

# Variables[indices]

## Événements

## Fonctions

explication,  
condition ou fréquence

de l'événement,  
{plage de valeurs}  
[unité]

variable mise à jour  
automatiquement

timer.period[0-1] [ms]  
timer0 chaque timer.period[0] ms  
timer1 chaque timer.period[1] ms

# Capteurs

prox.horizontal[0-4] {0...~4300}  
prox 10 Hz

button.forward {0,1}  
button.forward appuyé ou relâché

button.left {0,1}  
button.left appuyé ou relâché

button.center {0,1}  
button.center appuyé ou relâché

temperature [1/10 °C]  
temperature 1 Hz

button.backward {0,1}  
button.backward appuyé ou relâché

prox.horizontal[5-6] {0...~4300}  
prox 10 Hz

prox.ground.delta[0-1] =reflected-ambient  
prox.ground.reflected[0-1] {0...1023}  
prox.ground.ambient[0-1] {0...1023}  
prox 10 Hz

buttons 20 Hz

button.right {0,1}  
button.right appuyé ou relâché

rc5.address  
rc5.command  
rc5 signal reçu

mic.threshold {0..255}  
mic.intensity {0..255}

mic mic.intensity>mic.threshold  
sound.record(N) N: {0...32767}, enregistrer 'rN.wav'  
N=-1, arrêter l'enregistrement

acc[0-2] {-32...32}, 23=1g  
acc 16 Hz  
tap choc détecté

leds.prox.h(led0, led1, led2, led3, led4, led5, led6, led7) {0...32}

leds.buttons(led0, led1, led2, led3) {0...32}

leds.circle(led0, led1, led2, led3, led4, led5, led6, led7) {0...32}

leds.bottom.left(red, green, blue) {0...32}

leds.temperature(red, blue) {0...32}

motor.left.target vitesse souhaitée{-500...500}, 500 = ~20 cm/s

motor.left.speed vitesse réelle

motor.left.pwm commande moteur  
motor 100 Hz

leds.top(red, green, blue) {0...32}

leds.prox.h(led0, led1, led2, led3, led4, led5, led6, led7) {0...32}

leds.prox.v(led0, led1) {0...32}

leds.rc(led) {0...32}

leds.bottom.right(red, green, blue) {0...32}

leds.sound(led) {0...32}

motor.right.target vitesse souhaitée {-500...500}, 500 = ~20 cm/s

motor.right.speed vitesse réelle

motor.right.pwm commande moteur  
motor 100 Hz

sound.finished lecture d'un son terminée

sound.system(N) N: {0...7}, jouer le son système N. N=-1, arrêter

sound.freq(Hz,ds) [Hz],[1/60 s]

sound.wave(wave[142]) changer l'onde primaire, wave[i]: {-128...127}

sound.play(N) N: {0...32767}, jouer 'pN.wav'. N=-1, arrêter

sound.replay(N) N: {0...32767}, rejouer 'rN.wav'. N=-1, arrêter

# Actuateurs