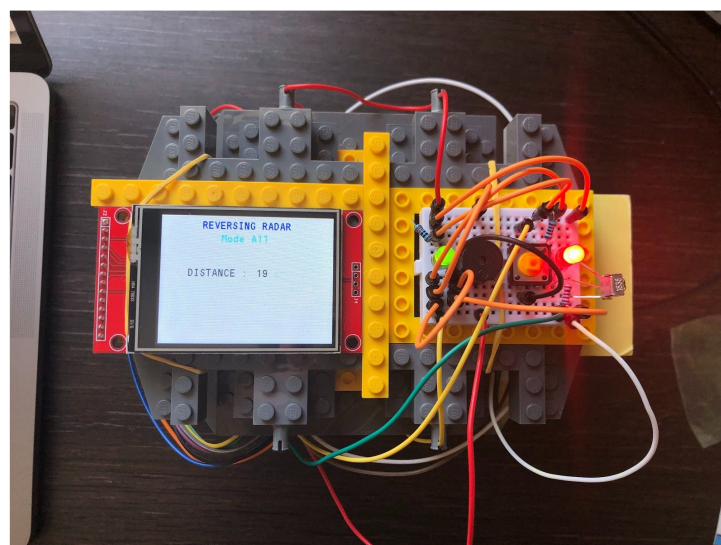
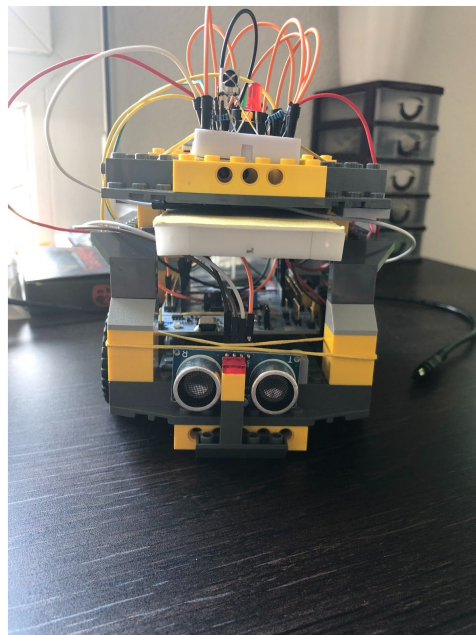


Rapport d'électronique Numérique

REVERSING RADAR B1-S2 (2020)



J'ai choisi pour mon projet d'électronique numérique la réalisation d'un radar de recul comme j'avais déjà pu le réaliser sur un Arduino uno à l'aide d'un écran lcd(2x16). Cette fois je réutiliserai un capteur à ultrasons HC-SR04 mais l'accompagnerai d'un écran TFT ILI9341 et apporterai des fonctionnalités intéressantes.

1 Cahier des charges

Il s'agit ici de développer un radar de recul pouvant être perçu comme une aide au stationnement. Ce radar devra afficher en temps réel la distance à laquelle se trouve le capteur d'un objet via un écran. Le radar sera monté sur des legos pour une approche plus amusante et sympathique. Il sera muni d'un écran tft, d'un capteur à ultrasons, d'une diode, d'un buzzer, d'un récepteur infrarouge et de sa télécommande ainsi que d'un bouton poussoir (et de quelques résistances). Le principe sera de faire clignoter et sonner la led et le buzzer à une fréquence qui augmentera avec la distance pour avertir l'utilisateur qu'il se rapproche d'un obstacle.

Quatre modes de fonctionnement seront alors disponibles, chacun permettant l'activation ou non des périphériques.

- 1 - Mode All : TFT + LED + BUZZER
- 2 - Mode Screen Only : TFT
- 3 - Mode Buzzer Only : TFT + BUZZER
- 4 - Mode Led Only : TFT + LED

On pourra passer d'un mode à l'autre en pressant le bouton poussoir directement sur le radar ou en utilisant la télécommande ir à distance.

Le projet se limite à une utilisation sur une courte distance en vue du capteur HC-SR04 utilisé (les ultrasons étant beaucoup moins efficaces que les OEM sur des longues distances).

Les contraintes sont celles des composants utilisés (mode de fonctionnement/alimentation idéale/possible distance de fonctionnement (portée)/ etc.).

2 Manuel d'utilisation

Alimentation :

Commencez par alimenter la STM32 Nucleo via un câble mini USB que vous pouvez directement brancher entre votre PC (côté USB) et la Nucleo (sur son port mini USB). Vous pouvez également brancher l'extrémité PC de votre USB via un adaptateur sur le secteur, cependant, veillez à ce que la tension de votre adaptateur ne délivre pas plus de 5V (ces informations sont inscrites sur l'adaptateur lui-même).

Démarrage :

Vous devriez constater l'allumage de l'écran et d'une led verte au démarrage. L'écran vous précise alors que vous êtes dans le mode all et affiche la distance.

Fonctionnement :

Il vous est possible de sélectionner un mode parmi les 4 présentés ci-dessous. Il vous suffit de presser le bouton poussoir (jaune); ou un des bouton de la télécommande prévus à cet effet. Ces différents modes vous permettent d'activer ou de désactiver la led ou le buzzer comme bon vous semble :

Mode all : Ecran + Buzzer + LED

Mode screen only : Ecran

Mode buzzer only : Ecran + Buzzer

Mode led only : Ecran + LED

Ainsi, si le buzzer vous agace, vous pouvez le désactiver et continuer de visualiser la distance.

Précautions d'utilisation :

Le tout est monté sur des legos néanmoins, des fils restent accessibles. Si vous n'êtes pas conscient des dangers évitez tout contact avec les composants électroniques. Le dispositif est à tenir hors de portée des enfants. Pour toute information complémentaire, n'hésitez pas à nous contacter par mail.

Contact :

Mail : florentin.lepeltier@reseau.eseo.fr

3 Schéma électrique

Périphérique	Périphérique → Microcontrôleur	Numéro de la broche Nucleo correspondante	Usage
TFT	VCC → +5v/+3v3 GND → GND CS → PA12 Reset → PA9 DC → PA8 MOSI → PA7 SCK → PA5 LED → +3v3 MISO → PA6 T_CS → PA11	POWER(5V) POWER(GND) 6ème broche tout à droite D8 D7 D11 D13 POWER(3V3) D12 7ème broche tout à droite	Active/désactive l'adressage Reset le périphérique Directive current ? (ça sert à quoi ?) Transmission des données (M->S) Synchronise l'échange (clock) Active l'affichage de l'écran Transmission des données (S->M) Pour le tactile ?
HC-SR04	VCC → +5v GND → GND TRIG → PC7 ECHO → PB6	POWER(5V) POWER(GND) D9 D10	Émet l'onde de départ pour mesure Etat haut entre l'émission et la réception (temps de trajet)
BUZZER	PB5	D4	Passer la broche de la nucleo à haut pour activer le buzzer
LED	PA10	D2	Passer la broche de la nucleo à haut pour activer la led
IR RECEIVER	VCC → +5v GND → GND SIGNAL → PB0	POWER(5V) POWER(GND) A3	Passer à l'état bas lorsque qu'une info est reçue (initialement à l'état haut)
BUTTON	VCC → +5v GND → GND SIGNAL → PB3	POWER(5V) POWER(GND) D3	Passer à l'état haut lorsque le bouton est pressé.

A noter qu'une breadboard est nécessaire pour connecter tous les périphériques en même temps (il n'y a qu'une broche 5v/3v3 sur la Nucleo).

4 Structure du programme

Pour développer le projet, j'ai créé les fichiers respectifs « reversing_radar.c » et « reversing_radar.h ». Ces fichiers contiennent l'ensemble des fonctions que j'ai créées hormis la fonction nommée « HCSR04_get_distance » qui a été ajoutée à la librairie (HCSR04.c/h). Elles composent l'ensemble du programme, en voici les fonctions principales :

FICHIER « reversing_radar.c »

Fonction	Description
US_REVERSING_RADAR_init	Initialise les variables et périphériques
US_REVERSING_RADAR_mode_management	Gère le switch entre les différents modes (state machine)
US_REVERSING_RADAR_lcd_display	Gère l'affichage sur TFT en fonction du mode
US_REVERSING_RADAR_process_main	Gère le calcul de distance/fréquence + mode_management
HCSR04_demo_state_machine	Calcule la distance
US_REVERSING_RADAR_compute_frequency	Détermine la fréquence en fonction de la distance
US_REVERSING_RADAR_manage_peripherals	Gère l'activation ou non des périphériques en fonction du mode

Quelques lignes ont également été rajoutées dans le fichier config pour le bouton poussoir.

5 État d'avancement et analyse du projet réalisé

Module/périphérique	Avancement	Pistes d'améliorations
HC-SR04	Le capteur renvoie bien une distance cohérente.	Ce capteur n'est pas d'une très grande précision sur les longues distances (ultrasons moins adaptés...). J'opterai pour un TOF si je devais refaire le projet.
TFT ILI9341 240x320	L'affichage est fonctionnel et permet à l'utilisateur de lire les données reçues par le capteur ainsi que le mode dans lequel il se situe.	L'affichage aurait pu être plus poussé pour une interface plus jolie rendant le projet moins « amateur » (importation d'images etc.) De plus on aurait pu rajouter une interaction entre le tactile de l'écran et l'utilisateur (Pour sélectionner un thème différent par exemple)
Contrôle buzzer/LED	Le buzzer et la led s'activent bien selon le mode sélectionné. Plus on se rapproche plus la fréquence augmente.	Le point négatif est au niveau du timer de l'affichage de la distance qui influe sur la fréquence. C'est comme si la led se "resetait" à chaque affichage...L'effet est donc différent et le son coupé...On est obligé de trouver un équilibre entre les deux et on ne peut pas choisir n'importe quelle fréquence (ou alors le son est "morcelé").

Bouton poussoir 4 broches	Le bouton permet de changer de mode mais affaiblit la luminosité de l'écran lorsqu'il est pressé.	Un moyen aurait sûrement permis d'éviter cette situation. (Brancher le bouton sur une broche d'alim différente de celle du TFT (soit au 3v3) ?)
Switch entre les différents modes	Machine à état fonctionnelle.	L'utilisation d'une variable "entrance" aurait pu être utile (notamment pour l'affichage) mais j'avais déjà contourné une partie du problème autrement...
Récepteur IR DP838/Télécommande	Le mode change lorsque l'on presse un bouton de la télécommande	On pourrait effectuer d'autres actions intéressantes avec la télécommande comme faire déplacer le dispositif.
Design et conception	Le design du dispositif est ok mais un peu encombrant...	J'aurais aimé pouvoir faire rouler le radar à l'aide d'un servomoteur mais la configuration actuelle en lego (axes des roues etc.) ne me permettait pas directement de le faire.

La limite la plus gênante a été côté matériel avec la répartition des broches , notamment au niveau de l'alimentation.(De plus, le SPI du TFT utilise beaucoup de broches). Il a fallu envoyer le 5V et le GND de la Nucleo sur une breadboard annexe.

Je n'ai parfois pas pensé à utiliser les breakpoints qui sont pourtant d'une utilité primordiale lorsque l'on débogue, cela m'aurait permis de résoudre quelques problèmes plus rapidement et surtout seul...

6 Conclusion

La réalisation du projet m'a personnellement permis de bien comprendre la hiérarchisation des libraires, leur interaction ainsi que leur utilité par rapport à la STM32. Au premier abord, on observe des lignes de code dans tous les sens sans réellement saisir leur signification puis on comprend que la HAL et les librairies permettent à l'utilisateur de faire abstraction de la partie hardware (comme son nom l'indique) et donc plus technique de la Nucleo.

Ainsi cela facilite grandement le développement d'un quelconque projet où le développeur n'a par exemple nul besoin de se soucier de quel registre de périphérique modifier pour changer l'état d'une broche. Il utilise une fonction générique qui le fait à sa place. Cependant, même si ce contenu nous simplifie le travail, je pense qu'il reste essentiel et nécessaire d'allier la compréhension avec l'utilisation de ces fonctions afin d'exploiter correctement le matériel.