaire est amené le long de l'axe de détection perpendiculaire à cause de la force de Coriolis. La vitesse angulaire peut être calculée en mesurant cette rotation secondaire.

À l'heure actuelle les capteurs MEMS peuvent ne pas correspondre à la précision des dispositifs optiques, mais ils sont censés le faire dans l'avenir. Voici une liste des propriétés avantageuses des capteurs MEMS.

- Petite taille;
- Faible poids;
- Une construction robuste;
- Basse consommation énergétique;
- Court temps de démarrage;
- Peu coûteux à produire (en grande quantité);
- Grande fiabilité;
- Faible entretien;
- Compatible avec l'exploitation dans des environnements hostiles;

#### **I.4.4. Les Caractéristiques d'erreur de MEMS Gyroscope**

Nous examinons les erreurs qui se produisent dans des gyroscopes MEMS, et leur effet sur le signal intégré (d'orientation).

##### **I.4.4.1. Biais constant**

Le biais d'un gyromètre est la production moyenne du gyroscope quand il ne subit aucune rotation (à savoir: le décalage de la sortie de la valeur réelle), en 0 / h. Une erreur de polarisation constante de  $\epsilon$ , lorsqu'ils sont intégrés, provoque une erreur angulaire qui croît linéairement avec le temps

$$\theta(t) = \epsilon * t \quad (I.2)$$

L'erreur de polarisation constante d'un gyromètre peut être estimée en prenant une moyenne à long terme de la sortie du gyroscope alors qu'il ne subit aucune rotation. Une fois que le biais est connu, il est trivial de le compenser en soustrayant simplement le biais de la sortie.

### I.4.4.2. Effets de la température

Les fluctuations de température en raison de changements dans l'environnement et le capteur auto-échauffement induisent un mouvement dans le biais. Notez que ces mouvements ne sont pas inclus dans les mesures de stabilité de polarisation qui sont prises dans les conditions fixées.

Présenter toute distorsion résiduelle due à un changement de température provoque une erreur dans l'orientation qui croît linéairement avec le temps. La relation entre le biais et la température est souvent fortement non linéaire pour les capteurs MEMS. La plupart des unités de mesure inertielle (IMU) contiennent des capteurs de température internes qui permettent de corriger pour la température induit des effets de polarisation. Certains IMU comme le Xsens4 Mtx effectuer ces corrections en interne.

L'importance relative de chaque source d'erreur varie selon les gyroscopes. Pour angle MEMS des gyroscopes marche aléatoire (bruit) des erreurs et des erreurs de polarisation non corrigées soit en raison de non compensées les fluctuations de température ou d'une erreur dans l'estimation de polarisation initiale sont généralement les plus importantes sources d'erreur. Angle marche aléatoire peut être utilisé comme une borne inférieure de l'incertitude dans l'orientation obtenue à partir de l'intégration d'un signal de vitesse gyroscopes.

## I.5. Accéléromètres linéaires

### I.5.1. Définition de l'accéléromètre

Un accéléromètre est un appareil qui mesure l'accélération correcte ("G-Force"). Accélération propre est pas la même que la coordination accélération (taux de variation de vitesse). Par exemple, un accéléromètre au repos sur la surface de la Terre permettra de mesurer une accélération  $g = 9,81 \text{ m/s}^2$  droit vers le haut. En revanche, les accéléromètres en chute libre en orbite et l'accélération due à la gravité de la Terre mesureront zéro [8].

Accéléromètres ont de multiples applications dans l'industrie et la science. Hautement accéléromètres sensibles sont des composants de systèmes de navigation inertielle pour avions et missiles. Les accéléromètres sont utilisés pour détecter et surveiller les vibrations dans les machines tournantes. Accéléromètres sont utilisés dans les ordinateurs tablettes et les appareils photo numériques afin que les images sur les écrans sont toujours affichées en position verticale. Accéléromètres sont utilisés dans les drones pour la stabilisation de vol. Des paires d'accéléromètres étendus sur une région de l'espace peuvent être utilisées pour détecter des différences (gradients) dans les accélérations propres des trames de références associées à ces points, ces dispositifs sont appelés gradiométrie gravité, car ils mesurent gradients dans le champ de gravitation. De telles paires d'accéléromètres en théorie peuvent également être en mesure de détecter des ondes gravitationnelles. modèles mono et multi-axes de l'accéléromètre sont disponibles pour détecter l'ampleur et la direction de l'accélération propre (ou G-Force), comme une quantité de vecteur, et peuvent être utilisés pour détecter l'orientation (parce que la direction des changements de poids), de coordonner accélération (tant qu'il produit g-force ou un changement de g-force), les vibrations, les chocs, et de tomber dans un milieu résistif (un cas où les changements d'accélération appropriées, car il commence à zéro, puis

augmente). Accéléromètres micro-usinés sont de plus en plus présent dans les appareils électroniques portables et les contrôleurs de jeux vidéo, pour détecter la position de l'appareil ou de fournir pour l'entrée de jeu.

## **I.5.2. Types d'Accéléromètre**

Un accéléromètre peut être classé comme étant soit un dispositif de l'état mécanique ou solide. Ces deux types d'accéléromètre sont décrits, de même que les accéléromètres MEMS.

### **I.5.2.1. Mécanique**

Un accéléromètre mécanique se compose d'une masse suspendue par des ressorts, le déplacement de la masse est mesuré en utilisant un capteur de déplacement d'arrêt, ce qui donne un signal qui est proportionnel à la force  $F$  agissant sur la masse dans la direction de l'axe d'entrée. la deuxième loi de Newton  $F = ma$  est ensuite utilisée pour calculer l'accélération agissant sur le dispositif.

### **I.5.2.2. État solide**

Accéléromètres à semi-conducteurs peuvent être divisés en plusieurs sous-groupes, y compris les ondes acoustiques de surface, tory vibrations, de silicium et de dispositifs de quartz. Accéléromètres à semi-conducteurs sont de petite taille, fiable et robuste.

Un exemple d'un accéléromètre à semi-conducteurs est de l'accéléromètre à ondes acoustiques de surface (SAW), un accéléromètre à ondes acoustiques est constitué d'une poutre en porte à faux qui est en résonance à une fréquence particulière, une masse est fixée à une extrémité de la poutre qui est libre de se déplacer. L'autre extrémité est rigidement fixée au boîtier. Lorsque l'accélération est appliquée le long l'axe d'entrée des virages de faisceau. Cela provoque la fréquence de l'onde acoustique de surface pour modifier proportionnellement à la contrainte appliquée. En mesurant cette variation de fréquence de l'accélération peut être déterminée.

## **I.5.3. MEMS Accéléromètres**

Accéléromètres en silicium micro-usinées utilisent les mêmes principes que les capteurs d'état mécanique et solides. Il existe deux grandes classes d'accéléromètre MEMS. La première classe est constituée d'accéléromètres mécaniques (à savoir: dispositifs qui mesurent le déplacement d'une masse supportée) fabriqué en utilisant des techniques MEMS. La deuxième catégorie se compose d'appareils qui mesurent la variation de la fréquence d'un élément vibrant provoqué par un changement de la tension, comme dans accéléromètres SAW.

Les avantages des dispositifs MEMS appliquent également aux accéléromètres comme ils le font à des gyroscopes. Ils sont petits, légers et ont une faible consommation d'énergie et des temps de démarrage. Leur principal inconvénient est qu'ils ne sont pas actuellement aussi précis qu'accéléromètres fabriqués en utilisant des techniques traditionnelles, bien que les performances des dispositifs MEMS s'améliorent rapidement.

## I.5.4. Accéléromètre MEMS erreur Caractéristiques

Nous examinons les erreurs qui se posent dans les accéléromètres MEMS. Les types d'erreurs décrites ci-dessous sont analogues à ces quatre gyroscopes, et sont donc présentés dans moins de détails ici. La différence importante entre les erreurs résultant des accéléromètres est qu'ils sont intégrés à deux reprises afin de suivre la position, alors que les signaux de fréquence-gyro ne sont intégrés une fois de suivre l'orientation.

### I.5.4.1. Biais constant

Le biais d'un accéléromètre est le décalage de son signal de sortie de la valeur réelle, en  $m / s^2$ . Une erreur de polarisation constante de  $\varrho$ , lors d'un double intégré, provoque une erreur dans la position qui pousse quadratique avec le temps. L'erreur cumulée en position est la suivante:

$$s(t) = \varepsilon \cdot \frac{t^2}{2} \quad (I.3)$$

Où  $t$  est le temps d'intégration.

Il est possible d'estimer le biais de la mesure de la moyenne à long terme de la sortie des accéléromètres quand il ne subit aucune accélération. Malheureusement, ceci est compliqué par gravité, depuis un composant de gravité agissant sur l'accéléromètre apparaîtra comme une polarisation. Il est donc nécessaire de connaître précisément l'orientation du dispositif par rapport au champ de gravitation pour mesurer la polarisation. Dans la pratique, cela peut être réalisé par les routines de calibrage dans lequel le dispositif est monté sur un plateau tournant dont l'orientation peut être commandée de façon extrêmement précise.

### I.5.4.2. Flicker bruit / Biais stabilité

Accéléromètres MEMS sont soumis à clignoter bruit, ce qui provoque le biais de se promener au fil du temps. Ces fluctuations sont généralement modélisées comme un biais marche aléatoire, l'aide de ce modèle, le bruit de scintillation crée un second ordre de marche aléatoire de la vitesse dont l'incertitude croît proportionnellement à  $t^{(3/2)}$ , et un troisième ordre de marche aléatoire en position qui croît proportionnellement to  $t^{(2/5)}$ .

### I.5.4.3. Effets de la température

Comme dans le cas des gyroscopes, des changements de température provoquent des fluctuations de la polarisation du signal de sortie. La relation entre le biais et la température dépend de l'appareil spécifique, mais il est souvent fortement non linéaire. Tout biais résiduel apporté provoque une erreur dans la position qui pousse quadratique avec le temps. Si l'IMU contient un capteur de température, alors il est possible d'appliquer des corrections aux signaux de sortie afin de compenser les effets dépendant de la température.

### I.5.4.4. Erreurs d'étalonnage

Les erreurs d'étalonnage (erreurs de facteurs d'échelle, les alignements et sortie linéarité) apparaissent comme des erreurs de polarisation qui ne sont visibles tandis que le dispositif subit une

accélération. Notez que ces erreurs de biais «temporaires» peuvent être observées même lorsque l'appareil est à l'arrêt en raison de l'accélération gravitationnelle.

Les principales sources d'erreur pour accéléromètres MEMS sont similaires à ceux des gyroscopes. Pour les dispositifs MEMS angle de marche aléatoire (bruit) et des erreurs de polarisation non corrigées sont généralement les sources d'erreur qui limitent les performances de l'appareil, mais l'importance relative de chaque source d'erreur dépend de l'appareil spécifique utilisé. [8]

## I.6.Filtre de kalman

### I.6.1. Introduction

Le système d'analyse agit comme un intermédiaire entre les données brutes de l'IMU et le système de reconnaissance gestuelle. Son but est de transformer les données de manière à rendre la phase de reconnaissance plus facile et plus précis. La méthode choisie doit avoir une faible latence et, avec un oeil vers l'avenir, requièrent une puissance de traitement limitée. Trois algorithmes pour atteindre cet objectif sont examinés. Filtrage de Kalman peut être utilisé pour transformer les valeurs IMU de son cadre de coordonnées local à un cadre mondial fixe, qui est nécessaire si l'objet instrumenté n'est pas le seul élément d'intérêt et une sorte de suivi de position est nécessaire. Filtrer les données dans l'espace de fréquence peut être utilisé pour distiller les informations essentielles à partir des données en supprimant la basse fréquence de base et des décalages de bruit et les chocs à haute fréquence parasites.

### I.6.2. Définition

Le filtrage de Kalman est la principale technique d'analyse des données inertielles et est utilisée presque exclusivement pour le suivi inertiel, la détermination de la position et de l'orientation à partir des lectures d'inertie et un état initial. En tant que composants inertiels sont devenus plus fréquents dans les interfaces utilisateur, un certain nombre de variations sur les algorithmes originaux ont été créés pour assurer la robustesse et la précision lorsque l'on travaille avec des composants à faible coût et de matériel de traitement. Ces extensions seront examinées, de même que les limites imposées à ce suivi au fil du temps par des erreurs de détection. Filtre standard de Kalman est un estimateur linéaire, qui a la capacité de réduire au minimum la variance d'erreur. Les équations sont caractérisées par le type de calcul récursif, ce qui peut être relativement facile à réaliser sur un microprocesseur. Courant de sortie du filtre de Kalman dépend entrées de courant de l'Etat et actuels.

Le filtre est basant sur les équations de systèmes linéaires  $x$

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (\text{II.1})$$

$$y_k = Hx_k + z_k \quad (\text{II.2})$$

Où  $A$ ,  $B$  et  $M$  sont des matrices état, d'entrée et de mesure respectivement,  $x$  est l'état du système,  $u$  est connue entrée dans le système,  $y$  est la sortie mesurée,  $w$  est un bruit de processus et  $z$  est un bruit de mesure. Vecteur  $x$  contient état actuel du système, qui peut être estimée à partir des mesures fournies dans le vecteur  $y$  et donné entrée dans le système déterminé vecteur  $u$ . Cependant, nous ne pouvons pas compter entièrement sur des données de  $y$  à obtenir le vecteur  $x$ , parce que  $y$  est le bruit. [8] En fonction du système, toutes ces quantités peuvent être des vecteurs ( $y$  compris les éléments de plus d'un élément). Filtre de Kalman suppose que le bruit de processus et de mesure de bruit ne sont pas corrélés avec l'autre, tandis que leurs valeurs moyennes sont égales à zéro. Afin de satisfaire les hypothèses susmentionnées, matrices de covariance du bruit de processus ( $Q$ ), et le bruit de mesure ( $R$ ) sont donnés par les formules (3) et (4).

$$Q = E(w_k w_k^T) \quad (\text{II.3})$$

$$R = E(z_k z_k^T) \quad (\text{II.4})$$

Algorithme de filtre de Kalman peut être subdivisé à deux étapes différentes appelées respectivement la mise à jour de temps et de mise à jour de mesure. Au cours de la première étape, les valeurs du vecteur  $x$  (5) sont prévus et matrice de covariance  $P_k^-$  (6) est calculée.

$$x_k^- = Ax_{k-1} + Bu_{k-1} \quad (\text{II.5})$$

$$P_k^- = AP_{k-1}A^T + Q \quad (\text{II.6})$$

où  $x_k^-$  est l'estimation de l'état prédit à l'étape de temps réel,  $P_k^-$  est l'estimation de matrice de covariance prédite,  $P_{k-1}$  est la covariance estimée mise à jour à l'étape de temps précédent et  $x_{k-1}$  est mis à jour l'estimation d'état de l'étape de temps précédent.

(valeurs finales mises à jour) sont déterminées au cours de la deuxième étape de filtre de Kalman. La matrice de gain de Kalman est calculée à partir de l'équation donnée en (7). De Riccati Kalman et des équations de filtre sont donnés par:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (\text{II.7})$$

$$x_k = x_k^- + K_k (y_k - Hx_k^-) \quad (\text{II.8})$$

$$P_k = (I - K_k H) P_k^- \quad (\text{II.9})$$

où  $K_k$  est le gain de Kalman à l'étape de temps réel,  $x_k$  est le vecteur d'estimation état updat-ed à l'étape de temps réel,  $y_k$  est le vecteur de mesure-ment à l'étape de temps réel,  $P_k$  est l'estimation de matrice de covariance mise à jour au moment de réelle STEP, et  $I$  est la matrice identité

### I.6.3. Orientation

Afin de rendre filtre de Kalman applicable aux paramètres estimation des processus spécifiques, la connaissance de la description du système dynamique est nécessaire. Dans le cas de l'évaluation de l'orientation, qui système dynamique est donnée par les équations (10) et (11) si l'on note  $\alpha$  comme angle,  $\omega$  rythme que angulaire,  $q_b$  biais de gyroscope et T la période de l'échantillon

$$\alpha_k = \alpha_{k-1} + (\omega_k - q_{bk-1})T \quad (\text{II.10})$$

$$q_{bk} = q_{bk-1} \quad (\text{II.11})$$

Informations de Gyroscope peut être faussé par erreur de polarisation, qui devrait donc être, et est estimé en vecteur d'état et inclus dans l'équation dynamique du système mentionné ci-dessus, ce qui donne l'algorithme la possibilité de corriger la valeur de vitesse angulaire.

Prenant (12) en tant que vecteur d'état, matrices A, B et H sont donnés respectivement par (13), (14) et (15).

$$x = \begin{bmatrix} \alpha \\ q_b \end{bmatrix} \quad (\text{II.12})$$

$$A = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} \quad (\text{II.13})$$

$$B = \begin{bmatrix} T \\ 0 \end{bmatrix} \quad (\text{II.14})$$

$$H = [1 \quad 0] \quad (\text{II.15})$$

Vecteurs d'entrée et de mesures sont définies dans (16) et (17) équations respectivement. Entrée du filtre est la vitesse angulaire tandis que la mesure est l'angle.

$$u = [\omega] \quad (\text{II.16})$$

$$y = [\alpha] \quad (\text{II.17})$$

Matrice de covariance Q représentant de bruit du système, et R qui est la matrice du bruit de mesure de covariance sont les suivantes:

$$Q = \begin{bmatrix} T^2 \sigma_\alpha^2 & 0 \\ 0 & \sigma_\alpha^2 \end{bmatrix} \quad (\text{II.18})$$

$$R = [\sigma_\omega^2] \quad (\text{II.19})$$

Où  $\sigma_\alpha$  représente bruit de mesure d'orientation et  $\sigma_\omega$  représente bruit de mesure de vitesse angulaire

L'utilisation de trois consécutifs implémentations de filtrage de Kalman pour tangage, de roulis et de lacet conséquence est adoptée afin de réduire matrices taille et donc diminuer les besoins de calcul.

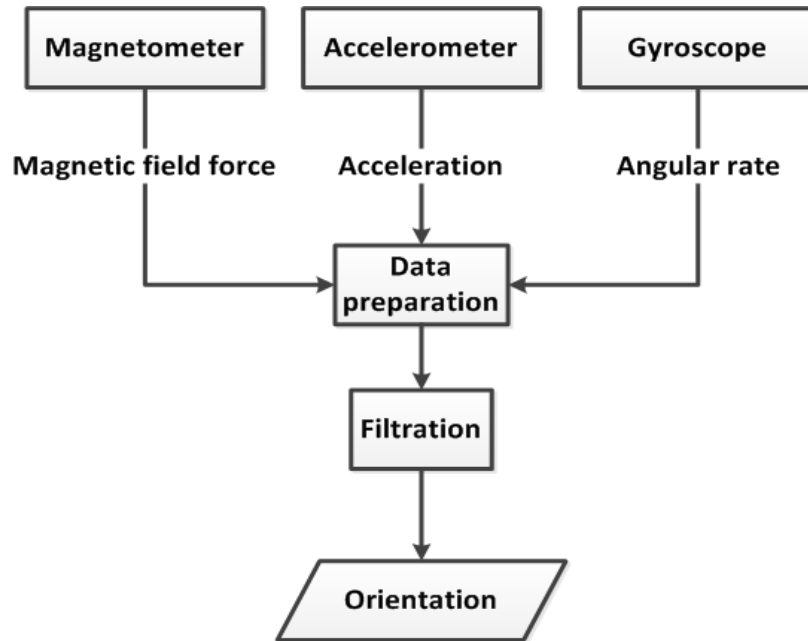


Figure .I.4. Flux d'information pour estimation d'orientation (Romaniuk, 2013)

## I.6.4. Position

Dans le cas de la position de filtration, l'information disponible correspond à la position actuelle et sa dérivée seconde a comme accélération. S prenant comme la distance parcourue, la vitesse V comme une que l'accélération,  $a_b$  comme biais accéléromètre et T en tant que période d'échantillonnage, système dynamique utilisée dans l'algorithme de filtre de Kalman, est décrit par trois équations. La première équation (20) décrit la manière dont la position est calculée. Il utilise des valeurs de position précédente, la vitesse précédente, l'accélération actuelle et biais accéléromètre précédente. La dernière valeur mentionnée peut, en fonction de la précision algorithme d'estimation, pratiquement réduire précisément l'erreur de mesure associée. La seconde équation (21) est utilisée pour calculer la vitesse, ayant sa valeur précédente, une accélération, ainsi qu'un biais d'accélération. La dernière équation (22) attribue une valeur de biais de l'accéléromètre de l'étape de temps avant l'actuel.

$$S_k = S_{k-1} + v_{k-1}T + \frac{(a_k - a_{bk-1})T^2}{2} \quad (\text{II. 20})$$

$$v_k = v_{k-1} + (a_k - a_{bk-1})T \quad (\text{II.21})$$

$$a_{bk} = a_{bk-1} \quad (\text{II.22})$$

Les équations indiquées ci-dessus peuvent être représentés sous la forme de l'espace d'état, où les trois variables d'état sont incluses dans le vecteur d'état (23).

$$x = \begin{bmatrix} S \\ v \\ a_b \end{bmatrix} \quad (\text{II.23})$$

Vecteurs d'entrée et de mesures sont définies dans (24) et (25) équations respectivement. Entrée du filtre est l'accélération tandis que la mesure est la position.

$$u=[a] \quad (\text{II.24})$$

$$y=[S] \quad (\text{II.25})$$

Matrices A, B et H décrivant le système dynamique sont donnés sous la forme suivante:

$$A = \begin{bmatrix} 1 & dt & \frac{-T^2}{2} \\ 0 & 1 & -T \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.26})$$

$$B = \begin{bmatrix} \frac{T^2}{2} \\ T \\ 0 \end{bmatrix} \quad (\text{II.27})$$

$$H = [1 \ 0 \ 0] \quad (\text{II.28})$$

Représentations de covariance de bruit du système et le bruit de mesure covariance sont adoptées par Q (29) et R (30) matrices respectivement.

$$Q = \begin{bmatrix} \frac{T^4}{4} \sigma_p^2 & \frac{T^3}{2} \sigma_p^2 & 0 \\ \frac{T^3}{2} \sigma_p^2 & T^2 \sigma_p^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{II.29})$$

$$R = [\sigma_a] \quad (\text{II.30})$$

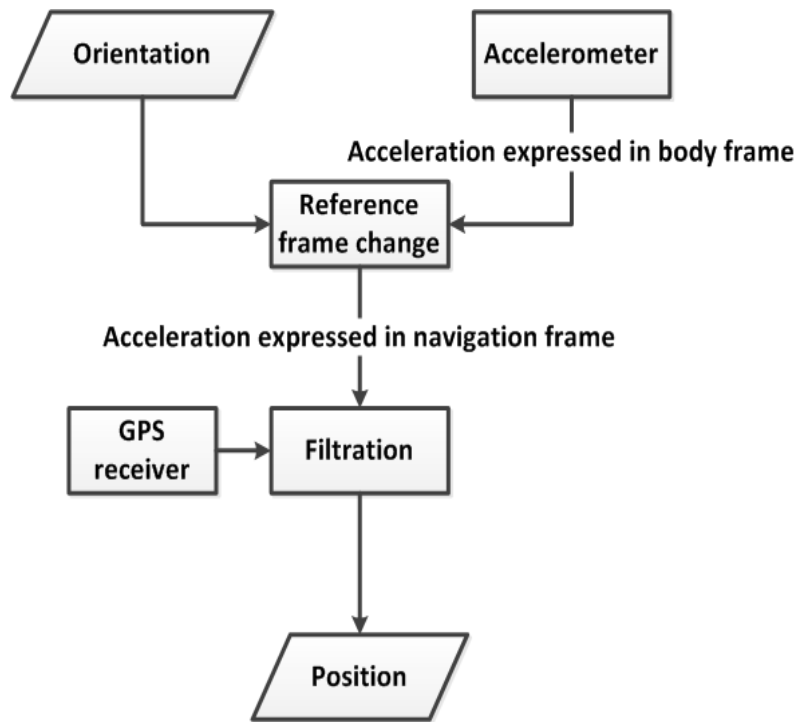


Figure .I.5. Flux d'information pour estimation de la position.

## I.7. conclusion

L'importance relative de chaque source d'erreur varie selon les gyroscopes. Pour angle MEMS des gyroscopes marche aléatoire (bruit) des erreurs et des erreurs de polarisation non corrigées soit en raison de non compensées les fluctuations de température ou d'une erreur dans l'estimation de polarisation initiale sont généralement les plus importantes sources d'erreur. Angle marche aléatoire peut être utilisé comme une borne inférieure de l'incertitude dans l'orientation obtenue à partir de l'intégration d'un signal de vitesse gyroscopes.

Les principales sources d'erreur pour accéléromètres MEMS sont similaires à ceux des gyroscopes. Pour les dispositifs MEMS angle de marche aléatoire (bruit) et des erreurs de polarisation non corrigées sont généralement les sources d'erreur qui limitent les performances de l'appareil, mais l'importance relative de chaque source d'erreur dépend de l'appareil spécifique utilisé.

# Chapitre II

# Chapitre II

## Arduino

---

### II.1. Introduction

Le projet Arduino est issu d'une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea 1 (Italie). Ils rencontraient un problème majeur à cette période (avant 2003 - 2004) : les outils nécessaires à la création de projets d'interactivité étaient complexes et onéreux (entre 80 et 100 euros). Ces coûts souvent trop élevés rendaient difficiles le développement par les étudiants de nombreux projets et ceci ralentissait la mise en œuvre concrète de leur apprentissage.

Jusqu'alors, les outils de prototypage étaient principalement dédiés à l'ingénierie, la robotique et aux domaines techniques. Ils sont puissants mais leurs processus de développement sont longs et ils sont difficiles à apprendre et à utiliser pour les artistes, les designers d'interactions et, plus généralement, pour les débutants [9].

### II.2. Qu'est-ce qu'un microcontrôleur ?

Un microcontrôleur est un circuit intégré (ou IC, Integrated Circuit), qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit. Au temps des pionniers de l'électronique, on soudait un grand nombre de composants encombrants, tels que les transistors, les résistances ou les condensateurs, sur des cartes plus ou moins grandes. Aujourd'hui, tout peut loger dans un petit boîtier en plastique noir muni d'un certain nombre de broches. Ces dernières sont les connexions du circuit intégré au moyen desquelles s'effectue la communication. La figure II-1 montre un microcontrôleur ATmega328, qu'on trouve sur la carte Arduino [10].

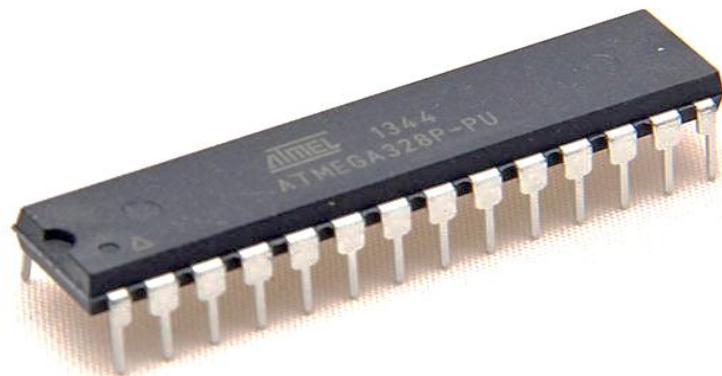


Figure II.1. Microcontrôleur ATmega328

Avec ses dimensions réduites, il dispose pourtant d'une grande puissance de calcul. En fait, il suffit de le souder sur une carte et de le mettre sous tension pour pouvoir l'utiliser. Certes, il manque encore quelques composants (par exemple, des stabilisateurs de tension, des connexions pour la programmation, et d'autres sur lesquels nous reviendrons plus tard), mais il est cependant sous cette forme déjà (presque) prêt à l'emploi.

## II.3. À quoi sert-il ?

Les microcontrôleurs jouent un rôle prépondérant dans les domaines suivants – cette liste est loin d'être exhaustive et sert surtout à se faire une idée des diverses possibilités d'utilisation.

- Fonctions de surveillance dans des environnements critiques, par exemple dans des cages thoraciques (température, humidité, fréquence cardiaque, pression sanguine du prématuré...).
- Commande de chauffage : contrôle de la température externe ou interne pour le chauffage optimal de locaux.
- Stimulateurs cardiaques : surveillance de la fréquence cardiaque et, le cas échéant, stimulation du cœur.
- Appareils ménagers : par exemple, commande par programme enregistré dans des lave-linge ou lave-vaisselle modernes.
- Électronique de loisirs : lecteurs MP3, téléphones portables, appareils photo...
- Robotique : par exemple, commande de robots industriels pour le montage de pièces automobiles.

Cette liste peut ainsi se poursuivre à l'infini, mais nous pouvons d'ores et déjà remarquer une chose : les microcontrôleurs perçoivent des influences extérieures par le biais de capteurs, les traitent en interne à l'aide d'un programme, puis envoient des ordres de commande correspondants vers l'extérieur. Ils font donc preuve d'une certaine intelligence, qui dépend bien évidemment du programme mis en œuvre. Un microcontrôleur peut assurer des fonctions de mesure, de commande et de régulation. de plus près le fonctionnement d'une boucle de régulation. Elle se compose d'un processus en boucle fermée comportant une perturbation. Un capteur transmet cette dernière au microcontrôleur qui réagit alors en fonction de son programme.

Si on trouve au sein du système de contrôle de chauffage qui régule la température de votre local de travail. Le capteur dit au microcontrôleur : « Dis donc, il fait plutôt chaud dans le local de travail ! » Le microcontrôleur réagit alors en régulant la température. Le chauffage apporte donc moins d'énergie sous forme de chaleur dans le local. Le capteur le remarque et dit au microcontrôleur : « La température est maintenant celle souhaitée, soit 20 °C. » De l'air froid provenant de l'extérieur rentre petit à petit. Le capteur donne l'alerte et dit au microcontrôleur : « Il commence à faire froid ici et mon bonhomme va tomber malade, il faut faire quelque chose ! » Le microcontrôleur augmente la température en conséquence. Vous voyez, c'est un jeu de ping-pong : ici, en l'occurrence, une boucle de régulation qui réagit à des influences perturbatrices extérieures liées à des variations de température [11].

## II.4. Structure d'un microcontrôleur :

La structure générale d'un microcontrôleur et les différents composants de la puce, c'est un ordinateur complet sur un espace réduit au maximum, avec donc les éléments suivants :

- unité centrale (CPU) ;
- mémoire de travail ;
- mémoire de données ;
- horloge interne ;
- ports d'entrée et de sortie.

Un microcontrôleur se divise grossièrement en trois parties :

- unité centrale (CPU) ;
- mémoires (ROM et RAM) ;
- ports d'entrée et de sortie.

L'horloge interne, ou l'oscillateur qui permet de piloter l'unité centrale, a été laissée de côté pour le moment. Les éléments qui composent un microcontrôleur sont comparables aux périphériques d'un ordinateur. La différence réside dans le fait que les trois parties citées précédemment sont intégrées au microcontrôleur. Elles se trouvent toutes dans le même boîtier, ce qui est plus simple et plus compact. Maintenant un coup d'œil au schéma fonctionnel de notre microcontrôleur [12].

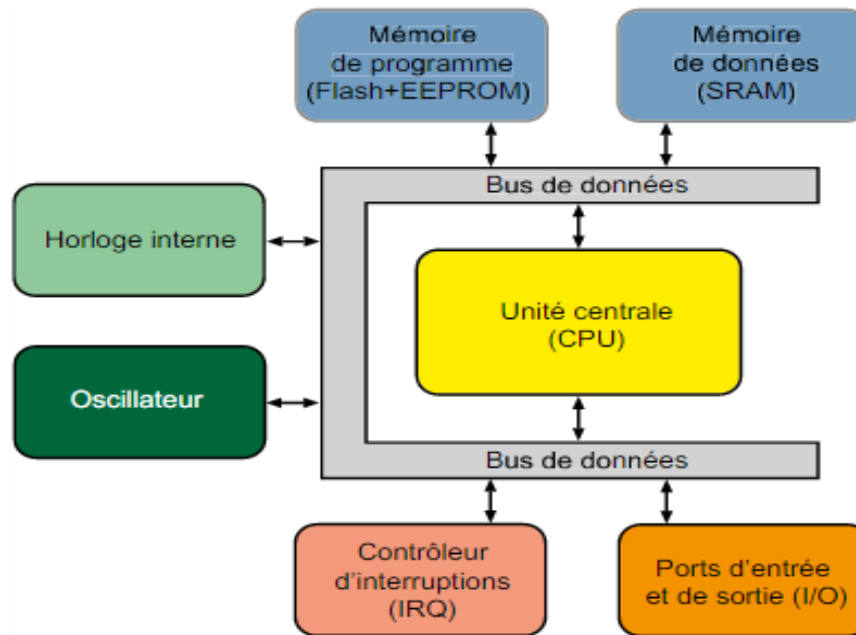


Figure .II.2. Schéma fonctionnel d'un microcontrôleur

#### II.4.1. L'unité centrale (CPU)

L'élément le plus important dans un microcontrôleur est l'unité centrale, appelée également CPU (Central Processing Unit). Sa fonction principale consiste à décoder et à exécuter des instructions. Elle peut adresser des mémoires, gérer des entrées ou sorties et réagir à des interruptions (interrupts) . Une interruption (IRQ, ou Interrupt Request ) est un signal qui demande au CPU d'interrompre un cycle de calcul en cours pour pouvoir réagir à un certain événement.

#### II.4.2. Le bus de données

Le bus de données sert à transporter les données d'un bloc à un autre. Par exemple, le CPU demande des données provenant de la mémoire, qui sont prises en charge par le bus et immédiatement mises à disposition pour traitement. Lorsque le résultat du calcul est disponible, il est à nouveau transféré sur le bus et transmis à un port de sortie qui, par exemple, pilote un moteur de robot pour atteindre un but précis. Cette structure de bus est une autoroute de données utilisable en commun par tous ceux qui sont desservis.



Figure .II.3. Sur l'autoroute des données : « Prochain arrêt, mémoire !

### II.4.3. Les zones de mémoire

En principe, il existe deux types de mémoires d'un microcontrôleur :

- la mémoire de programme ;
- la mémoire de données.

La première accueille le programme que le CPU doit exploiter, alors que la seconde est utilisée pour gérer les résultats de calcul du moment.

C'est vrai et c'est en cela que la mémoire de programme d'un microcontrôleur est particulière. Un microcontrôleur n'a bien sûr pas de disque dur, mais il garde son programme en mémoire en l'absence de tension d'alimentation externe. Un type de mémoire particulier est utilisé à cet effet, qu'on appelle mémoire flash. Comme son nom l'indique, c'est une mémoire non volatile, c'est-à-dire que bits et octets ne « s'envolent » pas en cas de coupure d'alimentation et restent disponibles.

Cette forme de mémoire des milliers de fois sur votre ordinateur. Le BIOS est hébergé dans une mémoire flash de type EEPROM, et ses données peuvent être écrasées au besoin par l'insertion d'une nouvelle version. On dit alors que le BIOS est « reflashé ». Le contraire se produit dans les mémoires de données dites SRAM. Ces dispositifs sont volatils et les données mémorisées sont perdues dès qu'il y a coupure d'alimentation. Mais rien de grave puisque ces dernières ne sont nécessaires que lorsque le programme est exécuté. Quand le microcontrôleur est sans courant, il n'a rien besoin de calculer. Mais la mémoire SRAM a sur la mémoire flash un avantage très important : elle offre un accès plus rapide.

Les ports d'entrées-sorties sont les connexions qui relient le micro- contrôleur au monde extérieur. Ils constituent une interface à laquelle la périphérie peut être connectée. On entend par périphérie tout ce qui peut être avantageusement raccordé à l'interface. Il peut s'agir, par exemple, des composants électroniques ou électriques suivants :

- LED (diode électroluminescente) ;
- bouton-poussoir ; • commutateur ;
- LDR (Light Dépendant Résistor ou photorésistance) ;
- transistor ;
- résistance ;

- haut-parleur ou élément piézoélectrique ;
- etc.



Figure .II.4. Ports d'entrée et de sortie

En principe, les ports d'entrées-sorties sont soit numériques, soit analogiques – plus loin ce qui les différencie [13].

## II.5. Arduino

### II.5.1. Arduino est-il un microcontrôleur ?

Cette partie est consacré aux principes généraux d'un microcontrôleur- leur. Maintenant les principaux composants (unité centrale, mémoires, ports). Mais on pose la question suivante : « Arduino est-il un microcontrôleur à proprement parler ? » La réponse est oui, sans aucun doute ! Il possède bien tous les composants dont nous avons parlé et les réunit en son sein. Mais il cohabite aussi avec d'autres composants sur une carte compacte, dont nous allons parler dans le prochain chapitre.

### II.5.2. Les différentes cartes Arduino

Les cartes Arduino doivent satisfaire des exigences diverses et variées. Certains utilisateurs souhaiteront effectuer du prototypage et tester de nouveaux montages ou idées, la carte Arduino Uno disposant d'assez d'entrées-sorties pour les projets d'envergure raisonnable. De par sa taille, la carte devrait aussi pouvoir être utilisée à l'avenir dans des projets plus ambitieux. D'autres auront besoin d'un grand nombre de ports afin de pouvoir raccorder de nombreux capteurs ou actionneurs. Une troisième catégorie d'utilisateurs ne cherchera qu'à attirer l'attention et à transmettre des signaux à leurs semblables à l'aide de diodes clignotantes. Ces exemples ne constituent qu'une partie des attentes que les makers auront vis-à-vis de leur carte Arduino.

La forme, la taille ou les possibilités de connexion jouent un rôle décisif dans le choix de la carte adaptée. C'est pourquoi les développeurs d'Arduino ont mis au point un vaste choix de cartes à micro- contrôleur afin que chacun trouve le modèle qui réponde à ses besoins.

Dans ce chapitre, nous allons donc passer en revue les principaux membres de la famille Arduino. Même si nous ne détaillerons pas tous les modèles, cette présentation vous sera certainement utile lorsque vous devrez choisir votre carte tout en ménageant votre porte- monnaie. Pourquoi se

ruiner en achetant la carte originale quand de nombreux fabricants proposent des clones dont les fonctionnalités sont similaires ? Je me limiterai (le plus souvent) aux modèles d'origine, mais libre à vous de choisir une autre voie. Je vais commencer ce petit tour d'horizon par la pièce maîtresse.

### II.5.2.1. Arduino Uno

La carte indispensable pour débiter demeure la carte Arduino Uno dont le prix est très abordable. Elle s'est hissée au rang de standard de fait et elle convient parfaitement à tous ceux qui veulent faire leurs premières armes avec un microcontrôleur. C'est une valeur sûre.

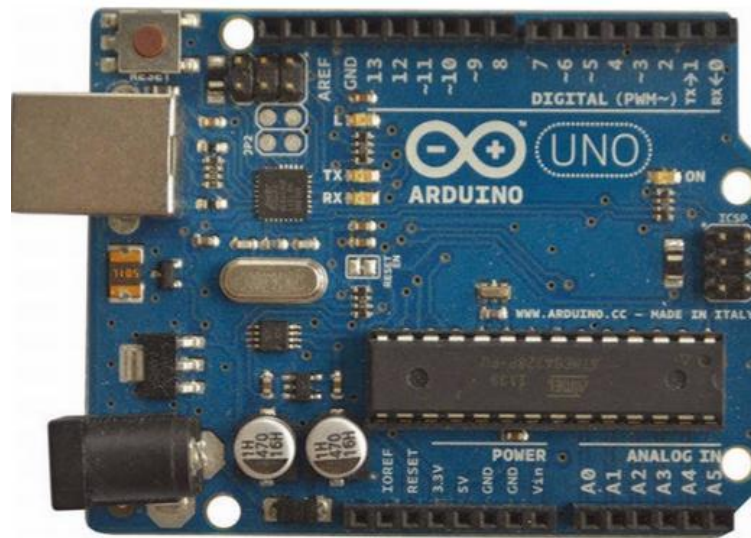


Figure .II.5. La carte Arduino Uno.

Catégorie	Valeur
Microcontrôleur	ATmega 328
Fréquence d'horloge	16 MHz
Tension de service	5 V
Tension d'entrée (recommandée)	7-12 V
Tension d'entrée (limites)	6-20 V
Ports numériques	14 entrées et sorties (6 sorties commutables en MLI)
Ports analogiques	6 entrées analogiques
Courant maxi. par broche d'E/S (c.c.)	40 mA
Courant maxi. par broche 3,3 V	50 mA
Mémoire	32 Ko Flash, 2 Ko SRAM, 1 Ko EEPROM
Chargeur d'amorçage	0,5 Ko (en mémoire Flash)
Interface	USB
Dimensions	6,86 cm × 5,3 cm

Tableau .II.1. La carte Arduino Uno

## Les avantages et les inconvénients de la carte

### • Avantages

- carte Arduino par excellence pour laquelle de nombreux exemples de montage sont disponibles sur Internet ;
- nombre suffisant de broches d'entrées-sorties pour les projets élémentaires ;
- vaste choix de shields ;
- bon marché

### • Inconvénients

- nombre insuffisant de broches d'entrées-sorties pour les projets ambitieux ;
- la mémoire disponible risque d'être un peu juste pour les gros projets ;
- ne peut pas être utilisée comme hôte USB pour simuler un clavier ou une souris

### II.5.2.2. Arduino Mega 2560

J'ai mentionné plus haut que le nombre de broches d'E/S pouvait être insuffisant pour certains projets. Pour y remédier, vous pouvez envisager de passer à la carte Arduino Mega 2560. Vous ne manquerez pas de remarquer sur la figure 2-3 les nombreuses rangées de prises qui viennent compléter les ports d'entrée-sortie. La carte est inévitablement plus grande que l'Arduino Uno.



Figure .II.6. La carte Arduino Mega 2560

Voici un résumé des spécifications de la carte Arduino Mega 2560

Catégorie	Valeur
Microcontrôleur	ATmega 2560
Fréquence d'horloge	16 MHz
Tension de service	5 V
Tension d'entrée (recommandée)	7-12 V
Tension d'entrée (limites)	6-20 V
Ports numériques	54 entrées et sorties (15 sorties commutables en MLI)
Ports analogiques	16 entrées analogiques
Courant maxi. par broche d'E/S (c.c.)	40 mA
Courant maxi. par broche 3,3 V	50 mA
Mémoire	256 Ko Flash, 8 Ko SRAM, 4 Ko EEPROM
Chargeur d'amorçage	8 Ko (en mémoire Flash)
Interface	USB
Dimensions	10,16 cm × 5,3 cm

Tableau .II.2. La carte Arduino Uno

Maintenant les avantages et les inconvénients de la carte :

#### • Avantages

- nombreuses entrées et sorties pour raccorder des capteurs ou des actionneurs ;
- capacité de mémoire suffisante pour les gros projets ;
- plus de broches UART (4 ports de communication série) ;
- plus de broches MLI (15 sorties numériques peuvent être utilisées comme MLI) ;
- compatible avec la plupart des shields conçus pour l'Arduino Uno, par exemple ;
- il existe des shields spéciaux pour le prototypage qui, en raison de leur surface supérieure, peuvent recevoir plus de composants ;
- de nombreux schémas et exemples sont disponibles sur Internet.

#### • Inconvénients

- facteur de forme plus élevé que pour l'Arduino Uno, par exemple ;
- deux fois plus chère que l'Arduino Uno.

### II.5.3. Arduino Nano

La carte Arduino Nano possède des connecteurs au dos qui permettent de l'enficher facilement sur une plaque d'essais, ce qui évite d'avoir recours à des cavaliers flexibles, comme pour l'Arduino Uno. Ne vous laissez pas abuser par les dimensions de cette minicarte dont les performances n'ont (presque) rien à envier à l'Arduino Uno.

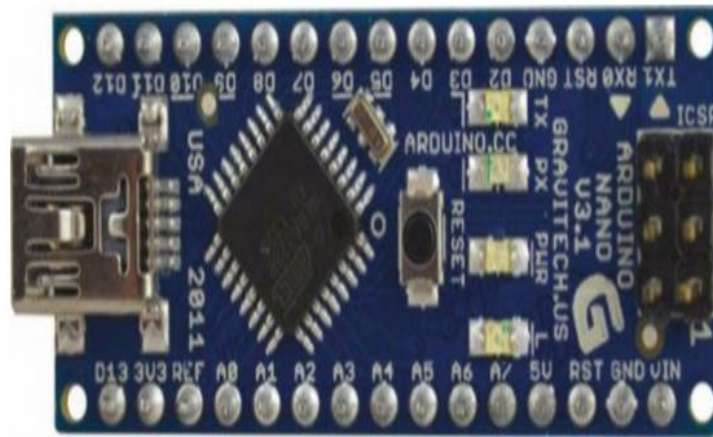


Figure .II.7. La carte Arduino Nano

Voici un résumé des spécifications de la carte Arduino Nano :

Catégorie	Valeur
Microcontrôleur	ATmega 168 ou 328
Fréquence d'horloge	16 MHz
Tension de service	5 V
Tension d'entrée (recommandée)	7-12 V
Tension d'entrée (limites)	6-20 V
Ports numériques	14 entrées et sorties (6 sorties commutables en MLI)
Ports analogiques	8 entrées analogiques
Courant maxi. par broche d'E/S (c.c.)	40 mA
Mémoire	ATmega 168 : 16 Ko mémoire Flash 1 Ko SRAM 512 octets d'EEPROM ATmega 328 : 32 Ko mémoire Flash 2 Ko SRAM 1 Ko EEPROM
Chargeur d'amorçage	2 Ko (en mémoire Flash)
Interface	USB
Dimensions	1,9 cm × 4,3 cm

Tableau .II.3. La carte Arduino Nano

Maintenant les avantages et les inconvénients de la carte :

- **Avantages**

- encombrement réduit ;
- peut être enfichée directement sur la plaque d'essais.

- **Inconvénients**

- il n'est pas possible d'utiliser des shields.

## II.5.4. Structure de la carte Arduino

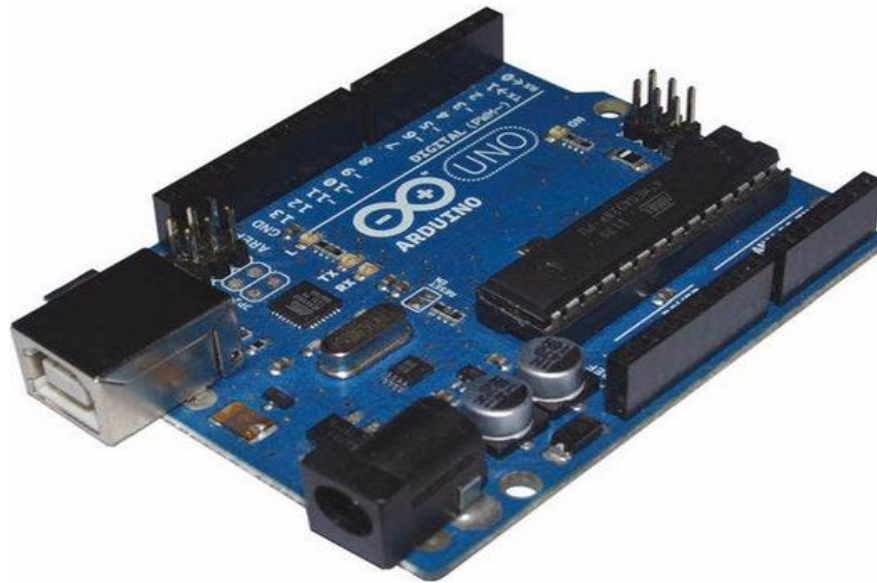


Figure .II.8. La carte Arduino Nano.

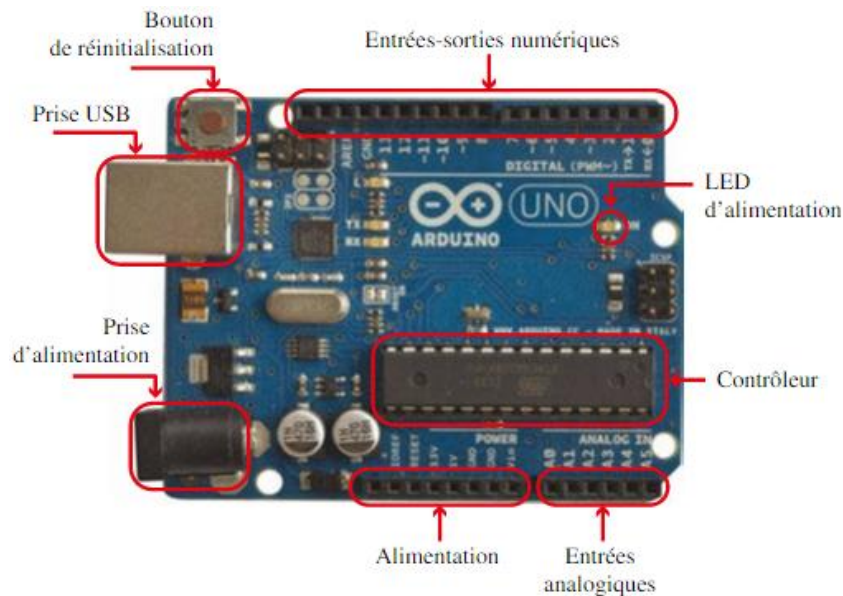
Sur cette image, vous ne pouvez évidemment pas vous rendre compte à quel point les dimensions de la carte Arduino sont réduites (environ 7 cm de large et 5 cm de long) ; elle est vraiment très maniable, tient sans problème dans une main et s'avère vraiment compacte.

Au fil des années, plusieurs cartes Arduino ont été développées. J'ai choisi de prendre l'exemple de la Uno, car c'est la plus populaire.

Les composants les plus divers y sont reconnaissables (voir figure 3-2), lesquels feront l'objet d'une explication détaillée. Certains penseront certainement qu'on ne peut pas concevoir quelque chose de sérieux sur une surface aussi réduite. Mais grâce à la miniaturisation des composants ces dernières années, ce qui nécessitait auparavant cinq puces électroniques n'en requiert aujourd'hui plus qu'une.

Le plus gros élément qui saute directement aux yeux est le microcontrôleur proprement dit. Il est de type ATmega328.

J'ai choisi la carte Arduino Uno R3 car, même si d'autres modèles ont suivi depuis, comme Arduino Due ou Arduino Yún, elle s'est quasiment hissée au rang de standard. Elle convient tout particulièrement à ceux qui veulent faire leurs premiers pas dans le monde des microcontrôleurs.



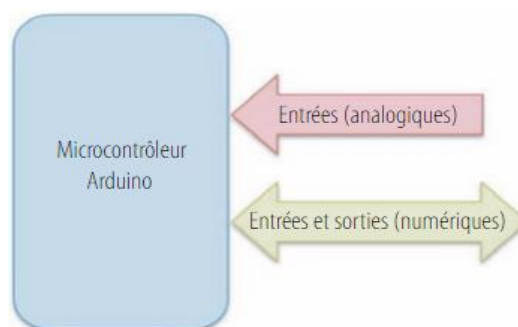
**Figure .II.9. Que trouve-t-on sur la carte Arduino ?**

Ces éléments sont les plus importants de la carte Arduino mais, bien entendu, cela ne veut pas dire que les autres sont à négliger.

Voici les principales caractéristiques de la carte Arduino :

- 2 Ko de SRAM ;
- 1 Ko d'EEPROM ;
- fréquence d'horloge 16 MHz ;
- interface USB.
- microcontrôleur ATmega328 ; • tension de service 5 V ;
- 14 entrées et sorties numériques (6 sorties commutables en MLI) ;
- 6 entrées analogiques (résolution 10 bits) ;
- 32 Ko de mémoire flash (0,5 Ko occupé par le chargeur d'amorçage ou boot loader) ;

Comme vous pouvez le constater, un certain nombre d'entrées ou de sorties sont disponibles pour communiquer avec la carte Arduino. Elles constituent l'interface avec le monde extérieur et permettent d'échanger des données avec le microcontrôleur, comme l'indique la figure II-10.



**Figure .II.10. Entrées et sorties de la carte Arduino**

Le microcontrôleur Arduino, peut communiquer avec nous via certaines interfaces. Certains ports servent d'entrées, et d'autres d'entrées et de sorties. Un port est ici un chemin d'accès défini au microcontrôleur, pratique- ment une porte vers l'intérieur qu'il est possible d'actionner.

Vous apercevez également des réglettes de raccordement noires sur ses bords supérieur et inférieur.

#### II.5.4.1. L'alimentation électrique

Notre carte Arduino doit être alimentée en énergie pour pouvoir travailler. Cette alimentation peut s'effectuer tout d'abord via l'interface USB qui relie la carte à l'ordinateur – ce chemin sert aussi à l'échange de données entre la carte et l'ordinateur. En phase de développement avec votre Arduino, la connexion USB va servir d'alimentation primaire de la carte. La seconde possibilité consiste à brancher une batterie ou un bloc secteur au connecteur, appelé prise jack. Vous pouvez, par exemple, employer cette variante si vous avez construit un engin manœuvrable, commandé par la carte Arduino. Le véhicule doit pouvoir évoluer librement dans l'espace, sans câble. En effet, l'utilisation d'un câble USB, généralement trop court, limiterait alors la mobilité de l'engin. L'emploi d'une batterie rend le dispositif autonome.

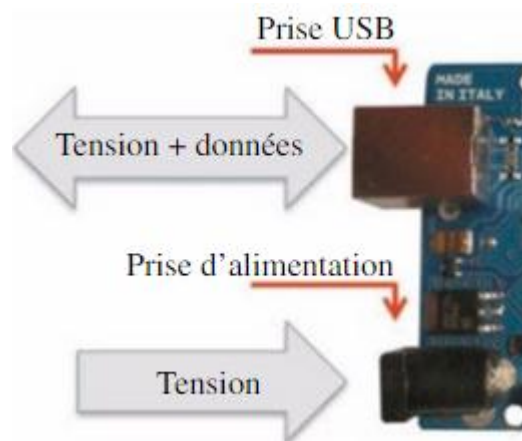


Figure .II.11. Alimentation de la carte Arduino.

#### II.5.4.2. Les modes de communication

Un microcontrôleur de type Arduino a déjà beaucoup de connexions qu'il s'agit de bien distinguer.

#### II.5.4.3. Le port USB

Sans le port USB, vous ne seriez pas en mesure d'initialiser une communication.

Le travail avec la carte Arduino peut se diviser en deux étapes : le temps consacré à la mise en œuvre du montage et celui dédié à la programmation, appelé phase de développement (design time).

La programmation s'effectue dans un environnement de développe- ment que vous allez apprendre à connaître très rapidement. C'est dans cet environnement que vous allez saisir le programme créé par vos soins pour le transmettre au microcontrôleur. Si tout s'est bien passé, le temps de l'exécution (runtime) commence. Vous n'avez pas besoin de dire explicitement au microcontrôleur. Il démarre en effet immédiatement après avoir reçu toutes les instructions de votre part. Vous pouvez

en outre échanger des données avec votre ordinateur via le port USB. Nous verrons plus tard comment cela fonctionne.

### II.5.4.4. Les ports d'entrée ou de sortie (E/S)

Les ports E/S représentent l'interface du microcontrôleur. Il existe plusieurs chemins ou canaux pour échanger des données, comme chez L'homme avec les yeux, les oreilles et la bouche. Il se produit, grâce et par le biais de ces canaux de communication, une interaction avec l'environnement.

Votre carte Arduino utilise des données provenant de capteurs (par exemple, de température, de lumière ou d'humidité) pour réagir en conséquence et entreprendre des actions appropriées. Elle peut aussi activer des dispositifs lumineux et sonores, ou agir sur des actionneurs (moteurs et capteurs).

Vous avez certainement compris que nous avons affaire à deux types de signaux de commande. Des capteurs fournissent des données, et des actionneurs convertissent des grandeurs d'entrée en grandeurs de sortie. Ce processus se déroule selon le principe ETS (Entrée, Traitement, Sortie).



FigureII-12 : Le principe ETS

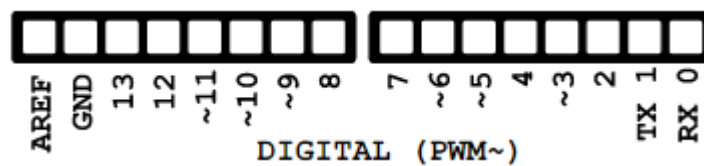


Figure .II.12. Entrées et sorties numériques.

Il est bien entendu important de savoir pour chaque port quelle est son adresse afin de pouvoir communiquer avec lui au cours de la programmation. C'est pour cela que chaque broche possède un numéro. La numérotation de la première broche commence par 0 – en programmation, presque toutes les numérotations commencent par 0. Sous certains chiffres se trouve un tilde (~) indiquant que la broche peut être commutée en sortie analogique. Il s'agit là d'une broche MLI (rappelez-vous la modulation de largeur d'impulsion ; on y reviendra un peu plus loin). À l'extrémité inférieure de la carte se situent les ports d'alimentation (à gauche) et les ports d'entrée analogiques (un bloc de huit broches et un autre de six broches).

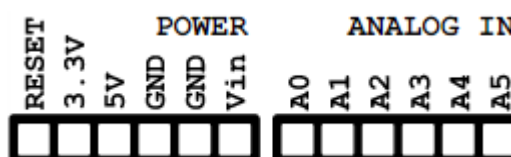


Figure .II.13. Alimentation et entrées analogiques.

La numérotation du port analogique commence par 0 mais, cette fois- ci, en partant de la gauche.

#### II.5.4.5. Les langages de programmation C/C++

Pour que la communication avec la carte Arduino se déroule sans problème, les développeurs doivent convenir d'une base de langage, afin qu'ils puissent se comprendre entre eux et exploiter un flux d'informations. C'est la même chose que lorsque vous allez à l'étranger et que vous ne maîtrisez pas la langue. Dans ce cas, vous devez vous adapter d'une façon ou d'une autre, peu importe la manière (gestes...).

Le microcontrôleur ne connaît à son niveau d'interprétation que le langage machine, appelé aussi code natif, composé exclusivement de valeurs numériques. Il est très difficile à comprendre, car nous avons appris tout petit à échanger à l'aide de mots et de phrases, et non de valeurs numériques. Nous devons donc trouver un moyen de pouvoir communiquer de manière compréhensible avec le microcontrôleur. C'est pourquoi un environnement de développement traduisant les commandes dans un langage dit évolué – autrement dit, se présentant sous une forme semblable à notre langage – a été créé. Pour autant, nous ne sommes pas plus avancés puisque le microcontrôleur ne comprend pas ce langage. En effet, il manque une sorte de traducteur servant de lien entre lui et l'environnement de développement. C'est le rôle du compilateur qui convertit un programme écrit en langage évolué en un langage cible compréhensible par le destinataire (ici, le CPU, ou Central Processing Unit , de notre microcontrôleur).



Figure .II.14. Le compilateur sert de traducteur.

Presque tous les langages de programmation font appel au vocabulaire anglais ; nous n'avons donc pas d'autre choix que de nous y mettre. Une autre étape de traduction sera donc nécessaire, mais je pense que l'anglais scolaire suffira ici dans la plupart des cas. Les instructions – autrement dit, les ordres – que l'environnement de développement comprend sont concises et semblables à celles du langage militaire, et représentent ce qu'il faut faire.

Ne vous en faites pas, elles vous seront enseignées au fur et à mesure. Comme l'indique fort justement le titre de cette section, C et C++ sont également des langages évolués. Aujourd'hui, tous les programmes professionnels sont écrits en C/C++ ou dans des langages apparentés tels que C# ou Java, qui ont tous une forme de syntaxe similaire.

À tous les programmeurs qui s'offusquent de ne pas voir ici leur langage favori, je tiens à préciser que cela ne signifie pas que je considère tous les autres langages (et il y en a beaucoup) comme non professionnels. Nous en restons ici à C/C++ car Arduino, tout comme le compilateur, dispose d'une partie de la fonctionnalité des langages C/CC++. Ainsi, ceux qui ont déjà programmé avec ne se sentiront pas perdus, et nous ferons en sorte que les autres se sentent eux aussi rapidement à l'aise. Par ailleurs, beaucoup d'autres packs de développe- ment avec microcontrôleur utilisent des

compilateurs compatibles C/ C++ ; autrement dit, l'étude de ces langages va bientôt se révéler utile. Mais concentrons-nous maintenant sur Arduino [14].

## II.6. Protocoles de communication

Aujourd'hui, à l'extrémité inférieure des protocoles de communication, nous trouvons I<sup>2</sup>C (pour Circuit intégrée », protocole) et le SPI (pour ' Serial Peripheral Interface»). Ces deux protocoles sont bien adaptés pour les communications entre les circuits intégrés, pour la lenteur des communications avec les périphériques embarqués. À l'origine de ces deux protocoles populaires, on trouve deux grandes entreprises - Philips pour I<sup>2</sup>C et Motorola pour SPI - et deux histoires différentes sur pourquoi, quand et comment les protocoles ont été créés

### II.6.1. Le protocole I<sup>2</sup>C

I<sup>2</sup>C (signifie : Inter-Integrated Circuit, en anglais) est un bus de données qui a émergé de la « guerre des standards » lancée par les acteurs du monde électronique. Conçu par Philips pour les applications de domotique et d'électronique domestique, il permet de relier facilement un micro processor et différents circuits, notamment ceux d'une télévision moderne : récepteur de la télécommande, réglages des amplificateurs basses fréquences, tuner, horloge, gestion de la prise péritel, etc.

Il existe d'innombrables périphériques exploitant ce bus, il est même imprésentable par logiciel dans n'importe quel microcontrôleur. Le poids de l'industrie de l'électronique grand public a permis des prix très bas grâce à ces nombreux composants.

Ce bus porte parfois le nom de TWI (Two Wire Interface) chez certains constructeurs.

#### II.6.1.1. Le support physique utilisé

Comme indiqué précédemment, pour se connecter à un bus I<sup>2</sup>C il faut une masse, et deux fils de communication. Le premier fil, SDA (Signal Data), est utilisé pour transmettre les données. L'autre fil, SCL (Signal CLock) est utilisé pour transmettre un signal d'horloge synchrone (signal qui indique le rythme d'évolution de la ligne SDA). Les tensions associées aux niveaux logiques vont dépendre de la technologie des circuits en présence (CMOS, TTL). Il faudra que tous les circuits connectés au bus I<sup>2</sup>C utilisent les mêmes potentiels pour définir les niveaux haut et bas. En définitive, cela implique que tous les composants connectés à un même bus soient alimentés de façon identique. Cela ne signifie pas que les composants doivent utiliser la même source pour s'alimenter ; il suffit que la tension d'alimentation soit à la même valeur pour tous les composants, le fil de masse permettant d'unifier les références.

Il reste maintenant un problème crucial. Comment permettre à plusieurs circuits logiques de connecter leurs sorties ensemble, sachant que certains circuits voudront imposer un niveau haut tandis que d'autres voudront imposer un niveau bas ? La réponse est connue depuis longtemps. Il faut utiliser des sorties à collecteur ouvert (ou à drain ouvert pour des circuits CMOS). Le niveau résultant sur la ligne est alors une fonction « ET » de toutes les sorties connectées.

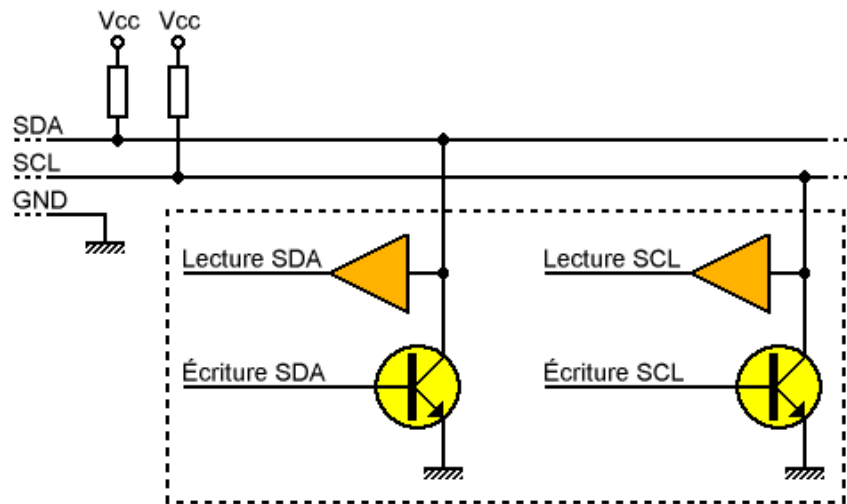


Figure .II.15. Présentation de Communication Serial.

Les résistances de rappel au potentiel VCC permettent aux signaux SDA et SCL d'être à 1 si toutes les sorties à collecteurs ouverts sont aussi au niveau 1 (résultat de la fonction « ET »). Si une ou plusieurs sorties tentent d'imposer un niveau bas sur une ligne, le ou les transistors associés vont conduire, ce qui entraîne un niveau bas sur la ligne correspondante (ce qui est conforme au résultat de la fonction « ET »).

En ce qui concerne la lecture des signaux SDA et SCL, cela ne pose pas de problème. Les signaux peuvent être lus en permanence sans risque d'interférer sur le niveau de la ligne.

Au repos, tous les circuits connectés doivent imposer un niveau haut sur leurs sorties respectives. Si les lignes SDA et SCL sont au niveau haut dans ces conditions, cela signifie qu'aucun circuit ne tente de prendre le contrôle du bus. Si une des lignes SDA ou SCL passe à un niveau bas dans les mêmes conditions, c'est qu'un des circuits désire prendre le contrôle du bus. Mais il peut aussi y avoir deux circuits qui tentent de prendre le contrôle du bus en même temps (ou à quelques nanosecondes d'écart près). Il faut donc mettre en place un protocole pour gérer les conflits possibles.

### II.6.1.2. Structure d'I2C

I2C est approprié pour l'interfaçage avec des dispositifs sur une seule carte, et peut être étiré sur plusieurs conseils d'administration dans un système fermé. Un exemple est un processeur de l'hôte sur une carte embarquée (CPU) principal en utilisant I2C pour communiquer avec les périphériques d'interface utilisateur situés sur une carte du panneau avant séparé. I2C est un bus série à deux fils, comme le montre la Figure 1. Il n'y a pas besoin de sélection de puce ou de la logique d'arbitrage, ce qui n'en fait pas cher et simple à mettre en œuvre dans le matériel. Les deux signaux de données en série sont I2C et de l'horloge série. Ensemble, ces signaux, permettent de soutenir la transmission série d'octets de 8 bits d'adresses de l'appareil de données 7 bits plus des bits de commande-sur le bus série à deux fils.

Dans une impasse, un esclave I2C peut tenir à distance le maître au milieu d'une transaction en utilisant ce qu'on appelle horloge étirement (l'esclave garde SCL tirée vers le bas jusqu'à ce qu'il soit prêt à continuer). Plus Le protocole I2C ne peut également prendre en charge plusieurs maîtres. Il peut

y avoir un ou plusieurs esclaves sur le bus. Les deux maîtres et esclaves peuvent recevoir et transmettre des octets de données.

Chaque dispositif esclave du matériel compatible I2C est livré avec une adresse de dispositif prédéfini, les bits inférieurs de ce qui peut être configurable au niveau du conseil d'administration. Le maître transmet l'adresse de l'appareil de l'esclave prévu au début de chaque transaction. Chaque esclave est chargé de surveiller le bus et de répondre seulement à sa propre adresse. Ce schéma d'adressage limite le nombre de dispositifs esclaves identiques qui peuvent exister sur un bus I2C sans prétention, à la limite fixée par le nombre de bits d'adresse configurable par l'utilisateur.

### II.6.1.3. Communication d'I2C

Le protocole de signalisation I2C fournit dispositif d'adressage, un drapeau de lecture / écriture, et un mécanisme de reconnaissance simple. D'autres éléments du protocole I2C sont appel général (de diffusion) et 10 bits d'adressage étendu. Périphériques I2C standard fonctionnent jusqu'à 100 Kbps, tandis que les dispositifs mode rapide fonctionnent à jusqu'à 400 Kbps. Le plus souvent, le maître I2C est le CPU ou un microcontrôleur dans le système. Certains microcontrôleurs sont même dotés de matériel pour mettre en œuvre le protocole I2C. Vous pouvez également créer une mise en œuvre de tous les logiciels utilisant une paire de fins générales broches E / S. Depuis la transaction chronométrage commandes principales I2C, le protocole de bus n'impose pas de contraintes en temps réel sur le processeur au-delà de celles de l'application. Pour une I2C fixe, les logiques hautes et bas sont définis à 3.0 V et 1.5 V. Pour I2C charge, ceux-ci sont définis à  $0.7 * V_{dd}$  et  $0.3 * V_{dd}$  respectivement. Les valeurs de résistance de rappel nécessaires pour I2C sont typiquement à 1K pour 3.0V de  $V_{dd}$  et 1.6K pour 5V de la  $V_{dd}$ . températures de fonctionnement typiques sont comprises entre -40 degrés et +85 degrés Celsius.

### II.6.1.4. La prise de contrôle du bus

Pour transmettre des données sur le bus I<sup>2</sup>C, il faut surveiller deux conditions particulières : la condition de départ et la condition d'arrêt

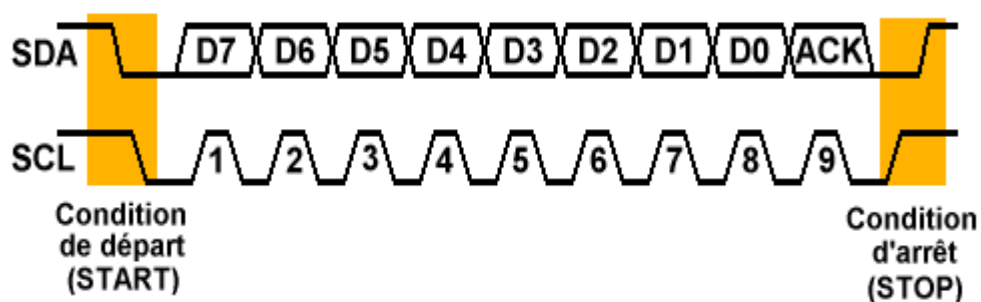


Figure .II.16. L'adressage d'I2C.

Avant de tenter de prendre le contrôle du bus, un circuit doit vérifier que les lignes SDA et SCL sont au repos, c'est-à-dire à l'état haut. Si c'est le cas, le circuit indique qu'il prend le contrôle du bus en mettant la ligne SDA à 0. A partir de ce moment-là, les autres circuits savent que le bus est occupé et ils ne devraient pas tenter d'en prendre contrôle. Le circuit qui vient de prendre le contrôle du bus en devient le maître (en anglais « master »). C'est lui qui génère le signal d'horloge, quel que soit le sens du transfert.

## II.6.2. Le protocole SPI

### II.6.2.1. Aperçue de SPI

SPI est une interface série synchrone à des fins générales. Lors d'un transfert SPI, transmettre et recevoir des données simultanément décalé et en série. Une ligne d'horloge synchronise le décalage série et l'échantillonnage des informations sur deux lignes de données en série. Motorola a créé le port SPI dans le milieu des années 1980 à utiliser dans leurs familles de produits de microcontrôleur. Le SPI est principalement utilisé pour permettre à un microcontrôleur de communiquer avec des périphériques tels qu'E2 PROM. Dispositifs SPI communiquent à l'aide d'une relation maître-esclave. En raison de son manque de dispositif intégré dans la lutte, SPI nécessite plus d'efforts et plus de ressources matérielles que I2 C lorsque plus d'un esclave est impliqué. Mais SPI a tendance à être plus simple et plus efficace que I2 C, point-à-point (un seul maître, esclave simple) des applications pour la même raison; l'absence de dispositif de moyens d'adressage moins de frais généraux [14].

### II.6.2.2. SPI Détails

SPI est un bus série standard établi par Motorola et soutenue dans les produits de silicium de différents fabricants. Interfaces SPI sont disponibles sur les processeurs et microcontrôleurs de communication populaires. Il est un lien de données série synchrone qui fonctionne en full duplex (signaux porteurs de données vont dans les deux sens en même temps). Appareils communiquent en utilisant une relation maître / esclave, dans lequel le maître initie la trame de données. Lorsque le maître génère une horloge et sélectionne un dispositif esclave, les données peuvent être transférées dans l'une ou les deux directions simultanément. En fait, en ce qui concerne SPI, les données sont toujours transférées dans les deux sens. Il est à la hauteur des dispositifs maître et esclave de savoir si un octet reçu est significatif ou non. Ainsi, un dispositif doit jeter l'octet reçu dans un "ne transmettre que" cadre ou générer un octet factice pour un "reçoivent" cadre. SPI spécifie quatre signaux: horloge (SCK 1); sortie principale de données, entrée de données esclave (SI1); entrée des données de base, la production de données esclave (SO1); et sélection de puce (CS). La figure 1 montre ces signaux dans une configuration à un seul esclave. SCK1 est généré par le maître et l'entrée à tous les esclaves. SI1 transporte les données du maître à esclave. SO1 transporte les données d'esclave à maître. Un dispositif esclave est sélectionné quand le maître affirme son signal CS [15].

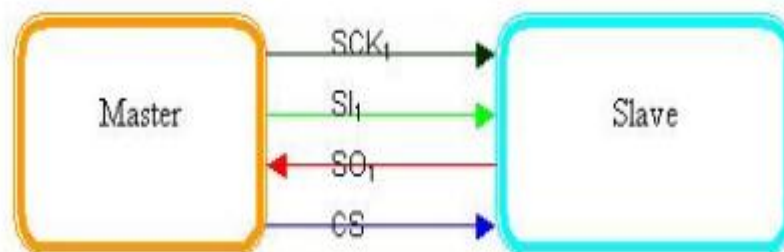
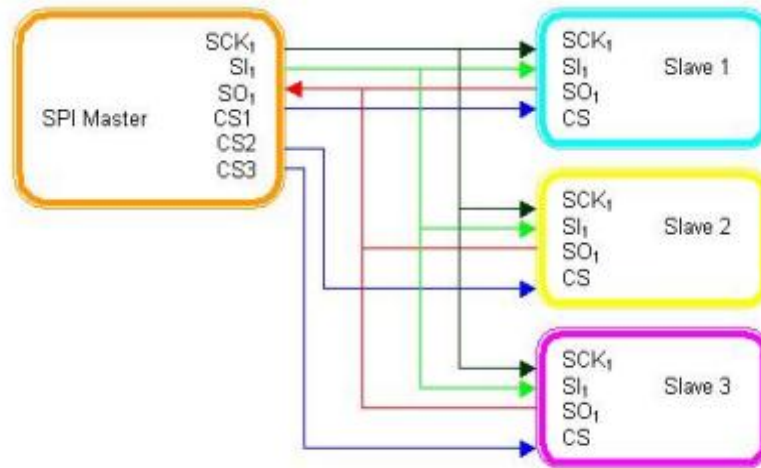


Figure .II.17. Seul maître, esclave seule implémentation SPI

Si les dispositifs esclaves multiples existent, le maître génère un signal de sélection esclave distinct pour chaque esclave. Ces relations sont représentées sur la figure (II-15).



**Figure .II.18. Seul maître, esclaves multiples implémentations SPI.**

Le maître génère esclaves signaux de sélection à l'aide de broches entrées / sorties TOR à usage général ou une autre logique. Il est composé de l'ancien claquement de bits et peut être assez sensible. Vous avez en temps il par rapport aux autres signaux et d'assurer, par exemple, que vous ne basculer une ligne de sélection au milieu d'un cadre. Alors que SPI ne décrit pas d'une manière spécifique à mettre en œuvre des systèmes multi-mâtres, certains périphériques SPI en charge les signaux supplémentaires qui font de telles mises en œuvre possibles. Cependant, il est compliqué et généralement pas nécessaire, il est donc pas fait souvent. Une paire de paramètres appelés horloge polarité (CPOL) et la phase d'horloge (ACSP) déterminer les flancs du signal d'horloge à laquelle les données sont entraînés et échantillonnés. Chacun des deux paramètres a deux états possibles, ce qui permet quatre combinaisons possibles, qui sont tous incompatibles entre eux. Donc, une paire maître / esclave doit utiliser les mêmes valeurs paramètre de paires de communiquer. Si plusieurs esclaves sont utilisés qui sont résolus dans différentes configurations, le maître devra se reconfigurer à chaque fois qu'il a besoin de communiquer avec un esclave différent. SPI ne dispose pas d'un mécanisme d'acquiescement pour confirmer la réception des données. En effet, sans un protocole de communication, le maître SPI n'a pas connaissance du fait que l'esclave existe. SPI offre également aucun contrôle de flux. Si vous avez besoin d'un contrôle de flux matériel, vous pourriez avoir besoin de faire quelque chose en dehors du SPI. Les esclaves peuvent être considérés comme des dispositifs d'entrée / sortie du maître. SPI ne spécifie pas un protocole de niveau supérieur notamment pour dialogue maître-esclave. Dans certaines applications, un protocole de niveau supérieur est pas nécessaire et que les données brutes sont échangés. Un exemple de ceci est une interface à un codec simple. Dans d'autres applications, un protocole de niveau supérieur, comme un protocole de commande de réponse, peut être nécessaire. Notez que le maître doit initier les cadres à la fois pour sa commande et la réponse de l'esclave.

### II.6.2.3. Les données et les contrôles des lignes de la SPI

La SPI a besoin de deux lignes de commande (CS et SCK) et deux lignes de données (SI) et de SO. Avec CS (Chip-Select) le périphérique correspondant est sélectionné. Cette broche est surtout actif bas. Dans l'état non sélectionné les lignes ne sont donc salut-Z et donc inactive. Le maître décide avec quel appareil périphérique il veut communiquer. SCLK la ligne d'horloge est amené à l'appareil si elle est sélectionnée ou non. L'horloge sert de synchronisation de la transmission de données. La majorité des dispositifs SPI fournir ces quatre lignes. Parfois, il arrive que la SDI et SDO soient multiplexés, par exemple dans le LM74 du capteur de température de National Semi-conducteur ou que l'une de ces lignes est manquant. Un périphérique qui doit ou ne peut pas être configuré, ne nécessite pas de ligne d'entrée, seule une sortie de données. Dès qu'il est sélectionné, il commence à envoyer des données. Dans certains CAN donc la ligne SDI est manquant (par exemple MCP3001 de Micro chip). Il existe également des dispositifs qui ont des pas de sortie de données. Par exemple, les contrôleurs LCD (par exemple de la COP 472-3 de National Semi-conducteur), qui peuvent être configurés, mais ne peut pas envoyer des données ou des messages d'état.

### II.6.2.4. SPI Configuration

Parce qu'il n'y a pas de spécification officielle, exactement ce que SPI est et ce pas, il est nécessaire de consulter les fiches techniques des composants que l'on veut utiliser. Important sont les fréquences d'horloge autorisées et le type de transitions valides. Il n'y a pas de règles générales pour les transitions où les données doivent être verrouillées. Bien que non spécifiée par Motorola, dans la pratique, quatre modes sont utilisés. Ces quatre modes sont les combinaisons de CPOL et l'ACSP. Dans le tableau 1, les quatres modes sont repertories.

SPI-mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

Tableau .II.4. SPI Modes.

Si la phase de l'horloge est égale à zéro, à savoir CPHA = 0, les données sont verrouillées sur le front montant de l'horloge avec CPOL = 0, et le front descendant de l'horloge avec CPOL = 1. Si CPHA = 1, les polarités sont renversé. CPOL = 0 signifie descendant, CPOL = 1 front montant. Les microcontrôleurs de Motorola permettent la polarité et la phase de l'horloge à être ajustés. Il en résulte une polarité positive à verrouiller les données sur le front montant de l'horloge. Toutefois, les données est placée sur la ligne de données déjà au front descendant, afin de stabiliser. La plupart des périphériques qui ne peuvent être esclaves, travaillent avec cette configuration. S'il devient nécessaire d'utiliser l'autre polarité, les transitions sont inversées

### II.6.3. SPI vs. I2C

Bien que les deux SPI et I2C fournissent un bon support pour communiquer avec des périphériques lents qui sont accessibles par intermittence, chacun de la voie de communication a ses propres avantages vers l'autre. SPI est mieux adapté que I2C pour les applications qui sont naturellement considérées comme des flux de données (par opposition à la lecture et l'écriture adressée endroits dans un dispositif esclave). Un exemple d'une application "courant" est la communication de données entre des microprocesseurs ou des processeurs de signaux numériques. Une autre est le transfert de données à partir de convertisseurs analogique-numérique. SPI peut également obtenir des taux nettement plus élevés de données qu'I2C qui est limitée à 400 KHz dans la plupart des cas. Interfaces SPI compatibles vont souvent dans les dizaines de mégahertz. SPI obtient réellement l'efficacité dans des applications qui tirent parti de la capacité de transmission bidirectionnelle, telle que la communication entre un (codeur-décodeur) "codec" et un processeur de signal numérique, qui consiste à envoyer simultanément des échantillons avant et arrière.

En raison de l'absence de dispositif SPI intégré dans le traitement, il faut plus d'efforts et plus de ressources matérielles d'I2C lorsque plus d'un esclave est impliqué. L'inconvénient réside ici qu'il est une interface à trois fils et si vous avez plus de 1 appareil, alors vous devez fournir à chaque appareil avec différents chip select broches (CS). Mais SPI a tendance à être plus simple et plus efficace que I2C au point-à-point (un seul maître, esclave simple) des applications pour la même raison; l'absence de dispositif de moyens d'adressage moins de frais généraux. D'autre part, I2C nécessite seulement deux fils pour mettre en œuvre et a une adresse unique de sorte que la relation maître / esclave peut être maintenue comparer à SPI, qui a besoin de trois fils à mettre en œuvre le mode d'adressage. I2C offre également un meilleur support pour la communication avec des dispositifs embarqués qui sont accessibles sur une base occasionnelle. L'avantage concurrentiel de I2C par rapport aux autres systèmes de communication à faible vitesse à courte distance est que son coût et la complexité n'échelle avec le nombre de périphériques sur le bus en raison de la nature générique de l'interface de bus.

En outre, la complexité des justificatifs composants logiciels I2C peut être significativement supérieure à celui de plusieurs systèmes concurrents tels que SPI dans une configuration très simple. Avec son haut-système d'adressage et de simples moyens de transférer des chaînes d'octets, I2C est une solution élégante et minimaliste pour modeste, "à l'intérieur des boîtes" besoins de communication. I2C est aussi un bus multi-maître vrai, car il a la détection de collision et d'arbitrage pour prévenir la corruption de données si deux ou plusieurs maîtres lancer simultanément le transfert de données. En outre, I2C préservé également l'intégrité des données par filtrage rejette pointes sur la ligne de données du bus [16].

<b>SPI</b>	<b>I2C</b>
<p><b>1)</b> Trois lignes de bus sont nécessaires; une ligne d'entrée de données (SI1), une ligne de sortie de données (SO1) et une ligne d'horloge série (SCK 1) [plus 1 Chip Select (CS)].</p> <p><b>2)</b> Aucune spécification officielle (composant dépendant).</p> <p><b>3)</b> des débits de données plus élevés (jusqu'à 10 MHz ou plus).</p> <p><b>4)</b> plus efficace au point-à-point (applications seul maître, esclave simple).</p> <p><b>5)</b> L'absence de dispositif intégré dans la lutte.</p> <p><b>6)</b> Ne pas disposer d'un mécanisme d'accusé de réception pour confirmer la réception des données.</p> <p><b>7)</b> Moins de tête lors de la manipulation application de point-à-point.</p> <p><b>8)</b> mieux adapté pour les applications qui sont naturellement considérés comme des flux de données.</p>	<p><b>1)</b> Deux lignes de bus sont nécessaires; une ligne de données série (SDA) et une ligne d'horloge série (SCL).</p> <p><b>2)</b> Avec la spécification officielle (I2C du protocole créé par Philips).</p> <p><b>3)</b> des vitesses de transfert de soutien de l'ordre de 100 kHz (norme d'origine, ou 400 kHz en utilisant la norme la plus récente).</p> <p><b>4)</b> plus efficace dans les applications multi-maître, multi-esclaves.</p> <p><b>5)</b> Encastré système d'adressage et simple.</p> <p><b>6)</b> Avez-vous un mécanisme d'acquittement pour confirmer la réception des données.</p> <p><b>7)</b> Plus de tête lors de la manipulation application de point-à-point.</p> <p><b>8)</b> mieux pour communication adaptés aux dispositifs embarqués qui sont accessibles sur une base occasionnelle.</p>

## II.7. Conclusion

- Un arduino n'est pas un petit PC.
- Ses domaines de prédilections sont les systèmes embarqués.
- C'est un microcontrôleur simplifié.
- L'idéal pour commencer est l'arduino UNO.
- Il est possible d'ajouter des cartes offrant des options ou connectiques supplémentaires à votre arduino (appelé Shield).
- Ce projet est open source et sous licence libre, il y a donc des cartes ressemblant aux arduino (et ayant souvent « duino » dans leurs noms) mais vendu sous une autre dénomination. Ce ne sont pas des contrefaçons.

# Chapitre III

# Chapitre III

## Modélisation d'un drone quadrotor

### III.1. Introduction

Afin de concevoir un contrôleur de vol, on doit d'abord comprendre profondément les mouvements de l'avion, sa dynamique et par conséquent ses équations dynamiques. Cette compréhension est nécessaire non simplement pour la conception du contrôleur, mais aussi pour s'assurer que les simulations du comportement de véhicule sont plus proche que possible de la réalité lorsque la commande est appliquée.

Le quadrirotor est classé dans la catégorie des systèmes volants les plus complexes vu le nombre d'effets physiques qui affectent sa dynamique à savoir les effets aérodynamiques, la gravité, les effets gyroscopiques, les frottements et le moment d'inertie [17, 18, 19]. Cette complexité résulte essentiellement du fait que l'expression de ces effets diffèrent pour chaque mode de vol. En effet les modèles dynamiques du quadrirotor proposés dans [20]. Changent en fonction des tâches planifiées et en fonction des milieux de navigation définis à priori par l'opérateur.

### III.2. Présentation

#### III.2.1. Qu'est-ce qu'un quadri-rotor ?

Le quadri-rotor est un engin volant doté de quatre rotors placés aux extrémités d'une croix. Ce sont ces quatre rotors qui fournissent la force verticale (portance) qui permet à l'appareil de s'élever. En vol, le quadri-rotor peut évoluer suivant ses axes de lacet, tangage et roulis ainsi qu'en translation dans l'ensemble des directions. Aussi, un tel engin est complètement instable et pour le contrôler, il faut mettre au point un système qui pilote la puissance des moteurs individuellement pour contrer l'inclinaison sur les différents axes. C'est la réalisation de ce système de stabilisation qui est débattue dans ce rapport.

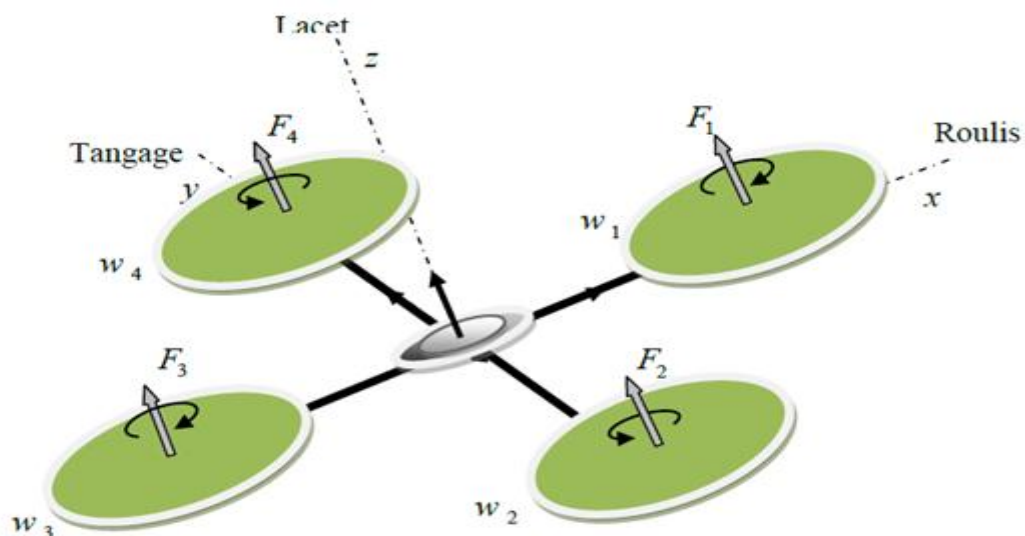


Figure .III.1. Structure générale d'un quadrirotor.

### III.2.2. Comment le piloter ?

Avant de penser pouvoir le piloter, il faut arriver à le stabiliser. Pour cela, les quatre moteurs sont sollicités de la manière suivante

Axe de roulis : en agissant sur les moteurs gauche et droit, on peut maintenir l'horizontalité. Par exemple, si le quadri-rotor penche à gauche, on augmente la vitesse du moteur gauche tout en diminuant le droit.

Axe de tangage : les moteurs avant et arrière permettent de garder une assiette stable sur cet axe. Si le quadri-rotor est orienté vers l'arrière, il faut donner plus de vitesse au moteur arrière et baisser celle du moteur avant.

Ensuite, pour des soucis de commodité, il se doit d'être stabilisé sur son axe de lacet. On réalise cela en agissant sur les moteurs par paires : on augmente la vitesse de rotation de deux moteurs face à face (avant et arrière) et on diminue la vitesse des deux autres (gauche et droite). Pour finir, l'altitude doit être stable et cela est facilement réalisé en jouant sur la vitesse de rotation de l'ensemble des moteurs. On diminue la vitesse des quatre moteurs pour faire descendre le quadri-rotor tandis qu'on l'augmente pour le faire monter.

### III.2.3. Applications possibles

Ce genre de réalisation est très à la mode en ce moment, car il permet de répondre à des problématiques de facilité de pilotage et d'autonomie.

Séduite par ces qualités, la police britannique a fait l'acquisition d'hélico-robots de la même catégorie que le quadri-rotor. Ces drones embarquent une caméra vidéo et un appareil photo pour collecter des preuves si des comportements antisociaux sont détectés.

Aussi, l'armée est très intéressée par ce genre d'engin. Elle organise en effet des concours de drones afin d'explorer les nombreuses capacités de ces appareils. On peut facilement imaginer qu'ils peuvent être utiles pour explorer des zones dangereuses ou inaccessibles par l'homme.

Enfin, le marché du jouet n'est pas à oublier. De nombreux fans de modélisme sont passionnés par le quadri-rotor et s'amuse à en fabriquer ou à en faire l'acquisition auprès des nombreuses sociétés spécialisées qui redistribuent.

### III.3. Modèle dynamique

Le modèle dynamique du quadri-rotor peut être vu comme un système où les évolutions spatiales du quadri-rotor sont les sorties et la tension de chacun de ses moteurs sont les entrées.

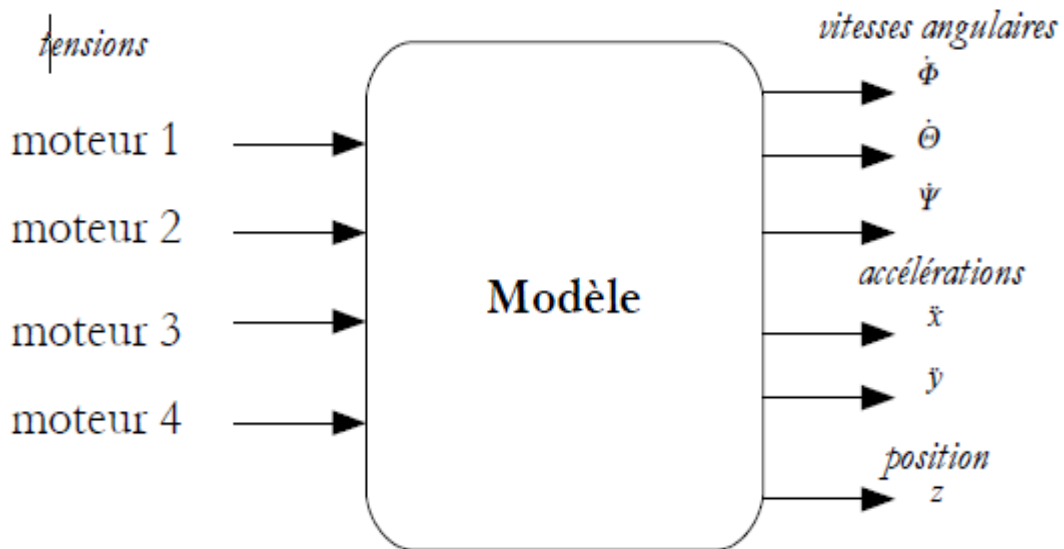


Figure .III.2. Configuration du quadrotor [18].

Le modèle se divise naturellement en coordonnées de translation et de rotation:

$$\zeta = (x, y, z) \in \mathbb{R}^3; \quad \eta = (\theta, \varphi, \psi) \in S^3 \quad (\text{III.1})$$

:

$$\begin{cases} \ddot{x} = -\frac{u}{m} \sin \theta \\ \ddot{y} = \frac{u}{m} \cos \theta \cdot \sin \varphi \\ \ddot{z} = \frac{u}{m} \cos \theta \cdot \cos \varphi - g \\ \ddot{\theta} = \tilde{\tau}_\theta \\ \ddot{\varphi} = \tilde{\tau}_\varphi \\ \ddot{\psi} = \tilde{\tau}_\psi \end{cases} \quad (\text{III.2})$$

En peut remarquez que les donner fournie par IMU sont indispensable dans model dynamique et ne peut pas faire réalisation ce système son bien connaître IMU parce que est le cœur de ce system, la commande et filtrage et stabilité sont tous base sur les donner fourni par IMU.

### III.4. les téléphones intelligents

Les téléphones et tablettes Android basés sur le processeur Intel Atom peuvent prendre en charge une grande gamme de capteurs matériels. Ces capteurs permettent de détecter le mouvement et les changements de position et de signaler les paramètres ambiants. Le schéma fonctionnel de la

Figure 1 montre une configuration de capteur possible sur un appareil Android typique basé sur un processeur Intel Atom. [22]

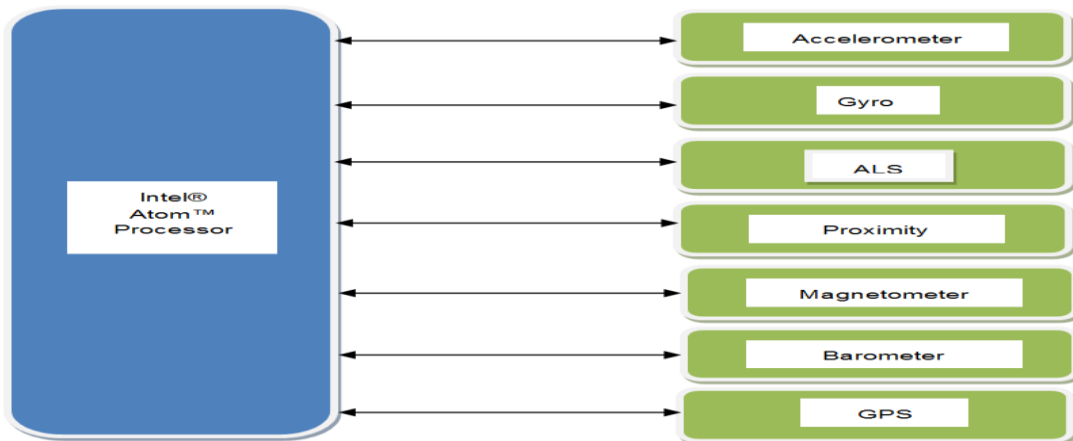


Figure .III.3. Capteurs sur un système Android basé sur Intel® Atom™

Nous pouvons classer les capteurs Android en différents types et classes, indiqués dans le Tableau 1, en fonction des données qu'ils fournissent.

Capteurs de mouvement	Accéléromètre (TYPE_ACCELEROMETER)	Mesure l'accélération d'un appareil en m/s <sup>2</sup>	Détection de mouvement
	Gyroscope (TYPE_GYROSCOPE)	Mesure la vitesse de rotation d'un appareil	Détection de la rotation
Capteurs de position	Magnétomètre (TYPE_MAGNETIC_FIELD)	Mesure l'intensité du champ géomagnétique terrestre en $\mu$ T	Boussole
	GPS (pas un type android.hardware.Sensor)	Obtient la géolocalisation précise de l'appareil	Détection précise de la géolocalisation
	ALS (TYPE_LIGHT)	Mesure le niveau de lumière ambiante en lx	Contrôle automatique de la luminosité de l'écran
Capteurs environnementaux	Baromètre	Mesure la pression de l'air ambiant en mbar	Détection de l'altitude

Tableau III.1. Types de capteurs pris en charge par la plate-forme Android

### III.4.1. Système de coordonnées des capteurs

La structure de capteurs indique les données des capteurs en utilisant un système de coordonnées standard à 3 axes, où X, Y et Z sont représentés par values[0], values[1] et values[2] dans l'objet SensorEvent, respectivement.

Certains capteurs, comme les capteurs de lumière, de température, de proximité et de pression, retournent uniquement des valeurs uniques. Pour ces capteurs, seule la valeur values[0] de l'objet SensorEvent est utilisée.[22]

Les autres capteurs indiquent les données en utilisant le système de coordonnées standard à 3 axes des capteurs. Vous trouverez ci-dessous une liste de ces capteurs :

- Accéléromètre
- Capteur de gravité
- Gyroscope
- Capteur de champ géomagnétique

Le système de coordonnées à 3 axes des capteurs est défini en fonction de l'orientation naturelle (par défaut) de l'écran de l'appareil. Pour un téléphone, l'orientation par défaut est Portrait, pour une tablette, l'orientation naturelle est Paysage. Lorsqu'un appareil est tenu dans son orientation naturelle, l'axe x est horizontal et pointe vers la droite, l'axe y est vertical et pointe vers le haut, et l'axe z pointe vers l'avant de l'écran. La Figure 4 montre le système de coordonnées des capteurs d'un téléphone et la Figure 5 celui d'une tablette.

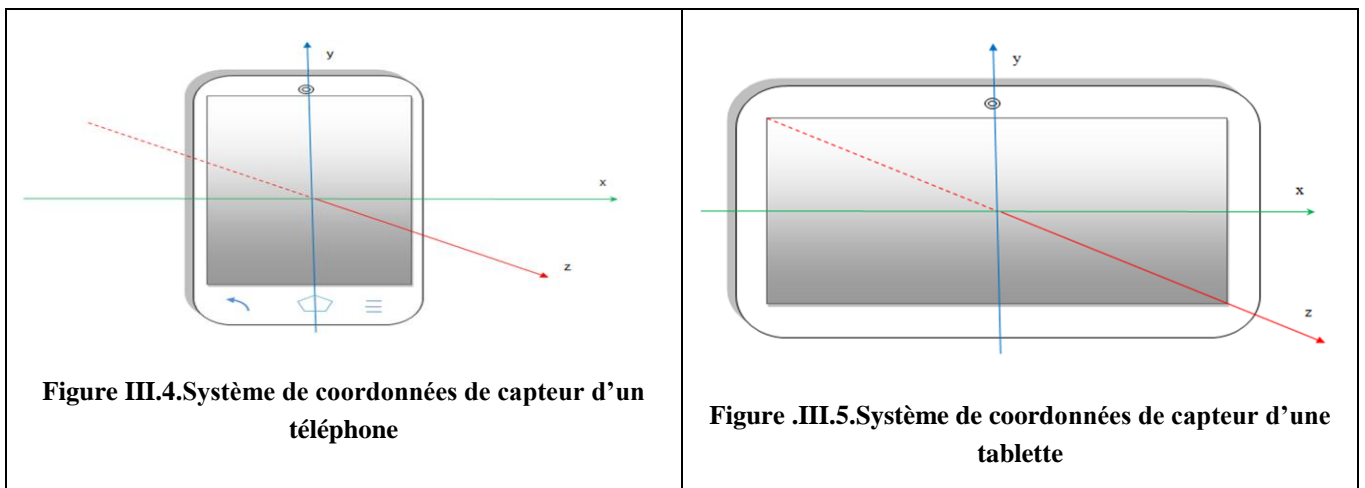


Figure III.4.Système de coordonnées de capteur d'un téléphone

Figure .III.5.Système de coordonnées de capteur d'une tablette

Le point le plus important à retenir concernant le système de coordonnées de capteur est que le système de coordonnées du capteur ne change jamais lorsque l'appareil bouge ou que son orientation change.

### III.5.conclusion

Le principe de fonctionnement en terme de navigation d'une centrale inertie est l'estime. En effet le système fournit la position de la carte avec une bonne précision même après plus heur avec pour celle recalage la position précise du parking point de départ de la navigation.

**Conclusion générale  
Et  
Perspective**

## Conclusion Générale

---

Au cours de mon travail de mémoire, j'ai eu l'occasion de découvrir la richesse des problèmes posés à l'automaticien pour le contrôle d'un robot leur instabilité naturelle, leur caractère fortement non linéaire, le fort couplage entre les axes et surtout, cette présence incontournable du vent.

Le titre choisi, décrit bien le cheminement réalisé pour passer d'un modèle d'aéromodélisme à une plate-forme contrôlée en position capable de servir d'œil déporté d'un opérateur. De mon point de vue, l'intérêt majeur que j'ai trouvé à ce travail de recherche a été la multidisciplinarité : toutes les facettes traditionnelles de l'Automatique ont ainsi été abordées l'identification, la modélisation, la commande, le traitement du signal et la navigation autonome.

Ce travail m'a conduit à défricher plusieurs domaines, au lieu d'en explorer un à fond comme c'est le cas pour certaines mémoires. Après tout, une caractérisation aérodynamique complète du (imu) eût suffi à elle seule à motiver plusieurs mémoires. Cependant, apporter une solution complète à un problème de robotique nécessite de considérer simultanément des problèmes de perception, de décision et de commande. C'est pourquoi les sujets traités sont multiples et s'articulent tous autour d'un fil conducteur omniprésent : l'expérimentation sur le quadricoptère.

## Perspective

---

A l'issue de ces travaux, ce mémoire ouvre de nouvelles perspectives de recherche parmi lesquelles nous citons :

- La généralisation d'étude des lois de commande non linéaire proposées pour le cas discret.
- La proposition des algorithmes d'optimisations pour la détermination des différents paramètres et gains de la loi de commande non linéaire.

# **Références bibliographiques**

## Références bibliographiques

---

- [1] N. Hachelef, P. Lavoie, D. Li, R. Landry, «Experiments of low-cost INS/GPS Navigation platform based on PC104», Department of Electrical Engineering École de Technologie Supérieure (ETS), LACIME Laboratory, Canada, 2004
- [2] Basic Principles of Inertial Navigation, Seminar on inertial navigation systems, Tampere University of Technology.
- [3] W. Durfee, «Arduino Microcontroller Guide » , University of Minnesota, oct-2011.
- [4] Nastro V. (2004), Navigazione inerziale ed integrata, Guida Editore, 2004.
- [5] Batu Ferringhi ,Proceedings of the International Conference on Man-Machine Systems (ICoMMS), Penang, MALAYSIA, 11 – 13 October 2009,
- [6] Oliver J. Woodman , , «An introduction to inertial navigation. » , Cambridge CB3 0FD United Kingdom, August 2007
- [7] ZEGHLACHE Samir, «Commande non linéaire d'un appareil à vol vertical», Université de M'sila , 23/09/2014.
- [8] D. Titterton and J. Weston. Strapdown Inertial Navigation Technology. The American Institute of Aeronautics and Astronautics, second edition, 2004.
- [9] Alan G. Smith , «Introduction to Arduino» , September 30, 2011.
- [10] Peter Alley, « Introductory Microcontroller Programming» , WORCESTER POLYTECHNIC INSTITUTE , May 2011.
- [11] Walt Kester , « The Data Conversion Handbook. Elsevier:» , Newnes, 2005.
- [12] John A. Chandy , « Microcontroller Applications in Engineering » university of connectcut, January 24, 2008.
- [13] Bartmann, Erik, « Le grand livre d'Arduino» Editeur, Eyrolles, 2015.
- [14] Philips Semiconductors, , « I2C Bus, San Jose » , March 24, 2003
- [15] Stevens, Roger “Serial Communications Using PIC® Micro-Controllers”, 1999.
- [16] The I2C-Bus Specification (Version 2.1), Philips Semiconductor, January 2000.
- [17] M. Feki, « An adaptive feedback control of linearisable chaotic systems, »Chaos, Solitons and Fractals, vol. 15, no.5, pp. 883-890, 2003.
- [18] E.m. Jafrov, M.N.A. Parlakcl, and Y. Istefanopulos, « A new variable structure PID-controller design for robot manipulators,» IEEE. Trans. On Contr. Syst. Tech., vol. 13, no. 1, pp. 122-130, 2005.

- [19] Boubakir, S. Labiod, and T.M. Guerra, « Commande PID Adaptative des Systèmes non linéaires Affines en la Commande », CGE'06, EMP, Algérie, 13-14 Avril, 2009.
- [20] P. Y. GLORONNEC " Algorithme d'apprentissage pour systèmes d'inférences floues " - INSA de Rennes (IRISIA)- 1999.
- [21] <http://www.patentstorm.us/patents/5067084/description.html> Description of IMU aiding from Roll isolated Gyro
- [22] Inertial Navigation: 40 Years of Evolution - Overview at <http://www.imar-navigation.de>  
[www.imar-navigation.de](http://www.imar-navigation.de)

**MEMOIRE DE FIN D'ETUDE EN VUE DE L'OBTENTION DU DIPLOME  
MASTER EN AUTOMATIQUE**

**SPECIALITE : AUTOMATIQUE**

**Proposé et dirigé par : KHODJA Mohammed Abdallah**

**Présenté par : RAHMANI Nadir**

## **Thème**

# **Réalisation d'un IMU (Inertial Measurement Unit)"**

### **Résumé**

Le travail à réaliser dans cet mémoire concerne l'étude de mesure inertielle, Utiliser l'ordinateur et les détecteurs de mouvement (accéléromètre) et les détecteurs de rotation (gyroscope) afin de tenir compte sur une base continue traversez évaluer la situation, la Direction et la vitesse ( Direction et la vitesse de mouvement) d'un objet en mouvement sans la nécessité de compter les références externes, elle est munie de données de position et de vitesse à partir de nombreuses sources, Ce dans le premier chapitre, Le deuxième chapitre traite du rôle d'arduino qui doit se retirer les données de position et la vitesse spécifique grâce à l'intégration de l'information qui a été reçue par les capteurs de mouvement.et le troisième chapitre nous avons mis la modélisation d'un quadricopter à partir d'n PID seulement à l'absence de possibilités de fabrication de ce plan

### **Mots clés**

inertielle, arduino, modélisation, accéléromètre, gyroscope.

**N° d'ordre : AUT\_195**