CAN Testing using Canoe Tool

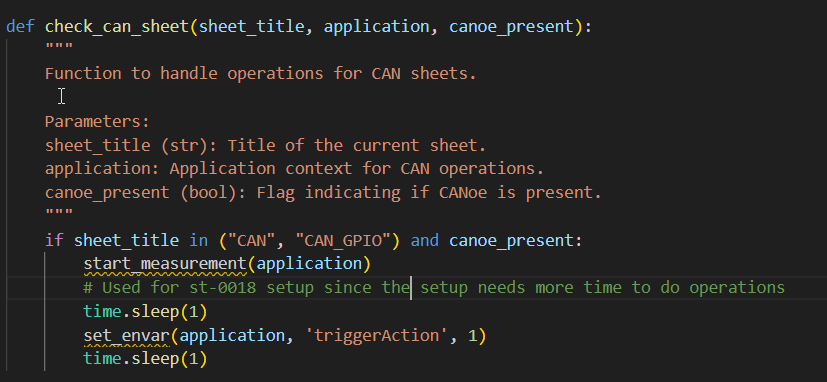
# Prerequisites

* A valid Canoe license is required. (Currently, the license is available only for st-0026 (B1 setup); in st-0028 no Canoe license is available, so CAN testing cannot be performed.)
* Canoe configuration files for all three samples are located at:
* test\_scripts\tester\VPC-P\_Testing\_Framework\config\board\_specifications\<sample-name>\canoe\_config\canoe\_<sample\_name>.cfg
* CAPL scripts are located at:
* test\_scripts\tester\VPC-P\_Testing\_Framework\config\board\_specifications\<sample-name>\canoe\_config\capl\_scripts
* The CAN database is located at:
* test\_scripts\tester\VPC-P\_Testing\_Framework\config\board\_specifications\<sample-name>\canoe\_config\CANdb\easy.dbc

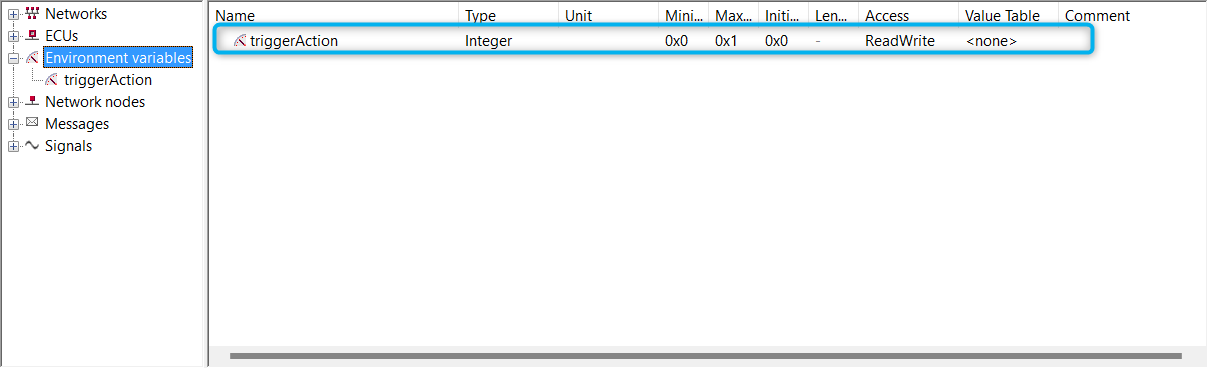
# Execution Flow of CAN Tests

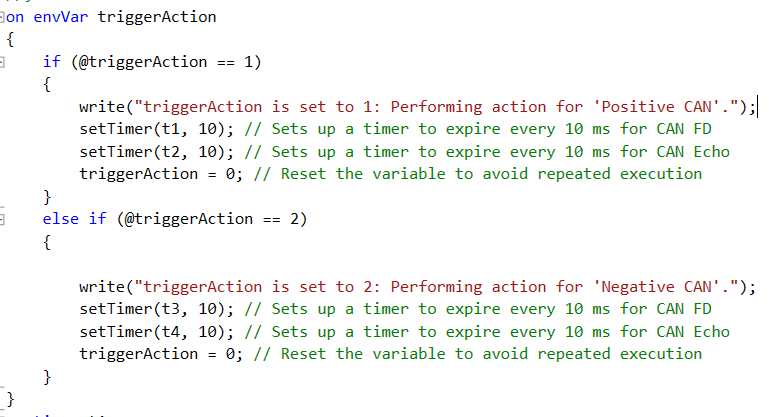
1. In send.py, the function check\_can\_sheet(sheet.title, application, canoe\_present) is continuously executed to determine if the sheet is either 'CAN' or 'CAN\_GPIO'.

2. If the sheet matches, check\_can\_sheet is invoked and start\_measurement() is called to start Canoe and also set the environment variable triggerAction to 1.



3. triggerAction is already initialized in easy.dc hence once triggerAction is set to 1, the capl script which have on envVar handle will be executed





4. start\_measurement() initiates measurement by calling **application.Measurement.Start()** instructing CANoe to begin its measurement cycle and activate all defined CAPL scripts.

5. As part of CANoe's configuration, CAPL scripts and their event handlers (e.g., on start, on envVar, including the one listening on triggerAction) are pre-loaded in memory and execute when their conditions are met.

6. After triggerAction is set via setenv(), the on-envVar handler verifies its value and activates the timers accordingly.

7. Timer t2 is used to send CAN echo frames and timer t1 is used to send CAN FD frames.

8. For CAN echo frames, the DLC (Data Length Code) is set to 8; for CAN FD frames, the DLC is set to 15.

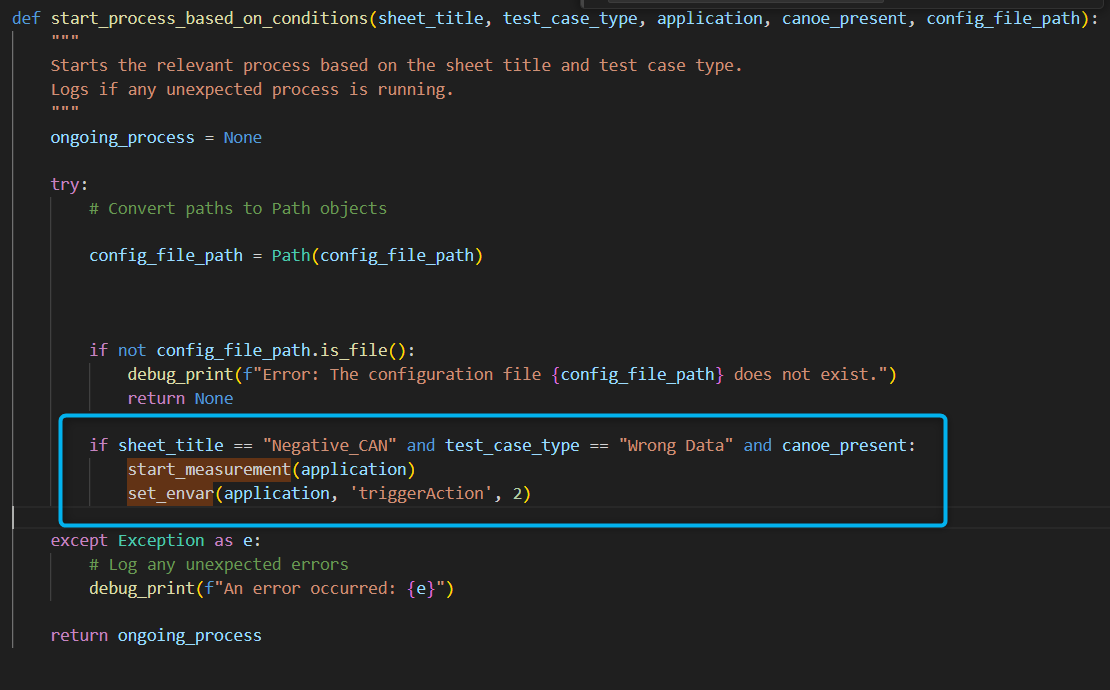
9. For CAN FD frames, FDF (FD Format) should be set to 1.

10. For CAN FD frames, BRS (Bit Rate Switching) should be set to 1 to allow faster transmission of the payload.

11. For CAN FD (positive) data, the payload values should range from 0xFE down to 0xBF.

12. For CAN echo, the payload values should range from 0xA0 to 0xA7.

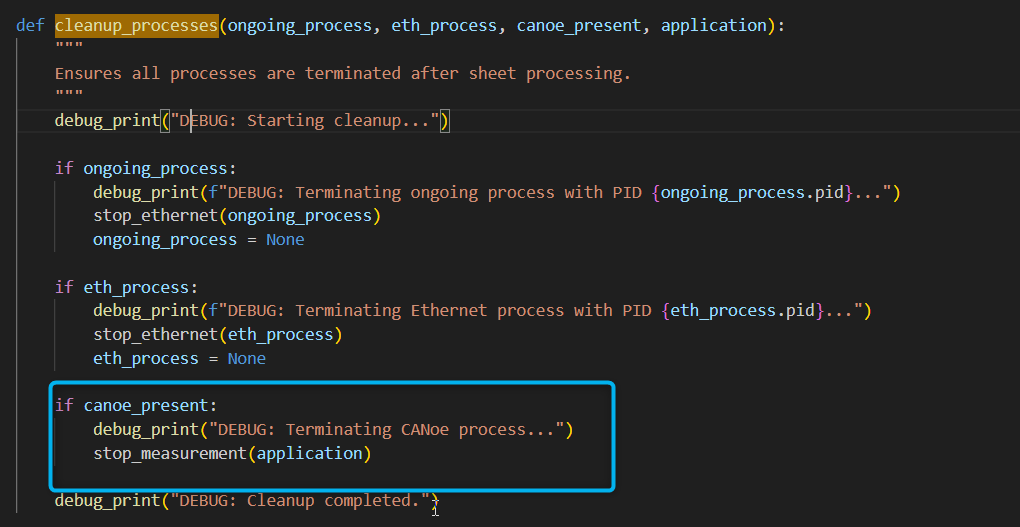
13. For sheets with Negative\_CAN, start\_process\_based\_on\_conditions() is called, which executes only if the sheet name is 'Negative\_CAN' and the test\_case\_type is 'Wrong Data'.



14. for wrong data test cases, timers t3 and t4 are called which have the wrong data values

15. If the test\_case\_type is 'No\_Response', no action is taken.

16. Once all test cases in the CAN, CAN\_GPIO, and Negative\_CAN sheets are processed, cleanup\_processes(ongoing\_process, eth\_process, canoe\_present, application) is invoked. If canoe\_present is true, stop\_measurement() is then called to stop Canoe without closing the application.



17. Finally, finalize\_test(canoe\_present, application, tester) is called to close the Canoe application using close\_application().

