import can

import time

def detect\_ecu\_vcan(interface="socketcan", channel="vcan0", timeout=5):

"""

Detect if an ECU is connected (or simulates activity) on the virtual CAN bus.

Parameters:

- interface: The CAN interface type (default is 'socketcan' for virtual CAN).

- channel: The CAN channel to use (default is 'vcan0').

- timeout: Time (in seconds) to listen for CAN messages.

Returns:

- bool: True if ECU activity is detected, False otherwise.

"""

try:

print(f"Connecting to virtual CAN interface: {channel}")

bus = can.Bus(interface=interface, channel=channel)

print("Listening for CAN messages...")

start\_time = time.time()

while time.time() - start\_time < timeout:

msg = bus.recv(timeout=1) # Wait for a message with a 1-second timeout

if msg:

print(f"Message received: {msg}")

print("ECU activity detected on the CAN bus.")

return True

print("No CAN messages detected within the timeout period.")

return False

except can.CanError as e:

print(f"Error connecting to the CAN interface: {e}")

return False

finally:

if 'bus' in locals():

bus.shutdown()

# Run the ECU detection function on vcan0

if \_\_name\_\_ == "\_\_main\_\_":

if detect\_ecu\_vcan():

print("ECU is connected and active (or simulated) on the virtual CAN bus.")

else:

print("No ECU activity detected on the virtual CAN bus.")