République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de L'**E**nseignement **S**upérieur et de la **R**echerche Scientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DE **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire

de fin d'études

En vue de l'obtention du diplôme de MASTER ACADEMIQUE en Electronique
Option : Télécommunications et Réseaux

Thème

Conception et réalisation d'un système de mesure de flambement et de déformation à base d'une Arduino Uno

Proposé et dirigé par :
Mr. K.BENNAMANE

Présenté par :
Mr. KROUBI Tarik

Année universitaire : **2013/2014**

Remerciement

Je remercie ALLAH le tout puissant de m'avoir donné le courage et la volonté de mener à terme ce présent travail.

Je remercie énormément et infiniment Messieurs K.BENNAMANE et M.LAGHROUCHE d'avoir accepté de m'encadrer et je leurs suis très reconnaissant pour leur précieuse aide pendant les moments difficiles.

Je tiens aussi à remercier Mr H.Achour, Mr Kanane, Mr Berchiche pour leurs conseils et leur dévouement afin de mener à bien ce mémoire.

Je tiens à remercier les membres du jury qui ont accepté de juger ce modeste travail.

Dédicace

Je remercie Dieu Le tout puissant de m'avoir donné la force de réaliser ce projet.

Je voudrais dédier le présent travail à mes chers parents qui m'ont élevé et soutenu tout au long de ma vie, et tout spécialement à mon cher frère Rabah qui m'a encouragé et soutenu.

Je dédie également ce projet à mes chers frères et sœurs et leurs enfants ainsi que Nassim et ma grand-mère.

Je tiens énormément à remercier mes chers amis et camarades de ma promotion et surtout ceux avec lesquels j'ai travaillé au Laboratoire projet : Yazid, Sofiane, Hamza, MouMouh, Mustapha, Salim.

Je voudrais aussi dédier cette thèse à la responsable du laboratoire projet Mme Slimani.

Enfin, je voudrais dédier ce travail à toute personne ayant participé de près ou de loin à sa réalisation.

Table des matières

Chapitre III : Réalisation du système de mesure.

Introduction générale

Introduction :

La métrologie au sens étymologique du terme se traduit par « la science de la mesure », elle s'intéresse traditionnellement à la détermination de caractéristiques (appelées grandeurs) qui peuvent être fondamentales (longueur, masse, temps...) ou dérivées (surface, vitesse…). Dans nos jours, Assurer un positionnement micrométrique sur une grande course reste une difficulté scientifique. L'objectifs des métrologues demeure d'avoir un système de mesure suffisamment fiable et précis, indépendant du temps et du lieu.

Cependant, dans les domaines courants des essais, La détermination des lois de comportement mécanique des matériaux pose le problème de la mesure des champs de flambements ou de déformations. Par l'obtention d'une information de champ, sans contact et avec une résolution spatiale élevée, les méthodes optiques s'imposent dans la mesure des contraintes. En effet La lumière possède des atouts remarquables qui lui assurent un rôle privilégié en métrologie : son absence d'inertie autorise des mesures rapides et l'absence de contact matériel avec l'objet étudié supprime toute déformation ou usure.

Toutefois l'électronique moderne et la technologie informatique ont apporté à la métrologie un tout nouveau champ d'application.

Notre objectif se résume par : concevoir un système de mesure pour une machine d'essai et de tests, qui répond à des besoins bien spécifiques et dictés par le cahier des charges.

Ce système de mesure est basé sur l'acquisition des données issues des différents capteurs installés sur la machine à travers une chaine d'acquisition à base de la carte Arduino Uno, permettant le calcul de la position d'un vérin hydraulique, le flambement et la déformation des poutres en béton. Puis la transmission des résultats de la mesure via le port USB de l'Arduino vers un logiciel de traitement et d'affichage.

Ce mémoire s’organise en trois chapitres, à travers lesquels nous décrivons le travail effectué pour la conception et la réalisation de notre système :

Dans le premier chapitre nous présentons les spécifications du cahier des charges sur lequel nous nous sommes basés dans la réalisation du projet.

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

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

Nous conclurons notre travail par une conclusion générale et perspective.

Une classification des capteurs qui a contribuée dans le choix des capteurs utilisés dans ce mémoire est rajoutée en annexe A.

Chapitre I

Le Cahier Des Charges

I.1 Introduction :

La conception et la réalisation d'un système de mesure de flambement et de déformation font l'objet de ce mémoire. L'objectif principal de ce système est de récupérer les résultats de mesure de flambement des poutres, et de la déformation des éprouvettes de béton et la position exacte du vérin hydraulique (la presse), issues des différents capteurs installés sur la machine.

La machine fait partie du matériel de laboratoire de recherche du département de génie-civil à l'Université Mouloud Mammeri Tizi-Ouzou. Son principal rôle est d'effectuer une compression sur les tuyaux de canalisation, et de calculer leurs résistances. Du fait de l'application, qui se résume que sur l'étude de la compression sur les tuyaux de canalisations en fonction de temps, les responsables du laboratoire ont pressenti le besoin d'équiper la machine d'un nouveau système capable de mesurer le flambement des poutres et le déplacement ou la déformation des éprouvettes en béton en temps réel, qui contribuera à étendre l'utilisation de la machine sur différents sujets d'étude.

I.2 Présentation de la machine :

La machine est de référence P460/70 de chez TECNOTEST, du constructeur italien. Elle sert pour les essais et les tests, elle est composée de deux parties essentielles :

- Structure mécanique d'application de la charge *figure 1*.
- Centrale Hydraulique et de Commande *figure 2.*

Figure 1 : La Machine P460/70.

Figure 2 : Centrale Hydraulique et de Commande.

I.2.1 Structure mécanique d'application de la charge :

Pour accomplir les essais de compression sur les tuyaux en béton d'un diamètre compris entre 450 et 3000 mm max, la machine de dimensions : 2620 x 2500 x 5060 mm et d'un poids : 3800 kg, doit être associée à :

- Un système hydraulique de charge de 700 kN et une cellule de charge -*figure 3 (1)*-
- Un treuil électrique à 2 vitesses pour la traverse intermédiaire -*figure 3 (2)*- .
- 2 couteaux inférieurs d'appui d'une longueur de 2500 mm -*figure 3 (3)*- et un couteau supérieur, oscillant de 2500 mm de longueur –*figure 3 (4)*- pour l'essai de charge de compression.

- Une structure complètement zinguée à chaud.
- Une embase de la structure à haute rigidité.

.

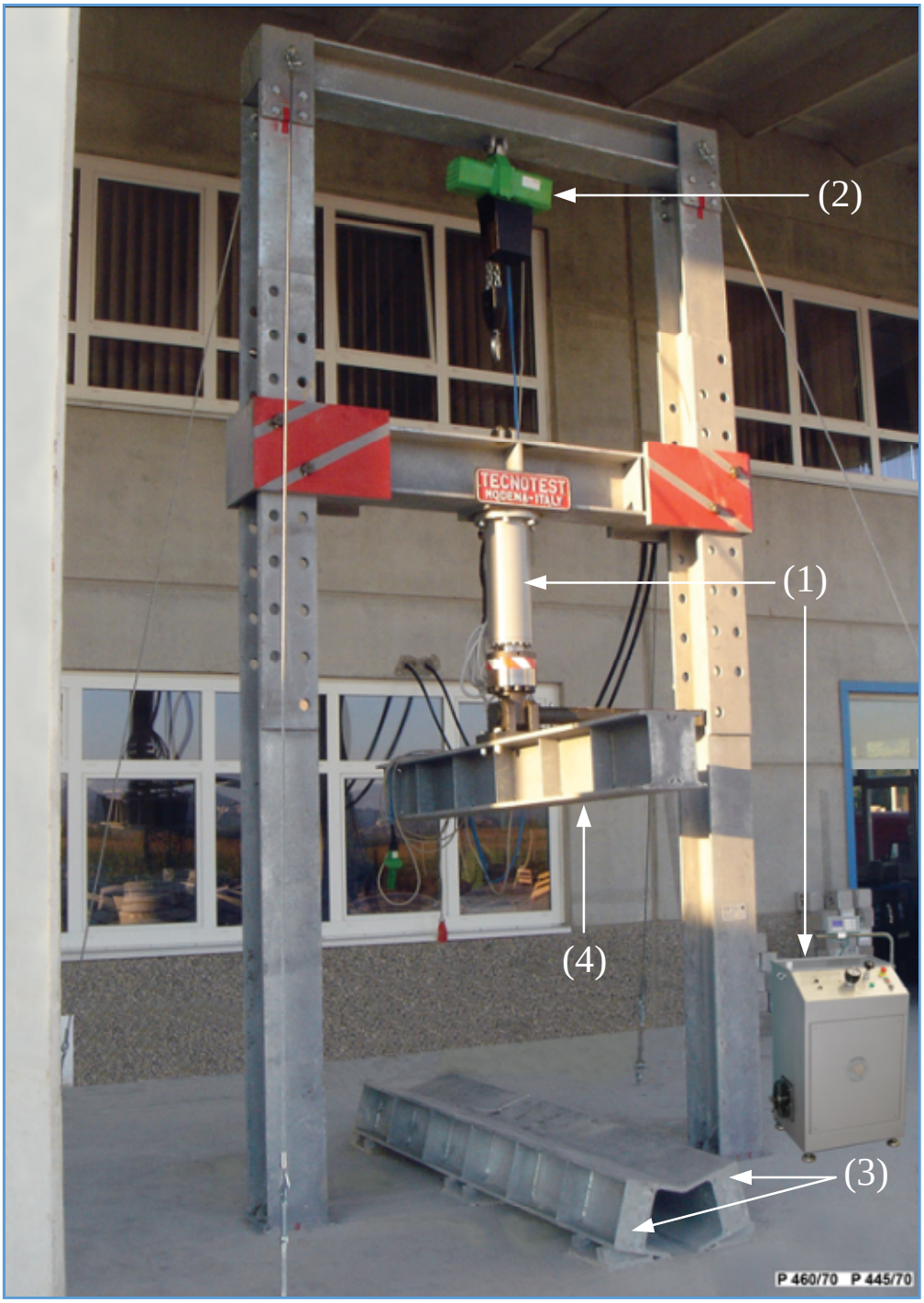


Figure 3 : Diffèrent élément de la structure mécanique.

I.2.2 Centrale hydraulique et de commande :

Le système complet comprend :

- La centrale hydraulique équipée de meuble –*figure 2*-.
- réglage de la cadence de mise en charge dotée de deux pompes : une à basse pression et à haut débit, l'autre à haute pression et à bas débit. La première détermine l'approche rapide, la seconde est utilisée pour l'essai. L'ensemble est complété par deux dispositifs spéciaux du projet Tecnotest : le Régulateur de Débit, avec lequel on détermine la cadence de mise en charge et le robinet pour décharger la pression et faire renter le vérin.

Figure 4 : Système de charge P460/70 et la cellule de charge.

- Système de lecture digitale et d'affichage Eurotronic -*figure 5*- avec cadence-mètre électronique, à pleine échelle 700.0 kN (sensibilité 10 N). Ecran avec l'indication de la cadence de mise en charge.
- Capteur de la charge : cellule de charge électronique.
- Cylindre hydraulique à double effet :
  - ✓ Capacité 700 kN (300 bar)
  - ✓ Course 400 mm
  - ✓ Fixation supérieure au disque (ø 310 mm) pour la traverse de la structure.
  - ✓ Poids : 200 kg
- 2 tuyaux flexibles de haute pression, d'une longueur de 10 mètres pour la connexion du cylindre à la centrale oléo-dynamique
- Dimensions de l'armoire : 700 x 530 x 1050 mm.
- Poids : 150 kg

Figure 5 : Système de lecture digitale et d'affichage EUROTRONIC

I.3 Fonctionnement du système à réaliser :

I.3.1 La mesure de flambement à l'aide d'une caméra :

- La camera doit être fixe et immobile, pour éviter les bruits dans les images capturés.
- L'algorithme qui retourne le résultat de la mesure, doit capturer l'arrière-plan (background) le mémoriser et le différentier des images de premier plan (foreground), calculer le flambement et afficher le résultat sur l'écran en temps réel.
- Afficher les résultats de mesure sous forme d'un graphe et sous forme d'une table utilisable par d'autre logiciel comme Excel.

I.3.2 La mesure de la déformation à l'aide d'un capteur optique :

- Le capteur en question, est une fourche optique. Installé sur le sujet de teste avec des anneaux de fixation, il doit retourner le résultat de mesure de la déformation avec une précision de 0.01 mm.
- La possibilité d'une remise à zéro à n'importe laquelle position.

I.3.3 La mesure de la position du vérin :

- Le capteur de position en question, est un capteur ultrason. Monté et fixé sur le vérin, il doit être capable de mesurer la distance le séparant du sujet de teste, et l'afficher sur l'écran d'un ordinateur.
- La possibilité de prendre n'importe quelle position du vérin comme le début de la mesure « un zéro flottant ».
- Deux capteur de fin de course pour initialiser les deux capteurs ultrason et optique.

I.3.4 Unité d'acquisition et interface graphique :

- Centrale d'acquisition à base d'un microcontrôleur.
- Création d'une interface graphique et logiciel d'acquisition de données des différents capteurs installés sur la machine.
- Visualisation des données en temps réel.

I.3.5 Du point de vue technique, le système doit être :

- Facile à installer : le logiciel comme le matériel.
- Partie électronique compacte et peu encombrante, en utilisant un microcontrôleur qui assurera la gestion des capteurs.
- Fiable, avec une probabilité de fausse mesure infiniment petite.
- D'un coût relativement abordable par rapport aux systèmes disponibles proposés par la marque dans ce domaine.

I.4 Conclusion :

Pour répondre à ce cahier des charges, le système de mesure de flambement et de déformation sera divisé principalement en deux modules : le module d'acquisition de donnée et de mesure, le module de traitement et d'affichage.

Avant de montrer les détails de notre réalisation, nous présentons dans le chapitre suivant le schéma de conception du système à réaliser.

Chapitre II

Conception du système de mesure

II.1 Introduction :

Après que nous avons donné la description de la machine et du système à réaliser, et rédigé le cahier des charges, il y a lieu d'étudier la structure de système pour pouvoir traiter, visualiser, ou sauvegarder les différentes données qui y transitent. Dans ce chapitre nous allons présenter la conception du système de mesure et les critères de choix de chaque composant.

II.2 Structure du système :

Le système de mesure que nous avons conçu est constitué de deux modules essentiels chacun accomplit une ou plusieurs fonctions qui lui sont propres *figure 6.* Le premier est le module d'acquisition de données et de mesure, le second est le module de traitement et d'affichage. On commence par le premier module qui est réalisé à base d'une carte Arduino Uno, du moment que c'est lui qui s'occupe en premier lieu de l'acquisition de données des différents capteurs, puis on passe au second module.

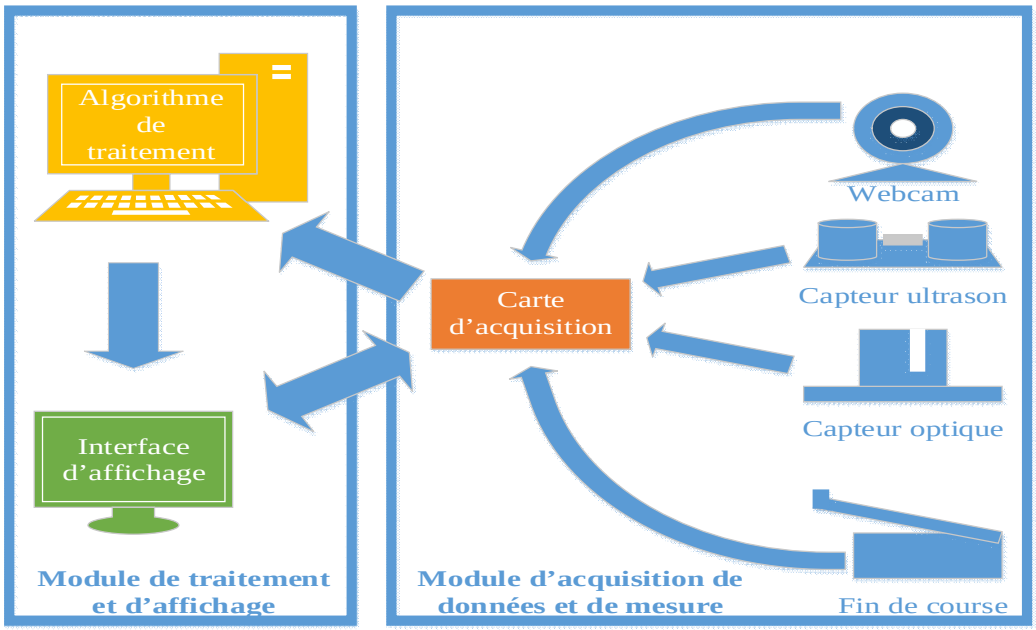


Figure 6 : Schéma bloc de système de mesure

- **Le module d'acquisition de données et de mesure :** c'est la partie *hardware* du système à base de la carte Arduino Uno, qui prend en charge de recevoir et d'envoyer les différentes informations entre les deux modules : c'est lui qui gère et maintien le fonctionnement du circuit et tous les capteurs y sont connectés.
- **Le module de traitement et d'affichage :** c'est la partie *software* du système, qui assure la capture et le traitement des images issues de la webcam, le tous englobé par une interface logiciel écrite en C++ et qui a le rôle de communiquer avec et entre les deux modules, d'afficher les différentes valeurs des capteurs sous formes différentes, afficher la capture d'image de la webcam et enfin sauvegarder les données acquises sous forme d'un fichier.

II.3 Module d'acquisition de données et de mesure :

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

- Une carte Arduino Uno R3.
- Une webcam HD 1.3Mp.
- Capteur optique de type fourche optique TOR.
- Module ultrasonique pour carte arduino HC – SR04.
- Capteurs de Fin de course.

II.3.1 Présentation de la carte Arduino Uno R3 :

II.3.1.1 Description :

Le modèle UNO *figure 7* de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATMega328. Le microcontrôleur ATMega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C. Cette carte possède 14 entrées/sorties numériques (dont 6 peuvent être utilisées comme étant des sorties PWM (*Pulse Width Modulation*), 6 entrées analogiques avec un convertisseur Analogique/Numérique de 10 bits de résolution, 1 résonateur céramique (quartz) de 16 Mhz, 1 connecteur ICSP (*In-Circuit Serial Programing)* qui permet la programmation du microcontrôleur sur le circuit sans avoir à l'enlever, 1 connecteur jack pour une alimentation extérieur, un bouton de reset pour mettre le processus à zéro.

L'avantage de cette carte c'est qu'elle n'a pas besoin de pilote pour faire la conversion FTDI USB/Série, elle a juste un petit microcontrôleur ATMega 16U2 (pour la version 3) programmé comme convertisseur USB/Série.

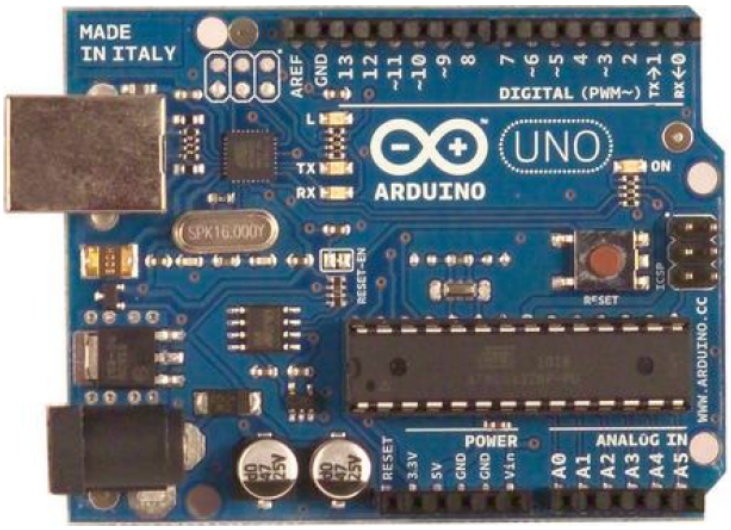


Figure 7 : Description de la carte Arduino Uno R3.

L’intérêt principal des cartes Arduino est leur facilité de mise en œuvre. Arduino fournit un environnement de développement s'appuyant sur des outils open source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties courantes : gestion des E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus I2C, exploitation de servomoteurs ...etc.

II.3.1.2 Alimentation de carte arduino :

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 (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus).

L’adaptateur secteur peut être connecté en branchant une prise de 2.1mm, dédié du pole 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 20V. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Ainsi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.

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

Les broches d'alimentation sont les suivantes :

- **VIN :** (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, on peut accéder à la tension d'alimentation sur cette broche.
- **5V :** la tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (un régulateur de tension est intégré dans la carte). La 5V fournie par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit aussi 5V régulé) ou de toute autre source d'alimentation régulée.
- **3V3 :** une alimentation de 3.3V fournie par le régulateur de 3.3V de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu de 5V). L'intensité maximale disponible sur cette broche est de 50mA.

II.3.1.3 Protection du port USB contre la surcharge en intensité :

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

II.3.1.4 Gestion des mémoires dans la carte Arduino :

L'ATMega 328 a 32ko de mémoire FLASH pour stocker le programme (dont 0.5k0 également utilisé par le bootloader). L'ATMega 328 a également 2ko

de mémoire SRAM (volatile) et 1ko d'EEPROM (non volatile : mémoire qui peut être lue à l'aide de la librairie EEPROM).

II.3.1.5 les entrée/sorties numériques :

Chacune des 14 broches numériques de la carte Uno (numérotées de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode(), digitalWrite(), et digitalRead() du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne « résistance de rappel » (*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**).

Il y a entre ces broches celles qui ont des fonctionnalités en plus :

- **Communication série :** broche 0 (RX) et 1 (TX). Utilisée pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATMega16U2 programmé en convertisseur USB/Série de la carte (composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur). On fait appel à la transmission série à travers ces broches avec l'instruction Serial.print(), à condition que le câble USB soit déconnecté, sinon il va y avoir un chevauchement.
- **Interruptions externes :** broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction attachInterrupt() pour plus de détails.
- **Impulsion PWM (largeur d'impulsion modulée) :** broches 3, 5, 6, 9, 10 et 11. Fournissent une impulsion PWM 8 bits à l'aide de l'instruction analogWrite().

- **SPI (Interface Série Périphérique) :** broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI disponible avec une librairie pour la communication SPI. Les broches SPI sont également connectées sur le connecteur ISCP.
- **I2C :** broche 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C, disponible en utilisant la librairie Wire/I2C.
- **LED :** broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsqu'elle est au niveau BAS, la LED est éteinte.

II.3.1.6 Les entrées analogiques :

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

NB : les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19, aux cas où le nombre de broches numériques n'est suffisant.

- **Autres broches :** il y a deux autres broches disponibles sur la carte :
  - **AREF :** tension de référence pour les entrées analogiques (si différent du 5V), utilisée avec l'instruction analogReference(). Elle utilisée pour comparer la valeur d'une tension d'entrée par rapport à la valeur d'une tension de référence choisie.
  - **Reset :** mettre cette broche au niveau BAS entraine la réinitialisation du microcontrôleur. Cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

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

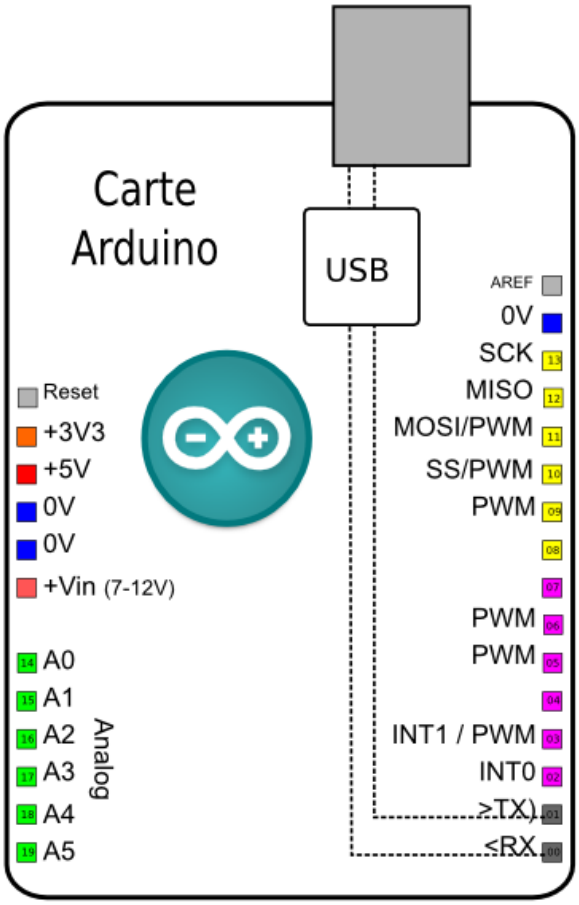


Figure 8 : Brochage de la carte Arduino Uno.

II.3.1.7 Arduino et la communication avec l'extérieur :

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

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

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

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

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

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

- **Dimension de la carte :**

Les longueurs et largeurs maximales de la Uno sont respectivement 6.86 cm et 5.33 cm, avec le connecteur USB et le connecteur d'alimentation jack, elle s'étend au-delà des dimensions de la carte. Quatre trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier (pour l'embarquer sur un système). Noter que la distance entre les broches 7 et 8 est de 0.16 pouces, et 0.1 pouces séparant les autres broches.

II.3.1.8 Le microcontrôleur ATMega328 :

Le microcontrôleur utilisé sur la carte Arduino Uno est un microcontrôleur ATMega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits.

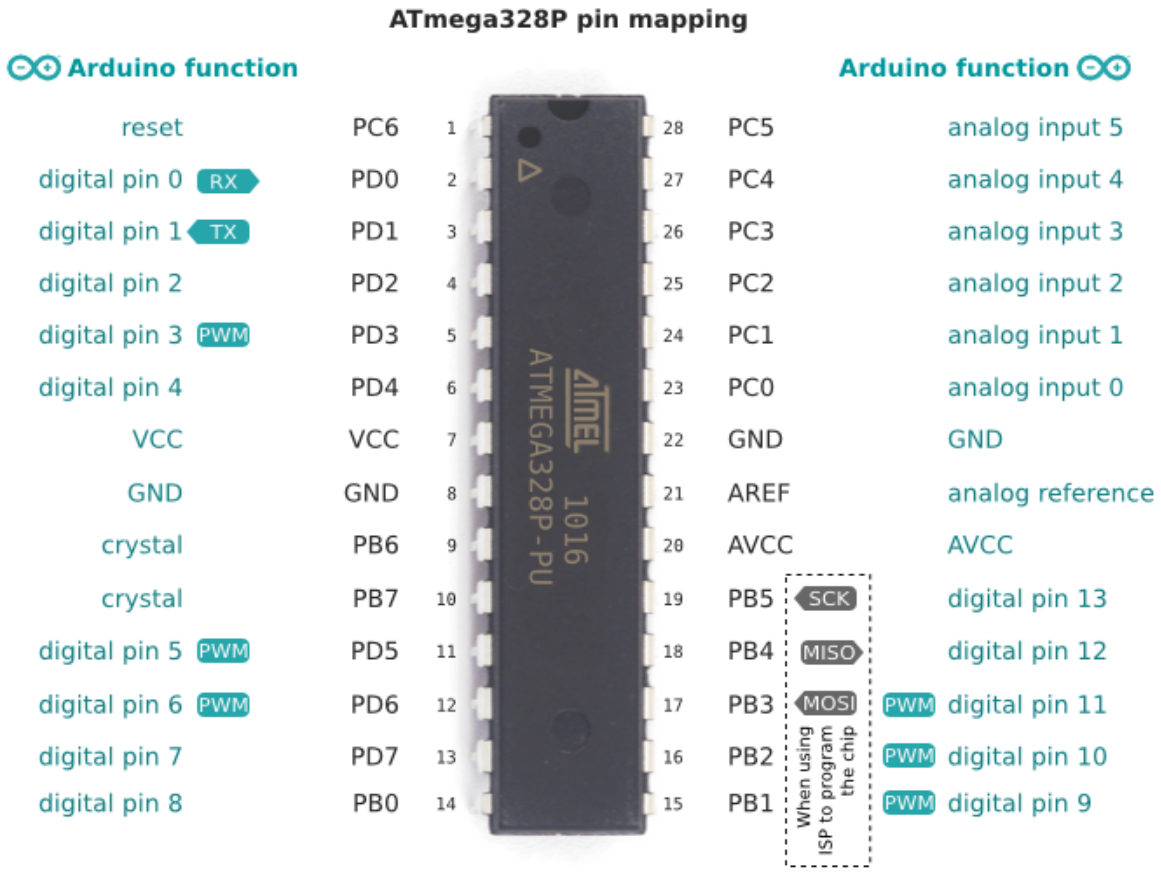


Figure 9 : Brochage des pattes d'Atmega328.

- **Les principales caractéristiques sont :**
  - **FLASH =** mémoire programme de 32Ko
  - **SRAM** = données (volatiles) 2Ko
  - **EEPROM =** données (non volatiles) 1Ko
  - **Digital I/O (entrées-sorties Tout Ou Rien) =** 3 ports PortB, PortC, PortD (soit 23 broches en tout I/O).
  - **Tension d'alimentation interne** = 5V.

- **tension d'alimentation (recommandée) =** 7 à 12V, limites = 6 à 20 V.
- **Courant max par broches E/S =** 40 mA.
- **Courant max sur sortie 3,3V** = 50mA.
- **Fréquence horloge** = 16 MHz.
- **Dimensions** = 68.6mm x 53.3mm.
- **Timers/Counters : Timer0 et Timer2** (comptage 8 bits), **Timer1** (comptage 16bits). Chaque timer peut être utilisé pour générer deux signaux **PWM**. (6 broches OCxA/OCxB)
- **Plusieurs broches multifonctions :** certaines broches peuvent avoir plusieurs fonctions différentes choisies par l'utilisateur.
- **PWM =** 6 broche OC0A(PD6), OC0B(PD5), 0C1A(PB1), OC1B(PB3), OC2A(PB3), OC2B(PD3)
- **Analog to Digital Converter (résolution 10bits) =** 6 entrées multiplexées ADC0(PC0) à ADC5(PC5)
- **Gestion bus I2C (TWI Two Wire Interface) =** le bus est exploité via les broches SDA (PC5)/SCL (PC4).
- **Port série (USART) =** émission/réception série via les broches TXD(PD1)/RXD(PD0).
- **Comparateur Analogique =** broches AIN0(PD6) et AIN1 (PD7) peuvent déclencher interruption.
- **Watchdog Timer programmable :** l'ATMega possède un compteur dit de chien de garde programmable pour générer des interruptions à la fin de son comptage et il peut être utilisé comme étant un simple compteur.
- **Gestion d'interruptions (24 sources possibles) :**
  - ✓ Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3).

- ✓ Interruptions sur changement d'état des broches PCINT0 à PCINT23.
- ✓ Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables).
- ✓ Interruption liée au comparateur analogique.
- ✓ Interruption de fin de conversion ADC.
- ✓ Interruptions du port série USART.
- ✓ Interruption du bus I2C.

- **Pourquoi Arduino :** Un microcontrôleur d'une autre famille comme le PIC 16f887, un puissant microcontrôleur de la famille Microchip, aurai pu être choisi pour le projet, mais l'Arduino Uno a été choisi pour les avantages suivants :
  - L'Atmega328 répond au cahier des charges. En effet le nombre des entrées/sorties dont on aura besoin est égale a six entrées/sorties sans les VCC et GND.
  - L'utilisation de deux interruptions externes pour la fourche optique. L'arduino Uno en possède deux (pin digitale 2 et 3).
  - S'épargner le travail de conception du circuit de configuration du microcontrôleur, de circuit de conversion Port Série → USB.
  - Microcontrôleur préprogrammé avec un bootloader (ISP), donc le programmateur dédié n'est pas nécessaire.
  - Une communauté active avec un site officiel et un forum officiel.
  - Environnement de programmation claire, simple et Multi-plateforme.

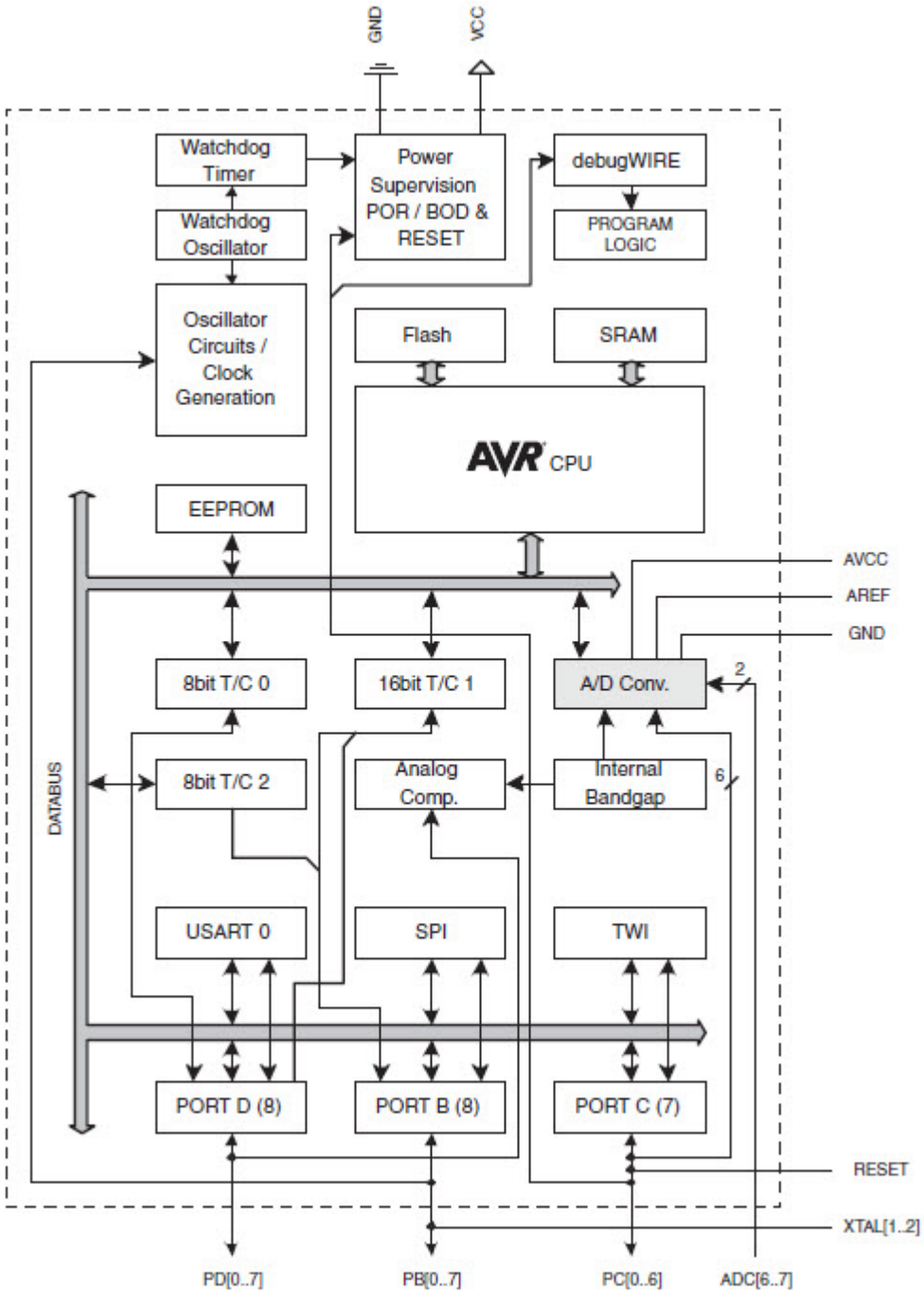


Figure 10 : Architecture interne de l'ATMega328.

II.3.1.9 Le capteur de vision :

- **Camera IP Wi-Fi :**

Une caméra IP ou caméra réseau est une caméra de surveillance utilisant le Protocole Internet pour transmettre des images et des signaux de commande via une liaison sans fil Wi-Fi (ou Fast Ethernet dans le cas d'une caméra IP filaire). Certaines caméras IP sont reliées à un enregistreur vidéo numérique (DVR) ou un enregistreur vidéo en réseau (NVR) pour former un système de surveillance vidéo *figure 11*.

- L'avantage des caméras IP est qu'elles permettent aux propriétaires et aux entreprises de consulter leurs caméras depuis n'importe quelle connexion internet via un ordinateur portable ou un téléphone 3G.
- Prix relativement élevé par rapport aux caméras simples et webcams.

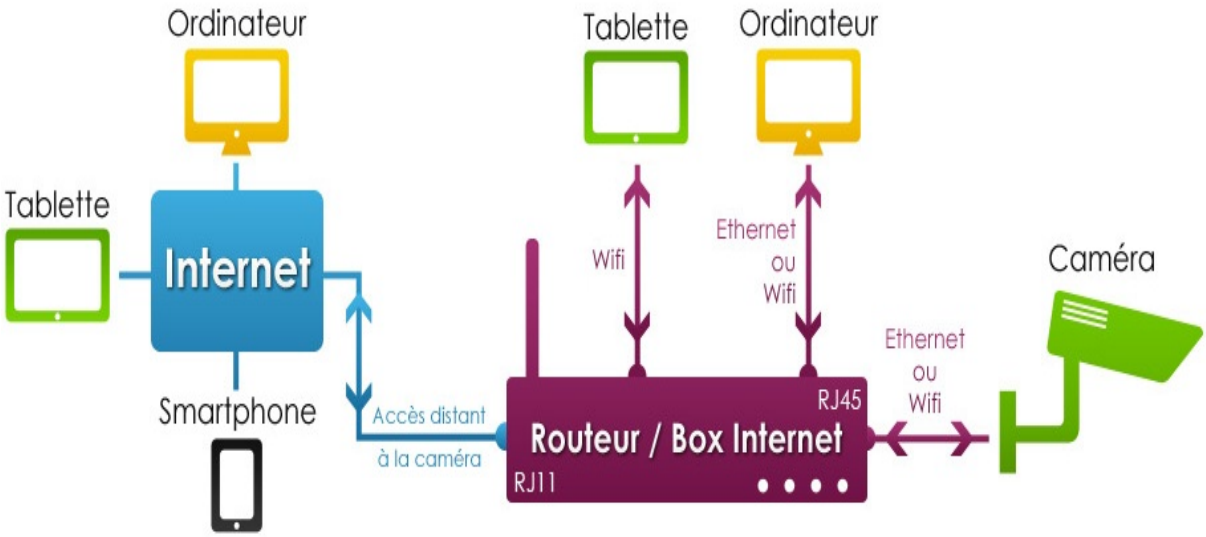


Figure 11 : Schéma de raccordement d'un camera IP.

- **Webcam :**

Une webcam, est une caméra conçue pour être utilisée comme un périphérique d'ordinateur, et qui produit une vidéo dont la finalité n'est pas d'atteindre une haute qualité, mais de pouvoir être transmise en direct à travers un réseau local, ou Internet. L'explosion des débits de connexions ont vu naître les Webcams HD de haute qualité et d'un rapport qualité/prix raisonnable.

Il est vrai que l'utilisation d'une caméra IP Wi-Fi, aurai pu être intéressant et apporter une certaine souplesse à la réalisation du projet. Mais un prix élevé, une configuration fastidieuse en réseau *figure 11* et une programmation supplémentaire pour la capture des images qui transitent en réseau local, nous ont incités à opter pour un Webcam HD 1.3MP.

II.3.1.10 Le capteur de déformation :

Le capteur de déplacement inductif linéaire de type LVDT (*Linear Variable Differential Transformer)*, est le capteur idéal pour la mesure de déformation du fait de :

- son excellente fiabilité.
- sa longue durée de vie.
- sa robustesse dans les milieux les plus difficiles.
- sa résolution infinie qui dépend que de l'électronique de traitement (conditionneur).

Mais son prix très cher (qui dépasse le 100000DA), nous a forcés à le remplacer par un autre capteur d'une grande résolution.

La solution : une fourche optique avec un film gradué *figure 12*. Utilisée principalement dans l'industrie des imprimantes pour les avantages suivant :

- Très bonne résolution jusqu'à 0.01mm.
- Grande précision.
- Longue durée de vie due à l'absence des frottements.
- Une gamme de mesure dynamique qui dépend de la longueur de film gradué.

La *figure 13* nous renseigne sur la largeur des graduations en les comparants à un point d'un simple stylo à bille.

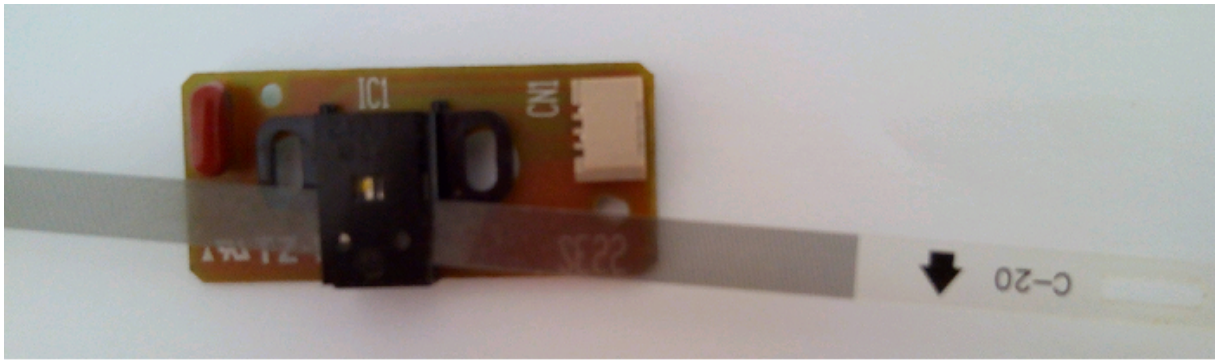


Figure 12 : Fourche optique avec le film gradué.

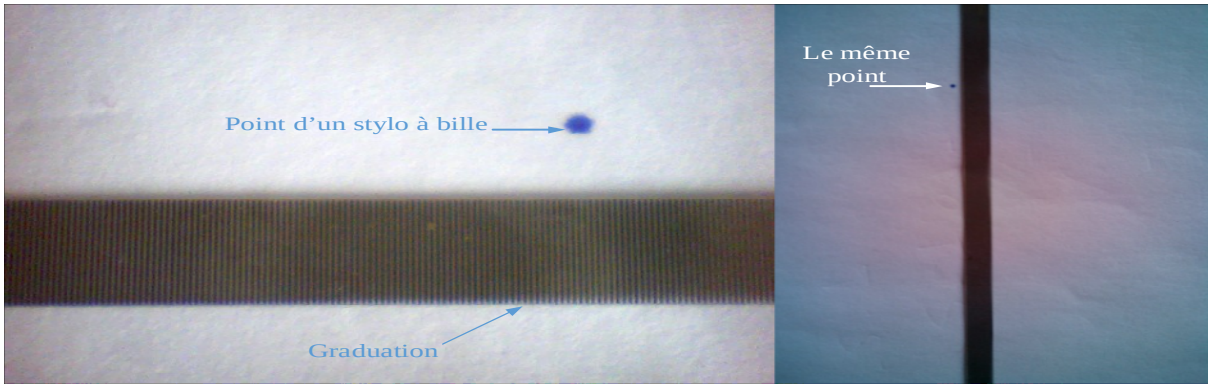


Figure 13 : Zoom sur le film gradué.

II.3.1.11 Le capteur ultrasonique :

Le choix du capteur ultrasonique est un module pour arduino HC-SR04 qui est composé de deux capteurs ultrason : l'un pour l'émission des ondes ultrasons, l'autre pour la réception de ces mêmes ondes réfléchies par un obstacle, ses principales caractéristiques sont :

- Alimentation = 5v CC.
- Consommation en utilisation = 15 mA.
- Gamme de distance = 2 cm à 4m.
- Fréquence d'utilisation = 40 Hz.
- Angle max = 15°.
- Résolution de 0,3 cm (référence datasheet, qui, avec un très bon conditionneur, pourrait augmenter la résolution).

- **Principe de fonctionnement :**

On envoie via une broche de sortie d'Arduino une impulsion de niveau haut (d'amplitude +5V) pendant au moins 10 µs sur la broche *Trig* du module HC-SR04, le retour de l'impulsion par l'écho, réfléchit par un obstacle, est transformer en une impulsion de même amplitude que celle envoyée dont la durée est proportionnelle à la distance entre le module et l'obstacle. Cette impulsion est reçue par une broche d'Arduino configuré en entrée. La *figure 14* illustre le principe.

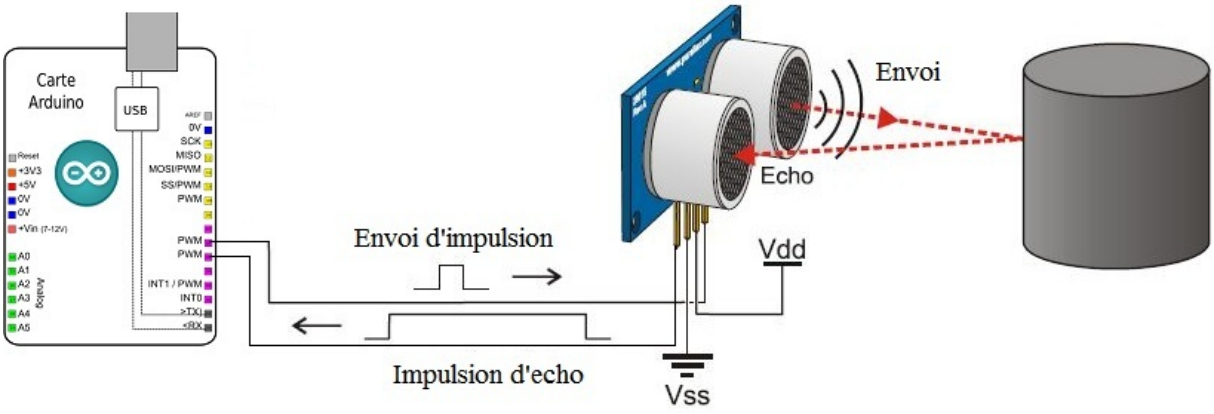


Figure 14 : Schéma de principe de fonctionnement du HC-SR04.

La distance est calculée, en utilisant la formule suivante :

$$\text{Distance (cm)} = \frac{\text{durée de l'impulsion } (en\ \mu s)}{58}$$

II.3.1.12 Capteur de fin de course :

Capteurs de fin de fin de course analogique standard.

Figure 15 : Capteur de fin de course.

II.4 Module de traitement et d'affichage :

II.4.1 Critère de choix des composants :

Pour la réalisation de ce projet, nous avons opté pour les composants suivants :

- Langage de programmation C/C++.
- Environnement de programmation Visual Studio 2012.
- Librairie supplémentaire pour Visual Studio nommé OpenCV.
- Webcam HD 1.3MP pour la capture d'images.

II.4.1.1 Choix de langage :

- **Matlab :** C'est un langage de haut niveau et un environnement interactif pour le calcul numérique, la visualisation et la programmation. Permet de manipuler des matrices, analyser des données, développer des algorithmes et créer des modèles et des applications.

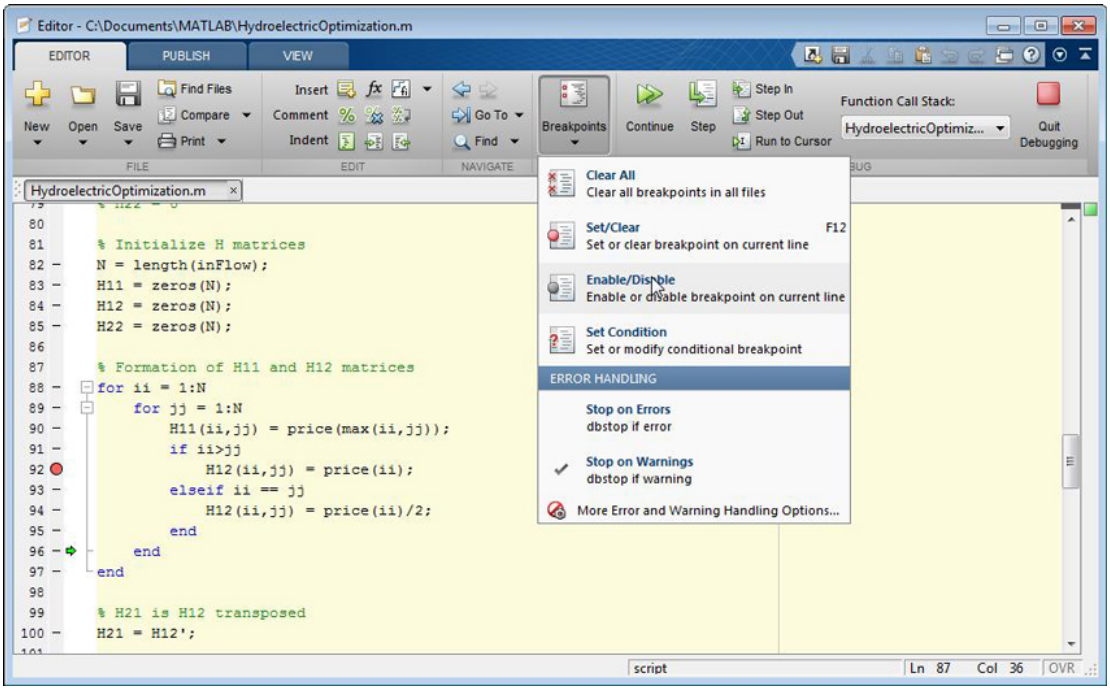


Figure 16 : Environnement de développent de Matlab.

- **Avantages :**
  - ✓ Programmation très rapide pour le calcul et pour l'affichage.

- ✓ Une librairie très riche (toolbox).
- ✓ Possibilité d'inclure un programme en C/C++.
- ✓ Langage interprété : Pas de compilation donc pas d'attente pour compiler.
- ✓ Scripte pouvant être injecté dans un microcontrôleur.
- ✓ Documentation débordant (plus de 1 million d'utilisateur en 2004 tout domaine confondus).
- ✓ Possibilité d'exécuter du code en dehors du programme.
- ✓ Code facile à comprendre et très lisible.
- ✓ Une aide très bien faite.

- **Inconvénients :**
  - ✓ Vitesse de calcul moins rapide qu'en C/C++.
  - ✓ Très gourmand en ressources informatiques (nouvel version).
  - ✓ Payant (équivalent gratuit Scilab).
  - ✓ Application auto-exécutable peu pratique.

De manière générale, Matlab est utilisé pour faire des expériences de calcul très rapidement. Certains programmes qui nécessiteraient une journée de programmation en C/C++ peuvent se réaliser en une heure sous Matlab. Par contre, une fois programmé, le temps de calcul sous Matlab peut être 100 fois supérieur à celui du C/C++. De ce fait, on ne l'utilise que très peu pour réaliser un produit finit destiné aux particuliers.

- **Langage C/C++ :** Le C++ (qui est composé du C en grande partie) est un langage assez éloigné du binaire, qui permet généralement de développer de façon plus souple et rapide et c'est l'un des langages de programmation les plus utilisés. Il est à la fois facile à utiliser et très efficace, c'est un langage dit «de bas niveau ». La complexité du langage est inévitable

lorsqu'on cherche à avoir beaucoup de fonctionnalités, néanmoins on aura entre les mains un langage très puissant et particulièrement rapide.

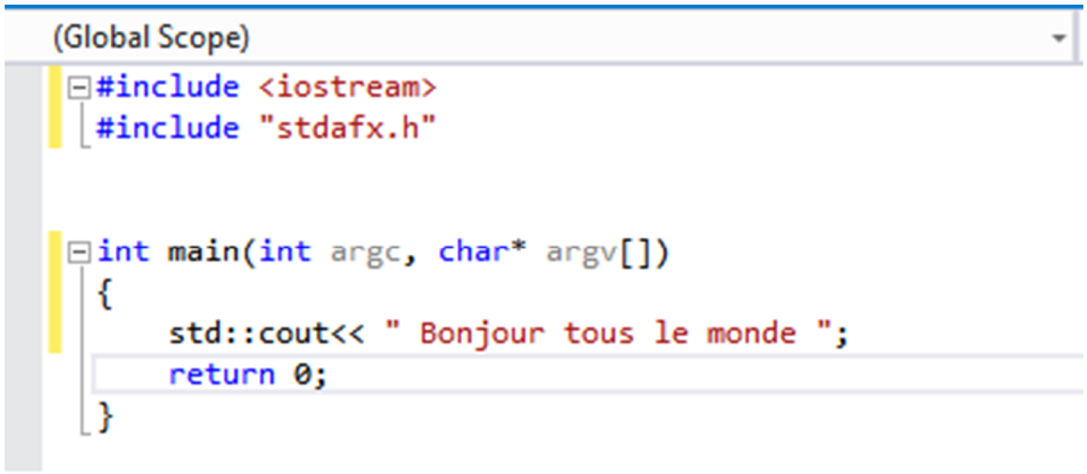


Figure 17 : exemple d'un programme en C++.

- **Avantages :**
  - ✓ Il est très répandu. Comme nous l'avons vu, il fait partie des langages de programmation les plus utilisés sur la planète. On trouve donc beaucoup de documentation sur Internet et on peut facilement avoir de l'aide sur les forums.
  - ✓ Il est rapide, très rapide même, ce qui en fait un langage de choix pour les applications critiques qui ont besoin de performances. C'est en particulier le cas d'algorithmes de traitement d'images qui doivent fonctionner en temps réel.
  - ✓ Il est portable : un même code source peut théoriquement être transformé sans problème en exécutable sous Windows, Mac OS et Linux.
  - ✓ Il est multi-paradigmes, qui signifie qu'on peut programmer de différentes façons en C++.
  - ✓ Programmation Orientée Objet (POO).

- **Inconvénients :**
  - ✓ Utilisation lourde : pas d'interactivité dans le développement (étape de compilation peut être pénible). Syntaxe compliquée (&, ::, }}, ; etc.). Gestion de la mémoire délicate. C'est l'un des langages les plus difficiles à manier pour le non-informaticien.
  - ✓ il est difficile d'écrire des programmes portables car le comportement exact des exécutables dépend de l'ordinateur cible.
  - ✓ le support de l'allocation de mémoire et des chaînes de caractères est minimaliste, ce qui oblige les programmeurs à s'occuper de détails fastidieux et sources de bugs ; il n'y a notamment pas de ramasse-miettes standard.

Il existe d'autres langages de haut niveau tels que JAVA, Python, etc. Mais pour la réalisation de notre projet, le langage C/C++ a été choisi. Bien entendu, le C/C++ n'est pas LE langage incontournable. Il a lui-même ses défauts par rapport à d'autres langages, sa complexité en particulier. On a beaucoup de contrôle sur le fonctionnement de l'ordinateur (et sur la gestion de la mémoire) : cela offre une grande puissance mais, une mauvaise utilisation, pourrait plus facilement faire planter le programme.

II.4.1.2 Le choix d'IDE :

Un IDE (*Integrated Developement Environment)* ou environnement de développement intégré, est un ensemble d'outils, à savoir un éditeur de texte destiné à la programmation, des fonctions qui permettent, par simple clique sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de

construction. Certains environnements sont dédiés à un langage de programmation en particulier.

Le choix d'IDE n'était pas des plus faciles, en effet la plupart des IDE populaires tels que Visual Studio, Eclipse, Code ::Blocks, Qt-creator, pouvaient nous offrir les mêmes fonctionnalités. Pour la réalisation du projet le choix s'est porté donc sur Visual Studio 2012 sous Windows Seven *figure 18* pour les critères suivants :

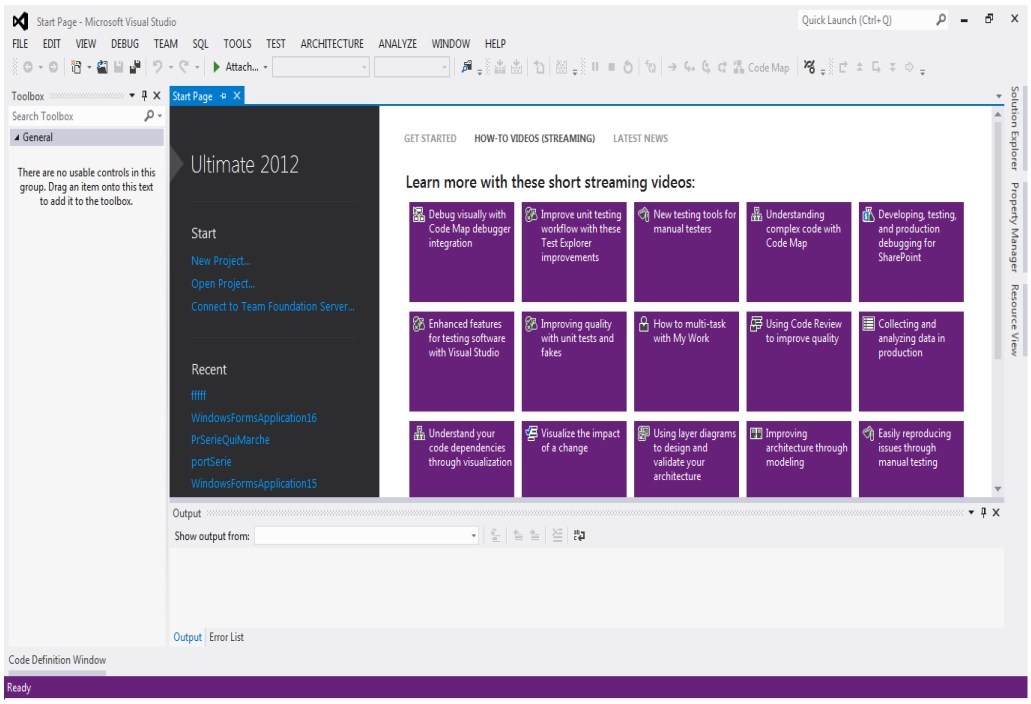


Figure 18 : IDE Visual Studio 2012.

- Le premier critère et sans doute le plus important, est la facilité d'intégration et d'utilisation de la fameuse bibliothèque OpenCV.
- Un plus grand nombre d'utilisateur que ses concurrents, donc une documentation riche.
- Un environnement agréable et interactif.

- Il comporte un bref descriptif sur les différentes fonctions dès l'insertion dans le programme, qui apparait sous forme de petit texte attaché au curseur.
- Une auto-complétion des plus rapides.
- Grande stabilité et un nombre de crash quasiment nul
- Compatibilité infaillible avec Windows du fait qu'ils soient conçus du même développeur « Microsoft ».
- Disponibilité d'une version gratuite intitulée Visual C++ Express.

II.4.2 Manipulation des images :

Pour la réalisation de cette partie du projet, on a dû recourir aux notions et outils suivant :

II.4.2.1 Présentation d'OpenCV :

OpenCV (Open Source Computer Vision) est une bibliothèque écrite en C/C++ visant principalement à la vision par ordinateur en temps réel. Elle a été initialement développée par Intel, maintenant soutenue par Willow Garage (un laboratoire de recherche en robotique créé en fin 2006 en Californie). Elle est gratuite sous la licence open source et multi- plateforme.

Avec plus de 500 algorithmes optimisés, elle comprend un ensemble complet d'algorithmes d'apprentissage classique et les dernières innovations en vision par ordinateur.

Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, détection de mouvements, et toutes autres sortes de traitements des images qui puissent exister.

II.4.2.2 Images en couleur :

La théorie Trichromatique des couleurs aussi appelée théorie de *Young-Helmholtz (Young 1802, Helmholtz 1850)*, postulant que l'œil humain est constitué de trois types de photorécepteur (les cônes), chacune particulièrement sensible à une longueur d'onde du domaine du visible.

Grace à la superposition de trois sources colorées (rouge, vert, bleu) à différentes intensités, notre œil est capable de distinguer plus de 1500 nuances par couleur, et un totale de plusieurs millions de couleurs.

Les trois types de cônes ont été catégorisés en fonction de leur sensibilité :

- aux ondes longues, les cônes L, sensibles au rouge (580 nm).
- aux ondes moyennes, les cônes M, sensibles au vert (545 nm).
- aux ondes courtes, les cônes S, sensibles au bleu (440 nm).

Dans ce chapitre nous allons présenter deux espaces colorimétriques fréquemment utilisés et qui sont nécessaire pour réaliser cette partie du projet.

- **L'espace RGB :** est un espace permettant de définir les couleurs de manière additive : chaque couleur est un mélange de Rouge (Red), Vert (Green), Bleu (Blue). Les écrans à tube cathodique, à cristaux liquides ou encore plasma codent les couleurs dans cet espace.

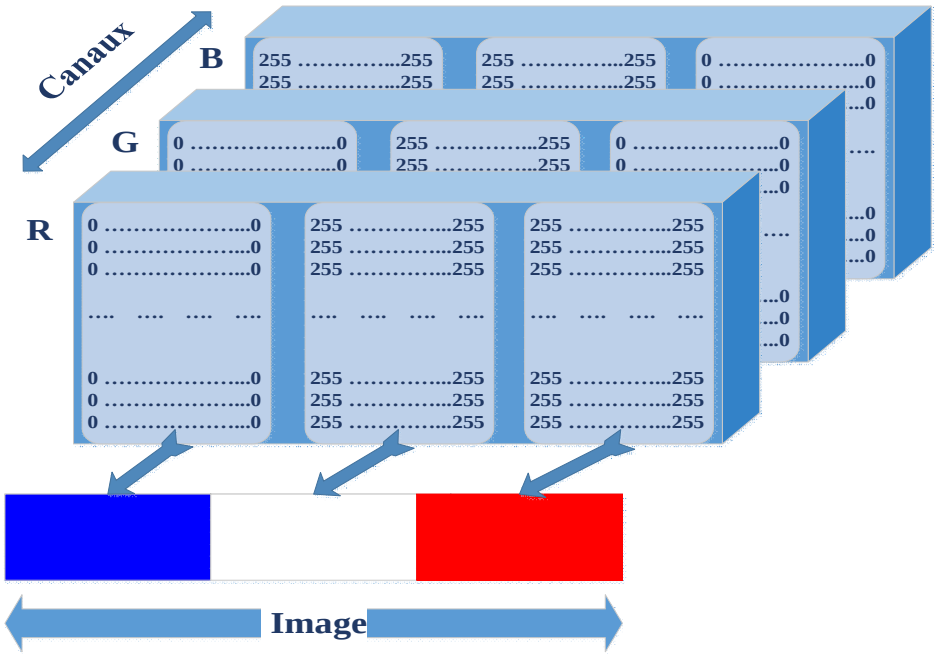


Figure 19 : Espace RGB avec ses trios canaux.

- **L'espace YC$_b$C$_r$:** il a été introduit en codage d'images pour la vidéo. Il permet, dans le cas d'images couleur, de compenser les erreurs de transmission de l'information
  - **Y :** représente la Luminance, l'information de niveau de gris.
  - **C$_b$** et **C$_r$:** représentent la chrominance et sont respectivement le bleu moins Y et le rouge moins Y.

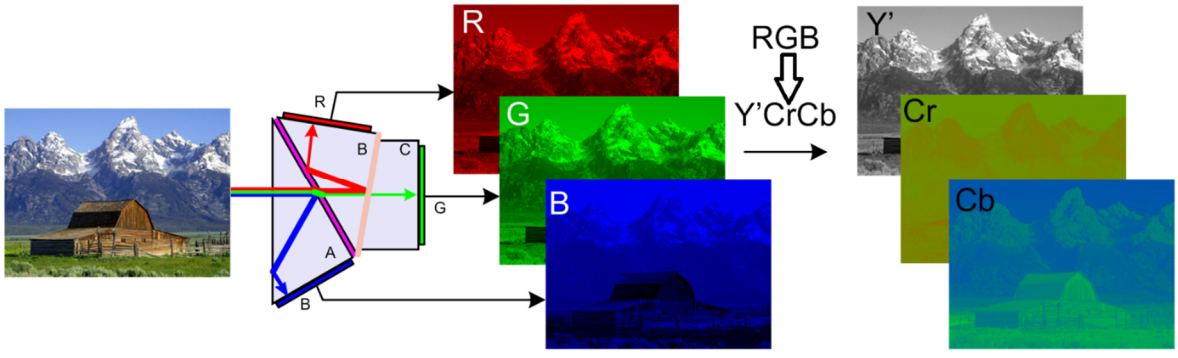


Figure 20 : résultat de conversion de RGB à YCbCr.

II.4.2.3 Binarisation des images couleur :

La binarisation d'une image consiste à affecter à chaque pixel une classe parmi deux classes possible C1 et C2. Un pixel a alors une valeur de 0 ou 1.

Pour une image monochrome, la binarisation la plus simple s'appuie sur un seuillage des niveaux de gris : si la classe C1 est celle de l'ensemble des pixels de position (x, y) dont l'intensité I(x, y) est supérieur ou égale à un seuil s, alors C2 est la classe des pixels d'intensité strictement inférieur à ce seuil. Cette opération de seuillage réalisée sur toute l'image conduit à une image binaire constituée de 0 et de 1 *figure 21*.

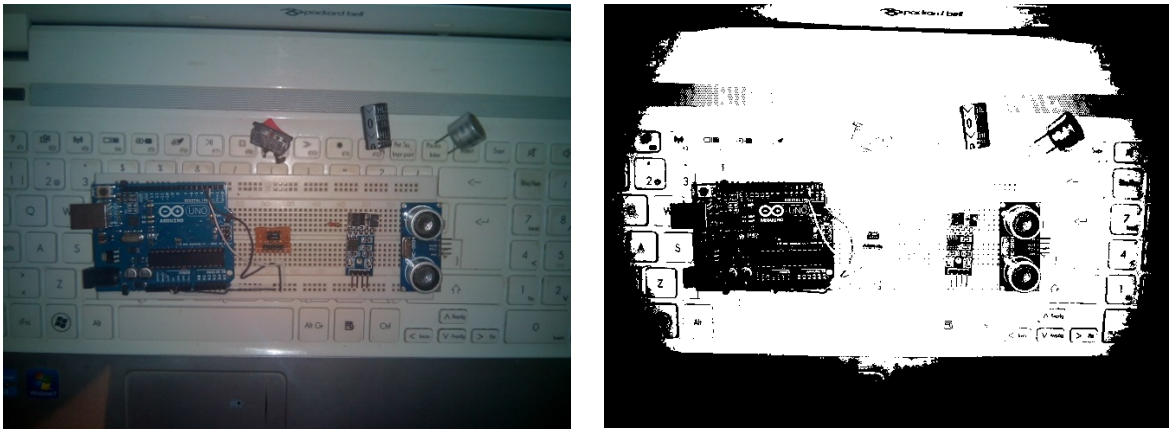


Figure 21 : Résultat de seuillage sur une image en couleur.

Binariser une image en couleur consiste à seuiller chacun de ses canaux, puis opérer un ET logique entre les images binaires obtenues (élément par élément)

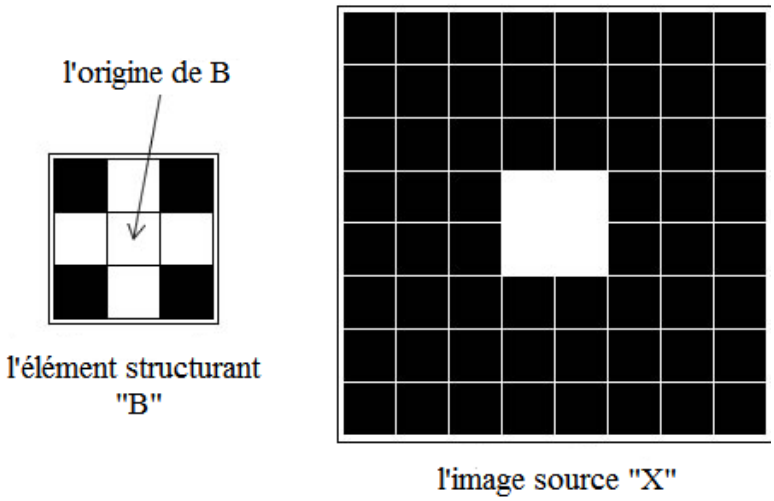
La fonction cvInRangeS de la bibliothèque OpenCV réalise un seuillage simultané de plusieurs canaux en une seule instruction.

II.4.2.4 Morphologie mathématique :

Les algorithmes de morphologie mathématique permettent de travailler sur les images binaires pour mettre en évidence leurs propriétés particulières.

Ces algorithmes sont basés sur l'utilisation d'"éléments structurants" (noyau), c'est à dire des configurations élémentaires de pixels à rechercher dans l'image.

Les éléments structurants sont beaucoup plus petits que l'image à traiter (généralement 3*3 pixels, mais peut être plus grand), ces éléments structurants parcourent l'ensemble des pixels de l'image et la transforme.



Les plus basic opérations sont **: Dilatation** et **Erosion**

- **Dilatation :** Pour chaque position du noyau B sur l'image X, si un, au moins, des pixels de B fait partie de X, alors l'origine de B appartient à l'image générée.

$$y = X \oplus B$$

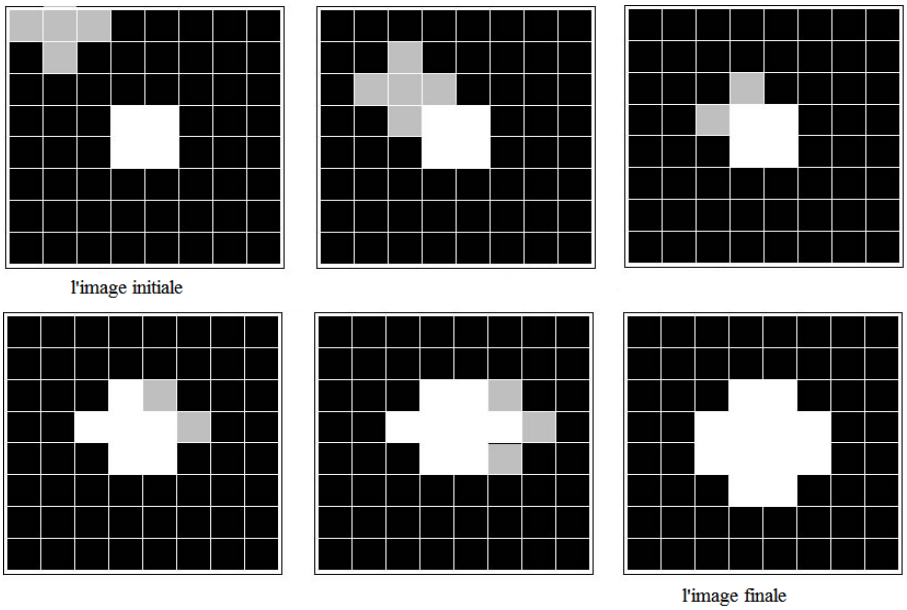


Figure 22 : Illustration de l'opération Dilatation.

- **Erosion :** Pour chaque position de B sur l'image X, si tous les pixels de B font partie de X, alors l'origine de B appartient à l'image générée.

$$y = X \ominus B$$

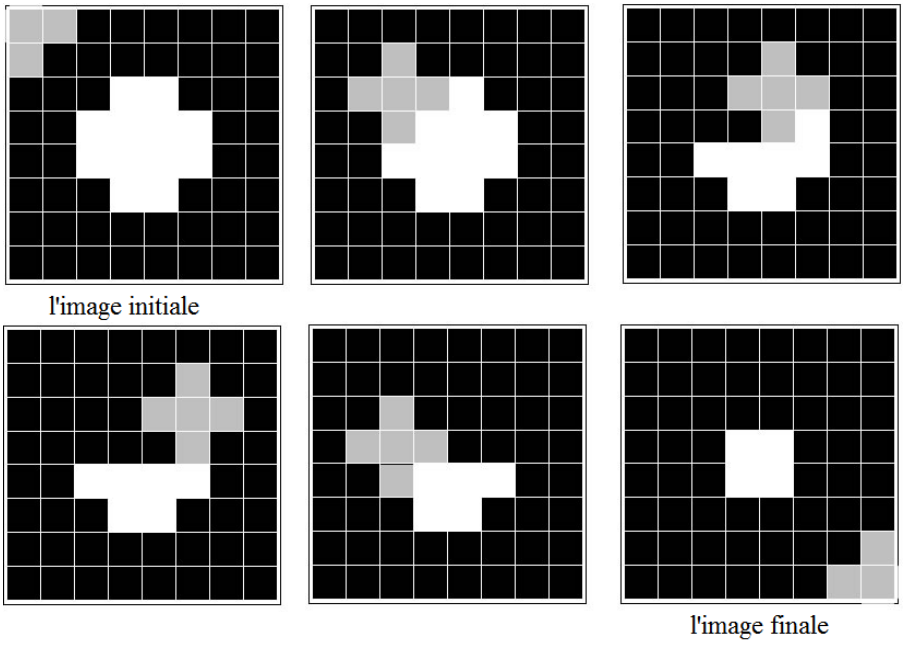


Figure 23 : illustration de l'opération Erosion

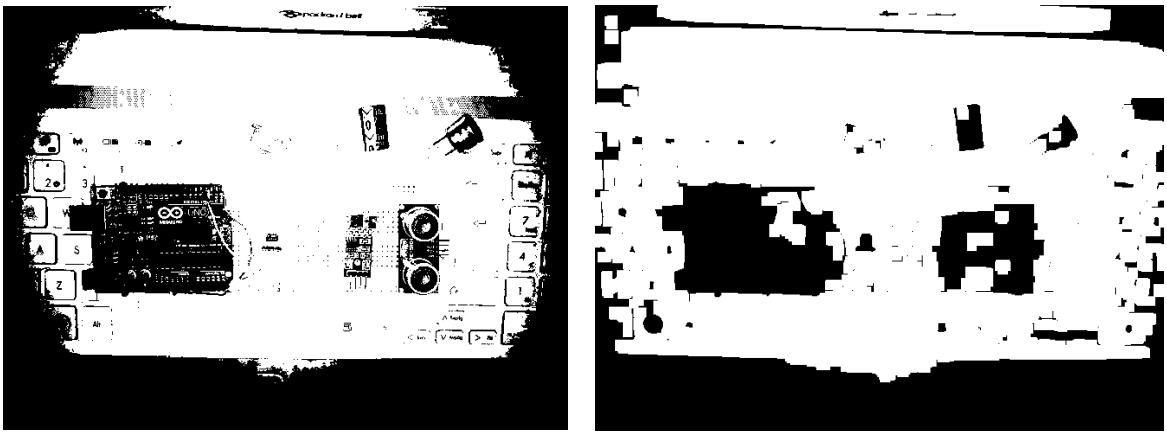


Figure 24 : Résulta d'application d'operateur morphologique sur une image binaire (Erosion → Dilatation).

II.4.2.5 Etiquetage des composantes connexes :

Une image binaire est constituée de plusieurs régions homogènes : chaque région est un ensemble de pixels connexes appelé composante connexe.

Le concept de d'étiquetage en composantes connexes s'applique généralement sur les images binaires, et consiste à trouver les différentes composantes connexes de l'image et à les étiqueter. L'étiquetage consiste à affecter une étiquette identique à tous les pixels d'une même composante connexe.

II.4.2.6 Détection de mouvement par différence d'images :

De manière plus générale, la détection des pixels en mouvement se fait à l'aide de la différence entre l'image courante, à l'instant *t*, et une image de référence I_{ref} à l'instant *t-1* :

$$\Delta I(x, y, t) = I(x, y, t) - I_{ref}(x, y, t-1), \qquad \forall (x, y)$$

L'image I_{ref} peut être celle du fond immobile dans le cas d'une caméra fixe, mais peut aussi être estimée par la moyenne des N images précédentes de la séquence.

Le seuillage se fait sur le module de la différence d'image :

$$D(x, y, t) = \|\Delta I(x, y, t)\| > seuil$$

D(t) est une image binaire ou les pixels en mouvement sont mis à 255, les autres à 0. Les images de la *figure 25* montrent montre le principe de la soustraction de deux images :

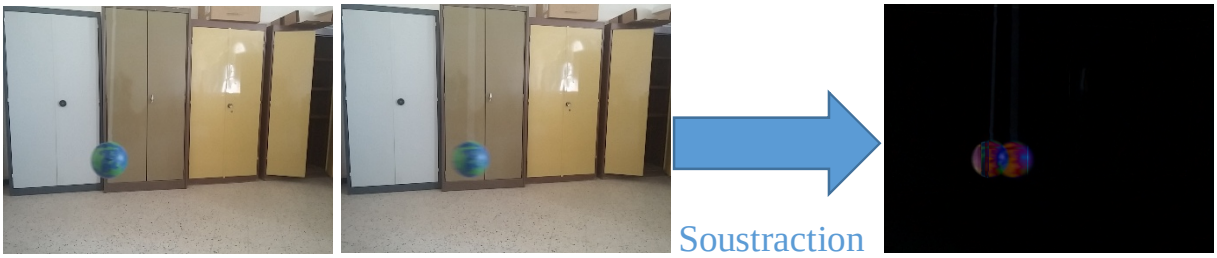


Image (a) t-1. *Image (b) t.* *Image (a – b).*

Le mouvement (binarisation (a-b)).

Figure 25 : Principe de détection de mouvement par différence d'images.

II.4.3 Interface d'affichage :

Cette partie du projet consiste en la programmation d'une interface graphique, qui aura pour rôle d'afficher les données acquises du module d'acquisition et la capture des images issues de la webcam en quelques clics sur boutons.

L'environnement de développement choisi est Visual Studio 2012 pour les critères suivants :

- La programmation de l'algorithme de traitement de la vidéo est réalisée en C++, de ce fait une continuation avec le même langage est de rigueur, si on veut s'évité les problèmes de compatibilité entre deux langages de programmation différents.
- Visual Studio offre la possibilité de créer un projet de programmation graphique nommé Windows Forms Application *figure 26* qui est riche en toolbox constitué de différents objets interactifs tels des boutons, bar de progression etc.
- La facilité d'intégration de la bibliothèque OpenCV dans Windows Forms Application.
- La possibilité d'exporter le projet final en un exécutable fini installable sur les autres machine Windows.

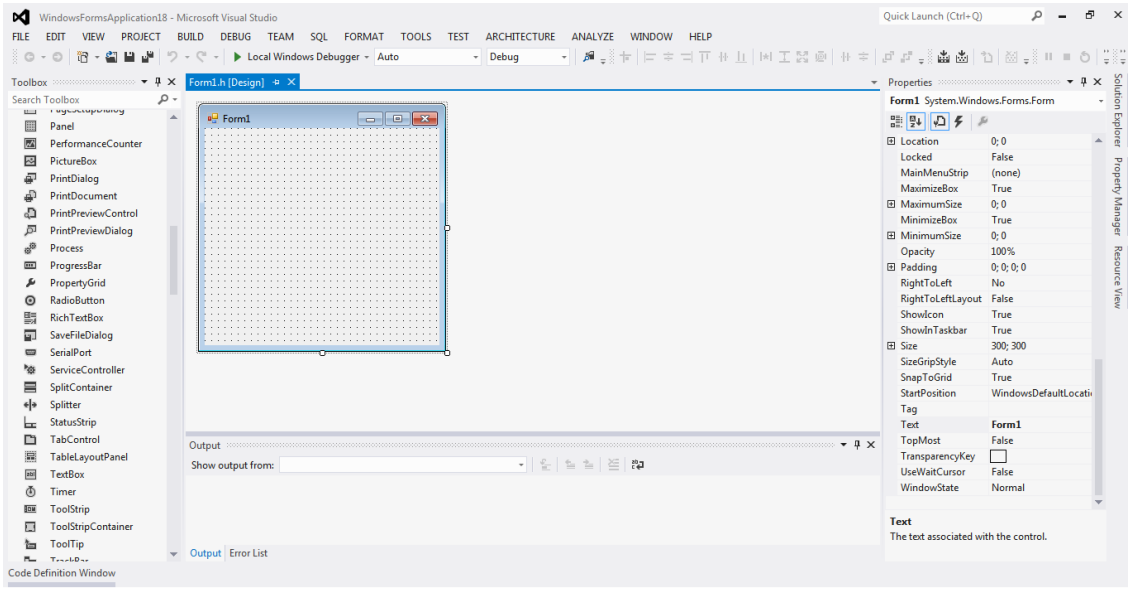


Figure 26 : Windows Forms Application.

II.5 Conclusion :

Dans ce chapitre, nous avons passé en revue l'architecture matérielle et logiciel de notre système de mesure de flambement et de déformation ainsi que les principales caractéristiques des composants qui vont le constituer et pour lesquels, nous avons justifié leurs sélection. De même, nous avons présenté leurs configurations externes et internes et la manière avec laquelle, chacun d'eux, peut être connecté avec les autres.

Chapitre III

Réalisation du système de mesure

III.1 Introduction :

Après avoir conçu le système de mesure de flambement et de déformation, la réalisation pratique et tests du système font l'objet de ce chapitre, qui se présente sous la forme matériel et logiciel.

III.2 Réalisation :

III.2.1 Réalisation matérielle :

La réalisation matérielle est faite en premier lieu, chaque partie est réalisée et testée séparément. Les montages sont d'abord construits sur des « *breadboard* » ou Lab d'essai.

Après les avoir expérimentés et adoptés séparément, nous les avons regroupés et réalisés sur un circuit imprimé qui s'emboite parfaitement sur la carte Arduino Uno. La *figure 27* est le schéma de conception du circuit imprimé.

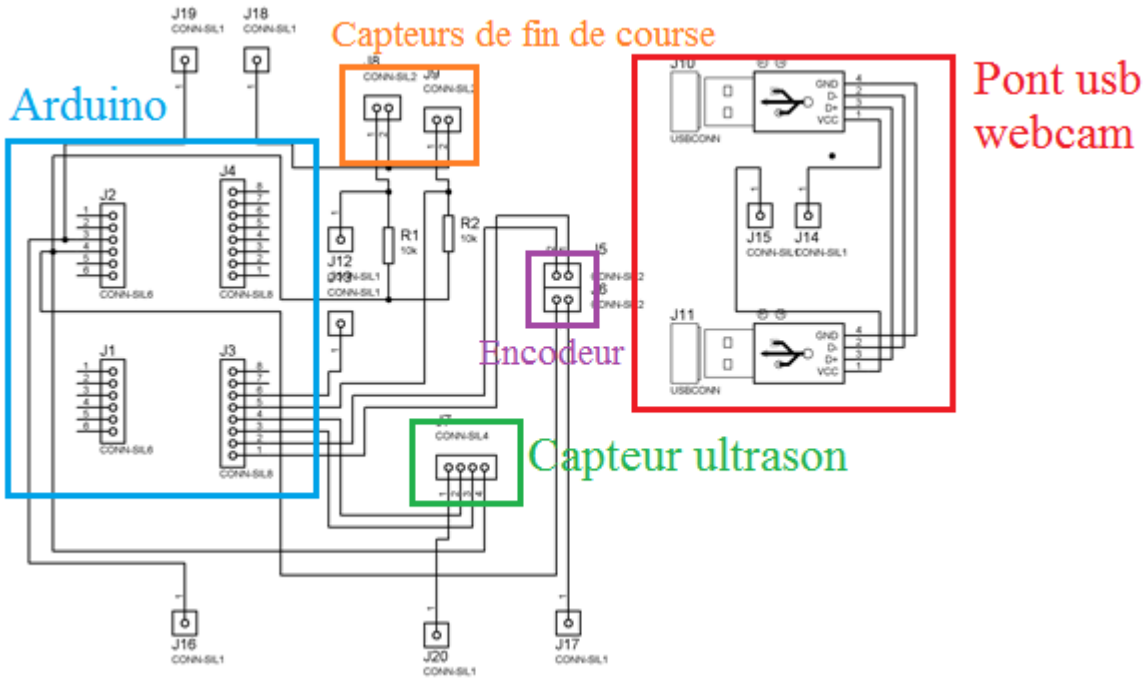


Figure 27 : Schéma de conception du circuit.

Le circuit électronique qui englobe le système de mesure est divisé en quatre parties principales :

- La carte Arduino Uno.
- Le capteur ultrason.
- L'encodeur.
- Deux capteurs de fin de course.

III.2.1.1 La carte Arduino :

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

- Les entrées/sorties numériques ont été configurées.
- Les broches 2 et 3 attachées aux interruptions externes 0 et 1 respectivement, sont réservées aux sorties de l'encodeur.
- Les broches 4 et 5 sont réservées au capteur ultrason HC-SR04, la broche 4 en entrée pour la sortie *Echo* du capteur et la broche 5 en sortie pour la broche *Trig* du capteur ultrason.
- Les broche 6 et 7 sont configurées en entrées, et sont réservées aux deux capteurs de fin de course.

III.2.1.2 Le capteur ultrason :

Le principe de fonctionnement du capteur ultrason HC-SR04 est simple, il est expliqué en détail dans le chapitre précédant : on envoie une impulsion d'une durée de 10µs sur la broche 5 de l'Arduino qui correspond à la patte *Trig* du capteur et on reçoit une impulsion sur la broche 4 de l'Arduino qui correspond à la patte *Echo* du capteur, d'une durée proportionnelle à la distance qui sépare le

capteur de l'obstacle. Le schéma de brochage est réalisé à l'aide du logiciel open source « Fritzing », comme le montre la *figure 28* :

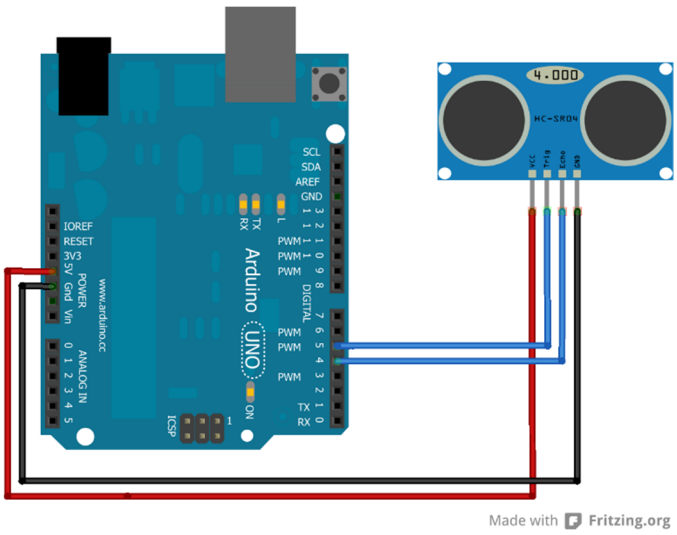


Figure 28 : Schéma de brochage du HC-SR04.

L'organigramme de fonctionnement du capteur ultrason est le suivant :

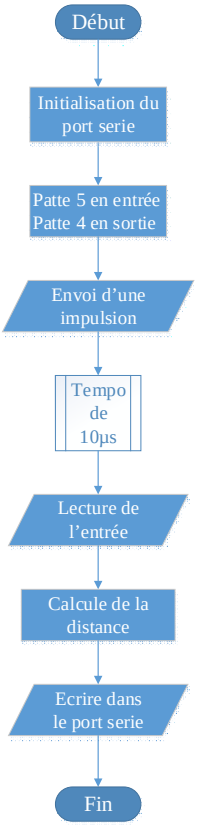
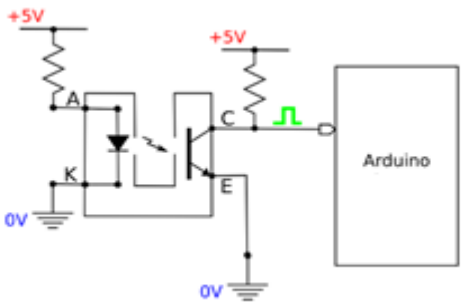


Figure 29 : Organigramme du capteur ultrason.

III.2.1.3 L'encodeur :



L'encodeur est un capteur optique TOR composé de deux éléments une led infrarouge et un phototransistor (ou deux). L'utilisation d'un film transparent gradué (graduation de couleur noir très dense) a pour rôle de couper le faisceau lumineux entre la led et la base du phototransistor, le résultat est un train de d'impulsion qu'on peut exploiter pour calculer le déplacement. L'encodeur utilisé possède deux sortie A et B idéal pour la détection du sens de déplacement, qui seront connectés aux broches 2 et 3 *figure 29* de l'Arduino le déplacement du film à travers l'encodeur se traduit par une impulsion qui génère une interruption externe a l'Arduino. Le nombre d'interruptions est égale au nombre d'impulsions générés par l'encodeur. La *figure 30* montre le principe de détection de sens de déplacement avec l'encodeur.

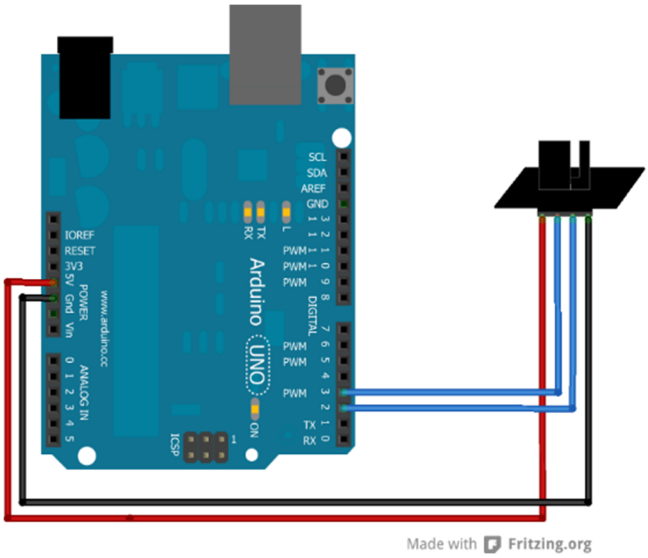


Figure 29 : Schéma de brochage de l'encodeur.

Pin 3	Pin 2	Pin 3	Pin 2	Résultat
T2	T2	T1	T1	
0	0	0	0	Pas de mvt
0	0	0	1	+1
0	0	1	0	-1
0	0	1	1	+2 (supposé pin1 =1)
0	1	0	0	-1
0	1	0	1	Pas de mvt
0	1	1	0	-2
0	1	1	1	+1
1	0	0	0	+1
1	0	0	1	-2 (pin 1 = 1)
1	0	1	0	Pas de mvt
1	0	1	1	-1
1	1	0	0	+2 (pin 1 =1)
1	1	0	1	-1
1	1	1	0	+1
1	1	1	1	Pas de mvt

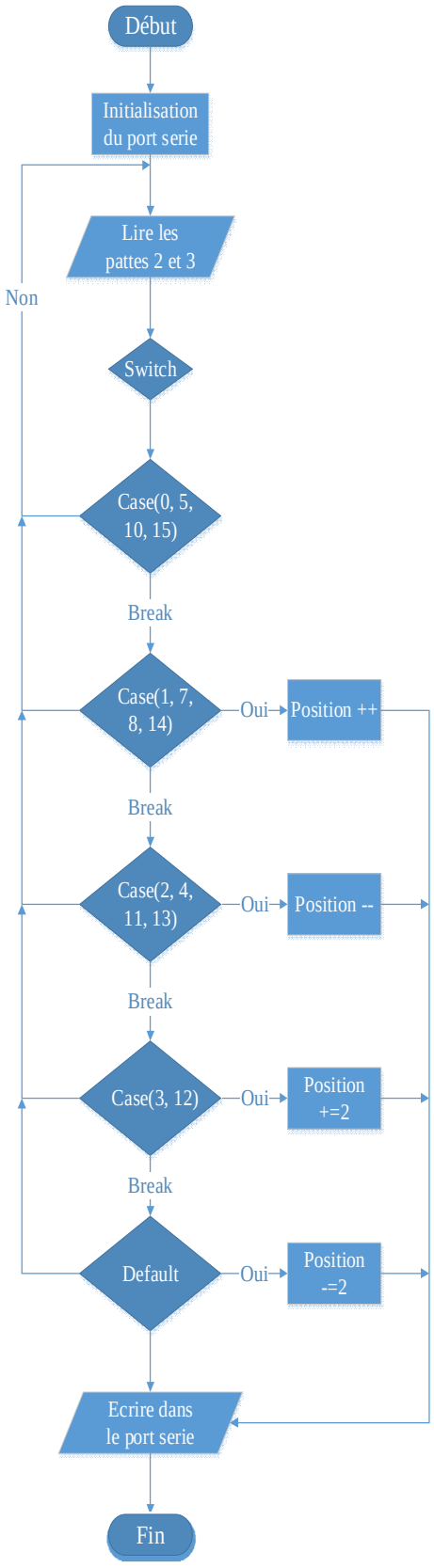


Figure 30 : Table de vérité et organigramme de détection de sens de l'encodeur.

III.2.1.4 Les deux capteurs de fin de course :

Les deux capteurs de fin de course vont servir de point de synchronisation, l'un pour le vérin hydraulique, l'autre pour le capteur de déformation, seront connectés aux pattes 6 et 7 de l'Arduino configurés en entrées. Le schéma de brochage est représenté dans la *figure 31* :

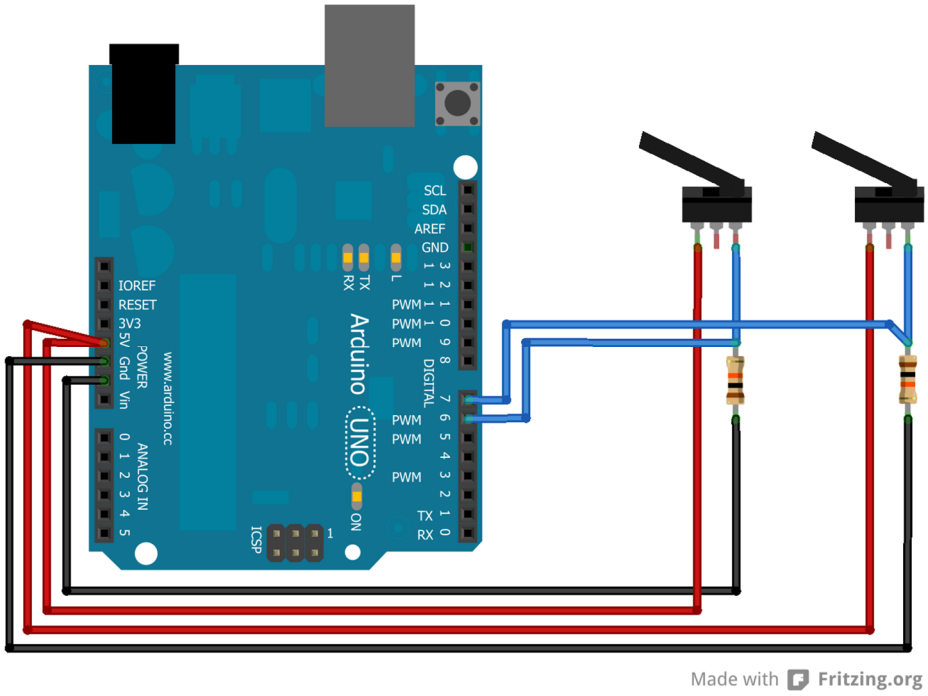


Figure 31 : Schéma de brochage des fins de course.

III.2.2 Réalisation logicielle :

La réalisation logicielle consiste en grande partie en la mesure du flambement à l'aide d'une webcam HD, et la création d'une interface graphique facile à installer et à utiliser sous Windows.

L'environnement de développement est Visual Studio 2012, en langage C++. Le programme de traitement d'image permet de faire :

- La capture des images de la webcam.
- La sauvegarde de l'arrière-plan.

- La soustraction de deux images consécutives.
- Le seuillage du résultat de la soustraction.
- La détection de contour.
- Le calcul de la distance séparant les deux contours.
- Afficher le résultat sur l'écran.

L'organigramme du traitement d'image est le suivant :

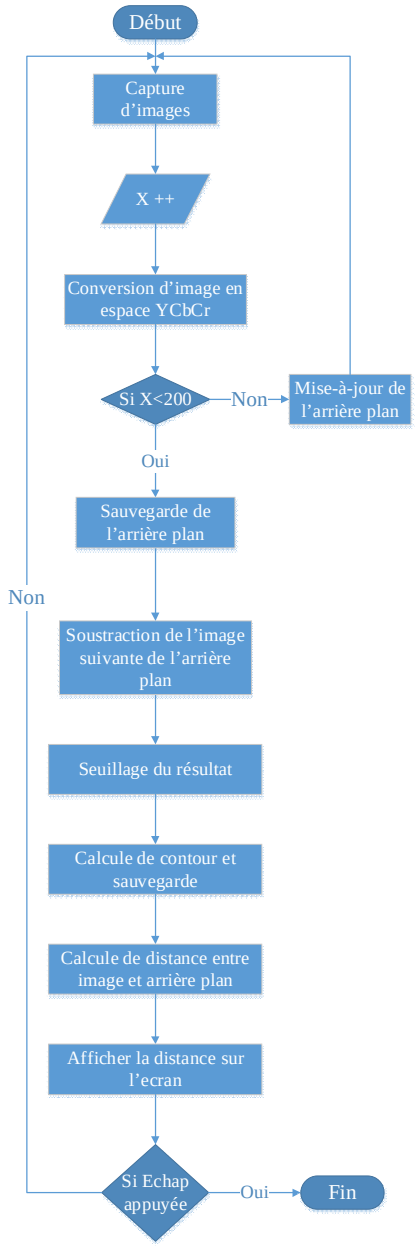


Figure 32 : Organigramme de calcul de flambement avec la webcam.

III.3 Fonctionnement du système :

Le système de mesure est piloté par une interface graphique nommé Terminal Série *figure 33*. Les différentes parties du logiciel sont expliquées dans ce qui suit :

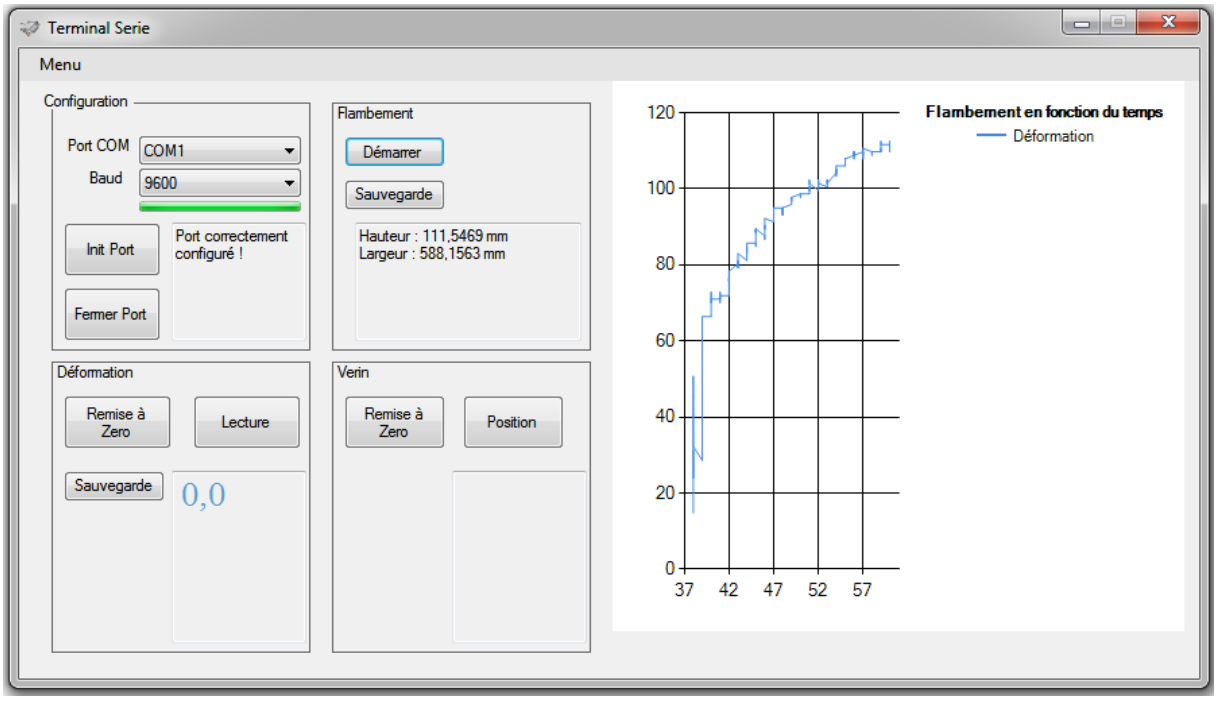
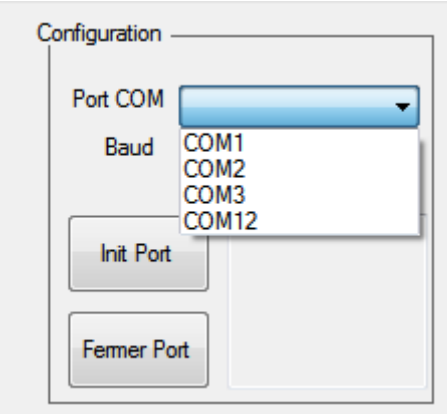


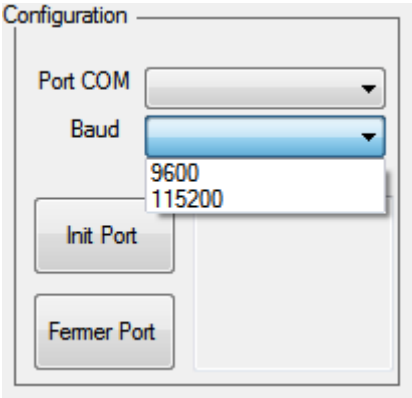
Figure 33 : Le Logiciel Terminal Série

➢ La partie Configuration :

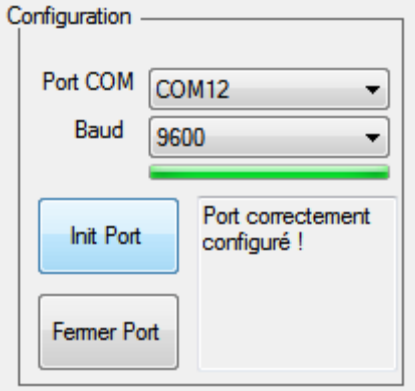
La partie Configuration s'occupe de la communication entre l'Arduino et le logiciel Terminal Série (l'ouverture, la configuration et la fermeture du port série), elle est constituée de :



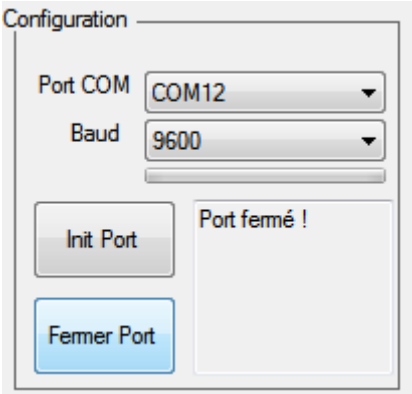
- **Port COM :** pour choisir le nom du port virtuel série utilisé, pour l'Arduino le COM12 est utilisé.



- **Baud :** pour choisir le débit, généralement 9600baud/s



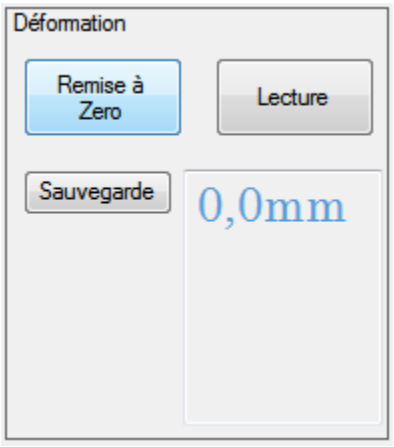
- **Le bouton Init Port :** une fois le port COM nommé, le débit choisi, l'ouverture du port virtuel s'effectue en appuyant sur ce bouton.



- **Le bouton Fermer Port :** l'appui sur ce bouton provoque la fermeture du port virtuel
- **Une zone d'affichage :** elle affiche l'état du port virtuel.

➢ La partie déformation :

La partie déformation du logiciel se charge de récupérer les données du capteur de déformation (l'encodeur) de l'Arduino vers le Terminal Série et est constituée de :



- **Le bouton Remise à Zéro :** comme son nom l'indique, en cliquant sur ce bouton une remise à zéro du capteur s'effectue.

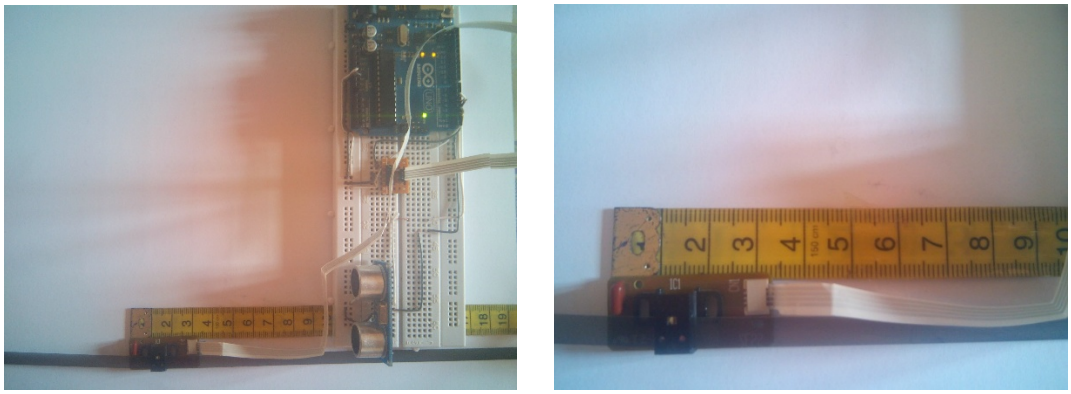


Figure 34 : Position initiale de l'encodeur.

- **Le bouton Lecture :** cliquer sur ce bouton permet de lire la mesure de déplacement fournie par le capteur.
- **La zone d'affichage :** permet d'afficher la mesure en temps réel.
- **Le bouton sauvegarder :** permet de sauvegarder le résultat de la mesure sur un fichier texte.

Figure 35 : Déplacement de l'encodeur.

➢ La partie flambement :

La partie flambement du logiciel s'occupe du calcule de flambement. Elle est constituée de :

- **bouton démarrer :** un seul clique sur ce bouton déclenche la capture des images de la webcam, l'algorithme de traitement

d'image, le calcule de flambement et l'affichage sur l'écran comme le montre la *figure 36.*

- **bouton sauvegarder** : permet de sauvegarder le résultat de la mesure dans un fichier texte.
- **La zone d'affichage :** permet d'afficher le résultat de la mesure.

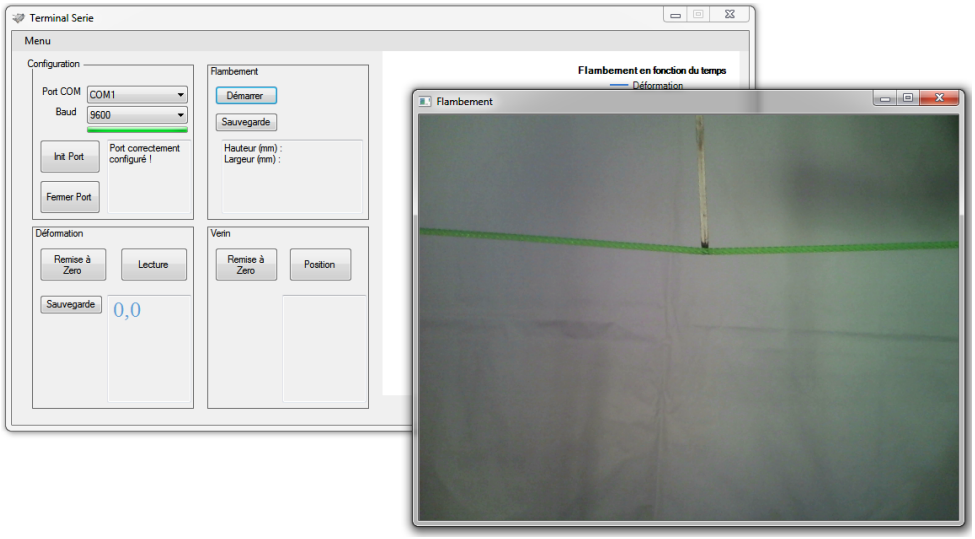


Figure 36 : Début de la capture des images et calcul de flambement.

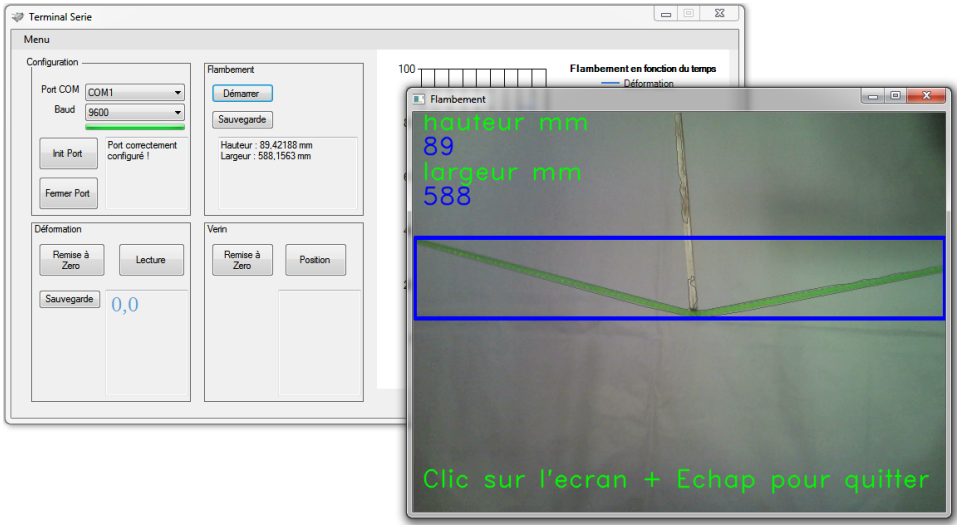


Figure 37 : Traitement de la vidéo et affichage du résultat de calcul de flambement en temps réel sur l'écran.

➢ La partie vérin :

La partie vérin du logiciel s'occupe de récupérer la position du vérin en temps réel et composée de :

- **Le bouton position :** cliquer sur ce bouton permet d'afficher la position exacte du vérin hydraulique.
- **Le bouton Remise à Zéro :** comme pour la partie déformation, ce bouton permet de réinitialiser la position du vérin.
- **La zone d'affichage :** pour afficher la position du capteur.

Le résultat des tests sont les suivants :

Figure 38 : Résultat de calcul de position avec le capteur ultrason et affichage sur l'interface.

➢ La zone graphique :

Permet de tracer un graphe de la mesure de déformation en fonction du temps en temps réel.

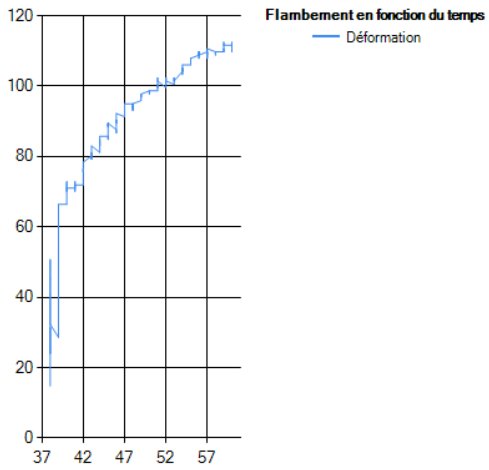


Figure 39 : Graphe représentant le flambement en fonction du temps.

➢ La barre de menu :

Composée de deux boutons :

- **Le bouton about :** contient un bref descriptif du logiciel Terminal Série.

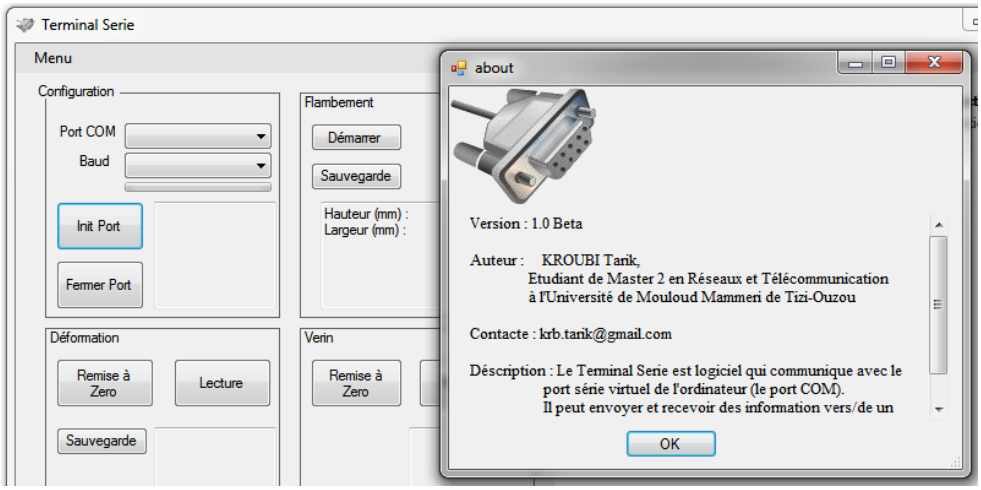


Figure 40 : fenêtre About du logiciel Terminal Série.

- **Le bouton Exit :** permet de quitter complètement l'application.

III.4 Installation :

L'installation des capteurs sur la machine est une opération délicate. En effet les capteurs doivent être installés dans la zone ou la fidélité et précision atteignent les plus hauts niveaux possibles (voir Annexe A), en plus de ne pas altérer le fonctionnement normal de la machine.

En raison de manque de temps, l'installation des capteurs et de la webcam n'a pas pu être achevée, mais les zones de fixation ont été repérées et étudiées, ce qui nous a permet de supposer leurs emplacements comme le montre la *figure 41*.

L'utilisation des anneaux de fixation pour l'encodeur pourrait s'avérer d'une grande importance pour des mesures exactes et fidèles de l'encodeur comme le KR 06 de Tecnotest.

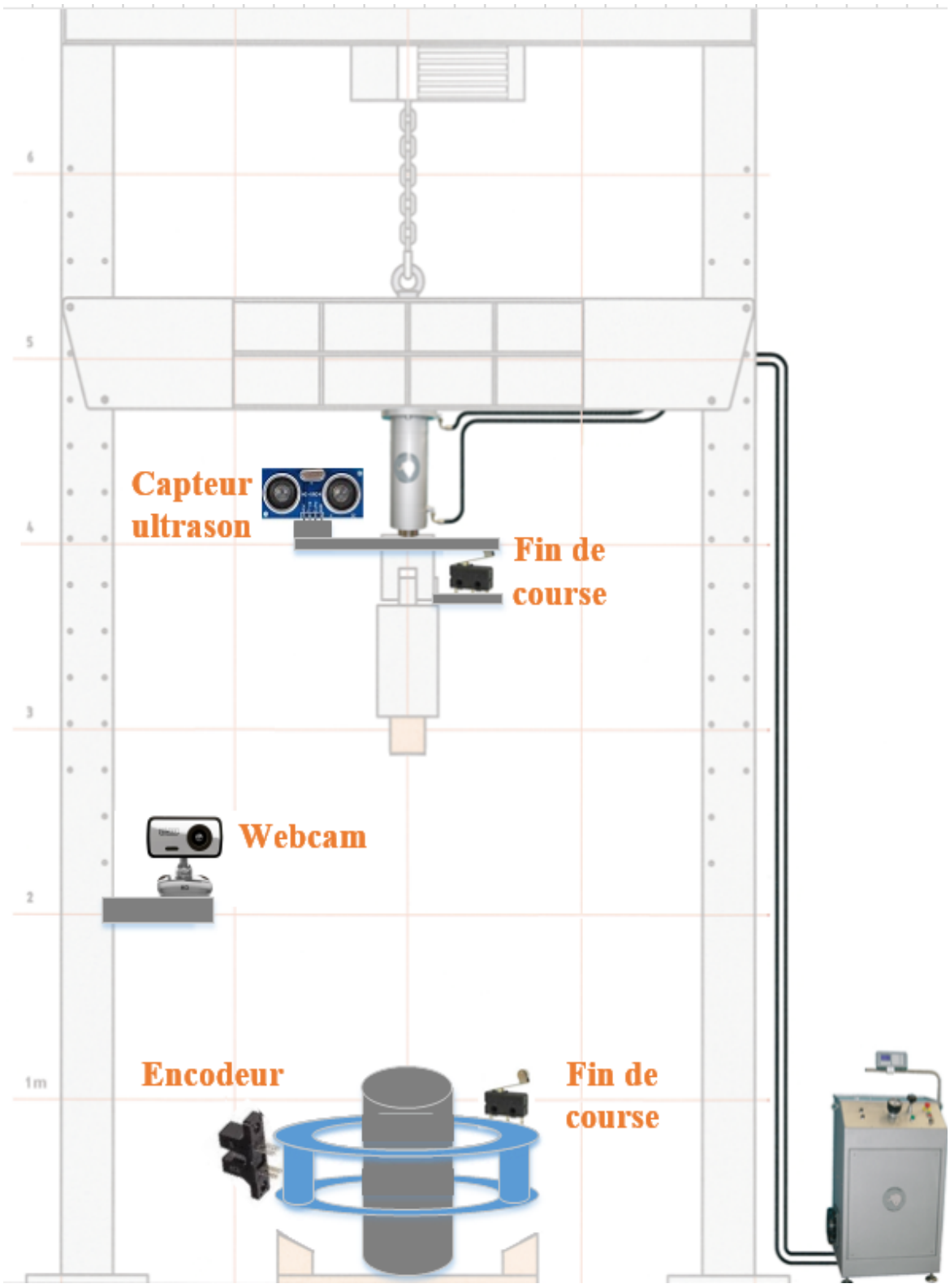


Figure 41 : Emplacement des capteurs sur la machine.

III.5 Conclusion :

Dans ce chapitre, nous avons décrit le processus de la réalisation matérielle et logicielle du système de mesure ainsi que sont l'installation et présenté les résultats obtenus.

Conclusion générale

Conclusion générale et perspective :

Dans ce travail, nous avons conçu et réalisé un système de mesure de flambement et de déformation à base de la carte Arduino Uno avec une transmission série des données vers un ordinateur. La visualisation des résultats de la mesure a été faite en utilisant le logiciel de programmation graphique Visual Studio.

Noun nous sommes intéressés à travers ce projet à développer un outil permettant d'aider les responsables de laboratoire de recherche du département de génie-civil à étendre le champ d'utilisation de la machine des essais et de tests.

La réalisation matérielle et logicielle de ce projet a donné des résultats satisfaisants, en effet elle apporte une souplesse et facilite l'utilisation de la machine à l'opérateur.

En perspective, notre système peut être amélioré en :

- Utilisant un encodeur incrémental avec une roue codeuse au lieu du capteur ultrason pour la mesure de la position du vérin hydraulique, pour éviter les fausses mesures dues aux éventuels obstacles entre le capteur ultrason et le plan de référence.
- L'utilisation d'une caméra IP Wifi avec un système de stéréovision (deux caméra) pour le calcul de la profondeur ce qui permet la liberté de déplacer la caméra.

Bibliographie

[1] Simon Monk, "**30 Arduino Projects For The Evil Genius**", 2010 edition The McGraw-Hill

[2] Martin Evans & Joshua Noble & Jordan Hochenbaum, "**Arduino in Action**", 2013 edition Manning Publications Co.

[3] John-David Warren & Josh Adams & Harald Molle, "**Arduino Robotics**", edition Technology In Action.

[4] Don Wilcher, "**Learn Electronics With Arduino**", edition Technology In Action.

[5] Michael Margolis, "**Make an Arduino-Controlled Robot**", 2013 edition O'Reilly.

[6] Michael Margolis**,** "**Arduino Cookbook**"**,** 2012 edition O'Reilly.

[7] Robert Laganière, "**OpenCV 2 Computer Vision Application Programming Cookbook** ", 2011 edition Packt.

[8] Open source, "**The OpenCV Tutorials Release 2.4.9.0**", 14 Avril 2014.

[9] Maria Petrou & Costas Petrou, "**Image Processing: The Fundamentals**", 2010 edition John Wiley & Sons Ltd.

[10] Gary Bradski and Adrian Kaehler, "**Learning OpenCV**", 2008 edition O'Reilly.

[11] Daniel Lélis Baggio, Shervin Emami, David Millán Escrivá, Khvedchenia Ievgen, Naureen Mahmood, Jason Saragih, Roy Shilkrot, "**Mastering OpenCV with Practical Computer Vision Projects**", 2012 edition Packt.

[12] Samarth Brahmbhatt, "**Practical OpenCV**", 2013 edition Technology In Action.

[13] Eskimon & olyte, "**Arduino pour bien commencer en électronique et en programmation**", 2012 openclassrooms.

[14] Ivor Horton, "**Beginning Visual C++ 2012**", 2012 edition John Wiley & Sons, Inc.

[15] Richard Banks, "**Visual Studio 2012 Cookbook**", 2012 edition Packt.

[16] Anonyme, "**20_Unbelievable_Arduino_Projects**".

[17] Tecnotest, "**Complete_Concrete_Catalogue**".

[18] Mathieu Nebra, "**apprenez à programmer en C**", mars 2012.

[19] Mathieu Nebra & Matthieu Schaller, "**programmez avec le langage C++**", juillet 2011.

[20] Rachid Bellaroussi, "**Traitement de l'image et de la vidéo avec exercices pratiques en Matlab et C++**", 2010 edition ellipses.

Sites Web :

[1] www.arduino.cc

[2] www.abcelectronique.com

[3] www.willowgarage.com/pages/software/opencv.com

[4] www.fr.wikipedia.org

[5] www.opencv.org

[6] www.fr.openclassrooms.com

[7] www.developpez.net

[8] www.sourceforge.net

[9] www.github.com

[10] www.microsoft.com/france/visual-studio.com

[11] www.msdn.microsoft.com

[12] www.fritzing.org

[13] www.mon-club-elec.fr

[14] www.alldatsheet.com

Annexe A

1. Généralités sur les Capteurs :

1.1 Définition :

Un capteur est un dispositif transformant une grandeur physique (température, pression, position, concentration, etc.) en un signal (souvent électrique) qui renseigne sur cette grandeur *figure 1*. Il se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable.

Figure 1 : Schéma d'un capteur.

Mesurer une grandeur physique c'est attribuer une valeur quantitative en prenant pour référence une grandeur de même nature appelée unité

- **Le mesurande :** c'est l'objet de la mesure ou plus simplement la grandeur à mesurer.
- **Le mesurage :** c'est l'ensemble des opérations pour déterminer la valeur du mesurande.
- **La mesure :** c'est le résultat du mesurage. Autrement dit c'est la valeur du mesurande.

Ces définitions permettent de donner une définition claire d'un capteur. En effet un capteur est un dispositif dont les caractéristiques physiques sont sensibles à un mesurande. Une relation mathématique tirée des lois physiques entre la grandeur d'entrée (mesurande m) et la grandeur de sortie S (S = f(m)) s'appelle courbe d'étalonnage du capteur.

Le capteur est dit linéaire si la courbe d'étalonnage est une droite ou sinon le capteur est dit non linéaire.

1.2 Classification des capteurs :

On peut classer les capteurs de plusieurs manières :

- par le mesurande qu'il traduit (capteur de position, de température, de pression, etc.)
- par son rôle dans le processus industriel (contrôle de produit finis, de sécurité, etc.)
- par le signal qu'il fournit en sortie qui peut être numérique, analogique, logique ou digital.

- par leur principe de traduction du mesurande (capteur résistif, piézoélectrique, etc.)
- par leur principe de fonctionnement : capteur Actif ou Passif

Toutes ces classifications permettent d'avoir une vue d'ensemble des capteurs et bien sur aucune des méthodes de classification n'est meilleure que l'autre car toutes présentent des avantages et des inconvénients.

Dans la suite ce chapitre, nous avons décidé de classer les différents capteurs par le mesurande qu'il traduit.

1.2.1 Apport énergétique :

- **Capteurs actifs :** Ce capteur fonctionne comme un générateur, dès qu'il est soumis à l'action d'un mesurande celui-ci transforme celle-ci en une grandeur directement exploitable à savoir en énergie électrique.

Le tableau suivant donne les principes physiques les plus utilisés en fonction d'un mesurande :

Mesurande	**Énergie propre du mesurande**	**Principe physique**	**Grandeur de sortie**
Température	Énergie thermique	Effet thermoélectrique Effet pyroélectrique	Tension Charge
Flux lumineux	Énergie électromagnétique	Effet photoémissif Effet photovoltaïque Effet photoélectrique	Courant Tension Tension
Force Pression Accélération	Énergie mécanique	Effet piézoélectrique	Charge
Vitesse	Énergie mécanique	Effet d'induction électromagnétique	Tension
Position	Énergie mécanique	Effet Hall	Tension

Table 1 : principes physiques des capteurs actifs.

- **Capteurs passifs :** Un capteur passif est considéré comme une impédance dont l'un des paramètres déterminants (résistivité, longueur, section) est sensible au mesurande. Cette impédance doit ensuite être intégrée dans un circuit pour pouvoir retrouver une grandeur électrique en sortie. Le montage qui permet ceci est appelé conditionneur. Il existe plusieurs sortes de conditionneur comme le montage potentiométrique, le pont de Wheatstone, les circuits oscillants ou les amplificateurs opérationnels.

Le tableau suivant donne différents capteurs passifs :

Mesurande	**Caractéristiques électriques sensibles**	**Types de matériaux utilisés**
Température Très basse température	Résistivité Constante diélectrique	Métaux : platine, nickel, cuivre Verres
Flux lumineux	Résistivité	Semi-conducteur
Déformation	Résistivité Perméabilité magnétique	Alliages de nickel, silicium dopé Alliages ferromagnétiques
Position	Résistivité	Matériaux magnéto-résistants : bismuth, antimoine d'indium
Humidité	Résistivité Constante diélectrique	Chlorure de lithium Alumine, polymères
Niveau	Constante diélectrique	Liquides isolants

Table 2 : principes physiques des capteurs actifs.

1.2.2 Types de sortie :

Les capteurs peuvent aussi faire l'objet d'une classification par type de sortie :

- **Capteurs analogiques :** La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur. La sortie peut prendre une infinité de valeurs continues. Le signal des capteurs analogiques peut être du type :
  - sortie tension.
  - sortie courant.
  - règle graduée, cadran, jauge (avec une aiguille ou un fluide).

  Quelques capteurs analogiques typiques :
  - capteur à jauge de contrainte.
  - LVDT.
  - thermocouple.
  - LDR.

- **Capteurs numériques :** La sortie est une séquence d'états logiques qui, en se suivant, forment un nombre.

  La sortie peut prendre une infinité de valeurs discrètes. Le signal des capteurs numériques peut être du type :
  - train d'impulsions, avec un nombre précis d'impulsions ou avec une fréquence précise.
  - code numérique binaire.
  - bus de terrain.

  Quelques capteurs numériques typiques :
  - codeur rotatif incrémental.
  - fourche optique.
  - capteur de température numérique DS18B20.

- **Capteurs logiques :** Ou capteurs TOR. La sortie est un état logique que l'on note 1 ou 0. La sortie peut prendre ces deux valeurs. Le signal des capteurs logiques peut être du type :
  - courant présent/absent dans un circuit.
  - potentiel, souvent 5 V/0 V.
  - DEL allumée/éteinte.
  - signal pneumatique (pression normale/forte pression).

  Quelques capteurs logiques typiques :
  - capteurs de fin de course.
  - capteurs de rupture d'un faisceau lumineux.
  - divers capteurs de position.

1.2.3 Type de détection :

- **détection avec contact :** Le capteur doit entrer en contact physique avec un phénomène pour le détecter.
- **détection sans contact :** Le capteur détecte le phénomène à proximité de celui-ci.

1.2.4 Capteurs composites :

Un capteur composite est un capteur constitué d'un corps d'épreuve et d'un capteur actif ou passif. Le corps d'épreuve quant à lui est un capteur qui soumis au mesurande donne une grandeur physique non électrique appelée mesurande secondaire qui elle va être traduit en une grandeur électrique par un capteur comme le montre le schéma suivant.

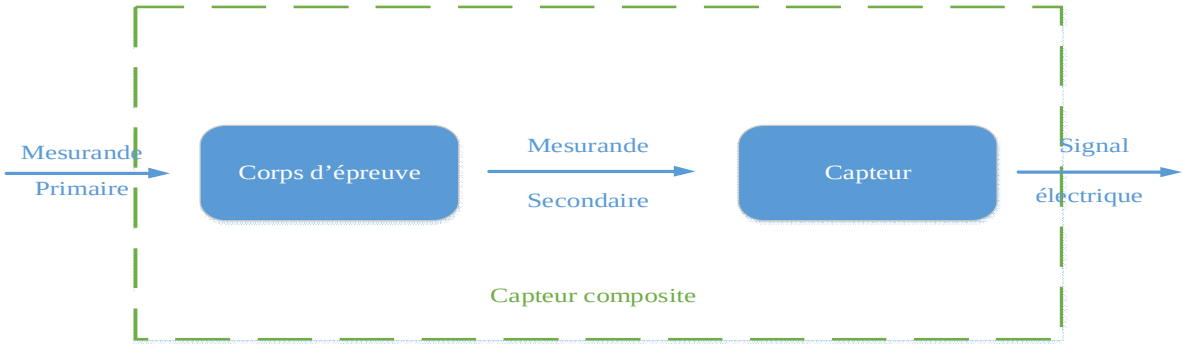


Figure 2 : Schéma d'un capteur composite.

Exemple de capteur composite :

- la mesure d'une force à partir d'un capteur de déplacement. Dans ce cas le corps d'épreuve est un ressort qui traduit la force (mesurande primaire) en élongation (mesurande secondaire) ensuite un capteur de déplacement traduira cette élongation en signal électrique.
- la mesure d'une accélération à partir d'un capteur de force. Le corps d'épreuve est une masse sismique qui traduit l'accélération (mesurande primaire) en force (mesurande secondaire). Celui-ci est ensuite transformé en signal électrique grâce à un capteur de force.

1.2.5 Les capteurs intégrés :

Un capteur intégré est un capteur qui utilise la microélectronique. Ce capteur est constitué d'une plaque en silicium dans lequel on a fixé le capteur, le corps d'épreuve si besoin et d'autres composants électroniques qui peuvent servir à linéariser, amplifier, convertir le courant en tension, etc.

Ce type de capteur est très utile vu qu'il fournit un signal linéaire avec une grande sensibilité, une miniaturisation et un coût faible.

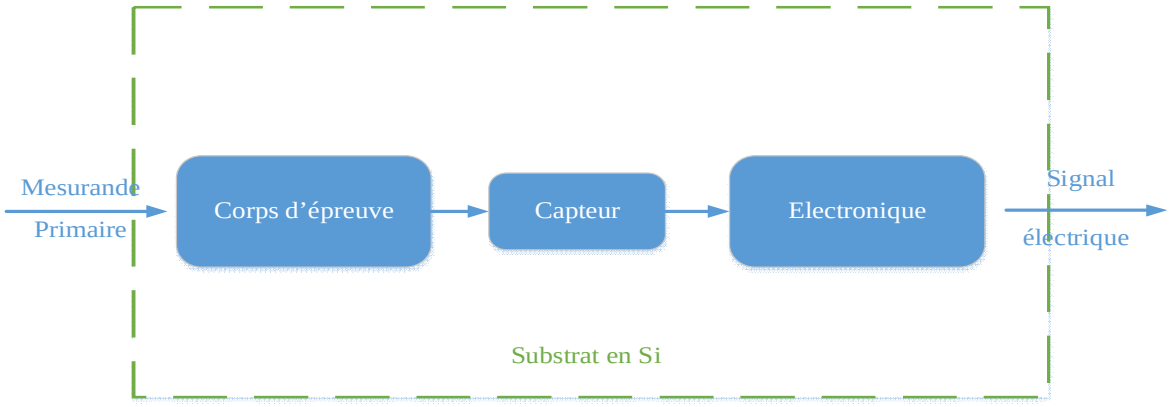


Figure 3 : Schéma d'un capteur intégré.

1.2.6 Les capteurs intelligents :

On désigne par capteur intelligent l'ensemble de mesure constitué de deux parties : une chaine de mesure pilotée par microprocesseur et une interface de communication bidirectionnelle.

La chaine de mesure comporte :

- Le capteur principal (spécifique au mesurande) et identifiable par un code stocké en PROM.
- Les capteurs secondaires propres aux grandeurs d'influence.
- Les dispositifs classiques de numérisation de la réponse de chaque capteur : conditionneur, multiplexeur, amplificateur etc. :

. En plus de leur faculté de mesurer une grandeur physique, il possède d'autres fonctionnalités dont voici quelques exemples :

- fonctions configurables de traitement du signal (filtre, gains…)
- fonctions d'auto-test et d'auto-contrôle
- étalonnage automatique
- sortie sur des bus de terrain

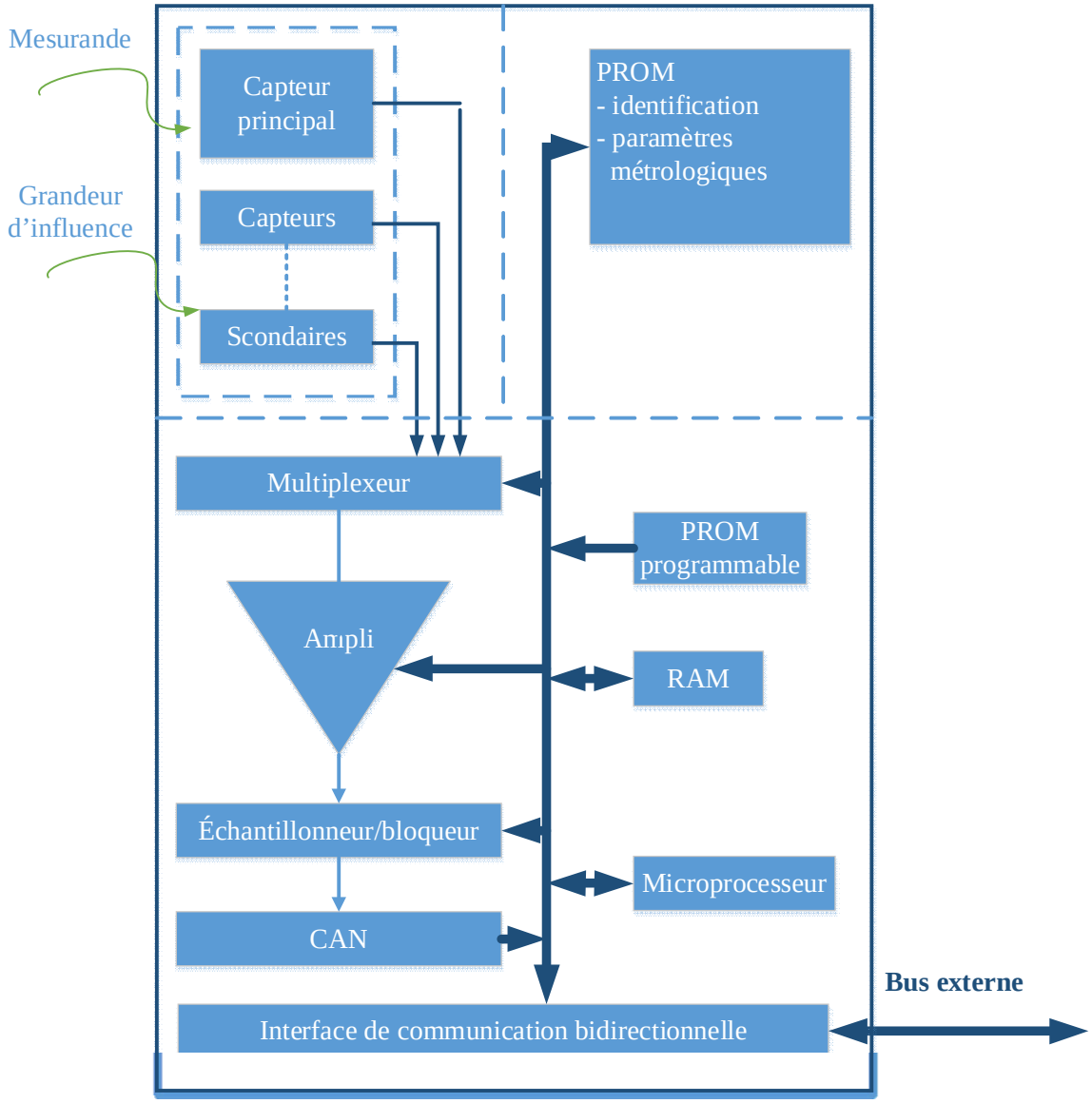


Figure 4 : Schéma d'un capteur intelligent.

1.3 Caractéristiques métrologiques :

Apres avoir fait le tour des capteurs et leurs classifications, il est maintenant utile de définir quelques caractéristiques métrologiques concernant les capteurs et qui peuvent s'avérer très utile dans le choix d'un capteur pour une mesure.

1.3.1 Limites d'utilisation et étendue de mesure :

Des modifications des propriétés et des caractéristiques du capteur peuvent apparaitre si celui-ci est soumis à des grandeurs d'influence telle la température, des contraintes

mécaniques ou électriques. Comme vous pouvez le voir sur le schéma il existe quatre domaines d'utilisation du capteur qui peuvent plus ou moins affecter les caractéristiques de ce capteur.

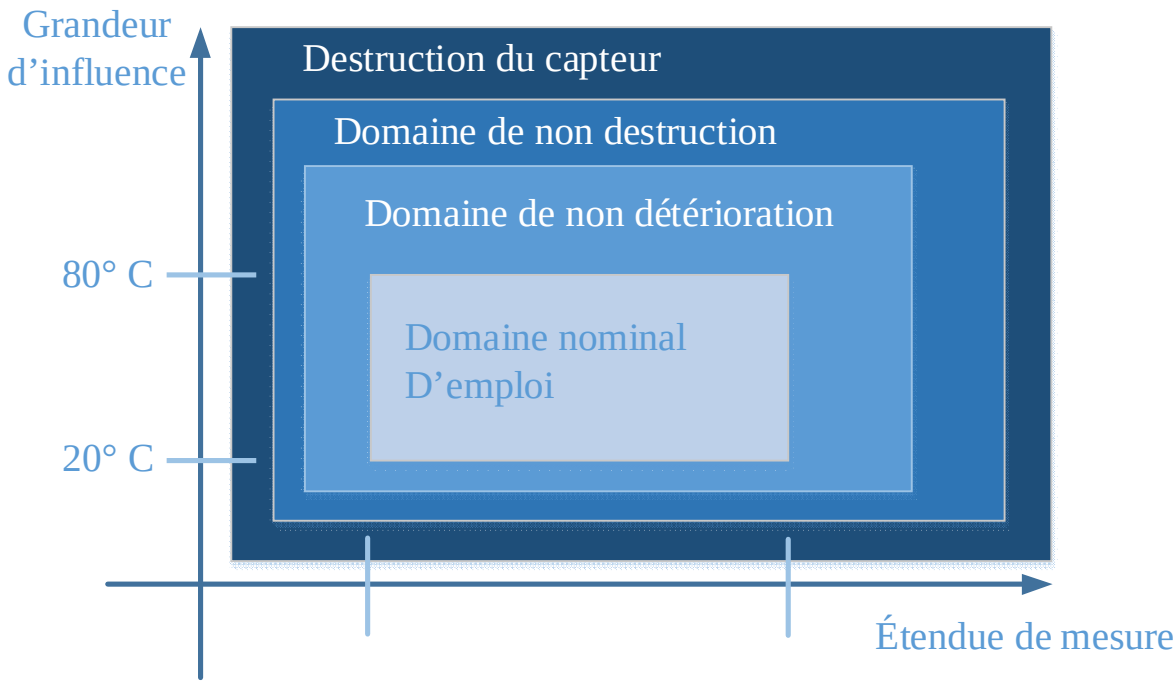


Figure 5 : limites d'utilisation d'un capteur.

Ainsi :

- Le domaine d'emploi nominal correspond aux conditions normales d'utilisation du capteur.
- Lorsque les valeurs du mesurande ou les grandeurs d'influences arrivent dans le domaine de non-détérioration, les caractéristiques métrologiques risquent d'être modifiées mais cette altération est réversible et le capteur pourra retrouver ces caractéristiques normales lorsqu'il retrouvera son domaine nominal d'emploi.
- Si le capteur est utilisé dans le domaine de non-destruction, cette fois-ci les altérations seront irréversibles et seuls un réétalonnage permettront de remesurer dans le domaine nominal d'emploi.
- Enfin si le capteur est utilisé dans le domaine de destruction, celui-ci ne sera plus utilisable et même un étalonnage ne pourra le modifier, la seule solution étant de racheter un capteur. L'étendue de mesure est définie par la différence des valeurs extrêmes de la plage du mesurande dans lequel le fonctionnement du capteur satisfait à

des spécifications données. Le plus souvent l'étendue de mesure correspond au domaine nominal d'emploi.

1.3.2 Sensibilité :

La sensibilité S, pour une valeur donnée de la mesurande, détermine l'évolution de la grandeur de sortie du capteur en fonction de la grandeur d'entrée.

$$S = \frac{d(sortie)}{d(mesurande)}$$

1.3.3 Résolution :

La résolution est la plus petite variation du mesurande que le capteur est capable de déceler. La résolution doit être regardée avec importance lors de la mesure d'une grandeur car celle-ci conditionne la précision du résultat obtenu. En effet si on désire par exemple mesurer une température de l'ordre de 1 °C, le choix d'un capteur d'une résolution de 0,5°, est conséquent d'une mesure imprécise, par contre si on mesure une température d'une dizaine de degré Celsius alors ce capteur sera assez bon.

1.3.4 Finesse :

La finesse permet à l'utilisateur d'estimer l'influence de la présence du capteur sur la valeur du mesurande. Un exemple peut être pris pour la mesure de température en effet pour celle-ci le capteur doit avoir une faible capacité calorifique afin de ne pas perturber le système.

La sensibilité et la finesse sont en général antagonistes et doivent aboutir à un compromis.

1.3.5 Fidélité - Justesse – Précision :

Ces trois termes sont souvent confondus par l'utilisateur soit dans les documents techniques donc il est très important de comprendre la différence entre ces trois termes.

- La fidélité est l'aptitude d'un capteur à délivrer, pour une même valeur de la grandeur mesurée, des mesures répétitives concordantes entre elles. L'erreur de

fidélité correspond à l'écart type obtenu sur une série de mesures correspondant à un mesurande constant.

- La justesse est l'aptitude d'un capteur à délivrer une réponse proche de la valeur vraie et ceci indépendamment de la notion de fidélité. Elle est liée à la valeur moyenne obtenue sur un grand nombre de mesures par rapport à la valeur réelle.
- Et enfin la précision aussi appelé exactitude est définie par l'écart en pourcentage que l'on peut obtenir entre la valeur réelle et la valeur obtenue en sortie du capteur. La précision est souvent donnée en pourcentage de l'étendue de mesure.

Un capteur exact est à la fois juste et fidèle. Afin de mieux visualiser ces trois termes, un schéma reprenant ces trois définitions a été réalisé en prenant la forme d'une cible, le centre étant bien entendu la valeur que l'on cherche à atteindre ici il s'agit de la valeur vraie.

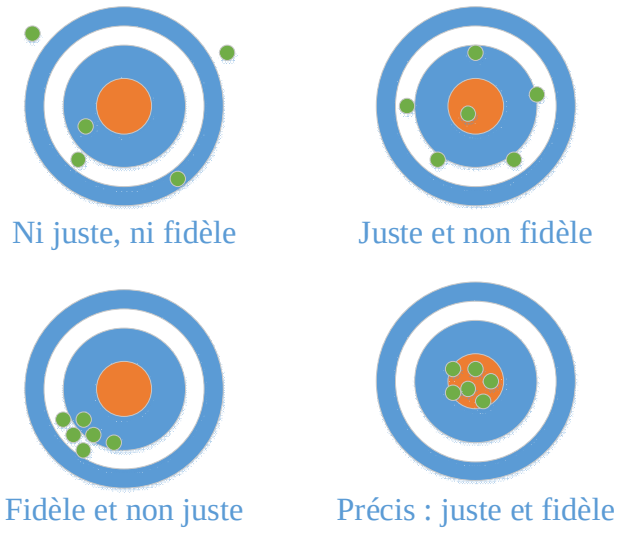


Figure 5 : La précision d'un capteur.

1.3.6 Hystérésis ou réversibilité :

L'hystérésis caractérise l'aptitude du capteur à fournir la même indication lorsqu'on atteint une même valeur du mesurande soit par variation croissante ou décroissante. L'hystérésis est la différence maximale entre les deux grandeurs de sortie obtenues pour un même mesurande. L'hystérésis est souvent donnée en % de l'étendue de mesure.

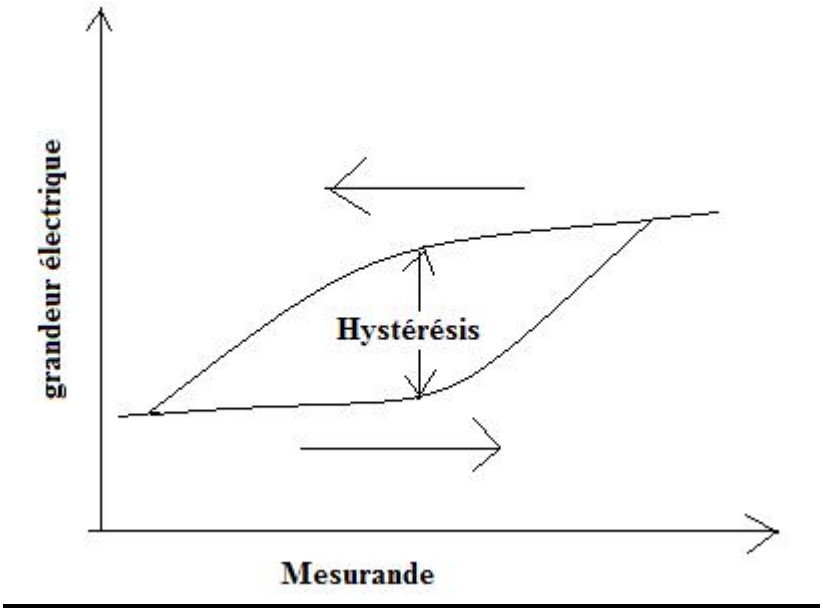


Figure 7 : Hystérésis d'un capteur.

1.3.7 Reproductibilité ou répétabilité :

Ces deux termes sont souvent mélangés par erreur, c'est pour cela que nous allons définir la différence qu'il y a entre la reproductibilité et la répétabilité.

La répétabilité est la mesure d'un même échantillon avec la même méthode dans le même laboratoire avec la même personne et le même équipement alors que la reproductibilité est la mesure du même échantillon avec la même méthode mais dans un laboratoire diffèrent avec des personnes différentes et des équipements différents.

L'écart-type permet de quantifier ces deux notions, dans ces cas-là on parlera d'écart-type de répétabilité ou de reproductibilité. L'écart-type montre la dispersion des valeurs autours de la moyenne, cela permet de savoir si la répétition des mesures n'engendre pas une trop grande dispersion autour de la moyenne. Si l'écart-type est faible, cela signifie qu'exécuter deux mesures donnent des valeurs proches de ce que l'on attend et donc le capteur est dit répétable ou reproductible. La connaissance de ces écart-types permet d'avoir un ordre d'idée sur l'estimation des incertitudes. Bien sûr seul une analyse à partir des documents techniques ainsi que des expériences permettent d'avoir une incertitude globale la plus fidèle possible à la réalité.

1.3.8 Temps de réponse :

Le temps de réponse d'un capteur est l'intervalle de temps qui s'écoule après une variation brusque du mesurande jusqu'à ce que la variation de la sortie du capteur ne diffère plus que d'un écart supérieur à une limite ε conventionnellement fixée. Celui-ci est toujours fournit avec l'écart ε auquel il correspond.

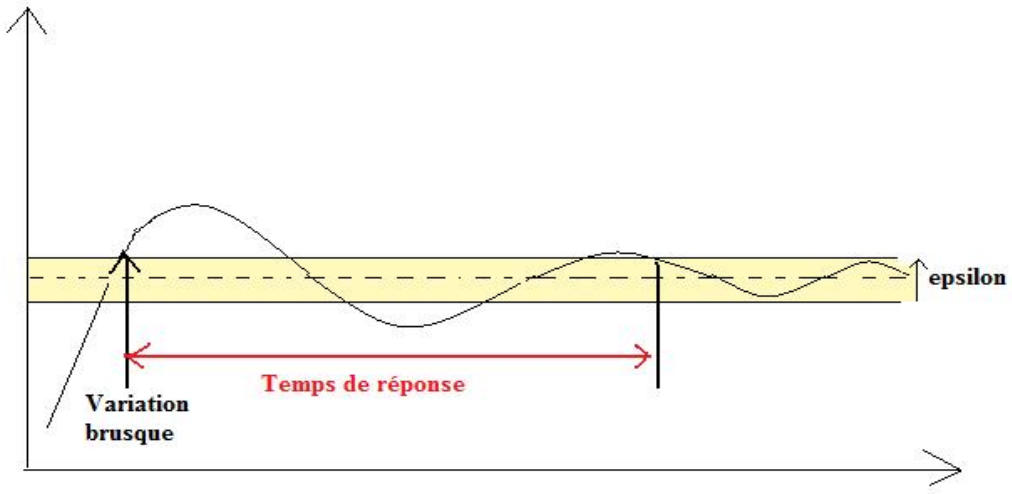


Figure 8 : Temps de réponse d'un capteur.

1.4 Erreurs et incertitudes :

Pour bien choisir un capteur, il est important de connaître ses caractéristiques métrologiques. En effet les caractéristiques métrologiques permettent de savoir quel capteur utiliser, dans quelle gamme, avec quelle précision le résultat nous sera donné et beaucoup d'autres précisions qui peuvent s'avérer très utile dans le choix du capteur. Mais avant de conclure le sujet des capteurs, il est bon de voir les erreurs qui apparaissent lors de la mesure.

1.4.1 Les erreurs de mesures :

Les grandeurs étalons sont les seuls mesurandes dont on puisse connaître la valeur vraie contrairement aux autres mesurandes qui ne peuvent être connu qu'après traitement par une chaîne de mesure. La valeur vraie du mesurande détermine l'excitation du capteur mais l'utilisateur n'y a pas accès vu qu'il n'a accès qu'à la réponse de la chaîne de mesure appelé

valeur mesurée. L'écart entre la valeur vraie et la valeur mesurée s'appelle l'erreur de mesure. La valeur vraie du mesurande ne pouvant être connu, l'erreur de mesure ne peut être qu'estimé. Il existe plusieurs sortes d'erreurs que nous allons essayer de détailler ci-dessous.

1.4.2 Les erreurs systématiques :

Pour une valeur donnée du mesurande, une erreur systématique est soit constante soit à variation lente par rapport à la durée de mesure. Il y a donc un décalage entre la valeur vraie et la valeur mesurée. Cette erreur peut être facilement corrigée en fonction du type d'erreur systématique. Pour une erreur sur la valeur d'une grandeur de référence comme par exemple le décalage du zéro, l'erreur peut facilement être réparée en vérifiant soigneusement les appareils associés ou en étalonnant l'appareil. Une erreur sur les caractéristiques du capteur telle qu'une erreur sur la sensibilité peut facilement être réparée en réétalonnant le capteur. Une erreur due aux conditions d'emploi comme une erreur de rapidité peut être diminuée en attendant que le système soit stabilisé avant d'effectuer la mesure.

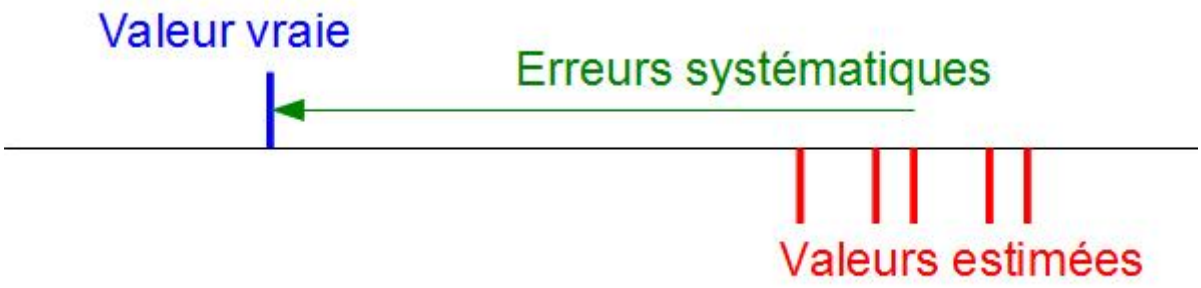


Figure 9 : Erreurs systématiques

1.4.3 Erreurs aléatoires :

L'erreur aléatoire peut se situer de part et d'autre de la valeur vraie. Plusieurs types d'erreurs aléatoires peuvent être définies comme l'erreur de mobilité, l'erreur de lecture, l'erreur due à l'hystérésis, l'erreur de quantification d'un convertisseur analogique-numérique, les bruit de fond, les dérives, etc. Bien que celle-ci soit difficile à déterminer, des solutions existent pour diminuer ces erreurs comme effectuer cette mesure un grand nombre de fois afin que la valeur moyenne se centre autour de la valeur vraie, ou en empêchant les dérives en protégeant la chaîne de mesure de l'environnement extérieur.

Annexe B

HC-SR04 Ultrasonic Range Finder

Manual

Features

- Distance measurement range: 2cm - 400cm
- Accuracy: 0.3cm
- Detect angle: 15 degree
- Single +5V DC operation
- Current comsuption: 15mA

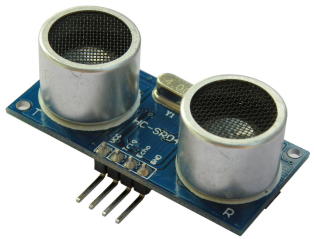


Fig. 1

How It Works

HC-SR04 consists of ultrasonic transmitter, receiver, and control circuits. When trigged it sends out a series of 40KHz ultrasonic pulses and receives echo from an object. The distance between the unit and the object is calculated by measuring the traveling time of sound and output it as the width of a TTL pulse.

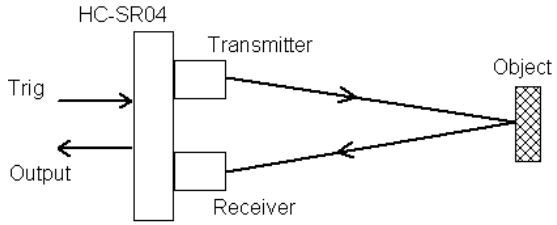


Fig. 2

How To Use It

To measure distance you need to generate a trig signal and drive it to the Trig Input pin. The trig signal leve must meet TTL level requirements (i.e. High level > 2.4V, low level < 0.8V) and its width must be greater than 10us. At the same time you need to monitor the Output pin by measuring the pulse width of output signal. The detected distance can be calculated by the formula below.

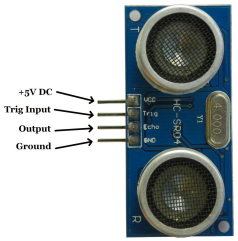


Fig.3

$$\text{Distance} = \frac{\text{Pulse Width} * \text{Sound Speed}}{2}$$

where the pulse width is in unit of second and sound speed is in unit of meter/second. Normally sound speed is 340m/s under room temperature.

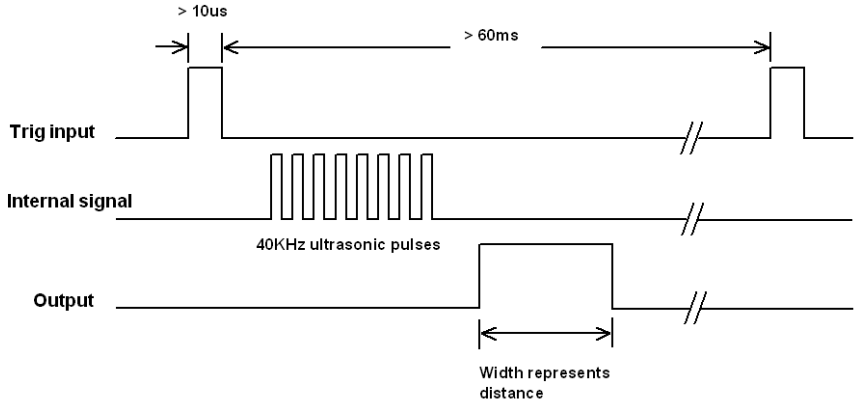
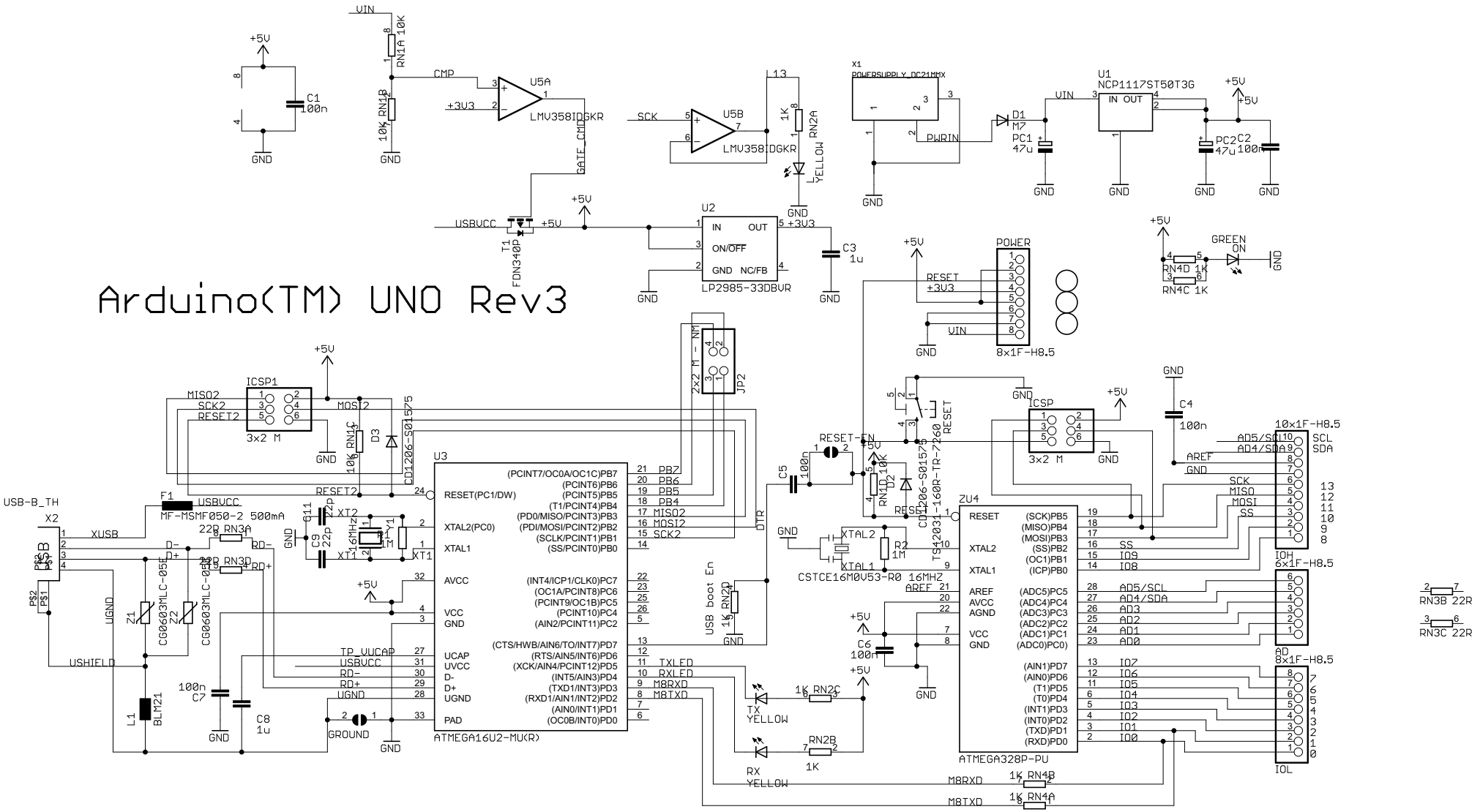


Fig. 4

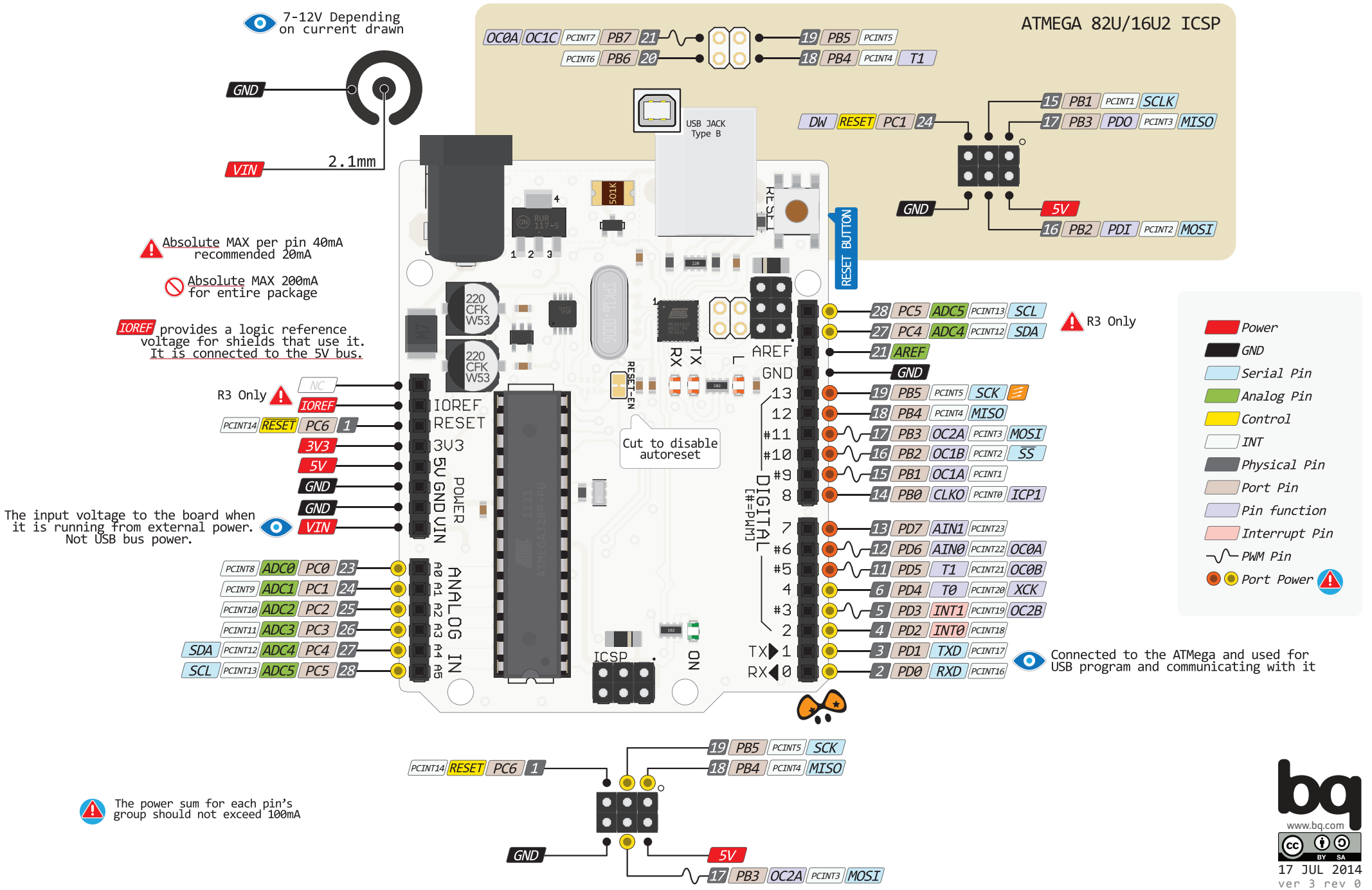
Notes:
1. The width of trig signal must be greater than 10us
2. The repeat interval of trig signal should be greater than 60ms to avoid interference between connective measurements.

Specifications

Parameters	Specification
Operating Voltage	+5V DC
Operating Current	15mA
Operating Frequency	40KHz
Maximum Distance	400cm
Minimum Distance	2cm
Detect Angle	15 degree
Resolution	0.3cm
Input Trig Signal	>10us TTL pulse
Output Signal	TTL pulse with width representing distance
Weight	
Dimension	45 x 20 x 15 mm



UNO PINOUT



www.bq.com
17 JUL 2014
ver 3 rev 0