

Université François-Rabelais de Tours

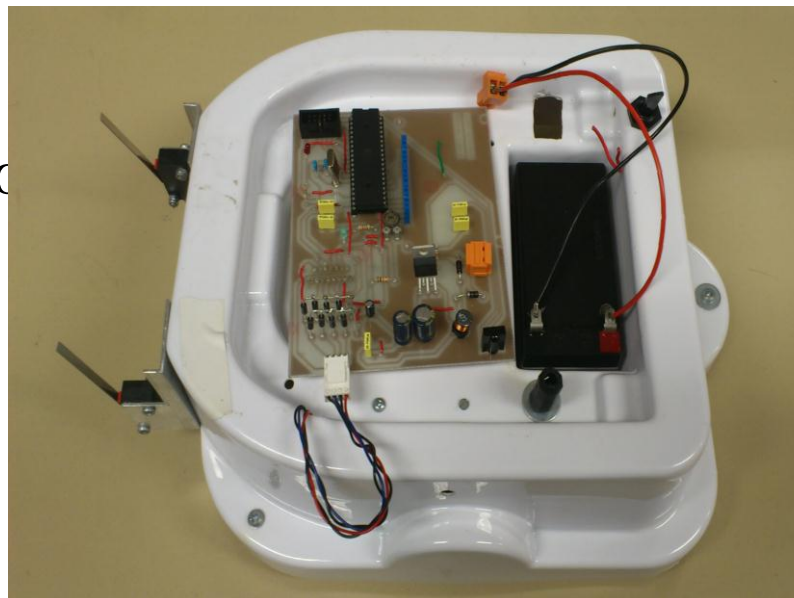
Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

Projet d'étude et réalisation

Robot télécommandé par infra-rouge

Robo



e

Giraud Bastien
Bouguereau Thomas
Groupe Q2

Enseignants
Lequeu Thierry
Auger Philippe

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

UNIVERSITE FRANCOIS-RABELAIS
TOURS



Institut Universitaire de Technologie

Département
GENIE ELECTRIQUE ET
INFORMATIQUE INDUSTRIELLE

Giraud Bastien
Bouguereau Thomas
Groupe Q2

Enseignants
Lequeu Thierry
Auger Philippe



Sommaire

Introduction.....	4
1.Présentation du projet.....	5
1.1.Cahier des charges.....	5
1.2.Planning prévisionnel et réel.....	6
2.L'électronique.....	7
2.1.Présentation des composants utilisés à la création de la carte.....	7
2.2.Visualisation des composants utilisés sur la carte.....	13
3.La programmation.....	14
3.1.La réception de la trame rc5.....	14
3.2.Traitement des données de la trame.....	16
4.Mécanique.....	17
4.1.Le châssis.....	17
4.2.Le bloc moteur.....	17
5.Tests.....	19
5.1.Sur la carte électronique.....	19
5.2.Visualisation des trames de la télécommande TV.....	20
5.3.Mécanique.....	20
Conclusion.....	21
Résumé.....	22
Bibliographie.....	24
Annexes.....	25

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle



Introduction

Durant le 3^{ème} semestre de la 2^{ème} année de la formation GEII nous sommes conduits à réaliser un projet d'étude et réalisation.

Nous avons choisi notre propre sujet, c'est-à-dire de télécommander par infrarouge un robot réalisé par des étudiants des années antérieures. Nous avons pour cela treize séances pour réaliser ce projet.

Pour réaliser celui-ci il nous a fallu refaire une carte électronique en implantant les composants nécessaires au guidage infrarouge, programmer celle-ci pour que le robot respecte les contraintes imposées par le projet.

1. Présentation du projet

1.1. Cahier des charges

Robot télécommandé par infrarouges

Programmation d'un micro-contrôleur gérant les actions de la télécommande infrarouge.

Partie électronique : réalisation de la carte électronique

- Alimentation +5V
- Pont en H (moteur)
- Récepteur infra-rouge (TSOP)
- ATMEGA8535 (micro-contrôleur)

Partie informatique : gérer les événements créés par la télécommande

- bouton de marche
- tourner
- marche avant et arrière

Partie mécanique : vérification de son fonctionnement

- Châssis et moteurs récupérés sur un ancien projet
- Tests moteurs

1.2. Planning prévisionnel et réel



Taches \ Planning	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3	
Prise de connaissance du sujet	■					■									■	■			
Recherche d'informations	✕					■									■	■			
Formation Orcad						■									■	■			
Réalisation du typon						■									■	■			
Soudure						■									■	■			
Programmation						■									■	■			
Assemblage						■									■	■			
Test						■									■	■			
Rédaction du projet						■									■	■			

- Planning prévisionnel
- Planning réel
- Vacances scolaire

Giraud / Bouguereau
Groupe Q2

2. L'électronique

Cette partie a été la plus longue durant notre projet à cause des tests qui ne convenaient pas à nos attentes.

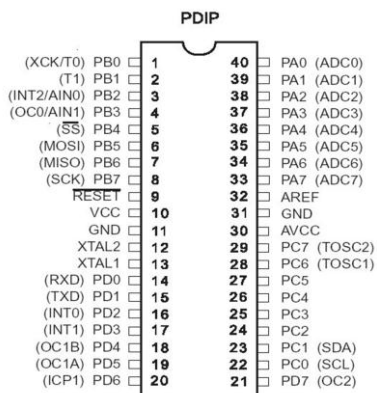
2.1. Présentation des composants utilisés à la création de la carte

2.1.1. L'ATMEGA 8535

L'ATMEGA8535 est un microcontrôleur, c'est ce composant qui va être programmé. Il est la pièce centrale de notre projet. Il sert à recevoir la trame de la télécommande infra-rouge transmise par le récepteur infra-rouge. Il traitera aussi cette trame pour détecter la touche sur laquelle on a

appui
é et

agir en fonction pour faire avancer, reculer ou tourner le robot.



Principales caractéristiques :

- Vitesse de cadence jusqu'à 16 MHz
- Tension d'alimentation de 4.5 V à 5.5 V
- Courant d'alimentation : 10mA
- 40 PINs
- 32 entrées/sorties
- 8 canaux analogiques

- Convertisseur CAN 8 bits ($0 \rightarrow 1023 = 1024$ possibilités)
- 2 sorties PWM (MLI)
- Courant maximal par entrées/sorties : 40 mA
- 8 Ko de mémoire programme
- EEPROM de 512 Octets
- SRAM de 512 Octets
- 3 timers (2 de 8bits, 1 de 16 bits)
- Programmation langage

2.1.2. Le L298N

L'ATMEGA8535 ne fournissant pas assez de courant pour alimenter les moteurs directement, nous avons choisi de passer par ce composant qui va être commandé par l'ATMEGA8535 et qui fournira de la puissance aux moteurs. Ceux-ci demandent effectivement 1 A pour fonctionner alors que l'ATMEGA8535 ne fournit que 40mA à ses sorties (voir précédemment).

Principales caractéristiques :

- Tension d'alimentation du circuit logique : 5 V
- Tension d'alimentation du circuit puissance : $0 \rightarrow 48V$
(nous serons à 12 V)
- Tension minimale de sensibilité MLI : 2.3 V
- Courant maximal du circuit puissance : 3 A

Ce composant est un composant à 15 broches. Chacune d'elles ont une fonctionnalité bien précise. Comme on peut le constater, le L298N est composé de 8 portes logiques ET. C'est grâce à ces portes logiques que nous pourrions choisir le sens de rotation du moteur et l'inverser.

Les entrées ENA et ENB servent d'autorisation (d'enable en anglais), sans un 1 logique sur ces entrées, aucun moteur ne peut fonctionner.

Les composants se divisent en deux parties identiques (on n'utilise qu'un composant pour commander les deux moteurs). On dira donc que l'entrée IN3 est équivalent à l'entrée IN1 et de même pour les entrées IN2 et IN4.

IN1	IN2	Sens de rotation du moteur
0	0	Ne tourne pas
0	1	Tourne dans un sens
1	0	Tourne dans le sens inverse du précédent.
1	1	Ne tourne pas

Le pin Vs est l'entrée sur laquelle on va mettre la tension d'alimentation des moteurs avec la puissance requise, cette tension sera décernée au moteur quand les différentes entrées IN seront configurées pour avoir le sens de rotation du moteur souhaité.

Le pin V_{SS} est l'in sur lequel on met la tension de référence pour que le composant reconnaisse les changements de niveau logique sur les entrées IN.

Les pins SENSE servent à comparer le courant requis par le moteur pour leur fournir celui-ci.

Les pins OUT sont ceux où nous mettrons nos moteurs, ils sont couplés par deux, un moteur sera branché entre OUT1 et OUT2, le deuxième moteur sera branché entre OUT3 et OUT4.

Nous avons aussi choisis de mettre des diodes avant les moteurs pour que ceux-ci ne se retrouvent pas en court-circuit. En effet, pendant un changement de sens de rotation des moteurs, les transistors se retrouvent en commutation simultanément et cela peut produire des court-circuits.

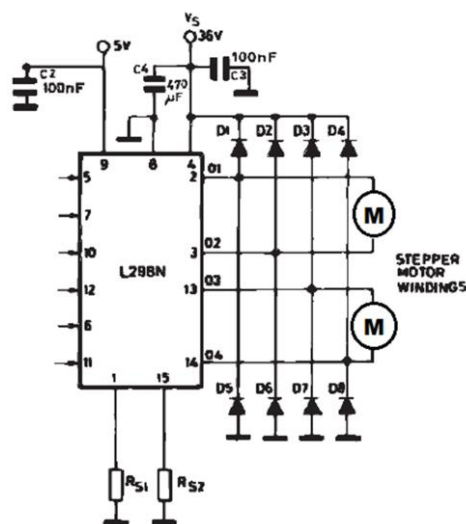


Illustration 5: pont en H montage diode

2.1.3. Le régulateur LM2575

Le choix de ce régulateur est dû à la consommation des moteurs en charge. Nous avons constaté que ces moteurs demandent 1 A en charge normal (avec le poids de la batterie, du châssis et de la carte). Ce régulateur nous permet d'avoir cette puissance, ce qui justifie notre choix.



Illustration 6:
LM2575

Principales caractéristiques :

- Tension maximale d'entrée : 18 V
- Tension de sortie : 5 V
- Courant maximal de sortie : 1 A

2.1.4. L'écran LCD

Pour simplifier la programmation surtout lors du débogage, nous avons choisi de mettre sur notre carte un écran LCD sur lequel nous pourrions afficher certaines variables internes au programme. Nous pourrions ainsi vérifier l'exactitude de nos valeurs.



Illustration 7: Écran LCD

2.1.5. La batterie

Pour que le robot soit autonome, nous avons gardé la batterie déjà installée sur le robot. Celle-ci était imposée par le concours de Vierzon. C'est une batterie de 9,6 V pour 1,2Ah. Elle a l'avantage d'être assez puissante pour fournir l'alimentation nécessaire à la carte électronique et aux moteurs.

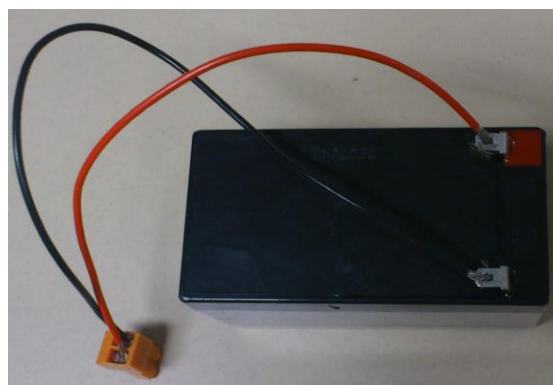
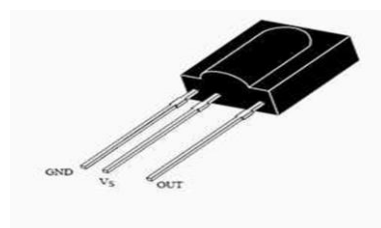


Illustration 8: Notre batterie

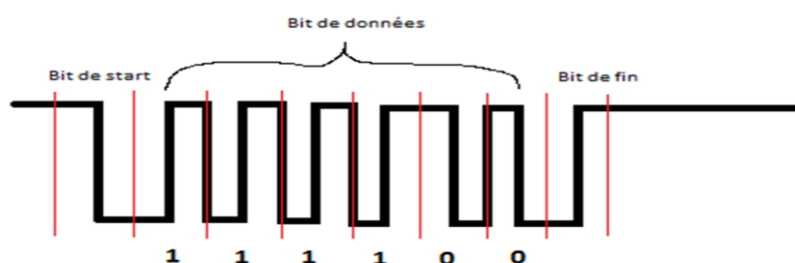
Pour recharger la batterie, il nous suffira d'avoir une alimentation réglée sur 13,5 V et de limiter le courant à 1 A.

2.1.6. Le capteur infrarouge

Le capteur infrarouge est un pilier de notre système. C'est grâce à lui que nous pourrons récupérer et transmettre les instructions de la télécommande au microcontrôleur. Lors de l'appuie sur une touche de la télécommande, celle-ci envoie une trame en infrarouge, le récepteur infrarouge reçoit cette trame et la transmet au microcontrôleur qui la traitera pour reconnaître sur quelle touche nous avons appuyé.



Voici la forme de la trame :



La trame envoyée par la télécommande se compose de différentes parties.

Elle a : _ un bit de start

_ des bits de données (contient l'adresse et les données)

_ un ou deux bits de fin

Cette trame est appelée RC5.

2.1.7. Télécommande Phillips et Code RC5

Nous avons donc une télécommande de télévision phillips qui est codé en RC5.

Le code RC5 a été mis au point chez Philips, il est devenu une norme pour les transmissions de commandes en infrarouge.

Le code RC5 se compose d'une suite de 14 bits:

⑩ 2 bits de départ

⑩ 1 bit de basculement

⑩ 5bits d'adressage

⑩ 6 bits d'instruction

Les bits du code RC5 sont codés en biphase appelé code Manchester c'est à-dire qu'un bit est composé de 2 demi-bits alternés. La combinaison bas/haut caractérise un bit positionné à 1 et la combinaison haut/bas un bit remis à 0. Il a une période de 1,778ms.

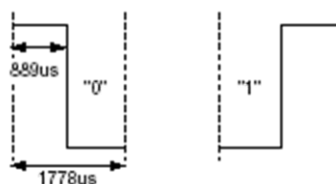


Illustration 11: Code Manchester

2.2. Visualisation des composants utilisés sur la carte

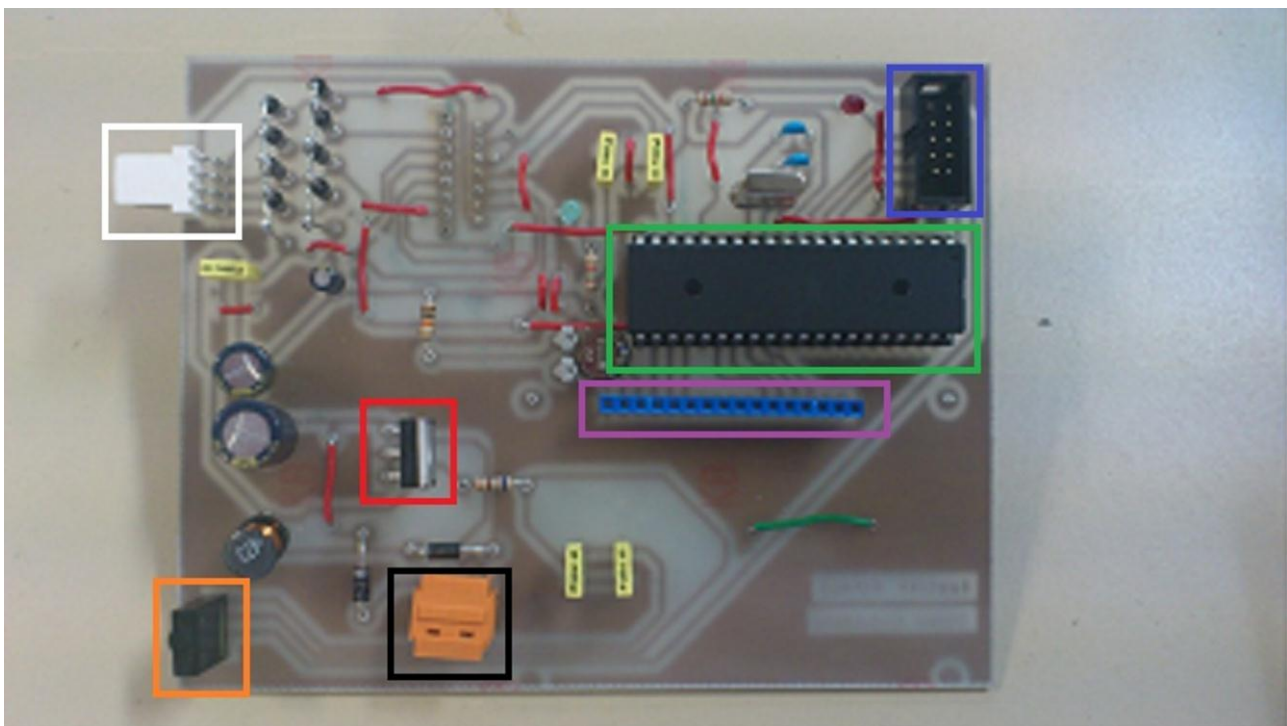


Illustration 12: Carte électronique

Voici le placement des différents composants de notre carte électronique.

Numérotation	Composants	Couleur
1	ATMEGA8535	Vert
2	TSOP1638	Orange
3	Connecteur LCD	Violet
4	LM2575	Rouge
6	Connecteur de programmation	Bleu
7	Alimentation +12V (Batterie)	Noir
8	Connecteur moteur	Blanc

3. La programmation

3.1. La réception de la trame RC5

La réception de la trame est la partie la plus délicate de la programmation. En effet, si on ne réussit pas à récupérer la trame correctement, on ne pourra pas la traiter plus tard pour pouvoir donner les bonnes instructions au robot.

Pour effectuer cette tâche, nous avons d'abord pensé à envoyer la trame dans la liaison série, mais les types de trame (longueur et contenu) étant différents, nous ne pouvions pas emprunter ce chemin.

Nous avons donc pensé à faire une interruption qui détecterait le bit de start (un front descendant). Cette interruption déclenche un timer qui va permettre d'avoir une interruption à chaque bit se qui nous permettra de récupérer tous les bits de la trame un à un. Une fois la trame entière mise dans un tableau dentier, il nous suffit juste de récupérer les données dont on a besoin.

Fonction d'interruption (détection du front)

```
interrupt [EXT_INT0] void Inter (void)
{
    TCNT2 = 0;           // mise à zéro du compteur, début de comptage
    GICR &= 0b1011111; // interdiction du flag d'interruption externe
}
```

Dans cette fonction d'interruption qui détecte le front descendant du début de la trame, on met à zéro le compteur pour qu'il puisse commencer à compter de zéro. On interdit aussi le flag

d'interruption externe (donc de détection du front descendant du début de la trame) pour éviter que l'on remette le compteur à zéro avant la fin du premier comptage.

Cette fonction se déclenche dès que le compteur a fini de compter. Ensuite, on compte le nombre de bits qu'il y a dans la trame et on récupère les bits un à un pour les placer dans un tableau d'entier. Une fois tous les bits de la trame récupérée, on met un drapeau (la variable Flag) à 1 pour repérer dans le programme principal le moment où l'on pourra traiter la trame.

Récupération de la donnée de la trame

```

if (Flag)
{
    Fonction d'interruption du compteur
    for (j = 0, j < 10, j++)
    {
        Data[j] = tram[j+14]; // récupération de la data de la
    }
}
interrupt [TIM0_COMP] void Top (void)
{
    tram[i] = Trame;
    Flag = 0; // On met le niveau logique de la trame dans un tableau d'entier
    if (i == Nb_bit)
    {
        GICR |= 0b0100000; // autorisation du flag d'interruption externe
    }
    Donnee = tram[0]*1000000000+tram[1]*100000000+tram[2]*10000000+tram[3]*1000000+
    tram[4]*100000+tram[5]*10000+tram[6]*1000+tram[7]*100+tram[8]*10+tram[9];
    Flag = 1; // mise a zéro du compteur de nombre de bits
}
else
{
    // mise du table tram en une seul variable
    {
        i++; //incrémentation du compteur de nombre de bits
    }
}
}
    
```

Ici, On récupère la donnée de la trame pour la mettre dans un tableau d'entier, puis on met la donnée sous un seul variable pour pouvoir l'intégrer dans un switch.

3.2. Traitement des données de la trame

Le traitement des données n'est pas une partie compliquée, elle est juste longue, car c'est dans celle-ci que l'on va dire ce que fait le robot en fonction de quelle touche (donc quelle donnée) on aura reçu. On peut aussi intégrer du MLI si on veut que le robot n'avance pas à sa vitesse

maximale. Ou même faire une correction de défaut mécanique,. Si le robot ne va pas droit quand on fait tourner les deux moteurs à la même vitesse, on peut faire tourner un moteur plus vite que l'autre

Traitement de la donnée de la trame

switch (Donnee)

```
{  
case Haut :    IN1 = 1;  
               IN2 = 0;  
               IN3 = 0;  
               IN4 = 1;  
               delay_ms (8);  
               IN1 = 0;  
               IN2 = 0;  
               IN3 = 0;  
               IN4 = 0;  
               delay_ms ( 2);  
               break;  
  
case Bas :    IN1 = 0;  
              IN2 = 1;  
              IN3 = 1;  
              IN4 = 0;  
              delay_ms (8);  
              IN1 = 0;  
              IN2 = 0;  
              IN3 = 0;  
              IN4 = 0;  
              delay_ms ( 2);  
              break;
```

```
case Gauche : IN1 = 1;  
              IN2 = 0;  
              IN3 = 0;  
              IN4 = 0;  
              delay_ms (2);  
              IN1 = 0;  
              IN2 = 0;  
              IN3 = 0;  
              IN4 = 0;  
              Donnee = 0;  
              delay_ms (8);  
              break;  
  
case Droite : IN1 = 0;  
              IN2 = 0;  
              IN3 = 0;  
              IN4 = 1;  
              delay_ms (2);  
              IN1 = 0;  
              IN2 = 0;  
              IN3 = 0;  
              IN4 = 0;  
              delay_ms (8);  
              Donnee = 0;  
              break;  
}
```

pour corriger cette déviation.

En fonction de la valeur qu'il y a dans la variable « Donnée », on commandera le robot en mettant pendant un court moment les variables pour, par exemple, avancer à 1 puis à zéro. C'est le principe de la MLI (Modulation de Largeur d'Impulsion).

4. Mécanique

La partie mécanique est la partie qui nous a pris le moins de temps, car le châssis comprenant roues, moteurs et batterie 12V nous a été fournis par l'IUT.

4.1. Le châssis

Le châssis est en plastique de couleur blanche, de forme rectangulaire aux bouts arrondies, les deux roues en plastiques sont placées au centre du robot, de chaque côté. Sur le dessus du châssis est placée la carte électronique, un interrupteur On/Off ainsi que la batterie et enfin un connecteur qui vient du moteur pour le connecter à la carte électronique. Puis sur le dessous nous avons les deux moteurs.

4.2. Le bloc moteur

Les moteurs peuvent fonctionner en 12V au maximum de leur puissance, de plus nous avons remarqué que les moteurs avaient besoin de 1A en charge pour fonctionner. Nous avons donc adapté notre carte électronique pour avoir le courant désiré. Pour avoir une vitesse maximale nous avons

appliqué une tension de 12V et pour arrêter les moteurs, une tension de 0V.

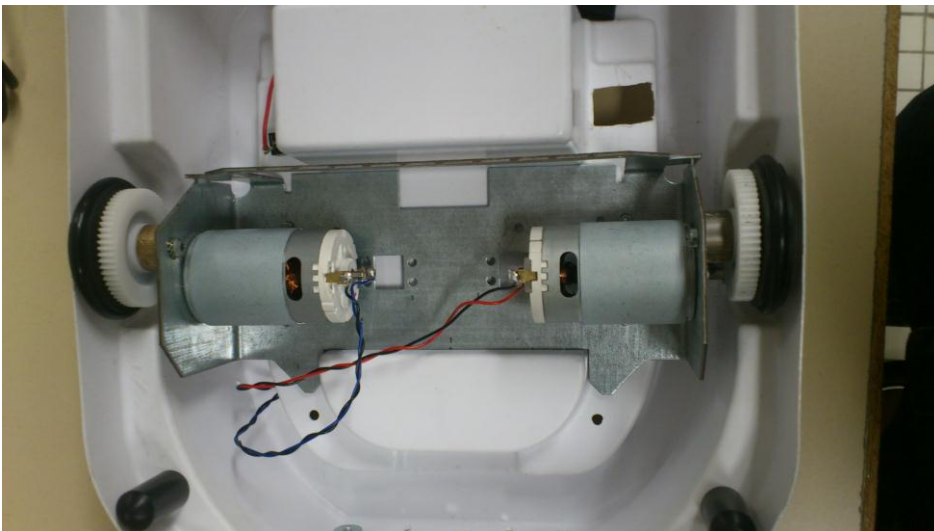


Illustration 13: Bloc moteur

Les principales caractéristiques :

- Tension d'alimentation nominale : 12V
- Courant d'alimentation maximal : 1 A

Voici une petite illustration du fonctionnement du bloc moteur d'après un schéma repris du projet

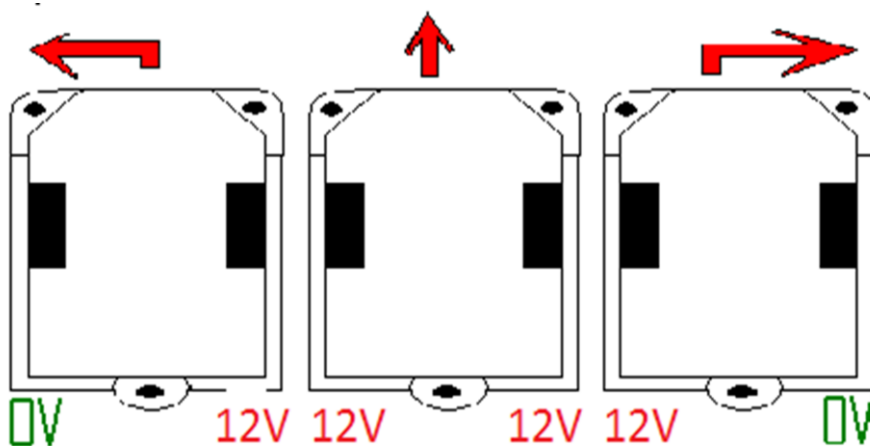


Illustration 14: fonctionnement moteurs

de Vidal et Lallemand que nous avons un peu modifié.

Nous avons donc 4 principaux cas, comme aller tout droit ou reculer à pleine puissance, tourner rapidement à droite et à gauche. Nous pourrions bien sûr adapter la vitesse, mais notre projet étant principalement la réception infrarouge nous ne ferons que ces 4 cas là.

5. Tests

Tout au long de nos séances nous avons été amenés à faire plusieurs batteries de tests afin de vérifier le bon fonctionnement de notre projet.

5.1. Sur la carte électronique

La carte électronique est celle qui nous a pris le plus de temps, en effet nous avons eu plusieurs petits soucis que nous avons été amenés à régler.

5.1.1. Vérifications des tensions

Notre premier test sur la carte a été de vérifier à l'aide d'un oscilloscope et d'un multimètre si nous avons les bonnes tensions sur différents points de notre carte.

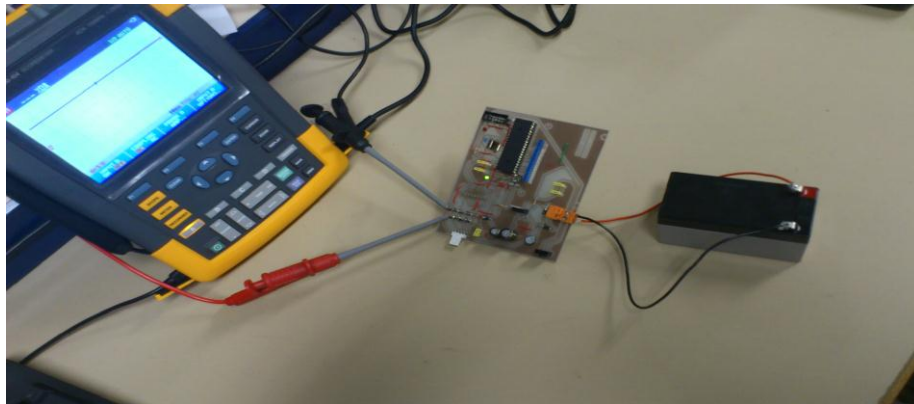
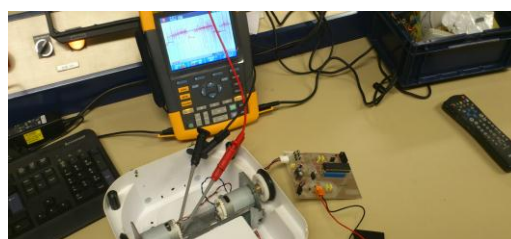


Illustration 15: Test tensions

Nous avons donc mesuré du 12V sur l'entrée de notre régulateur LM2575 et du 5V à sa sortie, 5V sur l'alimentation de l'ATMEGA8535, 12V sur notre L298N et 5V sur le TSOP. La LED qui nous indique si la carte est alimentée, s'allumait donc, pour l'instant, nous avons eu aucun problème.

5.1.2. Sorties du pont en H

Pour les tester nous avons programmé sur Code Vision AVR de quoi faire tourner le moteur et donc comme nous l'avons exposé dans la partie mécanique du 12V/12V pour le faire avancer. A vide nous avons bien un potentiel de 12V entre chaque pattes de nos sorties du pont.



Mais en charge nous nous sommes aperçus que nous n'avions plus les tensions escomptées après plusieurs visualisations à l'oscilloscope, et en relisant la datasheet du L298N nous avons remarqué que nous avons oublié de brancher les 2 pattes de mesure de courant qui se relie tout simplement à la masse et le problème fut résolu le moteur tourné comme nous le voulions.

5.2. Visualisation des trames de la télécommande TV

C'est ce que nous avons fait dès le début de notre projet afin de vérifier s'il était possible de recevoir un signal que nous pourrions, par la suite, utiliser.

Nous avons visualisé le signal à l'aide du TSOP et d'un oscilloscope.

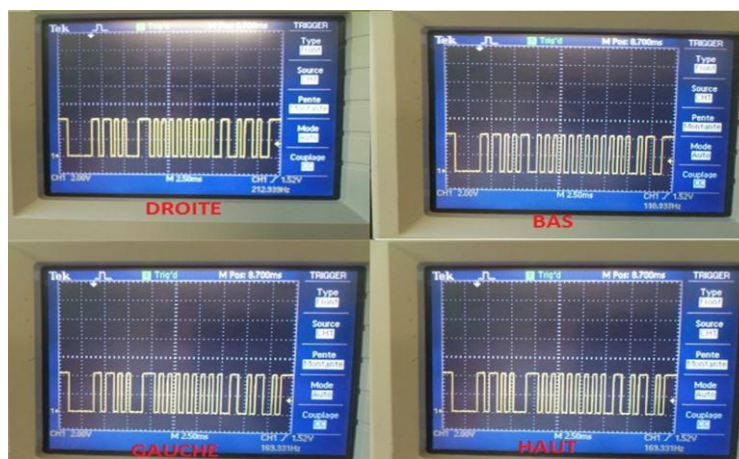


Illustration 17: Visualisation des trames

5.3. Mécanique

Après avoir testé le bon fonctionnement des moteurs nous avons remarqué un dysfonctionnement au niveau des roues du robot : une roue se serrait contre le châssis quand elle tournait ce qui posait un réel problème. Pour y remédier nous avons alors démonté la roue pour étudier le problème.

Après l'avoir démonté on a remarqué que la partie fixe de la roue n'était pas assez serrée sur le châssis en plastique ce qui a resserré le tout, et bloqué la roue. Pour y remédier nous avons

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle



resserré le tout et ajouté une rondelle de frein. Résultat nous avons quand même une roue qui tourne plus vite que l'autre.

Conclusion

La réalisation de ce projet nous a aussi permis de travailler en autonomie, respecter un cahier des charges et les contraintes de temps.

Grâce à celui-ci nous avons pu utiliser des notions de 1^{ère} année et de 2^{ème} de l'IUT GEII. En effet nous avons eu besoin de notions d'électronique, d'informatique et de mécanique de première année, mais nous avons quand même eu quelques recherches à faire pour connaître le fonctionnement de certaines choses comme l'infrarouge. Ce qui nous a permis d'améliorer des bases dans certains domaines mais aussi de découvrir l'utilisation de logiciel comme Orcad et de micro-contrôleur tel que l'ATMEGA8535.

Le fonctionnement de notre carte électronique est valide, même si nous avons eu quelques soucis avec le pont en H. La partie informatique nous a posé problème à cause de la trame RC5 de la télécommande.

Malgré ces problèmes ce fût une expérience enrichissante qui a été pour nous un certain reflet du monde du travail, nous avons donc eu une certaine expérience professionnelle à travers ce projet.

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle



Résumé

Notre projet a été de réaliser un robot télécommandé par infra-rouge, nous avons pour cela dû créer notre propre carte électronique. Nous avons repris le châssis et le moteur du robot sur un projet déjà existant. Pour télécommander ce robot nous avons dû étudier la trame infrarouge RC5, le pont en H pour l'alimentation du moteur ainsi que certains composants utiles à la fabrication de notre carte. Nous avons ensuite fait certains tests pour vérifier le bon fonctionnement de notre moteur, quelques problèmes sont survenus, mais ont été vite résolus. Suite à cela nous avons beaucoup réfléchi au traitement de la trame de la télécommande pour la traiter dans notre programme, nous avons donc fait un programme que nous n'avons pas encore testé, car nous n'avons pas accès à notre salle, mais nous pourrions le tester lors de notre dernière séance.

Index des illustrations

Illustration 1: ATMEGA8535.....	7
Illustration 2: Implantation des pattes de l'atmega8535.....	7
Illustration 3: L298N.....	8
Illustration 4: Fonctionnement du L298N.....	8
Illustration 5: pont en H montage diode.....	9
Illustration 6: LM2575	10
Illustration 7: Écran LCD.....	10
Illustration 8: Notre batterie	11
Illustration 9: TSOP1738	11
Illustration 10: trame de la télécommande	11
Illustration 11: Code Manchester	12
Illustration 12: Carte électronique.....	13
Illustration 13: Bloc moteur	17
Illustration 14: fonctionnement moteurs	18
Illustration 15: Test tensions	19
Illustration 16: Test du pont en H.....	19
Illustration 17: Visualisation des trames	20

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

UNIVERSITE FRANCOIS-RABELAIS
TOURS



Institut Universitaire de Technologie

Département
GENIE ELECTRIQUE ET
INFORMATIQUE INDUSTRIELLE

Bibliographie

Code RC5 : <<http://www.positron-libre.com/electronique/protocole/code-rc5/code-rc5.php>>

Documentation constructeur :

⑩ ATMEGA8535 : <<http://www.thierry-lequeu.fr/data/AT-MEGA-8535L.pdf>>

⑩ LM2575 : <http://www.datasheetcatalog.com/datasheets_pdf/L/M/2/5/LM2575.shtml>

⑩ L298N : <http://www.datasheetcatalog.com/datasheets_pdf/L/2/9/8/L298N.shtml>

⑩ Universalis : <<http://www.universalisedu.com.portail.scd.univtours.fr/encyclopedie/infrarouge/>>

>

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

UNIVERSITE FRANCOIS-RABELAIS
TOURS



Institut Universitaire de Technologie

Département
GENIE ELECTRIQUE ET
INFORMATIQUE INDUSTRIELLE

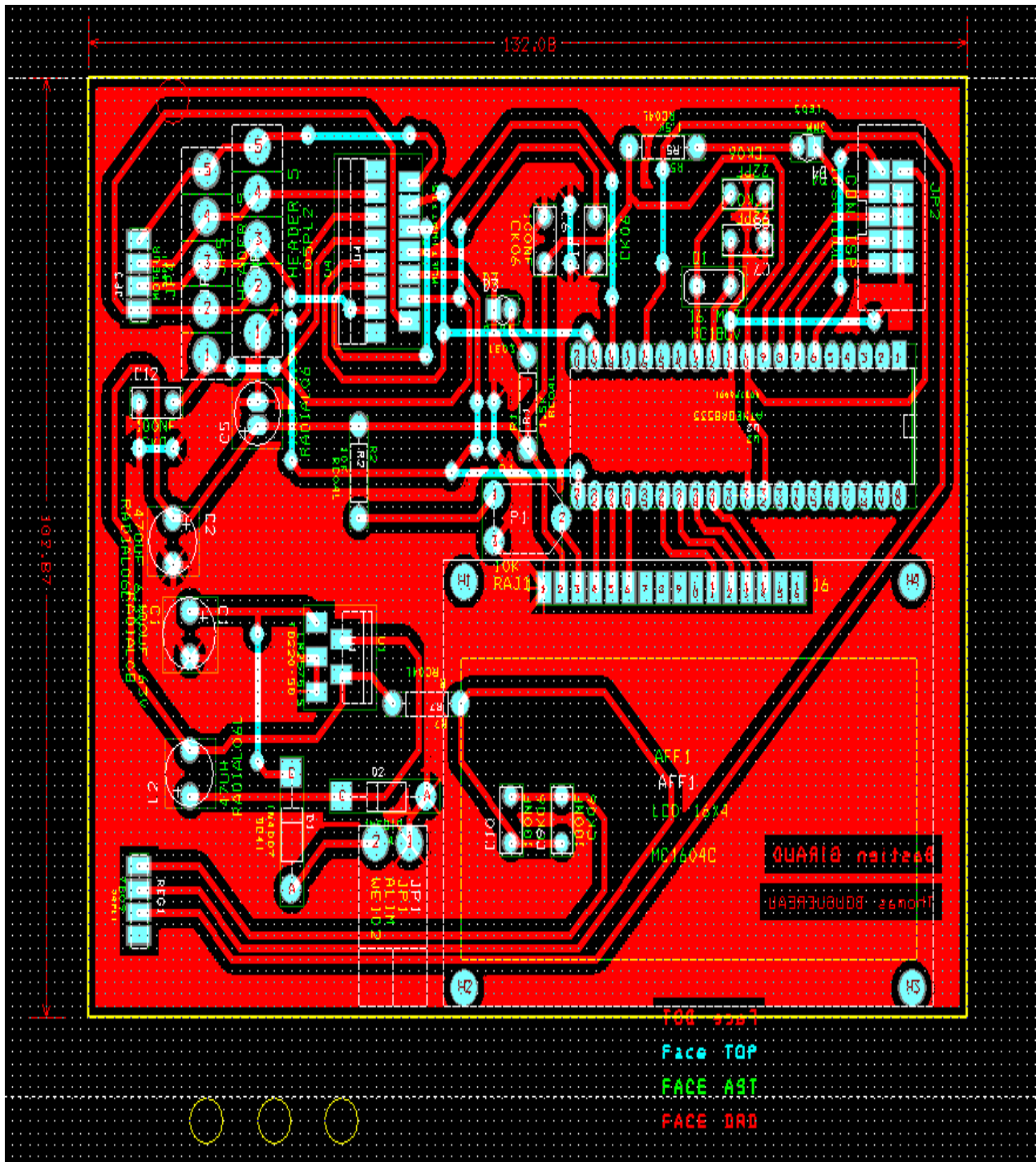
Annexes

Annexe 1 : Typon de la carte électronique

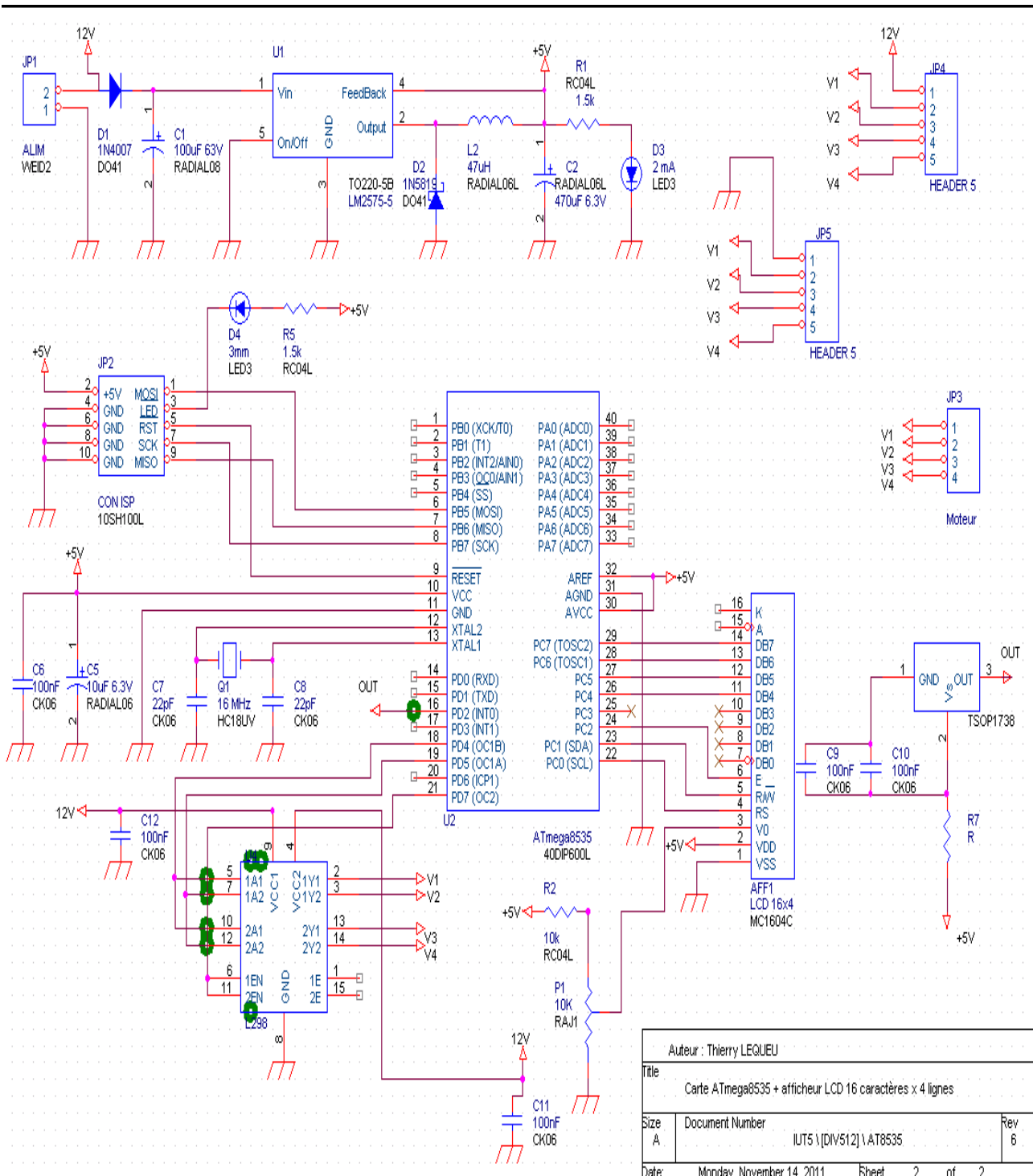
Annexe 2 : Schéma électrique du typon

Annexe 3 : Listes des composants du typon

Annexe 1 : typon de la carte électronique



Annexe 2 : Schéma électrique du typon



Auteur : Thierry LEQUEU			
Titre Carte ATmega8535 + afficheur LCD 16 caractères x 4 lignes			
Size A	Document Number IUTS \ [DIV512] \ AT8535	Rev 6	
Date: Monday, November 14, 2011	Sheet 2	of 2	

Annexe 3 : Listes des composants du typon

Ref Des	Enabled	Footprint Name	Package Name	Comp Rotation	Location X, Y	Flags
AFF1	Yes	MC1604C_AFF1	CON16_1	0	53.34,81.28	No
C1	Yes	RADIAL08_C1	CAPA_POL	270	0.00,78.74	No
C2	Yes	RADIAL06L_C2	CAPA_POL	270	-2.54,88.90	No
C5	Yes	RADIAL06_C5	CAPA_POL	90	10.16,99.06	No
C6	Yes	CK06_C6	CAP_NP	270	53.34,116.84	No
C7	Yes	CK06_C7	CAP_NP	180	81.28,119.38	No
C8	Yes	CK06_C8	CAP_NP	180	81.28,124.46	No
C9	Yes	CK06_C9	100NF	90	55.88,58.42	No
C10	Yes	CK06_C10	100NF	90	48.26,58.42	No
C11	Yes	CK06_C11	100NF	90	60.96,121.92	No
C12	Yes	CK06_C12	100NF	0	-2.54,101.60	No
D1	Yes	DO41_D1	1N4007	270	15.24,60.96	No
D2	Yes	DO41_D2	DIODE_SCHOTTKY	0	22.86,58.42	No
D3	Yes	LED3_D3	LED_RONDE	0	48.26,111.76	No
D4	Yes	LED3_D4	LED_RONDE	180	91.44,129.54	No
JP1	Yes	WEID2_JP1	HEADER_2	270	33.02,53.34	No
JP2	Yes	10SH100L_JP2	CON_ISP	270	104.14,127.00	No
JP3	Yes	04PL1_JP3	4_HEADER	90	-7.62,111.76	No
JP4	Yes	05PL2_JP4	HEADER_5	90	2.54,106.68	No
JP5	Yes	05PL2_JP5	HEADER_5	90	10.16,109.22	No
L2	Yes	RADIAL06L_L2	INDUCTOR	90	0.00,58.42	No
P1	Yes	RAJ1_P1	RESISTOR_TAPPED	0	45.72,91.44	No
Q1	Yes	HC18UV_Q1	CRYSTAL	0	81.28,114.30	No
R1	Yes	RC04L_R1	R	90	50.80,96.52	No
R2	Yes	RC04L_R2	R	270	25.40,99.06	No
R5	Yes	RC04L_R5	R	180	76.20,129.54	No
R7	Yes	RC04L_R7	R	180	40.64,68.58	No
REG	Yes	04PL1_REG1	7809_0	270	-7.62,50.80	No
U1	Yes	T0220-5B_U1	LM2575T_1	270	25.40,76.20	No
U2	Yes	40DIP600L_U2	6801U4NM_3	180	106.68,106.68	No
U4	Yes	MULTIWATT15_U4	L298_1	90	27.94,109.22	No