

Université Mohammed V
Faculté des Sciences
Rabat

Projet de fin d'études
Département de physique
Filière: Licence Fondamentale de Physique, Parcours I, S6
Titre Du Projet

Conception Et Mise En Oeuvre D'un Système De Télémétrie 3D

Réalisé Par:

- **HLANI Adnane**
- **CHETOUANI Yassin**

Encadrées Par: Pr. OUADOU Mohamed

Membres Du jury

Ouadou Mohamed

Minaoui Khalid

Meddad Majdouline

Année Universitaire: 2018-2019

Remerciements

C'est avec un grand plaisir que nous réservons ces lignes en signe de reconnaissance à tous ceux qui ont de près ou de loin contribué à rendre ce travail possible. Même si des fois nos expressions nous trahissent et n'expriment pas réellement nos sentiments les plus sincères.

Notre profonde gratitude va en premier, à Allah le tout puissant, le miséricordieux de m'avoir donné la force et la patience d'arriver à mettre ce travail à terme.

Puis; On voudrais remercier notre encadreur Dr. MOHAMMED OUADOU; pour les efforts qu'il a déployé et le temps qui nous a donné, pour nous aider, conseiller, encourager et corriger, qu'il trouve ici toute notre gratitude et surtout notre parfaite considération de l'intérêt qu'il portait à ce travail depuis notre proposition de ce sujet.

Nous remercions très sincèrement, les membres de jury d'avoir bien voulu accepter de faire partie de la commission d'examination.

A toute personne qui a participé de près ou de loin pour l'accomplissement de ce modeste travail.

Yassin Chetouani & Adnane Hiani

Dédicaces

Avant tout, nous tenons à remercier le bon dieu, et l'unique qui nous offre le courage et la volonté nécessaire pour affronter les différentes difficultés de la vie, nous dédions ce modeste travail:

À nos parents.

À nos frères et sœurs.

À toute la famille.

À nos amis et collègues de la promotion 2019 « SMP:1&A ».

À tous ceux que nous aimons et nous sont très chers.

Résumé

La télémétrie est Procédé technique permettant de mesurer la distance d'un objet lointain par utilisation d'éléments optiques (télémètre laser, pour la télémétrie laser sur satellites par exemple), acoustiques (sonar) ou radioélectriques (radar).

Dans le cadre de notre projet de fin d'étude en IEA (Informatique, Electronique et Automatique) nous avons eu l'occasion de réaliser un projet dans le but de réaliser un système embarqué de télémétrie qui permet de donner un représentation 3D de l'espace.

Comme nous avons déjà mentionné le but de ce projet est le réalisation d'un petit système embarqué qui permet de faire une représentation 3D de l'espace, dont on a besoin de plusieurs dispositifs qui vont bien nous aider pour la réalisation du projet, carte arduino uno (dispositif programmable qui gère le fonctionnement des autres périphériques qui constitue le système), Lidar (capteur optique utile pour extraire la distance des obstacles) et servomoteurs (assurent la dynamique du système et servent pour extraire les angles).

La réalisation de ce projet, nécessite plusieurs étapes. Premièrement, une présentation brève de la télémétrie et la détection, les systèmes embarqués ainsi les différents méthodes pour la réalisation du système; en second temps, présenter une étude générale des différentes composantes du systèmes, que se soit principe de fonctionnement ou éléments internes. Après cette étude on passe a la realisation pratique du système qui va construire la dernière partie du projet.

Afin de finir la réalisation du projet, on finit par sortir avec une conclusion générale et donner des perspectives du projet pour améliorer le comportement du projet.

Sommaire

Remerciements	2
Dédicaces	3
Résumé	4
Liste des figures	8
Liste des abréviations	9
Introduction Générale	10
Chapitre 1 Introduction à la télémétrie	11
Introduction	11
Définition de la télémétrie	11
Définition d'un système de télémétrie	11
Détection et télémétrie	12
Technologies Principaux De Télédétection	13
LIDAR	13
Point Historique	13
Principe de fonctionnement d'un capteur optique	14
Principe general de Lidar	14
L'étonnante diversité des Lidars	16
Diversité des applications	16
Diversité des domaines d'applications	16
Diversité d'émission	17
Diversité de taille	17
Diversité des types de lasers utilisés	17
Les différents types de Lidar et les modes de fonctionnement	17
LiDAR à cible dure	17
LiDAR à cible diffuse – lidar atmosphérique	19
Intérêts, avantages et inconvénients du lidar	22
RADAR	23
Point Historique	23
Principe de fonctionnement	23
SONAR	25
Point Historique	25

Structure des systèmes sonar	25
Les principales utilisations	27
Chapitre 2 Introduction au systèmes embarqués	28
Introduction	28
Caractéristiques des systèmes embarqués	28
Complexité des systèmes embarqués	29
Définition des systèmes embarqués	29
Classification des systèmes embarqués	31
Système Transformationnel	31
Système Interactif	31
Système Réactif ou Temps Réel	31
Les Applications	31
Arduino	31
Qu'est ce que c'est arduino?	31
Types des cartes Arduino	32
Cartes Arduino Officielles (Classique)	32
Cartes Arduino Compatibles (Dérivées)	32
Avantage de la carte Arduino UNO: Pourquoi Arduino?	33
Caractéristiques techniques de la carte arduino Uno:	34
Microcontrôleur ATmega328	34
C'est quoi la différence des cartes officielles "édition SMD/CMS" ?	35
Présentation Physique	36
Descriptions des broches	37
Les sources de l'alimentation de la carte	38
Les broches d'alimentation	38
Les entrées et sorties	39
Les ports de communications	41
Bus I2C	41
Le logiciel ARDUINO : Espace de Développement Intégré (EDI)	42
Principe général d'utilisation	43
Le terminal série	44
Exemple d'application: Contrôle d'un servo moteur avec arduino	44
Fonctionnement	44
Caractéristiques	47

Chapitre 3 Conception matérielle et logicielle du projet	50
Introduction	50
Cahier de charges	50
Introduction	50
Le projet	50
Électronique	50
Arduino	51
L'interface	51
Structure du système	52
Concept et prototypes	52
Défis	52
Solutions	53
Matériel	53
Schémas et code Arduino	57
Organigramme Du Projet	58
Logiciel	59
Résultats	64
Conclusion Générale	66
Annexe	68
Références	72

Liste des figures

Figure 1:	Taxonomie des technologies de télédétection
Figure 2:	Schéma fonctionnel d'un capteur optique
Figure 3:	Schéma d'un système LiDAR et du principe de la mesure
Figure 4:	Schéma de principe d'un télémètre à détection directe
Figure 5:	Schéma Principe de la mesure de distance par triangulation
Figure 6:	Schéma général du LiDAR à détection cohérente
Figure 7:	Schéma général du LiDAR à rétrodiffusion
Figure 8:	Schéma fluorescence caractérisée par l'émission d'un photon
Figure 9:	Principe du LiDAR DIAL
Figure 10:	Principe de fonctionnement d'un système RADAR
Figure 11:	Émission pulsée
Figure 12:	Émissions continues
Figure 13:	Structure générique d'un système sonar.
figure 14:	Les éléments constitutants d'un SONAR
Figure 15:	Système Électronique Embarqué
Figure 16:	Carte Arduino officielle
Figure 17:	Carte Arduino Compatible
Figure 18:	Microcontrôleur ATmega328
Figure 19:	Synoptique générale d'un ATMEGA.
Figure 20:	Brochage typique d'un ATMEGA32 en boîtier PDIP et PLCC
Figure 21:	Sources de l'Alimentation de la carte Arduino UNO.
Figure 22:	Les différents broches entrées sortie d'arduino uno
Figure 23:	Constitution de la carte Arduino UNO
Figure 24:	Interface de logiciel ARDUINO IDE
Figure 25:	Interface du terminal Série
Figure 26:	Synoptique de fonctionnement de l'asservissement du servomoteur
Figure 27:	Position en fonction de la pulsation
Figure 28:	Vue interne d'un servomoteur
Figure 29:	Signal de commande par PWM
Figure 30:	Exemple du contrôle du moteur servo par une carte Arduino
Figure 31:	Structure du système de détection
Figure 32:	Les supports qui les maintiennent LIDAR et les deux servos
Figure 33:	Branchement de tous les éléments du montage
Figure 34:	Interface du logiciel
Figure 35:	Menu de sélection de la carte cible
Figure 36:	Menu de sélection du port com
Figure 37:	Commande qui permet d'enregistrer les mesures
Figure 38:	Forme du fichier enregistré
Figure 39:	Représentation des mesures dans le plan du plateforme
Figure 40:	Exemplaire du nuage de points 3D obtenus pendant le scan
Figure 41:	Le prototype réalisé

Liste des abréviations

SoC	S ystem o n a C hip
RF	R adio F réquence
MEMS	M icro E lectro M echanical S ystem
IDE	I ntegrated D evelopment E nvironment
RS232	R ecommended S tandard 232
PSR	P rimary S urveillance R adar
PWM	P ulse W idth M odulation
I2C	I nter I ntegrated C ircuit
UNO	ONE (in Italy)
SMD	S urface M ount D evice
USB	U niversal A synchronous R eceiver T ransmitter
RAM	R andom- A ccess M emory
EEPROM	E lectrically- E rasable P rogrammable R ead- O nly M emory
SRAM	S tatic R andom- A ccess M emory
AVR	A utomatic V oltage R egulator
ALU	A rithmetic- L ogic U nit
UART	U niversal A synchronous R eceiver/ T ransmitter
SDA	S erial D ata L ine
SCL	S erial C lock L ine
SPI	I nterface S érie P ériphérique
I2C	T wo- W ire I nterface
Gnd	G round
RX	R ecieve/ R eceiver/ R eception
TTX	T ransmission
TTL	T ime T o L ive
SCK	S erial C lock
SONAR	S ound N avigation A nd R anging
RADAR	R adio D etection A nd R anging
LIDAR	L ight D etection A nd R anging

Introduction Générale

Un système embarqué est un système complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires. Le système matériel et l'application (logiciel) sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau PC(Personal Computer).

Un système embarqué est autonome et ne possède pas des entrées/sorties standards tels qu'un clavier ou un écran d'ordinateur. Contrairement à un PC, l'interface IHM (Interface Homme machine) d'un système embarqué peut être aussi simple qu'une diode électroluminescente LED(Light Emitter Diode) qui clignote ou aussi complexe qu'un système de vision de nuit en temps réel ; les afficheurs à cristaux liquides LCD (Liquid Crystal Display) de structure généralement simple sont couramment utilisés. Afin d'optimiser les performances et la fiabilité de ces systèmes, des circuits numériques programmables FPGA (Field Programmable Gate Array), des circuits dédiés à des applications spécifiques ASIC (Application Specific Integrated Circuits) ou des modules analogiques sont en plus utilisés. Le logiciel a une fonctionnalité fixe à exécuter qui est spécifique à une application. L'utilisateur n'a pas la possibilité de modifier les programmes. Bien souvent, il n'a pas conscience d'utiliser un système à base des microprocesseurs.

Les systèmes embarqués sont désormais utilisés dans des applications diverses tels que le transport (avionique, espace, automobile, ferroviaire), dans les appareils électriques et électroniques (appareils photo, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables), dans la distribution d'énergie, dans l'automatisation, ..., etc.

Ces systèmes représentent le premier marché de l'électronique. Ils sont interconnectés et agissent à travers des réseaux, dans un environnement où la vitesse et la quantité de données échangées croissent rapidement. Les enjeux dans ce domaine se situent dans la maîtrise de la complexité, la diminution de coûts de production du logiciel, la performance et la robustesse des architectures matérielles et logicielles.

Chapitre 1

Introduction à la télémétrie

I. Introduction

Dans ce 1^{er} chapitre nous allons essayer de donner une idée générale sur le monde de la télémétrie. Nous allons commencer par définir la télémétrie, ensuite nous répondrons à la question c'est quoi un système de télémétrie ? Et nous finissons par citer les applications actuelles de télémétrie.

II. Définition de la télémétrie

Le mot télémétrie est dérivé des racines grecques télé (à distance) et metron (mesure), Bien que le terme télémétrie soit généralement utilisé pour les systèmes distants sans câbles, dans certaines bibliographies, on peut également le trouver pour les systèmes de transmission câblés. La télémétrie comprend un ensemble de procédures pour mesurer les grandeurs physique et chimique d'une position éloignée à l'endroit où les phénomènes que nous voulons analyser se produisent et en plus, couvre l'envoi ultérieur d'informations à l'opérateur du système.

A. Définition d'un système de télémétrie

Un système de télémétrie est généralement constitué d'un transducteur en tant que périphérique d'entrée, un support de transmission sous la forme de lignes de câble ou d'ondes radio, dispositifs de traitement du signal, et dispositifs d'enregistrement ou affichage des données.

Le périphérique d'entrée peut être distingué comme un ensemble de deux parties Fondamentals:

- Le capteur, qui est l'élément sensible primaire qui répond aux variations d'état des grandeurs physiques de l'étude, et le transducteur, qui est responsable de la conversion de la valeur de la température, de la pression ou des vibrations dans le signal électrique correspondant.
- Le support de transmission peut être établi de manière guidée par des moyens tels que réseaux téléphoniques classiques les réseaux informatiques ou les liaisons par fibre optique, ou de manière non guidée, par ondes radio, par Bluetooth ou par communication Wifi ou même par les réseaux de téléphonie mobile.
- Le dispositif de traitement de signal est composé d'un serveur responsable de l'analyse et de la transformation des données, le cas échéant, pour stocker toutes les informations dans une base de données interne de l'ordinateur lui-même.
- Pour la visualisation des données, nous utilisons une application capable d'afficher les valeurs collectées

III. Détection et télémétrie

La télédétection est une technique qui consiste à acquérir de l'information sur un objet défini se situant à proximité d'un capteur, et ce, sans aucun contact physique. On y a recours pour d'innombrables applications, notamment l'aide à la conduite automobile, le guidage de robots, la gestion de la circulation, la détection de niveaux de liquides, pour n'en nommer que quelques-unes.

De nombreuses technologies se prêtent à la télédétection. Elles peuvent être divisées en trois grandes catégories:

- **La détection de présence ou de proximité**, qui constitue la plus simple forme de télédétection, permet de détecter l'absence ou la présence d'un objet dans une zone générale là où il s'agit de l'unique renseignement requis, par exemple, pour des applications de sécurité.
- **La mesure de la vitesse**, qui est utilisée lorsque la position exacte d'un objet n'est pas nécessaire, permet de déterminer la vitesse précise d'un corps en mouvement, par exemple, à des fins d'application de la loi.
- **La détection et la télémétrie**, qui permet de déterminer avec précision la position exacte d'un objet par rapport au capteur.

Le présent document technique se concentre sur les technologies dotées d'une fonction de détection et de télémétrie, car il s'agit de la plus complexe des trois énoncées ci-dessus. Comme les données de position permettent de récupérer des informations sur la présence et la vitesse d'un objet, la détection et la télémétrie sont universellement applicables à toutes les applications de télédétection.

Bien que certaines technologies passives telles que la stéréotriangulation d'images de caméra soient capables de fournir des renseignements sur la distance d'un objet, leurs capacités sont généralement très limitées. Par exemple, pour réaliser une stéréotriangulation, les points de mesure doivent être bien définis, sans quoi les algorithmes correspondants ne fonctionnent pas. Ainsi, pour mesurer la position d'un ou plusieurs objets, on a le plus souvent recours à des technologies qui transmettent de l'énergie vers la matière à mesurer afin d'en analyser le retour d'écho et, ainsi, d'en déterminer la position. Comme l'énergie est intentionnellement émise vers l'objet à mesurer, on dira de ces technologies qu'elles sont « actives ».

Parmi celles-ci, certaines utilisent le lieu géométrique de l'écho pour déduire les données de positionnement de l'objet mesuré, notamment la technique de lumière structurée, qui consiste à projeter une multitude de points lumineux vers l'objet, puis à analyser leur dispersion géométrique en fonction des images obtenues à l'aide d'une caméra.

D'autres technologies, quant à elles, déterminent la position de l'objet à mesurer d'après le caractère temporel de l'écho. Dans la plupart des cas, il s'agit de technologies de mesure par temps de vol.

Même si divers procédés de mesure par temps de vol existent, cette méthode repose généralement sur l'utilisation d'ondes radioélectriques (radar), acoustiques ou ultrasoniques (sonar), ou lumineuses (lidar).

L'image de la figure 1 illustre les différentes technologies de télédétection disponibles à l'heure actuelle. Dans la partie suivante, les technologies de mesure optiques par temps de vol seront approfondies.

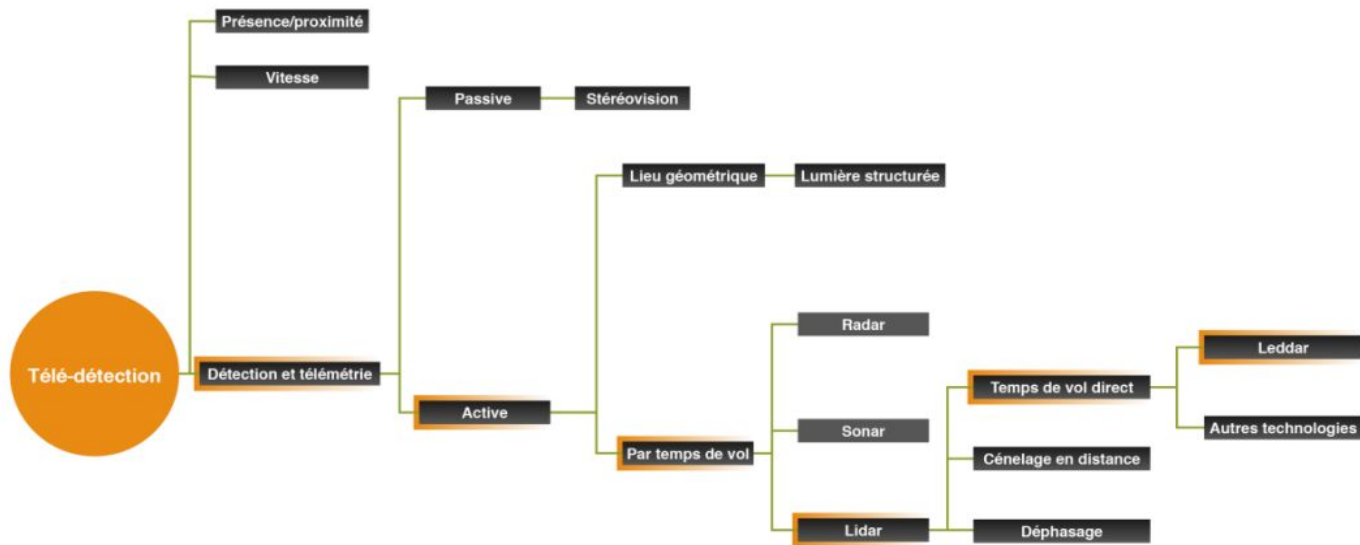


Figure 1: Taxonomie des technologies de télédétection

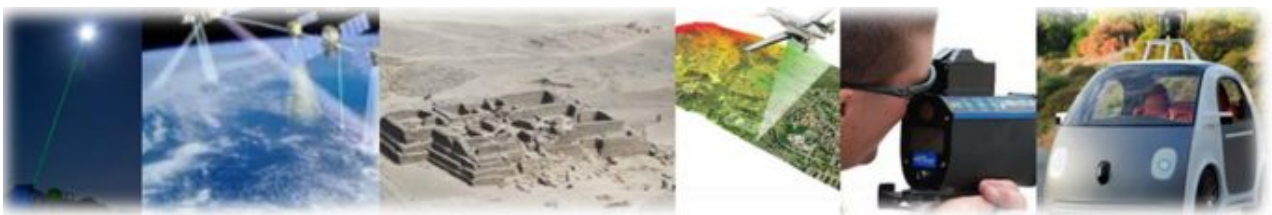
IV. Technologies Principaux De Télédétection

A. LIDAR

1. Point Historique

Au travers de l'écholocalisation, la chauve-souris est la première forme de « LiDAR ». Le moyen de perception de l'environnement par la chauve-souris repose sur un principe d'émission/réception d'une onde sonore.

Le LiDAR, quant à lui, fonctionne sur un principe similaire mais avec une onde lumineuse. Le développement de la technologie du LiDAR a débuté grâce à l'invention du laser en 1960 notamment le laser «Q-switched» en 1962 et le laser à Rubis en 1963. Environ 10 ans après l'invention du laser, la technologie LiDAR a commencé à se développer.



En 1976, ED. Hinkley publie «Laser Monitoring of the Atmosphere», premier livre abordant la thématique du LiDAR et témoignant du développement de cette technologie. La technologie LiDAR a permis de réaliser de nombreuses avancées dans le domaine scientifique. La première utilisation importante du LiDAR fut le calcul de la distance Terre-Lune pour le projet Luna See en 1962. Le LiDAR est ensuite principalement utilisé

dans le domaine de la météorologie, ainsi en 1963 Goyer et Watson utilisent le LiDAR pour calculer la hauteur des nuages. Le LiDAR a aussi permis de cartographier la Lune en 1971 lors de la mission Apollo 15. Par la suite, son utilisation dans le domaine militaire et spatial s'est développé avec l'apparition du GPS. Aujourd'hui des calculs de distances sont toujours utilisés par la technologie LiDAR et ne cessent de se développer et de s'améliorer en particulier dans les domaines de la robotique et de l'automobile.

2. Principe de fonctionnement d'un capteur optique

Un capteur optique analogique est conçu pour recueillir et focaliser un rayonnement lumineux (partie optique) et pour sensibiliser une surface photosensible (partie chimique). Un capteur optronique est conçu pour recueillir et focaliser un rayonnement lumineux (partie optique) et pour la convertir en un signal électrique (partie électronique). Le principe de fonctionnement d'un capteur optronique est décrit par le diagramme ci-après.

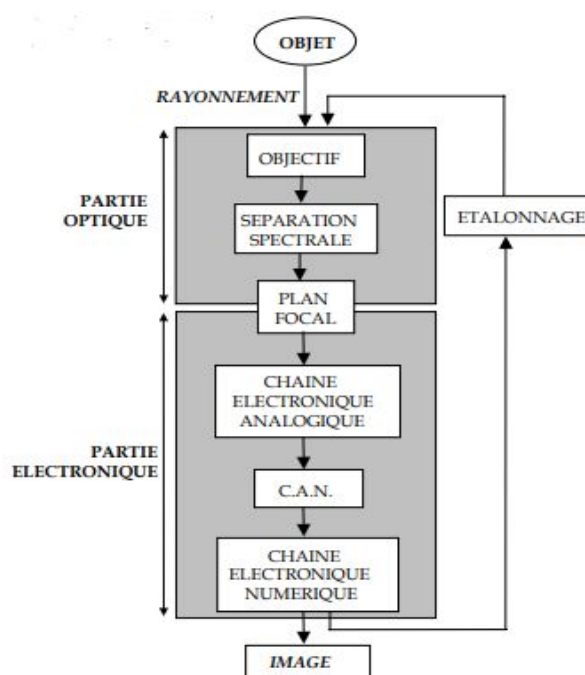


Figure 2: Schéma fonctionnel d'un capteur optique

- ★ L'image est une information bidimensionnelle : son acquisition exige donc un balayage approprié du paysage qui dépend de la géométrie du plan focal.

3. Principe general de Lidar

Le LiDAR, acronyme de l'expression anglaise « Light Detection And Ranging» (Détection de la lumière et mesure à distance) est une technique de Mesure Électronique de Distance (MED) permettant de mesurer la distance entre la surface d'un objet et l'appareil (émetteur/récepteur) grâce aux propriétés des faisceaux de lumière.

Le principe du LiDAR est voisin de celui du radar. Tous deux envoient une impulsion de radiation électromagnétique sur une cible ou un obstacle. L'onde est réfléchi vers la source émettrice. En connaissant la vitesse de propagation de l'onde émise, on est ainsi capable de calculer la distance qui sépare la cible de l'émetteur.

À la différence du radar qui fonctionne dans le domaine des micro-ondes (fréquences de 1 à 100 GHz) ou du sonar qui utilise des ondes sonores, le LiDAR utilise de la lumière et plus précisément des lasers rayonnant dans le domaine infrarouge, le domaine visible ou le domaine ultraviolet (fréquences supérieures à 10 THz).

De plus, le LiDAR, puisqu'il fonctionne à partir d'un laser, travaille avec des longueurs d'ondes optiques, c'est-à-dire des longueurs d'ondes qui vont de quelques centaines de nanomètres à quelques microns. L'onde émise est donc cohérente et polarisée.

Rappel: Deux sources sont cohérentes si elles émettent des ondes sinusoïdales de même fréquence et si le retard de l'une par rapport à l'autre ne varie pas au cours du temps. La lumière est une onde électromagnétique. Les ondes les plus simples qui décrivent la lumière sont les ondes polarisées linéairement.

Le principe de la mesure LiDAR requiert généralement l'utilisation d'un laser impulsif. A l'inverse d'un laser conventionnel, qui produit habituellement un rayonnement continu, le laser impulsif produit des flashes de lumière régulièrement espacés dans le temps et très brefs, que l'on appelle impulsions. Chaque impulsion dure de quelques femtosecondes (fs) à 100 fs.

Un LiDAR est composé principalement de 2 éléments : un émetteur laser (fonction d'émission) qui envoie un signal lumineux et un télescope (fonction de réception) comprenant un collecteur de lumière qui reçoit la lumière réfléchi. Le signal reçu doit être analysé, ainsi on trouve après le télescope un système électro-optique d'acquisition : un photo-détecteur qui transforme la lumière en signal électrique ainsi qu'une chaîne électronique qui traite ce signal pour obtenir l'information recherchée.

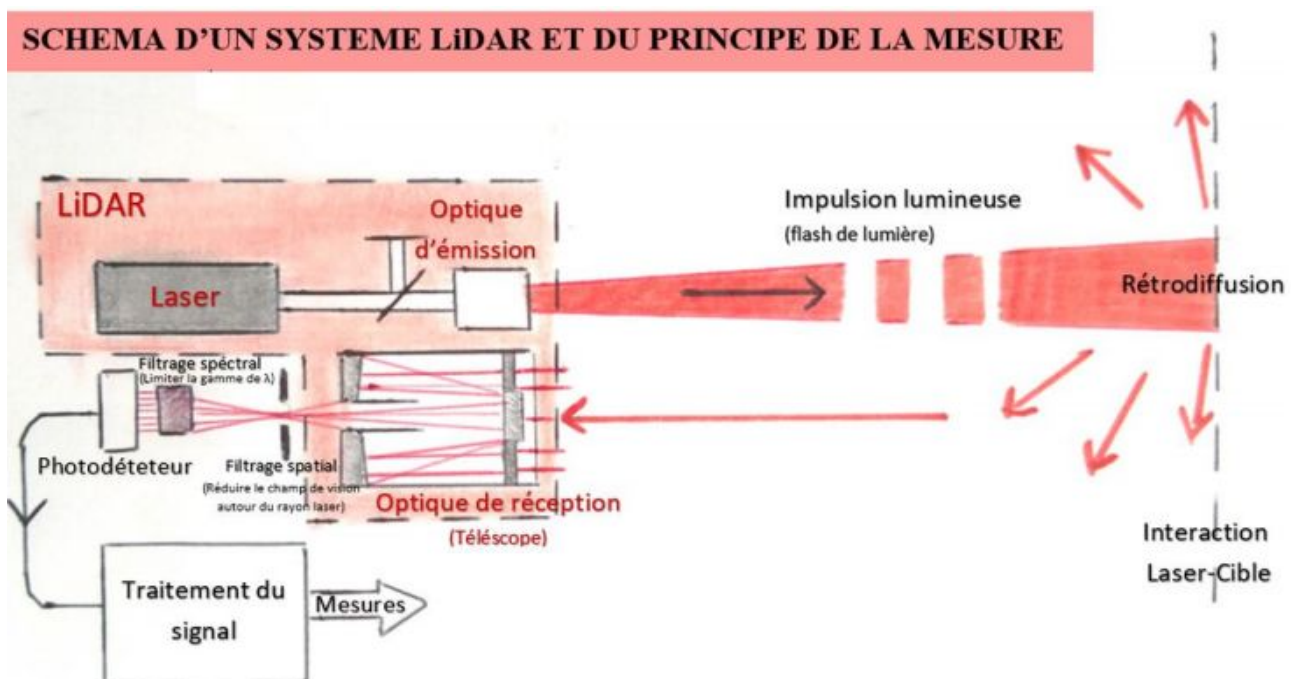


Figure 3: Schéma d'un système LiDAR et du principe de la mesure

3. L'étonnante diversité des Lidars

On remarque une grande diversité des LiDARs.

→ Diversité des applications

- Mesure de distance

La distance est donnée par la mesure du délai entre l'émission d'une impulsion et la détection d'une impulsion réfléchie, en connaissant la vitesse de la lumière.

- Mesure de vitesse

La mesure de vitesse emploie un laser à spectre d'émission fin avec une fréquence bien déterminée. L'effet Doppler-Fizeau, (décalage de la fréquence de l'onde réfléchie et reçue) permet de déterminer la vitesse de l'objet.

- Mesure de température, de concentration, de densité, ...

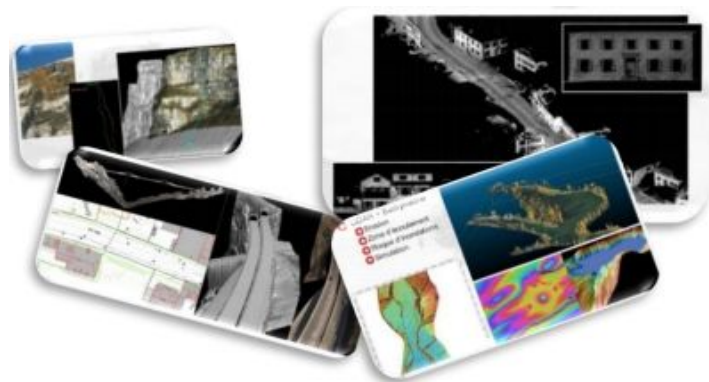
Dans l'atmosphère, beaucoup d'autres paramètres peuvent être mesurés si l'on sait isoler l'effet des différentes interactions entre la lumière et la matière le long du faisceau laser.

→ Diversité des domaines d'applications

Le LiDAR trouve ses applications en:

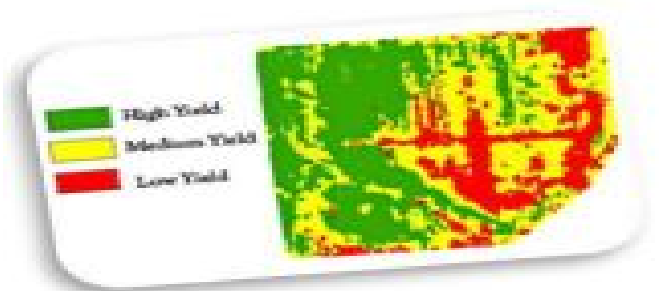
Topographie

- Falaises, pentes raides
- Infrastructures (routes, rails, ...)
- Mobiles terrestres urbains - Hydrologie



Géosciences

- Risque sismique
- Météorologie
- Physique de l'atmosphère



Sciences de l'environnement

- Etude de la pollution atmosphérique

- Agronomie : Le LiDAR peut être utilisé pour aider les agriculteurs à déterminer les zones des parcelles sur lesquelles ils doivent répandre plus d'engrais afin d'obtenir un meilleur rendement agricole pour leurs cultures.
- Le LiDAR peut créer une carte topographique des champs et ainsi révéler les pentes et l'exposition au soleil.

Infrastructure de transport

- La régulation du trafic aérien
- Le guidage automatique de véhicules terrestres ou spatiaux
- La sécurité routière : Les LiDAR sont utilisés par les gendarmes pour contrôler la vitesse des véhicules.



Patrimoine et Archéologie

Le LiDAR est utilisé pour la détection de sites enfouis sous la végétation ou l'eau.

L'optimisation de parcs éoliens

Le LiDAR peut être utilisé pour augmenter la production d'énergie d'un parc éolien en mesurant avec précision les vitesses et les turbulences du vent.

→ Diversité d'émission

L'éloignement de la cible va de quelques dizaines de centimètres à plusieurs centaines de milliers de kilomètres (calcul de la distance Terre-Lune).

→ Diversité de taille

La taille du LiDAR va de l'occupation d'un bâtiment entier au LiDAR portable.

→ Diversité des types de lasers utilisés

A gaz, à colorant, à solide, à semi-conducteur, accordable ou non, pulsé ou continu. C/C ; Il est donc difficile de classer les lidars par principe ou par application ; toutefois, dans la suite, on se propose de les aborder par type de cible (dure ou diffuse) et type d'interaction.

4. Les différents types de Lidar et les modes de fonctionnement

LiDAR à cible dure

Le télémètre laser

Les LiDARs peuvent être catégorisés en fonction de leurs fonctions, ici nous étudierons le télémètre laser qui a pour fonction la mesure de distance. Le télémètre laser fonctionne selon le principe d'un radar, une impulsion est envoyée puis réfléchi par un objet. Le signal retour est ensuite détecté puis traité de manière à avoir une distance entre le télémètre et la cible.

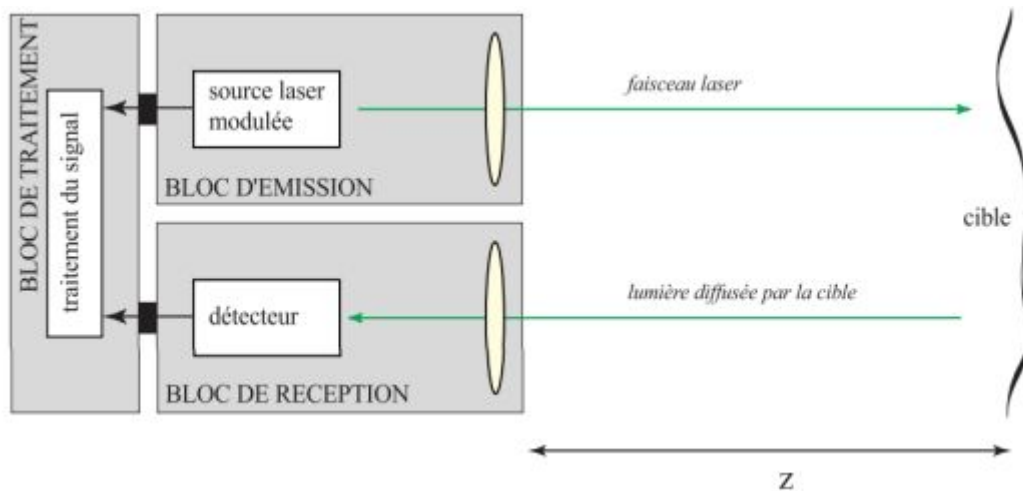


Figure 4: Schéma de principe d'un télémètre à détection directe

Les ondes émises sont généralement dans l'ordre de quelques kHz à plusieurs THz. C'est pourquoi on parle de télémètre laser qui émet dans la partie visible ou infrarouge du spectre. Il existe plusieurs façons de mesurer la distance grâce au même système. Une première approche est de calculer le temps entre l'aller et le retour de l'impulsion. Cette approche est dite télémétrie temps de vol, un chronomètre est déclenché au moment où l'impulsion est envoyée, puis arrêté au moment de sa détection. Une autre manière de calculer peut se faire par l'utilisation de la télémétrie laser à triangulation qui consiste à calculer la distance grâce à la position du rayon retour sur le capteur et l'angle de celui-ci.

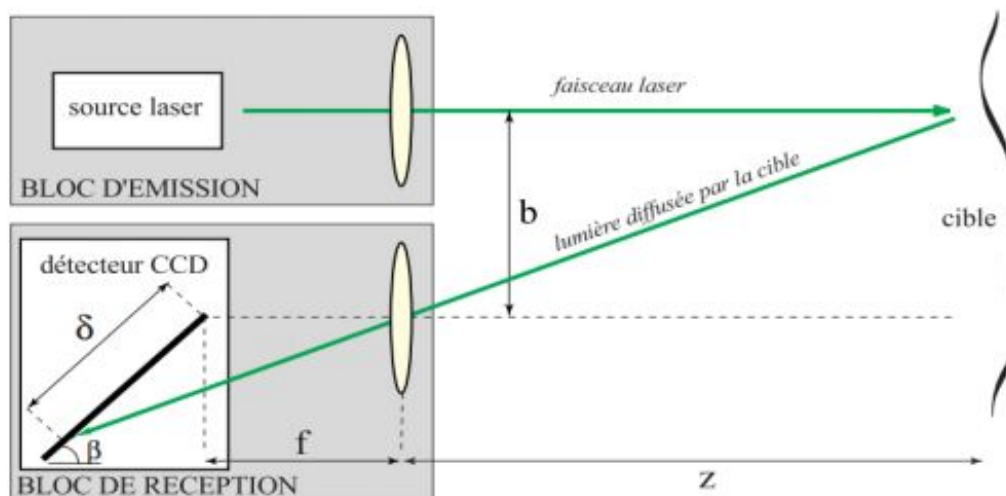


Figure 5: Schéma Principe de la mesure de distance par triangulation

La télémétrie laser est un moyen très pratique de mesurer des distances néanmoins cette méthode présente des incertitudes liées à différents facteurs. Tout d'abord la cible sur laquelle est projeté le rayon n'est pas toujours lisse ou verticale, ainsi en fonction de l'inclinaison ou de l'irrégularité de la cible, la distance ne sera pas calculée très précisément.

En plus de ces incertitudes, le signal envoyé par l'émetteur peut être perturbé par l'air qui doit être traversé, le signal peut perdre en précision et le télémètre donner des mesures encore une fois moins précises. Le télémètre laser permet donc d'obtenir la distance en un point précis de l'espace. Pour pouvoir obtenir une multitude de points, le

télémètre à balayage est intéressant. Celui-ci utilise un miroir rotatif permettant l'envoi d'une sorte de nappe laser et ainsi la mesure de plusieurs points. Cet appareil permet de fournir une carte en trois dimensions, grâce aux points recueillis on peut donc détecter des obstacles ou cartographier un environnement.

LiDAR à détection cohérente pour la mesure de vitesse

Un autre mode de fonctionnement des lidars est le lidar à détection cohérente ou hétérodyne. Ce système s'applique directement à la mesure précise de vitesse d'une cible dure via l'effet Doppler: Si une onde est émise à une certaine fréquence, lorsque la distance entre l'émetteur et le récepteur varie en fonction du temps, la fréquence de l'onde semble varier. Par conséquent cet effet provoque un décalage de fréquence de l'onde réfléchie par un objet proportionnellement à sa vitesse en direction de l'observateur, dite vitesse radiale. Le traitement du signal s'effectue par l'estimation de la fréquence instantanée du signal hétérodyne, afin de remonter à la vitesse de la cible.

$$i_{het}(t) = i_0 \cos(2\pi(\Delta f_0 + 2\frac{V_r}{\lambda})t + \phi)$$

Avec

Δf_0 : le décalage fixe imposé à l'oscillateur local

V_r : la vitesse radiale de la cible

λ : la longueur d'onde émise

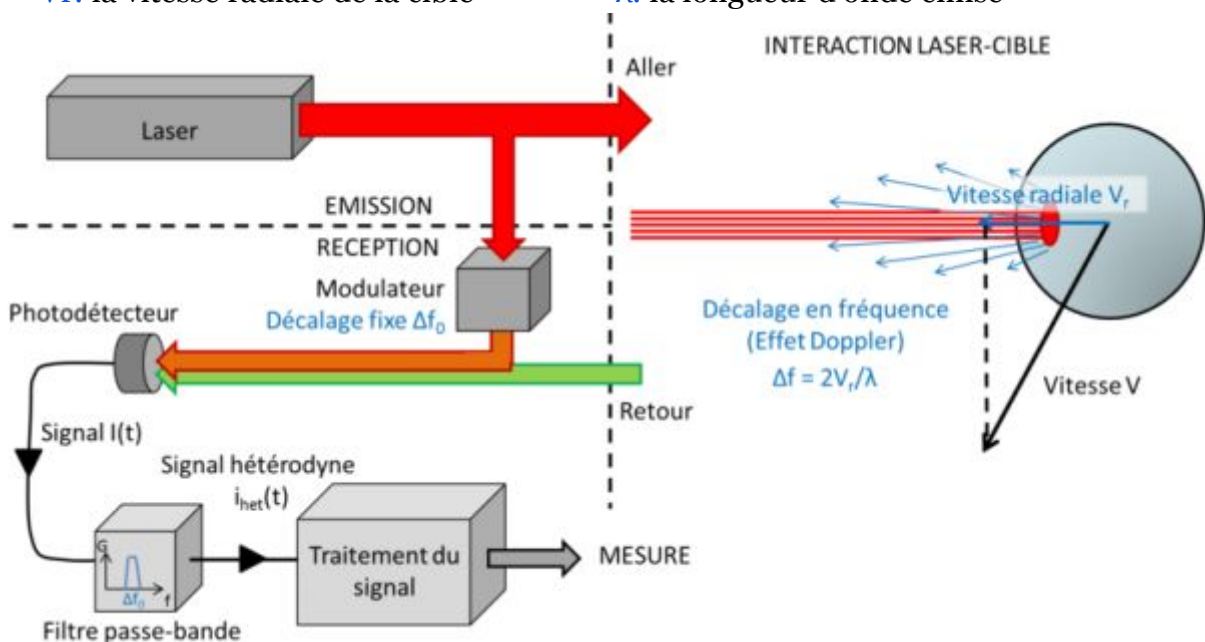


Figure 6: Schéma général du LiDAR à détection cohérente

LiDAR à cible diffuse – lidar atmosphérique

LiDAR à rétrodiffusion (Rayleigh, Mie ou Raman)

Le LiDAR rétrodiffusion est le plus ancien des lidars, crée en 1968 il est le plus simple des LiDARS. Ce LiDAR conduit à des lasers de forte puissance encombrants. La rapide évolution ces dix dernières années des technologies pourrait conduire à des architectures LiDAR d'encombrement acceptable.

Le LIDAR à rétrodiffusion mesure la puissance du rayonnement optique rétrodiffusé par les particules de l'atmosphère. Il existe deux types de LIDAR à rétrodiffusion : le LIDAR à rétrodiffusion élastique (LIDAR Rayleigh et Mie) et le LIDAR à rétrodiffusion inélastique (LIDAR Raman).

Le LIDAR Raman utilise la diffusion Raman pour caractériser les espèces chimiques rencontrées : la mesure de la concentration d'un gaz particulier caractérisé par un décalage spectral donné entre émission et réception.

Les LiDARs Rayleigh s'appuient sur la rétrodiffusion des molécules dont la présence est assurée à toute altitude. Il est utilisé pour mesurer la densité de l'atmosphère stratosphérique et permet un calcul de profil de température.

Une version appelée: LIDAR de Mie, utilise une source laser pour mesurer à distance les propriétés de l'atmosphère. Généralement utilisé en météorologie, il sert également à déterminer la concentration et la répartition des aérosols dans l'air et notamment à proximité de voies de circulation. Le principe de fonctionnement du LIDAR à rétrodiffusion est présenté dans le schéma ci-dessous.

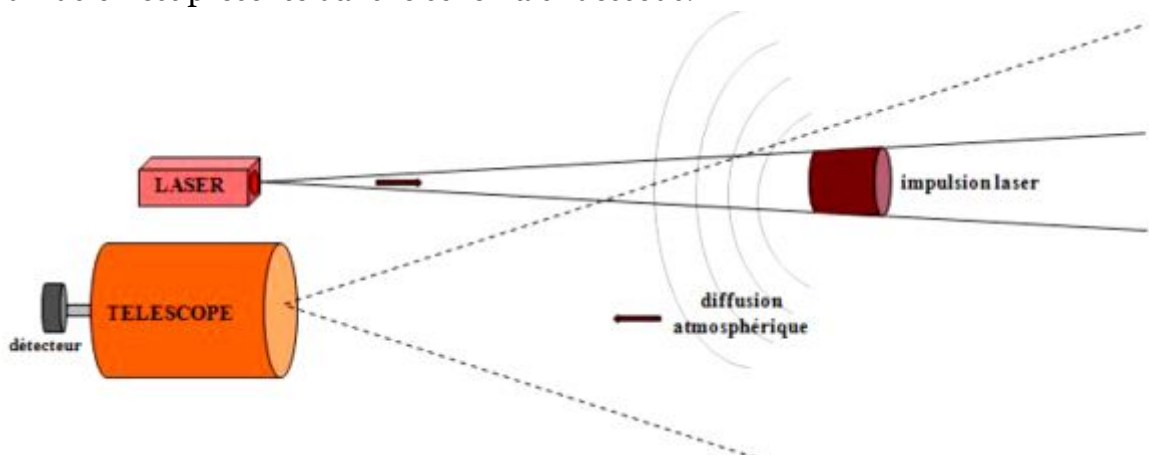


Figure 7: Schéma général du LiDAR à rétrodiffusion

Un système d'émission envoie sous forme d'impulsions un faisceau lumineux, monochromatique, directif et cohérent en direction du milieu à étudier. Quand le faisceau de lumière rencontre des aérosols, la diffusion renvoyée en direction de l'appareil (rétrodiffusion élastique) est récupérée par le module de détection du LIDAR.

Le récepteur est constitué d'un télescope qui récupère la lumière diffusée par l'atmosphère dans sa direction pour être transformée en signal électrique après détection puis convertie en signal numérique. Puisque le télescope est placé à proximité immédiate du laser, on parle de lumière rétrodiffusée.

Le faisceau laser se déplace à la vitesse de la lumière dans le milieu atmosphérique et lorsque qu'il rencontre les molécules de l'air et les aérosols il est en partie rétrodiffusé. La mesure se fait à intervalles de temps décalés par rapport à l'émission laser, ce décalage entre émission et réception permet de calculer la distance des couches diffusantes.

LiDAR à fluorescence

Le lidar par fluorescence est une technique du lidar permettant d'analyser un objet/matériau sans contact avec celui-ci permettant ainsi de ne pas le dégrader. La

technique consiste à envoyer une onde lumineuse avec un laser à pulsation sur l'objet avec une longueur d'onde définie et d'analyser la lumière renvoyée par cet objet.

La fluorescence est une émission lumineuse créée après l'excitation des électrons d'une molécule fluorescente qui après absorption d'énergie rentre dans un état instable noté S1 elle va donc chercher naturellement à retourner dans un état stable. C'est lors de son retour à son état le plus stable E0 qu'on a une émission lumineuse liée au dégagement d'énergie. Le retour dans l'état stable peut se faire de deux manières différentes: par **fluorescence** ou par **phosphorescence**. La fluorescence est caractérisée par l'émission d'un photon de manière très rapide. Cette rapidité s'explique par le fait que l'émission respecte une des règles de sélections de l'émission de photons de la mécanique quantique qui est $\Delta S=0$, ce qui signifie que la molécule reste dans un état singulet.

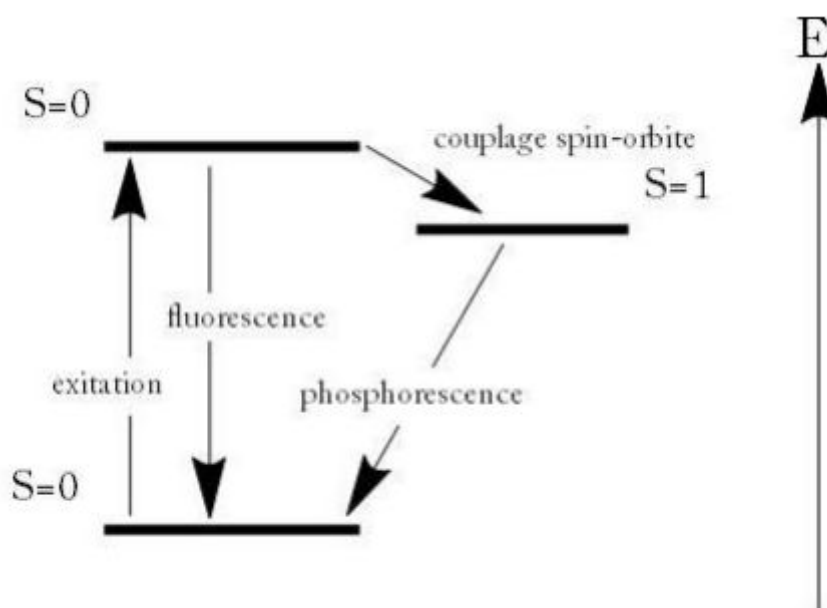


Figure 8: Schéma fluorescence caractérisée par l'émission d'un photon

La technique a été développée sur le principe du LiDAR classique et a commencé à être utilisée pour la détection et classification de matière organique dissoute et l'étude des phytoplanctons. C'est dans les années 80 que le Lidar à fluorescence a vu son champ d'application étendu. Ainsi le LiDAR par fluorescence a commencé à être utilisé dans les relevés liés à l'environnement tels que l'étude des plantes par exemple.

L'application principale du Lidar par fluorescence, à savoir: le diagnostic de surfaces de monuments historiques, n'a vu le jour que dans les années 90 avec le premier essai sur le Baptistère et la Cathédrale de Parme en 1994. Le test mené a permis de mettre en valeur la précision de ce type de Lidar en comparant les pierres de ses monuments avec les différentes pierres disponibles dans la région de Parme. De plus le Lidar par fluorescence s'est révélé être capable de déterminer les traitements protecteurs appliqués sur les matériaux ainsi que depuis peu la faisabilité d'une caractérisation à base de fluorescence de litho types différents et d'autres matériaux de construction comme des mortiers ou ciments.

LiDAR à absorption différentielle : « DIAL »

Le LIDAR à absorption différentielle peut être utilisé pour la mesure des principaux gaz dans l'atmosphère.

Le LIDAR DIAL utilise deux longueurs d'ondes, émises par la source laser : Une longueur d'onde correspond à une raie spectrale d'absorption d'une espèce moléculaire de l'atmosphère (λ_a) tandis que l'autre longueur d'onde est hors de la raie d'absorption et sert de référence (λ_o).

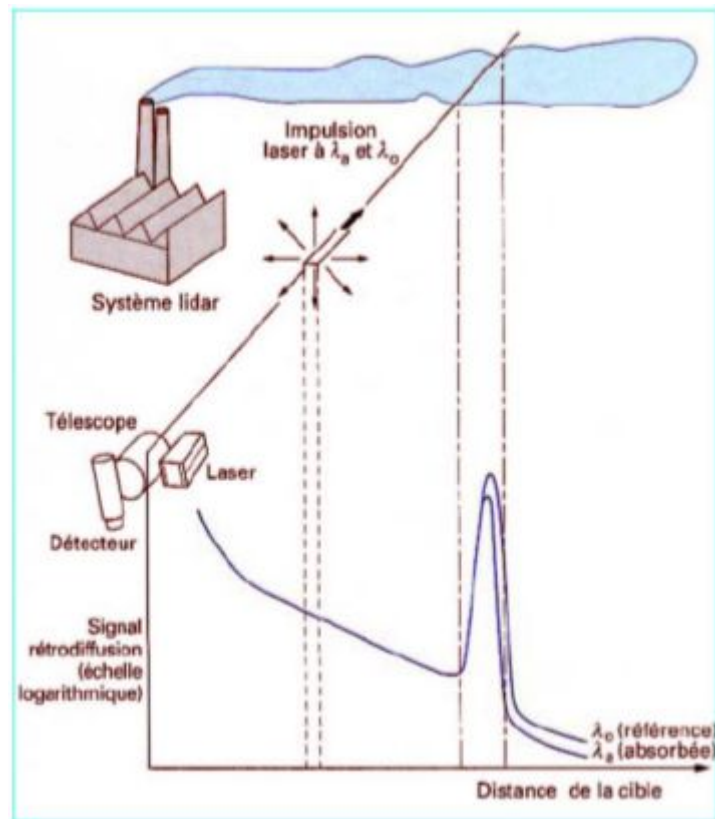


Figure 9: Principe du LiDAR DIAL

Certains lidars DIAL utilisent la technique de la détection hétérodyne pour s'affranchir totalement du parasite de la lumière ambiante et effectuer des mesures de jour comme de nuit.

Plusieurs difficultés subsistent. Il faut disposer d'un laser bi-fréquence accordable en longueur d'onde pour viser le sommet de la raie d'absorption de l'espèce à mesurer. Le phénomène de décalage par la pression de raies spectrales d'absorption étroites fausse les mesures Lidar faites sur de grandes gammes d'altitude.

Les applications importantes du Lidar DIAL sont par exemple la surveillance de la couche d'ozone stratosphérique, la mesure des concentrations de gaz à effet de serre tel que le dioxyde de carbone ou le méthane ou la mesure de gaz polluants divers dans la basse atmosphère.

5. Intérêts, avantages et inconvénients du lidar

Les LiDARs deviennent des instruments importants répondant aussi bien à des objectifs scientifiques de connaissance de l'atmosphère qu'à des besoins de mesure et de contrôle de l'environnement.

En effet, de nombreux avantages le placent comme un capteur indispensable, d'où la multiplication de réalisations et de projets de LiDARs. Un avantage du LiDAR est une

grande précision dans la mesure de distance (une précision de l'ordre du centimètre), grâce aux très courtes durées d'impulsions.

Un deuxième avantage est l'indépendance à la luminosité. On peut effectuer certaines mesures en plein jour ou la nuit. Le lidar utilise un capteur d'éclairage actif dont les données peuvent être récupérées jour et nuit, ce qui n'est pas toujours le cas d'autres techniques. Un autre atout du LiDAR est la rapidité d'acquisition et de traitement des données avec une extrême précision. Il est donc possible d'obtenir une cartographie 3D en temps réel grâce à un seul instrument LiDAR.

Le premier inconvénient du LiDAR est son prix élevé. Les LiDARs sont sensibles aux conditions atmosphériques. Ainsi par temps de pluie ou de brouillard, le faisceau peut être perturbé, à cause du principe de réfraction: lorsque le rayon traverse un milieu différent (comme une goutte d'eau), la vitesse de l'onde change, et ainsi la superposition de différentes couches de différents milieux rend approximatif l'évaluation de la distance.

B. RADAR

1. Point Historique

En 1864, James Clerk Maxwell décrit les lois de l'électromagnétisme, ce qui permet pour la première fois de travailler sur leur source. En 1888, Heinrich Rudolf Hertz montre que les ondes électromagnétiques sont réfléchies par les surfaces métalliques. Au début du ^{xx}e siècle, le développement de la radio et de la TSF (par Marconi, entre autres) permet de développer les antennes nécessaires à l'utilisation du radar.

2. Principe de fonctionnement

Le radar exploite la propriété des ondes radio à se réfléchir sur les objets; Une onde radio à haute fréquence produite par un émetteur (oscillateur et amplificateur) et acheminée à une antenne via un guide d'onde est émise dans l'espace. Lorsqu'elle rencontre un objet, elle est réfléchi. La mesure du signal réfléchi (ou de l'écho) et son analyse comparée au signal émis permettent de fournir des informations (distance, vitesse, position) sur l'objet cible.

La qualité de la détection dépend non seulement du matériau de la cible, de sa forme, mais aussi de la fréquence du signal émis. Les antennes des radars peuvent être fixes ou tournantes. Les radars eux-mêmes pouvant être fixes ou installés sur des véhicules (terrestres, aériens ou navires).

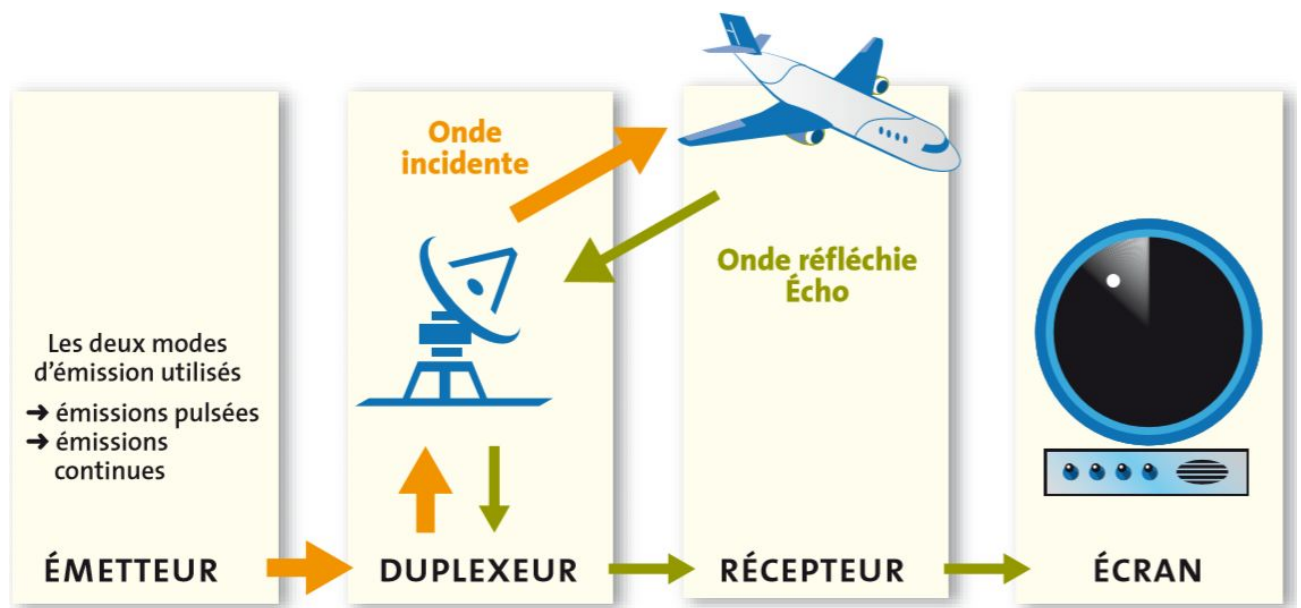


Figure 10: Principe de fonctionnement d'un système RADAR

Les deux modes d'émissions utilisés sont:

Les émissions pulsées: des impulsions très courtes sont émises et le temps plus ou moins long entre chaque impulsion est exploité pour lire les éventuels échos reçus. Les radars à impulsions permettent de mesurer la distance, l'altitude, la direction et la taille de la cible ;

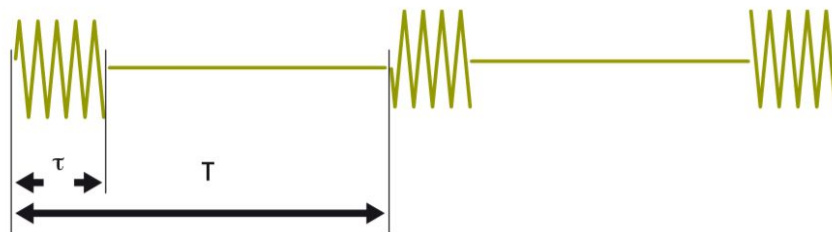


Figure 11: Émission pulsée

Impulsions de durée τ et de période de répétition T

Les émissions continues: le signal est émis de façon continue. Les radars utilisant ce type d'émission sont destinés à mesurer la vitesse en exploitant l'effet Doppler (application courante pour le contrôle de vitesse des véhicules).

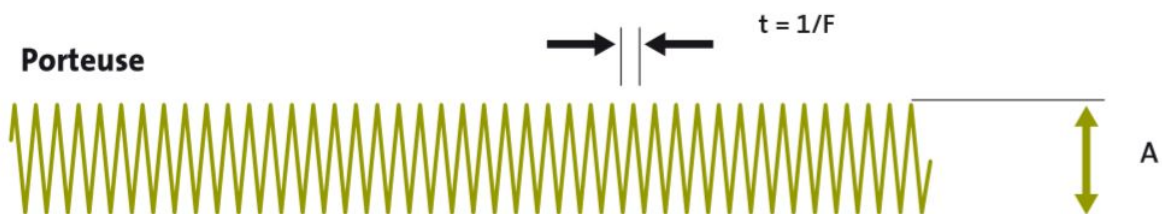


Figure 12: Émissions continues

On considère différents types de radars:

Les radars primaires émettent une onde et détectent le signal réfléchi pour fournir des informations (position, direction...);

Les radars secondaires (utilisés en navigation aérienne) se limitent à interroger un aéronef en émettant un signal codé et à mesurer les signaux émis par le transpondeur de l'avion. Cela permet d'obtenir la position et l'identification de l'avion sous réserve que celui-ci soit coopératif (qu'il utilise son transpondeur pour émettre ces informations). Les radars secondaires n'exploitant pas le signal réfléchi par la cible, ils n'ont pas besoin d'être très puissants (environ 1 000 fois moins qu'un radar primaire).

C.SONAR

1. Point Historique

Bien que la capacité de l'eau à propager des ondes sonores ait été observée depuis fort longtemps, l'exploitation pratique de cette potentialité, se heurtant à des contraintes technologiques difficiles, en est relativement récente. L'idée de détecter activement, en envoyant des signaux acoustiques, des obstacles à la navigation ou des cibles n'a été concrétisée qu'au début du XXe siècle, pendant la Première Guerre mondiale. Le physicien français Paul Langevin démontra (de 1915 à 1918) la faisabilité de la transmission de signaux sonores, puis celle de la détection acoustique active et de la localisation de sous-marins, en utilisant des transducteurs piézo-électriques (cf. Technologie des systèmes sonar) dont le principe allait être quasi universellement repris dans les développements ultérieurs. À la même époque, les marines alliées commençaient à utiliser des systèmes passifs d'écoute de bruit pour la détection des sous-marins ennemis.

La technologie sonar se perfectionna de manière décisive pendant l'entre-deux-guerres, bénéficiant de l'apparition de l'électronique de première génération et des progrès de l'industrie naissante de la radio. Au début de la Seconde Guerre mondiale, la technologie du sonar actif était suffisamment au point pour pouvoir être utilisée opérationnellement à grande échelle sur les navires alliés (les systèmes A.S.D.I.C. – Allied Submarine Detection Investigation Committee – de la Royal Navy) et jouer un rôle déterminant dans la bataille de l'Atlantique contre les sous-marins allemands. À partir de leur entrée en guerre, les États-Unis allaient consentir un effort important de recherche scientifique et technologique dans ce domaine, et faire considérablement progresser, d'une part, les performances des systèmes sonar et, d'autre part, les approches théoriques de la propagation des ondes sonores dans le milieu marin et les théories liées à la détection et la mesure des signaux dans le bruit. La base des connaissances et théories utilisées aujourd'hui.

2. Structure des systèmes sonar

L'ensemble des systèmes de détection/ localisation d'une cible, ou d'un obstacle immergé, à l'aide d'un signal acoustique sont couramment désignés par le terme générique de sonar (de l'acronyme anglais un peu douteux SOund NAVigation and Ranging). Le sonar ainsi présenté est donc l'équivalent, en acoustique sous-marine, du radar (RAdio Detection And Ranging) en électromagnétisme. La détection et la localisation d'une cible par un système sonar sont basées sur la réception:

- Soit de l'écho d'un signal émis par le système et réfléchi sur la cible (sonar actif);

- Soit directement du bruit rayonné par la cible (sonar passif).

Dans le premier cas, le sonar doit disposer d'un système d'émission de signaux acoustiques de forte puissance. Les signaux émis, et éventuellement renvoyés par la cible, se propagent dans l'océan, où ils sont affaiblis, déformés et bruités. La réception s'effectue sur une antenne, souvent constituée de plusieurs capteurs (hydrophones) dont les sorties sont conjuguées de manière adéquate. Les signaux passent ensuite dans une chaîne de traitement adaptée, destinée à améliorer le rapport signal à bruit, et font l'objet d'une décision de détection. Ils sont finalement utilisés pour la mesure (par exemple à des fins de localisation) et éventuellement l'identification ou la caractérisation de la cible. La structure générale de fonctionnement d'un système sonar est schématisée en figure.

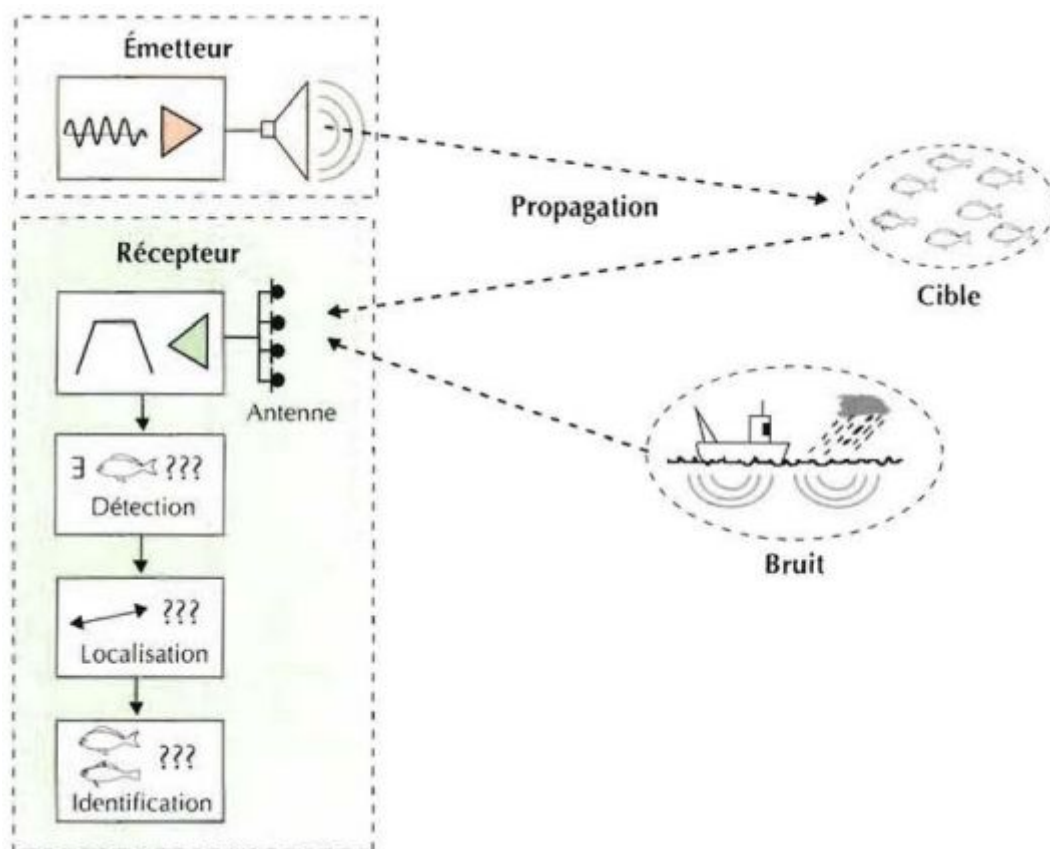


Figure 13: Structure générique d'un système sonar

De quoi est réellement constitué cet équipement que l'on appelle un "SONAR" ?

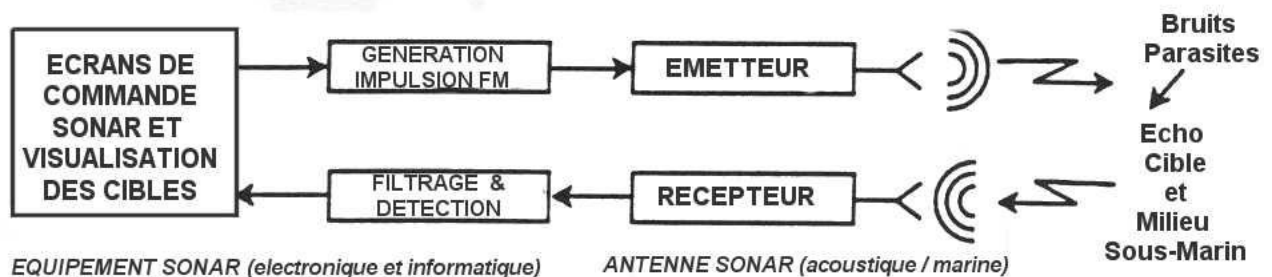


figure 14: Les éléments constitutifs d'un SONAR

3. Les principales utilisations

Il est utilisé:

- par les marines de guerre pour la détection de sous-marins (et de bâtiments de surface par les sous-marins), ainsi que des mines et objets posés sur le fond par les mouilleurs de mines, mais également pour le guidage des torpilles lors de leur trajectoire sous-marine ;
- pour la pêche, pour la détection des bancs de poissons ;
- pour la navigation maritime et fluviale et en hydrographie, pour cartographier le fond des océans et d'autres plans d'eau — on parle alors de sondeur bathymétrique ;
- en archéologie subaquatique et sous-marine ;
- dans des capteurs de pollutions aquatiques.

Chapitre 2

Introduction au systèmes embarqués

I. Introduction

Un système embarqué est défini comme un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche bien précise. Le terme désigne aussi bien le matériel informatique que le logiciel utilisé. Ses ressources sont généralement limitées. Cette limitation est généralement d'ordre spatial (encombrement réduit) et énergétique (consommation restreinte).

Le premier système moderne embarqué reconnaissable a été le Apollo Guidance Computer, le système de guidage de la mission lunaire Apollo, développé par Charles Stark Draper du Massachusetts Institute of Technology. Chaque mission lunaire était équipée de deux systèmes (AGC), un chargé du système de guidage inertiel et un pour le Module lunaire.

Un système embarqué est un système servant à résoudre des fonctions et des tâches spécifiques et limitées.

- Associé à contraintes en temps réel
- Souvent conçu en matériel avec des parties en logiciel.
- La complexité du système et des fonctions varient largement
- Contrôleur d'un lave-vaisselle
- Portable MP3
- Contrôleur de missiles
- Le logiciel est utilisé pour la flexibilité
- Le matériel est utilisé pour la performance et la consommation

II. Caractéristiques des systèmes embarqués

Les systèmes embarqués fonctionnent généralement en Temps Réel (TR) : les opérations de calcul sont alors faites en réponse à un événement extérieur (interruption matérielle). La validité et la pertinence d'un résultat dépendent du moment où il est délivré. Une échéance manquée induit une erreur de fonctionnement qui peut entraîner soit une panne du système (plantage), soit une dégradation non dramatique de ses performances.

Lorsque les systèmes embarqués sont utilisés dans les produits de grande consommation, ils sont fabriqués en grande série. Les exigences de coût se traduisent alors en contraintes sur les différentes composantes du système: utilisation de faibles capacités mémoires et de petits processeurs (4 bits ou 8 bits), mais en grand nombre. Ainsi, les systèmes embarqués sont particulièrement sensibles au coût de production. Il existe des

applications dans lesquelles les contraintes de coût de production et de maintenance ont une importance de même niveau que les performances envisagées.

Dans les systèmes embarqués autonomes, la consommation d'énergie est un point critique pour le coût. En effet, une consommation excessive augmente le prix de revient du système embarqué, car il faut alors des batteries de forte capacité.

III. Complexité des systèmes embarqués

Les systèmes embarqués requièrent souvent un faible encombrement (faible poids) PDA(Personal Digital Assistant) , Internet et téléphone mobiles, ...). Leur technologie fait alors appel à une électronique et à des applications portables où l'on doit minimiser aussi bien l'encombrement que la consommation électrique. Par conséquent, la réalisation du packaging afin de faire cohabiter sur une faible surface de l'électronique analogique, de l'électronique numérique, des composants RF (Radiofréquence) sans interférences est une tâche difficile. En effet, les performances des systèmes sur carte deviennent obsolètes dans le contexte des besoins actuels. Dans les stratégies de conception actuelles, un système embarqué est généralement intégré sur un support silicium unique constituant ainsi un système complet intégré sur une puce SoC(System on a Chip).

Les systèmes sur puce contiennent généralement une grande variété de dispositifs programmables tels que des microcontrôleurs, des processeurs de traitement de signaux DSP (Digital-Signal Processor) et des ASIC qui sont développés pour des applications complexes nécessitant une production en grande série.

Les mémoires (ROM et RAM) y sont intégrés pour le stockage des données et des programmes. Ces composants digitaux cohabitent généralement sur le même support de silicium avec des composants analogiques et mixtes divers tels que des composants radiofréquence (RF) comme moyen de communication, des composants optiques pour le transfert de données à haut débit, des MEMS (Micro Electro Mechanical System) pour l'interface avec le monde externe, des convertisseurs analogiques/numérique et numérique/analogique requis pour le dialogue interne. L'objectif est d'obtenir une coopération harmonieuse entre composants embarqués afin de garantir des services globaux. Des contraintes d'implémentation physique sont liées à la consommation de ressources et au contexte de déploiement tel que le poids, la taille physique, la résistance aux vibrations, ou aux irradiations, ..., etc.

IV. Définition des systèmes embarqués

Quelle que soit la nature et la complexité du système, on décompose un système embarqués en:

- le système contrôlé
- le système de contrôle

Le système contrôlé = environnement (procédé) équipé d'une instrumentation qui réalise l'interface avec le système de contrôle.

Le système de contrôle = éléments matériels (microprocesseurs...) et logiciels dont la mission est d'agir sur le procédé via les actionneurs en fonction de l'état de ce procédé indiqué par les capteurs de manière maintenir ou conduire le procédé dans un état donné.

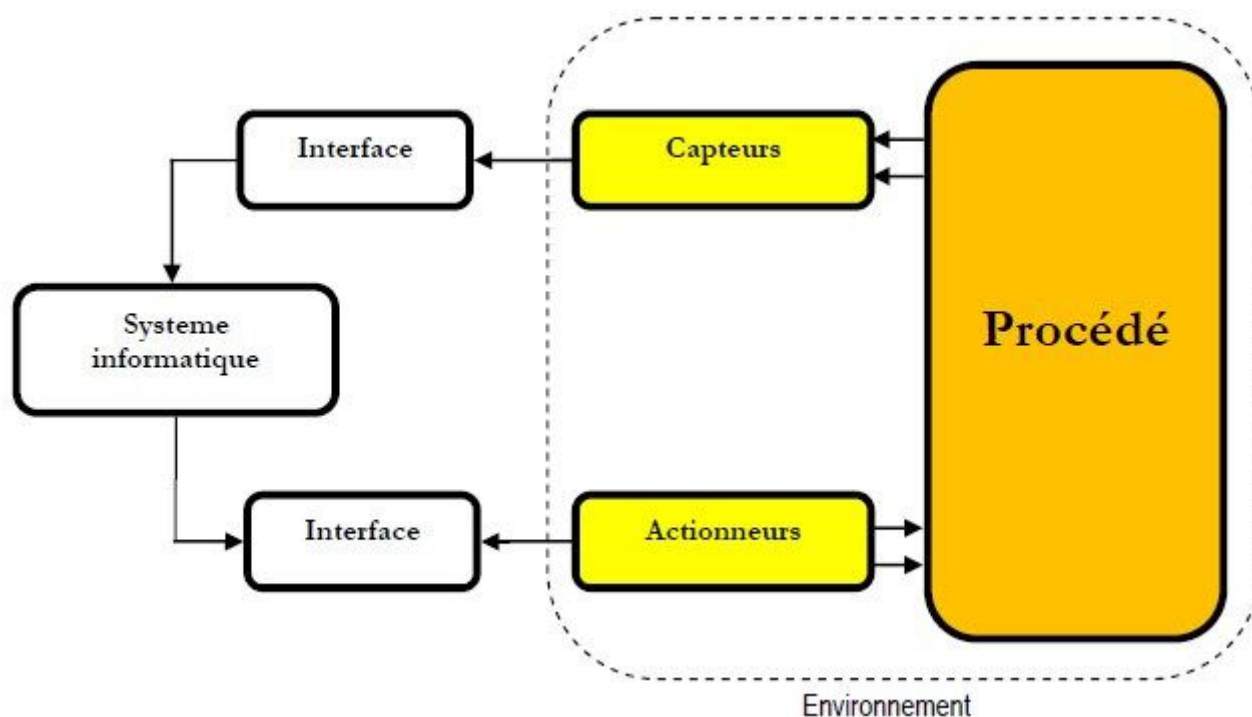


Figure 15: Système Électronique Embarqué

Un système électronique embarqué ou enfoui est un élément constitutif d'un système plus complexe pour lequel il rend des services bien précis (contrôle, surveillance, communication...). Il est constitué de parties matérielles et logicielles qui sont conçues spécifiquement pour réaliser une fonction dédiée.

Système embarqué = Système électronique/informatique conçu pour réaliser une ou plusieurs tâches précise.

Les caractéristiques principales d'un système électronique embarqué sont:

- Autonomes: Une fois enfouis dans l'application ils ne sont (le plus souvent) plus accessibles
- Temps réel: Les temps de réponses de ces systèmes sont aussi importants que l'exactitude des résultats.
- Réactifs: Il doit réagir à l'arrivée d'informations extérieures non prévues.

Informatique	Embarqué
<ul style="list-style-type: none"> • Processeur standard • Multiples unités fonctionnelles (flottant) • Vitesse élevée (> GHz) • Consommation électrique élevée • Chaleur • Taille 	<ul style="list-style-type: none"> • Processeur dédié (contrôleur) • Architecture adaptée • Vitesse faible (~200 MHz) • 8-32 bits : mémoire limitée • Basse consommation • Petite taille, grand volume => faible coût • Processeur DSP (traitements)

<ul style="list-style-type: none"> • MMU (mémoire virtuelle) • OS • Cache • Grand nombre de périphériques 	<ul style="list-style-type: none"> • Très puissants • Quelques Mo de mémoire • RTOS
---	--

Tableau 1: Comparaison aux systèmes informatiques standards

V. Classification des systèmes embarqués

- Temps réel dur ('hard real-time') : le non respect des contraintes temporelles entraîne la faute du système
 - Exemple: contrôle de trafic aérien, système de conduite de missile, ...
- Temps réel souple ('soft real-time') : le respect des échéances est important mais le non respect des échéances n'a pas de graves conséquences
 - Exemple : système d'acquisition de données pour affichage
- Temps réel ferme ('firm real-time') : temps réel souple, mais si l'échéance est dépassée le résultat obtenu n'a plus de valeur (et est donc écarté)
 - Exemple : Projection vidéo

A. Système Transformationnel

Activité de calcul, qui lit ses données et ses entrées lors de son démarrage, qui fournit ses sorties, puis meurt.

B. Système Interactif

Système en interaction quasi permanente avec son environnement, y compris après l'initialisation du système; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passés); le rythme de l'interaction est déterminé par le système et non par l'environnement.

C. Système Réactif ou Temps Réel

Système en interaction permanente avec son environnement, y compris après l'initialisation du système; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passées); mais le rythme de l'interaction est déterminé par l'environnement et non par le système.

VI. Les Applications

A. Arduino

1. Qu'est ce que c'est arduino?

Arduino est un projet créé par une équipe de développeurs, composée de six individus : Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti. Cette équipe a créé le "système Arduino". C'est un outil qui va

permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes.

Il s'agit d'une carte électronique basée autour d'un microcontrôleur Atmega du fabricant Atmel, dont le prix est relativement bas pour l'étendue possible des applications.

Arduino est une plateforme électronique open-source basé sur le matériel et le logiciel facile à utiliser. Les cartes Arduino sont capables de lire les entrées - La lumière sur un capteur, un doigt sur un bouton ou un message Twitter - et la transformer en une sortie - activation d'un moteur, d'allumer une LED, afficher une écriture. On peut commander la carte Arduino en envoyant un ensemble d'instructions au microcontrôleur disposé sur la carte. Pour ce fait, on utilise le langage de programmation Arduino (Basé sur le câblage) et le logiciel Arduino (IDE) , basé sur le traitement.

2. Types des cartes Arduino

On peut classer les cartes Arduino en deux grandes familles.

- Cartes Arduino Officielles (Classique)
- Cartes Arduino Compatibles (Dérivées)

Cartes Arduino Officielles (Classique)

Ces cartes sont fabriquées en Italie par le fabricant officiel : Smart Projects, son site officiel est Arduino.cc ou Arduino.org , Pour tout ce qui est des cartes Arduino dites « Officielles » elles sont basées généralement sur le même micro-contrôleur AVR à savoir un ATmega328p du fabricant ATMEL .



Figure 16: Carte Arduino officielle

Cartes Arduino Compatibles (Dérivées)

Ces cartes ne sont pas fabriquées par Smart Projects, mais qui sont totalement compatibles avec les shields Arduino classiques (Mais pas avec l'IDE Arduino de base), elles sont fabriquées par diverse entreprises et commercialisées sous un nom différent (Exemples Cartes Arduino Compatible: Chinoises, Olimex, Selectronic, Freeduino, Seeeduino, Femtoduino).



Figure 17: Carte Arduino Compatible

3. Avantage de la carte Arduino UNO: Pourquoi Arduino?

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant aux personnes intéressées plusieurs avantages cités comme suit:

1. **Le prix (réduit):** les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main.
2. **Multi plateforme:** le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
3. **Un environnement de programmation clair et simple:** l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
4. **Logiciel Open Source et extensible:** le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).
5. **Matériel Open source et extensible:** les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence creative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût

4. Caractéristiques techniques de la carte arduino Uno:

La carte Arduino Uno est basée sur un Microcontrôleur ATmega328 cadencé à 16 MHz. C'est la plus récente et la plus économique carte à microcontrôleur. Les caractéristiques techniques de la carte Arduino UNO sont présentées dans le tableau:

Microcontrôleur	ATmega328
Tension de fonctionnement	5V
Tension d'alimentation – entrée - (Recommandée)	7-12V
Tension d'alimentation – entrée - (Limites)	6-20V
Broches E/S numériques	14 (Dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (Utilisables en broches E/S numériques)
Intensité maximum disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maximum disponible pour la sortie 3.3V	50 mA
Intensité maximum disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	32 KB (ATmega328) dont 0.5 KB sont utilisés par le bootloader
Mémoire SRAM (Mémoire volatile)	2 KB (ATmega328)

Tableau 2: Caractéristiques de la Carte Arduino UNO

5. Microcontrôleur ATmega328

Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleur ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8 bits. Un microcontrôleur ATmega328 est un **circuit intégré** qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit.

Aujourd'hui, en soudant un grand nombre de composants encombrants; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en

plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C.

Les deux types de microcontrôleur ATmega328, qu'on trouve sur les cartes Arduino sont:

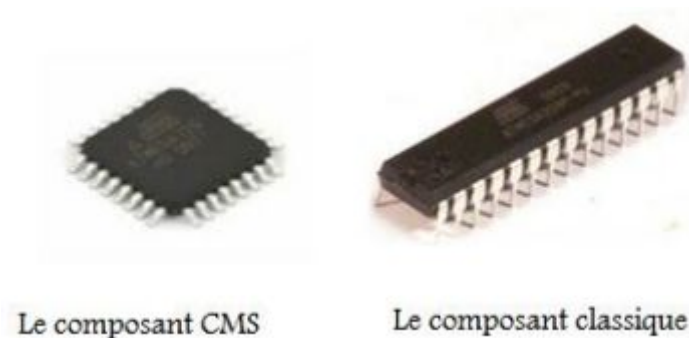


Figure 18: Microcontrôleur ATmega328

C'est quoi la différence des cartes officielles "édition SMD/CMS" ?

Il n'y a pas de différence! enfin presque... "SMD" signifie "Surface Mount Device", en français on appelle ça des "CMS" pour Composants Montés en Surface". Ces composants sont soudés directement sur le cuivre de la carte, il ne la traverse pas comme les autres. Pour les cartes Arduino, on retrouve le composant principal en édition SMD dans ces cartes. La carte est donc la même, aucune différence pour le tuto. Les composants sont les mêmes, seule l'allure "physique" est différente. Par exemple, ci-dessus la "Mega" est en SMD et la Uno est "classique".

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement de:

- **Mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire effaçable et réinscriptible est une mémoire programmée de 32Ko (dont bootloader de 0.5 ko).
- **RAM:** C'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **EEPROM:** C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.

Nous entrons dans le vif du sujet avec le synoptique qui présente le fonctionnement général du microcontrôleur ATMEGA:

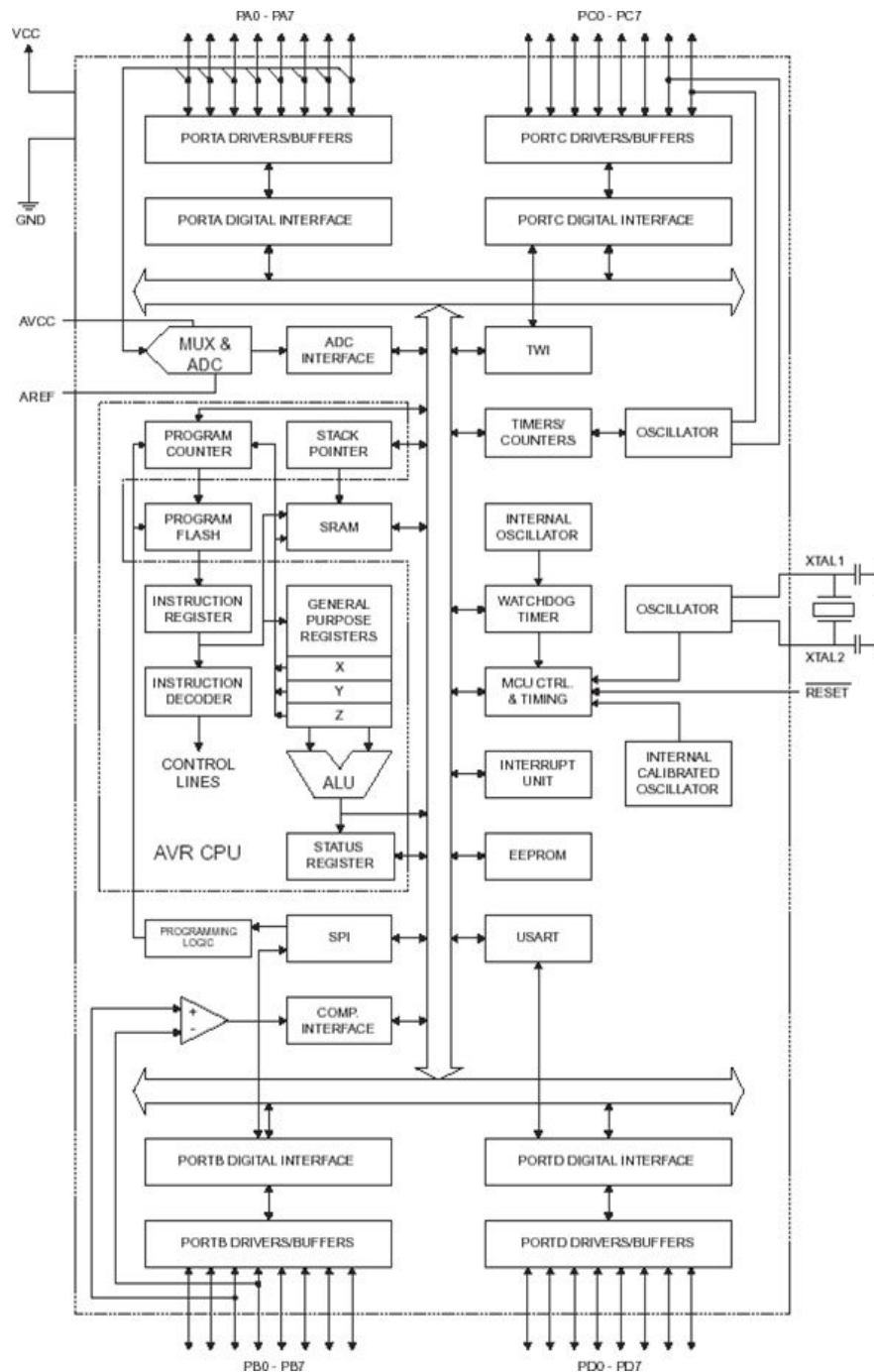


Figure 19: Synoptique générale d'un ATMEGA.

Présentation Physique

L'ATMEGA se présente sous la forme d'un circuit intégré à 40 broches pour le modèle ATMEGA32 en boîtier DPID ou le boîtier TQFP/MLF à 44 broches.

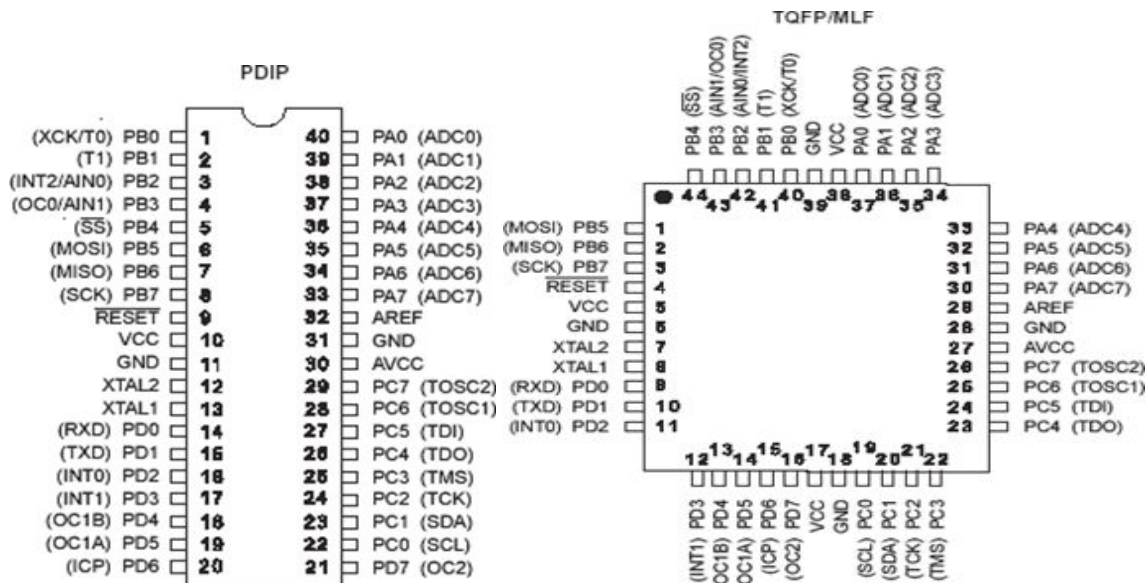


Figure 20: brochage typique d'un ATMEGA32 en boîtier PDIP et PLCC

Descriptions des broches

- **Port A** (PA7.. PA0) le Port A est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi pour les entrées analogiques du convertisseur A/D. Le Port A (comme le B, C et D) est en position trois états quand une condition de reset devient active, même si l'horloge ne court pas.
- **Port B** (PB7.. PB0) le Port B est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi de comparateur analogique (sortie sur PB2, PB3), ou de SPI.
- **Port C** (PC7.. PC0) le Port C est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi comme oscillateur pour le Timer/Compteur 2 et d'interface I2C.
- **Port D** (PD7.. PD0) le Port D est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi de USART et d'entrées pour les interruptions externes.
- **RESET** déclenché par un front descendant maintenue plus de 50 ns il produira le Reset du microcontrôleur, même si l'horloge ne court pas.
- **XTAL1** Entrée de l'oscillateur externe ou libre pour l'horloge interne.
- **XTAL2** Production de l'amplificateur d'oscillateur.
- **AVCC** est une broche de tension d'alimentation pour le Convertisseur A/D qui doit être connectée à VCC via un filtre passe-bas pour éviter les parasites.
- **AREF** est l'entrée de référence analogique pour le Convertisseur A/D avec une tension dans la gamme de 2 V à AVCC avec filtre passe bas.
- **AGND** masse Analogique. Si la masse analogique est séparée de la masse générale, brancher cette broche sur la masse analogique, sinon, connecter cette broche à la masse générale GND.
- **VCC** broches d'alimentation du microcontrôleur (+3 à +5V).
- **GND** masse de l'alimentation.

Les sources de l'alimentation de la carte

La carte Arduino Uno peut-être alimentée soit via la **connexion USB** (qui fournit **5V** jusqu'à 500mA) ou à l'aide d'une **alimentation externe**.

La source d'alimentation est **sélectionnée automatiquement** par la carte .
L'alimentation externe peut être soit un **adaptateur secteur** (Pouvant fournir typiquement de **3V** à **12V** sous **500mA**) ou **des piles**.

L'adaptateur secteur peut être connecté en branchant une prise (2.1mm) positif au centre dans le connecteur jack de la carte.

Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de (**6 à 20 volts**). Cependant, si la carte est alimentée avec **moins de (7V)**, la broche (5V) pourrait fournir moins de (5V) et la carte pourrait être **instable**. Si on utilise plus de (12V), le régulateur de tension de la carte pourrait **chauffer et endommager** la carte. Aussi, la plage idéale recommandée pour alimenter la carte Arduino Uno est entre (**7V et 12V**).

Les broches d'alimentation

VIN: La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source (5V) régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.

5V: la tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (Pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "Tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le (5V) régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (Qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

3.3V: Une alimentation de 3.3V fournie par le circuit intégré FTDI (Circuit Intégré Faisant l'Adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible: Ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V).

GND: Broche de masse (Ou 0V)

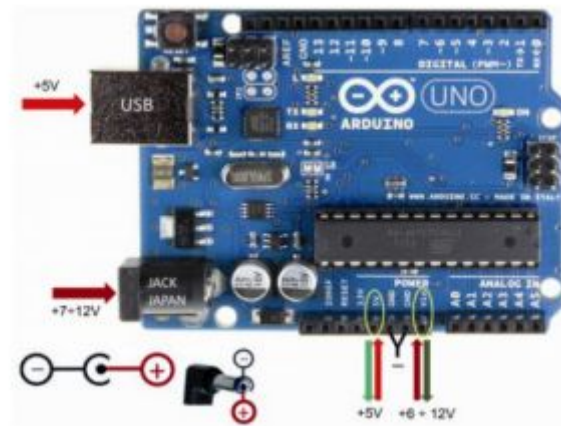


Figure 21: Sources de l'Alimentation de la carte Arduino UNO.

Les entrées et sorties

L'Arduino UNO possède **14 broches** d'entrée/sortie **digitale** (Numérotées des 0 à 13), ces broches peuvent être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en (5V).

Chaque broche peut fournir ou recevoir un maximum de (**40 mA**) d'intensité et dispose d'une résistance interne de "Rappel au plus" (Pull-up) (Déconnectée par défaut) de (**20-50 KOhms**). Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite` (broche, HIGH)

De plus, certaines broches ont des fonctions spécialisées (Voir annexe pour plus de détails):

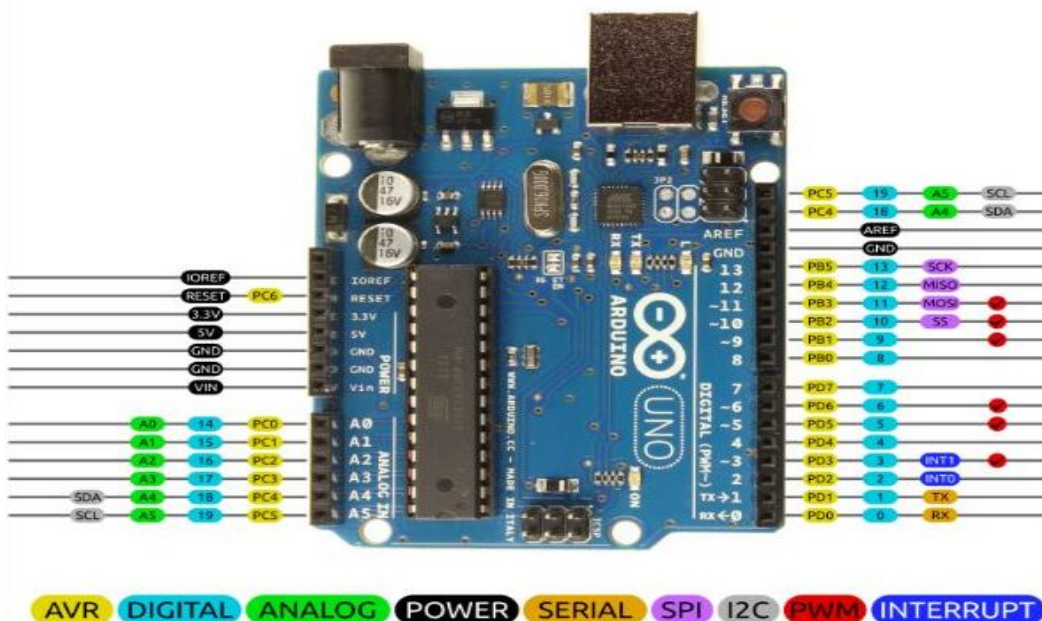


figure 22: les différents broches entrées sortie d'arduino uno

Communication Série: Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur

USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Interruptions Externes: Broches (2 et 3). Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.

Impulsion PWM (largeur d'impulsion modulée): Broches (3, 5, 6, 9, 10, et 11). Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`. Les applications de modulation de largeur d'impulsion (PWM) peuvent être trouvées dans le nombre d'applications, par exemple les télécommunications, le contrôle des servomoteurs, la régulation de la tension, la remise en puissance, etc. mesurer la largeur de PWM à l'aide d'un microcontrôleur. De plus, comment les capteurs à ultrasons (Qui peuvent être utilisés pour la mesure à distance) peuvent fonctionner conjointement avec PWM.

SPI (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI.

I2C: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou TWI - Two-Wire Interface - Interface "2 fils").

LED: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose de (06) **six entrées analogiques** (Numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de **(10 bits)** (Càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (Valeur 0) et le 5V (Valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

Note: *Les broches analogiques peuvent être utilisées en tant que broches numériques : Elles sont numérotées en tant que broches numériques de 14 à 19.*

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

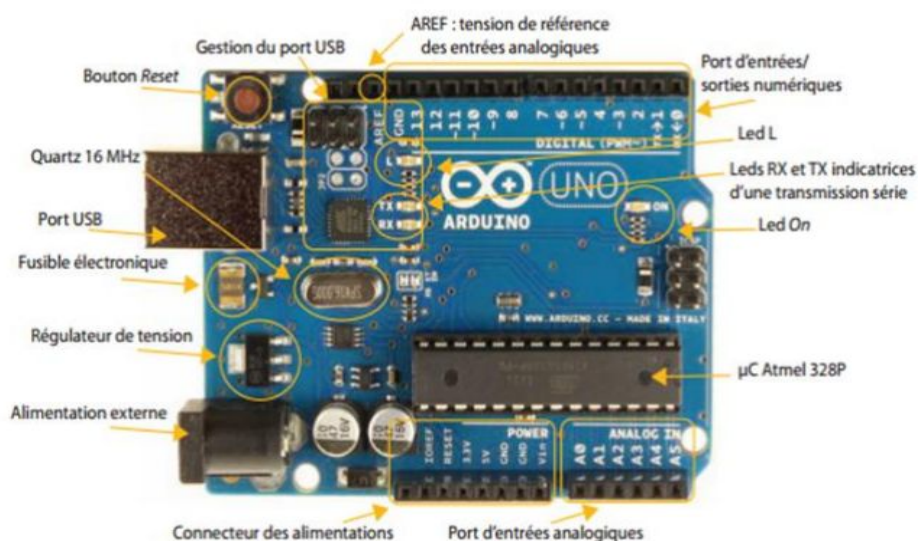


Figure 23: Constitution de la carte Arduino UNO

Les ports de communications

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série [UART](#) TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) pour transmettre les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB-vers-série de la carte, pour assurer l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignotent lorsque les données sont transmises via le circuit intégré USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Bus I2C

Le bus I2C, dont le sigle signifie [Inter Integrated Circuit](#) ce qui donne IIC et par contraction I2C.

Le protocole est initialement proposé par Philips mais adopté de nos jours par de très nombreux fabricants. C'est un bus de communication de type [série](#).

Chez certains constructeurs, ce bus est parfois nommé sous le nom de TWI ([Two Wire Interface](#)).

Ce bus n'utilise que 3 fils:

- un signal de [données](#) (SDA)
- un signal d'[horloge](#) (SCL)
- un signal de référence électrique ([masse](#))

Le bus I2C qui n'utilise que deux lignes de signal permet à un certain nombre d'appareils d'échanger des informations sous forme série avec un débit pouvant atteindre 100 Kbps ou 400 Kbps pour les versions les plus récentes.

Ceci étant précisé, voici quels sont les points forts du bus I2C:

- C'est un **bus série bifilaire** utilisant une ligne de données appelée SDA (Serial Data) et une ligne d'horloge appelée SCL (Serial Clock);
- Les données peuvent être échangées dans les deux sens sans restriction.
- Le bus est **multi-maître**.
- Chaque abonné dispose d'une adresse codée sur 7 bits. On peut donc connecter simultanément 128 abonnés d'adresses différentes sur le même bus, sous réserve de ne pas le surcharger électriquement.
- Un acquittement est généré pour chaque octet de donnée transféré.
- Le bus peut travailler à une vitesse maximum de **100 Kbps** (ou 400 Kbps) le protocole permet de ralentir automatiquement l'équipement le plus rapide pour s'adapter à la vitesse de l'élément le plus lent, lors d'un transfert.

6. Le logiciel ARDUINO : Espace de Développement Intégré (EDI)

Le logiciel ARDUINO a pour fonctions principales:

- De pouvoir écrire et compiler des programmes pour la carte ARDUINO.
- De se connecter avec la carte ARDUINO pour y transférer les programmes.
- De communiquer avec la carte ARDUINO.

Description de logiciel ARDUINO

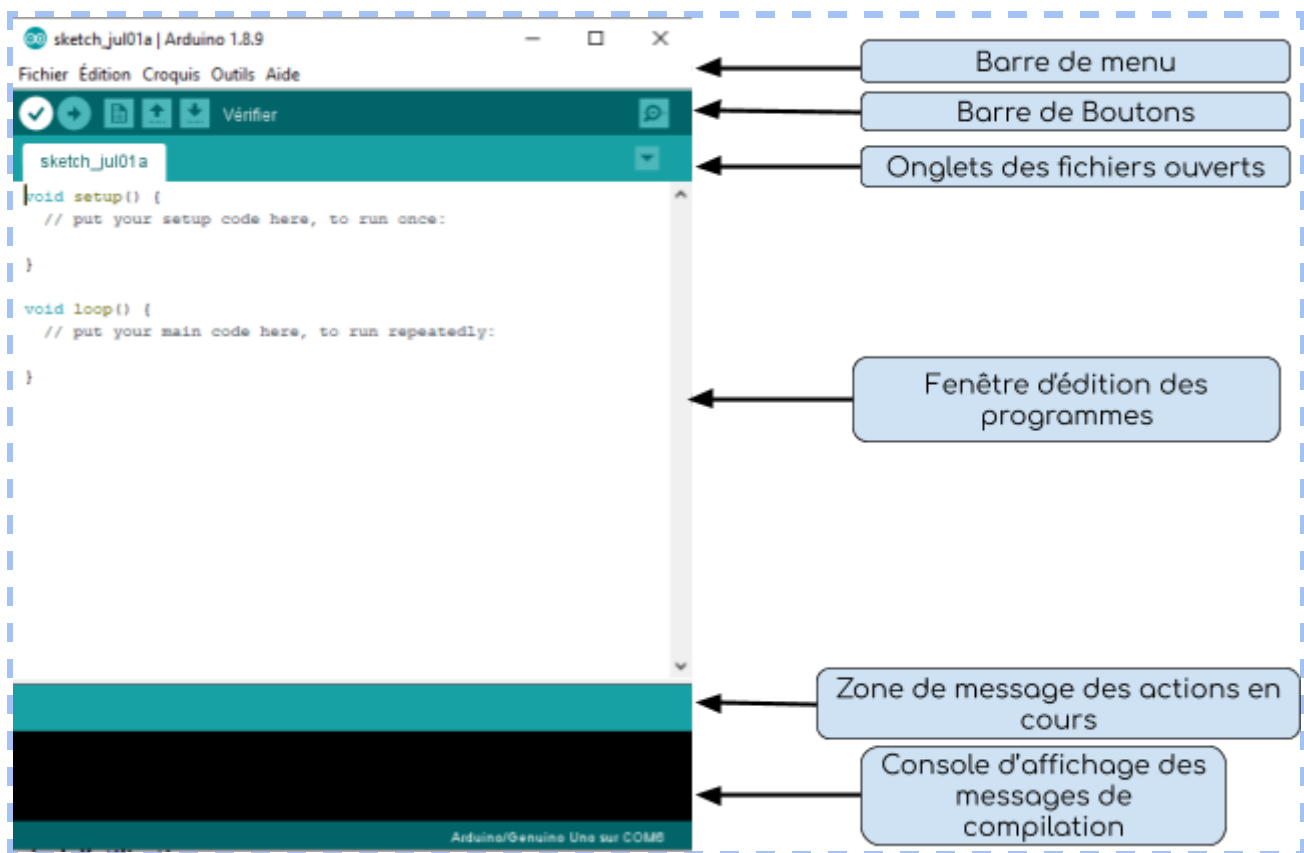


Figure 24: Interface de logiciel ARDUINO IDE

Principe général d'utilisation

Le code écrit avec le logiciel ARDUINO est appelé un programme (ou une séquence - sketch en anglais):

- Ces programmes sont écrits dans l'éditeur de texte. Celui-ci a les fonctionnalités usuelles de copier/coller et de rechercher/remplacer le texte.
- La zone de messages donne l'état de l'opération en cours lors des sauvegardes, des exportations et affiche également les erreurs.
- La console texte affiche les messages produits par le logiciel ARDUINO incluant des messages d'erreur détaillés et autres informations utiles.
- La barre de boutons vous permet de vérifier la syntaxe et de transférer les programmes, créer, ouvrir et sauvegarder votre code, et ouvrir le moniteur série.
- La barre des menus vous permet d'accéder à toutes les fonctionnalités du logiciel ARDUINO.

Le logiciel ARDUINO intègre également une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte ARDUINO. Les LED RX et TX sur la carte clignote lorsque les données sont transmises.

Le terminal série

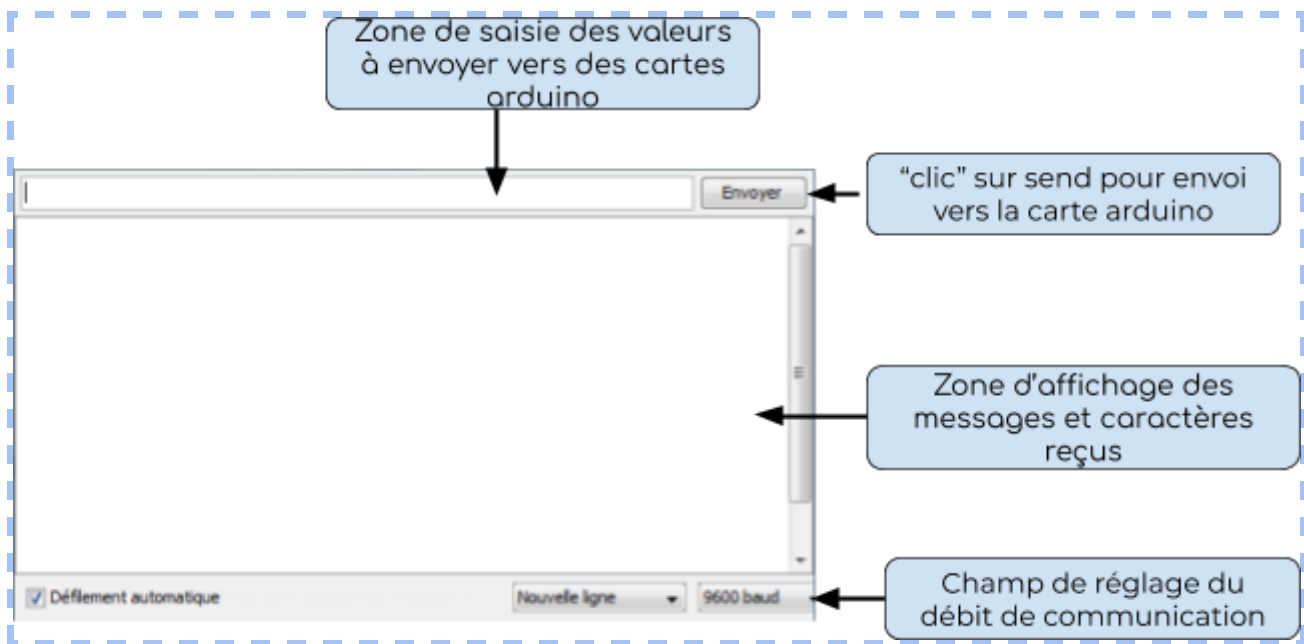


Figure 25: Interface du terminal Série

7. Exemple d'application: Contrôle d'un servo moteur avec arduino

Un servomoteur est un type de moteur électrique. C'est un dispositif typiquement utilisé en modélisation pour, par exemple, contrôler la direction d'une voiture télécommandée. Sur un servomoteur, l'angle de l'axe reste fixé dans une position et peut varier entre (0 et 180°) en fonction du signal envoyé

→ Fonctionnement

Le servomoteur est commandé par l'intermédiaire d'un câble électrique à 3 fils qui permettent d'alimenter le moteur et de lui transmettre des ordres de positions sous forme d'un signal codé en largeur d'impulsion plus communément appelés PWM (Pulse Width Modulation ou Modulation de Largeur d'Impulsion) ou RCO (Rapport Cyclique d'Ouverture).

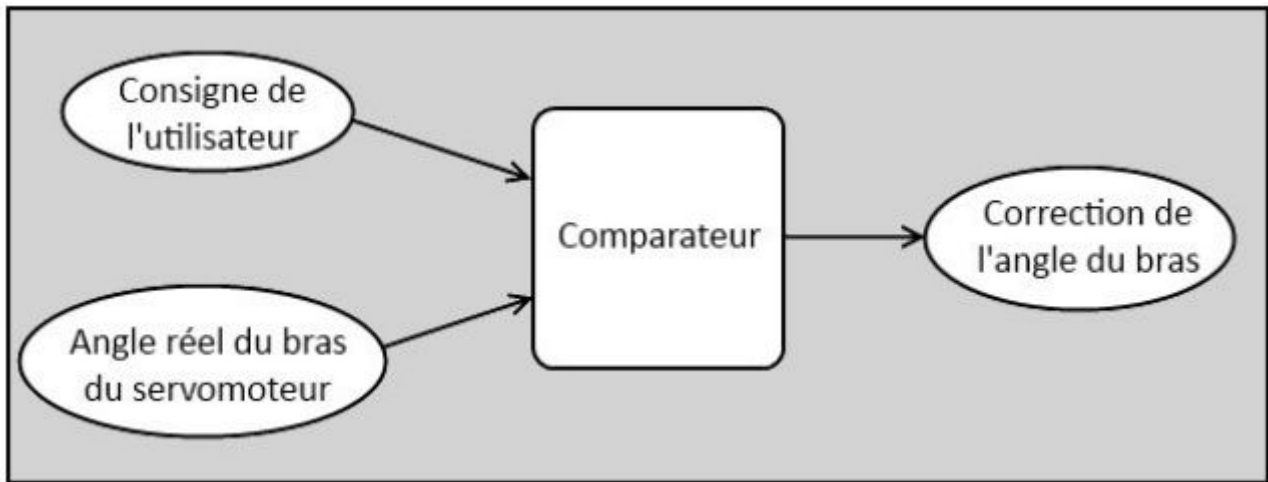


Figure 26: Synoptique de fonctionnement de l'asservissement du servomoteur

La durée de l'état HAUT

Pourquoi est-ce si important ? Qu'avons-nous à savoir sur la durée de l'état HAUT du signal PWM ? À quoi cela sert-il, finalement ?

Cette durée est ce qui compose l'essentiel du signal. Car c'est selon elle que le servomoteur va savoir comment positionner son bras à un angle précis.

Nous connaissons bien comment fonctionne un signal [PWM](#), qui sert également à piloter la vitesse d'un moteur à courant continu. c'est quelque peu semblable. En fait, un signal ayant une durée d'état HAUT très faible donnera un angle à 0°, le même signal avec une durée d'état HAUT plus grande donnera un angle au maximum de ce que peut admettre le servomoteur. Précisément l'état HAUT est limité par des valeurs bien précises, qui sont entre une valeur de 1 ms au minimum et au maximum de 2 ms pour les servos standards.

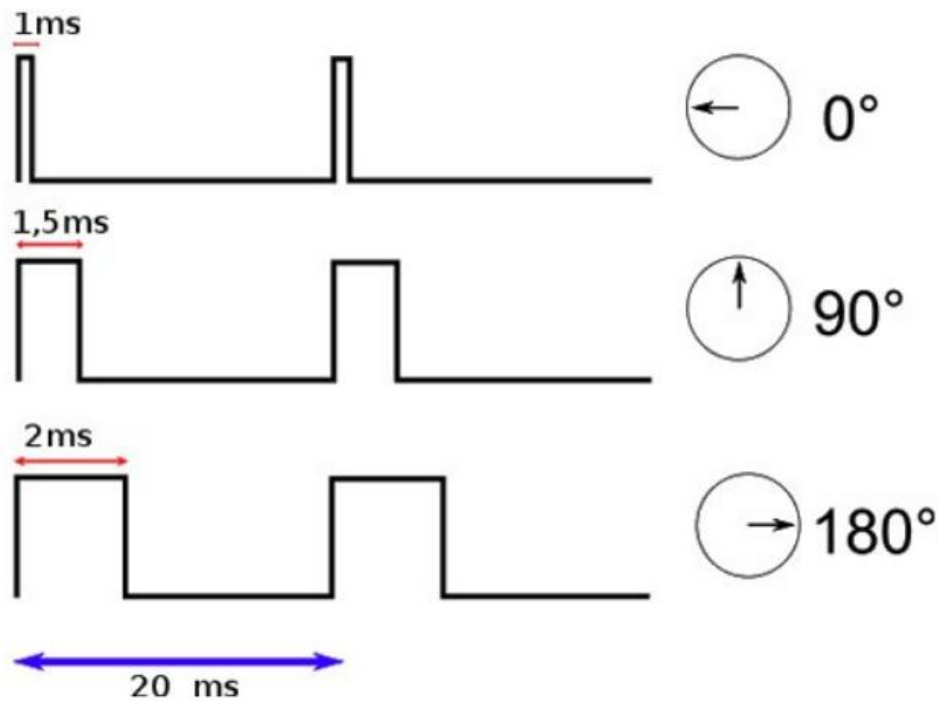


Figure 27: Position en fonction de la pulsation

Vous aurez deviné, à travers cette illustration, que la durée de l'état HAUT fixe la position du bras du servomoteur à un angle déterminé.

Apparence

On en trouve de toutes les tailles et de toutes les puissances. La plupart du temps la sortie peut se positionner entre 0 et 180°. Cela dit, il en existe également dont la sortie peut se débattre sur seulement 90° et d'autres, ayant un plus grand débattement, sur 360°. Ceux qui ont la possibilité de faire plusieurs tours sont souvent appelés servo-treuil. Enfin, les derniers, qui peuvent faire tourner leur axe sans jamais se buter, sont appelés servomoteurs à rotation continue. Les servomoteurs sont très fréquemment employés dans les applications de modélisme pour piloter le safran d'un bateau, le gouvernail d'un avion ou bien même les roues d'une voiture téléguidée dont on a parlé jusqu'à présent. Maintenant que les présentations sont faites, mettons-le à nu ! Il est composé de plusieurs éléments visibles ... :

- Les fils, qui sont au nombre de trois (nous y reviendrons)
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal
- Le boîtier qui le protège
 - ... mais aussi de plusieurs éléments que l'on ne voit pas :
- Un moteur à courant continu
- des engrenages pour former un réducteur (en plastique ou en métal)
- un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent)
- une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu

Voilà une image 3D de vue de l'extérieur et de l'intérieur d'un servomoteur :

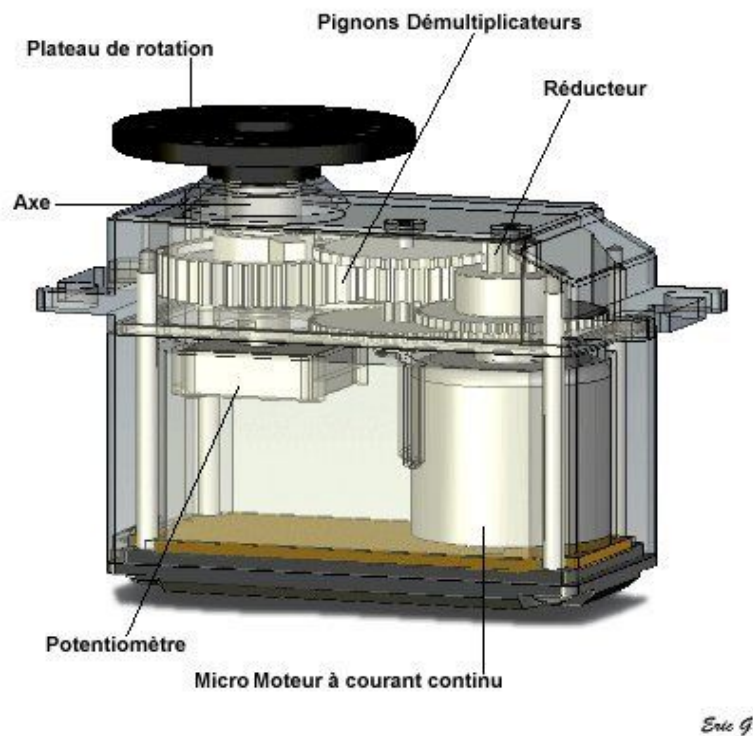
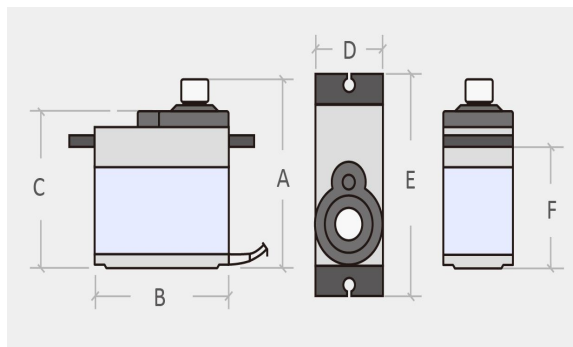


Figure 28: Vue interne d'un servomoteur

→ Caractéristiques



Weight(g)	13.4
Torque(kg)(4.8v)	1.8
Speed(sec/60deg)	0.1
A(mm)	32.5
B(mm)	22.8
C(mm)	28.4
D(mm)	12.4
E(mm)	32.1
F(mm)	18.5

Spécifications

- Poids: 13,4 g
- Dimension: 22,5 x 12 x 35,5
- Couple de décrochage: 1,8 kgf · cm (4,8V)
- Vitesse de fonctionnement: 0,1 s / 60 degrés
- Tension de fonctionnement: 4,8 V -
- Largeur de la bande morte: 5 μ s

Connectique

Le servomoteur a besoin de trois fils de connexion pour fonctionner. Deux fils servent à son alimentation, le dernier étant celui qui reçoit le signal de commande:

- rouge : pour l'alimentation positive (4.5V à 6V en général)
- marron : pour la masse (0V)
- orange: entrée du signal de commande

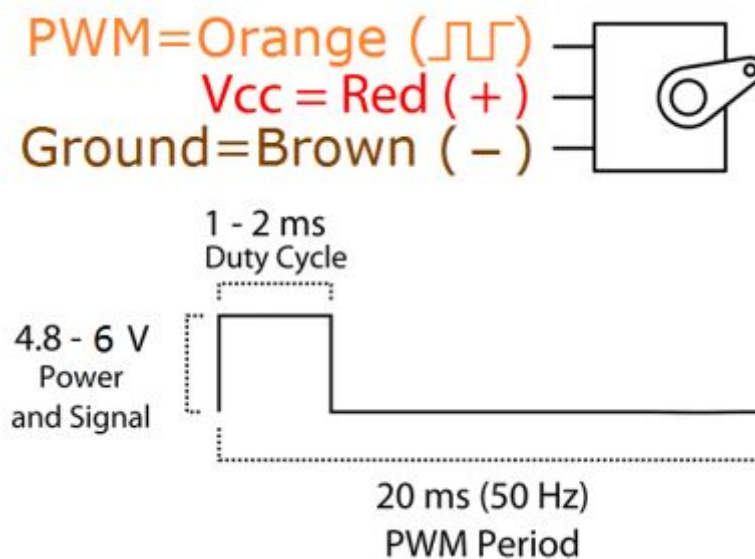


figure 29: Signal de commande par PWM

En fin de compte on y contrôle le moteur en téléversant le code ci dessous dans la carte Arduino et en faisant les branchements nécessaires entre la carte et le moteur: (voir figure 30)

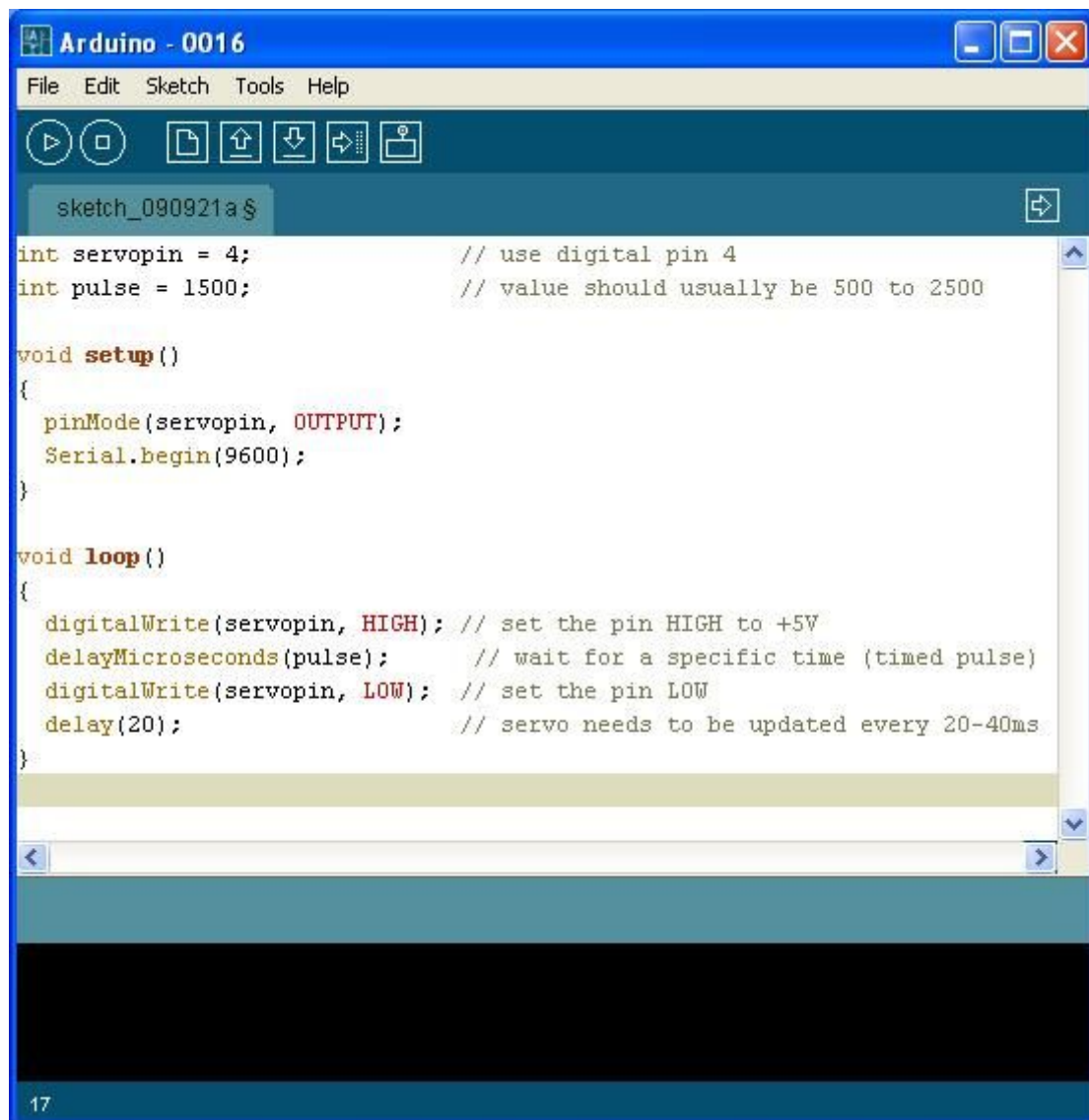


Figure 30: Exemple du contrôle du moteur servo par une carte Arduino

Chapitre 3

Conception matérielle et logicielle du projet

I. Introduction

Avec le développement de matériel et de logiciel embarqué, diverses technologies sont de plus en plus intégrées. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, en réduisant ainsi le coût de fabrication d'un produit. et en résulte des systèmes plus complexes et performants pour un espace réduit. Au cours des dernières années, le mouvement open source du matériel est populaire dans le monde [13]. Arduino est un chef de file dans ce mouvement d'où les groupes d'utilisateurs répartis des ingénieurs aux étudiants, puis aux élèves de collège ou même les enfants de l'école primaire. L'émergence d'une variété de plates-formes matérielles open source réduit considérablement la courbe d'apprentissage, stimule l'innovation et accélère la conversion de l'idée à la réalisation.

II. Cahier de charges

A. Introduction

Le but de ce projet est d'utiliser un télémètre monté dans des servos, qui tournerait et lui permettrait de mesurer différentes parties de la pièce. Les angles et les distances sont envoyés en temps réel à l'ordinateur qui lit, convertit et crée une scène 3D du lieu. En fin de compte, cela fonctionne comme un scanner 3D de la pièce et de tout objet à l'intérieur.

Nous avons d'abord essayé d'utiliser un télémètre sonar pour effectuer les mesures de distance, mais même avec un sonar à faisceau étroit, il n'était pas assez étroit pour permettre des lectures précises. De plus, les ondes émises par le sonar rebondissent sur des surfaces inclinées et donnent des erreurs de lecture. Pour surmonter ce problème, nous avons utilisé un LIDAR (télémètre laser) très précis, avec un faisceau extrêmement étroit et qui ne rebondit pas sur les surfaces. Le seul inconvénient est qu'elle n'est pas aussi précise avec les surfaces brillantes et ne permet pas de lire contre des surfaces transparentes, telles que les fenêtres, mais c'est assez bien pour le but recherché.

B. Le projet

Le projet peut être divisé en trois parties: l'électronique, la programmation Arduino et la programmation par interface.

1. Électronique

Pour construire le prototype, nous avons utilisé deux servomoteurs (modèle MG90S Tower Pro), capables de pivoter à 180 degrés et de suivre leur position. En tant que télémètre, nous avons utilisé un LIDAR-Lite, le LIDAR le plus abordable que nous ayons pu trouver, mais qui reste un excellent produit pour son prix. Tout était alimenté par

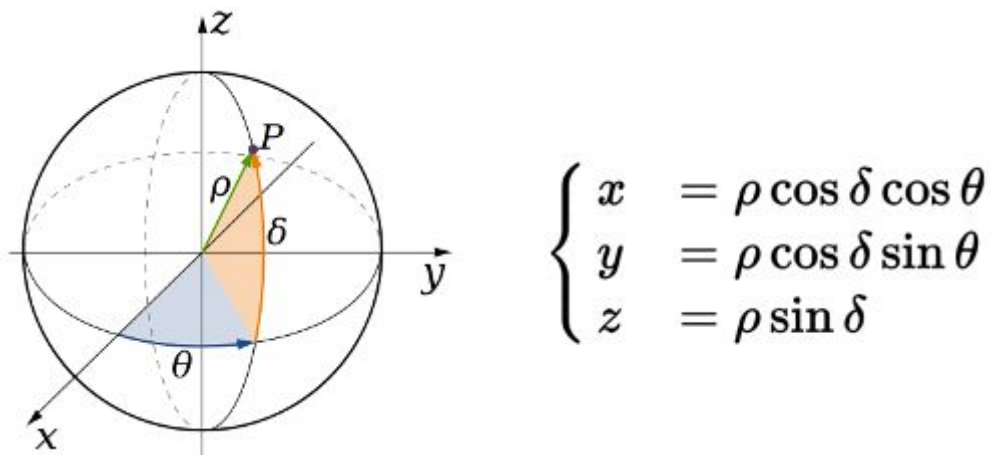
power bank et contrôlé par un arduino. Les schémas de branchement va être énoncé ci dessus.

2. Arduino

Le code géré par l'arduino est assez simple. Il déplace les servos (en gardant la trace de sa position) et lit la distance du capteur après chaque mouvement. Chaque fois qu'il lit la distance, une chaîne est envoyée via le port série avec des informations provenant de la distance et de la position des servos. L'un des problèmes rencontrés était que la lecture du capteur à l'intérieur de boucles imbriquées créerait un comportement inattendu d'arduino. La logique a donc dû être modifiée pour itérer sur la position des servos à l'aide de la fonction loop() intégrée de arduino.

Grâce aux bibliothèques Arduino Servo.h et LIDARLite.h , le code permettant de contrôler ces éléments est beaucoup plus simple.

Le micro-contrôleur ne connaît que les angles des servos et la distance à l'obstacle. Une petite quantité de calcul est nécessaire pour convertir les informations d'angle de lacet, d'inclinaison de pas et de distance en coordonnées X, Y et Z beaucoup plus utilisables.



À l'aide de ces formules de conversion de système de coordonnées sphériques à cartésiennes, il affiche la position du point sur le port série.

3. L'interface

Méthode 1: Affichage à l'aide d'un logiciel développé avec Microsoft Visual Studio

L'interface informatique du prototype, écrite en C ++ et OpenGL, lit les données envoyées par l'arduino, les analyse, les convertit en coordonnées cartésiennes et les courbes à l'écran. Chaque point est créé à l'écran sous la forme d'un cube dont la couleur varie en fonction de la distance et des angles. L'interface est également chargée de communiquer avec arduino et d'envoyer un code de départ en indiquant au prototype de lancer la numérisation. Toutes les numérisations sont affichées à l'écran en temps réel, à mesure que les distances sont lues et transmises. Il est possible de naviguer dans la vue 3D pendant que la numérisation avance. Le logiciel permet d'enregistrer le scan dans un fichier et de le charger plus tard. Un aperçu des fichiers enregistrés est disponible pour faciliter la sélection de l'analyse souhaitée.

Méthode 2: Processing

Logiciel de création multimédia, Processing possède la particularité d'utiliser des instructions informatiques pour dessiner, réaliser des animations en 2 ou 3 dimensions, créer des œuvres sonores et visuelles, concevoir des objets communicants qui interagissent avec leur environnement.

Processing est un logiciel libre et gratuit. Il fonctionne sur les plates-formes Windows, Linux, Mac (et sur toute autre plate-forme pouvant faire fonctionner des logiciels conçus en Java). Il lit les données envoyées par l'arduino, les analyse, acquisition et la visualisation des données.

Afin de collecter, afficher et utiliser la tourelle, LidarViewer lit le port série pour obtenir les données; l'affiche sous forme de nuage de points 3D que nous pouvons zoomer, faire un panoramique, faire pivoter et déplacer; et l'enregistre dans un fichier afin que nous puissions l'utiliser ultérieurement.

Structure du système

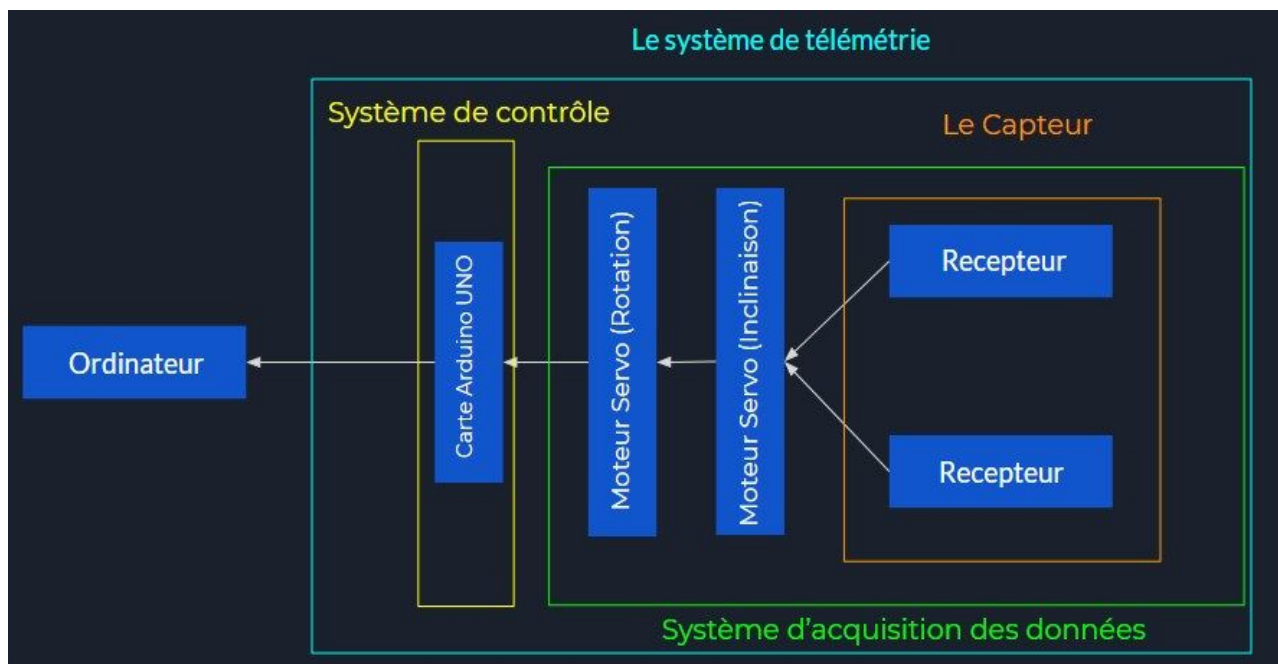


Figure 31: Structure du système de détection

Concept et prototypes

- Utilisation d'un lidar comme télémètre pour créer un modèle 3D d'une pièce
- Servos pour déplacer le capteur dans un chemin sphérique.
- Affichage 3D interactif sur ordinateur.

Défis

- Mesures précises
- Communication en C ++
- Source de courant
- Utiliser OpenGL (Pour méthode 1)

- Conversion de coordonnées
- contrôle des deux servo

Solutions

- Lidar-LITE v3
- Chaîne de longueur fixe et lecture en morceaux
- Alimentation externe

III. Matériel

Après une étude et recherche sur les matériels disponibles sur le marché marocain ayant répondant aux conditions imposées par le cahier de charges on s'est décidé d'utiliser les matériels suivants pour réaliser notre expérience

Carte de commande: Arduino Uno


La carte Arduino Uno est une carte à microcontrôleur construite autour de l'ATmega328. Elle possède 14 broches d'entrée/sortie numériques (dont 6 peuvent servir de sorties PWM), 6 entrées analogiques, d'un oscillateur à quartz de 16 MHz, un connecteur USB, un jack d'alimentation, une embase ICSP, et un bouton d'initialisation (reset). La carte Uno contient tout ce qui est nécessaire au fonctionnement du microcontrôleur. Pour l'utiliser, il suffit de la relier à un ordinateur avec un câble USB, ou encore de l'alimenter à l'aide d'un bloc secteur externe ou de piles.



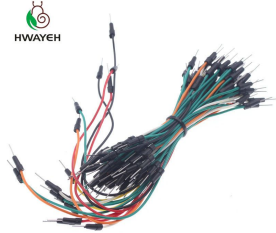
- version: Rev. 3
- alimentation:
 - via port USB ou
 - 7 à 12 V sur connecteur alim
- dimensions: 5,5 x 2,1 mm
- microprocesseur: ATmega328
- mémoire flash: 32 kB
- mémoire SRAM: 2 kB
- mémoire EEPROM: 1 kB
- 14 broches d'E/S dont 6 PWM
- 6 entrées analogiques 10 bits
- intensité par E/S: 40 mA
- cadencement: 16 MHz
- bus série, I2C et SPI
- gestion des interruptions
- fiche USB B
- dimensions: 74 x 53 x 15 mm

	Prix : 120,00 Dh
--	------------------

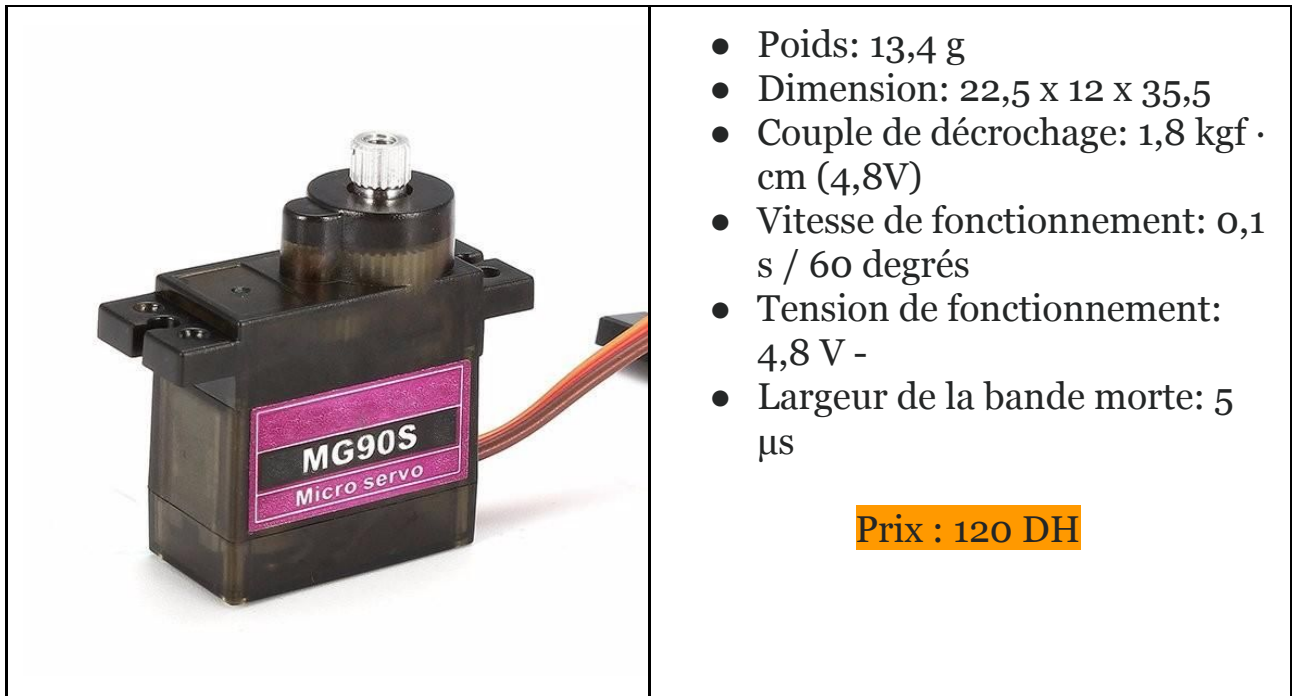
Breadboard:

	<p>Plaque d'essais à connexions sans soudure (breadboard), au pas de 2,54 mm.</p> <p>Prix: 15 dhs</p>
---	---

Câbles:

	<p>Pour la connexion entre les composantes</p>
--	--

Servo moteur: MG90S Tower Pro



Actionneurs pour la rotation du lidar. Bien que ces moteurs faibles constituent un goulot d'étranglement majeur pour le système, on les a utilisés simplement parce que nous les avons déjà. Des servos plus puissants et moins "instables" amélioreraient certainement la qualité et la rapidité de la capture.

Sur le plan matériel, le scanner est plutôt basique: deux micro-servomoteurs de 9 g contrôlent les axes de rotation du **roulis** et du **pas** . Les supports qui les maintiennent sont imprimés en 3D

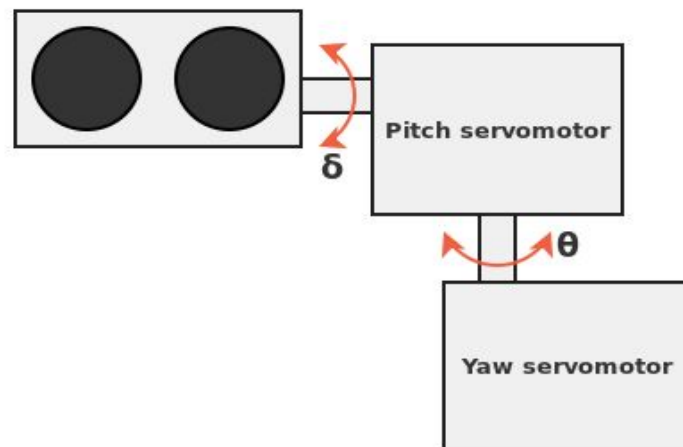


Figure 32: Les supports qui les maintiennent LIDAR et les deux servos

Capteur: LIDAR-Lite v3

	<ul style="list-style-type: none"> • Dimensions de l'appareil (H x L x P) : 20 x 48 x 40 mm • Poids : 22 g • Résolution : 1 cm • Précision : +/- 2,5 cm à plus d'un mètre. Consultez le manuel pour connaître les caractéristiques de fonctionnement complètes. • Portée : de 5 cm à 40 mètres • Taux de mise à jour : jusqu'à 500 Hz • Interface : I2C ou PWM • Alimentation (tension de fonctionnement) : 4,75-5,5 V c.c. ; 6 V maximum • Consommation actuelle : 105 mA inactif ; 130 mA en utilisation continue • Température de fonctionnement : de -20 à 60 °C • Longueur d'onde du laser/pic de puissance : 905 nm/1,3 Watts • Divergence du faisceau : radian 8 m • Ouverture optique : 12,5 mm <p>Prix : 2 800 dh</p>
---	--

Vue d'ensemble

- Capteur optique de mesure des distances haute performance
- Capteur de proximité et de longue portée fiable et puissant pour les drones, les robots ou les véhicules sans pilote
- Compact, léger et économe en énergie
- Les paramètres modifiables par l'utilisateur permettent un réglage de la précision, de la portée et du temps de mesure
- Communication via I2C et PWM
- Nécessite une source d'alimentation et un microcontrôleur externe exécutant une application

Remarque

1. *Nécessite une source d'alimentation et un microcontrôleur externe exécutant une application.*
2. *LIDAR-Lite est un appareil laser de Classe 1 qui n'est pas dangereux pour les yeux lorsqu'il est utilisé normalement, dans toutes les conditions. Cela signifie que vous pouvez regarder le laser à l'œil nu, mais qu'il ne faut pas dépasser l'exposition maximale permise. Cependant, si vous utilisez l'appareil LIDAR-Lite sans boîtier ni lentille ou si vous modifiez ces éléments de manière à exposer le laser, vous risquez d'être exposé à des rayonnements laser et pourriez subir des dommages oculaires permanents. LIDAR-Lite ne contient aucune pièce pouvant être entretenue par l'utilisateur*

On a choisi les matériels cités ci dessus car ils étaient valables dans le marché pendant la periode de realisation du projet sans mentionner qu'ils satisfassent le conditions imposés par le cahier de charges en terme de prix et performance.

III. Schémas et code Arduino

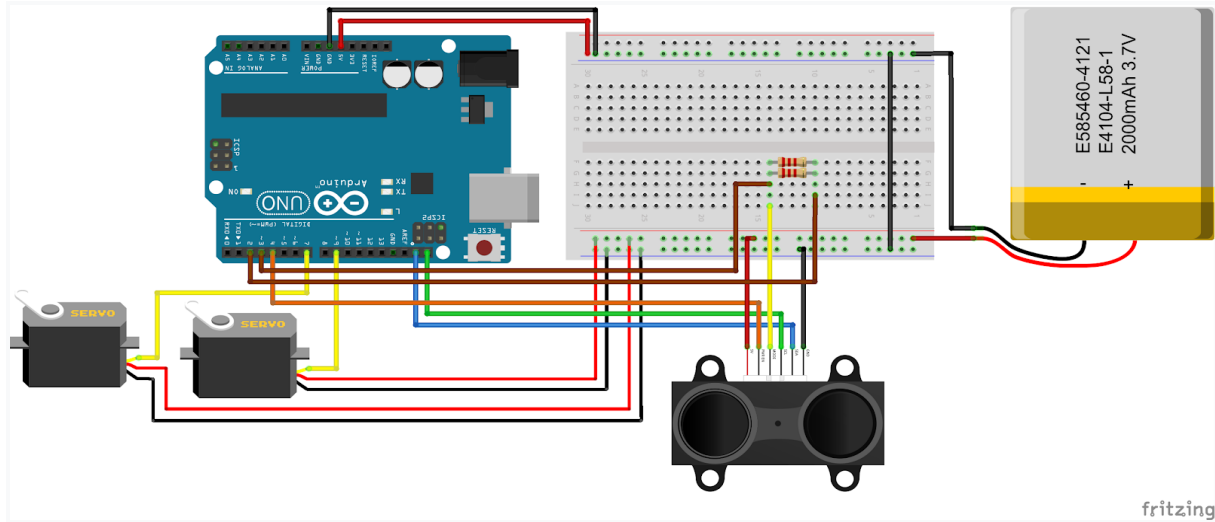
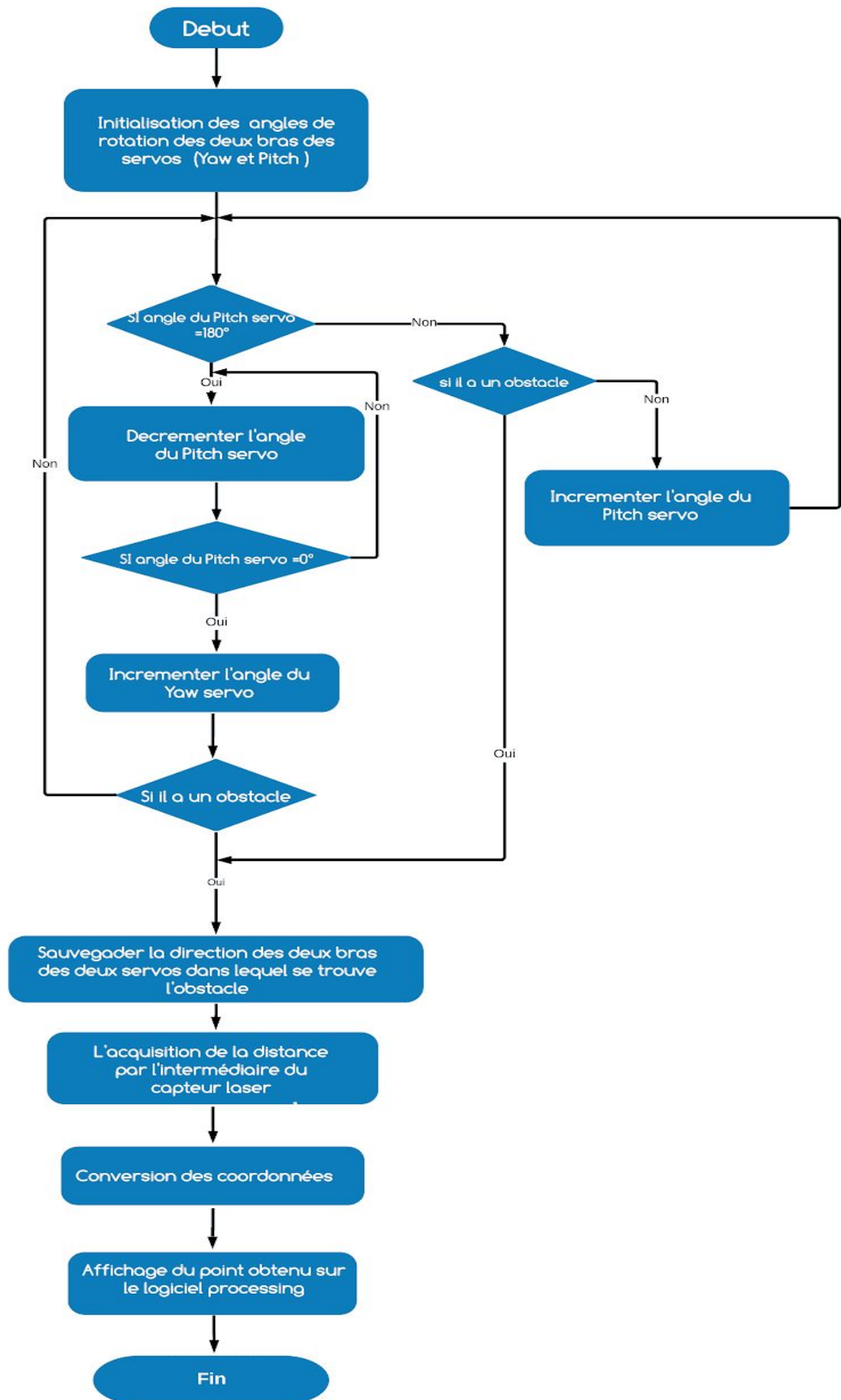


Figure 33: Branchement de tous les éléments du montage

Le code arduino est assez simple: déplacer les servos et lire à partir du capteur. Au cours du développement, deux problèmes principaux ont été trouvés. La lecture du capteur à l'intérieur des boucles imbriquées ne fonctionnait pas correctement, ce qui a provoqué un comportement inattendu des servos. Une solution consistait à utiliser des conditions dans la boucle arduino () pour faire bouger les servos. Un autre problème était la communication avec le PC: il nous a fallu un certain temps pour comprendre que la chaîne transmise par le port série devait avoir une longueur fixe, ce qui a donc permis d'ajouter des zéros à gauche aux angles et aux valeurs du capteur.

Organigramme Du Projet



IV. Logiciel

Méthode 1: Avec Microsoft Visual Studio

Il est difficile d'entrer dans les détails de le logiciel utiliser pour affichage, mais nous pouvons au moins expliquer comment cela fonctionne.

Les premières lignes de codes créent tous nos objets, à savoir: Fenêtre de rendu, un gestionnaire de saisie, un gestionnaire de fenêtres, un menu et un tableau d'objets à afficher à l'écran.

Comme il s'agit d'un projet assez petit (environ 1100 lignes de code), nous avons décidé de laisser la boucle dans le fichier principal, mais nous avons essayé de le garder simple et propre.

Notre boucle principale est assez simple. Il vérifie quel est l'état actuel du programme pour décider quoi faire. Donc, fondamentalement, la boucle principale ressemble à ceci (ce n'est pas le code complet mais il peut donner une idée de ce qui se passe):

```
1  while (running)
2      {
3          //Update objets
4          manager.update();
5          input.update();
6          //Clear scene
7          glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
8          switch (state) {
9              case RUNNING:
10                 ...
11                 Draw manager
12                 break;
13             case MENU:
14                 Draw Menu
15                 break;
16             case EXIT:
17                 Exit
18                 break;
19         }
20         //Check if user has exited the program by clicking the X button
21         if (input.getExit())
22         {
23             running = false;
24         }
25
26         //Swap buffers and display the backbuffer on the screen
27         window.display();
28
29     }
30 }
```

L'objet Input Manager est responsable de tout ce qui concerne IO. Cela signifie qu'il est responsable de la sauvegarde des fichiers, du chargement des fichiers, de la lecture des frappes de touche, de la lecture de la série et de tout ce qui concerne l'IO. Lorsqu'un objet

doit appeler une fonction IO, il appelle une méthode sur InputManager qui renvoie le résultat souhaité.

WindowManager dessine la vue 3D à l'écran et permet à l'utilisateur d'interagir avec cette vue. Même lorsque l'utilisateur est dans le menu, cet objet est chargé de dessiner la vue principale de la scène. Il configure également les paramètres OpenGL au démarrage du programme et modifie les paramètres à adapter pour les vues 3D et 2D.

Pour afficher les données à l'écran, nous utilisons des cubes. Nous avons la classe Cube qui a la position, la taille et la couleur utilisées pour stocker toutes les informations de cubes. Tous les cubes affichés dans la vue 3D sont conservés dans un tableau que la fonction principale crée. Tout ce que les données d'un fichier vont être lues, ce tableau est transmis par référence à InputManager qui efface les données de prévisualisation et se charge avec les informations du fichier.

Le menu est composé d'une interface graphique qui conserve tous les éléments GUIE. Une interface graphique est essentiellement une classe qui gère tous les GUIElements. Chaque GUIElement a une position, une taille, du texte et d'autres attributs pouvant être partagés entre tous les types d'éléments tels que les boutons, les cadres (pour afficher les fichiers sauvegardés) et les panneaux. Le menu vérifie si un bouton est enfoncé et déclenche une fonction en conséquence.

Un GUIElement particulier est le GUIFrame qui affiche également une vue 3D. Cet élément affiche un aperçu du fichier enregistré afin que l'utilisateur puisse visualiser le fichier avant de le charger dans la vue principale. Lorsque vous utilisez des vues 2D et 3D dans OpenGL, vous devez vous occuper de Matrix Mode et de la perspective. Ainsi, chaque fois qu'une image est affichée, elle dessine d'abord son composant 2D, puis modifie les paramètres OpenGL en 3D pour dessiner l'aperçu du fichier.

Nous avons commencé à implémenter une fonctionnalité qui permet à l'utilisateur de «se promener» dans la salle numérisée en 3D à l'aide de la vue classique à la première personne, mais nous n'avons pas eu le temps d'aller plus loin car elle était assez complexe sous OpenGL.

Les bases

Afin de visualiser toutes les données collectées à partir du capteur de distance, nous avons utilisé un logiciel permettant de visualiser toutes les mesures dans un espace 3D.

L'idée derrière le logiciel est assez simple

1. L'arduino lit le capteur et envoie les données en série au PC. Les données envoyées à l'ordinateur sont constituées de la distance et de l'angle de chaque moteur pas à pas.
2. Avec une bibliothèque de séries C ++, nous lisons les données de série par lots. Comme nous utilisons uniquement le câble USB pour lire à partir du port série, nous devons utiliser des solutions de contournement afin d'obtenir les données sans aucun bruit. Il serait préférable d'acheter un adaptateur USB / port série.

3. Le logiciel lit les données et change le système de coordonnées en cartésien normal. Cela peut être fait facilement car les données de l'arduino sont en coordonnées sphériques.
4. Toutes les données de toutes les mesures sont enregistrées dans un fichier et affichées à l'écran.

Pour l'instant, nous affichons simplement les données et construisons un logiciel robuste. Après avoir terminé tous les détails du logiciel, nous allons essayer de travailler sur les données pour appliquer des filtres et créer une scène 3D plus réaliste.

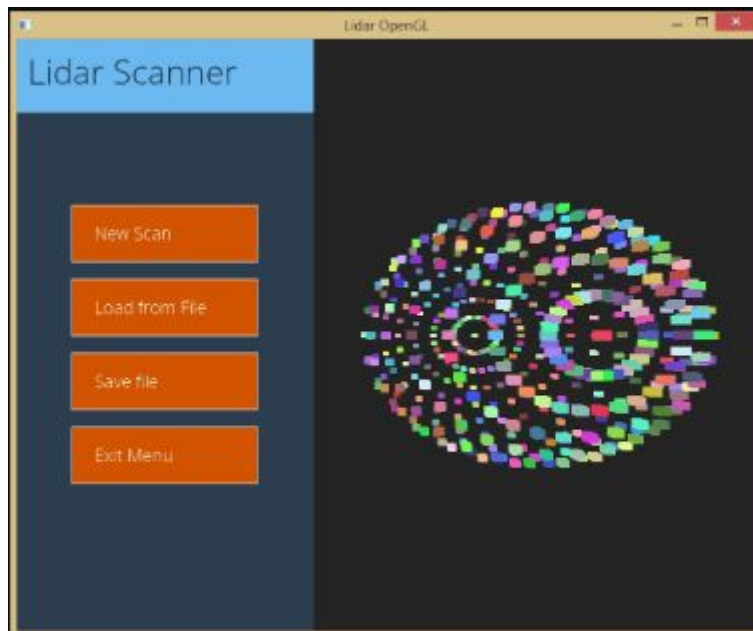


Figure 34: Interface du logiciel

Si vous choisissez de charger un fichier, il en affichera un aperçu avant de le charger dans la scène principale. Si vous enregistrez un fichier, il sera automatiquement répertorié avec tous les fichiers de prévisualisation.

Le mode de capture est également en temps réel. Vous pouvez voir toutes les données tracées à l'écran pendant la capture.

Le logiciel était délicat à compiler car nous utilisons OpenGL et C++. Il pourrait être plus facilement implémenté en utilisant une bibliothèque d'interface de haut niveau. Nous utilisons des cubes pour afficher les mesures car ce sont les plus similaires aux points dans un espace 2D. Nous stockons également tout dans des tableaux pour faciliter le travail et accéder rapidement aux éléments.

Méthode 2: Avec Processing

Réalisation du montage et vérification pratique

❖ Etape 1

Avant d'envoyer le programme Blink qui est déjà préparé (voir annexe) vers la carte, il faut dire au logiciel quel est le nom de la carte et sur quel port elle est branchée.

Choisir la carte que l'on va programmer. Ce n'est pas très compliqué, le nom de votre carte est indiqué sur elle. Pour nous, il s'agit de la carte "Uno". Allez dans le menu Tools ("outils" en français) puis dans Board ("carte" en français). Découverte de l'Arduino Vérifiez que c'est bien le nom "Arduino Uno" qui est coché. Si ce n'est pas le cas, on le coche.

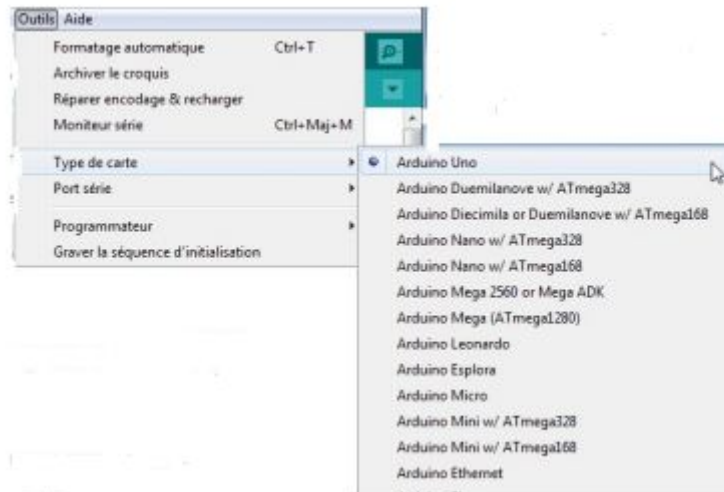


Figure 35: Menu de sélection de la carte cible

Choisissez le port de connexion de la carte. Allez dans le menu Tools, puis Serial port. Là, vous choisissez le port COMX, X étant le numéro du port qui est affiché. Ne choisissez pas COM1 car il n'est quasiment jamais connecté à la carte. Dans mon cas, il s'agit de COM9

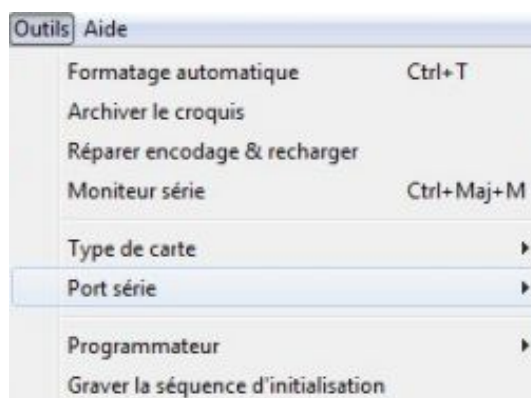



Figure 36: Menu de sélection du port com

Remarque: Pour trouver le port de connexion de la carte, vous pouvez aller dans le gestionnaire de périphériques qui se trouve dans le panneau de configuration. Regardez à la ligne Ports (COM et LPT) et là, vous devriez avoir Arduino Uno (COMX).

Maintenant, il va falloir envoyer le programme dans la carte. Pour ce faire, il suffit de cliquer sur le bouton *Téléverser* .

◆ Etape 2

Après avoir transmis le code arduino vers la carte Arduino Uno et vérifier son fonctionnement. Afin de collecter, afficher et utiliser la tourelle, nous avons créé un logiciel compact avec notre code arduino en utilisant Processing (le code est bien expliquer dans annexe) C'est un framework Java open-source très pratique conçu pour les travaux

graphiques. LidarViewer (Nom du projet processing) lit le port série pour obtenir les données; l'affiche sous forme de nuage de points 3D que nous pouvons zoomer, faire un panoramique, faire pivoter et déplacer.

L'enregistrement des données se fait à l'aide la commande 's' (bouton du clavier) dans un fichier afin que nous puissions l'utiliser ultérieurement .

```

93 // Draw point with a variable color
94 stroke(255-v.z, 255-v.y, 255-v.x);
95 point(v.x, v.y, v.z);
96 }
97 }
98
99 // Handle keyboard events : movement, save & clear all points.
100 void keyPressed() {
101   if (key == 'x') {
102     // erase all points
103     pointList.clear();
104   } else if (key == 's') {
105     saveToFile();
106   }
107   else if (key == CODED) {
108     if (keyCode == UP) {
109       cam.rotateX(angleIncrement);
110     } else if (keyCode == DOWN) {
111       cam.rotateX(-angleIncrement);
112     } else if (keyCode == LEFT) {

```

figure 37: Commande qui permet d'enregistrer les mesures

Ce fichier peut être utiliser dans 'Meshlab' qui donne un représentation tridimensionnelle bien traiter et filtrer.

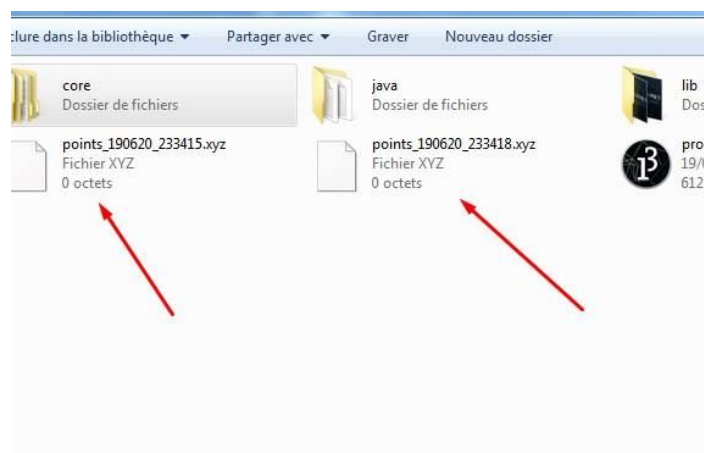


Figure 38: Forme du fichier enregistré

❖ Etape 3

Maintenant, il va falloir d'attendre le scanner après de cliquer sur le bouton  pour actionner le démarrage du programme PROCESSING .

Résultat du scanner

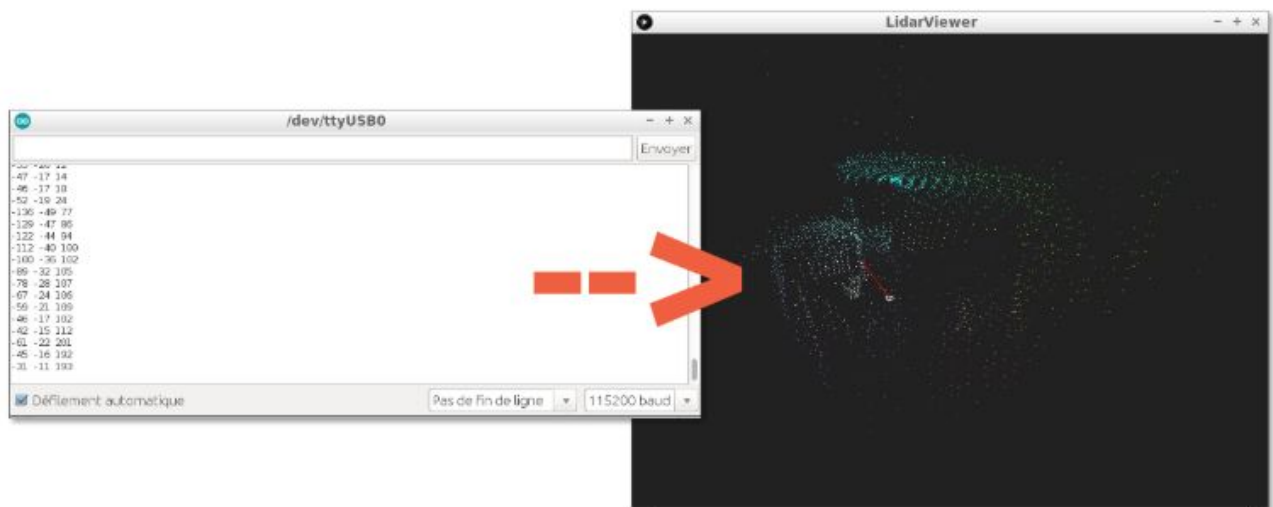


figure 39: Représentation des mesures dans le plan du plateforme

V. Résultats

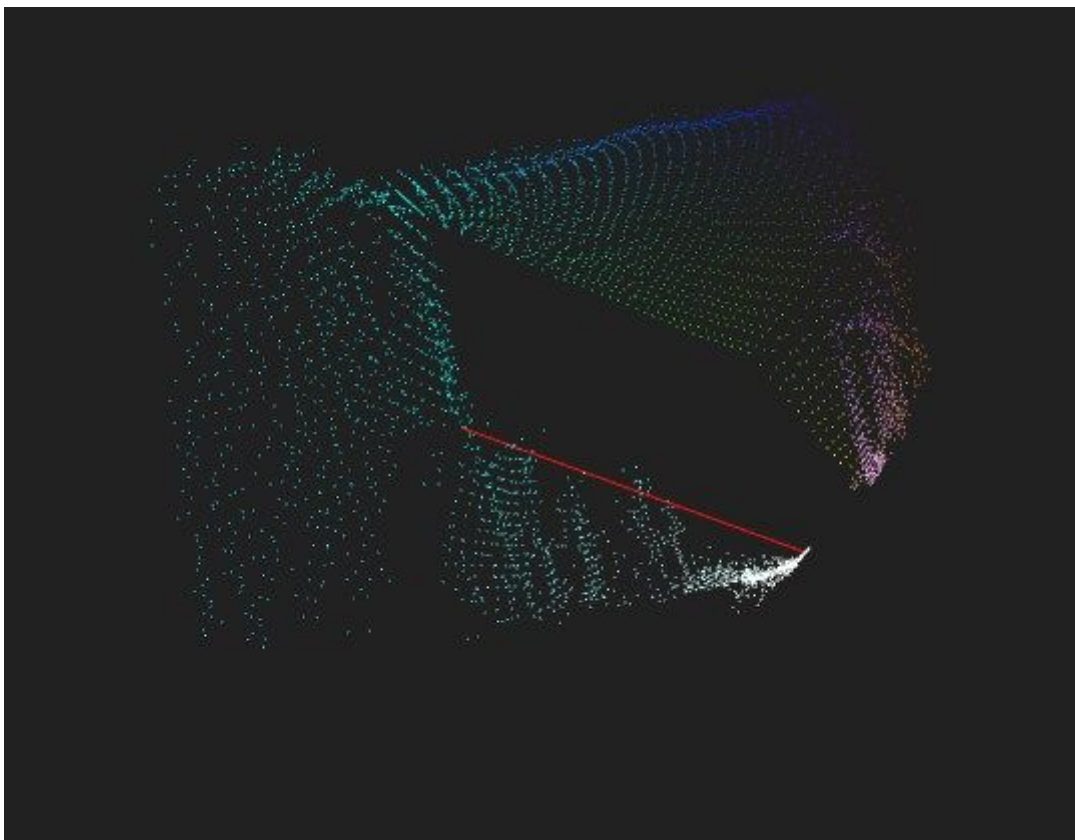


Figure 40: Exemple du nuage de points 3D obtenus pendant le scan

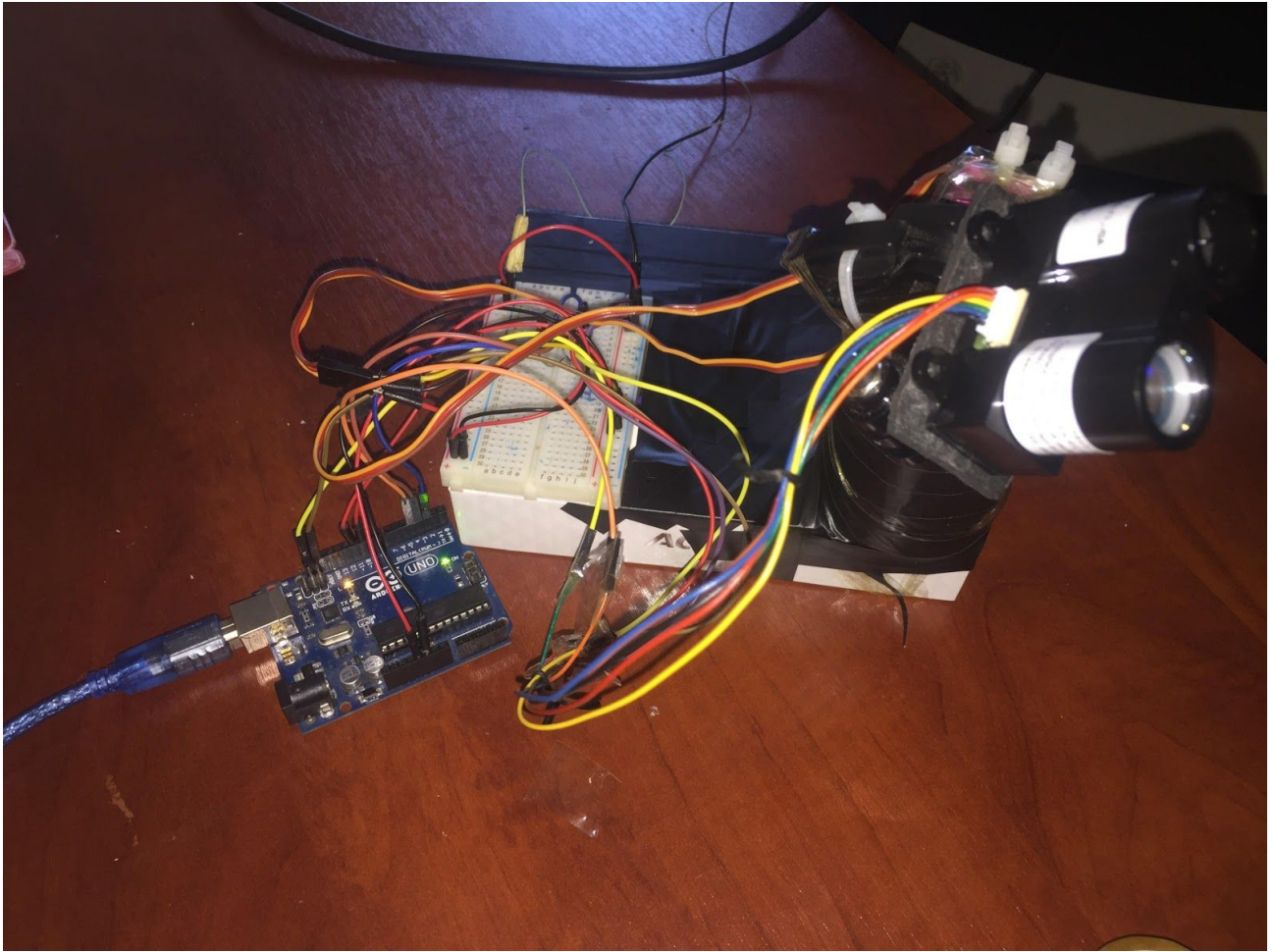


Figure 41: Le prototype réalisé

Conclusion Générale

Dans ce travail nous avons réussi à réaliser un système de télémétrie qui permet de faire une représentation 3D de l'espace. Ce système est constitué de plusieurs dispositifs et chaque dispositif aura un rôle important dans notre système.

Afin de réaliser un tel système nous avons divisé notre travail en 3 parties. Dans la première partie nous avons essayé de choisir les dispositifs qui vont nous servir à la réalisation du projet en tenant compte de l'efficacité, la précision et la disponibilité, bien que le prix. En deuxième partie nous passons à la partie programmation de la carte arduino en essayant de lier les dispositifs entre eux et le stockage des données afin de les utiliser pour la représentation 3D, qui sera notre but dans la troisième partie dont il s'agit de présenter les données acquises de l'interface série à l'aide d'un environnement de développement libre "Processing" ou on utilise le langage JAVA pour le programmer.

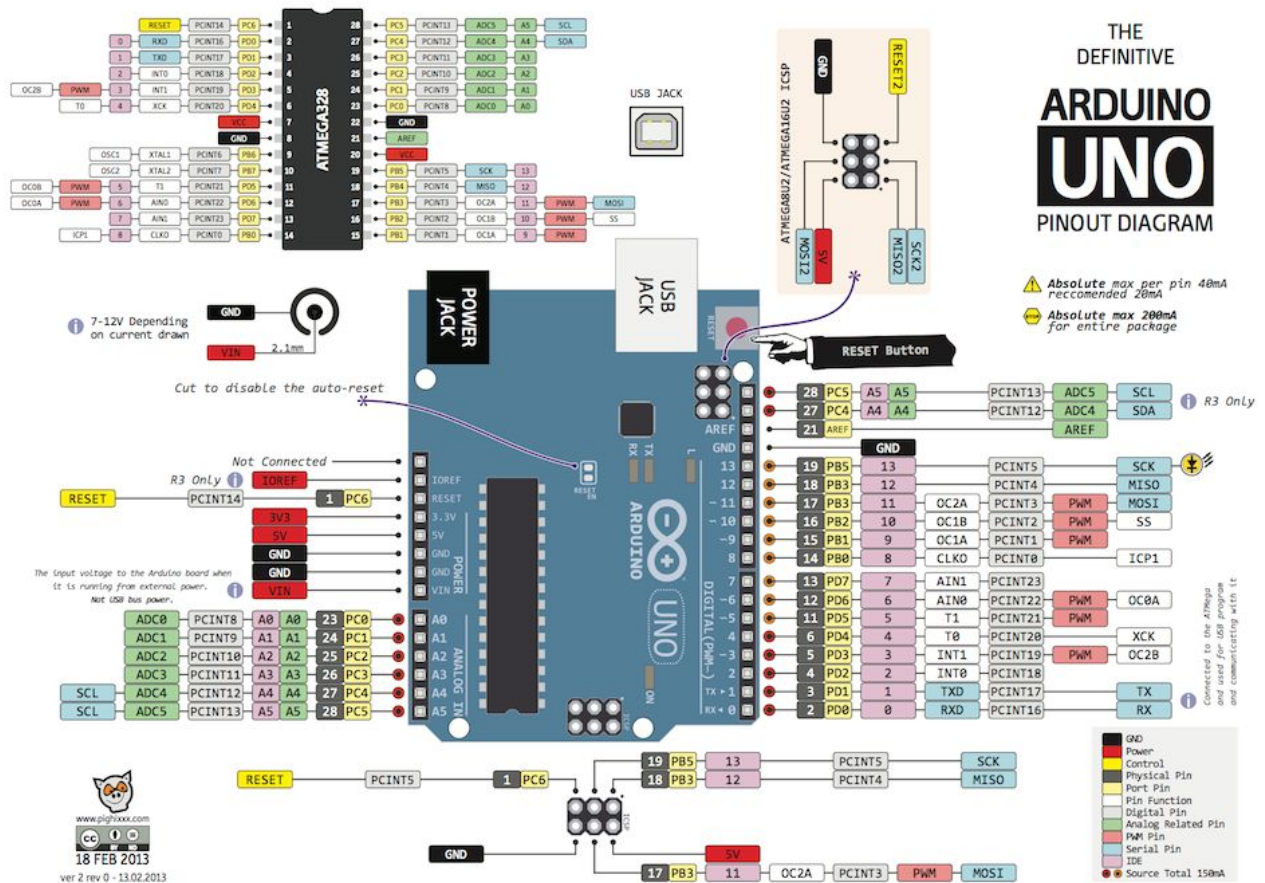
La conception de ce projet nous a permis d'acquérir des connaissances supplémentaires en conception et en développement des systèmes embarqués, en fait, nous avons pu accéder à la maîtrise du langage d'Arduino, également appris à travailler sur des cartes programmables ayant des capacités et des performances élevées, à savoir ARDUINO ainsi que la découverte du langage JAVA.

La réalisation de ce projet nous a énormément appris, autant au niveau de l'électronique, de la programmation des cartes électroniques. Nous avons aussi appris de nouvelles connaissances au niveau de la gestion du temps, des crises et d'équipes.

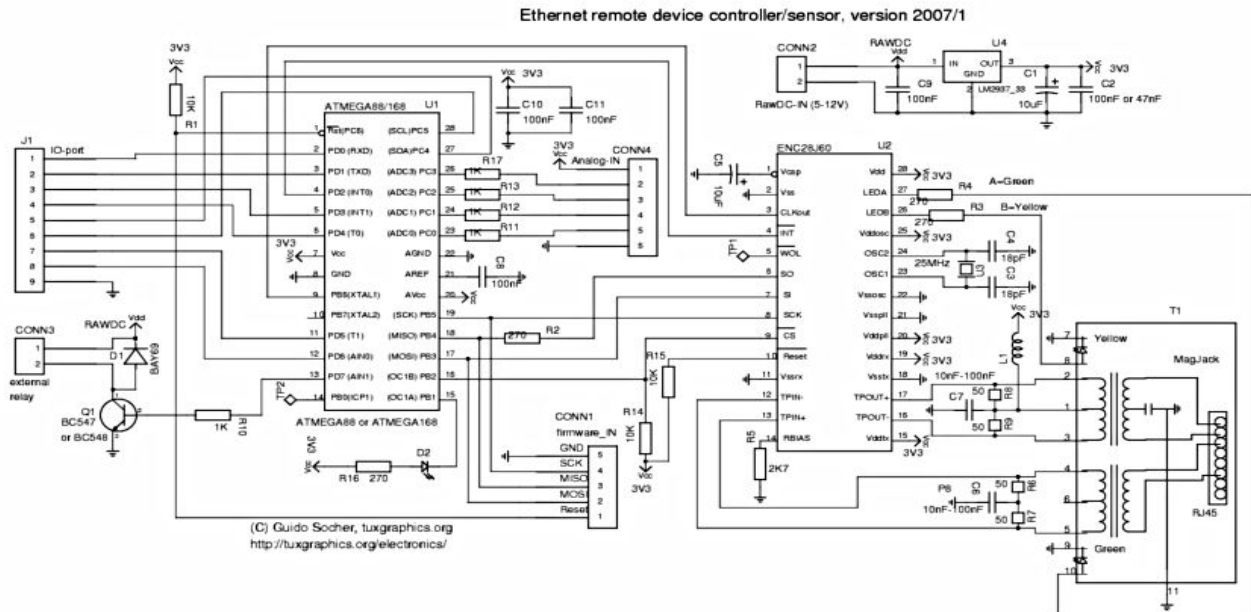
Ce travail reste, comme toute œuvre humaine, incomplet et perfectible, nous recommandons d'en améliorer la conception et pour cela nous proposons ci-dessous des améliorations pour les futurs développements:

- Améliorer la précision du système au niveau des mesure (utilisation des servos moteurs plus puissants pour diminuer les erreurs des mesures).
- Améliorer graphiquement l'interface de présentation.
- Améliorer le système de détection et de stockage des données.
- Ajouter plusieurs dispositifs pour une utilisation plus vague , polyvalente et multi-fonction dans plusieurs domaines.

Annexe



ARDUINO PINOUT



Structure du microcontrôleur AVR

Arduino code source

```
#include <Servo.h>
#include <Wire.h>
#include <LIDARLite.h>
```

```
//Délai entre chaque échantillon pour éviter les oscillations mécaniques
#define DELAY_BETWEEN_SAMPLES 100
//Taille des pas de YAW / PITCH en degrés (1 = résolution complète)
#define YAW_STEP 2
#define PITCH_STEP 2
#define MATH_PI 3.1415f
```

```
//Variables
LIDARLite myLidarLite;
Servo servoYaw,servoPitch;
char s[15];
int yawAngle,pitchAngle;
int x,y,z,r;
float theta,phi;
```

```
void setup() {
  // Initialiser la connexion série pour afficher les lectures de distance
  Serial.begin(115200);
```

```
  //Servo init
  servoYaw.attach(7);
  servoPitch.attach(9);
```

```
  //Lidar Lite v3 init
  myLidarLite.begin(0, true);
  myLidarLite.configure(0);
```

```

}

void loop() {
  delay(5000);
  //Balayage Yaw servomotor
  for (yawAngle = 0; yawAngle <= 180; yawAngle += YAW_STEP) {
    servoYaw.write(yawAngle);

    //Balayage Pitch servomotor. The direction depends on the current directory
    if(pitchAngle < 90){
      for (pitchAngle = 0; pitchAngle <= 180;pitchAngle+= PITCH_STEP){
        servoPitch.write(pitchAngle);
        sendMeasurement();
      }
    } else {
      for (pitchAngle = 180; pitchAngle >= 0;pitchAngle-= PITCH_STEP){
        servoPitch.write(pitchAngle);
        sendMeasurement();
      }
    }
  }
}

// Fonction pour acquérir, convertir et envoyer la mesure.
void sendMeasurement(){
  delay(DELAY_BETWEEN_SAMPLES);

  //Obtenir des coordonnées sphériques
  r = myLidarLite.distance(false);
  theta = (float)yawAngle * PI / 180.of;
  phi = (float)pitchAngle * PI / 180.of;

  // Convertissez et envoyez les
  sprintf(s,"%d %d %d\n\o",(int)(r*cos(phi)*cos(theta)),(int)(r*cos(phi)*sin(theta)),(int)(r*sin(phi)));
  Serial.print(s);
}

```

Processing code source

```

// LidarViewer.pde Processing sketch
// http://www.charleslabs.fr/en/project-3D+Lidar+Scanner

// Load sketch into Processing, available from:
// https://processing.org/
// You also need to install the PeasyCam library.

// This software read coordinates of the Arduino Lidar
// (format: "X Y Z") and displays it as a 3D point cloud that you
// can move around using the mouse.
// Press 's' to save the cloud as a text file in Processing directory.

import processing.serial.*;
import peasy.*;
import java.io.FileWriter;
import java.util.Calendar;
import java.text.SimpleDateFormat;
import static javax.swing.JOptionPane.*;

```



```

Serial serial;
PeasyCam cam;
final float angleIncrement=0.1f;

ArrayList<PVector> pointList;

final int SERIAL_SPEED = 115200;

void setup() {
  size(800, 640, P3D);
  colorMode(RGB, 255, 255, 255);
  pointList = new ArrayList<PVector>();

  // PeasyCam
  cam = new PeasyCam(this, 800);
  cam.rotateZ(-3.1415/4);
  cam.rotateX(-3.1415/4);

  // Serial Port (added dialog)
  try {
    if (Serial.list().length == 0) {
      println("No serial device connected");
      exit();
    }
    else if (Serial.list().length == 1) {
      // only one device, select it
      serial = new Serial(this, Serial.list()[0], SERIAL_SPEED);
    }
    else {
      // more than 1, show dialog
      StringBuffer ttyList = new StringBuffer();
      for (int j = 0; j < Serial.list().length; j++)
        ttyList.append("\n" + Serial.list()[j]);

      String selection = showInputDialog("Enter serial port to use:" + ttyList.toString());
      if (selection == null || selection.isEmpty())
        exit();

      serial = new Serial(this, selection, SERIAL_SPEED); // change baud rate to your liking
    }
  }
  catch (Exception e) {
    println("Not able to connect to serialPort (error:"+e.getClass().getName()+" " + e.getMessage() + ")");
    exit();
  }
}

void draw() {
  // Prepare window & drawing
  perspective();
  background(33);
  stroke(255,255,255);
  sphere(0.5f);
  fill(50);
  ellipse(0, 0, 10, 10);

  //Read Serial Port (if we can)
  String serialString = serial.readStringUntil('\n');

```

```

if (serialString != null) {
    String[] coordinates = split(serialString, ' ');
    if (coordinates.length == 3) {
        pointList.add(new PVector(float(coordinates[0]), float(coordinates[1]), float(coordinates[2])));
    }
}

// Draw the actual point cloud
for (int index = 0; index < pointList.size(); index++) {
    PVector v = pointList.get(index);
    if (index == pointList.size() - 1) {
        // Draw a line between Lidar and last point
        stroke(255, 15, 15);
        line(0, 0, 0, v.x, v.y, v.z);
    }
    // Draw point with a variable color
    stroke(255-v.z, 255-v.y, 255-v.x);
    point(v.x, v.y, v.z);
}
}

// Handle keyboard events : movement, save & clear all points.
void keyReleased() {
    if (key == 'x') {
        // erase all points
        pointList.clear();
    } else if (key == 's'){
        saveToFile();
    }
    else if (key == CODED) {
        if (keyCode == UP) {
            cam.rotateX(angleIncrement);
        } else if (keyCode == DOWN) {
            cam.rotateX(-angleIncrement);
        } else if (keyCode == LEFT) {
            cam.rotateY(angleIncrement);
        } else if (keyCode == RIGHT) {
            cam.rotateY(-angleIncrement);
        }
    }
}

// Function to save the point cloud in the Processing install directory
void saveToFile(){
    String fileName = "./points_" +
        new SimpleDateFormat("yyMMdd_HHmmss").format(Calendar.getInstance().getTime())+".xyz";
    PrintWriter pw = null;
    try{
        pw = new PrintWriter(new FileWriter(fileName,true));
        for(int i=0;i<pointList.size();i++)
            pw.println((int)pointList.get(i).x + " " +
                (int)pointList.get(i).y + " " +
                (int)pointList.get(i).z);
    }
    catch(Exception e){
    }
    finally {
        if(pw != null) pw.close();
    }
}

```

Références

https://eskimon.fr/extra/ebooks/arduino-premiers-pas-en-informatique-embarquee.pdf?fbclid=IwARo4ocMYFsXhBojMoNAoKzioYQ6_Q2GCeJc99u5DA_n6qZKWw6lK4JrI5X8

https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/742_decouverte-de-larduino/3417_le-materiel/

<https://www.teachmemicro.com/arduino-uno-pinout-diagram/>

<https://fr.wikipedia.org/wiki/Processing>

<https://www.instructables.com/id/Ultrasonic-3D-Maps-With-Python-and-Arduino/>

<https://web.archive.org/web/20140827195331/http://coolarduino.wordpress.com:80/2014/08/10/virtual-touch-screen-3d-ultrasonic-radar/>

<https://lidarscanner.wordpress.com/>

<https://github.com/andrefedalto/lidar-scanner>

<http://www.charleslabs.fr/en/project-3D+Lidar+Scanner>

<https://forum.arduino.cc/index.php?topic=457798.0>

<http://ericaeromodelisme974.unblog.fr/2011/06/04/le-servomoteur/>

https://eskimon.fr/extra/ebooks/arduino-premiers-pas-en-informatique-embarquee.pdf?fbclid=IwARo4ocMYFsXhBojMoNAoKzioYQ6_Q2GCeJc99u5DA_n6qZKWw6lK4JrI5X8

Cours Conception Systèmes Embarqués du Pr Ridha GHAYOULA de l'INSE

http://perso.citi.insa-lyon.fr/afraboul/rts6/slides/cours_RTS6_1X2.pdf

Cours Systèmes Embarqués INSA Lyon

http://infocenter.arm.com/help/topic/com.arm.doc.ddio301h/DDIo301H_arm1176jzfs_rop7_trm.pdf

<https://tel.archives-ouvertes.fr/tel-00163980/document> Thèse Doctorat INP Grenoble

http://infocenter.arm.com/help/topic/com.arm.doc.ddio301h/DDIo301H_arm1176jzfs_rop7_trm.pdf

<http://dspace.univ-tlemcen.dz/bitstream/112/10914/1/Ms.Tel.Djedoui%2BBoukern.PDF>