□h□pitre I□

□□□□□□ □□□□ □N□□ □□ □I□I□□□

□□ □U□I□□ □□ □□□□□□ □□□□ □N□

L'environnement de travail utilisé est Cross-GCC, compilé pour la cible powerpc-eabi. Seulement Cross GCC ne dispose pas d'outils de debuggage, la première étape du travail a été de créer un débuggeur avec un terminal ASCII. Pour cela, le premier programme a compiler sur le nœud, était celui de la liaison série. Une fois ce programme compilé, il a suffi de connecter à la carte ROBOSOFT un terminal ASCII VT100 par l'intermédiaire d'un câble série. Il était alors possible de faire communiquer la carte MPC avec le terminal.

Pour tester l'état d'un registre, il fallait alors écrire une fonction en assembleur Power-PC qui allait lire le contenu du registre. De même il était possible de voir les zones où le programme défaillait. Pour cela, il fallait vérifier l'affichage d'un message prédéfini sur l'écran du terminal. Ainsi on pouvait vérifier que le programme ne buggait pas à l'endroit où était située l'instruction d'envoi du message sur la liaison série.

De même, l'utilisation d'une LED, connectée à une sortie générale, permettait de tester le passage dans une boucle. Cette méthode est plus fiable que la précédente, car on ne sait pas exactement à quel moment le buffer de l'UART affiche le message à l'écran.

□□ □ I□□ □N □ U□R□ □□□ □N□R□□□ □□□R□I□□ □ □N□RA□□□

La carte à wrapper nécessitant l'utilisation d'une sortie générale pour commander l'aiguillage entre les capteurs avant et arrière, il était donc nécessaire d'apprendre à configurer les registres pour générer un signal de sortie.

L'électronique du nœud ROBOSOFT impose l'utilisation des broches MPIO[12 :15] en entrées et des broches MPIO[4 :11] en sorties. Les registres MPIOSMDDR et MPIOSMDR permettent de commander les entrées / sorties générales MPIO.

Pour simplifier la commande des sorties, on peut utiliser la fonction C écrite par ROBOSOFT ***int WritePWMIO(int value)***. La valeur passée en argument correspond à la broche que l'on souhaite voir changer d'état. Pour la carte à wrapper, la sortie commandant l'aiguillage, est la sortie MPIO10, ce qui correspond à 0x40.

□□ □ I□□ □N □ U□R□ □U □□ □

Un niveau haut sur la broche INIT provoque l'initialisation d'un compteur et le début d'une mesure, par contre un niveau bas sur cette même broche entraîne la remise à zéro du compteur et l'arrêt des capteurs. La commande des capteurs va donc être faite par un signal PWM qui sera branché sur la broche INIT.

Pour définir la forme du signal PWM, il faut définir la période (registre MPWMSPERRX) et le rapport cyclique (registre MPWMSPULRX). De plus, le lancement du signal PWM, se fait à l'aide du bit 5 du registre MPWMSMSCRX.

Là encore, on peut utiliser la bibliothèque C libnode555, où est définie la fonction ***PWMInit()*** qui détermine la fréquence. Il faut donc modifier cette bibliothèque (pwmasm.c) pour avoir la fréquence souhaitée de 16 Hz. Ainsi, la fréquence de travail de 40 MHz doit être divisée par 2, grâce au registre PWMSCR et divisée par 4, par le registre MPWMSM□ (X représentant le PWM souhaité). Les derniers digits du registre MPWMSM1_SC permettent une division par 10 de la fréquence (0x54F6). Enfin, le registre MPWMSM1_Pe contrôle la période, la valeur de ce registre indique le facteur diviseur final de la fréquence de travail. Soit un facteur de 32767 (0x7FFF) pour avoir une fréquence de signal PWM de 16 HZ.

La fonction ***PWMWrite(int choix_pwm, short int *valeur)*** permet de définir le rapport cyclique et donc la largeur de l'impulsion. *choix_pwm* correspond au générateur PWM que l'on souhaite activer (PWM[0 :3]) et **valeur* correspond à la largeur de l'impulsion que l'on souhaite avoir (valeur comprise entre 0 et la valeur du registre MPWMSM_Pe).

Lorsque le signal PWM est à l'état haut un compteur doit permettre de connaître le temps de vol de l'onde, par contre, lorsqu'il est à l'état bas, ce compteur doit être remis à zéro. Il a fallu écrire une fonction, pour connaître l'état du PWM (haut ou bas). Cette fonction ***PWMRead(int choix_pwm, int &etat)***, fonctionne de manière identique à PWMWrite. Ainsi *choix_pwm* sélectionne le PWM que l'on souhaite observer et *&etat* donne l'état de ce PWM (si etat Σ 1, alors le PWM est à l'état haut).

□□ □ I□□ □N □ U□R□ □U □□U

La fonction PWM doit être associée à une unité de comptage précise. Le comptage doit débuter lorsque le PWM démarre et doit s'arrêter lorsqu'un signal ECHO est détecté. La valeur du compteur permet alors de déduire la distance. Le compteur ayant la plus grande précision de mesure est le timer TPU. Ce timer est entièrement programmable et comporte des fonctionnalités très intéressantes préprogrammées en ROM.

Pour la mesure précise du temps de vol, on désire compter le temps écoulé entre l'envoi du signal INIT (généré par le PWM) et la réception d'un signal ECHO. Pour cela, on utilise la fonction *ITC*[10], qui permet de capturer la valeur d'un compteur TPU en levant une interruption. Le compteur TPU, ayant généré cette interruption, est arrêté et la routine d'interruption est traitée dans le *handler* d'interruption. Il faut étoffer cette fonction en lui ajoutant des actions d'arrêt, d'initialisation, ainsi que de reset et de lecture de l'état de chaque compteur TPU. Ces fonctionnalités ont été écrites dans une bibliothèque (ultraasm.s).

- ***TPUStop()*** : arrête uniquement le compteur TPU,
- ***TPU3Inittcr1()*** : gère le timer en définissant l'emplacement en RAM des 7 compteurs,
- ***TPUInitUltra()*** : initialise la fonction ITC (remise à zéro des interruptions),
- ***TPUReset()*** : remise à zéro du compteur TPU,
- ***UltraRead(int &distance)*** : lit la valeur du compteur TPU qui a généré l'interruption.

[10] ITC : Input Transition Counter.

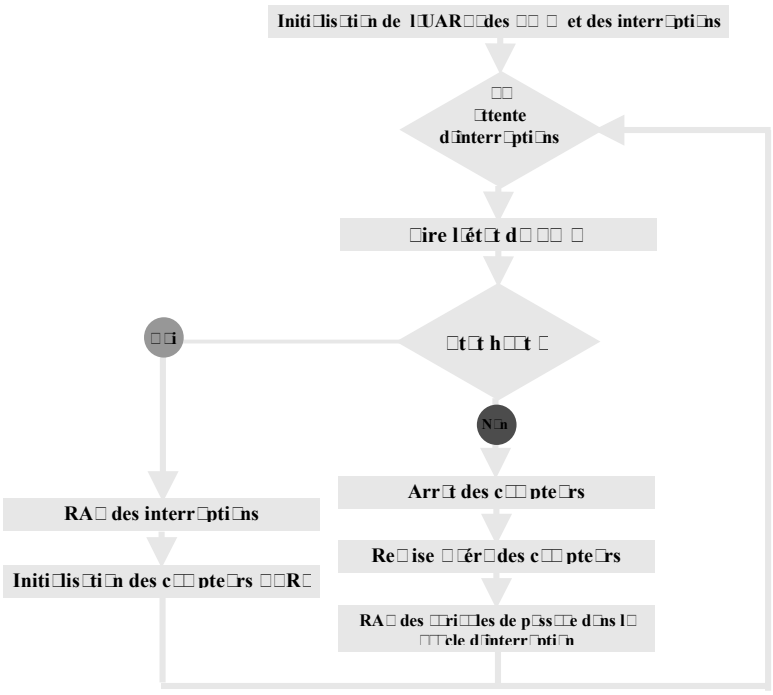
Dans le but de diminuer au maximum le temps de traitement, le compteur TPU a été configuré pour avoir une lecture directe de la distance en millimètre.

Idéalement, chaque capteur devrait être associé à un signal PWM. Cela permettrait d'arrêter la mesure lorsque le temps nécessaire à la mesure de la distance maximale est écoulé. Ceci nécessiterait 14 canaux TPU associés à 14 PWM. Or, tant que le signal INIT envoyé sur un capteur est à l'état bas, un reset du compteur TPU est nécessaire pour démarrer une nouvelle mesure avec un compteur à zéro, lorsque INIT repasse à l'état haut. Seulement la TPU ne possède qu'un seul compteur pour les 16 canaux. Ainsi, lorsqu'un reset est fait, c'est l'ensemble des 16 compteurs TPU qui sont resetés. Par conséquent, le signal INIT doit être commun aux 7 capteurs et donc à 7 TPU. Ce signal INIT possède les caractéristiques nécessaires au fonctionnement du capteur ayant la plus grande distance à mesurer. Pour adapter ce signal aux capteurs ayant des mesures de distances plus courtes, il faudra masquer les échos des capteurs dont le temps, correspondant à la distance maximale couverte, est écoulé.

□□ □□□□UR□ □U □U□ □□ □□NN□□□

La fonction ***ReadEtatCap(int &val)***, permet de lire les bits de données D[9 :15]. Elle a été créée à partir de la fonction ***WatchDog()***, qui permet de s'assurer que le programme n'est pas bloqué, en levant régulièrement le chip Select 2, avec une lecture à vide (à une zone mémoire donnée). Donc, la fonction ReadEtatCap(int &val) permet de stocker dans la variable *val* l'état du bus de données situé à l'adresse 0x0FF00000, levant ainsi le CS2. Ce chip select sert à valider le latch durant le temps de la lecture.

□□ □□RU□□UR□ □U □R□□RA□ □ □



□i□□re I□ □□□□: □l□□rith□e d□pr□□r□□ □e princip□

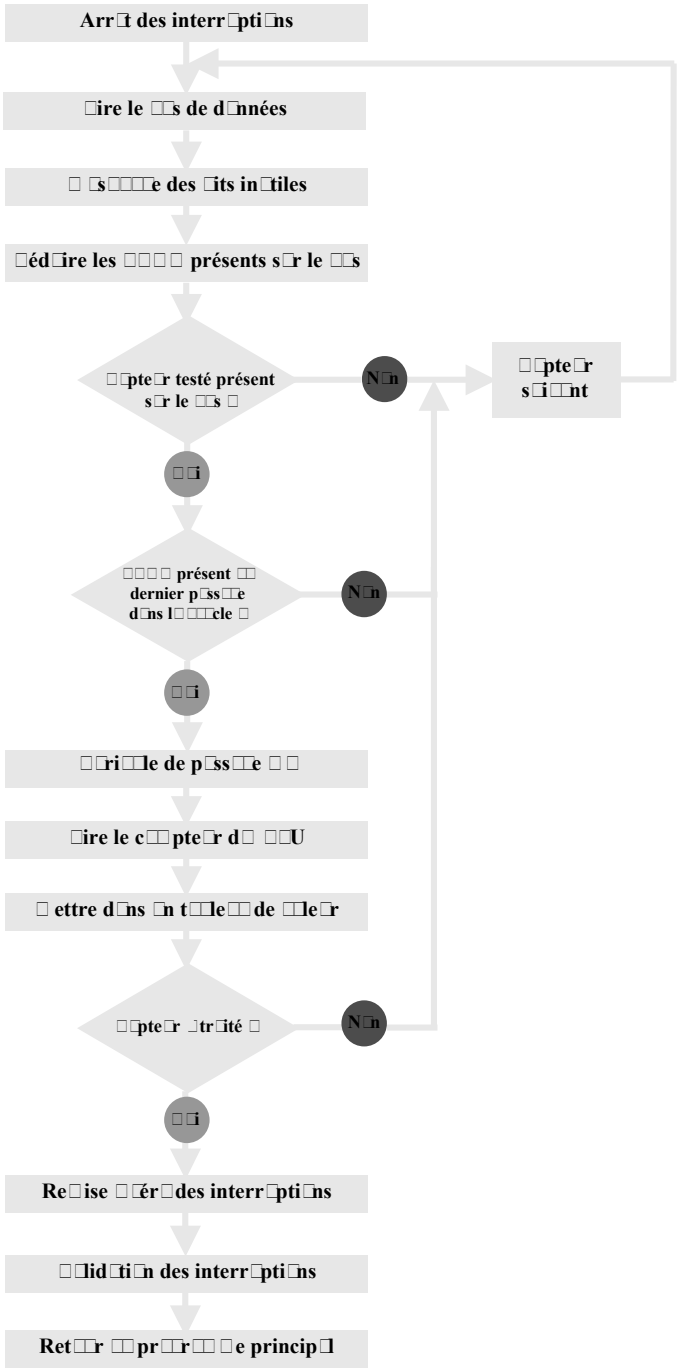


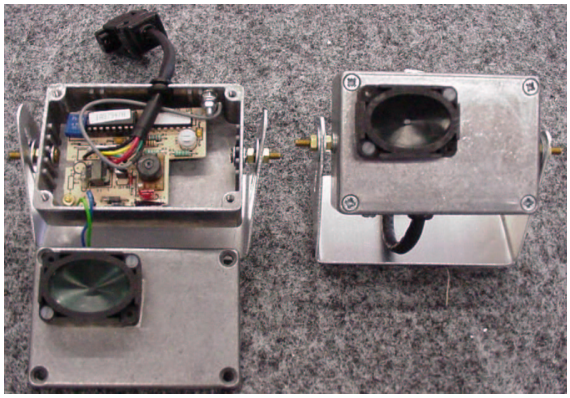
Figure IV-1 : Algorithme de la routine d'interruption

□h□pitre □
□ I□□ □N □□I□I□R □□ □□□□□

□□ □ I□□ □N □□I□I□R

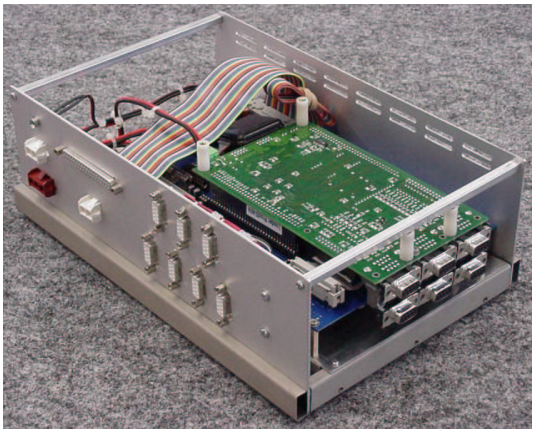
La mise en boîtier des cartes 6500 et des capteurs piézoélectriques a posé quelques problèmes.

En effet, le cristal piézoélectrique rayonnait dans les boîtiers, à cause des 400V nécessaires à sa résonance. Pour stopper ce rayonnement, une mousse isolante de 1 millimètre d'épaisseur a été insérée entre le piézoélectrique et le boîtier métallique.



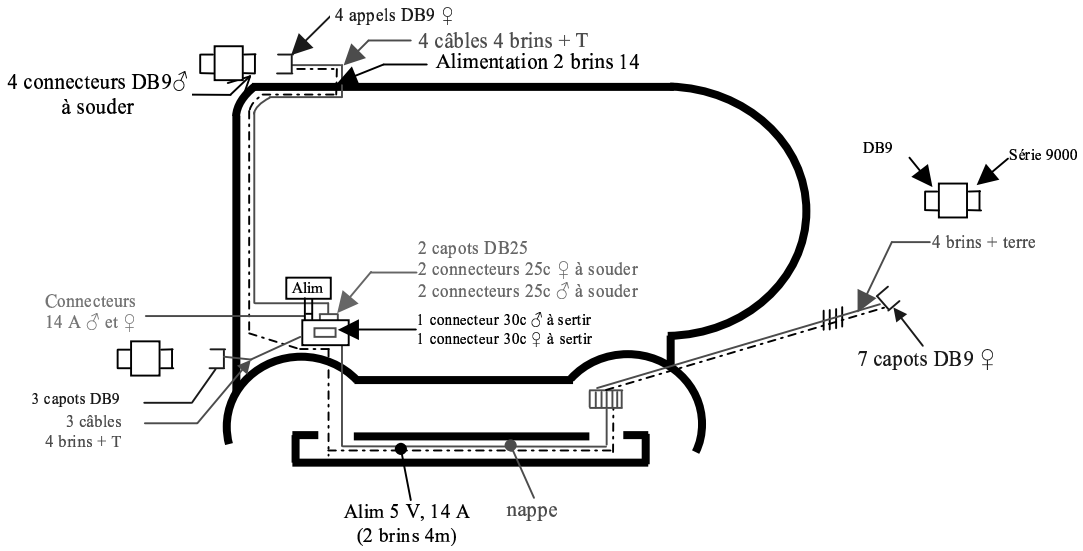
□i□re □.□□□: c□rte □□□□et s□n crist□l série □□□□inté□rés d□ns le□r □□tier

La mise en boîtier du MPC et de la carte à wrapper n'a pas posé de réels problèmes, si ce n'est qu'il fallait faire attention à ce que les connecteurs de la carte à wrapper ne soient pas en contact avec le boîtier métallique. La mise en boîtier a dû être réalisée avec méthode, compte tenu du nombre assez important de fils à intégrer dans le boîtier.



□i□re □.□□□: c□rte d□inter□□□□e et c□rte □ □□□□□inté□rées d□ns le□r □□tier

Pour connecter les capteurs avant et arrière au boîtier de commande, de nombreux câbles et connecteurs doivent être installés sur le CyCab. Le schéma d'implantation et l'orientation de chaque capteur sont donnés en annexe page 36 et 37



□i□re □.□□□: sché□ □ d□i□ pl□nt□ti□n d□ □ □teriel

□□ □□□□□ □□ □A □AR□□ □□ □U □R□□RA□ □ □

L'idéal, pour la fonction ***ReadEtatCap(int &val)*** aurait été de pouvoir lever une autre interruption que CS2, mais je n'ai pas su associer le CS3 au mapping que j'avais configuré. Quand j'allais lire l'état du bus de données aucun chip select n'était levé.

Afin de ne pas endommager la carte MPC et plus particulièrement le microcontrôleur MPC555, un grand soin a été apporté à la réalisation de la carte wrappée en vérifiant minutieusement que les fils wrappés ne causaient aucun court circuit. De plus, les fonctions ont été montées et testées l'une après l'autre sur la carte à wrapper et sur le nœud ROBOSOFT. Toutefois, notons que quelques problèmes subsistent.

Tout d'abord, il a été possible de mettre en évidence un disfonctionnement du TPU4 sur la carte ROBOSOFT, par conséquent le TPU4 a été remplacé par le TPU8.

Ensuite, le TPU0 ne fonctionne pas correctement. En effet, son fonctionnement paraît être aléatoire et le fait que le TPU1 fonctionne parfaitement, semble orienter les recherches sur un fil mal wrappé ou sur les composants SN75172 (carte à wrapper) ou LTC489 (nœud ROBOSOFT) endommagés. Toutefois, notons qu'il suffit d'une petite pression sur une extrémité de la carte à wrapper pour faire fonctionner le TPU0.

Rappelons que pour ne pas avoir à changer les valeurs des résistances d'attaques des INIT, il faut bien connecter la totalité des capteurs sur la carte d'interfaçage.

Enfin, la consommation de la carte wrappée connectée à la carte MPC est de 65 mA. Dans cette même configuration, la carte MPC consomme 90 mA et la batterie 200mA en marche avant et 280 mA en marche arrière (l'écart est dû à l'énergie de maintien du relais).

□□ □□□□ □U □□I□I□R □□ □□□ □ AN□□ □□ □□□ □A□□□UR□

ATTENTION : le branchement de plus d'un capteur sur la carte à wrapper, provoquait des mesures erronées. Après, une longue période de recherche de cette panne, il s'est avéré qu'il s'agissait d'un bruit très faible (0,5 V) et très ponctuel présent sur les signaux BINH et BLNK de la carte 6500.

Je n'ai pas pu déterminer exactement la nature de ce bruit, il semble que ce ne soit ni le kit 6500, ni l'alimentation qui s'écroule, ni un problème d'atténuation de signal (la longueur des câbles est quand même de 2,5 mètres). Toutefois, cette perturbation de 0,5V est la tension maximale que doit avoir le signal bas de la carte 6500. Cette tension est alors suffisante pour forcer le kit à basculer en mode réception. Or, le signal de perturbation apparaît avant la relaxation du cristal, ce qui entraîne la réception d'un signal fantôme.

La solution retenue n'est pas très satisfaisante, car elle annihile les efforts faits pour garder la possibilité d'utiliser le mode multi-écho. En effet, pour remédier à ce problème, les fils fixant les signaux BINH et BNLK sont déconnectés des cartes 6500 à l'intérieur du boîtier et les 2 résistances de 6,8 K∠ sont directement soudées sur la carte.

Ainsi si le besoin se fait sentir d'utiliser le mode multi-écho, il faudra résoudre ce problème de bruit, dessouder les résistances fixant le mode d'utilisation et reconnecter les fils provenant de la carte à wrapper.

Pour effectuer le test du système, les modifications citées précédemment ont été apportées à 7 capteurs.

Pour des raisons de temps, le système de détection n'a pu être testé qu'en statique. Ces tests sont très encourageants, car les capteurs, associés à la carte MPC donnent des mesures très précises. La précision est de 3 à 5% pour les mesures de distances inférieures à 0,8 mètre et de 1% à partir de 1 mètre.

On peut aussi souligner qu'avec l'utilisation de toutes ces fonctions dédiées, la charge CPU est très faible et permet ainsi d'être utilisée pour le traitement d'autres tâches. Une mesure de 6 capteurs regardant un objet situé exactement à la même distance de chacun des capteurs a permis de vérifier que le programme ne perdait pas le temps réel et affichait la même distance pour chacun des capteurs.

Bien que les tests en dynamique n'aient pas pu être réalisés, il semblerait que le système ne soit pas très sensible à la forme des objets.

En effet, un pseudo test dynamique a été effectué en plaçant 6 capteurs sur une table roulante, en essayant de leur donner des orientations proches de celles qu'ils auront sur le CyCab. En faisant avancer la table roulante vers des objets orientés de manière à ce qu'ils soient difficilement détectables, force est de constater que le système détecte assez facilement les objets, alors qu'un seul capteur ne le permettait pas. Par conséquent, Il faudra peut être revoir le gain et l'orientation des capteurs, car il sera peut-être possible d'augmenter la zone de détection.

□h□pitre □I
□□N□□U□I□N

Pour le service robotique, ce stage a été mené à terme. Son fonctionnement est conforme au cahier des charges, et les disfonctionnements de la carte wrappée constatés, devraient être résolus sans grande difficulté. De toute façon, ces dommages n'altèrent pas le fonctionnement du dispositif de détection d'obstacles.

L'intégration à l'architecture matérielle du CyCab par le bus CAN2 et le traitement de la position des obstacles par triangulation ou par modèle probabiliste, se fera dans le cadre d'un autre projet.

Sur un plan personnel, ce stage a été très formateur, car il m'a permis de gérer et de mettre en œuvre toutes les étapes de la réalisation d'un projet, en partant de la spécification, de l'étude des capteurs, de la réalisation d'une carte d'interfaçage, en passant par l'étude d'un microcontrôleur complexe, de sa programmation en assembleur et en langage C, pour finir par le câblage et la conception des boîtiers pour intégrer l'ensemble du système.

J'ai ainsi pu voir tous les aspects d'une étude, en travaillant, d' une part, sur l'élaboration et l'intégration du matériel, d'autre part, sur la mise en œuvre par le microcontrôleur et enfin sur la maintenance et le debuggage de ce système.

Pour réaliser ce système de détection d'obstacles, il m'a fallu acquérir une bonne méthode de travail à la fois pour résoudre les multiples problèmes matériels que j'ai rencontrés et pour gérer l'avancement de toutes les étapes du projet. Ainsi il m'a fallu gérer ma propre charge de travail, mais aussi celle des personnes qui possédaient les compétences nécessaires à la résolution des problèmes qui se présentaient.

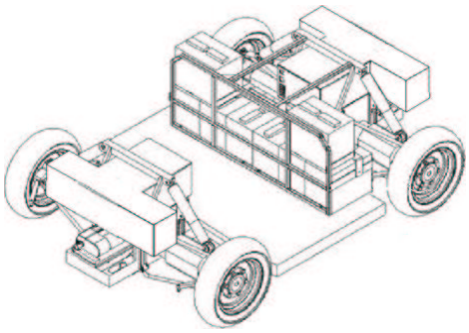
J'ai aussi pu me familiariser avec l'environnement de travail et de développement UNIX, qui est extrêmement implanté dans l'industrie et à l'INRIA.

Ce stage représente pour moi une réelle immersion dans la vie professionnelle et achève ma formation en IUP, formation que j'avais choisie pour agir sur des procédés physiques à l'aide d'outils informatiques.

Anne□es

□□ □R□□□N□A□I□N □ □□ANI□U□ □U □□□A□

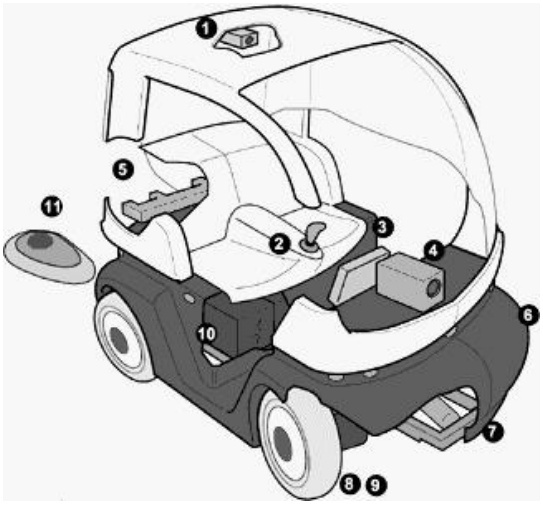
Le CyCab est un prototype de petit véhicule électrique devant servir de plate-forme expérimentale aux équipes de robotique et vision de l'Unité de Recherche Rhône-Alpes. Il a été construit sur la base d'un châssis tubulaire raccourci qui supporte une coque.



□i□□re □□□: sché□ □d□ ch□ssis dés□ssé d□ □□□□

Ce véhicule est en fait une voiture de golf Andruet totalement revue et corrigée.

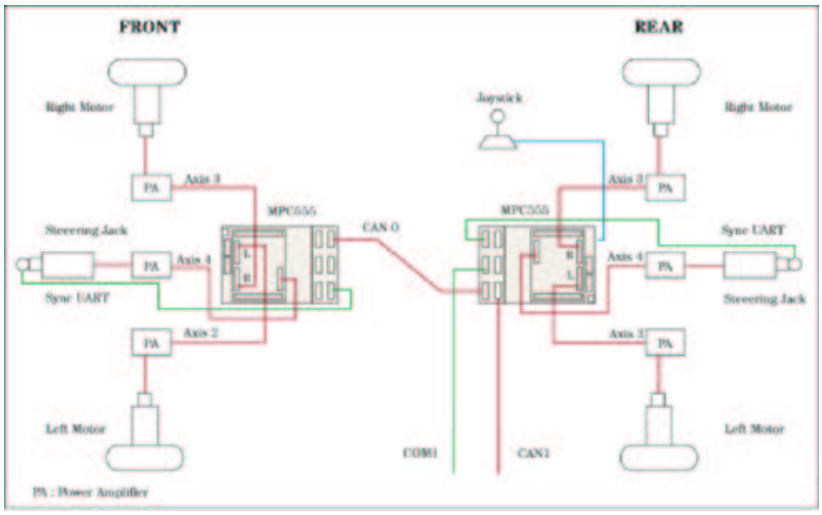
- 4 roues motrices et directrices indépendantes ;
- un dispositif de frein mécanique actionnable électriquement ;
- Poids du véhicule à vide : environ 300 Kg ;
- Précision des déplacements : 1 cm/s
- Rayon de braquage : 3 m
- Vitesse maximale du véhicule : 30 Km/h sur terrain plat ;
- Pente maximale 10% ;



1. Caméra CCD pour la télé opération
2. Joystick central de commande pour la conduite sécurisée
3. Terminal multimédia
4. Caméra linéaire pour la conduite en train
5. Balises infra rouges pour la conduite en train
6. Capteurs ultrasons pour la détection d'obstacles
7. Vérin de direction électrique
8. moteur électrique par roue
9. frein électrique par roue
10. batteries + gestionnaire automatique de charge
11. Borne de recharge par induction fixée sur la voirie

□i□□re □□□: □□e écl□tée d□ □□□□

□□ □R□□□N□A□I□N □□□□□RI□U□ □U □□□A□



□i□re □□: □ché□ □de c□□l□□e des n□□ds et des □ □te□rs d□ □□□ □□

L'alimentation du CyCab est assurée par 8 batteries de 12 V qui assurent une alimentation générale de 24 ou 48 V. Le CyCab est un véhicule électrique composé de 4 moteurs à courant continu de 900 Watts chacun assurant la propulsion et de 2 moteurs MCC pour la direction. Les moteurs de traction ont été modifiés. Chaque moteur 900 W (48 V) est couplé à un frein à manque de courant et d'un codeur incrémental.

Comme on peut le voir sur la figure précédente, chaque train est contrôlé par un boîtier technique contenant toute l'électronique de contrôle, de commande et de puissance nécessaires. Ces boîtiers contiennent 3 PWM (un pour la commande du moteur de direction et deux pour les moteurs de tractions), 2 relais pour les freins à manque de courant, les codeurs incrémentaux et bien sûr le nœud ROBOSOFT MPC555 pour commander tout ceci.

Afin de parer à tous les problèmes, deux arrêts d'urgence sont installés, l'un est situé dans l'habitacle et l'autre est actionnable à distance. Un relais statique sur la boucle d'arrêt d'urgence met le véhicule dans l'état freiné s'il n'est pas alimenté. En série avec le secondaire de ce relais statique est monté un interrupteur de commande manuelle des freins de parking.

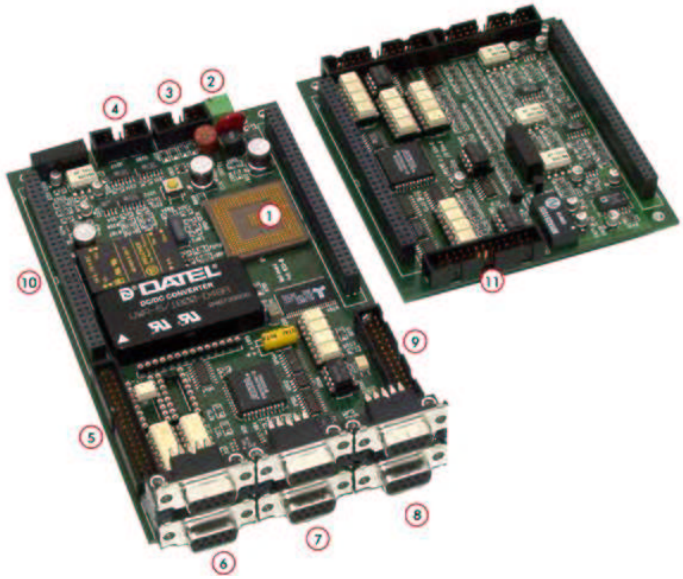
Le véhicule dispose de 2 nœuds (à base de power PC MPC555, voir descriptif plus loin), pour assurer la direction, le freinage et le contrôle de la vitesse des moteurs.

Au niveau de la communication, les deux nœuds communiquent via 2 réseaux *CAN*[11] et un réseau Ethernet avec un hub disponible pour la connexion de caméra et la programmation des nœuds. En plus des 2 nœuds, ce véhicule possède un PC embarqué permettant le dialogue avec un utilisateur à travers un écran tactile et un ensemble clavier / souris.

[11] CAN : bus série asynchrone, à 2 fils symétriques (trés utilisé dans le monde automobile).

□□ □□□□RI□□I□N □ A□□RI□□□□ □□UN N□ □U□

Le nœud ROBOSOFT est en fait une carte mère plus une carte mezzanine composées de nombreux ports d'entrées et sorties commandés par un microcontrôleur 32 bits MPC555, à cœur Power-PC (η). Pour l'application de la ceinture ultrasonique, nous n'utiliserons pas la carte mezzanine, car elle sera remplacée par la carte à wrapper.



□i□□re □□□: ph□t□d□□n n□□d R□ □□ □□ □□

Le principal avantage de cette carte est de posséder de nombreux connecteurs d'extension, qui la rende très évolutive :

- Un connecteur d'alimentation (ι),
- Un connecteur BDM utilisé pour débuguer la carte par les outils SDS livrés (φ),
- Un connecteur de 8 entrées analogiques (κ),
- Un connecteur pour 8 entrées et 8 sorties optocouplées (λ),
- 6 embases DB9 (2 ports série synchrones μ, 2 ports RS232 ν, 2 ports CAN ο),
- Un connecteur pour contrôler un axe (π),
- 2 connecteurs 64 points (θ) vers lesquels vient s'enficher la carte mezzanine (11)
- Un écran LCD permet de créer une petite interface homme machine,

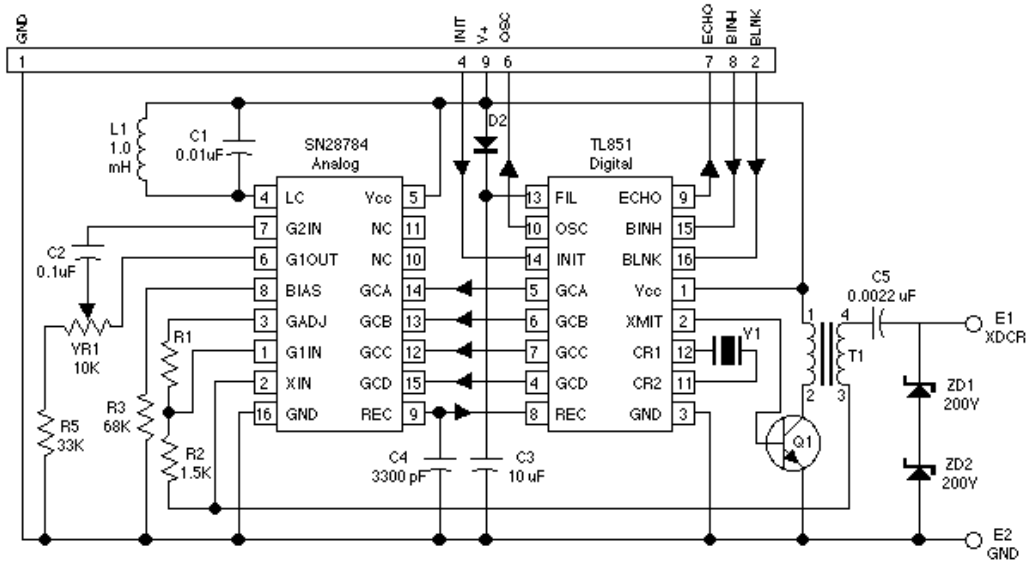
Le microcontrôleur MPC555 présent sur la carte mère, est un microcontrôleur 32 bytes nouvelle génération de MOTOROLA. Il se présente sous la forme d'un boîtier *FPGA*[12] de 272 broches, alimenté en 3,3 et 5 V et fonctionnant à 40 MHz entre –40 et +125 degrés. Ce microcontrôleur possède en interne :

- Un cœur Power-PC avec une unité de calcul en virgule flottante,
- 26 Kbytes de mémoire RAM rapide et 6 Kbytes de RAM pour microcode TPU,
- 448 Kbytes de mémoire Flash EEPROM pour implanter le programme,
- De nombreuses entrées / sorties 5V,
- Système timer 50 canaux avec 2 unités TPU,
- 32 entrées analogiques,

[12] FPGA : Plastic Ball Grid Array.

- 16 sorties PWM, etc…

□□ □□□AI□ □U □□N□□I□NN□□ □N□ □□□ □I□□ □□NAR □□ □AR□I□ □□□□



□i□□re □□□: sché□ □électr□ni□□e de l□c□rte □□□□

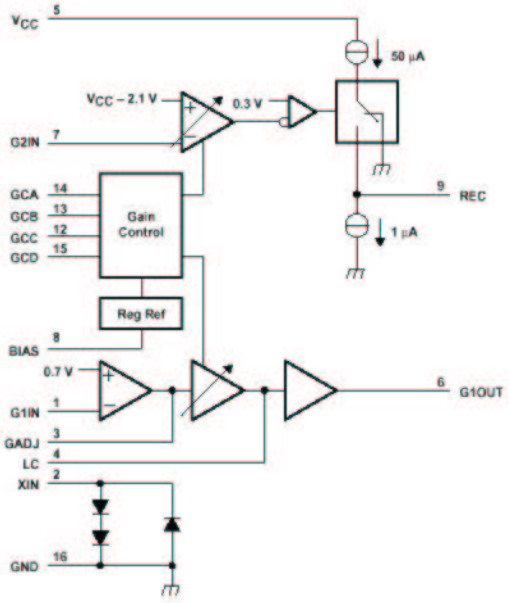
Ce kit 6500 est composé d'un cristal piézoélectrique qui est excité par des circuits dédiés, le TLP851 et le TLP852 (SN28784). Le TLP852 est un circuit permettant de détecter d'infimes vibrations (réceptions d'une onde). Ces vibrations sont amplifiées par un dispositif de gain progressif qui est appliqué par le circuit TLP851 dédié quant à lui au calcul de distance. Il comporte la commande du gain progressif et un quartz servant d'horloge pour le calcul de la distance. Il permet aussi d'exciter un cristal piézoélectrique (émission d'une onde).

L'excitation de ce cristal se fait après avoir appliqué un signal haut sur l'entrée INIT du TLP851, ce qui provoque l'émission de 16 salves à la fréquence du quartz divisé par 8,5. Ses 16 salves sont alors amplifiées par un transformateur pour fournir une dynamique optimale au cristal. Le dispositif cesse alors d'émettre et une attente de 2,8 ms est alors générée par le TLP851. Cette attente est générée pour permettre une relaxation totale du cristal piézoélectrique, car le même cristal sert à la fois pour l'émission et la réception d'une onde sonore ce qui permet un gain de place évident. Après ce délai de 2,8 ms le TLP851 cesse de masquer son entrée REC (réception). Le kit 6500 bascule alors en mode réception.

Une commande gain progressif est alors envoyée au TLP852, ce qui a pour effet d'appliquer de manière graduelle le gain au signal envoyé par le cristal. On peut ainsi mesurer une distance de 15 cm à gain nul à 100 m lorsque le gain est maximal. Voir la figure 4.2 page suivante.

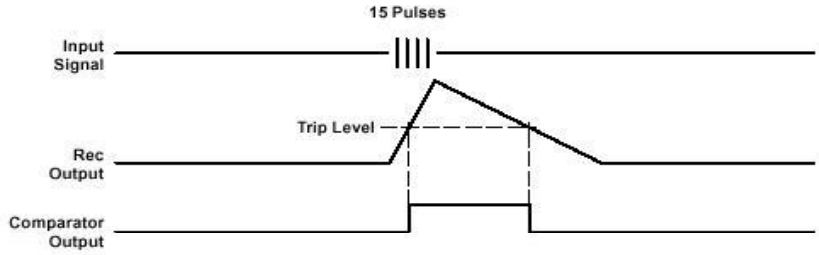
Cette amplification croissante permet de ne pas amplifier du bruit, qui de toute façon ne peut pas être élevé, car le cristal piézoélectrique a une plage de variation très faible. L'émission et la réception sont optimales pour une onde de fréquence 45kHz alors que les ondes de fréquences différentes sont rapidement et fortement atténuées.

Une fois qu'un signal est reçu et amplifié, il est comparé, par le TLP851, à une valeur de référence de 1,2V. Dés que le signal reçu dépasse ce seuil, un créneau est généré par un comparateur. Ce créneau indique qu'un écho a été détecté.



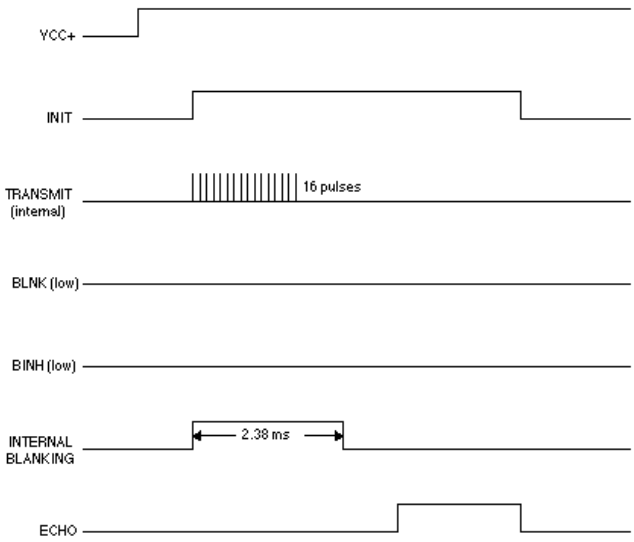
□i□□re □□□: sché□ □de principe d□c□ntrôle□r de □□in

Pour connaître le temps de vol aller et retour de l'onde sonore, il suffit de compter le temps écoulé entre le moment de l'application du signal d'initialisation et la réception de l'écho.



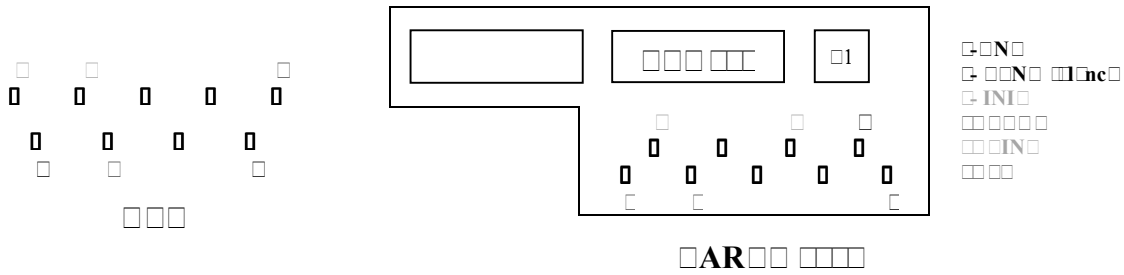
□i□□re □□□: sché□ □d□si□n□l □□□□ □is en □r□e

En mode single-echo, le signal ECHO est initialisé à 0, quand le INIT passe à l'état bas.



□i□□re □□□: chr□n□□r□□ □e en □ □de sin□le-ech□

□□ □□□□□□ A □□ □A□□A□□ □□□□ □I□□ □□ □AR□I□ □UR □□□ □□□

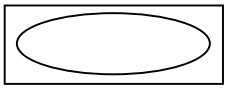


□AR□□ □□□□

□□rte □□□□	□□□	□□□le□r des □ils	□□ncti□n s□r le □it □□□□	□□ncti□n s□r l□c□rte □□r□pper
4	1	Jaune	INIT	PWM
2	6	Blanc	BLNK	Mode single-echo
8	2	Vert	BINH	Mode single-echo
7	7	Marron	ECHO	TPU
1	9	Noir	GND	Relais
9	5	Rouge	+ 5V	Relais
	carcasse	Gris	Connecté aux boîtiers	Connecté au boîtier

□□ □□□□□□ □□□□ □RI□□AU□ □I□□□□ □□□□□□RI□U□□

Le constructeur donne un angle asymétrique d'émission et de réception : 35° ± 3° et de 17° ± 2°.



Or, les tests des cristaux que nous avons effectués vérifient bien ces données, mais il est plus vraisemblable de prendre en compte l'angle de détection réel d'objets. Il s'agit de l'angle qui a permis d'avoir un retour d'écho sans avoir eu besoin de bouger l'objet.. Ainsi on s'oriente plutôt ver des angles de 25° ± 3° et de 22° ± 4°.

□□□□□□siti□n □□□thé□ri□□e

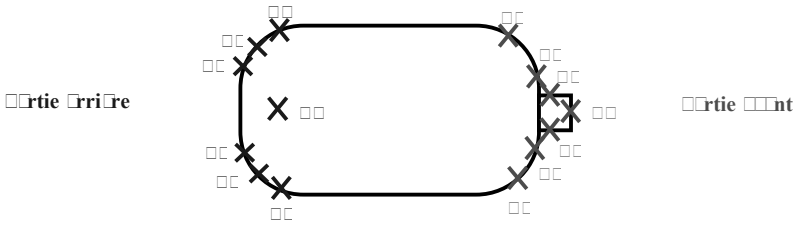
C1 = 39,5°	C5 = 19,5°	C9 = 37°	C13 = 51°	C17 = 29°	C21 = 42°
C2 = 42°	C6 = 32°	C10 = 21°	C14 = 36°	C18 = 27°	
C3 = 30°	C7 = 30°	C11 = 26°	C15 = 29°	C19 = 43°	
C4 = 29°	C8 = 27°	C12 = 29°	C16 = 29°	C20 = 36°	

□□□□□□siti□n □□□thé□ri□□e

C1 = 31°	C5 =	C9 = 34°	C13 = 27°	C17 =	C21 = 33°
C2 = 21°	C6 = 40°	C10 =	C14 = 32,5°	C18 =	
C3 = 33°	C7 =	C11 =	C15 = 31,5°	C19 = 31,5°	
C4 =	C8 =	C12 = 44°	C16 = 28°	C20 = 36°	

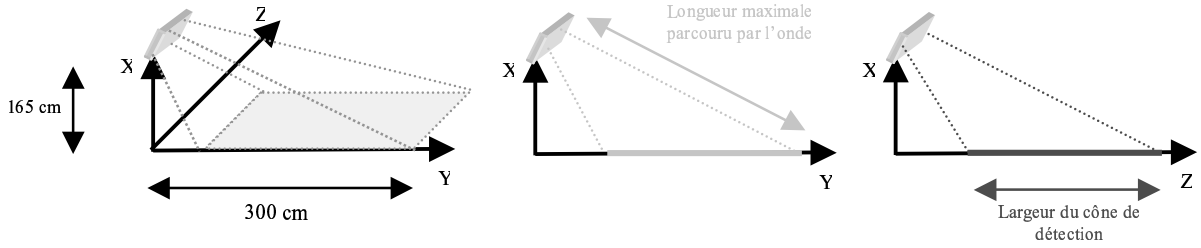
□□ □I□□ □N □□A□□ □□ □RI□N□A□I□N □□□ □A□□□UR□

Le positionnement des capteurs sur le CyCab, pour avoir une surface de détection maximale, est donné par le schéma suivant. Notons que les capteurs C1, C2, et C7 de la partie arrière, sont positionnés sur le toit, les autres sont sur la ceinture du CyCab.



□i□□re □□□: rép□rtiti□n des c□pte□rs s□r le □□□□

L'orientation bidimensionnelle des capteurs est assez complexe. Elle se fait par paire de capteurs. La moindre variation de l'orientation entraîne une déformation importante de la projection du cône de détection sur le sol. Bien évidemment les angles théoriques sont très difficiles à respecter compte tenu du manque de graduation sur les supports des capteurs. Toutefois, la projection du cône de détection se fait selon les trois axes (X, Y, Z).



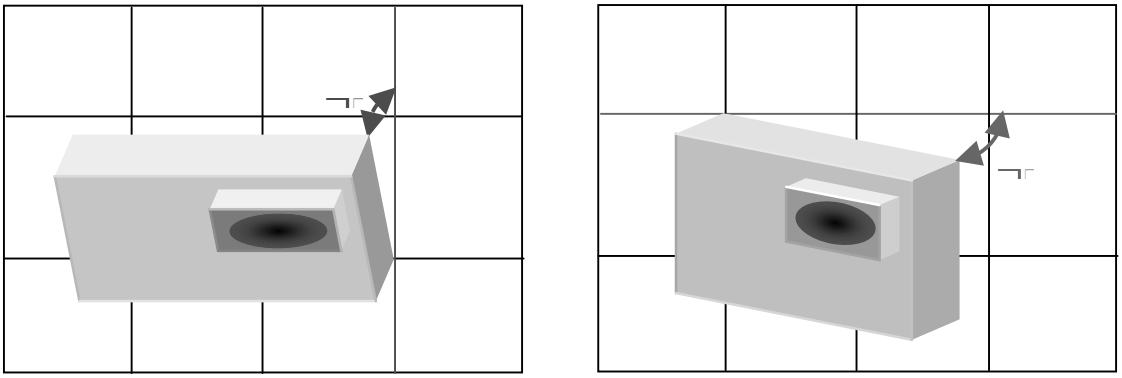
□i□□re □□□: pr□□ecti□n tridi□ensi□nnelle de l□□nde □ltr□s□n□re

En effet, pour déterminer le cône de détection, il faut faire une projection de l'onde sur chaque repère. La projection ne pose pas de problème, car la hauteur (165 cm pour les capteurs de toit et 65 cm pour les autres), la distance maximale de mesure et les angles d'émission et de réception réelles des piézoélectriques, sont connus.

Après la projection de la zone de détection sous la forme d'un rectangle, il ne reste plus qu'à extrapoler une ellipse pour représenter au mieux la position de l'onde. Une extrapolation suffit car les angles des piézoélectriques ne sont en fait qu'une moyenne et les ondes sonores ne se répandent pas uniformément dans cette zone.

Le calcul de ces angles nous amène à la zone de détection donnée en page 13 et aux résultats qui suivent.

Considérons la représentation suivante pour déterminer les angles matérialisant l'orientation des capteurs.



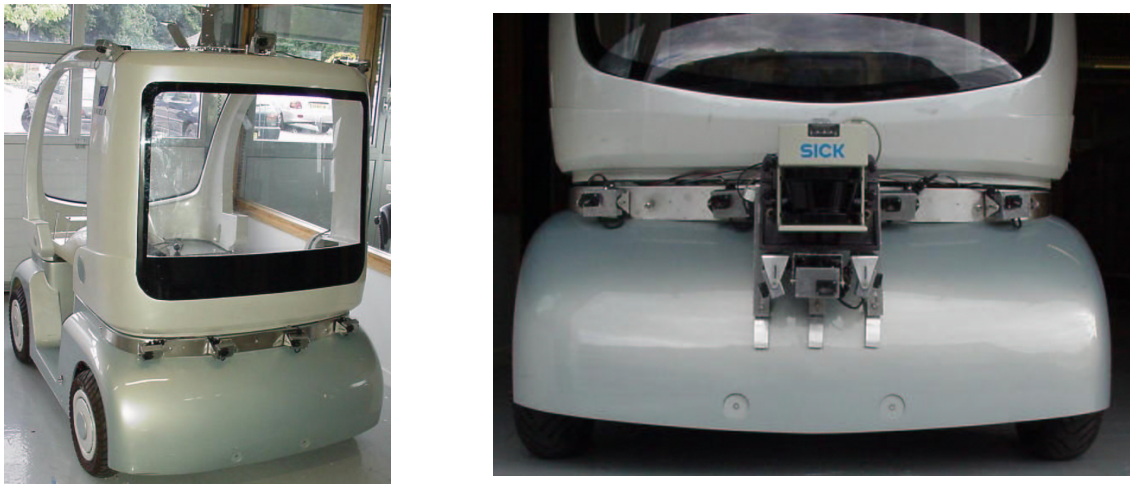
□i□□re □□□: □n□le d□□rient□ti□n des c□pte□rs déter□in□nt le□r p□siti□n □idi□ensi□nnelle

Les deux tableaux suivants, révèlent les angles qu'il faut appliquer aux capteurs, pour avoir la zone de détection citée auparavant.

	□□pte□rs □□□nt						
	C1	C2	C3	C4	C5	C6	C7
¬1	30°	9°	14°	22°	30°	14°	22°
¬2	15°	0°	60°	40°	- 15°	- 60°	- 40°
R_{GAIN}	22 K∠	22 K∠	22 K∠	22 K∠	22 K∠	22 K∠	22 K∠

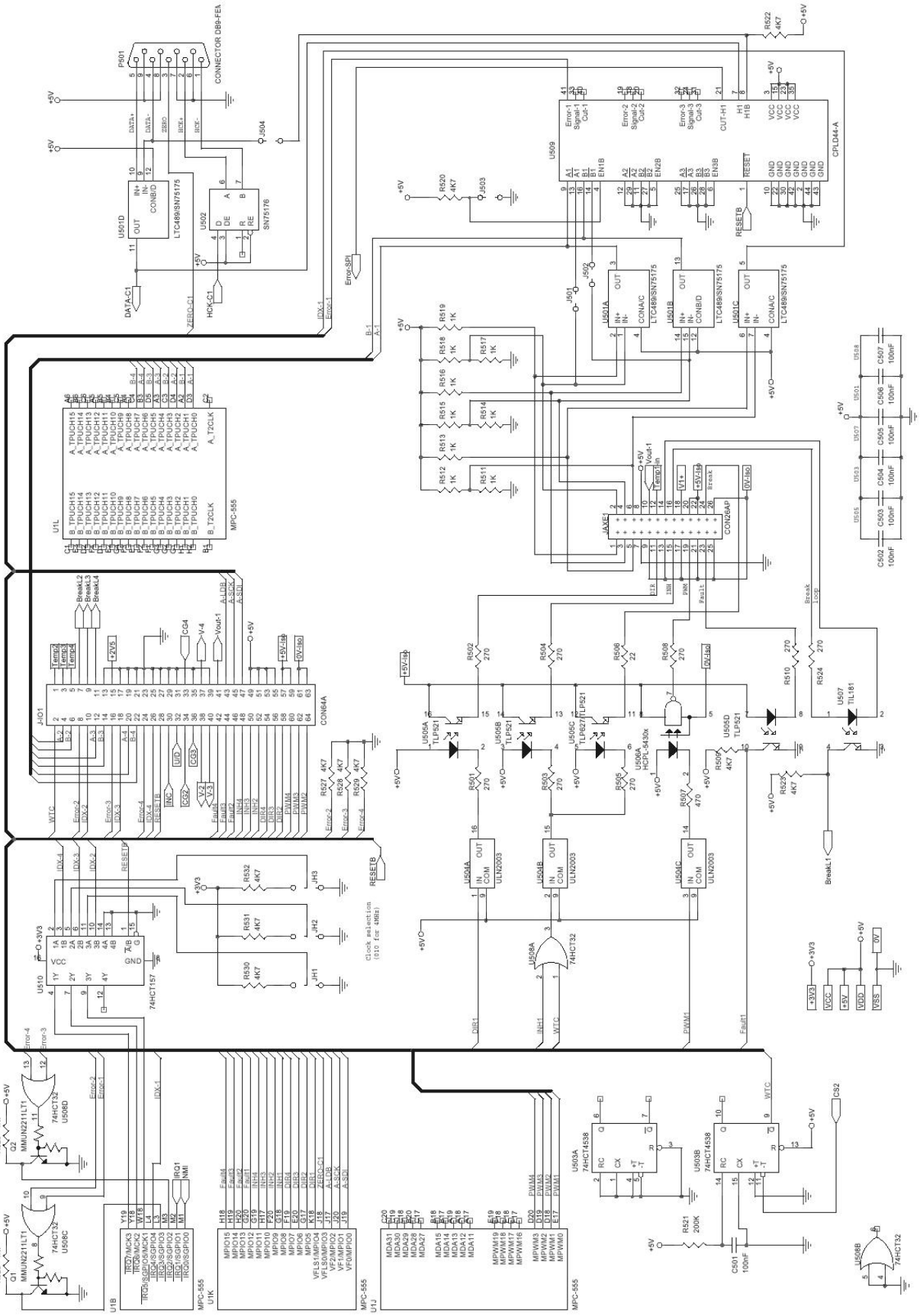
	□□pte□rs □rri□re						
	C1	C2	C3	C4	C5	C6	C7
¬1	24°	65°	22°	14°	14°	22°	65°
¬2	0°	15°	40°	60°	- 60°	- 40°	-15°
R_{GAIN}	68 K∠	68 K∠	22 K∠	22 K∠	22 K∠	22 K∠	68 K∠

Pour mieux visualiser l'emplacement des capteurs, voilà comment ils ont été implantés :

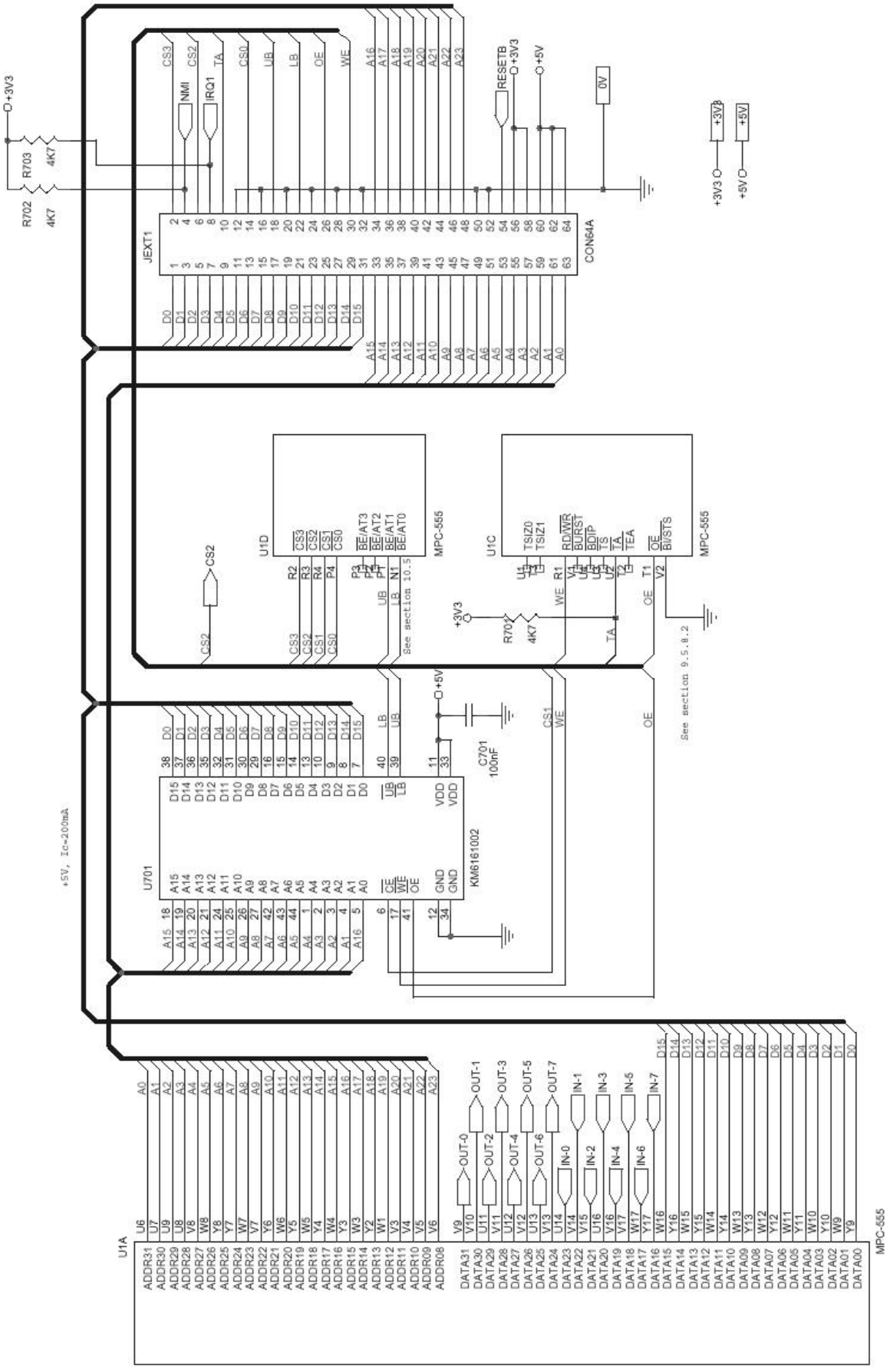


□i□□re □□□: e□pl□ce□ent des c□pte□rs s□r le □□□□

□□ □□□□□□ A □U N□ □U□ R□ □□ □□ □□

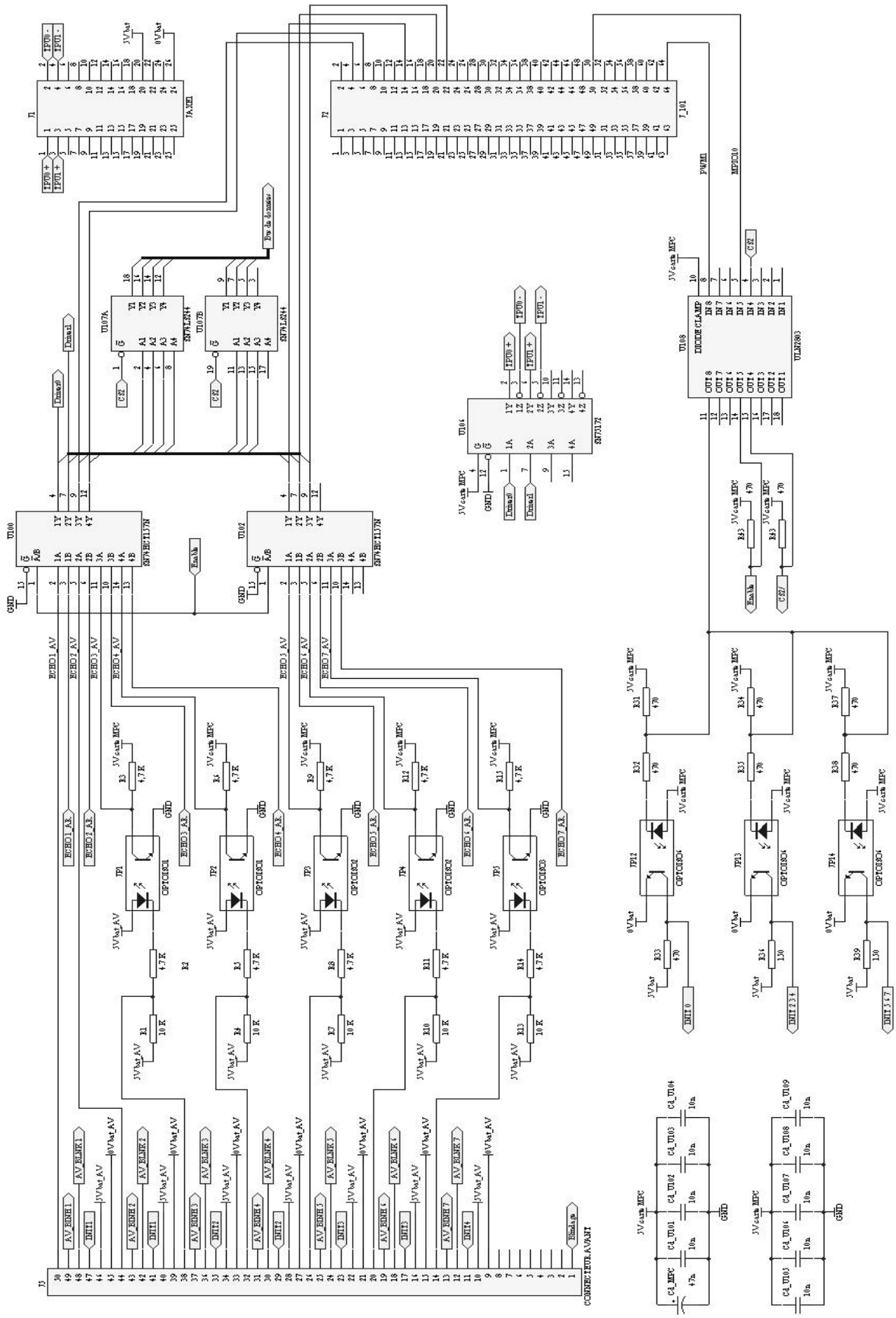


□i□□re □□□: sché□ □d□n□□d R□ □□ □□ □□ □□ □énér□le et □□ □ □

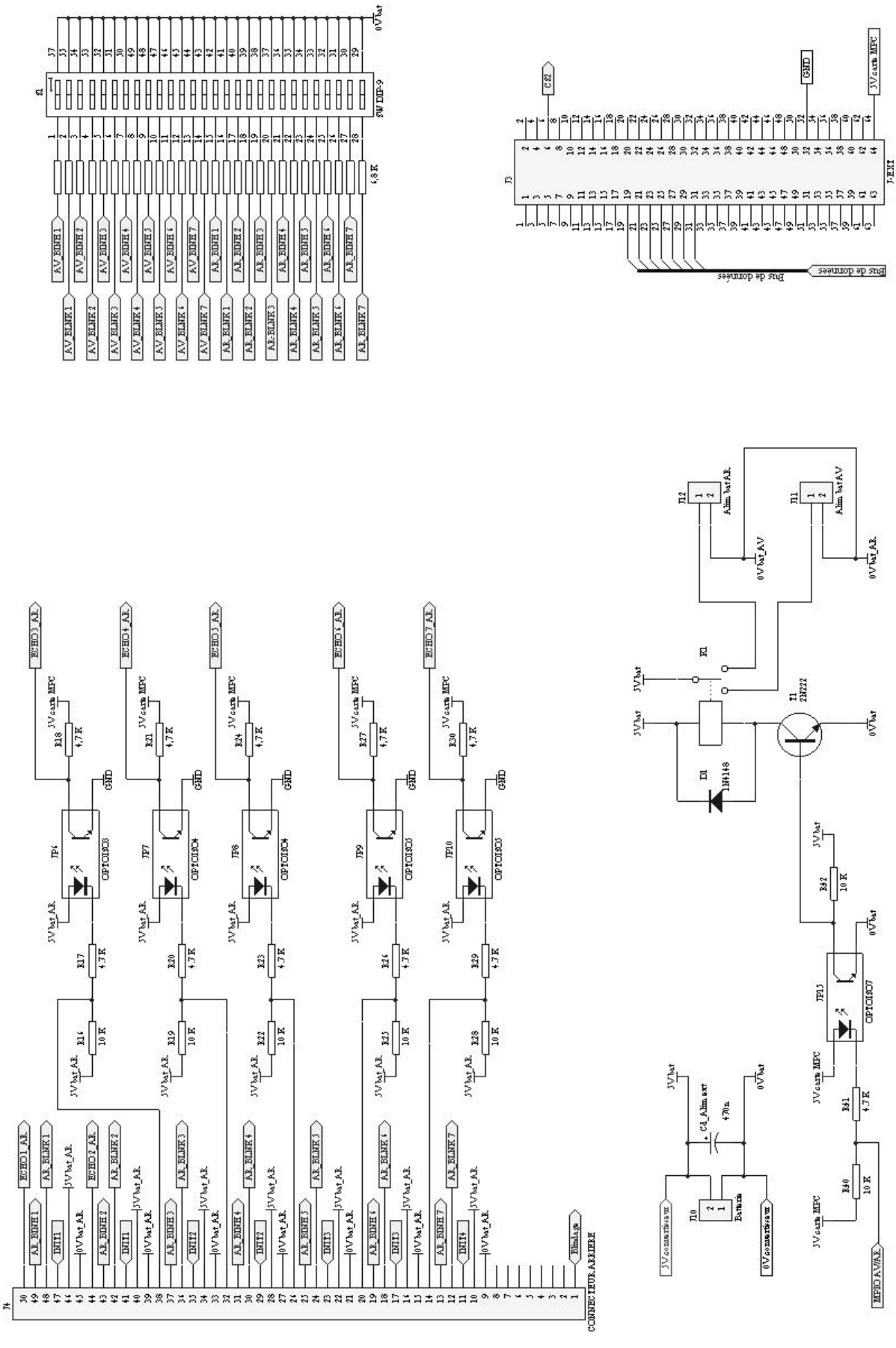


□i□re □□□: sché□ □d□n□□d R□ □□ □□ □□ □□é□ □ire □-RA□ et □□s de d□nnées□

□□ □□□□□□ A □□ □A □AR□□ A □ RA□□□R



□i□re □□: □ché□ □de l□c□rte □□r□pper □pt□c□□ple□rs□I□ □énér□le□□□ □ et □□U□



□i□re □□: □ché□ □de l□c□rte □□r□pper □pt□c□□ple□r□□s de d□nnées et rel□is□

□□N□□ □N□□A□UR□

descripti□n	□□rt □□pe	dési□n□ti□n	c□□ □ent□ires
Multiplexeur	SN74HCT157N	U100	
Multiplexeur	SN74HCT157N	U102	
Driver de ligne	SN75172	U106	
Latch	SN74LS244	U107	
Buffer darlinton	ULN2803	U108	
Transitor NPN	2N222	T1	
Relais	G2RL-1-E	Relais	5 V, 16 A
Diode de roue libre	1N4148	D1	
Optocoupleur	OPTOISO1	JP1	
Optocoupleur	OPTOISO1	JP2	
Optocoupleur	OPTOISO2	JP3	
Optocoupleur	OPTOISO2	JP4	
Optocoupleur	OPTOISO3	JP6	
Optocoupleur	OPTOISO3	JP5	
Optocoupleur	OPTOISO4	JP7	
Optocoupleur	OPTOISO4	JP8	
Optocoupleur	OPTOISO5	JP9	
Optocoupleur	OPTOISO5	JP10	
Optocoupleur	OPTOISO6	JP13	
Optocoupleur	OPTOISO6	JP12	
Optocoupleur	OPTOISO6	JP14	
Optocoupleur	OPTOISO7	JP15	
Résistances de tirage INIT	150	R39, R36	¼ de Watt
Résistances de limitation et de tirage	470	R31, R32, R33, R34	¼ de Watt
Résistances de limitation et de tirage	470	R35, R37, R38, R43	¼ de Watt
Résistances de tirage	4,7 K	R2, R3,R5, R6, R8, R9	¼ de Watt
Résistances de tirage	4,7 K	R11, R12, R14, R15	¼ de Watt
Résistances de tirage	4,7 K	R17, R18, R20, R21	¼ de Watt
Résistances de tirage	4,7 K	R23, R24, R26, R27	¼ de Watt
Résistances de tirage	4,7 K	R29, R30, R41	¼ de Watt
Résistances de réglage du mode	6,8K		¼ de Watt
Résistances de limitation de courant	10 K	R1, R4, R7, R10, R13	¼ de Watt
Résistances de limitation de courant	10 K	R16, R19, R22, R25	¼ de Watt
Résistances de limitation de courant	10 K	R28, R40, R42	¼ de Watt
Capacités de découplage	10n	Cd_U106 à Cd_U109	Aluminium
Capacité de découplage alim. MPC	47n	Cd_MPC	Chimique
Capacité de découplage	470n	Cd_Alim ext	Chimique
Connecteur carte MPC555	JAXE1	J1	
Connecteur carte MPC555	J_101	J2	
Connecteur carte MPC555	J-EXT	J3	
Connecteur de connexion capteurs	CONNECTEUR ARRIERE	J4	
Connecteur de connexion capteurs	CONNECTEUR AVANT	J5	wrapper
Connecteur	Batterie	J10	wrapper
Connecteur	Alim bat AV	J11	
Connecteur	Alim bat AR	J12	
Switch de selection du mode	SW DIP-9	S1	wrapper

□i□li□□r□phie

□□R□pp□rts techni□□es

[1] Gerard Baille, Philippe Garnier, Herve Mathieu, Roger Pissard-Gibollet : ∀Le Cycab de l'INRIA Rhone-Alpes∀, # 0229, avril 1999

[2] J-M Bourgeat, P-A Degaches et C. Despinasse : ∀Ceinture à ultrasons∀, IUT GEII, 99/00

□□□c□□ent□ti□n c□nstr□cte□r

[3] Motorola, ∀MPC555 User's Manual∀

[4] Motorola, ∀MPC55Time Processor Unit Manual∀

[5] Polaroid, ∀6500 Series Sonar Ranging Module∀

[6] Polaroid, ∀ Polaroid Series 9000 Environmental Grade Transducer∀

□□□ites internets

[7] La page de l'INRIA Rhône Alpes : http://www.inrialpes.fr

[8] La page du service Robotique, Vision et Réalité Virtuelle : http://www.inrialpes.fr

[9] La page du distributeur du kit 6500 : www.acroname.com/robotics/info/articles/sonar/sonar.html

[10] La page de Motorola : http://www.motorola.com

[11] La page de ROBOSOFT : http://www.robosoft.fr

[12] La page de Texas Instrument : http://www.ti.com