

```

# compute a sinus wave for sound
for i in 0:141 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
end

onevent buttons
  when button.backward == 1 do
    motor.left.target = 150
    motor.right.target = 150
    call leds.top(0,32,0)
    emit_pair_run 0
  end
end

onevent prox
  when prox.horizontal[5] >= 2000 and
  prox.horizontal[6] >= 2000 do
    motor.left.target = 0
    motor.right.target = 0
    call leds.bottom.left(0,0,32)
    call leds.bottom.right(0,0,32)
    emit_pair_run 1
  end
end

onevent tap
  call math.copy(notes[0:5], [440, 524, 440, 370,
311, 370])
  call math.copy(durations[0:5], [7, 7, 14, 7, 14,
14])
  note_index = 1
  note_count = 6
  call_sound.freq(notes[0], durations[0])
  emit_pair_run 2

```

Evénements

Actions

Compilation terminée avec succès

Evénements

Actions