Rapport de mini projet

Réaliser par:

Trifi Bechir

Henchir Elaa

Mnasri Yassine

Institut Supérieur des Technologies de l'Information et de la Communication

SOMMAIRE

Chapitre 1 : Présentation générale

1. Introduction :

Dans le cadre de notre seconde année en Système embarqué et internet des objets à l'ISTIC, il nous est proposé un projet qui consiste à programmer un robot appelé Robotino pour chercher un objet de couleur et approcher ce dernier à l'aide de webcam.

Notre objectif est de tester les limites en traitement d'images.

Les étapes nécessaires sont :

- Détecter un objet rouge à travers des fonctions de traitement d'images.

- Etablir les conditions nécessaires pour détecter la couleur.

- Détecter une zone colorée qui représente un but.

- Diriger Robotino vers l'objet colorée.

Ces étapes ont été les premières parties de notre programmation, ensuite notre objectif était de rendre ce programme plus robuste.

2. Définition :

La robotique est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques. C'est un ensemble des domaines scientifiques et industriels en rapport avec la conception et la réalisation de robots.

Un robot est un dispositif mécatronique conçue pour exécuter une ou plusieurs tâches à plusieurs reprises, avec rapidité et précision.

Dans ce mini projet on va utiliser le robot « Robotino » qui est basé sur un assemblage d'entraînement omnidirectionnel, qui permet au système de se déplacer librement. Le robot est contrôlé par un système PC standard de l'industrie, qui est assez puissant pour planifier des itinéraires pour une conduite entièrement autonome. Via un WLAN-Link, Robotino peut envoyer toutes les lectures de capteurs à un PC externe. Dans l'autre sens, les commandes de commande peuvent être émises par le PC externe. De cette façon, les programmes de contrôle peuvent s'exécuter sur le PC

externe ou sur Robotino directement. Un mode mixte ou un contrôle partagé sont également possibles.

Figure 1 : Robotino

3. Domaines d'applications :

Il est possible d'acquérir des connaissances dans les domaines suivants :

• **Mécanique**

– Architecture mécanique d'un système robotique mobile

• **Electrotechnique**

– Commande de moteur

– Mesure et exploitation de diverses grandeurs électriques

• **Capteurs**

– Commande de trajectoire gérée par capteurs

– Commande de trajectoire sans collision à l'aide de capteurs de distance

4. Structure d'un robot mobile :

4.1. Structure général :

En général un robot mobile est constitué dc trois structures :

- **Structure mécanique:** Elle assure le mouvement du robot par des roues motrices placées selon le type de mouvement et la précision de la tache voulue.
- **Structure instrumentale:** Un robot est équipé d'un certain nombre de capteurs de sécurité afin de leur donner une certaine connaissance de l'environnement.
- **Structure informatique:** Une commande numérique est impérative, afin de bien analyser les différentes informations, soit du système de perception ou de localisation. Cette commande peut être à base d'un microprocesseur ou microcontrôleur.

4.2. Les composants :

Capteurs :

- Une structure de protection en caoutchouc intégrant un détecteur anticollision

- Neuf capteurs de distance analogiques à infrarouge :

Entrées	Type	Unité	Standard	Description
Sorties				
Valeur	float	Volt		Indique, en V, la tension analogique délivrée par le capteur. La mise à l'échelle et la conversion de ces valeurs en une valeur de distance ayant la dimension d'une longueur doivent être réalisées par l'utilisateur.
Orientation	float	Degré		Orientation du capteur dans le système de coordonnées local de Robotino. La valeur se calcule avec le numéro du capteur de distance selon la formule orientation = 40° x (numéro - 1)

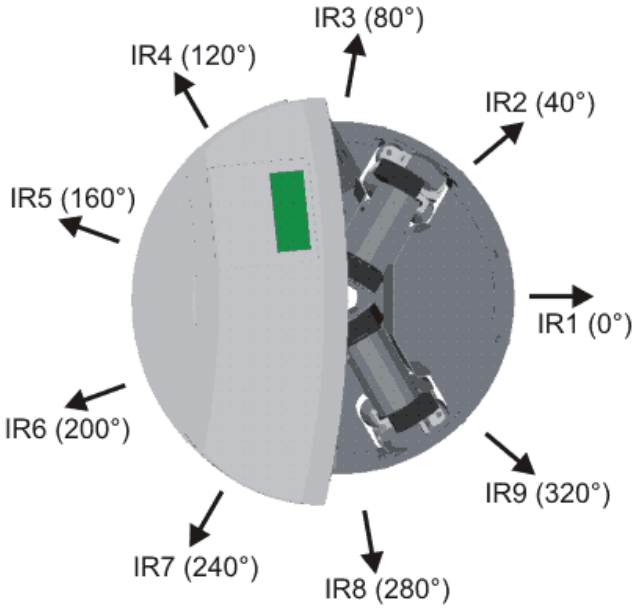


Figure 2 : Capteurs de distances

- Un capteur inductif analogique

- Deux capteurs optiques numériques

- Une caméra Web couleur à interface USB et compression jpeg

<u>Système moteur :</u>

1- Entraînement omnidirectionnel :

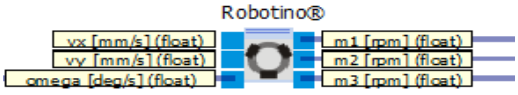


Figure 3 : La commande des moteurs

- Ce bloc permet de calculer la vitesse de consigne des moteurs de Robotino sur la base de la consigne de vitesse dans les axes x et y ainsi que de la consigne de la vitesse de rotation.

2- Les moteurs :

- Il s'agit de l'un des trois moteurs de Robotino. On trouve le numéro de chaque moteur dans le symbole graphique du bloc de fonction.

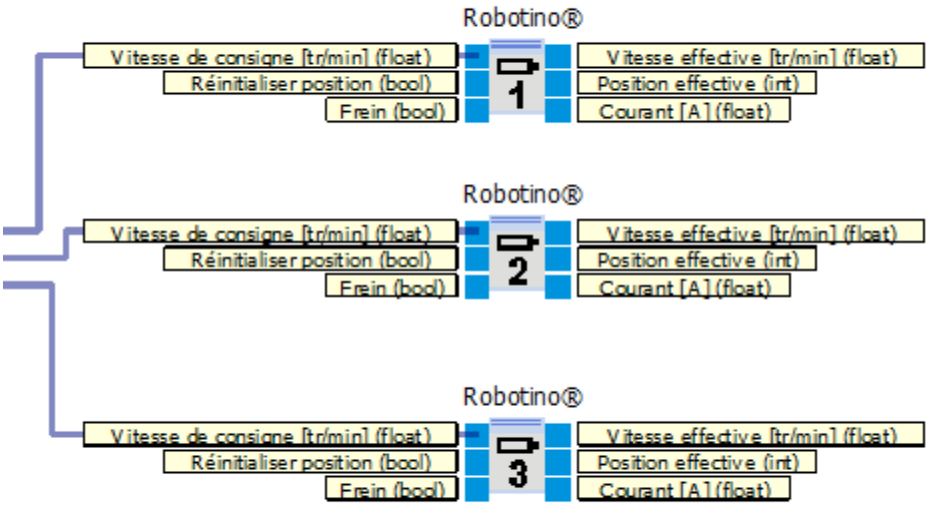


Figure 4 : Les blocs moteurs

Commande :

- Un PC 104 plus embarqué avec OS Linux Ubuntu temps réel et plusieurs interfaces de communication :

- Ethernet

- Deux ports USB

- Deux liaisons RS232

- Un port parallèle

- Un connecteur VGA

- Un Point d'accès Wi-Fi performant avec antenne, que l'on peut faire basculer en mode client et avoir un Cryptage WPA2 en option.

5. Problématique :

Comment concevoir et réaliser, de manière rapide, simple, et efficace, un robot répondant à nos besoins ?

Chapitre 2 : Réalisation du projet :

1. Conception :

1.1. Cahier des charges :

Fonctionnement :

- Positionnez la webcam.
- Connecter la caméra au port USB du boîtier de commande.
- Vérifier le bon fonctionnement de la webcam.
- Créer un programme de test dans Robotino® View.
- Établir une connexion Wi-Fi à Robotino®.
- Démarrer le programme.
- Rotation de Robotino® jusqu'à ce que le centre de gravité du segment se trouve au centre de l'image
- Marche avant en direction du centre de gravité du segment jusqu'à la distance optimale déterminée.

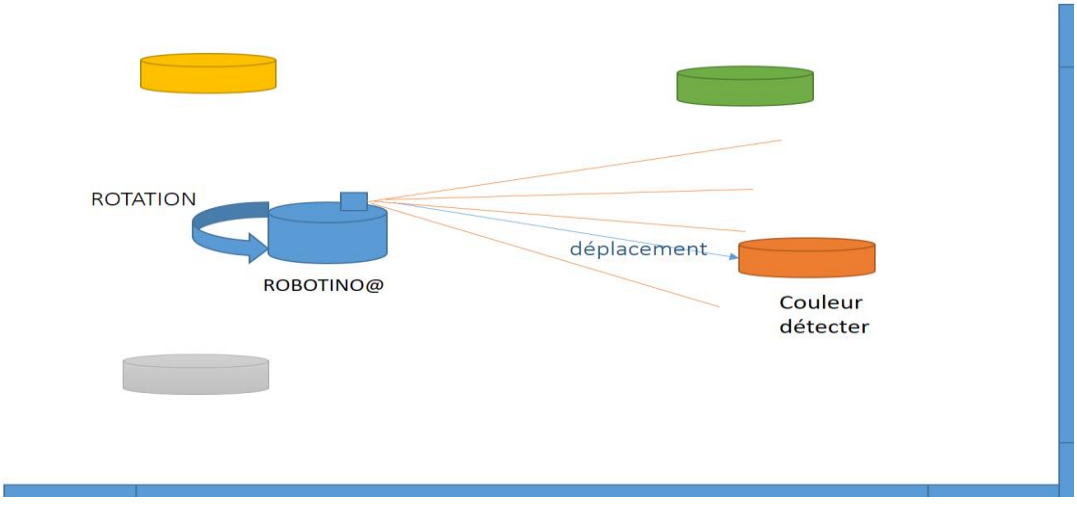
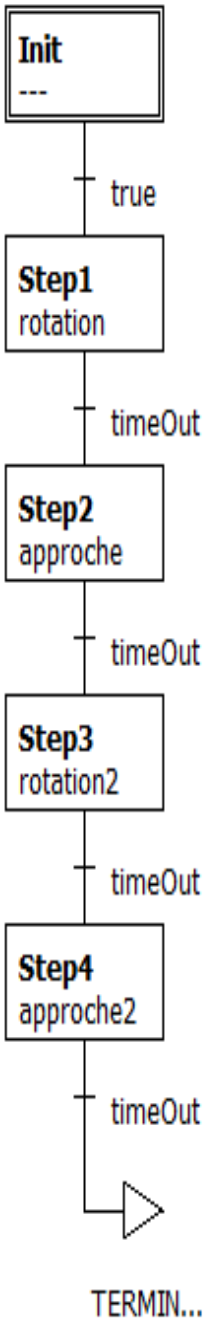


Figure 5 : Principe de fonctionnement

1.2. Algorithme :

- Initialisation
- Passage à l'étape suivante
- Step 1 : Rotation ; Ce sous-programme tourne Robotino vers l'objet coloré
- timeOut : Condition de rupture de la rotation.
- Step 2 : Approche ; Ce sous-programme conduit Robotino à l'objet coloré et s'arrête dès que la zone de l'objet dans l'image de la caméra dépasse un certain seuil.
- timeOut : Condition de rupture de l'approche.
- Step 3 : Deuxième rotation ; Ce sous-programme tourne Robotino vers le deuxième objet coloré
- timeOut : Condition de rupture de la deuxième rotation.
- Step 4 : Deuxième approche ; Ce sous-programme conduit Robotino à le deuxième objet coloré et s'arrête dès que la zone de l'objet dans l'image de la caméra dépasse un certain seuil.
- timeOut : Condition de rupture de la deuxième approche.
- TERMINATE : Le programme principal se termine par un saut à Terminate.



2. Programmation :

Approche :

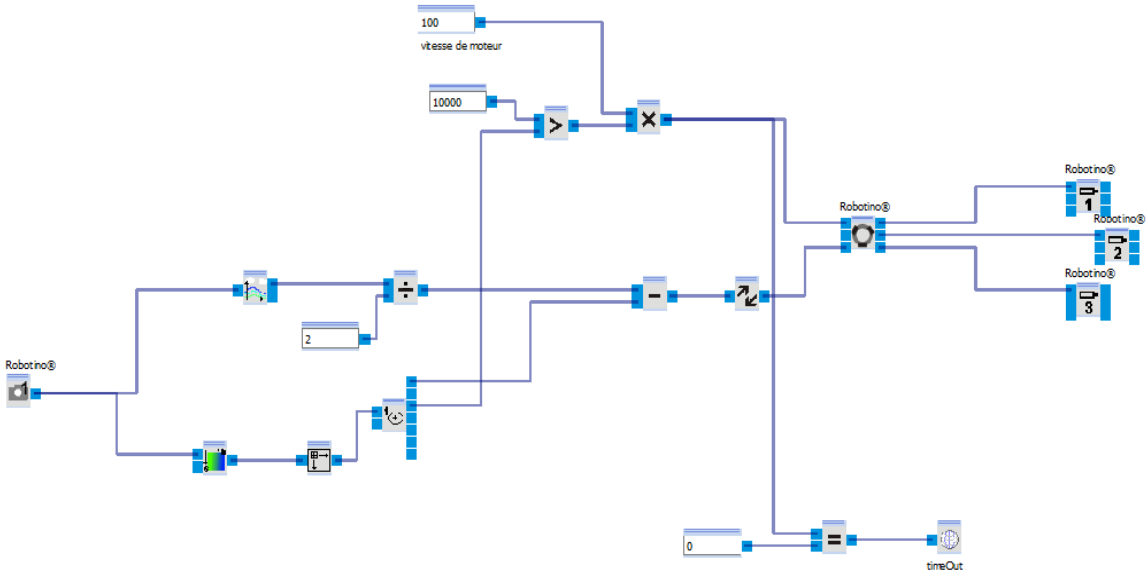


Figure 6 : approche

Rotation :

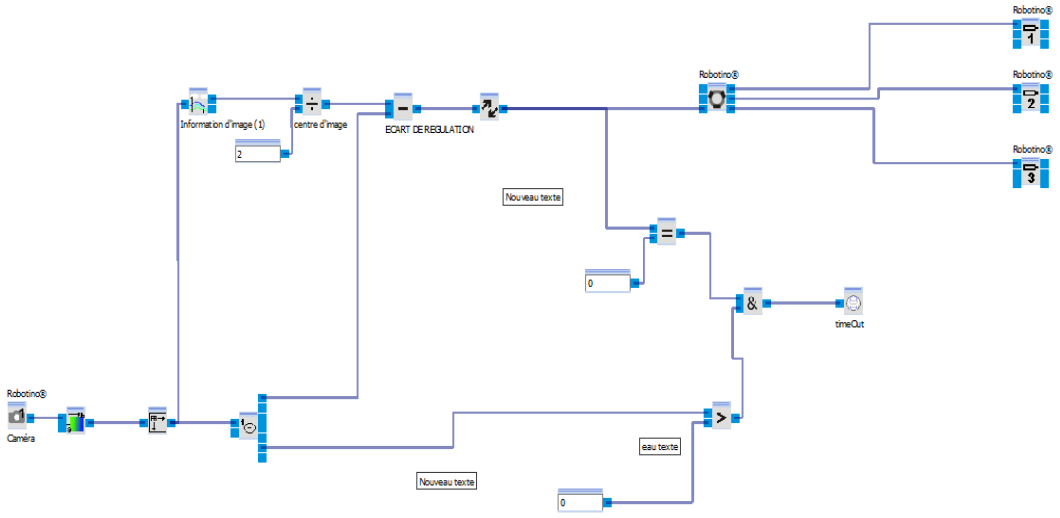


Figure 7 : rotation

3. Construction :

3.1. Approche :

3.1.1. Les blocs nécessaires :

3.1.1.1. Bloc 1 :

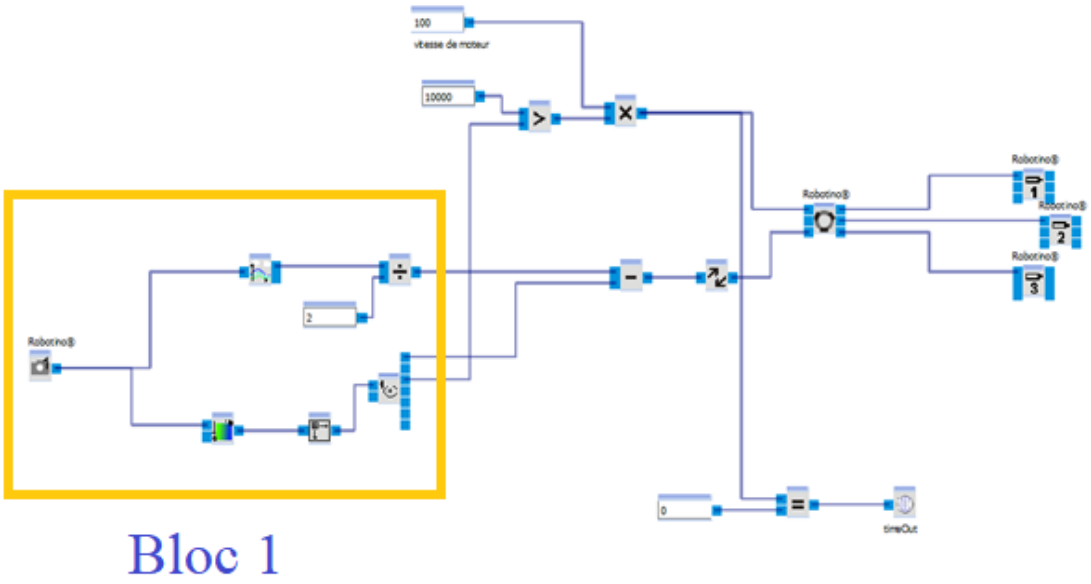


Figure 8 : Bloc 1

Traitement d'image :

On obtient une image grâce à la camera USB placé au-dessus du Robotino. On peut visualiser et transmettre le résultat avec ce bloc :

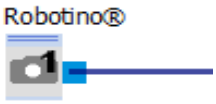


Figure 9 : camera

Information d'image :

Ce bloc a une seule entrée et deux sorties ;

Une entrée est de type image

Deux sorties :

- Hauteur de type int ; pour donner la hauteur de l'image en pixel.
- Largeur de type int ; pour donner la largeur de l'image en pixel.

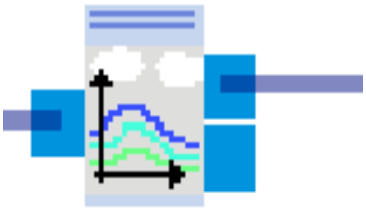


Figure 10 : Info Image

Recherche de plage colorimétrique:

Ce bloc permet de sélectionner sur l'image une zone de couleur et en relâchant le bouton de la souris, le bloc garde en mémoire la plage colorimétrique choisie au format TSV (Teinte, Saturation, Valeur).

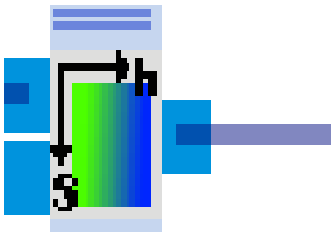


Figure 11 : Plage colorimétrique

Dans ce bloc on a :

Deux entrées :

- Une entrée de type image.
- Plage colorimétrique.

Une seule sortie :

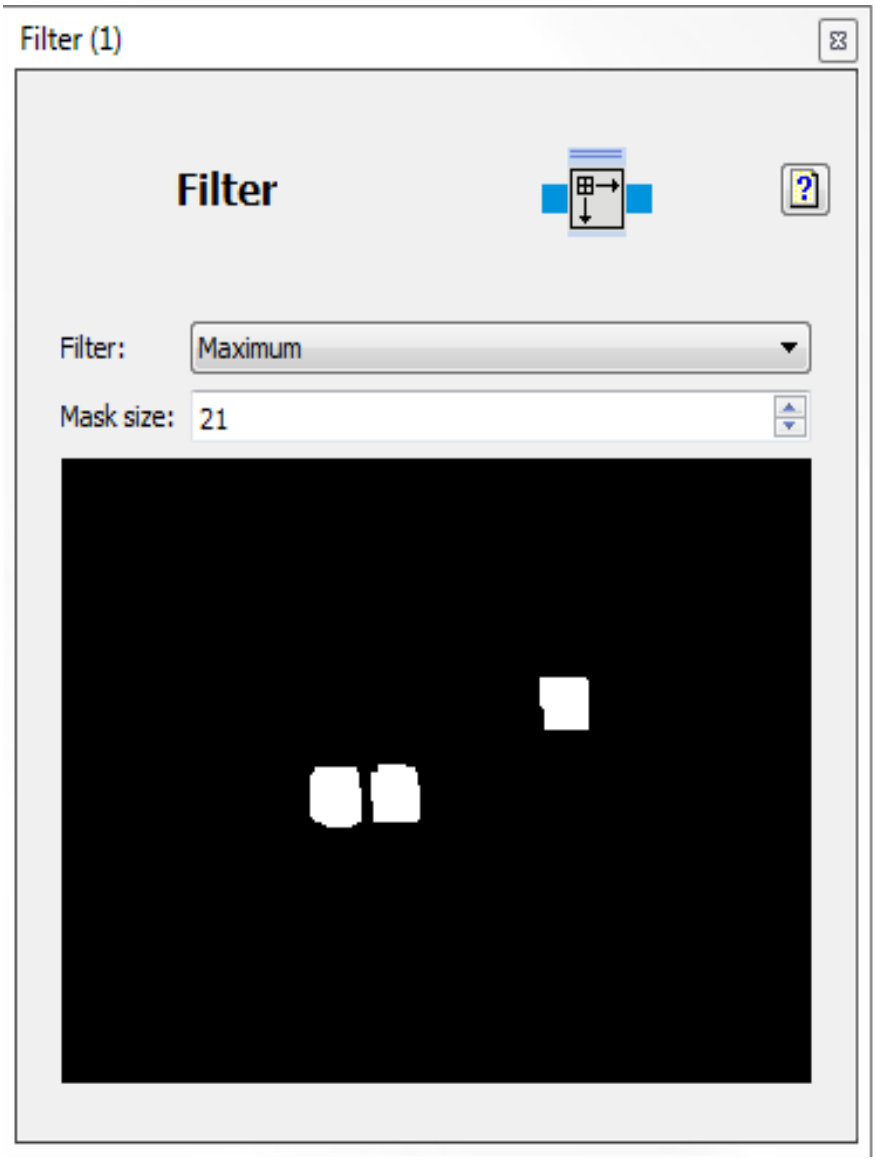
- Image

Filtre:

Dans notre cas, on va utiliser le filtre minimum ; qui va mettre la valeur colorimétrique du pixel actuel au minimum des valeurs des pixels au sein du masque. La taille de l'image de sortie est réduite en hauteur et en largeur de la taille de masque-1.

Figure 12 : Filtre

On a comme entré une image et comme sortie une image filtrée.


Filter (1)
Filter
Filter:
Maximum
Mask size:
21

Traqueur de segment:

Ce bloc de fonction trouve dans une image en noir et blanc, telle que générée par la recherche de plage colorimétrique, des segments d'un seul tenant et traque un segment défini. Pour traquer tous les segments d'une image en noir et blanc, il suffit de connecter en série le nombre de traqueurs de segment voulu.

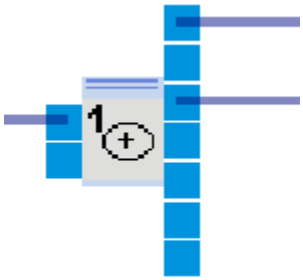


Figure 13 : Traquer de segments

Dans ce bloc on a :

Deux entrées :

- Entrée est de type image.
- Redémarrage de type booléen.

Sept sorties :

- X de type int.
- Y de type int.
- Surface de type int.
- Vx de type int.
- Vy de type int.
- Nombre de segments trouvés de type int.
- Sortie de type image.

Division :

Ce bloc permet de mettre l'image toujours centré.

Il prend comme entrée la largeur de l'image et la divise par « 2 ».

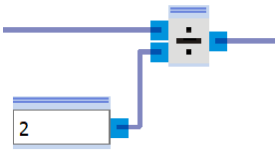


Figure 14 : division

3.1.1.2. Bloc 2 :

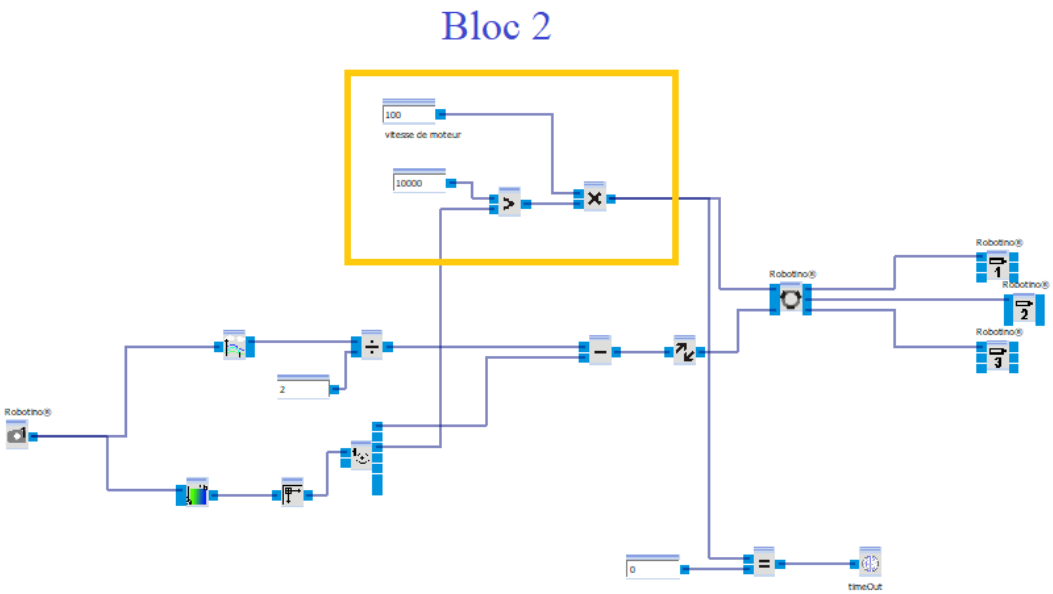


Figure 15 : Bloc 2

Avancement du robot :

- Cette fonction permet de comparer la surface du cylindre rouge en pixels par rapport à 10000 pixels. Si la zone est inférieure à 10000 pixels, la sortie est vrai. Si la zone est supérieure à 10000 pixels la sortie prend 0.

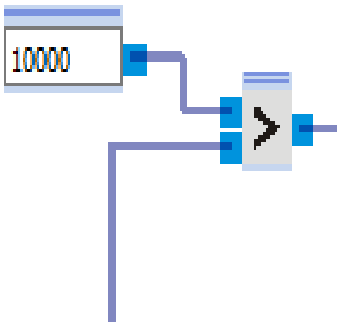


Figure 16 : Supérieur

- La fonction de multiplication prend comme entrer la sortie de fonction de comparaison, quel que soit l'état 1 ou 0, et la multiplie par 100.

Figure 17 : multiplication

3.1.1.3. Bloc 3 :

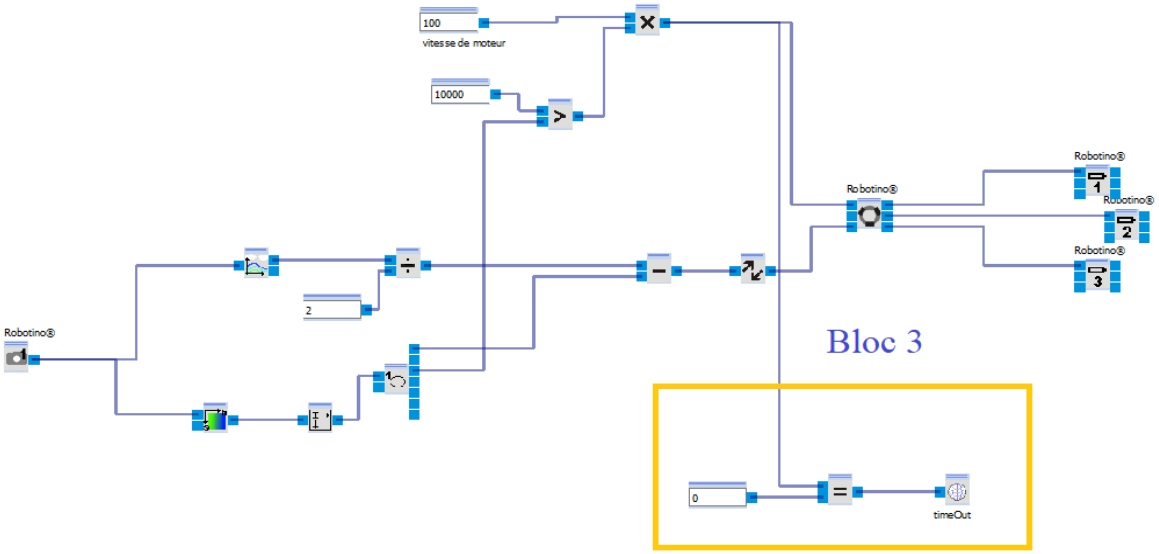


Figure 18 : Bloc 3

Fonction de rupture :

- Cette fonction prend entrer la valeur donnée par bloc 2. Si cette valeur est égale à 0, alors la condition de rupture est vrai et on passe à timeOut. Sinon, la vitesse d'avancement par rapport à x est 100mm/s. Cette valeur sera liée à l'entré Vx du moteur.

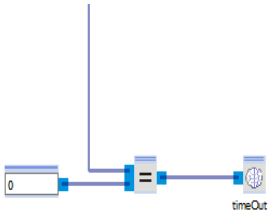


Figure 19 : Fonction de rupture

3.2. Rotation :

3.2.1. Les blocs nécessaires :

3.2.1.1. Bloc 1 :

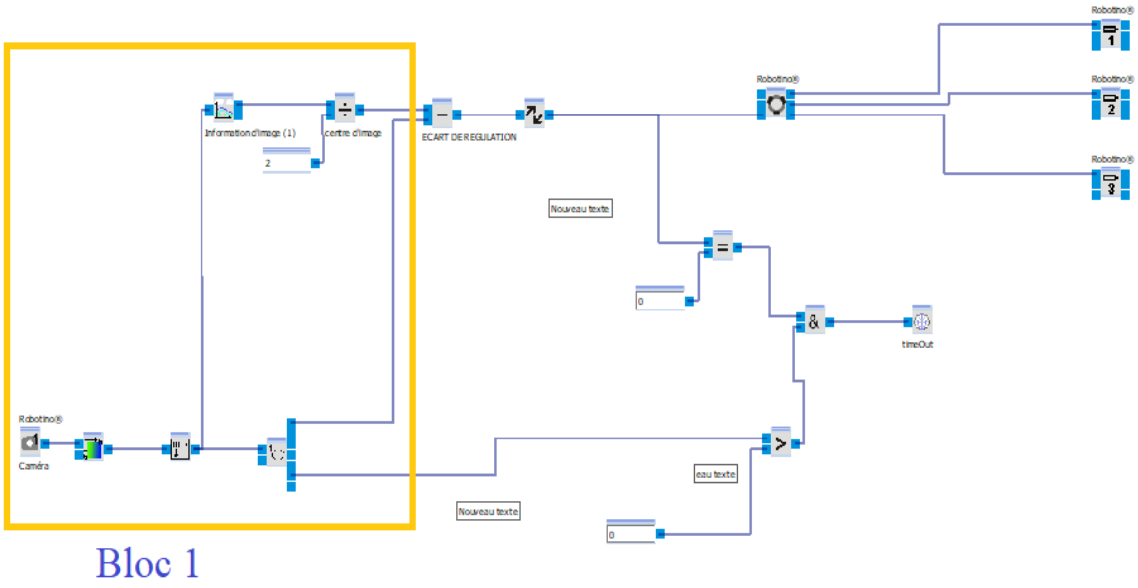


Figure 20 : Bloc 1

Traitement d'image :

On obtient une image grâce à la camera USB placé au-dessus du Robotino. On peut visualiser et transmettre le résultat avec ce bloc :

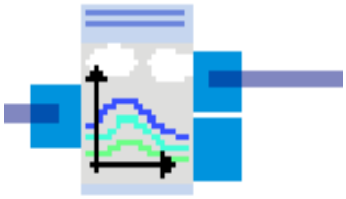
Information d'image :

Ce bloc a une seule entrée et deux sorties ;

Une entrée est de type image

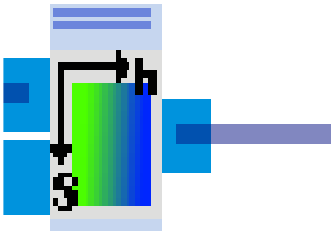
Deux sorties :

- Hauteur de type int ; pour donner la hauteur de l'image en pixel.
- Largeur de type int ; pour donner la largeur de l'image en pixel.



Recherche de plage colorimétrique:

Ce bloc permet de sélectionner sur l'image une zone de couleur et en relâchant le bouton de la souris, le bloc garde en mémoire la plage colorimétrique choisie au format TSV (Teinte, Saturation, Valeur).



Dans ce bloc on a :

Deux entrées :

- Une entrée de type image.
- Plage colorimétrique.

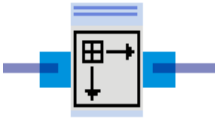
Une seule sortie :

- Image

Filtre:

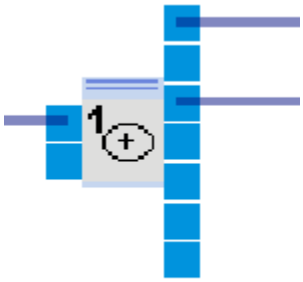
Ce bloc permet de filtrer une image.

Dans notre cas, on va utiliser le filtre minimum ; qui va mettre la valeur colorimétrique du pixel actuel au minimum des valeurs des pixels au sein du masque. La taille de l'image de sortie est réduite en hauteur et en largeur de la taille de masque-1.



Traqueur de segment:

Ce bloc de fonction trouve dans une image en noir et blanc, telle que générée par la recherche de plage colorimétrique, des segments d'un seul tenant et traque un segment défini. Pour traquer tous les segments d'une image en noir et blanc, il suffit de connecter en série le nombre de traqueurs de segment voulu.



Dans ce bloc on a :

Deux entrées :

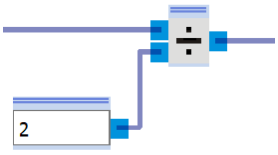
- Entrée est de type image.
- Redémarrage de type booléen.

Sept sorties :

- X de type int.
- Y de type int.
- Surface de type int.
- Vx de type int.
- Vy de type int.
- Nombre de segments trouvés de type int.
- Sortie de type image.

Division :

Ce bloc permet de mettre l'image toujours centré.



Il prend comme entrée la largeur de l'image et la divise par « 2 ».

3.2.1.2. Bloc 2 :

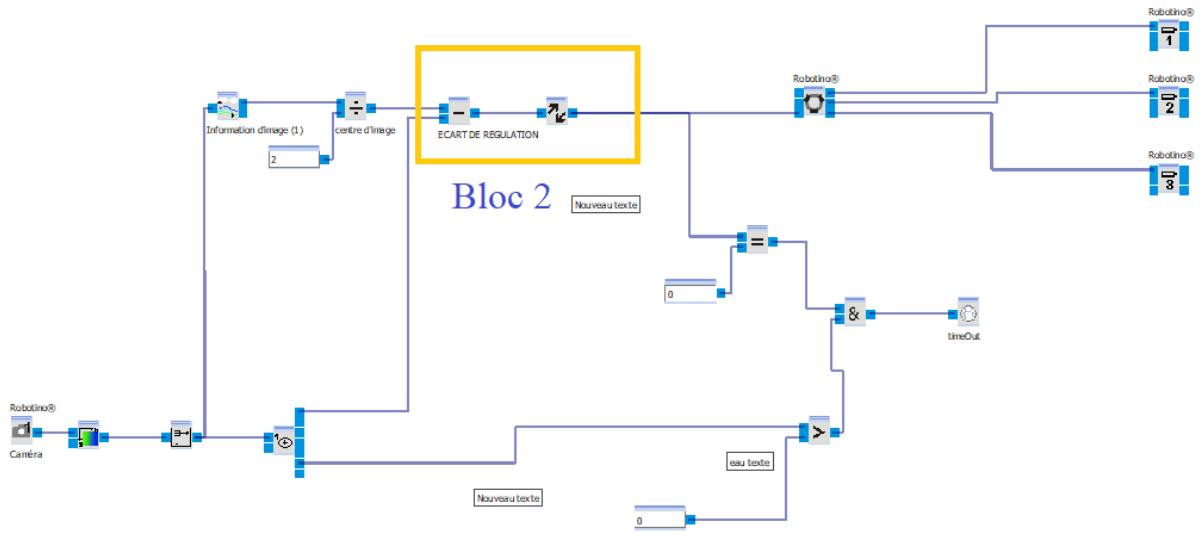


Figure 21 : Bloc 2

Ecart de régulation :

- La vitesse de rotation est calculée par la moitié de la largeur de l'image moins les coordonnées x du cylindre.

La fonction de transfert :

- Ce bloc de fonction permet de transformer le signal d'entrée x (pixel) en un signal de sortie y (oméga) quelconque. Dans notre cas, la transformation pixel-vitesse se fait grâce à cette fonction. En premier lieu, la fonction de transfert prend comme entrer la sortie de l'écart de régulation. Tant que l'image est n'est pas encore centré, la sortie de la fonction de transfert donne la permission à Vx de tourner jusqu'à ce que l'image soit centré.

3.2.1.3. Bloc 3 :

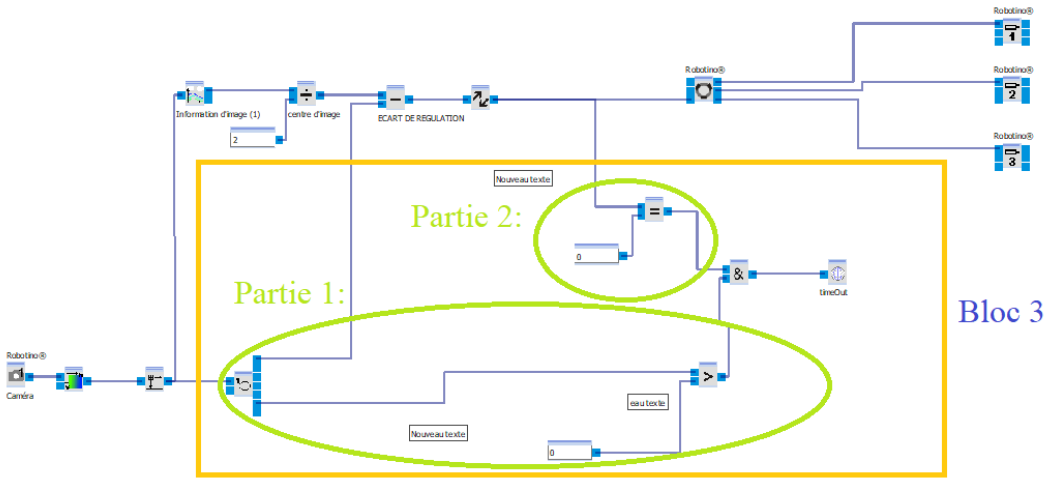


Figure 22 : Bloc 3

Pour que timeOut soit vraie, on va définir deux fonctions liée à une ET logique :

-Partie 1

- Partie 2

-Partie 1 :

Tant qu'il y a une image, la sortie du traqueur qui est le nombre de segments trouvés est supérieure à 0. Dans ce cas, la première condition prend comme sortie « True ».

- Partie 2 :

- Dans cette partie on va comparer, si la vitesse est nulle ou non.

Si la vitesse est égale à 0, c'est-à-dire que l'image est centrée. La sortie est vrai.

Conclusion : Tant que la sortie de partie 1 est vraie, et la sortie de partie 2 est vraie. Alors timeOut est fonctionnelle.

Conclusion

Nous pensons avoir répondu aux objectifs du projet, c'est-à-dire l'utilisation du Robotino pour aller chercher un objet de couleur et approche de ce dernier à l'aide de webcam.

Ce projet nous a permis d'utiliser un Robotino et de découvrir un nouveau logiciel de programmation avec RobotinoView3, dont le développement se fait avec des blocs qui représente des fonctions plus ou moins évoluées (allant de la simple divisons à une fonction de transfert ou une fonction de traitement d'images). Et dont le programme principal s'écrit comme un Grafcet.