

Rapport du TerrorBot

Vincent Buisset
Julien Faidide

Description du projet

Le TerrorBot avance lentement en ligne droite avec les leds avant allumées et s'arrête dès qu'il touche un obstacle. Il recule jusqu'à sa position de départ en faisant clignoter ses feux arrières. Ensuite, il fonce rapidement tout droit avec les leds avant allumées ! Va-t'il s'écraser contre l'obstacle ? Et non ! Il s'arrête juste avant de toucher l'obstacle. Quel effroi !

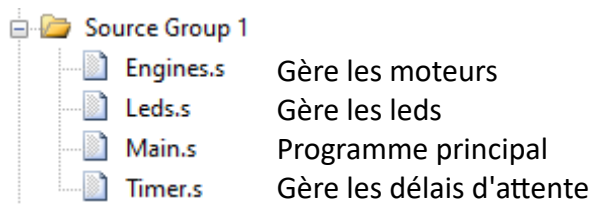
Vidéo du programme

<https://youtu.be/UHmo6leS6oE>

Différents scénarios

Le TerrorBot ne fonctionne que dans un seul scénario, il avance jusqu'à toucher un mur, la distance avec le mur peut varier.

Architecture du projet



Utilisation des registres

; Register usage :	
; r0: Utils	Utilisé pour des calculs
; r1: Utils	Utilisé pour des calculs
; r2: Unused	Inutilisé
; r3: Unused	Inutilisé
; r4: Bumper 1	Adresse de l'état des bumper
; r5: Leds address	Adresse de l'état des leds
; r6: Engines address	Adresse de l'état des moteurs
; r7: Unused	Inutilisé
; r8: Used by Timers.s	Utilisé pour les délais d'attentes dans Timer.s
; r9: Time calculation	Utilisé pour des calculs avec le temps

Engines.s : Gestion des moteurs

Fichier fourni dans le cours, comporte des sous programmes pour simplifier l'utilisation des moteurs. La seule modification apportée est l'ajout d'un sous programme MOTEUR_SET_SPEED_R0, qui permet d'assigner la vitesse des moteurs avec la valeur du registre r0.

Timer.s : Gestion des moteurs

Sous programmes servant à gerer des temps d'attentes au sein du projet. Actuellement, ce fichier ne comporte que 1 seul sous programme : WAIT_R8. Celui-ci contient une simple boucle, qui décrémente R8 jusqu'a 0 avant de se stopper. Pour generer un temps d'attente, il suffit donc d'assigner R8 avec le temps voulu et d'appeler WAIT_R8.

Leds.s : Gestion des leds

Comporte des sous programmes pour faciliter l'utilisation des leds.

Choix des GPIO

```
; Broches select
LEDS_PIN_INIT      EQU      0x3C      ; 00111100 mask of all used leds
                                   ; (internet leds only activates if bit 3 & 4 are 0)
LEDS_FORWARD_PIN   EQU      0x3C      ; 00111100 led1 & led2 on pin 5 & 6
LEDS_BACKWARD_PIN  EQU      0x00      ; 00000000 internet1 & internet2 on pin 3 & 4 when 0

LEDS_ALL_PIN       EQU      0x30      ; 00110000 Active all leds (Forward & backward)
LEDS_STOP_PIN      EQU      0x0C      ; 00001100 Stop all leds

;GPIO_O_DR2R : The GPIODR2R register is the 2-mA drive control register; By default, all GPIO pins have 2-mA drive.

GPIO_PORTF_BASE    EQU      0x40025000 ; GPIO Port F (APB) base: 0x4002.5000 (p416 datasheet de lm3s9B92.pdf)
GPIO_O_DIR          EQU      0x00000400 ; GPIO Direction (p417 datasheet de lm3s9B92.pdf)
GPIO_O_DR2R         EQU      0x00000500 ; GPIO 2-mA Drive Select (p428 datasheet de lm3s9B92.pdf)
GPIO_O_DEN          EQU      0x0000051C ; GPIO Digital Enable (p437 datasheet de lm3s9B92.pdf)
```

LEDS_PIN_INIT : Masque des leds à utiliser
LEDS_FORWARD_PIN : valeur d'allumage des leds avant
LEDS_BACKWARD_PIN : valeur d'allumage des leds arriere
LEDS_ALL_PIN : valeur d'allumage avant & arriere
LEDS_STOP_PIN: valeur pour éteindre toute les leds

GPIO_PORTF_BASE: Adresse du port F
GPIO_O_DIR : Adresse du Pin 1 sur le port F
GPIO_O_DR2R : Adresse e selection de l'intensité (2mA) sur le port F
GPIO_O_DEN : Adresse des fonctions digitales sur le port F

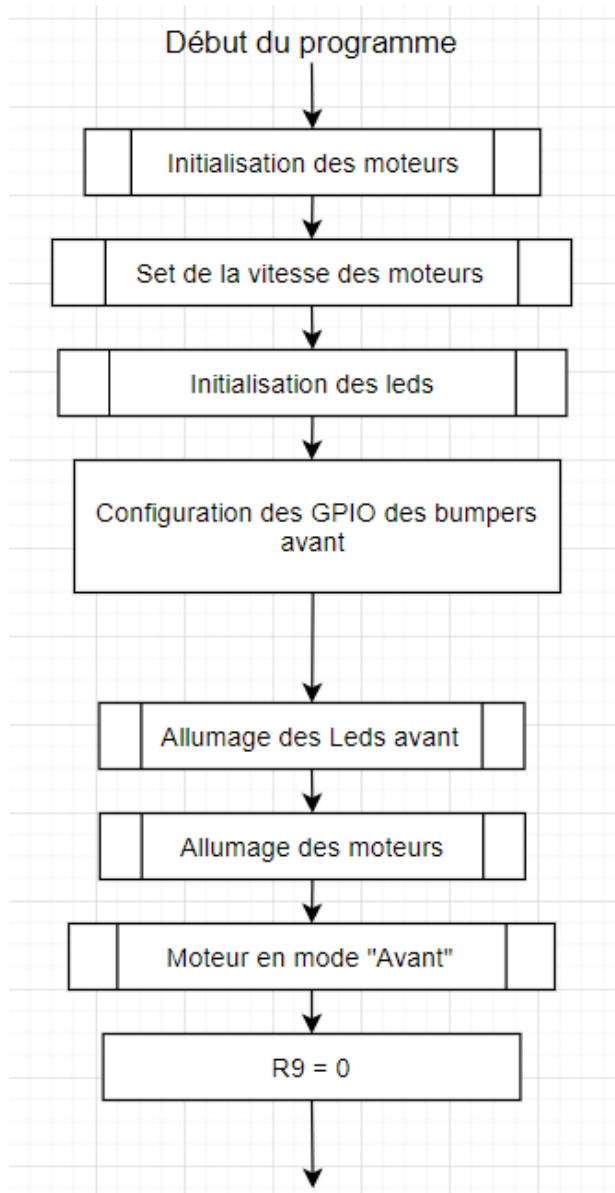
Sous programmes

EXPORT	LEDS_INIT	Initialise les GPIO
EXPORT	LEDS_FORWARD_ON	Allume les leds avant
EXPORT	LEDS_BACKWARD_ON	Allume les leds arriere
EXPORT	LEDS_ON	Allume toute les leds
EXPORT	LEDS_OFF	Eteind toute les leds
EXPORT	LEDS_BACKWARD_INVERT	Inverse l'état des leds arriere, utile pour le clignotement

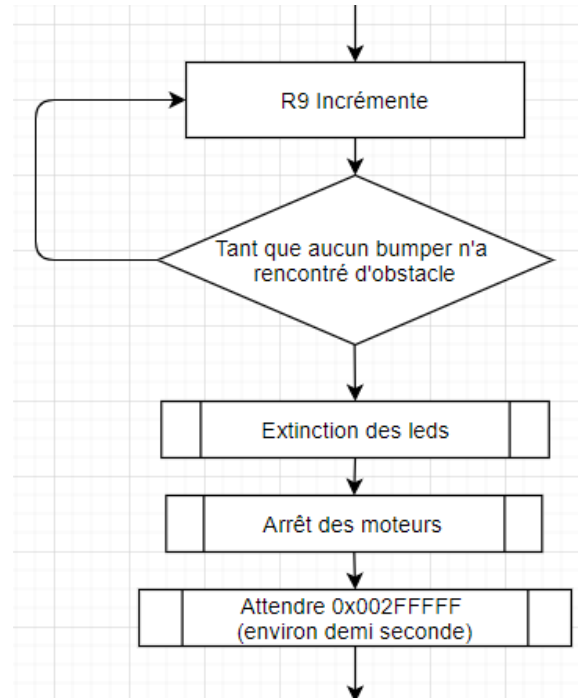
Main.s : Programme principal

Programme qui utilise Engine.s, Leds.s et Timer.s afin de réaliser le comportement en description du projet. Le programme peut etre découpé en 4 parties :

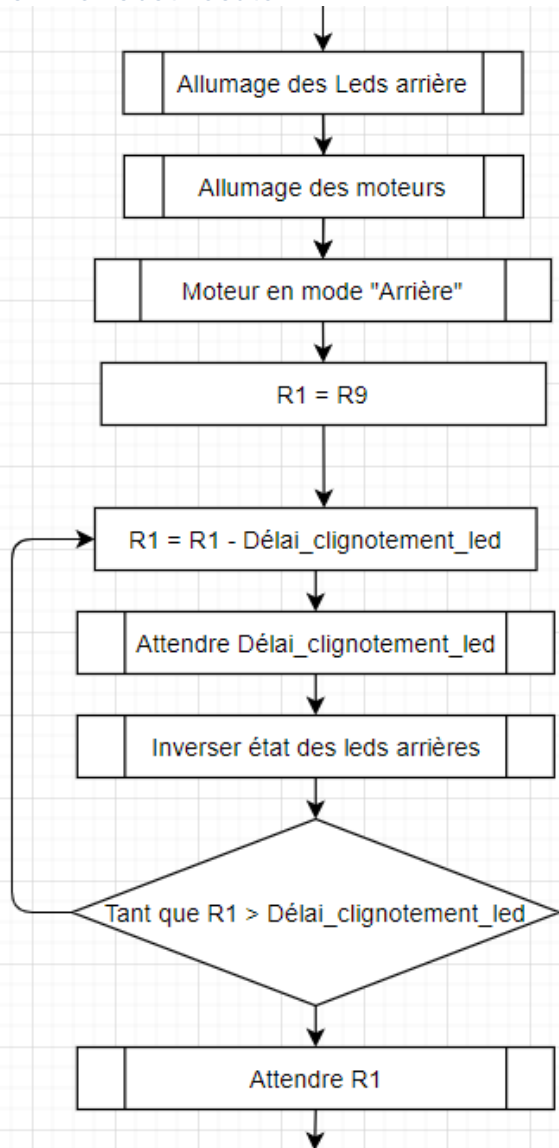
1. Initialisation



2. Le robot avance



3. Le robot recule



4. Le robot fonce !

