



UNIVERSITÉ
TOULOUSE III
PAUL SABATIER



Université
de Toulouse

Département EEA - Faculté Sciences et Ingénierie

Licence EEA Fondamentale

Projet EEA

Année 2021-2022

Compte rendu de projet EEA - Partie électronique

Table des matières

1 Objectifs du projet	3
1.1 Cahier des charges	3
1.2 Schéma fonctionnel	3
2 Caractérisation du capteur	4
2.1 Datasheet	4
2.2 Caractérisation expérimentale	4
3 Création des blocs	6
3.1 Bloc 1 : Conversion 5V/3.3V	6
3.2 Bloc 2 : Signal Trigger	6
3.3 Bloc 2 : Signal Dispo	7
3.4 Bloc 3 : Conversion temps-tension	7
3.5 Bloc 4 : Remise à zéro	8
3.6 Bloc 5 : Protection contre les surtensions	8
4 Circuit final	9
4.1 Circuit final	9
4.2 Résultats du circuit final	10
5 Conclusion	11
A Annexe	13
A.1 Simulations des différents blocs	13

Objectifs du projet

Le but du projet EEA du semestre 6 est de construire et commander un robot. Le robot est une voiture, et a pour but d'être utilisé avec d'autres robots pour simuler et étudier des embouteillages. La partie électronique de ce projet consiste à créer une interface entre l'Arduino qui contrôle le robot et le capteur de distance qui permet de détecter les obstacles. Cette interface devra permettre à l'Arduino de demander une mesure de distance, et la recevoir rapidement sous une forme déterminée.

1.1 Cahier des charges

Le cahier des charges est le suivant :

- Trigger le capteur à sa tension nominale.
- Convertir le signal du capteur en tension proportionnelle à la distance, avec une pente d' 1 V/m.
- Créer et envoyer à l'Arduino un signal "Dispo" prévenant que la mesure est prête.
- Envoyer à l'arduino la tension continue représentant la mesure de la distance.
- Remettre le circuit à zéro une fois la mesure terminée et récupérée par l'Arduino.
- Protéger l'Arduino contre les surtensions.
- Le temps maximal entre la demande d'une mesure et sa réception ne doit pas dépasser 10 ms.
- La distance maximale mesurable doit être de 3 m.

1.2 Schéma fonctionnel

On commence par faire un schéma fonctionnel de tous les blocs dont on aura besoin pour utiliser le capteur, convertir les divers signaux, et communiquer avec l'Arduino. Après étude du cahier des charges, nous arrivons au schéma fonctionnel de la figure 1.1 :

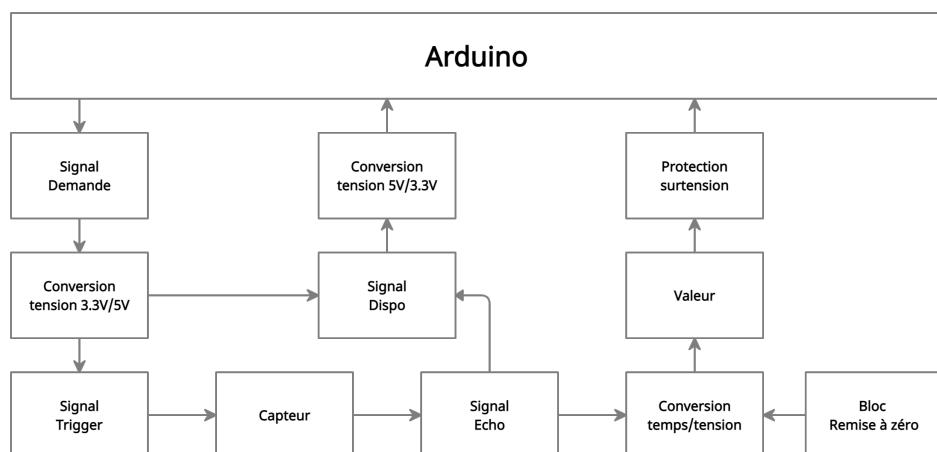


FIGURE 1.1 – Schéma fonctionnel du circuit.

Caractérisation du capteur

2.1 Datasheet

Avant de procéder au design des différents blocs, nous devons caractériser le capteur. Il s'agit d'un capteur de distance à ultrasons HC-SR04. Pour ce faire, nous utilisons la datasheet du constructeur [1] pour en extraire les informations importantes. On note que :

- La tension nominale du capteur est 5V.
- Le signal Trigger doit être un pulse de 10 μ s.
- Le signal Echo est un pulse dont la durée dépend de la distance de l'obstacle.

On peut déjà voir que théoriquement, pour une distance de 3m, le signal Echo renvoyé par le capteur sera de 17 ms, ce qui est supérieur aux 10 ms demandés par le cahier des charges. Il faudra modifier le cahier des charges, sur la distance maximale ou sur le temps maximal du signal Demande.

On note également que la datasheet[1] précise que l'obstacle devrait idéalement avoir une surface plane, et supérieure à 0.5 m^2 . Ce sont des recommandations qui ne seront pas respectées pour notre robot, pour la forme et la taille.

2.2 Caractérisation expérimentale

On peut ensuite utiliser un GBF et un oscilloscope pour générer un signal Trigger, mesurer la réponse de Echo, et ainsi caractériser notre capteur spécifique. On obtient la caractérisation présentée en figure 2.1

Echo (ms) en fonction de la distance d'un obstacle (cm)

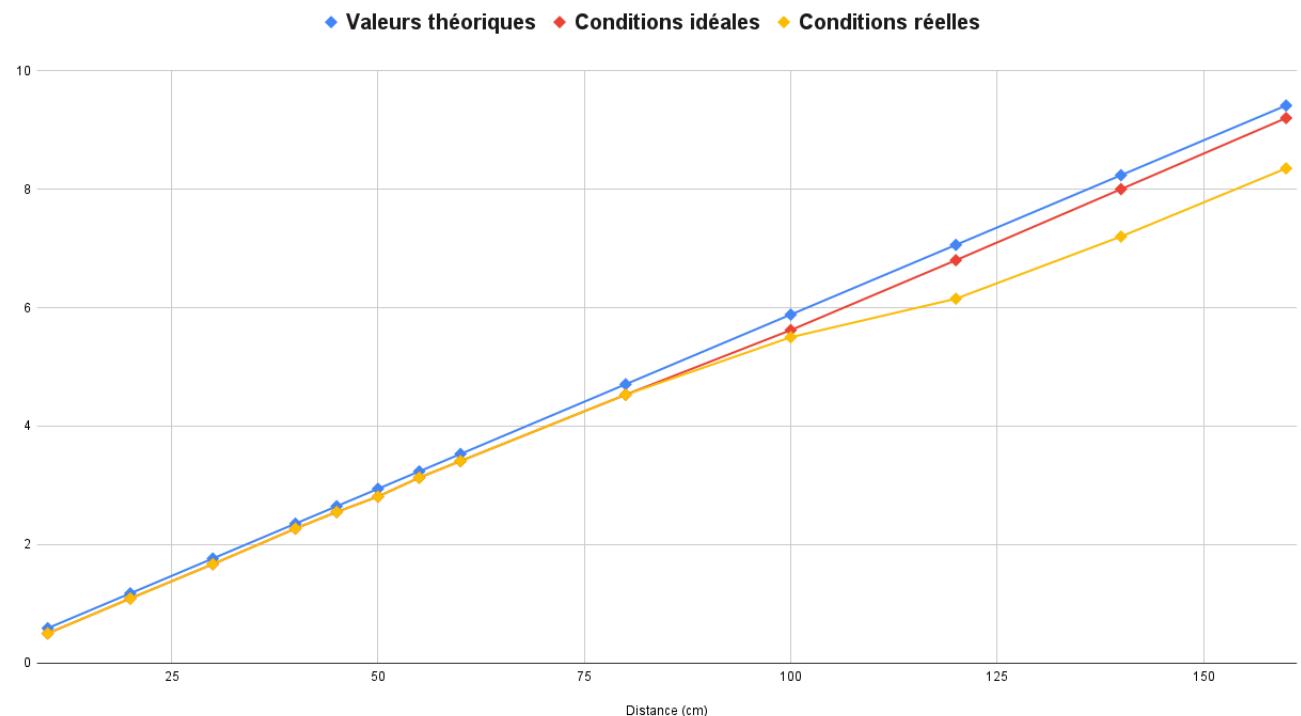


FIGURE 2.1 – Caractérisation du capteur.

Le capteur en conditions idéales, avec un obstacle de grande taille, complètement plat, et avec le capteur suspendu dans les airs présente une caractéristique linéaire et très proche des valeurs obtenues théoriquement.

Cependant, lors du test en conditions réelles, avec le capteur près du sol, et un obstacle de la taille du robot (environ 20x15 cm), on constate que la réponse du capteur n'est plus fiable à partir d'une distance d'1,2 m.

En consultant nos collègues d'Automatique, nous apprenons cependant que les lois de commande du robot seront de rester à 25 cm ou 1 m d'un obstacle. Cette information nous permet de dire que le capteur fonctionnera comme prévu pour les tests qui seront effectués. Cette information règle aussi le problème de la distance max de 3 m, et de la durée maximale de 10 ms de la communication Demande, puisque 1 m correspond à environ 5,9 ms.

Le cahier des charges initial ne sera donc pas respecté, mais le fonctionnement réel ne posera pas de problèmes.

Création des blocs

3.1 Bloc 1 : Conversion 5V/3.3V

L'arduino utilisé fonctionne à une tension de 3.3V, et le capteur à une tension de 5V. Pour fournir une tension adaptée au capteur, et pour protéger l'arduino, il faut une interface de conversion entre ces niveaux. Pour ce faire, nous utilisons des "Bi-directional Logic Level Converter" de Sparkfun[2]. La page de fonctionnement de ce composant[3] en explique l'utilisation. Il utilise des MOSFET canal N pour convertir un signal logique d'une tension à l'autre.

On pourra directement l'alimenter grâce aux sorties 5V et 3.3V de l'Arduino, pour passer d'un niveau à l'autre facilement.

3.2 Bloc 2 : Signal Trigger

Le signal Trigger déclenche le capteur, pour qu'il fasse une mesure. C'est un signal pulse qui doit durer environ 10 μ s. Il doit se déclencher lorsque le signal Demande est envoyé par l'Arduino.

La solution retenue est celle présentée dans la figure 3.1

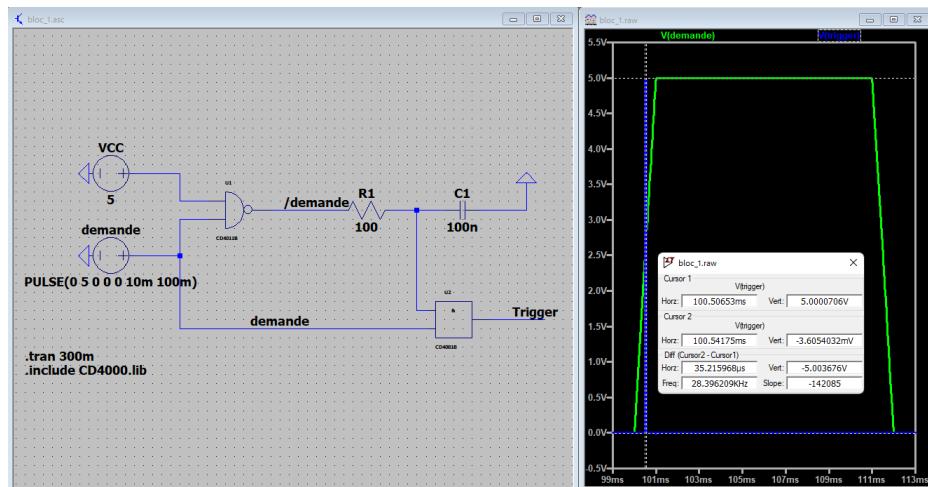


FIGURE 3.1 – Simulation du bloc 1 qui crée le signal Trigger.

On utilise une porte NAND pour créer le signal Demande. On utilise ensuite une porte AND entre le signal Demande, et le montage RC alimenté par Demande. Le condensateur C1 est chargé par Demande, et lorsque le signal Demande est activé, le signal Trigger devient vrai tant que le condensateur C1 garde de la charge.

On peut donc modifier la durée du signal Trigger en modifiant le couple RC. Dans notre cas, nous avons choisi le couple :

$$R = 100\Omega$$

$$C = 100nF$$

qui nous permet d'obtenir un Trigger qui reste activé 35 μ s, ce qui devrait fonctionner pour activer le capteur.

On utilise les composants fournis SN74LS09N (4 portes AND[4]), et HCF4011B (4 portes NAND[5]) pour réaliser ce montage.

Un test expérimental a permis de confirmer que ce montage permettait bien d'activer le capteur une fois pour un signal Demande.

3.3 Bloc 2 : Signal Dispo

Le signal Dispo doit signaler à l'Arduino que la mesure de distance est prête. C'est un signal qui sera activé quand le signal Echo de sortie du capteur sera terminé, et le signal Demande de l'arduino sera encore actif.

Pour le réaliser, on utilise un circuit à base de bascule RS, dont le schéma est donné dans le sujet du projet. La bascule RS et les circuits RC permettent d'activer la sortie Dispo uniquement sur les fronts descendants d'Echo. Les valeurs proposées par le sujet ne fonctionnaient pas comme prévu pendant la simulation, elles ont donc été changées.

La solution retenue et sa simulation sont présentées dans la figure 3.2

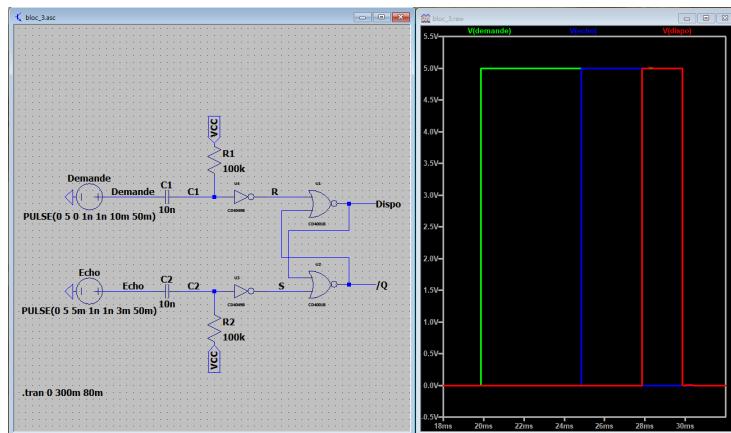


FIGURE 3.2 – Simulation du bloc 2 qui crée le signal Dispo.

Pour la réalisation de ce montage, on utilise le composant CD4001BE (4 portes NOR[6]). On utilisera des portes NOR avec les signaux requis et GND pour faire les portes inverseuses nécessaires.

Un test expérimental du montage sur platine LABDEC nous a confirmé son bon fonctionnement, le signal Dispo était renvoyé au moment voulu.

3.4 Bloc 3 : Conversion temps-tension

Ce bloc est chargé de convertir le pulse Echo de sortie du capteur en tension continue analogique, dont l'amplitude dépend de la distance mesurée (avec 1 V/m).

La solution initiale pour ce montage est l'utilisation d'un AOP intégrateur non inverseur. En effet, le signal Echo étant un pulse, en utilisant un tel AOP, on obtiendra une partie d'un signal triangulaire, et une sortie avec une tension variable selon la durée d'Echo. La pente de ce signal pourra être ajustée en faisant varier les couples RC de l'AOP.

Cependant, on remarque que le début de la charge d'un condensateur est presque linéaire. Comme la distance demandée par les lois de commande sera faible, on pourrait se passer de l'AOP, et utiliser directement un condensateur. La figure 3.3 présente le résultat des deux solutions pour un obstacle situé à 1 m de distance.

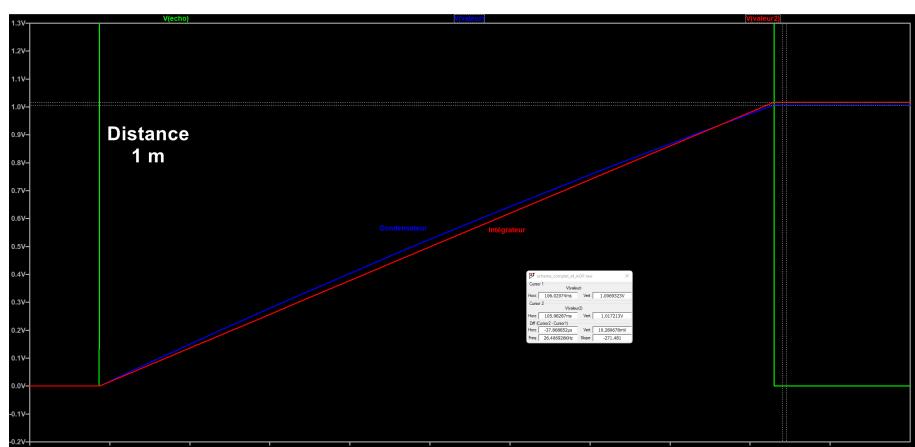


FIGURE 3.3 – Simulation des différences entre le condensateur seul et l'AOP, pour un obstacle à 1 m de distance.

Le circuit de cette simulation est présenté en annexe A.3.

On a adapté le couple RC pour obtenir un étalonnage parfait à une distance d'1 m. Cependant, de part la forme de la tension, la solution sans AOP indique un léger dépassement pour 25 cm. La simulation des différences à 25 cm est disponible dans la figure A.1 de l'annexe. On a au final une solution calibrée à 1 m, et qui donne 1 à 2 cm de trop entre 25 et 50 cm.

On pourra donc finalement retenir la solution de la charge de condensateur seul, ce qui simplifie le montage, réduit le coût des composants et fait baisser la consommation électrique du circuit d'environ 10%. Comme le robot fonctionne sur batterie, la réduction de consommation n'est pas négligeable même si la consommation du circuit est faible, environ 15 mA.

On obtient finalement le montage de la figure A.2 de l'annexe. On choisit les valeurs théoriques de R et de C pour régler la pente, et on obtient les composants suivants :

$$R = 230k\Omega$$

$$C = 100nF$$

Un test du montage sur platine LABDEC confirme bien les résultats, nous obtenons la tension continue Valeur qui varie en fonction de la distance d'un obstacle. La résistance R réelle du circuit a été adaptée, à cause des tolérances des composants réels.

3.5 Bloc 4 : Remise à zéro

La remise à zéro doit permettre de remettre le circuit à son état initial après une mesure récupérée par l'Arduino, pour se préparer à la mesure suivante.

Le principe du circuit retenu est simple : on utilise un interrupteur commandé par le signal Demande pour décharger le condensateur en mettant les deux bornes à la masse lorsque la communication est terminée.

Pour la réalisation, on utilise le composant MC14066 (4 interrupteurs contrôlés[7]) en reliant le pin de contrôle au signal Demande créé dans le bloc 1.

Le test sur platine LABDEC a confirmé que la remise à zéro fonctionnait comme prévu.

3.6 Bloc 5 : Protection contre les surtensions

L'arduino fonctionnant en 3.3 V, il ne faut pas dépasser cette tension. Le signal Dispo est pris en charge par le convertisseur logique de la conversion 5V/3.3V, mais il faut ajouter une protection pour le signal Valeur entrant dans l'Arduino.

On peut utiliser une diode de clamp pour limiter la tension. En prenant en compte la chute de tension de la diode, on doit avoir une tension d'environ 2.7 V à la sortie de la diode. On peut donc faire un pont diviseur de tension à partir de 5 V. On relie ensuite ce circuit en parallèle au signal Valeur, et il protègera l'Arduino pour les tensions dépassant 3.3 V. Le montage est présenté dans la figure A.5 de l'annexe. En utilisant une LED à la place de la diode, on peut avoir un indicateur lumineux de surtension.

Circuit final

4.1 Circuit final

Une fois tous nos blocs modélisés, et testés individuellement sur platine, on peut enfin faire une simulation du circuit complet, puis le montage en entier sur platine.

Le circuit complet sur platine Labdec et le circuit complet simulé sont présenté dans les figures 4.1 et 4.2.

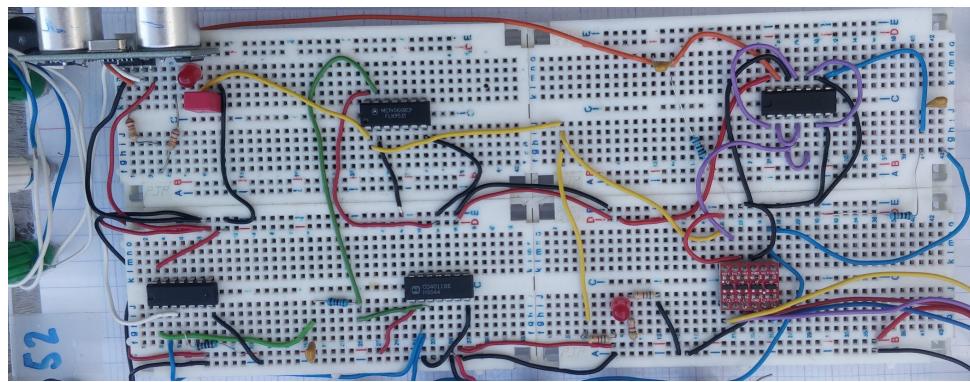


FIGURE 4.1 – Circuit complet monté sur platine Labdec.

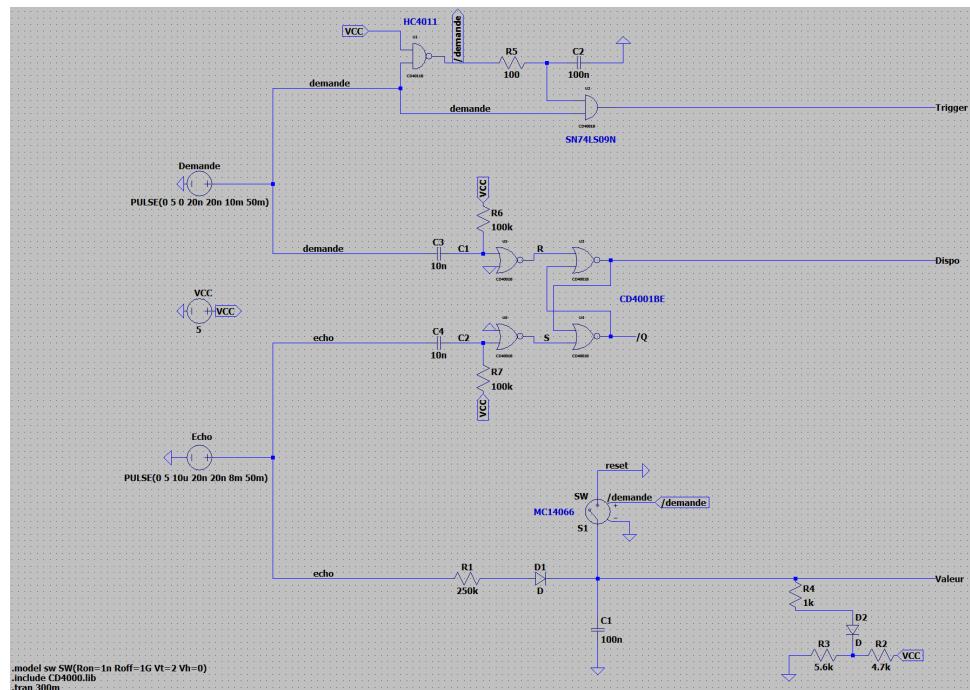


FIGURE 4.2 – Simulation LTSpice du circuit complet.

4.2 Résultats du circuit final

Une fois notre circuit monté, nous avons pu le tester et caractériser notre circuit. La figure 4.3 présente les résultats.

Résultats du circuit de mesure de distance

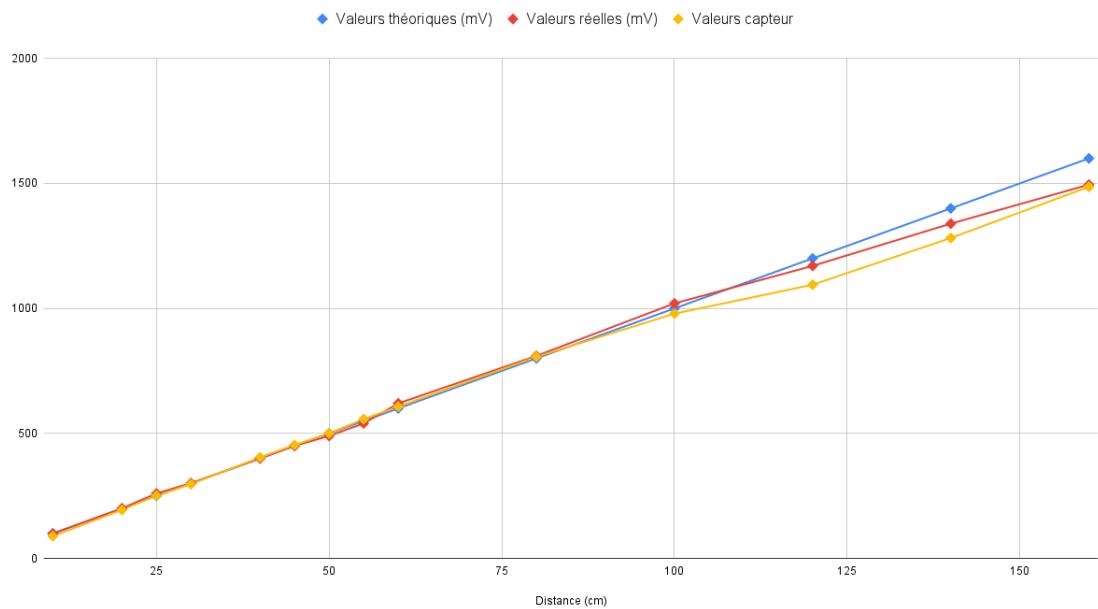


FIGURE 4.3 – Résultats du circuit de mesure de distance final.

On voit que, comme prévu, la distance mesurée est fiable jusqu'à 1 m de distance. On a un léger dépassement pour 25 cm, qui ne devrait pas impacter le fonctionnement du robot.

CHAPITRE 5

Conclusion

Les améliorations possibles pour notre circuit serait de passer à la solution avec un AOP intégrateur, pour améliorer la portée effective du montage. On pourrait aussi remplacer le capteur à ultrasons peu cher mais lent et peu fiable par un autre capteur ultrasonique plus performant, ou même un capteur avec technologie LED ou LIDAR.

Un autre axe d'amélioration serait sur le découplage. Ajouter des condensateurs de découplage pour tous les composants permettrait de compenser les chutes de tension dues aux pics de courant demandées par les composants.

En conclusion, ce projet nous a permis de monter en compétences sur l'utilisation de plusieurs composants discrets fonctionnant ensemble, la conversion de signaux, et une nouvelle utilisation de l'AOP intégrateur.

Bibliographie

- [1] «Ultrasonic Ranging Module HC - SR04» Internet : <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>, [consulté le 16/05/2022].
- [2] «SparkFun Logic Level Converter - Bi-Directional» Internet : <https://www.sparkfun.com/products/12009>, [consulté le 16/05/2022].
- [3] «Bi-Directional Logic Level Converter Hookup Guide» Internet : <https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide>, [consulté le 16/05/2022].
- [4] «Datasheet SN74LS09N» Internet : <https://www.ti.com/lit/ds/symlink/sn74ls09.pdf>, [consulté le 16/05/2022].
- [5] «Datasheet HCF4011B» Internet : <https://www.st.com/resource/en/datasheet/cd00000361.pdf>, [consulté le 16/05/2022].
- [6] «Datasheet CD4001B» Internet : <https://www.ti.com/lit/ds/sch8015c/sch8015c.pdf>, [consulté le 16/05/2022].
- [7] «Datasheet MC14066» Internet : <https://www.onsemi.com/pdf/datasheet/mc14066b-d.pdf>, [consulté le 16/05/2022].

A.1 Simulations des différents blocs

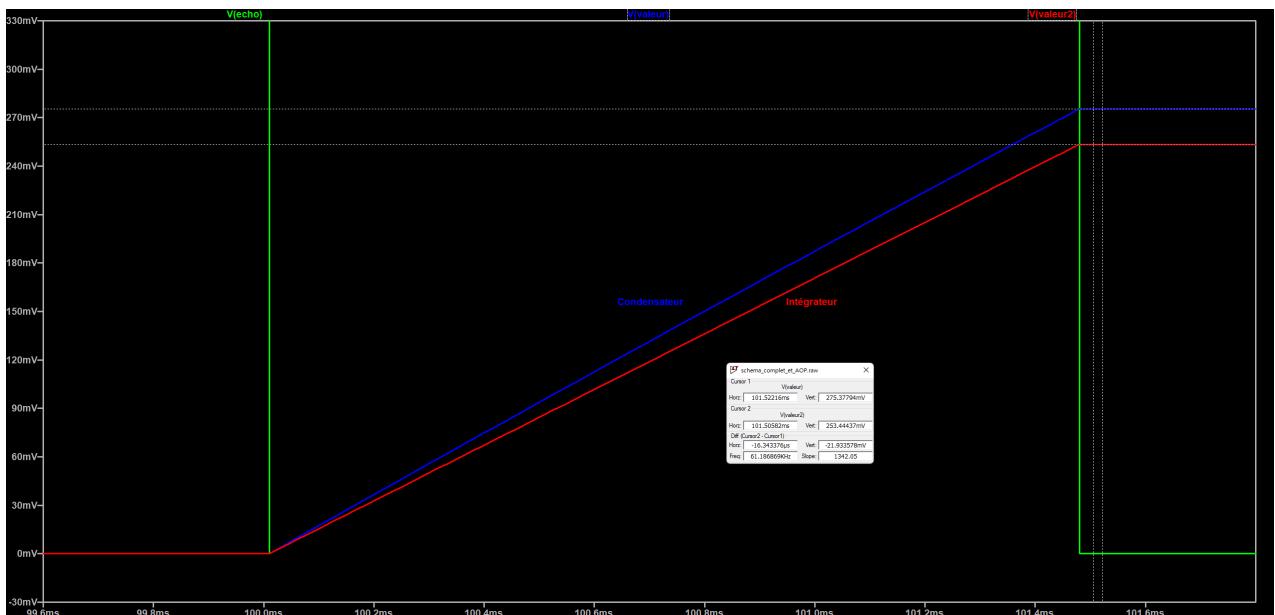


FIGURE A.1 – Comparaison des solutions condensateur et intégrateur pour un obstacle à 25 cm de distance.

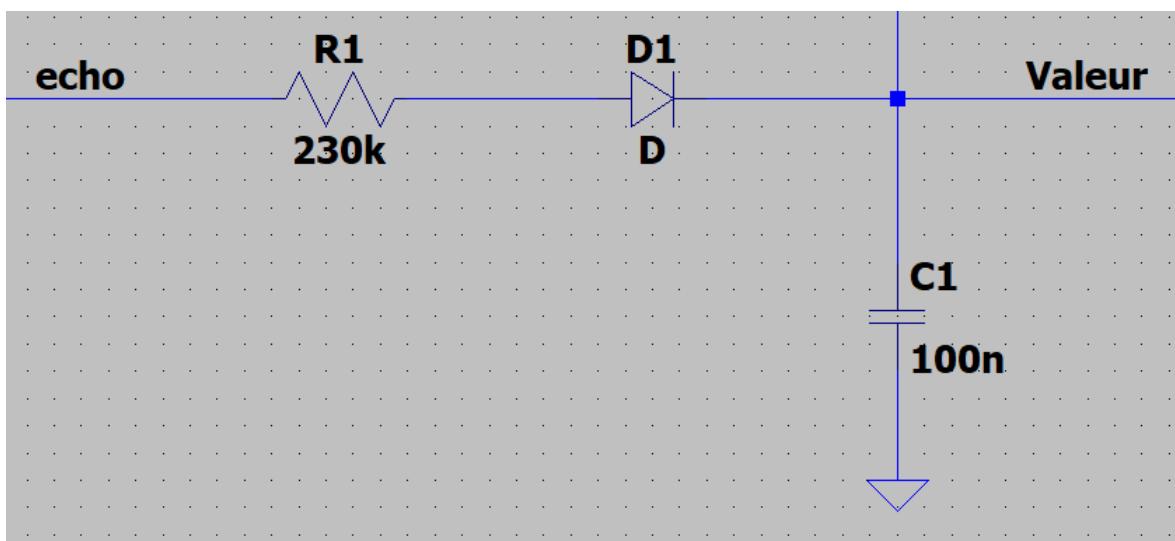


FIGURE A.2 – Montage RC de conversion temps-tension.

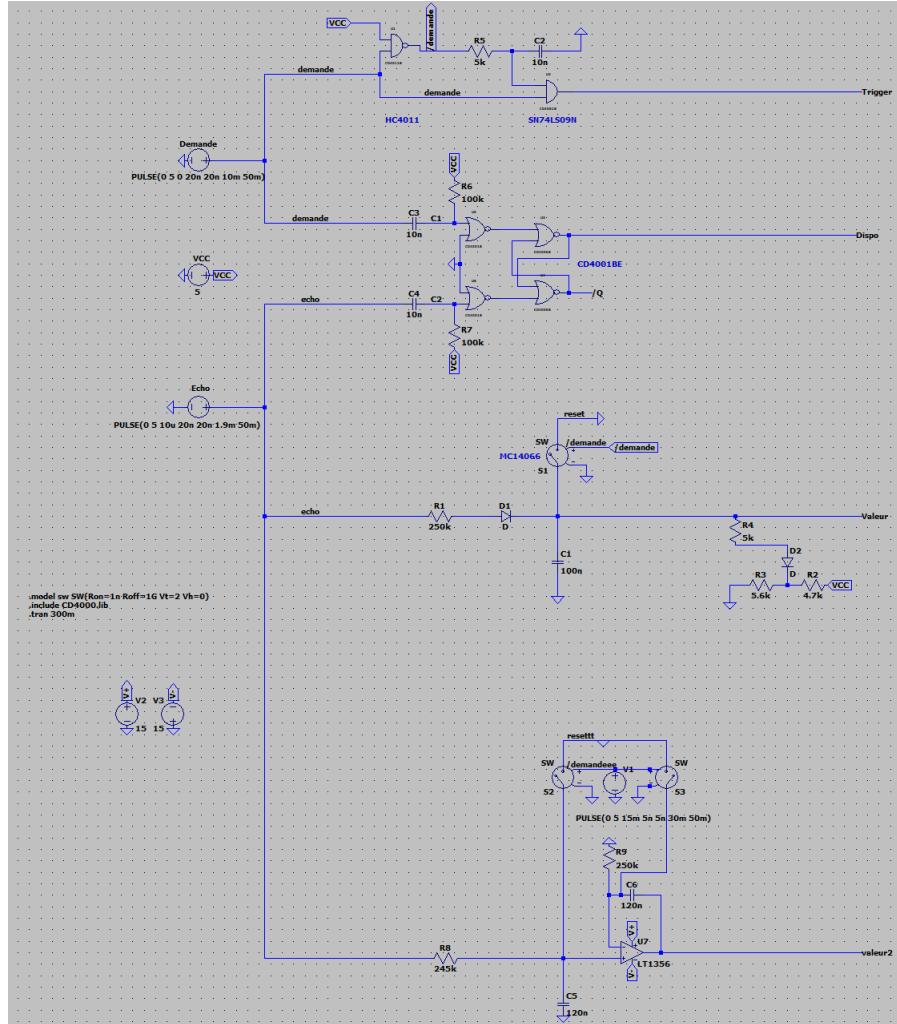


FIGURE A.3 – Circuit de simulation comparant les solutions AOP et condensateur simple

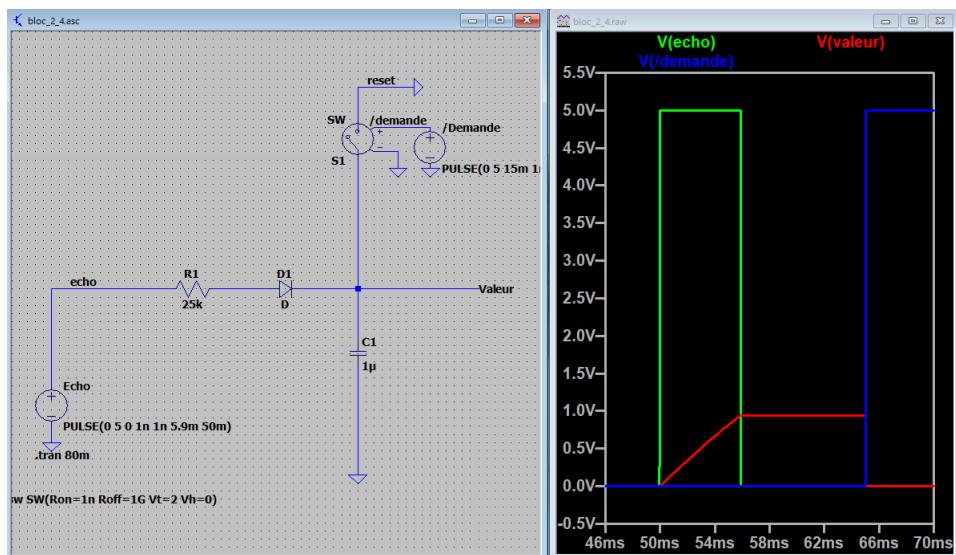


FIGURE A.4 – Simulation de la remise à zéro du circuit.

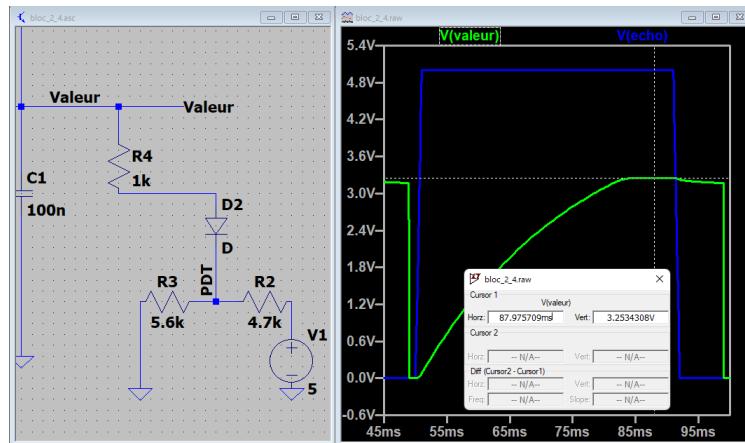


FIGURE A.5 – Simulation de la protection de l’Arduino.

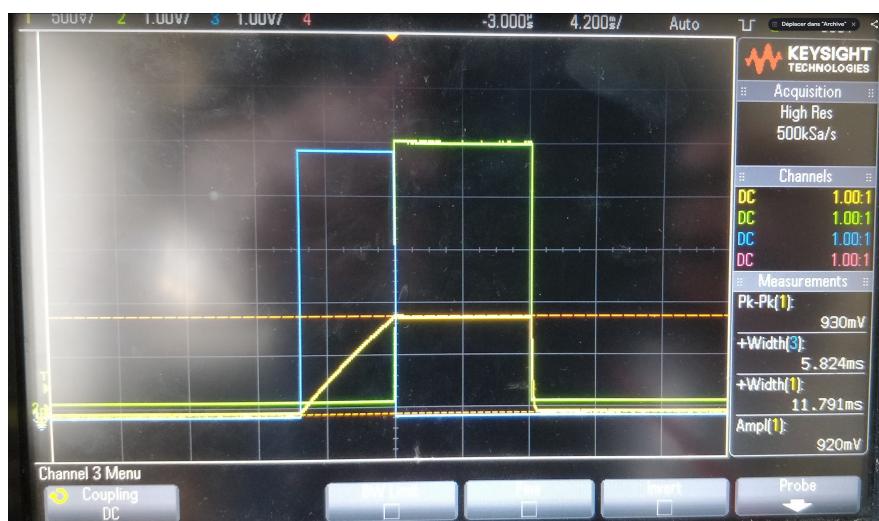


FIGURE A.6 – Photo de la réponse du circuit final sur un oscilloscope avec les signaux Echo (en bleu), Dispo (en vert) et Valeur (en jaune).



FIGURE A.7 – Photo des conditions de test du circuit final.