

# Sommaire

<b>Sommaire .....</b>	<b>1</b>
<b>1) Présentation .....</b>	<b>2</b>
<b>2) Constitution du robot .....</b>	<b>2</b>
<b>3) Carte de gestion .....</b>	<b>3</b>
3.1) Alimentation .....	3
3.2) Commande des moteurs .....	3
3.3) Suivie de ligne.....	3
3.4) Capteur de proximité.....	4
3.6) Capteur de lumière .....	4
3.5) Interface sonore .....	4
3.6) Microcontrôleur .....	4
3.7) Extensions .....	5
<b>4) Bibliothèque logicielle .....</b>	<b>6</b>
4.1) Simulation du robot .....	6
4.2) Commande des moteurs .....	6
4.2.1) Forward ( power ) .....	6
4.2.2) Reverse ( power ) .....	6
4.2.3) Stop ( ).....	6
4.2.4) SpinLeft ( power ).....	6
4.2.5) SpinRight ( power ).....	6
4.2.6) SetMotors ( left_power, right_power ) .....	7
4.3) Commande des LED.....	7
4.3.1) WriteLEDs ( byte_out ) .....	7
4.3.2) LEDOn ( LED ) .....	7
4.3.3) LEDOff ( LED ) .....	7
4.4) Gestion bouton poussoir .....	8
4.4.1) ReadSwitch ( Switch ) .....	8
4.4.2) WaitForSwitch ( Switch ) .....	8
4.5) Production d'un son et gestion microphone .....	8
4.5.1) PlayNote ( note, delay_ms ) .....	8
4.5.2) ReadMic ( ) .....	8
4.6) Lumière ambiante.....	9
4.6.1) ReadLDR ( ) .....	9
4.7) Suivi de ligne.....	9
4.7.1) ReadLineSensor ( sensor ) .....	9
4.8) Capteurs IR de proximité .....	9
4.8.1) CheckIR ( sensor ) .....	9
4.8.2) ReadIRSensor ( sensor ) .....	10
4.9) Component Properties.....	11
4.9.1) Wait for Button .....	11
4.9.2) Return Value .....	11
4.9.3) Threshold Values.....	11

## 1) Présentation

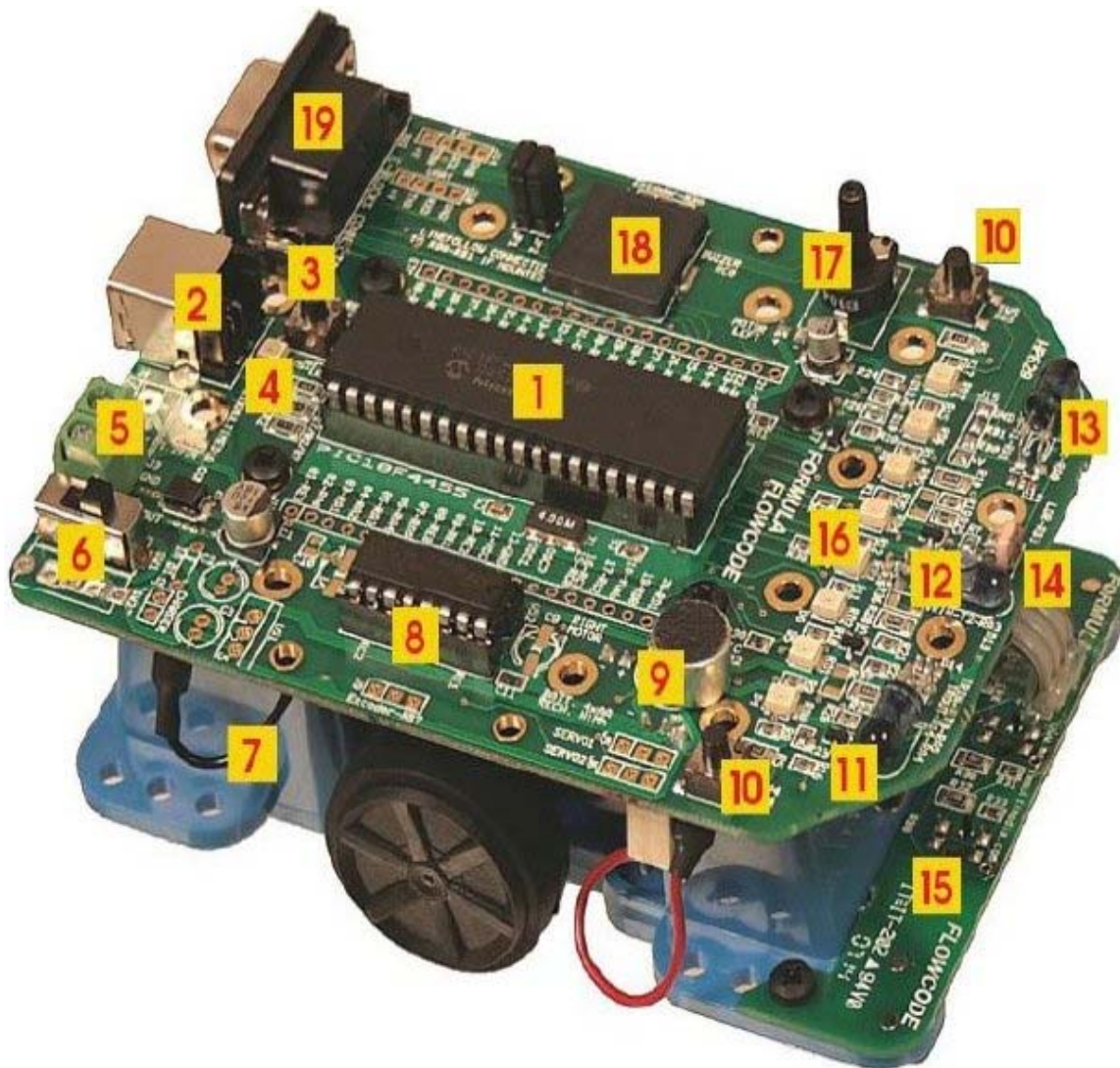
Le robot FORMULA permet de s'initier à la programmation d'un robot. Cette programmation est pour l'essentiel graphique et se fait par la saisie d'ordinogrammes avec le logiciel FlowCode. Ce logiciel produit un fichier qui est téléchargé dans le robot par une liaison USB.

Le robot peut se déplacer grâce à deux moteurs il est pourvu de nombreux capteurs qui permettent d'appréhender son environnement.

Sa programmation par le logiciel FlowCode est facilitée par une bibliothèque permettant la gestion des moteurs et des capteurs.

Ce document propose une l'analyse succincte des structures électroniques et la présentation de la bibliothèque logicielle du robot avec le texte original et sa traduction.

## 2) Constitution du robot

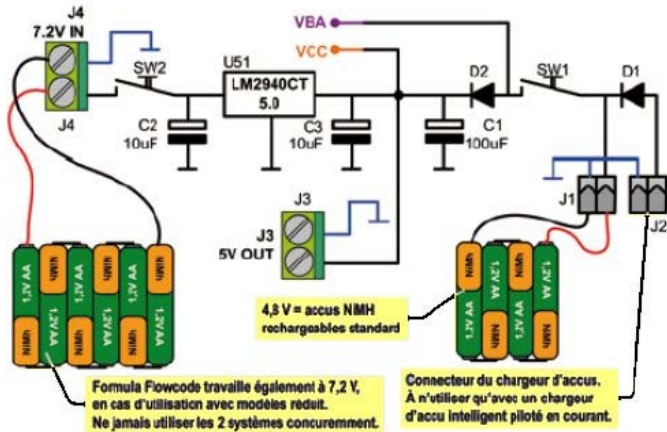


- |  |   |
|--|---|
| 1. Le cerveau du robot : une puce PIC18F4455                 | 11. Capteur de proximité droit                            |
| 2. Connecteur USB de programmation                           | 12. Capteur de proximité central                          |
| 3. Bouton de remise à zéro                                   | 13. Capteur de proximité gauche                           |
| 4. LED d'indication de programmation                         | 14. Capteur de lumière                                    |
| 5. Alimentation externe 5 V                                  | 15. Module de suivi de ligne                              |
| 6. Interrupteur général                                      | 16. 8 LED à fonction définie par l'utilisateur            |
| 7. Châssis, compartiment batterie, moteurs, réducteur, roues | 17. Potentiomètre de réglage de sensibilité du microphone |
| 8. Circuit de commande des moteurs : L293D                   | 18. Haut-parleur  |
| 9. Microphone avec circuit amplificateur de niveau           | 19. Connecteur DB9 d'extension pour E-Blocks              |
| 10. Poussoirs à fonction définie par l'utilisateur           |   |



### 3) Carte de gestion

#### 3.1) Alimentation



Sous le châssis du robot se trouvent deux logements pour 4 piles type AA ou 4 accus. Ces derniers pourront être rechargés par le connecteur J2. L'interrupteur SW1 permet la mise en service pour l'alimentation par piles ou accus.

⚡ Il faudra utiliser un chargeur compatible avec les accus et non une simple alimentation secteur car le robot n'assure pas la gestion de la charge.

Il est possible d'alimenter le robot en 7,2V par le connecteur J4. Dans ce cas on utilise le régulateur de tension U51 pour obtenir une tension stable de 5V nécessaire au fonctionnement du  $\mu C$ . Cette tension est disponible sur le connecteur J3. L'interrupteur SW2 permet la mise en service pour cette alimentation externe.

La diode D2 évite que le régulateur U51 ne débite du courant dans les accus ou piles 4,8V. La diode D1 permet la charge des accus sans retour dans le chargeur.

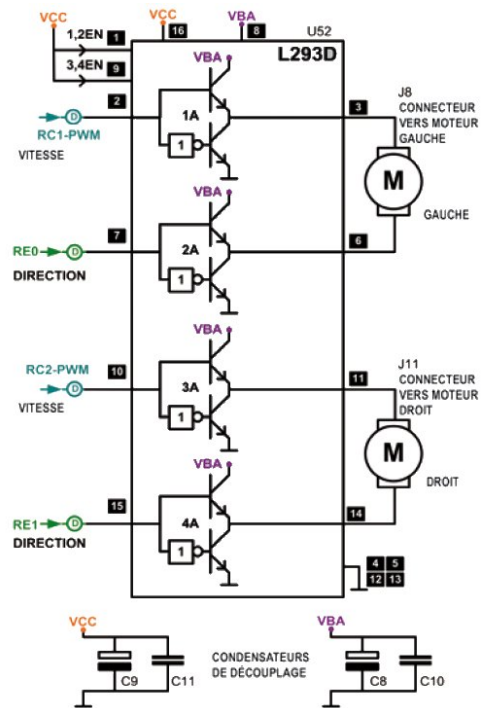
#### 3.2) Commande des moteurs

Le robot se déplace grâce à deux roues motrices indépendantes. Un réducteur de vitesse est placé entre chaque roue et son moteur.

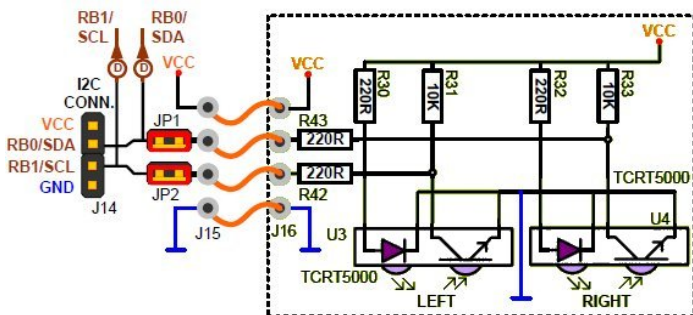
Un circuit L293D assure l'interface entre le  $\mu C$  et les moteurs.

Ce circuit est un double pont en H qui va permettre :

- d'inverser la tension d'alimentation des moteurs et donc le sens de rotation.
- de faire varier cette tension d'alimentation et donc la vitesse. C'est un signal PWM qui permet cette variation.



#### 3.3) Suivre de ligne

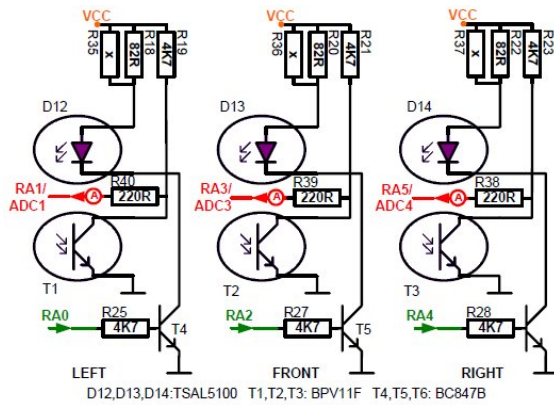


Le suivi de ligne est réalisé grâce à deux modules qui captent la lumière réfléchiée par le sol.

Les résistances R30 et R32 limitent l'intensité dans les LED émettrices. R33 et R31 sont des résistances de pullup.

⚡ Les cavaliers JP1 et JP2 devront être retirés si l'on désire utiliser l'interface I2C du  $\mu C$ . Les signaux SCL et SDA de l'I2C ainsi que l'alimentation 5V sont disponibles sur le connecteur J14.

### 3.4) Capteur de proximité



Les trois capteurs ont un fonctionnement identique.  
Une diode LED (D12, D13, D14) activée par un transistor (T4, T5, T6) produit un signal IR capté par un phototransistor (T1, T2, T3).

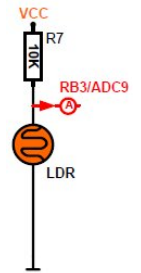
Ce phototransistor polarisé par une résistance (R19, R21, R23) permet d'obtenir au point A une différence de potentiel proportionnelle à la distance.

La ddp au point A est traitée par un CAN intégré au  $\mu C$ .

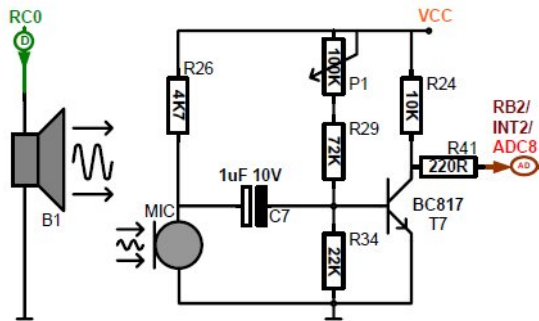
### 3.6) Capteur de lumière

Pour capter la lumière ambiante on utilise une LDR ( Light Dependent Résistor ).

La résistance R7 et la LDR forment un pont diviseur dont la tension de sortie sera traitée par un CAN intégré au  $\mu C$ .



### 3.5) Interface sonore



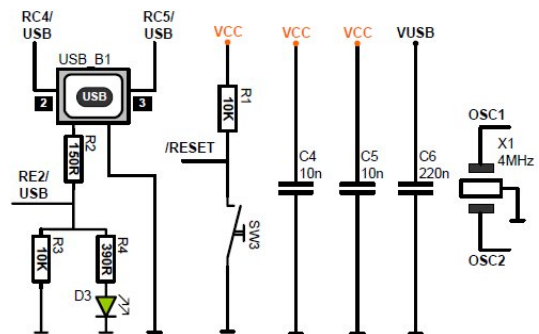
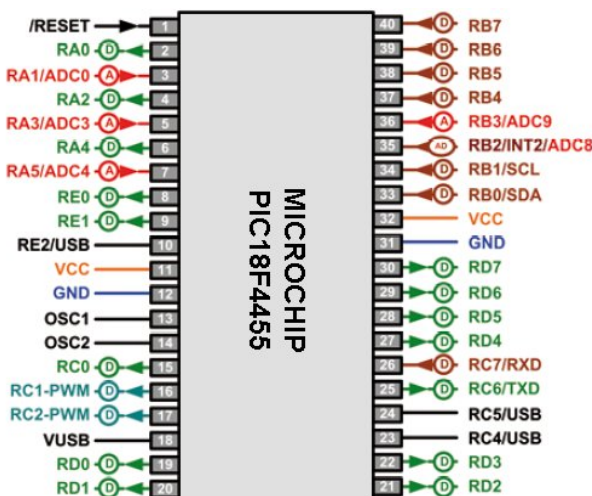
La reproduction des sons est réalisée par un haut parleur piézoélectrique.

Pour capter les sons on utilise un microphone electret alimenté par la résistance R26.

C'est le transistor T7 qui assure l'amplification du signal qui sera traité par un CAN intégré au  $\mu C$ .

Le potentiomètre P1 permet de régler la sensibilité de l'ensemble.

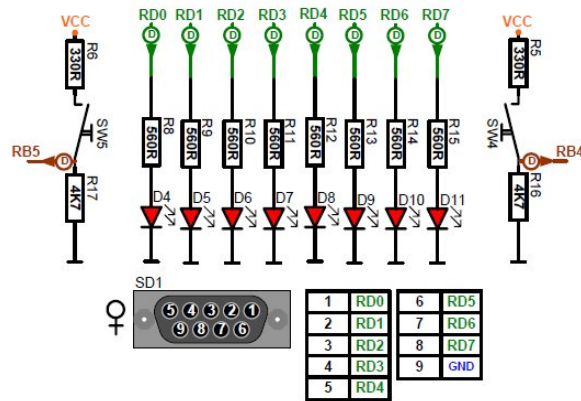
### 3.6) Microcontrôleur



Le  $\mu C$  est un PC18F4455 de Microchip, en boîtier DIL 40 broches. Il dispose d'une interface USB qui permet la programmation du robot.

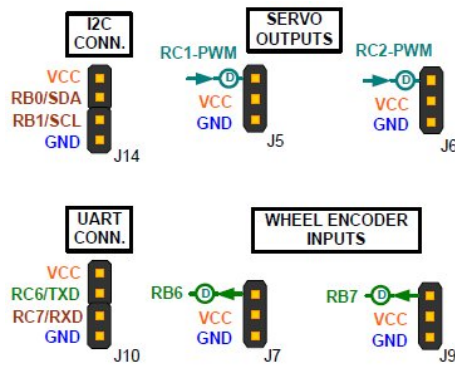
Son horloge est réalisée par le résonateur X1 ( 4MHz ).

### 3.7) Extensions



C'est le port D (RD0..RD7) qui assure la commande des LED, les boutons poussoir SW4 et SW5 sont connectés sur le port B en RB4 et RB5.

Le port D est disponible sur le connecteur DB9 d'extension pour E-Blocks. RD0 = pin 1 à RD7 = pin 8 masse = pin 9.



- Le connecteur J10 permet d'utiliser la liaison série asynchrone.
- Le connecteur J14 permet d'utiliser la liaison I2C.  
⚡ Les cavaliers JP1 et JP2 devront être retirés si l'on désire utiliser l'interface I2C du  $\mu C$  la fonction suivie de ligne ne sera plus disponible.
- Les connecteurs J4 et J5 permettront la commande de servomécanismes.  
⚡ Les signaux PWM RC1 et RC2 sont aussi utilisés pour les moteurs.
- Sur J7 et J9 pourront être connectés des encodeur pour les roues.

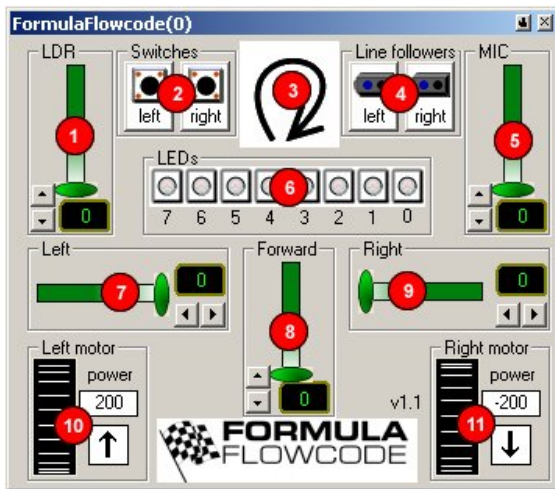
Notes

## 4) Bibliothèque logicielle



L'icône ci contre active la simulation du robot Formula et permet de disposer de macros spécifiques.

### 4.1) Simulation du robot



1. Capteur de luminosité ( LDR ).
2. Etat des deux boutons poussoir.
3. Direction prise par le robot..
4. Etat des capteurs de suivi de ligne.
5. Simulation du microphone.
6. Etat des 8 LEDS.
7. Simulation du capteur de proximité gauche.
8. Simulation du capteur de proximité avant.
9. Simulation du capteur de proximité droit.
10. Indication de la direction et de la vitesse du moteur gauche.
11. Indication de la direction et de la vitesse du moteur droit.

### 4.2) Commande des moteurs

#### 4.2.1) Forward ( power )

Drives the Micromouse forwards. A valid power input can be between 0 and 255 where 0 is minimum and 255 is maximum power. The motors should start turning after approx 40% power.

Cette macro permet de commander les moteurs en avant. Le paramètre **power** aura pour valeur 0 = minimum et 255 = maximum. Toutefois les moteurs ne tournent qu'a partir de 40% de la valeur de **power**.

#### 4.2.2) Reverse ( power )

Drives the Micromouse backwards. A valid power input can be between 0 and 255 where 0 is minimum and 255 is maximum power. The motors should start turning after approx 40% power.

Cette macro permet de commander les moteurs en arrière. Le paramètre **power** aura pour valeur 0 = minimum et 255 = maximum. Toutefois les moteurs ne tournent qu'a partir de 40% de la valeur de **power**.

#### 4.2.3) Stop ( )

Stops the motors and reduces the overall operating current therefore increasing the battery life.

Cette macro permet l'arrêt des moteurs, elle ne nécessite aucun paramètre.

#### 4.2.4) SpinLeft ( power )

Spins the Micromouse anti-clockwise on the spot. A valid power input can be between 0 and 255 where 0 is minimum and 255 is maximum power. The motors should start turning after approx 40% power.

Cette macro permet de faire tourner le robot sur lui même dans le sens inverse des aiguilles d'une montre. Le paramètre **power** aura pour valeur 0 = minimum et 255 = maximum. Toutefois les moteurs ne tournent qu'a partir de 40% de la valeur de **power**.

① Le moteur droit est commandé avec la valeur **power**, le moteur gauche avec la valeur **-power**.

#### 4.2.5) SpinRight ( power )

Spins the Micromouse clockwise on the spot. A valid power input can be between 0 and 255 where 0 is minimum and 255 is maximum power. The motors should start turning after approx 40% power.

Cette macro permet de faire tourner le robot sur lui même dans le sens des aiguilles d'une montre. Le paramètre **power** aura pour valeur 0 = minimum et 255 = maximum. Toutefois les moteurs ne tournent qu'a partir de 40% de la valeur de **power**.

① Le moteur gauche est commandé avec la valeur **power**, le moteur droit avec la valeur **-power**.



## 4.2.6) SetMotors ( left\_power, right\_power )

Allows for full control over both motors. Power inputs (left\_power and right\_power) control the power going to the motors. A valid power input can be between -255 and 255 where -255 is maximum reverse, 0 is minimum and 255 is maximum forward power. The motors should start turning after approx 40% power in either direction.

Cette macro permet de commander les deux moteurs. Elle nécessite deux paramètres, **left\_power** et **right\_power** pour le moteur gauche et droit. Les valeurs sont comprises entre -255 et 255, la valeur négative permet la marche arrière, la valeur 0 arrête le moteur. Toutefois les moteurs ne tournent qu'à partir de 40% de la valeur de x\_power.

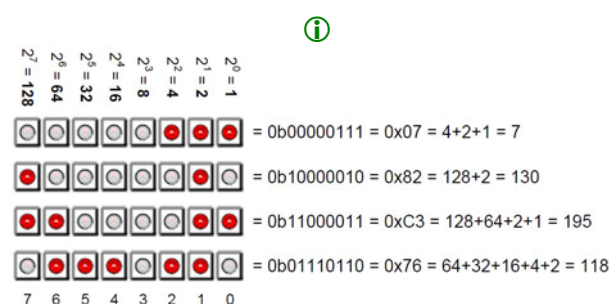
① Cette macro pourra être utilisée pour obtenir des trajectoires circulaires.

## 4.3) Commande des LED

### 4.3.1) WriteLEDs ( byte\_out )

Outputs a data byte onto the LEDs. Can also be used to send a byte to the E-Blocks port.

Cette macro permet d'envoyer un octet pour l'allumage des 8 LED. Les LED étant connectées au port D du  $\mu C$  cette macro peut aussi être utilisée pour envoyer un octet sur ce port disponible sur le connecteur E-Blocks c'est à dire le connecteur SD1 de type DB9.



Binaire	Hexadécimal	Décimal
WriteLEDs ( 0b00000111 )	WriteLEDs ( 0x07 )	WriteLEDs ( 7 )
WriteLEDs ( 0b10000010 )	WriteLEDs ( 0x82 )	WriteLEDs ( 130 )
WriteLEDs ( 0b11000011 )	WriteLEDs ( 0xC3 )	WriteLEDs ( 195 )
WriteLEDs ( 0b01110110 )	WriteLEDs ( 0x76 )	WriteLEDs ( 118 )

### 4.3.2) LEDOn ( LED )

Allows for a single LED to be switched on. If the LED equals 0 then the macro will turn on the right hand LED (LSb). If the LED equals 7 then the macro will turn on the left hand LED (MSb).

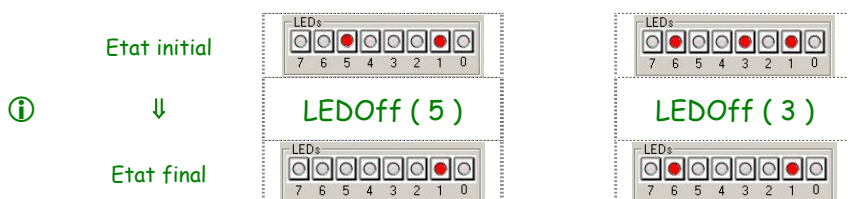
Cette macro permet l'allumage d'une seule LED. Le paramètre **LED**, entre 0 et 7, désigne la LED qui doit être allumée. Le "0" est le poids faible (LSb) à droite, le "7" est le poids fort (MSb) à gauche.



### 4.3.3) LEDOff ( LED )

Allows for a single LED to be switched off. If the LED equals 0 then the macro will turn off the right hand LED (LSb). If the LED equals 7 then the macro will turn off the left hand LED (MSb).

Cette macro permet l'extinction d'une seule LED. Le paramètre **LED**, entre 0 et 7, désigne la LED qui doit être éteinte. Le "0" est le poids faible (LSb) à droite, le "7" est le poids fort (MSb) à gauche.





## 4.4) Gestion bouton poussoir

### 4.4.1) ReadSwitch ( Switch )

Returns the value of one of the switches at the front of the Micromouse. If the switch input equals 'L' or 'l' or 0 then the left switch will be read. If the switch input equals 'R' or 'r' or 1 then the right switch will be read. This macro returns a 1 for a pressed switch and a 0 for a released switch.

Cette macro retourne une valeur indiquant l'état du bouton poussoir sélectionné par le paramètre **Switch** comme indiqué dans le tableau ci dessous.

Switch égal ...		Lecture bouton poussoir ...
'L' ou 'l' ou '0'	⇒	gauche
'R' ou 'r' ou '1'		droit

La valeur retournée doit être ainsi interprétée :

Valeur retournée ...	Bouton poussoir :
0	relâché
1	pressé

### 4.4.2) WaitForSwitch ( Switch )

Waits forever for one of the switches at the front of the Micromouse to be pressed. If the switch input equals 'L' or 'l' or 0 then the Micromouse will wait for the left switch to be pressed and released. If the switch input equals 'R' or 'r' or 1 then the Micromouse will wait for the left switch to be pressed and released.

Cette macro attend que le bouton poussoir désigné par la paramètre **Switch** soit appuyé puis relâché.

Switch égal ...		Attente bouton poussoir ...
'L' ou 'l' ou '0'	⇒	gauche
'R' ou 'r' ou '1'		droit

## 4.5) Production d'un son et gestion microphone

### 4.5.1) PlayNote ( note, delay\_ms )

Outputs a note via the piezo sounder. Note can be between 0 and 200 and will play for between 0 and 1000 milliseconds.

Cette macro permet de produire une note via le haut parleur.

Paramètre	Valeur du paramètre	Action
<b>note</b>	0 ... 255	hauteur de la note
<b>delay_ms</b>	0 ... 1000	durée de la note 0 à 1000ms

Le tableau ci dessous donne pour la valeur de **note** ( 0...223 ) la note de musique produite.

G	G#	A	A#	B	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C	C#
SOL	SOL#	LA	LA#	SI	DO	DO#	RE	RE#	MI	FA	FA#	SOL	SOL#	LA	LA#	SI	DO	DO#
0	14	28	40	53	64	75	85	94	103	112	120	127	135	141	148	154	159	165

D	D#	E	F	F#	G	G#	A	A#	B	C	C#	D	D#	E	F	F#	G
RE	RE#	MI	FA	FA#	SOL	SOL#	LA	LA#	SI	DO	DO#	RE	RE#	MI	FA	FA#	SOL
170	175	179	183	187	191	195	198	201	204	207	210	212	215	217	219	221	223

### 4.5.2) ReadMic ( )

Returns the Microphone reading in 8 bit byte form ( 0 - 255 ) or 10 bit int form ( 0 - 1023 ) depending on the return value property. The potentiometer P1 can be used to adjust the sensitivity of the Microphone.

Cette macro retourne une valeur numérique proportionnelle à l'amplitude du signal capté par le micro. Cette valeur peut être un octet ( 0...255 ) ou un entier ( 0...1023 ) suivant la valeur définie dans les propriétés ( voir § 4.9 ).

Le potentiomètre P1 permet de régler la sensibilité du microphone.

## 4.6) Lumière ambiante

### 4.6.1) ReadLDR ( )

Returns the Light Dependant Resistor reading in 8 bit byte form (0 - 255) or 10 bit int form (0 - 1023) depending on the return value property. 0 represents 100% light saturation and 255 / 1023 represents 100% darkness.

Cette macro retourne la valeur numérique de la luminosité captée par la LDR. Cette valeur peut être un octet (0..255) ou un entier (0..1023) suivant la valeur définie dans les propriétés ( voir § 4.9 ). La valeur 0 (zéro) indique une luminosité maximale, la valeur 255 ou 1023 indique l'obscurité.

## 4.7) Suivi de ligne

### 4.7.1) ReadLineSensor ( sensor )

Returns the value of one of the digital line sensors.

If the sensor input is 'L' or 'l' or 0 then the value returned will be the status of the left sensor.

If the sensor input is 'R' or 'r' or 1 then the value returned will be the status of the right sensor.

The sensor readings are a 0 for a white surface and a 1 for a black surface.

Cette macro retourne l'information du capteur de suivi de ligne désigné par la paramètre **sensor** comme indiqué dans le tableau ci dessous.

Sensor égal ...	⇒	Suivi de ligne ...
'L' ou 'l' ou '0'		gauche
'R' ou 'r' ou '1'		droit

Valeur retournée ...	Capteur sur surface :
0	blanche
1	noire

## 4.8) Capteurs IR de proximité

### 4.8.1) CheckIR ( sensor )

Returns a boolean value representing the IR sensor reading compared to the threshold value. (See Threshold Properties)

If the sensor input equals 'L' or 'l' or 0 then the left IR sensor will be checked.

If the sensor input equals 'F' or 'f' or 1 then the front IR sensor will be checked.

If the sensor input equals 'R' or 'r' or 2 then the right IR sensor will be checked.

If the sensor reading is above the threshold value i.e. there is no obstacle present then the function returns a 0.

If the sensor reading is below the threshold value i.e. there is an obstacle present then the function returns a 1.

Cette macro retourne une valeur Booléenne (0..1) qui est le résultat de la comparaison du signal du capteur IR avec la valeur du seuil fixée dans les propriétés ( voir § 4.9 ). La sélection du capteur est fixée par le paramètre **sensor** comme indiqué dans le tableau ci dessous.

sensor égal ...	⇒	Capteur IR
'L' ou 'l' ou '0'		gauche
'F' ou 'f' ou '1'		avant
'R' ou 'r' ou '2'		droit

La valeur retournée doit être ainsi interprétée :

Valeur retournée ...	Valeur / seuil	Interprétation
0	Valeur capteur > seuil	pas d'obstacle détecté
1	Valeur capteur < seuil	obstacle détecté

## 4.8.2) ReadIRSensor ( sensor )

Returns the Infra Red Proximity sensor reading in 8 bit byte form (0 - 255) or 10 bit int form (0 - 1023) depending on the return value property.

If the sensor input equals 'L' or 'l' or 0 then the left IR sensor will be read.

If the sensor input equals 'F' or 'f' or 1 then the front IR sensor will be read.

If the sensor input equals 'R' or 'r' or 2 then the right IR sensor will be read.

[Click here for a graph of the IR response](#)

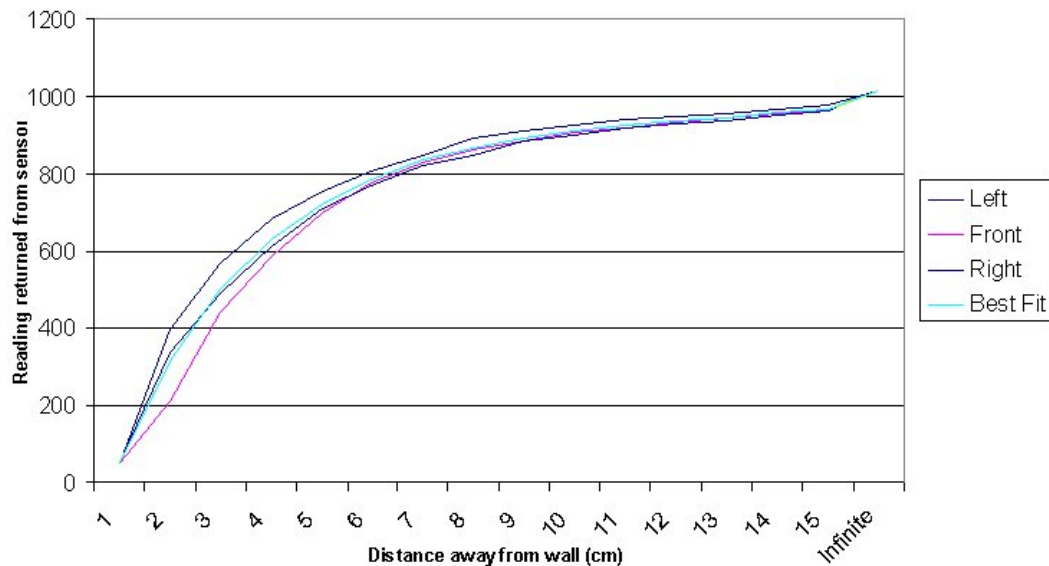
Cette macro retourne la valeur numérique du signal du capteur de proximité. Cette valeur peut être un octet ( 0...255 ) ou un entier ( 0...1023 ) suivant la valeur définie dans les propriétés ( voir § 4.9 ).

sensor égal . . .	Capteur de proximité
'L' ou 'l' ou '0'	Gauche
'F' ou 'f' ou '1'	avant
'R' ou 'r' ou '2'	Droit

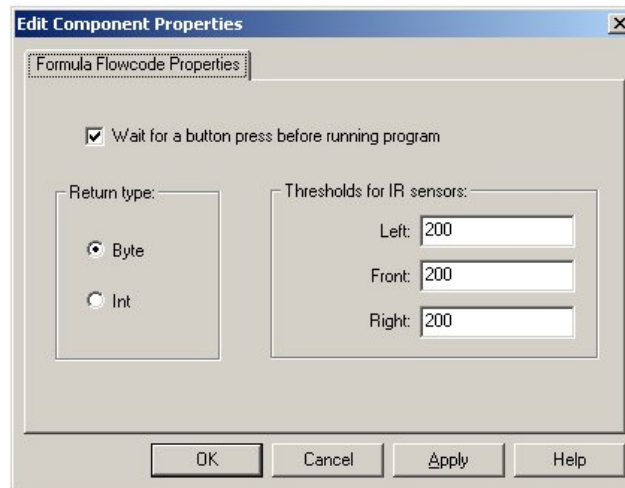
Le graphe ci dessous indique les valeurs, sur 10 bits, issues du capteur en fonction de la distance.

These readings are based on the distance away from a plain white wall in a well lit room using florescent tube lighting.

Ces valeurs sont obtenues en fonction de la distance à mur x blanc dans une pièce x éclairée par tubes fluorescents.



## 4.9) Component Properties



The Formula Flowcode Micromouse component has the following properties:

### 4.9.1) Wait for Button

If Ticked then the program will wait for button SW4 or SW5 to be pressed before running. If unticked then the program will run straight away after programming or a reset.

Si cette case est cochée le programme attend l'appui sur l'un des boutons poussoir SW4 ou SW5 pour démarrer après un reset. Si elle n'est pas cochée le programme démarre immédiatement après un reset.

### 4.9.2) Return Value

Configures the type of the variable returned from the ReadLDR and ReadIR Functions.

Byte - Returns the sensor readings as an 8 bit number (0 to 255)

Int - Returns the sensor readings as a 10 bit number (0 to 1023)

Cette option permet de choisir le format de la valeur retournée par les fonctions : ReadMic, ReadLDR et ReadIRSensor.

- Byte retourne un octet ( 8 bits, 0...255 ).
- Int retourne un mot de 10 bits ( 0...1023 ).

### 4.9.3) Threshold Values

Configures the cutoff threshold values for the left, front and right IR sensors. The threshold values can be either 8 bit (0 to 255) or 10 bit (0 to 1023) depending on the return value property.

The threshold is used as a simple way to:

A. Detect the prescence of a wall or obstacle.

B. Allow for inconsistencies between different buggies and different ambient light levels.

Ces trois valeurs sont les seuils de comparaison pour les capteurs IR. Ces valeurs sont sur 8 bits ( 0...255 ) ou sur 10bits ( 0...1024 ) en fonction de la propriété **Return Value**.

Ces seuils sont utilisés comme un moyen simple pour :

A. Détecter la présence d'un mur ou d'un obstacle.

B. Eviter les interférences entre les robots et tenir compte de l'éclairage ambiant.