```
__author__ = 'kanehiro'
from ev3dev.auto import *
from time import sleep
class GoAround:
  SPEED = 50
  def __init__(self):
    self.motors = [LargeMotor(address) for address in (OUTPUT_B,
OUTPUT_C)]
    assert all([m.connected for m in self.motors]),\
      "Two large motors should be connected to ports B and C"
    self.us = UltrasonicSensor(); assert self.us.connected
  def green(self):
    for light in (Leds.LEFT, Leds.RIGHT):
      Leds.set_color(light, Leds.GREEN)
  def red(self):
    for light in (Leds.LEFT, Leds.RIGHT):
      Leds.set_color(light, Leds.RED)
  def start(self):
    # Turn Forward lights on:
    self.green()
    # Start Alarm
    self.alarm(3)
    sleep(3)
    for m in self.motors:
      m.run_direct(duty_cycle_sp=self.SPEED)
  def forward(self):
    # Turn Forward lights on:
    self.green()
    for m in self.motors:
      m.run_direct(duty_cycle_sp=self.SPEED)
  def back(self):
    self.red()
    self.alarm(2)
    for m in self.motors:
      m.run_timed(duty_cycle_sp=self.SPEED * -1, time_sp=1000)
    # When motor is stopped, its `state` attribute returns empty
list.
    # Wait until both motors are stopped:
    # any([True,True,False]) # => True
    self.motor_wait(0.1)
  def turn(self):
    self.green()
    speed = self.SPEED
    for m in self.motors:
      m.run_timed(duty_cycle_sp=speed, time_sp=500)
      speed *= -1
    self.motor_wait(0.1)
  def motor_wait(self,interval):
    while any(m.state for m in self.motors):
      sleep(interval)
  def alarm(self,count):
    # Start Alarm
```

```
    Sound.tone([(1000, 500, 500)] * count)
  def stop(self):
    Leds.all_off()
    for m in self.motors:
      m.stop(stop_command='brake')
  @property
  def us_value(self):
    """
    Check if 'up' button is pressed.
    """
    return self.us.value()
if __name__ == "__main__":
  # We will need to check EV3 buttons state.
  btn = Button()
  gd = GoAround()
  gd.start()
  while not btn.any():
    distance = gd.us_value
    # Go Forward
    if distance < 100:
      gd.back()
      gd.turn()
    gd.forward()
  gd.stop()
```