



INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
RENNES

Systèmes à microprocesseurs

Capteur à Ultrasons

3ESET

Selima Sahraoui

Interface Capteur à Ultrasons

PRESENTATION

Le projet traite de l'utilisation des GPIOs, des Timers et des interruptions.

- Utilisation d'une méthode par **échantillonnage**
- Utilisation d'un **timer**
- Mise en place d'un '**IRQ Handler**' (routine d'interruption)

Matériel

• Le simulateur keil sera utilisé avec la carte STM32F446 et le capteur US HC-SR04 en salle de TP

I. Système de gestion du capteur à ultrasons à mettre en œuvre :

a. Description du système

Les capteurs à ultrasons peuvent être utilisés pour détecter la présence d'objets devant celui-ci. Le système est composé de deux transducteurs à ultrasons comprenant un émetteur et un récepteur et de deux modules électroniques permettant de générer le signal fourni à l'émetteur ultrasons et de traiter le signal issu du récepteur ultrasons. Le synoptique du système est présenté à la Figure 1.

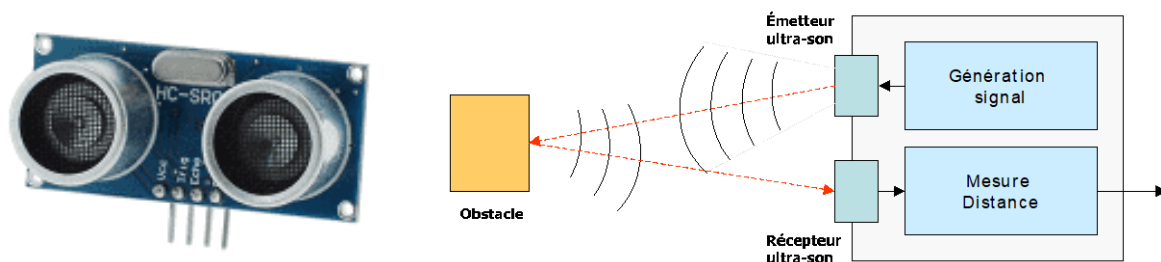


Figure 1 : Synoptique du système de mesure de distance par ultrason

Le transducteur à ultrasons émet à intervalles réguliers un *burst* (i.e. une salve) d'impulsions à 40KHz. Ces impulsions se propagent dans l'air à la vitesse du son (340 m/s). Lorsque les ondes sonores rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au récepteur ultrasons. La distance d entre les transducteurs à ultrasons et la cible est calculée à partir du temps, T_{dist} , écoulé entre l'émission du signal et la réception de l'écho. L'onde sonore ayant parcouru la distance aller et retour, la distance d est égale à :

$$d = \frac{T_{dist} * 340}{2} \quad (m/s)$$

Pour transducteur à ultrasons, nous utilisons le module HC-SR04. Celui-ci est composé d'un signal de commande Trig et d'un signal de sortie Echo permettant de mesurer la distance de l'obstacle. Le chronogramme des signaux en entrée et en sortie du module HC-SR04 est présenté à la Figure 2. Une impulsion est envoyée sur l'entrée Trig pour déclencher la génération d'un burst d'impulsion. Cette impulsion doit avoir une durée T_{gen} supérieure à 10 μs . Ensuite, sur la sortie Echo, est présent un signal

binaire dont la durée T_{dist} à l'état haut est proportionnelle à la distance de l'obstacle. La durée de T_{dist} est comprise entre 0.15 ms et 25 ms. Nous fixons la durée d'un cycle à 50 ms.

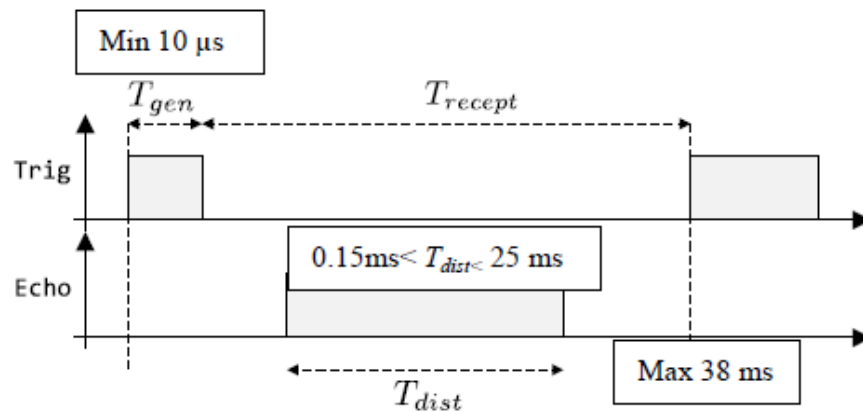


Figure2: Chronogramme des signaux en entrée et en sortie du module HC-SR04

Afin d'étudier les interruptions, nous allons utiliser une méthode basée sur un échantillonnage périodique du signal d'Echo.

- 1- A l'instant t_0 , l'entrée Trig est mise à 1 et à l'instant t_1 , l'entrée Trig est mise à 0, comme illustré à la Figure 3.
- 2- Ensuite, dans l'intervalle de temps t_1 et $t_1 + T_{recept}$, l'objectif est d'évaluer la durée du front haut T_{dist} . Pour cela, à chaque interruption, donc tous les intervalles de temps T_e , le signal Echo est échantillonné. Suivant la valeur du signal Echo, vous serez en mesure de calculer combien de temps le signal Echo reste à 1.
- 3- Une fonction computeDistance, est appelée à l'instant $t_1 + T_{recept}$ et cette fonction calcule la distance à partir de la durée du front haut du signal Echo.

La distance de l'obstacle en cm est obtenue à l'aide de l'équation suivante en considérant la durée T_{dist} en μs :

$$d(cm) = \frac{T_{dist}(\mu s)}{58,8235}$$

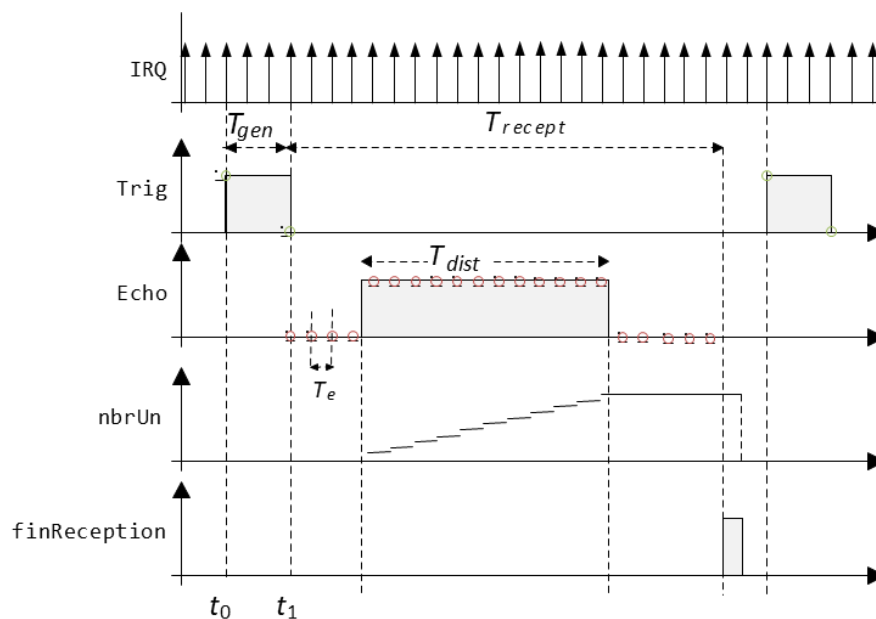


Figure 3: Chronogramme des signaux Trig et Echo après échantillonnage

II. Développement du code de gestion du capteur à ultrasons sur carte STM32F446RE

Etape 1: poser le contexte

Dans une introduction vous proposez une interprétation des objectifs du TP et de l'approche que vous allez adopter. C'est le même système que celui étudié en TD .

Etape 2 : initialiser les GPIOs et vérifier la configuration

Développer le code permettant l'initialisation des GPIOs pour la carte **STM32F446RE** de la façon suivante et vérifier la configuration. Vous configurerez les broches de la façon suivante :

- PA1 : utilisé pour contrôler la période d'échantillonnage
- PA5: utilisé pour contrôler la période d'interruption
- PA12: sortie numérique pour le signal Trig
- PA8: entrée numérique pour le signal Echo

En mode debug, grâce au menu Peripherals , vous pourrez vérifier la bonne configuration de vos registres. Vous pourrez également utiliser le logic port pour vérifier le comportement de vos GPIOs.

Etape3:

Développer le code permettant l'initialisation du Timer2 pour qu'il génère une interruption de débordement à la fréquence d'échantillonnage choisie $T_e = 100\mu s$, en considérant que Timer input clock : TIMx_CLK = 180MHz.

Etape 4:

La base de temps du timer 2 étant configurée, vous pouvez désormais autoriser une interruption de débordement et valider la fréquence d'interruption en observant PA5 .
Comment procédez-vous ?

Etape 5:

Une fois votre interruption fonctionnelle, vous développerez les deux fonctions décrites ci-dessous :
UltraSoundMgt () qui sera la fonction au cœur de la routine d'interruption et ComputeDistance() qui sera appelée quand c'est nécessaire dans la fonction main.
Valider le fonctionnement global de votre code avec votre capteur.

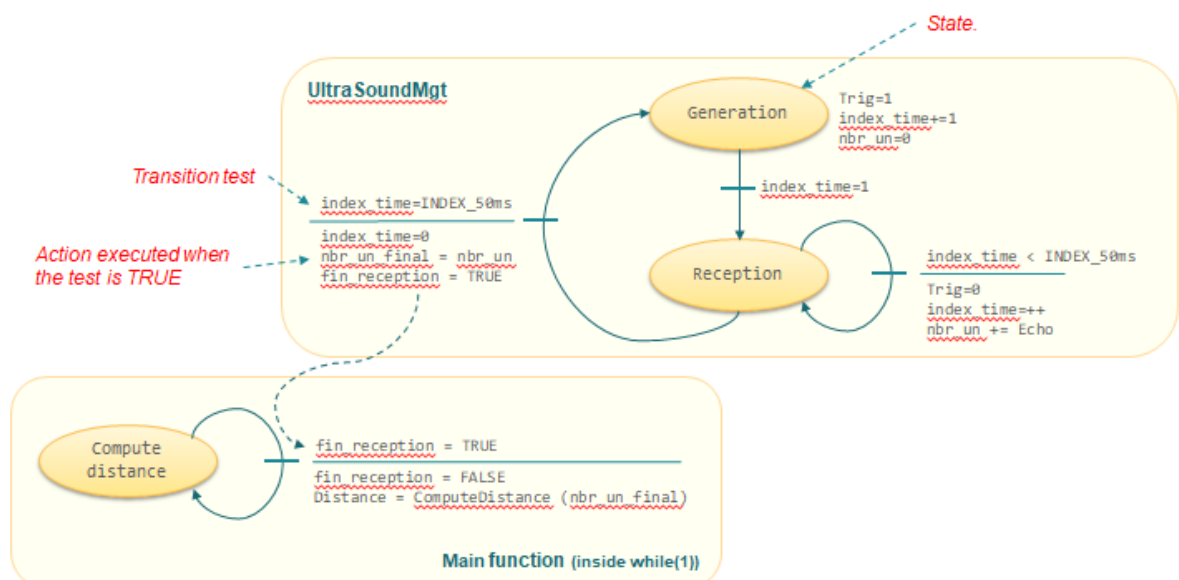


Figure 4 : Proposition de machine d'état du système

Spécification UltraSoundmgt		
Au cœur de la routine d'interruption activée par le timer à une fréquence de $1/T_e$		
Entrées		
Echo		Valeur binaire du signal Echo du module ultrasons
Entrées/Sorties		
finReception	unsigned int (variable globale)	Booléen de synchronisation qui permettra de lancer le calcul de la distance dans la fonction main
Sorties		
Trig		Valeur binaire positionnée sur l'entrée Trig du module ultrasons présent
nbrUn	unsigned int (variable globale)	Nombre de périodes T_e . pour lesquelles le signal Echo est à 1
Fonctions		
		Gestion du temps - incrémentant d'une variable globale indexTime tout au long de l'intervalle de temps T_{gen} - Génération de la valeur sur la broche Trig du module ultrasons pendant l'intervalle de temps T_{recept} - Lecture de la valeur de la broche Echo du module ultrasons et mise à jour adéquate de la variable nbrUn

Les spécifications de la fonction de calcul de la distance sont présentées ci-dessous

Spécification ComputeDistance		
Fonction appelée à chaque positionnement de la variable finReception à 1		
Entrée		
nbrUn	unsigned int (variable globale)	Nombre de périodes T_e . pour lesquelles le signal Echo est à 1
Sorties		
obstacleDistance		Distance de l'obstacle (basée sur le nbrUn et la période d'échantillonnage)
Fonctions		
		Calcule la distance en cm à partir de la mesure temporelle de l'état haut du signal echo.

III. *Rappels*

a. Configuration basique d'un timer

1. Enable clock for selected TIM in RCC_APBxENR register
2. Set up the timer so that it starts counting
 - Set the Prescaler value PSC
 - Set the Auto-reload value ARR

In our context, the source clock is set as $Apb1BusClock = Apb2BusClock = 180Mhz$.

The frequency of internal timer clock is $Apb1BusClock / (PSC.+1)$

The frequency of occurrence of the IT is $Apb1BusClock / ((PSC.+1) \times ARR)$

3. When all is set, you can enable the TIM counter in CR1 register.

b. Configuration des interruptions

- 1) Enable interrupt at processor level (NVIC functions). You can check the correct interrupt ID at the top of stm32f446xx.h file
- 2) Enable and configure interrupt at peripheral level (in our case in TIMx_DIER register)
- 3) Overwrite the corresponding handler function with your interrupt subroutine. Check the name of the handler in startup_stm32f446xx.s.
In this subroutine, you need to clear the flag of the pending interrupt at processor level and peripheral level. (NVIC function and TIMx_SR).

INSA de Rennes

20 Avenue des Buttes de Coësmes
CS 70839
35708 Rennes Cedex 7

Tél. +33 (0) 2 23 23 82 00
Fax +33 (0) 2 23 23 83 96

www.insa-rennes.fr

INSA

