```
import time

import random

from geopy.distance import geodesic


# Import the necessary libraries from the first pseudocode

import sensors

import communication

import vehicle_control


# Define global variables from the second pseudocode

emergency_button_pressed = False

driver_location = (0, 0)

medical_sensors_data = {}  # Placeholder for medical sensor data


# Define system parameters from the first pseudocode

emergency_button = Button()

seatbelt_sensor = MedicalSensor()

communication_module = CommunicationModule()

vehicle_controller = VehicleController()

time_limit = 60  # seconds


# Main system loop

while True:

    # Simulate periodic checks and actions from the second pseudocode

    emergency_button_pressed = emergency_button.is_pressed()

    sensors.read_medical_sensor_data()
    sensors.analyze_medical_sensor_data()

    if emergency_button_pressed:
        # Driver initiated emergency

        # Notify the nearest hospital (combine the logic from both pseudocodes)
        hospital = find_nearest_hospital()
        communication_module.notify_hospital(hospital)

        # Start a timer for the driver's response (from the first pseudocode)
        response_timer = sensors.start_timer()

        while not response_timer.is_expired():
            if emergency_button.is_pressed():
                response_timer.reset()
            else:
                vital_signs = seatbelt_sensor.measure_vital_signs()
                if is_emergency(vital_signs):
                    # Automatic intervention required (from the first pseudocode)
                    communication_module.notify_emergency_services()
                    vehicle_controller.take_control()
                    vehicle_controller.safely_stop_vehicle()
                    break  # Exit the loop

        if response_timer.is_expired():
            # Driver didn't respond in time, take necessary actions (from the first
pseudocode)
```

```
            communication_module.notify_emergency_services()
            vehicle_controller.take_control()
            vehicle_controller.safely_stop_vehicle()

    else:
        # Regular operation
        continue

    # Simulate providing feedback to the driver from the second pseudocode
    sensors.provide_user_feedback()

    # Simulate periodic checks
    time.sleep(5)

# Functions for specific tasks from the first pseudocode
def find_nearest_hospital():
    # Logic to determine the nearest hospital based on the vehicle's GPS
coordinates
    pass

def start_timer():
    # Start a timer for the predefined time limit
    pass

def is_emergency(vital_signs):
    # Analyze the vital signs to detect if there's an emergency
    pass
```