

WiCanne v 3.44

YLC→MHK 08/03/2021

Prototype expérimental MY HUMAN KIT

Cette version de la WiCanne a pour particularité d'utiliser un module d'intelligence artificielle pour tenter d'identifier les obstacles rencontrés.

Elle se décompose en 3 programmes qui utilisent tous les trois la même plateforme matérielle.

- Un programme préalable d'acquisition de mesures servant à la constitution de la librairie d'IA.
- Un programme d'utilisation de la WiCanne assistée par la librairie d'IA.
- Un programme de détermination expérimentale des paramètres de détection sonore.

La librairie d'IA est produite par le logiciel **NanoEdgeAI.Studio** qui a été gracieusement mis à notre disposition par la start-up **Cartesiam** de Toulon pour l'élaboration de ce projet.

Acquisition des données. (Prog. *Wi-Canne_v3-44_Acquisition*)

Pour pouvoir générer une librairie qui soit capable d'identifier des obstacles, il faut fournir un nombre suffisamment important de mesures corrélées à des obstacles précis pour que le logiciel NanoEdgeAI.Studio puisse les analyser et en déduire un algorithme de reconnaissance.

Il s'agit d'un processus dit de « machine learning » en mode « apprentissage supervisé ».

Ce programme permet d'enregistrer sur la carte SD intégrée au Teensy, des fichiers constitués chacun d'une série d'enregistrements correspondant à un obstacle particulier appelé « classe » (d'obstacle).

Il y aura donc à enregistrer autant de fichiers que de classes (ie : d'obstacles) différentes.

Chaque enregistrement correspond à une série de mesures faites lors du balayage de l'environnement (comme avec une canne blanche réelle).

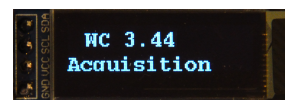
Le nombre d'enregistrement par classe, le nombre de mesures par enregistrement et la fréquence du Lidar sont fournis au programme par le fichier paramètres « WICAPAR » stocké sur la carte SD.

En l'absence de ce fichier paramètres, ce sont les paramètres par défaut du programme qui sont alors utilisés. Et un fichier paramètres avec ces valeurs par défaut est alors enregistré sur la carte SD afin de pouvoir réutiliser le mêmes valeur dans le second programme « TestAI ».

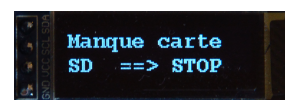
En effet, pour que la reconnaissance des obstacles puisse se faire correctement dans le second programme, il faut que l'acquisition des mesures soient faites dans ce premier programme de façon identiques. C'est à dire avec un geste de balayage de même amplitude, à la même fréquence et avec un nombre de mesures par geste identique.

A noter que d'autres paramètres, utilisés seulement par le second programme « TestAI », sont enregistrés en même temps dans la carte SD avec leurs valeurs par défaut. Il s'agit des paramètres utilisés pour la détection sonore des obstacle.

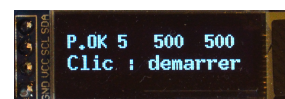
Au lancement du programme, l'écran affiche brièvement la version du programme et sa fonction.



Ensuite, celui-ci vérifie la présence d'une carte SD. Si celle-ci est absente, le programme s'interrompt.



Il lit ensuite le fichier des paramètres sur la carte SD. Si celui-ci est absent, un message indique que les paramètres par défaut seront utilisés. Sinon l'écran affiche « P.OK » et les paramètres qui ont été lus.



Il vérifie ensuite la présence ou non de fichiers de mesures déjà sur la carte SD. (fichiers « MESURE n.txt » où n est l'ordre d'enregistrement du fichier).

Si aucun fichier de mesure n'existe, il propose de démarrer l'enregistrement du premier fichier.

Si un nombre n de fichiers de mesures sont déjà présents sur la carte SD, il commence par les lister un par un en les comptant. Il ignore tous les autres fichiers dont le label ne commence pas par « MESURE ».

Puis il propose soit de tous les supprimer, soit de continuer à faire des enregistrements dans un nouveau fichier n+1.

Si on choisit la suppression, le nombre de fichiers supprimés est affiché et le programme propose de démarrer l'enregistrement du premier fichier.

Lorsque qu'on enregistre un nouveau fichier, le nom de celui-ci est affiché pendant 1 seconde puis l'enregistrement débute.

Pendant l'enregistrement d'un fichier, son numéro est affiché et un compteur indique en permanence le numéro d'enregistrement en cours.

Un bip est émis à chaque enregistrement pour aider à la synchronisation avec le geste de balayage.

À la fin de l'enregistrement, l'écran affiche le nom du fichier enregistré et propose de passer à l'enregistrement du fichier suivant.

NB : Il est important de noter la correspondance entre le numéro du fichier et le nom de l'obstacle qu'on y a enregistré.

Lors des tests, il a été décidé de mesurer 9 classes ou types d'obstacles, mais ceci n'est pas exhaustif. *Sol plat, Mur à gauche, Mur à droite, Mur devant, Poteau, Marche montante, Marche descendante, Escalier montant, Escalier descendant.*

Remarque : à l'utilisation, il s'avère qu'un trop grand nombre de mesures pour un même obstacle est pénalisant et ralentit la reconnaissance des obstacles lors de l'utilisation du second programme.

Il faut donc, en fonction de la vitesse à laquelle on balaye l'environnement, adapter la fréquence d'acquisition du Lidar et le nombre de mesures pour que le quotient des deux corresponde à la durée du geste.

Exemple : pour une durée du geste mesurée à 1,2 secondes en moyenne, si on veut enregistrer 500 mesures par geste, il faut régler la fréquence d'échantillonnage du Lidar à 417 Hz. Quant au nombre d'enregistrements, cela dépendra des capacités de l'ordinateur qui en fera le benchmark ; mais 100 enregistrements sembleraient suffisants.

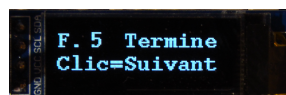
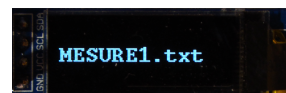
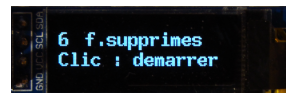
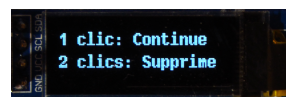
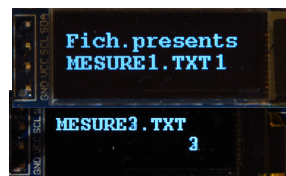
A noter que ce premier programme n'est pas destiné à être utilisé par une personne mal-voyante ou alors elle devra se faire assister pour effectuer cette acquisition de mesures.

Utilisation de la WiCanne. (Prog. Wi-Canne_v3-44_testAI)

C'est le programme destiné à la personne mal-voyante pour se déplacer dans son environnement.

Il utilise la même plateforme matérielle que le programme d'acquisition mais celle-ci est complétée par un smartphone Android avec lequel elle communique en bluetooth LE pour synthétiser oralement le nom des obstacles identifiés.

Il est donc naturel de démarrer l'application WiCanne sur le smartphone afin que la communication entre les deux appareils puisse se faire.



Au lancement du programme, celui-ci vérifie la présence de la carte SD et aussi la présence sur celle-ci du fichier des paramètres. Il faut, en effet pouvoir utiliser les mêmes paramètres que lors de l'acquisition des mesures en ce qui concerne la fréquence du lidar et le nombre de mesures à soumettre à l'IA.

Si la carte ou le fichier paramètres sont absents, un signal d'alarme sonore est émis et le programme s'interrompt.

Le programme explore en permanence l'environnement balayé par l'utilisateur. Il détecte les changements de distance dus à des surfaces non continues ou faisant un angle entre elles en émettant un bip sonore dont la tonalité et inversement proportionnelle à la distance de l'obstacle.

Ainsi, le balayage continu d'un sol plat ne donne lieu à aucun signal.

Par contre, dès qu'une variation brutale de distance suggère la présence d'un obstacle, un bip sonore proportionnel à la distance mesurée est émis. Plus l'obstacle est proche et plus la fréquence du bip est élevée.

NB : Un certain apprentissage sera certainement nécessaire pour appréhender l'environnement au son des bips entendus. Cela peut être aussi difficile à interpréter dans un environnement complexe avec de multiples obstacles rapprochés (grille ou branches par exemple).

Lorsque des bips se font entendre dans une direction donnée, cela permet de supposer la présence d'un obstacle dans cette direction. Compte tenu de la faible largeur du faisceau de détection, la taille de l'obstacle n'est pas un problème, et pour un obstacle étroit, il faudra faire un geste de balayage plus précis pour en estimer sa largeur.

La détermination sonore d'un obstacle est fonction de deux paramètres. Le premier détermine l'intervalle de mesures entre les 2 mesures prises en compte et le second est le seuil minimal de variation de distance à partir duquel on déclenche l'alerte sonore.

Ces paramètres sont initialisés à leur valeur par défaut dans le fichier paramètres dans le premier programme d'acquisition des classes d'obstacle. Ils peuvent être modifiés à la main dans un éditeur de texte ou par le programme de détermination expérimentale de ces paramètres. (CF : ce programme).

En présence d'un obstacle détecté par les bips sonores, l'utilisateur peut déclencher la demande d'identification de cet obstacle en appuyant sur le bouton.

Le programme va alors solliciter les routines d'intelligence artificielle pour essayer de reconnaître l'obstacle détecté.

Si l'obstacle est identifié, le code de cet obstacle est transmis, accompagné de sa distance par rapport à l'utilisateur, à l'application « WiCanne » du smartphone.

Le smartphone énonce alors le nom de l'obstacle qui a été identifié ainsi que son éloignement.

Lorsque l'utilisateur relâche le bouton, la détection continue et les bips sonores sont à nouveau émis mais le smartphone, lui, reste alors muet.

L'écran Oled de l'appareil affiche en permanence les distances mesurées ainsi que la classe des obstacles identifiés mais n'a d'utilité que pour contrôler le bon fonctionnement de l'appareil.

Remarques : Pendant l'utilisation sans appel à l'IA, donc bouton relâché, la boucle d'exploration de l'environnement est très rapide et le rendu sonore fonctionne à la même vitesse que le geste de balayage qui peut alors être plus rapide..

Par contre lorsqu'on appuie sur le bouton, les envois des codes des obstacles au smartphone sont cadencés à 2 secondes par envoi pour permettre l'énoncé du nom de l'obstacle et de sa distance. Le geste de balayage doit s'apparenter, à ce moment, à celui fait lors de l'acquisition des mesures.

Le balayage de l'environnement se fait naturellement de droite à gauche et de gauche à droite à environ 45° vers le bas. Mais en présence d'un obstacle identifié comme un escalier par exemple, on peut aussi compter le nombre de marches en effectuant un balayage de haut en bas, un bip devant être émis pour chaque marche balayée.

Application sur le Smartphone

Cette application ne fonctionne que sur un smartphone Android doté du Bluetooth LE (BLE). Elle est complémentaire du programme précédent puisque sa fonction est d'énoncer vocalement le nom des obstacles détectés par la WiCanne.

Au lancement de cette application, il suffit d'appuyer sur le bouton [scan] pour que le smartphone recherche la présence des appareils bluetooth dans son voisinage.

Un écran affiche, au fur et à mesure de leur détection, la liste des appareils disposant d'une connexion BLE à proximité.

Il suffit de cliquer sur la ligne correspondante à la WiCanne (xxxxxBT05-xx dans l'exemple ci contre) pour que le scan s'arrête et que la connexion s'établisse automatiquement avec le dispositif de la WiCanne.

Un nouvel écran s'affiche alors et indique le code identifiant l'appareil auquel on est connecté.

Dans une première fenêtre sont affichées entre crochets, les 4 valeurs des données brutes reçues, séparées par une virgule.

Ce sont de la gauche vers la droite :

- le code de l'obstacle retourné par l'IA.
- la partie « mètres » de la distance de cet obstacle,
- la partie « centimètres » de cette même distance,
- la distance réelle en centimètres de l'obstacle.

Une seconde fenêtre donne l'interprétation du code d'obstacle rencontré en affichant le rappel de son code et le nom auquel il correspond et qui sera énoncé vocalement.

Il affiche aussi la distance telle qu'elle est formulée vocalement.

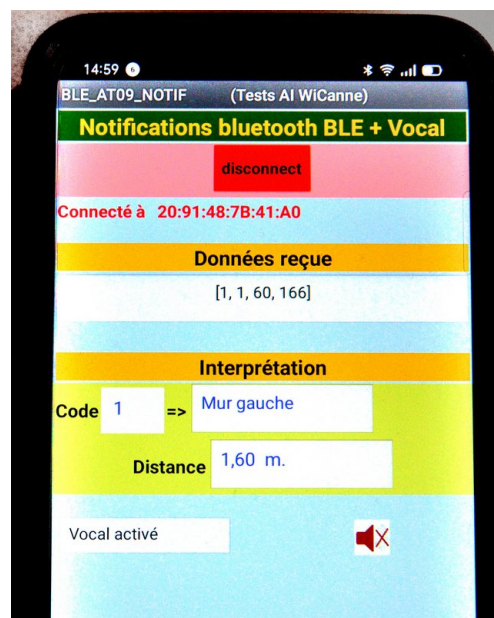
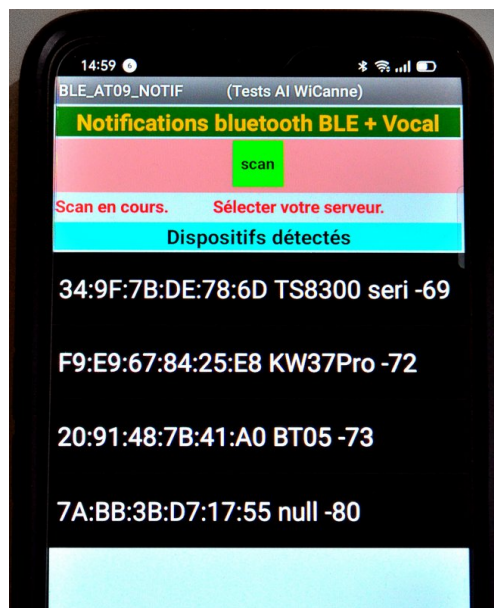
A noter que celle-ci est arrondie à la dizaine de centimètres la plus proche. Cela raccourcit le temps de l'énoncé et la précision centimétrique est inutile ici.

Un bouton sur l'écran du smartphone permet de supprimer ou d'activer la synthèse vocale tout en conservant l'affichage des données.

Tous ce qui est affiché ici n'est pas utile à l'utilisateur mal-voyant et ne sert réellement que pour le contrôle du bon fonctionnement du système.

D'ailleurs, l'utilisateur, une fois l'application lancée, coupe son téléphone et le range dans sa poche. Celui-ci continuera néanmoins à fonctionner en tâche de fond, écran éteint, et à annoncer vocalement les obstacles à son utilisateur en consommant de ce fait moins de batterie.

En cas de dysfonctionnement, on peut couper la liaison en appuyant sur le bouton [disconnect] et effectuer un nouveau scan (écran précédent).



Notes techniques

La plateforme matérielle est constituée de :

un micro-contrôleur Teensy 4.1 avec SD, un Lidar TF-Mini-Plus, un écran Oled (SSD 1306), une carte AT09 BLE, un buzzer, un bouton à contact non permanent.

L'alimentation se fait à l'aide d'un power-pack 5v. branché sur la prise USB du Teensy.

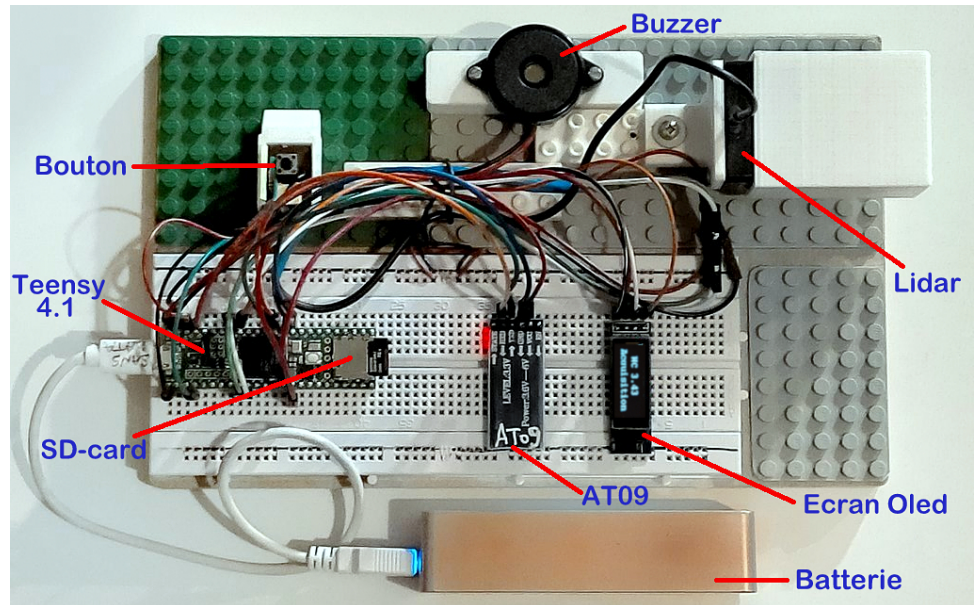
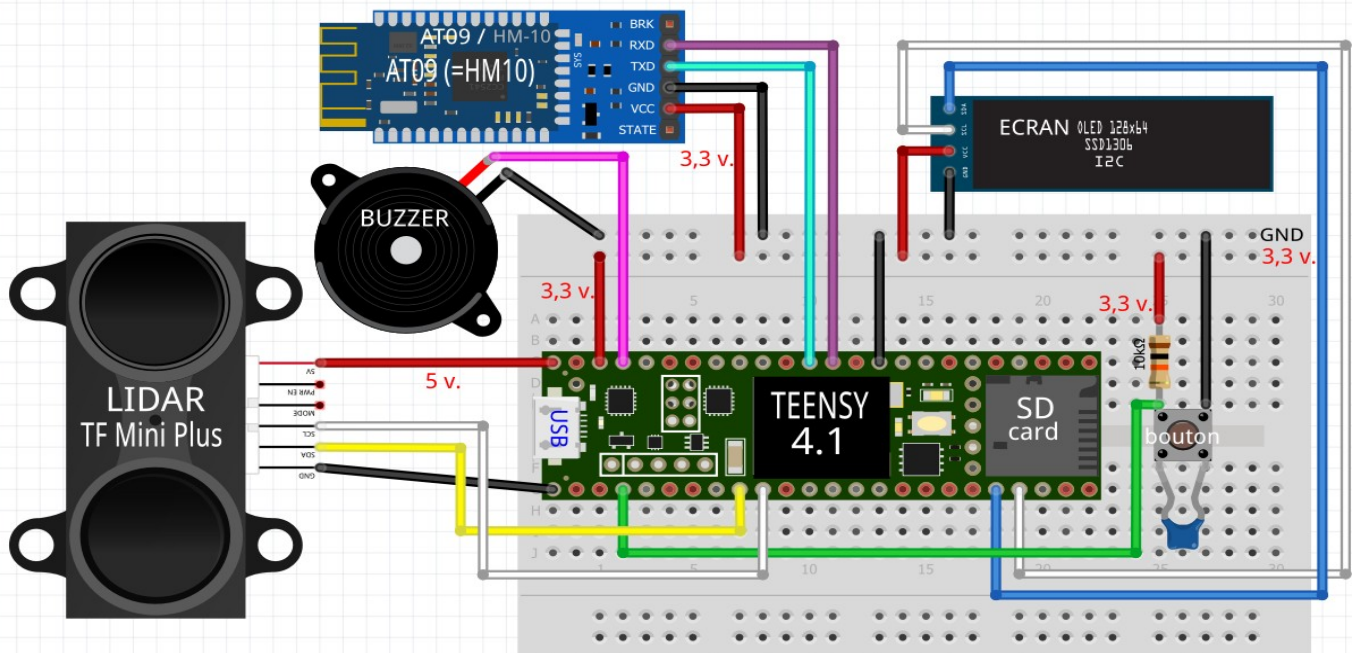


Schéma de branchement des composants des 2 premiers programmes



WiCanne v3-44 Acquis + TestIA

Connections sur la carte Teensy :

pin 23 → Buzzer
pin 7 → RX port série 2 pour lecture Lidar
pin 8 → TX port série 2 pour écriture Lidar
pin 14 → RX port série 3 pour lecture AT09
pin 15 → TX port série 3 pour écriture AT09

pin 18 → SDA Ecran Oled
pin 19 → SCL Ecran Oled
pin 2 → bouton d'activation
pin 13 → Builtin Led

NB : le lidar est alimenté en 5v, les autres composants sont alimentés en 3,3v.

Les programmes peuvent être adaptés à l'emploi d'autres composants comme un lidar d'un autre modèle ou une carte Teensy 4.1 avec plus de mémoire.

Il suffit pour cela de modifier dans le code les 2 constantes du programme spécifiques aux composants utilisés. Et éventuellement changer aussi les paramètres correspondants dans le fichier.

Ces constantes portent le même nom dans les deux programmes concernés (acquisition et exploitation) :

maxmesur = maximum de mesures pouvant être stockées dans un même enregistrement. Ce nombre dépend de la capacité mémoire du second programme pour lequel la capacité disponible est moindre que dans le premier.

freqmax = Fréquence maximale de fonctionnement du Lidar en Hertz. (Limitée à 1000 Hz pour le TF_Mini_Plus)

Le fichier paramètres est un fichier texte nommé « WICAPAR.txt ».

Il ne comporte qu'un seul enregistrement de 26 octets dont le format est le suivant :

`$;eeee;mmmm;ffff;$;dd;cc;$` Soit 8 champs séparés par des point-virgules.

Les caractères « \$ » caractérisent le début, le milieu et la fin de l'enregistrement

Ils servent d'identifiant du format de l'enregistrement.

Pour les 3 premiers paramètres, les valeurs sont toujours des entiers de 4 chiffres, avec des zéros à gauche si nécessaire.

`eeee` est le nombre d'enregistrements à faire par classe (ie : par obstacle).

`mmmm` est le nombre de mesures à effectuer pour un même enregistrement.

`ffff` est la fréquence en Hertz à laquelle on veut faire fonctionner le lidar.

Pour les 2 derniers paramètres, les valeurs sont toujours des entiers de 2 chiffres, avec un zéro à gauche si nécessaire.

`dd` est l'intervalle (en nombre de mesures) entre 2 mesures pour la détection sonore des obstacles

`cc` est le seuil en centimètres de variation de distance pour la détection sonore des obstacles.

Ce fichier paramètre est commun aux 3 programmes.

En son absence, il est créé avec des valeurs par défaut dans le programme d'acquisition des classes d'obstacles mais il peut être aussi créé à la main avec un éditeur de texte.

Il est mis à jour par le programme de détermination expérimentale pour les 2 derniers paramètres ci-dessus.

Exemple de fichier paramètre : `$;0100;0500;0417;$;01;10;$`

Soit 100 enregistrements de 500 mesures à la fréquence de 417 Hz (soit une durée par enregistrement de 1,2 secondes) et comparaison des mesures pour la fonction sonore entre chaque mesure avec une alerte sonore si l'écart de distance entre ces 2 mesures est de plus de 10 cm.

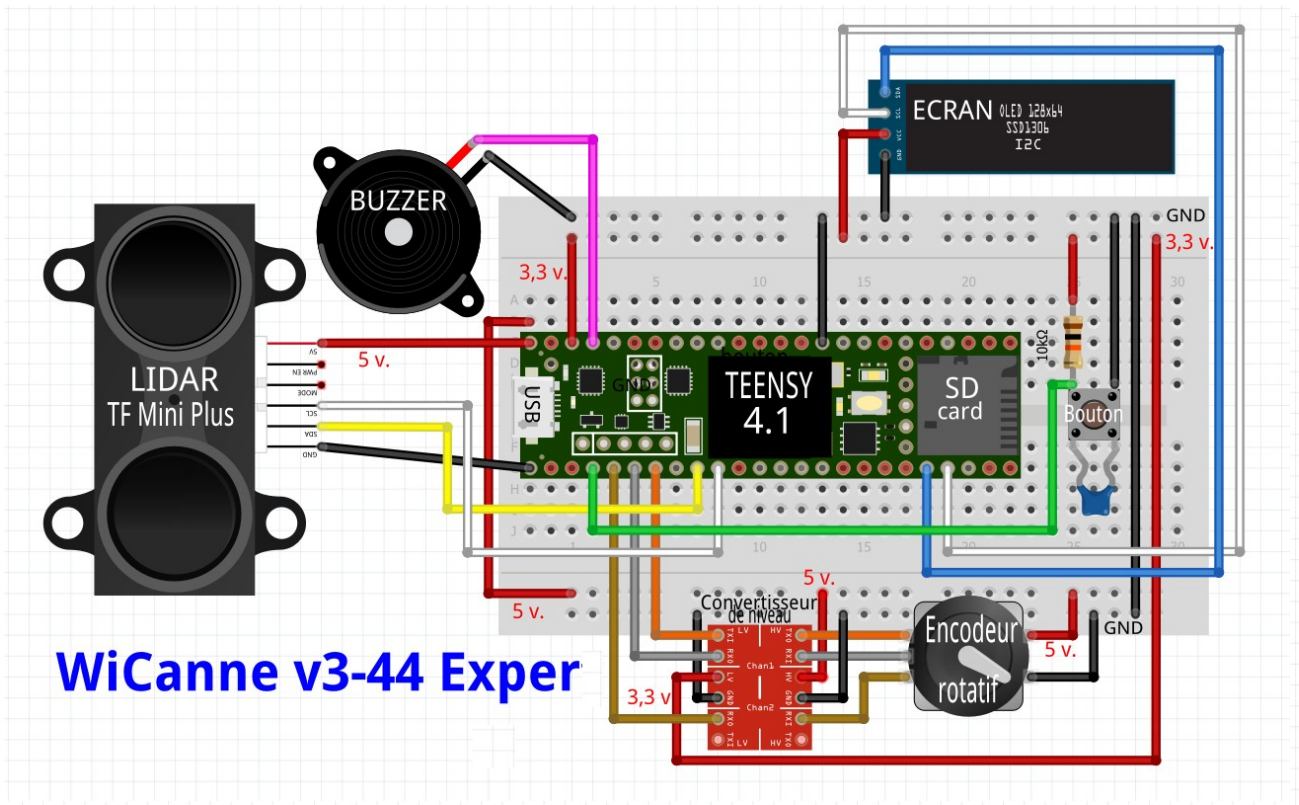
Tous les fichiers de mesures créés dans le programme d'acquisition sont nommés « MESURE n.txt » dans lequel « n » est le numéro d'ordre d'enregistrement du fichier et, par commodité, le numéro de la classe de l'obstacle lors du benchmark.

Ils coexistent pacifiquement avec le fichier paramètres sur la carte SD.

Détermination des paramètres de détection. (Prog. Wi-Canne_v3-44_Exper)

Ce programme est un peu analogue au programme d'utilisation « TestAI » décrit ci-dessus, mais en conservant seulement les fonctions d'exploration sonore de l'environnement.

Le but de celui-ci est de tester de façon empirique, par expérimentations successives, le fonctionnement de la WiCanne en faisant varier les valeurs des 2 paramètres (intervalle de mesures entre 2 mesures prises en compte et seuil minimal de variation de distance à partir duquel on déclenche l'alerte sonore).



WiCanne v3-44 Exper

Connexions sur la carte Teensy :

- pin 2 → bouton de validation
- pin 3 et 4 → rotation droite/gauche de l'encodeur rotatif
- pin 5 → push button de l'encodeur rotatif
- pin 23 → Buzzer
- pin 7 → RX port série 2 pour lecture Lidar
- pin 8 → TX port série 2 pour écriture Lidar
- pin 18 → SDA Ecran Oled
- pin 19 → SCL Ecran Oled
- pin 13 → Builtin Led

NB : l'encodeur rotatif et le lidar sont alimentés en 5v les autres composants sont alimentés en 3,3v.

La plateforme matérielle de ce programme est celle des programmes précédents à laquelle on a ajouté un encodeur rotatif avec appui et sur laquelle on n'utilise pas la carte HM09 Bluetooth. Un convertisseur de niveau logique a été utilisé pour permettre la connexion d'un encodeur fonctionnant sous 5v. avec les entrées de la carte Teesy en 3,3v.

Pour accéder à la modification des paramètres, il faut appuyer une première fois sur le bouton de l'encodeur rotatif.

Cela permet d'accéder à la modification de l'intervalle entre 2 mesures prises en compte en tournant le bouton dans un sens (incréméntation de la variable) ou dans l'autre (décréméntation de la variable).

Un second appui sur le bouton de l'encodeur permet de modifier par rotation du bouton la variable de variation minimale de distance au-delà de laquelle on déclenche un bip sonore.

Un troisième appui sort de la fonction de modification et permet de tester la pertinence des valeurs choisies.

A noter que l'amplitude des valeurs possibles pour ces 2 variables a été arbitrairement limité à une plage de 1 à 20. (1 à 20 mesures d'intervalle et 1 à 20 cm)

Lorsque les tests sont satisfaisants pour l'utilisateur, celui-ci appuie sur le bouton de validation et les paramètres sont alors conservés dans le fichier paramètres sur la carte SD.

YLC→MHK 08/03/2021

