



UNIVERSITÉ PARIS 8 - VINCENNES SAINT-DENIS

Master Informatique des Systèmes Embarqués

Sonar ROVIN

Guillaume VILAIRE

Etudiant : 16710314

UE EMB3MICC : Microcontrôleurs, FPGA Bus de terrain

Table des matières

Introduction	2
1 Matériels	3
1.0.1 Microcontrôleur	3
1.0.2 Modules	4
1.0.3 Connection	6
2 Implémentation des modules	7
2.0.1 Carte ROVIN	7
2.0.2 Capteur Ultra son	7
2.0.3 Servo Moteur	9
2.0.4 Recepteur IR	10
2.0.5 LCD	11
3 Implémentation Graphique	12
3.0.1 Liaison serie	12
3.0.2 Interprétation des données	12
4 Annexe	15
4.0.1 Code source	15

Introduction

Dans ce rapport, je présente un système de type sonar mis au point avec un module ROVIN. Il sera décrit tout au long de ce rapport le matériels et les solutions utilisés pour répondre au besoin de la création d'un sonar.

La présentation de ce rapport est découpé en deux parties. Une première partie consistera à présenter la partie matériels du sonar, la seconde partie présentera la partie graphique qui est déporté sur un ordinateur de type Windows ou OSX.

Le sonar réagira une fois que l'utilisateur aura envoyer un évènement au micro-contrôleur via une télécommande de type infra-rouge. Le résultat sera affiché sur un d'ordinateur qui sera relié via un port serial RS-232 coté micro-contrôleur et USB coté Ordinateur.

Chapitre 1

Matériels

1.0.1 Microcontrôleur

Le sonar est basé sur un "super" module microcontrôlé compact appeler "ROVIN" équipé d'un processeur ARM 32 bits. Il intègre son propre système d'exploitation développé par Comfile Technology. Il dispose de son propre compilateur "C" avec un environnement de développement IDE intégrant un éditeur, un compilateur et un débogueur.

Afin de faciliter le développement du sonar, nous utilisons la platine "Quick Start Board" adapté au module ROVIN. Cette plaque est équipé d'une plaque de connexion sans soudure afin d'apposer les connecteurs des différents modules utilisé pour la fabrication de notre sonar.

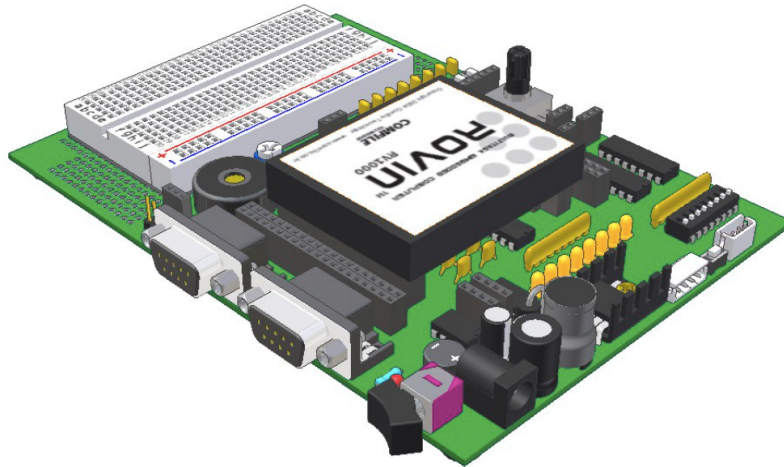


FIGURE 1.1 – Module ROVIN sur sa carte Quick Start Board.

1.0.2 Modules

Afin de détecter une distance, nous utilisons un module "capteur à ultrason" de type SRF05. Nous utiliserons 4 pins pour relier ce module à notre carte ROVIN. 2 pins dédiés à l'alimentation en 5V, et 1 pin de type input (Trigger) et 1 pin de type output (Echo)



FIGURE 1.2 – Module SRF05.

La détection 360 degrés se fait via un servo moteur sur lequel sera fixé le précédent module "SRF05". Nous utiliserons un moteur "HSR-1425CR" qui est un servo moteur à rotation continue et piloter via un signal PWM. Il se relie via 3 pins à notre carte ROVIN, 2 pins pour l'alimentation et 1 pin pour le signal à envoyer au moteur.



FIGURE 1.3 – Moteur HSR-1425CR.

Pour lancé une capture, le sonar attend un évènement qui sera déclencher via un contrôle infra rouge. Le module est relié via 8 pin de type digital, et 1 pin TX afin d'envoyer un signal permettant au ROVIN de lire les 8 pin digital.

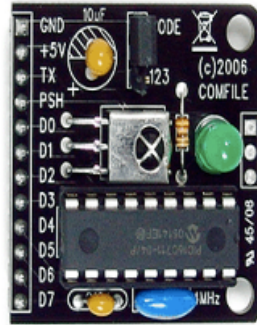


FIGURE 1.4 – IR REMOCON RX

Le sonar sera équipé d'un écran LCD "OTM 854 B-W 1" afin d'afficher l'état du sonar. Celui-ci permettra de spécifier a l'utilisateur la bonne réception de la commande envoyer par la télécommande infra-rouge en affichant si une commande est en cours d'exécution. La communication entre le LCD et la carte ROVIN est faite via une liaison serie UART.



FIGURE 1.5 – LCD OTM 854 B-W 1.

1.0.3 Connection

Afin de communiquer entre la carte ROVIN et les différents module, la mise en place d'un câblage a été réaliser. La figure suivante montre le détails de ce dernier.

PIN ROVIN	NOM PIN	NOM MODULE	PIN MODULE
10	PA7	IR REMOCON RX	D7
11	PA6	IR REMOCON RX	D6
12	PA5	IR REMOCON RX	D5
13	PA4	IR REMOCON RX	D4
14	PA3	IR REMOCON RX	D3
15	PA2	IR REMOCON RX	D2
16	PA1	IR REMOCON RX	D1
17	PA0	IR REMOCON RX	D0
21	EXPB4	IR REMOCON RX	TX
22	EXPB3	HSR-1425CR	PWM
70	EXPD4	SRF05	ECHO (Output)
80	PD3	SRF05	TRIGGER (Input)
ALCD	ALCD	LCD OTM 854 BW	PC_LEVEL

FIGURE 1.6 – Communication ROVIN / MODULES.

Chapitre 2

Implémentation des modules

Dans ce chapitre, nous décrirons les méthodes utiliser afin de piloter les modules connecté. On ce concentrera sur l'implémentation de la carte ROVIN sans parler de la partie interface graphique qui sera décrit dans le chapitre suivant.

2.0.1 Carte ROVIN

Comme décrit dans le chapitre précédent, la carte ROVIN est notre carte maîtresse, puisqu'elle permet de contrôler l'ensemble des modules connecté, et des communiqué en même temps avec l'ordinateur.

2.0.2 Capteur Ultra son

Pour capter une distance, le module SRF05 attend un signal sur sa broche "Trigger", une fois le signal envoyer, un ultrason est émis, il réfléchira sur un obstacle qui renverra le son vers le module. En fonction du temps que l'ultrason mettra pour revenir au module, nous pouvons déterminé une distance fiable. Les figures suivantes illustrent un cycle de détection de distance.

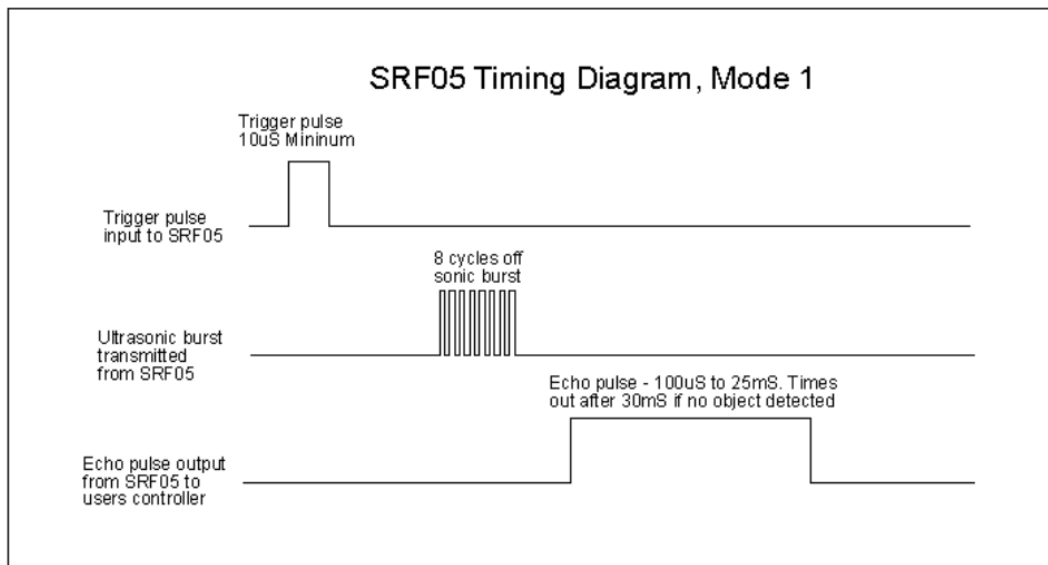


FIGURE 2.1 – Cycle d’une détection de distance.

```

1  void main(void)
2  ▼ {
3      // Initialisation des ports du ROVIN
4      PPI_Init();
5      EXPORT_Init();
6
7      //On envoi un trigger au sonar
8      //(Etat Haut, on attend, Etat Bas)
9      PPI_BitOut(PD,0,1);
10     Delay(0x15);
11     PPI_BitOut(PD,0,0);
12     //On lit la data du sonar
13     CaptureRead(0, &CP_LOW_TIME, &CP_HIGH_TIME);
14     //On écrit sur le port serie la data
15     UartWrShort(1, CP_HIGH_TIME);
16 }
17
18
19

```

Code Source

utilisation SRF05.

2.0.3 Servo Moteur

Le servo moteur hsr1425cr attend un signal de type pwm afin de tourner dans un sens ou dans un autre. L'IDE de la carte ROVIN possède une librairie "RC-SERVO" qui simplifie l'utilisation des servo moteur. La figure suivante montre une utilisation basique d'un servomoteur avec la librairie incluse.

```
1  void main(void)
2  {
3      //Initialise un servo moteur
4      //On configure la PIN PWM3 en mode RC-SERVO
5      //La periode du servo est de 15ms
6      Pwm3_Set(PWM_RCSERVO,15);
7
8      //Utilise le servo N°3
9      //Determine la valeur rapport cyclique position 0° -> 0x600
10     //Determine la position extrême -> 180°
11     //Determine la valeur rapport cyclique position 180° -> 0x1390
12     RcServo_Regulate(3, 0x600, 180, 0x1390);
13
14     //On autorise l'activation sur signal PWM pour le servo
15     Pwm3_On(PWM_CHA);
16
17     //On déplace le servo sur la position 90°
18     RcServo_Move(3, 90);
19     Delay(0xffff);
20
21     //On déplace le servo sur la position 180°
22     RcServo_Move(3, 180);
23     Delay(0xffff);
24
25     //On déplace le servo sur la position 0°
26     RcServo_Move(3, 90);
27     Delay(0xffff*2);
28
29     //On descative le signal PWM pour le servo
30     Pwm3_Off(PWM_CHA)
31 }
```

Code Source utilisation HSR-1425CR.

2.0.4 Recepteur IR

Lors d'une commande effectuée avec une télécommande infrarouge le récepteur infrarouge, envoie un signal a la carte ROVIN gérer par évènement. Lors de l'évènement, si ce dernier correspond a l'évènement associer a la réception d'une commande, on lit les données reçu par le récepteur. Le code source suivant montre une capture une fois l'évènement déclenché. La figure suivante montre l'implémentation.

```
1  void IRQ_EVENT_(void) {
2  unsigned char iMSG;
3  iMSG=GetMsg();
4
5  switch(iMSG) {
6  case MSG_EXINT4:
7      //On stock la donnée transmise par le recepteur
8      receive = PPI_In(PA);
9      //On affiche dans la console de debug la donnée
10     DebugCHAR(receive, DEC);
11     break;
12     }
13 }
14
```

Code Source

utilisation Recepteur IR.

2.0.5 LCD

L'écran LCD est contrôlé avec la bibliothèque de l'IDE du ROVIN. Son utilisation est relativement simple. Il suffit de spécifier une ligne ainsi qu'une colonne pour écrire à un endroit voulu sur le LCD. La figure suivante montre comment écrire sur ce dernier avec la bibliothèque intégrée.

```
1  void main(void)
2  ▼ {
3      //Gestion LCD
4      //On utilise l'UART 0 en spécifiant le BAUD
5      UartSetBaud(0,115200);
6      UartOn(0);
7      //On initialise le LCD UART 0
8      AlcdInit(0);
9      //On efface les données afficher du LCD UART 0
10     AlcdClear(0);
11     //On spécifie au LCD UART 0 la ligne 0 et la colonne 5
12     AlcdLocate(0,5,0);
13     //On écrit sur le LCD UART 0 le message "En attente"
14     AlcdPrint(0,"En attente");
15     //On spécifie au LCD UART 0 la ligne 3 et la colonne 1
16     AlcdLocate(0, 3, 1);
17     //On écrit sur le LCD UART 0 le message d'une commande"
18     AlcdPrint(0, "d'une commande");
19 }
```

FIGURE 2.2 – Cycle d'une détection de distance.

Chapitre 3

Implémentation Graphique

L'implémentation graphique a été réalisée avec le langage de programmation Python et sa librairie tkinter permettant de réaliser des interfaces graphiques pratiques et simples. L'interface réagit lorsque la carte ROVIN transmet des données par port série. Elle récupère deux données, la première correspond à l'angle du servo-moteur, et la seconde à la distance reçue par le capteur à ultrason.

3.0.1 Liaison série

Afin d'interpréter les données reçues par le port série, il est nécessaire de traiter les données reçues. Avant de pouvoir traiter ses données pour interpréter graphiquement le résultat des données, on stocke les deux valeurs. La première s'appellera "i" qui correspondra à l'angle du servo moteur. Et la seconde "distance" qui porte bien son nom pour stocker la distance lue par le capteur ultrason. Pour lire ces données, on effectue une boucle infinie tant que le port série transmet sur le portcom de l'ordinateur. Ainsi nous enregistrerons toutes les données transmises.

3.0.2 Interprétation des données

Lors d'un balayage, la carte ROVIN transmet au total 90 fois deux valeurs. On sait donc que notre interface graphique devra avoir un pas de 4 afin de remplir le cercle graphique. On applique une suite de calculs afin de pouvoir interpréter graphiquement les données reçues :

Dans un premier temps on doit multiplier "i" par 4 pour avoir une étendu de "0" à 360". Ensuite on transforme notre position en degrés donc "i" en radians pour pouvoir calculer les positions "x" et "y" de notre droite représentant la direction de l'angle. A partir de la on peu calculer les positions "x" et "y". La figure suivante représente les calcul détaillé précédemment en python.

```
1  rad = math.radians(i*4)
2  x = math.cos(rad) * distance + 300
3  y = math.sin(rad) * distance + 300
```

FIGURE 3.1 – Formule d'interprétation des données en python.

Une fois la formule appliqué on dessine sur notre sonar graphique un point rouge pour chaque distance calculer. La figure suivante représente l'interface graphique du sonar. En partant du milieu, chaque cercle représente une distance de 50cm. Le milieu représente donc une distance de 0 et le dernier cercle représente une distance de 3 mètres.

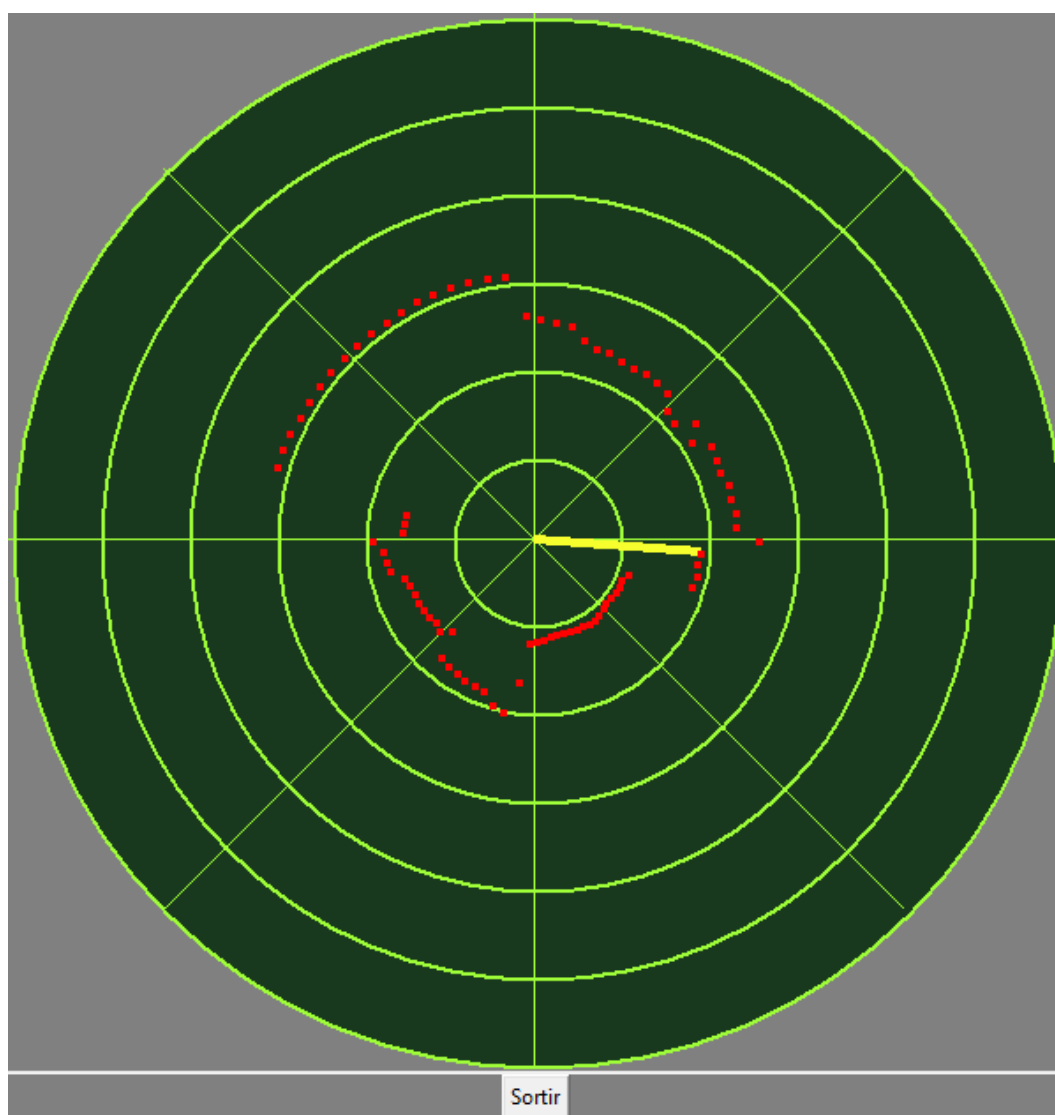


FIGURE 3.2 – Interface graphique.

Chapitre 4

Annexe

4.0.1 Code source

L'ensemble du code source est disponible sur GitHub à l'adresse suivante :
[https ://github.com/Junior77/SonarRovin](https://github.com/Junior77/SonarRovin)