```
<!DOCTYPE aesl-source>
<network>


<!--list of global events-->
<constant value="2" name="BACKWARD"/>
<constant value="500" name="BLACK_THRESHOLD"/>
<constant value="12" name="CALIB"/>
<constant value="1" name="FORWARD"/>
<constant value="3" name="LEFT"/>
<constant value="-500" name="MINUS_COUNTER"/>
<constant value="0" name="NOTHING"/>
<constant value="4" name="RIGHT"/>
<constant value="0" name="ROUGE"/>
<constant value="500" name="SAFETY_TIME"/>
<constant value="200" name="SPEED"/>
<constant value="13" name="START"/>
<constant value="10" name="START_UP"/>
<constant value="16" name="TABLE_SIZE"/>
<constant value="14" name="TYPING"/>
<constant value="1" name="VERT"/>
<constant value="11" name="WAIT"/>


<!--node Thymio Beebot-->
<node name="Thymio Beebot" nodeId="{177cf107-e5a0-46d5-aa69-ecfd18fb9cac}"><![CDATA[########## CODE THYMIO BEEBOT-LIKE ####################
#
# Code écrite par Christophe Barraud dernière modification le 2.11.2018
# Code modifier par Michael Bonani 08.12.2021
#
################################################

var mode = START_UP
var submode = 0
var counter = 0
var led[3] = [0,0,0]

var table[TABLE_SIZE]
var i = 0
var j = 0

var circle_counter
var circle_intensity
var circle_turn
var circle_led[8]=[0,0,0,0,0,0,0,0]
var circle_i

var counter_calib = 0

var led_clignote = ROUGE

var time_straight_15cm = 1250
var time_turn_360 =2370
var time_turn_90 = 590

var status_SD = 0
var data[2] = [0,0]
var read = 0
var written = 0

timer.period[0] = 2
timer.period[1] = 500

circle_intensity = 256/TABLE_SIZE

sub led
call leds.bottom.left(led[0],led[1],led[2])
call leds.bottom.right(led[0],led[1],led[2])
call leds.top(led[0],led[1],led[2])

sub read_values
if sd.present==1 then
	call sd.open(0,status_SD)
	if status_SD == 0 then
		call sd.read(data,read)
		if read == 2 then
			time_straight_15cm = data[0]
			time_turn_90 = data[1]
		end
	end
	call sd.open(-1,status_SD)
else
	call _system.settings.read(20,data[0])
	call _system.settings.read(21,data[1])
	if data[0]!=0 then
		time_straight_15cm = data[0]
	end
	if data[1]!=0 then
		time_turn_90 = data[1]
	end
end

sub write_values
data[0] = time_straight_15cm
data[1] = time_turn_90
if sd.present==1 then
	call sd.open(0,status_SD)
	call sd.write(data,written)
	if written == 2 then
		call sound.play(1)
		written = 0
	end
	call sd.open(-1,status_SD)
else
	call _system.settings.write(20,data[0])
	call _system.settings.write(21,data[1])
	call _system.settings.flash()
	call sound.play(1)
end

sub show_counter
for  circle_i in 0:7 do
	circle_led[circle_i]=((circle_counter+7-circle_i)/8)*circle_intensity
end
call leds.circle(circle_led[0],circle_led[1],circle_led[2],circle_led[3],circle_led[4],circle_led[5],circle_led[6],circle_led[7])

onevent timer0
if mode == START_UP then
	callsub read_values
	call math.fill(table,0)
	mode = WAIT
```

```
elseif mode == WAIT then
		led = [0,0,0]
		call leds.circle(0,0,0,0,0,0,0,0)
		callsub led
		counter = 0
		motor.left.target = 0
		motor.right.target = 0
elseif mode == CALIB then
		counter++
		if submode == 1 and prox.ground.delta[0] > BLACK_THRESHOLD and prox.ground.delta[1] < BLACK_THRESHOLD then
				led_clignote = VERT
		elseif submode == 1 then
				led_clignote = ROUGE
		end
		if submode == 2  and prox.ground.delta[0] < BLACK_THRESHOLD and prox.ground.delta[1] < BLACK_THRESHOLD then
				counter = 0
				timer.period[1] = 0
				led = [15,15,0]
				callsub led
				submode = 3
		elseif submode == 3 and counter > SAFETY_TIME and prox.ground.delta[0] < BLACK_THRESHOLD and prox.ground.delta[1] < BLACK_THRESHOLD then
				time_straight_15cm = counter
				motor.left.target =0
				motor.right.target = 0
				counter = 0
				submode = 0
				callsub write_values
				mode = WAIT
		elseif submode == 4 and counter > SAFETY_TIME and  prox.ground.delta[0] > BLACK_THRESHOLD and prox.ground.delta[1] < BLACK_THRESHOLD then
				motor.left.target = 0
				motor.right.target = 0
				 time_turn_360=counter
				call math.div(time_turn_90,time_turn_360,4)
				submode = 0
				callsub write_values
				mode = WAIT
		end
		callsub led
elseif mode == START then
		if counter >= 0 then
				when counter==0 do
						if table[j] == FORWARD then
								call sound.system(7)
						elseif table[j] == BACKWARD then
								call sound.system(6)
						elseif table[j] == LEFT then
						   call sound.system(4)
						elseif table[j] == RIGHT then
								call sound.system(4)
						end
				end
				if table[j] == FORWARD then
						motor.left.target = SPEED
						motor.right.target = SPEED
						if counter >= time_straight_15cm then
								j++
								counter=MINUS_COUNTER
								motor.left.target = 0
								motor.right.target = 0
								circle_counter=j
								callsub   show_counter
						end
						led = [0,31,0]
						callsub led
				elseif table[j] == BACKWARD then
						motor.left.target = -SPEED
						motor.right.target = -SPEED
						if counter >= time_straight_15cm then
								j++
								counter=MINUS_COUNTER
								motor.left.target = 0
								motor.right.target = 0
								circle_counter=j
								callsub   show_counter
						end
						led = [31,0,0]
						callsub led
				elseif table[j] == LEFT then
						motor.left.target = -SPEED
						motor.right.target = SPEED
						if counter >= time_turn_90 then
								j++
								counter=MINUS_COUNTER
								motor.left.target = 0
								motor.right.target = 0
								circle_counter=j
								callsub   show_counter
						end
						led = [0,31,31]
						callsub led
				elseif table[j] == RIGHT then
						motor.left.target = SPEED
						motor.right.target = -SPEED
						if counter >= time_turn_90 then
								j++
								counter=MINUS_COUNTER
								motor.left.target = 0
								motor.right.target = 0
								circle_counter=j
								callsub   show_counter
						end
						led = [31,31,0]
						callsub led
				else
						call sound.system(0)
						j=0
						i=0
						call math.fill(table,0)
						mode=WAIT
				end
		end
		counter++
end



########## TIMER 1 CLIGNOTEMENT ###########
onevent timer1
if mode == CALIB then
		if led_clignote == ROUGE then
```

```
                        if led[0] > 0 then
                                led = [0,0,0]
                        else
                                led = [31,0,0]
                        end
        else
                        if led[1] > 0 then
                                led = [0,0,0]
                        else
                                led = [0,31,0]
                        end
        end
end

########## BOUTONS ##############
onevent buttons
if button.left == 1 and button.right == 1 then
        counter_calib++
        if counter_calib > 100 then
                mode = CALIB
                timer.period[1] = 500
                submode = 1
                counter_calib = 0
        end
else
        counter_calib = 0
end
if mode == TYPING then
        when button.forward == 1 do
                table[i] = FORWARD
                i++
        end
        when button.backward == 1 do
                table[i] = BACKWARD
                i++
        end
        when button.left == 1 do
                table[i] = LEFT
                i++
        end
        when button.right == 1 do
                table[i] = RIGHT
                i++
        end
        if i >= TABLE_SIZE then
                mode = START
                j=0
                counter=0
        end
        circle_counter=i
        callsub   show_counter
end

onevent button.forward
if button.forward == 1 then
        call sound.system(2)
        if mode == CALIB then
                if submode == 1 and prox.ground.delta[0] > BLACK_THRESHOLD and prox.ground.delta[1] < BLACK_THRESHOLD then
                        submode = 2
                        counter = -1
                        motor.right.target = SPEED
                        motor.left.target = SPEED
                end
        end
end

onevent button.left
if button.left == 1 then
        call sound.system(2)
        if mode == CALIB then
                if submode == 1 and prox.ground.delta[0] > BLACK_THRESHOLD and prox.ground.delta[1] < BLACK_THRESHOLD then
                        submode = 4
                        counter = -1
                        motor.right.target = SPEED
                        motor.left.target = -SPEED
                end
        end
end

onevent button.backward
if button.backward == 1 then
        call sound.system(2)
end

onevent button.right
if button.right == 1 then
        call sound.system(2)
end


onevent button.center
if button.center == 1 then
        call sound.system(3)
        if mode == WAIT then
                mode = TYPING
                j=0
                counter=0
                led = [31,31,31]
                callsub led
        elseif mode == TYPING then
                mode = START
                circle_counter=j
                callsub   show_counter
        else
                mode = WAIT
                i=0
                call math.fill(table,0)
        end
end]]></node>


</network>
```