

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de La Recherche Scientifique
Université Mouloud MAMMERY de Tizi-Ouzou
Faculté de Génie électrique et Informatique
Département Electronique



Mémoire

De fin d'Etudes
En vue de l'obtention du diplôme de Master professionnel
Option : Electronique industrielle

Thème

*Réalisation d'un système d'acquisition
d'images à base d'Arduino*

Réalisé par :

M. BOUDARENE Aguellidh

Proposé par :

M. LAZRI

Promotion : 2017/2018

Remerciements

Avant de commencer la présentation de ce travail, Je profite de l'occasion pour remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire de fin d'études.

***Je* tiens à exprimer mes vifs remerciements pour mon grand et respectueux professeur ; **M.LAZRI**, d'avoir accepté de m'encadrer pour mon mémoire de fin d'études, ainsi pour son soutien, ses remarques pertinentes et son encouragement.**

***je* tiens à lui exprimer ma profonde reconnaissance et toutes mes pensées de gratitude, il ma accompagné de près durant tout ce travail, pour sa disponibilité, pour la confiance qu'il a su m'accorder et les conseils précieux qu'il ma prodigué tout au long de la réalisation de ce mémoire.**

***je* tiens aussi à remercier Mrs. Le président et membres du jury de m'avoir honoré en acceptant de juger mon modeste travail. Veuillez trouver ici le témoignage de mon respect le plus profond.**

mes remerciements vont aussi à tous mes professeurs, enseignants et toutes les personnes qui m'ont soutenu jusqu'au bout, et qui n'ont cessé de nous donner des conseils très importants en signe de reconnaissance.

Dédicaces

Je dédie ce modeste travail à :

A mes parents .Aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler, je leur serai à jamais reconnaissant. Que dieu leur procure bonne santé et longue vie.

Ma mère qui m'a entouré d'amour, d'affection et qui fait tout pour ma réussite, que dieu la garde ;

Mon père qui m'a aidé à devenir ce que je suis aujourd'hui, que dieu le garde et le protège ;

Mon **frère et ma sœur**, et tout les membres de ma famille qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés,

Mes biens chers amis: **Lamine Azzoug , Amghidh Djera , Anis Aichaoui, Mourad belmadi, ghiles cherbi, yacine tehraoui** pour leur soutien et leurs conseils qui m'ont étaient très utiles au long de mon travail.

A mon très cher ami et frère **Smail Hadji** et sa famille qui m'ont accueillis sous leur toit, qui ont étaient ma deuxième famille et aidé à élaborer mon travail jusqu'a la fin .

Mon très cher cousin **Berdous Tarik** et son épouse **Rosa** , qui m'ont soutenu et pour leurs encouragement , malgré leurs absence

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible,

Je vous dis merci

BOUDARENE Aguellidh

Sommaire

| | |
|--|----------|
| Introduction générale..... | 1 |
| <u>Chapitre I : introduction sur les systèmes a microcontrôleur (Arduino)</u> | |
| Préambule..... | 3 |
| I.1 Définition du module Arduino..... | 3 |
| I.2 Historique..... | 3 |
| I.3 Gamme de la carte arduino..... | 4 |
| I.4 Description de la carte Arduino Uno..... | 9 |
| I.4.1 Le microcontrôleur ATmega328..... | 10 |
| I.4.2 Les principales caractéristiques de l'ATmega328..... | 11 |
| I.5 Alimentation de la carte Arduino UNO..... | 13 |
| I.6 Protection du port USB contre la surcharge en intensité..... | 14 |
| I.7 Les entrées/ sorties numériques..... | 15 |
| I.8 Les broches analogiques..... | 16 |
| I.9 Arduino et la communication avec l'extérieur..... | 16 |
| I.10 Gestion des mémoires dans la carte Arduino..... | 17 |
| I.11 L'environnement de programmation Arduino..... | 17 |
| I.12 Description du logiciel Arduino..... | 18 |
| I.13 Description de la barre des boutons..... | 19 |
| I.14 Description de la barre des menus..... | 21 |
| I-15 Les étapes de télé versement d'un programme vers la carte Arduino..... | 21 |
| Discussion..... | 22 |
| <u>Chapitre II : Description du banc didactique de pression PCT-M3</u> | |
| Préambule..... | 23 |
| II.1 Définition de l'image..... | 23 |
| II.2 Types d'images..... | 23 |
| II.3 Codages des couleur..... | 24 |

| | |
|---|----|
| II-3-1 Image noir et blanc | 24 |
| II-3-2 Niveaux de gris | 24 |
| II-3-3 Image couleur | 25 |
| II.4 Caractéristiques d'une image numérique..... | 25 |
| II.5 Acquisition d'une image | 27 |
| II.6 La caméra | 27 |
| II.7 structure de la camera | 28 |
| II.7.1 Objectif de la camera..... | 28 |
| II.7.2 Les paramètres de l'objectif | 28 |
| II.8 Le capteur d'images | 29 |
| II.9 Le capteur CCD | 29 |
| II.10 Réponse spectrale | 30 |
| II.11 Le capteur CMOS | 31 |
| II.12 Réponse spectrale | 32 |
| II.13 Domaine d'utilisation des capteurs | 32 |
| II.14 Taille du capteur | 33 |
| II.15 Ouverture du champ visuel de la camera | 34 |
| II.16 numérisation d'image | 35 |
| II.17 Les filtres anti-repliement..... | 36 |
| II.18 Choix à effectuer lors de la numérisation | 36 |
| Discussion | 37 |

Chapitre III : Conception et réalisation d'une application a base d'Arduino

| | |
|---|----|
| Préambule..... | 38 |
| III.1 Camera ArduCAM 2MP | 38 |
| III.1.1 Caractéristiques | 39 |
| III.1.2 Spécifications techniques | 39 |
| III.1.3 Composition ArduCAM | 40 |

| | |
|---|-----------|
| III.1.4 Fonctionnement | 41 |
| III.1.5 Câblage du matériel de la caméra42 | |
| III.1.5.1 Câblage typique..... | 42 |
| III-1-5-1-1 Câblage d'une seule caméra..... | 42 |
| III-1-5-1-2 Câblage multi caméras | 43 |
| III-1-5-2 Interface I2C | 44 |
| III-1-5-3 Interface périphérique série (SPI)..... | 44 |
| III.1.6 Composition du logiciel de caméra SPI | 45 |
| III.1.7 Guide de démarrage rapide..... | 46 |
| III.1.8 Exemples d'esquisses | 47 |
| III.1.9 Montage de la maquette..... | 48 |
| III.1.10 Mode opératoire..... | 49 |
| III.1.11 Test et résultats obtenus | 49 |
| III.2 Caméra 2MP Arducam avec Module Nano ESP8266 | 49 |
| III.2.1 Composition du kit | 50 |
| III.2.2 Caractéristiques techniques | 50 |
| III.2.3 Constitution | 50 |
| III.2.4 Schéma de câblage | 51 |
| III.2.5 Mise en marche ESP8266 avec IDE Arduino | 52 |
| III.2.5.1 installation du gestionnaire de planche..... | 52 |
| III.2.5.2 Utilisation l'IDE Arduino..... | 52 |
| III-2-6 Exemple | 54 |
| Discussion | 57 |
| Conclusion générale | 58 |

Bibliographie

Liste des figures :

| | |
|---------------|---|
| Figure I-1 | Une des première carte wiring |
| Figure I-2 | description de la carte Arduino UNO |
| Figure I-3 | Le microcontrôleur ATmega328 |
| Figure I-4 | Architecture interne de l'ATmega328 |
| Figure I-5 | Brochage de la carte ARDUINO UNO |
| kFigure I-6 | Fenêtre principale de l'environnement de programmation ARDUINO |
| Figure I-7 | Les différentes parties de la fenêtre principale du logiciel Arduino |
| Figure I-8 | Fenêtre associée au moniteur série |
| Figure I-9 | Structure de la barre des boutons |
| Figure I-10 | La barre des menus |
| Figure II-1 | image vectorielle et image bitmap |
| Figure II-2 | l'objectif de la camera |
| Figure II-3 | Constitution d'un photo-site |
| Figure II-4 | fonctionnement général d'une caméra à capteur CCD |
| Figure II-5 | réponse spectrale d'un capteur CCD |
| Figure II-6 | fonctionnement général d'une caméra à capteur CCD |
| Figure II-7 | réponse spectrale d'un capteur CCD |
| Figure II-8 | Illustration de dimension de taille de capteur |
| Figure II-9 | Champ visuel de caméra |
| Figure II-10 | Schéma permettant le calcul du champ de vue |
| Figure II-11 | signal analogique et signal numérique |
| Figure III-1 | cameraArduCAM mini |
| Figure III-2 | câblage typique |
| Figure III-3 | câblage pour Arduino UNO R3 |
| Figure III-4 | câblage multi-cameras |
| Figure III-5 | câblage multi-cameras sur Arduino UNO R3 |
| Figure III-6 | Liaison SPI (un maître et un esclave) |
| Figure III-7 | structure des bibliothèques ArduCAM |
| Figure III-8 | Exemple Arduino IDE. |
| Figure III-9 | ArduCAM-LF |
| Figure III-10 | application ArduCAM hostV2.0 |
| Figure III-11 | Exemple de structure de dossier |
| Figure III-12 | montage générale de la maquette |
| Figure III-13 | capture d'image |
| Figure III-14 | ArduCAM-Mini-2MP-V2 et ESP8266-Nano |
| Figure III-15 | schéma de câblage |
| Figure III-16 | Pack d'addonArduCAM ESP8266 UNO |
| Figure III-17 | sélection de planche |
| Figure III-18 | sélection d'exemple |
| Figure III-19 | configuration de la camera |
| Figure III-20 | exemple camera wifi |
| Figure III-21 | identification de l'adresse IP |
| Figure III-22 | exemple page HTML |

Liste des tableaux :

| | |
|---------------|--|
| Tableau II-1 | Dimensions des capteurs CCD |
| Tableau II-2 | Distance d'un objet en fonction de la focale |
| Tableau III-1 | définition des pin de l'ArduCAM |
| Tableau III-2 | ArduCAM-ESP8266-Nano Pin Définition |

Introduction:

Depuis quelques années, la technologie évolue rapidement dans le domaine de l'acquisition d'image que ce soit dans l'industrie, la science ou le grand public. A cet effet, des systèmes d'acquisition d'images sont de plus en plus utilisés pour collecter de l'information. En général, ce sont systèmes embarqués dans des applications comme des drones, domotique, robotique etc. L'élément de base de ces systèmes est le microcontrôleur.

Un microcontrôleur est en fait un système à microprocesseur mais qui rassemble dans un seul circuit intégré. C'est un composant complètement autonome qui, dès qu'il est alimenté va exécuter le programme présent dans sa mémoire morte. L'Arduino peut être simplement vu comme un microcontrôleur avec des entrées et sorties, il est conçu avec un environnement de développement. Le produit prend donc la forme d'un circuit imprimé sur lequel un microcontrôleur a été intégré et auquel a été ajouté tout un environnement et écosystème permettant de développer simplement et facilement des applications. On va précisément utiliser Arduino pour développer des objets interactifs et (, des objets) connectés ayant une couche de communication.

Ce travail n'a pas pour objectif de traiter tous les thèmes relatifs à cette évolution. Le but recherché est d'apporter quelques éclaircissements sur l'interfaçage des caméras avec les systèmes informatiques.

Depuis 1980, année de la présentation de la première caméra vidéo à base de capteur CCD par Sony, les évolutions vers le numérique n'ont plus cessé. Ces caméras à capteur CCD n'étaient pas totalement numériques. Le signal de sortie de celles-ci était plutôt analogique pour qu'il puisse être compatible avec les systèmes d'affichage et les supports d'enregistrement.

Les capteurs CCD ont longtemps été privilégiés pour la qualité de leur photos . Le capteur CMOS était alors considéré comme un capteur un peu bas de gamme. Mais le développement des appareils photos pour Smartphones a rendu le CMOS très populaire en privilégiant ce capteur qui était moins cher à produire et moins consommateur d'énergie. Aujourd'hui, le offre des résultats très satisfaisants côté qualité et on peut même y intégrer des circuits de traitement de l'image.

Si le capteur CCD reste la référence pour de nombreux appareils photo et caméras, le CMOS a su faire oublier ses défauts et devenir suffisamment bon pour que certains constructeurs d'appareils photo optent à présent pour lui.

Dans ce mémoire, nous allons nous intéresser à un système d'acquisition d'image en utilisant un type de camera ov 2640

Notre travail est organisé comme suit :

- Le premier chapitre est consacré au système a microcontrôleur (arduino).
- Le deuxième chapitre est consacré aux généralités sur les systèmes d'acquisition d'images
- Le troisième chapitre on a fait une conception et réalisation d'un système d'acquisition d'image en utilisant la carte aduino uno et deux camera de type ov 2640

En dernier nous allons finir avec une conclusion générale

I-Introduction sur les systèmes a microcontrôleur(Arduino)

Préambule :

Aujourd'hui, l'électronique est de plus en plus remplacée par l'électronique programmée, pour la réalisation de mon système d'acquisition d'image, j'ai utilisé une carte d'acquisition de type arduino afin d'assurer la communication entre les deux parties a savoir son but est de simplifier les schéma électronique et par conséquent réduire l'utilisation de composant électroniques, réduisant ainsi le cout de fabrication d'un produit, il en résulte des systèmes plus complexes e performants pour un espace réduit.

Dans ce travail, je présente la description de la carte Arduino qui sera utilisée pour mon application. En effet, les différentes parties constituant cette carte seront détaillées, a savoir un microcontrôleur, des ports d'entrée/sorties, une mémoire de type EPROM. Aussi, le logiciel IDE utilisé pour la programmation sera présenté dans ce chapitre

I-1 Définition du module arduino :

La carte arduino est un circuit imprimé spécifiquement conçu pour héberger un microcontrôleur et donner accès a toutes ses entrées et sorties. Elle comprend aussi quelques autres composants électroniques qui permettent de faire fonctionner le microcontrôleur .

Un microcontrôleur est un petit ordinateur confiné dans un unique circuit intégré (une puce) . Il constitue un excellent moyen pour programmer et pour contrôler des équipements électroniques de manière à effectuer des taches très divers.

Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée, le modélisme, la domotique mais aussi dans les domaines différents comme l'art contemporain et le pilotage d'un robot ,commander des appareils mobiles , chaque module d'arduino possède un régulateur de tension +5V et un oscillateur a quartz 16 MHz (ou un résonateur céramique dans certains modelés)

Le logiciel qui permet de programmer la carte arduino porte le nom d'IDE (Environnement de Développement Intégré) , cette application intègre l'édition des programmes

I-2 Historique :

En 2001, Casey Reas et Benjamin Fry ont commencé un projet nommé Processing. L'objectif était de permettre à des non-programmeurs de se mettre à la programmation en permettant facilement et rapidement de produire des représentations à l'écran. Le projet permettrait à l'utilisateur de disposer d'un carnet de croquis numériques (des programmes) pour tester des idées au prix d'un investissement temporel minimal. S'appuyant sur les mêmes bases que Processing, Hernando Barragan a commencé à développer une carte à microcontrôleur en 2004, qu'il a nommée Wiring. Cette carte était le prédécesseur d'Arduino.



1.

Figure I-1 : Une des premières carte wiring.

C'est ainsi qu'en 2005, le projet Arduino a été lancé pour fournir un matériel bon marché et facile à utiliser à destination des étudiants qui se plaignent de ne pas avoir accès à des solutions bas prix pour accomplir leurs projets de robotique. Massimo Banzi et David Cuartielles, un ingénieur espagnol spécialisé sur les microcontrôleurs, en embarquant dans leur histoire un des étudiants de Banzi, David Mellis, qui sera chargé de créer le langage de programmation.

Avec la carte, en deux jours David Mellis écrivit le code ! Trois jours de plus la carte était créée, ils décidèrent de l'appeler Arduino d'après Arduin d'Ivera, un roi italien. Ça devient un succès tout de suite auprès des étudiants, tout le monde arrive à en faire quelque chose très rapidement sans même avoir de connaissances particulières ni en électronique ni en informatique, réponse à des capteurs, faire clignoter des LEDs, contrôler des moteurs Ils publient les schémas, investissent 3000 euros pour créer les premiers lots de carte.

Les 50 premières partent directement a des élèves de l'école en 2006 5000 cartes vendues.. En 2007 plus de 30 000 En 2011 →120000

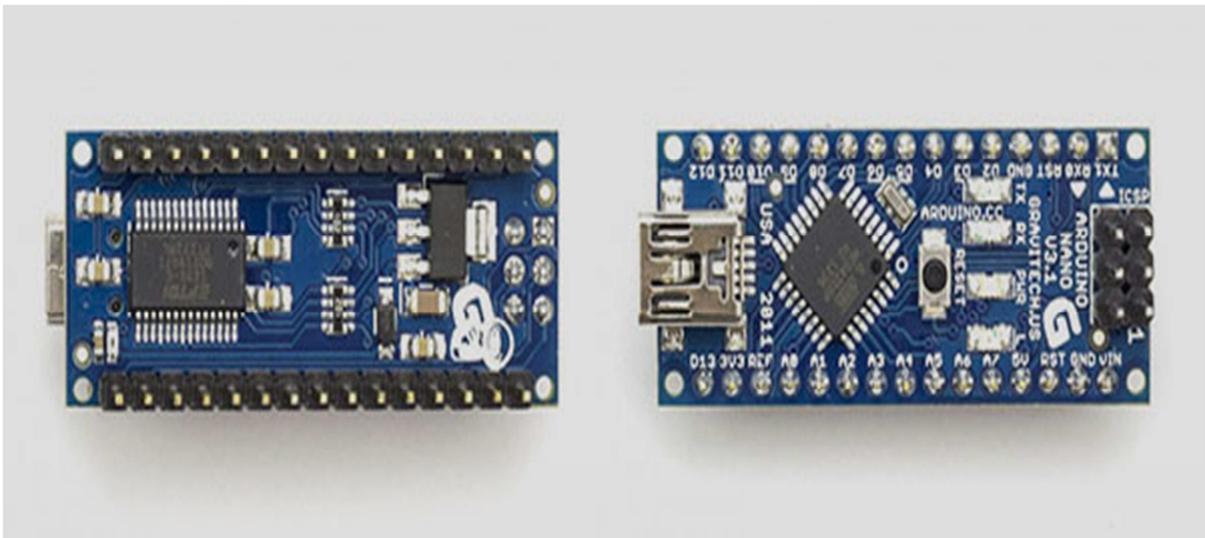
I-3 Gamme de la carte arduino :

Actuellement, il existe plus de 20 versions de module Arduino, je cite quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique :

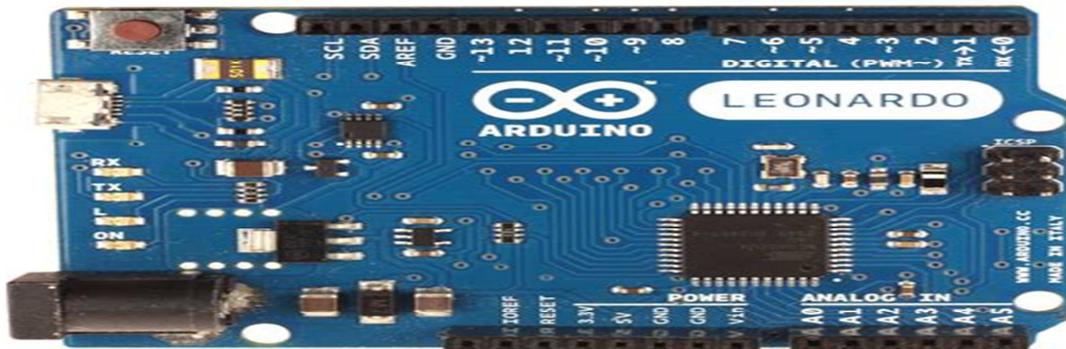
- **Arduino Uno** : c'est la carte idéal pour découvrir l'environnement Aduino , elle permet a tout débutant de se lancer dans ses premiers petits projets.



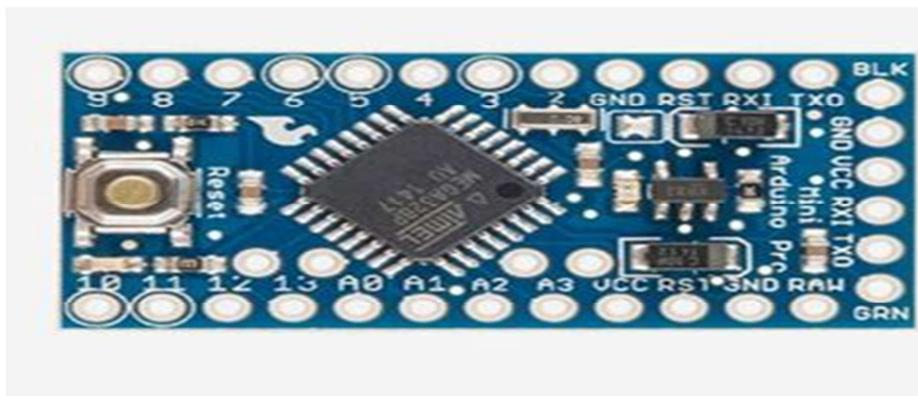
- **Arduino Nano** : La carte Arduino nano n'est ni plus ni moins qu'une carte Arduino uno miniaturisée. Sa taille et son poids réduits la destinent à une utilisation dans des espaces réduits ou dans des applications de robotique ou de modélisme pour lesquels le poids et la taille sont des facteurs déterminant (hélicoptères, drones...)



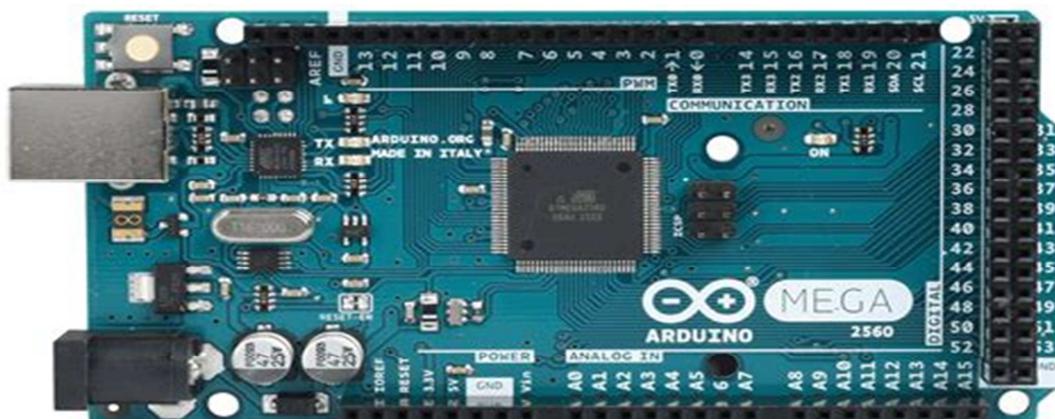
- **Arduino Leonardo** : C'est la carte qui est prévue pour succéder à la carte Arduino Uno en présentant des caractéristiques équivalentes mais une ergonomie revue et une stabilité plus éprouvée.



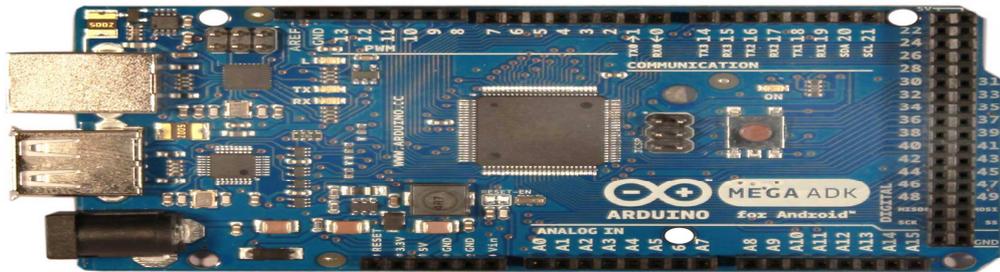
- **Arduino Mini pro** : La carte arduino Mini Pro est une carte Arduino Uno simplifiée à l'extrême permettant néanmoins de piloter de petits projets ou certains éléments d'un projet. cette carte n'intègre pas de port USB ce qui rends sa connectivité délicate.



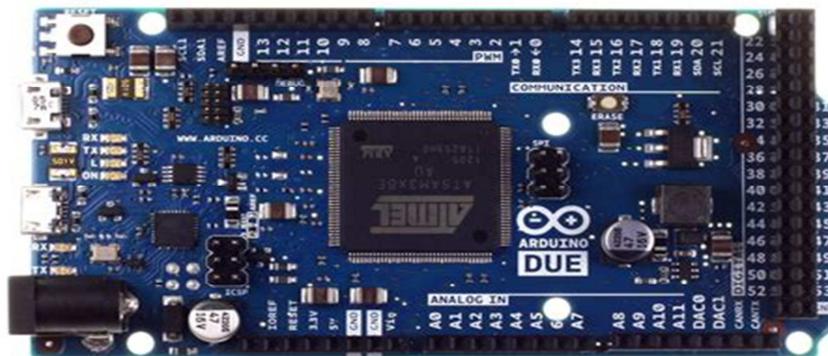
- **Arduino Mega** : La carte Arduino Mega est la carte la plus diffusée après la carte Arduino Uno. Elle offre un nombre d'entrées/sorties beaucoup plus important , un processeur plus puissant doté d'une mémoire plus vaste qui permet d'exploiter des algorithmes plus complexes .



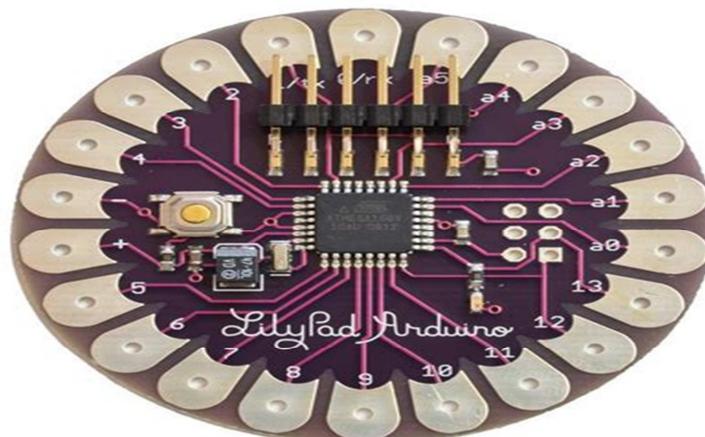
- **Arduino mega ADK** : offre les mêmes caractéristiques techniques que la carte Arduino mega mais son port USB permet de la connecter avec un environnement Androïde ouvrant de nouvelles perspectives d'interaction avec le monde des Smartphones et des capteurs dont ils sont dotés. Sa mise en œuvre nécessite par contre de solides connaissances en Java et la capacité à développer ses propres applications.



- **Arduino Due** : La carte Arduino Due est une évolution de la carte Arduino Mega et offre des performances réputées 3 fois supérieures. Elle permet de manipuler rapidement des algorithmes lourds particulièrement utiles dans le monde de la robotique par exemple.

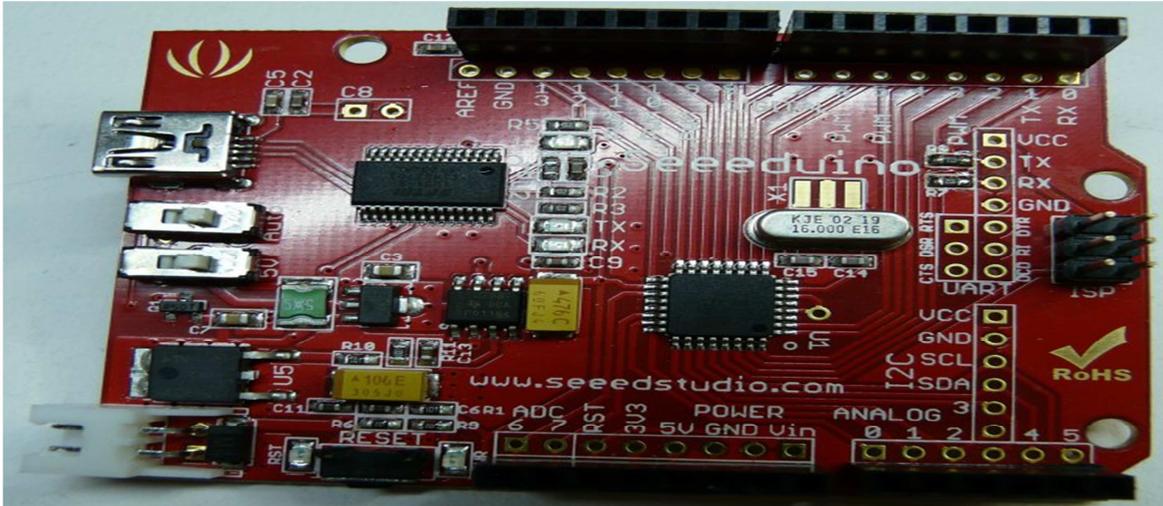


- **Arduino Lily Pad** : La plateforme arduino LilyPad est destinée aux plus créatifs qui veulent coudre des circuits électroniques sur leurs vêtements

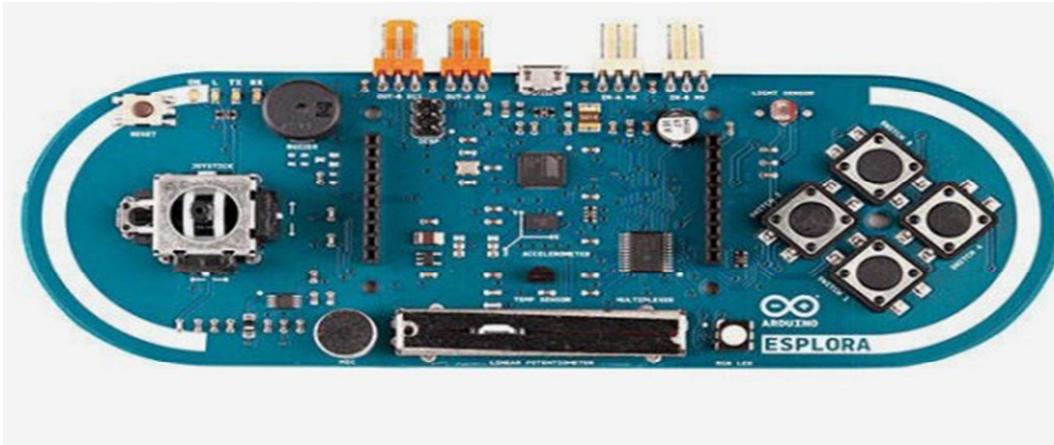


- **Arduino Duemilanove (seeeduno)** : La seeeduno est une carte arduino 2009 (arduino Duemilanove)

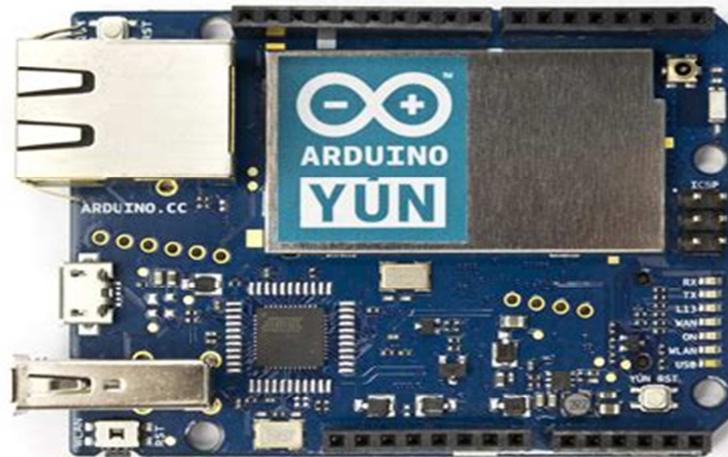
En gros Seeedstudio a regardé tout ce qui allait pas dans l'arduino 2009 et y a apporté sa petite touche de « seed ».



- **Arduino Esplora** : ressemblant a un contrôleur visuel de jeu, avec un manche et des sondes intégrés pour le bruit, la lumière, la température, et l'accélération



- **Arduino Yun** : La carte Arduino Yun, récemment proposée par Arduino, Elle est un dérivé de la carte Leonardo et a pour objectif de combiner la puissance de Linux avec la facilité d'utilisation d'une carte Arduino. Elle est également la première carte Arduino à être dotée nativement d'un wifi intégré.



I-4 Description de la carte Arduino UNO :

Modèle UNO de la société ARDUINO est une carte électronique dont le noyau 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, un connecteur céramique (quartz) de 16 MHz, connecteur **ICSP (In Circuit Serial Programming)** qui permet d'injecter le bootloader a l'intérieur du microcontrôleur, un résonateur jack pour une alimentation extérieur, un bouton de reset pour mettre le processus a zéro.

La carte Arduino UNO diffère de toute les cartes précédentes car elle n'utilise pas le circuit intégré **FTDI USB-vers-SERIE**, Elle a juste un petit microcontrôleur **ATmega8U2** programmé comme convertisseur USB-vers-SERIE.

"UNO" signifie un en Italien et ce nom marque la venue prochaine de la version 1.0 du logiciel Arduino. La carte UNO et la version 1.0 du logiciel seront la référence des versions Arduino a venir. La carte UNO est la dernière d'une série de la carte USB Arduino, et le modèle de référence des plateformes Arduino, pour une comparaison avec les versions précédentes .

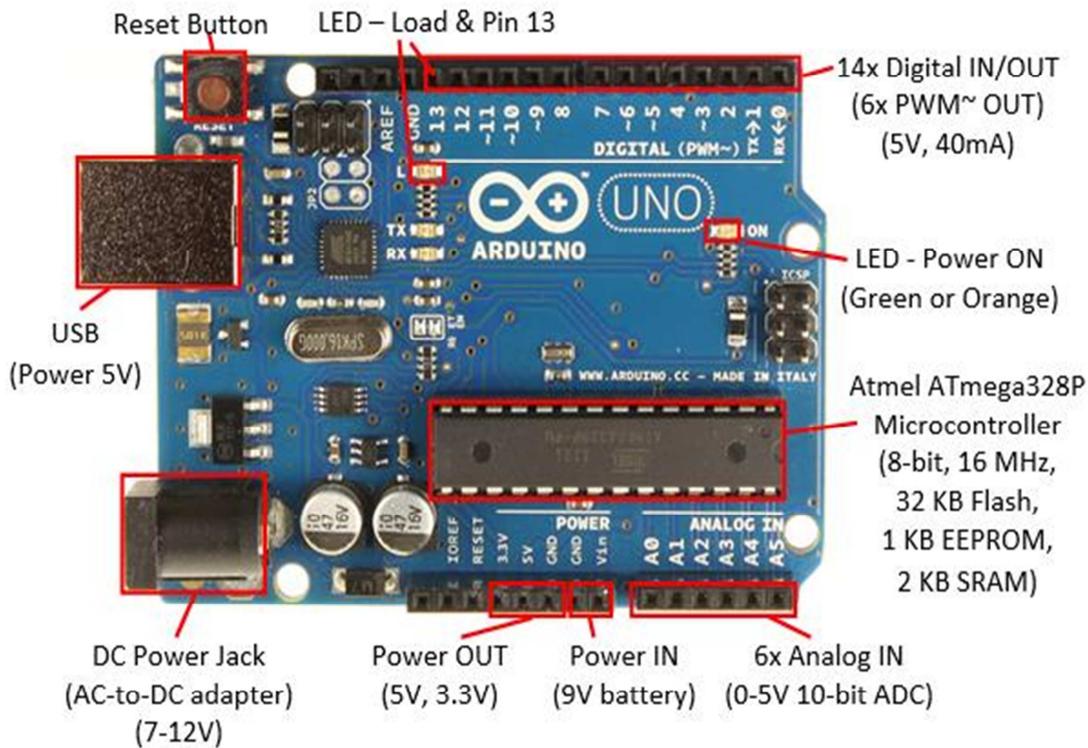


Figure I-2: description de la carte Arduino UNO

I-4-1 Le microcontrôleur ATmega328 :

Le microcontrôleur utilise sur la carte arduino UNO est un microcontrôleur **ATmega 328** c'est un microcontrôleur ATMEL de la famille AVR 8bits

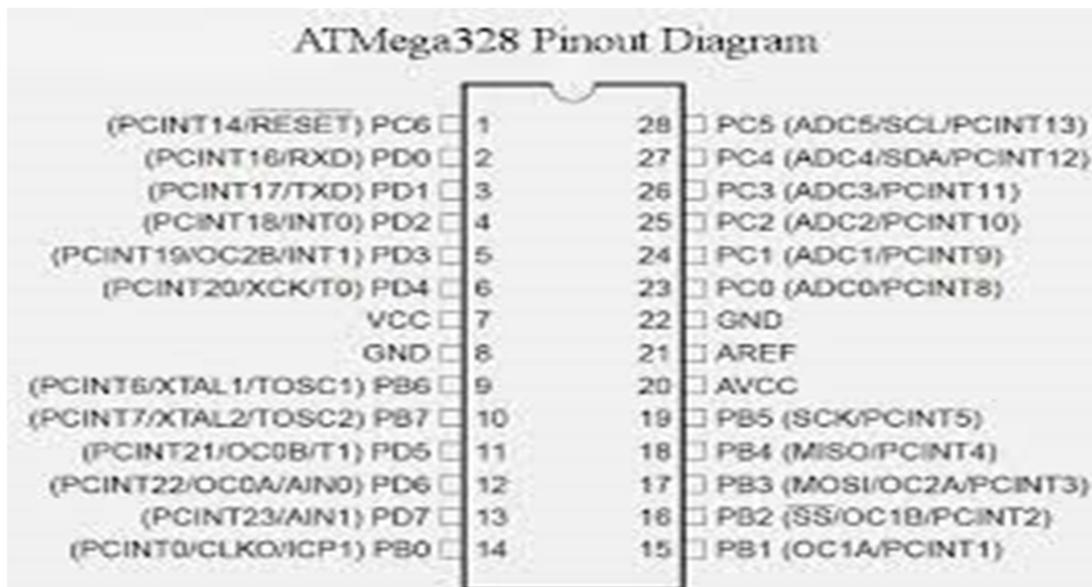


Figure I-3 :Le microcontrôleur ATmega328

I-4-2 Les principales caractéristiques de l'ATMega328 :

le microcontrôleur ATMega328 de Atmel dispose :

- De 14 broches numérique d'entrées/sorties, dont 6 peut être utilisées en sorties PWM (largeur d'impulsion modulée), réparties selon l'ordre suivant 0C0A(PD6), 0C0B(PD5), 0C1A(PB1), 0C1B(PB3), 0C2A(PB3), 0C2B(PD3) et 2 (0 et 1) pour réception/émission série.
- De 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques). Ces entrées/sorties sont réparties sur trois ports : port B, port D port C (soit 23 broches E/S en tout).
- D'un courant max par broches E/S= 40mA.
- D'un courant max sur sortie 3.3V=50mA.
- D'une mémoire Flash de 32KB dont 512 utilisée par le boot loader .
- D'une mémoire SERAM de 2 KB.
- D'une mémoire EEPROM de 1 KB.
- Il contient aussi trois compteurs (timer0, timer1, timer2), le timer0 et le timer2 sont à comptage 8 bits, le timer1 est à comptage 16bits. Les timers peuvent être utilisés pour générer les signaux PWM.

Certaines broches peuvent avoir plusieurs fonctions de ces programmations

PWM : Le ATMega328 a 6 broches qui peuvent servir a cette fonction qui sont les broches 0C0A(PD6), 0C0B(PD5), 0C1A(PB1), 0C1B(PB3), 0C2A(PB3), 0C2B(PD3).

Convertisseur Analogique/Numérique : le ATMega328 possède un convertisseur Analogique/Numérique d'une résolution de 10 bits, ce convertisseur peut être utilisé à travers 6 entrées multiplexées de ADC0(PC0) jusqu'à ADC5(PC5).

Gestion du bus I2C : ce bus est exploité via les deux broches SDA(PC5)/SCL(PC4).

Port série (USART) : émission/réception série via les broches TXD(PD1)/RXD(PD0).

Comparateur Analogique : le comparateur analogique intégré dans l'ATMega peut être utilisé a travers les deux broches AIN(PD6) et AIN1(PD7), ce comparateur peut déclencher une interruption.

Watchdog Timer programmable : l'ATMega possède un compteur dit de **chien de garde** programmable pour générer des interruptions a la fin de son comptage et il peut être utilisé comme étant un simple compteur.

Gestion d'interruption (24 sources possibles) : en résumé

- Interruptions liées aux entrées INT(PD2) et INT(PD3).
 - Interruptions sur changement d'état des broches PCINT0 a PCINT23.
 - Interruptions liées au timers 0,1 et 2.
 - Interruption liée au comparateur analogique.
 - Interruption de fin de conversation **ADC**.
 - Interruption du port série **USART**.
 - Interruption du bus TWI **I2C**.

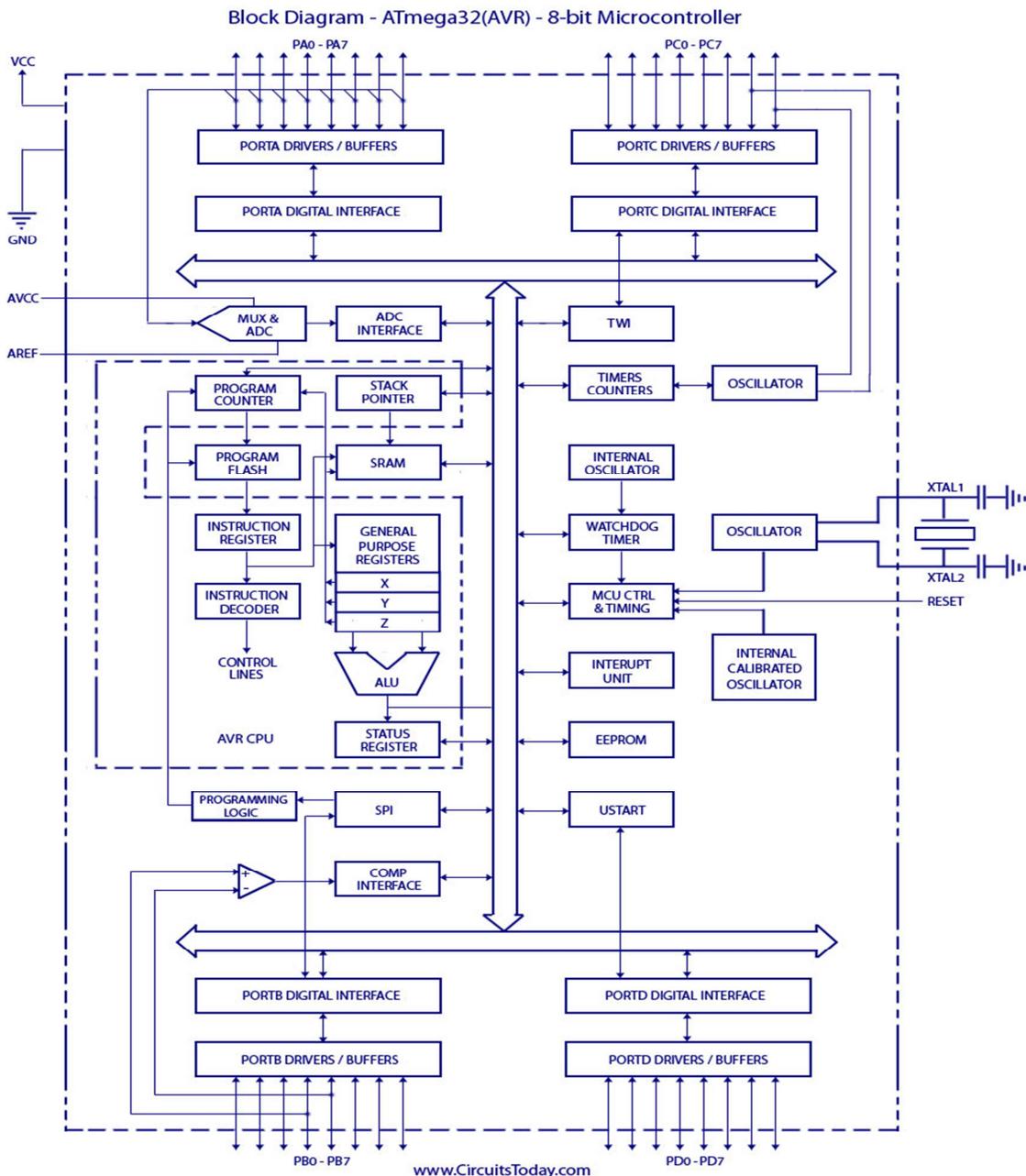


Figure I-4 :Architecture interne de l'ATMega328

I-5 Alimentation de la carte Arduino UNO :

La carte Arduino Uno peut être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500 mA) 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 a 12V sous 500 mA) ou des piles.

L'adaptateur secteur peut être connecté en branchant une prise de 2.1mm dédiée 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 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte 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 (risque d'endommager la carte)

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.
- ❖ **5 V** : la tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite <tension régulée> obtenue à l'aide d'un régulateur intégré dans la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation Vin via le régulateur de la carte ou bien de la connexion USB (qui fournit du 5V régulé) ou de toute autre source d'alimentation régulée.
- ❖ **3V3** : une alimentation de 3.3 V fournie par le régulateur de 3.3 V de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). l'intensité maximale disponible sur cette broche est de 50 mA.
- ❖ **GND** : broche de masse (ou 0V).

I-6 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 500 mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500 mA sont appliquées au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

I-7 Les entrées/ sorties numériques :

Chacun des 14 broches numériques de la carte UNO (numéroté de 0 a 13) peut être utilisée soit comme 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 a l'aide de l'instruction **digitalWrite** (broche ,**HIGH**).

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

- **Communication série** : broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB vers série de la carte (composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur). On fait appel a la transmission série a travers ces broches avec l'instruction **Serial.print()**, a condition que le câble USB soit déconnecté.
- **Interruptions Externes** : broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.
- **Impulsion PWM (largeur d'impulsion modulée)**: Broches 3,5,6,9,10 et 11 Fournissent une impulsion PWM 8-bits a 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 (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP.
- **I2C** : broches 4 (SDA) et 5 (SCL), supportent les communications de protocole I2C, disponible en utilisant la librairie **Wire/I2C**.
- **LED** : broche 13. Il y a LED incluse dans la care connectée a la broche 13 lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

I-8 Les broches 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 fonction **analogRead()** du langage Arduino. Par défaut, ces broches mesurent entre 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction **analogReference()** du langage Arduino.

• **Autre 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()**
- **Reset** : mettre cette broche au niveau BAS entraîne la réinitialisation (le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit.

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

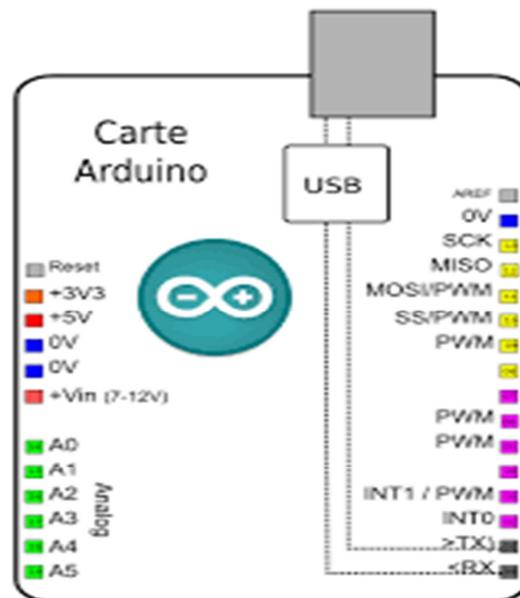


Figure I-5 : Brochage de la carte ARDUINO UNO

I-9 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'ATMega328 dispose d'une UART (Universal Asynchron Receiver Transmitter ou émetteur-récepteur universel asynchrone en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX).

Un circuit intégré ATMega8U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparait comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATMega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

Le logiciel 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 clignote lorsque les données sont transmises via le circuit intégré USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et une librairie série logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO

- L'ATMega328 supporte également ces protocoles I2C et SPI
- Le logiciel Arduino inclut la librairie **Wire** qui simplifie l'utilisation du bus I2C
- Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible, il suffit de la faire inclure dans le programme au niveau du logiciel Arduino lors de la programmation .

I-10 Gestion des mémoires dans la carte Arduino :

L'ATMega328 a 32Ko de mémoire FLASH pour stocker le programme (dont 0.5Ko également utilisés par le bootloader) L'ATMega328 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)

I-11 L'environnement de programmation Arduino :

La carte Arduino présente le noyau de mon système pour que je puisse le programmer je dois disposer du logiciel compatible avec cette carte, ce logiciel est l'**Arduino EDI (Espace de Développement Intégré)** , qui porte le nom même de la carte Arduino .

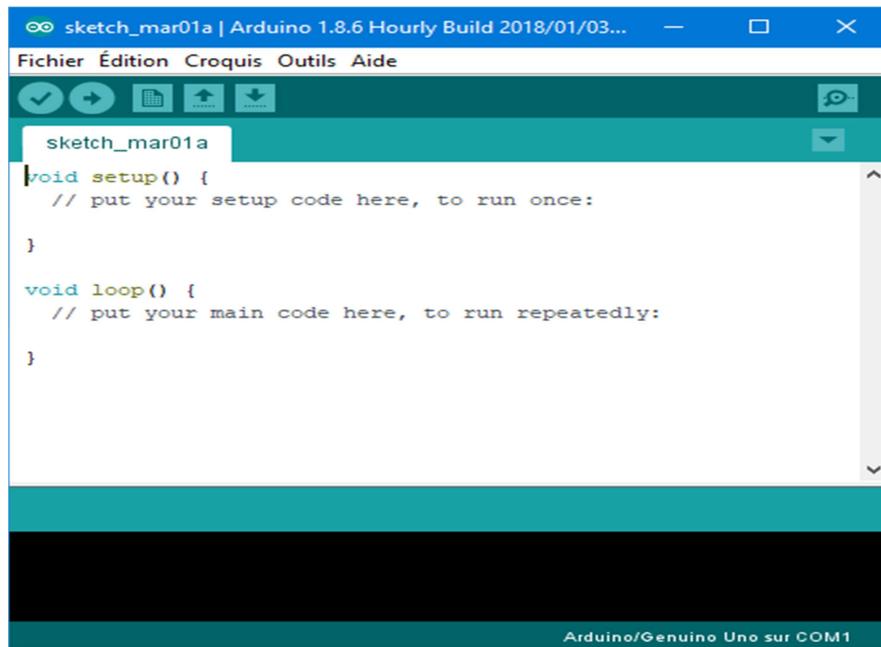


Figure I-6 :Fenêtre principale de l'environnement de programmation ARDUINO

I-12 Description du logiciel Arduino :

Le logiciel Arduino a pour fonctions principales :

- Le pouvoir d'écrire et compiler des programmes pour la carte Arduino .
- De se connecter avec la carte Arduino pour y transférer les programmes.
- De communiquer avec la carte Arduino.

Cet espace de développement intégré (EDI) dédié au langage Arduino et a la programmation des cartes Arduino.

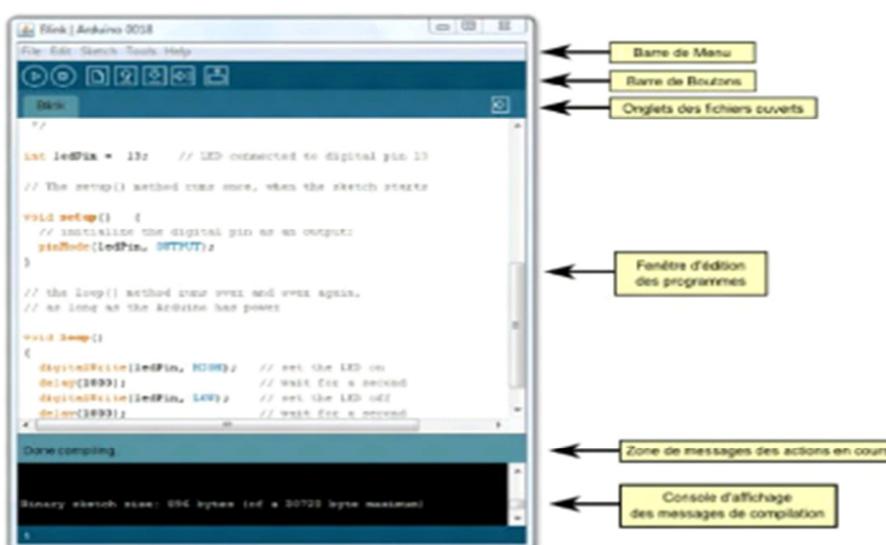


Figure I-7 :Les différentes parties de la fenêtre principale du logiciel Arduino

Il comporte :

- Une **BARRE DE MENUS**: comme pour tout logiciel une interface graphique (GUI).
- Une **BARRE DE BOUTONS**: cette barre donne un accès direct au fonctions essentiels du logiciel .
- Un **EDITEUR** (a coloration syntaxique) : c'est dans cette espace ou on va écrire notre programme, il dispose aussi d'onglets de navigation.
- Une **ZONE DE MESSAGES** : permet d'afficher et d'indiquer l'état des action en cours (ex : vérification, télé versement)
- Une **CONSOLE TEXTE** : elle permet d'afficher les messages concernant le résultat de la compilation du programme (il indique s'il y a des erreurs).
- Un **MONITEUR SERIE** : ce moniteur est utilisé pour la visualisation des données (message, valeurs, caractères) transmises vers l'ordinateur via le câble et d'envoyer des caractères vers la carte Arduino.
- Cette fonctionnalité permet une mise au point des programmes et d'afficher sur l'ordinateur l'état des variables d'un programme ou il y a des calculs ou des mesures, aussi le résultat de la conversion analogique-numérique .

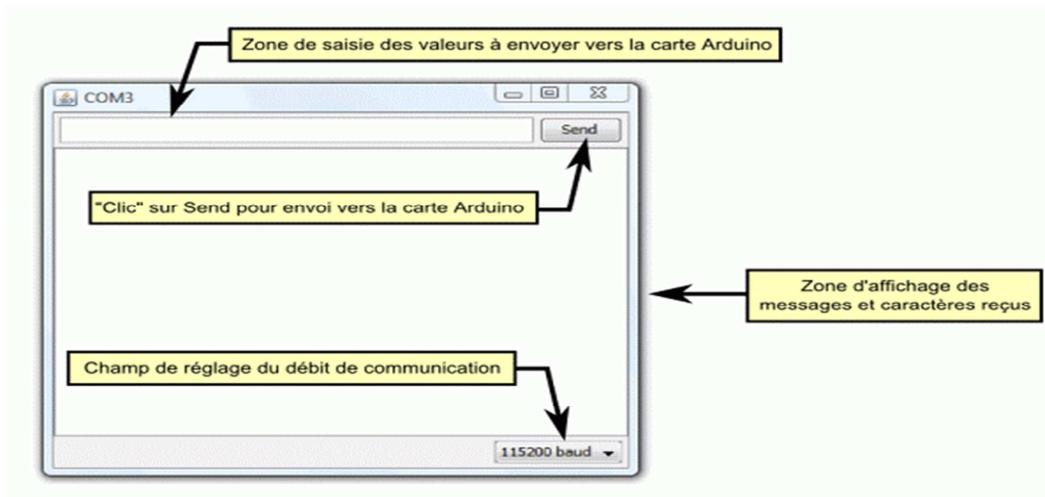


Figure I-8 :Fenêtre associée au moniteur série

I-13 Description de la barre des boutons :

La barre des boutons nous permet de vérifier la syntaxe, le transfert des programmes, aussi la création, l'ouverture et la sauvegarde d'un code écrit, elle permet aussi d'ouvrir le moniteur série, elle dispose de sept bouton ou icones :

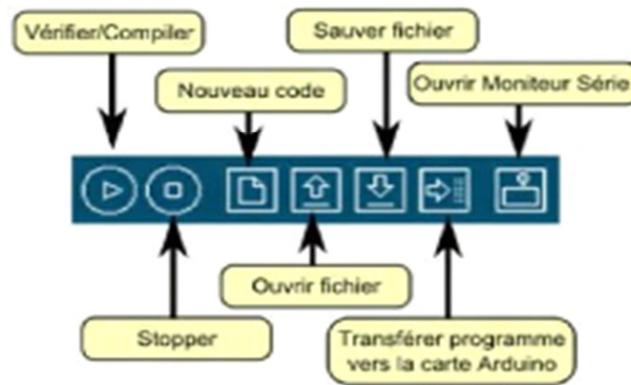


Figure I-9 :Structure de la barre des boutons



vérifier/compiler : ce bouton nous permet de vérifier le code a la recherche d'erreur dans le programme en sélectionnant la ligne ou se trouve l'erreur.



Stop : ce bouton oblige le moniteur série ou les autres boutons activés de s'arrêter.



Nouveau : en appuyant sur cette icone, une fenêtre d'édition vide s'ouvre pour saisie d'un nouveau programme.



Ouvrir : il suffit de cliquer sur ce bouton pour accéder a la liste de tous les programmes qui se trouvent dans le <<livre de programmes>> cliquer sur l'un des programmes s'ouvre dans la fenêtre courante .



Sauvegarder : avant de quitter le programme saisi, il faut l'enregistrer, et ça peut se faire en cliquant sur cette icone .



Transférer ou télé verser vers la carte : ce bouton permet de compiler votre code et de le transférer vers la carte Arduino.



Moniteur série : pour ouvrir la fenêtre du moniteur il suffit d'appuyer sur ce bouton .

I-14 Description de la barre des menus :

Le logiciel Arduino est équipé d'une barre des menus qui contient un ensemble de menus (d'où le nom <<barre des menus >>= permettant de faire la gestion du programme.

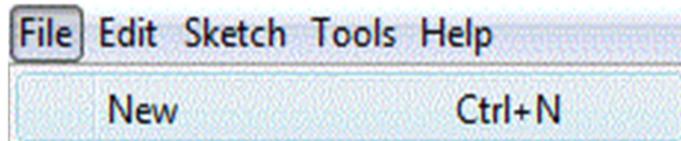


Figure I-10 :La barre des menus

Cette barre contient les icones suivantes :

- **File (Fichier) :** ce menu contient les différentes options de création, d'ouverture, de sauvegarde, d'impression d'un programme, ou l'ouverture d'un exemple parmi les exemples qui accompagnent le logiciel Arduino.
- **Edite (Editer) :** ce menu contient les options de copier/coller, sélection, et les options de recherches .
- **Sketch (Programme ou séquence):** ce menu contient les différentes fonctions de la barre des boutons, ainsi que les options d'ajout de librairie ou de fichiers.
- **Tools (Outils) :** c'est dans ce menu qu'on sélectionne le type de carte a programmer, et le port série utilisé ainsi que la fonction de chargement du boot loader dans l'ATMega .
- **Help (Aide) :** ce menu est fait pour donner de l'aide concernant les différents problèmes rencontrés au niveau du logiciel Arduino .

I-15 Les étapes de télé versement d'un programme vers la carte Arduino :

Une fois le programme est saisie (première phase), une deuxième phase consiste à vérifier s'il y a des erreurs dans le programme, et pour faire il suffit de cliquer sur le bouton de **vérification** se trouvant à l'extrémité gauche de la barre des boutons et attendre un petit moment .

Après cette vérification s'il y des erreurs, un message en orange s'affiche au-dessus de l'onglet d'édition des programmes indiquant qu'elle est l'erreur et la ligne ou se trouve. Une fois la correction des erreurs est terminée, on passe à la phase de la sélection du type de la carte à programmer et le port série sur lequel est branchée cette carte. Ce petit réglage peut se

faire à l'aide du menu **Tools(Outils)** . une fois terminé il reste qu'à cliquer sur le bouton du moniteur série .

Si tout marche bien les deux LEDs Rx et Tx de la carte s'allument, sinon un message d'erreur s'affiche dans la console des messages de compilation.

A ce niveau la carte est programmée est prête pour l'exécution du programme.

Discussion :

Dans ce chapitre, mon étude s'est focalisée sur la présentation de la carte arduino, et ça commence par un bref historique et quelques détails concernant la création et l'utilité de cette carte. Puis j'ai présenté son côté matériel. La compréhension de donnée exposées dans ce chapitre nous offre des bases de connaissances pour l'usage de la carte arduino .

Chapitre II : système d'acquisition des images

Préambule :

Dans ce chapitre, nous allons montrer le principe d'acquisition d'images d'une part et d'autre part nous allons expliciter les concepts et les technologies mises en œuvre dans l'acquisition d'images

L'acquisition d'images est une partie intégrante d'une machine de vision, que l'on trouve en milieu industriel ou de laboratoire.

L'acquisition d'images représente le module qui relie le monde réel, de l'image stockée en mémoire et accessible via le programme applicatif. On peut le décomposer en plusieurs sections : l'optique, le capteur, le transport des données images (format et support)

II-1 Définition de l'image :

Une image est une représentation du monde. En traitement d'image, la majorité du temps, on considère qu'il s'agit d'une fonction mathématique de $R \times R$ dans R où le couplet d'entrée est considéré comme une position spatiale, le singleton de sortie comme l'intensité (couleur ou niveaux de gris) du phénomène physique. Il arrive cependant que l'image soit dite "3D" donc la fonction est de $R \times R \times R$ dans R . Les images couleurs peuvent être représentées soit par trois images représentant les trois couleurs fondamentales, soit par une image de $R \times R$ dans $R \times R \times R$.

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter.

II-2 Types d'images :

On distingue deux types, images vectorielles et les images matricielles.

- Une image vectorielle est composée de formes géométriques pouvant faire l'objet d'une description mathématique (droites, cercles, points, ...). Une image vectorielle se redimensionne aisément, sans aucune perte de qualité. Ces images sont très utiles pour des reproductions à grande échelle, des calculs pouvant être réalisés à chaque fois, afin d'obtenir l'image exacte, quelle que soit la taille choisie.
- Une image matricielle est composée d'un ensemble de points (pixels). Chacun d'eux a une position et une couleur bien précises. Les photos numériques et les documents

scannés sont des images matricielles. En cas d'agrandissement, une perte de qualité peut être remarquée.

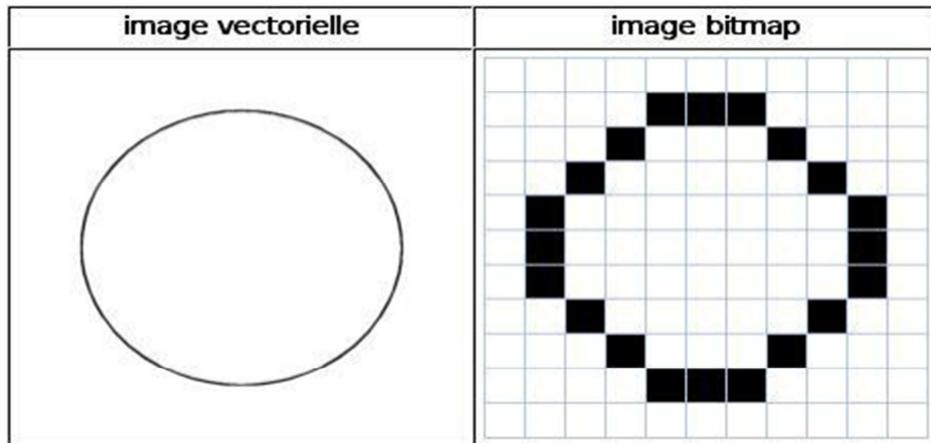


Figure II-1 : Image vectorielle et image bitmap

II.3 Codages des couleurs

Une image apparaît comme une matrice où chaque case contient des nombres associés à une couleur. Usuellement on distingue 3 grands types de couleurs pour une image numérique :

- Le noir et blanc.
- Les niveaux de gris
- La couleur.

Ces types sont généralement à choisir lors d'une numérisation par scanner ou lors de la configuration d'un appareil photographique.

II-3-1 Image noir et blanc :

Le noir et blanc est le plus simple. Le contenu de chaque case de la matrice est soit un 0 (noir) soit 1 (blanc). Le nombre de couleurs n'est que de 2 et le rendu de l'image le moins performant mais parfois suffisant dans le cadre par exemple de documents scripturaux.

II-3-2 Niveaux de gris :

Le codage dit en niveaux de gris permet d'obtenir plus de nuances que le simple noir et blanc. Il offre des possibilités supplémentaires pour coder le niveau de l'intensité lumineuse. La couleur est codée souvent sur un octet soit 8 bits ce qui offre la possibilité d'obtenir 256 niveaux de gris (0 pour le noir et 255 pour le blanc). On peut aussi le faire avec 16 niveaux de gris (4 bits).

Plus le niveau de gris est élevé, meilleur est la distinction des détails sur l'image. L'usage de ce codage est utilisé fréquemment pour la presse écrite ou l'envoi par messagerie électronique de fichier d'image de taille réduite avec une perte de lisibilité de l'image moindre.

II-3-3 Image couleur :

1 Principe

La couleur d'un pixel est obtenue, comme le ferait un peintre, par le mélange de couleurs fondamentales. Il ne s'agit pas ici de décrire toutes les techniques utilisées. Nous allons décrire un des principes les plus couramment utilisé qui est celui de la synthèse additive.

2 Codage RVB:

Le principe consiste à mélanger les 3 couleurs : rouge, vert et bleu (noté RVB ou RGB en anglais). A l'aide de ces 3 couleurs, on obtient toute une palette de nuances allant du noir au blanc. A chaque couleur est associé un octet (donc 256 niveaux de luminosité) de chacune des couleurs fondamentales.

II-4 Caractéristiques d'une image numérique:

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

Dimension :

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

Résolution :

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution.

Bruit :

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

Histogramme :

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant

Luminance:

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par:

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

Contraste :

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport

$$C = \frac{L1 - L2}{L1 + L2}$$

II-5 Acquisition d'une image :

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales :

- les caméras numériques.
- et les scanners.

A ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, les arts graphiques, la médecine, etc. Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device). La carte d'acquisition reçoit les images de la camera, de la T.V. ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier. L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément (pixel), mais aussi l'intercorrélation de deux éléments voisins : plus cette intercorrélation est faible, meilleure est l'image.

II-6 La caméra :

Une caméra, c'est une boîte contenant le capteur d'images, Le but de la camera est de convertir une image formée par un faisceau de photons, en un signal électrique compréhensible par un autre système. Pour remplir cette tâche, la camera possède un certain nombre d'entrées et de sorties accessibles à l'extérieur du boîtier. Une ouverture, située à l'avant de la camera, permet la fixation de l'objectif.

II-7 structure de la camera :

la camera est structurée par :

II-7-1 Objectif de la camera :

L'objectif se situe entre le monde réel et le capteur d'images. Il permet de faire converger les rayons lumineux d'un point de l'espace 3D sur un point du plan image.

Chaque point de l'objet que l'on observe émet des photons dans une multitude de direction. Le premier principe de l'objectif est de faire converger le plus de rayons possible en un point unique, le second principe repose sur le fait que l'ensemble des points images formes se projette sur un même plan L'objectif se présente sous la forme d'un cylindre.

L'intérieur de l'objectif est compose d'un certain nombre de lentilles assemblées de façon très précises.



Figure II-2: l'objectif de la camera

II-7-2 Les paramètres de l'objectif :

Un objectif se caractérise par plusieurs paramètres.

- **La focale** : elle indique la distance entre le centre optique et le plan focal. Son unite est le millimètre (mm). Elle caractérise l'angle de vue. Plus la focale est petite et plus l'angle de vue est grand.

- **Lafixation ou monture** : elle désigne la façon de fixer l'objectif sur la camera.
- **L'iris** : il régule la quantité de lumière qui traverse l'objectif.
- **Le focus (netteté ou mise au point)** : il permet de régler la distance entre l'objectif et le plan principal de focus. La donnée est qu'un objectif ne peut faire converger correctement (de façon nette) qu'un seul plan du monde réel sur le plan image. Régler le focus, correspond à changer la valeur de ce plan de manière à le faire coïncider avec le plan réel qui convient à l'utilisateur.
- **Pour quelle camera** : chaque objectif a un paramètre indiquant pour quelle taille de capteur d'image il a été réalisé, Sa valeur peut être 1/4, 1/2, 2/3 ou 1 pouce et indique que le capteur image ne peut être plus grand que ce paramètre
- **La Qualité** : elle est pour une grande part liée aux lentilles utilisées pour construire l'objectif. Il faut plus particulièrement s'intéresser à plusieurs points.
 - La réponse spectrale : elle définit comment se comporte l'objectif en fonction de la longueur d'onde du photon. C'est un point important pour analyser la couleur.
 - La distorsion : elle est responsable de la déformation de l'image.

II-8 Le capteur d'images :

Qu'ils soient de type CCD ou CMOS, ce sont tous deux des capteurs photographiques convertissant une image de départ issue d'un rayonnement électromagnétique en un signal analogique (ou numérique) à la fin.

Le principe de base d'un capteur repose sur l'effet photoélectrique, qui permet à un photon incident d'arracher des électrons à chaque élément sensible (appelé photosite) d'une matrice de petits capteurs élémentaires.

II-9 Le capteur CCD :

Ce sont les capteurs inventés en premier. Ils sont apparus en 1970 dans les Laboratoires Bell.

Les capteurs appelés CCD (ChargedCoupledDevice ou en français Dispositif de Transfert de Charges), ceux que l'on trouve sur la plupart des Appareils Photo Numérique (APN) sont des systèmes à transfert de charges.

Un CCD transforme les photons lumineux qu'il reçoit sur l'électrode semi-transparente en électron par effet photoélectrique, puis collecte les électrons dans la zone désertée. Le

nombre d'électrons collectés est proportionnel à la quantité de lumière reçue. À la fin de l'exposition, les charges sont transférées de photosite en photosite par le jeu de variations de potentiel cycliques provenant d'une horloge externe. Puis elles sont envoyées dans un registre de sortie. Enfin les charges sont transformées en tension proportionnelle au nombre d'électrons. Ce signal sera, à l'extérieur du CCD, amplifié et numérisé.

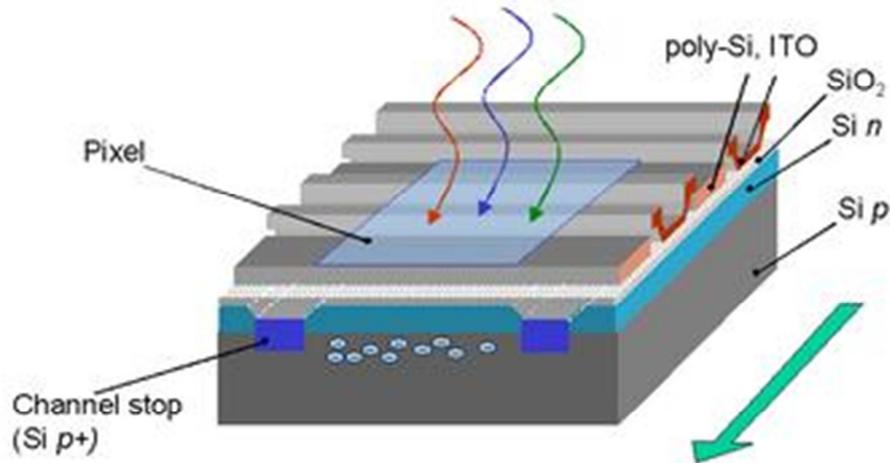


Figure II-3: Constitution d'un photo-site

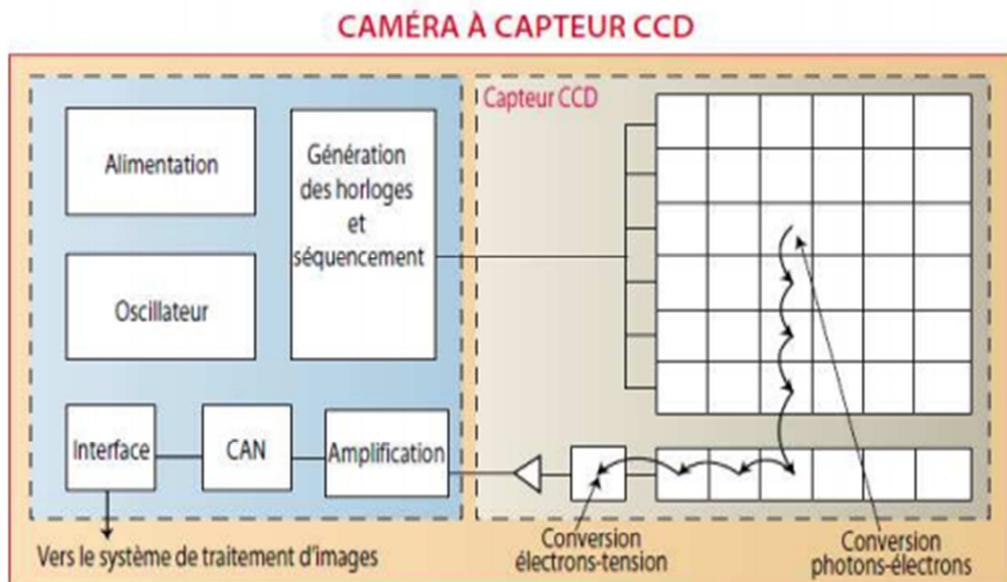


Figure II-4: Fonctionnement général d'une caméra à capteur CCD

II-10 Réponse spectrale :

Les capteurs CCD "interligne" ont une courbe de réponse spectrale allant de 400 à 1000 nm (environ) comme le montre le graphique ci-dessous. La sensibilité à la lumière

infrarouge proche (700 à 10000 nm) peut permettre à une caméra de fonctionner de jour comme de nuit. Le passage du visible à l'infrarouge est possible grâce à un filtre passe-bas. Ce filtre est utilisé pendant la journée permettant l'imagerie couleur. Et la nuit on obtient une image monochrome

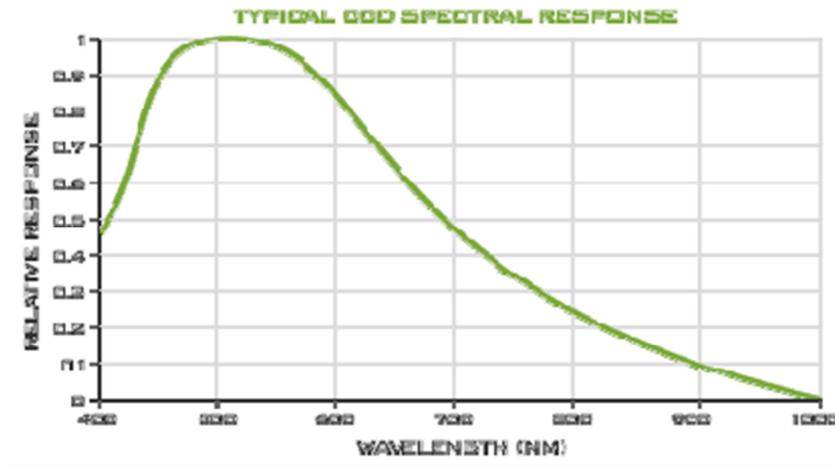


Figure II-5 :Réponse spectrale d'un capteur CCD

II-11 Le capteur CMOS :

Les capteurs CMOS (ComplementaryMetalOxideSemiconductor ou en français Semi-conducteur à Oxyde de Métal Complémentaire) fonctionnent différemment, même si le principe de base reste le même. Les photo sites, comme sur un CCD sont sensibles à la lumière et acquièrent une certaine charge électrique en fonction de la quantité de lumière reçue. Jusqu'ici pas de différence. Mais ensuite, à l'intérieur de chaque photo site, la charge générée est convertie directement en tension utilisable.

Suivant le schéma ci-après :

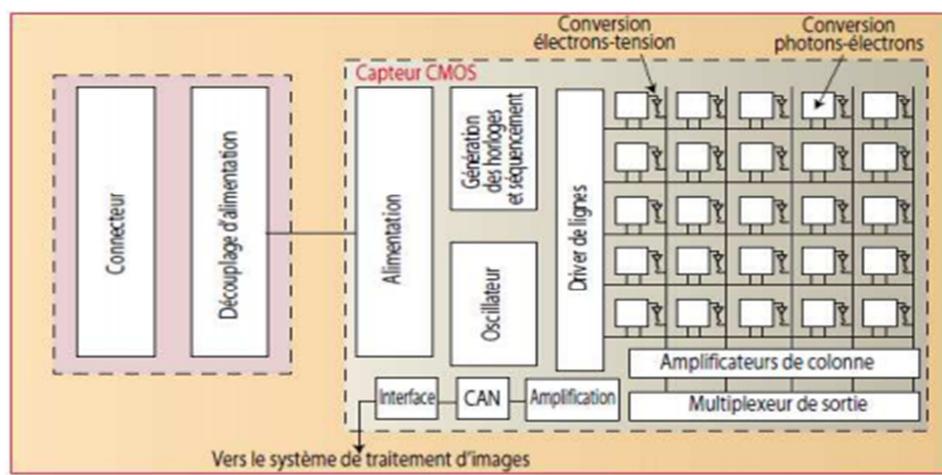


Figure II-6: fonctionnement général d'une caméra à capteur CMOS

La principale différence est la nature du signal sortant d'un photo site. Pour le CCD il s'agit de charges et pour le CMOS c'est déjà un signal sous forme numérique (concernant le CCD, les charges seront converties en tension après le registre de sortie).

II-12 Réponse spectrale :

Le graphique ci-après représente la réponse spectrale d'un capteur CMOS. Le pic de la réponse spectrale est proche de celui des capteurs CCD "plein cadre". Pour les CMOS il se situe vers 700 nm :

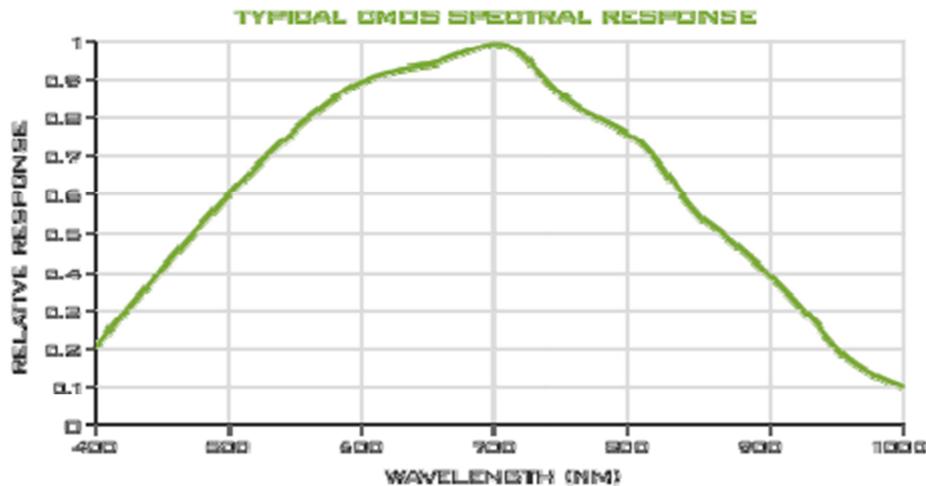


Figure II-7: Réponse spectrale d'un capteur CCD

II-13 Domaine d'utilisation des capteurs :

Les capteurs CCD sont à l'origine employés pour toutes les applications nécessitant un haut niveau de précision, pour les systèmes de traitement d'images complexes, les applications scientifiques, pour celles présentant un faible niveau d'éclairage.

Les capteurs CMOS eux sont utilisés pour le contrôle industriel simple, ne nécessitant pas impérativement une qualité d'image très élevée. Par ailleurs, en utilisant la méthode de global shutter, les capteurs CMOS ont un intérêt pour l'acquisition d'objets en mouvement ou à grande vitesse. Le global shutter permet une exposition entière et en même temps de tous les photo sites présent sur le capteur. Normalement les CMOS fonctionnent ligne de photo sites par ligne de photo sites (méthode de rollingshutter).

A l'origine les capteurs CCD étaient plus performants. Ils sont apparus en premier. Et les capteurs CMOS étaient utilisés pour des appareils ne devant pas donner des images de très

grandes qualités (les webcams, les appareils photo numériques bas de gamme ou de téléphone portable).

Avec le temps, le CCD s'est amélioré mais le CMOS également. Aujourd'hui, un capteur CMOS obtient les mêmes performances qu'un capteur CCD.

II-14 Taille du capteur :

L'appellation de la taille des capteurs type CCD ou CMOS est une abréviation. En effet, pour spécifier la taille d'un capteur, on parle de quart de pouce, de demi pouce, ces dimensions ne correspondent à rien du point de vue physique. Il s'agit en fait d'équivalence avec la diagonale des capteurs.

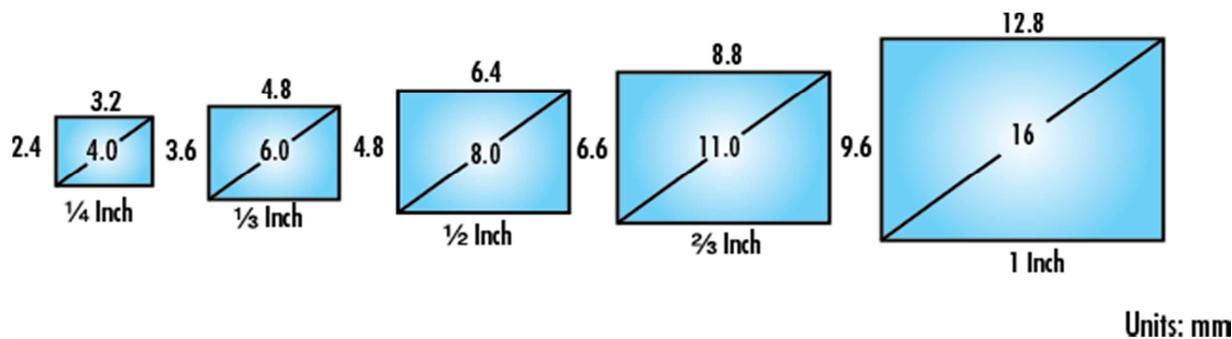


Figure II-8: Illustration de dimension de taille de capteur

à partir des caractéristiques dimensionnelles du capteur CCD de la camera et de celles de la scène à visualiser, les deux formules ci-dessous permettent de déterminer le choix de l'objectif :

$$f = l * D / W \quad = h * D / H$$

Avec f la focale de l'objectif (mm), H la hauteur du plan à visualiser (m), W la largeur du plan à visualiser (m), D la distance entre l'objectif et le plan (m), h la hauteur du capteur CCD (mm) et l la largeur du capteur CCD (mm), les dimensions des capteurs CCD sont données dans le tableau suivant :

| Taille | Diagonale (en mm) | Hauteur (en mm) | Largeur (en mm) |
|--------|----------------------|--------------------|--------------------|
| 35mm | 43,3 | 36 | 24 |
| APS | 30,1 | 25,1 | 16,7 |
| 1/3,6" | 5 | 4 | 3 |
| 1/3,2" | 5,68 | 4,536 | 3,416 |
| 1/3" | 6 | 4,8 | 3,6 |
| 1/2,7" | 6,592 | 5,27 | 3,96 |
| 1/2" | 8 | 6,4 | 4,8 |
| 1/1,8" | 8,933 | 7,176 | 5,319 |
| 2/3" | 11 | 8,8 | 6,6 |
| 1" | 16 | 12,8 | 9,6 |
| 4/3" | 22,5 | 18 | 13,5 |

Tableau II-1 :Dimensions des capteurs CCD

II-15 Ouverture du champ visuel de la camera :

Le but est de calculer le champ visuel en fonction du matériel. Les paramètres principaux entrant en ligne de compte sont : les dimensions (horizontale et verticale) du capteur, la focale de l'objectif et les angles de vue (horizontale et verticale). Le schéma au-dessous montre les grandeurs mises en jeu. Il peut être interprète pour le calcul horizontal ou vertical du champ.

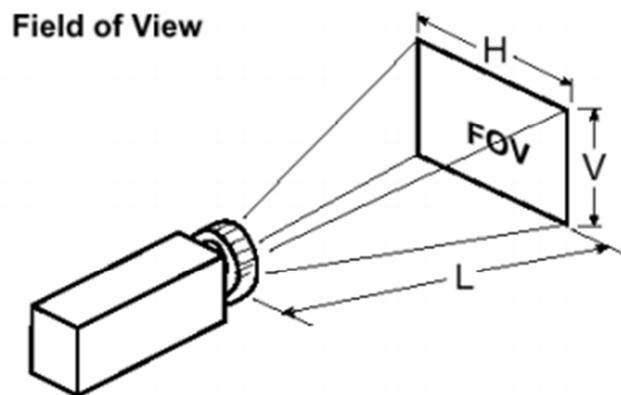


Figure II-9 : Champ visuel de caméra

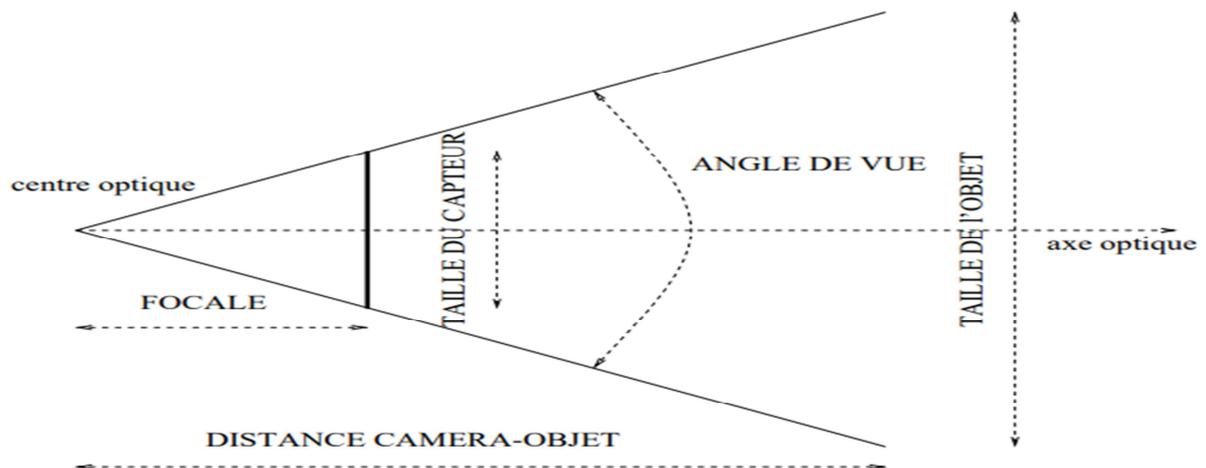


Figure II-10:Schéma permettant le calcul du champ de vue

| focale(mm) | distance(m) | objectif photo |
|------------|-------------|----------------|
| 6 | 1.25 | 30 |
| 8 | 1.67 | 40 |
| 10 | 2.08 | 50 |
| 12 | 2.5 | 60 |
| 16 | 3.33 | 80 |
| 25 | 5.20 | 125 |

Tableau II-2 : Distance d'un objet en fonction de la focale

on cherche à déterminer l'angle de vue, ou encore la taille maximale d'un l'objet a une certaine distance. On a alors :

- angle de vue = arc tangente(taille du capteur/focale)
- taille objet = (taille du capteur * distance objet/focale)
- focale = (taille du capteur * distance objet/taille objet)

II-16 numérisation d'image :

Un signal analogique est un signal continu qui peut prendre une infinité de valeurs, alors que le signal numérique est un signal discret (discontinu), qui se résume en une succession de « 0 » et de « 1 ».

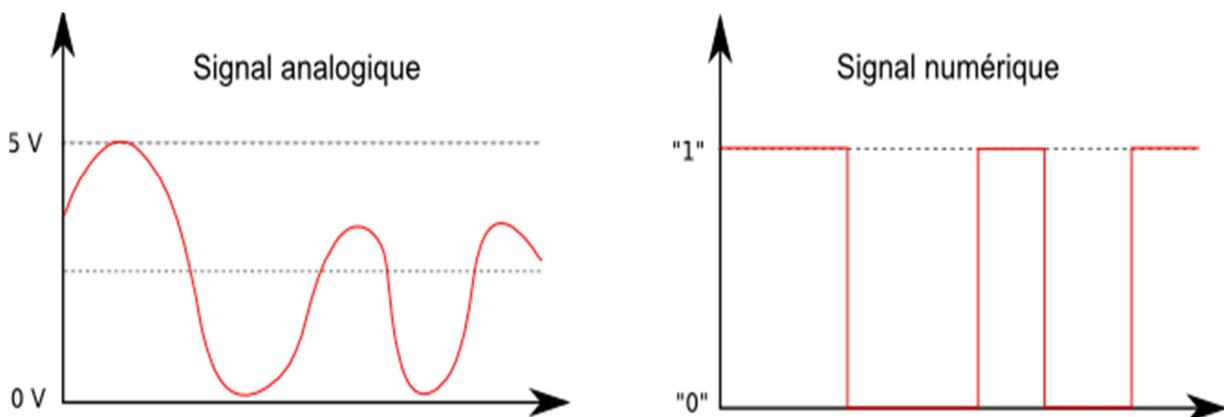


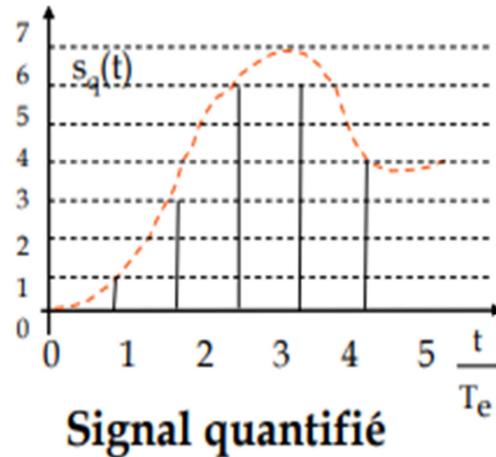
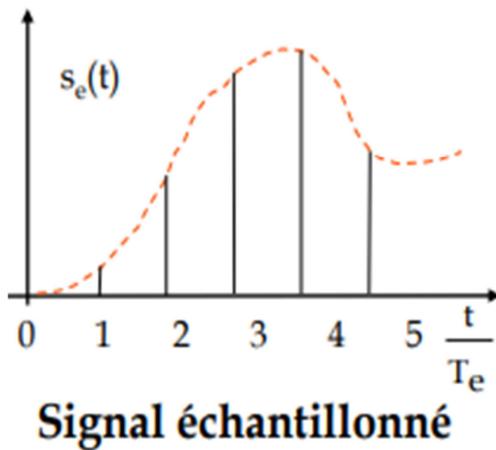
Figure II-11 : signal analogique et signal numérique

L'objectif de la numérisation est de transformer le signal analogique qui contient une quantité infinie d'amplitudes en un signal numérique contenant lui une quantité finie de valeurs.

La conversion est caractérisée par deux discrétisations

- la 1ère concerne **le temps** et porte le nom **d'échantillonnage** : cela consiste à prendre des échantillons du signal analogique à des instants régulièrement espacés

– La 2e concerne l'**amplitude** et porte le nom de **quantification** : cela consiste à coder l'amplitude du signal sur un nombre fini d'éléments binaires : CAN



II-17 Les filtres anti-repliement :

Pour limiter le contenu fréquentiel du signal en entrée, un *filtre passe-bas* (un filtre qui laisse passer les basses fréquences mais qui atténue les hautes fréquences) est ajouté à l'échantillonneur et au C A/N. Ce filtre est un filtre anti-repliement car en atténuant les hautes fréquences (supérieures à la fréquence de Nyquist), il empêche d'échantillonner les composantes qui pourraient se replier. Comme à ce point-ci (avant l'échantillonneur et le C A/N), on est encore dans le monde analogique, le filtre anti-repliement est un filtre analogique

II-18 Choix à effectuer lors de la numérisation :

- Précision de numérisation via le pas de **quantification** – Ce problème peut assez facilement être traité en augmentant le nombre de bits du CAN
- Précision de discrétisation via la fréquence **d'échantillonnage** f_e – f_e doit être suffisamment élevée si l'on ne veut pas perdre trop d'informations sur le signal. – Cependant plus f_e est élevée (T_e faible), plus le nombre d'échantillons à traiter sera important et plus le temps disponible pour effectuer les traitements numériques sera court

Discussion :

Ce chapitre a fourni une description, étape par étape, des différents éléments qui constituent un système d'acquisition d'images. L'acquisition d'images concerne plusieurs technologies, comme par exemple, l'optique ou les capteurs sur circuits intégrés, et s'appuie sur différentes normes. On a donné l'état actuel de ces éléments, en apportant à chaque fois, les bases nécessaires à notre compréhension. Ce chapitre est, par conséquent, aussi un point de départ pour approfondir un domaine particulier.

Chapitre III : Conception et réalisation d'un système d'acquisition d'images à base Arduino.

Préambule

Dans ce chapitre nous allons concevoir et réaliser une application d'acquisition d'images à base d'arduino en utilisant le module Arducam 2mp camera module. Ce système peut être utilisé sur de multiples projet tels que : drone, véhicule, robotique, domotique

Nous allons utiliser une caméra ov2640 fixe reliée à une carte Arduino. Cette caméra se compose de capteur d'image, interface I2C, interface SPI.), dès que la camera s'enclenche elle capture des images avec une dimension prédéfinie qui seront envoyés vers l'ordinateur via une liaison série, les réglages se font à l'aide du logiciel Arducam Host V2.0. Ce dernier dispose d'une interface graphique et d'un moniteur dans lequel on pourrait visionner les images capturées.

III-1 Camera ArduCAM 2MP :

ArduCAM Mini est une version optimisée du blindage ArduCAMRev.C, avec une caméra SPI haute définition, qui réduit la complexité de l'interface de contrôle de la caméra. Il intègre le capteur d'image CMOS 2MP OV2640, fournit une taille miniature, ainsi qu'une interface matérielle facile à utiliser et une bibliothèque de code « open source ».

L'ArduCAM mini peut être utilisé sur toutes les plates-formes comme Arduino, Raspberry Pi, Maple, Chipkit, Beaglebone noir, tant qu'elles ont une interface SPI et I2C et peuvent être bien accouplés avec des cartes Arduino standard.

ArduCAM mini offre non seulement la possibilité d'ajouter une interface de caméra qui n'est pas dans certains microcontrôleurs à faible coût, mais offre également la possibilité d'ajouter plusieurs caméras à un seul microcontrôleur.



Figure III-1: cameraArduCAM mini

III-1-1 Caractéristiques :

Les caractéristiques de la caméra sont

- Monture M12 ou support de lentille CS avec options d'objectif interchangeable
- IR sensible avec la bonne combinaison de lentilles
- Interface I2C pour la configuration du capteur
- Interface SPI pour les commandes de la caméra et le flux de données
- Tous les ports IO sont tolérants 5V / 3.3V
- Soutenir le mode de compression de JPEG, le mode de tir simple et multiple, l'opération multiple de lecture de capture d'une fois, l'opération de lecture d'éclatement, le mode de basse puissance et etc.
- Sortie vidéo 3 ~ 10fps à basse résolution
- Bien accouplé avec les cartes Arduino standard
- Fournir une bibliothèque de code source libre pour Arduino, STM32, Chipkit, Raspberry Pi, BeagleBone Black
- Petite forme de facteur

III-1-2 Spécifications techniques:

- Capture d'écran : OV2640.
- Capteur d'image 2MP OV2640.
- IR sensible avec la bonne combinaison de lentilles.
- Interface I2C pour la configuration du capteur.
- Tous les ports IO sont tolérants 5V / 3.3V.
- Sortie vidéo 3 ~ 10fps à basse résolution.
- Taille active du tableau : 1600 x 1200.
- Obturateur : volet roulant.
- Lentille : ¼ pouce.

III-1-4 Fonctionnement :

❖ Mode de capture unique :

Le mode de capture unique est le mode de capture par défaut de la caméra.

Après l'émission d'une commande de capture via le port SPI, l'ArduCAM attendra une nouvelle trame et tamponnera les données d'une image entière dans le tampon d'image, puis affinera le bit d'indicateur d'achèvement dans le registre. L'utilisateur doit seulement interroger le bit de drapeau du registre pour vérifier si la capture est faite.

❖ Mode de capture multiple :

Le mode de capture multiple est le mode de capture avancé. En définissant le nombre d'images dans le registre de capture, l'ArduCAM capturera les trames consécutives après l'émission de la commande de capture. Notez que le nombre d'images doit être réglé correctement et s'assurer de ne pas dépasser le maximum espace mémoire.

❖ Compression JPEG :

La fonction de compression JPEG est implémentée dans le capteur d'image. Avec des paramètres de registre appropriés pour le capteur, l'utilisateur peut obtenir une résolution différente avec la sortie de flux d'image JPEG. Il est recommandé d'utiliser la sortie JPEG pour obtenir une résolution plus élevée que le mode RVB, en raison de la limitation du tampon d'image.

❖ Lecture normale et opération de lecture en rafale :

L'opération de lecture normale lit chaque donnée d'image en envoyant une commande de lecture dans un cycle d'opération de lecture SPI. Pendant l'opération de lecture en rafale, il suffit d'envoyer une commande de lecture puis de lire plusieurs données d'image dans un cycle d'opération de lecture SPI. Il est recommandé d'utiliser l'opération de lecture en rafale pour obtenir de meilleures performances de débit.

❖ Rembobiner l'opération de lecture :

Parfois, l'utilisateur veut lire la même image de données d'image plusieurs fois pour le traitement, l'opération de lecture de rembobinage est conçue à cet effet. En réinitialisant le pointeur de lecture au début des données d'image, l'utilisateur peut à nouveau lire les mêmes données d'image à partir du point de départ.

❖ Mode faible consommation :

Certains périphériques d'alimentation de batterie nécessitent une économie d'énergie lorsqu'ils sont en mode inactif, l'ArduCAM offre le mode de faible consommation d'énergie pour réduire la consommation d'énergie, en arrêtant le capteur et les circuits de mémoire.

❖ Contrôle du capteur d'image :

La fonction de commande du capteur d'image est implémentée dans le capteur d'image. En réglant correctement les paramètres du registre, l'utilisateur peut contrôler l'exposition, la balance des blancs, la luminosité, le contraste, la saturation des couleurs, etc.

III-1-5 Câblage du matériel de la caméra :

Cette note d'application décrit le fonctionnement matériel détaillé de l'appareil photo ArduCAM-M-2MP bouclier.

III-1-5-1 Câblage typique :

III-1-5-1-1 Câblage d'une seule caméra :

La connexion typique entre ArduCAMshield et Arduino ou plate-forme . est montrée dans la figure 1. Plus typiquement la figure 2 montre le câblage pour la carte Arduino UNO R3.

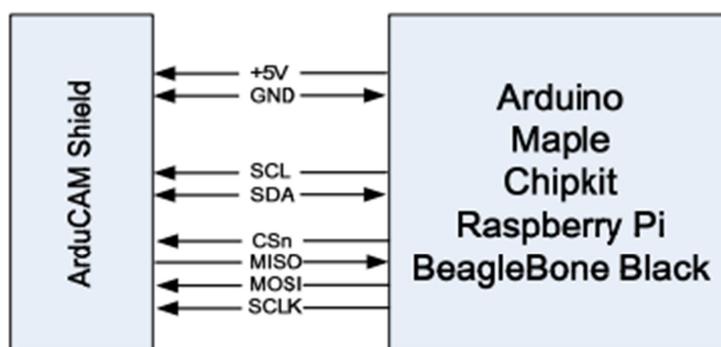


Figure III-2: câblage typique

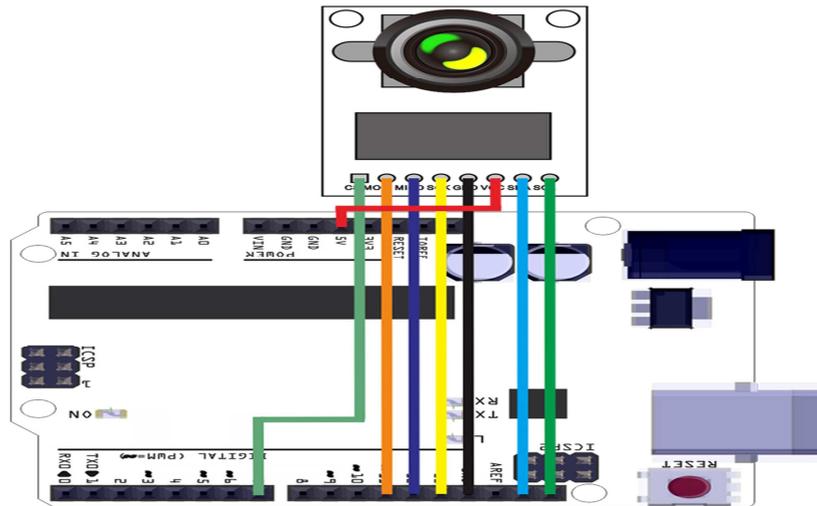


Figure III-3: câblage pour Arduino UNO R3

III-1-5-1-2 Câblage multi caméras :

La connexion multi-caméras entre ArduCAMshield et Arduino ou plate-forme est montré dans la figure 3. Plus typiquement la figure 4 montre le câblage multi-caméras pour Arduino UNO R3

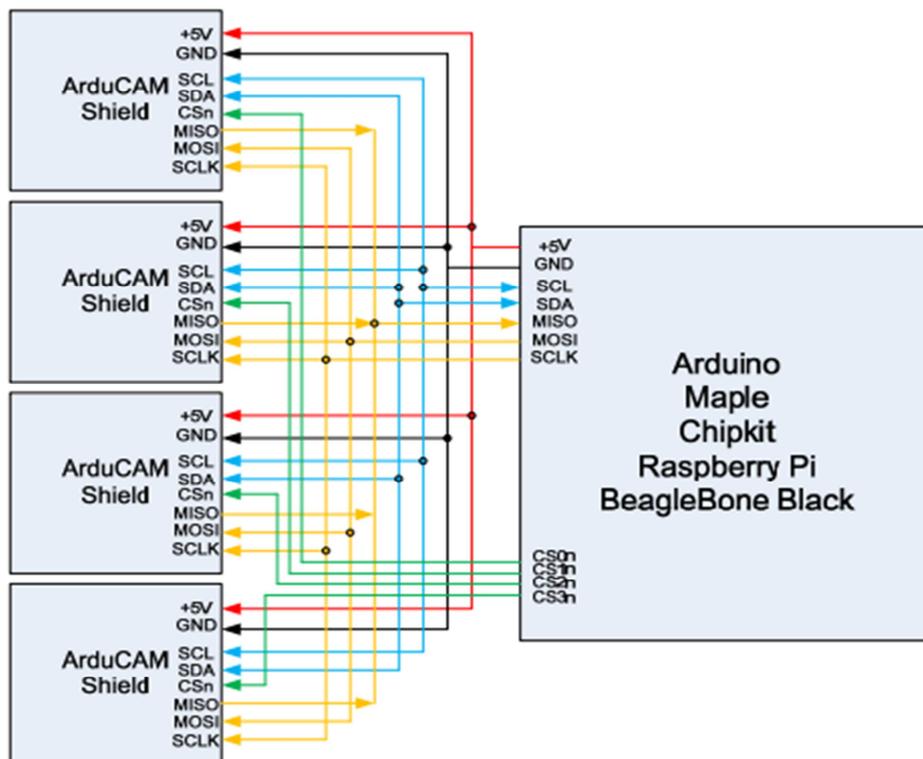


Figure III-4: câblage multi-caméras

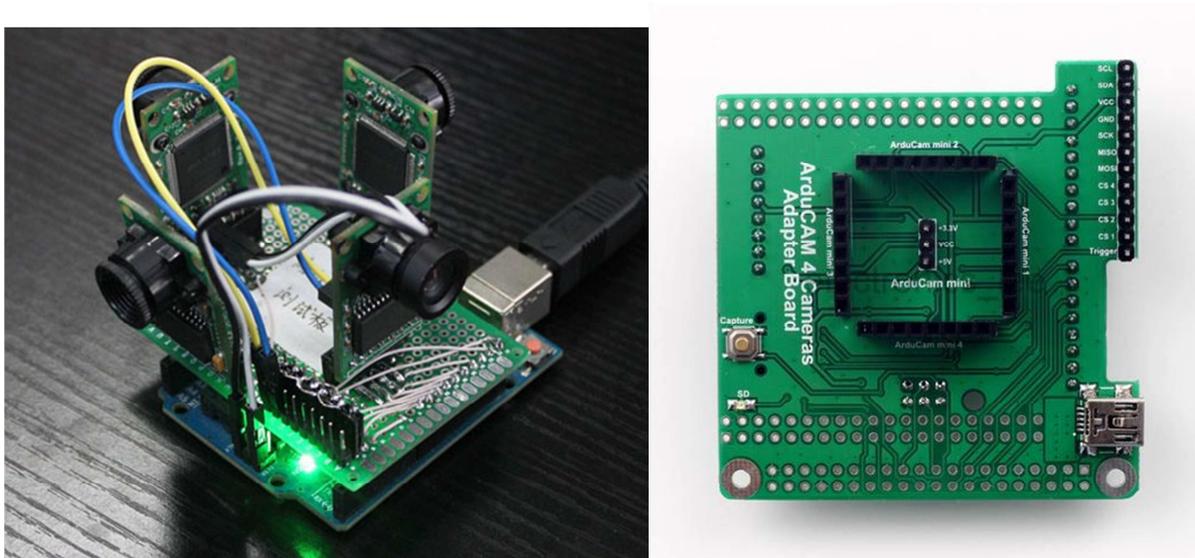


Figure III-5: câblage multi-cameras sur Arduino UNO R3

III-1-5-2 Interface I2C:

Le bus I2C (Inter Integrated Circuit) a été développé au début des années 80 par Philips semi-conducteurs pour permettre de relier facilement des systèmes électroniques. Prévu initialement pour des équipements domestiques, il est devenu un moyen privilégié de configuration de composants électroniques complexes (encodeur vidéo).

Le bus I2C permet de faire communiquer entre des composants électroniques très divers grâce à seulement trois fils : Un signal de donnée (SDA), un signal d'horloge (SCL), et un signal de référence électrique (Masse).

SDA (Serial Data) : Ligne permettant au maître et à l'esclave d'envoyer et de recevoir des données.

SCL (Serial Clock) : La ligne qui porte le signal d'horloge.

III-1-5-3 Interface périphérique série (SPI) :

Serial peripheral Interface ou **SPI**, est un protocole de communication très couramment utilisé pour une communication bidirectionnelle entre deux dispositifs. Un bus SPI standard se compose de 4 signaux, **Master Out Slave In (MOSI)**, **Master In Slave Out (MISO)**, l'horloge (**SCK**) et **Slave Select (SS)**. Contrairement à une interface série asynchrone, SPI n'est pas symétrique. Un bus SPI en a un **maître** et un ou plusieurs **esclaves**. Le maître peut parler à n'importe quel esclave sur le bus, mais chaque esclave peut seulement parler au maître. Chaque esclave sur le bus doit posséder son propre signal de sélection d'esclave. Le

maître utilise les signaux de *sélection d'esclave* pour *sélectionner l'esclave avec lequel il va parler*. Comme SPI inclut également un signal d'horloge, les deux appareils n'ont pas besoin de se mettre d'accord sur un débit de données auparavant. La seule exigence est que l'horloge soit inférieure à la fréquence maximale pour tous les appareils concernés.

- **SCLK** — Serial Clock, Horloge (généré par le maître)
- **MOSI** — Master Output, Slave Input (généré par le maître)
- **MISO** — Master Input, Slave Output (généré par l'esclave)
- **SS** — Slave Select, Actif à l'état bas (généré par le maître)

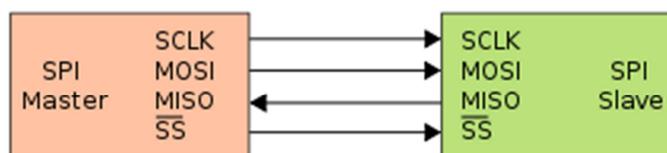


Figure III-6: Liaison SPI (un maître et un esclave)

III-1-6 Composition du logiciel de caméra SPI:

Le logiciel de la caméra SPI est composé :

Structure de la bibliothèque du logiciel :

La bibliothèque ArdCAM est conçue pour la plateforme Arduino, composée de deux sous-bibliothèques on aArduCAM et l'autre est UTFT4ArduCAM_SPI. Ces deux bibliothèques devraient être copiées juste sous les bibliothèques du répertoire Arduino afin d'être reconnues par l'ArduinoIDE. La structure des bibliothèques ArduCAM est représentée par la figure 1



Figure III-7: structure des bibliothèques ArduCAM

III-1-7 Guide de démarrage rapide :

Les bibliothèques doivent être configurées avant d'exécuter des exemples, sinon vous obtiendrez un message d'erreur de compilation. Ouvrez le fichier memorysaver.h dans le dossier ArduCAM et activez la plate-forme matérielle et module de caméra qui correspond à votre matériel par un commentaire ou dé-commenter la définition de la macro dans le fichier. Par exemple, si vous avez un ArduCAM-Mini-2MP vous devrait dé-commenter la ligne "#define OV2640_MINI_2MP" et commenter toutes les autres lignes.

Après avoir ouvert l'IDE Arduino, les exemples ArduCAM peuvent être trouvés dans le menu Fichier-> Exemples->ArduCAM comme le montre la figure 2.

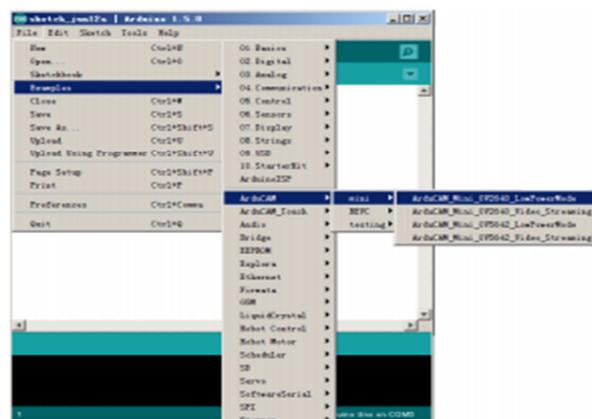


Figure III-8: Exemple Arduino IDE.

Après le téléchargement de l'esquisse d'exemple, l'utilisateur peut pré visualiser la vidéo en direct sur l'écran LCD si vous utilisez Modèle ArduCAM-LF (voir figure 5). Ou télécharger l'application hôte Windows(voir figure6) pour capture d'image si vous utilisez Modèle ArduCAM Mini.



Figure III-9: ArduCAM-LF

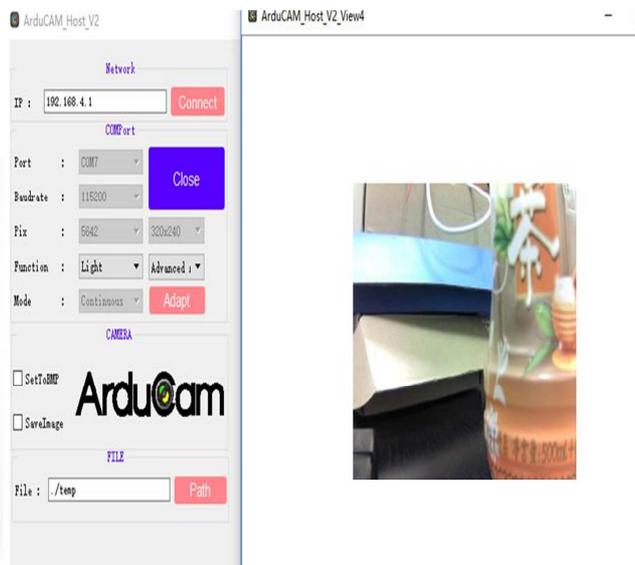


Figure III-10: application ArduCAM hostV2.0

III-1-8 Exemples d'esquisses :

Dans le dossier d'exemple, il y a six sous-répertoires pour différents modèles ArduCAM et le application hôte. Répertorie les listes de structure comme illustré dans la Figure 7. Le dossier ESP8266 est pourExemples de cartes ArduCAM-ESP8266-UNO. Le dossier Mini est pour ArduCAM-Mini-2MP . Le dossier Mini_5MP_Plus est pour ArduCAM-Mini-5MP-Plus (OV5640 / OV5642) modules. Le dossier RevC est pour ArduCAM-Shield-RevC ou Boucliers ArduCAM-Shield-RevC +. Le dossier Shield_V2 est pour le bouclier ArduCAM-Shield-V2. Le dossier host_app est une application de capture et d'affichage de l'hôte pour tous les modules ArduCAM.

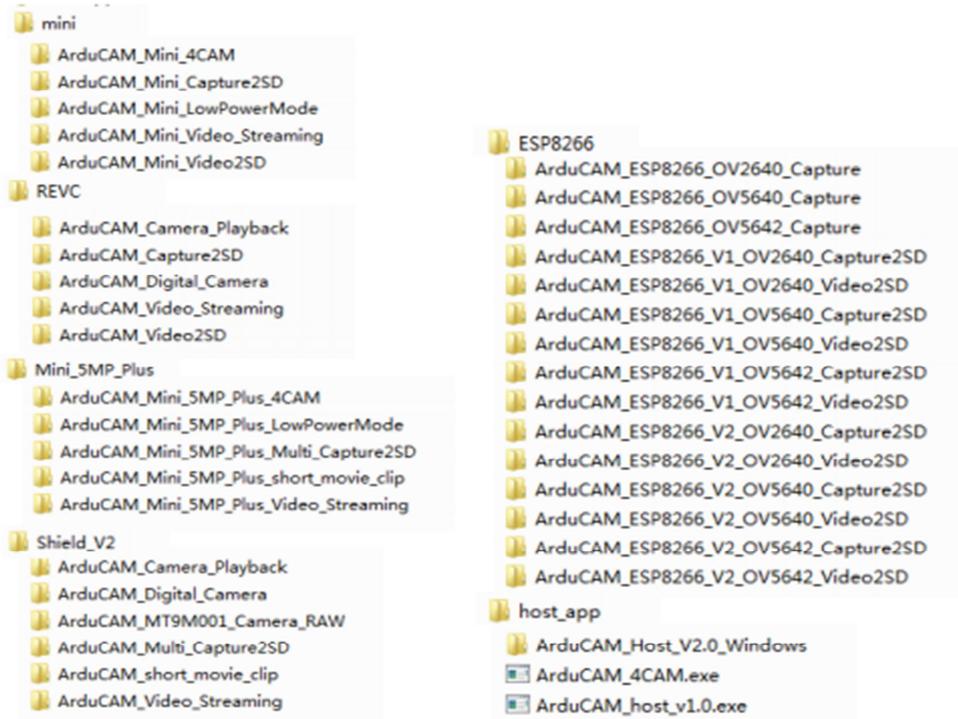


Figure III-11: Exemple de structure de dossier.

III-1-9 Montage de la maquette :

Le montage est simple, on utilise les deux composants (la carte Arduino UNO, et la camera Arducam), on raccorde l'appareil à la carte Arduino en utilisant les quatre broches numériques et les broches I2C .



Figure III-12: montage générale de la maquette.

III-1-10 Mode opératoire:

En 1^{er} étape on programme la carte arduino avec le langage IDE qui nous assure le fonctionnement souhaité de notre réalisation ; celle-ci consiste à capturer une image ensuite elle sera sauvegardée automatiquement.

III-1-11 Test et résultats obtenus :

- **Test de capture :**



Figure III-13: capture d'image.

Cette figure (9) illustre une capture d'image ; on constate lorsque la photo est prise, elle est automatiquement sauvegardée .

III-2 Caméra 2MP Arducam avec Module Nano ESP8266 :

Le kit d'évaluation est conçu pour une caméra IoT WIFI à bas prix basée sur Modules ArduCAM-Mini-2MP-V2 et ArduCAM-ESP8266-Nano. L'utilisateur peut mettre en œuvre un 2MPCaméra WIFI utilisant le protocole HTTP ou WebSocket sur ESP8266, téléphone portable / PC peuvent se connecter directement à l'appareil photo ou agir comme une station connectée au routeur domestique. Le kit peut prendre une image fixe JPEG de 2MP en pleine résolution, mais en streaming faible résolution vidéo basse fréquence en raison de la limitation de ESP8266. Le kit peut être alimenté par USB ou batterie alimentée par des circuits de charge buildin. Le kit peut également être utilisé séparément, il est identique à un appareil photo ArduCAM-Mini-2MP et un module ESP8266.

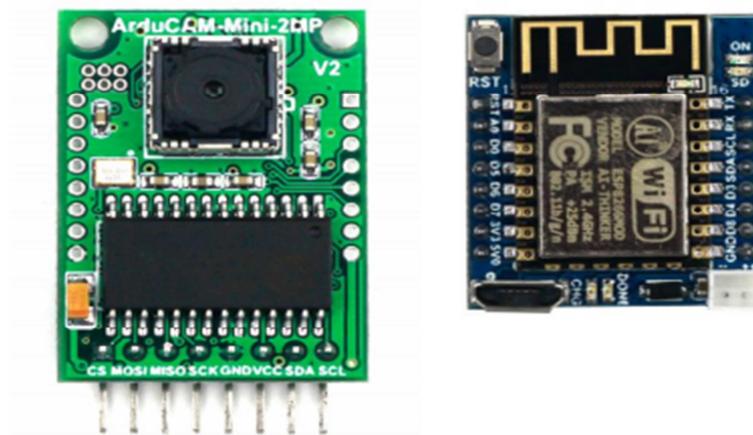


Figure III-14: ArduCAM-Mini-2MP-V2 et ESP8266-Nano.

III-2-1 Composition du kit:

- ArduCAM-Mini-2MP-V2
- ArduCAM-ESP8266-Nano
- Câble d'alimentation de la batterie

III-2-2 Caractéristiques techniques :

- Capteur d'image 2MP OV2640
- support JPEG Stock FOV standard à 60 °
- Interface I2C pour la configuration du capteur
- Interface SPI pour les commandes de la caméra et le flux de données
- Module ES8266-12F intégré
- Construire dans la batterie au lithium recharge 3.7V / 500mA max
- Construire dans une prise de carte SD / TF
- Construire dans le convertisseur micro USB-série (CH340g)
- Compatible avec l'IDE Arduino
- Petite forme de facteur

III-2-3 Constitution :

Il y a deux connecteurs sur le module de caméra ArduCAM-Mini-2MP, l'un est le connecteur à 8 broches sur le côté négatif appelé le connecteur ArduCAM standard, l'autre est le connecteur 16 broches double ligne est dédié connecteur pour ArduCAM-ESP8266-Nano module, ils peuvent être bien accouplés avec ESP8266 Nanomodule directement.

| N° PIN | Nom Pin | type | Description |
|--------|------------|-----------------|------------------------------------|
| 1 | RST | Input | ESP8266 reset input |
| 2 | A0 | Input | Analog input |
| 3 | D0/GPIO16 | Input | Chip select for camera |
| 4 | D5/SCK | Ground | Hardware SPI SCLK |
| 5 | D6/MISO | Input | Hardware SPI MISO |
| 6 | D7/MOSI | Output | Hardware SPI MOSI |
| 7 | 3.3V | Power | 3.3V Power supply |
| 8 | 5V | Power | 5V Power supply |
| 9 | GND | Ground | Power ground |
| 10 | D8/GPIO15 | Input | GPIO |
| 11 | D4/DPIO2 | Input | GPIO |
| 12 | D3/DPIO0 | Input | Chip select for SD card |
| 13 | SDA/GPIO04 | Bi-directionnel | Two-Wire Serial Interface Data I/O |
| 14 | SCL/GPIO05 | Output | Two-Wire Serial Interface Clock |
| 15 | RX/GPIO03 | Input | Hardware UART RX |
| 16 | TX/GPIO01 | output | Hardware UART TX |

Tableau III-2: ArduCAM-ESP8266-Nano Pin Définition.

III-2-4 Schéma de câblage:

La figure 2 montre le schéma de câblage entre l'ArduCAM mini 2MP et l'ESP8266 Nano module

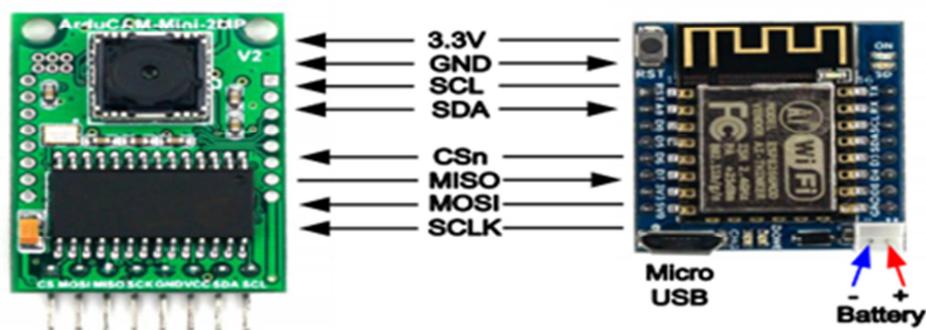


Figure III-15: schéma de câblage

III-2-5 Mise en marche ESP8266 avec IDE Arduino :

Dans cette partie nous allons développer une application pour ArduCAM ESP8266 Carte Nano utilisant IDE Arduino.

III-2-5-1 installation du gestionnaire de planche :

- Lancez Arduino et ouvrez la fenêtre Préférences.
- Entrez le lien suivant dans le champ URL supplémentaire du gestionnaire de forum. Vous pouvez ajouter plusieurs URL, en les séparant par des virgules.
http://www.arducam.com/downloads/ESP8266_UNO/package_ArduCAM_index.json
Ouvrir le gestionnaire des tableaux à partir du menu Outils -> Conseil et installer ArduCAM_ESP8266_UNO paquet d'adon.
- Notez que le fichier source du package installé se trouve dans C: \ Users \ Your computer nom \ AppData \ Local \ Arduino15 \ dossier, voici C: \ Utilisateurs \ zk109 \ AppData \ Local \ Arduino15 \ par exemple.

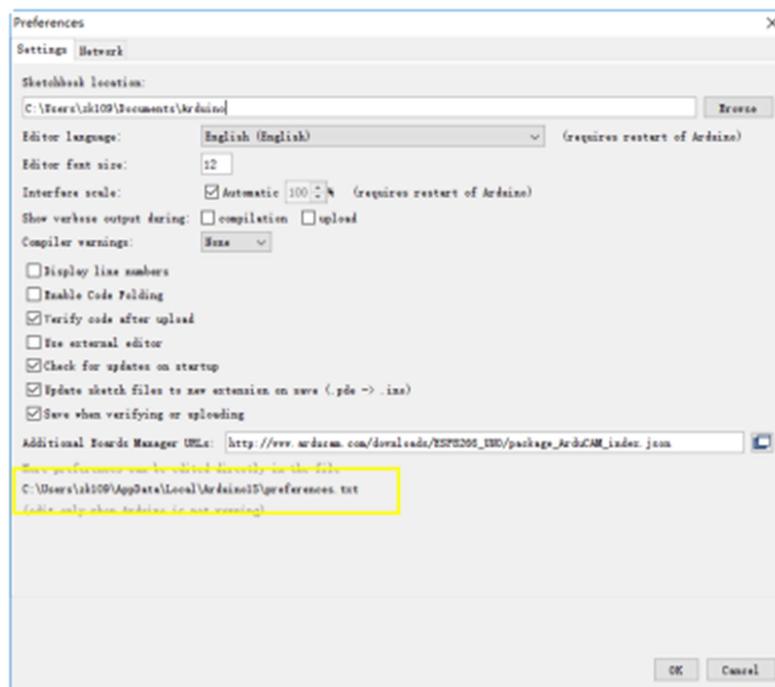


Figure III-16: Pack d'addonArduCAM ESP8266 UNO

III-2-5-2 Utilisation l'IDE Arduino :

Après l'installation du paquet additionnel ArduCAM ESP8266 Nano board, vous pouvez sélectionner cette carte dans le menu Tool->Board. Et là plusieurs prêts à l'emploi ; exemples du fichier-> Exemples->ArduCAM. Vous pouvez utiliser ces exemples directement ou comme point de départ pour développer votre propre code.

- Sélectionnez la carte ArduCAM_ESP8266_UNO dans le menu Tool->Board

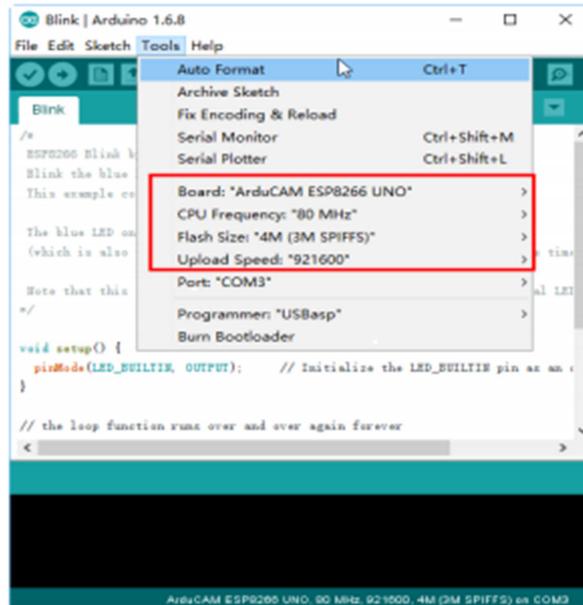


Figure III-17: sélection de planche.

- Sélectionnez l'exemple dans Fichier-> Exemples->ArduCAM.

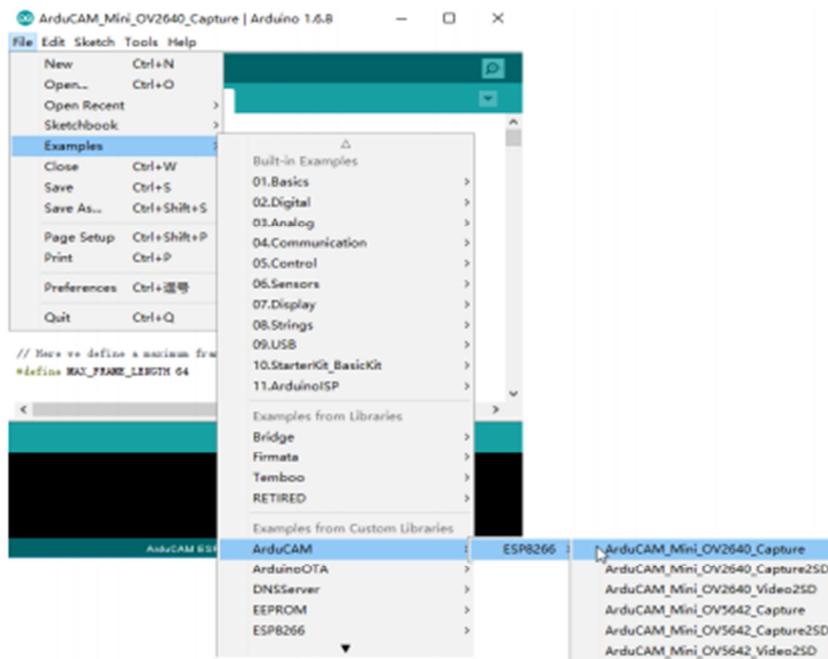


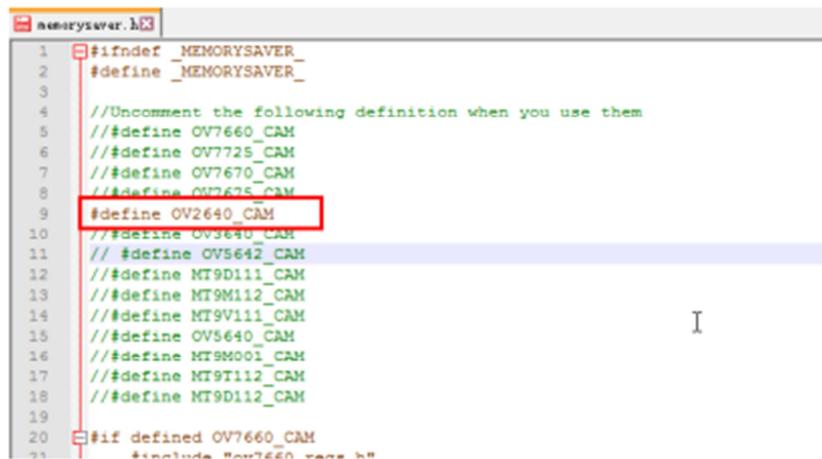
Figure III-18: selection d'exemple.

- **Configuration des paramètres de la caméra**

Vous devez modifier le fichier memorysaver.h pour activer l'appareil photo OV2640.

- Pour les modules d'appareil photo ArduCAM Mini 2MP ; une seule caméra peut être activée à un temps ; le fichier memorysaver.h se trouve sur C: \ Users \ votre ordinateur

nom \ AppData \ Local \ Arduino15 \ packages \ ArduCAM_ESP8266_UNO \ matériel
\ ArduCAM_ESP8266_UNO \ 2.1.0 \ libraries \ ArduCAM



```
1 #ifndef _MEMORYSAVER_
2 #define _MEMORYSAVER_
3
4 //Uncomment the following definition when you use them
5 // #define OV7660_CAM
6 // #define OV7725_CAM
7 // #define OV7670_CAM
8 // #define OV7675_CAM
9 #define OV2640_CAM
10 // #define OV3640_CAM
11 // #define OV5642_CAM
12 // #define MT9D111_CAM
13 // #define MT9M112_CAM
14 // #define MT9V111_CAM
15 // #define OV5640_CAM
16 // #define MT9M001_CAM
17 // #define MT9T112_CAM
18 // #define MT9D112_CAM
19
20 #if defined OV7660_CAM
21 #include "ov7660_pins.h"
```

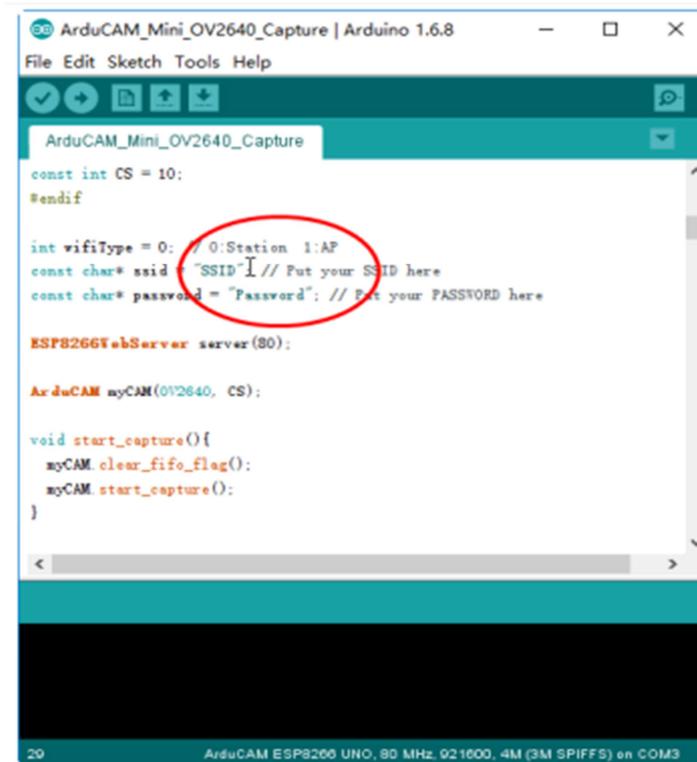
Figure III-19: configuration de la camera.

➤ Compiler et télécharger

Avant de compiler l'exemple changez le SSID et le mot de passe si nécessaire avec votre propre environnement réseau. Cliquez sur le téléchargement de l'exemple sera automatiquement flashé dans le tableau.

III-2-6 Exemples:

Il y a trois exemples pour les deux modules de caméra mini ArduCAM 2MP. ArduCAM_Mini_OV2640_Capture Cet exemple utilise le protocole HTTP pour capturer des images fixes ou vidéo sur le réseau Wi-Fi réseau de ArduCAM mini 2MP et afficher sur le navigateur Web. En utilisant cet exemple, le ssid et le mot de passe doivent être modifiés avant le téléchargement.



```

ArduCAM_Mini_OV2640_Capture
const int CS = 10;
#endif

int wifiType = 0; // 0:Station 1:AP
const char* ssid = "SSID"; // Put your SSID here
const char* password = "Password"; // Put your PASSWORD here

ESP8266WebServer server(80);

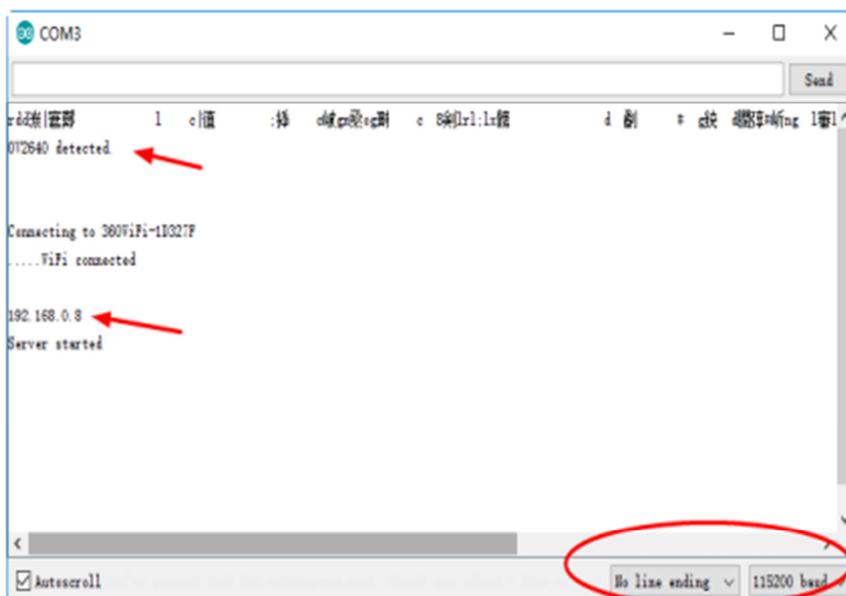
ArduCAM myCAM(OV2640, CS);

void start_capture(){
  myCAM.clear_fifo_flag();
  myCAM.start_capture();
}

```

Figure III-20: exemple camera wifi.

Après le téléchargement, l'adresse IP de la carte est obtenue via le protocole DHCP. Vous pouvez obtenir l'adresse IP à travers le moniteur série comme illustré à la figure 8. Le défaut le réglage du débit du moniteur série est de 115200 bps.



```

COM3
OV2640 detected
Connecting to 360WiFi-18021F
.....WiFi connected
192.168.0.8
Server started

```

Figure III-21: identification de l'adresse IP.

Enfin, ouvrez le fichier index.html ou video.html, entrez l'adresse IP obtenue à partir du moniteur série, puis prenez des photos ou des vidéos. Les fichiers html sont situés à C: \ Utilisateurs \ Nom de votre ordinateur \ Local \ Arduino15 \ packages \ ArduCAM_ESP8266_UNO \ matériel \ ArduCAM_ESP8266_UNO \ 2.1.0 \ libraries \ ArduCAM \ exemples \ ESP8266 \ ArduCAM_Mini_OV2640_Capture \ html

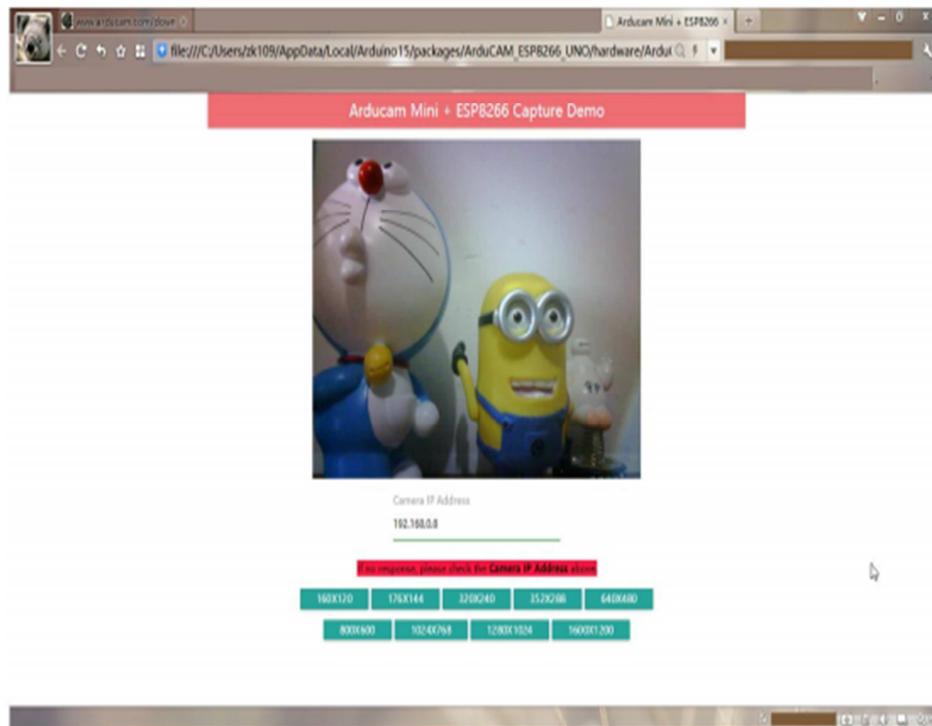


Figure III-22: exemple page HTML.

➤ ArduCAM_Mini_OV2640_Capture2SD

Cet exemple prend du temps à s'écouler des photos avec ArduCAM mini 2MP, puis stocké sur la carte TF / SD. La LED indique quand la carte TF / SD est en cours d'écriture.

➤ ArduCAM_Mini_OV2640_Video2SD

Cet exemple prend des clips vidéo JPEG en utilisant ArduCAM mini 2MP et puis stocké sur la carte TF / SD en format AVI, environ 4 minutes pour terminer.

Discussion

Dans ce chapitre, nous avons présenté et réalisé notre application, à savoir un système d'acquisition d'image. En effet, pour ce faire, vous avons utilisé une caméra OV2640 et une carte Arduino uno. Après un ensemble de tests que nous avons réalisés, nous avons constaté que notre application a bien fonctionné. Des images ont été collectées et enregistrées. L'acquisition de ces images peut être réalisée à des échelles de résolution différentes. Ces images peuvent être utilisées ensuite pour des traitements ultérieurs.

Conclusion :

L'objectif de ce travail est de réaliser un système d'acquisition des images performant et à faible cout à base de la carte arduino.

En élaborant ce mémoire nous avons étudié :

Dans le premier chapitre, notre étude s'est focalisée sur la présentation de la carte arduino, et ça commence par un bref historique et quelques détails concernant la création et l'utilité de cette carte. Puis on a présenté son côté matériel. La compréhension de donnée exposées dans ce chapitre nous a offert des bases de connaissances pour l'usage de la carte arduino .

Dans le deuxième chapitre nous avons constaté que L'acquisition d'images concerne plusieurs technologies, comme par exemple, l'optique ou les capteurs sur circuits intégrés, et s'appuie sur différentes normes. Nous avons donné l'état actuel de ces éléments, en apportant à chaque fois, les bases nécessaires à notre compréhension. Ce chapitre est, par conséquent, aussi un point de départ pour approfondir un domaine particulier.

Dans le troisième chapitre nous avons réalisé notre application en utilisant une carte arduino uno et une caméra module arducam de 2 méga pixels.

Selon le test que nous avons effectué, nous avons constaté que dès que la camera s'enclenche des captures d'images sont réalisées et transmises au PC pour être stockées dans un fichier destiné à cet effet. Par conséquent, ce résultat de test montre que notre application a bien fonctionné.

En perspective, il est intéressant d'intégrer une caméra dotée d'un module WIFI permettant de faire l'acquisition d'images en temps réel. A titre d'exemple, le module ESP8266 peut être utilisé pour la création d'un serveur pageweb.

Bibliographie

Ouvrages :

- [1]MONK Simon, 30 ARDUINO projects for the evilgenius : Me Graw Hi (2010)
- [2]BERTRAND Cotteceau, carte ARDUINO UNO microcontrolleur ARM328 : ISTIA (2010)
- [3]EVANS Brian, beginning Arduino programming: APRESS (2011)
- [4]BENZI Massimo, Getting Started with Arduino : ORELLY (2011)
- [5]MARION André, acquisition & visualisation des images : Eyrolles (1997)

Site internet :

- <http://thesis.univ-biskra.dz/2271/6/Chapitre%2003.pdf>
- http://www.ljk.imag.fr/Publications/Basilic/com.lmc.publi.PUBLI_Techreport@1176ddd04d5_11e735e/RT-0246.pdf
- https://mtlnumerique.uqam.ca/upload/files/presentation1_LeonRobinchaud_theorie.pdf
- https://www.researchgate.net/publication/259326499_Qu'est-ce_qu'une_image_numerique
- https://www.robotshop.com/media/files/pdf2/arducam_mini_2mp_camera_shield_ds.pdf
- https://www.robotshop.com/media/files/pdf2/arducam_mini_2mp_camera_shield_hardware_application_note.pdf
- https://www.robotshop.com/media/files/pdf2/arducam_camera_shield_software_application_note.pdf
- <https://www.robotshop.com/media/files/pdf/arducam-2mp-v2-mini-camera-shield-esp8266-nano-module-datasheet.pdf>