

```

var notes[6]
var durations[6]
var note_index = 6
var note_count = 6
var wave[1442]
var i
var wave_phase
var wave_intensity



# compute a sinus wave for sound
for i in 0:1441 do
  wave_phase = (i-70)*468
  call math.cos(wave_intensity, wave_phase)
  wave[i] = wave_intensity/256
end
call sound.wave(wave)
# reset outputs
call sound.system(-1)
call leds.top(0,0,0)
call leds.bottom.left(0,0,0)
call leds.bottom.right(0,0,0)
call leds.circle(0,0,0,0,0,0,0)

# when a note is finished, play the next note
onevent sound.finished
  if note_index != note_count then
    call sound.freq(notes[note_index],
      durations[note_index])
    note_index += 1
  end

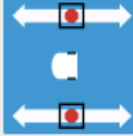


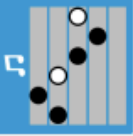
onevent buttons
  when button.backward == 1 do
    motor.left.target = -350
    motor.right.target = -350
    call leds.top(32,0,0)
    emit pair_run 0
  end






onevent tap
  motor.left.target = 350
  motor.right.target = 350
  call leds.top(0,32,0)
  emit pair_run 1

onevent prox
  when prox.horizontal[3] >= 2000 and
  prox.horizontal[4] >= 2000 do
    motor.left.target = 0
    motor.right.target = 0
  
```






Actions

Compilation terminée avec succès

Evénements

