

Affichage de la vitesse d'une auto-laveuse et de la tension des batteries



**Rapport de projet tuteuré
2^{ème} année
Études & Réalisations**

Affichage de la vitesse d'une auto-laveuse et de la tension des batteries

Rapport de projet tuteuré 2^{ème} Année

Études & Réalisations

Sommaire

Introduction.....	4
1. Cahier des charges et spécifications.....	5
1.1. Coût de production.....	5
1.2. Alimentation du système.....	5
1.3. Matériel à disposition.....	5
1.4. Mécanique.....	6
2. Plannings.....	7
3. Analyse technique du projet.....	8
3.1. Analyse fonctionnelle.....	8
3.2. Étude de la solution.....	9
4. Conception de la carte.....	25
4.1. Réalisation.....	25
4.2. Montage de la carte.....	27
4.3. Tests et essais.....	28
Conclusion.....	30
Résumé.....	31
Table des illustrations.....	32
Table des tableaux.....	33
Bibliographie.....	34
Tables des annexes.....	35

Introduction

Durant la seconde année de notre formation, nous sommes amenés à réaliser un projet tuteuré dans le cadre des séances d'études et réalisations. Le thème que nous avons choisi porte sur la réalisation de mesures et sur l'affichage de la vitesse et de la tension des batteries d'une auto-laveuse (Nilfisk BA 410). [1]

Une auto-laveuse est une machine électrique conçue pour un entretien rapide des sols. Elle est capable de laver les sols et de les sécher. Les roues de l'auto-laveuse ne sont pas motorisées, il est nécessaire de pousser l'appareil pour le faire avancer. L'objectif du projet sera donc de relever la vitesse de l'appareil, sa température interne et la tension des batteries qui l'alimentent. Toutes ces informations ainsi que l'état de charge des batteries seront affichés. En ce sens, l'utilisateur pourra connaître l'état de la machine pour une efficacité optimale et également prévoir de recharger ou de changer les batteries du système.

Dans cet écrit, nous présenterons dans un premier temps le cahier des charges du projet puis aborderons l'analyse technique du projet et enfin sa réalisation. A travers cette étude, des tests et essais seront effectués de manière à valider la conformité du dispositif conçu.

1. Cahier des charges et spécifications

Pour mieux définir les spécifications de base du projet à réaliser, il nous a fallu rédiger un cahier des charges. Il s'agit d'une étape importante, un référentiel à prendre en compte et à satisfaire pendant toute la réalisation du projet. Afin de répondre efficacement au cahier des charges, voici les fonctionnalités que le système devra être capable de réaliser :

- x mesure et affichage de la vitesse d'avancement
- x mesure et affichage de la température de l'appareil
- x mesure et affichage de la tension des batteries alimentant le système
- x évaluation de l'état de charge des batteries

1.1. Coût de production

Comme pour tout projet, le système présente des contraintes de réalisation. L'une des premières contraintes est une contrainte financière : le projet doit être accompli avec un budget restreint, celui de l'IUT, d'environ 100 €.

1.2. Alimentation du système

Les autres contraintes seront principalement techniques. Effectivement, l'auto-laveuse est alimentée par deux batteries de 12 V montées en série, délivrant ainsi une tension totale de 24 V. L'alimentation sera présente sur deux borniers prévus sur le système.

1.3. Matériel à disposition

Pour réaliser le projet, nous aurons à notre disposition un micro-contrôleur de type ATmega8535. C'est pourquoi il sera nécessaire d'utiliser un ordinateur équipé d'un logiciel de programmation. L'ATmega8535 sera effectivement programmé avec le logiciel *CodeVision AVR* disponible en salle de projet via la liaison série (configuration disponible en annexe 3).

D'autre part, les mesures de vitesse seront effectuées par le biais d'un capteur ILS¹. Quant à la mesure de température, un capteur de température sera utilisée.

Les différentes informations relevées seront affichées grâce à un afficheur LCD avec un affichage sur 4 lignes et 16 caractères par ligne (réf. Farnell 944-9019).

Afin de réaliser les tests durant la phase de programmation, une carte d'essai composée d'un micro-contrôleur et d'un afficheur sera mise à notre disposition. Une maquette avec un moteur, un capteur ILS et deux aimants placés sur l'axe de rotation du moteur permettront de simuler la mesure de vitesse de l'auto-laveuse et de réaliser les tests.

¹ Capteur ILS : capteur à lame souples. Il s'agit d'un détecteur de proximité à commande magnétique associé à un aimant (cf. principe de fonctionnement – page 11).

1.4. Mécanique

Nous verrons par la suite que nous aurons des contraintes mécaniques pour placer le capteur ILS sur le système de l'auto-laveuse dans le sens où l'emplacement pour ce dispositif n'est pas initialement prévu.

De plus, la carte électronique devra être implantée dans un boîtier déjà existant qui sera, par la suite, fixé sur l'auto-laveuse. Le boîtier mis à notre disposition (réf. PC 100/35 LG) a les côtes indiquées sur l'illustration 1.

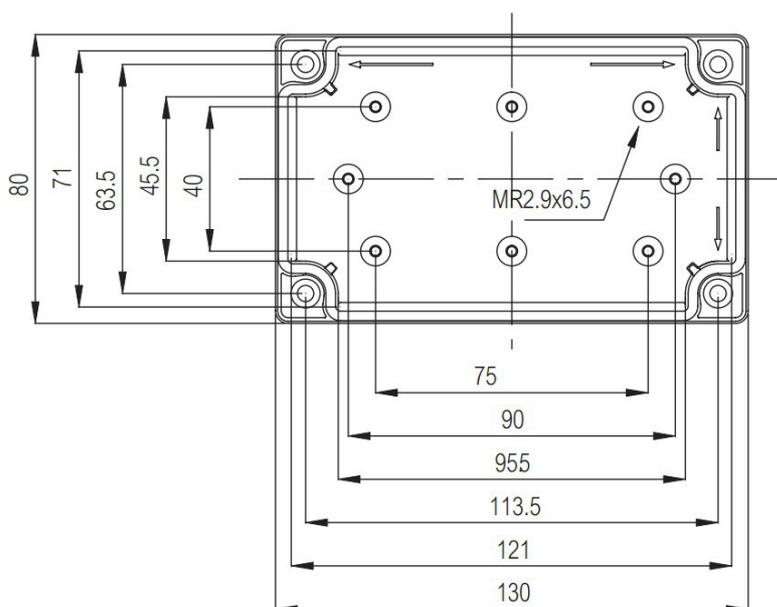


Illustration 1: Dimensions du boîtier PC 100/35 LG [2]

La carte devra donc avoir les dimensions suivantes : 120 x 70 mm.

L'auto-laveuse est normalement prévue pour fonctionner dans un environnement protégé, en intérieur.



Illustration 2: Auto-laveuse [1]

2. Plannings

La durée de réalisation du projet étant limitée à 7 semaines, il était nécessaire d'établir avant tout un planning prévisionnel regroupant les différentes étapes de réalisation.

Pour comparer et repérer les divers problèmes liés à la réalisation du projet, nous avons également établi un planning réel (cf. légende ci-après).

Semaine	05	06	07	08	09	10	11	12	13	14
Recherche documentaire et découverte du sujet	Planning prévisionnel Planning réel				Périodes de vacances	Périodes de vacances				
Identification et élaboration d'une solution :										
♦ Mesure/affichage de la vitesse d'avancement	Planning prévisionnel Planning réel	Planning prévisionnel			Périodes de vacances	Périodes de vacances				
♦ Mesure/affichage de la tension des batteries			Planning prévisionnel Planning réel	Planning réel	Périodes de vacances	Périodes de vacances				
♦ Mesure/affichage de la température du système				Planning prévisionnel	Planning réel	Planning réel				
Saisie schématique et réalisation du typon de la carte					Périodes de vacances	Périodes de vacances	Planning prévisionnel Planning réel			
Réalisation de la carte					Périodes de vacances	Périodes de vacances		Planning prévisionnel Planning réel	Planning prévisionnel	
Essais du système et montage des éléments sur l'auto-laveuse					Périodes de vacances	Périodes de vacances		Planning réel	Planning prévisionnel Planning réel	
Finitions et rédaction du projet					Périodes de vacances	Périodes de vacances			Planning prévisionnel Planning réel	
Passage à l'oral					Périodes de vacances	Périodes de vacances				Planning prévisionnel Planning réel

 Planning prévisionnel
  Planning réel
  Périodes de vacances

Tableau 1: Planning prévisionnel et réel de réalisation du projet [3]

Les tests et validations du programme ont été réalisées au fur-et-à-mesure de la programmation de manière à vérifier les résultats. Lors de la réalisation du projet, différents problèmes sont intervenus. Tout d'abord, un problème de rebonds a été remarqué lors de la mesure de vitesse du moteur sur la maquette (problème traité page 11). Les vacances scolaires nous ont permis de réfléchir à l'élaboration du programme informatique et de réaliser les algorigrammes. Des contraintes mécaniques liées au montage du système sur l'auto-laveuse ont également retardé la finalisation du projet.

3. Analyse technique du projet

3.1. Analyse fonctionnelle

Le principe de notre projet est de relever des grandeurs physiques, de les traiter et d'établir en perspective l'affichage d'informations utiles à l'opérateur (vitesse de l'auto-laveuse, température, état de charge des batteries) sur un écran visible par ce dernier. C'est ce qu'illustre le schéma fonctionnel suivant.

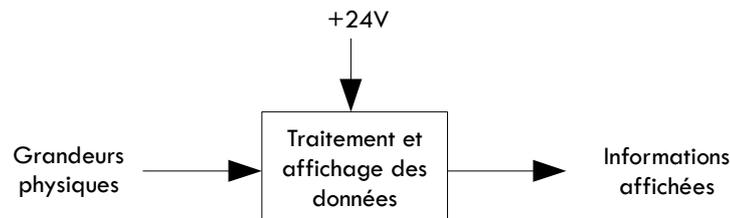


Illustration 3: Schéma fonctionnel de niveau 1 du système [3]

Les informations utiles à l'opérateur ne seront pas directement mesurées. Effectivement, il faudra dans un premier temps acquérir des grandeurs physiques puis calculer à partir de celles-ci les données utiles et enfin les afficher. Chaque fonction est représentée par un bloc fonctionnel.

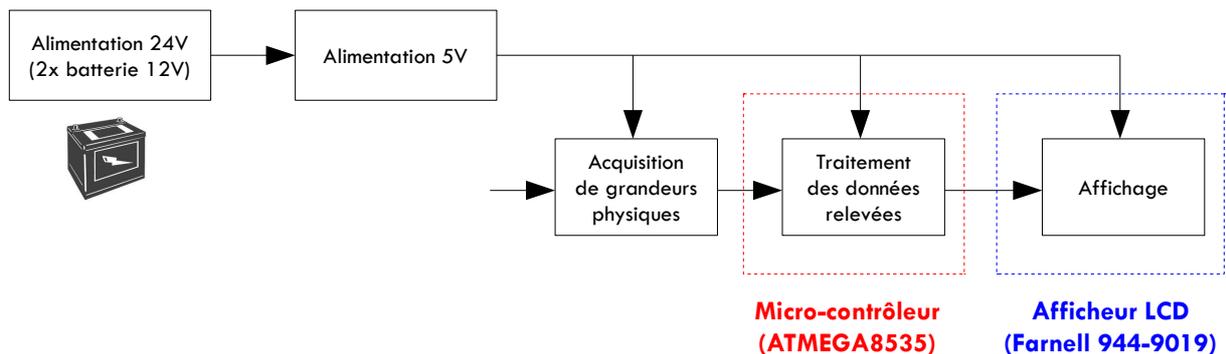


Illustration 4: Schéma fonctionnel de niveau 2 du système [3]

3.2. Étude de la solution

Le micro-contrôleur sera en fait l'élément central qui va permettre de traiter toutes les informations désirées.

Selon la documentation du constructeur (cf. annexe 2), le brochage de l'ATmega8535 est le suivant :

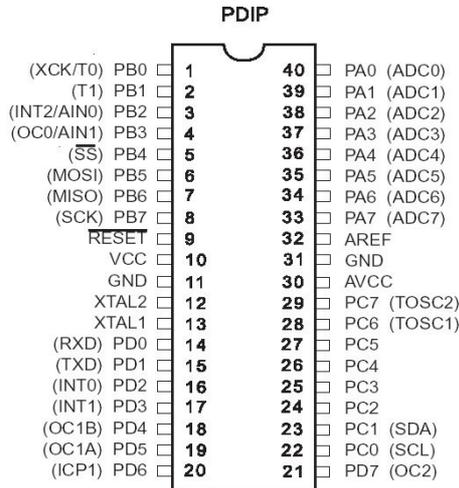


Illustration 5: Brochage de l'ATmega 8635 [4]

Il comporte effectivement 4 ports avec au total 32 entrées/sorties programmables :

- × Port A (PA0 à PA7) : entrées analogiques (Convertisseur Analogique Numérique interne au micro-contrôleur)
- × Ports B (PB0 à PB7), Port C (PC0 à PC7) et Port D (PD0 à PD7) : entrées/sorties numériques

Une partie de ces entrées/sorties sera ainsi utilisée pour acquérir les informations relatives à l'auto-laveuse.

De plus, entre les broches 12 (XTAL1) et 13 (XTAL2), un quartz de 16 MHz et deux condensateurs ont été placés de manière à cadencer le micro-contrôleur.

Pour une meilleure compréhension de la solution adoptée, nous nous intéresserons successivement à l'étude de chacune des fonctionnalités énoncées dans le cahier des charges.

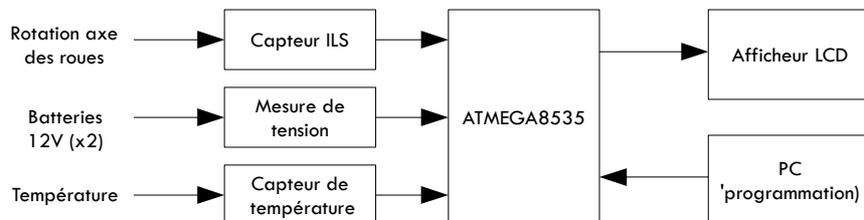


Illustration 6: Fonctions à réaliser [3]

Le schéma structurel de la carte réalisée est présent en illustration 24 et le programme informatique en annexe 4.

3.2.1. Alimentation de la carte

L'utilisation de l'ATmega8535 nécessite une alimentation +5V. La tension fournie par les batteries étant de 24V, nous devons ramener cette tension à 5V, c'est pourquoi nous avons utilisé un régulateur de tension, il s'agit d'une solution peu onéreuse et très fiable. Le rôle d'un régulateur est effectivement de rendre quasiment continue une tension si elle présente une ondulation et de stabiliser sa valeur. Il existe plusieurs familles de régulateurs de tension. Notre choix s'est porté sur un régulateur à tension fixe et positive capable de délivrer un courant maximum de 1 Ampère ; un régulateur LM2575 a donc été utilisé. A partir de la tension des batteries, il fournira en sortie une tension constante +5V.

Le constructeur propose la structure suivante pour une alimentation régulée +5V :

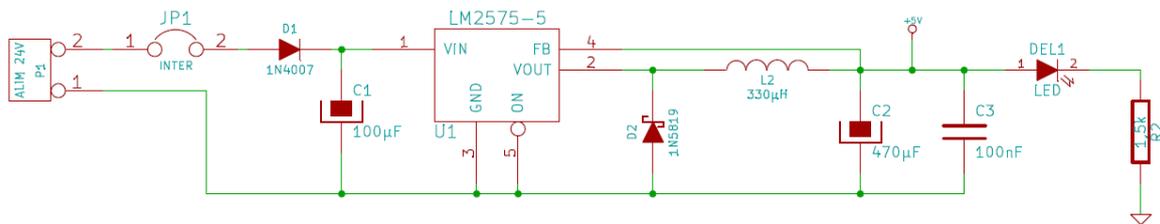


Illustration 7: Schéma d'alimentation de la carte avec régulateur [5]

Le régulateur comporte cinq broches : l'entrée, la masse, la sortie, le choix du mode et le feedback. (cf. illustration 8).

Dans le cas d'une utilisation normale, la broche « ON/OFF » doit être mise au niveau bas. Quant à la broche « FeedBack », elle permet de mesurer continuellement l'écart entre la consigne et la tension mesurée en vue d'annuler cet écart (principe de régulation).

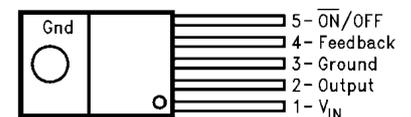


Illustration 8: Brochage du régulateur LM2575 [5]

Le condensateur C_1 est un condensateur de filtrage. Il a pour but de réduire les éventuelles ondulations présentes en entrée et ainsi de rendre plus stable la tension pour un fonctionnement optimal du régulateur.

Le condensateur C_2 joue le rôle de « réservoir d'énergie » pour la carte puisqu'elle profite de l'alimentation régulée (le condensateur va pallier aux appels de courant importants).

L'inductance L_2 va permettre de « lisser » le courant en sortie du régulateur pour réduire les variations les plus rapides de manière à obtenir une valeur approximativement continue.

La diode placée en sortie (D_2) assure une protection contre les surcharges dues à la décharge du condensateur de sortie C_2 .

Les valeurs de ces composants sont préconisées par le constructeur.

En entrée du régulateur, nous avons ajouté une diode (D_1) de manière à éviter toute destruction de composants dans le cas où la polarité des batteries n'aurait pas été respectée. Dans ce cas, la diode ne conduit pas, le reste du montage n'est donc pas alimenté, il s'agit d'une protection. Une diode classique de type 1N4007 sera suffisante puisque nous avons effectivement besoin d'une diode capable de protéger le régulateur de tensions négatives à son entrée et pouvant supporter des courants jusqu'à 1 A (courant maximum délivré par le régulateur).

De plus, le condensateur C_3 a été ajouté de manière à éviter que des pics de consommations dus à l'ATmega8535 se transmettent dans toutes les pistes d'alimentation de la carte. Sur la carte, ce condensateur de découplage sera donc placé au plus près du micro-processeur. Pour supprimer ces parasites, un condensateur d'une valeur de 100nF est suffisant.

Enfin, une LED témoin permettra à l'utilisateur de voir si la carte est alimentée. La résistance R_2 servira à limiter le courant qui traverse la diode.

3.2.2. Mesure et affichage de la vitesse du système

Pour connaître la vitesse d'avancement de l'auto-laveuse, nous placerons plusieurs aimants (réf. P6250000) sur les roues (côté intérieur) ainsi qu'un capteur ILS (Celduc PLA137) sur la carrosserie de l'auto-laveuse. Ainsi, nous aurons une partie mobile (aimants) et une partie fixe (capteur). Voici ci-contre la modélisation de ces deux éléments.



Illustration 9: Aimant P6250000 [6]

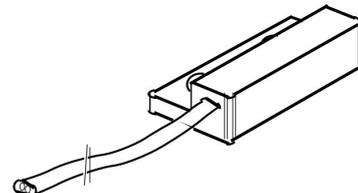


Illustration 10: Capteur PLA 137 [6]

Le capteur ILS (ou détecteur de proximité à commande magnétique) capte un champ magnétique. A chaque fois qu'un aimant se trouvera devant le capteur, le contact de ce dernier changera d'état.

Le principe de fonctionnement est illustré ci-après.

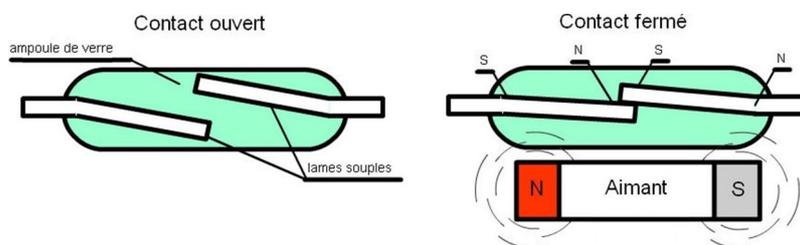


Illustration 11: Principe de fonctionnement du capteur ILS [7]

Le contact est normalement ouvert (les lames ne sont pas en contact). A l'approche d'un aimant, les lames souples se polarisent et s'attirent, le contact se ferme et laisse passer le courant.

Ayant mis en place quatre aimants, l'axe des roues aura fait un tour lorsque les quatre aimants seront passés devant le capteur. Lorsque l'opérateur poussera l'auto-laveuse, le capteur produira des impulsions (0 – 5V) et formera ainsi un signal carré avec un rapport cyclique valant normalement 0,5 (à condition que les aimants soient placés à distances égales). La mesure est d'autant plus précise que le nombre d'aimants mis en place est élevé.

La solution pour mesurer la vitesse serait d'utiliser une interruption externe (l'ATmega 8535 dispose de trois entrées pour interruptions externes). A chaque changement d'état, le micro-contrôleur déclenchera automatiquement l'interruption programmée.

Les premiers essais ont révélé un problème lié à l'utilisation du capteur ILS puisque ce dernier produisait des rebonds non-négligeables du point de vue du micro-contrôleur. Les rebonds présents sur son entrée sont représentées ci-dessous et faussaient les mesures.

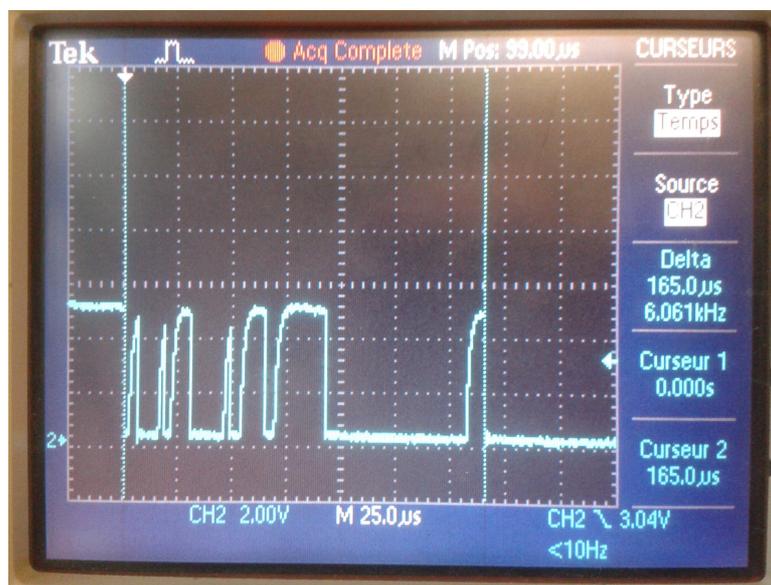


Illustration 12: Rebonds produits par le capteur ILS (entre les deux curseurs) [3]

Effectivement, un capteur ILS n'est pas adapté à une mesure de vitesse, il est plutôt sollicité pour détecter des fins de course de vérin. Afin de résoudre ce problème, il convient d'ajouter un filtre de type RC (composé d'une résistance et d'un condensateur) comme l'indique le schéma ci-contre.

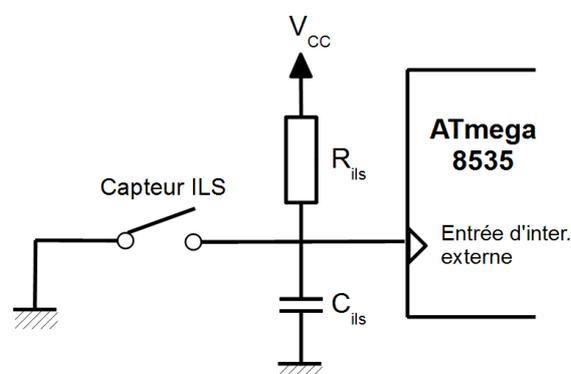


Illustration 13: Solution anti-rebonds pour le capteur ILS [3]

La résistance jouera le rôle de résistance de pull-up, elle permet d'éviter de laisser l'entrée d'interruption externe en l'air lorsque le contact du capteur ILS est ouvert. Dans ce cas, la tension appliquée sur l'entrée est « tirée vers le haut », soit ramenée à +5V. Le condensateur permettra ainsi de réaliser le filtrage.

Le calcul de la vitesse se fait avec avec la relation bien connue : $v = \frac{d}{t}$ avec d la distance parcourue par le véhicule et t le temps mis pour la parcourir. Si on considère maintenant p le diamètre de la roue du véhicule, alors son périmètre vaut $p = \pi \times d$. A chaque tour de roue, l'auto-laveuse avance donc d'une longueur p . On a finalement la relation suivante : $v = \frac{p}{t}$. Il s'agit ainsi du calcul utilisé dans la programmation. Le diamètre des roues de l'auto-laveuse est de 25 cm. [1] Nous mesurerons ainsi le temps que met la roue pour faire un tour, autrement dit le temps qu'auront mis les quatre aimants à passer devant le capteur. Les ordinogrammes illustrent effectivement la solution retenue pour l'aspect programmation.

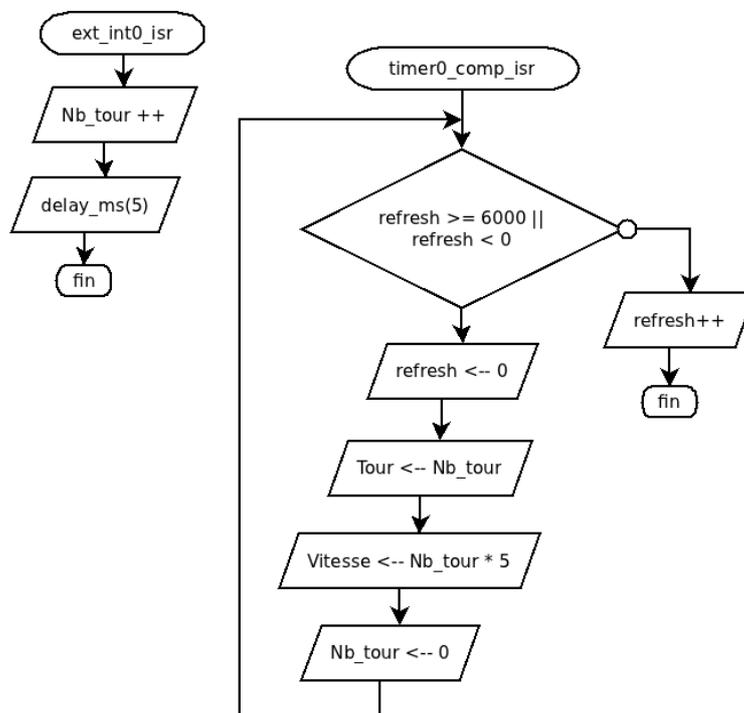


Illustration 14: Ordinogramme - Mesure de la vitesse [3]

L'ordinogramme figurant à gauche correspond à l'interruption externe déclenchée à chaque fois qu'un aimant se présente devant le capteur. Celui-ci incrémente une variable et attend 5ms afin d'éviter les rebonds du capteur ILS. L'ordinogramme de droite représente la fonction du timer qui va permettre de calculer la vitesse de l'auto-laveuse selon le nombre de tours effectués par la roue, à intervalle de temps régulier (rafraîchissement de la vitesse). Le programme est visible en annexe 4.

Afin de valider les mesures établies, un tachymètre a été utilisé pendant la phase de test sur la maquette. L'affichage de la vitesse est géré dans le *main*² (visible page 19).

² *Main* : terme anglais désignant la fonction principale d'un programme

3.2.3. Mesure et affichage de la tension des batteries

Intéressons-nous à présent à la mesure de tension des batteries. Il s'agit de deux batteries 12V de capacité 75Ah (réf. OPTIMA-S,55), elles assurent l'alimentation de l'auto-laveuse. Les batteries sont connectées via deux connecteurs deux voies (masse et 12V). Le câblage des batteries est illustré ci-après.

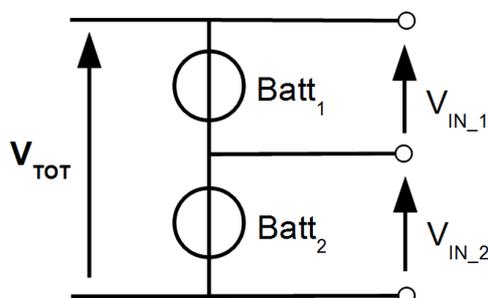


Illustration 15: Câblage des deux batteries sur l'auto-laveuse [3]

La tension des deux batteries devra être mesurée de façon à vérifier leur charge et être certain de l'équilibre des tensions sur les deux batteries. En effet, si l'une des deux batteries a une tension plus élevée que l'autre, il est nécessaire de changer l'une des deux batteries ou les deux. Selon la documentation du constructeur de la batterie (annexe 1), cette dernière est considérée comme chargée à 13,1V et comme déchargée à 10,6V. Lorsqu'elle est en charge, sa tension évoluera de 13,65V à 15V.

La solution pour mesurer la tension de chaque batterie est d'utiliser le convertisseur analogique-numérique du micro-contrôleur. Celui-ci possède effectivement huit entrées analogiques (ADC0 à ADC7). N'ayant pas de masse entre les deux batteries, nous devons mesurer la tension des deux batteries en série et la tension de la deuxième batterie ; deux entrées analogiques seront donc utilisées.

Sur ces entrées analogiques, une tension de 5V maximum est applicable. L'utilisation de deux ponts diviseurs de tension sera donc nécessaire pour réduire la tension des batteries, comme l'illustre le schéma suivant :

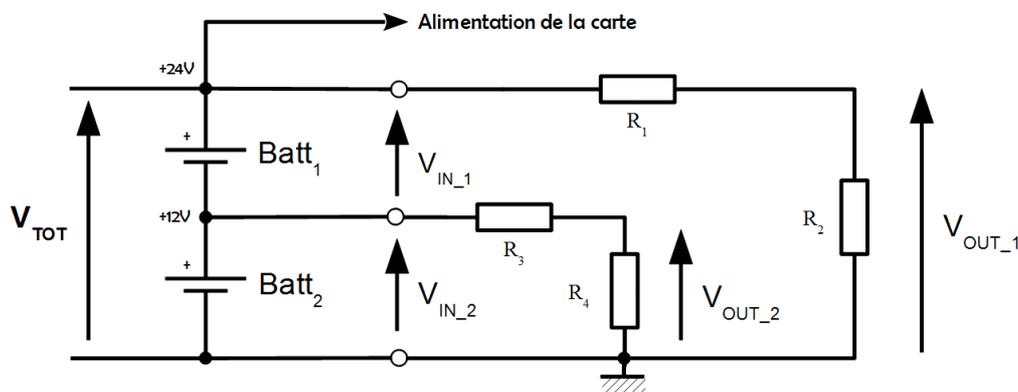


Illustration 16: Schéma électrique pour la mesure de tension des batteries [3]

La tension de sortie du premier pont diviseur de tension s'exprime de la façon suivante :

$$V_{OUT_1} = V_{TOT} \times \frac{R_2}{R_1 + R_2}$$

On peut ainsi obtenir le rapport entre les deux résistances :

$$\begin{aligned} V_{OUT_1} &= V_{TOT} \times \frac{R_2}{R_1 + R_2} \\ \Leftrightarrow \frac{V_{OUT_1}}{V_{TOT}} &= \frac{R_2}{R_1 + R_2} \\ \Leftrightarrow \frac{V_{TOT}}{V_{OUT_1}} &= 1 + \frac{R_1}{R_2} \\ \Leftrightarrow \frac{R_1}{R_2} &= \frac{V_{TOT}}{V_{OUT_1}} - 1 \\ \Leftrightarrow R_1 &= \left(\frac{V_{TOT}}{V_{OUT_1}} - 1 \right) \times R_2 \end{aligned}$$

L'ATmega8535 dispose également une entrée de référence fixée à 2,56V par le constructeur. Nous déterminerons ainsi les résistances pour lesquelles, à la tension d'entrée moyenne, nous avons la tension de référence (2,56V) en sortie.

Nous fixerons $R_2 = R_4 = 1 \text{ k}\Omega$.

Ce premier pont diviseur de tension (R_1 , R_2) sera utilisé pour mesurer la tension aux bornes des deux batteries. La tension moyenne d'entrée des deux batteries en série est alors la suivante :

$$\overline{V_{TOT}} = 2 \times \overline{V_{BATT}} = 2 \times 11,85 = 23,7 \text{ V} \quad (\text{cf. paragraphe 3.2.4})$$

$$\text{On obtient : } R_1 = \left(\frac{23,7}{2,56} - 1 \right) \times 1000 = 8258 \Omega$$

* * * *

De la même manière, le deuxième pont diviseur de tension (R_3 , R_4) est défini par la relation suivante :

$$V_{OUT_2} = V_{TOT} \times \frac{R_4}{R_3 + R_4} \quad \Leftrightarrow \quad R_3 = \left(\frac{V_{IN_2}}{V_{OUT_2}} - 1 \right) \times R_4$$

Ce deuxième pont diviseur de tension sera utilisé pour mesurer seulement la tension aux bornes de la deuxième batterie. La tension moyenne d'une batterie est alors :

$$\overline{V_{IN_2}} = \overline{V_{BATT}} = 11,85 \text{ V} \quad (\text{cf. paragraphe 3.2.4})$$

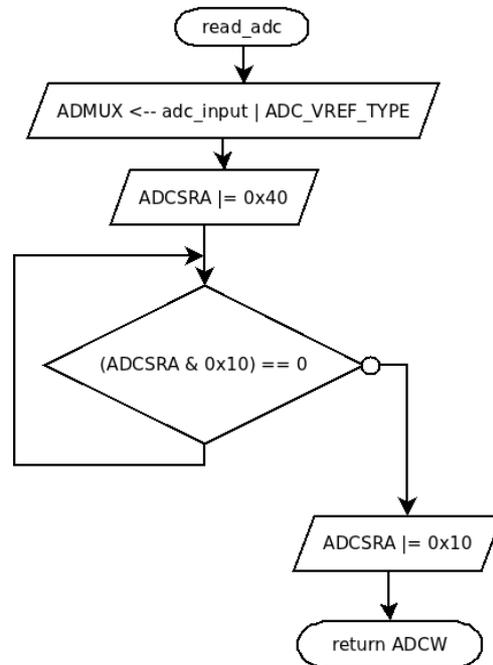
$$\text{On obtient } R_3 = \left(\frac{11,85}{2,56} - 1 \right) \times 1000 = 3629 \Omega$$

* * * *

Nous avons choisi les valeurs normalisées $R_1=8,2 k\Omega$, $R_3=3,6 k\Omega$ et $R_2=R_4=1 k\Omega$.

La tension de la première batterie sera calculée en effectuant la différence des deux valeurs mesurées puisque d'après la loi des mailles : $V_{IN_1}=V_{TOT}-V_{IN_2}$ (cf. schéma de câblage des deux batteries – illustration 16).

L'ordinogramme permettant la mesure de tension des batteries est le suivant :



*Illustration 17: Ordinogramme -
Mesure de la tension des batteries [3]*

La fonction `read_adc` permettra de configurer et de lancer la conversion analogique-numérique. Cette fonction sera elle aussi appelée dans la fonction principale (*main* - cf. page 19).

3.2.4. Évaluation de l'état de charge des batteries

Afficher la tension des batteries n'est pas la solution la plus efficace pour l'utilisateur, il ne connaît pas nécessairement la tension que doivent avoir des batteries au minimum. Il sera en effet plus adapté d'afficher le pourcentage du niveau de charge de chacune des deux batteries.

Il s'agit de batteries ayant une tension nominale de 12V. A la tension minimale, la batterie est considérée comme déchargée. Les différents niveaux de tensions sont donnés par le constructeur (cf. annexe 1). Les tensions minimales et maximales seront respectivement notés par la suite V_{MIN} et V_{MAX} . Nous aurons ainsi $V_{MIN}=10,5V$ et $V_{MAX}=13,1V$.

Pour réaliser le calcul en pourcentage du niveau de charge, la courbe suivante a été tracée.

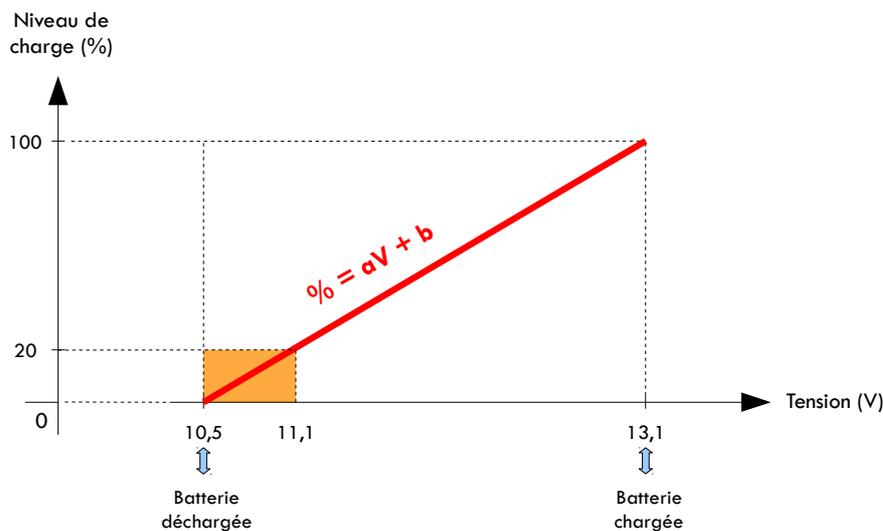


Illustration 18: Pourcentage du niveau de charge en fonction de la tension de la batterie [3]

L'équation de la droite proportionnelle entre la tension et le niveau de charge peut alors être établie :

$$a = \frac{\Delta \%}{\Delta V} = \frac{100 - 0}{13,1 - 10,5} = \frac{100}{2,6} \approx 38,45 \quad \Rightarrow \quad \% = 38,5V + b$$

Pour $\% = 0$, on a $V = 10,6$

$$\Leftrightarrow 38,5V + b = 0$$

$$\Leftrightarrow b = -38,5V = -38,5 \times 10,6 = -408,1$$

L'équation de la droite est donc la suivante : $\% = 38,5V - 408,1$

Il s'agit donc du calcul figurant dans le programme pour le micro-contrôleur (cf. programme en annexe 4) permettant de donner le pourcentage du niveau de charge de chacune des deux batteries.

Lorsque le niveau de charge sera inférieur à 20 % (modélisé en orange sur l'illustration 18), un message signalera à l'utilisateur de mettre en charge les batteries.

Pendant la charge de la batterie (tension supérieure à 13,65V), le pourcentage ne sera plus affiché, mais l'utilisateur pourra voir l'instruction « En charge ». Lorsque la charge sera terminée (tension d'environ 15V), l'instruction « Batterie chargée » s'affichera.

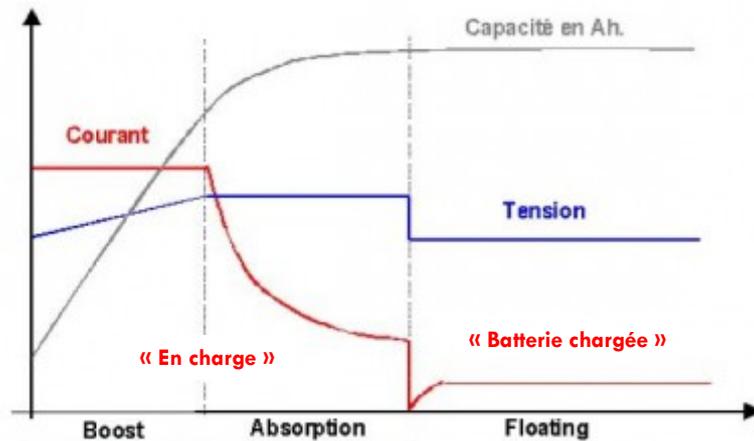


Illustration 19: Courbe de charge des batteries à 3 étapes [8]

La charge des batteries s'effectuera en 3 étapes :

- × Etape 1 : « boost » ou « pulse » - la batterie est chargée rapidement jusqu'à environ 80 %
- × Etape 2 : « absorption » ou « égalisation » - la batterie est chargée de 80 à 100 %
- × Etape 3 : « floating » ou « entretien » - la charge de la batterie est maintenue à 100 %.

L'état de la batterie est entièrement géré dans la fonction principale (*main*). Il s'agit de la fonction qui va permettre de gérer intégralement le programme. Voici donc page suivante l'ordinogramme de cette fonction.

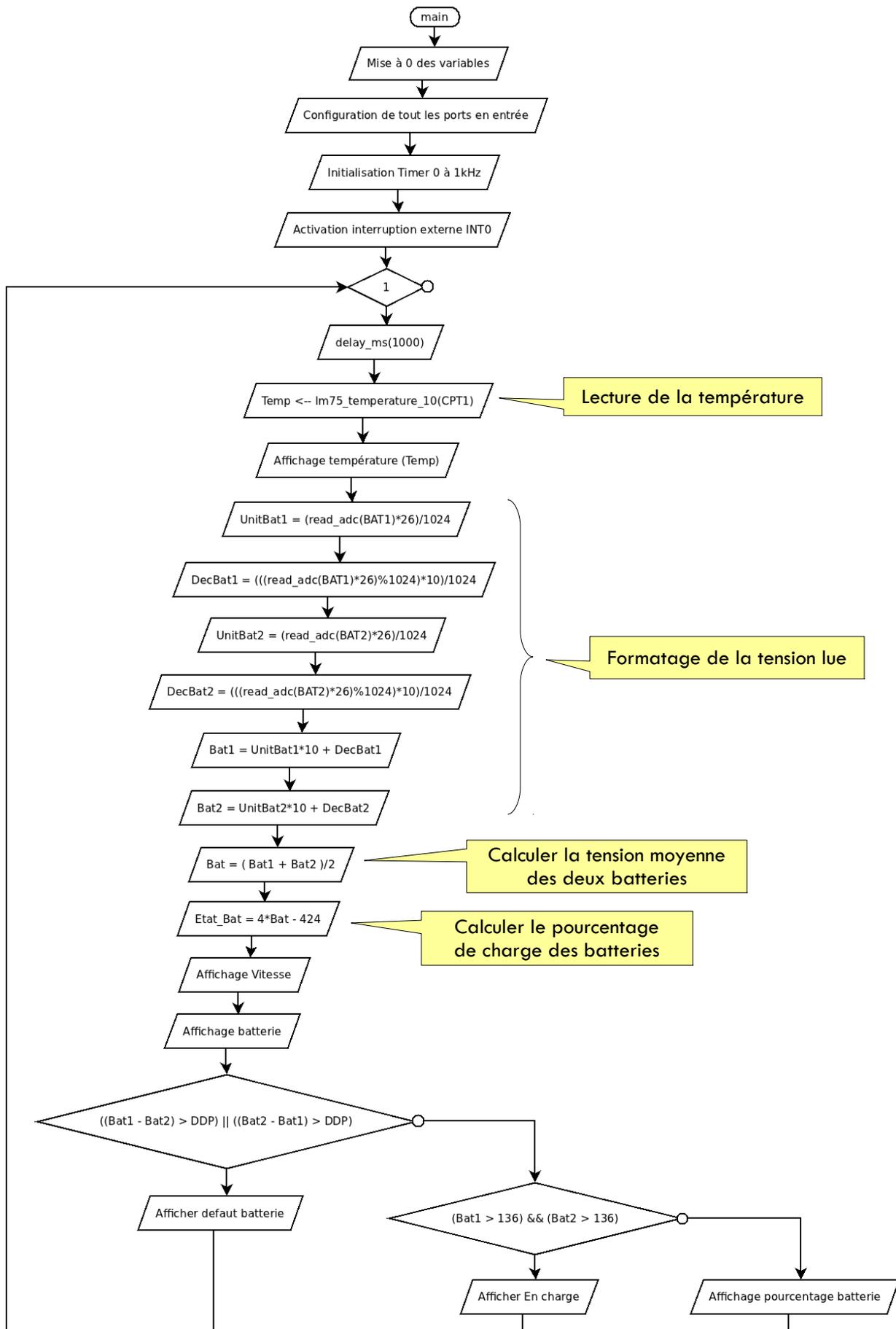


Illustration 20: Ordigramme - Fonction principale du programme [3]

3.2.5. Mesure et affichage de la température du système

Dans le but de mesurer la température du système, nous utiliserons le capteur LM75. Il s'agit d'un composant qui retourne une valeur numérique de la température ambiante sans l'aide de composants externes. Sous le carter de l'auto-laveuse, ce capteur permettra de vérifier que la température interne n'augmente pas dangereusement. Le capteur peut relever des températures allant de -55°C à $+125^{\circ}\text{C}$.

Si la température devient critique (température supérieure à 40°), un message avertira l'utilisateur en vue d'arrêter le système manuellement.

Le LM75 est programmable via une liaison série de type I²C. Les seuils de l'hystérésis et les paramètres peuvent ainsi être configurés par programmation. Le capteur possède aussi une interruption programmable qui se déclenche lorsqu'un seuil de l'hystérésis est dépassé. Dans le projet, une interface déjà existante a été utilisée puis implantée dans la machine. Il s'agit d'une carte contenant un LM75 (schéma page 21). Néanmoins, il a donc fallu prévoir un connecteur pour brancher cette interface sur notre carte.

Le capteur LM75 possède 2 broches d'alimentation, 2 broches pour le bus I²C, 3 broches d'adresse et une sortie de détection.

Le bus I²C est composé de :

- x une ligne de données (SDA)
- x un signal d'horloge (SCL)

La sortie O.S. est une sortie de détection utilisée lorsque la température dépasse le seuil programmé via le bus I²C.

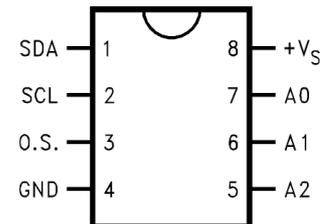


Illustration 21:

Brochage du LM75 [9]

Dans notre cas, le capteur n'utilisera qu'un seul fil de données (SDA).

Sur un même bus I²C, plusieurs composants peuvent être branchés. Chaque composant dispose d'une adresse pour que l'information aille vers le bon composant. Les bits A0 à A2 servent ainsi à configurer l'adressage du composant I²C.

Le signal d'horloge (SCL) sera généré par l'ATmega8535 pour relever la température à la période définie.

La trame de données (SDA) est représentée sur 9 bits en complément à 2 (D7-15). Le bit de poids faible (LSB) de la trame vaut $0,5^{\circ}$:

(Read Only):

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
MSB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	LSB	X	X	X	X	X	X	X

D0–D6: Undefined

D7–D15: Temperature Data. One LSB = 0.5°C . Two's complement format.

Illustration 22: Registre de température (I²C) [9]

Voici donc ci-contre quelques correspondances entre la température et la valeur de l'octet que retourne le capteur (en binaire et en hexadécimal) :

Temperature	Digital Output	
	Binary	Hex
+125°C	0 1111 1010	0FAh
+25°C	0 0011 0010	032h
+0.5°C	0 0000 0001	001h
0°C	0 0000 0000	000h
-0.5°C	1 1111 1111	1FFh
-25°C	1 1100 1110	1CEh
-55°C	1 1001 0010	192h

Tableau 2: Équivalent des températures en binaire et hexadécimal [9]

L'interface qui sera utilisée et placée sous le capot de l'auto-laveuse a le schéma électrique suivant :

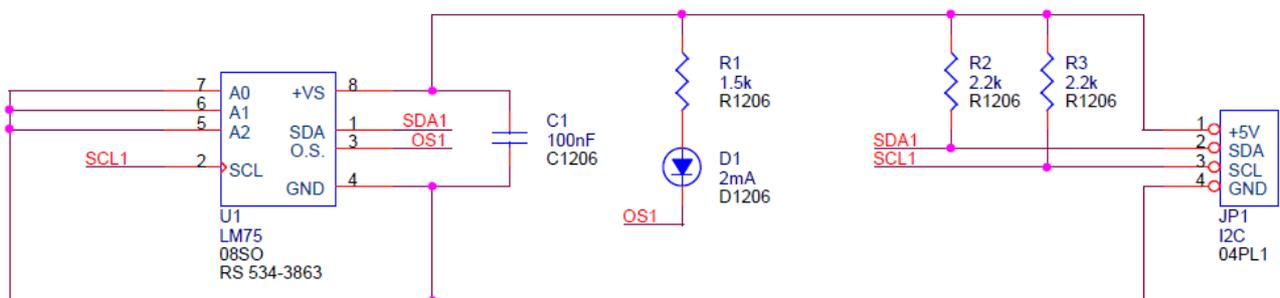


Illustration 23: Schéma électrique de l'interface I²C [10]

La température est relevée à partir de la fonction principale (cf. ordinogramme page 19).

Le programme est disponible en annexe 4.

Référence	Valeur	Désignation/Technologie	Fabrication (boîtier)	Prix
Résistances :				
R1	10 kΩ	Carbone 1/4 W - 5%	RC05	0,03 €
R2	1,5 kΩ	Carbone 1/4 W - 5%	RC05	0,03 €
R3	1 kΩ	Carbone 1/4 W - 5%	RC05	0,03 €
R4	1 kΩ	Carbone 1/4 W - 5%	RC05	0,03 €
R5	8,2 kΩ	Carbone 1/4 W - 5%	RC05	0,03 €
R6	3,6 kΩ	Carbone 1/4 W - 5%	RC05	0,03 €
R7	1,5 kΩ	Carbone 1/4 W - 5%	RC05	0,03 €
Rils	2,2 kΩ	Carbone 1/4 W - 5%	RC05	0,03 €
Résistance variable :				
RV1	10 kΩ	Piste cermet	Ajustable horizontal	0,25 €
Condensateurs :				
C1	100 μF / 64 V	Électrochimique polarisé	Radial	0,65 €
C2	470 μF /	Électrochimique polarisé	Radial	0,20 €
C3	100 nF	Plastique		0,15 €
C4	100 nF	Plastique		0,15 €
CQ1	22 pF	Plastique		0,15 €
CQ2	22 pF	Plastique		0,15 €
Cils	100 nF	Plastique		0,15 €
Inductance :				
L2	470 μH			0,70 €
Circuits intégrés :				
U1	LM2575-5	Régulateur +5V	TO 220	6,10 €
AT1	ATmega8535	Micro-contrôleur	DIP 40	7,50 €
Diodes :				
D1	1N4007		DO 35	0,09 €
D2	1N5819		DO 35	0,20 €
D3		LED jaune	∅ 3 mm ronde	0,20 €
DEL1		LED verte	∅ 5 mm ronde	0,10 €
Divers :				
JP1		Interrupteur à bascule		0,75 €
		Cosses à sertir		1,50 €
		Porte-fusible		1,00 €
		Fusible		1,50 €
		Support CI	Support tulipe	0,36 €
		Support afficheur	Support tulipe	0,90 €

Référence	Valeur	Désignation/Technologie	Fabrication (boîtier)	Prix
Divers :				
P2		Afficheur LCD	16 caractères x 4 lignes	26,94 €
		Connecteur SUB-D (mâle et femelle)	9 broches	3,00 €
		Connecteurs batterie		1,60 €
X1		Quartz 16 MHz	HC49S	1,00 €
		Capteur ILS		0,75 €
		4 aimants		0,80 €
		Boîtier avec couvercle	PC 100/35 LG	26,36 €
TOTAL :				85,50 €

Tableau 3: Nomenclature et coût de fabrication de la carte³ [3]

3 Les prix des composants ont été indiqués à titre indicatif et proviennent du site « Elecdiv », site de vente en ligne de matériel électronique.

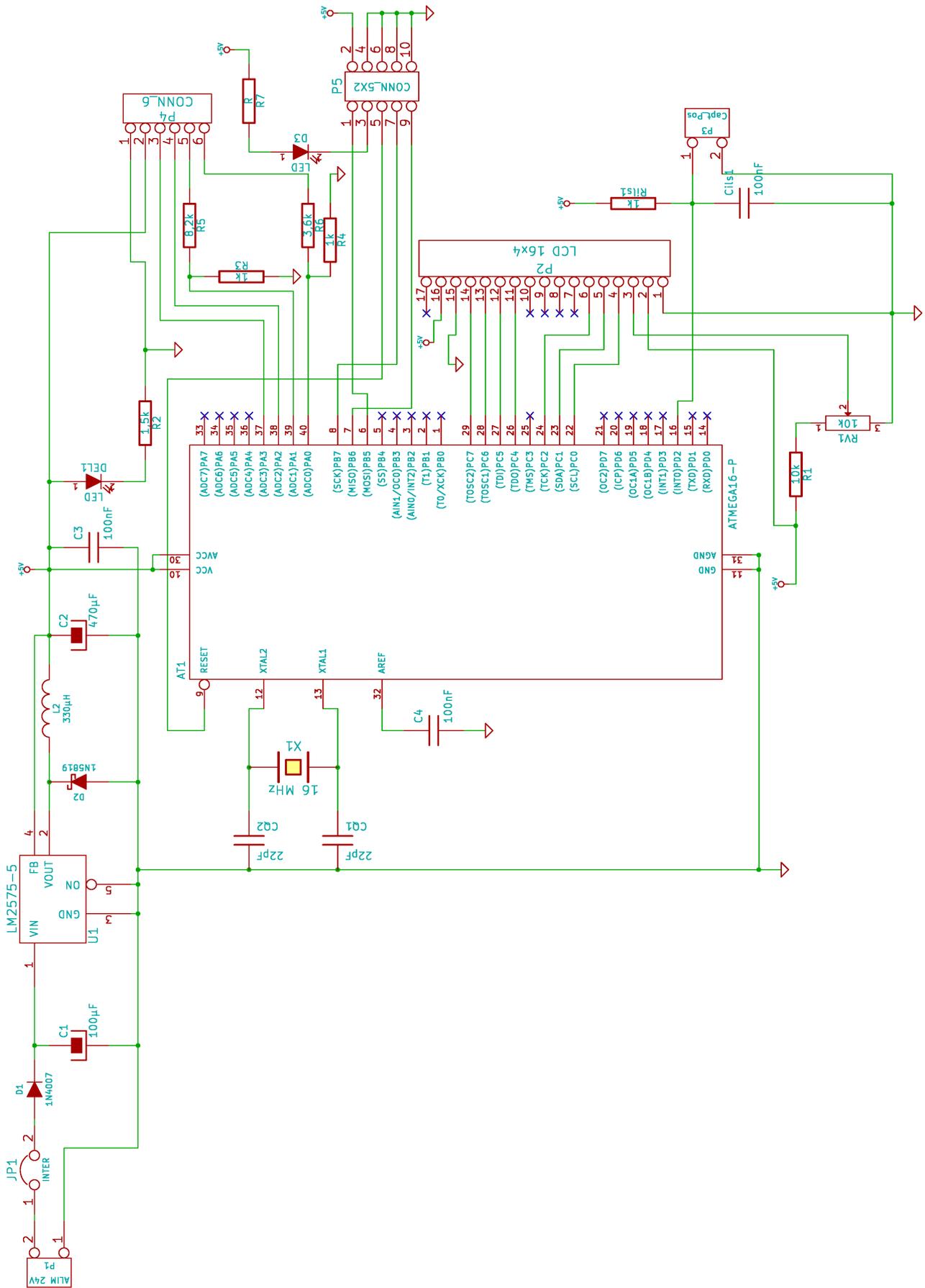


Illustration 24: Schéma structurel de la carte [3]

4. Conception de la carte

La solution pour réaliser le projet étant trouvée et étudiée, il devient dès lors possible de concevoir la carte électronique qui sera implantée dans le boîtier en plastique.

4.1. Réalisation

La réalisation de la carte passe par les étapes suivantes : saisie du schéma structurel, réalisation du typon, impression du typon et de la nomenclature, insolation, révélation, gravure, perçage, élimination de la couche de vernis, coupure de la carte aux bonnes dimensions puis soudage des composants.

Nous prendrons le soin d'expliquer chaque étape :

- La carte a été réalisée avec le logiciel de CAO disponible à l'IUT intitulé *KiCad* (la saisie du schéma structurel avec *Eeschema Capture* puis le typon avec *PcbNew*). Afin de terminer le projet dans les délais impartis, nous avons préféré avancer la conception du typon pendant la période de vacances.

- Après s'être équipé d'une plaque d'époxy et avant de placer cette dernière dans la machine à insoler, le film plastique protecteur collé sur la plaque a été retiré.



- Une fois la protection retirée, placer le typon dans la machine à insoler ainsi que la plaque en respectant le sens. Mettre la machine à insoler en marche pour une durée de 2 min 30 s.

- Retirer la plaque de la machine à insoler et la plonger dans le révélateur (soude).

- Allumer le jet (pompe), mettre en marche la graveuse puis placer la plaque à l'entrée du système.

- Une fois le cycle terminé, frotter la carte avec l'éliminateur puis rincer à l'eau tiède.



- Percer les trous avec un foret de diamètre 0,8 mm.

- Couper la plaque aux dimensions prévues (cf. Cahier des charges et spécifications) ainsi que les coins afin que la carte puisse être implantée dans le coffret.

- S'équiper du schéma d'implantation de la carte, de la nomenclature, du schéma structurel et des composants puis commencer la soudure de ces derniers en faisant attention à bien polariser les composants qui le nécessitent.

- Pour placer les différents borniers, le boîtier devra être percé aux dimensions adéquates.

Voici ci-dessous un aperçu de la carte électronique après réalisation.

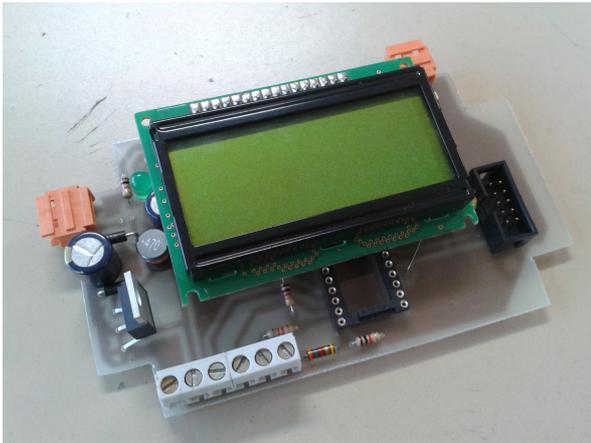


Illustration 25: Carte électronique réalisée [3]



Illustration 26: Carte électronique implantée dans son boîtier [3]

Nous trouverons ci-après les typons et schémas d'implantation des deux cartes.

4.1.1. Schéma structurel et typon

- Saisie du schéma structurel : cf. page 24.
- Réalisation du typon : carte simple face avec un unique via⁴ et plan de masse.

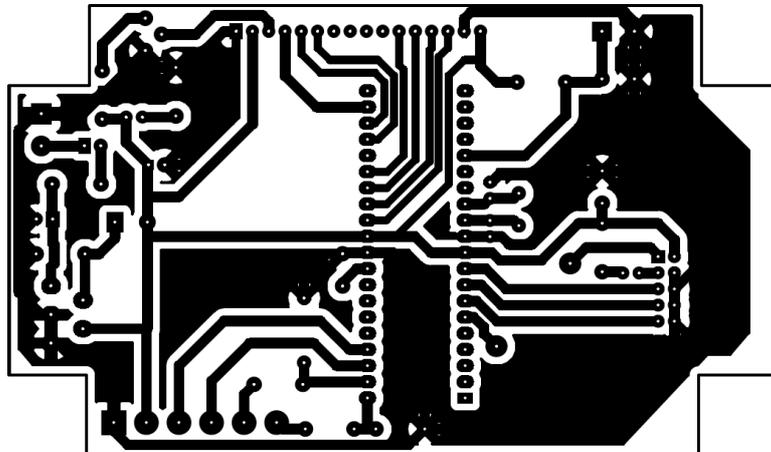


Illustration 27: Typon de la carte [3]

⁴ Un via (ou strap) est une connexion effectuée au moyen d'un simple fil. Ce type de liaison est utilisé lorsque réaliser une piste sur une carte électronique devient difficile ou impossible.

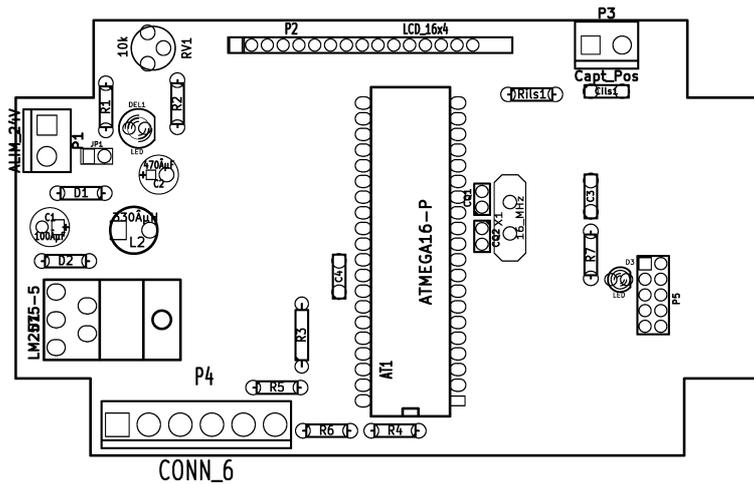


Illustration 28: Schéma d'implantation de la carte [3]

4.2. Montage de la carte

La mise en place de la carte électronique sur l'auto-laveuse présentait un certain nombre de contraintes. L'espace disponible sur l'auto-laveuse était effectivement restreint et il a notamment fallu démonter une roue afin de pouvoir y insérer les aimants et le capteur. Le capteur a été placé face à la roue côté intérieur et de façon parallèle aux aimants.

De plus, nous avons choisi de donner à l'utilisateur la possibilité de pouvoir débrancher la carte électronique de manière rapide. Pour cela, nous avons décidé d'utiliser des connexions SUB-D9. L'avantage de cette connexion est le nombre de broche qu'elle dispose, toutes les informations passeront par cette connexion :

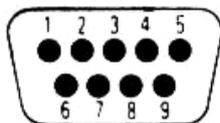


Illustration 29: Brochage de la connexion SUB-D9 [6]

N° de broche	Couleur fil	Désignation
1	Noir	0V (masse)
2	Bleu	+12V
3	Rouge	+24V
4		
5	Orange	+5V
6		
7	Vert	SDA
8	Jaune	SCL
9		

Tableau 4: Brochage de la connexion SUB-D9 [3]



4.3. Tests et essais

Il s'agit désormais de la phase finale du projet, l'une des plus importantes puisque c'est après réalisation des tests que nous verrons si le projet est fonctionnel.

Des essais sur maquette ont été réalisés avant la réalisation de la carte électronique. Les tests s'avéraient positifs et fonctionnels. Toutes les fonctions évoquées dans le cahier des charges répondaient aux attentes

Après réalisation de la carte électronique, nous avons procédé aux tests finaux.

Dans un premier temps, tous les circuits intégrés ont été retiré s'ils étaient placés sur leur support. En effet, une mauvaise manipulation ou un défaut de conception pourrait détruire les circuits intégrés à la première mise en service.

Après avoir branché les différents câbles à l'auto-laveuse, il était important de s'assurer que l'alimentation était correcte. Tout d'abord, il a fallu vérifier que la tension des batteries avoisinait bien les 24V prévus. En sortie du régulateur, nous devons obtenir une tension +5V stable (pour relever cette tension, cf. brochage LM2575 page 10). Lorsque l'alimentation +5V est fonctionnelle, la LED verte doit s'allumer. Les tensions d'alimentation ont aussi été vérifiées sur les broches

- du capteur de température LM75 (cf. brochage page 20) :

Sur la broche 8 : +5 V

Sur la broche 4 : 0 V

- de l'ATmega8535 (cf. brochage page 9) :

Sur la broche 10 et 30 : +5 V

Sur la broche 11 et 31 : 0 V

Une fois le circuit d'alimentation vérifié et fonctionnel, il devenait alors possible de mettre les circuits intégrés sur leur support et de vérifier que la carte était bien programmable en branchant la carte à l'ordinateur par le biais du bornier prévu à cet effet.

A ce stade, l'implantation du programme développé dans le micro-contrôleur provoquait le scintillement de la la LED jaune témoignant de l'échange d'informations.

Suite à l'implantation du programme dans l'ATmega8535, nous devons vérifier les différentes informations que l'utilisateur était en mesure de voir sur l'afficheur. Cependant, par manque de temps, les tests sur le programme n'ont pu être vérifiés et ré-ajustés qu'à la dernière séance. La procédure de tests prévue est ainsi explicitée ci-après mais le résultat n'est pas mentionné dans ce rapport.

La procédure de tests est alors la suivante :

× Mesure des tensions des batteries à l'aide d'un voltmètre puis comparaison avec les tensions affichées.

× Vérifier que l'état de charge des deux batteries en pourcentage est correct.

× Mesurer la vitesse de l'auto-laveuse à l'aide d'un tachymètre et la comparer à la valeur affichée sur l'écran (la vitesse maximale de l'auto-laveuse est de 4 km/h).

× Mesurer la température à l'aide d'un thermomètre et s'assurer qu'elle corresponde à l'affichage

Les tests sur maquette correspondaient aux attentes théoriques, ces tests doivent donc normalement correspondre aux attentes théoriques. Le fonctionnement de la carte réalisée et du programme développé sera donc validé au cours de la dernière séance. Le projet deviendra ainsi normalement fonctionnel.

Conclusion

Au travers de cette analyse et des séances d'études et réalisations, nous avons pu réaliser presque intégralement le projet qui nous était confié. Cette expérience nous a permis à la fois de concrétiser des connaissances en informatique et d'en acquérir de nouvelles en utilisant un nouveau logiciel *CodeVision AVR*.

La finalité du projet sera certainement utile par la suite sur le système de l'auto-laveuse, voir même nécessaire surtout pour la mesure de tension des batteries puisque si une batterie est déchargée, l'autre peut être endommagée. Il est nécessaire que la tension des deux batteries soit équilibrée.

Un système plus performant aurait pu être imaginé mais nécessiterait cependant une étude plus approfondie et des séances supplémentaires.

Une amélioration souhaitable serait le renforcement de la précision pour la mesure de vitesse. On pourrait également imaginer l'ajout de nouvelles fonctions telles que la motorisation de l'auto-laveuse avec plusieurs seuils de vitesse ou encore l'acquisition de plus d'informations liées à l'auto-laveuse et le développement de la navigation dans un menu.

Résumé

Notre choix de projet tuteuré s'est porté sur la mesure de vitesse d'une auto-laveuse et l'affichage de la tension des batteries. L'objectif était alors la mise en œuvre d'une carte électronique et le développement d'un programme informatique. Disposant d'un cahier des charges assez contraignant, notamment sur le plan mécanique, la réalisation du projet n'a pas été simple.

Des essais sur maquette avant la réalisation concrète du projet nous ont apporté une meilleure approche du sujet. Les tests étant positifs, nous avons pu réaliser la carte électronique à l'aide du logiciel de CAO électronique *KiCad*. Une fois la carte électronique testée, le micro-contrôleur implanté sur cette dernière a été programmé. Enfin, nous avons pu approuver le fonctionnement et le respect du cahier des charges du système.

Ce projet en binôme nous a permis l'acquisition et/ou la mise à l'épreuve de compétences applicables aux domaines de l'électronique et de l'informatique. Ce fut également l'occasion de découvrir un logiciel de programmation (*CodeVision AVR*) que nous ne connaissions pas.

Table des illustrations

Illustration 1: Dimensions du boîtier PC 100/35 LG [2].....	6
Illustration 2: Auto-laveuse [1].....	6
Illustration 3: Schéma fonctionnel de niveau 1 du système [3].....	8
Illustration 4: Schéma fonctionnel de niveau 2 du système [3].....	8
Illustration 5: Brochage de l'ATmega 8635 [4].....	9
Illustration 6: Fonctions à réaliser [3].....	9
Illustration 7: Schéma d'alimentation de la carte avec régulateur [5].....	10
Illustration 8: Brochage du régulateur LM2575 [5].....	10
Illustration 9: Aimant P62500000 [6].....	11
Illustration 10: Capteur PLA 137 [6].....	11
Illustration 11: Principe de fonctionnement du capteur ILS [7].....	11
Illustration 12: Rebonds produits par le capteur ILS (entre les deux curseurs) [3].....	12
Illustration 13: Solution anti-rebonds pour le capteur ILS [3].....	12
Illustration 14: Ordinogramme - Mesure de la vitesse [3].....	13
Illustration 15: Câblage des deux batteries sur l'auto-laveuse [3].....	14
Illustration 16: Schéma électrique pour la mesure de tension des batteries [3].....	14
Illustration 17: Ordinogramme - Mesure de la tension des batteries [3].....	16
Illustration 18: Pourcentage du niveau de charge en fonction de la tension de la batterie [3].....	17
Illustration 19: Courbe de charge des batteries à 3 étapes [8].....	18
Illustration 20: Ordinogramme - Fonction principale du programme [3].....	19
Illustration 21: Brochage du LM75 [9].....	20
Illustration 22: Registre de température (I2C) [9].....	20
Illustration 23: Schéma électrique de l'interface I2C [10].....	21
Illustration 24: Schéma structurel de la carte [3].....	24
Illustration 25: Carte électronique réalisée [3].....	26
Illustration 26: Carte électronique implantée dans son boîtier [3].....	26
Illustration 27: Typon de la carte [3].....	26
Illustration 28: Schéma d'implantation de la carte [3].....	27
Illustration 29: Brochage de la connexion SUB-D9 [6].....	27

Table des tableaux

Tableau 1: Planning prévisionnel et réel de réalisation du projet [3].....	7
Tableau 2: Équivalent des températures en binaire et hexadécimal [9].....	21
Tableau 3: Nomenclature et coût de fabrication de la carte [3].....	23
Tableau 4: Brochage de la connexion SUB-D9 [3].....	27

Bibliographie

- [1] **NILFISK**. *BA 410 - Autolaveuse futée pour tous les sols*, 2011. (page consultée le 12 Février 2012) <http://www.nilfisk.fr/Products/FloorCleaning-Maintenance/Scrubber_Dryers/SmallScrubber_Dryers/CA_BA410/BA410/Introduction.aspx>
- [2] **FIBOX**. *Fibox enclosures*, 2011. (page consultée le 25 Février 2012) <http://catalogs.fibox.com/catalogs/pdf_productcard.pl?pr_id=1&lang_code=ENG1>
- [3] **Anthony AUMOND - Jérémy HERGAULT**, Production personnelle, 2012.
- [4] **ATMEL**. *ATmega8535*, Octobre 2006. (page consultée le 31 Janvier 2012) <<http://www.atmel.com/Images/doc2502.pdf>>
- [5] **National Semiconductor**. *SIMPLE SWITCHER 1A Step-Down Voltage Regulator*, 2011. (page consultée le 27 Février 2012) <<http://www.national.com/mpf/LM/LM2575.html#Overview>>
- [6] **Farnell**. *Composants électroniques*, 2012. (page consultée le 10 Février 2012) <<http://fr.farnell.com>>
- [7] **GEEA**. *Capteur ILS*, 14 Mai 2004. (page consultée le 31 Janvier 2012) <<http://www.geea.org/spip.php?article210>>
- [8] **TecSup**. *Contrôle et interfaçage*, 2011. (page consultée le 20 Mars 2012) <<http://www.tecsup.fr/engineering/technologies/controle-interfaçage>>
- [9] **National Semiconductor**. *LM75 I2C Digital Temperature Sensor and Thermal Watchdog*, Février 1996. (page consultée le 9 Mars 2012) <<http://osg.informatik.tu-chemnitz.de/lehre/sem/ds/LM75.pdf>>
- [10] **Thierry LEQUEU**. *La documentation de Thierry LEQUEU sur OVH*, 31 Janvier 2012. (page consultée le 31 Janvier 2012) <<http://www.thierry-lequeu.fr/>>

Tables des annexes

Annexe 1: Documentation constructeur - batterie OPTIMA S5,5 [10].....	36
Annexe 2: Documentation constructeur – ATmega8535 [4].....	37
Annexe 3: Configuration du logiciel de programmation CodeVision AVR [3].....	40
Annexe 4: Programme complet (CodeVision AVR) [3].....	41

OPTIMA® YellowTop S 5,5



Battery Model: YT S 5,5
Part Number: 851 187 000 888 2
Nominal Voltage: 12 volts
NSN: 6140 01 502 4973
Description: High power, dual purpose engine start and deep cycle, sealed lead acid battery

Performance Data:

Open Circuit Voltage (fully charged): 13.1 volts
Internal Resistance (fully charged): 0.0025 ohms
Capacity: 75 Ah (C/20)
Reserve Capacity: BCI: 155 minutes
(25 amp discharge, 80°F (26.7°C), to 10.5 volts cut-off)

Recommended Charging Information:

Alternator: 13.65 to 15.0 volts
Battery Charger (Constant Voltage): 13.8 to 15.0 volts; 10 amps maximum; 6-12 hours approximate
Float Charge: 13.2 to 13.8 volts; 1 amp maximum (indefinite time at lower voltages)
Rapid Recharge: Maximum voltage 15.6 volts. No current limit as long as battery temperature remains below 50°C (125°F). Charge until current drops below 1 amp.
(Constant voltage charger)
Cyclic or Series String Applications: 14.7 volts. No current limit as long as battery temperature remains below 50°C (125°F). When current falls below 1 amp, finish with 3 amp constant current for 1 hour.
All limits must be strictly adhered to.

Recharge Time: (example assuming 100% discharge – 10.5 volts)

Current	Approx. time to 90% charge
100 amps	52 minutes
50 amps	112 minutes
25 amps	210 minutes

Recharge time will vary according to temperature and charger characteristics. When using Constant Voltage chargers, amperage will taper down as the battery becomes recharged. When amperage drops below 1 amp, the battery will be close to a full state charge.

(All charge recommendations assume an average room temperature of 25°C, 77°C)

Always wear safety glasses when working with batteries.

Always use a voltage regulated battery charger with limits set to the above ratings. Overcharging can cause the safety valves to open and battery gases to escape, causing premature end of life. These gases are flammable! You cannot replace water in sealed batteries that have been overcharged. Any battery that becomes very hot while charging should be disconnected immediately.

Not fully charging a battery can result in poor performance and a reduction in capacity.

Annexe 1: Documentation constructeur - batterie OPTIMA S5,5 [10]

Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 130 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 8K Bytes of In-System Self-Programmable Flash
Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
 - 512 Bytes EEPROM
Endurance: 100,000 Write/Erase Cycles
 - 512 Bytes Internal SRAM
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels for TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V for ATmega8535L
 - 4.5 - 5.5V for ATmega8535
- Speed Grades
 - 0 - 8 MHz for ATmega8535L
 - 0 - 16 MHz for ATmega8535



**8-bit AVR[®]
Microcontroller
with 8K Bytes
In-System
Programmable
Flash**

**ATmega8535
ATmega8535L**

2502K-AVR-10/06

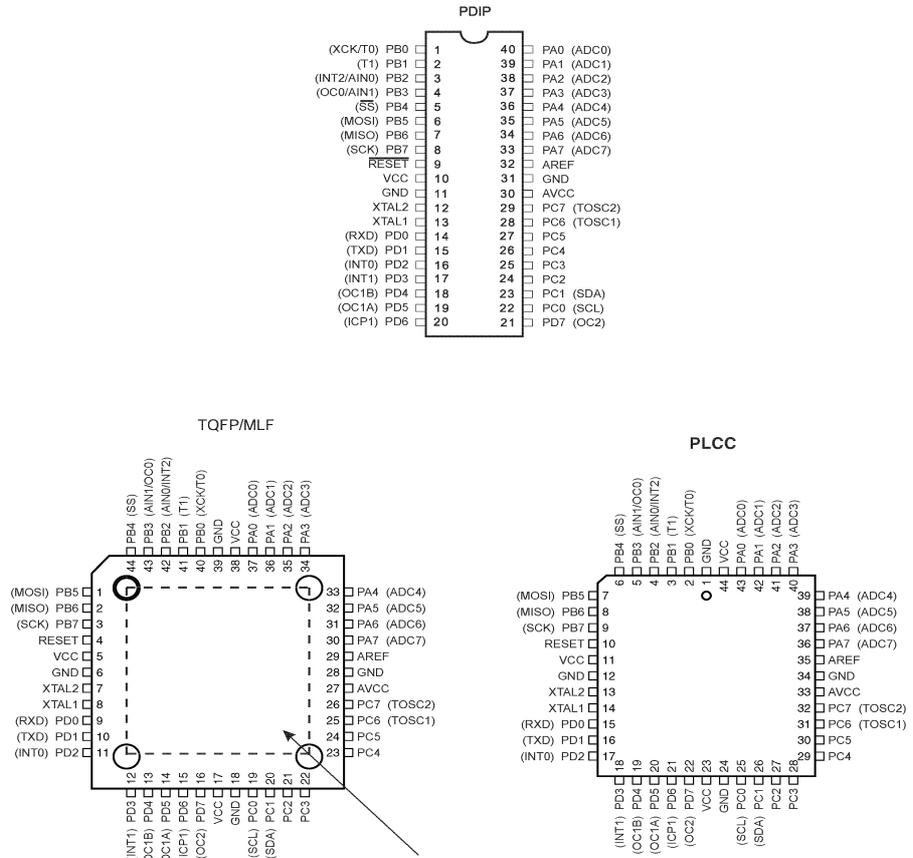


Annexe 2: Documentation constructeur – ATmega8535 [4]



Pin Configurations

Figure 1. Pinout ATmega8535



NOTE: MLF Bottom pad should be soldered to ground.

Disclaimer

Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.



Interrupts

This section describes the specifics of the interrupt handling as performed in ATmega8535. For a general explanation of the AVR interrupt handling, refer to “Reset and Interrupt Handling” on page 13.

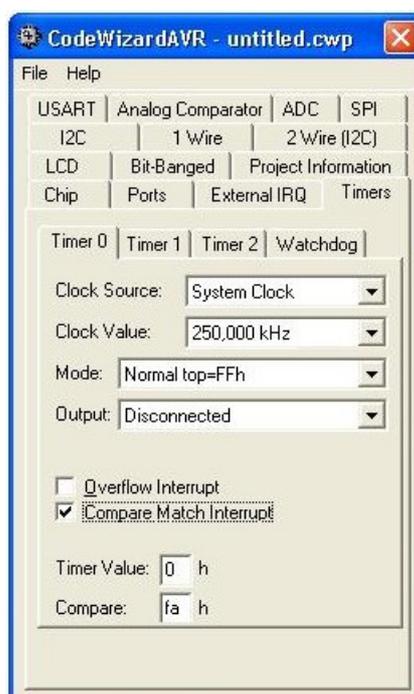
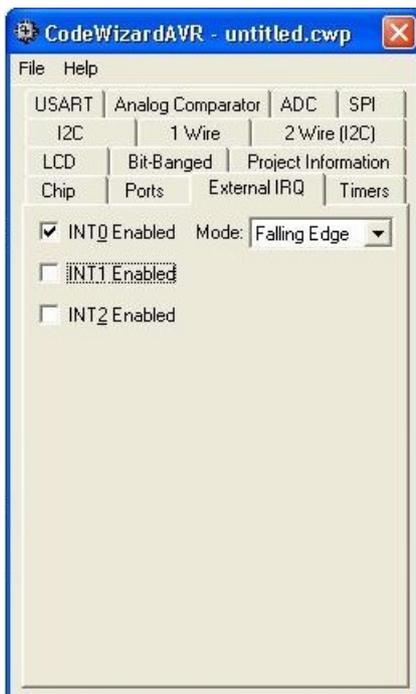
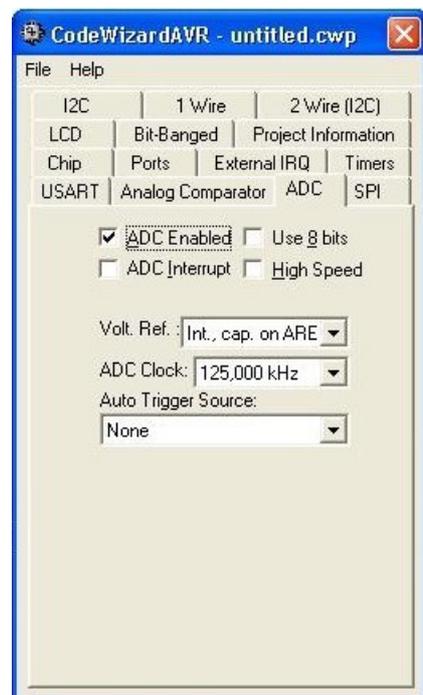
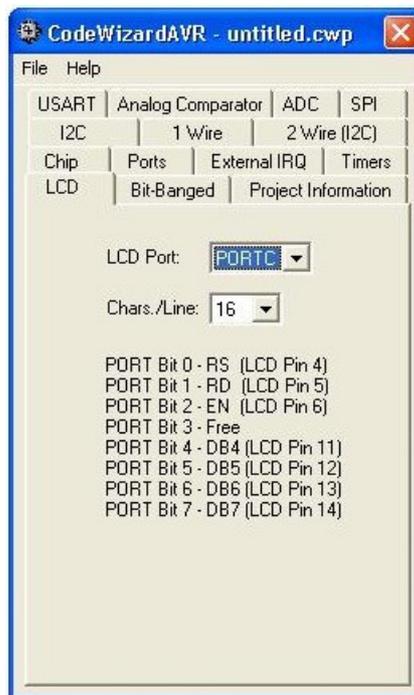
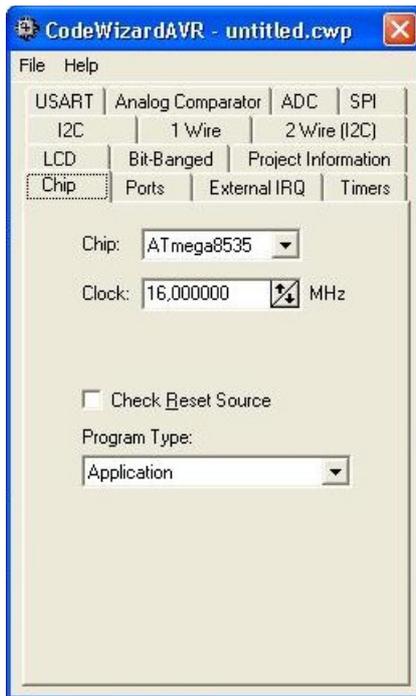
Interrupt Vectors in ATmega8535

Table 19. Reset and Interrupt Vectors

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	TWI	Two-wire Serial Interface
19	0x012	INT2	External Interrupt Request 2
20	0x013	TIMER0 COMP	Timer/Counter0 Compare Match
21	0x014	SPM_RDY	Store Program Memory Ready

- Notes:
1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see “Boot Loader Support – Read-While-Write Self-Programming” on page 224.
 2. When the IVSEL bit in GICR is set, Interrupt Vectors will be moved to the start of the Boot Flash section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash section.

Table 20 shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.



Annexe 3: Configuration du logiciel de programmation CodeVision AVR [3]

```

1  /*****
2  This program was produced by the
3  CodeWizardAVR V1.24.2c Professional
4  Automatic Program Generator
5  © Copyright 1998-2004 Pavel Haiduc, HP InfoTech s.r.l.
6  http://www.hpinfotech.ro
7  e-mail:office@hpinfotech.ro
8
9  Project : ProgAUTOLAV
10 Version : 1.0
11 Date    : 31/01/2012
12 Author  : HERGAULT Jeremy, AUMOND Anthony
13 Company : IUT GEII Tours
14 Comments: Programme AUTOlaveuse
15
16
17 Chip type      : ATmega8535
18 Program type   : Application
19 Clock frequency : 16,000000 MHz
20 Memory model   : Small
21 External SRAM size : 0
22 Data Stack size : 128
23 *****/
24
25 #include <mega8535.h>
26 #include <delay.h>
27 #include <stdio.h>
28
29 // I2C Bus functions
30 #asm
31     .equ __i2c_port=0x1B ;PORTA
32     .equ __sda_bit=2
33     .equ __scl_bit=3
34 #endasm
35 #include <i2c.h>
36
37 // LM75 Temperature Sensor functions
38 #include <lm75.h>
39
40 // Numero des entrées ADCx
41 #define BAT1 1
42 #define BAT2 6
43
44 // Adresse capteur de température
45 #define CAPT1 7
46
47 // Différence potentiel limite batteries en mV
48 #define DDP 10
49
50 #define ADC_VREF_TYPE 0xC0
51 // Fonction de conversion analogique numérique
52 unsigned int read_adc(unsigned char adc_input)
53 {
54     ADMUX=adc_input|ADC_VREF_TYPE;
55     // Début de la conversion
56     ADCSRA|=0x40;
57     // Attente de fin de conversion
58     while ((ADCSRA & 0x10)==0);
59     ADCSRA|=0x10;

```

Annexe 4: Programme complet (CodeVision AVR) [3]

```

60     return ADCW;
61 }
62
63 // Alphanumeric LCD Module functions
64 #asm
65     .equ __lcd_port=0x15 ;PORTC
66 #endasm
67 #include <lcd.h>
68
69 // Variable temporaire chaine LCD
70 unsigned char Valeur[16];
71
72 // Variable vitesse roue
73 long Tour;
74 long Nb_tour;
75 long Vitesse;
76
77 // Compteur rafraichissement timer
78 long refresh;
79
80 // External Interrupt 0 service routine
81 interrupt [EXT_INT0] void ext_int0_isr(void)
82 {
83     Nb_tour++; // Compteur de tour fonction de vitesse
84     delay_ms(2); // Attente avant de relancer une interruption
85 }
86
87 // Timer 0 overflow interrupt service routine
88 interrupt [TIM0_COMP] void timer0_comp_isr(void)
89 {
90     if((refresh) >= 6000 || (refresh < 0))
91     {
92         refresh = 0; // RAZ
93         Tour = Nb_tour;
94         Vitesse = ((Nb_tour)*5); // Calcul de la vitesse
95         Nb_tour = 0; // RAZ
96     }
97     refresh++; // incrementation du compteur de rafraichissement
98 }
99
100 void main(void)
101 {
102
103     // Variable valeur batterie
104     unsigned int UnitBat1 = 0;
105     unsigned int DecBat1 = 0;
106     unsigned int UnitBat2 = 0;
107     unsigned int DecBat2 = 0;
108     long Bat1 = 0;
109     long Bat2 = 0;
110     long Bat = 0;
111
112     // Variable pourcentage batterie
113     unsigned char Etat_Bat = 0;
114
115     // Variable temperature
116     long Temp = 0;
117
118

```

```

119 // Input/Output Ports initialization
120 // Port A initialization
121 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
122 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
123 PORTA=0x00;
124 DDRA=0x00;
125
126 // Port B initialization
127 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
128 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
129 PORTB=0x00;
130 DDRB=0x00;
131
132 // Port C initialization
133 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
134 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
135 PORTC=0x00;
136 DDRC=0x00;
137
138 // Port D initialization
139 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
140 // State7=T State6=T State5=T State4=T State3=T State2=P State1=T State0=T
141 PORTD=0x00;
142 DDRD=0x00;
143
144 // Timer/Counter 0 initialization
145 // Clock source: System Clock
146 // Clock value: 250,000 kHz
147 // Mode: Normal top=FFh
148 // OC0 output: Disconnected
149 TCCR0=0x03; // F = 250kHz
150 TCNT0=0x00;
151 OCR0=0xFA; //F = 1kHz, T=1ms
152
153 // Timer/Counter 1 initialization
154 // Clock source: System Clock
155 // Clock value: Timer 1 Stopped
156 // Mode: Normal top=FFFFh
157 // OC1A output: Discon.
158 // OC1B output: Discon.
159 // Noise Canceler: Off
160 // Input Capture on Falling Edge
161 TCCR1A=0x00;
162 TCCR1B=0x00;
163 TCNT1H=0x00;
164 TCNT1L=0x00;
165 ICR1H=0x00;
166 ICR1L=0x00;
167 OCR1AH=0x00;
168 OCR1AL=0x00;
169 OCR1BH=0x00;
170 OCR1BL=0x00;
171
172 // Timer/Counter 2 initialization
173 // Clock source: System Clock
174 // Clock value: Timer 2 Stopped
175 // Mode: Normal top=FFh
176 // OC2 output: Disconnected
177 ASSR=0x00;

```

```

178 TCCR2=0x00;
179 TCNT2=0x00;
180 OCR2=0x00;
181
182 // External Interrupt(s) initialization
183 // INT0: On
184 // INT0 Mode: Rising Edge
185 // INT1: Off
186
187 // INT2: Off
188 GICR|=0x40;
189 MCUCR=0x02;
190 MCUCSR=0x00;
191 GIFR=0x40;
192
193 // Timer(s)/Counter(s) Interrupt(s) initialization
194 TIMSK=0x02;
195
196 // ADC initialization
197 // ADC Clock frequency: 62,500 kHz
198 // ADC Voltage Reference: Int., cap. on AREF
199 // ADC High Speed Mode: Off
200 // ADC Auto Trigger Source: None
201 ADMUX=ADC_VREF_TYPE;
202 ADCSRA=0x87;
203 SFIOR&=0xEF;
204
205 // LCD module initialization
206 lcd_init(16);
207
208 // I2C Bus initialization
209 i2c_init();
210
211 // LM75 Temperature Sensor initialization
212 // thyst: 35°C
213 // tos: 40°C
214 // O.S. polarity: 0
215 lm75_init(CAPT1,35,40,0);
216
217 // Global enable interrupts
218 #asm("sei")
219
220 // Initialisation des variable a 0
221 Vitesse = 0;
222 refresh = 0;
223 Nb_tour = 0;
224 Tour = 0;
225
226
227 lcd_gotoxy(10,1);
228 lcd_putsf("tr/min");
229
230
231 while (1)
232     {
233
234         delay_ms(1000);
235
236         Temp = lm75_temperature_10(CPT1);

```

```

237     lcd_gotoxy(0,0);
238     lcd_putsf("AUTOLAV %d°    ", Temp);
239
240     // Lecture valeur batterie
241     UnitBat1 = (read_adc(BAT1)*26)/1024;
242     DecBat1 = (((read_adc(BAT1)*26)%1024)*10)/1024;
243     UnitBat2 = (read_adc(BAT2)*26)/1024;
244     DecBat2 = (((read_adc(BAT2)*26)%1024)*10)/1024;
245
246     Bat1 = UnitBat1*10 + DecBat1;
247     Bat2 = UnitBat2*10 + DecBat2;
248
249     Bat = ( Bat1 + Bat2 )/2;
250
251     Etat_Bat = 4*Bat - 424;
252
253     sprintf(Valeur,"%d    ", Vitesse);
254     lcd_gotoxy(0,1);
255     lcd_puts(Valeur);
256
257     sprintf(Valeur, "1:%d,%dV 2:%d,%dV  ", UnitBat1, DecBat1, UnitBat2, DecBat2);
258     lcd_gotoxy(0,2);
259     lcd_puts(Valeur);
260
261
262     if(((Bat1 - Bat2) > DDP) || ((Bat2 - Bat1) > DDP))
263     {
264         sprintf(Valeur, "Defaut batterie ");
265         lcd_gotoxy(0,3);
266         lcd_puts(Valeur);
267     }
268     else if((Bat1 > 136) && (Bat2 > 136))
269     {
270         sprintf(Valeur, "En charge    ");
271         lcd_gotoxy(0,3);
272         lcd_puts(Valeur);
273     }
274     }
275     else
276     {
277         sprintf(Valeur, "%d%%    ", Etat_Bat);
278         lcd_gotoxy(0,3);
279         lcd_puts(Valeur);
280     }
281     };
282 }
283

```