Comment permettre à une personne tétraplégique de jouer à un jeu vidéo à l'aide d'un dispositif sans contact ?

Sciences à l'École

Maryse Franzen
Florian Hector
Marlon Hoffmann
Lucas Korbas

FONDATION CGENIAL

Condorcet Schoeneck

Concevoir, créer, réaliser...

Equipe Pédagogique : M. Schneider Pascal - M. Albanesi Serge

RESUME

David Martinel, chirurgien-dentiste, est à l'origine de ce projet initié en Septembre 2015. Ludovic Genco, son jeune cousin, passionné de jeux vidéo est devenu tétraplégique à la suite d'une tumeur qui lui a détruit la moelle épinière au-dessous de la nuque.

David décide alors de se tourner vers le Lycée Condorcet de Schœneck avec l'idée de développer une « manette buccale » pour que son cousin puisse rejouer à Gran Turismo, son jeu favori.

A la rentrée 2016, David a souhaité améliorer la «manette buccale» en proposant de concevoir un nouveau prototype qui élimine tout contact physique avec le joueur.

Quatre élèves de Terminale STIDD (Maryse Franzen, Florian Hector, Marlon Hoffmann et Lucas Korbas) motivés à la fois par l'aspect technologique et par la dimension humaine du projet, se sont portés volontaires.

Ils entendent relever ce défi en proposant une solution capable d'apporter une réponse pertinente à la problématique suivante : « comment permettre à une personne tétraplégique de jouer à un jeu vidéo à l'aide d'un dispositif sans contact ? ».

Pour ces élèves, ce sujet sert également de support à l'épreuve de projet du Baccalauréat de la Série STIDD option Systèmes d'Information et Numérique.

Deux partenaires les accompagnent dans cette aventure : David Martinel pour l'énoncé du besoin et ses conseils sur la dynamique de l'œil et du visage et l'IUT de Moselle-Est (Département Science et Génie des Matériaux) en ce qui concerne le choix des matériaux composites notamment.

Le Cahier des Charges Fonctionnel (CdCF) impose la reconnaissance des quatre touches indispensables pour pouvoir jouer à Gran Turismo, à savoir : l'accélérateur, le frein ainsi que les virages à droite et à gauche.

Les touches relatives à l'accélérateur et au frein seront délivrées par un module programmable équipé d'une caméra mobile positionnée sur le front du joueur. Il s'agit de détecter le mouvement vertical du menton.

Les touches permettant au joueur de tourner à droite ou à gauche, générées à partir de la détection de la position axiale du visage du joueur, seront fournies par un module programmable équipé d'une caméra fixe.

Le CdCF indique également que la transmission des informations délivrées par les deux modules programmables équipés de caméra devra être réalisée par l'intermédiaire d'une transmission par Bus I2C de type Maitre – Esclave.

SOMMAIRE

1. INTRODUCTION 3

1.1 Septembre 2015… naissance d'un projet solidaire, 3

1.2 Septembre 2016… le projet évolue ! 3

1.21 Une problématique motivante à double titre… 3

1.22 Deux partenaires nous accompagnent… 4

2. DESCRIPTION GENERALE DU PROJET 4

2.1 Le Cahier des Charges Fonctionnel (CdCF) 4

2.2 Le Diagramme SysML des Cas d'Utilisation 5

2.3 Synoptique complet du projet 5

3. MODULE OPEN MV CAM : LA SOLUTION POUR LA RECONNAISSANCE SANS CONTACT 6

3.1 Ressources et lignes d'Entrées-Sorties du module OPEN MV Cam 6

3.2 Principales fonctions du module OPEN MV Cam 6

3.21 Techniques de détection du visage 7

3.22 Caractéristiques d'un détecteur de visage idéal 7

3.23 La méthode de Haar 7

3.24 OpenMV Cam dispose d'une fonction de détection de type « Haar Cascade » 7

3.3 Environnement de développement OpenMV IDE 8

3.4 Micro Python, le langage pour programmer l'OpenMV Cam 8

4. BUS I2C - COMMUNICATION ENTRE LES MODULES CAMERA ET LA CARTE ARDUINO UNO 9

4.1 Principe du Bus I2C 9

4.2 Organisation matérielle de la communication série 9

4.3 Organisation logicielle de la communication série 10

4.4 Format des variables contenant les commandes du joystick 11

4.41 Format de l'information « Accélérateur – Frein » 11

4.42 Format de l'information « Droite – Gauche » 11

5. ALGORITHMES SIMPLIFIES DES PROGRAMMES DES ESCLAVES ET DU MAITRE 12

6. PROCEDURE DE TRANSFORMATION DE LA CARTE ARDUINO UNO EN UN PERIPHERIQUE USB 13

6.1 Principales caractéristiques de la carte 13

6.2 Transformation de la carte Arduino Uno en manette de jeu (joystick) 14

6.21 Initialisation Logicielle 14

6.22 Vérification Matérielle : test du bouton « Triangle » de la manette 15

6.23 Transfert du bootloader dans l'ATMEGA8U2 de la carte Arduino Uno 16

6.231 Passage de la carte Arduino Uno en mode DFU 16

6.232 La carte Arduino Uno est reconnue comme un Joystick 16

7. CONCLUSION 17

8. SITOGRAPHIE 18

1. INTRODUCTION

1.1 Septembre 2015… naissance d'un projet solidaire,

L'idée est née à la rentrée de Septembre 2015. David Martinel, chirurgien-dentiste à Morsbach (Moselle), en est à l'origine. Ludovic Genco, son jeune cousin passionné de jeu vidéo est devenu tétraplégique à la suite d'une tumeur qui lui a détruit la moelle épinière au-dessous de la nuque. Pour que Ludovic puisse rejouer à Gran Turismo, son jeu favori, David Martinel accompagné d'Anthony Vullo, prothésiste dentaire à Cocheren (Moselle) décide de se tourner vers le Lycée Condorcet de Schœneck avec l'idée de développer et de mettre au point une « *manette buccale* » innovante. Dès les premières semaines qui suivent leur rentrée scolaire, cinq élèves de Terminale STIDD se mettent au travail et quelques mois plus tard ils sont en mesure de présenter un prototype fonctionnel. Celui-ci, baptisé *Mouth Game*, permet de jouer sur la console Playstation. La *Mouth Game* a été testée et approuvée par Ludovic Genco à son domicile de Rosbruck (Moselle). La vidéo, réalisée par les journalistes du quotidien régional « Le Républicain Lorrain », retraçant l'historique du projet et l'essai effectué par Ludovic, est disponible à l'adresse : http://www.republicain-lorrain.fr/edition-de-forbach/2016/07/02/des-lyceens-de-schoeneck-creent-un-joystick-pour-tetraplegique).

1.2 Septembre 2016… le projet évolue !

Lors de la rentrée de Septembre 2016, le Docteur David Martinel, a souhaité poursuivre la collaboration avec le Lycée Condorcet avec comme objectif d'améliorer le prototype de «*manette buccale*» développé durant l'année scolaire 2015/2016. Son idée consiste à faire évoluer le projet en développant un nouveau prototype qui élimine tout contact physique entre le joueur et la « manette de jeu ».

1.21 Une problématique motivante à double titre…

Quatre élèves de Terminale STIDD (Maryse Franzen, Florian Hector, Marlon Hoffmann et Lucas Korbas) motivés à la fois par l'aspect technologique et par la dimension humaine du projet, se sont portés volontaires pour relever ce défi avec la ferme intention de développer une solution répondant aux exigences du cahier des charges élaboré par le Docteur David Martinel et capable d'apporter une réponse pertinente à la problématique suivante :

COMMENT PERMETTRE A UNE PERSONNE TETRAPLEGIQUE DE JOUER A UN JEU VIDEO A L'AIDE D'UN DISPOSITIF SANS CONTACT ?

Pour ce groupe d'élèves, ce sujet sert également de support à l'épreuve de projet de la spécialité Systèmes d'Information et Numérique de la session de Juin 2017 du Baccalauréat Technologique Sciences et Technologies de l'Industrie et du Développement Durable.

1.22 Deux partenaires nous accompagnent…

Dr David MARTINEL Chirurgien-dentiste	IUT de Moselle-Est (Science et Génie des Matériaux)
96, Rue nationale 57600Morsbach	5, Rue Camille Weiss 57600 Forbach
Enoncé du besoin Dynamique de l'œil et du visage	Matériaux composites, Métaux et alliages, Polymères
Réunions de suivi de projet : 1 toutes les 6 semaines Revues de projet : 3	

2. Description Generale du projet

La première étape qui a consisté à faire le bilan des dispositifs existants et à rechercher les brevets qui auraient été déposés auprès de l'INPI (http://bases-brevets.inpi.fr/fr/accueil.html). Ce travail de recherche a permis de se rendre compte que les systèmes proposés par le marché ou développés par des passionnésne correspondent pas aux attentes particulières formulées par Ludovic Genco et le Docteur David Martinel dans le Cahier des Charges Fonctionnel (CdCF).

2.1 Le Cahier des Charges Fonctionnel (CdCF)

Le CdCF, rédigé conjointement par nos professeurs et nos partenaires, impose la reconnaissance d'au minimum quatre touches indispensables pour pouvoir jouer à GranTurismo, à savoir :

- l'accélérateur et le frein (information numérique codée sur un octet non signé),
- et les virages à droite et à gauche (information numérique codée sur un octet non signé).

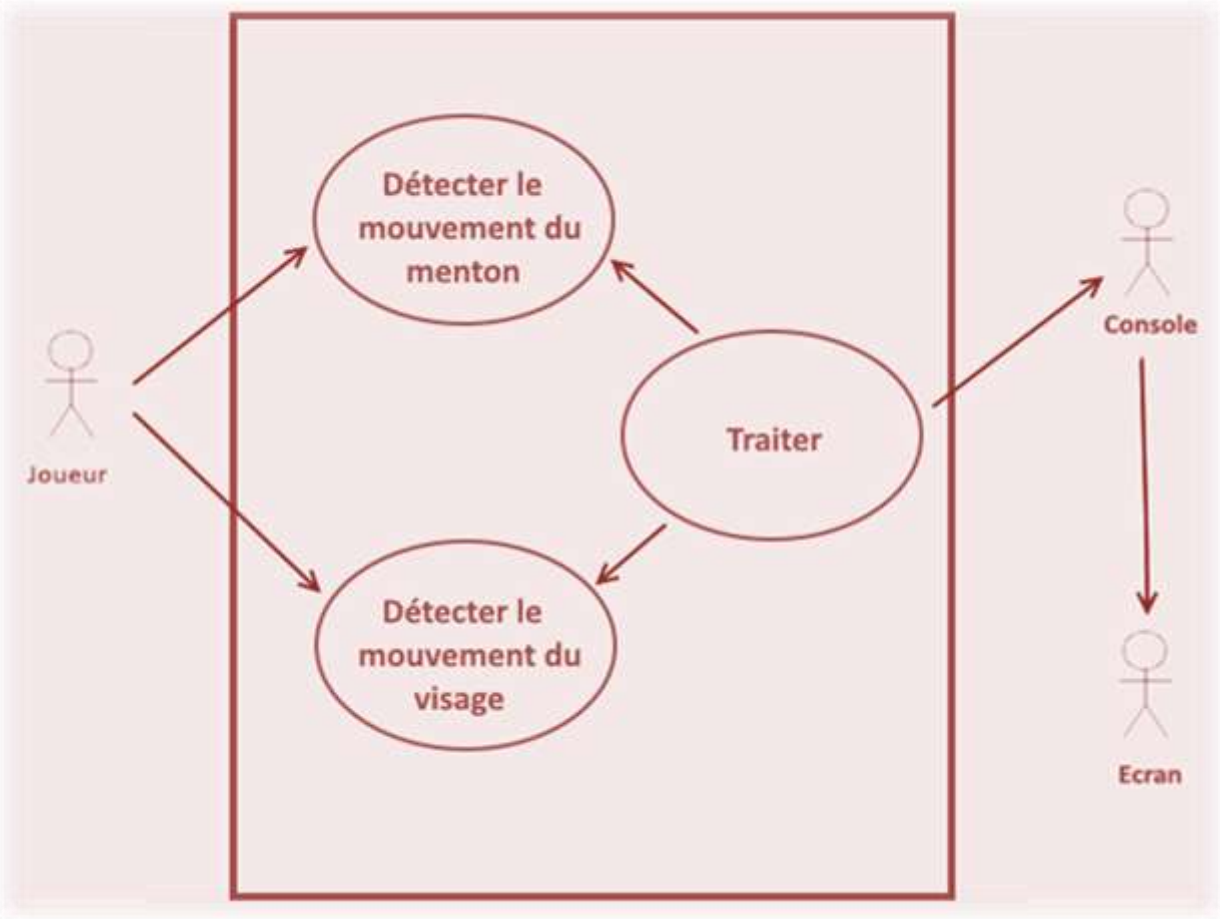
Les touches relatives à l'accélérateur et au frein seront délivrées par un module programmable équipé d'une caméra mobile positionnée sur le front du joueur (détection du mouvement du menton). Les touches permettant au joueur de tourner à droite ou à gauche (détection de la position axiale du visage) seront fournies par un module programmable équipé d'une caméra fixe.

Le CdCF indique également que la transmission des informations délivrées par les deux modules programmables équipés de caméra devra être réalisée par l'intermédiaire d'une liaison série synchrone.

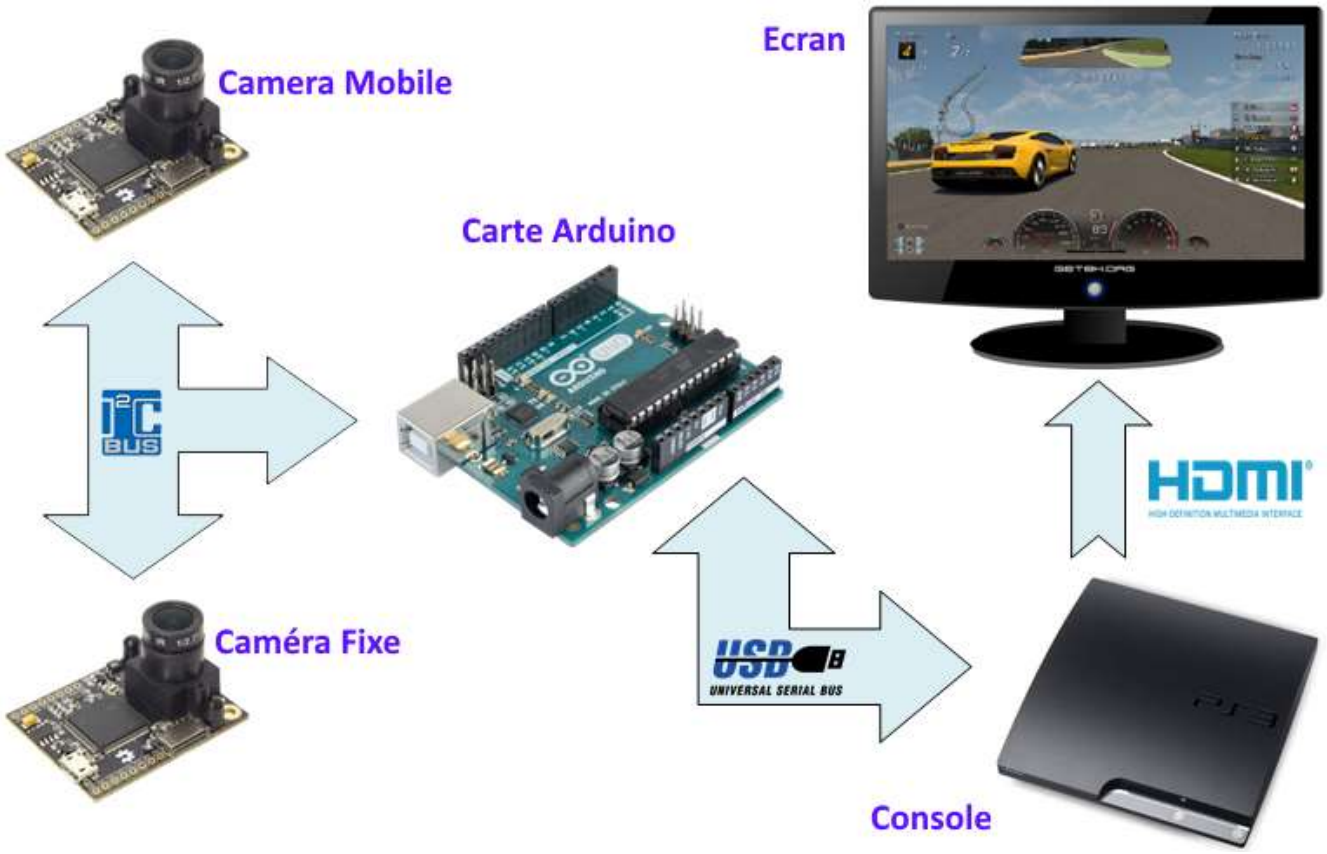
Pour répondre à cette exigence technologique, nous avons fait le choix de mettre en œuvre une transmission Maitre – Esclave par Bus I2C. Ce mode de transmission permet également de s'inscrire dans une démarche de développement durable en raison de sa simplicité de mise en œuvre et des économies de ressources qu'elle entraine (deux lignes suffisent pour assurer la transmission et les composants sont connectés en dérivation sur le bus).

Le diagramme de cas d'utilisation et le synoptique complet du projet représentés à la page suivante permettent d'illustrer la structure des principales fonctionnalités nécessaires à l'utilisateur du système.

2.2 Le Diagramme SysML des Cas d'Utilisation



2.3 Synoptique complet du projet

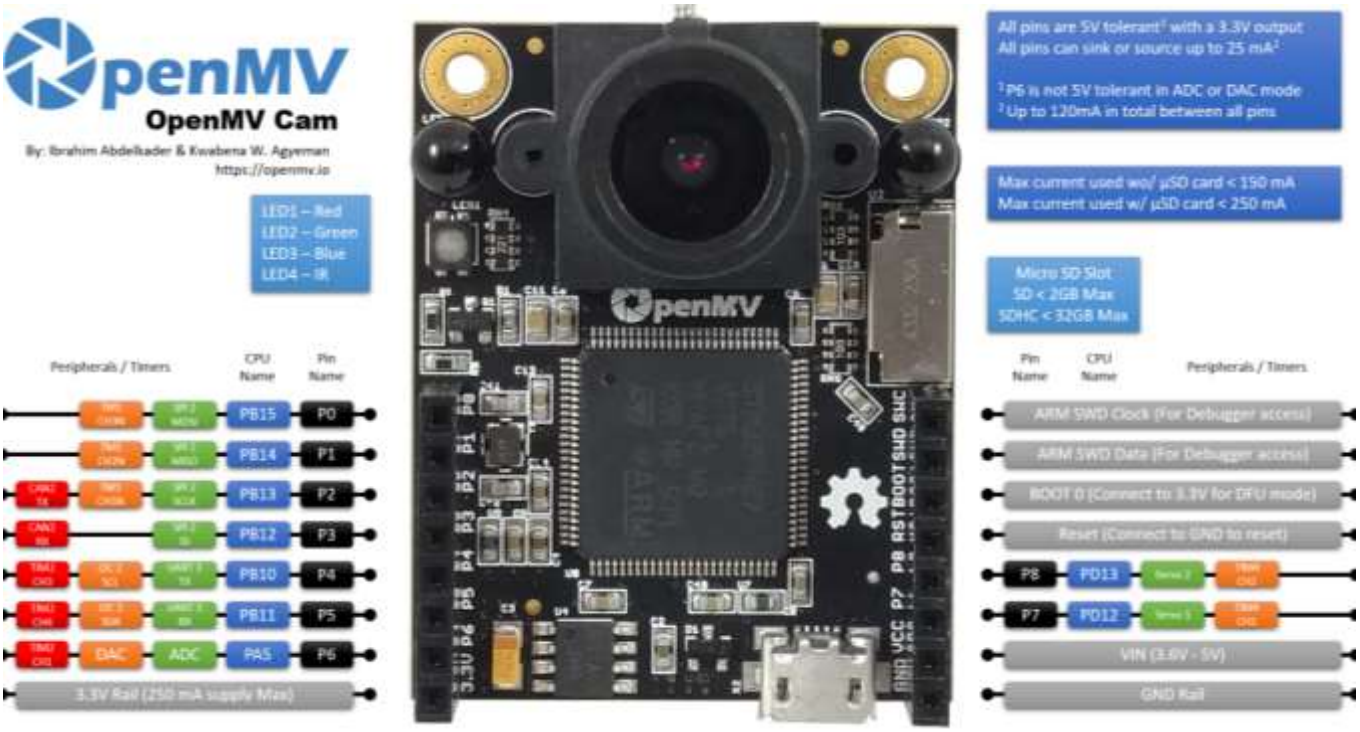


3. Module Open MV Cam : la solution pour la reconnaissance sans contact

OpenMV Cam est un module architecturé autour du puissant microcontrôleur STM32F427VG, développé par STMicroelectronics, et de la caméra OV1725 fabriquée par Ominivision. L'environnement de développement assure le contrôle total de la caméra. Il est notamment possible de déclencher la prise de photos ou de vidéos sur des événements externes et d'effectuer une sauvegarde sur une carte mémoire aux formats SD.

3.1 Ressources et lignes d'Entrées-Sorties du module Open MV Cam

Le module, qui consomme environ 150 mA en fonctionnement, possède 9 broches d'Entrées-Sorties configurables par l'utilisateur. Le module dispose d'une broche donnant accès à un convertisseur CAN ou ADC, d'un bus SPI, d'un bus I2C, d'un port série et de deux broches permettant la commande de « Servomoteurs ».



La connexion avec l'ordinateur est assurée à l'aide d'un port série USB à haute vitesse (12 Mbits/s). Le module est équipé de 3 diodes électroluminescentes (deux infrarouges et une RVB). La caméra dispose d'une monture d'objectif M12 avec un objectif doté d'un champ de vision de 4m à 92 degrés et un filtre infrarouge démontable. La résolution maximale des images est de 320 pixels x 240 pixels.

3.2 Principales fonctions du module Open MV Cam

3.21 Techniques de détection du visage

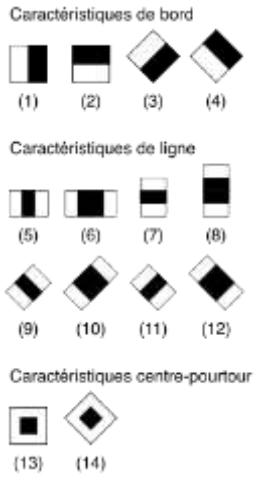
Il existe une variété d'approches consacrées à la détection de visage. Certaines techniques localisent des visages par détection de mouvements tels que le clignement des yeux ou en considérant les positions relatives entre les yeux, le nez et la bouche. Mais des approches bien plus complexes, telle que la détection de visages en utilisant des réseaux de neurones, sont aussi mises en œuvre. D'autres méthodes sont également utilisées comme la détection de contours ou bien la reconnaissance de forme. En ce qui nous concerne nous nous sommes intéressés à l'une des méthodes les plus utilisées, la classification par caractéristiques de Haar car elle est, de loin, la plus rapide.

3.22 Caractéristiques d'un détecteur de visage idéal

Un détecteur de visage idéal possède trois caractéristiques, robustesse, rapidité et simplicité. Robustesse d'abord, car il doit être en mesure de gérer les variations d'apparence selon la pose, la taille, l'éclairage, l'occlusion, les milieux complexes, les expressions faciales et la résolution. Rapidité ensuite, car il doit pouvoir effectuer le traitement numérique de l'image en temps réel. Pour cela il doit être implémenté dans un processeur dédié ayant une puissance de calcul élevée. Simplicité enfin, car le temps d'apprentissage doit être court, le nombre de paramètres doit être peu important, et les échantillons d'apprentissage doivent être recueillis à moindre coût.

3.23 La méthode de Haar

Les valeurs d'un pixel ne nous informent que sur la luminance et la couleur d'un point donné. Il est donc plus judicieux de trouver des « descripteurs » fondés sur des caractéristiques plus globales. C'est le cas des classificateurs de Haar qui sont des fonctions permettant de connaître la différence de contraste entre plusieurs régions rectangulaires contiguës d'une image. Dans cette méthode, on code les contrastes existants dans un visage d'une manière binaire : valeur « 1 » si dans la région de l'image analysée on trouve une ressemblance avec l'objet recherché, valeur 0 » dans le cas contraire. Pour détecter un « objet » à l'intérieur d'une image, on déplace une fenêtre de recherche pixel après pixel de manière à couvrir l'ensemble de la surface. Le classificateur est conçu de manière à être « redimensionnable », ce qui lui permet de rechercher des « objets » de dimensions très diverses à l'intérieur d'une même image. Ces descripteurs, constitués de rectangles blancs et noirs (cf. image ci-contre), sont appliqués à l'image permettant ainsi de calculer la valeur f_i d'un « Haar-like-features », différence entre la somme des pixels se trouvant dans les zones blanches et la somme des pixels se trouvant des zones noires avec la relation suivante :



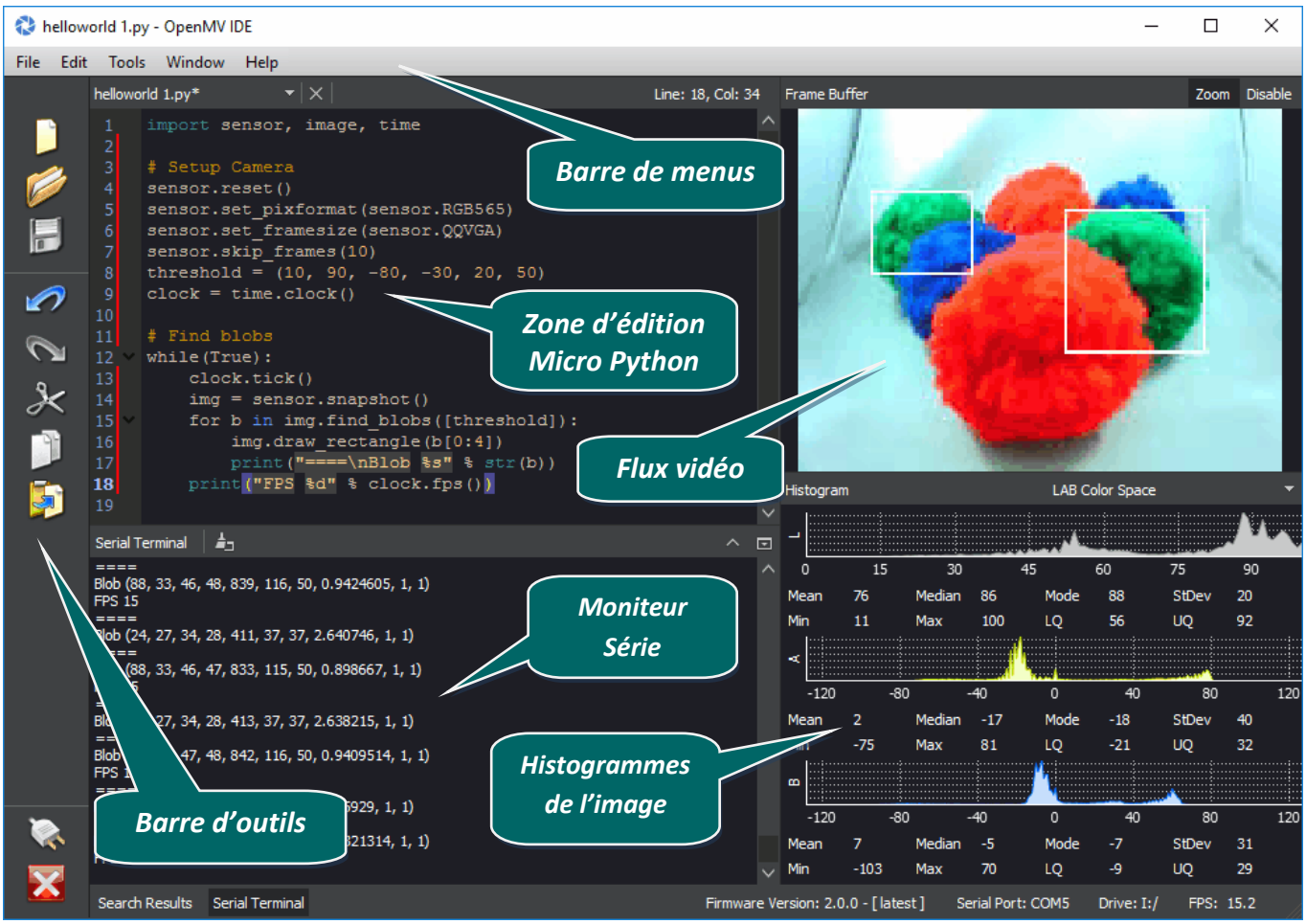
$$f_i = \text{Haar-like-features} = \text{Sum}(r_{i,\text{blanche}}) - \text{Sum}(r_{i,\text{noire}})$$

3.24 OpenMV Cam dispose d'une fonction de détection de type « Haar Cascade »

OpenMV Cam dispose de fonctions directement utilisables pour la détection de visages à l'intérieur d'une image. Ces algorithmes mettent en œuvre la méthode de Haar décrite au paragraphe précédent. Par défaut, l'algorithme déroule toutes les étapes. Cependant, il est possible de le paramétrer réduisant le nombre d'étapes pour accélérer le traitement au détriment, bien entendu, de la précision. Le « juste » compromis précision-rapidité est réglé lors des essais pratiques.

3.3 Environnement de développement OpenMV IDE

La programmation du module caméra est réalisée à l'aide de l'IDE OpenMV qui permet de coder les instructions en langage Micro Python.



L'IDE OpenMV comprend un éditeur de texte, une fenêtre qui visualise le flux vidéo, un terminal série pour le débogage et une zone d'affichage d'histogrammes pour analyser la répartition des tons.

3.4 Micro Python pour programmer l'OpenMV Cam

Micro Python est une version optimisée et allégée de Python 3 pour système embarqué. Ce langage, qui dispose d'une large communauté est une réécriture en C ANSI du langage de programmation Python. Il est particulièrement adapté au traitement de l'image. Par exemple, la méthode find_blobs () trouve les groupes de pixels dans une image qui partagent une propriété commune (valeur de gris par exemple). Dessiner un rectangle autour de chaque blob se fait facilement en seulement deux lignes de code.

Micro Python s'exécute sur la carte OpenMV Cam en « Bare Metal » c'est-à-dire dans un environnement dépourvu de système d'exploitation. La programmation s'appuie uniquement sur les ressources du processeur et des périphériques associés. La gestion des interruptions matérielles est prise en charge par l'application et non plus en s'appuyant sur le système d'exploitation. Les applications « Bare Metal » permettent ainsi de répondre aux contraintes drastiques de systèmes embarqués comportant un traitement numérique de l'image.

4. Bus I2C - Communication entre les modules camera et la carte Arduino Uno

La carte Arduino Uno et les deux modules caméra Open MV (fixe et mobile) communiquent leurs informations par l'intermédiaire d'une transmission série : le bus I2C. Le protocole de la liaison est du type Maître/Esclave. Chaque circuit est reconnu par son adresse et peut être soit transmetteur soit receveur de l'information.

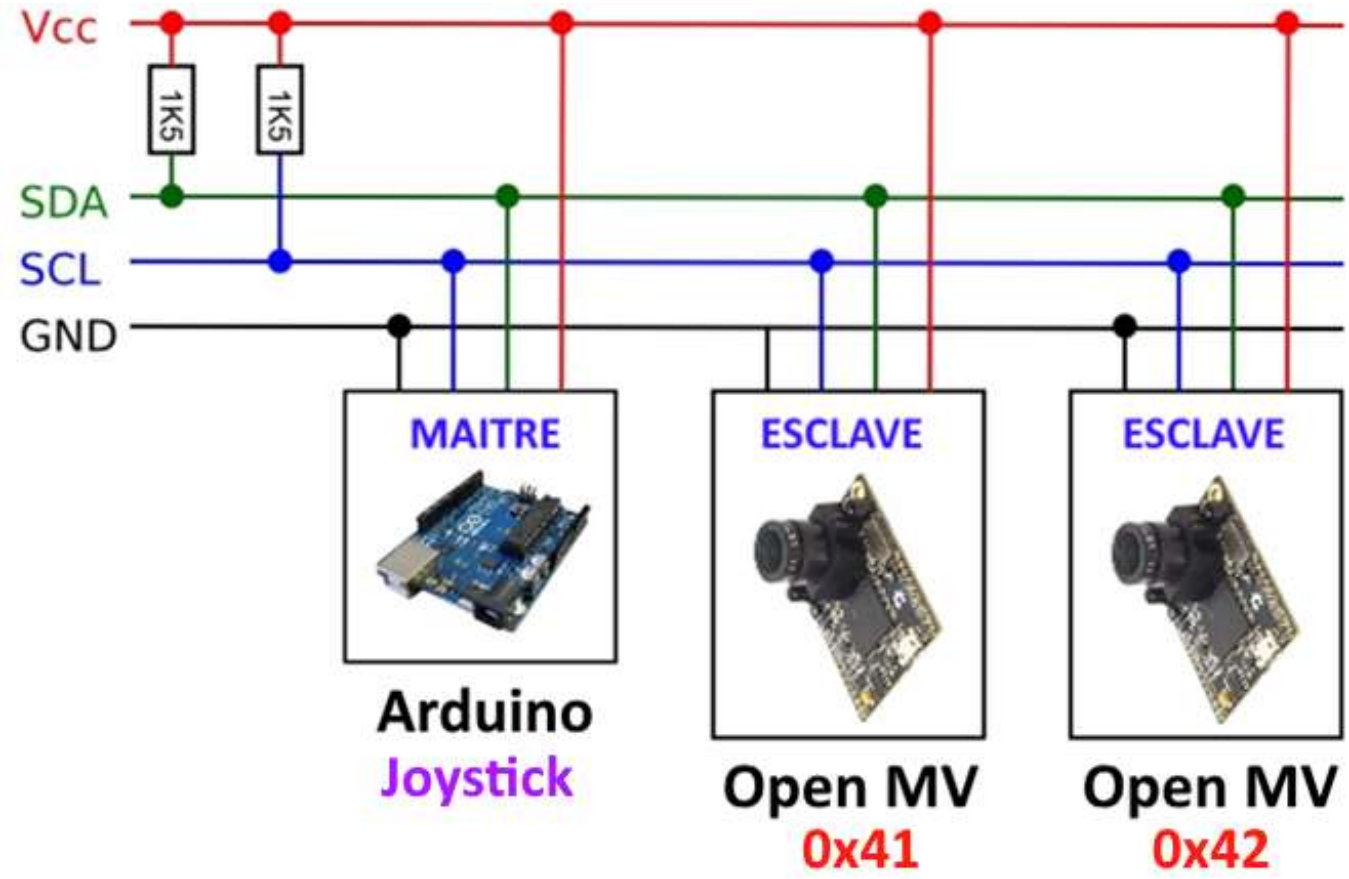
Dans le protocole du bus I2C le circuit Maître est celui qui demande un transfert d'information sur le bus et qui génère le signal d'horloge qui permet le transfert. Ainsi un circuit adressé est considéré comme un Esclave.

4.1 Principe du Bus I2C

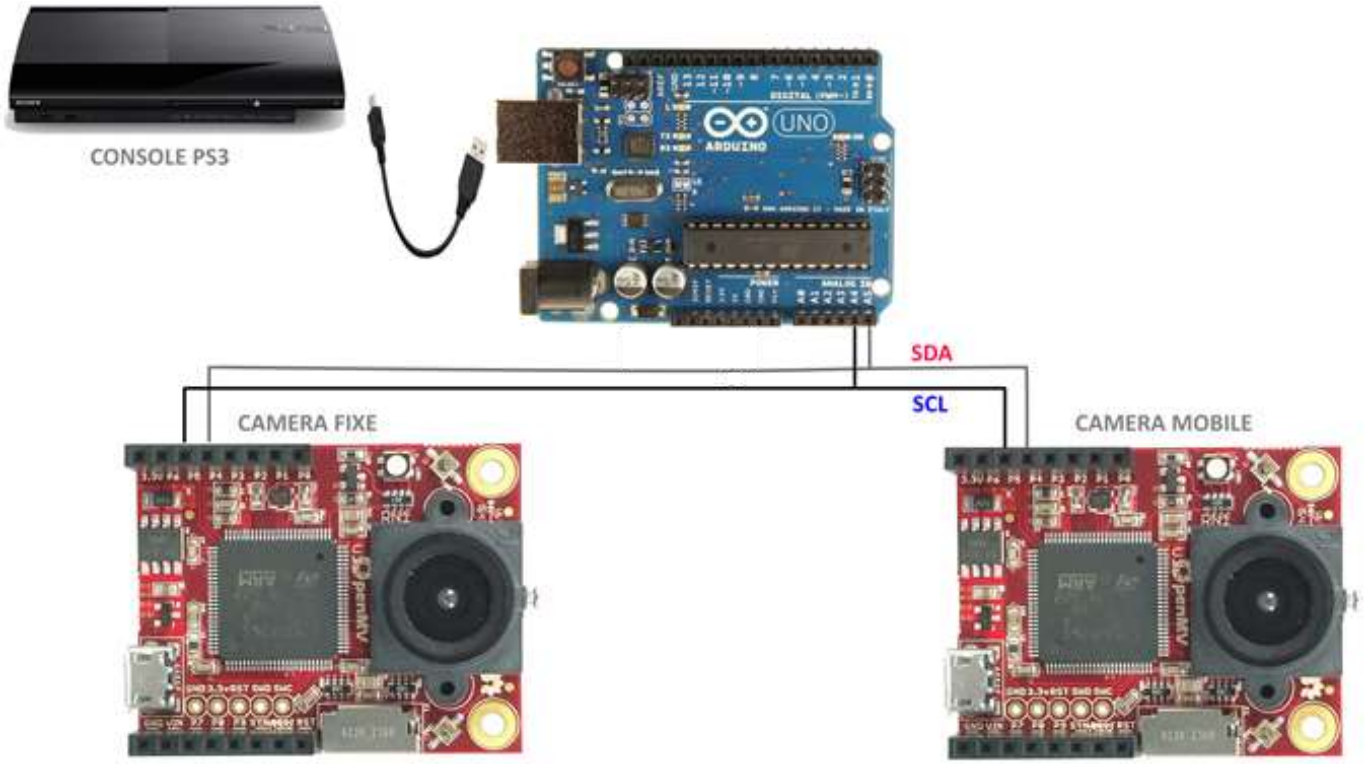
Le bus I2C est caractérisé par une liaison série synchrone réalisée à l'aide de 2 fils. C'est la société Philips qui a créé le concept au début des années 80. Son succès est lié à sa simplicité. Les données transitent par les lignes : SDA (Serial Data : donnée générée par le Maître ou l'Esclave) et SCL (Serial Clock : horloge imposée par le Maître). Son débit est de 100 kbits/s en mode Standard, de 400 kbits/s en mode Fast, de 1Mbits/s en mode Fast Plus et 3,4 Mbits/s en mode High Speed.

Afin de d'éviter les conflits électriques les Entrées/Sorties SDA et SCL sont de type « Drain Ouvert ». SDA et SCL sont des lignes bidirectionnelles, connectées à Vcc par l'intermédiaire de deux résistances de polarisation de 1,5 kΩ (résistances de tirage). Quand le bus est libre, c'est à dire quand il n'y a pas de transfert de données, les deux lignes SDA et SCL sont à l'état haut.

4.2 Organisation matérielle de la communication série



Du côté des caméras OpenMV on utilise les broches P4 pour SCL et P5 pour SDA. Du côté de la carte Arduino Uno, on utilise les broches analogiques A5 pour SCL et A4 pour SDA.



4.3 Organisation logicielle de la communication série

FORMAT DE LA TRAME I2C – MAITRE EN MODE LECTURE

Start	Address	R/W	Ack	Data0	Ack0	Data1	Ack1	...	DataN	AckN	Stop
	7 bits	1 bit	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit	8 bits	1 bit	

START	**Le Maître envoie sur le bus, une condition de START. (passage de l'état haut à l'état bas de SDA pendant que SCL est à l'état haut). Les Esclaves sont informés que le Maitre va débuter une communication.**
Address	**Le Maître place sur le bus, les 7 bits d'adresse [A6…. A0] de l'Esclave avec lequel il communique.**
R/W	**Le Maitre ajoute un bit le lecture/écriture (R/W : READ/WRITE). Il forme ainsi un octet avec les 7 bits d'adresse [A6… A0 R/W]**
Ack	**L'Esclave qui reconnaît son adresse [A6… A0] répond par un Ack à 0.**
$Data_0$	**L'Esclave transmet le premier octet [D7… D0] de donnée.**
Ack_0	**Le Maitre confirme la réception de la donnée par un Ack à 0.**
…	**……**
$Data_N$	**L'Esclave transmet le dernier octet [D7… D0] de donnée.**
Ack_N	**Le Maitre confirme la réception de la donnée par un Ack à 0.**
STOP	**La Maitre met fin à la transmission avec une condition de Stop. (passage de l'état bas à l'état haut de SDA pendant que SCL est à l'état haut). Les Esclaves sont informés que le Maitre met fin à la communication.**

4.4 Format des variables contenant les commandes du joystick

Le CdCF impose la reconnaissance des 4 touches indispensables pour pouvoir jouer à GranTurismo.

4.41 Format de l'information « Accélérateur – Frein »

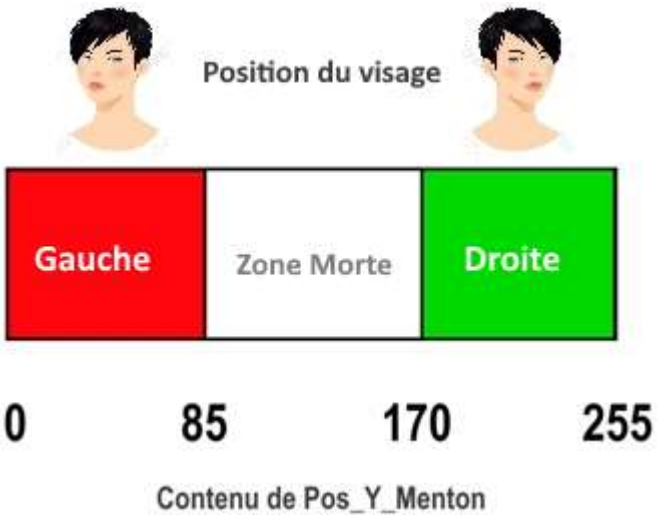
Pour délivrer ces 2 informations nous utilisons la caméra mobile qui permettra la détection du mouvement vertical du menton. Une variable entière « *Pos_Y_Menton* » codée sur un octet non signé contient une valeur entière image de la position verticale du menton du joueur.

Variable « Pos_Y_Menton »	Camera Mobile	Adresse Esclave bus I2C
	Position Menton	0x41
Freinage		
$0 \leq Pos_Y_Menton \leq 85$		
Zone Morte		
$86 \leq Pos_Y_Menton \leq 170$		
Accélération		
$171 \leq Pos_Y_Menton \leq 255$		

4.42 Format de l'information « Droite – Gauche »

Pour délivrer ces 2 informations nous utilisons la caméra fixe qui permettra la détection du mouvement vertical du menton. Une variable entière « *Pos_X_Visage* » codée sur un octet non signé contient une valeur entière image de la position axiale du visage du joueur.

Variable « Pos_X_Visage »	Camera Fixe	Adresse Esclave bus I2C
	Position Visage	0x42
Tourner à Gauche		
$0 \leq Pos_X_Visage \leq 85$		
Zone Morte		
$86 \leq Pos_X_Visage \leq 170$		
Tourner à Droite		
$171 \leq Pos_X_Visage \leq 255$		



5. Algorithmes des programmes Esclaves et Maitre

OPENCV CAM - MOBILE		OPENCV CAM - FIXE	
	0x41		0x42
	ESCLAVE		ESCLAVE
	Pos_Y_Menton		Pos_X_Visage
LANGAGE	MICRO PYTHON	ENVIRONNEMENT DEVELOPPEMENT	OPEN MV

ALGORITHMES SIMPLIFIES

CHARGEMENT DE LA BIBLIOTHEQUE PYB
ACTIVER CAMERA MOBILE MODE ESCLAVE,@0x41
DESACTIVER CAMERA MOBILE
ACTIVER CAMERA MOBILE MODE ESCLAVE,@0x41

DEBUT_ESCLAVE_CAMERA_MOBILE
APPEL SOUS-PROGRAMME DETECTION_POSITION_MENTON
ATTENTE REQUETE MAITRE
SI REQUETE DU MAITRE
ECRITURE DE POS_Y_MENTON SUR BUS
FIN_ ESCLAVE_CAMERA_MOBILE

CHARGEMENT DE LA BIBLIOTHEQUE PYB
ACTIVER CAMERA FIXE MODE ESCLAVE,@0x42
DESACTIVER CAMERA FIXE
ACTIVER CAMERA MOBILE MODE ESCLAVE,@0x42

DEBUT_ESCLAVE_CAMERA_FIXE
APPEL SOUS-PROGRAMME DETECTION_POSITION_VISAGE
ATTENTE REQUETE MAITRE
SI REQUETE DU MAITRE
ECRITURE DE POS_X_VISAGE SUR BUS
FIN_ ESCLAVE_CAMERA_FIXE

ARDUINO UNO REV 3			
		MAITRE	
LANGAGE	ARDUINO	ENVIRONNEMENT DEVELOPPEMENT	ARDUINO

ALGORITHME SIMPLIFIE

DECLARATION DES VARIABLES
CHARGEMENT DE LA BIBLIOTHEQUE I2C

DEBUT_INITIALISATION
INITIALISATION DE LA LIAISON SERIE A 19200 BAUDS
INITIALISATION LE BUS I2C
TEMPORISATION 1S
FIN_ INITIALISATION

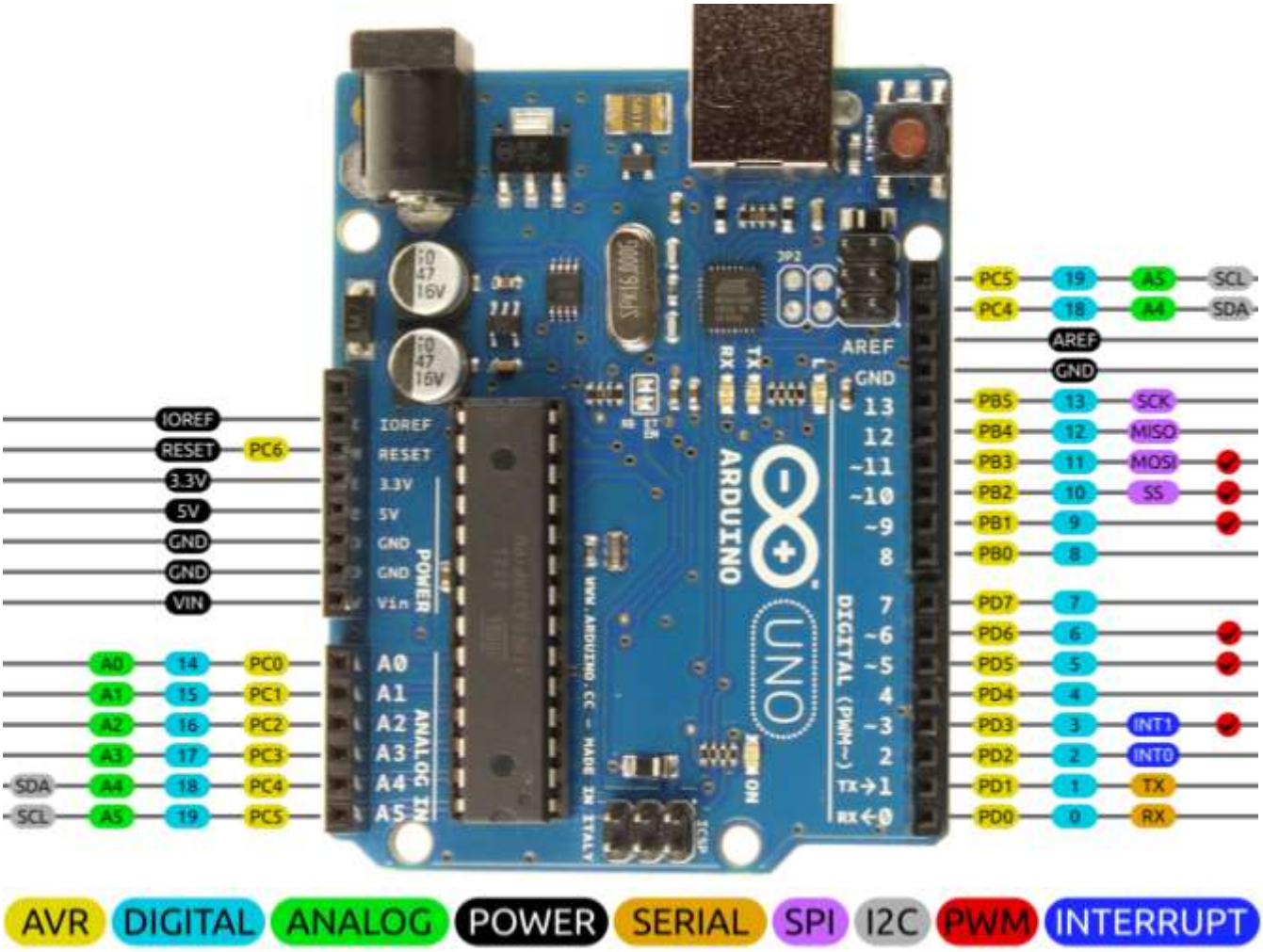
DEBUT_MAITRE
REQUETE VERS ESCLAVE CAMERA FIXE
LECTURE DE POS_X_VISAGE
REQUETE VERS ESCLAVE CAMERA MOBILE
LECTURE DE POS_Y_MENTON
ECRITURE DE POS_X_VISAGE VERS CONSOLE
ECRITURE DE POS_Y_MENTON VERS CONSOLE
TEMPORISATION 1MS
FIN_MAITRE

6. Procedure de transformation de la carte Arduino Uno en un peripherique USB

La carte Arduino Uno est une carte architecturée autour d'un microcontrôleur ATMEL, l'ATmega328.

6.1 Principales caractéristiques de la carte

Arduino Uno dispose de 14 broches numériques d'entrées-sorties (6 peuvent être utilisées en sorties Modulation en Largeur d'Impulsion), de 6 entrées analogiques qui peuvent également être utilisées en broches entrées-sorties numériques,d'un quartz 16Mhz,d'une connexion USB,d'un connecteur d'alimentation jack,d'un connecteur ICSP (programmation « in-circuit »), de 4 lignes I2C (2 broches SDA : Serial Data et 2 broches SCL : Serial Clock) et d'un bouton de réinitialisation (Reset).



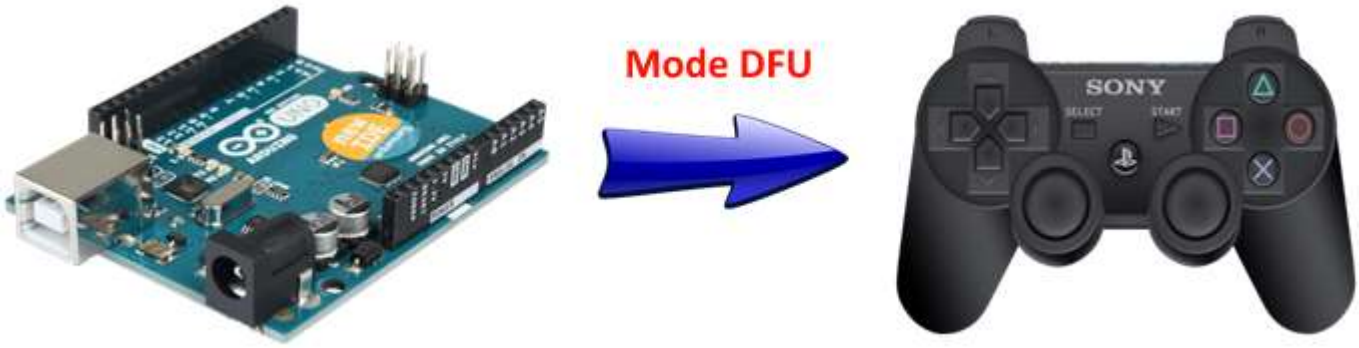
Le microcontrôleur ATMEGA328 présent sur la carte Arduino Uno est livré avec un bootloader. Un bootloader est un programme installé en principe une fois pour toute dans l'ATMEGA328 et qui permet la communication avec l'IDE Arduino (installé sur notre ordinateur) via un port série.

Le protocole STK500 développé par la société ATMEL est utilisé pour assurer cette communication asynchrone (Vitesse de la transmission : 115 200 bauds - Format de la transmission : 8 bits de données, 1 bit de stop, pas de bit de parité).

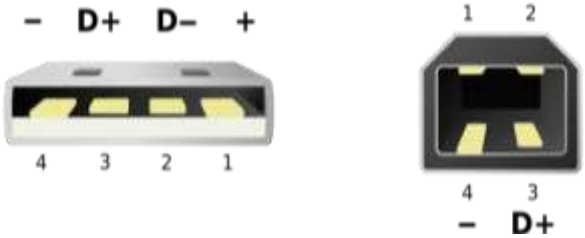
La carte Arduino Uno utilise également un deuxième microcontrôleur l'ATMEGA8U2, contenant 16Ko de mémoire flash, programmé en convertisseur « USB vers Série ». Ce composant traite les informations reçues par le biais de l'USB et l'ATMEGA328.

6.2 Transformation de la carte Arduino Uno en manette de jeu (joystick)

Pour que la carte Arduino Uno soit en mesure de transmettre les 4 commandes (accélérateur, frein et virages à droite et à gauche) à la console de la Playstation il faut que la carte Arduino Uno soit reconnue par la console comme un périphérique USB (manette ou joystick). Pour effectuer cette opération, il faut travailler en mode DFU (Device Firmware Update).



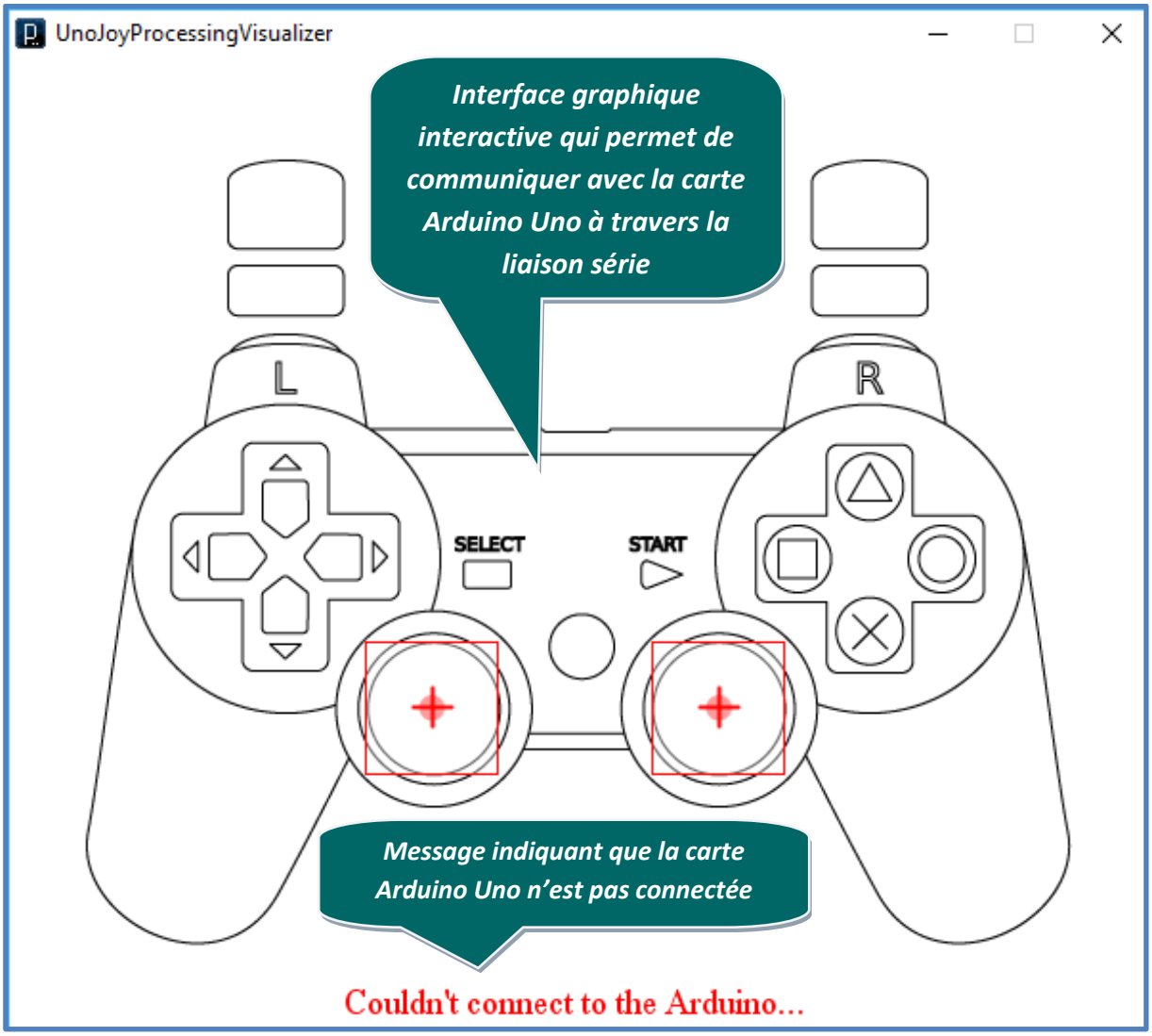
Le mode DFU est un protocole, développé en commun par plusieurs fabricants dont ATMEL, pour permettre la mise à jour du logiciel, appelé Firmware et embarqué dans le microcontrôleur, avec un ordinateur et un port USB. Ce protocole n'est pas limité aux microcontrôleurs ATMEL mais est aussi utilisé pour mettre à jour de nombreux appareils comme les smartphones par exemple. Le code DFU est installé d'origine dans les microcontrôleurs ATMEGA « U ». Il suffit donc des 2 fils de données D+ et D- et des deux et d'alimentation pour qu'un ATMEGA « U » puisse entrer en communication avec un ordinateur, et que ce dernier soit prêt à charger un programme dans le microcontrôleur.



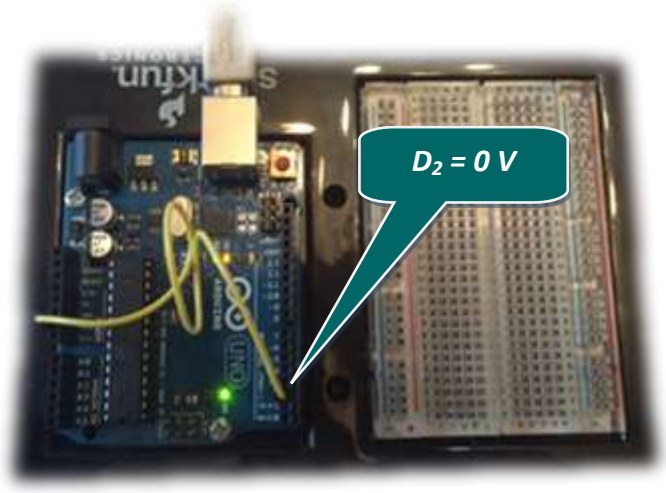
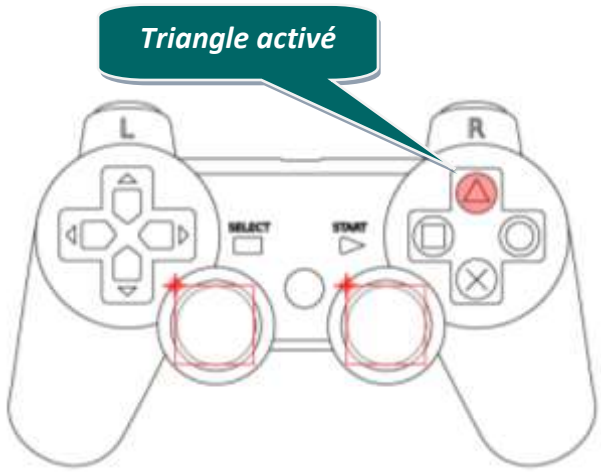
6.21 Initialisation Logicielle

Dans un premier temps, il convient de charger le sketch « *UnoJoyArduinoSample.ino* » dans l'IDE Arduino. Dans ce programme on configure toutes les entrées numériques en entrée (à l'exception des deux lignes de la liaison série) en validant la résistance de rappel respective. Ce programme, qui utilise les commandes de la bibliothèque open source « *Unojoy* », doit être compilé puis téléversé ver la carte Arduino Uno.

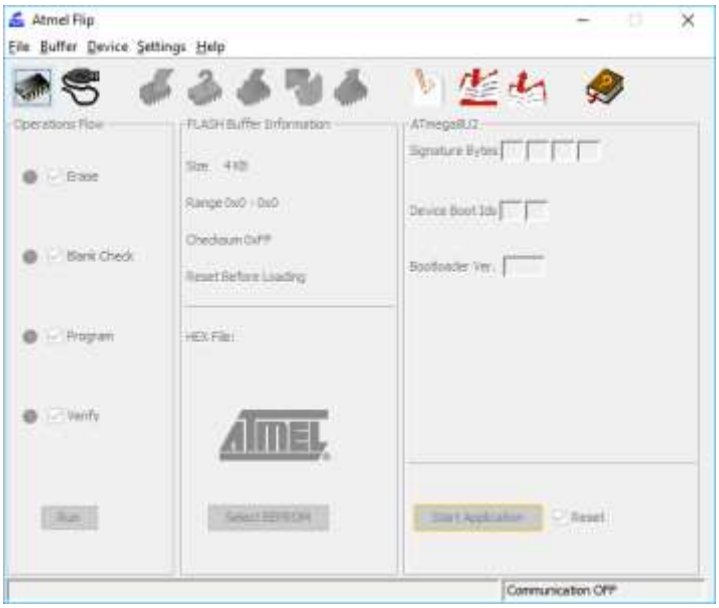
Ensuite il faut lancer l'application « *UnoJoyProcessingVisualizer.exe* » un exécutable généré à partir de l'IDE Processing qui permet de visualiser, par l'intermédiaire d'une interface graphique, le lien entre les entrées numériques de la carte Arduino Uno et la manette de jeu.



6.22 Vérification Matérielle : test du bouton « Triangle » de la manette

Vérification : Entrée numérique D2 forcée à l'état bas → Mise en surbrillance du bouton « Triangle »	
	

6.23 Transfert du bootloader dans l'ATMEGA8U2 de la carte Arduino Uno



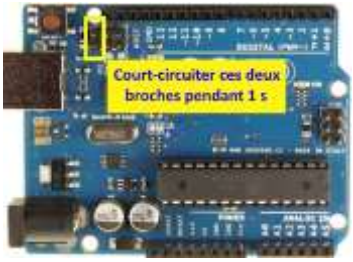
Dans la carte Arduino Uno, le Firmware de l'ATMEGA8U2 est chargé de la conversion « USB vers Série ». Il nous faut donc le remplacer par un Firmware qui permette à la console de reconnaitre la carte Arduino Uno comme une manette de jeu. Pour effectuer cette opération nous devons utiliser le logiciel Flip. A noter que cette procédure est entièrement réversible.

Dans un premier temps, il faut installer le logiciel Flip (Flexible In-system programmer – interface représentée ci-contre) développé par ATMEL.

6.231 Passage de la carte Arduino Uno en mode DFU (Device Firmware Update)

Pour passe en mode DFU il faut :

- **d'abord de court-circuiter pendant une seconde environ les deux broches situées à proximité du connecteur USB comme indiqué ci-contre,**
- **ensuite installer les pilotes nécessaires au logiciel Flip,**
- **et enfin transférer le nouveau Firmware dans l'ATMEGA8U2**



6.232 La carte Arduino Uno est reconnue comme un Joystick

Avant… une carte ARDUINO	Après… un JOYSTICK
Périphériques (3) Cruzer Blade MARYSE USB-PS/2 Optical Mouse Périphériques multimédias (1) Freebox Server Non spécifié(e) (1) Arduino Uno (COM4)	Imprimantes (5) Envoyer à OneNote 2010 Fax HP ePrint Microsoft Print to PDF Microsoft XPS Document Writer Périphériques (4) UnoJoy Joystick Cruzer Blade MARYSE USB-PS/2 Optical Mouse Périphériques multimédias (1) Freebox Server

7. CONCLUSION

L'essor exponentiel des nouvelles technologies de l'information et de la communication à ouvert, notamment dans le domaine de la compensation du handicap, des perspectives et des horizons qui relevaient, il n'y a encore pas très longtemps, essentiellement des domaines de l'imaginaire et de la science-fiction.

Notre projet s'inscrit pleinement dans cette évolution en essayant d'apporter une solution à un problème relevant principalement du domaine « social », l'un des trois piliers du développement durable cher aux élèves de la filière STIDD que nous sommes.

L'objectif fixé dans la « note de cadrage » a consisté à répondre de manière adaptée à la question posée par la problématique du Docteur Martinel : « *Comment permettre à une personne tétraplégique de jouer à un jeu vidéo à l'aide d'un dispositif sans contact ?* ».

A la date du 8 Mars 2017, où en sommes-nous exactement ? Au moment où ce document de synthèse est mis en ligne le projet n'est pas encore entièrement opérationnel. Actuellement, notre travail se concentre essentiellement sur la programmation de la transmission du bus I2C. Le transfert des données entre les deux caméras OpenMV et la carte Arduino est du type Maitre-Esclave. Les autres tâches sont dès à présent terminées et fonctionnelles, à savoir :

- **la détection des commandes (Droite-Gauche et Accélération-Frein),**
- **transformation de la carte Arduino en joystick,**
- **et la liaison série USB entre la carte Arduino et la console Playstation.**

Les difficultés que nous rencontrons sont principalement liées à des problèmes inhérents à la maitrise de la programmation des échanges de données qui transitent sur le bus I2C. En effet, l'écriture des codes doit être réalisée au moyen de deux langages différents, à savoir :

- **côté Maitre, il s'agit de coder les lignes du programme en utilisant le langage Arduino. Cet environnement nous est certes relativement familier mais pas au point de rendre la tâche évidente et aisée,**
- **côté Esclaves, l'apprentissage du langage Micro Python a été très chronophage. D'autant plus que les « ressources pédagogiques » ne sont pas forcément immédiatement mobilisables car ce langage n'est pas enseigné au Lycée Condorcet de Schœneck dans les classes de Terminale STIDD Option SIN.**

Quoi qu'il en soit notre volonté d'aboutir n'est aucunement altérée par ces réelles difficultés techniques qui si l'on en croit nos professeurs dépassent parfois ce qu'un élève « moyen » de notre niveau scolaire est capable de maitriser. En clair, nous touchons là à des limites technologiques liées à notre niveau scolaire.

Mais, pour ce qui nous concerne, les dimensions solidaire et humaine de ce projet constituent des sources de motivation inégalables. Pouvoir apporter un peu de sérénité à Ludovic constitue, en réalité, le véritable « carburant » de notre implication dans ce projet.

SITOGRAPHIE

OpenMV Team, A.K.A Professional Pixel Manipulators, 2013
OpenMV | Small - Affordable - Expandable
https://openmv.io/products/openmv-cam

MicroPython documentation, 2017
Documentation for MicroPython v1.8, last updated 26 Jan 2017.
http://docs.openmv.io/

Référence du langage Arduino en français, 2010
http://playground.arduino.cc/French/Reference

I2C manual - NXP Semiconductors, 2003
http://www.nxp.com/documents/application_note/AN10216.pdf

Référence Arduino français Main/Librairie Wire - Mon Club-Elec, 2013
http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.LibrairieWire

DFU Programmer - Arduino, 2017
Updating the Atmega8U2 and 16U2 on an Uno or Mega2560 using DFU
https://www.arduino.cc/en/Hacking/DFUProgramming8U2

FLIP - Atmel, 2016
Supports in-system programming of flash devices
http://www.atmel.com/tools/flip.aspx

UNOJOY – A USB JOYSTICK FOR MAC, PC, LINUX OR PS3, 2012
https://blog.arduino.cc/2012/05/14/unojoy-a-usb-joystick-for-mac-pc-linux-or-ps3/

Tutorials \ Processing.org
https://processing.org/tutorials/

Face detection con OpenCV 2.4 (Haar-like Feature based), 2013
https://brainsnippetsdotorg.wordpress.com/2013/03/31/face-detection-con-opencv-2-4-haar-like-feature-based/

Eduscol - Technologie 179 - avril 2012
SysML : les diagrammes
http://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/techniques/979/979-179-p100.pdf