Relie chaque couleur à son nom comme sur l'exemple

PEUREUX

Indice : quand il est coincé, Thymio sonne l'alarme

OBÉISSANT

Indice : Thymio change de direction lorsqu'on touche les flèches

AMICAL

Indice : quand Thymio détecte une main devant lui, il la suit

EXPLORATEUR

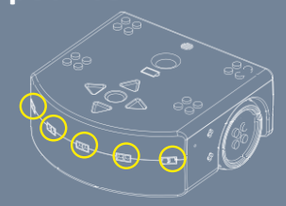
Indice : Thymio avance tout seul jusqu'à ce qu'il rencontre un obstacle

Annexe 1 - Donner un nom à chaque comportement

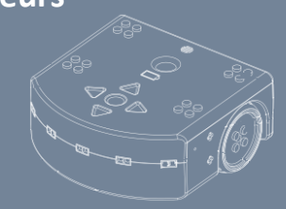
	SI	ALORS
AMICAL 	**s'il** détecte un objet devant lui **s'il** détecte un objet à droite **s'il** arrive au bord d'une table	il tourne à gauche il tourne à droite il avance il s'arrête
PEUREUX 	**s'il** détecte un objet devant lui **s'il** détecte un objet à droite **si** on tapote son dos **s'il** détecte un objet derrière lui	il recule il avance il tourne à droite il recule à gauche il fait du bruit
EXPLORATEUR 	**s'il** détecte un objet devant lui **s'il** détecte un objet à droite **s'il** détecte un objet à gauche **s'il** détecte un objet derrière lui **s'il** arrive au bord d'une table	il recule il s'arrête il tourne à gauche il tourne à droite il ne fait rien
OBÉISSANT 	**si** on appuie sur la flèche avant **si** on appuie sur la flèche arrière **si** on appuie sur la flèche de droite **si** on appuie sur la flèche de gauche	il avance il recule il tourne à gauche il tourne à droite il ne fait rien

Capteurs utilisés pour ce comportement

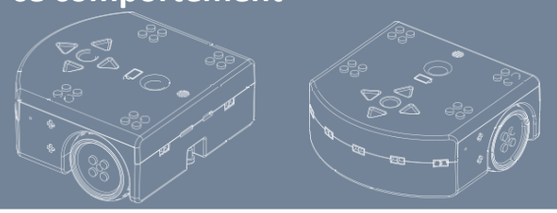
Tous les capteurs de distance à l'avant de Thymio



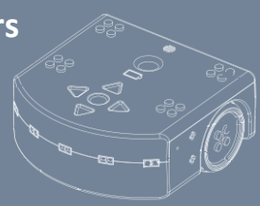
Entoure les capteurs utilisés pour ce comportement



Entoure les capteurs utilisés pour ce comportement



Entoure les capteurs utilisés pour ce comportement



Carte de référence VPL (Version 1.4)

Événements

Boutons touchés
gris: ignore le bouton, rouge: doit être touché

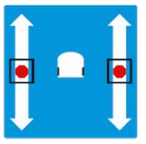
Détecteurs d'obstacle
gris: ignore, rouge: objet proche, blanc: objet loin

Détecteurs de sol
gris: ignore, rouge: sol, blanc: pas de sol

Robot tapé
Le robot a reçu un choc.

Claquement de main
Le roobt a entendu un fort bruit.

Actions



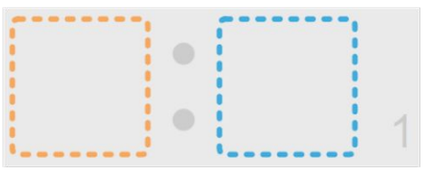
Vitesse des moteurs
Défini la vitesse des moteurs et roues gauches et droites.

Couleur du haut
Colore le haut avec un mélange de rouge, vert et bleu.

Couleur du bas
Colore le bas avec un mélange de rouge, vert et bleu.

Jouer une musique
Choisir la hauteur, blanche deux fois la durée de noire.

Construire votre programme



Glissez/déposez des événements dans le carré gauche, des actions dans le carré droite.
Lorsque l'événement se produit, le robot fait l'action.
Multiples actions associées à un seul événement.
Lorsque l'événement se produit, le robot fait toutes les actions.

Les capteurs sont combinés avec ET dans un événement

Si deux capteurs sont séléctionnés, les deux conditions doivent être vraies pour que l'événement se produise.
Gauche **et** droite doivent être touchés/avoir un objet à proximité.

M. Ben-Ari, S. Magnenat, J. Shin - CC-BY-SA

Mode Avancé

Événements

Détecteurs d'obstacle
Les curseurs définissent les seuils haut (objet proche) et bas (objet loin)

Détecteurs de sol
Les curseurs définissent les seuils haut (blanc / sol proche) et bas (noir / sol loin)

Robot tapé
Le robot a reçu un choc.

Accéléromètre tangage
Le tangage (avant / arrière) est dans le segment rouge.

Accéléromètre roulis
Le roulis (droite / gauche) est dans le segment rouge.

Temps écoulé
Le temps du minuteur est écoulé.

Actions

Démarrer un minuteur
Un événement temps écoulé se produira après un temps.

Définir l'état du robot
Défini les 4 bits de l'état interne du robot.

M. Ben-Ari, S. Magnenat, J. Shin - CC-BY-SA

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 every timer.period[0] ms
timer1 every timer.period[1] ms

Capteurs

prox.comm.tx {0,2047}
prox.comm.rx
prox 10 Hz
prox.comm.enable(state) {0,1}

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

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

buttons 20 Hz

button.forward {0,1}
button.forward pressed or released

button.right {0,1}
button.right pressed or released

button.left {0,1}
button.left pressed or released

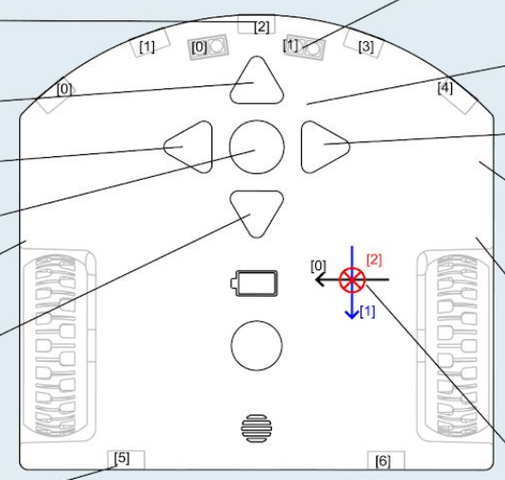
button.center {0,1}
button.center pressed or released

rc5.address
rc5.command
rc5 signal received

temperature [1/10 °C]
temperature 1 Hz

mic.threshold {0..255}
mic.intensity {0..255}
mic mic.intensity>mic.threshold
sound.record(N) N: {0...32767}, record as 'rN.wav'. N=-1, stop recording

button.backward {0,1}
button.backward pressed or released



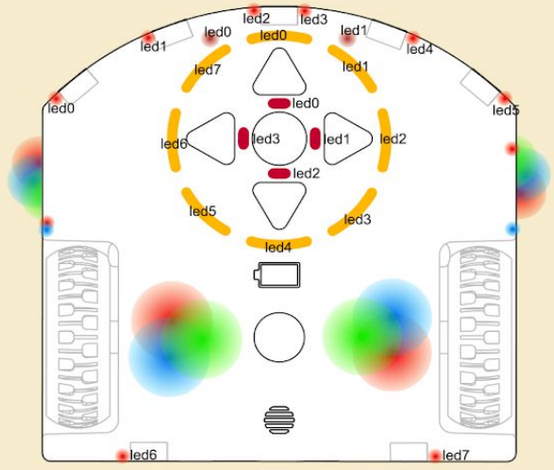
prox.horizontal[5-6] {0...~4300}
prox 10 Hz

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

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

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

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



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

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

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

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

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

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

motor.right.target desired speed {-500...500}, 500 = ~20 cm/s
motor.right.speed actual speed
motor.right.pwm motor command
motor 100 Hz

motor.left.target desired speed {-500...500}, 500 = ~20 cm/s
motor.left.speed actual speed
motor.left.pwm motor command
motor 100 Hz

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

sound.finished a sound finished playing
sound.system(N) N: {0...7}, play system sound N. N=-1, stop playing
sound.freq(Hz,ds) [Hz],[1/60 s]
sound.wave(wave[142]) change primary wave, wave[i] : {-128...127}
sound.play(N) N: {0...32767}, play 'pN.wav'. N=-1, stop playing
sound.replay(N) N: {0...32767}, replay 'rN.wav'. N=-1, stop playing

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

Actuateurs

Annexe 7 - Documentation du langage de Thymio