

Développement d'un robot mobile Wifi commandé par manette de Wii

Résumé :

Ce projet consiste en le développement d'un robot mobile commandé à distance en Wifi et par manette de Wii. Nous abordons ici toute notre démarche de recherche, présentons notre prototype final mais aussi tous les choix et problèmes auxquels nous avons été confrontés.

Abstract :

This project consists in the development of a mobile robot. It's remotely driven by a wiimote device trough a wireless connection. This document shows our researches on the robot, the final prototype as well as our choices and problems we could have encountered.

Mots clés	Key Words
Carte Fox	Fox Board
Système embarqué	Embedded system
Flux vidéo	Video streaming
PWM	PWM
WIFI	WLAN
RS2R2	RS232
Wiimote	Wiimote

Document rédigé par :

Jeremy MAHU
Yoann SCULO



Sommaire

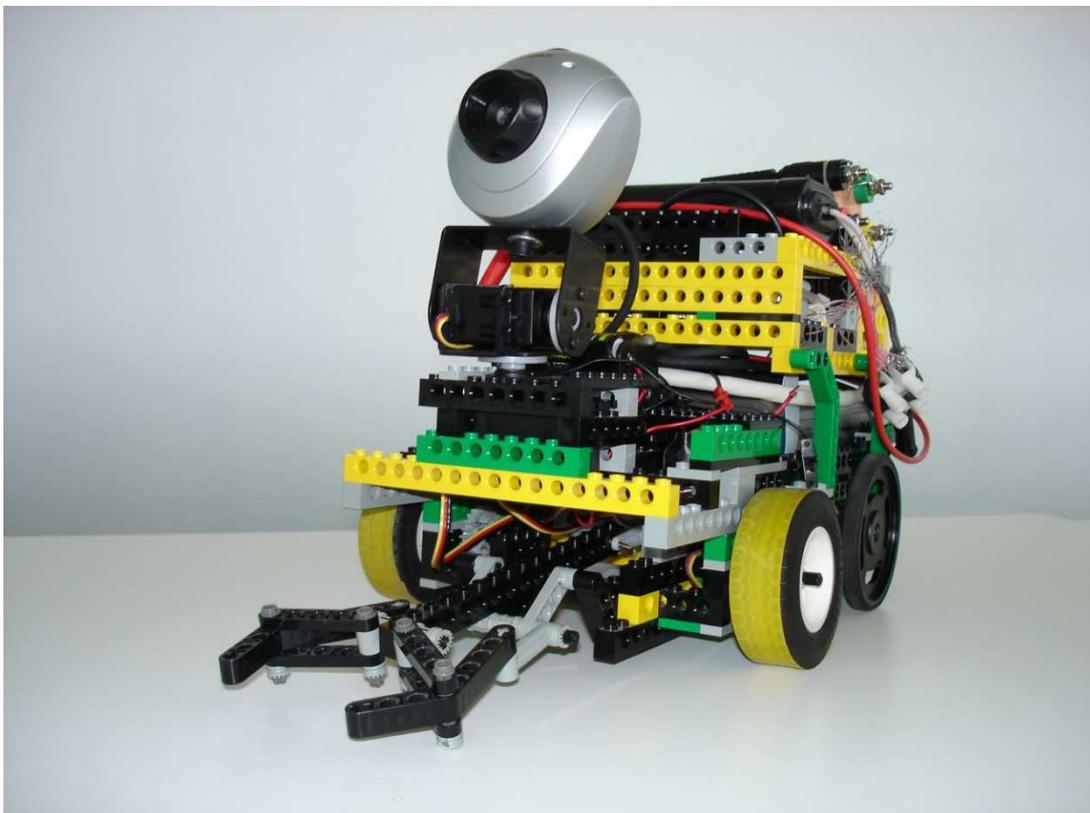
Introduction.....	3
1. Présentation du projet	4
a. Analyse fonctionnelle	4
b. Le matériel.....	5
2. Choix de développement et cheminement.....	8
a. Carte mère / centre de contrôle du robot	8
b. Choix de la motorisation	9
c. Génération d'un signal PWM	10
d. Programmation sur PIC	11
e. Utilisation de servomoteurs	12
f. Commande à distance	13
g. La vidéo.....	14
h. Camera 360	15
i. Pince	15
j. Buzzer	15
k. Alimentation.....	16
3. Utilisation du robot	17
4. Améliorations possibles.....	19
5. Perspectives.....	21
Conclusion	22
Bibliographie.....	23
Annexes	24

Introduction

Ce projet consiste à développer un robot mobile commandé à distance par une manette de Wii avec retransmission vidéo de la vision de ce dernier par connexion wifi. L'idée est de pouvoir concevoir un système embarqué de A à Z en prenant en compte les contraintes techniques et matérielles pour arriver au développement d'un produit fini et fonctionnel.

Nous commencerons tout d'abord par présenter le projet, les fonctionnalités du robot ainsi que le matériel final choisi. Nous expliquerons ensuite tous nos choix de développement ainsi que le cheminement du projet. Nous aborderons notamment les problèmes que nous aurons rencontrés et les solutions que nous avons tenté d'apporter. Nous enchaînerons alors sur la mise en place et l'installation de ce robot, qui nécessite un certain nombre de procédure de lancement.

Nous finirons par un aperçu des améliorations possibles ainsi que les perspectives d'évolution et d'utilisation d'un tel robot. En effet nous pourrions voir que le secteur de la robotique est en plein essor et que ce projet s'inscrit dans une introduction aux possibilités offertes par ce domaine.



1. Présentation du projet

La robotique et les systèmes embarqués sont des domaines en pleine évolution. Les technologies mises à notre disposition sont de plus en plus faciles d'accès. Suite à nos cours de systèmes embarqués et sans fils, nous avons eu l'idée de réaliser un robot dans l'optique de mettre en œuvre nos acquis sur un projet de conception de système embarqué de A à Z.

a. Cahier des charges et analyse fonctionnelle

Nous avons fixé notre propre cahier des charges en proposant ce sujet. L'idée était donc de créer notre propre robot mobile idéalement commandé à distance. Nous avons donc réfléchi aux fonctionnalités que nous pouvions mettre en place de façon raisonnable en un semestre. Nous avons donc pu détacher différentes fonctions du lot que nous avons estimées réalisable durant le temps disponible.

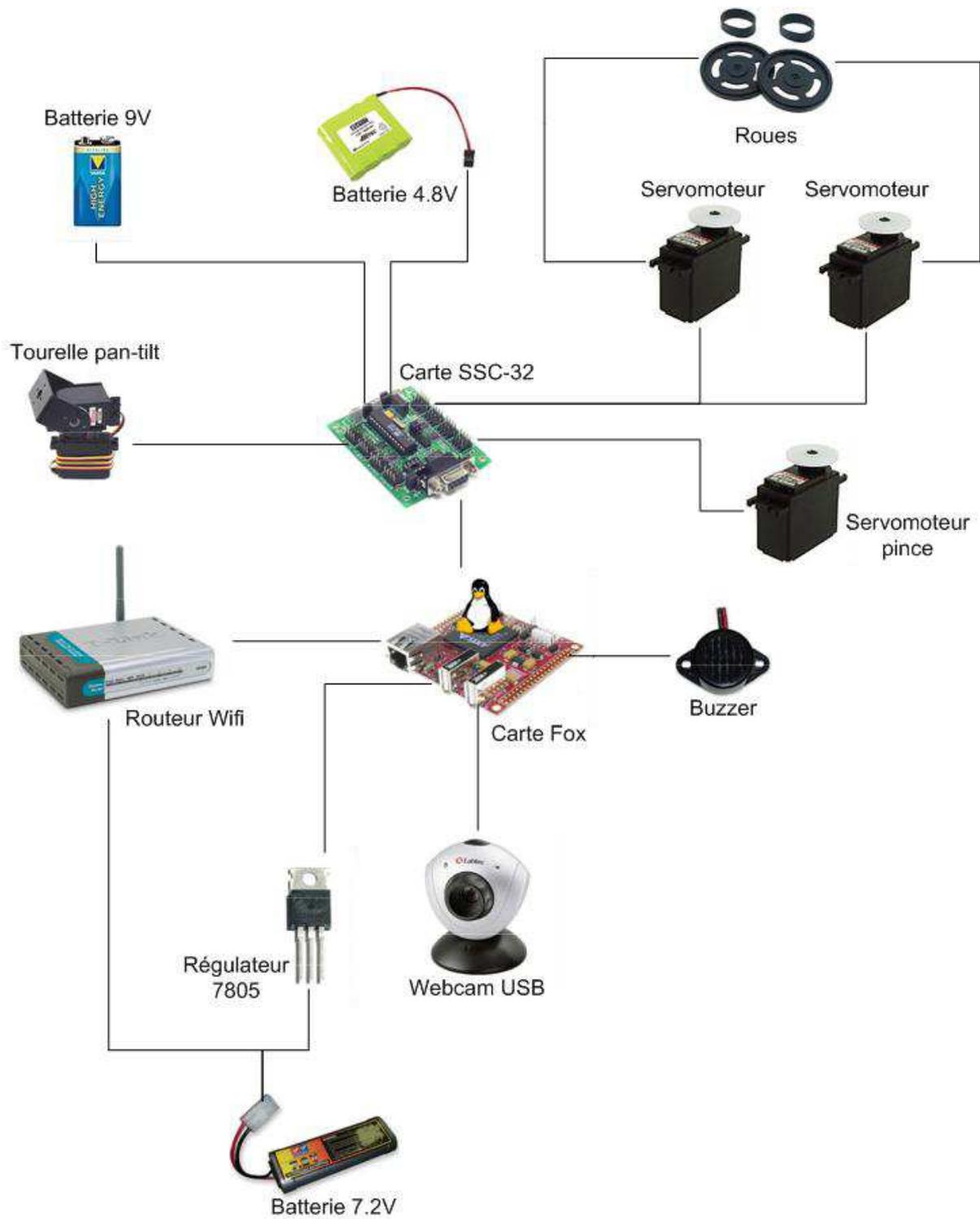
La fonction première du robot était donc de pouvoir se déplacer de manière autonome grâce à un système embarqué sur batteries. Initialement la commande devait être effectuée grâce à une manette de Wii, cependant nous avons étendu cette fonction au clavier et au joystick, ce qui permet d'avoir différentes approches pour la direction du robot. Etant autonome et mobile, l'idée également était de mettre en place un système de vision à distance afin de pouvoir contrôler le robot sans pour autant le suivre de près. Les caméras embarquées ne bénéficiant pas toujours d'un très grand angle de vision, nous avons proposé de mettre en place une tourelle de direction afin de couvrir presque 360° autour du robot. Ensuite, afin d'interagir simplement avec l'environnement, nous avons émis la possibilité de mise en place d'un bras articulé dans le but de saisir de petits objets. Enfin, dans le cas où nous aurions terminé nous tôt, nous avons imaginé la possibilité de mettre en place des voyants, phares et klaxon.

On peut alors distinguer différents blocs fonctionnels sur le robot :

- Commande et réception sans fil
- Direction et Déplacement
- Vision
- Saisie d'objets
- Alimentation
- Gadgets ...

Nous allons donc voir par la suite, le processus de développement de toutes ces fonctionnalités.

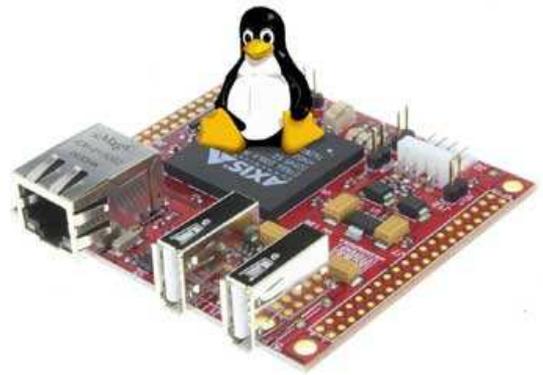
b. Le matériel



Architecture du robot

Carte Fox

La carte FOX est un système embarqué à bas coût utilisant le système d'exploitation Linux. Elle est dotée d'un processeur AXIS à 100MHz et de 32 MB de RAM, de connectiques USB, Ethernet, d'un port série console ainsi que de nombreuses pines pouvant être reliées à plusieurs BUS de données. Elle permet de développer en bon nombre de langages : C, C++, PHP, PYTHON, Shell, ... et permet des connexions HTTP, FTP, SSH et TELNET. De petite taille et faible consommation électrique (1 W) est idéale pour mettre en place rapidement des projets de type embarqué.



Prix : 170 €

Routeur Wifi

Nous avons utilisé un simple routeur D-Link DI-524. N'importe quel routeur peut faire affaire. Celui-ci est alimenté en 7,5V et consomme du 1,5 A. Nous verrons par la suite la justification de son utilisation.



Prix : 30 €



Servomoteurs et roues

Deux servomoteurs à rotation continue et deux roues.



Prix : 18 € x 2 + 9 €

Tourelle pan/tilt

Tourelle 2 axes composée de 2 servomoteurs, pour la caméra.

Prix : 45 €



Caméra : Labtec Webcam Pro

Connexion USB

Prix: 22 €



Carte SSC-32

Carte de contrôle des servomoteurs.

40 sorties PWM

1 port série

Prix : 45 €



Wii mote

Manette de jeu Bluetooth de la console Wii.

Prix : 40€

2. Développement du robot

Le développement de ce robot a été très intéressant dans le sens où nous avons dû prendre beaucoup de décisions, faire des choix en fonction de nos besoins, du temps, de l'argent et du matériel disponible. Nous allons alors expliquer le cheminement que nous avons suivi tout le long du projet.

a. Carte mère / centre de contrôle du robot

Nous sommes initialement partis sur l'idée d'un robot se basant sur un ordinateur EeePC mis sur roues et motorisé. Nous avons fait ce choix en raison de la puissance de calcul intéressante par rapport à d'autres cartes embarquées et en raison de la batterie longue durée du portable. Cependant, nous avons rapidement réalisé que cette option ne nous offrait pas d'entrée sorties numériques pour le contrôle des moteurs, de la caméra 360° et d'éventuels capteurs. De plus, l'idée d'utiliser la batterie de l'ordinateur comme source de courant pour la traction du robot n'était pas forcément une bonne idée et risquait, entre autres, d'endommager l'ordinateur.

Bénéficiant d'une expérience sur la carte Fox, et en ayant plusieurs à notre disposition à l'UTT, nous avons finalement opté pour cette carte embarquée, qui nous offrait directement une interaction plus forte avec le matériel et l'électronique du robot. La différence majeure avec l'EeePC réside dans la puissance de calcul. Cependant, les caractéristiques de la carte Fox sont amplement suffisantes pour notre utilisation. En effet, l'idée a été d'utiliser cette carte comme centre névralgique du robot communiquant avec l'extérieur. La commande se faisant à distance sur un autre ordinateur, bénéficiant d'une puissance de calcul supérieure, nous avons choisi d'effectuer le traitement de manière déportée. Par conséquent la carte Fox permet de rediriger les commandes venues de l'extérieur, et n'a donc pas besoin d'une grande puissance de calcul.

b. Choix de la motorisation

Après avoir choisi notre carte embarquée, nous avons réfléchi au type de motorisation du robot. Différentes solutions s'offraient à nous :

- Les moteurs à courant continu
- Les moteurs pas à pas
- Les moteurs Brushless
- Les servomoteurs

Ces deux premières options impliquaient l'ajout (construction ou achat) d'une carte électronique supplémentaire pour le contrôle des moteurs. Le club de robotique de l'UTT possédant des moteurs Brushless et leurs contrôleurs associés, nous sommes d'abord partis sur cette option.

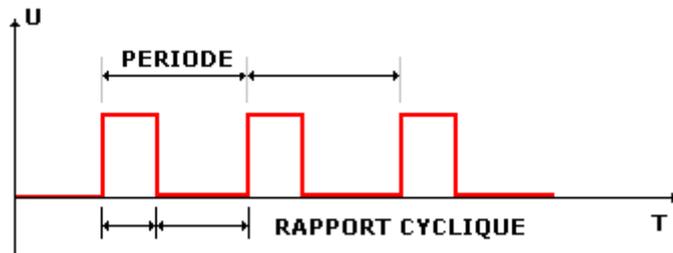


Un moteur Brushless est un moteur électrique dont le rotor est constitué d'un ou de plusieurs aimants permanents et pourvu d'origine d'un capteur de position. Contrairement aux moteurs classiques, sa cage et son axe tournent ensemble autour d'une base qui permet de fixer le moteur. Ces moteurs sont généralement utilisés en modélisme pour les avions étant donné qu'ils peuvent tourner très vite. Ces moteurs sont commandés par un composant appelé « contrôleur ». Il suffit alors d'y brancher les 3 phases du moteur, d'y brancher l'alimentation et d'envoyer par le câble adéquat le signal dit PWM.



c. Génération d'un signal PWM

Les contrôleurs des moteurs Brushless ont donc besoin d'un signal PWM pour fonctionner. PWM pour Pulse Width Modulation. Il s'agit d'un signal carré dont la période est fixe, mais dont le rapport cyclique (rapport entre l'état haut et l'état bas) varie au cours du temps.



Ces signaux sont beaucoup utilisés en modélisme pour contrôler différents moteurs. Les rapports cycliques attendus varient suivant les constructeurs et les moteurs. Mais de manière générale, il existe une plage donnée dans laquelle le moteur sera réactif, par exemple certains moteurs pourront changer de sens en passant le cap des 50% de rapport cyclique. D'autres pourront faire varier leur vitesse progressivement dans un intervalle de 10% autour d'un rapport cyclique de 50%, etc.

Nous avons alors cherché à produire un signal PWM à partir de la carte fox. Ainsi nous avons commencé à travailler sur un programme en C envoyant des 0 et 1 sur une sortie de la Fox à une fréquence donnée. Cependant, après lecture sur oscilloscope nous avons découvert que la carte Fox ne pouvait descendre en dessous de périodes de l'ordre de la seconde. La commande sleep ne nous permettait pas d'obtenir la période de quelques millisecondes nécessaire au fonctionnement du moteur Brushless.

Nous avons alors tenté de contourner le problème en travaillant cette fois ci avec les cycles d'horloge de la carte fox. En travaillant sur une simple boucle while et des levées de drapeau, et associant N cycles d'horloges à une période temporelle, nous avons pu descendre en dessous de la seconde et atteindre la milliseconde. Le signal PWM alors sorti de la Fox a fait fonctionner le moteur Brushless sans problème. Cependant, nous avons découvert que ce programme ne fonctionnait plus du tout lorsque nous exécutions un autre programme en parallèle sur la Fox. En effet, en toute logique, l'ordonnanceur répartit les cycles d'horloge sur les différents programmes de la carte. Par conséquent, notre relation cycle/temps n'était alors plus valable et le PWM produit ne ressemblait plus à rien. Pour résoudre ce problème, la solution aurait été de modifier le noyau de la carte Fox pour le rendre temps réel, et donner la priorité à notre programme PWM sur les autres. Cependant

une telle entreprise nous aurait pris bien trop de temps. C'est pourquoi, nous avons recherché une autre manière de générer un signal PWM propre pour nos moteurs.

d. Programmation sur PIC

Le club de robotique de l'UTT possédant des outils de développement pour PIC, nous avons jeté notre dévolu sur ces microcontrôleurs programmables. Ces microcontrôleurs conçus par la société Microchip sont des unités de traitement de l'information de type microprocesseur auxquelles ont été rajoutés des périphériques internes permettant de réaliser des montages sans nécessiter l'ajout de composants annexes. Ce microcontrôleur a l'avantage de fonctionner de manière autonome après avoir été programmé.



Nous avons dans notre cas entrepris de travailler avec un PIC18F4580 avec l'environnement de développement MPLAB et le compilateur MCC18, nous permettant de programmer en C directement sur le composant. L'avantage de ce genre de PIC est qu'il bénéficie d'une à plusieurs sorties PWM et d'un port série pour communiquer avec d'autres cartes électroniques.

Nous avons alors entrepris de faire sortir un signal PWM de notre PIC, ce qui a été chose assez simple en raison de ses caractéristiques. En effet, le PIC utilisé possédait déjà en effet une sortie PWM. Nous avons alors pu produire notre signal et faire démarrer notre moteur sans soucis. Cependant, le PIC ne possédant qu'une seule sortie PWM, il ne nous permettait pas de gérer les 2 moteurs en parallèle. Nous avons alors dû faire un choix, soit commander un autre PIC avec 2 sorties PWM, soit utiliser une autre technique de programmation du PIC pour transformer une des pattes du PIC en seconde sortie PWM. Nous avons opté pour cette deuxième solution plus rapide, et nos moteurs ont démarré sans aucun problème. Une fois le contrôle des moteurs en mains, nous avons entrepris de relier le PIC à la carte fox par communication série. Cependant, après de nombreuses tentatives, nous n'avons pas réussi à mettre en place de communication stable et à double sens. Nous avions de nombreux parasites inexplicables qui rendaient nos moteurs hors de contrôle. N'ayant pas de moyen à notre disposition pour débayer simplement notre programme (pas d'écran), nous avons finalement abandonné l'utilisation du PIC pour le contrôle des moteurs. A cela s'est ajouté le fait que les contrôleurs des moteurs Brushless se comportaient de manière assez particulière. En effet, le démarrage des moteurs ne se fait que par une accélération progressive de la vitesse. Cette technique difficile à reproduire simplement avec un signal PWM classique ne faisait que compliquer le travail d'asservissement des moteurs que nous avions à réaliser. A cela s'est ajouté le fait qu'un moteur Brushless et son contrôleur sont extrêmement coûteux. Souhaitant rester dans nos frais,

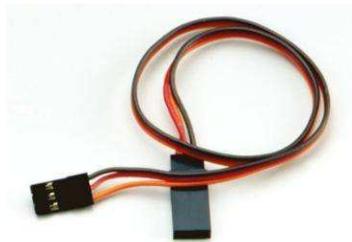
nous avons commencé à rechercher un moyen moins couteux de motoriser le robot. Nous avons donc abandonné l'idée des moteurs Brushless, qui étaient aussi chers que complexes à utiliser.

e. Utilisation de servomoteurs

L'abandon des moteurs Brushless s'explique également par le fait que nous avons réalisé que nous pouvions en fait utiliser de simples servomoteurs sur le robot. Un servomoteur est un système motorisé capable d'atteindre des positions prédéterminées, puis de les maintenir où à l'inverse de tourner de façon continue dans un sens donné. Nous avons donc opté pour ce type de moteur pour la suite.



Les servomoteurs sont commandés par un signal PWM tout comme les contrôleurs de moteurs Brushless. Ils possèdent une connectique composée de 3 câbles :



- **Rouge** : +5V
- **Noir** : Masse
- **Orange** ou **Jaune** : Signal PWM

Nous avons donc opté pour ces servomoteurs, étant finalement plus simples à utiliser que les moteurs Brushless. Toutefois, nous avons toujours à générer un signal PWM... Nous avons alors cherché une autre méthode pour générer un tel signal. Nous avons trouvé une carte beaucoup utilisée en robotique appelée **SSC-32**. Il s'agit d'une carte de contrôle de servomoteurs bénéficiant de 32 sorties PWM, programmable en série par commandes ASCII. Cette carte a grandement simplifié notre robot. Nous n'avions plus qu'à connecter la carte à la Fox et à lui envoyer les commandes ASCII nécessaires au fonctionnement des moteurs.



```
#1P500<cr> // Fait avancer la roue de gauche
#0P2000<cr> // Fait avancer la roue de droite
#1P2000<cr> // Fait reculer la roue de gauche
#0P500<cr> // Fait reculer la roue de droite

#1P0<cr> // Arrête la roue gauche
#0P0<cr> // Arrête la roue droite
```

```
#5PXXXX<cr> // Monte/Descend la caméra verticalement
#4PXXXX<cr> // Tourne la caméra de gauche à droite
```

Avec <cr>, le caractère retour chariot ayant pour valeur 13 en décimal.

A noter que cette carte est alimentée en logique grâce à une simple pile 9V. On vient brancher un accu de 4.8 V pour alimenter les servomoteurs de manière indépendante.

f. Commande à distance

Afin d'ouvrir le maximum de perspective à notre projet, nous avons décidé de la possibilité de piloter le robot soit par le clavier, soit par une manette de jeux, soit par une Wiimote (manette de Wii). Pour cela nous avons décidé de scinder notre application en deux programmes de type client/serveur. Le programme que l'on nommera serveur se situe sur l'ordinateur portable. Il permet de recevoir les trames de la Wiimote, de la manette de jeux ou du clavier.



La manette de Wii est une manette de jeu Bluetooth qui permet d'envoyer des trames d'informations relatives aux boutons pressés et à la position et accélération de la manette. Nous avons fait le choix de la connecter à notre portable et non au robot. En effet il nous semblait plus intéressant de travailler sur la mobilité du robot et de le commander à distance depuis l'écran de l'ordinateur. Par conséquent, les données de la manette sont encapsulées dans la transmission Wifi et la manette dialogue avec l'ordinateur directement. Nous aurions parfaitement pu relier la manette à la carte fox. Cependant cela impliquait un changement du noyau, l'installation des drivers de la clé Bluetooth et la programmation d'une bibliothèque en C embarqué de gestion de la Wiimote. C'était parfaitement faisable, mais afin de gagner du temps, nous avons opté pour la liaison à l'ordinateur. Et cela d'autant plus que nous bénéficions de plusieurs choix de bibliothèques libres pour manipuler la manette. Nous avons ainsi décidé d'utiliser la librairie libwiimote (développée en C) permettant de faciliter l'analyse des trames Bluetooth envoyées par la Wiimote.



Les trames sont donc analysées par le serveur afin de normaliser les données contenues pour pouvoir les transférer plus aisément par Wifi vers la carte FOX. L'organisation de notre programme est relativement simple puisqu'elle s'appuie sur la manipulation de socket. Nous avons différencié deux parties bien distinctes : la gestion des trames provenant de la Wiimote et la gestion des trames

provenant du clavier ou du joystick. Cette distinction est nécessaire car les méthodes/packages utilisés ne sont pas les mêmes. Pour la manipulation des informations provenant du clavier et du joystick nous avons utilisé la librairie SDL. Cette bibliothèque est très utilisée dans le monde de la création d'applications multimédias en deux dimensions comme les jeux vidéo, les démos graphiques, les émulateurs, etc. Elle bénéficie d'outils intéressants pour la gestion d'événements claviers du clavier et du joystick, ce qui nous intéressait dans notre cas.

C'est sur la carte FOX que se situe notre programme client. Il se connecte au programme serveur par socket et réceptionne les données normalisées. Il ne recevra donc que les ordres qui suivent : AVANT, ARRIERE, GAUCHE, DROITE, MODE1, MODE2, MODE3, KLAXON. Nous allons rapidement expliquer la structure de notre code qui se trouve en annexe. Afin d'avoir un mode de déplacement ergonomique, nous avons décidé d'utiliser les mêmes touches pour le déplacement de la caméra, du robot ou de la pince. Pour cela, nous avons été obligés de distinguer plusieurs « états » à notre robot. Le MODE1 est l'état dans lequel le robot peut se déplacer. Le MODE2 est l'état dans lequel la caméra peut bouger, les servomoteurs de déplacement s'arrêtent automatiquement. Le MODE3 est l'état permettant le déplacement de la pince. Nous reviendrons par la suite sur ce point afin de le détailler d'avantage en faisant référence au code.

g. La vidéo

L'intérêt du robot commandé à distance était de bénéficier de la vue interne du robot afin de se repérer dans l'environnement sans pour autant être à côté. Nous nous sommes basé sur le travail d'une précédente TX afin de mettre en place un client/serveur vidéo. Ce système avait été testé avec succès en Ethernet, mais ne fonctionnait pas avec une clé Wifi. Cette dernière ne permettait pas d'envoyer plus d'une image par seconde, ce qui est absolument inutilisable pour du temps réel. Nous suspectons la clé utilisée ainsi que l'efficacité de transmission du bus USB de la carte fox. Afin de contourner le problème, nous avons essayé de brancher la carte à un routeur wifi simplement par Ethernet. Et comme nous l'avions supposé, la transmission s'est faite de manière très fluide avec celui-ci. Nous bénéficions alors d'une transmission vidéo sans fil. Toutefois, cet ajout de routeur eu une incidence importante sur la structure du robot. En effet, le routeur est relativement encombrant et nécessite une alimentation supplémentaire (7,5V), qui n'est de surcroit pas la même que celle de la carte fox. Nous aborderons par la suite les moyens mis en place pour alimenter le système.

h. Camera 360

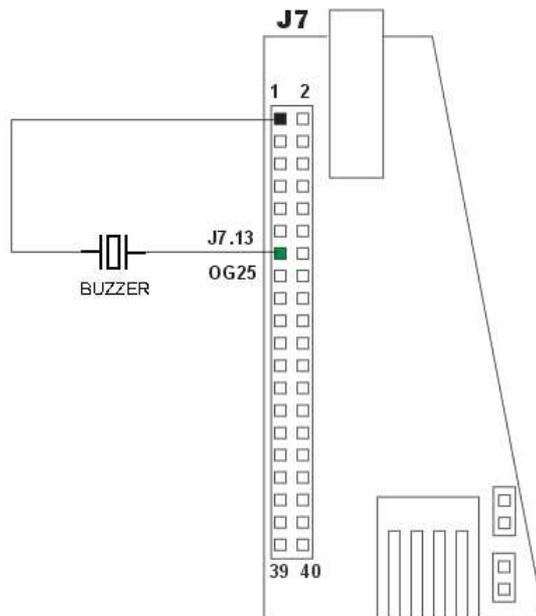
Une fois la transmission vidéo en place, nous n'avons plus qu'à mettre en place la tourelle pan/tilt afin de déplacer le champ de vision à distance. Il s'agit simplement de deux servomoteurs assemblés de telle sorte que la tête de la structure peut balayer presque 360°. L'interfaçage avec la carte de contrôle est identique à celui des servomoteurs de propulsion du robot. A la différence que l'asservissement se fait ici en position et que l'on précise la position souhaitée selon deux composantes (gauche/droite et haut/bas).

i. Pince

Possédant un servomoteur supplémentaire, nous avons monté une petite pince simple en Lego à l'avant du robot qui permet de saisir de petits objets posés sur le sol. Le mécanisme est actionné par un servomoteur qui va tirer ou pousser une tige. Celle-ci ouvre ou ferme la pince suivant l'état.

j. Buzzer

Profitant des sorties de la carte Fox, nous avons branché un simple buzzer électrique sur l'une des sorties de la carte pour faire office de Klaxon.



k. Alimentation

Jusque là, notre robot était alimenté par plusieurs fils reliés à la carte fox, au routeur et à la carte SSC-32. Le robot ne pouvait pas aller bien loin et n'utilisait pas toutes ses possibilités autonomes initialement prévues. Nous avons alors commencé à réfléchir au moyen de mettre ce robot sur batterie afin de l'affranchir des câbles d'alimentation.

Tout d'abord, l'alimentation des servomoteurs étant relativement standardisée notamment en modélisme, il est très facile d'alimenter ces derniers avec de simples accumulateurs que l'on trouve dans les magasins spécialisés. Nous avons dans notre cas utilisé un accumulateur de 4 piles 1,2V produisant du 4,8V à 800mA. Ce dernier alimente à la fois les servomoteurs de propulsion du robot ainsi que ceux de la tourelle pan/tilt et de la pince.



Les caractéristiques particulières d'alimentation de la carte fox (5V) et du routeur (7,5V) ne nous ont pas permis d'alimenter simplement tout le robot par une seule batterie. De plus, il est plus que conseillé de séparer la partie puissance (moteurs) de la partie logique pour les circuits électroniques pour éviter tout parasitage. Nous avons donc opté pour une seconde alimentation, cette fois dédiée à la carte Fox et au routeur. Cependant, pour la carte FOX, on ne trouve pas de batterie 5V simplement dans le commerce, ce n'est pas une tension « standard ». Les batteries les plus proches sont 4,8 V et 7,5V. La carte FOX ayant besoin de 5V ni plus ni moins, nous avons utilisé un composant appelé régulateur de tension afin de transformer les 7,5V en 5V. Il s'agit d'un régulateur de type 7805, ce composant est dans notre cas d'autant plus intéressant qu'il nécessite une tension minimum de fonctionnement, que lui offre la différence entre les 7,5 et les 5V. La perte d'énergie étant répercutée en chaleur, nous avons placé un dissipateur thermique sur le composant. Il est alors relativement simple d'alimenter à la fois le routeur et la carte FOX depuis une seule batterie 7,5V.

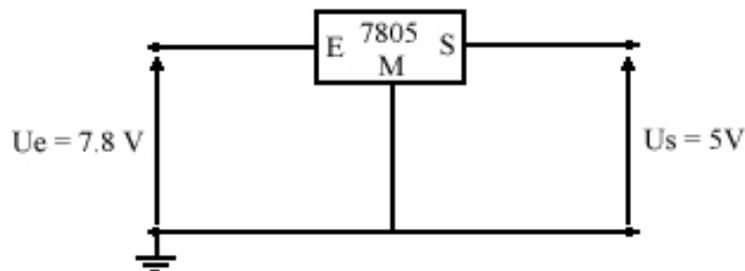


Schéma de branchement du régulateur de tension 7805

3. Utilisation du robot

Afin d'éviter tout débranchement de fil, nous avons opté pour l'insertion d'interrupteur sur toute les sources d'énergie. Il n'empêche qu'il existe malgré tout un procédé d'allumage pour que le robot puisse fonctionner. Tout d'abord il est nécessaire d'alimenter l'ensemble du robot. Vérifier que le routeur et la carte FOX sont correctement alimentés. Il faut ensuite mettre sous tension la carte SSC-32 et les servomoteurs. Une fois toutes les cartes alimentées, il est nécessaire d'établir la connexion entre le PC portable et la carte FOX. Pour cela on utilise la fonctionnalité WIFI du routeur. Une fois la communication effectuée, il ne reste plus qu'à lancer les deux applications (serveur et cliente). Au lancement de l'application serveur, nous avons à choisir entre le mode clavier/joystick (1) et le mode Wiimote (2).

Le serveur se lance alors de la façon suivante :

```
./serveur 1          // Pour le mode clavier
./serveur 2          // Pour le mode wiimote
```

Lors du démarrage du mode Wiimote, il est demandé à l'utilisateur de presser simultanément les touches (1) et (2) de la manette afin de la synchroniser avec l'ordinateur. Le serveur démarre alors et attend la connexion du client. A noter que l'adresse MAC de la wiimote est à entrer manuellement dans le code du serveur et nécessite une recompilation pour sa prise en compte. Par une simple modification du code, nous pourrions donc proposer à l'utilisateur de passer l'adresse MAC en argument. Nous avons préféré l'adresse en dur pour simplifier l'utilisation du serveur, et car nous n'avions qu'une seule Wiimote.

La liaison avec le robot nécessite une connexion ssh sur celui-ci. Au démarrage du routeur, la carte fox est intégrée directement au réseau grâce au DHCP intégré au routeur. Toutefois, dans le cas où le robot est connecté directement au PC, et que le DHCP n'est pas automatique, il faut alors taper la commande suivante sur l'ordinateur déporté (fixe ou portable) :

```
sudo ifconfig eth0 192.168.0.90 netmask 255.255.255.0
```

avec eth0 pour interface réseau de l'ordinateur et 192.168.0.90 l'adresse ip de la carte fox.

La connexion avec le robot se fait simplement par ssh :

```
ssh root@192.168.0.90
```

avec comme mot de passe par défaut : « pass »

Il suffit alors de se rendre dans le répertoire où nous avons envoyé le code du client et de l'exécuter :

```
./client 192.168.0.109
```

Avec 192.168.0.109 comme adresse IP du serveur dans notre cas.

Le serveur démarre alors et commence à envoyer les commandes au robot suivant le mode choisi. A noter que pour utiliser la commande clavier, le focus de la souris doit être placé dans la fenêtre SDL qui s'est ouverte au lancement du serveur. La wiimote et le joystick ne nécessitent pas quant à eux du focus pour fonctionner.

Une fois le robot en état de marche, il faut mettre en œuvre la communication vidéo. Celle-ci s'active en se connectant à nouveau en ssh sur le robot et en exécutant le code suivant :

```
/bin/spcaload                # chargement des drivers de la caméra  
sleep 3  
/bin/servfox -d /dev/video0 -s 640x480 -w 888 # lancement vidéo
```

Ce code peut être automatisé en le plaçant dans un script serveurvideo.sh que l'on disposera dans le répertoire /etc/init.d/boottime. Il s'agit du dossier d'exécution d'applications au démarrage.

Ce dernier sera alors chargé automatiquement au démarrage de la carte fox.

L'affichage de la vidéo se fait en se connectant à la fox via l'adresse <http://192.168.0.90/index.htm> . Le client télécharge alors un applet java qui va aller se connecter automatiquement au serveur vidéo. La vidéo est ainsi simplement affichée dans le navigateur en temps réel.

A noter que pour la mise en place du serveur vidéo nous avons utilisé le travail de Flavien Boutoille et Guillaume Vercoutère, ayant réalisé une TX intitulée « Gestion de la vidéo et du son sur un système embarqué linux »

4. Améliorations possibles

Les possibilités en robotique sont infinies, et ne dépendent finalement que du budget que l'on peut mettre dans notre robot. Toutefois, nous pourrions entrevoir différentes améliorations et ajouts à notre robot.

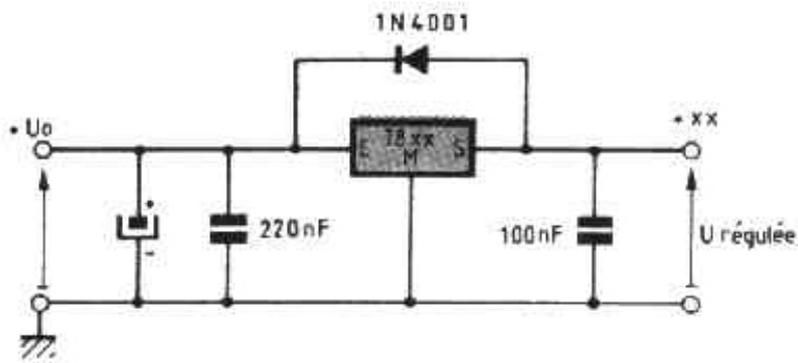
- Tout d'abord, maintenant que la génération de signal PWM est résolue, nous pourrions parfaitement revenir sur les moteurs Brushless afin de permettre au robot d'aller beaucoup plus vite et de bénéficier d'une plus grande puissance.
- Ensuite, nous avons utilisé un routeur uniquement pour transmettre la vidéo en temps réel. Il s'avère qu'il existe des routeurs particuliers aujourd'hui dont le firmware est modifiable. Ces derniers peuvent embarquer un système linux sans problème et par conséquent supplantent la carte Fox. En d'autres termes, nous pourrions retirer la carte fox et ne travailler qu'avec un routeur modifié. Cela permettrait à la fois un gain en place et en consommation énergie. Ou inversement trouver une clé wifi qui permette d'envoyer de la vidéo à un débit correct et supprimer le routeur et sa batterie.
- Jusqu'à présent, nous avons commandé notre robot à distance, nous pourrions parfaitement le rendre autonome. Cela nécessiterait l'installation de différents capteurs pour que le robot puisse se repérer dans l'environnement. Nous pensons par exemple à :
 - des capteurs de distance de type infra rouge (capteurs Sharp)
 - des capteurs de distance ultra sons (création d'un radar)
 - des odomètres permettant de déterminer la distance parcourue par le robot.
 - un module boussole électronique pour repérer dans quel sens le robot se situe.
 - un module GPS pour situer le robot à l'échelle de la planète
 - des gadgets tels que des phares, clignotants, haut-parleurs micros, capteurs de température, humidité, etc.

Cela passerait certainement par le développement de cartes à PIC dédiées à l'analyse et à la transmission des données des capteurs à la carte fox via port série ou bus I2C voir CAN.

- Ce genre d'application nécessiterait le développement d'une intelligence artificielle afin que le robot s'adapte à son environnement. Il pourrait alors être intéressant de développer une application d'apprentissage permettant au robot d'interagir au mieux avec son entourage.
- Nous pouvons également songer à utiliser la caméra pour travailler sur du traitement d'image, notamment pour repérer des formes et des objets à partir de la vidéo. Ce genre d'application,

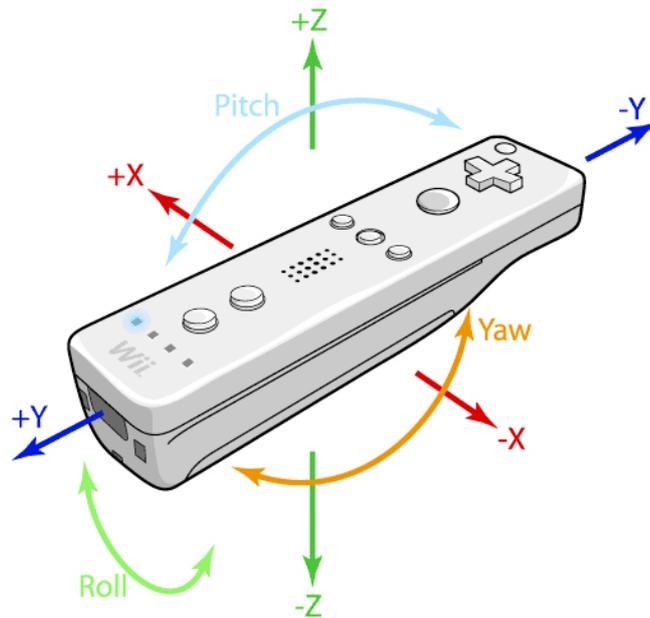
associée aux capteurs, permettrait certainement de récupérer bon nombre d'informations sur l'environnement du robot.

- L'utilisation que nous avons faite du régulateur de tension 7805 est correcte, mais pourrait être nettement amélioré. Nous pourrions protéger notamment nos composants en lissant le courant et en empêchant tout renversement de la polarité. Le schéma suivant pourrait être mis en œuvre :



Régulateur de tension 7805 amélioré

- Nous utilisons actuellement un applet java pour visionner la vidéo du robot. Nous pourrions entreprendre de migrer la vidéo vers une application graphique (SDL, GTK, QT, ...) qui permettrait de centraliser toutes les commandes du robot.
- Il pourrait être également intéressant de pouvoir faire cohabiter la Wiimote, le clavier et le joystick en même temps. Ils fonctionnent actuellement de façon indépendante suivant deux modes, mais nous pourrions parfaitement, avec quelques modifications, intégrer la Wiimote au sein du code SDL du clavier et du joystick.
- Enfin, nous pourrions commander le robot à l'aide de la Wiimote grâce au mode accéléromètre. En effet, actuellement nous utilisons les croix directionnelles dans le but de simplifier l'acquisition de données. Toutefois, la gestion des accéléromètres serait parfaitement faisable. Il suffit de récupérer les informations de pitch et roll qui correspondent à l'inclinaison de la manette (voir figure suivante). L'idée ensuite serait de borner ces valeurs et provoquer l'envoi de commande de déplacement suivant l'intervalle dans lequel on se situe. L'intérêt actuel est amoindri car les moteurs du robot sont des servomoteurs continus dont la vitesse ne peut pas varier (ou très peu). Par conséquent, l'expérience serait sans doute plus concluante sur des moteurs de type Brushless étant donné qu'ils offrent une grande plage de manœuvre en terme de précision et donc d'accélération.



Axes de la Wiimote

5. Perspectives

Les robots et drones sont dans l'ère du temps et de plus en plus utilisés et vendus de nos jours. On peut retrouver des robots mobiles dans certains magasins de jouets. Il est alors possible de commander à distance ces robots wifi et de bénéficier de la vision embarquée. Certains peuvent même aller se recharger d'eux même à une borne. De tels robots peuvent être aussi utilisés en domotique pour de la télésurveillance, ou bien dans notre jardin pour tondre la pelouse de façon automatique. On les retrouve également au sein de la sécurité civile tout comme dans les rangs des armées partout dans le monde. Qu'ils soient roulants ou volants, leurs utilisations et applications sont multiples. Les technologies diffèrent, mais on retrouve également ce genre de robot au fond des océans, dans le ciel ou bien encore plus loin comme sur mars. L'étendue des possibilités est infinie.



Notre robot n'est qu'un aperçu infime de ce qui peut se faire dans le domaine. Les possibilités sont infinies. Nous sommes en plein dans la période de développement de la robotique, au même titre que l'informatique l'a été dans les années 80.

Notre robot n'est qu'un aperçu infime de ce qui peut se faire dans le domaine. Les possibilités sont infinies. Nous sommes en plein dans la période de développement de la robotique, au même titre que l'informatique l'a été dans les années 80.

Conclusion

Ce projet a été à la fois très intéressant et très formateur. Nous avons pu concevoir et développer un robot mobile de A à Z, mettant en application nos connaissances acquises en cours. Nous avons choisi notre matériel en fonction de nos besoins puis essayé différentes solutions afin d'arriver aux fonctionnalités souhaitées. Cette approche nous a permis de réaliser qu'il faut parfois savoir jongler entre le temps de développement, le coût du matériel, les contraintes techniques et la complexité de mise en place. La réalisation de ce robot nous a également donné l'opportunité de travailler sur plusieurs domaines à la fois notamment la programmation, l'électronique, la gestion de l'énergie et la mécanique. Nous avons donc au final pu mener un vrai travail de conception de système, ce qui s'apparente le plus à un travail d'ingénieur.

De manière générale ce projet nous a ouvert au monde de la robotique et à toutes ses perspectives, notamment professionnelles.

Nous tenons enfin à remercier le club de robotique de l'UTT pour leur prêt de matériel ainsi que leurs conseils avisés.

Bibliographie

Constructeurs

<http://foxl.acmesystems.it>

<http://www.lynxmotion.com>

Magasins

<http://www.gotronic.fr>

<http://www.lextronic.fr>

Sites Internet et documentation

<http://www.abcelectronique.com/bigonoff/>

http://kudelsko.free.fr/articles/appli_regul.htm

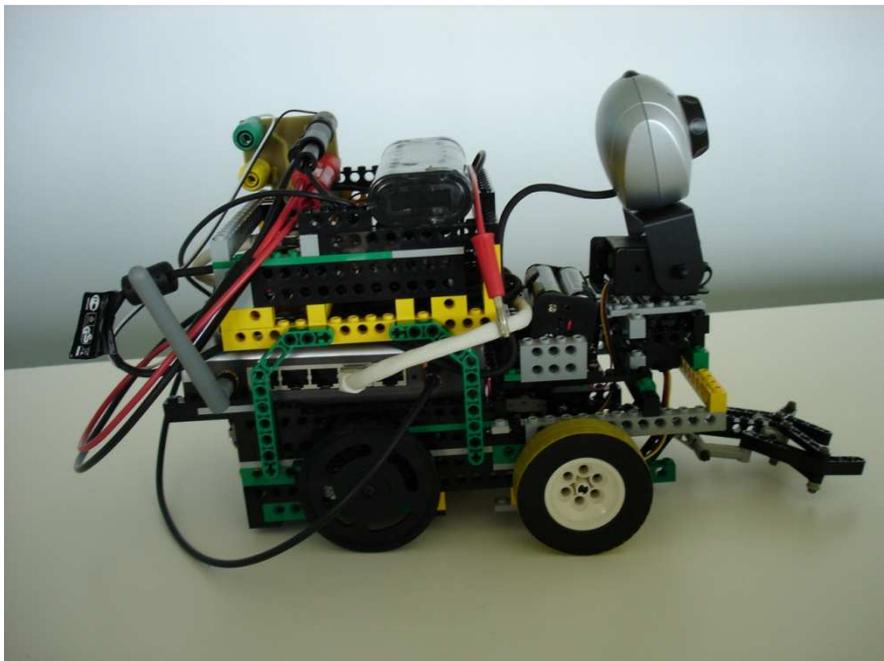
<http://kudelsko.free.fr/articles/alimentation2.htm>

Documents

Flavien Bouteille, Guillaume Vercoutère. *Gestion de la vidéo et du son sur un système embarqué linux*. UTT, Troyes

Annexes

Photos du robot :



Vidéos :

Premier prototype du robot (24 Avril 2009)

<http://www.youtube.com/watch?v=eSErBSeCOCU>

Version finale du robot – test sur batterie commandé par le clavier (3 Juin 2009)

<http://www.youtube.com/watch?v=rzsYxudxMRQ>

Version finale du robot – test sur batterie commandé par une Wiimote (3 Juin 2009)

<http://www.youtube.com/watch?v=CM0XtpQLot8>