



RAPPORT DE PROJET FIN D'ETUDE

FACULTÉ SCIENCE ET TECHNIQUE BENI MELLAL

Réalisé par : SAMINE Hassan

Encadré par : Mr. MABROUKI Mustapha

SOMMAIRE

LISTE DES FIGURES	4
REMERCIEMENT.....	5
INTRODUCTION.....	6
I. HISTORIQUE	7
II. QU'EST-CE QUE C'EST UNE CARTE ARDUINO ?	7
III. APPLICATIONS DE LA CARTE ARDUINO.....	8
IV. PRESENTATION DE LA CARTE ARDUINO UNO.....	9
V. L'IDE ARDUINO.....	11
VI. LE LANGAGE ET LA SYNTAXE	12
A. <i>La structure d'un programme Arduino d'une manière générale.....</i>	<i>12</i>
B. <i>La structure d'un programme Arduino avec blockly Arduino.....</i>	<i>12</i>
VII. LE MODE DE LIAISON AVEC L'ORDINATEUR :	13
VIII. INTRODUCTION SUR LABVIEW :	14
IX. PROGRAMMER LA CARTE ARDUINO AVEC LE LOGICIEL LABVIEW :	14
A. <i>LES ETAPES POUR PROGRAMMER UNE CARTE ARDUINO AVEC LABVIEW :.....</i>	<i>16</i>
X. INTRODUCTION DE L'ARDUCAM :	18
XI. LES CARACTERISTIQUES DE L'OV7670 :	18
XII. LA CONNEXION ENTRE CAMERA ET ARDUINO :	19
A. <i>LES PINS DE L'OV7670 :.....</i>	<i>19</i>
B. <i>LE CABLAGE ENTRE LA CARTE ARDUINO ET LA CAMERA :</i>	<i>20</i>
XIII. LA CONNEXION AVEC L'ORDINATEUR :	20
XIV. ACQUISITION DES IMAGES ET DES VIDEOS EN LABVIEW	22
A. <i>AFFICHER LES IMAGES A L'AIDE DU LOGICIEL D'ACQUISITION DE VISION (VISION ACQUISITION SOFTWARE) :</i>	<i>22</i>
B. <i>STOCKER LES IMAGES DANS UN DOSSIER PUIS LES AFFICHER EN LABVIEW ...</i>	<i>24</i>
1. CAPTURE DES IMAGES PAR OV7670 :	24
2. AFFICHER LES IMAGES EN LABVIEW	29
C. <i>STOCKER LES VIDEOS DANS UN DOSSIER PUIS LES AFFICHER EN LABVIEW : .</i>	<i>30</i>
1. CAPTURE DES VIDEOS PAR OV7670 :	30
2. AFFICHER LES VIDEOS EN LABVIEW	32

LISTE DES FIGURES

Figure 1 : LE LOGO DE LA CARTE ARDUINO	7
Figure 2 : PLUSIEURS MODULE DE LA CARTE ARDUINO	8
Figure 3 : CARTE ARDUINO UNO	9
Figure 4 : INTERFACE DE L'IDE ARDUINO	11
Figure 5 : EXEMPLE DE PROGRAMME ARDUINO	12
Figure 6: EXEMPLE DE PROGRAMME ARDUINO AVEC BLOCKLY ARDUINO	13
Figure 7: PRINCIPE DU MONTAGE.....	13
Figure 8: INTERFACE DE LOGICIEL LABVIEW	14
Figure 9: EMBLEMEMENT DE VIs SOUS LABVIEW	15
Figure 10: L'INTERFACE DE LOGICIEL PACKAGE MANAGER	15
Figure 11: L'INSTALLATION DE L'INTERFACE ARDUINO SOUS LABVIEW	16
Figure 12: LE CODE ARDUINO LIFA BASE.....	17
Figure 13: ARDUINO SOUS LABVIEW	17
Figure 14: CAPTEUR D'IMAGE MODULE OV7670.....	18
Figure 15: LES PINS DE MODULE OV7670	19
Figure 16: CONNEXION DE L'ARDUINO ET L'OV7670	20
Figure 17: LES BIBLIOTHEQUES D'ARDUCAM.....	21
Figure 18: MODIFICATION DU MEMORYSAVER.H.....	21
Figure 19: PROGRAMMER L'ARDUCAM AVEC L'ARDUINO	22
Figure 20: "VISION AND MOTION" SOUS LABVIEW	23
Figure 21: LA FACE AVANT DU PROGRAMME 1	23
Figure 22: LE BLOC DIAGRAMME DU PROGRAMME 1	24
Figure 23: INSTALLATION DE LOGICIEL JAVA JDK	25
Figure 24: VERIFICATION DE NUMERO DE PORT	26
Figure 25: LE DOSSIER DE DESTINATION.....	26
Figure 26: LE FICHIER EXTRA.RAR	27
Figure 27: LE FICHIER CMD.....	28
Figure 28: LES IMAGES CAPTUREES PAR LA CAMERA	28
Figure 29: LA FACE D'AVANT DU PROGRAMME 2.....	29
Figure 30: LE BLOC DIAGRAMME DU PROGRAMME 2.....	29
Figure 31: REDIMENSIONNER L'IMAGE.....	30
Figure 32: INTERFACE DE LOGICIEL ARDUINO	31
Figure 33: LE BLOC DIAGRAMME DU PROGRAMME 3.....	32
Figure 34: LA FACE D'AVANT DU PROGRAMME 3.....	32

REMERCIEMENT

La première personne que je tiens à remercier est mon encadrant Mr. Mustapha MABROUKI, pour l'orientation, la confiance, la patience qui ont constitué un apport considérable. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Mes remerciements s'étendent également à tous les professeurs du département de physique de la Faculté des sciences et techniques, qui nous ont enseigné et formé dans ce cursus universitaire.

Enfin, je veux remercier tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

INTRODUCTION

Le travail de fin d'étude qui m'a été proposé est un projet très intéressant qui porte sur les nouvelles cartes types Arduino avec le logiciel LabVIEW en vue d'acquisition des images et des vidéos. En utilisant principalement ces deux outils on doit afficher des vidéos et des images.

Dans ce rapport on explique les différentes approches qu'on a essayées pour réussir cette tâche, le manuscrit est présenté comme suit : on commence tout d'abord par une présentation de la carte Arduino, ces applications, le logiciel et le langage utilisé pour la programmer et comment on peut la connecter avec l'ordinateur. Puis on introduit le logiciel LabVIEW. On a expliqué aussi comment on peut le connecter avec la carte Arduino à l'aide d'une interface Arduino spécial pour LabVIEW. Dans le chapitre suivant on définit le dernier outil nécessaire pour réaliser notre tâche l'ArduCAM, qui se connecte avec notre carte Arduino pour capturer des images et des vidéos. Enfin vient l'étape de programmation de la carte Arduino avec notre caméra, et la programmation en LabVIEW pour pouvoir acquérir et afficher les images et les vidéos.

I. HISTORIQUE DE LA CARTE ARDUINO

L'année 2005 a été marquée par l'invention de l'Arduino, un projet créé par une équipe italienne de l'Institut de conception et d'Interaction d'Ivrea qui se compose de Massimo Banzi, David Cuartielles, Nicholas Zambetti et David Mellis, rejoint après par Tom Igoe et Gianluca Martino, dont le but est de réaliser tout type de projets grâce à une petite carte électronique facilement programmable via un logiciel qui s'appelle ARDUINO IDE. A cette carte, on peut brancher des LEDs, des moteurs, des capteurs, des éléments électroniques et plusieurs autres choses dans le but de créer des projets.

D'ailleurs, l'un de ces avantages les plus importants est le fait qu'elle est une carte Open Source : c'est-à-dire que vous êtes libre de la modifier et créer votre propre carte copie conforme. Il faut seulement protéger et ne pas reproduire le nom Arduino et le logo de la marque.



Figure 1 : LE LOGO DE LA CARTE ARDUINO

II. QU'EST-CE QUE C'EST UNE CARTE ARDUINO ?

Arduino est une petite carte électronique de 74 x 53 x 15 mm à la base d'un microcontrôleur: c'est un circuit programmable intégré qui est capable de recevoir et de traiter de l'information qui vient par exemple d'un capteur par exemple, et en fonction de cette information, le circuit intégré va commander autre circuit électronique comme un moteur un écran, ...etc.

Il y a dix ans programmer un microcontrôleur était assez compliqué, très coûteux, et consommateur de temps.

Pour cette raison que des enseignants de l'institut de conception et d'intégration d'Ivrea ont proposé à leurs étudiants de développer une carte électronique qui permettant de simplifier cette programmation.

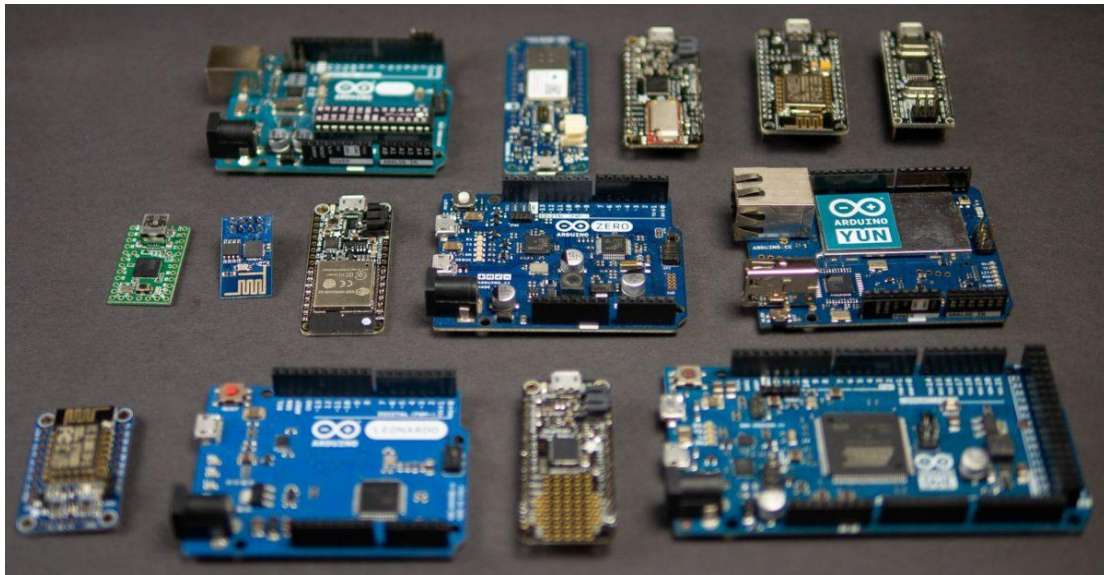


Figure 2 : PLUSIEURS MODULE DE LA CARTE ARDUINO

Parmi les avantages de la carte Arduino :

- Elle est accessible à tout le monde et facile à utiliser
- Elle utilise une interface de programmation très simplifiée.
- Il suffit d'un ordinateur, d'un câble USB, et d'un minimum de connaissance dans la programmation pour commencer et à réaliser des projets. Vous pourrez écrire votre programme, le compiler, puis le charger sur votre carte Arduino.
- Il existe plusieurs modules préprogrammés servant différentes actions qu'on peut ajouter à la carte Arduino
- Il coute un prix abordable avoisinant les 250 Dh.

III. APPLICATIONS DE LA CARTE ARDUINO

Les possibilités de son utilisation sont infinies. On peut détecter plusieurs événements, par exemple : des variations de température, mouvement, distance, présence ... etc. Et en fonction de ces événements, on agit sur le monde réel à l'aide de la carte Arduino.

La carte Arduino la plus utilisée est la carte Arduino UNO. Avec cette carte, il est possible de programmer plusieurs systèmes. Soit des systèmes simples de clignotement d'une led, ou des systèmes plus compliqués d'arrosage automatique pour vos plantes, ou encore un servomoteur qui sert à faire tourner une hélice.

Pour cela, il faut effectuer quelques branchements électroniques de votre carte à votre matériel, grâce à des composants obtenus dans les kits de démarrage Arduino.

Comme vous pouvez le comprendre, la carte Arduino Uno permet de programmer plusieurs projets divers, il est possible de toucher à plusieurs domaines par exemple l'électricité, la domotique, micro robotique, ou systèmes mécaniques ... etc.

IV. PRESENTATION DE LA CARTE ARDUINO UNO

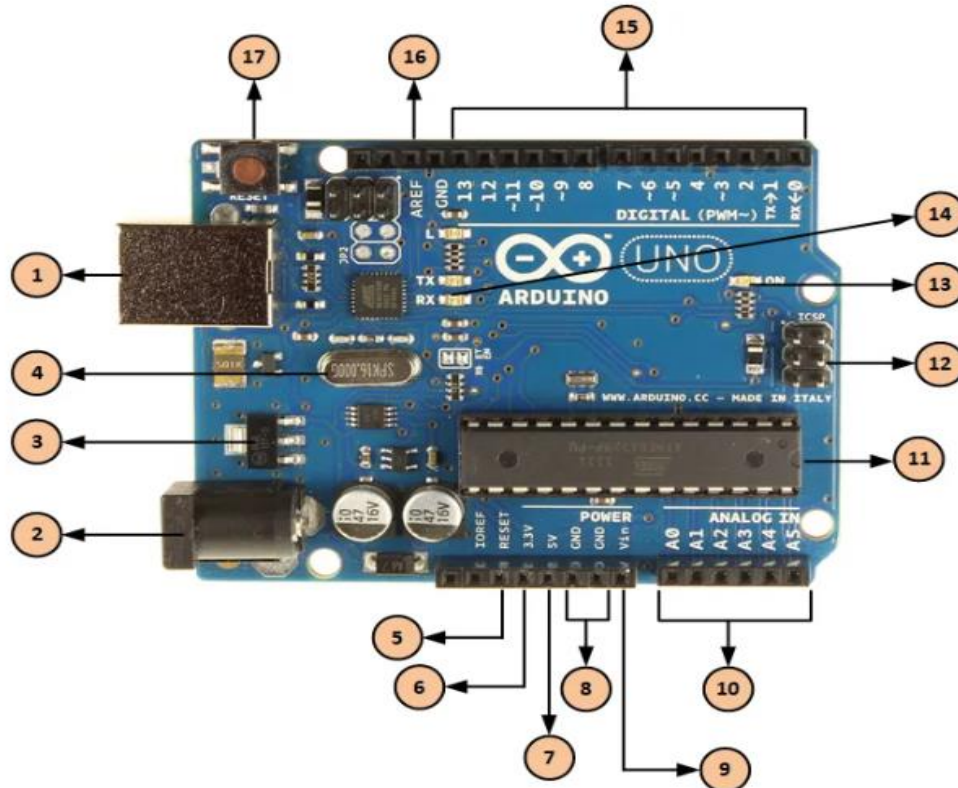


Figure 3 : CARTE ARDUINO UNO

La carte Arduino Uno est la plus simple et la plus économique carte à microcontrôleur d'Arduino, vous trouvez ci-dessous une définition des différents pins de ce module :

1 : Alimentation / Programmation par USB : La carte Arduino peut être alimentée avec un câble USB relié à votre ordinateur. Tout ce dont vous avez besoin, c'est de connecter votre carte Arduino à votre ordinateur avec le câble USB type A/B.

2 : Prise jack : La carte Arduino peut être directement alimentée par ce connecteur Jack DC. Ce connecteur (2) est relié au régulateur de tension intégré à la carte. L'alimentation via ce connecteur (2) doit être comprise entre 5 et 12 V.

3 : Régulateur de tension : La fonction du régulateur de tension (3) est de contrôler la tension d'alimentation de l'Arduino pour la stabiliser à la bonne tension du microcontrôleur et de chaque élément de la carte. La tension de stabilisation est de 5 Volts sur les cartes UNO.

4 : Oscillateur à quartz : Un oscillateur à quartz est un élément électronique qui a la particularité de posséder un quartz à l'intérieur qui vibre sous l'effet piézoélectrique. Les propriétés électromécaniques du quartz sont telles qu'on arrive à faire vibrer le quartz à une fréquence très précise. Cet élément aide l'Arduino UNO à calculer les

données de temps. Sur le dessus du composant, on peut lire 16.000H9H. Cela signifie que la fréquence est de 16 MHz.

5, 17: Arduino Reset : Vous pouvez redémarrer un Arduino avec un “Reset”. Cela aura pour effet de redémarrer votre programme depuis le début. Vous pouvez redémarrer l’Arduino UNO de deux manières : soit en utilisant le bouton “Reset” (17), soit en connectant un bouton externe sur la broche de la carte Arduino mentionnée “RESET” (5).

6, 7, 8, 9 : Broches (3.3, 5, GND, Vin) :

- 3.3V (6) – Broche d’alimentation de tension 3.3 Volts
- 5V (7) – Broche d’alimentation de tension 5 Volts
- La plupart des composants destinés à fonctionner avec Arduino fonctionnent bien en 3.3 Volts ou 5 Volts.
- GND (8) (la masse) – Il y a plusieurs broches de ce type présentes sur la carte Arduino, elles sont toutes communes et peuvent être utilisées comme masse (potentiel 0 Volts) pour vos circuits.
- Vin (9) – Cette broche permet d’alimenter l’Arduino depuis une source de tension extérieure. Elle est reliée au circuit d’alimentation principale de la carte Arduino.

10 : Broches analogiques : L’Arduino UNO possède 5 broches d’entrées analogiques numérotée de A0 jusqu’à A5. Ces broches permettent de brancher des capteurs et des détecteurs analogiques, Ces six entrées peuvent admettre 1024 valeurs analogiques comprises entre 0 et 5 Volts.

11 : Microcontrôleur principal : Chaque carte Arduino possède son propre microcontrôleur (11). Vous pouvez le considérer comme le cerveau de cette carte, il Permet de stocker le programme et de l’exécuter

12 : Connecteur ICSP (In-Circuit Serial Programming) : Permet à l’Arduino de communiquer avec des composants externes via le protocole SPI (Serial Peripheral Interface)

13 : Indicateur LED d’alimentation : Ce voyant doit s’allumer lorsque vous branchez votre Arduino sur une source d’alimentation pour indiquer que votre carte est correctement alimentée.

14 : LEDs TX et RX : Sur la carte, vous trouverez deux indicateurs : TX (émission) et RX (réception). Ils indiquent les broches responsables de la communication série. Le voyant TX clignote à une vitesse variable lors de l’envoi des données série et RX clignote pendant le processus de réception.

15 : 14 entrées et sorties numériques : Permet de brancher des actionneurs ou des détecteurs

16 : AREF : signifie référence analogique. Il est parfois utilisé pour régler une tension de référence externe (entre 0 et 5 volts) comme limite supérieure pour les broches d’entrée analogiques.

V. L'IDE ARDUINO

Le logiciel qu'on utilise pour programmer la carte Arduino s'appelle ARDUINO IDE, un logiciel simple et gratuit qu'on peut télécharger sur le site officiel d'Arduino.

L'IDE Arduino permet :

- d'éditer un programme : des croquis (sketch en Anglais),
- de compiler ce programme dans le langage « machine » de l'Arduino,
- de télé verser le programme dans la mémoire de l'Arduino,
- de communiquer avec la carte Arduino grâce au terminal.



Figure 4 : INTERFACE DE L'IDE ARDUINO

MENU : Il contient les boutons qu'on va utiliser lorsqu'on va programmer nos cartes, les deux premiers boutons dans ce cadre sert à vérifier le programme et le télé verser sur la carte.

Fenêtre de programmation : ce bloc va contenir le programme que nous allons créer

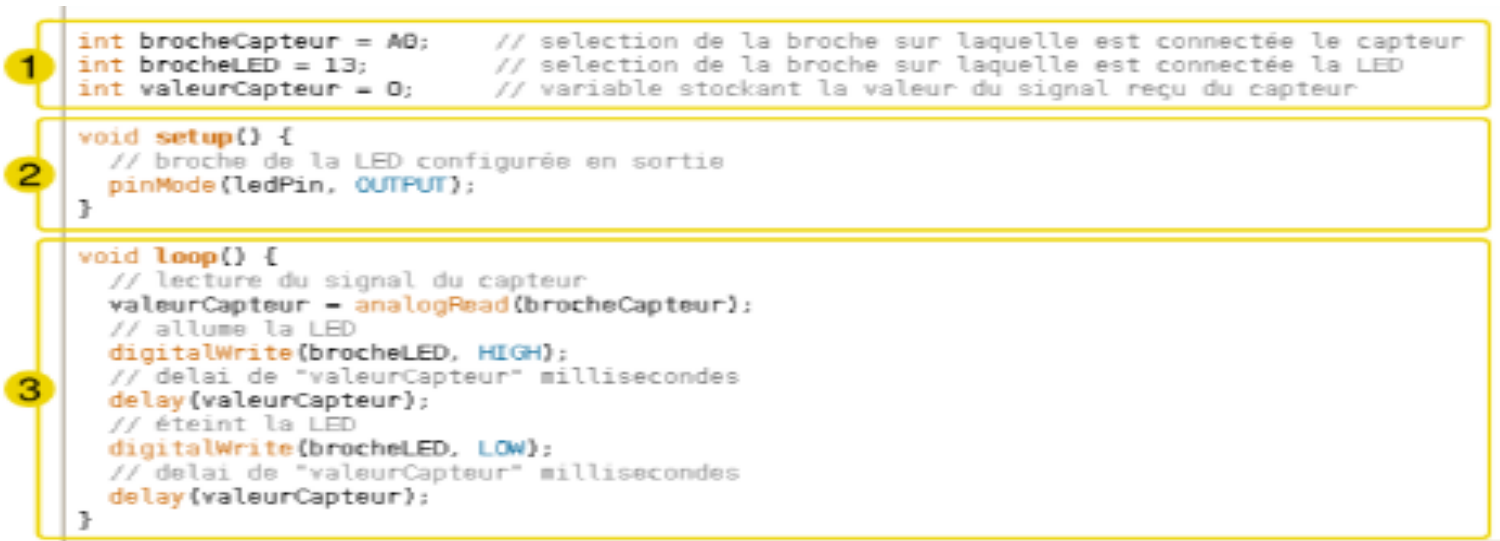
Fenêtre de console : ce dernier bloc est important car il va nous aider à corriger les fautes dans notre programme

VI. LE LANGAGE ET LA SYNTAXE

A. La structure d'un programme Arduino d'une manière générale

Un langage de programmation est un ensemble d'instructions écrit par l'être humain pour communiquer avec une machine quelconque grâce à un compilateur. Le langage utilisé pour communiquer avec la carte Arduino est un programme proche de programme C, et il est ce qu'on appelle un programme séquentiel : ça veut dire que les instructions sont exécutées l'une après l'autre.

Un programme Arduino comporte trois parties :



```
1 int brocheCapteur = A0;    // selection de la broche sur laquelle est connectée le capteur
  int brocheLED = 13;       // selection de la broche sur laquelle est connectée la LED
  int valeurCapteur = 0;    // variable stockant la valeur du signal reçu du capteur

2 void setup() {
  // broche de la LED configurée en sortie
  pinMode(ledPin, OUTPUT);
}

3 void loop() {
  // lecture du signal du capteur
  valeurCapteur = analogRead(brocheCapteur);
  // allume la LED
  digitalWrite(brocheLED, HIGH);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
  // éteint la LED
  digitalWrite(brocheLED, LOW);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
}
```

Figure 5 : EXEMPLE DE PROGRAMME ARDUINO

1 : La partie déclaration de variable: on trouve dans cette partie la déclaration des variables qu'on va utiliser dans le programme. C'est une partie optionnelle.

2 : La partie initialisation et configuration des entrées/sorties : la fonction setup () : Cette fonction est la partie où on écrit le code qui ne s'exécute qu'une seule fois lorsque le programme commence et quand le bouton reset est pressé. On trouve dans cette partie les déclarations des entrées et de sorties des autres réglages.

3 : La partie principale: la fonction loop () : Cette fonction est la partie principale qui contenu le programme et il s'exécute en loop, ça veut dire on le ré exécute chaque fois il est terminé encore et encore

B. La structure d'un programme Arduino avec blockly Arduino

La logique de programmation par bloc : Cette logique de programmation s'appelle OPENBLOCK, il était développée par le MIT (Institut de technologie du Massachusetts). Cette méthode reprend graphiquement la structure des langages de programmations type C. Le principe est simple, chaque type de commande possède une

forme (et une couleur) et on peut les rassembler en respectant les formes d'assemblage. Les ordres peuvent être donnés à la suite. Comme pour un programme en langage syntaxique.

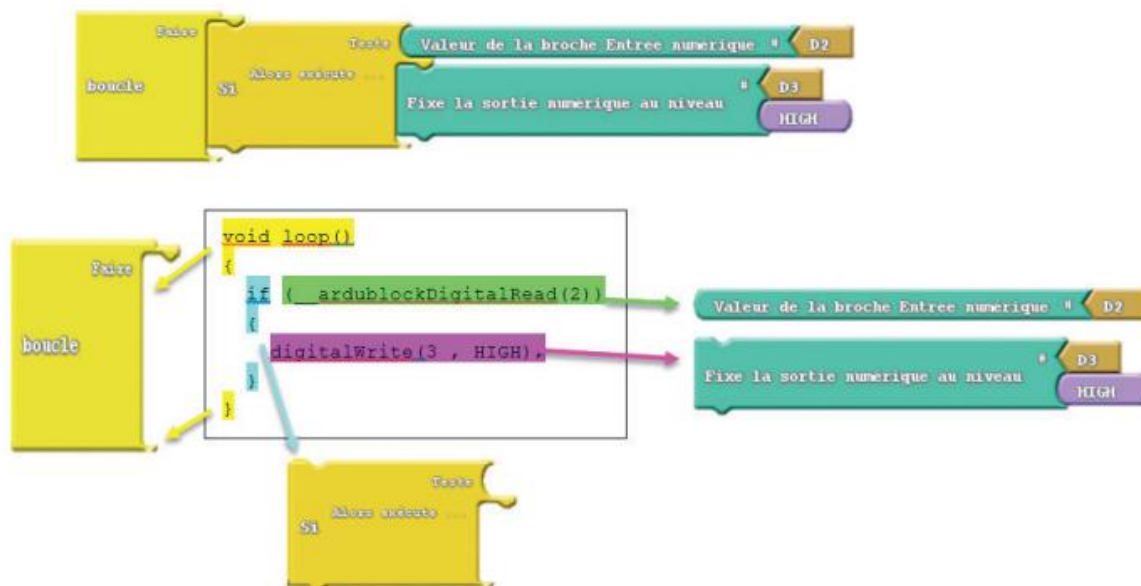


Figure 6: EXEMPLE DE PROGRAMME ARDUINO AVEC BLOCKLY ARDUINO

VII. LE MODE DE LIAISON AVEC L'ORDINATEUR :

La communication entre le PC et la carte Arduino se fait à l'aide d'un protocole de communication (une spécification de plusieurs règles pour un type de communication particulier) qui s'appelle la liaison série : un mode de transmission bit à bit des données sur deux conducteurs l'un RX (réception) et l'autre TX (Transmission). Le débit est l'unité de mesure de débit qui correspond au nombre de bits transmis par seconde, et c'est l'essentiel à savoir de la communication série car il peut être différent d'un périphérique à un autre.

- **Réalisation d'une liaison série entre le PC et l'Arduino :**

Ce montage ne comporte aucun composant externe, il suffit de câbler l'Arduino avec le PC via le câble USB.



Figure 7: PRINCIPE DU MONTAGE

VIII. INTRODUCTION SUR LABVIEW :

LabVIEW est un logiciel de programmation créé par National Instruments en 1986. Sa particularité est qu'il spécialise dans la programmation graphique (le langage G) qui facilitent la complexité de programmation et donc diminuer les délais de développement. Ce logiciel se compose de deux fenêtres ce qui permet de réaliser et exécuter rapidement les programmes, ces deux fenêtres sont la face avant qui sert d'interface avec l'utilisateur et le bloc diagramme qui contient le programme source en langage graphique G.

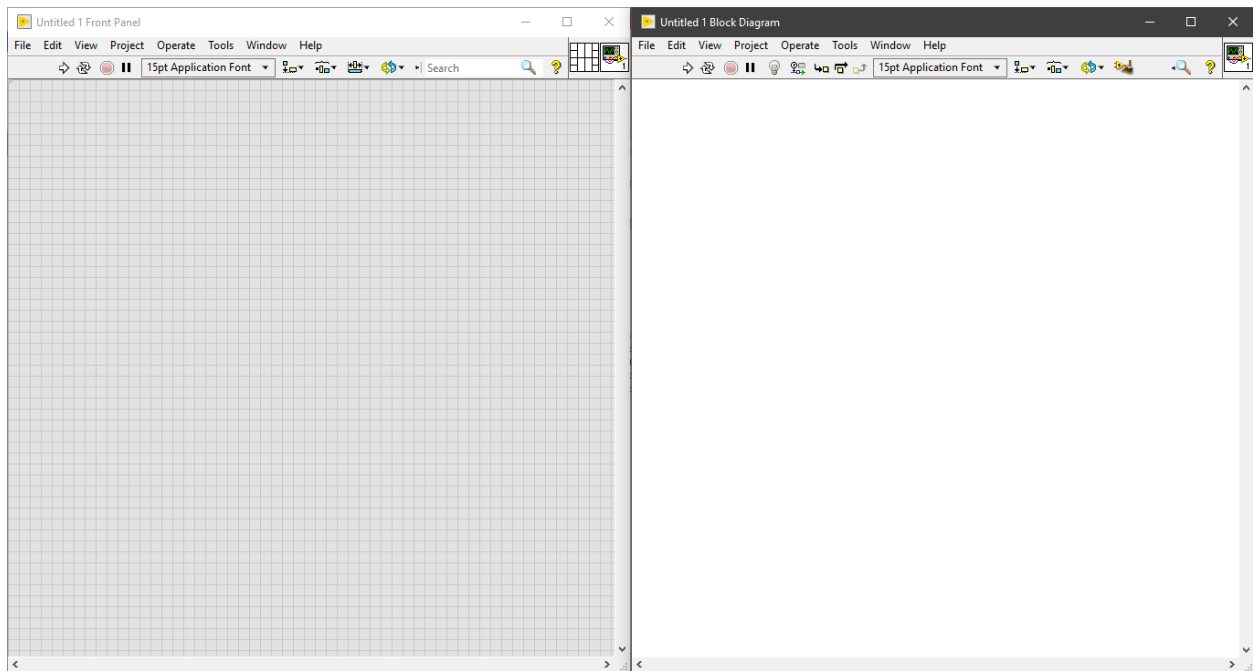


Figure 8: INTERFACE DE LOGICIEL LABVIEW

IX. PROGRAMMER LA CARTE ARDUINO AVEC LE LOGICIEL LABVIEW :

Pour qu'on puisse programmer la carte Arduino utilisant le logiciel LabVIEW il faut d'abord avoir installé les quatre outils suivants :

- LABVIEW
- ARDUINO IDE
- VISA NI
- VI PACKAGE MANAGER

On a déjà parlé des deux premiers logiciels précédemment LabVIEW et Arduino IDE.

VISA NI : C'est un outil qui permet au logiciel LabVIEW de communiquer avec la carte Arduino, et de savoir dans quelle port exactement la carte est liée avec l'ordinateur. C'est un logiciel gratuit qu'on peut trouver dans le site officiel de NATIONAL INSTRUMENTS, après l'installation on doit le mettre sous le dossier National instruments du LabVIEW

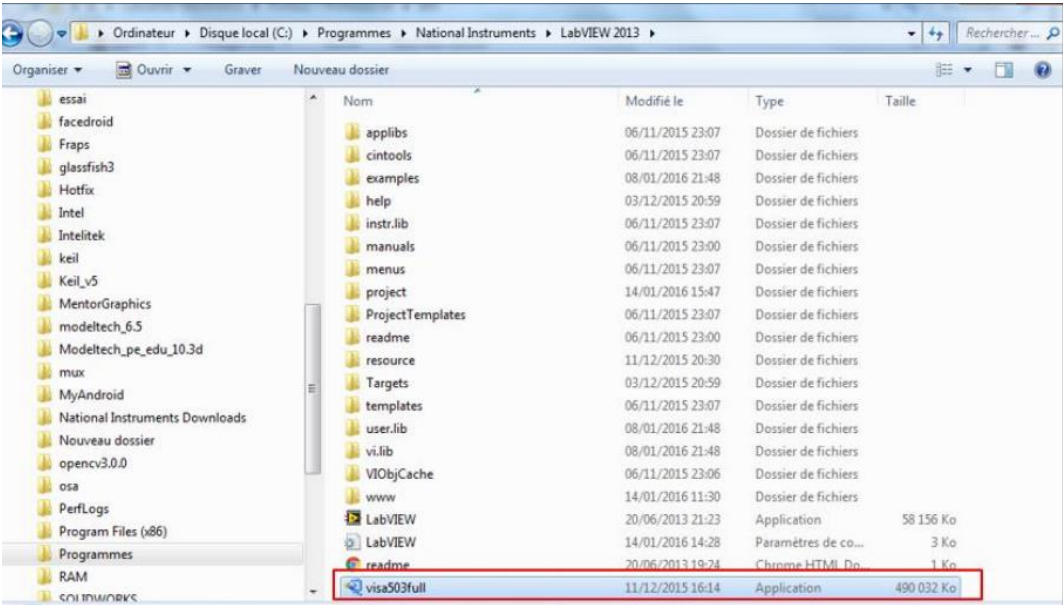


Figure 9: EMLACEMENT DE VIs SOUS LABVIEW

VI PACKAGE MANAGER : est un logiciel qui organise et gère des paquets pour le logiciel LabVIEW, il est l'outil pour obtenir des bibliothèques. On va l'utiliser pour installer l'interface Arduino sous LabVIEW.

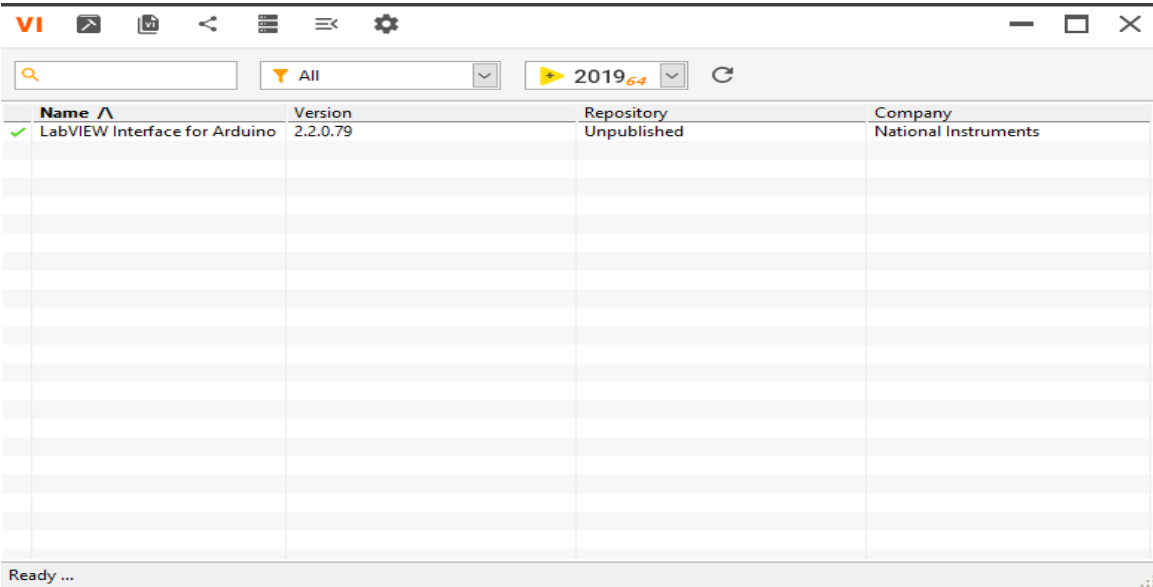


Figure 10: L'INTERFACE DE LOGICIEL PACKAGE MANAGER

A. LES ETAPES POUR PROGRAMMER UNE CARTE ARDUINO AVEC LABVIEW :

Premièrement les types de carte Arduino qui peut être programmé avec le logiciel LabVIEW sont la carte Arduino Uno et la carte Arduino Méga.

La carte Arduino doit être connectée à l'ordinateur à l'aide d'un câble USB pour la programmer.

Puis on utilise logiciel VI PACKAGE MANAGER pour installer la bibliothèque "INTERFACE ARDUINO sous LABVIEW"

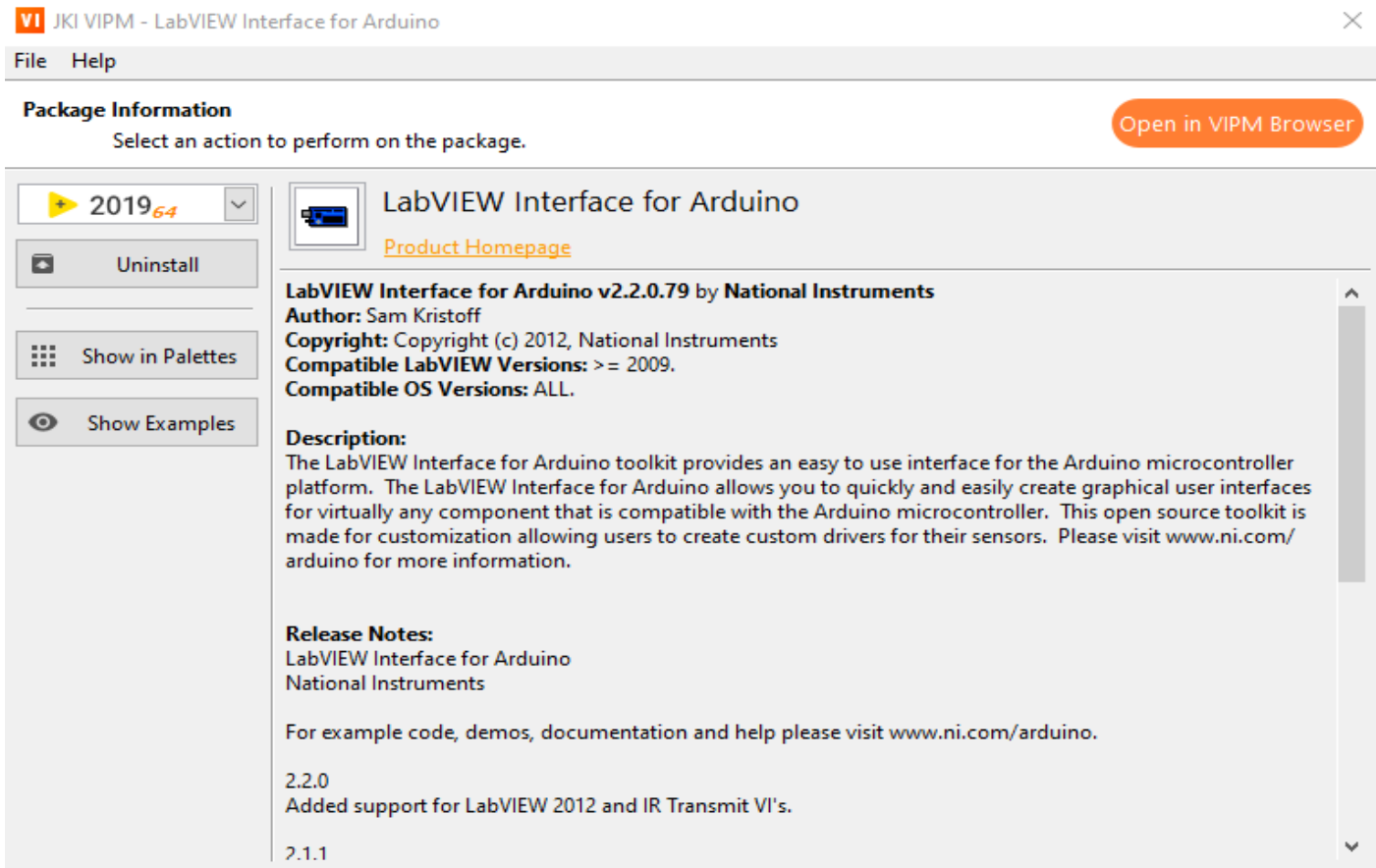


Figure 11:L'INSTALLATION DE L'INTERFACE ARDUINO SOUS LABVIEW

Après l'installation de la bibliothèque un dossier sous le nom "INTERFACE ARDUINO SOUS LABVIEW" s'ajoute dans le dossier vi.lib où se trouvent les bibliothèques de logiciel LabVIEW. Au sein de ce document se trouve un programme Arduino sous le nom LIFA_Base, qu'on doit compiler sur la carte Arduino liée à l'ordinateur. Ce code assure la communication entre la carte Arduino et le logiciel LabVIEW.

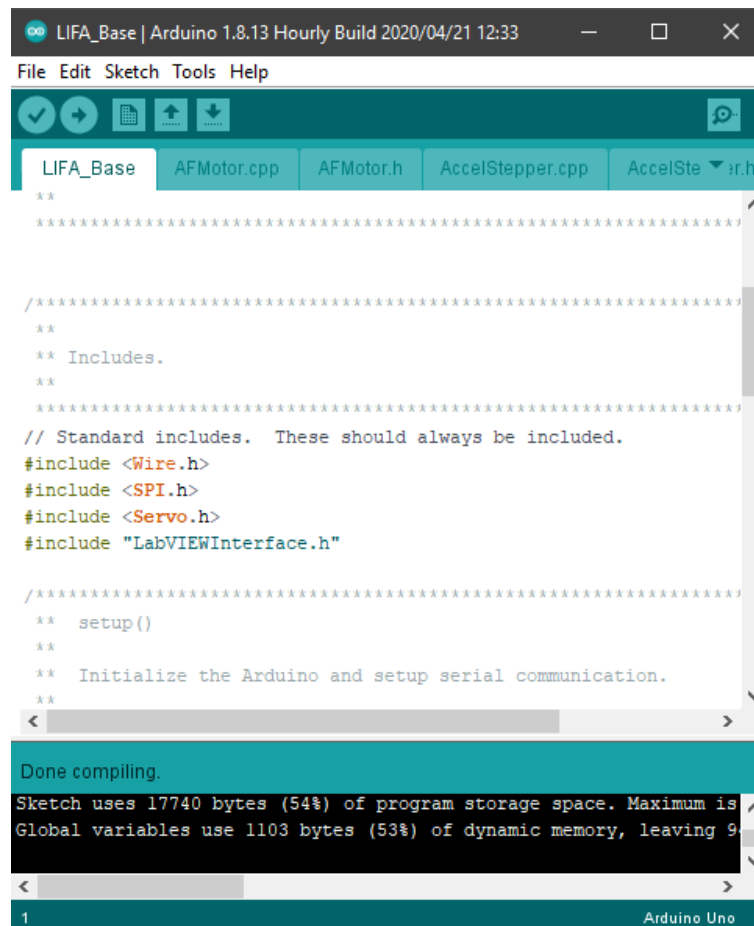


Figure 12: LE CODE ARDUINO LIFA BASE

L'installation de la bibliothèque "INTERFACE ARDUINO SOUS LABVIEW" agit aussi sur le logiciel LabVIEW, car quand on lance le logiciel on trouve une fenêtre qui contient les composants qui définissent Arduino sous LabVIEW en cliquant cliquer droit et chercher le mot Arduino dans la palette de fonctions, à l'aide de cette fenêtre on peut programmer la carte Arduino sous le logiciel LabVIEW.

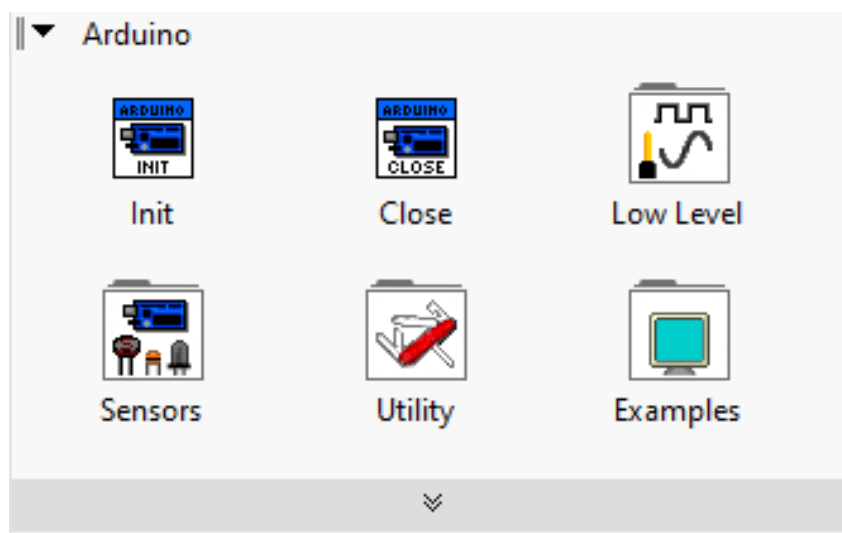


Figure 13: ARDUINO SOUS LABVIEW

X. INTRODUCTION DE L'ARDUCAM :

Dans cette étape du projet on devait connecter la carte Arduino avec une caméra qui va capturer des images ou vidéo et les envoyer à notre carte Arduino. J'ai choisis d'utiliser une ArduCAM de module **OV7670** car il possède plusieurs avantages, il est d'une petite taille, une faible tension de fonctionnement et il est possible de paramétrer la qualité des images, le format des données et le mode de transmission, et il peut facilement être interfacée sur une carte Arduino, en plus il ne coûte pas très cher.



**Figure 14: CAPTEUR D'IMAGE MODULE
OV7670**

XI. LES CARACTERISTIQUES DE L'OV7670 :

Taille : 3,5cm x 3,5 cm

Taille de la lentille : 1/6"

Images par secondes : 30fps

Dimensions : 640x480 px

Tension de fonctionnement : 2,5 à 3 volts

Champ de vision : 25°

Formats supportées : YUV YCbCr 4/2/2 / RGB565 / RGB555 / GRB 4/2/2 / Raw RGB Data

Haute sensibilité à faible luminosité

Basse tension appropriée pour les applications embarquées

XII. LA CONNEXION ENTRE CAMERA ET ARDUINO :

Même si la carte Arduino dispose d'une prise USB on ne peut pas utiliser une webcam USB car elle ne fait pas hôte, et la carte Arduino ne permet pas de piloter un périphérique USB. Puisqu'on a besoin d'une caméra avec un signal analogique et un flux de vidéo composite pour chaque couleur la tâche sera plus difficile. Les caméras qu'on peut utiliser pour cette étape sont celles qui mettent à disposition chaque image dans un format numérique via une interface série ou parallèle. Comme déjà cité j'ai choisis de travailler avec le module OV7670 qui dispose de ce qu'on a besoin.

A. LES PINS DE L'OV7670 :

OV7670		PIN	DESCRIPTION
3v3		3V3	Alimentation 3.3 V
GND		GND	La masse d'alimentation
SIOC		SIOC	Horloge de l'interface série SCCB
SIOD		SIOD	Données I/O de l'interface SCCB
VSYN		VSYN	Sortie de synchronisation vertical
HREF		HREF	Sortie de hypertext reference
PCLK		PCLK	Sortie de l'horloge de pixel
XCLK		XCLK	Entrée de l'horloge de système
D7		D0 à D7	Sortie des données de pixel
D6			
D5			
D4			
D3			
D2			
D1		RESET	Réinitialisation
D0		PWDN	Sélection du mode de mise hors tension
RESET			
PWDN			

Figure 15: LES PINS DE MODULE OV7670

B. LE CABLAGE ENTRE LA CARTE ARDUINO ET LA CAMERA :

La figure ci-dessous présente le câblage utilisé pour connecter la caméra OV7670 à l'Arduino :

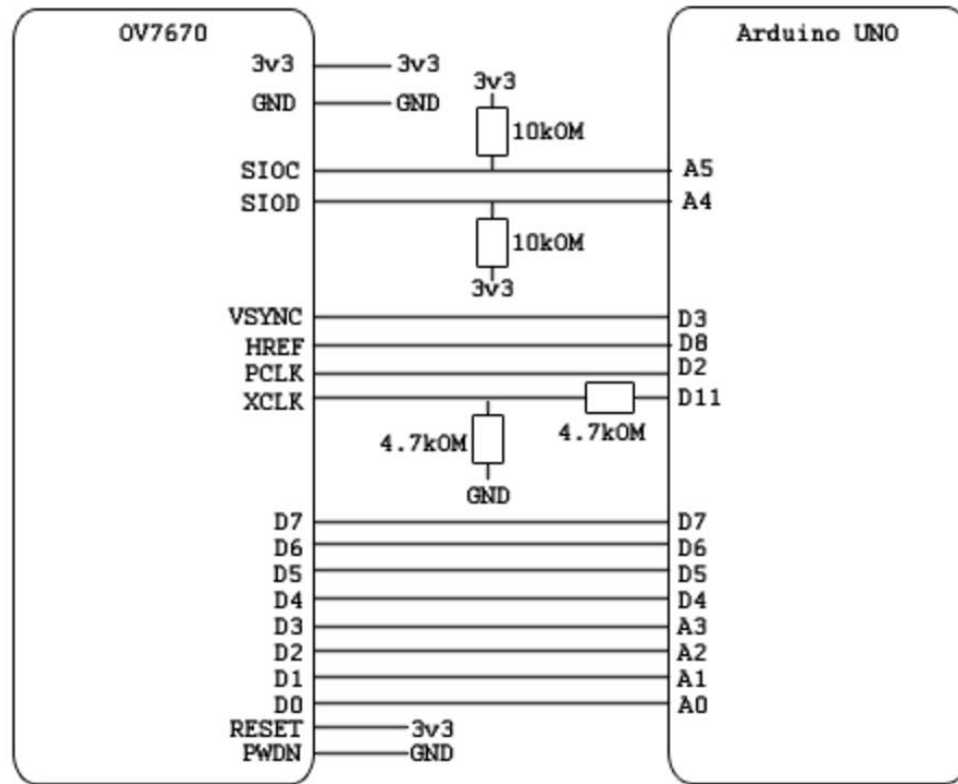


Figure 16: CONNEXION DE L'ARDUINO ET L'OV7670

Le principe de fonctionnement de ce montage est l'Arduino demande l'acquisition de l'image ou vidéo à la camera et via le port USB en Arduino il envoie cette information au PC, où on va essayer d'afficher l'image recueillie sur le logiciel LabView. Avant tout, il faut compiler le code d'ArduCAM sur la carte Arduino.

XIII. LA CONNEXION AVEC L'ORDINATEUR :

Le module OV7670 qu'on utilise dans ce projet appartient à la famille d'ArduCAM ; un projet open source assez simple pour être facilement utilisé avec un Arduino, il est utilisé dans plusieurs applications comme les caméras de robot, caméras scientifiques etc. il peut aussi s'intégrer avec d'autres cartes comme le Raspberry Pi et le Beaglebone noir. Cela signifie que pour programmer notre OV7670 il faut ajouter la bibliothèque ArduCAM dans la plateforme Arduino, qui se compose de quatre sous-bibliothèques qu'il faut télécharger et ajouter dans les bibliothèques du logiciel d'Arduino IDE.

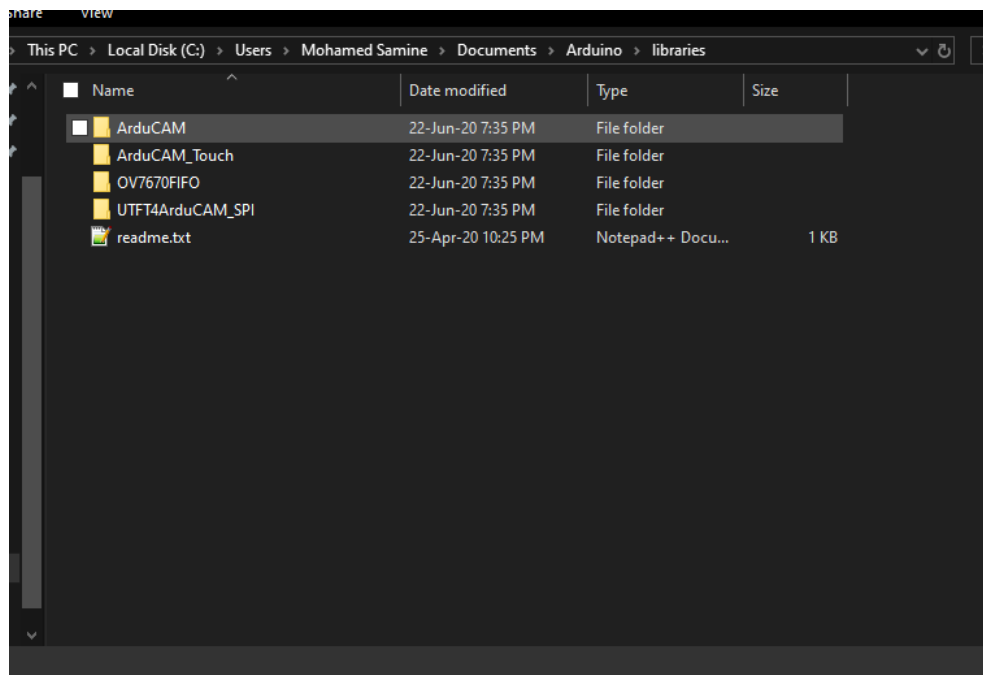


Figure 17: LES BIBLIOTHEQUES D'ARDUCAM

Les bibliothèques de base sont composées de quatre sous bibliothèques; ArduCAM, ArduCAM_touch, OV7670FIFO et UTFT4ArduCAM_SPI. Pour que le programme se compile correctement il faut modifier le fichier memorysaver.h qui se trouve dans le dossier ArduCAM et activer le module de caméra qui correspond. Par exemple on utilise le module OV7670 il faut décommenter la ligne qui définit ce module dans le fichier.

```
ArduCAM-Mini series platform doesn't

//Step 1: select the hardware platform, only one at a time
//#define OV2640_MINI_2MP
//#define OV3640_MINI_3MP
//#define OV5642_MINI_5MP
//#define OV5642_MINI_5MP_BIT_ROTATION_FIXED
//#define OV2640_MINI_2MP_PLUS
//#define OV5642_MINI_5MP_PLUS
//#define OV5640_MINI_5MP_PLUS

.

//#define ARDUCAM_SHIELD_REVC
#define ARDUCAM_SHIELD_V2

//Step 2: Select one of the camera module, only one at a time
#if (defined(ARDUCAM_SHIELD_REVC) || defined(ARDUCAM_SHIELD_V2))
    //#define OV7660_CAM
    //#define OV7725_CAM
    #define OV7670_CAM
    //#define OV7675_CAM
    //#define OV2640_CAM
    //#define OV3640_CAM
    //#define OV5642_CAM
    //#define OV5640_CAM

    //#define MT9D111A_CAM
    //#define MT9D111B_CAM
    //#define MT9M112_CAM
    //#define MT9V111_CAM
    .....

```

Figure 18: MODIFICATION DU MEMORYSAVER.H

Une fois on configure les paramètres de la caméra on peut compiler l'un des plusieurs exemples qui existent et prêt à l'emploi dans notre carte Arduino UNO. On trouve notre code dans Fichier—Exemples—ArduCAM—Arducam_Shield_V2_Camera_Playback.

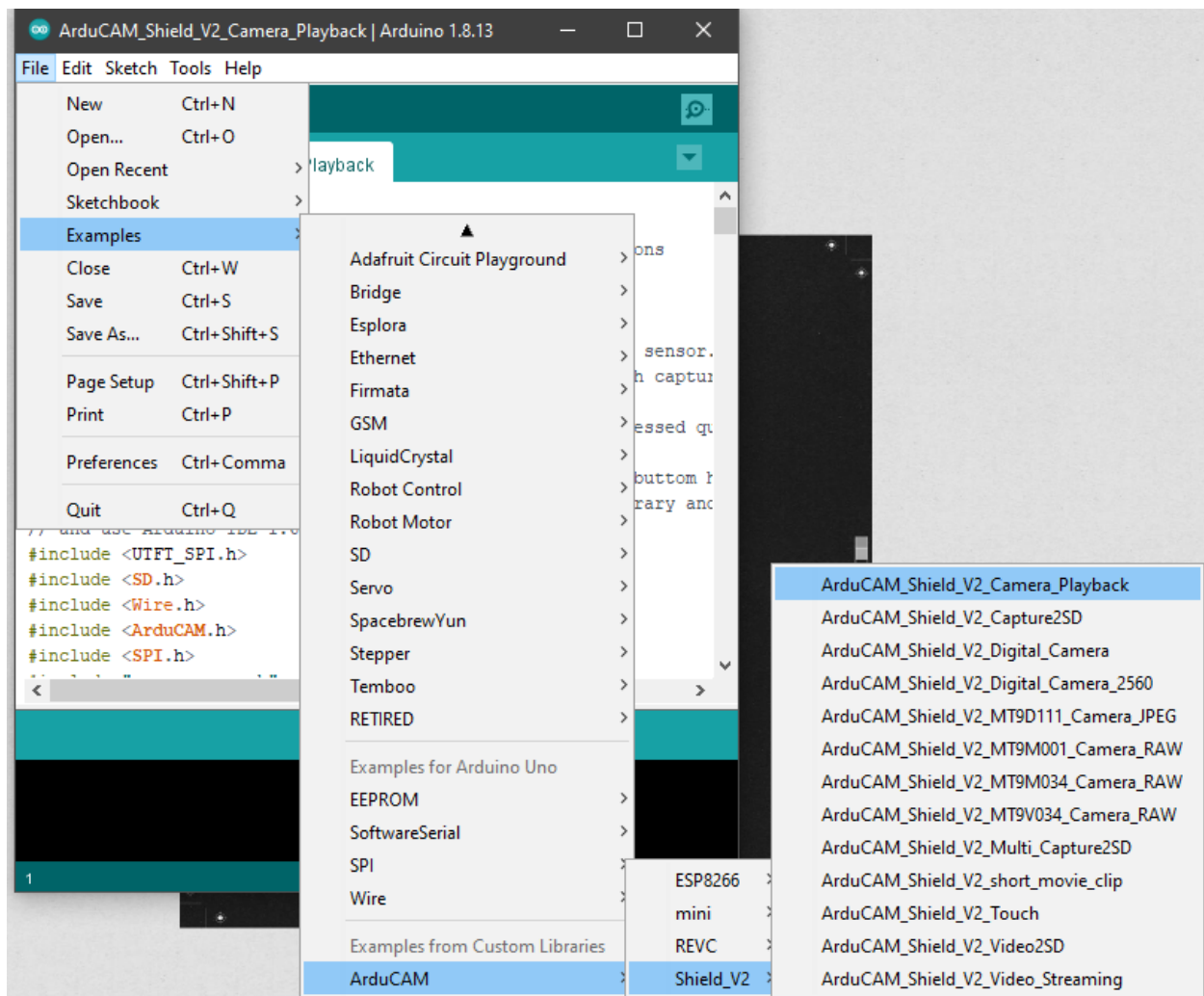


Figure 19: PROGRAMMER L'ARDUCAM AVEC L'ARDUINO

XIV. ACQUISITION DES IMAGES ET DES VIDEOS EN LABVIEW

Pour trouver la meilleure méthode pour afficher des images et vidéos en LabVIEW j'ai dû faire beaucoup de recherche et essayer plusieurs approches que je vais expliquer par suite.

A. AFFICHER LES IMAGES A L'AIDE DU LOGICIEL D'ACQUISITION DE VISION (VISION ACQUISITION SOFTWARE) :

La première méthode que j’ai essayé consiste d’installé un logiciel qui porte le nom “VISION ACQUISITION SOFTWARE” et appartient à la famille “NATIONAL INSTRUMENTS”.

A l’aide de ce logiciel on peut visualiser et enregistrer des images à partir d’une série d’interfaces de caméra standard, et on peut programmer à travers LabVIEW des applications à l’aide de différentes instructions qui s’ajoute sous le nom “vision and motion” dans la palette de fonctions dès qu’on installe NI VAS.

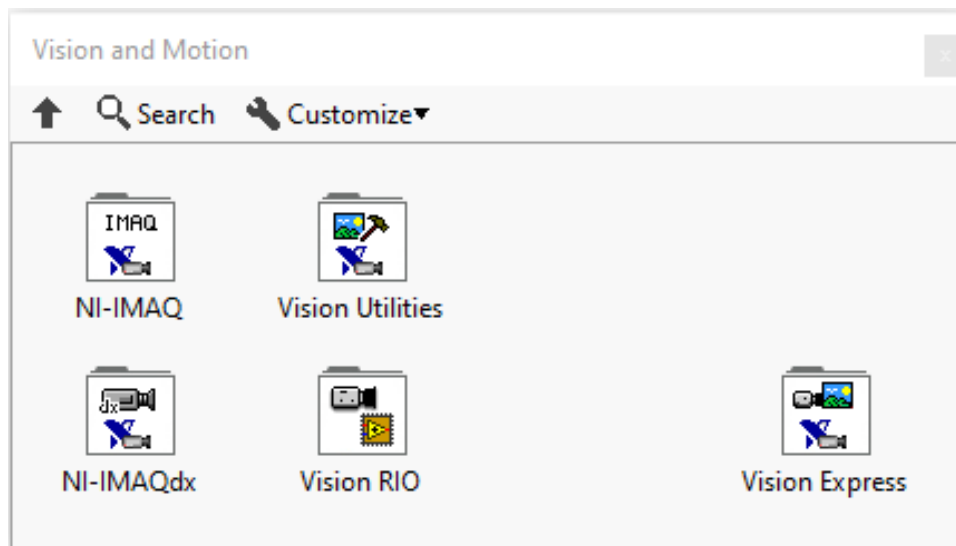


Figure 20: “VISION AND MOTION” SOUS LABVIEW

En utilisant les sous VI qu’on a ajouté j’ai trouvé une méthode pour afficher une image à l’aide d’une caméra connecté à mon ordinateur à l’aide d’un port USB. Vous trouverez la face avant et le bloc diagramme de cette méthode en figure 21 et figure 22.

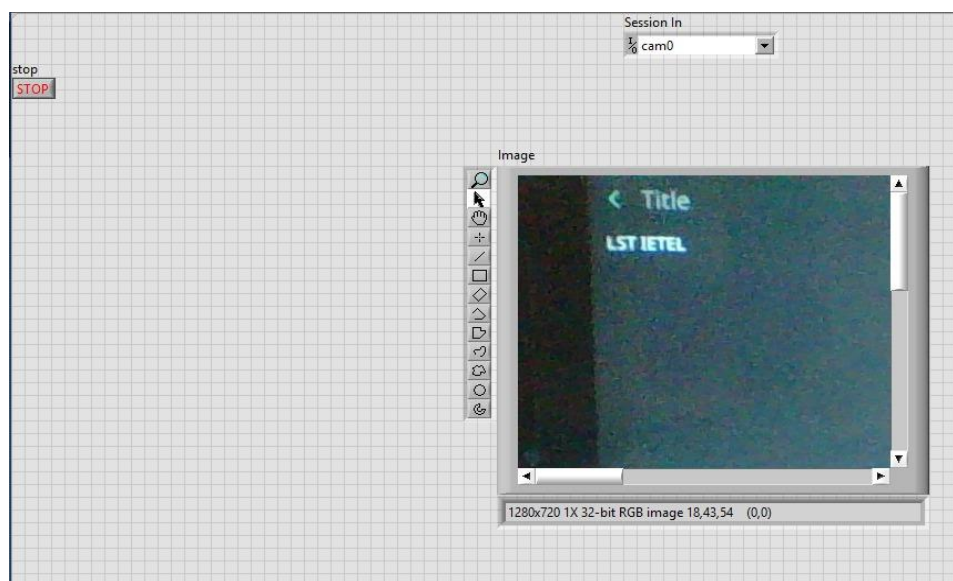


Figure 21: LA FACE AVANT DU PROGRAMME 1

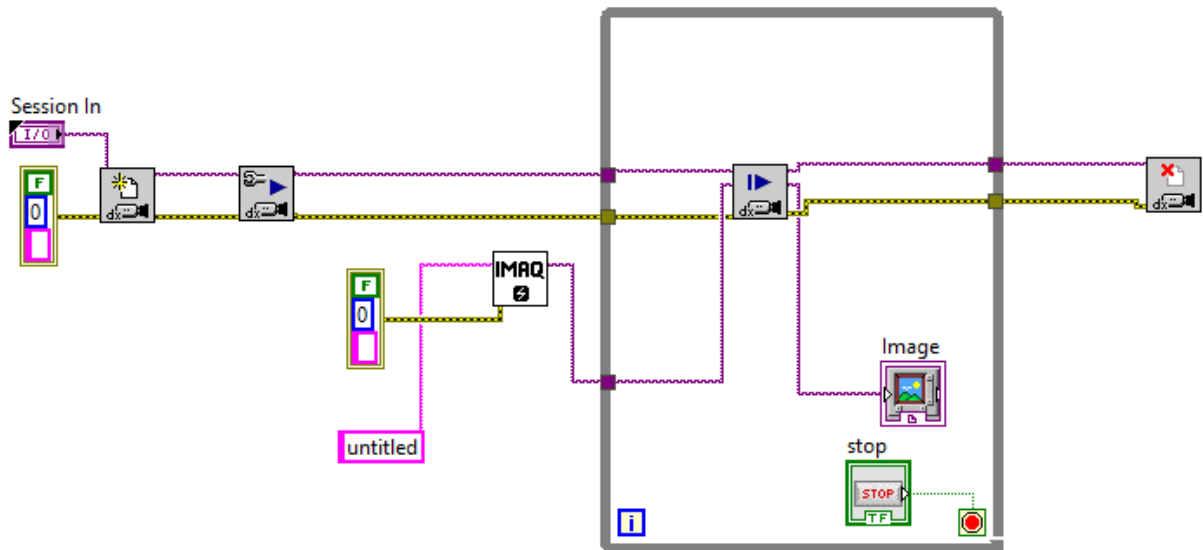


Figure 22: LE BLOC DIAGRAMME DU PROGRAMME 1



: **OPEN CAMERA VI** : Ouvre la caméra et définit le port auquel la caméra est connecter.

: **CONFIGURE GRAB VI** : Configure et démarre une acquisition instantanée.

: **IMAQ CREATE VI** : Crée un emplacement de mémoire temporaire pour une image

: **SNAP VI** : Sert à capturer l'image

: **IMAGE DISPLAY** : Où nous allons afficher l'image

: **CLOSE CAMERA VI** : Arrête l'acquisition en cours

L'idée principale été de connecter la carte Arduino avec l'ordinateur via port USB et choisir le port qu'on a utilisé pour la connexion, mais cette approche ne marche qu'avec une webcam donc j'étais obligé de voir une autre approche pour afficher l'image capturer par notre ArduCAM de module OV7670.

B. STOCKER LES IMAGES DANS UN DOSSIER PUIS LES AFFICHER EN LABVIEW

1. CAPTURE DES IMAGES PAR OV7670 :

Après la connexion de la carte Arduino avec l'ArduCAM avec succès et compiler le programme qui correspond à notre module OV7670 dans la carte, il faut suivre quelques étapes pour capturer les images par notre camera et les stocker dans notre ordinateur pour les afficher dans la prochaine étape en logiciel LabVIEW.

La première instruction qu'il faut suivre dans cette étape c'est l'installation du logiciel Java JDK pour pouvoir interfacé notre camera avec l'ordinateur.

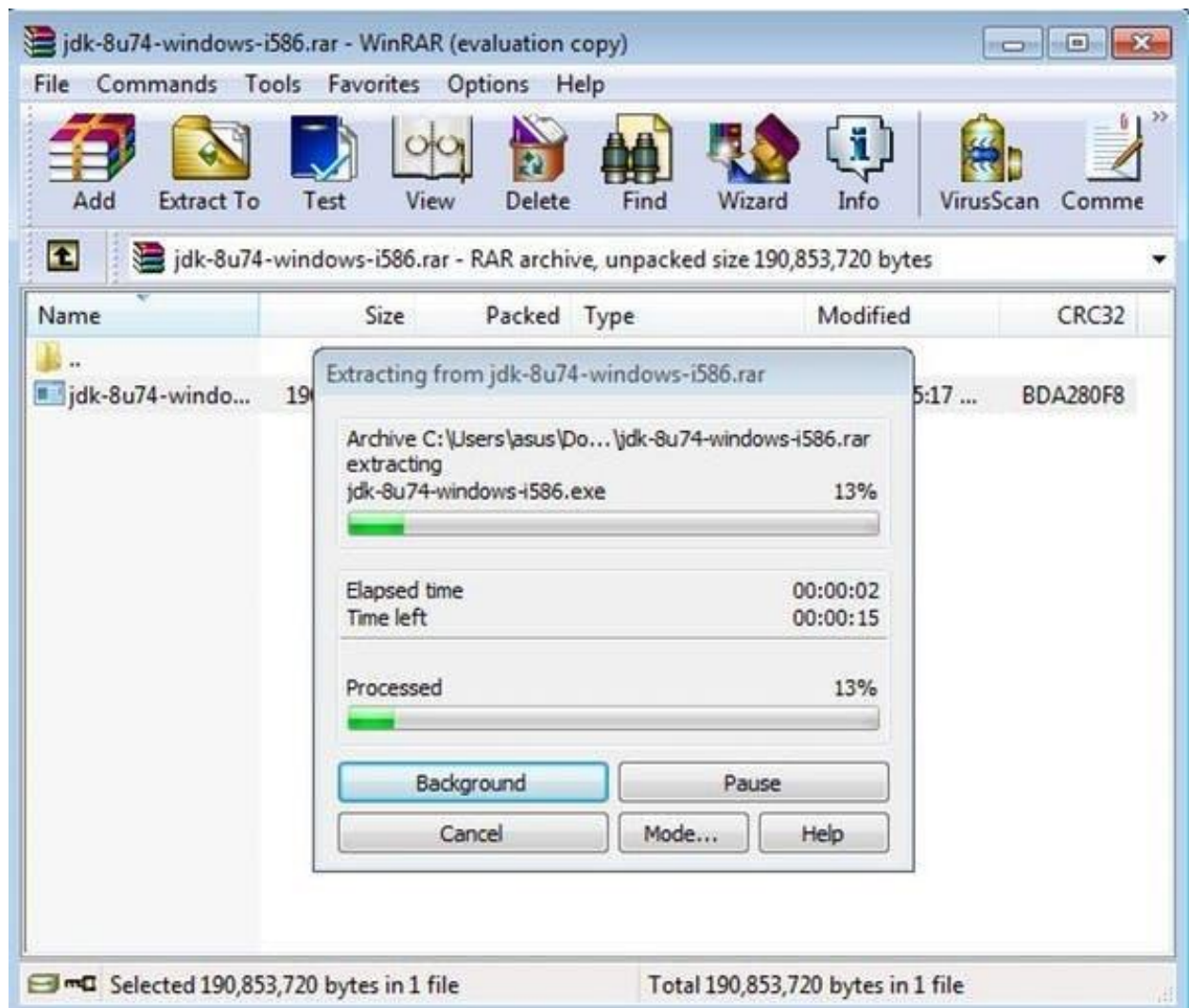


Figure 23: INSTALATION DE LOGICIEL JAVA JDK

Comme étape suivante on doit vérifier à quelle port notre carte Arduino est connecté on peut vérifier par accéder Panneau de configuration> Gestionnaire de périphériques> Ports (COM & LPT)> clic droit> Propriétés> Paramètres de port> Avancé> Numéro de port COM>

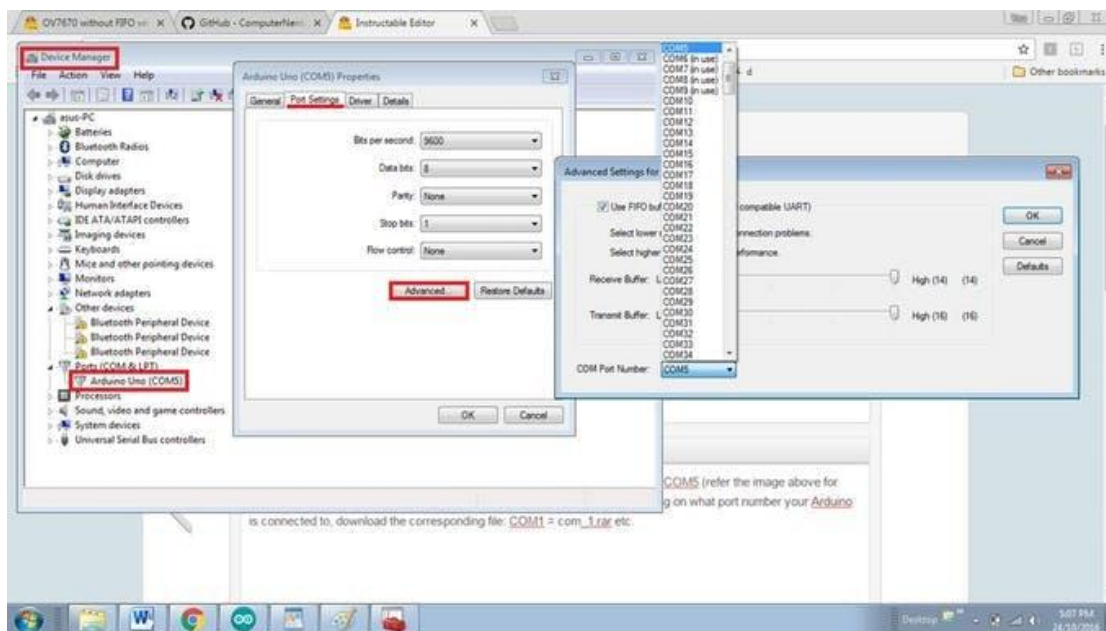


Figure 24: VERIFICATION DE NUMERO DE PORT

Selon le numéro auquel la carte Arduino est connectée il faut télécharger un fichier correspondant par exemple COM 5.rar et dans ce fichier Zip ce trouve un dossier 'code' qu'il faut coller dans Disque local> Fichiers programme> Java> jdk1.8.0_74> bin, après on devait créer un nouveau dossier sur le lecteur C qui sera la destination de la photo prise par le capteur OV7670.

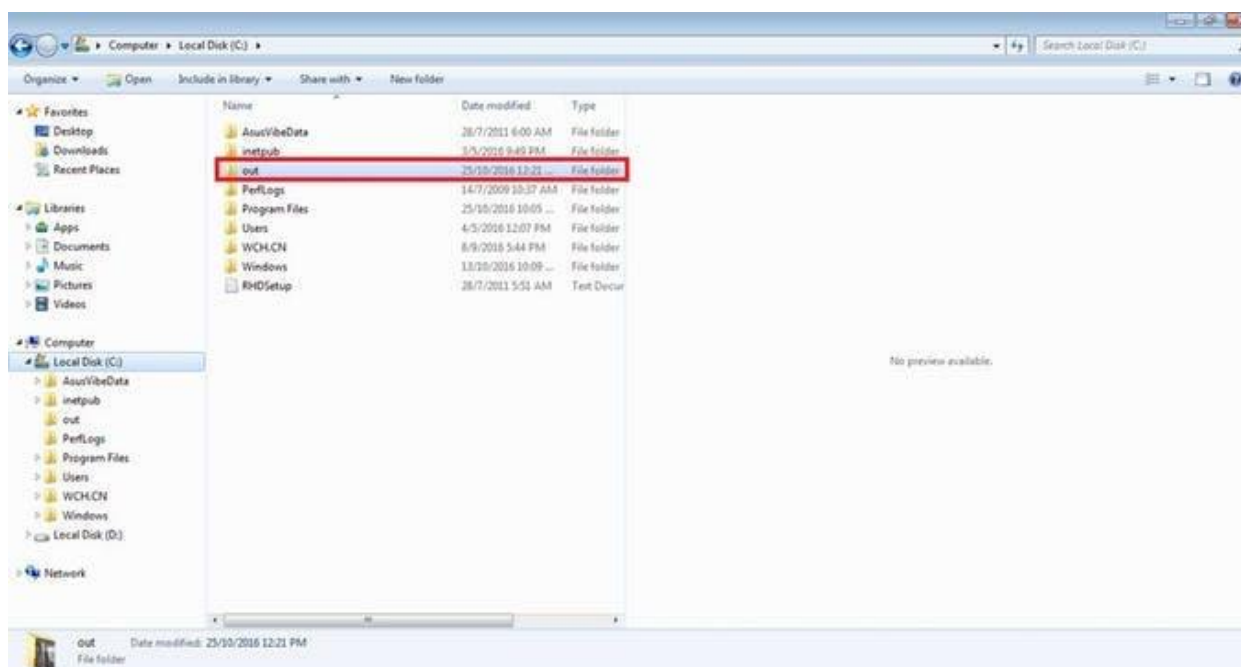


Figure 25: LE DOSSIER DE DESTINATION

Ensuite on doit télécharger et extrayez le fichier Extra.rar qui se dispose d'un fichier 'win32com.dll' qu'on doit placer dans "C: \ Program Files \ Java \ jdk1.8.0_74 \ jre \ bin", un fichier 'comm.jar' qu'on doit placer dans "C: \ Program Files \ Java \ jdk1.8.0_74 \ jre \ lib \ ext" et 'javax.comm.properties' dans le "C: \ Program Files \ Java \ jdk1 .8.0_74 \ jre \ lib .

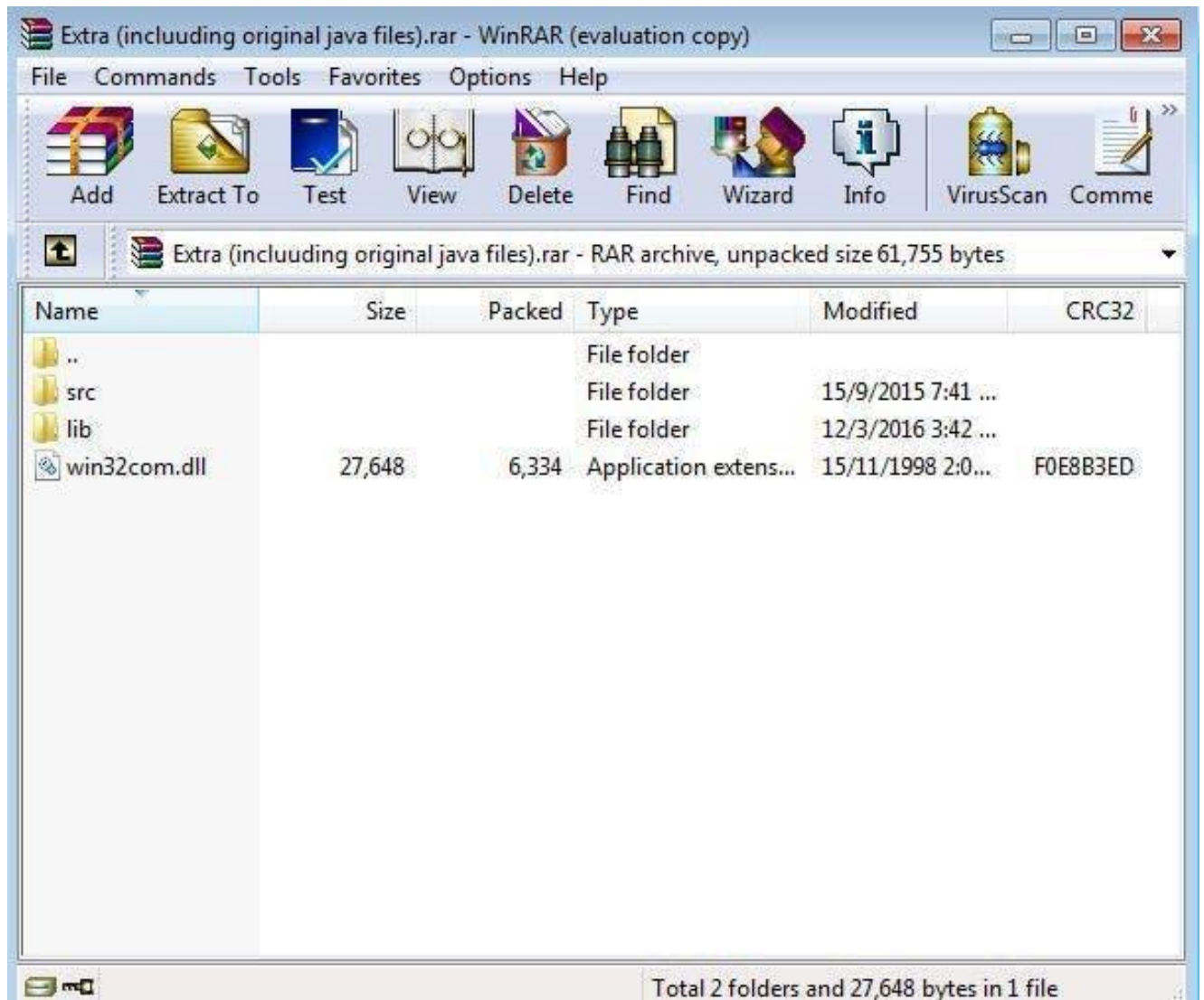


Figure 26: LE FICHIER EXTRA.RAR

L'étape qui suit consiste d'ouvrir le dossier "C: \ Program Files \ Java \ jdk1.8.0_74 \ bin", et en écrivant 'CMD' dans l'adresse de fichier une fenêtre de CMD s'affiche, maintenant il faut écrire "java code.SimpleRead" et il devrait être comme ça --> C: \ Program Files \ Java \ jdk1.8.0_74 \ bin> java code. SimpleRead.

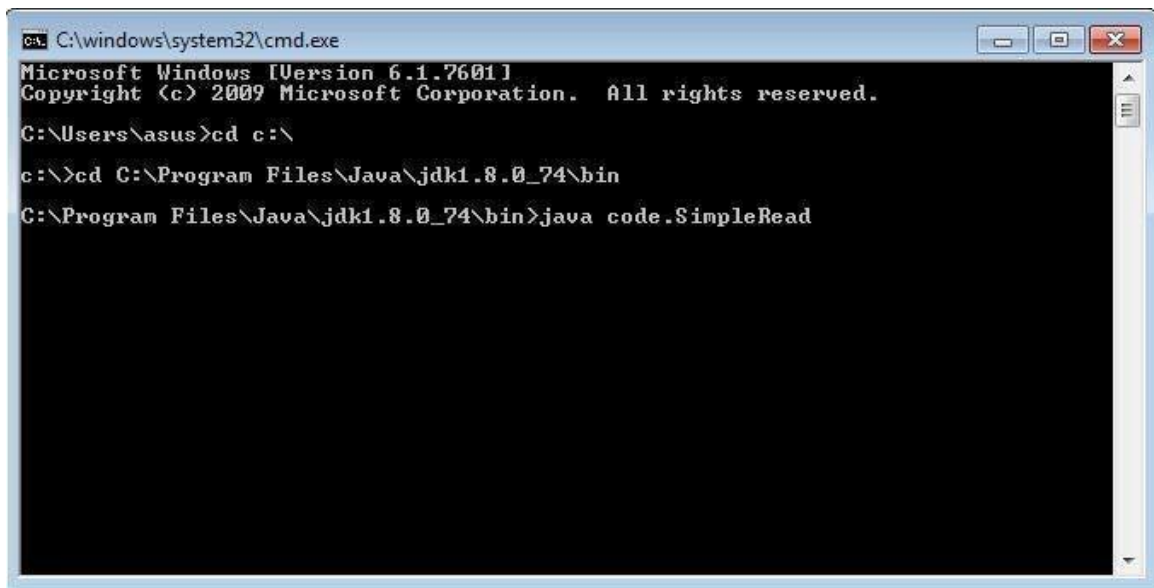


Figure 27: LE FICHER CMD

Finalement comme résultat des images capturées par le capteur d'image OV7670 automatiquement s'affiche dans le dossier de destination créer du nom 'out', les premières images peut s'affiche avec une mauvaise qualité mais après quelque tentative la qualité s'améliore.

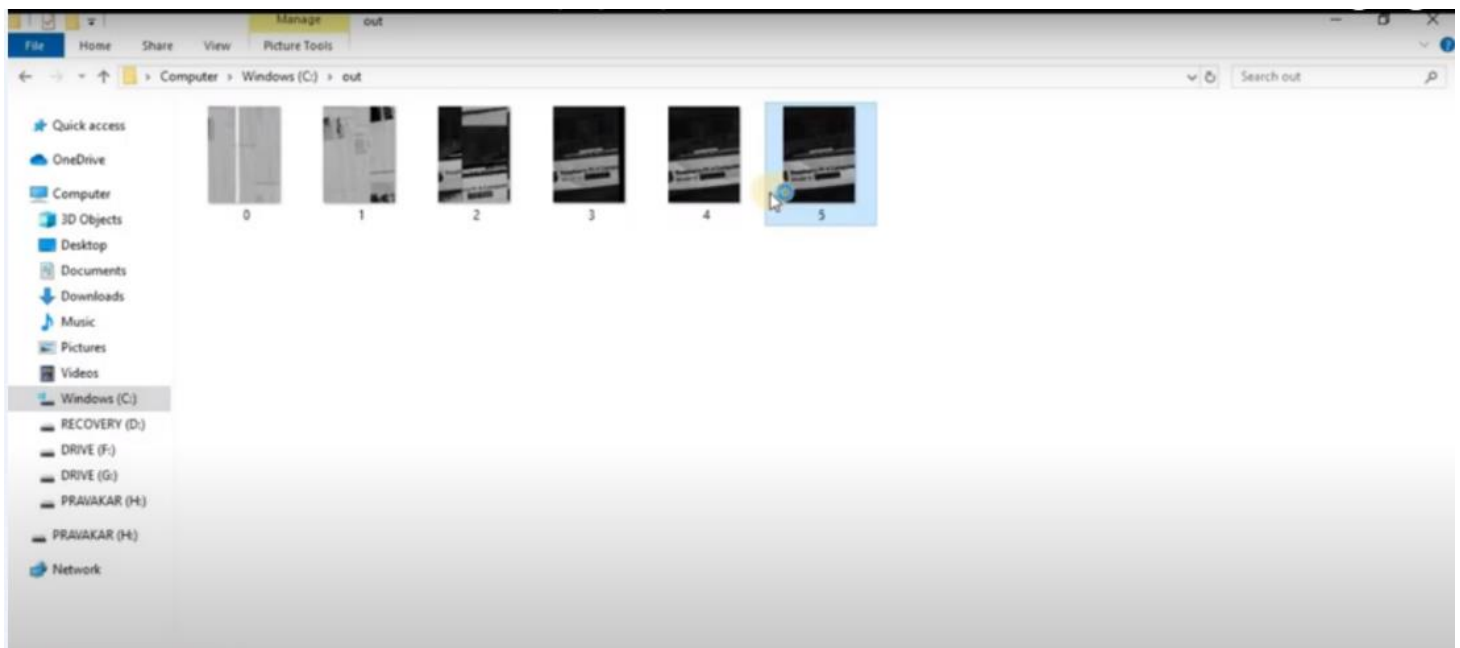


Figure 28: LES IMAGES CAPTUREES PAR LA CAMERA

2. AFFICHER LES IMAGES EN LABVIEW

Dans l'étape précédente nous étions capables de stocker les images obtenir par l'ArduCAM dans un dossier, pour cette étape on va les lire par le logiciel LabVIEW. Et pour pouvoir le faire on doit construire le programme suivant qui sert à réaliser cette tâche.

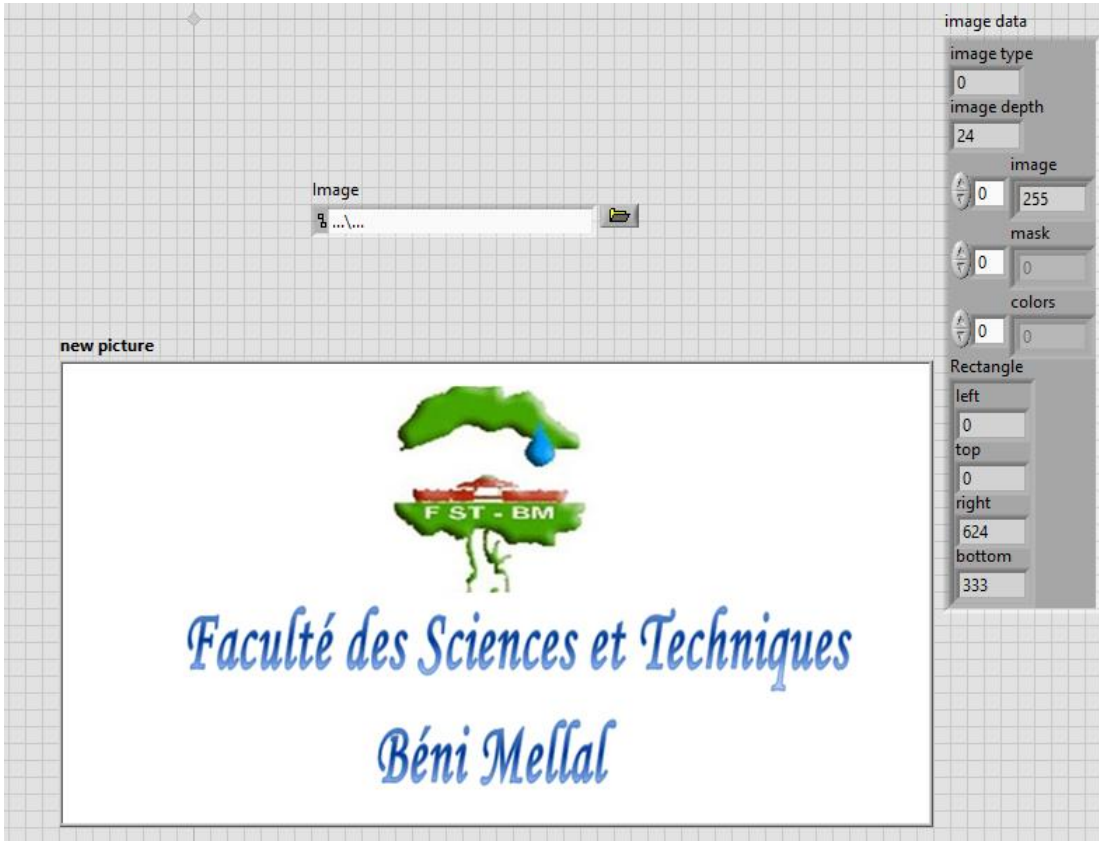


Figure 29: LA FACE D'AVANT DU PROGRAMME 2

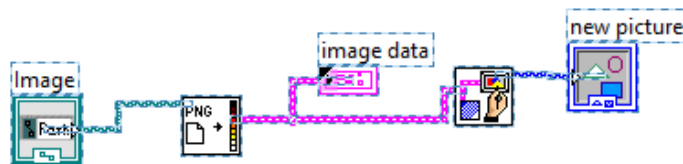


Figure 30: LE BLOC DIAGRAMME DU PROGRAMME 2

Le programme utilisé est simple, il compose d’une instruction ‘File Path Control’ qu’on peut trouver dans la palette de contrôle sous ‘string & path’ qui sert à accéder le fichier où se trouve l’image qu’on veut utiliser, un sous VI du nom ‘Read PNG file’ (l’image utilisé dans cette exemple est de format PNG) qu’on trouve cette fois dans la palette de fonctions sous ‘Graphics & Sound’ sert à lire notre image.

On peut créer une indicateur dans ce sous VI pour pouvoir afficher les informations sur la photo utilisé. Et dernièrement sous ‘Graphics & Sound’ une autre fois on choisit le sous VI ‘Draw Fattened Pixmap’ qui sert à examiner le cluster d’images, puis crée les informations appropriées pour un indicateur d’image.

Finalement on peut redimensionner l’image en choisissons ‘Set Width and Height’ selon les dimensions donner dans les informations d’image.

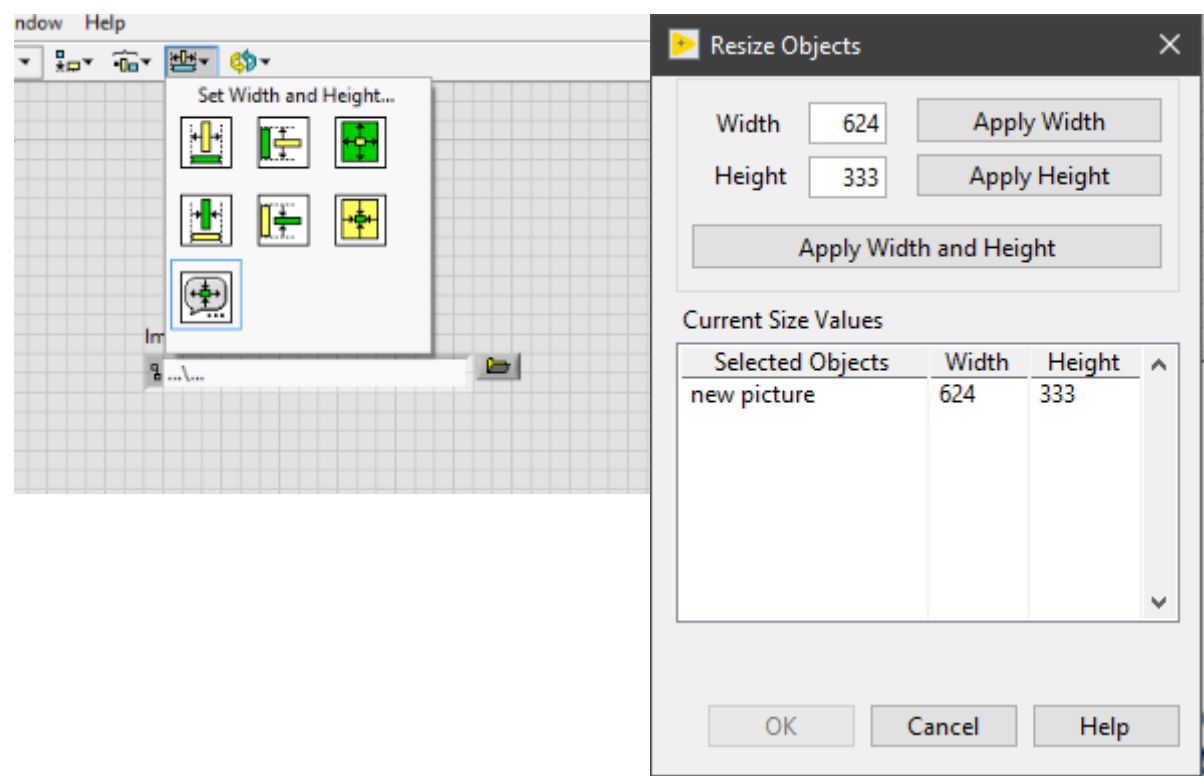


Figure 31: REDIMENSIONNER L’IMAGE

C. STOCKER LES VIDEOS DANS UN DOSSIER PUIS LES AFFICHER EN LABVIEW :

1. CAPTURE DES VIDEOS PAR OV7670 :

On a déjà pu stocker les images capturées par la carte Arduino dans un dossier dans l’étape précédente, mais nous avons fait face d’un seul inconvénient ; on ne peut pas stocker et afficher des vidéos en utilisant cette méthode. Donc on va utiliser un logiciel qui porte le nom “ArduCAM host App”, et il sert à afficher des images et des vidéos capturées par une ArduCAM de n’importe quelle module et les stocker dans un dossier.

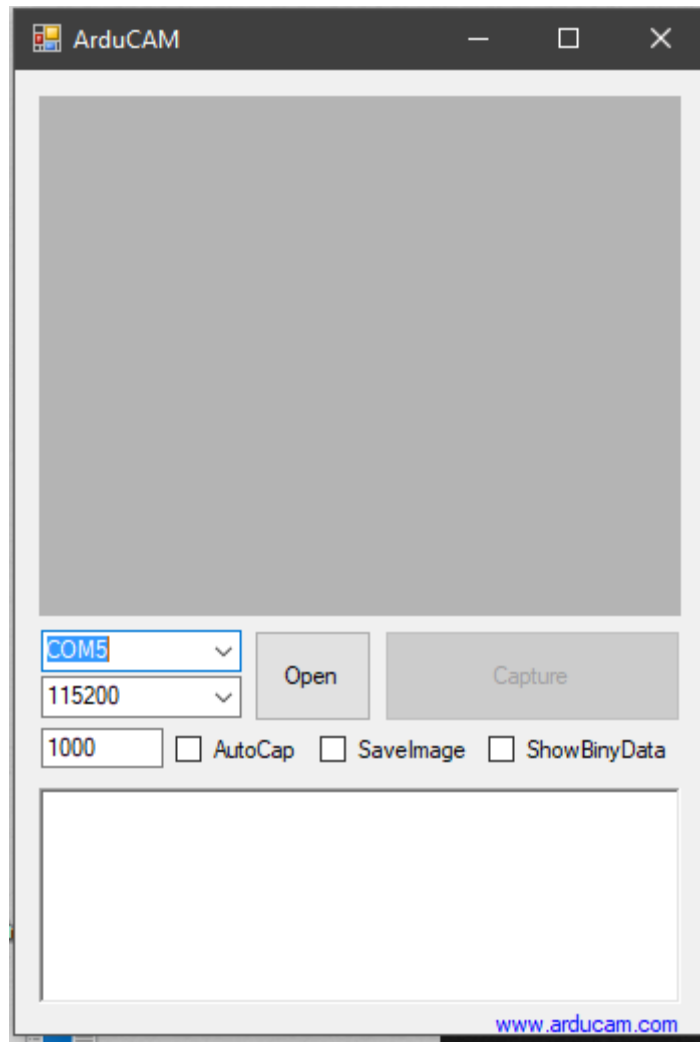


Figure 32: INTERFACE DE LOGICIEL ARDUCAM

En ouvrant le logiciel ‘‘ArduCAM Host App’’ on doit sélectionner le bon port COM pour notre carte Arduino et sélectionner le bon débit en bauds pour le COM. Une fois qu’on cliqué sur le bouton ouvrir, le bouton capture s’est activé et une image s’affiche dans l’interface de logiciel, et on peut maintenant exécuter manuellement la commande de capture en cliquant sur le bouton capture ou en définissant l’intervalle de temps en milliseconde pour la capture automatique. L’option SaveImage doit être sélectionnée si on veut enregistrer l’image et la stocker dans un fichier dans notre ordinateur. Et pour capturer une Vidéo il faut tout simplement sélectionner l’option SaveImage et l’option AutoCap qui sert à capturé automatiquement à base de l’intervalle de temps définit et ensuite combiner toutes les images de capture dans un fichier AVI pour créer un vidéo qu’on va l’afficher en logiciel LabVIEW dans la prochaine étape.

CONCLUSION

Premièrement, je tiens à témoigner de l'importance du projet de fin d'étude par les étudiants de License. Il constitue un vrai test des connaissances acquises et le monde du travail.

Le travail qui m'a été proposer c'est un projet qui touche plusieurs aspects de l'électronique, principalement les nouvelles carte de contrôle Arduino ces avantages et comment on peut les utiliser. J'ai aussi touché à la programmation en relation avec l'exploitation de ces cartes par les logiciels IDE Arduino, Ardublock et LabVIEW, l'aspect relation entre LabVIEW et l'interfaçage d'Arduino a été aussi maîtrisé après plusieurs tentatives très difficiles. J'ai aussi maîtrisé l'utilisation des modules caméra avec Arduino pour faire des captures et les sauvegarder et enfin, l'exploitation des images et vidéos sauvegardées par le composant traitement d'images de LabVIEW à savoir « LabVIEW vision ».